# 树莓派+MegaPi实现web显示传感器数值

目录

[树莓派+MegaPi实现web显示传感器数值 1](#_Toc34901310)

[案例说明 1](#_Toc34901311)

[软硬件准备工作 1](#_Toc34901312)

[MegaPi初始化： 1](#_Toc34901313)

[MegaPi与树莓派连线 1](#_Toc34901314)

[树莓派电源连接 1](#_Toc34901315)

[树莓派ip查询 1](#_Toc34901316)

[树莓派开启SSH服务 1](#_Toc34901317)

[安装MegaPi库 1](#_Toc34901318)

[安装树莓派vim 1](#_Toc34901319)

[树莓派flask及简单应用 1](#_Toc34901320)

[案例：声音传感器数值在web端显示案例 1](#_Toc34901321)

[文件目录结构 1](#_Toc34901322)

[代码展示： 1](#_Toc34901323)

[代码运行及网页访问 1](#_Toc34901324)

## 案例说明

本案例是使用树莓派+MegaPi+python+flask+传感器，实现手机或电脑访问对应网页端显示传感器数值，展示了树莓派python调用makeblock体系中RJ25传感器数值的方法，为树莓派高阶使用（如数据分析，生成网页图表）提供了可能。

## 软硬件准备工作

本案例需要使用以下软硬件

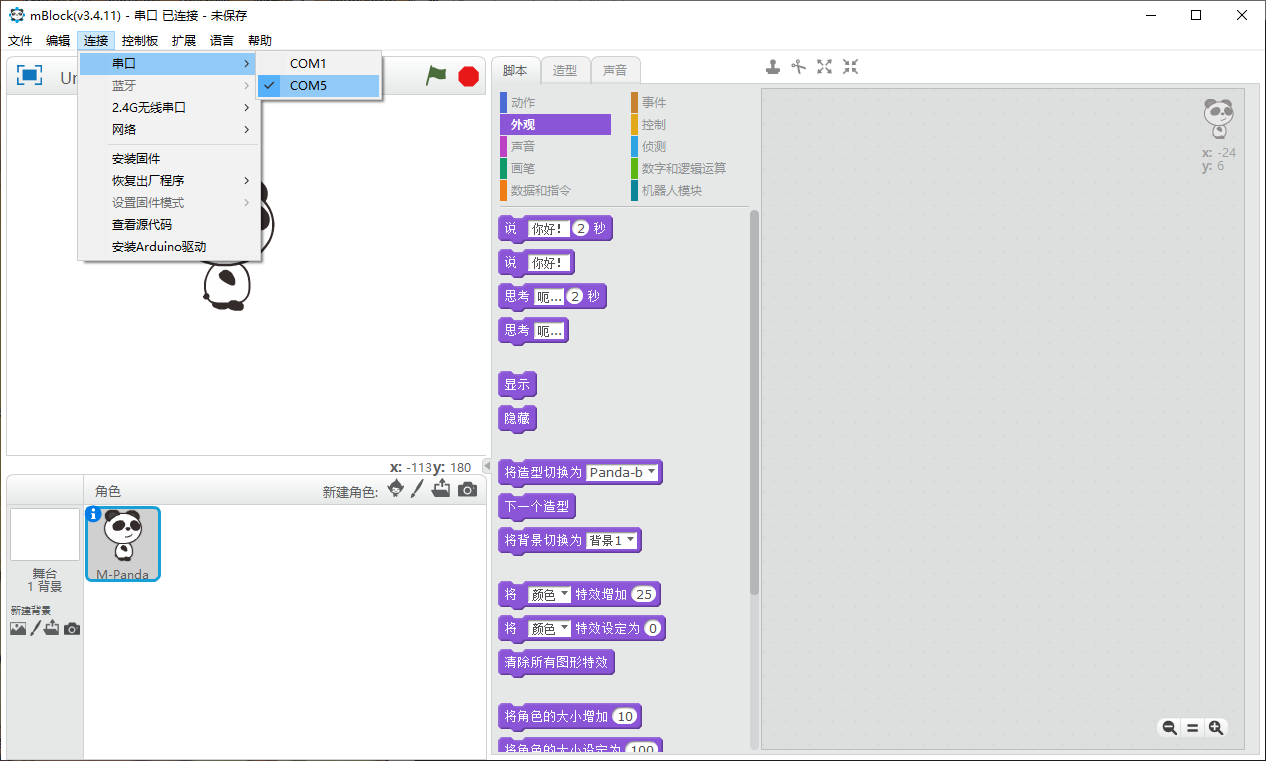
硬件：树莓派3B+、MegaPi、方口USB线、RJ25声音传感器、RJ25线、树莓派电源

软件：树莓派python2.7、MegaPi默认固件

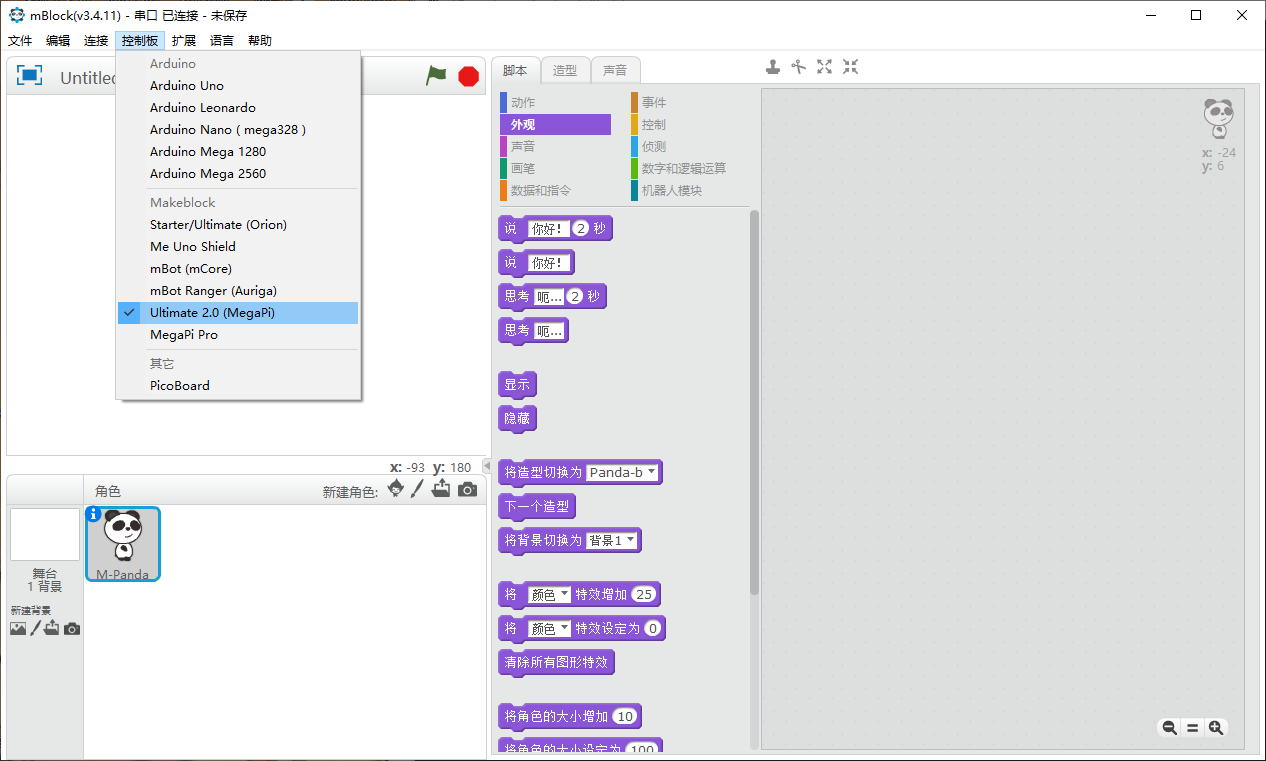
### MegaPi初始化：

用方口USB线连接电脑和MegaPi

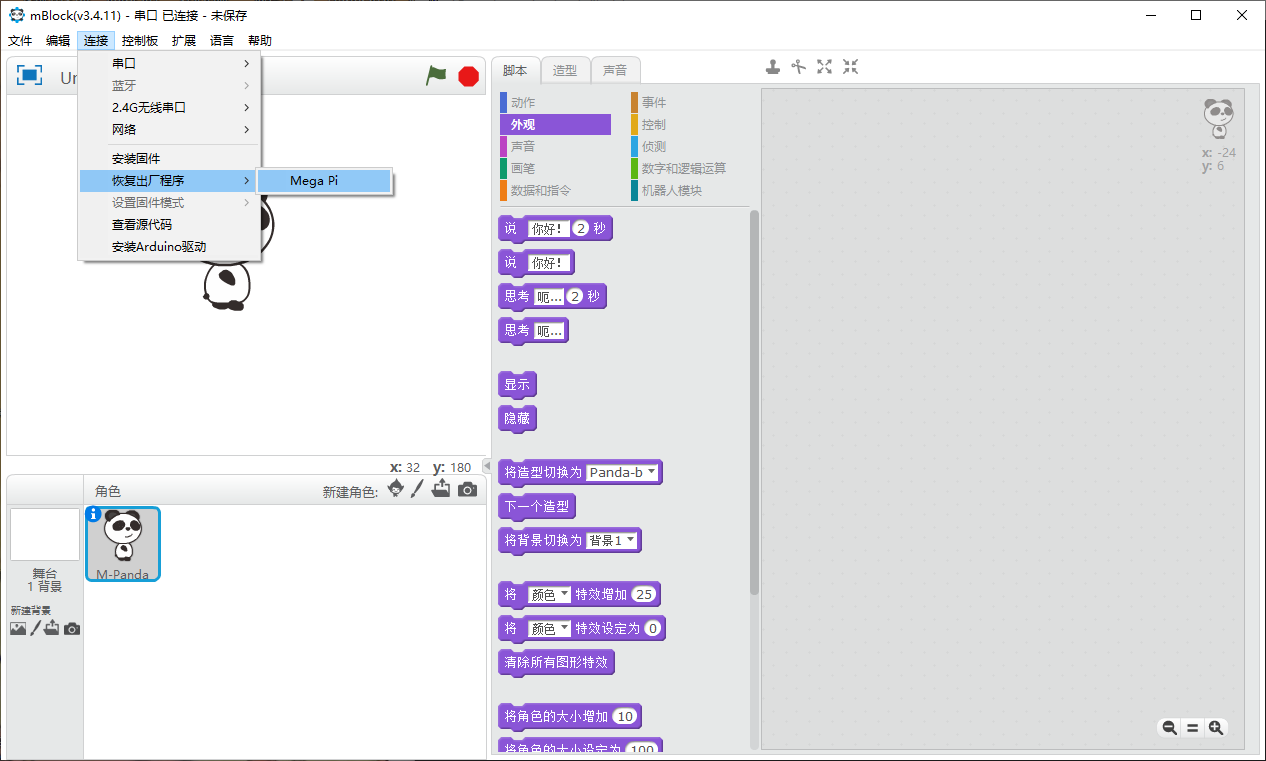
电脑端打开mBlock3，连接串口



点击控制板-选择MegaPi



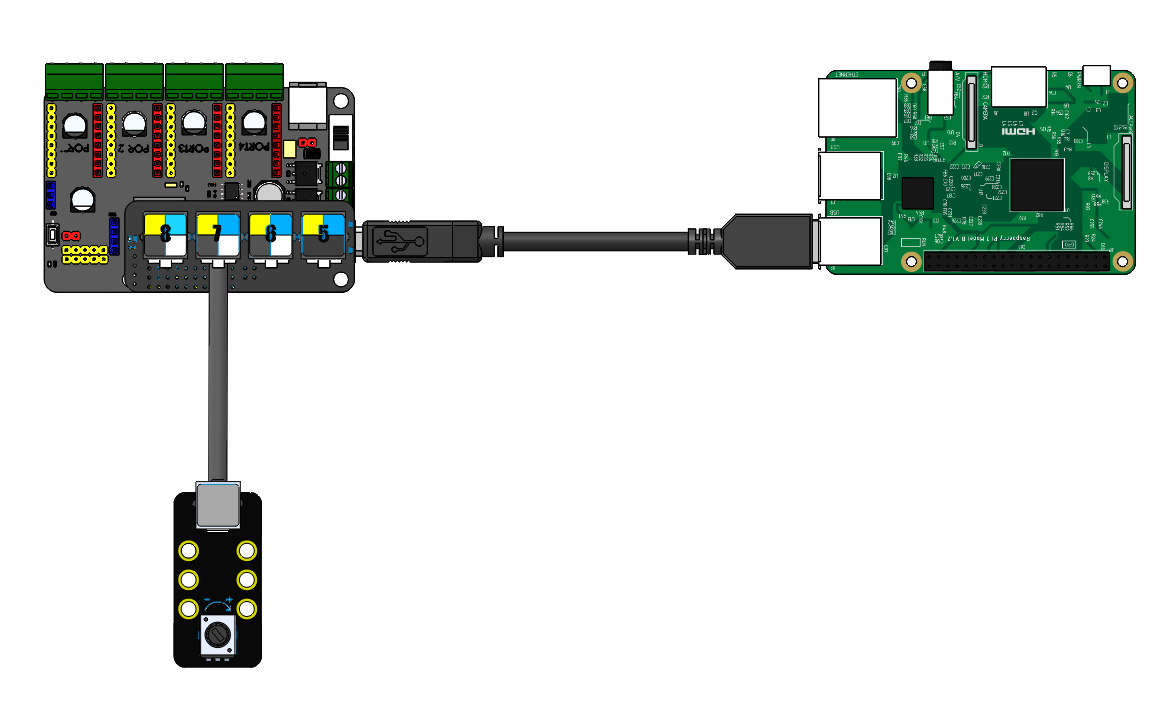
点击连接-恢复出厂程序-MegaPi，等待程序上传完成。



### MegaPi与树莓派连线

MegaPi与树莓派连接有三种方式

#### 方口USB连接（此案例使用此方法）

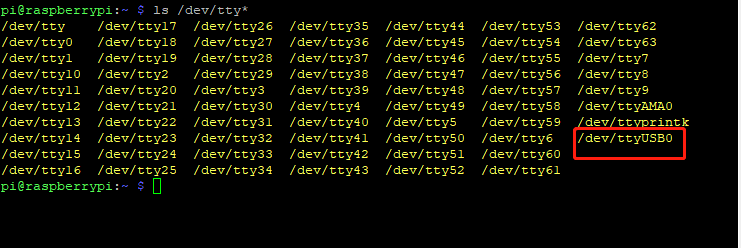


使用方口USB连接树莓派和MegaPi时，MegaPi驱动初始化命令需要使用下列命令

bot.start('/dev/ttyUSB0')

ttyUSB0串口是否启用可以使用下列命令查询：

ls /devtty\*

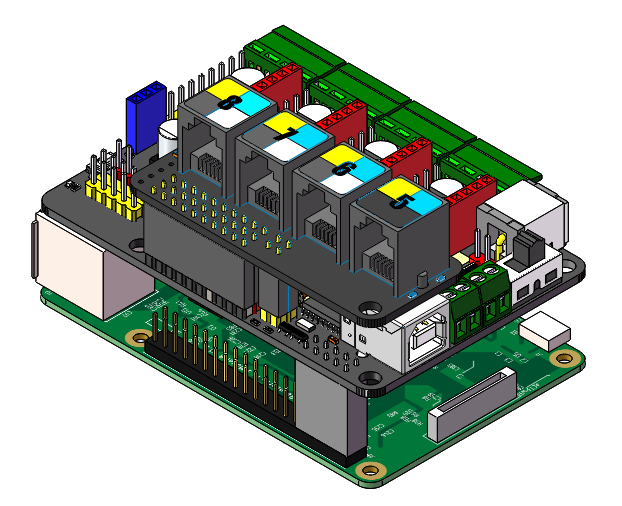


如图说明ttyUSB0串口正常工作

#### MegaPi插针直插

MegaPi的USB通信端口旁边预留了10针的树莓派连接端口，可以通过MegaPi套装附件中的10针排线连接树莓派，如图。

使用此方法连接可以大大减少线材使用，简洁，MegaPi直插电源即可给树莓派供电，整机只需要一个电源即可驱动大功率电机等设备。



树莓派和MegaPi连接引脚定义

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名  树莓派 | 物理引脚  BOARD编码 | | 功能名  MegaPi |
| 3.3V | 1 | 2 | 5V |
| SDA.1 | 3 | 4 | 5V |
| SCL.1 | 5 | 6 | GND |
| GPIO.7 | 7 | 8 | TXD |
| GND | 9 | 10 | RXD |

注意：使用此方法连接MegaPi需要关闭树莓派蓝牙，并且需要恢复硬件串口

完成后初始化驱动需要使用下列语句

bot.start('/dev/ttyAMA0')

### 树莓派电源连接

1. 充电宝用microUSB线连接到树莓派microUSB口供电
2. 用树莓派电源供电

### 树莓派ip查询

1. 有显示器

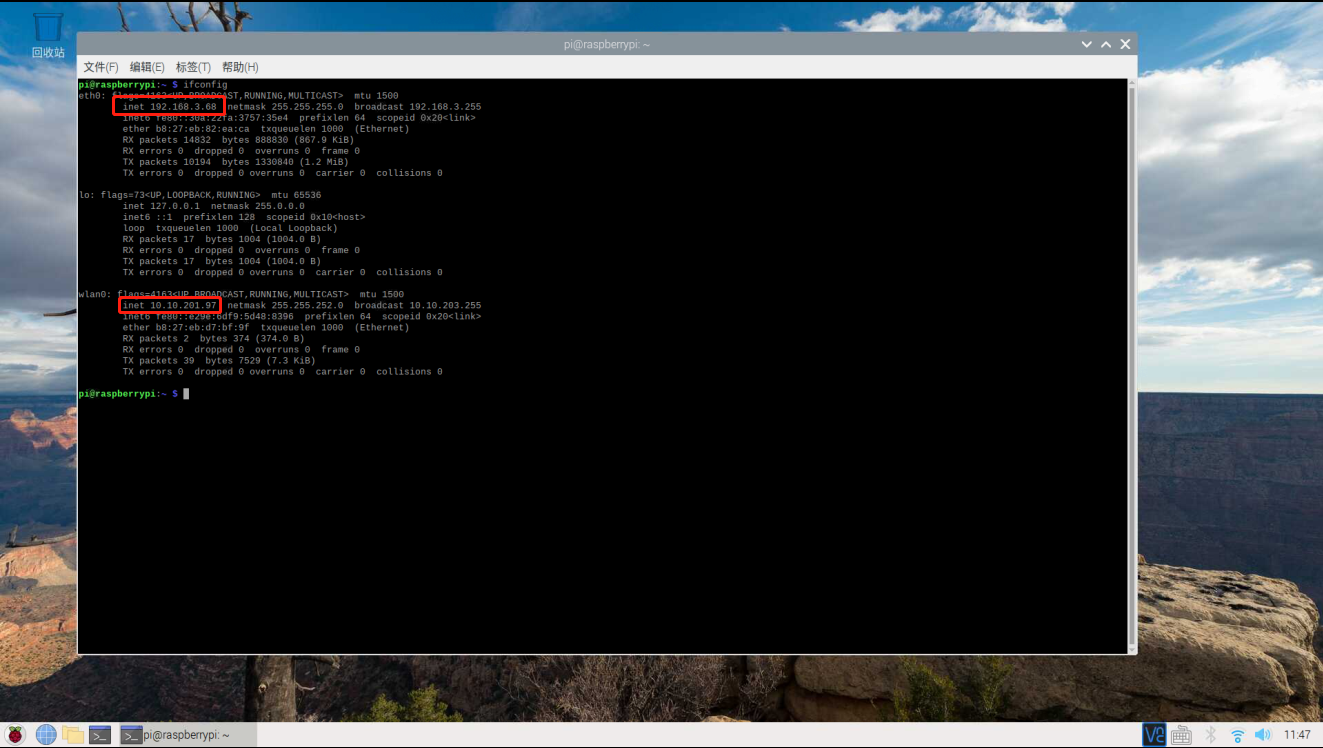
树莓派连接显示器，鼠标键盘

在树莓派终端的命令行里面输入

ifconfig

就可以看到树莓派的ip地址。

如图，eth0代表树莓派的有线ip地址，wlan0代表树莓派的无线ip地址（如图本机的有线ip为 192.168.3.68, 无线ip为 10.10.201.97）



1. 没有显示器

通过路由器查询，将路由器网线和树莓派连接，在电脑端输入路由器地址（如： 192.168.1.1）

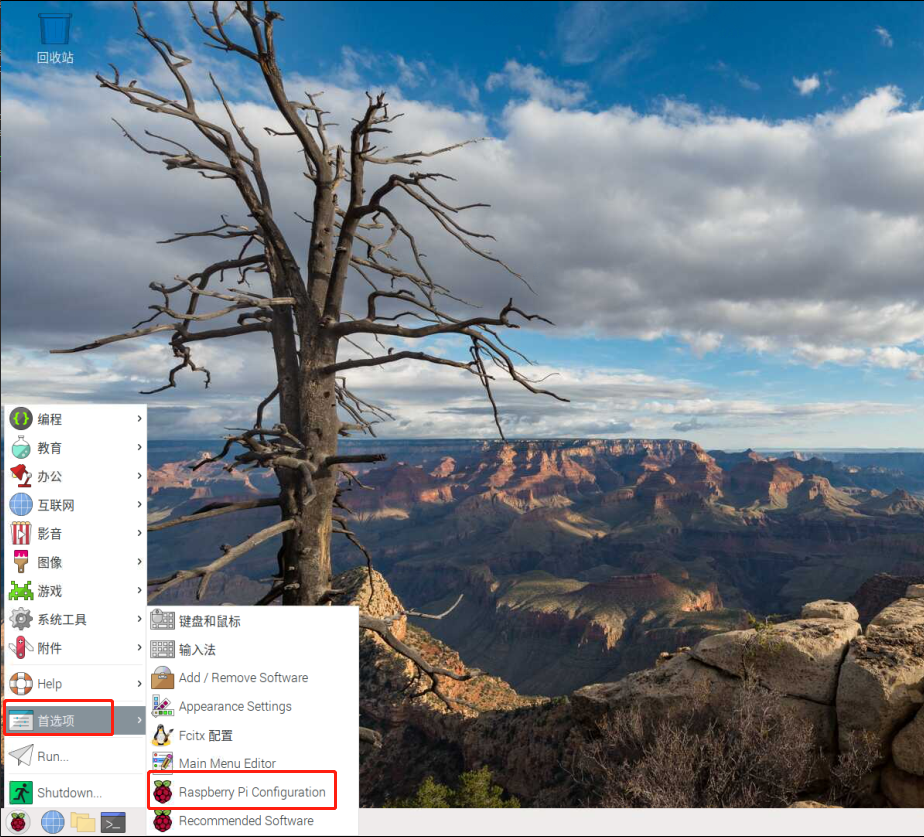
一般能在你的路由器的标签上面寻找到进入之后就能在上面找到相关的树莓派的ip地址。一般树莓派的mac地址是b开头的。

### 树莓派开启SSH服务

SSH简介：

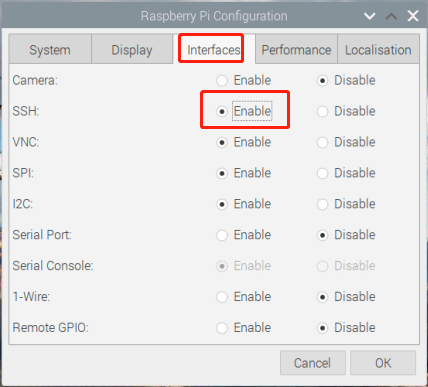
#### 有显示用图形界面开启ssh服务

点击树莓派界面左上角的树莓图标，点Preferences（中文名为首选项）的这个图标，再点击下面倒数第二个树莓图标打开树莓派的设置。



打开菜单的树莓派设置

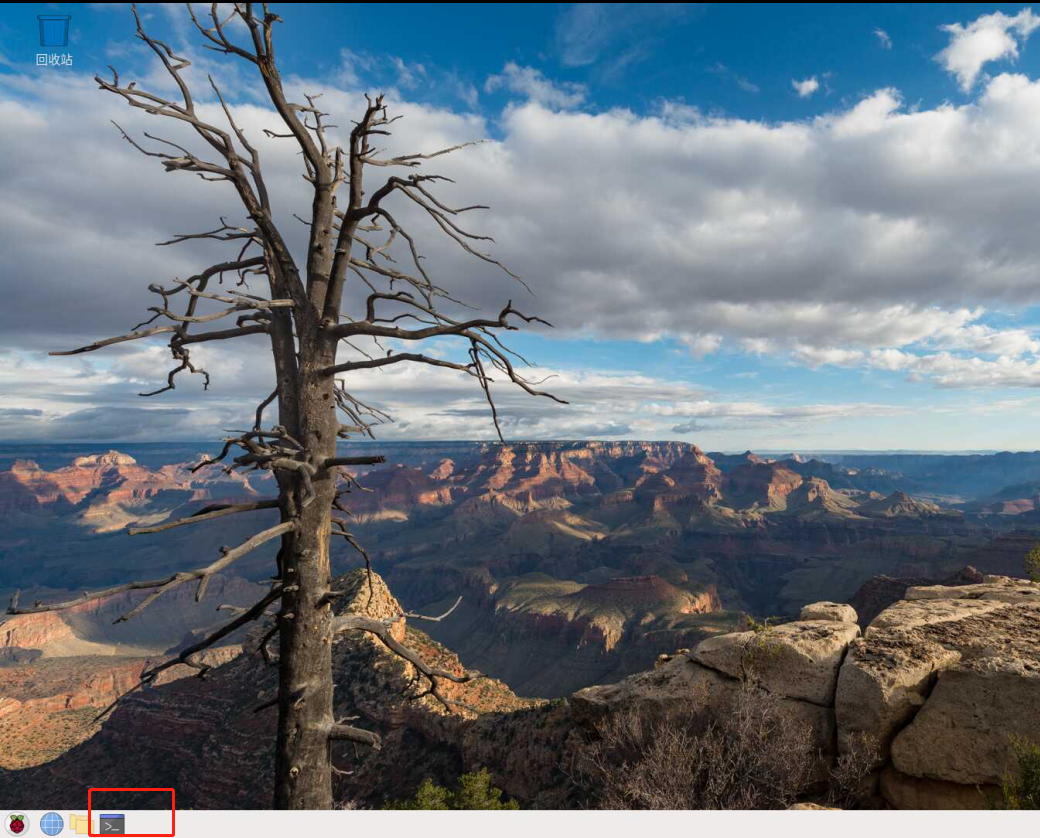
选择第二个标签Interfaces，将SSH那一行Enabled前面的那个圆点上，按右下角的OK保存，重启树莓派后就会自动开启SSH服务。



图形界面开启ssh服务

#### 有显示用命令行开启ssh服务

点击及界面上菜单栏黑色框打开可以输入命令行的终端。

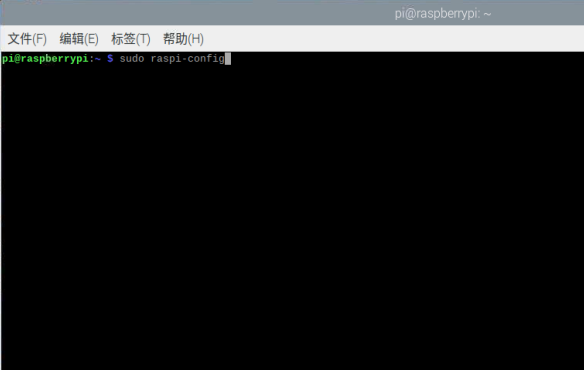


打开终端

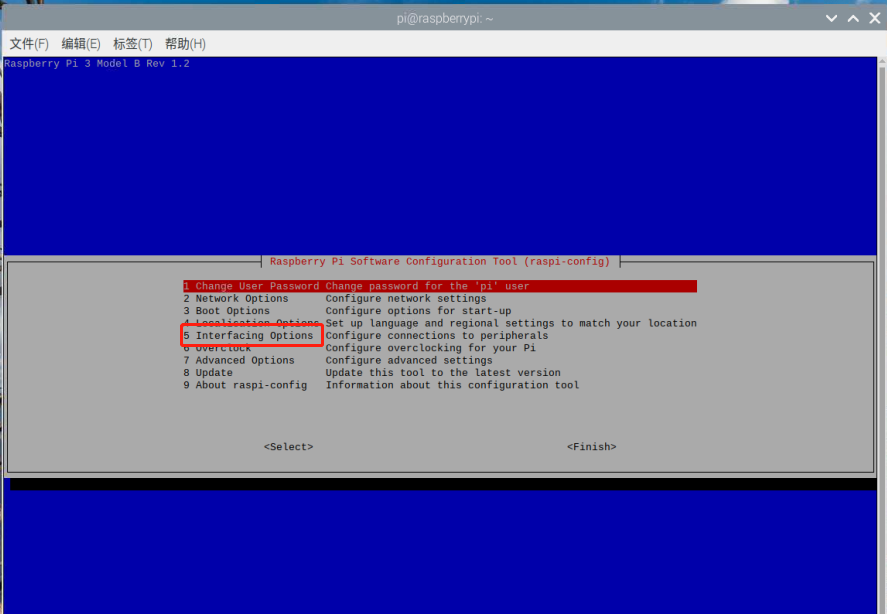
在打开的界面命令行输入：

sudo raspi-config

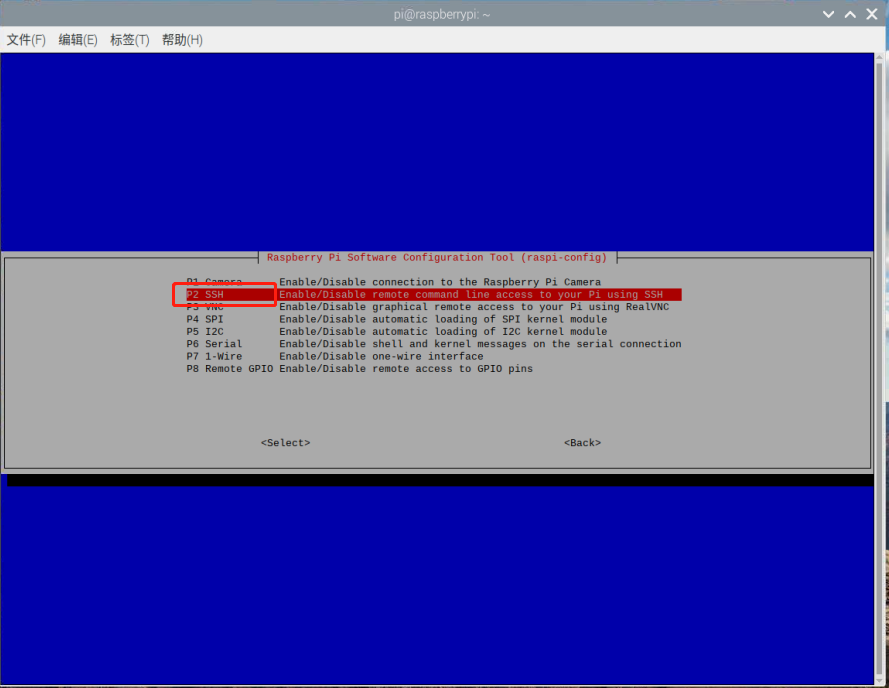
按“enter”键运行。

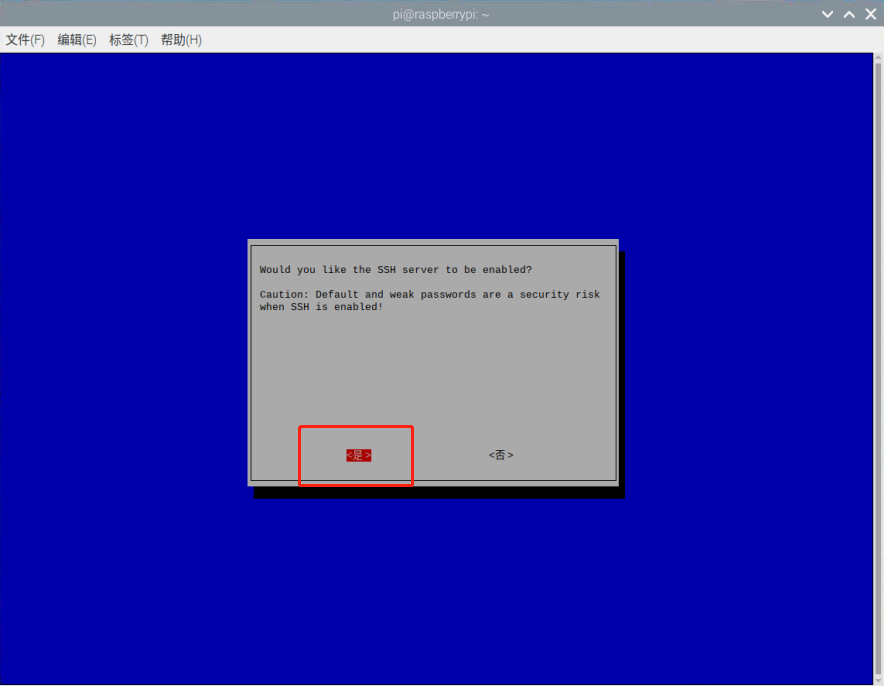


键盘下移移动到“Interfacing Options”，按“enter”键。



选择ssh，按yes保存，重启树莓派后就会自动开启SSH服务。





#### SSH通讯

将电脑和树莓派连入同一个网络使用putty进行电脑和树莓派SSH通讯

既然你已经启用了SSH功能并且知道了树莓派的 IP 地址，你可以从任何电脑 SSH 进入你的树莓派。

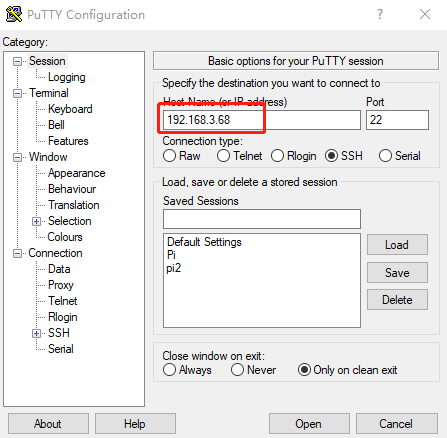
树莓派的默认用户名和密码是：

用户名：pi

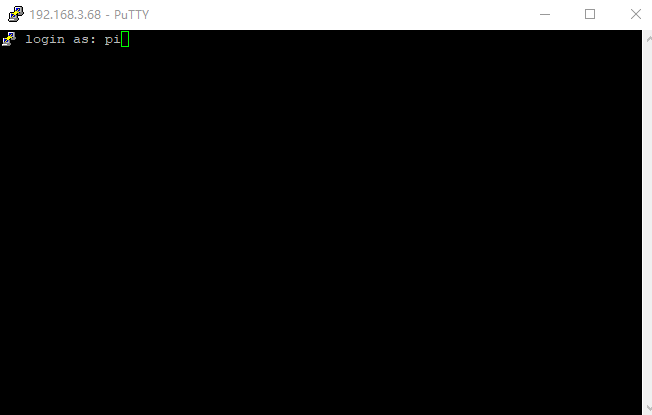
密码：raspberry

（注意：确保树莓派和电脑接入同一个网络）

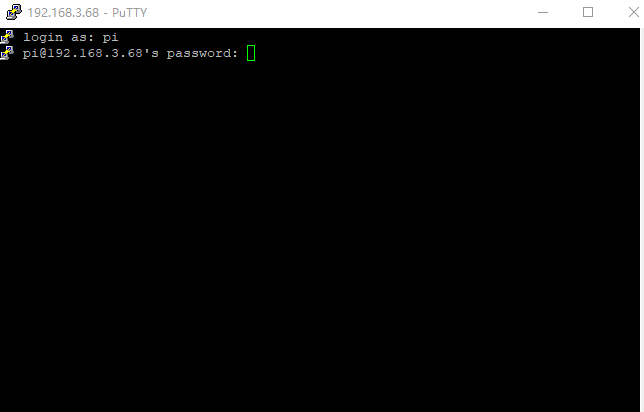
打开电脑端工具putty，在host name框中填入树莓派ip地址电机open



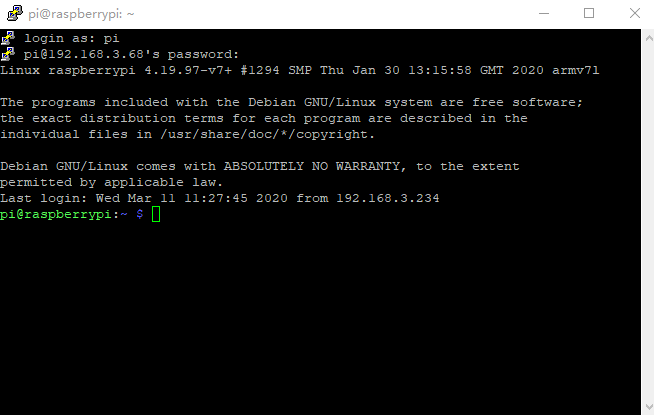
填入用户名并回车



填入密码（注意：密码不会显示，请输入完毕回车）



成功登入，现在你可以通过命令行远程访问你的树莓派了。



### 安装MegaPi库

在树莓派命令行输入下列命令并回车等待安装结束（如已安装可跳过此步骤）

sudo apt-get install MegaPi

### 安装树莓派vim

安装vim（如已安装可跳过此步骤）

在树莓派命令行输入一下命令并回车安装vim编辑器

sudo apt-get install vim

### 树莓派flask及简单应用

#### Flask简介：

Flask是一个使用python编写的轻量级Web应用框架，通过引入flask包，就可以把树莓派变成一个动态服务器

#### Flask安装：

在命令行输入

sudo apt-get install python-pip

等待安装完成在命令行输入

sudo pip install flask

#### flask最小应用测试：

在命令行输入下列命令并回车建立MegaPi文件夹

mkdir MegaPi

在命令行输入下列命令并回车进入MegaPi文件夹

cd MegaPi

在命令行输入下列命令并回车建立hello.py文件

touch hello.py

在命令行输入下列命令并回车进入hello.py文件进行编辑（vim编辑器使用方法请自行学习）

vim hello.py

在hello.py中输入以下代码：

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

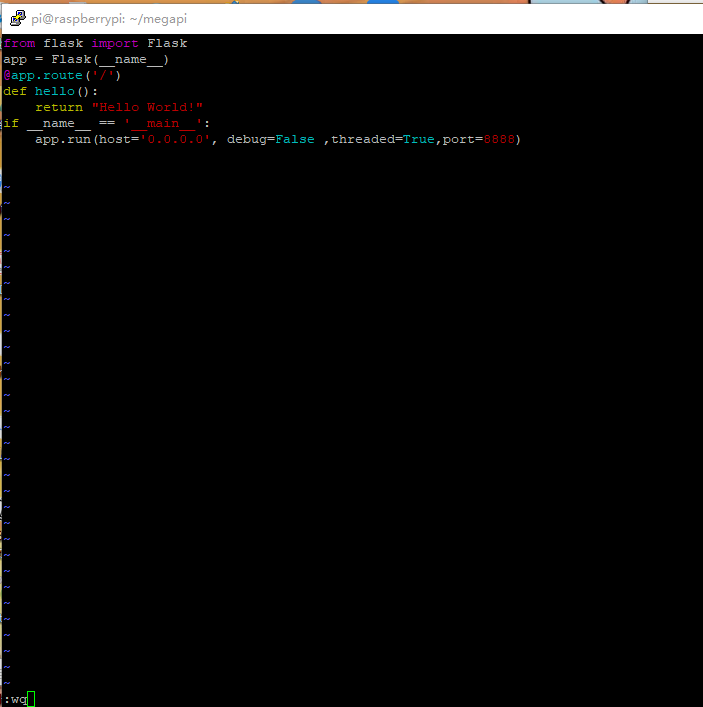
def hello():

return "Hello World!"

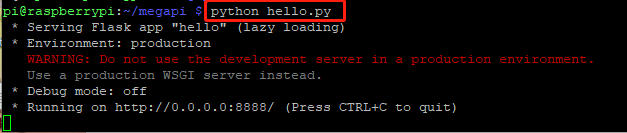
if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', debug=False ,threaded=True,port=8888)

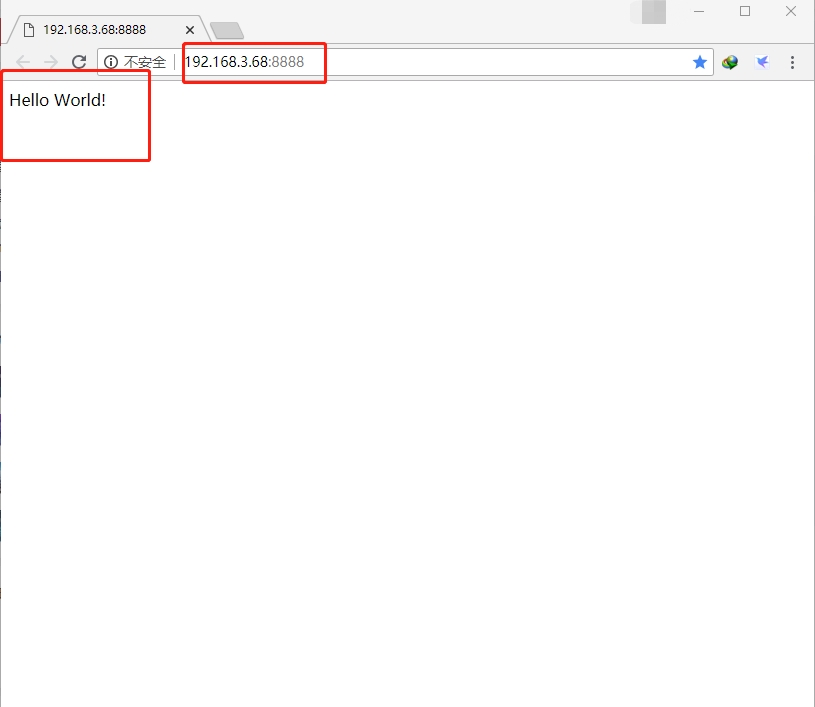
在vim编辑器编写完成后按下 Esc键，接着输入:wq 回车保存退出vim剪辑器



在命令行输入 python hello.py 并回车运行python文件



打开网页输入 树莓派地址及端口并回车就可以看到输出结果了，例如本树莓派 192.168.3.68:8888



代码作用解释：

from flask import Flask

app = Flask(\_\_name\_\_)

#创建一个名为app的Flask对象

@app.route(‘/’)

#当有人访问网页服务器的根目录时，执行下面的代码

def hello():

return "Hello World!"

#向客户端发送“Hello World!”字符串

if \_\_name\_\_ == '\_\_main\_\_':

#判断是否这个脚本是从命令行直接运行

app.run(host='0.0.0.0', debug=False ,threaded=True,port=8888)

## 案例：声音传感器数值在web端显示案例

### 文件目录结构

/static

　　/js

　　　　jquery.min.js

/templates

　　index.html

app.py

### 代码展示：

#### index.html代码：

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>树莓派传感器输入显示</title>

</head>

<body>

<h2>当前数值：<span id="g\_data"></span></h2>

<script src="/static/js/jquery.min.js" charset="utf-8"></script>

<script type="text/javascript">

// 前端每0.5s向后台发起ajax请求，获取当前传感器数值

setInterval(function() {

$.get('/read',function(data){

$("#g\_data").html(data)

})

},500)

</script>

</body>

</html>

#### app.py代码：

# encoding=utf-8

from flask import Flask, render\_template, Response, request

from MegaPi import \*

app = Flask(\_\_name\_\_)

#设置全局变量

g\_data = None

def onRead(data):

global g\_data

g\_data = data

print g\_data

# MegaPi 驱动初始化

bot = MegaPi()

bot.start('/dev/ttyUSB0')

#返回index.html页面

@app.route('/')

def index():

return render\_template("index.html")

#获取传感器最新值

@app.route('/read')

def onReadReq():

bot.soundSensorRead(7,onRead)

return str(g\_data )

print str(g\_data)

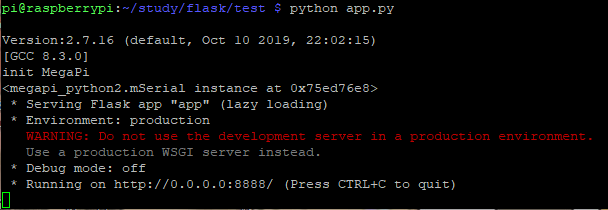
if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', debug=False ,threaded=True,port=8888)

### 代码运行及网页访问

在soundsensorweb文件夹下输入下列命令运行app.py

python app.py



打开浏览器输入目标地址查看结果，（如本树莓派192.168.3.68:8888）

