

# Easy-RT1052 快速入门指南

基于 Easy-RT1052 Rev D

UM01010101 1.1.01 Date:2023/3/28

类别	内容
关键词	Easy-RT1052
摘要	介绍Easy-RT1052 Rev D开发板,及其快速入门指南。



# Easy-RT1052 快速入门指南 基于 Easy-RT1052 Rev.D 开发套件

User Manual

# 修订历史

版本	日期	原因
V1.0.00	2018/10/11	创建文档
V1.1.01	2023/03/28	更新文档模板

# 目 录

1.	Easy-	-RT1052	2	]
	1.1	i.M	IX RT1050 跨界处理器	
		1.1.1	概述	
		1.1.2	特性	2
	1.2	Eas	sy-RT1052 开发套件	3
		1.1.1	功能框图	
		1.1.2	性能参数	6
		1.1.3	接口说明	8
2.	资料	目录结	构说明	13
3.	例程	使用指	南	15
	3.1	例	程简介	15
	3.2	Ke	il 工程设置与编译	15
	3.3	Ke	il 软件调试	16
		3.3.1	CMSIS-DAP	16
	3.4		序下载	
4.	MCU	Xpresso	o IDE 使用指南	22
	4.1	MO	CUXpresso IDE 简介	22
	4.2	SD	ok 的获取及导入	22
		4.2.1	获取 RT1052 的 SDK	22
		4.2.2	向 IDE 导入 RT1052 的 SDK	24
	4.3	工	程的创建	25
		4.3.1	导入模板工程	25
		4.3.2	新建工程	28
	4.4	调	试指南	29
		4.4.1	编译工程	29
		4.4.2	进入调试界面	3
		4.4.3	调试代码说明	33
		4.4.4	启动文件配置	34
		4.4.5	替换 XIP 文件	36
	4.5	固	件生成及 IDE 烧写	36
		4.5.1	固件生成	37
		4.5.2	IDE 烧写固件	37
	4.6	ID:	E 使用技巧	37
		4.6.1	头文件路径添加	37
		4.6.2	代码优化等级的调整	38
_	女 丰	吉明		30

# 1. Easy-RT1052 开发套件简介

# 1.1 i.MX RT1050 跨界处理器

作为成千上万的智能物联网产品的核心,嵌入式处理器的独特之处在于其对功能和性能的精确定制。传统嵌入式处理器与千变万化的物联网需求之间存在的差距正在不断增大,人们需要处理器提供更低功耗、更经济的可扩展性,更强的计算性能、更高的安全性,以及更好的用户体验。为满足这一需求,恩智浦打破应用处理器和 MCU 之间的界限,打造一类新的"跨界"嵌入式处理器。这种新型应用处理器采用了 MCU 内核,但基于应用处理器的架构方式,既能实现应用处理器的高性能和丰富功能,同时又兼具传统 MCU 的易用性和实时低功耗运行特性,填补高性能与易用性之间的空白。

### 1.1.1 概述

### 1. i.MX RT 跨界处理器

凭借数十年来为消费者、工业和物联网市场提供 MCU 和应用处理器的领导经验,恩智浦开发出了全新嵌入式处理器系列—i.MX RT 跨界系列处理器。新型的 i.MX RT 跨界系列处理器基于 ARM® Cortex®-M7 内核,采用了 i.MX 6ULL 应用处理器架构,囊括了上文提到的跨界处理器的所有特性,提供了市场上首款真正能够填补 MCU 和应用处理器之间空白的解决方案。它把应用处理器级别的性能带到了 MCU 领域,能够支持实时、高性能处理功能,例如摄像头和显示功能。

### 2. 高性能、实时处理

i.MX RT 系列的内核运行速度可达 600 MHz(相比之下,现有 MCU 的最高速度只有 400MHz)。这是目前具有最高性能水平的 Cortex-M7 解决方案,可提供 3015 CoreMark®/1284 DMIPS (> 2 DMIPS/MHz)。得益于高密度的 512KB TCM SRAM, i.MX RT 系列还能够实现 超快的实时响应。其中断延迟低至 20 纳秒,在全世界所有基于 ARM Cortex 的产品中,这是最短的中断延迟,这都归功于高性能和 Cortex-M7 内核的强强组合。总而言之, i.MX RT 跨界处理器的性能可达竞争产品的 2 倍。



i.MX RT 系列不仅具备应用处理器级别的性能,它们还提供基于 MCU 的产品设计空间 里所具备的关键可用性特性:易用性、低成本以及与现有实时软件基础设施和工具链的兼容性。

i.MX RT 系列旨在让基于 MCU 的开发人员能够轻松使用这些新型跨界处理器,而无需投入大量精力来开发新软件实现工具或学习更高级别的操作系统,例如 Linux 或 Android。现有的 MCU 客户能够利用 i.MX RT 系列大幅提升性能功能,同时继续使用他们现有的工具链(例如 MCUXpresso、 IAR 和 Keil)。

i.MX RT 还支持使用恩智浦的 FreeRTOS、SDK、ARM® mbed 以及提供软件库、在线工 具和相应支持的全球 ARM 生态系统来实现快速原型制作和开发。使用与 Arduino™硬件盾板兼容的低成本评估套件 (EVK),还可以进一步加快开发速度。此外,i.MX RT 解决方案还采用单电压输入来简化电路设计。

### 1.1.2 特性

- 系统:
  - -ARM® Cortex®-M7 处理器,最大主频 600 MHz;
  - -32KB I-Cache, 32KB D-Cache;
  - 一浮点运算单元 (FPU), 支持 VFPv5 架构;
  - -MPU, 最大支持 16 个分区;
  - -支持串行线调试(SWD)模式与JTAG模式。
- 实时的低延迟响应:
  - -512KB 紧耦合内存(与片上 RAM 共享);
  - 一最快速的实时响应,延迟低至 20ns。
- 外部存储接口:
  - **−8/16** 位 SDRAM;
  - -8/16 位 SLC NAND Flash, 需要使用软件 ECC;
  - -SD/eMMC;
  - -SPI NOR Flash:
  - 一并行 NOR Flash, XIP 支持;
  - -双通道 QSPI Flash, XIP 支持。
- 定时器:
  - -2 路 32 位通用可编程定时器 (GPT), 4 通道, 支持匹配和捕获功能:
  - -4 路 16 位周期中断定时器 (PIT);
  - -4 路 16 位 Quad 定时器, 4 通道, 支持匹配和捕获功能, 正交解码功能;
  - -4 路 16 位 FlexPWM 定时器, 最大支持 8 通道 PWM 输出;
  - 一4路正交编解码器。
- 8/16/24 位 LCD 接口,支持最大 800×480 分辨率,支持 8/16-bit MPU/8080 接口。
- 音频接口:
  - -S/PDIF 输入输出接口:

- -3 通道同步音频接口(SAI), 支持 I2S、AC97、TDM 和 codec/DSP 接口;
- -MQS 接口用于中等质量的音频解码。
- 2D 图像处理引擎,支持图像的混合、旋转、缩放、颜色格式转换等操作。
- CSI 接口,支持 8/16/24 位数据输入。
- 连接性:
  - -2 路 USB2.0 OTG 控制器,芯片内部包含 USB-PHY 接口;
  - -2 路 uSDHC 接口, 支持 MMC/SD/SDIO 设备;
  - -一个 10/100M 以太网接口,支持 IEEE1588;
  - -8 路 UART:
  - -4 路 I2C;
  - -4 路 SPI:
  - −2 路 FlexCAN;
- 安全性:
  - 一数据协处理器(DCP), 支持 AES-128、SHA-1、SHA256、CRC32 计算;
  - 一总线加密引擎 (BEE), 支持 QSPI 的运行时解密;
  - 一随机数产生器;
  - -安全非易失存储(SNVS)。
- 调试接口:
  - —ARM CoreSight 和 Trace 组件;
  - 一支持 JTAG 和 SWD 调试接口。

# 1.2 Easy-RT1052 开发套件

Easy-RT1052 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 MIMXRT1052,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、工业技术设计、电动机控制和电源转换等场合。图 1.1、图 1.2 为 Easy-RT1052 实物图及相应的接口说明。



图 1.1 Easy-RT1052 实物图

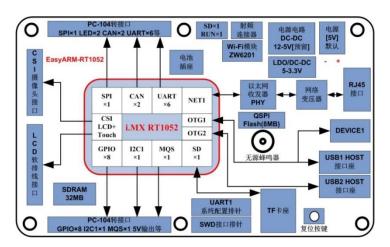
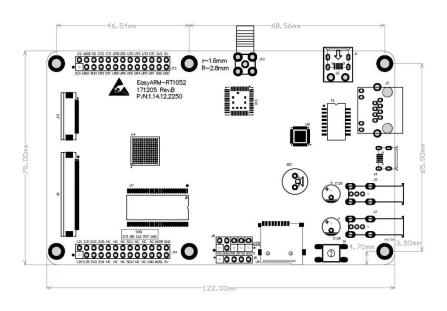
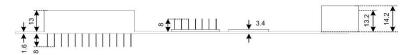


图 1.2 Easy-RT1052 接口说明

Easy-RT1052 机械尺寸如图 1.3 所示,单位为 mm。



A.俯视图(单位mm)



B.侧视图 (单位mm)

图 1.3 Easy-RT1052 机械尺寸

注 1: 该机械尺寸图主要给出一些重要的机械尺寸,需要更具体的机械尺寸图详见 "Easy-RT1052 硬件设计 V1.00/04. 机械尺寸/Easy-RT1052 机械尺寸.DWG" 文件。

# 1.1.1 功能框图

Easy-RT1052 功能框图如

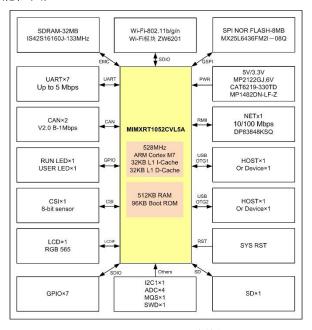


图 1.4 Easy-RT1052 功能框图

表 1 Easy-RT1052 参数表

产品型号	Easy-RT1052
处理器	MIMXRT1052
主频	528/600MHz
内存	32MB SDRAM
硬盘	8MB FLASH
LCD	RGB-565,支持高达 1366×768 分辨率
触摸屏	支持电阻和电容触摸屏
调试	1 路 SWD 调试接口
SD	1 路 SD 卡接口
CSI	1 路 8 位 CSI 接口
USB	2 路 USB OTG
以太网	1 路十兆百兆以太网口
Wi-Fi	1 路 Wi-Fi 无线模块,符合 IEEE 802.11 b/g/n 规范
蜂鸣器	1 路无源蜂鸣器
SPI	1路 SPI 接口,使用 CSI 接口时,不支持该路 SPI
LED	2路 LED 指示灯
CAN	2 路 CAN 接口,使用 CSI 接口时,只支持 CAN2
UART	7路 UART,使用 CSI 接口时,不支持 UART3/8,使用 ADC 通道 2/3 时,不支持 UART6
GPIO	7 路 GPIO
I2C	1 路 I2C
ADC	1 路 4 通道 ADC
MQS	1 路 MQS 接口

# 1.1.2 性能参数

表 1-2 绝对最大额定值

名称 符号		规格				说明
<b>口</b> 柳	14 5	最小	典型	最大	单位	<b>近</b> 地
非易失性存储	VDD SNVS IN	2.4	3.3	3.6	V	最先上电
及实时时钟电源	VDD_SNVS_IN	2.4	3.3	3.0	V	产生 1.1V 电压

DCDC 输入电源	DCDC_IN	3	3.3	3.6	V	经内部 DCDC 产生 1.1V 内核电 压
内核输入电源	VDD_SOC_IN	0.925	1.1	1.26	V	由 DCDC 输出 的 1.1V 进行供电
LDO 输入电源	VDD_HIGH_IN	2.8	3.3	3.6	V	经内部两个 LDO 产生 1.1V 及 2.5V 电压
USB VBUS 电源	USB_OTG1_V BUS USB_OTG2_V BUS	4.4	5	5.5	V	经内部 LDO 产生 3V 电压
IO 及 SD 卡 输入电源	NVCC_SD0 NVCC_SD1	3	3.3	3.6	V	SD 支持 1.8V 或 3.3V,此处为 3.3V
IO 及存储器 输入电源	NVCC_EMC	3	3.3	3.6	V	EMC 支持 1.8V 或 3.3V,此处为 3.3V
ADC 输入电源	VDDA_ADC	3	3.3	3.6	V	12 位 ADC
输入/出电压范围	Vin/Vout	-0.5	3.3	OVDD+0.	V	OVDD 为 IO 电源
工作组度英国	Tr:	-40	_	105	$^{\circ}$ C	工业级温度范围
工作温度范围	Tj	0	_	95	$^{\circ}\!$	消费级温度范围

注1: 超出所列的范围可能会导致设备永久损坏。

表 1-3 系统时钟列表

⟨z ĭh	ゲロ	规格				2H 00
名称	符号	最小	典型	最大	单位	说明
XTALI	fxtal	_	24	_	MHz	de Merc dell
						电源轨 NVCC_PLL
高电平 DC 输入	Vih	0.88		1.1	V	使用外部有源晶
低电平 DC 输入	Vil	0	_	0.2	V	振作为时钟输入 时

RTC_XTALI	fckil	_	32.768	_	KHz	. I. Ner-fel
						电源轨 VDD_SNVS_CAP
高电平 DC 输入	Vih	0.88	l	1.1	V	使用外部有源晶
低电平 DC 输入	Vil	0	_	0.2	V	- 振作为时钟输入 时

表 1-4 GPIO 直流参数

名称	符号		说明		
白仦	切写	最小	最大	单位	近り
高电平输出电压	$V_{\mathrm{OH}}$	NVCC-0.15	_	V	测试条件见注
低电平输出电压	V <sub>OL</sub>	_	0.15	V	测试条件见注
高电平输入电压	$V_{\mathrm{IH}}$	0.7*NVCC	NVCC	V	_
低电平输入电压	V <sub>IL</sub>	_	0.3*NVCC	V	_
输入迟滞电压	VHYS_LowVDD	250	_	mV	NVCC=1.8V 时
输入迟滞电压	VHYS_HighVDD	250	_	mV	NVCC=3.3V 时
施密特触发器 VT+	VTH+	0.5*NVCC	_	mV	-
施密特触发器 VT-	VTH-	_	0.5*NVCC	mV	-

注1: 芯片管脚上的过冲和欠冲必须保持在0.6V以下,且过冲/欠冲的持续时间不得超过系统时钟周期的10%。过冲/欠冲必须通过电路板布局、传输线阻抗匹配、信号线路终端或者其他方法控制。不符合此规范可能会影响设备的可靠性或对器件造成永久损坏。

# 1.1.3 接口说明

# 1. 指示灯及蜂鸣器

为方便用户了解整个系统的运行状况,Easy-RT1052 设计了 4 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)、SD 卡插入指示灯 (丝印为 SD)。此外,电路还加入了一个无源蜂鸣器 BZ1 (丝印 BZ),用于提示系统启动的状态,具体功能描述见表 1-5。

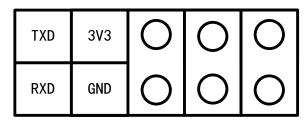
表 1-5 指示灯及蜂鸣器

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	系统正常运行时,此灯会以一定频率闪烁

SD	SD 卡插入指示灯	当 SD 卡安全插入时,此灯常亮
BZ1	蜂鸣器	系统正常启动时蜂鸣器会鸣叫、SD 卡固件更新完成后会 鸣叫

### 2. 配置说明

Easy-RT1052 支持 1 个手动复位按键、3 个功能按键,4 个功能排针,3 对与系统相关的配置排针,排针可用短路冒进行通断切换,使系统进入 SD 卡启动、Device 下载模式,具体功能分别见表 1-6 所述。



USB\_BT SD\_BT HOST1

图 1.5 调试串口与 BOOT 配置引脚 表 1-6 功能配置说明

丝印名称	默认配置	功能描述	
TXD	-	串口1发送引脚	
RXD	-	串口1接收引脚	
3V3	-	3V3 电源	
GND	-	电源地	
USB_BT	断开	断开:从 SPI Flash 启动系统 短接:从 USB DEVICE 升级程序	
SD	断开	断开:从 SPI Flash 启动系统 短接:从 SD 卡启动	
HOST1	-	预留跳线帽,暂时未定义	

# 3. 接口说明

Easy-RT1052 将 RT1052 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-RT1052 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 1.6 为 J13/14 管脚序号。

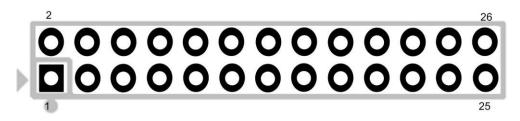


图 1.6 PC-104 转接口管脚定义

# 4. J13 接口管脚说明

表 1-7 J13 接口管脚说明

位号	丝印名称	默认配置	功能描述	备注
1	CLK		SPI3 时钟信号	
2	CS	SPI3	SPI3 片选信号	与 CSI 接口为复用关系
3	MISO	3F13	SPI3 主入从出	与 CSI 按口內复用天东
4	MOSI		SPI3 主出从入	
5	RUN	GPIO	运行正常指示灯	使用 Wi-Fi 功能时悬空
6	USER	GFIO	运行错误指示灯	使用 Wi-ri 功能的总主
7	CR2	CAN2	CAN2 接收数据线	符合 CAN2.0 B 规范
8	CT2	CANZ	CAN2 发送数据线	17日 CAINZ.U B 水烷
9	CR1	CAN1	CAN1 接收数据线	符合 CAN2.0 B 规范
10	CT1	CANI	CAN1 发送数据线	17日 CAINZ.U B 水烷
11	UR8	AUART8	应用串口 8 接收数据线	波特率高达 5 Mbps
12	UT8	AUAKIS	应用串口8发送数据线	仮付平同心 3 Mops
13	UR5	AUART5	应用串口5接收数据线	油柱交直计 5 Mbms
14	UT5	AUARIS	应用串口 5 发送数据线	波特率高达 5 Mbps
15	UR4	AUART4	应用串口 4 接收数据线	沙快变百计 5 Mbms
16	UT4	AUAR14	应用串口 4 发送数据线	波特率高达 5 Mbps
17	UR3	AUART3	应用串口 3 接收数据线	波特率高达 5 Mbps
18	UT3		应用串口3发送数据线	仮付平同心 3 Mops
19	UR2	AUART2	应用串口 2 接收数据线	
20	UT2		应用串口 2 发送数据线	油特家直升 5 Mb-s
21	UR1	AUART1	应用串口1接收数据线	波特率高达 5 Mbps
22	UT1		应用串口1发送数据线	

23	GND	电源地	功能固定,不可改变	只作为输出
24	3V3	3.3V 输出	功能固定,不可改变	<b>州下</b> 为制山
25	GND	电源地	功能固定,不可改变	只作为输出
26	5V	5V 输出	电源 5V 输出	六15万制山

# 5. J14 接口管脚说明

表 1-8 J14 接口管脚说明

位号	丝印名称	默认配置	功能描述	备注
1	1.08	GPIO	GPIO1_IO08	默认输入,100K PU
2	1.19	GPIO	GPIO1_IO19	默认输入,启用保持器
3	1.20	GPIO	GPIO1_IO20	默认输入,启用保持器
4	1.21	GPIO	GPIO1_IO21	默认输入,启用保持器
5	2.31	GPIO	GPIO2_IO31	默认输入,启用保持器
6	3.02	GPIO	GPIO3_IO02	默认输入,启用保持器
7	3.03	GPIO	GPIO3_IO03	默认输入,启用保持器
8	NC		NC	
9	NC		NC	
10	NC		NC	
11	NC	悬空	NC	
12	NC		NC	
13	NC		NC	
14	NC		NC	
15	SDA1		I2C1 数据线	和中京學術區井田 1901
16	SCL1	I2C1	I2C1 时钟线	和电容触摸屏共用 I <sup>2</sup> C1
17	AI3		ADC1 通道 3	采样率高达 1MS/s
18	AI2		ADC1 通道 2	采样率高达 1MS/s
19	AI1	ADC1	ADC1 通道 1	USB_OTG1_ID 信号复 用
20	AI0		ADC1 通道 0	采样率高达 1MS/s
21	GND	电源地	GND	
22	NC	悬空	NC	

23	MQSL	MQS	MQS 音频左声道输出	通过两个 GPIO 生成类
24	MQSR		MQS 音频右声道输出	PWM 双通道中等音质音 频
25	5V	5V 输出	5V 输出	只作为输出
26	GND	电源地	GND	バトグ制山

# 6. J8 接口说明

J6 为 SWD 的接线座子,接口示意图如图 1.7 所示,SWD 接口管脚定义如

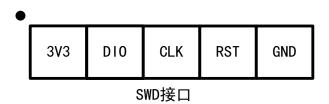


图 1.7 5 脚 SWD 接口

表 1-9 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	SWD_DIO	数据线	输入/输出
3	SWD_CLK	时钟	输入
4	SWD_RST	复位	输入
5	GND	电源地	

# 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-RT1052 开发板的入门指导文档。
- "3.集成开发环境"中包含了开发 Easy-RT1052 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-RT1052 后的 SDK 包。如图 2.1 所示是 SDK 包的目录结构。

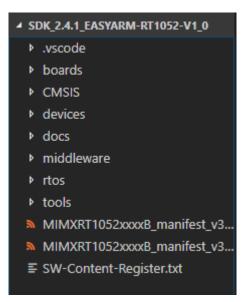


图 2.1 SDK 包目录结构

"CMSIS"目录中是 CMSIS 规范相关的头文件和库文件;

"devices"目录中是 MIMXRT1050 系列芯片的外设驱动库以及不同开发环境(Keil MDK、MCUXpressore、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、lwip 协议栈、SD/MMC 协议、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-RT1052 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,"emwin\_examples"目录中是 emWin 的例程,该例程需要 LCD 屏幕,"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;

# ■ boards ■ EasyARM-RT1052-V1\_0 □ cmsis\_driver\_examples □ demo\_apps □ driver\_examples □ emwin\_examples □ fatfs\_examples □ lwip\_examples □ project\_template □ rtos\_examples □ usb\_examples

图 2.2 例程目录

# 3. 例程使用指南

这里以 SDK 包中目录"boards\EasyARM-RT1052-V1\_0\driver\_examples\gpio\led\_output\" 例程为例介绍如何使用 SDK 包中的例程。

# 3.1 例程简介

使用 Keil MDK 打开工程文件"igpio led output.uvprojx"的结构如图 3.1 所示。

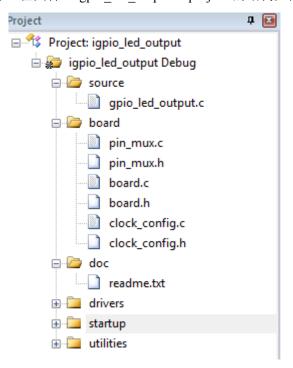


图 3.1 工程结构

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)。

# 3.2 Keil 工程设置与编译

### 1. 打开工程

点击如图 3.2 所示按钮选择不同配置的工程。

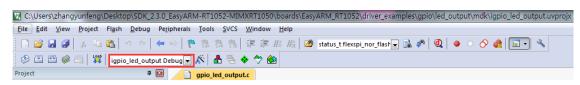


图 3.2 选择工程配置项

如图 3.3 所示每个工程中添加了四种配置选项:

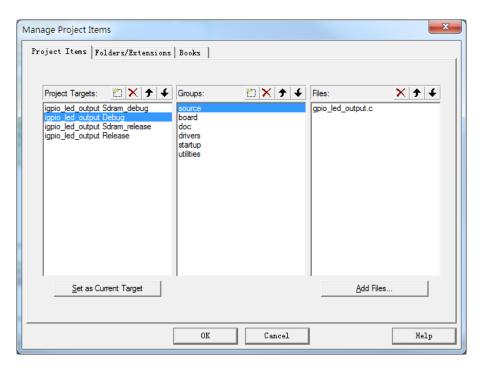


图 3.3 工程配置示意

- igpio\_led\_output Sdram\_debug: 生成加载到 sdram 中的固件,携带 debug 信息;
- igpio led output Debug: 生成加载到 ram 中的固件,携带 debug 信息;
- igpio\_led\_output Sdram\_release: 生成加载到 sdram 中的固件,移除 debug 信息;
- igpio\_led\_output Release: 生成加载到 ram 中的固件,移除 debug 信息;

### 2. 编译程序

点击编译按钮编译程序:



图 3.4 编译按钮

# 3.3 Keil 软件调试

CMSIS-DAP 是 ARM 官方推出的开源仿真器,支持所有的 Cortex-A/R/M 器件,支持 JTAG/SWD 接口。因为其开源免费以及与 ARM 公司渊源颇深等原因,民间流传了很多种以 CMSIS-DAP 为基础的调试工具,LPC-Link,GD-Link,ST-Link 和 ULink 2 等。

### 3.3.1 CMSIS-DAP

1. CMSIS-DAP 特点

CMSIS-DAP 仿真器具有以下特点:

- 完全开源,没有版权限制,所以相应的价格会很便宜;
- 无需驱动,即插即用;
- 在新版本的 DAP 里集成了串口,除了下载调试还能充当 USB 转串口模块,实现一机两用;

- 性能方面已经可以满足一般用户的需求。
- 2. KEIL 编译环境下的调试

首先,将 CMSIS-DAP 仿真器下载器与 Easy-RT1052 进行连接,Easy-RT1052 板子上的 JTAG 接口原理图如图 3.5 所示。

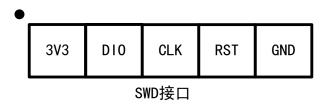


图 3.5 SWD 接口图

接下来就可以打开相应 Keil 工程文件, Keil 工程文件如图 3.6 所示。

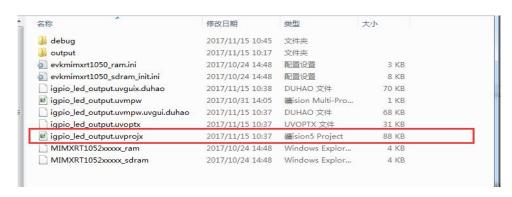


图 3.6 Keil 工程文件

双击打开如上图所示的 Keil 工程文件, 打开后的界面如图 3.7 所示。

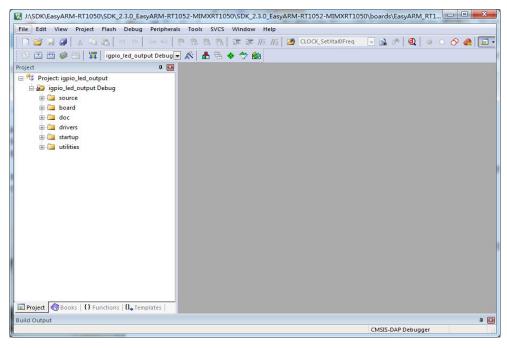


图 3.7 打开 Keil 工程文件

接下来对就可以对工程文件进行编译,点击如图 3.8 所示中的红色部分即开始编译。



图 3.8 工程文件编译

工程文件编译完成后将会出现如图 3.9 所示的提示。

```
compiling fsl_io.c...
compiling fsl_log.c...
compiling fsl_str.c...
compiling fsl_bbug_console.c...
compiling fsl_debug_console.c...
compiling fsl_debug_console.c...
compiling fsl_assert.c...
linking...
Program Size: Code=8384 RO-data=1272 RW-data=28 ZI-data=2084
"debug\ignio_led_output.out" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:04:51
Error: Target DLL has been cancelled. Debugger aborted !
```

图 3.9 工程文件编译成功

工程文件编译成功后即可进入到 Debug 调试界面,点击如图 3.10 所示红色部分即可开始调试。



图 3.10 软件调试

成功进入到软件调试后的界面如图 3.11 所示。

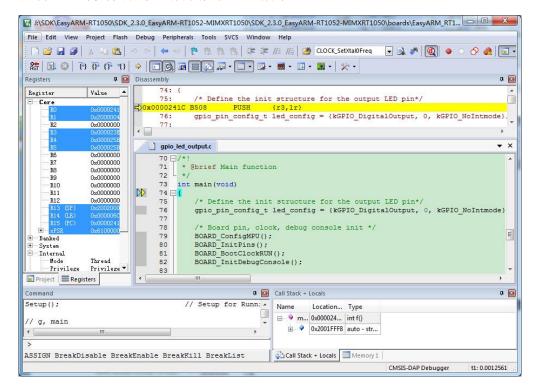


图 3.11 成功进入到调试界面

对下图 3.12 所示红色部分进行操作即可实现相应的软件的调试。

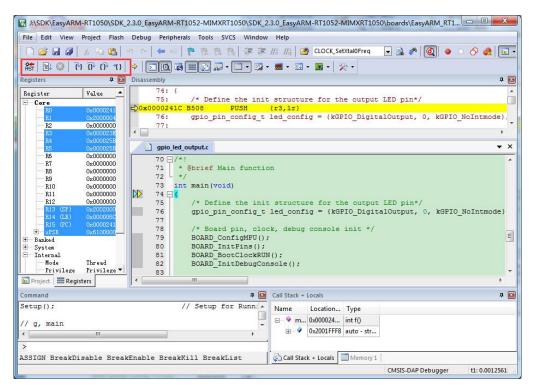


图 3.12 软件调试操作

### 3. IAR 编译环境下的调试

仿真器与板子的连接方式以及驱动安装在前面已经介绍,这里不做过多讲解,下面从直接打开 IAR 工程文件开始, IAR 的工程文件如图 3.13 所示。

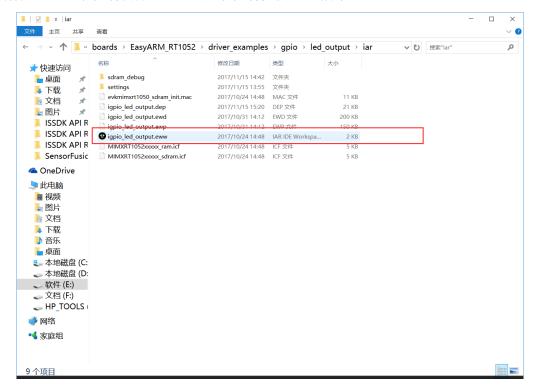


图 3.13 IAR 工程文件

双击打开如上图所示的 IAR 工程文件,打开后的界面如图 3.14 所示。

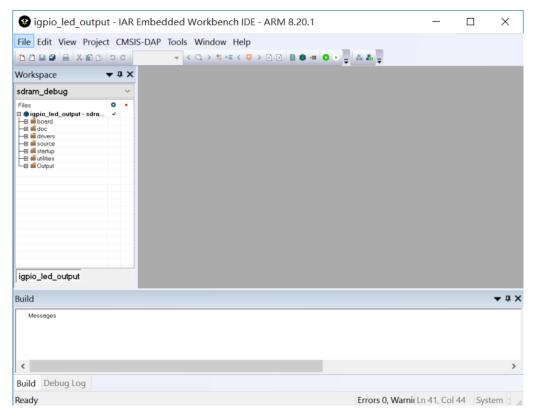


图 3.14 IAR 工程打开界面

接下来对就可以对工程文件进行调试了,点击如图 3.15 所示中的红色部分即开始调试。



图 3.15 开始调试操作

成功进入到软件调试后的界面如图 3.16 所示。

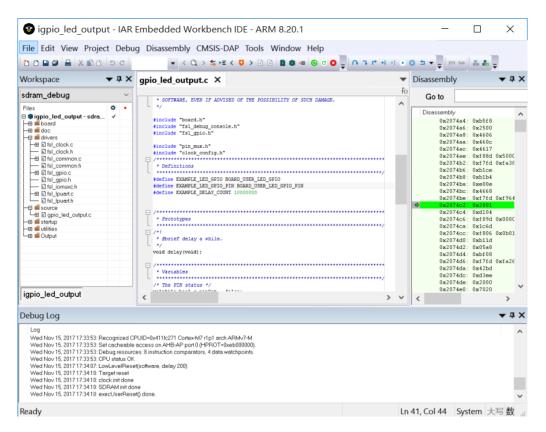


图 3.16 软件调试界面

对下图 3.17 所示红色部分进行操作即可实现相应的软件的调试。



图 3.17 软件调试操作

# 3.4 程序下载

程序下载请参考文档《i.MX RT1050 外部 Flash 启动与下载.pdf》。

# 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。

# 4.2 SDK 的获取及导入

这里介绍如何从 NXP 官网获取 rt1052 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 RT1052 的 SDK

### 1. 登陆 SDK 构建界面

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

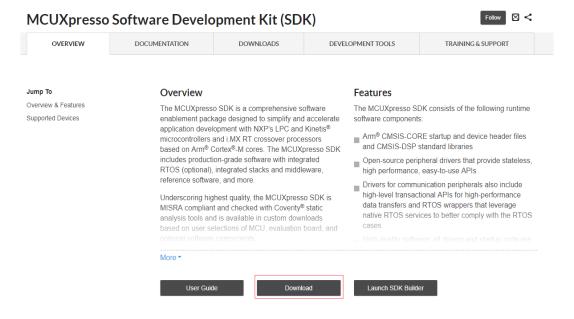


图 4.1 SDK 界面

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。

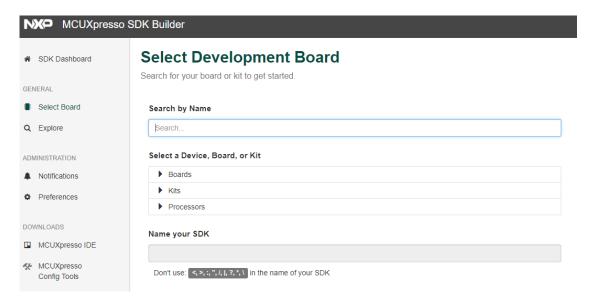


图 4.3 SDK 构建界面

# 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 MIMXRT1052xxxxB,然后点击 "Build MCUXpresso SDK",示范如图 4.4,可以根据需要修改 SDK 的名字。

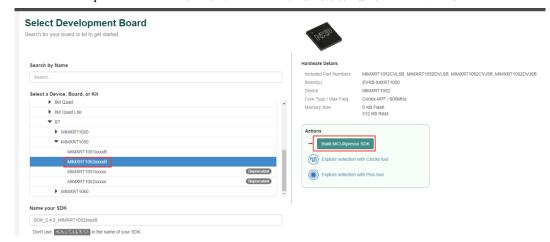


图 4.4 SDK 芯片选择

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"添加了四个组件,其中包含 FreeRTOS 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。

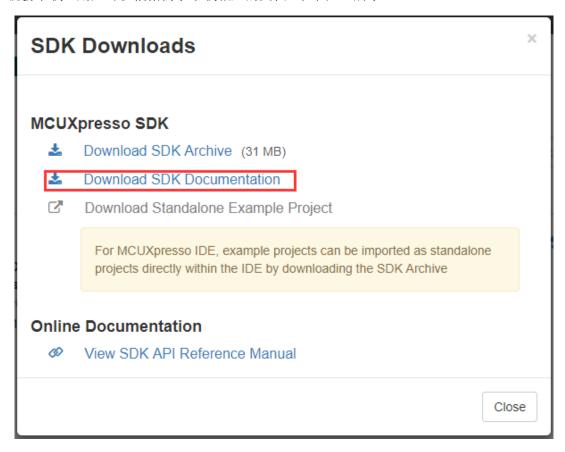


图 4.6 SDK Downloads 界面

### 4.2.2 向 IDE 导入 RT1052 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。

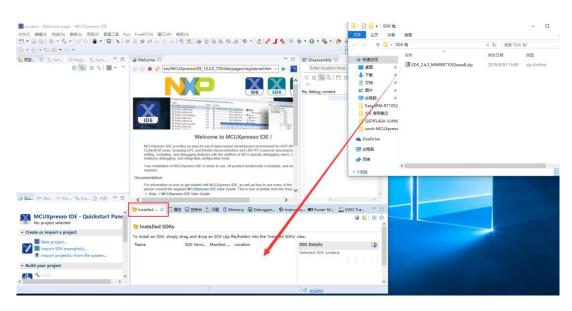


图 4.7 IDE 的 Installed SDKs

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。

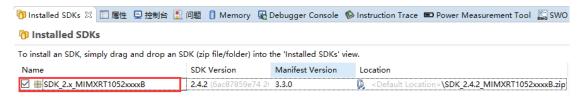


图 4.8 SDK 成功导入界面

# 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。

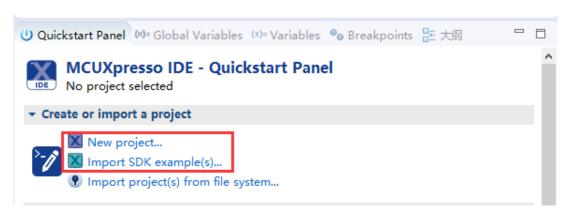


图 4.9 "create or import a project" 窗口

### 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。

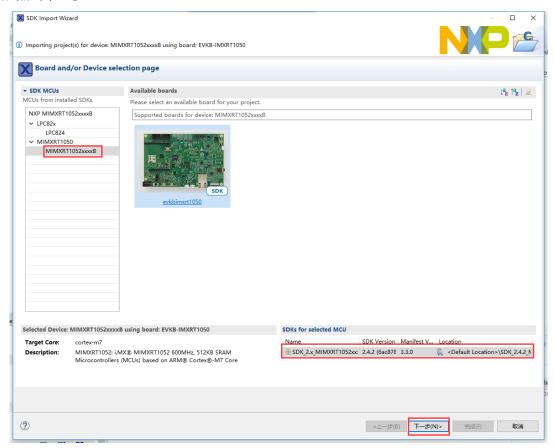


图 4.10 选择 SDK 创建工程

# 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。

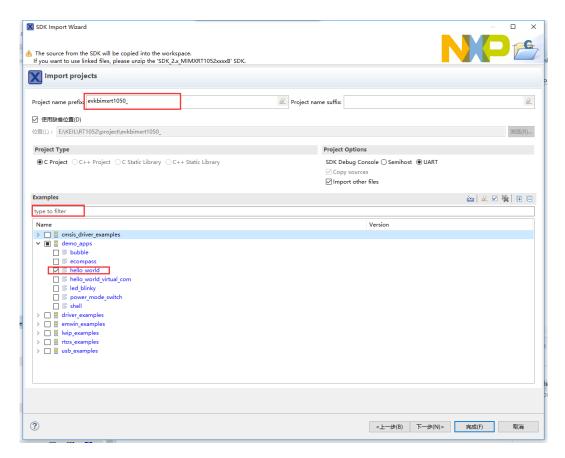


图 4.11 选择要导入的模板工程

# 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。

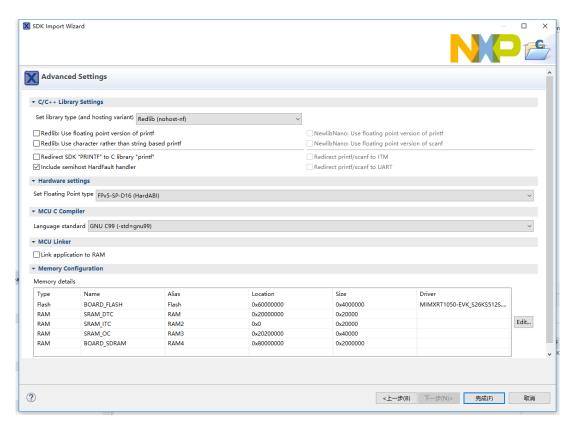


图 4.12 工程的高级设置窗口

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。

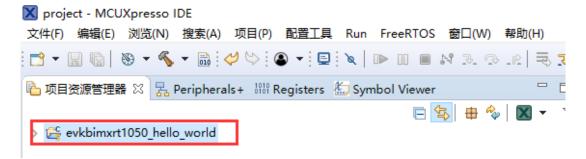


图 4.13 完成的导入模板工程

### 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 MIMXRT1052CVJ5B、创建 C 语言工程、调试控制台输出选择 UART。

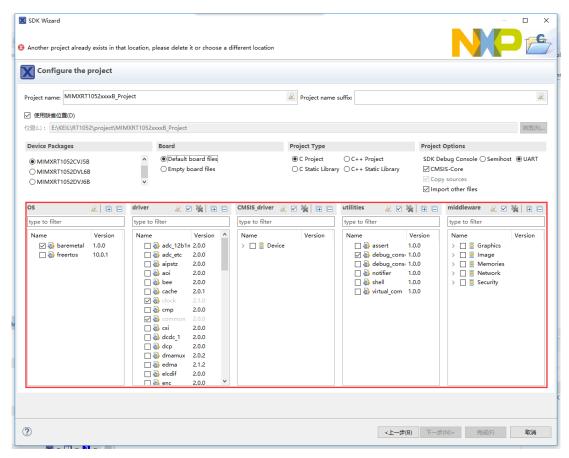


图 4.14 工程基本设置及选择组件

红框标注部分,就是用来选择该工程所要包含的组件,组件分为 5 类,每一类组件都可以提供搜索功能以用于在某类组件里快速找到需要的组件。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

在编译前,最好先设置 "Save automatically before build",这样就不用先手动保存后再编译了。设置过程: "窗口(W)" -> "首选项(P)" -> "常规" -> "工作空间",将图 4.15 中的红色框选项打钩。



图 4.15 设置 "Save automatically before build"

在"项目资源管理器"窗口选择要编译的工程,然后点击 5 即可编译,如图 4.16 所示,

正在编译 "evkbimxrt1050\_hello world" 工程。

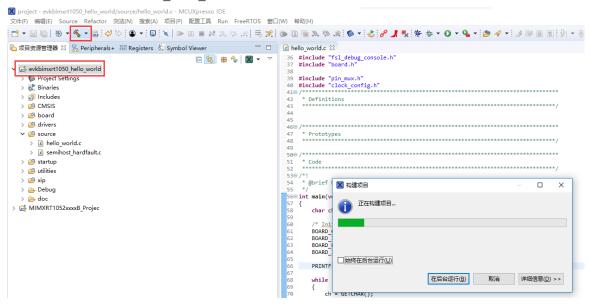


图 4.16 编译工程

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 4.17 所示红色线标注的信息信息输出,同时在该工程的目录下生成了一个 Debug 目录。

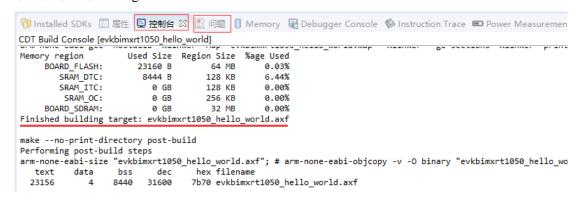


图 4.17 控制台的编译输出信息

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 4.18 中红色框标记即可。

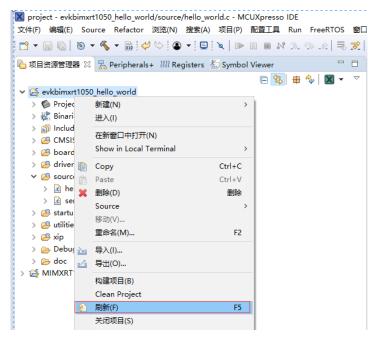


图 4.18 刷新工程以更新 axf 文件

# 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 0x600000000 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里。此时如果复位后程序却不运行,是因为 XIP 文件与板载 FLASH 不匹配,解决办法见后文"替换 XIP 文件"。

# 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 4.19 所示。

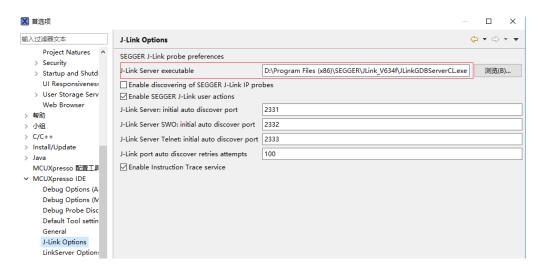


图 4.19 设置 JLinkGDBServerCL.exe 路径

### 3. 进入调试

选择"evkbimxrt1050\_hello\_world"工程,点击节即可,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。

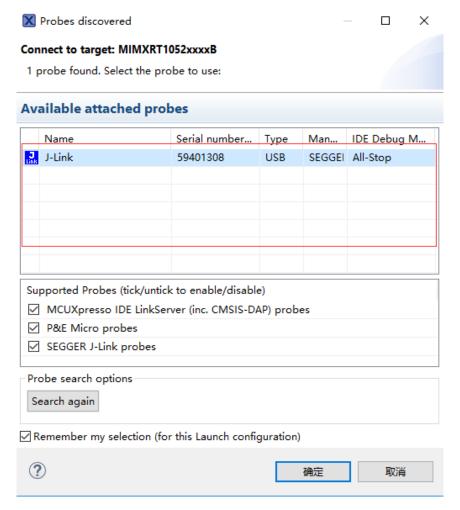


图 4.20 调试器发现及选择

接着就进入了调试界面,如图 4.21 所示。

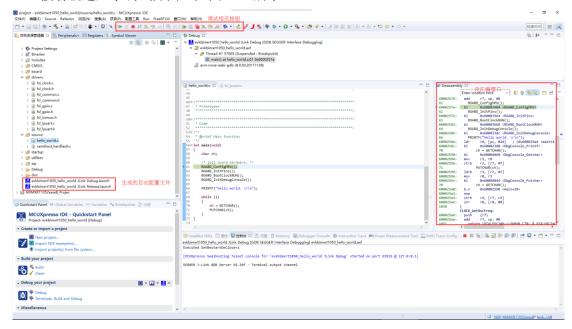


图 4.21 调试界面

# 4.4.3 调试代码说明

# 1. 调试时常用的窗口

如图 4.22 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。

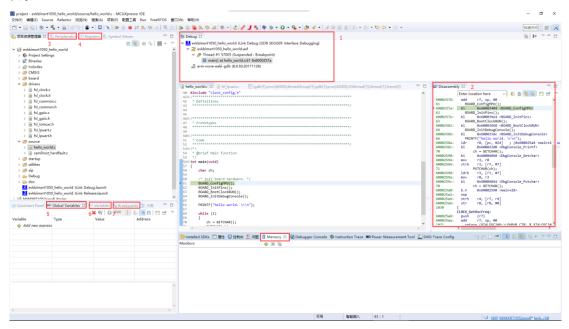


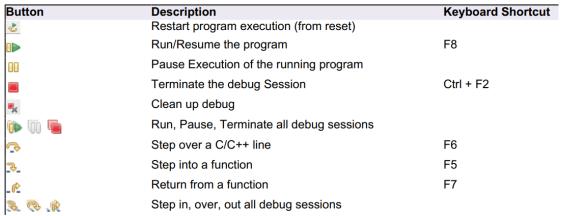
图 4.22 调试常用窗口

图示中,1可以查看堆栈调用,2为调试工具,3为外设寄存器窗口,4为通用寄存器窗口,5为全局变量窗口,6为变量窗口(主要显示局部变量),7为断点窗口(所有的断点都显示在该窗口)。

### 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



# 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 4.23 所示, 该界面可以 Jlink 连接方式, 设备等等。

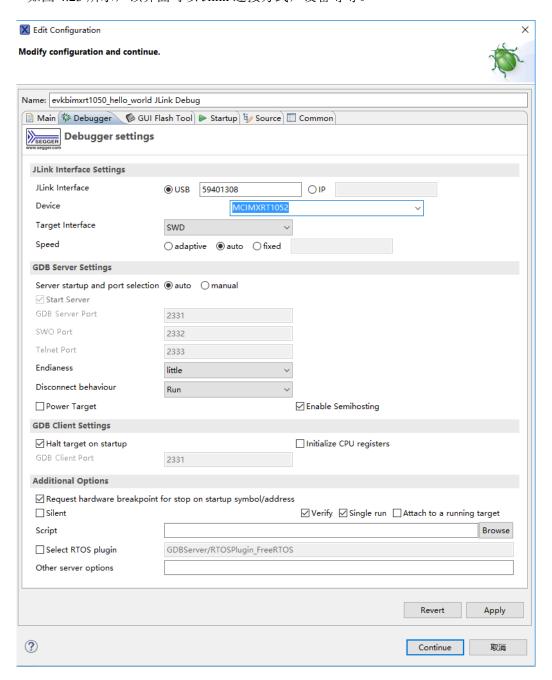


图 4.23 启动文件配置的 Debugger 界面

### 2. Starup 界面

如图 4.24 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。

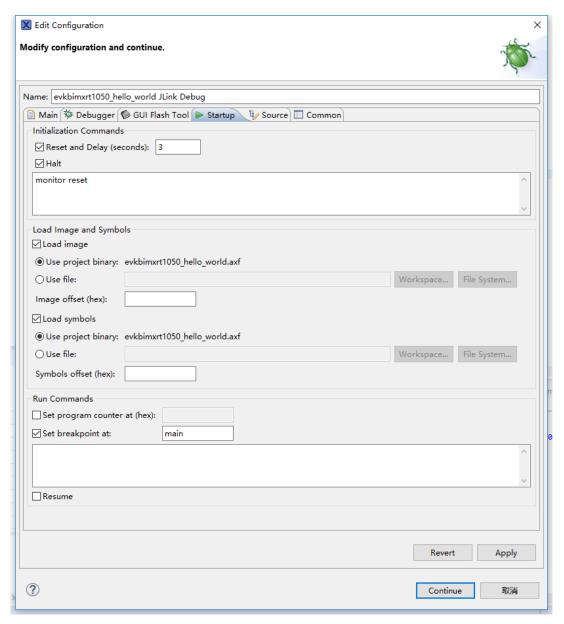


图 4.24 启动文件配置的 Starup 界面

# 4.4.5 替换 XIP 文件

立功科技的开发板使用的是 QSPIFLASH, NXP 开发板使用的是 hyperflash, SDK 的 xip 文件就是驱动 hyperflash。所以因 XIP 文件不匹配 Flash,虽然在调试时可以运行,但会导致 固件下载到 Flash 后不会运行。因此使用我司开发板需要替换 xip 文件,从立功科技适配 Easy-RT1052 后的 SDK 包中拷贝 xip 文件夹,替换掉工程中的 xip 文件夹即可。

替换之后,固件下载到 flash 就可以正常运行。

注意:替换之后,在编译之前需先将工程目录下的 Debug 文件夹删除掉,重新编译生成新的 Debug。

# 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 4.25 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。

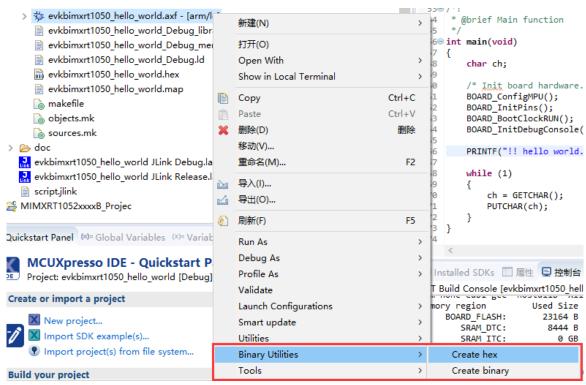


图 4.25 手动生成固件示例

### 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 4.26 红框标注的选项,即进行固件烧写。

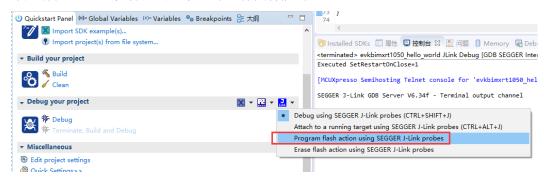


图 4.26 IDE 烧写固件

# 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 4.27 所示。

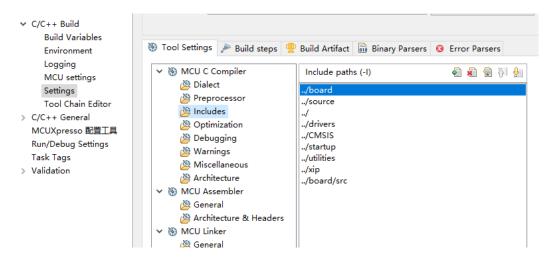


图 4.27 添加头文件路径

# 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 4.28 所示。默认是 0 级优化,即不优化。

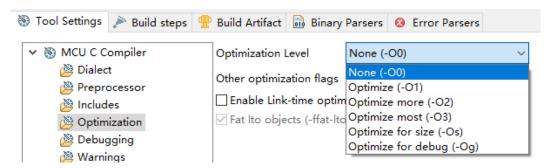


图 4.28 代码优化等级

# Easy-RT1052 快速入门指南

基于 Easy-RT1052 Rev.D 开发套件

User Manual

# 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。但介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.