

类别	内容
关键词	Easy-RT1011
摘要	介绍 Easy-RT1011 Rev B 开发板，及其快速入门指南。

修订历史

版本	日期	原因
V0.0.90	2019/11/08	创建文档
V1.0.00	2019/11/28	修订文档格式： 1. 重新编排第三章例程使用指南； 2. 修订文中的图片大小和表格格式。
V1.0.01	2023/03/23	更新文档模板

目 录

1. Easy-RT1011 开发套件简介.....	1
1.1 概述.....	1
1.1.1 i.MX RT1010 跨界处理器	1
1.1.2 特性.....	1
1.2 Easy-RT1011 开发套件.....	2
1.2.1 功能框图.....	3
1.2.2 板级资源.....	3
1.2.3 接口说明.....	4
2. 资料目录结构说明.....	9
3. 例程使用指南.....	11
3.1 JLINK 仿真器.....	11
3.1.1 概述.....	11
3.1.2 特点.....	11
3.2 硬件连接.....	11
3.3 例程简介.....	11
3.4 KEIL 工程配置选择与编译.....	13
3.4.1 选择配置标签.....	13
3.4.2 编译程序.....	13
3.5 KEIL 软件调试.....	14
3.5.1 工程配置.....	14
3.5.2 仿真调试.....	15
3.5.3 调试工具说明.....	16
3.6 程序下载.....	16
4. MCUXpresso IDE 使用指南.....	18
4.1 MCUXpresso IDE 简介.....	18
4.2 SDK 的获取及导入.....	18
4.2.1 获取 RT1011 的 SDK.....	18
4.2.2 向 IDE 导入 RT1011 的 SDK.....	20
4.3 工程的创建.....	21
4.3.1 导入模板工程.....	21
4.3.2 新建工程.....	23
4.4 调试指南.....	24
4.4.1 编译工程.....	24
4.4.2 进入调试界面.....	26
4.4.3 调试代码说明.....	27
4.4.4 启动文件配置.....	28
4.5 固件生成及 IDE 烧写.....	29
4.5.1 固件生成.....	30
4.5.2 IDE 烧写固件.....	30
4.6 IDE 使用技巧.....	30
4.6.1 头文件路径添加.....	30

4.6.2	代码优化等级的调整.....	31
5.	免责声明.....	32

1. Easy-RT1011 开发套件简介

1.1 概述

1.1.1 i.MX RT1010 跨界处理器

i.MX RT1010 是 NXP 半导体公司新推出的一款基于 ARM® Cortex®-M7 内核的微处理器，继承了 i.MX RT 系列产品一贯的高性能和易用性。处理器采用了 i.MX 6ULL 应用处理器架构，集成了应用处理器的高性能及微控制器易用、实时及低功耗的特性于一身，并且使用了低成本的 LQFP 封装方式，为急需提升产品性能而又不想增加过多成本的客户提供了一项全新的选择。

1.1.2 特性

- 系统：
 - ARM® Cortex®-M7 处理器，最大主频 500 MHz；
 - 16KB I-Cache，8KB D-Cache；
 - 浮点运算单元（FPU），支持 VFPv5 架构；
 - MPU，最大支持 16 个分区；
 - 128KB 的 I-TCM 和 D-TCM；
- 片上内存：
 - 64KB 的 Boot ROM；
 - 128KB 紧耦合内存（与 TCM 共享）；
- 实时的低延迟响应：
 - 最快速的实时响应，延迟低至 20ns。
- 外部存储接口：
 - SPI NOR Flash；
 - 单/双/四通道 SPI FLASH 并支持 XIP。
- 定时器：
 - 2 路通用可编程定时器（GPT）；
 - 4 通道的周期中断定时器（PIT）；
 - 1 路 FlexPWM。
- 音频接口：
 - SPDIF 输入输出接口；
 - 2 路同步音频接口（SAI），支持 I2S，AC97，TDM 和编解码器应用在 DSP 接口；
 - MQS 接口，用于经由 GPIO 端口输出的中等质量音频。
- 模拟接口：
 - 1 路的 ADC 转换器（高达 16 个通道，有一个通道不能使用）。
- 连接性：
 - USB2.0 OTG 控制器，芯片内部包含 USB-PHY 接口；
 - 4 路 UART；

- 2 路 I2C;
- 2 路 SPI;
- 1 路 FlexIO。
- 安全性：
 - 安全 JTAG 控制器 (SJC);
 - 高可靠性启动 (HAB);
 - 数据协处理器 (DCP), 支持 AES-128、SHA-1、SHA256、CRC32 计算;
 - FlexSPI, 内置即时 AES 解密 (OTFAD), 支持 AES-128, CTR 模式, 即时 QSPI 闪存解密;
 - 随机数产生器 (TRNG);
 - 安全非易失存储 (SNVS)。
- 调试接口：
 - ARM CoreSight 和 Trace 组件;
 - 支持 JTAG 和 SWD 调试接口。
- 电源管理
 - 全 PMIC 集成, 包括片上 DCDC 和 LDO;
 - 可编程温度传感器;
 - GPC 硬件电源管理控制器。

1.2 Easy-RT1011 开发套件

Easy-RT1011 是广州立功科技股份有限公司针对 NXP 的 i.MX RT 系列跨界处理器精心设计的一款开发套件。主控采用了 NXP i.MX RT 系列跨界处理器 MIMXRT1011, 具有低功耗应用处理器和高性能微控制器的优势, 提供超高性能的 ARM Cortex-M7 内核、实时功能和 MCU 级可用性, 可以应用于音频子系统、消费类产品、家庭和楼宇自动化、智能家居、电动机控制和电源转换等场合。Easy-RT1011 实物图如图 1.1 所示, 对应的接口说明如图 1.2 所示。

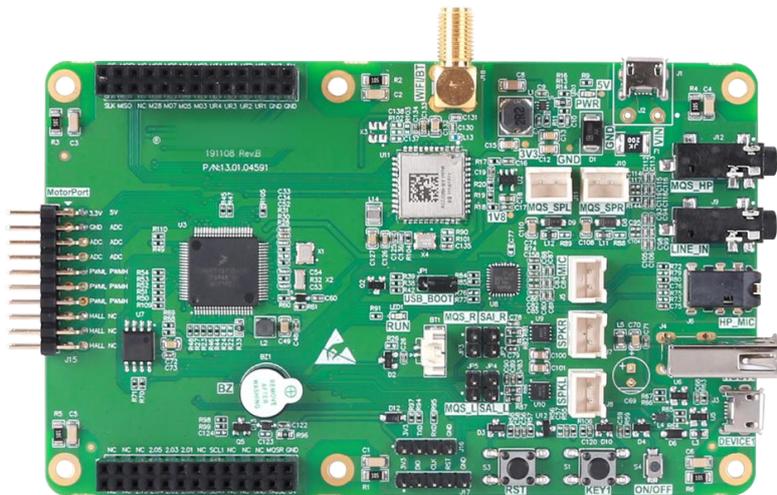


图 1.1 Easy-RT1011 实物图

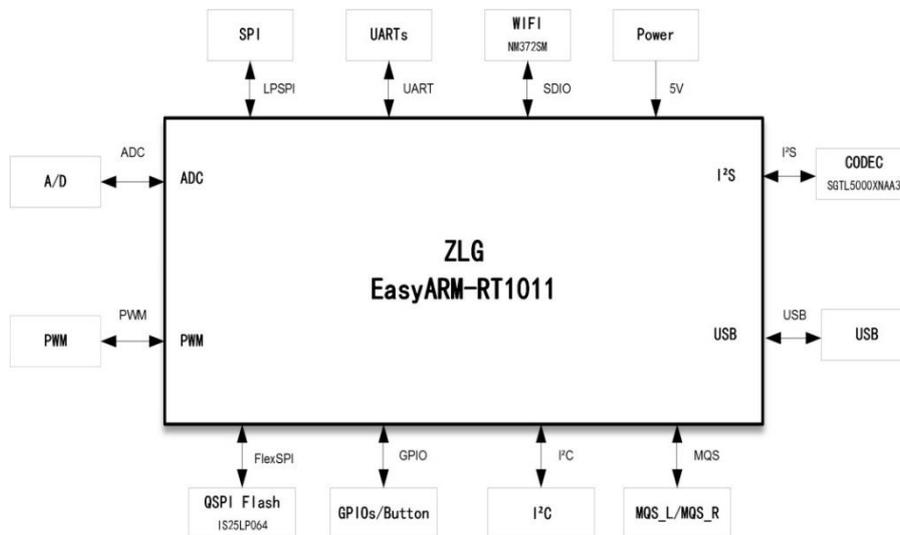


图 1.2 Easy-RT1011 接口说明

1.2.1 功能框图

Easy-RT1011 功能框图如下图 1.3 所示。

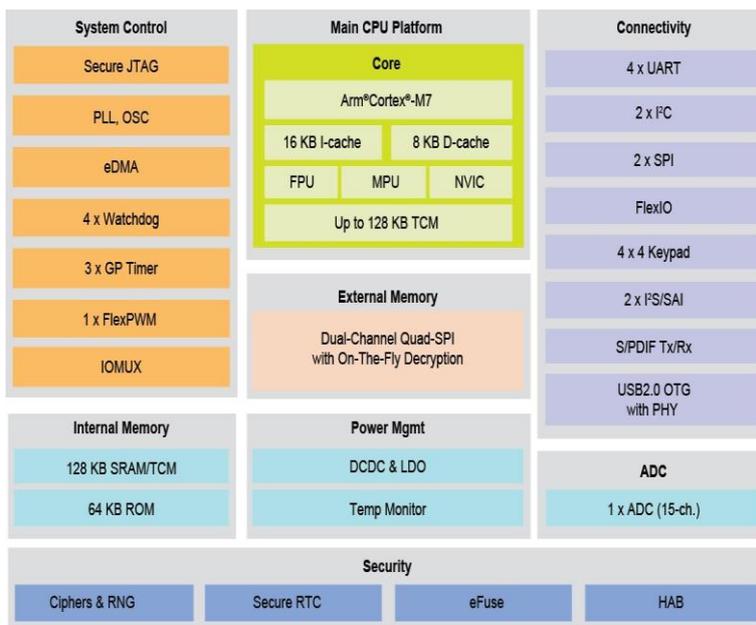


图 1.3 Easy-RT1011 功能框图

1.2.2 板级资源

Easy-RT1011 板级资源如下表 1.1 所示。

表 1.1 Easy-RT1011 参数表

产品型号	Easy-RT1011
处理器	MIMXRT1011
主频	500MHz
硬盘	8MB NOR FLASH

续上表

调试	1 路 SWD 调试接口
USB	1 路 USB OTG
蜂鸣器	1 路无源蜂鸣器
SPI	1 路 SPI 接口
LED	2 路 LED 指示灯
UART	4 路 UART
GPIO	3 路 GPIO
SDIO	1 路 SDIO, 与 NM372SM 相连
I2C	1 路 I2C
I2S	1 路 I2S, 与 SGTL5000XNAA3 相连
ADC	1 路 ADC, 15 通道
MQS	2 路 MQS 接口

1.2.3 接口说明

1. 指示灯与蜂鸣器

为方便用户了解整个系统的运行状况，Easy-RT1011 设计了 2 个 LED 指示灯：电源指示灯 LED1（丝印为 PWR）、系统运行指示灯（丝印标示为 RUN）。此外，电路还加入了一个无源蜂鸣器 BZ1（丝印 BZ），可用于提示系统启动的状态，具体功能描述见表 1.2。

表 1.2 指示灯及蜂鸣器

丝印标号	默认功能	功能描述
PWR	5V 电源指示灯	5V 电源接入后，电源指示灯会长亮
RUN	系统运行指示灯	--
BZ1	蜂鸣器	--

2. 配置说明

Easy-RT1011 支持 1 个手动复位按键、1 个功能按键，4 个调试串口引脚，用于输出串口调试信息。调试串口引脚定义及说明分别见图 1.4、表 1.3 所述。

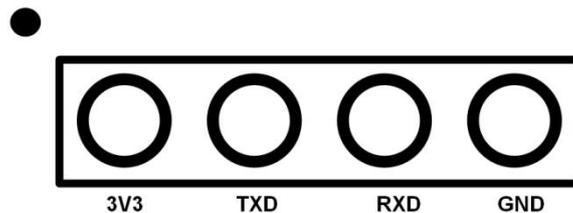


图 1.4 调试串口引脚

表 1.3 调试串口引脚说明

丝印名称	默认配置	功能描述
3V3	--	3V3 电源
TXD	--	串口 1 发送引脚
RXD	--	串口 1 接收引脚
GND	--	电源地

3. PC-104 转接口说明

Easy-RT1011 将 RT1011 处理器引脚功能进行重定义，规范每个 I/O 引脚为固定功能，以配合标准程序的开发，保证产品设计具有良好的兼容性和稳定性。Easy-RT1011 采用 PC-104 转接口 26 管脚排针的形式将相应功能引脚和 I/O 管脚引出，图 1.5 为 J13/14 管脚序号。

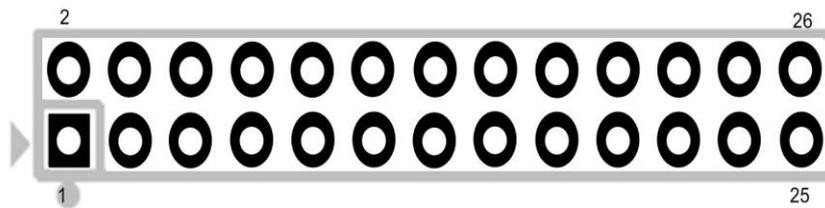


图 1.5 PC-104 转接口管脚定义

4. J13 接口引脚说明

表 1.4 J13 接口管脚说明

位号	丝印名称	默认配置	功能描述	备注
1	NC	悬空	NC	IO 口功能不局限于此， 可通过找到对应 IO 口复 用为其它功能
2	NC		NC	
3	NC		NC	
4	NC		NC	
5	2.13	GPIO_SD_13	GPIO	
6	NC	悬空	GPIO	
7	2.04	GPIO_SD_04	GPIO	
8	2.05	GPIO_SD_05	GPIO	
9	2.02	GPIO_SD_02	GPIO	
10	2.03	GPIO_SD_03	GPIO	
11	2.00	GPIO_SD_00	GPIO	
12	2.01	GPIO_SD_01	GPIO	
13	NC	悬空	--	
14	NC		--	

15	SDA1	I2C1	I2C1_SDA		
16	SCL1		I2C1_SCL		
17	NC	悬空	--		
18	NC		--		
19	NC		NC		
20	NC		NC		
21	GND	电源地	GND		
22	NC	悬空	NC		
23	MQSL	MQS	MQS 音频左声道输出		
24	MQSR	MQS	MQS 音频右声道输出		
25	5V	5V 输出	5V		只作为输出
26	GND	电源地	GND		

5. J14 接口引脚说明

表 1.5 J14 接口管脚说明

位号	丝印名称	默认配置	功能描述	备注
1	SLK	SPI1	SPI1 时钟信号	IO 口功能不局限于此， 可通过找到对应 IO 口复 用为其它功能
2	CS		SPI1 片选信号	
3	MISO		SPI1 主入从出	
4	MOSI		SPI1 主出从入	
5	NC	悬空	NC	
6	NC		NC	
7	M28	GPIO_AD_14	预留引脚	
8	M08	GPIO_08	预留引脚	
9	M07	GPIO_07	预留引脚	
10	M06	GPIO_06	预留引脚	
11	M05	GPIO_05	预留引脚	
12	M04	GPIO_04	预留引脚	
13	M03	GPIO_03	预留引脚	
14	M02	GPIO_02	预留引脚	
15	UR4	UART4	UART4 接收数据线	

16	UT4		UART4 发送数据线	
17	UR3	UART3	UART3 接收数据线	
18	UT3		UART3 发送数据线	
19	UR2	UART2	UART2 接收数据线	
20	UT2		UART2 发送数据线	
21	UR1	UART1	UART1 接收数据线	
22	UT1	UART1	UART1 发送数据线	
23	GND	3.3V 输出	3.3V	只作为输出
24	3V3	电源地	GND	
25	GND	电源地	GND	只作为输出
26	5V	5V 输出	5V	

6. J17 接口说明

J17 为 SWD 的接线座子，接口示意图如图 1.6 所示，SWD 接口管脚定义如表 1.6 所示。

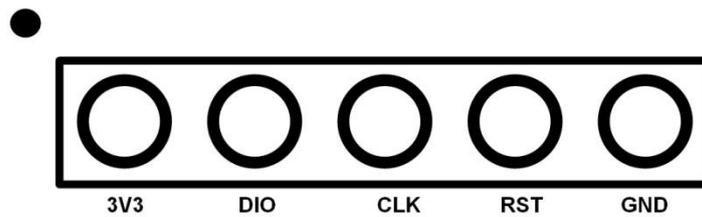


图 1.6 SWD 接口

表 1.6 SWD 接口管脚定义

位号	丝印名称	功能描述	备注
1	3V3	3.3V 电源	输出
2	DIO	数据线	输入/输出
3	CLK	时钟	输入
4	RST	复位	输入
5	GND	电源地	--

7. MotorPort 接口说明

MotorPort 接口为 20 管脚排针，接口示意图如图 1.7 所示，MotorPort 接口管脚说明说明如表 1.7 所示。

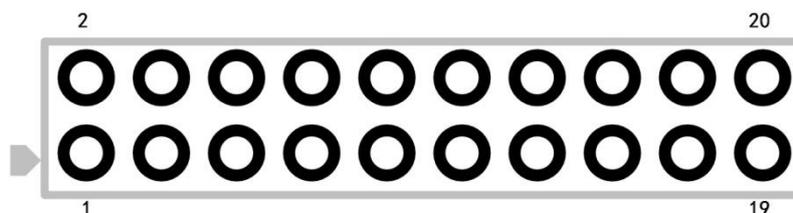


图 1.7 MotorPort 接口示意图

表 1.7 MotorPort 接口管脚说明

位号	丝印名称	默认配置	功能描述	备注
1	3.3V	3.3V 电源	3.3V	IO 口功能不局限于此，可通过找到对应 IO 口复用为其它功能
2	5V	5V 电源	5V	
3	GND	电源地	GND	
4	ADC	ADC_Ibus	ADC	
5	ADC	ADC_Ubus		
6	ADC	ADC_IC/UC		
7	ADC	ADC_IB/UB		
8	ADC	ADC_IA/UA	PWM	
9	PWML	GPIO_07		
10	PWMH	GPIO_08		
11	PWML	GPIO_05		
12	PWMH	GPIO_06		
13	PWML	GPIO_03	PWM	
14	PWMH	GPIO_04		
15	HALL	HALL_W	HALL	
16	NC	悬空	NC	
17	HALL	HALL_V	HALL	
18	NC	悬空	NC	
19	HALL	HALL_U	HALL	
20	NC	悬空	NC	

2. 资料目录结构说明

开发板提供以下资料：

资料	所在目录
开发板图片	1.开发板图片
快速入门指导	2.快速入门
相关集成开发环境的插件	3.集成开发环境
例程代码	4.示例代码
原理图和硬件设计指南	5.硬件设计
芯片手册	6.数据用户（参考）勘误手册
其他	7.其他

“2.快速入门”中包含使用 Easy-RT1011 开发板的入门指导文档。

“3.集成开发环境”中包含了开发 Easy-RT1011 支持的开发环境安装说明与下载链接。

“4.例程代码”包含了原厂提供的 SDK 包及立功科技适配 Easy-RT1011 后的 SDK 包。如图 2.1 所示是 SDK 包的目录结构。

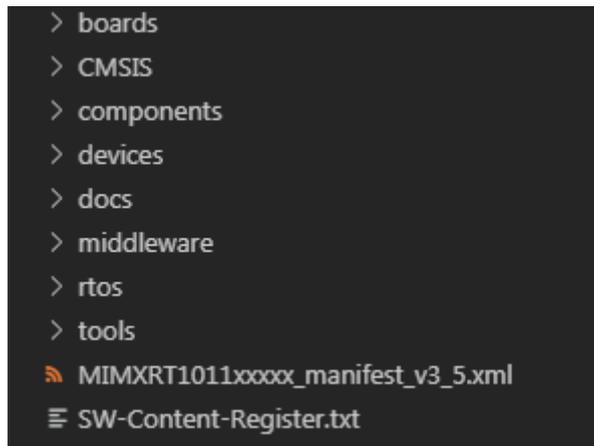


图 2.1 SDK 包目录结构

“CMSIS”目录中是 CMSIS 规范相关的头文件和库文件；

“devices”目录中是 MIMXRT1011 系列芯片的外设驱动库以及不同开发环境(Keil MDK、MCUXpresso、IAR 等)的启动代码；

“docs”目录是 NXP 原厂提供 SDK 包使用的指导文档，如果需要详细了解 SDK 包的使用，可以参考该目录下的文档；

“middleware”目录中包含了常用的中间件，有 USB 协议栈、mcuboot、issdk、littlevgl、WIFI 及 FATFS 文件系统的源码，以及 emWin 的库文件和头文件；

“rtos”目录中是实时操作系统，包含 FreeRTOS 的源码；

“tools”目录中是 cmake 工具需要的相关文件。

SDK 包中“board”目录中是针对不同开发板的例程。如图 2.2 所示是针对 Easy-RT1011

开发板的例程目录。其中”**driver_examples**”目录中的例程是针对芯片每个模块的单独例程，”**rtos_examples**”目录中是 FreeRTOS 的例程，”**usb_examples**”目录中是 USB 的例程，包括 USB 设备和 USB 主机的例程：

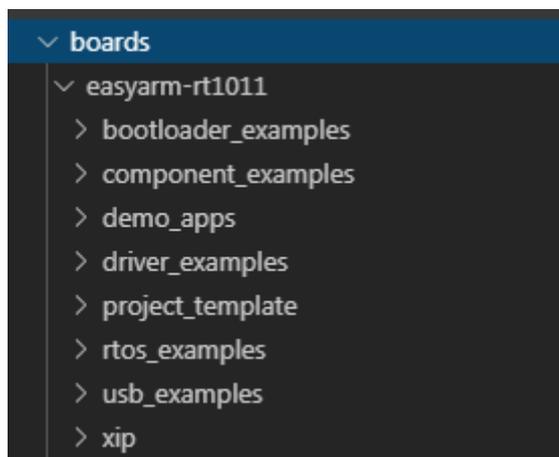


图 2.2 例程目录

3. 例程使用指南

3.1 JLINK 仿真器

3.1.1 概述

J-Link 是 Segger 公司为支持仿真 ARM 内核芯片推出的 JTAG 仿真器。配合 IAREWAR, ADS, KEIL, WINARM, RealView 等集成开发环境支持所有 ARM7/ARM9/ARM11, Cortex M0/M1/M3/M4/M7, Cortex A5/A8/A9 等内核芯片的仿真, 与 IAR, KEIL 等编译环境无缝连接, 操作方便。

3.1.2 特点

J-Link 仿真器具有以下特点:

- 支持 ADS、IAR、KEIL 开发环境;
- 完全即插即用;
- 标准 20 芯 JTAG 仿真插头, 支持多 JTAG 器件串行连接;
- 支持 SWD 接口;
- 带 J-link TCP/I server, 允许通过 TCP/IP 网络使用 J-Link。

3.2 硬件连接

进入例程调试之前, 将 J-Link 仿真器下载器与 Easy-RT1011 进行连接, Easy-RT1011 板子上的 SWD 接口原理图如图 3.1 所示。

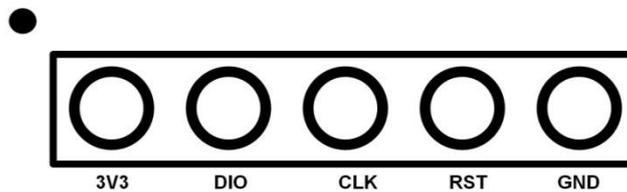


图 3.1 SWD 接口图

3.3 例程简介

这里以 SDK 包中目录“boards\EasyARM-RT1011\driver_examples\gpio\led_output\”例程为例, 介绍如何使用 SDK 包中的例程。

双击打开如图 3.2 所示红框标注的“igpio_led_output.uvprojx”工程文件, 打开后的界面如图 3.3 所示。



图 3.2 Keil 工程文件

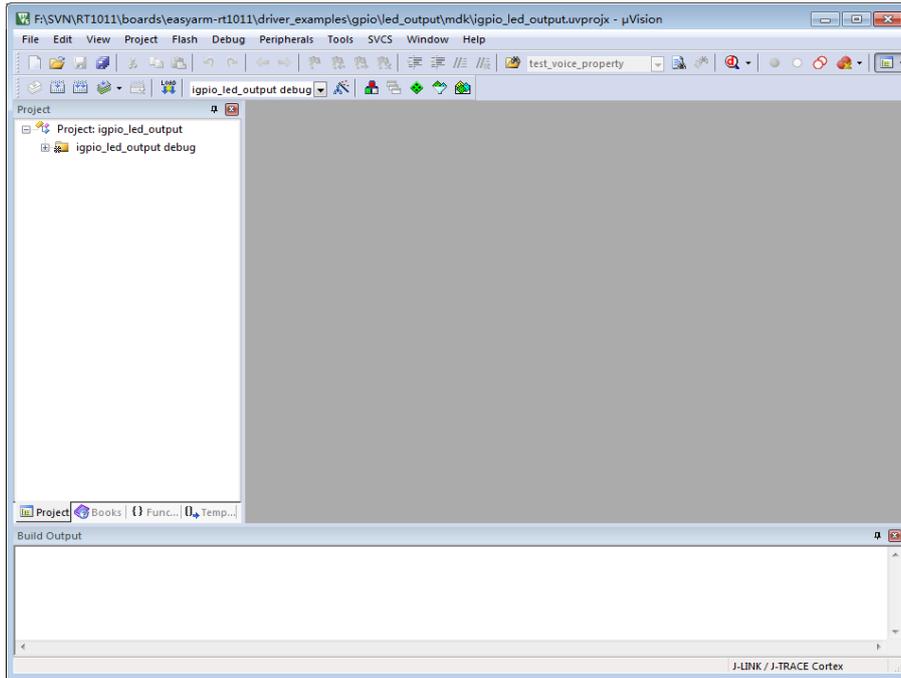


图 3.3 打开 Keil 工程文件

展开整个工程文件，结构如图 3.4 所示。

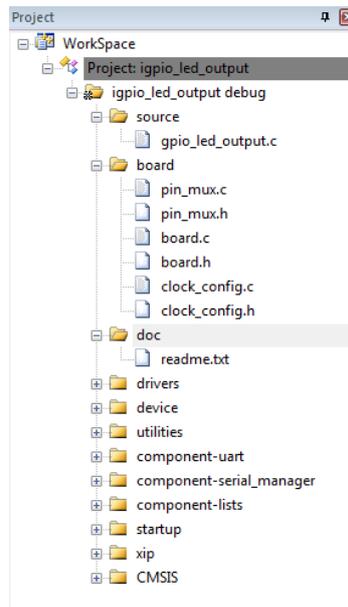


图 3.4 工程结构

“**startup**”目录中是芯片的启动代码和系统初始化代码；

“**doc**”目录下是例程的说明文件；

“**drivers**”目录是芯片的驱动代码，包括芯片所有片上模块的驱动；

“**source**”目录是应用程序代码，main 函数位于该目录的某个文件中，本例程的 main 函数就位于 igpio_led_output.c 文件中；

“**board**”目录是针对特定板子的引脚复用配置代码(pin_mux.c pin_mux.h)、时钟配置代码(clock_config.c clock_config.h)以及板级初始化代码(board.c board.h)。

3.4 KEIL 工程配置选择与编译

3.4.1 选择配置标签

点击如图 3.5 所示红色部分的图标，弹出如图 3.6 所示的“Manage Project Items”对话框，可以看到，工程中添加了四种配置选项：

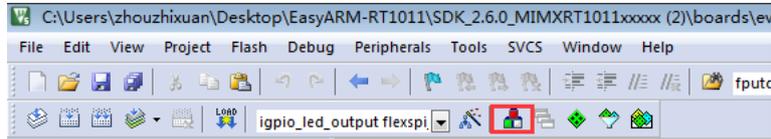


图 3.5 选择工程标签

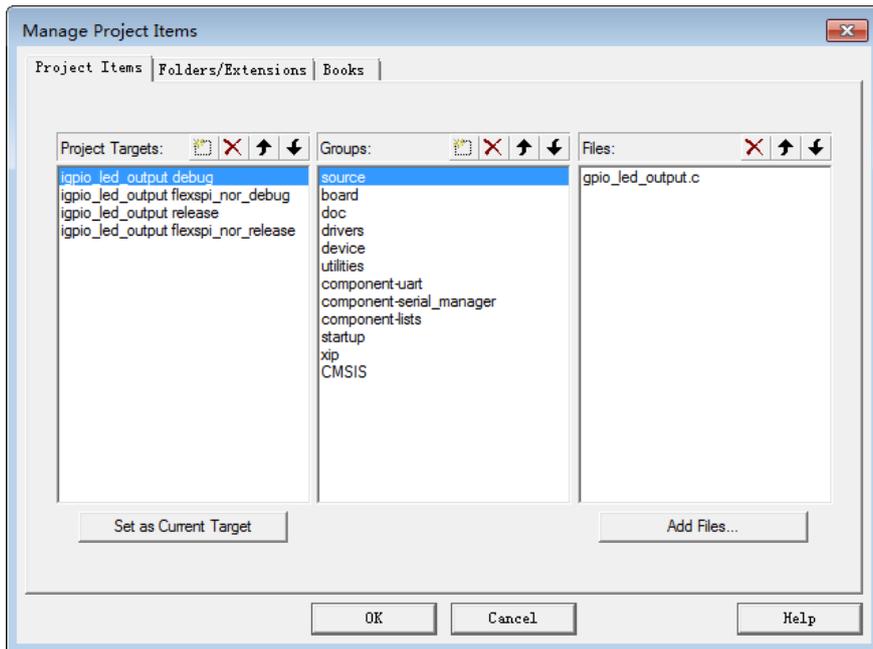


图 3.6 工程配置示意

- igpio_led_output debug: 生成加载到 ram 中的固件，携带 debug 信息；
- igpio_led_output flexspi_nor_debug: 生成加载到 nor flash 中的固件，携带 debug 信息；
- igpio_led_output release: 生成加载到 ram 中的固件，移除 debug 信息；
- igpio_led_output flexspi_nor_release: 生成加载到 nor flash 中的固件，移除 debug 信息；

3.4.2 编译程序

接下来可以对工程文件进行编译，点击如图 3.7 所示中的红色部分即开始编译。

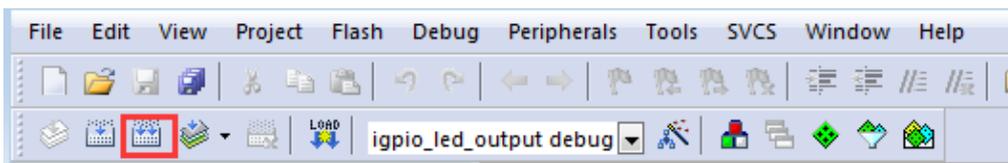


图 3.7 工程文件编译

工程文件编译完成后将会出现如图 3.8 所示的提示。

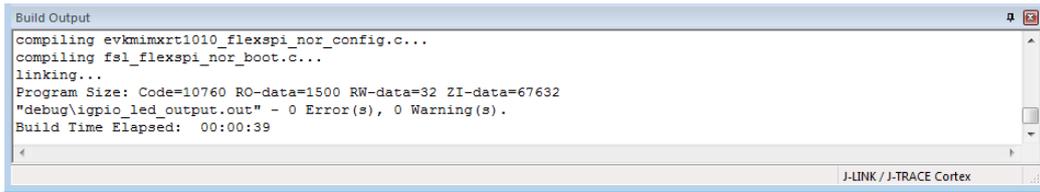


图 3.8 工程文件编译成功

3.5 KEIL 软件调试

3.5.1 工程配置

将仿真器与开发板正确相连，同时确保能给开发板正常供电。选择图 3.6 中的 igpio_led_output flexspi_nor_debug 配置选项，点击【OK】后配置仿真器信息，为后续仿真与下载做准备。

点击图 3.9 中的魔术棒按钮，弹出“Options for Target ‘igpio_led_output flexspi_nor_debug 配置’”对话框，切换到其中的【Debug】设置界面，如图 3.10 所示。



图 3.9 打开工程配置卡

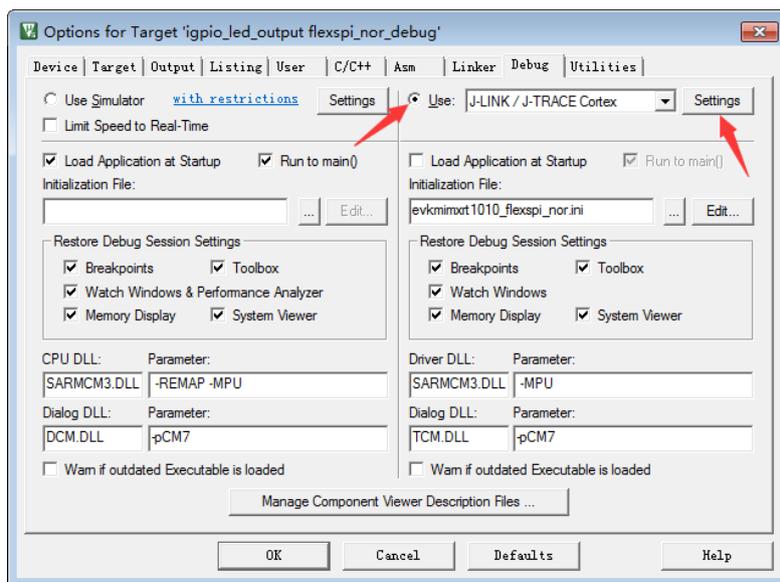


图 3.10 工程配置 Debug 界面

勾选【Use:】前面的圆点，并在下拉栏当中找到“J-LINK/J-TRACE Cortex”，然后点击【Settings】进入详细设置界面，如图 3.11 所示：

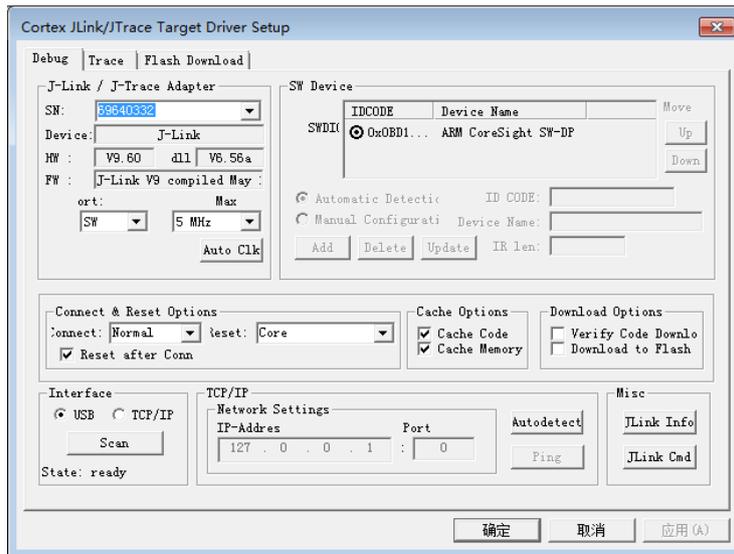


图 3.11 Settings 界面

在对话框左边的【Port】下拉栏中选择“SW”，正常情况下，【SW Device】栏将会自动显示出仿真器 ID 信息，再点击【确定】→【OK】完成设置。

3.5.2 仿真调试

工程编译成功及仿真器配置完成后即可进入到 Debug 调试界面，点击如图 3.12 所示红色部分即可开始调试。



图 3.12 软件调试

成功进入到软件调试后的界面如图 3.13 所示。

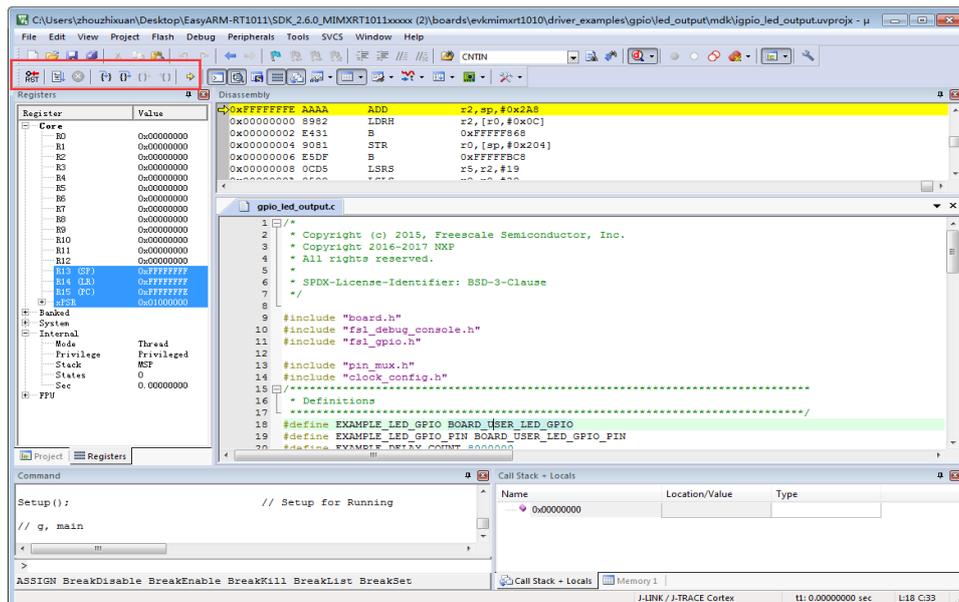


图 3.13 成功进入到调试界面

对上图 3.13 所示红色部分的调试工具进行操作即可实现相应的软件的调试。

3.5.3 调试工具说明

调试工具的使用具体参照表 3.1。

表 3.1 调试工具功能说明

符号	功能说明	符号	功能说明
	复位 CPU		运行到当前行（中间有断点则运行至断点处）
	全速运行		设置/移除断点
	停止		使能/禁能断点
	步进		禁能所有断点
	步越		移除所有断点
	步出		

3.6 程序下载

参照 3.5.1 的工程配置小节，完成 J-Link 配置后，单击工程配置界面的【Utilities】选项卡，如下图 3.14 所示，按照下图 3.15 所示选择 flash 编程算法，以及算法使用 RAM 的起始地址和大小。

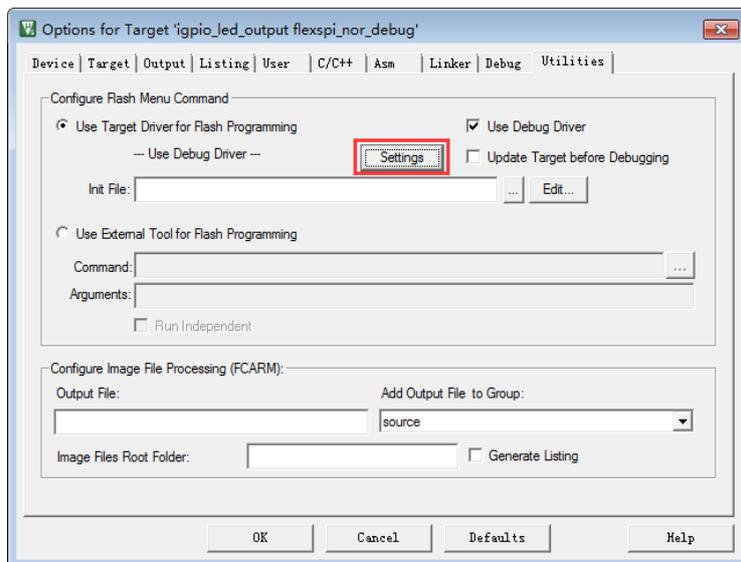


图 3.14 Utilities 选项

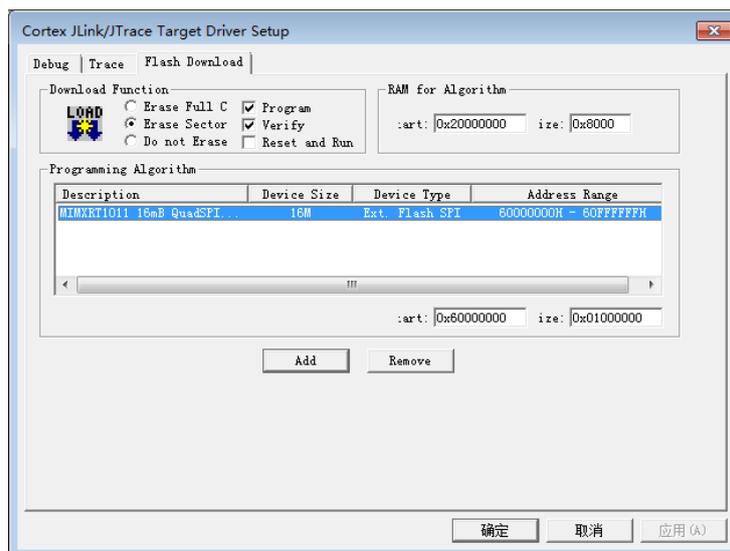


图 3.15 KEIL 下载配置

重新编译工程，等待编译完成后点击图 3.16 的下载按钮，开始下载。

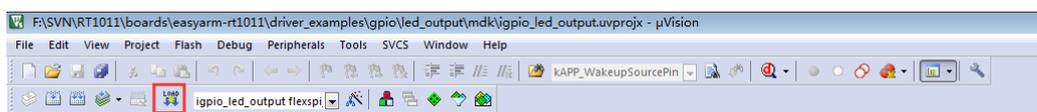


图 3.16 KEIL 下载代码

程序下载完成时，在【Build Output】窗口将会显示“Flash Load finished”等内容，如图 3.17 所示。



图 3.17 Build Ouput

下载完成后，按下复位键程序即开始运行，此时可以从串口中看到运行信息输出，并且板载 LED 灯亮灭交替。

4. MCUXpresso IDE 使用指南

4.1 MCUXpresso IDE 简介

MCUXpresso IDE 为开发人员提供了一个易于使用的基于 Eclipse 的开发环境，适用于基于 Arm®Cortex®-M 内核的恩智浦 MCU，包括 LPC 和 Kinetis®微控制器以及 i.MX RT 交叉处理器。MCUXpresso IDE 提供高级编辑，编译和调试功能，并增加了特定于 MCU 的调试视图，代码跟踪和分析，多核调试以及集成配置工具。MCUXpresso IDE 调试连接支持 Freedom, Tower®系统, LPCXpresso, i.MX RT 和您的定制开发板，以及来自恩智浦, P & EMicro®和 SEGGER®的业界领先的开源和商用调试探针。

对于 IDE 环境搭建及 JLINK 软件的安装本章不做介绍，注意 IDE 安装时默认自动在 c 盘下安装 JLINK 软件，并导航到该软件路径。在下载调试时，用户若想使用自己安装的 JLINK，需要配置 JLINK 路径到用户 JLINK 安装位置，具体操作见“调试指南”小节。

本文所安装的 IDE 环境版本为 MCUXpresso IDE v11.0.0_2516。

4.2 SDK 的获取及导入

这里介绍如何从 NXP 官网获取 RT1011 相应的 SDK 包，以及如何将下载的 SDK 包导入到 MCUXpresso IDE。

4.2.1 获取 RT1011 的 SDK

1. 登陆 SDK 构建界面

在浏览器中登陆 nxp 官网 <https://www.nxp.com/>，搜索“mcuxpresso sdk”，点击选择“MCUXpresso Software Development Kit (SDK)”。如图 4.1 所示，单击红色框。



图 4.1 SDK 界面

如图 4.2 所示，接着选择“Select Development Board”。



图 4.2 SDK Builder 界面

输入您账号以登录 NXP，如图 4.3 所示，即进入了 SDK 构建界面。

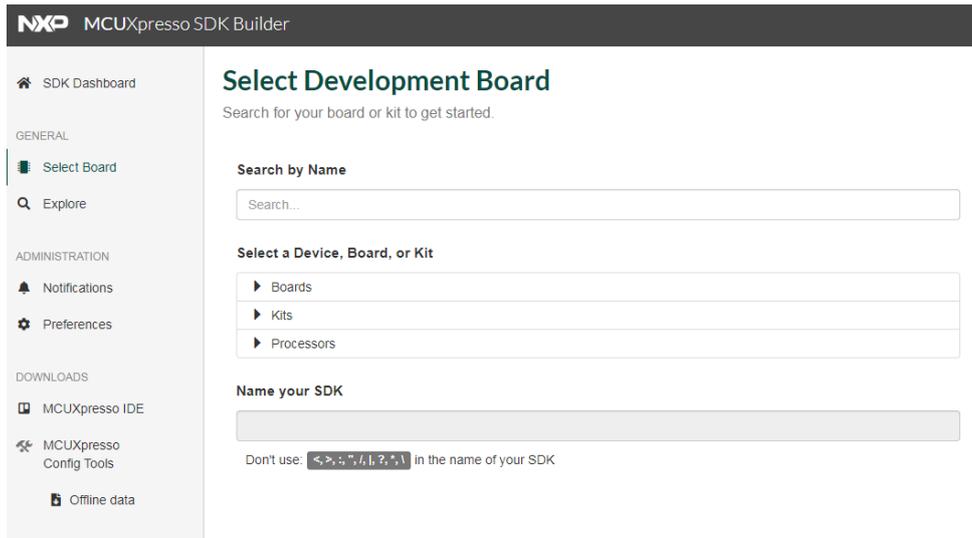


图 4.3 SDK 构建界面

2. 构建 SDK 并下载

在构建界面选择所要构建 SDK 对应的芯片，这里选择 MIMXRT1011xxxxx，然后点击“Build MCUXpresso SDK”，如图 4.4 所示，可以根据需要修改 SDK 的名字。

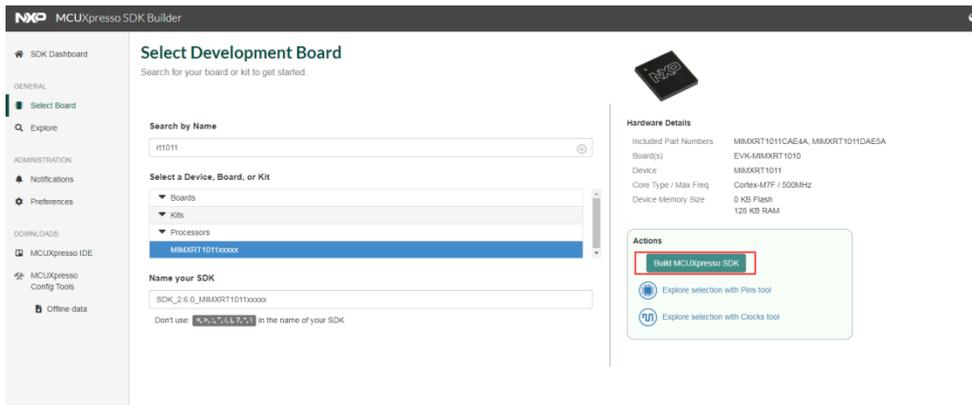


图 4.4 SDK 芯片选择

在图 4.5 中，配置 SDK，示例是选择操作系统为 windows，IDE 工具为 MCUXpresso，通过“Add software component”添加组件，默认包含 FatFS，USB stack 组件。

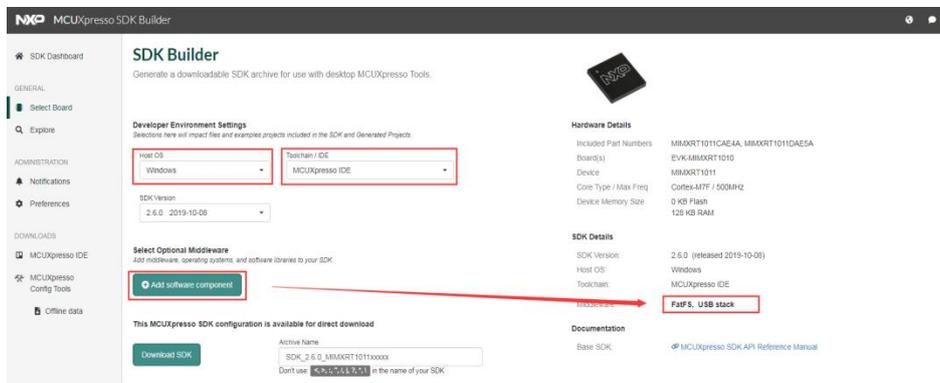


图 4.5 SDK 内容配置界面

配置完成之后，点击“Download SDK”。弹出的界面，选择“I Agree”，自动构建完成后

就会下载，用户可以根据需求下载相应的文档，如图 4.6 所示。

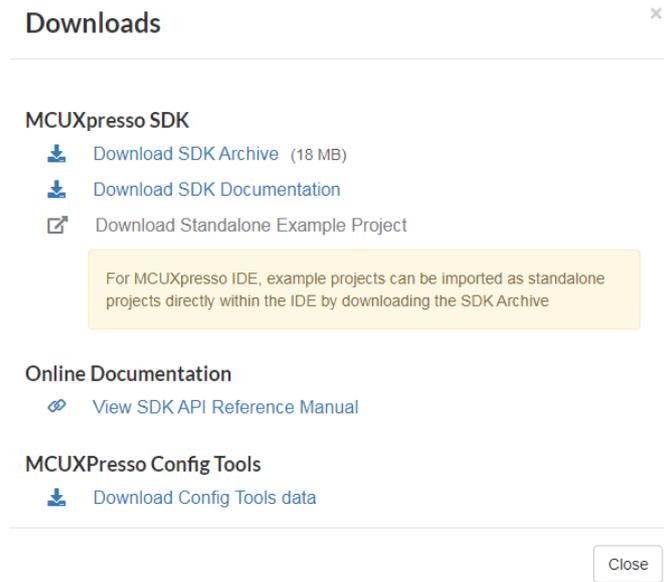


图 4.6 SDK Downloads 界面

4.2.2 向 IDE 导入 RT1011 的 SDK

打开 MCUXpresso IDE，首次打开会要求选择工作空间，之后创建的工程都是放在工作空间里，用户可以根据需要来重定位工作空间。进入之后，在图 4.7 的“Install SDKs”窗口导入用户下载的 SDK。导入方式，是直接将待导入的 SDK 拖到该窗口下。

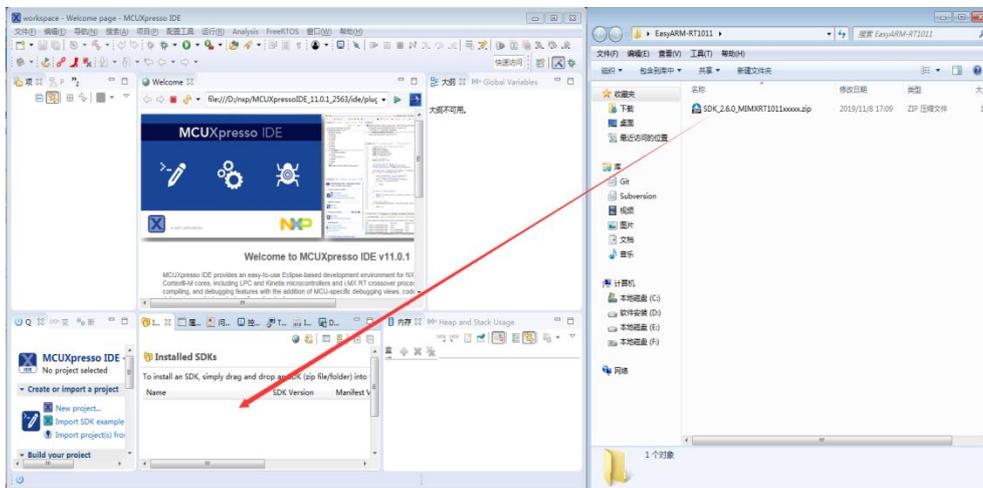


图 4.7 IDE 的 Installed SDKs

拖入之后，弹出一个窗口，选择确定即可。等待导入结束之后，“Install SDKs”窗口下就导入了 SDK，如图 4.8 所示。

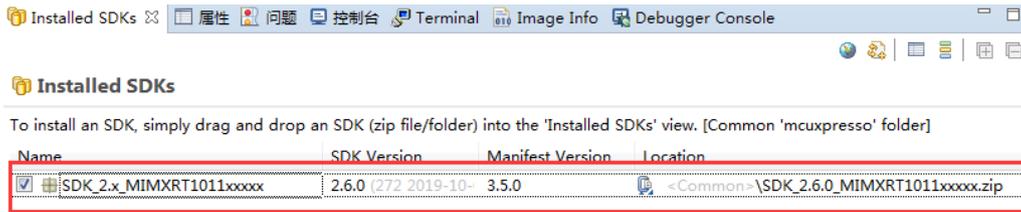


图 4.8 SDK 成功导入界面

4.3 工程的创建

工程的创建就是利用所安装的 SDK 包来创建工程，利用 SDK 可以导入模板工程，也可以新建工程。在“create or import a project”窗口通过选择“New project”或者“Import SDK example(s)”来新建工程或者导入模板工程，如图 4.9 所示。

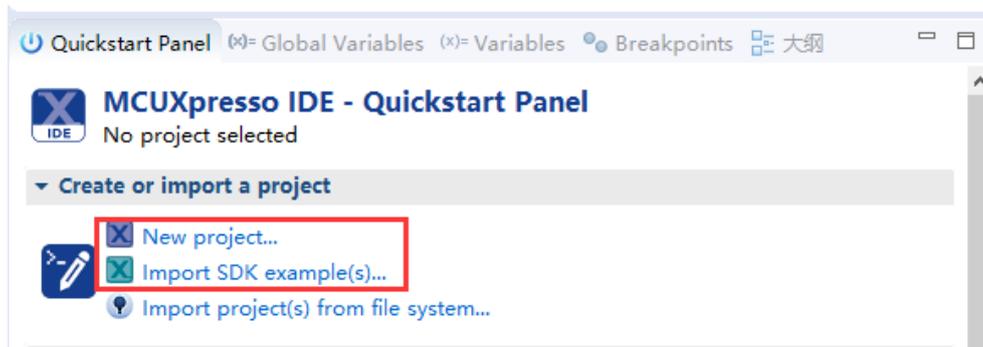


图 4.9 “create or import a project” 窗口

4.3.1 导入模板工程

MCUXpresso IDE 提供用户导入模板工程方式创建工程，用户无需“从零”开始建立工程，导入一个模板工程即可。如图 4.9 所示，选择“Import SDK example(s)”。

1. 选择使用的 SDK

在“SDK MCUs”栏选择所要使用的芯片即可选择对应的 SDK 包，如图 4.10 所示，然后点击下一步。

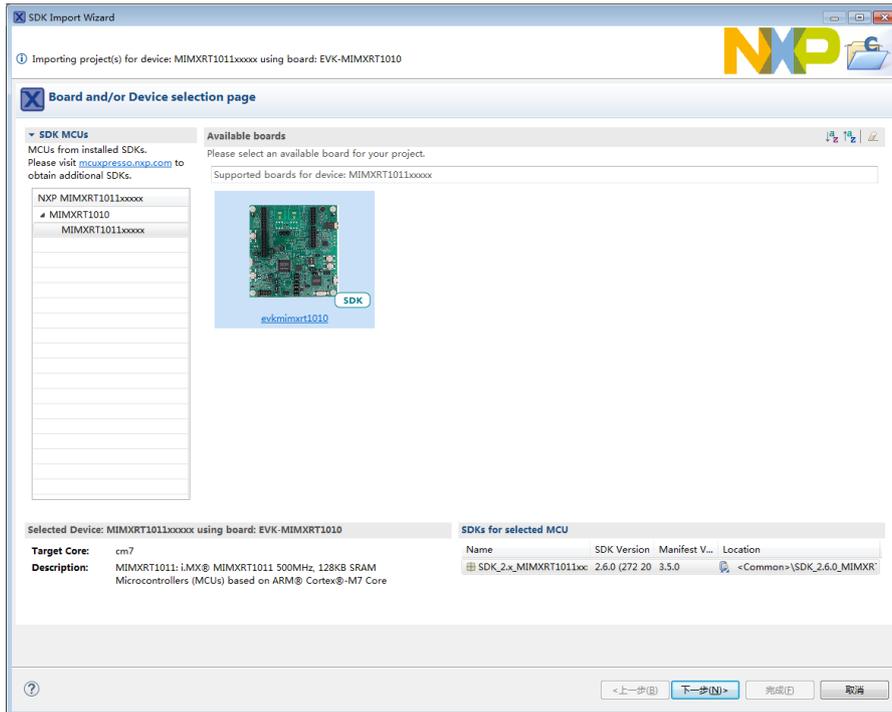


图 4.10 选择 SDK 创建工程

2. 选择要导入工程

这里以导入 hello_world 例程为例，如图 4.11 所示。用户可以根据需要修改工程名，也可以缺省以使用默认的工程名。将要导入的模板工程打勾即可，也可以“Examples”栏输入模板工程名字从而在模板工程中快速找到要导入的工程。

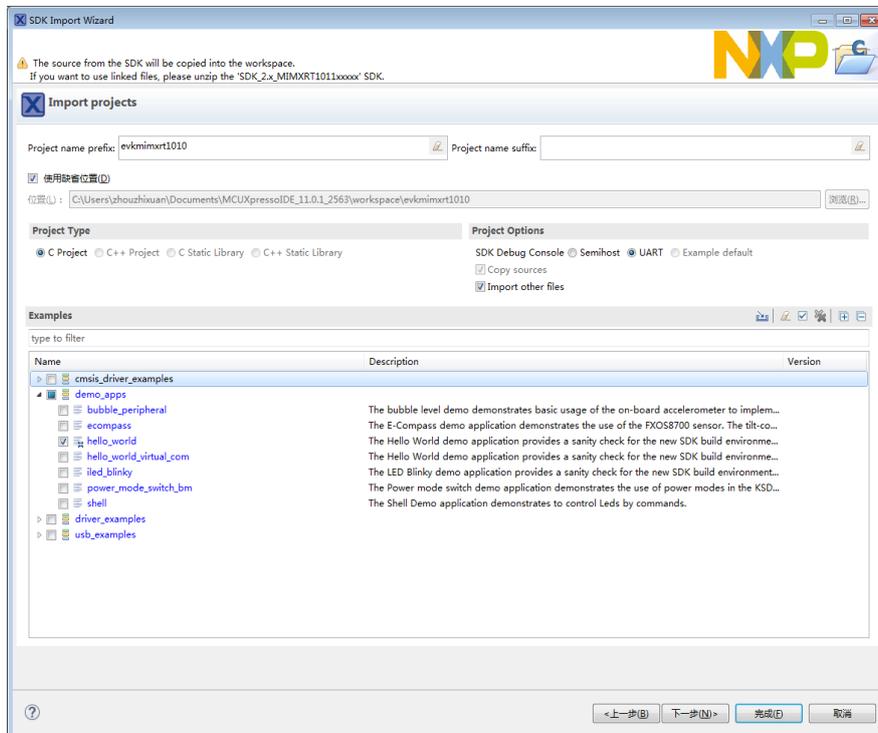


图 4.11 选择要导入的模板工程

3. 工程高级设置

这一步是可选择的，一般使用默认设置即可，即在上一步选择完成就配置结束了。如果用户需要进一步配置工程，则在上一步选择“下一步”进入“高级设置”，如图 4.12 所示，在这里根据需要设置芯片的硬 FPU、RAM/FLASH 范围及大小等等。

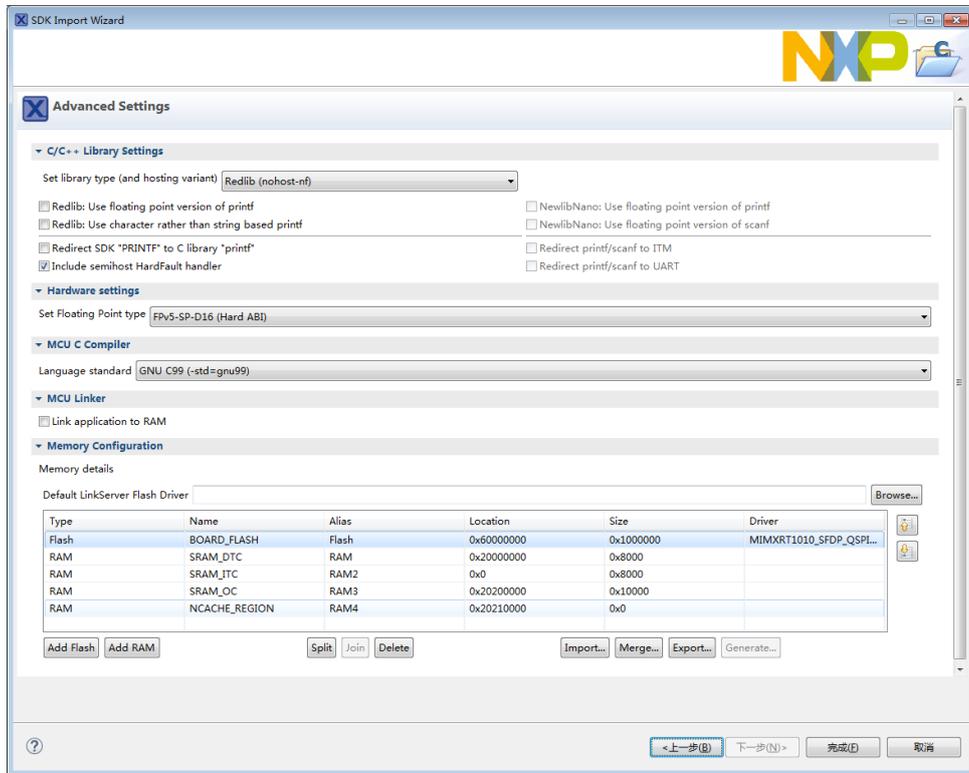


图 4.12 工程的高级设置窗口

最终完成之后，在“项目资源管理器”里就可以看到导入的模板工程，如图 4.13 所示。

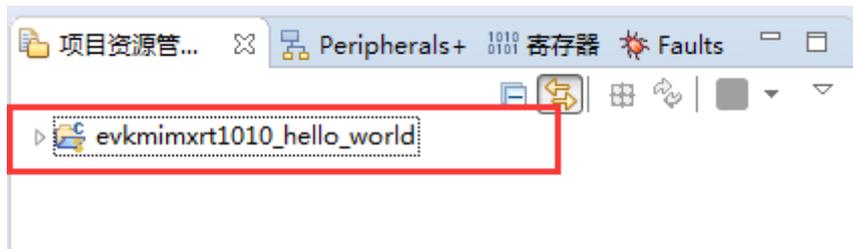


图 4.13 完成的导入模板工程

4.3.2 新建工程

如图 4.9 所示，选择“New project”。

1. 选择使用的 SDK

在“SDK MCUs”栏选择所要使用的芯片即可选择对应的 SDK 包，如图 4.10 所示，然后点击下一步。

2. 设置工程以及选择所需的组件

如图 4.14 所示，用户可以根据需要修改工程名，也可以缺省以使用默认的工程名。进行一些基本设置，包括根据具体的芯片型号选择 Device Packages、工程类型等等，这里选择 MIMXRT1011DAE5A、创建 C 语言工程、调试控制台输出选择 UART。

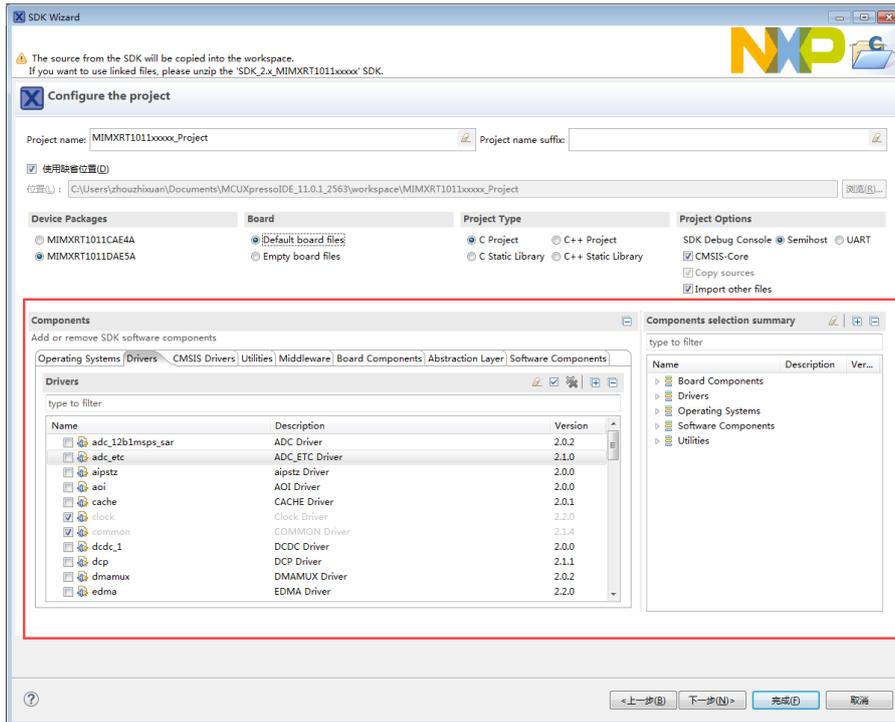


图 4.14 工程基本设置及选择组件

红框标注部分,就是用来选择该工程所要包含的组件。选择完组件之后可以选择“完成”,即开始创建工程,接下来就可以在“项目资源管理器”里看到新建的工程。如果用户需要进一步配置工程,则在上一步选择“下一步”进入“高级设置”。

3. 工程高级设置

如图 4.12 所示,在这里用户可以根据需要设置芯片的硬 FPU、RAM/FLASH 范围及大小等等。

4.4 调试指南

工程创建完之后就可以,进行编译调试了,这里导入的模板工程为例介绍 MCUXpresso 的调试。

4.4.1 编译工程

“项目资源管理器”窗口选择要编译的工程,然后点击  即可编译,如图 4.15 所示,正在编译“evkbimxrt1010_hello_world”工程。

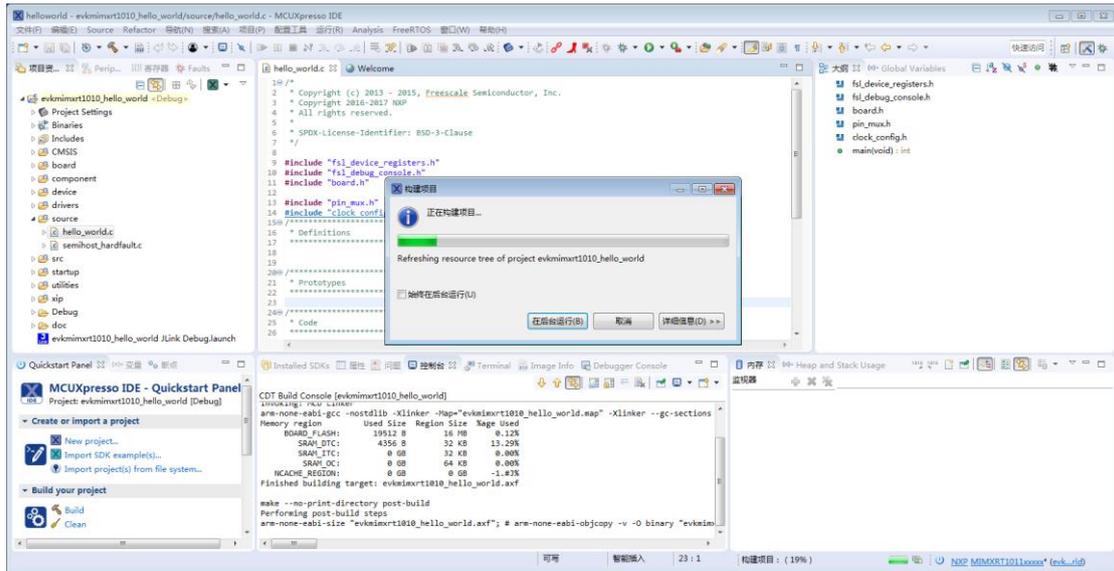


图 4.15 编译工程

编译时，可以在如所示的“控制台”窗口中查看编译过程的一些信息，在“问题”框中，可以查看错误或警告信息，双击错误项即可跳转到相应代码中。

成功编译后可以看到如图 4.16 所示红色线标注的信息输出，同时在该工程的目录下生成了一个 Debug 目录。

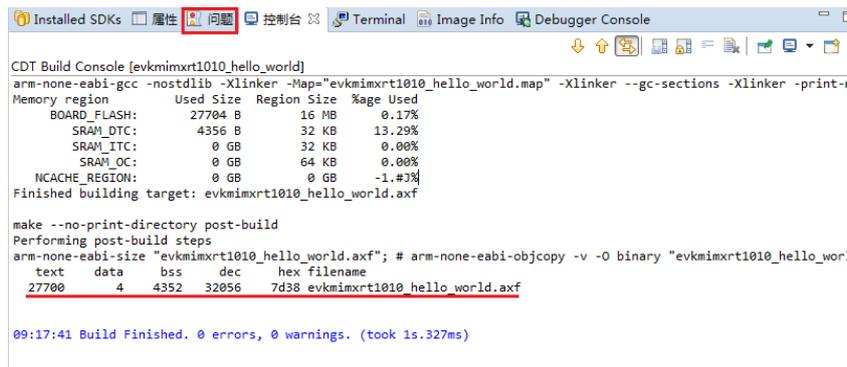


图 4.16 控制台的编译输出信息

如果编译完成之后，在工程目录中的 Debug 文件夹里没有看到生成的 axf 文件，可以执行更新操作即可看到相应文件。操作过程：先点击选中工程，然后右键选择图 4.17 中刷新按钮即可。

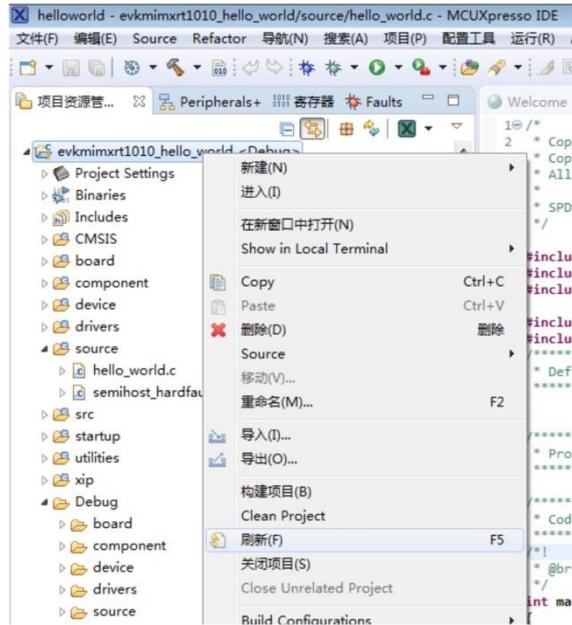


图 4.17 刷新工程以更新 axf 文件

4.4.2 进入调试界面

MCUXpresso 进入调试时就默认将代码下载到以 0x60000000 为起始地址的 FLASH 里，所以退出调试时，固件已经烧录到开发板上。

1. 使用 JLINK 连接开发板

我司开发板采用 JLINK 调试器的 SWD 接口连接，用户只需按照接口对应连接即可，确保正确连接。使用 USB 线缆连接 JLINK 和 PC 机。

2. 配置 JLINK 路径

IDE 在安装时默认自动在 c 盘下安装 JLINK 软件，并导航到该软件路径。若要需要重置路径到用户的 JLINK 路径请参照接下来的设置：窗口(W)"->"首选项(P)"->"MCUXpresso IDE" ->"j-link options”，在“J-link server executable”栏设置 JLinkGDBServerCL.exe 所在的安装路径，如图 4.18 所示。

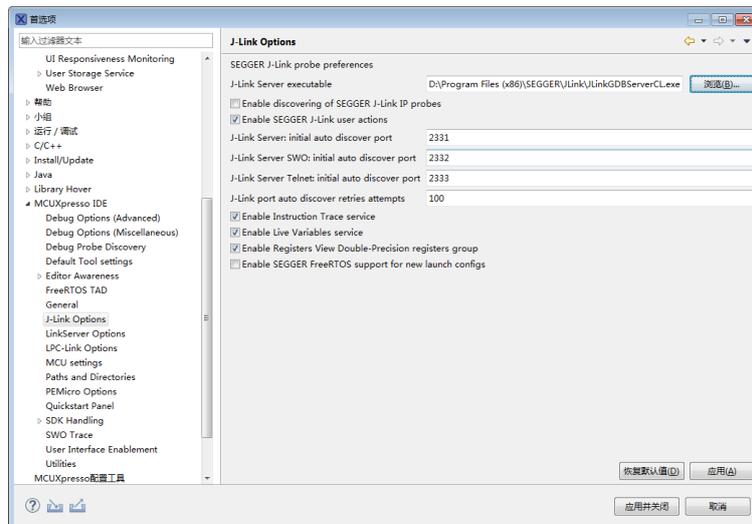


图 4.18 设置 JLinkGDBServerCL.exe 路径

3. 进入调试

选择“evkbimxrt1010_hello_world”工程，点击即可，第一次进入调试时，会触发一系列默认行为，包括调试器发现及选择调试器、自动创建该工程的启动配置文件、下载 axf 文件的二进制内容到开发板 FLASH 等等。这些默认行为，有些只是在第一次进入调试界面才有，后续再进入调试界面可能就跳过了。

点击蓝色蜘蛛按钮之后，弹出如图 4.19 的调试器选择界面，红框内会列出所有发现的调试器，用户只需要在其中选择一个调试器然后确定即可，因当前只连接了一个 JLINK，所以这里就只有一个调试器被发现。

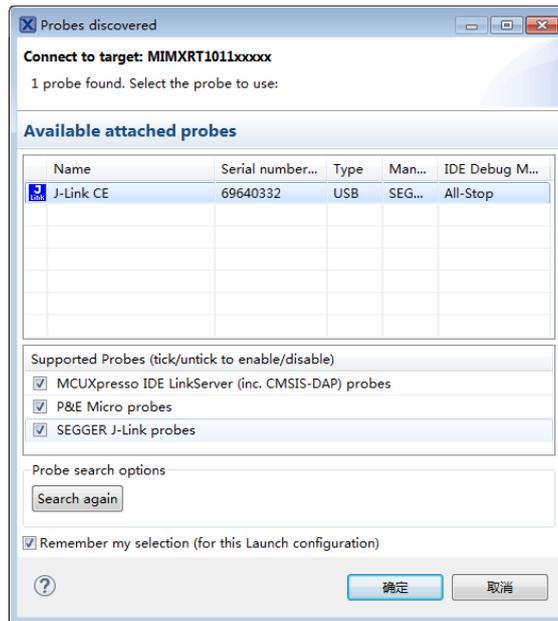


图 4.19 调试器发现及选择

接着就进入了调试界面，如图 4.20 所示。

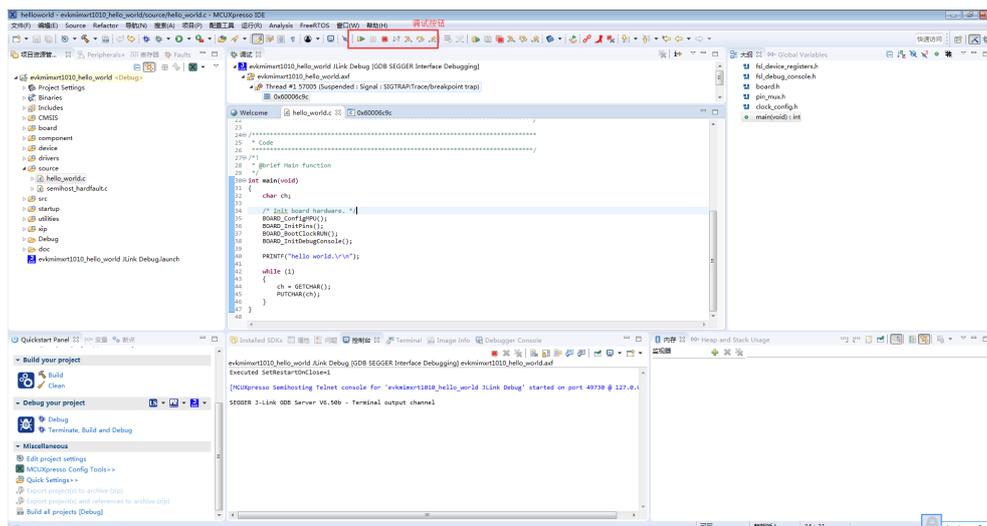


图 4.20 调试界面

4.4.3 调试代码说明

1. 调试时常用的窗口

如图 4.21 所示，这些窗口都可以调用出来，用户只需根据需要来调用相应窗口，具体

设置是：“窗口(W)”->“显示视图(V)”->“其他(O)”，在弹出的小窗口里选择需要调用的窗口。

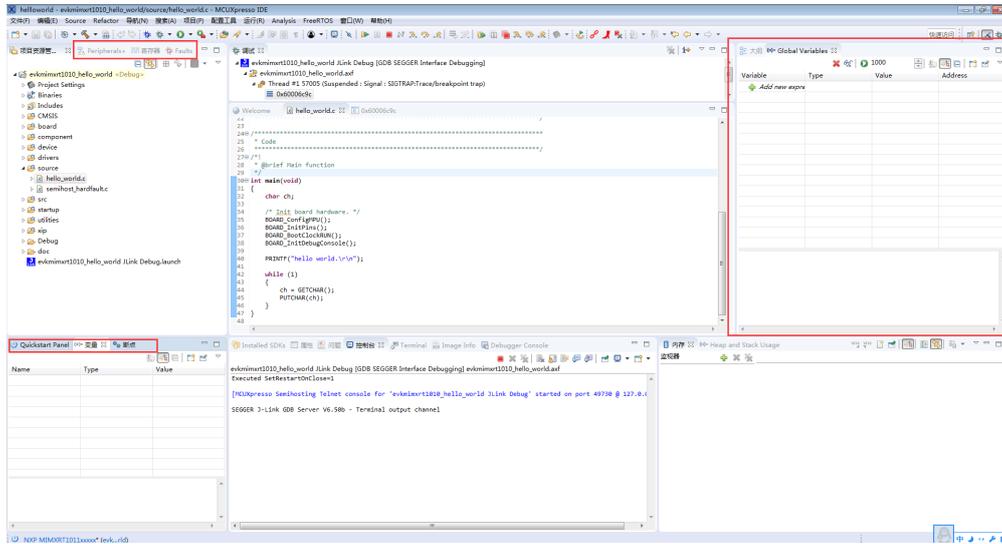


图 4.21 调试常用窗口

2. 调试工具的使用

调试工具的使用具体参照表 4.1。

表 4.1 调试工具使用说明

Button	Description	Keyboard Shortcut
	Restart program execution (from reset)	
	Run/Resume the program	F8
	Pause Execution of the running program	
	Terminate the debug Session	Ctrl + F2
	Clean up debug	
	Run, Pause, Terminate all debug sessions	
	Step over a C/C++ line	F6
	Step into a function	F5
	Return from a function	F7
	Step in, over, out all debug sessions	

3. 设置断点

在代码行左边空白处双击鼠标左键即可设置一个断点，设置成功后会在行前出现图标，并且所有断点都会列在断点窗口里。

4.4.4 启动文件配置

通常自动创建的配置文件无需再编辑，因为 IDE 创建的默认设置将是合适。但仍可以根据需要来编辑启动文件。有两种方式来进入编辑，一是直接双击启动文件，二是通过 Debug Configurations 进入。进入启动文件配置最简单的方法是，双击启动文件(.launch 文件)即可。

1. Debugger 界面

如图 4.22 所示，该界面可以 Jlink 连接方式，设备等等。

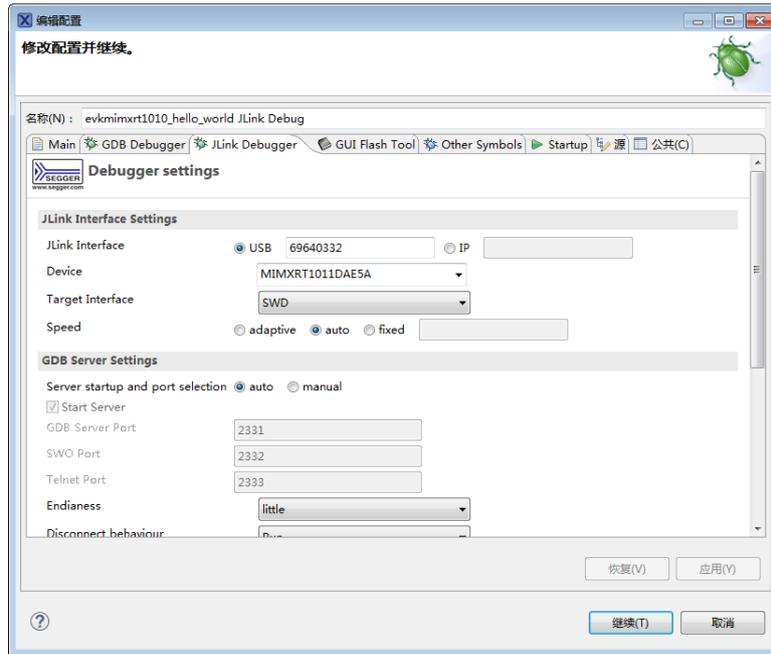


图 4.22 启动文件配置的 Debugger 界面

2. Starup 界面

如图 4.23 所示，“Set breakpoint at”是用来设置默认断点，使能就会设置一个默认断点，当前断点位置在 main 函数处，用户可以根据需要更改这个默认断点的位置。

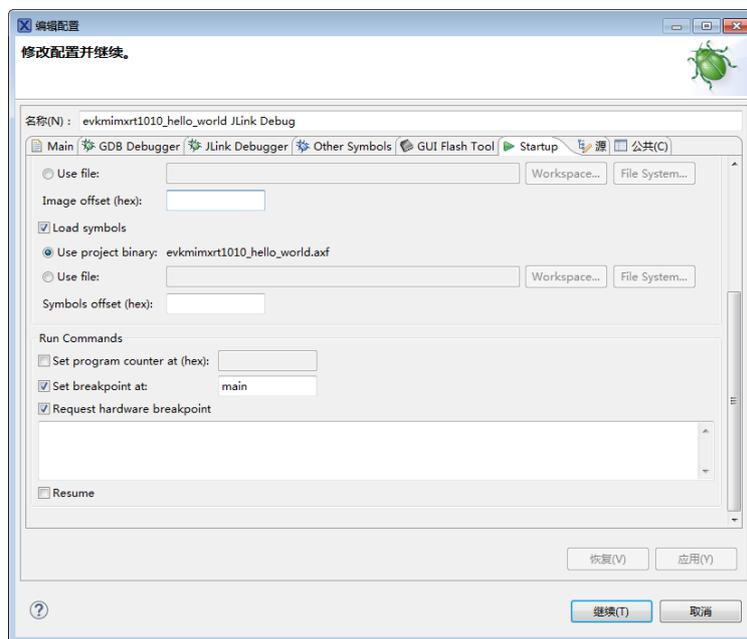


图 4.23 启动文件配置的 Starup 界面

4.5 固件生成及 IDE 烧写

在 IDE 里可以利用 .axf 文件手动或者自动生成 hex 文件或者 bin 文件，这里介绍手动生成固件，生成的固件可以通过第三方软件比如 J-Flash 进行烧写。当然该 IDE 也支持烧写功能。

注意：在 IDE 中进入调试时其实就已经将代码烧写到 Flash 中了。

4.5.1 固件生成

手动生成固件，是在工程目录下找到相应的.axf 文件，如图 4.24 所示，选中该文件之后，右键选择“Binary Utilities”，选择“create hex”或者“create binary”即可生成 hex 或者 bin 文件。

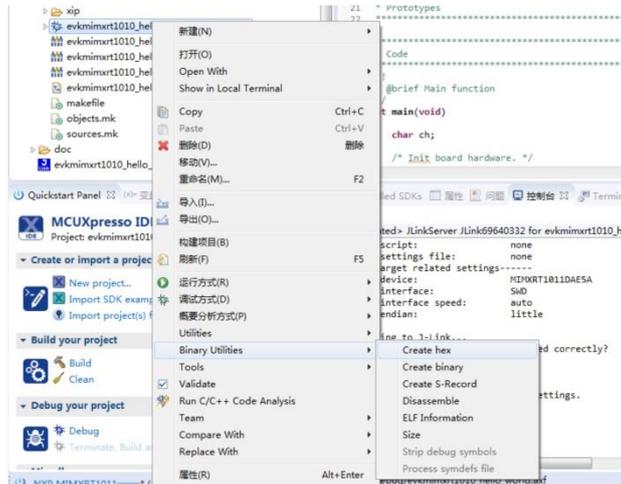


图 4.24 手动生成固件示例

4.5.2 IDE 烧写固件

确保烧写之前工程已经编译过了。在“Quickstart Panel”窗口下的“Debug your project”区域选择图 4.25 红框标注的选项，即进行固件烧写。

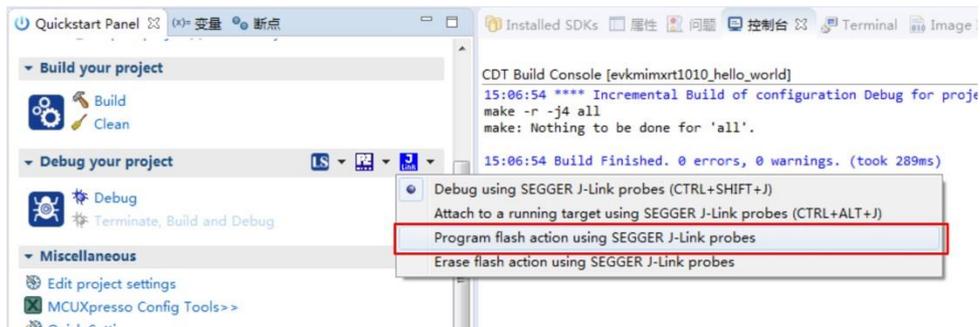


图 4.25 IDE 烧写固件

4.6 IDE 使用技巧

4.6.1 头文件路径添加

首先在“项目资源管理器”单击选择一个工程，然后：“项目(P)”->“属性”->“c/c++ Build”->“Settings”->“Tool Settings”->“MCU C Compiler”->“Includes”，如图 4.26 所示。

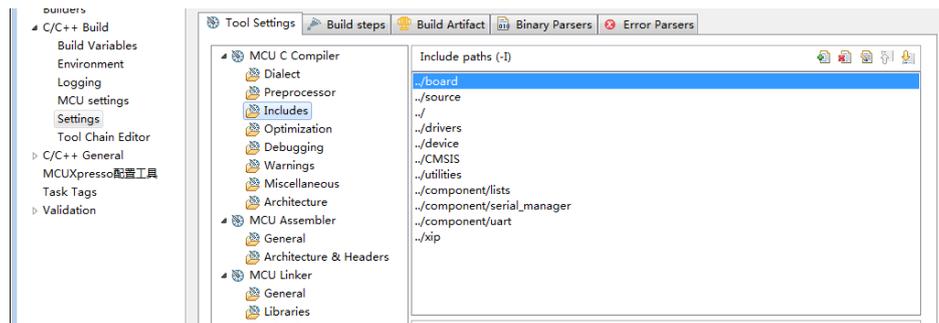


图 4.26 添加头文件路径

4.6.2 代码优化等级的调整

首先在“项目资源管理器”单击选择一个工程，然后：“项目(P)”->“属性”->“c/c++ Build”->“Settings”->“Tool Settings”->“MCU C Compiler”->“Optimization”，如图 4.27 所示。默认是 0 级优化，即不优化。

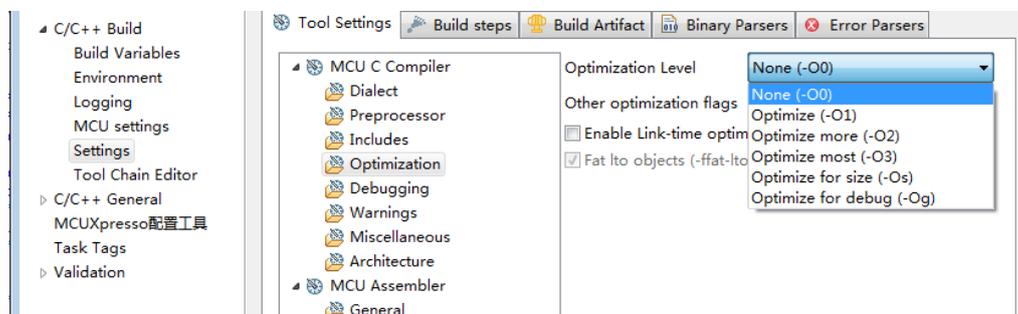


图 4.27 代码优化等级

5. 免责声明

本着为用户提供更好服务的原则，广州立功科技股份有限公司（下称“立功科技”）在本手册中将尽可能地为用户呈现详实、准确的产品信息。但鉴于本手册的内容具有一定的时效性，立功科技不能完全保证该文档在任何时段的时效性与适用性。立功科技有权在没有通知的情况下对本手册上的内容进行更新，恕不另行通知。为了得到最新版本的信息，请尊敬的用户定时访问立功科技官方网站或者与立功科技工作人员联系。感谢您的包容与支持！

专业 · 专注成就梦想

Dreams come true with professionalism and dedication.

广州立功科技股份有限公司

更多详情请访问

www.zlgmcu.com

欢迎拨打全国服务热线

400-888-2705

