



摘要

本应用手册可协助您完成从 STMicroelectronics STM8L 和 STM8S 平台至德州仪器 (TI) MSPM0L 和 MSPM0C MCU 生态系统的迁移。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 STM8 开发环境知识快速提升 MSPM0 系列 MCU 的性能。

内容

1 MSPM0 产品系列概述	4
1.1 简介.....	4
1.2 STM8 MCU 与 MSPM0 MCU 的产品系列比较.....	4
2 生态系统和迁移	5
2.1 生态系统比较.....	5
2.2 迁移过程.....	12
2.3 示例.....	27
3 内核架构比较	34
3.1 CPU.....	34
3.2 嵌入式存储器比较.....	34
3.3 上电和复位总结和比较.....	36
3.4 时钟总结和比较.....	38
3.5 MSPM0 工作模式总结和比较.....	39
3.6 中断和事件比较.....	41
3.7 调试和编程比较.....	45
4 数字外设比较	47
4.1 通用 I/O (GPIO、IOMUX)	47
4.2 通用异步接收器/发送器 (UART).....	48
4.3 串行外设接口 (SPI).....	48
4.4 内部集成电路接口 (I2C).....	49
4.5 计时器 (TIMGx、TIMAx)	50
4.6 窗口化看门狗计时器 (WWDT).....	51
5 模拟外设比较	52
5.1 模数转换器 (ADC).....	52
5.2 比较器 (COMP).....	52
5.3 电压基准 (VREF).....	53
6 总结	54
7 参考资料	54
8 修订历史记录	54

插图清单

图 2-1. MSPM0 生态系统概览.....	5
图 2-2. MSPM0 SDK 结构.....	6
图 2-3. MSPM0 SysConfig.....	8
图 2-4. SysConfig 中的 GPIO 配置.....	9
图 2-5. MSPM0 调试.....	9
图 2-6. MSPM0G3507 LaunchPad 概览.....	10
图 2-7. Arm Cortex 10 引脚定义.....	11
图 2-8. MSPM0G3507 LaunchPad 特性功能.....	11
图 2-9. MSPM0 迁移流程图.....	12

图 2-10. MSPM0C、MSPM0L 和 MSPM0G 产品系列.....	13
图 2-11. MSPM0 产品选择工具.....	14
图 2-12. MSPM0 重要文档列表.....	14
图 2-13. MSPM0 相关技术文档列表.....	15
图 2-14. 器件订购和质量视图.....	15
图 2-15. CCS 安装.....	16
图 2-16. 安装 CCS - 选择 MSPM0 支持元件.....	16
图 2-17. 安装 CCS - 选择下载 J-Link.....	17
图 2-18. CCS 启动工作区.....	17
图 2-19. 在 CCS 中创建新工程.....	18
图 2-20. 常用功能.....	18
图 2-21. 常用调试功能.....	18
图 2-22. 常用工程设置.....	19
图 2-23. 下载 MSPM0 SDK.....	19
图 2-24. 安装 MSPM0 SDK.....	19
图 2-25. MSPM0 SDK 文件夹.....	20
图 2-26. 文档概览.....	21
图 2-27. 导入 CCS 工程.....	21
图 2-28. 从 SDK 中选择程序.....	22
图 2-29. 删除重复的工程.....	22
图 2-30. 工程和 README.md.....	23
图 2-31. CCS 工程概述.....	23
图 2-32. 添加相关文件.....	24
图 2-33. 设置“Include Options”.....	24
图 2-34. 成功构建.....	24
图 2-35. Ultra Librarian 工具入口.....	25
图 2-36. MSPM0 最小系统.....	25
图 2-37. MSPM0 最小系统注意事项.....	25
图 2-38. 创建程序文件.....	26
图 2-39. 程序软件和工具.....	26
图 2-40. 代码示例文件.....	27
图 2-41. SysConfig 中的 PWM 配置.....	28
图 2-42. 获取每一项的详细信息.....	28
图 2-43. 引脚配置.....	29
图 2-44. SysConfig 更新的文件.....	29
图 2-45. 硬件设置.....	30
图 2-46. 添加断点解决方案.....	30
图 2-47. Ultra Librarian 工具下载.....	31
图 2-48. 运行 Altium Designer 脚本.....	32
图 2-49. PCB 库和原理图文件.....	32
图 2-50. 导入库.....	33
图 3-1. MSP 复位功能.....	37
图 3-2. MSPM0 的外设中断层次结构.....	42
图 3-3. 中断处理流程图.....	42
图 3-4. 通用事件路由.....	43
图 3-5. 事件管理寄存器关系.....	43
图 3-6. MSPM0 事件和中断处理.....	44

表格清单

表 1-1. TI MSPM0Lx, MSPM0Cx 与 STM8Lx, STM8Sx 进行比较.....	4
表 2-1. 生态系统比较.....	5
表 2-2. CCS 和 STVD 之间的比较.....	7
表 2-3. MSPM0 支持的 IDE 概览.....	7
表 2-4. MSPM0 调试器比较.....	10
表 2-5. MSPM0 示例覆盖范围.....	20
表 2-6. 空工程描述.....	20
表 3-1. CPU 功能集比较.....	34
表 3-2. 闪存和 EEPROM 的特性.....	34

表 3-3. 闪存和 EEPROM 区域.....	35
表 3-4. SRAM 功能比较.....	36
表 3-5. 上电总结和比较.....	36
表 3-6. 振荡器比较.....	38
表 3-7. 时钟信号比较.....	38
表 3-8. 外设时钟源.....	39
表 3-9. STM8 器件和 MSPM0 器件的工作模式比较.....	39
表 3-10. 中断比较.....	41
表 3-11. 事件管理比较.....	45
表 3-12. BSL 功能比较.....	46
表 4-1. GPIO 功能比较.....	47
表 4-2. UART 标准功能比较.....	48
表 4-3. UART 高级功能比较.....	48
表 4-4. SPI 功能比较.....	49
表 4-5. I2C 功能比较.....	49
表 4-6. 计时器命名.....	50
表 4-7. 计时器功能比较.....	50
表 4-8. 计时器模块替换.....	50
表 4-9. 计时器用例比较.....	50
表 4-10. WWDT 命名.....	51
表 4-11. WDT 功能比较.....	51
表 5-1. 功能集比较.....	52
表 5-2. 转换模式.....	52
表 5-3. COMP 功能集比较.....	53
表 5-4. VREF 功能集比较.....	53

商标

LaunchPad™, Code Composer Studio™, EnergyTrace™, and BoosterPack™ are trademarks of Texas Instruments.

Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. μ Vision® is a registered trademark of Arm Limited.

所有商标均为其各自所有者的财产。

1 MSPM0 产品系列概述

1.1 简介

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，可让客户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 STM8 MCU 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，帮助完成从 STM8 MCU 到 MSPM0 MCU 的迁移。

1.2 STM8 MCU 与 MSPM0 MCU 的产品系列比较

表 1-1. TI MSPM0Lx, MSPM0Cx 与 STM8Lx, STM8Sx 进行比较

		ST Micro STM8 L 系列	ST Micro STM8 S 系列	TI MSPM0 Lx 系列	TI MSPM0 Cx 系列	TI MSPM0 Hx 系列
内核		STM8 CPU 内核		Arm Cortex-M0+		
最大频率		16MHz	24MHz	32MHz	24MHz, 32MHz ⁽¹⁾	32MHz
电源电压		1.6-3.6V	2.95-5.5V	1.62-3.6V	1.62-3.6V	4.5-5.5V ⁽²⁾
最高温度		-40-125°C	-40 至 125°C	-40 至 125°C	-40 至 125°C	-40 至 125°C
存储器		64KB 至 2KB	128KB 至 4KB	64KB 至 8KB	16KB 至 8KB	64KB 至 32KB
RAM		高达 4 KB	高达 6 KB	高达 4 KB	高达 8 KB	高达 8 KB
GPIO (最大值)		41	38	28	45	45
RTC		是	否	是	是 (C1103 和 C1104 不支持)	是
模拟	ADC	高达 12 位 x 28 通道	高达 10 位 x 16 通道	1.68x 1Msps 12 位 ADC (16 通道)	1x 1.5-Msps 12 位 ADC (最多 27 通道)	1x 1.5-Msps 12 位 ADC (最多 27 通道)
	DAC	高达 12 位 x 2 通道	无	8 位	8 位 (C1103 和 C1104 不支持)	无
	比较器	3 μs 传播延迟	无	1x 高速	1x 高速 (C1103 和 C1104 不支持)	无
通信	UART	1Mbit/s, 最多 3 个 UART	1Mbit/s, 最多 2 个 UART	2x 4Mbit/s	2x 4Mbit/s	3x 4Mbit/s
	I2C	1、100 和 400Kbit/s	1、100 和 400Kbit/s	2, 高达 1Mbit/s	1, 高达 1Mbit/s	2, 高达 1Mbit/s
	SPI	10Mbit/s	10Mbit/s	16Mbit/s	12Mbit/s	16Mbit/s
	CAN	无	1Mbit/s, 最多 3 个邮箱	无	无	无
	LIN	UART 支持		UART 支持		
其他重要外设/特性		LCD 驱动器 蜂鸣器 触摸感应 (STM8L CT 库) DMA IR 接口	蜂鸣器 触摸感应 (STM8S RC 库)	2 个运算放大器 LCD (L2228) DMA	超小型 QFN 封装 (2x2)、BGA 封装 (0.861x1.6)、0.5/0.65mm 间距封装, 引脚与业内 DMA 兼容	5V 电源、DMA、窗口看门狗计时器
计时器数量		3、4、5	3/4	7	5 (C1103 和 C1104 支持 3 个计时器)	4
引脚数		高达 68	高达 68	16-80 引脚	8-48 引脚	20-48 引脚
低功耗		有效: 在 16MHz 和 1.6mA 时 停止: 0.3 μA	有效: 在 16MHz 和 1.8mA 时 停止: 5 μA	有效: 71 μA/MHz 待机: 1 μA	有效: 100 μA/MHz 待机: 5 μA	不适用

(1) MSPM0C1103 和 MSPM0C1104 : 24MHz、MSPM0C1105 和 MSPM0C1106 : 32MHz

(2) H 系列首款器件的电压为 4.5 - 5.5V，未来支持宽电源范围。

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

特性	STM8 器件	MSPM0 器件
代码源	标准外设库 (嵌入式软件驱动程序和示例集) 固件包	MSPM0-SDK (DriverLib、中间件、RTOS、代码示例)
IDE	STVD、IDEA、IAR-EWSTM8、iSYS-WinIDEA 和 Arduino IDE	CCS、IAR、Keil
软件配置	STM8CubeMX	SysConfig
闪存编程工具	FLASHER-STM8 STVP(STM8)	UniFlash
编程器	FlashRunner	MSP-GANG，C-GANG
调试器	ST-Link v2，ic5000	XDS110，J-LINK
硬件	STM8-SO8-DISCO、STM8S-DISCOVERY、STM8SVDISCOVERY、STM8L-DISCOVERY、NUCLEO-8S208RB、NUCLEO-8L152R8	LP-MSPM0G3507 LaunchPad、LP-MSPM0L1306 LaunchPad、LP-MSPM0C1104 LaunchPad、LP-MSPM0C1106 LaunchPad、LP-MSPM0H3216 LaunchPad

图 2-1 显示了 MSPM0 生态系统概览。

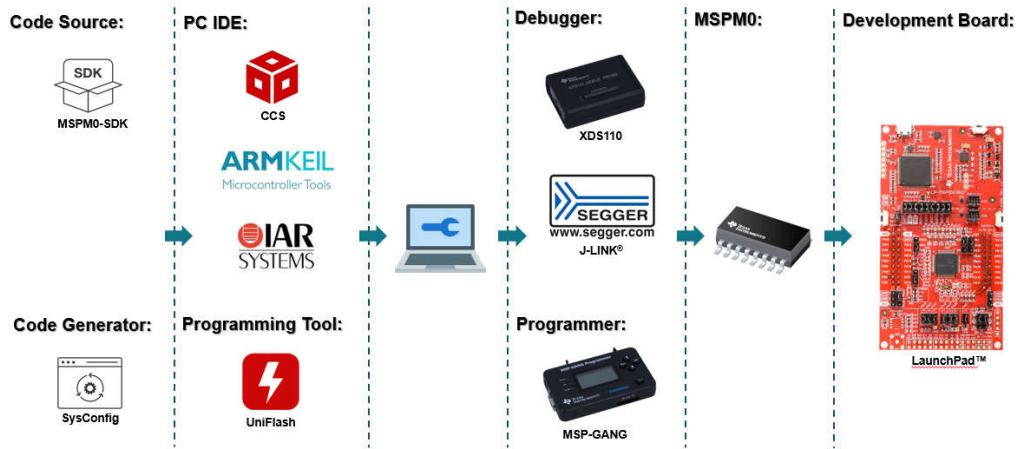


图 2-1. MSPM0 生态系统概览

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。

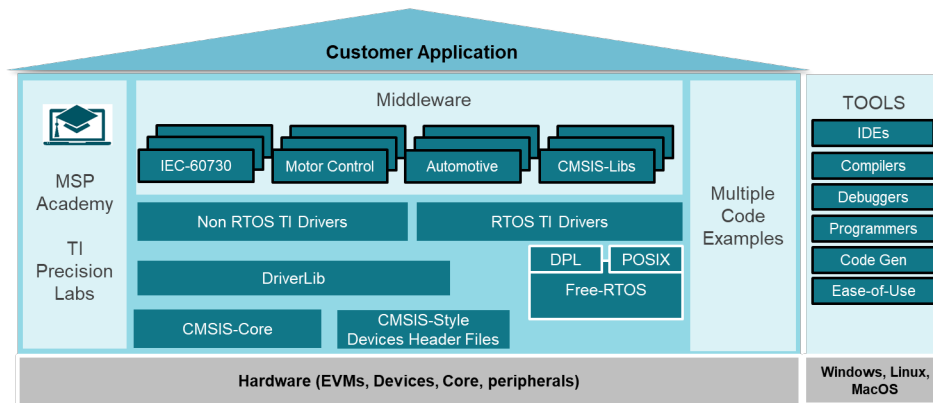


图 2-2. MSPM0 SDK 结构

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。

对于 STM8，ST 免费提供了标准外设库、RTOS、引导加载程序等。STM8 还有大量的中间件和应用领域，如触摸感应、电机控制、Lin 库，MSPM0 几乎涵盖了这些内容。

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。

STM8 的典型 IDE 是 STMicroelectronics 提供的 STVD。它可以下载示例代码并具有易于使用的 Eclipse 代码编辑器。STVD 只有一个汇编编译器，而不是 C 编译器，因此用户需要安装额外的 C 编译器，即 Cosmic 工具。

Cosmic 推出了适用于 STM8 的编译器。可免费使用高达 32KB 的代码。因此，STM8 用户倾向于通过 IAR 开发一个工程，MSPM0 也可以与 IAR 配合使用。

在 TI 中，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。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。两者均可完全免费使用 CCS。

两个 IDE 之间的差异和相似性如表 2-2 所示。

表 2-2. CCS 和 STVD 之间的比较

IDE	CCS	STVD
许可	免费	免费
编译器	TI Arm Clang , GCC	Cosmic , raisonance
IDE 中集成的电流消耗	EnergyTrace	不支持 (STM8CubeMX 支持)
外设的 API 函数帮助	不支持	不支持
显示语言	英语	英语
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	STM8CubeMX

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 2-3 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 2-3. MSPM0 支持的 IDE 概览

IDE	CCS	IAR	Keil
许可	免费	付费	付费
编译器	TI Arm Clang , GCC	适用于 Arm 的 IAR C/C++ 编译器	Arm 编译器版本 6
磁盘大小	0.88G (ccs20.1.1)	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 STM8CubeMX 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，如图 2-3 所示。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](#)。

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。

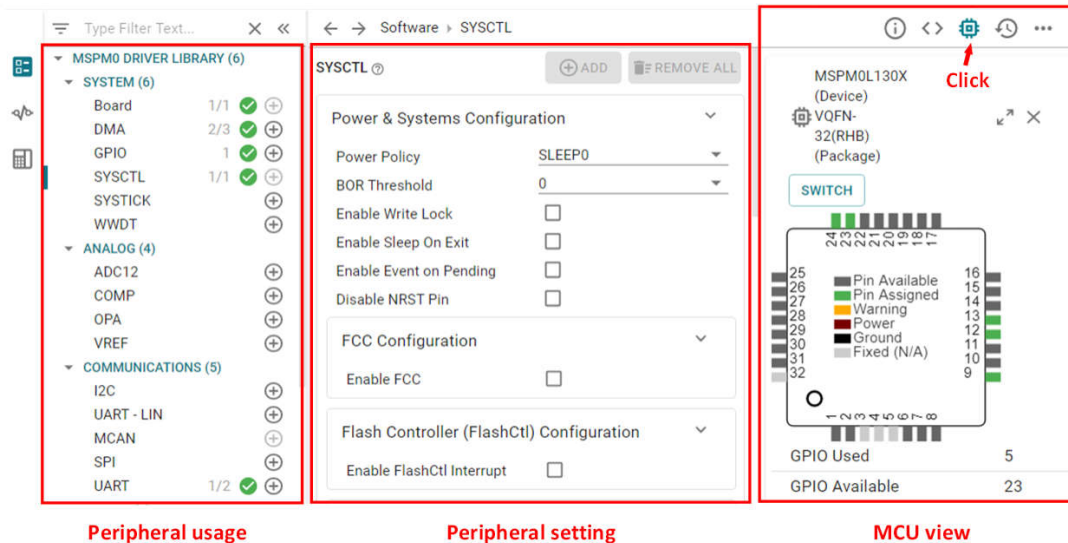


图 2-3. MSPM0 SysConfig

以下是 SysConfig 和 STM8CubeMX 之间相同和不同的功能：

- 在这两者中都允许创建、保存和导入以前保存的工程。STM8CubeMX 工程附带了 .ioc8 文件，该文件可以保存在其他 .ioc8 文件旁边的任何位置。但 STM8CubeMX 不支持 C 代码生成，它与 SysConfig 不同并可能会延长开发时间。
- 这两者都可以轻松实现引脚排列和时钟树配置，可从芯片视图中看到。
- STM8CubeMX 支持功耗计算，对于 MSPM0，可以在 CCS IDE 中实现。
- SysConfig 支持更全面的配置功能。与只能配置引脚排列或外设简单配置的 STM8CubeMX 不同，SysConfig 提供了更具体和详细的初始化配置。如图 2-4 所示，SysConfig 可以配置 GPIO 类型以及内部上拉/下拉电阻器等功能。
- STM8CubeMX 附带了更新程序机制，可配置为自动或按需更新检查，支持 STM8CubeMX 自更新。SysConfig 不支持该功能，用户可以从 [SysConfig 下载处](#) 下载最新版本。

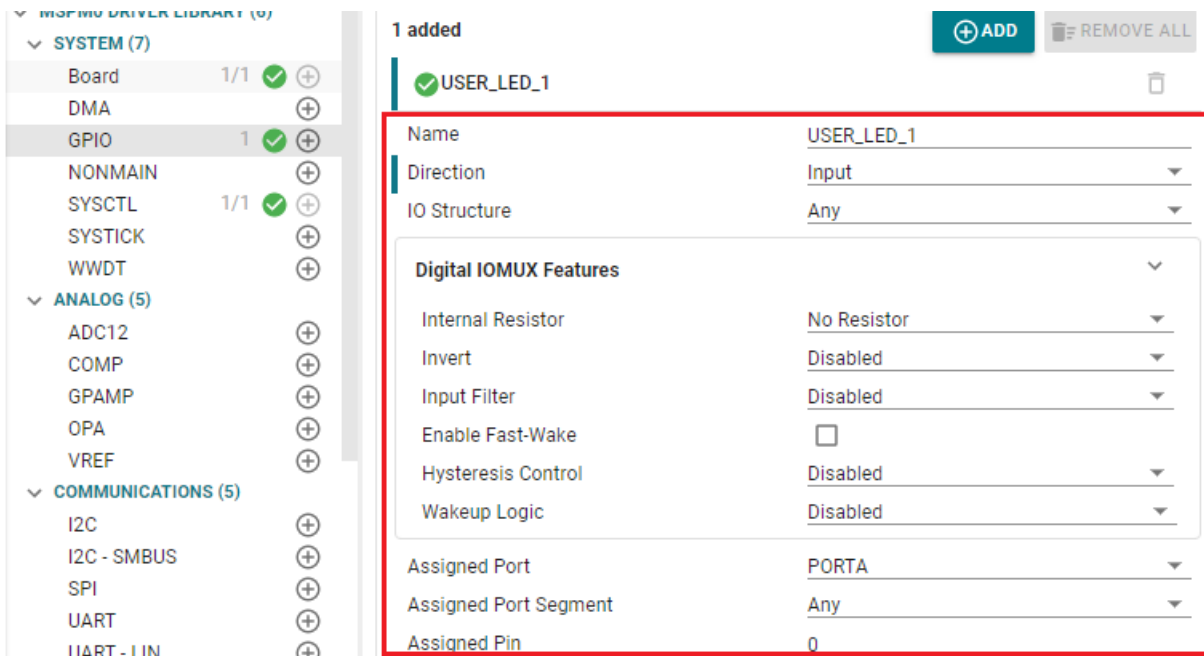


图 2-4. SysConfig 中的 GPIO 配置

2.1.4 调试工具

STM 支持单线制接口模块 (SWIM) 和调试模块 (DM)。电路内调试模式或电路内编程模式通过具有超快存储器编程功能的单线硬件接口进行管理。与电路内调试模块耦合，它还提供了非侵入式仿真模式，这使得电路内调试器极其强大，性能接近于功能全面的仿真器。STM8 的典型调试器是 ST-LINK。SWIM 和 JTAG/串行线调试 (SWD) 接口用于与位于应用板上的任一 STM8 微控制器进行通信。

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 2-5。

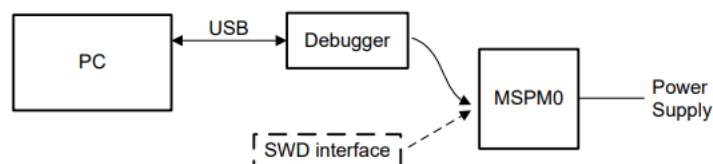


图 2-5. MSPM0 调试

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功能。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](#)。

表 2-4 显示了支持 MSPM0 的 XDS110 和 J-LINK 调试器之间的不同功能汇总。

表 2-4. MSPM0 调试器比较

特性	XDS110	XDS110 OB ⁽¹⁾	J-Link
cJTAG (SBW)	✓	✓	✓
BSL ⁽²⁾ 工具	✓	✓	
反向通道 UART	✓	✓	2.5G (µVision V5.37.0)
电源	1.8V 至 3.6V	3.3, 5V	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 和 Keil。

2.1.5 LaunchPad

与 STM8 一样，MSPM0 也具有相应的 LaunchPad 开发套件来支持快速开发。

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。

图 2-6 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。

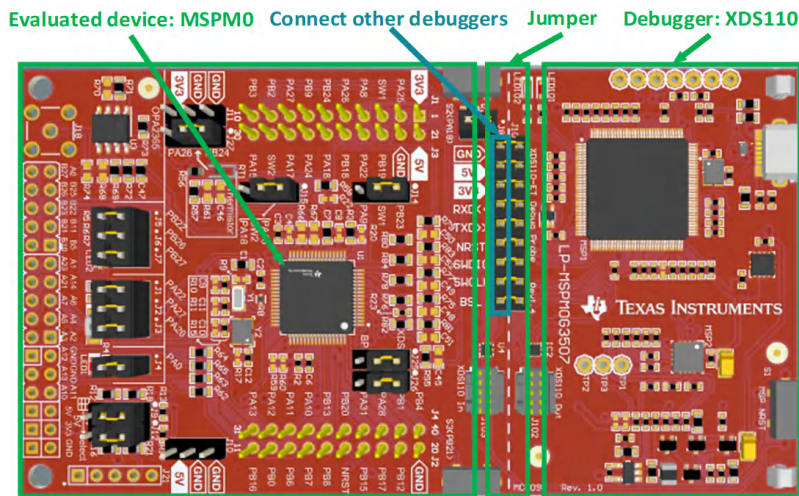


图 2-6. MSPM0G3507 LaunchPad 概览

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 2-7 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。

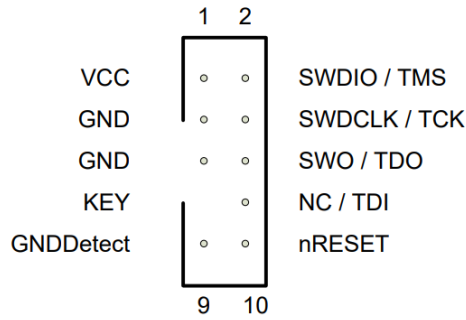


图 2-7. Arm Cortex 10 引脚定义

图 2-8 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](#)
- LP-MSPM0C1104 LaunchPad 开发套件 [LP-MSPM0C1106 评估板 | 德州仪器 TI.com](#)

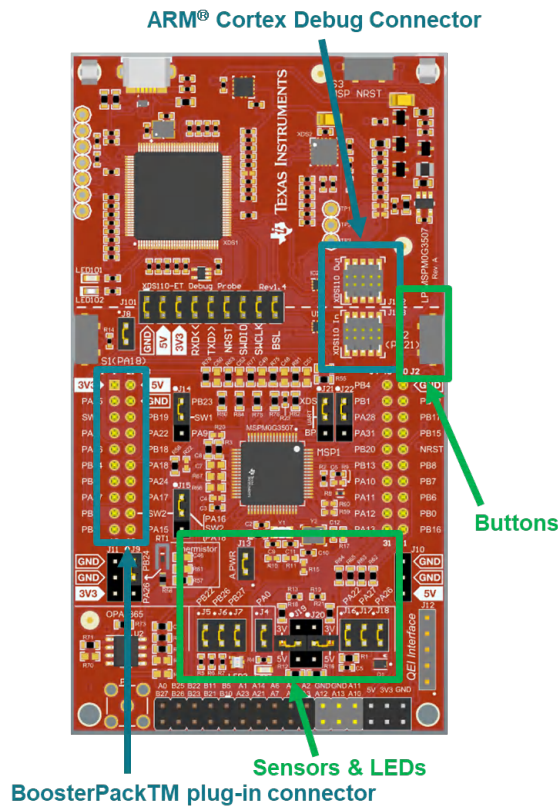


图 2-8. MSPM0G3507 LaunchPad 特性功能

2.2 迁移过程

为了顺利迁移到 MSPM0，详细过程将按图 2-9 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。

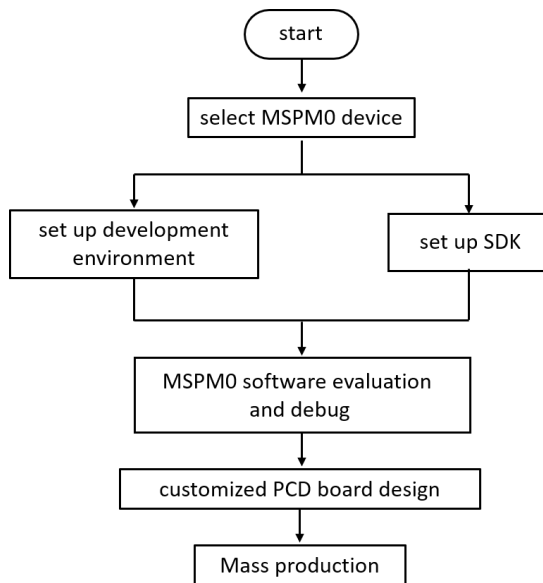


图 2-9. MSPM0 迁移流程图

2.2.1 第 1 步：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 2-10 所示为 MSPM0 C、L 和 G 产品系列，相关信息可在 TI 官方网站上找到。这两个产品系列都根据存储器和封装来区分器件，因此选择器件更容易。

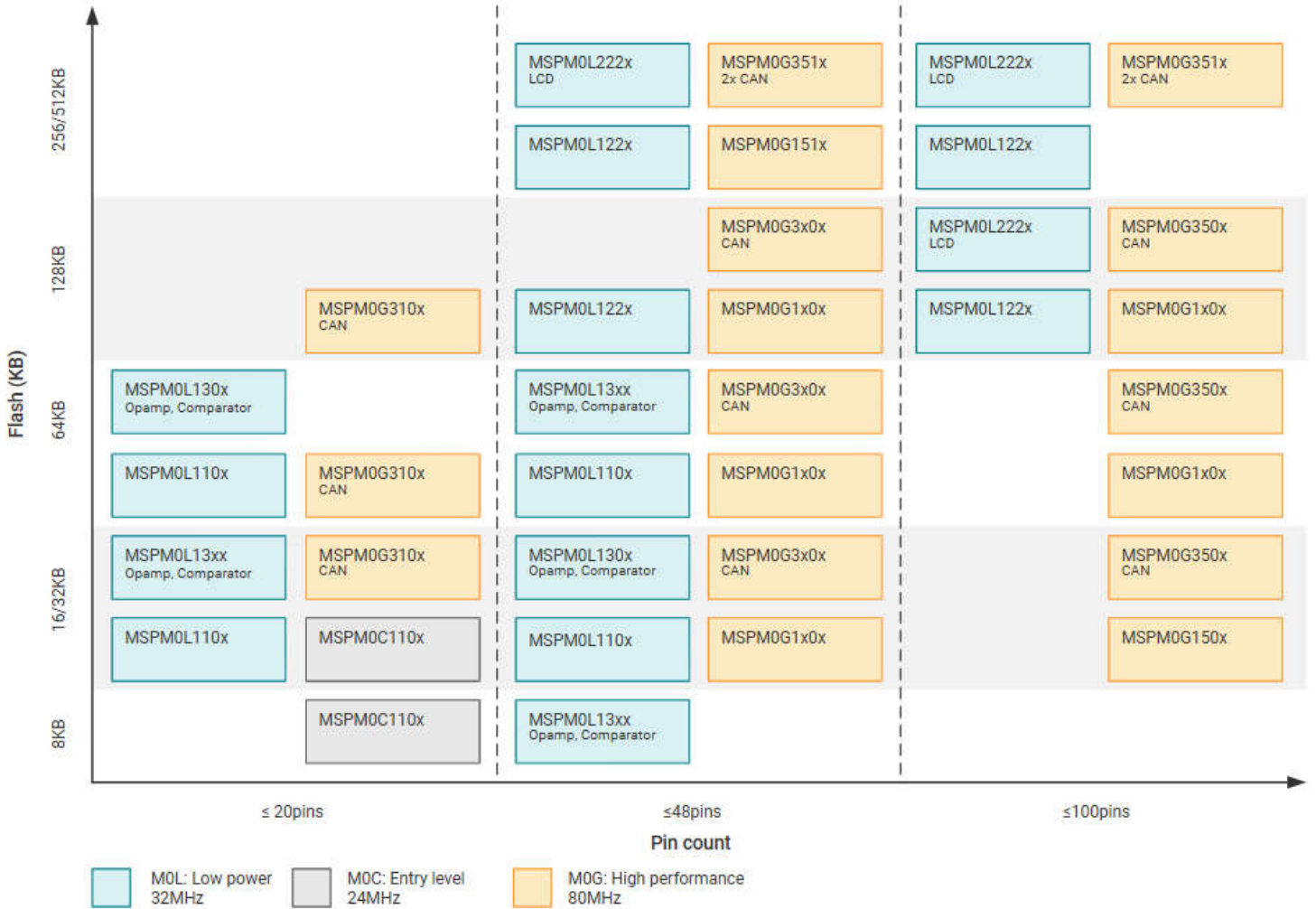


图 2-10. MSPM0C、MSPM0L 和 MSPM0G 产品系列

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，用户可以根据 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，用户可以直接使用筛选工具进行选择，右侧将弹出合格的 MCU 器件，如图 2-11 中所示。用户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。

Hide filters Columns Reset table 23 of 23 total products Email Download Excel Log in to view invento

Product number	Images	Price/Quantity (USD)	Frequency (MHz)	Flash memory (kByte)	RAM (kByte)	Number of GPIOs	UART	Number of I2Cs	SPI
<input type="checkbox"/> MSPM0L1345 - NEW Data sheet: PDF HTML View alternates		US\$0.484 1ku	32	32	4	22	2	2	1
<input type="checkbox"/> MSPM0L1346 - NEW Data sheet: PDF HTML		US\$0.544 1ku	32	64	4	22	2	2	1
<input type="checkbox"/> MSPM0L1343 - NEW Data sheet: PDF HTML View alternates		US\$0.412 1ku	32	8	2	15	2	2	1
<input type="checkbox"/> MSPM0L1344 - NEW Data sheet: PDF HTML View alternates		US\$0.422 1ku	32	16	2	15	2	2	1

图 2-11. 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 2-12 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。

NEW
MSPM0L1345 ACTIVE

32-MHz Arm® Cortex®-M0+ MCU with 32-KB flash, 4-KB SRAM, 12-bit ADC, comparator, TIA

DATA SHEET [MSPM0L130x Mixed-Signal Microcontrollers datasheet \(Rev. C\)](#) PDF | HTML **datasheet**

USER GUIDES [MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual \(Rev. C\)](#) **User Guides**

ERRATA [MSPM0L110x, MSPM0L13xx Microcontrollers Errata \(Rev. A\)](#) **errata**

图 2-12. MSPM0 重要文档列表

如图 2-13 所示，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。

Technical documentation

★ = Top documentation for this product selected by TI

Type	Title	Date
★ Data sheet	MSPM0L130x Mixed-Signal Microcontrollers datasheet (Rev. C)	27 Jun 2023
★ Errata	MSPM0L110x, MSPM0L13xx Microcontrollers Errata (Rev. A)	28 Apr 2023
★ User guide	MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual (Rev. C)	05 May 2023
Application note	MSPM0 Bootloader (BSL) Host Implementation (Rev. A)	06 Jul 2023
Application note	EEPROM Emulation Type A Solution	18 Apr 2023
Application note	STM32에서 Arm 기반 MSPM0으로의 마이그레이션 가이드 (Rev. A)	12 Apr 2023
Application note	從 STM32 到 Arm 架構的 MSPM0 移植指南 (Rev. A)	12 Apr 2023
Application note	EEPROM Emulation Type B Design	11 Apr 2023

图 2-13. 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 2-14 所示。

MSPM0L1306 PREVIEW Data sheet Order now

Product details | Technical documentation | Design & development | **Ordering & quality** | Support & training

Ordering & quality

Part number	Buy	TI.com inventory	Qty Price (USD)	Package qty Carrier
XMSM0L1306SDGS20R ACTIVE	<input type="text" value="Enter quantity"/> Add to cart Limit: 5	93	1ku	1 LARGE T&R
XMSM0L1306SDGS28R ACTIVE	<input type="text" value="Enter quantity"/> Add to cart Limit: 10	170	1ku	5,000 LARGE T&R

图 2-14. 器件订购和质量视图

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[链接](#)进行下载，然后开始安装。在安装过程中，继续按下一步。

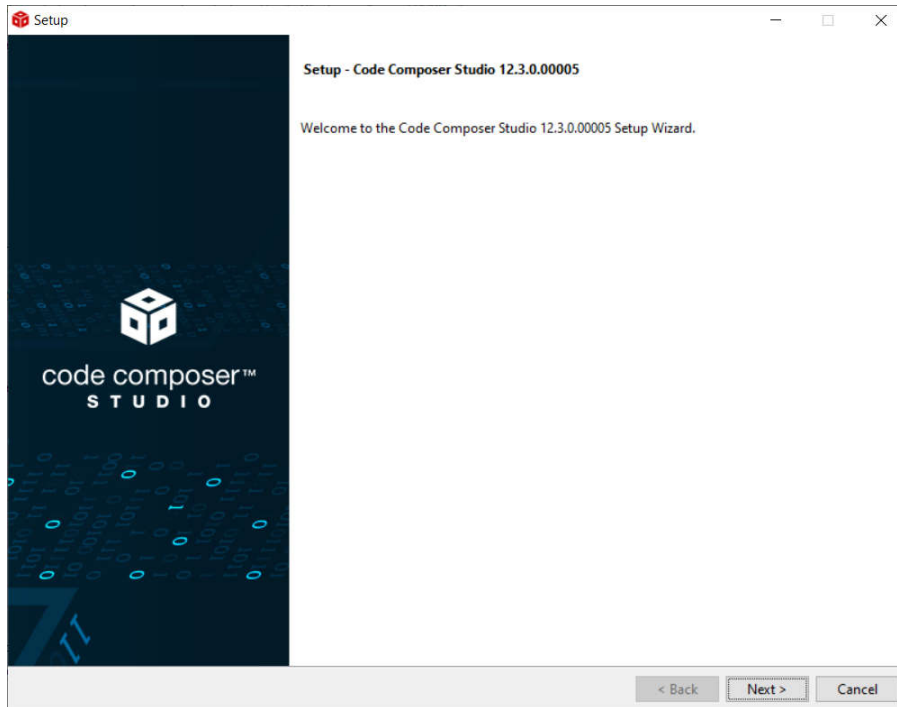


图 2-15. CCS 安装

2. 选择 MSPM0 支持元件。

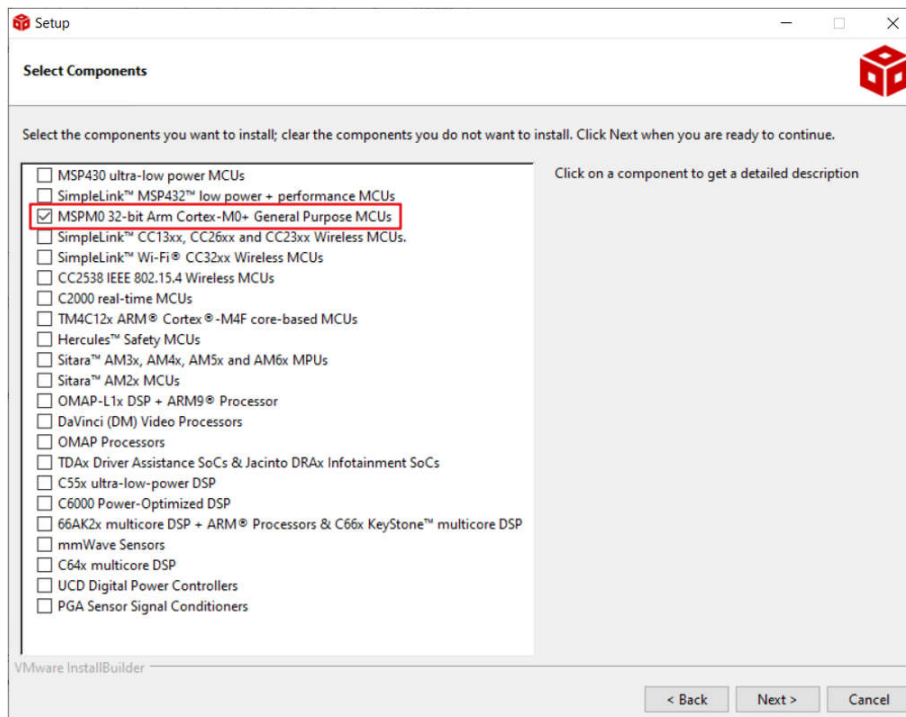


图 2-16. 安装 CCS - 选择 MSPM0 支持元件

3. 如果需要，选择 J-link。

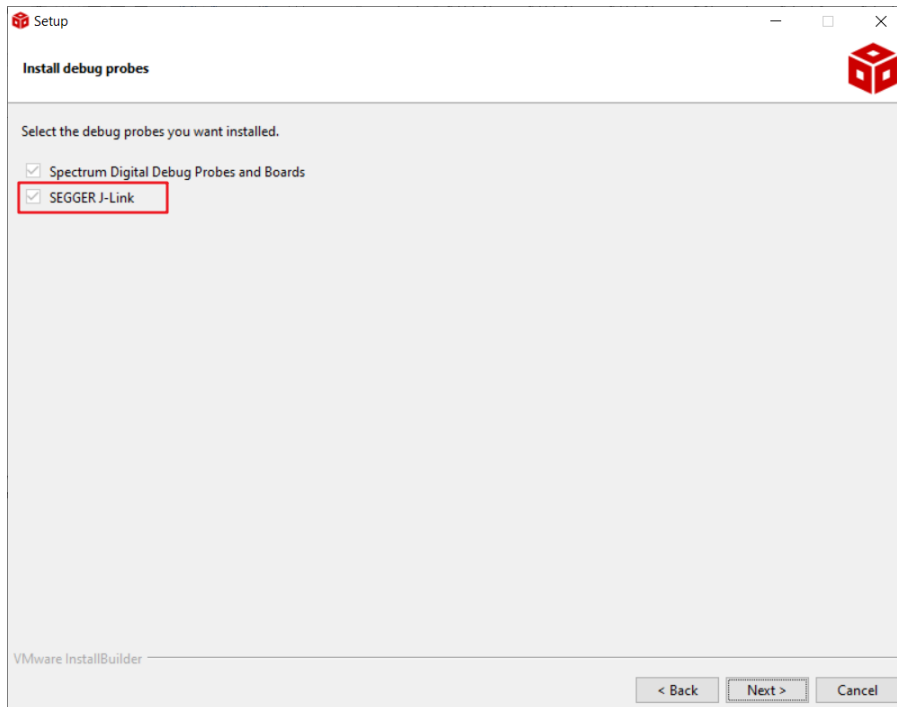


图 2-17. 安装 CCS - 选择下载 J-Link

4. 完成 CCS 下载。

2.2.2.2 CCS 简介

1. 启动一个新的工作区。工作区是指复制已导入的工程地址。

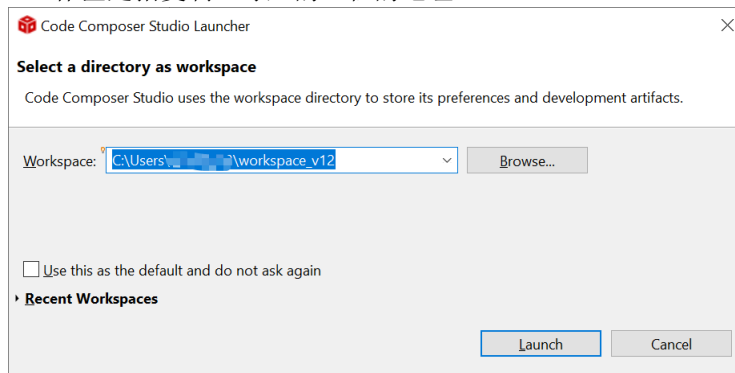


图 2-18. CCS 启动工作区

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 2-19 所示。然后，添加工程名称并按下完成按钮后即可创建程序。TI 建议查找 **MSPM0 SDK 示例**，该示例介绍了如何使用节 2.2.4 中的 **CCS**。

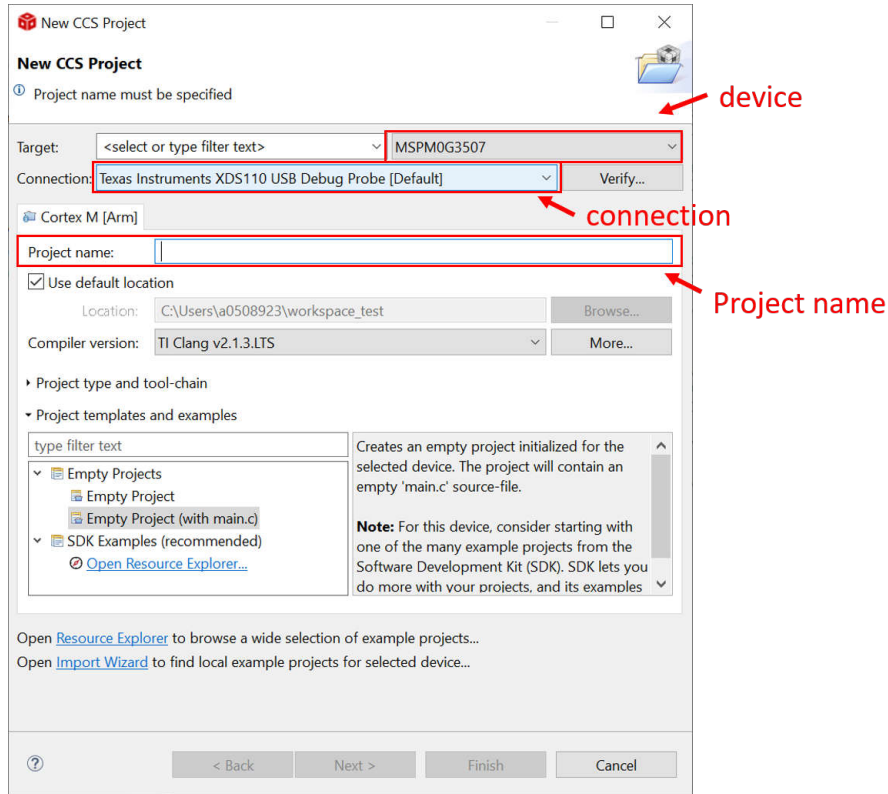


图 2-19. 在 CCS 中创建新工程

3. 图 2-20 和图 2-21 简要介绍了 CCS 函数。

快捷键功能：

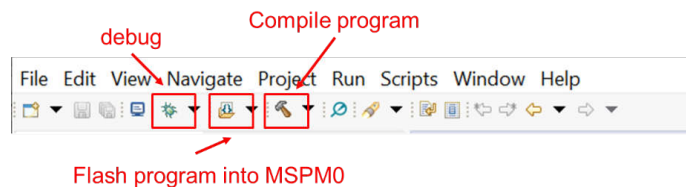


图 2-20. 常用功能

调试功能：

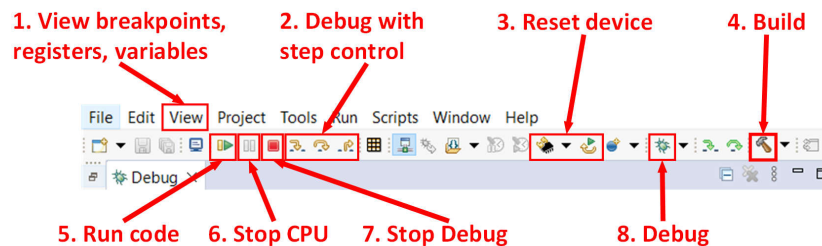


图 2-21. 常用调试功能

工程属性常用设置：

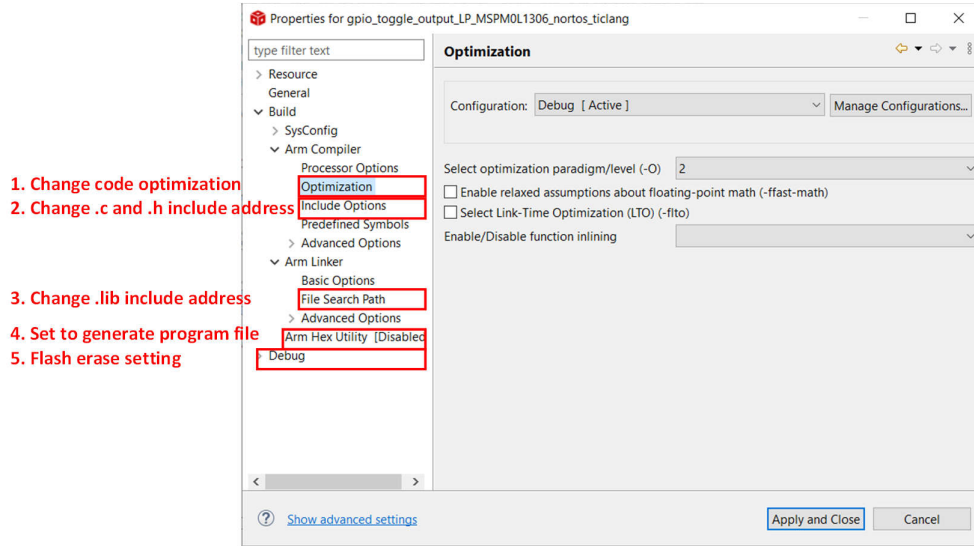


图 2-22. 常用工程设置

有关详细信息、请参阅适用于 MSPM0 MCU 的 Code Composer Studio IDE 版本 12.4+。

2.2.3 步骤 3. 设置 MSPM0 SDK 和 MSPM0 SDK 简介

STM8 提供了一个“标准外设库”，使开发人员能够轻松利用 STM8 微控制器的所有功能来覆盖多种应用。TI 也提供了类似的支持。

2.2.3.1 设置 MSPM0 SDK

1. 点击链接下载 MSPM0 SDK。



图 2-23. 下载 MSPM0 SDK

2. 按下一步安装 SDK。

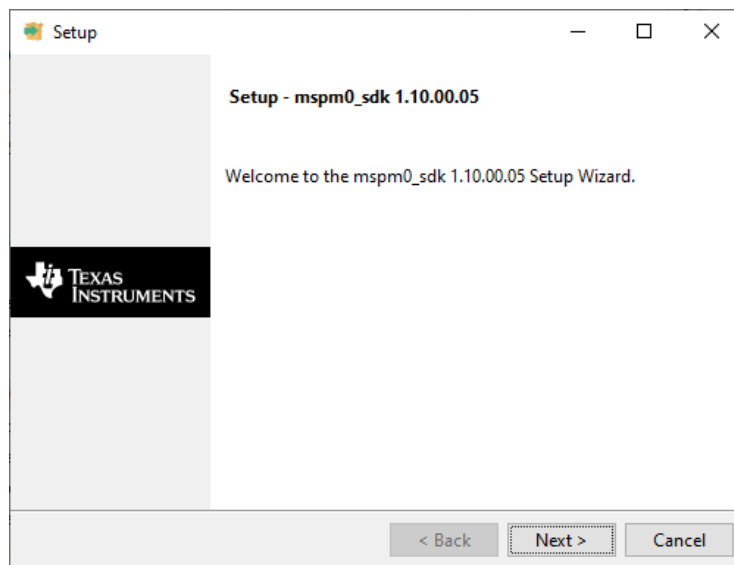


图 2-24. 安装 MSPM0 SDK

3. 完成 MSPM0 SDK 下载。

2.2.3.2 SDK 简介

下载完成后，文件内容将显示在 SDK 文件夹中，默认为 `c:/ti/mspm0_sdk_xxx`，如图 2-25 所示。最常用的文件夹是示例文件夹和 docs 文件夹。

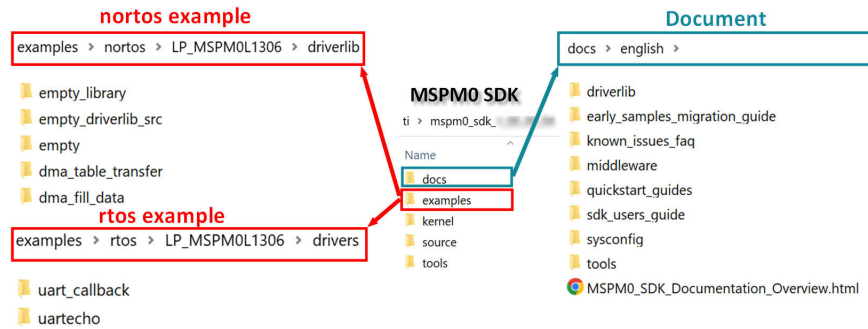


图 2-25. MSPM0 SDK 文件夹

表 2-5 汇总了示例覆盖。

表 2-5. 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一个工程。在表 2-6 中显示了这些差异。

表 2-6. 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	是
empty_driverlib_src (建议)	是	是

对于 docs 文件夹，结构和重要文档如图 2-26 所示。

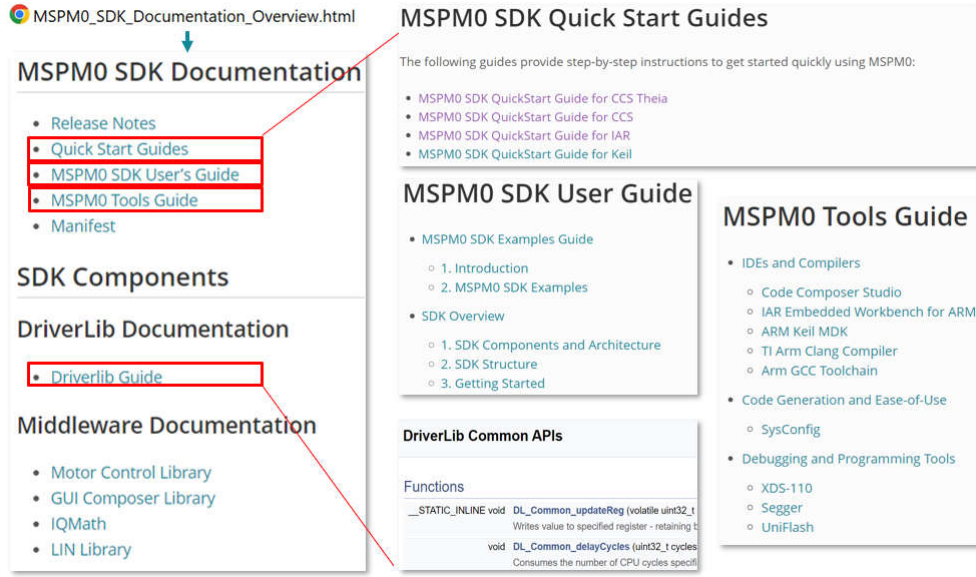


图 2-26. 文档概览

2.2.4 第 4 步：软件评估

以下是将示例移植到 CCS 的一些步骤。

1. 选择 **Project**，然后从菜单中导入 CCS Projects。

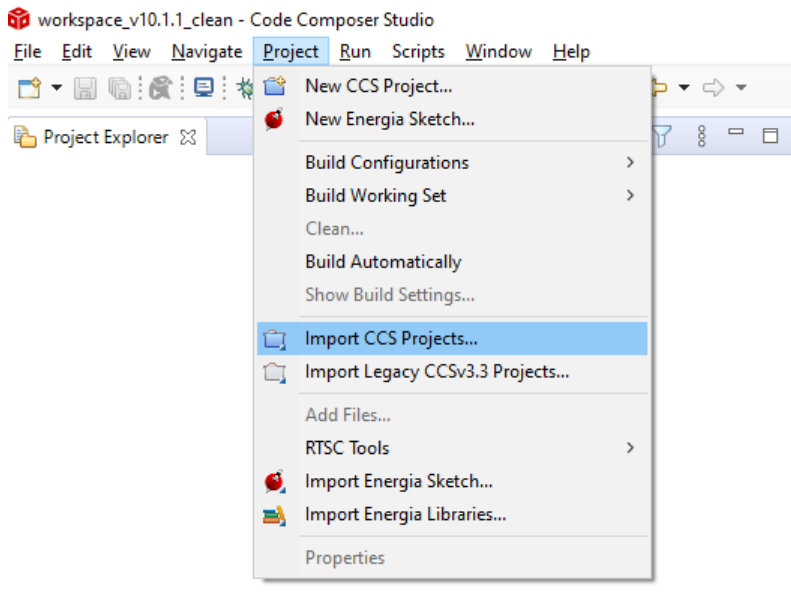


图 2-27. 导入 CCS 工程

2. 从 SDK 中选择程序。以 MSPM0L1306 为例。

\\mspm0_sdk_1_10_00_05\examples\nortos\LP_MSPM0L1306\driverlib

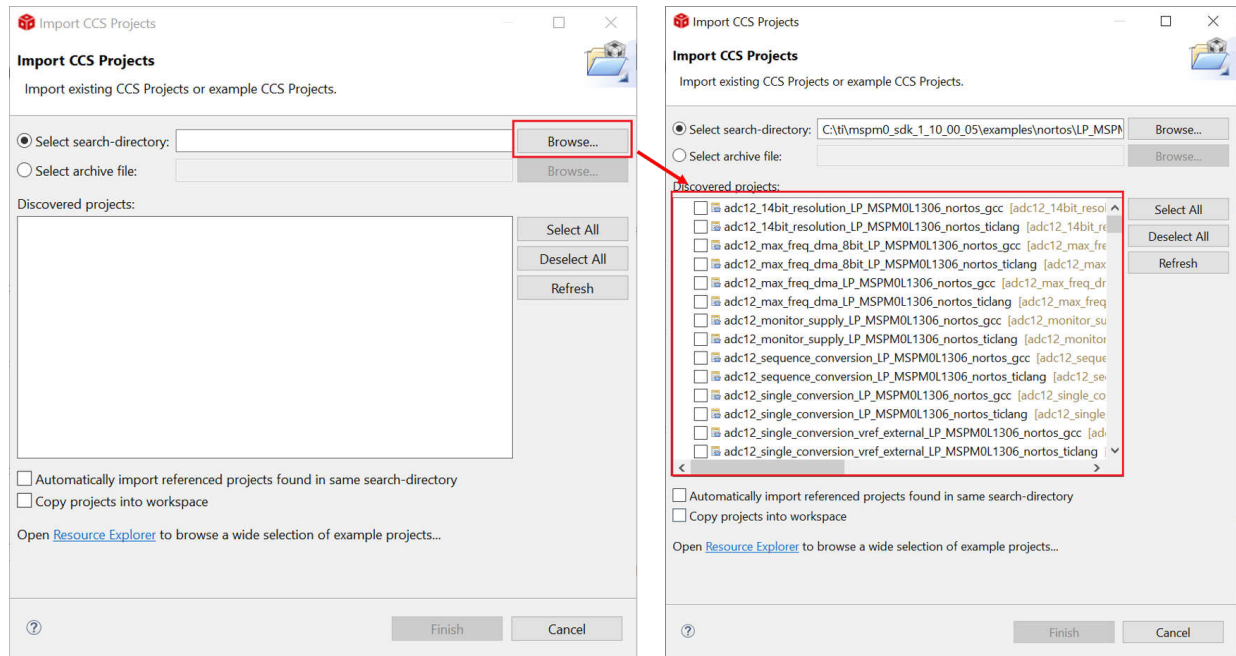


图 2-28. 从 SDK 中选择程序

如果无法导入工程，请删除工作区中的同名工程。

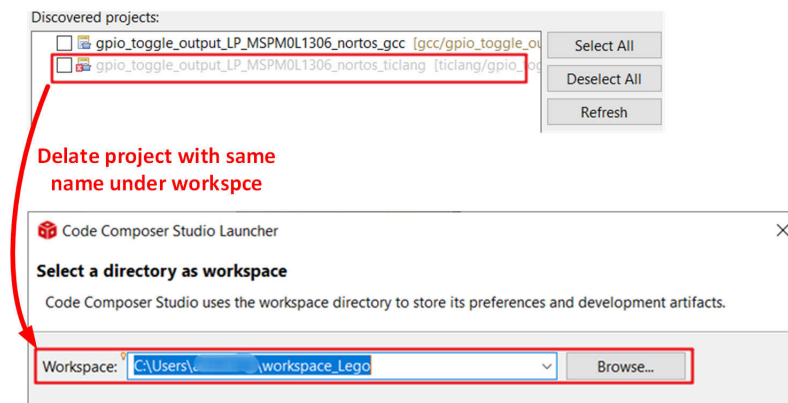


图 2-29. 删除重复的工程

3. 导入后，左侧将出现一个工程，并自动打开一个 README.md 文件。TI 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。

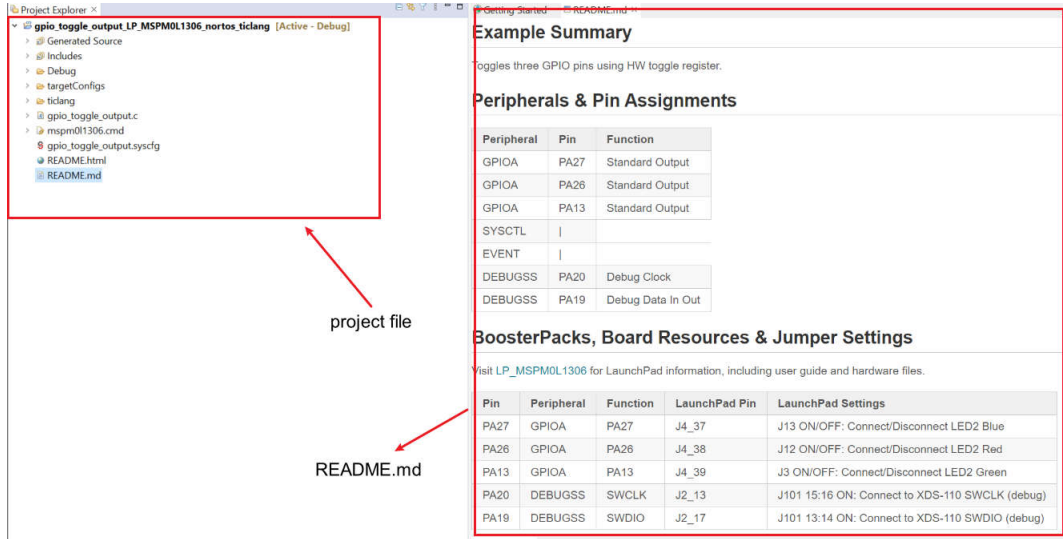


图 2-30. 工程和 README.md

4. 图 2-31 显示了工程中最重要文件。



图 2-31. CCS 工程概述

5. 就像 STM 开发一样，TI 也支持芯片视图。双击 .syscfg 文件，可以进入 SysConfig，在这里用户可以通过图形界面配置所需的外设。此外，TI 建议先使用 SysConfig 的 MCU 视图来帮助用户与软件工程师一起修复引脚功能，这类似于 STM8CubeMAX 中的 MCU/MPU 封装。
6. 根据代码和 SysConfig 示例，用户可以根据 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. 如果用户要添加第三方库，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 2-32 所示。

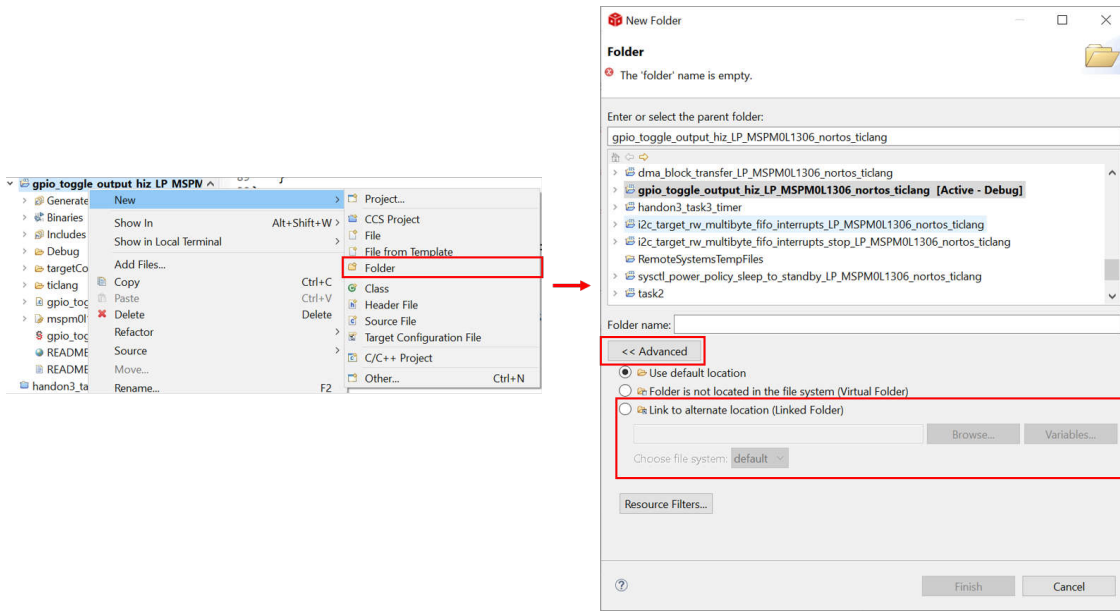


图 2-32. 添加相关文件

然后，需要执行其他步骤以通知编译器添加标头文件。

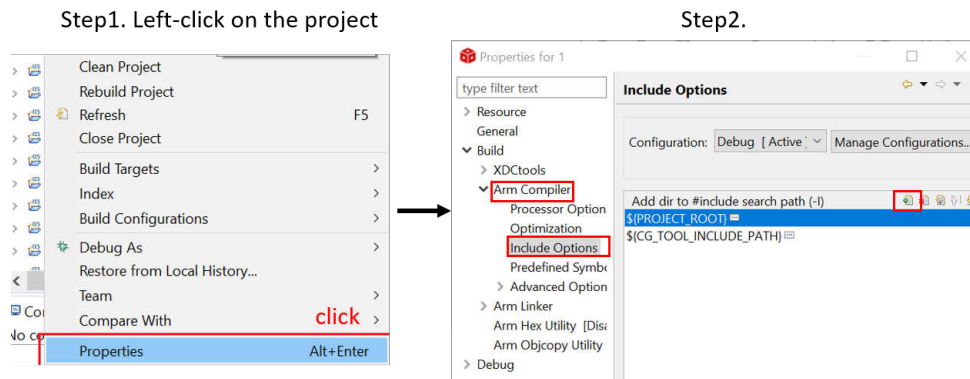


图 2-33. 设置“Include Options”

8. 用户完成软件评估后，请点击主工具栏中的 *build* 图标，如图 2-34 所示。*Build Finished* 外观表示编译成功。

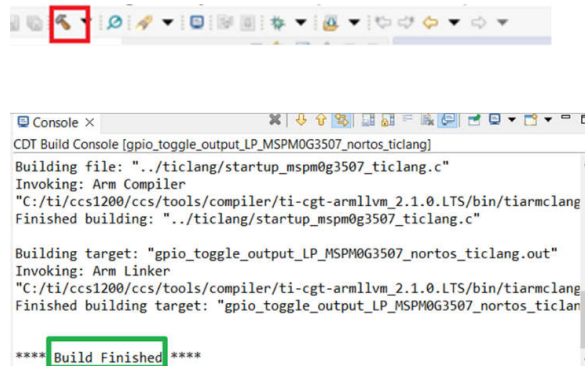


图 2-34. 成功构建

2.2.5 步骤 5.PCB 板设计

1. 要获取设计包，请访问 [Ti.com](https://www.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。

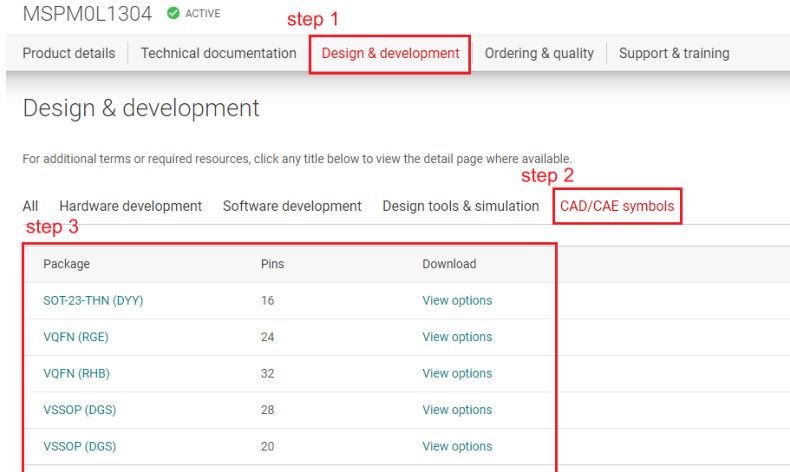


图 2-35. Ultra Librarian 工具入口

3. 对于 MSPM0，可将硬件设计到您自己的电路板中。图 2-36 所示为最小系统设计示例。

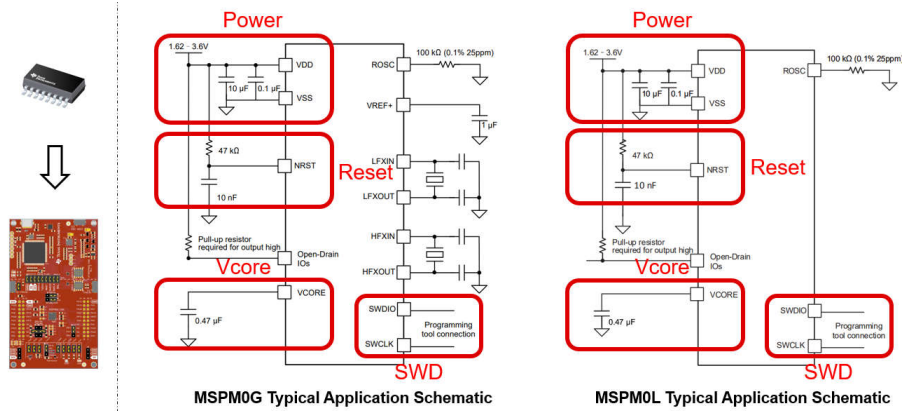


图 2-36. MSPM0 最小系统

对于最小系统，需要注意以下几点：

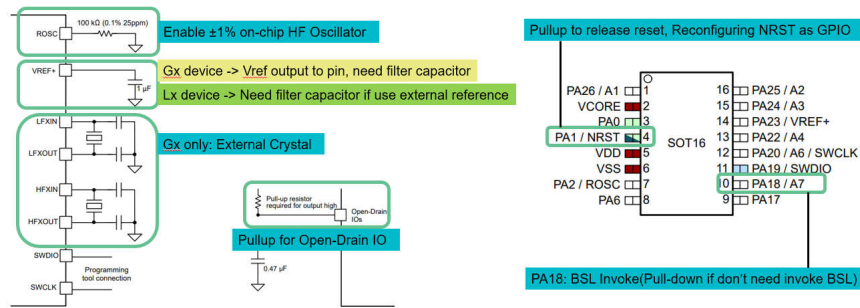


图 2-37. MSPM0 最小系统注意事项

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/...)。

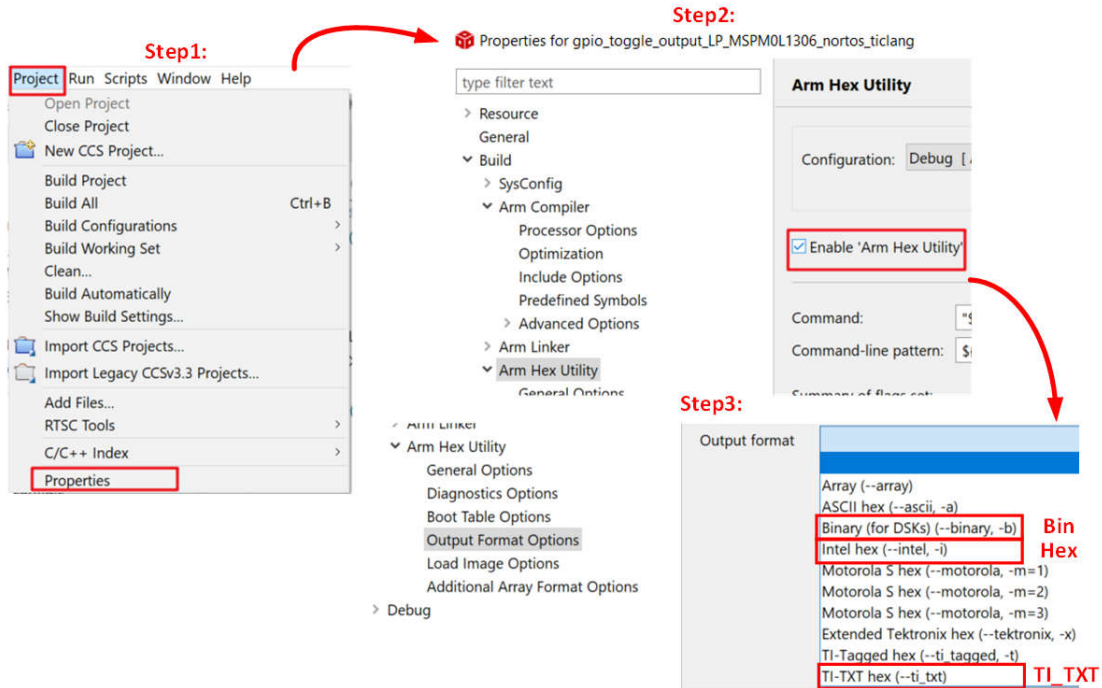


图 2-38. 创建程序文件

2. 选择编程器/调试器来对 MSP 器件进行编程。

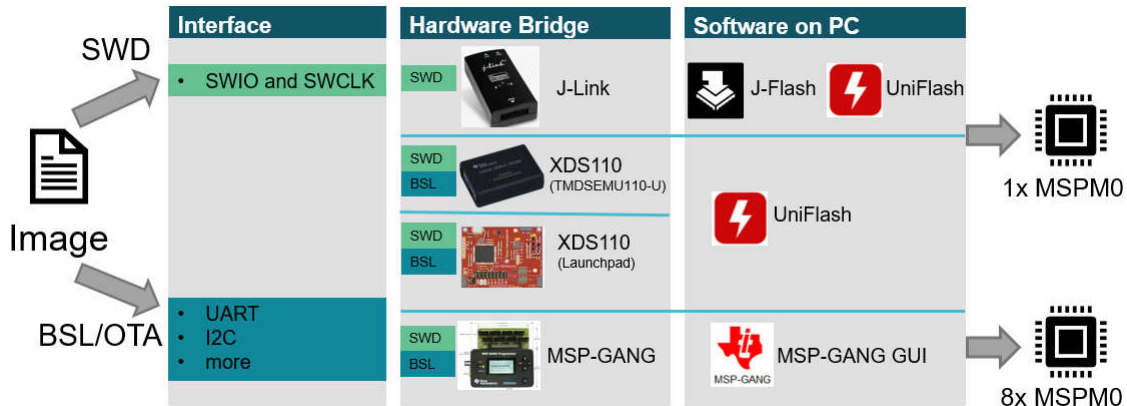


图 2-39. 程序软件和工具

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 STM8 和 MSPM0 之间的计时器模块差异，并在 MSPM0 的 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。

```

18 *   from this software without specific prior written permission.
19 *
20 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
21 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
22 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
23 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
24 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
25 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
26 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
27 * OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
28 * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
29 * OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
30 * EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
31 */
32
33 #include "ti_msp_dl_config.h"
34
35 int main(void)
36 {
37     SYSCFG_DL_init();
38     DL_TimerG_startCounter(PWM_0_INST);
39
40     while (1) {
41         __WFI();
42     }
43 }
44
45

```

图 2-40. 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 2-41 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。

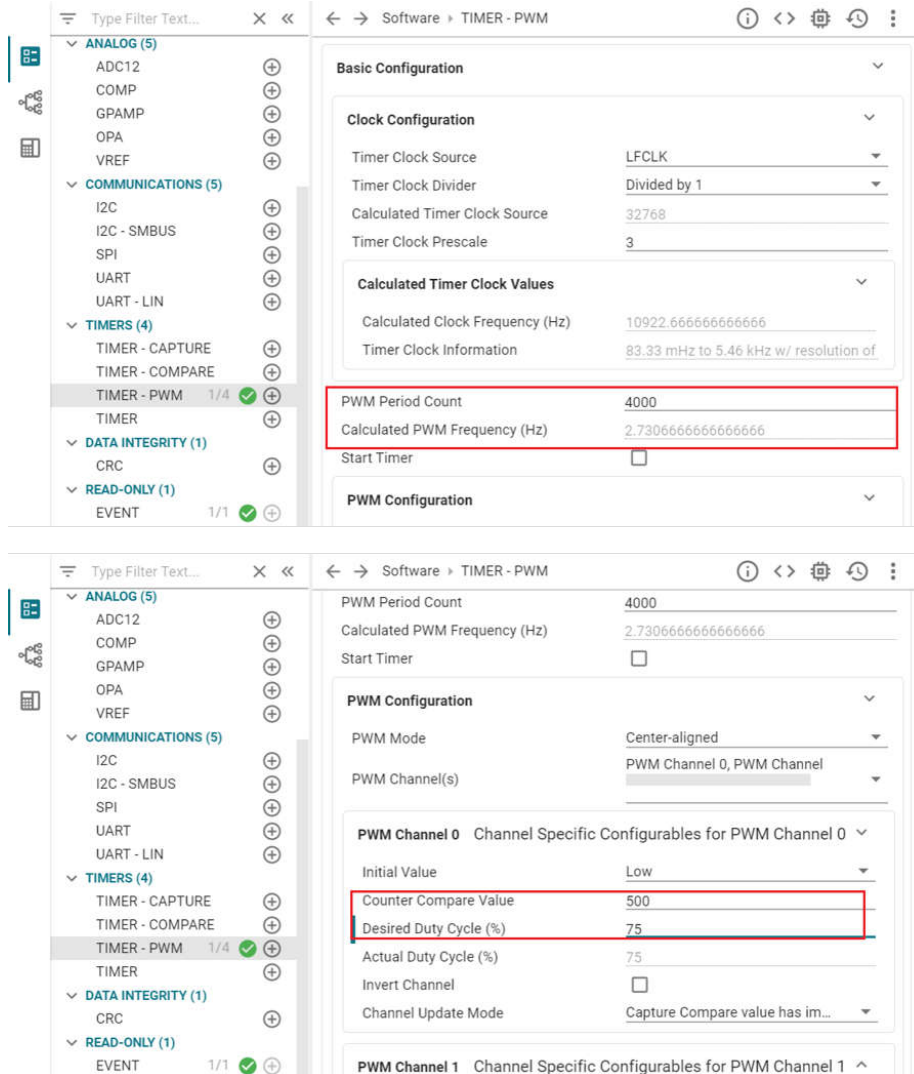


图 2-41. SysConfig 中的 PWM 配置

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。

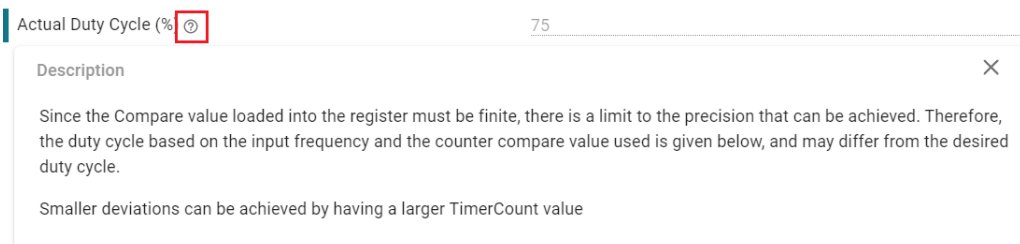


图 2-42. 获取每一项的详细信息

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。

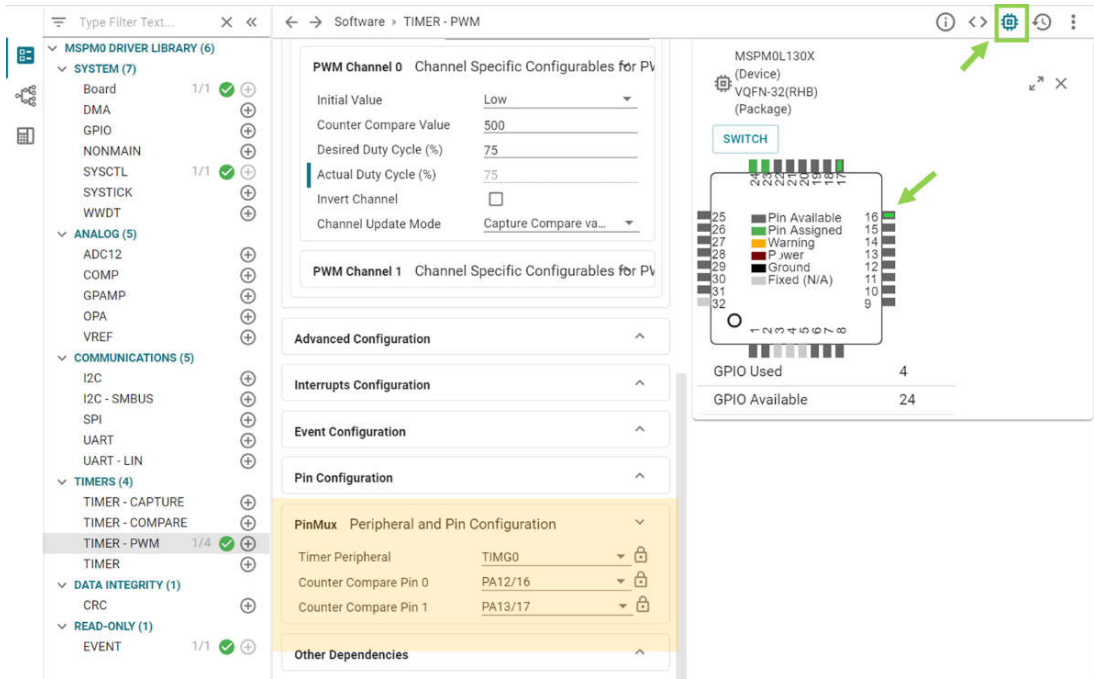


图 2-43. 引脚配置

保存并重建工程后，SysConfig 会更新图 2-44 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。

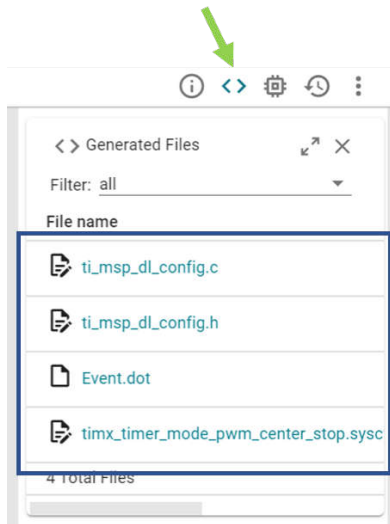


图 2-44. SysConfig 更新的文件

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。

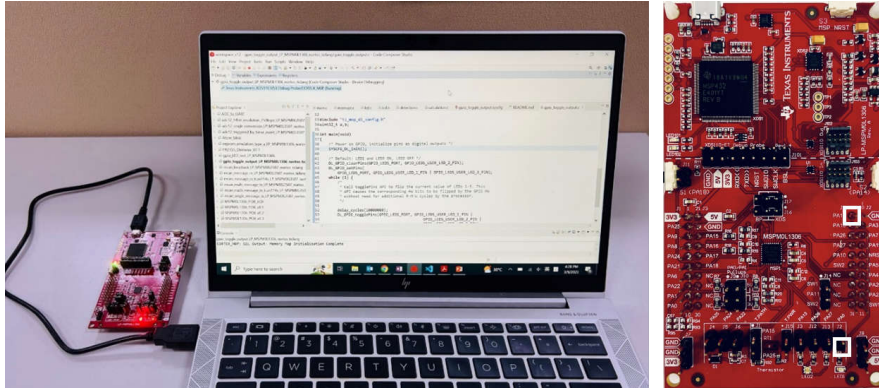


图 2-45. 硬件设置

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。

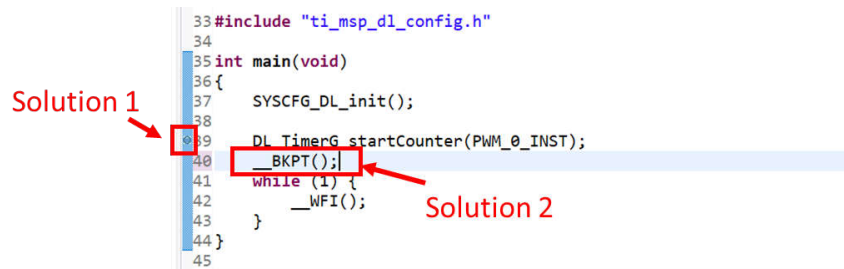


图 2-46. 添加断点解决方案

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 2-47 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

MSPM0L1306 ✔ ACTIVE

[Product details](#) |
 [Technical documentation](#) |
 [Design & development](#) |
 [Ordering & quality](#) |
 [Support & training](#)

Design & development

For additional terms or required resources, click any title below to view the detail page where available.

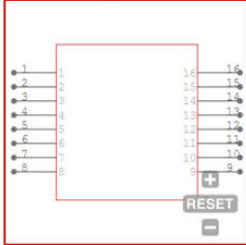
[All](#) |
 [Hardware development](#) |
 [Software development](#) |
 [Design tools & simulation](#) |
 [CAD/CAE symbols](#)

Package	Pins	Download
SOT-23-THN (DYY)	16	View options
VQFN (RGE)	24	View options
VQFN (RHB)	32	View options
VSSOP (DGS)	20	View options
VSSOP (DGS)	28	View options

step2

Ultra Librarian | Texas Instruments - XMSM0L1306SDYYR | English

Symbol



Normal View

SOT_06SDYYR 1

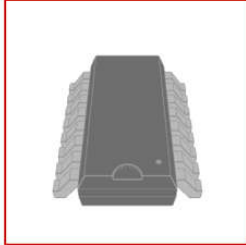
Footprint



Basic View

SOT_06SDYYR_TEX

3D Model



Choose CAD Formats & Download

step3


Ultra Librarian | Texas Instruments - XMSM0L1306SDYYR | English

Choose CAD Format(s) Return to Previews

<p>3D CAD Model</p> <p>Altium</p> <p><input checked="" type="checkbox"/> Altium Designer</p> <p><input type="checkbox"/> PCAD v14</p> <p><input type="checkbox"/> PCAD v15</p> <p>Autodesk</p> <p>Cadence</p> <p>DesignSpark</p> <p>KiCAD</p>	<p>Mentor</p> <p>Pulsonix</p> <p>Quadcept</p> <p>TARGET 30011</p> <p>Zuken</p>
--	--

Symbol Pin Ordering: Sequential
 Footprint Units: English (mil)

I have read and agree to the Ultra Librarian Terms And Conditions

进行人机身份验证 

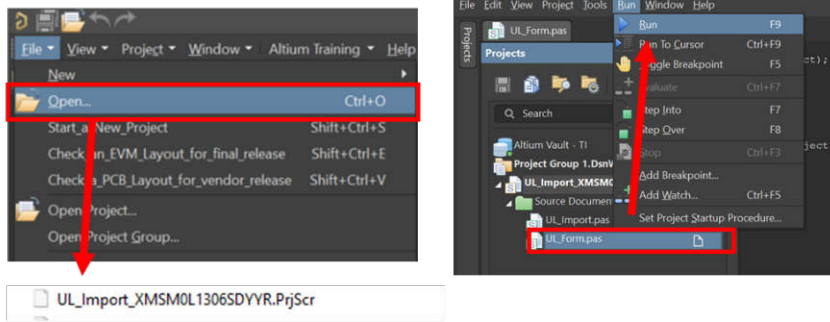
Submit

图 2-47. 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 2-48 所示。

Step 1. Open .PjScr file

Step 2. run UL_Form.pas



Step 3. import file

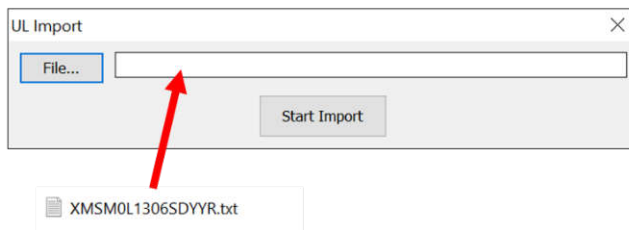


图 2-48. 运行 Altium Designer 脚本

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。

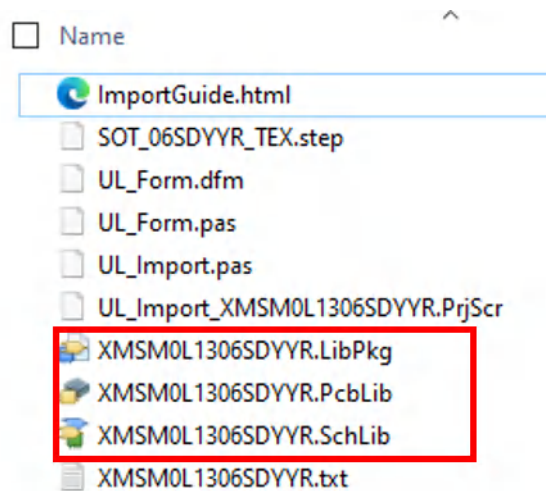


图 2-49. PCB 库和原理图文件

最后一步是将它们导入到您的 AD 库中，如图 2-50 所示。在此基础上，可以设计原理图和 PCB。

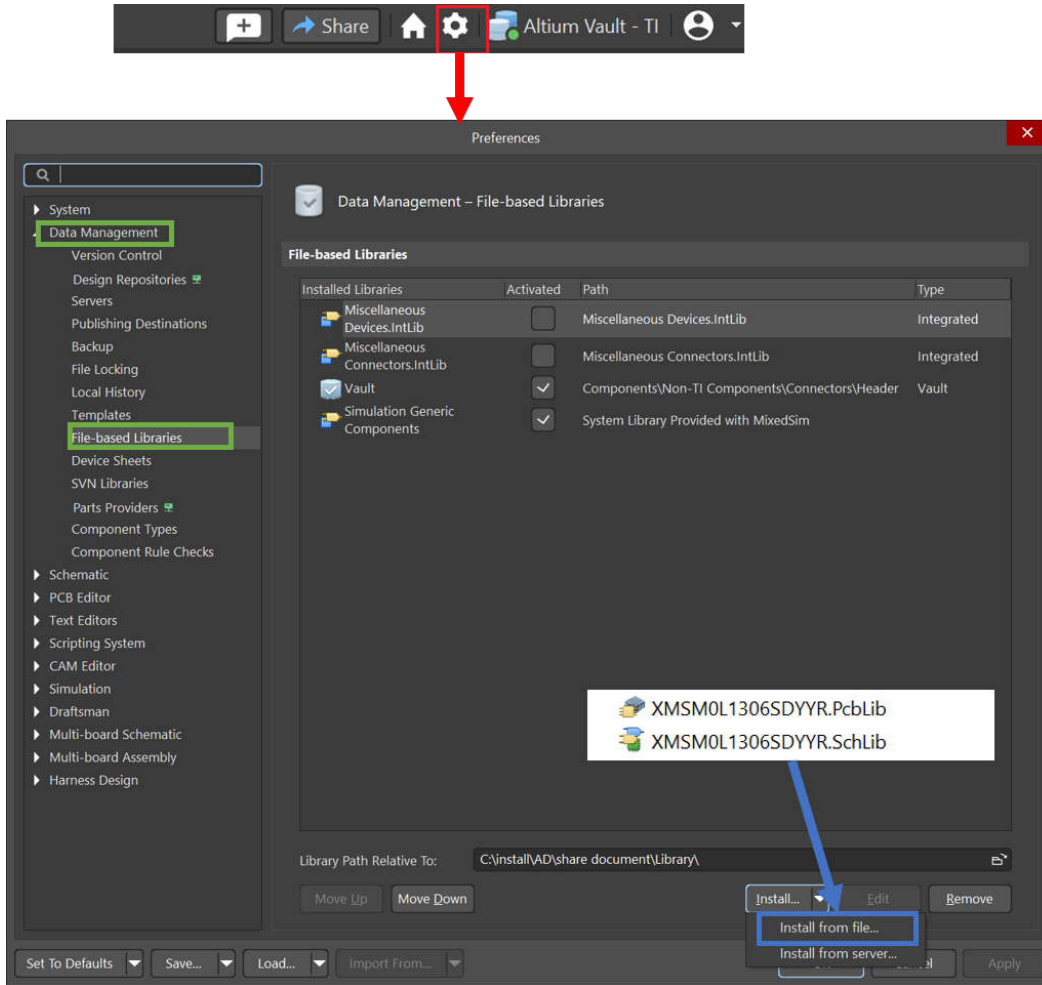


图 2-50. 导入库

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。STM8 系列基于 STM8 CPU 内核架构。表 3-1 概要展示了 MSPM0 系列 CPU 与 STM8 的一般功能比较。

表 3-1. CPU 功能集比较

特性	STM8L 和 STM8S	MSPM0C、MSPM0L 和 MSPM0H
架构	增强型 STM8 CPU 内核	Arm Cortex M0+
数据总线宽度	8 位	32 位
指令集	复杂指令集	精简指令集
指令数量	80	56
乘法指令	MUL (8 × 8)	MULS (32 × 32)
除法指令	DIV (16 × 8)、DIVW (16 × 16)	MATHACL 支持 32 位除法 ⁽¹⁾
流水线	3 级	2 级
工作频率 (最大值)	16MHz 或 24MHz ⁽²⁾	24MHz 或 32MHz ⁽³⁾
DMA	是	是
Coremark/MHz	不可用 ⁽⁴⁾	2.39 ⁽⁵⁾

- (1) MSPM0Gxx 系列具有可提高 32 位除法速度的数学加速器 (MATHACL)。
- (2) STM8Lxx 的最大工作频率为 16MHz，STM8Sxx 的最大工作频率为 24MHz。
- (3) MSPM0Cxx 的最大工作频率为 24MHz，MSPM0Lxx 的最大工作频率为 32MHz。
- (4) STM8 的 coremark 在 st.com 和 eembc.com 上未提供。
- (5) 砂心记号分数由 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 闪存和 EEPROM 特性

MSPM0 和 STM8 系列 MCU 具有非易失性闪存，EEPROM 存储器用于存储可执行程序代码和应用数据。表 3-2 显示了闪存和 EEPROM 的特性。请注意，并非所有器件都具有所有功能。有关更多详细信息，请参阅器件特定的数据表。

表 3-2. 闪存和 EEPROM 的特性

特性	STM8L 和 STM8S	MSPM0L、MSPM0C 和 MSPM0H	
闪存	STM8Lxx 范围 2KB 至 64KB STM8Sxx 范围 4KB 至 128KB	MSPM0Lxx 范围 8KB 至 64KB MSPM0Cxx 范围 8KB 或 64KB MSPM0Hxx 32KB 至 64KB	
EEPROM	高达 2 KB	使用闪存的 EEPROM 仿真	
存储器组织	块大小 (64B/128B) 页大小 (一组块)	字线 (128B) 扇区大小 (1KB) 存储体大小 (可变) : 器件高达 256KB-1bank	
等待状态	0 ($f_{CPU} < 16\text{MHz}$) 1 ($f_{CPU} > 16\text{MHz}$)	0 (MCLK、CPUCLK < 24MHz) 1 (MCLK、CPUCLK < 48MHz)	
单字大小	32 位	64 位 (使用 ECC 时为 72 位)	
编程模式	字节, 单闪存字, 块	分辨率	单闪存字、32 位、16 位或 8 位
		多字	2、4 或 8 个字 (最多 64 个字节)
擦除	块擦除	扇区擦除 存储体擦除 (最大 256KB)	
错误码纠正	支持	支持	
写保护	是	是, 静态和动态	
读保护	是	是	

表 3-2. 闪存和 EEPROM 的特性 (续)

特性	STM8L 和 STM8S		MSPM0L、MSPM0C 和 MSPM0H
擦除和写入周期	程序存储器	100	100K (下部 32KB) 或 10k (上部 32KB)
	数据存储器	100k	

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。
- 互编程电压生成。

3.2.2 闪存和 EEPROM 组织

闪存和 EEPROM 存储器进一步映射到一个或多个逻辑存储器区域，并分配系统地址空间以供应用使用。

3.2.2.1 闪存和 EEPROM 区域

表 3-3 显示了 STM8 器件和 MSPM0 器件的闪存和 EEPROM 区域。

表 3-3. 闪存和 EEPROM 区域

STM8L 和 STM8S		MSPM0L、MSPM0C 和 MSPM0H	
主程序区域	应用代码。	MAIN	应用代码和数据。
选项字节	器件硬件特性和存储器保护的配置。	NONMAIN	BCR 配置。
专有代码区域(PCODE) ⁽¹⁾	保护用于驱动外设的专有软件库。		BSL 配置。
用户引导区域(UBC) ⁽²⁾	复位和中断向量。	FACTORY	器件 ID 和其他参数。
数据 EEPROM	应用数据。	DATA ⁽³⁾	数据或 EEPROM 仿真。

(1) PCODE 仅在低、中+ 和高密度器件中可用。

(2) STM8Sx03xx、STM8S001xx、STM8L101xx 和 STM8L001xx 没有嵌入式引导加载程序（微控制器内部未实现 ROM 引导加载程序）。使用这些器件时，用户必须编写自己的引导加载程序代码并将自己的引导加载程序保存在 UBC 程序区域。

(3) 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）。

3.2.3 嵌入式 SRAM

MSPM0 和 STM8 系列 MCU 具有用于存储应用数据的 SRAM。表 3-4 显示了 SRAM 特性的比较。相关详细信息，请参阅器件特定数据表。

表 3-4. SRAM 功能比较

特性	STM8L 和 STM8S	MSPM0L、MSPM0C 和 MSPM0H
SRAM 存储器	STM8L 超值系列：1KB 至 4KB STM8L101：1.5KB STM8Sxx：1KB 至 6KB	MSPM0Lxx：2KB 至 4KB MSPM0Cxx：1KB 或 8KB MSPM0Hxx：8KB
奇偶效验检查	不支持	MSPM0Lxx：受支持 ⁽¹⁾ ，MSPM0Cxx：不受支持， MSPM0Hxx：不受支持
ECC	不支持	MSPM0Lxx：受支持 ⁽²⁾ ，MSPM0Cxx：不受支持， MSPM0Hxx：不受支持
写保护 (RAM 防护)	不支持	支持

(1) 仅支持 MSPM0Lx22x 和 MSPM0L111x。

(2) 仅支持 MSPM0Lx22x。

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 1KB 的分辨率对低 32KB SRAM 进行动态写保护。在 SRAM 小于 32KB 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

STM8 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

STM8		MSPM0	
上电复位 (POR)	上升检测： $V_{DD} > V_{POR}$ ，POR 状态释放，BOR 开始工作。	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
断电复位 (PDR)	下降检测： $V_{DD} < V_{PDR}$ ，PDR 使器件保持复位状态。		
欠压复位 (BOR) ⁽¹⁾	上升检测： $V_{DD} > V_{BOR+}$ ，BOR 状态被释放，器件继续执行引导过程。	欠压复位 (BOR) - 0 级 ⁽²⁾	上升检测： $V_{DD} > BOR+$ ，器件继续执行引导过程，并启动 PMU 下降检测： $V_{DD} < BOR-$ ，器件保持 BOR 状态
	下降检测： $V_{DD} < V_{BOR-}$ ，产生 BOR 状态。 BOR 有五个不同的级别可供选择。	欠压复位 (BOR) - 1 至 3 级 ⁽²⁾	下降检测： 1) $V_{DD} < BOR-x$ (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
可编程电压检测器 (PVD) ⁽³⁾	上升检测： $V_{DD} > V_{PVD}$ ，生成 PVD 事件。 下降检测： $V_{DD} < V_{PVD}$ ，生成 PVD 事件。PVD 有 7 个不同的级别可供选择。	不适用	不适用

表 3-5. 上电总结和比较 (续)

STM8		MSPM0	
RTC 复位	RTC 及相关寄存器通过系统复位或上电复位进行复位。	RTC 复位	RTC 和相关时钟通过 BOOTRST、BOR 或 POR 复位

- (1) 启动期间，BOR 需要配置为 BOR 电平 0，以确保最低工作电压。BOR 在上电时始终处于活动状态，保持 MCU 处于复位状态，直到达到应用运行阈值。如果 BOR 在断电时被禁用，复位阈值为 V_{PDR} 以确保 VDD 最小值。
- (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。
- (3) STM8L001xx、STM8L101xx 和 STM8S 系列微控制器没有 PVD。

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)。

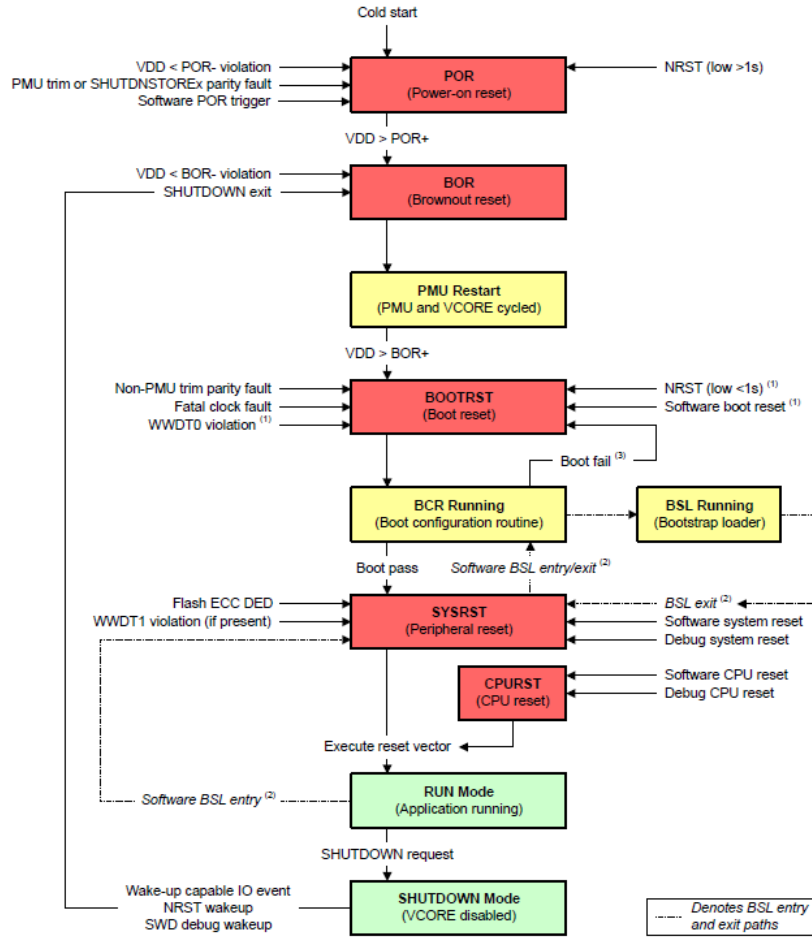


图 3-1. MSP 复位功能

3.4 时钟总结和比较

3.4.1 振荡器

STM8 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 STM8 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。有关更多详细信息，请参阅器件特定的数据表。

表 3-6. 振荡器比较

类型		STM8L	STM8S	MSPM0L	MSPM0C	MSPM0H
内部振荡器	高速	HSI : 内部 16MHz RC 振荡器	HSI : 内部 16MHz RC 振荡器	SYSOSC : 4MHz 至 32MHz 的内部振荡器	SYSOSC : 内部 24/32 MHz 振荡器	SYSOSC : 内部 32MHz 振荡器
	低速	LSI : 内部 38kHz RC 振荡器	LSI : 内部 128kHz RC 振荡器	LFOSC : 内部 32kHz 振荡器	LFOSC : 内部 32kHz 振荡器	LFOSC : 内部 32kHz 振荡器
外部振荡器	高速	HSE : 1-16MHz 外部晶体	HSE : 1-24MHz 外部晶体	4MHz 至 32MHz ⁽¹⁾	4MHz 至 32MHz ⁽¹⁾	不可用
	低速	LSE : 32.768 kHz 外部晶体	不可用	32kHz ⁽¹⁾	32kHz ⁽¹⁾	不可用

(1) 仅支持 MSPM0Lx22x、MSPM0C1105 和 MSPM0C11056。

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 3-7. 时钟信号比较

时钟说明		STM8L 时钟	STM8S 时钟	MSPM0L、MSPM0C 和 MSPM0H 时钟
外部数字时钟输入	高频率	外部源：高达 16MHz ⁽¹⁾	HSE 外部：高达 24MHz ⁽¹⁾	Lx22x 和 C1106：支持 32MHz
	低频率	外部源：32.768kHz ⁽¹⁾	不适用	Lx22x 和 C1106：支持 32KHz
主时钟的高频源		HSI、HSE	f_{HSE} 、 f_{HSIDIV}	SYSOSC
主时钟的低频源		LSI、LSE	f_{LSI}	LFCLK (固定 32kHz)
主系统时钟		SYSCLK、 f_{MASTER}	f_{MASTER}	MCLK，ULPCLK (BUSCLK) ⁽²⁾
源 CPU		SYSCLK、 f_{MASTER}	f_{CPU}	CPUCLK
大多数外设硬件的时钟		PCLK (SYSCLK)、 f_{MASTER}	f_{MASTER}	MCLK，ULPCLK ⁽²⁾
外设专用时钟		BEEPCLK、IWDGCLK、RTCCLK、 f_{LSI} 、 $f_{HSI/2}$ ⁽³⁾	不适用	ADCCLK
固定频率时钟		不适用	不适用	MFCLK : 4MHz，与 MCLK/ULPCLK 同步

- (1) 使用外部时钟源时，需要关闭 HSE 晶体和 LSE 晶体。STM8L001xx 和 STM8L101xx 系列不支持外部数字时钟输入。
- (2) MCLK 是 PD1 的主系统时钟，源自 MCLK 的 ULPCLK 是 PD0 的主系统时钟。PD1 (电源域 1) 包含 CPU 子系统、存储器接口和高速外设。PD0 (电源域 0) 包含低速低功耗外设。
- (3) STM8L001xx 和 STM8L101xx 系列的 f_{LSI} 仅用于为 AWU、BEEP、SWIM、IWDG 提供时钟源。 $f_{HSI/2}$ 只用于为 SWIM 提供时钟源。

表 3-8. 外设时钟源

外设	STM8L, STM8S	MSPM0L, MSPM0C, MSPM0H
UART/USART	SYSCLK, f _{MASTER}	SYSCLK, MFCLK, LFCLK
SPI	SYSCLK, f _{MASTER}	SYSCLK, MFCLK, LFCLK
I2C	SYSCLK, f _{MASTER}	BUSCLK, MFCLK
ADC	PCLK 或 PCLK/2 ⁽¹⁾ , f _{ADC} (f _{MASTER} 除以 2 到 18)	ADCCLK (由 ULPCLK 或 SYSOSC 提供)
计时器	SYSCLK, f _{MASTER} , f _{MASTER} /DIV	BUSCLK, MFCLK, LFCLK
比较器	PCLK, f _{MASTER} ⁽²⁾	BUSCLK
看门狗	LSI, SYSCLK, f _{CPU} ⁽³⁾	LFCLK

(1) STM8L001xx 和 STM8L101xx 微控制器系列没有 ADC。

(2) STM8S 系列微控制器没有比较器。

(3) LSI 用于为独立看门狗 (IWDG) 提供时钟源。SYSCLK 或 f_{CPU} 用于为窗口看门狗 (WWDG) 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 3-9 简要比较了 STM8 和 MSPM0 器件。

表 3-9. STM8 器件和 MSPM0 器件的工作模式比较

STM8		MSPM0	
运行模式	说明	运行模式	说明
运行模式	系统或电源复位后，CPU 和外设正常运行。	RUN	0 MCLK 和 CPUCLK 通过快速时钟源 (SYSOSC) 运行
低功耗运行模式	CPU 和外设使用低速振荡器 (LSI 或 LSE) 运行。所有中断都必须被屏蔽。		1 MCLK 和 CPUCLK 通过 LFCLK (32kHz) 运行。
			2
等待模式	CPU 停止运行。振荡器保持启用状态。选定的外设继续运行。通过执行 WFI 或 WFE 指令从运行模式进入等待模式。	SLEEP	0 CPU 停止运行。SYSOSC 保持启用状态。LFOSC 保持启用状态。MCLK 通过快速时钟源 (SYSOSC) 运行。
			1 CPU 停止运行。SYSOSC 保持启用状态。LFOSC 保持启用状态。MCLK 通过 LFCLK 运行。
			2 CPU 停止运行。SYSOSC 保持禁用状态。LFOSC 保持启用状态。MCLK 通过 LFCLK 运行。
主动停机模式 (STM8S)	CPU 停止运行。除 LSI 或 HSE 之外的振荡器均被禁用。除 AWU 外，几乎所有外设均停止。MVR 稳压器已上电。	STOP ⁽²⁾	0 CPU 停止运行。SYSOSC 的状态被保留 ⁽¹⁾ 。LFOSC 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。
带 MVR 自动断电的主动停机模式(STM8S)	CPU 停止运行。仅 LSI 以外的振荡器被禁用。除 AWU 外，几乎所有外设均停止。MVR 稳压器已断电。		2 CPU 停止运行。SYSOSC 禁用。ULPCLK 以 32kHz 的频率运行。PD0 启用且 PD1 禁用。

表 3-9. STM8 器件和 MSPM0 器件的工作模式比较 (续)

STM8		MSPM0		
运行模式	说明	运行模式	说明	
主动停机模式 (非 STM8S 系列)	CPU 停止运行。除 LSI 或 LSE 之外的振荡器均被禁用。除 RTC、AWU 等外, 几乎所有外设均停止。稳压器处于低功耗模式。	STANDBY	0	CPU 停止运行。SYSOSC 已禁用。所有 PD0 外设均接收 ULPCLK 和 LFCLK。
停机模式	CPU 停止运行。振荡器为禁用状态 ⁽³⁾ 。几乎所有外设均已停止。稳压器处于低功耗模式 ⁽³⁾ 。		1	与 STANDBY0 类似, 仅 TIM0/1 接收 ULPCLK 或 LFCLK。
不适用	不适用	关断	没有可用时钟且器件关断。	

- (1) 如果从 RUN1 进入 STOP0 (SYSOSC 启用, 但 MCLK 来自 LFCLK), 则 SYSOSC 保持启用状态, 就像它在 RUN1 中一样。如果从 RUN2 进入 STOP0 (SYSOSC 禁用且 MCLK 源自 LFCLK), 则 SYSOSC 保持启用状态, 就像它在 RUN2 中一样。
- (2) MSPM0C 器件没有 STOP1 模式。
- (3) 如果 IWDG 被激活并且 *no watchdog in Halt* 选项被禁用, 则 STM8L001xx 和 STM8L101xx 器件的 LSI 振荡器在停机模式下被启用。如果停机模式被激活并且禁用了 *no watchdog in Halt* 选项, 则只有 BEEP 和 IWDG 保持在停机模式下保持运行状态。

STM8L05xx 器件具有五种低功耗模式: 等待模式、低功耗运行模式、低功耗等待模式、主动停机模式和停机模式。STM8L001xx 和 STM8L101xx 器件具有三种低功耗模式: 等待模式、主动停机模式和停机模式。STM8 系列有四种低功耗模式: 等待模式、主动停机模式、带 MVR 自动断电的主动停机以及停机模式。

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 or 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 器件的数据表中的按操作模式支持的功能表。下面列出了两个相关数据表, 示例如下:

- [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 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 L 系列 32MHz 微控制器技术参考手册](#)
- [MSPM0 L 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 STM8 都根据器件的可用外设来寄存和映射中断和异常矢量。表 3-10 中包含每个器件系列的中断矢量的总结和比较。

表 3-10. 中断比较

特性	STM8L 和 STM8S	MSPM0L、MSPM0C 和 MSPM0H
中断类型	外设中断：由特定器件确定	外设中断：MSPM0L 的 NVIC 支持多达 19 个外设中断矢量 MSPM0C 的 NVIC 支持多达 23 个外设中断矢量 MSPM0H 支持多达 22 个外设中断矢量 ⁽²⁾
	外部中断：STM8L 值行有 11 个矢量 STM8L101x 有 10 个矢量 STM8S 有 5 个矢量 ⁽¹⁾	
	不可屏蔽中断：RESET、TRAP (软件中断)、TLI (顶级硬件中断) ⁽³⁾	复位、硬故障、SVCall、PendSV、SysTick NMI：软件触发，来自 SYSCTL 的硬件错误信号
优先级	硬件优先级：中断映射的 IRQ 编号	默认优先级：NVIC 编号 ⁽⁴⁾
	不可屏蔽中断被视为具有最高的软件优先级	系统异常 (复位、NMI、硬故障) 具有固定的优先级 -3、-2 和 -1
	可屏蔽中断有 4 个软件优先级：0 (主)、1、2 和 3 (软件优先级被禁用)	外设中断有 4 个可编程优先级：0、64、128、192
优先级设置	ITC_SPRx 寄存器：用来定义每个中断矢量的软件优先级 ⁽⁵⁾ CCR 寄存器：用来自动载入当前中断请求的软件优先级 ⁽⁶⁾	NVIC 中的 IPRx 寄存器：用于设置外设中断优先级
中断屏蔽	相应的中断使能位在外设控制寄存器中设置	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 ⁽⁷⁾
		NVIC 中的 ISER 和 ICER 寄存器：用于启用或禁用外设中断

- (1) 要生成外部中断，必须将相应的 GPIO 端口配置为输入模式并启用中断。
- (2) 除了 NVIC 外，MSPM0 器件上还可以存在中断分组模块 (INT_GROUP0 和 INT_GROUP1)，以便能够将更多外设中断连接到 NVIC。外部中断/GPIO 中断位于 INT_GROUP1 模块中。
- (3) 只有 STM8S 器件支持顶级硬件中断 (TLI)。
- (4) 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。
- (5) 禁止向 VECTxSPR[1:0] 写入 10 (优先级 0)。如果写入 10，则保留之前的值，中断优先级保持不变。
- (6) 无论 CCR 寄存器的位 I1 和 I0 的状态如何，都会处理不可屏蔽中断源。
- (7) MSPM0 的事件处理程序和相关管理寄存器如 [MSPM0 的事件处理程序](#) 所示。

对于 MSPM0 器件，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优先级的中断抢占。对于 STM8 器件，中断或异常的优先级值越高，优先级就越低。STM8 器件具有两种中断管理模式：并发模式和嵌套模式。相关详细信息，请参阅器件特定数据表。

3.6.1.1 MSPM0 的中断管理

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 3-2 显示了外设中断层次结构。

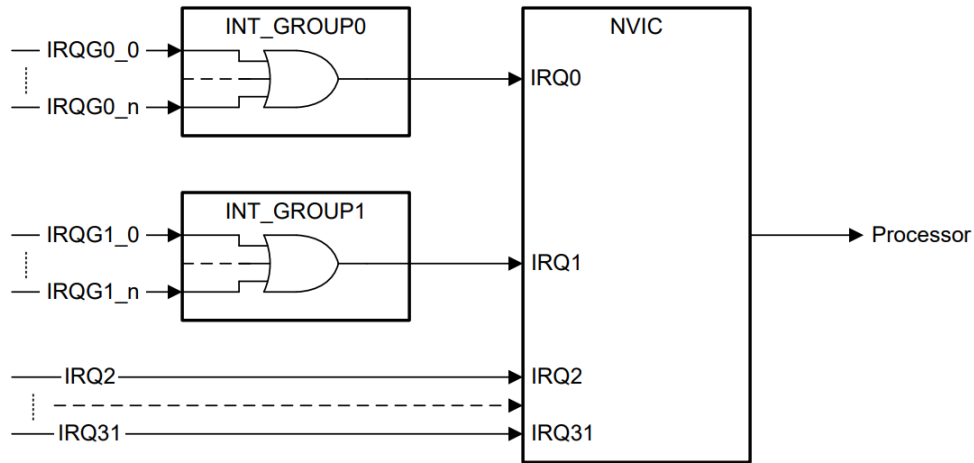


图 3-2. MSPM0 的外设中断层次结构

3.6.1.2 STM8 的中断控制器 (ITC)

图 3-3 显示了 STM8 的中断处理流程图。如果在中断服务例程 (ISR) 中使用指令 SIM 设置中断屏蔽位 I0 和 I1，使用 RIM 删除中断屏蔽会导致软件优先级设置为 0 级。中断服务例程需要以 IRET 指令结束，该指令会导致从栈中恢复已保存寄存器的内容。IRET 指令的结果是，位 I1 和 I0 从栈中恢复并且程序继续执行。

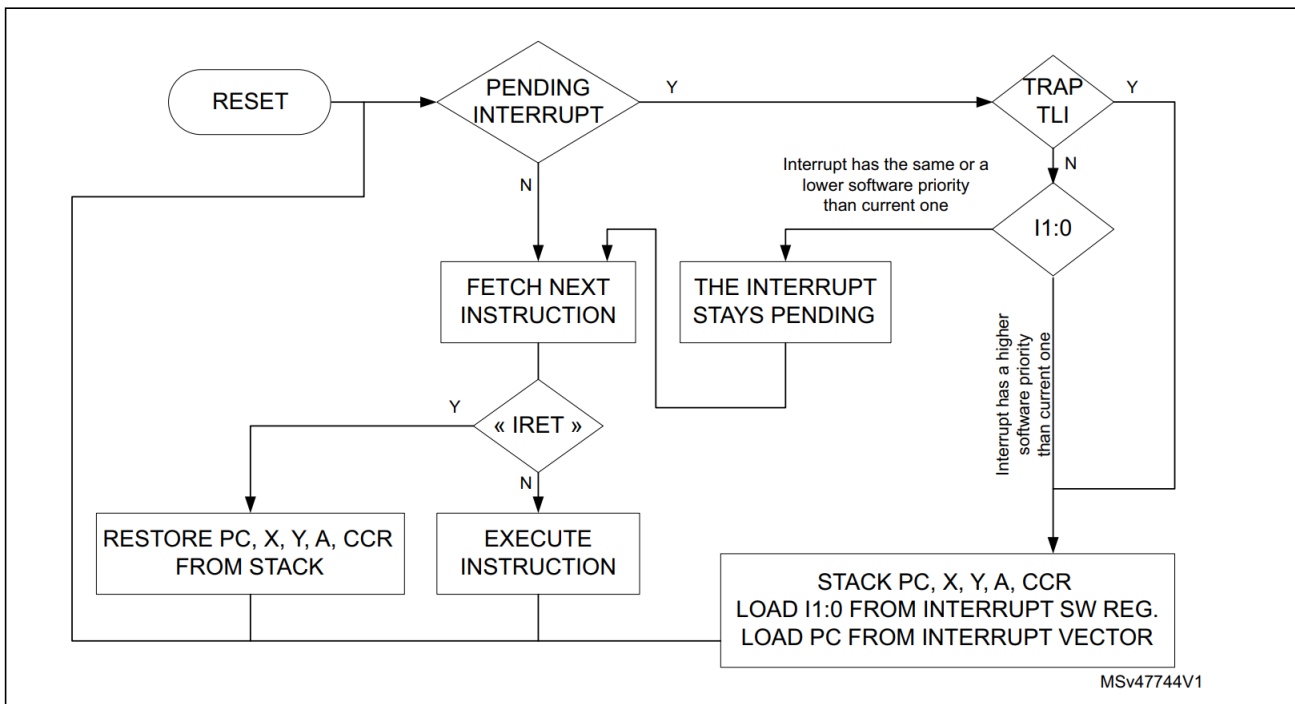


图 3-3. 中断处理流程图

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。

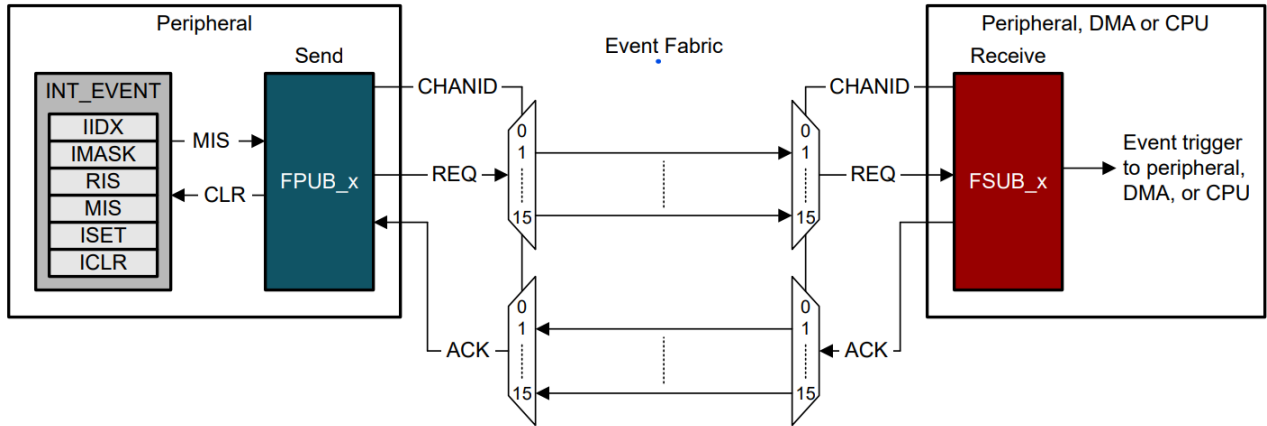


图 3-4. 通用事件路由

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。

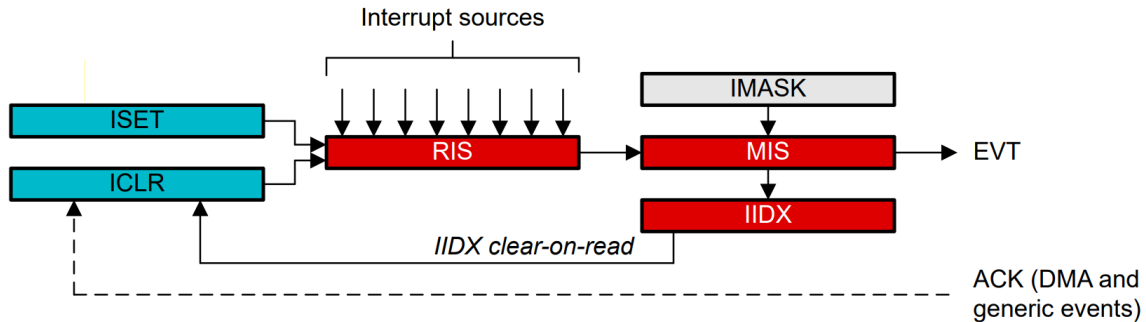


图 3-5. 事件管理寄存器关系

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 MSPM0 L 系列 32MHz 微控制器技术参考手册、或者 MSPM0 C 系列 24MHz 微控制器技术参考手册的 Event 部分。

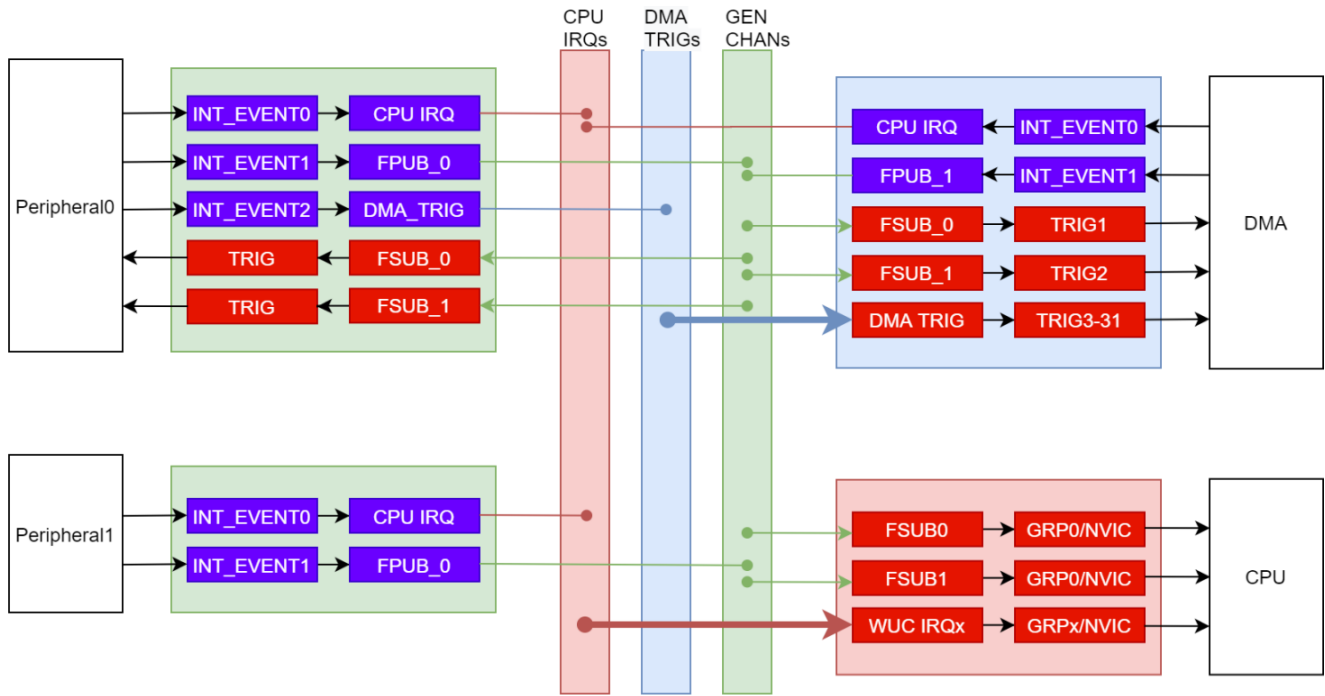


图 3-6. MSPM0 事件和中断处理

3.6.3 事件管理比较

MSPM0 和 STM8 具有不同的事件管理结构和功能。MSPM0 MCU 使用事件处理程序来管理不同的事件。STM8 MCU 使用不同的控制器管理不同的事件。MSPM0 的事件处理程序和 STM8 的事件管理的比较情况如表 3-11 所示。

表 3-11. 事件管理比较

特性		STM8L 和 STM8S	MSPM0L、MSPM0C 和 MSPM0H
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	中断控制器 (ITC)	事件管理器
	外设 → DMA	DMA 控制器 ⁽¹⁾	
	外设 → 外设	触发控制 ⁽²⁾	
路由类型		点对点 (1:1)	点对点
			一分二 (分离器) ⁽³⁾

(1) 并非所有 STM8 器件都具有 DMA 控制器。STM8L001xx 系列、STM8L101xx 系列和 STM8S 系列不支持 DMA 功能。

(2) TIMx 的触发控制器可以生成触发输出信号 (TRGO)，以触发其他 TIM 计时器、ADC 和 DAC。

(3) 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 调试模式比较

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，STM8 器件没有 SWD 两线制 JTAG 端口。具有超快存储器编程功能的单线硬件接口 (SWIM) 是 STM8 器件的主要调试和编程接口。如果您还想使用 SWIM 引脚进行调试，则 SWIM 引脚可以用作标准 I/O，但有一些限制。最安全的方法是在 PCB 上提供一个搭接选项。

3.7.2 编程模式比较

引导加载程序 (BSL) 编程接口是 ARM SWD 和 STM8 SWIM 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 也可用于初始生产编程。

3.7.2.1 引导加载程序 (BSL) 编程选项

MSPM0 和 STM8 器件都支持 BSL 编程接口。表 3-12 比较了 MSPM0 和 STM8 器件系列的不同选项和功能。

表 3-12. BSL 功能比较

BSL 特性	STM8L 和 STM8S	MSPM0L	MSPM0C	MSPM0H
嵌入式 BSL 代码存储	ROM ⁽¹⁾	ROM ⁽²⁾	不支持	不支持
可定制	否	可配置调用引脚、 COMM 引脚和插件特性	否	否
辅助 BSL 代码存储	UBC 程序区域 ⁽¹⁾	主闪存 ⁽²⁾	主闪存	主闪存
BSL 在空白器件上启动	是	是	不适用	不适用
自动检测编程接口	是	是	不适用	不适用
安全性	读出保护 (ROP) ; 命令校验和	安全引导选项 ; CRC 保护 AES256 (带密钥存储 区) ; TRNG	CRC ; 防火墙 ; IP 保护	CRC ; 防火墙 ; IP 保护
调用方法	检查 BSL 选项字节是否 为 0x55AA, 或者程序存 储器是否为原始格式	在 BOOTRST 时调用高 电平引脚 ; SW 进入	在 BOOTRST 时调用高 电平引脚 ; SW 进入	在 BOOTRST 时调用高 电平引脚 ; SW 进入
支持的接口				
UART	是	是	辅助 BSL	辅助 BSL
I2C	不支持	是	辅助 BSL	辅助 BSL
SPI	是	需要自定义插件	辅助 BSL	辅助 BSL
CAN	是 ⁽³⁾	已计划提供插件 ⁽⁴⁾	辅助 BSL	辅助 BSL

- (1) STM8Sx03xx、STM8S001xx、STM8L101xx 和 STM8L001xx 器件没有嵌入式 BSL (微控制器内部未实现 ROM BSL)。使用这些器件时, 用户必须编写自己的 BSL 代码并在 UBC 程序区域中保存自己的 BSL 代码。
- (2) MSPM0C 器件没有 ROM BSL 代码。在使用这些器件时, 用户必须编写自己的 BSL 代码, 包括定制的插件接口和 BSL 内核, 并将自己的代码保存在主闪存中。
- (3) 仅当存在外部时钟 (8MHz、16MHz 或 24MHz) 时, 才能使用 STM8 器件的 CAN 外设。
- (4) 仅适用于选定器件。

4 数字外设比较

4.1 通用 I/O (GPIO、IOMUX)

MSPM0 GPIO 功能几乎涵盖了 STM8S 和 STML 系列提供的所有功能。STM8 使用术语“引脚功能”和“端口功能”来指代负责管理器件引脚和生成中断等的功能。以下是 MSPM0 GPIO 和 IOMUX 功能的描述：

- 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 STM8 GPIO 相同的功能。此外，MSPM0 提供了 STM8L 和 STM8S 系列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

特性	STM8S 和 STM8L	MSPM0L、MSPM0L、MSPM0H
输出模式	推挽 开漏	推挽 具有下拉的开漏 高阻态
输入模式	上拉 悬空 模拟	悬空 上拉或下拉 模拟
GPIO 速度选择	为每个 I/O 选择速度 高达 2MHz 的 Speed0 速度高达 10MHz	MSPM0 在所有 IO 引脚上提供标准 IO (SDIO)。 MSPM0 高速 IO (HSIO) 在选定引脚上可用。 在 VDD ≥ 2.7V 时，SDIO 和 HSIO 均支持高达 32MHz，在 VDD ≥ 1.71V 时，HSIO 支持高达 24MHz
原子位设置和复位	是	是
替代功能	使用 ODR、IDR 和 DDR 寄存器	使用 IOMUX
快速切换	至少每两个时钟	每个时钟周期切换引脚
唤醒	外部中断	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)

STM8S 系列和 MSPM0 都提供用于执行异步 (无时钟) 通信的外设。而 STM8L 系列具有 USART (通用同步异步接收发送器)，可提供灵活的方式来与外部设备进行全双工数据交换。

表 4-2. UART 标准功能比较

特性	STM8S 和 STML	MSPM0L、MSPM0L、MSPM0H
数据方向	LSB 在前	MSB 在前或 LSB 在前
硬件流控制	流	是
可配置的停止位	1、2	1、2
Tx, Rx FIFO 深度	2	4
多处理器	是	是
在低功耗模式下有效	是 (仅在等待模式下)	是 (均为低功耗)
单线半双工通信	是	是 ⁽¹⁾
从低功耗模式唤醒	是	是
奇偶校验	偶数, 奇数	偶数, 奇数
使用 DMA 进行通信	是 ⁽²⁾	是

(1) 需要在发送和接收之间重新配置外设。

(2) 只有 STM8L 超值系列具有 DMA。

表 4-3. UART 高级功能比较

特性	STMS 和 STML	MSPM0L、MSPM0L、MSPM0H
同步模式	是 ⁽¹⁾	否
数据长度	8、9	5、6、7、8
过采样	否	16x 8x 3x
LIN 硬件支持	是 ⁽²⁾	是
DALI 硬件支持	否	是
IrDA 硬件支持	是	是
曼彻斯特编码 HW 支持	否	是
智能卡模式 (ISO7816)	是	是
外部驱动器使能	否	是
干扰滤波器	否	是

(1) 对于 STM8S，只有 UART1、UART2 和 UART4 具有用于同步通信的发送器时钟输出。STM8L 具有 UASART 模块进行同步通信。

(2) 相比于 MSPM0，STM8 LIN 模块的硬件模块具有更全面的功能，例如：LIN 中断和定界符生成、报头错误检测等。

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 STM8 都支持串行外设接口 (SPI)。总的来说，MSPM0 和 STM8 SPI 支持是相当的，但存在表 4-4 中列出的差异。

表 4-4. SPI 功能比较

特性	STM8S 和 STM8L	MSPM0L、MSPM0C 和 MSPM0H
操作线	SCK、MOSI、MISO、NSS	SCLK、PICO、POCI、CSx
控制器或外设操作	是	是
多控制器功能	是	否
数据顺序	MSB 在前或 LSB 在前	MSB 在前或 LSB 在前
数据位宽 (控制器模式)	未提及	4 至 16 位
数据位宽 (外设模式)		7 至 16 位
最大速度	10MHz	16MHz
单工传输 (单向数据线)	是	是
硬件芯片选择管理	是 (1 个外设)	是 (4 个外设)
I/O 时钟的相位控制	是	是
SPI 格式支持	Motorola	Motorola、TI、MICROWIRE
硬件 CRC	是 (STM8S)	否, MSPM0 提供 SPI 奇偶校验模式
低功耗模式	等待模式	睡眠模式
TX FIFO 深度	1 (缓冲器)	4
RX FIFO 深度	1 (缓冲器)	4

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 内部集成电路接口 (I2C)

MSPM0 和 STM8 均支持 I2C 外设。总的来说，MSPM0 和 STM8 I2C 支持是相当的，但存在表 4-5 中列出的显著差异。

表 4-5. I2C 功能比较

特性	STM8S 和 STM8L	MSPM0L、MSPM0C 和 MSPM0H
控制器和目标模式	是	是
多控制器功能	是	是
最大传输速率	400kHz (快速)	1Mbps (超快速)
寻址模式	7 位或 10 位	7 位
地址数量 (目标模式)	2 个地址	2 个地址
常规调用	是	是
事件管理	是	是
PEC 管理	是	是
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	是	是
可编程模拟和数字噪声滤波器	是	是

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)

STM8 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 4-6. 计时器命名

STM8L		STM8S		MSPM0L		MSPM0C		MSPM0H	
计时器名称	缩写名称	计时器名称	缩写名称	计时器名称	缩写名称	计时器名称	缩写名称	计时器名称	缩写名称
		高级控制	TIM1	高级控制	TIMA0 (仅在 MSPM0Lx22 x 中)	高级控制	TIMA0	高级控制	TIMA0
通用	TIM2,3	通用	TIM2、3、5	通用	TIMG0,1,2,4, 5,8,12	通用	TIMG1,2,8,14	通用	TIMG1,2,8,14
基础型	TIM4	基础型	TIM4,6						

表 4-7. 计时器功能比较

特性	STM8L	STM8S	MSPM0L、MSPM0C 和 MSPM0H
分辨率	8、16 位	8、16 位	16 位
PWM	是	是	是
捕获	是	是	是
比较	是	是	是
重复计数器	否	是	是
单次触发	是	是	是
向上/向下计数功能	是	是	是
低功耗模式	是	是	是
QE1 支持	否	否	是
可编程预分频器	是	是	是
影子寄存器模式	是	是	是
事件，中断	是	是	是
自动重新加载功能	是	是	是
故障处理	是	是	是

表 4-8. 计时器模块替换

STM8	MSPM0 等效功能	原因
TIM1	TIMA0	高级控制、16 位分辨率、加/减计数器、重复计数器
TIM2、3、5	TIMG0-11, TIMG14	16 位分辨率、通用、捕捉/比较功能
TIM4、6	任何 ⁽¹⁾	基本计时器

表 4-9. 计时器用例比较

特性	STM8L 和 STM8S	MSPM0L、MSPM0C 和 MSPM0H
PWM	TIM1TIM2、3、5	所有计时器
捕获	TIM1TIM2、3、5	所有计时器
比较	TIM1TIM2、3、5	所有计时器
单次触发	TIM1TIM2、3、5	所有计时器
预分频器	所有计时器	所有计时器
同步	所有计时器	所有计时器

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)

STM8 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 4-10. WWDT 命名

STM8	MSPM0
独立看门狗 (IWDG) 窗口看门狗 (WWDG) ⁽¹⁾	窗口化看门狗计时器 (WWDT)

(1) 只有 STM8L 具有 WWDG。

表 4-11. WDT 功能比较

特性	STM8L 和 STM8S	MSPM0L、MSPM0C 和 MSPM0H
窗口模式	是	是
间隔计时器模式	否	是
LFCLK 源	是	是
中断	是	是
计数器分辨率	8 位 (IWDG) 7 位 (WWDG)	25 位
时钟分频器	是	是

WWDT 代码示例：有关 WWDT 代码示例的信息，请参阅 [MSPM0 SDK 示例指南](#)。

5 模拟外设比较

5.1 模数转换器 (ADC)

STM8 和 MSPM0 都提供 ADC 外设来将模拟信号转换为数字等效信号。对于 STM8，STM8L001XX 和 STM8L101XX 没有 ADC 模块。STM8S 系列和 STM8L 系列的其余产品分别提供 10 位或 12 位 ADC。对于 MSPM0，两个器件系列均具有 12 位 ADC。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

特性	STM8S	STM8L	MSPM0
分辨率 (位)	10	12	12、10、8
转换率	0.43 Msps	1 Msps	1.68 Msps
硬件均值计算	否	否	是
FIFO	否	否	是
ADC 基准 (V)	内部：1.224	内部：1.48, VDD	内部：1.4、2.5、VDD
	外部： $2.75V \leq V_{REF} \leq V_{DDA}$	外部： $2.4V \leq V_{REF} \leq V_{DDA}$	外部： $1.4 \leq V_{REF} \leq V_{DD}$ ⁽¹⁾
工作电源模式	等待和低功耗等待	等待	运行、睡眠、停止、待机 ⁽²⁾
自动断电	否	否	是
外部输入通道	高达 16	高达 28	MSPM0L：高达 16，MSPM0C：高达 27，MSPM0H：高达 27
内部输入通道	温度传感器 内部基准电压	温度传感器 内部基准电压	温度传感器 电源监控，模拟信号链
DMA 支持	否	是	是
ADC 窗口比较器单元	否	否	是
ADC 数量	高达 1	高达 1	高达 1

(1) 只有 C 系列中的 MSPM0C1105 和 MSPM0C1106 支持外部基准。

(2) ADC 可以在待机模式下触发，从而改变工作模式。

表 5-2. 转换模式

STM8	MSPM0	注释
单次转换模式	单通道单次转换	ADC 对单个通道进行一次采样和转换
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。
连续和缓冲连续模式	单通道重复转换	一个所选通道将被重复采样和转换
连续扫描模式	序列通道重复转换	一组通道将被重复采样和转换。

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)

在 STM8 中，STML 值系列有比较器，但 S 系列和 L 系列的其余部分没有。至于 MSPM0，L 系列提供集成比较器作为可选外设，而 C 系列器件则不提供。在这两个器件比较器系列中，这些都表示为 COMP_x，其中最后一个字符 x 指所考虑的特定比较器模块。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控制 PWM 信号。表 5-3 汇总了 MSPM0 和 STM8 比较器模块逐功能比较结果。

表 5-3. COMP 功能集比较

特性	STM8L 超值系列	MSPM0L 和 MSPM0C ³
可用的比较器	高达 2	高达 1
输出路由	外部 I/O	多路复用 I/O 引脚
	TIM1 BRK 或 OCREFCLR 输入 TIM2/TIM3 输入捕捉 2	-
	中断或事件接口	中断或事件接口
正输入	外部 I/O	多路复用 I/O 引脚 OPA1 输出 DAC8 ¹ 输出
负输入	内部基准电压 内部基准电压的约量 (1/4、1/2、3/4) DAC 输出 三个外部 I/O 中的一个	多路复用 I/O 引脚 内部温度传感器 OPA0 输出 DAC8 输出
可编程迟滞	否	无、10mV、20mV、30mV
		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)
窗口比较器配置	是	否 (单个 COMP)
输入短路模式	否	是
工作模式	更优速度、消耗率	高速, 低功耗
输出滤波	否	消隐滤波器
		可调节模拟滤波器
输出极性控制	否	是
中断	上升沿	上升沿
	下降沿	下降沿
	双边沿	输出就绪
交换输入 ⁽²⁾	否	是

(1) 8 位 DAC 集成在 COMP 中。

(2) 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。

(3) MSPM0H 当前没有 COMP。

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 电压基准 (VREF)

STM8 和 MSPM0 都具有内部基准, 可用于为内部外设提供基准电压, 但可以向外部外设输出内部基准电压。在 STM8 中, 只有 STM8L 值系列具有内部基准。

表 5-4. VREF 功能集比较

特性	STM8S	STM8L	MSPM0
内部基准 (V)	1.8, 1.2	1.22	1.4、2.5
外部基准 (V)	$2.4 \leq V_{REFP} \leq V_{DDA} V$	$2.4 \leq V_{REFP} \leq V_{DDA} V$	$1.4 \leq V_{REF} \leq V_{DD} V^{(1)}$
输出内部基准	否	是	是
在内部连接到 ADC	是	是	是
在内部连接到 DAC	不适用	是	不适用
在内部连接到 COMP	不适用	是	是
在内部连接到 OPA	不适用	不适用	支持 MSPM0L ; 不支持 MSPM0C 和 H

(1) MSPM0C1103 和 MSPM0C1104 不支持外部基准。

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 代码示例的信息，请参阅 [MSPM0 SDK 示例指南](#)。

6 总结

本应用手册旨在帮助工程师从 STM8 器件迁移到 MSPM0。本文档对 STM8 器件和 MSPM0 在生态系统、CPU、模拟和数字外设方面的差异进行了比较。该应用手册还详细介绍了如何通过示例快速开始使用 MSPM0，以帮助用户加快开发进度。

7 参考资料

- 德州仪器 (TI), [MSPM0-SDK 用户指南](#)
- 德州仪器 (TI), [MSPM0 工具指南](#)
- 德州仪器 (TI), [Driverlib API 指南](#)
- 德州仪器 (TI), [Code Composer Studio \(CCS\)](#)
- 德州仪器 (TI), [MSPM0 L 系列 32MHz 微控制器技术参考手册](#), 技术参考手册
- 德州仪器 (TI), [MSPM0 C 系列 24MHz 微控制器](#), 技术参考手册
- IAR, [IAR 网站](#)
- Keil, [Keil 网站](#)
- 德州仪器 (TI), [CCS 快速入门指南](#)
- 德州仪器 (TI), [CCS 培训视频](#)
- 德州仪器 (TI), [CCS 用户指南](#)
- 德州仪器 (TI), [IAR 快速入门指南](#)
- 德州仪器 (TI), [IAR 培训视频](#)
- IAR, [IAR 用户指南](#)
- Keil, [Keil 快速入门指南](#)
- Keil, [Keil 培训视频](#)
- Keil, [Keil 入门](#)
- 德州仪器 (TI), [SYSCONFIG IDE](#)
- 德州仪器 (TI), [MSPM0 SysConfig 指南](#)
- 德州仪器 (TI), [XDS110 调试探针](#)
- 德州仪器 (TI), [J-Link 调试探针页面](#)
- 德州仪器 (TI), [LP-MSPM0G3507 LaunchPad 开发套件](#)
- 德州仪器 (TI), [LP-MSPM0L1306 LaunchPad 开发套件](#)
- 德州仪器 (TI), [LP-MSPM0C1104 LaunchPad 开发套件](#)

8 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision * (December 2023) to Revision A (May 2025)	Page
• 在整个文档中将 MSPM0H 系列更新为与 STM8 器件相比较.....	4
• 更新了整个文档的 MSPM0C 系列规格.....	4
• 更新了新生态系统、包括 IDE 版本和 MSP 闪存工具.....	5

重要通知和免责声明

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 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。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
版权所有 © 2025，德州仪器 (TI) 公司