

ESP32-C3 状态灯客户安装与测试说明

这是一份给 Windows 用户的最小安装说明。按照下面步骤操作，可以完成：

1. 安装 Arduino CLI
2. 将程序下载到 ESP32-C3
3. 用 Python 做简单测试
4. 在 Codex 中接入状态灯
5. 做一次实际联调测试

本文尽量按“小白也能照着做”的方式写。

有经验的客户请直接跳转到第14步，[14. 速通版](#)

直接买的成品的直接跳转到第8步，[8. 用 Python 做简单测试](#)

0. 你会拿到什么

目录结构如下：

```
1 sketch_may27a/  
2 └─ .codex/  
3 └─ agent_light_control.py  
4 └─ codex_light_hook.py  
5 └─ codex_light_serial.py  
6 └─ setup_esp32c3_test_env.ps1  
7 └─ build_esp32c3.ps1  
8 └─ flash_esp32c3.ps1  
9 └─ sketch_may27a.ino  
10 └─ python_test_tools/
```

建议把整个文件夹放到：

```
1 C:\Users\你的用户名\Documents\Arduino\sketch_may27a
```

1. 硬件准备

需要准备：

- 一根支持数据传输的 USB 线
- 成品AI状态灯

注意：

- USB 线必须是“数据线”，不能只是充电线
- 如果电脑上插了多个串口设备，后面看到的 COM 口可能不是固定数字，这属于正常情况

2. 安装 Python 3

2.1 检查是否已经安装

打开 PowerShell, 输入:

```
1 | python --version
```

如果能看到类似:

```
1 | Python 3.10.6
```

就说明已经安装好了。

2.2 如果没有安装

去 Python 官网下载安装 Python 3。

安装时务必勾选:

```
1 | Add python.exe to PATH
```

安装完成后, 关闭 PowerShell, 再重新打开, 再执行一次:

```
1 | python --version
```

3. 安装 Arduino CLI

Arduino CLI 是用来编译和下载 ESP32-C3 固件的命令行工具。

3.1 最简单安装方式

在 PowerShell 中输入:

```
1 | winget install ArduinoSA.CLI
```

安装完成后, 关闭 PowerShell, 再重新打开。

3.2 检查是否安装成功

输入:

```
1 | arduino-cli version
```

如果能看到版本号, 就说明安装成功。

4. 打开项目目录

在 PowerShell 中进入项目目录：

```
1 | Set-Location C:\Users\你的用户名\Documents\Arduino\sketch_may27a
```

如果你的文件夹不在这个位置，请换成你自己的实际路径。

5. 一键安装 ESP32-C3 所需环境

在项目目录执行：

```
1 | .\setup_esp32c3_test_env.ps1
```

这个脚本会自动检查并安装：

- ESP32 开发板核心
- Adafruit 相关库
- Python 串口依赖 `pyserial`

如果最后看到了类似：

```
1 | Detected ESP32 port: COM40
```

说明电脑已经识别到了开发板。

6. 查看开发板端口

插好 ESP32-C3 后，在项目目录执行：

```
1 | arduino-cli board list
```

你会看到类似：

```
1 | COM40 serial Serial Port (USB) ESP32 Family Device
```

这里的 `COM40` 就是当前开发板端口。

注意：

- 这个端口号可能会变
- 今天是 `COM40`，明天可能变成 `COM38`
- 这是正常现象，不代表设备坏了

7. 编译并下载程序到 ESP32-C3

7.1 推荐方式：直接使用项目自带脚本

在项目目录执行：

```
1 | .\flash_esp32c3.ps1
```

这个脚本会自动做两件事：

1. 编译程序
2. 下载到当前检测到的 ESP32-C3

7.2 成功的典型输出

成功时一般能看到类似：

```
1 | Connected to ESP32-C3 on COM40
2 | Hash of data verified.
3 | Hard resetting via RTS pin...
```

7.3 如果下载失败

优先检查这几项：

- 是否使用了支持数据传输的 USB 线
- 是否选中了正确的开发板
- 是否有别的软件占用了串口
- 是否把 Arduino IDE 的串口监视器关掉了
- 是否把 Python 测试程序关掉了

8. 用 Python 做简单测试

8.1 进入 Python 测试目录

```
1 | Set-Location C:\Users\你的用户名
   | \Documents\Arduino\sketch_may27a\python_test_tools
```

8.2 安装 Python 依赖

```
1 | python -m pip install -r requirements.txt
```

8.3 发送一条测试命令

```
1 | python .\send_state.py THINKING
```

如果看到类似：

```
1 | Port: COM40
2 | Sent: THINKING
```

说明 Python 已经能把命令发给 ESP32-C3。

8.4 运行一整套状态测试

```
1 | python .\run_sequence.py
```

它会依次发送：

- IDLE
- THINKING
- WRITING
- RUNNING
- DONE
- ERROR
- NEED_CONFIRM
- TOKEN:75

8.5 如果自动识别不到串口

先手动指定端口，再执行测试：

```
1 | $env:CODEX_LIGHT_PORT="COM40"
2 | python .\send_state.py THINKING
```

把 COM40 换成你机器上实际看到的端口号。

9. 用菜单程序测试

回到项目根目录：

```
1 | Set-Location C:\Users\你的用户名\Documents\Arduino\sketch_may27a
```

运行：

```
1 | python .\agent_light_control.py
```

你会看到一个菜单：

```
1 1 IDLE
2 2 THINKING
3 3 WRITING
4 4 RUNNING
5 5 DONE
6 6 ERROR
7 7 TOKEN:75
8 q 退出
```

输入数字后回车，就会发送对应状态。

这是最适合客户自己手工试灯的方式。

10. 配置 Codex

10.1 重要说明

项目里已经自带好了：

- `.codex\hooks.json`
- `codex_light_hook.py`
- `codex_light_serial.py`

所以你要做的重点不是“自己写 hook”，而是：

1. 保留整个项目目录
2. 不要删除 `.codex` 文件夹
3. 用 Codex 打开这个项目目录

10.2 启动后台串口桥

在项目根目录执行：

```
1 .\start_codex_light_daemon.ps1
```

这个后台进程会：

- 持续占用 ESP32-C3 串口
- 接收 Codex hook 发来的状态
- 再转发到灯板

如果不启动这个后台进程，Codex hook 虽然会触发，但不会真正把状态送到灯上。

10.3 检查后台是否已经启动

执行：

```
1 Get-NetTCPConnection -LocalPort 37637 -State Listen
```

如果能看到监听信息，说明 daemon 已经正常运行。

11. 在 Codex 中做实际测试

11.1 准备动作

先确认：

- ESP32-C3 已经烧录成功
- Python 手工测试已经通过
- `start_codex_light_daemon.ps1` 已经启动
- 用 Codex 打开的是整个项目目录

11.2 测试方法

在 Codex 中正常发一个任务，例如：

```
1 | 帮我查看这个目录里有哪些文件
```

然后观察灯效。

11.3 正常情况下的状态变化

通常会看到：

- 用户提交消息后： `THINKING`
- Codex 开始执行命令时： `RUNNING`
- Codex 修改文件时： `WRITING`
- 任务结束时： `DONE`

12. 如果 Codex 里没反应，怎么排查

12.1 先看 hook 日志

在项目根目录执行：

```
1 | Get-Content .\codex_light_hook.log -Tail 30
```

如果能看到类似：

```
1 | event=UserPromptSubmit tool= command=THINKING
2 | sent=THINKING
```

说明 Codex hook 已经触发。

12.2 再看串口桥日志

执行：

```
1 | Get-Content .\codex_light.log -Tail 30
```

如果能看到类似：

```
1 | 2026-06-05 11:03:40 THINKING
2 | 2026-06-05 11:03:46 RUNNING
```

说明状态已经真正发给 ESP32-C3。

12.3 如果 Python 手工测试正常，但 Codex 没反应

优先检查：

- `start_codex_light_daemon.ps1` 有没有启动
- 是否用 Codex 打开的是整个项目目录
- `.codex` 文件夹是否还在
- `codex_light_hook.log` 里有没有新内容

13. 常见问题

问题 1：看不到串口

检查：

- USB 线是不是数据线
- 有没有换 USB 口
- 设备管理器里有没有新 COM 口

问题 2：端口号变了

这是正常情况。

重新执行：

```
1 | arduino-cli board list
```

看当前实际端口号。

问题 3：串口被占用

如果提示 `PermissionError`、`port is busy`、`could not open port`，请先关闭这些程序：

- Arduino IDE 串口监视器
- `agent_light_control.py`
- `python_test_tools` 里的测试脚本
- 旧的 daemon 进程

问题 4：只复制了 `.ino` 文件

这是不推荐的。

因为除了 `.ino` 之外，还需要：

- Python 测试脚本
- Codex hook 脚本

- `.codex\hooks.json`
- PowerShell 脚本

正确做法是复制整个项目文件夹。

14. 速通版

只想最快跑通，请按下面做：

1. 安装 Python
2. 安装 Arduino CLI
3. 打开 PowerShell
4. 进入项目目录

```
1 | Set-Location C:\Users\你的用户名\Documents\Arduino\sketch_may27a
```

5. 初始化环境

```
1 | .\setup_esp32c3_test_env.ps1
```

6. 下载固件

```
1 | .\flash_esp32c3.ps1
```

7. 测试灯

```
1 | python .\agent_light_control.py
```

8. 启动 Codex 后台串口桥

```
1 | .\start_codex_light_daemon.ps1
```

9. 用 Codex 打开这个项目目录并发送一个任务

```
1 | 帮我查看这个目录里有哪些文件
```