

# 智慧安防案例设备开发指导手册

## （基于 NB-IoT）



扫一扫，关注我们，获取更多资料

（教程若有误，请与小助手联系，微信号：BearPi\_Helper）

Revision 1.0

2020 年 4 月 19 日

目录

首页-技术文档.....	1
前言 .....	3
1. 基于 IoT Studio 设备开发 .....	4
2. 基于 LiteOS Studio 设备开发.....	4
2.1. LiteOS Studio 工程导入.....	4
2.2. LiteOS Studio 程序编译及烧录 .....	5
3. 基于 MDK 5 设备开发.....	9
3.1. MDK v5 工程导入 .....	9
3.2. MDK v5 程序编译及烧录.....	10
技术支持获取.....	16

## 前言

本文旨在指导开发者进行基于小熊派 IoT 开发板的程序开发，为了适应不同开发者的需求，小熊派 IoT 开发板目前提供适配了（IoT Studio、LiteOS Studio、MDK v5）三种编译器的程序，开发者可任选其中一种编译器进行开发，开发过程中不建议切换编译器。目前基于 IoT Studio 的代码还不支持 2G 通信板，需要使用 2G 的开发者请使用 LiteOS Studio 或者 KEIL 5 进行开发。编译器请使用资料包中提供版本，以免因为版本不同出现未知的问题，影响开发进度。在开发前请根据自己选择的编译器的安装指导手册完成编译器的安装。

## 1. 基于 IoT Studio 设备开发

智慧安防目前还不支持 IoT Studio 开发，请暂时使用 LiteOS Studio 或者 MDK 5 进行开发。

## 2. 基于 LiteOS Studio 设备开发

基于 LiteOS Studio，将程序样例烧录进小熊派开发板。

### 2.1. LiteOS Studio 工程导入

**步骤1** 打开 LiteOS Studio 软件，在首页上单击“导入其他嵌入式工程（gcc）”，如图 2-1 所示。

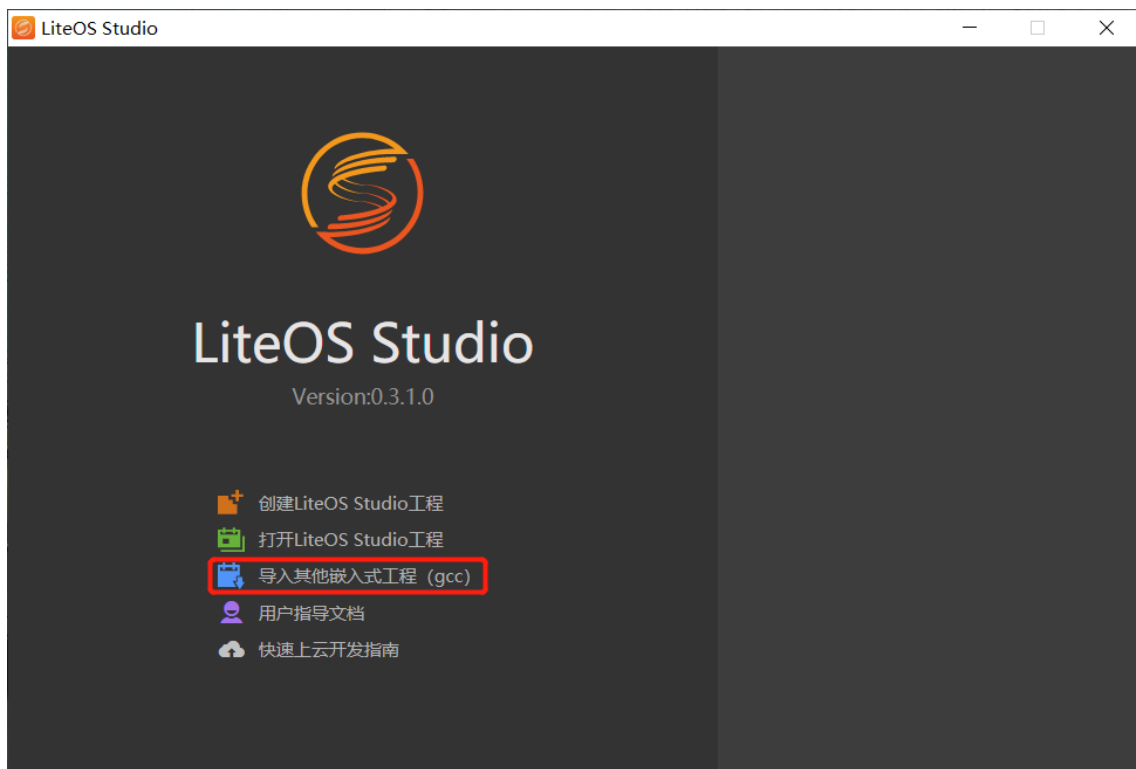


图 2-1 导入工程

**步骤2** 在弹出导入界面，选择需要导入的工程目录及对应的 MCU 类型。此处需要导入小熊派资料包的“\03 案例及代码\05\_E53\_IS1\_智慧安防”路径下的代码，**此处先要将工程代码移至非中文路径下，也不可有空格等特殊字符**。小熊派开发板使用的 MCU 为 STM32L431RCT6，所以此处 MCU 类型选择 STM32L431RC，单击“完成”按钮导入相应工程，如图 2-2 所示，工程目录请根据实际情况选择。

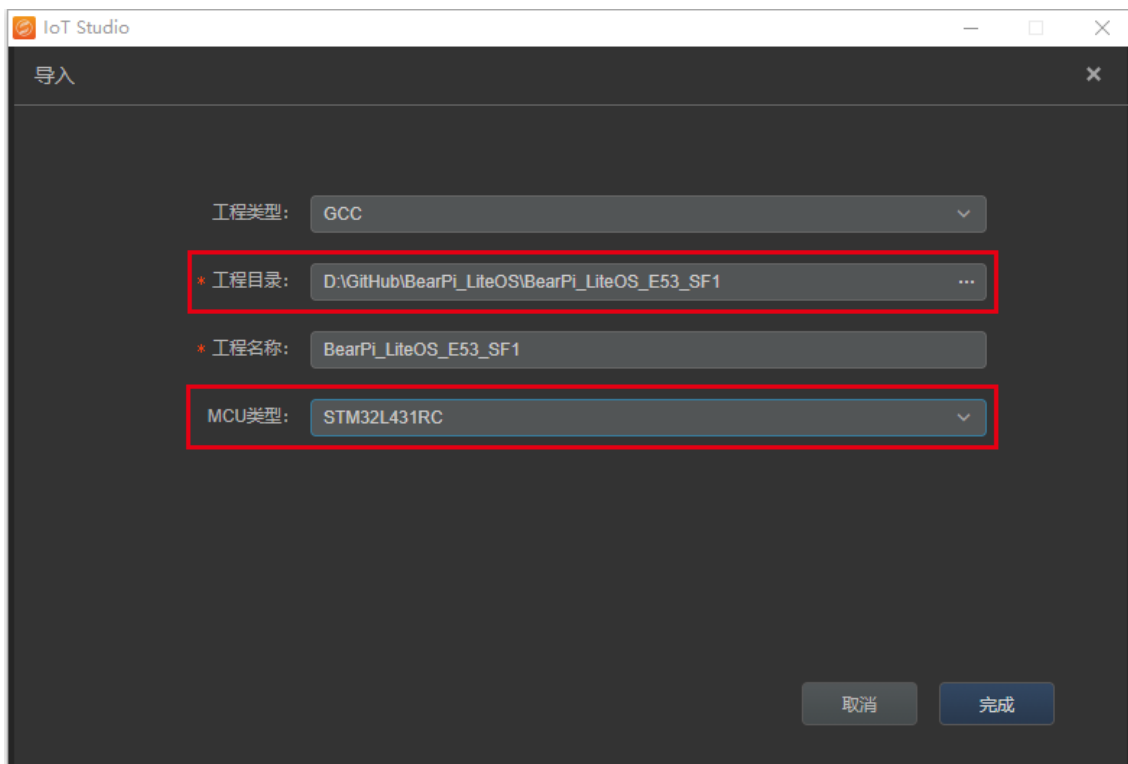



图 2-2 设置工程参数

## 2.2. LiteOS Studio 程序编译及烧录

**步骤1** 在程序编译之前需要对工程进行如下配置：

1. 在打开的 STM32L431xx 工程中，单击工具栏中的 ，进行工程配置。
2. 单击“编译输出”，“输出目录”选择当前工程下的 build 输出目录，具体路径请根据实际情况修改，其他参数默认配置，如图 2-3 所示。

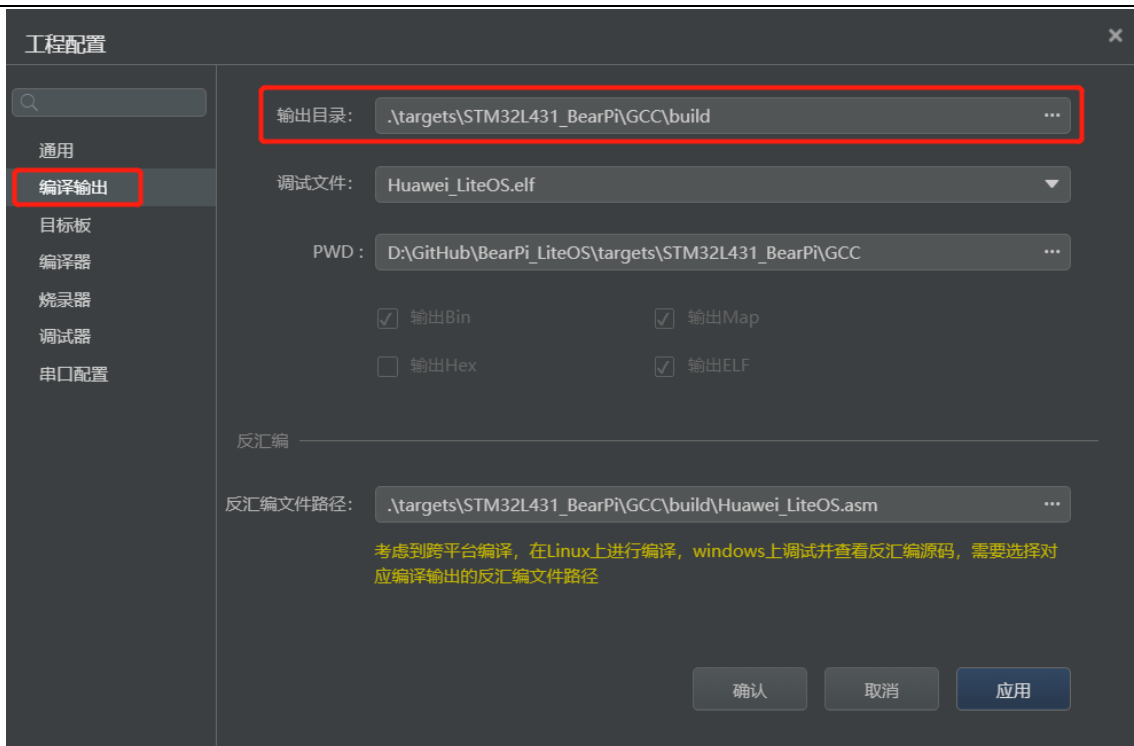


图 2-3 设置输出目录

- 单击“编译器”，“Makefile 脚本”选择当前工程下的主 Makefile，具体路径请根据实际情况修改。Make 参数”可配置为“-j8”，如图 2-4 所示。

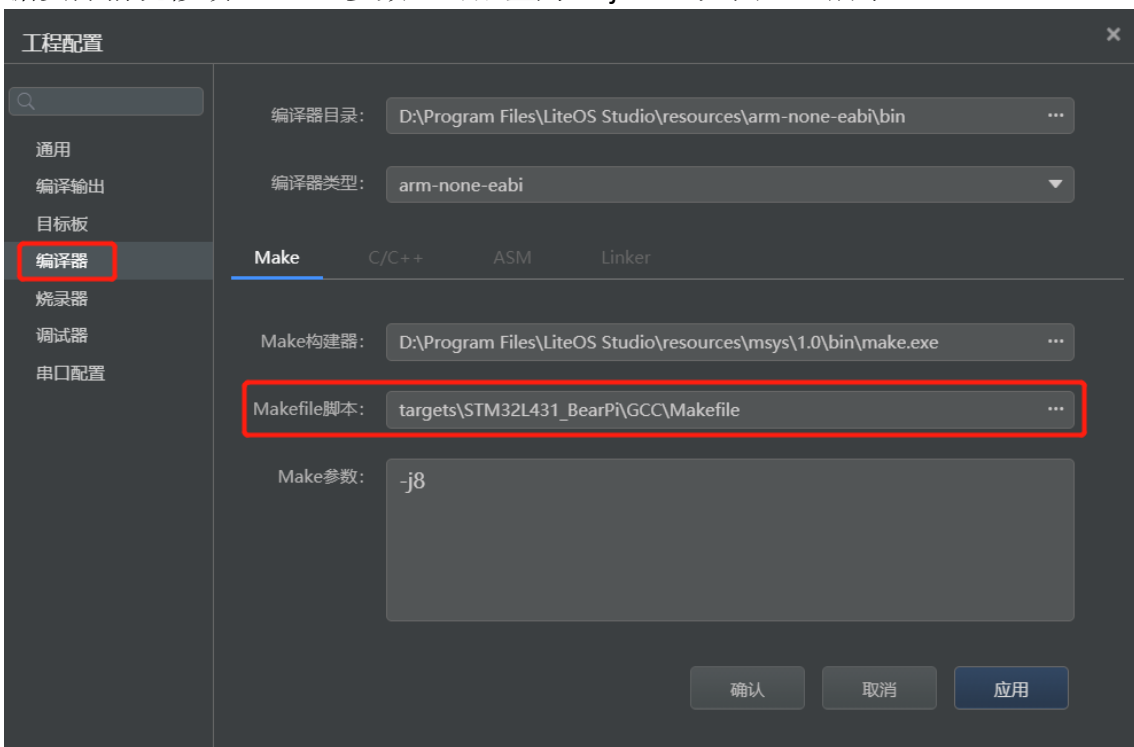


图 2-4 设置编译脚本

4. 单击“烧录器”，烧录方式选择“STLink/V2”，其他参数默认配置，如图 2-5 所示。

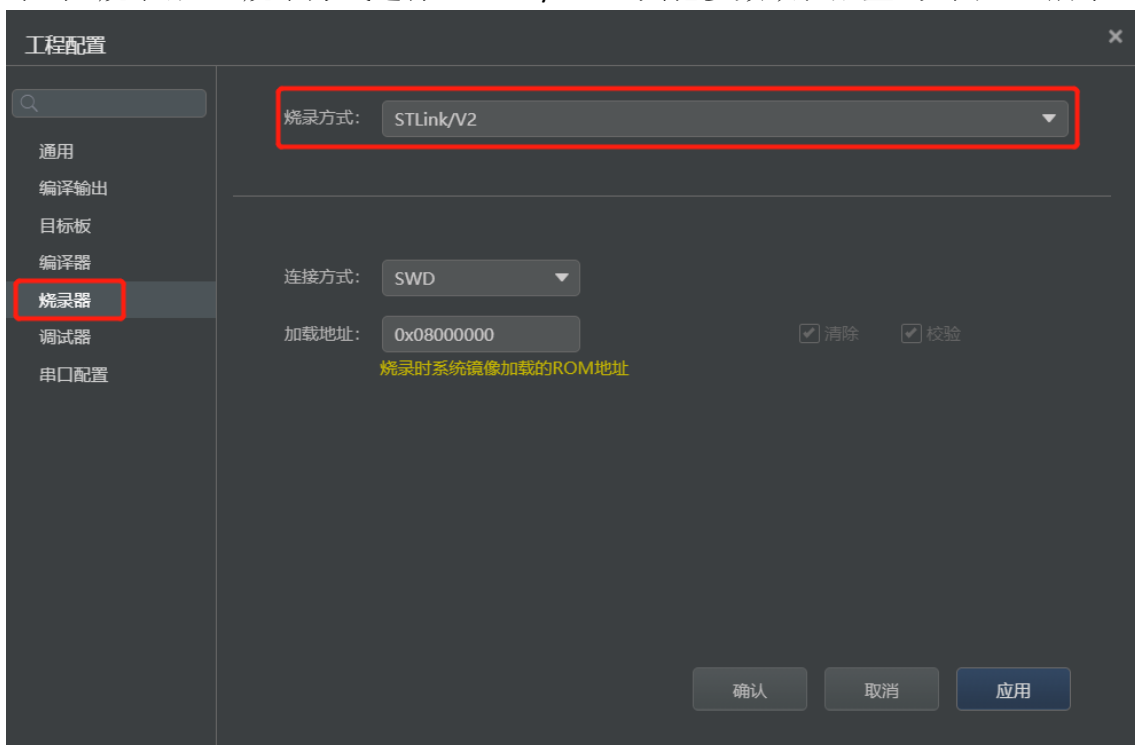


图 2-5 设置烧录器参数

5. 单击“调试器”，调试方式选择“STLink/V2”，其他参数默认配置，如图 2-6 所示。

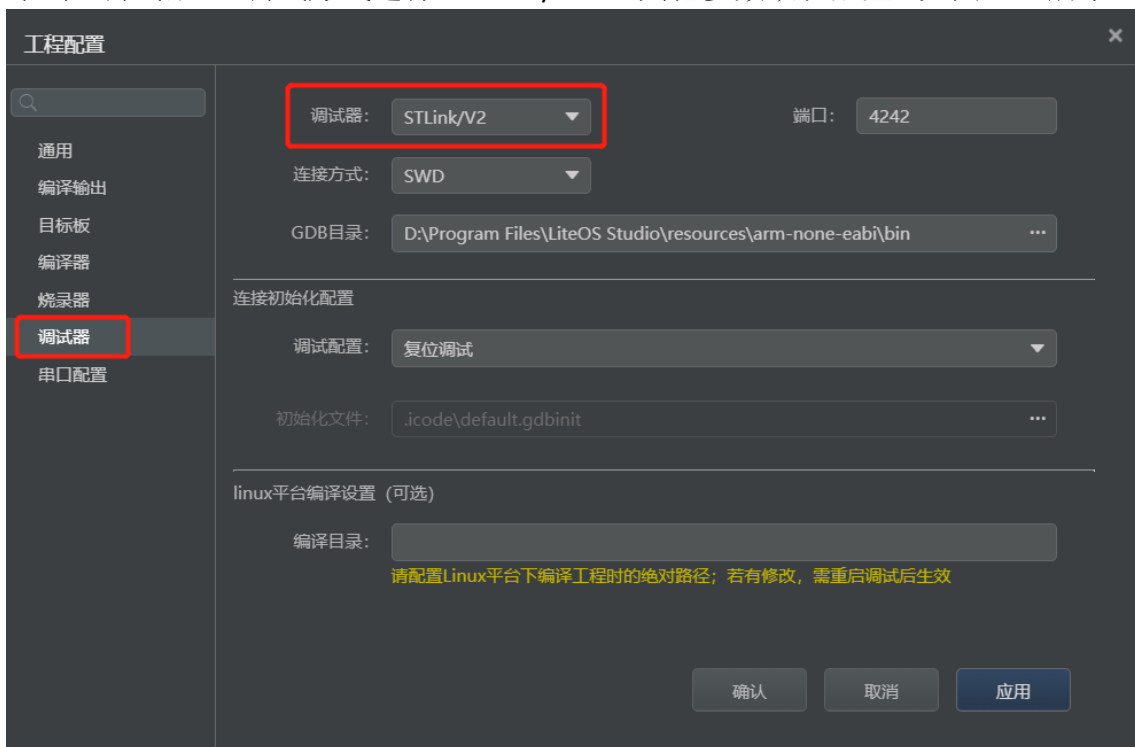



图 2-6 设置调试器参数

**步骤2** 单击工具栏中的，对当前工程进行编译。编译成功后，在控制台输出面板中显示“编译成功”，如图 2-7 所示。

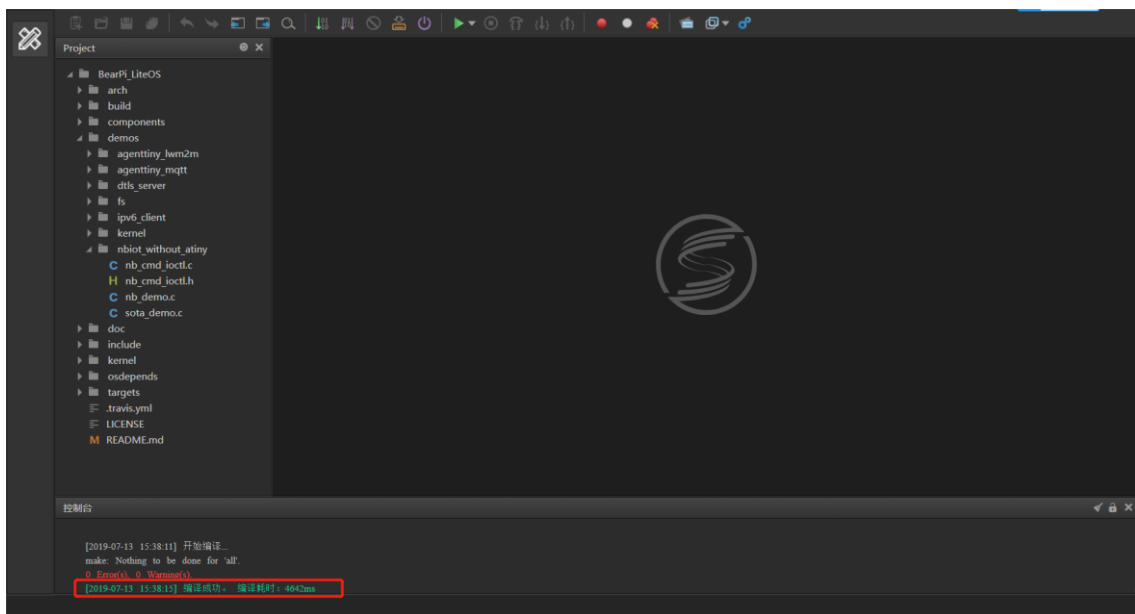



图 2-7 编译成功

**步骤3** 连接好开发板，单击工具栏中的，即可将已编译好的程序烧录至开发板，如图 2-8 所示。

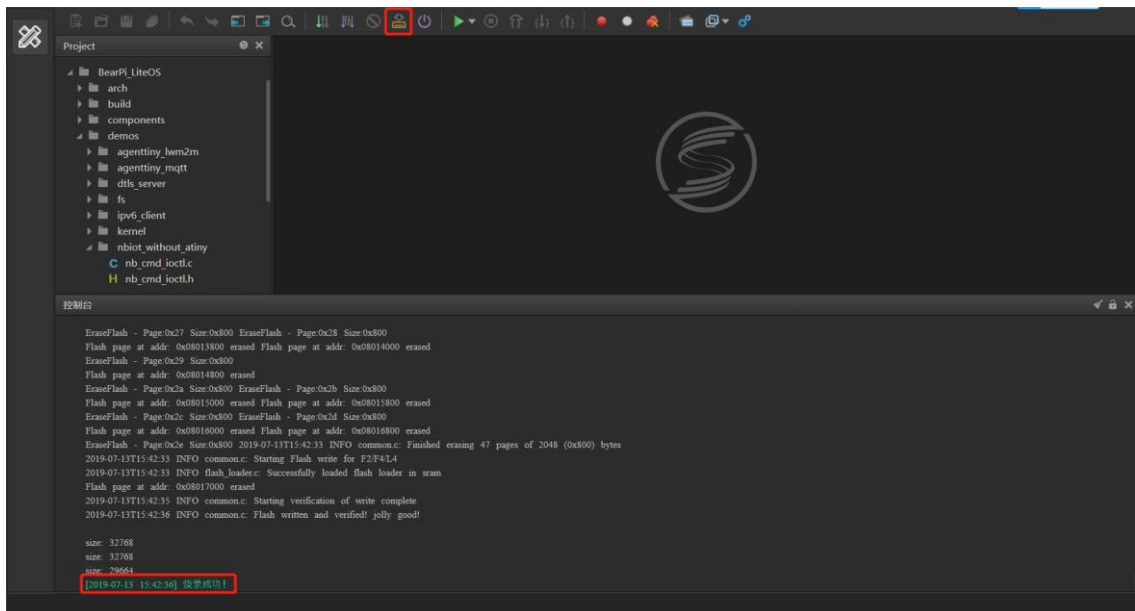


图 2-8 烧录成功

----结束



### 3. 基于 MDK 5 设备开发

基于 MDK v5，将程序样例烧录进小熊派开发板。

#### 3.1. MDK v5 工程导入

**步骤1** 打开 MDK v5 软件，在首页上单击“Project”→“Open Project”，如图 3-1 所示。

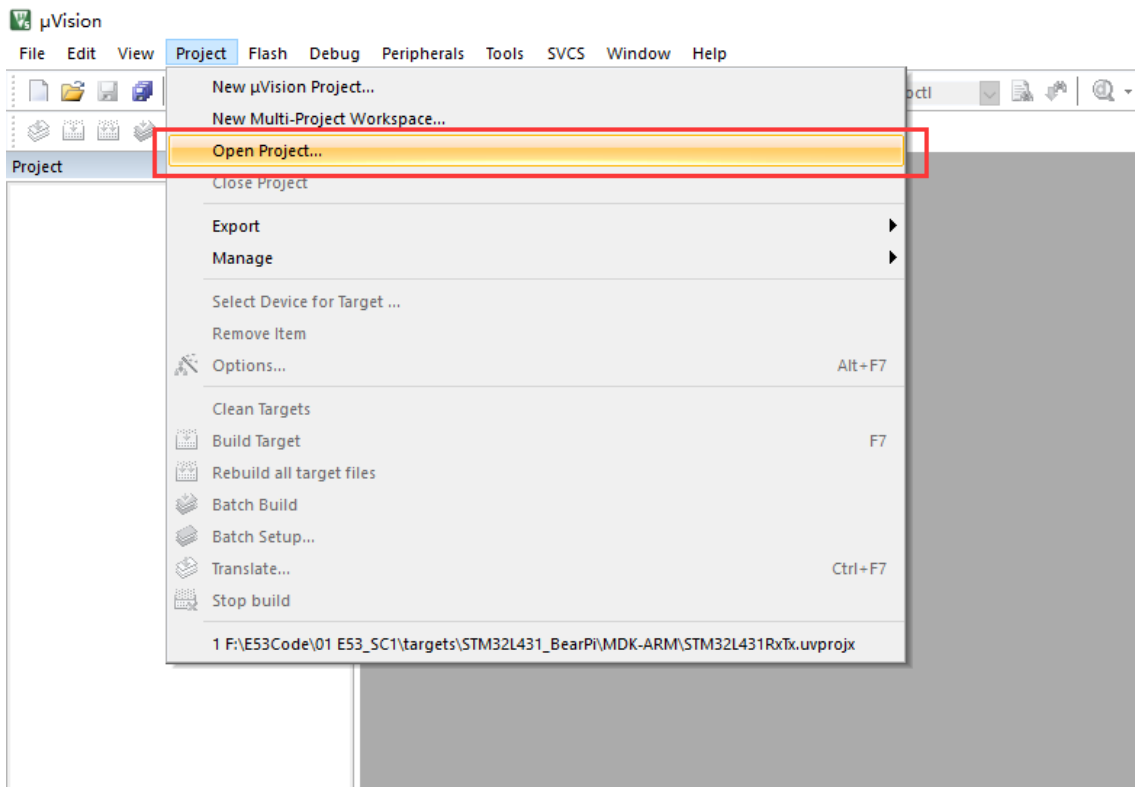


图 3-1 打开工程

**步骤2** 在弹出“Select Project File”界面，选择需要导入的工程目录。此处需要导入小熊派资料包的“\03 案例及代码\05\_E53\_IS1\targets\STM32L431\_BearPi\MDK-ARM”路径下的工程，**此处先要将工程代码移至非中文路径下**。单击“完成”按钮导入相应工程，此处也可直接在文件夹中双击对应工程文件打开工程，如图 3-2 所示。

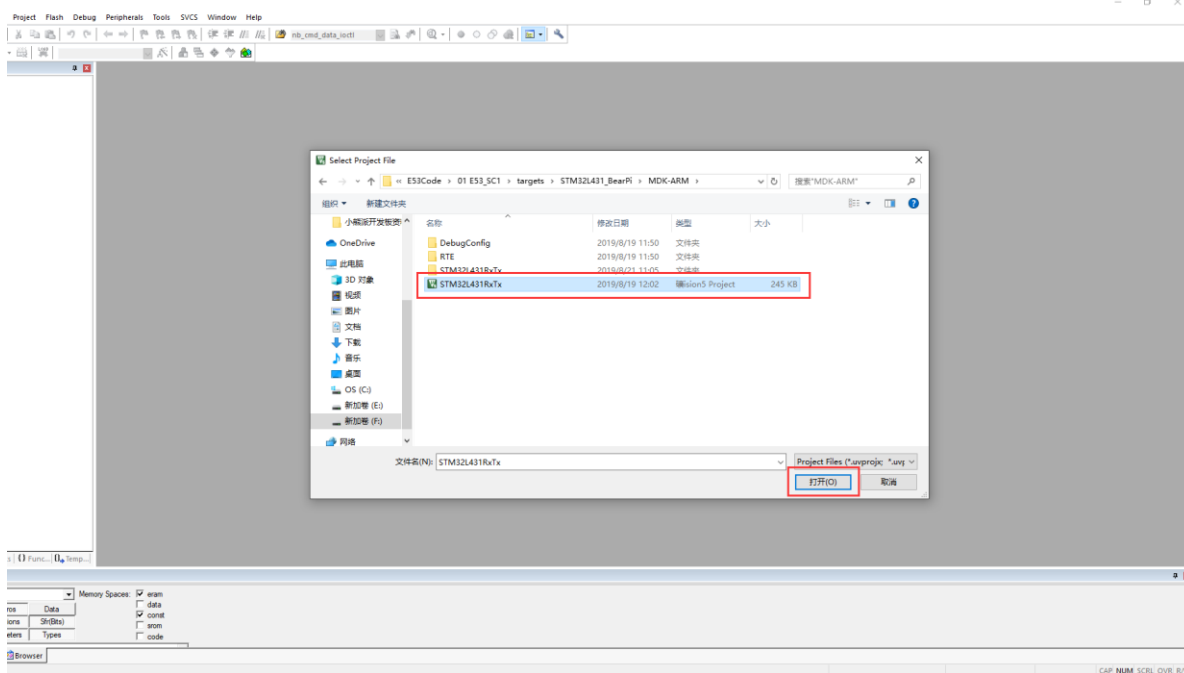


图 3-2 导入工程

## 3.2. MDK v5 程序编译及烧录

**步骤1** 在程序编译之前需要对工程进行如下配置：

1. 打开 MDK 软件后，单击工具栏中按钮进入工程配置界面图标，如图 3-3 进入工程配置界面

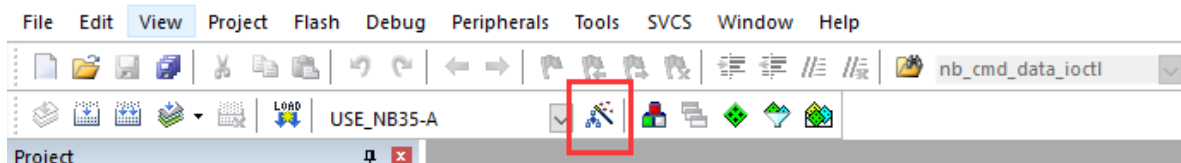


图 3-3 进入工程配置界面

2. 选择“Debug”选项，进入仿真器设置界面，如图 3-4 进入仿真器设置界面所示。

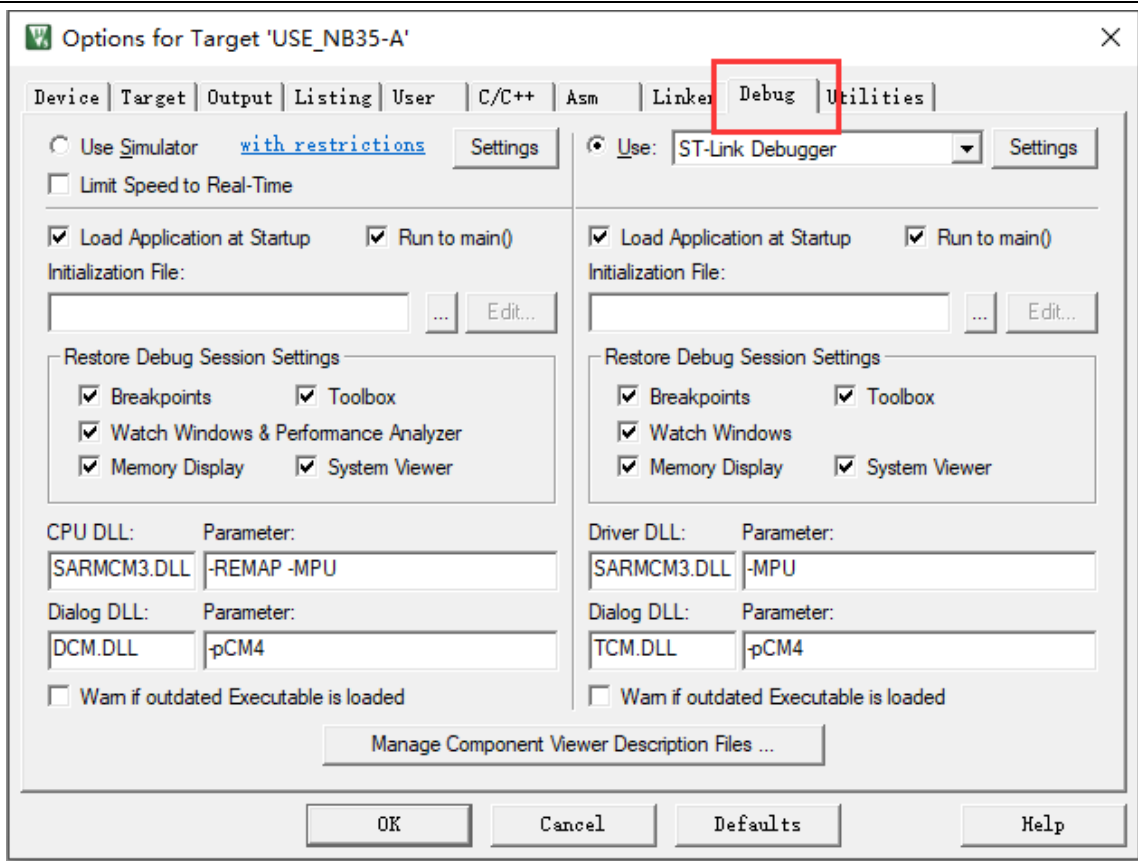


图 3-4 进入仿真器设置界面

3. 下拉仿真器选择列表，选择“ST-Link Debugger”选项，并勾选右侧“Use”，单击“Settings”按钮进入 ST-Link V2 仿真器配置界面，如图 3-5 所示。

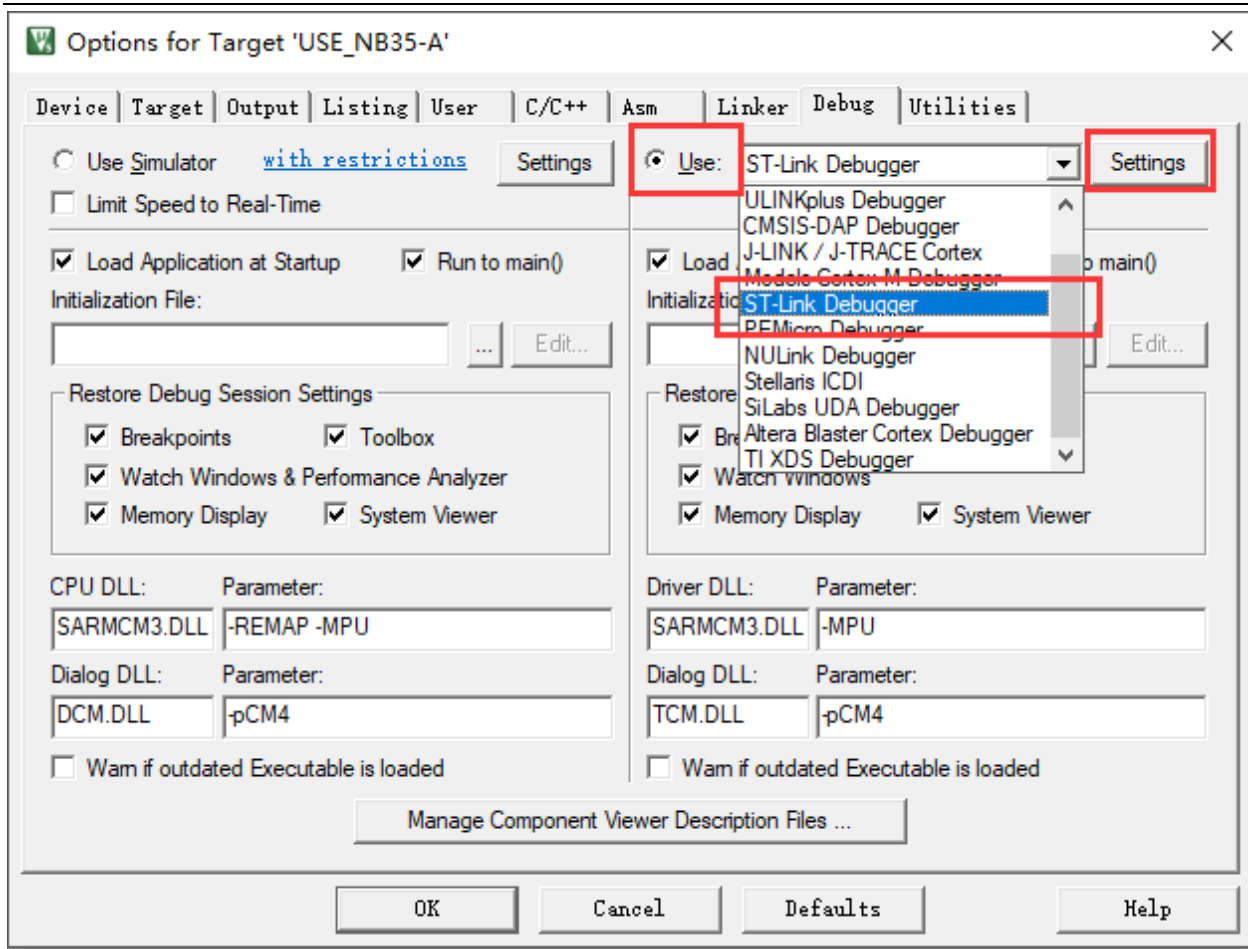


图 3-5 选择 ST-Link Debugger

4. 小熊派开发板设计的程序烧录方式为 SWD 方式，此处“Unit”选项选择“ST-Link/V2”，且“Port”选项选择“SW”，并确认右侧框内是否检测出 SW 设备，如图 3-6 所示。若未检测出 SW 设备，请检查设备连接是否正确。

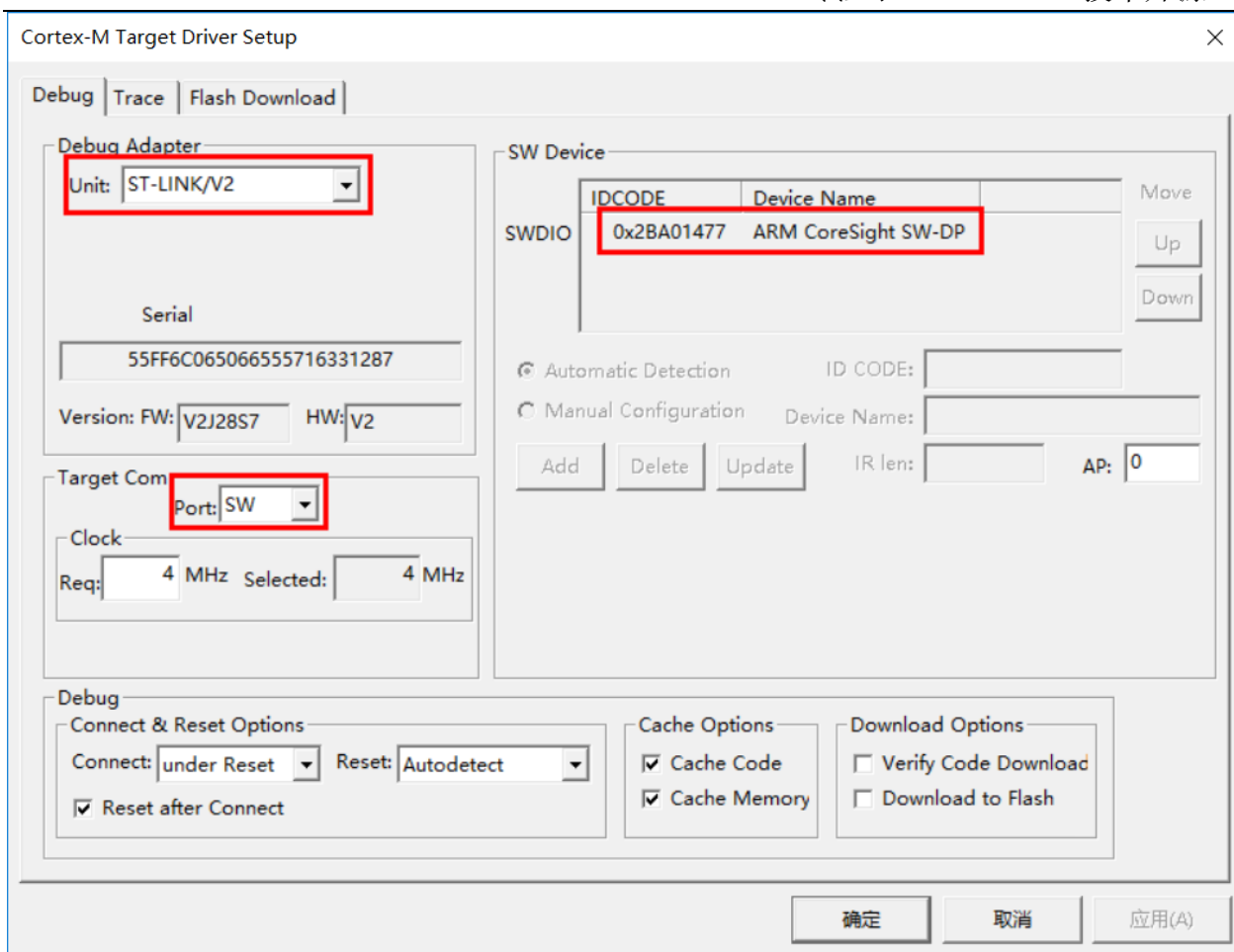


图 3-6 配置 ST-Link

5. 在上一步设置完成之后单击顶部选项栏中的“Flash Download”按钮，进入 Flash 算法设置界面，如图 3-7 所示。这里 MDK5 会根据新建工程时选择的目標器件，自动设置 Flash 算法。小熊派开发板使用的单片机为 STM32L431RCT6，Flash 容量为 256K 字节，所以 Programming Algorithm 里面默认会有 STM32L4xx 256KB Flash 算法。另外，如果这里没有 Flash 算法，可以单击“Add”按钮打开 Flash 算法选择对话框，在此对话框中选择“STM32L4xx 256 KB Flash”算法并单击“Add”按钮完成算法添加。最后，选中 Reset and Run 选项，以实现在编程后自动运行，其他默认设置即可。设置完成之后单击“应用”按钮保存设置。

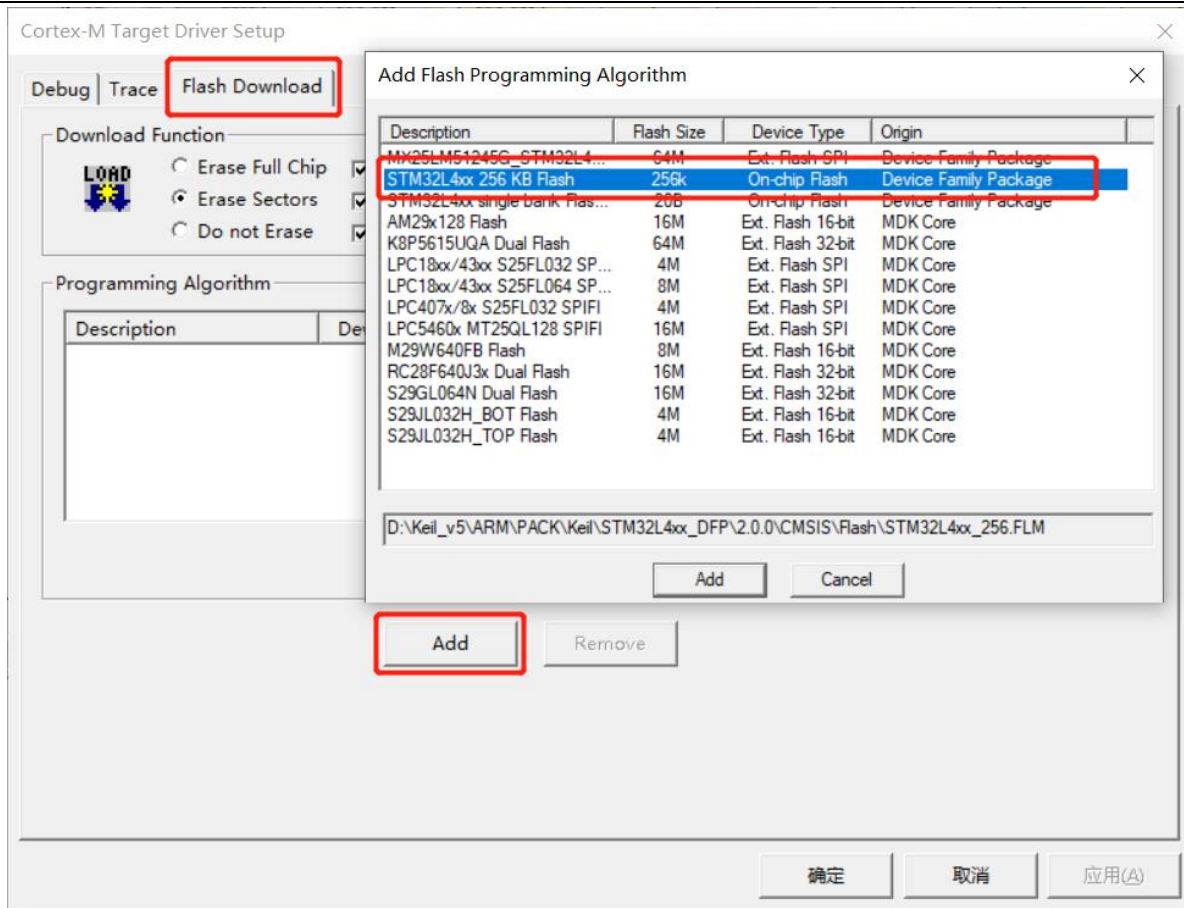


图 3-7 选择 Flash 算法

**步骤2** 工程配置完后，单击工具栏中的“Build”按钮，如图 3-8 所示，对当前工程进行编译。编译成功后，在控制台输出面板中显示如图 3-9 所示字样。

#### 说明

如果编译失败，请检查软件是否激活，可百度 MDK 5 激活教程，进行激活。

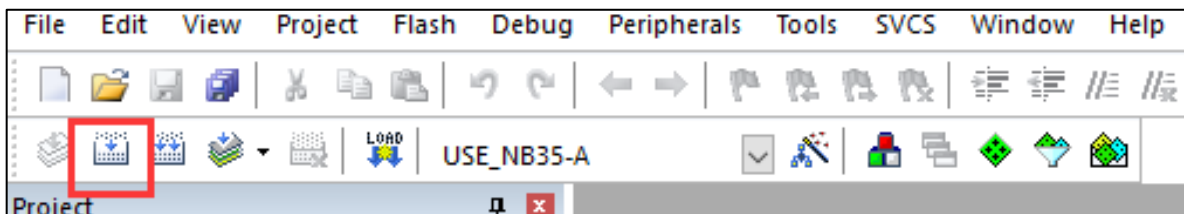


图 3-8 编译

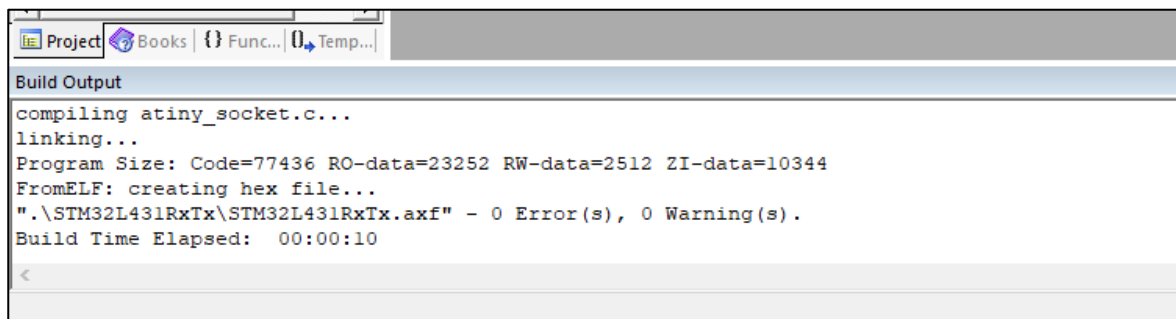



图 3-9 编译成功

**步骤3** 连接好开发板，单击工具栏中的  按钮，即可将已编译好的程序烧录至开发板，如图 3-10 所示。

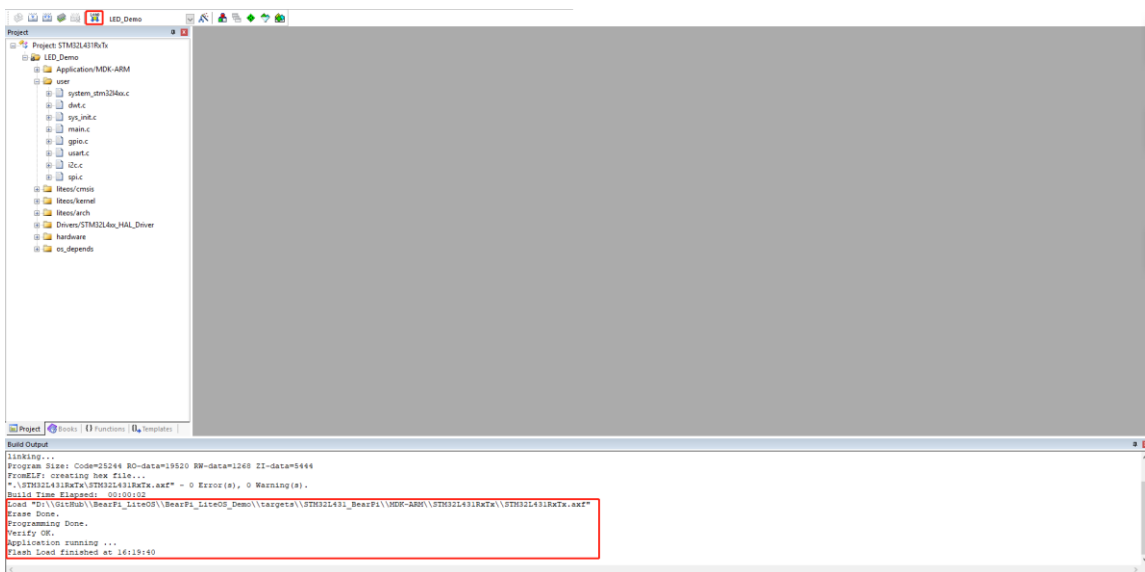


图 3-10 烧录成功

----结束

## 技术支持获取

若学习及开发过程中有任何问题，可以进入华为 IoT 论坛进行提问，或者添加小助手获取帮助。



华为 IoT 论坛-小熊派版块



小熊派小助手微信号: BearPi\_Helper