Lesson 4 How to View or Edit the Code Program in Raspberry Pi

To make daily use of the Raspberry Pi more convenient, we usually do not connect peripherals such as mouse, keyboard, and monitor to the Raspberry Pi. Since our Raspberry Pi is installed inside the robot, often with peripherals to control the Raspberry Pi, the efficiency of programming and testing will be seriously affected. Therefore, we introduce a method of programming in 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).
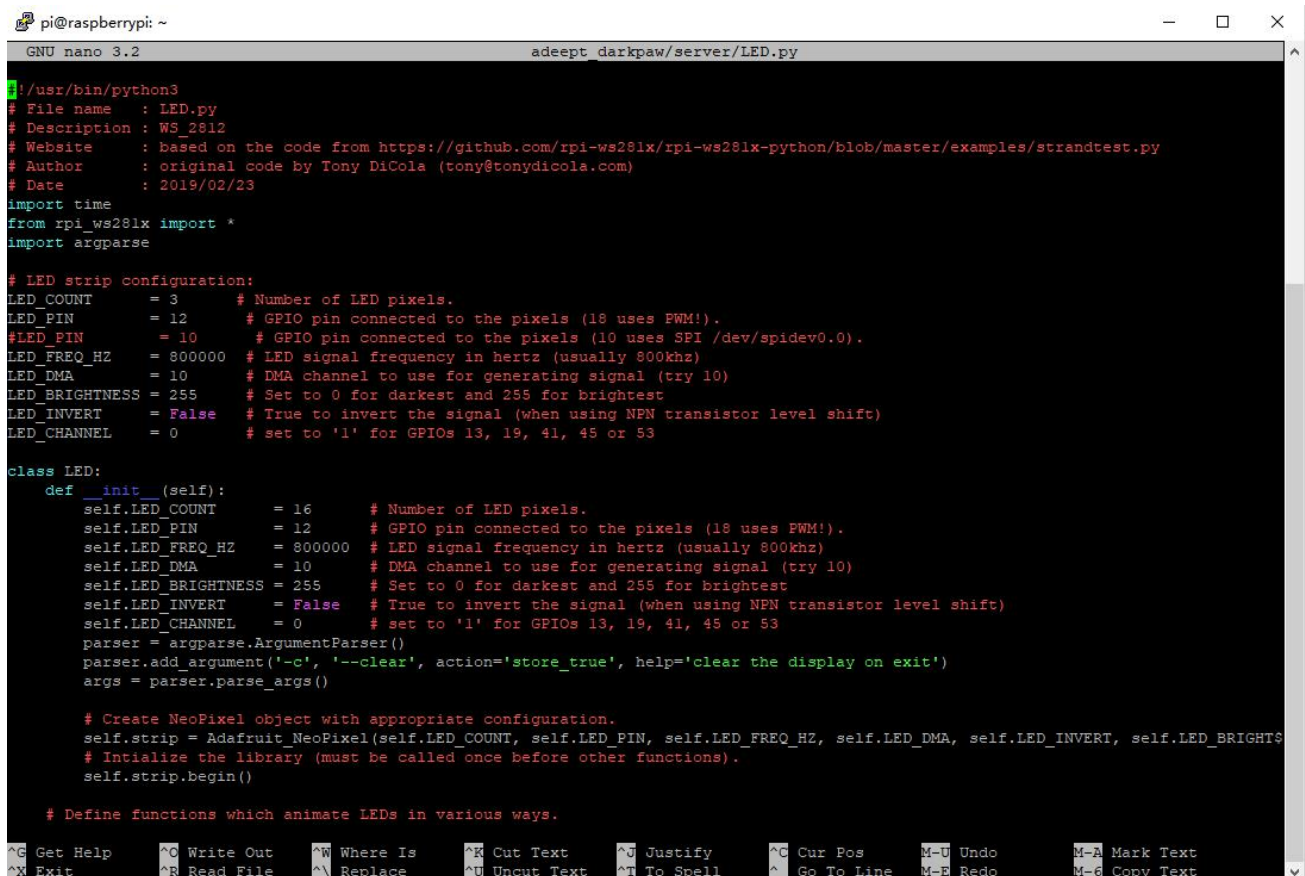
Operation Steps

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: raspberry.

2. Type in "ls" to show the files to be edited or viewed.

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

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

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

```
pi@raspberrypi: ~                                                                    —    □    ×

  GNU nano 3.2                            adeept_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_BRIGHT$
        # Intialize the library (must be called once before other functions).
        self.strip.begin()

    # Define functions which animate LEDs in various ways.

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^  Go To Line  M-E Redo       M-6 Copy Text
```

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).

Support email: support@adeept.com

To run MobaXterm, firstly, create a new session, click Session in the upper left corner, click SSH in the pop-up window, fill in the IP address of the Raspberry Pi behind Remote host, and finally click OK, the default account name of the Raspberry Pi is pi , The default password is raspberry. Just the password doesn't appear on the screen when you enter it and the * number doesn't mean nothing **Enter** successfully, press after login to log in to the Raspberry Pi, MobaXterm will remind you to save the password.You need to choose.
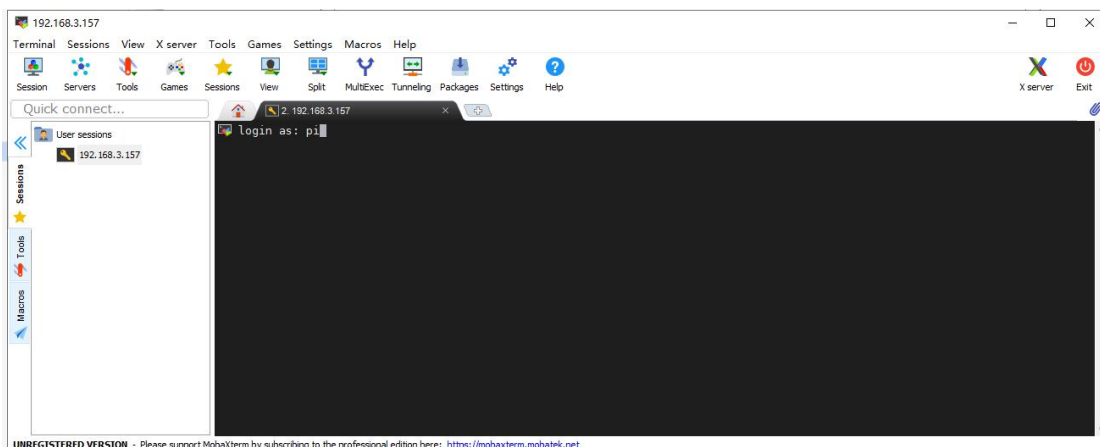
- If the user name and password are correct, you can change the user name and password according to the prompt in the terminal, which is more secure.

- After the success of the login, MobaXterm will automatically save the conversation, when connected to the raspberry pie again next time only need to double click on the left side of the IP address can be connected to the Raspberry Pi again, if there is no save username and password will need to input the user name and password, if the IP address of the Raspberry Pi changed, you need to start a new dialogue.

- After a successful login, the left column is replaced with a file transfer system, which allows you to interact with the system inside the Raspberry Pi. If you want to return to session selection, just click Sessions.

- Programs you write on other devices can be transferred to the Raspberry Pi by simple drag and drop, and then the Raspberry Pi can be controlled in the terminal to execute the program, or the files in the raspberry Pi can be dragged to other devices.

- If you want to use another IDE to edit files in Raspberry Pi, you can find the file you want to edit in the file transfer system on the left side of the MobaXterm. Right-click on this file and select your IDE so you can use your favorite on other devices IDE to edit the Raspberry Pi file, after editing, press "CTRL+S" to save the file and it will be automatically synchronized to the Raspberry Pi.
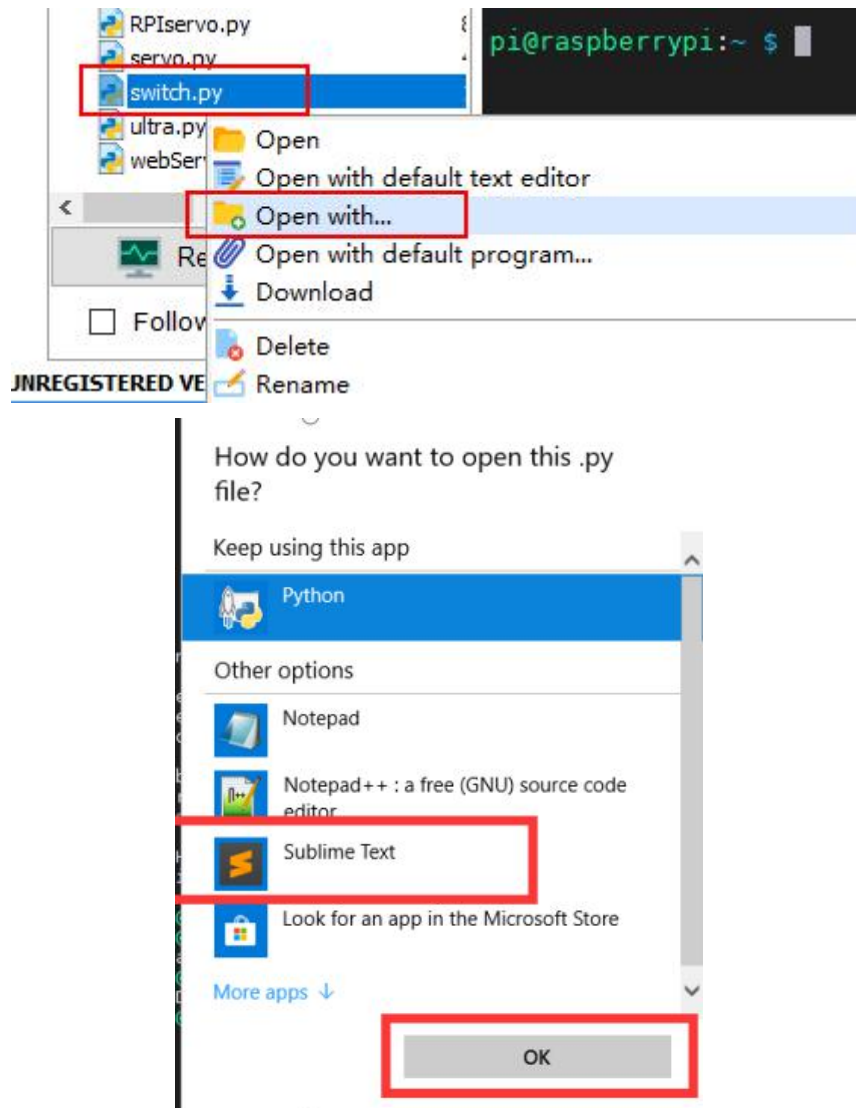
- However, it should be noted that when you use MobaXterm's file transfer system to edit files in the Raspberry Pi, you need to pay attention to the permissions problem, because the file transfer system does not have root permissions, so if you are prompted to save after editing the file The permission denied error causes the file cannot be saved after editing. You need to use the following command to give the file you want to edit permission to be edited by MobaXterm:

sudo chmod 777 [FileName]　, For example:

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

- Give permissions, where *** is the Linux permission code, we do not recommend the use of permissions such as 777, but for novices 777 can avoid many account and permissions problems, of course, you can also set it to 700 , So that only the owner can read, write and execute. You can learn more about Linux permissions through maketecheasier article from the article link

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

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