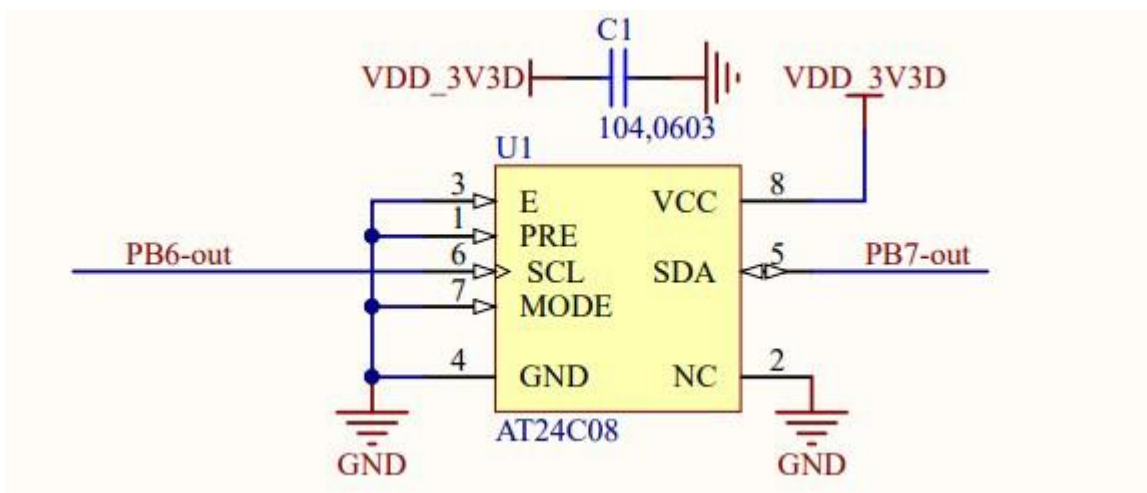


本课学习内容：

- 了解 I2C 是一种芯片级别直接的通讯
- 实现 24C08 实现读写操作
- 利用 24C08 器件，实现设备开机次数记录

24c08 是一个非遗失 eeprom 存储器器件，采用的 IIC 总线技术。24c08 在许多试验中都有出现。24c02 的应用，主要在存储一些掉电后还要保存数据的场合，在上次运行时，保存的数据，在下次运行时还能够调出。24c02 采用的 IIC 总线，是一种 2 线总线，在 Pyboard 中可以使用硬件 I2C 方式也可以使用软件 I2C，至于 I2C 总线的时序和原理，请参考其他相关资料。如果您不想研究，也没有关系，Python 精髓就是在有时候不太明白时候不耽误使用，现在和今后您都可以只调用就是，不必花时间和精力去研究。

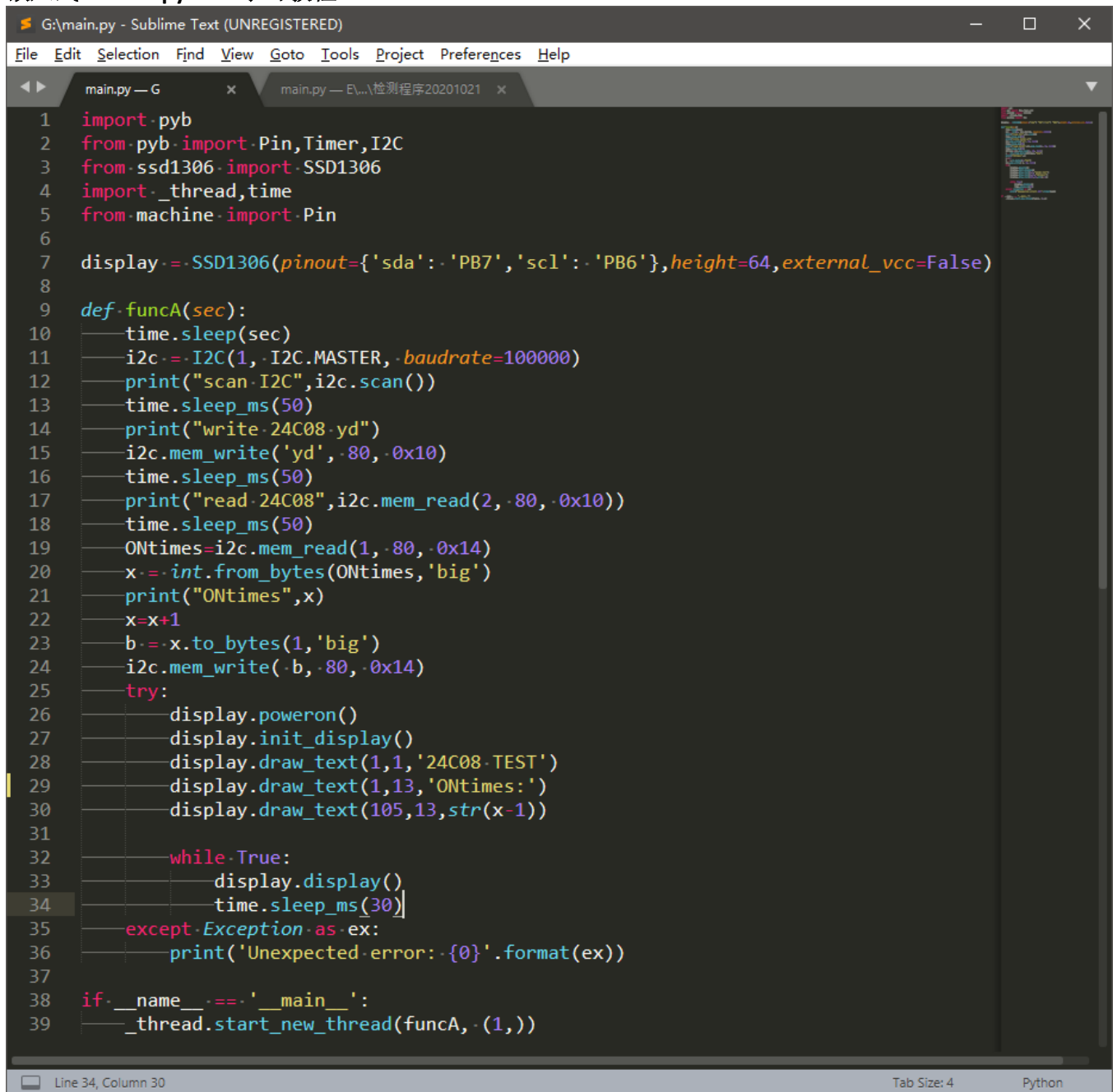
首先看一下原理图，24C08 器件硬件连接。PB6 连接 SCL，PB7 连接 SDA，这两个通讯线都实际进行了上拉电阻，大部分 I2C 都需要进行上拉处理。



这次来个直接的，我直接上程序，该程序是为实现记录开机次数，OLED 和 REPL 都可以显示。我们在宏观层面分析一下，我们用短短的 39 行程序，我们实现了以下功能：

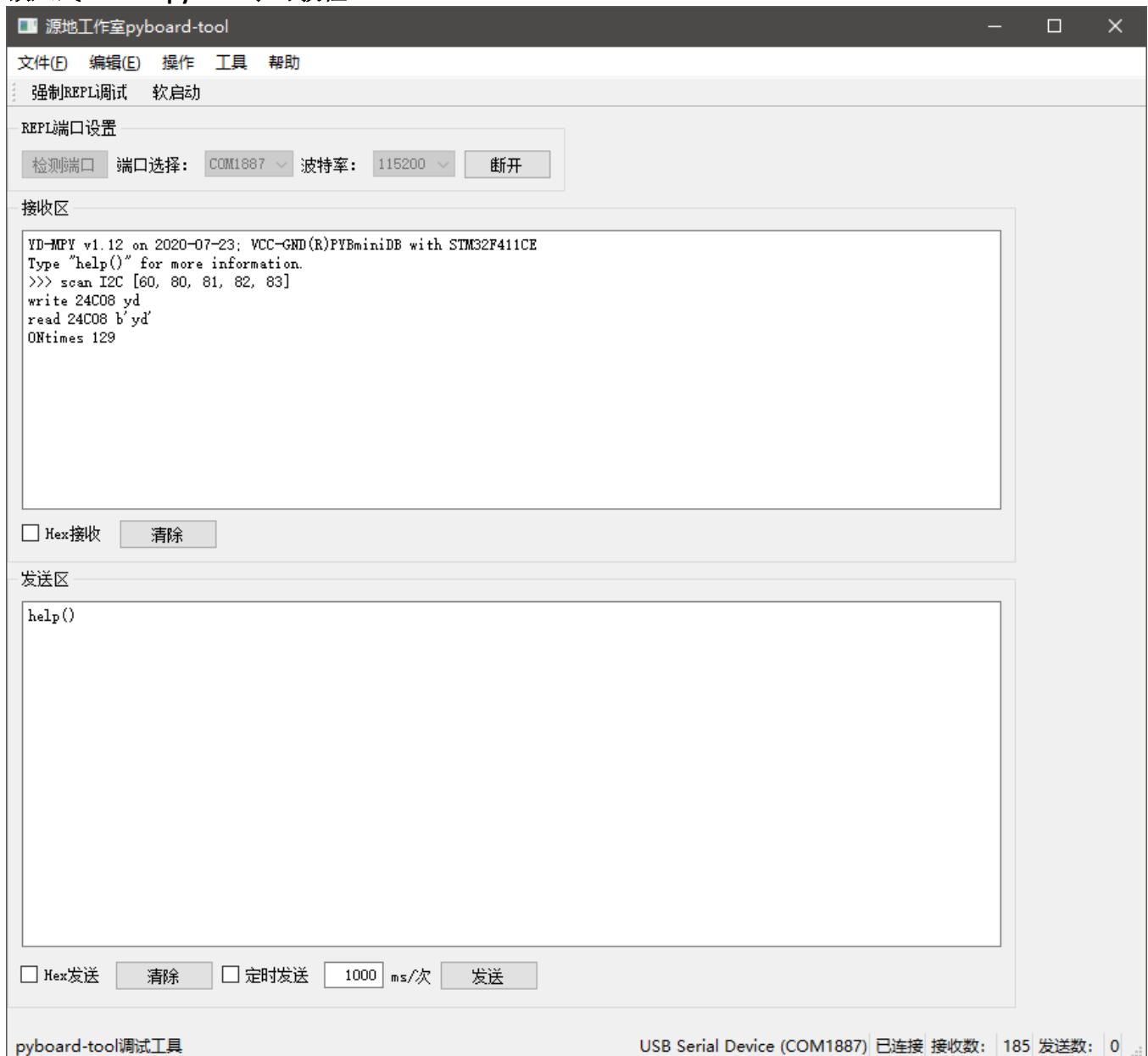
- REPL 信息的打印，例如 I2C 扫描信息，I2C 操作信息。
- 完成了对 24C08 读写。
- 完成了 OLED 显示任务。
- 完成了开机次数的提取与加一次的处理，并在此储存。

如果使用 C 语言进行编写完成上述任务，我们可能花费时间要远远超过我们编写这 39 行代码的时间，工作量也是远远超过这 39 行的代码量，当然这里我们走了一下捷径，没有考虑 SSD1306 脚本和其他的脚本引入，其实也不用在意这些脚本细节，只要是知道他们什么作用就行。

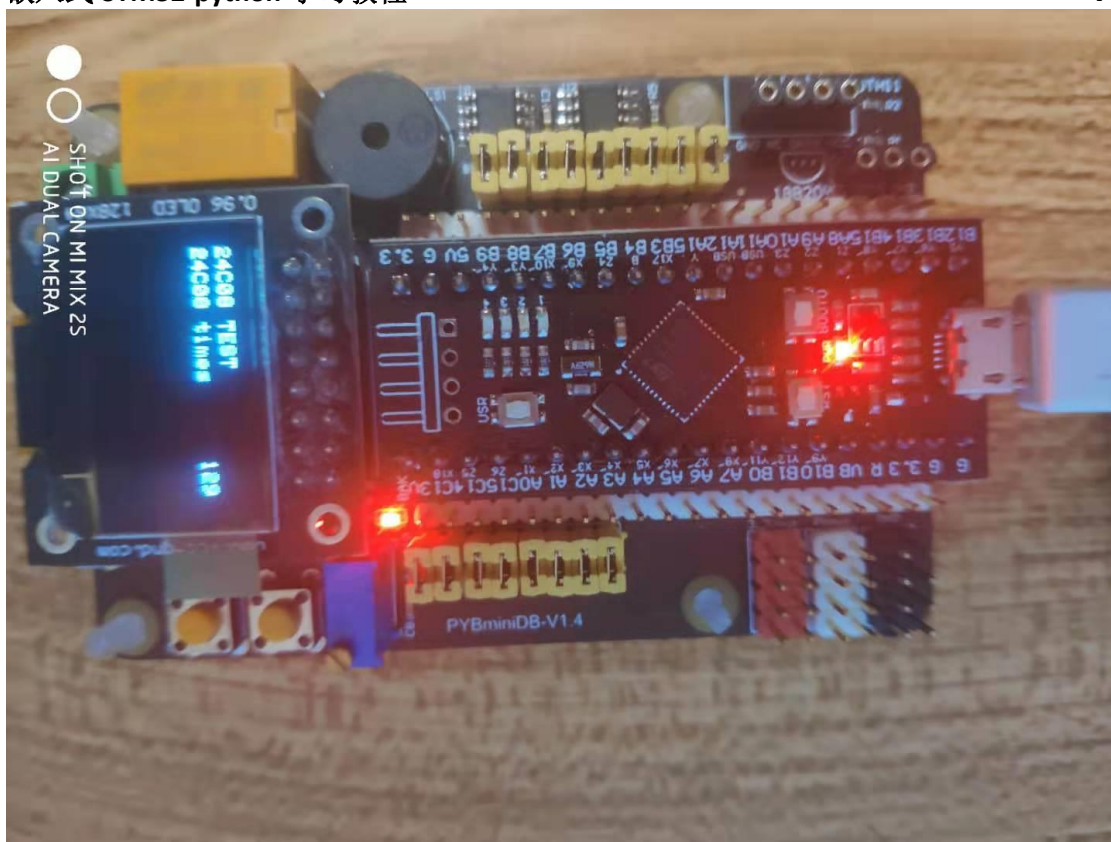


```
1 import pyb
2 from pyb import Pin, Timer, I2C
3 from ssd1306 import SSD1306
4 import _thread, time
5 from machine import Pin
6
7 display = SSD1306(pinout={'sda': 'PB7', 'scl': 'PB6'}, height=64, external_vcc=False)
8
9 def funcA(sec):
10     time.sleep(sec)
11     i2c = I2C(1, I2C.MASTER, baudrate=100000)
12     print("scan I2C", i2c.scan())
13     time.sleep_ms(50)
14     print("write 24C08 yd")
15     i2c.mem_write('yd', -80, -0x10)
16     time.sleep_ms(50)
17     print("read 24C08", i2c.mem_read(2, -80, -0x10))
18     time.sleep_ms(50)
19     ONtimes = i2c.mem_read(1, -80, -0x14)
20     x = int.from_bytes(ONtimes, 'big')
21     print("ONtimes", x)
22     x = x + 1
23     b = x.to_bytes(1, 'big')
24     i2c.mem_write(b, -80, -0x14)
25     try:
26         display.poweron()
27         display.init_display()
28         display.draw_text(1, 1, '24C08-TEST')
29         display.draw_text(1, 13, 'ONtimes:')
30         display.draw_text(105, 13, str(x-1))
31
32     while True:
33         display.display()
34         time.sleep_ms(30)
35     except Exception as ex:
36         print('Unexpected error: {}'.format(ex))
37
38 if __name__ == '__main__':
39     _thread.start_new_thread(funcA, (1,))
```

保存好全部的脚本程序后，还是不厌其烦的说一下，点击保存后，等 LED1 熄灭后才算保存完毕，然后通过 Pyboard-tool 软件上按一下“软启动”，在 REPL 会打印相应的信息，OLED 会显示显示对应的信息。



同时在 OLED 中也可以显示开机次数。显示中的 129 证明已经开机或者复位次数已经是 129 了。



作业：利用上节课中的中断按键功能，在本节课的基础之上，编写一个开机次数清零的程序。按下按键后，开机次数清零。再上电开机次数 1……