嵌入式课笔记

电脑睡眠后的密码：student

Linux下

输入telnet 192.168.208.113

Passin时输入root

密码为123456

或ssh root@192.168.208.113

然后输入密码：123456

即可连接如芯片终端(可以理解为此时电脑屏幕显示的终端其实换了一个CPU)

或者输入vncviewer 192.168.208.113打开图形界面

Windows端可以用putty来打开图形界面

cd /sys/class/gpio/

输入ls 会显示export …… unexport

查看可以激活的gpio（芯片背后的引脚？）（gpio为设备文件）

激活后的引脚可以当作可以输入输出的点（激活时默认为输入方式）

gpio：通用输入输出General Purpose Input Output

当把某gpio设置为输出模式时，该引脚可以输出高电压或者低电压（即1和0）

先ls一遍，可以看到RPi?GPOP

然后ls RPi?GPOP/

然后rm RPi?GPOP/ -rf

从<https://github.com/yfang1644/RPi.GPIO>上面clone下模块文件

用cd RPi.GPIO/进入该目录

python3 setup.py build

python3 setup.py install

用cd..从目录中退出

输入 rm -rf RPi.GPIO/

之后就可以用python3 filename来运行文件了

运行前需要在本地电脑终端中将文件传输如芯片的终端

使用scp filename [root@192.168.208.xxx](mailto:root@192.168.208.xxx)传输file

GPIO.setmode(GPIO.BOARD)和GPIO.setmode(GPIO.BCM)两种方式引入GPIO的引脚号，前者为物理的引脚排列，后者为逻辑排列（微信pdf里的那个图表，即2表示GPIO2的引脚而不是物理序号2的引脚）

import RPI.GPIO as G

import time

在电脑上装VNC Master就可以连接到芯片系统的图形界面

不要通过任何途径让树莓派的电源接地

红外传感器

vcc ：电源引脚(接3.3V那个)

GND：接地引脚

DO(Digital output)：数字信号输出(用这个)

AO：模拟信号输出(不使用)

两个灯泡一样的东西，一个是红外发射器，另一个是红外接收器

channel = GPIO.wait\_for\_edge(2, GPIO.FALLING, timeout = 5000)

用于检测某个变化是否发生

参数解释：2表示引脚2，GPIO.FALLING表示电平从高到低(RISING是从低到高)，timeout是监测时长，超过时长则返回None

event\_detected()函数，当我们需要观测的事件发生使，安排一个回调函数处理该事件

GPIO.add\_event\_detect(channel , GPIO.RISING)

do\_something()

if GPIO.event\_detected(channel):

print(‘Button pressed’)

加上回调函数GPIO.add\_event\_detect(channel , GPIO.RISING , callback = my\_callback)

其中my\_callback是一个需要使用者预先定义的函数，即检测到事件发生后执行回调函数来处理事件

加上去抖动GPIO.add\_event\_detect(channel , GPIO.RISING , callback = my\_callback , bouncetime = 200)，在指定回调函数时使用参数bouncetime来设定一个延时(ms)

声波传感器

vcc：电源引脚

Trig：发射声波控制引脚

Echo：接受回声引脚

Gnd：接地引脚

新线程用法：

为了避免不同版本里模块的差别，需要这么做

Import \_thread as thread (\_thread在python3里)

Import thread as thread (thread在python2里)

或者使用

Import threading (在python2和python3里名字一样，只是底层实现有些不同)

\_thread.start\_new\_thread (func, (“asasd”, 2))

//这个函数会创建一个新线程，第一个是新线程的主程序函数，第二个参数可有可无(如果第二个只有一个元素，则需要写成(para1,)的形式，括号里逗号不可少)，第二个参数是第一个参数的参数

python建议使用threading

t1 = threading.Thread(target = func, args = (“asdqwe”,2)) // Thread()方法首字母大写，要注意

//这里参数的意义与上面的\_thread的意义一样

t1.start()

t1.isAlive()

t1.join() //等待另一线程结束后再运行

横着看时，上下竖排的各5个孔是分上下连通的，5个孔中，一个做输入，其他四个可以做输出，不允许两个不同输出同时接到一个接受器里

小车的那个电路板（电流分配板），大头在下面，对着自己看

下面的蓝色的一排中out4 out3 out2 out1是4个输出，对应板上方的in4 in3 in2 in1

输入的电压不变，电流变大了，从而让功率变大

GND是地

有一根一头大另一头有红黑两个细头的线，黑色细头是地线，橘红细头的接在VDD上

上方的GND和+5两个接口间电压差为5V

侧面有也两个蓝色柱子GND和+5，是个给数字系统供电的

建议树莓派和其他元件的供电分开

充电器，指示灯红色表示在充电，绿色表示充完了

网络连接

服务器

import socket

sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_addr = (‘’/\*IP地址\*/, 10000/\*端口号\*/)

sock.bind(server\_addr)

socket.listen(5)/\*最大服务数\*/

conn, client\_addr = sock.accept()

while True:

data = conn.recv(100)

print(data.decode())

用户

import socket

import sys

sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_addr = (‘127.0.0.1’, 10000)

socket.connet(server\_addr)

while True:

x = input(“xxx:”)

sock.send(x.encode())