

# 树莓派脱盲

---Python控制外设

# 引脚资源图

树莓派 40Pin 引脚对照表

wiringPi 编码	BCM 编码	功能名	物理引脚 BOARD编码		功能名	BCM 编码	wiringPi 编码
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29

表格由树莓派实验室绘制 <http://shumeipai.nxez.com>

# GPIO

- 使用RPI.GPIO库

```
import RPi.GPIO as GPIO
```

树莓派引脚有BOARD和BCM两种编号方式, 我们一般都是用BCM编号

- 查看引脚

```
gpio readall
```

- 查看第三方库源码

- dir命令

- API

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setwarnings(False)
```

- 配置引脚

```
setup(channel, GPIO.IN)
```

```
setup(channel, GPIO.OUT, initial=GPIO.HIGH)
```

# GPIO

- 输出电平(GPIO.HIGH,GPIO.LOW)

output(channel, state)

- 输入

input(channel)

setup(channel,GPIO.IN,pull\_up\_down=GPIO.PUD\_UP)

setup(channel,GPIO.IN,pull\_up\_down=GPIO.PUD\_DOWN)

- 中断(RISING, FALLING, BOTH)

GPIO.add\_event\_detect(channel, GPIO.RISING, callback=my\_cb)

# GPIO

- PWM (软件PWM, 精度不高, 不能做精确控制)

PWM(channel,frequency)

start(duty)

ChangeFrequency(freq)

ChangeDutyCycle(duty)

stop

- 恢复

cleanup

# IIC

- i2c\_detect -y
- 使用smbus协议访问i2c
- import smbus
- API
  - SMBus(bus\_num)
  - write\_quick(addr)
  - read\_byte(addr)
  - write\_byte(addr, val)
  - read\_byte\_data(addr, reg)
  - write\_byte\_data(addr, reg, val)

# IIC

`read_word_data(addr,reg)`

`write_word_data(addr,reg,val)`

`read_block_data(addr, reg)`

`write_block_data(addr,reg,vals)`

- 看一下如何驱动urm09
- 自己编写i2cdetect

# SPI

- 使用spidev模块驱动spi外设

```
import spidev
```

- bus\_num和device\_num

- API

```
SpiDev()
```

```
open(bus_num,device_num)
```

```
readbytes(n)
```

```
writebytes(vals)
```



# SPI

```
writebytes2(vals)
```

```
xfer(vals,[speed_hz,delay_usec,bit_per_word])
```

```
xfer2(vals,[speed_hz,delay_usec,bit_per_word])
```

```
xfer3(vals,[speed_hz,delay_usec,bit_per_word])
```

```
close()
```

- 回环传输
- 驱动ST7789

# UART

- import serial
- API

```
Serial("/dev/ttyS0",115200,timeout=1)  
read(n)  
readline()  
write(string)  
inWaiting() in_waiting  
flush()  
close()
```

# UART

setRTS(level=True) 参考SSCOM

setDTR(level=True)

- 获取串口列表
- 写一个回环程序
- 驱动一个传感器(作业)

# 驱动与应用

- 驱动与应用都可以操作硬件资源
- 它们的使用方式
- 它们各自的目的

# USB口

- 常见树莓派usb外设的驱动原理
- usb 串口
- u盘
- hid设备, 键盘鼠标
- usb液晶屏幕
- usb触摸屏
- rndis设备

# 其他接口（显示部分再讲解）

- HDMI
- DSI
- CSI
- ETH