

Internet of Things Project

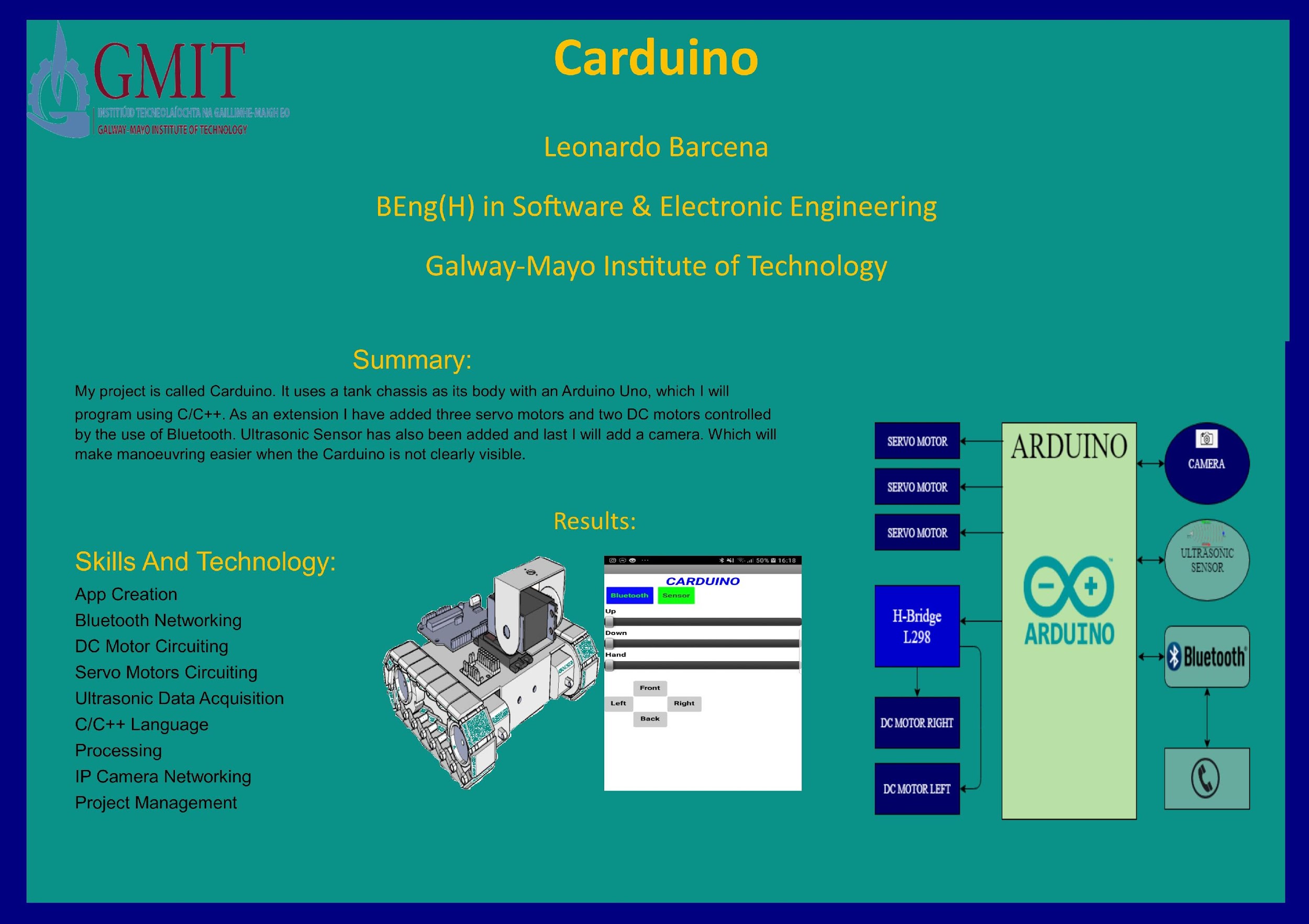
Carduino

Leonardo Barcena

Bachelor of Software & Electronic Engineering

Galway-Mayo Institute of Technology

2018/2019



**Declaration**

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Software & Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgements**

I would like to thank Michelle Lynch and also thank Brian O’Shea for all the help they have provided through the year.

**Table of Contents**

[1](#_gjdgxs) Summary 7

[2](#_30j0zll) Introduction 9

[3](#_1fob9te) Project Architecture 10

[4](#_3znysh7) Development Platform and Tools 11

[4.1](#_2et92p0) MIT APP Inventor 2 11

[5](#_tyjcwt) Motors 14

[5.1](#_3dy6vkm) DC Motors 14

[5.2](#_1t3h5sf) Servo Motors 14

[5.2.1 MG996R Servo Motor 15](#_4d34og8)

[5.2.2 SG90 Micro Servo 16](#_2s8eyo1)

[5.3](#_17dp8vu) Motor Control Code 17

[5.3.1 DC Motors Code 17](#_3rdcrjn)

[5.3.2 Servo Motors Code 18](#_26in1rg)

[6](#_lnxbz9) HC-05 Bluetooth Module 19

[7](#_35nkun2) Problem Solving 20

[8](#_1ksv4uv) Conclusion 22

[9](#_44sinio) References 23

[10](#_2jxsxqh) Code 24

# Summary

* + **The goal of the project.**

The goal of my project is to move my DC motors and my servo motors through Bluetooth using my Android App.

* + **The important features of the project.**
* App
* Servo Motors
* DC Motors
* Bluetooth Module
* C/C++ Language
  + **The approach to the project.**

First step I took was getting sample code for my Servos because I hadn’t used them before, for my Dc motors I had an easier approach because in first year of college I used dc motors to drive a vehicle. My next step was to learn how to use the Bluetooth so I used a pre-made app to get data from it, and my final step was to make my own app and use it to control my project.

* + **The main methods & technologies used.**
* MIT APP Inventor 2
* Bluetooth Networking
* Dc Circuiting
* Servo Motor Circuiting
* C/C++ Language
* Project Management
* Arduino IDE
  + **What was accomplished.**

Coming to the end of the year, I accomplished making the app and moving the Dc motors and Servo motors using Bluetooth.

* + **The main conclusions.**

I have reached the end of my project. I wanted to control my motors using Bluetooth and create an App. I have encountered a lot of issues these past few months but I managed to find a way to fix those issues or even find a way around it. I can say I have had success in the end as all my main goals have been achieved in the end.

# Introduction

In the last couple of years we have noticed an increase in development in the areas of assisted parking and manoeuvring through the use of smart devices such as phones, computer and tablets. That has inspired me to do my project. In first year of college we made a vehicle that had similar features to that so for this project I decided to continue from that. So for my project I made a vehicle that will be controlled through an android phone app. I created the app in MIT App Inventor 2. The app will control two servo motors and two DC motors through a HC-05 Bluetooth module. Which will be linked to an Arduino Uno board. The phone app will send data to the Arduino and that will initialize the movement of the motors.

# Project Architecture

I’m using an Arduino Uno microcontroller board, it contains 14 digital input/output pins, 6 analog inputs, A USB connection, a power jack, an ICSP header and a reset button. To program and code I used the Arduino Integrated Development Environment (IDE). It supports the languages C/C++ and these are the languages I have used to develop this project.



**Figure 3-1 Architecture Block Diagram**

# Development Platform and Tools

## MIT APP Inventor 2

Another development platform I have used is MIT App Inventor. MIT app inventor is an open-

source cloud based web application, it allows you to make your own mobile apps using a block

based programming language, very similar to Scratch. You can access and make changes to your

apps through an internet browser. This here is the app I have created. The first image is the

app and second image is the coding that makes the app function Blue button ‘Select BT Device

is used to search for Bluetooth device, And this is the code that searches for Bluetooth Devices

that are turned on, it will pick up other phones and most Bluetooth devices.

BluetoothCodeApp

These second code blocks are the for the driving the DC motors. So it will send a 1-4 to the arduino board, but those numbers reach the arduino it will be in ASCII numbers

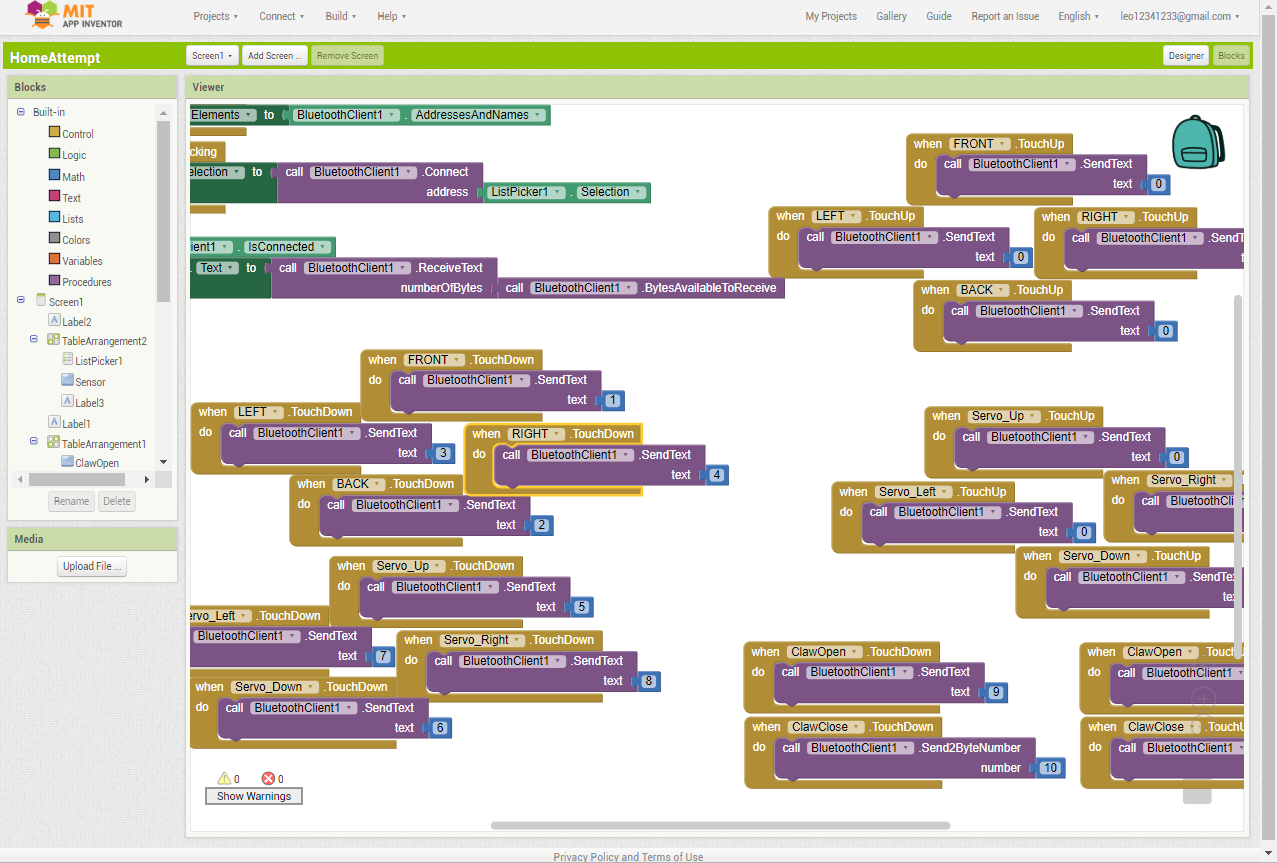
A screenshot of a cell phone

Description automatically generated

The yellow buttons are used to move the vehicle while the red buttons are used to move the servo motors.



This is an overview of the of the whole block code for the app.

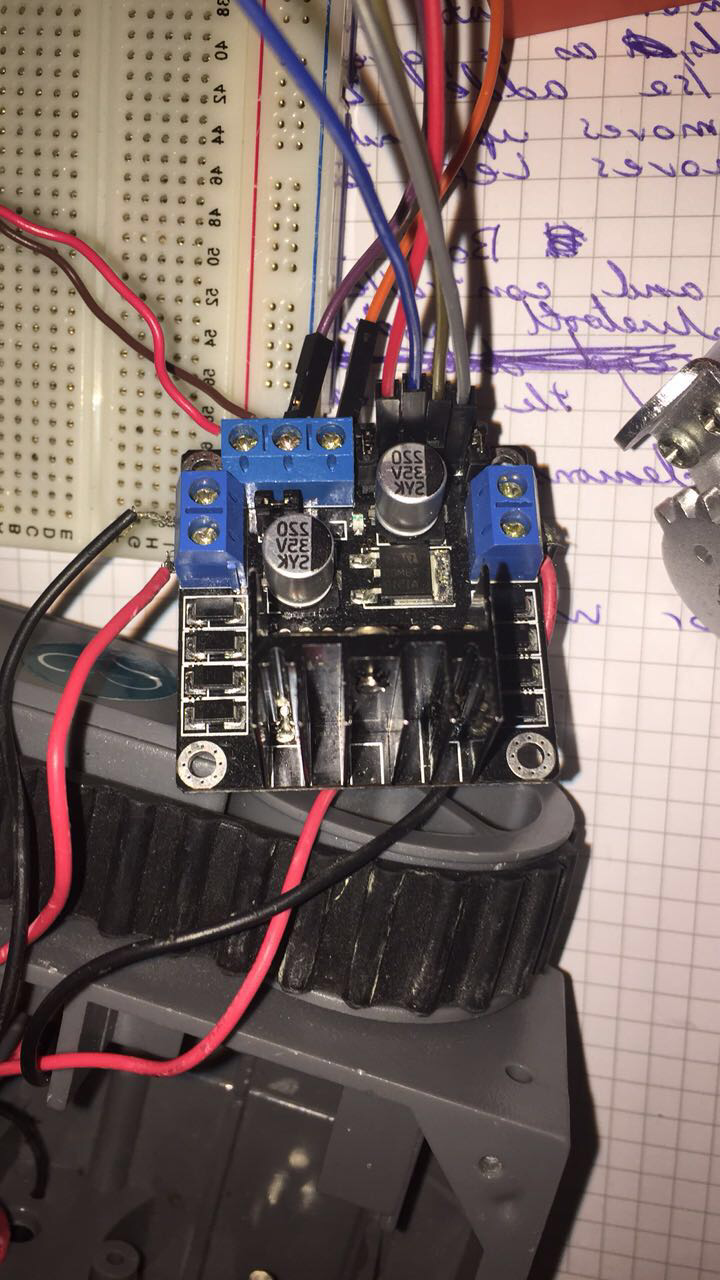
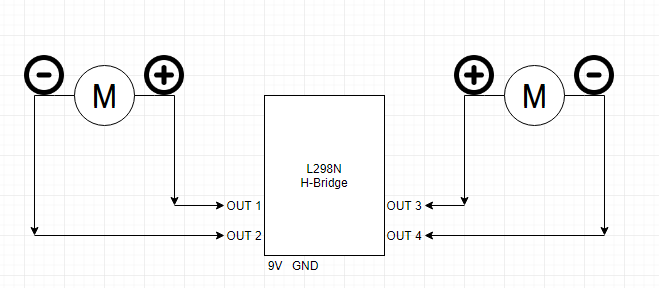


# Motors

My project contains two types of motors Dc and Servo motors.

## DC Motors

Here I have the Dc motors. They have a positive and a negative connection that I have connected to my L298N H-Bridge motor driver. It allows speed and direction control of two DC motors that have voltages between 5 and 35V, with a peak current up to 2A. In my case I’m using a 9V battery to power them.

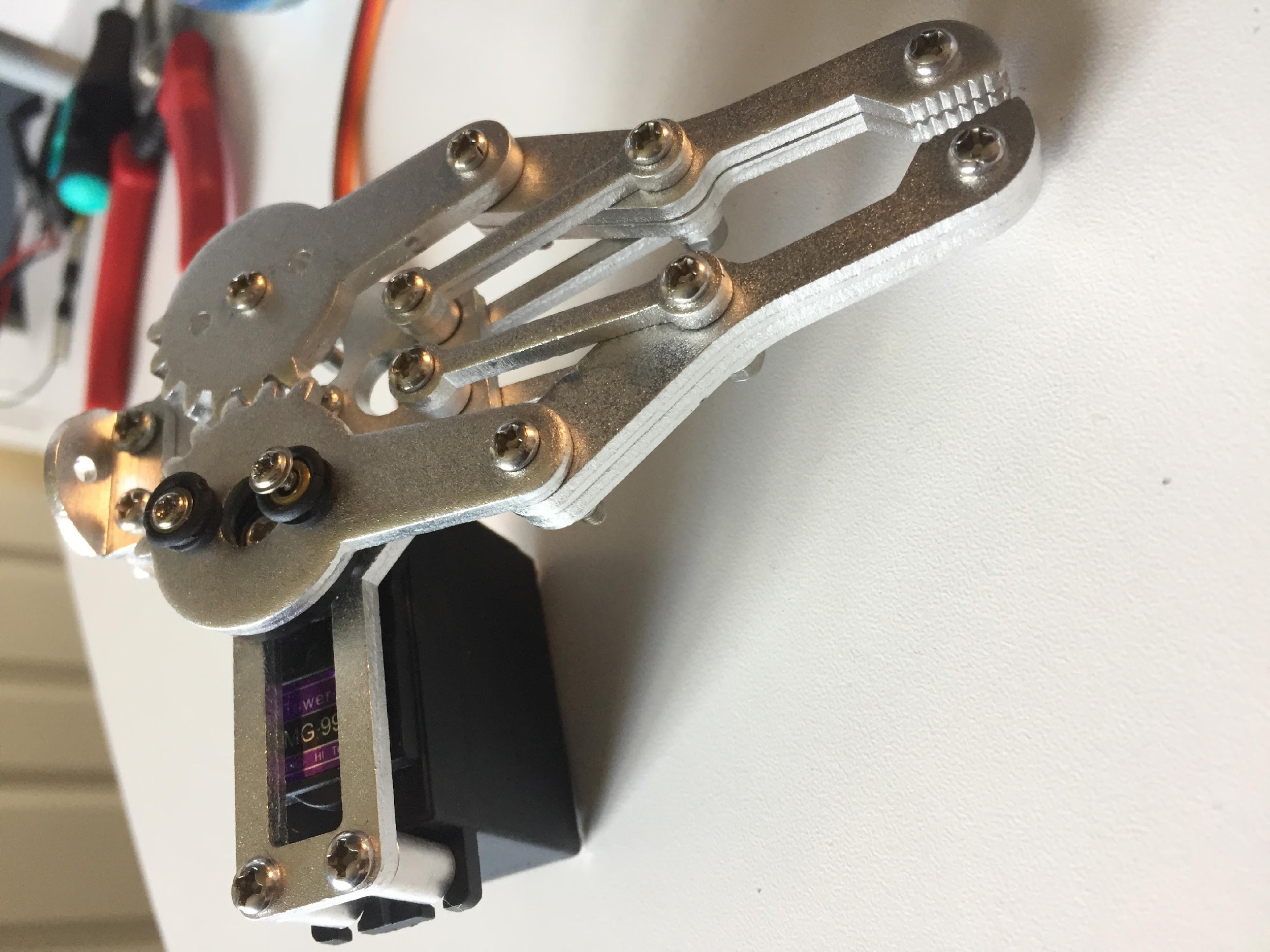
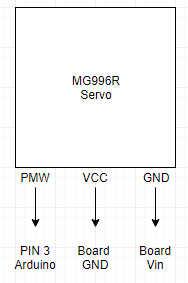
**Figure 5-1** DC motors and L298N Circuit diagram

## Servo Motors

I’m using 3 servo motors on this project. Consists of 2 SG90 Micro Servos and a single MG 996R Servo.

### MG996R Servo Motor

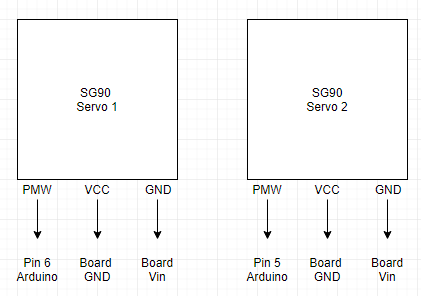
The MG996R Servo motor I’m using as a hand/claw. This high speed standard servo can rotate approximately 120 degrees, 60 in each direction. It uses from 4V up to 7V. The current used in its idle state is 10mA when it operates with no load it uses 170mA and stalls at 1.2A

**Figure 5-2.1** MG996R Servo Motor Circuit Diagram

### SG90 Micro Servo

This servo I will be using to control the camera on top of my project. It is small and very light weight with a good output power. This servo can rotate more than the SG996R servo, it has a range of 180 degrees, 90 in each direction. This servo needs 4.8 – 5V to operate. Its idle state uses 5mA, no load 100mA and stalls at 700mA. I’m using a pan and tilt support so it can control the camera up, down, left and right.

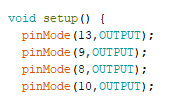
 

**Figure 5-2.2** SG90 Servo Motor Circuit Diagram

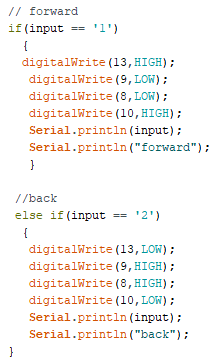
## Motor Control Code

### DC Motors Code

Here I set the pins 13, 9, 8, 10 to be outputs. I need to tell the pins that they are an output so they can either send a low or a high signal to my Dc motors that will be connected to those pins. These pins will then be connected to IN1, IN2, IN3, IN4 of my H-Bridge.



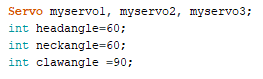
Here I made some if statements, to tell my motors which direction they should go. The input will be decided by the button pressed on my app so here for example whenever the serial monitor reads ‘1’ which will equal to 49 in the serial monitor. It will send 5V to pin 13, 0V to pin 9, 0V to 8 and 5V to 10. The Serial.println(input); will show the value that the serial monitor is reading at the time the if statement is activated and it will also print the word “forward”. So I will know which direction the car should be going. It works exactly the same for the other directions, all you need to do is just change the value going to the pins at the directions desired.



### Servo Motors Code

First thing I done was to look for the servo library which allows the arduino to control the servo motors. .

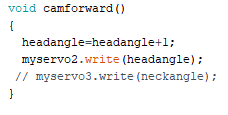
Next step was to name the servos, since I’m using 3 servos I labelled them myservo1-3. I also made 3 integers with values 60, 60 and 80. That will later be used to be the values that I will initialize my servos in. I gave them these value so that the will start in the middle.



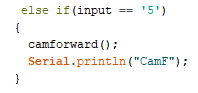
After this I attached the servos PMW pins to the arduino pins. That will allow me to control the angle of the servos.



I made functions to control the value of the servos, so here I have the camforward(); function. This is where the integers I made come in, so everything this function is called it will increase the value of the integer by 1. It will then write the new value in headangle int.



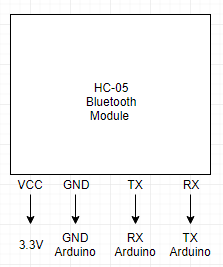
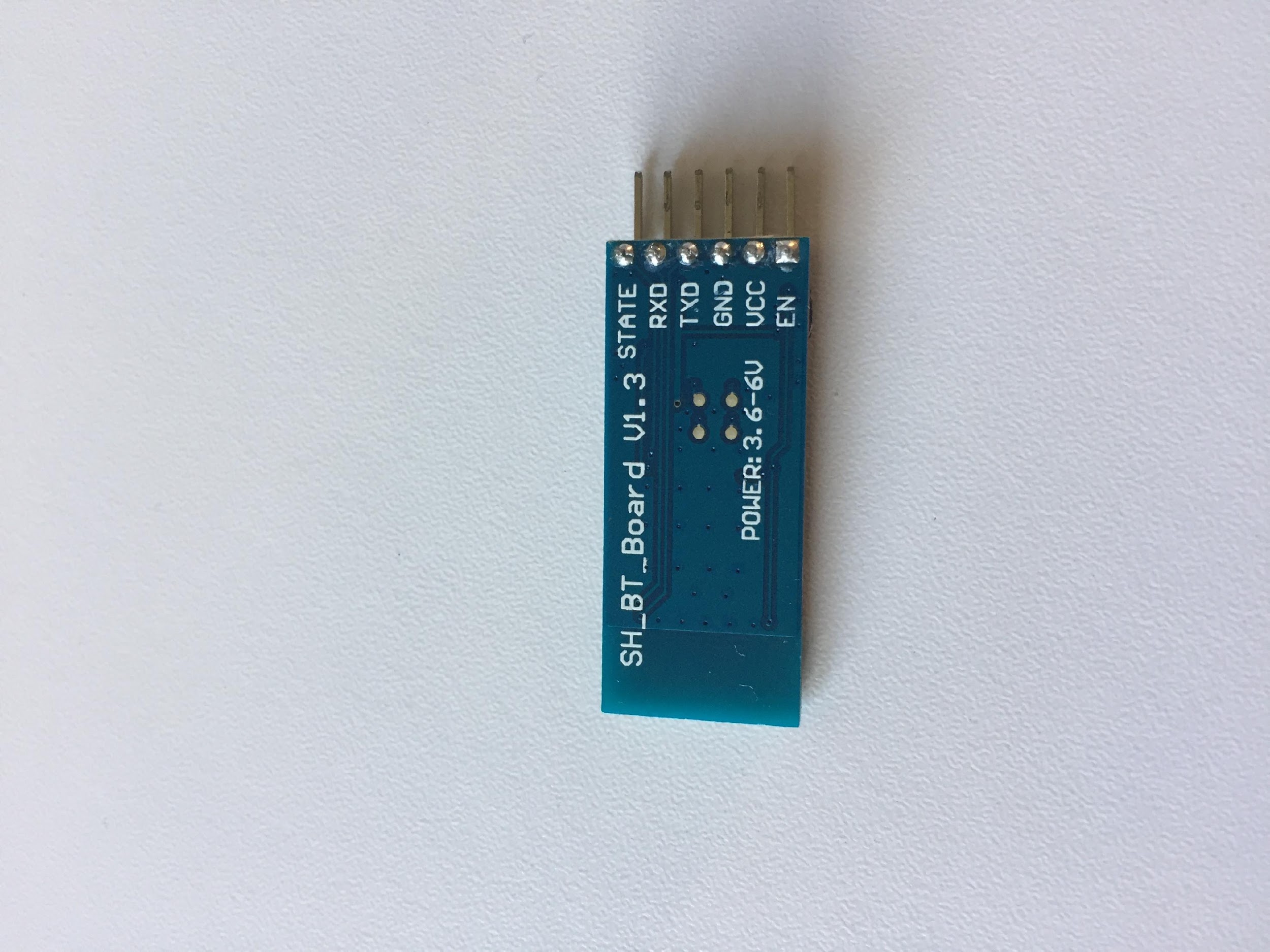
Here im calling the function, exaclty the same was as the Dc motors work but this time instead of being output high or low, it will just call the functions I have. So whenever the serial monitor reads ‘5’ which = 53, it will call the function and it will move the servo motor. I just made more of there just do different movements but I works the same for all the other directions.



# HC-05 Bluetooth Module

So I decided to use a HC-05 bluetooth module to controll the Arduino using a smartphone.

It consists of 4 pins that are in going to be used. Vcc is connected to 3.3V in the Arduino Board, Gnd pin is connected to the second GND pin in the arduino, RX pin connected to TX pin in the arduino board, and the TX pin is connected to the RX arduino pin.

This line of code here is the communication rate of the Bluetooth module.

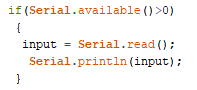
So it allows the Bluetooth to communicate with the serial port of the arduino.



If (Serial.available()>0) will check if the date is coming from the serial port. returns the number of characters which have arrived in the serial buffer and that are ready to be read.

Input = Serial.read(); will read the data that was ready to be read from the serial port. So when all the bytes of data are read and no new serial data have arrived, the buffer is empty and Serial.available() will return 0.

And this will activate the if statements that are used to move the DC motors and Servo Motors.



# Problem Solving

During the making of my project I ended up coming across a few problems, mainly involving the app. One because there isn’t much help with what I needed online, so it was hard for me to figure out exactly what was going wrong with what I needed. The first main issue was to make my servo motors move. The first thing I had done was to make sliders buttons to make my servo motors move. So I made them. Labelled and gave them values 1000-1180, and on the Arduino I mapped them to 0-180, because with the sliders I would be able to be very accurate and move them fast from 0-180 degrees, but after attempting to move one of the servos I found out that they were all reading over each other, so from there I decided to give each of the different values, so in the app I labelled them to be 1000-1180, 2000-2180 and 3000-3180. After that they were working as they should, so my next step was to make the DC motors move. So I created buttons, forward, backwards, left and right. I made so that when button is pressed on the app that it would send values 1-4. So whenever I pressed let’s say Forward which was equal to 1 in the app the serial monitor would receive the value 49. which was the least of my problems at that time. But then when I tried bringing both apps together I would have encountered the biggest problem of my app. My servos which were mapped from 0-180, would write over my DC motors which would turn them on when the servo values reached 49-52. So whenever I dragged my sliders buttons across when it reached 49 it would turn on my DC motors and it would activate one of my functions to move my vehicle. And it took me a good few weeks to figure that out. I tried modifying my app multiple times thinking the issue was there, and that was the reason I was getting 49-52 values, which I did with not much success. But then I used a premade app that was in the app store and it was also sending the same values as my app was sending. So apparently sending a ‘1’ to the serial monitor is a character 1 and not a number one. That is when I decided I had to put printfs all over my code to see what was being read in the serial monitor. So then I figured out that the Bluetooth module is the reason that integers are read in ASCII So to fix that problem I decided to make buttons to move my dc motors with a fixed value and not a range of numbers, which ended up working in the end. This time I labelled my Servos from 5-8. So the servos values are still 0-180 in the end but every time I press one of the servos buttons it will either increase the value of the servo by 1 or decrease it by 1.

# Conclusion

Having implemented most of my ideas to this project. I am very happy with the outcome. After making a working android App, controlling 2 DC motors and 3 Servo motors through Bluetooth by sending data from a phone to the arduino. Just to finish off, for future plans I would like to add a working camera to facilitate the manoeuvring of my vehicle.

# References

Some example references are given here. For example, the section in your report on your servo motors should have a sentence that includes [1] in it, referring to reference [1] here.

[1] Arduino, “Servo library”, arduino.cc, 2018.

[2] Arduino, “Serial.begin();”, arduino.cc, 2018.

[3] Stack Overflow, “Bluetooth ASCII protocol ”, stackoverflow.com, 2012.

[4] MIT App Inventor, “Tutorials for App Inventor”, appinventor.mit.edu, 2012.

# Code

