# TY\_RTL871x\_SDK开发手册

## 1 目录结构

TY\_RTL871x\_SDK的目录结构如下图1所示：

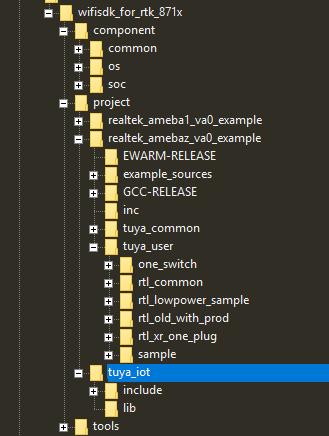


图1 目录结构

1） component：realtek底层环境。

os ：操作系统（FreeRTOS）相关。

soc ：硬件层面相关。

common ：外设驱动相关。

2）project : 项目相关。

realtek\_amebaz\_va0\_example：8710bn对应此目录。

GCC\_RELEASE：编译相关文档。

tuya\_common: gpio、led、key以及uart驱动以及tuya\_main.c(tuya\_iot入口)。

tuya\_user:具体的项目，例如rtlbn\_sample。

realtek\_ameba1\_va0\_example：8711af对应此目录，子目录请参考8710bn。

tuya\_iot：tuya\_iot库libtuya\_iot\_lib.a以及相关头文件。

3）tools : 工具。

amebaz-image-tool-v2.2：烧写工具。

arm-none-eabi-gcc：编译工具，由于realtek已经把编译工具集成到此目录，因此开发人员不需要另外安装编译工具。

## 项目开发

### 2.1 新建项目

在tuya\_user目录下新建一个目录，目录名字即代表项目名称，项目名要以”rtlbn\_”(模块为8710bn)和“rtlaf\_”(模块为8711af)开头，例如小睿科技的单插项目可以命名为rtlbn\_xr\_one\_plug。建好项目目录后，具体的设计可以参考sample。

### 2.2 编译项目

模块如果是8710bn则将当前工作目录切到project/realtek\_amebaz\_va0\_example，如果是8711af则将工作目录切换到project/realtek\_ameba1\_va0\_example，该目录下有编译脚本build\_app.sh。执行命令 “sh build\_app.sh 项目名 版本号” ，以小瑞单插为例的编译命令为:sh build\_app.sh rtlbn\_xr\_one\_plug 1.0.1。

特别地，开发调试结束之后，执行命令“sh build\_app.sh rtlbn\_xr\_one\_plug 1.0.1 mp”生成用于生产固件要用的bin文件 mp\_rtlbn\_xr\_one\_plug\_(1)\_1.0.1.bin.

要清除编译输出文件执行：sh build\_app.sh 项目名 版本号 clean。

### 2.3 输出文件

还是以小瑞单插为例，在执行命令“sh build\_app.sh rtlbn\_xr\_one\_plug 1.0.1”之后，可以在tuya\_user/rtlbn\_xr\_one\_plug/output/1.0.1/ 目录看到三个bin文件（若执行过带mp选项的编译命令则会多一个bin文件），如下图2所示：

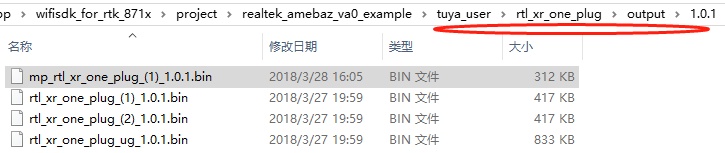


图2 输出文件

realtek代码区有A、B区之分，因此编译输出bin文件也有分别适用于A、B区的rtlbn\_xr\_one\_plug\_(1)\_1.0.1.bin和rtlbn\_xr\_one\_plug\_(2)\_1.0.1.bin文件，还有用于ota升级的

rtlbn\_xr\_one\_plug\_ug\_1.0.1.bin文件以及用于生成生产固件用的mp\_rtlbn\_one\_plug\_(1)\_1.0.1.bin（这个是要执行带mp选项的编译命令才有的）文件。

### 2.4 烧写

### 2.4.1 烧写工具

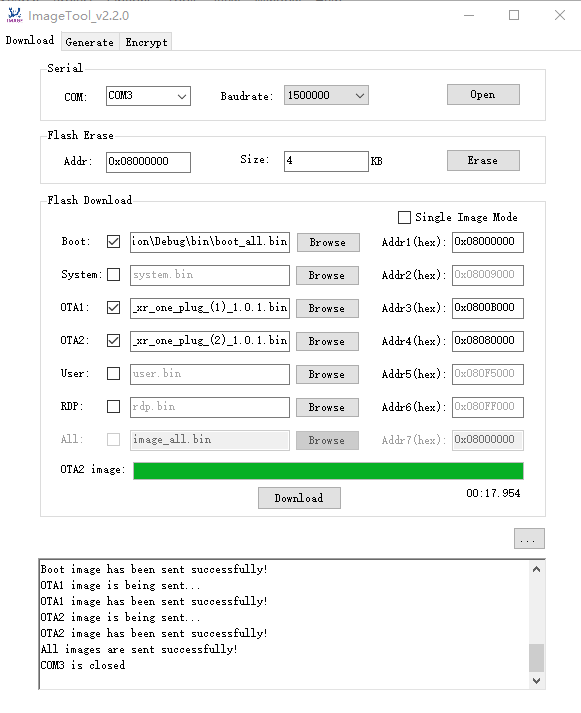
烧写工具在tools/amebaz-image-tool-v2.2，打开ImageTool.exe，界面如下图3所示：

图3 烧写界面

### 2.4.2 烧写步骤

realtek烧写步骤有点繁琐，具体如下：

1. 选择烧录文件：boot：GCC-RELEASE\application\Debug\bin\boot\_all.bin。

ota1:编译输出文件的A区bin文件即rtlbn\_xr\_one\_plug\_(1)\_1.0.1.bin。

ota2:编译输出文件的B区bin文件即rtlbn\_xr\_one\_plug\_(1)\_1.0.1.bin。

1. 选择串口号（模块调试串口），点击open，再点击download。
2. 在模块端，将模块的日志串口的tx拉低，再重新上电，再将tx连到电脑的rx。
3. 在3操作之后就可以看到下载工具的进度条在走，如果这时进度条没走，请按3步骤重新操作一遍。
4. 烧写结束之后断电重启设备就可以正常跑起来。

注：3步骤这么操作的原因是重新上电之后，bootloader在监测到tx为低就会进入下载流程。

### 2.4.3 生产固件

点击工具的Generate,界面如下图4所示：

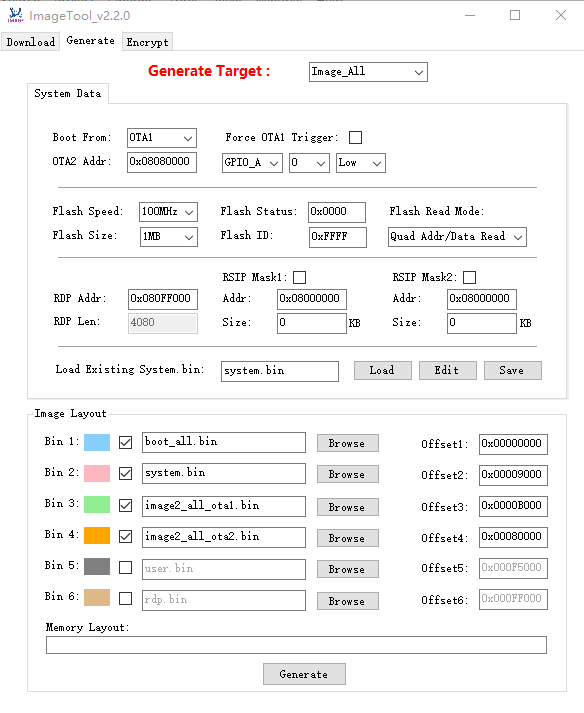


图4 生成生产固件界面

1. Generate Target：填写生产固件名称，默认是Image\_all.bin。
2. Flash Speed、Flash Size和Flash Read Mode：根据实际的flash参数填写，tuya内部的realtek模块选用的flash则分别填写100MHz、1MB和Quad Addr/Data Read。
3. Load Existing System.bin 若项目目录里找不到system.bin,则点击save保存system.bin。
4. 选bin文件：

Bin1栏的boot\_all.bin在工程目录GCC-RELEASE\application\Debug\bin\ 下。

Bin2栏的system.bin就是3 保存的文件。

Bin3栏选择编译时带mp选项生成的mp\_rtlbn\_xr\_one\_plug\_(1)\_1.0.1.bin文件。

Bin4栏的文件选择编译生成的rtlbn\_xr\_one\_plug\_(1)\_1.0.1.bin文件。

注：在上传固件到固件管理后台时，生产固件那栏上传的是生产固件。

1. 烧写生产固件：如图3所示，在boot区选择生成的生产固件，去掉OTA1和OTA2的选项，再启动烧录步骤。烧录结束后重启，程序是在厂测流程里，可以敲入ATSC命令，重启后就会进入B区的代码。敲入ATSC后再敲入ATSR，重启后还是进入厂测流程。

注：在生成生产固件之前，要调用SetLogManageAttr接口将DEBUG功能关掉，否则影响响应速度。

## api描述

### 3.1 公共接口，详见”tuya\_iot\_com\_api.h”

### 3.1.1 tuya\_iot\_init

功能描述：用于tuya iot sdk系统的初始化，必须最先调用。

### 3.1.2 tuya\_iot\_get\_sdk\_info

功能描述：获取tuya iot sdk版本信息。

### 3.1.3 tuya\_iot\_upgrade\_gw

功能描述：联网模块固件升级处理接口。

### 3.1.4 tuya\_iot\_force\_reg\_gw\_ug\_cb

功能描述：联网模块固件升级处理接口，相比3.1.3的接口，此接口具有更高优先级。

### 3.1.5 tuya\_iot\_upgrade\_dev

功能描述：设备固件升级接口库，包含网关上的主控设备以及各终端子设备的升级。

### 3.1.6 tuya\_iot\_get\_dp\_desc

功能描述：获取某个设备特定功能点的描述信息。

### 3.1.7tuya\_iot\_get\_dp\_prop\_value

功能描述：获取某个设备特定功能点的缓存数据。

### 3.1.8 tuya\_iot\_get\_dev\_if

功能描述：获取设备信息接口。

### 3.1.9 tuya\_iot\_gw\_bind\_dev

功能描述：网关绑定子设备接口，当涂鸦app使能网关添加子设备时，调用此接口将发现的子设备绑定。

### 3.1.10 tuya\_iot\_gw\_unbind\_dev

功能描述：网关解绑子设备接口。

### 3.1.11 tuya\_iot\_dev\_online\_stat\_update

功能描述：网关更新子设备在线/离线状态。

### 3.1.12 dev\_report\_dp\_json\_async

功能描述：子设备结构化数据异步上报接口，后台保障数据上传的可靠性。

### 3.1.13 dev\_report\_dp\_raw\_sync

功能描述：子设备透传数据同步上报接口，由调用者保障数据上报的可靠性。

### 3.1.14 dev\_report\_dp\_stat\_sync

功能描述：子设备结构化数据同步上报接口，由调用者保障数据上报的可靠性，通常用于统计数据的上报。

### 3.1.15 tuya\_iot\_sys\_mag\_hb\_init

功能描述：启动子设备心跳管理能力。

### 3.1.16 tuya\_iot\_set\_dev\_hb\_timeout

功能描述：子设备心跳超时时间设置，若网关再超过设置时间内未收到子设备心跳，则网关会将子设备设置为离线。

### 3.1.17 tuya\_iot\_fresh\_dev\_hb

功能描述：子设备刷新超时离线时间

### 3.1.18 tuya\_iot\_gw\_subdevice\_update

功能描述：子设备固件版本更新

### 3.1.19 tuya\_iot\_dev\_traversal

功能描述：子设备遍历，通过此接口可以遍历网关下所有的子设备。

### 3.2 wifi配置类设备接口，详见”tuya\_iot\_wifi\_api.h”

### 3.2.1 tuya\_iot\_set\_wf\_gw\_prod\_info

功能描述：设置配置类wifi设备的授权信息，授权信息需通过涂鸦后台获取，否则设备无法正常使用。

### 3.2.2 tuya\_iot\_wf\_mcu\_dev\_init

功能描述：wifi mcu 设备初始化接口。

### 3.2.3 tuya\_iot\_wf\_soc\_dev\_init

功能描述：wifi soc 设备初始化接口。

### 3.2.4 tuya\_iot\_wf\_gw\_init

功能描述:wifi网关初始化接口。

### 3.2.5 tuya\_iot\_wf\_gw\_dev\_init

功能描述：wifi网关加设备初始化接口。

### 3.2.6 tuya\_iot\_reg\_get\_wf\_nw\_stat\_cb

功能描述：获取wifi状态接口。

### 3.2.7 tuya\_iot\_wf\_gw\_unactive

功能描述：wifi设备重置接口。

### 3.2.8 tuya\_iot\_set\_user\_def\_ap\_if

功能描述：设置wifi设备ap配网时默认的ssid和passwd。如不设置，设备处于ap状态时，默认热点名称为：SmartLife-xxxx（xxxx为mac地址后四位）。

### 3.2.9 tuya\_iot\_wf\_nw\_cfg\_ap\_pri\_set

功能描述：设备重置配网时ap优先还是smartconfig优先，默认smartconfig优先。

### 3.2.10 tuya\_iot\_gw\_wf\_user\_cfg

功能描述：当通过其他模式配网时（非ap 非smartconfig），比如摄像头的二维码配网，声波配网，调用此接口处理。

## 授权流程说明

SDK厂测实现机制：创建了一个高优先级的线程。在线程里不断接收用户串口数据，若收到启动测产命令则进入厂测流程，待厂测结束之后重启设备。若线程在500ms之内未收到厂测启动命令，则会退出该线程并启动正常流程。厂测具体协议请查看《涂鸦WIFI模块标准通用产测协议1.0.4》。

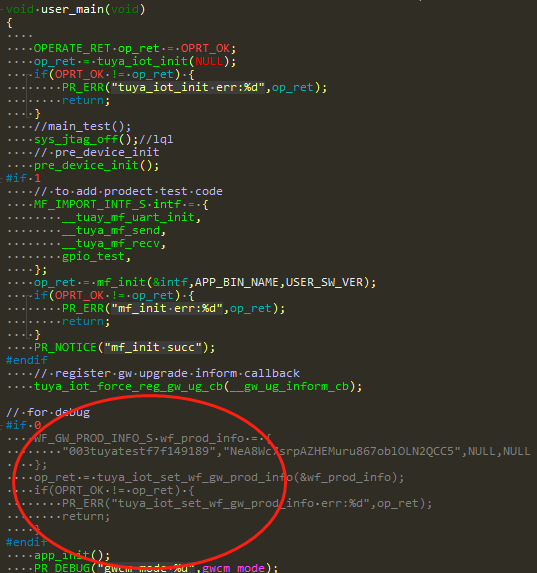
特别地，若设备未进行过厂测，设备无法正常启动。但在调试过程中可以在user\_main()函数手动写入厂测信息，如下图5所示：

图5 手动写入厂测信息代码示例

将标红处#if 0 改成 #if 1 就可以正常启动设备。需要注意的是经过厂测之后要将此处改回#if 0。

## 应用层的入口函数介绍

1. tuya\_main.c : user\_main() :

tuya\_iot的入口函数,tuya\_iot的各模块初始化（定时器，事件，flash相关）、厂测流程、网关升级注册、最后启动用户代码入口函数device\_init()。

1. tuya\_device.c : device\_init() :

启动设备初始化流程比如soc设备tuya\_iot\_wf\_soc\_dev\_init、启动wifi联网状态监测定时器以及项目相关功能。

**6flash读写接口**

flash读写接口头文件 tuya\_ws\_db.h，用户会接触到的是wd\_common\_write 和 wd\_common\_read这两个接口，以及oem固件的配置信息读写接口ws\_db\_user\_param\_read 和 ws\_db\_user\_param\_write。

**7日志打印相关**

日志等级如下：

#define LOG\_LEVEL\_ERR 0 // 错误信息，程序正常运行不应发生的信息

#define LOG\_LEVEL\_WARN 1 // 警告信息

#define LOG\_LEVEL\_NOTICE 2 // 需要注意的信息

#define LOG\_LEVEL\_INFO 3 // 通知信息

#define LOG\_LEVEL\_DEBUG 4 // 程序运行调试信息，RELEASE版本中删除

#define LOG\_LEVEL\_TRACE 5 // 程序运行路径信息，RELEASE版本中删除

可以通过SetLogManageAttr接口设置相应的打印等级，例如SetLogManageAttr(LOG\_LEVEL\_INFO)，那么LOG\_LEVEL\_INFO以下等级的日志信息（LOG\_LEVEL\_DEBUG 和 LOG\_LEVEL\_TRACE）就不会打印。

**8gpio 测试**

Gpio测试只需要在tuya\_device.c的gpio\_test接口调用gpio\_test\_cb即可，gpio\_test\_cb的传参是BOARD\_TYPE，定义如下：

#define RTL\_BOARD\_WR1 0

#define RTL\_BOARD\_WR2 1

#define RTL\_BOARD\_WR3 2

这三种分别代表realtek的WR1、WR2和WR3。