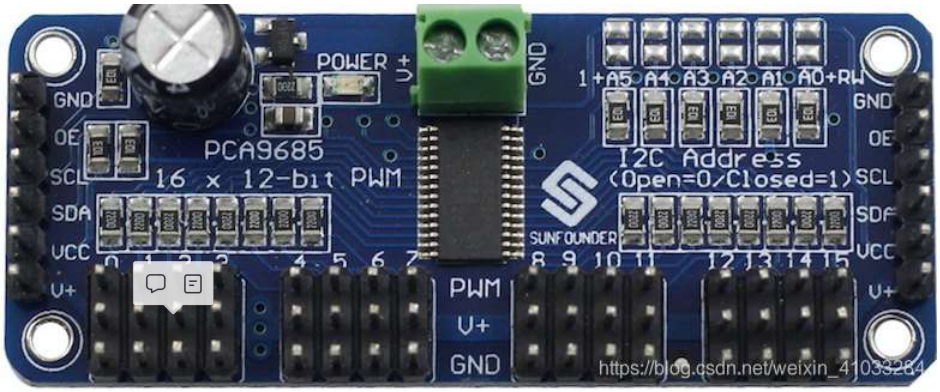
PCA9685 16路12位pwm信号发生器

1.16路12位PWM信号发生器，可用于控制舵机 、led、电机等设备，i2c通信，节省主机资源。



一、概述和硬件

1、概述

很常见的模块板子如上图所示，这个板子也比较便宜，十几块钱一个。i2c通信，只需要几根i2c线就可以控制16路pwm，周期和占空比都可控。可以多个模块级联。可控制16路通道的四种工作模式：关、开、pwm、可变pwm。精度是12位：

工作频率 时间分辨率 通常舵机500~2500us可分成份数 通常舵机500~2500us，旋转角180°的角度分辨率 对于电调，500us对应100%的占空比分辨率

工作频率	时间分辨率	通常舵机500~2500us可分成份数	通常舵机500~2500us，旋转角180°的角度分辨率
50Hz	4.88us	410份	0.439°
60Hz	4us	492份	0.366°

驱动方式可以选择开漏输出或推挽输出。

2、硬件

1、电压

数字电路电压范围可接受3.3和5v电平。

此外还有一个v+引脚，这个引脚是给舵机供电用的，可以接稍微高一点的电压。

2、i2c地址

有6个地址控制脚，通过这些引脚可以控制设备的i2c地址。

7位的I2C地址为：0x40 + A5:A0，A5到A0如果不做任何处理的话是0，想要把哪一位置1就把那个引脚焊到一起。

另外用i2cdetect检测出还有一个0x70地址一直存在，这是一个通用地址，可以给所有从机下达指令。

3、使能脚

模块有一个OE反使能脚，这个引脚低电平使能，不接的话模块内部默认已经接地使能了，所以正常使用可以不接。

二、寄存器功能

内部地址(hex)	名称	功能
00	MODE1	设置寄存器1
01	MODE2	设置寄存器2
02	SUBADR1	i2c-bus subaddress1
03	SUBADR2	i2c-bus subaddress2
04	SUBADR3	i2c-bus subaddress3
05	ALLCALLADR	
06	LED0_ON_L	
07	LED0_ON_H	
08	LED0_OFF_L	
09	LED0_OFF_H	
...	...	...
0x06 + 4*X	LEDX_ON_L	
0x06 + 4*X + 1	LEDX_ON_H	
0x06 + 4*X + 2	LEDX_OFF_L	
0x06 + 4*X + 3	LEDX_OFF_H	
...	...	... 上面共16路通道
FA	ALL_LED_ON_L	
FB	ALL_LED_ON_H	
FC	ALL_LED_OFF_L	
FD	ALL_LED_OFF_H	
FE	PRE_SCALE	控制周期的寄存器
FF	TestMode	

MODE1寄存器

位	名称	功能
D7	RESTART	写1复位，写完后此位自动清除。**一定要在SLEEP位写0后至少500us后才能对此位写1进行复位。**
D6	EXTCLOCK	0-使用内部时钟（25MHz）。1-使用外部时钟引脚的时钟。**修改此位前，一定要先SLEEP，再修改此位（此时SLEEP位仍然写1），再退出SLEEP。**
D5	AI	0-内部地址读写后不自动增加。1-内部地址读写后自动增加。一般i2c设备在对从机读写后内部地址都会自动增加，这个芯片可以手动设置是否自动增加，我们一般都会设成自动增加。
D4	SLEEP	0-退出SLEEP模式。1-进入SLEEP模式。注：1、写0退出sleep模式后，最多等500us后即可产生稳定的时钟信号。2、写1进入sleep模式后，时钟会关闭。此时可以修改时钟源寄存器EXTCLOCK和周期寄存器PRE_SCALE，修改这两个寄存器之前必须先进入sleep模式。
D3	SUB1	
D2	SUB2	
D1	SUB3	
D0	ALLCALL	0-不响应0x70通用i2c地址。1-响应0x70通用i2c地址。这个芯片除了可以通过A5:A0自定义i2c地址外，还有一个通用i2c地址0x70，此寄存器可以控制是否响应这个通用地址。注意啊：这个寄存器的设置好像掉电会保存的！

各个通道的ON和OFF寄存器
总共16个通道，每个通道都有 LEDX_ON_L、LEDX_ON_H、LEDX_OFF_L、LEDX_OFF_H 四个寄存器。
系统中有一个12位的计数ACK，ACK根据PRE_SCALE寄存器设置的周期进行增加，没增加一次就会和上述四个寄存器对比：
当发现 ACK == LEDX_ON_H[3:0]:LEDX_ON_L 时，X通道输出高电平；
当发现 ACK == LEDX_OFF_H[3:0]:LEDX_OFF_L 时，X通道输出低电平。

PRE_SCALE寄存器
这个寄存器是用来设置周期的，具体原理可以不用管，只要记住这个公式：

$$prescale\ value = round\left(\frac{osc\_clock}{4096 \times update\_rate}\right) - 1$$

其中osc_clock是时钟，根据上面的寄存器设置

其中osc_clock是时钟，根据上面的寄存器设置选择是内部25MHz时钟还是外部时钟；
update_rate是频率，比如周期是20ms，那么频率就是50。
注意：实际应用中发现有误差，需要加入校准，要把udpate_rate乘以0.915。
包括从网上下载的arduino驱动中也加入了此校准。