Application Note

从 Renesas RL78 到基于 Arm 的 MSPM0 的迁移指南

Helic Chi, Zoey Wei, and Janz Bai

摘要

本应用手册可协助您从 Renesas RL78 平台迁移至德州仪器 (TI) MSPM0 MCU 生态系统。本文档介绍了 MSPM0 开发和工具生态系统、内核架构、外设注意事项以及软件开发套件。目的是突出两个系列之间的差异，并利用现有的 RL78 开发环境知识快速提升 MSPM0 系列 MCU 的性能。

内容

插图清单

表格清单

商标

1 MSPM0 产品系列概述

1.1 引言

MSPM0 微控制器 (MCU) 产品属于 MSP 高度集成的超低功耗 32 位 MCU 系列，基于增强型 Arm® Cortex®-M0+ 32 位内核平台运行。这些成本优化型 MCU 提供高性能模拟外设集成，支持扩展的工作温度范围并提供小尺寸封装。TI MSPM0 系列低功耗 MCU 包含具有不同模拟和数字集成度的器件，使工程师能够找到满足其工程需求的 MCU。MSPM0 MCU 系列将 Arm Cortex-M0+ 平台与超低功耗系统架构相结合，使系统设计人员能够在降低能耗的同时提高性能。

MSPM0 MCU 是 Renesas RL78 的替代产品，具有很强的竞争力。本应用手册通过比较器件功能和生态系统来帮助从 RL78 MCU 迁移到 MSPM0 MCU。

1.2 Renesas RL78 MCU 与 MSPM0 MCU 的产品系列比较

表 1-1. TI MSPM0Gx/Lx/Cx 和 Renesas RL78 系列的比较

		Renesas RL78 G 系列	Renesas RL78 L 系列	Renesas RL78 I 系列	Renesas RL78 F 系列	TI MSPM0 Gx 系列	TI MSPM0 Lx 系列	TI MSPM0 Cx 系列
内核		RL78 CPU 内核				Arm Cortex-M0+		
频率		16/20/24/32MHz	24MHz	24/32MHz	24/32/40MHz	80MHz	32MHz	24MHz
电源电压		1.6/1.8/2/2.7-5.5V、1.6-3.6V	1.6-5.5V、1.8-3.6/5.5V	1.7/1.9/2.4/2.7-5.5V、1.6-3.6V	2.7-5.5V、1.8-5.5V	1.62-3.6 V	1.62-3.6 V	1.62-3.6 V
温度		-40-125°C、-25-75°C	-40-85°C、-40-125°C	-40-85°C、-40-105°C、-40-125°C	-40-105°C、-40-125°C、-40-150°C	-40-125°C	-40-125°C	-40-125°C
存储器		768KB 至 1KB	256KB 至 8KB	512KB 至 8KB	512KB 至 8KB	128KB 至 32KB	64KB 至 8KB	16KB 至 8KB
RAM		高达 144KB	高达 32KB	高达 32KB	高达 4KB	高达 32KB	高达 4KB	1KB
GPIO（最大值）		130	79	76	130	60	28	18
模拟	ADC	高达 12 位 x 28 通道	高达 12 位 x 14 通道	高达 12 位 x 17 通道	高达 12 位 x 25 通道	2x 4Msps 12 位	1 个 1Msps 12 位 ADC	1x 1Msps 12 位 ADC（10 通道）
	DAC	高达 10 位 x 2 通道	高达 12 位 x 2 通道	12 位 x 1 通道 (RL/I1E)	8 位 x 1 通道	12 位	8 位	none
	比较器	高达 2 通道	高达 2 通道	高达 2 通道	高达 1 通道	3x 高速	1x 高速	none
通信（最大数量）	UART	4	4	4	5	4	2	1
	I2C	10	5	4	5	2（快速）	2（快速）	1
	SPI	0	4	1	4	2	1	1
	CAN	0	0	0	2	1 (CAN-FD)	0	0
	LIN	1（支持 UART）			3（支持 UART）	1（支持 UART）		
其他重要外设/特性		内部升压 LCD USB (RL/G1A) 单通道 PGA 蓝牙 (RL/G1D) 1% 振荡器	内部升压 LCD USB (RL78/L1C) 3 通道放大器	LCD 单通道 PGA 3 通道放大器 USB VBAT，Σ-Δ AFE	MATHACL，ASIL-B，150°C	2 个运算放大器 CAN-FD，USB，Fast4Msps sim-Sam ADC，数学加速	2 个运算放大器 LCD (L2228)	超小型 QFN 封装 (2x2)，0.5/0.65mm 间距封装，引脚与业内通用产品兼容
计时器数量		1/2/4/5	1/2/3	1/2/5	1/2	4	7	4
引脚数		16-128 引脚	32-100 引脚	20-100 引脚	20-144 引脚	20-100 引脚	16-80 引脚	8-48 引脚

表 1-1. TI MSPM0Gx/Lx/Cx 和 Renesas RL78 系列的比较（续）

	Renesas RL78 G 系列	Renesas RL78 L 系列	Renesas RL78 I 系列	Renesas RL78 F 系列	TI MSPM0 Gx 系列	TI MSPM0 Lx 系列	TI MSPM0 Cx 系列
安全性	CRC，RNG，AES 库，SHA 哈希函数库，RSA 库	CRC，AES GCM	CRC	CRC	CRC、TRNG、AES256	CRC	CRC
低功耗(1)	有效：低至 37.5 µA/MHz，停止：低至 0.2 µA	有效：低至 66 µA/MHz，停止：低至 0.23 µA	有效：低至 96 µA/MHz，停止：低至 0.23 µA	（未提及）	有效：85µA/MHz，待机：1.5µA	有效：71µA/MHz，待机：1µA	有效：100µA/MHz，待机：5µA

(1) RL78 停止模式类似于 MSPM0 关断模式（CPU、时钟、外设均关断）

此处提供了 RL78 和 MSPM0 的一些性能比较。以下几节提供了详细信息。

2 生态系统和迁移

MSPM0 MCU 由广泛的硬件和软件生态系统提供支持，随附参考设计和代码示例，便于您快速开始设计。MSPM0 MCU 还具有在线资源、MSP Academy 培训支持和 TI E2E™ 支持论坛提供的在线支持。

2.1 生态系统比较

表 2-1. 生态系统比较

功能	RL78 器件	MSPM0 器件
代码源	中间件 Renesas 闪存驱动程序/自编程库 驱动程序 操作系统	MSPM0-SDK（DriverLib、中间件、RTOS、代码示例）
IDE	e² studio CS+ CC-RL IAR	CCS IAR Keil
软件配置	Smart Configurator	SysConfig
闪存编程工具	Renesas Flash Programmer	UniFlash
编程器	PG-FP6	MSP-GANG
调试器	E2 Emulator E2 Emulator Lite EZ-CUBE	XDS110 J-LINK
硬件	快速原型开发板 目标板 入门套件	LP-MSPM0G3507 LaunchPad LP-MSPM0L1306 LaunchPadLP-MSPM0C1104 LaunchPad

图 2-1 显示了 MSPM0 生态系统概览。

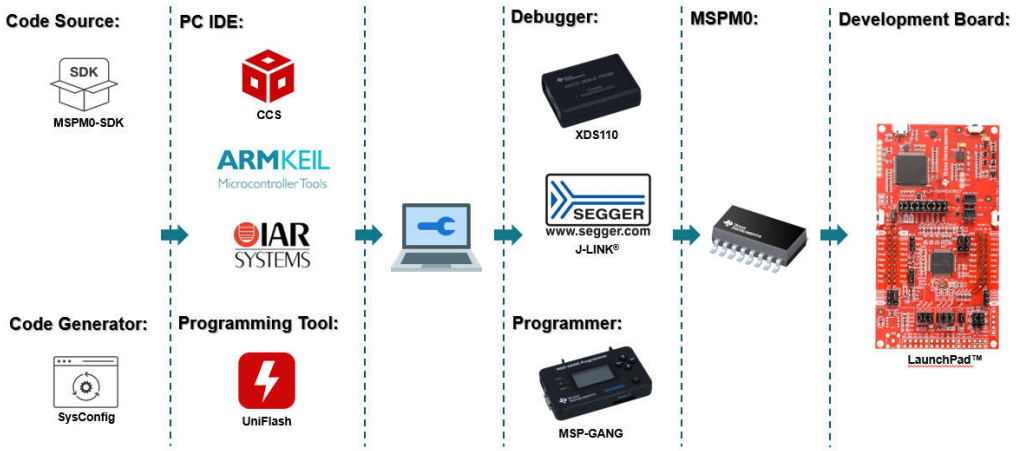


图 2-1. MSPM0 生态系统概览

2.1.1 MSPM0 软件开发套件 (MSPM0 SDK)

MSPM0 SDK 随附多种代码示例可供选择，使工程师能够在德州仪器 (TI) MSPM0+ 微控制器器件上开发应用。提供了各种示例来展示如何在每个受支持的器件上使用各功能区，这些示例可用作您开发自己的工程的起点。图 2-2 展示了 MSPM0 SDK 的结构。

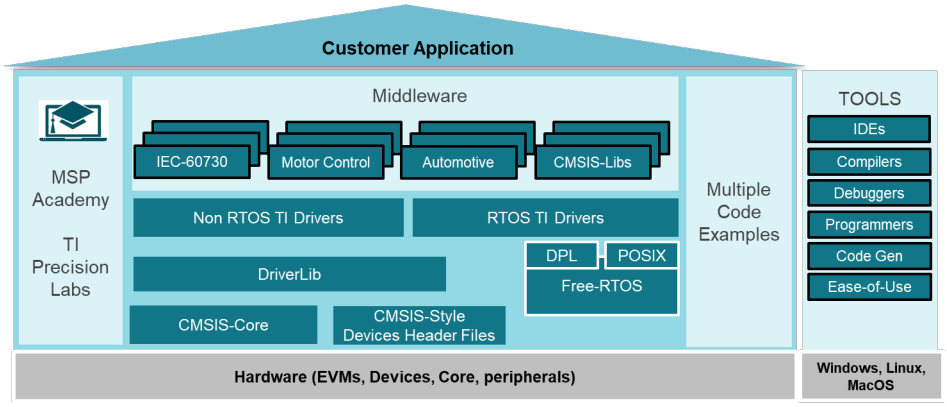


图 2-2. MSPM0 SDK 结构

可从 MSPM0-SDK 支持软件 | 德州仪器 TI.com.cn 下载 MSPM0 SDK。MSPM0 SDK 中包含四个文件夹：

示例：示例文件夹分为 RTOS 和非 RTOS 子文件夹（目前仅支持非 RTOS）。这些文件夹包含每个 LaunchPad™ 的示例，并根据功能进行整理，例如较低层的 Driverlib 示例、较高层的 *TI 驱动程序* 示例以及 GUI Composer、LIN、IQMath 等*中间件* 的示例。

文档：包含所有相关文档，包括用户指南和 API 指南。

源： 适用于所有驱动程序和中间件的源代码和库。

工具: 帮助开发和/或测试 MSPM0 应用的工具集。
虽然 Renesas RL78 支持大中型的广泛示例代码，例如 DSP、USB 驱动程序，但是没有用于代码开发程序的软件包，且 RL78 也没有示例代码，这意味着用户需要从头开始创建新工程并设置配置（例如调试器）。相比之下，MSPM0 SDK 将所有源代码与中间件和驱动程序库集成在一起，以便于开发。示例代码可帮助客户快速入门并详细了解 MCU 外设。

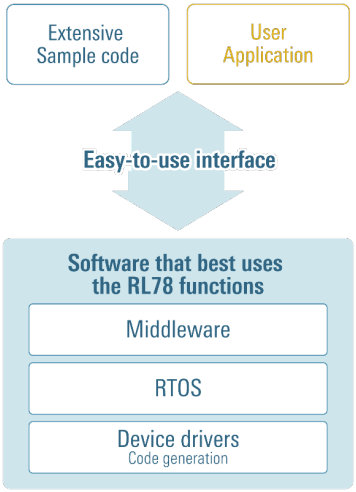


图 2-3. RL78 软件开发环境

表 2-2. 软件生态系统

功能	RL78 软件	MSPM0 SDK
寄存器级代码	否	否
驱动程序库	是	是
中间件	是	是
自编程	是	否
开箱即用代码	否	有
免费 RTOS	是	是

大多数 MSPM0 示例都支持 SysConfig，以便简化器件配置和加快软件开发。

其他参考文档如下所示：

- MSPM0 SDK 用户指南
- MSPM0 工具指南
- DriverLib API 指南

2.1.2 MSPM0 支持的 IDE

集成开发环境 (IDE) 是帮助程序员高效开发软件代码的软件应用程序，通常包括编辑器、编译器、调试器等。

RL78 的典型 IDE 是 e^{2}studio。它可以下载示例代码并具有易于使用的 Eclipse 代码编辑器。对于 TI，强烈建议使用 Code Composer Studio IDE (CCS)，因为其支持 TI 的微控制器 (MCU) 和嵌入式处理器产品系列。由于 CCS 也是基于 Eclipse 的 IDE，因此用户更容易上手。尤其是 CCS 包含一系列用于开发和调试嵌入式应用程序的工具，其中包括优化的 C/C++ 编译器、源代码编辑器、工程构建环境、调试器、性能评测工具和许多其他功能。此外，CCS 的使用完全免费。

表 2-3. CCS 和 e^2 Studio 之间的比较

IDE	CCS	e^{2}studio
许可	免费	免费
编译器	TI Arm Clang/GCC	CC-RL/LLVM

表 2-3. CCS 和 e² Studio 之间的比较（续）

IDE	CCS	e²studio
IDE 中集成的电流消耗分析工具	EnergyTrace	Renesas QE
外设的 API 函数帮助	不支持	支持
显示语言	英语	英语 日语 中文
转换文件	十六进制文件 二进制文件 Motorola S-record 文件 Ti_txt 文件	十六进制文件 二进制文件 Motorola S-record 文件
生成代码 GUI	SysConfig	Smart Configuration

CCS 集成了 SysConfig 的 MSPM0 器件配置和自动代码生成功能，并在集成式 TI Resource explorer 中集成了 MSPM0 代码示例和 Academy 培训。此外，CCS 提供一体式开发工具体验。

除 CCS 之外，表 2-4 中列出的业界通用 IDE 也支持 MSPM0 器件。

- CCS：https://www.ti.com.cn/tool/cn/CCSTUDIO
- IAR：https://www.iar.com/
- Keil：https://www.keil.com/

表 2-4. MSPM0 支持的 IDE 概览

IDE	CCS (Eclipse)	IAR	Keil
许可	免费	付费	付费
编译器	TI Arm Clang GCC	IAR C/C++ Compiler™ for Arm	Arm 编译器版本 6
磁盘大小	3.44G (ccs1220)	6.33G (Arm 8.50.4)	2.5G (µVision V5.37.0)
XDS110	支持	支持	支持
J-Link	支持	支持	支持
EnergyTrace	支持	否	否
MISRA-C	否	支持	否
安全性	否	支持	否
ULINKplus	否	否	支持
功能安全	否	支持	支持

节 2.2.2.2 介绍了 CCS 的使用和一些功能。其他参考资料如下所示：

- CCS 快速入门指南
- CCS
- CCS 培训视频
- CCS 用户指南
- IAR 快速入门指南
- IAR 培训视频
- IAR 用户指南
- Keil 快速入门指南
- Keil 培训视频
- Keil 入门

2.1.3 SysConfig

与 Smart Configuration 类似，SysConfig 是一个直观而全面的图形实用程序集合，用于配置引脚、外设、无线电、子系统和其他组件。SysConfig 可帮助您直观地管理、发现和解决冲突，以便您有更多时间创建差异化应用程序。该工具的输出包括 C 头文件和代码文件，这些文件可与 MSPM0 SDK 示例配合使用，或用于配置定制软件。

SysConfig 集成在 CCS 中，但也可以与 Keil 和 IAR 一起使用。可通过以下 URL 下载 SysConfig：SYSCONFIG IDE、配置、编译器或调试器 | 德州仪器 TI.com.cn。

此外，SysConfig 可以在没有 IDE 的情况下运行。独立版本可用于生成代码和评估器件功能，但不能运行示例。

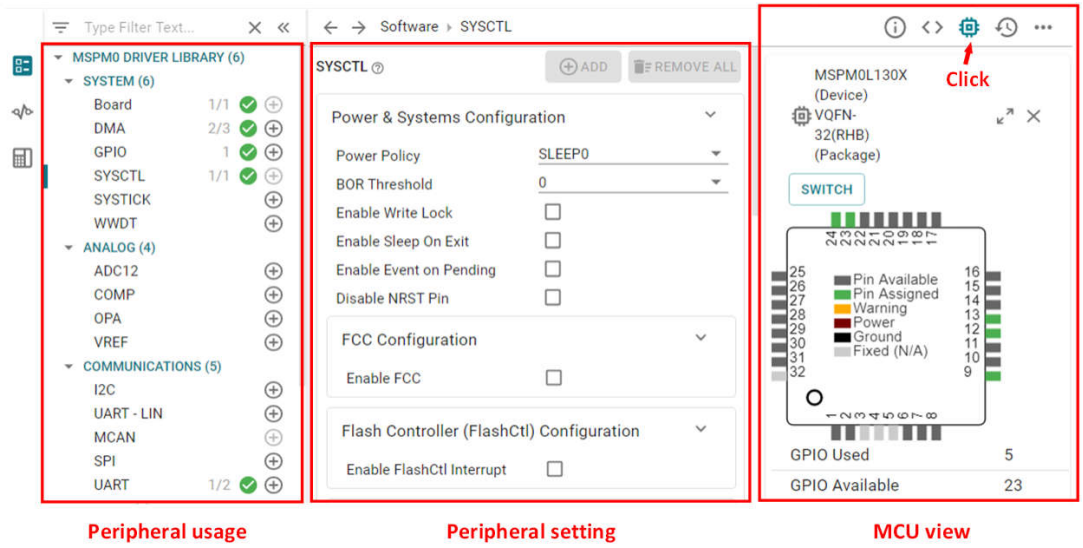


图 2-4. MSPM0 SysConfig

有关详细信息，请参阅 MSPM0 SysConfig 指南。

与 Smart Configuration 相比，SysConfig 具有以下优势：

- SysConfig 中的外设分类更清晰，整体界面更易于辨识，人机交互界面很出色，如图 2-5 所示。
- SysConfig 会显示硬件原理图，并提供每个外设的详细说明，以及所有 GUI 界面配置的详细功能说明。
- Smart Configuration 中断和引脚设置位于单独的模块中，比较杂乱，不方便开发。TI 的 SysConfig 可以直接在特定的外设模块中配置，从而使开发变得简单明了。图 2-6 展示了中断设置的比较情况。SysConfig 可以实现多种外设联动配置，例如 ADC 模块中的事件配置。
- 在配置外设的特定功能时，SysConfig 可以实时显示代码变化，而在 Smart Configuration 中无法立即看到。

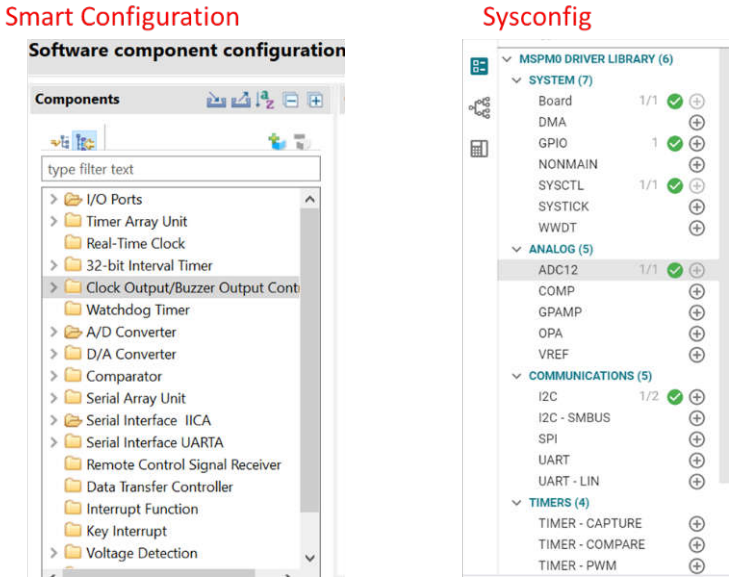


图 2-5. 外设列表比较

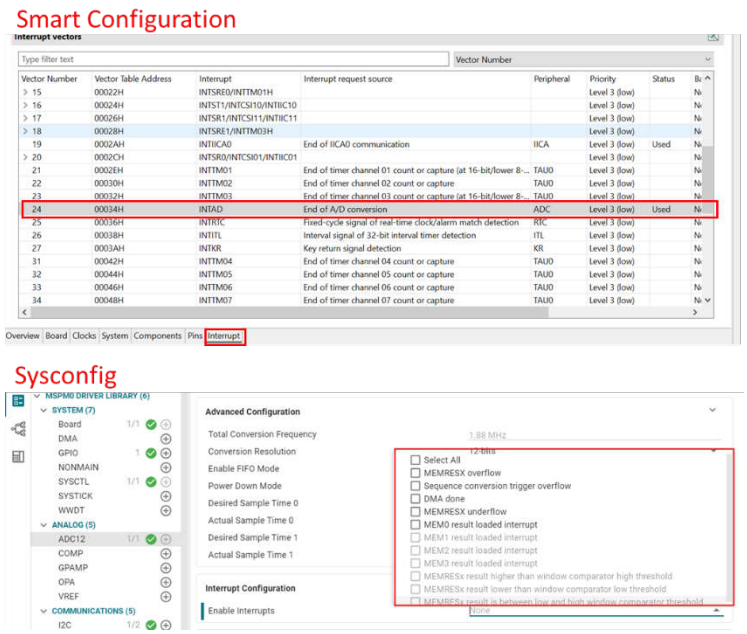


图 2-6. 中断设置比较

2.1.4 调试工具

对于 RL78，TOOL0 引脚用于通过专用的单线 UART 进行操作，以连接调试器（如 E2 Emulator 或 E2 Emulator Lite）和 RL78/F23。RL78 的典型调试器工具是 E2 Emulator，它支持测量电流消耗、监控点和设置外部触发器输入/输出。

对于 MSPM0，调试子系统 (DEBUGSS) 将串行线调试 (SWD) 两线制物理接口连接到器件内的多个调试功能。MSPM0 器件支持调试处理器执行情况、器件状态和电源状态（使用 EnergyTrace 技术）。更多有关调试器连接的详细信息，请参阅图 2-7。

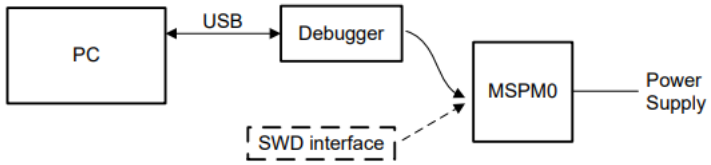


图 2-7. MSPM0 调试

MSPM0 支持用于标准串行线调试的 XDS110 和 J-Link 调试器。

德州仪器 (TI) XDS110 用于 TI 嵌入式处理器。XDS110 通过 TI 20 引脚连接器（提供适用于 TI 14 引脚、Arm 10 引脚和 Arm 20 引脚连接器的多个适配器）连接到目标板，通过 USB2.0 高速 (480Mbps) 连接器连接到主机 PC。XDS110 在单个单元中支持更广泛的标准（IEEE1149.1、IEEE1149.7、SWD）。所有 XDS 调试探针都支持所有具有嵌入式跟踪缓冲器 (ETB) 的 Arm 和 DSP 处理器中的内核和系统跟踪。有关详细信息，请参阅 XDS110 调试探针。

J-Link 调试探针是优化调试和闪存编程体验的最常见的选择。这得益于其创纪录的闪存加载程序、高达 3MiB/s 的 RAM 下载速度以及在 MCU 闪存中设置无限数量断点的能力。J-Link 还支持各种 CPU 和架构，包括 CortexM0+。有关详细信息，请参阅 J-Link 调试探针页面。

以下是支持 MSPM0 的 XDS110 和 J-LINK 调试器之间的不同功能汇总。

表 2-5. MSPM0 调试器比较

特性	XDS110	XDS110 OB (1)	J-Link
cJTAG (SBW)	√	√	√

表 2-5. MSPM0 调试器比较（续）

特性	XDS110	XDS110 OB (1)	J-Link
BSL(2) 工具	√	√	
反向通道 UART	√	√	2.5G (µVision V5.37.0)
电源	1.8 至 3.6 V	3.3/5 V	5V
IDE (3)：CCS	√	√	√
IDE：第三方 (4)	IAR/Keil	IAR/Keil	IAR/Keil

(1) XDS110 OB 是指板载 XDS110。
(2) BSL 表示引导加载程序。
(3) IDE 表示集成开发环境。
(4) 第三方包括 IAR/Keil。

2.1.5 LaunchPad

不同于具有入门套件、FPB 和目标板的 RL78，LaunchPad 开发套件是 MSPM0 的唯一评估模块。

LaunchPad 套件是易于使用的 EVM，其中包含在 MSPM0 上开始开发所需的一切内容。这包括一个板载调试探针，用于使用 EnergyTrace™ 技术进行编程、调试和测量功耗。MSPM0 LaunchPad 套件还具有板载按钮、LED 和温度传感器以及其他电路。40 引脚 BoosterPack™ 插件模块接头简化了快速原型设计，支持市面上的多种 BoosterPack 插件模块。您可快速添加无线连接、图形显示、环境检测等功能。

- LP-MSPM0G3507 LaunchPad 开发套件
- LP-MSPM0L1306 LaunchPad 开发套件

图 2-8 展示了 LaunchPad 的概览，其中包含 MCU 和 XDS110 调试器。您还可以在移除跳线后使用 J-Link 等其他调试器来调试 MCU。

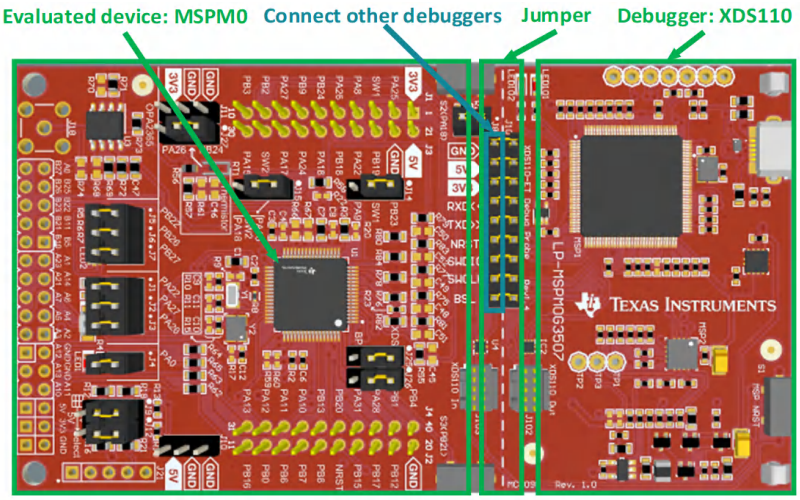


图 2-8. MSPM0G3507 LaunchPad 概览

跳线隔离块包含电源（GND、5V、3.3V）、UART（RXD、TXD）、复位引脚、ARM 调试通道（SWDIO、SWCLK）和 BSL。

除了跳线帽外，还可以使用位于 LaunchPad 右侧的标准 Arm Cortex 10 引脚连接器（如图 2-9 所示）进行烧写。Cortex 调试连接器支持 JTAG 调试、串行线调试和串行线查看器（使用串行线调试模式时通过 SWO 连接）运行。

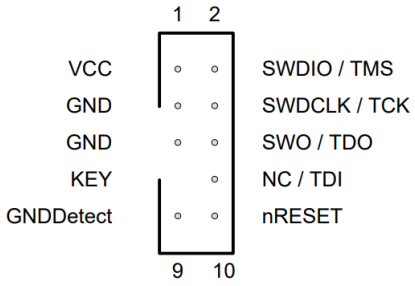


图 2-9. Arm Cortex 10 引脚定义

图 2-10 显示了 MSPM0G3507 LaunchPad 的一些特性功能。

该 LaunchPad 下侧的 BoosterPack 连接器用于直接插入特定功能模块并快速构建原型。除此之外，还可以单独使用 DuPont 线进行引出以便快速使用。该 LaunchPad 有一个用户定义的按钮（位于每一侧）、一个温度传感器、一个光传感器、一个单色 LED 和一个 RGB LED（位于下方）。

- LP-MSPM0G3507 LaunchPad 开发套件 LP-MSPM0G3507 评估板 | 德州仪器 TI.com.cn
- LP-MSPM0L1306 LaunchPad 开发套件 LP-MSPM0L1306 评估板 | 德州仪器 TI.com.cn

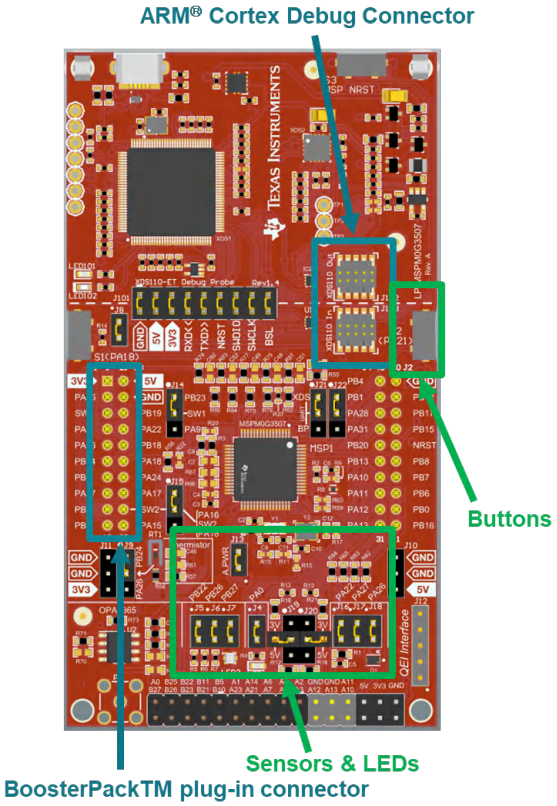


图 2-10. MSPM0G3507 LaunchPad 特性功能

2.2 迁移过程

为了顺利迁移到 MSPM0，详细过程将按图 2-11 所示的流程编写。以下各节将详细介绍每个步骤，并给出示例。

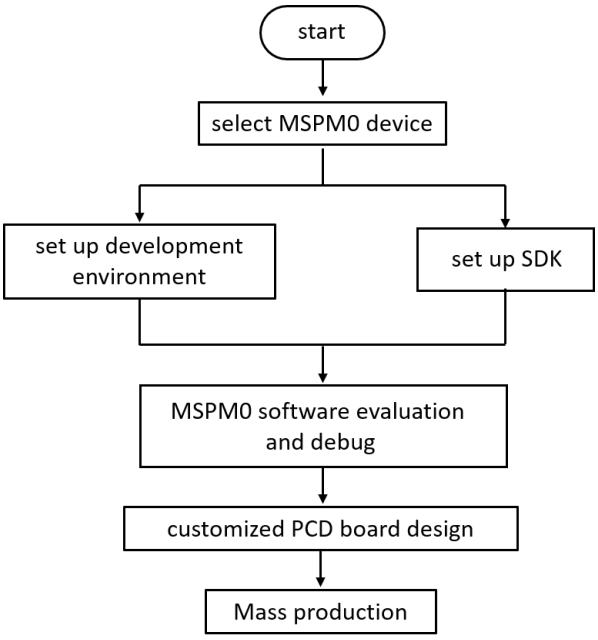


图 2-11. MSPM0 迁移流程图

2.2.1 步骤 1.选择合适的 MSPM0 MCU

迁移的第一步是为应用程序选择正确的 MSPM0 器件。图 2-12 所示为 MSPM0 L 和 G 产品系列，相关信息可在 TI 官方网站上找到。此外，图 2-13 展示了 MSPM0 C 产品系列。这两个产品系列都根据存储器和封装来区分器件，因此更容易进行简单的选择。

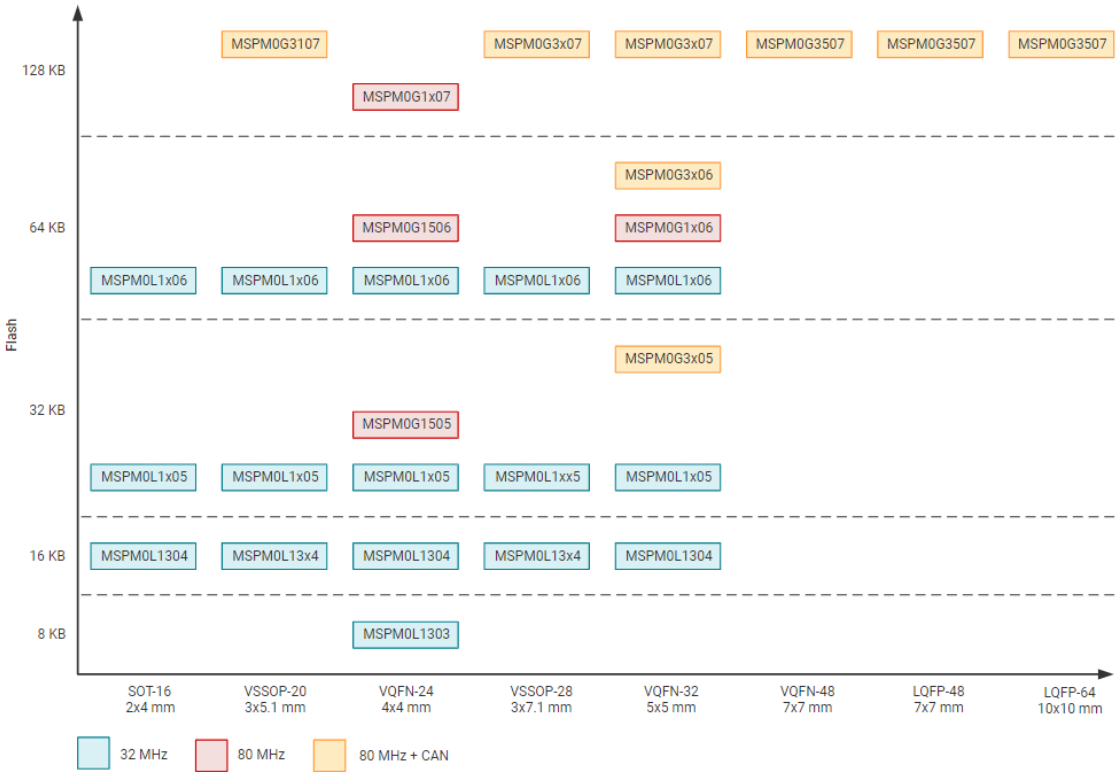


图 2-12. MSPM0L 和 MSPM0G 产品系列

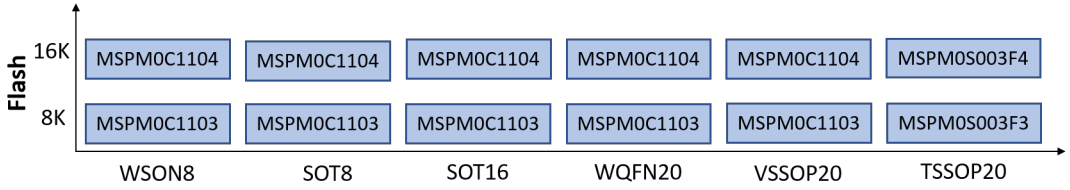


图 2-13. MSPM0C 产品系列

要将范围缩小至特定的器件，产品选择工具可发挥重要作用。在此链接中，您可以根据您的 MCU 外设要求，使用左侧的筛选器进行初步筛选。例如，要筛选出符合 UART 编号的 MCU，可以直接使用筛选工具进行选择，右侧将弹出符合条件的 MCU 器件，如图 2-14 所示。您可以通过左侧的搜索文本框直接转到器件页面，以获取相应器件的详细信息。

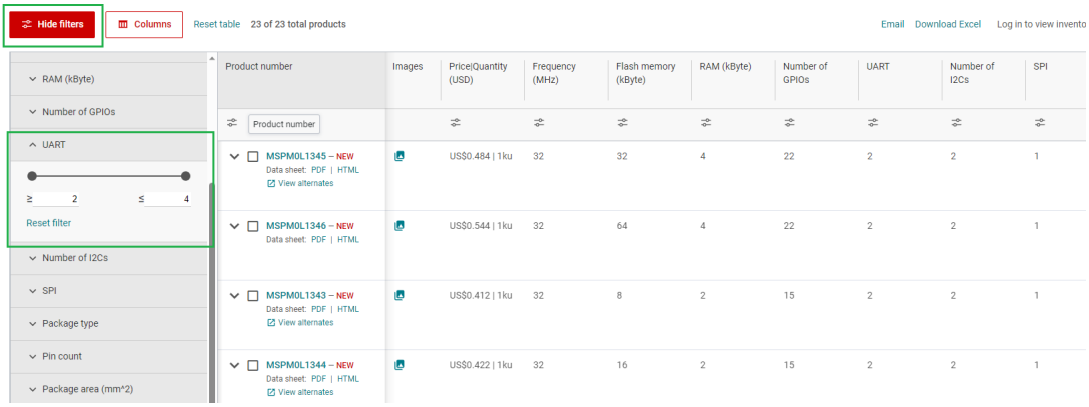


图 2-14. MSPM0 产品选择工具

在器件页面上，可以轻松找到并下载数据表、技术参考手册 (TRM) 和勘误表等重要文档，如图 2-15 所示。器件专用数据表介绍了专用 MSPM0 的参数和功能数据信息。器件专用 TRM 介绍了 MSPM0 系列的应用方法和特性。器件专用勘误表介绍了 MSPM0 相关系列或版本的勘误说明。

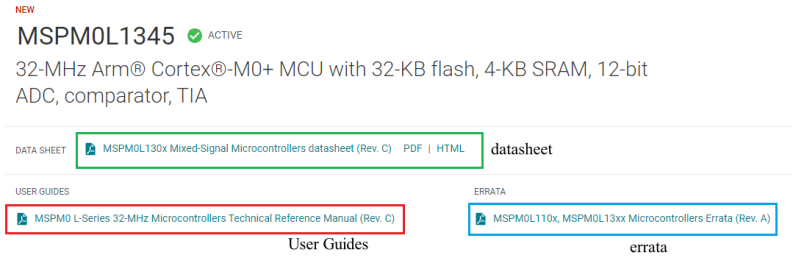


图 2-15. MSPM0 重要文档列表

该网站还列出了有关 MSPM0 的相关技术文档，其中最常见的是应用手册。

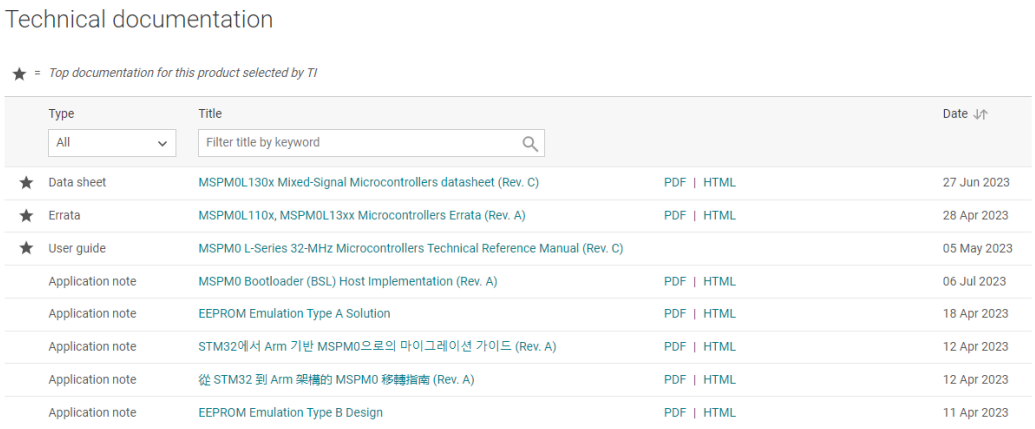


图 2-16. MSPM0 相关技术文档列表

完成选择后，您可以通过“订购和质量”检查价格和其他信息，如图 2-17 所示。

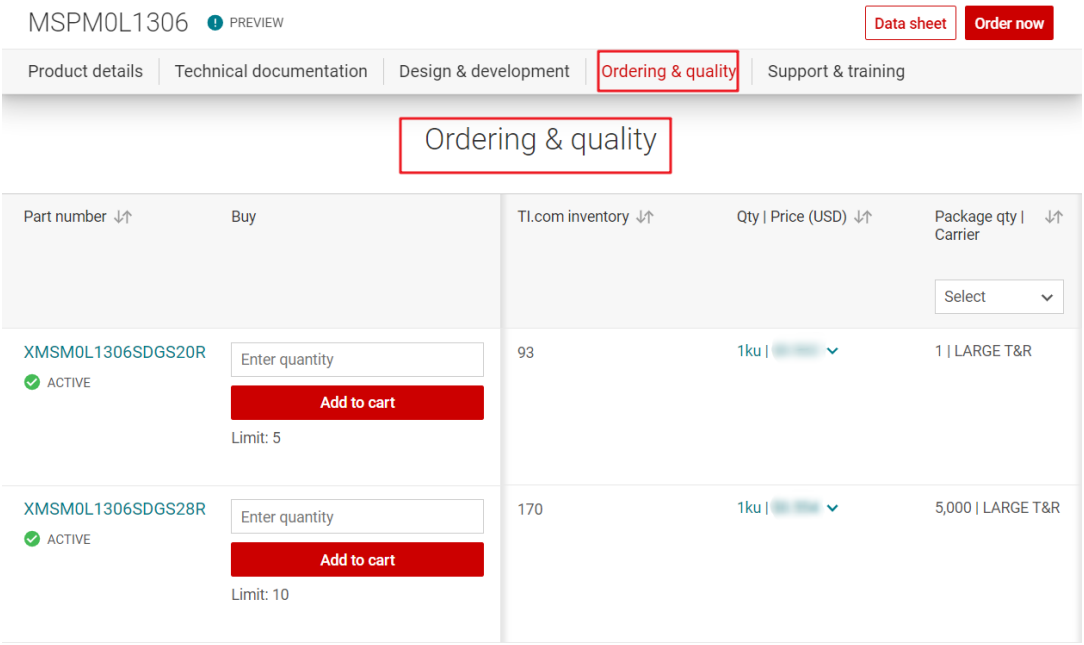


图 2-17. 订购和质量器件视图

2.2.2 步骤 2.设置 IDE 和 CCS 简介

2.2.2.1 设置 IDE

所选的 IDE 为 TI CCS。

1. 点击下载链接并开始安装，一直按下一步。

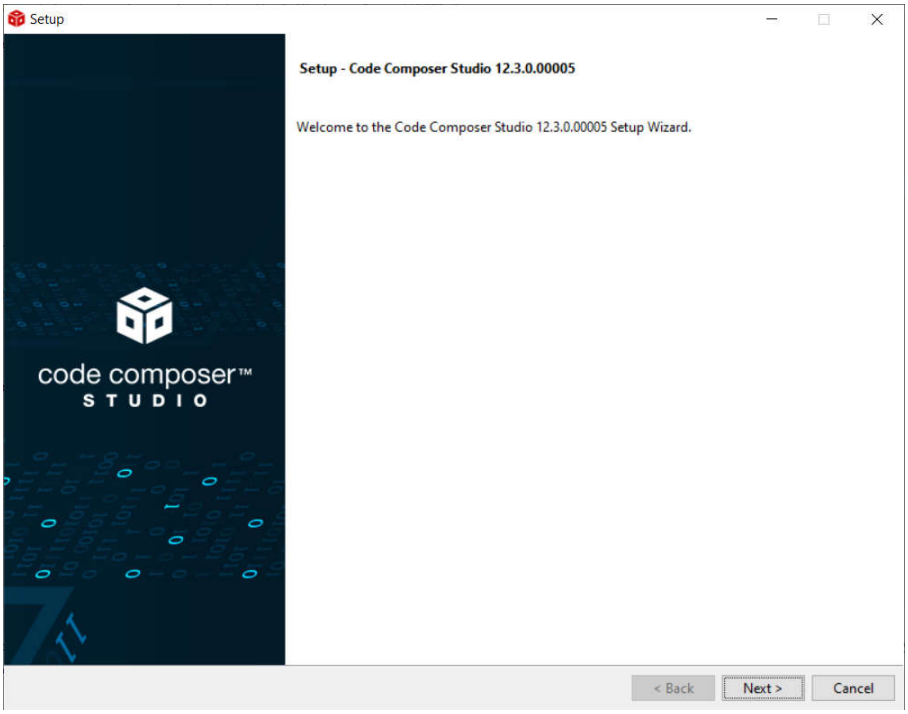


图 2-18. CCS 安装

2. 选择 MSPM0 支持元件。

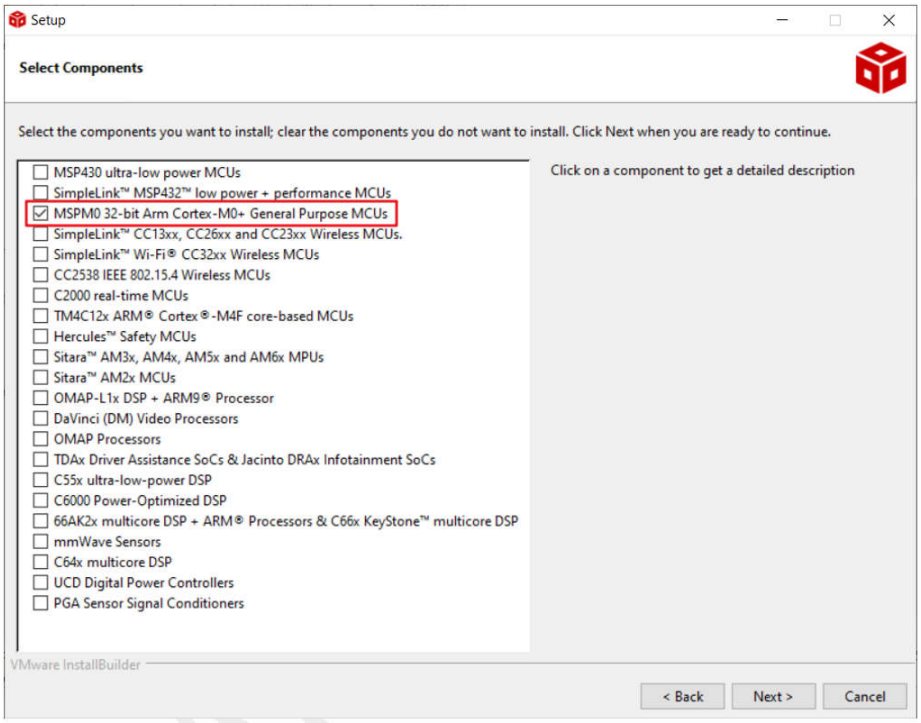


图 2-19. 安装 CCS - 选择 MSPM0 支持元件

3. 如果需要，选择 J-link。

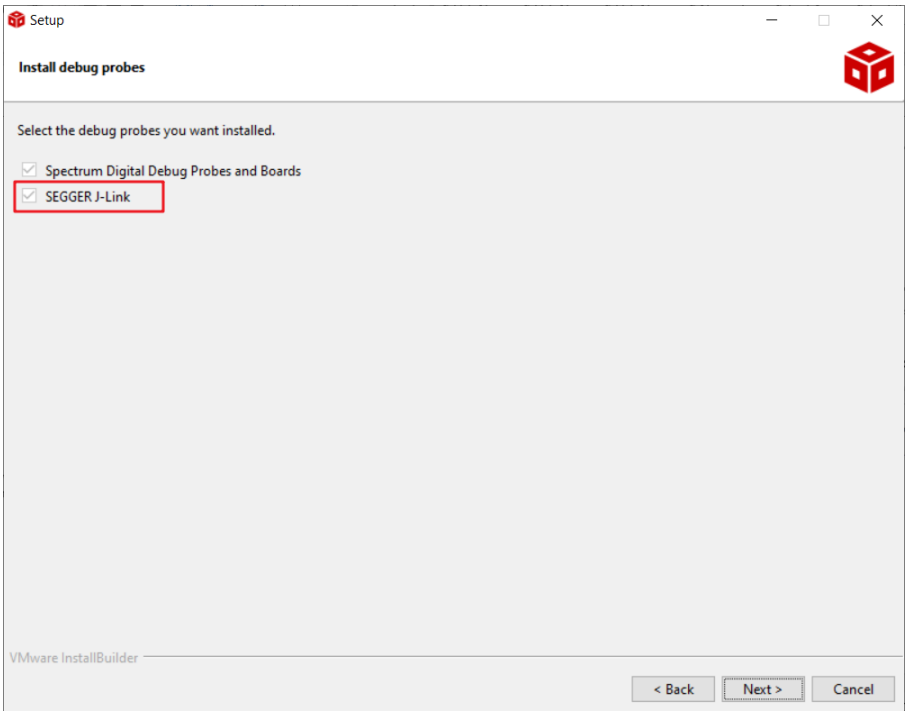


图 2-20. 安装 CCS - 选择下载 J-Link

4. 完成 CCS 下载。

2.2.2.2 CCS 简介

1. 启动一个新的工作区。工作区是指复制导入的工程的地址。CCS 中的步骤与 e²studio 中的步骤相同。

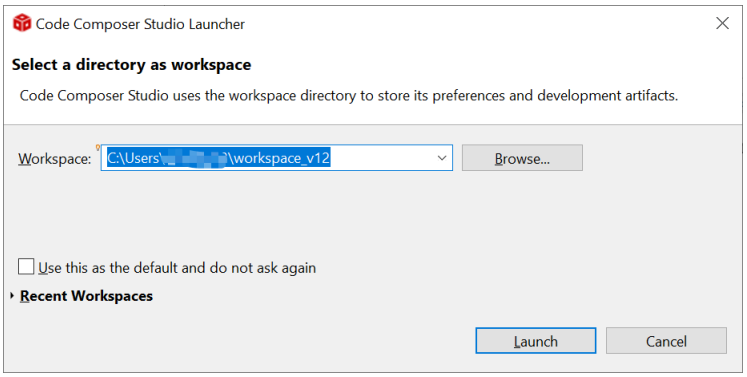


图 2-21. CCS 启动工作区

2. 如果要创建新工程，请转到“File”>“New”>“CCS Project”。有两项重要的设置需要完成。首先是选择 MSPM0 器件，其次是选择连接，如图 2-22 所示。然后，添加工程名称并按完成按钮后即可创建程序。

   此过程与在 e²studio 中创建新工程类似，后者需要选择器件、工具链和程序名称。

   建议查找 MSPM0 SDK 示例，该示例介绍了如何使用 CCS，请参阅节 2.2.4。

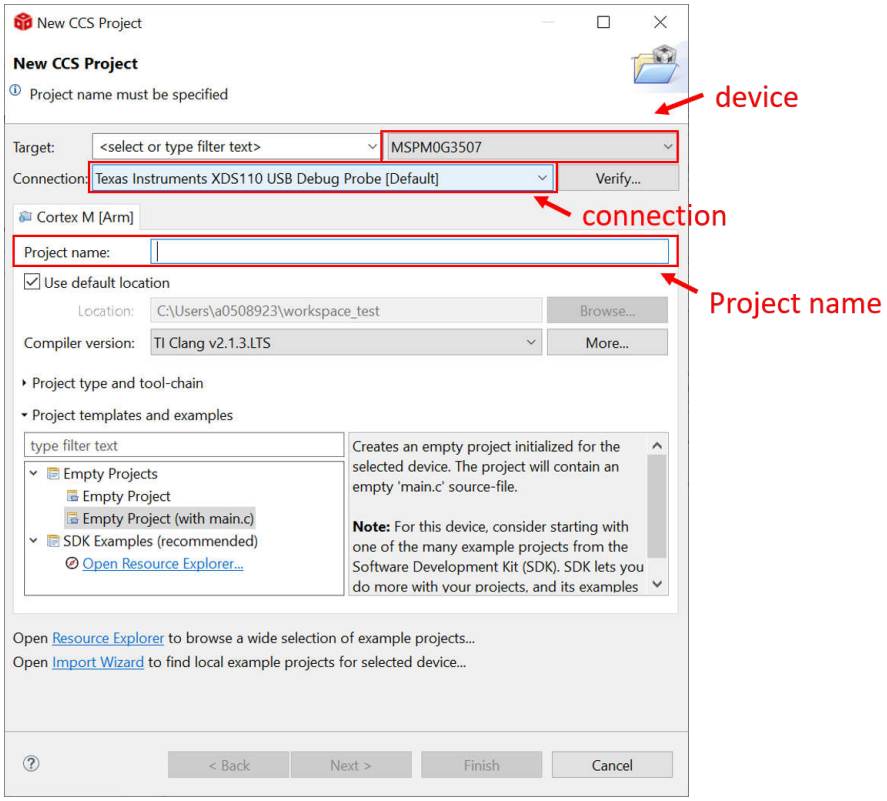


图 2-22. 在 CCS 中创建新工程

3. 图 2-23 简要介绍了 CCS 函数。

快捷键功能：

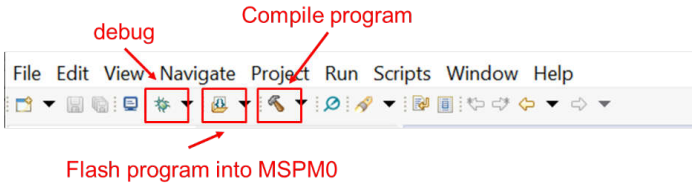


图 2-23. 常用功能

调试功能：

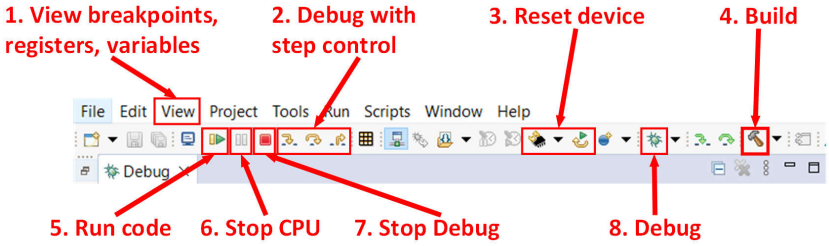


图 2-24. 常用调试功能

工程属性常用设置：

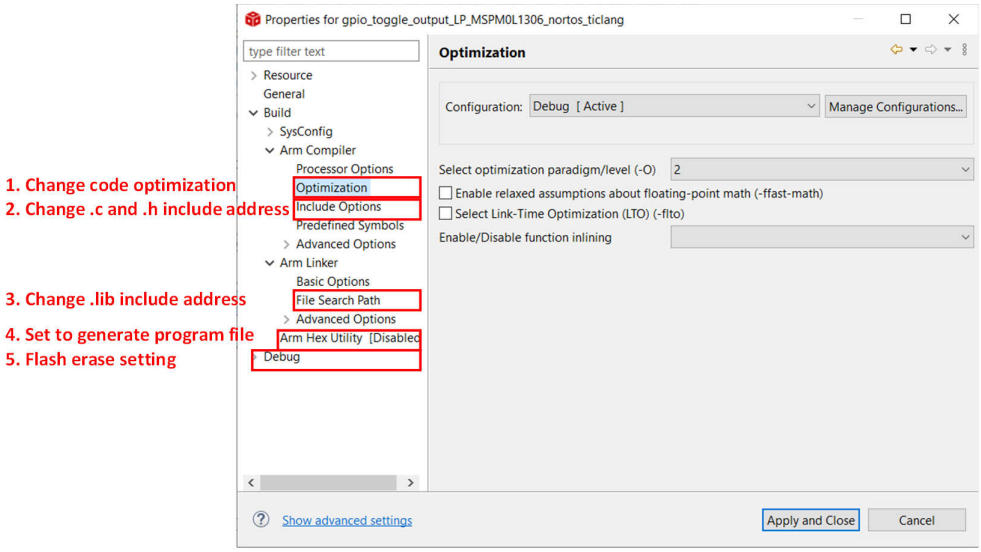


图 2-25. 常用工程设置

有关详细信息，请参阅：适用于 MSPM0 MCU 的 Code Composer Studio IDE 版本 12.4+ — 适用于 MSPM0 MCU 的 Code Composer Studio IDE 1.0 文档

2.2.3 步骤 3.设置 MSPM0 SDK 和 MSPM0 SDK 简介

对于 RL78，没有软件包，用户只能从 IDE 的“Developer Assistance”中搜索示例代码或从网站下载特定的示例代码。但 TI 具有配套的软件开发套件，有助于简化开发。

2.2.3.1 设置 MSPM0 SDK

1. 点击链接下载 MSPM0 SDK。

图 2-26. 下载 MSPM0 SDK

2. 按下一步安装 SDK。

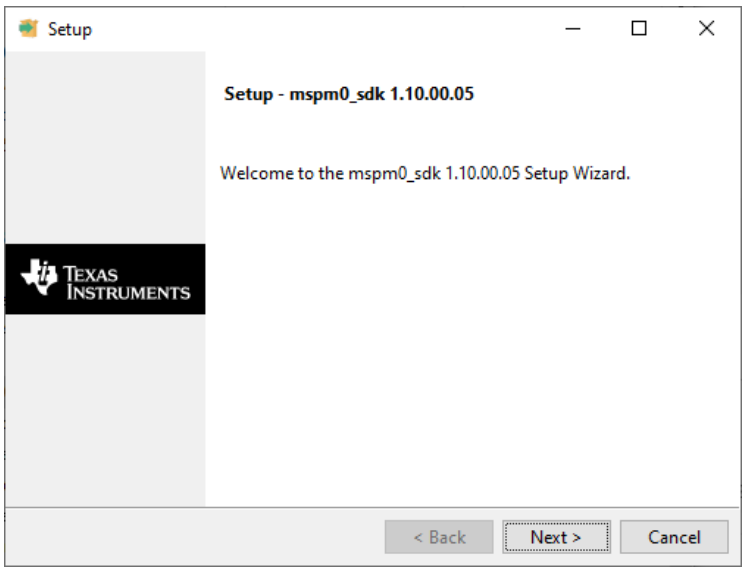


图 2-27. 安装 MSPM0 SDK

3. 完成 MSPM0 SDK 下载。

2.2.3.2 SDK 简介

完成下载后，文件内容将显示在 SDK 文件夹中，此文件夹默认为“c:/ti/mspm0_sdk_xxx”，如图 2-28 所示，其中最常用的文件夹是 examples 文件夹和 docs 文件夹。

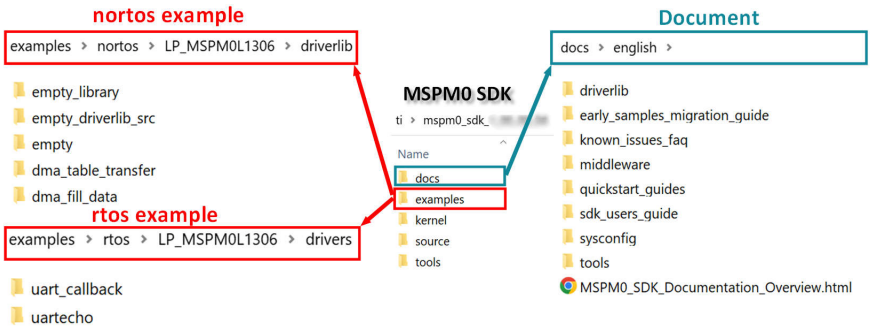


图 2-28. MSPM0 SDK 文件夹

表 2-6 汇总了示例覆盖。

表 2-6. MSPM0 示例覆盖范围

受 SDK 支持	平台			
IDE	CCS		Keil	IAR
编译器	TI Arm-Clang	GUN Arm	Arm/Keil 编译器	IAR Arm 编译器
RTOS	FreeRTOS			
代码示例	Driverlib/TI 驱动程序（驱动程序）			

在 nortos 示例中，还可以找到三个空的工程供用户构建自己的工程。表 2-7 显示了差异。

表 2-7. 空工程描述

示例	使用 Sysconfig	将库文件包含到工程中
empty	是	否
empty_library	否	有
empty_driverlib_src（建议）	是	是

对于 docs 文件夹，结构和重要文档如图 2-29 所示。

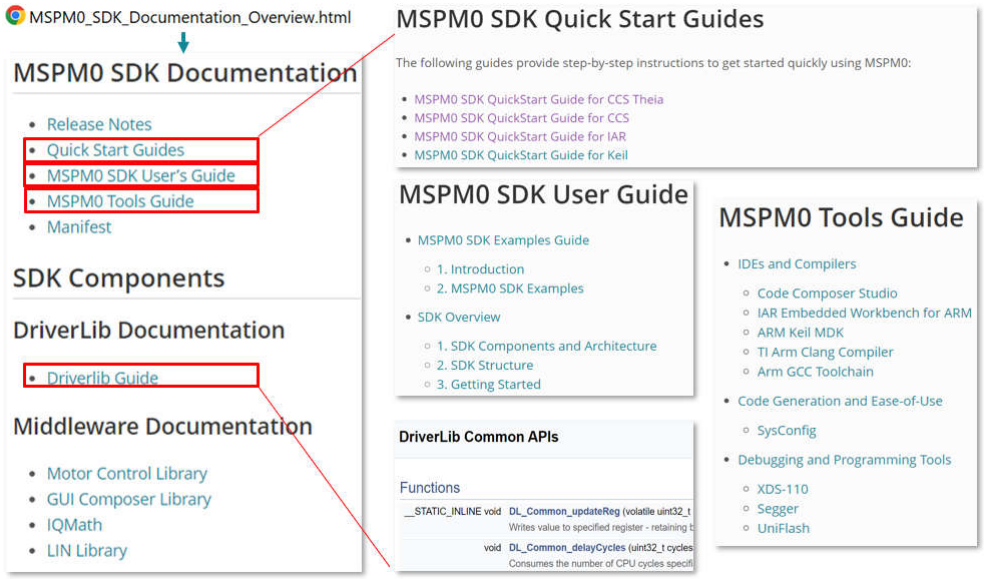


图 2-29. 文档概览

2.2.4 步骤 4.软件评估

以下是将示例移植到 CCS 的一些简单步骤。

1. 选择“Project”，然后从菜单中选择“Import CCS Projects”以导入 CCS 工程。

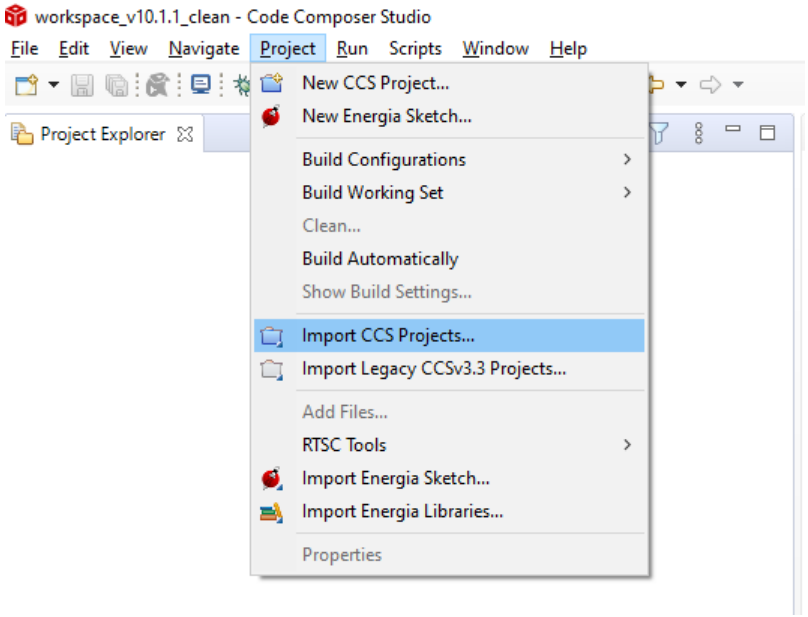


图 2-30. 导入 CCS 工程

2. 从 SDK 中选择程序。以 MSPM0L1306 为例。

   \mspm0_sdk_1_10_00_05\examples\nortos\LP_MSPM0L1306\driverlib

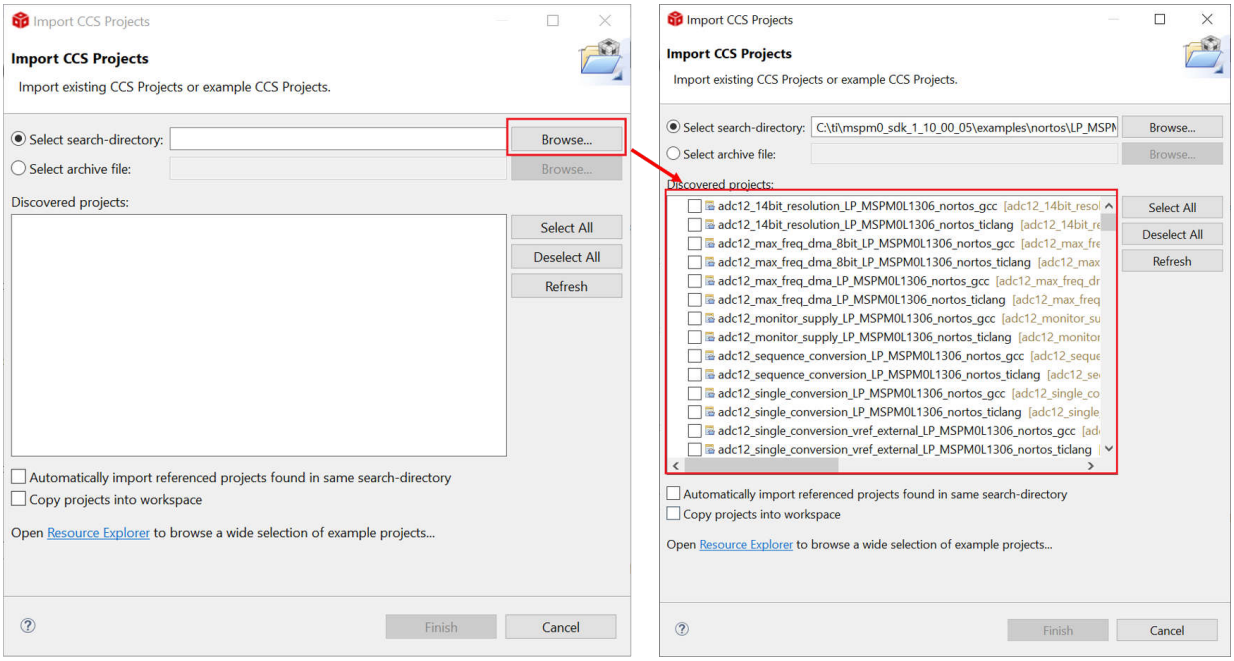


图 2-31. 从 SDK 中选择程序

如果无法导入文件，请删除工作区下的同名工程。

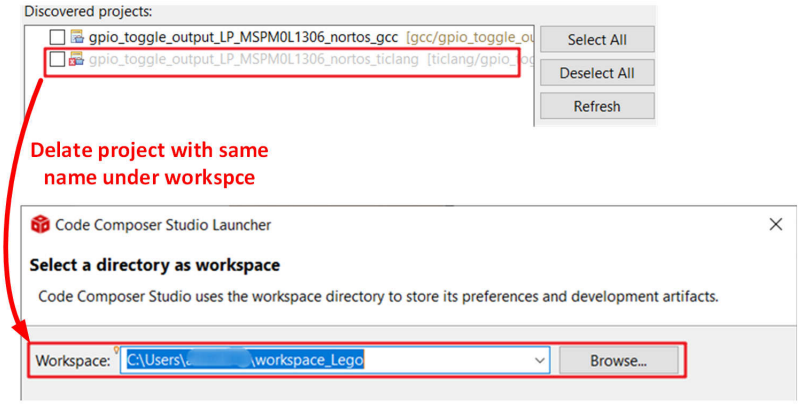


图 2-32. 删除重复的工程

3. 导入后，左侧将出现一个工程，并自动打开一个 REAME.md 文件。建议您首先阅读 README.md 文件，其中包含此示例的用途以及硬件配置。

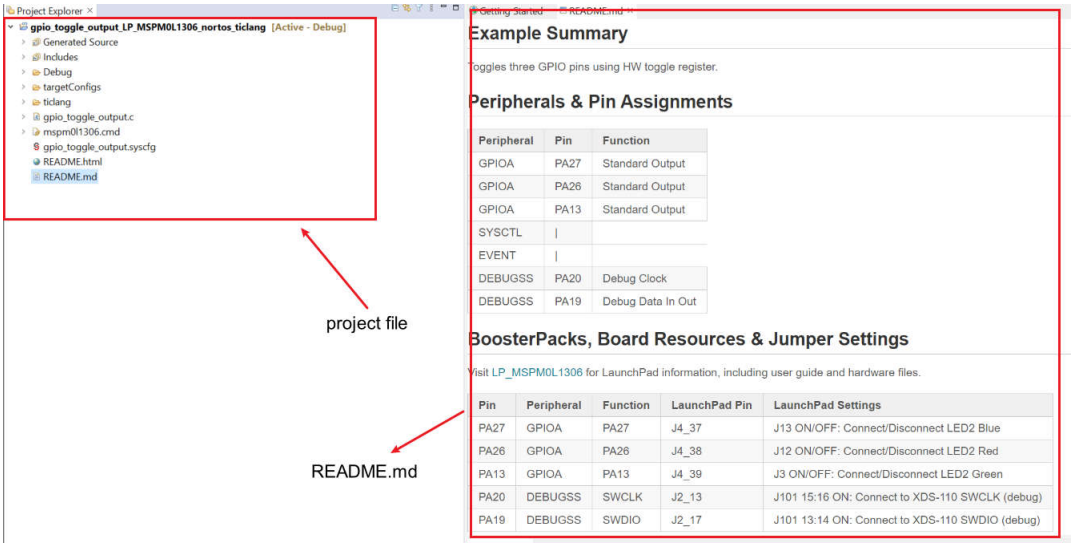


图 2-33. 工程和 README.md

4. 图 2-34 显示了工程中最重要的文件。

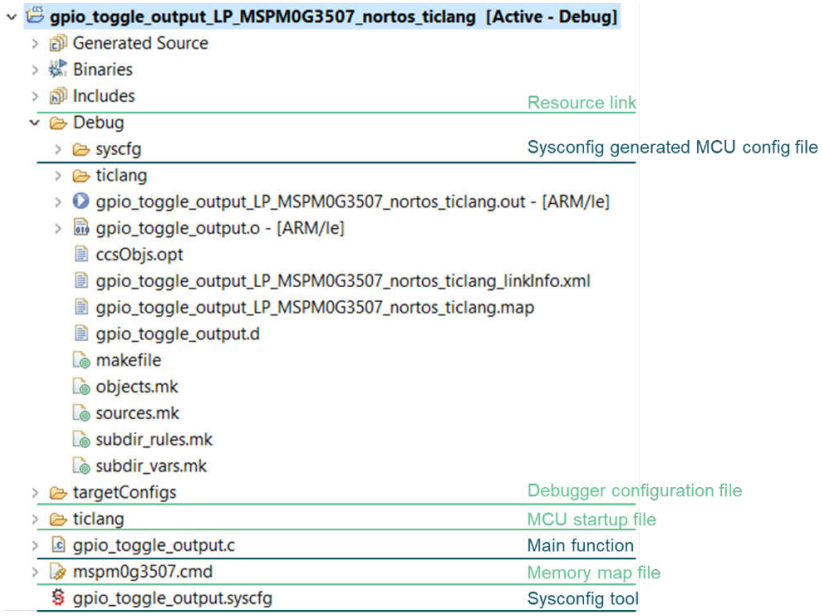


图 2-34. CCS 工程概述

5. 就像在 RL78 开发中一样，双击 .scfg 文件到达“Smart Configuration”界面，双击 .syscfg 文件可以到达 SysConfig，在这里可以通过图形界面配置所需的外设。此外，建议先使用 SysConfig 的 MCU 视图来帮助您与软件工程师一起修复引脚功能，这类似于 e²studio 中的 MCU/MPU Package。

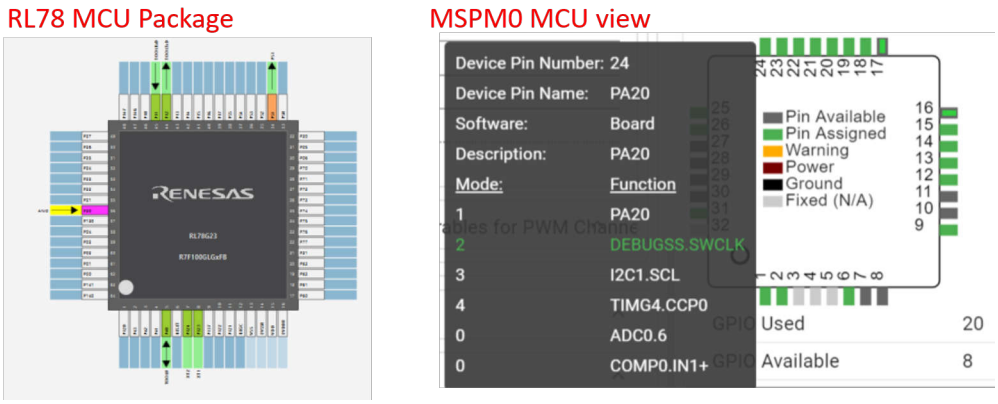


图 2-35. Smart Configuration 和 SysconfigSysConfig 中的 MCU 视图

6. 根据代码和 SysConfig 示例，您可以根据 TI.com 上发布的器件特定 TRM 或应用手册来完善工程或对其进行修改。
7. 如果要添加第三方库，可按照以下步骤操作。首先，必须将相关文件添加到工程中，如图 2-36 所示。

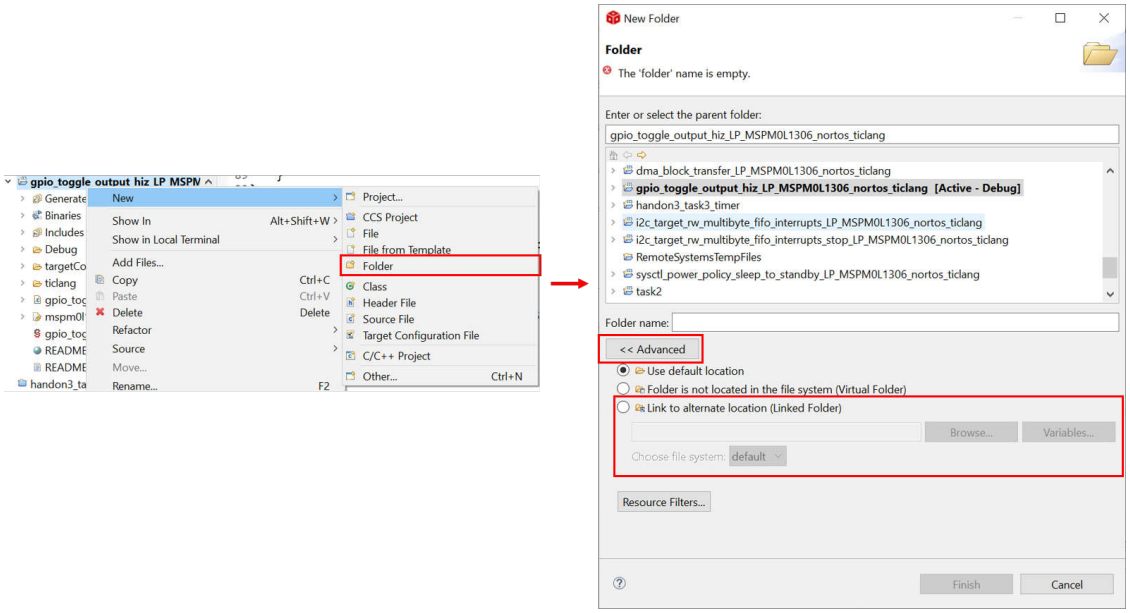


图 2-36. 添加相关文件

然后，需要执行其他步骤以告诉编译器您已添加头文件。

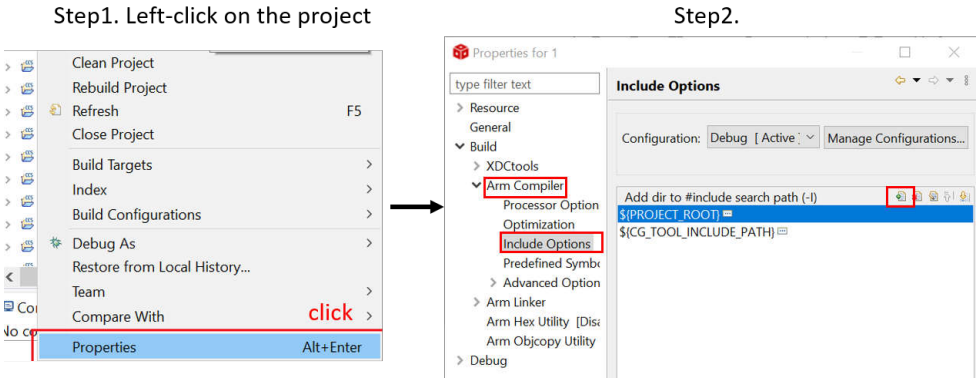


图 2-37. 设置“Include Options”

8. 完成软件评估后，请点击主工具栏中的“Build”图标，如图 2-38 所示。出现“Build Finished”字样即表示已成功编译。

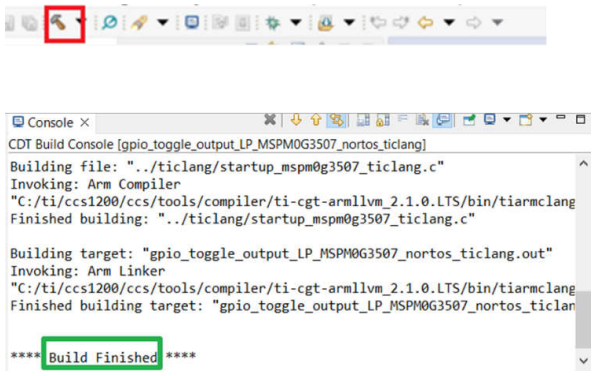


图 2-38. 成功构建

2.2.5 步骤 5.PCB 板设计

1. 要获取设计包，请访问 Ti.com 并进入特定器件页面。以 MSPM0L1304 为例。
2. 点击“Design & development”>“CAD/CAE symbols”，根据需要选择要下载的不同封装模型。

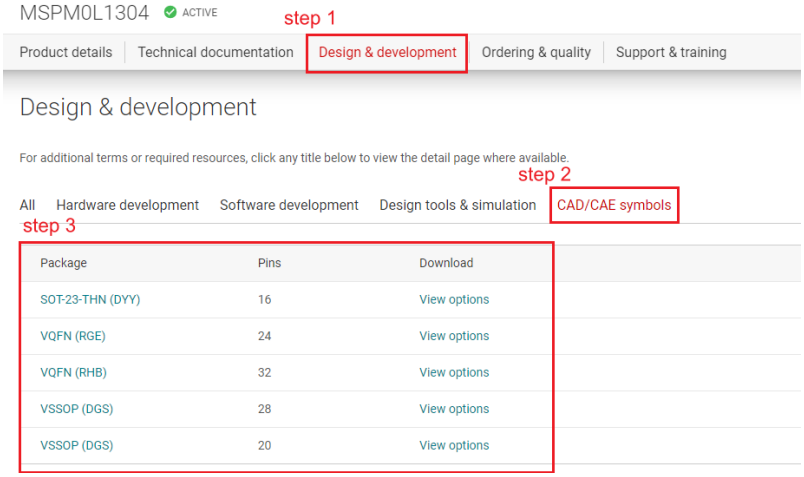


图 2-39. Ultra Librarian 工具入口

3. 对于 MSPM0，可将硬件设计到您自己的电路板中。图 2-40 所示为最小系统设计示例。

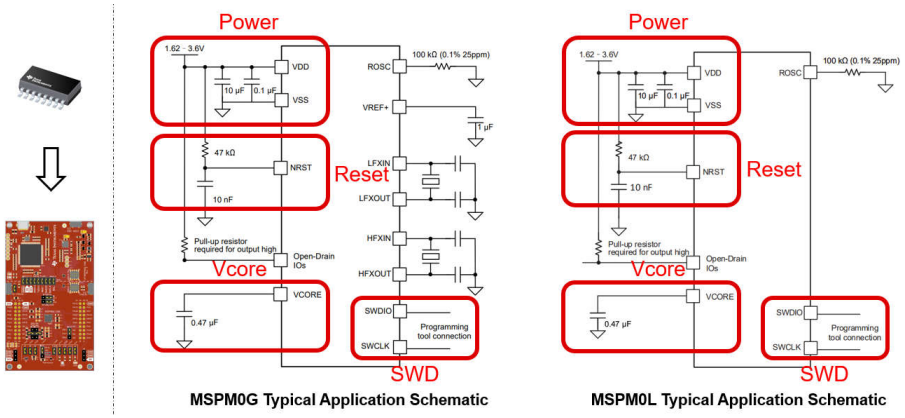


图 2-40. MSPM0 最小系统

对于最小系统，需要注意以下几点：

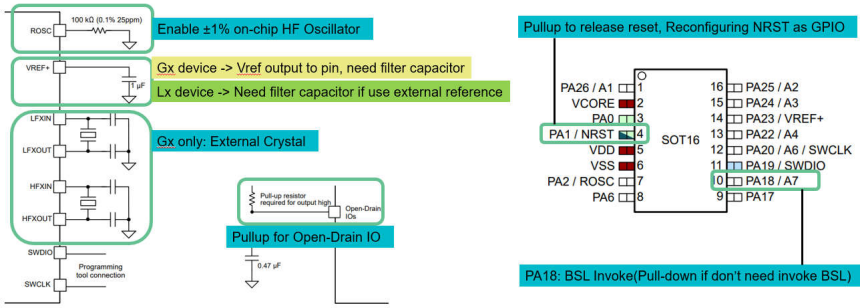


图 2-41. MSPM0 最小系统注意事项

更多有关硬件开发的详细信息，请参阅以下资料：

- MSPM0 G 系列 MCU 硬件开发指南
- MSPM0 L 系列 MCU 硬件开发指南

2.2.6 步骤 6.大规模生产

1. 通过 CCS 生成生产文件 (.bin/.txt/⋯)。

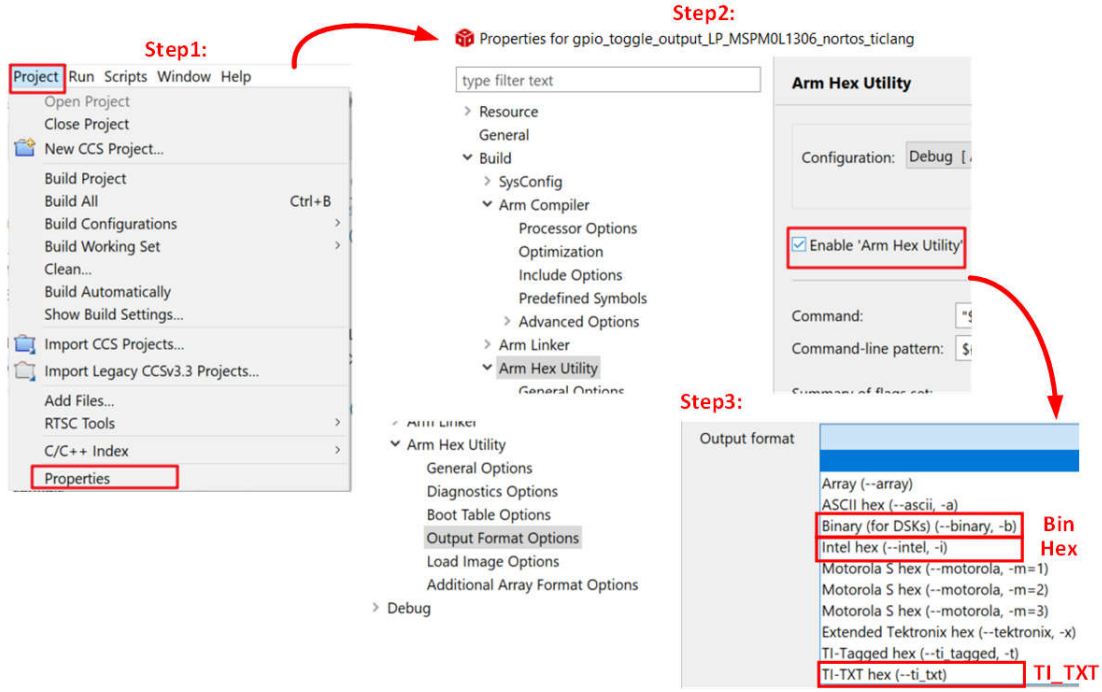


图 2-42. 创建程序文件

2. 选择编程器/调试器来对 MSP 器件进行编程。

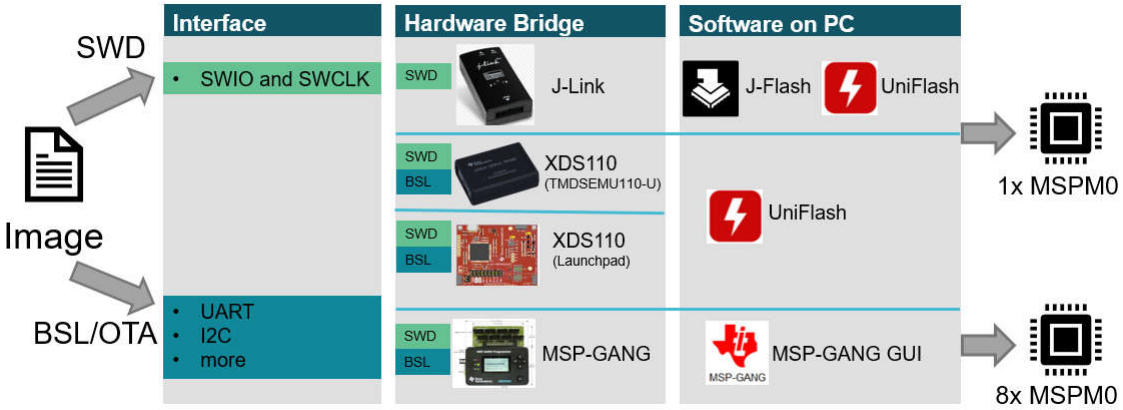


图 2-43. 程序软件和工具

如需了解如何使用 MSP-GANG 和 J-LINK，请参阅：MSPM0 设计流程指南。

更多有关调试的信息，请参阅：调试和编程工具指南。

2.3 示例

MSPM0 设计流程如下所示。本示例旨在使用 PWM 来驱动 LED。

1. 选择合适的 MSPM0 MCU、选择硬件并订购 EVM，我们在这里使用的是 LaunchPad MSPM0L1306。
2. 设置 CCS 和 SDK，详情如节 2.2 所述。

3. 代码导入。

   当环境准备就绪后，即可将代码导入 CCS。本示例中使用计时器来控制 PWM。首先要了解 RL78 和 MSPM0 之间的计时器模块差异，并在 SDK 中选择类似的示例。

   SDK 中最接近的示例可能是“timx_timer_mode_pwm_center_stop”。找到类似的示例后，打开 CCS 并导入代码示例，方法是转到“Project”>“Import CCS Projects...”并导航至 MSPM0 SDK 的示例文件夹。

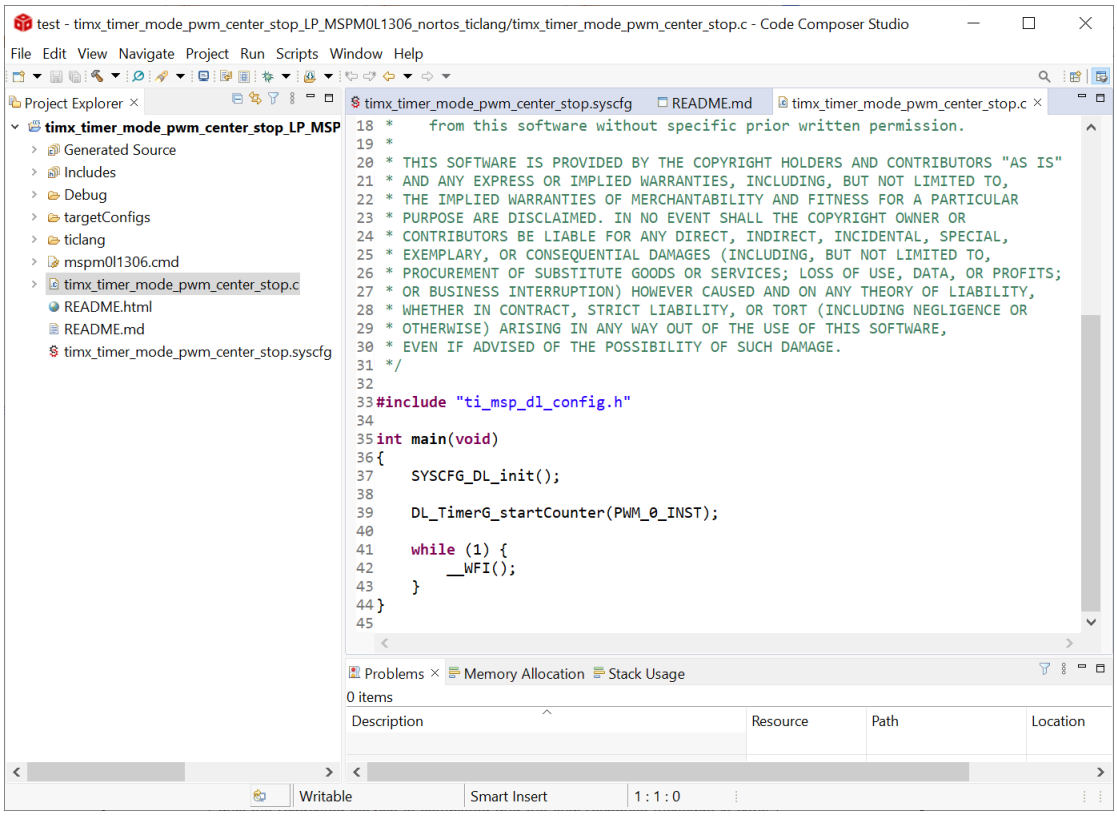


图 2-44. 代码示例文件

4. 修改工程。

要查看 SysConfig 配置，请打开 .syscfg 文件。选择“TIMER-PWM”部分以生成 PWM，如图 2-45 所示。检查 PWM 的时钟配置，如自发频率和占空比。在本例中，PWM 频率为 2.7Hz，占空比为 75%。您可以通过在“Desired Duty Cycle”中输入 50% 来轻松更改占空比，然后“Counter Compare Value”会自动更改。

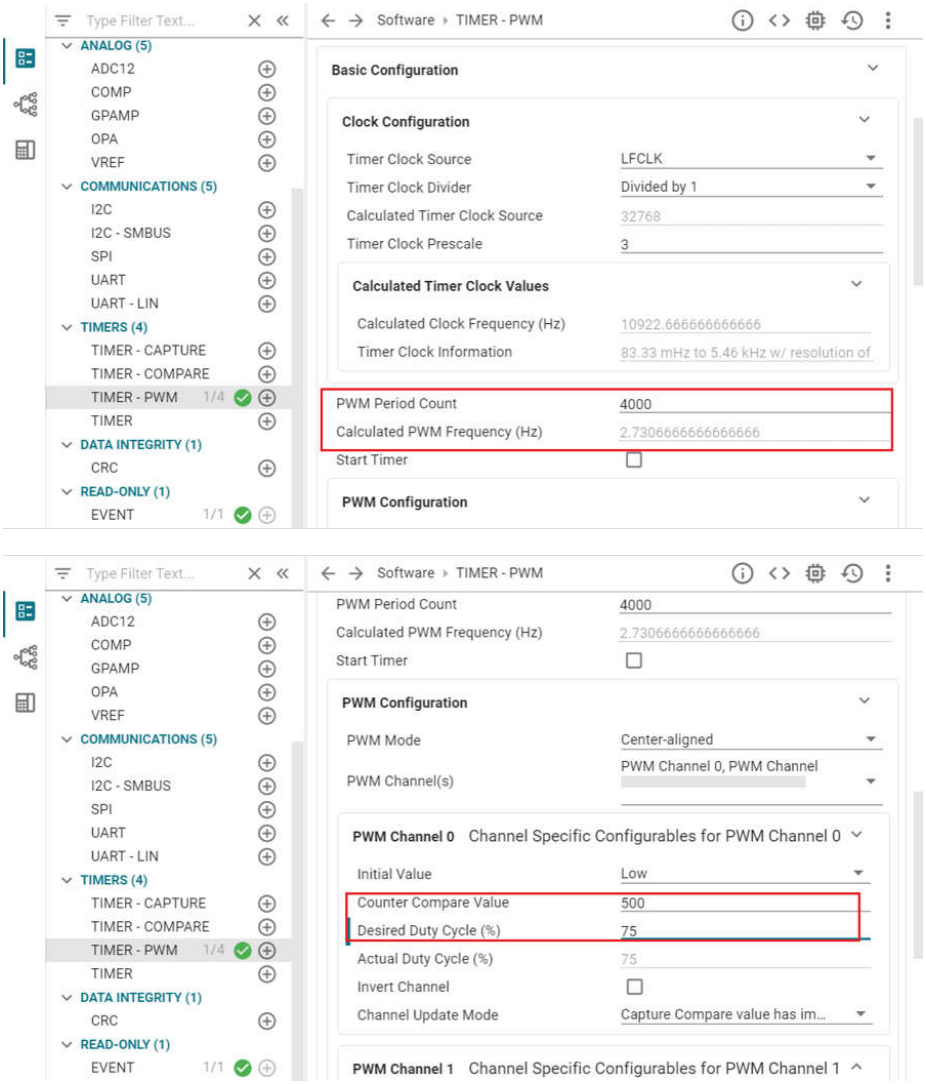


图 2-45. SysConfig 中的 PWM 配置

要获取每个功能模块的详细说明，您可以点击每一项旁边的“?”符号。

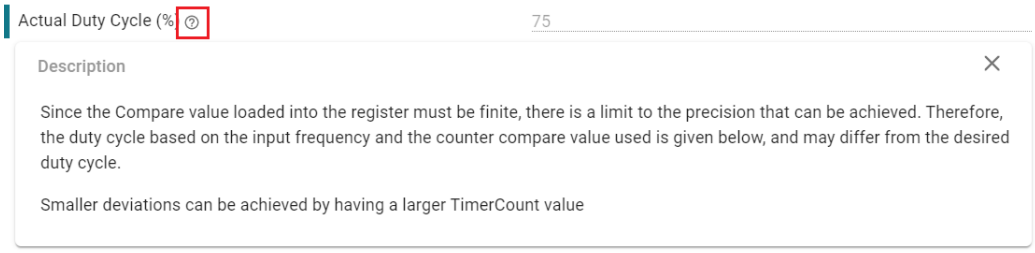


图 2-46. 获取每一项的详细信息

此外，检查 TIMER-POWER 模块的其余功能，点击右上角的芯片图标并检查 PWM 的突出显示引脚来检查正在使用的引脚。

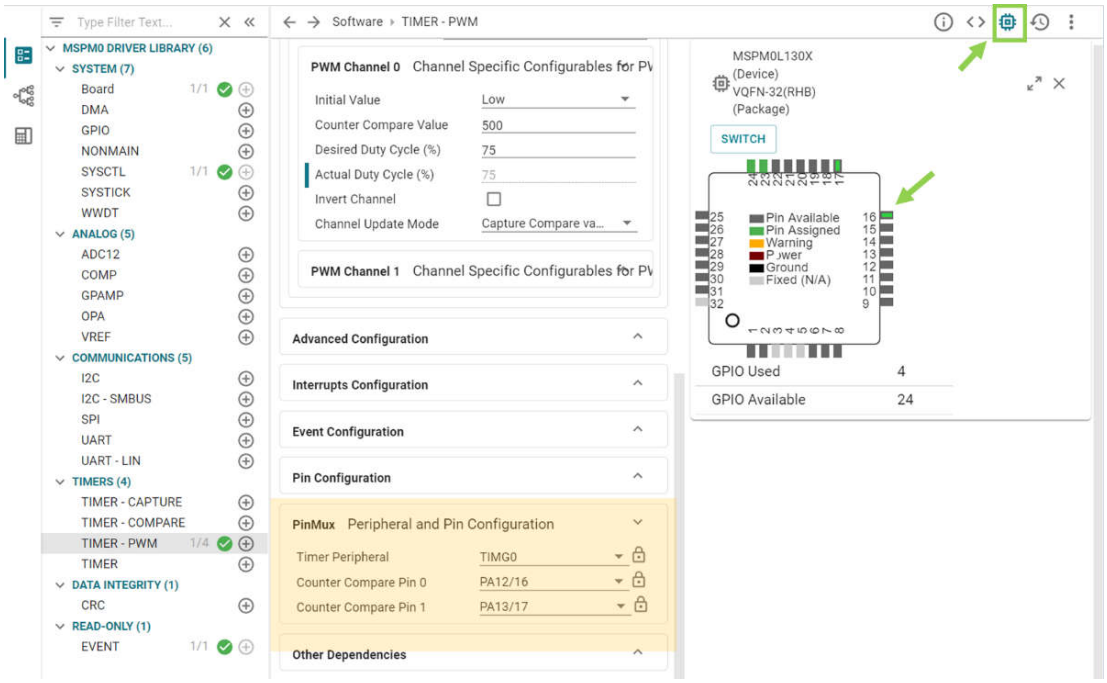


图 2-47. 引脚配置

保存并重建工程后，SysConfig 会更新图 2-48 所示的文件。此时，已修改示例硬件配置，以匹配正在移植的原始软件的全部功能。

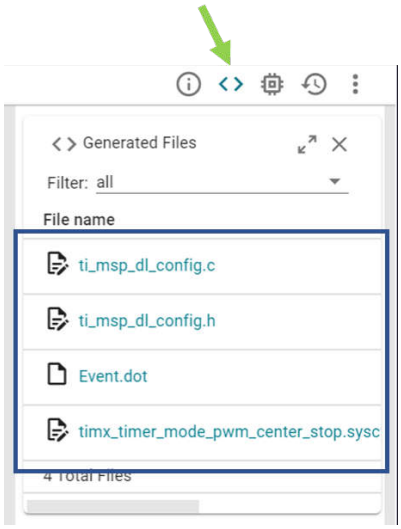


图 2-48. SysConfig 更新的文件

唯一剩下的工作是检查应用程序级软件。本示例像 SDK 代码一样生成 PWM 波，因此无需更改 .c 文件。

5. 硬件设置。

   将 LaunchPad 插入计算机。根据引脚配置，使用 DuPont 电缆将 PA12 连接到 LED 引脚。

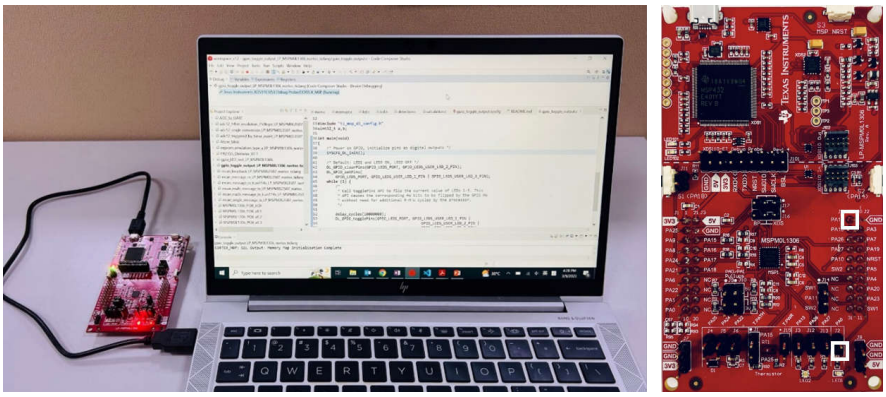


图 2-49. 硬件设置

6. 调试和验证。

   点击调试图标开始调试。您可以通过双击行号前面的空格或添加一行代码 *__BKPT();* 来设置断点。

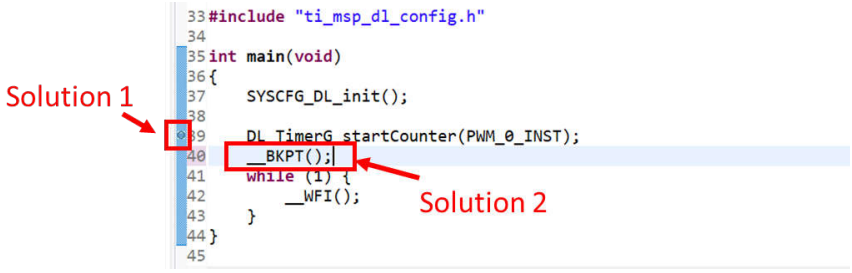


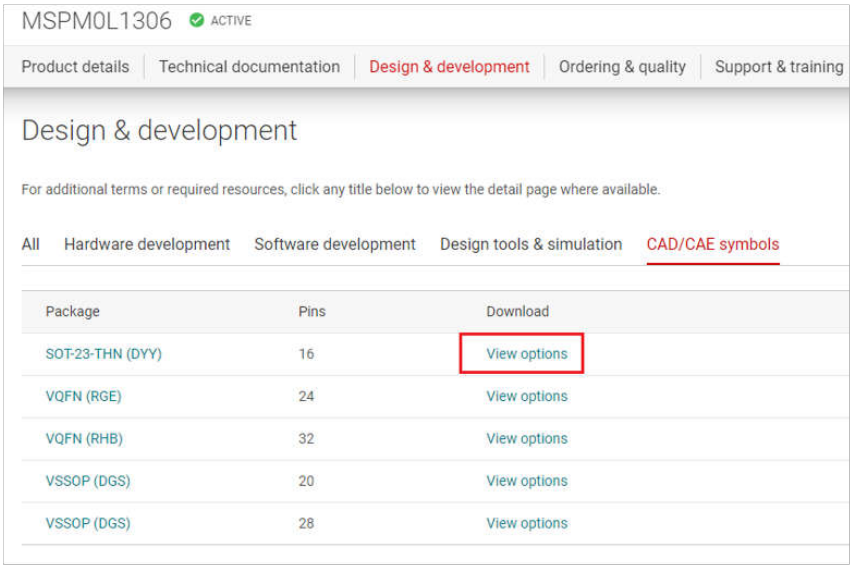
图 2-50. 添加断点解决方案

   尝试使用调试功能（详见节 2.2.2.2）并验证该过程的可行性。在调试时，可随着代码逐步运行而切换 LED。

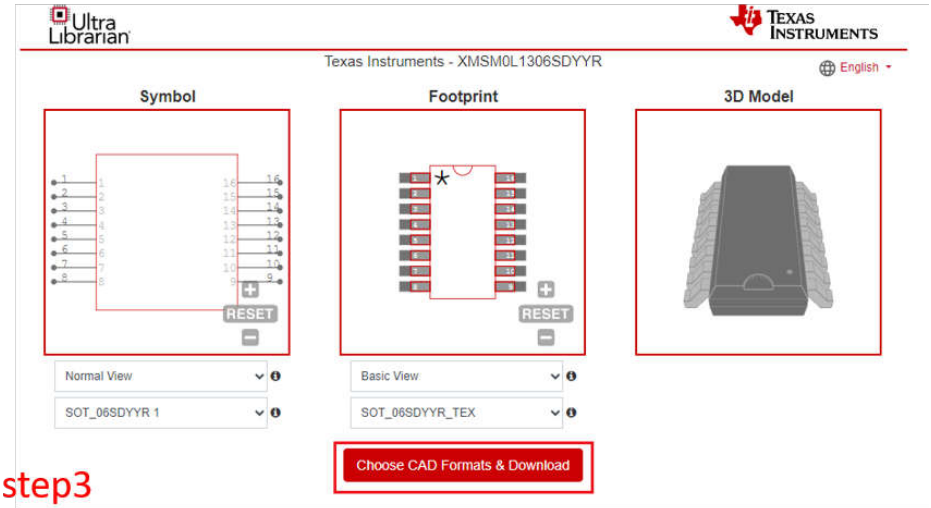
7. 生成 PCB 库并导入到 Altium Design。

   具体步骤如图 2-51 所示。转到 MSPM0 器件页面下 Ultra Librarian 工具的入口（详见节 2.2.5）。点击 *View options*。选择所需的 CAD 格式和引脚排序，然后您可以获得 Altium 设计 lib 文件。

step1



step2



step3

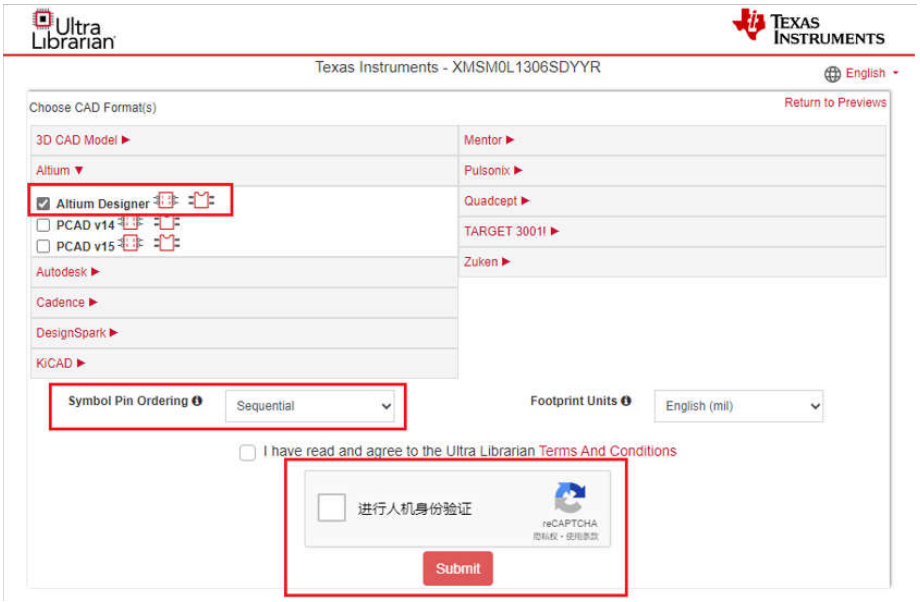


图 2-51. Ultra Librarian 工具下载

库下载完毕后，下一步是运行 Altium Designer 脚本并生成 PCB 库和原理图库，如图 2-52 所示。

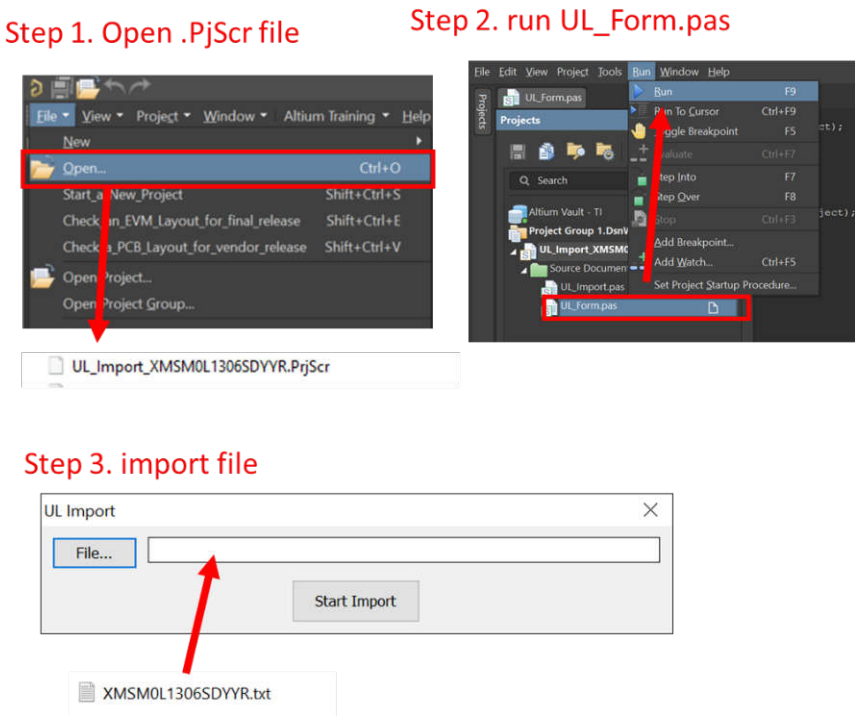


图 2-52. 运行 Altium Designer 脚本

完成这些步骤后，系统将在同一源文件夹中生成以下新文件。

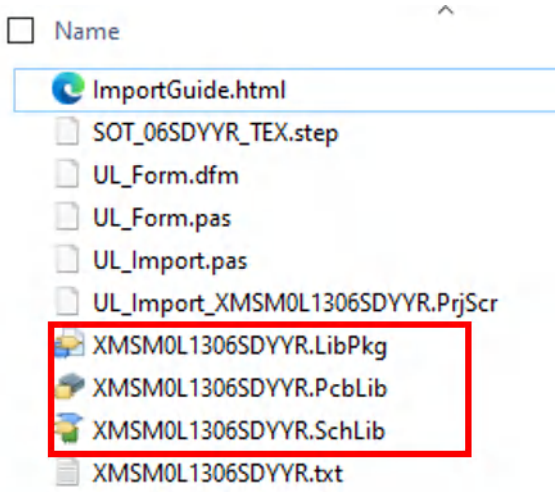


图 2-53. PCB 库和原理图文件

最后一步是将它们导入到您的 AD 库中，如图 2-54 所示。在此基础上，可以设计原理图和 PCB。

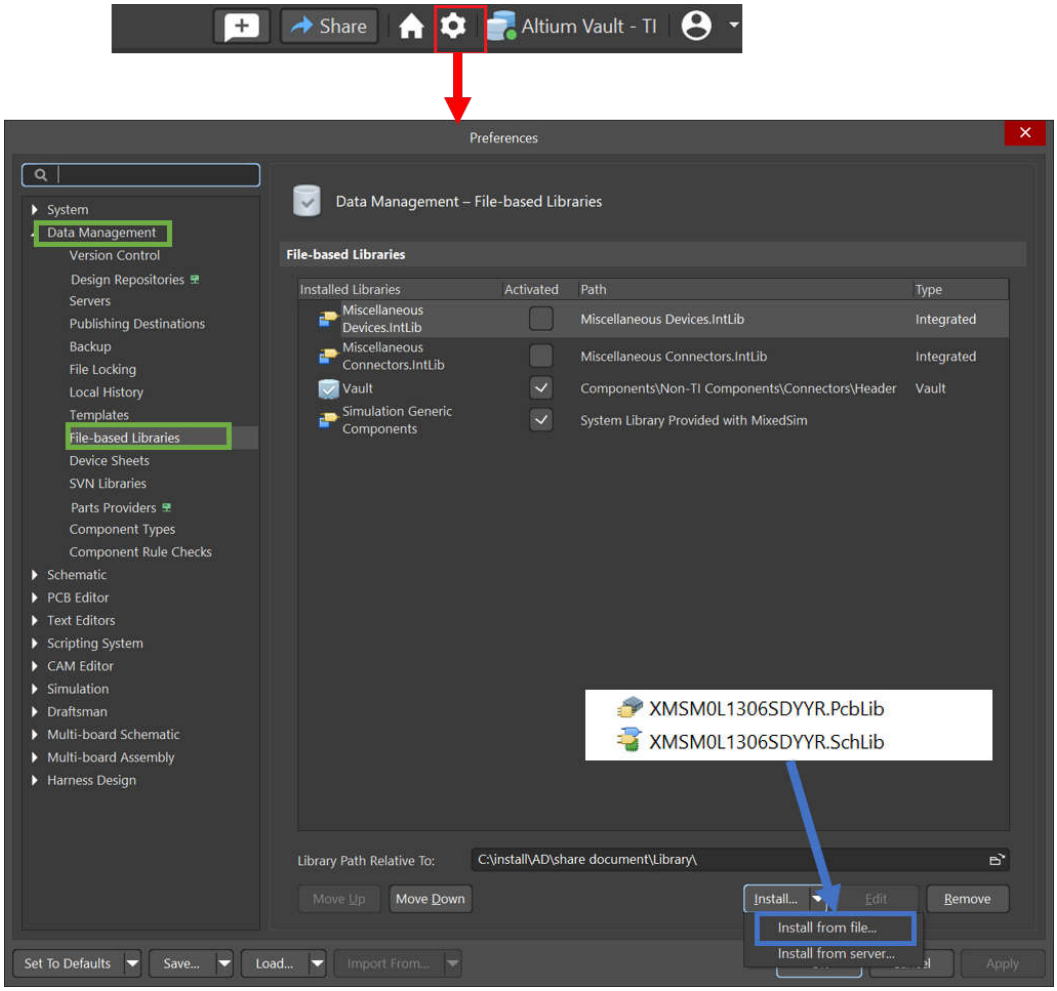


图 2-54. 导入库

8. 在 MSPM0 中设计。
9. 大规模生产。

3 内核架构比较

3.1 CPU

MSPM0 系列基于 ARM Cortex M0+ CPU 内核架构。RL78 系列基于其自身的 RL78 CPU 内核架构。表 3-1 概要展示了 MSPM0 系列 CPU 与 RL78 的一般功能比较。

表 3-1. CPU 功能集比较

特性	RL78	MSPM0G	MSPM0L	MSPM0C
架构	专用 RL78 内核	Arm Cortex M0+	ARM Cortex M0+	Arm Cortex M0+
指令集	CISC	RISC	RISC	RISC
流水线	3 级	2 级	2 级	2 级
工作频率（最大值）	40MHz	80MHz	32MHz	24MHz
DMA	是	是	是	是
Coremark/MHz	1.5952 (1)	2.39 (2)	2.39 (2)	2.39 (2)

(1) RL78G23 的分数是在 32MHz 工作频率和 Renesas CC-RL V1.12 编译器下获得的。
(2) 该分数由 ARM 官方网站上的“Arm Cortex-m0+ 处理器数据表”提供。

3.2 嵌入式存储器比较

3.2.1 闪存功能

MSPM0 和 RL78 系列 MCU 具有非易失性闪存，用于存储可执行程序代码和应用程序数据。表 3-2 显示了闪存特性的比较。

表 3-2. 闪存功能比较

特性	RL78		MSPM0
闪存	程序闪存	RL78Gxx 范围 1KB 至 768KB RL78Lxx 范围 8KB 至 256KB RL78Ixx、RL78Hxx 范围 8KB 至 512KB RL78Fxx 范围 8KB 至 512KB	MSPM0Gxx 范围 32KB 至 128KB MSPM0Lxx 范围 8KB 至 64KB MSPM0Cxx 8KB 或 16KB
	数据闪存	RL78Gxx 范围 0KB 至 8KB RL78Lxx 范围 2KB 至 8KB RL78Ixx、RL78Hxx 范围 0KB 至 4KB RL78Fxx 范围 4KB 至 16KB	
单个闪存大小	程序闪存	32 位	64 位
	数据闪存	32 位或 8 位	
存储器组织	块大小（512B 或 1KB） 存储体大小（可变） 大多数器件 - 2 个存储体 I1C 器件 (512KB) - 3 个存储体(1)		扇区大小 (1KB) 存储体大小（可变） 不超过 256KB 的器件 - 1 个存储体 超过 256KB 的器件 - 2 个存储体
访问	8 位或 16 位		单个闪存字（64 位）或多个字
编程模式	程序闪存	单个闪存字（32 位）	单个闪存字（64 位）或多个字
	数据闪存	单个闪存字（32 位或 8 位）	
擦除	块擦除		扇区擦除 存储体擦除（最大 256KB）
错误码纠正	支持（RL78F23、F24）		支持
写保护	是		是，静态和动态

表 3-2. 闪存功能比较（续）

特性	RL78	MSPM0
读保护	是	是
周期	1000k（典型值）	100K（下部 32 KB）或 10k（上部 32 KB）

(1) 大多数 RL78 器件有两个存储体（一个代码存储体和一个数据存储体），一些具有 512KB 代码闪存的 RL78I1C 器件有三个存储体（两个代码存储体和一个数据存储体）。

除了上表中列出的闪存功能外，MSPM0 闪存还具有以下功能：

- 在整个电源电压范围内支持电路内编程和擦除操作。
- 内部编程电压生成。

3.2.2 闪存组织

闪存分为一个或多个存储体，每个存储体中的存储器进一步映射到一个或多个逻辑存储区域，并分配有系统地址空间以供应用程序使用。

3.2.2.1 闪存区域

表 3-3 显示了 RL78 器件和 MSPM0 器件的闪存区域。

表 3-3. 闪存区域比较

RL78（代码闪存）		MSPM0	
程序区域		MAIN	应用程序代码和数据
引导簇 (1)	小程序区域	NONMAIN	BCR 配置
	片上调试安全 ID 设置		
	闪存串行编程安全 ID		BSL 配置
	选项字节		
	CALLT 表	FACTORY	器件 ID 等
	向量表	DATA(2)	数据或 EEPROM 仿真

(1) 某些具有两个引导簇（簇 0 和簇 1）的 RL78 器件可以实现引导交换功能。
(2) 具有一个存储体的 MSPM0 器件在 BANK0（唯一存在的存储体）上实现 FACTORY、NONMAIN 和 MAIN 区域，并且 DATA 区域不可用。具有多个存储体的 MSPM0 器件也在 BANK0 上实现 FACTORY、NONMAIN 和 MAIN 区域，但包括可实现 MAIN 或 DATA 区域的其他存储体（BANK1 至 BANK4）。

3.2.2.2 MSPM0 的 NONMAIN 存储器

NONMAIN 闪存包含由 BCR 和 BSL 用于引导器件的配置寄存器，例如 FLASHSWP0 和 FLASHSWP1（静态写保护策略）。该区域不用于任何其他目的。BCR 和 BSL 都具有配置策略，这些策略可以保留为默认值（在开发和评估期间是典型值），也可以通过更改编程到 NONMAIN 闪存区域中的值来针对特定用途进行修改（在生产编程期间是典型值）。

3.2.2.3 RL78 的闪存寄存器

特殊功能寄存器区域中的闪存寄存器（SFR 和第二个 SFR）用于控制闪存编程。例如，FLPMC 寄存器可以控制代码和数据闪存的启用或禁用编程，而 DFLCTL 寄存器用于启用或禁用对数据闪存的访问。

3.2.3 嵌入式 SRAM

MSPM0 和 RL78 系列 MCU 具有用于存储应用程序数据的 SRAM。

表 3-4. SRAM 功能比较

特性	RL78	MSPM0
SRAM 存储器 (1)	RL78Gxx 范围 0.1KB 至 48KB RL78Lxx 范围 1KB 至 16KB RL78Ixx、RL78Hxx 范围 0.7KB 至 32KB RL78Fxx 范围 0.5KB 至 32KB	MSPM0Gxx 16KB 至 32KB MSPM0Lxx 2KB 至 4KB MSPM0Cxx 1KB
奇偶效验检查	支持	MSPM0Gxx：支持 MSPM0Lxx：支持 MSPM0Cxx：不支持
ECC	支持（RL78F13、F14、F15、F23、F24）	MSPM0Gxx：支持 MSPM0Lxx：支持 MSPM0Cxx：不支持
写保护（RAM 防护）	是	是

(1) RL78 SRAM 中有一个特定区域用于通用寄存器。

MSPM0 MCU 包含低功耗高性能 SRAM，可在器件支持的 CPU 频率范围内实现零等待状态访问。除代码之外，SRAM 还可用于存储信息，例如调用栈、堆和全局数据。SRAM 内容在运行、睡眠、停止和待机工作模式下完全保留，但在关断模式下会丢失。提供了一种写保护机制，允许应用程序以 1 KB 的分辨率对低 32 KB SRAM 进行动态写保护。在 SRAM 小于 32 KB 的器件上，器件为整个 SRAM 提供了写保护。在将可执行代码放入 SRAM 时写保护很有用，因为写保护可以在一定程度上防止 CPU 或 DMA 意外覆盖代码。将代码放置在 SRAM 中可以通过实现零等待状态操作和降低功耗来提高关键循环的性能。

3.3 上电和复位总结和比较

RL78 器件和 MSPM0 器件均具有最低工作电压，并具有相应的模块，可通过将器件或器件的某些部分保持在复位状态来确保器件正常启动。表 3-5 比较了这两个系列的实现方式以及哪些模块控制整个系列的上电过程和复位。

表 3-5. 上电总结和比较

RL78		MSPM0	
POR（上电复位电路）(1)	上升检测：V_{DD}>V_{POR}，释放 POR 复位信号 下降检测：V_{DD}<V_{PDR}，生成 POR 复位信号	上电复位 (POR)	上升检测：V_{DD}>POR+，释放 POR 状态，并启动带隙基准和 BOR 下降检测：V_{DD}<POR-，器件保持 POR 状态
LVD（电压检测器）- 复位模式	上升检测：V_{DD}>V_{LVD}，释放 LVD 复位信号 下降检测：V_{DD}<V_{LVD}，生成 LVD 复位信号	欠压复位 (BOR) - 0 级 (2)	上升检测：V_{DD}>BOR0+， 器件继续执行引导过程，并启动 PMU 下降检测：V_{DD}<BOR0-、 器件保持 BOR 状态
LVD（电压检测器）- 中断和复位模式	上升检测：V_{DD}>V_{LVDH}，释放 LVD 复位信号 下降检测： 1) V_{DD}<V_{LVDH}：生成中断请求信号 2) V_{DD}<V_{LVDL}：生成 LVD 复位信号	欠压复位 (BOR) - 1 至 3 级 (2)	下降检测： 1) V_{DD}<BORx- (x=1, 2, 3)，生成中断请求，BOR 电路自动将 BOR 阈值电平切换为 BOR0 2) V_{DD}<BOR0-，器件保持 BOR 状态
LVD（电压检测器）- 中断模式	上升检测：V_{DD}>V_{LVD}，释放 LVD 复位信号 在释放 LVD 复位信号后，当 V_{DD}>V_{LVD} 或 V_{DD}<V_{LVD} 时生成中断请求信号	不适用	不适用

表 3-5. 上电总结和比较（续）

RL78		MSPM0
RTCPOR（RTC 上电复位）	通过比较 RTCPOR 电路的电源电压和检测电压来复位 RTC 和 XT1 振荡器	RTC 和相关时钟通过 BOOTRST、BOR 或 POR 复位

(1) 一些 RL78 器件具有 SPOR（可选上电复位电路），其电源检测的检测电平可以通过使用选项字节来选择。
(2) 有四个可选的 BOR 阈值电平 (BOR0-BOR3)。在启动期间，BOR 阈值始终为 BOR0（最低值），使器件始终以指定的 V_{DD} 最小值启动。启动后，软件可以选择性地重新配置 BOR 电路以使用不同（更高）的阈值电平。

RL78 各个电压阈值之间的关系为：V_{PDR} < V_{POR} < 工作电压下限 < V_{LVDL} < V_{LVDH}。MSPM0 各个电压阈值之间的关系为：POR- < POR+ < BOR0- < BOR0+，BOR0+ 是能够使内部电路正常运行的指定 V_{DD} 最小值。

图 3-1 显示了 MSPM0 复位功能。MSPM0 器件有五个复位级别：上电复位 (POR)、欠压复位 (BOR)、引导复位 (BOOTRST)、系统复位 (SYSRST) 和 CPU 复位 (CPURST)。

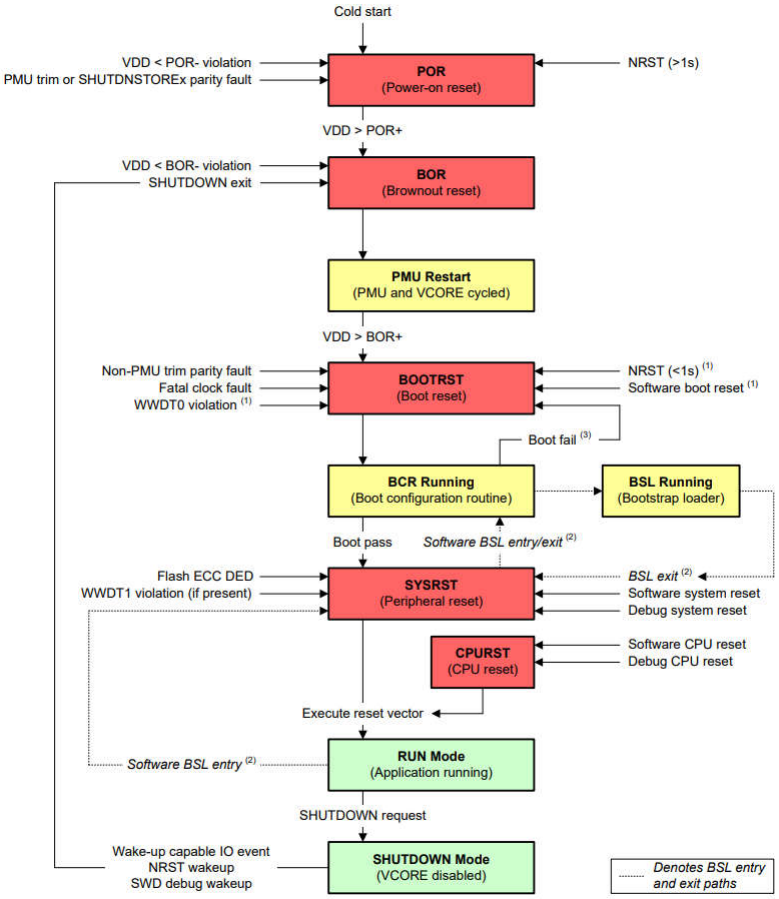


图 3-1. MSP 复位功能

3.4 时钟总结和比较

3.4.1 振荡器

RL78 和 MSPM0 器件具有多种类型的时钟源，包括内部和外部时钟源，可实现低系统成本和低功耗。表 3-6 列出了 RL78 和 MSPM0 器件中的各种不同时钟源。请注意，并非所有器件都具有所有类型的时钟源。相关详细信息，请参阅器件特定数据表。

表 3-6. 振荡器比较

类型	RL78	MSPM0L	MSPM0G	MSPM0C
内部振荡器	f_{HOCO}/f_{IH}：内部高速振荡器，频率高达 40MHz，可为 PLL 提供 80MHz 的时钟源	SYSOSC：4 MHz 至 32 MHz 的内部振荡器	SYSOSC：4 MHz 至 32 MHz 的内部振荡器，可为 PLL 提供 80 MHz 的时钟源	SYSOSC：内部 24MHz 振荡器
	f_{IM}：内部中速振荡器，频率高达 4MHz			
	f_{IL}：内部低速振荡器（15kHz 或 32.768kHz）	LFOSC：内部 32kHz 低频振荡器	LFOSC：内部 32kHz 低频振荡器	LFOSC：内部 32kHz 低频振荡器
	f_{WDT}：WDT 的内部振荡器 (15kHz)	不适用	不适用	不适用
外部振荡器	f_X：高达 20MHz 的高频振荡器，可为 PLL 提供 80MHz 的时钟源	不可用	HFXT：4MHz 至 48MHz 的外部高频振荡器，可为 PLL 提供 80MHz 的时钟源	不可用
	f_{XT}：低频振荡器（32.768kHz 或 38.4kHz）	不可用	LFXT：外部 32kHz 低频振荡器	不可用

3.4.1.1 MSPM0 振荡器

MSPM0 器件具有多种类型的时钟源，包括内部和外部时钟源，可实现低成本和低功耗。表 3-7 列出了 MSPM0 器件中的时钟源。请注意，并非所有器件都具有所有类型的时钟源。相关详细信息，请参阅器件特定数据表。

表 3-7. MSPM0 MCU 中的振荡器

类型	时钟源	说明
内部	SYSOSC	系统振荡器（出厂修整频率为 4MHz 或 32MHz，用户修整频率为 16MHz 或 24MHz）
	LFSOSC	低频振荡器（典型频率为 32kHz）
	SYSPLL (1)	具有可编程频率的系统 PLL
外部	LFXT	低频、低功耗晶体振荡器（典型频率为 32kHz）
	HFXT	高频晶体振荡器（典型频率为 4MHz 至 48MHz）

(1) 仅 MSPM0Gx 器件具有 SYSPLL 振荡器。

3.4.2 时钟信号比较

不同的时钟信号可被分频，从而为其他时钟提供信号源并分配到多个外设上。

表 3-8. 时钟信号比较

时钟说明		RL78 时钟	MSPM0 时钟
外部数字时钟输入	**高频率**	EXCLK (f_{EX})	HFCLK_IN
	低频率	EXCLKS (f_{EXS})	LFCLK_IN
高频外部时钟		f_{MX}	HFCLK
低频外部时钟		f_{SUB} (1)	选择 LFCLK_IN 和 LFXT
PLL 电路输出时钟		f_{PLL}	SYSPLLCLK0、SYSPLLCLK1、SYSPLLCLK2x (2)
主系统时钟		f_{MAIN} (3)	MCLK、ULPCLK (4) (BUSCLK)

表 3-8. 时钟信号比较（续）

时钟说明	RL78 时钟	MSPM0 时钟
CPU/外设的高频时钟	f_{MAIN} 或 f_{MP}/n (5)	选择 HSCLK (6) 和 SYSOSC
CPU/外设的低频时钟	f_{SL}	LFCLK（固定 32kHz）
源 CPU	f_{CLK}	CPUCLK
大多数外设硬件的时钟	f_{CLK}	MCLK、ULPCLK
高速外设的可用时钟	f_{MX}、f_{IH}、f_{MAIN}、f_{PLL}、f_{MP}、f_{CLK}	MCLK
低速低功耗外设的可用时钟	f_{SUB}、f_{IL}、f_{SL}、f_{CLK}	ULPCLK
固定频率时钟	不适用	MFCLK：4MHz，与 MCLK 同步
		MFPCLK：4 MHz

(1) f_{SUB} 是可能来源于低频外部振荡器 (f_{XT}) 或低频外部数字时钟输入 (EXCLKS) 的子系统时钟。
(2) SYSPLLCLK2x 的速度是 PLL 输出速度的两倍，可进行分频。
(3) RL78 的主系统时钟来源于 f_{MX} 或 f_{HOCO}/f_{IH}。
(4) MSPM0 的主系统时钟来源于 LFCLK、HSCLK 或 SYSOSC。MCLK 是 PD1 的主系统时钟，源自 MCLK 的 ULPCLK 是 PD0 的主系统时钟。
(5) f_{MP} 是主系统/PLL 选择时钟，n 可选择为 1、2、4…… 大多数 RL78 器件选择 f_{MAIN} 作为 CPU/外设的高频时钟，一些 RL78 器件选择 f_{MP}/n 作为 CPU/外设的高频时钟。
(6) HSCLK 来源于 SYSPLL 或 HFCLK。

表 3-9. 外设时钟源

外设	RL78	MSPM0
实时时钟 (RTC)	f_{IH}、f_{IL}、f_{MX}、f_{SUB}、f_{CLK}	LFCLK（LFOSC、LFXT）
UART	f_{CLK}	BUSCLK、MFCLK、LFCLK
SPI/CSI（简化的 SPI）	f_{CLK}	BUSCLK、MFCLK、LFCLK
I2C	f_{CLK}	BUSCLK、MFCLK
CAN	f_{MX}、f_{MP}、f_{CLK}	PLLCLK1、HFCLK
ADC	f_{CLK}	ULPCLK、HFCLK、SYSOSC
定时器	f_{HOCO}、f_{IL}、f_{MX}、f_{SL}、f_{PLL}、f_{MP}、f_{CLK}、f_{TMKB2} (1)	BUSCLK、MFCLK、LFCLK
比较器	f_{PLL}、f_{CLK}	ULPCLK

(1) f_{TMKB2} 时钟仅用于为 RL78 MCU 的 16 位计时器提供时钟源。

3.5 MSPM0 工作模式总结和比较

MSPM0L MCU 提供五种主要工作模式（电源模式），可根据应用要求优化器件功耗。这些模式按照功耗从高到低排列如下：RUN、SLEEP、STOP、STANDY 和 SHUTDOWN。CPU 会在运行模式中执行代码。外设中断事件可将器件从睡眠、停止或待机模式唤醒至运行模式。关断模式会完全禁用内部内核稳压器，以更大限度地降低功耗，并且只能通过 NRST、SWD 或某些 IO 上的逻辑电平匹配来实现唤醒。运行、睡眠、停止和待机模式还包括多个可配置的策略选项（例如，RUN.x），用于平衡性能与功耗。

为了进一步平衡性能和功耗，MSPM0L 器件实现了两个电源域：PD1（用于 CPU、存储器和高性能外设）和 PD0（用于低速、低功耗外设）。在运行和睡眠模式下，PD1 始终通电，但在所有其他模式下会禁用。PD0 在运行、睡眠、停止和待机模式下始终通电。PD1 和 PD0 在关断模式下都会禁用。

3.5.1 工作模式比较

表 3-10 简要比较了 RL78 和 MSPM0 器件。

表 3-10. RL78 器件和 MSPM0 器件的工作模式比较

RL78			MSPM0		
工作模式	**说明**		**工作模式**		**说明**
MAIN RUN	CPU 在主系统时钟上运行[1]	CPU、时钟和外设工作	运行	0	MCLK 和 CPUCLK 通过快速时钟源（SYSOSC、HFCLK 或 SYSPLL）运行。
	CPU 在子系统时钟上运行	CPU、时钟和外设工作		1	MCLK 和 CPUCLK 通过 LFCLK (32kHz) 运行。
				2	
HALT	CPU 在主系统时钟上运行[1]	CPU 停止运行。主系统时钟继续运行。子系统时钟的状态被保留。大多数外设功能都可以运行。	SLEEP	0	CPU 停止运行。SYSOSC 保持启用状态，而其他高速振荡器是可选的。低速振荡器保持启用状态。MCLK 通过快速时钟源运行。
	不适用	不适用		1	CPU 停止运行。SYSOSC 保持启用状态，而其他高速振荡器为禁用状态。低速振荡器保持启用状态。MCLK 通过 LFCLK 运行。
	CPU 在子系统时钟上运行	CPU 停止运行。主系统时钟停止运行。子系统时钟继续运行。大多数外设功能都可以运行。		2	CPU 停止运行。高速振荡器为禁用状态。低速振荡器保持启用状态。MCLK 通过 LFCLK 运行。
SNOOZE[2]、[3]	CPU 停止运行。f_{HOCO}/f_{IH} 开始运行，f_X、f_{EX} 和 f_{PLL} 停止运行。在 STOP 模式下使用的子系统时钟的状态将继续。ADC、UART 或 CSI 等外设功能可以在不运行 CPU 的情况下运行。		STOP	0	CPU 停止运行。SYSOSC 的状态被保留。其他高速振荡器为禁用状态。低速振荡器保持启用状态。ULPCLK 限制为 4MHz。PD0 启用且 PD1 禁用。ADC 等模拟外设可以运行。
				1	与 STOP0 相同，SYSOSC 和 ULPCLK 档位切换至 4MHz
	不适用			2	CPU 停止运行。高速振荡器为禁用状态。ULPCLK 以 32kHz 的频率运行。PD0 启用且 PD1 禁用。不支持使用 ADC。
不适用	不适用		STANDBY	0	CPU 停止运行。高速振荡器为禁用状态。所有 PD0 外设均接收 ULPCLK 和 LFCLK。不支持 ADC。
				1	与 STANDBY0 类似，仅 TIMG0/1 接收 ULPCLK 或 LFCLK。
STOP[3]	CPU 停止运行。主系统时钟停止运行。设置 STOP 模式之前的子系统时钟状态将保留。整个系统将停止。		关断		没有可用时钟且器件关断。

(1) CPU 可以在 f_{IH}/f_{HOCO}、f_X、f_{EX} 或 f_{PLL} 上运行。
(2) 仅当为 CPU/外设硬件时钟 (f_{CLK}) 选择了高速片上振荡器时，才能指定 SNOOZE 模式。
(3) 只能为 CSI、UART 和模数转换器等指定 SNOOZE 模式。在 CSI 或 UART 数据接收、计时器触发信号发出模数转换请求等情况下，MCU 将从 STOP 模式切换为 SNOOZE 模式。然后在不运行 CPU 的情况下接收 CSI 或 UART 数据，执行模数转换，等等。

3.5.2 低功耗模式下的 MSPM0 功能

在低功耗工作模式下，MSPM0 外设或外设模式的可用性或运行速度可能会受到限制。有关具体详细信息，请参阅特定于 MSPM0 器件的数据表中的“不同工作模式下支持的功能”表，例如：

- MSPM0G350x 混合信号微控制器数据表
- MSPM0L134x、MSPM0L130x 混合信号微控制器数据表
- MSPM0C110x、MSPS003 混合信号微控制器数据表

MSPM0 器件的另一项功能是某些外设能够执行异步快速时钟请求。这使 MSPM0 器件能够处于低功耗模式，在该模式下外设未处于活动状态，但仍然能够触发或激活外设。当异步快速时钟请求发生时，MSPM0 器件能够快速将内部振荡器提升至更高的速度和/或暂时使其进入更高的工作模式以处理即将发生的操作。这允许通过计时器、比较器、GPIO 和 RTC 快速唤醒 CPU；接收 SPI、UART 和 I2C；或触发 DMA 传输和 ADC 转换，同时在最低功耗模式下睡眠。有关异步时钟请求实现以及外设支持和用途的具体详细信息，请参阅 MSPM0 器件特定 TRM 中相应的章节。

- MSPM0 G 系列 80MHz 微控制器技术参考手册
- MSPM0 L 系列 32MHz 微控制器技术参考手册
- MSPM0 C 系列 24MHz 微控制器技术参考手册

3.5.3 进入低功耗模式

MSPM0 器件在执行等待事件 **_WFE( )** 或等待中断 **_WFI( )** 指令时会进入低功耗模式。低功耗模式由当前电源策略设置决定。器件电源策略由驱动程序库函数设置。以下函数调用将该电源策略设置为待机 0。

```
DL_SYSCTL_setPowerPolicySTANDBY0 ( );
```

STANDBY0 可替换为选择的工作模式。有关管理电源策略的 driverlib API 的完整列表，请参阅 MSPM0 SDK DriverLib API 指南的这一部分。另请参阅以下代码示例，这些示例演示了如何进入不同的工作模式。每个 MSPM0 器件都有类似的示例。

3.5.4 低功耗模式代码示例

导航至 SDK 安装目录，在 examples > nortos > LP name > driverlib 中找到低功耗模式代码示例。

3.6 中断和事件比较

3.6.1 中断和异常

MSPM0 和 RL78 都根据器件的可用外设来寄存和映射中断和异常矢量。表 3-11 中包含每个器件系列的中断矢量的总结和比较。中断或异常的优先级值越低，优先级就越高。当处理器当前正在处理中断时，处理器只能被具有高可编程优先级的中断抢占。

表 3-11. 中断比较

特性	RL78	MSPM0x
中断类型	可屏蔽：由器件确定并分为内部中断和外部中断。	外设中断：NVIC 支持多达 32 个本机外设中断源 (1)。
	复位：由器件确定	复位、NMI、硬故障、SVCall、PendSV、SysTick
优先级	默认优先级：由器件确定 (2)	默认优先级：NVIC 编号 (3)
	可屏蔽中断有 4 个可编程优先级：0、1、2、3	系统异常（复位、NMI、硬故障）具有固定的优先级 -3、-2 和 -1
		外设中断有 4 个可编程优先级：0、64、128、192
优先级设置	PR0xy 和 PR1xy 寄存器：用于设置可屏蔽中断优先级	NVIC 中的 IPRx 寄存器：用于设置外设中断优先级
中断屏蔽	MKxy 寄存器：用于启用/禁用相应的可屏蔽中断	外设侧的 IMASK 寄存器：用于配置哪些中断条件会传播到事件中 (4)
		NVIC 中的 ISER 和 ICER 寄存器：用于启用或禁用外设中断

(1) 除了 NVIC 外，MSPM0 器件上还可以存在中断分组模块（INT_GROUP0 和 INT_GROUP1），以便能够将 32 个以上的外设中断连接到 NVIC。
(2) 如果多个可屏蔽中断具有相同的可编程优先级，默认优先级表示相对中断优先级。

(3) 如果多个 NVIC 中断具有相同的可编程优先级，NVIC 编号表示相对中断优先级。
(4) MSPM0 的事件处理程序和相关管理寄存器如节 3.6.2 所示。

3.6.1.1 RL78 的中断管理

RL78 器件通过 PR0xy 和 PR1xy 寄存器设置每个中断条件的优先级，并通过 MKxy 寄存器启用/禁用中断条件。例如，图 3-2 显示了 RL78 的内部可屏蔽中断层次结构。在每个中断请求被应答前，必须始终发出 EI 指令来设置 IE=1 以启用中断请求应答。

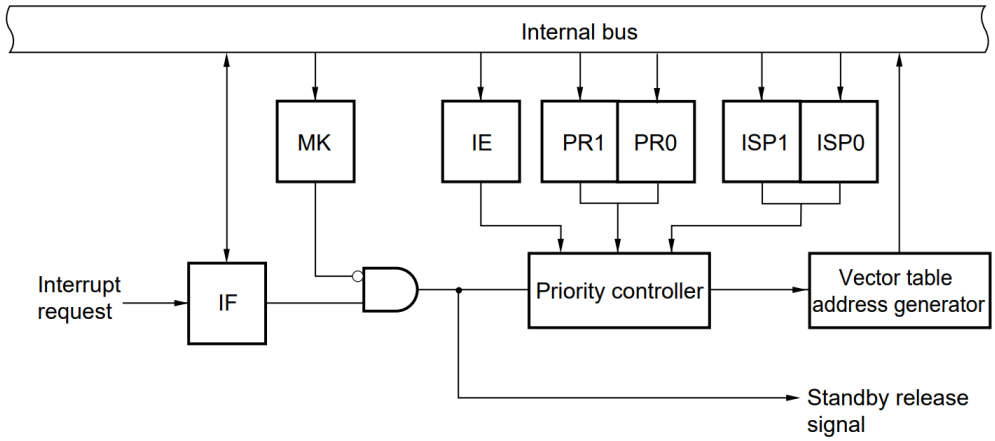


图 3-2. RL78 的内部可屏蔽中断层次结构

3.6.1.2 MSPM0 的中断管理

与 RL78 器件不同，MSPM0 器件通过 NVIC 中的 IPRx 寄存器设置每个外设中断源的优先级，并通过 NVIC 中的 ISER 和 ICER 寄存器屏蔽/取消屏蔽外设中断源。每个外设中断都包含各种中断条件。例如，作为外设中断源，UARTx 有多个中断条件，例如发送中断和接收中断等。中断条件由外设端的六个标准寄存器进行管理。

图 3-3 显示了外设中断层次结构。

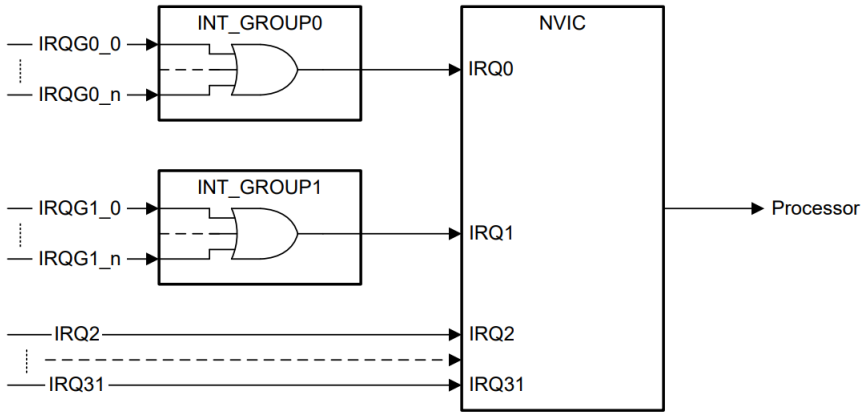


图 3-3. MSPM0 的外设中断层次结构

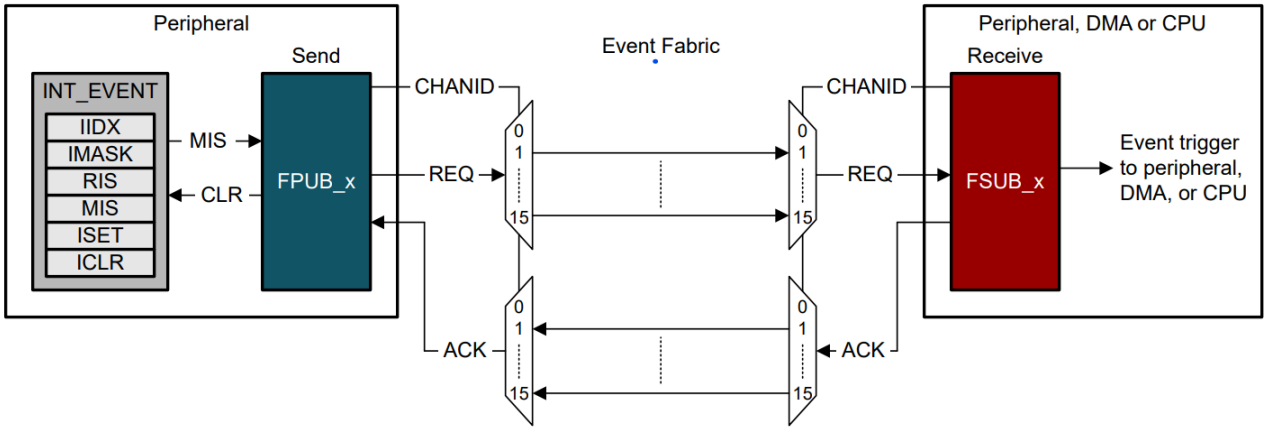
3.6.2 MSPM0 的事件处理程序

MSPM0 MCU 具有一个事件管理器，可将数字事件从一个实体传输到另一个实体。事件管理器通过一组定义的事件发布者（发生器）和订阅者（接收器）实现事件传输，这些事件发布者和订阅者通过包含静态路由和可编程路由组合的事件结构进行互连。事件管理器还可以与电源管理和时钟单元 (PMCU) 进行握手，以确保存在必要的时钟和电源域，从而执行触发事件操作。

事件管理器传输的事件包括：

- 作为中断请求 (IRQ) 传输到 CPU 的外设事件
- 作为 DMA 触发器传输到 DMA 的外设事件
- 传输到另一个外设以直接触发硬件中操作的外设事件

事件管理器通过事件结构将事件发布者连接到事件订阅者。事件结构分为三种类型：CPU 中断（固定事件路由）、DMA 路由和通用路由。例如，图 3-4 所示为通用路由。



事件管理寄存器组包含 6 个标准寄存器：RIS、IMASK、MIS、ISET、ICLR 和 IIDX。这些事件寄存器相互连接，如图 3-5 所示。取消屏蔽后，RIS 和 MIS 寄存器将指示挂起的中断，并生成一个事件。如果 CPU 中断具有 CPU 中断事件路由，则读取 IIDX 寄存器将清除 RIS 和 MIS 寄存器中的最高优先级挂起中断，并将最高优先级挂起中断的索引返回给应用软件。

图 3-4. 通用事件路由

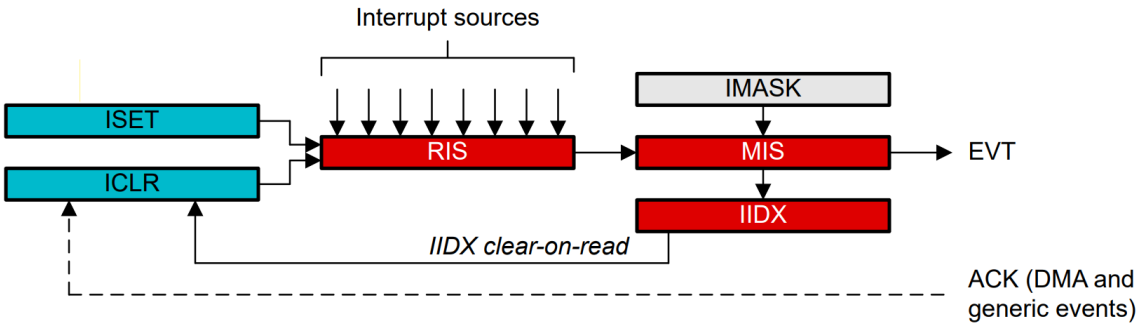


图 3-5. 事件管理寄存器关系

图 3-6 展示了事件映射。不同的外设通过不同的事件结构进行路由，可实现不同的事件转换。更多有关使用 MSPM0 事件处理程序的详细信息，请参阅 MSPM0G 技术参考手册、MSPM0L 技术参考手册或 **MSPM0C 技术参考手册**的*事件*部分。

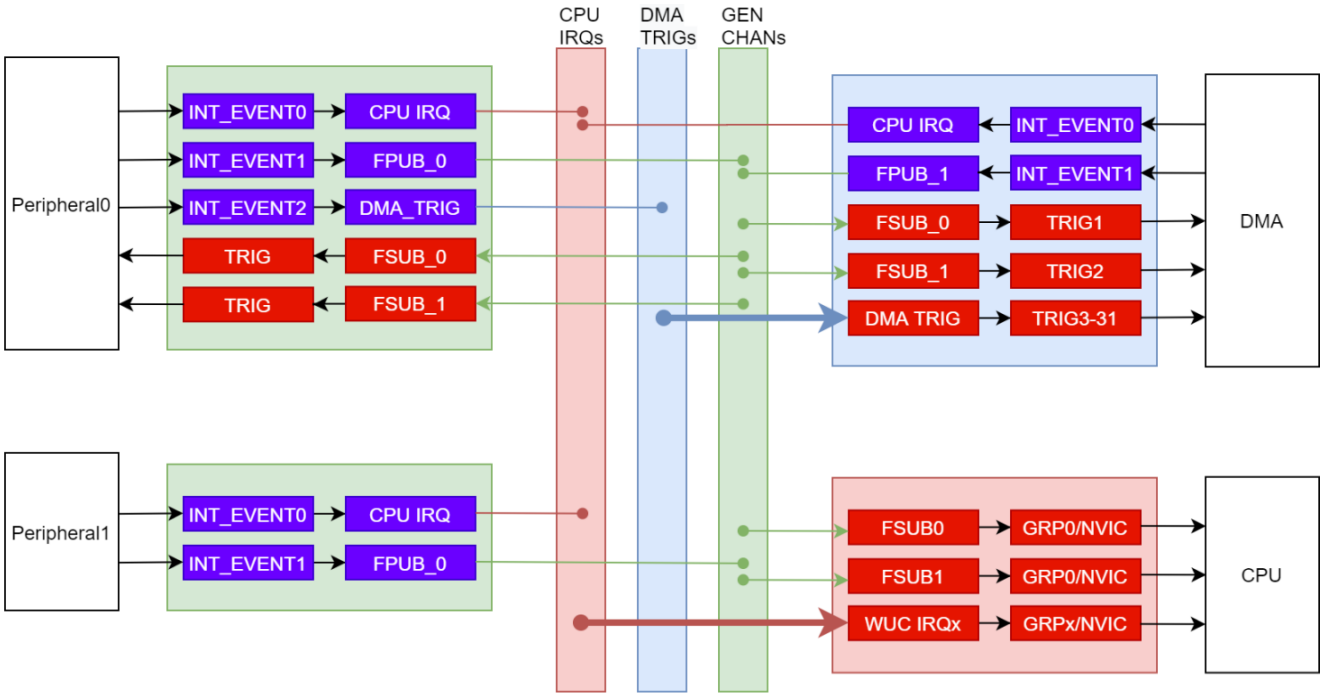


图 3-6. MSPM0 事件和中断处理

3.6.3 RL78 的事件链接控制器 (ELC)

一些 RL78 MCU 的事件链接控制器 (ELC) 可以相互连接（链接）从每个外设功能输出的事件。通过链接这些事件，RL78 MCU 可以直接协调外设功能之间的互动，而无需经过 CPU。

图 3-7 显示了 ELC 功能方框图。ELSELRn 寄存器（某些器件使用 ELISELn 和 ELOSELn 寄存器）在接收每个事件信号后将其链接到事件接收外设功能（链接目标外设功能）的操作。不同的 ELSELRn 寄存器表示不同的事件发生器，而设置到 ELSELRn 寄存器的值决定了链接目标外设功能的操作。

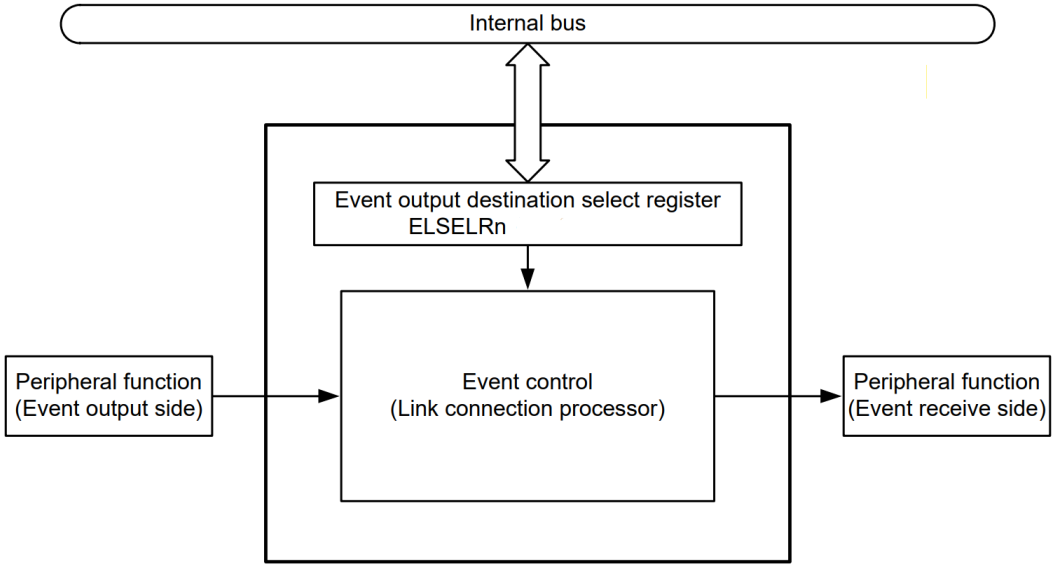


图 3-7. RL78 事件链接控制器

图 3-8 显示了中断处理和 ELC 之间的关系。将外设功能生成的事件信号用作中断控制电路中断请求的路径独立于将该信号用作 ELC 事件的路径。因此，无论中断控制如何，每个事件信号都可以用作事件接收外设功能操作的事件信号。

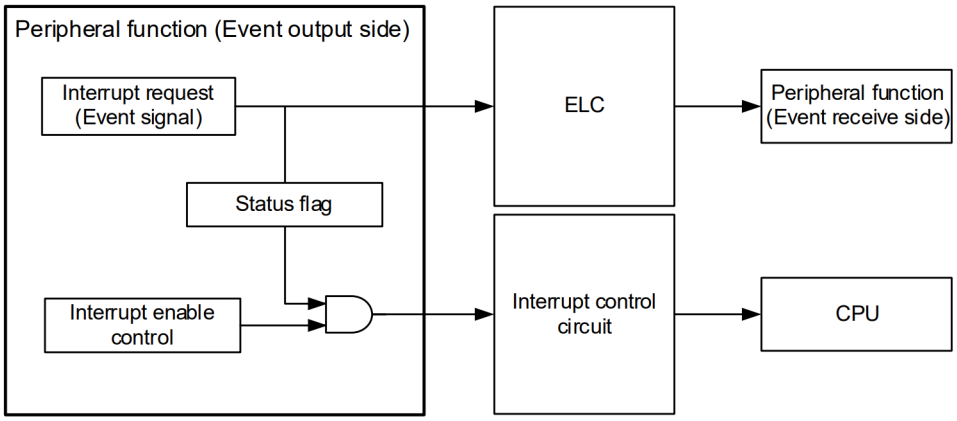


图 3-8. RL78 事件和中断处理

3.6.4 事件管理比较

RL78 和 MSPM0 具有不同的事件管理结构和功能。表 3-12 展示了节 3.6.2 和节 3.6.3 的比较。

表 3-12. 事件管理比较

特性		RL78	MSPM0
发布者		外设	外设
订阅者		CPU、DMA 触发器、外设	CPU、DMA 触发器、外设
管理模式	外设 → CPU	中断控制电路	事件管理器
	外设 → DMA	DMA 控制器 (1)	
	外设 → 外设	事件链接控制器 (1)	
路由类型		点对点 (1:1)	点对点
			一分二（分离器）(2)

(1) 并非所有 RL78 器件都具有 DMA 控制器和事件链接控制器，某些 RL78 器件既没有 A，也没有 B。
(2) 通用路由通道可以配置为一个订阅者（1:1 路由）或两个订阅者（1:2 分离器路由），具体取决于选择的通道。

3.7 调试和编程比较

3.7.1 调试比较

Arm SWD 两线制 JTAG 端口是 MSPM0 器件的主要调试和编程接口。该接口通常在应用程序开发期间和生产编程期间使用。

与 MSPM0 器件不同，RL78 器件没有 SWD 两线制 JTAG 端口。RL78 器件使用专用的闪存编程器（PG-FP5/6、E1/E2/E2 Lite/E20 片上调试仿真器）进行调试和编程。编程器和 RL78 微控制器之间的通信是使用 TOOL0 引脚通过 RL78 微控制器的专用单线 UART 进行串行通信建立的。

3.7.2 编程模式比较

3.7.2.1 MSPM0 的引导加载程序 (BSL) 编程

引导加载程序 (BSL) 编程接口是 Arm SWD 的替代编程接口。该接口仅提供编程功能，通常通过标准嵌入式通信接口使用。这允许通过与系统中用于连接其他嵌入式器件的现有接头或外部端口进行固件更新。尽管编程更新是该接口的主要用途，但 BSL 编程接口也可用于初始生产编程。

3.7.2.2 RL78 的串行编程（使用外部器件）

RL78 的串行编程模式允许使用 RL78 微控制器和连接到 UART 的外部器件（微控制器或 ASIC）进行固件更新。板载处理过程使用外部器件向 RL78 微控制器写入数据或从该微控制器删除数据。无法进行非板载写入。表 3-13 比较了 MSPM0 和 RL78 器件系列之间的编程模式差异。

表 3-13. 编程模式比较

编程功能	RL78	MSPM0
类型	使用外部器件进行串行编程	引导加载程序
安全性	存储器安全和访问限制选项(1)	安全启动选项：CRC 保护
可定制	否	是，可配置调用引脚和插件功能
调用方法	复位释放后 TOOL0 引脚处于高电平	在发生 BOOTRST、SW 进入时 1 个引脚处于高电平
命令	擦除、写入、获取信息、安全性等 (1)	连接、解锁、擦除、写入、存储器读回、恢复出厂设置、获取信息等。
密码保护	否	有

表 3-13. 编程模式比较（续）

编程功能	RL78	MSPM0
支持的接口	专用 UART	UART、I2C、SPI（需要自定义插件）、CAN（已规划插件）

(1) 默认设置会启用块擦除、写入命令和重写引导簇。使用 Security Set 命令可以禁用块擦除、禁用写入和禁用重写引导簇。

此外，RL78 器件支持自编程模式，该模式可用于通过用户程序重写闪存，因为该模式允许用户应用程序使用 RL78 微控制器自编程库来重写闪存。

4 数字外设比较

4.1 通用 I/O（GPIO、IOMUX）

MSPM0 GPIO 功能涵盖了 RL78G 提供的所有功能。RL78 使用术语“引脚功能”和“端口功能”来指代负责管理器件引脚、生成中断等的所有功能。以下是 MSPM0 GPIO 和 IOMUX 功能的说明：

- MSPM0 GPIO 指能够读取和写入 IO、生成中断等的硬件。
- MSPM0 IOMUX 指负责将不同内部数字外设连接到引脚的硬件。IOMUX 为许多不同的数字外设提供服务，包括但不限于 GPIO。

MSPM0 GPIO 和 IOMUX 共同涵盖了与 RL78 端口功能和引脚功能相同的功能。此外，MSPM0 提供了 RL78 器件所不具备的功能，例如 DMA 连接、可控输入滤波和事件功能。

表 4-1. GPIO 功能比较

功能	RL78	MSPM0
输出模式	上拉 开漏，N 沟道	推挽，上拉或下拉 开漏，下拉 高阻态
输入模式	上拉 输入阈值电平 CMOS 或 TTL 输入缓冲器 模拟	悬空 上拉或下拉 模拟
GPIO 速度选择	否	MSPM0 在所有 IO 引脚上提供标准 IO (SDIO)。 在部分引脚上提供 MSPM0 高速 IO (HSIO)。
高驱动 GPIO	根据端口和芯片类型，最大值约为 56mA @Vdd=5V 和 13.3mA @Vdd=3.3V	大约 20mA @Vdd=3.3V，称为高驱动 IO (HDIO)
原子位设置和复位	是	是
替代功能	使用 PIOR 寄存器	使用 IOMUX
快速切换	否	MSPM0 可以每个时钟周期切换一次引脚
唤醒	否	GPIO 引脚状态更改
DMA 控制的 GPIO	否	*仅在 MSPM0 上可用*
用户控制的输入滤波，可抑制小于 1、3 或 8 个 ULPCLK 周期的干扰	否	*仅在 MSPM0 上可用*
用户可控制的输入迟滞	否	*仅在 MSPM0 上可用*

GPIO 代码示例

有关 GPIO 代码示例的信息，请参阅 MSPM0 SDK 示例指南。

4.2 通用异步接收器/发送器 (UART)

RL78 和 MSPM0 都提供用于执行异步（无时钟）通信的外设。对于 RL78 的 UART，标准功能位于用作 UART 的串行阵列单元 (SAU) 外设中，高级功能位于 SAU 或独立 UART 外设（例如 LIN-UART、DALI/UART 等）中。对于 MSPM0，标准功能被命名为 main，高级功能被命名为 extend。此外，MSPM0 还提供 RL78 器件所不具备的功能，例如 IrDA 硬件支持、智能卡模式和硬件流控制等。

表 4-2. UART 功能比较

功能	RL78	MSPM0
数据方向	MSB 或 LSE	等效
使用 DMA 的连续通信	是	是
数据段	是（反向输出）	否
SNOOZE 模式下的接收	是	是，在所有低功耗模式下均有效。
单线半双工通信	是	是(1)
数据长度	5、7、8、9、16 (2)	5、6、7、8
LIN 硬件支持	是	是
DALI 硬件支持	是	是
IrDA 硬件支持	否	是
曼彻斯特编码硬件支持	否	有
智能卡模式 (ISO7816)	否	是
从低功耗模式唤醒	是 (LIN)	是
自动波特率检测	是 (LIN)	否
外部驱动器使能	否	有
硬件流控制	否	有
多处理器	否	有
Tx/Rx FIFO 深度	否	4

(1) 需要在发送和接收之间重新配置外设
(2) 数据长度 5 位适用于 UARTAx，9 位适用于 UARTx，16 位仅适用于某些器件。

UART 代码示例

有关 UART 代码示例的信息，请参阅 MSPM0 SDK 示例指南。

4.3 串行外设接口 (SPI)

MSPM0 和 RL78 都支持串行外设接口 (SPI)。对于 RL78 的 SPI，SPI 功能在 SAU（串行阵列单元）外设中用作 CSI（时钟串行接口）。此外，MSPM0 使用**控制器**和**外设**来表示 SPI 的通信双方。总的来说，MSPM0 和 RL78 SPI 支持是相当的，但存在表 4-3 列出的差异。

备注

对于 RL78，不同的器件会提供不同的 SPI 支持级别，分别称为 SPI 和简化的 SPI。

表 4-3. SPI 功能比较

功能	RL78	MSPM0
操作线	SCK、SI、SO	SCLK、PICO、POCI、CSx
控制器或外设操作	是	是
数据位宽（控制器模式）	7 至 16 位 (1)	4 至 16 位
数据位宽（外设模式）		7 至 16 位

表 4-3. SPI 功能比较（续）

功能	RL78	MSPM0
最大速度	16MHz（仅 CSI00，控制器） 8MHz（其他，控制器） 5.33MHz（其他，外设）	MSPM0L：16MHz
		MSPM0G：32MHz
单工传输（单向数据线）	是	是
硬件芯片选择管理	否	是（4 个外设）
I/O 时钟的相位控制	是	是
具有 MSB 优先或 LSB 优先移位的传输方向设置	是	是
SPI 格式支持	Motorola	Motorola、TI、MICROWIRE
硬件 CRC	否	否，MSPM0 提供 SPI 奇偶校验模式
TX FIFO 深度	否	4
RX FIFO 深度	否	4

(1) 支持的数据位长度因器件和工作模式而异。

SPI 代码示例

有关 SPI 代码示例的信息，请参阅 MSPM0 SDK 示例指南。

4.4 内部集成电路 (I2C)

MSPM0 和 RL78 均支持 I2C 外设。对于 RL78 的 I2C，串行阵列单元 (SAU) 外设（用作简化的 I2C）中提供了基本功能，而 IICA 外设中提供了高级功能。对于 MSPM0，I2C 外设中提供了 I2C 基本功能和高级功能。MSPM0 使用**控制器**和**目标**来表示通信的双方。总的来说，MSPM0 和 RL78 I2C 支持是相当的，但具有下表列出的显著差异。

表 4-4. I2C 功能比较

功能	RL78	MSPM0
控制器和目标模式	是 (IICA)	是
多控制器功能	是 (IICA)	是
最大传输速率	1MHz	等效（超快速模式）
寻址模式	7 位	7 位
地址数量（目标模式）	1 个地址	2 个地址
事件管理	否	有
时钟延展	是	是
唤醒功能（低功耗模式）	是	是
软件复位	是	是
FIFO/缓冲器	1 字节	TX：8 字节
		RX：8 字节
DMA	否	有
可编程模拟和数字噪声滤波器	否	有

I2C 代码示例

有关 I2C 代码示例的信息，请参阅 MSPM0 SDK 示例指南。

4.5 计时器（TIMGx、TIMAx）

RL78 和 MSPM0 都提供各种计时器。MSPM0 提供具有不同功能的计时器，支持从低功耗监控到高级电机控制的各种用例。

表 4-5. 计时器命名

RL78		MSPM0	
计时器名称	缩写名称	计时器名称	缩写名称
		高级控制	TIMA0
通用	计时器 RJ、RD、RG、RX	通用	TIMG0-11
时间间隔生成	间隔计时器	高分辨率	TIMG12
基础型	计时器阵列单元 (TAU)		

表 4-6. 计时器功能比较

功能	RL78 计时器	MSPM0G 计时器	MSPM0L 计时器
分辨率	8、12、16、32 位	16 位、32 位	16 位
PWM	是	是	是
捕获	是	是	是
比较	是	是	是
单次触发	是	是	是
向上/向下计数功能	是	是	是
电源模式	是	是	是
QEI 支持	否	是	否
可编程预分频器	是	是	是
影子寄存器模式	否	是	是
事件/中断	是	是	是
自动重新加载功能	是	是	是
故障处理	否	是	是

表 4-7. 计时器模块替换

RL78 计时器	MSPM0 等效功能	理由
TAU	不限	通用，16 位分辨率
32 位间隔计时器	TIMG12	32 位分辨率
计时器 RJ、RD、RG、RX	TIMA、TIMG0-11	高级用途、PWM、捕捉、比较

表 4-8. 计时器用例比较

功能	RL78 计时器	MSPM0 计时器
PWM	计时器 RJ、RD、RG	所有计时器都具有边沿对齐或中心对齐选项
捕获	计时器 RD、RG、RX	所有计时器
比较	计时器 RD	所有计时器
单次触发	所有计时器	所有计时器
预分频器	4 位预分频器	8 位预分频器
同步	TAU	所有计时器都具有该功能

计时器代码示例

有关计时器代码示例的信息，请参阅 MSPM0 SDK 示例指南。

4.6 窗口化看门狗计时器 (WWDT)

RL78 和 MSPM0 都提供窗口看门狗计时器。当应用程序未能在指定的时间窗口内签入时，窗口看门狗计时器 (WWDT) 会启动系统复位。

表 4-9. WWDT 命名

RL78	MSPM0
看门狗定时器 (WDT)	窗口化看门狗计时器 (WWDT)

表 4-10. WDT 功能比较

功能	RL78	MSPM0G
窗口模式	是	是
间隔定时器模式	否	有
LFCLK 源	是	是
中断	是	是
计数器分辨率	17 位	25 位
时钟分频器	否	有

WWDT 代码示例

有关 WWDT 代码示例的信息，请参阅 MSPM0 SDK 示例指南。

4.7 实时时钟 (RTC)

RL78 和 MSPM0[1] 都提供实时时钟 (RTC)。实时时钟 (RTC) 模块为应用程序提供时间跟踪，并以可选的二进制或二进制编码十进制数格式提供秒、分钟、小时、星期几、一月中的第几日和年的计数器。

表 4-11. RTC 功能比较

功能	RL78	MSPM0G
电源模式	是	是
二进制编码格式	否	有
年数累加	99 年	199 年
闰年校正	是	是
可定制警报的数量	1	2
内部和外部晶体	是	是
偏移校准	是	是
中断	是	是

RTC 代码示例

有关 RTC 代码示例的信息，请参阅 MSPM0 SDK 示例指南。

[1] 仅 MSPM0G 器件支持 RTC。

5 模拟外设比较

5.1 模数转换器 (ADC)

RL78 和 MSPM0 都提供 ADC 外设来将模拟信号转换为数字等效信号。两个器件系列都具有 12 位 ADC。此外，RL78 I 系列还提供 24 位 Δ-Σ ADC（不在本讨论范围内），可以考虑 MSP430F676x。表 5-1 和表 5-2 比较了 ADC 的不同功能和模式。

表 5-1. 功能集比较

功能	RL78	MSPM0G	MSPM0L
分辨率（位）	12、10、8	12、10、8	12、10、8
转换速率 (Msps)	0.888	4	1.4
过采样（位）	不适用	14	不适用
硬件过采样	16x	128x	不适用
FIFO	否	是	是
ADC 基准 (V)	内部：1.48，VDD	内部：1.4、2.5、VDD	内部：1.4、2.5、VDD
	外部： 2.4≤ V_{REFP} ≤ V_{DD} ≤ 5.5V V_{REFM} = VSS 或 V_{REFM}	外部： 1.4 ≤ V_{REF} ≤ V_{DD}	外部： 1.4 ≤ V_{REF} ≤ V_{DD}
工作电源模式	运行、SNOOZE	运行、睡眠、停止、待机 (1)	运行、睡眠、停止、待机 (1)
自动断电	是	是	是
外部输入通道 (2)	高达 31	高达 16	高达 16
内部输入通道	温度传感器、内部基准电压、触摸传感器电容	温度传感器、电源监控、模拟信号链	温度传感器、电源监控、模拟信号链
DMA 支持	是 (DTC/DMA)	是 (DMA)	是 (DMA)
ADC 窗口比较器单元	是 (ADxL)	是	是
同时采样	否	是	否
ADC 数量 (3)	高达 1	高达 2	高达 1

(1) ADC 可以在待机模式下触发，从而改变工作模式。
(2) 外部输入通道的数量因器件而异。
(3) ADC 的数量因器件而异。

表 5-2. 转换模式

RL78 (1)	MSPM0	说明
选择模式、单次转换模式/单次扫描模式	单通道单次转换	ADC 对单个通道进行一次采样和转换
扫描模式、单次转换模式/单次扫描模式	序列通道单次转换	ADC 对序列通道进行采样并转换一次。
选择模式、顺序转换模式/连续扫描模式	单通道重复转换	重复单通道连续采样，转换一个通道
扫描模式、顺序转换模式/连续扫描模式	序列通道重复转换	对序列通道进行采样和转换，然后重复相同的序列

(1) RL78 ADC 转换模式的名称因器件而异，此处显示了使用“/”分隔的两种命名方式。

ADC 代码示例

有关 ADC 代码示例的信息，请参阅 MSPM0 SDK 示例指南。

5.2 比较器 (COMP)

RL78 和 MSPM0 系列器件都在某些器件上提供集成比较器作为可选外设。在 RL78 中，比较器表示为 CMP、COMP 或 COMPARATOR x，而在 MSPM0 中表示为 COMPx。在 RL78 G1F 系列中，这些 x 编号为 0-1，而在 MSPM0 系列中，这些 x 编号为 0-2。在 RL78 系列中，G1F 系列用于 BLDC 电机并具有高级功能比较器，而 RL78 的其他系列主要具有基本功能比较器。比较器模块可以从各种内部和外部源获取输入，并可用于触发电源模式变化或截断/控制 PWM 信号。表 5-3 汇总了 MSPM0 和 RL78 比较器模块逐功能比较结果。

表 5-3. COMP 功能集比较

功能	RL78	MSPM0G	MSPM0L
可用的比较器	高达 2	高达 3	高达 1
输出路由	外部	多路复用 I/O 引脚	多路复用 I/O 引脚
	事件链接器控制器	中断/事件接口	中断/事件接口
正输入	外部 4 个模拟引脚输入	多路复用 I/O 引脚	多路复用 I/O 引脚
		DAC12 输出 (1)	DAC8 输出
		DAC8 输出	
	PGA 输出 比较器 0	内部 V_{REF}：1.4 V 和 2.5 V	/
		OPA1 输出 (2)	OPA1 输出 (2)
负输入	外部模拟引脚输入	多路复用 I/O 引脚	多路复用 I/O 引脚
		内部温度传感器	内部温度传感器
	内部 V_{REF}：1.45 V	内部 V_{REF}：1.4 V 和 2.5 V	DAC8 输出
	8 位 DAC 比较器 1	DAC8 输出	OPA0 (3) 输出
		OPA0 输出 (3)	
可编程迟滞	无、10mV、20mV、30mV	无、10mV、20mV、30mV	无、10mV、20mV、30mV
		从 0V 到 V_{REF}/V_{DD} 的其他值（使用 DAC8）	从 0V 到 V_{DD} 的其他值（使用 DAC8）
寄存器锁	否	是，某些 COMP 寄存器（写入时需要密钥）	是，某些 COMP 寄存器（写入时需要密钥）
窗口比较器配置	带有 TAU0 的计时器窗口	是	否（单个 COMP）
输入短路模式	否	是	是
工作模式	运行	高速，低功耗	高速，低功耗
快速 PWM 关断	是	是（通过 TIMA 故障处理程序）	否
输出滤波	消除数字滤波器（3 个周期）	消隐滤波器	消隐滤波器
		可调节模拟滤波器	可调节模拟滤波器
输出极性控制	是	是	是
中断	上升沿	上升沿	上升沿
	下降沿	下降沿	下降沿
	双边沿	输出就绪	输出就绪
交换输入模式 (4)	否	是	是

(1) 仅适用于具有 DAC12 外设的器件
(2) 仅适用于具有 OPA1 外设的器件
(3) 仅适用于具有 OPA0 外设的器件
(4) 在启用交换输入模式时，比较器正负端子的输入信号会进行交换。此外，比较器的输出信号也会反相。

COMP 代码示例

有关 COMP 代码示例的信息，请参阅 MSPM0 SDK 示例指南。

5.3 模数转换器 (DAC)

RL78 和 MSPM0 系列器件都提供 12 位、8 位 DAC 外设，用于为各种应用执行数模转换。在 RL78 文档中，12 位 DAC 称为 12 位数模转换器，8 位 DAC 称为 8 位数模转换器 (DAC)。只有 RL78 I1E 和 L1A 系列器件上提供 12 位数模转换器。在 MSPM0 中，12 位 DAC 外设称为 DAC12。这将 DAC12 与 8 位 DAC 区分开来，后者可用于给定 MSPM0 器件中包含的每个比较器外设。本文档的比较器部分介绍了这些额外的 8 位 DAC。该 DAC12 外设仅在 MSPM0G 系列器件上可用。

表 5-4 总结了 RL78 和 MSPM0G 的 12 位 DAC 外设的功能。

表 5-4. DAC 功能集比较

功能	RL78	MSPM0
分辨率	12 位	12 位 (11 ENOB)
输出速率	33kSPS	1MSPS
输出通道	2 (1)	1 (2)
数据格式	12 位右对齐、12 位左对齐	8 位右对齐，12 位右对齐，二进制补码或直接二进制
DMA 集成	是 (DTC)	是
输出路由	外部引脚	外部引脚
		内部外设连接：OPA IN+、COMP IN+、ADC0
内部基准电压	是，1.48V	是，2.5V 或 1.4V
外部基准电压	是	是
FIFO	否	是
输出缓冲器	否	有
可配置输出失调电压	否	有
自校准模式	否	有
触发源	事件链接	内部专用采样时间发生器、DMA 中断/事件、FIFO 阈值中断/事件、两个硬件触发器 (可从事件结构获得)

(1) 仅在具有 12 位分辨率的 L1A 器件上提供。
(2) 双 DAC 通道计划用于未来的 MSPM0G 器件。

DAC12 代码示例

有关 DAC12 代码示例的信息，请参阅 MSPM0 SDK 示例指南。

5.4 运算放大器 (OPA)

RL78 器件系列包含一系列具有不同类型放大器的器件。RL78 中的 PGA 和放大器单元包含三个 OPA 和一个 PGA。MSPM0 OPA 模块非常灵活，可以单独或组合替换检测或控制应用中的许多分立式放大器。MSPM0 OPA 模块和 RL78 放大器单元之间的差异如表 5-5 所示。

表 5-5. OPA 功能集比较

功能	RL78	MSPM0
GBW (低功耗模式)	1.5kHz (PGA)、60kHz (OPA)	1MHz (低功耗模式)
GBW (标准模式)	67kHz (PGA)、1.7MHz (OPA)	6MHz (标准模式)
放大器配置或类型	通用放大器	通用模式
	缓冲模式	等效
	PGA 外设	PGA 模式 (反相或同相)
	差分放大器模式	等效
	否	级联放大器模式
输入/输出路由	外部引脚布线	外部引脚布线
	与 DAC 和 ADC 的内部连接	ADC 和 COMP 模块的内部连接

表 5-5. OPA 功能集比较（续）

功能	RL78	MSPM0
故障检测	否	烧毁电流源 (BCS)
斩波稳定	否	标准（可选斩波频率）
		ADC 辅助斩波
		禁用
基准电压	内部偏置电压 1.0V	内部 VREF（仅限 MSPM0G 器件）
	DAC12 (H1D)	DAC12（仅限 MSPM0G 器件）
	DAC8 (H1D)	DAC8（仅限具有 COMP 模块的器件）

OPA 代码示例

有关 OPA 代码示例的信息，请参阅 MSPM0 SDK 示例指南。

5.5 电压基准 (VREF)

RL78 和 MSPM0 都具有内部基准，可用于为内部外设提供基准电压，但只有 MSPM0 可以向外部外设输出内部基准电压。

表 5-6. VREF 功能集比较

功能	RL78	MSPM0G	MSPM0L
内部基准 (V)	1.45、SBIAS	1.4、2.5	1.4、2.5
外部基准 (V)	2.4 ⩽ V_{REFP} ⩽ 5.5V	外部：1.4 ⩽ V_{REF} ⩽ V_{DD}	外部：1.4 ⩽ V_{REF} ⩽ V_{DD}
输出内部基准	是	是	是
在内部连接到 ADC	是	是	是
在内部连接到 DAC	是	是	否
在内部连接到 COMP	是	是	否
在内部连接到 OPA	是	是	否

对于 MSPM0 VREF，您必须启用电源位 PWREN 位 0 (ENABLE)。

VREF 代码示例

有关使用 VREF 的代码示例，请参阅 MSPM0 SDK 示例指南。

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 TI 的销售条款或 ti.com 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265