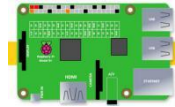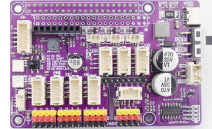# Lesson 5 How to Control Onboard LED

In this lesson, we will learn how to control Onboard LED.

5.1 Components Used in This Course

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

5.2 Introduction of Onboard LED

Adeept Robot HAT V3.1 comes with 3 LEDs on the motherboard, which are called onboard LEDs. The 3 onboard LEDs are connected to GPIO9, GPIO25 and GPIO11 pins respectively. These three pins are each connected in parallel with a switch. If other devices are connected to the switch, the onboard LED lights and LED interface devices will be controlled by the program at the same time.

support@adeept.com
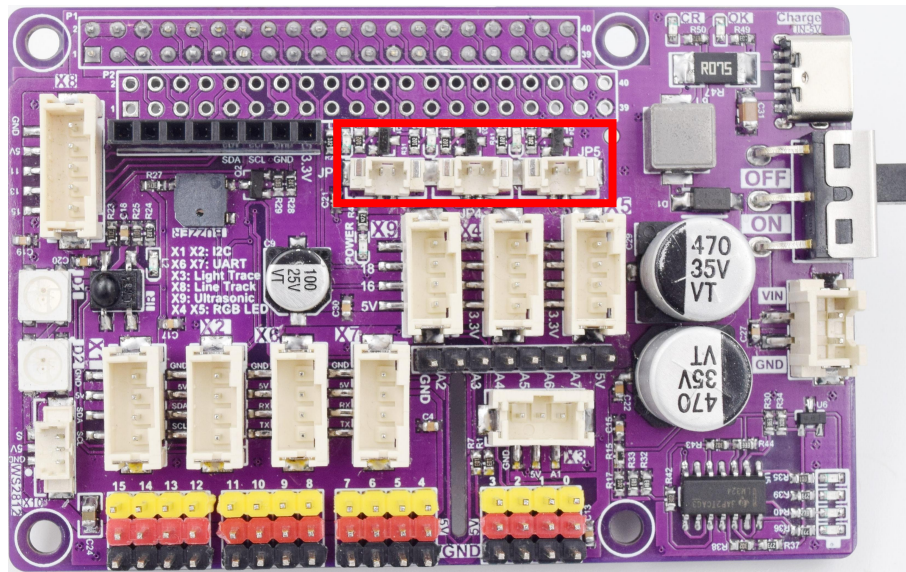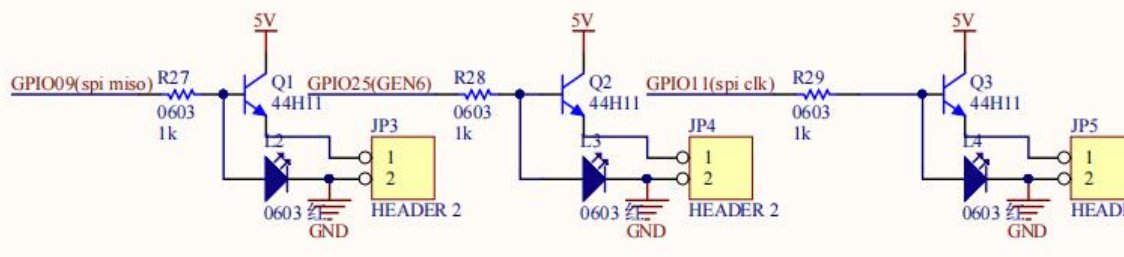
5.2 How to Control Onboard LED

Run the code

1. Remotely log in to the Raspberry Pi terminal.


```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Aug 29 08:17:49 2020 from 192.168.3.208

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.

pi@raspberrypi:~ $
```

support@adeept.com

2. Enter the command and press Enter to enter the folder where the program is located:

cd adeept_rasptank2/examples/

```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ cd adeept_rasptank2/examples/
pi@raspberrypi:~/adeept_rasptank2/examples $
pi@raspberrypi:~/adeept_rasptank2/examples $
```

3. View the contents of the current directory file:

ls

```
pi@raspberrypi:~/adeept_rasptank2/examples $ ls
01_LED.py      03_Servo.py   05_Spi_WS2812.py   06_Ultrasonic.py
02_Buzzer.py   04_Motor.py   05_WS2812.py       07_TrackingLine.py
```

4. Enter the command and press Enter to run the program:

Older versions of the Raspberry Pi operating system include Python2 and python3.

In order for the program to run properly in older versions of Raspberry Pi OS, use "python3" to run the program. Of course, in new versions of Raspberry Pi OS, you can also use the "python" command to run programs.

In order to prevent users from misunderstanding, this textbook uses the "**python3**" command to run the program.

sudo python3 01_LED.py

```
pi@raspberrypi:~/adeept_rasptank2/examples $ sudo python3 01_LED.py
LED1 on
LED2 on
LED3 on
All LED off
LED1 on
```

5. After running the program successfully, you will see three LEDs light up in turn.

6. When you want to terminate the running program, you can press the shortcut key "Ctrl + C" on the keyboard.

5.4 Code

Complete code refer to 01_LED.py.

Control the servo to rotate to a certain angle

```python
1.  #!/usr/bin/env/python3
2.  import time
3.  from gpiozero import LED
4.
5.  def switchSetup():
6.      global led1,led2,led3
7.      led1 = LED(9)
8.      led2 = LED(25)
9.      led3 = LED(11)
10.
11. def switch(port, status):
12.     if port == 1:
13.         if status == 1:
14.             led1.on()
15.         elif status == 0:
16.             led1.off()
17.     elif port == 2:
18.         if status == 1:
19.             led2.on()
20.         elif status == 0:
21.             led2.off()
22.     elif port == 3:
23.         if status == 1:
24.             led3.on()
25.         elif status == 0:
26.             led3.off()
27.     else:
28.         print('Wrong Command: Example--switch(3, 1)->to switch on port3')
29.
30. def set_all_switch_off():
31.     switch(1,0)
32.     switch(2,0)
33.     switch(3,0)
34.
```

```
35.
36. if __name__ == "__main__":
37.     switchSetup()
38.     while True:
39.         switch(1,1)
40.         print("LED1 on")
41.         time.sleep(1)
42.         switch(2,1)
43.         print("LED2 on")
44.         time.sleep(1)
45.         switch(3,1)
46.         print("LED3 on")
47.         time.sleep(1)
48.         set_all_switch_off()
49.         print("All LED off")
50.         time.sleep(1)
```