



# 目录 Contents

---

- 第一部分  PWM简介
- 第二部分  PWM工作原理
- 第三部分  PWM驱动LED灯
- 第四部分  呼吸灯

# PWM简介

- PWM全称Pulse Width Modulation，指脉冲宽度调制

PWM是利用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术，广泛应用在从测量、通信到功率控制与变换的许多领域中。

- PWM产生和传输数字式信号

PWM只对一个方波序列信号的占空比按要求进行调制，而不改变信号的幅度和周期，因此PWM信号的产生和传输都是数字式的。

- PWM实现模拟信号

若调制信号的频率远远大于信号接受者的分辨率，则接收者并不能感知数字信号的0和1，它获得的是信号的平均效果，平均值与占空比成有关(正比关系)。

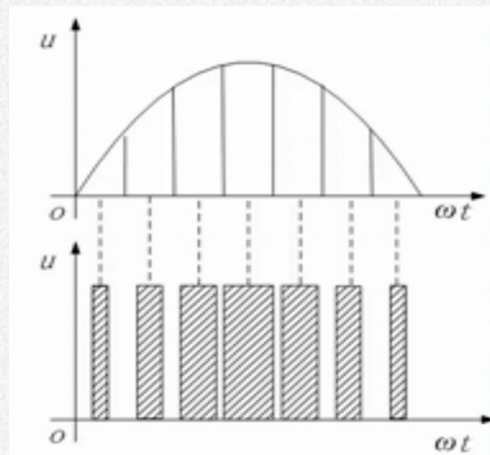


图1 PWM等效正弦波



# PWM工作原理

- 基本原理

PWM的控制方式是对逆变电路开关器件的通断进行控制，使输出端得到一系列幅值相等的脉冲，用这些脉冲来代替正弦波或所需要的波形。也就是在输出波形的半个周期中产生多个脉冲，使各脉冲的等值电压为正弦波形，所获得的输出平滑且低次谐波少。按一定的规则对各脉冲的宽度进行调制，即可改变逆变电路输出电压的大小，也可改变输出频率。

- 实现方法

PWM是一种对模拟信号电平进行数字编码的方法。通过高分辨率计数器的使用，方波的占空比被调制用来对一个具体模拟信号的电平进行编码。PWM信号仍然是数字的，因为在给定的任何时刻，满幅值的直流供电要么完全有(ON)，要么完全无(OFF)。电压或电流源是以一种通(ON)或断(OFF)的重复脉冲序列被加到模拟负载上去的。通的时候即是直流供电被加到负载上的时候，断的时候即是供电被断开的时候。只要带宽足够，任何模拟值都可以使用PWM进行编码。

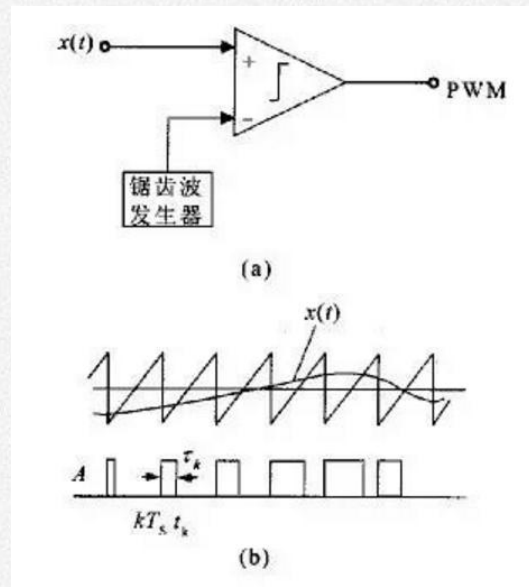


图2 脉冲宽度调制过程  
(a) 调制原理图 (b) 调制后的波形图

# PWM工作原理

- 占空比与调制频率

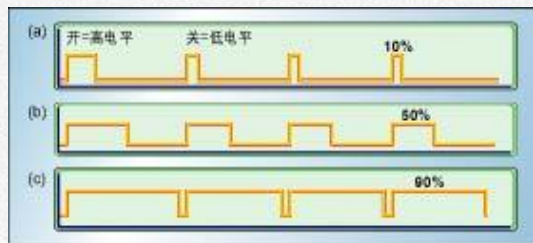


图 3

图3显示了三种不同的PWM信号。图3 (a) 是一个占空比为10%的PWM输出，即在信号周期中，10%的时间通，其余90%的时间断。图3 (b) 和图3 (c) 显示的分别是占空比为50%和90%的PWM输出。这三种PWM输出编码的分别是强度为满度值的10%、50%和90%的三种不同模拟信号值。例如，假设供电电源为9V，占空比为10%，则对应的是一个幅度为0.9V的模拟信号。



图 4

图4是一个可以使用PWM进行驱动的简单电路。图中使用9V电池来给一个白炽灯泡供电。如果将连接电池和灯泡的开关闭合50ms，灯泡在这段时间中将得到9V供电。如果在下一个50ms中将开关断开，灯泡得到的供电将为0V。如果在1秒钟内将此过程重复10次，灯泡将会点亮并象连接到了一个4.5V电池(9V的50%)上一样。这种情况下，占空比为50%，调制频率为10Hz。



# PWM驱动LED

- 定时器

一般开发板上的多个定时器都有PWM功能，在这里我们也是使用定时器上的PWM功能来进行测试和讲解。

之前在第二章我们已经讲了RTC和定时器，了解了定时器的使用。因此在这里我们简单地讲一下在本章需要用到的方法。



- 定义定时器Timer  
`tm=Timer(n) #n=1-14`
- 更多定义方式:  
`tm=Timer(n, freq)`  
`tm=Timer(n, freq, callback)`
- 设置频率  
`tm.freq(100)`
- 定义回调函数 (中断)  
`tm.callback(f)`
- `timer.channel(channel, mode, ...)`, 设置定时器通道
  - `channel`, 定时器通道号
  - `mode`, 模式
    - `Timer.PWM`, PWM模式 (高电平方式)
    - `Timer.PWM_INVERTED`, PWM模式 (反相方式)
    - .....
  - `callback`, 每个通道的回调函数
  - `pin`, 驱动GPIO, 可以是None
- 在 `Timer.PWM` 模式下的参数
  - `pulse_width`, 脉冲宽度
  - `pulse_width_percent`, 百分比计算的占空比

# PWM驱动LED

- LED

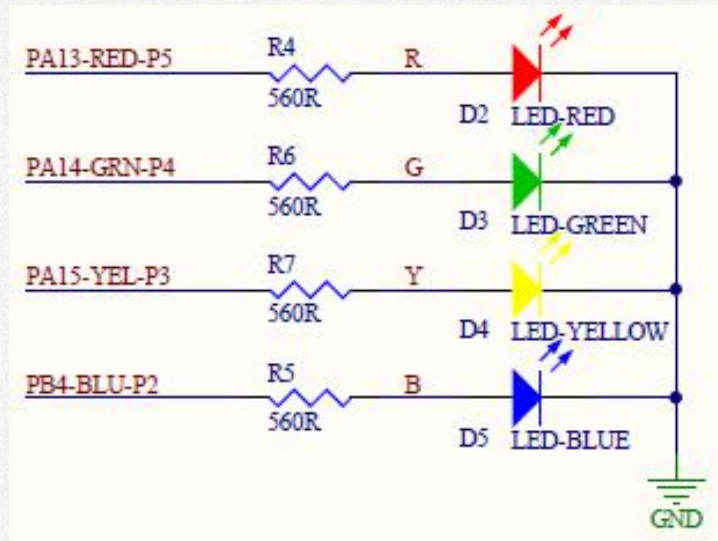


图5 开发板上LED电路原理图

- 如图5的LED电路图所示，开发板上有四个LED灯，颜色分别为红、绿、黄、蓝，对应的引脚依次为PA13、PA14、PA15、PB4。
- 在本次演示实验中，我们使用黄灯和蓝灯，引脚为PA15，PB4，然后我们在STM32F405RGT6芯片电路图中查找这两个引脚。



# PWM驱动LED

- LED

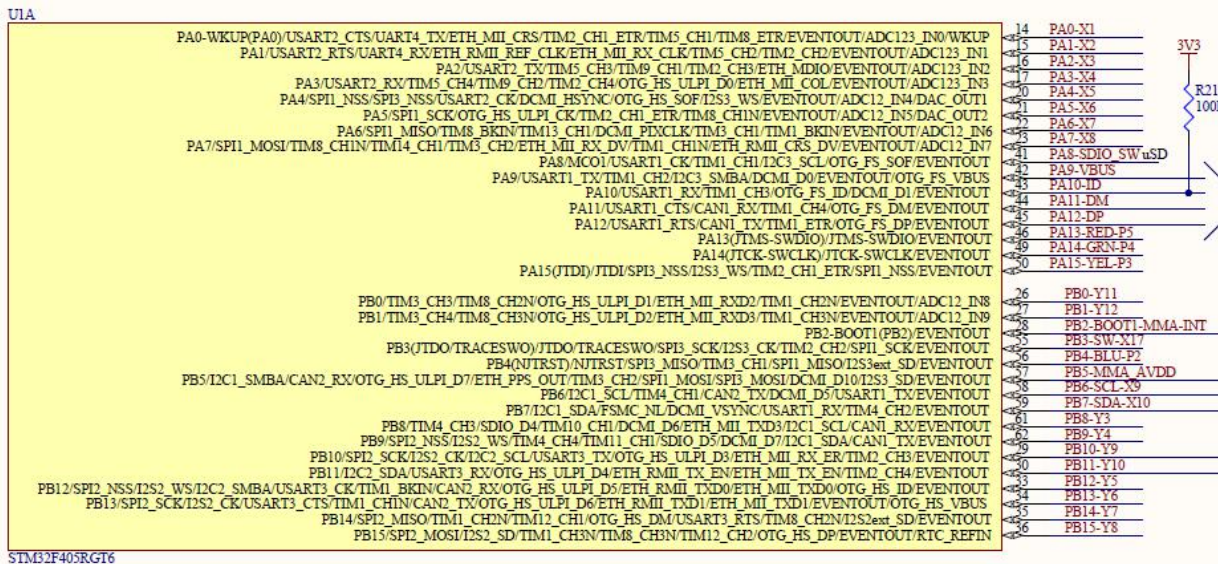


图6 STM32F405RGT6电路原理图

- 黄灯的引脚PA15对应有定时器2、通道1 (TIM2\_CH1\_ETR)。
- 蓝灯的引脚PB4对应有定时器3、通道1 (TIM3\_CH1\_ETR)。



# ■ PWM驱动LED

- 代码实现

根据前面分析，我们用定时器2（TIM2\_CH1\_ETR）的PWM驱动黄灯，用定时器3（TIM3\_CH1\_ETR）的PWM驱动蓝灯。代码实现如下：

```
# main.py -- put your code here!
```

```
# 导入Timer和Pin两个库
```

```
from pyb import Pin, Timer
```

```
# 定义计时器2和3，并指定频率
```

```
tm2=Timer(2, freq=5)
```

```
tm3=Timer(3, freq=50)
```

```
# 设置定时器通道，PWM工作模式，引脚号，占空比
```

```
led3=tm2.channel(1, Timer.PWM, pin=Pin.cpu.A15, pulse_width_percent=30)
```

```
led4=tm3.channel(1, Timer.PWM, pin=Pin.cpu.B4, pulse_width_percent=50)
```

## PWM驱动LED

- 效果展示

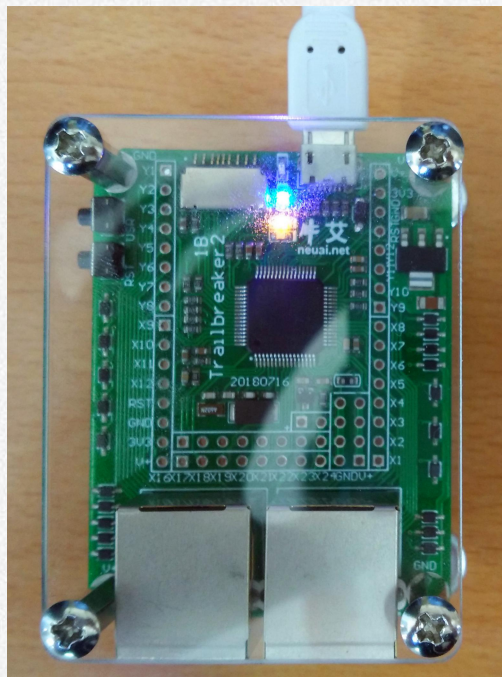


图7 开发板LED实物图

- 在开发板上运行代码后，会发现黄灯以5Hz的频率在闪烁，而蓝灯的闪烁频率很快，以至于肉眼观察到的是常亮状态。
- 可以修改定时器频率以自定义LED的闪烁频率，如：  
`tm2=Timer(2, freq=10)`
- 可以修改PWM占空比以自定义LED点亮的持续时间，如：  
`pulse_width_percent=10`
- 本次演示实验中只用到了黄LED和蓝LED，课后可以自己尝试用PWM驱动红LED和绿LED。



## ■ 呼吸灯代码实现:

- `from pyb import Pin,Timer`
- `tm2=Timer(2,freq=100)`#选择计时器2， 设置频率为100Hz
- `intensity3=0`#设置初始亮度（占空比）为0
- `led3=tm2.channel(1,Timer.PWM,pin=Pin.cpu.A15)`#指定通道1及引脚， 设置PWM模式
- `while True:#循环`
  - `if intensity3==99:#如果占空比达到最大（灯最亮）， 减小占空比使灯变暗`
    - `while intensity3>0:`
      - `led3.pulse_width_percent(intensity3)`
      - `intensity3=(intensity3-1)%100`
      - `pyb.delay(50)`
    - `intensity3=(intensity3+1)%100`#增加占空比， 使灯变亮
    - `led3.pulse_width_percent(intensity3)`
    - `pyb.delay(50)`#延时





# THANK YOU

Concise business business report PPT

牛艾科技