Project Report

Sumo Robot Using Arduino

Ahmad Hashi 1046819

Omar Altawil 1047982

Mohamed Alnuaimi 1059177

Mohammed Tareq 1059223

Supervised By: Prof.Mohammad Ghazal
Eng. Yasmina Al Khalil

جامعة أبوظبي
ABU DHABI UNIVERSITY

Submitted: December 15, 2017

Contents

List of Figures

List of Tables

Abstract

This project is about creating you're own Sumo Robot, designing and coding were up to us in the process of making our Robot.The winner of the Sumo Robot fight competition will be considered the best Robot design.The process of making our robot included making a PCB design, a 3D Design and a Design using some plastic and laser cutting. We faced a lot of challenges through out our project mainly on the chassis part. The coding was very fun to do, and we feel like we learned a lot about the embedded systems thanks to this project.

1 Introduction

Our project is about designing, assembling and programming our own Sumo Robot. We had a budget of 100 US Dollar and were not allowed to exceed it.At the end of the semester, there were a competition to see which Sumo Robot is better and we will hopefully win the competition.

2 Procedure

Here, we are going to talk about everything that we did, including the difficulties that we faced, and we will end up with the FSM and the Final code.

2.1 Motors

In our robot, we had four 12 V DC motors which were supplied using a 12 volt(1.2AH)battery. some of the stuff didn't make it to the final design due to reasons that are going to be mentioned and explained in the rest of the project report.

- Motor Shield : it's driver module for motors that allows the user to control the state of the motor(on/off), the speed and the direction of the motor [1].We didn't use the motor shield due to having insufficient pins for the rest of the sensors. A picture of the motor shield is shown in the figure 1.

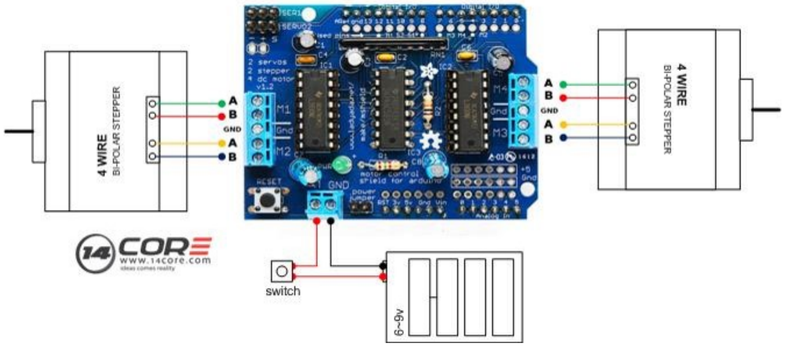

Figure 1: The Motor Shield

- L293d (H-Bridge) : To fix the problem that we faced with the Motor Shield, we decided to use L293d. We solider-out the l293d (2) from the motor shield, and we used to control the four motors. This was our ideal method of controlling the motors, the L293d had a very convenient size and it would have done its job fair and square.However, due to facing problems with the 3D design and the PCB print,we had to do the soldering by ourselves,but the connections were too big and it didn't fit into our chassis.The L293d couldn't make it to the final design.A picture of the L293d is shown in the figure 2.

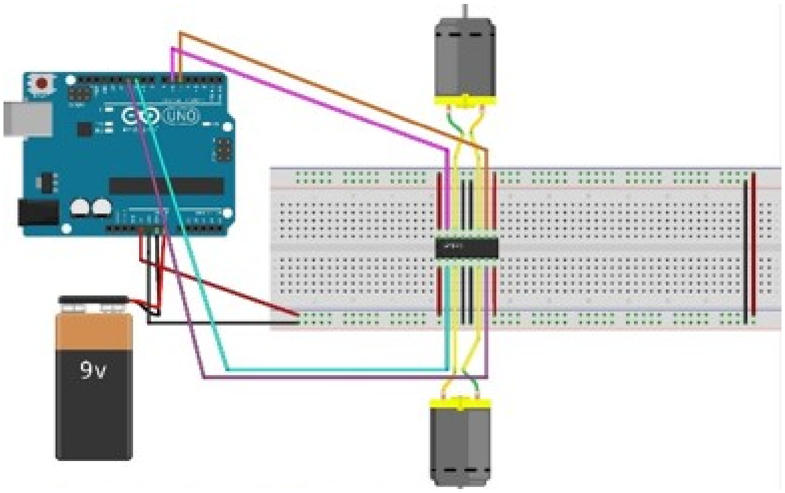

Figure 2: The H-Bridge

- L9110S : After having problems with the H-Bridge, we decided to use the L9110S motor driver. The L9110S was small, cheap and would fit perfectly in our chassis, and according to its data sheet it said that it can handle an range from 3.5 to 12 volt which was exactly what we wanted. Yet again, we faced an other problem, when we supplied the L9110S with 12v it worked approximately for 30 seconds before burning up due to overheat. Thus,the L9110S couldn't make it to the final design.A picture of the L9110S is shown in the figure 3.

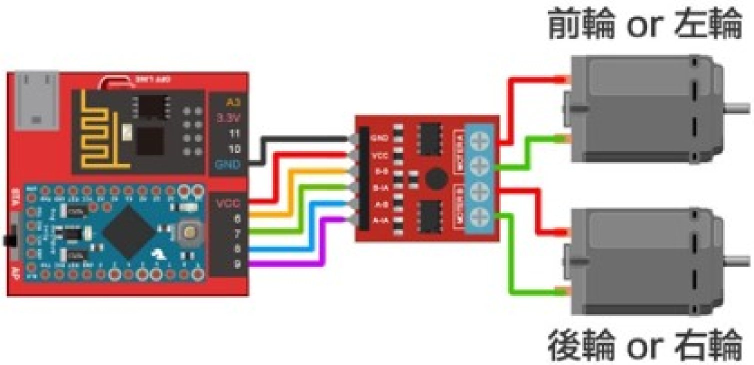

Figure 3: The L9110S

- L298n : we got this motor driver as a final solution, although the size of this motor driver was not what we intended to have in the beginning, it was the only thing that would work without overheating nor taking extra pins and its connections were acceptable. Therefor,the L298n was included into our final design.A picture of the L298n is shown in the figure 4.

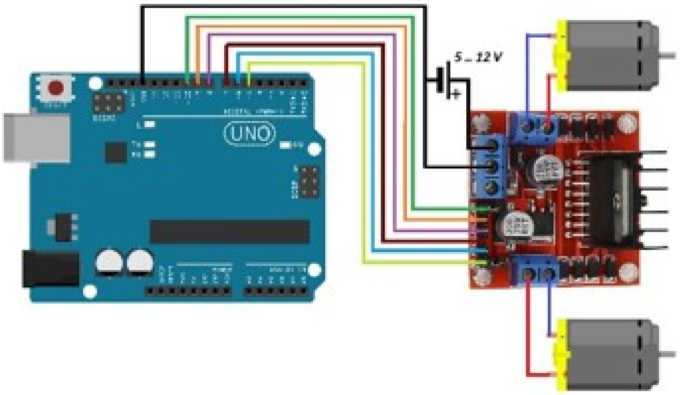

Figure 4: The L298n

2.2 Sensors

Sensors in general are devices that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena [2].

- Ultrasonic sensor : An Ultrasonic sensor is a device that can measure the distance from object A to object B using sound waves.It measures distance by sending out a sound wave at a specific frequency and waits for its Eco to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object [3]. We used this sensor to detect whether there is an other robot in front of our robot or not.A picture of the Ultrasonic sensor is shown in the figure 5.

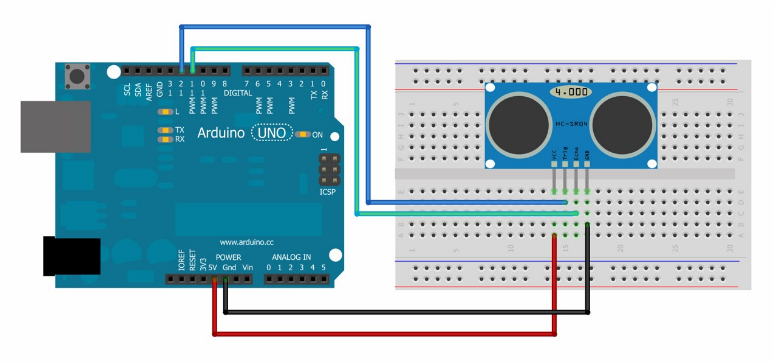

Figure 5: The Ultrasonic sensor

In the code, instead of using pulseIn() function, we included $< NewPing.h >$ library [4]. The test code of the sensor is shown downward :-

```
#include <NewPing.h>
#define TRIGGER_PIN A5
#define ECHO_PIN   A5
#define MAX_DISTANCE 200

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

void setup() {
  Serial.begin(9600);
  pinMode(A0,OUTPUT);
}

void loop() {
  delay(50);
  Serial.print("Ping: ");
  Serial.print(sonar.ping_cm());
  Serial.println("cm");

  if(sonar.ping_cm()<8 && sonar.ping_cm()>0 )
  digitalWrite(A0,HIGH);
   if(sonar.ping_cm()>8)
  digitalWrite(A0,LOW);
}
```

We used the function $sonar.ping_c m()$ to obtain the distance in cm. In the test code, we made the sensor work in a range of 0 to 200 cm in the new ping constructor and we set up a range of 0 to 8 cm for our output X to be HIGH. We had a delay of 50 milliseconds between each ping.

- IR Sensor : IR Sensor works by using a specific light sensor to detect a select light wavelength in the Infra-Red (IR) spectrum. The sensor has two main parts, one part sends the IR while the other part senses the reflected light, so it is possible to have a sensor that can return the value of the reflected light and it can be used to measure how "bright" the object is [5].We used the IR sensor to detect whether the robot has reached the edge of the arena or not.A picture of the IR sensor is shown in the figure 6.

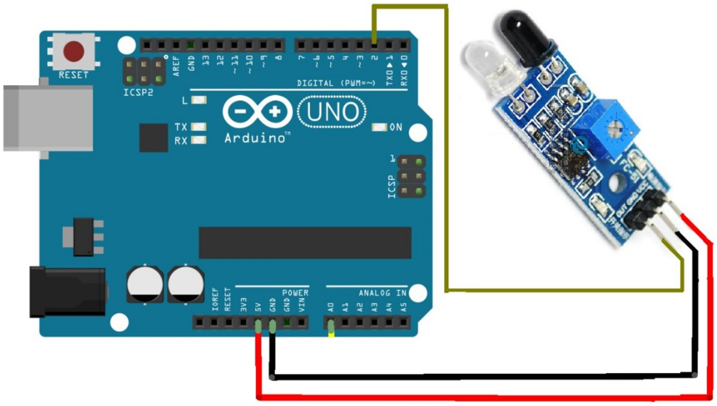

Figure 6: The IR sensor

This is our test code for the IR sensor :-

```
int obstaclePin = 6; // This is our input pin
int hasObstacle = LOW; // HIGH MEANS NO OBSTACLE

void setup() {
  pinMode(A0, OUTPUT);
  pinMode(obstaclePin, INPUT);
  Serial.begin(9600);
}
void loop() {
  hasObstacle = digitalRead(obstaclePin); //Reads the output of the obstacle sensor from the 7th PIN of the Digital section of the arduino
  if (hasObstacle == HIGH) //LOW means something is ahead, so it illuminates the 13th Port connected to an LED
  {
    Serial.println("Stop something is ahead!!");
    digitalWrite(7, HIGH);//Illuminates the 13th Port LED
  }
  else
  {
    Serial.println("Path is clear");
    digitalWrite(7, LOW);
  }
  delay(200);
}
```

In the test code, when hasObstacle is HIGH it means that the sensor is saying that there is some obstacle ahead and then a light is turned on to indicate that.

- The Remote controller : Our first choice was the IR remote controller due to its simplicity and reasonable price, but we didn't use it because we faced A weird problem with it, the sensor stopped responding when the motors started moving. After debugging we noticed that the IR sensor works fine within 0 to 70 of analog input and when we tried to exceed that range it stops working. More debugging led us to notice that if we put the IR sensor away from the motor it works fine (up to 255) so the problem might be with the magnetic field that is generated by the motors.A picture of the IR sensor is shown in the figure 7.

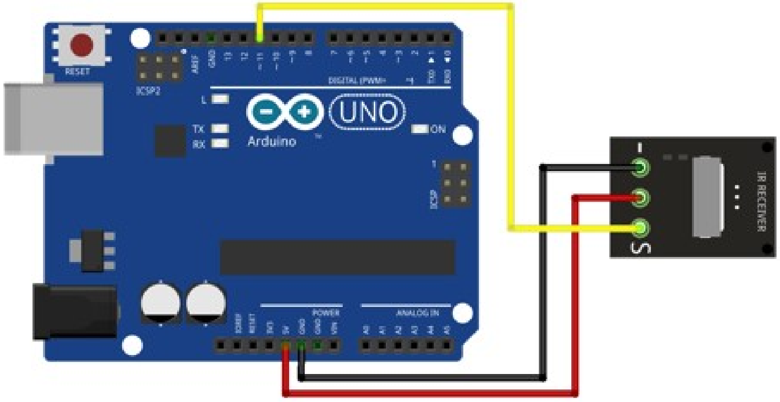

Figure 7: The Remote controller(IR)

In the test code for the remote using IR, we were controlling four motors. In order to use the IR remote, we had to include $< IRremote.h >$ library.

```
#include <IRremote.h>

int RECV_PIN = A4;
IRrecv irrecv(RECV_PIN);
decode_results results;

#define BUTTON_front 0xEFA25D
#define BUTTON_back 0xEFB24D
#define BUTTON_left 0xEF50AF
#define BUTTON_right 0xEF12ED
#define BUTTON_speedup 0xEFE817
#define BUTTON_speeddown 0xEF6897

void setup()
{
    Serial.begin(9600);
  irrecv.enableIRIn();
}
void loop() {
  if (irrecv.decode(&results)){
      Serial.println(results.value, HEX );
      if (results.value == BUTTON_front){
    // do something   }
      if (results.value == BUTTON_back){
    // do something     }
      if (results.value == BUTTON_back){
           // do something
      }
      if (results.value == BUTTON_back){
              // do something
      }
           irrecv.resume();
  }
 }
```

We faced one problem in our code which was the fact that new ping library and IR library were using the same timer (timer 1), so we edited new ping library and changed the timer$_{e}nablefromtruetofalse.Althoughwewereabletofixone$

- Bluetooth hc-05 : HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.It can function as both master and slave or we can say its a transceiver device. Master in referring to the controller or the device who sends the control signal, while slave in referring to the device that receives and obeys the control signal [6].We used a mobile app to control our robot via Bluetooth.A picture of the hc-05 sensor is shown in the figure 8.

  The test code for the Bluetooth sensor is shown downward :-

```
int state;
int flag=0;
int stateStop=0;
void setup() {

    Serial.begin(9600);
}

void loop() {
    if(Serial.available() > 0){
```

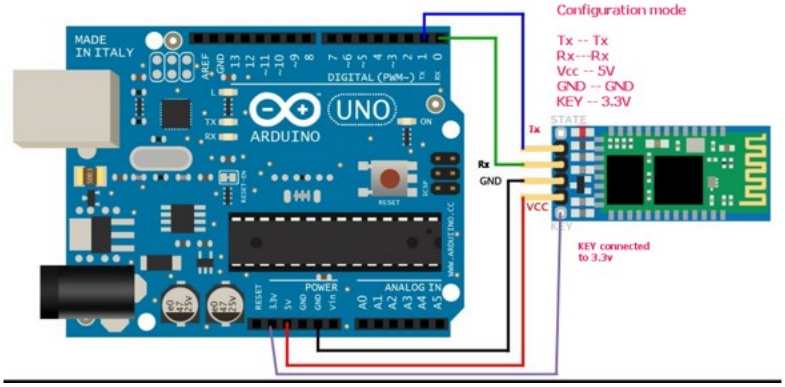

Figure 8: The Remote controller(Bluetooth)

```
   state = Serial.read();
   flag=0;
  }
  if (state == 'u') {
  // Do something
    if(flag == 0){
     Serial.println("u is pressed");
     flag=1;
    }
  }
  else if (state == 'd') {
   // Do something
    if(flag == 0){
      Serial.println("d is pressed");
     flag=1;
    }
  }
  else if (state == 's' || stateStop == 1) {
 // Do something
    if(flag == 0){
     Serial.println("s is pressed");
     flag=1;
    }
    stateStop=0;
  }
   else if (state == 'a') {
    // Do something
        if(flag == 0){
           Serial.println("a is pressed");
          flag=1;
        }
        stateStop=0;
    }

}
```

2.3 The Chassis

Here we discuss how did we make our chassis.

- PCB Print :- A printed circuit board (PCB) is a board base for physically supporting and wiring the surface-mounted and socketed components in most electronics [7].This was what we intended to have in the beginning of our project, but due to not having enough time and the 3D modeling problem, the PCB print of our circuit didn't make it to the final design.We used eagle to design our PCB,and a picture of the design is shown in the figure 9 and 10.

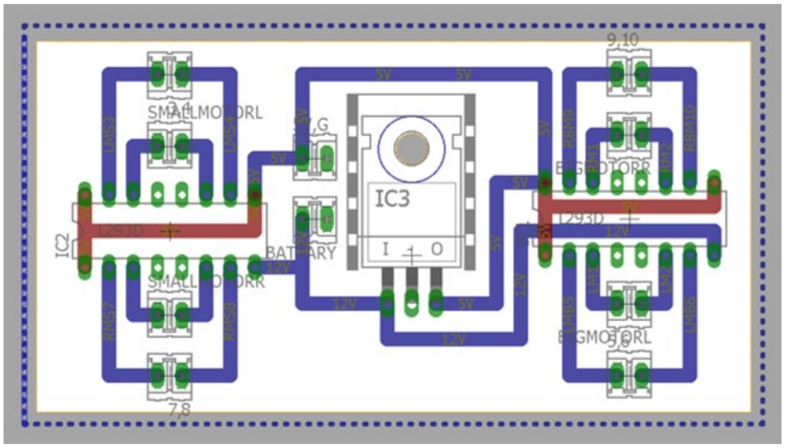

Figure 9: PCB Broad 1

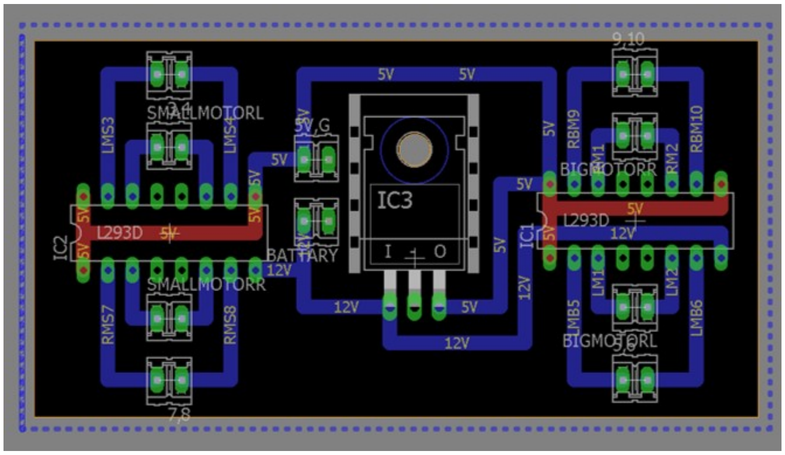

Figure 10: PCB Broad 2

- 3D Print :- A 3D printer is a computer-aided manufacturing (CAM) device that creates three-dimensional objects. Like a traditional printer, a 3D printer receives digital data from a computer as input. However, instead of printing the output on paper, a 3D printer builds a three-dimensional model out of a custom material [8].We made a 3D design of our chassis using Tinkercad, and printed it in the labs, our design had a length of 20 cm, a weight of 15 cm and a height of 6 cm. After printing, we were surprised that after the rescaling process, the design shrinked significantly and it didn't even fit one motor, thus we were not able to use it and it wasn't included in the final design.A picture of the 3D model is shown in the figure 11 and 12.

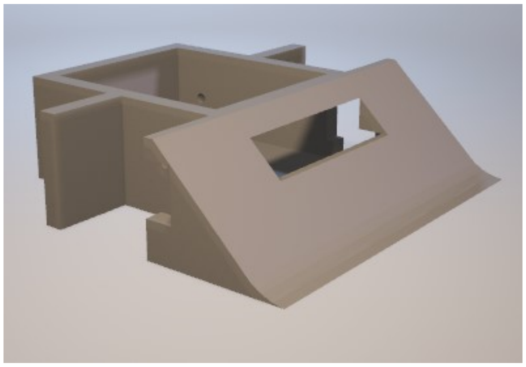

Figure 11: The 3D Model

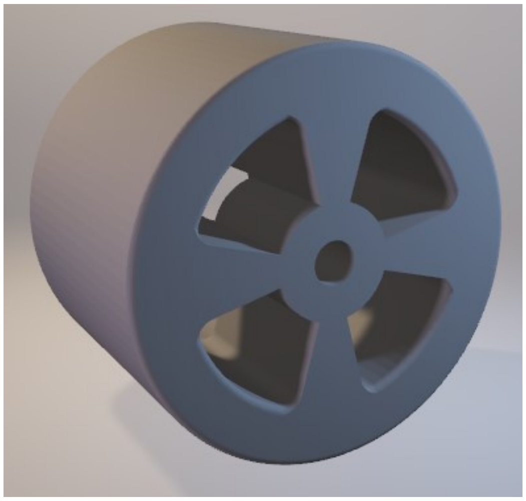

Figure 12: The Wheels

- plastic Laser Cutting :- This was our last resolve, getting thick pieces of plastic and use laser to form it in our desired shape.A picture of the final model and it's connections are shown in the figure 13, 14 and 15.

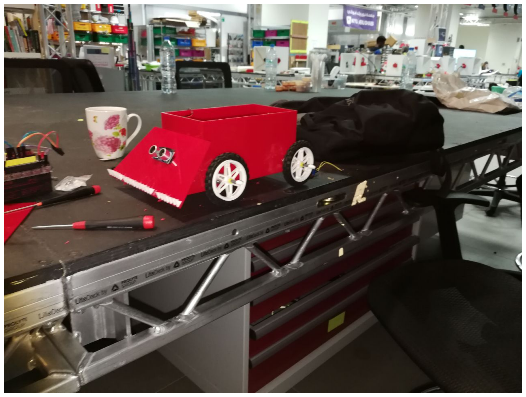

Figure 13: The Final Chassis

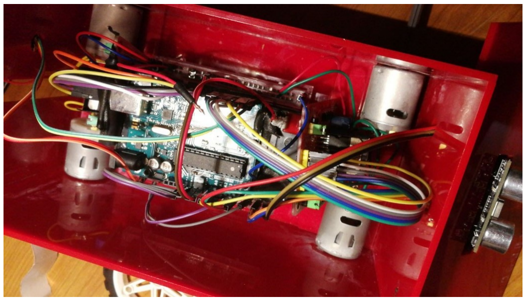

Figure 14: The Connections

Figure 15: The Wheel

2.4 Extra : The Piezo

Piezoelectricity, also called the piezoelectric effect, is the ability of certain materials to generate an AC (alternating current) voltage when subjected to mechanical stress or vibration, or to vibrate when subjected to an AC voltage, or both. The most common piezoelectric material is quartz. Certain ceramics, Rochelle salts, and various other solids also exhibit this effect. A Piezo is a device that uses the piezoelectric effect to generate sounds on certain frequency [9]. The function *tone*() allows us to decide the frequency of the sound that we want to hear.A picture of the Piezo is shown in the figure 16 and the test code in also downward,we used it in the final code to play Star war song.

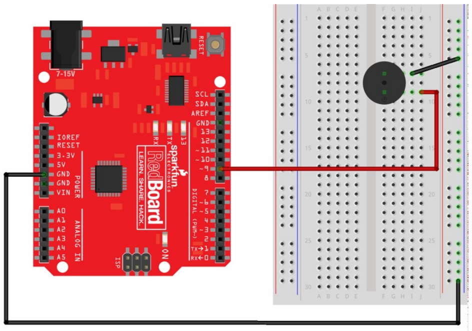

Figure 16: The Piezo

```
 void setup()
{
  Serial.begin(9600);
}

void loop()
{
  if (something)
  {
    tone(7,analogRead(A1),1000);
  }

  Serial.println(analogRead(A1));
}
```

2.5 The FSM

This is our final FSM for the project :-

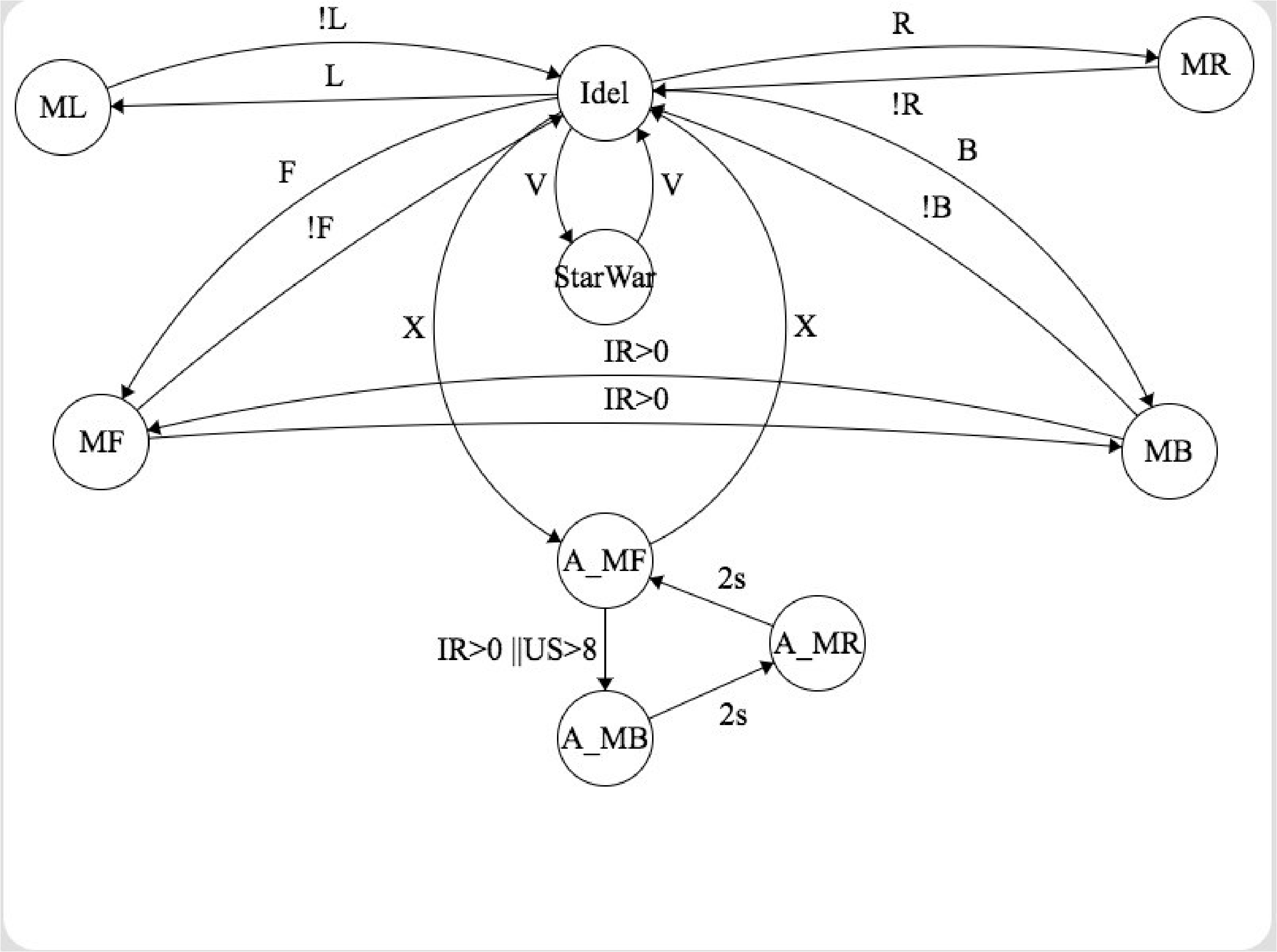

Figure 17: The Final FSM

2.6 The Final Code

This is our final code, everything is explained in the comments.

```
//calling newPing library to control ultrasoinc
#include <NewPing.h>
//define all the state that we have in the FSM
enum {Idle,MovingForward,Backward,Turnleft,Turnright,
Turn,AM_Backward,AM_MovingForward};
//define initial state to be Idle
int currentState = Idle;
// define the speed for the 4 motor to controll them later
int bmr,bml,smr,sml;
//right back motor(PWM)
#define Enable11 11
//right back motor(direction)
#define ControlMotor13 13
#define ControlMotor12 12
///left back motor(PWM)
#define Enable10 10
///left back motor(direction)
#define ControlMotor9 9
#define ControlMotor8 8
///left small motor(PWM)
#define Enable6 6
///left small motor(direction)
#define ControlMotor7 7
#define ControlMotor5 5
///right small motor(PWM)
#define Enable3 3
///right small motor(direction)
#define ControlMotor4 4
#define ControlMotor2 2
//define the state that we gone use to read the output from the blutooth
int state;
/////////////////////////////
//PIZO
#define buzzerPin A2
//define freucancy srt of(STAR WAR Song)
const int c = 261;const int d = 294;const int e = 329;const int f
```

```
    = 349;const int g = 391;const int gS = 415;const int a =
    440;const int aS = 455;const int b = 466;const int cH =
    523;const int cSH = 554;const int dH = 587;const int dSH =
    622;const int eH = 659;const int fH = 698;const int fSH =
    740;const int gH = 784;const int gSH = 830;const int aH = 880;
//counter for the pizo delay
int counter = 0;
//IR (edge sensor)
#define IRS A5
//ultrasoinc (distance sensor)
#define TRIGGER_PIN A0
#define ECHO_PIN   A1
//the MAX DISTANCE that the Ultrasoinc will detect
#define MAX_DISTANCE 200
//define the paremter for sonar method from the NewPing libray
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

void setup() {
//display Output in the serial monitor,buad is 9600
Serial.begin(9600);
//PIZO is an Output
pinMode(buzzerPin,OUTPUT);
//(PWM pin)to controll the speed for back right motor
pinMode(Enable11, OUTPUT);
//(PWM pin)to controll the speed for back left motor
pinMode(Enable10, OUTPUT);
//(PWM pin)to controll the speed for small left motor
pinMode(Enable6, OUTPUT);
//(PWM pin)to controll the speed for small right motor
pinMode(Enable3, OUTPUT);
//to controll the diraction of the back right motor
pinMode(ControlMotor13, OUTPUT);
pinMode(ControlMotor12, OUTPUT);
//to controll the diraction of the back left motor
pinMode(ControlMotor9, OUTPUT);
pinMode(ControlMotor8, OUTPUT);
//to controll the diraction of the small left motor
pinMode(ControlMotor7, OUTPUT);
pinMode(ControlMotor5, OUTPUT);
//to controll the diraction of the small right motor
```

```
pinMode(ControlMotor4, OUTPUT);
pinMode(ControlMotor2, OUTPUT);
 }



void loop() {
    //Calling IRInputCheck method which will check the output from the Buletooth
    IRInputCheck();}



//PIZO(StarWar) START
//beep method take two paremters(frecuany and duration).
void beep(int note, int duration)
{if(state=='V'){tone(buzzerPin, note, duration);if(counter % 2 == 0){delay(duration);}else{delay(duration);}noTone(buzzerPin);delay(50);counter++;}}
//this is the first saction of the star war song
void firstSection(){
 beep(a,500);beep(a,500);beep(a,500);beep(f,350);
 beep(cH,150);beep(a,500);beep(f,350);beep(cH,150);beep(a,650);
 delay(500);
 beep(eH,500);beep(eH,500);beep(eH,500);beep(fH,350);
 beep(cH,150);beep(gS,500);beep(f,350);beep(cH,150);beep(a,650);
 delay(500);}
//this is the second saction of the star war song
void secondSection(){
 beep(aH,500);beep(a,300);beep(a,150);beep(aH,500);
 beep(gSH,325);beep(gH,175);beep(fSH,125);beep(fH,125);beep(fSH,250);
 delay(325);
 beep(aS,250);beep(dSH,500);beep(dH,325);
 beep(cSH,175);beep(cH,125);beep(b,125);beep(cH,250);
 delay(350);}
//this is the third saction of the star war song
void thardpart(){beep(f,250);beep(gS,500);beep(f,350);beep(a,125);beep(cH,500);beep(a,375);beep(cH,125);beep(eH,650);
  delay(500);
  secondSection();beep(f,250);beep(gS,500);beep(f,375);
  beep(cH,125);beep(a,500);beep(f,375);beep(cH,125);beep(a,650);
  delay(650);}
//PIZO END


//this method lisean for the BLUETHOOTH
void IRInputCheck() {
  //cheack if there is signal coming from the bluethooth
  if(Serial.available() > 0){
    //we set the state to what the bluetooth send
     state = Serial.read();}
    //chack the state output
    //state can be 3 type of button
    //(Swith Button)first press (on),second press (off)
    //(Press Button)pressing (on),stop pressing (off)
    //(moveing bar)eacth move for the bar triger a diffrent button.
    switch (state) {
     //when no button is pressed
      default :
       Serial.println("Idle");
       //currentState go to the Idle
       currentState = Idle;
           //shut down all motors
           digitalWrite(ControlMotor13, LOW);
           digitalWrite(ControlMotor12, LOW);
           digitalWrite(ControlMotor9, LOW);
           digitalWrite(ControlMotor8, LOW);
           digitalWrite(ControlMotor7, LOW);
           digitalWrite(ControlMotor5, LOW);
           digitalWrite(ControlMotor4, LOW);
           digitalWrite(ControlMotor2, LOW);
       break;
     //when X is pressed(Swith Button)
     case ’X’:
       //currentState go to the AM_MovingForward
       //this is the auto obstacle avoider
      currentState = AM_MovingForward;
      //if the currentState is AM_MovingForward
      if (currentState == AM_MovingForward) {
           Serial.println("AM_MovingForward");
           //set the speed of the motors
           analogWrite(Enable11, bmr); analogWrite(Enable10, bml);
           analogWrite(Enable6, smr); analogWrite(Enable3, sml);
           //all motors go forward
           digitalWrite(ControlMotor13, LOW);
           digitalWrite(ControlMotor12, HIGH);
           digitalWrite(ControlMotor9, LOW);
           digitalWrite(ControlMotor8, HIGH);
           digitalWrite(ControlMotor7, LOW);
           digitalWrite(ControlMotor5, HIGH);
           digitalWrite(ControlMotor4, LOW);
           digitalWrite(ControlMotor2, HIGH);
           //if the car about to fall
           if (digitalRead(IRS) == HIGH)
              //currentState is AM_Backward
              currentState = AM_Backward ;
           //if the car about to hit somthing that far from us 8cm
           //snor.ping_cm retuen the distance of the object in cm
           if(sonar.ping_cm()<8 && sonar.ping_cm()>0 )
              //currentState is AM_Backward
              currentState = AM_Backward ;
        }
      //if the currentState is AM_MovingForward
      if (currentState == AM_Backward) {
            Serial.println("AM_Backward");
           //set the speed of the motors
           analogWrite(Enable11, bmr); analogWrite(Enable10, bml);
           analogWrite(Enable6, smr); analogWrite(Enable3, sml);
           //all motors go backward
           digitalWrite(ControlMotor13, HIGH);
           digitalWrite(ControlMotor12, LOW);
           digitalWrite(ControlMotor9, HIGH);
           digitalWrite(ControlMotor8, LOW);
           digitalWrite(ControlMotor7, HIGH);
           digitalWrite(ControlMotor5, LOW);
           digitalWrite(ControlMotor4, HIGH);
           digitalWrite(ControlMotor2, LOW);
           //wait 2 s
           delay(2000);
            //currentState is Turn
            currentState = Turn;
```

```
     }
     if (currentState == Turn) {
          Serial.println("Turn");
          //set the speed of the motors
          analogWrite(Enable11, bmr); analogWrite(Enable10, bml);
          analogWrite(Enable6, smr); analogWrite(Enable3, sml);
          //motors Turn Right
          digitalWrite(ControlMotor13, HIGH);
          digitalWrite(ControlMotor12, LOW);
          digitalWrite(ControlMotor9, LOW);
          digitalWrite(ControlMotor8, LOW);
          digitalWrite(ControlMotor7, HIGH);
          digitalWrite(ControlMotor5, LOW);
          digitalWrite(ControlMotor4, LOW);
          digitalWrite(ControlMotor2, LOW);
          //wait 2 s
          delay(2000);
          //currentState is AM_MovingForward(after we escape the obstacle we continue moving)
          currentState = AM_MovingForward;
     }
     break;
    //when F is pressed(press button)
    case ’F’:
    //currentState is MovingForward
    currentState = MovingForward;
    if(currentState==MovingForward){
        Serial.println("Moving Forward");
          //set the speed of the motors
          analogWrite(Enable11, bmr); analogWrite(Enable10, bml);
          analogWrite(Enable6, smr); analogWrite(Enable3, sml);
          //all motors go forward
          digitalWrite(ControlMotor13, LOW);
          digitalWrite(ControlMotor12, HIGH);
          digitalWrite(ControlMotor9, LOW);
          digitalWrite(ControlMotor8, HIGH);
          digitalWrite(ControlMotor7, LOW);
          digitalWrite(ControlMotor5, HIGH);
          digitalWrite(ControlMotor4, LOW);
          digitalWrite(ControlMotor2, HIGH);
```

```
         //if the car about to fall
         if (digitalRead(IRS) == HIGH)
         //move back
         state='B';
        }
    //when B is pressed(press button)
    case 'B':
          Serial.println("Backward");
         //currentState is Backward
          currentState = Backward;
         //set the speed of the motors
         analogWrite(Enable11, bmr); analogWrite(Enable10, bml);
         analogWrite(Enable6, smr); analogWrite(Enable3, sml);
         //all motors go backward
         digitalWrite(ControlMotor13, HIGH);
         digitalWrite(ControlMotor12, LOW);
         digitalWrite(ControlMotor9, HIGH);
         digitalWrite(ControlMotor8, LOW);
         digitalWrite(ControlMotor7, HIGH);
         digitalWrite(ControlMotor5, LOW);
         digitalWrite(ControlMotor4, HIGH);
         digitalWrite(ControlMotor2, LOW);
         //if the car about to fall
          if (digitalRead(IRS) == HIGH)
         //move foward
          state='F' ;
      break;
    //when R is pressed(press button)
    case 'R':
      Serial.println("Turn right");
    //currentState is Turnright
      currentState = Turnright;
         //set the speed of the motors
         analogWrite(Enable11, bmr); analogWrite(Enable10, bml);
         analogWrite(Enable6, smr); analogWrite(Enable3, sml);
         //motors Turnright
         digitalWrite(ControlMotor13, HIGH);
         digitalWrite(ControlMotor12, LOW);
         digitalWrite(ControlMotor9, LOW);
         digitalWrite(ControlMotor8, LOW);
```

```
            digitalWrite(ControlMotor7, HIGH);
            digitalWrite(ControlMotor5, LOW);
            digitalWrite(ControlMotor4, LOW);
            digitalWrite(ControlMotor2, LOW);
        break;
      //when L is pressed(press button)
      case 'L':
        Serial.println("Turn left");
        //currentState is Turnleft
        currentState = Turnleft;
            //set the speed of the motors
            analogWrite(Enable11, bmr); analogWrite(Enable10, bml);
            analogWrite(Enable6, smr); analogWrite(Enable3, sml);
            //motors Turnleft
            digitalWrite(ControlMotor13, LOW);
            digitalWrite(ControlMotor12, HIGH);
            digitalWrite(ControlMotor9, LOW);
            digitalWrite(ControlMotor8, LOW);
            digitalWrite(ControlMotor7, LOW);
            digitalWrite(ControlMotor5, HIGH);
            digitalWrite(ControlMotor4, LOW);
            digitalWrite(ControlMotor2, LOW);
        break;
      //when V is pressed(Swith Button)
      case 'V':
        Serial.println("PIZO");
        //play the (Star war song)
        firstSection();
        secondSection();
        thardpart();
        break;

      //controll the speed
      //when 0 is pressed(moveing bar)
      case '0':
        //set the speed to 0
        bmr=0;bml=0;smr=0;sml=0;
        break;
      //when 1 is pressed(moveing bar)
      case '1':
        //set the speed to 25
        bmr=25;bml=25;smr=25;sml=25;
        break;
      //when 2 is pressed(moveing bar)
      case '2':
        //set the speed to 50
         bmr=50;bml=50;smr=50;sml=50;
        break;
      //when 3 is pressed(moveing bar)
      case '3':
        //set the speed to 80
        bmr=80;bml=80;smr=80;sml=80;
        break;
      //when 4 is pressed(moveing bar)
      case '4':
        //set the speed to 100
        bmr=100;bml=100;smr=100;sml=100;
        break;
      //when 5 is pressed(moveing bar)
      case '5':
        //set the speed to 125
        bmr=125;bml=125;smr=125;sml=125;
         break;
      //when 6 is pressed(moveing bar)
      case '6':
        //set the speed to 150
        bmr=150;bml=150;smr=150;sml=150;
         break;
      //when 7 is pressed(moveing bar)
      case '7':
        //set the speed to 180
        bmr=180;bml=180;smr=180;sml=180;
         break;
      //when 8 is pressed(moveing bar)
      case '8':
        //set the speed to 205
        bmr=205;bml=205;smr=205;sml=205;
         break;
      //when 9 is pressed(moveing bar)
      case '9':
        //set the speed to 225
        bmr=225;bml=225;smr=225;sml=225;
         break;
      //when q is pressed(moveing bar)
      case 'q':
        //set the speed to 255
        bmr=255;bml=255;smr=255;sml=255;
        break;
       }
  }
//BLUETHOOTH END
```

3 Conclusion

In conclusion, The project was about making our own Sumo Robot.Our initial intentions were to design a PCb and a 3D model, then print both of them and use it as the base of our design. But we faced problems with both printing PCB and the 3D model for the chassis.We managed to solder the connection by ourselves, and made the body using laser cutting. Unfortunately, we were not able to participate in the Sumo competition due to a problem with the wheels, our motors were too strong for the wheels that we got from a Lego store.We used a round 83 US Dollar of our budget and were manage to save 17 US Dollars, more details are explained in the order proof.Ignoring the fact that we were not able to participate, the project was a really fun educational experience.

4 Appendix

Here the stuff that we have used :-

1. The 3D design
2. The PCB Design
3. The Order Proof
4. The .ion file for the code

References

[1] (2015) Motor Shield. [Online]. Available: http://wiki.seeedstudio.com/wiki/Motor$_{S}hield$

[2] M. Rouse. (2012) Sensors. [Online]. Available: http://whatis.techtarget.com/definition/sensor

[3] What is an Ultrasonic Sensor? [Online]. Available: http://education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic$_{s}ensor/1.html$

[4] T. Eckel. (2016) NewPing Library for Arduino. [Online]. Available: https://playground.arduino.cc/Code/NewPing

[5] What is an IR Sensor? [Online]. Available: http://education.rec.ri.cmu.edu/content/electronics/boe/ir$_{s}ensor/1.html$

[6] (2017) Serial Port Bluetooth Module (Master/Slave) : HC-05. [Online]. Available: https://www.itead.cc/wiki/Serial$_{P}ort_{B}luetooth_{M}odule_{(}Master/Slave)_{:H}C-$05

[7] M. Rouse. (2013) printed circuit board (PCB). [Online]. Available: http://whatis.techtarget.com/definition/printed-circuit-board-PCB

[8] P. Christensson. (2014) 3D Printer. [Online]. Available: https://techterms.com/definition/3d$_{p}rinter$

[9] M. Rouse. (2016) piezoelectricity. [Online]. Available: http://whatis.techtarget.com/definition/piezoelectricity