

```
include <IOXhop_FirebaseESP32.h>

#include <Robojax_L298N_DC_motor.h>

#include "DFRobotDFPlayerMini.h"

DFRobotDFPlayerMini myDFPlayer;


#define FIREBASE_HOST "voicerobot-aa637-default-rtdb.firebaseio.com"

#define FIREBASE_AUTH "KuZnNRqYQjvCDEAdAtTqblpZc4dp6WW5fjPTfljf"

#define WIFI_SSID "Smile King" // input your home or public wifi name

#define WIFI_PASSWORD "sampathgowda"


// motor 1 settings

#define CHA 0

#define ENA 19 // this pin must be PWM enabled pin if Arduino board is used

#define IN1 18

#define IN2 5


// motor 2 settings

#define IN3 33

#define IN4 32

#define ENB 4 // this pin must be PWM enabled pin if Arduino board is used

#define CHB 1


const int CCW = 2; // do not change

const int CW = 1; // do not change
```

```
#define motor1 1 // do not change
```

```
#define motor2 2 // do not change
```

```
// for single motor
```

```
//Robojax_L298N_DC_motor robot(IN1, IN2, ENA, CHA, true);
```

```
Robojax_L298N_DC_motor robot(IN1, IN2, ENA, CHA, IN3, IN4, ENB, CHB);
```

```
const int trigPin = 26;
```

```
const int echoPin = 25;
```

```
int i = 0;
```

```
long duration;
```

```
int distance;
```

```
String status1;
```

```
String Command;
```

```
String lastCommand;
```

```
void setup()
```

```
{
```

```
    // put your setup code here, to run once:
```

```
    Serial.begin(115200);
```

```
    Serial.println("WELCOME");
```

```
    delay(100);
```

```
robot.begin();

WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

Serial.print("connecting");

while (WiFi.status() != WL_CONNECTED)

{

    Serial.print(".");

    delay(500);

}

delay(200);

Serial.println();

Serial.print("connected: ");

Serial.println(WiFi.localIP());

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

delay(200);

Serial2.begin(9600);

delay(500);

Serial.println();

Serial.println(F("DFRobot DFPlayer Mini Demo"));

Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));

myDFPlayer.begin(Serial2);

delay(200);

Serial.println(F("DFPlayer Mini online."));

myDFPlayer.volume(30); //Set volume value. From 0 to 30

delay(100);

myDFPlayer.EQ(1);
```

```
delay(100);  
  
myDFPlayer.play(8); //play specific mp3 in SD:/MP3/0004.mp3; File Name(0~65535)  
  
delay(3000);
```

```
  
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output  
  
pinMode(echoPin, INPUT);  
  
// forward();  
  
// backward();  
  
// left();  
  
// right0();  
  
// stop1();  
  
}
```

```
  
void ultra() {  
  
    digitalWrite(trigPin, LOW);  
  
    delayMicroseconds(2);  
  
    digitalWrite(trigPin, HIGH);  
  
    delayMicroseconds(10);  
  
    digitalWrite(trigPin, LOW);  
  
    duration = pulseIn(echoPin, HIGH);  
  
    distance = duration * 0.034 / 2;  
  
    Serial.print("Distance: ");
```

```
Serial.println(distance);  
  
delay(300);  
  
}
```

```
void stop1() {  
    if (Command.equals(lastCommand))  
    {  
        Serial.println("repeated command");  
    }  
    else  
    {  
        Serial.println("New command");  
        myDFPlayer.play(5);  
        delay(1000);  
    }  
    Serial.println("Motor stopped");  
    robot.brake(1);  
    robot.brake(2);  
    delay(1000);  
  
}
```

```
void readvoice()
{
    Command = Firebase.getString("MOVEMENT");
    Serial.println(Command);
    Command.remove(0, 2);
    Command.remove(4, 7);
    Serial.println(Command);
    delay(100);

    //Serial.println("Moving forward");

    if (Command == "forw") {

        forward();

    }

    if (Command == "back") {

        backward();
```

```
}

if (Command == "left") {

    left();

}

if (Command == "righ") {

    right();

}

if (Command == "morn") {

    if (Command.equals(lastCommand))

    {

        Serial.println("repeated command");

    }

    else

    {

        Serial.println("New command");

        myDFPlayer.play(6);

        delay(1000);

    }

}

}
```

```
if (Command == "afte") {  
    if (Command.equals(lastCommand))  
    {  
        Serial.println("repeated command");  
    }  
    else  
    {  
        Serial.println("New command");  
        myDFPlayer.play(7);  
        delay(1000);  
    }  
  
}  
  
if (Command == "good") {  
    if (Command.equals(lastCommand))  
    {  
        Serial.println("repeated command");  
    }  
    else  
    {  
        Serial.println("New command");  
        myDFPlayer.play(9);  
        delay(1000);  
    }  
}
```



```
}  
  
if (Command == "stop") {  
    Serial.println("Voice Status Cleared");  
    stop1();  
    delay(100);  
  
}  
  
lastCommand = Command;  
}
```

```
void forward()  
{  
    digitalWrite(trigPin, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPin, LOW);  
    duration = pulseIn(echoPin, HIGH);  
    distance = duration * 0.034 / 2;  
    Serial.print("Distance: ");  
    Serial.println(distance);  
    delay(100);  
}
```

```
if (distance < 20)
{
    stop1();
    myDFPlayer.play(10);
    delay(1000);

}

else if (distance > 20 && distance < 50)
{
    if (Command.equals(lastCommand))
    {
        Serial.println("repeated command");
    }
    else
    {
        Serial.println("New command");
        myDFPlayer.play(1);
        delay(2000);
    }

    robot.rotate(motor1, 50, CW);
    robot.rotate(motor2, 50, CW);
    delay(1000);
//  robot.brake(1);
//  robot.brake(2);
```

```
// delay(1000);

}

else

{

    if (Command.equals(lastCommand))

    {

        Serial.println("repeated command");

    }

    else

    {

        Serial.println("New command");

        myDFPlayer.play(1);

        delay(2000);

    }


    robot.rotate(motor1, 100, CW);

    robot.rotate(motor2, 100, CW);

    delay(1000);

    // robot.brake(1);

    // robot.brake(2);

    // delay(1000);

}

}

void backward()
```

```
{  
  if (Command.equals(lastCommand))  
  {  
    Serial.println("repeated command");  
  }  
  else  
  {  
    Serial.println("New command");  
    myDFPlayer.play(2);  
    delay(1000);  
  }  
  Serial.println("Backward ");  
  robot.rotate(motor1, 80, CCW);  
  robot.rotate(motor2, 80, CCW);  
  delay(1000);  
  // robot.brake(1);  
  // robot.brake(2);  
  // delay(1000);  
}
```

```
void right()  
{  
  if (Command.equals(lastCommand))  
  {
```

```
    Serial.println("repeated command");
}
else
{
    Serial.println("New command");
    myDFPlayer.play(4);
    delay(1000);
}
Serial.println("right");

robot.rotate(motor1, 90, CW);
robot.rotate(motor2, 90, CCW);
delay(500);
robot.brake(1);
robot.brake(2);
delay(1000);
Firebase.setString("MOVEMENT","None");
delay(200);

}

void left()
{
    if (Command.equals(lastCommand))
    {
```

```
    Serial.println("repeated command");
}
else
{
    Serial.println("New command");
    myDFPlayer.play(3);
    delay(1000);
}
Serial.println("left");

robot.rotate(motor1, 90, CCW);
robot.rotate(motor2, 90, CW);
delay(500);
robot.brake(1);
robot.brake(2);
delay(1000);
Firebase.setString("MOVEMENT","None");
delay(200);

}

void loop()
{
    // ultra();
    readvoice();
}
```

}