树莓派基础外设 - GPIO

主讲人：曹家英

第一部分：树莓派底层硬件

- 1. 树莓派介绍
- 2. 树莓派环境搭建
- 3. 树莓派与命令行
- 4. 相关知识扩展
- 5. 树莓派基础外设操作
- 6. 树莓派Buildroot SDK环境搭建

摩尔吧 MOORE 8

课程概览

- 什么是GPIO?
- Python3实现GPIO
- 实战1——LED
- 实战2——Button
- wiringPi介绍

摩尔吧 MOORE 8

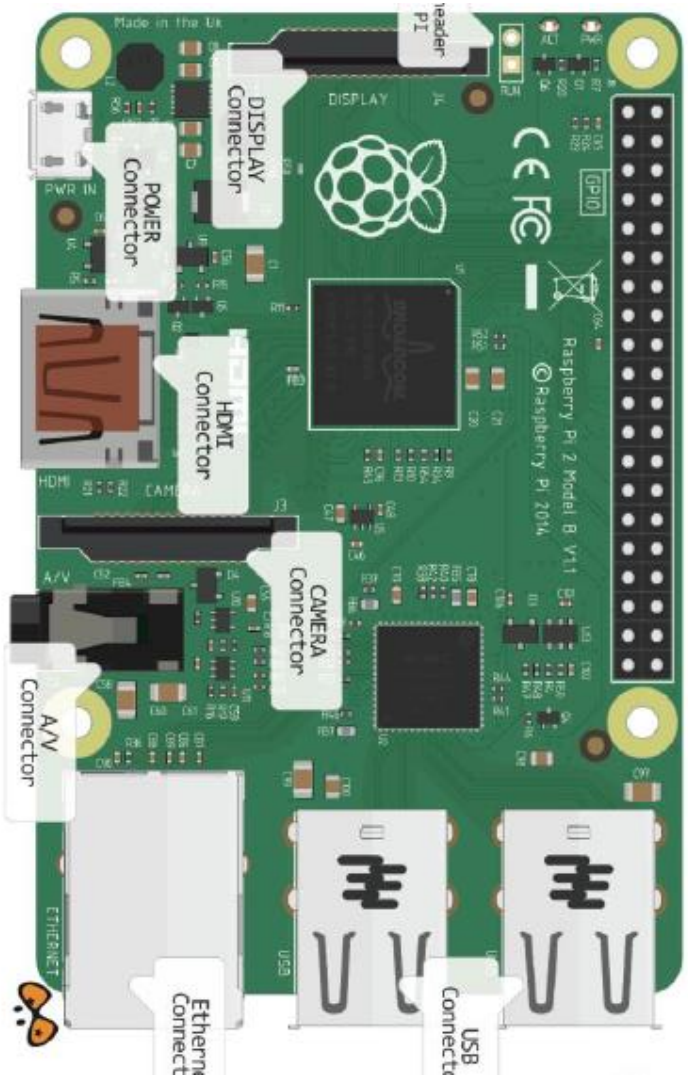
什么是GPIO?

什么是GPIO

GPIO（General Purpose Input Output）：通用输入/输出，简称为GPIO，功能类似8051的P0—P3，其接脚可以供使用者由程控自由使用，PIN脚依现实考量可作为通用输入（GPI）或通用输出（GPO）或通用输入与输出（GPIO），如当clk generator, chip select等。

GPIO的用途：

1. 开关用途，控制GPIO设备，比如LED，Button
2. 模拟总线，通过GPIO模拟SPI或者I2C
3. 实时性需求高的情况下，提供中断功能。



寄存器的概念

寄存器（Register）：是中央处理器内的其中组成部分。寄存器是有限存储容量的高速存存储部件，它们可用来暂存指令、数据和地址。在中央处理器的控制部件中，包含的寄存器有指令寄存器（IR）和程序计数器。在中央处理器的算术及逻辑部件中，包含的寄存器有累加器。

硬件寄存器：是不同种类的硬件I/O (输出/输入) 的存储区域。这些硬件寄存器是包含在某些周边组件之内，使用存储器映射或 端口映射在计算机的中央处理器CPU中出现。

Address	Field Name	Description	Size	Read/ Write
0x 7E20 0000	GPFSEL0	GPIO Function Select 0	32	R/W
0x 7E20 0000	GPFSEL0	GPIO Function Select 0	32	R/W
0x 7E20 0004	GPFSEL1	GPIO Function Select 1	32	R/W
0x 7E20 0008	GPFSEL2	GPIO Function Select 2	32	R/W
0x 7E20 000C	GPFSEL3	GPIO Function Select 3	32	R/W
0x 7E20 0010	GPFSEL4	GPIO Function Select 4	32	R/W
0x 7E20 0014	GPFSEL5	GPIO Function Select 5	32	R/W
0x 7E20 0018	-	Reserved	-	-
0x 7E20 001C	GPSET0	GPIO Pin Output Set 0	32	W
0x 7E20 0020	GPSET1	GPIO Pin Output Set 1	32	W
0x 7E20 0024	-	Reserved	-	-

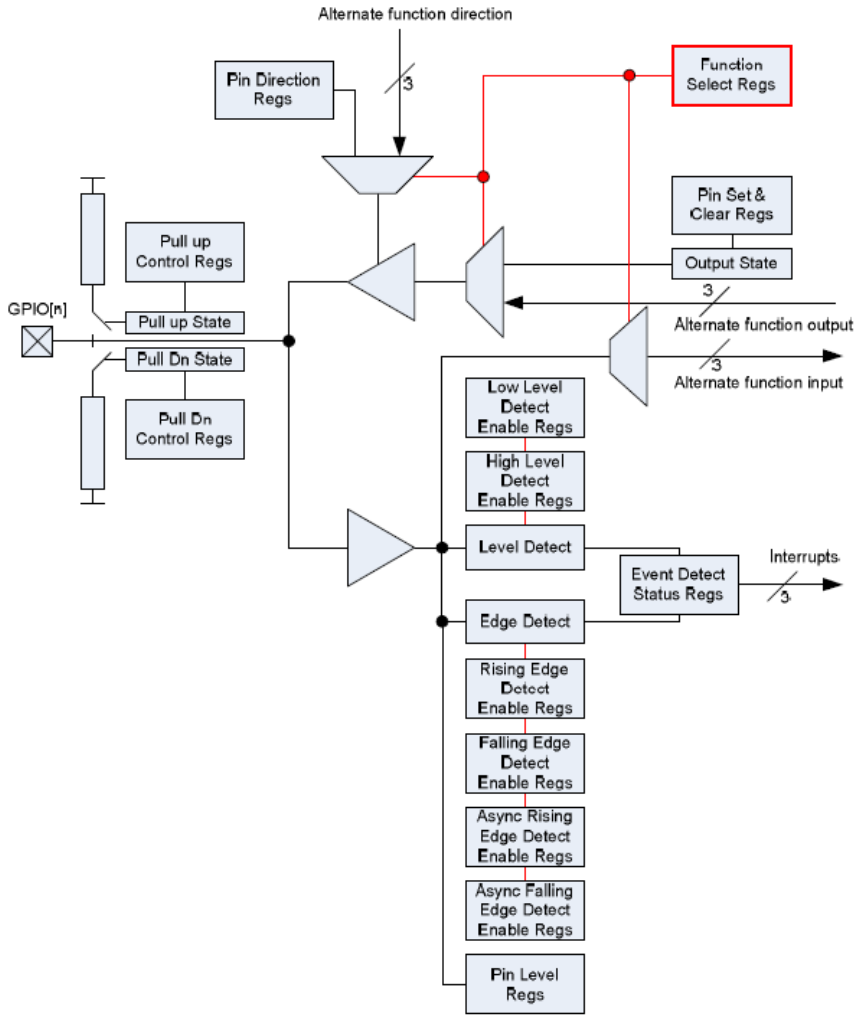


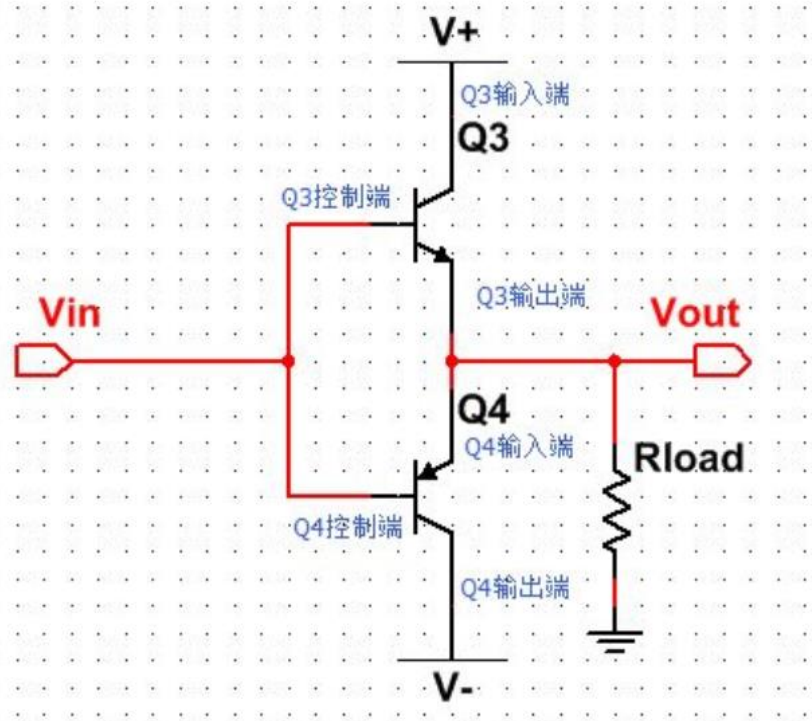
Figure 6-1 GPIO Block Diagram

摩尔吧 MOORE 8

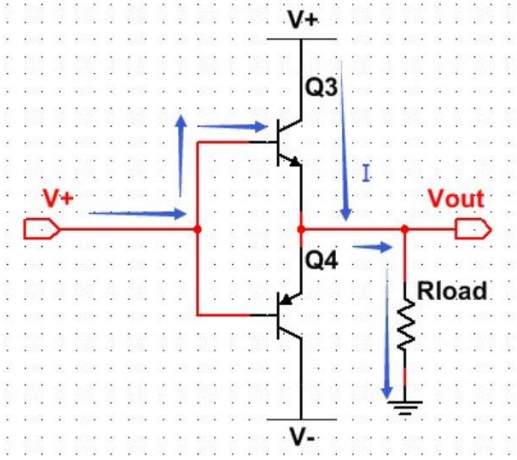
GPIO的实现模型

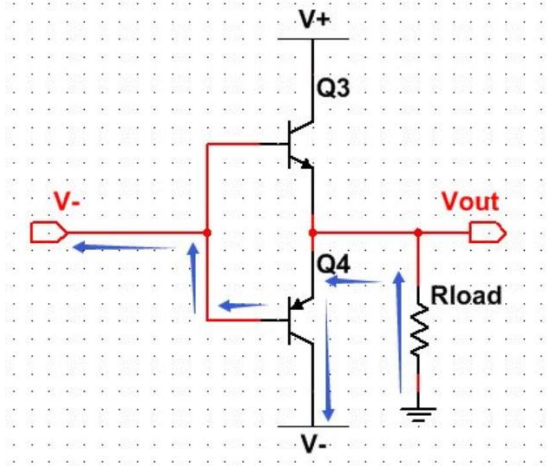
一般来说：GPIO的工作模式包括以下几种

（1）GPIO_Mode_AIN 模拟输入
（2）GPIO_Mode_IN_FLOATING 浮空输入
（3）GPIO_Mode_IPD 下拉输入
（4）GPIO_Mode_IPU 上拉输入
（5）GPIO_Mode_Out_OD 开漏输出
（6）GPIO_Mode_Out_PP 推挽输出
（7）GPIO_Mode_AF_OD 复用开漏输出
（8）GPIO_Mode_AF_PP 复用推挽输出



推挽输出：
上面的三极管是N型三极管，下面的三极管是P型三极管。





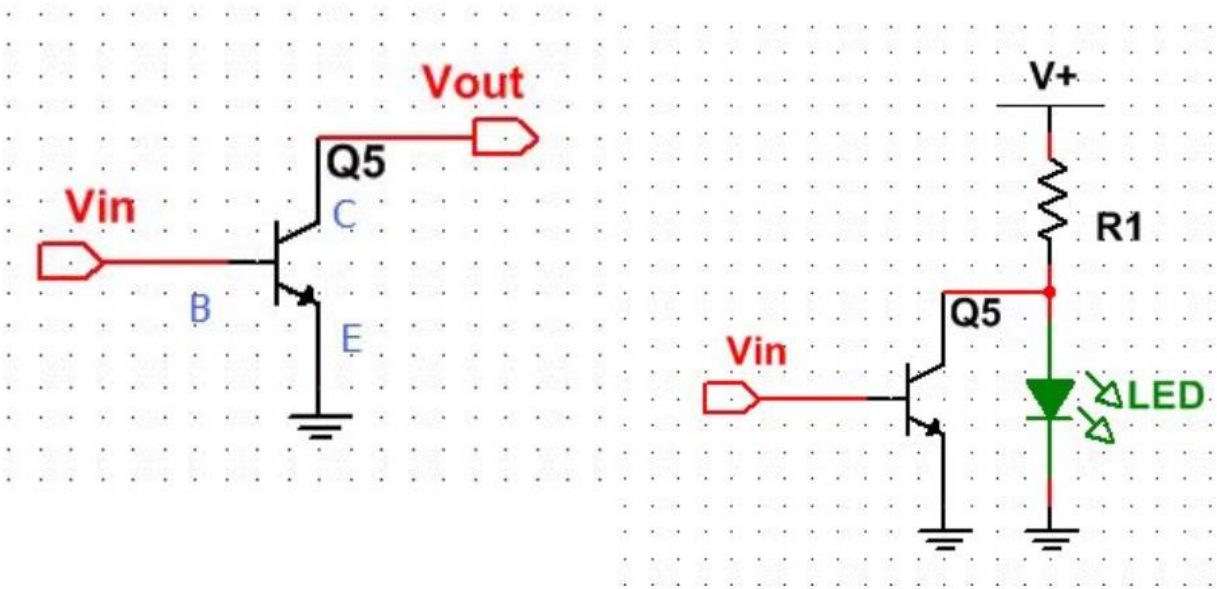
推挽输出的最大特点是可以真正能真正的输出高电平和低电平，在两种电平下都具有驱动能力。

摩尔吧 MOORE8

GPIO的实现模型

开漏输出：
一个N型场效应管（MOSFET）开集，C集直接输出。

常说的与推挽输出相对的就是开漏输出，对于开漏输出和推挽输出的区别最普遍的说法就是开漏输出无法真正输出高电平，即高电平时没有驱动能力，需要借助外部上拉电阻完成对外驱动。下面就从内部结构和原理上说明为什么开漏输出输出高电平时没有驱动能力，以及进一步比较与推挽输出的区别。



开漏输出最主要的特性就是高电平没有驱动能力，需要借助外部上拉电阻才能真正输出高电平

	推挽输出	开漏输出
高电平驱动能力	强	由外部上拉电阻提供
低电平驱动能力	强	强
电平跳变速度	快	由外部上拉电阻决定
电瓶转换	不支持	支持

摩尔吧 MOORE8

树莓派上的GPIO

如右图：

在树莓派的40PIN接口中，有BCM或wiriingPi编码的管脚均可用作GPIO功能。

40PIN的管脚中，有些GPIO是与其他功能复用的如：

BCM编码2号管角就与I2C的SDA功能复用

BCM编码10号管脚就与SPI的MOSI功能复用。

所谓的复用功能就是这个管角既可以用作这个功能，也可以用作那个功能，但是同一个时刻只能使用一个功能。

在默认情况下，管角都是GPIO INPUT的功能。

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

NXEZ.COM

摩尔吧 MOORE8

Python3实现GPIO

树莓派Python的GPIO库

import RPi.GPIO as GPIO

导入Rpi.GPIO 的模块

GPIO.setmode(GPIO.BOARD)
GPIO.setmode(GPIO.BCM)

配置引脚编号模式

GPIO.setwarnings(False)

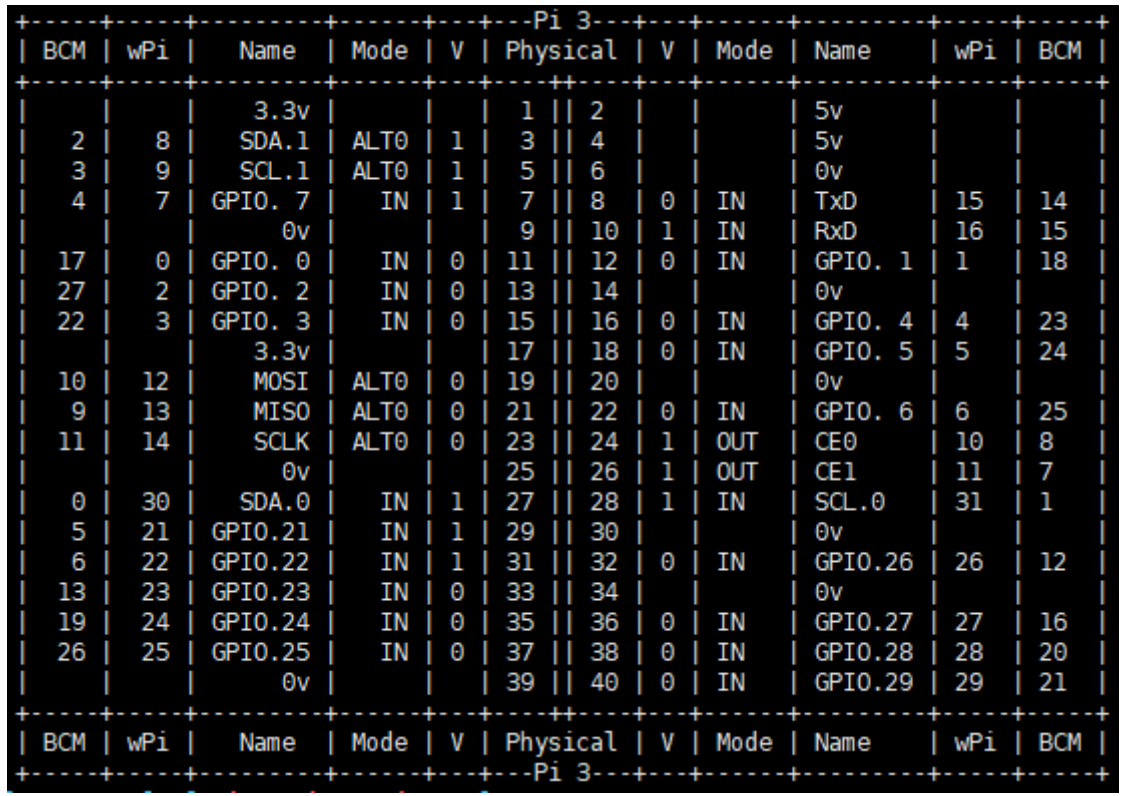
Rpi.GPIO会监测GPIO是否为默认状态如果不是会提出警告，这里禁用警告

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

设置Channel的方向

chan_list = [11,12]
GPIO.setup(chan_list, GPIO.OUT)

批量配置GPIO



BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
		3.3v			1 || 2			5v		
2	8	SDA.1	ALT0	1	3 || 4			5v		
3	9	SCL.1	ALT0	1	5 || 6			0v		
4	7	GPIO. 7	IN	1	7 || 8	0	IN	TxD	15	14
		0v			9 || 10	1	IN	RxD	16	15
17	0	GPIO. 0	IN	0	11 || 12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13 || 14			0v		
22	3	GPIO. 3	IN	0	15 || 16	0	IN	GPIO. 4	4	23
		3.3v			17 || 18	0	IN	GPIO. 5	5	24
10	12	MOSI	ALT0	0	19 || 20			0v		
9	13	MISO	ALT0	0	21 || 22	0	IN	GPIO. 6	6	25
11	14	SCLK	ALT0	0	23 || 24	1	OUT	CE0	10	8
		0v			25 || 26	1	OUT	CE1	11	7
0	30	SDA.0	IN	1	27 || 28	1	IN	SCL.0	31	1
5	21	GPIO.21	IN	1	29 || 30			0v		
6	22	GPIO.22	IN	1	31 || 32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33 || 34			0v		
19	24	GPIO.24	IN	0	35 || 36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37 || 38	0	IN	GPIO.28	28	20
		0v			39 || 40	0	IN	GPIO.29	29	21
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM

摩尔吧 MOORE8

树莓派Python的GPIO库

GPIO用作输出：
GPIO.output(channel, state)
　　配置channel的输出状态（高低电平）
　　state： PIO.LOW， GPIO.HIGH

chan_list = [11,12]
GPIO.output(chan_list, GPIO.LOW)
GPIO.output(chan_list, (GPIO.HIGH, GPIO.LOW))
　　批量配置GPIO输出电平

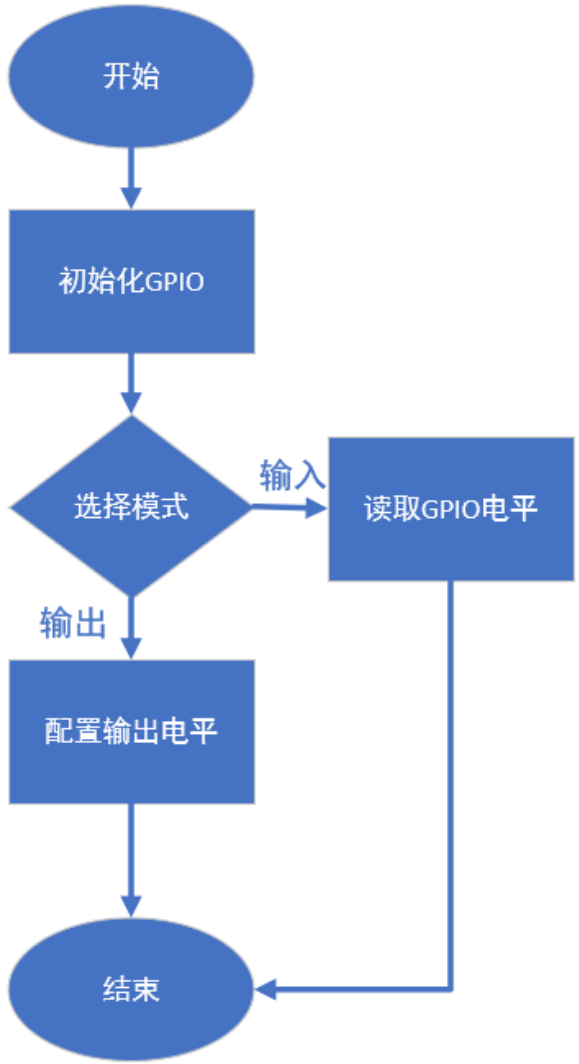
GPIO用作输入：
GPIO.setup(channel, GPIO.IN, pull_up_down=GPIO.PUD_UP)
　　配置channel为输入， 内部上拉

value = GPIO.input(channel)
　　读取GPIO当前电平

GPIO.cleanup()
　　清除GPIO当前状态。

摩尔吧 MOORE 8

树莓派GPIO使用流程



摩尔吧 MOORE8

实战1——GPIO Output LED

实战流程

1. 通过使用Python脚本语言的特性直接在Python shell中实现LED功能
2. 通过编写.py文件来实现LED功能
3. 将LED功能封装成类方便我们之后的调用

摩尔吧 MOORE 8

实战2——GPIO Input Button

使用的代码

1. 通过使用Python脚本语言的特性直接在Python shell中实现Button功能
2. 通过编写.py文件来实现Button功能
3. 将LED功能封装成类方便我们之后的调用

摩尔吧 MOORE 8

wiringPi介绍

wiringPi的介绍

C语言调试，我们透过wiringPi库来实现我们案例：

wiringPi库是由Gordon Henderson所编写开发维护的一个用C语言写成的类库。起初，主要是作为BCM2835芯片的GPIO库。现在，已经非常丰富，除了GPIO库，还包括了I2C库、SPI库、UART库和软件PWM库等。

wiringPi库包含了一个命令行工具gpio，它可以用来设置GPIO管脚，可以用来读写GPIO管脚，甚至可以在Shell脚本中使用来达到控制GPIO管脚的目的。

官方网站： http://wiringpi.com/

BCM2835 lib

```
[ 12:35am ]  [ pi@raspberrypi:~/c ]
$ gpio readall
 +-----+-----+---------+------+---+---Pi 3+--+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
 |   2 |   8 |   SDA.1 | ALT0 | 1 |  3 || 4  |   |      | 5v      |     |     |
 |   3 |   9 |   SCL.1 | ALT0 | 1 |  5 || 6  |   |      | 0v      |     |     |
 |   4 |   7 | GPIO. 7 |   IN | 1 |  7 || 8  | 0 | IN   | TxD     | 15  | 14  |
 |     |     |      0v |      |   |  9 || 10 | 1 | IN   | RxD     | 16  | 15  |
 |  17 |   0 | GPIO. 0 |  OUT | 1 | 11 || 12 | 1 | ALT0 | GPIO. 1 | 1   | 18  |
 |  27 |   2 | GPIO. 2 |   IN | 1 | 13 || 14 |   |      | 0v      |     |     |
 |  22 |   3 | GPIO. 3 |   IN | 1 | 15 || 16 | 0 | IN   | GPIO. 4 | 4   | 23  |
 |     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 | 5   | 24  |
 |  10 |  12 |    MOSI | ALT0 | 1 | 19 || 20 |   |      | 0v      |     |     |
 |   9 |  13 |    MISO | ALT0 | 0 | 21 || 22 | 0 | IN   | GPIO. 6 | 6   | 25  |
 |  11 |  14 |    SCLK | ALT0 | 0 | 23 || 24 | 1 | OUT  | CE0     | 10  | 8   |
 |     |     |      0v |      |   | 25 || 26 | 1 | OUT  | CE1     | 11  | 7   |
 |   0 |  30 |   SDA.0 |   IN | 1 | 27 || 28 | 1 | IN   | SCL.0   | 31  | 1   |
 |   5 |  21 | GPIO.21 |  OUT | 1 | 29 || 30 |   |      | 0v      |     |     |
 |   6 |  22 | GPIO.22 |  OUT | 1 | 31 || 32 | 0 | IN   | GPIO.26 | 26  | 12  |
 |  13 |  23 | GPIO.23 |   IN | 0 | 33 || 34 |   |      | 0v      |     |     |
 |  19 |  24 | GPIO.24 | ALT0 | 1 | 35 || 36 | 0 | IN   | GPIO.27 | 27  | 16  |
 |  26 |  25 | GPIO.25 |   IN | 1 | 37 || 38 | 0 | ALT0 | GPIO.28 | 28  | 20  |
 |     |     |      0v |      |   | 39 || 40 | 0 | ALT0 | GPIO.29 | 29  | 21  |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+---Pi 3+--+---+------+---------+-----+-----+
[  1:24am ]  [ pi@raspberrypi:~/c ]
$ gpio -v
gpio version: 2.46
Copyright (c) 2012-2018 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
  Type: Pi 3+, Revision: 03, Memory: 1024MB, Maker: Sony
  * Device tree is enabled.
  *--> Raspberry Pi 3 Model B Plus Rev 1.3
  * This Raspberry Pi supports user-level GPIO access.
```

摩尔吧 MOORE 8

wiringPi的安装

安装：

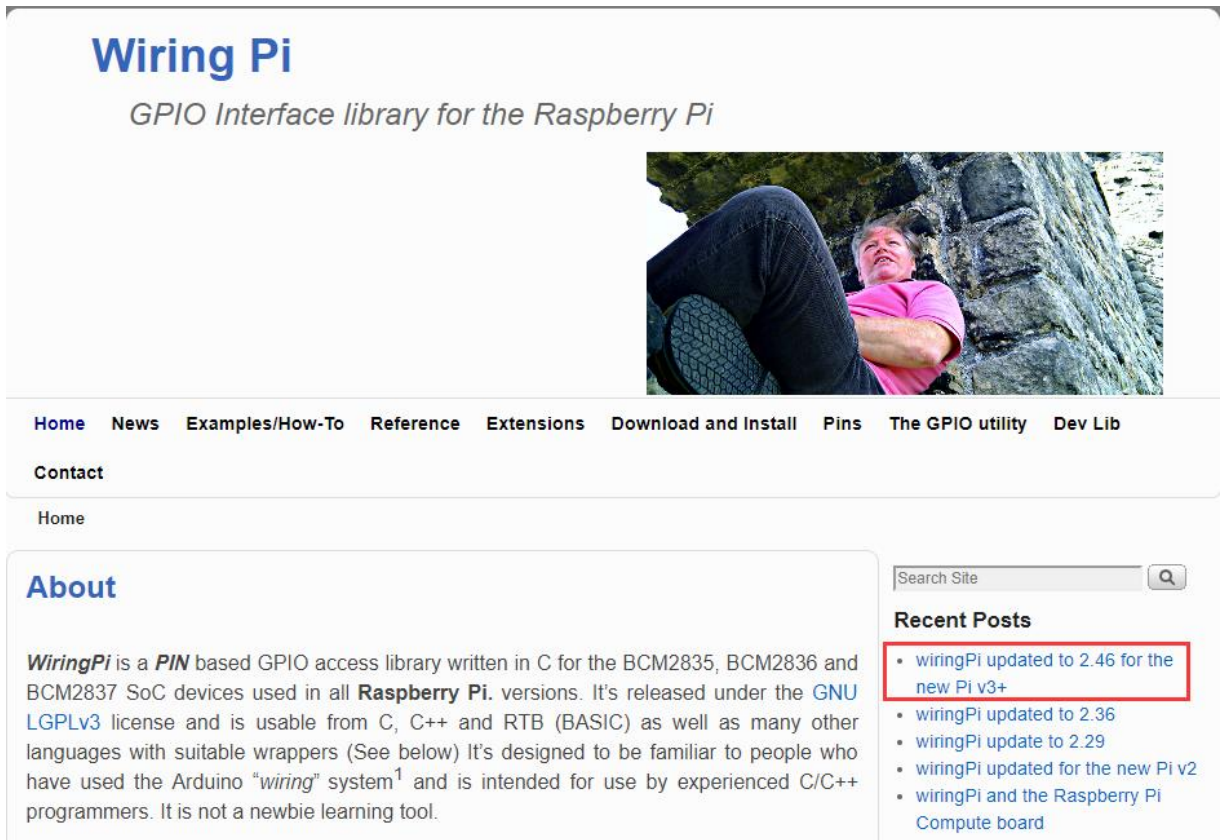
一般树莓派默认会自带wiringPi，版本可能会比较低，如果使用新版本的RPi 3B+，可能会出现不支持板卡的现象，这样我就需要重新安装wiringPi。方法如下：

1. 查看当前wiringPi版本：gpio -v
2. 更新源：sudo apt-get update
3. 更新wiringPi：sudo apt-get install wiringpi
4. 再次查看wiringPi版本： gpio –v

确定当前版本为2.46版本。

2.46版本的更新主要是support Pi V3+

```
[ 11:27pm ]  [ pi@raspberrypi:~/c ]
 $ gpio readall
Oops - unable to determine board type... model: 13
```



摩尔吧 MOORE8

wiringPi的使用函数

初始化：

wiringPiSetup()　　：初始化wiringPi使用wiringPi管脚定义
wiringPiSetupGpio()　　：初始化wiringPi使用BCM管脚定义
wiringPiSetupPhys ()　　：初始化wiringPi使用HW管脚定义（BOARD）
wiringPiSetupSys ()　　：使用sys接口，而非直接操作寄存器

核心函数：

void pinMode(int pin, int mode);

配置管脚工作模式。
pin为管角号码，根据初始化的模式选择管角编码。
mode为工作模式：INPUT，OUTPUT，PWM_OUTPUT，GPIO_CLOCK。
需要注意的是仅有管脚1（BCM_GPIO 18）支持PWM_OUTPUT模式，仅有管脚7（BCM_GPIO 4）支持CLOCK输出模式。

void pullUpDnControl(int pin, int pud);

配置管脚的上下拉电阻。
pin同上。
pud为上下拉电阻模式：PUD_OFF，PUD_DOWN，PUD_UP。
内部上拉和下拉电阻有接近50KΩ。
该函数在Sys模式下无作用。如果你需要激活上拉或下拉电阻的话，在启动程序前，可以通过在脚本中调用gpio命令来实现。

摩尔吧 MOORE8

wiringPi的安装

void digitalWrite(int pin, int value);

配置管脚的输出值，需要优先将管角设置为OUTPUT。

pin同上。

value：可配置HIGH（高）或者LOW（低）

void digitalRead(int pin);

读取管脚上的电平值。

void pwmWrite(int pin, int value);

使用该函数可以将值写入指定管脚的PWM寄存器中。

树莓派板上仅有一个PWM管脚，即wiringPi管脚1（BCM_GPIO 18，物理管脚号为12）。可以配置的值为0~1024。

当在Sys模式时，该函数不可以控制树莓派的板上PWM。

int piBoardRev(void);

该函数远回树莓派的硬件版本。

摩尔吧 MOORE 8

使用C语言通过wiringPi库操作GPIO

void digitalWrite(int pin, int value);

配置管脚的输出值，需要优先将管角设置为OUTPUT。

pin同上。

value：可配置HIGH（高）或者LOW（低）

void digitalRead(int pin);

读取管脚上的电平值。

void pwmWrite(int pin, int value);

使用该函数可以将值写入指定管脚的PWM寄存器中。

树莓派板上仅有一个PWM管脚，即wiringPi管脚1（BCM_GPIO 18，物理管脚号为12）。可以配置的值为0~1024。

当在Sys模式时，该函数不可以控制树莓派的板上PWM。

int piBoardRev(void);

该函数远回树莓派的硬件版本。

摩尔吧 MOORE8

谢谢您的观看

www.moore8.com

微信公众平台：
moore_8

QQ群： 327350729