

## Chapter7: Control GPIO with BST-AI expansion board

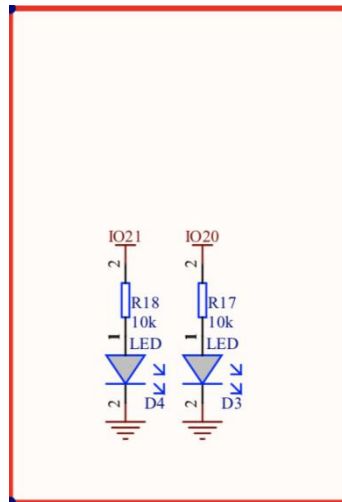


Figure 1-1 Diode

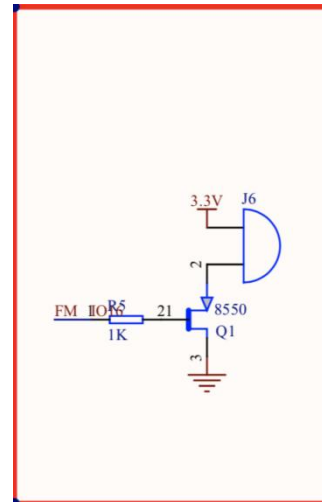


Figure 1-2 Buzzer

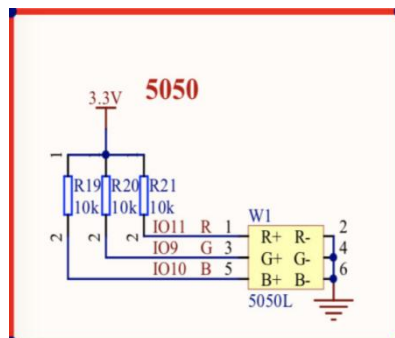


Figure 1-3 RGB

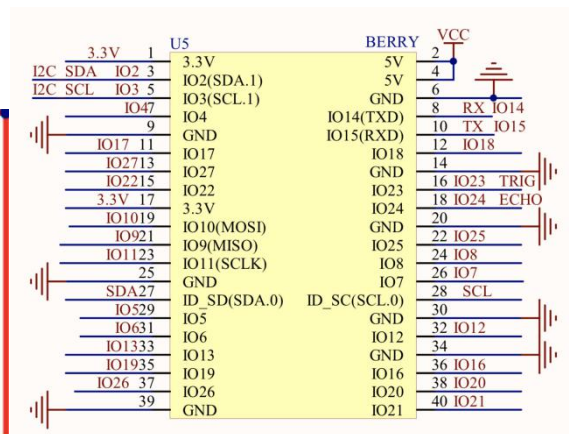


Figure 1-4 Raspberry Pi Pin Diagram

1. The first experiment is to make the blue and red LED lights flash alternately. According to the circuit diagram, we can know that the two LED small lights are controlled by GPIO20, 21.

Introduction to the way Python controls the Raspberry Pi GPIO:

- 1) we need to introduce the RPi.GPIO library file.
- 2) we need to- set the working mode of the Raspberry Pi GPIO. The parameters are BCM and BOARD. The BCM represents the GPIO number declaration of the pin, and the BOARD represents the pin number. It is recommended that users use BCM as the mode of declaration of working mode.
- 3) We need to declare the input and output of the specified pin with the .setup() function.
- 4) We need to change the level of the output pin to control the LED.
- 5) Finally, we need to release the working status with GPIO.cleanup(). Otherwise it will affect the next use.

The source code of the program is located at: </home/pi/yahboom/GPIO/led.py>

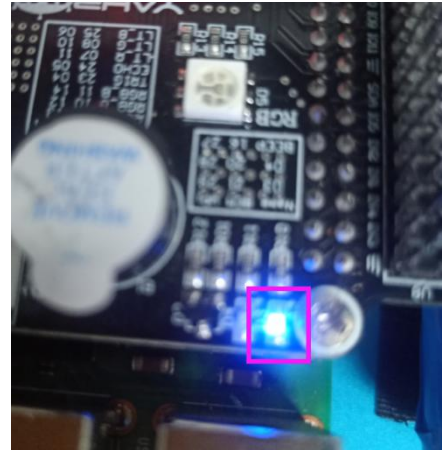
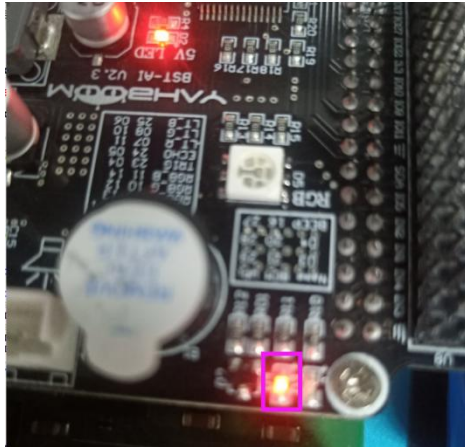
The code as shown in the figure below.

```

1  #!/usr/bin/env python2
2  # -*- coding: utf-8 -*-
3  """
4      Created on Tue Nov  6 01:18:45 2018
5      * @par Copyright (c): 2010-2019, Shenzhen Yahboom Tech
6      * @file          led-buzzer
7      * @version       V1.0
8      * @details
9      * @par History
10
11      @author: longfuSun
12  """
13  import RPi.GPIO as GPIO
14  import time
15  #Control LED pin is 9,10,11;buzzer pin is 16
16  R,G,B=9,10,11
17  buzzer=16
18  GPIO.setmode(GPIO.BCM)
19
20  GPIO.setup(R, GPIO.OUT)
21  GPIO.setup(G, GPIO.OUT)
22  GPIO.setup(B, GPIO.OUT)
23  GPIO.setup(buzzer, GPIO.OUT)
24  #Make buzzer no sound
25
26  GPIO.output(buzzer, False)
27  time.sleep(2)
28  GPIO.output(buzzer, True)
29  #Set work mode of PWM
30  pwmR = GPIO.PWM(R, 70)
31  pwmG = GPIO.PWM(G, 70)
32  pwmB = GPIO.PWM(B, 70)
33
34  pwmR.start(0)
35  pwmG.start(0)
36  pwmB.start(0)
37
38  #Four modes
39  try:
40      t = 0.01
41      while True:
42          for i in range(0,71):
43              pwmG.ChangeDutyCycle(70)
44              pwmB.ChangeDutyCycle(i)
45              pwmR.ChangeDutyCycle(70-i)
46              print(i)
47              time.sleep(t)
48          for i in range(70,-1,-1):
49              pwmG.ChangeDutyCycle(0)
50              pwmB.ChangeDutyCycle(i)
51              pwmR.ChangeDutyCycle(70-i)
52              print(i-1000)
53              time.sleep(t)
54
55  except KeyboardInterrupt:
56      pass
57  pwmR.stop()
58  pwmG.stop()
59  pwmB.stop()
60  GPIO.cleanup()

```

The result is as shown in the figure below.



2. In second experiment, we will use the PWM mode of operation to control the LED lights. The program provides a method for controlling the LED lights by pulse modulation and demodulation. The program will realize a breathing light to make the RGB lights, and the color of the gradient between red, green and blue.

The source code of the program is located at: [/home/pi/yahboom/GPIO/BST-AI.py](#)

The code as shown in the figure below:

```
1 #!/usr/bin/env python2
2 # -*- coding: utf-8 -*-
3 """
4     Created on Tue Nov  6 01:18:45 2018
5     * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
6     * @file      led-buzzer
7     * @version    V1.0
8     * @details
9     * @par History
10
11     @author: longfuSun
12 """
13 import RPi.GPIO as GPIO
14 import time
15 #Control LED pin is 9,10,11;buzzer pin is 16
16 R,G,B=9,10,11
17 buzzer=16
18 GPIO.setmode(GPIO.BCM)
19
20 GPIO.setup(R, GPIO.OUT)
21 GPIO.setup(G, GPIO.OUT)
22 GPIO.setup(B, GPIO.OUT)
23 GPIO.setup(buzzer, GPIO.OUT)
24 #Make buzzer no sound
25
26 GPIO.output(buzzer, False)
27 time.sleep(2)
28 GPIO.output(buzzer, True)
29 #Set work mode of PWM
30 pwmR = GPIO.PWM(R, 70)
31 pwmG = GPIO.PWM(G, 70)
32 pwmB = GPIO.PWM(B, 70)
33
34 pwmR.start(0)
35 pwmG.start(0)
36 pwmB.start(0)
```

```
37 #Four modes
38 try:
39     t = 0.01
40     while True:
41         for i in range(0,71):
42             pwmG.ChangeDutyCycle(70)
43             pwmB.ChangeDutyCycle(i)
44             pwmR.ChangeDutyCycle(70-i)
45             print(i)
46             time.sleep(t)
47         for i in range(70,-1,-1):
48             pwmG.ChangeDutyCycle(0)
49             pwmB.ChangeDutyCycle(i)
50             pwmR.ChangeDutyCycle(70-i)
51             print(i-1000)
52             time.sleep(t)
53
54
55 except KeyboardInterrupt:
56     pass
57 pwmR.stop()
58 pwmG.stop()
59 pwmB.stop()
60 GPIO.cleanup()
```