批量烧写工具
使用说明

版本：*1.0.1*

Contents

1 批量烧写工具简介

tg_mass_production_tool 是天猫提供的工厂批量烧写工具，支持 1 拖 12 同时烧录，支持单文件、开发、动态三种烧写模式。本文档旨在说明工厂批量烧写工具的使用步骤。用户可以进入烧录工具根目录，双击 `tg_mass_production_tool.exe`，进入主界面。

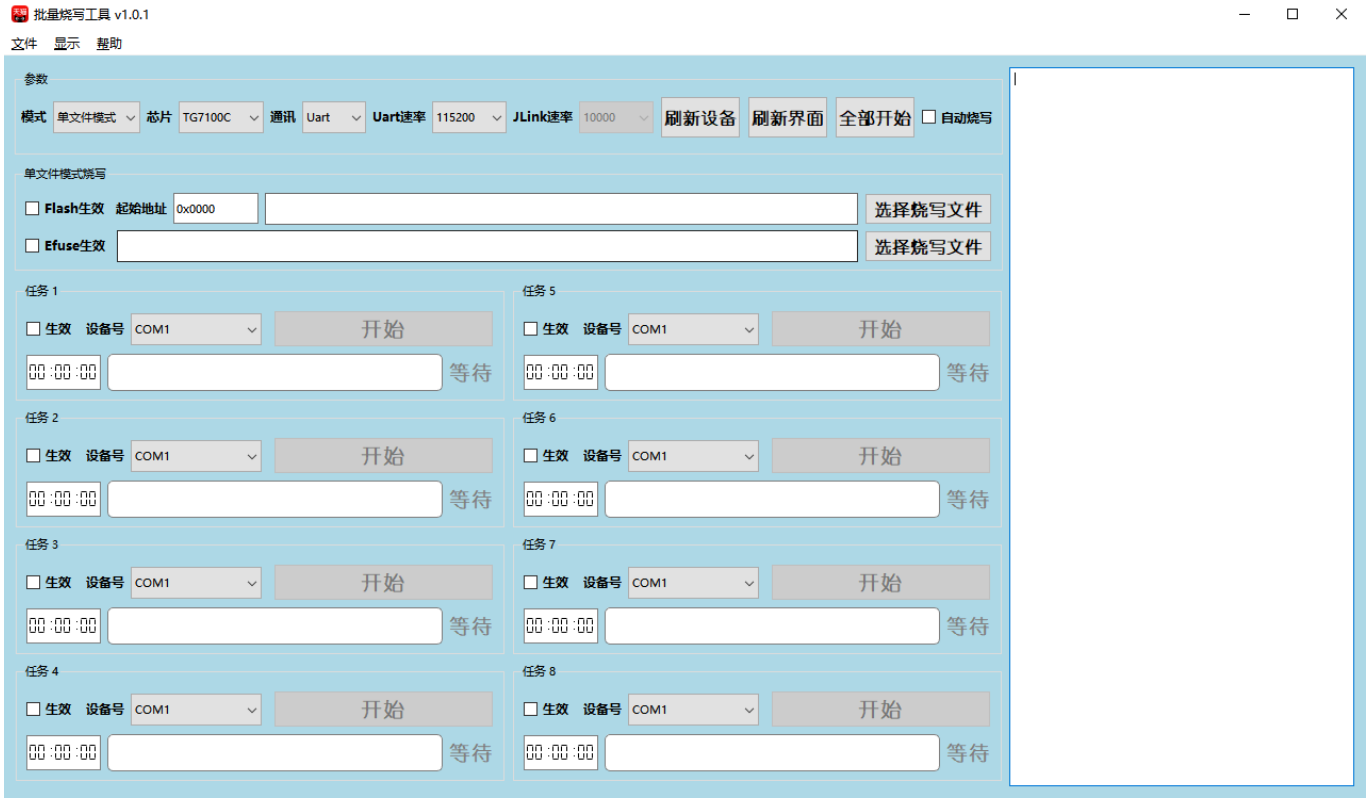

图 1.1: 主界面

工具界面主要分为 4 部分：烧写通信参数配置区，烧写文件配置区，烧写任务配置区和 Log 显示区。

2 烧录前的准备

众所周知，要实现芯片的程序烧录，需要设置芯片进入烧录模式，针对 TG 系列芯片而言，当芯片复位 (包括上电复位) 的时候，如果 Boot 引脚为高电平，则芯片会进入 UART 启动模式 (BTW:Boot 引脚为低电平的时候从 Flash 启动)，在该启动模式下，可以通过串口下载烧录程序到 RAM 中，完成 Flash 或者 Efuse 的烧录。

在工厂烧录的环境中，一般是使用夹具实现，在这种情况下，芯片的电源引脚，串口的通信引脚，以及 Boot 控制引脚都是通过顶针与芯片或者模组连接，在夹具下压的过程中，上电时序存在着不确定性，如果上电时序不能满足芯片启动要求，烧写过程就无法进行。为了保证烧录的稳定性，可以将芯片的 Boot 引脚和 Reset 引脚引出来，使用 USB 转串口的 DTR 和 RTS 来进行控制，这样烧录软件在进行烧录之前，先通过 DTR 控制 Boot 引脚为高电平。然后通过 RTS 控制 Reset 引脚，这样可以确保芯片启动后是进入烧录模式。

但是即使这样，当夹具顶针下压的时候，芯片电源供电的顶针也会存在不稳定性，我们强烈推荐使用调试烧写一体板来进行 Flash 烧写。烧写板可对目标芯片的电源、复位引脚、Boot 引脚进行控制，可以实现更高的烧写稳定性。

调试烧写板 10Pin 排线连接接口定义如下：

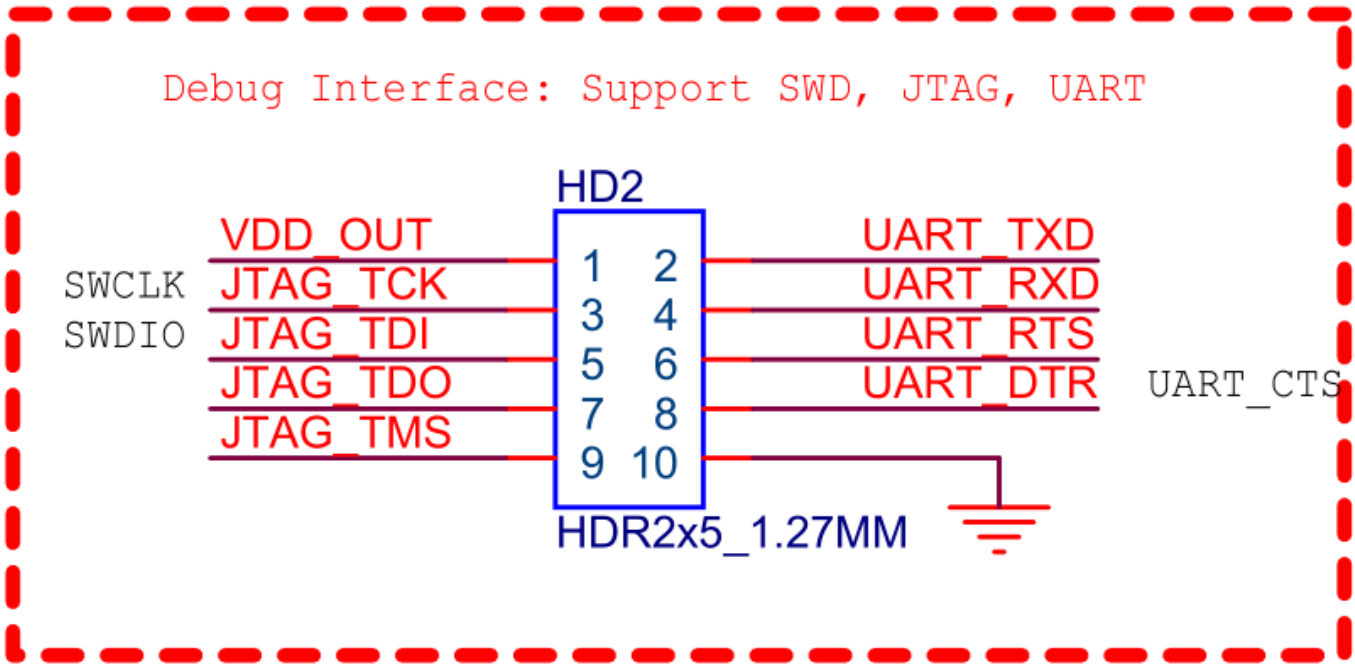

图 2.1: 烧写板连接接口定义

烧写板连接要求：

- VDD_OUT：3.3V, 与芯片/目标板 VCC 电源相连
- GND：与芯片/目标板 GND 相连
- UART_TXD：与芯片/目标板 RXD 引脚相连
- UART_RXD：与芯片/目标板 TXD 引脚相连
- UART_RTS：与芯片/目标板 RST 引脚相连，用于控制芯片的 Reset
- UART_DTR：与芯片/目标板 bootpin 引脚相连，用于控制芯片从 UART 启动

注解：TG7100C 的 bootpin 引脚是 GPIO8

将烧写板与模组连接好后，通过 Mini USB 将烧写板连接到电脑，如果驱动安装成功，会在电脑的设备管理器中出现 2 个 USB 转串口, 我们选择使用 com 口号较小的这个串口做烧录。

图 2.2: 设备管理器显示 2 个串口

烧写软件会通过选定的串口的 RTS 信号来控制模组的上电和复位，通过控制串口的 DTR 信号来控制模组的 bootpin 拉高或拉低，这样可以确保芯片进入到预期的烧写模式，提高烧写的可靠性。

3 烧写参数配置

- 配置参数包括：
  - 烧写模式：可选单文件模式、开发模式、动态模式
  - 芯片：根据实际烧写的芯片型号进行选择
  - 通讯方式：根据需求选择 Uart 或者 JLink。若选择 Uart，需要配置合适的 Uart 速率，建议选择 2M。此时 JLink 烧写相关的 JLink 速率和序列号两个下拉菜单栏自动变灰无法选择
  - 刷新设备：可刷新电脑上连接的所有串口设备或者刷新连接的 JLINK 设备
  - 刷新界面：可刷新烧写状态和进度条
  - 全部开始：用于多任务烧写同时启动，点击按钮，勾选生效的任务会全部开始烧写
  - 自动烧写：在勾选该选项后，点击全部开始，启动的各个烧写任务会一直检测芯片，检测到后自动烧写，不需要重复点击开始按钮，若检测到芯片已烧写过程序，任务框中进度条右侧会显示已烧。

图 3.1: 参数配置界面

4 烧写文件配置

4.1 单文件模式烧写

单文件模式烧写支持烧写 flash 和 efuse，可以根据实际烧写需要勾选对应的生效选择进行配置。当烧写 flash 时，需要选择烧录的 bin 文件及对应的烧录地址。当烧写 efuse 的时候，只要选择对应的烧写文件即可。如果 Flash 和 efuse 都勾选烧写，则会先烧写 flash 再烧写 efuse。

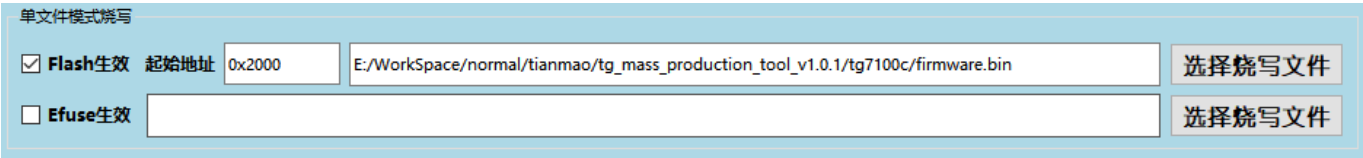

图 4.1: 文件选择界面

一般情况下，单文件模式在量产烧写 flash 时，选择的 bin 文件是 TGFlashEnv 工具生成的 whole_flash_data.bin，填写的烧写地址是 0x00。当 TGFlashEnv 工具选择好烧写文件并点击“Create&Download”后，会在其对应芯片型号目录的 img_create 文件夹下生成 whole_flash_data.bin 文件，whole_flash_data.bin 文件是将若干个待烧写 bin 文件按对应烧写地址排列打包成一个大的 bin 文件，空闲部分填充 0xFF。选择将 whole_flash_data.bin 文件烧写到 flash 的 0x00 地址，相当于将若干个待烧写文件烧写到指定的 flash 地址中。

一般情况下，单文件模式在量产烧写 efuse 时，选择的 bin 文件是 TGFlashEnv 工具生成的 efusedata.bin。当 TGFlashEnv 工具勾选了“AES-Encrypt”并按要求填写 flash 加密 Key 和 IV，点击“Create&Download”后 (镜像也会同时被加密)，会在其对应芯片型号目录的 efuse_bootheader 文件夹下生成 efusedata.bin 文件，efusedata.bin 文件内容包含 flash 加密密钥以及使能 flash 加密，选择将 efusedata.bin 文件烧写到 efuse 中，即完成 flash 加密密钥的烧写并使能 flash 加密功能。

注解: 在烧写加密过的镜像和 efuse key 时，一定是先向 flash 烧写加密固件，再烧写 efuse key，如果先烧写 efuse key，则之后的加密固件会烧写失败。因为写过 efuse key 后，芯片使能了 flash 解密功能，烧写工具无法正常给芯片发送烧写 flash 命令。

4.2 开发模式烧写

单文件模式烧写操作简单，只要选择对应的文件和设置相应的烧录地址即可。但是应用程序往往都是由若干个 bin 文件组成的，这些文件需要打包成一个大的 bin 文件，这样大的文件中包含了很多无意义的数据，如果程序 bin 文件位置分配不合理，会导致烧录的 bin 文件很大，但是真正的程序实际很小，无形中增加了生产成本。为此，我们提供开发模式烧写，在该模式下，可以一次性烧录多个文件到 Flash 的不同地址中，这样按需烧写，可以缩短整体烧录时间，提高烧录效率。

在实际的烧录场景中，可以通过导入 TGFlashEnv 工具生成的烧写文件包 (烧写文件包位于 TGFlashEnv 工具对应芯片型号目录的 img_create/whole_img.pack)，导入成功后，工具会自动配置烧写文件，使用开发人员的烧录配置完成烧录 (一般情况下，开发人员的烧录都是文件按需烧录)。这也是将其命名为开发模式烧写的原因。如下图展示了导入 TGFlashEnv 工具的“tg7100c/img_create/whole_img.pack”烧写文件包，点击“选择烧写包”按钮，在弹出的窗口找到“whole_img.pack”文件并点击打开即可。这样开发模式的烧录文件就配置好了，一般情况下，不需要再做额外的配置修改。

当 TGFlashEnv 工具生成文件包时勾选了“AES-Encrypt”并按要求填写 flash 加密 Key 和 IV，烧写文件包也会包含待烧写的 efusedata.bin，量产工具导入烧写文件包会将 efusedata.bin 一起导入，并在烧写完 flash 镜像后接着烧写 efusedata.bin 到 efuse 中。

图 4.2: 开发模式导入烧写文件包

注解: 这里 TGFlashEnv 下的 whole_imag.pack 烧写文件包，需要是在 TGFlashEnv 工具界面选择好对应烧写文件并点击“Create&Program”按钮后生成的，否则 tg7100c 文件夹下没有生成实际需要烧写的文件以及烧写文件包。

开发模式下烧写的 bin 文件和对应的烧录地址是在对应芯片型号的 eflash_loader 目录下的 eflash_loader_cfg.ini 配置文件中指定，配置文件中“file”用于配置烧写的文件，每个文件使用空格隔开，“address”用于指定对应文件的烧录

地址，地址是 16 进制，每个地址之间使用空格隔开。

注解: 不同的芯片，文件对应的路径不同，一定要根据实际烧写的芯片类型进行设置。

```
[FLASH_CFG]
flash_id = c84015
flash_clock_div = 0
#0:NIO, 1:DO, 2:QO, 3:DIO, 4:QIO
flash_io_mode = 4
#empty: auto,0: internal flash with io switch,1: internal flash no io switch,2: GPIO 0-5,3: GPIO 22,23,37-40
flash_pin = ""
#empty for auto, otherwise specified para file path: eg: ./efuse_bootheader/flash_para.bin
flash_para = ""
file = tg7100c/img_create/bootinfo_boot2.bin tg7100c/img_create/img_boot2.bin
address = 00000000 00002000
```

图 4.3: 指定烧写文件和对应的烧写地址

上图表示将文件 tg7100c/img_create/whole_img_boot2.bin 和 tg7100c/img_create/whole_img.bin 分别烧写到 flash 的 0x0 和 0x10000 地址。

TGFlashEnv 工具生成的烧写文件包,实际上是将对应芯片型号目录下的待烧写文件以及 eflash_loader 目录下的 eflash_loader_cfg.ini 配置文件一起打包成 whole_img.pack。批量烧写工具在导入烧写文件包时，会根据 whole_img.pack 文件包里的 eflash_loader_cfg.ini 文件里的“file”和“address”配置，将待烧写文件替换到对应位置，完成待烧写文件的导入。

当用户需要额外向 flash 中烧写其它文件时，只需要在 eflash_loader_cfg.ini 配置文件的“file”和“address”配置中添加文件路径和烧写地址即可。

4.3 动态模式烧写

动态模式烧写适用类似于一机一码这种对于每台待烧写设备，烧写文件有部分内容是动态变化的。选择动态模式烧写时，烧写工具会自动打开一个动态服务程序。

动态模式的烧写文件和烧写配置的设定,和开发模式类似,需要在动态服务程序中导入 TGFlashEnv 工具生成的 whole_img.pack 烧写文件包，动态服务程序会根据导入的烧写文件包替换待烧写文件。导入成功后，动态模式会自动将选择的芯片型号目录复制 12 份，分别放到 task1~task12 中，动态模式烧写时，实际是各个任务到对应的 task 中寻找烧写文件并开始烧写，每个 task 目录还会包含动态生成的文件，会一起参与烧写。

动态服务程序可以在烧写过程中自动生成动态烧写文件。动态服务程序可以根据客户实际产品烧写需求做定制化，可以提供对 MAC 地址、lisence 码等具有唯一性的标签做定制化生成烧写文件，并自动烧写。

5 烧写任务配置

在选择好烧写模式，配置完要烧写的文件或者拷贝需要烧录的文件夹以后，就可以根据实际同时烧录的芯片数量，配置对应的烧录任务，将需要启动任务的“生效”勾选，然后配置对应使用的串口号或者 Jlink 序列号，即表示该任务生效。然后点击参数选项框中的“全部开始”按钮，此时开始执行勾上“生效”框的所有设备的烧写，绿色进度条向前走动，控制台打印所有任务的日志信息，下方时钟显示执行时长。烧写成功进度条右侧会分别显示成功。

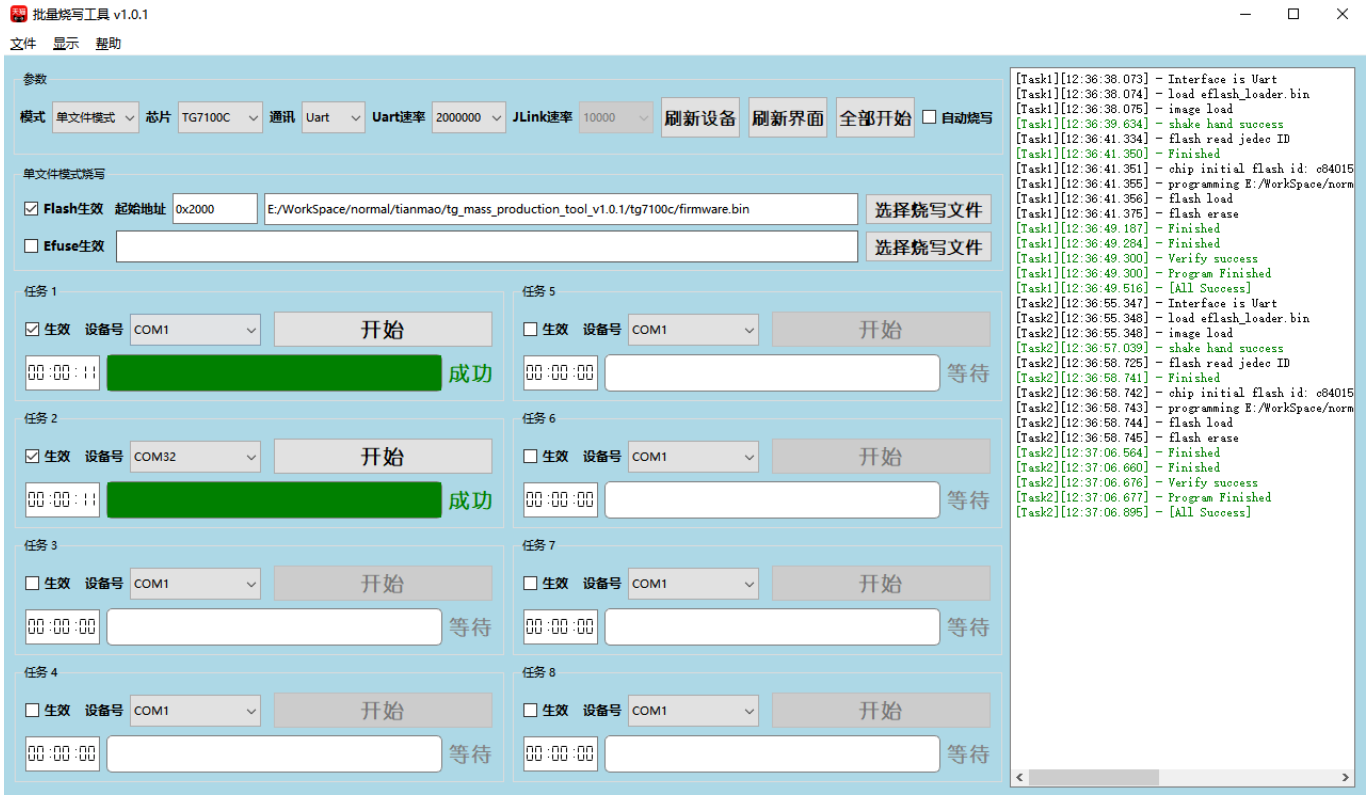

图 5.1: 多任务烧写成功

如果多任务烧写过程中，有些任务烧录失败，需要再次尝试，可以点击对应任务右侧的“开始”按钮，此时开始执行单个设备的烧写，绿色进度条向前走动，控制台打印单个任务的日志信息，下方时钟显示执行时长。烧写成功进度条右侧会显示成功。

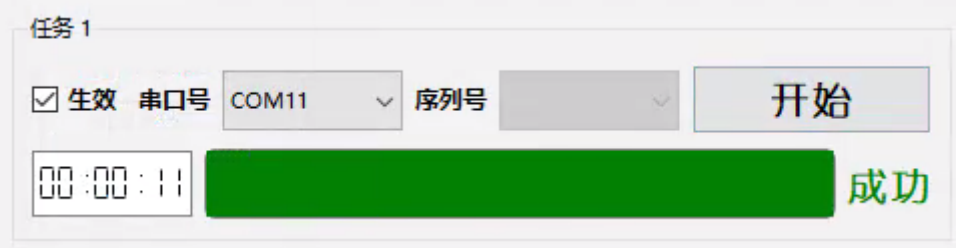

图 5.2: 单任务烧写成功

6 启动烧写

6.1 手动烧写

配置好各个任务后，点击“全部开始”按钮，开始执行所有设备的烧写，烧录只进行一次，如果某个任务烧录失败，点击对应任务的开始按钮再次进行烧录即可。

6.2 自动烧写

将参数配置中的“自动烧写”复选框勾上，点击“全部开始”按钮，则会执行自动烧写。自动烧写是指配置完烧写任务以后，该任务会一直在执行烧写任务，并自动判断是否是已经烧录的芯片。如果芯片烧录成功，则状态显示为” 成功”，由于是自动烧写，烧写任务会再次执行，这个时候，烧写任务检测到该芯片已经被烧录，则显示” 已烧”，所以” 成功” 状态是个暂态。如果更换芯片，此时烧录任务还在执行，这时候检测不到芯片，则显示” 等待”。当芯片更换完毕后，烧录软件可自动检测到新的芯片，开始新一轮的烧录。同样，如果烧写过程中出现错误，则状态变为” 失败”，由于是自动烧录模式，所以” 失败” 状态也是个暂态，会被新一轮的烧写状态替换。如果是硬件电路异常或者模组异常，烧录软件无法正确的检测到芯片，也是处在” 等待” 状态，所以，如果烧录任务长期处在” 等待” 状态，则可能是烧写失败。

1. 若检测到某一个芯片已经烧录过文件，烧录结果会显示已烧

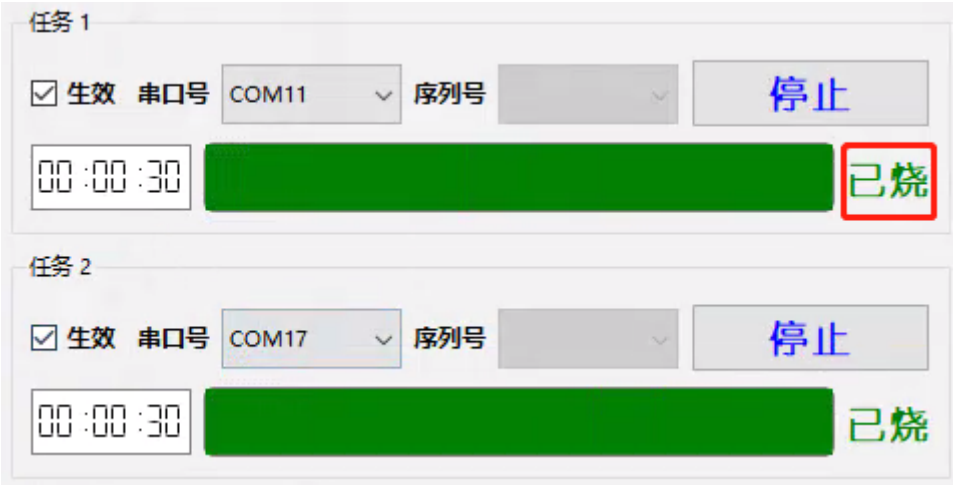

图 6.1: 检测到芯片已烧过 bin 文件

2. 在更换芯片的时候，烧录结果会显示等待

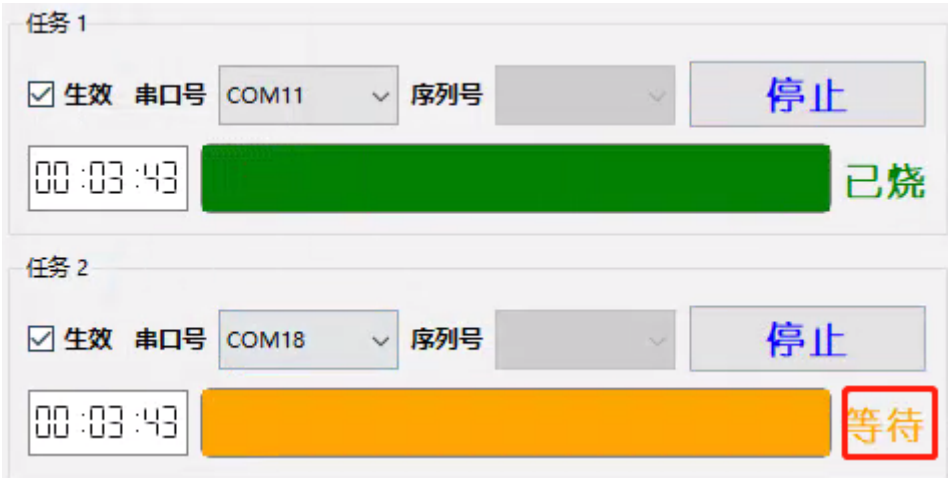

图 6.2: 等待更换芯片

3 若烧写过程中发生错误，烧录结果会显示失败

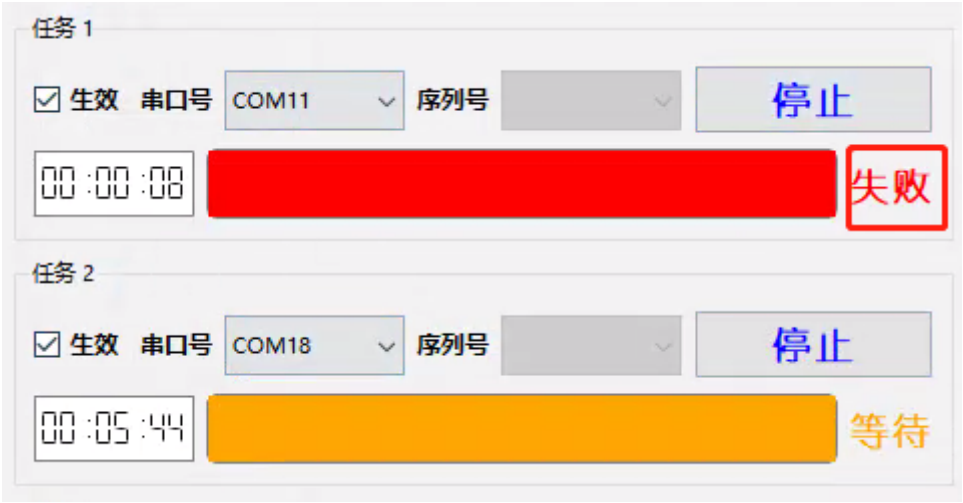

图 6.3: 烧写失败

4. 若烧写过程中想要停止烧写，则点击“停止”按钮，烧录结果会显示中止

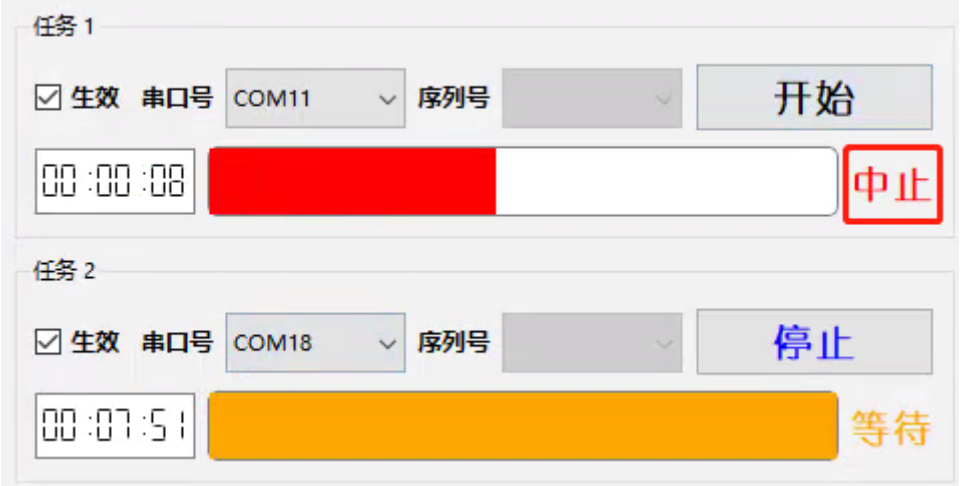
任务 1
生效
串口号
COM11
序列号
开始
00:00:08
中止
任务 2
生效
串口号
COM18
序列号
停止
00:07:51
等待

图 6.4: 中止烧写