# Instruction

## Parameter:

Voltage: 4.5-5.5V

Current: <40mA

Digital interface: 5V TTL level UART interface

Analog input: 3.5mm mono microphone connector + microphone pin connector

Size: 30mm x 47.5mm

This module can store 15 voice commands. These 15 pieces are divided into 3 groups, 5 pieces in each group. First, we should record the voice commands group by group. After that, before it can recognize the 5 voice commands in the group, import a command group through serial commands. If we need to implement instructions from other groups, we should first import the required instruction groups. This module has an independent speaker. If your friend enters commands by voice instead of your voice commands, it may not recognize the commands.

## Example 1

Here, I give an example to show how to use language to control the light to emit red, green and blue.

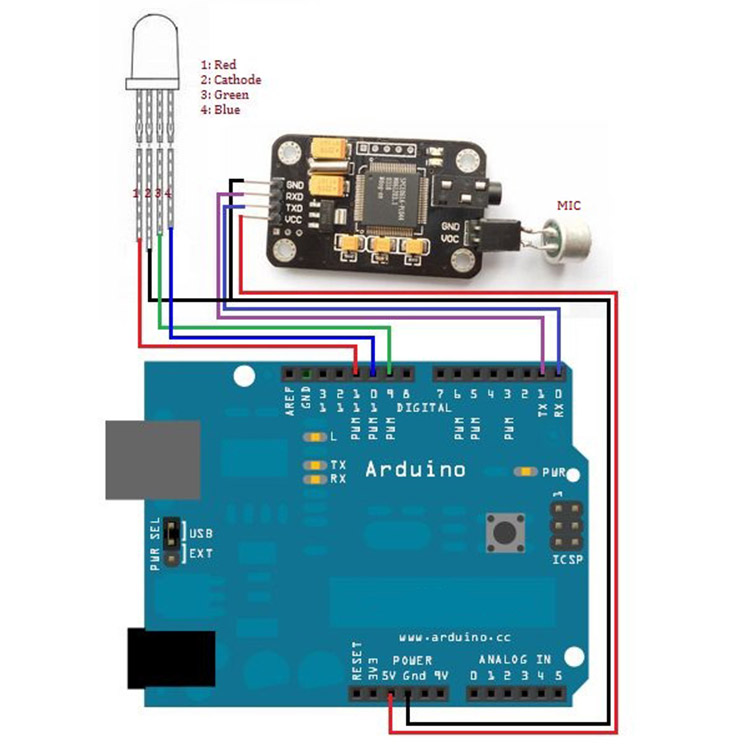
### 1. Recording

We need to send a serial command to the module. You may need a USB-TTL module to connect it to the PC. Then send the command 0xaa11 to record. For more detailed information, please refer to the product manual.

Please record the following voice instrctions in order:

* **WHITE**
* **RED**
* **GREEN**
* **BLUE**
* **OFF**

### Hardware connection



The figure above shows the connection diagram of the V1 version of the product. For the V2 version, the connection method is the same.

### Code

int redPin = 11; // R petal on RGB LED module connected to digital pin 11  
int greenPin = 9; // G petal on RGB LED module connected to digital pin 9  
int bluePin = 10; // B petal on RGB LED module connected to digital pin 10  
byte com = 0; //reply from voice recognition

void setup()  
{  
Serial.begin(9600);  
pinMode(ledPin, OUTPUT); // sets the ledPin to be an output  
pinMode(redPin, OUTPUT); // sets the redPin to be an output  
pinMode(greenPin, OUTPUT); // sets the greenPin to be an output  
pinMode(bluePin, OUTPUT); // sets the bluePin to be an output  
delay(2000);  
Serial.write(0xAA);  
Serial.write(0x37);  
delay(1000);  
Serial.write(0xAA);  
Serial.write(0x21);  
}

void loop() // run over and over again  
{

while(Serial.available())  
{  
com = Serial.read();  
switch(com)  
{  
case 0x11:  
color(255,255,255); // turn RGB LED on -- white  
break;

case 0x12:  
color(255, 0, 0); // turn the RGB LED red  
break;

case 0x13:  
color(0,255, 0); // turn the RGB LED green  
break;

case 0x14:  
color(0, 0, 255); // turn the RGB LED blue  
break;

case 0x15:  
color(0,0,0); // turn the RGB LED off  
break;

}  
}

}

void color (unsigned char red, unsigned char green, unsigned char blue) // the color generating function  
{  
analogWrite(redPin, red\*102/255);  
analogWrite(bluePin, blue\*173/255);  
analogWrite(greenPin, green\*173/255);  
}

Load the above code into Arduino. Please disconnect TX and RX when loading the code, otherwise the serial port may be damaged.

1. video

After the code is loaded, connect RX and TX, and press the Arduino RESET button.

## Example 2

Here, we will tell you how to use GPIO output to control other devices.

### Step 1

First you have to record the voice command. Send the command 0xAA12 (group 2).

Please record the following 5 types of voice commands in the order given:

"One"

"Two"

"Three"

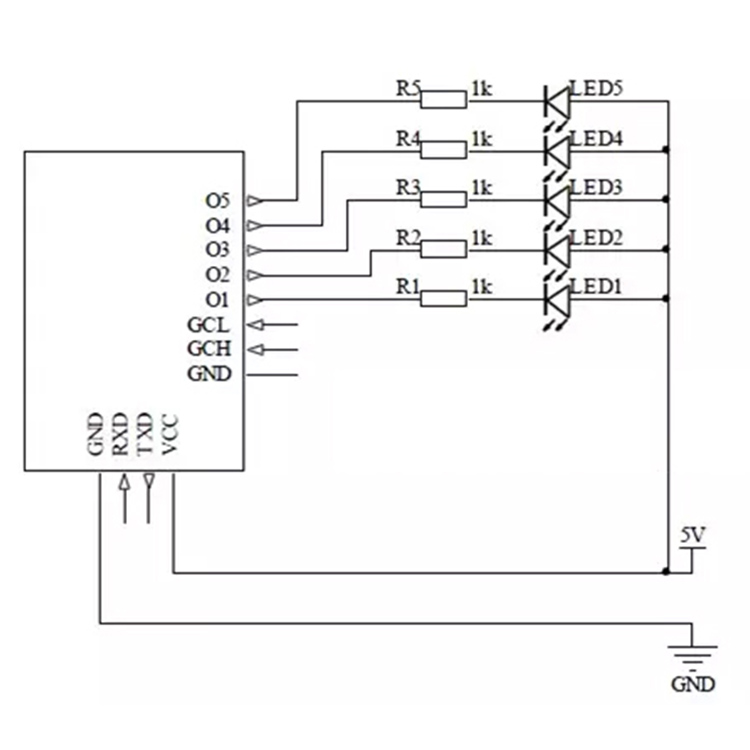
"Four"

"Five"

Please note that the pronunciation must be clear.

### Step 2

Connect mobile devices with LEDs in the following ways:



### Step 3

Use the command 20xAA2 to import the second group of voice command group commands, or set the GCH pin high and the GCL pin low.

Speak the voice command.

Now, you can control the LED by voice.