

Lesson 4 How to View or Edit Code on a Raspberry Pi

For convenient use, we usually do not connect peripherals such as mouse, keyboard, and monitor to the Raspberry Pi. Since the Raspberry Pi is installed inside this robot, often with peripherals, the efficiency of programming and testing will be greatly affected. To achieve better results, here we introduce a method of programming the Raspberry Pi.

4.1 For Mac

- A third-party software [Putty](#) will be used for this method. For detailed instructions, refer to [3.6.2 Remotely logging in to the Raspberry Pi system](#).
- [Putty](#) is a terminal software to remotely control Raspberry Pi (need to enable SSH).

Operation Steps

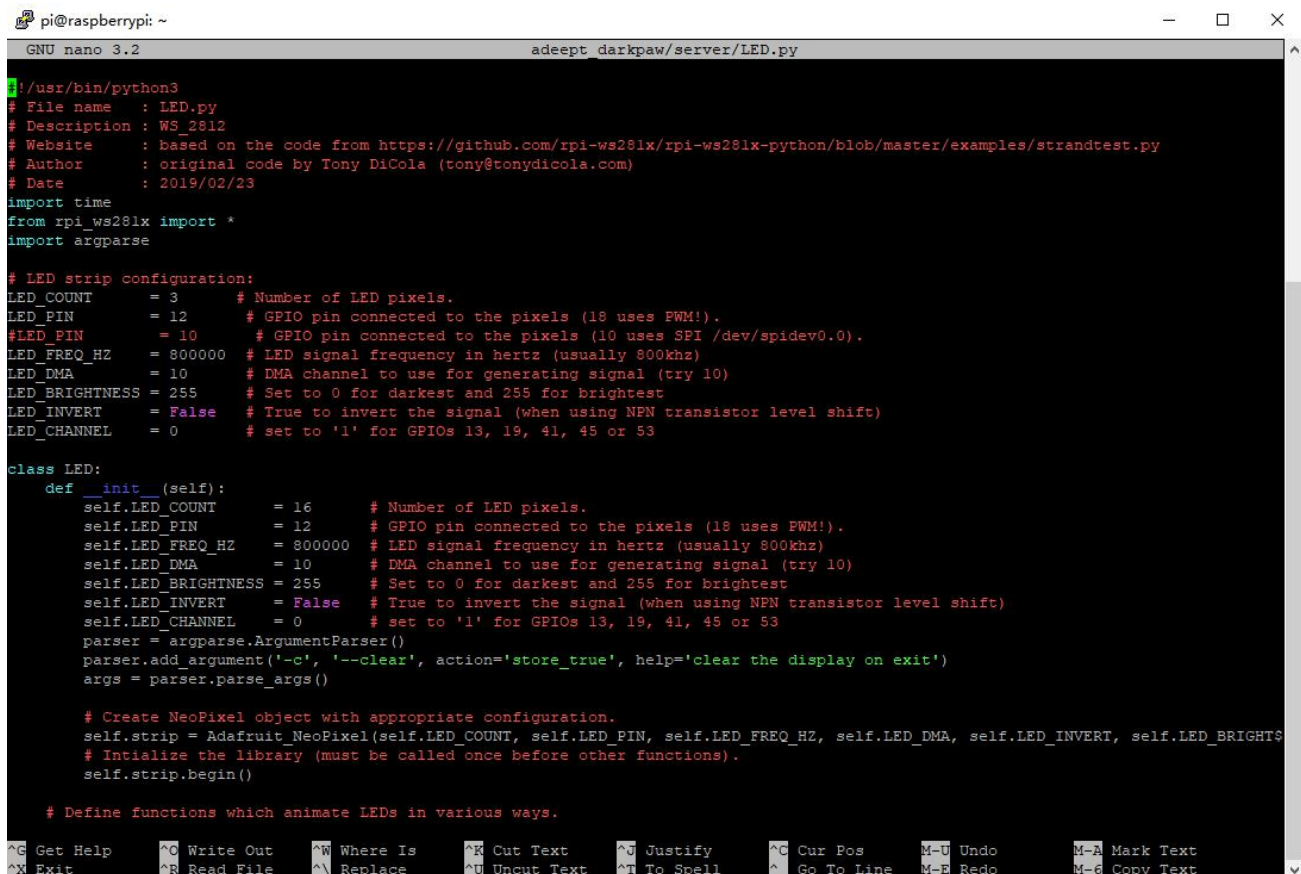
1. Run Putty, type in the IP address of your Raspberry Pi and click open. Enter your user name and password – with the initial user name: [pi](#), password: [raspberrypi](#).
2. Type in "ls" to show the files to be edited or viewed.

```
pi@raspberrypi:~$ ls
adeept_AWR      create_ap  Pictures      sphinxbase-5prealpha
adeept_darkpaw Desktop    pocketsphinx-5prealpha sphinxbase.tar.gz
adeept_picar-b Documents  pocketsphinx.tar.gz  startup.sh
adeept_raspclaws Downloads Public      Templates
Bookshelf       Music     rpi-backup    Videos
pi@raspberrypi:~$
```

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

```
sudo nano adept_darkpaw/server/LED.py
```

```
pi@raspberrypi:~$
pi@raspberrypi:~$ sudo nano adept_darkpaw/server/LED.py
```



```
GNU nano 3.2 adept_darkpaw/server/LED.py
#!/usr/bin/python3
# File name : LED.py
# Description : WS_2812
# Website : based on the code from https://github.com/rpi-ws281x/rpi-ws281x-python/blob/master/examples/strandtest.py
# Author : original code by Tony DiCola (tony@tonydicola.com)
# Date : 2019/02/23
import time
from rpi_ws281x import *
import argparse

# LED strip configuration:
LED_COUNT = 3 # Number of LED pixels.
LED_PIN = 12 # GPIO pin connected to the pixels (18 uses PWM!).
#LED_PIN = 10 # GPIO pin connected to the pixels (10 uses SPI /dev/spidev0.0).
LED_FREQ_HZ = 800000 # LED signal frequency in hertz (usually 800khz)
LED_DMA = 10 # DMA channel to use for generating signal (try 10)
LED_BRIGHTNESS = 255 # Set to 0 for darkest and 255 for brightest
LED_INVERT = False # True to invert the signal (when using NPN transistor level shift)
LED_CHANNEL = 0 # set to '1' for GPIOs 13, 19, 41, 45 or 53

class LED:
    def __init__(self):
        self.LED_COUNT = 16 # Number of LED pixels.
        self.LED_PIN = 12 # GPIO pin connected to the pixels (18 uses PWM!).
        self.LED_FREQ_HZ = 800000 # LED signal frequency in hertz (usually 800khz)
        self.LED_DMA = 10 # DMA channel to use for generating signal (try 10)
        self.LED_BRIGHTNESS = 255 # Set to 0 for darkest and 255 for brightest
        self.LED_INVERT = False # True to invert the signal (when using NPN transistor level shift)
        self.LED_CHANNEL = 0 # set to '1' for GPIOs 13, 19, 41, 45 or 53
        parser = argparse.ArgumentParser()
        parser.add_argument('-c', '--clear', action='store_true', help='clear the display on exit')
        args = parser.parse_args()

        # Create NeoPixel object with appropriate configuration.
        self.strip = Adafruit_NeoPixel(self.LED_COUNT, self.LED_PIN, self.LED_FREQ_HZ, self.LED_DMA, self.LED_INVERT, self.LED_BRIGHTNESS)
        # Initialize the library (must be called once before other functions).
        self.strip.begin()

        # Define functions which animate LEDs in various ways.
```

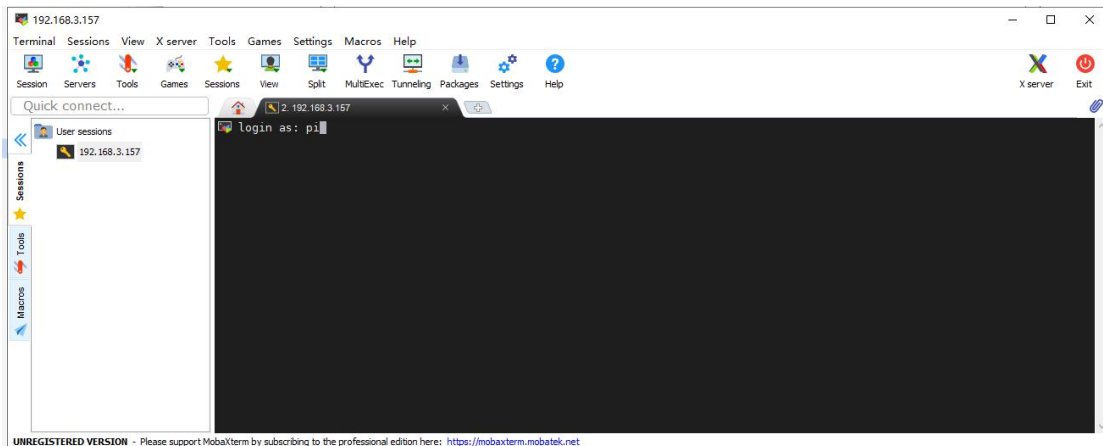
4. Exit. Press Ctrl + X on your keyboard to exit if nothing is changed. Otherwise press Y and Enter key to save and exit.

4.2 For Windows

- A third-party software [MobaXterm](#) will be used for this method. For detailed instructions, refer to [3.6.2 Remotely logging in to the Raspberry Pi system](#).
- [MobaXterm](#) is a terminal software to remotely control Raspberry Pi (need to enable SSH).

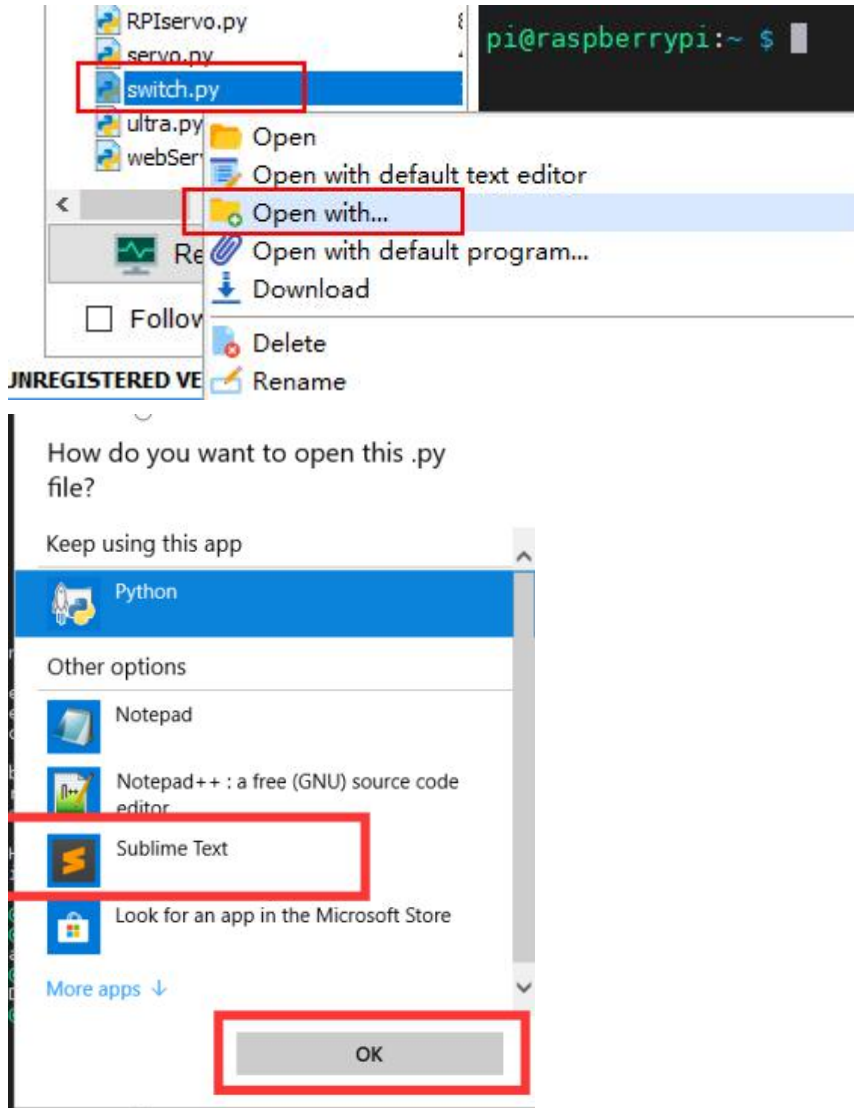
To run MobaXterm, firstly create a new session. Click **Session** in the upper left corner, and

pop-up window will be shown. Click **SSH**, fill in the IP address of the Raspberry Pi next to **Remote host**, and click **OK**. The default user name of the Raspberry Pi is *pi* and the password is *raspberrypi*. When you type in the password, it doesn't display anything on the screen, yet it doesn't mean you aren't entering anything. Press Enter after typing to log into the Raspberry Pi, and MobaXterm will remind you to save your password.



- If the user name and password are correct, you can change the user name and password, which is recommended to do for security reason, according to the prompt in the terminal.
- After the successful login, MobaXterm will automatically save the conversation. Next time to connect to the Raspberry Pi, you only need to double click on the left side of the IP address to log in. But if the IP address of the Raspberry Pi was changed, you need to start a new dialogue.
- After a successful login, the left column will be replaced with a file transfer system, which allows you to interact with the system inside the Raspberry Pi. To return to session selection, just click **Sessions**.
- You can transfer programs written on other devices to the Raspberry Pi by simple drag and drop and control the Raspberry Pi in the terminal to execute the program, or transfer files in the Raspberry Pi to other devices by dragging.
- If you want to use another IDE to edit files in Raspberry Pi, just look for the file in the file transfer system on the left side of the MobaXterm, right-click on the file, click **Open with...** and

select the IDE desired. After editing, press **CTRL+S** to save the file and it will be automatically synchronized to the Raspberry Pi.



- However, when you use MobaXterm's file transfer system to edit files in the Raspberry Pi, you need to pay attention to the permissions problem. Since the file transfer system does not have root permissions, when prompted to save after editing the file, the "Permission denied" error will fail the saving. You need to use the following command to give the edit permission to MobaXterm:

`sudo chmod 777 [FileName]`

For example:

```
sudo chmod 777 adeept_darkpaw/server/LED.py
```

- When granting permission, where ******* is the Linux permission code, we do not recommend to use the permission **777**; but to novice, 777 can avoid many account and permission problems. Surely you can also set it to 700, so that only the owner can read, write and execute. Learn more about Linux permissions through the [maketecheasier](https://www.maketecheasier.com/file-permissions-what-does-chmod-777-means) article at this link:

<https://www.maketecheasier.com/file-permissions-what-does-chmod-777-means>

- If you encounter any problems, please contact us by email: support@adeept.com