



# **ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

## **ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

### **ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

#### **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Ασύρματος έλεγχος ρομποτικού βραχίονα μέσω Arduino και  
εφαρμογής android**

**Μαντακίδης Παντελεήμων**

**Μπούρας Ηλίας**

**Εισηγητής: Δρ. Πάρις Μαστοροκώστας, Καθηγητής**

## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΩΝ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Οι κάτωθι υπογεγραμμένοι **Μαντακίδης Παντελεήμων** του **Ιωάννη**, με αριθμό μητρώου **40749** και **Μπούρας Ηλίας** του **Αλέξιου**, με αριθμό μητρώου **40347**, φοιτητές του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής, πριν αναλάβουμε την εκπόνηση της Πτυχιακής Εργασίας μας, δηλώνουμε ότι ενημερωθήκαμε 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 απόφασή 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 βμήνου από την ημερομηνία ανάθεσής της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

## Ευχαριστίες

Για την διεκπεραίωση της παρούσας Πτυχιακής Εργασίας, θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα καθηγητή μας, κ. Πάρι Μαστοροκόστα για την επικοινωνιακή συνεργασία και την πολύτιμη συμβολή του στην ολοκλήρωσή της.

Ευχαριστούμε, επίσης, όλους τους καθηγητές μας για τις γνώσεις και δεξιότητες με τις οποίες μας εφοδίασαν κατά την διάρκεια των σπουδών μας.

Ευχαριστούμε, τέλος, τις οικογένειές μας, που μας στήριξαν τόσο κατά τη διάρκεια των σπουδών, όσο και κατά την συγγραφή της παρούσας εργασίας μας.

*This page is left blank intentionally*

## Περίληψη

Σκοπός της πτυχιακής αυτής είναι αφενός η ολοκλήρωσή της, ως υποχρέωσή μας για την διεκδίκηση του τίτλου σπουδών, αφετέρου δε, η παρουσίαση ταυτοχρόνως μιας ολοκληρωμένης λύσης που συνδυάζει Software και hardware, η οποία θα είναι και δείγμα του εργαστηρίου για τους νεότερους φοιτητές ότι το ίδρυμα σε μαθαίνει εκτός από θεωρία και να δημιουργείς.

Η θεματική της πτυχιακής είναι το Arduino και πως ένας ρομποτικός βραχίονας μπορεί να λειτουργήσει ασύρματα με bluetooth μέσω εφαρμογής android την οποία υλοποιήσαμε εμείς.

Σε hardware αναπτύσσουμε τον ρομποτικό βραχίονα που μας δόθηκε απ' το εργαστήριο και τον συνδέουμε πάνω στην πλακέτα με την βοήθεια 2 ολοκληρωμένων τύπου L293D τα οποία θα είναι υπεύθυνα και για την κίνηση του βραχίονα. Στη συνέχεια συνδέουμε πάνω στην πλακέτα το bluetooth για την ασύρματη σύνδεση με το κινητό/tablet και στην συνέχεια το Arduino.

Σε Software αναπτύσσουμε έναν κώδικα για Arduino ο οποίος θα δέχεται το σήμα απ' την εφαρμογή μέσω του bluetooth και με την βοήθεια και των ολοκληρωμένων θα το μετατρέπει σε κίνηση. Επιπλέον αναπτύσσουμε τον κώδικα της εφαρμογής ο οποίος αρχικά θα σου προτείνει διάφορες bluetooth συσκευές που αναγνωρίζει κοντά και στη συνέχεια θα επιλέγει ο χρήστης αυτή που είναι συνδεδεμένη στον βραχίονα. Κατόπιν και ανάλογα με το κλικ που θα γίνεται στην οθόνη ο κώδικας θα στέλνει στο bluetooth το αντίστοιχο σήμα για κίνηση μιας εκ των τριών αρθρώσεων η της αρπάγης.

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** Arduino, Android app, Bluetooth, βραχίονας

## Abstract

The purpose of this thesis is to complete our studies and acquire our degree, but at the same time, to present a demonstration to the younger students that the school wants you to create along with the studying.

The title of our thesis is defined as followed: “Wireless control of robotic arm OWI-535 through Arduino and android app”

As far as hardware is concerned the robotic arm along with the Arduino plate was provided to us from the school. We connected the robotic arm to the breadboard with the help of two L293D motor drivers (each responsible for moving 2 joints). Next, we connect on the breadboard the bluetooth module for wireless connection with the smartphone and the Arduino.

In terms of software, we develop an Arduino code which is going to get the signal from the android app through bluetooth and will move each joint. Furthermore, we developed the app code which suggests all available devices nearby and the user selects the one connected to the robotic arm. Next, depending on the screen click the code will send the proper signals to the Arduino in order to move one of the four joints.

**KEY WORDS:** Arduino, Android app, Bluetooth, robotic arm

## Περιεχόμενα

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΩΝ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ.....	2
Ευχαριστίες.....	3
Περίληψη.....	5
Abstract.....	6
Κεφάλαιο 1 <sup>ο</sup> : ΕΙΣΑΓΩΓΗ.....	10
1.1 Γενικά.....	10
1.2 Σκοπός και στόχοι της εργασίας.....	10
1.3 Δομή της εργασίας.....	10
1.4 BOM (Bill of Materials).....	12
Κεφάλαιο 2 <sup>ο</sup> : ΕΞΟΠΛΙΣΜΟΣ.....	13
2.1 Arduino.....	13
2.1.1 Τι είναι το Arduino.....	13
2.1.2 Τεχνολογία.....	13
2.1.3 Λογισμικό.....	14
2.1.4 Πλεονεκτήματα και μειονεκτήματα.....	15
2.2 Ρομποτικός βραχίονας OWI-535.....	17
2.2.1 Τι είναι ο ρομποτικός βραχίονας OWI-535.....	17
2.2.2 Εφαρμογές.....	17
2.2.3 Ο βραχίονας της εργασίας.....	17
2.3 Bluetooth module HC-05.....	18
2.3.1 Τι είναι το Bluetooth module HC-05;.....	18
2.3.2 Χαρακτηριστικά του HC-05.....	18
2.4 L293D motor drivers.....	19
2.4.1 Τι είναι οι L293D motor drivers.....	19
2.4.2 Χαρακτηριστικά L293D.....	19
Κεφάλαιο 3 <sup>ο</sup> : Συνδεσμολογία υλικού εξοπλισμού.....	21
3.1 Διάγραμμα κυκλώματος.....	21
3.2 Επεξήγηση συνδεσμολογιών.....	22
Κεφάλαιο 4 <sup>ο</sup> : Λογισμικά.....	23
4.1 Arduino software (IDE).....	23
4.1.1 Τι είναι το Arduino software (IDE).....	23
4.1.2 Χαρακτηριστικά.....	23

4.1.3 Πού χρησιμοποιήθηκε το IDE .....	24
4.2 MIT App Inventor .....	25
4.2.1 Τι είναι το MIT App Inventor.....	25
4.2.2 Χρήσεις.....	25
4.2.3 Λόγοι χρήσης MIT App Inventor .....	25
4.2.4 Περιβάλλον MIT App Inventor .....	25
Κεφάλαιο 5 <sup>ο</sup> : Χρήση λογισμικών για την δημιουργία του επιθυμητού αποτελέσματος.....	27
5.1 Χρήση MIT App Inventor .....	27
5.2 Χρήση Arduino IDE.....	31
5.2.1 Αναλυση κωδικα Arduino .....	37
Κεφάλαιο 6 <sup>ο</sup> : Επεξήγηση διαδικασίας εργασίας .....	41
6.1 Περιγραφή.....	41
6.2 Διαδικασία χρήσης της εφαρμογής.....	41
Βιβλιογραφία .....	46
Δικτυογραφία.....	47



## Πίνακας εικόνων

Εικόνα 1.Το arduino board .....	13
Εικόνα 2.Ο ρομποτικός βραχίονας OWI-535 .....	17
Εικόνα 3.Το HC-05 bluetooth module.....	18
Εικόνα 4.Ο L293D motor driver.....	19
Εικόνα 5.Τα pins του L293D .....	20
Εικόνα 6. Σχεδιασμος κυκλωματος με το προγραμμα Fritzing.....	21
Εικόνα 7.Arduino IDE .....	23
Εικόνα 8.MIT App Inventor .....	26
Εικόνα 9.Τυχαιο παραδειγμα.....	26
Εικόνα 10.Αρχικη οθονη .....	41
Εικόνα 11.Οθονη μετα το πατημα του connect .....	42
Εικόνα 12.Οθονη αφου συνδεθει.....	43
Εικόνα 13.About page.....	44

# Κεφάλαιο 1<sup>ο</sup>: ΕΙΣΑΓΩΓΗ

## 1.1 Γενικά

Το αποτέλεσμα της παρούσας πτυχιακής εργασίας προήλθε από τις γνώσεις που αποκτήθηκαν παρακολουθώντας τα μαθήματα Μηχατρονική, ασφάλεια και διαχείριση δικτυακών συστημάτων, λειτουργικά συστήματα, προγραμματισμός πυρήνα συστημάτων, μικροελεγκτές, δίκτυα, μικροηλεκτρονική, θεωρία κυκλωμάτων, ηλεκτρονικά.

## 1.2 Σκοπός και στόχοι της εργασίας

Το κίνητρο για την εκπόνηση της παρούσας εργασίας δόθηκε από την ιδέα της δημιουργίας ενός ρομποτικού βραχίονα ο οποίος θα μπορεί να έχει ελευθερία κινήσεων κατά το δοκούν. Ο κ. Πάρις Μαστοροκόστας δέχτηκε να μας κατευθύνει στην πραγματοποίηση της ιδέας μας, που ταυτοχρόνως είναι και η πτυχιακή μας εργασία.

Βασικός σκοπός της παρούσας εργασίας είναι (όπως αναφέρεται και στην περίληψη), η δημιουργία ενός ρομποτικού βραχίονα ο οποίος θα μπορεί να έχει ελευθερία κινήσεων κατά το δοκούν άλλα ταυτόχρονα θα είναι και δείγμα του εργαστηρίου για τους νεότερους φοιτητές ότι το ίδρυμα σε μαθαίνει εκτός από θεωρία και να δημιουργείς.

## 1.3 Δομή της εργασίας

Στο δεύτερο κεφάλαιο θα παρουσιαστεί αναλυτικά ο εξοπλισμός (hardware) που χρησιμοποιήθηκε για την δημιουργία του ρομποτικού βραχίονα. Για την διευκόλυνσή μας (συγγραφέων και αναγνωστών), θα χρησιμοποιηθούν υποκεφάλαια για την ανάλυση του κάθε κομματιού του εξοπλισμού, που βρίσκεται μεμονωμένο στην αγορά.

Στο τρίτο κεφάλαιο θα παρουσιαστεί αναλυτικά η συνδεσμολογία και ο συνδυασμός των προαναφερθέντων κομματιών υλικού εξοπλισμού, για την δημιουργία του επιθυμητού αποτελέσματος.

Στο τέταρτο κεφάλαιο θα παρουσιαστούν αναλυτικά τα λογισμικά που χρησιμοποιήθηκαν για την δημιουργία και της εφαρμογής αλλά και του Arduino. Για την διευκόλυνσή μας, θα χρησιμοποιηθούν υποκεφάλαια για την ανάλυση του κάθε λογισμικού, που βρίσκεται μεμονωμένο προς διανομή στο διαδίκτυο.

Στο πέμπτο κεφάλαιο θα παρουσιαστεί ο συνδυασμός των λογισμικών που χρησιμοποιήθηκαν για την κίνηση του βραχίονα σε βήματα ώστε να γίνει κατανοητή όλη η συλλογιστική και η εκτέλεση.

Στο έκτο κεφάλαιο θα περιγράψουμε το πώς πραγματοποιήσαμε την εργασία και θα εξηγήσουμε τη διαδικασία της πτυχιακής για την σωστή λειτουργία του project βήμα βήμα αναλυτικά από την στιγμή που θα ανοίξουμε το κινητό μέχρι και την στιγμή που θα κινηθεί ο βραχίονας.

#### 1.4 BOM (Bill of Materials)

<b>Parts</b>	<b>Price</b>
2x L293D motor drivers	2,00 €
Cables	4,60 €
Bluetooth HC-05	6,00 €
Power brick 12V DC	8,80 €
Breadboard	4,40 €
OWI-535	Provided
Arduino UNO	Provided

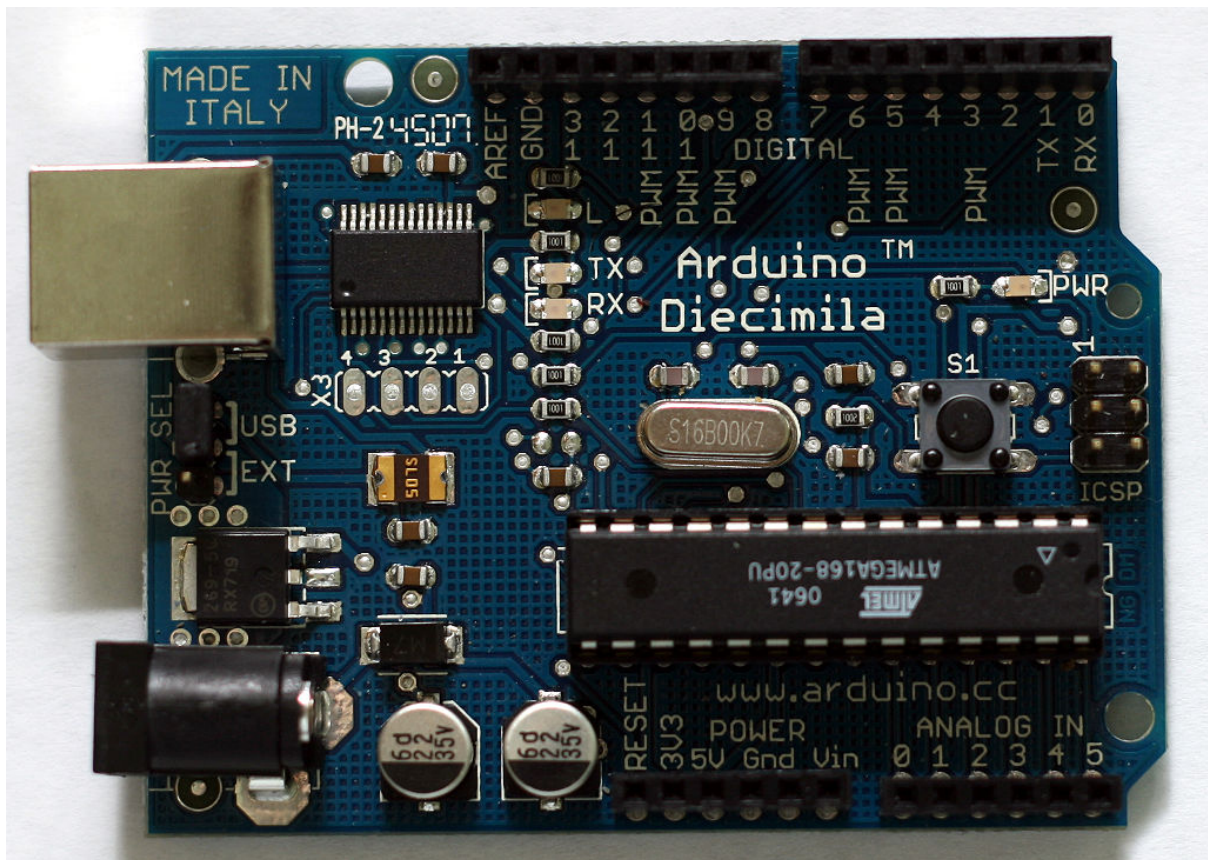
Total 25.80€

## Κεφάλαιο 2<sup>ο</sup>: ΕΞΟΠΛΙΣΜΟΣ

### 2.1 Arduino

#### 2.1.1 Τι είναι το Arduino

Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++).



Εικόνα 1. Το arduino board

#### 2.1.2 Τεχνολογία

Μία πλακέτα Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz. Ο μικροελεγκτής είναι από κατασκευής

προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Σε εννοιολογικό επίπεδο, στην χρήση του Arduino, όλα τα boards προγραμματίζονται με μία RS-232 σειριακή σύνδεση, αλλά ο τρόπος που επιτυγχάνεται αυτό διαφέρει σε κάθε hardware εκδοχή. Οι σειριακές πλάκες Arduino περιέχουν ένα απλό level shifter κύκλωμα για την μετατροπή του σήματος επιπέδου RS-232 σε TTL. Τα σημερινά Arduino προγραμματίζονται μέσω USB· αυτό καθίσταται δυνατό μέσω της εφαρμογής προσαρμογέων chip USB-to-Serial όπως το FTDI FT232. Κάποιες παραλλαγές, όπως το Arduino mini και το ανεπίσημο Boarduino, χρησιμοποιούν ένα αφαιρούμενο USB-to-Serial καλώδιο ή board, Bluetooth ή άλλες μεθόδους. (Όταν χρησιμοποιείται με παραδοσιακά εργαλεία microcontroller αντί για το Arduino IDE, χρησιμοποιείται πρότυπος προγραμματισμός AVR ISP).

Ο πίνακας Arduino εκθέτει τα περισσότερα microcontroller I/O pins για χρήση από άλλα κυκλώματα. Τα Diecimila, Duemilanove και το τρέχον Uno παρέχουν 14 ψηφιακά I/O pins, έξι από τα οποία μπορούν να παράγουν pulse-width διαμορφωμένα σήματα, και έξι αναλογικά δεδομένα. Αυτά τα pins βρίσκονται στην κορυφή του πίνακα μέσω female headers 0.1 ιντσών (2,2mm). Διάφορες εφαρμογές ασπίδων plug-in είναι εμπορικώς διαθέσιμες.

Το Arduino nano και το Arduino-Compatible Bare Bones Board και Boarduino Board ενδέχεται να παρέχουν male header pins στο κάτω μέρος του board προκειμένου να συνδέονται σε Breadboards. Υπάρχουν πολλά boards συμβατά με και προερχόμενα από Arduino boards. Κάποια είναι λειτουργικά ισάξια με ένα Arduino και μπορεί να χρησιμοποιηθούν εναλλακτικά. Πολλοί είναι το βασικό Arduino με την προσθήκη καινοτόμων output drivers, συχνά για την χρήση σχολικής μόρφωσης για να απλοποιήσουν την κατασκευή buggies και μικρών robot. Άλλες είναι ηλεκτρικά ισάξια αλλά αλλάζουν τον παράγοντα μορφής, επιτρέποντας κάποιες φορές την συνεχόμενη χρήση των Shields ενώ κάποιες όχι.

### 2.1.3 Λογισμικό

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino είναι μία εφαρμογή γραμμένη σε Java, που λειτουργεί σε πολλές πλατφόρμες και προέρχεται από το IDE για τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Έχει σχεδιαστεί για να εισαγάγει στον προγραμματισμό τους καλλιτέχνες και τους νέους που δεν είναι εξοικειωμένοι με την ανάπτυξη λογισμικού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης και ο συνδυασμός αγκύλων και είναι επίσης σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα με ένα μόνο κλικ. Δεν υπάρχει συνήθως καμία ανάγκη να επεξεργαστείτε αρχεία make ή να τρέξετε προγράμματα σε ένα περιβάλλον γραμμής εντολών. Ένα πρόγραμμα ή κώδικας που γράφτηκε για Arduino ονομάζεται σκίτσο (sketch).

Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring", από το πρωτότυπο σχέδιο Wiring, γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Οι χρήστες πρέπει μόνο να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

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

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

Ένα τυπικό πρώτο πρόγραμμα για έναν μικροελεγκτή αναβοσβήνει απλά ένα LED. Είναι ένα χαρακτηριστικό των περισσότερων πλακετών Arduino ότι έχουν ένα LED και μία αντίσταση φορτίου που συνδέονται μεταξύ του pin 13 και του εδάφους, ένα βολικό χαρακτηριστικό για πολλά απλά τεστ. Ο προηγούμενος κώδικας δεν θα αναγνωριστεί από ένα κανονικό μεταγλωττιστή C++ ως έγκυρο πρόγραμμα, έτσι ώστε όταν ο χρήστης κάνει κλικ στο κουμπί "Upload to I / O board" στο IDE, ένα αντίγραφο του κώδικα θα γραφτεί σε ένα προσωρινό αρχείο με ένα παραπάνω include στην κορυφή και μία πολύ απλή συνάρτηση main() στο τέλος, για να φτιάξει ένα έγκυρο C++ πρόγραμμα.

Το IDE του Arduino χρησιμοποιεί το GNU toolchain και το AVR Libc για να μεταγλωττίζει προγράμματα και το avrdude για να φορτώνει προγράμματα στην πλακέτα.

Δεδομένου ότι η πλατφόρμα Arduino χρησιμοποιεί Atmel μικροελεγκτές, το περιβάλλον ανάπτυξης της Atmel, το AVR Studio ή το νεότερη έκδοση του Atmel Studio, μπορεί επίσης να χρησιμοποιηθεί για την ανάπτυξη λογισμικού για το Arduino.

#### 2.1.4 Πλεονεκτήματα και μειονεκτήματα

Το Arduino έχει τα εξής πλεονεκτήματα:

- Πάρα πολύ εύκολα για να χρησιμοποιηθούν καθώς είναι έτοιμα για χρήση ανά πάσα στιγμή.
- Εύκολα προγραμματίσιμα λόγω της τεχνολογίας που χρησιμοποιούν.
- Έχει πολλά παραδείγματα για να μπορέσει και κάποιος που δεν είναι προγραμματιστής να το δουλέψει χωρίς πρόβλημα.
- Εύχρηστο UI σε βοηθάει στο debugging και στις πολλές επιλογές που έχει (serial monitor, επιλογή εξόδου κ.α.).

Ως μειονέκτημα θα μπορούσε να καταλογίσει κανείς στο Arduino:

- Μικρή επεξεργαστική ισχύ.
- Περιορισμός σε Input / output pins.



## 2.2 Ρομποτικός βραχίονας OWI-535

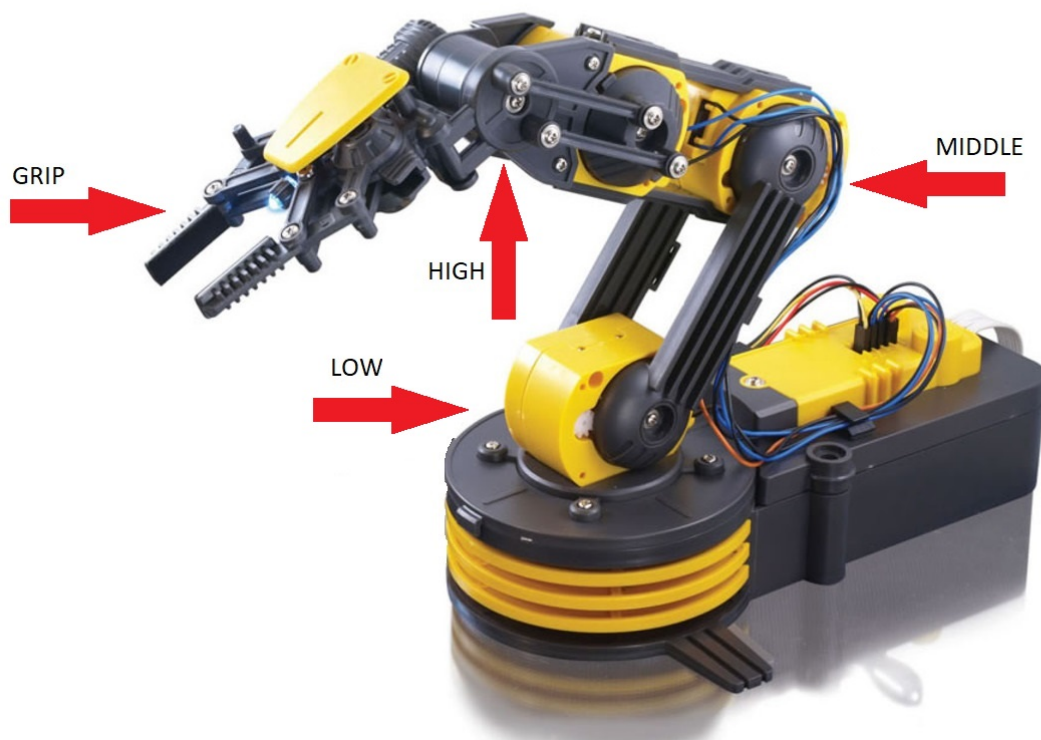
### 2.2.1 Τι είναι ο ρομποτικός βραχίονας OWI-535

Ο OWI-535 είναι ένας ρομποτικός βραχίονας ο οποίος έχει 4 αρθρώσεις. Η αρπάγη έχει δυνατότητα να ανοιγοκλείνει μέχρι 4cm, η ψηλή άρθρωση έχει δυνατότητα κίνησης 120°, η μεσαία άρθρωση έχει δυνατότητα κίνησης σχεδόν 280°, ενώ η βάση έχει σχεδόν 180°. Επιπλέον η αρπάγη μπορεί να σηκώσει και μέχρι 100g.

### 2.2.2 Εφαρμογές

Η εφαρμογή του συγκεκριμένου ρομποτικού βραχίονα είναι καθαρά για μικρές ανάγκες. Σε μια μεγαλύτερη κλίμακα και με μεγαλύτερη δύναμη όμως μπορούμε να πούμε ότι θα μπορούσε να χρησιμεύσει σε διάφορες καταστάσεις όπως για παράδειγμα η μικροιατρική και η μικροηλεκτρονικές κατασκευές εκεί όπου τα ανθρώπινα χέρια δεν έχουν την κατάλληλη σταθερότητα/δύναμη. Οι τηλεχειριζόμενοι γερανοί επίσης βασίζονται στην ίδια λογική για κατασκευές μεγάλων κτιρίων. Συνεπώς, ναι μεν για την κλίμακα μπορεί να είναι κάτι το οποίο δεν έχει πολλές πρακτικές εφαρμογές όμως σε μια πιο μεγάλη κλίμακα το βλέπουμε ήδη σε πολλές περιπτώσεις γύρω μας.

### 2.2.3 Ο βραχίονας της εργασίας



Εικόνα 2.Ο ρομποτικός βραχίονας OWI-535

## 2.3 Bluetooth module HC-05

### 2.3.1 Τι είναι το Bluetooth module HC-05;

Το HC-05 είναι ένας εκπομπός/δέκτης Bluetooth ο οποίος συστήνεται για android εφαρμογές και μπορεί να επικοινωνεί με έναν αντίστοιχο εκπομπό/δέκτη ανταλλάσσοντας σήματα και μετατρέποντας τα σε οτιδήποτε θέλει ο κάθε χρήστης.

### 2.3.2 Χαρακτηριστικά του HC-05

Το HC-05 έχει μικρό μέγεθος και διαθέτει 6 ακίδες για μεταφορά δεδομένων. Αυτές είναι οι εξής:



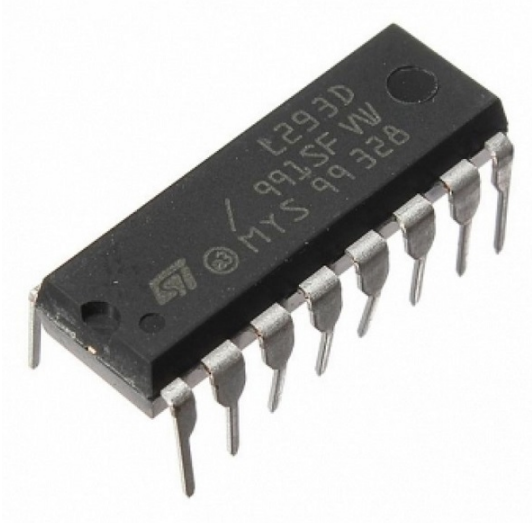
Εικόνα 3. Το HC-05 bluetooth module

- STATE: Η ακίδα που δείχνει στον χρήστη αν το HC-05 είναι συνδεδεμένο ή όχι.
- RXD: Η ακίδα για την ασύρματη υποδοχή δεδομένων
- TXD: Η ακίδα που υπάρχει για την ασύρματη εκπομπή των δεδομένων.
- GND: Η ακίδα για την γείωση
- VCC: Υπεύθυνο για την παροχή ρεύματος στα 5V ή 3.3V
- EN: Το EN χρησιμοποιείται για να μπορεί ο χρήστης να βάλει το HC-05 σε κατάσταση για να μπορεί να δέχεται δεδομένα (αν αυτό μέσα απ' το πρόγραμμα έχει την τιμή LOW). Η προκαθορισμένη τιμή είναι αυτή οπότε βοηθάει στην κατευθείαν λειτουργία του πομπού.

## 2.4 L293D motor drivers

### 2.4.1 Τι είναι οι L293D motor drivers

Οι L293D motor drivers είναι ολοκληρωμένα τα οποία ελέγχουν την κίνηση σε κινητήρες DC. Διαθέτουν 16 ακίδες (8 από κάθε μεριά) οι οποίες μπορούν να δώσουν κίνηση σε 2 κινητήρες. Μπορούν επίσης να δώσουν κίνηση και στις δυο κατευθύνσεις καθώς και να ελέγξουν τον χρόνο που θα κινείται ο κάθε κινητήρας.

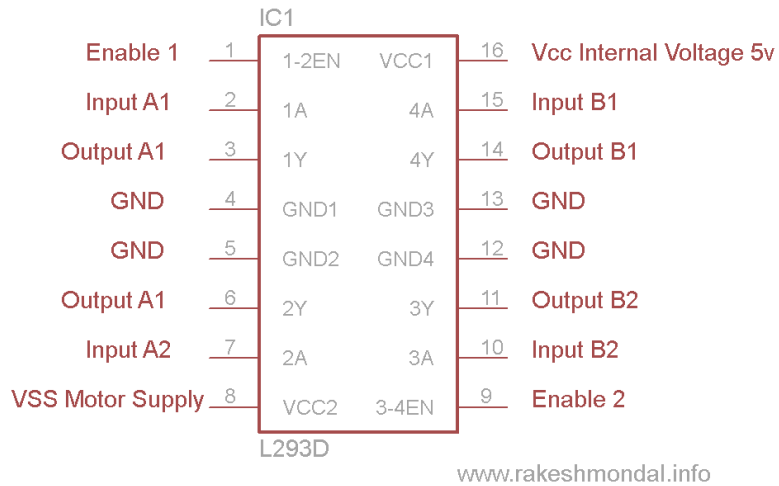


Εικόνα 4.Ο L293D motor driver

### 2.4.2 Χαρακτηριστικά L293D

Χαρακτηριστικά:

- Μπορεί να κινήσει δυο κινητήρες μαζί. Η κάθε μεριά μπορεί να κινήσει έναν κινητήρα.
- Μέσω του προγράμματος μπορεί να κινήσει τον κάθε κινητήρα σε διαφορετική μεριά και με διαφορετική ταχύτητα.
- Δέχεται εξωτερική παροχή ρεύματος για να μπορέσουν να κινηθούν και οι τέσσερις αρθρώσεις (ακίδα No 8).
- Κλείνει αυτόματα όταν θερμανθεί παραπάνω.



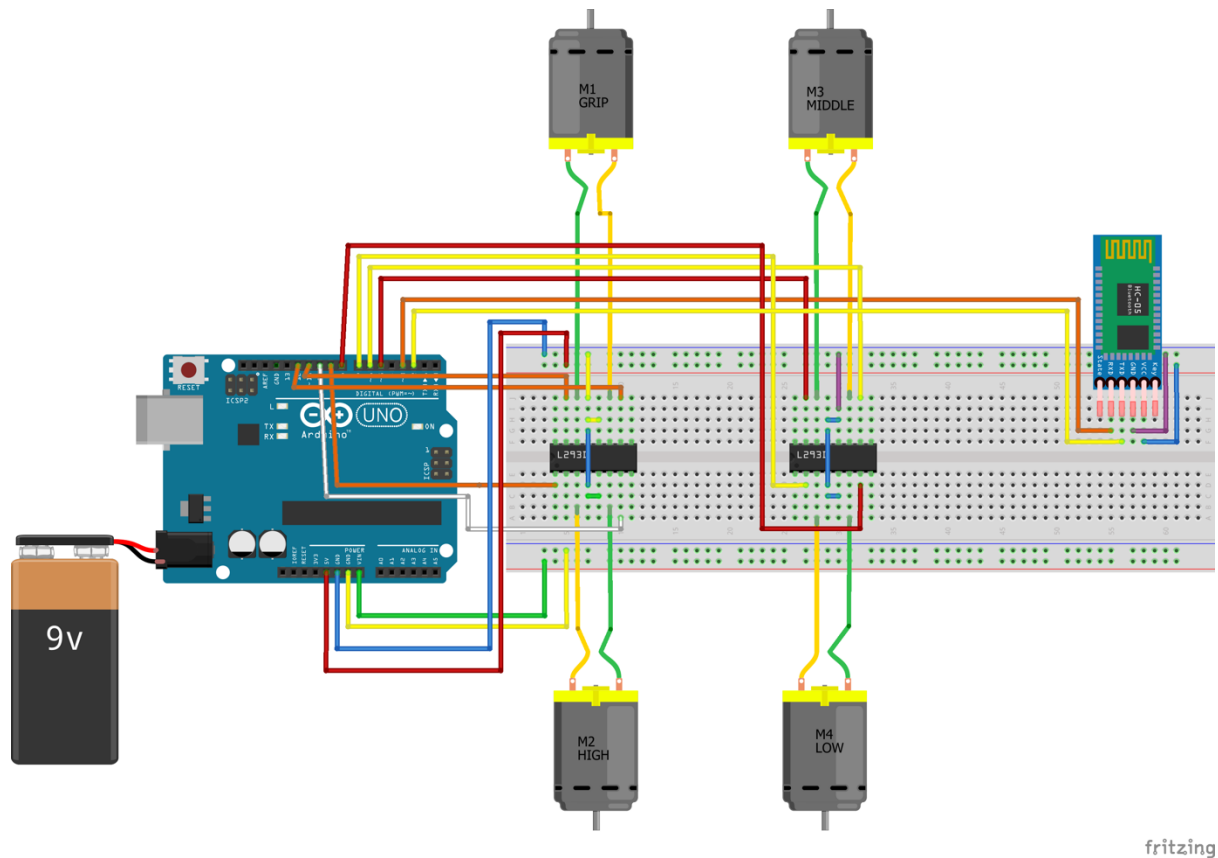
*Εικόνα 5. Τα pins του L293D*

Αναλυτικά η λειτουργία των ακίδων:

1. Ενεργοποιεί τις ακίδες 2, 7.
2. Ελέγχει την ακίδα 3.
3. Συνδέεται με το ένα άκρο του κινητήρα 1.
4. Γείωση.
5. Γείωση.
6. Συνδέεται με το άλλο άκρο του κινητήρα 1.
7. Ελέγχει την ακίδα 6.
8. Συνδέεται με την παροχή ρεύματος (4.5V–36V).
9. Ενεργοποιεί τις ακίδες 10, 15.
10. Ελέγχει την ακίδα 11.
11. Συνδέεται με το ένα άκρο του κινητήρα 2.
12. Γείωση.
13. Γείωση.
14. Συνδέεται με το άλλο άκρο του κινητήρα 2.
15. Ελέγχει την ακίδα 14.
16. Συνδέεται στην παροχή +5V.

## Κεφάλαιο 3<sup>ο</sup>: Συνδεσμολογία υλικού εξοπλισμού

### 3.1 Διάγραμμα κυκλώματος



fritzing

Εικόνα 6. Σχεδιασμός κυκλώματος με το πρόγραμμα Fritzing

### 3.2 Επεξήγηση συνδεσμολογιών

Σε αυτό το κομμάτι θα περιγράψουμε τις συνδεσμολογίες του Arduino. Αρχικά συνδέουμε το pin TXD με το No2 του Arduino και το RXD με το No3 του Arduino. Το GND του bluetooth με το (-) του breadboard και αντίστοιχα το VCC με το (+) του breadboard.

Για τη συνδεσμολογία του motor driver θα περιγράψουμε μια φορά (από τις 4 παρόμοιες που έχουμε κάνει) και είναι ακριβώς ίδια και για τα δυο motor drivers και για τους 4 κινητήρες που έχουμε συνδέσει πάνω τους.

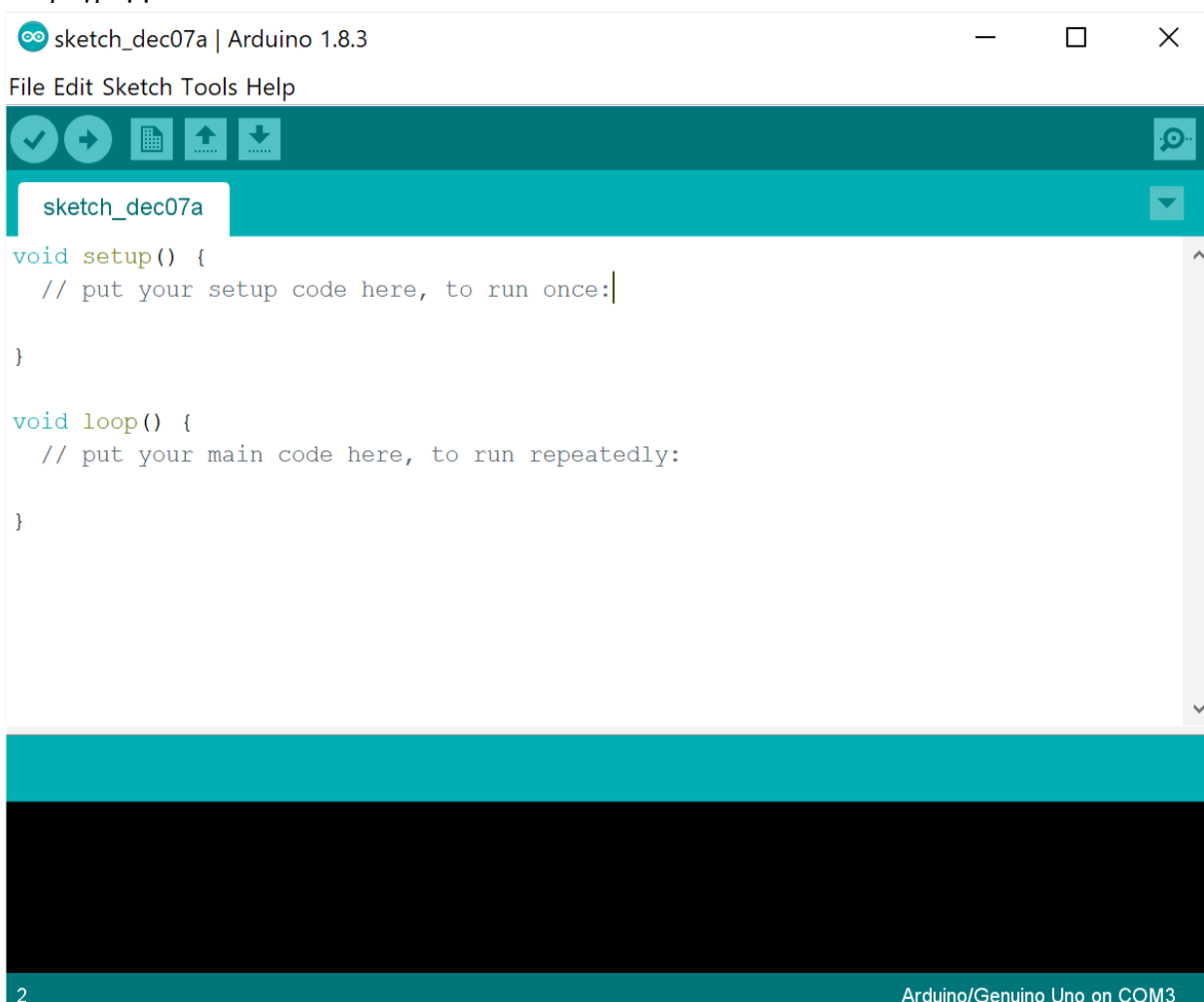
Ξεκινάμε από το δεύτερο pin το οποίο είναι συνδεδεμένο με καλώδιο στην υποδοχή του Arduino No7. Στα pins 3 και 6 συνδέσαμε τις δυο άκρες των κινητήρων ενώ τα pin 4 και 5 τα γειώσαμε (γεφυρώνοντας μαζί με τον «απέναντι» κινητήρα στο (-) του breadboard). Και τέλος το pin 7 το συνδέσαμε στην υποδοχή No8 στο Arduino.

## Κεφάλαιο 4<sup>ο</sup>: Λογισμικά

### 4.1 Arduino software (IDE)

#### 4.1.1 Τι είναι το Arduino software (IDE)

Το Arduino Integrated Development Environment (IDE) είναι μια πλατφόρμα ανάπτυξης λογισμικού το οποίο περιέχει έναν επεξεργαστή κειμένου για παραγωγή κώδικα, μια περιοχή η οποία αφορά το debugging και μια μπάρα με εργαλεία τα οποία βοηθούν τον χρήστη να έχει άμεση πρόσβαση σε βασικές λειτουργίες του προγράμματος. Μέσω αυτών συνδέεται με το εκάστοτε Arduino για να ανεβάσει στην πλακέτα του το αντίστοιχο πρόγραμμα.



Εικόνα 7. Arduino IDE

#### 4.1.2 Χαρακτηριστικά

Το Arduino IDE είναι μια εφαρμογή η οποία είναι συμβατή με όλα τα λογισμικά (Windows, macOS, Linux) και είναι γραμμένη σε Java. Προήλθε από τις γλώσσες Processing και Wiring. Υποστηρίζει τις γλώσσες C και C++ χρησιμοποιώντας διαφορετικούς κανόνες σχετικά με την δομή του κώδικα. Έχει δυο βασικές συναρτήσεις που πρέπει να

συμπληρωθούν για να δουλέψει το πρόγραμμα. Αυτές είναι οι `void setup()` και `void loop()`. Στην πρώτη γίνεται η αρχικοποίηση και η δήλωση των μεταβλητών που θα χρησιμοποιηθούν στο πρόγραμμα και στην δεύτερη που είναι και το κυρίως μέρος της πλατφόρμας γράφεται ο κώδικας. Χαρακτηριστικό της πλατφόρμας είναι ότι ο βασικός κώδικας όταν φορτωθεί επιτυχώς στην πλακέτα, επαναλαμβάνεται συνέχεια μέχρι να διακοπή η σύνδεση της πλακέτας με την παροχή ρεύματος.

#### 4.1.3 Πού χρησιμοποιήθηκε το IDE

Χρησιμοποιήσαμε το IDE για την παραγωγή κώδικα ώστε να φορτωθεί στο Arduino ο οποίος δέχεται ένα σήμα από το HC-05 και ανάλογα με την τιμή που θα δεχτεί θα δώσει εντολή στα δυο ολοκληρωμένα να κινήσουν τον αντίστοιχο κινητήρα σύμφωνα με την φορά που θέλει ο χρήστης.

#### 4.1.3 Ανάλυση βασικών λειτουργιών του IDE

Χρησιμοποιήσαμε το IDE για την παραγωγή κώδικα ώστε να φορτωθεί στο Arduino ο οποίος δέχεται ένα σήμα από το HC-05 και ανάλογα με την τιμή που θα δεχτεί θα δώσει εντολή



Verify. Ελέγχει αν ο κώδικας έχει λάθη.



Upload. Ανεβάζει τον κώδικα στην πλακέτα Arduino αφού τον κάνει compile.



New. Δημιουργεί ένα καινούργιο σχέδιο (sketch)



Open. Ανοίγει ένα παράθυρο με όλα τα σχέδια του χρήστη.



Save. Αποθηκεύει την πρόοδο του σχεδίου.



Serial Monitor. Ανοίγει την εικονική οθόνη



## 4.2 MIT App Inventor

### 4.2.1 Τι είναι το MIT App Inventor

Το MIT App Inventor είναι μια δημοφιλής υπηρεσία της Google η οποία δίνει την ευκαιρία σε νέους όχι και τόσο έμπειρους δημιουργούς εφαρμογών να φτιάξουν με απλό τρόπο εφαρμογές οι οποίες (προς το παρόν) είναι μόνο για λειτουργικά συστήματα android. Χαρακτηριστικό της υπηρεσίας αυτής είναι ότι δεν έχει κώδικα (με την κοινή έννοια) αλλά έχει blocks τα οποία είναι πιο εύκολα παραμετροποιήσιμα και πιο απλά για έναν άπειρο προγραμματιστή. Η υπηρεσία αυτή δημιουργήθηκε απ' την Google αλλά υποστηρίζεται απ' το πανεπιστήμιο MIT.

### 4.2.2 Χρήσεις

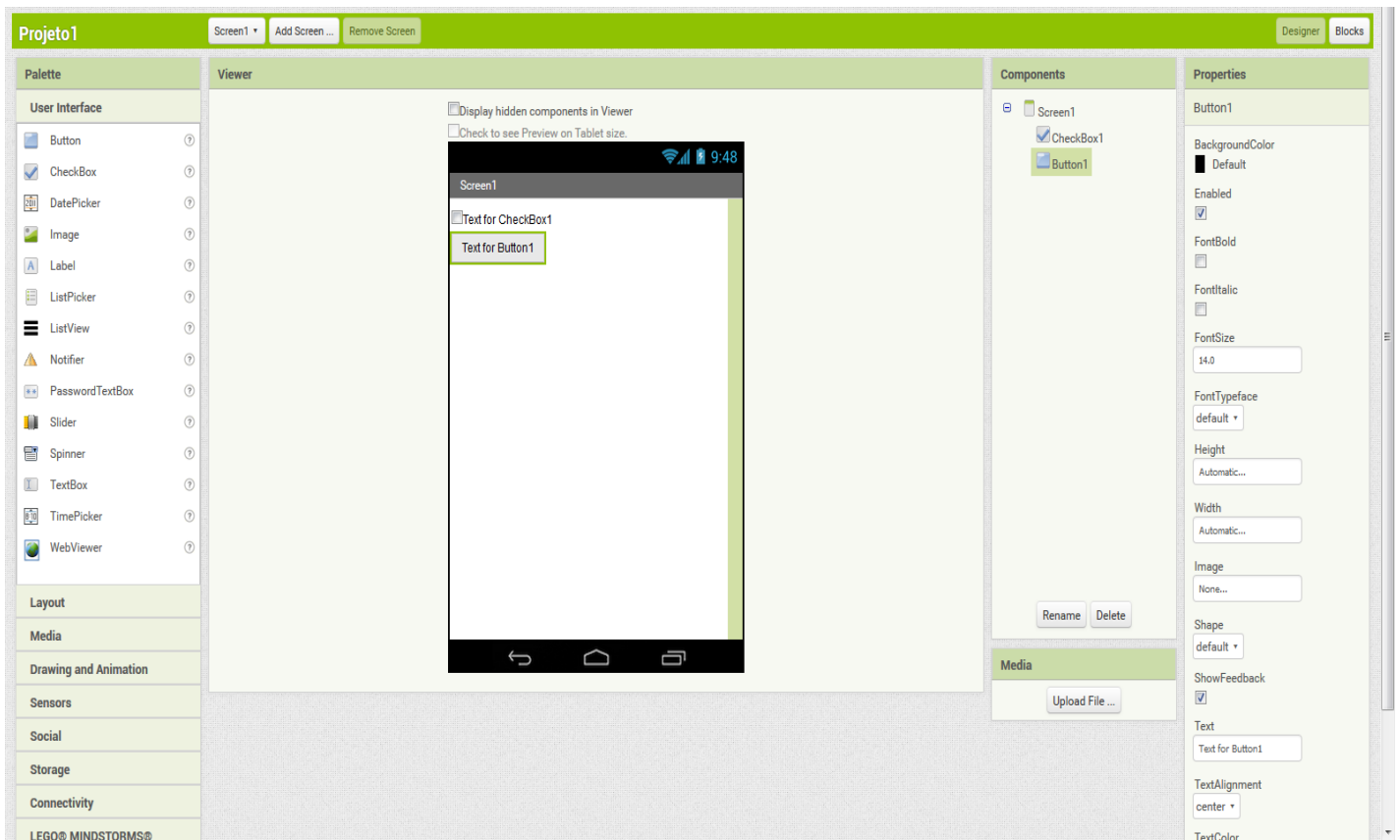
Το MIT App Inventor είναι δημοφιλές σε project τα οποία περιέχουν ρομποτικούς βραχιόνες η ρομποτικά οχήματα και χρησιμοποιείται για τον χειρισμό από το κινητό/tablet. Πέρα από αυτές τις χρήσεις μην ξεχνάμε ότι είναι μια υπηρεσία που αφήνει κάποιον να κάνει την δική του εφαρμογή οπότε οι χρήσεις είναι τόσες όσες και η φαντασία του καθενός.

### 4.2.3 Λόγοι χρήσης MIT App Inventor

Χρησιμοποιήσαμε το MIT App Inventor για τον λόγο ότι είναι πολύ εύκολο σε χρήση με τα blocks και παρότι δεν είχαμε καμία εμπειρία με τον συγκεκριμένο τρόπο λειτουργίας τον βρήκαμε πάρα πολύ εύκολο για να τον μάθουμε και να δημιουργήσουμε την εφαρμογή απ' την αρχή.

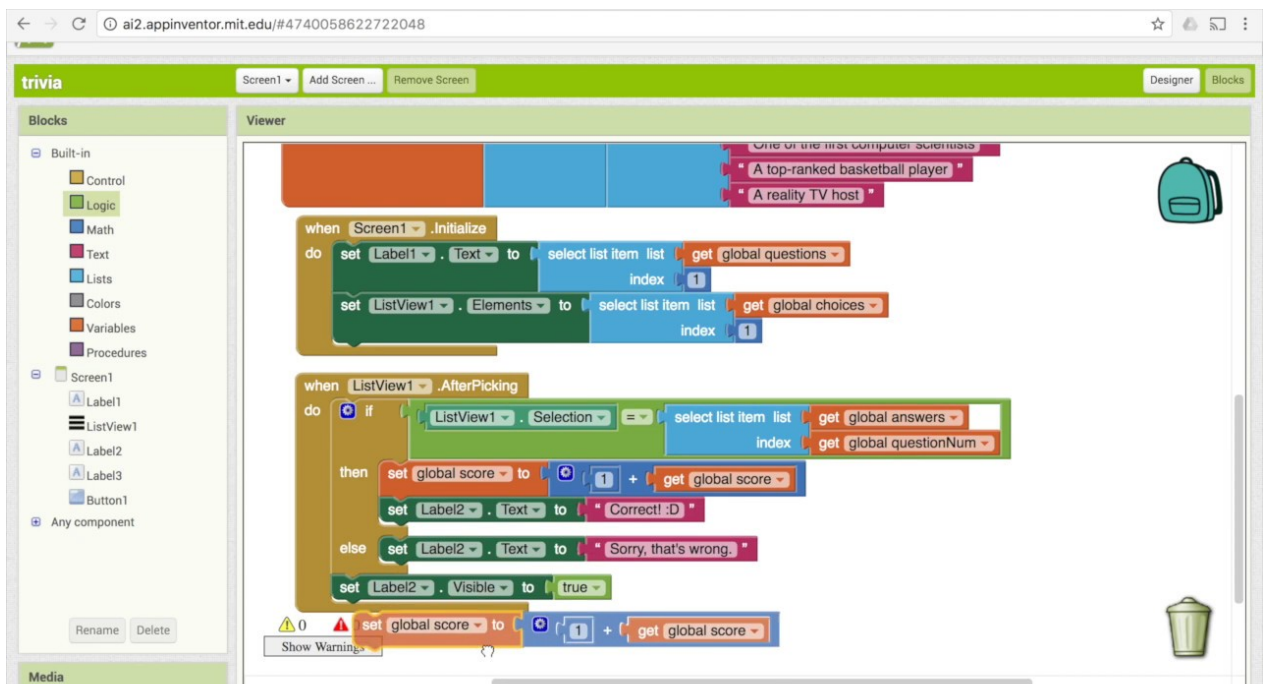
### 4.2.4 Περιβάλλον MIT App Inventor

Το περιβάλλον του MIT App Inventor χωρίζεται σε δυο τμήματα. Το πρώτο είναι το τμήμα design της εφαρμογής και αφορά οτιδήποτε έχει να κάνει με αυτά που φαίνονται στην οθόνη κατά το άνοιγμα της. Κουμπιά, checkboxes, labels, texts, χρώματα και πολλά αλλά είναι σε αυτό το τμήμα και μετακινούνται όπως θέλει ο χρήστης.



Εικόνα 8. MIT App Inventor

Στο δεύτερο τμήμα είναι τα blocks. Εκεί ο χρήστης μπορεί να δηλώσει τι ακριβώς θα κάνει το κάθε στοιχείο στο τμήμα του design. Το κάθε κουμπί, label, checkbox



Εικόνα 9. Τυχαίο παραδειγμα

προγραμματίζεται από αυτό το τμήμα για να κάνει αυτό που θέλει ο χρήστης.

## Κεφάλαιο 5<sup>ο</sup>: Χρήση λογισμικών για την δημιουργία του επιθυμητού αποτελέσματος

### 5.1 Χρήση MIT App Inventor

Το MIT App Inventor χρησιμοποιήθηκε για να μπορέσουμε να δημιουργήσουμε την εφαρμογή android για τον τηλεχειρισμό του ρομποτικού βραχίονα. Παρακάτω φαίνεται ο κώδικας με τα blocks τον οποίο φτιάξαμε και θα τον δούμε αναλυτικά κομμάτι κομμάτι εξηγώντας και την λειτουργία του κάθε κομματιού.

```
when any ListPicker.BeforePicking
  component notAlreadyHandled
do
  set ListPicker.Elements
    of component get component
    to BluetoothClient1.AddressesAndNames
```

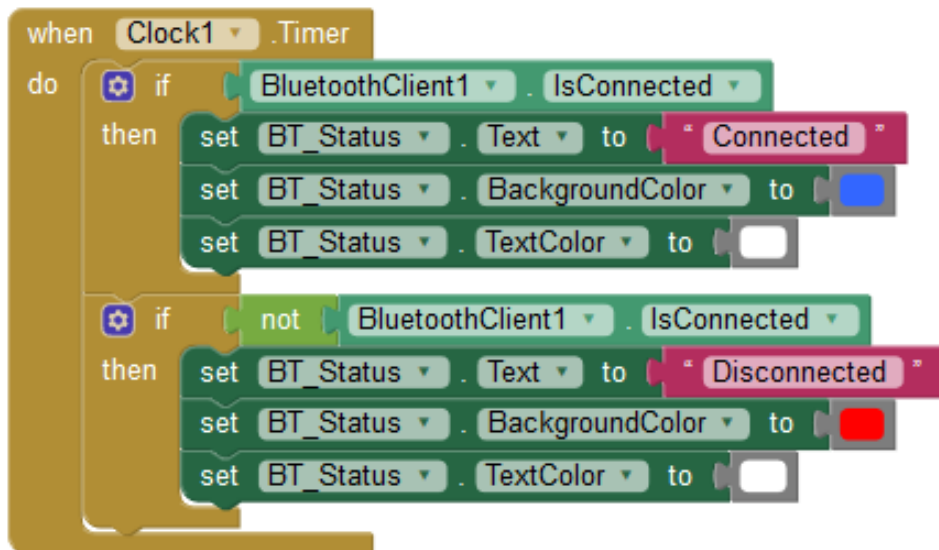
Κομμάτι 1

Σε αυτό το πρώτο κομμάτι ο κώδικας δημιουργεί μια λίστα η οποία περιέχει όλες τις διαθέσιμες συσκευές Bluetooth και ζητάει στην ουσία από τον χρήστη να επιλέξει μια για να γίνει η σύνδεση.

```
when BluetoothList.AfterPicking
do
  if
    call BluetoothClient1.Connect
      address BluetoothList.Selection
  then
    set BluetoothList.Elements to BluetoothClient1.AddressesAndNames
```

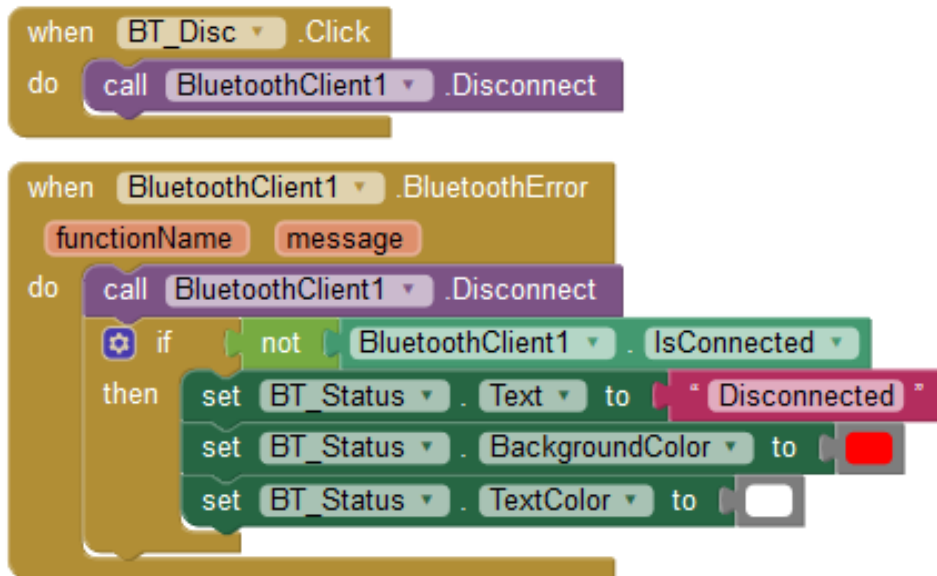
Κομμάτι 2

Στο δεύτερο κομμάτι ο χρήστης επιλέγει την συσκευή Bluetooth που θέλει και ο κώδικας κάνει την σύνδεση για την μεταφορά δεδομένων με την συγκεκριμένη συσκευή.



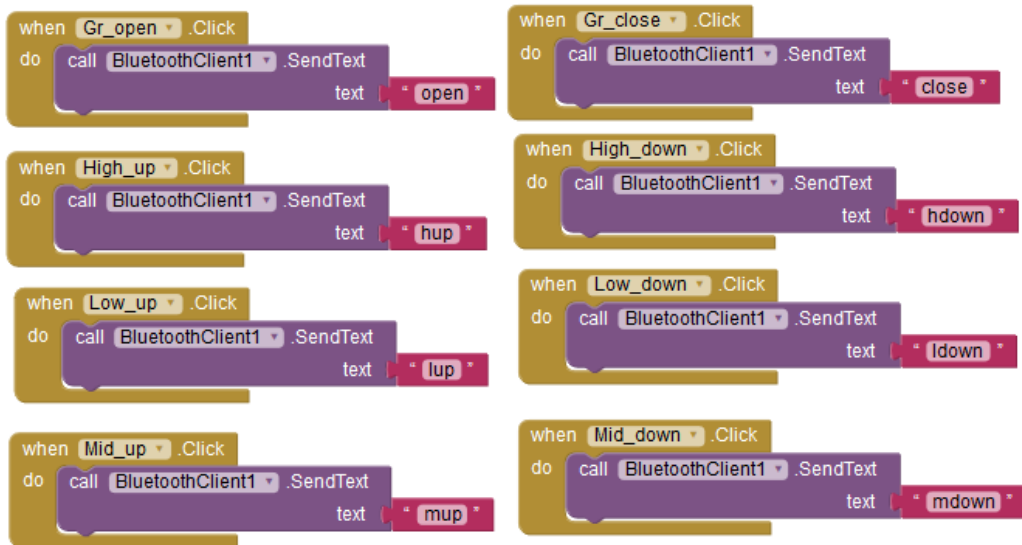
Κομμάτι 3

Στο τρίτο κομμάτι ο εκπομπό/δέκτης στέλνει σήμα πίσω στον κώδικα της εφαρμογής αν είναι όλα καλά με την σύνδεση. Αν είναι όλα εντάξει και μπορεί να γίνει ανταλλαγή δεδομένων τότε το background του label box που μας δείχνει την κατάσταση της σύνδεσης γίνεται μπλε και το κείμενο μέσα αλλάζει σε “Connected”. Σε αντίθετη περίπτωση που δεν έχει γίνει επιτυχώς η σύνδεση το background γίνεται κόκκινο και το κείμενο “Disconnected”.



Κομμάτι 4

Στο τέταρτο κομμάτι γίνονται δυο παρόμοιες διαδικασίες. Αν πατηθεί το κουμπί που ονομάσαμε “BT\_Disc” και στην ουσία είναι το κουμπί αποσύνδεσης των δυο μερών τότε ο κώδικας αποσυνδέεται. Στο κάτω μέρος είναι μια δικλείδα ασφαλείας έτσι ώστε αν γίνει το οποιοδήποτε λάθος στην σύνδεση και είτε δεν γίνει σωστά είτε δεν γίνει καθόλου τότε πάλι αλλάζει το κείμενο και το background.



Κομμάτι 5

Στο πέμπτο και τελευταίο κομμάτι γίνεται η αποστολή σημάτων για την κίνηση του βραχίονα. Όπως βλέπουμε και στο σχήμα με κάθε κλικ του αντίστοιχου κουμπιού στο περιβάλλον της εφαρμογής ο κώδικας στέλνει και ένα διαφορετικό κάθε φορά μήνυμα. Ο κώδικας του Arduino το βλέπει, το διαβάζει και ανάλογα με το περιεχόμενο του κειμένου στέλνει σήμα και στο αντίστοιχο ολοκληρωμένο L293D στο οποίο και οφείλεται η κίνηση του βραχίονα.

## 5.2 Χρήση Arduino IDE

Στον παρακάτω κώδικα φαίνεται η διαδικασία που ακολουθείται στην πλακέτα του Arduino από την αρχικοποίηση των δεδομένων που θα χρειαστούμε μέχρι το τέλος των εντολών όπου στην ουσία δεν κλείνει το πρόγραμμα και απλά περιμένει να δεχτεί δεδομένα.

```
1. #include<SoftwareSerial.h>
2. /* Create object named bt of the class SoftwareSerial */
3. SoftwareSerial bt(2, 3); /* (Rx,Tx) */
4. const int motorPin1 = 6; // Middle
5. const int motorPin2 = 5; // Middle
6. const int motorPin3 = 7; // Low
7. const int motorPin4 = 8; // Low
8. const int motorPin5 = 9; // Grip
9. const int motorPin6 = 10; // Grip
10. const int motorPin7 = 11; // High
11. const int motorPin8 = 12; // High
12. int data;
13. void setup() {
14.     bt.begin(9600); //Define baud rate for communication
15.     Serial.begin(9600); // Define baud rate for communication
16.     pinMode(motorPin1, OUTPUT);
17.     pinMode(motorPin2, OUTPUT);
18.     pinMode(motorPin3, OUTPUT);
19.     pinMode(motorPin4, OUTPUT);
20.     pinMode(motorPin5, OUTPUT);
21.     pinMode(motorPin6, OUTPUT);
22.     pinMode(motorPin7, OUTPUT);
23.     pinMode(motorPin8, OUTPUT);
24. }
25. void loop() {
```

```

26.  if (bt.available()) // If data is available on serial port
27.  {
28.      data = bt.read();
29.      Serial.println(data);
30.      if (data == 103) {
31.          Serial.print("OK");
32.          digitalWrite(motorPin1, HIGH);
33.          digitalWrite(motorPin2, LOW);
34.          delay (100);
35.          data = 0;
36.      }
37.      if (data == 0) {
38.          digitalWrite(motorPin1, LOW);
39.          digitalWrite(motorPin2, LOW);
40.          //Serial.println("\n");
41.          //Serial.println(data);
42.      }
43.
44.  } /* Print character received on to the serial monitor */
45.
46.  if (data == 104) {
47.      Serial.print("OK");
48.      digitalWrite(motorPin1, LOW);
49.      digitalWrite(motorPin2, HIGH);
50.      delay (100);
51.      data = 0;
52.  }
53.  if (data == 0) {
54.      digitalWrite(motorPin1, LOW);

```



```

55.     digitalWrite(motorPin2, LOW);
56.     //Serial.println("\n");
57.     //Serial.println(data);
58. }
59.
60.  /* Print character received on to the serial monitor */
61.  // _____
62.  if (data == 101) {
63.      Serial.print("OK");
64.      digitalWrite(motorPin3, LOW);
65.      digitalWrite(motorPin4, HIGH);
66.      delay (100);
67.      data = 0;
68.  }
69.  if (data == 0) {
70.      digitalWrite(motorPin3, LOW);
71.      digitalWrite(motorPin4, LOW);
72.      //Serial.println("\n");
73.      //Serial.println(data);
74.  }
75.
76.  if (data == 102) {
77.      Serial.print("OK");
78.      digitalWrite(motorPin3, HIGH);
79.      digitalWrite(motorPin4, LOW);
80.      delay (100);
81.      data = 0;
82.  }
83.  if (data == 0) {

```

```
84.     digitalWrite(motorPin3, LOW);
85.     digitalWrite(motorPin4, LOW);
86.     //Serial.println("\n");
87.     //Serial.println(data);
88. }
89. // _____
90. if (data == 99) {
91.     Serial.print("OK");
92.     digitalWrite(motorPin5, HIGH);
93.     digitalWrite(motorPin6, LOW);
94.     delay (100);
95.     data = 0;
96. }
97. if (data == 0) {
98.     digitalWrite(motorPin5, LOW);
99.     digitalWrite(motorPin6, LOW);
100.    //Serial.println("\n");
101.    //Serial.println(data);
102. }
103.
104. /* Print character received on to the serial monitor */
105.
106. if (data == 100) {
107.     Serial.print("OK");
108.     digitalWrite(motorPin5, LOW);
109.     digitalWrite(motorPin6, HIGH);
110.     delay (100);
111.     data = 0;
112. }
```

```

113.   if (data == 0) {
114.       digitalWrite(motorPin5, LOW);
115.       digitalWrite(motorPin6, LOW);
116.       //Serial.println("\n");
117.       //Serial.println(data);
118.   }
119.
120.   /* Print character received on to the serial monitor */
121.   // _____
122.   if (data == 97) {
123.       Serial.print("OK");
124.       digitalWrite(motorPin7, HIGH);
125.       digitalWrite(motorPin8, LOW);
126.       delay (100);
127.       data = 0;
128.   }
129.   if (data == 0) {
130.       digitalWrite(motorPin7, LOW);
131.       digitalWrite(motorPin8, LOW);
132.       //Serial.println("\n");
133.       //Serial.println(data);
134.   }
135.
136.   if (data == 98) {
137.       Serial.print("OK");
138.       digitalWrite(motorPin7, LOW);
139.       digitalWrite(motorPin8, HIGH);
140.       delay (100);
141.       data = 0;

```

```
142. }
143. if (data == 0) {
144.     digitalWrite(motorPin7, LOW);
145.     digitalWrite(motorPin8, LOW);
146.     //Serial.println("\n");
147.     //Serial.println(data);
148. }
149. // _____
150. }
```

### 5.2.1 Αναλυση κώδικα Arduino

Εδώ αναλύσουμε τον κώδικα κομματι-κομματι για να δούμε τι κάνει με τα δεδομένα που δέχεται και πως μετατρέπει σε κίνηση το κλικ από την εφαρμογή. Ο κώδικας είναι χωρισμένος στην ουσία σε 7 τμήματα εκ των οποίων τα 4 τελευταία κάνουν την ίδια δουλειά οπότε δεν χρειάζεται να αναλύσουμε και τα 4. Θα δούμε το ένα και κατά τον ίδιο τρόπο λειτουργούν και τα άλλα 3. Πάμε να δούμε το πρώτο κομμάτι το οποίο είναι το παρακάτω:

```
1. #include<SoftwareSerial.h>
2. /* Create object named bt of the class SoftwareSerial */
3. SoftwareSerial bt(2, 3); /* (Rx,Tx) */
4. const int motorPin1 = 6; // Middle
5. const int motorPin2 = 5; // Middle
6. const int motorPin3 = 7; // Low
7. const int motorPin4 = 8; // Low
8. const int motorPin5 = 9; // Grip
9. const int motorPin6 = 10; // Grip
10. const int motorPin7 = 11; // High
11. const int motorPin8 = 12; // High
12. int data;
```

Αυτό είναι το τμήμα στο οποίο δηλώνουμε τις μεταβλητές. Στις γραμμές 4-11 δηλώνουμε τα motorPins τα οποία δηλώνονται ανάλογα με το pin πάνω στο Arduino στο οποίο συνδέεται το ένα από τα δυο καλώδια της άρθρωσης. Δηλαδή όπως παρατηρούμε και στα σχόλια του κώδικα στο κάθε ζευγάρι motorPins αντιστοιχεί μια άρθρωση για την οποία το ζευγάρι είναι υπεύθυνο. Στην γραμμή 3 δηλώνουμε σε ποια pins του Arduino είναι συνδεδεμένο το Bluetooth module HC-05 και τέλος στην γραμμή 12 δηλώνουμε μια μεταβλητή data η οποία δέχεται τα δεδομένα από την εφαρμογή και δίνει κατάλληλο σήμα στους κινητήρες.

```
13. void setup() {
14.     bt.begin(9600); //Define baud rate for communication
15.     Serial.begin(9600); // Define baud rate for communication
```

```
16. pinMode(motorPin1, OUTPUT);
17. pinMode(motorPin2, OUTPUT);
18. pinMode(motorPin3, OUTPUT);
19. pinMode(motorPin4, OUTPUT);
20. pinMode(motorPin5, OUTPUT);
21. pinMode(motorPin6, OUTPUT);
22. pinMode(motorPin7, OUTPUT);
23. pinMode(motorPin8, OUTPUT);
24. }
```

Εδώ έχουμε το τμήμα που δηλώνουμε το `pinMode` δηλαδή τι περιμένουμε να κάνει το κάθε `pin`. Τα έχουμε δηλώσει όλα ως `OUTPUT` γιατί δεν δεχόμαστε κάτι από τα `pins` του Arduino διότι όλα τα εισερχόμενα δεδομένα γίνονται από το bluetooth το οποίο ενεργοποιούμε στην γραμμή 14 και του δηλώνουμε το `baud rate` στο οποίο θα συγχρονιστεί με το HC-05 module (9600 ήταν το προεπιλεγμένο το οποίο δούλεψε απροβλημάτιστα από την πρώτη στιγμή οπότε δεν χρειάστηκε να το τροποποιήσουμε).

```
25. void loop() {
26.   if (bt.available()) // If data is available on serial port
27.   {
28.     data = bt.read();
29.     Serial.println(data);
```

Σε αυτό το μικρό τμήμα ξεκινάει η επανάληψη στην οποία ο κώδικας θα περιμένει διαρκώς δεδομένα από την εφαρμογή και αν υπάρχουν (γρ. 26) τότε συνεχίζει και διαβάζει τα δεδομένα από την εφαρμογή και τα βάζει στην μεταβλητή `data` και τα εκτυπώνει στην σειριακή οθόνη (η εκτύπωση έγινε για να μπορούμε να βλέπουμε τι δεδομένα δέχεται ο κώδικας και να κινήσουμε τον ανάλογο κινητήρα).

```

30.     if (data == 103) {
31.         Serial.print("OK");
32.         digitalWrite(motorPin1, HIGH);
33.         digitalWrite(motorPin2, LOW);
34.         delay (100);
35.         data = 0;
36.     }
37.     if (data == 0) {
38.         digitalWrite(motorPin1, LOW);
39.         digitalWrite(motorPin2, LOW);
40.         //Serial.println("\n");
41.         //Serial.println(data);
42.     }
43.
44. } /* Print character received on to the serial monitor */
45.
46. if (data == 104) {
47.     Serial.print("OK");
48.     digitalWrite(motorPin1, LOW);
49.     digitalWrite(motorPin2, HIGH);
50.     delay (100);
51.     data = 0;
52. }
53. if (data == 0) {
54.     digitalWrite(motorPin1, LOW);
55.     digitalWrite(motorPin2, LOW);
56.     //Serial.println("\n");
57.     //Serial.println(data);
58. }

```

```
59.  
60.  /* Print character received on to the serial monitor */  
61.  //
```

Στο τέταρτο κομμάτι το οποίο είναι ίδιο με τα υπόλοιπα 3 έχουμε την κίνηση των κινητήρων την οποία θα αναλύσουμε. Αρχικά εξετάζουμε αν τα δεδομένα είναι μια συγκεκριμένη τιμή (103 είναι η τιμή για να κινηθεί η μεσαία άρθρωση επάνω) η οποία είναι και η τιμή που στέλνει η εφαρμογή κατά το πάτημα μιας επιλογής. Αν η τιμή αυτή είναι εκείνη που πρέπει τότε εκτυπώνει στην σειριακή οθόνη ένα μήνυμα επιτυχίας «OK» και κατόπιν στέλνει ρεύμα στο ένα καλώδιο της άρθρωσης με αποτέλεσμα να κινηθεί εκείνη (πάνω στο συγκεκριμένο παράδειγμα). Βάζουμε στη συνέχεια ένα delay (100) ώστε να περάσει ένα ms με την κίνηση να είναι ενεργή και με την εντολή data = 0. Έτσι όταν ελέγξει στην σειρά 37 αν η τιμή της μεταβλητής data είναι 0 θα γυρίσει TRUE και θα μπει να συνεχίσει τις εντολές. Αυτές είναι στην ουσία για να μην στείλουμε ρεύμα σε κανένα από τα καλώδια και αρά να σταματήσουμε την κίνηση. Στην σειρά 46-58 γίνεται η ίδια δουλειά για την ίδια άρθρωση αλλά για διαφορετική εντολή από την εφαρμογή. Η τιμή αλλάζει (104 είναι η τιμή για να κινηθεί η μεσαία άρθρωση κάτω) και κατά τον ίδιο τρόπο αλλάζουμε το ρεύμα στο κάθε καλώδιο και έτσι αντιστρέφεται η κίνηση.

Στην σειρά 61 έχουμε βάλει σαν σχόλια μια γραμμή η οποία μας βοήθησε να καταλαβαίνουμε που κλείνει το κάθε ζευγάρι κινήσεων (πάνω-κάτω) για κάθε άρθρωση.



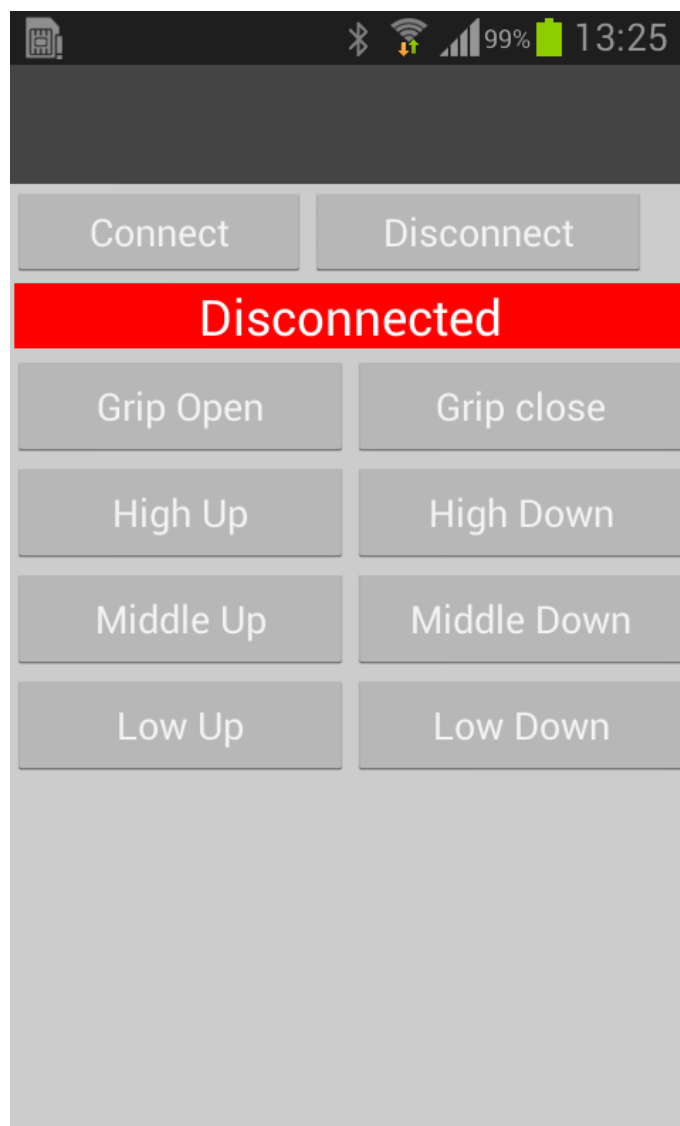
## Κεφάλαιο 6<sup>ο</sup>: Επεξήγηση διαδικασίας εργασίας

### 6.1 Περιγραφή

Σε αυτό το κομμάτι θα δούμε όλη την διαδικασία της σύνδεσης της εφαρμογής με το ρομποτικό βραχίονα καθώς και το user interface (UI) της εφαρμογής. Η διαδικασία είναι αρκετά απλή οπότε δεν θα έχει πρόβλημα κανείς για να συνδεθεί και να ελέγξει την κίνηση του βραχίονα.

### 6.2 Διαδικασία χρήσης της εφαρμογής

Η αρχική οθόνη της εφαρμογής είναι αυτή πριν κάνει ο χρήστης το οτιδήποτε. Εδώ η εφαρμογή περιμένει να πατήσει ο χρήστης connect για να ξεκινήσει να ψάχνει και να φέρει μια λίστα με όλα τα διαθέσιμα bluetooth και να επιλέξει ο χρήστης αυτό που θέλει.



Εικόνα 10. Αρχική οθονη

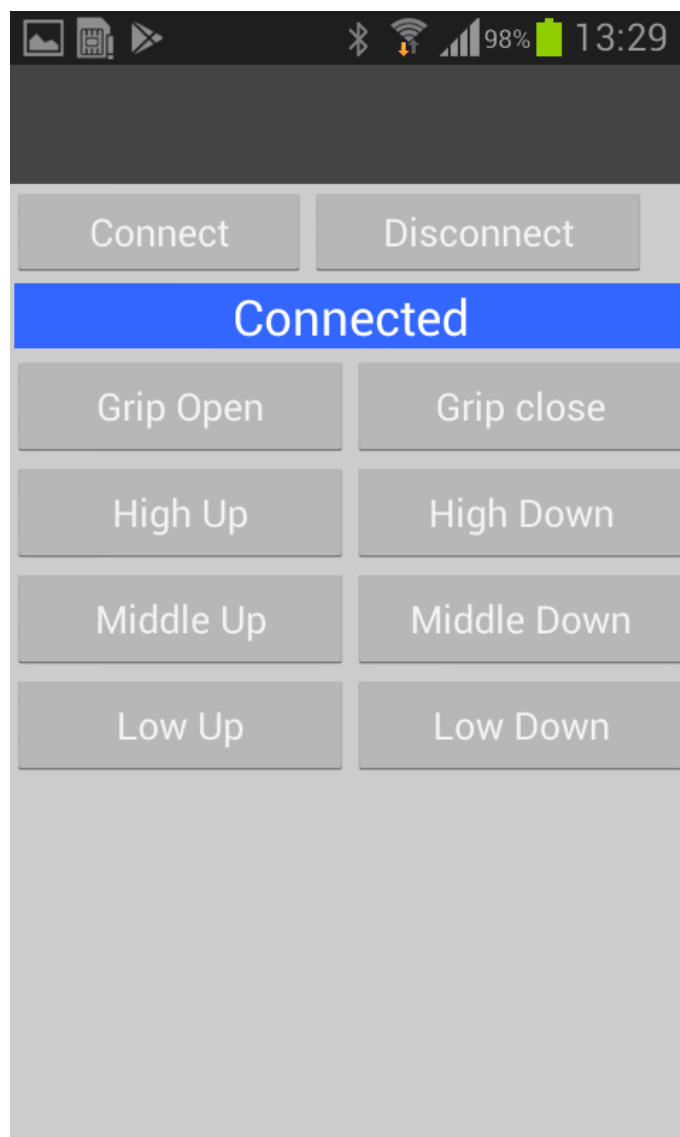
Στην επόμενη οθόνη και αφού πατήσουμε connect μας δείχνει την παρακάτω οθόνη η οποία είναι η λίστα με όλα τα διαθέσιμα bluetooth που έχει βρει η συσκευή. Επιλέγουμε το μοναδικό διαθέσιμο στην προκειμένη περίπτωση με χαρακτηριστικό το HC-05 στο τέλος του ονόματος.

(Την πρώτη φορά που θα συνδεθούμε με το bluetooth module HC-05 θα μας ζητήσει τον default κωδικό ο οποίος στην περίπτωση μας ήταν 1234).



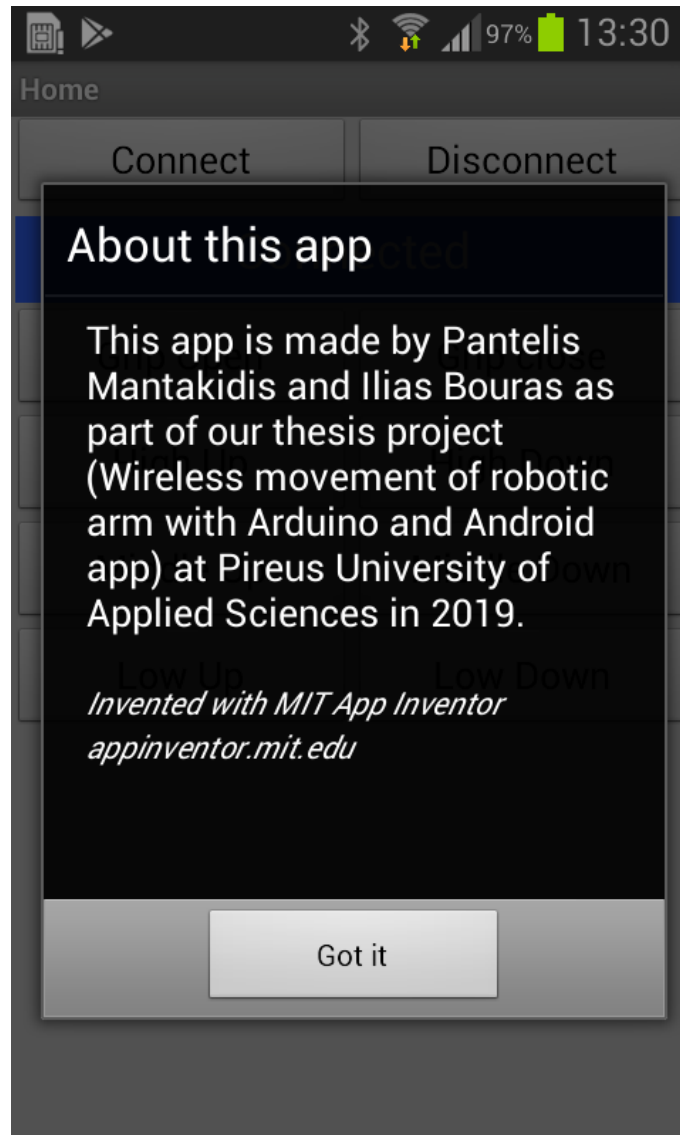
*Εικόνα 11. Οθονη μεσα το πατημα του connect*

Παρακάτω έχουμε την οθόνη που θα εμφανιστεί όταν συνδεθούμε επιτυχώς με το ρομποτικό βραχίονα με ένδειξη connected και το μπλε χρώμα. Από εκεί είμαστε έτοιμοι να δώσουμε οποιαδήποτε από τις εντολές για να δώσουμε κίνηση στον βραχίονα. Φυσικά με την κάθε εντολή κινείται ο αντίστοιχος κινητήρας με όνομα όπως στην φωτογραφία (βλ. σελ. 12).



Εικόνα 12.Οθονη αφου συνδεθει

Στην επιλογή About βρίσκουμε την περιγραφή των δημιουργών της εφαρμογής.



Εικόνα 13. About page

*This page is left blank intentionally*

## Βιβλιογραφία

Πέρα απ' τους παρακάτω συνδέσμους αξίζει να σημειωθεί ότι χρησιμοποιήθηκαν γνώσεις από τα παρακάτω μαθήματα:

1. Εισαγωγή στον προγραμματισμό
2. Ηλεκτρονικά
3. Λογικά Κυκλώματα
4. Θεωρία Κυκλωμάτων
5. Μικροηλεκτρονική
6. Δομημένος προγραμματισμός
7. Αλγόριθμοι και δομές δεδομένων
8. Αντικειμενοστραφής προγραμματισμός
9. Βάσεις δεδομένων
10. Ψηφιακές Επικοινωνίες
11. Δίκτυα Η/Υ
12. Σ.Α.Μ. - Μικροελεγκτές
13. Ανάπτυξη Διαδικτυακών Εφαρμογών
14. Λειτουργικά Συστήματα
15. Ασφάλεια & Διαχείριση Διαδικτυακών Συστημάτων
16. Μηχατρονικά Συστήματα

## Δικτυογραφία

1. About Us. Owirobot.com, από <http://www.owirobot.com/about-us-1/>
2. 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))
3. Arduino. En.wikipedia.org. Ανακτήθηκε στις 16 Ιουνίου 2017, από τη διεύθυνση <https://en.wikipedia.org/wiki/Arduino>
4. Arduino - BuildProcess. Arduino.cc. Ανακτήθηκε στις 16 Ιουνίου 2017, από τη διεύθυνση <https://www.arduino.cc/en/Hacking/BuildProcess>
5. Arduino - Συχνές ερωτήσεις. Arduino.cc. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://www.arduino.cc/en/Main/FAQ>
6. Εισαγωγή στο Arduino – Το απόλυτο geek toy. (2009). deltaHacker. Retrieved 16 June 2017, from <https://deltahacker.gr/arduino-intro/>
7. 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/>
8. Arduino - Εισαγωγή. Arduino.cc. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://www.arduino.cc/en/guide/introduction>
9. 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/>
10. AVR-GCC. Ccrma.stanford.edu. Ανακτήθηκε στις 16 Ιουνίου 2017, από [https://ccrma.stanford.edu/~juanig/articles/wiriavrlib/AVR\\_GCC.html](https://ccrma.stanford.edu/~juanig/articles/wiriavrlib/AVR_GCC.html)
11. Βασικά στοιχεία Bluetooth - learn.sparkfun.com. Learn.sparkfun.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://learn.sparkfun.com/tutorials/bluetooth-basics>
12. BluetoothAdapter | Προγραμματιστές Android. Developer.android.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://developer.android.com/reference/android/bluetooth/BluetoothAdapter.html>
13. 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/>
14. Evans, B. (2013). Ξεκινώντας τον προγραμματισμό Arduino (σελ. 1,3,5,8). [S.I.]: Απρίλιος.
15. 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/>
16. 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](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)
17. Ιστορικό βιομηχανικών ρομπότ. Robots.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://www.robots.com/education/industrial-robot-history>
18. Πρόθεση | Προγραμματιστές Android. Developer.android.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://developer.android.com/reference/android/content/Intent.html>
19. Τρεις νόμοι της ρομποτικής του Ισαάκ Ασίμοφ. Auburn.edu. Ανακτήθηκε στις 16 Ιουνίου 2017, από <http://www.auburn.edu/~vestmon/robotics.html>

20. Manifest.permission | Προγραμματιστές Android Developer.android.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://developer.android.com/reference/android/Manifest.permission.html>
21. Pololu - OWI-535 Ρομποτικό άκρο Edge Kit. Pololu.com. Ανακτήθηκε στις 16 Ιουνίου 2017, από <https://www.pololu.com/product/947>
22. 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/>