



前面的文章我们介绍了WiringPi的配置函数，今天这篇文章，我们来了解下核心函数。



1、void pinMode(int pin,int mode)

该函数用来设置树莓派引脚的工作模式，共有两个参数：

第一个参数是引脚的编号，具体采用哪种编号规则由配置函数决定；

第二个参数是要设置的工作模式，共有四种工作模式可以选择，包括：输入(INPUT)、输出(OUTPUT)、PWM输出(PWM_OUTPUT)或者GPIO时钟(GPIO_CLOCK)。

需要注意的是：

树莓派3B+、树莓派4总共支持2路PWM硬件输出，分别是PWM0和PWM1。

其中：PWM0可以使用**wiringPi** pin1(BCM_GPIO18)或者**wiringPi** pin26(BCM_GPIO12)；PWM1可以使用**wiringPi** pin23(BCM_GPIO13)或者**wiringPi** pin24(BCM_GPIO19)；

总共支持3路GPIO时钟输出，分别是：GPCLK0、GPCLK1、GPCLK2；

其中：GPCLK0使用**wiringPi** pin7(BCM_GPIO4)；GPCLK1使用**wiringPi** pin21(BCM_GPIO5)；GPCLK2使用**wiringPi** pin22(BCM_GPIO6)；

举个例子：将wiringPi的pin1设置为PWM_OUTPUT

```
7 //initialize wiringPi Pin defination
8 wiringPiSetup();
9 //initialize pin1 as PWM OUTPUT
10 pinMode(1,PWM_OUTPUT);
```

2、void pullUpDnControl (int pin, int pud)

每个树莓派引脚内部都有一个上拉电阻和下拉电阻，阻值大约是50KΩ。当引脚被配置为输入(INPUT)模式，使用函数pullUpDnControl来激活其内部的上拉电阻或下拉电阻。

该函数有两个参数：第一个是引脚的编号；第二个是要设置的模式，包括三种：

PUD_OFF、**PUD_DOWN**和**PUD_UP**；

PUD_OFF不激活引脚的任何电阻；

PUD_DOWN激活引脚的下拉电阻，下拉至GND；

PUD_UP激活引脚的上拉电阻，上拉至3.3V；

举个例子：

将wiringPi的pin0设置为输入模式，并激活上拉电阻：

```
//initialize wiringPi Pin defination
wiringPiSetup();
//initialize pin0 as INPUT
pinMode(0, INPUT);
//active pin0 as pull up
pullUpDnControl(0, PUD_UP);
```

3、void digitalWrite (int pin, int value) ;

该函数用来将输出值**HIGH**(高电平)或**LOW**(低电平)写入到指定的引脚，该引脚必须提前配置为输出(OUTPUT)模式；

函数有两个参数：

第一个参数pin是引脚编号；

第二个参数value是要写入的值，可以是**HIGH**或者**LOW**；如果写入的是数值，wiringPi认为任何非0的值都是HIGH，而0为LOW；

4、void pwmWrite (int pin, int value)

该函数用来将PWM脉宽时间值写入到PWM寄存器中，有两个参数：

第一个参数pin表示引脚的编号，注意我们前面介绍的硬件PWM引脚；

第二参数value是要发送的脉宽时间值，其取值范围为0~1024；

举个例子：

将wiringPi的pin1设置为PWM输出，并输出占空比为100/1024的方波。

```
//initialize wiringPi Pin defination
wiringPiSetup();
//initialize pin1 as PWM output
pinMode(1, PWM_OUTPUT);
//output 100 of 1024
pwmWrite(1, 100); //输出占空比100/1024的方波
```

注：要持续输出方波需要使用循环语句。

5、int digitalRead (int pin)

该函数用来返回特定引脚的电平信号。

参数pin用来指定引脚的编号；

返回值为int型，有高电平(HIGH)或低电平(LOW)两种；

6、analogRead (int pin) ;

该函数用来读取指定引脚的模拟量的转换数值。树莓派的引脚本身并不支持AD转换，所以需要外接AD转换模块才能使用该功能。

7、analogWrite (int pin, int value)

该函数将指定的数值写入到模拟量寄存器用于DA转换。需要说明的是，树莓派的引脚本身不支持DA转换，因此需要外接DA转换模块才能实现功能。

好了，关于wiringPi的核心函数就先介绍到这里。欢迎扫描下方二维码关注我们的微信公众号。



方正智芯
Founder Chip

长按扫码关注我们



方正智芯

公众号：founderchip

官方网站：www.founderchip.com

原创工业智能控制领域（PLC、单片机、通信）的技术分享