参考文献：

<http://shumeipai.nxez.com/2016/09/28/rpi-gpio-module-outputs.html>

<http://shumeipai.nxez.com/2016/09/28/rpi-gpio-module-inputs.html>

<http://shumeipai.nxez.com/2016/09/28/rpi-gpio-module-basics.html>

<http://shumeipai.nxez.com/2014/11/13/rpi-gpio-module-pwm-basic-function.html>

<http://shumeipai.nxez.com/raspberry-pi-pins-version-40>

RPi.GPIO 模块不适合应用到追求实时性或计数周期的应用中。

**导入模块**

Import RPi.GPIO as GPIO

**针脚编号**

BOARD编码系统

GPIO.setmode(GPIO.BOARD)

BCM编码系统

GPIO.setmode(GPIO.BCM)



**警告**

GPIO.setwarnings(False)#取消警告

**配置通道**

输入：GPIO.setup(channel,GPIO.IN)

上拉或下拉：GPIO.setup(channel,GPIO.IN,pull\_up\_down=GPIO.PUD\_UP/GPIO.PUD\_DOWN)

输出：GPIO.setup(channel,GPIO,OUT)

指定输出初始化状态：GPIO.setup(channel,GPIO.OUT,initial=GPIO.HIGH)

**输入**

GPIO.input(channel)

**轮训（polling）**

if GPIO.input(channel):

  print('Input was HIGH')

else:

  print('Input was LOW')

或

while GPIO.input(channel) == GPIO.LOW:

  time.sleep(0.01)  # 为 CPU 留出 10 毫秒，供其处理其它事物

**中断和边沿检测**

**wait\_for\_edge() 函数**

GPIO.wait\_for\_edge(channel,GPIO.RISING/GPIO.FALLING/GPIO.BOTH)

**event\_detected() 函数**

GPIO.add\_event\_detect(channel, GPIO.RISING)#在通道上添加上升临界值检测

do\_something()

if GPIO.event\_detected(channel):

 print('Button pressed')

**线程回调**

def my\_callback(channel):

    print('这是一个边缘事件回调函数！')

    print('在通道 %s 上进行边缘检测'%channel)

    print('该程序与您的主程序运行在不同的进程中')

GPIO.add\_event\_detect(channel, GPIO.RISING, callback=my\_callback)# 在通道上

多个回调函数

def my\_callback\_one(channel):

    print('回调 1')

def my\_callback\_two(channel):

    print('回调 2')

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

GPIO.add\_event\_callback(channel, my\_callback\_one)

GPIO.add\_event\_callback(channel, my\_callback\_two)

**开关防抖**

# 在通道上添加上升临界值检测，忽略由于开关抖动引起的小于 200ms 的边缘操作

GPIO.add\_event\_detect(channel, GPIO.RISING, callback=my\_callback, bouncetime=200)

**停止边缘检测**

GPIO.remove\_event\_detect(channel)

**输出**

GPIO.output(channel,state)

GPIO.output(12, not GPIO.input(12))# 使用 input() 函数读取IO口状态

**清理**

清理用过的资源(引脚默认为输入)

GPIO.cleanup()

**检测Raspberry Pi修订版本和RPi.GPIO版本**

GPIO.RPI\_REVISION

GPIO.VERSION

**PWM波**

创建一个PWM实例

P = GPIO.PWM(channel,frequency)

启动PWM

p.start(dc)#dc为占空比（范围：0.0 <= dc <= 100.0）

更改频率

p.ChangeFrequency(freq)

更改占空比

p.ChangeDutyCycle(dc)

停止PWM

p.stop()

