

## Lesson 4 View and Edit Code on the Raspberry Pi

In order to make the daily use of the Raspberry Pi more convenient, we usually choose not to connect peripherals such as mice, keyboards, and monitors to the Raspberry Pi. This is because our Raspberry Pi is installed inside the robot. If we frequently rely on peripherals to control it, the efficiency of programming and testing will be severely affected. For example, the process of setting up and connecting these peripherals can be time-consuming, and it may also limit the flexibility of operation. Therefore, we are going to introduce a method of programming for the Raspberry Pi, which can help improve the programming and testing efficiency.

### 4.1 PuTTY

- A third-party software [Putty](#) will be used for this method. For detailed instructions, refer to [3.5.2 Remotely log in to the Raspberry Pi OS](#).
- [Putty](#) is a terminal software to remotely control Raspberry Pi (need to enable SSH).

### 4.2 View and Edit Code

1. Run Putty, type in the IP address of your Raspberry Pi and click open. Enter your user name and password – with the initial user name: [pi](#), password: [raspberrypi](#).
2. Type in "ls" to show the files to be edited or viewed.

```
pi@raspberrypi:~ $ ls
adeept_picar-b2      Adeept_RaspClaws-V3  create_ap  Downloads  Pictures  Templates
adeept_raspclaws     Adeept_RaspTank_Metal Desktop    image.jpg  Public    Videos
Adeept_RaspClaws-20250620 Bookshelf             Documents  Music      startup.sh
pi@raspberrypi:~ $
```

3. Type in "**sudo nano <file path>**" to view or edit a file, as shown below:

```
sudo nano Adeept_RaspClaws-V3/Examples/01_LED/LED.py
pi@raspberrypi:~ $ sudo nano Adeept_RaspClaws-V3/Examples/01_LED/LED.py
pi@raspberrypi:~ $
```

```

pi@raspberrypi: ~
GNU nano 7.2 Adeept RaspClaws/Examples/01_LED/LED.py
#!/usr/bin/env/python
# File name      : LED.py
# Website       : www.Adeept.com
# Author        : Adeept
# Date          : 2025/04/8

import time
from gpiozero import LED

def switchSetup():
    global led1,led2,led3
    led1 = LED(9)
    led2 = LED(25)
    led3 = LED(11)

def switch(port, status):
    if port == 1:
        if status == 1:
            led1.on()
        elif status == 0:
            led1.off()
    elif port == 2:
        if status == 1:
            led2.on()
        elif status == 0:
            led2.off()
    elif port == 3:
        if status == 1:
            led3.on()
        elif status == 0:
            led3.off()
    else:
        print('Wrong Command: Example--switch(3, 1)->to switch on port3')

def set_all_switch_off():
    switch(1,0)
    switch(2,0)
    switch(3,0)

if __name__ == "__main__":
    switchSetup()
    while True:
        switch(1,1)
        print("LED1 on")
        time.sleep(1)
        switch(2,1)
        print("LED2 on")
        time.sleep(1)
        switch(3,1)
        print("LED3 on")
        time.sleep(1)
        set_all_switch_off()
        print("All LED off")

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line

```

4. Exit. Press **Ctrl + X** on your keyboard to exit if nothing is changed. Otherwise press **Y** and **Enter** to save and exit.