

# 树莓派 python 编程（notepad++、上传、运行）

## 目录

一、 实验概述.....	2
二、 实验器材.....	2
三、 设备连接.....	3
四、 树莓派 IO 口介绍.....	6
五、 代码编写（notepad++） .....	8
六、 代码上传及运行（WINSXP） .....	11
官 网： <a href="http://www.xiao-r.com">www.xiao-r.com</a> .....	13
论 坛： <a href="http://www.wifi-robots.com">www.wifi-robots.com</a> .....	13
官方商城： <a href="http://wifi-robots.taobao.com">wifi-robots.taobao.com</a> .....	13
微信公众号： .....	13

## 一、实验概述

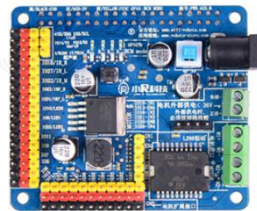
小R科技 Raspberry Pi WiFi 智能小车,提供的镜像是基于树莓派官方的 raspbian 系统,并已经集成了 python 库。可以直接使用 python2.7 来编程,然后控制树莓派的引脚。

小R科技研发设计的 PWR.A53 系列电源电机驱动板,将树莓派上 40PIN 插针的 IO 口全部引出。并根据功能划分了引脚。电源板上自带了 3 颗 LED 灯由 IO 口的低电平驱动,我们以此为例来讲解 python 编程的基本操作。

## 二、实验器材



1、树莓派主板



2、PWR.A53 电源板



3、HDMI 显示屏、HDMI 线、USB 数据线（HDMI 接口的显示器、电视都可以）



4、电池套装



全键盘 免驱 即插即用 大触控板

5、树莓派无线键鼠

### 三、设备连接

- 1、树莓派主板和 PWR.A53 电源板扣在一起；
- 2、电池接 PWR.A53 主板的电源口；
- 3、拿出无线键鼠后盖里面的接收头，接在树莓派的一个 USB 口上，并打开键鼠右上角的开关到 ON；
- 4、数据线连接树莓派 USB 口和屏幕，并打开屏幕背后开关到 ON；
- 5、HDMI 线连接树莓派和屏幕的 HDMI 接口。
- 6、打开电源板开关，等待系统启动完成。

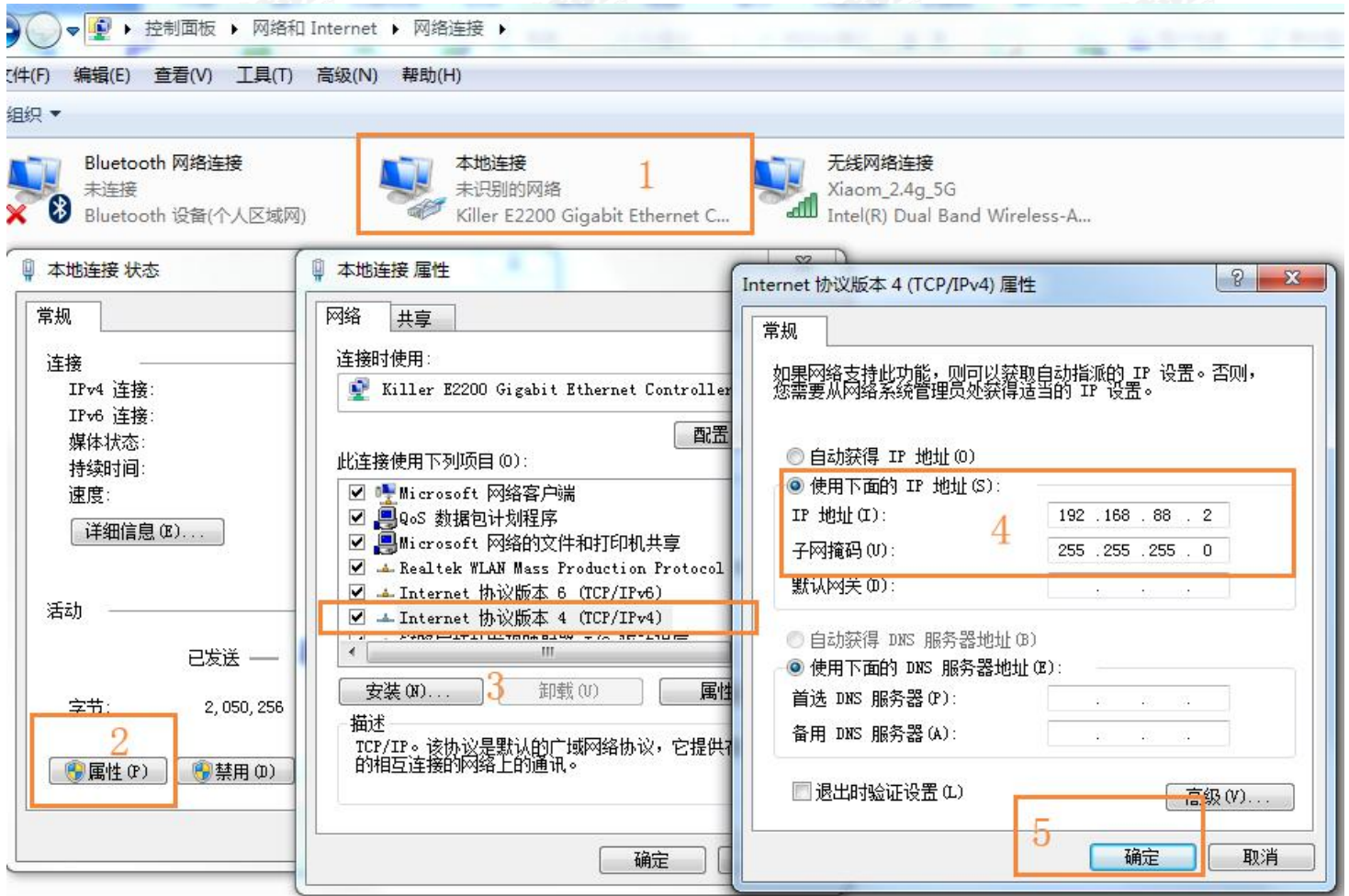
连接示意图如下：



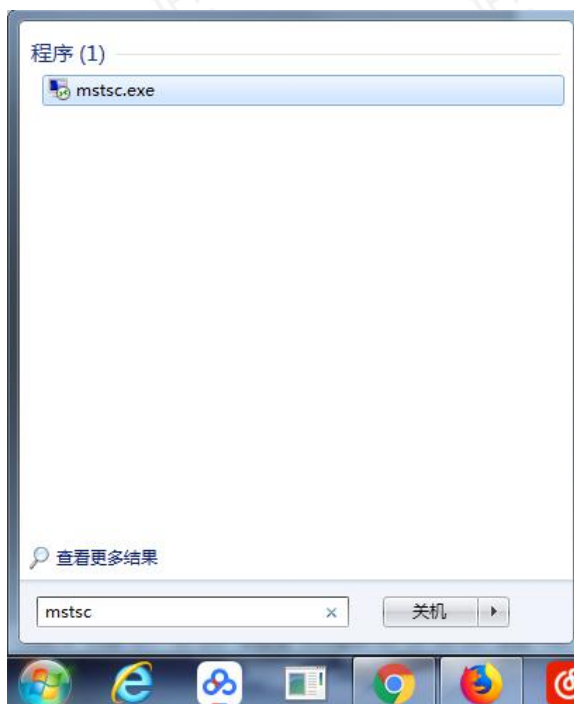
如使用远程桌面连接，则按照如下步骤进行后续操作：

- A、连接网线到树莓派网口，在网络连接管理界面双击对应的网卡 → 打开属性 → 双击 IPV4 设置 → 设置 IP:192.168.88.2 子网掩码:255.255.255.0 → 确定

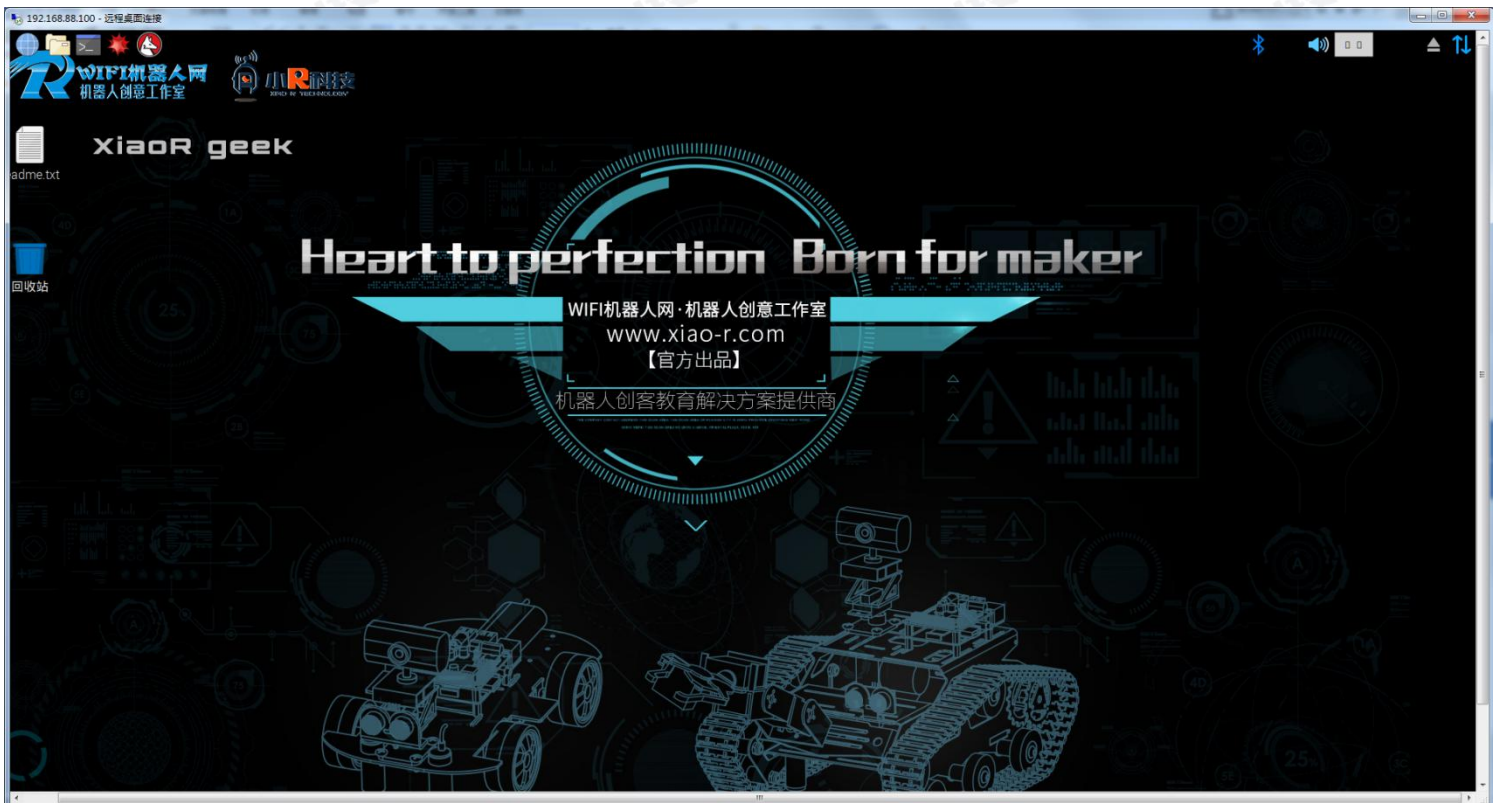
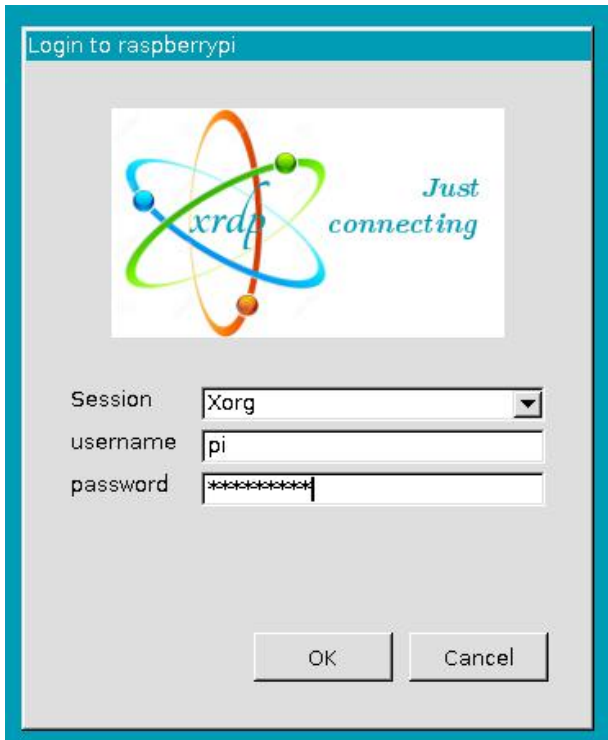




B、WIN+R 输入 mstsc，运行电脑远程桌面连接。输入树莓派有线网口地址 192.168.88.100 并连接。



C、使用默认连接方式，输入用户名：pi 密码：raspberrypi（全部小写），点OK 连接到树莓派远程桌面。如果弹出下图右侧日志框图，可能是密码输入错误，重新输入即可。





#### 四、树莓派 IO 口介绍

树莓派 python 库对引脚有两种定义方式。一种是以 40PIN 插针的顺序定义 IO 口（.BOARD 模式），另一种是以引脚在主芯片内部实际的 IO 序号来定义 IO 口（.BCM 模式）。我们使用第二种（.BCM 模式），这样更接近芯片本身和硬件原理的设计，不会把 3V3 的引脚当做 IO1 来使用。下图是树莓派主板 IO 口以及 PWR.A53 主板上 IO 口分布图。

#### ► 引脚定义



Raspberry Pi 3B



Raspberry Pi 3B+

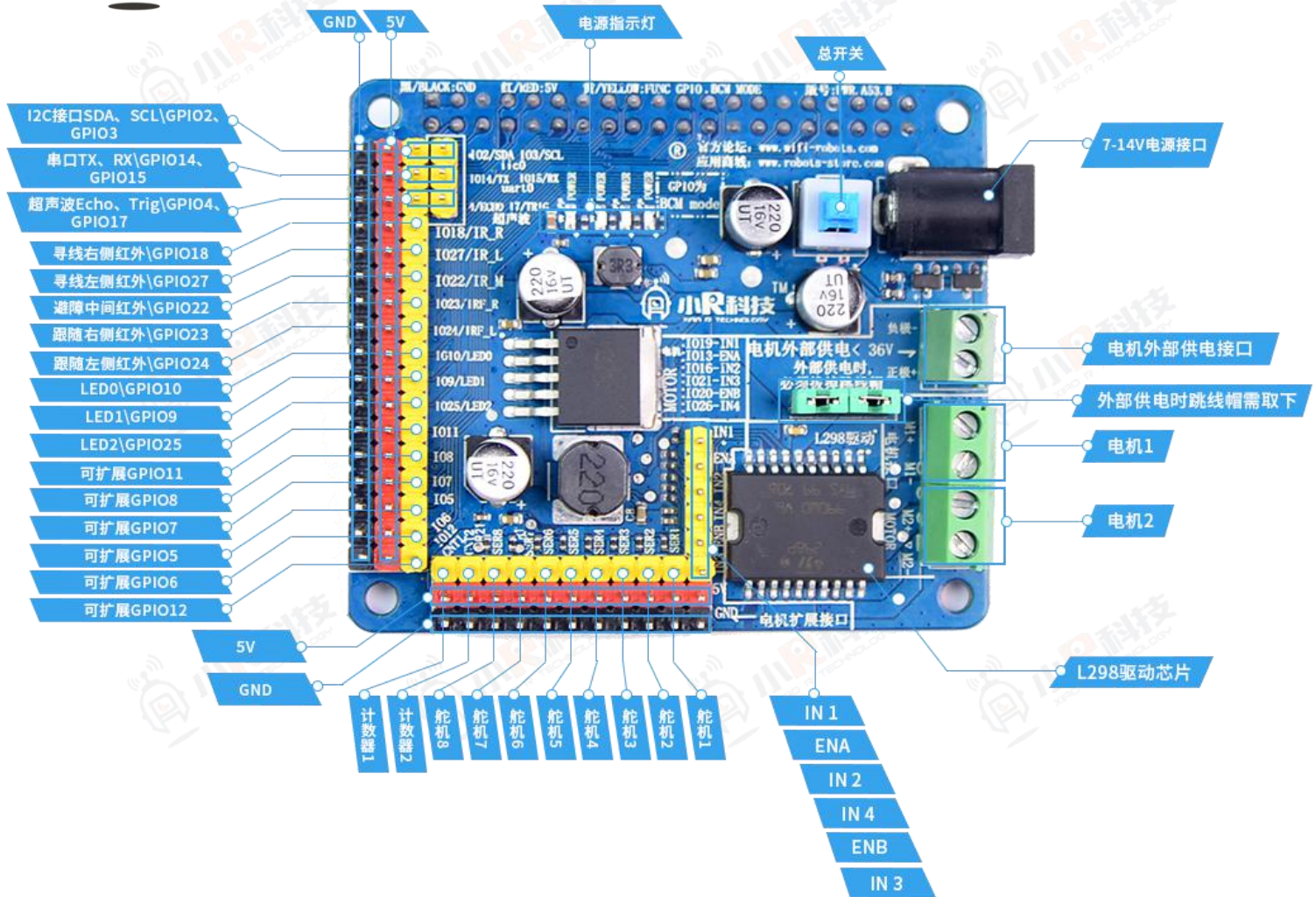
---4Pin PoE 供电引脚

#### Raspberry Pi 3B / 3B+ GPIO 引脚

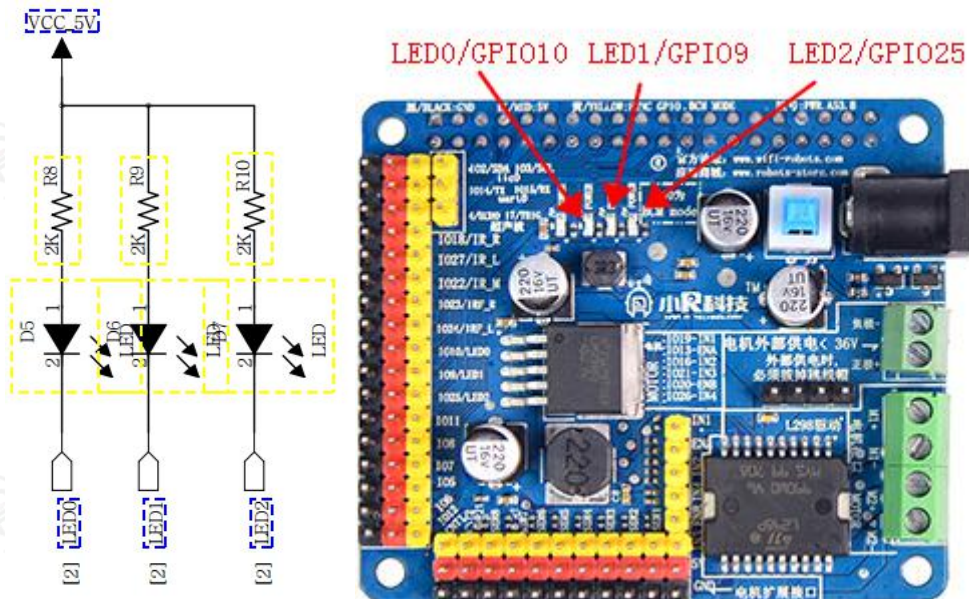
3V3	1	2	5V
GPIO 2	3	4	5V
GPIO 3	5	6	GND
GPIO 4	7	8	GPIO 14
GND	9	10	GPIO 15
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
3V3	17	18	GPIO 24
GPIO 10	19	20	GND
GPIO 9	21	22	GPIO 25
GPIO 11	23	24	GPIO 8
GND	25	26	GPIO 7
ID_SC	27	28	ID_SC
GPIO 5	29	30	GND
GPIO 6	31	32	GPIO 12
GPIO 13	33	34	GND
GPIO 19	35	36	GPIO 16
GPIO 26	37	38	GPIO 20
GND	39	40	GPIO 21

#### Raspberry Pi 3B+ PoE 引脚

TR01	01	02	TR00
TR03	03	04	TR02



电源板上有三个 LED 灯，对应 IO 口、原理图和实际位置如下图所示。由原理图可知，LED 为上拉模式，IO 口输出 0 时，LED 亮，IO 输出 1 时，LED 灭。我们用 python 来实现控制三个 LED 灯闪烁。





## 五、代码编写 (notepad++)

小 R 科技树莓派系统，控制程序是基于 python2.7 编写的。python2.7 是一种脚本语言，编程容易。但是唯一问题是需要特别注意格式，保证对齐。

Python 是纯粹的自由软件，源代码和解释器 CPython 遵循 GPL(GNU General Public License)协议。Python 语法简洁清晰，特色之一是强制用空白符 (white space)作为语句缩进。而空格键和 TAB 键的缩进位置不同，使用普通文本编辑器容易混淆，导致程序出错，因此建议使用 notepad++来编辑。

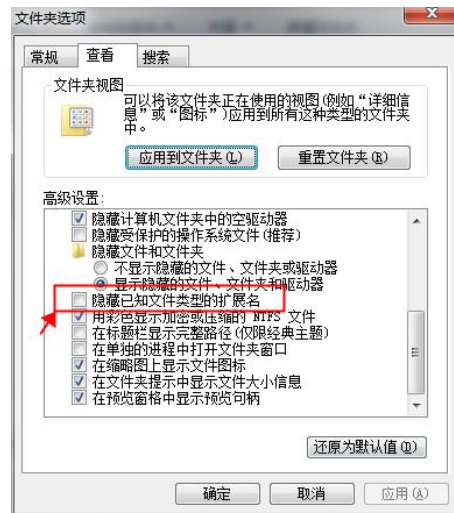
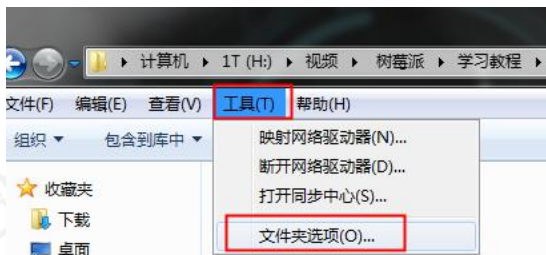
- 1、电脑安装 Notepad++ 可以百度最新版或者直接下载安装包并安装。

Notepad++最新官方版下载\_百度软件中心



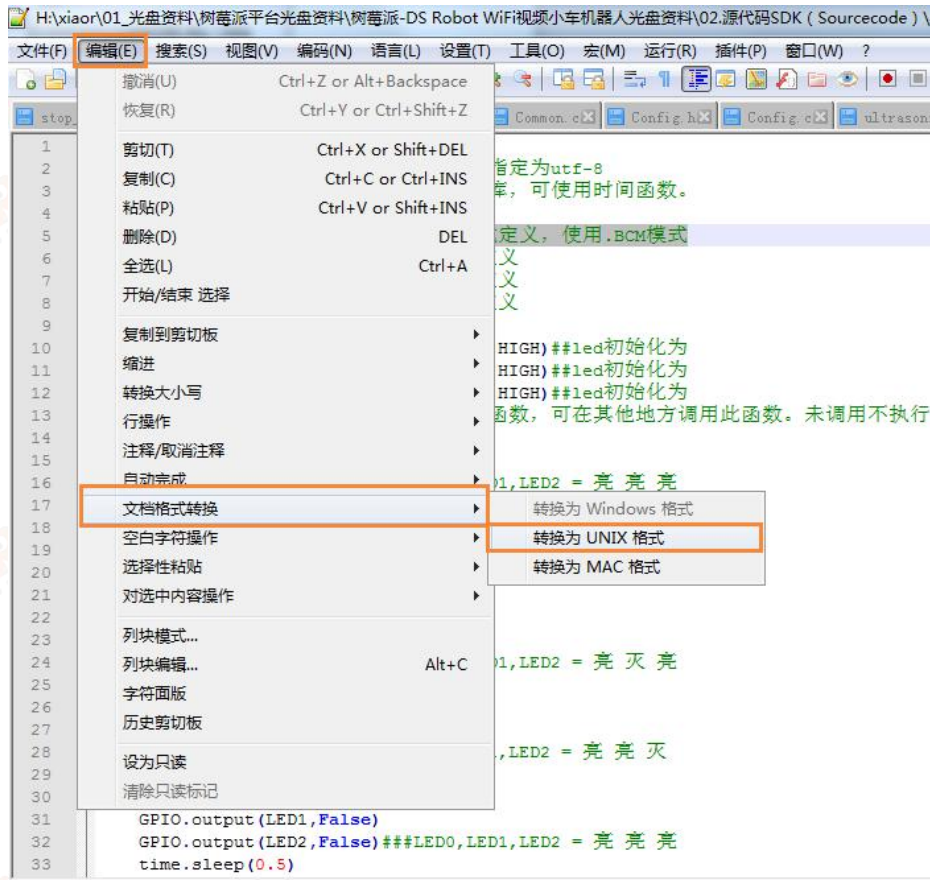
- 2、电脑新建一个文件夹，并开启查看已知文件扩展名。

步骤：工具（可用 ALT 调出）→ 文件夹选项→ 查看 → 取消勾选 隐藏已知文件类型的扩展名 → 确认。





3、新建一个文本文件命名为 led.py，使用 notepad++编辑，设置格式为 UNIX 格式。 步骤：编辑 → 文档格式转换 → 转换为 UNIX 格式



4、输入如下代码并保存：

```
#coding:utf-8
#Python 中声明文件编码的注释，编码格式指定为 utf-8
import time                #导入 time 库，可使用时间函数。
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)    ##信号引脚模式定义，使用.BCM 模式
LED0 = 10                  ##LED0 的 IO 口定义
LED1 = 9                   ##LED1 的 IO 口定义
LED2 = 25                  ##LED2 的 IO 口定义
GPIO.setwarnings(False)
GPIO.setup(LED0,GPIO.OUT,initial=GPIO.HIGH)##led 初始化为
GPIO.setup(LED1,GPIO.OUT,initial=GPIO.HIGH)##led 初始化为
GPIO.setup(LED2,GPIO.OUT,initial=GPIO.HIGH)##led 初始化为
def init_light():          ##使用 def 定义函数，可在其他地方调用此函数。未调用不执行。
    GPIO.output(LED0,False)
    GPIO.output(LED1,False)
    GPIO.output(LED2,False)####LED0,LED1,LED2 = 亮 亮 亮
```

```
time.sleep(0.5)
GPIO.output(LED0,True)
GPIO.output(LED1,False)
GPIO.output(LED2,False)###LED0,LED1,LED2 = 灭 亮 亮
time.sleep(0.5)
GPIO.output(LED0,False)
GPIO.output(LED1,True)
GPIO.output(LED2,False)###LED0,LED1,LED2 = 亮 灭 亮
time.sleep(0.5)
GPIO.output(LED0,False)
GPIO.output(LED1,False)
GPIO.output(LED2,True)###LED0,LED1,LED2 = 亮 亮 灭
time.sleep(0.5)
GPIO.output(LED0,False)
GPIO.output(LED1,False)
GPIO.output(LED2,False)###LED0,LED1,LED2 = 亮 亮 亮
time.sleep(0.5)
GPIO.output(LED0,True)
GPIO.output(LED1,True)
GPIO.output(LED2,True)###LED0,LED1,LED2 = 灭 灭 灭
time.sleep(0.5)
```

for i in range(1,5):      #调用 rang ( ) 循环函数，功能类似 for ( i =1;i<5;i++ ) 执行 4 遍  
    init\_light()

'''

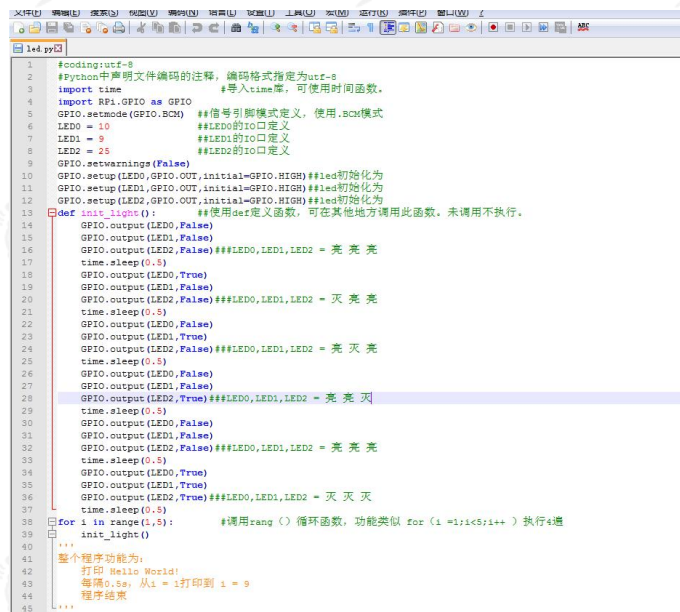
整个程序功能为：

每隔 0.5s，执行一次 LED 状态

并循环四次

程序结束

'''



```
1 #coding:utf-8
2 #Python中声明文件编码的注释，编码格式指定为utf-8
3 import time      #导入time库，可使用时间函数。
4 import RPi.GPIO as GPIO
5 GPIO.setmode(GPIO.BCM)      ##信号引脚模式定义，使用BCM模式
6 LED0 = 10      ##LED0的IO口定义
7 LED1 = 9      ##LED1的IO口定义
8 LED2 = 25      ##LED2的IO口定义
9 GPIO.setwarnings(False)
10 GPIO.setup(LED0,GPIO.OUT,initial=GPIO.HIGH)##led初始化为
11 GPIO.setup(LED1,GPIO.OUT,initial=GPIO.HIGH)##led初始化为
12 GPIO.setup(LED2,GPIO.OUT,initial=GPIO.HIGH)##led初始化为
13 def init_light():      ##使用def定义函数，可在其他地方调用此函数。未调用不执行。
14     GPIO.output(LED0,False)
15     GPIO.output(LED1,False)
16     GPIO.output(LED2,False)###LED0,LED1,LED2 = 亮 亮 亮
17     time.sleep(0.5)
18     GPIO.output(LED0,True)
19     GPIO.output(LED1,False)
20     GPIO.output(LED2,False)###LED0,LED1,LED2 = 灭 亮 亮
21     time.sleep(0.5)
22     GPIO.output(LED0,False)
23     GPIO.output(LED1,True)
24     GPIO.output(LED2,False)###LED0,LED1,LED2 = 亮 灭 亮
25     time.sleep(0.5)
26     GPIO.output(LED0,False)
27     GPIO.output(LED1,False)
28     GPIO.output(LED2,True)###LED0,LED1,LED2 = 亮 亮 灭
29     time.sleep(0.5)
30     GPIO.output(LED0,False)
31     GPIO.output(LED1,False)
32     GPIO.output(LED2,False)###LED0,LED1,LED2 = 亮 亮 亮
33     time.sleep(0.5)
34     GPIO.output(LED0,True)
35     GPIO.output(LED1,True)
36     GPIO.output(LED2,True)###LED0,LED1,LED2 = 灭 灭 灭
37     time.sleep(0.5)
38     #调用 rang ( ) 循环函数，功能类似 for ( i =1;i<5;i++ ) 执行4遍
39     for i in range(1,5):
40         init_light()
41     '''
42     整个程序功能为：
43     打印 Hello World!
44     每隔0.5s，从 i = 1打印到 i = 9
45     程序结束
46     '''
```

## 六、代码上传及运行（WINSFTP）

1、打开安装后的 WinSCP 程序

2、登录 WinSCP，



文件协议：选择 SCP

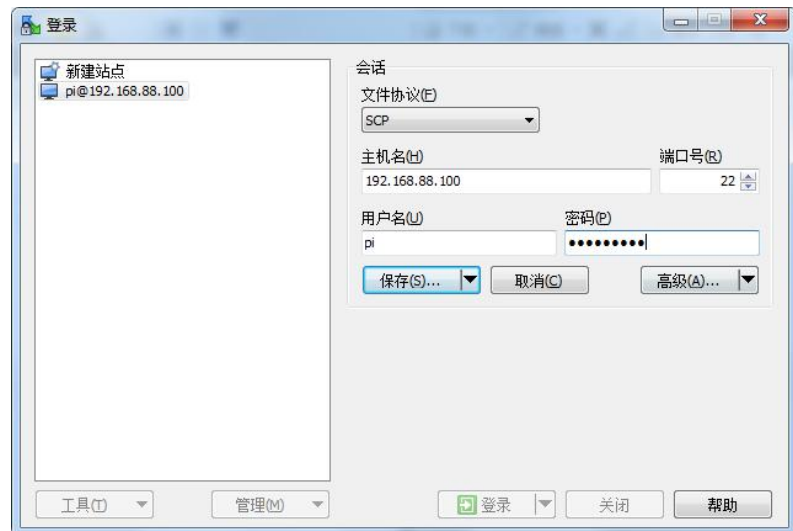
主机名：wifi 连接为 192.168.1.1

网口连接为 192.168.88.100

端口号：22

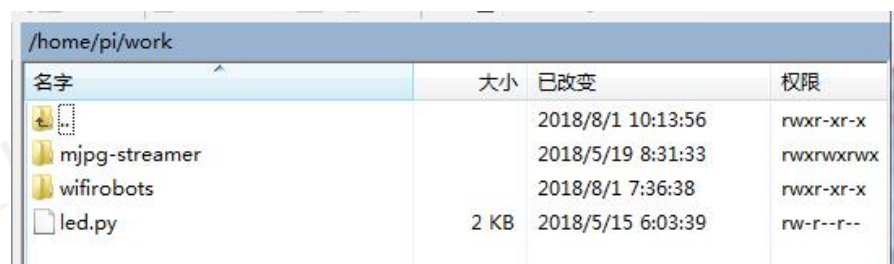
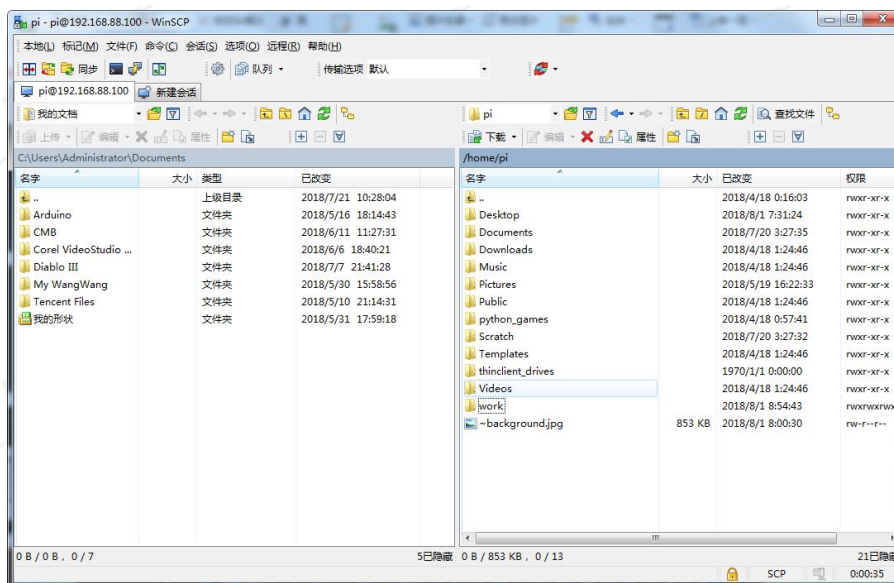
用户名：pi

密码为：raspberrypi



3、可以保存连接，下次直接使用。

4、点击登录，并把编辑好的程序拖入 work 文件夹（可先双击进入 work 文件夹，后拖入文件）





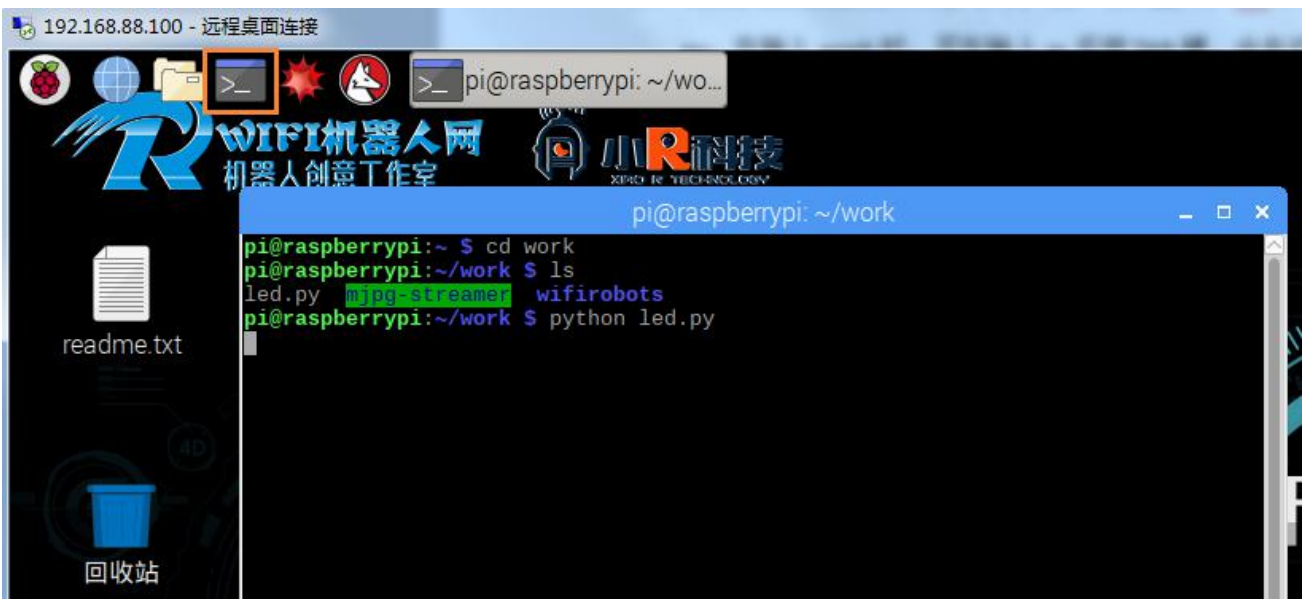
5、在远程桌面打开命令终端，并输入 `cd work` 回车进入到 `work` 文件夹。

ps: 在输入 `work` 时，可在输入 `w` 后按 `TAB` 键，会自动补全路径。

再通过指令 `ls` 回车，显示 `work` 内部的文件目录，已经有了 `led.py` 文件啦。

然后输入 `python led.py` 回车运行程序，可以观察到 LED 灯按照代码逻辑闪烁了。

ps: 在输入 `led.py` 时，可在输入 `l` 后按 `TAB` 键，会自动补全。





官 网: [www.xiao-r.com](http://www.xiao-r.com)

论 坛: [www.wifi-robots.com](http://www.wifi-robots.com)

官方商城: [wifi-robots.taobao.com](http://wifi-robots.taobao.com)

微信公众号:

