

**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Κίνηση ρομποτικού βραχίονα με κινητή συσκευή Android και
Bluetooth**

Gloria Dwomoh

Εισηγητής: Δρ. Ιωάννης Ν. Έλληνας, Καθηγητής

**ΠΕΙΡΑΙΑΣ
ΟΚΤΩΒΡΙΟΣ 2017**

Κίνηση Ρομποτικού Βραχίονα με Κινητή Συσσκευή Android και Bluetooth

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Κίνηση ρομποτικού βραχίονα με κινητή συσκευή Android και Bluetooth

Gloria Dwomoh

A.M. 40814

Εισηγητής:

Δρ. Ιωάννης Ν. Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης

(Κενό φύλλο)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑ

Η κάτωθι υπογεγραμμένη Gloria Dwomoh, του Francis, με αριθμό μητρώου 40814 φοιτήτρια του Τμήματος Μηχανικών Η/Υ Συστημάτων 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 απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση 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.»

(Κενό φύλλο)

ΕΥΧΑΡΙΣΤΙΕΣ

Αφιερώνω αυτήν την πτυχιακή στη μητέρα μου, Comfort. Κατά τη διάρκεια των εκπαιδευτικών μου χρόνων, μπορώ να πω ότι υπήρξε ένας από τους μεγαλύτερους θεμελιώδεις υποστηρικτές μου. Είναι αναζωογονητικό να έχω κάποιον που υποστηρίζει τα όνειρα μου παρόλη την αβεβαιότητα μου. Όποτε αισθανόμουν ότι θέλω να τα εγκαταλείψω, θυμόμουν την προσπάθεια της, τη σκληρή δουλειά και τη δύναμη που κατέβαλε για να διασφαλίσει την εκπαίδευση μου και αυτό με ώθησε να συνεχίσω να αγωνίζομαι.

Θέλω να πω ένα μεγάλο ευχαριστώ στην υπόλοιπη οικογένεια μου για την υποστήριξη τους. Ειδικότερα, τα αδέρφια μου και τον πατέρα μου που μου έδωσαν την ευκαιρία να μελετήσω ακόμη και σε στιγμές που αυτό ήταν αρκετά δύσκολο.

Στους υποστηρικτές και τους αγαπητούς ανθρώπους που γνώρισα κατά τη διάρκεια των σπουδών μου, σας εύχομαι ό,τι καλύτερο στη ζωή.

Θέλω επιπλέον να εκτιμήσω τον επιβλέποντα καθηγητή μου για την υπομονετική και ταχεία ανταπόκριση στα ερωτήματά μου, όποτε υπήρχε ανάγκη, και όλα τα μέλη των διδασκόντων που ήταν υποστηρικτικά και ευγενικά σε εμένα κατά τη διάρκεια των σπουδών μου.

Τέλος, ευχαριστώ τον Θεό που μου έδωσε τη χάρη να μην τα εγκαταλείψω. Έμεινα αληθινή στον εαυτό μου, και τήρησα την υπόσχεσή μου στο να αποφοιτήσω, ό,τι κι αν χρειαστεί. Θέλω να πιστεύω ότι το μέλλον θα είναι λαμπρό.

VOTE OF THANKS

I am dedicating this thesis to my mother, Comfort. Throughout my educational years, I can say she has been one of my biggest fundamental supporters. It is refreshing to have someone support the dreams that I was unsure of myself. Whenever I felt like giving up I remembered the effort, hard work, and strength she placed to ensure that I have a formal education, and that truly motivated me to keep fighting. Thank you, mum!

I want to say a big thank you to the rest of my family for their support. My siblings and father, in particular, for giving me the chance to study even at moments when that was a hard thing to do.

To the supportive and loving people, that I came to know during my studies, I wish you all the very best in life.

I want to appreciate my supervising professor for patiently and swiftly responding to my queries whenever there was a need and all the faculty members that were supportive and kind to me during my studies.

Lastly, I thank God for giving me the grace to not give up. I stayed true to myself, and kept my promise to graduate, whatever it took. I am looking forward to what lies ahead. I am hopeful that the future will be bright.

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια, αρκετοί από εμάς έχουμε δει την αύξηση των προϊόντων που χρησιμοποιούν ασύρματη επικοινωνία. Αυτό το έργο καταδεικνύει έναν πιθανό τρόπο ελέγχου ενός ρομποτικού βραχίονα μέσω της επικοινωνίας Bluetooth μεταξύ μιας εφαρμογής Android και ενός Arduino. Για αυτό, κάνω χρήση του ρομποτικού βραχίονα OWI, καθώς είναι μια προσιτή επιλογή που το καθιστά ιδανικό για έργα μικρής κλίμακας. Σ' αυτό το βιβλίο, θα περάσουμε από τη βασική ιστορία των ρομποτικών βραχιόνων ακολουθούμενη από την επεξήγηση μερικών ηλεκτρονικών εξαρτημάτων, τα πιο σημαντικά κομμάτια του κώδικα που χρησιμοποιήσα για τον έλεγχο του βραχίονα και τέλος τα στάδια υλοποίησης του έργου.

Λέξεις Κλειδιά: ρομποτική, μικροεπεξεργαστής, βραχίονας, πομποδέκτης, εφαρμογή

Επιστημονική Περιοχή: Ρομποτική, μηχανική λογισμικού και ενσωματωμένα συστήματα.

ABSTRACT

In the recent years, many of us have witnessed the increase of products that make use of wireless communication. This project demonstrates a possible way to manoeuvre a robotic arm through Bluetooth communication between an Android application and an Arduino circuit. For this, I make use of an OWI robotic arm as it is an affordable option which makes it perfect for small scale projects. In this booklet, you will go through some basic history of robotic arms, followed by what each of the major electronic components does, the most important pieces of code that involve the control of the arm, and finally, you will get a walkthrough of how to this project was created.

Keywords: Arduino, Bluetooth, Android, Robotic, Arm

Scientific Field :Robotics, software engineering, and embedded systems.

(Κενό φύλλο)

ΠΕΡΙΕΧΟΜΕΝΑ

1 - ΕΙΣΑΓΩΓΗ.....	13
1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας.....	13
1.2 Ιστορική αναδρομή	13
2 - ΕΞΟΠΛΙΣΜΟΣ/ΥΛΙΚΟ/ΛΟΓΙΣΜΙΚΟ	15
2.1 Arduino	15
2.2 Bluetooth.....	19
2.3 OWI robotic arm edge.....	25
2.4 Προγραμματισμός για Arduino.....	27
2.5 Android	29
3 - ΕΡΓΑΣΙΑ.....	31
3.1 Το κύκλωμα	31
3.2 Ο κώδικας Arduino.....	34
3.3 Ο κώδικας Android.....	37
4 - ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ	57
4.1 Σύνοψη της πτυχιακής εργασίας.....	57
4.2 Προοπτικές	57
5 - ΠΑΡΑΡΤΗΜΑ Α'	58
6 - ΒΙΒΛΙΟΓΡΑΦΙΑ.....	59

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1.1: Το Unimate.....	13
Εικόνα 2.1: Το Λογότυπο του Arduino	15
Εικόνα 2.2: Το Arduino UNO.....	16
Εικόνα 2.3: Το Arduino UNO σε κοντινή λήψη	18
Εικόνα 2.4: Το Λογότυπο του Bluetooth.....	19
Εικόνα 2.5: Η Master – Slave τοπολογία.....	20
Εικόνα 2.6: Το HC-05 Bluetooth σε κοντινή λήψη	22
Εικόνα 2.7: Το OWI robotic arm edge kit	25
Εικόνα 2.8: Οι 12 διαφορετικές κινήσεις του βραχίονα	26
Εικόνα 2.9: Διαδικασία εισαγωγής κώδικα σε πλακέτα Arduino	27
Εικόνα 2.10: Το Λογότυπο του Android.....	29
Εικόνα 3.1: Μερικά από τα υλικά κατασκευής της εργασίας.....	31
Εικόνα 3.2: Το ολοκληρωμένο L2930NE.....	32
Εικόνα 3.3: Το εσωτερικό κύκλωμα του L2930NE.	32
Εικόνα 3.4: Το κύκλωμα σχεδιασμένο στο Fritzing.	33
Εικόνα 3.5: Ενεργοποίηση πομποδέκτη κινητού.....	38
Εικόνα 3.6: Αρχική σελίδα εφαρμογής	39
Εικόνα 3.7: Πάτημα αντιστοίχισης.....	41
Εικόνα 3.8: Ρυθμίσεις Bluetooth.....	41
Εικόνα 3.9: Ζήτηση κωδικού για αντιστοίχιση	42
Εικόνα 3.10: Πάτημα κωδικού για αντιστοίχιση	42
Εικόνα 3.11: Μενού για αντιστοίχιση	43
Εικόνα 3.12: Μενού ελέγχου για το ρομποτικό βραχίονα	44
Εικόνα 3.13: Σχήμα ονομασιών αρθρώσεων βραχίονα.....	45
Εικόνα 3.14: Σελίδα About.....	47
Εικόνα 3.15: Εικονίδιο εφαρμογής	48

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

STEM Science, Technology, Engineering, and Mathematics

VCC Voltage at the Common Collector

PWM Pulse – Width Modulation

MAC Media Access Control

AT Attention Command

TXD Transmit Data

RXD Receive Data

DC Direct Current

GND Ground

EN Enable

(Κενό φύλλο)

ΚΕΦΑΛΑΙΟ 1

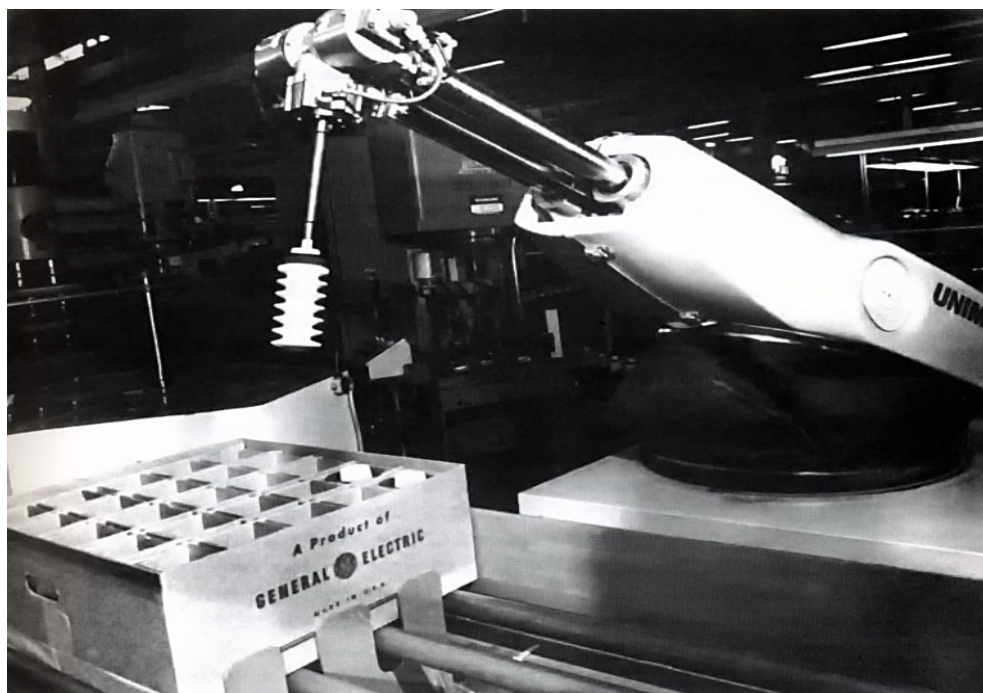
ΕΙΣΑΓΩΓΗ

1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η κίνηση ρομποτικού βραχίονα με την χρήση κινητής συσκευής Android και Bluetooth. Σύμφωνα με αυτή την πτυχιακή θα είναι δυνατή η δημιουργία, η παραλλαγή ή επέκταση αυτής της εργασίας σε άλλα έργα που χρησιμοποιούν παρόμοιες τεχνολογίες. Παράλληλα, μέσω αυτής έχουμε την ευκαιρία να διερευνήσουμε διάφορες τεχνολογίες και να δούμε πως μπορούν να αλληλοεπιδράσουν μεταξύ τους ομαλά.

1.2 Ιστορική αναδρομή

Ο George Charles Devol, εφεύρε το 1954 το πρώτο βιομηχανικό ρομπότ, το οποίο ονομάστηκε Unimate.



Εικόνα 1.1: Το Unimate

Το Unimate εργάστηκε σε μια γραμμή συναρμολόγησης της General Motors στο εργοστάσιο οδηγών Inland Fisher στην πόλη Ewing Township, New Jersey, το 1961. Το πρωτότυπο μοντέλο Unimate αποτελείται από ένα μεγάλο κουτί που μοιάζει με υπολογιστή, ενωμένο με ένα άλλο κιβώτιο και συνδεδεμένο με έναν βραχίονα, ο οποίος είχε αποθηκευμένες στη μνήμη του συστηματικές εργασίες. Αυτό το ρομπότ κοστίζει 65.000 δολάρια για την παραγωγή του, αλλά πωλήθηκε για \$18.000.

Σήμερα, με την εξέλιξη της τεχνολογίας έχουμε φτάσει στο σημείο που η αγορά ενός ποιο οικονομικού χόμπι ρομποτικού βραχίονα είναι εφικτή. Για την συγκεκριμένη εργασία κάνω την χρήση του OWI-535 robotic arm edge.

Τέλος, πριν προχωρήσουμε παραπέρα, πιστεύω ότι είναι χρήσιμο να κλείσουμε την εισαγωγή αυτής της διατριβής με τους τρεις νόμους της ρομποτικής:

1. Ένα ρομπότ δεν πρέπει να βλάψει έναν άνθρωπο, ή μέσω αδράνειας να επιτρέψει έναν άνθρωπο να τραυματιστεί.
2. Ένα ρομπότ πρέπει να υπακούει στις εντολές που του έχουν δοθεί από τους ανθρώπους εκτός αν οι εντολές αυτές έρχονται σε αντίθεση με τον Πρώτο Νόμο.
3. Ένα ρομπότ πρέπει να προστατεύει την ύπαρξη του εφόσον αυτή δεν έρχεται σε αντίθεση με τον Πρώτο ή Δεύτερο Νόμο.

ΚΕΦΑΛΑΙΟ 2

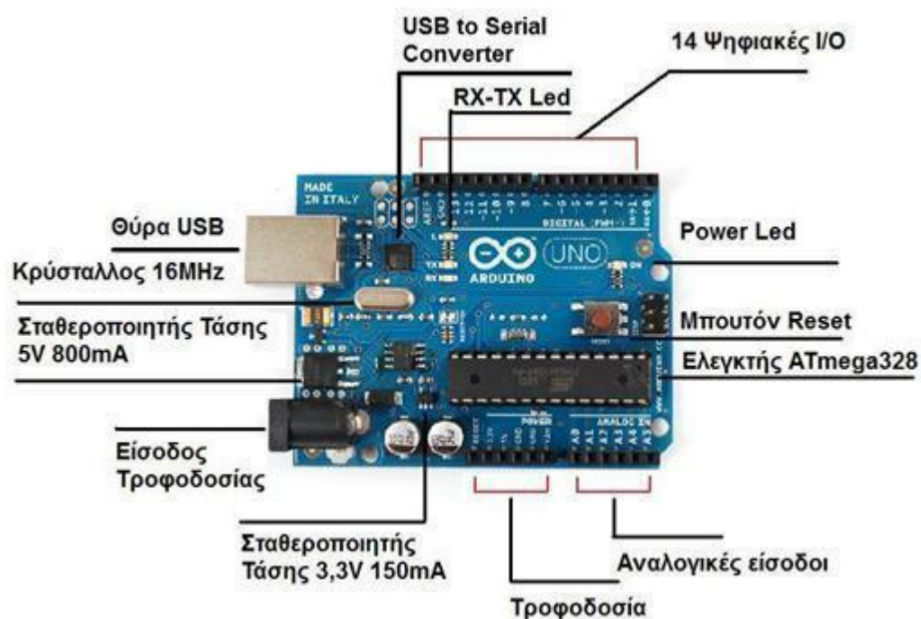
ΕΞΟΠΛΙΣΜΟΣ/ΥΛΙΚΟ/ΛΟΓΙΣΜΙΚΟ

2.1 Arduino



Εικόνα 2.1: Το Λογότυπο του Arduino

Το Arduino είναι μια online πλατφόρμα ανοικτού κώδικα βασισμένη σε συνδυασμό υλικό και λογισμικό. Το όνομα Arduino προέρχεται από ένα μπαρ στην Ivrea της Ιταλίας, όπου συναντήθηκαν μερικοί από τους ιδρυτές του έργου. Το μπαρ πήρε το όνομά του από τον Arduino της Ivrea. Τα προϊόντα του έργου διανέμονται ως υλικό και λογισμικό ανοιχτού κώδικα, τα οποία είναι εγκεκριμένα βάσει της GNU Lesser General Public License (LGPL) ή της Γενικής Δημόσιας Άδειας GNU (GPL) , επιτρέποντας την κατασκευή αρχείων για το Arduino και τη διανομή λογισμικού από οποιονδήποτε. Οι πλακέτες Arduino μπορούν να κάνουν διάφορα πράγματα όπως το να κινήσουν ένα μοτέρ ενεργοποιώντας ένα ενδεικτικό φως διαβάζοντας κάτι από το διαδίκτυο. Μπορούμε να πούμε στην πλακέτα το τι πρέπει να κάνει στέλνοντας ένα σύνολο οδηγιών στον μικροελεγκτή της πλακέτας. Για να το κάνουμε αυτό, χρησιμοποιούμε την γλώσσα προγραμματισμού Arduino (βασισμένη στο πλαίσιο για μικροελεγκτές το οποίο ονομάζεται wiring) και το λογισμικό Arduino (IDE) το οποίο είναι βασισμένο στο λογισμικό σχεδιασμού/γλώσσα προγραμματισμού processing.



Εικόνα 2.2: Το Arduino UNO

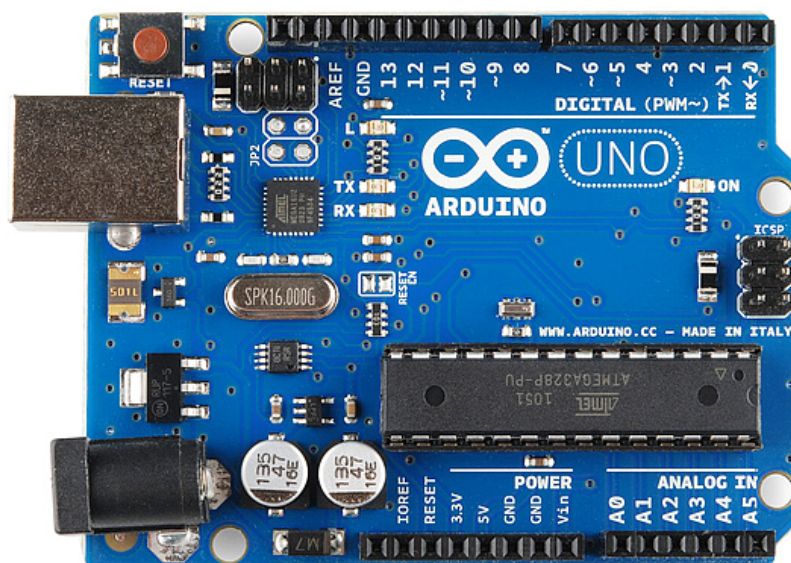
Ας μάθουμε λίγα παραπάνω στοιχεία για την πλακέτα Arduino Uno που χρησιμοποιείτε για την εκτέλεση αυτής την εργασίας.

- **Θύρα USB** - Κάθε πλακέτα Arduino χρειάζεται έναν τρόπο σύνδεσης με μια πηγή ενέργειας. Το Arduino UNO μπορεί να τροφοδοτηθεί από ένα καλώδιο USB που προέρχεται από τον υπολογιστή μας. Μέσω αυτής της θύρας η πλακέτα μπορεί να τροφοδοτηθεί με 5V, επιπλέον την χρησιμοποιούμε για να φορτώσουμε τον κώδικα στην πλακέτα.
- **Είσοδος τροφοδοσίας** - Μέσω αυτής μπορούμε να προσφέρουμε στην πλακέτα την συνιστάμενη επιπλέον τροφοδοσία μεταξύ των 6 και 12 βολτ. Η τροφοδοσία να μην ξεπερνά τα 20 βολτ.
- **Αναλογικές εισοδοί** - Η περιοχή των ακίδων κάτω από την ετικέτα 'Analog In' (A0 έως A5 στο UNO) είναι αναλογικές ακίδες. Αυτές οι ακίδες μπορούν να διαβάσουν το σήμα από έναν αναλογικό αισθητήρα (όπως έναν αισθητήρα θερμοκρασίας) και να το μετατρέψουν σε ψηφιακή τιμή που μπορούμε να διαβάσουμε.

- **Ψηφιακές I/O** - Πέρα από τις αναλογικές εισόδους/ακίδες υπάρχουν οι ψηφιακοί ακροδέκτες (0 έως 13 στον UNO). Αυτές οι καρφίτσες μπορούν να χρησιμοποιηθούν τόσο για ψηφιακή είσοδο όσο και για ψηφιακή έξοδο (όπως τροφοδοσία κινητήρα).
- **LED TX RX** - Το TX είναι συντομογραφία για τη μετάδοση, το RX είναι συντομογραφία για την λήψη. Αυτά τα σήματα αυτά εμφανίζονται πολύ συχνά στα ηλεκτρονικά για να σημάνουν τους ακροδέκτες που είναι υπεύθυνοι για τη σειριακή επικοινωνία. Στην περίπτωσή μας, υπάρχουν δύο θέσεις στο Arduino UNO όπου εμφανίζονται τα TX και RX - μία φορά με τους ψηφιακούς ακροδέκτες 0 και 1, και μια δεύτερη φορά δίπλα στις ενδεικτικές λυχνίες TX και RX. Αυτά τα LED θα μας δώσουν κάποιες ωραίες οπτικές ενδείξεις κάθε φορά που το Arduino μας λαμβάνει ή μεταδίδει δεδομένα (όπως για παράδειγμα όταν φορτώνουμε ένα καινούργιο πρόγραμμα στην πλακέτα).

Power LED - Ακριβώς κάτω και δεξιά από τη λέξη "UNO" στο σχήμα του κυκλώματος, υπάρχει μια μικρή LED δίπλα στη λέξη 'ON'. Αυτή η λυχνία LED ανάβει κάθε φορά που συνδέετε το Arduino σε πηγή ενέργειας. Αν αυτό το φως δεν ανάψει, υπάρχει μια καλή πιθανότητα κάτι να πηγαίνει στραβά.

- **Μπουτόν Reset** - Το Arduino έχει κουμπί επανεκκίνησης (reset). Με το πάτημα του θα συνδεθεί προσωρινά ο ακροδέκτης επαναφοράς στη γείωση και θα επανεκκινηθεί οποιοσδήποτε κωδικός που έχει φορτωθεί στο Arduino. Αυτό μπορεί να είναι πολύ χρήσιμο εάν ο κώδικας μας δεν επαναλαμβάνεται, αλλά θέλουμε να τον δοκιμάσουμε πολλές φορές.



Εικόνα 2.3: Το Arduino UNO σε κοντινή λήψη

- **PWM** - Δίπλα από κάποιες από τις ψηφιακές ακίδες του Arduino (ακίδες 3, 5, 6, 9, 10 και 11 στο UNO) υπάρχει ένα σύμβολο το οποίο ονομάζεται tilda (~). Αυτές οι ακίδες λειτουργούν ως κανονικές ψηφιακές ακίδες, αλλά μπορούν επίσης να χρησιμοποιηθούν για κάτι που ονομάζεται Pulse - Width Modulation (PWM). Έχουμε ένα σεμινάριο για το PWM, αλλά για τώρα, σκεφτείτε αυτές τις καρφίτσες ότι είναι σε θέση να προσομοιώνουν την αναλογική έξοδο. Έτσι, μπορούμε να συνδέσουμε λόγω χάρη έναν κινητήρα σε κάποιο από αυτά τα pin και να ελέγξουμε πλήρως την ταχύτητα του με ανάλυση 8bit (256 καταστάσεις από 0 - ακίνητο ως 255 - μέγιστη ταχύτητα) αντί να έχουμε απλά την δυνατότητα ελάχιστου - μέγιστου που παρέχουν οι υπόλοιπες ψηφιακές έξοδοι. Είναι σημαντικό να επισημάνουμε ότι το PWM δεν είναι πραγματικά αναλογικό σύστημα και ότι θέτοντας στην έξοδο την τιμή 127, δεν σημαίνει ότι η έξοδος θα δίνει 2.5V αντί της κανονικής τιμής των 5V, αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με μεγάλη συχνότητα και για ίσους χρόνους μεταξύ των τιμών 0 και 5V.

- **Ελεγκτής ATmega328** - Το μαύρο ορθογώνιο πράγμα με τα μεταλλικά πόδια είναι ένα ολοκληρωμένο ή αλλιώς ολοκληρωμένο κύκλωμα. Μπορούμε να το χαρακτηρίσουμε ως τον εγκέφαλο του Arduino μας. Το κύριο IC στο Arduino είναι ελαφρώς διαφορετικό ανάλογα με τον τύπο της πλακέτας, αλλά είναι συνήθως από τη γραμμή ATmega των IC από την εταιρεία ATMEL.
- **Σταθεροποιητές τάσεων** - Ο ρυθμιστής τάσης κάνει ακριβώς αυτό που λέει - ελέγχει την ποσότητα τάσης που επιτρέπει να χειριστεί η πλακέτα Arduino. Σκεφτείτε το ως ένα είδος προστατευτικού. Θα σβήσει μια επιπλέον τάση, αν χρειαστεί, που μπορεί να βλάψει το κύκλωμα. Φυσικά, έχει τα όριά της, οπότε δεν πρέπει να συνδέουμε το Arduino μας με τίποτα μεγαλύτερο από 20 βολτ.

2.2 Bluetooth



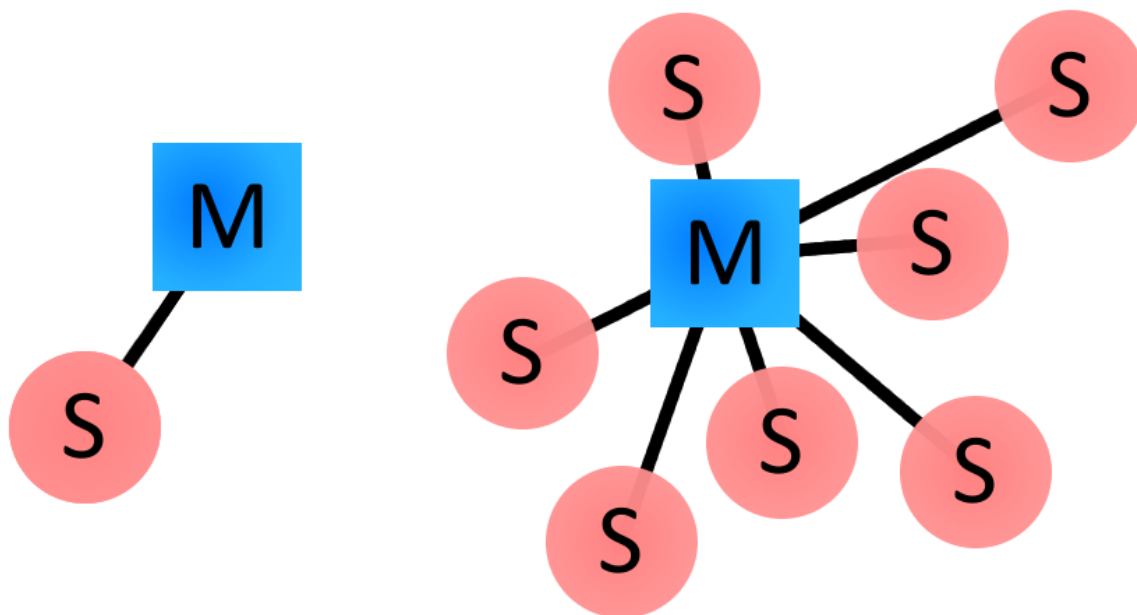
Εικόνα 2.4: Το Λογότυπο του Bluetooth

Το Bluetooth είναι ένα πρωτόκολλο με το οποίο μπορεί να γίνει αποστολή και λήψη δεδομένων μέσω ασύρματης σύνδεσης. Είναι ένα ασφαλές πρωτόκολλο και είναι ιδανικό για ασύρματες μεταδόσεις μικρής εμβέλειας, χαμηλής κατανάλωσης, χαμηλού κόστους μεταξύ ηλεκτρονικών συσκευών.

Τα δίκτυα Bluetooth χρησιμοποιούν το μοντέλο master / slave για να ελέγχουν πότε και πού μπορούν να στέλνουν δεδομένα τα προϊόντα. Σε αυτό το μοντέλο, μπορεί να συνδεθεί μία και μόνη κύρια (master) συσκευή σε επτά διαφορετικές συσκευές υποτελούς λειτουργίας. Οποιαδήποτε υποτελής (slave) συσκευή στο network μπορεί να συνδεθεί μόνο σε έναν μόνο κύριο.

2.2.1 Master / Slave τοπολογία

Ο master συντονίζει την επικοινωνία σε όλο το δίκτυο bluetooth. Μπορεί να στείλει δεδομένα σε οποιονδήποτε από τους slaves (σκλάβους) της και να ζητήσει δεδομένα από αυτά. Οι slave επιτρέπεται μόνο να μεταδίδουν και να λαμβάνουν από τον πλοίαρχο (master) τους. Δεν μπορούν να μιλήσουν με άλλους σκλάβους στο δίκτυο bluetooth.



Εικόνα 2.5: Η Master – Slave τοπολογία

2.2.2 Διαδικασία σύνδεσης μεταξύ συσκευών Bluetooth

Η δημιουργία σύνδεσης Bluetooth μεταξύ δύο συσκευών περιλαμβάνει τρεις προοδευτικές καταστάσεις:

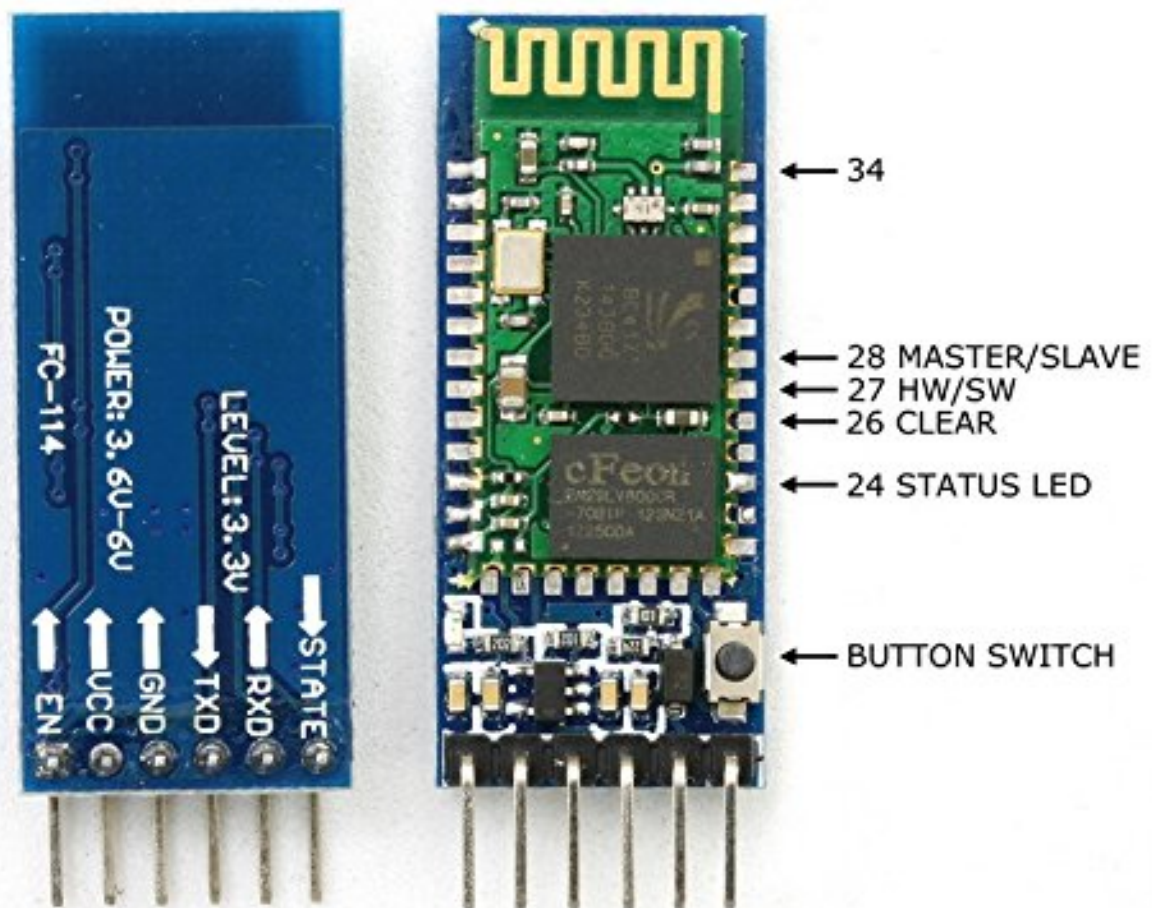
1. **Διερεύνηση (Inquiry)** – Εάν δύο συσκευές Bluetooth δεν γνωρίζουν απολύτως τίποτα ο ένας για τον άλλο, πρέπει να εκτελεστεί μια έρευνα για να προσπαθήσει η μια συσκευή να ανακαλύψει την άλλη. Μία συσκευή αποστέλλει την αίτηση έρευνας και κάθε συσκευή που ακούει για ένα τέτοιο αίτημα θα απαντήσει με τη διεύθυνσή της ενδεχομένως με το όνομά της και με άλλες πληροφορίες.
2. **Ενεργοποίηση (Connecting)** – Η ενεργοποίηση είναι η διαδικασία δημιουργίας μιας σύνδεσης μεταξύ δύο συσκευών Bluetooth. Πριν από την έναρξη αυτής της σύνδεσης, κάθε συσκευή πρέπει να γνωρίζει τη διεύθυνση του άλλου (που βρίσκεται στη διαδικασία διερεύνησης).
3. **Σύνδεση (Connection)** – Μετά την ολοκλήρωση της διαδικασίας ενεργοποίησης, η συσκευή εισέρχεται στην κατάσταση σύνδεσης. Κατά τη σύνδεση, μια συσκευή μπορεί είτε να συμμετέχει ενεργά είτε μπορεί να τεθεί σε λειτουργία χαμηλής κατανάλωσης ενέργειας.
 1. **Ενεργή λειτουργία (Active Mode)** – Αυτή είναι η κανονική συνδεδεμένη λειτουργία, όπου η συσκευή μεταδίδει ή λαμβάνει δεδομένα ενεργά.
 2. **Λειτουργία Sniff (Sniff Mode)** – Αυτή είναι μια λειτουργία εξοικονόμησης ενέργειας, όπου η συσκευή είναι λιγότερο ενεργή. Θα κοιμηθεί και θα ακούσει μόνο τις μεταδόσεις σε ένα καθορισμένο διάστημα (π.χ. κάθε 100ms).
 3. **Λειτουργία παύσης (Hold Mode)** – Η λειτουργία παύσης λειτουργίας είναι μια προσωρινή λειτουργία εξοικονόμησης ενέργειας, όπου μια συσκευή κοιμάται για μια καθορισμένη χρονική περίοδο και στη συνέχεια επιστρέφει στην ενεργό λειτουργία όταν έχει παρέλθει αυτό το διάστημα. Ο master μπορεί να εντοπίσει μια slave συσκευή.
 4. **Park Mode** – Park είναι η πιο βαθιά από τις λειτουργίες ύπνου. Ένας master μπορεί να διατάξει έναν slave να "παρκάρει", και ο

5. σκλάβος θα γίνει αδρανής μέχρι ο πλοίαρχος να του πει να ξυπνήσει.

2.2.3 Το Bluetooth HC-05

Στην εργασία αυτή θα κάνουμε χρήση του HC-05 Bluetooth κυκλώματος το οποίο είναι μια μονάδα πομποδέκτη. Παρακάτω θα επισημάνω κάποιες σημαντικές πληροφορίες για αυτό το υλικό.

HC-05 FC-114



Εικόνα 2.6: Το HC-05 Bluetooth σε κοντινή λήψη

Αυτός ο πομποδέκτης μπορεί να λειτουργήσει και ως master και ως slave. Ως master μπορεί να δεχτεί συνδέσεις και ως slave μπορεί να ξεκινήσει μια σύνδεση. Ο κάθε ακροδέκτης του έχει και διαφορετική λειτουργία.

1. **EN / KEY** – Ο EN ακροδέκτης χρησιμοποιείται για την είσοδο στην λειτουργία Command (AT). Η λειτουργία εντολών AT χρησιμοποιείται για την αλλαγή προεπιλεγμένων ρυθμίσεων της μονάδας Bluetooth. Για παράδειγμα, αν έχουμε την ανάγκη να αλλάξουμε το όνομα της συσκευής Bluetooth, τον ρόλο της συσκευής σε master ή slave, τον κωδικό πρόσβασης της πλακέτας Bluetooth πρέπει να ρυθμιστεί στη λειτουργία AT Command για να μπορούμε να αλλάξουμε την προεπιλεγμένη ρύθμιση. Για την συγκεκριμένη εργασία αυτή είναι μια ιδιότητα που δεν θα χρησιμοποιήσουμε.
2. **VCC** – Συνδέεται με την τάση η οποία μπορεί να είναι τα 5 βολτ του Arduino.
3. **GND** – Συνδέεται με την κοινή γείωση.
4. **TXD** – Συνδέεται με τον ακροδέκτη RX του Arduino.
5. **RXD** – Συνδέεται με τον ακροδέκτη TX του Arduino.
6. **STATE** – Ο ακροδέκτης STATE στην πλακέτα HC-05 συνδέεται με τον ακροδέκτη LED της μονάδας Bluetooth και ο ακροδέκτης LED χρησιμοποιείται για να υποδείξει πότε υπάρχει ενεργή σύνδεση. Αυτό σημαίνει ότι το Arduino μπορεί να συνδεθεί στον ακροδέκτη STATE και να καθορίσει πότε έχουμε μια σύνδεση. Μπορούμε επίσης να χρησιμοποιήσουμε το Arduino για να διαβάσουμε την τιμή του pin STATE. Ο ακροδέκτης STATE είναι ανενεργός όταν η πλακέτα HC-05 είναι ασύνδετη, ενώ όταν είναι συνδεδεμένη είναι ενεργός. Στην συγκεκριμένη εργασία δεν θα το χρησιμοποιήσουμε.

Η ενσωματωμένη λυχνία LED του HC-05 δείχνει την τρέχουσα κατάσταση της μονάδας. Αν αναβοσβήνει γρήγορα (περίπου 5 φορές το δευτερόλεπτο) ο πομποδέκτης είναι ενεργοποιημένος και αναμένει σύνδεση ή αντιστοιχίση (αυτές τις έννοιες θα τις δούμε παρακάτω). Αν ο πομποδέκτης κάνει ένα γρήγορο φλας μία φορά κάθε 2 δευτερόλεπτα, η μονάδα έχει μόλις αντιστοιχιστεί με άλλη συσκευή. Αν κάνει ένα διπλό γρήγορο φλας κάθε 2 δευτερόλεπτα σημαίνει ότι είναι συνδεδεμένο με άλλη συσκευή.

Το HC-05 θυμάται τις συσκευές με τις οποίες έχει αντιστοιχιστεί / ζευγαρώσει στο παρελθόν και μετά από μια επανασύνδεση δεν χρειάζεται να τα επανασυνδέσουμε. Αυτό σημαίνει ότι μπορούμε να πάμε από το να αναβοσβήνει ταχύτατα, η ενσωματωμένη λυχνία (περίπου 5 φορές το δευτερόλεπτο), που σημαίνει ότι η πλακέτα είναι ενεργοποιημένη και αναμένει για σύνδεση ή αντιστοίχιση, σε γρήγορο διπλό φλας ανά 2 δευτερόλεπτα που σημαίνει ότι είναι συνδεδεμένο απευθείας με άλλη συσκευή όπως εξηγήσαμε και προηγουμένως.

2.2.4 Συγκόλληση και σύζευξη

Όταν δύο συσκευές Bluetooth μοιράζονται μια συγγενική σχέση μπορούν να δεσμευτούν μεταξύ τους. Οι δεσμευμένες συσκευές δημιουργούν αυτόματα μια σύνδεση όποτε είναι αρκετά κοντά. Όταν ξεκινώ το αυτοκίνητό μου, για παράδειγμα, το τηλέφωνο στην τσέπη μου συνδέεται άμεσα με το σύστημα Bluetooth του αυτοκινήτου, επειδή μοιράζεται ένα δεσμό.

Η δέσμευση αυτή δημιουργείτε μέσω μιας διαδικασίας που ονομάζεται αντιστοίχιση (pairing). Όταν οι συσκευές ζευγαρώνονται, μοιράζονται τις διευθύνσεις, τα ονόματα και τα προφίλ τους και συνήθως τις αποθηκεύουν στη μνήμη. Επίσης, μοιράζονται ένα κοινό μυστικό κλειδί, το οποίο τους επιτρέπει να δεσμεύονται όποτε είναι μαζί στο μέλλον.

Η σύζευξη απαιτεί συνήθως μια διαδικασία ελέγχου ταυτότητας, όπου ένας χρήστης πρέπει να επικυρώσει τη σύνδεση μεταξύ συσκευών. Η ροή της διαδικασίας ελέγχου ταυτότητας ποικίλλει και συνήθως εξαρτάται από τις δυνατότητες διασύνδεσης της μιας ή της άλλης συσκευής. Μερικές φορές η σύζευξη είναι μια απλή λειτουργία, όπου το πάτημα ενός κουμπιού είναι το μόνο που χρειάζεται για το ζευγάρωμα. Άλλες φορές η αντιστοίχιση περιλαμβάνει αντιστοίχιση 6-ψηφίων αριθμητικών κωδικών ή ενός κοινού κωδικού PIN σε κάθε συσκευή. Ο κωδικός PIN μπορεί να κυμαίνεται σε μήκος και πολυπλοκότητα από τέσσερις αριθμούς (π.χ. "0000" ή "1234") σε μια αλφαριθμητική συμβολοσειρά 16 χαρακτήρων.

2.3 OWI robotic arm edge

Το OWI robotic arm edge είναι ένα τηλεχειριζόμενο κιτ ρομποτικού βραχίονα κατασκευασμένος από την OWI Incorporated με σκοπό να εισαγάγουν τους σπουδαστές στη ρομποτική τεχνολογία με έναν οικονομικό τρόπο. Μέσω αυτού μπορούν οι χρήστες του να αποκτήσουν γνώσεις μηχανολογίας, ηλεκτρολογίας, και της επιστήμης των υπολογιστών. Αυτού του είδους κιτ είναι δημοφιλή στον κλάδο της Επιστήμης, Τεχνολογίας, Μηχανικής, και Μαθηματικών (STEM).

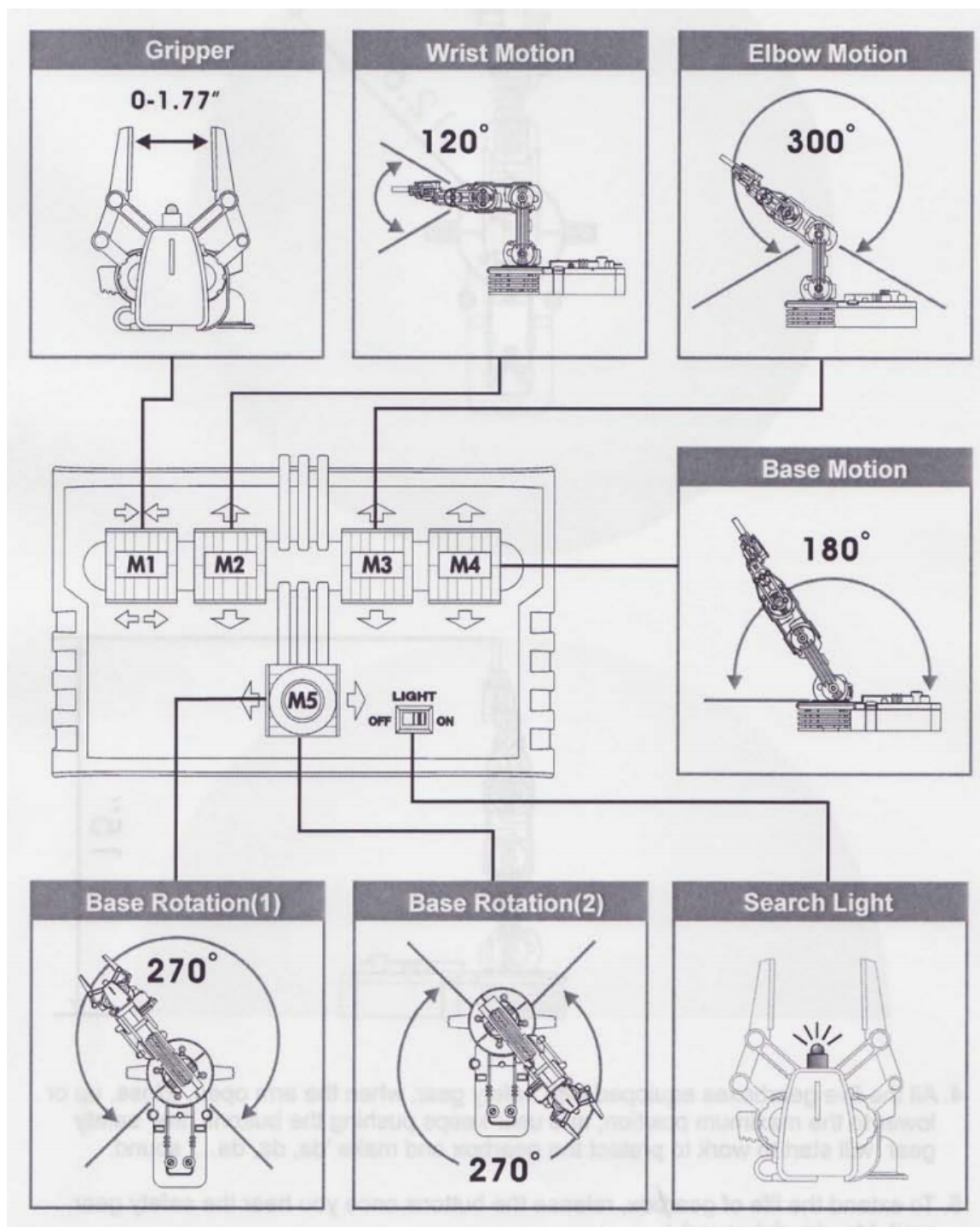


Εικόνα 2.7: Το OWI robotic arm edge κιτ

Ο βραχίονας αυτός έχει 5 βαθμούς ελευθερίας, δηλαδή μπορεί να σηκώσει ένα αντικείμενο, να το μετακινήσει οριζόντια και να το θέσει σε λειτουργία, σε χώρο X – Y – Z, ενώ αλλάζει τον προσανατολισμού του αντικειμένου κατά μήκος δύο αξόνων (στροφή και κλίση για παράδειγμα). Χρησιμοποιεί πέντε κινητήρες με κιβώτια ταχυτήτων, και έχει κίνηση καρπού 120 μοίρες, κίνηση αγκώνα 300 μοίρες, κίνηση ώμου 180 μοίρες, κίνηση βάσης 270 μοίρες και κίνηση

Κίνηση Ρομπωτικού Βραχίονα με Κινητή Συσκευή Android και Bluetooth

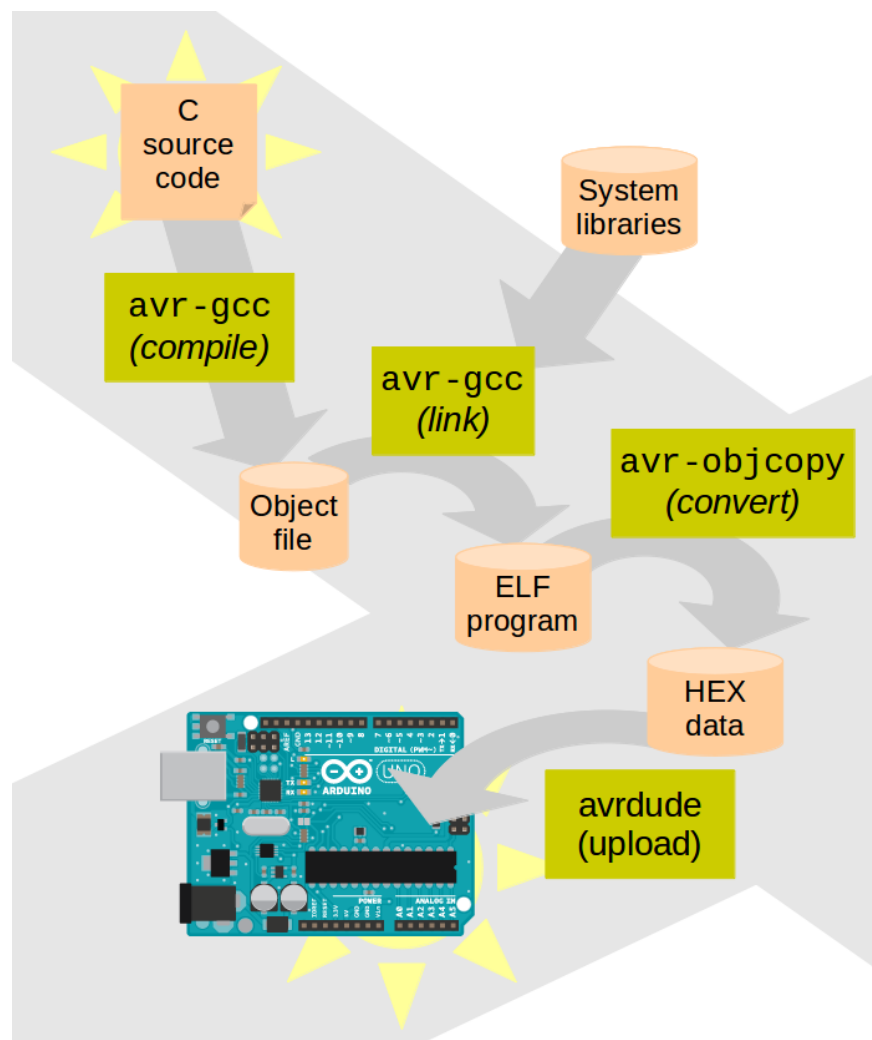
πιασίματος 0-1 (δηλαδή άνοιγμα 0-4,5 εκατοστά). Επιπλέον έχει ανυψωτική ικανότητα 100 γραμμαρίων και ένα φως LED.



Εικόνα 2.8: Οι 12 διαφορετικές κινήσεις του βραχίονα

2.4 Προγραμματισμός για Arduino

Για να προγραμματίσουμε την πλακέτα Arduino μας πρέπει να περάσει ο κώδικας μας από διάφορα στάδια. Πρώτον, το περιβάλλον Arduino εκτελεί μερικούς μικρούς μετασχηματισμούς για να βεβαιωθεί ότι ο κώδικας είναι σωστός C ή C ++. Η γλώσσα Arduino είναι απλώς ένα σύνολο αλγορίθμων C / C ++ που μπορούν να καλούνται από τον κωδικό μας. Ακολούθως διαβιβάζεται ο κώδικας αυτός σε έναν μεταγλωττιστή (avr-gcc), ο οποίος μετατρέπει τον κώδικα ο οποίος είναι αναγνώσιμος από τον άνθρωπο σε οδηγίες αναγνώσιμες από μια μηχανή (ή αρχεία αντικειμένων).



Εικόνα 2.9: Διαδικασία εισαγωγής κώδικα σε πλακέτα Arduino

Τα αρχεία αντικειμένων είναι αρχεία κώδικα που θα μπορούσαν να τρέξουν τα τσιπ AVR. Η AVR είναι μια οικογένεια μικροελεγκτών που αναπτύχθηκε από την Atmel το 1996. Η AVR ήταν μια από τις πρώτες οικογένειες μικροελεγκτών που χρησιμοποιούσαν μνήμη flash on-chip για αποθήκευση προγραμμάτων, σε αντίθεση με ένα προγραμματισμένο ROM, EPROM ή EEPROM Άλλους μικροελεγκτές την εποχή εκείνη. Ο σύνδεσμος AVR-ld θα λάβει όλα αυτά τα αρχεία συναρμολόγησης και τα ονόματα λειτουργιών διασταύρωσης για τη δημιουργία ενός μόνο αρχείου αντικειμένου. Ο σύνδεσμος θα λάβει επίσης ενότητες από τη βιβλιοθήκη «C» και θα τα καταστήσει σε ένα μόνο αντικείμενο. Κανονικά αυτό το συνδεδεμένο αντικείμενο είναι σε μορφή ELF και επιπλέον το AVR-objcopy χρησιμοποιείται για τη δημιουργία ενός αρχείου μορφής HEX.

Στη συνέχεια, ο κώδικας μας συνδυάζεται με τις τυπικές βιβλιοθήκες Arduino που παρέχουν βασικές λειτουργίες όπως το digitalWrite () ή το Serial.print (). Το αποτέλεσμα είναι ένα ενιαίο αρχείο hex, το οποίο περιέχει τα συγκεκριμένα byte που πρέπει να εγγραφούν στη μνήμη προγραμμάτων του chip στην πλακέτα του Arduino. Αυτό το αρχείο μεταφορτώνεται στην πλακέτα Arduino: μεταδίδεται μέσω USB ή σειριακής σύνδεσης μέσω του bootloader στο τσιπ ή με το εξωτερικό υλικό προγραμματισμού.

Όλες οι τυπικές κατασκευές C και C ++ που υποστηρίζονται από το avr-g ++ θα πρέπει να δουλεύουν στο Arduino. Το AVR-GCC είναι ένας μεταγλωττιστής (compiler) που λαμβάνει κώδικα υψηλού επιπέδου γλώσσας C και δημιουργεί μια δυαδική πηγή που μπορεί να μεταφορτωθεί σε μικροελεγκτή AVR. Επομένως, το AVR-GCC μπορεί να θεωρηθεί ως ένα «C» cross compiler για την παραγωγή κώδικα AVR. Τα αρχεία AVR-libc είναι βιβλιοθήκες χρόνου εκτέλεσης «C», αρχεία κεφαλίδων και τεκμηρίωση κυρίως για τον στόχο AVR και χρησιμοποιούνται σε συνδυασμό με το AVR-GCC. Σημειώστε ότι τα αρχεία AVR-libc και AVRLIB είναι διαφορετικά σύνολα βιβλιοθηκών αλλά και τα δύο λειτουργούν με τον μεταγλωττιστή AVR-GCC .

2.5 Android



Εικόνα 2.10: Το Λογότυπο του Android

Η εταιρεία Android Inc. ιδρύθηκε στο Palo Alto της Καλιφόρνια τον Οκτώβριο του 2003 από τους Andy Rubin, Rich Miner, Nick Sears και Chris White. Τον Ιούλιο του 2005, η Google εξαγόρασε την Android Inc. για τουλάχιστον 50 εκατομμύρια δολάρια. Οι βασικοί υπάλληλοι της, όπως οι Rubin, Miner και White, εντάχθηκαν στη Google ως υπάλληλοι ως μέρος της εξαγοράς.

Οι εφαρμογές Android προγραμματίζονται σε γλώσσα Java. Αφού προγραμματίσουμε μια εφαρμογή, τα εργαλεία Android SDK συμβάλλουν στη συγκέντρωση των δεδομένων και των πόρων όπως τα αρχεία XML, τα αρχεία jar, εικόνες και αντικείμενα σε ένα ενιαίο πακέτο αρχειοθέτησης με επέκταση ".apk". Αυτό το αρχείο APK μπορεί να χρησιμοποιηθεί για την εγκατάσταση της εφαρμογής με ένα απλό κλικ στις συσκευές Android. Τώρα, δεδομένου ότι το λειτουργικό σύστημα Android συμπεριφέρεται σαν περιβάλλον Linux, η συμπεριφορά της εφαρμογής είναι παρόμοια. Κάθε εφαρμογή θεωρείται ως μια ξεχωριστή οντότητα και εκτελείται με τη δική του εικονική μηχανή. Με αυτό τον τρόπο απομονώνεται ο προγραμματισμένος κώδικας της εφαρμογής και επίσης εμποδίζεται η επιπλοκή με άλλες εφαρμογές εκτός αν αυτό ζητείτε να γίνει ρητά. Κάθε άλλη εφαρμογή έχει το δικό της κωδικό αναγνώρισης (user ID) και κάθε άλλη διαδικασία έχει τη δική της εικονική μηχανή.

(Κενό φύλλο)

ΚΕΦΑΛΑΙΟ 3

ΕΡΓΑΣΙΑ



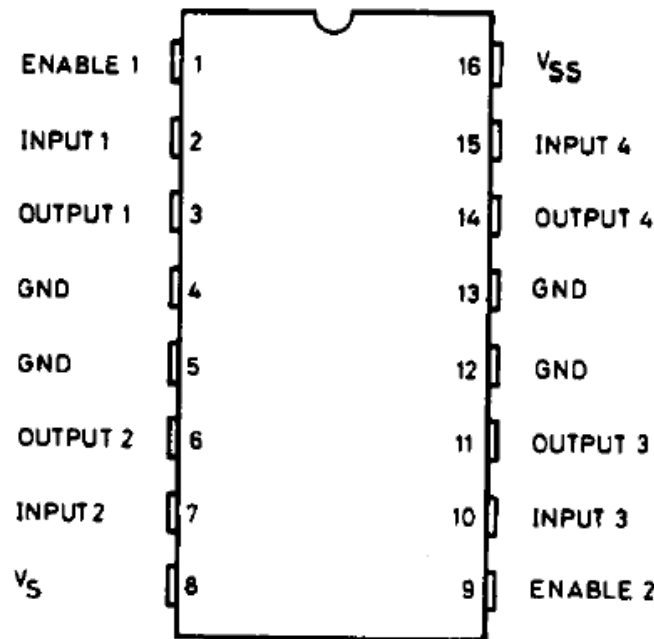
Εικόνα 3.1: Μερικά από τα υλικά κατασκευής της εργασίας

3.1 Το κύκλωμα

Ένα πολύ σημαντικό κομμάτι αυτής της εργασίας είναι το κύκλωμα γιατί αυτό είναι ένα από τα βασικά κομμάτια της. Βέβαια όλα τα υλικά και λογισμικά κομμάτια της εργασίας παίζουν ένα σημαντικό ρόλο γιατί όλα αυτά συνδέονται σαν ένας κρίκος. Αν δεν λειτουργεί κάποιο κομμάτι από αυτά, δεν λειτουργούν ούτε τα άλλα. Ας αναλύσουμε όμως τα πιο σημαντικά υλικά κομμάτια του κυκλώματος της εργασίας για να καταλάβουμε γιατί χρησιμοποιήσα τα συγκεκριμένα υλικά.

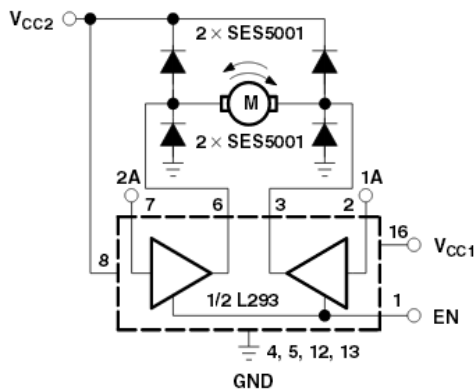
Ο ρομποτικός βραχίονας OWI έχει 5 κινητήρες, για τον λόγο αυτό για να τον κινήσουμε χρειαζόμαστε ειδικά ολοκληρωμένα. Επέλεξα να χρησιμοποιήσω το L2930NE. Το L293 είναι τσιπ τύπου H-bridge, δηλαδή έχει σχεδιαστεί για να παρέχει ρεύματα κίνησης διπλής κατεύθυνσης έως και 1 A σε τάσεις από 4,5 V έως 36 V. Το Arduino μπορεί στη συνέχεια να ελέγξει ψηφιακά το τσιπ H-bridge για να ενεργοποιήσει / απενεργοποιήσει τον κινητήρα και να αλλάξει την

κατεύθυνση του. Το κάθε τσιπ μπορεί να κινήσει 2 κινητήρες. Μιας και έχουμε 5 κινητήρες χρησιμοποιήσα 3 τσιπ L2930NE.



Εικόνα 3.2: Το ολοκληρωμένο L2930NE.

Οι τέσσερις έξοδοι (output) του τσιπ δίνουν την τάση που χρειαζόμαστε για να κινήσουμε το κύκλωμα.



EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

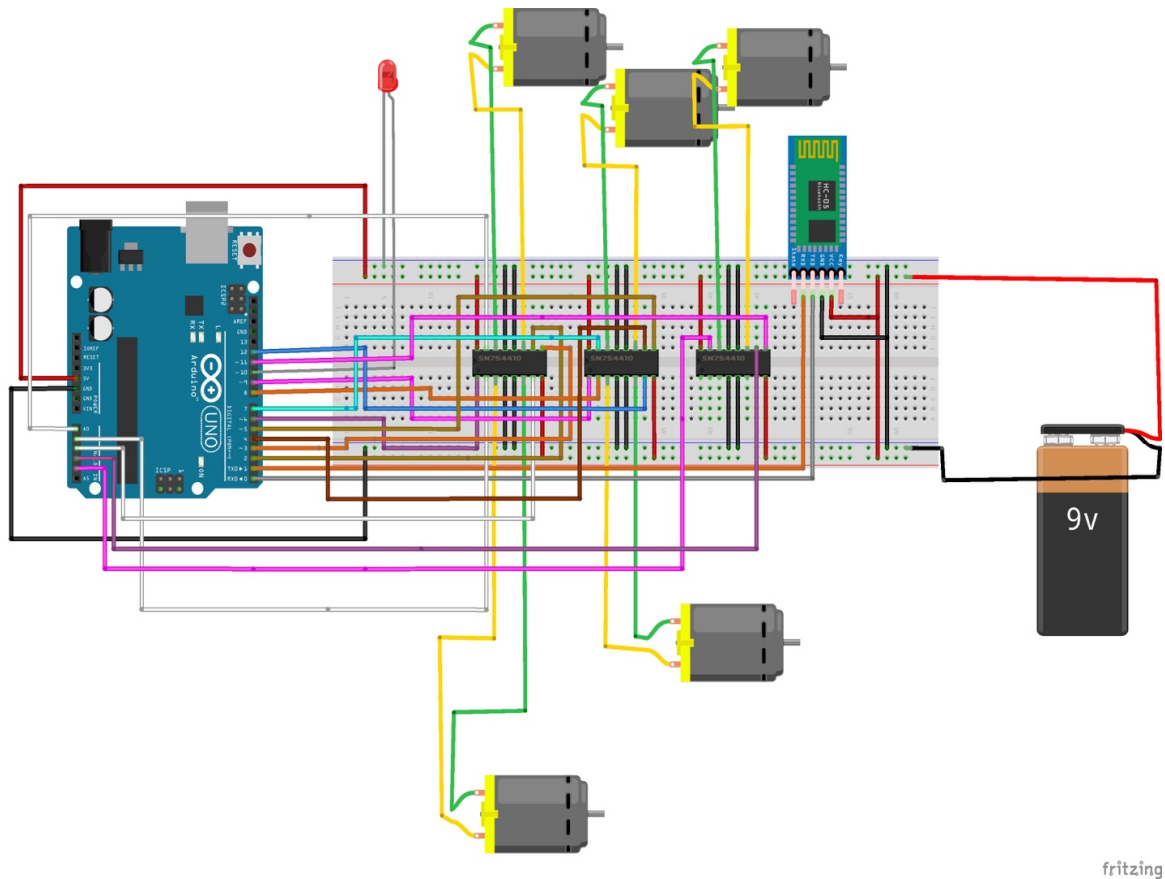
Figure 5. Bidirectional DC Motor Control

Εικόνα 3.3: Το εσωτερικό κύκλωμα του L2930NE.

Μέσω του πίνακα στο δεξί μέρος μπορούμε να δούμε την κατεύθυνση που θα κινηθεί ο κινητήρας σύμφωνα με την τάση που θα δώσουμε στα πόδια εισόδου (input) και enable.

Το τελικό κύκλωμα είναι αρκετά περίπλοκο για αυτό τον λόγο πρέπει να προσέξουμε τον τρόπο που συνδέουμε το κάθε καλώδιο.

Στο κύκλωμα αυτό χρησιμοποιώ μια μπαταρία των 6 βολτ των 1.3 αμπέρ και όχι των 9 βολτ όπως στην εικόνα που θα επισυνάψω παρακάτω. Με αυτή την μπαταρία ως πηγή ενέργειας και την 5 βολτ ακίδα του Arduino μπορώ να δώσω αρκετή τάση στο κύκλωμα μας για να κινήσουμε όλους τους κινητήρες.



Εικόνα 3.4: Το κύκλωμα σχεδιασμένο στο Fritzing.

Το LED πάνω στο κύκλωμα αντιπροσωπεύει το φως το οποίο βρίσκετε πάνω στον ρομποτικό βραχίονα. Παρόλο που το κύκλωμα φαίνεται αρκετά δύσκολο αρχικά καθώς το έφτιαχνα κατάλαβα ότι το τι κάνει το κάθε τσιπ L2930NE είναι παρόμοιο. Για αυτό τον λόγο έχουν και παρόμοια συνδεσμολογία. Ως είσοδο για την ακίδα Enable του τσιπ L2930NE έκανα ακίδες PWM του Arduino γιατί ήθελα να έχω μεγαλύτερη ελευθερία στο να ελέγξω την ταχύτητα των κινητήρων.

3.2 Ο κώδικας Arduino

```
1. // For outputs of left side of 1nd IC (base movement)
2.
3. int E11 = 6; // Enable Pin for motor 1
4. int J1 = A1; // Control pin 1 for motor 1
5. int J2 = A2; // Control pin 2 for motor 1
6.
7. // For outputs of right side of 1nd IC (shoulder movement)
8.
9. int E12 = 9; // Enable Pin for motor 1
10. int J11 = 2; // Control pin 1 for motor 1
11. int J21 = A0; // Control pin 2 for motor 1
12.
13. //=====
14. // For outputs of left side of 2nd IC (elbow movement)
15.
16. int E21 = 9; // Enable Pin for motor 1
17. int I1 = 8; // Control pin 1 for motor 1
18. int I2 = 10; // Control pin 2 for motor 1
19.
20. // For outputs of right side of 2nd IC (wrist movement)
21.
22. int E22 = 5; // Enable Pin for motor 1
23. int I11 = 4; // Control pin 1 for motor 1
24. int I21 = 7; // Control pin 2 for motor 1
25.
26. //=====
27. // For outputs of left side of 3rd IC (Gripper movement)
28.
29. int E3 = 11; // Enable Pin for motor 1
30. int K1 = A3; // Control pin 1 for motor 1
31. int K2 = A4; // Control pin 2 for motor 1
```


Στον παραπάνω κώδικα κάνουμε αρχικοποίηση κάποιων μεταβλητών για το κάθε μοτέρ του βραχίονα. Στη συνέχεια κάνουμε αρχικοποίηση μιας μεταβλητής data η οποία είναι η μεταβλητή στην οποία αργότερα θα της αναθέσω την τιμή της πληροφορίας που θα της στείλει η εφαρμογή Android μου.

Η τιμή της μεταβλητής PWM χρησιμεύει στο να ρυθμίσω την ταχύτητα των μοτέρ.

- **char** data = 0; // Variable for storing received data
- **int** pwm = 200; // Value of pwm

Παρακάτω μπορούμε να δούμε το δείγμα του κώδικα πάνω στο οποίο ρυθμίζω τις ακίδες του Arduino ως εξόδους. Αυτό το κάνω για να στέλνουν σήμα στα μοτέρ στο πότε θα κινηθούν μόλις λάβουν την εντολή από τον πομποδέκτη.

```
1. void setup()
2. {
3.   Serial.begin(9600); //Sets the data rate in bits per second (baud) for
   serial data transmission
4.
5.   pinMode(E11, OUTPUT);
6.   pinMode(J1, OUTPUT);
7.   pinMode(J2, OUTPUT);
```

Στην συνέχεια σε συνεχείς επανάληψη πρώτον διαβάζω το data που λαμβάνω σειριακά μια μέσω της τιμής που θα έχει αυτό, κινώ το κατάλληλο μοτέρ.

```
1. void loop()
2. {
3.   if(Serial.available() > 0) // Send data only when you receive data:
4.   {
5.     data = Serial.read(); //Read the incoming data and store it into variable data
6.     Serial.print(data); //Print Value inside data in Serial monitor
7.     Serial.print("\n"); //New line
```

Έχω δώσει την εντολή να κινείται το μοτέρ του βραχίονα όταν πάρει την τιμή 2.

```
1.  if(data == '2'){ // WRIST motion
2.    digitalWrite(I11, HIGH); // Run forward
3.    digitalWrite(I21, LOW);
4.    analogWrite(E22, pwm); // Run in half speed
5.  }
```

Βάζω την τιμή High (5V) στην μια είσοδο του Arduino, την τιμή 200 στο pwm, και την τιμή LOW (0V) σε μια δεύτερη είσοδο του ίδιου τσιπ. Έτσι το μοτέρ αυτό γυρίζει δεξιόστροφα.

```
1.  else if(data == '4'){
2.    digitalWrite(I11, LOW); // Run backwards
3.    digitalWrite(I21, HIGH);
4.    analogWrite(E22, pwm); // Run in half speed
5.  }
```

Βάζω την τιμή LOW (0V) στην μια είσοδο του Arduino, την τιμή 200 στο pwm, και την τιμή High (5V) σε μια δεύτερη είσοδο του ίδιου τσιπ. Έτσι το μοτέρ αυτό γυρίζει αριστερόστροφα (δηλαδή το ανάποδο από το τι έκανα προηγουμένως)..

```
6.  else if(data == '5'){
7.    digitalWrite(I11, LOW); // Stop motor
8.    digitalWrite(I21, LOW);
9.  }
```

Βάζοντας την τιμή LOW (0V) σε δύο εισόδους ταυτόχρονα, επιτυγχάνουμε το να μένει ακίνητος ο βραχίονας.

Μετά από κάθε κίνηση βάζουμε μια καθυστέρηση ως ώστε να αποφύγουμε το να γίνουν πολλαπλές απότομες κινήσεις ταυτόχρονα.

```
1. delay(25);
```

3.3 Ο κώδικας Android

```
> //We check if the device has bluetooth and if it is enabled.  
> myBluetooth = BluetoothAdapter.getDefaultAdapter();  
>  
> if(myBluetooth == null)  
> {  
>     //Display a message that the device has no bluetooth adapter.  
>     Toast.makeText(getApplicationContext(), "Bluetooth is not availab  
le on this device.", Toast.LENGTH_LONG).show();  
>  
>     //finish apk  
>     finish();  
> }
```

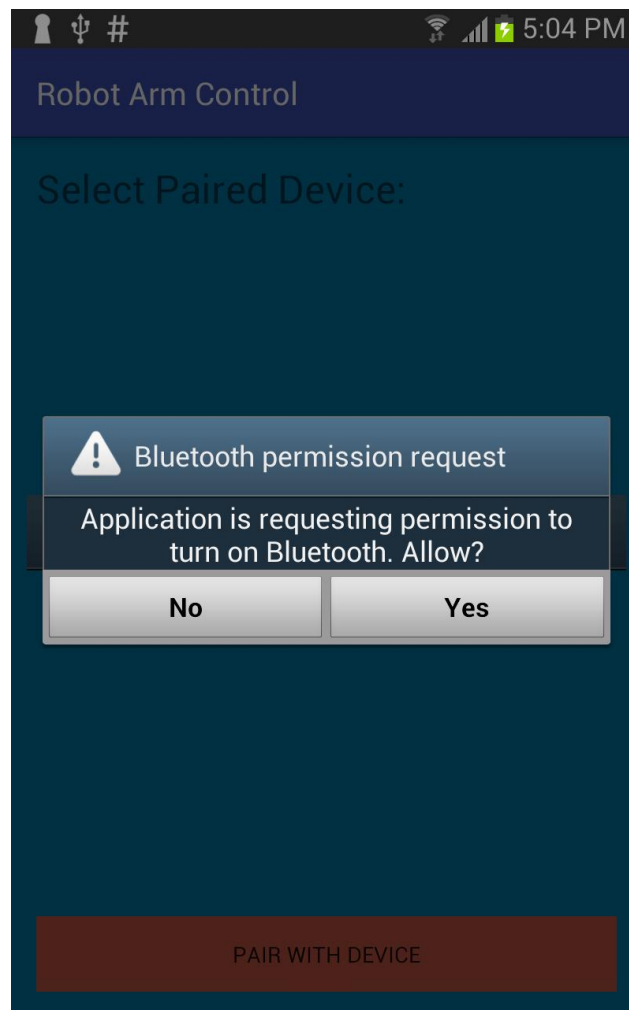
Παραπάνω εξετάζουμε αν το κινητό μας έχει πομποδέκτης και να είναι ενεργοποιημένος μέσω της συνάρτησης `getDefaultAdapter()` που επιστρέφει μια αφηρημένη αναφορά (handle) ως πόρο στην τοπική μεταβλητή του πομποδέκτη μας (Bluetooth Adapter). Αν η μεταβλητή μας `myBluetooth` έχει μηδενικό περιεχόμενο τότε αυτό σημαίνει ότι η συσκευή αυτή δεν έχει πομποδέκτη. Έτσι επιστρέφω ένα μήνυμα που λέει ότι δείχνουμε ένα μήνυμα που λέει ότι ο πομποδέκτης δεν είναι διαθέσιμος σε αυτήν την συσκευή. Αν όμως δεν έχει μηδενικό περιεχόμενο τότε αυτό σημαίνει ότι η κινητή συσκευή μας έχει πομποδέκτη. Αυτό που μένει να κάνω στην περίπτωση που το κινητό μου έχει πομποδέκτη είναι να κοιτάξω αν είναι ενεργοποιημένο, γιατί αν δεν είναι ενεργοποιημένο δεν μπορώ να λάβω ή να στείλω δεδομένα.

```
> else if(!myBluetooth.isEnabled())
```

```
//If bluetooth exists on the device but is disabled request user to enable it.
```

```
> Intent turnBTon = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);  
> startActivityForResult(turnBTon,1);  
> }
```

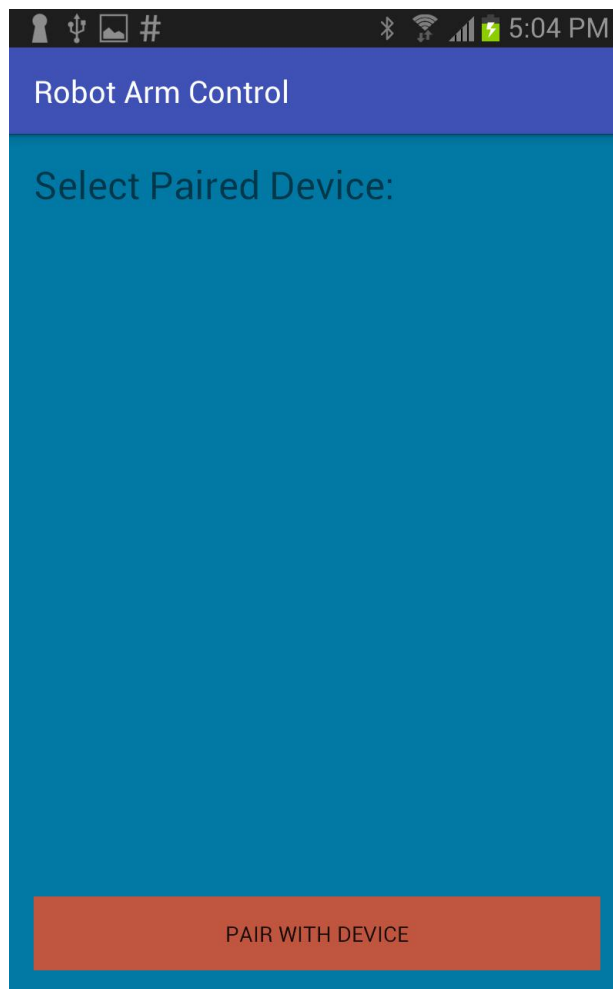
Αν ο πομποδέκτης μου δεν είναι ενεργοποιημένος, μέσω της εντολής ACTION_REQUEST_ENABLE εμφανίζεται μια δραστηριότητα του συστήματος (system activity) που επιτρέπει τον χρήστη να ενεργοποιήσει τον πομποδέκτη.



Εικόνα 3.5: Ενεργοποίηση πομποδέκτη κινητού

Στην προηγούμενη εικόνα βλέπουμε ότι η εφαρμογή μου ενεργοποιεί την δραστηριότητα του συστήματος που ζητάει αν θέλω να επιτρέψω στο σύστημα να

ενεργοποιήσει τον πομποδέκτη του κινητού μου. Αυτό γίνεται γιατί είχα τον πομποδέκτη του κινητού μου απενεργοποιημένο. Μόλις πατήσω το ναι μου επιτρέπει να μπω στην εφαρμογή.



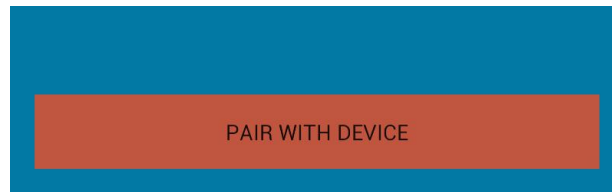
Εικόνα 3.6: Αρχική σελίδα εφαρμογής

```
➤ btnPair.setOnClickListener(new View.OnClickListener() {  
➤     @Override  
➤     public void onClick(View v)  
➤     {  
➤         pairedDevicesList();  
➤     }  
➤ });
```

Στο κουμπί PAIR WITH DEVICE έχω βάλει έναν ανιχνευτή πατήματος του κουμπιού. Μόλις πατηθεί το κουμπί αυτό καλείτε η συνάρτηση pairedDevicesList().

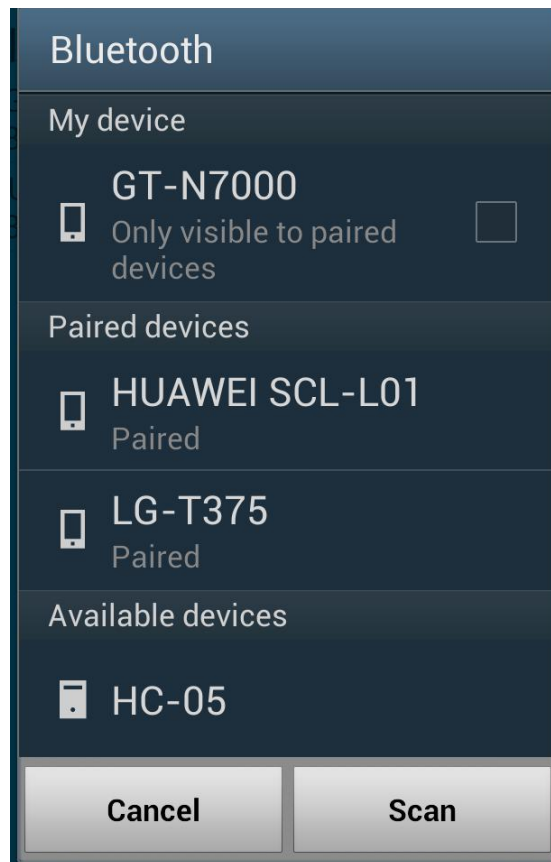
```
1. private void pairedDevicesList()
2. {
3.     pairedDevices = myBluetooth.getBondedDevices();
4.     ArrayList list = new ArrayList();
5.
6.     if (pairedDevices.size()>0)
7.     {
8.         for(BluetoothDevice bt : pairedDevices)
9.         {
10.            list.add(bt.getName() + "\n" + bt.getAddress()); //Get the device's name and the address.
11.        }
12.    }
13.    else
14.    {
15.        Toast.makeText(getApplicationContext(), "No paired bluetooth modules found. Pair and try again. ", Toast.LENGTH_LONG).show();
16.    }
```

Η συνάρτηση pairedDevicesList() κοιτάει αν υπάρχουν συσκευές που έχουν αντιστοιχιστεί με την κινητή εφαρμογή μου και αν υπάρχει τις δείχνει σε μια λίστα, αν δεν υπάρχει μου επιστρέφει ένα μήνυμα που λέει ότι δεν υπάρχουν αντιστοιχισμένοι πομποδέκτες για αυτό πρέπει να κάνω την αντιστοίχιση και να ξαναδοκιμάσω.



Εικόνα 3.7: Πάτημα αντιστοίχισης

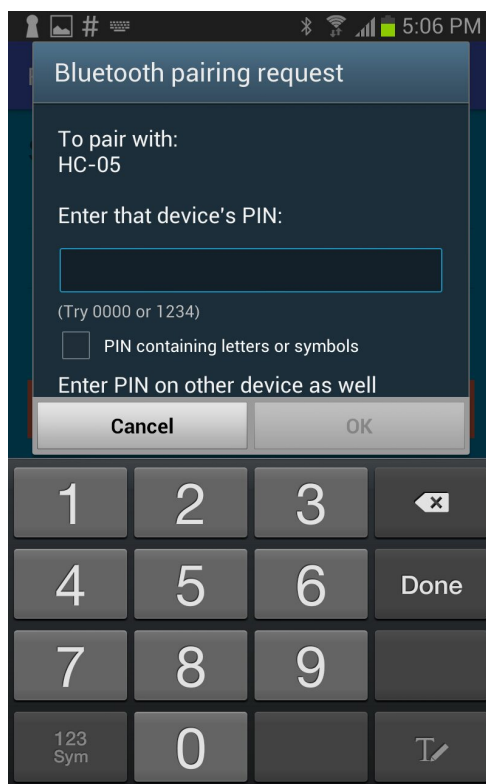
Με το πάτημα του κουμπιού αντιστοίχισης εμφανίστηκαν κάποιες συσκευές Bluetooth με τις οποίες έχω αντιστοιχήσει τον πομποδέκτη του κινητού μου αλλά δεν μπορώ να δω τον πομποδέκτη HC-05 του κυκλώματος μου. Για να λύσω αυτό το πρόβλημα πρέπει να κάνω αντιστοίχιση με το HC-05. Πάω στις ρυθμίσεις του Bluetooth του κινητού μου για να το κάνω αυτό.



Εικόνα 3.8: Ρυθμίσεις Bluetooth

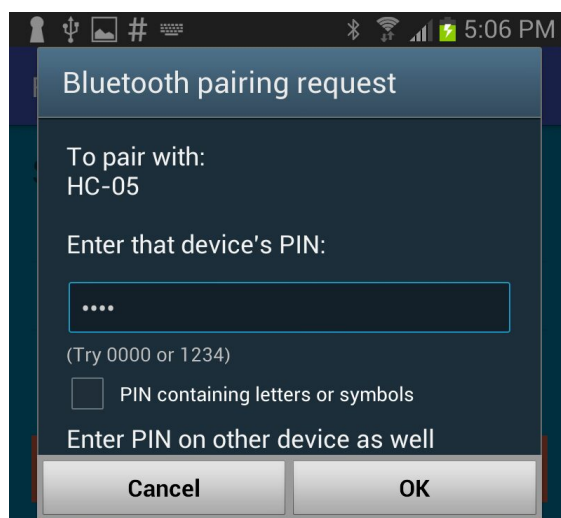
Μέσω αυτής της εικόνας επιβεβαιώνουμε ότι πραγματικά δεν έχει αντιστοιχιστεί το Bluetooth μας με το HC-05.

Επιλέγω το HC-05.



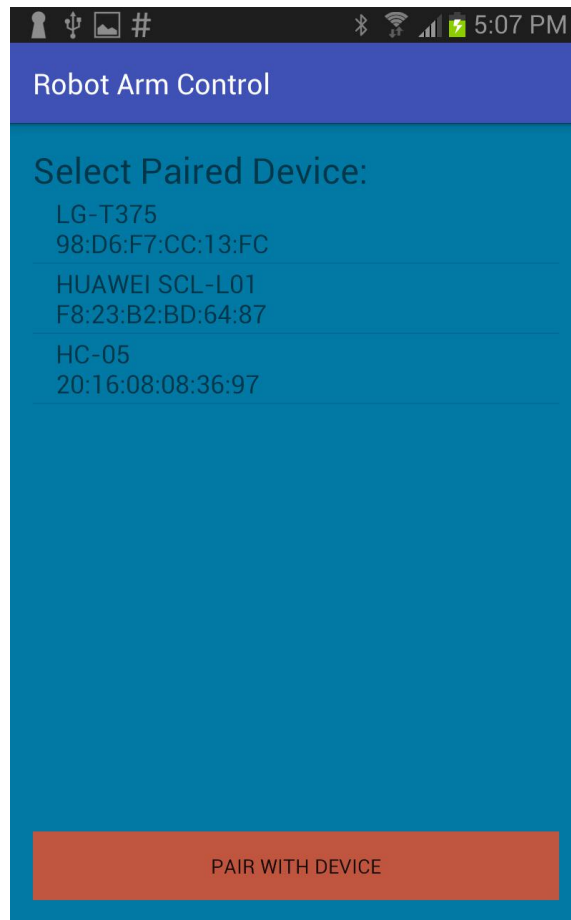
Εικόνα 3.9: Ζήτηση κωδικού για αντιστοίχιση

Μόλις το κάνω αυτό μου ζητάει τον κωδικό του πομποδέκτη HC-05. Ο κωδικός είναι συνήθως το 1234 ή το 0000, για αυτό και μας τα προτείνει. Ο δικός μου πομποδέκτης έχει κωδικό 1234. Εισάγω λοιπόν τον κωδικό και πατάω OK.



Εικόνα 3.10: Πάτημα κωδικού για αντιστοίχιση

Επιστρέφω στην εφαρμογή μου και ξαναπατάω το κουμπί της αντιστοίχισης.



Εικόνα 3.11: Μενού για αντιστοίχιση

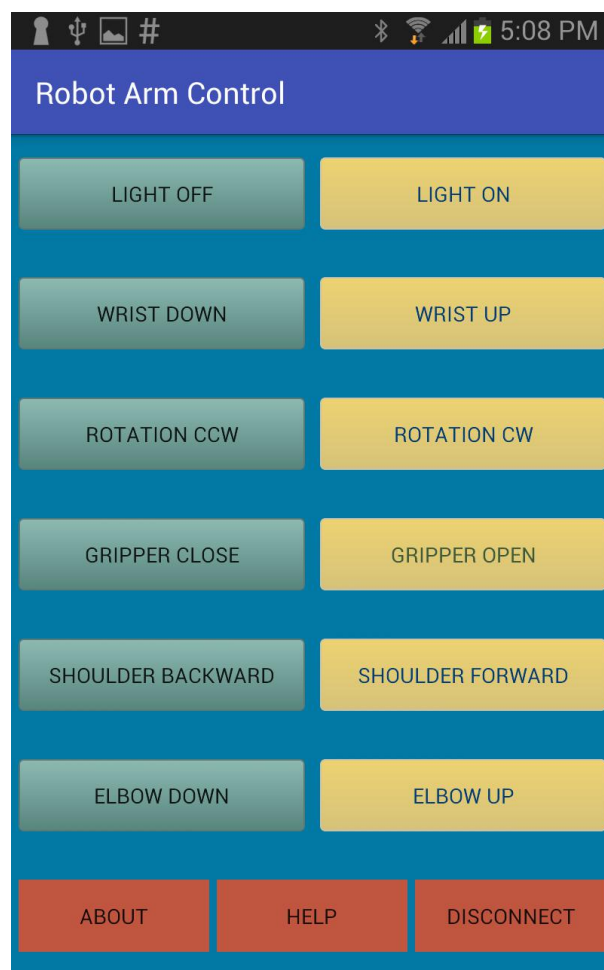
Τώρα μπορούμε να δούμε τον πομποδέκτη HC-05 ανάμεσα στις αντιστοιχισμένες συσκευές. Το επιλέγουμε με διπλό κλικ και μας πάει στην σελίδα με τα κουμπιά ελέγχου για το ρομποτικό βραχίονα. Αυτό γίνεται με το κάλεσμα της κλάσης ArmControl.class

```
1. //Method called when the device from the list is clicked.
```

```
2. deviceList.setOnItemClickListener(myItemClickListener);
```

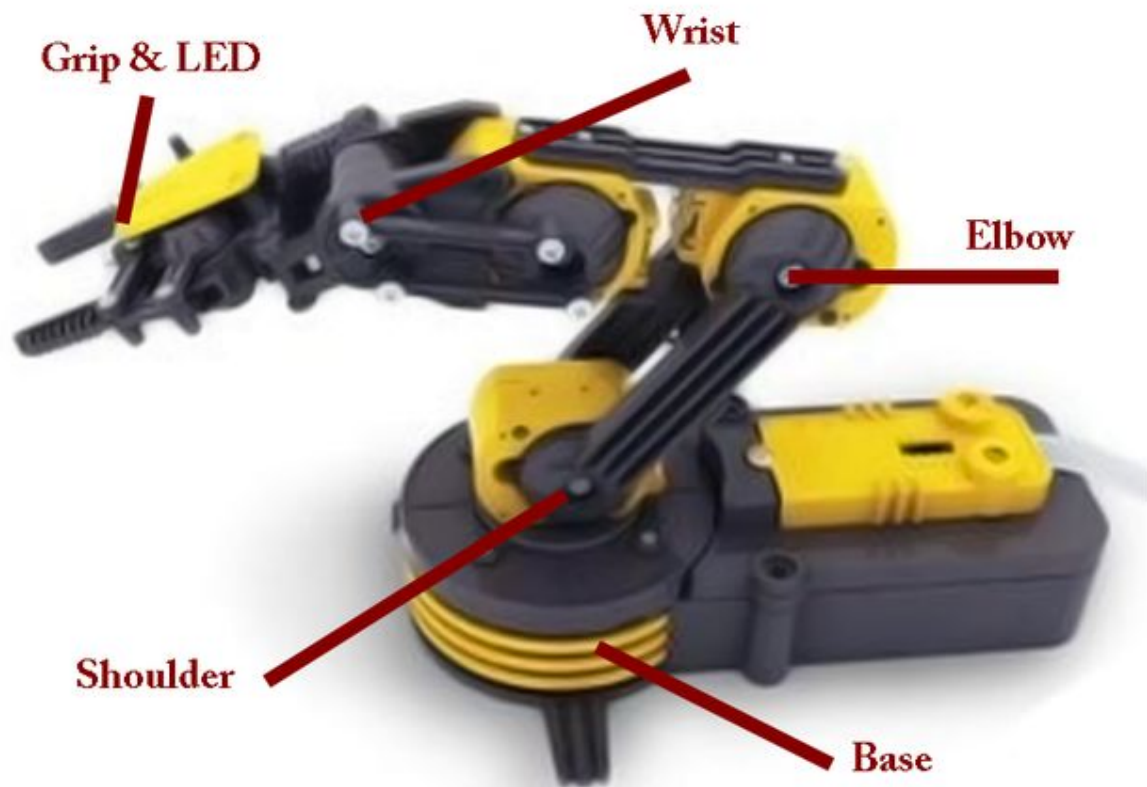
```
1. private AdapterView.OnItemClickListener myItemClickListener = new Adapter  
   AdapterView.OnItemClickListener()
```

```
public void onItemClick (AdapterView<?> av, View v, int arg2, long arg3)
2.     {
3.         // Get the device MAC address, the last 17 chars in the View.
4.         String info = ((TextView) v).getText().toString();
5.         String address = info.substring(info.length() - 17);
6.         // Make an intent to start next activity.
7.         Intent i = new Intent(PairingList.this, ArmControl.class);
8.         //Change the activity.
9.         i.putExtra(EXTRA_ADDRESS, address); //This will be received at Ar
        mControl (class) activity.
10.        startActivity(i);
11.    }
```



Εικόνα 3.12: Μενού ελέγχου για το ρομπωτικό βραχίονα

Για να καταλάβουμε σε ποιο κουμπί αντιστοιχεί η κάθε άρθρωση του βραχίονα, μας βοηθάει η παρακάτω εικόνα.



Εικόνα 3.13: Σχήμα ονομασιών αρθρώσεων βραχίονα

Εκτός από τα κουμπιά ελέγχου υπάρχουν και τα παρακάτω.

1. **Disconnect** (Αποσύνδεση) – Κουμπί αποσύνδεσης από το Bluetooth με το οποίο έχουμε συνδεθεί.

```
1. disconnect = (Button)findViewById(R.id.btnDisconnect);
2. disconnect.setOnClickListener(new View.OnClickListener()
3. {
4.     @Override
5.     public void onClick(View v)
6.     {
7.         Disconnect(); //close connection
8.     }
9. });
```

Με την χρήση του προηγούμενου κώδικα βάζω έναν ανιχνευτή κλικ OnClickListener στο κουμπί Disconnect. Μόλις κάποιος πληκτρολογήσει το κουμπί αυτό καλεί την συνάρτηση Disconnect() με την οποία αποσυνδεόμαστε από το πομποδέκτη με τον οποίο έχουμε συνδεθεί και γυρίζουμε πίσω στο αρχικό μενού με τις αντιστοιχισμένες συσκευές.

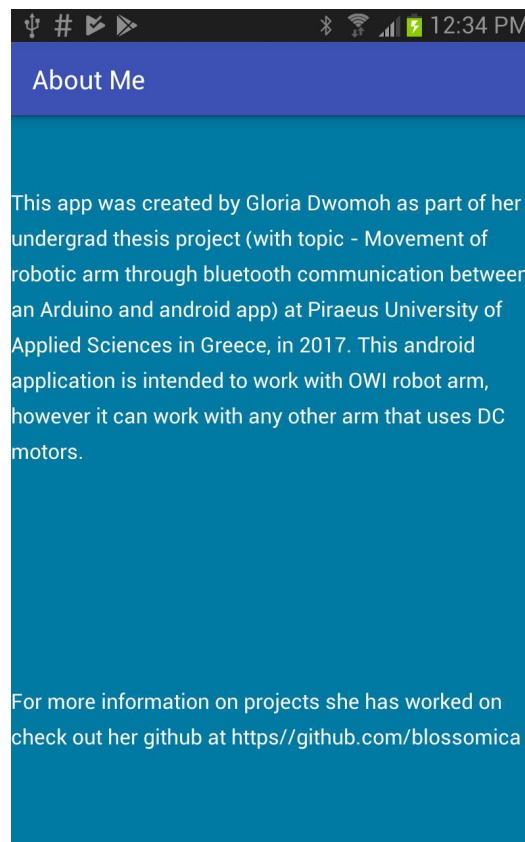
```
private void Disconnect()
{
    if (btSocket!=null) //If the btSocket is busy
    {
        try
        {
            btSocket.close(); //close connection
        }
        catch (IOException e)
        { msg("Error");}
    }
    finish(); //return to the first layout
```

10. Help (Βοήθεια) – Αυτό το συγκεκριμένο κουμπί δεν υπάρχει ποια στην εφαρμογή. Αλλά θα μπορούσε να υπάρξει στην περίπτωση που χρειάζονται πληροφορίες για το πως μπορεί να χρησιμοποιηθεί η εφαρμογή.

11. **About** (Σχετικές πληροφορίες) – Κουμπί με τα στοιχεία μου και πληροφορίες για την εφαρμογή.

```
public void aboutFunc(View v)
{
    if(v.getId() == R.id.btnAbout)
    {
        Intent i = new Intent(ArmControl.this, About.class);
        startActivity(i);
    }
}
```

Μέσω αυτού του κώδικα που βρίσκεται στην κλάση PairingList καλούμε την κλάση About η οποία θα εμφανίσει την νέα σελίδα.

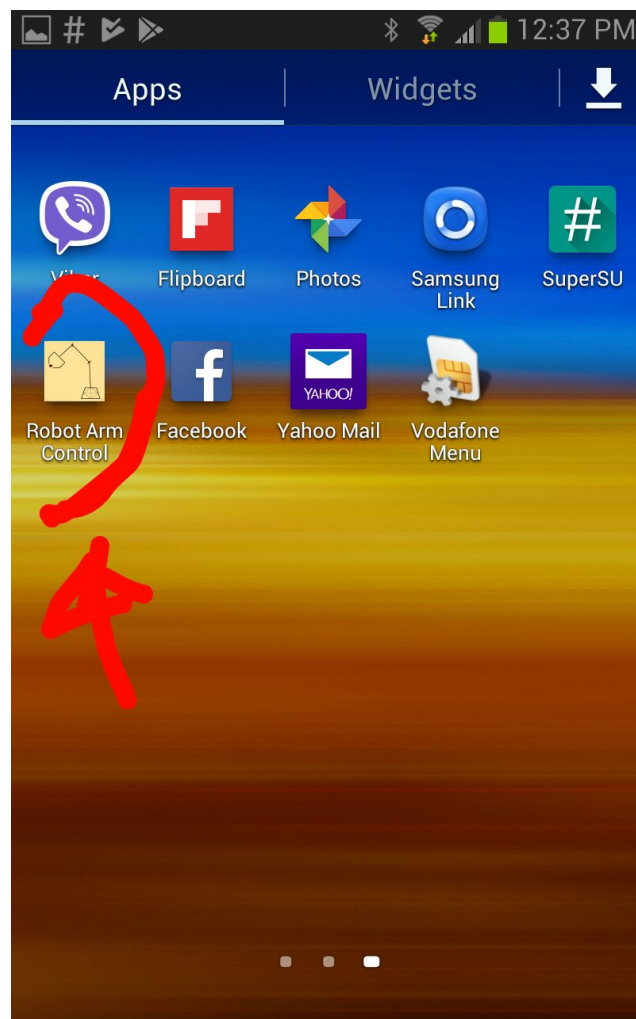


Εικόνα 3.14: Σελίδα About

Είναι αναγκαίο να επισημάνουμε ότι για να λειτουργήσει ο προηγούμενος κώδικας πρέπει να βάλουμε στο Android Manifest XML την παρακάτω εντολή.

1. `<activity`
2. `android:name="me.gloriadwomoh.robotarmproject.About"`
3. `android:label="About Me">`
4. `</activity>`

Αφού τρέξουμε την εφαρμογή για πρώτη φορά στο κινητό μας σε Developer Mode μπορούμε να την βρούμε στο κινητό μας και να το τρέξουμε ασύρματα. Το ειδικό εικονίδιο της εφαρμογής μπορείτε να το δείτε παρακάτω:



Εικόνα 3.15: Εικονίδιο εφαρμογής

Τώρα ας δούμε τι γίνεται στην κλάση ArmControl μέσω της οποίας χειρίζομαι την κίνηση του βραχίονα. Μπορείτε να δείτε την XML εικόνα της κλάσης αυτής από το σχήμα 3.12 που δόθηκε προηγουμένως.

Το Android έχει μια που ονομάζεται Bluetooth Adapter. Αυτή η βιβλιοθήκη αντιπροσωπεύει τον τοπικό προσαρμογέα Bluetooth της συσκευής. Ο προσαρμογέας Bluetooth μας επιτρέπει να εκτελέσουμε βασικές λειτουργίες Bluetooth, όπως τον εντοπισμό συσκευών, την αναζήτηση μιας λίστας συνδεδεμένων συσκευών, την δημιουργία ενός BluetoothDevice χρησιμοποιώντας μια γνωστή διεύθυνση MAC και δημιουργία ενός BluetoothServerSocket για να ακούσουμε τις αιτήσεις σύνδεσης από άλλες συσκευές και να ξεκινήσουμε την σάρωση για συσκευές Bluetooth.

```
1. private BluetoothAdapter myBluetooth = null;
2. private Set<BluetoothDevice> pairedDevices;
3. public static String EXTRA_ADDRESS = "device_address";
```

Στον παραπάνω κώδικα αρχικοποιώ κάποιες μεταβλητές. Η μία μεταβλητή είναι το Bluetooth Adapter myBluetooth η οποία είναι ο πομποδέκτης του κινητού μου. Ύστερα αρχικοποίησα ένα σέτ από πομποδέκτες που έχουμε δεσμεύσει με την συσκευή μας, μέσω αυτού θα κάνουμε μια λίστα με τις συσκευές μας. Τέλος βλέπουμε την μεταβλητή EXTRA_ADDRESS αφότου συνδεθούμε με έναν πομποδέκτη θα έχει την τιμή του πομποδέκτη αυτού.

Όταν πατηθεί το κουμπί Pair with Device που βλέπουμε στην εικόνα 3.11 καλείται η συνάρτηση pairedDeviceList() όπως βλέπουμε παρακάτω:

```
1. Button btnPair;
2. btnPair = (Button)findViewById(R.id.button);
3.
4. btnPair.setOnClickListener(new View.OnClickListener() {
5.     @Override
6.     public void onClick(View v)
7.     {
8.         pairedDevicesList(); } }
```

Στην συνάρτηση αυτή βλέπουμε αν υπάρχουν συσκευές με τις οποίες είναι δεσμευμένος ο πομποδέκτης του κινητού μας.

```
1.     private void pairedDevicesList()
2.     {
3.         pairedDevices = myBluetooth.getBondedDevices();
4.         ArrayList list = new ArrayList();
5.
6.         if (pairedDevices.size()>0)
7.         {
8.             for(BluetoothDevice bt : pairedDevices)
9.             {
10.                list.add(bt.getName() + "\n" + bt.getAddress()); //Get the device's
                name and the address.
11.            }
12.        }
13.        else
14.        {
15.            Toast.makeText(getApplicationContext(), "No paired bluetooth mod
                ules found. Pair and try again. ", Toast.LENGTH_LONG).show();
16.        }
17.
18.        final ArrayAdapter adapter = new ArrayAdapter(this,android.R.layout.
                simple_list_item_1, list);
19.        deviceList.setAdapter(adapter);
20.        //Method called when the device from the list is clicked.
21.        deviceList.setOnItemClickListener(myItemClickListener);
22.
23.    }
```

Αν ο πομποδέκτης μου δεν έχει δεσμευτεί με κανέναν άλλο πομποδέκτη δείχνω το μήνυμα "No paired bluetooth modules found. Pair and try again. " που λέει ότι δεν υπάρχουν δεσμευμένες συσκευές, επιπλέον με ενθαρρύνει να δεσμεύσω μια συσκευή πομποδέκτη και να ξαναδοκιμάσω.

Αν ο πομποδέκτης μου έχει δεσμευτεί με άλλους πομποδέκτες τους εμφανίζω έναν προς έναν σε μια λίστα. Μόλις πληκτρολογήσω έναν από τους πομποδέκτες καλείται η συνάρτηση `myItemClickListener`.

```
1. private AdapterView.OnItemClickListener myItemClickListener = new Adapter
   AdapterView.OnItemClickListener()
2. {
3.     public void onItemClick (AdapterView<?> av, View v, int arg2, long a
      rg3)
4.     {
5.         // Get the device MAC address, the last 17 chars in the View.
6.         String info = ((TextView) v).getText().toString();
7.         String address = info.substring(info.length() - 17);
8.
9.         // Make an intent to start next activity.
10.        Intent i = new Intent(PairingList.this, ArmControl.class);
11.
12.        //Change the activity.
13.        i.putExtra(EXTRA_ADDRESS, address); //This will be received at Ar
      mControl (class) activity.
14.        startActivity(i);
15.    }
16.};
```

Στην συνάρτηση αυτή βρίσκουμε την διεύθυνση MAC. Η διεύθυνση αυτή είναι ένας χρησιμοποιείται για την μοναδική αναγνώριση των προσαρμογέων (adapters) δικτύου υπολογιστών. Αυτοί οι αριθμοί ενσωματώνονται στο υλικό του δικτύου κατά τη διαδικασία κατασκευής ή αποθηκεύονται στο υλικό/λογισμικό και έχουν σχεδιαστεί έτσι ώστε να μην τροποποιούνται. Περνάω την διεύθυνση MAC της συσκευής που έχει πληκτρολογήσει ως επιπλέον πληροφορία στην μεταβλητή τύπου κλάσης `Intent` με την βοήθεια της συνάρτησης `putExtra()`. Η συνάρτηση `putExtra()` ανήκει στον τύπο αναγνωριστικών (attributes) `extras`. Τα αναγνωριστικά `Extras` είναι μια δέσμη αποτελούμενη από τυχόν πρόσθετες

πληροφορίες. Αυτό μπορεί να χρησιμοποιηθεί για την παροχή εκτεταμένων πληροφοριών στο στοιχείο.

Πριν πάμε παρακάτω ας κοιτάξουμε τον κώδικα Manifest.XML της εφαρμογής Android μας γιατί παίζει έναν πολύ σημαντικό ρόλο. Το Manifest αρχείο παρέχει βασικές πληροφορίες σχετικά με την εφαρμογή μας στο σύστημα Android, τις οποίες πρέπει να έχει το σύστημα πριν να μπορέσει να εκτελέσει οποιονδήποτε από τους κώδικες της εφαρμογής. Μεταξύ άλλων, το Manifest αρχείο κάνει τα εξής:

- Περιγράφει τα στοιχεία της εφαρμογής, τα οποία περιλαμβάνουν τις δραστηριότητες, τις υπηρεσίες, τους δέκτες εκπομπής και τους παρόχους περιεχομένου που συνθέτουν την εφαρμογή. Επίσης, ονομάζει τις κλάσεις που υλοποιούν κάθε ένα από τα συστατικά μέρη και δημοσιεύει τις δυνατότητές τους. Αυτές οι δηλώσεις ενημερώνουν το σύστημα Android για τα εξαρτήματα και τις συνθήκες υπό τις οποίες μπορούν να ξεκινήσουν.
- Ονομάζει το πακέτο Java για την εφαρμογή.
- Προσδιορίζει τις διαδικασίες που φιλοξενούν τα στοιχεία της εφαρμογής.
- Δηλώνει τα δικαιώματα που πρέπει να έχει η εφαρμογή για να αποκτήσει πρόσβαση σε προστατευμένα μέρη του API και να αλληλοεπιδράσει με άλλες εφαρμογές. Δηλώνει επίσης τα δικαιώματα που απαιτούνται από άλλους για να αλληλοεπιδράσουν με τα στοιχεία της εφαρμογής.
- Περιγράφει τις κλάσεις οργάνων που παρέχουν πληροφορίες σχεδίασης και άλλες πληροφορίες καθώς εκτελείται η εφαρμογή.
- Δηλώνει το ελάχιστο επίπεδο του API Android που απαιτείται από την εφαρμογή.
- Περιλαμβάνει τις βιβλιοθήκες στις οποίες πρέπει να συνδεθεί η εφαρμογή.

Ας δούμε ποιο αναλυτικά των κώδικα που περιέχει.

```
8. <uses-  
permission android:name="android.permission.BLUETOOTH_ADMIN" />  
9. <uses-permission android:name="android.permission.BLUETOOTH" />
```

Μέσω της γραμμής 8 επιτρέπω στις εφαρμογές να ανακαλύπτουν και να δεσμεύουν συσκευές Bluetooth και μέσω της γραμμής 9 επιτρέπω στις εφαρμογές να συνδεθούν με δεσμευμένες συσκευές Bluetooth.

```
6.      <activity android:name=".PairingList">
7.          <intent-filter>
8.              <action android:name="android.intent.action.MAIN" />
9.
10.         <category android:name="android.intent.category.LAUNCHER" /
11.         >
12.     </intent-filter>
13. </activity>
```

Το {action = android.app.action.MAIN} αντιστοιχεί σε όλες τις δραστηριότητες που μπορούν να χρησιμοποιηθούν ως σημεία εισόδου σε μια εφαρμογή.

{ category = android:name="android.intent.category.LAUNCHER" } χρησιμοποιείται από τον εκκινητή (launcher) για να συμπληρώσετε τη λίστα κορυφαίων επιπέδων.

```
1.      android:icon="@mipmap/ic_launcher"
2.      android:label="@string/app_name"
```

Μέσω αυτού του κώδικα λέμε στην εφαρμογή που να βρεί την εικόνα του εκκινητή της εφαρμογής και ποιο να είναι το αναγνωριστικό της εφαρμογής.

Ήρθε η ώρα να κοιτάξουμε το κομμάτι του κώδικα που κάνει μια από τις πιο σημαντικές δουλειές. Τι γίνεται όταν πατήσουμε το κουμπί που αντιστοιχεί στο να κινηθεί ο αγκώνας του ρομποτικού βραχίονα;

```
1. elbowB = (Button)findViewById(R.id.btnElbowBackward);
2. //+++++++start
3. elbowB.setOnClickListener(new View.OnClickListener() {
4.     @Override
5.     public boolean onTouch(View arg0, MotionEvent event) {
6.         if (event.getAction() == MotionEvent.ACTION_DOWN) {
7.             if (btSocket != null) {
8.                 try {
9.                     btSocket.getOutputStream().write("8".getBytes());
10.                } catch (IOException e) {
11.                    msg("Error");
12.                }
13.            }
14.        }
```

Όταν πατήσουμε το κουμπί της εφαρμογής Android μας που αντιστοιχεί στην κίνηση του αγκώνα, καλείται μια συνάρτηση που ονομάζεται onTouch. Για κάθε κίνηση του ρομπότ συνήθως έχουμε μια κίνηση προς τα μπρος ή προς τα πίσω, δεξιόστροφα ή αριστερόστροφα, πάνω ή κάτω. Αυτό είναι κάτι που γίνεται για την κίνηση του βραχίονα.

Όταν πατάω το κουμπί (MotionEvent.ACTION_DOWN), θα στείλει ως έξοδο μέσω του πομποδέκτη του κινητού τηλεφώνου ένα νούμερο. Αν γίνει οτιδήποτε λάθος θα στείλει ένα μήνυμα "Error" αντιθέτως αν πάνε όλα καλά το μήνυμα αυτό θα το λάβει ο πομποδέκτης που είναι συνδεδεμένος με το Arduino μου και θα κάνει την κίνηση που έχω αναθέσει στον αριθμό 8.

Πώς όμως σταματάμε το Arduino από το να κινεί τον βραχίονα επ'άπειρον; Ας δούμε το υπόλοιπο κομμάτι του κώδικα για αυτό.

```
1.     else if (event.getAction() == MotionEvent.ACTION_UP){
2.         if (btSocket!=null)
3.         {
4.             try
5.             {
6.                 btSocket.getOutputStream().write("9".getBytes());
7.             }
8.             catch (IOException e)
9.             {
10.                msg("Error");
11.            }
12.        }
13.    }
14.    else
15.        return false;
16.
17.    return false;
18. }
19. });
```

Από την στιγμή που αρχίζω να απελευθερώνω το πατημένο κουμπί που αντιστοιχεί στον αγκώνα του ρομποτικού βραχίονα ενεργοποιείται η συνάρτηση `motionEvent.ACTION_UP` η οποία θα στείλει ως έξοδο μέσω του πομποδέκτη του κινητού τηλεφώνου το νούμερο 9. Αν γίνει οτιδήποτε λάθος θα στείλει ένα μήνυμα "Error" αντίθετως αν πάνε όλα καλά το μήνυμα αυτό θα το λάβει ο πομποδέκτης που είναι συνδεδεμένος με το Arduino μου και θα σταματήσει την κίνηση του βραχίονα.

Μέσω αυτής της τεχνικής ενεργοποιούμε τις διάφορες κινήσεις του βραχίονα και ολοκληρώνουμε την υλοποίηση αυτής της εργασίας.

(Κενό φύλλο)

ΚΕΦΑΛΑΙΟ 4

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ

4.1 Σύνοψη της πτυχιακής εργασίας

Η εργασία αυτή καταδεικνύει έναν πιθανό τρόπο για τον χειρισμό ενός ρομποτικού βραχίονα μέσω της επικοινωνίας Bluetooth, μεταξύ μιας εφαρμογής Android και ενός κυκλώματος Arduino. Για αυτό το συγκεκριμένο έργο, χρησιμοποίησα έναν ρομποτικό βραχίονα OWI από το OWI Arm Edge Kit, καθώς είναι μια προσιτή επιλογή που το καθιστά ιδανικό για έργα μικρής κλίμακας. Ο ρομποτικός βραχίονας αυτός χρησιμοποιεί απλά DC μοτέρ (κινητήρες συνεχούς ρεύματος) σε κάθε άρθρωση για να πραγματοποιήσει την κίνηση του. Αντικαθιστώ το ενσωματωμένο κύκλωμα που περιλαμβάνει το kit OWI με άλλο κύκλωμα που σχεδιάστηκε με τη χρήση ενός board Arduino, PCB και των H-Bridge Chips. Για να ελέγξω την κίνηση του βραχίονα, δημιούργησα μια εφαρμογή Android που επιτρέπει σε ένα κινητό Android με λειτουργικότητα Bluetooth να επικοινωνεί με τη μονάδα Bluetooth που υπάρχει στο δικό μου κύκλωμα, επιτρέποντάς μου έτσι να εκτελώ στοχευμένες κινήσεις.

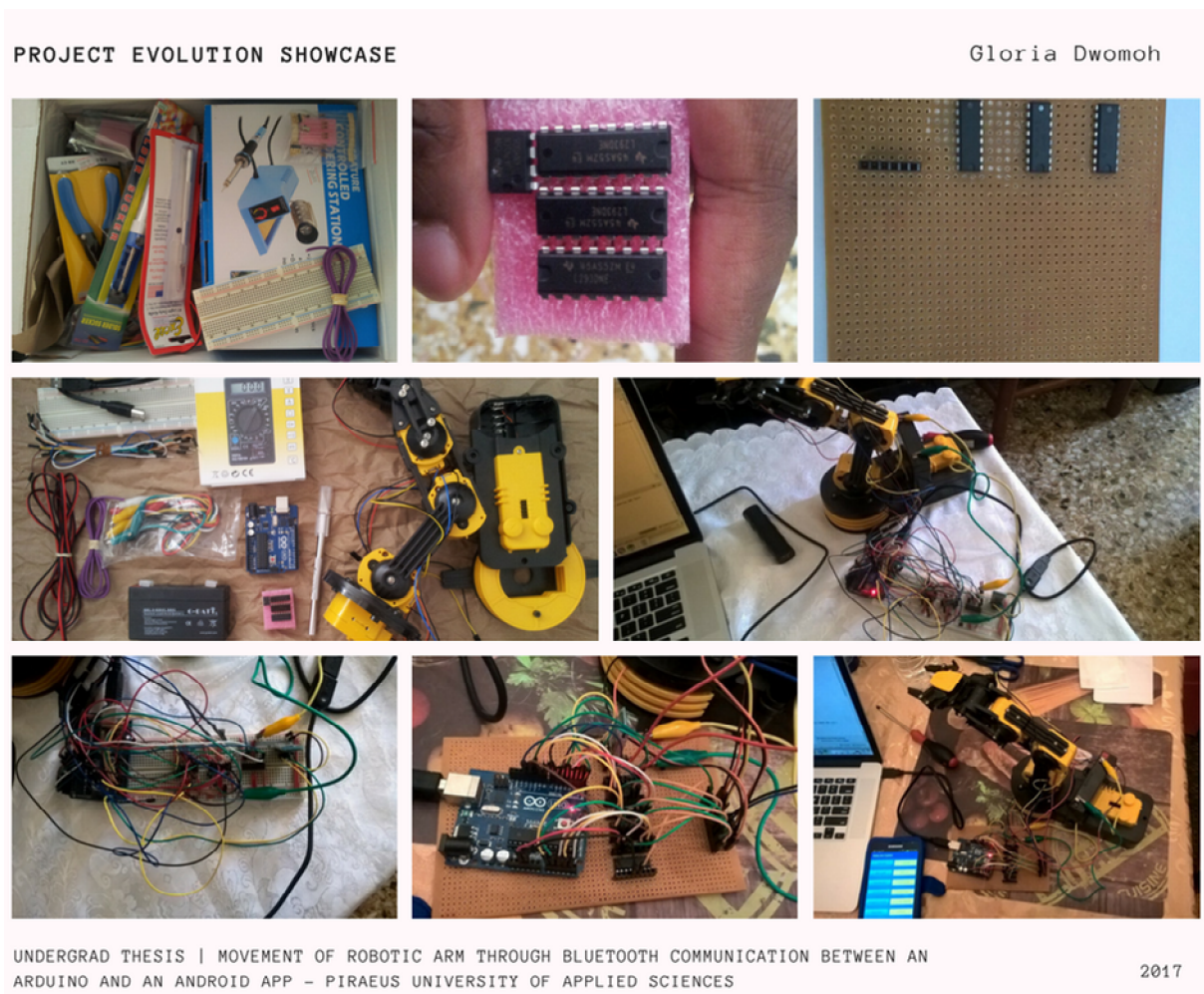
4.2 Προοπτικές

Μέσω της ασύρματης επικοινωνίας, και ειδικότερα μέσω του πρωτόκολλο Bluetooth μπορούμε να στέλνουμε και να λαμβάνουμε δεδομένα. Η ασύρματη επικοινωνία μας επέτρεψε να κάνουμε την καθημερινότητα πιο προσιτή σε άτομα με αναπηρίες ή κινητικούς περιορισμούς και να απλοποιήσουμε ορισμένες καθημερινές εργασίες.

Στο μέλλον, αυτή η εργασία θα μπορούσε να επεκταθεί για να επιτρέψει στον βραχίονα να κινηθεί και με την χρήση τροχών. Θα μπορούσε επιπλέον ο βραχίονας να κάνει χρήση της τεχνητής νοημοσύνης (artificial intelligence) και της τεχνητής όρασης (computer vision) δίνοντας του έτσι την δυνατότητα να αναγνωρίζει και να μετακινεί διάφορα αντικείμενα.

ΠΑΡΑΡΤΗΜΑ Α΄

Παρακάτω μπορείτε να δείτε μερικά από τα στάδια υλοποίησης της πτυχιακής. Το κύκλωμα άρχισε σαν ένα περίπλοκο κύκλωμα στο Breadboard ενώ στο τέλος κατέληξε σε μια απλούστερη μορφή σε PCB.



ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] About Us. Owirobot.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από <http://www.owirobot.com/about-us-1/>
- [2] Εικόνα Android Ανακτήθηκε από το <https://www.famouslogos.net/android-logo/>
- [3] Android (λειτουργικό σύστημα). En.wikipedia.org. Ανακτήθηκε στις 16 Ιουνίου 2017, από το [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [4] Arduino. En.wikipedia.org. Ανακτήθηκε στις 16 Ιουνίου 2017, από τη διεύθυνση <https://en.wikipedia.org/wiki/Arduino>
- [5] Arduino - BuildProcess. Arduino.cc. Ανακτήθηκε στις 16 Ιουνίου 2017, από τη διεύθυνση <https://www.arduino.cc/en/Hacking/BuildProcess>
- [6] Arduino - Συχνές ερωτήσεις. Arduino.cc. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://www.arduino.cc/en/Main/FAQ>
- [7] Εισαγωγή στο Arduino – Το απόλυτο geek toy. (2009). deltaHacker. Retrieved 16 June 2017, from <https://deltahacker.gr/arduino-intro/>
- [8] Arduino Με τη μονάδα Bluetooth HC-05 στη λειτουργία Slave Martyn Currey. (2014). Martyncurrey.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από το <http://www.martyncurrey.com/arduino-with-hc-05-bluetooth-module-in-slave-mode/>
- [9] Arduino - Εισαγωγή. Arduino.cc. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://www.arduino.cc/en/guide/introduction>
- [10] Arduino - Τροποποίηση βραχίονα ρομπότ - Lucky Larry. (2010). Luckylarry.co.uk. Ανακτήθηκε στις 16 Ιουνίου 2017, από το <http://luckylarry.co.uk/arduino-projects/arduino-modifying-a-robot-arm/>
- [11] Atmel AVR. En.wikipedia.org. Ανακτήθηκε στις 16 Ιουνίου 2017, από τη διεύθυνση https://en.wikipedia.org/wiki/Atmel_AVR
- [12] AVR-GCC. Ccrma.stanford.edu. Ανακτήθηκε στις 16 Ιουνίου 2017, από https://ccrma.stanford.edu/~juanig/articles/wiriavrlib/AVR_GCC.html

- [13]Βασικά στοιχεία Bluetooth - learn.sparkfun.com. Learn.sparkfun.com.
Ανακτήθηκε στις 16 Ιουνίου 2017, από
<https://learn.sparkfun.com/tutorials/bluetooth-basics>
- [14]BluetoothAdapter | Προγραμματιστές Android. Developer.android.com.
Ανακτήθηκε στις 16 Ιουνίου 2017, από
<https://developer.android.com/reference/android/bluetooth/BluetoothAdapter.html>
- [15]Διοικητής. Το Ras PiBerry και το Wiimote Controlled Robot Arm.
Ανακτήθηκε από το <http://www.instructables.com/id/Raspberry-Pi-and-Wiimote-controlled-Robot-Arm/>
- [16]Evans, B. (2013). Ξεκινώντας τον προγραμματισμό Arduino (σελ. 1,3,5,8).
[S.I.]: Απρίλιος.
- [17]Girish, M. (2016). Βασικό εκπαιδευτικό πρόγραμμα για το Bluetooth του Arduino. AVISHKAR. Ανακτήθηκε στις 16 Ιουνίου 2017, από
<https://igniteinnovateideas.wordpress.com/2016/04/18/arduino-bluetooth-basic-tutorial/>
- [18]HC05 Bluetooth Transceiver Module with TTL Outputs (with base).
Ανακτήθηκε από το http://www.amazon.in/HC05-Bluetooth-Transceiver-Module-Outputs/dp/B019OR9YVU/ref=pd_sbs_328_5/258-3163545-0504334?_encoding=UTF8&psc=1&refRID=15JK2AGK852KQ0XF0M4A
- [19]Ιστορικό βιομηχανικών ρομπότ. Robots.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://www.robots.com/education/industrial-robot-history>
- [20]Πρόθεση | Προγραμματιστές Android. Developer.android.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από
<https://developer.android.com/reference/android/content/Intent.html>
- [21]Τρεις νόμοι της ρομποτικής του Ισαάκ Ασίμοφ. Auburn.edu.
Ανακτήθηκε στις 16 Ιουνίου 2017, από
<http://www.auburn.edu/~vestmon/robotics.html>

- [22]Manifest.permission | Προγραμματιστές Android.
Developer.android.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://developer.android.com/reference/android/Manifest.permission.html>
- [23]Pololu - OWI-535 Ρομποτικό άκρο Edge Kit. Pololu.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://www.pololu.com/product/947>
- [24]Προγραμματισμός του Arduino Uno σε C. (2011). Ελευθερία ενσωματωμένη. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://balau82.wordpress.com/2011/03/29/programming-arduino-uno-in-pure-c/>
- [25]Unimate. En.wikipedia.org. Ανακτήθηκε στις 16 Ιουνίου 2017, από τη διεύθυνση <https://en.wikipedia.org/wiki/Unimate>
- [26]Εικόνα Unimate. Ανακτήθηκε από το http://images.computerhistory.org/timeline/timeline_ai_robotics_1961.unimate.jpg
- [27]Εισαγωγή στο Arduino - Το αληθινό geek παιχνίδι. (2009). DeltaHacker. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://deltahacker.gr/arduino-intro>

