Voice control car using Arduino

A Project Report submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology (B.Tech.)

in

Computer Science and Engineering

by

Ankit Prasad, (205112015004)
Parnab Mukherjee, (205112015003)
Garima Chakraborty, (205212015002)
Koyena Chattopadhyaya, (205212015001)
4th Semester

under the supervision of

Mr. Abhoy Kole

Department of Computer Science and Engineering
JIS University, Kolkata
April, 2020

INTRODUCTION

The Voice Controlled Robot is controlled through voice commands given by the user who is operating the project. These voice command needs to be given through an android app which is installed on the user's android mobile. Please note that user should have good internet connection in order to have a smooth operation of the android application. Speech recognition is done within the android app and then a respective command is sent to the voice controlled robot vehicle. Microcontroller fitted on the Robot decodes these commands and gives an appropriate command to the motors connected to the robot.

In voice operated robot the robotic vehicle movement controlled via voice command. This is android base project. In this project we use Bluetooth device to receive command from user. User use android application to give command to Bluetooth device .Bluetooth device receive command from application and transmit same to 8051 microcontroller then microcontroller controlling the robotic vehicle as per command. Microcontroller is the main unit of our project. It's a central processing unit (CPU) of the Robot. It receives various commands from BLUETOOTH decoder and gives the respective output motor driver ICs

This robotic vehicle operated on 7 command like forward, reverse, left, right, stop, left U turn, right U turn. This project we use 8051 microcontroller after receiving above command microcontroller to move the motors as per command. The communication between android application and Bluetooth is serial communication .here we use LCD display for the displaying project title it also shows different command given by user.

In this project, the robot is controlled by Android mobile phone that connects to Bluetooth decoder attached to the robot. Thus we need 1 mobile & 1 Bluetooth decoder. Bluetooth decoder is always connected to Robot and another mobile phone is used to control the movements of the Robot. It consists of Bluetooth decoder. It gives ASCII code output. This receiver enables wireless transmission & reception of serial data. It has 10 meters range. DC motor used for robotic vehicle movement

Project Hardware Requirements:

- **Arduino uno**
- **HC05 Bluetooth model**
- **L298 motor driver**
- **3-4 pieces of 300 rpm motor**
- **12v-battery**
- **Wires**
- **4 pieces of wheels**

Software Requirements:

- Arduino IDE 1.8.8
- Operating System Windows 7x64

Arduino Board Description:

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage(limit)	6-20V
Digital I/O Pins	14(of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC current per I/O Pin	20mA
DC current for 3.3V Pin	50mA
Flash Memory	32KB(ATmega328P)of which 0.5KB by boot loader
SRAM	2KB(ATmega328P)
EEPROM	1KB(ATmega328P)
Clock Speed	16MHz
LET_BUILTIN	13
Length	68.6mm
Width	53.4mm
Weight	25g

CIRCUIT DIAGRAM

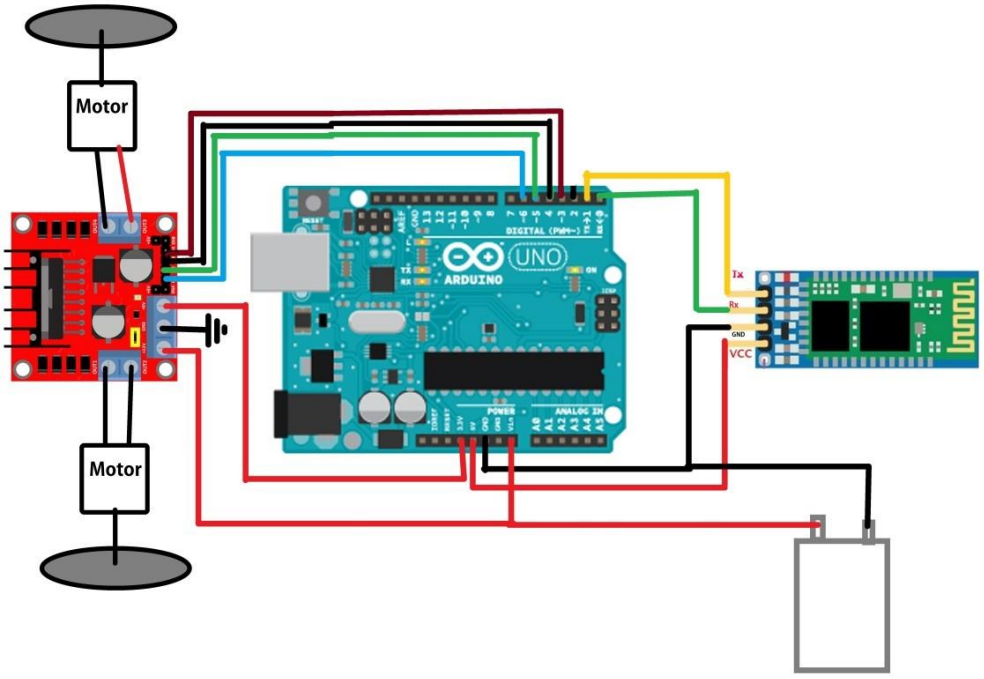

Figure 1:CIRCUIT DIAGRAM

WORKING PRINCIPLE

If you make all connections correctly then download **AMR_Voice** application from Play Store. Open the application, power up the robot and connect the application to the Bluetooth module. When we speak the specified commands the application sends the instruction to the Arduino through Bluetooth andthen Arduino performs the defined operation.

Note: Always disconnect the Rx and Tx pins of Bluetooth module when uploading code to Arduino otherwise it shows some errors and code does not upload. Try “1234” or “0000” password when you connect Bluetooth module with your mobile for the first time. If the wheels are rotating in opposite direction just swaps the connection of the motor from the Motor Driver Shield terminal. Controlling robot through voice is not an efficient way. It is fun to control but I recommend the Gesture control application.

Installation Steps of Arduino 1.8.8:

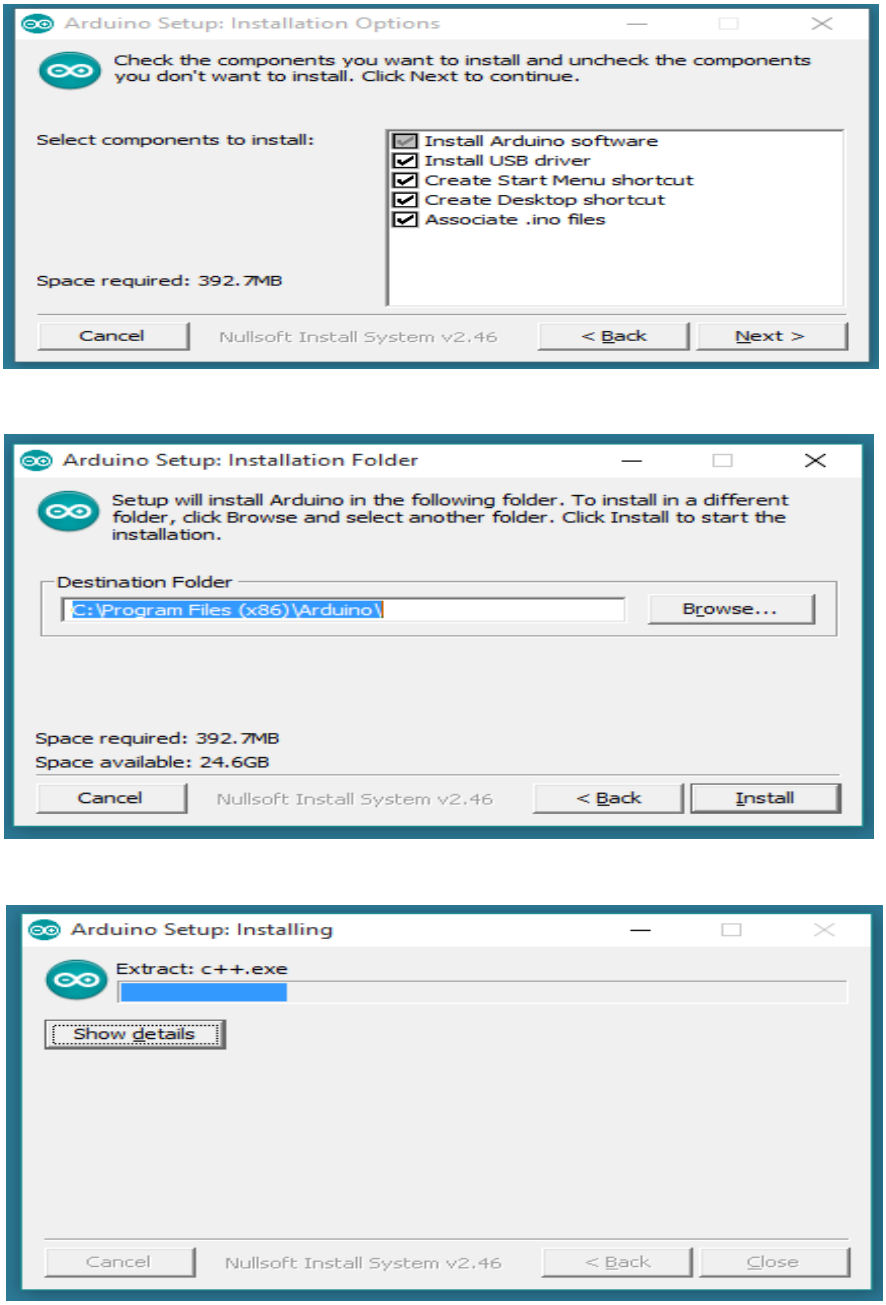

Figure 2:Installation steps

CHAPTER 1

Arduino uno:

The Arduino UNO ia an open-source microcontroller board based On the Microchip ATmega328p microcontroller and developed by Arduino.cc.The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits.

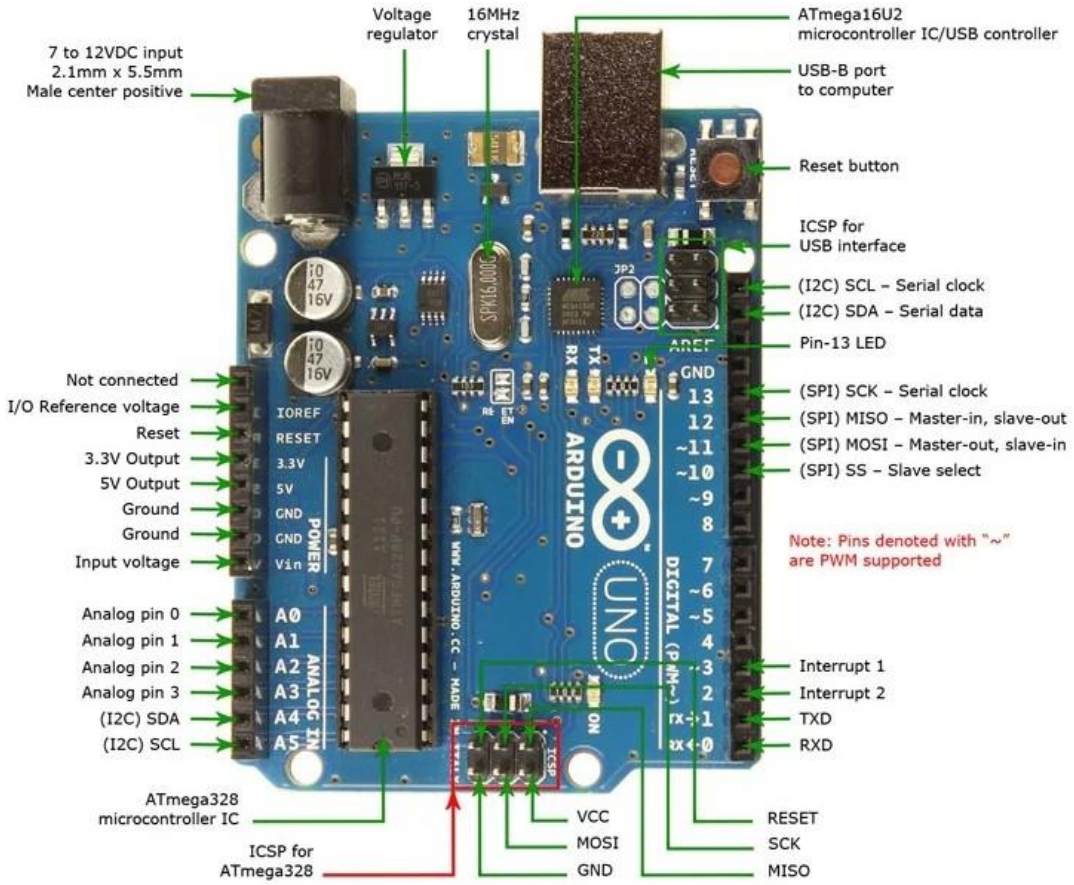

Figure 3: ARDUINO UNO

CHAPTER 2

HC-05 Bluetooth model:

HC-05 module is an easy to use Bluetooth SPP(Serial Port Protocol) module,designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR(Enhanced Data Rate)3Mbps Modulation with complete 2.4GHZ radio transreceiver and baseband.

Figure 4: HC05 Bluetooth model

CHAPTER 3

L298 motor driver:

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerS020 packages.It is a high voltage,high current dual full-bridge Driver de-signed to accept standard TTL logic level sand drive inductive Loads such as relays,solenoids,DC and stepping motors.

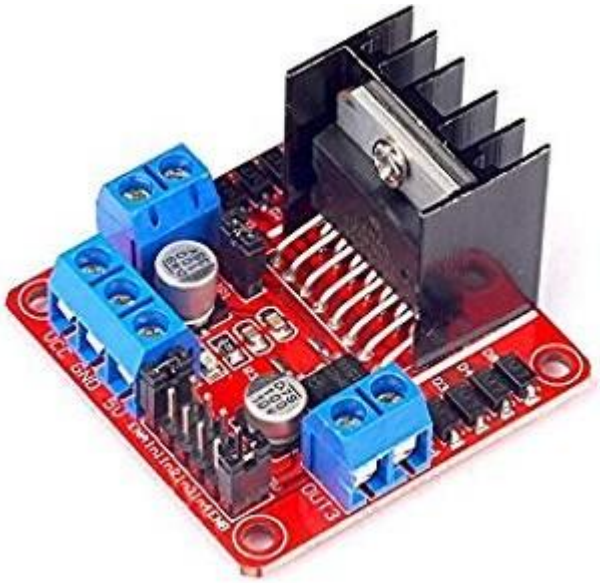

Figure 5: L298 motor driver

CHAPTER 4

300RPM motor:

300RPM motors for robotics applications.Nut and threads shaft to easily connect and internal threaded shaft for easily connecting it to wheel.High torque High precision Encoder DC geared Motor 300RPM.

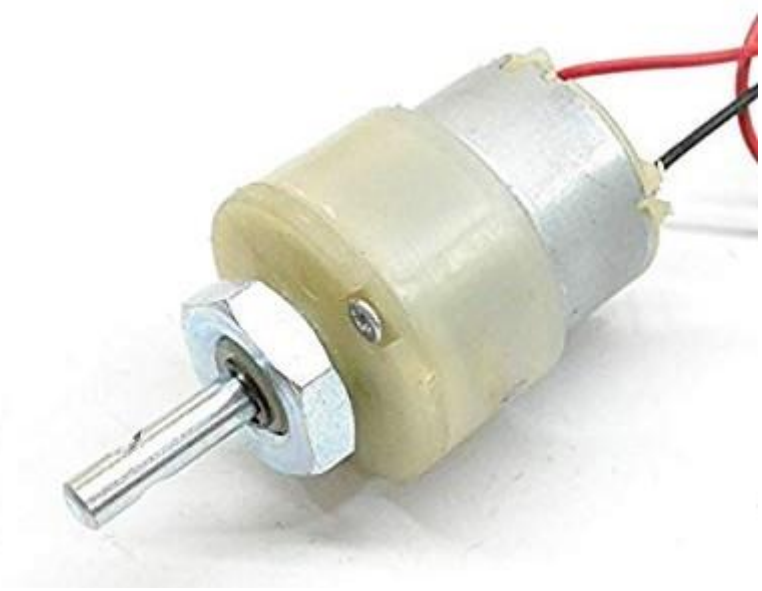

Figure 6: 300rpm motor

CHAPTER 5

12-volt battery:

12-volt battery may refer to:car battery(most batteries for vehicles and UPS are 12 volts) Lantern battery.A23 battery,a small battery(roughly 2/3 of an AAA battery in length)made for RF transmitters.

Figure 7: 12v-battery

CHAPTER 6

Jumper Wires:

Jumper wires are used for making connections between items on your breadboard and your Arduino's header pins. Use them to wire up all your circuits.

Figure 8: Jumper Wires

CHAPTER 7

Wheels:

Figure 9:Wheels

CHAPTER 8

OUTPUT-

- **When we say “go” as a command in “AMR voice” software , the car will move forward.**

Figure 10: PICTURE OF CAR MOVING FORWARD

- **When we say “back” as a command in “AMR voice” software , the car will move backward.**

Figure 11: PICTURE OF CAR MOVING BACKWARD

- **When we say “left” as a command in “AMR voice” software , the car will move left side.**

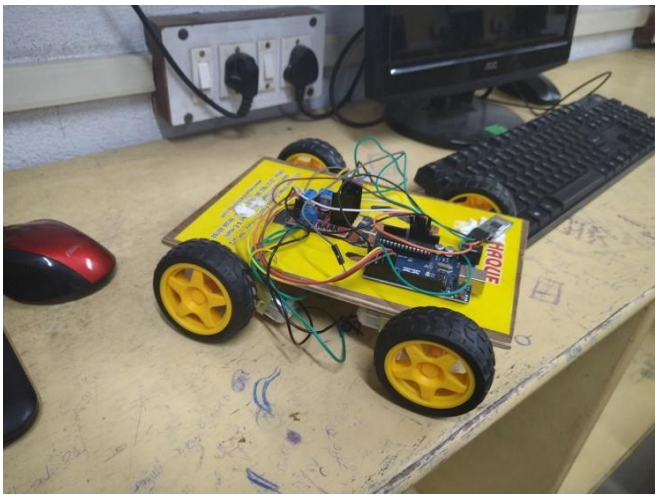

Figure 12: PICTURE OF CAR MOVING LEFT SIDE

- **When we say “right” as a command in “AMR voice” software , the car will move right.**

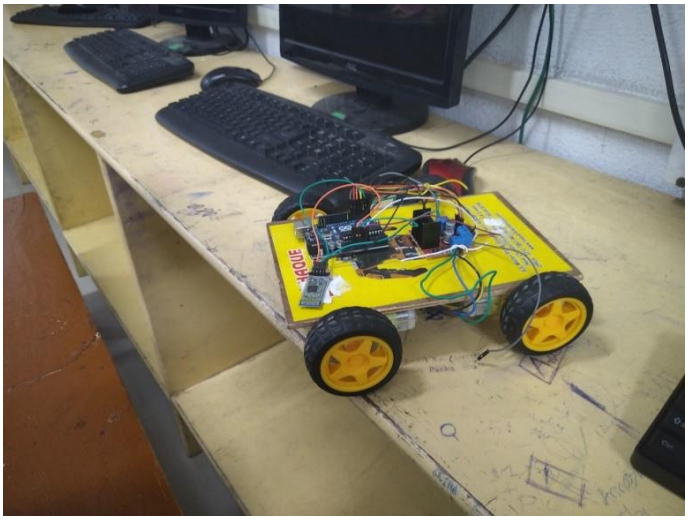

Figure 13: PICTURE OF CAR MOVING RIGHT SIDE

ADVANTAGES

- The robot is useful in places where human finds difficult to reach but human voice reaches.E.g. in a small pipeline, in a fire situations, in highly toxic areas.
- The robot can be used as a toy.
- It can be used to bring and place small objects.

DISADVANTAGES

- Even the best speech recognition systems sometimes make errors. If there is noise or some other sound in the room (e.g. the television or a kettle boiling), the number of errors will increase.
- Speech Recognition works best if the microphone is close to the user (e.g. in a phone, or if the user is wearing a microphone). More distant microphones (e.g. on a table or wall) will tend to increase the number of errors.
- In Speech recognition system, there is a possibility of unauthorized usage. Since this doesn't depends upon which person is speaking.
- No password protection

CONCLUSION

This project Voice Controlled Robotic Vehicle helps to control robot through voice commands received via android application. The integration of control unit with Bluetooth device is done to capture and read the voice commands. The robotic vehicle then operates as per the command received via android application. For this 8051 microcontroller is integrated in the system which makes it possible to operate the vehicle via android application. The controlling device may be any android based Smartphone/tab etc having an android OS. The android controlling system provides a good interactive GUI that makes it easy for the user to control the vehicle. The transmitter uses an android application required for transmitting the data.

The receiver end reads these commands and interprets them into controlling the robotic vehicle.

The android device sends commands to move the vehicle in forward, backward, right and left directions.

After receiving the commands, the microcontroller then operates the motors I order to move the vehicle in four directions. The communication between android device and receiver is sent as serial communication data. The microcontroller program is designed to move the motor through a motor driver IC as per the commands sent by android device.

REFERENCE

1. https://www.youtube.com/watch?v=fNFVHupdBic&t=268s
2. https://www.youtube.com/watch?v=BIjGDMlN3jI
3. https://www.youtube.com/watch?v=yXNq37mC0nU
4. https://www.youtube.com/watch?v=QYa02BYWD14&t=12s
5. https://create.arduino.cc/projecthub/Yug_Ajmera/voice-controlled-car-983ef1
6. https://www.instructables.com/id/How-to-make-a-voice-control-direction-control-remo/
7. https://www.instructables.com/id/Voice-Controlled-Bluetooth-Car/
8. https://www.pantechsolutions.net/voice-controlled-robot-using-arduino
9. https://microcontrollerslab.com/voice-controlled-robot-arduino/
10. https://www.robotshop.com/community/robots/show/voice-control-arduino-robot
11. https://www.instructables.com/id/Arduino-Voice-Controlled-Robot-Bluetooth-and-Smart/
12. http://www.ijetcse.com/wp-content/plugins/ijetcse/file/upload/docx/287WIRELESS-VOICE-CONTROLLED-ROBOTICS-ARM-pdf.pdf
13. http://www.ijirt.org/master/publishedpaper/IJIRT101709_PAPER.pdf
14. https://www.researchgate.net/publication/313574397_Design_and_Implementation_of_a_Voice_Controlled_Robot_with_Human_Interaction_Ability

APPENDIX

CODE FOR ARDUINO:

```
#include <SoftwareSerial.h>

SoftwareSerial BT(0, 1); //TX, RX respetively
String readvoice;

void setup() {
 BT.begin(9600);
 Serial.begin(9600);
 pinMode(3, OUTPUT);
 pinMode(4, OUTPUT);
 pinMode(5, OUTPUT);
 pinMode(6, OUTPUT);

}
//.....//
void loop() {
  while (BT.available()){ //Check if there is an available byte to read
  delay(10); //Delay added to make thing stable
  char c = BT.read(); //Conduct a serial read
  readvoice += c; //build the string- "forward", "reverse", "left" and "right"
  }
```

```
if (readvoice.length() > 0) {
  Serial.println(readvoice);

if(readvoice == "*go#")
{
  digitalWrite(3, HIGH);
  digitalWrite (4, HIGH);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  delay(100);
}

else if(readvoice == "*back#")
{
  digitalWrite(3,  LOW);
  digitalWrite(4,  LOW);
  digitalWrite(5, HIGH);
  digitalWrite(6,HIGH);
  delay(100);
}

else if (readvoice == "*left#")
{
  digitalWrite (3,HIGH);
  digitalWrite  (4,LOW);
  digitalWrite (5,LOW);
```

```
  digitalWrite (6,LOW);
 delay (800);
  digitalWrite(3, HIGH);
  digitalWrite (4, HIGH);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  delay(100);

 }

else if ( readvoice == "*right#")
{
 digitalWrite (3, LOW);
 digitalWrite (4, HIGH);
 digitalWrite (5, LOW);
 digitalWrite (6, LOW);
 delay (800);
  digitalWrite(3, HIGH);
  digitalWrite (4, HIGH);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
  delay(100);
}

else if (readvoice == "*stop#")
{
```

```
  digitalWrite (3, LOW);
  digitalWrite (4, LOW);
  digitalWrite (5, LOW);
  digitalWrite (6, LOW);
  delay (100);
}
else if (readvoice == "*keep watch in all direction#")
{
  digitalWrite (3, HIGH);
  digitalWrite (4, LOW);
  digitalWrite (5, LOW);
  digitalWrite (6, LOW);
  delay (100);
}
 else if (readvoice == "*show me Garba#")
{
digitalWrite (3, LOW);
  digitalWrite (4, HIGH);
  digitalWrite (5, LOW);
  digitalWrite (6, LOW);
  delay (400);
   digitalWrite(3, HIGH);
   digitalWrite (4, HIGH);
   digitalWrite(5,LOW);
   digitalWrite(6,LOW);
   delay(600);
```

```
  digitalWrite (3, LOW);
  digitalWrite (4, HIGH);
  digitalWrite (5, HIGH);
  digitalWrite (6, LOW);
  delay (500);
  digitalWrite (3, HIGH);
  digitalWrite (4, LOW);
  digitalWrite (5, LOW);
  digitalWrite (6, HIGH);
  delay (500);

digitalWrite (3, LOW);
  digitalWrite (4, HIGH);
  digitalWrite (5, LOW);
  digitalWrite (6, LOW);
  delay (400);
   digitalWrite(3, HIGH);
   digitalWrite (4, HIGH);
   digitalWrite(5,LOW);
   digitalWrite(6,LOW);
   delay(600);
   digitalWrite (3, LOW);
  digitalWrite (4, HIGH);
  digitalWrite (5, HIGH);
  digitalWrite (6, LOW);
  delay (500);
```

```
digitalWrite (3, HIGH);
digitalWrite (4, LOW);
digitalWrite (5, LOW);
digitalWrite (6, HIGH);
delay (500);digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
digitalWrite (6, LOW);
delay (400);
 digitalWrite(3, HIGH);
 digitalWrite (4, HIGH);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 delay(600);
 digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, HIGH);
digitalWrite (6, LOW);
delay (500);
digitalWrite (3, HIGH);
digitalWrite (4, LOW);
digitalWrite (5, LOW);
digitalWrite (6, HIGH);
delay (500);digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
```

```
digitalWrite (6, LOW);
delay (400);
 digitalWrite(3, HIGH);
 digitalWrite (4, HIGH);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 delay(600);
 digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, HIGH);
digitalWrite (6, LOW);
delay (500);
digitalWrite (3, HIGH);
digitalWrite (4, LOW);
digitalWrite (5, LOW);
digitalWrite (6, HIGH);
delay (500);digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
digitalWrite (6, LOW);
delay (400);
 digitalWrite(3, HIGH);
 digitalWrite (4, HIGH);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 delay(600);
```

```
 digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, HIGH);
digitalWrite (6, LOW);
delay (500);
digitalWrite (3, HIGH);
digitalWrite (4, LOW);
digitalWrite (5, LOW);
digitalWrite (6, HIGH);
delay (500);digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
digitalWrite (6, LOW);
delay (400);
 digitalWrite(3, HIGH);
 digitalWrite (4, HIGH);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 delay(600);
 digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, HIGH);
digitalWrite (6, LOW);
delay (500);
digitalWrite (3, HIGH);
digitalWrite (4, LOW);
```

```
digitalWrite (5, LOW);
digitalWrite (6, HIGH);
delay (500);digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
digitalWrite (6, LOW);
delay (400);
 digitalWrite(3, HIGH);
 digitalWrite (4, HIGH);
 digitalWrite(5,LOW);
 digitalWrite(6,LOW);
 delay(600);
 digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, HIGH);
digitalWrite (6, LOW);
delay (500);
digitalWrite (3, HIGH);
digitalWrite (4, LOW);
digitalWrite (5, LOW);
digitalWrite (6, HIGH);
delay (500);digitalWrite (3, LOW);
digitalWrite (4, HIGH);
digitalWrite (5, LOW);
digitalWrite (6, LOW);
delay (400);
```

```
    digitalWrite(3, HIGH);
   digitalWrite (4, HIGH);
   digitalWrite(5,LOW);
   digitalWrite(6,LOW);
   delay(600);
   digitalWrite (3, LOW);
  digitalWrite (4, HIGH);
  digitalWrite (5, HIGH);
  digitalWrite (6, LOW);
  delay (500);
  digitalWrite (3, HIGH);
  digitalWrite (4, LOW);
  digitalWrite (5, LOW);
  digitalWrite (6, HIGH);
  delay (500);
}



readvoice="";}} //Reset the variable
```