



# 基于 MicroPython 的 ESP32 开发快速入门

广东职业技术学院 欧浩源 (ohy3686@qq.com) 2022-07-23

本教程通过编写程序控制“LED灯闪烁”小案例，详细讲述基于 MicroPython 的 ESP32 应用开发的基本流程和步骤明细，快速入门 MicroPython 应用开发。

**硬件平台：** ESP32 开发套件 **XMF08A/ESP32** 相关模块

**软件平台：** 串口终端软件 **uPyLoader**、代码编辑器 **NotePad++**

**案例功能：** 控制开发板上的 **D3** 灯 (**GPIO12**) 实现循环亮 1 秒，灭 1 秒的秒闪功能。

**关联知识：** 《ESP32 上烧写 MicroPython 固件明细教程》  
《终端软件 uPyLoader 使用明细教程》

《基于 MicroPython 的 ESP32 嵌入式物联网应用开发》赛教资源目录汇总专题：

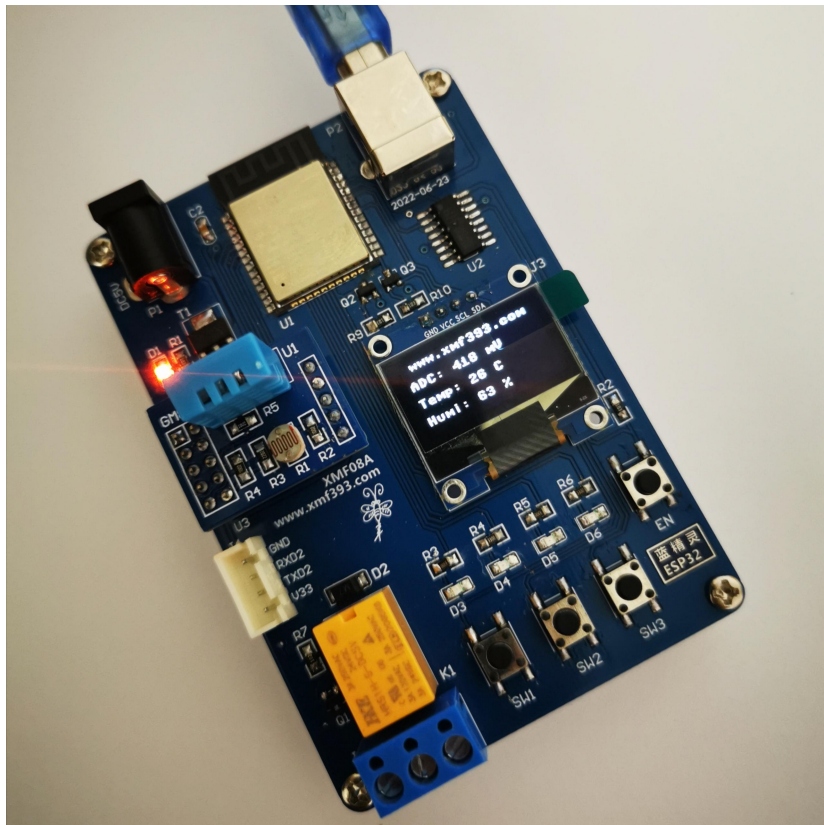
地址链接：<https://www.xmf393.com/2022/07/20/esp32/>

## 【01】烧录 MicroPython 固件。

如果你手上是一个没有 MicroPython 固件的 **ESP32** 模块或 **XMF08A** 开发板，可参照《ESP32 上烧写 MicroPython 固件明细教程》完成 **MicroPython 固件的烧录**。烧录完成后，记得按开发板上的复位键启动新系统。如果 ESP32 已完成固件的烧录，可以跳过该步骤。

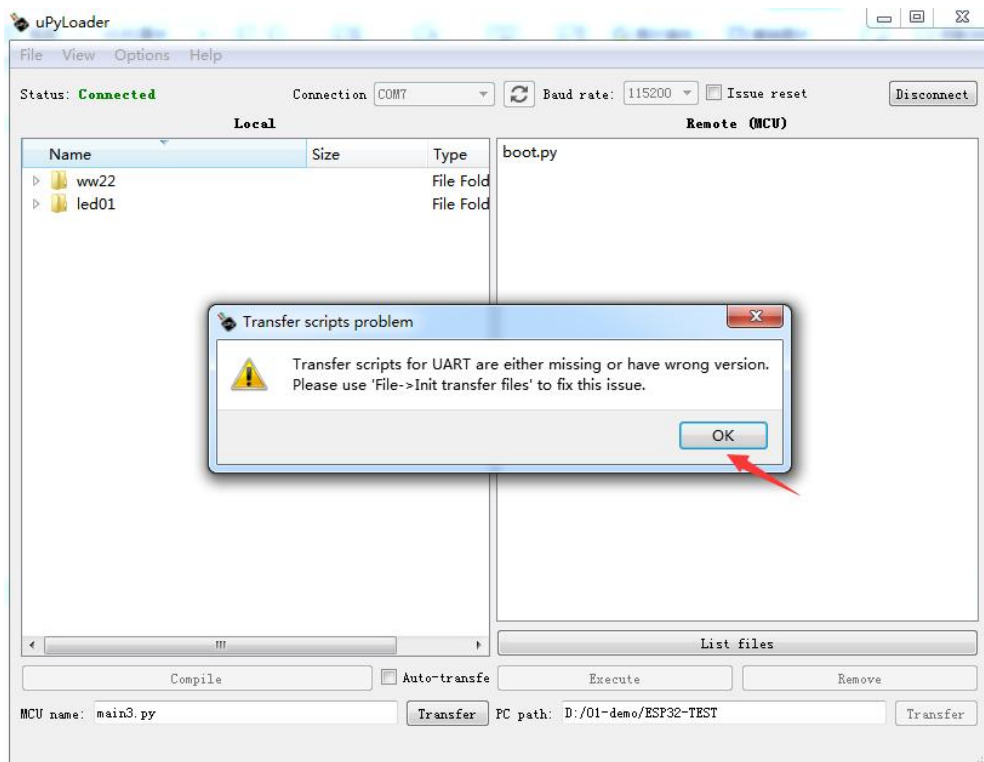
## 【02】uPyLoader 连接 ESP32。

首先，将 ESP32 模块或开发板正确连接到电脑。在电脑已经安装好 **CH340 芯片驱动**的前提下，将 ESP32 开发板 **XMF08A** 通过方头 USB 线连接到电脑上。如果你手上是其他的 ESP 开发板的，按照相关说明连接到电脑即可。

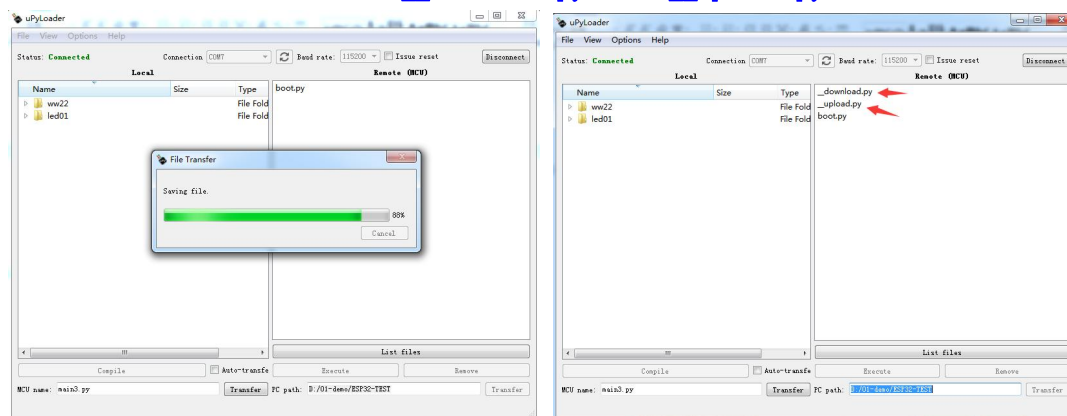




再点击“Connect”按钮进行连接。如果是 ESP32 刚刚烧录完固件，第一次与 uPyLoader 建立连接，或者缺少两个基础文件“\_\_download.py”和“\_\_upload.py”时，uPyLoader 会提示需要进行 UART 传输的初始化。



点击“OK”关闭提示框后，再点击 uPyLoader 主界面菜单栏上“File”菜单中的“Init transfer Files”菜单，便可将\_\_download.py”和“\_\_upload.py”传输到 ESP32 中。



如果你的 ESP32 芯片中已有这两个文件，可跳过上述步骤，直接进入以下 MicroPython 的程序设计环节。关于 uPyLoader 的使用详见《终端软件 uPyLoader 使用明细教程》。

### 【03】新建 PY 代码文件。

新建一个文本文件，重命名时将原后缀“.txt”改为“.py”，便创建了一个新的 Python 代码空白文件。

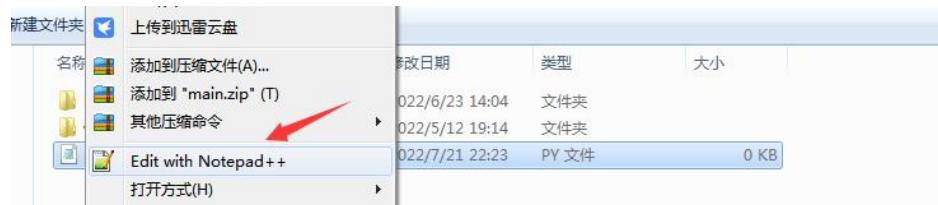
名称	修改日期	类型	大小
led01	2022/6/23 14:04	文件夹	
ww22	2022/5/12 19:14	文件夹	
main.py	2022/7/21 22:23	PY 文件	0 KB



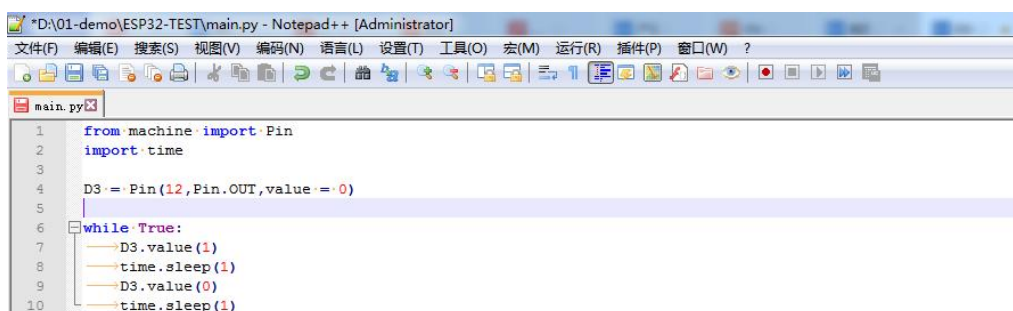
#### 【04】使用 Notepad++编写代码。

尽管使用 uPyLoader 中的“Code Editor”代码编辑器也可以实现代码的编写，但就个人开发习惯和调试经验而言，Notepad++编辑器感觉更友好，使用更方便。

首先，从 Windows “开始”菜单中，打开 Notepad++ 编辑器，或者鼠标右键单击 Python 代码文件，从弹出菜单中选择使用“Notepad++”打开。

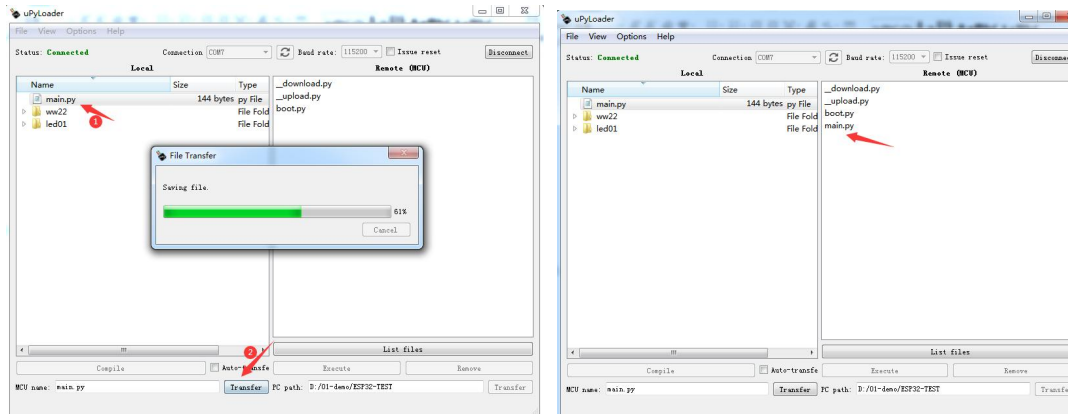


用 Notepad++ 编辑器打开代码文件后，便可根据应用需求编写合适的代码。就本教程而言，需要控制 ESP32 的 GPIO12 引脚每隔 1 秒输出高低电平，控制 D3 灯实现秒闪功能。代码编程完成后，及时保存。



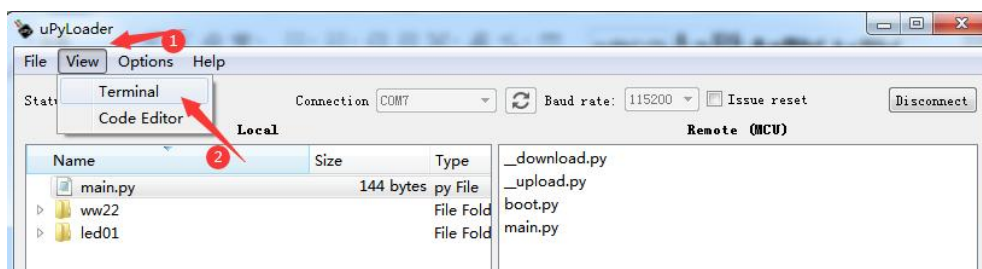
#### 【05】将代码文件传输到 ESP32。

在“计算机文件列表”中选中待传输文件“main.py”，点击左下方的“Transfer”按钮，将该 py 代码文件传输到 ESP32 芯片，传输完成该文件会出现在“MCU 文件列表”中。



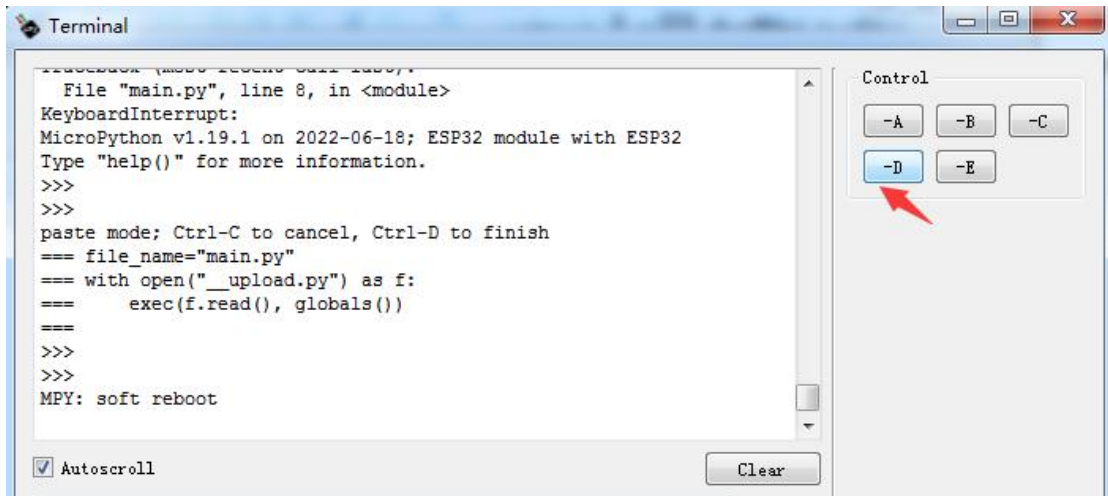
#### 【06】打开终端，执行代码。

点击 uPyLoader 菜单栏中“View”菜单中的“Terminal”菜单，可以进入终端模式。



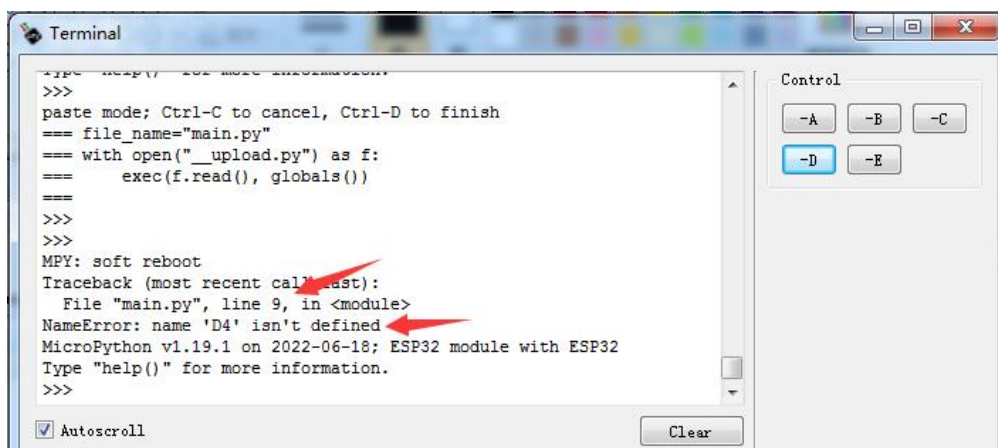


点击终端右上角“Control”控制面板中的“-D”按键，便可执行代码；在代码过程中，点击“Control”控制面板中的“-C”按键，可终止执行代码。

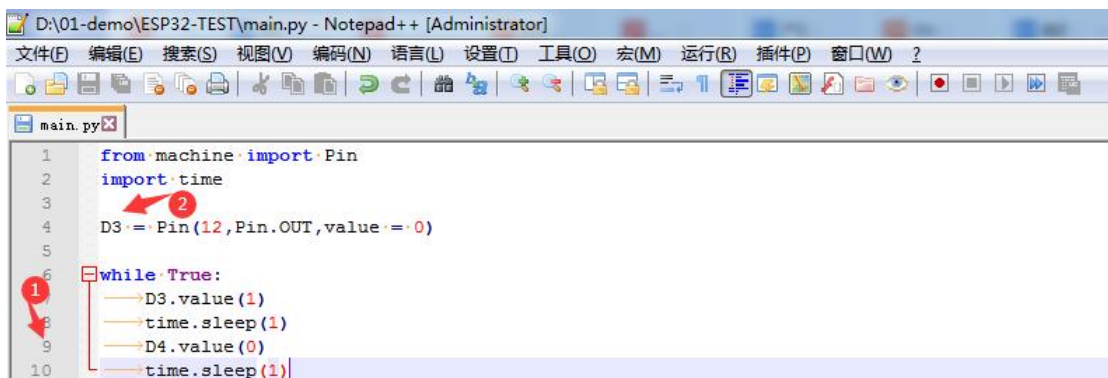


### 【07】调试代码。

MicroPython 是解释型代码，只有执行到的代码才知道是否有错误。在代码的执行过程中，遇到有问题的代码处便会终止执行，同时，通过 Terminal 终端返回错误代码的行号和错误代码的信息。这时，程序员便可根据这些信息，进行代码分析、修改和调试。



就本例而言，首先，从终端的错误信息分析可得，错误代码在第 9 行，错误的原因是，“D4”没有被定义。然后，我们鼠标双击“MCU 文件列表”中的“main.py”文件，用 uPyLoader 中的“Code Editor”代码编辑器打开，进行代码分析和修改。该编辑器不能显示代码的行号，只能人工一行一行的数！如果代码的行数比较多，会很不方便。你也可以用 NotePad++ 打开该代码文件，通过左侧的代码行号，很容易找到第 9 行。



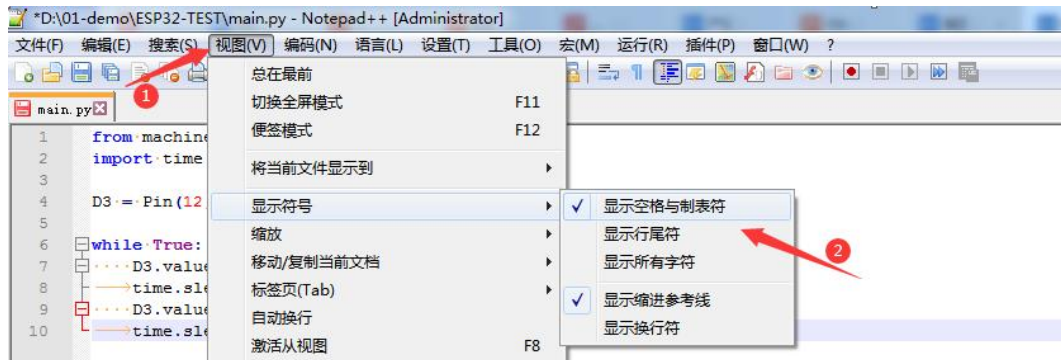


经分析，发现第9行的D4在前面的代码中并未定义，再根据案例的功能需求，发现此处的代码应该是D3。对代码进行修正后，及时保存。然后按照前面的步骤，重新将修改后的最新代码文件传输到ESP32芯片中，覆盖原来的“main.py”文件，重新执行。如果再有错误，则继续反复分析、修改、调试，直至编写代码完全正确，所需功能完全实现。

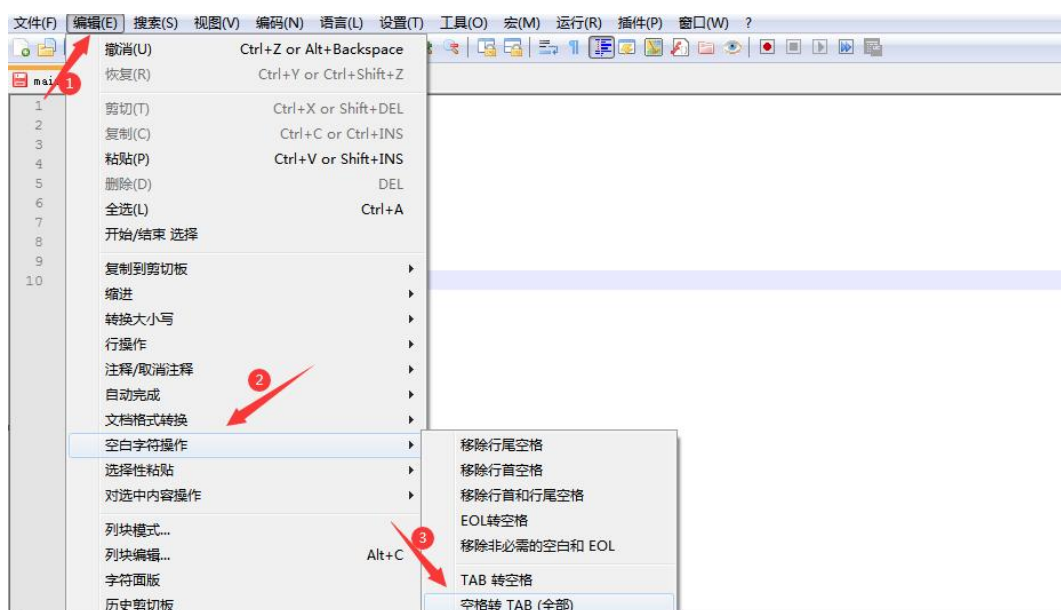
至此，基于MicroPython的ESP32应用开发的基本流程步骤全部完成。

### 【08】关于使用NotePad++几个经验。

第一个是，在“视图”中设置“显示空格与制表符”。



第二个是，每次编程程序完成时，在“编辑”中选择“空白字符操作”中的“空格转TAB(全部)”，将所有的空格统一转换成TAB制表符。



在Python中，**TAB建**和**空格**是分开的，如果在一个代码文本中有**TAB**和**空格混用**，那么在程序执行的时候，会报错“**unindent does not match any outer indentation level**”。

【ESP32 开发笔记与学习资源】地址：<https://www.xmf393.com/2022/07/20/esp32/>



### 资源目录汇总 【ESP32系列笔记】基于MicroPython的ESP32嵌入式物联网应用开发-赛教资源目录汇总 置顶

2022-07-20 小蜜蜂 阅读(37)

赞(0)

广东职业技术学院，欧浩源，以ESP32开发套件XMF08A为核心平台，开展基于MicroPython环境的嵌入式物联网应用开发，整合教学视频、开发笔记、源码分析、课件习题等，面向课堂教学，服务拓展创新，以构建全方位的赛教资源生态体系。