

# Voice Control RGB Light

## Voice Control RGB Light

[Experimental Objective](#)

[Experimental Procedure and Results](#)

[Main source code analysis](#)

## Experimental Objective

Understand and learn to use basic semantics to control the on/off behavior of the RGB lights on the robot.

## Experimental Procedure and Results

1. First, enter the following command in the terminal:

```
cd /home/pi/project_demo/10.Basic_voice_control/2.Speech_control_RGB/  
python3 speech_control_RGB.py
```

2. After entering this interface, use the wake-up phrase: **Hi, Yahboom** in English.

```
pi@yahboom:~/project_demo/10.Basic_voice_control/2.Speech_control_RGB $ python3 speech_control_RGB.py  
B.py  
Speech Serial Opened! Baudrate=115200
```

3. After successfully waking up, the English version responds: "Hi, I'm here." Then, use fixed statements to control the car.

### Fixed Statement Table

Wake-up Statement (International Users)	Operational Status	Car Response (English Version)
Car RGB light off	OK, light is off.	OK, light is closed.
Car RGB light turns red	OK, red light is on.	OK, red light is on.
Green light on	Car RGB light turns green	OK, green light is on.
Blue light up	The car's RGB light turns blue	OK, blue light is on.
Yellow light up	The car's RGB light turns yellow	OK, yellow light is on.
light A	The car's RGB light displays a running light effect	OK, light A is on.
light B	The car's RGB light displays a gradient light effect	OK, light B is on.
light C	The car's RGB light displays a breathing light effect	OK, light C is on.

## Main source code analysis

```

try:
    while 1:
        time.sleep(0.2)
        num = mySpeech.speech_read()
        if num !=999 and num !=0:
            #print(num)
            mySpeech.void_write(num)
            Scommand = num

            if Scommand == 10:
                bot.Ctrl_WQ2812_ALL(0,0)
                close_light = 1

            else:
                close_light = 0
                action_thread = threading.Thread(target=start_action)
                action_thread.daemon = True
                action_thread.start()

        num =999
except:
    mySpeech.__del__()
    print("serial close")

```

**start\_action:** Executes RGB color control based on fixed semantic recognition

**bot.Ctrl\_WQ2812\_ALL(0,0):** Turns off the RGB light

**speech\_read:** Reads the recognized message

**void\_write:** Selects the message sequence number to be fed back by the robot's speaker. This can be found in the protocol table.

**run\_river\_light(0.03)** Flowing water lighting effect. Parameter is the speed. Here, the default is 0.03s.

**gradient\_light(0.08)** Gradient lighting effect. Parameter is the speed. Here, it is 0.08.

**breathing\_light(0.03,4)** Breathing lighting effect. The first parameter is the speed, and the second parameter is the breathing light color. Here, the speed is 0.03s, purple.

The second parameter corresponds to the color:

0: Red 1: Green 2: Blue 3: Yellow 4: Purple 5: Cyan 6: White