

# Application Note

## MSPM0 设计流程指南



Eason Zhou, Zoey Wei, Helic Chi, and Janz Bai

### 摘要

本应用手册详细说明了有关如何开发 MSPM0 MCU 的步骤，并提供了相关材料和说明。

### 内容

1 概述.....	5
2 第 1 步：MSPM0 选择.....	6
3 第 2 步：MSPM0 评估.....	8
3.1 硬件设置.....	8
3.2 MSPM0-SDK 设置.....	10
3.3 SysConfig 设置.....	15
3.4 IDE 快速入门.....	21
4 第 3 步：硬件设计.....	46
4.1 获取 MSPM0 包.....	46
4.2 修复引脚功能.....	47
4.3 原理图和 PCB 生成.....	47
5 第 4 步：大规模生产.....	49
5.1 生成生产映像.....	49
5.2 对软件编程.....	50
5.3 对硬件编程.....	53
6 第 5 步：质量和可靠性说明.....	55
6.1 质量和可靠性材料入口.....	55
6.2 失效信息收集和分析指南.....	55
7 常见问题.....	56
7.1 MSPM0 编程故障.....	56
7.2 解锁 MCU.....	57
7.3 MCU 在调试和自由运行时的表现不同.....	60
7.4 BSL 相关问题.....	60
7.5 设置 SWD 密码.....	60
7.6 CCS 常见问题.....	62
7.7 Keil 常见问题.....	68
8 其他信息.....	69
8.1 点亮 LED 和 CCS 简介.....	69
8.2 生成 PCB 库的步骤.....	74
8.3 MSP-GANG 简介.....	77
9 总结.....	79
修订历史记录.....	79

### 插图清单

图 1-1. MSPM0 设计流程.....	5
图 2-1. MSPM0 器件列表.....	6
图 2-2. MSPM0 重要文档列表.....	6
图 2-3. 器件比较表.....	7
图 2-4. 器件订购和质量视图.....	7
图 3-1. MSPM0G3507 LaunchPad.....	9

图 3-2. LaunchPad 设置视图.....	9
图 3-3. MSPM0-SDK 下载.....	10
图 3-4. MSPM0-SDK 分步安装.....	10
图 3-5. MSPM0-SDK 结构.....	11
图 3-6. MSPM0-SDK 示例.....	11
图 3-7. SysConfig 安装.....	15
图 3-8. MSPM0 SysConfig.....	16
图 3-9. SysConfig 视图.....	16
图 3-10. 基本操作.....	17
图 3-11. 工程配置.....	17
图 3-12. 电路板视图.....	18
图 3-13. NONMAIN 视图.....	18
图 3-14. SYSCTL 视图.....	19
图 3-15. 外设视图.....	20
图 3-16. CCS 安装.....	22
图 3-17. MSPM0 支持选择.....	22
图 3-18. J-link 选择.....	23
图 3-19. 加载 SDK 产品.....	24
图 3-20. 选择或更改 SDK 版本.....	25
图 3-21. 选择 CCS 工作区.....	25
图 3-22. 导入工程.....	26
图 3-23. CCS 工程概述.....	26
图 3-24. 更改调试器选择.....	27
图 3-25. 调试代码.....	27
图 3-26. 常用调试功能.....	27
图 3-27. 在 MSPM0 衍生产品之间进行迁移.....	28
图 3-28. 生成十六进制文件.....	29
图 3-29. 对 NONMAIN 编程.....	29
图 3-30. 将 MSPM0 SDK 添加到 IAR.....	30
图 3-31. 为 MSPM0 安装 SysConfig.....	31
图 3-32. 导入 SDK 示例.....	32
图 3-33. 将 SysConfig 与 IAR 配合使用.....	32
图 3-34. 下载和调试.....	33
图 3-35. 在 MSPM0 衍生产品之间进行迁移.....	34
图 3-36. 生成十六进制文件.....	35
图 3-37. 对 NONMAIN 编程.....	35
图 3-38. 打开 Pack Installer.....	36
图 3-39. 搜索器件.....	36
图 3-40. 安装器件包.....	37
图 3-41. 批准许可证.....	37
图 3-42. 编辑 syscfg.bat.....	38
图 3-43. 编辑 MSPM0_SDK_syscfg_menu_import.cfg.....	38
图 3-44. Keil 自定义工具.....	38
图 3-45. 导入 MSPM0_SDK_syscfg_menu_import.cfg 文件.....	39
图 3-46. 完成 SysConfig 设置.....	39
图 3-47. 打开工程.....	39
图 3-48. 选择 Keil 工程.....	40
图 3-49. 打开 .syscfg 文件.....	41
图 3-50. 打开目标的选项.....	41
图 3-51. 选择调试窗格.....	42
图 3-52. 检查 XDS110 探针的设置.....	42
图 3-53. 检查 J-Link 探针的设置.....	43
图 3-54. Flash 下载设置.....	43
图 3-55. 下载工程.....	44
图 3-56. Keil 下的构建 RTOS 示例.....	44
图 3-57. 在 MSPM0 衍生产品之间进行迁移.....	45
图 3-58. 生成十六进制文件.....	45
图 3-59. 对 NONMAIN 编程.....	46

图 4-1. Ultra Librarian 工具入口.....	46
图 4-2. 生成外设和引脚分配文件.....	47
图 4-3. MSPM0 最小系统.....	47
图 4-4. MSPM0 原理图.....	48
图 5-1. 编程软件和工具.....	49
图 5-2. 通过 SWD 编程.....	50
图 5-3. 通过引导加载程序编程.....	51
图 5-4. J-Flash 快速入门.....	52
图 5-5. TMDSEMU110-U 的引脚连接.....	53
图 5-6. XDS110 板载元件.....	53
图 5-7. LP-XDS110ET.....	54
图 7-1. E2E 在线论坛.....	56
图 7-2. 设备管理器视图.....	56
图 7-3. CCS 误差.....	57
图 7-4. 通过 GUI 解锁.....	58
图 7-5. 通过 Uniflash 解锁.....	59
图 7-6. 通过 CCS 解锁.....	59
图 7-7. 禁用 BSL.....	60
图 7-8. 启用 SWD 密码.....	61
图 7-9. 对器件重新编程.....	62
图 7-10. 更改优化级别.....	63
图 7-11. 无法选择工程.....	63
图 7-12. 删除同名工程.....	64
图 7-13. 找不到 .h 文件.....	64
图 7-14. 安装 Arm GCC.....	65
图 7-15. 还原默认调试设置.....	66
图 7-16. 擦除所需的存储器.....	66
图 7-17. 从 CCS 输出数据日志.....	67
图 7-18. 从 SDK 复制 Keil 示例.....	68
图 A-1. CCS 安装.....	69
图 A-2. MSPM0 支持选择.....	69
图 A-3. J-link 选择.....	70
图 8-4. 硬件设置.....	70
图 8-5. 选择 CCS 工作区.....	71
图 8-6. 导入工程.....	71
图 8-7. 删除重复的工程.....	72
图 8-8. 调试代码.....	72
图 8-9. 常用工程设置.....	73
图 8-10. 常用的调试函数.....	73
图 8-11. Ultra Librarian 工具开始页面.....	74
图 8-12. Ultra Librarian 工具器件选择.....	74
图 8-13. Ultra Librarian 工具 CAD 下载.....	75
图 8-14. 运行 Altium Designer 脚本.....	75
图 8-15. 生成库.....	76
图 8-16. 选择封装.....	76
图 8-17. 导入库.....	76
图 8-18. MSP-GANG 引脚分配.....	77
图 8-19. 使用带有 GUI 的 MSP-GANG 下载代码.....	77
图 8-20. 启用非主编程.....	78
图 8-21. 生成并保存映像.....	78
图 8-22. 更改模式.....	78
图 8-23. 离线编程.....	79

## 表格清单

表 3-1. MSPM0 开发链.....	8
表 3-2. MSPM0 调试器比较.....	8
表 3-3. MSPM0 示例覆盖范围.....	14
表 3-4. 空工程描述.....	14

表 3-5. MSPM0 支持的 IDE 概览.....	21
表 5-1. IDE 生成的产品文件.....	49
表 5-2. XDS110 调试器摘要.....	53
表 7-1. 建议的工具版本.....	57
表 7-2. 解锁命令.....	57
表 7-3. 解锁方法选择.....	57

## 商标

LaunchPad™, Code Composer Studio™, SimpleLink™, C2000™, and TIVA™ are trademarks of Texas Instruments.

Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.  
所有商标均为其各自所有者的财产。



## 1 概述

该文档介绍了开发 MSPM0 MCU 工程的步骤，还提供相关文档列表和分步说明。有关开发人员会遇到的常见问题，请参阅[节 7](#)。

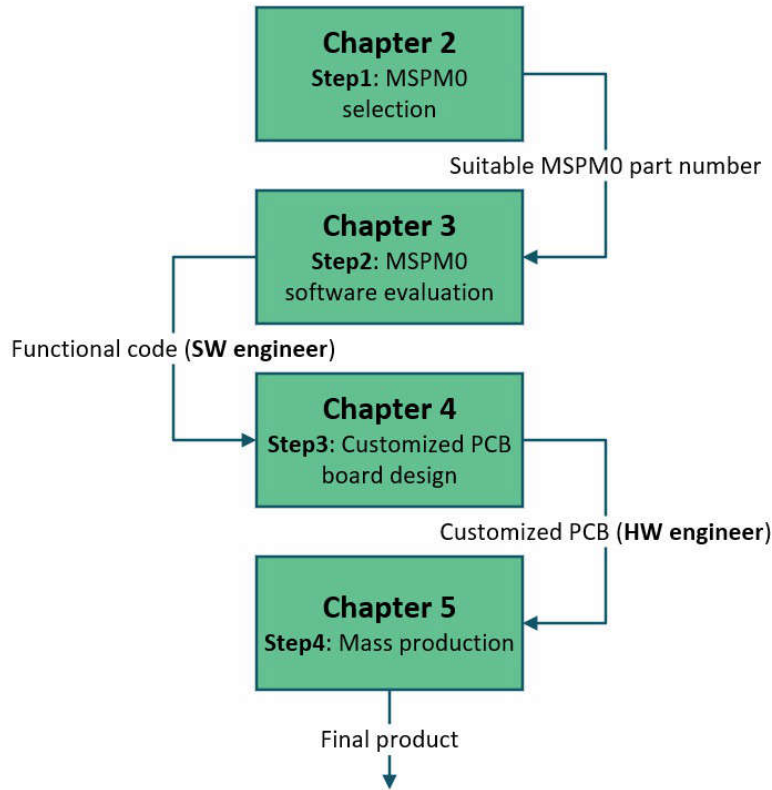


图 1-1. MSPM0 设计流程

## 2 第 1 步：MSPM0 选择

此步骤说明如何查找 MSPM0 可订购编号。

访问 [Arm Cortex-M0+ MCU 产品页](#)，查看 MSPM0 器件列表。导航到此页面后，使用左侧的筛选器根据 MCU 外设要求执行初始筛选，或者使用页面左侧的搜索框直接导航到器件页面。

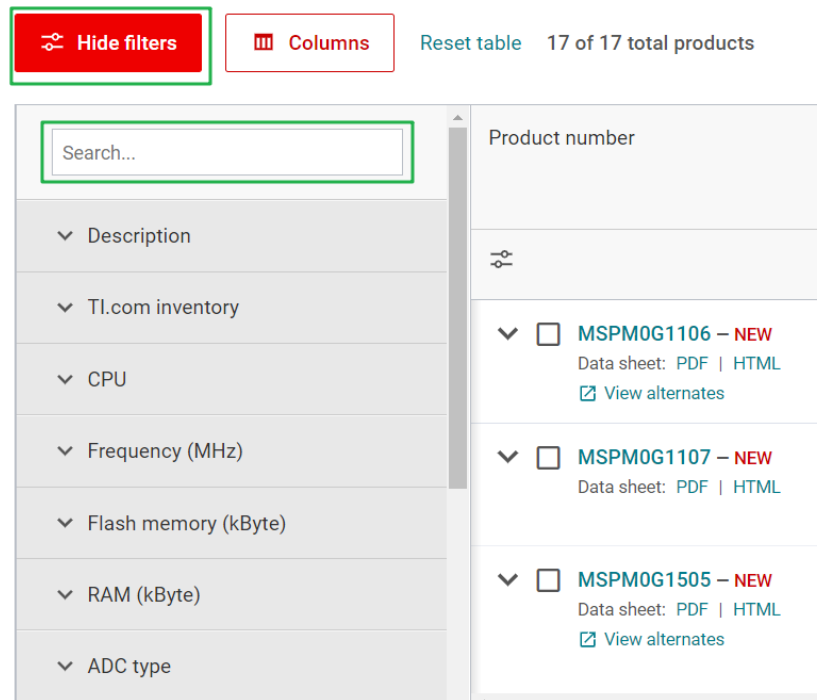


图 2-1. MSPM0 器件列表

导航至器件页面后，可以找到特定产品的更多规格或功能详细信息。主要文档包括数据表、技术参考手册 (TRM) 和勘误表。器件特定数据表介绍了 MSPM0 的参数和功能数据信息。器件特定 TRM 介绍了 MSPM0 器件的应用方法和特性。器件特定勘误表显示了 MSPM0 相关系列或版本的更正说明。



图 2-2. MSPM0 重要文档列表

图 2-3 展示了器件特定数据表中的 *器件比较表*。用户可使用此表比较不同的器件型号。

### 5 Device Comparison

Table 5-1. Device Comparison

DEVICE NAME <sup>(1) (2)</sup>	FLASH / SRAM (KB)	QUAL <sup>(3)</sup>	ADC CH.	COMP	OPA	GPAMP	UART/I2C/SPI	TIMG	GPIOs	5-V TOL. IO	PACKAGE [BODY SIZE] <sup>(4)</sup>
MSPM0L1306xRHB	64 / 4	T/S	10	1	2	1	2 / 2 / 1	4	28	2	32 VQFN [5 mm × 5 mm] <sup>(5)</sup>
MSM0L1305xRHB	32 / 4										
MSM0L1304xRHB	16 / 2										
MSPM0L1306xDGS28	64 / 4	T/S	10	1	2	1	2 / 2 / 1	4	24	2	28 VSSOP [7.1 mm × 3 mm]
MSPM0L1305xDGS28	32 / 4										
MSPM0L1304xDGS28	16 / 2										
MSPM0L1346xDGS28	64 / 4	T	9								
MSPM0L1345xDGS28	32 / 4										

图 2-3. 器件比较表

请参阅器件页面上的 *订购和质量* 页面，以查看可订购器件型号和参考价格。

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	Enter quantity <b>Add to cart</b> Limit: 5	93	1ku   <span>▼</span>	1   LARGE T&R <span>Select</span> ▼
XMSM0L1306SDGS28R ACTIVE	Enter quantity <b>Add to cart</b> Limit: 10	170	1ku   <span>▼</span>	5,000   LARGE T&R

图 2-4. 器件订购和质量视图

## 3 第 2 步：MSPM0 评估

此步骤说明如何为 MSPM0 设置硬件和软件评估环境。有关基于 CCS 和 LaunchPad 的分步说明，请参阅节 8.1。

表 3-1 汇总了 MSPM0 开发链中所有必需组件。后续章节单独说明了各个器件。

**表 3-1. MSPM0 开发链**

IDE	SysConfig ( 代码生成器 GUI )	SDK	调试器	硬件
集成了 SysConfig 的 CCS		MSPM0 SDK	具有板载 XDS110 的 LaunchPad	
Keil	独立 SysConfig		XDS110	定制板
IAR			J-Link	

### 3.1 硬件设置

#### 3.1.1 调试器选择

本节总结了支持 MSPM0 器件的不同调试器。XDS110 调试器归 TI 所有，与通用调试器相比，这些调试器支持更多功能。有关 XDS110 调试器的更多详细信息，请参阅节 5.3。

**表 3-2. MSPM0 调试器比较**

特性	XDS110 (TMDSEMU110-U)	XDS110 板载元件	J-Link
cJTAG (SBW)	√	√	√
BSL 工具	√	√	
反向通道 UART	√	√	
电源	1.8V 至 3.6V	3.3V 至 5V	5V
IDE	CCS、IAR、Keil	CCS、IAR、Keil	CCS、IAR、Keil

### 3.1.2 LaunchPad 简介

TI 建议使用 LaunchPad™ 开始 MSPM0 开发。图 3-1 展示了 LaunchPad 的概况。LaunchPad 包含 MCU 和 XDS110 调试器。用户可以在移除跳线后使用 J-Link 等调试器来调试 MCU。

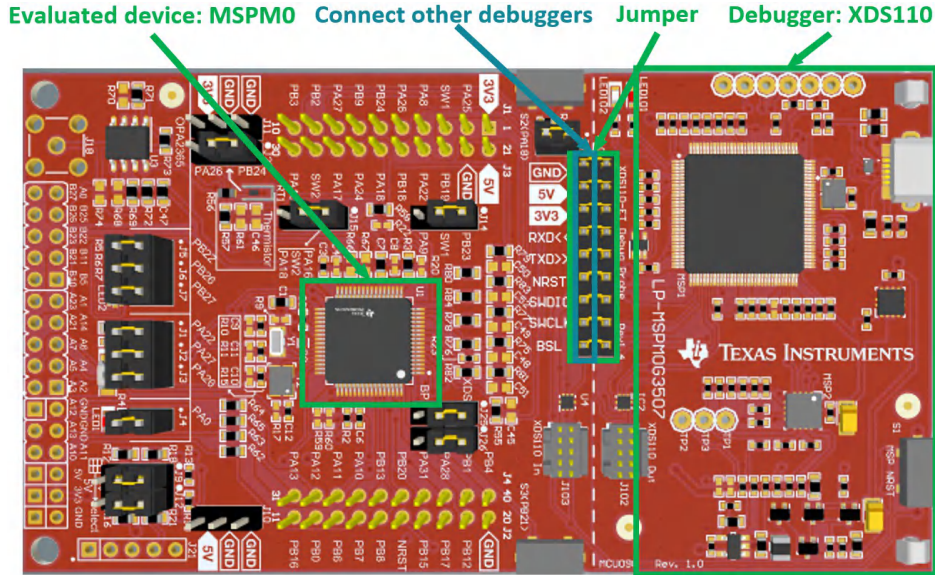


图 3-1. MSPM0G3507 LaunchPad

以下链接显示了可订购的 LaunchPad 器件和相关用户指南。

- [LP-MSPM0L1306 登录页面](#)
- [LP-MSPM0G3507 登录页面](#)
- [LP-MSPM0C1104 登录页面](#)
- [MSPM0L1306 LaunchPad 开发套件用户指南](#)
- [MSPM0G3507 LaunchPad 开发套件用户指南](#)
- [MSPM0C1104 LaunchPad 开发套件用户指南](#)

实际的 LaunchPad 设置条件如图 3-2 所示，可以通过 USB 端口对其进行调试和供电。

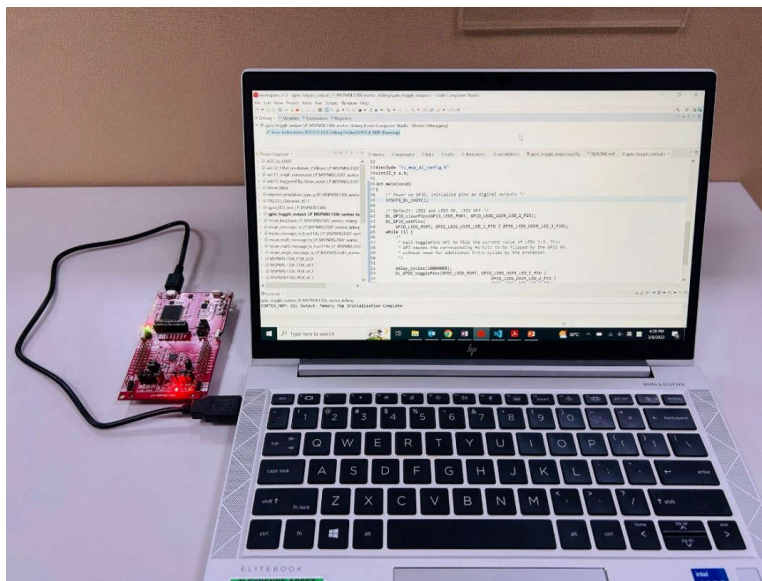


图 3-2. LaunchPad 设置视图

## 3.2 MSPM0-SDK 设置

MSPM0-SDK 提供了全面的软件、工具和文档集合，可加快 MSPM0 MCU 平台的应用程序开发，通过各种驱动程序、库和示例为单个软件包提供一致且连贯的体验。

### 3.2.1 MSPM0-SDK 安装

本节详细介绍了安装 MSPM0-SDK 的步骤。安装后，默认 SDK 目录路径为：`C:\ti\mspm0_sdk_x_xx_xx_xx`。

1. 下载之前，并且准备好 myTI 帐户。在[此处](#)注册 myTI 帐户。
2. 从[产品页](#)下载最新的 MSPM0-SDK。点击 *Download options*，选择操作系统，然后点击文件名开始下载。

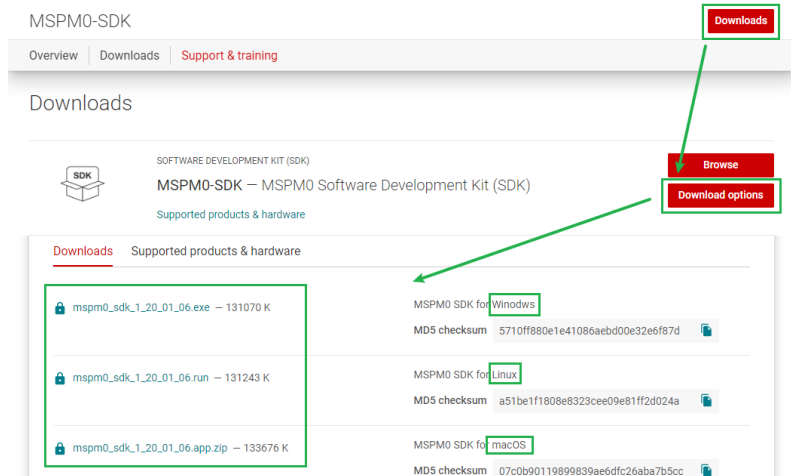


图 3-3. MSPM0-SDK 下载

3. 下载完成后，按照图 3-4 中的步骤完成安装。

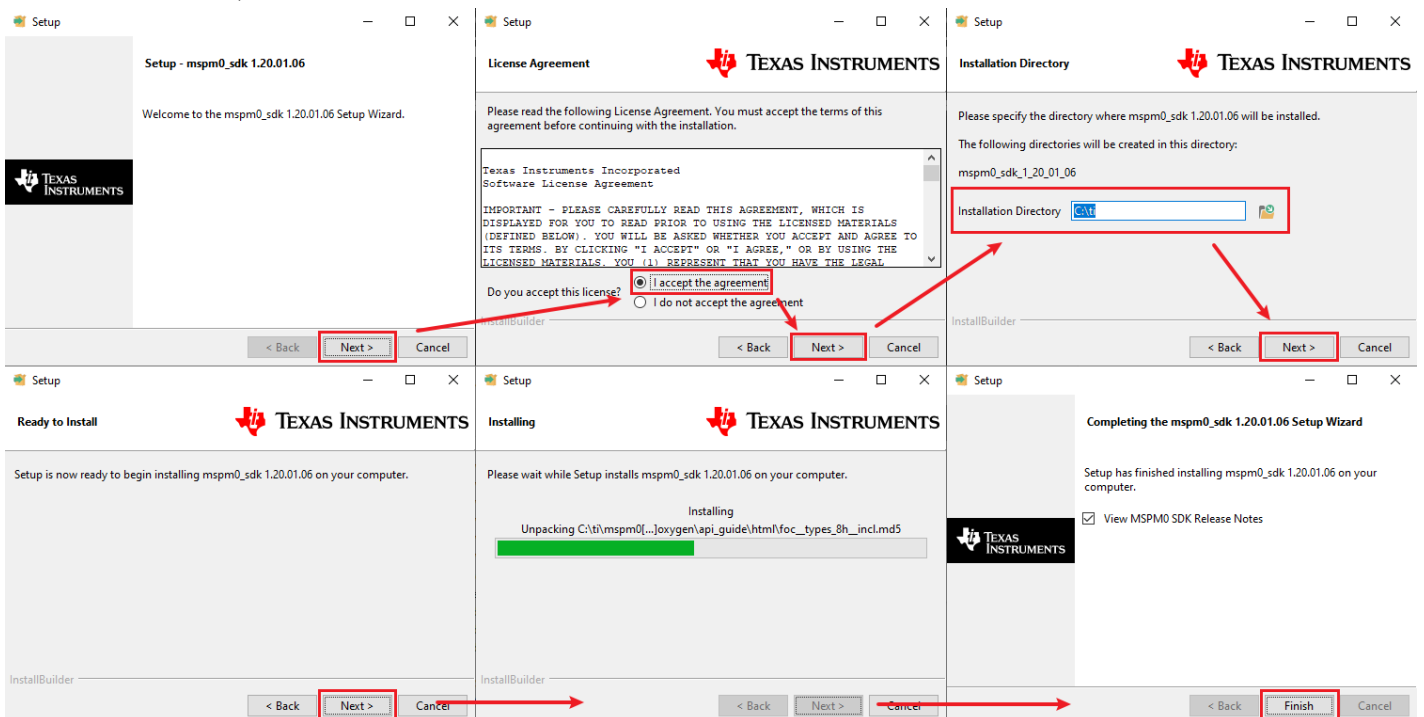


图 3-4. MSPM0-SDK 分步安装

### 3.2.2 MSPM0-SDK 简介

SDK 安装目录中有五个文件夹，如图 3-5 所列。本节简要介绍了所有五个文件夹。

- Docs 文件夹：包含 SDK 的所有文档。
- Examples 文件夹：包含所有示例以供参考，这些示例可用于提供参考，还可以作为起点来加快应用程序开发。有关更多详细信息，请参阅 [MSPM0-SDK 示例指南](#)。
- Kernel 文件夹：RTOS 和 nortos 的构建文件，包括在示例工程中，可加快工程构建速度。
- Source 文件夹：包含 TI 和第三方库的所有源代码。
- Tool 文件夹：包含与 SDK 相关的所有工具，例如 SysConfig 支持文件、BSL GUI 和计量 GUI。

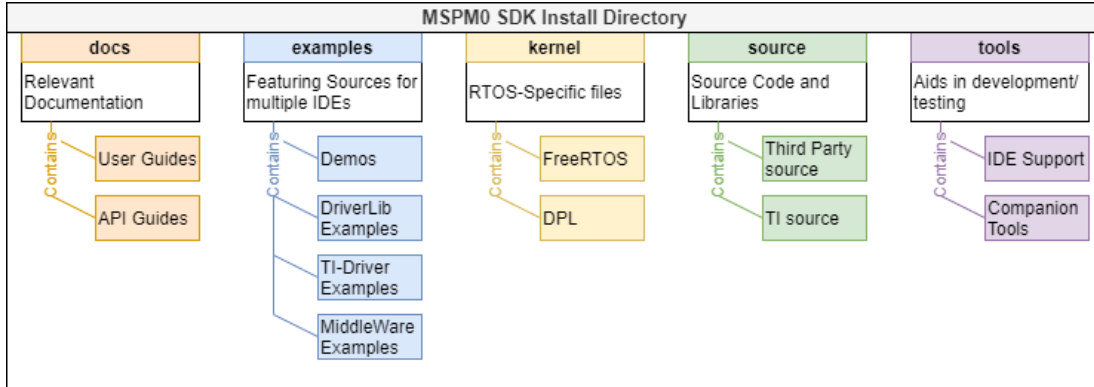


图 3-5. MSPM0-SDK 结构

最重要的文件夹是示例和文档文件夹。图 3-6 展示了 MSPM0-SDK 目录下文件夹的相关地址。

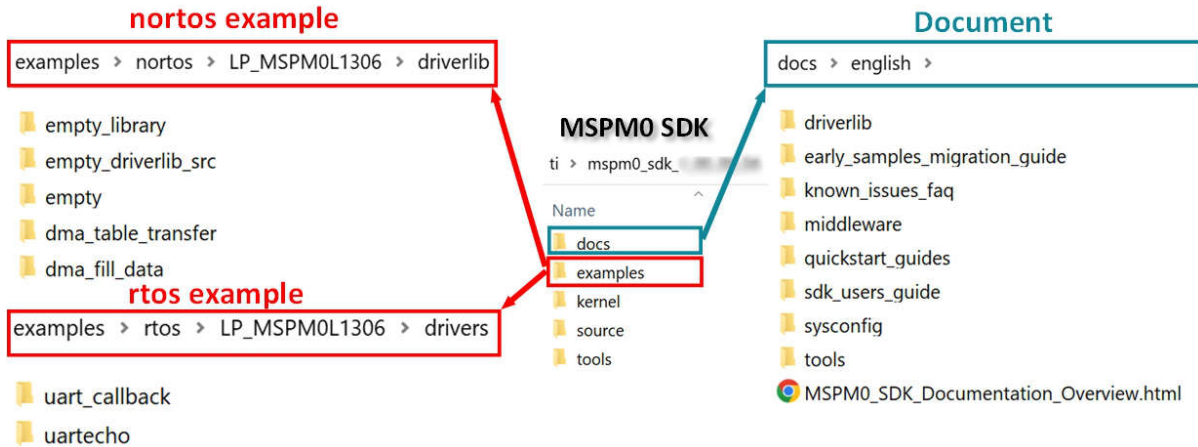


图 3-6. MSPM0-SDK 示例



### 3.2.2.1 文档文件夹简介

本节列出 MSPM0-SDK 中的所有文档。这里的内容基于版本 1\_20\_01\_06。

#### MSPM0 SDK 文档：

- [发布说明](#)：列出了 MSPM0-SDK 的所有内容和发布说明。
- [快速入门指南](#)：提供了分步说明，以便快速开始使用 MSPM0 和 Code Composer Studio™ (CCS) Theia、CCS、IAR 或 Keil。
- [MSPM0 SDK 用户指南](#)：MSPM0-SDK 的主页。提供 MSPM0-SDK 示例指南和 SDK 概述的导航。
- [清单](#)：列出了 SDK 中的所有内容以及每个组件的每个安装文件路径。
- [早期样片迁移指南](#)：介绍了支持量产样片的建议工具版本，并为使用 DriverLib 和 SysConfig 配置文件的应用程序提供迁移指南。

#### DriverLib 文档：

- [DriverLib 指南](#)：为程序员提供了一个软件层，相比直接寄存器访问，这样可以更方便地进行高级编程。

#### TI 驱动程序文档：

- [TI 驱动程序概述](#)：TI 驱动程序是 TI MSPM0 产品系列的外设驱动程序集合。这些驱动程序围绕着便携式应用程序编程接口 (API) 来构建，从而可以在 MSPM0-SDK 产品系列之间实现无缝迁移。除非另有特别说明，否则 TI 驱动程序在设计上考虑了线程安全性，可以在实时操作系统 (RTOS) 应用程序内无缝运行。

#### 中间件文档（不同应用程序的库和协议栈）：

- [中间件主文件夹](#)
- [安全引导和更新](#)
- [有刷电机控制库](#)
- [DALI 库](#)
- [诊断库](#)
- [EEPROM 仿真库](#)
- [能量计量库](#)
- [GUI Composer 库](#)
- [有霍尔传感器的梯形电机控制库](#)
- [IQMath 库](#)
- [LIN 库](#)
- [无传感器 FOC 电机控制库](#)
- [SENT 库](#)
- [SMBBus 库](#)
- [步进电机控制库](#)
- [PMBus 库](#)

#### 第三方文档：

- [CMSIS DSP](#)：德州仪器 (TI) 支持 Arm® Cortex® 微控制器软件接口标准 (CMSIS)，这是 Cortex-M 处理器系列的标准化硬件抽象层。
- [IO-Link](#)：除了提供无缝通信和改进的互操作性之外，在需要维护和维修时，传感器和执行器级上的数字接口（例如，IO-Link）也具有优势。
- [Zephyr](#)：德州仪器 (TI) 已开始开发工作，以便支持 Zephyr 作为 MSPM0 器件的实时操作选项。



### MSPM0 工具文档：

- IDE 和编译器：MSPM0 支持以下 IDE：Code Compose Studio (CCS)、[IAR Embedded Workbench for Arm](#)、[Arm Keil MDK](#)。对于工具链，MSPM0 支持 [TI Arm Clang 编译器](#)和 [Arm GCC 工具链](#)。
- 代码生成：MSPM0 支持 [SysConfig](#)。

### 调试和编程工具：

- [XDS-110](#)：德州仪器 (TI) 的 XDS110 是一款适用于 TI 嵌入式处理器的新型调试探针 ( 仿真器 )。
- [MSP-GANG](#)：MSP Gang 编程器 (MSP-GANG) 是一款支持 MSPM0 以及 MSP430 和 MSP432 所有变体的器件编程器。
- [UniFlash](#)：UniFlash 是一个独立工具，用于为 TI MCU 的片上闪存存储器和 Sitara 处理器的板载闪存存储器编程。要访问快速入门指南，请点击[此处](#)。
- [BSL 主机](#)：MSPM0 器件随附安装了基于 ROM 且高度可定制的引导加载程序，该引导加载程序默认情况下支持通用异步接收器/发送器 (UART) 和内部集成电路 (I2C) 通信。有关更多信息，请参阅 [MSPM0 引导加载程序 \(BSL\) 实现](#)。
- [MSPM0 恢复出厂设置 GUI 工具](#)：调试子系统邮箱 (DSSM) 可用于执行器件批量擦除、执行恢复出厂设置和发送密码来解锁 SWD 接口。
- [Elprotronic](#)：除了德州仪器 (TI) 的 MSP430 和 MSP432、SimpleLink™ (CC)、C2000™ 和 TIVA™-C MCU 之外，Elprotronic 还提供多种支持 MSPM0 的硬件和软件编程工具。Elprotronic 支持 MSPM0，包括 MSP-GANG、FlashPro-ARM 和 GangPro-ARM。
- [Segger](#)：[SEGGER J-Link](#) 调试探针是当今使用非常广泛的调试探针系列。有关更多详细信息，请参阅[将 Segger 编程器与 MSPM0 结合使用](#)。
- [PEmicro](#)：[PEmicro Multilink](#) 和 [Multilink FX](#) 调试探针为 TI MSPM0 开发提供了一种经济实惠的紧凑型方法，可轻松高效地完成调试和编程。
- [Lauterbach](#)：所有 Arm 调试工具都支持 MSPM0。一般用于 Cortex-M 控制器，首选工具是 [μTrace for Cortex-M](#)。

### 3.2.2.2 示例文件夹简介

TI 为一个 MSPM0 子系列制造了一个 LaunchPad，其上搭载了包含该子系列的超集 MSPM0。同样的示例代码可在此 MSPM0 子系列中重复使用。nortos 示例位于地址 `mspm0_sdk_x_x_x_x\examples\nortos\LP_MSPM0xxxx` 下，RTOS 示例位于地址 `mspm0_sdk_x_x_x_x\examples\RTOS\LP_MSPM0xxxx` 下。本节简要介绍了一些关键示例类型。

- RTOS 文件夹：
  - 驱动程序：示例使用内核功能并提供基于 TI 驱动程序的更高级别硬件操作。对于驱动程序移植层 (DPL)，DPL 将驱动程序分离出来，从而可在不同 RTOS 内核或非 RTOS 内核之间进行迁移。对于 POSIX 层，该层将 RTOS 功能分离出来，从而可以迁移至新内核。
- Nortos 文件夹：
  - [DriverLib](#)：简单模块化示例，显示了 MSPM0 功能，包含进行了高级优化的低级驱动程序。
  - 中间件：针对不同应用 ( 包括汽车、电器、楼宇自动化等 ) 进行设计，提供库和协议栈。如需了解受支持的中间件列表，请参阅 [MSPM0-SDK 文档概述](#)。
  - 演示：集成的即用型演示，例如可与 TI 模拟器件配合使用的驱动程序代码示例。

作为参考，*Drivers* 和 *DriverLib* 下的示例支持表 3-3 中列出的所有平台。其他文件夹中的示例至少支持 CCS 平台。

**表 3-3. MSPM0 示例覆盖范围**

受 SDK 支持	平台 1		平台 2	平台 3
IDE	CCS		Keil	IAR
编译器	TI Arm-Clang	GNU Arm (GCC)	Arm 和 Keil 编译器	IAR Arm 编译器
RTOS	FreeRTOS			
代码示例	DriverLib 和 TI 驱动程序			

在 RTOS 示例级别，最重要的文件夹是 *Drivers* 文件夹，其中演示了基于 TI 驱动程序的外设控制。

在 nortos 示例级别，最重要的文件夹是 *DriverLib* 文件夹，其中包含基于 *DriverLib* 的外设示例代码。在 nortos 示例中，有四个空示例，用户可以使用它们来构建工程。表 3-4 列出了它们之间的差异。

**表 3-4. 空工程描述**

示例	类型	语言	使用 SysConfig	工程中的库文件
empty	工程	C	是	否
empty_cpp	工程	C++	是	否
empty_library	静态库 (调试后 <i>Debug</i> 文件夹下的 .lib 文件)	C	否	否
empty_driverlib_src (建议)	工程	C	是	是

有关 MSPM0 外设快速入门，还请参阅 [MSPM0 Academy](#)，这里提供了针对 MSP MCU 产品系列中各种主题的培训模块。

### 3.3 SysConfig 设置

SysConfig 包含了用于配置引脚、外设和其他系统组件的各种图形实用工具。SysConfig 有助于直观地管理、发现和解决冲突，以便用户有更多时间创建差异化应用。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。

#### 3.3.1 SysConfig 安装

如果用户选择 CCS 作为 IDE 平台，因为已经集成 SysConfig，所以可以忽略本节内容。

如果用户选择 Keil 或 IAR 作为 IDE 平台，请下载独立的 SysConfig 配置工具并按照步骤完成安装，如图 3-7 所示。

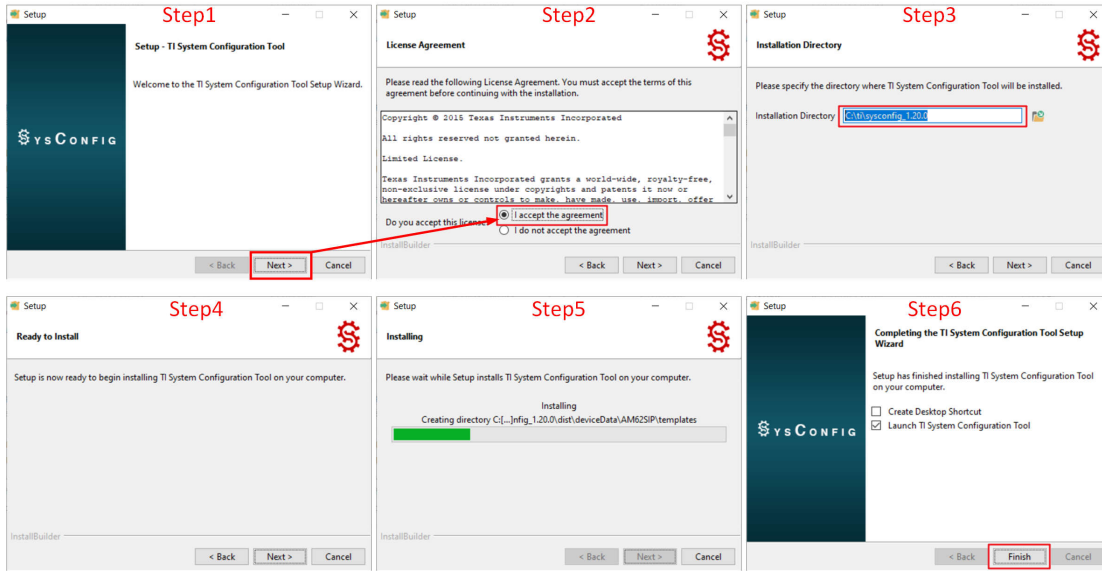


图 3-7. SysConfig 安装

### 3.3.2 SysConfig 介绍

本节简要介绍了如何使用 SysConfig。节 3.4 中的其他章节进一步介绍了如何将 SysConfig 与 IDE 一起使用。

- 在 *Peripheral Usage* 中添加所需的外设。
- 根据器件特定的技术参考手册，在 *Peripheral setting* 中设置参数。
- 调试后，外设可以直接生成 C 代码。

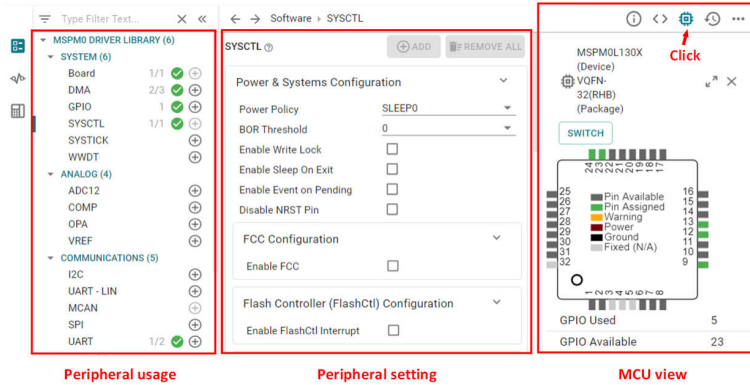


图 3-8. MSPM0 SysConfig

下一节介绍 SysConfig 中的组件，其内容摘自将 SysConfig 与 MSPM0 配合使用。

#### 3.3.2.1 基本概念

本节介绍 SysConfig 功能块和基本操作。

如图 3-9 所示，在打开 SysConfig 后会显示基本视图。SysConfig 有两个功能块：外设使用块，用于显示添加的外设和外设设置菜单入口。第二个功能块是外设设置，用于配置 MCU 外设。

点击屏幕右侧的按钮后，用户可以打开更多窗口。在工程构建后会显示生成的文件。用户可以单独点击文件，以了解在 SysConfig 上执行新设置后所做的更改。MCU 视图用于查看引脚分配和引脚资源，这也是 MSPM0 迁移的入口。

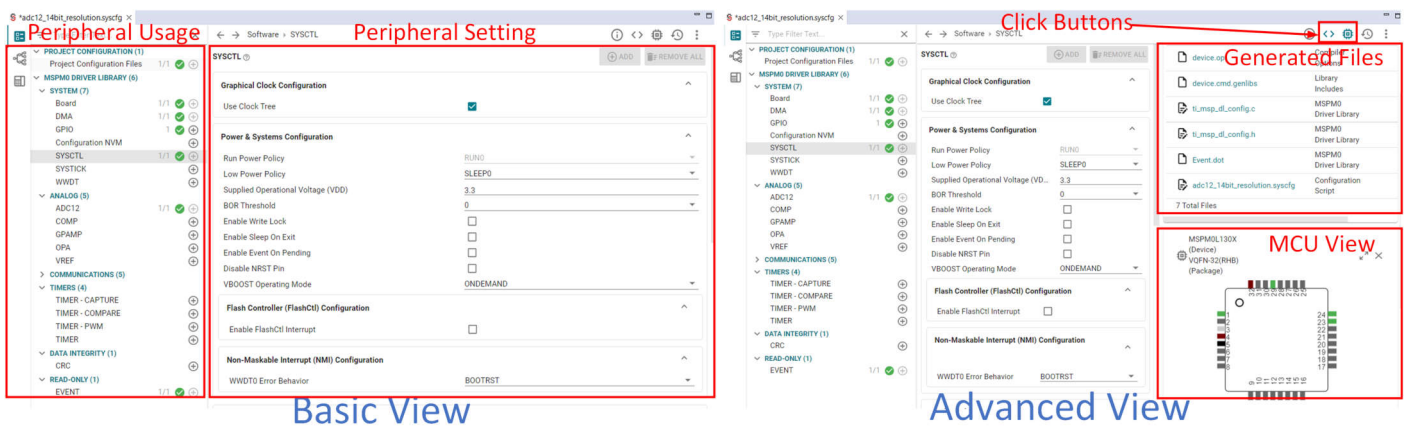


图 3-9. SysConfig 视图

SysConfig 的基本操作包括添加外设、删除外设以及查阅外设或功能描述。由于 SysConfig 是一个低级 MSPM0 外设设置 GUI，因此请参阅技术参考手册或外设示例，以便更好地理解。

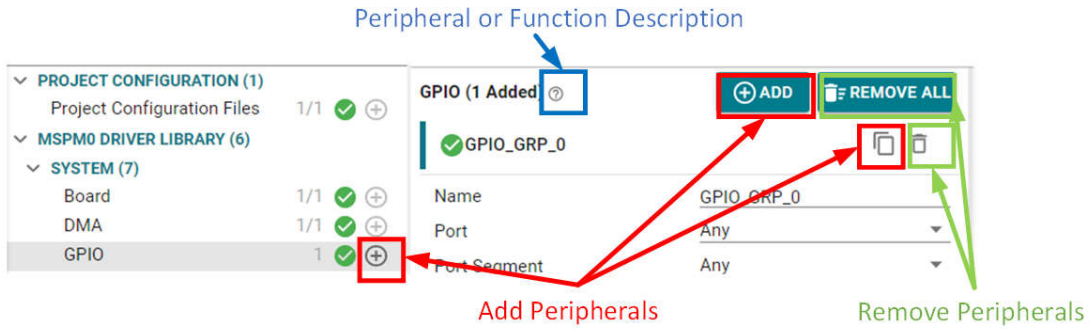


图 3-10. 基本操作

### 3.3.2.2 工程配置视图

这里是工程配置。该配置会影响总体 MCU 工程设置。本节介绍了一些重要特性。

- **File Generation** : 启用所有选择框后，SysConfig 会自动生成工程相关文件。我们建议您选择所有框。
- **Include Libraries** : 这里显示 SDK 中包括的所有库。启用选择框后，相关库会自动包括在工程中。
- **Select Device** : 由于该 SDK 示例适用于 LP，因此在迁移 MCU 后，可以更改此设置。

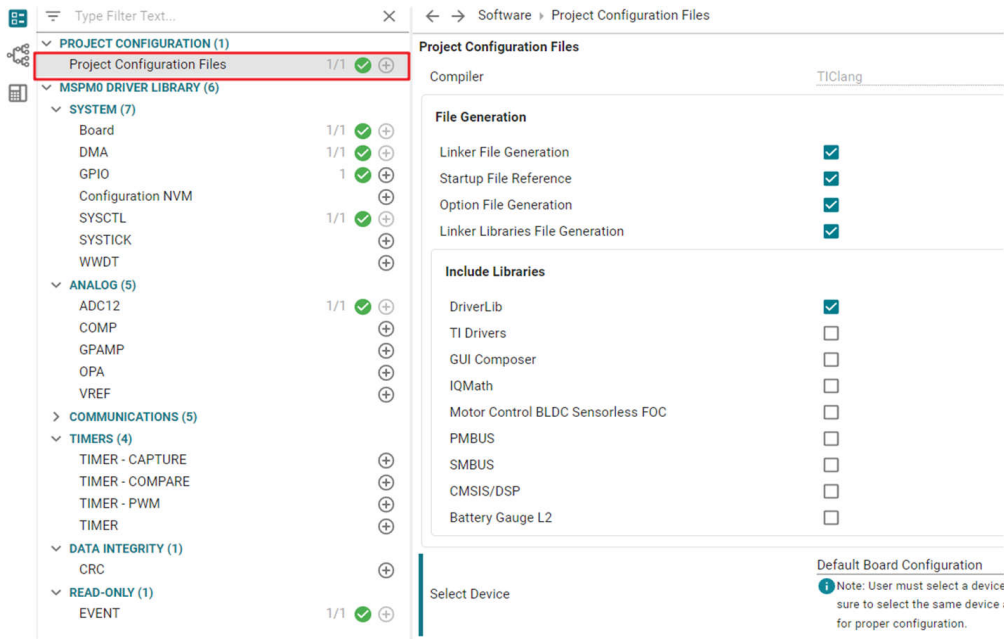


图 3-11. 工程配置

### 3.3.2.3 电路板视图

电路板视图用于配置总体 MCU 配置。

- **Debug Configuration**：对于某些 MSPM0，其配置将调试端口重新使用为外设功能。这是 SWD 已禁用的入口。
- **Global Pin Configuration**：
  - **Enable Global Fast-Wake**：这减少了从任何 GPIO 端口唤醒的时间。
  - **Generate Peripherals and Pin Assignments File**：如图 3-12 所示，启用后，在 **Debug** 文件夹中生成外设和引脚分配文件。

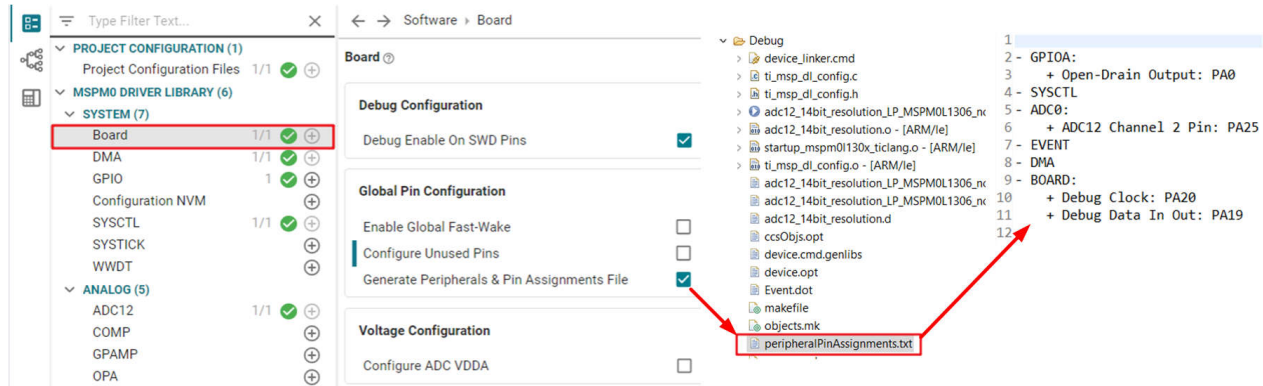


图 3-12. 电路板视图

### 3.3.2.4 NONMAIN 视图

NVM (NONMAIN) 用于配置与引导配置、安全性和引导加载程序相关的 MSPM0 保护区域。在 NONMAIN 中错误编程时，MSPM0 会中断。所以在进行配置之前，必须接受配置风险。由于此功能适用于高级用户，有关详细信息，请参阅 [MSPM0 NONMAIN 闪存操作指南](#)。

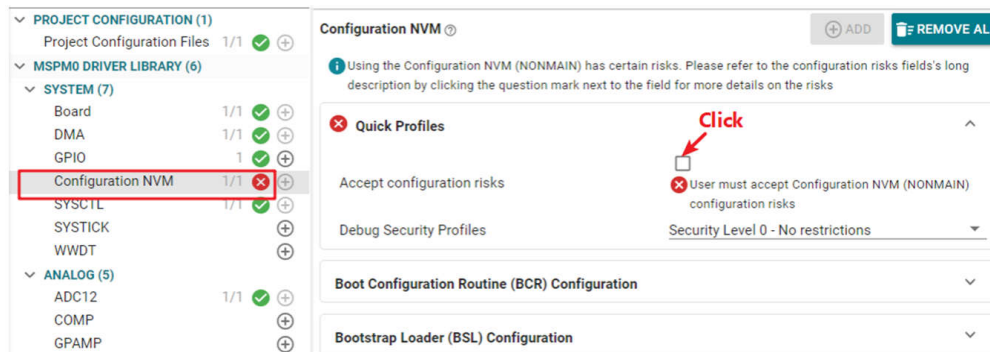


图 3-13. NONMAIN 视图



### 3.3.2.5 SYSCTL 视图

SYSCTL 用于配置 MCU 电源、时钟和复位模块。基本视图为菜单视图。本节介绍主要配置。

- Power and Systems Configuration :
  - Low Power Policy : 设置 MSPM0 的低功耗级别。
  - Disable NRST Pin : 对于某些 MSPM0 器件，NRST 引脚可重新用于外设功能。这是 NRST 引脚禁用的入口。

第二个视图是时钟树视图。用户可以使用时钟树功能，以图形方式配置器件的时钟，而不是使用 SYSCTL 菜单来配置，通过点击 SysConfig 左上角附近的信号图标可以找到这些菜单。在时钟树视图的左下方，用户可找到所有使用的时钟。有关每个时钟源的详细配置，请点击如图 3-14 所示的图标。

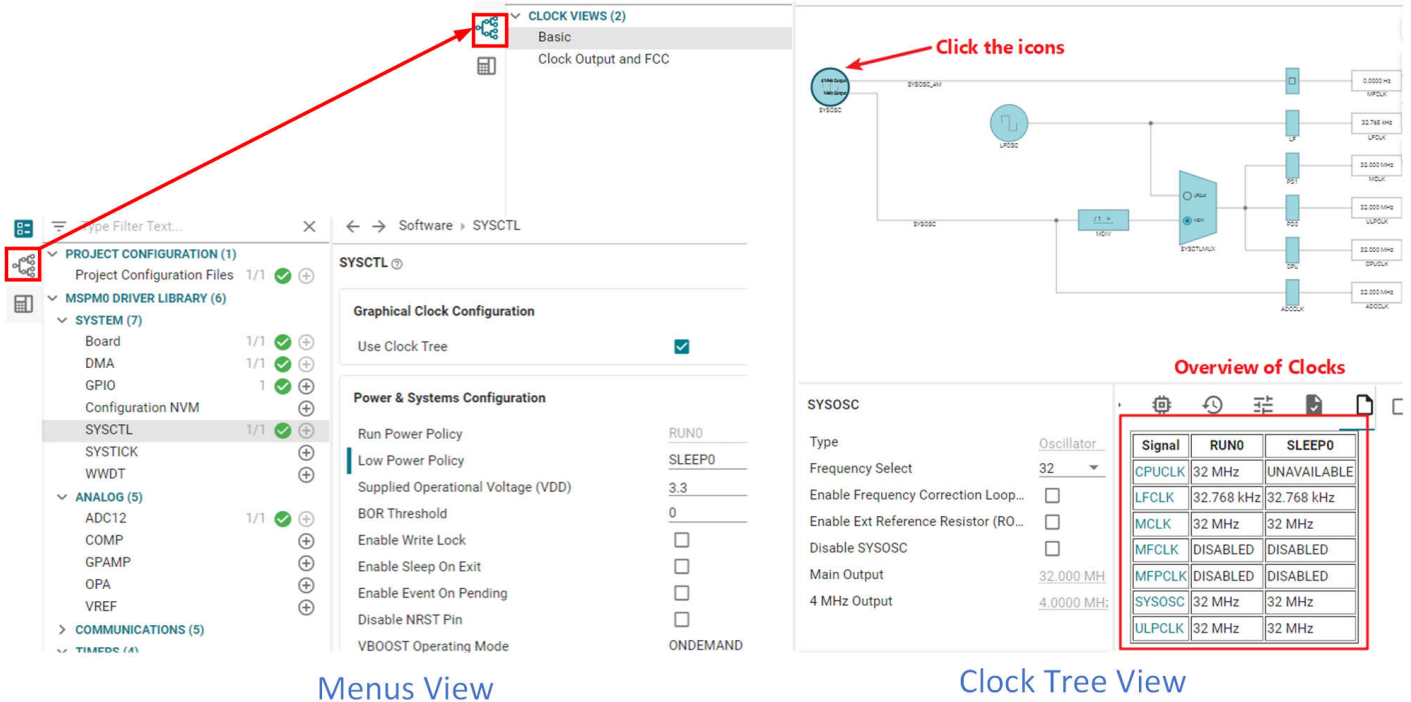


图 3-14. SYSCTL 视图

### 3.3.2.6 外设设置

本节介绍了外设设置，如图 3-15 所示。在添加说明之前先选择模块，即可打开软件模块说明。此说明包括模块功能的概述。如需更多信息，请参阅器件数据表或技术参考手册。

外设配置是以下配置的组合：

- Basic Configuration：基本外设配置
- Advanced Configuration：高级外设配置
- Interrupts Configuration：启用或禁用 MCU 中断
- Event Configuration：外设到外设触发配置
- Pin Configuration：启用上拉或下拉电阻器
- PinMux：为所选功能选择引脚输入或输出

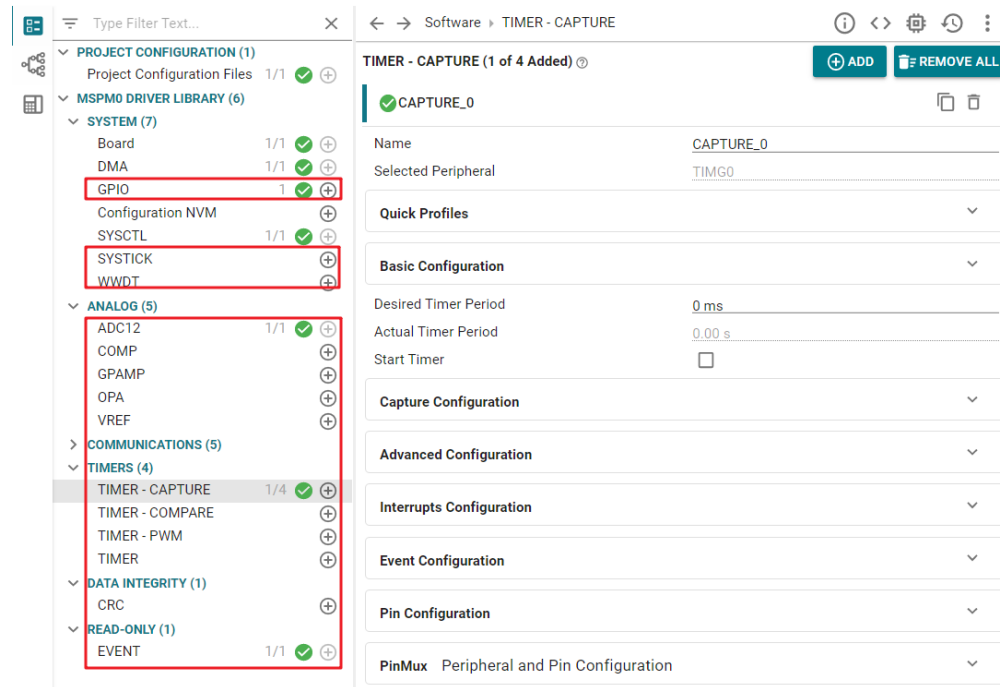


图 3-15. 外设视图



### 3.4 IDE 快速入门

MSPM0 系列支持使用三个 IDE 进行开发。建议将 CCS 作为首选选项，因为这是 TI 的 IDE，可与 MSPM0 兼容。表 3-5 列出并比较了三种不同类型的 IDE。

**表 3-5. 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	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

以下链接提供了不同 IDE 的相关指南。本节中的所有内容均摘自这些指南。

- [快速入门指南](#)
- [适用于 MSPM0 的 CCS IDE 指南](#)
- [适用于 MSPM0 的 IAR IDE 指南](#)
- [适用于 MSPM0 的 Keil IDE 指南](#)

### 3.4.1 CCS 快速入门

#### 3.4.1.1 CCS 安装

本节详细说明 CCS 安装的步骤和提示。务必将 CCS 保存在建议的地址和默认安装位置。

1. 下载 [CCS](#) (12.2 版本或更高版本)，开始安装，然后一直按 *Next*。

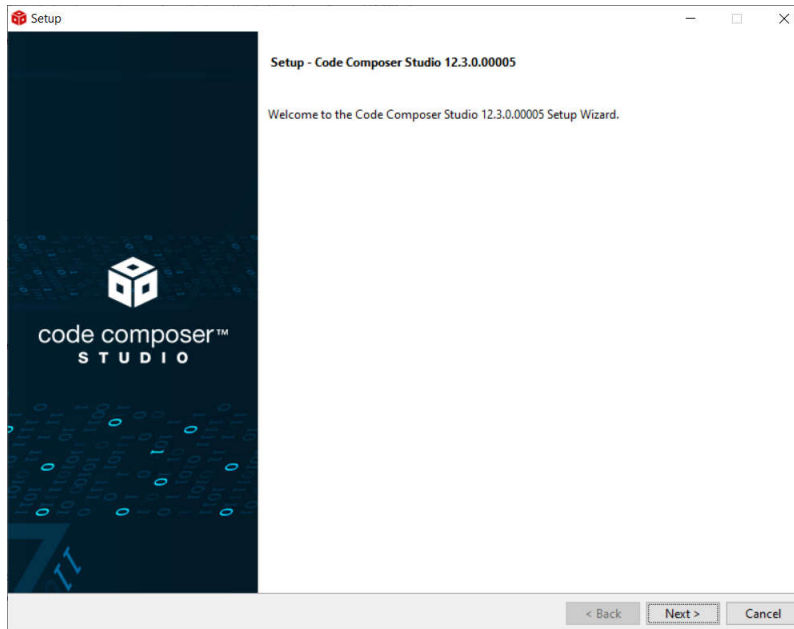


图 3-16. CCS 安装

2. 选择 MSPM0 支持元件。

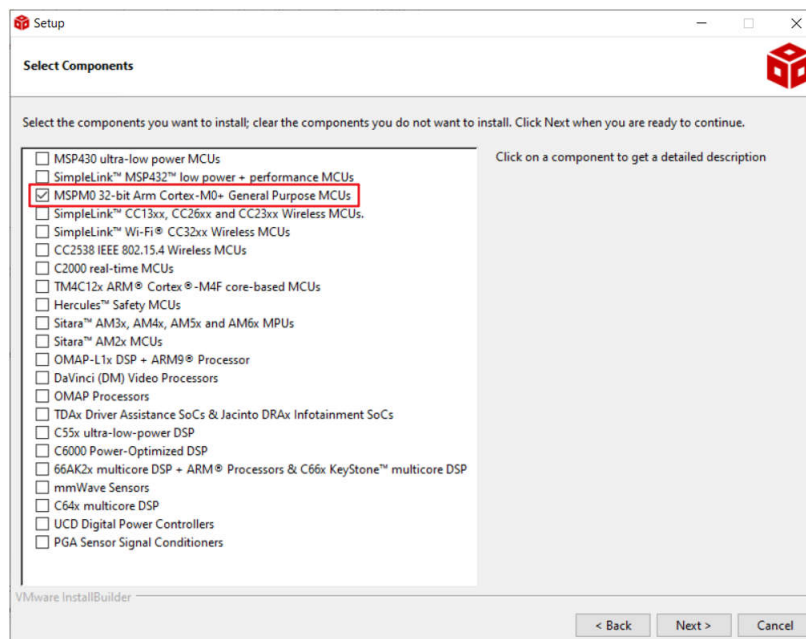


图 3-17. MSPM0 支持选择

3. 如果需要，选择 J-link。

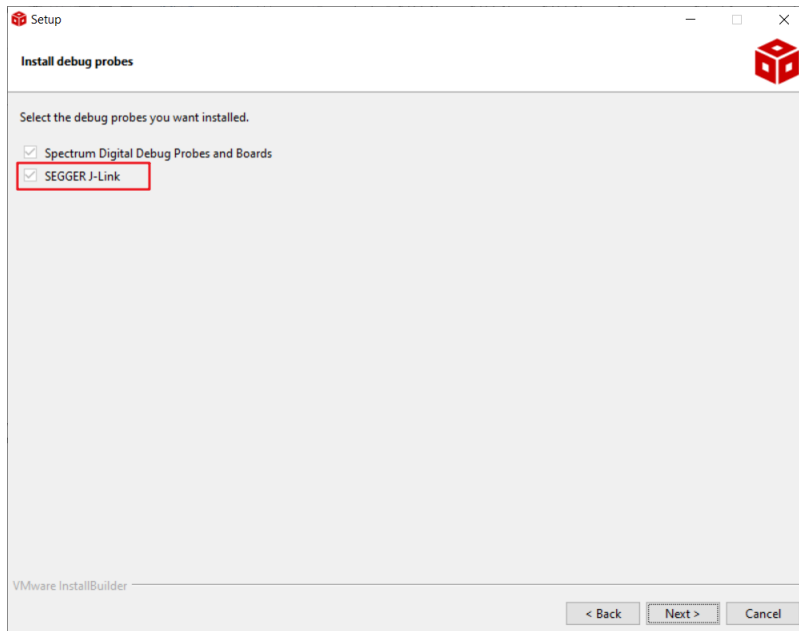


图 3-18. J-link 选择

### 3.4.1.2 环境设置

如果用户在以下默认地址文件夹中安装 CCS 和 SDK : (C:\ti\), 则导入示例时会自动加载相关的 SDK 和 SysConfig。可以跳过环境设置章节。

#### 3.4.1.2.1 SDK 支持设置

有关 SDK 介绍和安装信息, 请参阅节 3.2.2。如果 CCS 和 SDK 安装在自定义地址, 请按照以下步骤操作, 以便 CCS 成功加载 SDK。

1. 依次选择 *Window* → *Preferences*。
2. 由于 SDK 2.1.0.03 安装在 C:\ 盘中, 因此添加 C:\ 作为产品发现路径。
3. 刷新 *Discovered products* 窗口。系统会自动识别 SDK 2.1.0.03。
4. 点击 *Apply and Close* 按钮。新导入的工程会自动加载 SDK。

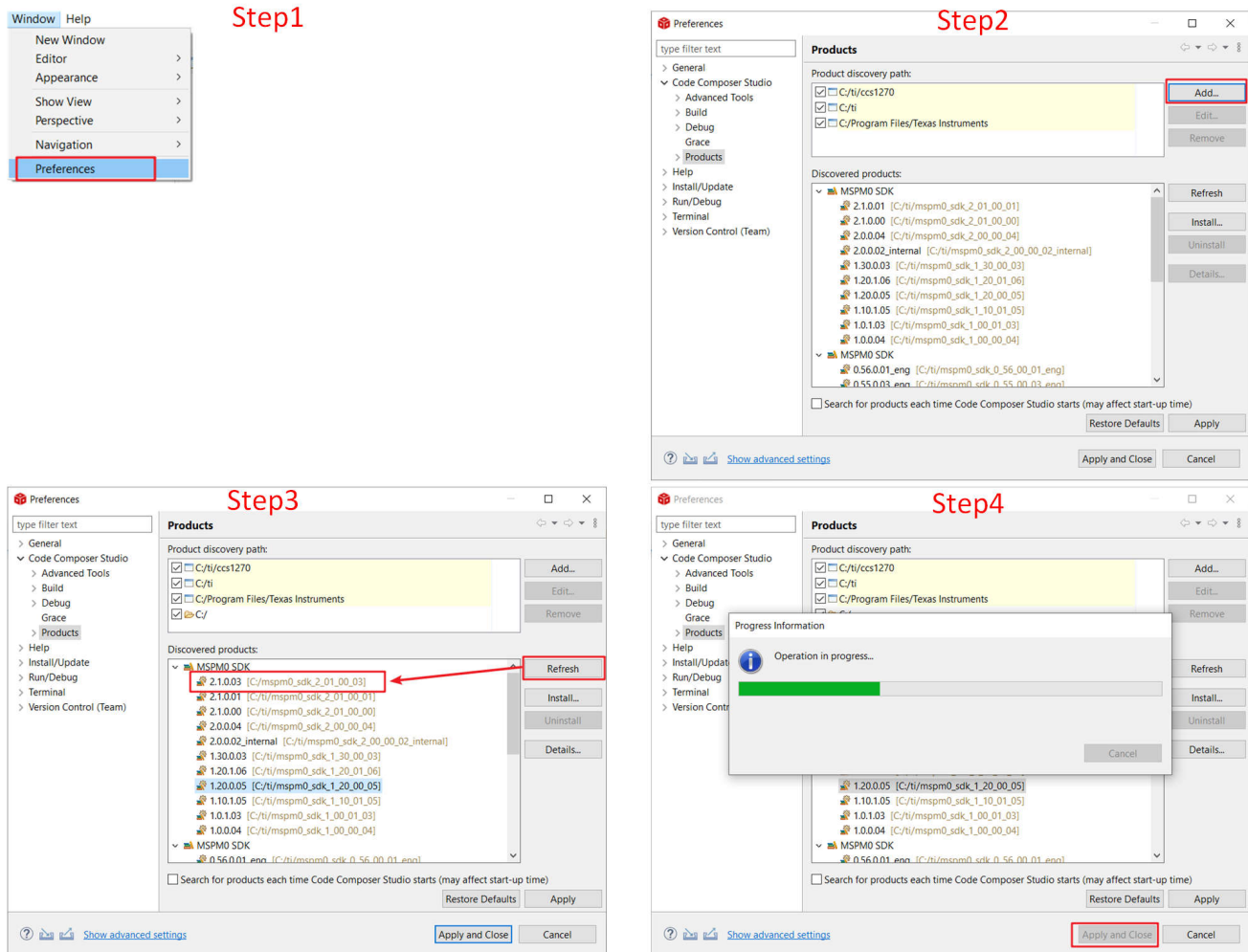


图 3-19. 加载 SDK 产品

导入示例后, 请按照步骤选择所需的 SDK 版本。当用户希望将示例从旧版本 SDK 迁移到新版本时, 也可以使用这些步骤。

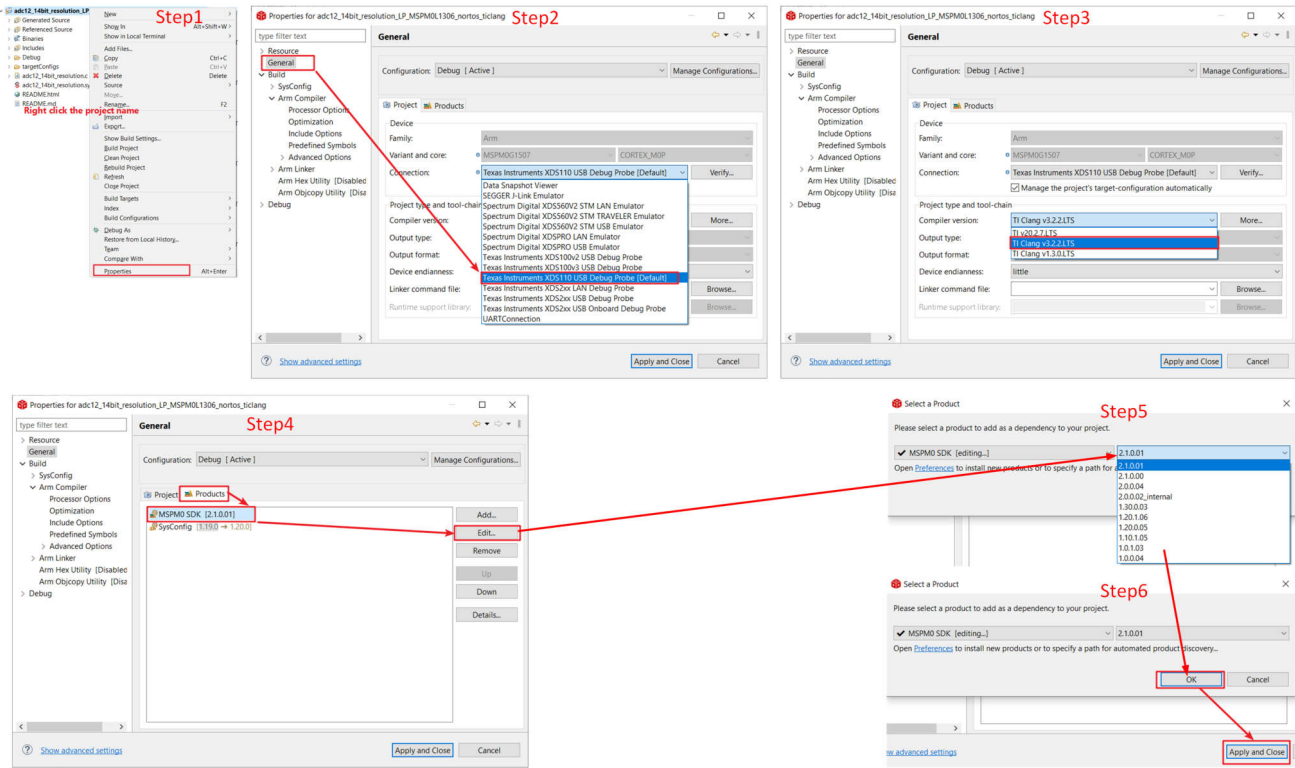


图 3-20. 选择或更改 SDK 版本

### 3.4.1.2.2 SysConfig 支持设置

由于 SysConfig 已随 CCS 一同安装，因此无需执行进一步操作。但是，旧 CCS 工程可能存在某些版本兼容性问题，或者在安装新的 CCS 版本时，打开的 SysConfig 会报告错误。请参阅节 3.4.1.2.1 中的 SDK 安装步骤，以便迁移到不同的 SysConfig 版本。有关 SysConfig 介绍和安装的信息，请参阅节 3.3。

### 3.4.1.3 导入 SDK 示例

打开 CCS。工作区是一个地址，导入的工程要复制到该地址。

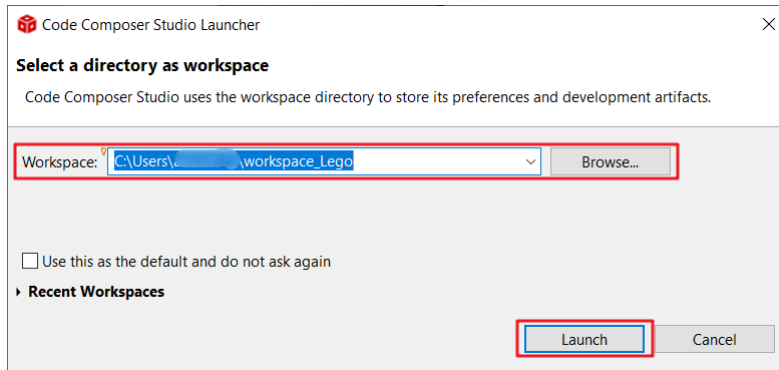


图 3-21. 选择 CCS 工作区

从已安装的 SDK 导入使用 TI-Clang 编译器的示例。

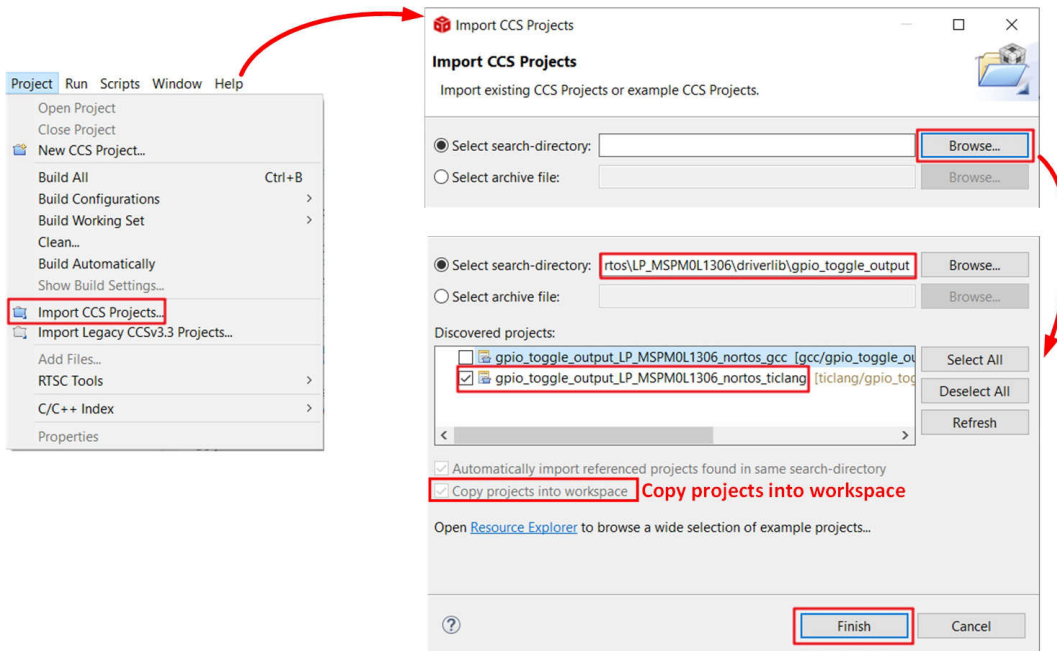


图 3-22. 导入工程

这是导入工程的视图。最重要的文件以红色标记。本节进行了简短的介绍。

- **SysConfig** 生成的代码：点击 **Build** 按钮，**SysConfig** 会在 **Debug\syscfg** 文件夹下生成代码。
- **.map** 文件：在 **Debug** 文件夹中，请参阅 **.map** 文件以了解有关存储器使用情况的更多信息。
- **Main** 函数 **.c** 文件：在文件中包括 **main** 函数。
- **.cmd** 文件：定义 MCU 存储器分配。在最新的 CCS 中，用户可以选择允许 **SysConfig** 自动生成分配。
- **SysConfig**：用于生成外设设置代码的 GUI 工具。

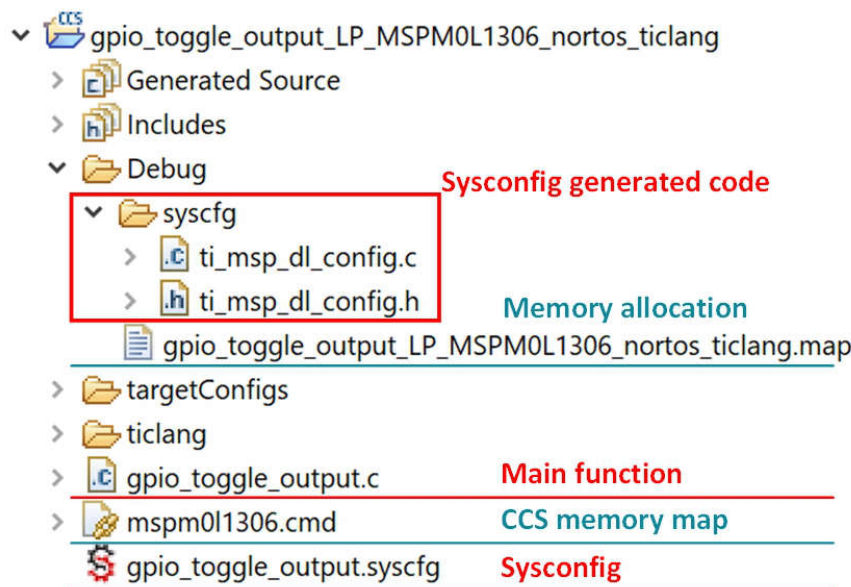


图 3-23. CCS 工程概述

#### 3.4.1.4 示例下载和调试

默认调试器选择为 XDS110。要选择 J-Link，请使用右键点击 **Project->Properties**，然后按照步骤选择 J-Link。



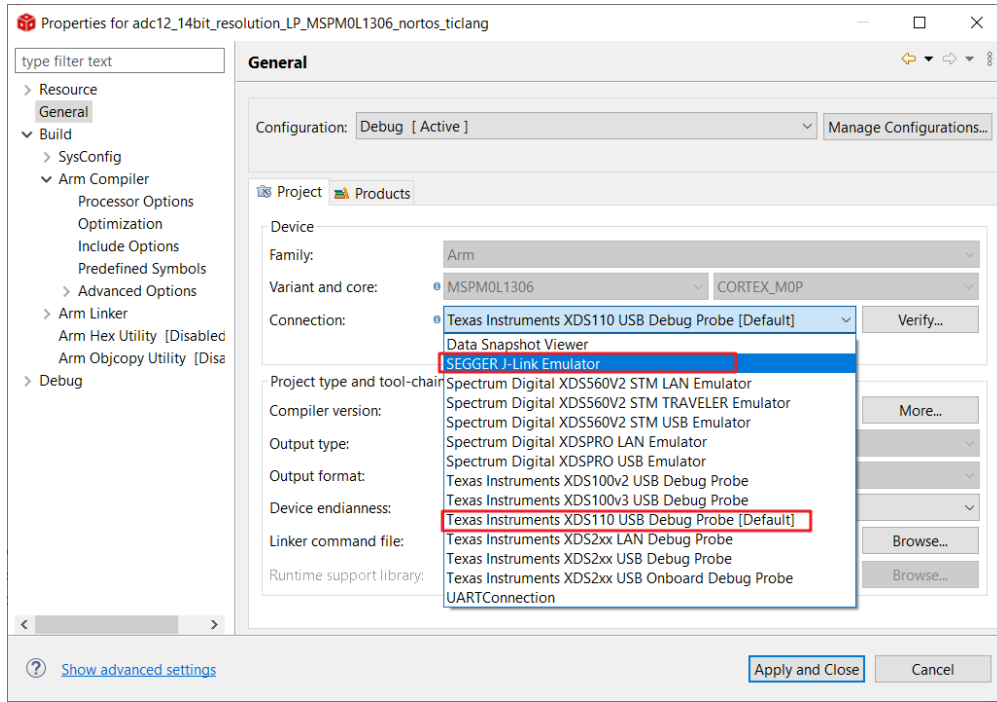


图 3-24. 更改调试器选择

点击顶部的 **Build** 按钮开始调试。之后，窗口会自动从 CCS 编辑视图切换至 CCS 调试视图。MCU 进入调试模式后，点击 **Run** 按钮以启用代码运行。

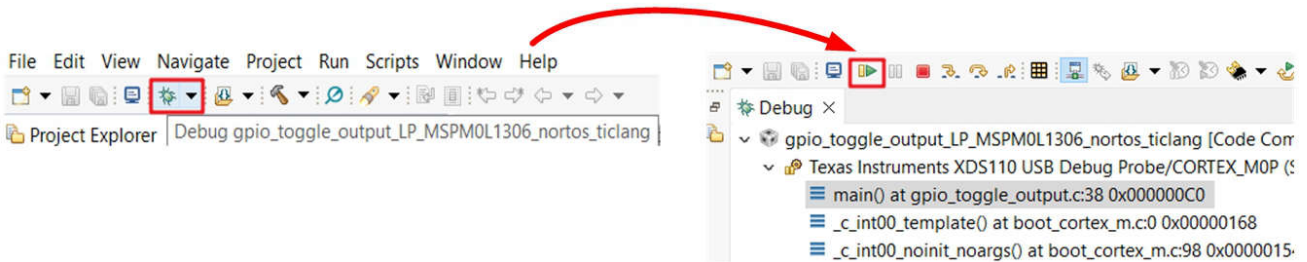


图 3-25. 调试代码

本节简要介绍了 CCS 功能。常用功能和含义如图 3-26 所示。

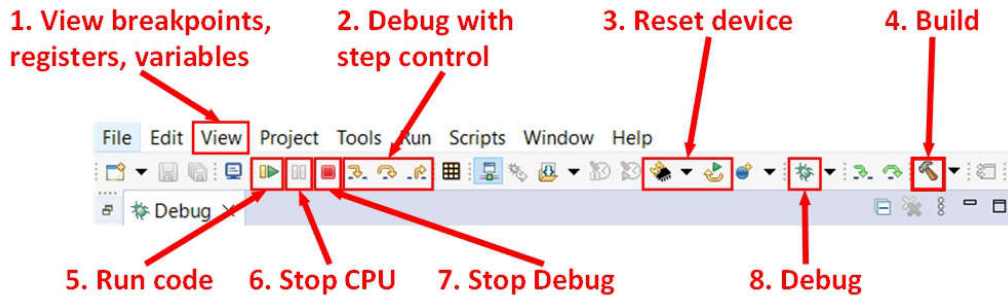


图 3-26. 常用调试功能

### 3.4.1.5 在 MSPM0 衍生产品之间进行迁移

此范围内的工程迁移需要更新特定于衍生产品的相关工程配置文件和设置，包括链接器文件、启动文件和包含的库。为了方便工程迁移，SysConfig 默认会生成工程配置文件，可通过工程配置模块来控制这些文件。

以下是基于 CCS 的迁移步骤：

1. 在 SysConfig 中，启用器件视图并点击 **SWITCH**。
2. 为 **Device**、**Package** 和 **CCS Launch Device** 选择 **New Values**，以便将工程配置迁移到新器件，然后点击 **CONFIRM**。
3. 确认新器件值后，SysConfig 会在工程配置模块中突出显示一个错误。用户必须在 **Select Device** 选项中选择新器件。确保器件选择与上一步中为 **CCS Launch Device** 选择的器件匹配。
4. 请注意，SysConfig 会突出显示迁移的任何冲突，例如不可用的引脚和外设。根据需要修复任何冲突，并将所有更改保存到 SysConfig 配置脚本。迁移现已完成，用户可以为新目标器件构建工程。

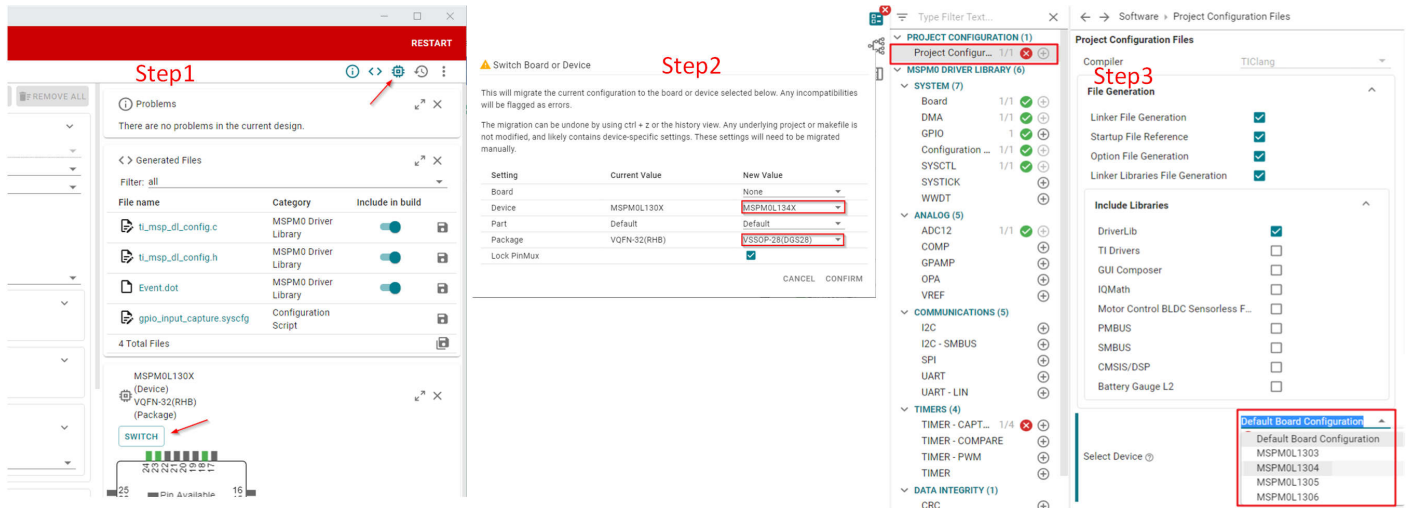


图 3-27. 在 MSPM0 衍生产品之间进行迁移

#### 3.4.1.6 生成十六进制文件

CCS 包括的实用程序可用于生成多种格式的输出对象，以便与编程工具配合使用。以下步骤说明了如何使用集成到 CCS 中的十六进制实用程序来支持十六进制文件。

1. 右键单击工程并选择 **Properties**。选择 **Build** → **Arm Hex Utility**，然后选择 **Enable Arm Hex Utility**。
2. 选择 **Output Format Options**。常用选项为 **Bin**、**Hex** 和 **TI\_TXT** 格式。选择所需的输出格式选项。
3. 如果选择了 Intel **HEX** 格式，则需要执行一个**额外步骤**来将存储器和 ROM 宽度指定为参数。在 **Properties** → **Arm Hex Utility** → **General Options** 中为存储器和 ROM 宽度选择 8。
4. 单击 **Build** 按钮后，会在调试文件夹中生成十六进制文件。



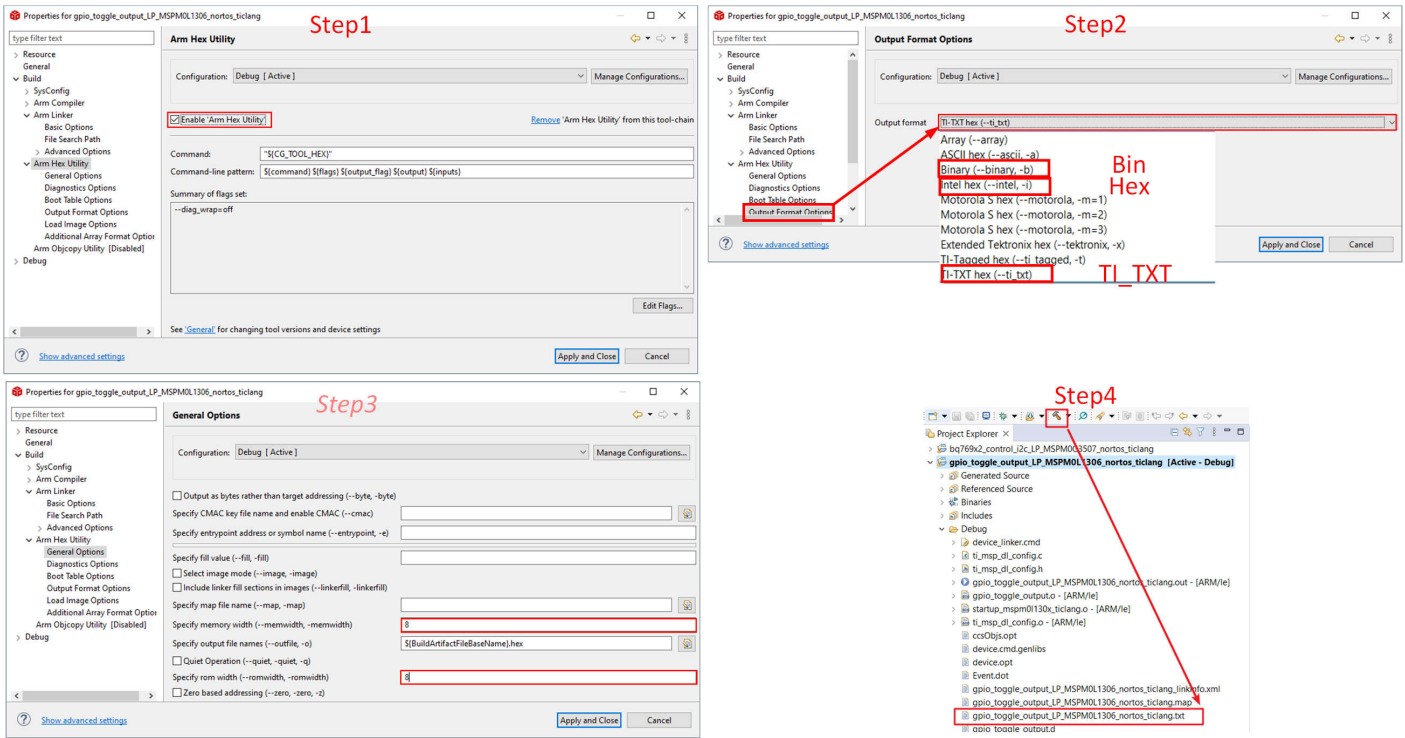


图 3-28. 生成十六进制文件

### 3.4.1.7 对 NONMAIN 编程

如果通过配置 NONMAIN 来更改引导加载程序或 MCU 安全设置，如节 3.3.2.4 所示，也要在 CCS 设置中启用 NONMAIN 擦除，如图 3-29 所示。否则，请保留默认设置。

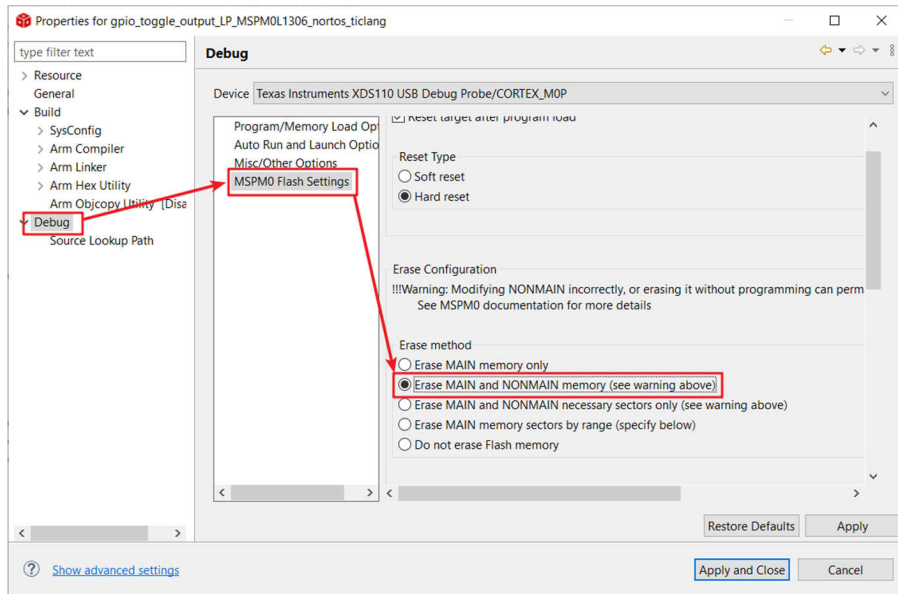


图 3-29. 对 NONMAIN 编程

#### 备注

在擦除和对 NONMAIN 编程时必须特别小心。如果操作不正确，例如在 NONMAIN 编程时断开连接，器件会锁定在永久不可恢复状态。

### 3.4.2 IAR 快速入门

TI 建议使用比 Arm 9.32.x 更高的 IAR Embedded Workbench 版本。较旧的版本不支持 MSPM0。

#### 3.4.2.1 环境设置

##### 3.4.2.1.1 SDK 支持设置

在 IAR 中，用户必须添加最新 MSPM0 SDK 版本。此步骤只需执行一次，或在更新 SDK 时执行。在 IAR 中，用户必须添加最新 MSPM0 SDK 版本。此步骤只需执行一次，或在更新 SDK 时执行。

- 第 1 步：在 IAR 中，依次点击 **Tools** → **Configure Custom Argument Variables**。
- 第 2 步：点击 **Global** 选项卡，然后点击 **Import**。
- 第 3 步：导航至 SDK 文件夹中的 **<MSPM0\_SDK\_INSTALL\_DIR>/tools/iar/**，然后打开 **MSPM0\_SDK.custom\_argvars**。
- 第 4 步：SDK 变量现在已安装在 IAR 中。点击 **OK** 关闭该窗口。

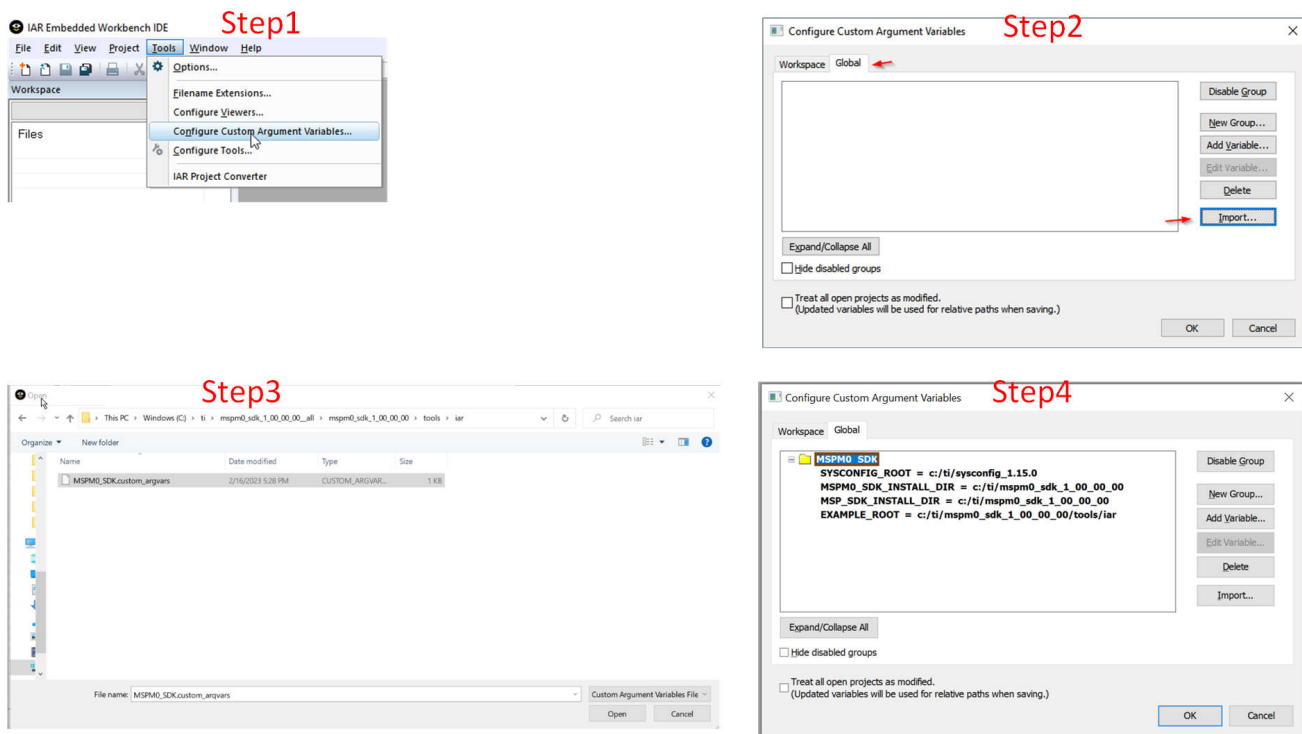


图 3-30. 将 MSPM0 SDK 添加到 IAR

#### 备注

请确保 MSPM0 SDK 路径和 SysConfig 路径与此 SDK 版本所需的位置和版本相匹配。如果安装了较早版本的 SDK，则确保将路径更新为当前版本。如果安装的 SysConfig 路径不正确或指向早期版本，请修改版本。

### 3.4.2.1.2 SysConfig 支持设置

此 SDK 包括 SysConfig 元数据的初始版本，可用于评估 MSPM0 SDK 的用户体验。

1. 在 IAR 中，从菜单中依次选择 *Tools* → *Configure Viewers*。
2. 点击 *Import*。
3. 导航至 SDK 文件夹中的 `<MSPM0_SDK_INSTALL_DIR>/tools/iar/`，然后打开 `sysconfig_iar_setup.xml`。
4. 独立版 SysConfig 与 `.syscfg` 文件关联。点击 *OK* 关闭窗口。
5. 仔细检查 `SYSCONFIG_ROOT` 自定义参数变量，确认是否正确指向 SysConfig 文件夹。

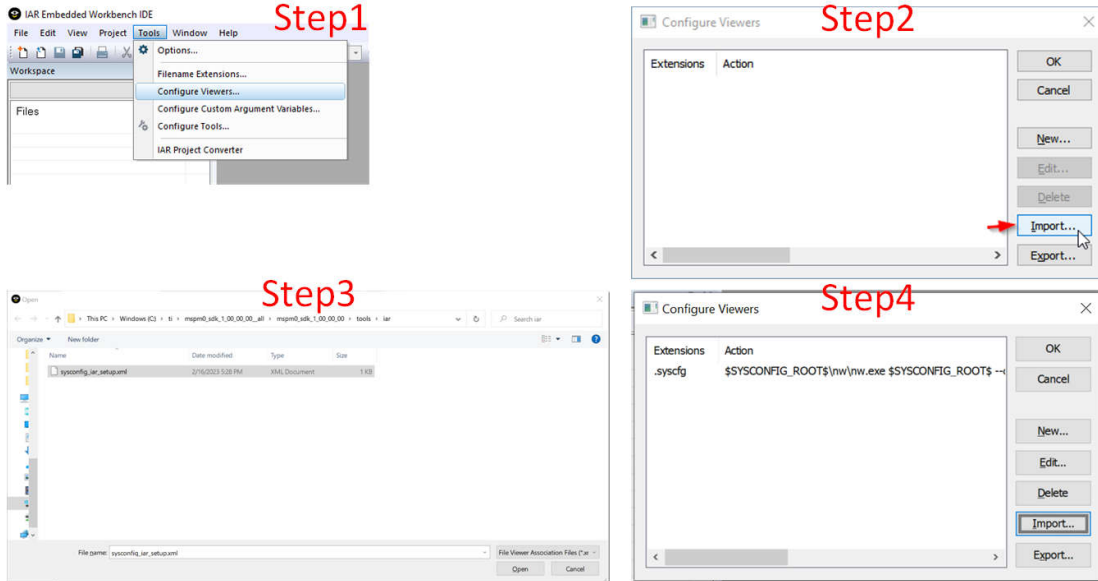


图 3-31. 为 MSPM0 安装 SysConfig

### 3.4.2.2 导入 SDK 示例

以下是从 SDK 导入 IAR 代码示例的步骤：

1. 在 IAR 中，从菜单中依次选择 *File* → *Open Workspace*。
2. 在位于 `<MSPM0_SDK_INSTALL_DIR>/examples/` 的 SDK 示例中导航至 IAR 文件夹，并打开 `.eww` 工作区文件。
3. 点击消息上的 *OK*。
4. 选择一个文件夹以安装示例。

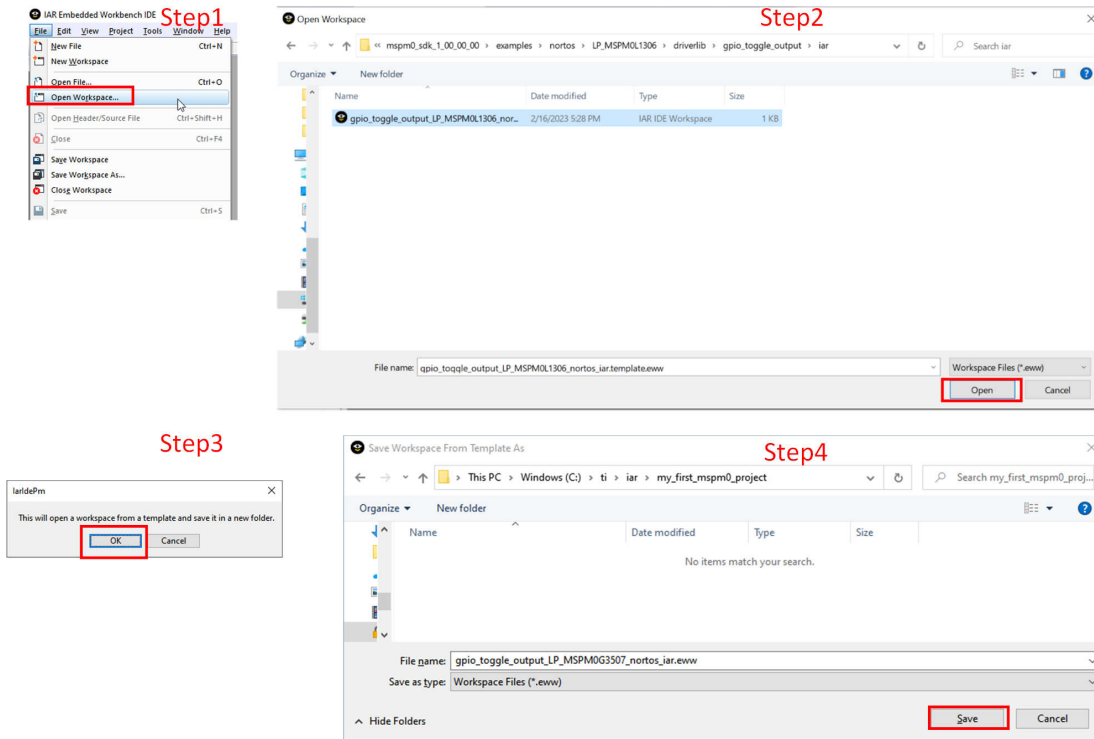


图 3-32. 导入 SDK 示例

以下是将 SysConfig 与 IAR 结合使用的简单说明。

1. 双击工程中的 .syscfg 文件。
2. 随即打开 SysConfig，用户可以在这里配置外设、IO 引脚和其他设置。
3. 保存更改，并切换回 IAR EWARM。编译代码示例。此时会更新 SysConfig Generate Files 文件夹中的文件。

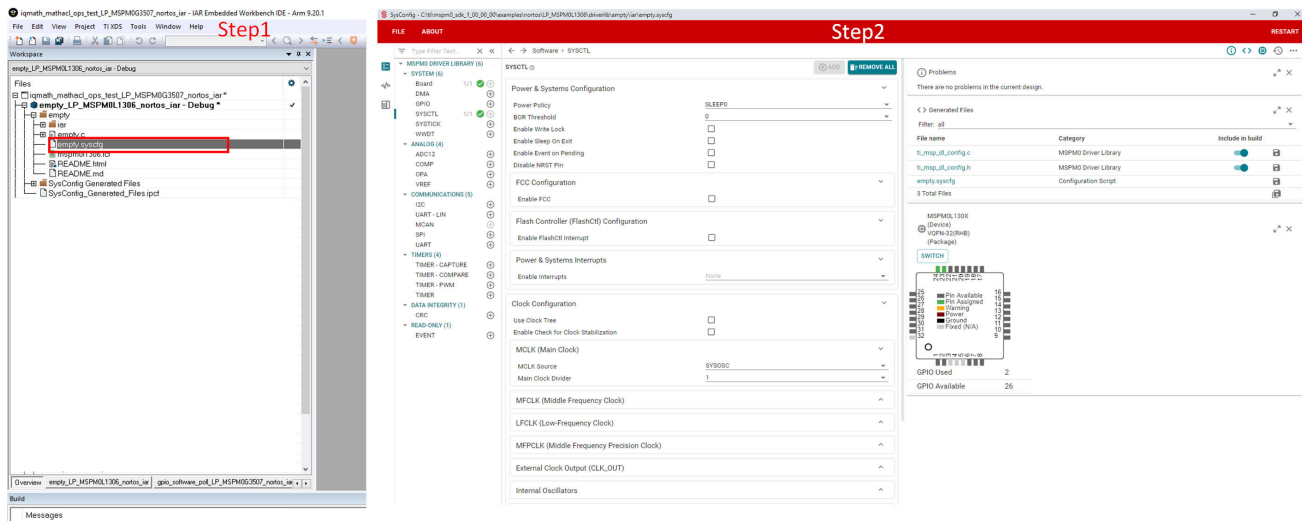


图 3-33. 将 SysConfig 与 IAR 配合使用

### 3.4.2.3 示例下载和调试

按照以下步骤在 IAR 下编译示例：

1. 要编译示例，在工程中右键点击并选择 **Make**。请注意，SysConfig 工程会在 **SysConfig Generated Files** 文件夹中自动生成文件。
2. 点击 **Download and Debug** 按钮下载代码。
3. 现在，开始调试代码。

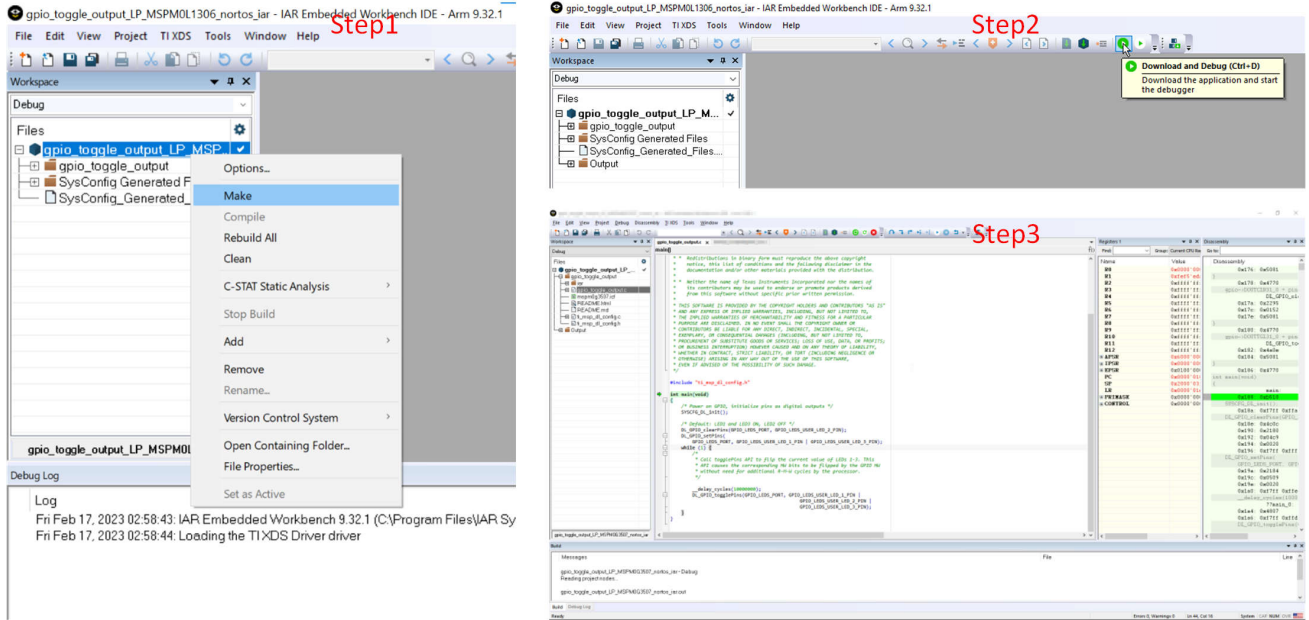


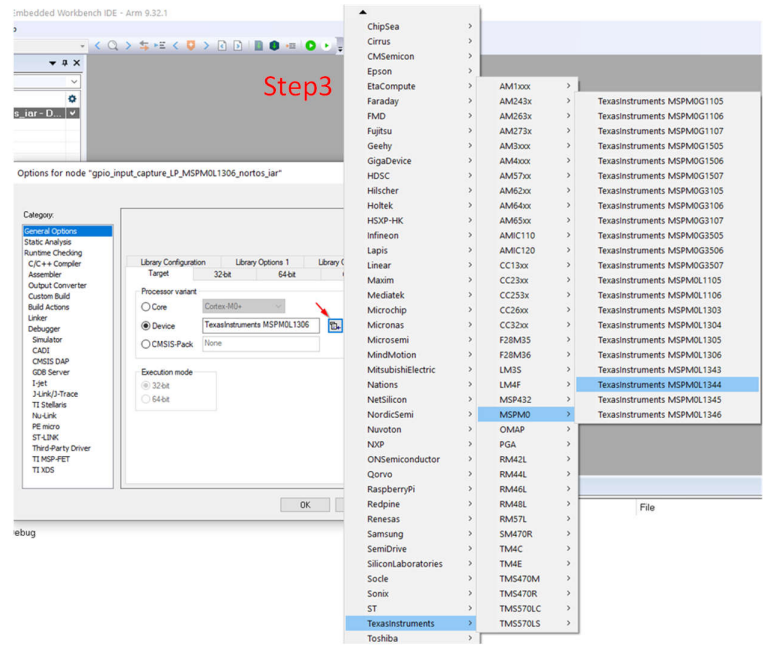
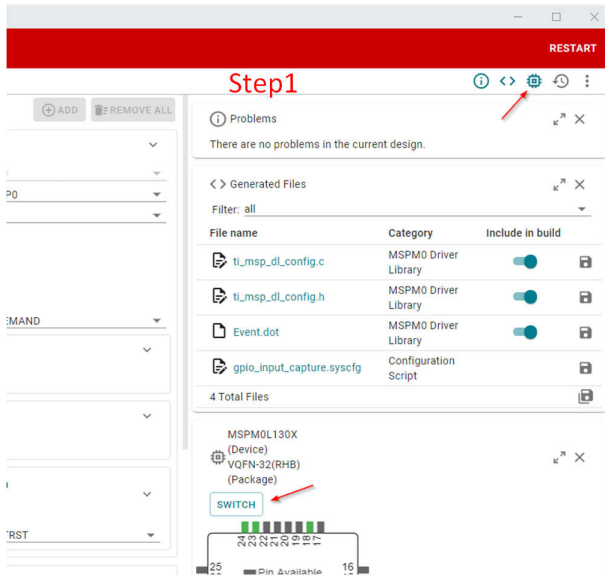
图 3-34. 下载和调试

### 3.4.2.4 在 MSPM0 衍生产品之间进行迁移

SysConfig 支持在 MSPM0 衍生产品之间更轻松地迁移。但在 IAR 上需要进行一些手动修改。以下是相关说明：

1. 在 SysConfig 中，启用器件视图并点击 **SWITCH**。
2. 选择新 MSPM0 器件的相应选项，然后点击 **CONFIRM**。请注意，SysConfig 会突出显示迁移的任何冲突，例如不可用的引脚和外设。根据需要修复任何冲突。
3. 在工程选项中，依次选择 **General Options** → **Target** → **Device**。选择 MSPM0 器件。
4. 在工程选项中，依次选择 **C/C++ Compiler** → **Preprocessor** → **Defined symbols**。根据所选器件添加器件定义。

第 2 步 : MSPM0 评估



**Step 2**

Switch Board or Device

This will migrate the current configuration to the board or device selected below. Any incompatibilities will be flagged as errors.

The migration can be undone by using ctrl + z or the history view. Any underlying project or makefile is not modified, and likely contains device-specific settings. These settings will need to be migrated manually.

Setting	Current Value	New Value
Board	None	None
Device	MSPM0L130X	MSPM0L134X
Part	Default	Default
Package	VQFN-32(RHB)	VSSOP-28(DGS28)
Lock PinMux		<input checked="" type="checkbox"/>

CANCEL CONFIRM

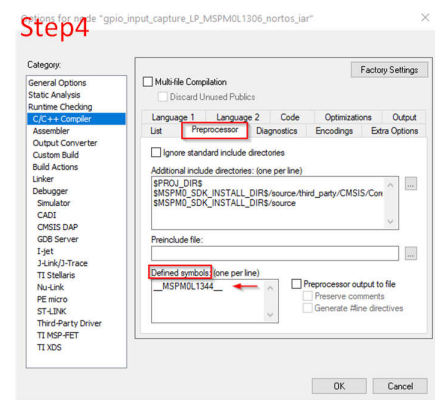


图 3-35. 在 MSPM0 衍生产品之间进行迁移



### 3.4.2.5 生成十六进制文件

以下是在 IAR 中生成十六进制文件的说明。依次点击 *Project* → *Options* → *Output Converter* → *Generate additional output* → *Output format* → *Texas Instruments TI-TXT*。也可以选择 Intel Hex 或其他格式。

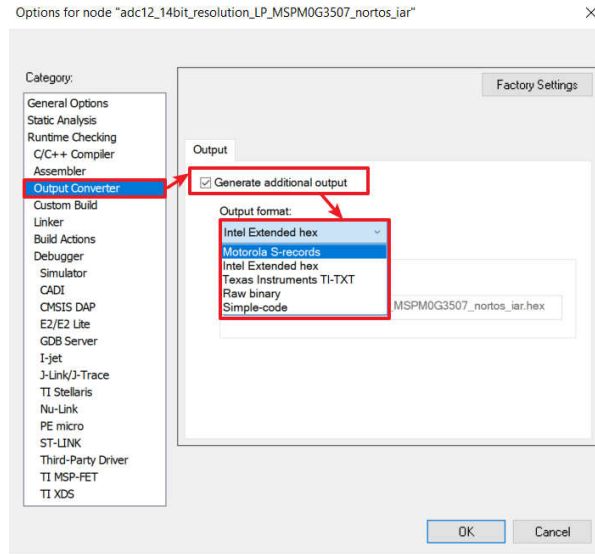


图 3-36. 生成十六进制文件

### 3.4.2.6 对 NONMAIN 编程

如果用户通过配置 NONMAIN 来更改引导加载程序或 MCU 安全设置，则用户还需要在 IAR 设置中启用 NONMAIN 擦除，如节 3.3.2.4 所示。请按照以下步骤操作，否则请保留默认值：

- 依次点击 *Options* → *Debugger* → *Download* → *Override default .board file* → *Edit*。选择第二个元素，然后点击 *Okay* ；
- 添加 `--non_main_erase` 作为额外参数。

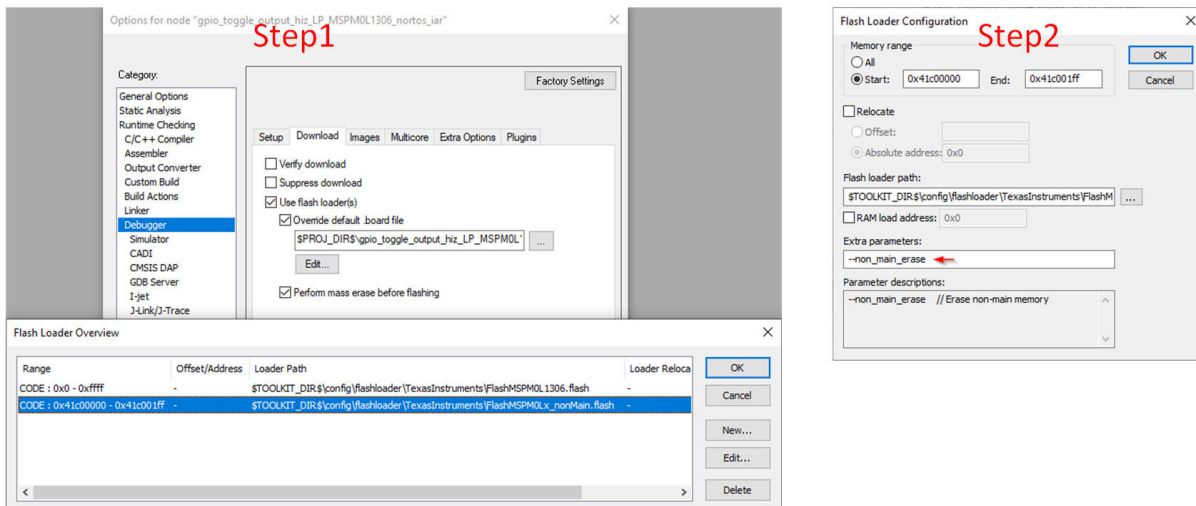


图 3-37. 对 NONMAIN 编程

#### 备注

在擦除和对 NONMAIN 编程时应特别小心。如果操作不正确，例如在 NONMAIN 编程时断开连接，则器件会锁定在永久不可恢复状态。

### 3.4.3 Keil 快速入门

#### 3.4.3.1 环境设置

与 IAR 不同，可以使用旧版本的 Keil，但要记得更新 MSPM0 CMSIS-Pack。

##### 3.4.3.1.1 MSPM0 CMSIS-Pack 设置

在开发 MSPM0 之前，需要先安装 Pack Installer。以下是更新 MSPM0 CMSIS-Pack 的步骤：

1. 在  $\mu$ Vision 中，通过快速指南打开 *Pack Installer*，或者选择 *Project* → *Manage* → *Pack Installer*。

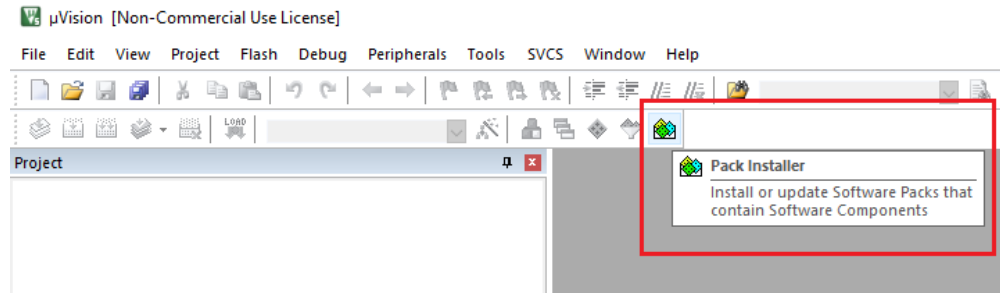


图 3-38. 打开 Pack Installer

2. 在 Pack Installer 中，在搜索文本框的左侧搜索 MSPM0。然后，屏幕上会显示相应的 MSPM0 系列。

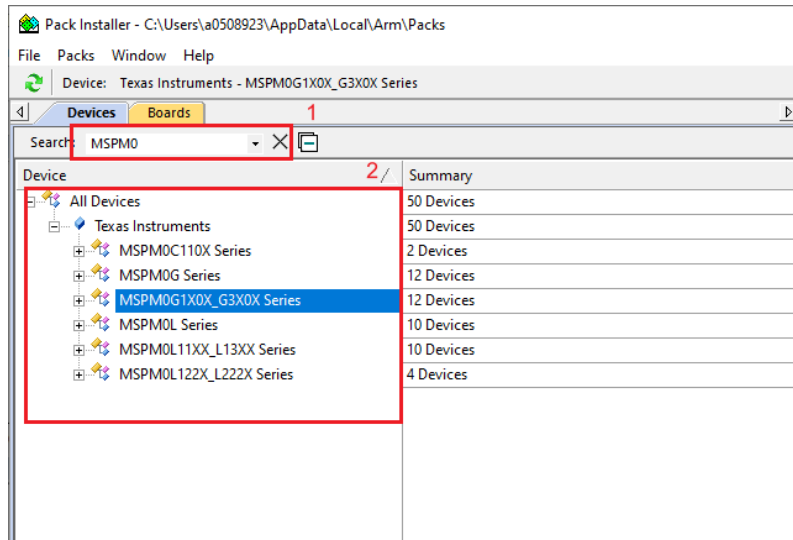


图 3-39. 搜索器件



3. 选择要安装包的器件。然后在右侧，安装特定于器件的包。

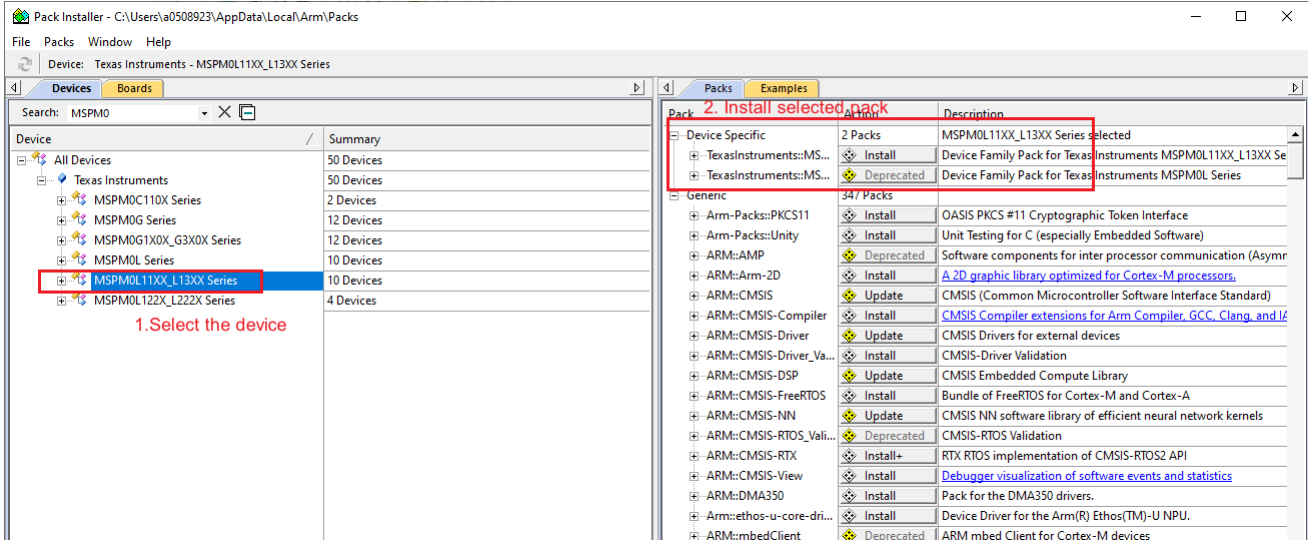


图 3-40. 安装器件包

4. 批准许可条款后，包即成功安装。

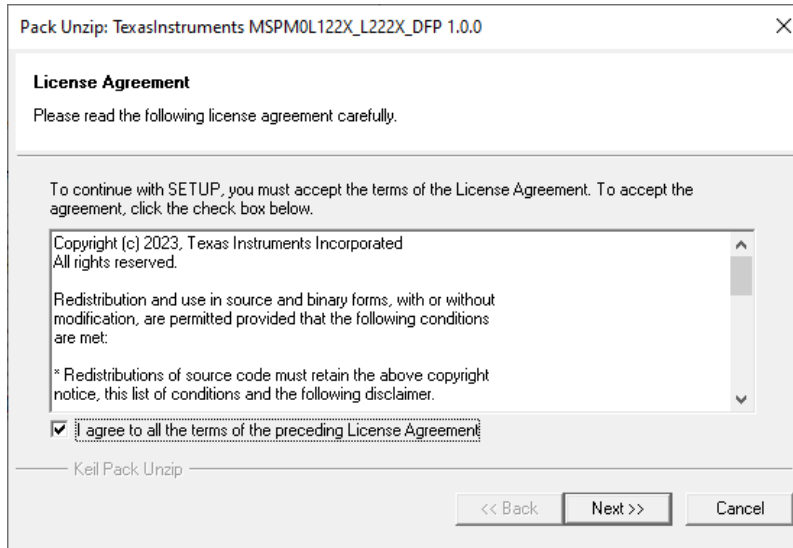


图 3-41. 批准许可证

### 3.4.3.1.2 Sysconfig 支持设置

如果需要 SysConfig，请按照以下步骤来启用。确保已提前安装 SysConfig 和 SDK。在这里，我们以 SDK v1.30 和 SysConfig v1.19 为例。

1. 导航到 SDK 文件夹 (...ti\mspm0\_sdk\_x\_xx\_xx\_xl\tools\keil)。编辑 `syscfg.bat` 中的 SysConfig 路径，以使其匹配下载的独立版 SysConfig 地址。

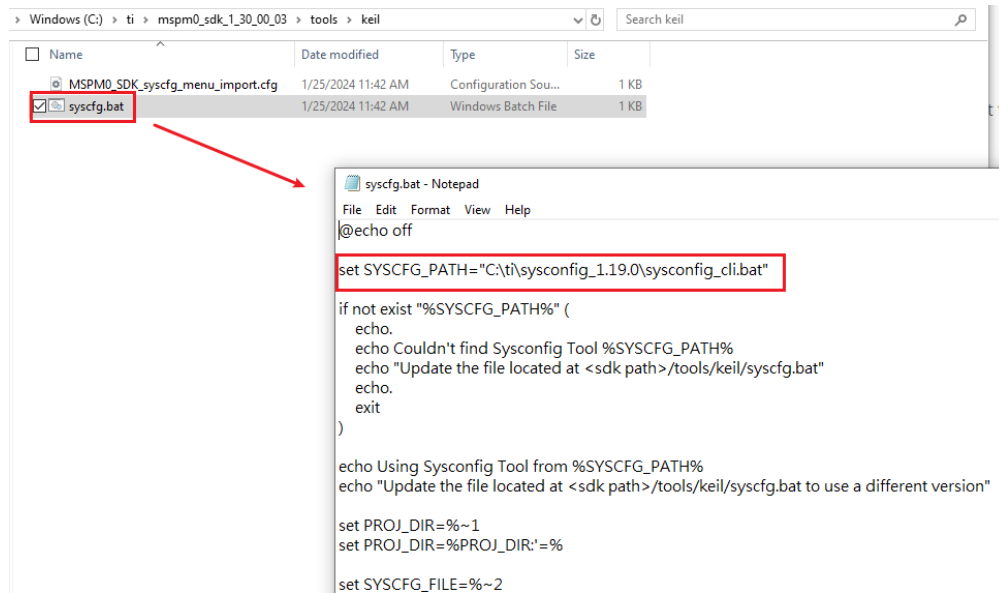


图 3-42. 编辑 syscfg.bat

2. 在同一文件夹中，打开另一个文件进行编辑。修改 SysConfig 和 SDK 版本与路径。

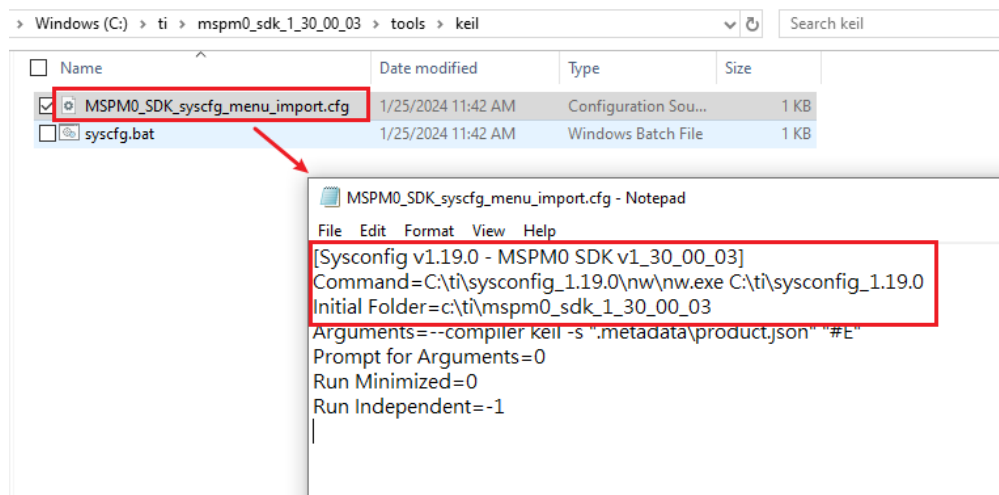


图 3-43. 编辑 MSPM0\_SDK\_syscfg\_menu\_import.cfg

3. 在 Keil 中，从菜单中依次选择 `Tools` → `Customize Tools Menu`。

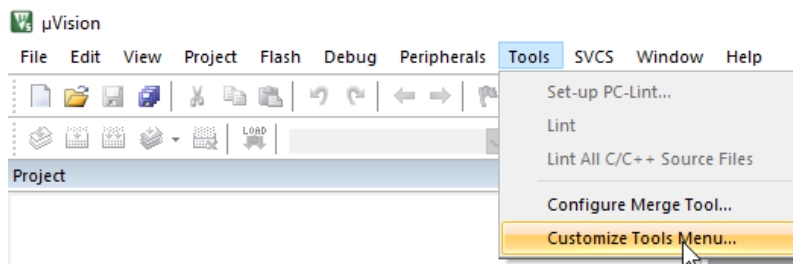


图 3-44. Keil 自定义工具

4. 将 MSPM0\_SDK\_syscfg\_menu\_import.cfg 文件导入自定义工具菜单。

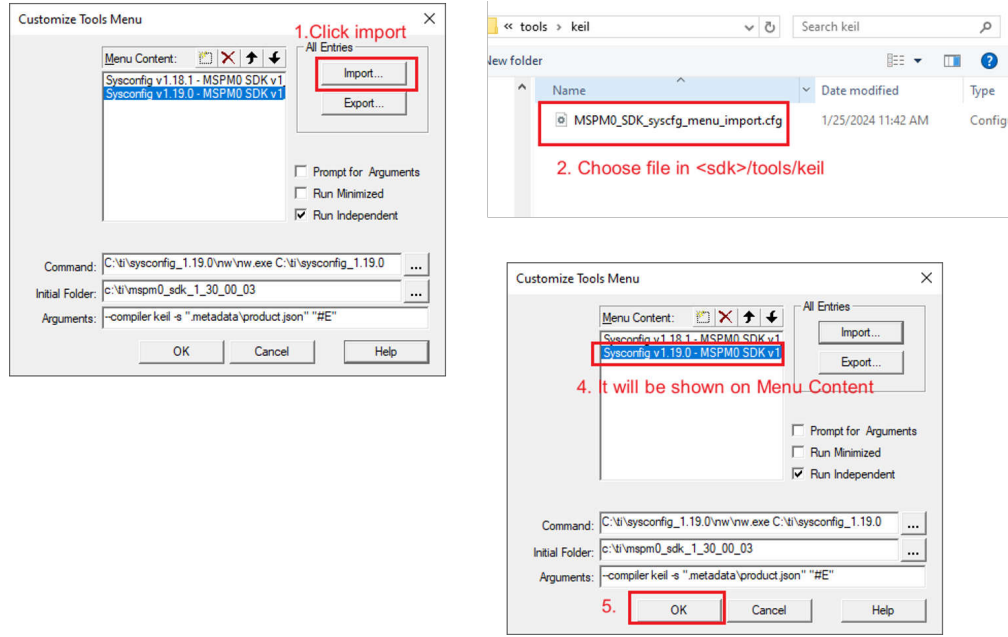


图 3-45. 导入 MSPM0\_SDK\_syscfg\_menu\_import.cfg 文件

5. 菜单上现在会显示 SysConfig 入口。您可以使用 SysConfig 在 Keil 上进行 MSPM0 开发。

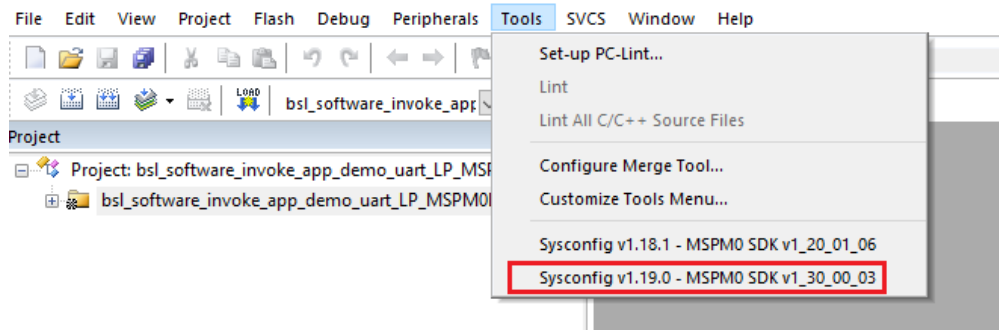


图 3-46. 完成 SysConfig 设置

### 3.4.3.2 导入 SDK 示例

以下指南说明了如何将 MSPM0 SDK 示例导入 Keil :

1. 在 Keil 中, 依次选择 *Project* → *Open Project*。

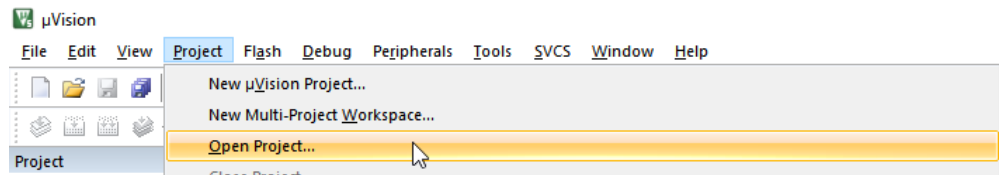


图 3-47. 打开工程

- 从 SDK 中选择一个演示工程。对于 nortos 示例，请使用 .uvprojx 工程文件。对于 RTOS 示例，请使用 .uvmppw 工作区文件。图 3-48 展示了一个示例。

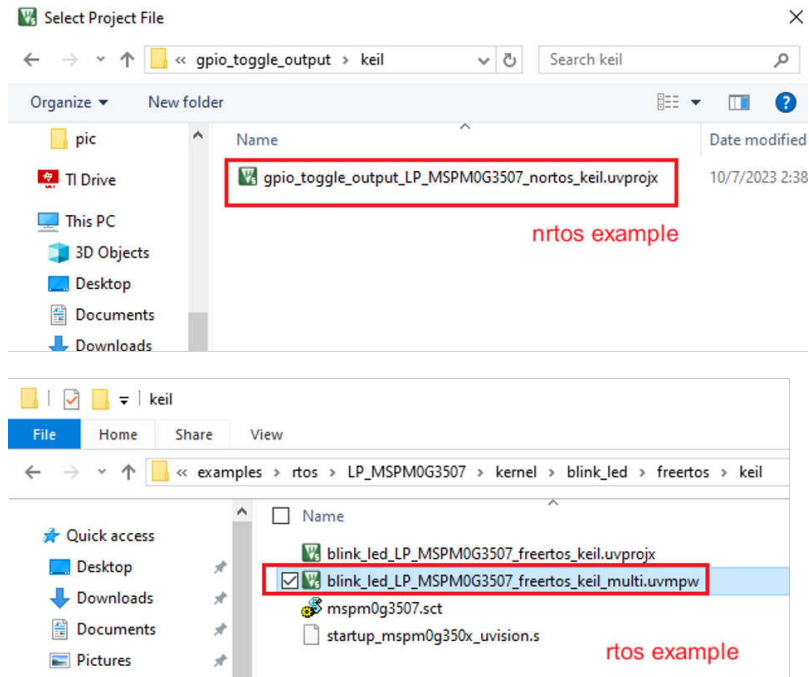


图 3-48. 选择 Keil 工程

- 要打开 .syscfg 文件，请双击 .syscfg 文件。然后选择 **Tools** → **Sysconfig v1.19.0 - MSPM0 SDK v1\_30\_00\_03**。 .syscfg 文件在单独的窗口中打开。

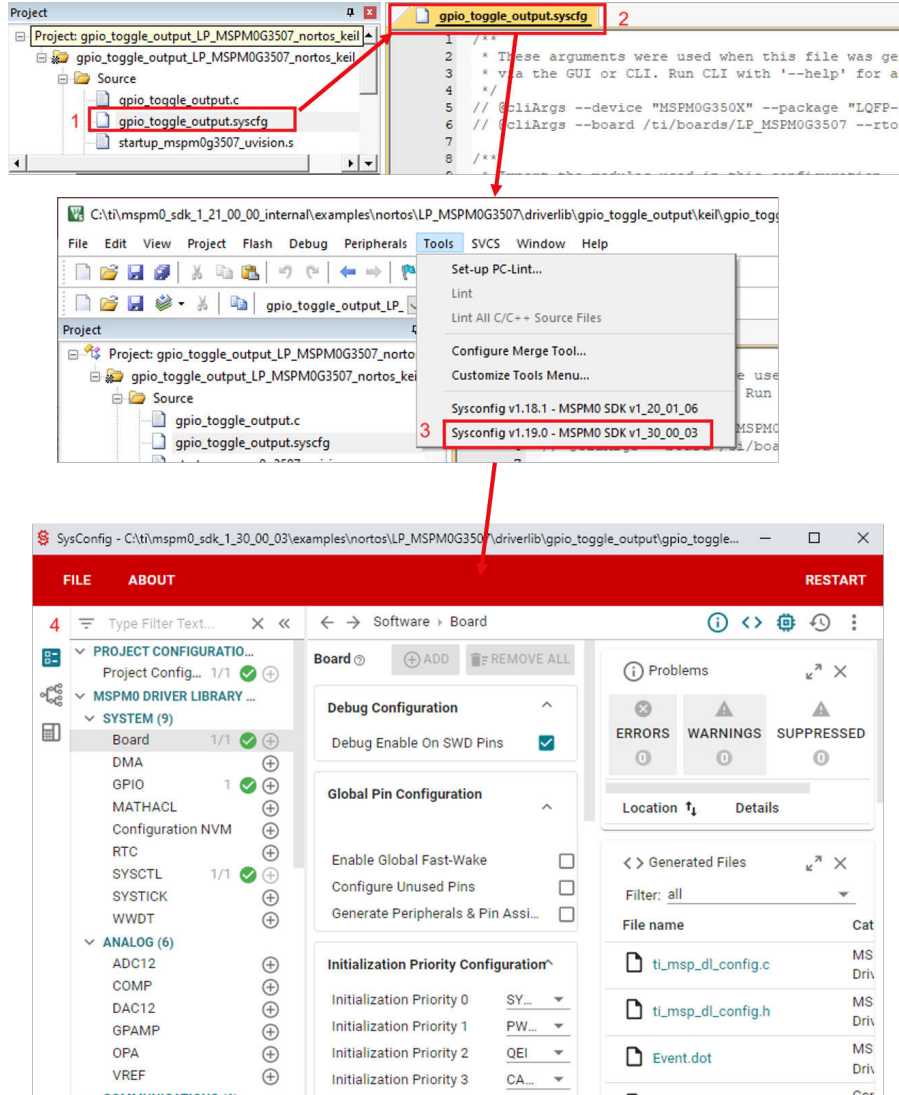


图 3-49. 打开 .syscfg 文件

### 3.4.3.3 示例下载和调试

以下指南说明了如何使用 Keil 将代码下载到 MSPM0：

- 右键单击工程文件，然后选择 **open options for target**

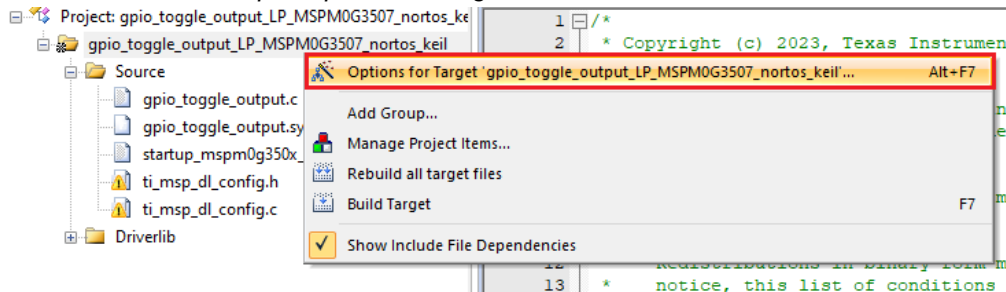


图 3-50. 打开目标的选项

- 从 *Target Options* 窗口中选择一个调试器。要使用 XDS-110，请选择 *CMSIS-DAP Debugger*。如果需要 J-Link，请选择 *J-LINK/J-TRACE Cortex*。

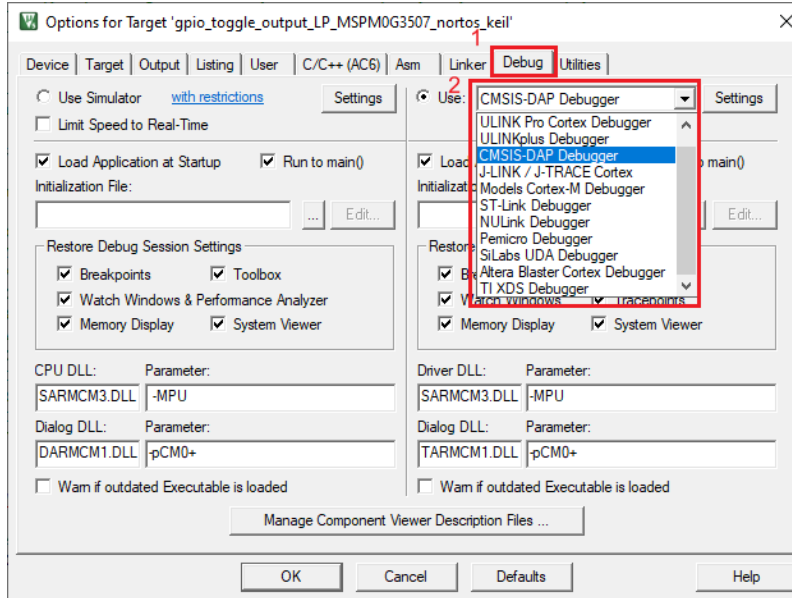


图 3-51. 选择调试窗格

- 点击 *Settings* 按钮。在 *Debug* 选项卡上，确保设置与图 3-52 和图 3-53 匹配。

### XDS110

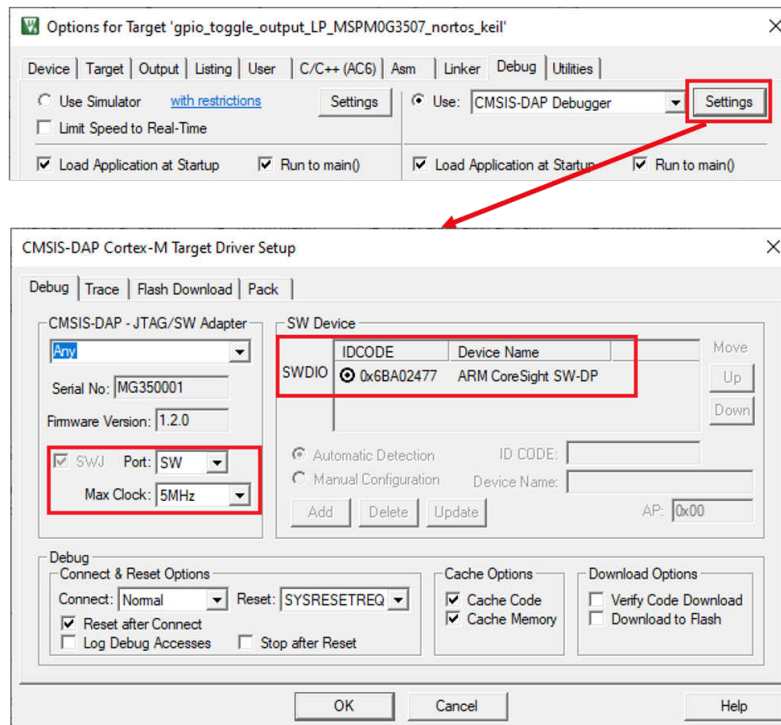


图 3-52. 检查 XDS110 探针的设置

## J-Link

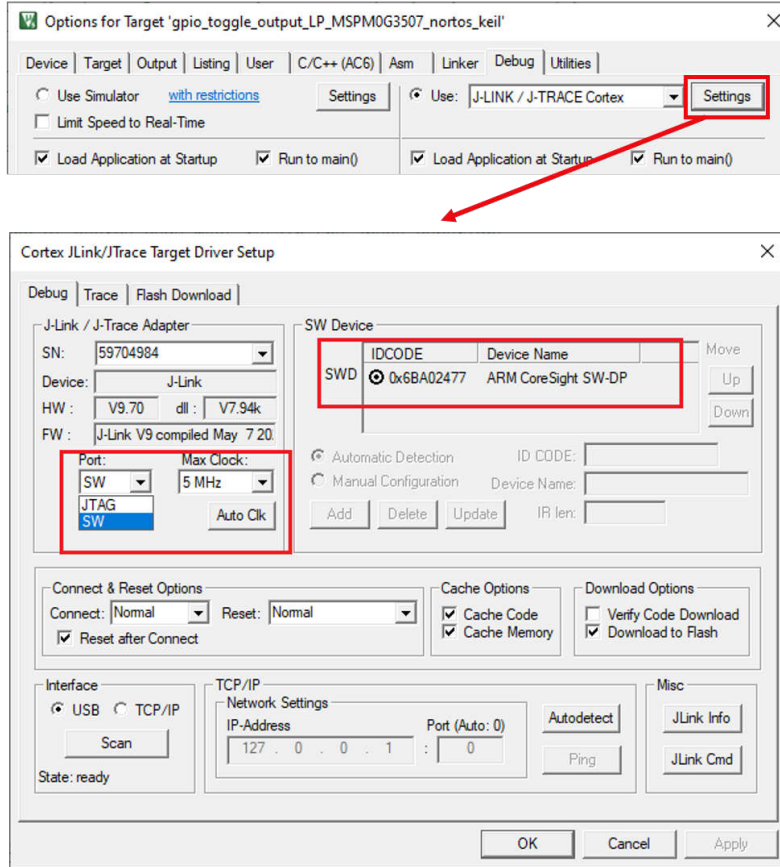


图 3-53. 检查 J-Link 探针的设置

4. 点击 **Flash Download** 选项卡，检查说明是否与图 3-54 匹配。如果不匹配，请点击 **Add** 按钮，然后选择相应的 MSPM0 MAIN 选项。器件类型为 **On-chip Flash**。最后选择 **Reset and Run**。

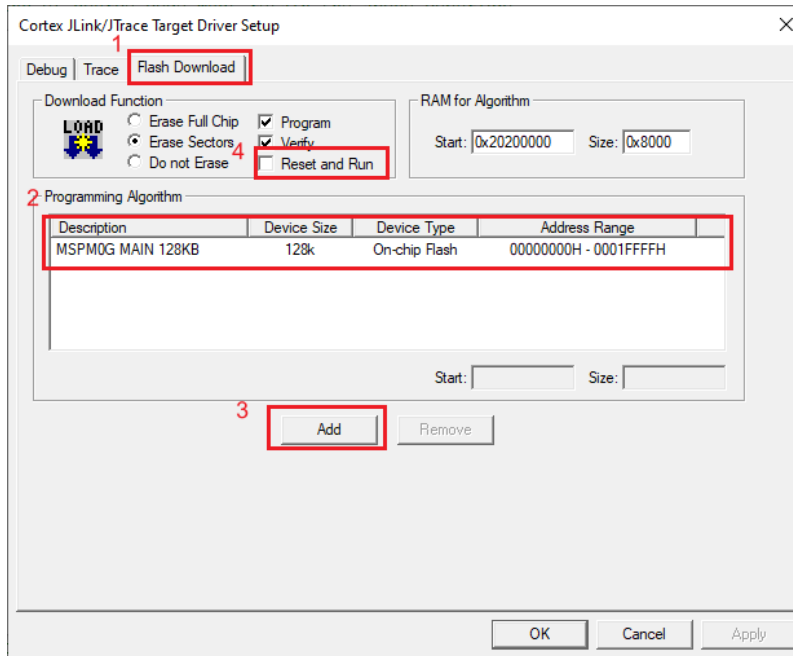


图 3-54. Flash 下载设置

5. 点击 **Build** 按钮来构建工程，然后点击 **Load** 按钮。



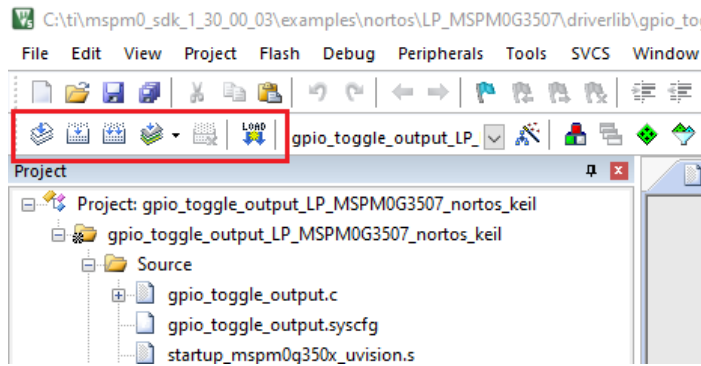


图 3-55. 下载工程

6. 要构建支持 FreeRTOS 的示例，请选择 **Project** → **Batch Setup**，然后选择该构建的所有工程目标。接下来，选择 **Batch Build** 以在工作区中构建所有工程。

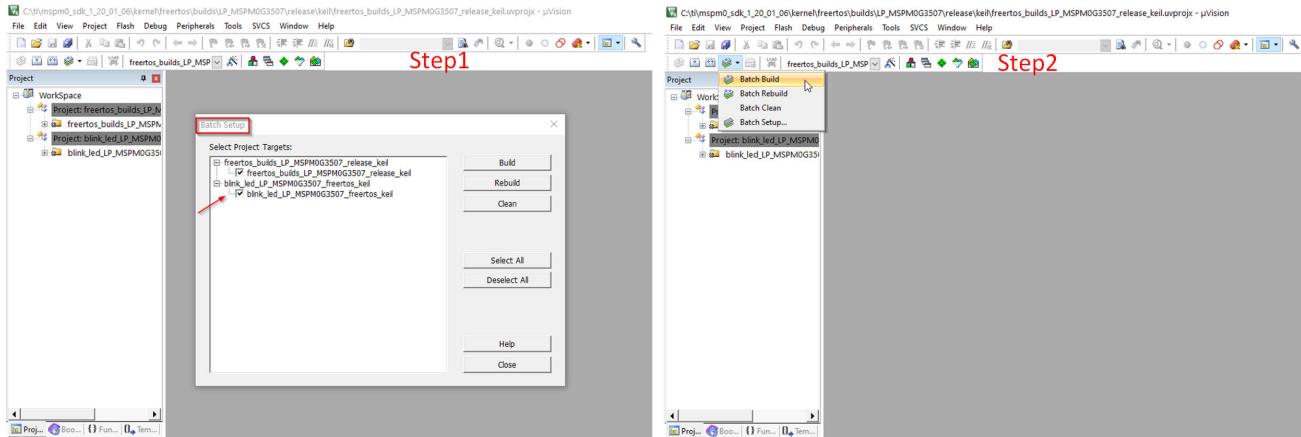


图 3-56. Keil 下的构建 RTOS 示例

#### 3.4.3.4 在 MSPM0 衍生产品之间进行迁移

SysConfig 支持在 MSPM0 衍生产品之间更轻松地迁移。但需要在 Keil 上进行一些手动修改。请按照以下步骤操作：

1. 在 SysConfig 中，启用器件视图并点击 **SWITCH**。
2. 选择新 MSPM0 器件的相应选项，然后点击 **CONFIRM**。请注意，SysConfig 会突出显示迁移的任何冲突，例如不可用的引脚和外设。根据需要修复任何冲突。
3. 在 Keil IDE 中，在工程选项中打开 **Device** 选项卡，然后选择新的 MSPM0 衍生产品。
4. 依次选择 **C/C++ (AC6)** → **Preprocessor Symbols** → **Define**，更新器件定义。根据所选器件添加器件定义。
5. 在 **Linker** → **Scatter File** 中更新链接器文件。MSPM0 SDK 在 `<sdk>\source\ti\devices\msp\m0p\linker_files\keil` 包含所有 MSPM0 衍生产品的默认文件。
6. 将新衍生产品的启动文件添加到工程，并删除现有启动文件。MSPM0 SDK 在 `<sdk>\source\ti\devices\msp\m0p\startup_system_files\keil` 包含所有 MSPM0 衍生产品的默认文件。

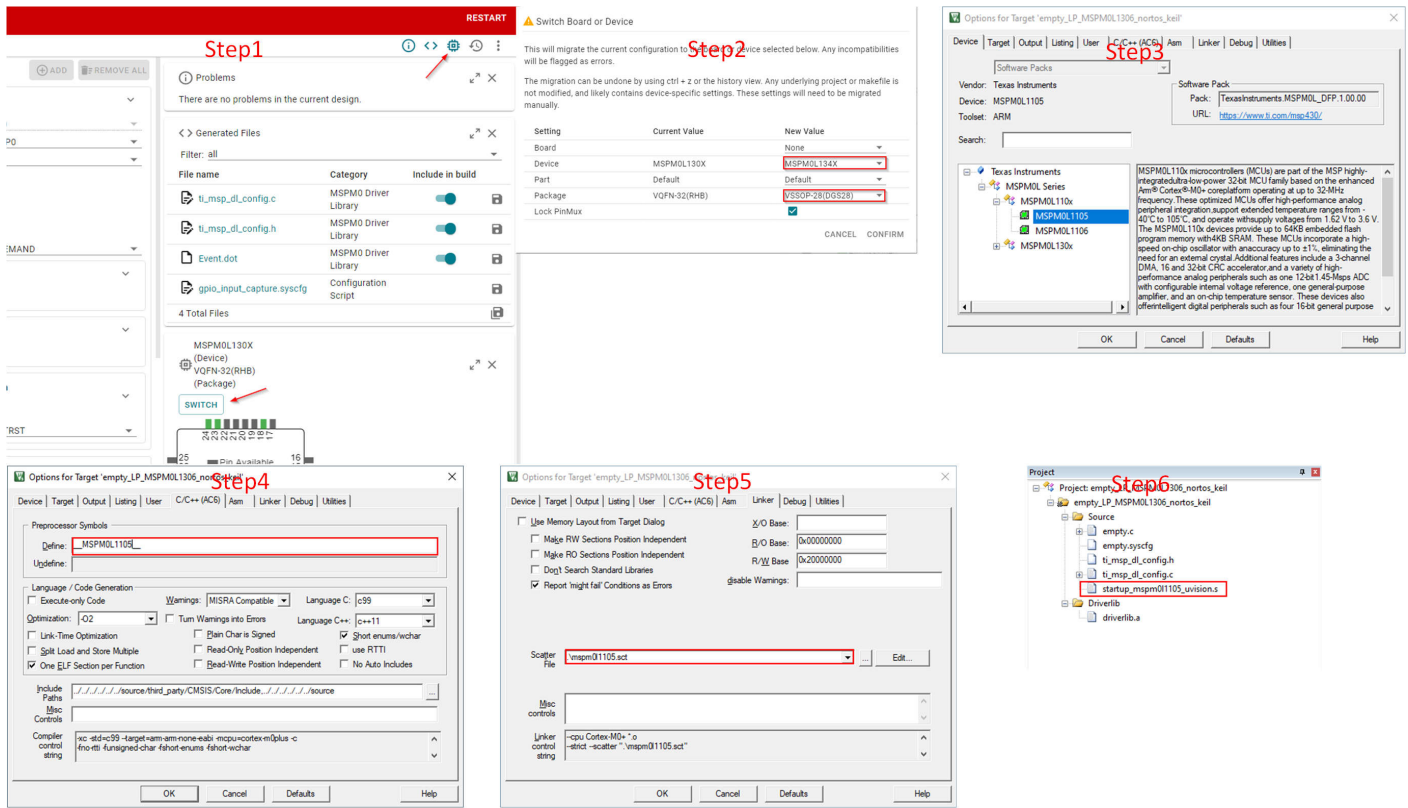


图 3-57. 在 MSPM0 衍生产品之间进行迁移

### 3.4.3.5 生成十六进制文件

以下是在 Keil 中生成十六进制文件的说明。依次点击 *Project* → *Options* → *Output* → *Create Hex File* → *OK*。您可以通过点击 *Select Folder for Objects* 来选择路径，从而找到十六进制文件。默认路径是工程文件下的目标文件夹。

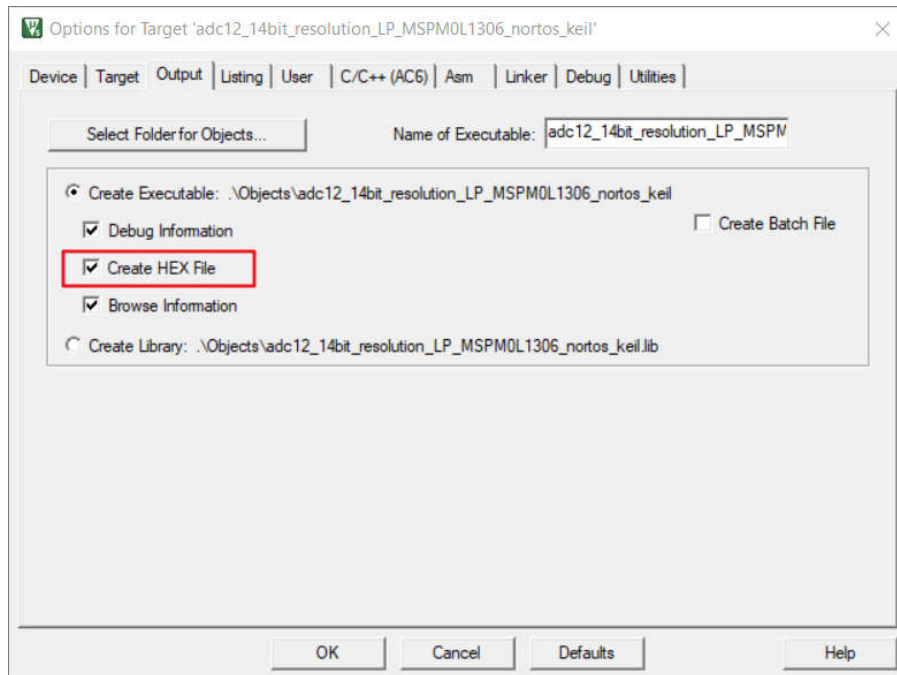


图 3-58. 生成十六进制文件

### 3.4.3.6 对 NONMAIN 编程

如果用户通过配置 NONMAIN 来更改引导加载程序或 MCU 安全设置，如节 3.3.2.4 所示，则还需要在 IAR 设置中启用 NONMAIN 擦除。请按照以下步骤操作，否则请将其保留为默认值：

1. 依次点击 *Options* → *Debug* → *Settings* → *Flash Download*。
2. 添加 NONMAIN 编程算法，然后点击 *OK*。

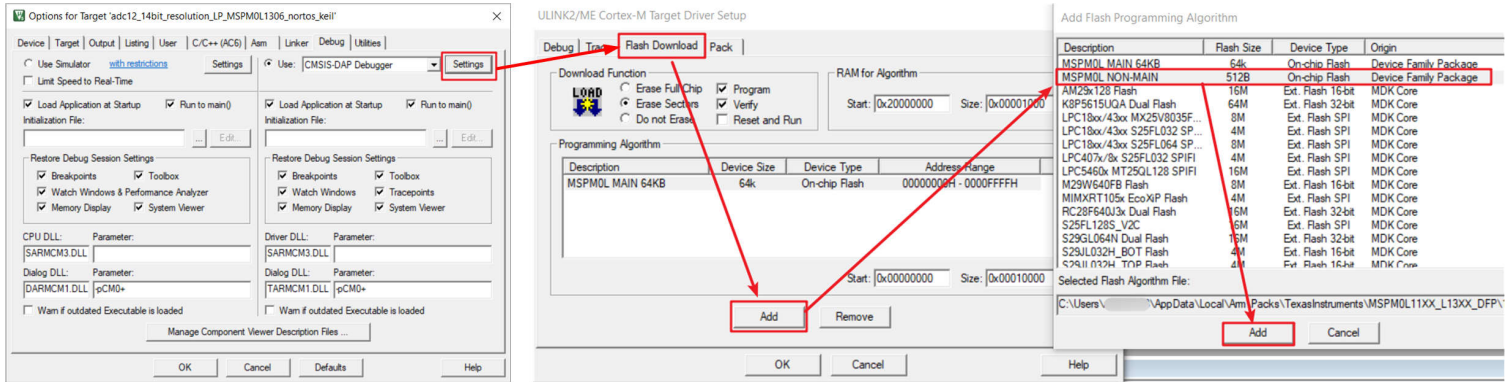


图 3-59. 对 NONMAIN 编程

## 4 第 3 步：硬件设计

### 4.1 获取 MSPM0 包

要获取 MSPM0 包，请使用 TI.com 上的 Ultra Librarian 工具，如图 4-1 所示。有关详细说明，请参阅节 8。

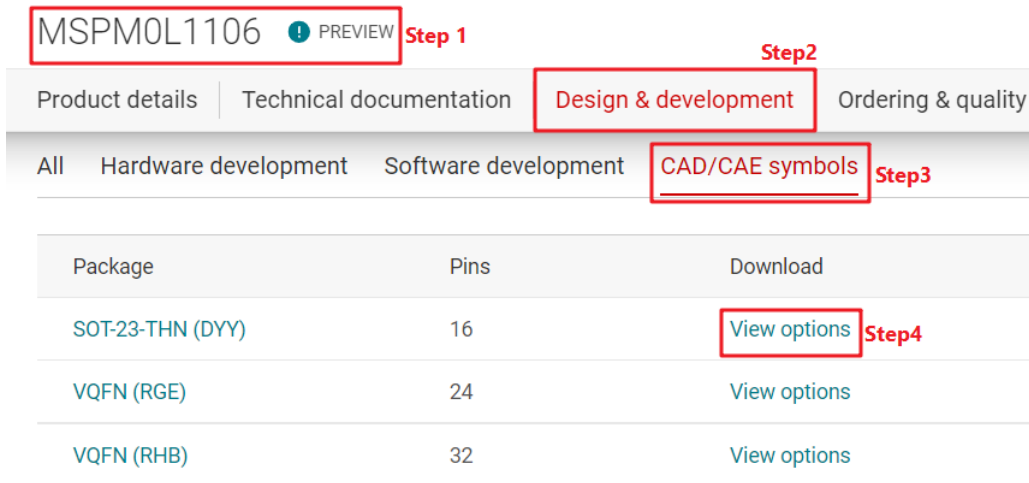


图 4-1. Ultra Librarian 工具入口

## 4.2 修复引脚功能

TI 建议硬件工程师按照图 4-2 中的说明，在软件工程师的协助下使用外设和引脚分配文件来修复引脚功能。

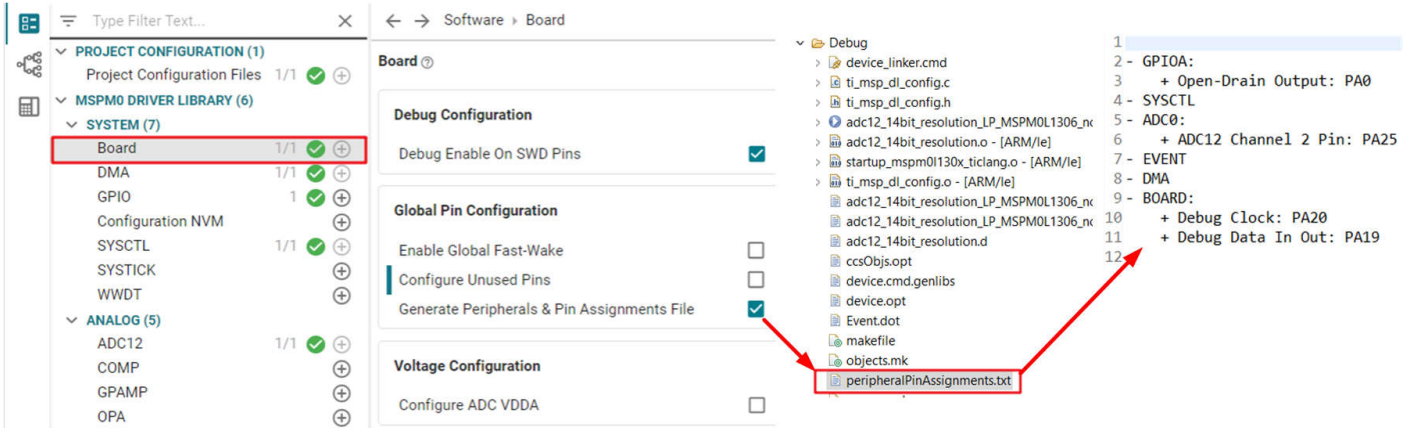


图 4-2. 生成外设和引脚分配文件

## 4.3 原理图和 PCB 生成

图 4-3 展示了 MSPM0 硬件设置的最低要求（电源、复位和 Vcore）和建议值。

- 电源引脚：TI 建议添加 10uF 和 0.1uF 电容器，用于消除电源轨上的交流噪声。
- 复位引脚：TI 建议添加一个 47kR 上拉电阻器和一个 10nF 下拉电容器。这可确保在电源轨稳定后，MSPM0 可解除复位状态。对于某些 MSPM0 器件，复位引脚可以重新配置为执行其他功能，比如 I2C 或 UART 功能。TI 建议减小电阻器和电容器，例如使用 2.2kR 上拉电阻器和 10pF 下拉电容器。
- Vcore 引脚：此引脚用于使 CPU 电压保持稳定。某些 MSPM0 器件不包含此引脚。如果包含该引脚，则将该引脚连接到 0.47uF 电容器。

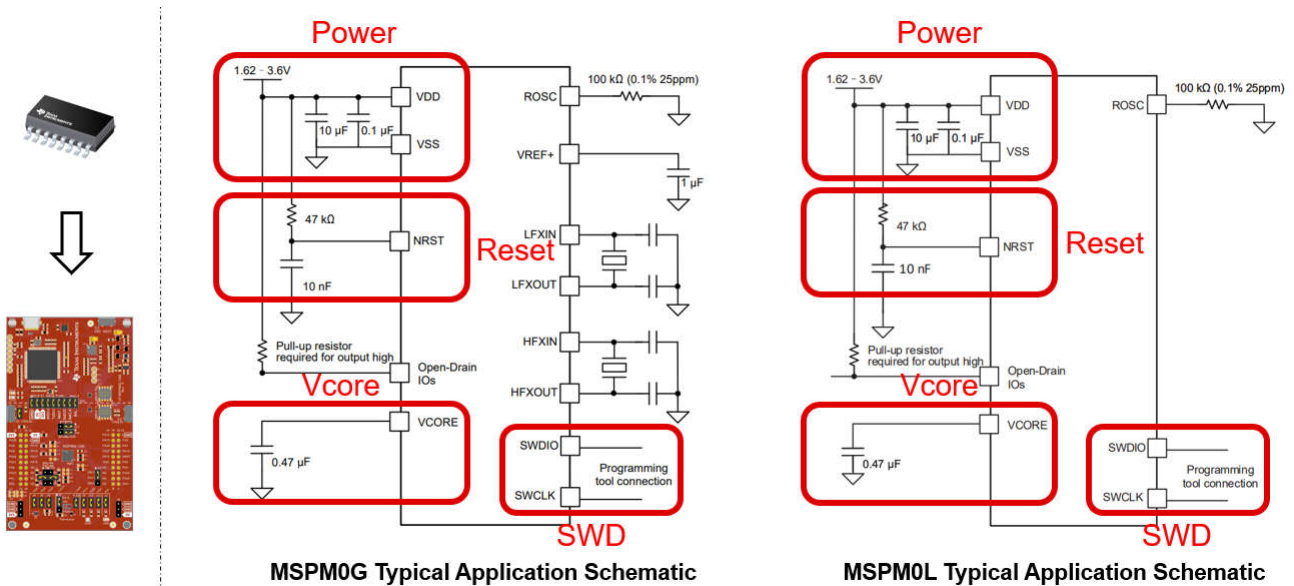


图 4-3. MSPM0 最小系统

图 4-4 中列出了绘制原理图文件时的其他注意事项。

- ROSC 引脚：如果用户希望使用内部 SYSOSC 实现精确的高频时钟，则建议使用 0.1% 电阻器。一些低成本器件无法提供此功能。
- VREF+/VREF- 引脚：

- 如果使用内部基准，则 G 系列需要在 VREF+ 和 VREF- 之间连接一个 1uF 电容器，以便支持 4MSPS ADC。对于 L 或 C 系列，不需要电容器，因为 ADC 速度仅支持 200KSPS 且使用内部 Vref。
- 如果使用外部基准，则所有 MSPM0 器件都需要在 VREF+ 和 VREF- 之间连接一个 1uF 的电容器。
- 开漏 IO：开漏 IO 无法从 MCU 侧输出高压，因此需要外部上拉电阻器，例如 4.7kR 电阻器。
- NRST：如果将复位引脚重新用作 GPIO，则仍需要上拉电阻器和下拉电容器。这可确保在电源稳定后 MCU 解除复位状态。
- PA18：PA18 是进入引导加载程序的调用引脚。确保此引脚未悬空或上拉。否则，用户可以在 SysConfig 中更改和禁用调用引脚，如节 7.3 所示。

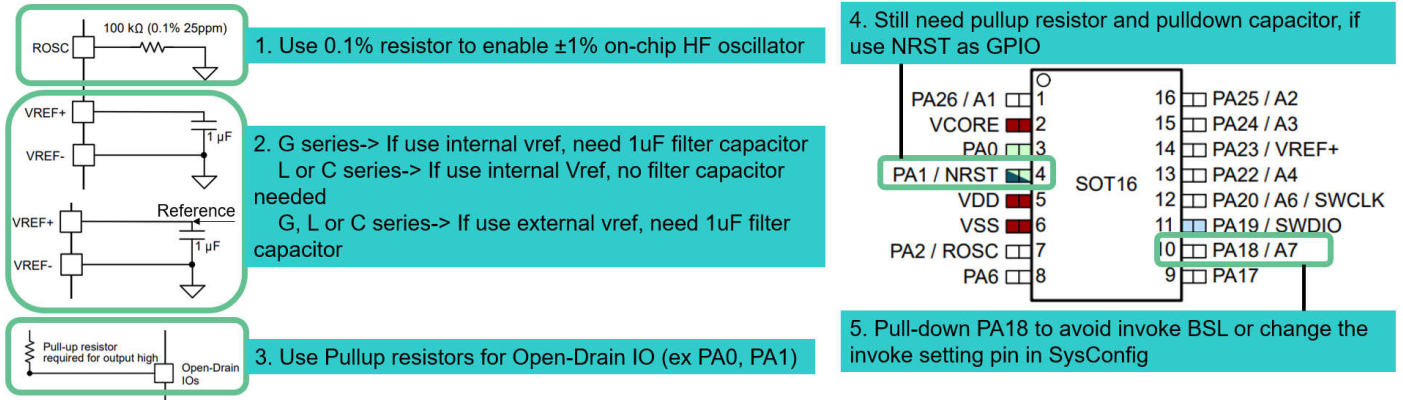


图 4-4. MSPM0 原理图

有关原理图或 PCB 设计参考的详细信息，请参阅以下链接。

- [MSPM0 L 系列 MCU 硬件开发指南](#)
- [MSPM0 G 系列 MCU 硬件开发指南](#)
- 器件特定 MSPM0 LaunchPad EVM 用户指南
- 器件特定 MSPM0 数据表



## 5 第 4 步：大规模生产

图 5-1 概述了编程软件和工具。可用的接口为 JTAG (SWD) 和引导加载程序 (BSL)。J-Link 仅支持 SWD。XDS110 和 MSP-GANG 支持 SWD 和通过 UART 的引导加载程序。

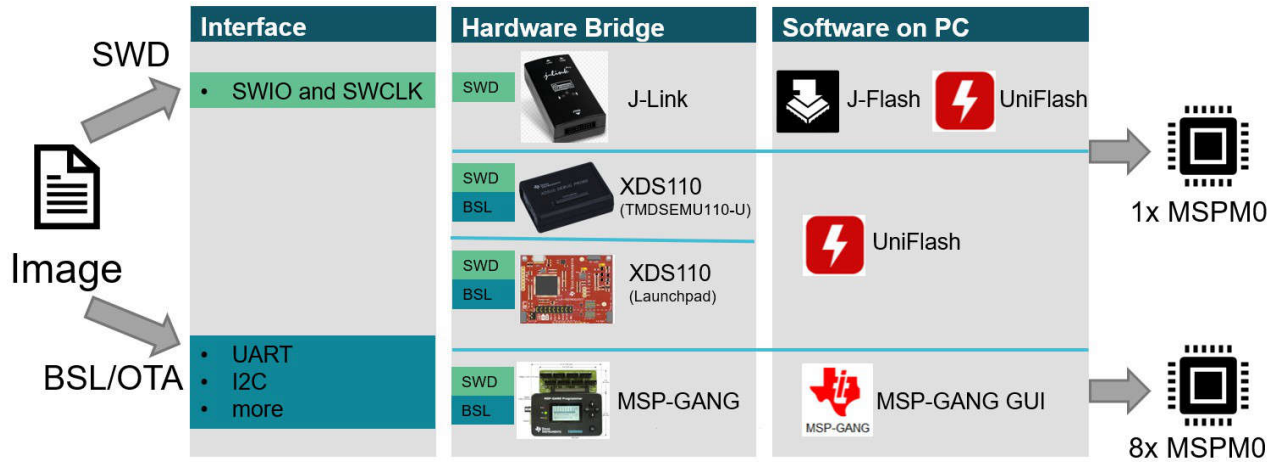


图 5-1. 编程软件和工具

有关引导加载程序的更多实现，请参阅 [MSPM0 引导加载程序 \(BSL\) 实现](#)。有关更多生产编程工具的信息，请参阅 [E2E 页面](#)。

### 5.1 生成生产映像

表 5-1 列出了由不同 IDE 生成的不同类型的映像。有关分步生成指南，请参阅 [节 3.4](#)。

表 5-1. IDE 生成的产品文件

IDE	TI_TXT (.txt)	Intel hex (.hex)	二进制 (.bin)	分步指导
CCS	Y	Y	Y	<a href="#">链接</a>
IAR	Y	Y	Y	<a href="#">链接</a>
Keil	N	Y	N	<a href="#">链接</a>

## 5.2 对软件编程

### 5.2.1 Uniflash 快速入门

本节介绍了如何在 TI 的 MSPM0 器件上安装 UniFlash 工具。有关更多信息，请参阅 [UniFlash 快速入门指南](#)。

#### 5.2.1.1 通过 SWD 编程

UniFlash 可使用调试接口 ( 如 XDS110 ) 来编程器件。所需的硬件引脚为 SWDIO、SWCLK、3V3 和 GND。请按照以下步骤操作：

1. 按照步骤选择调试器 ( XDS110 或 J-Link )。然后点击 **Start** 以开始编程。
2. 如果 NONMAIN 必须更改，请在编程之前更改擦除设置。如果不需要更改，请保留默认选项。
3. 选择映像并点击 **Load Image** 开始编程。
4. 在 **Memory** 选项卡中，只需选择 **Read Target Device**，UniFlash 还可检测器件闪存。

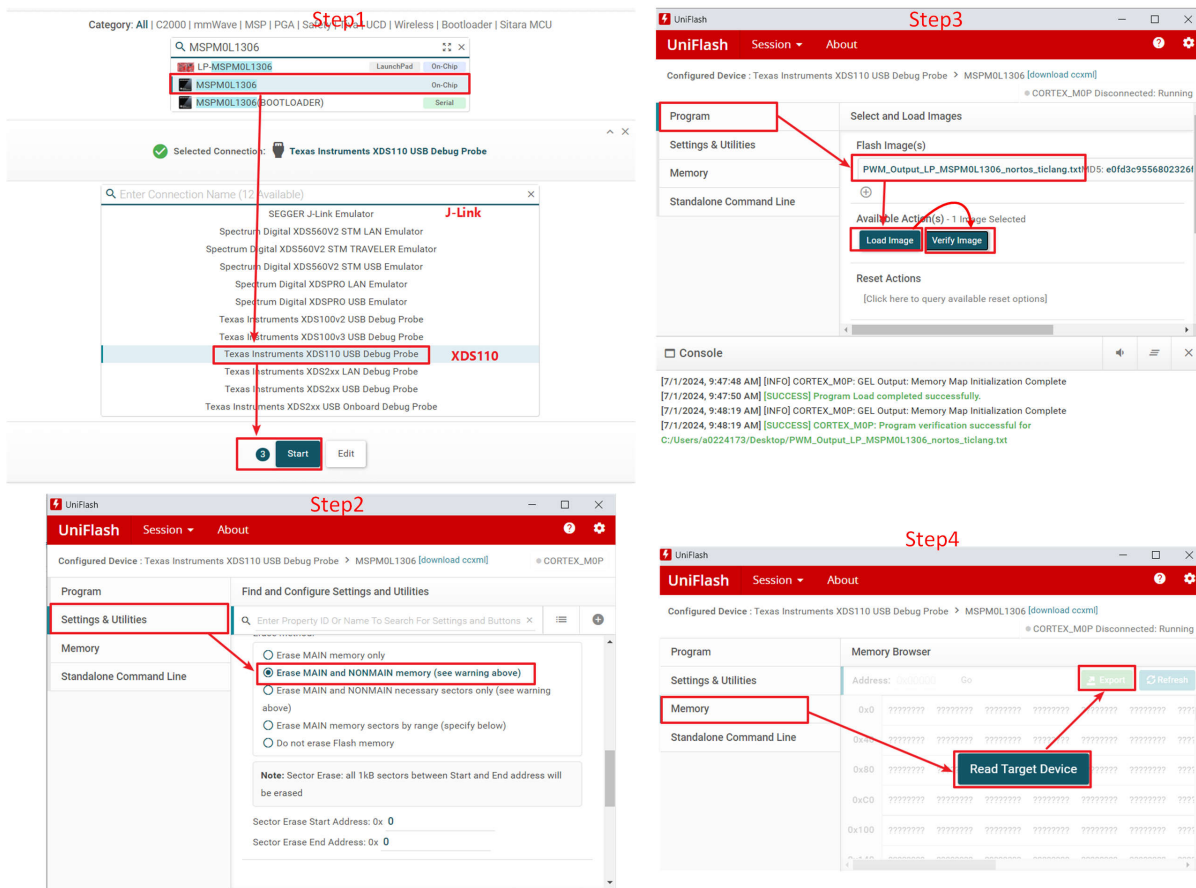


图 5-2. 通过 SWD 编程

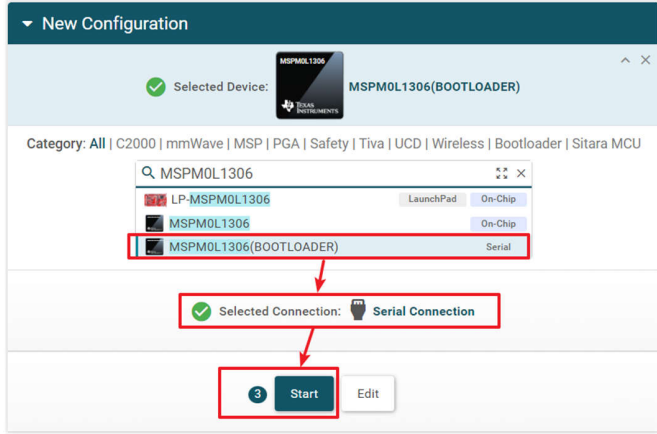
#### 5.2.1.2 通过引导加载程序编程

以下是使用 UniFlash 通过引导加载程序对 MSPM0 编程的步骤。所需的硬件引脚为 TX、RX、3V3、GND 和调用引脚。

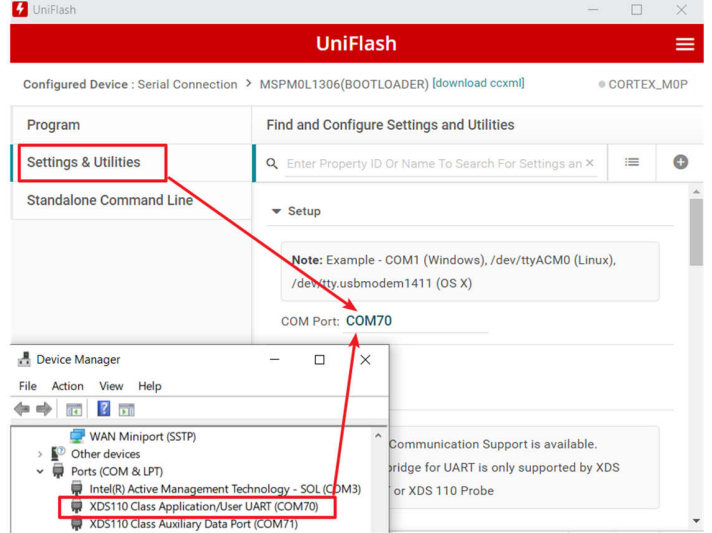
1. 搜索器件名称并选择器件的引导加载程序选项。
2. 通过查看设备管理器来检查 COM 端口。
3. 通过查阅数据表来检查 UART 引导加载程序端口。
4. 完成硬件连接 ( RX、TX、3V3、GND、调用 ) 并开始编程。



Step1



Step2



Step3

PIN Mx	PIN NAME	PIN FUNCTION	
		ANALOG	DIGITAL <sup>(1)</sup>
24	PA23	VREF+ / COMP0_IN1-	UART0_TX [2] / SPI0_CS3 [3] / TIMG0_C0 [4] / UART0_CTS [5] / UART1_TX [6] (Default BSL UART_TX)
23	PA22	A4 / GPAMP_OUT / OPA0_OUT	UART0_RX [2] / TIMG2_C1 [3] / UART0_RTS [4] / CLK_OUT [5] / UART1_RX [6] (Default BSL UART_RX)

Step4

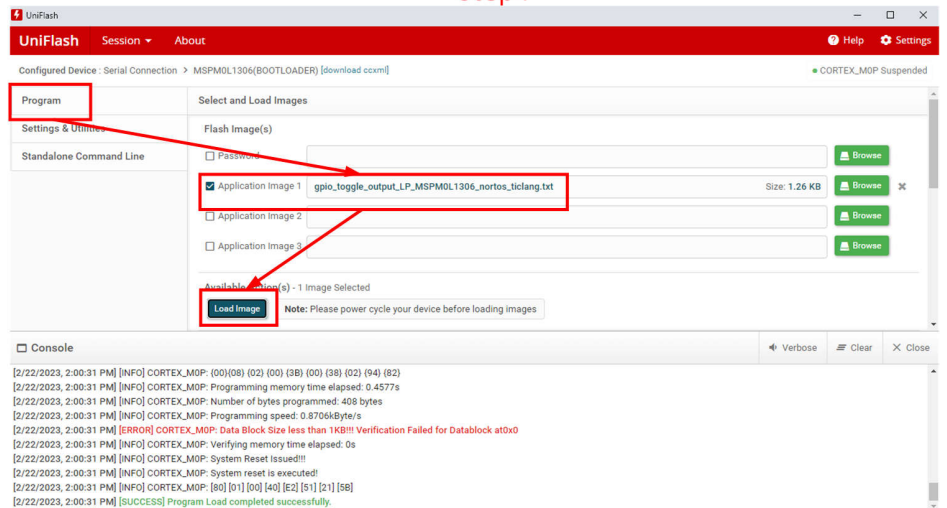


图 5-3. 通过引导加载程序编程

5.2.1.3 通过 CMD 线路接口编程

对于此要求，请参阅此 [E2E](#) 主题。

5.2.2 JFlash 快速入门

此说明根据 J-Flash V7.92n 编写而成。TI 建议使用支持所有最新版本 MSPM0 的最新 J-Flash 版本。按照以下步骤，使用 J-Flash 对 MSPM0 编程：

1. 点击 *New project*。
2. 选择相关的 MSPM0 器件型号。
3. 选择所需的编程存储器。如果 NONMAIN 不需要更改，请取消选择 *NONMAIN Memory*。
4. 点击 *Connect device*，然后点击 *Production Programming*。
5. 此时会出现确认屏幕。

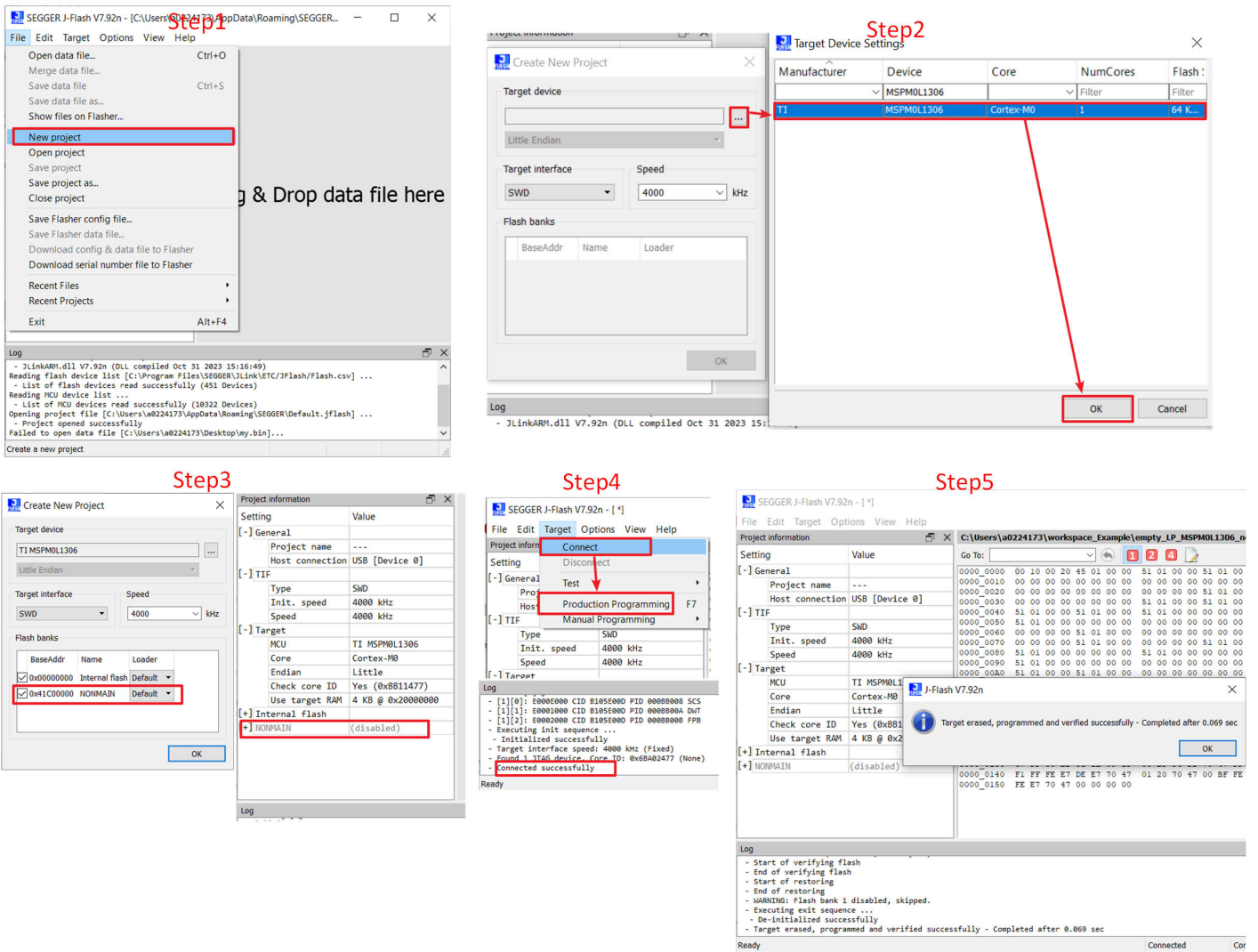


图 5-4. J-Flash 快速入门

### 5.2.3 MSP-GANG GUI 快速入门

MSP Gang 编程器 (MSP-GANG) 是一款支持 MSPM0L110x、MSPM0L130x、MSPM0G150x、MSPM0G350x 的器件编程器。目前，MSP-Gang 编程器不支持上面未提及的任何 MSPM0 器件。请参阅 E2E 页面，了解替代的生产编程工具。

有关 MSP-GANG 的快速入门指南，请参阅节 8.3。

### 5.3 对硬件编程

因为 J-Link 很常用，因此本节将重点介绍 XDS110 调试器。有关更多生产编程工具的信息，请参阅 [E2E 页面](#)。共有四种不同类型的 XDS110 调试器。表 5-2 列出了摘要表。

表 5-2. XDS110 调试器摘要

支持的功能	XDS110		XDS110 板载元件	
	TMDSEMU110-U	MSPM0 LaunchPad	LP-XDS110	LP-XDS110ET
JTAG	是	否	是	是
SBW	是	是	是	是
EnergyTrace	是	依赖类型	否	是
MSPM0 引导加载程序	是	依赖类型	否	否
备注	最高性能	最便宜	易于使用	易于使用

对于 TMDSEMU110-U 器件，使用的引脚如图 5-5 所示。用于引导加载程序时，GPIOOUT0 必须连接到 MCU 复位引脚。GPIOOUT1 必须连接到 MCU 调用引脚 (PA18)。

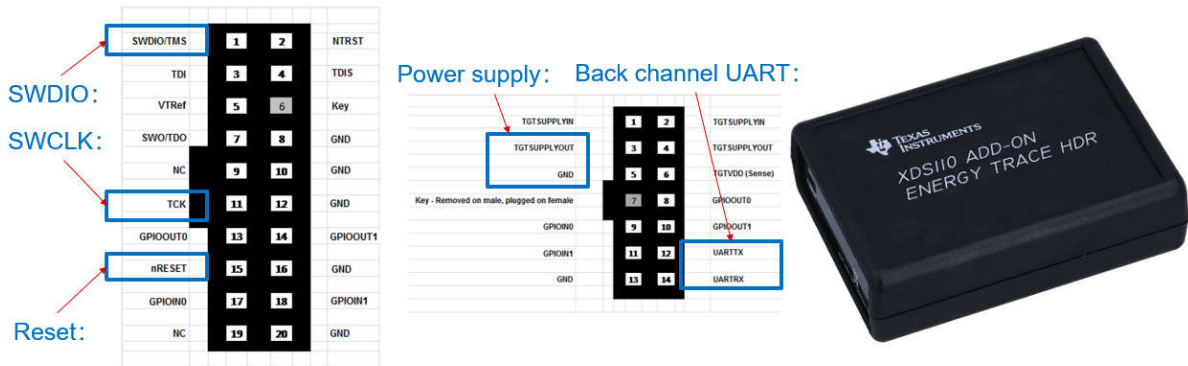
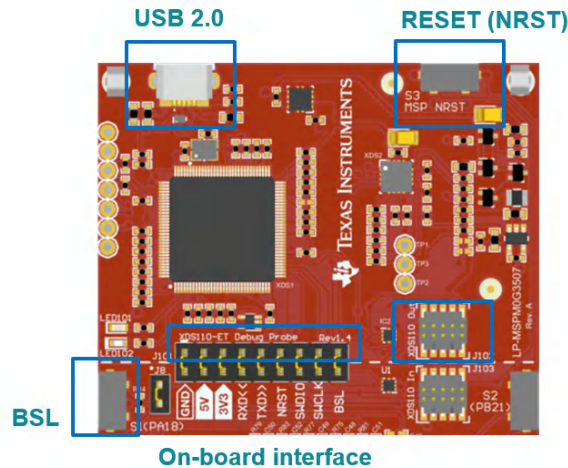


图 5-5. TMDSEMU110-U 的引脚连接

对于 LaunchPad 上的 XDS110，与 TMDSEMU110-U 相比，基本编程功能不变。图 5-6 展示了该电路板。LaunchPad 上最便宜的 XDS110 是 LP-MSPM0C1104。但是，LP-MSPM0C1104 仅支持 SBW，并且没有 EnergyTrace 或引导加载程序功能。



LP-XDS110 和 LP-XDS110ET 与 LaunchPad 上的 XDS110 类似。区别在于，其中一个具有 EnergyTrace 功能，而另一个没有 EnergyTrace 功能。图 5-7 展示了引脚分配。

对于 LP-XDS110 和 LP-XDS110ET，通过更改电路板左下角的跳线来启用电平转换功能。支持的电压范围为 1.2V 至 3.6V。

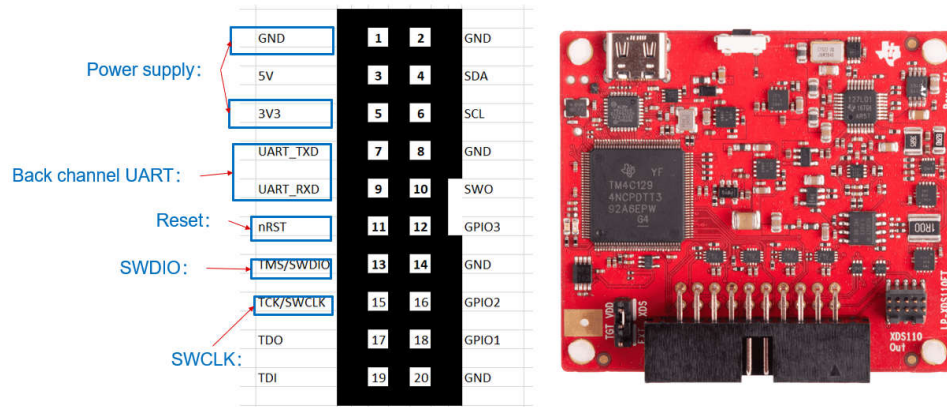


图 5-7. LP-XDS110ET

## 6 第 5 步：质量和可靠性说明

TI 致力于提供可满足客户需求的高品质和可靠半导体设计。我们的整体质量方法渗透到公司供应链的各个层面，包括从工艺技术和设计一直到制造、封装、测试和交付。

### 6.1 质量和可靠性材料入口

这是[质量和可靠性](#)的登录页面。以下是该页面下的常用工具和链接：

- **认证摘要★**：用于搜索相关器件的可靠性数据。器件系列所使用的材料组合、工艺和制造基地的代表性数据摘要。
- **可靠性测试**：列出 TI 对产品进行的各种可靠性测试。
- **客户故障分析**：[客户故障分析](#) 页面详细说明了如何将故障器件退回 TI。
- **DPPM/FIT/MTBF 估算器**：DPPM/FIT/MTBF 估算器搜索工具可基于技术分组查找通用数据，从而估算这些典型问题，并显示得出这些比率的条件。
- **持续可靠性监测**：持续可靠性监测 (ORM) 项目的搜索工具将按晶圆制造工艺或器件封装系列提供季度 ORM 报告。
- **封装**：用户可以通过此网站了解封装注意事项，包括封装尺寸、SMT 建议、可靠性和性能预期。

### 6.2 失效信息收集和分析指南

进行失效分析需要收集尽可能多的技术背景信息，以便缩小分析范围并加快分析速度。如果用户在 MSPM0 上遇到任何器件故障，则收集以下信息，并通过[客户故障分析](#)页面联系 TI 或为您的产品或业务提供支持的区域 CQE 和销售人員。

器件名称 ( TI 器件型号，包括封装标识符 )：

- 示例：MSPM0L1306SRGER

故障率 ( 已购器件与客户故障器件的比率 )：

- 示例：故障率：5% ( 总测试数量：2000，故障数量：100 )

检测位置 ( 现场故障、生产、传入... )：

- 示例：电路板级功能测试

应用的原理图：

- 示例：MCU 器件的原理图，每个输入和输出信号都有详细说明

详细的器件级故障说明

- 示例：MCU PA1 无法输出高电压

本节介绍了收集失效信息的常用方法。

- **方法 1：ABA 交换测试**。在某些失效情况下，很难判断问题是由器件引起，还是器件与整个系统之间的关系所引起。ABA 交换测试是解决此问题的好方法，可提供更多信息以供进一步分析。以下是执行 ABA 交换测试的步骤：从原故障电路板上拆下疑似出现故障的元件 (A)。使用已知正常的元件 (B) 替换疑似出现故障的元件 (A)，查看原故障电路板现在是否正常工作。将疑似出现故障的元件 (A) 安装到已知良好的电路板上，查看正常电路板上是否出现同样的故障。
- **方法 2：在低功耗模式下检查 MCU 电流消耗**。某些器件故障是由 EOS ( 静电过应力 ) 所引起。在将 MCU 设置为低功耗模式 ( 如待机模式 ) 后也可以检测到该故障。由于 EOS 会导致出现额外的漏电流，用户会发现电流消耗高于数据表规格。
- **方法 3：引脚阻抗检查**。一些 EOS ( 静电过应力 ) 完全在 I/O 上发生，使用引脚阻抗检查可以轻松捕获此故障，从而向 TI 提供更多信息。一种正式方法是使用曲线轨迹分析 (CT)。但最终用户也可以使用万用表进行大致评估。用户可以选择在为器件供电或不供电的情况下检测 IO 电阻。处于高阻抗状态的 GPIO 的电阻需要达到 MΩ 级。但是，因为万用表在进行测试时会向检测电路施加电流，这可能会导致关断器件损坏，所以在通电的情况下测试器件更安全。



- 方法 4：查找最小的系统或代码示例。典型应用和典型代码工程会发生一些失效情形。通过比较方法，逐步消除不相关的硬件设置和软件代码可以逐渐缩小分析范围。最好的结果是该问题仅与器件相关，并且是一个最简单的代码示例。因此，TI 可以更快地进行进一步的失效分析。

## 7 常见问题

本节列出了一些可供用户搜索的常见问题。如有进一步的问题，请搜索器件特定的数据表、技术参考手册或 E2E。TI 工程师将于 24 小时内在此在线支持平台上给予回复。

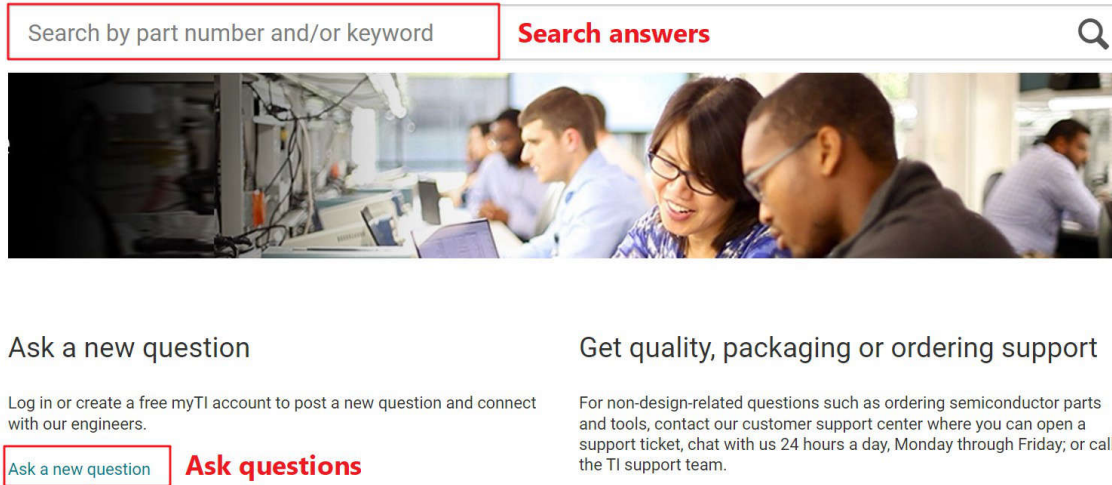


图 7-1. E2E 在线论坛

### 7.1 MSPM0 编程故障

如果首次出现编程故障，请逐一检查以下各项：

1. 在英文路径中安装最新的 IDE 或编程软件工具。建议使用默认安装路径。有关安装说明，请参阅本说明中的相关章节。
2. 插入调试器，并检查计算机是否找到调试器。如果调试器未如图 7-2 所示，请检查计算机限制。

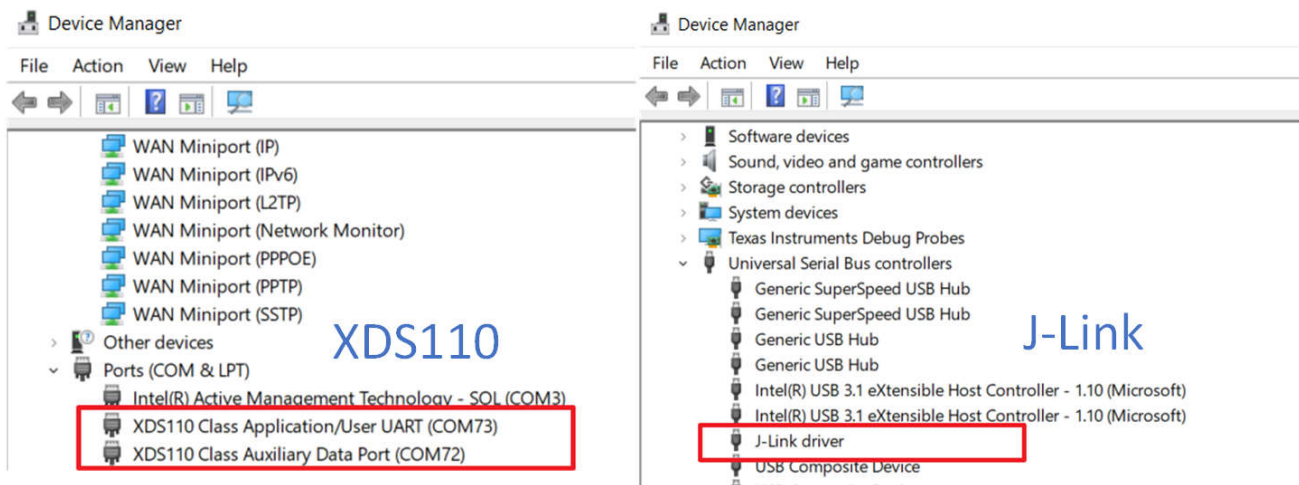


图 7-2. 设备管理器视图

3. 尝试使用 MSPM0 Launchpad 进行编程，以检查 PC 环境设置是否正常。
4. 对于您的定制板，请参考节 4.3 检查原理图。请注意 Vcc、Vcore 和 RESET 引脚设置。
5. 检查调试器和 MSPM0 之间的连接。用户可以参考节 5.3，使用万用表直接检查调试器侧的信号路径，并参考相关数据表，检查 MCU 引脚侧的信号路径。
6. 检查电路板上的电源。请记住，调试器的功率输出存在限制，输出电压只能是 3V3。需要额外的电源。
7. 使用示波器检查 SWDIO 和 SWCLK 上的信号波形，尤其是当导线非常长时。请确保信号建立时间足够。

如果第二次出现编程故障，并且之前可以对器件编程，请参阅节 7.2。

## 7.2 解锁 MCU

MSPM0 在进入 STOP、STANDBY 或 SHUTDOWN 模式时会遇到 SWD 连接问题。这种限制的影响取决于 IDE 和调试器的实现。请使用最新版本的工具，如表 7-1 中所示。有关更多详细信息，请参阅 [MSPM0 SDK 已知问题](#) 和 [常见问题解答](#) 中的“在低功耗模式下调试”一章。

表 7-1. 建议的工具版本

Keil CMSIS 软件包	IAR IDE	CCS IDE	J_Link
MSPM0L11XX_L13XX_DFP : 1.3.1+ MSPM0G1X0X_G3X0X_DFP : 1.3.1+ MSPM0C110X_DFP : 1.1.1+ MSPS003FX_DFP:1.1.0+ MSPM0L122X_L222X_DFP:1.1.0+	9.60.1+	12.80+	V 8.10+

下载错误代码后 MSPM0 也会断开连接，并且 CCS 会在编写新代码时报告错误。图 7-3 展示了一个示例。

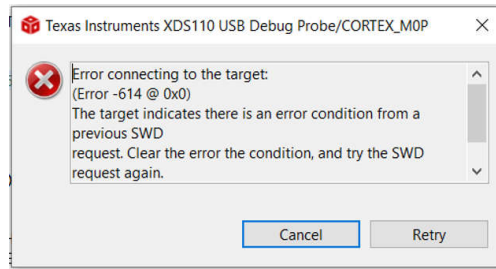


图 7-3. CCS 误差

借助调试子系统邮箱 (DSSM)，调试探针能够通过 SWD 接口将消息传递到 MSPM0 器件的引导 ROM。在工具有四条解锁命令可供选择。表 7-2 简要说明了这些命令。建议使用 **DSSM 恢复出厂设置**，其复位级别高于 DSSM 批量擦除。

表 7-2. 解锁命令

解锁命令	与调试器的硬件连接	复位引脚控制	命令影响
手动 DSSM 恢复出厂设置	3v3、GND、SWDIO、SWCLK、Reset	最终用户	擦除主闪存并复位 NONMAIN 闪存
自动 DSSM 恢复出厂设置		调试器	
手动批量擦除 DSSM		最终用户	擦除主闪存
自动批量擦除 DSSM		调试器	

有关提供的三种解锁方法的建议，请参阅表 7-3。需要注意的是，解锁方法仅支持 XDS110，当前不支持 J-Link。

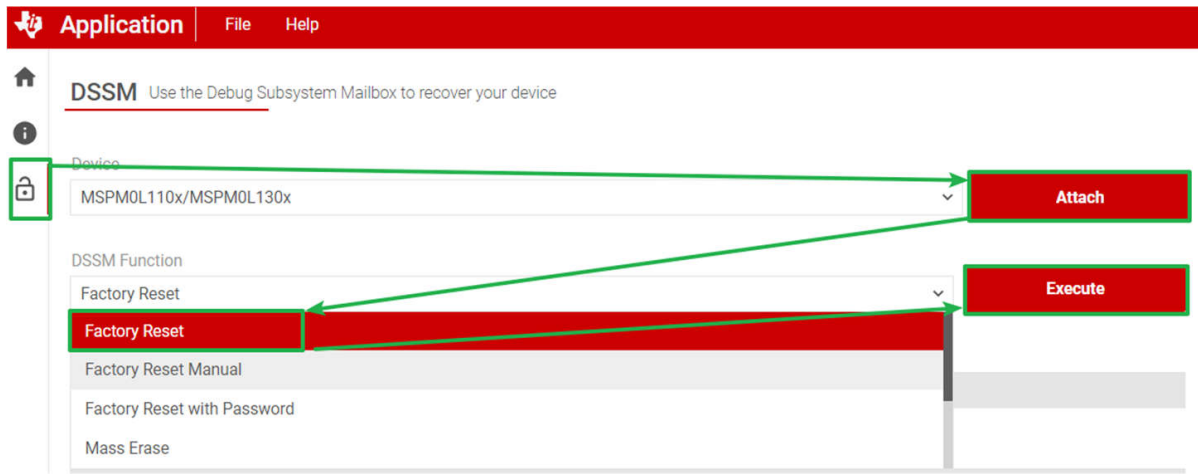
表 7-3. 解锁方法选择

解锁方法	支持的调试器	何时选择
恢复出厂设置 GUI 工具	XDS110	互联网连接可用
Uniflash	XDS110	互联网连接不可用
CCS	XDS110	使用 CCS 作为开发 IDE

### 7.2.1 通过恢复出厂设置 GUI 工具解锁

**MSPM0 恢复出厂设置 GUI 工具** 是一个独立的工具，用于通过该界面访问调试功能或恢复 MSPM0 器件。此工具免费提供。按照步骤复位 MSPM0。





### Output console

```

CS_DAP_0: GEL Output: SEC_AP Reconnect
CS_DAP_0: GEL Output: Command execution completed.
CORTEx_M0P: GEL Output: Factory Reset executed. Please terminate debug session, power-cycle and restart debug session.
DSService deconfigured. Core deattached/closed.
    
```

图 7-4. 通过 GUI 解锁

### 7.2.2 通过 Uniflash 解锁

8.7.0.4818 版本以上的 Uniflash 还支持解锁 MSPM0。首先，按照步骤连接 MSPM0 与 Uniflash，如节 5.2.1.1 所示。然后，按照图 7-5 中的说明解锁 MSPM0。

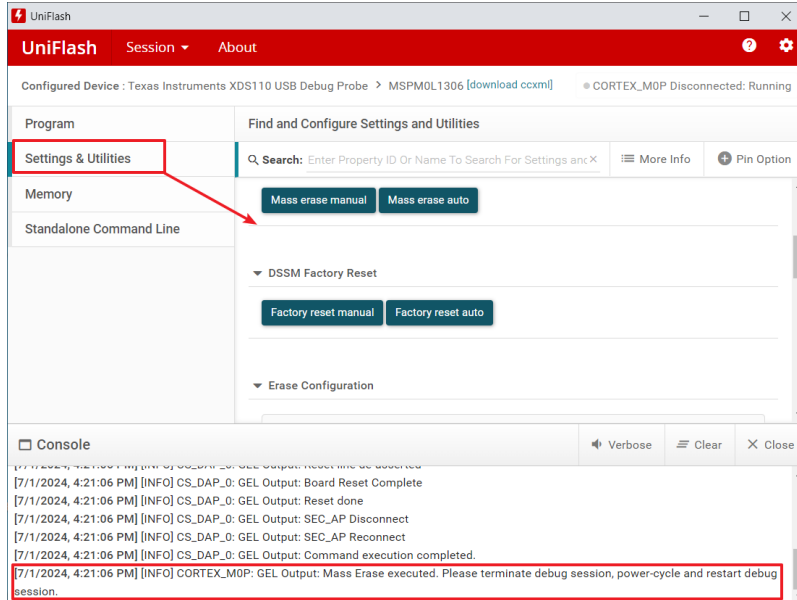


图 7-5. 通过 Uniflash 解锁

### 7.2.3 通过 CCS 解锁

以下是通过 CCS 解锁 MSPM0 的步骤：

1. 在 CCS 菜单上，依次选择 **View** → **Target Configurations**。在 **Target Configurations** 窗口中，右键点击活动工程的 .ccxml，然后选择 **Launch Selected Configuration**。
2. 点击 **Debug Probe**，然后依次选择 **Scripts** → **MSPM0xxx\_Commands**。
3. 如果您选择手动命令，则需要根据控制台中的命令复位器件。之后，可对器件重新供电。如果您选择自动命令，则可以按照说明对器件重新供电。

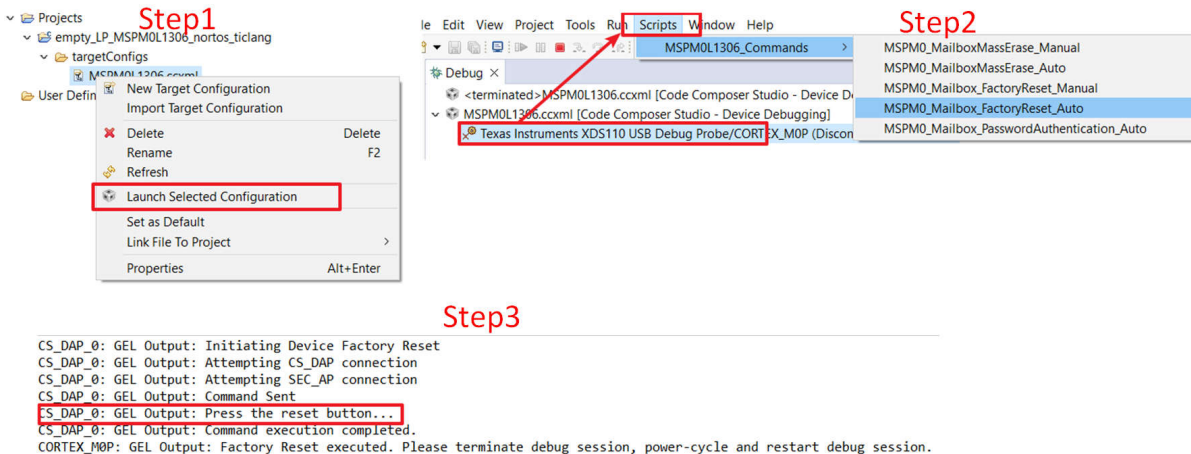


图 7-6. 通过 CCS 解锁

### 7.3 MCU 在调试和自由运行时的表现不同

MSPM0 在调试和自由运行时的表现不同。检查 PA18 上的设置。当 PA18 输入拉至高电平或受此引脚悬空的噪声影响时，在 MSPM0 复位或重新上电后，该器件会在自由运行模式下进入引导加载程序。如果遇到此问题，并且 PA18 无法使用外部电阻器拉至低电平，则可以按照图 7-7 中的步骤来禁用 BSL 或更改调用引脚分配。由于这些设置需要更改 NONMAIN，请参阅节 3.4 中相关 IDE 的“对 NONMAIN 编程”一章。

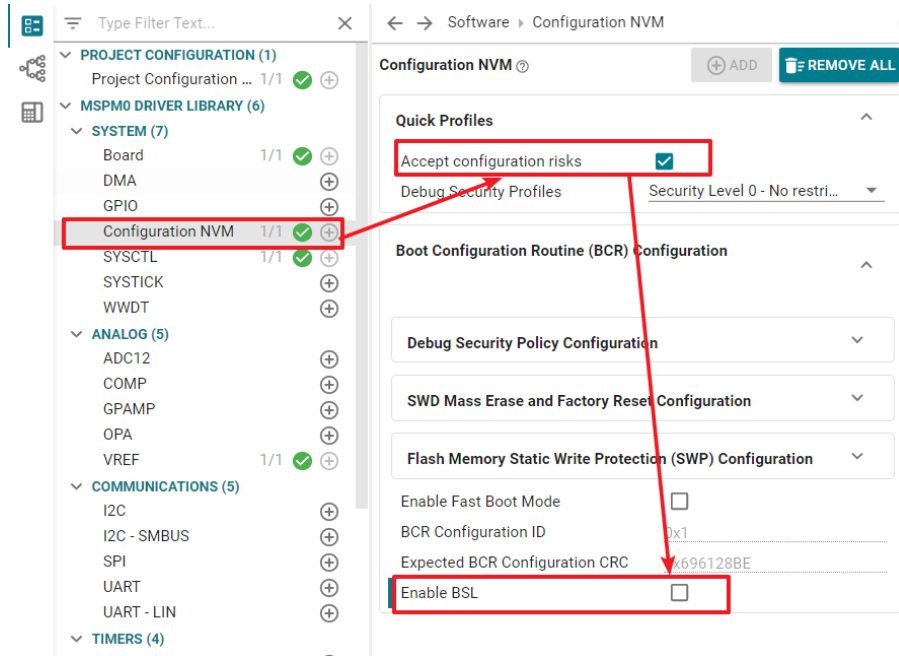


图 7-7. 禁用 BSL

### 7.4 BSL 相关问题

有关如何使用引导加载程序的问题，请参阅 [MSPM0 引导加载程序 \(BSL\) 实现](#)。这篇文章概述了引导加载程序的实现并提供了分步说明。

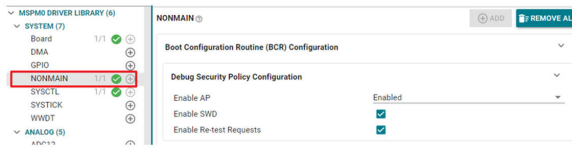
有关引导加载程序协议及规范的问题，请参阅 [MSPM0 引导加载程序用户指南](#)。

### 7.5 设置 SWD 密码

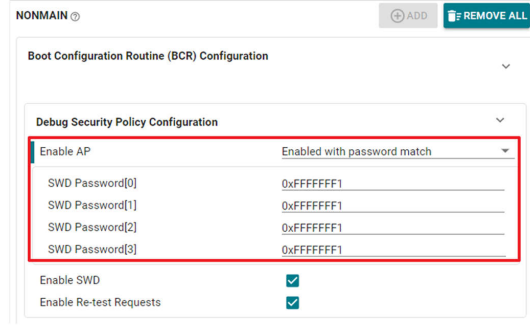
通过写入 NONMAIN 中的 BOOTCFG0 和 SWDPW 寄存器，可以将 SWD 接口配置为禁用、启用或使用 128 位密码启用。有关 NONMAIN 和 SWD 密码的详细信息，请参阅器件的技术参考手册。您可以按照以下步骤在 SWD 上添加密码。

## Step1: Change AP Setting, and add password

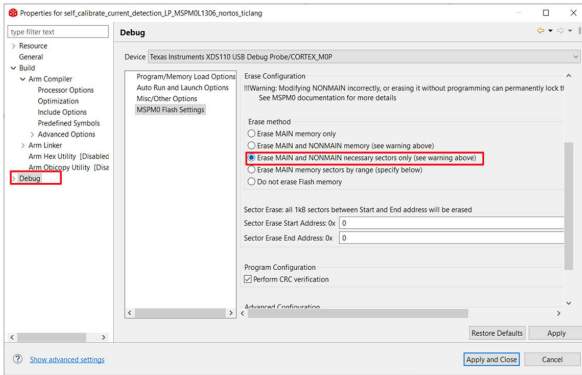
a:



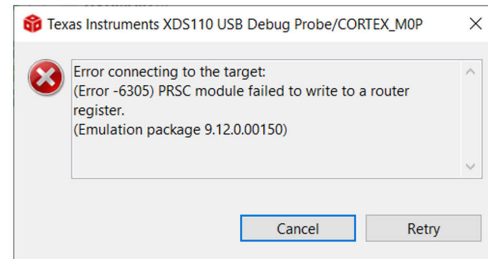
b:



## Step2: Enable NONMAIN Erase



## Step3: Repower, Device is locked

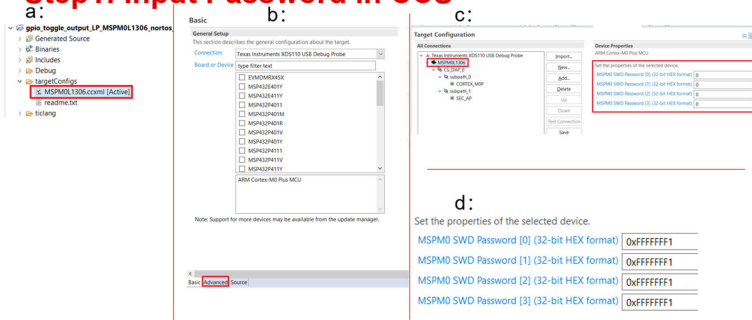


SBW security will work only after repower!

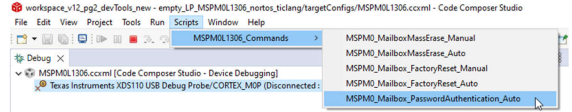
图 7-8. 启用 SWD 密码

以下使用密码对 MSPM0 重新编程的步骤。此操作不会擦除 NONMAIN，因此除非修改了 NONMAIN，否则密码会保持有效。

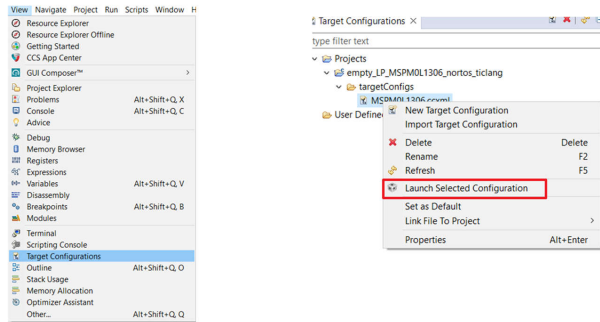
## Step1: Input Password in CCS



## Step3: Connect Device with Password



## Step2: Launch Configuration



## Step4: Reprogram

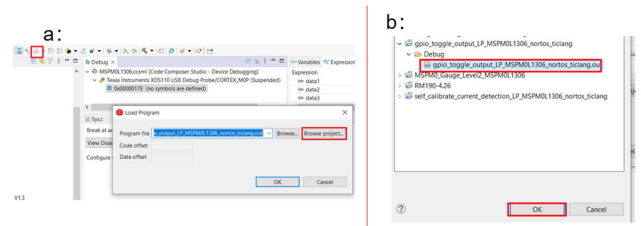


图 7-9. 对器件重新编程

## 7.6 CCS 常见问题

本部分介绍了在 CCS 中遇到的一些常见问题。以下是一些附加文档，如果在使用 TI 的编译器、链接器或 IDE 时遇到问题，可以参考这些文档：

- [ARM 汇编语言工具用户指南](#)
- [ARM 优化 C/C++ 编译器用户指南](#)
- [TI Arm Clang 编译器工具用户指南](#)

### 7.6.1 在所需的位置设置断点

默认 SDK 示例采用优化级别 2。减小了代码大小。但这会导致 C 代码不匹配，并且无法在特定 C 代码行添加汇编代码和断点。要解决此问题，请将优化级别 2 改为优化级别 0。

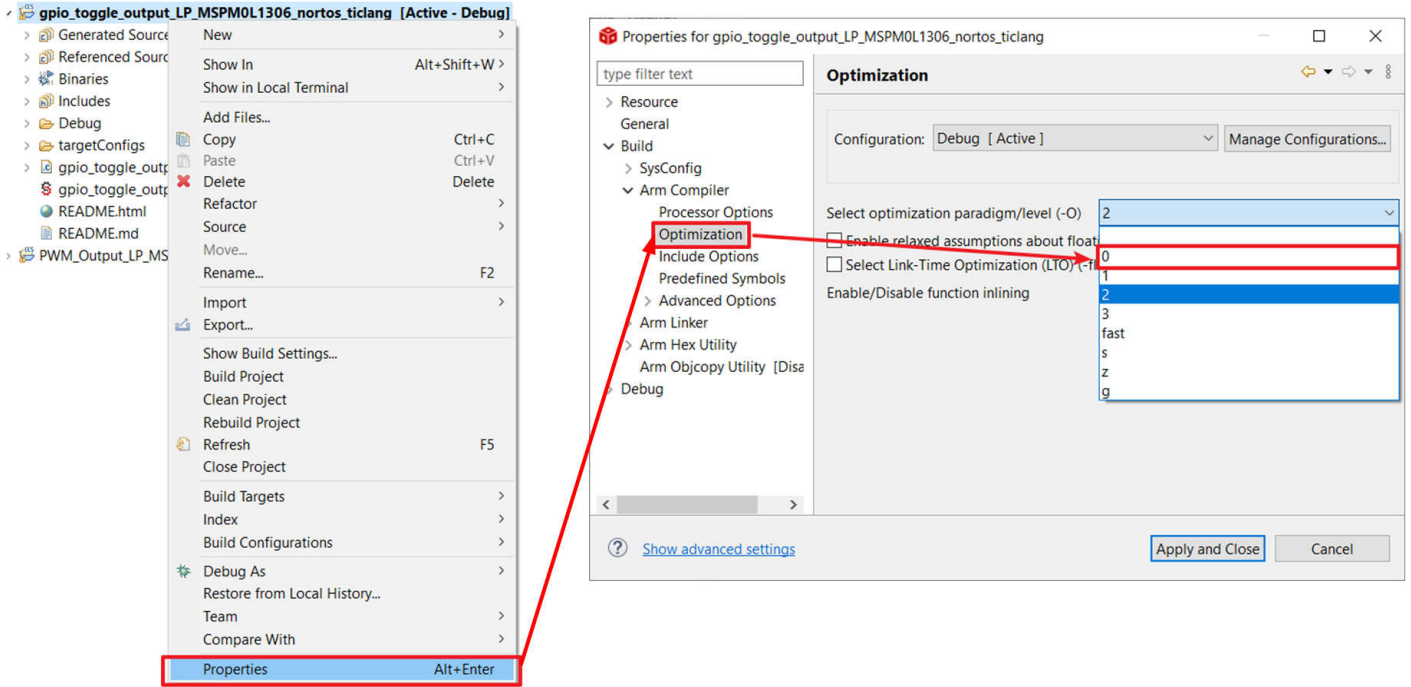


图 7-10. 更改优化级别

### 7.6.2 已发现工程变为灰显

CCS 有一个工作区概念。如果工作区包含同名工程，则在导入新工程时无法选择新工程。如果在删除工程时没有删除工作区中的副本，就会发生此问题。图 7-11 展示了一个示例。

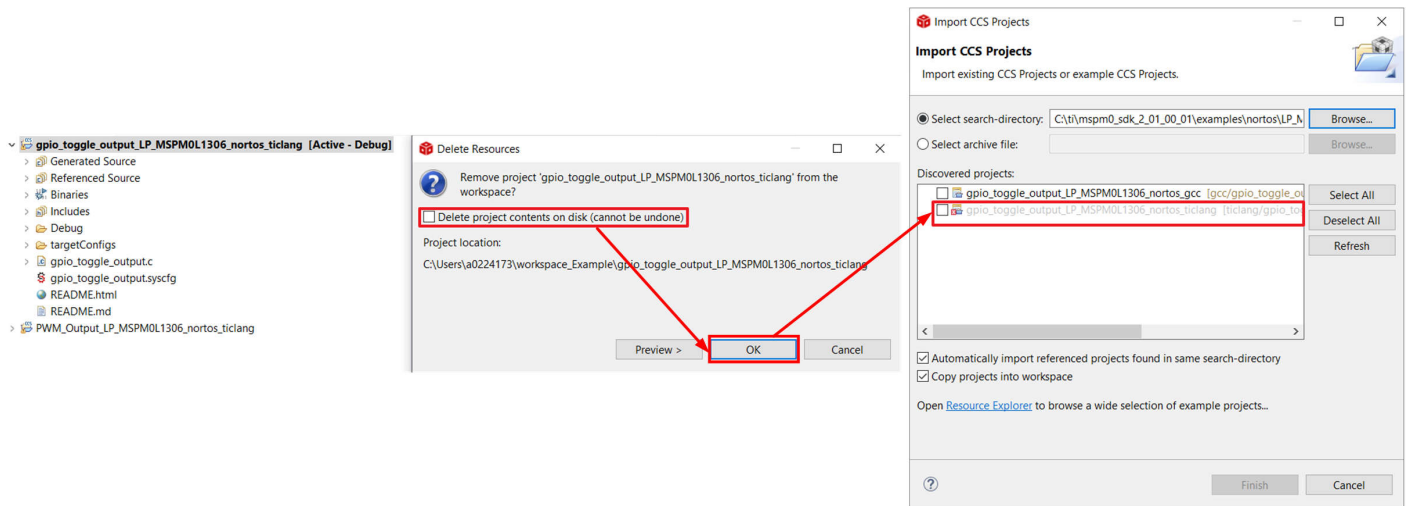


图 7-11. 无法选择工程



请按照以下步骤解决此问题：

1. 打开工作区地址。
2. 复制工作区地址并导航至工程文件夹。
3. 删除重复的工程。
4. 重新导入工程。

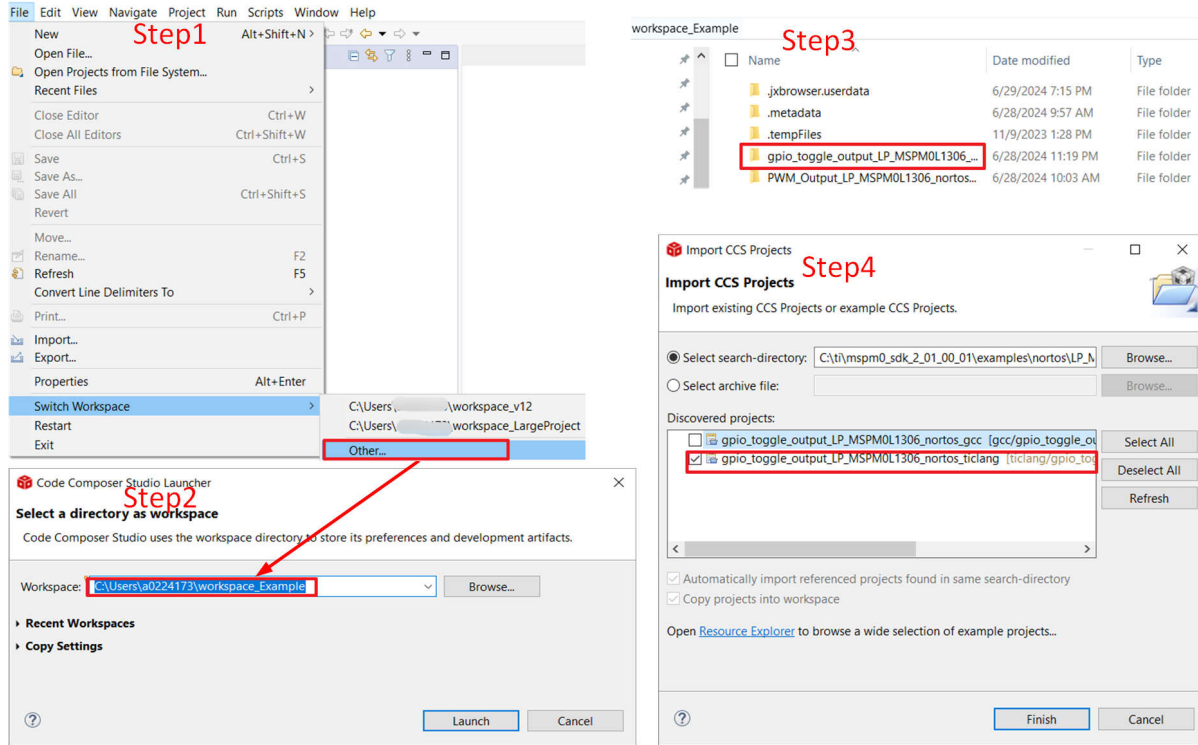


图 7-12. 删除同名工程

### 7.6.3 CCS 找不到 .h 文件

一些用户发现，在调试后，CCS 生成一个错误，指示找不到 .h 文件，而实际上该文件已经在工程中。出现此错误的原因是 CCS 未在导入的文件夹中包括 .c 和 .h 文件，用户需要添加链接。按照以下步骤将文件夹添加到包括地址：

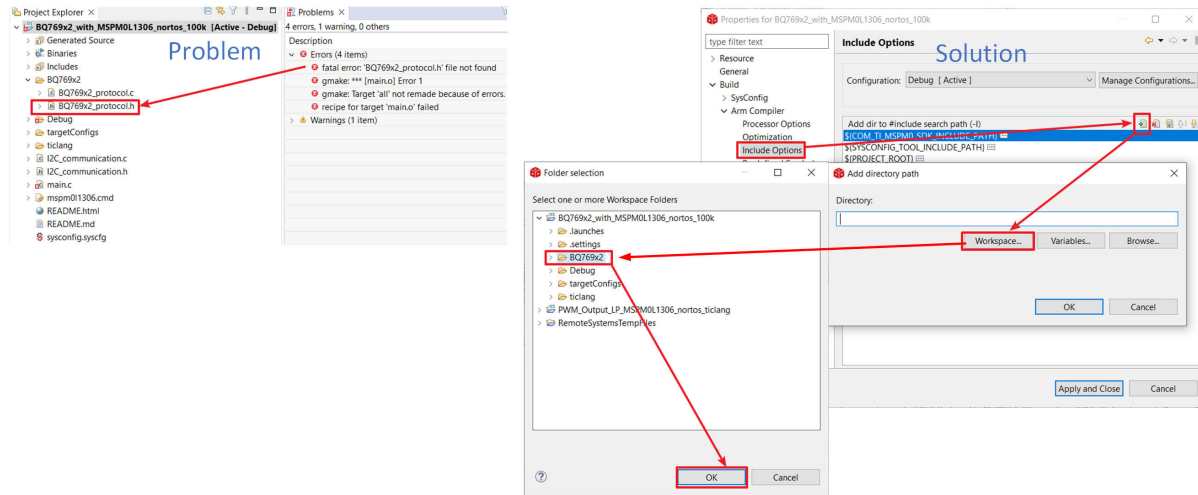


图 7-13. 找不到 .h 文件



### 7.6.4 安装 Arm GCC

MSPM0 SDK 包括支持 TI Arm Clang 和 GCC 的示例；但在 CCS 中默认安装了 TI Arm Clang，而没有安装 GCC。以下是使 CCS 支持 GCC 的步骤。

1. 可通过选择 *Help* → *Install GCC ARM Compiler Tools* 来安装 GCC
2. 选择要安装的版本。CCS 和 MSPM0-SDK 仅包括和支持某些版本的工具链。
3. 如果安装成功，请点击 *Window* → *Preferences*，然后点击 *Code Composer Studio* → *Build* → *Compilers* 来查看 CCS 中安装的编译器列表。

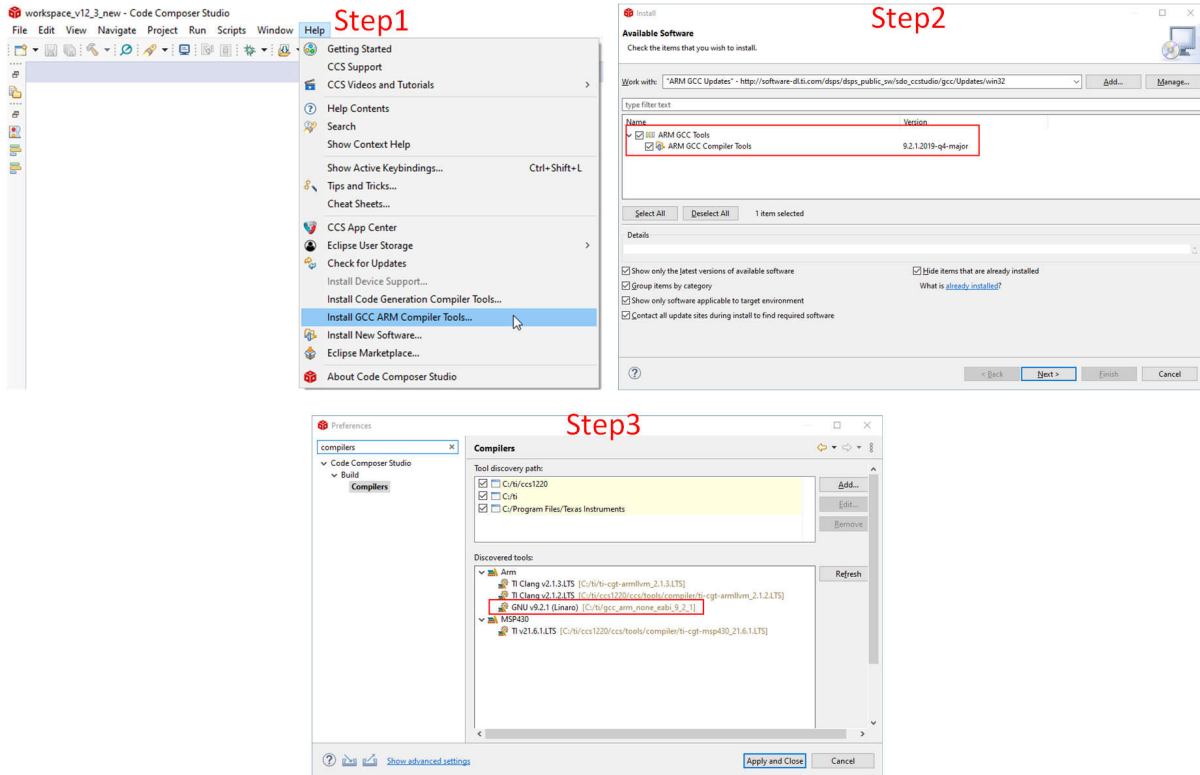


图 7-14. 安装 Arm GCC

### 7.6.5 进入调试模式后器件未连接

有时，在进入调试模式后，可能会出现错误，指示 MCU 未连接且 PC 不会自动跳转到 main 函数。当工程名称更改且 IDE 找不到正确的调试信息时，会发生这种情况。

对于此情况，请删除调试文件夹。右键点击该工程并进入 *Properties*，按照图 7-15 中的步骤还原默认调试设置。

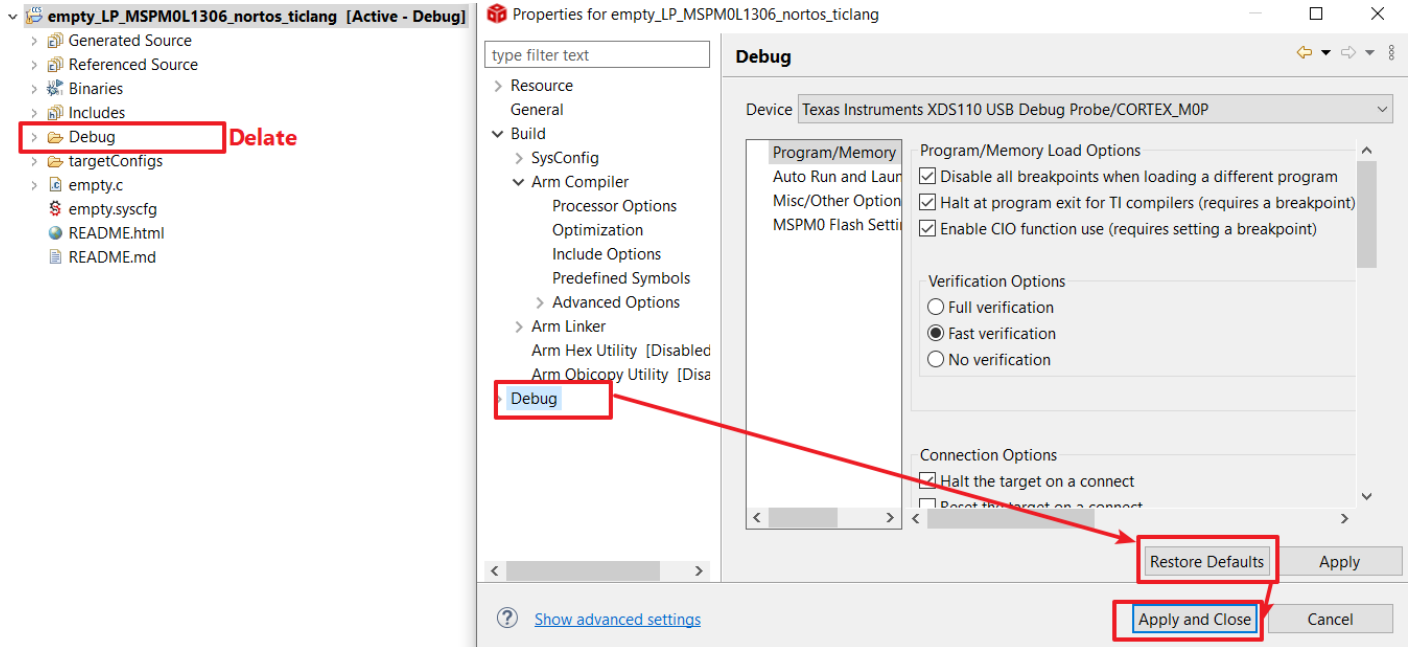


图 7-15. 还原默认调试设置

### 7.6.6 擦除所需的存储器

某些用户希望实现自定义存储器擦除。图 7-16 展示了 *Erase Method* 入口：默认设置为选项 1，该选项会擦除总存储器。当用户希望额外擦除 NONMAIN 时，请选择选项 2。当用户希望保留一些存储器范围不被擦除时，请选择具有声明的存储器范围的选项 3 或选项 4。

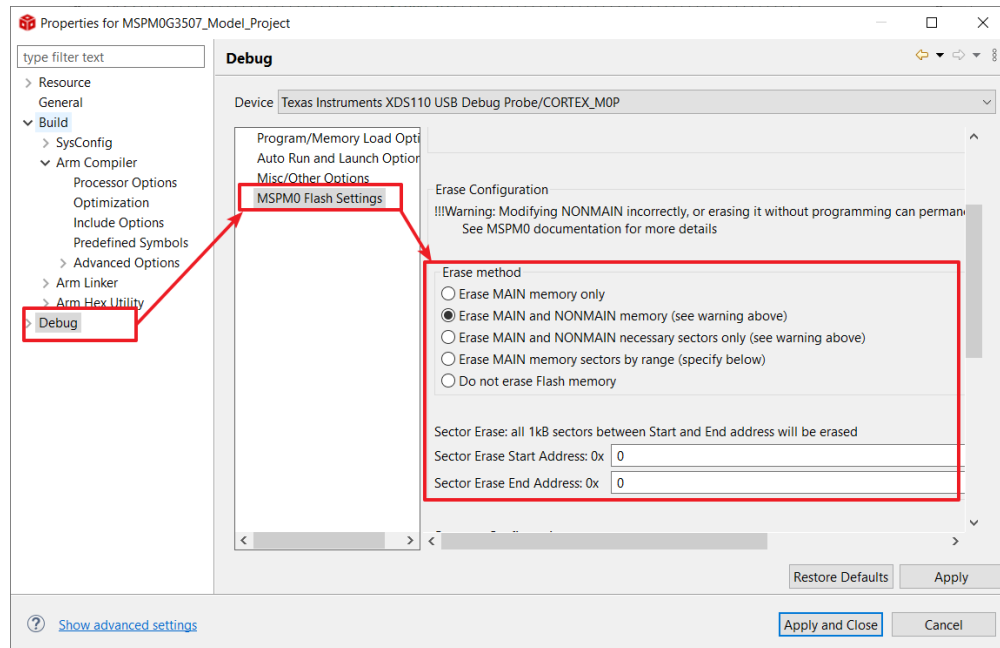


图 7-16. 擦除所需的存储器

### 7.6.7 从 CCS 输出数据日志

有些用户希望将数据从 CCS 输出到桌面以供进一步分析。简单的方法是先将数据保存在全局数组中，然后按照以下说明将数据输出为 Excel 文件。

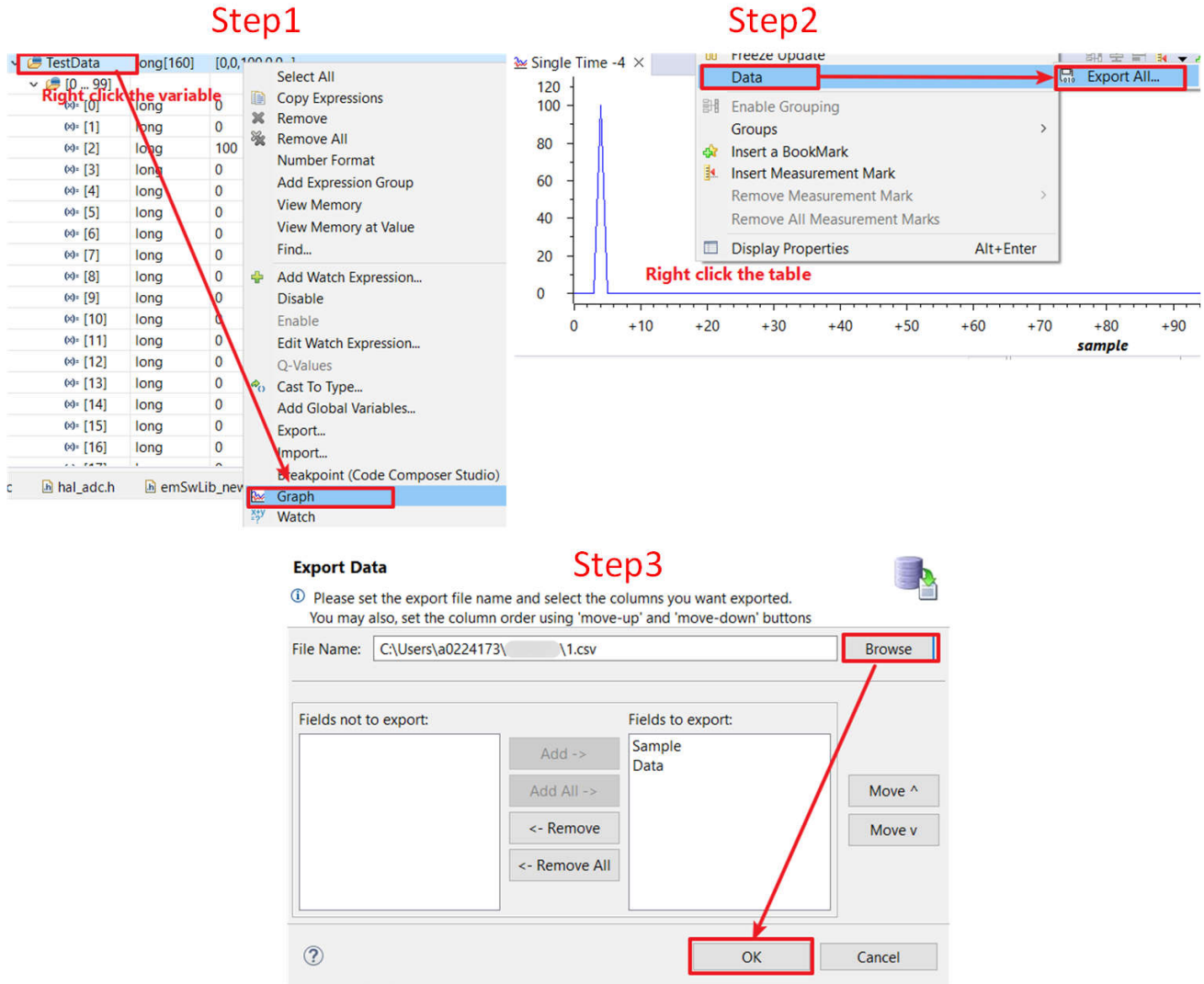


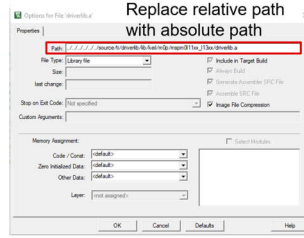
图 7-17. 从 CCS 输出数据日志

## 7.7 Keil 常见问题

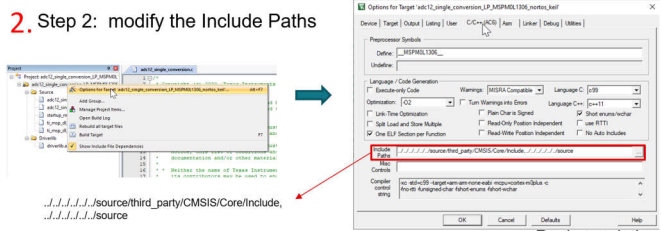
### 7.7.1 从 SDK 复制 Keil 示例

如果从 SDK 复制示例代码并直接编译，则会出现错误。根本原因在于代码示例中的 SDK 和 SysConfig 地址设置。要解决此问题，请参阅图 7-18。

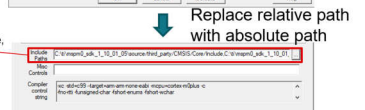
#### 1. Step 1: modify the path of driverlib.a



#### 2. Step 2: modify the Include Paths

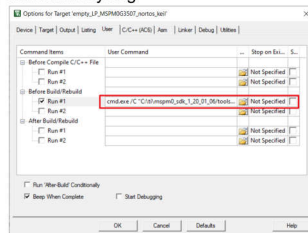
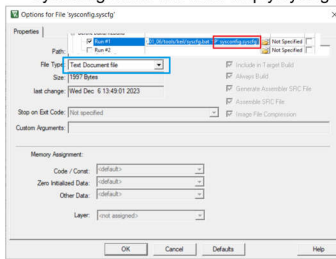


C:\ti\mspm0\_sdk\_1\_10\_01\_05\source\third\_party\CMSIS\Core\Include,  
C:\ti\mspm0\_sdk\_1\_10\_01\_05\source

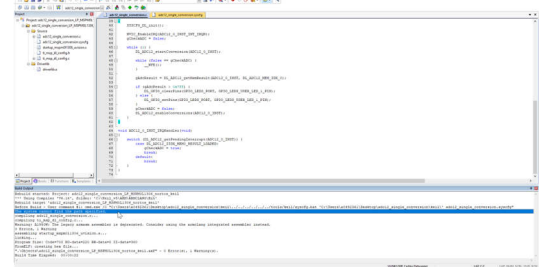


Replace relative path with absolute path

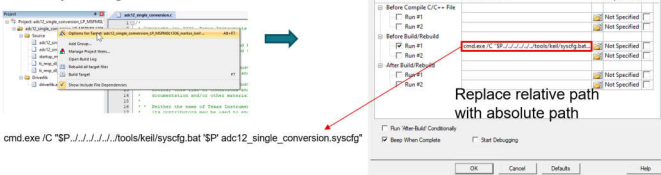
#### 3. If you copy the sysconfig, change the file type to text and may need change the sysconfig name like from empty.syscfg to username.syscfg



#### 4. Now you can compile normally, but SysConfig still cannot be used



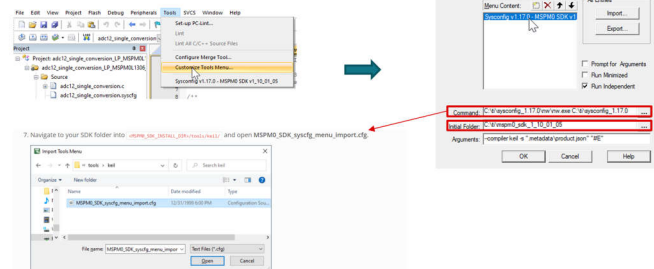
#### 5 Step 3: modify the User command paths for sysconfig



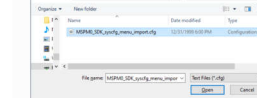
cmd.exe /C "SP".....\tools\keil\syscfg.bat "SP" adc12\_single\_conversion.syscfg

cmd.exe /C "C:\ti\mspm0\_sdk\_1\_10\_01\_05\tools\keil\syscfg.bat" "SP" adc12\_single\_conversion.syscfg

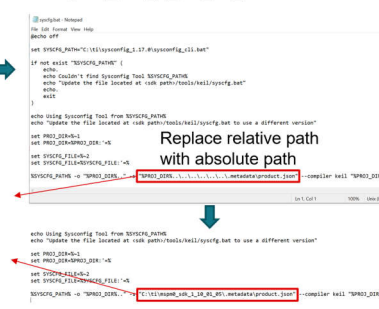
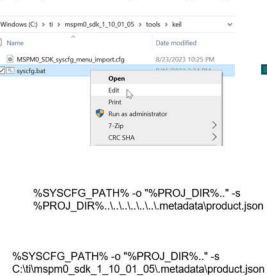
#### 6. Step 4: modify the tools configuration for sysconfig, import from SDK<MSPM0\_SDK\_INSTALL\_DIR>\tools\keil\



7. Navigate to your SDK folder under '.....\tools\keil\..' and open MSPM0\_SDK\_syscfg\_menu\_import.cfg



#### 7 Step 5: modify the syscfg.bat in C:\ti\mspm0\_sdk\_1\_10\_01\_05\tools\keil



#### 8. Now you can use SysConfig normally. Don't put the Keil project under an address name with ". For example, use "My\_path" instead of "My path".

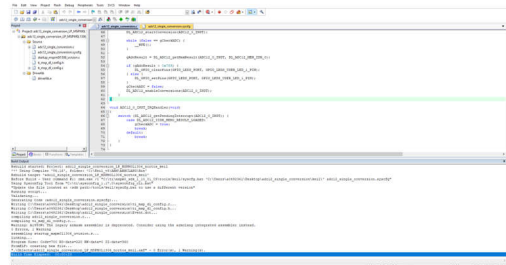


图 7-18. 从 SDK 复制 Keil 示例

## 8 其他信息

### 8.1 点亮 LED 和 CCS 简介

本节从头介绍了如何使用 CCS 点亮 LED。本节还向用户提供了 CCS 的简短说明，以及有关如何使用该工具的说明。

#### 8.1.A 安装 CCS 和 SDK

以下是 CCS 安装的重要步骤和提示。

1. 下载 [CCS \(12.2 以上版本\)](#) 并开始安装，然后继续按下一步。

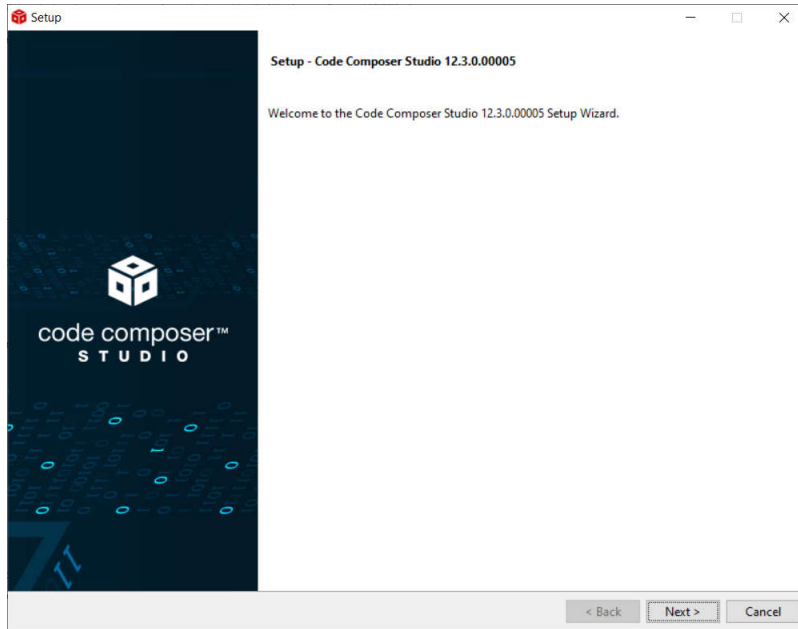


图 A-1. CCS 安装

2. 选择 MSPM0 支持元件。

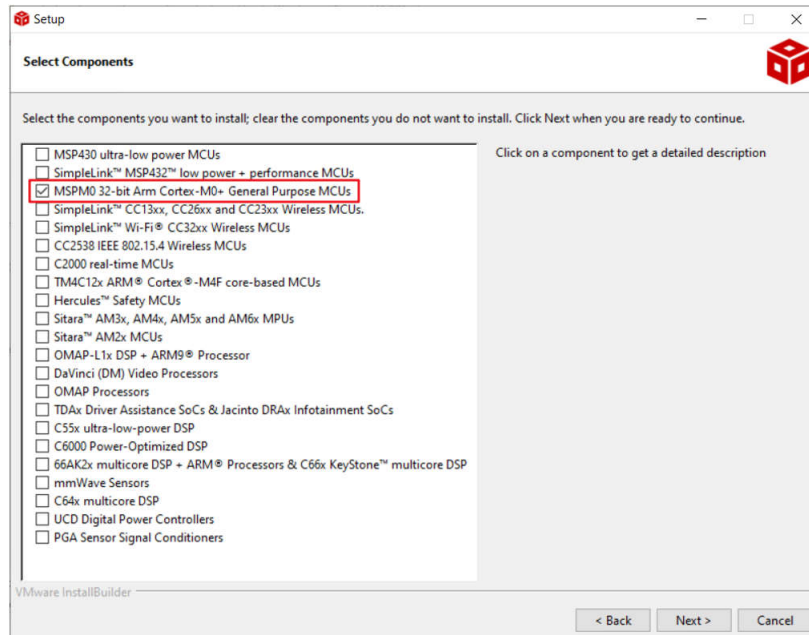


图 A-2. MSPM0 支持选择

3. 如果需要，选择 J-link。

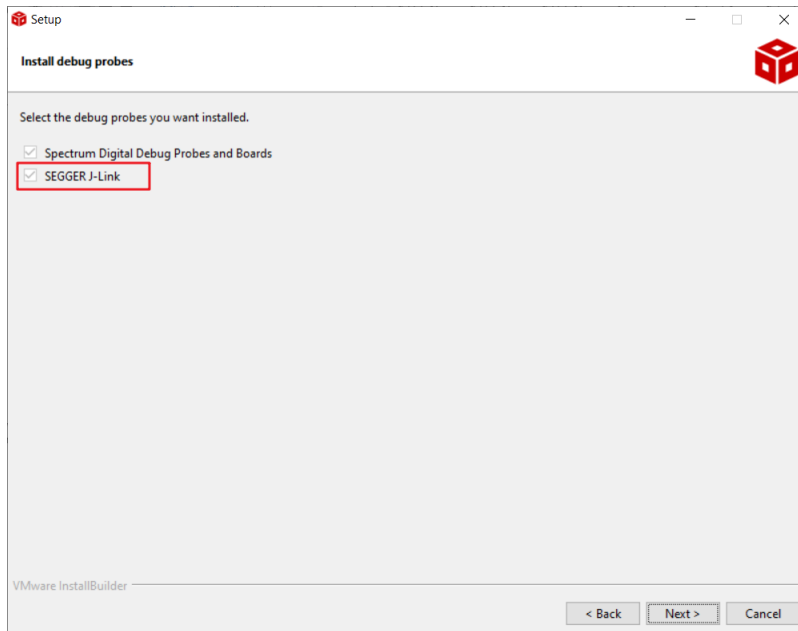


图 A-3. J-link 选择

4. 安装 MSPM0 SDK。

8.1.B 硬件设置

获取 LaunchPad 并插上计算机的电源。

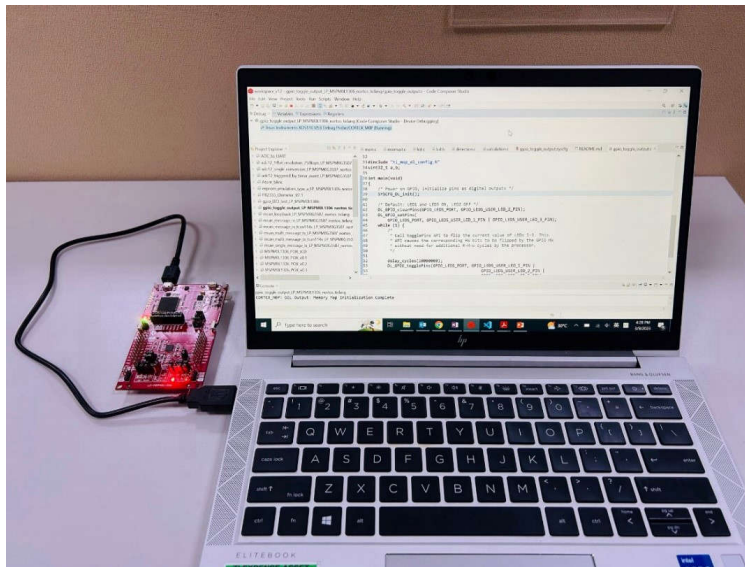


图 8-4. 硬件设置



### 8.1.C 代码导入

1. 打开 CCS。工作区是指复制导入的工程地址。

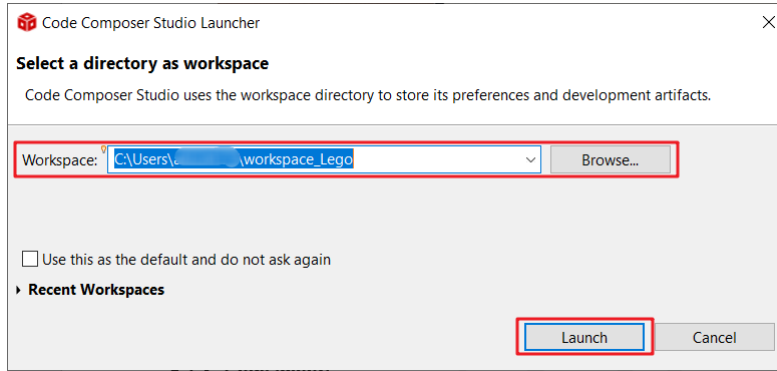


图 8-5. 选择 CCS 工作区

2. 使用 TI-Clang 编译器导入通用输入/输出 (GPIO) 切换工程。

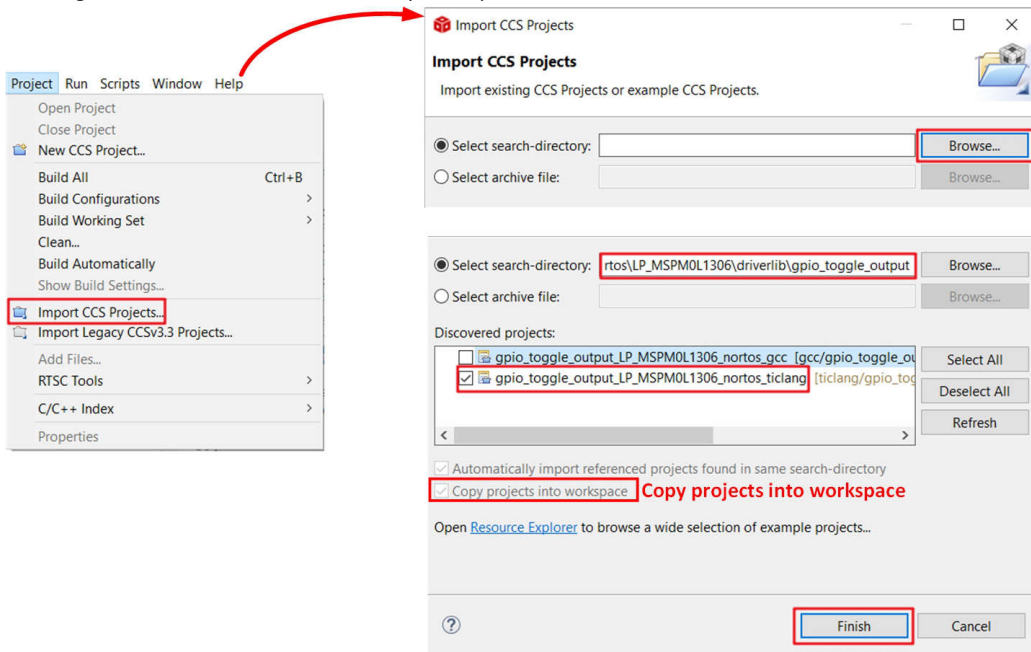


图 8-6. 导入工程

3. 如果无法导入工程，请删除工作区下的同名工程。

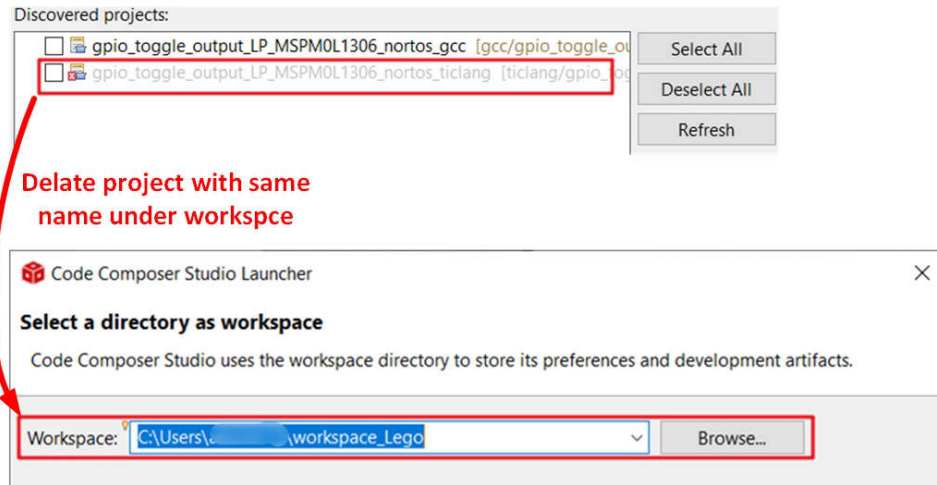


图 8-7. 删除重复的工程

### 8.1.D 调试和 CCS 简介

1. 开始调试，然后您可以看到 LP 上的 GPIO 切换。

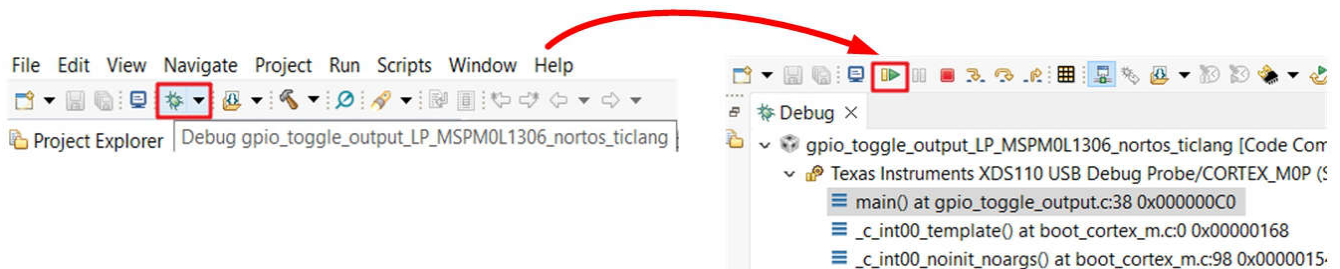


图 8-8. 调试代码

2. 这里我们快速介绍了 CCS 函数。
  - a. 工程属性常用设置：

1. Change code optimization
2. Change .c and .h include address
3. Change .lib include address
4. Set to generate program file
5. Flash erase setting

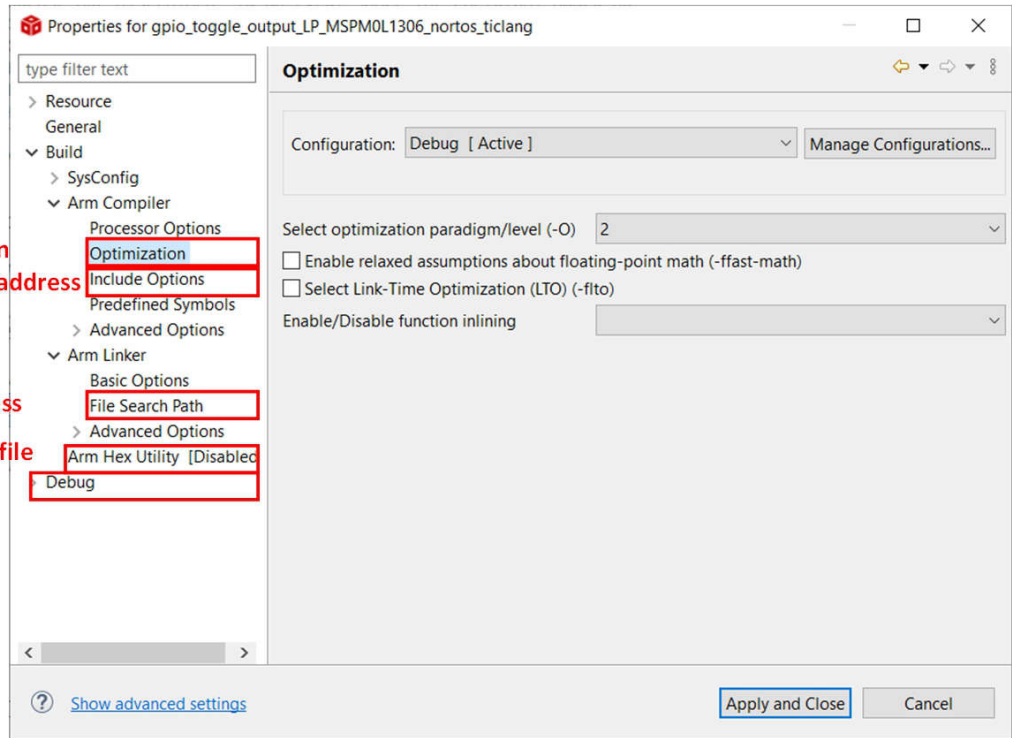


图 8-9. 常用工程设置

b. 调试常用函数。

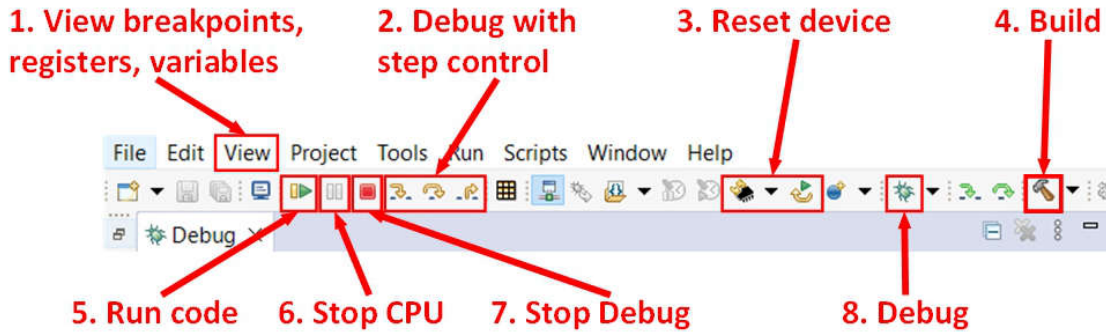
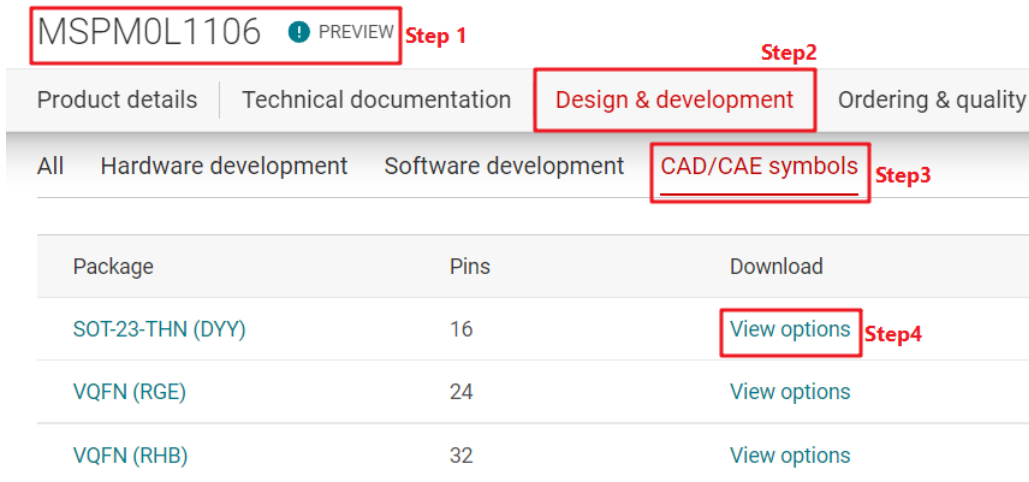


图 8-10. 常用的调试函数

## 8.2 生成 PCB 库的步骤

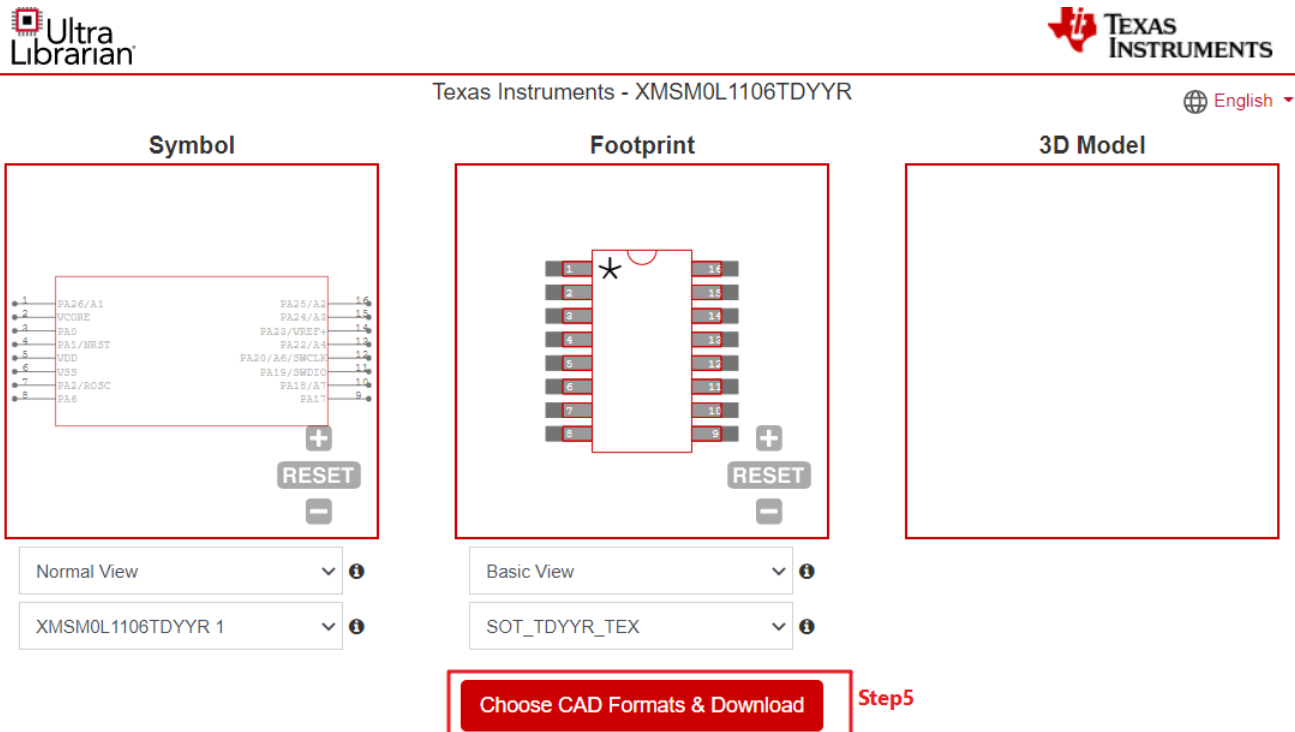
- 使用图 8-11 中显示的步骤转到 MSPM0 器件页面下 Ultra Librarian 工具的开始页面。



Package	Pins	Download
SOT-23-THN (DYY)	16	<a href="#">View options</a> Step4
VQFN (RGE)	24	<a href="#">View options</a>
VQFN (RHB)	32	<a href="#">View options</a>

图 8-11. Ultra Librarian 工具开始页面

- 选择所需的 CAD 格式和引脚顺序，以便获取 Altium 设计库文件。



Symbol: Normal View, XMSM0L1106TDYYR 1

Footprint: Basic View, SOT\_TDYYR\_TEX

3D Model

Choose CAD Formats & Download Step5

图 8-12. Ultra Librarian 工具器件选择

3. 这里以 Altium Designer 库文件为例。

Texas Instruments - XM5M0L1106TDYYR

English

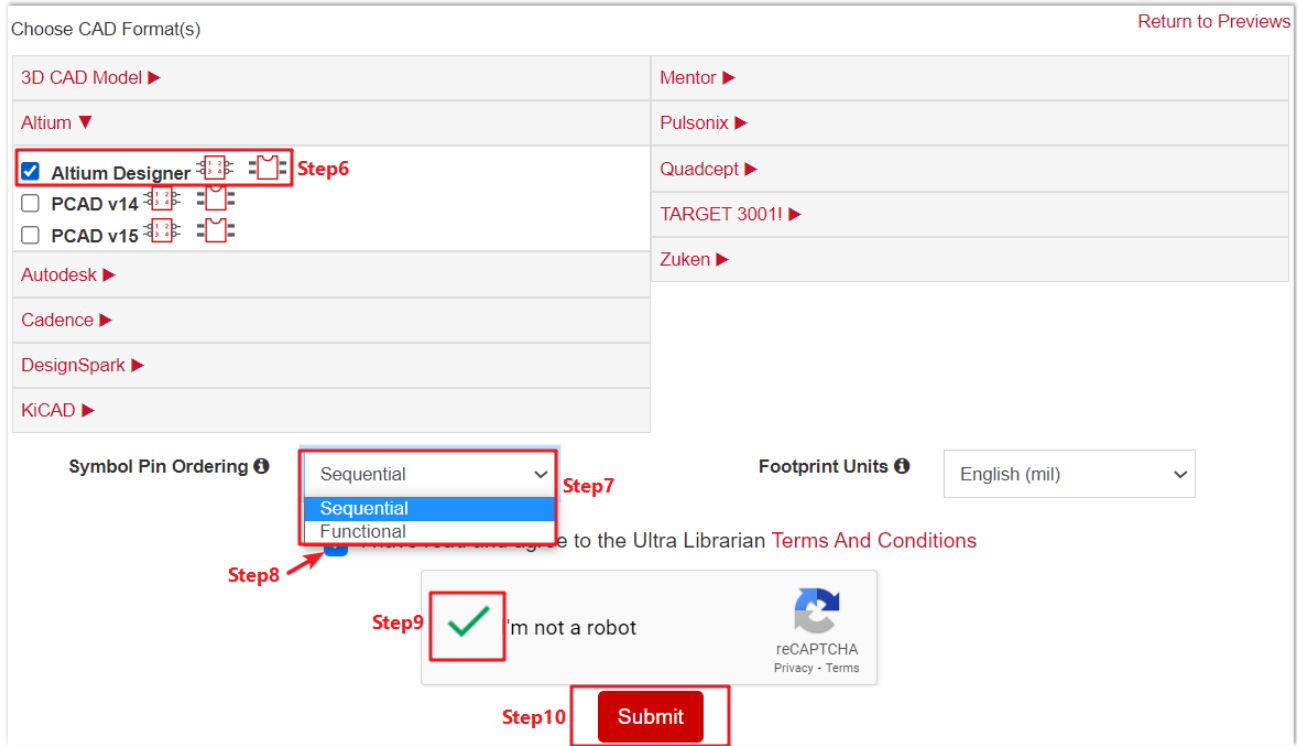


图 8-13. Ultra Librarian 工具 CAD 下载

4. 运行 Altium Designer 脚本，如图 8-14 所示。

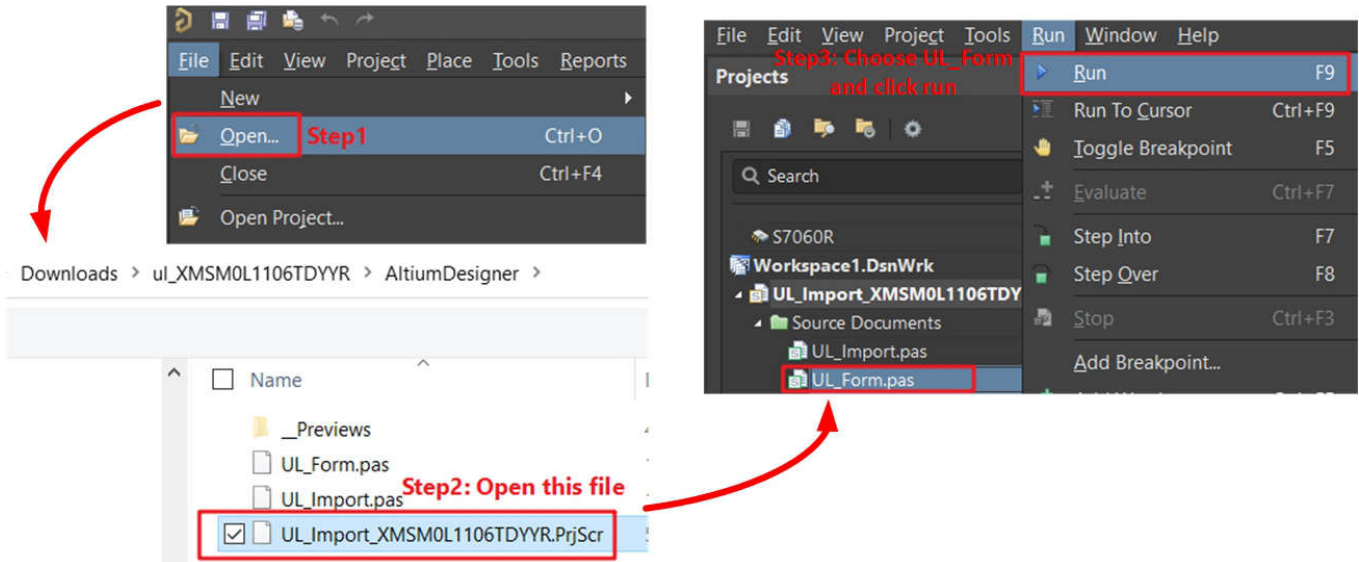


图 8-14. 运行 Altium Designer 脚本

5. 生成 PCB 库和原理图库，如图 8-15 所示。

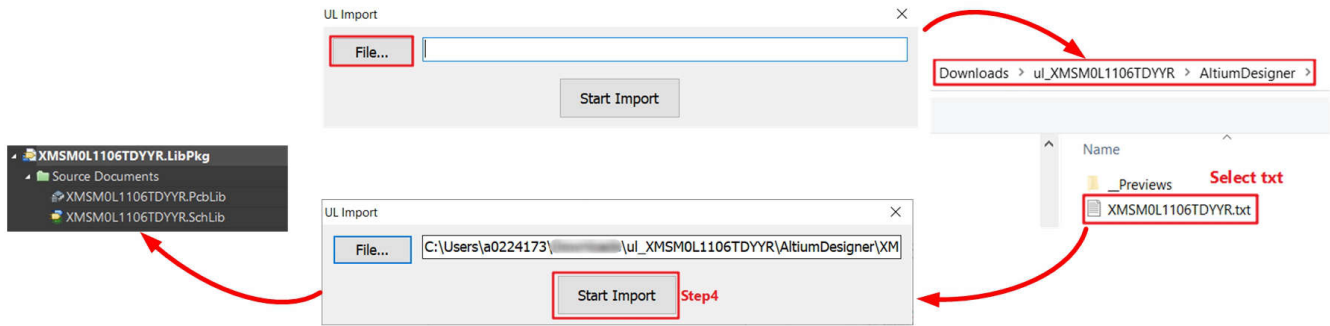


图 8-15. 生成库

6. 在 PCB Library 下面选择正确的封装。

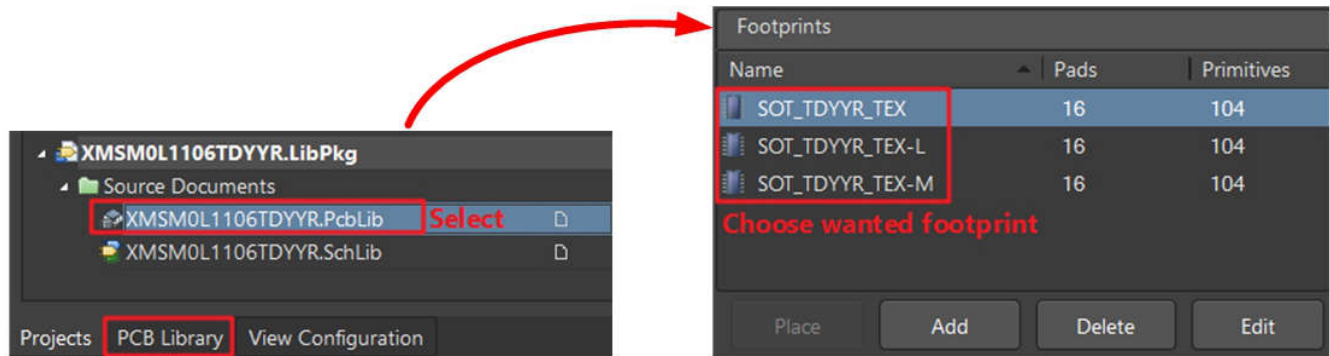


图 8-16. 选择封装

7. 导入 PCB 库和原理图库。

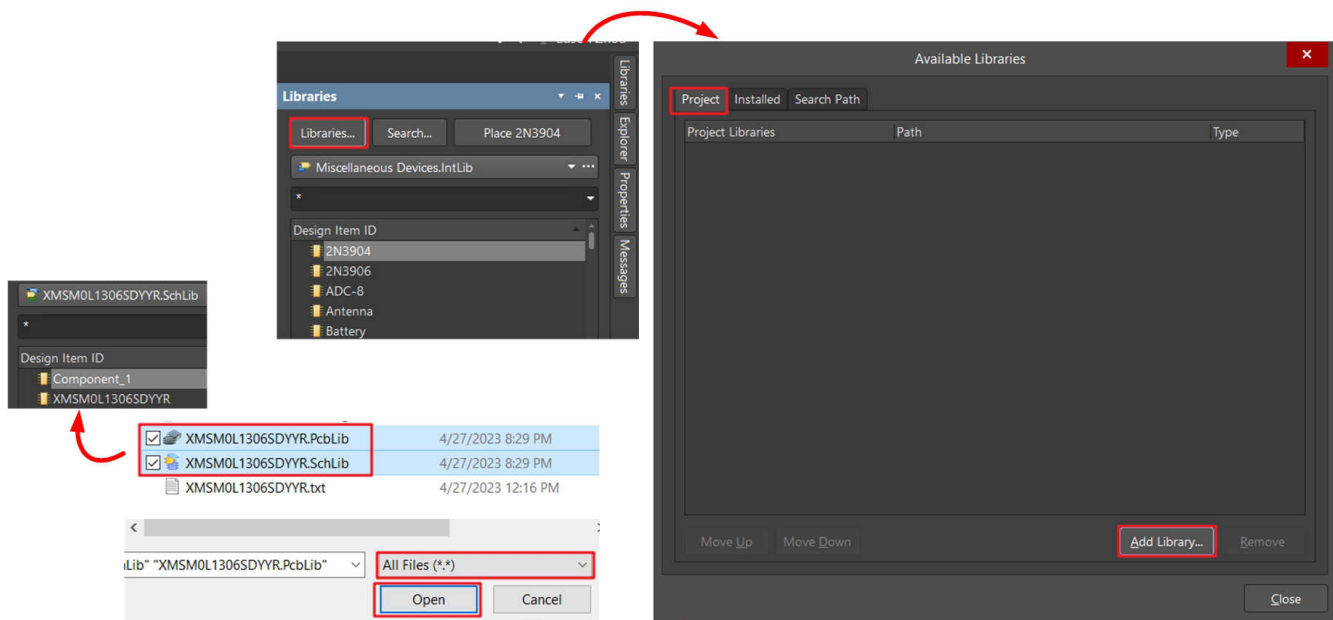


图 8-17. 导入库



### 8.3 MSP-GANG 简介

本节说明了如何使用 MSP-GANG 对 MSPM0 进行离线编程。本节说明了如何使用带 GUI 的 MSP-GANG 对 MSPM0 器件编程。

1. 完成用于软件的引脚连接，如图 8-18 所示。

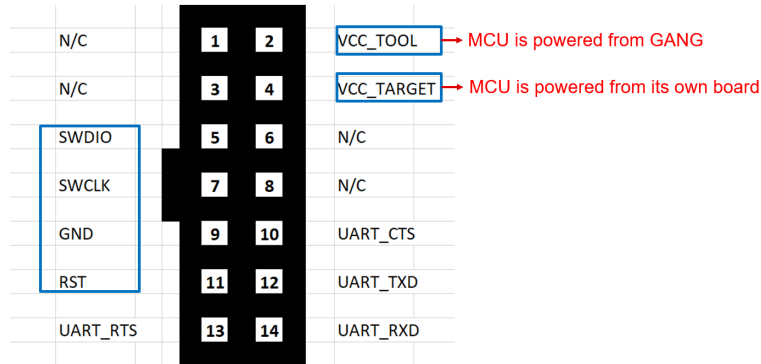


图 8-18. MSP-GANG 引脚分配

2. 硬件设置完成后，按照编程步骤操作。对于步骤 2，请参阅节 5.1 来生成代码文件。对于步骤 4，启用的目标与使用的硬件端口相关，在端口旁边以数字方式来标记端口。

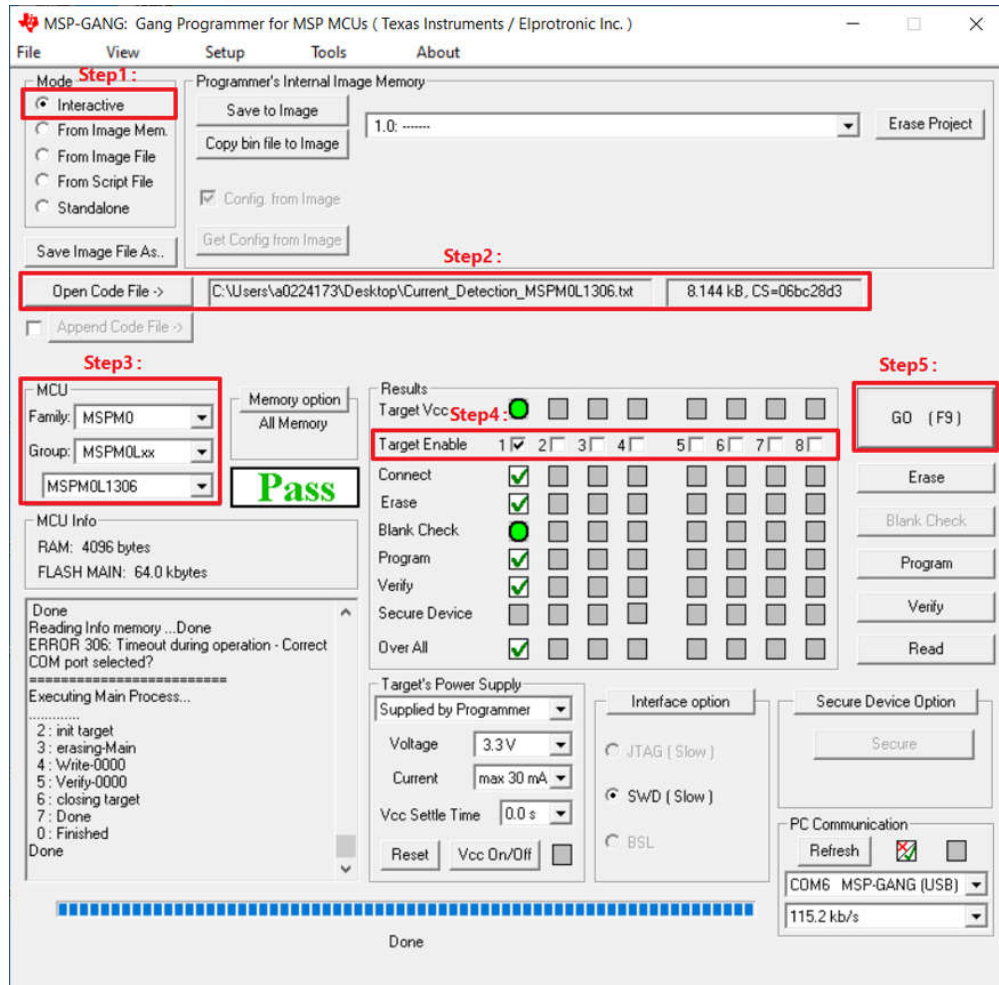


图 8-19. 使用带有 GUI 的 MSP-GANG 下载代码

3. 要更改非主 ( SWD 和 BSL 配置闪存区域 ) 中的代码文件, 请先启用此功能。

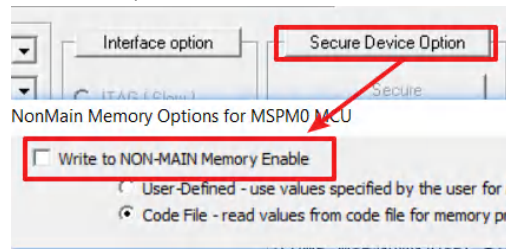


图 8-20. 启用非主编程

4. 将代码文件和设置保存到 MSP-GANG。为此映像指定一个工程名称。然后点击 **Save to Image**, 如图 8-21 所示。

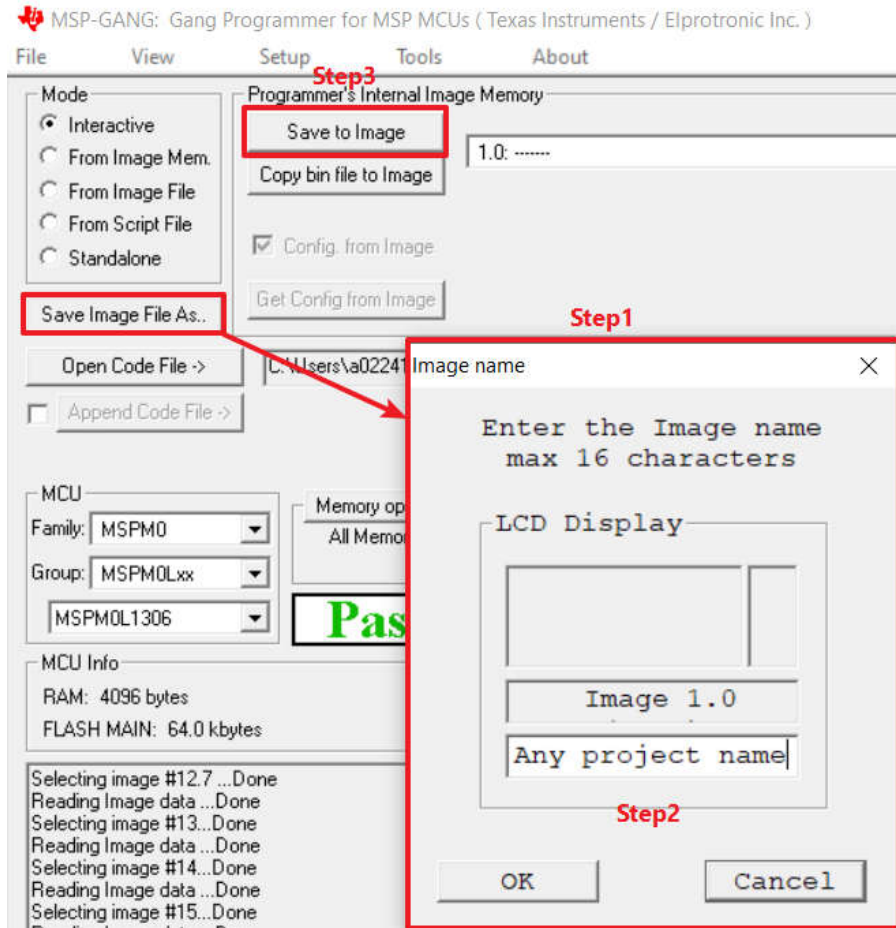


图 8-21. 生成并保存映像

5. 将模式更改为独立或直接关闭 GUI。

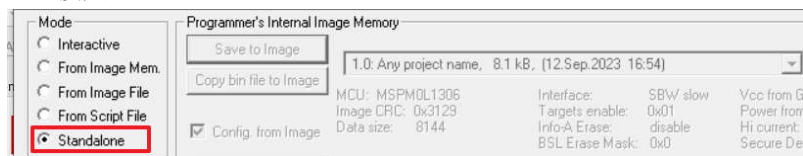


图 8-22. 更改模式

6. 如果 MSP-GANG 中只保存了一个映像，则点击 **Go** 来编程。如果保存了多个映像，请先切换到正确的映像。

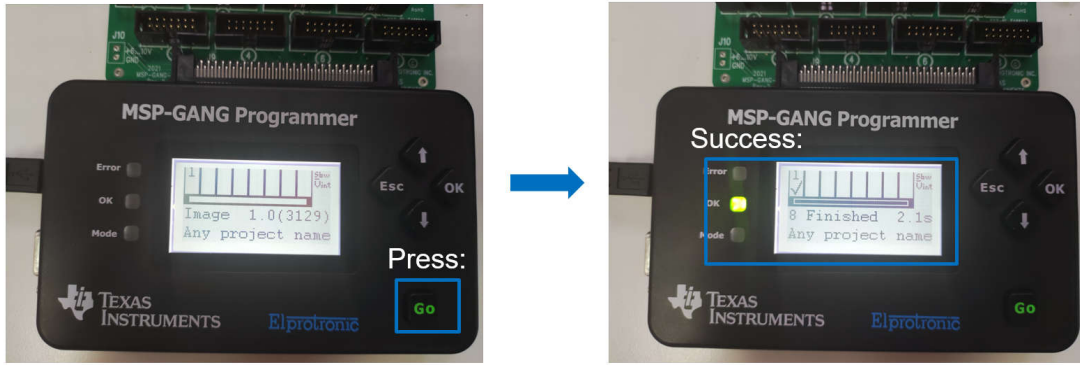


图 8-23. 离线编程

## 9 总结

刚开始进行 MSPM0 开发时请参考本文档，该文档提供了 MSPM0 生态系统概述和分步说明，还为用户提供了清晰的流程和图像说明。除了基本知识之外，该文档还列出了参考资料和进一步阅读材料，供用户进一步参考。TI 建议用户参照本文档快速进行 MSPM0 开发并克服常见障碍。

## 修订历史记录

Changes from Revision C (July 2024) to Revision D (October 2024)	Page
• 添加了 <i>质量和可靠性说明</i> 部分.....	55
• 在 <i>解锁 MCU</i> 部分添加了有关解锁 MSPM0 的更多信息.....	57
• 在从 <i>CCS 输出数据日志</i> 部分中添加了有关 CCS 的新常见问题.....	67

## 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  
Copyright © 2024，德州仪器 (TI) 公司