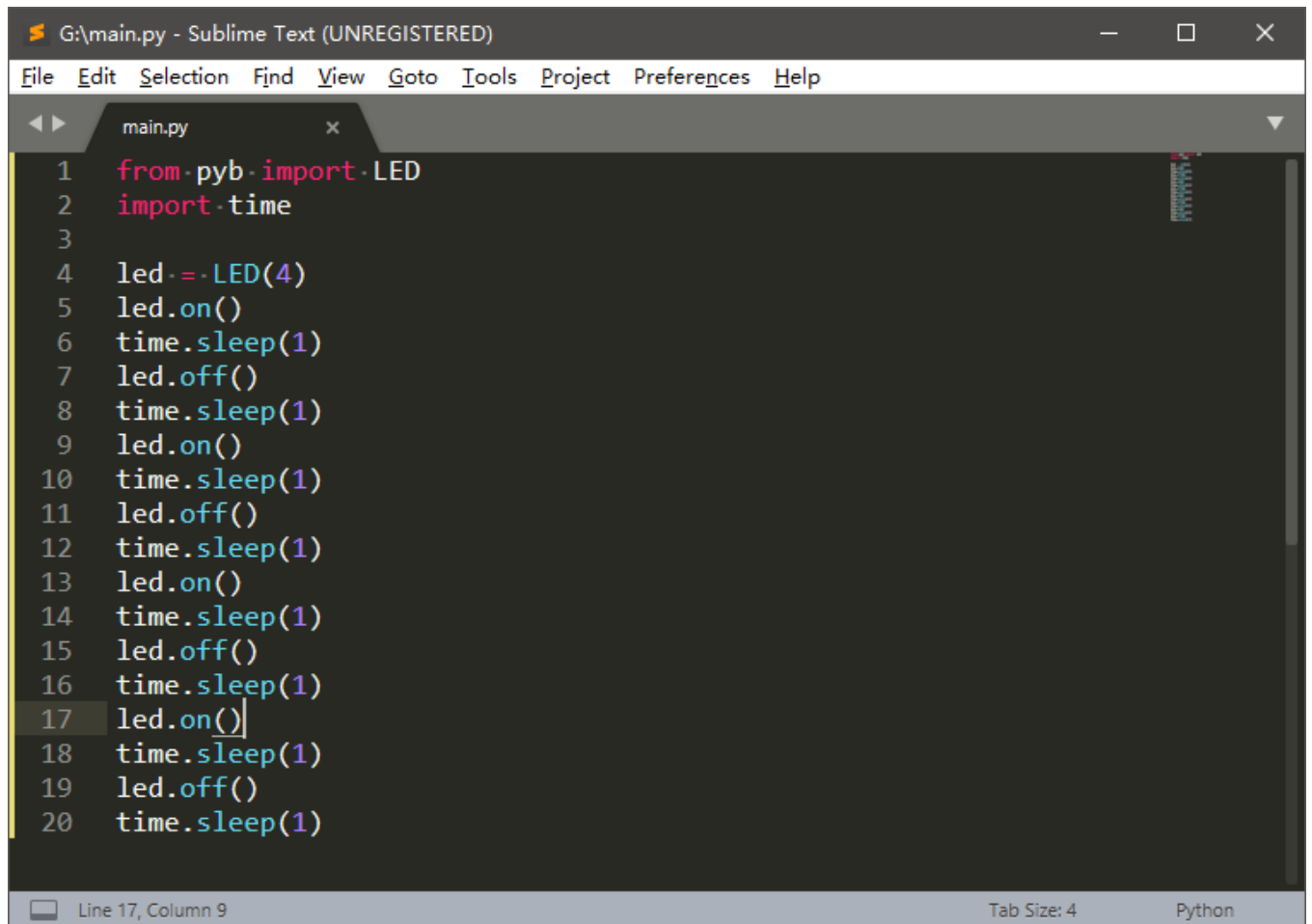


本课学习内容：

- 掌握 LED 的基本操作
- 熟悉 Python 的 import XXX 引用和 from pyb import XXX 引用
- 熟悉延时语句 time.sleep() 和 pyb.delay()
- 掌握 Pyboard-tool 软件的使用，基本操作与 REPL 交互调试
- 掌握 for in 循环
- 通过 while (1) :来实现循环闪烁 LED(4),但是注意这种方法对于 Python 来说很 Low，原因是一旦用了该结构容易，独占系统资源，就连 REPL 交互调试都不能使用了。
- Python 语言可以面向过程，也可以面向对象编程。C 语言只能面向过程编程。

首先我们先不用循环语句，只是机械地重复让 LED4 闪烁 4 次，程序要表达目的应该没有什么难理解的。

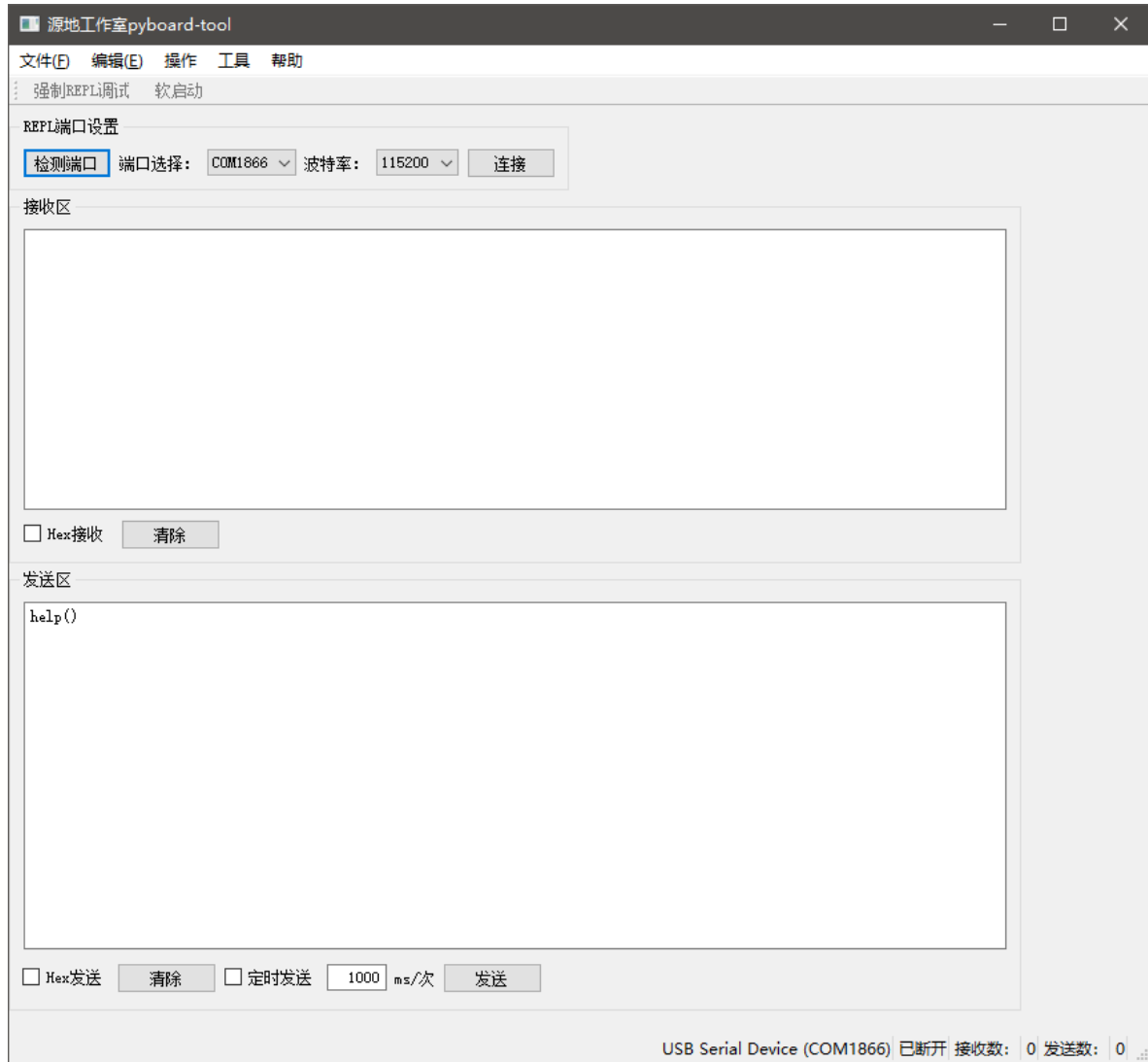


```
G:\main.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
main.py
1 from pyb import LED
2 import time
3
4 led = LED(4)
5 led.on()
6 time.sleep(1)
7 led.off()
8 time.sleep(1)
9 led.on()
10 time.sleep(1)
11 led.off()
12 time.sleep(1)
13 led.on()
14 time.sleep(1)
15 led.off()
16 time.sleep(1)
17 led.on()
18 time.sleep(1)
19 led.off()
20 time.sleep(1)
Line 17, Column 9 Tab Size: 4 Python
```

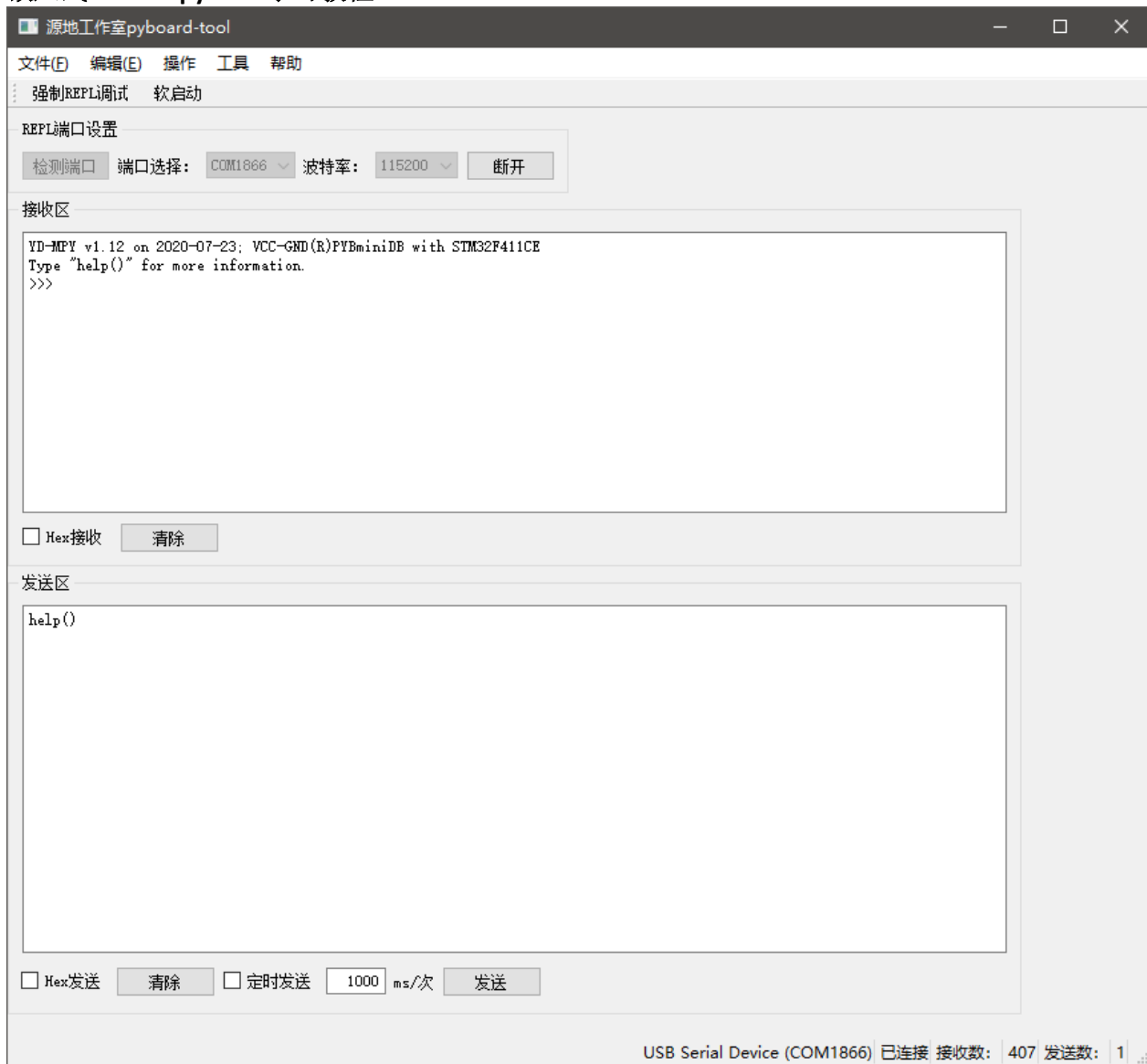
首先像上一课，打开生成的可移动硬盘中 main.py 脚本文件，这次我们就不用记事本软件了，而是用上一课安装的 Sublime Text 编程脚本软件来编写程序。如图键入代码脚本后，在 File 中点击 Save 或直接 Ctrl+S 保存到生成的可移动硬盘脚本文件中，还是重申注意操作要点，点击保存的一瞬间，第一个 LED 会

亮起，大约等待 3~4 秒会熄灭，才算完成，在 LED1 亮的过程中，千万别进行其他的动作，防止保存用户脚本出错。

再次，可以像上一课一样，按一下复位按键或者重新上电来启动程序。但是我们不推荐这两种方式，我们强烈建议使用我们提供的 Pyboard-tool 上的“软启动”按键进行程序的复位，具体操作如下：



首先点击检测端口，找到对应的 Pyboard 对应的 COM 口，波特率一般是 115200，不需要修改，点击连接。



软件会有如上信息显示，上面有个“软启动”的按键。以后编辑完脚本，保存好（等 LED1 熄灭后），可以直接点这个“软启动”就可以启动板子了。

点击“软启动”后，可以看到 LED4 蓝灯亮 1s，灭 1s 连续闪了 4 次后，最后以熄灭的状态结束。

同样要实现的功能不要改变，我们也这样编写程序。

```
1 from pyb import LED
2 import time
3
4 led = LED(4)
5
6 led.toggle()
7 time.sleep(1)
8 led.toggle()
9 time.sleep(1)
10 led.toggle()
11 time.sleep(1)
12 led.toggle()
13 time.sleep(1)
14 led.toggle()
15 time.sleep(1)
16 led.toggle()
17 time.sleep(1)
18 led.toggle()
19 time.sleep(1)
20 led.toggle()
21 time.sleep(1)
```

下面我们学习一下常用的 LED 操作语句：

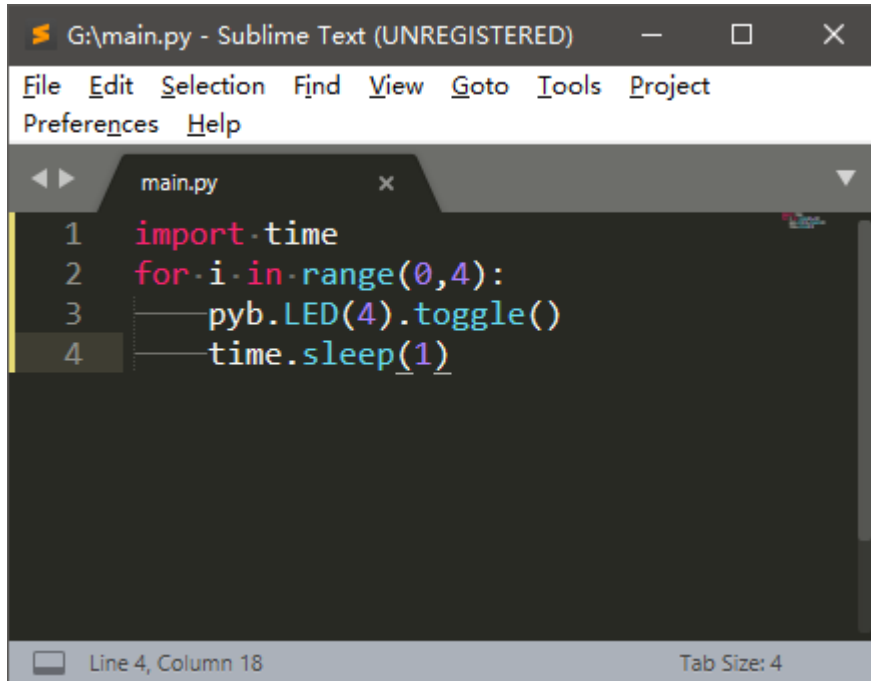
```
from pyb import LED #引用 LED 功能从 pyb 库
import time         #引用 time 库，下面我们需要引用 time.sleep ()
led = LED(4)        #将 LED(4)赋给了 led
led.toggle()        #led.toggle() LED(4)翻转
led.on()             #led.on() LED(4)点亮
led.off()            #led.off() LED(4)熄灭
```

还是上面的编程目的，开机闪 4 次也可以这样编写。

```
1 from pyb import LED
2 import time
3
4 led = LED(4)
5 for i in range(0,4):
6     led.toggle()
7     time.sleep(1)
8
```

这里用到是 for 循环，可以自行体会一下，for 循环是怎么用的。

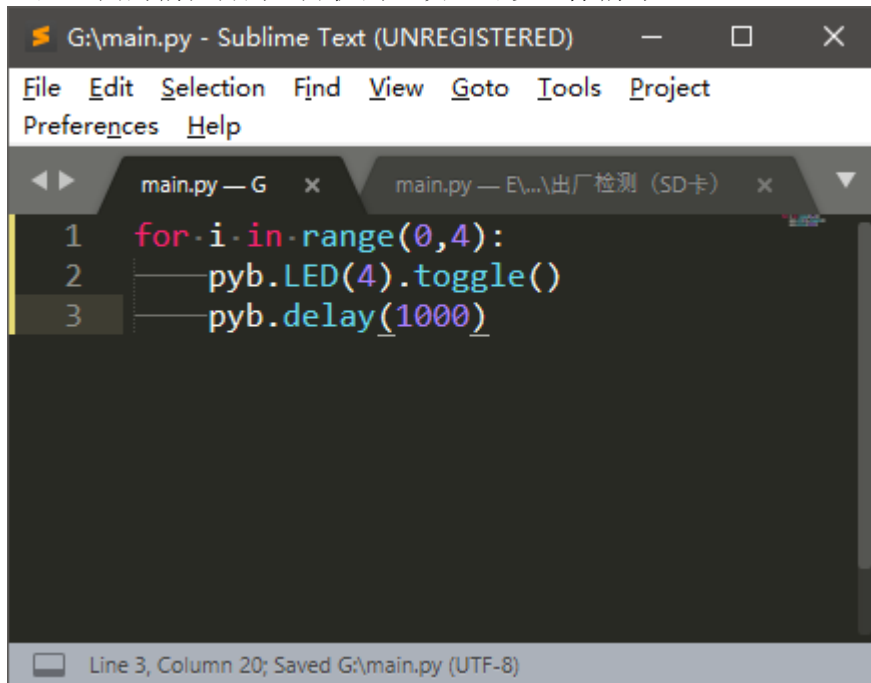
还是上面的目的，开机闪 4 次也可以这样编写。



```
G:\main.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project
Preferences Help
main.py
1 import time
2 for i in range(0,4):
3     pyb.LED(4).toggle()
4     time.sleep(1)
Line 4, Column 18 Tab Size: 4
```

这里也可以不用 `from pyb import LED` 这句话，可以省掉 `led=LED(4)`，但是对于操作需要变成 `pyb.LED(4).toggle()` 语句。可以自行体会一下。

还是上面的编程目的，开机闪 4 次也可以这样编写。



```
G:\main.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project
Preferences Help
main.py — G main.py — E:\...\出厂检测 (SD卡)
1 for i in range(0,4):
2     pyb.LED(4).toggle()
3     pyb.delay(1000)
Line 3, Column 20; Saved G:\main.py (UTF-8)
```

同样自行体会一下，到这里你会惊奇的发现，相同的功能的编程，行数从 21->7->4->3.所以 Python 是很牛的编程语言，python 编程哲学决定了总有很多骚操作，让人拍案叫绝。