密级状态:绝密(　　)　秘密()　内部资料(√)　　公开()

文档编号:(芯片型号) –ASR6501/ASR6502 (英文、数字)

ASR6501/ASR6502 应用开发 DEMO 说明

文件状态: [√] 正在修改 [　] 正式发布	当前版本:	V0.1
	作者:	Ruilin Hao
	启动日期:	2019-04-04
	审核:	
	完成日期:	2019-04-10

翱捷科技（上海）有限公司

ASR Microelectronics Co., Ltd

版本历史

版本号	修改日期	作 者	修 改 说 明
V0.1	2019.04.10	Ruilin Hao	Initial Version

Table of Contents

1 概述

本文档主要对 ASR6501/ASR6502 SDK 中的 lorawan demo 程序进行说明，方便客户在 ASR6501/ASR6502 上进行应用程序的二次开发。

2 准备

2.1 硬件准备

LoRa 节点必需硬件列表如下：

1） ASR6501/ASR6502 LoRa 子板 1 个

2） ASR6501/ASR6502 LoRa 母板 1 个

3） 天线 1 根

4） USB 线 2 根

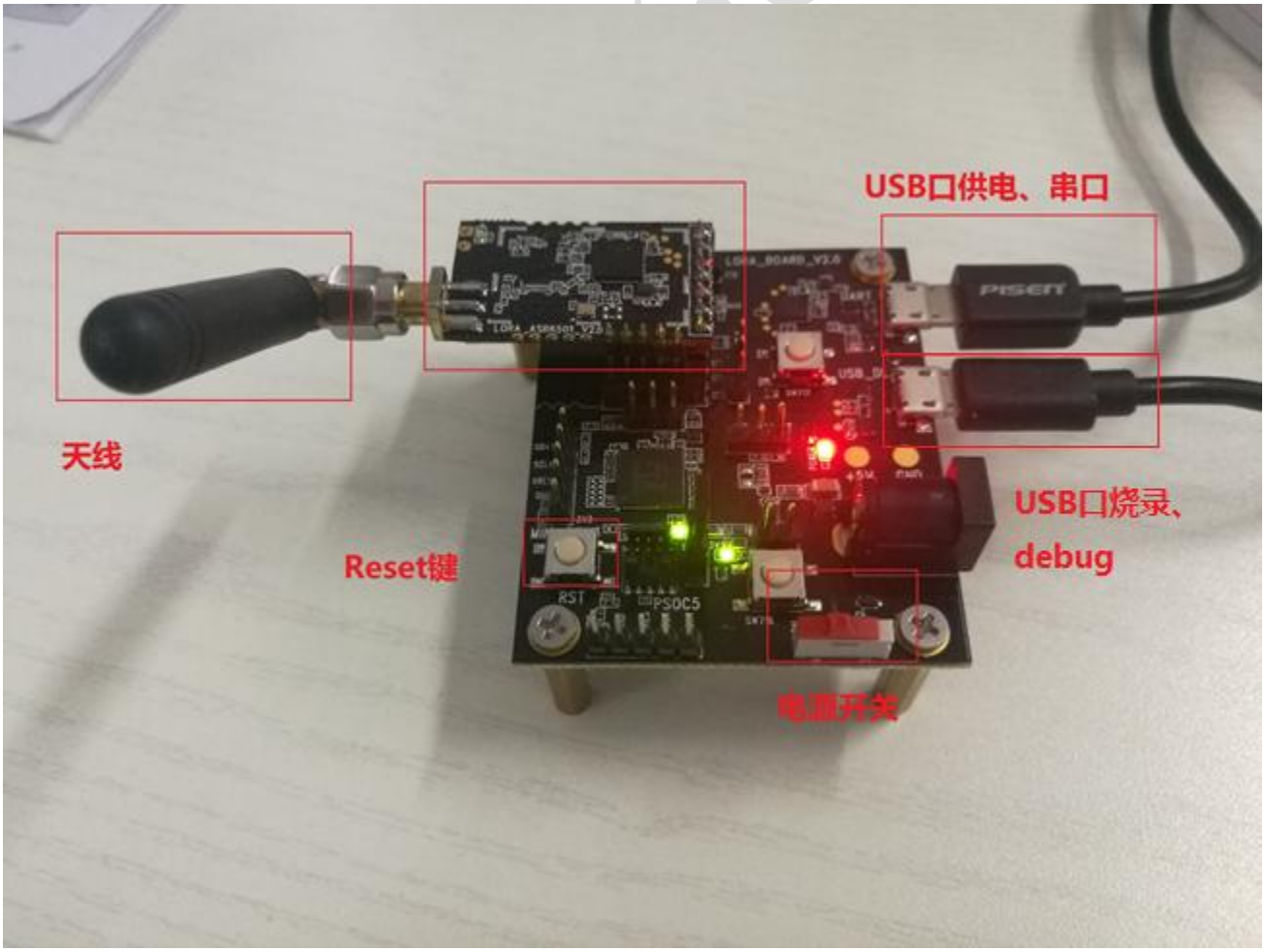
5） PC 机 1 台

另外，针对 LoRaWan/LinkWan 联网通信还需要以下硬件：

1） LoRaWan/LinkWan 网关及对应的服务器

针对 LoRa 点对点通信还需要以下硬件：

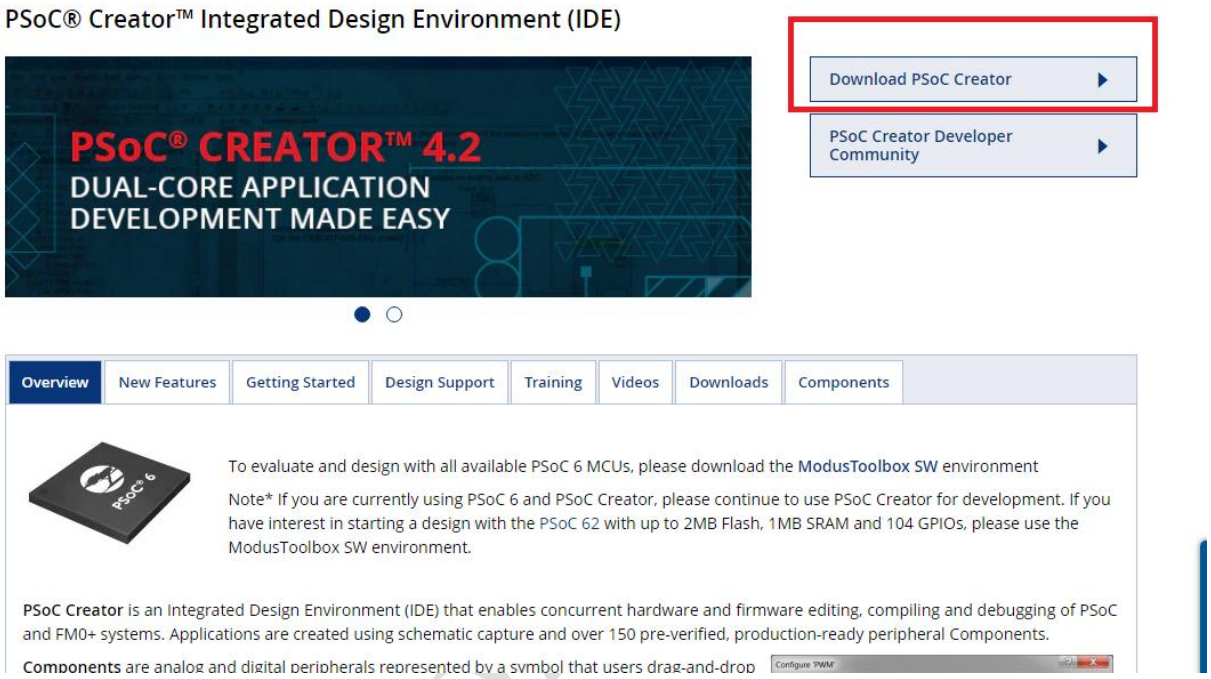
1） 另外一套 LoRa 节点硬件



硬件连接图

2.2 软件安装

首先登陆 Cypress 官网下载并安装 Creator IDE(https://www.cypress.com/products/psoc-creator-integrated-design-environment-ide)



2.3 SDK 获取

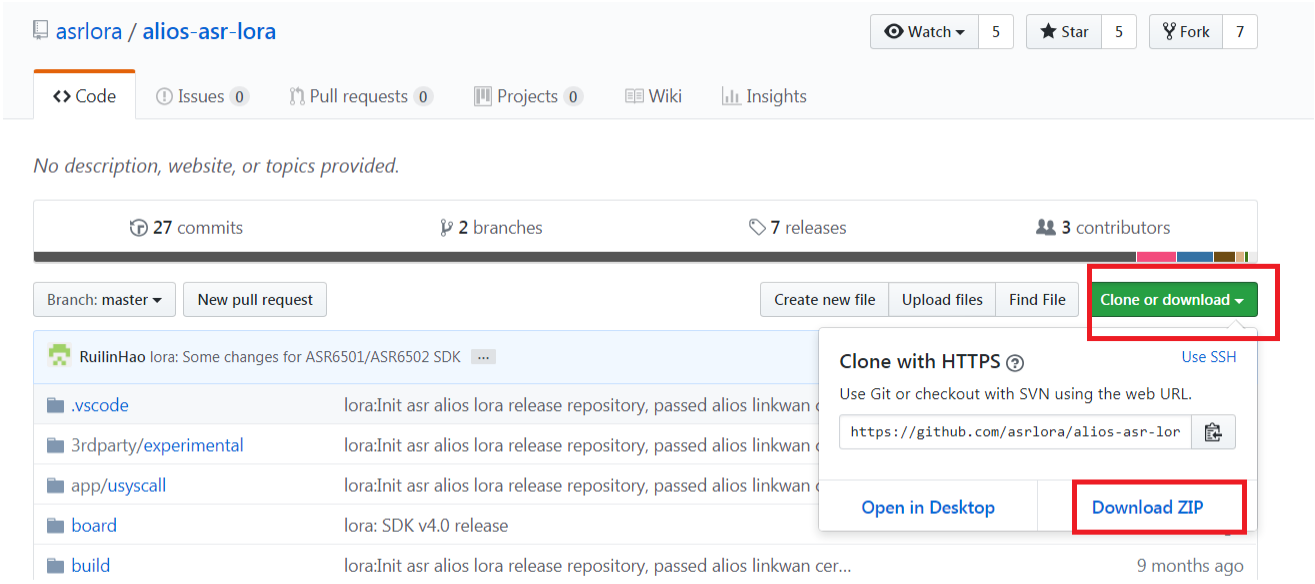
2.3.1 Git 获取

如安装有 git，可以直接使用 git clone 来获取代码，命令如下：

```
git clone https://github.com/asrlora/alios-asr-lora
```

2.3.2 直接下载

如未安装 git 工具，可以使用浏览器登陆 https://github.com/asrlora/alios-asr-lora，在 github 页面中直接下载 SDK 压缩包



2.4 联网准备（联网通信需要）

2.4.1 节点信息申请

节点设备的配置信息需要从服务器申请，通常 OTAA 设备需要 DEVEUI，APPEUI 和 APPKEY 等信息，ABP 设备需要 DEVADDR，NWKSKEY 和 APPSKEY 等信息。

2.4.2 网关配置

节点连接网关时，需要网关的信道配置信息，具体配置请咨询网关提供商，。

3 Demo 程序说明

3.1 Demo 简介

ASR6501/ASR6502 SDK 主要包含三个样例工程，位于 alios-asr-lora\projects\Creator\ASR6501 目录下：

1） alios_small 示例工程

默认支持 LinkWan，支持 ICA AT 指令集，可以使用 AT 指令快速地开发 LoRa 模块。

2） pingpong 示例工程

pingpong 工程来自 Semtech 的代码，主要为点对点通信提供示例；

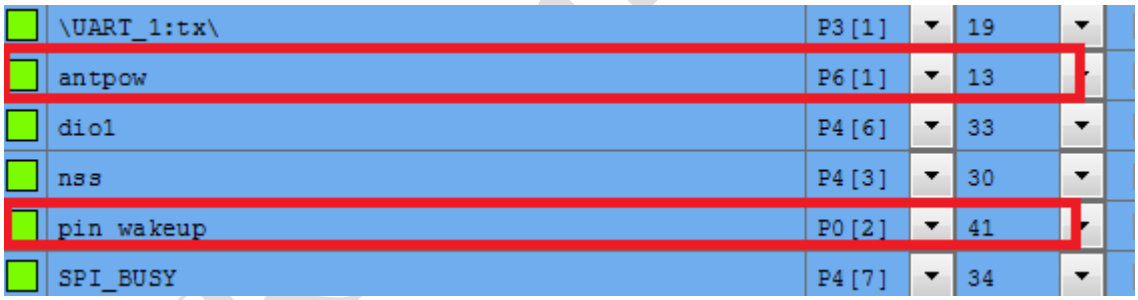
3） lorawan 示例工程

lorawan 工程来自 Semtech 的 classA 示例程序，主要为标准 LoRaWan 的通信提供示例，方便用 ASR6501/ASR6502 做主控 MCU 的客户开发。本文档主要基于此示例程序进行说明。

3.2 ASR6502 与 ASR6501 的区别

ASR6501 与 ASR6502 共用一份代码，主要差异在 antpow 与 pin_wakeup。

1） ASR6501 中这两个 Pin 脚的设置



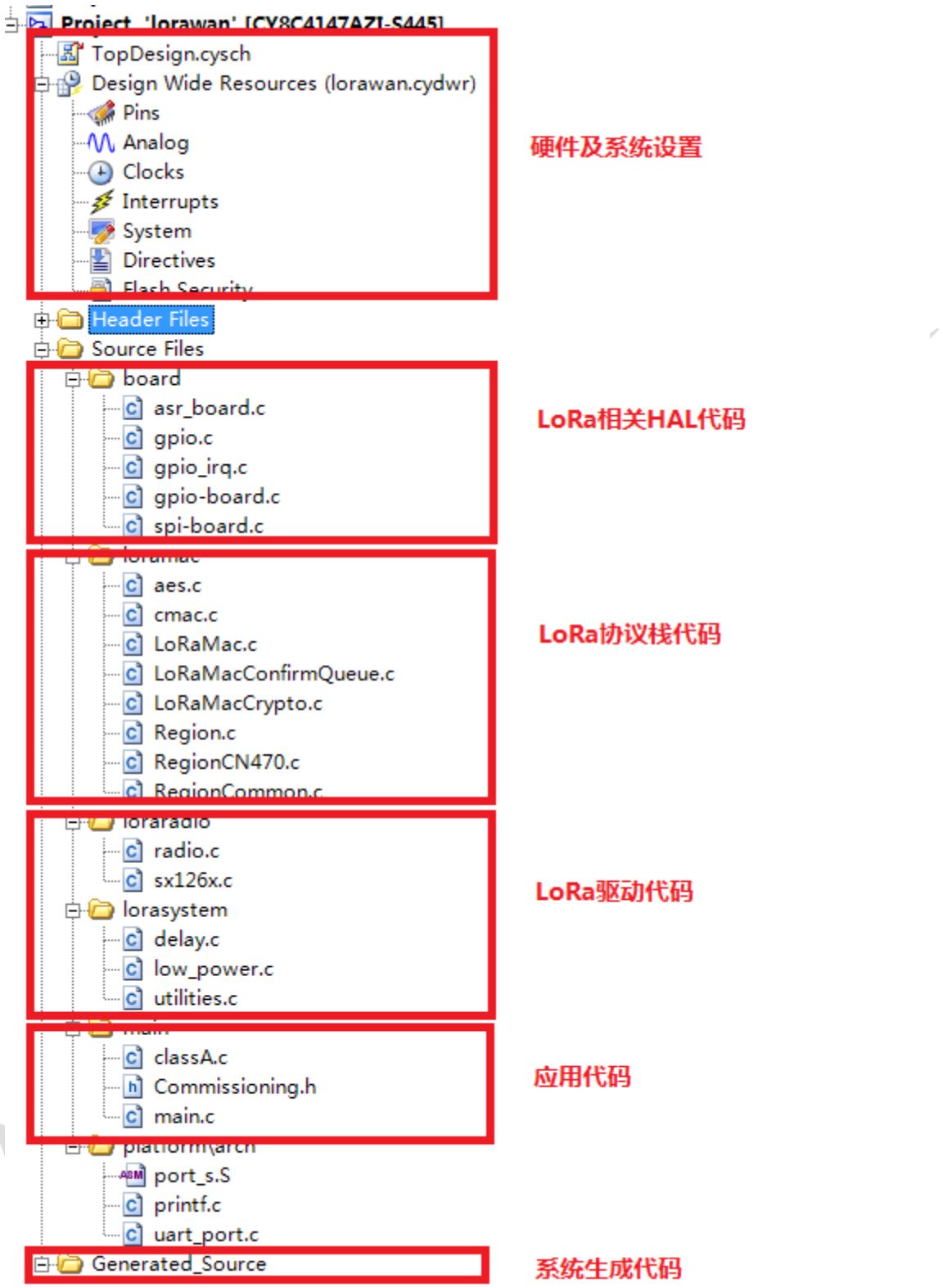
2） ASR6502 中这两个 pin 脚的配置

ASR6501 中的工程修改这两个 pin 的设置，即可以在 ASR6502 的板子上运行。

3.3 代码结构

如下图所示，LoRaWan 工程主要分为下面几个部分：硬件及系统配置，LoRa 相关 HAL 代码，LoRaWan 协议栈代码，LoRa 驱动代码，应用代码和系统生成代码。

其中硬件及系统配置部分是 Creator 特有的功能，可以通过拖拽控件的方法，非常简便直观地设置外设属性；编译后会把设置的属性生成到代码，放在 Generated_Source 里面； Board 目录下主要是 LoRa 相关 GPIO，SPI 等 HAL 部分代码；loramac 下面是 LoRaWan 协议栈相关的代码；loraradio 和 lorasystem 下面是 LoRa 驱动相关的代码；main 目录下是应用相关代码，用户可以参考 classA.c 中的逻辑，修改代码，添加自己的代码。


TopDesign.cysch
Design Wide Resources (lorawan.cydwr)
Pins
Analog
Clocks
Interrupts
System
Directives
Header Files
Source Files
board
asr_board.c
gpio.c
gpio_irq.c
gpio-board.c
spi-board.c
aes.c
cmac.c
LoRaMac.c
LoRaMacConfirmQueue.c
LoRaMacCrypto.c
Region.c
RegionCN470.c
radio.c
sx126x.c
lorasystem
delay.c
low_power.c
utilities.c
classA.c
Commissioning.h
main.c
port_s.S
printf.c
uart_port.c
Generated_Source
硬件及系统设置
LoRa相关HAL代码
LoRa协议栈代码
LoRa驱动代码
应用代码
系统生成代码

3.4 代码流程

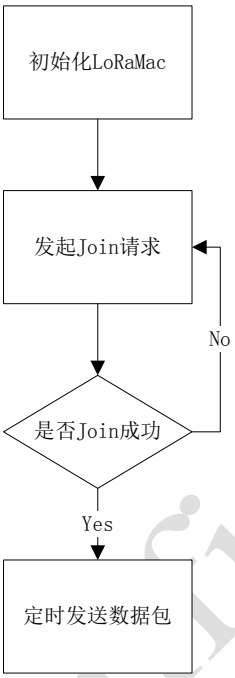
Demo 代码中的流程比较简单，主要流程如下：

1） 初始化 LoRaMac 协议栈；

2） 根据设置的三元组信息，发起 Join 请求；

3） Join 成功后，起定时器，定时发送 LoRa 数据包；

4） 如果 Join 不成功，则继续发送 Join 请求。



3.5 LoRaWan 配置与修改说明

3.5.1 三元组修改

Demo 程序中， 三元组信息可以在 projects\Creator\ASR6501\lorawan.cydsn\Commissioning.h 中进行修改。

量产时，可以考虑使用 chip_id 作为 DEVEUI，自定义算法生成 APPKEY 的方式，将三元组信息注册到服务器。也可以添加代码，将三元组信息烧录到 Flash。

Chip_id 可以通过 CyGetUniqueId 函数获取。

3.5.2 入网方式修改

Demo 程序中，入网方式可以在 projects\Creator\ASR6501\lorawan.cydsn\Commissioning.h 中进行修改。OVER_THE_AIR_ACTIVATION 为 1，使用 OTAA 的入网方式。OVER_THE_AIR_ACTIVATION 为 0，则使用 ABP 的入网方式。

3.5.3 ClassC 修改

Demo 程序中默认是 Class A 类型，如需修改为 Class C 类型，请在 LoRaMacInitialization 之后，调用 LoRaMacMibSetRequestConfirm 函数进行修改。 示例如下：

```
MibRequestConfirm_t mibReq;
mibReq.Type = MIB_DEVICE_CLASS;
mibReq.Param.Class = CLASS_C;
LoRaMacMibSetRequestConfirm(&mibReq);
```

3.5.4 频道掩码修改

Demo 程序中默认的频道设置是 0-7，用户可以调用 LoRaMacMibSetRequestConfirm 函数进行修改，具体可参考 lwan_dev_params_update 函数。

3.5.5 Mac 参数修改

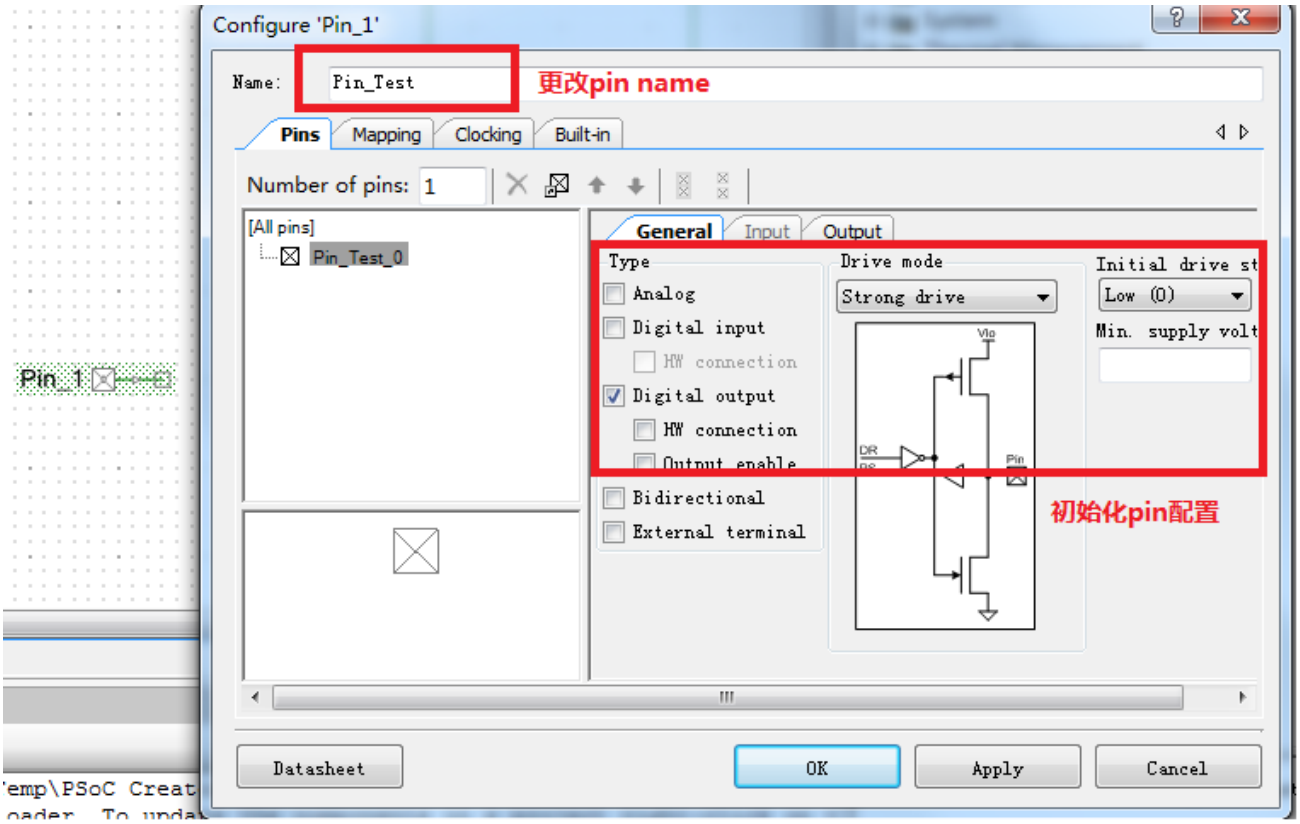
其他 Mac 参数的修改，可以直接调用 LoRaMacMibSetRequestConfirm 函数。

3.6 外设使用说明

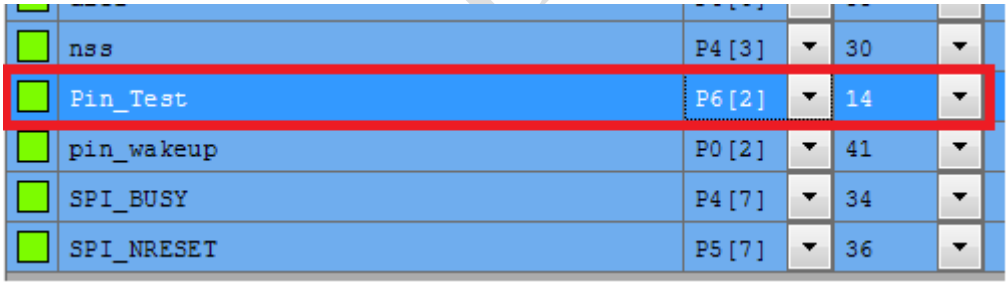
3.6.6 GPIO

添加 GPIO

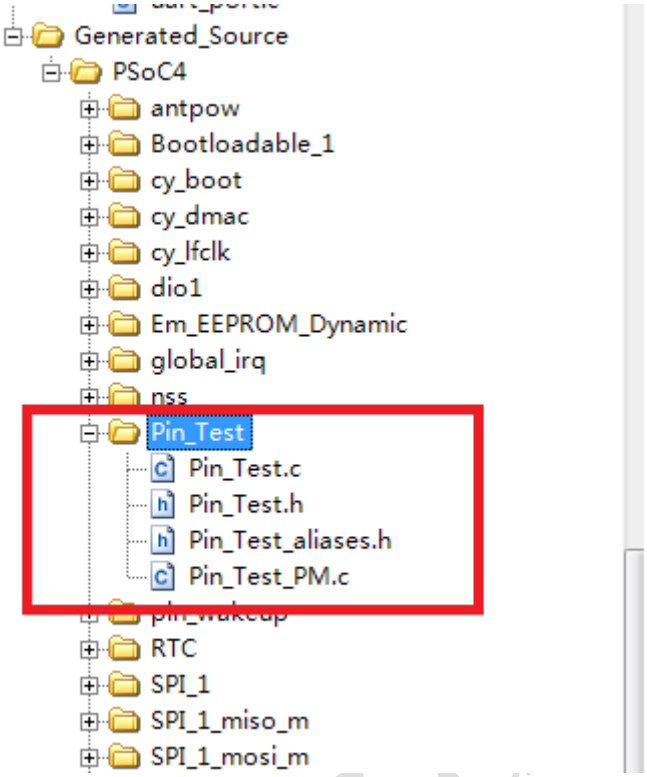
首先在打开工程左侧 TopDesign.cysch 界面，找到界面右侧的”Ports and Pins”，点开并拖拽 pin 组件到 TopDesign 界面，然后双击 pin 组件，修改 pin 的配置。



在工程左侧 Pins 界面找到该 pin，并修改 pin 脚设置



完成上面设置后，编译即可生成改 GPIO 的 API 接口。



GPIO 读

继续上面例子，GPIO 读的接口如下：

uint8 Pin_Test_Read(void);

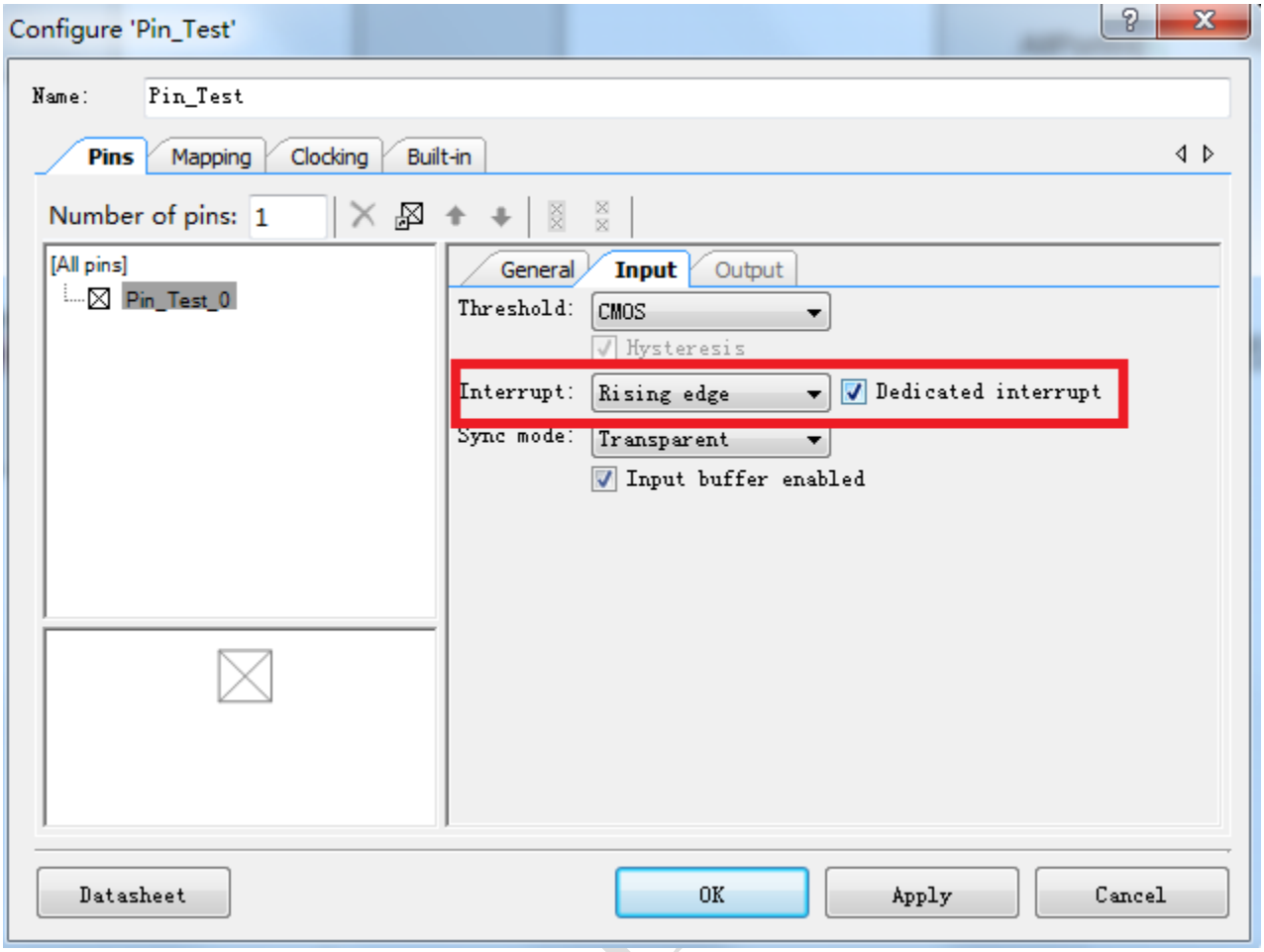
GPIO 写

GPIO 写的接口如下：

void Pin_Test_Write(uint8 value);

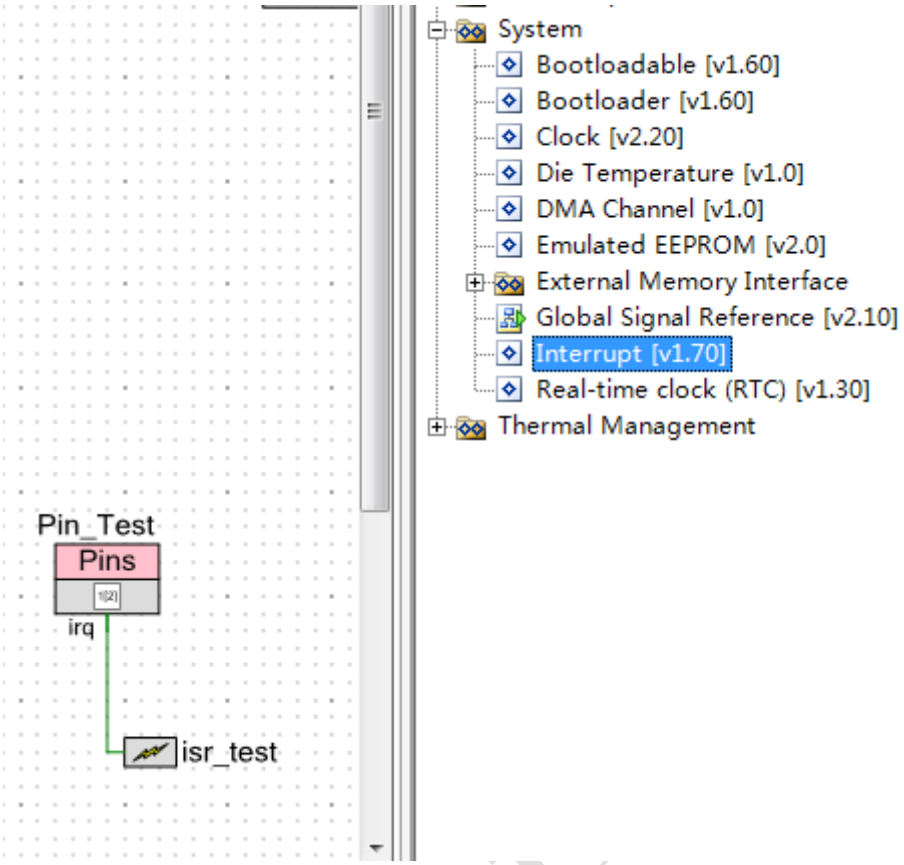
GPIO 中断

1) 如需使用 GPIO 中断，首先需要在 TopDesign 界面，将 GPIO 设置为 input，并初始化默认值，然后在下图 Input 界面设置中断类型。

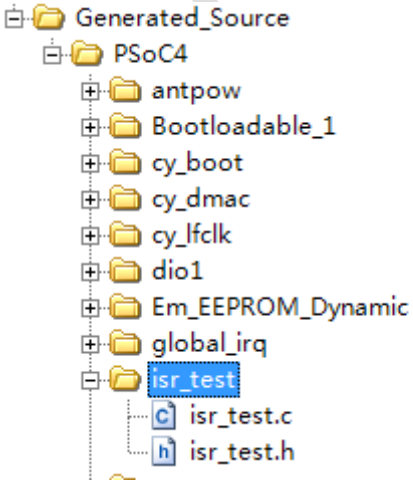


2) P0.X-P3.X 可以使用专有中断

a) 在 System 界面拖进一个 interrupt 与 Pin 关联



b) 然后编译代码即可生成中断相关接口



c) 使用 API 设置中断调用

```
CY_ISR_PROTO(GPIO_ISR);
CY_ISR(GPIO_ISR)
{
    Pin_Test_ClearInterrupt();
    /*ToDo*/
}
```

```
isr_test_StartEx(GPIO_ISR);
```

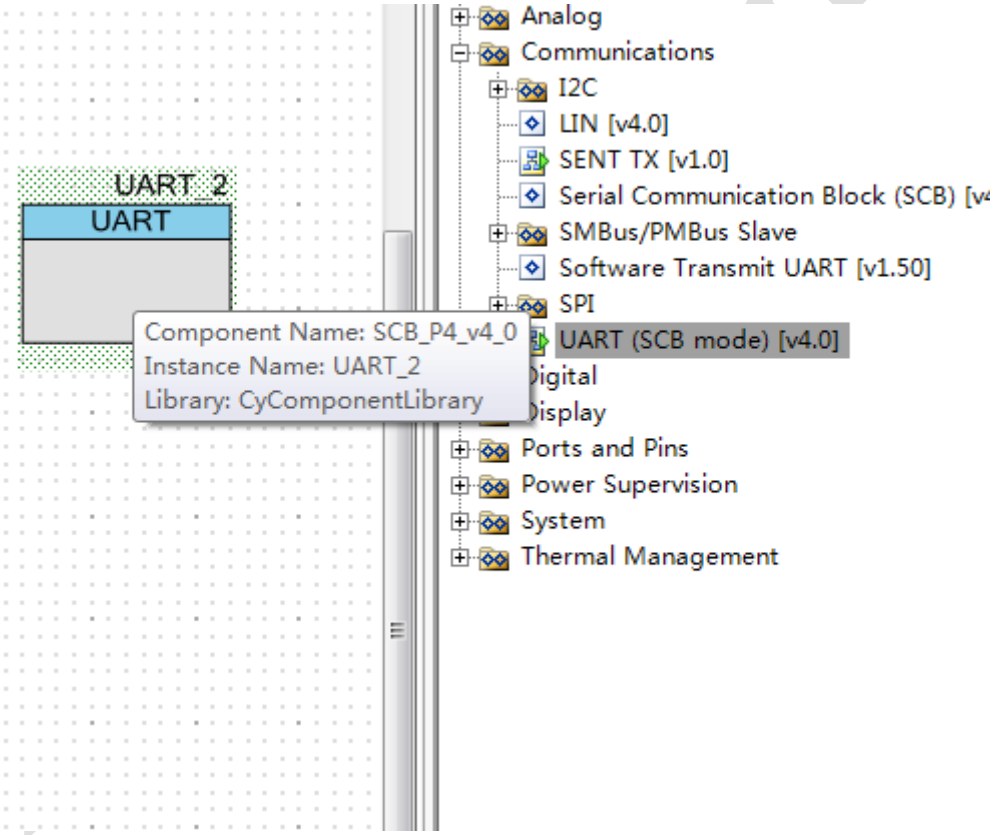
d) 注意：如果中断优先级比 GlobalSignal 低，中断会首先被 GlobalSignal 截获，所以需要在 Global_irq.c 的 GpioIsrEntry 函数中重新设置中断。

3) P4.X 及以上需要使用 GlobalSignal，在 Global_irq.c 的 GpioIsrEntry 函数中直接调用中断回调函数，具体可以参考 Demo 程序中 dio1 的设置。

3.6.7 UART

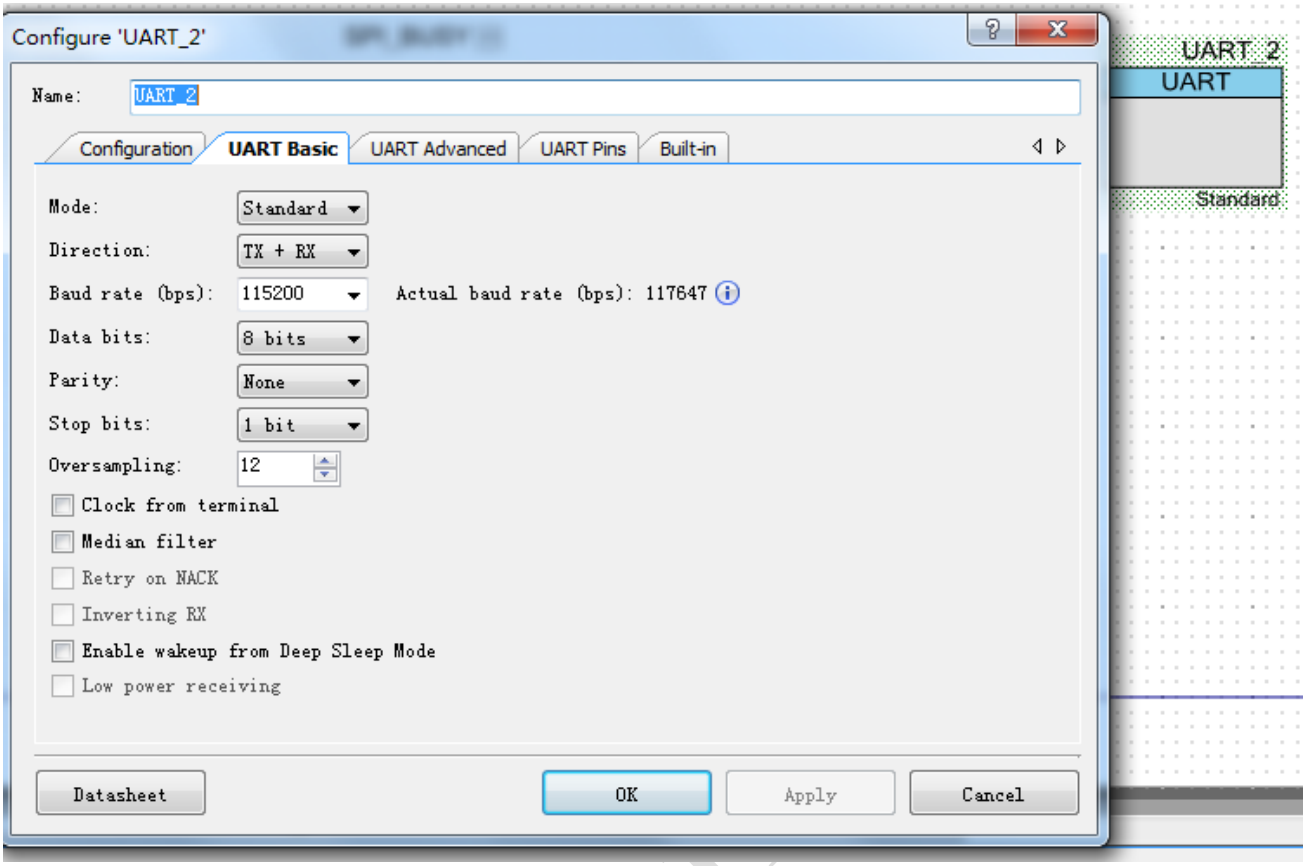
UART 添加

在 TopDesign 的 Communications 菜单中，拖拽 UART 组件。

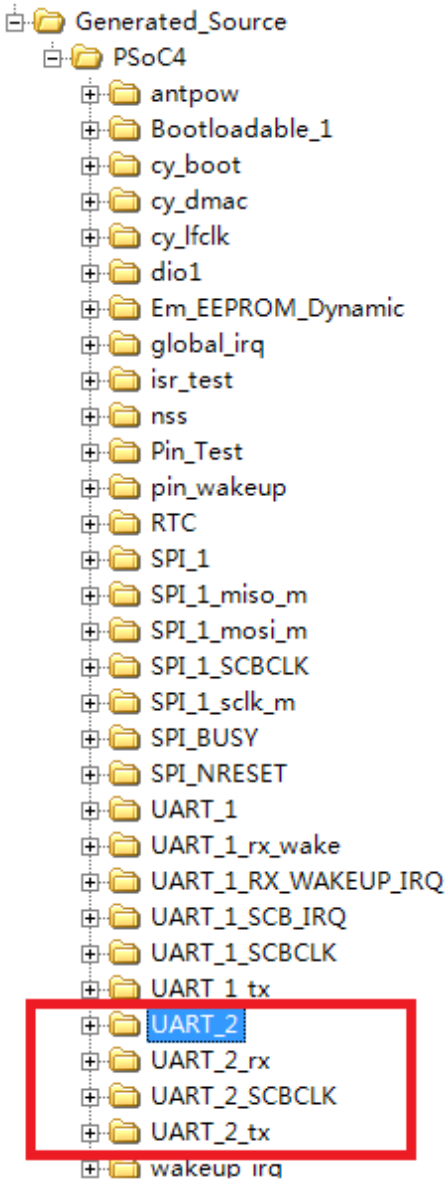


UART 参数设置

双击 UART 组件，可以配置 UART 参数。

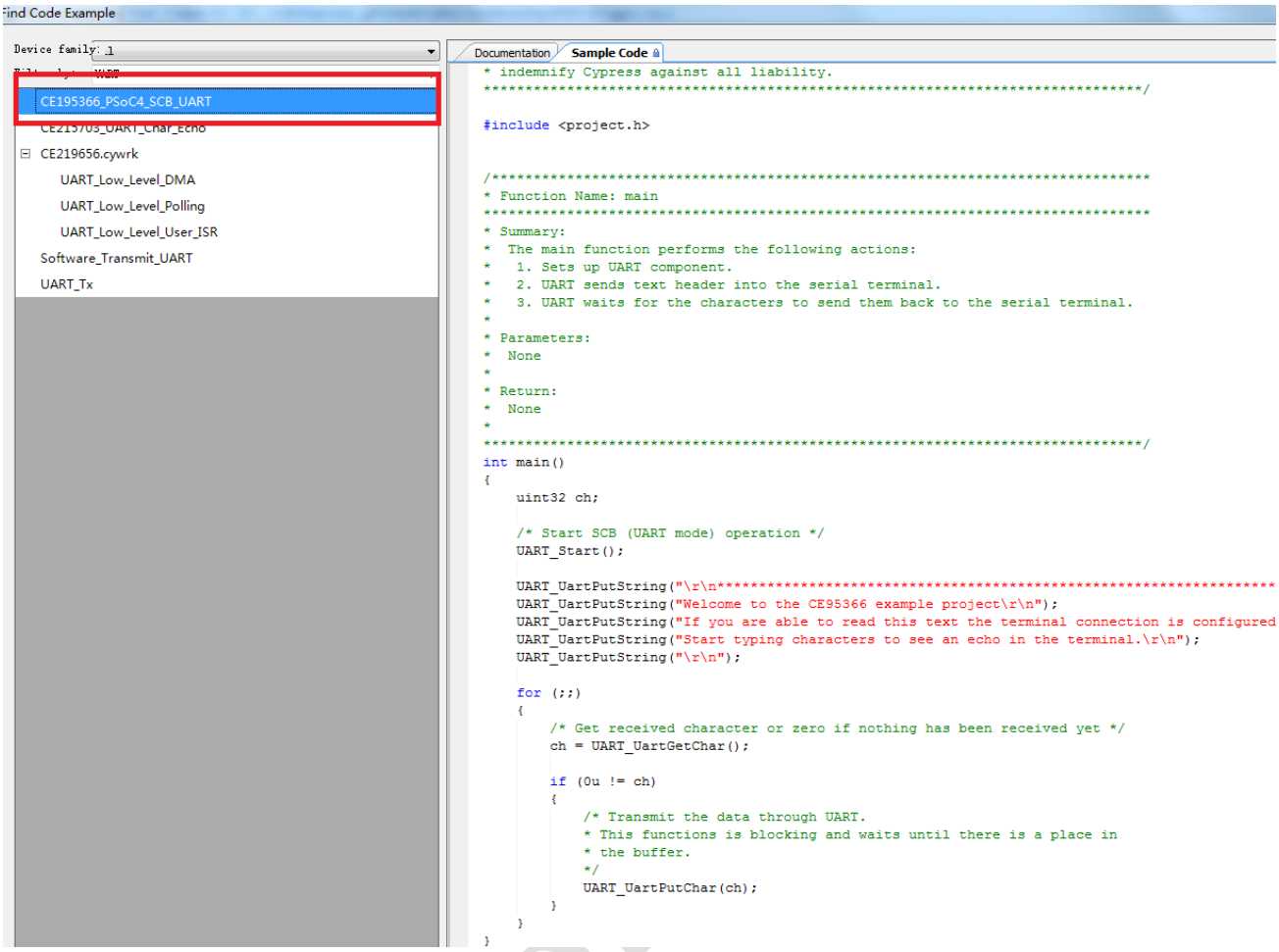


在 Pins 界面设置 pin 脚，编译后生成 UART 组件代码。



UART 使用

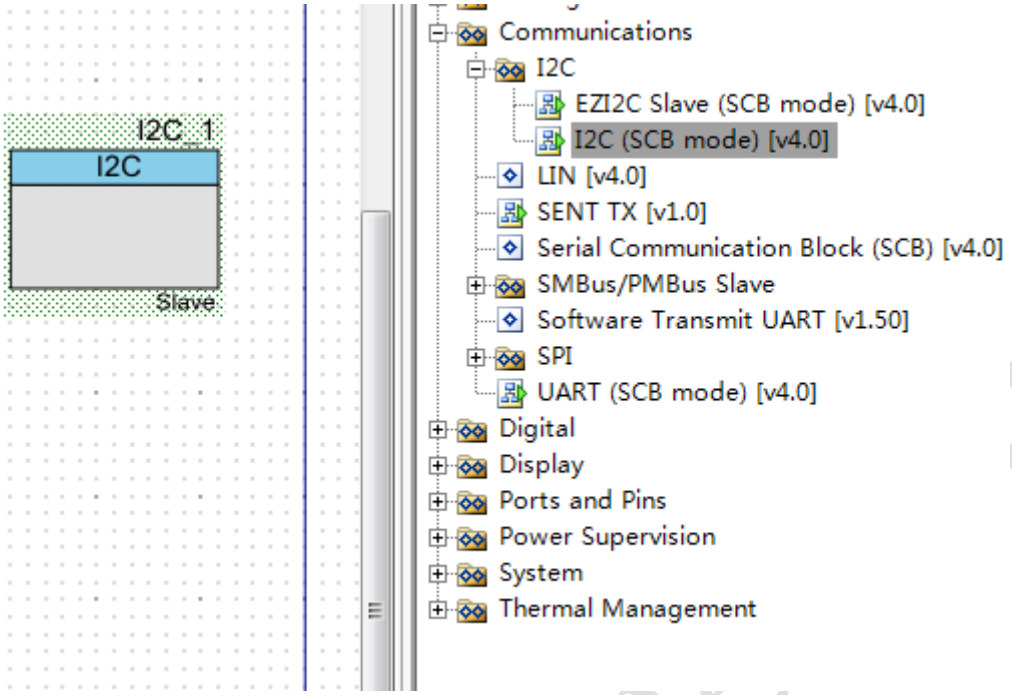
UART 的使用可参考 File->Code Example 中的 CE195366_PSoC4_SCB_UART。



3.6.8 **I2C**

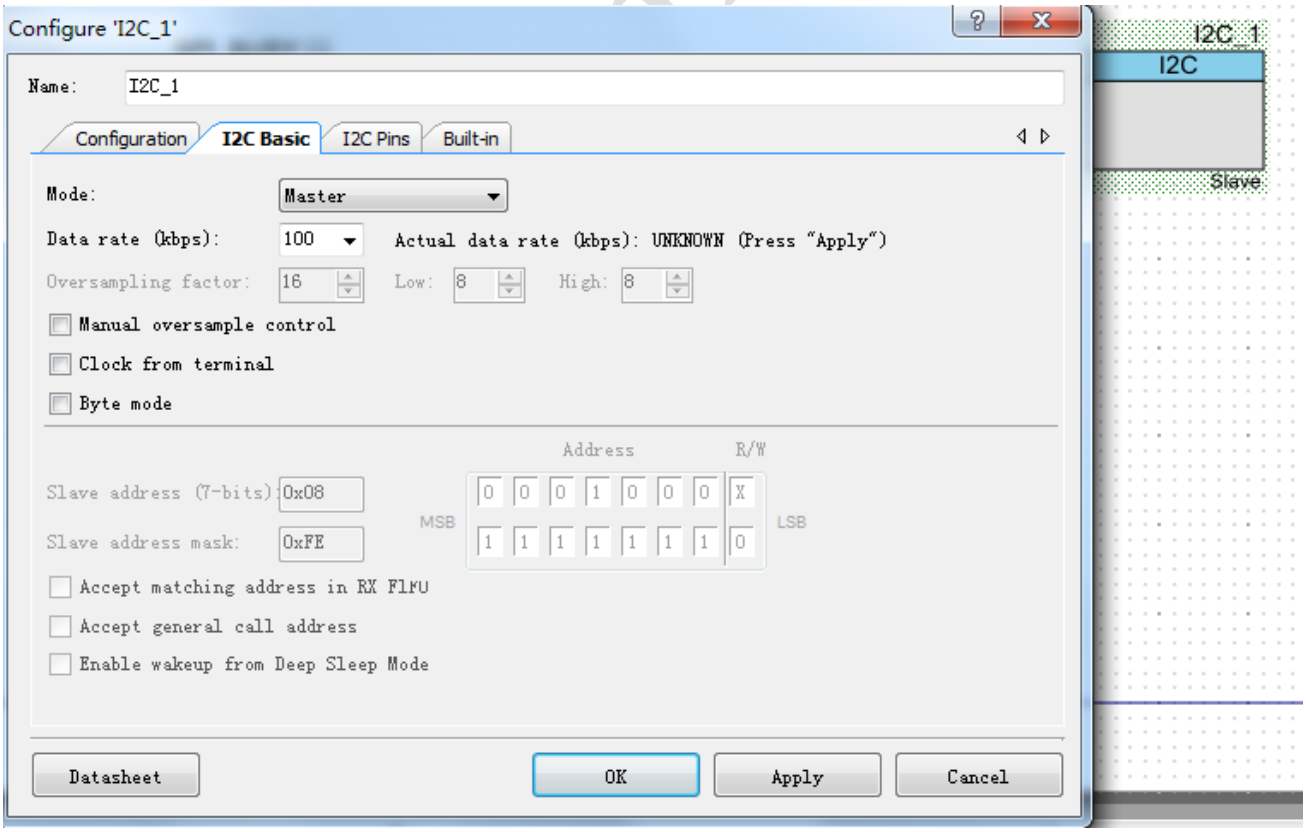
I2C 添加

在 TopDesign 的 Communications 菜单中，拖拽 I2C 组件。



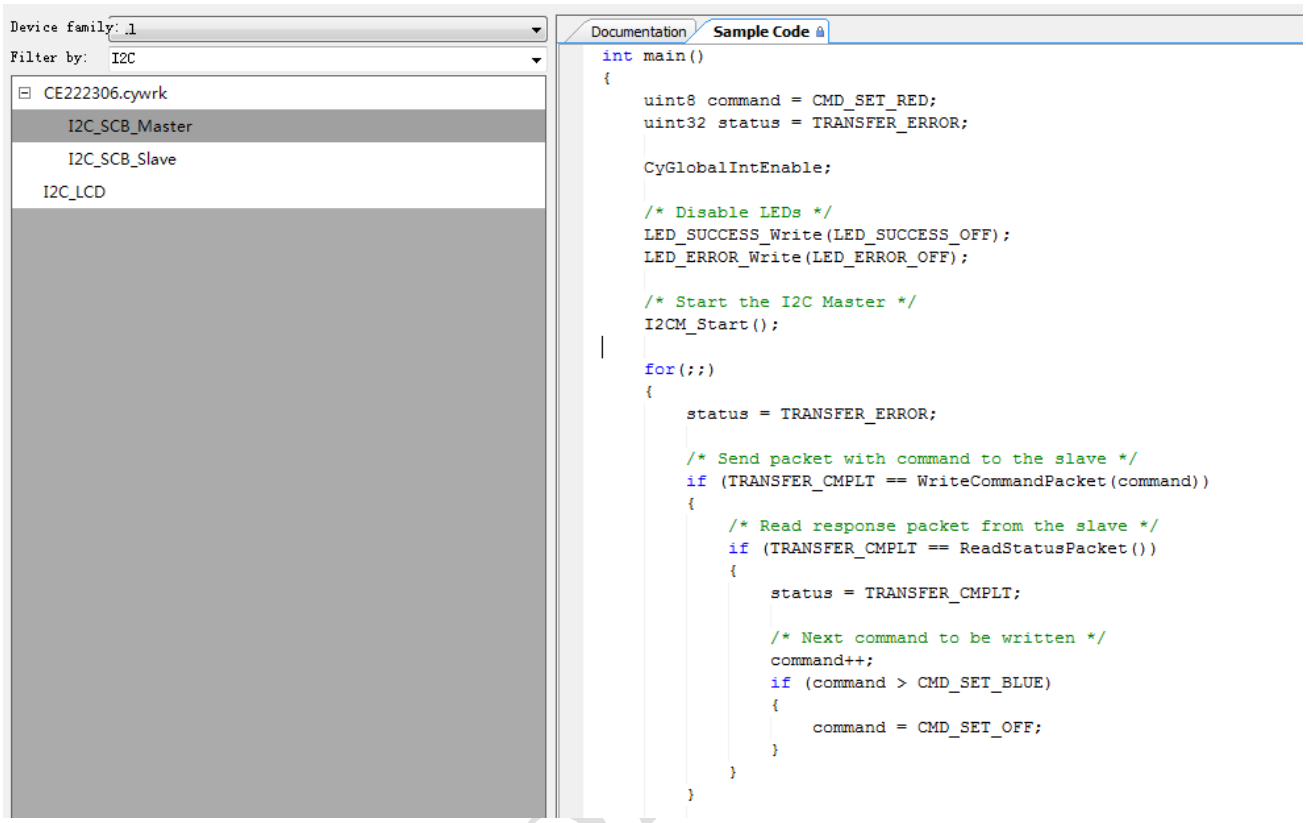
I2C 参数配置

双击 I2C 组件，进行参数设置，然后在 Pins 界面设置 pin 脚，编译生成代码。



I2C 使用

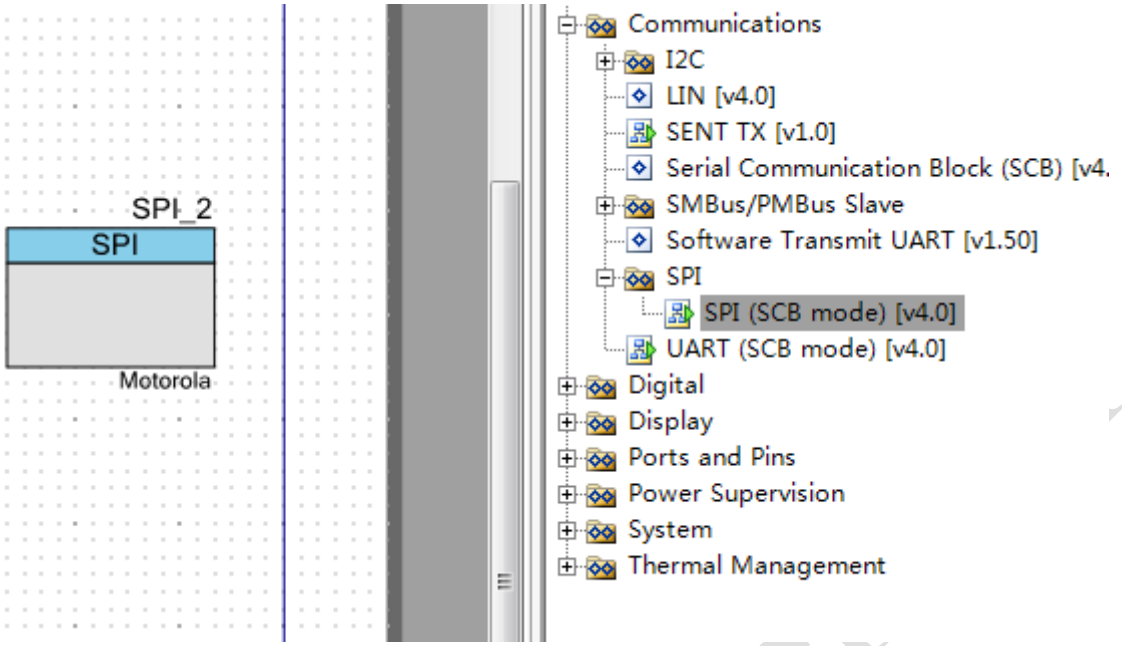
UART 的使用可参考 File->Code Example 中的 I2C_SCB_Master 和 I2C_SCB_Slave。



3.6.9 SPI

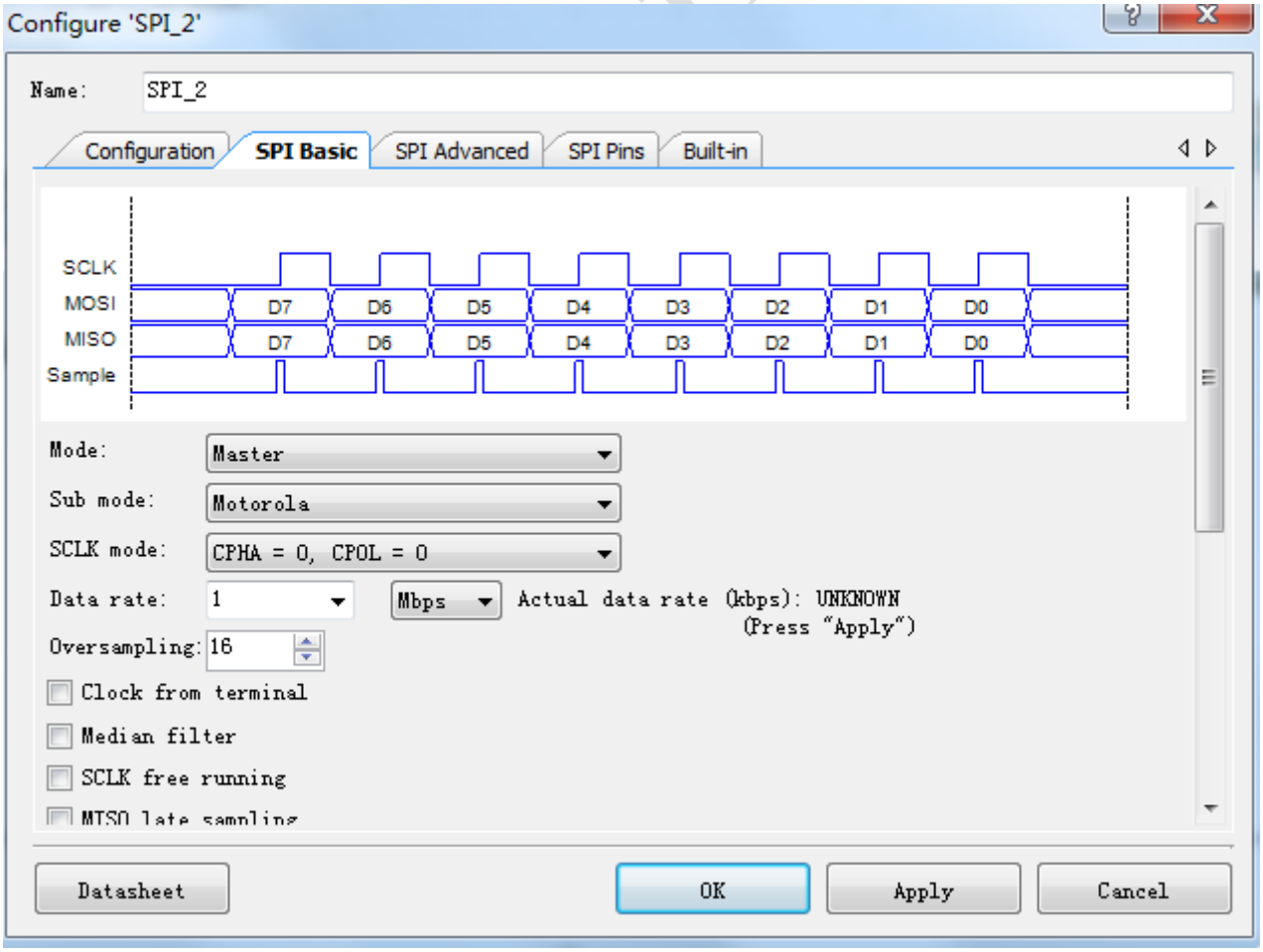
SPI 添加

在 TopDesign 的 Communications 菜单中，拖拽 SPI 组件。



SPI 参数设置

双击 SPI 组件，进行参数设置，然后在 Pins 界面设置 pin 脚，编译生成代码。



SPI 的使用

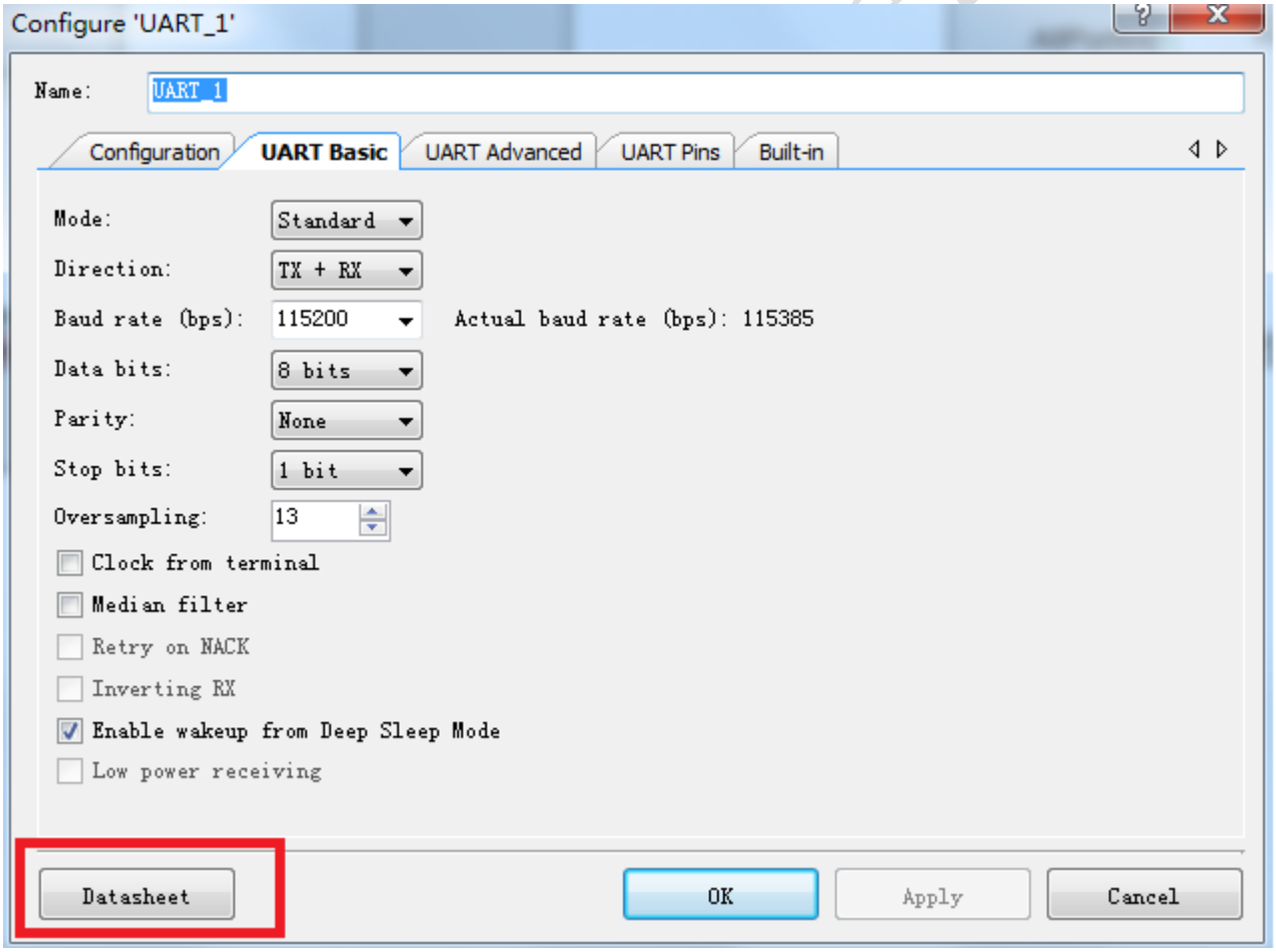
SPI 的使用可以参考 Demo 程序中的 SPI_1。

3.6.10 Timer

Timer 的使用不需要组件，可以直接调用 kernel\protocols\lorawan\lora\system\timeServer.h 中的接口。具体可以参考 classA.c 中的 TxNextPacketTimer

3.6.11 外设组件文档说明

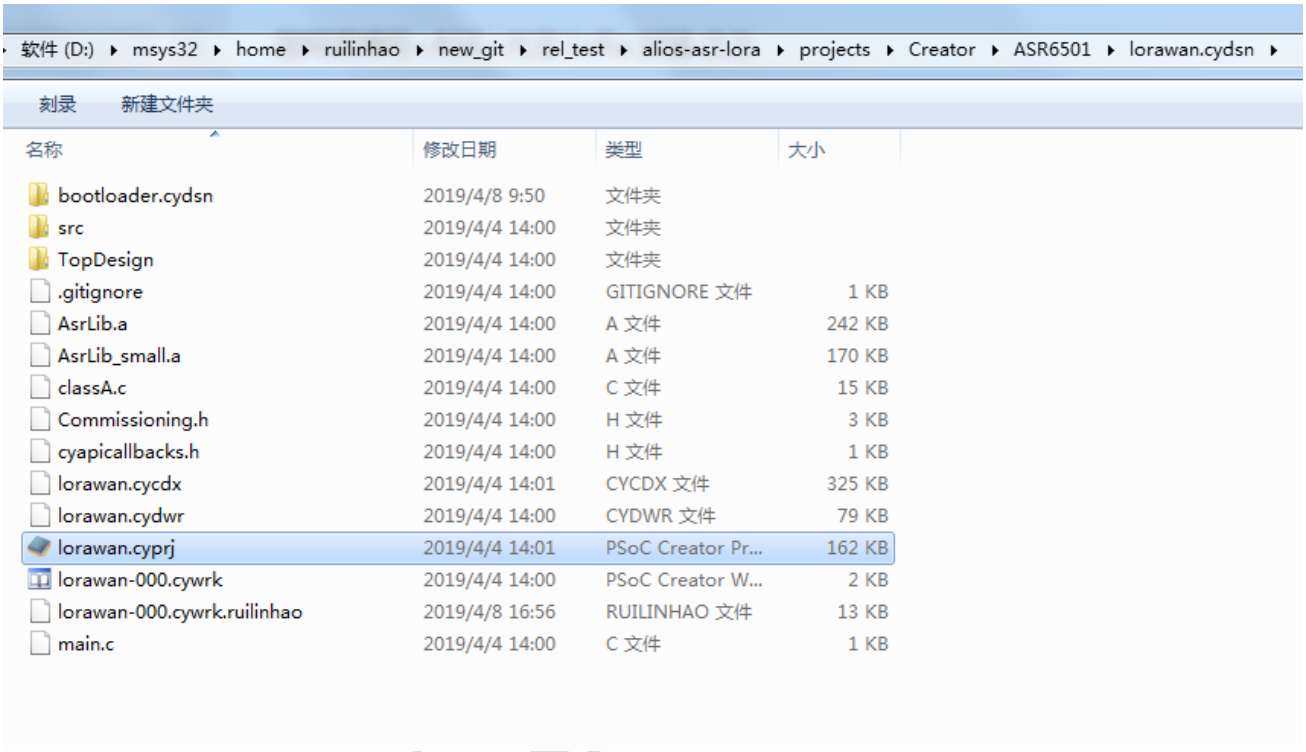
所有外设组件，均可双击组件进行参数配置，在配置页面的左下角均有组件的 DataSheet 文档可供参考，如下图



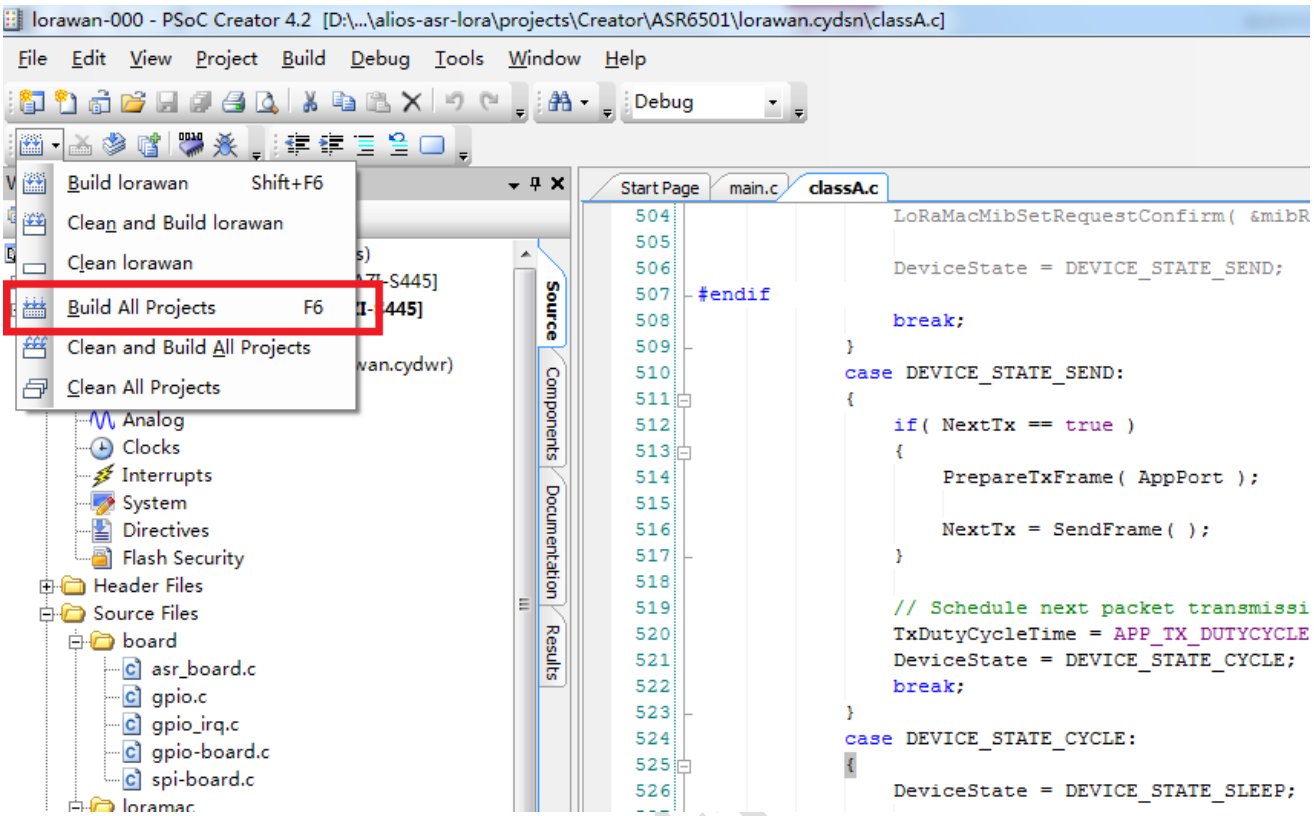
4 软件编译与烧录

4.1 编译

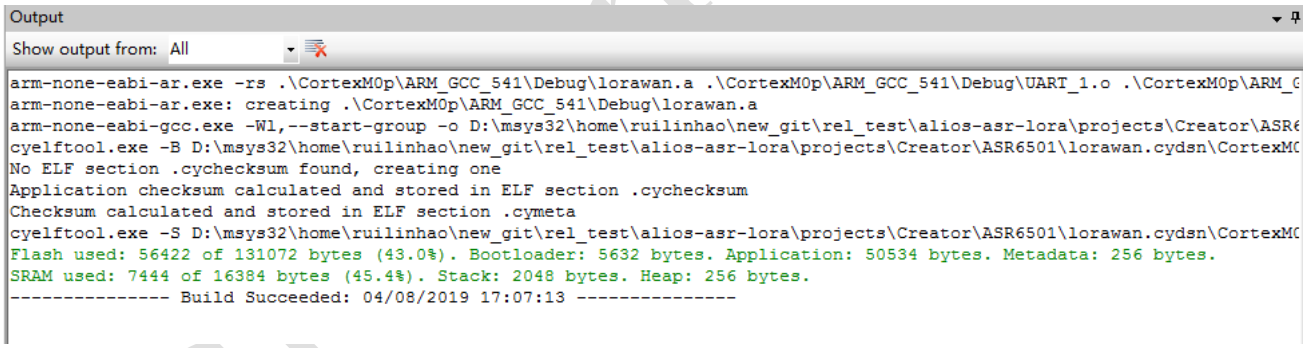
1) 在工程目录下打开后缀为.cyprj 的工程文件



2) 如下图所示，点击编译图标的下拉菜单，选择”Build All Projects”



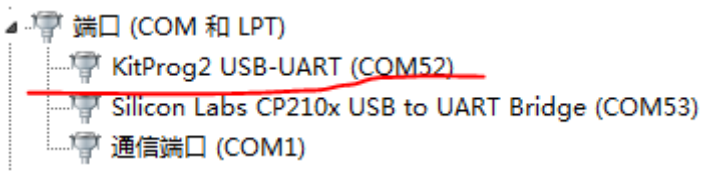
3) 编译完成后会在 Output 框显示编译结果：



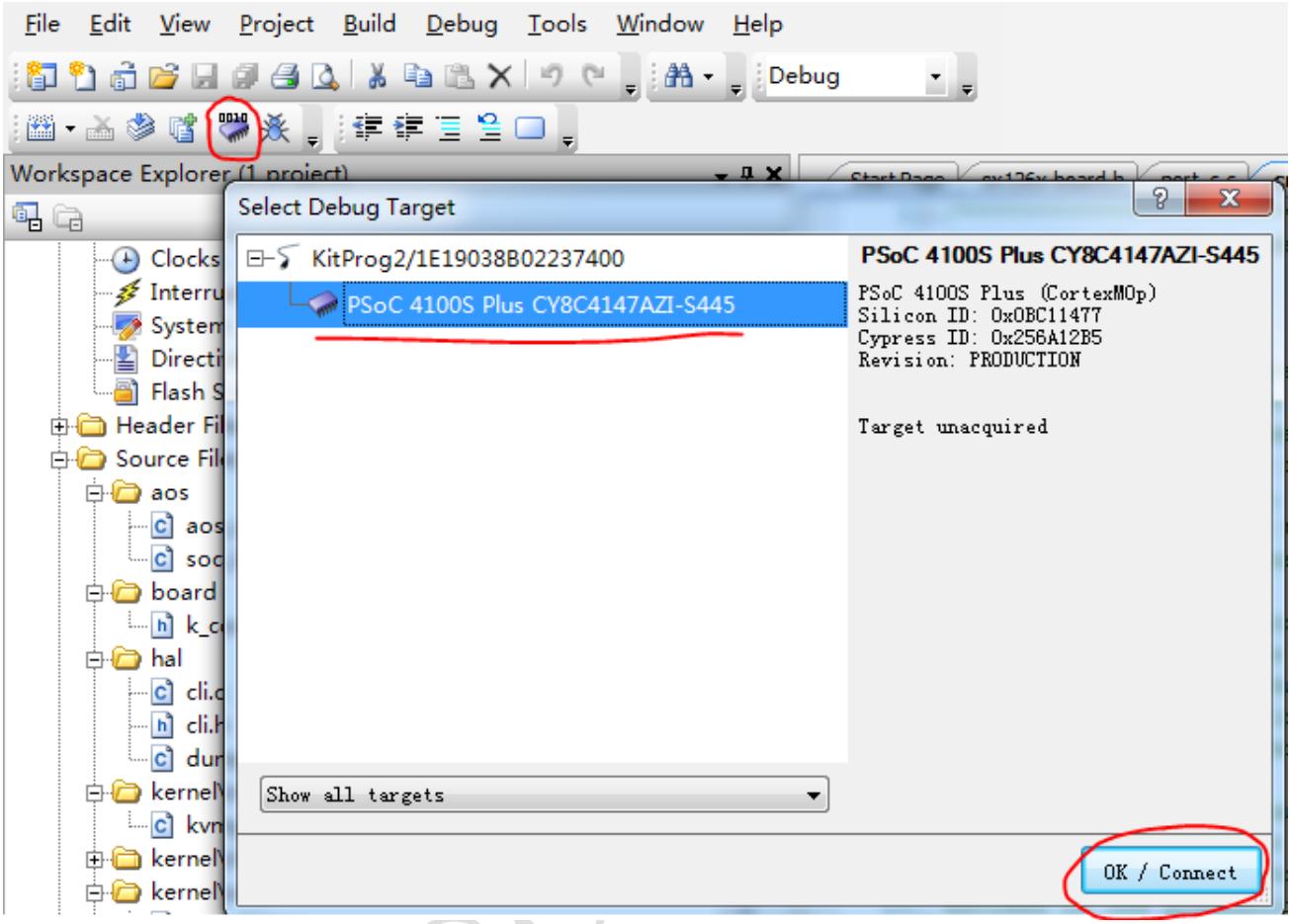
4.2 烧录

4.2.1 PSoC Creator 烧录

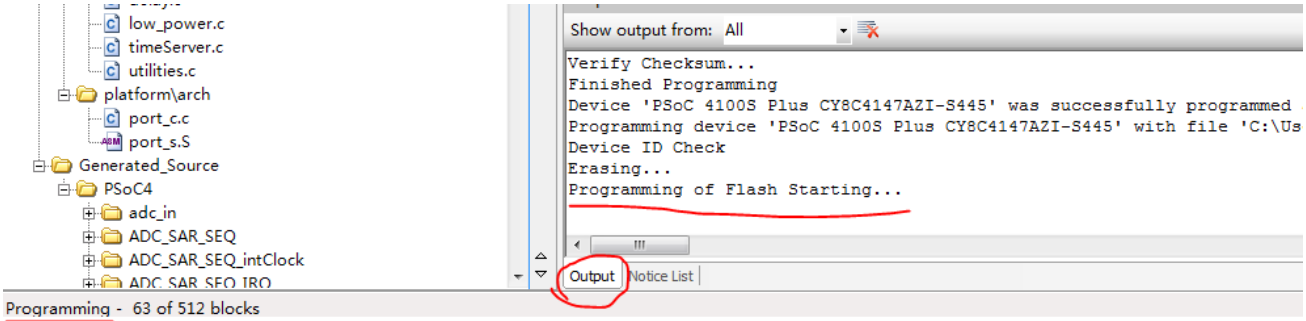
1） 连接 MicroUSB Cable 到 ASR6501/ASR6502 LoRa 母板的 USB_DL 口，并连接 PC，等待驱动安装完成，在 PC 上回出现 KitProg 设备。



2） 点击烧录按钮会弹出来如下界面，选中设备后点击“OK/Connect”按钮。



3） 在 PSoC Creator 底栏 Output 处会显示烧录的过程



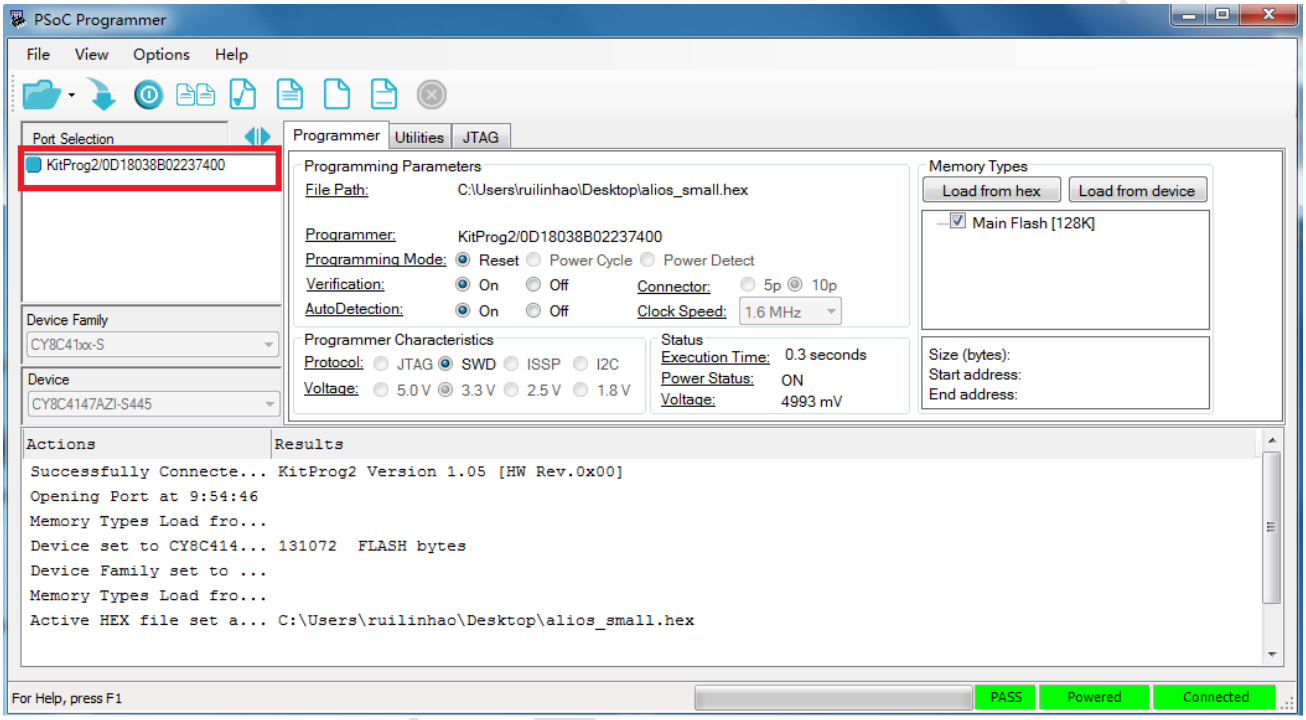
4） 烧录完成后在 Output 框显示：

```
Programming device 'PSoC 4100S Plus CY8C4147AZI-S445' with file 'D:\msys32\home\ruilinhao\new_git\rel_test\
Device ID Check
Erasing...
Programming of Flash Starting...
Protecting...
Verify Checksum...
Finished Programming
Device 'PSoC 4100S Plus CY8C4147AZI-S445' was successfully programmed at 03/27/2019 09:51:34.
```

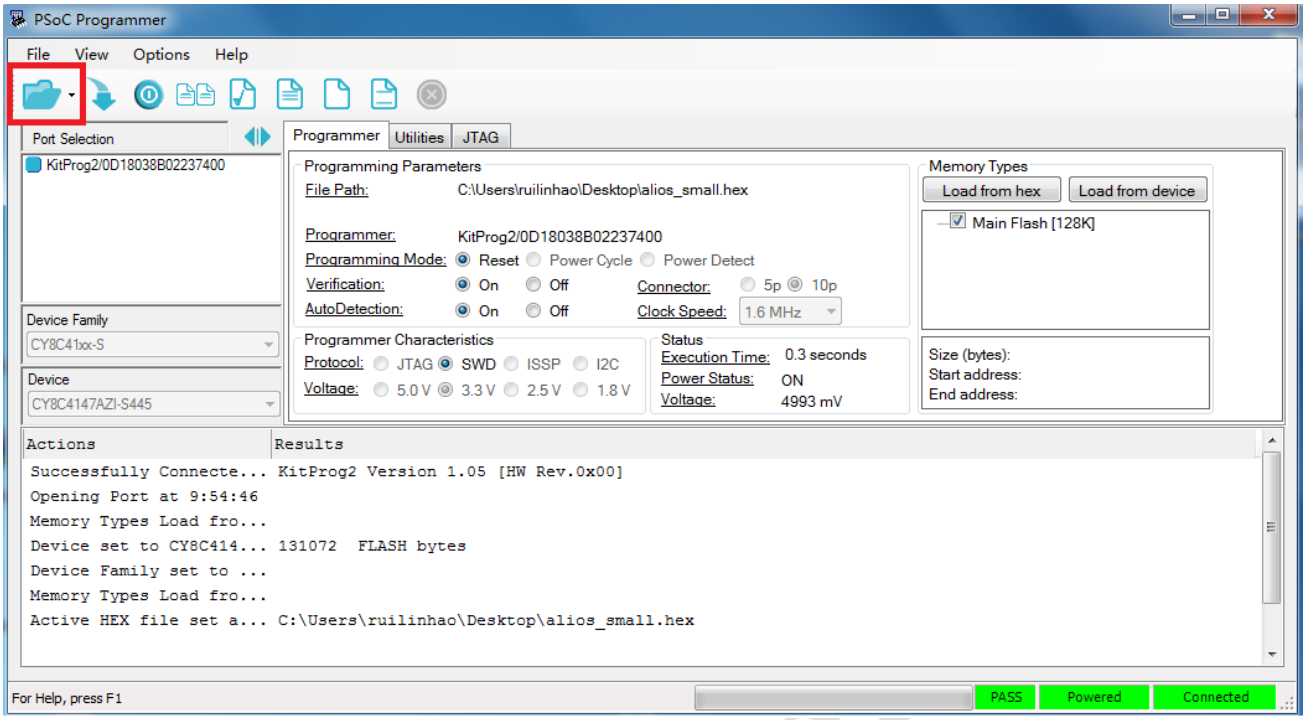
4.2.2 PSoC Programmer 烧录

用户也可以使用 PSoC Programmer，烧录 PSoC Creator 生成的 hex，步骤如下：

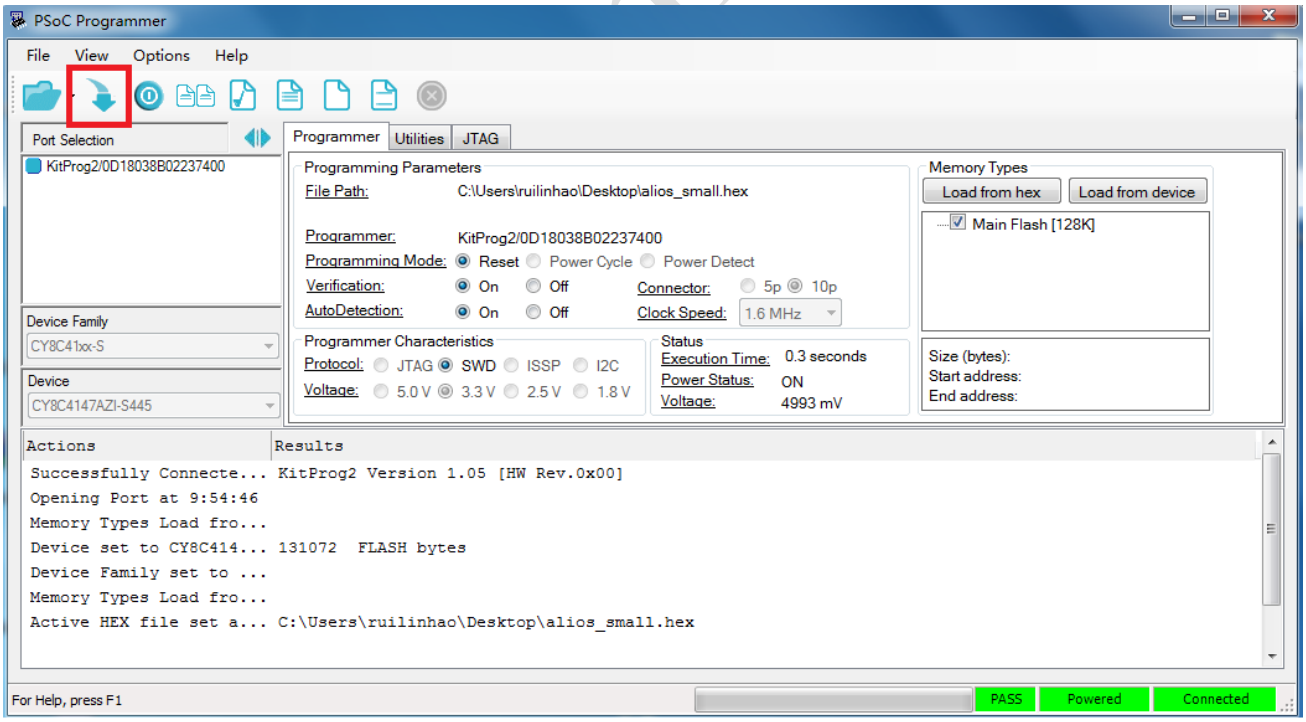
1） 分别连接 MicroUSB Cable 到 ASR6501/ASR6502 LoRa 母板的 USB_DL 口，并连接 PC，等待驱动安装完成， PSoC Programmer 工具端出现待烧录设备：



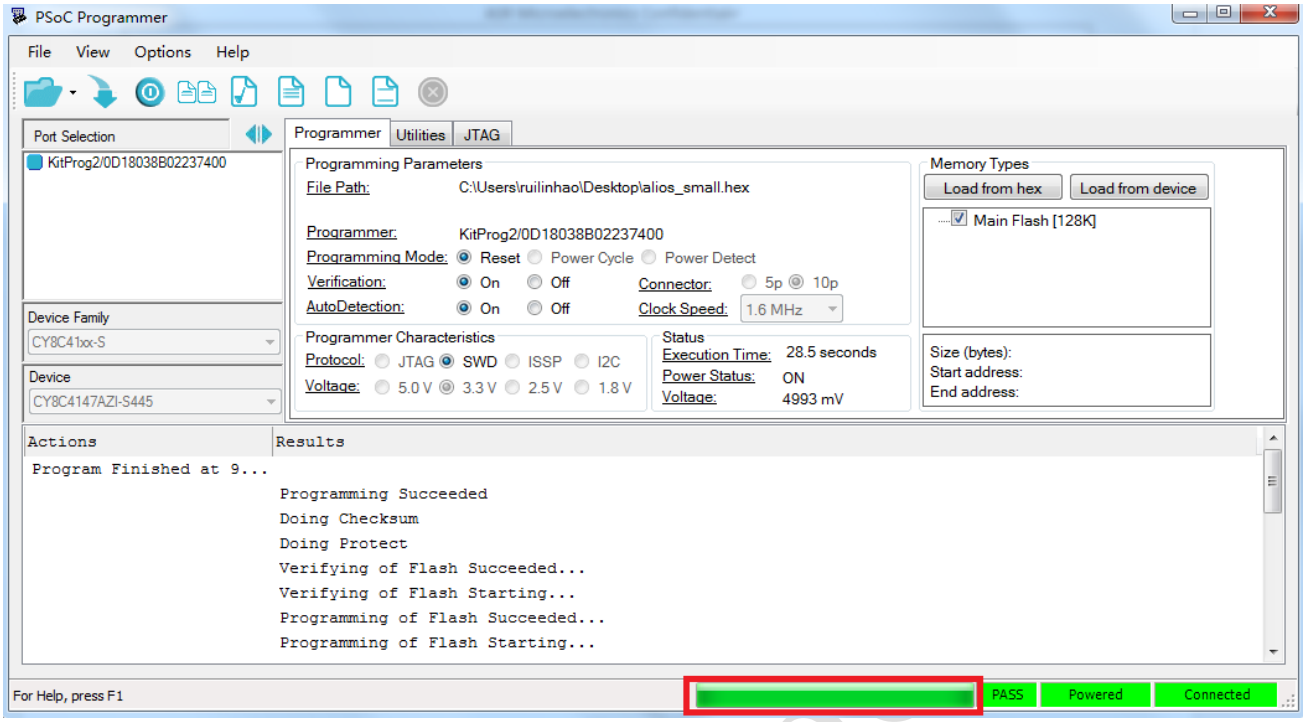
2） 选择下面打开文件的按钮，选择要烧录的 hex 文件，如：projects\Creator\ASR6501\lorawan.cydsn\CortexM0p\ARM_GCC_541\Debug\lorawan.hex



3） 然后选择 Program 按钮，等待烧录完成



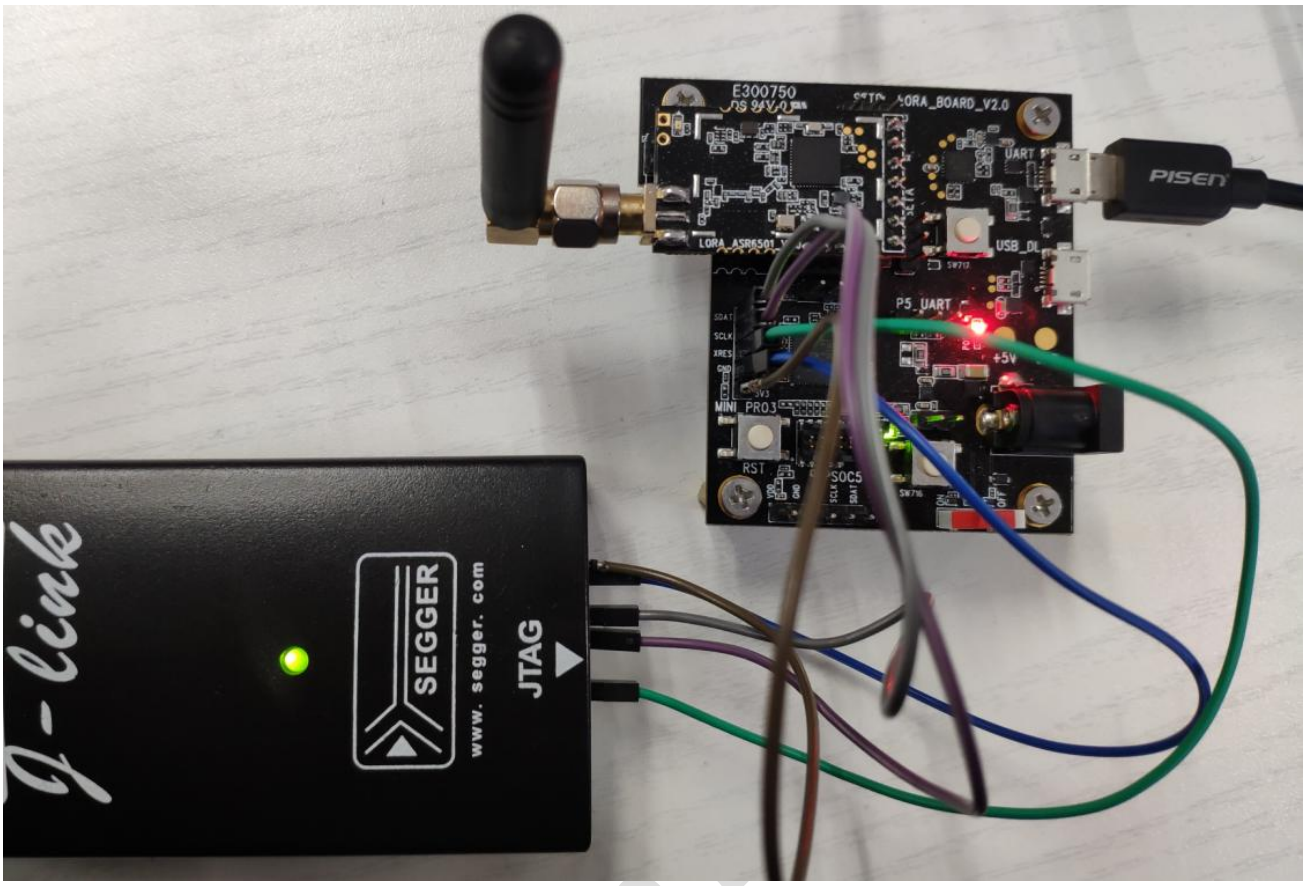
4） 烧录完成



4.2.3 J-Flash 烧录

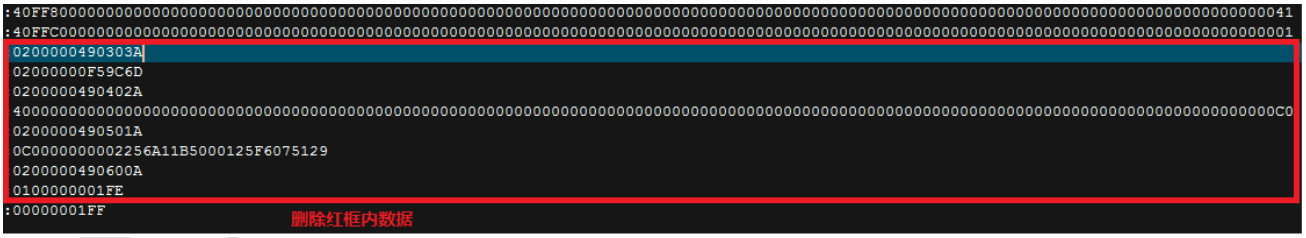
1） 硬件连接

使用 J-Link 烧录时，需要将 SDAT，SCLK，XRES，VCC，GND 等 5 根线对应连好，如使用自制母板，也请将对应的线接好。

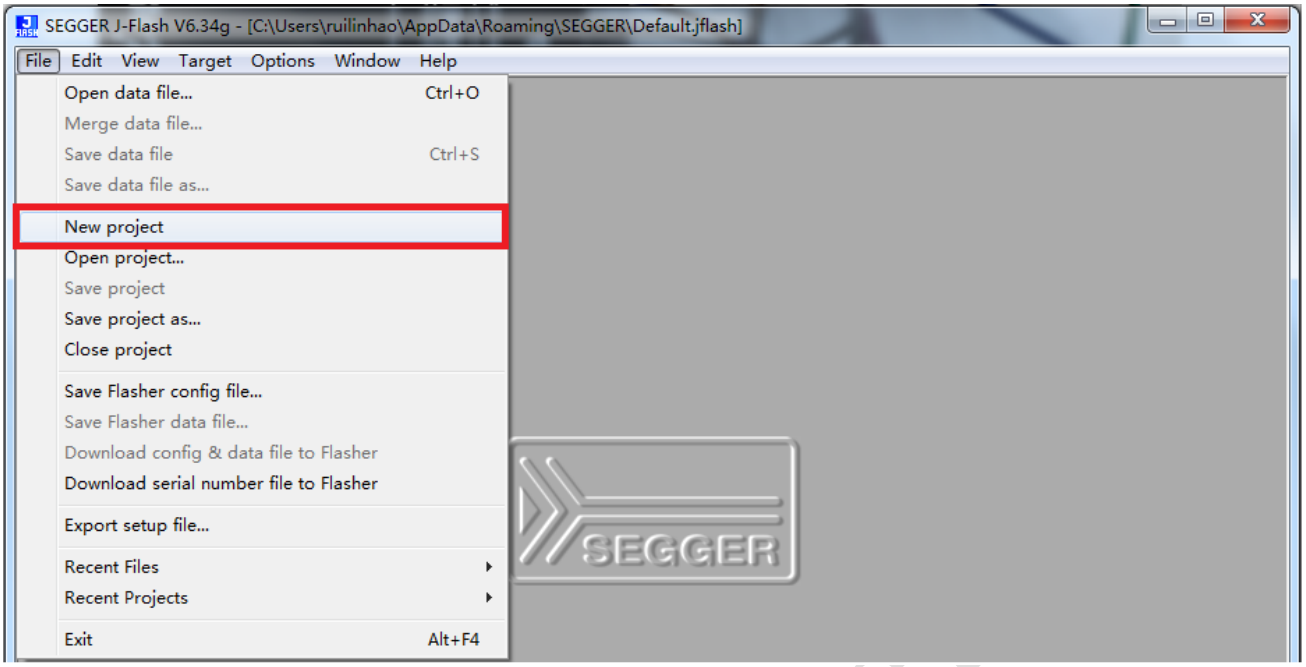


2） 修改 Hex 文件

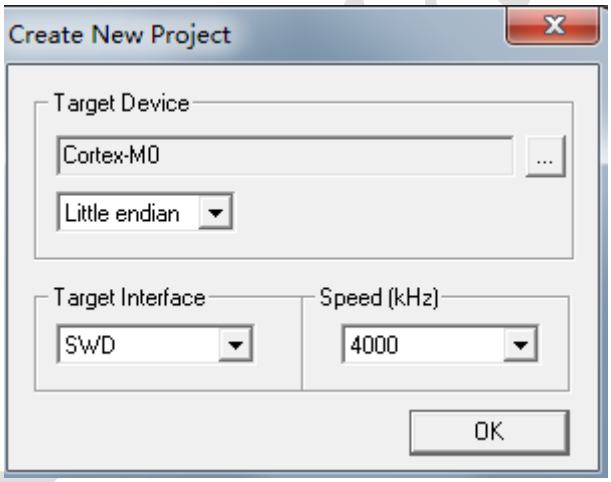
PSOC Creator 生成的 Hex 文件因为有 0x90300000,0x90400000, 0x90500000, 0x90600000 等非 flash 数据（此部分主要是 device protection 数据），所以无法直接使用 J-Flash 进行烧录，需要先将 Hex 末尾的这部分数据删除。

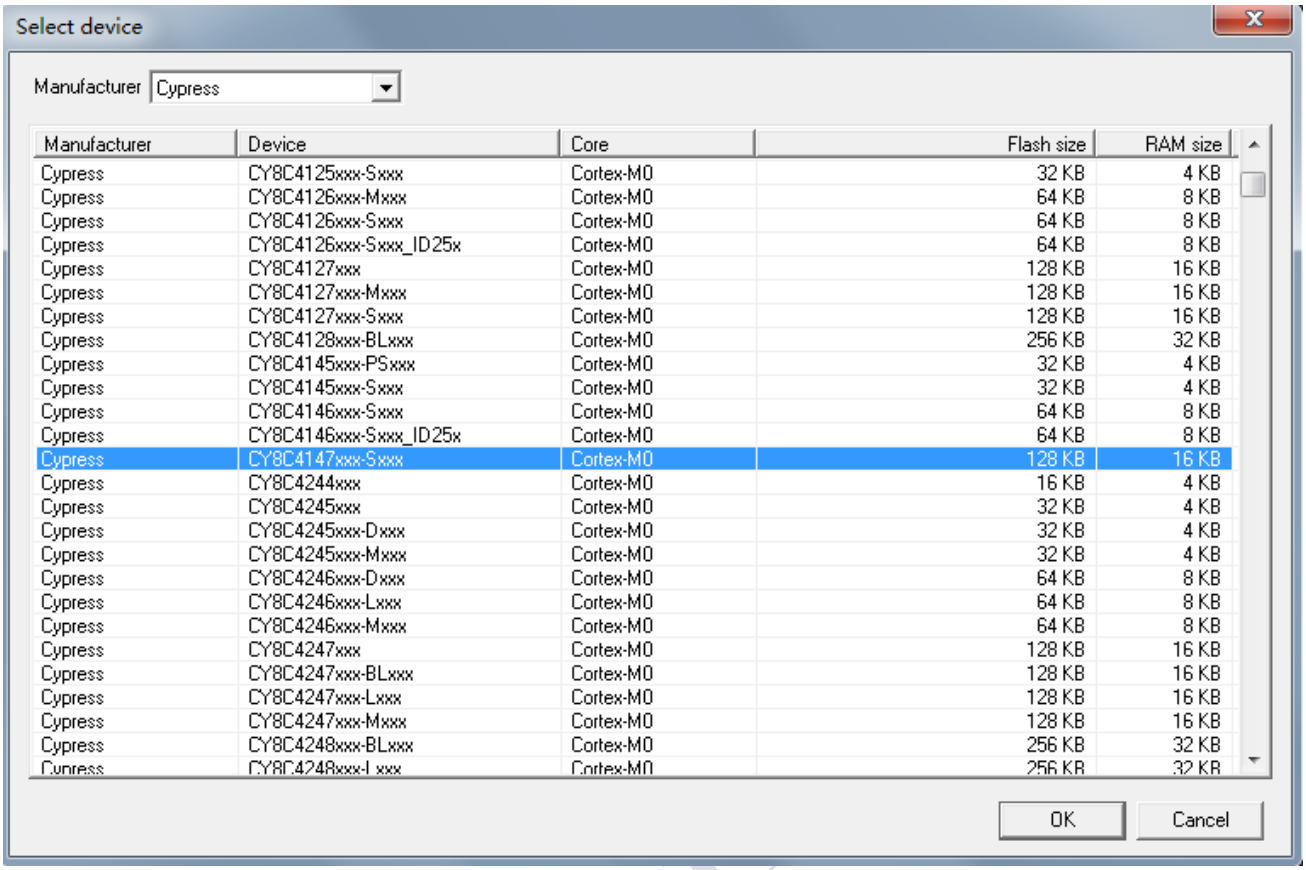


3） 启动 J-flash，新建工程

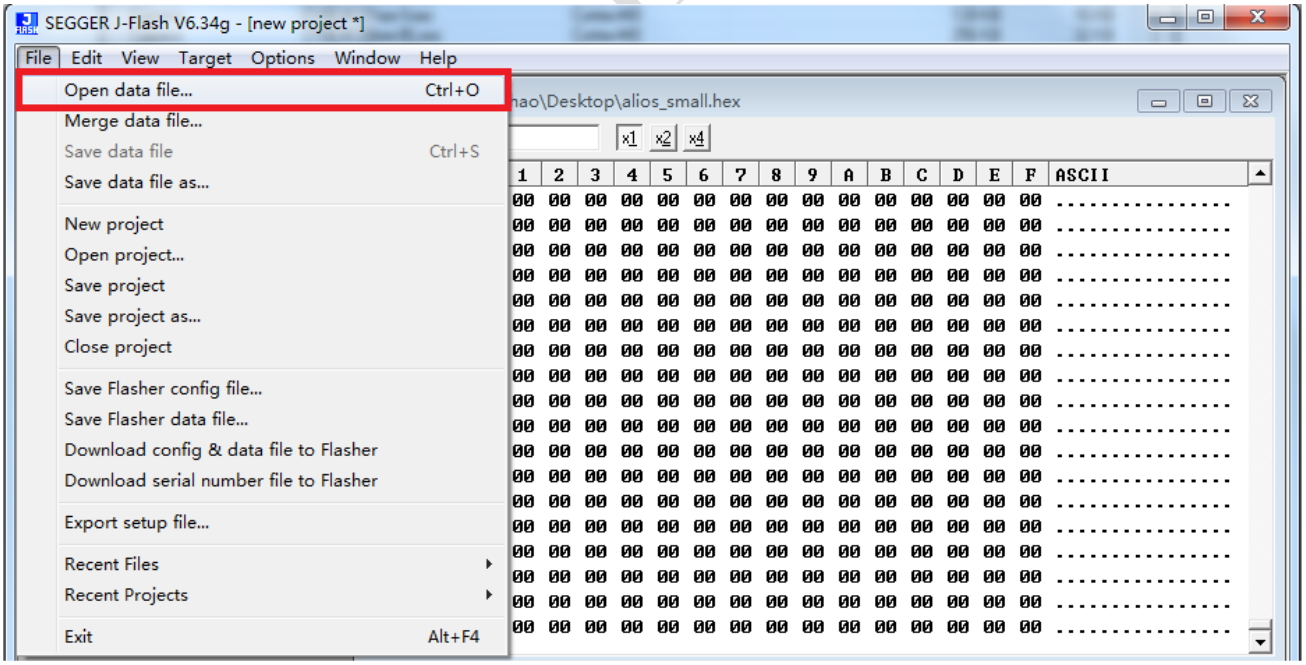


4） 选择 Cypress CY8C4147xxx-Sxxx

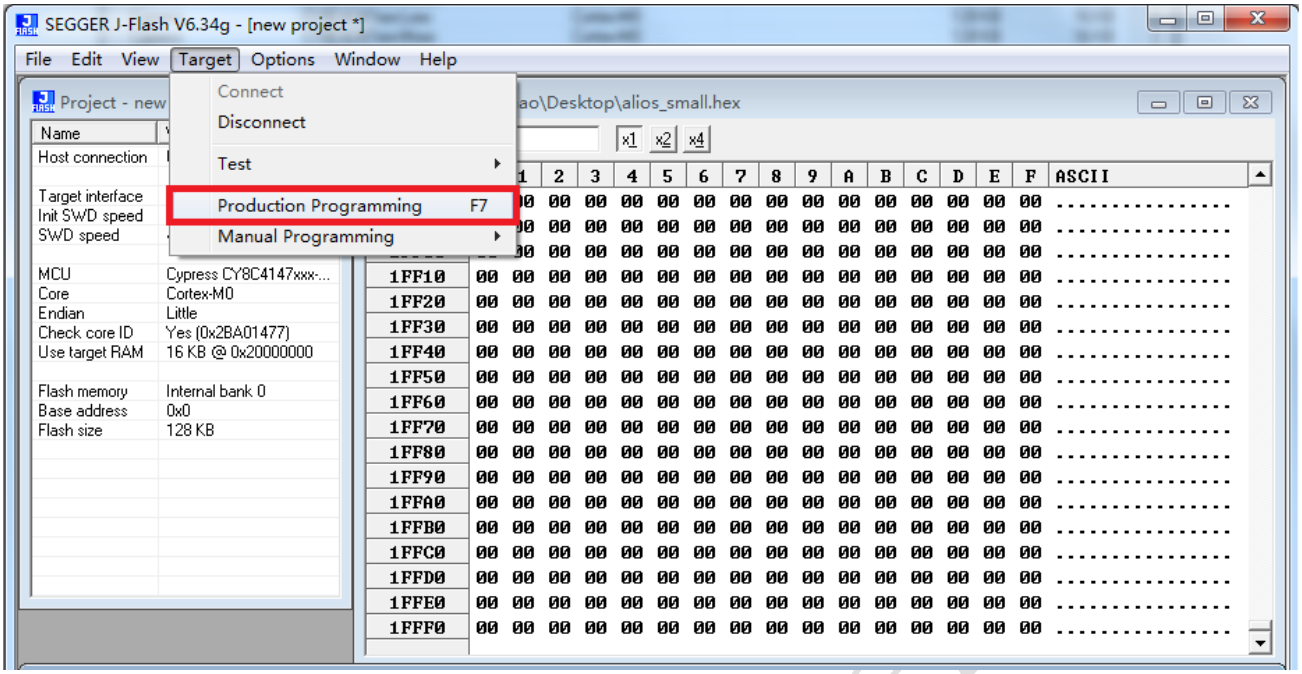




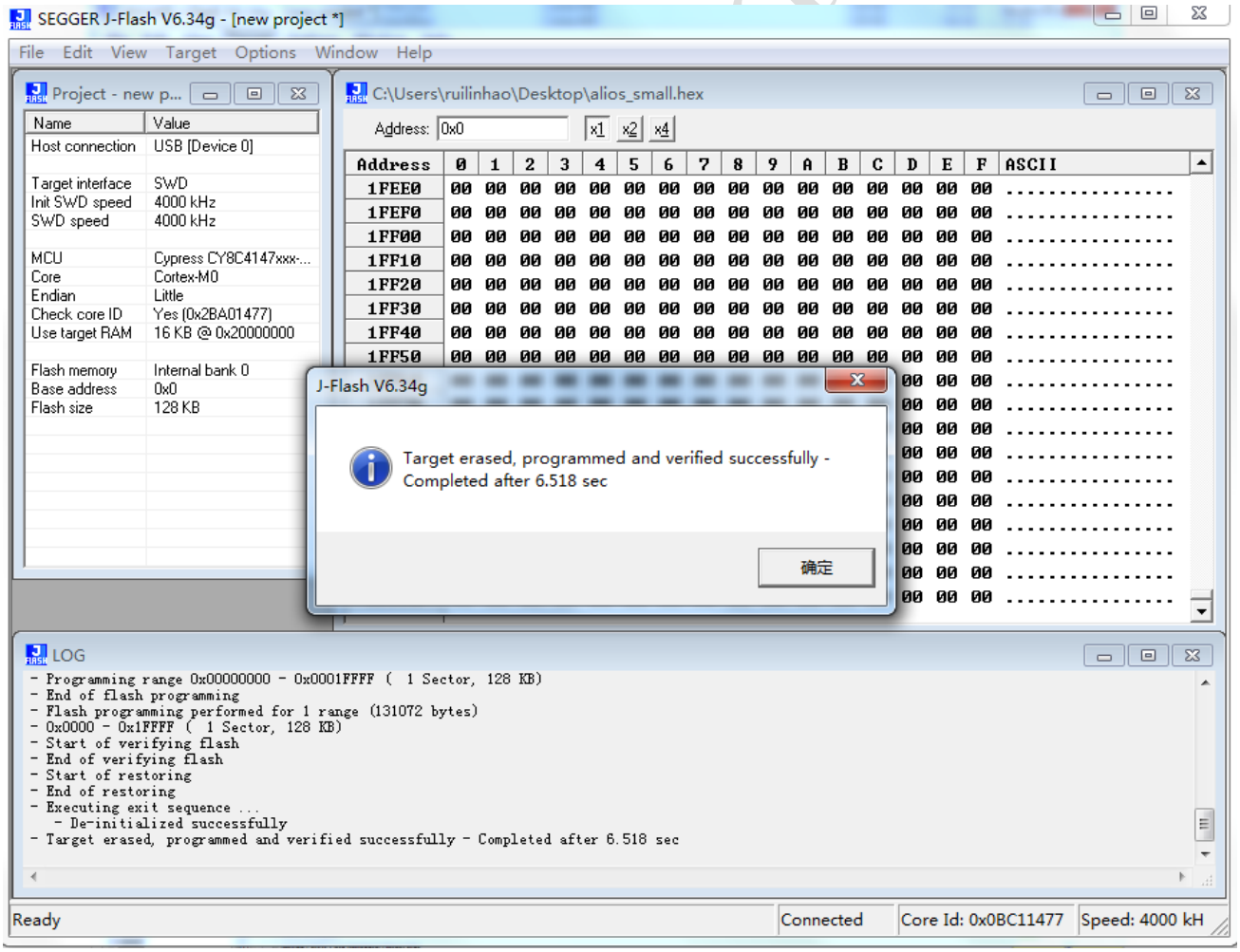
5） 打开修改后的 Hex 文件



6） 开始烧录


SEGGER J-Flash V6.34g - [new project *]
File Edit View Target Options Window Help
Connect
Disconnect
Test
Production Programming F7
Manual Programming
Project - new
Name
Host connection
Target interface
Init SWD speed
SWD speed
MCU Cypress CY8C4147xxx-...
Core Cortex-M0
Endian Little
Check core ID Yes (0x2BA01477)
Use target RAM 16 KB @ 0x20000000
Flash memory Internal bank 0
Base address 0x0
Flash size 128 KB
ao\Desktop\alios_small.hex
x1 x2 x4
ASCII

7） 烧录完成


SEGGER J-Flash V6.34g - [new project *]
File Edit View Target Options Window Help
Project - new p...
Name Value
Host connection USB [Device 0]
Target interface SWD
Init SWD speed 4000 kHz
SWD speed 4000 kHz
MCU Cypress CY8C4147xxx-...
Core Cortex-M0
Endian Little
Check core ID Yes (0x2BA01477)
Use target RAM 16 KB @ 0x20000000
Flash memory Internal bank 0
Base address 0x0
Flash size 128 KB
C:\Users\ruilinhao\Desktop\alios_small.hex
Address: 0x0
x1 x2 x4
ASCII
J-Flash V6.34g
Target erased, programmed and verified successfully - Completed after 6.518 sec
确定
LOG
- Programming range 0x00000000 - 0x0001FFFF (1 Sector, 128 KB)
- End of flash programming
- Flash programming performed for 1 range (131072 bytes)
- 0x0000 - 0x1FFFF (1 Sector, 128 KB)
- Start of verifying flash
- End of verifying flash
- Start of restoring
- End of restoring
- Executing exit sequence ...
- De-initialized successfully
- Target erased, programmed and verified successfully - Completed after 6.518 sec
Ready
Connected
Core Id: 0x0BC11477
Speed: 4000 kH

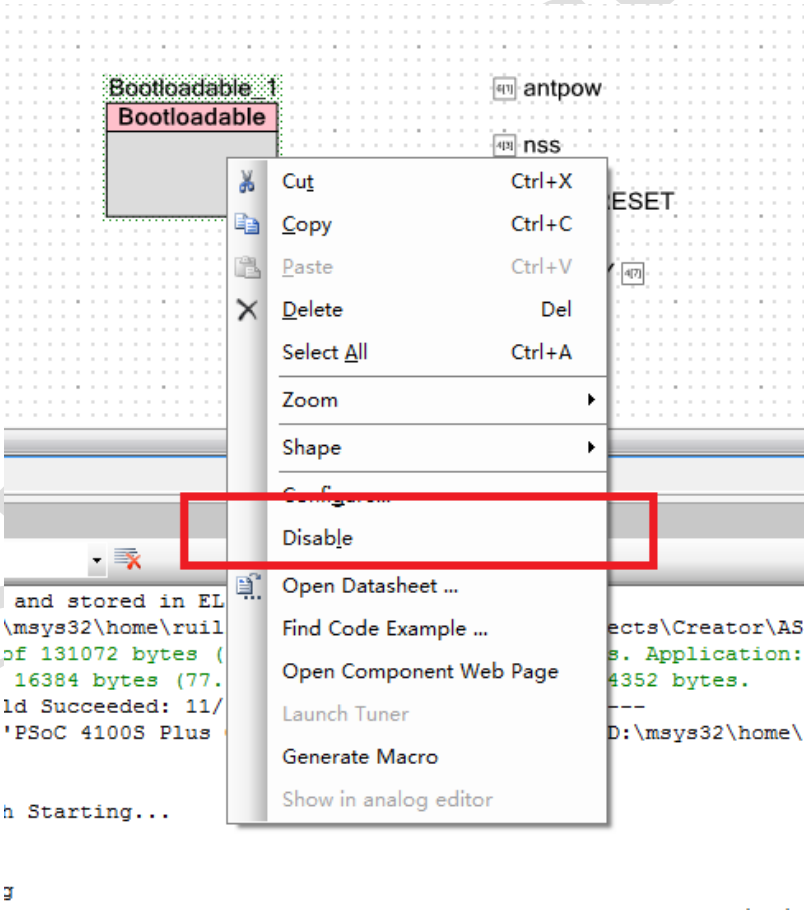
4.3 调试

使用 PSOC Creator 进行调试，需要做以下配置：

1） 设置 Debug 选项，SWD 选项为打开调试，GPIO 选项为关闭调试

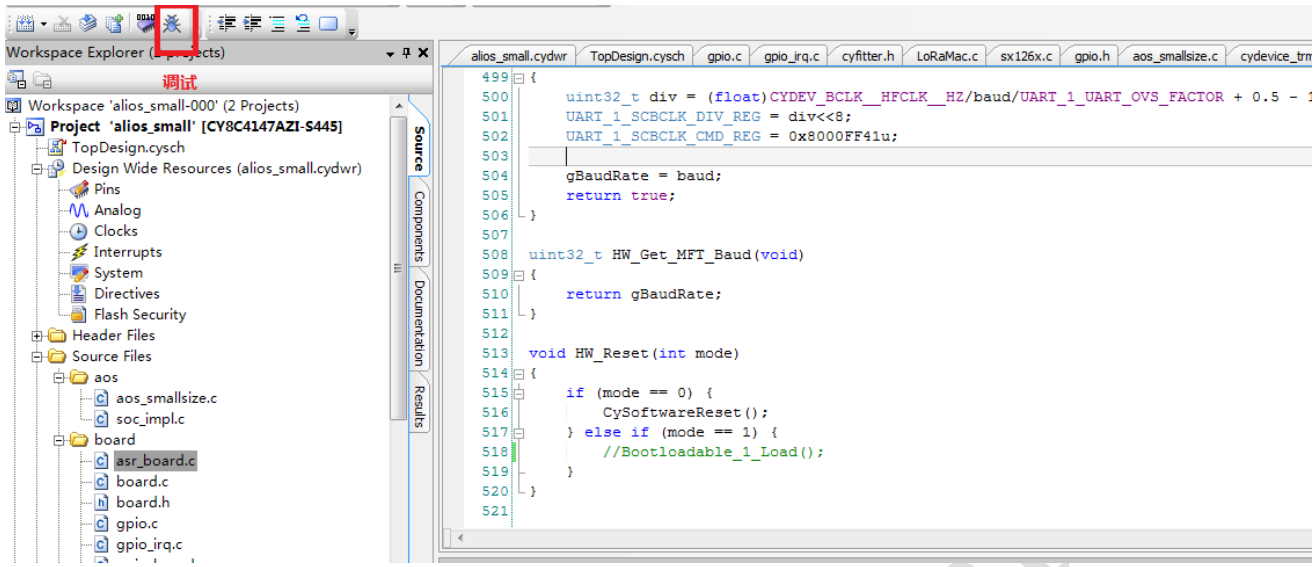
2）暂时去掉 bootloader

SDK V4.0 后增加了 uart bootloader，debug 时需要先将 bootloadable 组件 disable，同时在 asr_board.c 中注释掉 `Bootloadable_1_Load()` 的调用



4） 重新编译工程

5） 选择调试



4.4 UART 升级

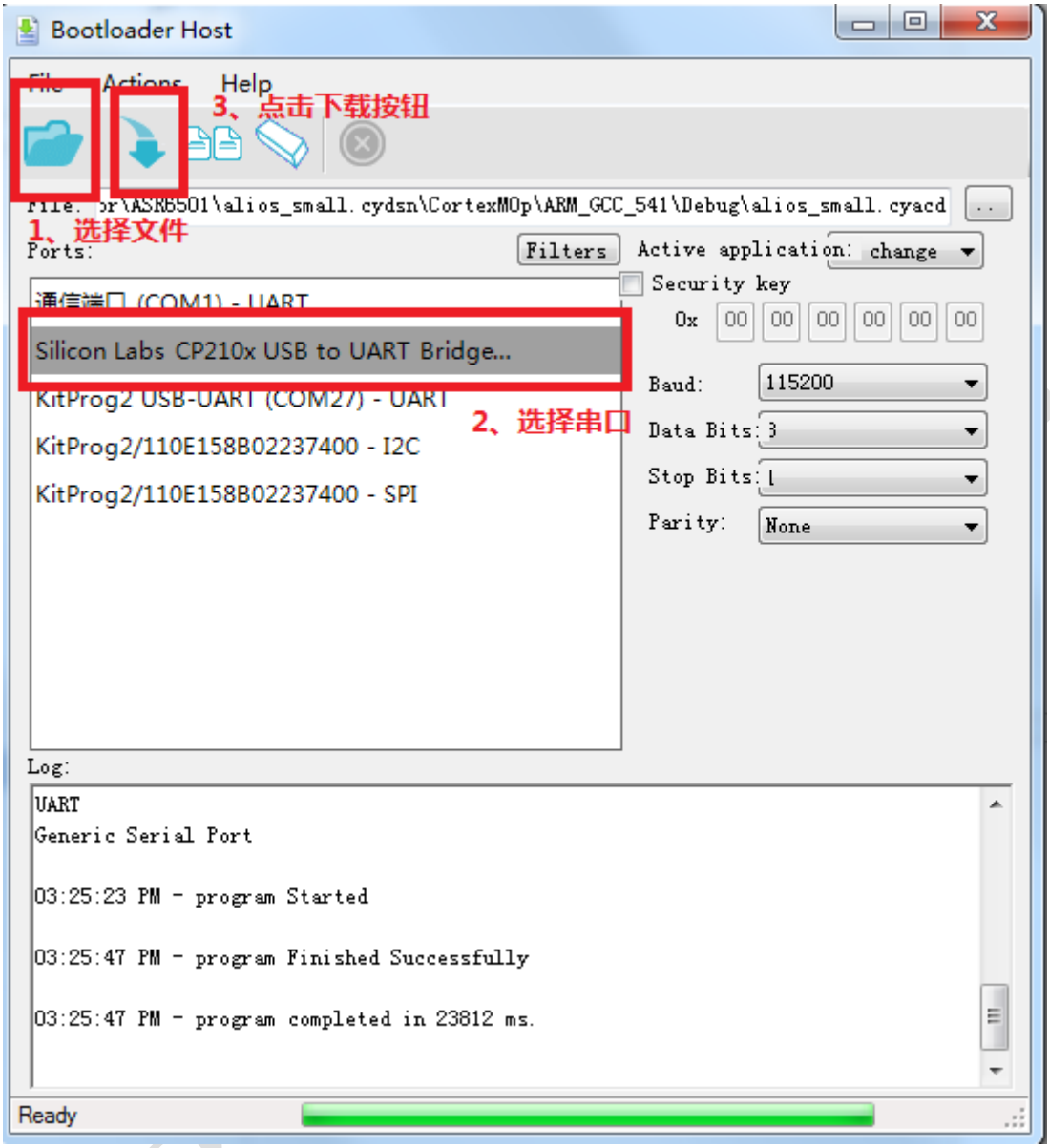
SDK v4.0 增加 bootloader 后，可使用 uart 进行升级，升级文件为 projects\Creator\ASR6501\lorawan.cydsn\CortexM0p\ARM_GCC_541\Debug\lorawan.cyacd。具体升级步骤如下：

1）在正常模式中添加代码调用 `Bootloadable_1_Load()`，使设备进入 bootloader；也可考虑在 bootloader 中添加硬件检测的方法，具体见下面注意事项 1)；

2）关闭 uart 串口连接；

3）打开 Bootloader Host (默认位置：C:\Program Files (x86)\Cypress\PSoC Creator\4.2\PSoC Creator\bin)；

4）选择升级文件，连接设备，然后点击下载按钮；



注意：

1) 如需使用硬件方式进入 bootloader，请在 bootloader 中自行修改，打开下图注释部分即可

```
int main(void)
{
    //if(Pin_DL_Read()==0){
    //    Bootloader_1_SET_RUN_TYPE(Bootloader_1_SCHEDULE_BTLDR);
    //}

    Bootloader_1_Start();
```

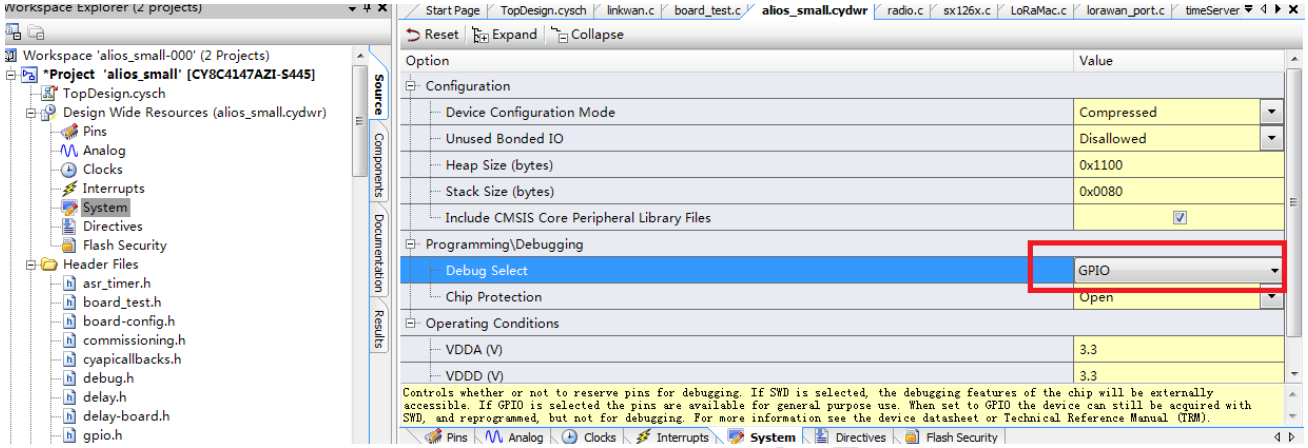
2) UART bootloader 参考文档:

http://www.cypress.com/documentation/application-notes/an68272-psoc-3-psoc-4-psoc-5lp-and-psoc-analog-coprocessor-uart

5 低功耗

5.1 配置低功耗

配置低功耗，首先要关闭调试功能：



5.2 低功耗唤醒

UART 中断，GPIO 中断，Timer 中断均可唤醒低功耗。

5.3 新增外设低功耗处理

如在开发中使用到了新的外设，请在系统进入低功耗前，将新加外设设置低功耗，在唤醒后再恢复设备状态，修改 kernel\protocols\lorawan\lora\system\low_power.c 中 LowPower_Handler 函数的处理，简单示意代码如下：

```
    SPI_1_Sleep();

    /*backup the IOs drivemode and set IOs Hiz*/
    uint32_t spi_mosi_mode = ( SPI_1_mosi_m_PC >> SPI_1_mosi_m__0__SHIFT *
SPI_1_mosi_m_DRIVE_MODE_BITS) & SPI_1_mosi_m_DRIVE_MODE_IND_MASK;
    SPI_1_mosi_m_SetDriveMode(SPI_1_mosi_m_DM_ALG_HIZ);
    ...

    /*deepsleep*/
    aos_lrwan_chg_mode.enter_stop_mode();


    /*restore the IOs drivemode */
    SPI_1_mosi_m_SetDriveMode(spi_mosi_mode);
```

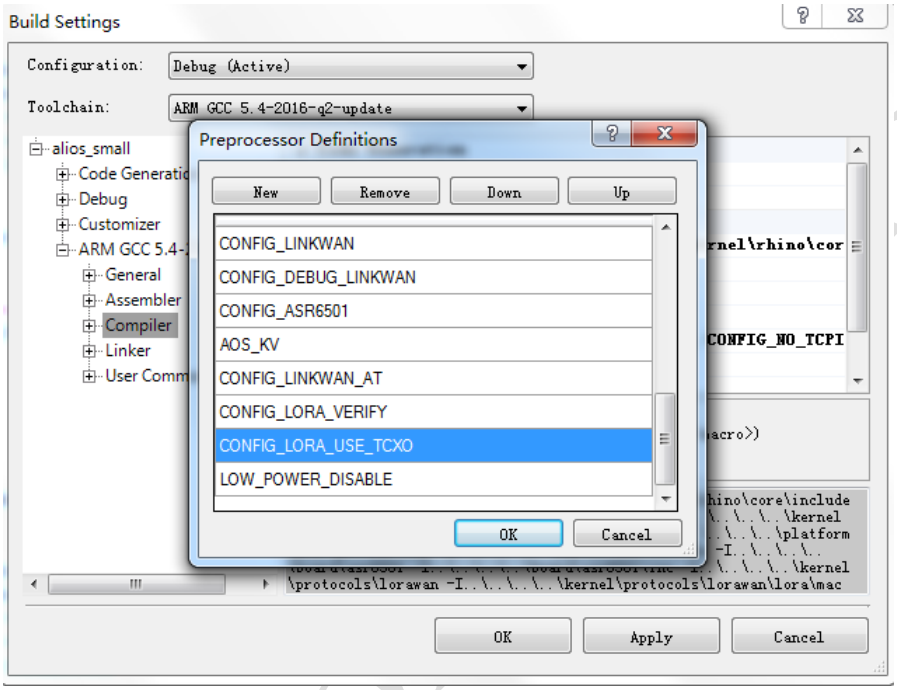
```
...

SPI_1_Wakeup();
```

6 Q&A

6.1 如何修改 SDK 支持 XO 晶振？

ASR6501/ASR6502 默认使用 TCXO 晶振，如要使用 XO 晶振，请在 Project->Build Settings 中宏定义中将 CONFIG_LORA_USE_TCXO 去掉。



6.2 如何在代码中更改设备信息？

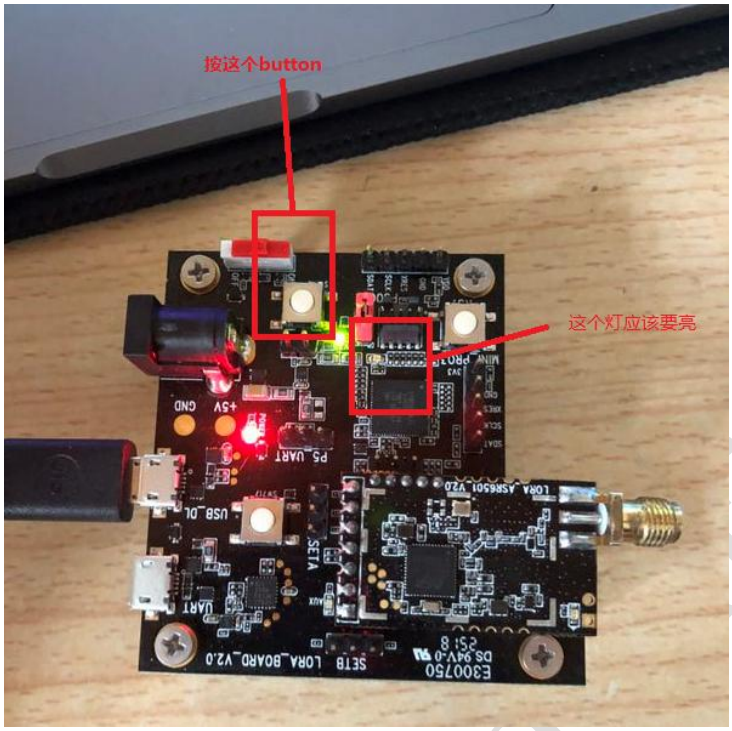
在 projects\Creator\ASR6501\lorawan.cydsn\Commissioning.h 中修改对应的三元组信息。

6.3 如何使用 ABP 模式？

在 projects\Creator\ASR6501\lorawan.cydsn\Commissioning.h 中修改宏定义 OVER_THE_AIR_ACTIVATION。

6.4 设备无法烧录?

在烧录时，底板正常应该亮 2 个绿灯，1 个红灯，如果如下图中出现只亮一个绿灯的情况，请按”SW716”按钮进行复位，绿灯亮即可烧录



6.5 SDK 编译不通过?

打开工程后，编译提示下面错误：

	Description	File	Error Location	Project
1	pft.M0086:Error in component: Bootloadable_1. The referenced Bootloader is invalid. Verify the Bootloader dependency is correct in the Bootloadable Component, then build project. Invalid bootloader hex file. Unable to read the hex file (D:\msys32 \home\ruilinhao\new_git\new\alios-lora\projects\Creator\ASR6501\alios_small.cydsn\bootloader.cydsn\CortexM0p\ARM_GCC_541 \Debug\bootloader.hex). The path does not exist.	TopDesign.cys...	Instance:Bootl...	alios_small
2	pft.M0086:Error in component: Bootloadable_1. The referenced Bootloader is invalid. Verify the Bootloader dependency is correct in the Bootloadable Component, then build project. Invalid bootloader elf file. Unable to read the elf file (D:\msys32 \home\ruilinhao\new_git\new\alios-lora\projects\Creator\ASR6501\alios_small.cydsn\bootloader.cydsn\CortexM0p\ARM_GCC_541 \Debug\bootloader.elf). The path does not exist.	TopDesign.cys...	Instance:Bootl...	alios_small
3	fit.M0050:The fitter aborted due to errors, please address all errors and rebuild.			alios_small

这是因为 SDK4.0 加入了 bootloader，需要先编译 bootloader，然后再编译 alios_small 工程。

7 参考资料

7.1 ALIOS 资料

https://github.com/alibaba/AliOS-Things/wiki

7.2 LoRaWan 资料

- Lorawan 代码

https://github.com/Lora-net/LoRaMac-node

- LORAMAC 介绍

http://stackforce.github.io/LoRaMac-doc/index.html

- Lora 联盟文档资料

https://lora-alliance.org/lorawan-for-developers

7.3 PSOC4 资料

- Creator 使用帮助

Creator 中点击 Help->PSOC Creator Help Topics

- PSOC4 示例代码

在 Creator 中点击 File->Code Example

- Cypreass 官网

http://www.cypress.com/

- PSOC4 资料

http://www.cypress.com/products/32-bit-arm-cortex-m0-psoc-4

- PSOC 4100s Plus 寄存器手册

http://www.cypress.com/documentation/technical-reference-manuals/psocr-4100s-plus-psoc-4-registers-technical-reference

- PSOC4 4100s Plus TRM 手册

http://www.cypress.com/documentation/technical-reference-manuals/psoc-4100s-and-psoc-4100s-plus-psoc-4-architecture