

保密等级	A	TXW81x 量产和烧录指南	文件编号	TX-0000
发行日期	2023-11-29		文件版本	V1.0
<div>TXW81x 量产和烧录指南</div> <div></div>				
		珠海泰芯半导体有限公司 Zhuhai Taixin Semiconductor Co.,Ltd	珠海市高新区港湾一号科技园港 11 栋 3 楼	
版权所有 侵权必究 Copyright © 2023 by Tai Xin All rights reserved				

保密等级	A	TXW81x 量产和烧录指南	文件编号	TX-0000
发行日期	2023-11-29		文件版本	V1.0

责任与版权

责任限制

由于产品版本升级或者其他原因，本文档会不定期更新。除非另行约定，泰芯半导体有限公司对本文档所有内容不提供任何担保或授权。

客户应在遵守法律、法规和安全要求的前提下进行产品设计，并做充分验证。泰芯半导体有限公司对应用帮助或客户产品设计不承担任何义务。客户应对其使用泰芯半导体有限公司的产品和应用自行负责。

在适用法律允许的范围内，泰芯半导体有限公司在任何情况下，都不对因使用本文档相关内容及本文档描述的产品而产生的损失和损害进行超过购买支付价款的赔偿（除在涉及人身伤害的情况中根据适用的法律规定的损害赔偿外）。

版权申明

泰芯半导体有限公司保留随时修改本文档中任何信息的权利，无需提前通知且不承担任何责任。


未经泰芯半导体有限公司书面同意，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。除非获得相关权利人的许可，否则，任何人不能以任何形式对前述软件进行复制、分发、修改、摘录、反编译、反汇编、解密、反向工程、出租、转让、分许可等侵犯本文档描述的享有版权的软件版权的行为，但是适用法禁止此类限制的除外。

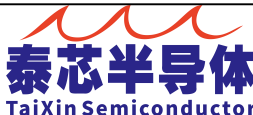


珠海泰芯半导体有限公司
Zhuhai Taixin Semiconductor Co., Ltd

珠海市高新区港湾一号科创园港 11 栋 3 楼

版权所有 侵权必究
Copyright © 2023 by Tai Xin All rights reserved

保密等级	A	TXW81x 量产和烧录指南	文件编号	TX-0000																				
发行日期	2023-11-29		文件版本	V1.0																				
<div>修订记录</div> <table><tr><th>日期</th><th>版本</th><th>描 述</th><th>修订人</th></tr><tr><td>2023-11-29</td><td>V1.0</td><td>初始版本</td><td>TX</td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>					日期	版本	描 述	修订人	2023-11-29	V1.0	初始版本	TX												
日期	版本	描 述	修订人																					
2023-11-29	V1.0	初始版本	TX																					
		珠海泰芯半导体有限公司 Zhuhai Taixin Semiconductor Co.,Ltd	珠海市高新区港湾一号科技园港 11 栋 3 楼																					
版权所有 侵权必究 Copyright © 2023 by Tai Xin All rights reserved																								

保密等级	A	TXW81x 量产和烧录指南	文件编号	TX-0000
发行日期	2023-11-29		文件版本	V1.0
<div>目录</div> <div><div><div>1. 概述..... 1</div><div>2. 量产烧录接口..... 2</div><div>3. 量产代码生成..... 3</div><div><div>3.1. 代码生成相关配置..... 3</div><div><div>3.1.1. 代码空间分配文件..... 3</div><div>3.1.2. 代码生成配置文件..... 3</div><div>3.1.3. 出厂参数配置文件..... 4</div></div><div>3.2. 代码生成..... 5</div><div>3.3. 代码加密..... 5</div></div><div>4. 芯片 FLASH 启动和运行机制..... 7</div><div>5. 代码烧写..... 8</div><div><div>5.1. MAC 地址滚码烧录..... 8</div><div>5.2. 参数文件烧录..... 8</div><div>5.3. 代码加密烧录..... 10</div><div>5.4. 离线烧录（简易烧写器：TXLINK-Lite） 10</div><div>5.5. 在线烧写..... 11</div><div><div>5.5.1. 简易烧写器：TXLINK-Lite..... 11</div><div>5.5.2. USB 烧录..... 13</div><div>5.5.3. CKLINK 烧录..... 13</div></div></div></div></div>				
		珠海泰芯半导体有限公司 Zhuhai Taixin Semiconductor Co.,Ltd	珠海市高新区港湾一号科创园港 11 栋 3 楼	
版权所有 侵权必究 Copyright © 2023 by Tai Xin All rights reserved				

1. 概述

本文主要描述 TXW81x 量产和烧录流程，包括量产 bin 文件的配置、生成、加密代码导出、以及量产烧录等流程。

本文档主要适用于以下工程师：

- 技术支持工程师
- 方案软件开发工程师

本文档适用的产品范围：

型号	封装	包装
TXW81x		

2. 量产烧录接口

量产烧录使用简易烧写器：TXLink-Lite 进行烧写，在板烧录时需要在板预留离线烧录接口：3V3(可选)、CHIP_EN、PA8、PA9、GND（不需要烧录器供电的话 3V3 可以不留），任何情况均可以离线烧录。

简易烧录器硬件如下：



离线烧录连接如下：

TXLink-Lite	Target
VREF	3V3
nRST	CHIP_EN
GND	GND
HDA	PA8
HCK	PA9

3. 量产代码生成

3.1. 代码生成相关配置

目前 SDK 基于集成开发环境 CDK 搭建，代码相关配置主要有涉及几个文件：

- 1) gcc_csky.ld: 代码空间分配文件
- 2) makecode.ini: 代码生成配置文件
- 3) parameter_ui.setcfg: 出厂参数配置文件

3.1.1. 代码空间分配文件

空间分配文件默认为 gcc_csky.ld，主要看代码空间分配，代码空间需要和 makecode.ini 的 CodeExeAddr 代码起始位置保持一致。

```
16
17 /*****
18 * @file      gcc_csky.ld
19 * @brief     csky linker file
20 * @version   V1.0
21 * @date      02. June 2017
22 *****/
23 MEMORY
24 {
25     SRAM : ORIGIN = 0x20000000 , LENGTH = 0x48000 /* on-chip SRAM 288KB */
26     SRAM2 : ORIGIN = 0x20051000 , LENGTH = 0x1800
27     FLASH : ORIGIN = 0x18000000 , LENGTH = 0x100000
28     PSRAM : ORIGIN = 0x18800000 , LENGTH = 0x100000
29 }
30
31 __heap_size = 0x11000;
32
33 PROVIDE (__ram_end = 0x20048000);
34 PROVIDE (__heap_end = 0x20048000);
35
36 REGION_ALIAS("REGION_TEXT", FLASH);
37 REGION_ALIAS("REGION_TEXT2", SRAM2);
38 REGION_ALIAS("REGION_DATA2", SRAM2);
39 REGION_ALIAS("REGION_RODATA", FLASH);
40 REGION_ALIAS("REGION_DATA", SRAM);
41 REGION_ALIAS("REGION_BSS", SRAM);
42 REGION_ALIAS("REGION_INIT", SRAM);
43
```

SRAM空间

SPI FLASH空间

PSRAM空间

代码空间

数据空间

3.1.2. 代码生成配置文件

空间分配文件为 makecode.ini，主要配置代码运行地址和 FLASH 时钟频偏，所有配置均为十六进制数。Makecode.exe 会解析 makecode.ini 配置、代码 bin 和参数配置文件生成一份带参数和头信息的代码文件，文件名会自动以 SDK 版本号，应用方案类型、日期和客户

方案名命名，比如：

txw81x_fpv_v2.5.0.7-25360_2023.10.13_TXW816_WAV_EVB.bin。

```
14 [SPI]
15 CodeCRC16=1 ; spi code crc enable
16 Flag=5A69 ; fixed
17 Version=0x0100 ; code version
18 CodeLoadToSramAddr=18000000 ; code load to sram start address, 代码运行地址, 和CDK工程ld文件保持一致
19 CodeExeAddr=18000000 ; code run start address
20 CodeAddrOffset=c00 ; code load from spi flash address Offset >= 1024 + param length
21 CodeLoadLen=2000 ; code load to sram length
22 SPI_SIZE=100000 ; spi flash size
23 SPI_CLK_MHZ=3c ; spi clk : 13Mhz
24 DriverStrength=1 ; DriverStrength : {0/1/2/3}, bigger sign stronger, FLASH 时钟频率, 配置时同步配置SampleDelay = 480/clock/2+1
25 PLL_SRC_MHZ=28 ;
26 PLL_EN=1 ;
27 DebugInfoEn=0 ;
28 AesEnable=0 ; aes encrypt enable, FLASH高级配置, 修改配置和原厂确认
29 ; advance config
30 ReadCmd=6B ; spi read cmd index : normal (03)/fast (0B)/dual (3B)/quad (6B)/QPI (EB)
31 ReadCmdDummy=8 ; dummy clk
32 ClockMode=3 ; clock mode {0,3}
33 SampleDelay=5 ; 16bit sample delay cnt in spi clk divisor
34 WireModeWhenCmd=1 ; wire mode : {1/2/4}, 6 for 3wire mode
35 WireModeWhenAddr=1 ; wire mode : {1/2/4}, 6 for 3wire mode
36 WireModeWhenData=4 ; wire mode : {1/2/4}, 6 for 3wire mode
37 WireMode4Select=0 ; 4wire mode : D2/D3 io select{0,2}
38 WireMode4En=1 ; 4wire mode enable{0,1}
39 SpecSequenceEn=1 ; spec cmd sequence before cmd read, eg.use for dual/quad cmd
40 SpecSequenceNumbers=3 ; spec cmd sequence (format: cmd + dummy + data_lens + data) numbers
41 SpecSequence0=06000000 ; spec cmd "write enable "
42 SpecSequence1=31000102 ; spec cmd "qual mode enable for winbond "
43 SpecSequence2=05800101 ; spec cmd "wait busy "
44 SpecSequence3=38000000 ; spec cmd "QPI enable"
45 SpecSequence4=C0000133 ; spec cmd "Set Read Param"
46
```

- 【CodeLoadToSramAddr】：代码要加载到的SRAM地址，一般和CodeExeAddr保持一致
- 【CodeExeAddr】：代码要运行的地址，一般和CodeLoadToSramAddr保持一致

3.1.3. 出厂参数配置文件

TXW81x 为客户提供高度个性化的出厂参数配置，用户可以根据实际需求任意修改参数名字和内容。

出厂参数配置文件为parameter_ui.setcfg和parameter.bincfg,CDK编译时,makecode会主动提取参数配置并配置到代码bin文件中。

此配置文件通过 TX_programmer.exe 工具配置，可以导入和导出。参数格式如下：

```
struct __param {
    uint16 param_len; /* 参数总长度 */
    uint16 param_flag; /* Magic Number: fix 0x2B1A */
    struct __param_fn fn0;
    struct __param_fn fn1;
    struct __param_fn fn2;

    struct __param_fn fn_x; /* 用户可自定义增加 */
}
```



```

};

struct __param_fn {
    uint8  fn_num; /* 功能码，建议用户新增从 0xFE 递减使用，避免冲突 */
                uint8  size; /* 此参数配置 size */
    uint8  buf[size]; /* 参数区*/
} ;

struct __param {
                uint16 param_len; /* 参数总长度 */
                uint16 param_flag; /* Magic Number: fix 0x2B1A */
    struct __param_fn fn0;
    struct __param_fn fn1;
    struct __param_fn fn2;

    struct __param_fn fn_x; /* 用户可自定义增加 */

};

struct __param_fn {
    uint8  fn_num; /* 功能码，建议用户新增从 0xFE 递减使用，避免冲突 */
                uint8  size; /* 此参数配置 size */
    uint8  buf[size]; /* 参数区*/
} ;

```

参数配置 Fn0/1/2 已使用，用户自定义配置需要避开。

3.2. 代码生成

代码生成相关配置配置好后，直接 CDK 编译即可生成最终 bin 文件，makecode 会同时生成明文 Bin 文件。带 PSRAM 的方案需要在 psram_cfg.h 配置正确的 PSRAM 版本。

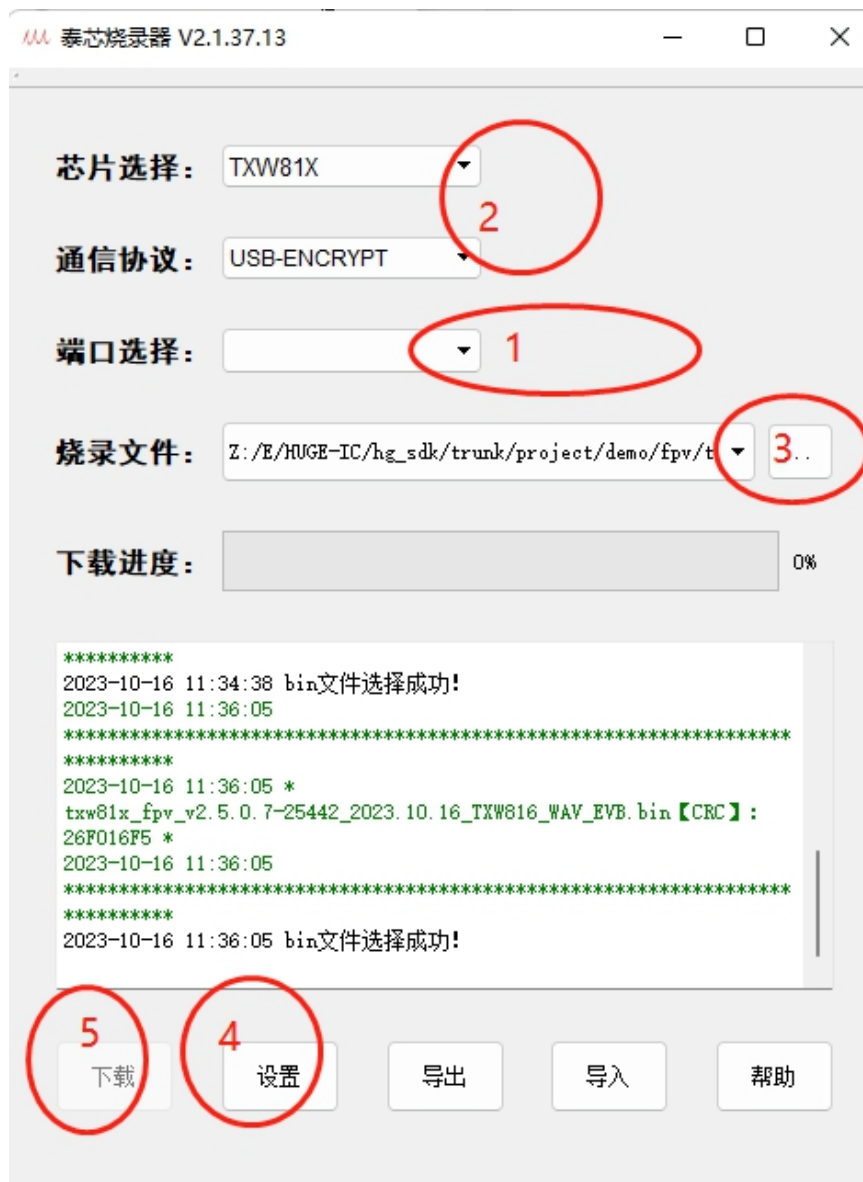
3.3. 代码加密

代码加密需要在已经烧录加密密钥（参考章节 5.3）的芯片上完成。

代码通过 TX_programmer.exe 工具进行加密，代码加密步骤如下：

- 1) 将板子通过USB连接到电脑（先擦除芯片FLASH，确保芯片FLASH内没有代码）
- 2) TX_programmer.exe 通信协议到“USB-ENCRYPT”
- 3) 选择编译出来的明文固件，例如：
`《txw81x_fpv_v2.5.0.7-25360_2023.10.13_TXW816_WAV_EVB.bin》`
- 4) “设置”->“芯片设置”->“master boot”保存
- 5) 点击“启动”开始导出加密代码
- 6) 在Infor窗口看到加密代码导出成功信息，加密代码会在原文件名后面增加后缀“_gen”

上述操作示意图如下：



4. 芯片 FLASH 启动和运行机制

TXW81x 支持双代码共存。芯片启动会默认从 FLASH 查找和加载代码，按照 256KB 步进遍历 2MB FLASH 空间查找代码，最多允许同时存在两份代码。代码查找完后会自己加载、校验和执行最新的一份代码，如果最新代码有问题（被破坏、跑死等），芯片则会智能自动回滚到旧的一份代码执行，防止代码异常后出现无法正常开机和升级等问题。

双代码机制通常留给固件需要升级的场景使用，但是如果 FLASH 大小和烧录时间允许，出厂前也是可以直接内置双代码的，这种方式可以提高产品稳定性和使用寿命（如果 FLASH 异常导致其中一份代码异常）。

双代码空间支持组合如下：

- 1) 128 KB 对齐地址boot (512KB/1024KB FLASH)
- 2) 256 KB 对齐地址boot (2/4/8MB FLASH)
- 3) 512 KB 对齐地址boot (16MB FLASH)
- 4) 1024 KB 对齐地址boot (32MB FLASH)

如果需要初次烧录时内置两份代码，可以使用 SDK 自带 BinScript 工具或者其他工具 Merge Bin 文件，两份代码中间空余位置填充 0xFF.

5. 代码烧写

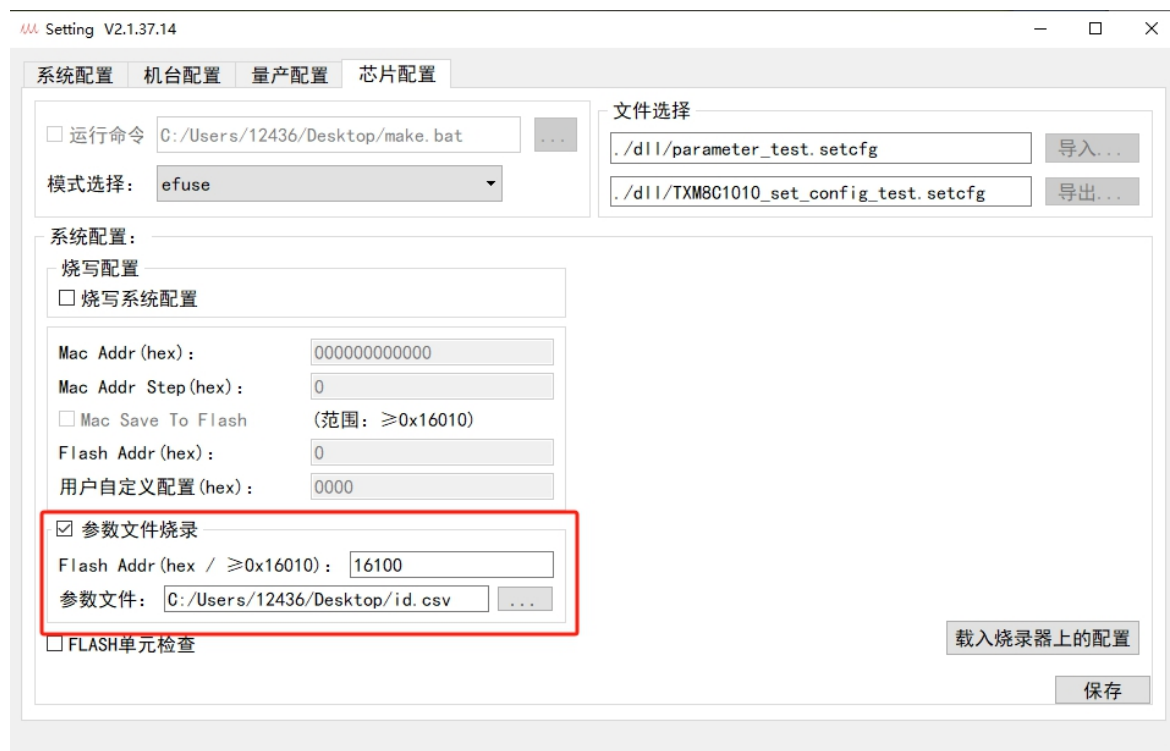
5.1. MAC 地址滚码烧录

烧写器支持 Mac 地址滚码烧录，在“设置”->“芯片设置”->模式选择“efuse”后设置 Mac 地址。如果需要将 Mac 地址存放到 SPI Flash，则勾选“Mac Save To Flash”，并且配置存放地址（可重复烧录）；如不勾选，Mac 地址会烧录到 efuse（仅允许烧录一次）。另外，用户自定义配置也是烧录到 efuse（仅允许烧录一次），配置界面如下：



5.2. 参数文件烧录

烧写器支持烧录客户自定义的参数，如 Cloud ID。在“设置”->“芯片设置”->模式选择“efuse”->选择准备好的参数文件，并配置到存放的地址。



参数文件“id.csv”格式如下：

HEAD	A	B	C	D	E	F	G	H	I	J	K	L	M	N
2	DGOC-000001-RFPYV	AMKSZR	KJK	AAA										
3	DGOC-000002-HJFTL	HRLJIA	DSD	BB										
4	DGOC-000003-TWYVN	NGIPYV	ADF	CCCCC										
5	DGOC-000004-RFPYV	AMKSZR	KJK	DDD										
6	DGOC-000005-HJFTL	HRLJIA	DSD	EEEE										
7	DGOC-000006-TWYVN	NGIPYV	ADF	FFFF										

烧录到目标芯片的内容格式如下：

	FC(x)_Flag/(=0x01,0x02,...)	FC(x)_Len/(数据长度+CRC8长度)	FC(x)_Data	FC(x)_CRC8/(数据的CRC)
长度(Byte)	1	2	n	1
			行数据中一个格子的内容	

5.3. 代码加密烧录

对于需要加密保护的方案代码，烧录是需要配置密钥文件，只有密钥文件和加密代码匹配方案代码才能正常运行。密钥文件为：efuse_key_***.dll，配置如下：

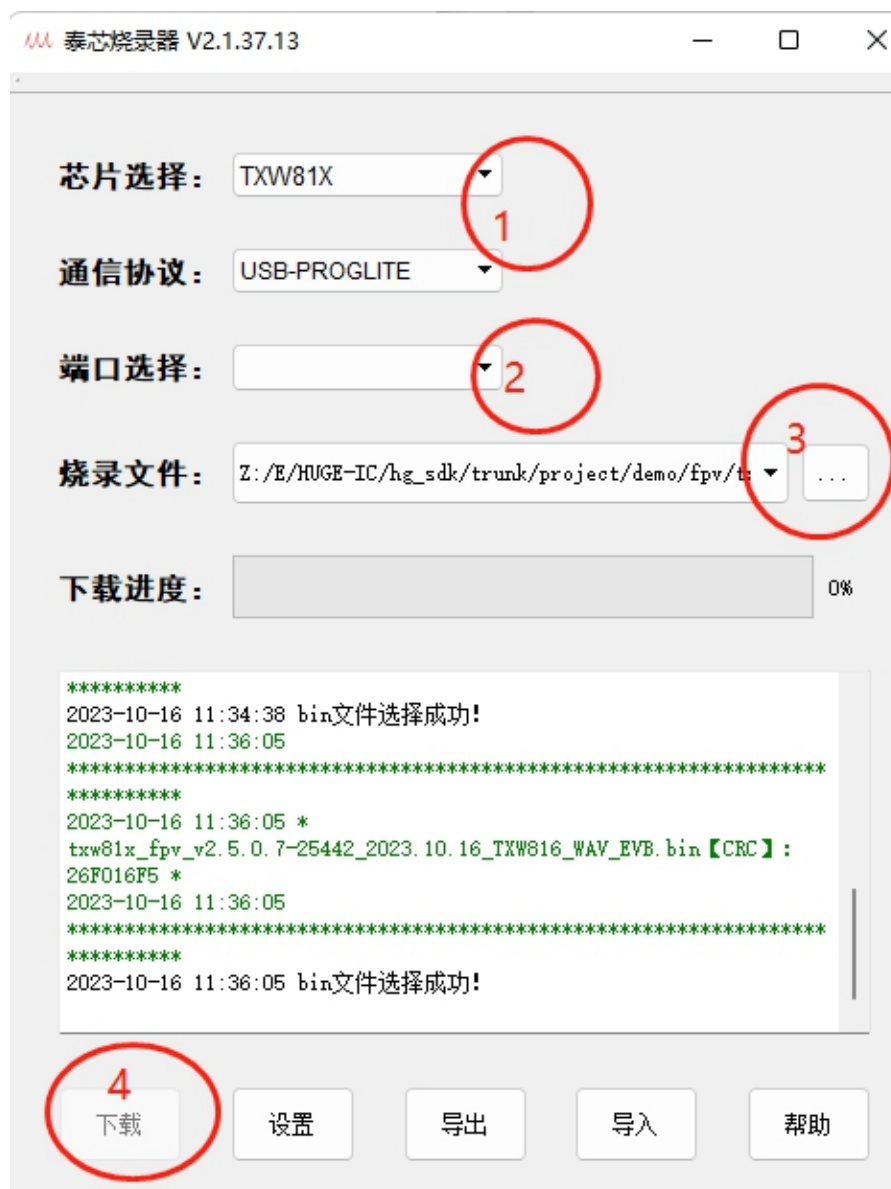


5.4. 离线烧录（简易烧写器：TXLINK-Lite）

烧写器 PC 软件

- 1) TX_programmer.exe 通信协议到 “USB-PROGLITE”
- 2) 将简易烧写器通过USB连接到电脑
- 3) 选择需要烧录的文件（明文或者密文）
- 4) 点击“启动”开始更新烧写器
- 5) 更新成功后将芯片Socket连接到烧写器进行手工烧录或者机台烧录

烧写器更新操作示意图如下：



5.5. 在线烧写

5.5.1. 简易烧写器：TXLINK-Lite

在线烧录配合简易烧写器（TXLINK-Lite）支持最多 1 拖 18 烧录。烧录各项细节参数配置和离线烧录一致。

在线烧录一拖多烧录时请注意通道绑定，通道会优先绑定先插入的设备，新插入的通道会清除该通道旧的烧录状态信息；

在线烧录烧写器移除时，通道将闪烁提示设备移除，并且保留该通道的烧录状态信息显

示，以供确认烧录失败和成功状态。

在线烧录一拖多烧录时配置如下：



5.5.2. USB 烧录

对于支持 USB 的方案或者第一次烧录的芯片可以使用 PC 工具直接在线烧录，操作步骤如下：

- 1) TX_programmer.exe通信协议到“USB-BOOT”
- 2) 将目标板通过USB连接到电脑
- 3) 选择需要烧录的文件（明文或者密文）
- 4) 点击“启动”开始烧写目标芯片
- 5) 烧录成功后断电重启运行代码

5.5.3. CKLINK 烧录

详见《SDK 快速入门手册.pdf》章节 2.4。