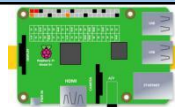



Lesson 7 Light up the LED

7.1 Overview

This lesson mainly focuses on the control of the Adeept Robot HAT's onboard LED. During the learning process, learners will understand the required components, circuit principles, and wiring methods, and learn to write control codes using Python in combination with the gpiozero library, and remotely operate on the Raspberry Pi terminal to achieve sequential flashing of the LEDs.

7.2 Required Components

Components	Quantity	Picture
Raspberry Pi	1	
Adeept Robot HAT V3.2	1	

7.3 Principle Introduction

The General-Purpose Input/Output (GPIO) pins of the Raspberry Pi are used to control three Light Emitting Diodes (LEDs). In terms of hardware connection, these three LEDs are respectively connected to GPIO pins 9, 25, and 11 of the Raspberry Pi. LEDs are polarized. The longer pin (the positive terminal) is connected to the positive output terminal of the power supply, and the shorter pin (the negative terminal) is grounded. In addition, each LED is connected in series with a suitable resistor to limit the current and ensure its safe operation.

When the program runs, it first initializes the GPIO pins connected to the LEDs to clarify the control tasks. Subsequently, it controls each LED in sequence. When the program issues a

lighting instruction, the corresponding pin of the Raspberry Pi outputs a high level. Then, the current passes through the LED and the resistor, and the LED lights up and remains on. In this way, the second and third LEDs are lit up one by one.

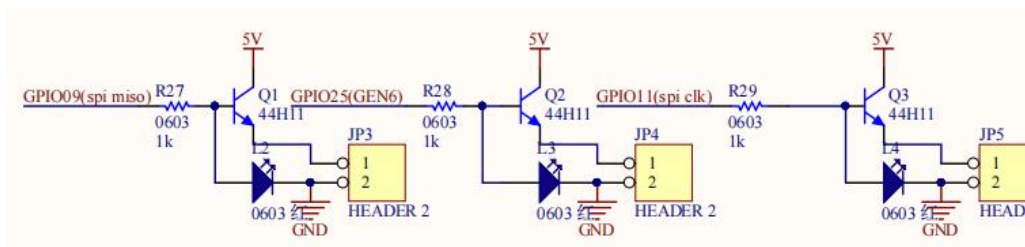
After the three LEDs are all lit up and remain on for the corresponding duration, the program makes the pins of the Raspberry Pi output a low level, cutting off the power supply, and the LEDs go out. Then, there is a one-second delay. The entire process repeats continuously, achieving the effect of the three LEDs lighting up and going out regularly in sequence.

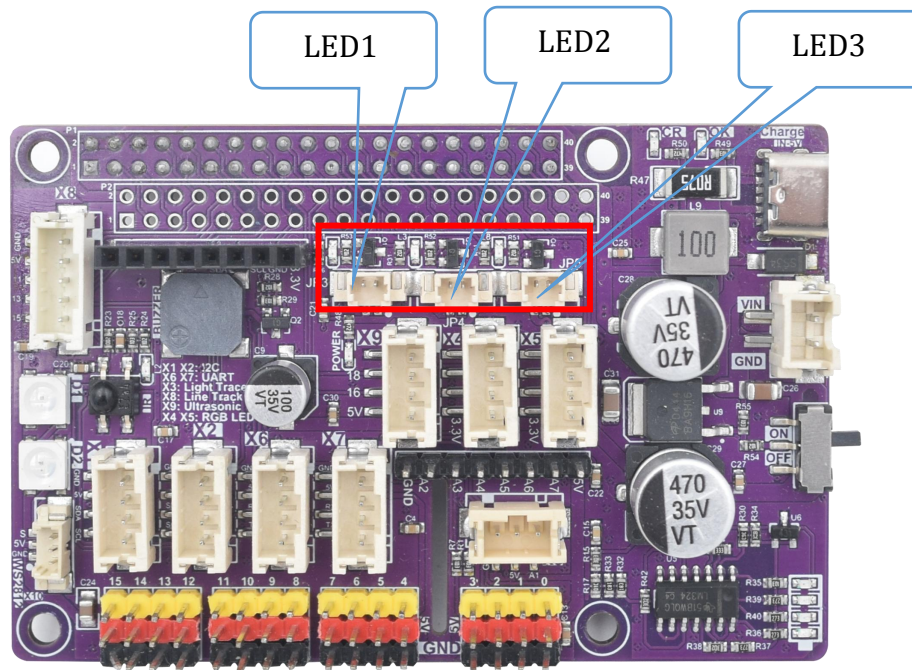
On the Adeept Robot HAT V3.2, the onboard LEDs and external LEDs are controlled by the same pin. According to the control instructions transmitted by this pin, they synchronously perform corresponding actions such as lighting up, turning off, or dimming.

PINS of Raspberry Pi	LED number
GPIO9	LED1
GPIO25	LED2
GPIO11	LED3

7.4 Wiring Diagram

The motherboard of the Adeept Robot HAT V3.2 is equipped with 3 LED lights, which are called onboard LEDs. These 3 onboard LEDs are respectively connected to the GPIO9, GPIO25, and GPIO11 pins. Each of these three pins is connected in parallel with a switch.





7.5 Light up LED

1. **Remotely log:** Remotely log in to the Raspberry Pi terminal.
2. **Navigate to the Program Folder:** Enter the following command in the terminal and press Enter to access the folder where the program is located:

```
cd Adeept_4WD_Smart_Car_for_RPi/Examples/01_LED/
```

```
pi@raspberrypi:~ $ cd Adeept_4WD_Smart_Car_for_RPi/Examples/01_LED/
pi@raspberrypi:~/Adeept_4WD_Smart_Car_for_RPi/Examples/01_LED $
```

3. **View Directory Contents:** Type "ls" in the terminal and press Enter. This will display all the files in the current directory, ensuring that the "LED.py" file is present:

```
ls
```

```
pi@raspberrypi:~/Adeept_4WD_Smart_Car_for_RPi/Examples/01_LED $ ls
LED.py
```

4. **Run the Program:** Enter the command below and press Enter to start the LED control program:

```
sudo python3 LED.py
```

```
pi@raspberrypi:~/Adeept_4WD_Smart_Car_for_RPi/Examples/01_LED $ sudo python3 LED.py
LED1 on
LED2 on
LED3 on
All LED off
LED1 on
LED2 on
LED3 on
```

5. Observation and Termination:After successfully running the program, you will see three LEDs light up in turn, and when you want to terminate the running program, you can press the "**Ctrl+C**" shortcut key on the keyboard.

7.6 Code

Complete code refer to [LED.py](#)

```
01  #!/usr/bin/env/python
02  # File name   : LED.py
03  # Website    : www.Adeept.com
04  # Author     : Adeept
05  # Date      : 2025/03/5
06
07  import time
08  from gpiozero import LED
09
10  def switchSetup():
11      global led1,led2,led3
12      led1 = LED(9)
13      led2 = LED(25)
14      led3 = LED(11)
15
16  def switch(port, status):
17      if port == 1:
18          if status == 1:
19              led1.on()
20          elif status == 0:
21              led1.off()
22      elif port == 2:
23          if status == 1:
24              led2.on()
25          elif status == 0:
26              led2.off()
27      elif port == 3:
28          if status == 1:
29              led3.on()
30          elif status == 0:
31              led3.off()
32      else:
```

```
33     print('Wrong Command: Example--switch(3, 1)->to switch on port3')
34
35     def set_all_switch_off():
36         switch(1,0)
37         switch(2,0)
38         switch(3,0)
39
40
41     if __name__ == "__main__":
42         switchSetup()
43         while True:
44             switch(1,1)
45             print("LED1 on")
46             time.sleep(1)
47             switch(2,1)
48             print("LED2 on")
49             time.sleep(1)
50             switch(3,1)
51             print("LED3 on")
52             time.sleep(1)
53             set_all_switch_off()
54             print("All LED off")
55             time.sleep(1)
```

Code explanation

Initialization Stage:

Call the switchSetup() function to complete the hardware initialization.

Loop Control Process:

After entering an infinite loop, execute the following steps in sequence:

Stage 1: Turn on LED1 → Delay for 1 second

Stage 2: Turn on LED2 → Delay for 1 second

Stage 3: Turn on LED3 → Delay for 1 second

Stage 4: Turn off all LEDs → Delay for 1 second

A cyclic effect is formed where the LEDs are turned on one by one and then all turned off collectively.