

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 LoRa SW User Guide

文件状态: [√] 正在修改 [] 正式发布	当前版本:	V4.2
	作者:	Qipan Li
	启动日期:	2018-07-04
	审核:	
	完成日期:	2019-03-27

翱捷科技（上海）有限公司

ASR Microelectronics Co., Ltd

(版本所有, 翻版必究)

版本历史

版本号	修改日期	作 者	修 改 说 明
V1.0	2018.07.5	Qipan Li	Initial Version
V2.0	2018.07.20	Qipan Li	Add: Change timer from eco to wco Add cascade timer Add deepsleep support Update AT commands
V3.0	2018.08.28	Ruilin Hao	增加功耗和 Q&A 部分；修改版本更新内容
V3.1	2018.09.29	Ruilin Hao	Update
V4.0	2018.11.14	Ruilin Hao	Update for SDK4.0
V4.1	2018.11.27	Ruilin Hao	Update for SDK4.1
V4.2	2019.03.05	Ruilin Hao	Update for SDK4.2
V4.2	2019.03.27	Ruilin Hao/Yeao	Change the doc to User Guide and update some chapters

Table of Contents

1	概述.....	5
2	主要功能.....	6
2.1	硬件示意图	6
2.2	软件示意图	6
2.3	通信示意图	6
2.4	主要功能描述	7
3	准备.....	8
3.1	硬件准备	8
3.2	软件安装	9
3.3	SDK 获取.....	9
3.3.1	Git 获取.....	9
3.3.2	直接下载	9
3.4	联网准备（联网通信需要）	10
3.4.1	节点信息申请	10
3.4.2	网关配置	10
4	软件编译与烧录.....	11
4.1	SDK 简介	11
4.1.1	工程说明	11
4.1.2	ASR6502 与 ASR6501 的区别.....	11
4.1.3	代码结构	12
4.2	编译	13
4.3	烧录	15
4.3.1	PSoC Creator 烧录	15
4.3.2	PSoC Programmer 烧录	16
4.3.3	J-Flash 烧录.....	19
4.4	调试	23
4.5	UART 升级	24
4.6	代码阅读	26
5	配置通信.....	27
5.1	LinkWan 通信	27
5.1.1	LinkWan 公网	27
5.1.2	LinkWan 认证实验室	29
5.2	LoRaWan 通信	29

6	低功耗.....	31
6.1	配置低功耗	31
6.2	低功耗唤醒	31
6.3	新增外设低功耗处理	32
7	测试.....	33
7.1	功耗测试	33
7.1.1	硬件连接	33
7.1.2	DeepSleep 功耗测试	33
7.1.3	接收功耗测试	34
7.1.4	发送功耗测试	34
7.2	功率测试	34
7.3	距离测试	34
8	Q&A.....	37
8.1	如何修改 SDK 支持 XO 晶振?	37
8.2	如何配置自动联网?	37
8.3	如何在代码中更改设备信息?	37
8.4	如何使用 ABP 模式?	38
8.5	如何配置同、异频节点?	38
8.6	如何配置 CLASS B 节点?	38
8.7	如何配置 Heap Size?	38
8.8	设备无法烧录?	38
8.9	如何加密三元组信息?	39
8.10	AsrLib.a 与 AsrLib_small.a 有什么差异?	39
8.11	SDK 编译不通过?	39
9	参考资料.....	40
9.1	ALIOS 资料.....	40
9.2	LoRaWan 资料.....	40
9.3	PSOC4 资料.....	40

1 概述

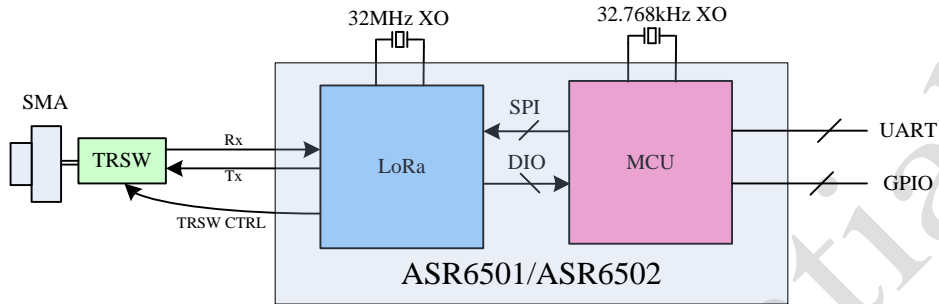
ASR6501/ASR6502 集成了 Semtech 最新的 LoRa 芯片 SX1262，结合 Cypress PSoC4100S 的优良性能打造最小功耗的 LoRa 芯片模组。软件方面通过集成 Alios Things，完成了节点-网关-Ali 云的通信过程，并配备了 AT 命令用以用户交互。

ASR Confidential

2 主要功能

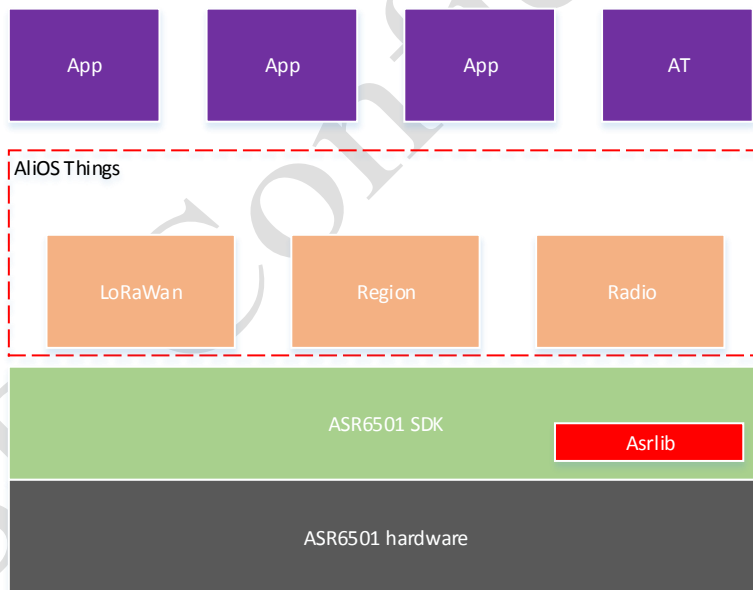
2.1 硬件示意图

ASR6501/ASR6502 硬件示意图如下描述，MCU 通过 SPI 完成与 LoRa 芯片的通讯，LoRa 芯片通过 DIO 中断 MCU 处理。



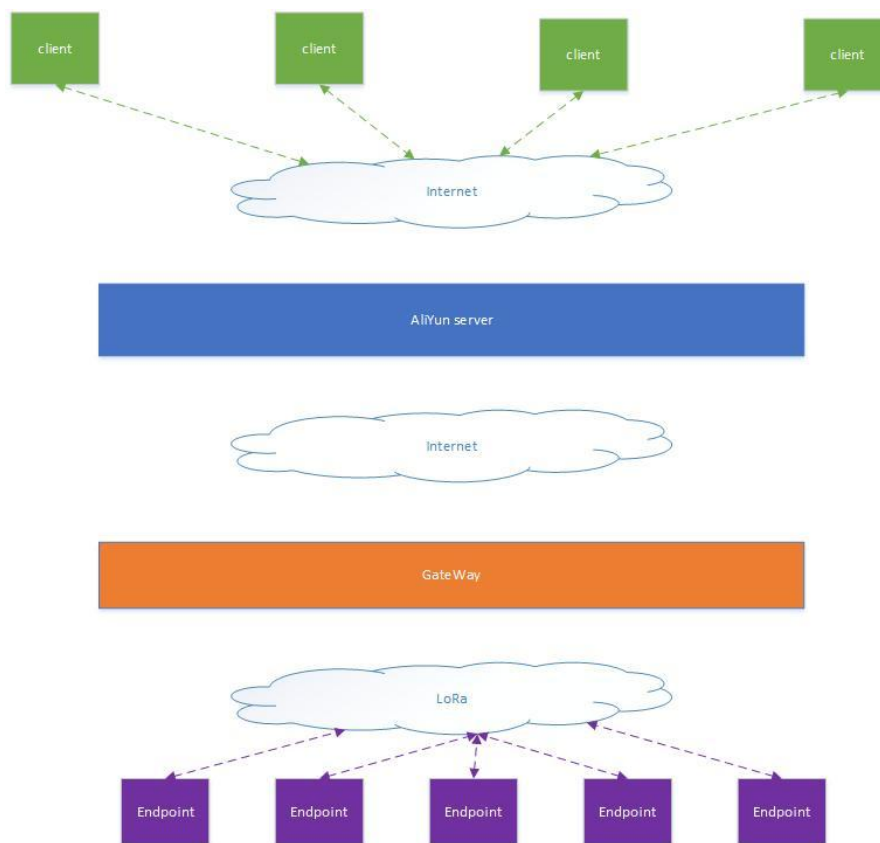
2.2 软件示意图

AliOS Things 内部集成 LoRaWAN 协议栈部分，ASR6501/ASR6502 完成对 AliOS Things 的对接并针对 SX1262 进行了部分优化，提高节点的稳定性。



2.3 通信示意图

ASR6501/ASR6502 作为节点设备，软件具备 LoRaWan 功能完成与网关、网络服务器的通信，已经通过了 Ali LoRaWan 认证，满足 LoRa 协议一致性要求，当前节点已支持 ClassA、ClassB 和 ClassC 模式。Ali 针对 CN470 频段作了划分，衍生出同频、异频的概念，ASR6501 节点同时支持同频、异频，因此网关部分需同时满足 Ali 对 LoRa 网关部分的要求。



2.4 主要功能描述

Semtech 发起成立的 LoRa 联盟制定了 LoRaWan 协议栈，保证节点、网关、网络服务器的互联互通，当前 ASR6501/ASR6502 默认支持 LinkWan 网关与阿里云通信。

支持的主要功能：

- 支持 LoRaWan 协议栈
- 支持 OTAA 入网
- 支持 ABP 入网
- Mac 命令识别和响应
- 上行数据和下行数据
- 支持 ClassA, ClassB 和 ClassC
- 支持同频 (LinkWan)、异频
- 通信频点随机切换
- 入网扫描

3 准备

3.1 硬件准备

LoRa 节点必需硬件列表如下：

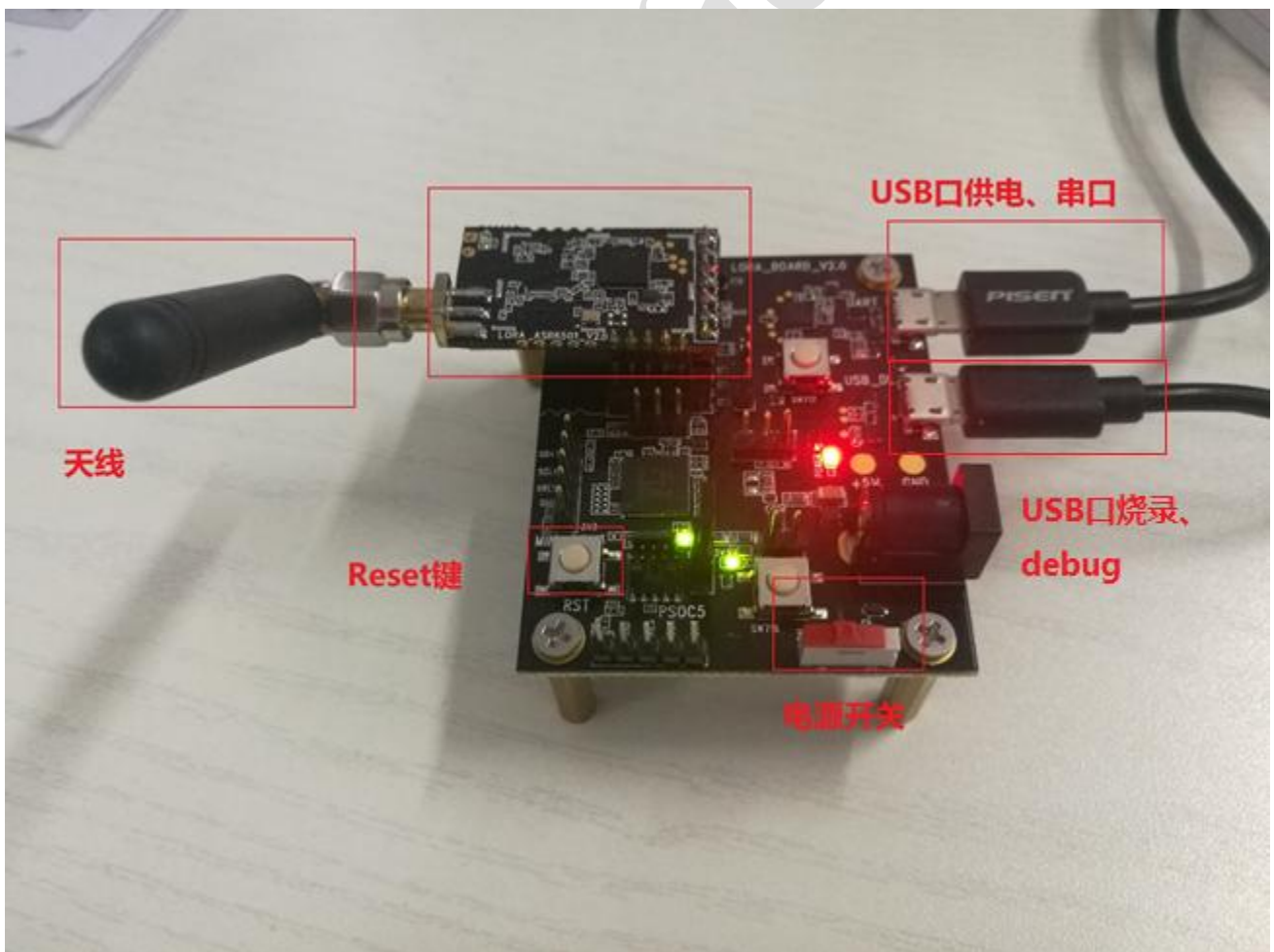
- 1) ASR6501/ASR6502 LoRa 子板 1 个
- 2) ASR6501/ASR6502 LoRa 母板 1 个
- 3) 天线 1 根
- 4) USB 线 2 根
- 5) PC 机 1 台

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

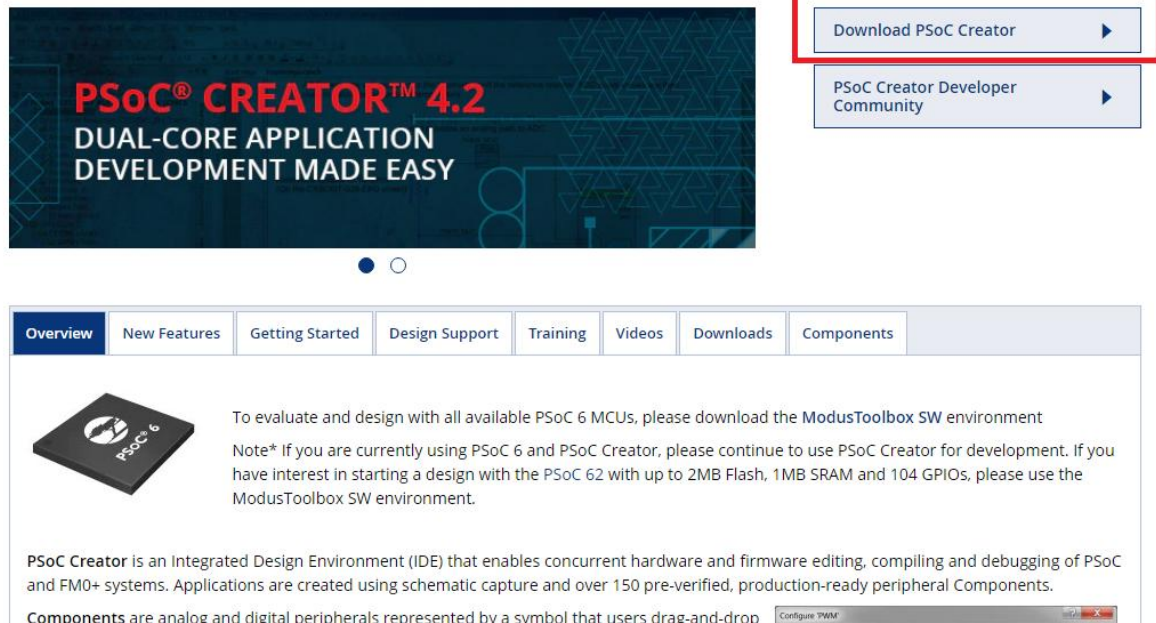


硬件连接图

3.2 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>)

PSoC® Creator™ Integrated Design Environment (IDE)



3.3 SDK 获取

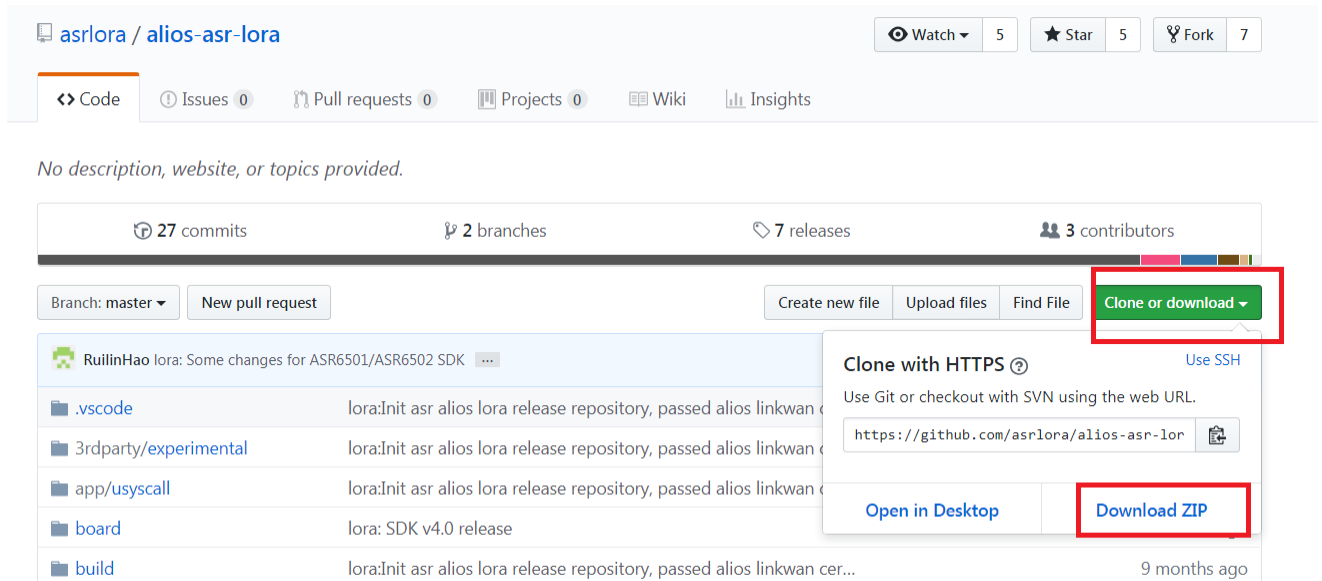
3.3.1 Git 获取

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
```

3.3.2 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



3.4 联网准备（联网通信需要）

3.4.1 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。

3.4.2 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，。

4 软件编译与烧录

4.1 SDK 简介

4.1.1 工程说明

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

此工程是主要的示例工程，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，本文档主要基于 alios_small 工程进行说明。

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。

4.1.2 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

\UART_1:tx\	P3 [1]	▼	19	▼	
antpow	P6 [1]	▼	13	▼	
diol	P4 [6]	▼	33	▼	
nss	P4 [3]	▼	30	▼	
pin wakeup	P0 [2]	▼	41	▼	
SPI_BUSY	P4 [7]	▼	34	▼	

2) ASR6502 中这两个 pin 脚的配置

\UART_1:tx\	P3 [1]	▼	19	▼	✓
antpow	P3 [4]	▼	22	▼	✓
diol	P4 [6]	▼	33	▼	✓
nss	P4 [3]	▼	30	▼	✓
pin_wakeup	P0 [3]	▼	42	▼	✓
SPI_BUSY	P4 [7]	▼	34	▼	✓

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。

4.1.3 代码结构

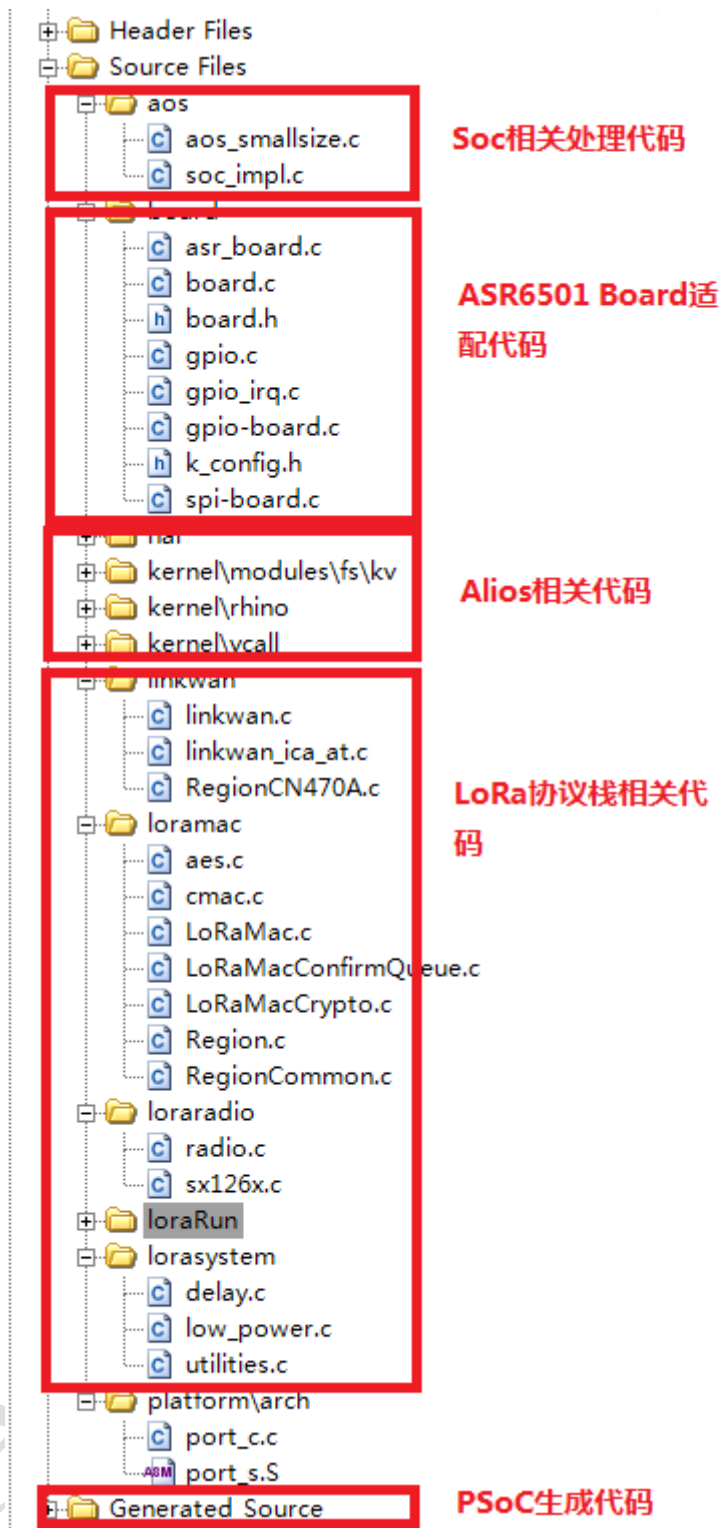
如下图所示，在 alios_small 工程中，代码主要分为 SOC 相关代码，ASR6501 Board 适配代码，Alios 相关代码，LoRa 协议栈相关代码和 PSOC4 生成代码。

LoRa 协议栈相关代码涉及文件较多，其中 linkwan_ica_at.c 集成有 AT 命令集，用户可在此新加或者修改相关命令项。

linkwan.c 是为了更好地使用 LoRaWan 协议栈而建立的有限状态机，帮助用户完成入网、数据发送、低功耗等功能。

RegionCN470A.c 针对 Ali 对 Lora 网段的划分，在原有 LoraWan 的 RegionCN470 上进行了相关细分，主要有同频、异频、网段划分、随机频率等。

其他如 aes.c、cmac.c、LoRaMac.c、LoRaMacCrypto.c、Region.c、RegionCommon.c 是 LoRaWan 通用的协议代码。

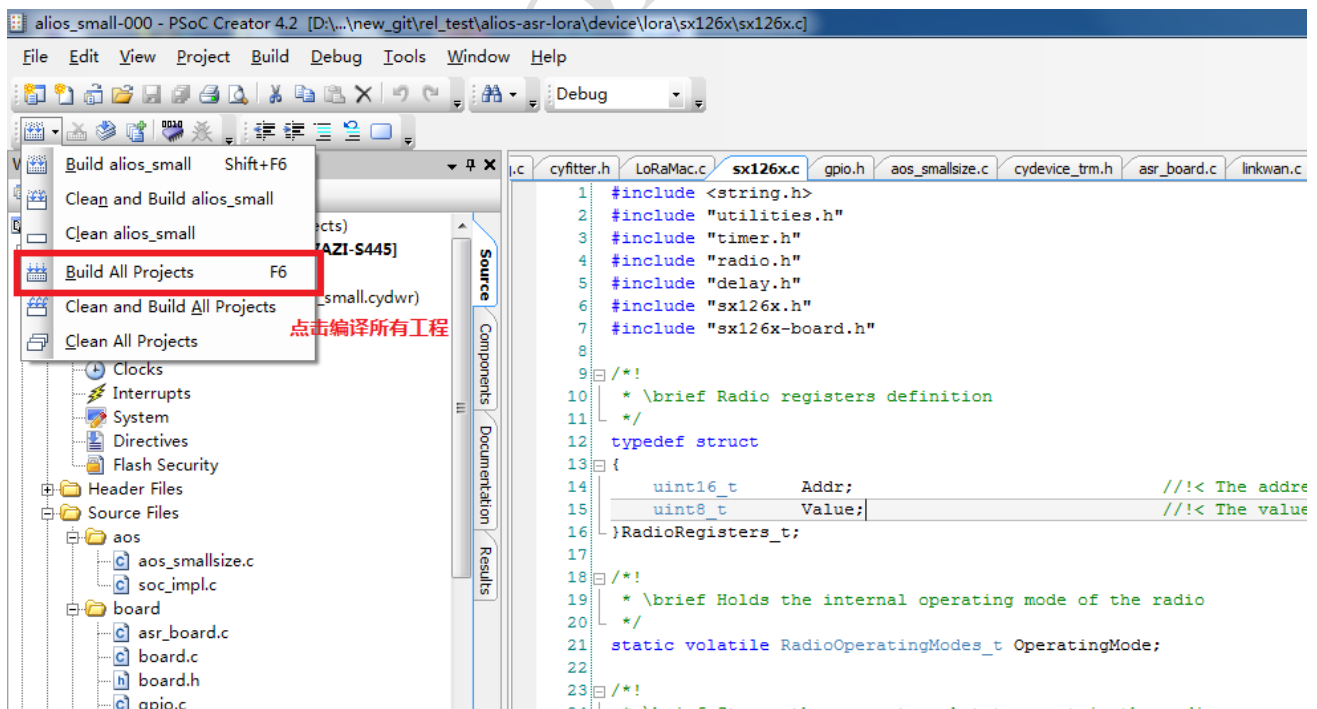


4.2 编译

- 1) 在工程目录下打开后缀为.cyprj 的工程文件

本地磁盘 (C:) > 用户 > qipanli > lora > github-asr-lora > alios-asr-lora > projects > Creator > ASR6501 > alios_small.cydsn >				
共享 刻录 新建文件夹				
名称	修改日期	类型	大小	
alios_kernel_certificate.cydsn	2018/7/5 13:54	文件夹		
Backup	2018/7/5 14:53	文件夹		
codegentemp	2018/7/5 14:57	文件夹		
CortexM0p	2018/7/5 14:57	文件夹		
Export	2018/7/5 14:57	文件夹		
Generated_Source	2018/7/5 14:57	文件夹		
TopDesign	2018/7/5 13:54	文件夹		
.gitignore	2018/7/5 13:54	文本文档	1 KB	
alios_small.cydx	2018/7/5 14:57	CYCDX 文件	422 KB	
alios_small.cydwr	2018/7/5 13:54	CYDWR 文件	78 KB	
alios_small.cyfit	2018/7/5 14:57	CYFIT 文件	149 KB	
alios_small.cypj	2018/7/5 14:57	PSoC Creator Pr...	223 KB	
alios_small.cypj.qipanli	2018/7/5 14:57	QIPANLI 文件	475 KB	
alios_small.rpt	2018/7/5 14:57	RPT 文件	95 KB	
alios_small-000.cywrk	2018/7/5 13:54	PSoC Creator W...	2 KB	
alios_small-000.cywrk.qipanli	2018/7/5 14:57	QIPANLI 文件	27 KB	
AsrLib.a	2018/7/5 13:54	A 文件	243 KB	
BUILD.log	2018/7/5 14:57	文本文档	365 KB	
cyapicallbacks.h	2018/7/5 13:54	C/C++ 标头	1 KB	
main.c	2018/7/5 13:54	C 源	2 KB	

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



3) 编译完成后会在 Output 框显示编译结果：

```

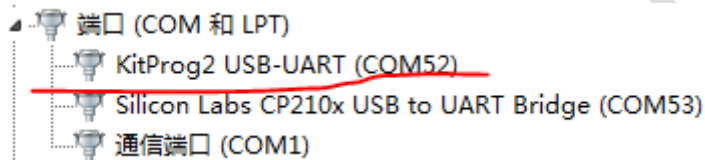
Output
Show output from: All
arm-none-eabi-ar.exe -rs .\CortexM0p\ARM_GCC_541\Debug\alios_small.a .\CortexM0p\ARM_GCC_541\Debug\UART_1.o .\CortexM0p\ARM_GCC_541\Debug\UART_1_SPI_UART.o
arm-none-eabi-ar.exe: creating .\CortexM0p\ARM_GCC_541\Debug\alios_small.a
arm-none-eabi-gcc.exe -Wl,--start-group -o D:\msys32\home\ruilinhao\new_git\rel_test\alios-asr-lora\projects\Creator\ASR6501\alios_small.cydsn\CortexM0p\AF
cyelftool.exe -B D:\msys32\home\ruilinhao\new_git\rel_test\alios-asr-lora\projects\Creator\ASR6501\alios_small.cydsn\CortexM0p\ARM_GCC_541\Debug\alios_small
No ELF section .cychecksum found, creating one
Application checksum calculated and stored in ELF section .cychecksum
Checksum calculated and stored in ELF section .cymeta
cyelftool.exe -S D:\msys32\home\ruilinhao\new_git\rel_test\alios-asr-lora\projects\Creator\ASR6501\alios_small.cydsn\CortexM0p\ARM_GCC_541\Debug\alios_small
Flash used: 101444 of 131072 bytes (77.4%). Bootloader: 5120 bytes. Application: 96068 bytes. Metadata: 256 bytes.
SRAM used: 13812 of 16384 bytes (84.3%). Stack: 128 bytes. Heap: 4352 bytes.
----- Rebuild Succeeded: 03/27/2019 09:43:57 -----

```

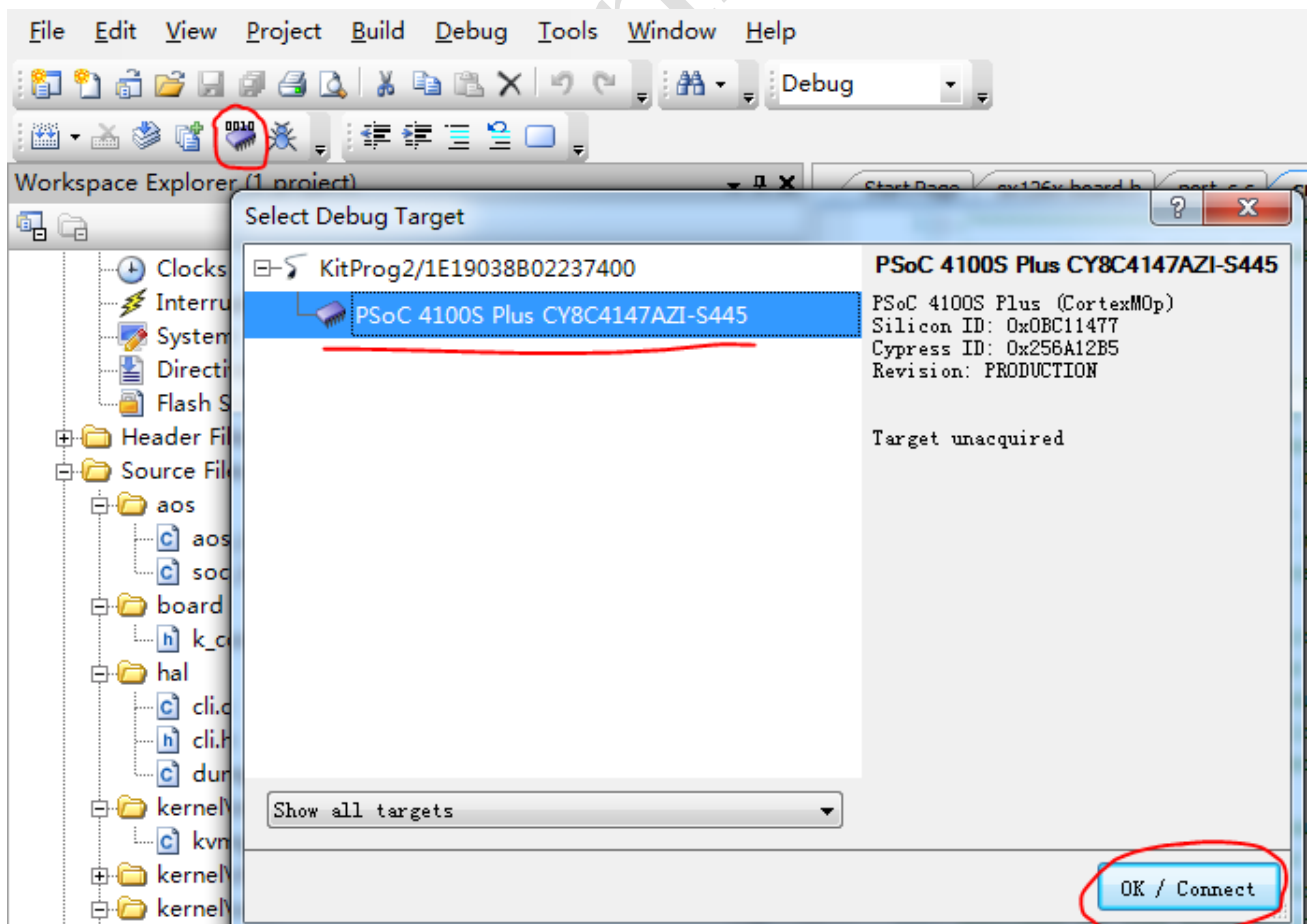
4.3 烧录

4.3.1 PSoC Creator 烧录

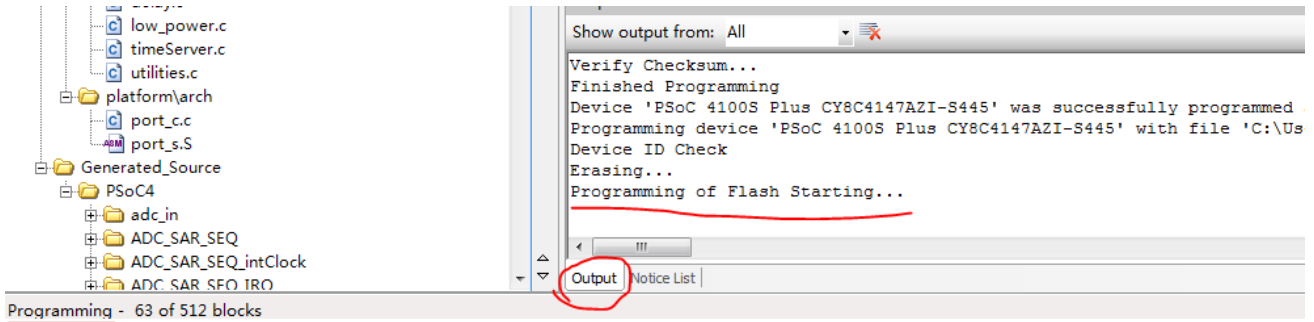
- 1) 连接 MicroUSB Cable 到 ASR6501/ASR6502 LoRa 母板的 USB_DL 口，并连接 PC，等待驱动安装完成，在 PC 上回出现 KitProg 设备。



- 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



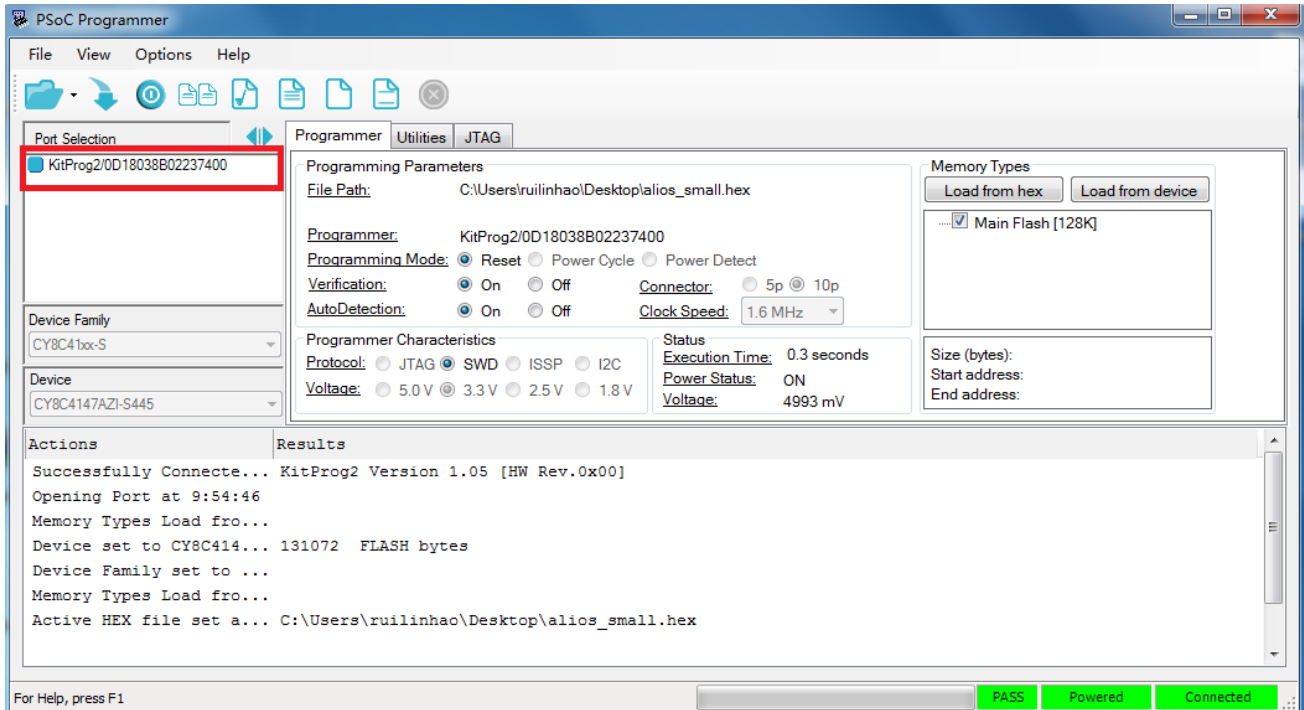
4) 烧录完成后在 Output 框显示:

```
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.
```

4.3.2 PSoC Programmer 烧录

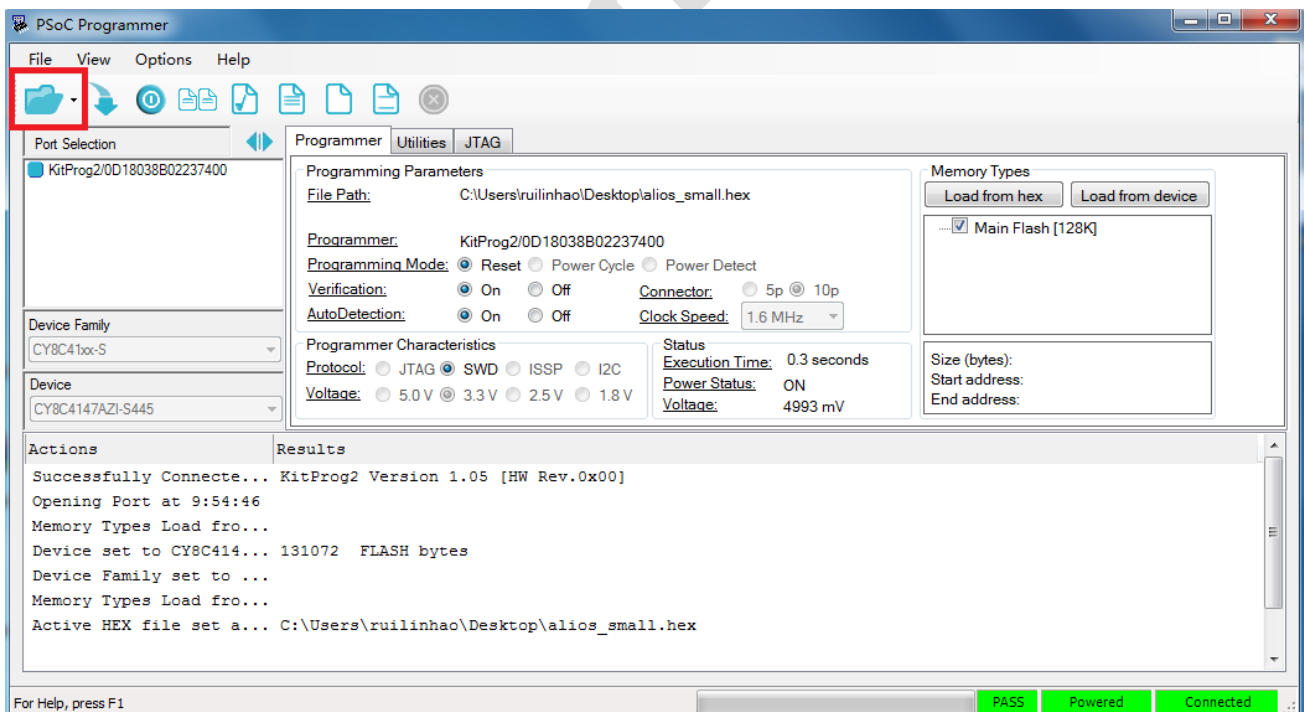
用户也可以使用 PSoC Programmer，烧录 PSoC Creator 生成的 hex，步骤如下：

- 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：

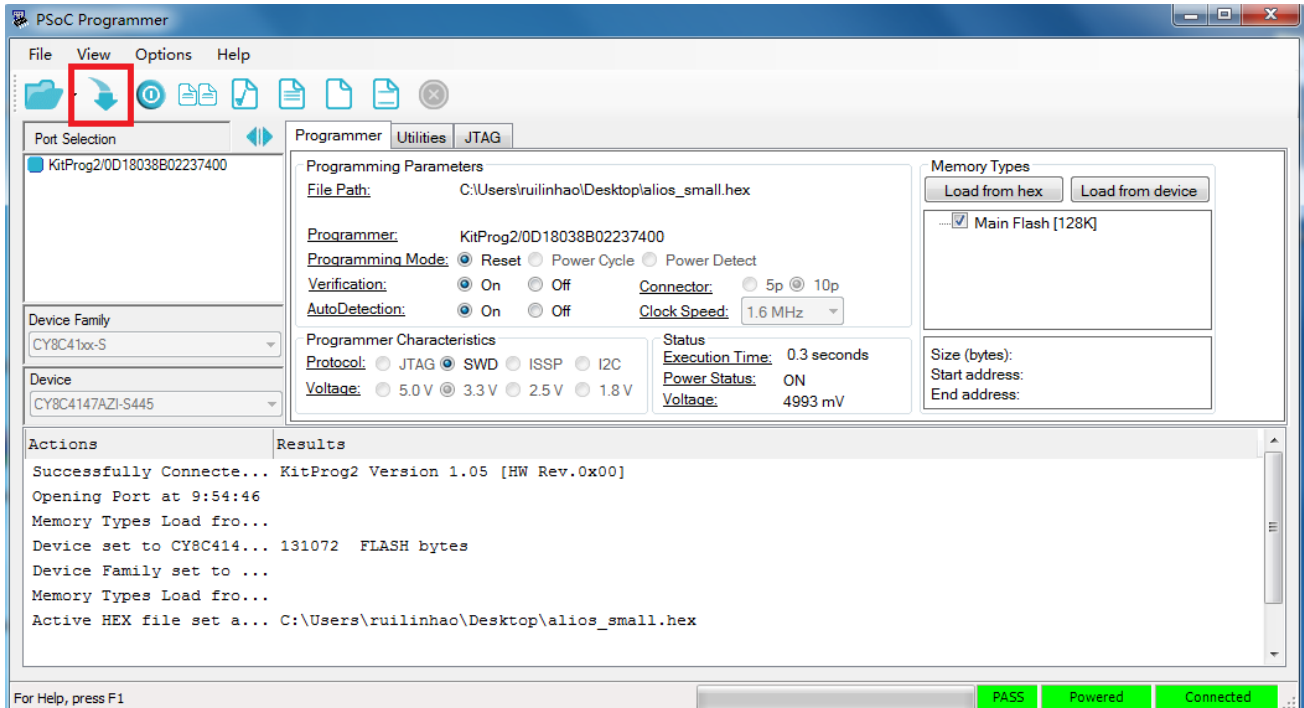


2) 选择下面打开文件的按钮，选择要烧录的 hex 文件，如：

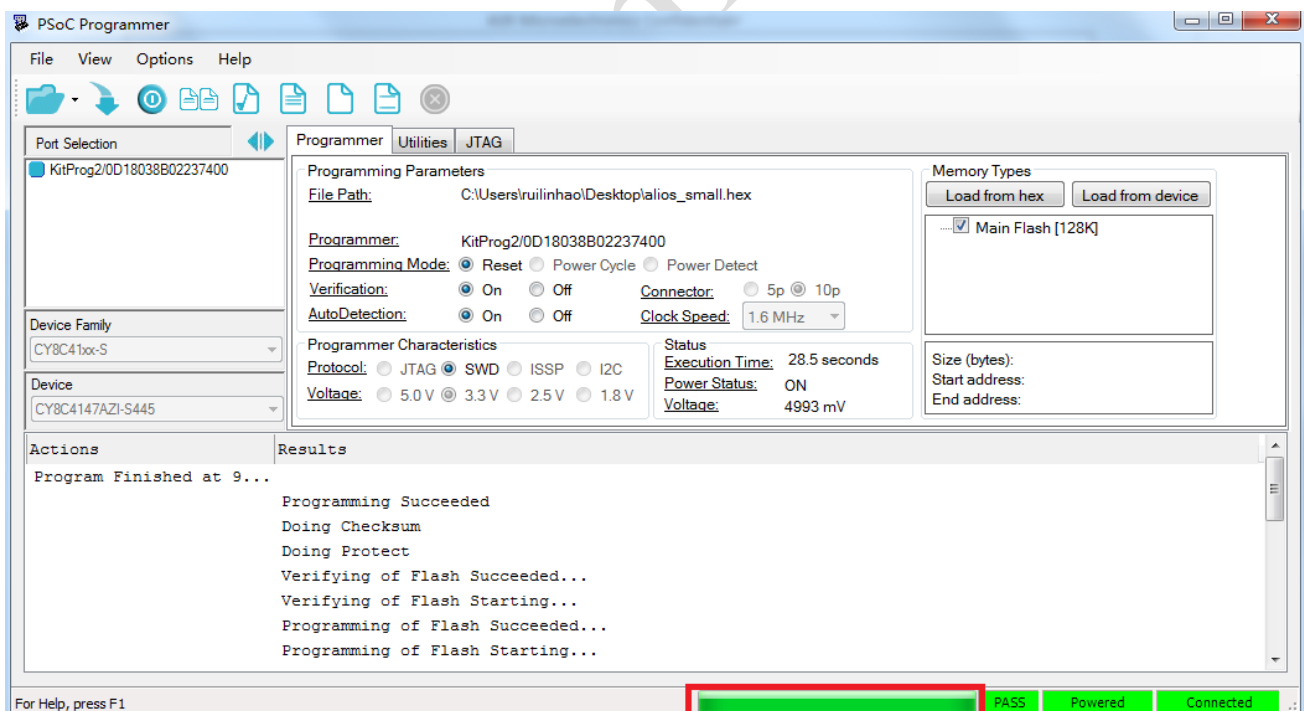
projects\Creator\ASR6501\alios_small.cydsn\CortexM0p\ARM_GCC_541\Debug\alios_small.hex

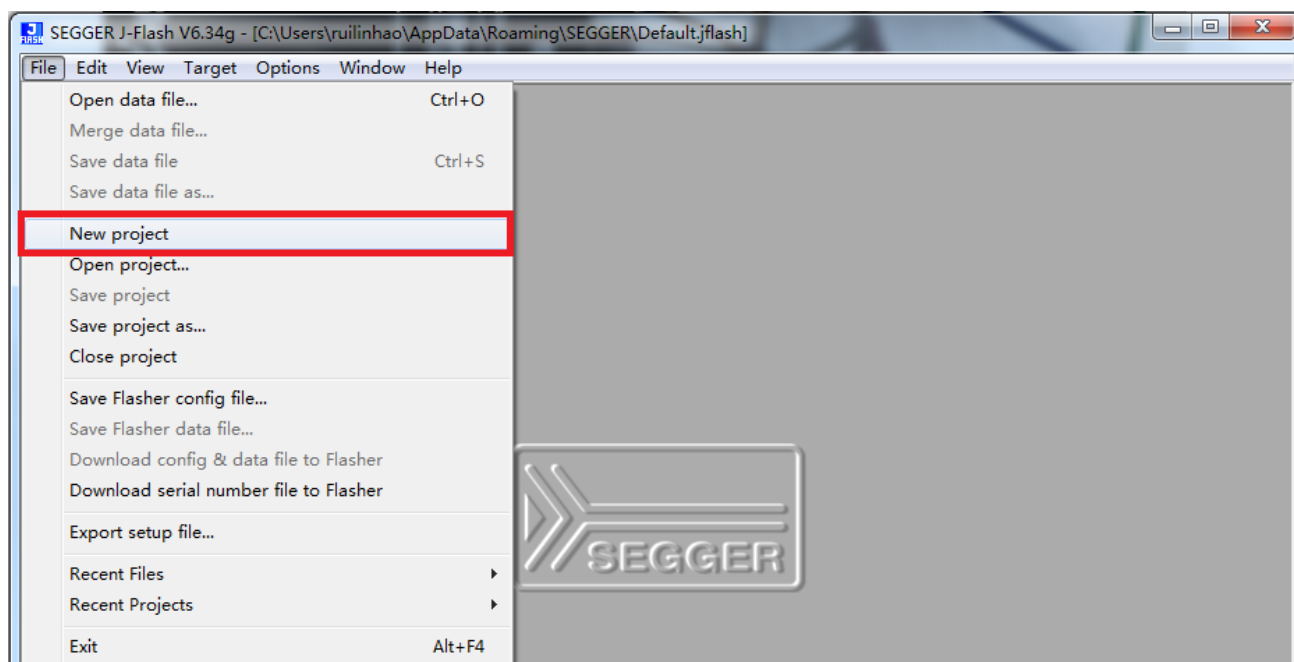


3) 然后选择 Program 按钮，等待烧录完成

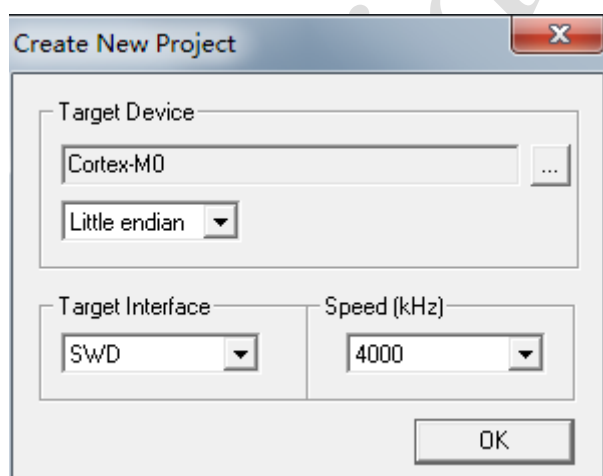


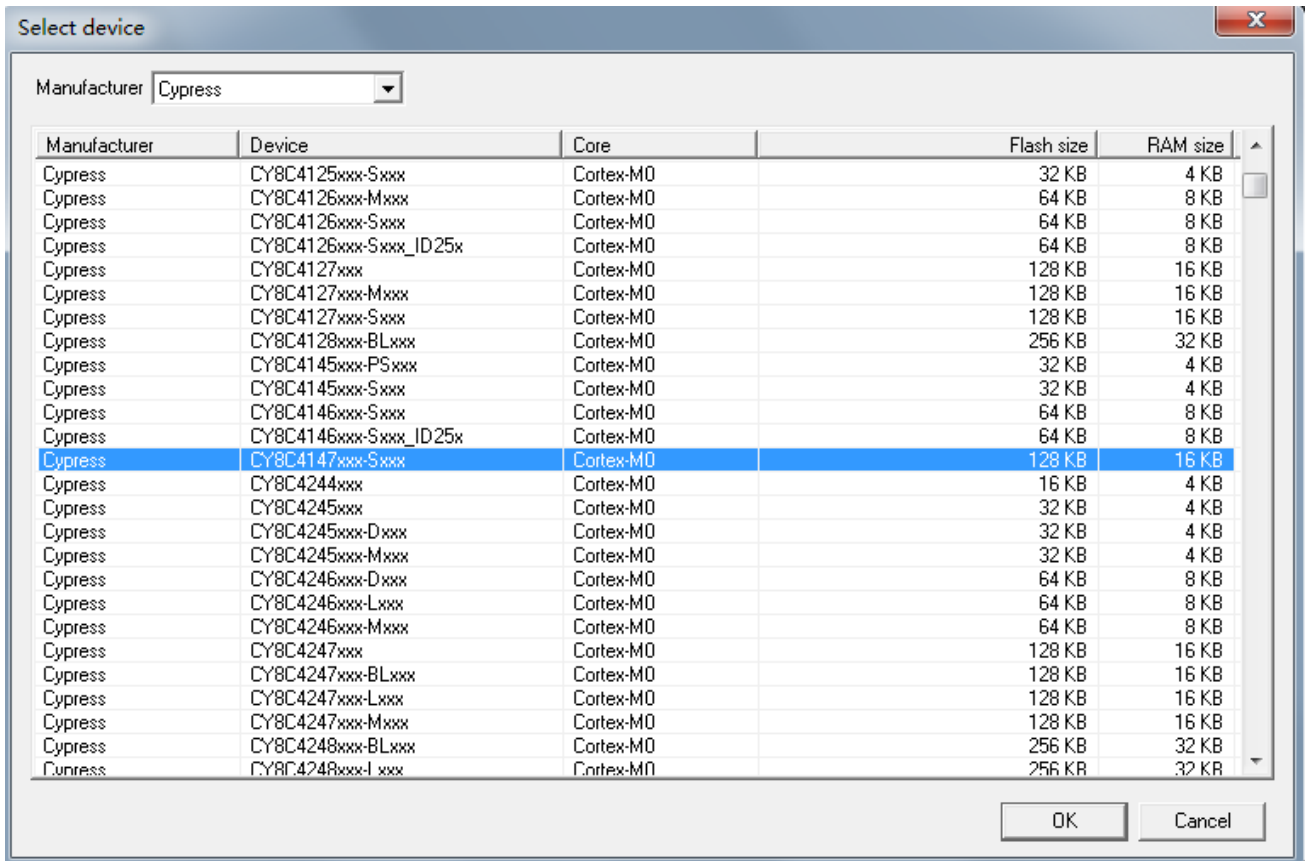
4) 烧录完成



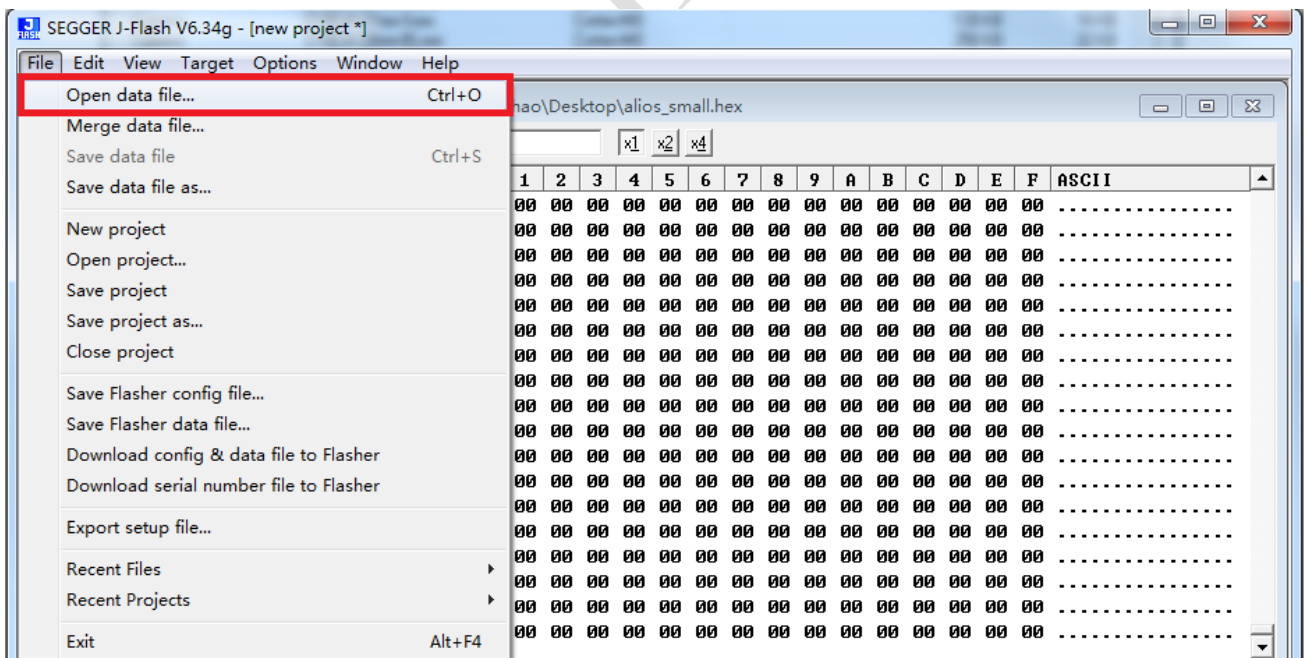


4) 选择 Cypress CY8C4147xxx-Sxxx

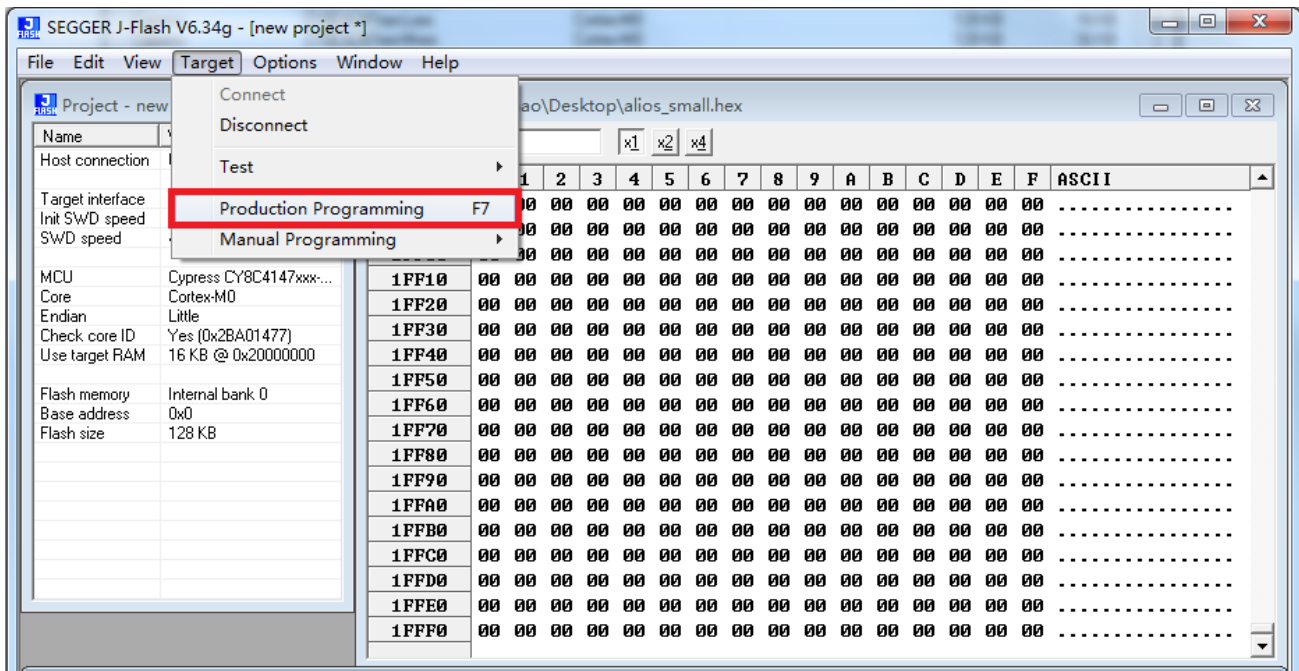




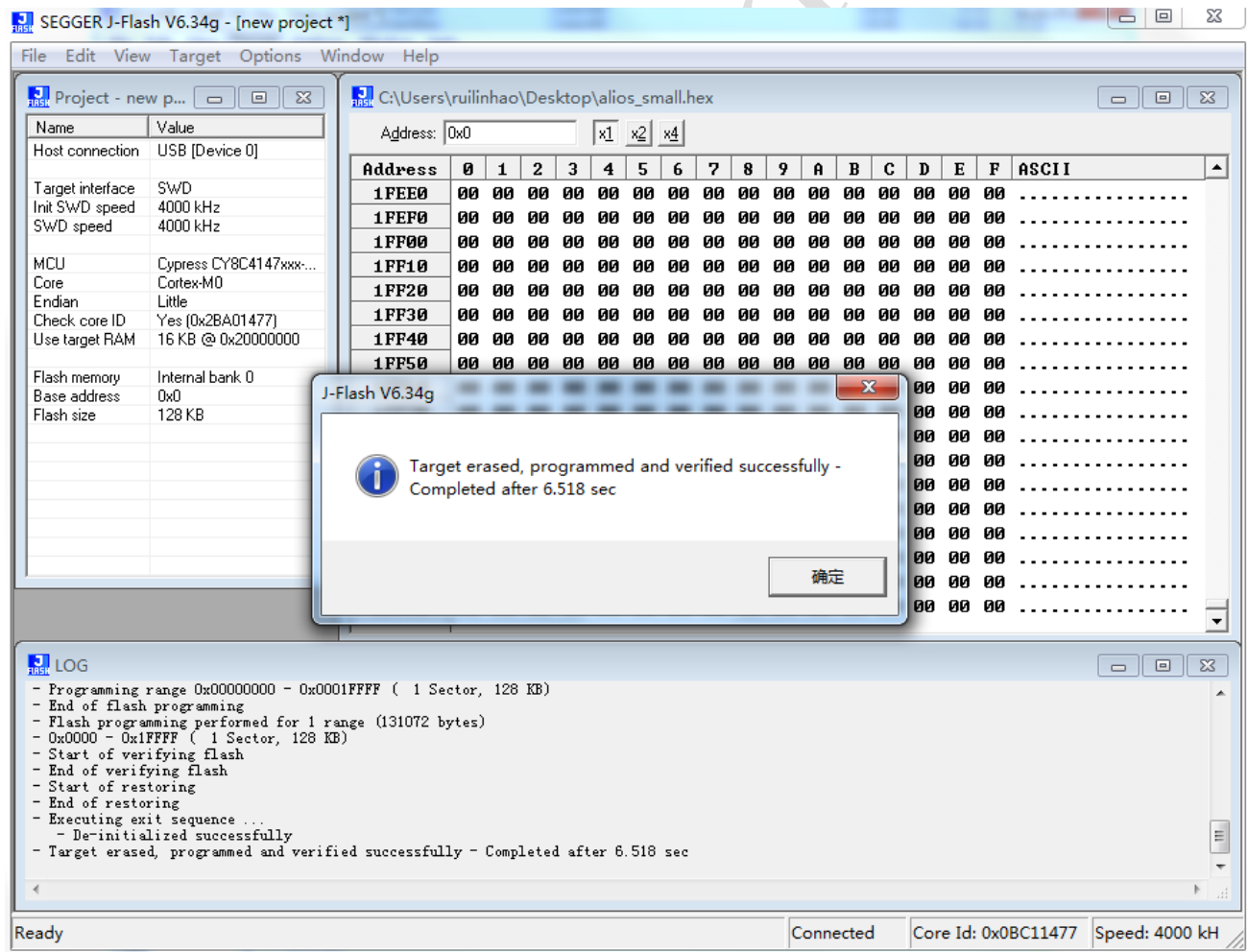
5) 打开修改后的 Hex 文件



6) 开始烧录



7) 烧录完成



4.4 调试

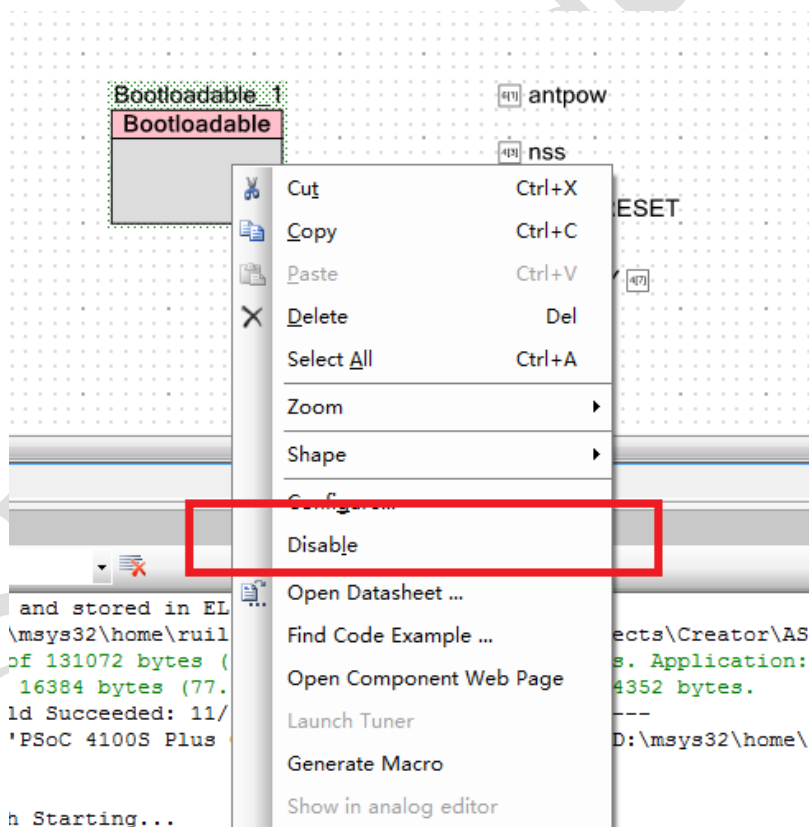
使用 PSOC Creator 进行调试，需要做以下配置：

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

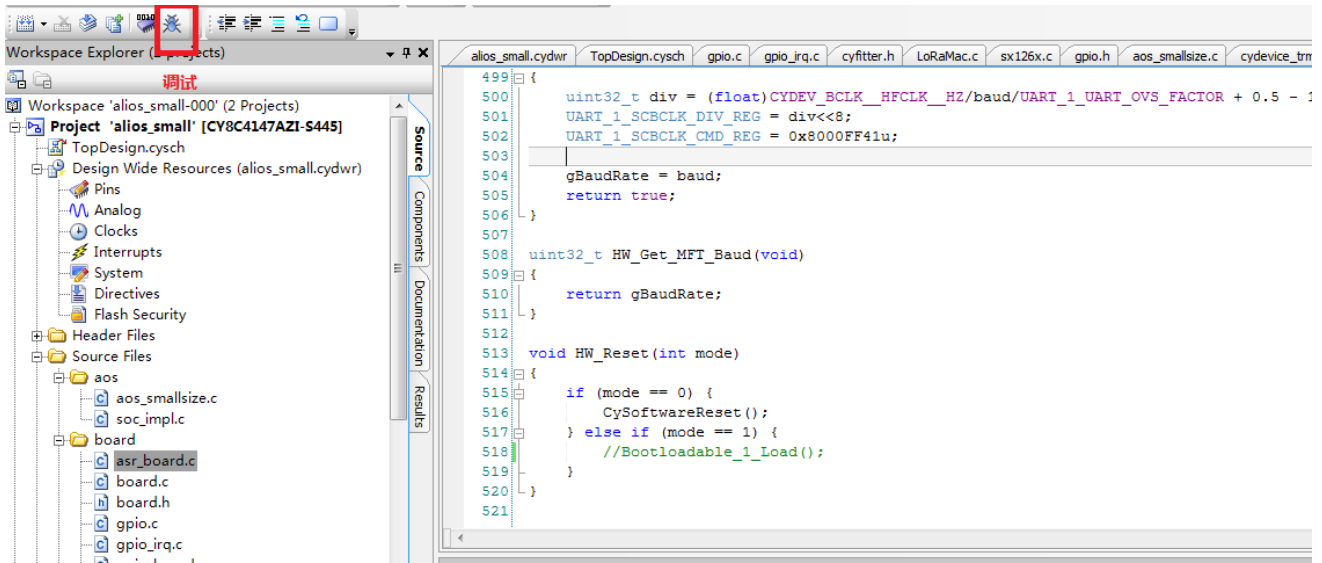


- 2) 暂时去掉 bootloader

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



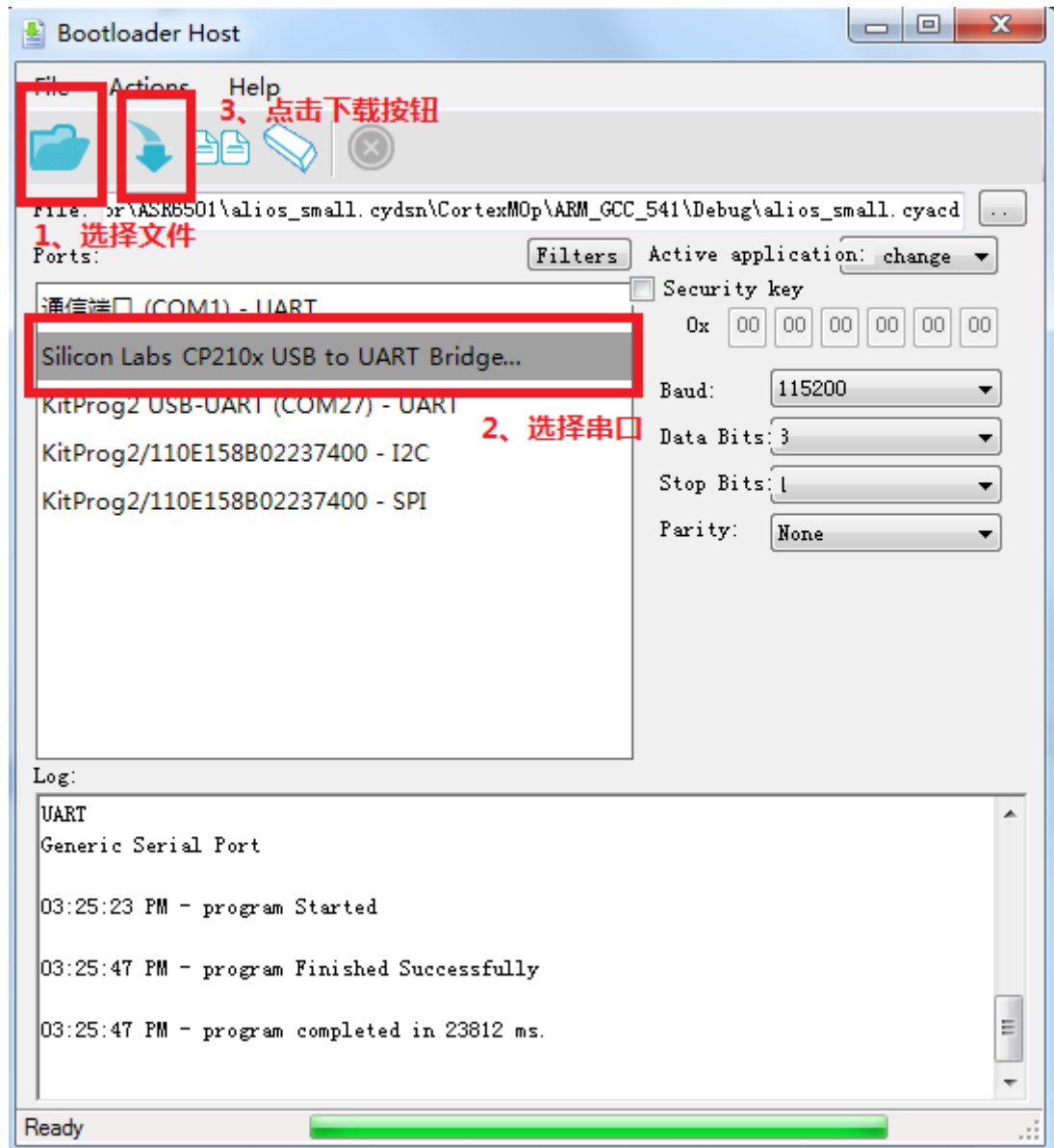
- 4) 重新编译工程
- 5) 选择调试



4.5 UART 升级

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

- 1) 在正常模式输入 AT+IREBOOT=7，使设备进入 bootloader；
- 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_BTLDLDR);
    //}
```

```
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>

4.6 代码阅读

为了便于使用 SourceInsight，至少需要添加以下目录的文件：

```
\platform\arch\arm\armv6m\gcc\m0
\platform\mcu\cy8c4147-aos\
\platform\mcu\cy8c4147-runapp\
\kernel\modules\fs\kv\
\kernel\vcall-aos\
\tools\cli\
\include-aos\
\board-asr6501\
\kernel-rhino-core\
\kernel-rhino-common\
\example-lorawan-loraRun\
\kernel\protocols-lorawan-linkwan\
\kernel\protocols-lorawan-linkwan-region\
\kernel\protocols-lorawan-lora-system-crypto\
\kernel\protocols-lorawan-lora-mac\
\kernel\protocols-lorawan-lora-mac-region\
\kernel\protocols-lorawan\
\kernel\protocols-lorawan-lora-system\
\board-asr6501\
\board-asr6501-inc\
\board-asr6501-src\
\kernel\protocols-lorawan-linkwan-include\
\device-lora-sx126x\
\device-lora-asr6501-lrwan\
```

ASR

5 配置通信

5.1 LinkWan 通信

阿里 LinkWan 网络分为 LinkWan 公网和 LinkWan 认证实验室两部分，两部分的网关和节点不同互通，所以在配置节点信息前，需先确定节点是公网节点还是认证实验室节点。

5.1.1 LinkWan 公网

下面节点 A 为例说明 LinkWan 配置入网的步骤，节点 A 信息：

```
DEVEUI: D896E0FF00000240
APPEUI: D896E0E000005203
APPKEY: 077EE45C6E4564D96D76AE55AFD3AA89
节点类型: ClassA
同异频类型: 同频
网关频组掩码: 0002
```

1) 节点三元组信息配置

```
ASR6501:~# AT+CDEVEUI=D896E0FF00000240
OK
ASR6501:~# AT+CAPPEUI=D896E0E000005203
OK
ASR6501:~# AT+CAPPKEY=077EE45C6E4564D96D76AE55AFD3AA89
OK
```

2) 节点同异频配置

```
ASR6501:~# AT+CULDLMODE=1
OK
```

3) 节点类型配置

```
ASR6501:~# AT+CCLASS=0
OK
```

4) 节点频组掩码设置

使用 AT+CFREQBANDMASK 进行频组掩码配置，掩码每个 bit 代表 1 个频组（8 个频点），最多表示 128 个频点，详细参数见 AT 命令文档。

```
ASR6501:~# AT+CFREQBANDMASK=0002
```

```
OK
```

5) 开始联网

```
ASR6501:~# AT+CJOIN=1,0,8,8
```

```
OK
```

```
ASR6501:~# [1232956]Tx, Power: 17, Band 1, Freq: 472900000,DR: 3, len: 23, duration 206, at 1232947
```

```
[1232969]Start to Join, method 1, nb_trials:8
```

```
[1238176]Rx, Freq 472900000, DR 3, window 1
```

```
+CJOIN:OK
```

```
[1238352]Joined
```

```
[1238388]Tx, Power: 17, Band 1, Freq: 473100000,DR: 3, len: 14, duration 165, at 1238386
```

```
[1239568]Rx, Freq 473100000, DR 3, window 1
```

```
OK+SENT:01
```

```
[1239722]receive data: rssi = -23, snr = 12, datarate = 3
```

```
[1239727]rx, ACK, index 0
```

```
OK+RECV:02,00,00
```

6) 发送数据

```
AT+DTRX=1,2,3,112233
```

```
OK+SEND:03
```

```
ASR6501:~# [1350420]Tx, Power: 17, Band 1, Freq: 472700000,DR: 3, len: 16, duration 165, at 1350418
```

```
[1351600]Rx, Freq 472700000, DR 3, window 1
```

```
OK+SENT:01
```

```
[1351754]receive data: rssi = -17, snr = 11, datarate = 3
```

```
[1351759]rx, ACK, index 1
```

```
OK+RECV:02,00,00
```

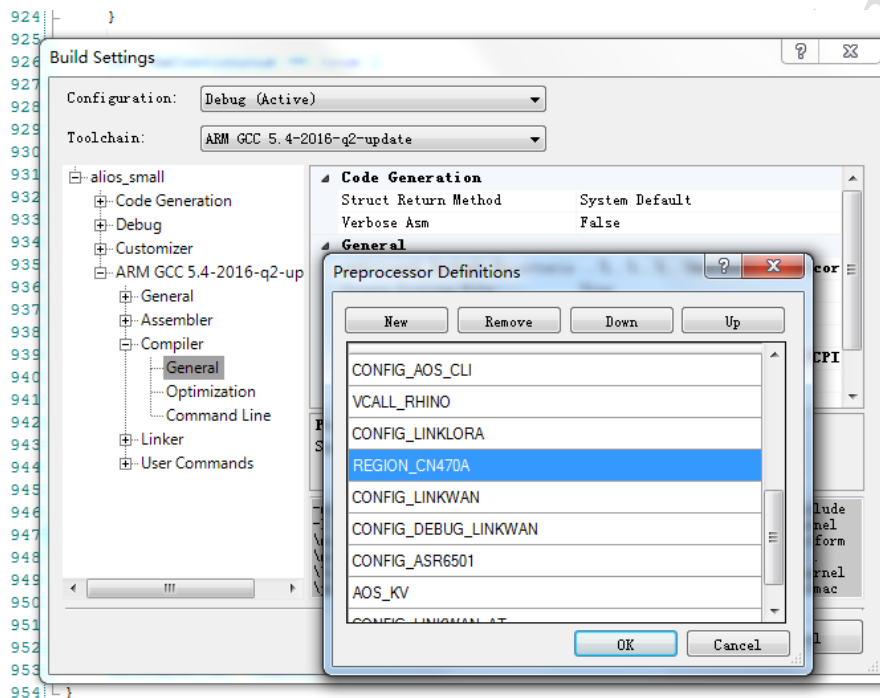
5.1.2 LinkWan 认证实验室

认证实验室的具体配置和操作说明参见《节点对接阿里云操作手册》。

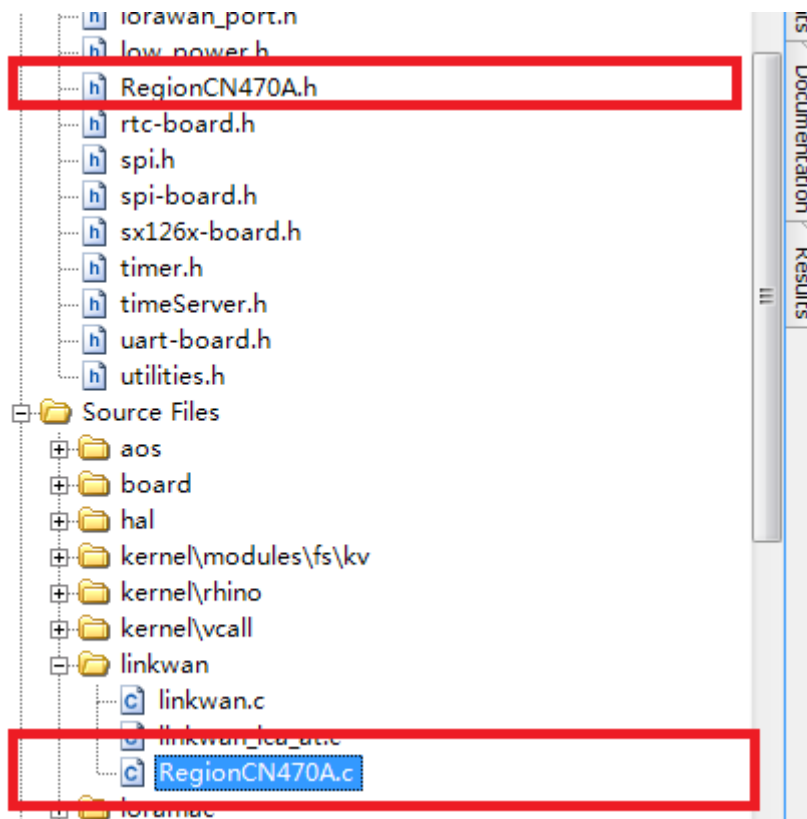
5.2 LoRaWan 通信

ASR6501/ASR6502 SDK 默认支持 LinkWan 通信，如要使用 LoRaWan 通信，需要修改部分代码，以 CN470 为例，修改过程如下：

- 1) 在 Project->Build Setting 的宏定义中将 REGION_CN470A 改成 REGION_CN470。



- 2) 在工程中去掉 CN470A 的文件，并增加 CN470 的文件(kernel\protocols\lorawan\lora\mac\region\)



3) 在 Project->Build Setting 的宏定义中去掉 CONFIG_LINKWAN

完成上述修改后，仿照 LinkWan 通信的过程，使用 AT 指令进行配置和联网。

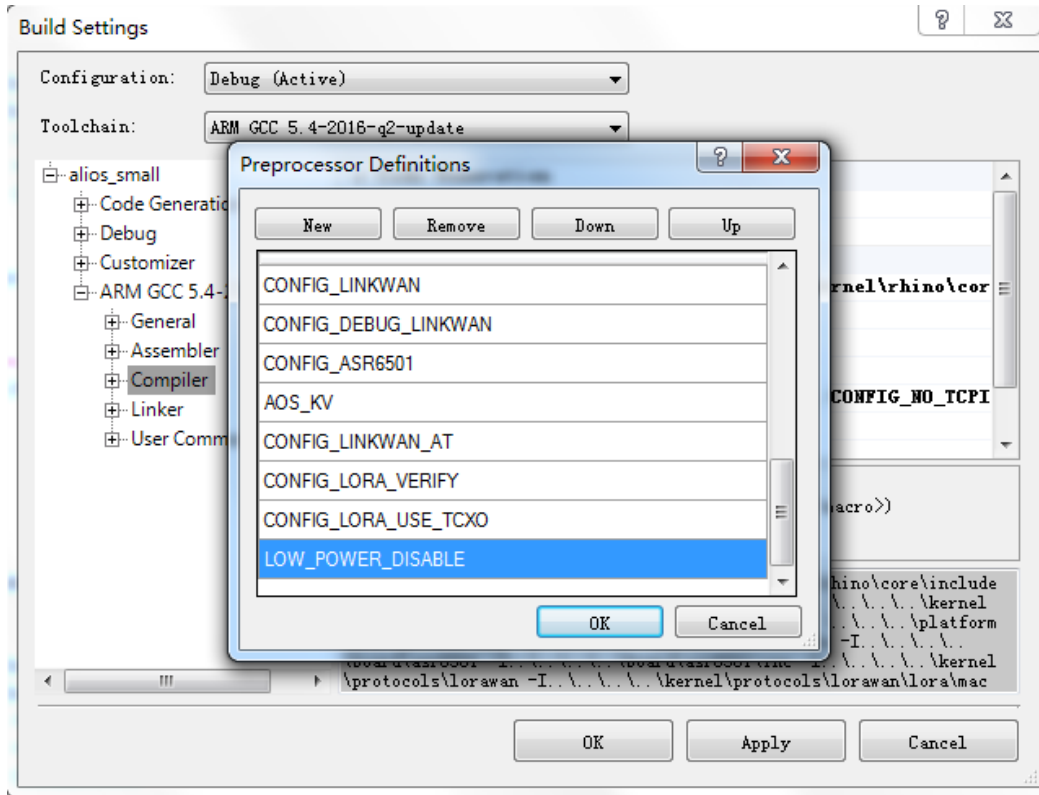
对于使用其它地区的频段如 EU868、AU915、US915、AS923、EU433、IN865 等，按照上述步骤 1、2、3，把宏定义改为对应地区的频段的宏定义，再把相应的文件替换掉以及把 CONFIG_LINKWAN 宏定义去掉。

6 低功耗

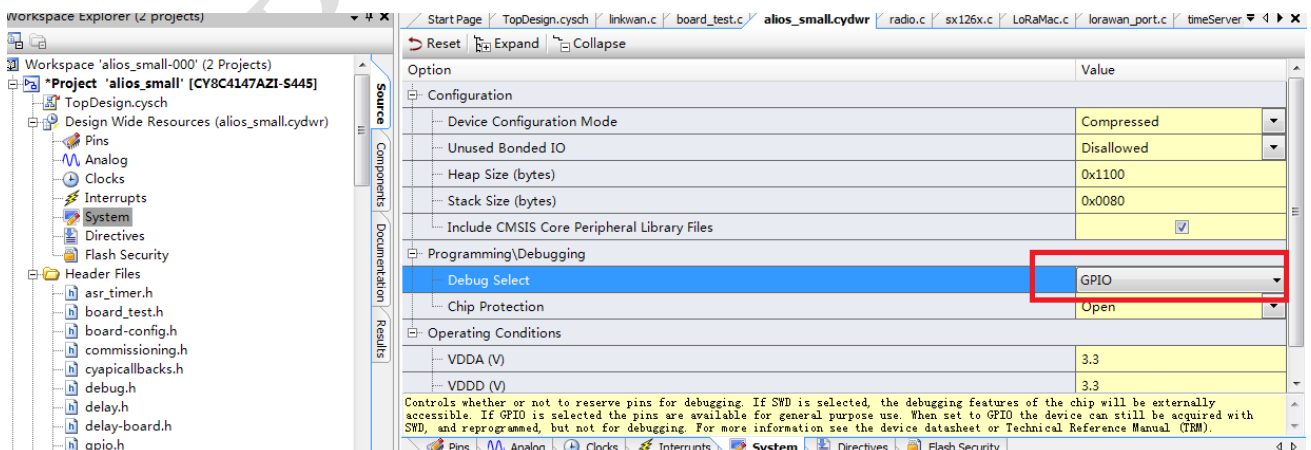
6.1 配置低功耗

SDK 默认关闭低功耗，如需配置低功耗，请按照下面步骤进行配置：

- 1) 在 Project->Build Setting 的宏定义中去除 LOW_POWER_DISABLE 宏



- 2) 关闭调试功能



6.2 低功耗唤醒

- 1) 使用唤醒词进行唤醒，例如：“00000000D0A”(16 进制)；

在 DeepSleep 模式下，可使用任意字符唤醒设备，但由于 40kbps 以上传输时，UART 起始部分字节可能传输错误，AT 命令会返回错误，建议使用“00000000D0A”(16 进制)。

2) 使用 AT+CLPM 命令设备再次进入低功耗；

当低功耗唤醒后，设备将一直处于 Active 状态，需要使用 AT+CLPM=1 使设备再次进入低功耗状态。

6.3 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();

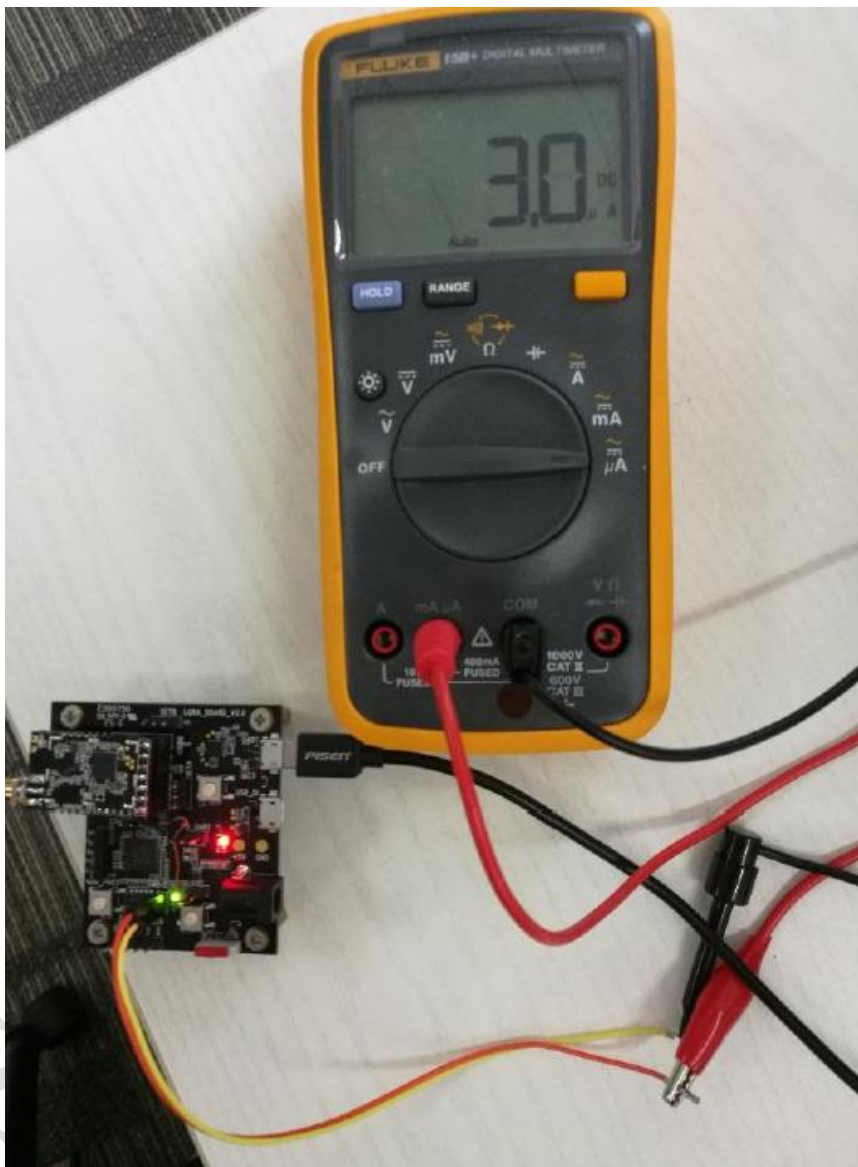
/*restore the IOs drivemode */
SPI_1_mosi_m_SetDriveMode(spi_mosi_mode);
...

SPI_1_Wakeup();
```


7 测试

7.1 功耗测试

7.1.1 硬件连接



7.1.2 DeepSleep 功耗测试

■ 注意事项

- 1) 关闭 DEBUG 模式进行编译，并在烧录完成后拔掉 USB_DL 的连线
- 2) 如果是烧录后测试，请重新上电（插拔给板子供电的 USB 线）

■ 测试命令

AT+CSLEEP=2

■ 参考值

DeepSleep 功耗为 3.1uA 左右, 其中 MCU 2.5uA, sx1262 0.6uA

7.1.3 接收功耗测试

■ 测试命令

AT+CRX=470000000,0

■ 参考值

接收时功耗约为 10.5mA

7.1.4 发送功耗测试

■ 注意事项

请接天线进行测试, 不同的天线可能功耗不同。

■ 测试命令

AT+CTXCW=470000000,22

■ 参考值

22dBm 发送时功耗约为 102mA

7.2 功率测试

■ 测试命令

AT+CTXCW=470000000,22

■ 参考值

22dBm 发送时, 输出功率约为 20.9dBm

7.3 距离测试

■ 测试环境搭建

- 1) 准备两个板子, 一个用于发送, 一个用于接收;
- 2) 在发送的板子上执行下面命令, 则开始每隔 1s 连续发送 (470Mhz, DR0, 22dBm);
AT+CTX=470000000,0,22
- 3) 在接收的板子上执行下面命令开始接收 (470Mhz, DR0);
AT+CRX=470000000,0
- 4) 接收端收到数据包后会打印已收到包数, 发送总包数 (从接收到第一个包开始), 数据包内容, rssi 和 snr, 如下:

[126/126]Received: 126, rssi = -15, snr = 4

数据内容

发送包数

收到包数

RSSI

SNR

4) 注意事项

如发送端重新发送，请重启接收端，否则统计发送包数会出问题。

5) 参考值

发射节点位于 10 号楼 9 楼西南角窗户口，窗户打开，接收节点从张江创新园出发，沿科苑路向南，市区环境内点对点测试：

TX Power 17dBm，通讯距离 4.3km；

TX Power 22dBm，通讯距离 6.3km；

17dBm 距离测试：



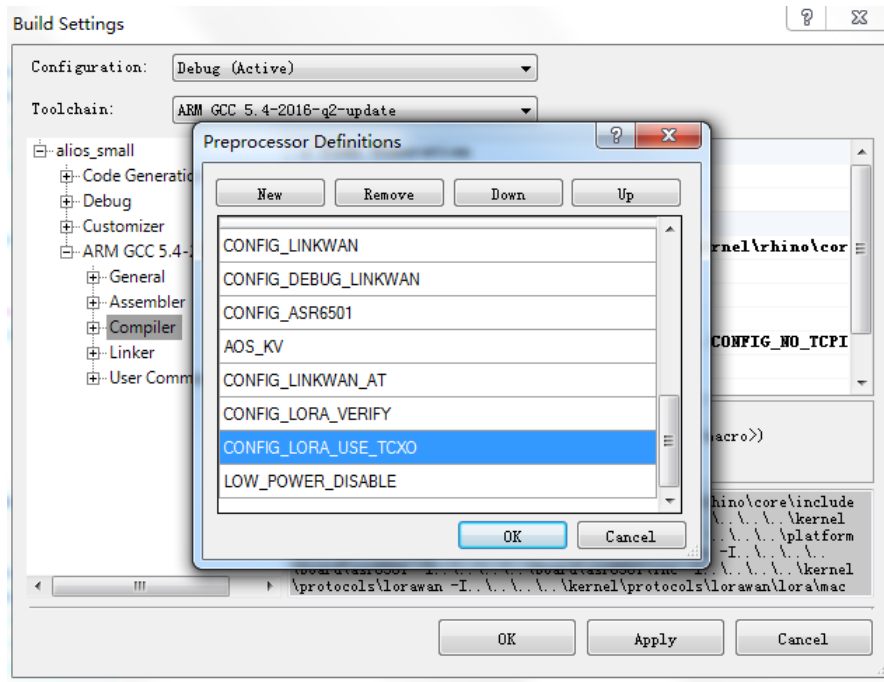
22dBm 距离测试:



8 Q&A

8.1 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。



8.2 如何配置自动联网？

为测试方便，默认进入 AT 命令行进行配置，如果需要设备上电后自动联网，有两种办法：

- AT 命令配置

AT+CJOIN=1,1,8,8

使用 AT+CJOIN 命令即可以完成自动联网的功能，具体参数描述见 AT 命令文档。

- 代码中修改

在 linkwan.c 的 init_lwan_configs 函数中修改 LWAN_DEV_CONFIG_DEFAULT。

8.3 如何在代码中更改设备信息？

- AT 命令配置

OTA 模式使用 AT+CDEVEUI, AT+CAPPEUI 和 AT+CAPPKEY 三个命令，ABP 模式使用 AT+CDEVADDR, AT+CNWKSKEY 和 AT+CAPPSKEY 三个命令。

- 代码中修改

设备三元组信息，请修改 init_lwan_configs 函数中的 LWAN_DEV_KEYS_DEFAULT。

8.4 如何使用 ABP 模式?

- AT 命令配置
使用 AT+CDEVADDR, AT+CNWKSKEY 和 AT+CAPPSKEY 三个命令修改设备信息, 并使用 AT+CJOINMODE 修改入网方式。
- 代码中修改
在 init_lwan_configs 函数中修改 LWAN_DEV_CONFIG_DEFAULT 和 LWAN_DEV_KEYS_DEFAULT。

8.5 如何配置同、异频节点?

- AT 命令配置
AT+CULDLMODE=2
- 代码中修改
在 init_lwan_configs 函数中修改 LWAN_DEV_CONFIG_DEFAULT。

8.6 如何配置 CLASS B 节点?

- AT 命令配置
在 JOIN 前, 使用 AT+CCLASS=1 命令更改设备为 CLASSB
- 代码中修改
在 init_lwan_configs 函数中修改 LWAN_DEV_CONFIG_DEFAULT。

8.7 如何配置 Heap Size?

目前设置的 heap size 较小, 如果工程中使用较多的 heap 内存, 请在这里将 heap size 调大。

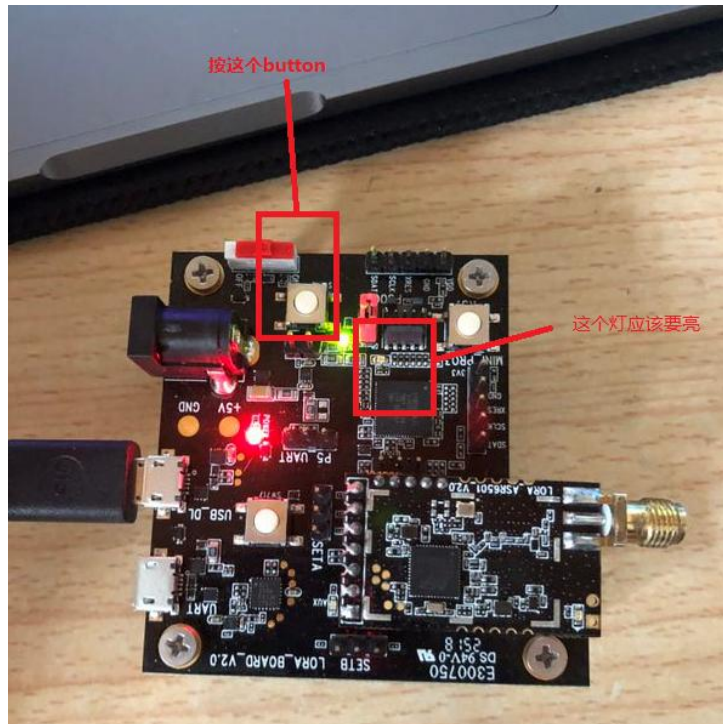


另外, 如果要将所有剩余的 ram 作为 heap, 可以在 cm0plusgcc.ld 中定义 __cy_heap_size 变量, 然后在 soc_impl.c 中将 g_mm_region 变量中的 CYDEV_HEAP_SIZE 替换为 __cy_heap_size。

注意: SDK 中设置的是最小 heap size, 请在具体项目实施中修改 heap size, 使剩余内存尽可能被利用。

8.8 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



8.9 如何加密三元组信息？

请使用 AT+CKEYSPROTECT 命令对三元组信息进行加密，具体使用方法见 AT 命令文档

8.10 AsrLib.a 与 AsrLib_small.a 有什么差异？

AsrLib_small.a 中去掉了 ClassB 与 AT 测试指令的代码，code size 可以更小一点。

8.11 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。

9 参考资料

9.1 ALIOS 资料

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

9.2 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://loro-alliance.org/lorawan-for-developers>

9.3 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

- Cypress 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/psoc-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>