第 12 课，学会使用 PWM 进行 LED 调节亮度和控制无源蜂鸣器

本课学习内容：

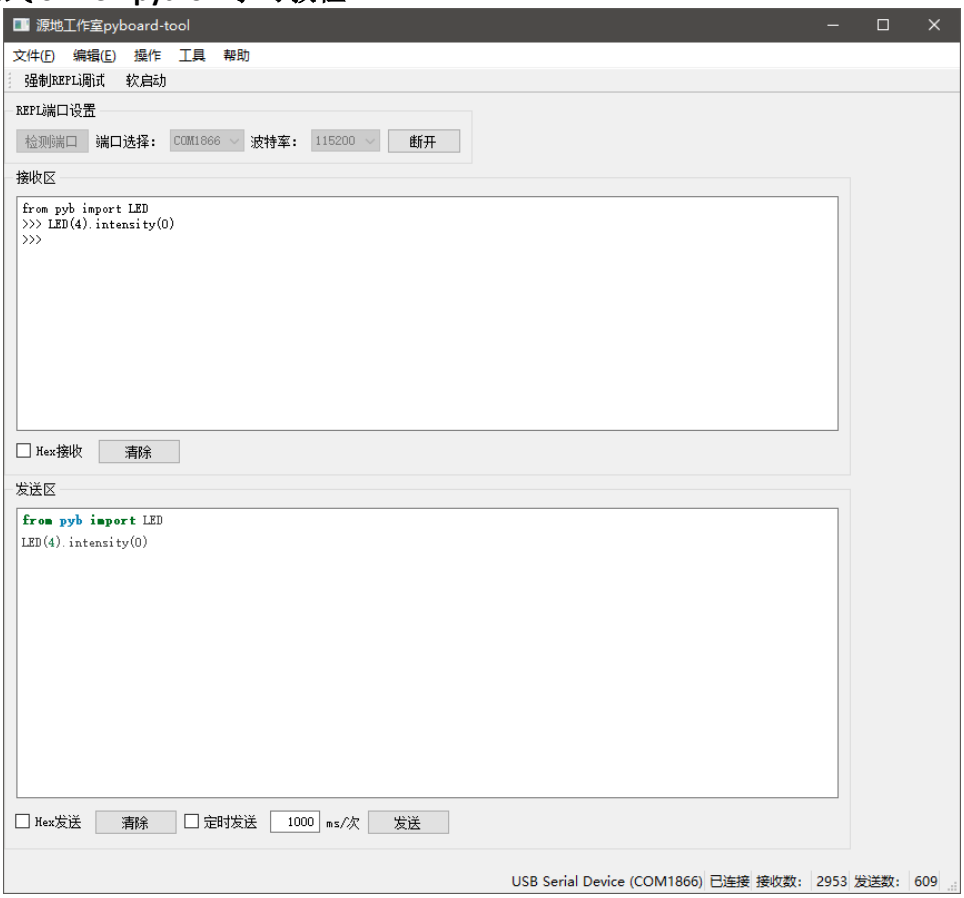
- 理解 PWM 为何物
- 学会使用 PWM 进行调节 LED 的亮度
- 学会询问当前产生的方法
- 学会使用 PWM 控制无源蜂鸣器的音调

LED 一般是恒流操作的，如何改变 LED 的亮度呢？答案就是 PWM 控制。在一定的频率的方波中，调整高电平和低电平的占空比，即可实现。比如我们用低电平点亮一个 LED 灯，我们假设把一个频率周期分为 10 个时间等份，如果方波中的高低电平占空比是 9：1，这是就是一个比较暗的亮度，如果方波中高低电平占空比是 10：0，这时，全部是高电平，灯是灭的。如果占空比是 5： 5，就是一个中间亮度，如果高低比是 1：9，是一个比较亮的亮度，如果高低是 0：10，这时全部是低电平，就是最亮的。咱们的 pyboard 板子如果实现 LED 亮度调节其实没有这么麻烦，在几乎所有的板子上都有 4 个 LED 用于指示控制 LED 灯，第 3 个和第 4 个 LED 灯可以直接控制其亮度，虽然有直接控制语句，但是内部的机理是一样的，通过改变占空比的大小来控制 LED 亮度。

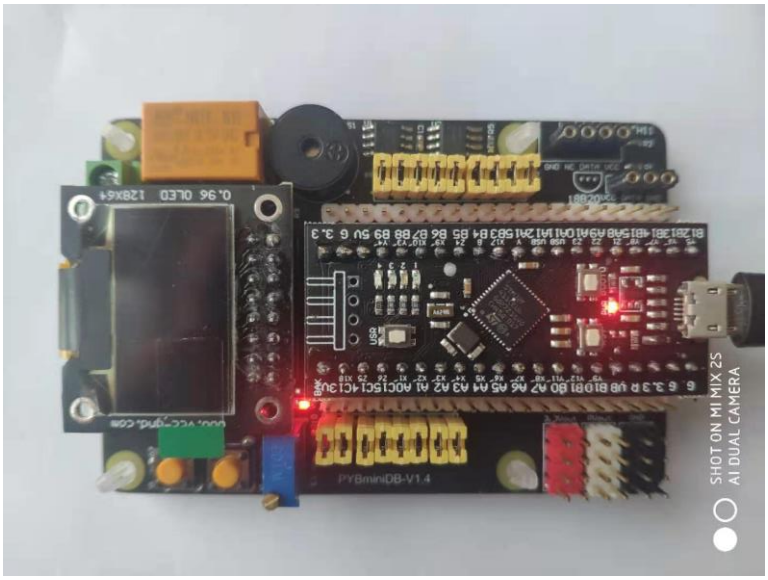
控制这两个 LED 灯的亮度编程语句为：

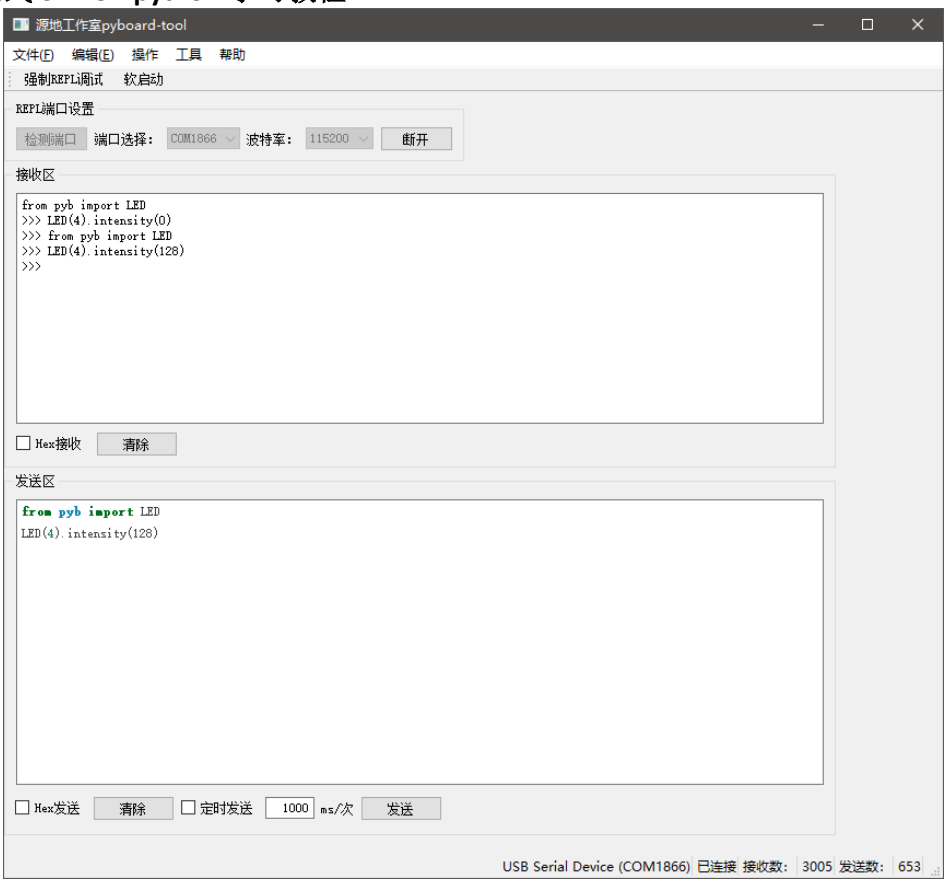
```
from pyb import LED

# LEDs 3 and 4 support PWM intensity (0-255)

LED(4).intensity()     # 获取当前的 PWM 数值

LED(4).intensity(128) #设定当前的 PWM 数值,这里 128 正好是最亮与最不亮的中间中间状态。
```

上述语言由于比较简单，使用 Pyboard-tool 工具进行语句的直接验证就可以。

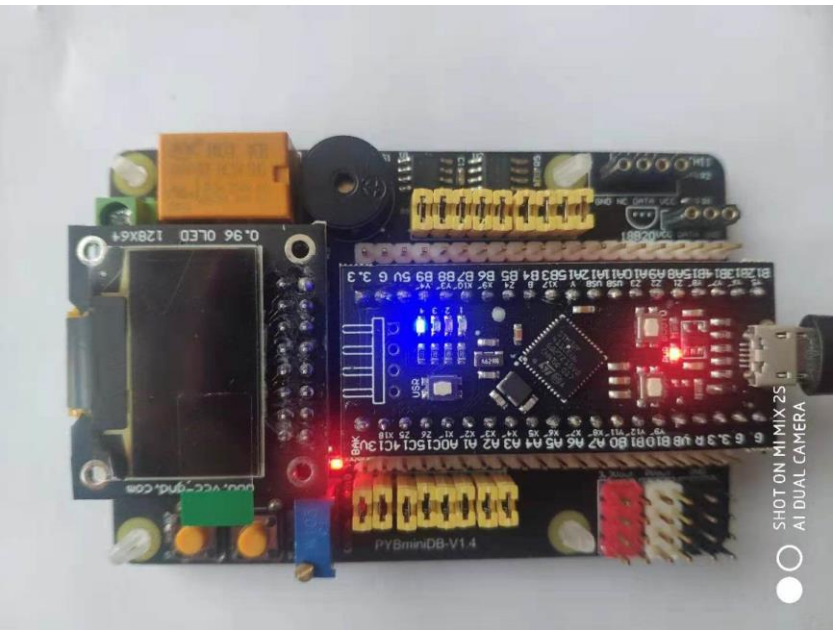


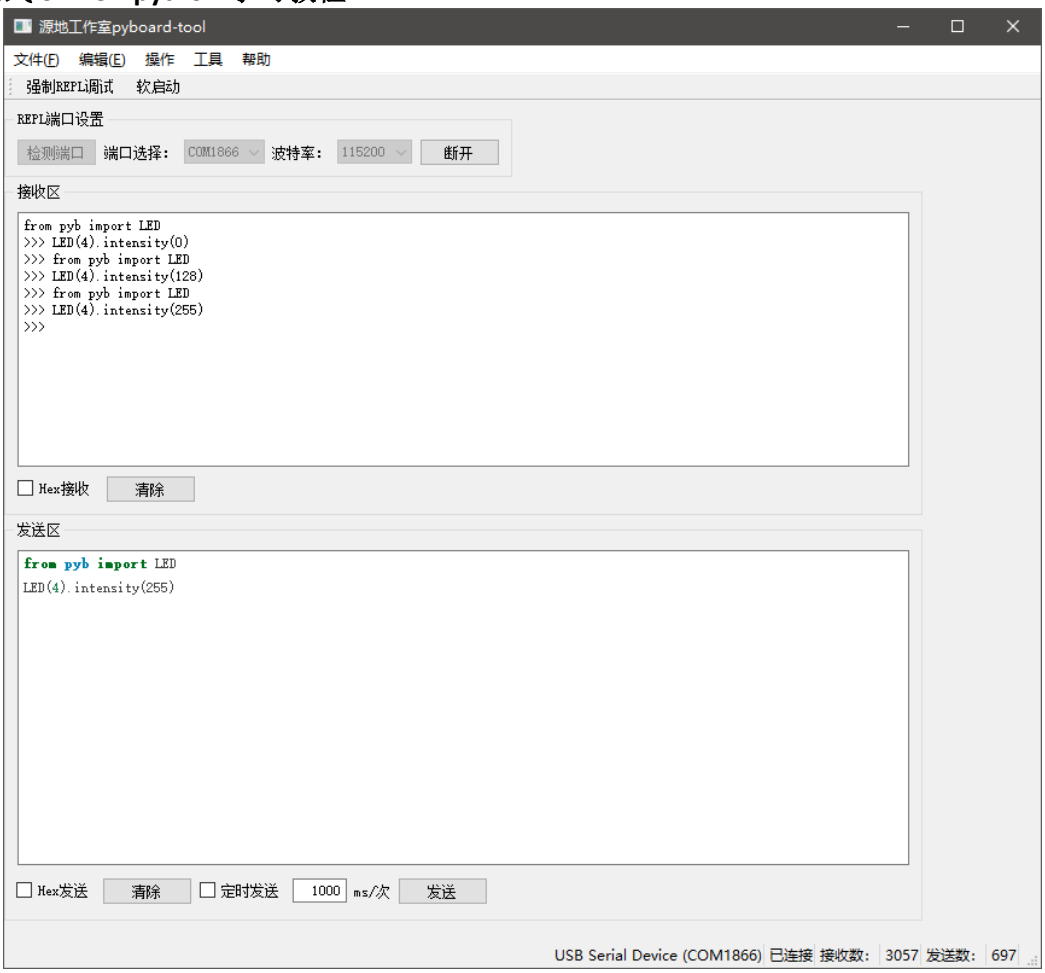
下图是不亮的状态（熄灭），即 PWM 占空比为 0。



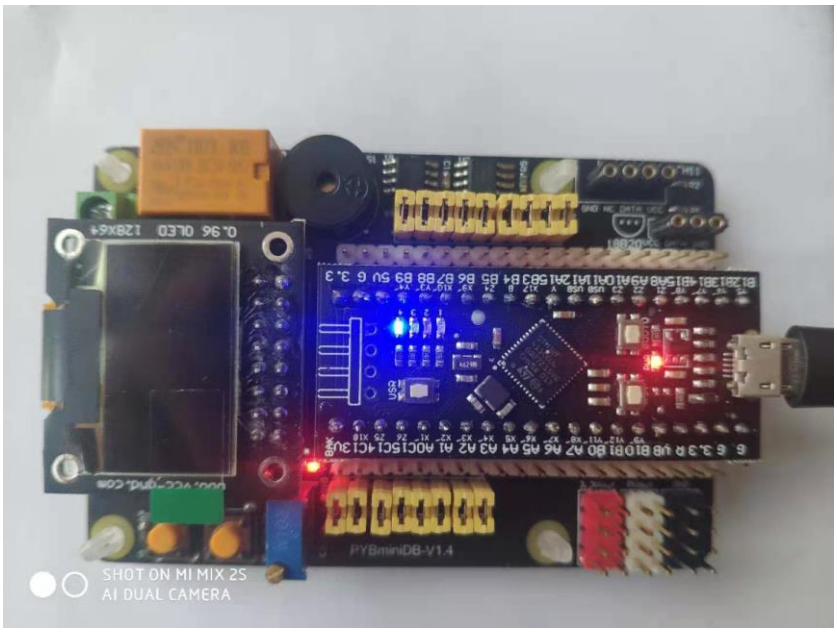


下图为中间亮的状态，即 PWM 占空比为 50%。



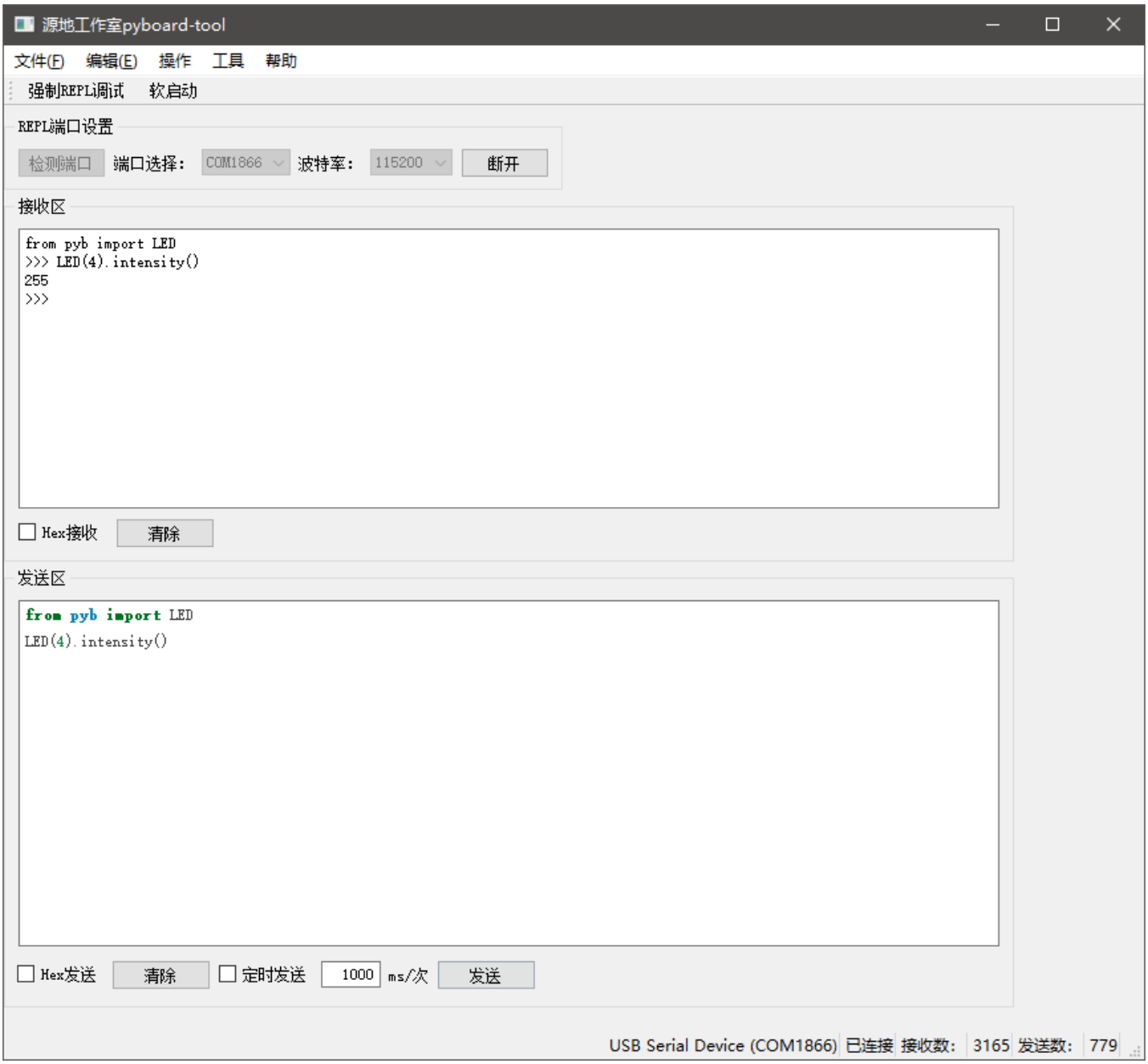


下图为最亮的状态，即 PWM 占空比为 100%。实际观看还是有却别的，只是图片对比没有这么明显。



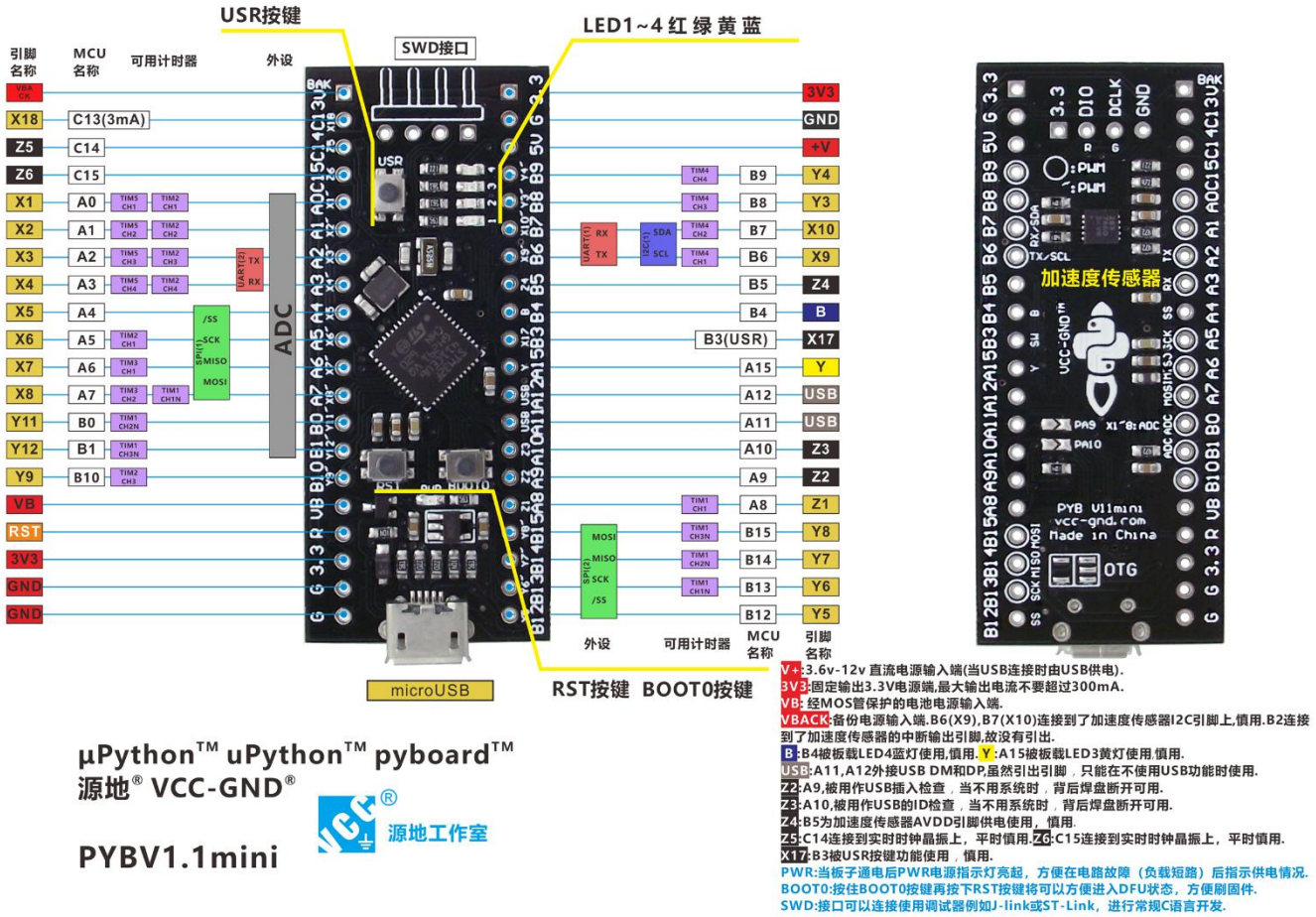
下面我们学一个重要的知识点，询问当前的参数，假设你想获取当前的 pwm 数值，就是用设置语句，但是不要加参数，这是系统就会将设置语句理解成询问语句，给你答案。

例如：



会给你当前的参数，当前为 255 数值。

上述语句只能控制这两个特定的 LED 灯，不能使用上述语句控制其他的引脚或其他的外设。我们怎么使用 PWM 控制呢？这时需要看 Pyboard 的引脚指示图了，带有定时器的引脚都是可以输出 PWM，以下就是引脚功能指示图。



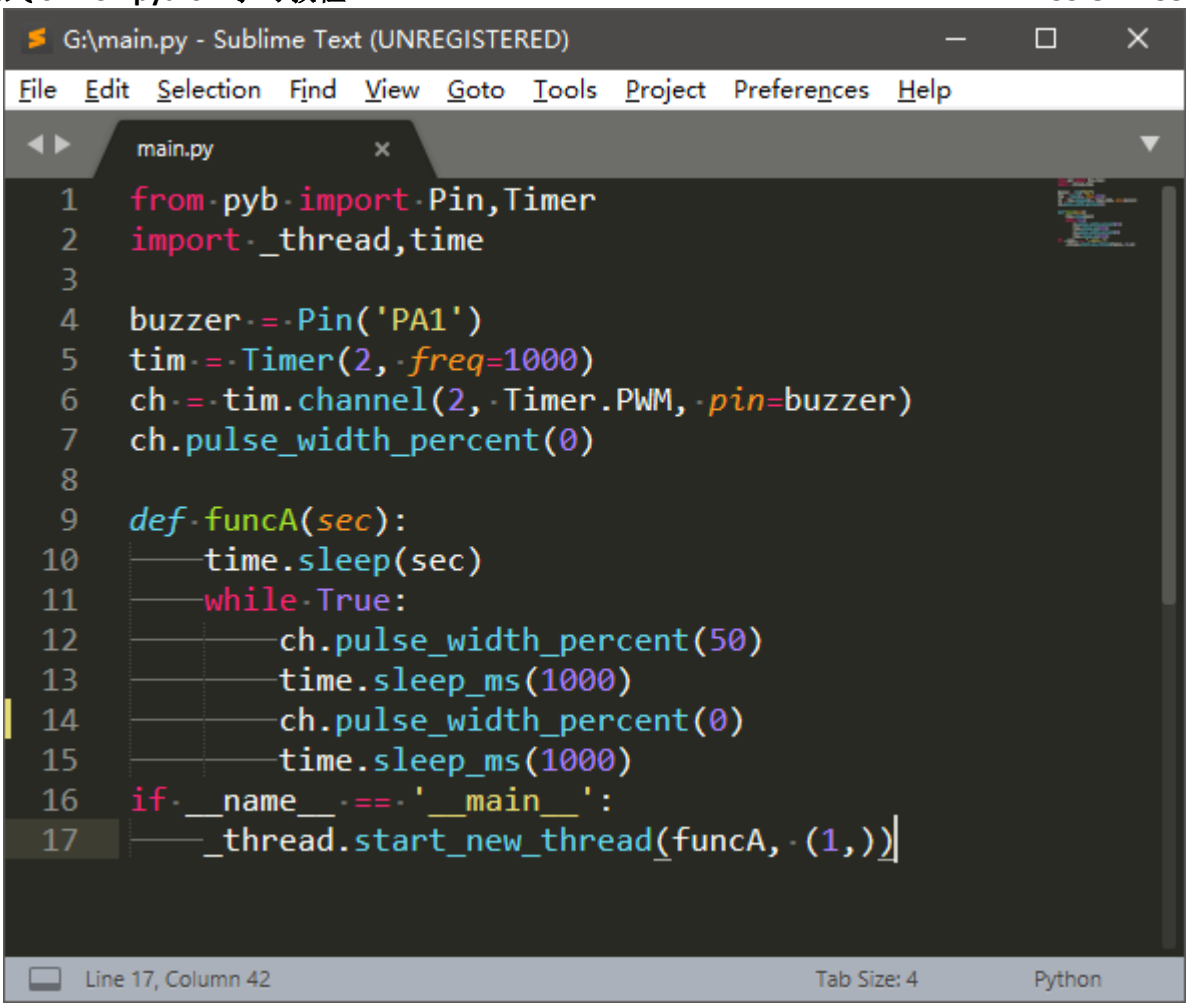
上图中引脚只有是带有 TIMxCHx 就可以使用 PWM 功能，没有带这个标识的是不能使用的，例如 PA4(X5)引脚,就不能使用。

以下就是具体让 A0 也就是 X1 引脚输出频率为 1kHz，占空比为 50%的 PWM.

```
from pyb import Pin, Timer  #引用引脚和定时器库
p = Pin('X1') # X1 has TIM2, CH1，该信息是从引脚功能图中找到的
tim = Timer(2, freq=1000)  #2 是代表是定时器 2，freq=1000 是设置为 1kHz
ch = tim.channel(1, Timer.PWM, pin=p)#1 代表是通道 1，模式为 PWM，pin=p 引脚上述设置的语句
ch.pulse_width_percent(50)#占空比为 50%
```

上述语言可以使用，Pyboard-tool 调试工具进行验证，如果有条件的可以使用示波器进行观察 PWM 波形。

如果没有示波器，可以用其他的外设表现出来，这里我们的 PYBminiDB 有一个无源蜂鸣器，通过无源蜂鸣器的响声来表现。



from pyb import Pin,Timer#引用相应的库

import _thread,time#引用线程和 time 函数

buzzer = Pin('PA1')#新建蜂鸣器引脚相关参数

tim = Timer(2, freq=1000)#需要看引脚功能指示图定时器 2，设定频率是 1000kHz

ch = tim.channel(2, Timer.PWM, pin=buzzer)#通道 2，PWM 模式，引脚为上述

ch.pulse_width_percent(0)#初始化占空比为 0，就是不响

def funcA(sec):#线程 A 函数

time.sleep(sec)#演示 sec 数值

while True:#线程中的循环

ch.pulse_width_percent(50)#设置占空比 50%，蜂鸣器响

```
        time.sleep_ms(1000)#延时 1s

        ch.pulse_width_percent(0)#设置占空比 0%，蜂鸣器不响

        time.sleep_ms(1000)#延时 1s

if __name__ == '__main__':

  _thread.start_new_thread(funcA, (1,))#开启线程
```

将上述脚本，保存好（再次提醒，等待 LED1 熄灭后，才算保存完毕，才能进行下一步操作，切勿心急），然后连接 Pyboard-tool 工具，点击“软启动”按钮，观察程序运行。这时蜂鸣器会 1s 响一次。