# ROBOTICS PROJECT
VOICE CONTROLLED ROBOT
# REPORT

Sri Aditya Deevi
B-TECH ECE(AVIONICS)
Indian Institute of Space science and Technology (IIST)

31 May, 2020

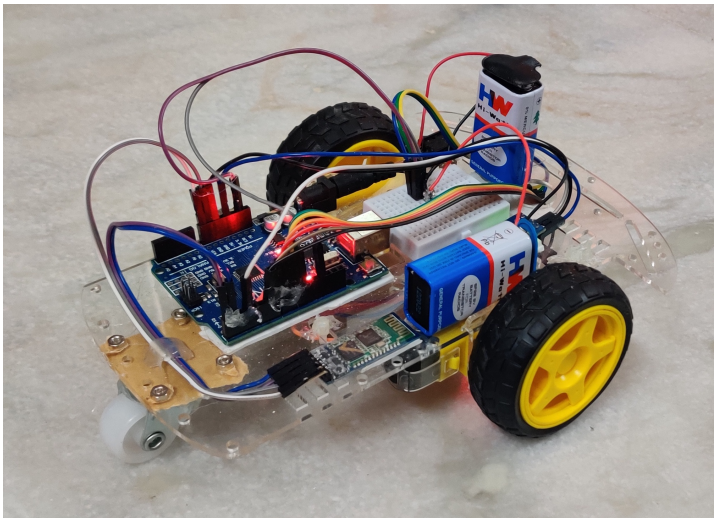
## BASIC DESCRIPTION

The problem statement (basic functional description) is that the robot built should be controlled by the given voice commands from the phone. The voice is transferred from the mobile phone application to the Bluetooth module connected to the Arduino UNO. So, basically, the bot can be made to move in a particular direction for a specific distance.

*Example :* If we say "*forward 10*" , the robot should move 10 centimetres forward.

## COMPONENTS REQUIRED

- Arduino UNO with cable
- Male and Female Jumper Connectors
- 9V batteries
- HC05 Bluetooth Module
- Wheels
- Chassis
- Wire Straps
- Soldering Iron
- L298N Motor Driver Module
- LM393 Speed Sensor
- Encoder Wheel
- Clip Connectors
- Barrel Jack
- DC Motors
- Mini Breadbook
- Glue Gun



*Final Model*

# OUTLINE OF PROCEDURE

1. The Robot chassis is built and various components are mounted on it in a planned fashion for stability.
2. The electrical connections are made according to the given schematic.
3. The program written is uploaded to the Arduino with the help of Arduino IDE.
4. The bot is tested[1] using the *AMR_Voice* application for giving voice commands to the bot.

# SOURCE CODE

```
#include<string.h> // String Library

char *p;
int dist;
String command;

//   Left Wheel
int ENA = 11;
int in1 =10;
int in2 =9;

//   Right Wheel
int in3 = 8;
int in4 = 7;
int ENB = 6;

const byte sensor = 3; // Attached to Left Wheel
String voice;

const float step_count = 20;      // No. of Slots in the encoder wheel
const float wheel_diameter = 60; // Diameter of the wheel used
volatile int count = 0;



void ISR_count(){  // Interrupt Service Routine
count++;
}



int CM_to_steps(float cm){   // Function to convert given command in cm to no: steps of
    the speed encoder
int result;
float circumference = (3.14*wheel_diameter)/10;
float cm_per_step = (circumference/step_count);
float f_result = (cm/cm_per_step);
result = (int) (f_result);
return result;
}

```

[1]Working video proof is attached with the report

```
//  Forward Function
void MoveForward(int steps)
{
count = 0;
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);

while(steps>count){
delay(200);
analogWrite(ENA,150);
analogWrite(ENB,150);
delay(200);
}
analogWrite(ENA,0);
analogWrite(ENB,0);

}


//  Reverse Function
void MoveReverse(int steps )
{
count = 0;
digitalWrite(in1,LOW);
digitalWrite(in2,HIGH);
digitalWrite(in3,LOW);
digitalWrite(in4,HIGH);

while(steps>count){
delay(200);
analogWrite(ENA,150);
analogWrite(ENB,150);
delay(200);
}
analogWrite(ENA,0);
analogWrite(ENB,0);

}



//  Left Function
void MoveLeft(int steps)
{
count = 0;

// Turn Left
digitalWrite(in1,LOW);
digitalWrite(in2,HIGH);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);
delay(200);
```

```
analogWrite(ENA,100);
analogWrite(ENB,100);
delay(600);

//Move Forward
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);

while(steps>count){
delay(100);
analogWrite(ENA,150);
analogWrite(ENB,150);
delay(200);
}
analogWrite(ENA,0);
analogWrite(ENB,0);

}



// Right Function

void MoveRight(int steps)
{
count = 0;

// Turn Right
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,LOW);
digitalWrite(in4,HIGH);
delay(200);
analogWrite(ENA,100);
analogWrite(ENB,100);
delay(600);

// Move Forward
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);

while(steps>count){
delay(100);
analogWrite(ENA,150);
analogWrite(ENB,150);
delay(200);
}
analogWrite(ENA,0);
analogWrite(ENB,0);

```

```
149 }
150
151
152
153 void setup() {
154 // put your setup code here, to run once:
155 pinMode(in1,OUTPUT);
156 pinMode(in2,OUTPUT);
157 pinMode(ENA,OUTPUT);
158 pinMode(in3,OUTPUT);
159 pinMode(in4,OUTPUT);
160 pinMode(ENB,OUTPUT);
161 Serial.begin(9600);
162 attachInterrupt(digitalPinToInterrupt(sensor),ISR_count,RISING);
163 }
164
165
166
167 void loop() {
168 // put your main code here, to run repeatedly:
169 while(Serial.available()){
170
171
172 delay(10);
173 char c = Serial.read();
174 if(c=='#'){
175 break;
176 }
177
178 voice += c;
179 }
180 if(voice.length() > 0){
181
182 Serial.println(voice);
183
184
185 p = strtok(voice.c_str(),"␣");
186 command = p;
187 p = strtok(NULL,"␣");
188 dist = String(p).toInt();
189
190 if(command == "*forward"){
191
192 MoveForward(CM_to_steps(dist));
193 Serial.println(dist);
194 }
195
196 else if(command == "*reverse"){
197
198 MoveReverse(CM_to_steps(dist));
199
200 }
201 else if(command == "*left"){
202
```

```
MoveLeft(CM_to_steps(dist));

}
else if(command == "*right" || command == "*write"){

MoveRight(CM_to_steps(dist));

}
}
voice ="";

}
```

## SCHEMATIC