



TXW80X 量产和烧录指南




注意

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，本文档仅作为使用指导，不做任何担保。

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

保密等级	A	TXW80X 量产和烧录指南	文件编号	TX-0000																								
发行日期	2022-08-08		文件版本	V1.1																								
<div>修订记录</div> <table><tr><td>日期</td><td>版本</td><td>描 述</td><td>修订人</td></tr><tr><td>2022-08-08</td><td>V1.1</td><td>1、增加在线烧录描述</td><td>TX</td></tr><tr><td>2022-07-22</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	2022-08-08	V1.1	1、增加在线烧录描述	TX	2022-07-22	V1.0	初始版本	TX												
日期	版本	描 述	修订人																									
2022-08-08	V1.1	1、增加在线烧录描述	TX																									
2022-07-22	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创园港 11 栋 3 楼																									
版权所有 侵权必究 Copyright © 2022 by Tai Xin All rights reserved																												

保密等级	A	TXW80X 量产和烧录指南	文件编号	TX-0000
发行日期	2022-08-08		文件版本	V1.1

目录

1. 概述.....

1

2. 量产烧录接口.....

1

3. 量产代码生成.....

2

3.1. 代码生成相关配置.....

2

3.1.1. 代码空间分配文件.....

2

3.1.2. 代码生成配置文件.....

3

3.1.3. 出厂参数配置文件.....

3

3.2. 代码生成.....

5

3.3. 代码加密.....

5

3.3.1. 不带 PSRAM 的方案.....

6

3.3.2. 带 PSRAM 的方案.....

8

4. 芯片 FLASH 启动和运行机制.....

8

5. 代码烧写.....

9

5.1. MAC 地址滚码烧录.....

9

5.2. 代码加密烧录.....

10

5.3. 离线烧录（简易烧写器：TXLINK-Lite）.....

10

5.4. 在线烧写.....

11

5.4.1. 简易烧写器：TXLINK-Lite.....

11

5.4.2. USB 烧录.....

13

5.4.3. CKLINK 烧录.....

13



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 © 2022 by Tai Xin All rights reserved

1. 概述

本文主要描述 TXW80X 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
TXW80X	QFN56/48/40/36/32	

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：txw80x_fpv_v2.0.0.7-17963_2022.5.24_TXW806_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Id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 FLASH 时钟频率，配置时同步配置SampleDelay = 480/clk/2+1
24 DriverStrength=1 ; DriverStrength : {0/1/2/3}, bigger sign stronger
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 squence before cmd read, eg.use for dual/quad cmd
40 SpecSequenceNumbers=3 ; spec cmd squence (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

TXW80X 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：

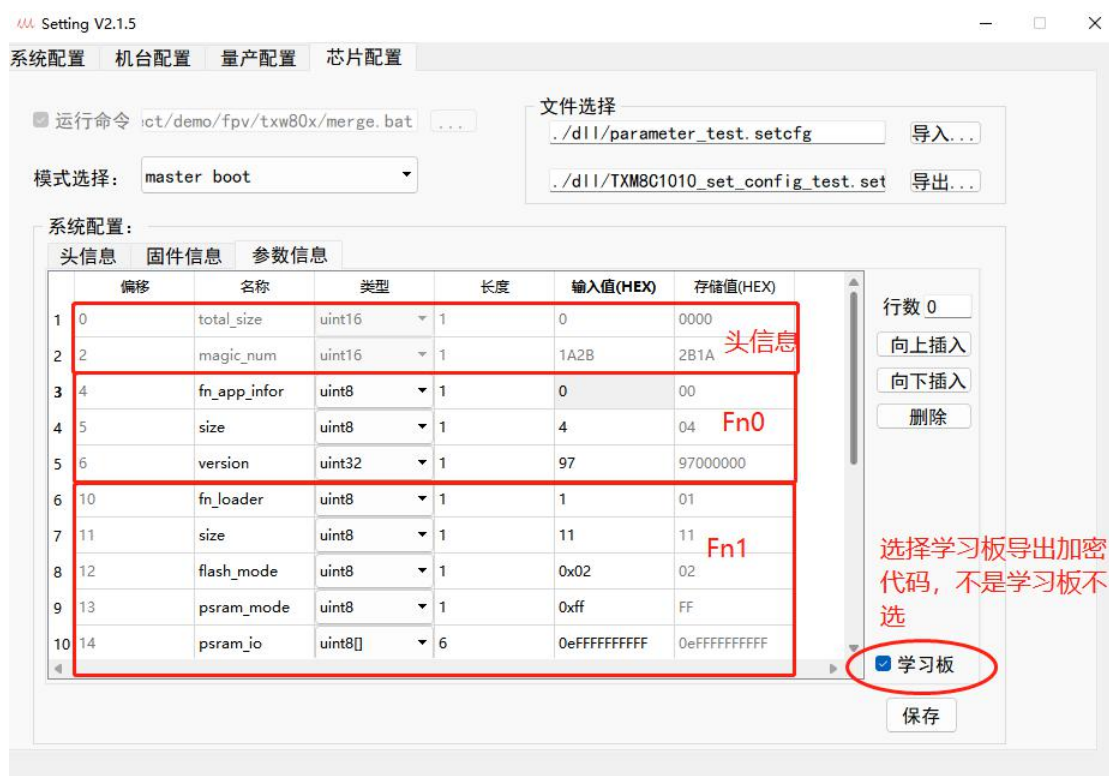
```
struct __param {
    uint16 param_len; /* 参数总长度 */
    uint16 param_flag; /* Magic Number: fix 0x2B1A */
    struct __param_fn fn0;
    struct __param_fn fn1;
```

```

struct __param_fn 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*/
} ;

```

参数配置界面如下：



注意：对于需要外挂 PSRAM 的方案，参数配置区 Fn1 一定要配置。

Fn1 配置说明如下（十六进制）：

【flash_mode】：

- 0:spi 模式
- 1:单线 qspi 模式
- 2:4 线 qspi 模式
- 0xff:无效错误

【psram_mode】：

- 0:异常(因为 psram 不存在 spi 模式, 所以不能设置为 0)
- 1:单线 qspi 模式

2:4 线 qspi 模式

0xff:代表与 flash 共用 io(flash 与 psram 共用 io[除了 cs], 所以当前 flash 一定要设置为 1[单线模式]或者 2[4 线模式])

【psram_io】: 配置 psram 的 io, 分别有 CS/CLK/D0/D1/D2/D3, 在 psram_mode=0xff 的时候, 只有 cs 有效, 在 psram_mode=1 或者 2 的时候, 所有引脚有效, 8bit 代表一个引脚, 前 3bit 代表 PA PB PC 等, 后 5bit 代表 pin_num, 例如:0E 代表 PA14, 2E:代表:PB14, 3E:代表 PB30.

例如:psram_io: 0FFFFFFF, 代表:CS:PA14, CLK、D0、D1、D2、D3 代表 0xff(无效)

【psram_clk_mhz】:设置 psram 的 clk 频率, 例如设置为 3c, 代表 (60*1M) Hz

【flash_clk_mhz】:与 psram_clk_mhz 计算方式一致

【psram_driver_strength】:设置 psram 的 io 驱动能力

【flash_driver_strength】:与 psram_driver_strength 一致

【psram_sample_dly】:设置 dly 值, 推荐使用 0xff(自动配置)

【flash_sample_dly】:与 psram_sample_dly 一致

【uart_tx】:设置 loader 的 uart 打印引脚, 前 3bit 代表 PA PB PC 等, 后 5bit 代表 pin_num

【uart_clk】:设置 uart 的波特率, 设置值*100 为实际波特率, 例如设置值 2400:代表是 (9216*100)波特率

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 会同时生成两份 Bin 文件, 均为明文。其中 bin 文件名后缀带 “_PSRAM” 的是给带 PSRAM 的方案使用的, 另外一个 Bin 是不带 PSRAM 的方案使用, 例如:

《txw80x_fpv_v2.0.0.7-17963_2022.5.24_TXW806_WAV_EVB》: 不带 PSRAM 的方案使用

《txw80x_fpv_v2.0.0.7-17963_2022.5.24_TXW806_WAV_EVB_PSRAM》: 带 PSRAM 的方案使用, 参数配置区 Fn1 一定要配置正确。

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 5.2) 的芯片上完成。

3.3.1. 不带 PSRAM 的方案

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：
《txw80x_fpv_v2.0.0.7-17963_2022.5.24_TXW806_WAV_EVB》
- 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：





3.3.2. 带 PSRAM 的方案

对于带 PSRAM 的方案，需要在不带 PSRAM 方案加密代码基础上额外执行《merge.bat》，主要有两种方式：

方案一：配置工具运行命令选择 merge.bat（“设置” -> “芯片设置”）

方案二：手动将已经加密的 Bin 文件拖拽到《merge.bat》上即可。

带 PSRAM 的方案加密 bin 文件名后缀带“_PSRAM”。

4. 芯片 FLASH 启动和运行机制

TXW80x 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、256 KB + 256 KB

2、512 KB + 512 KB

3、1024KB + 1024KB

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. 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：



5.3. 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：



5.4. 在线烧写

5.4.1. 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：

芯片选择: TXW80X ▼

通信协议: USB-PROGLITE-OL 1 ▼

在线烧录

端口选择: TXUSB-1 ▼

烧录文件: ./Release/project.bin ▼ ...

下载进度: 0%

2022-08-08 17:06:09 烧录文件异常: 文件不存在!
2022-08-08 17:06:09 bin文件选择失败!

2

启动

设置

导出

导入

帮助



5.4.2. 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

5.4.3. CKLINK 烧录

详见《SDK 快速入门手册.pdf》章节 2.4。