

ABSTRACT

This document is a good resource for finding important information about MSPM0[™] microcontrollers (MCUs). This application note can serve as a reference, a starting guide, a self-learning tool, or an application-development guide.

Table of Contents

1 Overview	4
2 MSPM0 Online Selection	5
3 Software Development Instructions	7
3.1 LaunchPad Setup	7
3.2 MSPM0-SDK Setup	9
3.3 SysConfig Setup	14
3.4 IDE Quick Start	20
4 Hardware Design Instructions	43
4.1 Obtaining a MSPM0 Package	43
4.2 Fix Pin Functions through Sysconfig	46
4.3 Schematic and PCB Attentions	
5 Mass Production Instructions	
5.1 Generate Production Image	
5.2 Program Software Tools Quick Start	49
5.3 Program Hardwares Quick Start	57
6 Quality and Reliability Instructions	59
6.1 Quality and Reliability Material Entrance	59
6.2 Failure Information Collection and Analysis Guidance	59
7 Common Development Questions.	60
7.1 Unlock MCU	60
7.2 MSPM0 Program Failure	<mark>64</mark>
7.3 Reprogram with SWD Disabled	<mark>64</mark>
7.4 MCU Performs Differently in Debug and Free Run	<mark>65</mark>
7.5 Set SWD Password	<mark>65</mark>
7.6 BSL Related Questions	<mark>67</mark>
7.7 Reach Expected Current in LPM Mode	<mark>67</mark>
7.8 CCS Common Questions	<mark>67</mark>
7.9 Keil Common Questions	<mark>69</mark>
8 Summary	70
9 Technical Documentation Resources	70
9.1 Technical Reference Manuals	70
9.2 Subsystems	70
9.3 Reference Designs	71
9.4 Hardware EVM User's Guides	71
9.5 Application Notes and Others	71
10 Revision History	74

List of Figures

Figure 1-1. MSPM0 Ecosystem	4
Figure 2-1. MSPM0 Device List	5
Figure 2-2. MSPM0 Important Document List	5
Figure 2-3. Device Comparison Table	6
Figure 2-4. Ordering and Quality Part View	6

1

Figure 3-1, MSPM0G3507 LaunchPad	8
Figure 3-2. Launchpad Setup View	8
Figure 3-3, MSPM0-SDK Download	9
Figure 3-4 MSPM0-SDK Install Step-by-Step	9
Figure 3-5 MSPM0-SDK Structure	10
Figure 3-6 RTOS and Nortos Code Examples	. 11
Figure 3-7 SysConfig Install	14
Figure 3-8 MSPM0 SysConfig	15
Figure 3-9 SysConfig View	15
Figure 3-10 Basic Operations	16
Figure 3-11 Project Configuration	16
Figure 3-12 Board View	17
Figure 3-13 NONMAIN View	17
Figure 3-14 SYSCTI View	18
Figure 3-15 Perinherals View	10
Figure 3-16, CCS Key Install Steps	21
Figure 3-17 Import CCS Project	22
Figure 3-18 CCS Project Overview	22
Figure 3-10. Coorde Debugger Selection	
Figure 5-19. Grange Debugger Selection	20
Figure 5-20. Durid, Debug and Kuri the Gode.	
Figure 5-21: Commonly Used Debug Functions	
Figure 3-22. Migrating Detween Mor Mu Derivatives	.20
Figure 3-23. Generate mex File	. 20
Figure 3-24. PTOgramming NONMAIN	
Figure 3-25. Add MISFMU SDK ID IAR.	
Figure 3-20. Install SysColling for MSPMU.	20
Figure 5-27. Import a SDK Example	
Figure 5-20. Use Systeming With IAR.	. 29
Figure 3-29. Download and Debug	30
Figure 3-30. Migrating Between MSPMU Derivatives.	.31
	. 32
Figure 3-32. Program NUNIMAIN	32
Figure 3-33. Open Fack Installer	. 33
Figure 3-34. Search Device	.33
Figure 3-35. Install Device Pack.	. 34
Figure 3-36. Approve the License.	
Figure 3-37. Edit Syscig.bal.	.35
Figure 3-38. Edit MSPINU_SDK_syscig_menu_import.crg	. 35
Figure 3-39. Kell Customize Tools	. 35
Figure 3-40. Import MSPMU_SDK_sysctg_menu_Import.ctg File	. 36
Figure 3-41. Finish Systeming Setup.	. 30
Figure 3-42. Open Project.	. 36
Figure 3-43. Select Kell Project.	. 37
Figure 3-44. Open sysctg file	38
Figure 3-45. Open Options for Target.	.38
Figure 3-46. Select the Debug Pane.	. 39
Figure 3-47. Check the Setting of XDS110 Probe.	. 39
Figure 3-48. Check the Setting of J-Link Probe	40
Figure 3-49. Flash Download Setting	. 40
Figure 3-50. Download Project	.41
Figure 3-51. Build RTOS Example Under Keil	41
Figure 3-52. Migrating Between MSPM0 Derivatives.	.42
Figure 3-53. Generate Hex Files.	. 42
Figure 3-54. Program NONMAIN.	43
Figure 4-1. Ultra Librarian Tool Start Page	. 43
Figure 4-2. Ultra Librarian Tool Device Selection	. 43
Figure 4-3. Ultra Librarian Tool CAD Download	44
Figure 4-4. Run Altium Designer Script	.44
Figure 4-5. Generate Library	. 45
Figure 4-6. Select Footprint.	. 45
Figure 4-7. Import Library	. 45
Figure 4-8. Generate Peripherals and Pin Assignments File	46
Figure 4-9. MSPM0 Minimum System	.46

Figure 4-10 MSPM0 Schematic	47
Figure 5-1 Program Software and Tools	48
Figure 5-2. Program Through SWD	49
Figure 5-3. Program Through Bootloader.	
Figure 5-4. J-Flash Quick Start	
Figure 5-5. GangPro-ARM Install	52
Figure 5-6. C-Gang Pin Assignment	53
Figure 5-7. Online Program	54
Figure 5-8. Enable Non-Main Programming	55
Figure 5-9. Save Image	55
Figure 5-10. Go Button Setting	<mark>56</mark>
Figure 5-11. Offline Downloading	<mark>56</mark>
Figure 5-12. Pin Connection of TMDSEMU110-U	57
Figure 5-13. XDS110 Onboard	57
Figure 5-14. LP-XDS110ET	<mark>58</mark>
Figure 7-1. E2E Online	60
Figure 7-2. CCS Error	60
Figure 7-3. Unlock Through GUI	62
Figure 7-4. Unlock Through Uniflash	63
Figure 7-5. Unlock Through CCS	63
Figure 7-6. Device Manager View	64
Figure 7-7. Disable BSL	65
Figure 7-8. Enable SWD Password	
Figure 7-9. Clear SWD Password	67
Figure /-10. Change Optimization Level.	68
Figure 7-11. Copy Kell Example Out of SDK	69

List of Tables

Table 3-1. MSPM0 Development Chain	7
Table 3-2. MSPM0 Debugger Comparison	7
Table 3-3. MSPM0 Example Coverage	11
Table 3-4. MSPM0 Supported IDEs Overview	20
Table 5-1. Product File Generated by IDE	
Table 5-2. XDS110 Debugger Summary	57
Table 7-1. Tools Suggested Version	60
Table 7-2. Unlock Commands	<mark>61</mark>
Table 7-3. Unlock Method Selection	61

Trademarks

MSPM0[™], LaunchPad[™], Code Composer Studio[™], SimpleLink[™], C2000[™], and TIVA[™] are trademarks of Texas Instruments.

Arm® and Cortex® are registered trademarks of Arm Limited.

are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All trademarks are the property of their respective owners.



1 Overview

The MSPM0 microcontroller family uses an enhanced Arm[®] Cortex[®]-M0+ 32-bit processor, operating at up to 80MHz and supports both industrial and automotive applications (with AEC-Q100 FS-QM and ASIL-B qualification). Designers can easily find a cost-effective MCU within the broad portfolio, which offers pin-to-pin compatibility across a wide range of memory and package sizes. TI's leadership in integrated precision analog endows this device family with the high precision and speed ADC, zero-drift chopper OPA, DAC, COMP, and so forth.

MSPM0 MCUs are supported by an extensive hardware and software ecosystem. The ecosystem includes easyto-use development tools, affordable evaluation boards, and a wide range of embedded software kits, drivers, and examples. This document describes these factors into four topics: MSPM0 Online Selection, Software Development Instructions, Hardware Design Instructions and Mass Production Instructions.



Figure 1-1. MSPM0 Ecosystem

Besides of common development topics, we also list all the Technical Documentation Resources, quality and reliability resources and common development questions. Please refer to the table of contents to choose your interested topic for reference.



2 MSPM0 Online Selection

This step discusses how to find an MSPM0 orderable number.

Visit the Arm Cortex-M0+ MCUs product page to view the list of MSPM0 devices. After navigating to this page, use the filters on the left to perform an initial screening based on MCU peripheral requirements, or directly navigate to the device page using the search box on the left side of the page.

<mark> </mark>	eset table 17 of 17 total products
Search	Product number
✓ Description	*
✓ TI.com inventory	✓
✓ CPU	Data sheet: PDF HTML
✓ Frequency (MHz)	MSPM0G1107 - NEW
✓ Flash memory (kByte)	Data sneet: PDF HIML
✓ RAM (kByte)	✓ MSPM0G1505 - NEW Data sheet: PDF HTML
✓ ADC type	View alternates

Figure 2-1. MSPM0 Device List

After navigating to the device page, more specification or functional details for a specific product are available. The key documents are the data sheet, technical reference manual (TRM), and errata. The device-specific data sheet introduces the parameters and functional data information for the MSPM0. The device-specific TRM introduces the application method and characteristics of a MSPM0 device. The device-specific errata shows descriptions of MSPM0 related series or versions.

NEW	
MSPM0L1306 I PREVIEW	
32-MHz Arm® Cortex®-M0+ MCU with 64-KB flas	h, 4-KB SRAM, 12-bit
ADC, comparator, OPA	
DATA SHEET MSPM0L130x Mixed-Signal Microcontrollers datasheet (Rev. A) PDF HTML	
USER GUIDES	ERRATA
MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual (Rev. B)	MSPM0 Microcontrollers Errata

Figure 2-2. MSPM0 Important Document List

5

Figure 2-3 shows the *Device Comparison* table in a device-specific data sheet. A user can compare different part numbers using this table.

5 Device Comparison

				Table 5	-1. Devid	e Compa	rison											
DEVICE NAME (1) (2)	FLASH / SRAM (KB)	QUAL ⁽³⁾	ADC CH.	СОМР	OPA	GPAMP	UART/I2C/SPI	TIMG	GPIOs	5-V TOL. IO	PACKAGE [BODY SIZE] (4)							
MSPM0L1306xRHB	64 / 4																	
MSM0L1305xRHB	32 / 4	T/S	10	1	2	1	2/2/1	4	28	2	32 VQFN [5 mm × 5 mm] ⁽⁵⁾							
MSM0L1304xRHB	16/2		1															[]
MSPM0L1306xDGS28	64 / 4																	
MSPM0L1305xDGS28	32 / 4	T/S	10						24		22,1/22,22							
MSPM0L1304xDGS28	16/2					1	2	1	2/2/1	4		2	28 VSSOP [7.1 mm × 3 mm]					
MSPM0L1346xDGS28	64 / 4	т	0]					22		for a straid							
MSPM0L1345xDGS28	32/4		Э	Э						22								

Figure 2-3. Device Comparison Table

See the Ordering and Quality page on the device page to view the orderable part number and the reference price.

MSPM0L1306	PREVIEW		Datas	Sheet Order now
Product details Techr	nical documentation Design & dev	elopment Ordering & qual	ity Support & training	
	Order	ing & quality		
Part number ↓↑	Buy	TI.com inventory ↓↑	Qty Price (USD) ↓↑	Package qty ↓↑ Carrier
XMSM0L1306SDGS20R	Enter quantity Add to cart Limit: 5	93	1ku 🛛 🗸	1 LARGE T&R
XMSM0L1306SDGS28R	Enter quantity Add to cart Limit: 10	170	1ku 🗸 🗸	5,000 LARGE T&R

Figure 2-4. Ordering and Quality Part View



3 Software Development Instructions

Table 3-1 lists a summary of all the required components in an MSPM0 development chain. Devices are described individually in the following sections. Users can also refer to Section 7, when encountering problems in MSPM0 development.

IDE	SysConfig (Code Generator GUI)	SDK	Debugger	Hardware	
CCS with SysConfig integrated			Launchpad with XDS110 On-Board		
Keil	Standalana SvaCanfig	MSPM0 SDK	XDS110	Customized board	
IAR	Stanualone Sysconing		J-Link		

 Table 3-1. MSPM0 Development Chain

3.1 LaunchPad Setup

3.1.1 Debugger Selection

This section summarizes different debuggers that support MSPM0 devices. The XDS110 debuggers are owned by TI, which support more functions, as compared to general debuggers. For more details about XDS110 debuggers, see Section 5.3.

Features	XDS110 (TMDSEMU110-U)	XDS110 On-Board	J-Link
cJTAG (SBW)	\checkmark	\checkmark	\checkmark
BSL tool	\checkmark	\checkmark	
Backchannel UART	\checkmark	\checkmark	
Power supply	1.8 - 3.6V	3.3 - 5V	5V
IDE	CCS, IAR, Keil	CCS, IAR, Keil	CCS, IAR, Keil

Table 3-2. MSPM0 Debugger Comparison

7



3.1.2 LaunchPad Introduction

TI recommends to start MSPM0 development with LaunchPad[™]. Figure 3-1 shows an overview of the LaunchPad. The LaunchPad contains the MCU and a XDS110 debugger. A user can use a debugger such as a J-Link to debug the MCU after removing the jumpers.



Figure 3-1. MSPM0G3507 LaunchPad

A real LaunchPad setup condition is shown in Figure 3-1, which can be debugged and powered with a USB port.



Figure 3-2. Launchpad Setup View

For all the orderable LaunchPad, refer to Arm® Cortex ®-M0+ MCUs design & development webpage. All the LaunchPad user's guides are also listed in Section 9.4.



3.2 MSPM0-SDK Setup

The MSPM0-SDK provides the ultimate collection of software, tools, and documentation to accelerate the development of applications for the MSPM0 MCU platform. The MSPM0-SDK provides a consistent and cohesive experience with a wide variety of drivers, libraries, and examples under a single software package.

3.2.1 MSPM0-SDK Installation

This section details steps to install MSPM0-SDK. After installation, the default SDK directory path is: *C:\ti\mspm0_sdk_x_xx_xx_x*.

- 1. Before downloading, a myTl account is required. Register for a myTl account here.
- 2. Download the latest MSPM0-SDK from the product page. Click *Download options*, select the operating system, and click the file name to start downloading.

MSPM0-SDK	Downloads
Overview Downloads Support & training	
Downloads	
SOFTWARE DEVELOPMENT KIT (SDK) MSPM0-SDK - MSPM0 Software De Supported products & hardware	evelopment Kit (SDK)
Downloads Supported products & hardware	
mspm0_sdk_1_20_01_06.exe - 131070 K	MSPM0 SDK for Winodws MD5 checksum 5710ff880e1e41086aebd00e32e6f87d
mspm0_sdk_1_20_01_06.run − 131243 K	MSPM0 SDK for Linux MD5 checksum a51be1f1808e8323cee09e81ff2d024a
â mspm0_sdk_1_20_01_06.app.zip - 133676 K	MSPM0 SDK formacOS MD5 checksum 07c0b90119899839ae6dfc26aba7b5cc

Figure 3-3. MSPM0-SDK Download

3. After downloading, follow the steps in Figure 3-4 to finish installation.

🝯 Setup	- 🗆 ×	🖷 Setup —		🚿 Setup	- 🗆 ×
	Setup - mspm0_sdk 1.20.01.06	License Agreement 🐺 TEXAS INST	RUMENTS	Installation Directory	🐺 Texas Instruments
	Welcome to the mspm0_sdk 1.20.01.06 Setup Wizard.	Please read the following License Agreement. You must accept the terms agreement before continuing with the installation.	of this	Please specify the direct	tory where mspm0_sdk 1.20.01.06 will be installed.
Texas Instruments		Texas Instruments Incorporated Software License Agreement	^	mspm0_sdk_1_20_01_06	5
		IMPORTANT - PLEASE CAREFULLY READ THIS AGREEMENT, WHICH DISPLAYED FOR YOU TO READ PRIOR TO USING THE LICENSED MA (DEFINED BELOW). YOU WILL BE ASKED WHETHER YOU ACCEPT M ITS TERMS. BY CLICKING "I ACCEPT" OR "I ACREE," OR BY US	IS ATERIALS ID AGREE TO SING THE	Installation Directory	
		LICENSED MATERIALS. YOU (1) REPRESENT THAT YOU HAVE THE Do you accept this license? I laccept the agreement O I do not accept the agreement	LEGAL V		\sim
	< Back Next > Cancel	HistanBuilder	Cancel	InstallBuilder	< Back Next > Cancel
📹 Setup	– 🗆 X	📲 Setup —		🚿 Setup	– 🗆 ×
Ready to Install	🔱 Texas Instruments	Installing TEXAS INST	RUMENTS		Completing the mspm0_sdk 1.20.01.06 Setup Wizard
Setup is now ready to be	gin installing mspm0_sdk 1.20.01.06 on your computer.	Please wait while Setup installs mspm0_sdk 1.20.01.06 on your computer.			Setup has finished installing mspm0_sdk 1.20.01.06 on your computer.
		Installing Unpacking C\tt\mspm0(]oxygen\api_guide\html\foc_types_8h	_incl.md5	TEXAS INSTRUMENTS	View MSPM0 SDK Release Notes
InstallBuilder	< Back Next > CanCE	InstallBuilder	Cancer		< Back Finish Cancel

Figure 3-4. MSPM0-SDK Install Step-by-Step



3.2.2 MSPM0-SDK Introduction

The SDK install directory contains five folders (Figure 3-5). This section briefly introduces each folder.

- Docs folder: Contains all the documentation for SDK.
- Examples folder: Contains all the examples for reference, which can be used to provide a reference and starting point to accelerate application development. For more details, see the MSPM0-SDK Example Guide.
- Kernel folder: Built files for RTOS and nortos, which is included in the example project and accelerates the speed of the project build.
- Source folder: Contains all the source code for TI and third party libraries.
- Tool folder: Contains all the tools related to SDK, such as sysconfig support files, BSL GUI, and metrology GUI.



Figure 3-5. MSPM0-SDK Structure

The most important folders are example and document folders, which are introduced in the following sections.

3.2.2.1 Examples Folder Introduction

TI manufactures a LaunchPad for one MSPM0 sub family with a superset MSPM0 on board. The same example code can be reused across this MSPM0 sub family. The nortos example is under the address $mspm0_sdk_x_x_x_x \ examples \ nortos \ LP_MSPM0xxxx$ and the RTOS example is under the address $mspm0_sdk_x_x_x_x \ examples \ RTOS \ LP_MSPM0xxxx$. This section shows a brief introduction for some key example types.

- RTOS Folder:
 - Drivers: Examples uses kernel functionality and provide higher-level hardware operation based on TI Drivers. For Driver Porting Layer (DPL), the DPL abstracts the drivers, allowing for migration between different RTOS kernels or No-RTOS. For POSIX layer, the layer abstracts RTOS functionality, allowing for migration to new kernels.
- Nortos Folder:
 - DriverLib: Simple modular examples showing MSPM0 functionality, consisting of low-level drivers with the highest optimization.
 - Middleware: Designs for different applications, with libraries and protocol stacks, including automotive, appliances, building automation, and so on. For a list of supported middleware, see MSPM0-SDK Document Overview.
 - Demos: Integrated ready-to-use demos, such as driver code examples to work with TI analog devices.

In the RTOS example level, the most important folder is the *Drivers* folder that demos the peripheral control based on TI Drivers. In the Nortos example level, the most important folder is the *DriverLib* folder, which contains the peripheral example code based on DriverLib. The address and content example are shown in Figure 3-6.

examples	>	rtos	>	LP_MSPM0G3507	>	drivers	>	
----------	---	------	---	---------------	---	---------	---	--

content example of rtos drivers folder

```
examples > nortos > LP_MSPM0G3507 > driverlib > 
content example of nortos driverlib folder
```

~ ^				
Name	Date modified	Туре	Name	Date modified
adc_singlechannel	9/24/2024 6:13 PM	File folder	gpio_simultaneous_interrupts	9/24/2024 6:06 PM
iii empty	9/24/2024 6:13 PM	File folder	gpio_software_poll	9/24/2024 6:07 PM
📁 gpio_interrupt	9/24/2024 6:13 PM	File folder	gpio_toggle_output	9/24/2024 6:08 PM
i2c_controller	9/24/2024 6:13 PM	File folder	📁 gpio_toggle_output_cpp	9/24/2024 6:07 PM
i2c_target	9/24/2024 6:13 PM	File folder	📁 gpio_toggle_output_hiz	9/24/2024 6:06 PM
i2c_tmp	9/24/2024 6:13 PM	File folder	i2c_controller_rw_multibyte_fifo_interrupts	9/24/2024 6:07 PM
spi_controller	9/24/2024 6:13 PM	File folder	i2c_controller_rw_multibyte_fifo_poll	9/24/2024 6:07 PM
spi_peripheral	9/24/2024 6:13 PM	File folder	i2c_controller_target_dynamic_switching	9/24/2024 6:07 PM
uart_callback	9/24/2024 6:13 PM	File folder	i2c_multicontroller_arbitration	9/24/2024 6:07 PM

Figure 3-6. RTOS and Nortos Code Examples

For reference, examples under *Drivers* and *DriverLib* supports all the platforms listed in Table 3-3. Examples under other folders at least support the CCS platform.

Supported by SDK	Platfo	orm 1	Platform 2	Platform 3				
IDE	CC	CS	Keil	IAR				
Compilers	TI Arm-Clang	GNU Arm (GCC)	Arm and Keil Compiler	IAR Arm compiler				
RTOS		FreeRTOS						
Code examples		DriverLib and TI Drivers						

Table 3-3. MSPM0 Example Coverage

For a MSPM0 peripheral quick start, see MSPM0 Academy. This delivers training modules for various topics in the MSP MCU portfolio.

3.2.2.2 Documents Folder Introduction

This section lists all the documents in MSPM0-SDK. This is based on version 1_20_01_06.

MSPM0 SDK Documentation:

- · Release Notes: Lists all the contents of the MSPM0-SDK and release notes.
- Quick Start Guides: Provides step-by-step instructions to get started quickly using MSPM0 with Code Composer Studio[™] (CCS) Theia, CCS, IAR or Keil.
- MSPM0 SDK User's Guide: Homepage of MSPM0-SDK. Provide navigation to MSPM0-SDK example guide and SDK overview.
- Manifest: Lists all the contents in SDK and every installation file path for each component.
- Early Samples Migration Guide: Describes the recommended tool versions that support production samples and provide migration guidelines for applications using DriverLib and SysConfig configuration files.

DriverLib Documentation:

 DriverLib Guide: Provides a software layer to the programmer to facilitate a higher level of programming compared to direct register access.

TI Drivers Documentation:

TI Drivers Overview: TI Drivers is a collective of peripheral drivers for TI's MSPM0 portfolio. The drivers
are centered around a portable application programming interface (API) which enables seamless migration
across the MSPM0-SDK portfolio. Unless specifically stated otherwise, TI Drivers are designed to be thread
safe and work seamlessly inside of a real-time operating system (RTOS) application.

Middleware Documentation (Libraries and protocol stacks for different applications):

- Middleware Main Folder
- Secure Booting and Updating
- Brushed Motor Control Library
- DALI Library



- Diagnostic Library
- EEPROM Emulation Library
- Energy Metrology Library
- GUI Composer Library
- Hall Sensored Trap Motor Control Library
- IQMath Library
- LIN Library
- Sensorless FOC Motor Control Library
- SENT Library
- SMBBus Library
- Stepper Motor Control Library
- PMBus Library

Third Party Documentation:

- CMSIS DSP: Texas Instruments supports Arm[®] Cortex[®] Microcontroller Software Interface Standard (CMSIS), a standardized hardware abstraction layer for the Cortex-M processor series.
- IO-Link: Digital interfaces such as IO-Link on the sensor and actuator level offer advantages when maintenance and repair is required in addition to providing seamless communication and improved interoperability.
- Zephyr: Texas Instruments has started development to support Zephyr as a real-time operating option for MSPM0 devices.



MSPM0 Tools Documentation:

- IDEs and Compilers: MSPM0 supports IDEs: Code Compose Studio (CCS), IAR Embedded Workbench for Arm, Arm Keil MDK. For the toolchain, MSPM0 supports both TI Arm Clang Compiler and Arm GCC Toolchain.
- Code Generation: MSPM0 supports SysConfig.

Debugging and Programmings Tools:

- XDS-110: The Texas Instruments XDS110 is a new class of debug probe (emulator) for TI embedded processors.
- MSP-GANG: The MSP Gang Programmer (MSP-GANG) is a device programmer that supports MSPM0 and all variants of MSP430 and MSP432.
- UniFlash: UniFlash is a standalone tool used to program on-chip flash memory on TI MCUs and on-board flash memory for Sitara processors. To access the quick start guide, click here.
- BSL Host: MSPM0 devices are shipped with a highly customizable ROM-based bootloader that supports universal asynchronous receiver/transmitter (UART) and inter-integrated circuit (I2C) communication by default. For more information, see the MSPM0 Bootloader (BSL) Implementation.
- MSPM0 Factory Reset GUI Tool: The Debug Subsystem Mailbox (DSSM) can be used to perform a device mass erase, perform a factory reset, and send a password to unlock the SWD interface.
- Elprotronic: Elprotronic offers multiple hardware and software programming tools supporting MSPM0 in addition to Texas Instruments' MSP430 and MSP432, SimpleLink[™] (CC), C2000[™], and TIVA[™]-C MCUs. Elprotronic supports MSPM0 include the MSP-GANG, FlashPro-ARM, and GangPro-ARM.
- Segger: SEGGER J-Link debug probes are the most widely used line of debug probes available today. For more details, see Using Segger programmers with MSPM0.
- PEmicro: PEmicro Multilink and Multilink FX debug probes offer an affordable and compact method for TI MSPM0 development, and allow debugging and programming to be accomplished simply and efficiently.
- Lauterbach: MSPM0 is supported by all Arm debug tools. Generally used for Cortex-M controllers, the preferred tool is the µTrace for Cortex-M.



3.3 SysConfig Setup

SysConfig is a collection of graphical utilities for configuring pins, peripherals, and other components. SysConfig helps manage, expose, and resolve conflicts visually so that a user has more time to create differentiated applications. The output of the tool includes the C header and code files that can be used with MSPM0-SDK examples or used to configure custom software.

3.3.1 SysConfig Installation

If a user selects CCS as the IDE platform, this section can be ignored, as SysConfig is already integrated.

If a user selects Keil or IAR as the IDE platform, download the standalone SysConfig configuration tool and follow the steps to finish the installation, as shown in Figure 3-7.



Figure 3-7. SysConfig Install



3.3.2 SysConfig Introduction

This section is a simple introduction on how to use SysConfig. Additional sections further introduce how to use SysConfig with IDE in Section 3.4.

- Add the required peripherals in *Peripheral Usage*.
- Set the parameters in Peripheral setting, paired with the device-specific technical reference manual.
- · After debugging, the peripheral can generate C code directly.



Figure 3-8. MSPM0 SysConfig

The next section introduces the components in SysConfig, which is abstracted from Using SysConfig with MSPM0.

3.3.2.1 Basic Concept

This section introduces SysConfig function blocks and basic operation.

As shown in Figure 3-9, the basic view is shown after SysConfig is opened. SysConfig has two function blocks: the peripheral usage block, which is used to show the added peripherals and the peripheral setting menu entrance. Second is the peripheral setting, which is used to configure the MCU peripherals.

After clicking the buttons on the right side of the screen, the user can open more windows. Generated files are shown after the project build. The user can click the files individually to know the changes after doing new settings on SysConfig. The MCU view is used to view the pin assignment and pin resources, which is also an entrance for MSPM0 migration.

§ *adc12_14bit_resolution.syscfg ×			and the second		§ *adc12_14bit_resolution.syscfg ×		Click Buttons	
Peripheral U	Jsage	← → Software → SYSCTL Peri	pheral Setting	() <> @ +9 :	📰 \Xi Type Filter Text	×	← → Software → SySCTL	
PROJECT CONFIGURATION (1) Project Configuration Files	1/1 🕑 🕀	SYSCTL 💿		⊕ ADD ■F REMOVE ALL	PROJECT CONFIGURATION (1) Project Configuration Files	1/1 🥑 🕣	SYSCTL ⊕ ⊕ ADD ■ REMOVE ALL	D device.op Generate Ct. Files
MSPM0 DRIVER LIBRARY (6)		Graphical Clock Configuration			MSPM0 DRIVER LIBRARY (6) SYSTEM (7)		Graphical Clock Configuration	device.cmd.genlibs Library Includes
Board	1/1 🔮 🕀	Use Clock Tree	2		Board DMA	1/1 ⊘ ⊕ 1/1 ⊘ ⊕	Use Clock Tree	ti_msp_dl_config.c MSPM0 Driver Library
GPIO Configuration NVM	1 ⊘ ⊕	Power & Systems Configuration		~	GPI0 Configuration NVM	1 🔮 🕀	Power & Systems Configuration	ti_msp_dl_config.h MSPM0 Driver Library
SYSCTL	1/1 🥑 🕀	Run Power Policy	RUNO	· ·	SYSCTL SYSTICK	1/1 🥑 💮	Run Power Policy RUNO * Low Power Policy SLEEP0 *	Event.dot MSPM0 Driver Library
WWDT	() ()	Low Power Policy Supplied Operational Voltage (VDD)	SLEEP0		WWDT ~ ANALOG (5)	۲	Supplied Operational Voltage (VD 3.3	adc12_14bit_resolution.syscfg Configuration
ANALOG (5) ADC12	1/1 🥥 🕀	BOR Threshold	0	*	ADC12 COMP	1/1 ⊘ ⊕ ⊕	BOR Threshold 0 * Enable Write Lock	7 Total Files
COMP	 (1) (2) (3) (4) (4)	Enable Write Lock			GPAMP OPA	÷	Enable Sleep On Exit	
OPA	•	Enable Sleep On Exit Enable Event On Pending			VREF	•	Enable Event On Pending	(Device) MCU View
VREF > COMMUNICATIONS (5)	Ð	Disable NRST Pin			 COMMUNICATIONS (5) TIMERS (4) 		VBOOST Operating Mode ONDEMAND *	(Package)
V TIMERS (4)	0	VB00ST Operating Mode	ONDEMAND		TIMER - CAPTURE TIMER - COMPARE	⊕ ⊕	Flash Controller (FlashCtl) Configuration	NECOSON NANA
TIMER - COMPARE	•	Flash Controller (FlashCtl) Configuration		0	TIMER - PWM TIMER	•	Enable FlashCtl Interrupt	1 24
TIMER - PWM TIMER	 ⊕ ⊕ 	Enable FlashCtl Interrupt			V DATA INTEGRITY (1)	•	Non-Maskable Interrupt (NMI) Configuration	
V DATA INTEGRITY (1)		Non-Maskable Interrupt (NMI) Configuration		<u>^</u>	CRC ~ READ-ONLY (1)	۲		
V READ-ONLY (1)	•	WWDT0 Error Behavior	BOOTRST	-	EVENT	1/1 🥑 🕀	WWDT0 Error Behavior BOOTRST +	0.0=200428
EVENT	1/1 🛃 (+)	Basic	View				Advanced Vie	eW W

Figure 3-9. SysConfig View



The basic operations of SysConfig, includes adding peripherals, removing peripherals and referring the peripheral or function descriptions. As SysConfig is a low level MSPM0 peripheral setting GUI, see the technical reference manual or the peripheral examples to obtain a better understanding.



Figure 3-10. Basic Operations

3.3.2.2 Project Configuration View

Here is the project configuration. The configuration influences the total MCU project setting. This section is an introduction to some important features.

- *File Generation*: After you enable all the selection box, the project related files are auto generated by SysConfig. We suggest you to keep them under selection.
- *Include Libraries*: This shows all the libraries included in the SDK. After the selection box is enabled, the related library is included into the project automatically.
- Select Device: As the SDK examples is for the LP, after the MCU is migrated, this setting can be changed.



Figure 3-11. Project Configuration



3.3.2.3 Board View

Board view is used to configure the total MCU configuration.

- *Debug Configuration*: For some MSPM0s, the configuration reuses the debug port as peripheral functions. This is the SWD disabled entrance.
- Global Pin Configuration:
 - Enable Global Fast-Wake: This reduces the wake-up time sourced from any GPIO port.
 - Generate Peripherals and Pin Assignments File: After enabling, a peripherals and pin assignments file is generated in the *Debug* folder, as shown in Figure 3-12.



Figure 3-12. Board View

3.3.2.4 NONMAIN View

The NVM (NONMAIN) is used to configure the MSPM0 protected area related to boot configuration, security, and bootloader. With the incorrect program in NONMAIN, MSPM0 breaks. That is why the configuration risks must be accepted before performing configurations. As this function is for high level users, for details, please refer to MSPM0 NONMAIN FLASH Operation Guide.

l (1) les 1/1 🥑	Configuration NVM (2)	⊕ ADD 📑	REMOVE ALL
6)	Using the Configuration NVM (NONMA	AIN) has certain risks. Please refer to the configuration risks fie	elds's long
	description by clicking the question m	ark next to the field for more details on the risks	
1/1 🥑	÷	Click	
1/1 🥥	Quick Profiles	Click	^
1 📀	\oplus		
1/1 😣	Accept configuration risks	💭 User must accept Configuration NVM (N	IONMAIN)
1/1 🥑	Ð	configuration risks	
	Debug Security Profiles	Security Level 0 - No restrictions	*
	Ð		
	Post Configuration Posting (PCP) C	- finnetion	~
1/1 🥑	Boot configuration Routine (BCR) C	onnguration	•
	\oplus		
	Bootstrap Loader (BSL) Configuration	n	~
	((1) les 1/1 ♥ 6) 1/1 ♥ 1/1 ♥ 1/1 ♥ 1/1 ♥	(1) les 1/1 ◆ ⊕ (6) ● ● 1/1 ◆ ⊕ ● ● 1/1 ◆ ⊕ ● ● 1/1 ◆ ⊕ ● ● 1/1 ◆ ⊕ ● ● 1/1 ◆ ⊕ ● ● 1/1 ◆ ⊕ ● ● 1/1 ◆ ⊕ ● ● 1/1 ◆ ⊕ ● ● 1/1 ◆ ⊕ ● ● 1/1 ◆ ⊕ ● ● ● ● ● ● ● ● ● ● ● ● ● ●	(1) (1)

Figure 3-13. NONMAIN View



3.3.2.5 SYSCTL View

SYSCTL is used to configure MCU power, clock, and reset modules. The basic view is menus view. This section introduces the main configurations.

- Power and Systems Configuration:
 - Low power policy: Sets the low-power level for MSPM0.
 - Disable NRST pin: For some MSPM0 devices, the NRST pin can be reused as peripheral functions. This
 is the NRST pin disabled entrance.

The second view is clock tree view. The clock tree feature allows the user to configure the clocking of a device graphically rather than using SYSCTL menus, which can be found by clicking the signal icon near the top left corner of SysConfig. At the bottom left of the clock tree view, a user can locate all of the used clocks. For a detailed configuration on every clock source, click the icons as shown in Figure 3-14.



Figure 3-14. SYSCTL View



3.3.2.6 Peripherals Setup

This section introduces peripheral settings, as shown in Figure 3-15. Open the software module description by selecting the module before adding the description. The description includes an overview of the functionality of the module. For more information, see the device data sheet or technical reference manual.

A peripheral configuration is a combination of these configurations:

- Basic configuration: Basic peripheral configuration ٠
- Advanced configuration: Advanced peripheral configuration •
- Interrupts configuration: Enable or disable MCU interrupt
- Event configuration: Peripheral to peripheral trigger configuration •
- Pin configuration: Enables pullup or pulldown resistors
- PinMux: Selects the pin input or output for the selected functions ٠

82	Ŧ	Type Filter Text	×	\leftarrow \rightarrow Software $ ightarrow$ TIMER - CAPTURE		() <> @ 49 🗄
999 999	~ P	ROJECT CONFIGURATION (1) Project Configuration Files	1/1 🕑 🕀	TIMER - CAPTURE (1 of 4 Added) ⑦		⊕ ADD ■ ■ ■ ■ ■ ■ ■ ■ ■
	~ M	ISPM0 DRIVER LIBRARY (6) SYSTEM (7)		CAPTURE_0		
		Board	1/1 🕑 🕀	Name	CAPTURE_0	
		DMA	1/1 🕑 🕀	Selected Peripheral	TIMG0	
		GPIO	1 🕜 🕀			
		SYSCTL	1/1 🔿 🕀	Quick Profiles		Ŷ
		SYSTICK	(†) (†)	Basic Configuration		~
	~	ANALOG (5)		Desired Timer Period	<u>0 ms</u>	
		ADC12	1/1 🥑 🕀	Actual Timer Period	0.00 s	
		COMP	\oplus	Start Timer		
		OPA VREF	(+) (+)	Capture Configuration		~
	> ~	COMMUNICATIONS (5) TIMERS (4)	U U	Advanced Configuration		~
		TIMER - CAPTURE TIMER - COMPARE	1/4 🥑 🕂 🕂	Interrupts Configuration		~
		TIMER	÷	Event Configuration		~
	~	DATA INTEGRITY (1)				
	~	CRC PEAD-ONLY (1)	\oplus	Pin Configuration		~
	Ť	EVENT	1/1 🥑 🕀	PinMux Peripheral and Pin Configurat	ion	~

Figure 3-15. Peripherals View



3.4 IDE Quick Start

The MSPM0 series supports three IDEs to develop. CCS is recommended as a preferred option, as this is TI's IDE, which is compatible with MSPM0. The three different types of IDEs are listed and compared in Table 3-4.

IDEs	CCS	IAR	Keil				
License	Free	Paid	Paid				
Compiler	TI Arm Clang GCC	IAR C/C++ Compiler™ for Arm	Arm Compiler Version 6				
Disk size	4.60G(ccs2001)	6.33G(Arm 8.50.4)	2.5G (µVision V5.37.0)				
XDS110	Supported	Supported	Supported				
J-Link	Supported	Supported	Supported				
EnergyTrace	Supported	No	No				
MISRA-C	No	Supported	No				
Security	No	Supported	No				
ULINKplus	No	No	Supported				
Function safety	No	Supported	Supported				

Table 3-4. MSPM0 Supported IDEs Overview

The following links provide the related guides for different IDEs. All the content in this part is abstracted from these guides, especially for new beginners.

- MSPM0 SDK Quick Start Guide for CCS
- MSPM0 SDK Quick Start Guide for IAR
- MSPM0 SDK Quick Start Guide for Keil
- CCS IDE Guide for MPSM0
- IAR IDE Guide for MSPM0
- Keil IDE Guide for MSPM0



3.4.1 CCS Quick Start

Note that in 2024, CCS changed platforms from CCS 12.x to CCS 20.x. CCS 20.x, built on the Theia IDE framework, is newer, with a modern interface and better features than CCS 12.x (based on Eclipse). Here are the selection suggestions:

- When to choose CCS 20.x: if users are starting a new project with the latest TI devices and want the most up-to-date features, or prefer a more modern and user-friendly IDE experience.
- When to choose CCS 12.x: if users have existing projects developed in CCS 12.x and want to avoid major code changes during migration.

This section details an introduction based on CCS 20.x. If users want to use CCS 12.x, then here is the latest version entrance: CCS12.8.1. For documentation, refer to MSPM0 SDK Quick Start Guide for CCS v12 (Eclipse) and CCS V12 Eclipse IDE for MSPM0.

3.4.1.1 CCS Installation

This section details steps and tips for CCS installation. Remember to save CCS at the address and the default installation place that is suggested.

- 1. Download CCS (20.0 version or above), start installation, and keep pressing Next.
- 2. Select I accept the agreement.
- 3. Suggest to keep the install address to be default.
- 4. Select MSPM0 support component.







3.4.1.2 Import a SDK Example

Import an example with the TI-Clang compiler from the installed SDK. Here are the steps:

- 1. Select File \rightarrow Import Projects.
- 2. Browse to the SDK installed address.
- 3. Select the code example folder.
- 4. Select the code example with the wanted complier. The ticlang complier is suggested.

	File	Edit Selection View	Go Project Run	Scr	Import Projects	Step2	×
		Create New Project			Search directory	please specify a directory to search	Browse
)	New Text File New File New Folder Add Files/Folders	Ctrl+N Ctrl+Alt+N	1	Discovered projec	ts	
iory	PM0G3507 → driverlib → a	adc12_14bit_resolution	C Search adc12_14bit_reso	olution 🖌	× Import Projects	Step4	×
folder				• (Search director	y C:\ti\mspm0_sdk_2_02_00_05\examples\nortos\LP_MSPM0G350	Browse
12_01# Ni	ame Agent Ag	Date modified 9/24/2024 6:08 PM 9/24/2024 6:08 PM 9/24/2024 6:08 PM 11/25/2024 5:48 PM	Type Size File folder File folder File folder File folder		Discovered pro	njects it_resolution_LP_MSPM0G3507_nortos_gcc {gcc\adc12_14bit_resolu it_resolution_LP_MSPM0G3507_nortos_ticlang {ticlang\adc12_14bit	ıtion_LP_M _resolution
ılder.			Select Folder	Cancel	create each p	roiect in: workspace ccstheia	
						Open 'Resource Explorer' to browse and import e	example projects
						Cance	el Finish

Figure 3-17. Import CCS Project

Here is the view of the imported project. The most important files are in red. This section shows a brief introduction.

- Sysconfig generated code: Click the *Build* button, the SysConfig generates the code under the *Debug*\syscfg folder or directly under the *Debug* folder.
- .map file: In the *Debug* folder, refer to the .map file to find out more about the memory usage condition.
- Main function .c file: Includes the main function in the file.
- .cmd file: Define the MCU memory allocation. In the latest CCS, SysConfig generates the allocation file automatically with the setting in *Project Configuration Files*.
- SysConfig: GUI tool to generate the peripheral setting code.



~

É	gpio_toggle_output_LP_MSPM	M0L1306_nortos_ticlang
>	🗊 Generated Source	
>	🔊 Includes	
~	🗁 Debug	Sysconfig generated code
	🗙 🗁 syscfg	
	> i_msp_dl_config.c	
	h ti_msp_dl_config.h	Memory allocation
	gpio_toggle_output_LP	_MSPM0L1306_nortos_ticlang.map
>	🗁 targetConfigs	
>	귿 ticlang	
>	gpio_toggle_output.c	Main function
>	🌛 mspm0l1306.cmd	CCS memory map
	Signio_toggle_output.syscfg	y Sysconfig

Figure 3-18. CCS Project Overview

3.4.1.3 Example Download and Debug

The default debugger selection is XDS110. If users are using the Launchpad, then keep default. To select J-Link, right-click the Project name, *Project -> Properties -> General* and follow the instruction as bellow to select J-Link.

Properties for: adc12_14bit_resolution_L	P_MSPM0G3507_norto	s_ticl	ang [Debug]	×
 ✓ General Dependencies ↓ Variables ✓ Build i E Steps ⊕ Link Order ✓ Tools > Ø SysConfig > Ø Arm Compiler > Ø Arm Compiler > Ø Arm Objcopy Utility [dis Executable Actions Clang-Tidy Debug 	Device family Device variant / core Connection Output type Compiler	(i) (i)	ARM MSPM0G3507 CORTE) Texas Instruments XDS110 USB Debug Probe [default] Blackhawk XDS560v2-LAN System Trace Emulator Blackhawk XDS560v2-USB Mezzanine Emulator Blackhawk XDS560v2-USB System Trace Emulator Data Snapshot Viewer SEGGER-L-Link Emulator Spectrum Digital XDS560v2 STM LAN Emulator Spectrum Digital XDS560v2 STM UAN Emulator Spectrum Digital XDS560v2 STM UAN Emulator Spectrum Digital XDS700v2 STM USB Emulator Spectrum Digital XDS700v2 USB Emulator Texas Instruments XDS100v2 USB Debug Probe Texas Instruments XDS110 USB Debug Probe Texas Instruments XDS2xx LAN Debug Probe Texas Instruments XDS2xx USB Debug Probe Texas Instruments XDS2xx USB Debug Probe	
			Texas Instruments XDS2xx USB Onboard Debug Probe UARTConnection Cancer Save and Cit	лъе

Figure 3-19. Change Debugger Selection

After the project is imported and the hardware is set up, users can follow the setup to run the code on MCU.

- 1. Start debug by right-clicking the project name. First click *Build Projects* and then click *Debug Project*. Users can also use Ctrl+B and F5 instead.
- 2. After that, the window automatically moves from the CCS edit view to CCS debug view. After the MCU enters debug mode, click the *Run* button to enable the code running.



✓ E adc12_1 > E Gener	New File Step)1		Step2		
 C Refere C Debug T target C adc12 	New Folder Add Files/Folders Reveal in File Explorer Open in Integrated Terminal	Ctrl+Alt+P	۹ ب	DEBUG adc12_14bit_resolution_LP_MSPM0G350 ✓ ② ▷ En ▷ ♂ ♀ ↑ ♡ □		
READI	Select for Compare	Alt+Shift+F		> THREADS > CALL STACK		
	Copy	Ctrl+C	de la	> VARIABLES > WATCH		
	Copy Path Copy Relative Path	Alt+Shift+C Ctrl+K. Ctrl+Shift+C	B	BREAKPOINTS TARGET CONFIGURATION		
I	Build Project(s) Clean Project(s)	Ctrl+B				
	Rebuild Project(s)	X				
	System Configurations Executable Actions	>				
i i i	Province protoco					



This section is a quick introduction to CCS functions. The commonly used functions and meanings are shown in Figure 3-21. To view registers, users need to click *View -> Registers* and to open a register view window.

Vie	w Go Project Run	Scripts Terminal H	Help	DEBUG
	Command Palette	Ctrl+Shift+P		
	Open View	3	3. Debug d	adc12_14bit_resolution_LP_MSPM0G3507 > 🛱 🖂 🗔
	Appearance		>	
	Editor Layout		> >	THREADS
0	Resource Explorer		>	CALL STACK
	Console	3	> >	VARIABLES
Ċ	Connected Targets			BREAKPOINTS and breakpoints
	Debug	Ctrl+Shift+D	>	TARGET CONFIGURATION
	Disassembly			
	Explorer	Ctrl+Shift+E		
C	GEL Files			
~	Graph			
02	Memory			
F	Memory Allocation			
0	Memory Map			
000	Modules			
	Outline	Ctrl+Shift+I		
	Output	Ctrl+Shift+U		
	Problems	Ctrl+Shift+M		
	Properties	Alt+Shift+P		
1010 0011	Registers 1. Viev	v registers		





3.4.1.4 Migrating Between MSPM0 Derivatives

Project migration in this scope means updating relevant project configuration files and settings that are specific to the derivative, including linker files, startup files, and included libraries. To facilitate project migration, SysConfig generates project configuration files by default, which can be controlled through the project configuration module.

Here are the migration steps based on CCS:

- 1. In SysConfig, enable the device view and click on SWITCH.
- 2. Select *New Values* for the *Device*, *Package*, and *CCS Launch Device* to migrate the project configuration to a new device, and then click *CONFIRM*.
- 3. After confirming the new device values, SysConfig highlights an error on the project configuration module. The user must select the new device in the *Select Device* options. Make sure the device selection matches what was selected for *CCS Launch Device* in the previous step.
- 4. Note that SysConfig highlights any conflicts with the migration, such as unavailable pins and peripherals. Fix any conflicts as needed, and save all the changes to the SysConfig configuration script. Migration is now complete and the user can build a project for the new target device.



Figure 3-22. Migrating Between MSPM0 Derivatives

3.4.1.5 Generate Hex Files

CCS includes utilities which can be used to generate output objects in multiple formats for use with programming tools. The following steps explain how to enable the hex files using the hex utility which is integrated into CCS.

- 1. Right-click on a project and select *Properties*. Select *Build* → Tools → *Arm Hex Utility* and select *Enable Arm Hex Utility*.
- 2. Select *Output Format Options*. The common selections are *Bin*, *Hex*, and TI_TXT format. Select the desired output format options.
- 3. If the Intel *HEX* format is selected, **one additional step** is required to specify the memory and ROM width as parameters. Select a memory and ROM width of 8in *Properties* → *Arm Hex Utility* → *General Options*.
- 4. Right-click on the project and select *Build Projects*. The hex file generates in the debug folder.



perties for: adc12_14bit_resolution	h_LP_MSPM0G2001_nortos_ttalang [Debug]	× Propert	ties for: adc12 14bit resolution	n LP MSPM0G3507 no	rtos ticlang IDebugi
Grand	Step1	~ виг	iu		Step2
General III Dependencies	Command "\${CG_TOOL_HEX}"	E	E Steps	Output format	
Variables		R	b Link Order		
Build	Arm Hex Utility flags	0 × %	ools Tools		Array (array) ASCII hex (ascii, -a)
i≡ Steps		> <	A SysConfig	Rin	Binary (binary, -b)
🕲 Link Order	memwidth=8romwidth=8diag_wrap=offintel	> 0	Arm Compiler		Motorola S hex (motorola, -m=1)
∕ ⅔ Tools		> 0	Arm Linker	Hex	Motorola S hex (-motorola, -m=2)
> 🌽 SysConfig			General Options		Extended Tektronix hex (tektronix, -x)
> / ^p Arm Compiler			Diagnostics Options		TI-Tagged hex (ti_tagged, -t)
> // Arm Linker	a		Boot Table Options	TI_TXT	
General Ontions	e		Output Format Options		
Diagnostics Options			Load Image Options		
Boot Table Options			Additional Array Format O		
Output Format Options	Enable 'Arm Hex Utility' Remove 'Arm Hex Utility' from	this tool-chain	Arm Obicopy Utility Idis		
		,			
perties for: adc12_14bit_resolutio	LP_MSPM0GS	,	× vworks		an/ េដ្ដេសគ
perties for: adc12_14bit_resolutio ชนเต	1_LP_MSPM0GSSCTOP3ang [Debug]	,,	× ∨ Works ∨ ⊟ _c ad	PACE_CCSTHEIA	다. で ひ の LingPhMog3507_nortos_ticlang [Debug]
<mark>perties for: adc12_14bit_resolutio</mark> випа ¡⊟ Steps	ո_LP_MSPM0GS <mark>ST_ODB</mark> ang [Debug]		× ∨ WORKS	PACE_CCSTHEIA St. lc12_14bit_resolution_L	다. 다 다 신 의 Ch 다 신 의 Ch SPM0G3507_nortos_ticlang [Debug]
perties for: adc12_14bit_resolutio випа I E Steps © Link Order	LP_MSPM0GSSterpBlang [Debug]		× • WORKS	PACE_CCSTHEIA Store	다. 다 아 의 다. PPMOG3507_nortos_ticlang (Debug)
perties for: adc12_14bit_resolutio вила E Steps © Link Order > Tools	Specify fill value (fill, -fill)		× ∨ ₩0RKS	PACE_CCSTHEIA SEC Ic12_14bit_resolution_L Senerated Source Referenced Source Debug add 12_14bit_resolution	다. MSRM0G3507_nortos_ticlang [Debug]
perties for: adc12_14bit_resolutio Buina	1.1P. MSPM06:S0[0:0] Specify fill value (fill, -fill) Specify rill value (fill, -fill)		× • works - ↓ ⊕ . - ↓ ⊕ . • ↓ . • ↓ . • ↓ ⊕ . • ↓ . • ↓ ⊕ . • ↓ ⊕ . • ↓ ⊕	PACE_CCSTHEIA Clc12_14bit_resolution_L Generated Source Referenced Source Debug adc12_14bit_resolution adc12_14bit_resolution	다.P.MSPM0G3507_nortos_ticlang_linkInfo n_LP_MSPM0G3507_nortos_ticlang_linkInfo
perties for: adc12_14bit_resolutio Bund I E Steps [©] Link Order × [™] Tools > [®] SysConfig > [®] Arm Compiler > [®] Arm Cinker	n. LP_MSPM0GSS0 Context Specify fill value (fill, -fill) Specify fill value (fill, -fill) Select image mode (image, -image) Include linker fill sections in images (linkerfill, -linkerfill)		× ∨ WORKS → C, a → C, → C,	PACE_CCSTHEIA ict2_14bit_resolution isenerated Source Referenced Source Pebug adc12_14bit_resolution adc12_14bit_resolution adc12_14bit_resolution	L+ C+ C G HSPM0G3507_nortos_ticlang [Debug] h_LP_MSPM0G3507_nortos_ticlang_linkInfo. n_LP_MSPM0G3507_nortos_ticlang.hex n_LP_MSPM0G3507_nortos_ticlang.map
perties for: adc12_14bit_resolution Bund I Steps [©] Link Order × [™] Tools > [®] SysConfig > [®] Arm Compiler > [®] Arm Hex Utility	n. LP. MSPM0GS Context Specify fill value (fill, -fill) Specify fill value (fill, -fill) Select image mode (image, -image) Include linker fill sections in images (linkerfill, -linkerfill) Specify map file name (map, -map)		× • ₩00RK5 ~ □ ad > 2 ad > 2 ad •	PACE_CCSTHEIA ST icht2_14bit_resolution_L icht2_14bit_resolution adc12_14bit_resolution adc12_14bit_resolution adc12_14bit_resolution	LL CL C
perties for: adc12_14bit_resolutio Buna I Steps [©] Link Order [×] ∯ Tools ^{>} ℬ SysConfig ^{>} ℬ Arm Compiler ^{>} ℬ Arm Ker [×] ℬ Arm Hex Utility General Options	n.1P.MSPM0GS Amount Amount Specify fill value (fill, -fill) Specify fill value (fill, -fill) Include linker fill sections in images (linkerfill, -linkerfill) Specify map file name (map, -map) Specify memory width (memwidth, -memwidth)	8	× ∨ WORKS ~ 2 d . 3 d . 2 d . 3 d . 3 d . 4	PACE_CCSTHEIA State icnt2_14bit_resolution_L icnt2_14bit_resolution adc12_14bit_resolution adc12_14bit_resolution adc12_14bit_resolution	C+ C+ C Ø PMVG3507_nortos_ticlang [Debug] h_LP_MSPM0G3507_nortos_ticlang_linkInfo. n_LP_MSPM0G3507_nortos_ticlang.map h_LP_MSPM0G3507_nortos_ticlang.map
perties for: adc12_14bit_resolutio Buna I Steps [©] Link Order [×] ∑ Tools ^{>} Ď SysConfig ^{>} Ď Arm Compiler ^{>} Ď Arm Compiler ^{>} Ď Arm Mex Utility <u>General Options</u> Diagnostics Options Diagnostics Options	LP_MSPM0GS Ang [Debug] Specify fill value (fill, -fill) Select image mode (image, -image) Include linker fill sections in images (linkerfill, -linkerfill) Specify map file name (map, -map) Specify memory width (memwidth, -memwidth) Specify output file names (outfile, -o)	8	× ₩00RK5 × □,ad > □, 2, 4 • □, 4 • □, 1 • □, 1	PACE_CCSTHEIA Senerated Source Verterenced Source Pobug adc12_14bit_resolution adc12_14bit_resolution adc12_14bit_resolution	LP_MSPM0G3507_nortos_ticlang_linkInfo. n_LP_MSPM0G3507_nortos_ticlang_linkInfo. n_LP_MSPM0G3507_nortos_ticlang.hex n_LP_MSPM0G3507_nortos_ticlang.map n_LP_MSPM0G3507_nortos_ticlang.out
perties for. adc12_14bit_resolutio Buno E Steps C Link Order > D SysConfig > D Arm Compiler > D Arm Compiler > D Arm Linker > D Arm Hex Utility General Options Diagnostics Options Boot Table Options Boot Table Options Output Format Options	LLP. MSPMOGS and the second s	8	× works:	PACE_CCSTHEIA Senerated Source Senerated Source Debug adc12_14bit_resolution adc12_14bit_resolution adc12_14bit_resolution	LP_MSPM0G3507_nortos_ticlang_linkInfo. n_LP_MSPM0G3507_nortos_ticlang_linkInfo. n_LP_MSPM0G3507_nortos_ticlang.hex n_LP_MSPM0G3507_nortos_ticlang.map n_LP_MSPM0G3507_nortos_ticlang.out
perties for. adc12_14bit_resolutio Buno E Steps C Link Order > ℬ SysConfig > ℬ Arm Compiler > ℬ Arm Linker > ℬ Arm Hex Utility General Options Diagnostics Options Boot Table Options Output Format Options Load Image Options	LLP. MSPMOGS and the second s	8	× works:	PACE_CCSTHEIA Senerated Source Debug adc12_14bit_resolution adc12_14bit_resolution adc12_14bit_resolution adc12_14bit_resolution	다. MSPM0G3507_nortos_ticlang_linkInfo n_LP_MSPM0G3507_nortos_ticlang_linkInfo n_LP_MSPM0G3507_nortos_ticlang.map n_LP_MSPM0G3507_nortos_ticlang.map
perties for: adc12_14bit_resolution Build I Steps [©] Link Order [≫] Tools [≫] Ø SysConfig [≫] Ø Arm Compiler [≫] Ø Arm Compiler [≫] Ø Arm Hex Utility General Options Diagnostics Options Boot Table Options Output Format Options Load Image Options Additional Array Format O [®] Arm Ohicory Utility (disc)	L.P. MSPM0GS Cancel Specify fill value (fill, -fill) Specify fill value (fill, -fill) Include linker fill sections in images (linkerfill, -linkerfill) Specify map file name (map, -map) Specify memory width (memwidth, -memwidth) Specify output file names (outfile, -o) Quiet Operation (quiet, -quiet, -q) Specify rom width (romwidth, -romwidth) Zero based addressing (zero, -zero, -z)	8	× • works	PACE_CCSTHEIA Senerated Source Verterenced Source V	LP_MSPM0G3507_nortos_ticlang_linkInfo. n_LP_MSPM0G3507_nortos_ticlang_linkInfo. n_LP_MSPM0G3507_nortos_ticlang.hex n_LP_MSPM0G3507_nortos_ticlang.map n_LP_MSPM0G3507_nortos_ticlang.out
perties for: adc12_14bit_resolutio Buno E Steps C Link Order > D SysConfig > D Arm Compiler > D Arm Compiler > D Arm Compiler > D Arm Lex Utility General Options Diagnostics Options Boot Table Options Dot Table Options Load Image Options Additional Array Format O D Arm Objcopy Utility [dis	LP. MSPMOGS and the second secon	8	× works → 2, a → 2, → 2,	PACE_CCSTHEIA Senerated Source Debug adc12_14bit_resolution adc12_14bit_resolution adc12_14bit_resolution adc12_14bit_resolution	LP_MSPM0G3507_nortos_ticlang_linkInfo. n_LP_MSPM0G3507_nortos_ticlang_linkInfo. n_LP_MSPM0G3507_nortos_ticlang.hex n_LP_MSPM0G3507_nortos_ticlang.map n_LP_MSPM0G3507_nortos_ticlang.out

Figure 3-23. Generate Hex File

3.4.1.6 Program NONMAIN

If changes are made on the bootloader or MCU security settings by configuring the NONMAIN as shown in Section 3.3.2.4, enable the NONMAIN erase in the CCS setting as well, as shown in Figure 3-24. Otherwise, keep the default settings.

Properties for: adc12_14bit_resolution_LF	_MSPM0G3	507_nortos_ticlang [Debug] X
 ✓ General M Dependencies () Variables ✓ Build i E Steps ② Link Order ✓ % Tools > Ø SysConfig > Ø Arm Compiler > Ø Arm Compiler > Ø Arm Linker > Ø Arm Objcopy utility [dis Executable Actions Clang-Tidy Debug 	Core Category Erase Co IIIWard Erase	Texas Instruments XDS110 USB Debug Probe/CORTEX_MOP MSPMO Flash Settings Program/Memory Load Options Debugger Options MSPMO Flash Settings g: Moonying NorNVAIN Incorrectly, or erasing it without program MO documentation for more details method se MAIN memory only se MAIN and DATA memory se MAIN and NONMAIN memory (see warning above) se MAIN and NONMAIN memory (see warning above) se MAIN and NONMAIN memory (see warning above) se MAIN and NONMAIN necessary sectors only (see warning above) se MAIN memory sectors by range (specify below) not erase Flash memory rase: all 1kB sectors between Start and End address will be erased
		Cancel Save and Close

Figure 3-24. Programming NONMAIN



Note

Extreme care must be taken when erasing and programming NONMAIN. If done incorrectly, like losing connection in NONMAIN programming, the device is locked in a permanently unrecoverable state.

3.4.2 IAR Quick Start

TI recommends an IAR Embedded Workbench version higher than Arm 9.32.x. The less recent versions do not support MSPM0.

3.4.2.1 Environment Setup

3.4.2.1.1 SDK Support Setup

In IAR, users must add the latest MSPM0 SDK version. This step only has to be done once, or when the SDK is updated. This step only has to be done once, or when the SDK is updated.

- Step 1: In IAR, click on Tools → Configure Custom Argument Variables.
- Step 2: Click the Global tab, and then Import.
- Step 3: Navigate to your SDK folder into
 MSPM0_SDK_INSTALL_DIR>/tools/iar/ and open MSPM0_SDK.custom_argvars.
- Step 4: The SDK variables is now installed in IAR. Click OK to close the window.

IAR Embedded Workbench IDE Step1	Configure Custom Argument Variables Step2 ×
Eile Edit View Project Tools Window Help	Wedness Clobal 4
🛅 🗋 🔛 🚔 🙏 🍄 Options	Workspace Global 🔶
Workspace Eilename Extensions	Disable Group
Configure Viewers	New Group
Files Configure Custom Argument Variables	Ten droup
Configure Tools	Add <u>V</u> anabie
IAR Project Converter	Edit Variable
	Delete
	Import
	Evnand / Glance All
	Ligiae associed groups
	Treat all open projects as modified.
	(updated variables will be used for relative paths when saving.)
Stop2	Ston/
Jieps	Configure Custom Argument Variables JLCP4 ×
← → ~ ↑ → This PC > Windows (C) > ti > mspm0_sdk_1_00_00all > mspm0_sdk_1_00_00_00 > tools > iar v & > See	rchiar Workspace Global
Organize 🔻 New folder	
Name Date modified Type Size MSPM0_SDK.custom_argues 2/14/2023 5/28 PM CLUSTOM_ARGUAR 1 KB	MSPM0_SDX Disable Group SYSCONFIG_ROOT = c:/ti/sysconfig_1.15.0 MSPM0_SDK_INSTALL_DIE = c:/ti/msm0_sdk_1_00_00_00
	MSP_SDK_INSTALL_DIR = c:/ti/mspm0_sdk_1_00_00_00
	EXAMPLE_ROOT = c:/ti/mspm0_sdk_1_00_00_00/tools/iar Add Variable
	Edit Variable
11	Delete
	Import
	Expand/Collapse All
	C Tine grane Alacha
Cite name: MCRMO CPV autom secure	Treat all open projects as modified.
Custom	n Cannel
	UN Caricei



Note

Make sure the MSPM0 SDK path and SysConfig path matches the location and version needed for this SDK release. If an earlier version of the SDK is installed, then make sure to update the path to the current version. If the SysConfig path installed is incorrect or pointing to an older version, then modify the version.



3.4.2.1.2 SysConfig Support Setup

The SDK includes a preliminary version of SysConfig metadata which can be used to evaluate the user experience of MSPM0 SDK.

- 1. In IAR, select *Tools* \rightarrow *Configure Viewers* from the menu.
- 2. Click Import.
- 3. Navigate to your SDK folder into <*MSPM0_SDK_INSTALL_DIR*>/tools/iar/ and open sysconfig_iar_setup.xml.
- 4. The standalone SysConfig is associated to .syscfg files. Click OK to close window.
- 5. Double-check that the SYSCONFIG_ROOT Custom Argument Variable is correctly pointing to the SysConfig folder.



Figure 3-26. Install SysConfig for MSPM0

3.4.2.2 Import a SDK Example

Here are the steps to import an IAR code example from SDK:

- 1. In IAR, select *File* \rightarrow *Open Workspace* from the menu.
- 2. Navigate to an IAR folder in SDK example at <MSPM0_SDK_INSTALL_DIR>/examples/ and open the .eww workspace file.
- 3. Click OK on the message.
- 4. Select a folder to install the example.

Embedded Workbench IDE Step1	Open World	kspace			Ste	ep2					
ew File Ctrl+N	$\leftrightarrow \rightarrow \star$	↑ 🦲 « mspm0_sdk_1_00_00_00 → exam	nples > nortos > LP_MS	PM0L1306 > driverlib >	gpio_toggle_out	put > iar	~	ō	,0 si	arch iar	
ew Workspace	0	Newfolder								Rec.	
pen File Ctrl+O	Organize •	New folder								811	
Ipen Workspace		lame	Date modified	Туре	Size						
ipen Header/Source File Ctrl+Shift+H	•	gpio_toggle_output_LP_MSPM0L1306_nor	2/16/2023 5:28 PM	IAR IDE Workspace	1 KB						
ose Ctrl+F4											
ve Workspace	-										
e Wor <u>k</u> space As											
sg Workspace											
re Ctrl+S	l I										
	-										
	5										
	-										
	🥏 🗸										
	*	File name: gnio toggle output IP	MSDM0L1306 portor jarte	molate euro				×	Works	nace Files (* ew	wł
	*	File name: qpio_togqle_output_LP_	MSPM0L1306_nortos_iar.te	mplate.eww				×	Works	oace Files (*.ew	w)
	3 -	File name: gpio_toggle_output_LP_	MSPM0L1306_nortos_iar.te	mplate.eww				÷	Worksp	oace Files (*.ew oen	w) Cancel
	* *	File name: qpio_toqqle_output_IP,	MSPM0L1306_nortos_iar.te	mplate.eww				Ŷ	Worksp	oace Files (*.ew	w) Cancel
Step3	* *	File name: qpio_toggle_output_UP	MSPM0L1306_nortos_iar.te	mplate.eww	C+			~	Worksp	oace Files (*.ew	w) Cancel
Step3	*	File name: qpio_togqle_output_LP	MSPM0L1306_nortos_iar.te	mplate.eww	St	ep4		~	Worksp	oace Files (*.ew	w) Cancel
Step3	•	File name gpio_toggle_output_UP	MSPM0L1306_nortos_iar.te As Windows (C:) > ti >	mplate.eww ∙ iar → my_first_msp	St m0_project	ep4	ٽ بر س		Worksp Op Searc	bace Files (*.ew	w) Cancel
Step3	×	File name: gpio_toggle_output_LP.	MSPM0L1306_nortos_iarte As Windows (C:) > ti >	mplate.eww • iar → my_first_msp	St. m0_project	ep4	ٽ ~	•]]]	Works 0 0 Searc	bace Files (*.ew ben	w) Cancel
Step3	۲ م طer.	File name: gpio.toqqle.ovtput.IP. Save Workspace From Template ← → ← ↑ ↓ This PC → Organize ▼ New folder	MSPM0L1306_nortos_iarte	mplate.eww iar > my_first_msp	St m0_project	ep4	✓ Č	~	Worksp Op Searc	h my_first_m	w) Cancel nspm0_pro
Step3	× der.	File name: gpio.toggle_output_UP. Save Workspace From Template ← ⇒ ~ ↑ → This PC → Organize ▼ New folder ↓ ^ Name	MSPM0L1306_nortos_iarte As: Windows (C:) > ti >	nplate.eww iar > my_first_msp Date modified	St m0_project Type	ep4	v ð Size		Worksp Op Searc	bace Files (".ew ben	w) Cancel nspm0_pro ∃ ▼ (
n a workspace from a template and save it in a new fold	× der.	File name: gpio.toqqle.output.P. Save Workspace From Template ← ⇒ ~ ↑ → This PC → Organize ~ New folder ↓ ^ Name	MSPM0L1306_nortos_ierte As Windows (C:) > ti >	mplate.eww iar > my_first_msp Date modified	St m0_project Type	ep4	マ ひ Size		Worksp Op Searc	bace Files (*.ew ben	w) Cancel nspm0_pro E ▼ (
Step3	der.	File name: gpio.toggle.output.D. Save Workspace From Template ← ⇒ × ↑ → This PC → Organize ▼ New folder ↓ ^ Name	MSPM0L1306_nortos_iarte	mplatoeww iar > my_first_msp Date modified No iter	St m0_project Type ms match your s	ep4	マ ひ Size		Works Q	bace Files (*.ew ben h my_first_m B≣	w) Cancel nspm0_pro Ξ ▼ (
n a workspace from a template and save it in a new fold	× der.	File name: gpio.toqqle.output_P. Save Workspace From Template ← → Organize Name	MSPM0L1806.nortos, iarte	mplateeww iar > my_first_msp Date modified No iter	Sta m0_project Type ns match your s	ep4	マ ひ Size		Worksp Og	bace Files (*.ew ben h my_first_m ≣≕	w) Cancel nspm0_pro ∃ (
Step3	X der.	File name: gpio.toggle, output, IP. Save Workspace From Template ← → ~ ↑ → This PC → Organize → New folder ↓ ↑ Name	MSPM0(1306, nortos, larte As Windows (C;) > ti >	iar > my_first_msp Date modified No iter	Sta m0_project Type ms match your s	ep4	ン ひ Size		Worksp Op Search	h my_first_n	w) Cancel nspm0_pro Ξ (
Step3	X der.	File name: gpio_toggle_output_LP. Save Workspace From Template ← → → ← → → This PC → Organize ← New folder Name	MSPM0(1106, nontos, Jarde As Windows (C) > ti >	iar > my.first_msp Date modified No iter	St m0_project Type ms match your s	ep4	v ð Size		Worksp Op Search	h my_first_n	w) Cancel nspm0_pro ∃ (
Step3	× Aee.	File name: gpic.togqle.output.P.	MSPM0(1306, nontos, Jarte As Windows (C;) > ti >	mplateeww iar > my_first_msp Date modified No iter	St rm0_project Type ns match your s	ep4	ン ひ Size		Vorksp Op Searc	h my_first_n	w) Cancel nspm0_pro ∃ ▼ (
Step3	× Aer.	File name: gpio.toggle.output.P. Save Workspace From Template ← → ← ↑ → This PC → Organize ← New folder ↓ ↑ Name	MSPM0(1306, nortos, larte As Windows (C:) > ti >	iar > my_first_msp Date modified No iter	St m0_project Type ns match your s	ep4	マ ひ Size		Worksp Op Search	h my_first_n	w) Cancel nspm0_pro ∃ ▼ (
Step3	× .	File name: gpio_toggle_output_LP. Save Workspace From Template	MSPM0(1106, nontos, Jarde As Windows (C) > ti >	mplate.eww iar > my_first_msp Date modified No iter 07_nortos_jar.eww	St m0_project Type ns match your s	ep4	v Ö Size		Worksp Op Search	bace Files (*.ew ben	w) Cancel nspm0_pro ∃ ◄ (
Step3	× .	File name: gpio.toqqle.output.P. Save Workspace From Template ← → ← ←	MSPM0(1306, nontos, Jante As Windows (C;) → ti →	mplate.eww iar > my_first_msp Date modified No iter 27_nortos_iar.eww	St. m0_project Type ns match your s	ep4	Size	*	Of Of ○ Searc	bace Files (*.ew seen	w) Cancel nspm0_pro Ξ ▼ (
Step3	₹ -	File name: gpio.toggle, output, P. Save Workspace From Template ← → ← ↑ → This PC → Organize ← New folder ↓ ↑ Name File game: gpio.toggle, Save as type: Workspace Fil	MSPM0(1306, nortos, larte As Windows (C;) > ti > output_LP_MSPM0G330 les (*.evw)	mplataeww iar > my_first_msp Date modified No iter 27_nortos_jar.eww	St m0_project Type ns match your s	ep4	V Ö Size	×] 	© Searc	bace Files (*.ew aen h my_first_m }	w) Cancel nspm0_pro E ▼ (

Figure 3-27. Import a SDK Example

This is a simple instruction to use SysConfig with IAR.

- 1. Double-click on the .syscfg file in your project.
- 2. This opens SysConfig and allows users to configure peripherals, IO pins, and other settings.
- 3. Save the changes and switch back to IAR EWARM. Build the code example. The Files in the SysConfig Generate Files folder is updated.

iqmath_mathacl_ops_test_LP_MSPM0G3507_nortos_iar - IAR Embedded Workbench IDE - Arm 9.20.1	§ SysCorfig - C!/bi/mspm0_sdk_1_00_00_0	00.examples/nortos/LP_MSPM0L1306\driveriib\empty\iar\empty.syscfg				-	σ×
File Edit View Project TIXDS Tools Window Help Step1	FILE ABOUT		Step	2			RESTART
Workspace V V	Type Filter Text X					() <> @	• • · · •
emply_LP_MSPM0L1306_notos_iar · Debug ~	 MSPM0 DRIVER LIBRARY (6) SYSTEM (6) 	SYSCTL.©	(ADD)	OVE ALL () Problems			$_{\kappa}{}^{\kappa}$ ×
Files G igmeth_methacl_ops_test_LP_MSPM0G3507_nortos_iar*	syb Board 1/1 ♥ ⊕ DMA ④	Power & Systems Configuration		 There are no problems in the current 	t design.		
-G empty_LP_MSPM0L1306_nortos_iar - Debug • ✓	SYSCTL 1/1 CO	D Power Policy BOR Threshold	SLEEP0 0	C > Generated Files			$_{\kappa}^{*} \times$
	WNDT G	Enable Write Lock Enable Sleep On Exit		File name	Category	Include in build	
E READUE Had	ADC12	Enable Event on Pending		ti_msp_dl_config.c	MSPM0 Driver Library		8
CREADME md C	OPA G	FCC Configuration	5	v empty.syscfg	MSPM0 Driver Library Configuration Script		8
SysConfig_Generated_Files.ipcf	- COMMUNICATIONS (5)	Enable FCC		3 Total Files			e
	UART-LIN MCAN SPI	Flash Controller (FlashCtl) Configuration		WSPM0L130X (Device) VOPN-32(RHB) (Package)			$_{\kappa}{}^{*}$ ×
	TIMERS (4) TIMER - CAPTURE TIMER - COMPARE	Power & Systems Interrupts Enable Interrupts	None	× SWITCH			
	TIMER	Clock Configuration Use Clock Tree Enable Check for Clock Stabilization		 25 Pin Available 16 26 Pin Available 15 27 Warning 13 28 Power 12 30 Goround 11 31 Fixed (NNA) 10 32 			
		MCLK (Main Clock) MCLK Source Main Clock Divider	svsosc 1	v v v v v v GPI0 Used 2 GPI0 hereitette			
		MFCLK (Middle Frequency Clock)		A			
		LFCLK (Low-Frequency Clock)		^			
		MFPCLK (Middle Frequency Precision Clock)		^			
· · · · · · · · · · · · · · · · · · ·		External Clock Output (CLK_OUT)		^			
Overview empty_LP_MSPM0L1306_notos_iar gpio_software_pol_LP_MSPM0G3507_notos_iar + >		Internal Oscillators		^			
Build							
Messages							

Figure 3-28. Use SysConfig With IAR



3.4.2.3 Example Download and Debug

Follow the steps below to build the example under IAR:

- 1. To build the example, right-click in the project and select *Make*. Note that SysConfig projects automatically generates files in the *SysConfig Generated Files* folder.
- 2. Click the *Download and Debug* button to download the code.
- 3. Now, start to debug the code.



Figure 3-29. Download and Debug

3.4.2.4 Migrating Between MSPM0 Derivatives

SysConfig allows for an easier migration between MSPM0 derivatives. However some manual modifications are required on IAR. Here are the instructions:

- 1. In SysConfig, enable the Device View and click on SWITCH.
- Select the corresponding options for the new MSPM0 device and click CONFIRM. Note that SysConfig highlights any conflicts with the migration, such as unavailable pins and peripherals. Fix any conflicts as needed.
- 3. In the project options, select General Options \rightarrow Target \rightarrow Device. Select the MSPM0 device.
- 4. In the project options, select C/C++ Compiler → Preprocessor → Defined symbols. Add the device definition as per the device selected.



Software Development Instructions



Figure 3-30. Migrating Between MSPM0 Derivatives



3.4.2.5 Generate Hex Files

Here is the instruction to generate hex files in IAR. Click $Project \rightarrow Options \rightarrow Output Converter \rightarrow Generate$ additional output $\rightarrow Output$ format $\rightarrow Texas$ Instruments TI-TXT. Intel Hex or other formats also can be selected.

Category:		Factory Settings
General Options Static Analysis Runtime Checking C/C++ Compiler Assembler	Output	
Cutsom Build Linker Build Actions Debugger Simulator CADI CMSIS DAP E2/E2 Lite GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Output format: Intel Extended hex Intel Extended hex Intel Extended hex Texas Instruments TI-TXT Raw binary Simple-code	0G3507_nortos_iar.hex

Figure 3-31. Generate Hex Files

3.4.2.6 Program NONMAIN

If users do the changes on Bootloader or MCU security setting by configuring the NONMAIN, then users need to enable the NONMAIN Erase in the IAR setting, as shown in Section 3.3.2.4. Follow the steps below, otherwise, please keep the default:

- 1. Click Options → Debugger → Download → Override default .board file → Edit. Select the 2nd element and then click Okay.
- 2. Add --non_main_erase as an extra parameter.

		Factory Settings		
ash Loader Overvi	General Opbons State: Analysis Runtme Checking C(C++ Compler Assembler Output Converter Output Converter Output Converter Output Converter Uniter Debaoger Simulator CADI CMSIS DAP CMSIS TAP CMSIS	Setup Download Images Multicore Edira Options Plugins Verify download Suppress download Use Rash loader(s) Verify download Vovenide default. board file SPROJ_DIRS'upio_Joggle_output_hiz_LP_MSPMOL? Edit Edit Verform mass erase before flashing		×
Range	Offset/Address	Loader Path	oader Reloca	ОК
CODE : 0x0 - 0xffff	-	\$TOOLKIT_DIR\$\config\flashloader\TexasInstruments\FlashMSPM0L1306.flash		Cancel
ODE: 0x41c00000	- 0x41c001ff -	\$TOOLKIT_DIR\$\config\flashloader\TexasInstruments\FlashMSPM0Lx_nonMain.flash -		
				New

Memory range JLC	
⊖ All	OK
Start: 0x41c00000 End: 0x41c001ff	Cancel
Relocate	
Offset:	
Absolute address: 0x0	
Flash loader path:	
<pre>\$TOOLKIT_DIR\$\config\flashloader\TexasInstruments\Flas</pre>	hM
RAM load address: 0x0	
Extra parameters:	
Extra parameters: non_main_erase	
Extra parameters: non_main_erase	
Extra parameters: ron_main_erase	_
Extra parameters: non_main_erase	^

Figure 3-32. Program NONMAIN

Note

Extreme care needs to be taken when erasing and programming NONMAIN. If done incorrectly like losing connection in NONMAIN programming, then the device becomes locked in a permanently unrecoverable state.

3.4.3 Keil Quick Start

3.4.3.1 Environment Setup

Unlike the IAR, this is OK to use old version, Keil, however, remember to update the MSPM0 CMSIS-Pack.

3.4.3.1.1 MSPM0 CMSIS-Pack Setup

The Pack installer needs to be installed first before the MSPM0 is developed. Here are the steps to update MSPM0 CMSIS-Pack:

1. In µVision, open Pack Installer through quick guide or select $Project \rightarrow Manage \rightarrow Pack$ Installer.

 Wission [Non-Commercial Use License]

 File
 Edit
 View
 Project
 Flash
 Debug
 Peripherals
 Tools
 SVCS
 Window
 Help

 Image: Solution of the s

Figure 3-33. Open Pack Installer

2. In Pack Installer, search MSPM0 on the left side in the search text box. Then, the corresponding MSPM0 family is shown on the screen.

🕸 Pack Installer - C:\Users\a0508923\AppData\Local\Arm	\\Packs
File Packs Window Help	
Device: Texas Instruments - MSPM0G1X0X_G3X0X Ser	ies
d Devices Boards 1	4
Search: MSPM0 - X	
Device 2/	Summary
E All Devices	50 Devices
🖃 🌳 Texas Instruments	50 Devices
🗄 🏤 MSPM0C110X Series	2 Devices
🕀 🏤 MSPM0G Series	12 Devices
MSPM0G1X0X_G3X0X Series	12 Devices
🕀 🏤 MSPM0L Series	10 Devices
MSPM0L11XX_L13XX Series	10 Devices
HISPMOL122X_L222X Series	4 Devices
11	1

Figure 3-34. Search Device

3. Select the device to install a pack. Then on the right side, install the device-specific pack.

🚳 Pack Installer - C:\Users\a0508923\AppData\Local\Arm	1\Packs					-		×
File Packs Window Help								
∂ Device: Texas Instruments - MSPM0L11XX_L13XX Seri	es							
1 Devices Boards	4	4	Packs Examples					₽
Search: MSPM0 - X 🖃		Par	ck 2. Install selecte	d _₄ pack	Description			
Device /	Summary	P	Device Specific	2 Packs	MSPM0L11XX_L13XX Series	elected		
🖃 🍄 All Devices	50 Devices		■ TexasInstruments::MS	🔅 Install	Device Family Pack for Texas	Instruments MSPM0L11XX	_L13XX Se	e
🖃 🤗 Texas Instruments	50 Devices			🚸 Deprecated	Device Family Pack for Texas	Instruments MSPM0L Serie	25	
MSPM0C110X Series	2 Devices	- E-	Generic	347 Packs				_
MSPM0G Series	12 Devices		Arm-Packs::PKCS11	🔅 Install	OASIS PKCS #11 Cryptograp	hic Token Interface		
MSPM0G1X0X_G3X0X Series	12 Devices		Arm-Packs::Unity	🚸 Install	Unit Testing for C (especially	Embedded Software)		
MSPM0L Series	10 Devices		ARM::AMP	🚸 Deprecated	Software components for in	ter processor communication	on (Asym	n
H MSPM0L11XX L13XX Series	10 Devices		ARM::Arm-2D	🚸 Install	A 2D graphic library optimiz	ed for Cortex-M processors		
H MSPM0L122X L222X Series	4 Devices		ARM::CMSIS	🚸 Update	CMSIS (Common Microcon	roller Software Interface Sta	indard)	
1 Select the device			ARM::CMSIS-Compiler	🚸 Install	CMSIS Compiler extensions	for Arm Compiler, GCC, Cla	ing, and l	4
1.Select the device			ARM::CMSIS-Driver	🚸 Update	CMSIS Drivers for external d	evices		-
			+ ARM::CMSIS-Driver_Va	🚸 Install	CMSIS-Driver Validation			-
			ARM::CMSIS-DSP	🚸 Update	CMSIS Embedded Compute	Library		-
			ARM::CMSIS-FreeRTOS	🚸 Install	Bundle of FreeRTOS for Cort	ex-M and Cortex-A		-
			ARM::CMSIS-NN	🚸 Update	CMSIS NN software library of	f efficient neural network k	ernels	
			ARM::CMSIS-RTOS_Vali	🚸 Deprecated	CMSIS-RTOS Validation			-
			ARM::CMSIS-RTX	Install+	RTX RTOS implementation of	CMSIS-RTOS2 API		-
			ARM::CMSIS-View	🔅 Install	Debugger visualization of so	ftware events and statistics		-
			ARM::DMA350	🚸 Install	Pack for the DMA350 drivers			
			⊕ Arm::ethos-u-core-dri	🚸 Install	Device Driver for the Arm(R)	Ethos(TM)-U NPU.		-
			+ ARM::mbedClient	Deprecated	ARM mbed Client for Cortex	-M devices		-

Figure 3-35. Install Device Pack

4. After approving the license terms, the pack is successfully installed.

cense Agreement		
sase read the following license agreer	nent carefully.	
To continue with SETUP, you must a	ccept the terms of the License Agreement	To accept the
agreement, click the check box below	y	
All rights reserved.	s incorporated	^
Redistribution and use in source and	binary forms, with or without	
modification, are permitted provided t are met:	hat the following conditions	
* Bedistributions of source code must	retain the above convight	
notice, this list of conditions and the f	ollowing disclaimer.	~
I agree to all the terms of the prec	eding License Agreement	
— Keil Pack Unzip ————		

Figure 3-36. Approve the License



3.4.3.1.2 Sysconfig Support Setup

If SysConfig is required, follow the steps below to enable use. Make sure that SysConfig and SDK are installed ahead. Here, we use SDK v1.30 and SysConfig v1.19 as an example.

 Navigate to the SDK folder (...\ti\mspm0_sdk_x_xx_xx_tools\keil). Edit SysConfig path in syscfg.bat to match the downloaded standalone SysConfig address.

> Windows (C:) > ti > mspm0_sdk_1_30_00_03	> tools > keil		~ Ö	Sear	ch keil	م	
Name ^	Date modified	Туре	Size				
MSPM0_SDK_syscfg_menu_import.cfg	1/25/2024 11:42 AM	Configuration Sou		1 KB			L
✓ 💿 syscfg.bat	1/25/2024 11:42 AM	Windows Batch File		1 KB			t fr
	syscfg.bat - File Edit Forr @echo off set SYSCFG_I if not exist "% echo. echo Coulc echo "Update echo. exit) echo Using S echo Update set PROJ_DIR set PROJ_DIR set SYSCFG_I	Notepad mat View Help PATH="C:\ti\syscol SYSCFG_PATH%" In't find Sysconfig ate the file located the file located at t=%~1 =%PROJ_DIR:'=% FILE=%~2	nfig_1. (Tool 9 at <sc sdk</sc 	19.0\/ 6SYS(lk pat	sysconfig_cli.bat" 2FG_PATH% h>/tools/keil/syscfg.bat" .PATH% /tools/keil/syscfg.bat to use a differe	nt version'	

Figure 3-37. Edit syscfg.bat

2. In the same folder, open another file for editing. Modify the SysConfig and SDK versions and paths.



Figure 3-38. Edit MSPM0_SDK_syscfg_menu_import.cfg

3. In Keil, select *Tools* \rightarrow *Customize Tools Menu* from the menu.







4. Import MSPM0_SDK_syscfg_menu_import.cfg file into the *Customize Tools Menu*.

Customize Tools Menu X	v tools > kei	il v Ö	Search keil	Ą
Menu Content: X + 4 Syscorifig v1.18.1 - MSPM0 SDK v1 Syscorifig v1.19.0 - MSPM0 SDK v1 Prompt for Arguments Run Minimized V Run Independent	lew folder	SPM0_SDK_syscfg_menu_import.cfg Choose file in <sdk>/tools</sdk>	Date modified 1/25/2024 11:42 AM	Type Config
Command: C:\U\sysconfig_1.19.0\nw\nw.exe C:\U\sysconfig_1.19.0 Initial Folder: c:\U\mpm0_sdk_1_30_00_03 Arguments: -compiler keil = "metadata\product json" "#E" OK Cancel Help	Customize To	ools Menu Menu Content: ID ★ ★ ↓ Sysconfig v1 18 1. MSPM0 SDK v1 Sysconfig v1 19.0 - MSPM0 SDK v1 I. t will be shown on Mer	X All Entries Import Export U Content Prompt for Arguments Run Minimized Z Bun Indexendent	
	Comman Initial Folde Argument	d: C:\ti\sysconfig_1.19.0\nw\nw.exe C:\ gr: c:\ti\mspm0_sdk_1_30_00_03 gr: c:\ti\mspm0_sdk_1_Cancel 5. OK	e' han neependenk e'sysconfig_1.19.0 Help	

Figure 3-40. Import MSPM0_SDK_syscfg_menu_import.cfg File

5. The SysConfig entrance now appears on the menu. You can use SysConfig for MSPM0 development on Keil.

File	Edit	View	Project	Flash	Debug	Peripherals	Tools	SVCS	Window	Help	
ĥ		a 🍘	ХÞ		2 61	$\leftarrow \rightarrow \uparrow \uparrow$	Se	et-up PC	Lint		
٢		🛎 🗳 ·	-	bs	l_software	e_invoke_apr	Li	nt nt All C/(Files	
Projec	t							III AII C/V	See Source	11103	_
<u>_</u> ?	🖇 Proj	ect: bsl_	software_	invoke_	app_demo	o_uart_LP_MS	C	onfigure	Merge Too	l	
H		bsl_softv	vare_invo	ke_app_	demo_ua	rt_LP_MSPM0	C	ustomize	Tools Men	u	
							Sj	/sconfig	v1.18.1 - MS	SPM0 SDK v1_20_01_06	
							Sj	/sconfig	v1.19.0 - MS	SPM0 SDK v1_30_00_03	
								_			_

Figure 3-41. Finish SysConfig Setup

3.4.3.2 Import a SDK Example

Here is the guide that explains how to import a MSPM0 SDK example into Keil:

1. In Keil, select $Project \rightarrow Open Project$.



Figure 3-42. Open Project


2. Select a demo project from SDK. For the nortos example, use the .uvprojx project file. For the RTOS example, use .the uvmpw work space file. An example is shown in Figure 3-43.



Figure 3-43. Select Keil Project



3. To open the .syscfg file, double click the .syscfg file. Then, select *Tools* \rightarrow *Sysconfig* v1.19.0 - *MSPM0 SDK* v1_30_00_03. The .syscfg file opens in a separate window.

t	P 🛛 🖊	gpio_tog	gle_output.syscfg	2		
roject: gpio_toggle_output_LP_MSPM0G350	7_nortos_keil	1 /*	*			
gpio_toggle_output_LP_MSPM0G3507	_nortos_keil	2 *	These argume	nts were	used when t	this file was
E Gource		3 ^ 4 *	/ the GUI	OF CLI. R	un CLI WIT	1neip. ioi
qpio_toggle_output.c		5 //	CcliArgsd	levice "MS	PMOG350X" -	-package "LQE
1 gpio_toggle_output.syscfg		6 //	CliArgsb	oard /ti/l	boards/LP_N	ISPM0G3507
startup_mspm0g3507_uvision.	s	7 8 /*	*			
			*		man and the	_
C:\ti\mspm0_sdk_1_21_00_00_inter	rnal\examples\norto	os\LP_MSPM0G	3507\driverlib\gpic	o_toggle_outp	ut\keil\gpio_to	99
File Edit View Project Flash	Debug Peripherals	s Tools SVC	S Window Hel	lp		
i 🗋 💕 🔛 🖉 i X 📭 🕵 i 🤊	0 0 ← → 1	Set-up	PC-Lint			_
	toggle output IP	Lint				_
Project	_toggle_output_tr_	Lint All	C/C++ Source Files	5		
Project: apio toggle output LE	MSPM0G3507 port	to Config	ure Merge Tool			
apio togale output LP MS	PM0G3507 nortos k	cei Custon	nize Tools Menu		e us	se
Source					Rui	n
apio togale output	e.	Syscon	fig v1.18.1 - MSPM0) SDK v1_20_01	_06	
gpio_toggle_output	syscfa	3 Syscon	fig v1.19.0 - MSPM0) SDK v1_30_00	_03 MSPI	40
					, JL/D	Ja
SysConfig - C:\ti\mspm0_sdk_1_30_00_03	\examples\nortos\LP	P_MSPM0G350	\driverlib\gpio_togg	gle_output\gpi	io_toggle –	- 🗆 X
\$ SysConfig - C:\ti\mspm0_sdk_1_30_00_03 FILE ABOUT 4 \Tau Type Filter Text X	$\langle examples \rangle$ nortos $\langle LP \rangle$	P_MSPM0G350 ware → Board	\driverlib\gpio_togg	gle_output\gpi	io_toggle (i) <>	RESTART
 SysConfig - C:\ti\mspm0_sdk_1_30_00_03 FILE ABOUT 4	\examples\nortos\LP	MSPM0G350 ware → Board	\driverlib\gpio_togg	gle_output\gpi	io_toggle (i) <>	RESTART
SysConfig - CAti\mspm0_sdk_1_30_00_03 FILE ABOUT 4 ₹ Type Filter Text X ≪ > PROJECT CONFIGURATIO Project Config 1/1	\examples\nortos\LP : ← → Softw Board ⊙ (v_MSPM0G350 ware → Board ⊕ ADD	\driverlib\gpio_togg	gle_output\gpi	io_toggle () <> lems	$- \Box \times \\ \frac{\text{RESTART}}{\textcircled{0}} \bigcirc \vdots \\ \frac{\omega^{n}}{\omega} \times \\ $
SysConfig - C\ti\mspm0_sdk_1_30_00_03 FILE ABOUT 4	\examples\nortos\LP : ← → Softw Board ⊙ (y_MSPM0G350 ware ► Board (+) ADD	\driverlib\gpio_togg	gle_output\gpi	io_toggle (i) <> lems	$- \Box \times \\ \frac{\text{RESTART}}{\textcircled{0} \odot } \\ \frac{1}{2} \times \\ \frac{1}{2}$
SysConfig - C\ti\mspm0_sdk_1_30_00_03 FILE ABOUT 4 Type Filter Text X ≪	\examples\nortos\LP : ← → Softw Board ۞ (Debug Confi	MSPM0G350 ware ► Board ⊕ ADD iguration	\driverlib\gpio_togg	gle_output\gpi	io_toggle (i) <> lems	$ \begin{array}{c} \square \\ \\ \hline \\ \\ \hline \\ \\ \hline \\ \\ \\ \\ \\ \\ \\ \\ \\ \\$
SysConfig - C1thimspm0_sdk_1_30_00_03 FILE ABOUT 4	 (examples\nortos\LP) ← → Softw Board ⊙ (Debug Confi Debug Enab 	MSPM0G350 ware ► Board ⊕ ADD iguration ble On SWD Pir	Ndriverlib\gpio_togg	gle_output\gpi	io_toggle (i) <> lems WARNINGS	ESTART
SysConfig - C\th\mspm0_sdk_1_30_00_03 FILE ABOUT Type Filter Text × « PROJECT CONFIGURATIO Project Config 1/1 ♥ ↔ MSPM0 DRIVER LIBRARY SYSTEM (9) Board 1/1 ♥ ↔ DMA ↔	 (examples\nortos\LP) (c) → Softw Board (c) Debug Confi Debug Enab 	ware ► Board ⊕ ADD iguration ble On SWD Pir	\driverlib\gpio_togg	(i) Prob ERRORS	io_toggle (i) <> lems WARNINGS ()	$\begin{array}{c c} & \square & \times \\ \hline \\ \hline \\ \hline \\ \hline \\ \hline \\ \hline \\ \\ \hline \\ \\ \\ \\$
i SysConfig - C\thimspm0_sdk_1_30_00_03 FILE ABOUT 4 Type Filter Text × « PROJECT CONFIGURATIO Project Config 1/1 Project Config 1/1 • SYSTEM OBRIVER UBBARY > SYSTEM 09 Board 1/1 • GPIO 1 • ⊕ 	 (examples\nortos\LP) (c) → Softw Board (c) Debug Confi Debug Enab Global Pin C 	ware ► Board ⊕ ADD aguration ble On SWD Pin configuration	\driverlib\gpio_togg	(i) Prob ERRORS	io_toggle (i) <> lems WARNINGS ()	$\begin{array}{c} \square \\ \hline \\$
SysConfig - C\tr\mspm0_sdk_1_30_00_03 FILE ABOUT ABOUT FOR Filter Text X ≪ Y PROJECT CONFIGURATIO Project Config 1/1 ♥ ⊕ MSPM0 DRIVER LIBRARY Y SYSTEM (9) Board 1/1 ♥ ⊕ DMA	 (examples\nortos\LP) ← → Softw Board ⊙ Debug Confi Debug Enab Global Pin C 	ware ► Board ADD arr of the second secon	\driverlib\gpio_togg	i Prob ERRORS Location	io_toggle (i) <> lems WARNINGS () 1, Detai	C C X RESTART () O i μ ^π × suppressed Ο Is
SysConfig - C\trimspm0_sdk_1_30_00_03 FILE ABOUT Type Filter Text X ≪ PROJECT CONFIGURATIO Project Config 1/1 ♥ ↔ SYSTEM (9) Board 1/1 ♥ ↔ GPIO 1 ♥ ↔ MATHACL ↔ Configuration NVM ↔	 (examples\nortos\LP) Examples > Softw Board ⊙ Debug Confi Debug Enab Global Pin Context 	ware → Board () ADD () Finite iguration ble On SWD Pinite configuration	\driverlib\gpie_togg	i Prob RRRORS	io_toggle (i) <> lems WARNINGS () t ₁ Detai	 C × RESTART Φ Φ : μⁿ × Δ SUPPRESSED 0
SysConfig - C\th\mspm0_sdk_1_30_00_03 FILE ABOUT 4	 ⟨examples\nortos\LP ⟨examples\nortos\LP Board () Debug Confi Debug Enable Global Pin C Enable Glob 	ammodia Solo ware ► Board	\driverlib\gpio_togg	(i) Prob ERRORS Location	io_toggle i <> lems WARNINGS 0 t ₁ Detai prated Files	$\begin{array}{c} \square \\ \\ \hline \\ \\ \hline \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\$
SysConfig - C\thimspm0_sdk_1_30_00_03 FILE ABOUT 4	 (examples\nortos\LP) ← → Softw Board () Debug Confi Debug Enab Global Pin C Enable Glob Configure II 	amon and a more	Vdriverlib\gpio_togg	() Prob C Prob C Prob C C C C C C C C C C C C C C C C C C C	io_toggle i <> lems WARNINGS o t ₁ Detai prated Files	$\begin{array}{c c} & \square & \times \\ \hline & \textbf{RESTART} \\ \hline & \textcircled{\bullet} & \textcircled{\bullet} & \vdots \\ & & & & \\ & & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ $
SysConfig - C\th\mspm0_sdk_1_30_00_03 FILE ABOUT 4	 (examples\nortos\LP) ← → Softw Board () Debug Confi Debug Enable Global Pin C Enable Glob Configure U Configure U 	a MSPM0G350 ware ► Board (ADD) iguration ble On SWD Pir configuration bal Fast-Wake Inused Pins	\driverlib\gpio_togg	() Prob ERRORS () Location () Gene Filter: al	io_toggle i <> lems wARNINGS 0 t Detai prated Files	$\begin{array}{c c} & \square & \times \\ \hline & \textbf{RESTART} \\ \hline & \textcircled{\bullet} & \textcircled{\bullet} & \vdots \\ & & & & \\ & & & & \\ & & & \\ & & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & $
SysConfig - C\ti\mspm0_sdk_1_30_00_03 FILE ABOUT 4 = Type Filter Text × ≪ > PROJECT CONFIGURATIO Project Config 1/1	 (examples\nortos\LP) ← → Softw Board () Debug Confi Debug Enable Global Pin C Enable Glob Configure U Generate Period 	a MSPM0G350 ware ► Board (ADD) iguration ble On SWD Pir configuration bal Fast-Wake print Fast-Wake print Fast-Wake print Fast-Wake print Fast-Wake print Fast-Wake Pirst Pir	\driverlib\gpio_togg	(i) Prob ERRORS O Location Filter: al File name	io_toggle (i) <> lems WARNINGS () t Deta prated Files I e	$\begin{array}{c c} & \square & \times \\ \hline & \textbf{RESTART} \\ \hline & \textcircled{O} & \textcircled{O} & \vdots \\ & & & & \\ & & & \\ & & & & \\ & & & &$
SysConfig - C\ti\mspm0_sdk_1_30_00_03 FILE ABOUT 4 = Type Filter Text X ≪	 (examples\nortos\LP, ← → Softw Board () Debug Confi Debug Enable Global Pin C Enable Glob Configure U Generate Per 	ware ➤ Board ADD aguration ble On SWD Pin configuration bal Fast-Wake Inused Pins eripherals & Pin	\driverlib\gpio_togg	(i) Prob ERRORS D Location Filter: <u>al</u> File name	io_toggle (i) <> lems WARNINGS () t Detai prated Files 1 e	$\begin{array}{c c} & \square & \times \\ \hline & \textbf{RESTART} \\ \hline & \textcircled{O} & \textcircled{O} & \vdots \\ & & & & \\ & & & \\ & & & \\ & & & \\ & & & & \\ & & & & \\ & & & \\ & & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & & \\ & & $
\$ SysConfig - C:\ti\mspm0_sdk_1_30_00_03 FILE ABOUT 4 Type Filter Text × « PROJECT CONFIGURATIO Project Config 1/1 + MSPM0 DRIVER LIBRARY > SYSTEM (9) Board 1/1 ⊕ MATHACL ⊕ SYSCTL 1/1 ⊕ SYSTICK ⊕ WWDT ⊕ ANALOG (6) ADC12 ⊕ 	 (examples\nortos\LP) ← → Softw Board ① Debug Confi Debug Enable Global Pin C Enable Glob Configure U Generate Per Initialization 	MSPM0G350 ware ► Board (+) ADD (== iguration ble On SWD Pin configuration bal Fast-Wake inused Pins eripherals & Pin Priority Confi	Adrivertib\gpio_togg	(i) Prob ERRORS (i) Location Kilter: al File name	io_toggle (i) <> lems WARNINGS (i) t, Detail rated Files I e ssp_dl_config.c	$\begin{array}{c c} & \square & \times \\ \hline \\$
SysConfig - C\ti\mspm0_sdk_1_30_00_03 FILE ABOUT 4 ¬ Type Filter Text × « Image: SysConfig Y ROJECT CONFIGURATIO Project Config × « Image: SysConfig Y ROJECT CONFIGURATIO Project Config Y (*) • (*)	 (examples\nortos\LP) ← → Softw Board () Debug Confi Debug Enable Global Pin C Enable Glob Configure U Generate Per Initialization Initialization 	ware ► Board ADD ADD F Guration ble On SWD Pir configuration bal Fast-Wake Inused Pins eripherals & Pi Priority Confi	\driverlib\gpio_togg	() Prob C Prob C Prob C C C C C C C C C C C C C C C C C C C	io_toggle i <> lems WARNINGS 0 1 t Detai prated Files 1 e sp_dl_config.co	$\begin{array}{c c} & \square & \times \\ \hline \\$
\$ SysConfig - C\\ti\mspm0_sdk_1_30_00_03 FILE ABOUT 4 ¬ Type Filter Text × « • PROJECT CONFIGURATIO Project Config 1/1 ● + • PROJECT CONFIGURATIO Project Config 1/1 ● + • SysTem (9) ■ Board 1/1 ● + • SYSTEM (9) Board ■ DMA	 (examples\nortos\LP) ← → Softw Board () Debug Confi Debug Enable Global Pin C Enable Glob Configure U Generate Per Initialization Initialization 	MSPM0G350 ware ► Board (→ ADD) iguration ble On SWD Pir configuration bal Fast-Wake priority Confi n Priority 0 Delevin	Ndriverlib\gpio_togg	() Prob ERRORS () Location Filter: al File name ti_m () ti_m	io_toggle i <> lems WARNINGS o t Deta t Deta sp_dl_config.c	RESTART (Cat Cat Cat Cat Cat Cat Cat
 SysConfig - C\ti\mspm0_sdk_1_30_00_03 FILE ABOUT Type Filter Text × « PROJECT CONFIGURATIO Project Config 1/1 ♥ ⊕ MSPM0 DRIVER (JBRARY > SYSTEM (9) Board 1/1 ♥ ⊕ DMA ⊕ GPI0 1 ♥ ⊕ MATHACL ⊕ Configuration NVM ⊕ RTC ⊕ SYSTICK ⊕ WWDT ⊕ SYSTICK ⊕ WWDT ⊕ ADC12 ⊕ DAC12 ⊕ GPAMP ⊕ 	 (examples\nortos\LP) ← → Softw Board () Debug Confi Debug Enable Global Pin C Enable Glob Configure U Generate Per Initialization Initialization 	MSPM0G350 ware ► Board () ADD iguration ble On SWD Pir- configuration bal Fast-Wake inused Pins eripherals & Pi- n Priority Confi n Priority 0 n Priority 1	\driverlib\gpio_togg REMOVE ALL ns ♥ in Assi 0 iguration SY ▼ PW ▼	() Prob Content of the second	io_toggle i <> lems wARNINGS o t Detai prated Files l e sp_dl_config.t	RESTART RESTART Call
 SysConfig - C\ti\mspm0_sdk_1_30_00_03 FILE ABOUT 4 = Type Filter Text × « PROJECT CONFIGURATIO Project Config 1/1 Project Config 1/1 • SYSTEM DRIVE9 (JBRARY > SYSTEM 01 10 DMA GPI0 1 SYSTICK SYSTICK WWDT SYSTICK WWDT COMP DAC12 GPAMP OPA OPA 	\examples\nortos\LP : ← → Softw Board ⊙ Debug Confi Debug Enable Global Pin C Enable Glob Configure U Generate Per Initialization	ADD ware ► Board ADD aguration ble On SWD Pir- configuration	<pre>\driverlib\gpio_togg REMOVE ALL</pre>	(i) Prob ERRORS O Location C S Gene Filter: al File name ti_m ti_m E tere	io_toggle io_toggle lems WARNINGS o t Deta rated Files I e sp_dl_config.t at.dot	$\begin{array}{c c} & \square & \times \\ \hline RESTART \\ \hline \hline \hline \hline \hline \hline \hline \hline \hline \\ \hline \hline \\ \hline \\ \hline \\ \hline $

Figure 3-44. Open .syscfg file

3.4.3.3 Example Download and Debug

Here is the guide that explains how to download the code into MSPM0 based on Keil:

1. Right-click project files, then select open options for target

Project: gpio_toggle_output_LP_MS	SPM0G3507_nortos_ke 1 🖂 / *	
🚊 ᇶ gpio_toggle_output_LP_MSPM	0G3507_nortos_keil 2 * Copyright (c) 2023, Texas	Instrumen
🖃 🗁 Source	🔆 Options for Target 'gpio_toggle_output_LP_MSPM0G3507_nortos_keil'	Alt+F7
gpio_toggle_output.c gpio_toggle_output.c gpio_toggle_output.sy startup_mspm0g350x_ ti_msp_dl_config.h	Add Group Manage Project Items Rebuild all target files	n e
📶 ti_msp_dl_config.c	Build Target	F7 ^m
🛛 🔁 Driverlib	Show Include File Dependencies	ry form m nditions





2. Select a debugger from the *Target Options* window. To use XDS-110, select *CMSIS-DAP Debugger*. If J-Link is required, then select *J-LINK/J-TRACE Cortex*.

Options for Target 'gpio_toggle_output_LP_MSPM0G3507_nortos_keil'						
Device Target Output Listing User C/C++ (AC6) At	m Linker Debug Utilities					
C Use Simulator with restrictions Settings Limit Speed to Real-Time	CMSIS-DAP Debugger Settings ULINK Pro Cortex Debugger					
Load Application at Startup Iv Run to main() Initialization File:	ClarkFulls Debugger Cost (J-LINK / J-TRACE Cortex Initializat Models Cortex-M Debugger ST-Link Debugger					
Restore Debug Session Settings Image: Contract of the setting settin	NULink Debugger Restor SiLabs UDA Debugger SiLabs UDA Debugger Image: Silabs UDA Debugger					
CPU DLL: Parameter: SARMCM3.DLL -MPU	Driver DLL: Parameter: SARMCM3.DLL -MPU					
Dialog DLL: Parameter: DARMCM1.DLL PCM0+	Dialog DLL: Parameter: TARMCM1.DLL pCM0+					
Warn if outdated Executable is loaded Warn if outdated Executable is loaded Manage Component Viewer Description Files						
OK Car	Icel Defaults Help					

Figure 3-46. Select the Debug Pane

3. Click on the *Settings* button. On the *Debug* tab, make sure the settings match with Figure 3-47 and Figure 3-48.

XDS110

Options for Target 'gpio_toggle	_output_LP_MSPM0G3507_nortos_keil'	×
Device Target Output Listing	Jser C/C++ (AC6) Asm Linker Debug Utiliti	ies
C Use Simulator with restriction	Settings G Use: CMSIS-DAP Deb	ugger 🗸 Settings
✓ Load Application at Startup	Run to main() I Load Application at Star	tun I▼ Run to main()
CMSIS-DAP Cortex-M Target Drive	r Setup	×
Debug Trace Flash Download	Pack	
CMSIS-DAP - JTAG/SW Adapter	SW Device	
Any	IDCODE Device Name	Move
Serial No: MG350001	SWDIO Ox6BA02477 ARM CoreSight SW-	-DP Up
Firmware Version: 1.2.0		Down
SWI Port: SW	Automatic Detection ID CODE:	
Max Clock: 5MHz	C Manual Configuration Device Name:	
	Add Delete Update	AP: 0x00
Debug		
Connect & Reset Options		- Download Options
Reset after Connect	Cache Memory	Download to Flash
Log Debug Accesses	Stop after Reset	
	OK Cancel	Help

Figure 3-47. Check the Setting of XDS110 Probe



J-Link

vice Target Output Listing User C/C+ Use Simulator <u>with restrictions</u> Limit Speed to Real-Time	+ (AC6) Asm Linker Debug Utilities Settings	ettings
Load Application at Startup	nain() 🔽 Load Application at Starture 🔽 Run to ma	iin()
rtex JLink/JTrace Target Driver Setup		×
ebug Trace Flash Download		
SN: 59704984 Device: J-Link HW : V9.70 dll : V7.94/-	IDCODE Device Name M SWD Image: Comparison of the state of the s	love Up
FW: J-Link V9 compiled May 7 20. For: Max Clock: SW JTAG SW Auto Clk	Automatic Detection ID CODE: Manual Configuration Device Name: Add Delete Update IR Ien:	
Connect & Reset Options Connect: Normal Reset: Normal Reset after Connect	Cache Options Cache Options Cache Code Cache Memory Download Options Cache Down Download Options Code Down Download Options	nload
Interface TCP/IP Image: Scan Network Settings Scan 127.0.	Port (Auto: 0) Autodetect JLink I 0 1 0 Ping	nfo Cmd

Figure 3-48. Check the Setting of J-Link Probe

4. Click on the *Flash Download* tab and check whether the description matches Figure 3-49. If this does not match, then click on the *Add* button and select the corresponding MSPM0 MAIN option. The device type is *On-chip Flash*. At last select *Reset and Run*.

Cortex JLink/JTrace Target Driver S	etup			Х
Debug Trace Rash Download Download Function Carase Full Chip Carase Sectors C Do not Erase 2 Programming Algorithm	I ✓ Program ✓ Venfy ✓ Reset and F	RAM for / Start: [Ngorithm x20200000 Size: 0x8000	
Description MSPM0G MAIN 128KB	Device Size 128k	Device Type On-chip Flash	Address Range 00000000H - 0001FFFFH	
3	Add	Start: Remove	Size:	
			OK Cancel Apple	

Figure 3-49. Flash Download Setting



5. Click the *Build* button to build the project, then click the *Load* button.

C:\ti\mspm0_sdk_1_30_00_03\examples\nortos\LP_MSPM0G3507\driverlib\gpio_tog



Figure 3-50. Download Project

6. To build the FreeRTOS supported example, select *Project* → *Batch Setup* and select all the project targets for the build. Next, select *Batch Build* to build all the projects in the workspace.

C:(ti\mspm0_sdk_1_20_01_06\kernel\freertos\builds\LP_MSPM0G3507\release\keil\freertos_builds_LP_MSPM0G3507_release_keil.uvprojx - µVision	🔢 C.\th\mspm0_sdk_1_20_01_06\kernel\freertos\builds\LP_MSPM0G3507\release\keil\freertos_builds_LP_MSPM0G3507_release_keil.uvprojx - µVision
	Hie bait view Project Hash Debug Perpherals 100/5 SVCS Window Help
Step1	🖉 🖾 🖾 🧼 - 🗟 🚆 freetos_builds_LP_MSP 🔽 🛠 着 👼 🔶 匆 🗴 Step2
Project Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.S.) Image: Transmission of the second state (E.	Popet Bach Build Image: State State Image: State State Image: State State State Image: State State Image: State State
Proj. Chan. D Fun. II. Tem.	

Figure 3-51. Build RTOS Example Under Keil

3.4.3.4 Migrating Between MSPM0 Derivatives

SysConfig allows for an easier migration between MSPM0 derivatives. However some manual modifications are required on Keil. Follow the steps below:

- 1. In SysConfig, enable the Device View and click on SWITCH.
- Select the corresponding options for the new MSPM0 device and click CONFIRM. Note that SysConfig highlights any conflicts with the migration, such as unavailable pins and peripherals. Fix any conflicts as needed.
- 3. In the Keil IDE, open the Device tab in project options, and select the new MSPM0 derivative.
- 4. Update the device definition by selecting *C*/*C*++ (*AC6*) → *Preprocessor Symbols* → *Define*. Add the device definition as per the device selected.
- 5. Update the linker file in *Linker* → *Scatter File*. The MSPM0 SDK includes default files for all MSPM0 derivatives at <*sdk*>*source*/*ti*/*devices*/*msp*/*m0p*/*linker_files*/*keil*.
- 6. Add the startup file of the new derivative to the project and remove existing one. The MSPM0 SDK includes default files for all MSPM0 derivatives at <sdk>\source\ti\devices\msp\m0p\startup_system_files\keil.



			RESTART	A Switch Board or De	evice			Options for Target 'empty_LP_	MSPM0L1306_nort	ros_keil'		×
	Step1	(i) <>	•••	This will migrate the cur	rent configuration to Steep	vice selected below. Any incompat	atibilities	Device Target Output Listing		Asm Linker Debur	a Utilities	
⊕ ADD IF REMOVE ALL	Problems There are no problems in the current	ent design.	_к л Х	will be flagged as errors. The migration can be un not modified, and likely o manually.	done by using ctrl + z or the histor contains device-specific settings.	y view. Any underlying project or mai These settings will need to be migrat	akefile is ited	Software Packs Vendor: Texas Instruments Device: MSPM0L1105	_ occp.	Software Pack Pack: Texasine	struments.MSPM0L_DFP.1.00	0.00
P0 •	Senerated Files		к ^я ×	Setting	Current Value	New Value		Toolset: ARM Search:		URL: https://w	ww.ti.com/msp430/	
· · · · · · · · · · · · · · · · · · ·	Filter: all	Category Inclus	-	Board Device	MSPM0L130X	None MSPM0L134X	* *	- 🖉 Texas Instruments	MSF	PMOL110x microcontrollers (*	MCUs) are part of the MSP hig	ahiy- ^
	ti_msp_dl_config.c	MSPM0 Driver	• 8	Part Package	Default VQFN-32(RHB)	VSSOP-28(DGS28)	- -	SPM0L Series	Arm [®] frequ	ratedultra-low-power 32-bit 1 © Cortex®-M0+ coreplatform uency. These optimized MCL	MCU family based on the enha operating at up to 32-MHz Js offer high-performance anal	unced log
	₿ ti_msp_dl_config.h	MSPM0 Driver Library	• •	Lock PinMux		CANCEL C	CONFIRM	MSPM0L11	5 perip 40°C)6 The	theral integration, support ext to 105°C, and operate with MSPM0L110x devices prov tram memory with 4KB SBAN	ended temperature ranges from supply voltages from 1.62 V to ride up to 64KB embedded flas A These MCUs incomprate a b	m - 9 3.6 V. sh
MAND	Event.dot	MSPM0 Driver Library	• •						spee need DM/	d on-chip oscillator with ana d for an external crystal Addr A, 16 and 32-bit CRC accele	sccuracy up to ±1%, eliminating tional features include a 3-cha rator,and a variety of high-	g the innel
Ť	B gpio_input_capture.syscfg	Configuration Script	8						perfo with ampi	smance analog peripherals s configurable internal voltage lifier, and an on-chip temper.	such as one 12-bit1.45-Msps / a reference, one general-purpo ature sensor. These devices a	ADC
~	4 Total Files		0									pose v
rrst 👻	WORN-32(RHB) (Package) SWITCH 25		R V									
Options for Target 'empty_LP_MSP!	MoL1306_norStep4		×	Options for Target 'emp	ty_LP_MSPM0L1306-Step	5	>	< Project	ct: emptSter	6306_nortos_keil	φ 🗵	
Device Target Output Listing User	C/C++ (AC6) Asm Linker Debug	Utilities	C	Device Target Output Li	sting User C/C++ (AC6) Asm	Linker Debug Utilities	_		npty_LP_MSPM0L1	306_nortos_keil		
Preprocessor Symbols Define: MSPM0L1105				Make RW Sections	Position Independent	<u>R</u> /O Base: 0x00000000	_	e	empty.c	í.		
Undefine:				Don't Search Stands	rostion independent and Libraries	R/ <u>W</u> Base 0x20000000			- ti_msp_dl_cor	nfig.h nfig.c		
Language / Code Generation	Warnings: MISRA Compatible 💌 Lan	iguage C: c99 💌		 Heport might fail Co 	nditions as Errors			e 2	Driverlib	n0I1105_uvision.s		
Optimization: -O2 Unk-Time Optimization	Tum Warnings into Errors Langu	age C++: c++11 Short enums/wchar							- driverlib.a			
Split Load and Store Multiple One ELF Section per Function	☐ Read-Only Position Independent ☐ <u>R</u> ead-Write Position Independent	t I No Auto Includes		Scatter File	i.sct		Edit					
Include Paths Misc Controls	party/CMSIS/Core/Include,/./././.so	urce	1	Misc controls			<u>^</u>					
Compiler control string	am-none-eabi -mcpu=cortex-mûplus -c short-enums fshort-wchar	Ĵ		Linker control string	10+ °.o r ".\mspm0l1105.sct"		Ŷ					
	OK Cancel Defaults	Heip			OK Cance	Defaults	Help					

Figure 3-52. Migrating Between MSPM0 Derivatives

3.4.3.5 Generate Hex Files

Here is the instruction to generate hex files in Keil. Click $Project \rightarrow Options \rightarrow Output \rightarrow Create Hex File \rightarrow OK$. You can select the paths through click *Select Folder for Objects* to locate the HEX file. The default path is the object folder under project file.

Create Executable: \Objects\add	12 14bit resolution LP MSPM	ADL 1306 nortos keil
✓ Debug Information		Create Batch File
I Create HEX File		
Browse Information		
Create Library: \Objects\adc12	14bit resolution LP MSPM0L1	306 nortos keil lib

Figure 3-53. Generate Hex Files

3.4.3.6 Program NONMAIN

If users make the changes on Bootloader or MCU security setting by configuring the NONMAIN, as shown in Section 3.3.2.4, then users need to enable the NONMAIN Erase in the IAR setting as well. Follow the steps below, otherwise and keep the default:

- 1. Click Options \rightarrow Debug \rightarrow Settings \rightarrow Flash Download.
- 2. Add the NONMAIN programming algorithm, and then click OK.



Figure 3-54. Program NONMAIN

4 Hardware Design Instructions

4.1 Obtaining a MSPM0 Package

To obtain a MSPM0 package, use the Ultra Librarian tool on TI.com. The detailed instructions are as below.

1. Go to the start page of the Ultra Librarian tool under the MSPM0 device page using the steps.

MSPM0L1106 I Previ	^{EW} Step 1		Step2	
Product details Technical de	ocumentation	Design 8	development	Ordering & quality
All Hardware development	Software deve	lopment	CAD/CAE sym	bols Step3
Package	Pins		Download	d
SOT-23-THN (DYY)	16		View opti	ions Step4
VQFN (RGE)	24		View opti	ions
VQFN (RHB)	32		View opti	ions

Figure 4-1. Ultra Librarian Tool Start Page

2. Select the desired CAD format and pin ordering to obtain the Altium design library file.



- Figure 4-2. Ultra Librarian Tool Device Selection
- 3. The Altium Designer library file is used as an example.





Texas Instruments -	XMSM0L1106TDYYR
Choose CAD Format(s)	Return to Previews
3D CAD Model ►	Mentor ►
Altium 🔻	Pulsonix ►
✓ Altium Designer ♣: = Step6	Quadcept ►
	TARGET 3001! ►
Autodesk >	Zuken ►
Cadence ►	
DesignSpark ►	
KICAD ►	
Symbol Pin Ordering Sequential Sequential Functional List the list	Footprint Units English (mil)
Step8	Sitra Librarian Terms And Conditions
Step9 / m not a robot	reCAPTCHA Privacy - Tems
Step10 St	ubmit

Figure 4-3. Ultra Librarian Tool CAD Download

4. Run the *Altium Designer* script.

9 🖷	🗐 🚔 to 🔿		File Edit View	Project Tools	Run	Window Help	
<u>E</u> ile <u>E</u> o	dit <u>V</u> iew Proje <u>c</u> t <u>P</u> lace <u>T</u> oo	ls <u>R</u> eports	Step3: Cho Projects	ose UL_Form	>	<u>R</u> un	F9
<u>N</u> e	w	•	anu c		Ξī	Run To <u>C</u> ursor	Ctrl+F9
📂 <u>О</u> р	en Step1	Ctrl+O		•	٠	Toggle Breakpoint	F5
<u>C</u> lo	ose	Ctrl+F4	Q Search		+		Ctrl+F7
y Op	en Project		★ \$7060P			Stan Into	E7
•			Workerson D		-	Step Into	F7
Downloads > ul_XMSM01	L1106TDYYR > AltiumDesigner	>	A BUL Import)	MSM0L1106TDY	•	Step Over	F8
			🔺 🖿 Source Doo	cuments	2		Ctrl+F3
^ 🗆	Name	I	UL_Impo	ort.pas n.pas		<u>A</u> dd Breakpoint	
	Previews	4					
		*					
	Step2: Oper	n this file					
		0/D D 10					
		TYK.PIJSCI					

Figure 4-4. Run Altium Designer Script



5. Generate the PCB library and schematic library as shown in Figure 4-5.



Figure 4-5. Generate Library

6. Select the correct footprint under PCB Library.

		Footprints		
		Name	 Pads 	Primitives
		SOT_TDYYR_TEX	16	104
A 🗟 XMSM0L1106TDYYR.LibPkg		SOT_TDYYR_TEX-L	16	104
🖌 🖿 Source Documents		SOT_TDYYR_TEX-M	16	104
XMSM0L1106TDYYR.PcbLib Select	L D	Choose wanted foo		
ZMSM0L1106TDYYR.SchLib	D			
Projects PCB Library View Configuration		Place Add	l Delete	Edit

Figure 4-6. Select Footprint

7. Import the PCB library and schematic library.

			Ę		Available Libraries	
	Libraries		▼ + × Proje	ct Installed Search P	Path	
	Libraries S	earch Place 2N39	904 <mark>10 Proj</mark>	ect Libraries	Path	Туре
	P Miscellaneous		• ₽			
	*		ropert			
	Design Item ID		ie s			
	2N3904 2N3906	_	Aessa			
M0L1306SDYYR.SchLib	ADC-8		ges			
	Battery					
tem ID						
mponent_1 1SM0L1306SDYYR						
	ASMOL 1306SDYVR Peblib	4/27/2023 8·29 PM	м			
	ISM0L1306SDYYR.SchLib	4/27/2023 8:29 PN	M			
XN	ISM0L1306SDYYR.txt	4/27/2023 12:16 P	PM			
<						
Lib" "XMSM0	1306SDYYR Prolib"	Files (* *)				Add Library Remo
		0000	acol			
		Open Can	icei			

Figure 4-7. Import Library

4.2 Fix Pin Functions through Sysconfig

TI recommends hardware engineers use the Peripherals and Pin Assignments File to fix the pin functions with assistance from a software engineer by following the instructions in Figure 4-8.

82	Type Filter Text	\times	\leftrightarrow \rightarrow Software $ ightarrow$ Board		
୶ଌ	✓ PROJECT CONFIGURATION (1) Project Configuration Files 1/1	> +	Board ⑦		✓ Debug 1 ✓ Ø device_linker.cmd 2 - GPIOA: ✓ Ø timsp.dl.confia.c 3 + Open-Drain Output: PAØ
	 MSPM0 DRIVER LIBRARY (6) SYSTEM (7) 		Debug Configuration		> > A - SYSCTL > ● adc12_14bit_resolution_LP_MSPM0L1306_nc 5 - ADC0:
	Board 1/1 DMA 1/1		Debug Enable On SWD Pins	\checkmark	B adc12_14bit_resolution.o - [ARM/le] b + AUC12 Channel 2 Pin: PA25 b startup_mspm0130x_ticlang.o - [ARM/le] C = CPUN S = CPUN
	GPIO 1 Configuration NVM SYSCTL 1/1	 •• •• •• •• •• •• •• 	Global Pin Configuration Enable Global Fast-Wake		<pre>b t_msp_a_conig.o = [AtkWire] adc12_14bit_resolution_LP_MSPM0L1306_nc 9 = BOARD: adc12_14bit_resolution_LP_MSPM0L1306_nc 10 + Debug Clock: PA20 adc12_14bit_resolution.d 11 + Debug Data In Out: PA19 ccsObjs.opt 12</pre>
	WWDT	Ð	Configure Unused Pins Generate Peripherals & Pin Assignments File		 device.cmd.genlibs device.opt Event.dot
	ADC12 1/1 COMP GPAMP	 ↔ ↔ ↔ 	Voltage Configuration		l makefile l objects.mk l peripheralPinAssignments.bt
	OPA	÷	Configure ADC VDDA		

Figure 4-8. Generate Peripherals and Pin Assignments File

4.3 Schematic and PCB Attentions

The minimum requirements (power, reset, and Vcore) with suggested values for MSPM0 hardware setup are shown in Figure 4-9.

- Power pin: TI recommends adding 10uF and 0.1uF capacitors, which are used to remove AC noise on the power rail.
- Reset pin: TI recommends adding a 47kR pullup resistor and a 10nF pulldown resistor. This makes sure that the MSPM0 releases from reset, after the power rail is stabilized.
- Vcore pin: This pin is used to stabilize the CPU voltage. For some MSPM0 devices, this pin is not included. If the pin is included, connect the pin to a 0.47uF capacitor.



MSPM0L Typical Application Schematic

Figure 4-9. MSPM0 Minimum System

Other considerations when drawing a schematic file are listed in Figure 4-10.

- ROSC Pin: If users want to reach accurate high frequency clock with internal SYSOSC, then 0.1% resistor is suggested. Some low-cost devices cannot have this function.
- VREF+/VREF- Pin:



- If using an internal reference, then the G series require a 1uF capacitor between VREF+ and VREF- to support 4Msps ADC. For L or C series, then the capacitor is not required as the ADC speed is only support 200Ksps with internal Vref.
- If using an external reference, then all the MSPM0 devices require a 1uF capacitor between VREF+ and VREF-.
- **Open-Drain IO:** Open-Drain IO cannot output high voltage from the MCU side, so external pullup resistors are required, such as a 4.7kR capacitor.
- **Reset Pin:** If reusing the reset pin as GPIO, I2C or UART, then the pullup resistor and the pulldown capacitor are still required. This makes sure that the MCU is released from reset state after the power is stable. TI recommends reducing the resistor and capacitor, such as using a 2.2kR pullup resistor and 10pF pulldown capacitor.
- **PA18**: PA18 is the invoke pin to enter bootloader. Make sure this pin is not in pullup or affected by noise or analog signals with this pin floating. Otherwise, the device enters the bootloader instead of the application code. More details and a software option to change and disable the invoke pin in sysconfig are shown in Section 7.4.



Figure 4-10. MSPM0 Schematic

For further information about schematics or PCB design references, see the following links.

- MSPM0 L-Series MCUs Hardware Development Guide
- MSPM0 G-Series MCUs Hardware Development Guide
- Device-specific MSPM0 Launchpad EVM user's guide
- Device-specific MSPM0 data sheet



5 Mass Production Instructions

An overview of the program software and tools is shown in Figure 5-1. The available interface is JTAG (SWD) and Bootloader (BSL). J-Link and C-GANG only support SWD. XDS110 supports SWD and Bootloader over UART.

J-Link and XDS110 can only program one MSPM0 at a time. C-GANG can program six MSPM0s at one time.



Figure 5-1. Program Software and Tools

For more implementation about bootloader, see MSPM0 Bootloader (BSL) Implementation. For more production programming tools, see E2E page.

5.1 Generate Production Image

Table 5-1 lists different types of image generated by different IDEs. For the step by step generation guidance, see Section 3.4.

IDE	TI_TXT (.txt)	Intel hex (.hex)	bin (.bin)	Step by Step Guidance
CCS	Y	Y	Y	Link
IAR	Y	Y	Y	Link
Keil	N	Y	N	Link

Table 5-1. Product File Generated by IDE



5.2 Program Software Tools Quick Start

5.2.1 Uniflash Quick Start

This section describes how to install the UniFlash tool with TI's MSPM0 devices. See the UniFlash Quick Start Guide for more information.

5.2.1.1 Program Through SWD

The debugging interface such as XDS110 can be used by UniFlash to program the device. The needed hardware pins are SWDIO, SWCLK, 3V3 and GND. Follow the steps below:

- 1. Follow the steps to select the debugger (either XDS110 or J-Link). Then click Start to start program.
- 2. If NONMAIN must change, change the erase setting before programming. If this is not required, keep the default option.
- 3. Select the image and start to program by clicking *Load Image*.
- 4. Using the *Memory* tab, UniFlash can also inspect the flash memory of the device simply by selecting *Read Target Device*.

Category: A					
	Q MSPM0L1306	51 ×	UniFlash Session	- About	•
	LP-MSPM0L1306	LaunchPad On-Chip	Configured Device : Texas Instr	ruments XDS110 USB Debug Probe > MSPM0L1306 do	vnload cexml]
	MSPM0L1306	On-Chip		• C(RTEX MOP Disconnected:
	MSPM0L1306(BOOTLOADER)	Serial	Deserver	Colort and Lord Images	
			~ X	Select and Load Images	
	Selected Connection:	ents XDS110 USB Debug Probe	Settings & Utilities	Flash Image(s)	
	•		Momony	PWM_Output_LP_MSPM0L1306_nortos_t	iclang.txt/1D5: e0fd3c9556
0.5	A		Memory		
C Enter	r Connection Name (12 Available)	×	Standalone Command Line		
	SEGGER J-Link Emul	lator J-LINK		Available Action(s) - 1 Intege Selected	
	Spectrum Digital XDS560V2 ST	M LAN Emulator		Load Image Verify Image	
	Spectrum Digital XDS560V2 STM T	RAVELER Emulator			
	Spectrum Digital XDS560V2 ST	M USB Emulator		Reset Actions	
	Spedtrum Digital XDSPRO LA	AN Emulator		[Click here to query availabletti	1
	Spectrum Digital XDSPRO U	SB Emulator		LUNCK HERE tO QUERY available reset options	5]
	Texas Instruments XDS100v2 U	SB Debug Probe		4	
	Texas Instruments XDS100v3 U	SB Debug Probe			
	Texas Instruments XDS110 US	8 Debug Probe XDS110	Console		• <i>=</i>
	Texas Instruments XDS2xx LA	N Debug Probe	[7/1/2024, 9:47:48 AM] [INFO] C	ORTEX_M0P: GEL Output: Memory Map Initialization Con	plete
	Texas Instruments XDS2xx USI	B Debug Probe	[7/1/2024, 9:47:50 AM] [SUCCES	SS] Program Load completed successfully.	
	Texas Instruments XDS2xx USB On	iboard Debug Probe	[7/1/2024, 9:48:19 AM] [INFO] C [7/1/2024, 9:48:19 AM] [SUCCES C:/Users/a0224173/Desktop/PV	:ORTEX_MOP: GEL Output: Memory Map Initialization Con SS] CORTEX_MOP: Program verification successful for WM_Output_LP_MSPM0L1306_nortos_ticlang.txt	spiete
UniFlash	Texas Instruments XDS2xx USB Onl	2 - · ×	[7/1/2024, 9:48:19 AM] [INFO] C [7/1/2024, 9:48:19 AM] [SUCCES C./Users/a0224173/Desktop/PV	ORTEX_MOP: GEL Output: Memory Map Initialization Con SECORTEX_MOP: Program verification successful for WM_Output_LP_MSPMOL1306_nortos_ticlang.txt	piete
UniFlash	Texas instruments XDS2xx USB On Start Edit Step: Session - About	2 - • ×	[7/1/2024, 948:19 AM] [NFO]O C [7/1/2024, 948:19 AM] [SUCCES C:/Users/a0224173/Desktop/PV	ORTEX_MOP: GEL Output: Memory Map Initialization Con SQ CONTEX_MOP: Program verification accessful for VM_Ourput_LP_MSPM0L1306_nortos_ticlang.txt	ipiete
UniFlash	Texas instruments XDS2xx USB On Texas instruments XDS2xx USB On Step; Session - About	2 - · · ×	[7/1/2024, 948:19 AM] [INFO] C [7/1/2024, 948:19 AM] [SuCCES C:/Users/a0224173/Desktop/PV	ORTEX_MOR ⁻ GEL Output: Memory Map Intilization Con S0 CORTEX_MOR ⁻ Program verification uscensiful for WM_Output_LP_MSPM0L1306_nortos_ticlang.txt	- E
UniFlash IniFlash	Texas Instruments XDS2cc USB On Edit Stepp: Session - About Texas Instruments XDS110 USB Debug Probe >	2 - □ × MSPMOL 1306 (download coxml) + CORTEX, MOP	[7/1/2024, 948:19 AM] [INFO] C [7/1/2024, 948:19 AM] [SUCCES C:/Users/a0224173/Desktop/PV	ORTEX_MOR ⁻ GEL Output: Memory Map Intilization Con SQ CORTEX_MOR ⁻ Program verification uscensiful for WM_Output_LP_MSPM0L1306_nortos_ticlang.txt Step4	- E
UniFlash	Texas Instruments XDS2xx USB On	2 - □ × MSPM0L1306 (formioad count) + CORTEX_MOP ngs and Utilities	[7/1/2024, 948:19 AM] [NF0]0 C [7/1/2024, 948:19 AM] [SUCCES C:/Users/a0224173/Desktop/PV C:/Users/a0224173/Desktop/PV C UniFlash Session	ORTEX_MOR [®] GEL Output: Memory Map Initialization Con SQCORTEX_MOR [®] Program verification accessful for VM_Output_LP_MSPM0L1306_nortos_ticlang.txt Step4 • About	- C
UniFlash JniFlash onfigured Device : Program Settings & Utilities	Texas Instruments XDS2xx USB On Stepp Session - About Texas Instruments XDS110 USB Debug Probe > Find and Configure Settil s Q Enter Property ID Or Na	2 - C X X X X X X X X X X X X X X X X X X	T/1/2024, 948:19 AMI [INFO] C T/1/2024, 948:19 AMI [ISUCCES C:/Users/a0224173/Desktop/PV UniFlash UniFlash Session Configured Device : Texas Instit	ORTEX_MOR ⁻ GEL Output: Memory Map Intuilization Con SQ CONTEX_MOR ⁻ Program verification accessful for WM_Ourput_LP_MSPM0L1306_nortos_ticlang.txt Step4 • About ruments X05110 USB Debug Probe > MSPM0L1306 [do	- E wnload ccxml[DRTEX_M0P Disconnected:
UniFlash IniFlash IniFlash rogram ettings & Utilities femory	Texas Instruments XDS2xe USB On	2 - C X MSPMOL 1306 [download coxm] = CORTEX_MOP ngs and Utilities me To Search For Settings and Buttons X III © ry only	T/1/2024, 948:19 AM[INF0]C T/1/2024, 948:19 AM[ISUCCE C:/Users/a0224173/Desktop/PV UniFlash UniFlash Configured Device : Texas Instr Program	ORTEX_MOR [®] GEL Output: Memory Map Initialization Con SQ CORTEX_MOR [®] program verification accessful for VM_Output_LP_MSPM0L1306_nortos_ticlang.txt Step4 About ruments XDS110 USB Debug Probe > MSPM0L1306 [do © Cl Memory Browser	- E whiload coxmil RTEX_MOP Disconnected:
UniFlash niFlash rogram ettings & Utilities lemory tandalone Comm	Texas Instruments XDS2xx USB On Stepp: Session - About Texas Instruments XDS110 USB Debug Probe > Find and Configure Settli s Enter Property ID Or Na Enter Property ID Or Na	2 - □ × MSPM0L1306 [download com!] = CORTEX_MOP ngs and Utilities me To Search For Settings and Buttons × III ● OMMAIN memory (see warning above) OMMAIN recessary sectors only (see warning	[7/1/2024, 948:19 AM] [INFO] C [7/1/2024, 948:19 AM] [INFO] C [7/1/2024, 948:19 AM] [ISUCES C:/Users/a0224173/Desktop/PV UniFlash UniFlash UniFlash Configured Device : Texas Inst Program Settings & Utilities	ORTEX_MOR [®] GEL Output: Memory Map Initialization Con SQUORTEX_MOR [®] Program verification accessful for WM_Ourput_LP_MSPM0L1306_nortos_ticlang.txt Step4 About Constraints XDS110 USB Debug Probe > MSPM0L1306 [do Constraints XDS110 USB Debug Probe > MSPM0L1306 [do Constraints XDS110 USB Debug Probe > MSPM0L1306 [do Constraints XDS110 USB Debug Probe > MSPM0L1306 [do	- C wnload coxmi) RTEX_MOP Disconnected:
UniFlash niFlash rogram ettings & Utilities lemory tandalone Comm	Texas Instruments XDS2xx USB On	2 - CARTEX_MOP MSPMOL 1306 [download comm] * CORTEX_MOP ngs and Utilities re To Search For Settings and Buttons × III ry only OMMAIN memory (see warning above) OMMAIN memory (see warning sove)	[7/1/2024, 948:19 AM] [INFO] C [7/1/2024, 948:19 AM] [Succes [7/1/2024, 948:19 AM] [Succes C:/Users/a0224173/Deaktop/PV [1] [2] [2] [3] [4] UniFlash Session Configured Device : Texas Insti Program Settings & Utilities Memory	ORTEX_MOP: GEL Output: Memory Map Initialization Con SQUORTEX_MOP: Program verification accessful for WM_Output_LP_MSPM0L1306_nortos_ticlang.txt Step4 About Memory Browser Address: 000000 00 Co	- C whited coxmil RTEX_MOP Disconnected:
IniFlash niFlash orgram ettings & Utilities lemory tandelone Comm	Texas Instruments XDS2xc USB On Start Edit Step: Session - About Texas Instruments XDS110 USB Debug Probe > Find and Configure Setti s Crase MAIN memo © Erase MAIN memo © Erase MAIN memo	abard Debug Priose	T/1/2224, 948:19 AM [INFO] C T/1/2224, 948:19 AM [ISUCCE C:/Users/a0224173/Desktop/PV UniFlash UniFlash Configured Device : Texas Instit Program Settings & Utilities Memory	ORTEX_MOR ⁺ GEL Output: Memory Map Initialization Con SQUORTEX_MOR ⁺ Program verification accessful for WM_Output_LP_MSPM0L1306_nortios_ticlang.txt Step4 About runnents XD3110 USB Debug Probe > MSPM0L1306 [do CO Address: COCCOC Co Co Co Co Co Co CO CO	wnload coxml] RRTEX_MOP Disconnected:
UniFlash IniFlash anfigured Device : Program ettings & Utilities Aemory tandalone Comm	Texas Instruments XDS2xx USB On Start Edit Step: Session - About Texas Instruments XDS110 USB Debug Probe > Find and Configure Settli s Q Enter Property ID Or Na Prince MAIN memo C Erase MAIN and N above) C Erase MAIN and N D Erase MAIN and N D On terase Flash	2.	[7/1/2024, 948:19 AM] [NixOIC [7/1/2024, 948:19 AM] [Succes [8/1/1/2024, 948:19 AM] [Succes S	ORTEX_MOR ⁻ GEL Output: Memory Map Intalization Con SQUORTEX_MOR ⁻ Program verification accessful for VM_Output_LP_MSPM0L1306_nortios_ticlang.txt Step4 • About ruments XDS110 USB Debug Probe > MSPM0L1306 [do e Cl Memory Browser Address: 000000 Ge 0.00 2722727 2722727 27272727 27	- C wnload coxmi] RFTEX_MOP Disconnected:
UniFash IniFlash Yrogram Hettings & Utilities Remory Handalone Comm	Texas Instruments XDS2xe USB On Stepp Session - About Texas Instruments XDS110 USB Debug Probe > Find and Configure Settil s Q Enter Property 10 Or Na Perses MAIN memo © Erase MAIN memo © Erase MAIN memo © Do not erase Flash	2 - C X MSPMOLI 306 [download coxml] = CORTEX_MOP ngs and Utilities and Differentiation and Differentiat	[7/1/2024, 948:19 AM] [INFO] C [7/1/2024, 948:19 AM] [Succes [7/1/2024, 948:19 AM] [Succes C:/Users/a0224173/Deaktop/PV Image: Standard State Sta	ORTEX.MOR [®] GEL Output: Memory Map Initialization Con SQUORTEX.MOP Program verification accessful for WM_Output_LP_MSPM0L1306_nortex_ticlang.txt Step4 About Memory Browser Address: 000000 G9 00 20 20 20 20 20 20 20 20 2	whited coxmil RTEX_MOP Disconnected:
UniFlash IniFlash anfigured Device : rogram ettings & Utilities femory tandalone Comn	Texas Instruments XDS2xc USB On Session - About Texas Instruments XDS110 USB Debug Probe > Find and Configure Settis a. Enter Property ID Or Man C Erase MAIN memo C Erase MAIN memo D Do not erase Flash Note: Sector Erase: all	abard Debug Priose 2	[7/1/2024, 948:19 AM] [NixOIC [7/1/2024, 948:19 AM] [Succes [7/1/2024, 948:19 AM] [Succes C:/Users/a0224173/Desktop/PV ImiFlash UniFlash Configured Device : Texas Instr Program Settings & Utilities Memory Standalone Command Line	ORTEX.MOR ⁻ GEL Output: Memory Map Initialization Con SQUORTEX.MOR ⁻ Program verification accessful for WM_Output_LP_MSPM0L1306_nortios_ticlang.txt Step4 About ruments XD3110 USB Debug Probe > MSPM0L1306 [do CO Memory Browser Address: COCC Co CO CO CO CO CO CO CO	wnload coxml] MrtEx_MOP Disconnected: MrtEx_MOP Disconnected: MrtEx_MOP Disconnected: Drawn mrtex Mrtex Mrtex Mrtex Mrtex Mrtex Mrtex Mrte
iniFlash niFlash infigured Device : rogram ttings & Utilitier lemory tandalone Comm	Texas Instruments XDS2xx USB On Step: Step: Session - About Texas Instruments XDS110 USB Debug Probe > Find and Configure Setti s Q Enter Property ID Or Na Proceedings of the Setting of the Setting Carase MAIN memo D Enter MAIN and N above) Carase MAIN memo D on to rease Flash Note: Sector Erase: all be erased	Assectors between Start and End address will	[7/1/2024, 948:19 AM] [NixOIC 5 [7/1/2024, 948:19 AM] [Succes [7/1/2024, 948:19 AM] [Succes C:/Users/a0224173/Desktop/PV UniFlash UniFlash Configured Device : Texas Instr Program Settings & Utilities Memory Standalone Command Line	Context. Monor GEL Output: Memory Map Initialization Con Store Store St	wnload coxmi] RTEX_MOP Disconnected: RTEX_MOP Disconnected: Device rmm rmm Device rmm rmm
UniFash IniFlash Yogram etings & Utilities temory tandalone Comn	Texas Instruments XDS2xx USB On Stepp: Session - About Texas Instruments XDS110 USB Debug Probe > Find and Configure Setti s Q Enter Property 10 Gr Na Persee MAIN memo © Erase MAIN memo © Do not erase Flash Note: Sector Erase: all be erased	2	[7/1/2024, 948:19 AM] [INFO] O [7/1/2024, 948:19 AM] [INFO] [INFO] [INFO] [INFO] [INFO] [INFO] [ORTEX_MOR ⁻ GEL Output: Memory Map Initialization Cons Storms, Mory Program verification accessful for VM_Output_LP_MSPM0L1306_nortos_ticlang.txt Step4 About Memory Browser Address: 0000 0x0 7777777 0x0 7777777 0x0 7777777	wnload.cxml] DRTEX_MOP Disconnected: TTTTTT Dravice TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
JniFlash niFlash nifgured Device : rogram ettlings & Utilities iemory tandalone Comm	Texas Instruments XDS2xc USB On Session - About Texas Instruments XDS110 USB Debug Probe > Find and Configure Settli a Enter Property ID Or Na Case MAIN memo Case MAIN memo Do not erase Flash Note: Sector Erase Start Addre Sector Erase Start Addre	abard Debug Probe	[7/1/2024, 948:19 AM] [NixOIC [7/1/2024, 948:19 AM] [Success C:/Users/a0224173/Desktop/PV [*] UniFlash UniFlash Configured Device : Texas Institution Program Settings & Utilities Memory Standalone Command Line	ORTEX_MOR* GEL Output: Memory Map Initialization Con Step4 Step4 About ruments XDB110 USB Debug Probe > MSPM0L1306 [do 00 Memory Browser Address: 000 Oco Trrrrrrr Trrrrrrrrrrrrrrrrrrrr	wnload coxml] MrtEx_MOP Disconnected: MrtEx_MOP Disconnected: MrtTx Transmit Transmit Device Transmit Transmit Transmit Transmit Transmit Transmit Transmit Transmit

Figure 5-2. Program Through SWD

5.2.1.2 Program Through Bootloader

Here are the steps to program MSPM0 through bootloader using Uniflash. The required hardware pins are TX, RX, 3V3, GND and invoke pins.

- 1. Search the device name and select the bootloader option for the device.
- 2. Check the COM port by referring to the device manager.
- 3. Check the UART Bootloader port by referring to the data sheet.
- 4. Finish the hardware connection (RX, TX, 3V3, GND, Invoke) and start program.



Step1



	Step2			
UniFlash	-	-		\times
	UniFlash			=
Configured Device : Serial Connec	tion > MSPM0L1306(BOOTLOADER) [download ccxml]	• C	ORTEX	_M0P
Program	Find and Configure Settings and Utilities			
Settings & Utilities	Q Enter Property ID Or Name To Search For Settings	s an ×	:=	e
Standalone Command Line	▼ Setup			
	/dehtty.usbmodem1411 (OS X)	io (Linux),		
Device Manager File Action View Help	×			
 WAN Miniport (SSTP) WOther devices Ports (COM & LPT) Intel(R) Active Management 	Communication Support is ava ridge for UART is only support	ilable. ed by XDS	6	

Step4

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

Ston2



Figure 5-3. Program Through Bootloader

5.2.1.3 Program Through CMD Line Interface

For this requirement, see this E2E thread.

5.2.2 JFlash Quick Start

This instruction is based on J-Flash V7.92n. TI recommends using the latest J-Flash version, which supports all the latest versions of MSPM0. Use the following steps to program MSPM0 with J-Flash:

- 1. Click New project.
- 2. Select the related MSPM0 part number.
- 3. Select the desired programming memory. If NONMAIN does not need to change, deselect NONMAIN memory.
- 4. Click Connect device and click Production Programming.
- 5. A confirmation screen appears.





Figure 5-4. J-Flash Quick Start



5.2.3 C-GANG Quick Start

This section shows how to use C-GANG to do the MSPM0 online and offline program. For more advanced usage, like setting password or Factory Reset, refer to the user's guide in the C-GANG product page and TI-CGANG-MSPM0 video.

1. Please follow the steps bellow to finish GangPro-ARM GUI installation and USB driver installation.

Step1		🔁 Elprotro	nic - FP-ARM / GP-ARM (vStera)	4) — 🗆	×
FlashPro-ARM and GangPro-ARM - Version 2.92 (3.March.2025)	License	e Agreement	ſ	
Installation package for Windows with GUI, DLL, and example applications with source code. 32-bit and 64-bit builds.		Please take Agree", the	e a moment to read the license agreement now. If you act n "Next". Otherwise click "Cancel".	cept the terms below, click	- *** . T
Linux shared library and examples with source code. Ubuntu amd64 build. Requires OpenSSL 3.0+		PLEAS THE S ELPRO ("ELP TO YO	SE READ THIS DOCUMENT CAREFUL SOFTWARE AND THE ASSOCIATED H OTRONIC INC. AND/OR ITS SUBSIDIA ROTRONIC'') IS WILLING TO LICENSI DU AS AN INDIVIDUAL, THE COMPAN	LY BEFORE USIN [ARDWARE. RIES 5 THE SOFTWARI VY, OR LEGAL	₩G ∥ E
		ENTIT (REFE CONE LICEN	TY THAT WILL BE USING THE SOFTW CRENCED BELOW AS "YOU" OR "YOU DITION THAT YOU AGREE TO ALL TE USE AGREEMENT. THIS IS A LEGAL A lot Agree	'ARE JR") ONLY ON TH RMS OF THIS IND ENFORCABLI	IE E
			Cancel	< Back Next:	>
Elprotronic - FP-ARM / GP-ARM (valation of the April 10 CUs (x64) ×		🔂 Elprotro	nic - FP-ARM / GP-ARM (v <mark>Steap14</mark> CUs (x6	4) — 🗆	×
Installation options		Installa	tion Complete		
✓ Create FlashPro-ARM icon on the desktop		Elprotronic Click "Close	- FP-ARM / GP-ARM (v2.91) for ARM MCUs (x64) has ber 9" to exit	en successfully installed.	
Install USB Drivers (Unplug Programmer from PC first)		Please use	Windows Update to check for any critical updates to the	.NET Framework.	
Cancel < Back Next>)		Cancel	< Back Close	e
Step5			Step6		
Description		Download Link	 Name A long time ago 	Date modified	
USB drivers for Elprotronic products: XS, X2S, SC-GANG, and CMSIS-DAP 2.0, with support Windows 11 (24H2) where applicable (11.0ct.2024)	up to		影 XS-DriverUninstaller.exe 区 影 XS-DriverInstaller.exe WIN-64	10/11/2024 1:30 10/11/2024 1:30 10/11/2024 1:3	0 PM 0 PM 1 PM
 			WIN-32	10/11/2024 1:3	1 PM

Figure 5-5. GangPro-ARM Install

2. Finish connection between C-GANG and the connector board, as shown in Figure 5-6. Finish the pin connection between MSPM0 and C-GANG. The least used pins are VCC, GND, SWDIO, SWCLK. If users want to use "Clear Locked Device" function, reset pin is also needed.







3. After the hardware setup is finished, follow the programming steps. If users open the GUI, then users can go through step 2 to scan the C-GANG. In step 3, see Section 5.1 to generate the code file. Remember the enabled target is related to the hardware port used, which is labeled numerically next to the port.



		Step2	
Communication via LISB [all adapters	5]		C Communication via Ethernet/LAN
Select the serial number of the desired F	PA to connect to this software.		Search for XStreamPro Adapters on
FPA list (1 8)	<u>^</u>	Successfully test connection to modify internal settings	Interface: 10.85.14.x (PC: 10.8
	C-GANG-TI		
c		BC-26-43 xxxxx xx	
C		C Custom MAC Address 00 00 00 00 00 00	
c		IP Address Assignment -> C DHCP C Fixed IP	
		Use the following IP address:	
	v	IP address: 0.0.0.0	
	Refresh USB Device List	Subnet mask: 0.0.0.0	
C Communication via UART [S-GANG c	nlv 1	Default gateway: 0.0.0.0	
Select the COM port of the desired FPA	to connect to this software.	Save to XStreamPro Reload from XStreamPro	
COM Port List	David Daka	Connect to XStreamPro Adams using these Settings (GUT/DLL)	-
	38400 ¥	LAN Connection -> C First found C IP C MAC C SN	<u> </u>
I TOTAL		Destination IP/MAC/SN	<<
	Refresh COM Port List		
[Label Check	Software will use this Adapter	Test Connection
		Dage Interface Type=3, Access	Key=1 Done
		HW PN=C-GANG-1.0 Adapter Desc=C-GANG	Done
		Accept	
GangPro-ARM (Gang Programmer for ARM MCUs) View Setup Serialization Measurement Sta Open Code File C:\User.\adc12_14bit_resolution_LP;	- Elprotronic Inc. (version 2.91, x6 andalone About/Help _MSPM0C1104_nortos_ticlang.txt	Append Power ON/DFF RESET	Interface: SWD Fast(2w)
GangPro-ARM (Gang Programmer for ARM MCUs) le View Setup Serialization Measurement St. Open Code File C:\User.Nadc12_14bit_resolution_LP SN Input File Microcontroller Type Vendor: T1-MSP Family: MSPM0 Cortex M0 Group: MSPM0C Cortex M0 Group: MSPM0C Cortex M0 Group: MSPM0C Cortex M0 Group: MSPM0C Cortex M0 Total Balance: 0 Target: MSPM0C1104 EFlash: No ExtFlash Support Selected Device Information RAM - 1.0 kB; FLASH - 16.0 kB Seport Erasing memory	- Elprotronic Inc. (version 2.91, x6 andalone About/Help MSPM0C1104_nortoe_ticlang.bt 	Append Power ON/DFF RESET Append Power ON/DFF RESET Target Enable 1 2 3 4 5 6 Connect 2 3 4 5 6 Connect 2 3 4 5 6 Program./Verify 2 0 0 0 0 Protection 2 0 0 0 Final Result 2 0 0 Since MCU Memory 1 0 0x01BC4892 2 0 3 0 Since Result 2 0 Since Result 2 0 Since Result 2 0 Protection 2 0 Since Result 2 0 <	Interface: SWD Fast[2w] Speed 500 For 8 Flash only ** Flas
GangPro-ARM (Gang Programmer for ARM MCUs) le View Setup Serialization Measurement St. Open Code File C:\User.\adc12_14bit_resolution_LP SN Input File Microcornolier Type Vendor: T1-MSP Family: MSPM0 Cortex M0 Group: MSPM0C Series Name: MSPM0C1104 Target: MSPM0C1104 EFlash: No ExtFlash Support Selected Device Information RAM - 1.0 kB; FLASH - 16.0 kB Report Erasing memory	- Elprotronic Inc. (version 2.91, x6 andalone About/Help MSPM0C1104_nortos_ticlang.bt. MCI Vcc 3.3V F Enable 0.00V Checksum from File and Optic Source: 0x01BC4892 Code: 0x01BC4892 Code: 0x01BC4892	Append Power ON/DFF RESET Target Devices Programming Result Target Enable 1 2 3 4 5 6 Connect 2 1 2 5 5 6 5 7 5 5 6 Connect 2 1 2 5 5 5 6 5 7 5 5 5 5 5 5 5 5 5 5 5 5 5 5	Interface: SWD Fast[2w] Speed 500 7 8 Memory Protect 7 8 Memory Protect Clear Lock Devi Clear
GangPro-ARM (Gang Programmer for ARM MCUs) le View Setup Serialization Measurement St. Open Code File C:\User.Nadc12_14bit_resolution_LP SN Input File Status Vendor: T1-MSP Family: MSPM0 Cortex M0 Group: MSPM0C Series Total: Balance: 0 Total: Balance: 0 Target: MSPM0C1104 EFlash: No ExtFlash Support / Add/Edt Selected Device Information RAM - 1.0 kB; FLASH - 16.0 kB teport Trasing memory	- Elprotronic Inc. (version 2.91, x6 andalone About/Help MSPM0C1104_nortos_ticlang.bt 	Append Power ON/OFF RESET Target Devices Programming Result Target Enable </td <td>Interface: SWD Fast(2w) Speed 500 F 7 8 Flash only *** Clear Lock Devic Clear Locked D Device Action Feload Code Enable Blank 4 B ERASE FLA* UVERIPY ACCE ERASE FLA*</td>	Interface: SWD Fast(2w) Speed 500 F 7 8 Flash only *** Clear Lock Devic Clear Locked D Device Action Feload Code Enable Blank 4 B ERASE FLA* UVERIPY ACCE ERASE FLA*
GangPro-ARM (Gang Programmer for ARM MCUs) e View Setup Serialization Measurement Str Open Code File C:\User.\adc12_14bit_resolution_LP SN Input File Wiccourandler Type Vendor: TI-MSP Family: MSPM0 Cortex M0 Group: MSPM0C Cortex M0 Total: Balance: 0 MSPM0C1104 Target: MSPM0C1104 EFlash: No ExtFlash Support Selected Device Information RAM - 1.0.kB; FLASH - 16.0.kB Report Traing memory	- Elprotronic Inc. (version 2.91, x6 andalone About/Help 	Append Power DN/DFF RESET Target Devices Programming Result Target Enable </td <td>Interface: SWD Fast(2w) Speed 500 T 7 8 Wenoy Protect Fash only *** Lock Devic Clear Locked Reload Code Fasble Blank UTO PROG VERIFY ACCI ERASE FLA UVRITE EN /N</td>	Interface: SWD Fast(2w) Speed 500 T 7 8 Wenoy Protect Fash only *** Lock Devic Clear Locked Reload Code Fasble Blank UTO PROG VERIFY ACCI ERASE FLA UVRITE EN /N
GangPro-ARM (Gang Programmer for ARM MCUs) le View Setup Serialization Measurement St Open Code File C:\User.\adc12_14bit_resolution_LP SN Input File Status Wendor: T1-MSP Status Family: MSPM0 Cortex M0 Status Group: MSPM0C Series Status Total: Balance: 0 MSPM0C1104 Status Balance: 0 MSPM0C1104 Add/Edt Selected Device Information RAM - 1.0.kB; FLASH - 16.0.kB Report Erasing memoy	- Elprotronic Inc. (version 2.91, x6 andalone About/Help 	Append Power ON/OFF RESET Target Devices Programming Result Target Enable 1 1 2 3 4 5 6 Connect Imaget Devices Programming Result Imaget Devices Programming Result Target Enable 1 1 2 3 4 5 6 Connect Imaget Devices Programming Result Target Enable 1 1 2 3 4 5 6 Connect Imaget Devices Programming Result Program/Verify Imaget Devices Programming Result Program/Verify Imaget Devices Programming Result Final Result Imaget Devices Programming Result Imaget Devices Programming Result Imaget Devices Programming Result Imaget Devices Programmer Imaget Devices Programmer Imaget Devices Programmer Imaget Devices Programer <t< td=""><td>Interface: SWD Fast(2w) Speed 500 T 8 Flash only " Flash</td></t<>	Interface: SWD Fast(2w) Speed 500 T 8 Flash only " Flash
GangPro-ARM (Gang Programmer for ARM MCUs) le View Setup Serialization Measurement Str Open Code File [C:\User.\add12_14bit_resolution_LP SN Input File [Input File] Microcorrectile=Type Status Family: MSPM0 Cortex M0 [Input File] Family: MSPM0 Cortex M0 [Input File] Name: MSPM0C1104 [Input File] Name: MSPM0C1104 [Input File] Family: No ExiFlash Support [Input File] Falence: 0 Selected Device Information RAM - 1.0 kB; FLASH - 16.0 kB Report [Input File] Unlocking target device	- Elprotronic Inc. (version 2.91, x6 andalone About/Help 	Append Power ON/OFF RESET Target Devices Programming Result Image Devices Programming Result Image Devices Programming Result Target Enable Image Devices Programming Result Image Devices Programming Result Image Devices Programming Result Target Enable Image Devices Programming Result Image Devices Programming Result Image Devices Programming Result Final Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result Image Devices Programmed Result	Interface: SWD Fast(2w) Speed 500 T 8 Fast ony *** Lock Devic Lock Devic Cere Locked E Faste Blank LUTO PROG VERIPY ACCE ERASE FLAS UNTO FROM VERIPY ACCE ERASE FLAS UNTO FROM VERIPY FLAS
GangPro-ARM (Gang Programmer for ARM MCUs) e View Setup Serialization Measurement Str Open Code File C:\User.\adc12_14bit_resolution_LP SN Input File Status Wendor: TI-MSP Status Family: MSPM0 Cotex M0 Totak Group: MSPM0C Series Balance: 0 Totak Balance: 0 Totak Balance: 0 Totak Balance: 0 Totak Balance: 0 C: User.NackStatus Family: MSPM0C1104 EFlash: No ExtFlash Support Add/Edl Selected Device Information RAM - 1.0 kB; FLASH - 16.0 kB Report Fash programming. H1: SD-24.01 MHz, Sys.CLK-24.03 MHz Tasht rogramming. H1: SD-24.01 MHz, Sys.CLK-24.03 MHz Tasht programming. H1: SD-24.01 MHz H1: SD-24.01 MHz H	- Elprotronic Inc. (version 2.91, x6 andalone About/Help 	Append Power ON/DFF RESET Target Devices Programming Result Image Devices Programming Result Image Devices Programming Result Target Enable Image Devices Programming Result Image Devices Programming Result Image Devices Programming Result Target Enable Image Devices Programming Result Image Devices Programming Result Image Devices Programming Result Target Enable Image Devices Programming Result Image Devices Programming Result Image Devices Programming Result Program/Verify Image Devices Programming Result Image Devices Programming Result Image Devices Programming Result Target Enable Image Devices Programming Result Image Devices Programming Result Image Devices Programming Result Target Devices Programm/Serify Image Devices Programming Result Image Devices Programming Result Image Devices Programming Result Trans Checksum from MCU Memory Image Devices Programming Result Image Devices Programming Result Image Devices Programming Result Target Devices Programme Image Devices Programme Image Devices Programme Image Devices Programme Target Devices Programme Image Devices Programme Image Devices Programme Image Devices	Interface: SWD Fast(2w) Speed 500 T 7 8 Hennoy Protect Fasth only mile Clear Locked Device Action Reload Code Enable Blank AUTO PROG VERIFY ACC ERASE FLA BLANK CHE WRITE SN // VERIFY FLA READ / CO
GangPro-ARM (Gang Programmer for ARM MCUs) e View Setup Serialization Measurement Str Open Code File C:\User.\adc12_14bit_resolution_LP SN Input File Status Vendor: TI-MSP Status Family: MSPM0 Cortex M0 Total Balance: 0 Total Balance: 0 Total Balance: 0 Total Balance: 0 Total Balance: 0 Total Balance: 0 Total Balance: 0 Total Balance: 0 Total Balance: 0 Total Balance: 0 Communication infibilization RAM - 1.0 kB; FLASH - 16.0 kB Report Ersing memory	- Elprotronic Inc. (version 2.91, x6 andalone About/Help 	4) Step3 Append Power DN/DFF Target Devices Programming Result Target Enable 1 1 2 3 4 5 6 Connect 3 4 5 6 Connect 3 4 5 6 Program/Verify 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Interface: SWD Fast(2w) Speed 500 T 8 T 8 T 8 T 8 T 8 T 8 T 8 T 8 T 8 T 8
GangPro-ARM (Gang Programmer for ARM MCUs) e View Setup Serialization Measurement Str Open Code File C:\User.\adc12_14bit_resolution_LP SN Input File Status Vendor: TI-MSP Status Family: MSPM0 Cortex M0 Total Balance: 0 Total Balance: 0 Communication mitolication RAM - 1.0 kB; FLASH - 16.0 kB Report Communication mitalization	- Elprotronic Inc. (version 2.91, x6 andalone About/Help 	4) Step3 Append Power DN/DFF Target Devices Programming Result Target Enable 1 1 2 3 4 5 6 Connect 3 4 5 6 Connect 3 4 5 6 Program/Verify 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Interface: SWD Fast(2w) Speed 500 F 7 8 Flash ony Protecti F Rash ony F Rash ony Protecti F Rash ony Prote
GangPro-ARM (Gang Programmer for ARM MCUs) View Setup Serialization Measurement Str Open Code File C:\User.\vadc12_14bit_resolution_LP SN Input File Status Wicrocontroller Type Status Wicrocontroller Type Status Wicrocontroller Type Status MisPMOC Series S amily, MSPM0 Cottex M0 Status Index MSPM0C1104 Status Balance: 0 arget: MSPM0C1104 Status Balance: 0 Add/Eddl Selected Device Information RAM - 10 kB; FLASH - 16.0 kB sepot rearing memory	- Elprotronic Inc. (version 2.91, x6 andalone About/Help 	Append Power DN/DFF RESET Target Devices Programming Result Target Enable Target Enable Target Enable Target Enable Torget Enable </td <td>Interface: SWD Fast[2w] Speed 50 F T 8 F T</td>	Interface: SWD Fast[2w] Speed 50 F T 8 F T
GangPro-ARM (Gang Programmer for ARM MCUs) View Setup Serialization Measurement Str Open Code File C:\User.\adc12_14bit_resolution_LP SN Input File Nicoconnoller.Type endor: TI-MSP mily: MSPM0 Cotrex M0 SN SPM0 Cotrex M0 SN SPM0 Cotrex M0 Totat Balance: 0 arget: MSPM0C1104 Tash: No ExiFlash Support Add/Edit Plash: No ExiFlash Support Add/Edit Plash: No ExiFlash Support Salign memory	- Elprotronic Inc. (version 2.91, x6 andalone About/Help MSPM0C1104_nortoe_ticlang.txt MCII Voc 3.3V F Enable 0.00V Del Checksum from File and Optic Source: 0x01BC4892 Code: 0x01BC4892 Code: 0x01BC4892 Code: Target Devi Target Devi Target Devi Target Devi Target Devi Target Devi Target Devi Target Devi Target Devi	Append Power DN/DFF RESET Target Devices Programming Result Target Enable I I 2 3 4 5 6 Connect Since Riser Blank Check I I I 2 3 4 5 6 Enser/Blank Check I I I I 2 3 4 5 6 Final Result I I I I 2 3 4 5 6 Protection I I I I I I I I I I I I I I I I I I I	Interface: SWD Fast(2w) Speed 500 Fast(2w) Fast(2w) Fast(2w) Speed 500 Fast(2w) Fast

Figure 5-7. Online Program

4. To change the code file in the non-main (SWD and BSL configure flash area), click the *Enable* button in the *memory protection* region. If it is not needed, please keep it disabled.





Figure 5-8. Enable Non-Main Programming

5. Save the code file and settings (Image) into C-GANG.





	Step4	
Current Image 01	[MSPM0L1306][229.376 kB, PSA=0x6079.F2F8] Date:2025.02.19-11:18	Erase Image
.CG Counter Image List * 01: Seq- 0 02: Seq- 0 03: Seq- 0 04: Seq- 0 05: Seq- 0 06: Seq- 0 07: Seq- 0 08: Seq- 0 09: Seq- 0 10: Seq- 0 11: Seq- 0 12: Seq- 0 History		Standalone mode Active Image 01 Set Current Image as active in Standalone mode
Create and Save Imag	e to File Setting Passwords OK Image Generating OK Erasing Image OK Writing New Image OK Verifying Image FSA OK	Сancel



MSPM0 MCUs Development Guide

56

6. Set the function for GO button.

Standalone About/Help	Standalone - GO Options	
GO button options Image Setup	E Dun Standalana mada fran CIU	
Change Erase Password	Enable GO button on adapter when GUI is ru	unning
	Standalone mode	
	I▼ Enable	
	Green GO Button Enable	MCUs Vcc
	GO on pin-13 Enable	Vcc min [V] : 1.00 V > 1.0 V
	GO on Vcc Out/Sense (pin-2) Enable	Vcc max [V] : 1.30 V < 4.2 V
	GO on Vcc-Sense (pin-4) Enable	
	C Used Vcc +/- 0.3V range	Settle time [ms]: 50 ms 501000ms
	C Defined Vc; min / Vc; max	
	Save in	Programmer
	-	

Figure 5-10. Go Button Setting

7. Now that the image is downloaded into C-GANG. Users can close the GUI to do the programming by pressing the green button.



Figure 5-11. Offline Downloading





5.3 Program Hardwares Quick Start

Due to J-Link is commonly used and C-GANG hardware is already introduced in Section 5.2.3, this section focuses on the XDS110 debugger. For more production programming tools, see E2E page.

There are four different types of XDS110 debuggers available. The summary table is listed Table 5-2.

		00		
Support Features	XDS110		XDS110 On Board	
	TMDSEMU110-U	MSPM0 LaunchPad	LP-XDS110	LP-XDS110ET
JTAG	Yes	No	Yes	Yes
SBW	Yes	Yes	Yes	Yes
EnergyTrace	Yes	Rely on type	No	Yes
MSPM0 bootloader	Yes	Rely on type	No	No
Comment	Highest Performance	Cheapest	Easy to use	Easy to use

Table 5-2. XDS110 Debugger Summary

With the TMDSEMU110-U device, the pin that is used is shown in Figure 5-12. When using for bootloader, GPIOOUT0 must connect to the MCU reset pin. GPIOOUT1 must connect to the MCU invoke pin (PA18).



Figure 5-12. Pin Connection of TMDSEMU110-U

For XDS110 on LaunchPad, the basic programming functions are intact compared to the TMDSEMU110-U. The board is shown in Figure 5-13. The cheapest XDS110 on LaunchPad is LP-MSPM0C1104. However, LP-MSPM0C1104 only supports SBW and there is no EnergyTrace or bootloader function.



On-board interface

Figure 5-13. XDS110 Onboard



LP-XDS110 and LP-XDS110ET are similar with XDS110 on a LaunchPad. The difference lies on that one has EnergyTrace function and the other does not. The pin assignment is shown in Figure 5-14.

For LP-XDS110 and LP-XDS110ET, the level shift function is enabled by changing the jumper at the left bottom of the board. The support voltage range is from 1.2V to 3.6V.



Figure 5-14. LP-XDS110ET

6 Quality and Reliability Instructions

TI is committed to delivering high quality and reliable semiconductor designs that meet our customers' needs. Our holistic approach to quality permeates every aspect of the company's supply chain from process technology and design through manufacturing, packaging, test and delivery.

6.1 Quality and Reliability Material Entrance

This is the landing page for Quality & reliability. The following are the common used tools and links under that page:

- Qualification summary★: Used to search reliability data of related devices. Representative data summary of the material sets, processes, and manufacturing sites used by the device family.
- Reliability testing: Listed the various types of testing that TI conducts for reliability of the products.
- Customer returns: The Customer Returns page provides detailed guidelines for returning material to TI.
- DPPM/FIT/MTBF estimator: The DPPM/FIT/MTBF estimator search tool allows you to find generic data based on technology groupings to estimate these typical questions and shows conditions under which the rates were derived.
- Ongoing reliability monitoring: The search tool of ongoing reliability monitor (ORM) program provides the quarterly ORM report by wafer fab process or device package family.
- Packaging: This website allows users to find package considerations including package size, SMT recommendations, reliability, and performance expectations.

6.2 Failure Information Collection and Analysis Guidance

Failure analysis needs to collect as much technical background information as possible to narrow down the scope of analysis and accelerate the analysis speed. If users meet any device failure on MSPM0, then collect the information as below, and connect to TI through the Customer returns page or the Regional CQE and Sales supporting your product or business.

Device name (TI Part Number, including package designator):

• Example: MSPM0L1306SRGER

Failure rate (purchased vs. customer failed units):

• Example: Failure rate: 5% (Total tested qty: 2000, Failed qty: 100)

Detection place (field return, production, incoming, and so forth):

· Example: Board level function test

Schematic of the application:

· Example: Schematic of the MCU part, with detailed description to every input and output signals

Detailed device level failure description

• Example: MCU PA1 cannot output high voltage

This is an introduction to the common methods to collected the failure information.

- Method 1: ABA swap test to judge whether the issue is caused from the device or the relativity between the
 device with the total system. Here are the steps to do ABA swap test: Remove the suspected component (A)
 from the original failing board. Replace the suspected component (A) with a known good component (B) and
 check if the original board now works properly. Mount the suspected component (A) to a known good board
 and see if the same failure occurs on the good board.
- Method 2: Compare MCU current consumption with the data sheet under the standby mode. Some device failure is caused from EOS (Electrostatic Overstress), which causes additional leakage current. This can be caught by current consumption test.
- Method 3: Pin impedance check. Some EOS (Electrostatic Overstress) is purely happened at I/Os, and using pin impedance check can easily catch this failure to give more information to TI. Users can choose to detect the IO resistance with or without powering the device. The resistance of a GPIO in high impedance state needs to be MΩ level.



• Method 4: Find a smallest system or code example. Some failure happens with the typical application and typical code project. Through comparison method, removing the unrelated hardware setup and software code step-by-step can gradually narrow down the scope of analysis. The best result is that the problem is purely related to the device and a simplest code example. With that, TI can carry the further failure analysis faster.

7 Common Development Questions

This section lists some common questions for users to search. For further questions, search the device-specific data sheet, technical reference manual, or E2E. TI engineers provide response in 24 hours on this online support platform.



Ask a new question

Log in or create a free myTl account to post a new question and connect with our engineers.

Ask a new question Ask questions

Get quality, packaging or ordering support

For non-design-related questions such as ordering semiconductor parts and tools, contact our customer support center where you can open a support ticket, chat with us 24 hours a day, Monday through Friday; or call the TI support team.

Figure 7-1. E2E Online

7.1 Unlock MCU

MSPM0 can experience SWD connection issues when going into STOP, STANDBY, or SHUTDOWN mode. The effect of this limitation depends on the IDE and debugger implementation. Please use the tools with the latest versions, shown in Table 7-1. For more details, please refer to the Debugging in Low Power Modes chapter in the MSPM0 SDK Known Issues and FAQ.

Table 7-1. Tools Suggested Version

Keil CMSIS Pack	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+

MSPM0 can also lose connection after downloading a wrong code, and CCS reports errors when programming a new code. An example is shown in Figure 7-2.



Figure 7-2. CCS Error



The Debug Subsystem Mailbox (DSSM) enables a debug probe to pass messages to the boot ROM of an MSPM0 device through the SWD interface. There are four unlock commands that you can choose in tools. The brief introduction is in Table 7-2. **DSSM Factory Reset is recommended**, which the reset level is higher than DSSM Mass Erase.

Unlock Commands	Hardware Connection With Debugger	Reset Pin Control	Command Influence		
DSSM Factory Reset Manual	3v3, GND, SWDIO, SWCLK, Reset	End users	Erase main flash and reset		
DSSM Factory Reset Auto		Debugger	NONMAIN flash		
DSSM Mass Erase Manual		End users	Erasa main flash		
DSSM Mass Erase Auto		Debugger			

Table 7-2. Unlock Commands

The suggestion on the provided three unlock methods is shown in Table 7-3. An important note is that the unlock method only supports XDS110 and does not support J-Link currently.

Unlock Method	Support Debugger	When to Choose
Factory Reset GUI Tool	XDS110	Internet connection is available
Uniflash	XDS110	Internet connection is unavailable
CCS	XDS110	Use CCS as the development IDE

Table 7-3. Unlock Method Selection

7.1.1 Unlock Through Bootloader

If users do not touch the NONMAIN memory and come to this problem, then the easiest way is to make the device enters Bootloader mode when the device powers on. Then, reprogram the flash. Follow the steps below:

- 1. Before powering on MSPM0, pull and hold PA18 to be high.
- 2. Program the flash with the right code. Then release PA18.

7.1.2 Unlock Through Factory Reset GUI Tool

The MSPM0 Factory Reset GUI tool is a standalone tool used to gain debug access or recover an MSPM0 device using this interface. This tool is available free of charge. Follow the steps to reset the MSPM0.





Output console

CS_DAP_0: GEL Output: SEC_AP Reconnect CS_DAP_0: GEL Output: Command execution completed. CORTEX_MOP: GEL Output: Factory Reset executed. Please terminate debug session, power-cycle and restart debug session.

DSService deconfigured. Core deattached/closed.

Figure 7-3. Unlock Through GUI



7.1.3 Unlock Through Uniflash

Uniflash above Version: 8.7.0.4818 also supports to unlock MSPM0. First, follow the steps to connect the MSPM0 with Uniflash, as shown in Section 5.2.1.1. Then, follow the instructions to unlock MSPM0 in Figure 7-4.

🗲 UniFlash			-		×		
UniFlash Session -	About			?	٠		
Configured Device : Texas Instrum	nents XDS110 USB Debug Probe > MSPM0L1306 [download ccxml]	CORTEX_M0	P Disconnect	ed: Run	ining		
Program	Find and Configure Settings and Utilities	Find and Configure Settings and Utilities					
Settings & Utilities	Q, Search: Enter Property ID Or Name To Search For Settings	anc× ≔ Mor	e Info 🛛 🕀	Pin O	otion		
Memory	Mass erase manual Mass erase auto				-		
Standalone Command Line					1		
	Factory reset manual Factory reset auto						
	▼ Erase Configuration				_		
Console		I Verbose	≡ Clear	×	Close		
[//1/2024, 4:21:06 PM] [INFO] CS_[[7/1/2024_4:21:06 PM] [INFO] CS_[DAP_U: GEL Output: Board Reset Complete						
[7/1/2024, 4:21:06 PM] [INFO] CS_[DAP_0: GEL Output: SEC_AP Disconnect						
[7/1/2024, 4:21:06 PM] [INFO] CS_[DAP_0: GEL Output: SEC_AP Reconnect						
[7/1/2024, 4:21:06 PM] [INFO] CS_[DAP_0: GEL Output: Command execution completed.				- 1		
[7/1/2024, 4:21:06 PM] [INFO] COR	TEX_M0P: GEL Output: Mass Erase executed. Please terminate debug	session, power-c	ycle and resta	irt debi	g		
session.							

Figure 7-4. Unlock Through Uniflash

7.1.4 Unlock Through CCS

Here are the steps to unlock MSPM0 through CCS:

- 1. In the CCS project, select *targetConfigs* → *MSPM0xxxx.ccxml*. Right-click the .ccxml and select *Start Project-less Debug*.
- 2. Select Scripts \rightarrow MSPM0xxxx_Commands.
- 3. If users choose the manual command, then users need to reset the device manually according to the command in the console. After that, users can repower the device. If users choose auto command, then the debugger resets the device.



Factory Reset executed. Please terminate debug session, power-cycle and restart debug session.

Figure 7-5. Unlock Through CCS



7.2 MSPM0 Program Failure

If the program failure is met for the first time, then check these items one by one:

- 1. Install the latest IDE or programming software tools at the English path. The default install path is suggested. For install instructions, please see the related chapter in this note.
- 2. Plug in the debugger and check whether the debugger is found by the computer. Check for computer limitations if the debugger does not show like Figure 7-6.



Figure 7-6. Device Manager View

- 3. Try to program with MSPM0 Launchpad to check whether the PC environment setting is OK.
- 4. For your customized board, check the schematic by referring to Section 4.3. Pay attention to the Vcc, Vcore and reset pin setting.
- 5. Then, check the connection between the debugger and the MSPM0. Users can use multimeter to directly check the signal path at debugger side by referring to Section 5.3, and at MCU pin side by referring to the related data sheet.
- 6. Check the power supply on the board. Remember the power output of the debugger has limitations and the output voltage can only be 3V3. An additional power supply can be needed.
- 7. Use oscilloscope to check the signal wave on SWDIO and SWCLK, especially when the wire is very long. Please make sure the signal establishment time is enough.

If the program failure is met for the second time and the device can be programmed before, then refer to Section 7.1.

7.3 Reprogram with SWD Disabled

On MSPM0, the SWD function is only enabled after reset and is impossible to re-enable through software in free run mode. Here are the common methods to reprogram MSPM0 under this condition.

- 1. Add a delay like 5 seconds before SYSCFG_DL_init(). With that, users have some time to connect MSPM0 before disabling SWD is executed.
- 2. Use ROM Bootloader. Before powering on MSPM0, pull and hold PA18 to be high. Program the flash with the right code. Then release PA18.
- 3. Use factory reset. Before powering on, press and hold the reset button. Perform a factory reset according to Section 7.1. When prompted to reset the chip, release the reset button. Then, the chip is blank.



7.4 MCU Performs Differently in Debug and Free Run

MSPM0 performs differently in debug and free run. Check the setting on PA18. The device enters the Bootloader in free run mode after MSPM0 is reset or repower, when PA18 input is **pulled to a high level** or **affected by noise with this pin floating**. If you meet this problem and PA18 cannot be pulled to a low level with an external resistor, you can follow the steps in Figure 7-7 to disable BSL or change the invoke pin assignment. As these settings need to change NOMAIN, please refer to the Program NONMAIN chapter for the related IDE in Section 3.4.



Figure 7-7. Disable BSL

7.5 Set SWD Password

The SWD interface can be configured to be disabled, enabled, or enabled with a 128-bit password by writing the BOOTCFG0 and SWDPW registers in NONMAIN. See the device Technical Reference Manual and Cybersecurity Enablers in MSPM0 MCUs for more information about NONMAIN and SWD password. Users can follow the steps to add password on SWD.

- 1. Enable and input SWD Password through sysconfig.
- 2. Enable Nonmain configuration.
- 3. After repowering, the device is locked.



PROJECT CONFIGURATION (1) Project Configur 1/1	Configuration NVM Step1	(+) ADD	EMOVE ALL	Properties for: adc12_14bit_resolution	on_LP_MSPM0G3	SUCD2
MSPM0 DRIVER LIBRARY (6) SYSTEM (7)	Quick Profiles		^	✓ General IIN Dependencies	Core	Texas Instruments XDS110 USB Debug Probe/CORTEX_M0P V
Board 1/1 <a>→ Configuration 1/1 <a>→ DMA	Accept configuration risks Debug Security Profiles	Security Level 1 - Custom restriction	ns 👻	{	Category Erase Co	MSPM0 Flash Settings Program/Memory Load Options Debugger Options USBN0 Flash Sattings
GPI0 2 ♥ ↔ SYSCTL 1/1 ♥ ↔ SYSTICK ↔ WWDT ↔	Boot Configuration Routine (BCR) Configurat	on	^	 ⊕ Link Order ✓ № Tools > Ø SysConfig 	IIIWarn See MSF Erase	nash mor rish company fig moonying conversion moorrectiy, or erasing it without program M0 documentation for more details method
✓ ANALOG (5) ADC12 (+) COMP (+)	Debug Security Policy Configuration Enable Physical Debug Port (SW-DP)	v	^	>	Eras C Eras O Eras	se MAIN memory only se DATA memory only se MAIN and DATA memory
GPAMP ⊕ OPA ⊕	Enable Application Debug Access ③	Enabled with password match		Arm Objcopy Stility [dis Executable Actions	O Eras	se MAIN and NONMAIN memory (see warning above)
VREF ⊕ ~ COMMUNICATIONS (5) 12C 12C ⊕ 12C - SMBUS ⊕ SPI ⊕	SWD Password[0] SWD Password[1] SWD Password[2]	0xFFFFFF1 0xFFFFFF1 0xFFFFFF1	_	Clang-Tidy Debug	O Eras O Eras O Do	se MAIN, DAIN, and NUNMAIN memory (see warning above) se MAIN and NONMAIN necessary sectors only (see warning above se MAIN memory sectors by range (specify below) not erase Flash memory
UARI (†) UART-LIN (*)	SWD Password[3]	0xFFFFFF1			Sector E	rase: all 1kB sectors between Start and End address will be erased
	Step3					
Connec router	nstruments XDS110 USB Debug Probe/CORTEX_ ting to the target: (Error -6305) PRSC module fa register. (Emulation package 20.0.0.3344)	M0P Error × iled to write to a Cancel Retry				

Figure 7-8. Enable SWD Password

Note

SBW security only works after repowering.

Here are the steps to reprogram MSPM0 with the password. This action does not erase NONMAIN, so the password remains active unless NONMAIN is modified.

- 1. In the CCS project, select *targetConfigs* → *MSPM0xxxx.ccxml*. Input password in .ccxml file. Ctrl+S to save .ccxml file.
- 2. Right-click the .ccxml and select *Start Project-less Debug*. First select *DebugAccessPasswordAuthentication*. After the GEL ouput shows *Command execution completed*, select *FactoryReset*. For more about factory reset, refer to Section 7.1.
- 3. Now, the device returns to empty and is ready to reprogram a new firmware.





Figure 7-9. Clear SWD Password

Note

Enabling a password can work with CCS, IAR and Keil. Clearing passwords can only work on CCS.

7.6 BSL Related Questions

For questions about how to use bootloader, see MSPM0 Bootloader (BSL) Implementation. This provides an overview of bootloader implementation and step-by-step instructions.

For questions about bootloader protocol and the spec, see the MSPM0 Bootloader User's Guide.

7.7 Reach Expected Current in LPM Mode

On MSPM0, if there are peripherals requiring high speed clock above the settled LPM mode, then the current consumption above the spec is listed on the data sheet. The best solution is to reset all the peripherals before entering the LPM mode. After getting out of the LPM mode, reconfigure the peripheral again.

For more detailed instructions, refer to MSPM0G3507 Low Power Test and Guidance.

7.8 CCS Common Questions

In this section, some common questions met in CCS are introduced. Here are some additional documents for reference when meeting questions with TI's complier, linker or IDE:

- Texas Instruments, MSPM0 SDK QuickStart Guide for CCS, webpage
- Texas Instruments, CCS IDE Guide for MPSM0, webpage
- Texas Instruments, Code Composer Studio User's Guide, webpage
- Texas Instruments, ARM Assembly Language Tools User's Guide, user's guide
- Texas Instruments, ARM Optimizing C/C++ Compiler User's Guide, user's guide
- Texas Instruments, TI Arm Clang Compiler Tools User's Guide, webpage



7.8.1 Change the Optimization Level

The default SDK example is with optimization level 2. The code size is smaller. However, this causes a mismatch of the C code and the assembly code and breakpoint cannot be added at the certain C code line. To solve this issue, choose the optimization from level 2 to level 0.



Figure 7-10. Change Optimization Level



7.9 Keil Common Questions

7.9.1 Copy Keil Example Out of SDK

If example code is copied out of SDK and compiled directly, then there are errors. The root cause lies on the SDK and SysConfig address setting in the code example. To solve this problem, see Figure 7-11.



Figure 7-11. Copy Keil Example Out of SDK



8 Summary

This document is a good start for the MSPM0 development and provides an overview of MSPM0 ecosystem and step-by-step instructions. Users are also provided with clear processes and image explanations. In addition to basic knowledge, the document also lists references and further reading materials for users to refer to further. TI recommends this document for users to quickly handle MSPM0 development work and overcome common obstacles.

9 Technical Documentation Resources

9.1 Technical Reference Manuals

Technical reference manuals introduce the application method and characteristic of MSPM0 MCUs, including but not limited to the abstract model of CPU and peripherals, working mode, and corresponding register configuration method.

- Texas Instruments, MSPM0 C-Series 24MHz Microcontrollers, technical reference manual
- Texas Instruments, MSPM0 G-Series 80MHz Microcontrollers, technical reference manual
- Texas Instruments, MSPM0 L-Series 32MHz Microcontrollers, technical reference manual

9.2 Subsystems

This section lists all the subsystem examples based on MSPM0 MCUs. As MSPM0s have become smaller and extremely cost-competitive, MSPM0s have begun to replace systems that were historically performed by fixed function analog devices. For more information, users can also refer to the Analog Engineer's Circuit Cookbook: M0+ MCUs e-book and the Arm® Cortex ®-M0+ MCUs subsystems product page.

- Texas Instruments, 5V Interface, subsystem design
- Texas Instruments, ADC to I2C, subsystem design
- Texas Instruments, ADC to SPI, subsystem design
- Texas Instruments, ADC to UART, subsystem design
- Texas Instruments, CAN to I2C Bridge, subsystem design
- Texas Instruments, CAN to SPI Bridge, subsystem design
- · Texas Instruments, CAN to UART bridge, subsystem design
- · Texas Instruments, Common Amplifier Topologies: PGA, subsystem design
- Texas Instruments, Connected Diode Matrix, subsystem design
- Texas Instruments, DMA Ping Pong With ADC, subsystem design
- Texas Instruments, Data Sensor Aggregator Subsystem Design, subsystem design
- Texas Instruments, Digital FIR Filter, subsystem design
- Texas Instruments, Digital IIR Filter, subsystem design
- Texas Instruments, Emulating a Digital MUX, subsystem design
- Texas Instruments, Frequency Counter: Tone Detection, subsystem design
- Texas Instruments, Function Generator Using DAC8, subsystem design
- Texas Instruments, I2C Expander Through UART Bridge, subsystem design
- Texas Instruments, I2C to UART Subsystem Design, subsystem design
- · Texas Instruments, IO Expander With SPI, I2C, and UART, subsystem design
- Texas Instruments, LED Driver With PWM, subsystem design
- Texas Instruments, Low-Cost MSPM0C MCUs as an I/O Expander, subsystem design
- Texas Instruments, MCU Design Techniques: ADC to PWM, subsystem design
- Texas Instruments, PWM DAC, subsystem design
- Texas Instruments, Parallel IO to UART Bridge, subsystem design
- Texas Instruments, Power Sequencer, subsystem design
- Texas Instruments, Scanning Comparator, subsystem design
- Texas Instruments, Task Scheduler, subsystem design
- · Texas Instruments, Thermistor Temperature Sensing, subsystem design
- Texas Instruments, Transimpedance Amplifier, subsystem design
- · Texas Instruments, Two OPA Instrumentation Amplifier With M0 Devices, subsystem design
- Texas Instruments, UART to I2C Bridge, subsystem design
- Texas Instruments, UART to SPI Bridge, subsystem design
- Texas Instruments, Emulate EEPROM With FLASH (Type B), subsystem design



· Texas Instruments, Emulate EEPROM with FLASH (Type A), subsystem design

9.3 Reference Designs

This section lists all the reference designs based on MSPM0 MCUs. The references contain full design resources and most are a reference for developing an end equipment.

- Texas Instruments, 24V, 35W sensorless FOC BLDC reference design with 85VAC to 265VAC, PF of 0.92, single-stage PFC, design guide
- Texas Instruments, 250W motor inverter reference design with GaN IPM DRV7308, design guide
- Texas Instruments, Cost-Effective, 3-Phase CT Electricity Meter Ref. Design Using Standalone ADC, design guide
- Texas Instruments, IO-Link device implementation for sensors and actuator reference design, design guide
- Texas Instruments, Low-Cost Blood Pressure and Heart Rate Monitor Reference Design, design guide
- Texas Instruments, Radiation-Hardened Space Battery Management System (BMS) Reference Design, design guide
- Texas Instruments, Single-Chip Pulse Oximeter Reference Design With 90dB Dynamic Range for Lower PI, design guide
- Texas Instruments, Smart Analog Sensor Interface for Smoke Detection With Ambient Light Cancellation Reference Design, design guide
- Texas Instruments, Three-Phase Shunt-Based Energy Metrology Reference Design, design guide

9.4 Hardware EVM User's Guides

The Hardware EVM User's Guides includes all the documentations of Launchpads and EVMs, related to MSPM0.

- Texas Instruments, LP-MSPM0C1104 Evaluation Module User's Guide, user's guide
- Texas Instruments, LP-MSPM0G3519 Evaluation Module User's Guide, user's guide
- Texas Instruments, LP-MSPM0L1117 Launchpad Development Kit, user's guide
- Texas Instruments, LP-MSPM0L2228 Evaluation Module User's Guide, user's guide
- Texas Instruments, MSP-DRV-ADAPT-EVM Evaluation Module User's Guide, user's guide
- Texas Instruments, MSP-LITO-L1306 Evaluation Module User's Guide, user's guide
- Texas Instruments, MSPM0G3507 LaunchPad Development Kit User's Guide (LP-MSPM0G3507), user's guide
- Texas Instruments, MSPM0L1306 LaunchPad Development Kit, user's guide
- Texas Instruments, XDS110-ETP Evaluation Module User's Guide, user's guide

9.5 Application Notes and Others

This section lists all the application notes, application briefs, product overviews, subsystem designs and functional safety informations based on MSPM0 MCUs and the peripherals. The application note is the technical document about device, device peripherals or applications, which is the most common type of technical documentation on ti.com.

- Texas Instruments, A Self-Calibratable Current Detection Solution Based on MSPM0, application note
- Texas Instruments, A2L Refrigerant Standard Overview and TI Mitigation Control Board Designs for Designers, application note
- Texas Instruments, Automotive Seat Comfort Module Using MSPM0, application brief
- Texas Instruments, BLDC and PMSM Control Using Sensorless FOC Algorithm Based on MSPM0 MCUs, application brief
- Texas Instruments, BQ769x2 Control Based on MSPM0 Through I2C, application note
- Texas Instruments, BQ79616 Control Based on MSPM0 Through UART to CAN, application note
- Texas Instruments, Build Scalability in Cordless Power and Garden Tools Using Low-Cost MSPM0 MCUs, application brief
- Texas Instruments, Closed Loop Constant Power Drive to Simplify Heater Element Control and Extend Battery Life, application note
- Texas Instruments, Cybersecurity Enablers in MSPM0 MCUs, application note
- · Texas Instruments, Designing Single- and Three-Axis Selfie Sticks With MSPM0 MCUs, application brief
- Texas Instruments, Designing Temperature Monitoring Systems with NTC and RTD, application note
- Texas Instruments, Designing With MSPM0 MCUs and Segment LCDs, application note



- Texas Instruments, Dual-Ray Smoke Detector with the TPS8802 and MSPM0 MCUs, application note
- Texas Instruments, EEPROM Emulation Type A Solution, application note
- Texas Instruments, EEPROM Emulation Type B Design, application note
- Texas Instruments, Flash Multi Bank Feature in MSPM0 Family, application note
- Texas Instruments, Full-Featured Automotive Side Mirror, application brief
- Texas Instruments, Functional Safety Manual for MSPM0G, functional safety information
- Texas Instruments, How to Charge With Smart Battery Using MCU in Between to Translate SMBus/I2C, application note
- Texas Instruments, Increasing Flexibility in Your Battery Management Designs With a Low-Cost MSPM0, application brief
- Texas Instruments, Increasing Flexibility in Your Electrical Thermometer Designs With Low-Cost MSPM0, application brief
- Texas Instruments, Isolated Loop Powered 4 to 20mA Field Transmitter Designs, application note
- Texas Instruments, Low-Frequency Subsystem and VBAT Feature in MSPM0L222X, application note
- Texas Instruments, MSPM0 Advanced Control Timer Helps for Better Control and Better Digital Output, application note
- Texas Instruments, MSPM0 ADC Noise Analysis and Application, application note
- Texas Instruments, MSPM0 Bootloader (BSL) Implementation, application note
- Texas Instruments, MSPM0 Bootloader, user's guide
- Texas Instruments, MSPM0 C-Series MCU Hardware Development Guide, application note
- Texas Instruments, MSPM0 Design Flow Guide, application note
- Texas Instruments, MSPM0 Enables Cost-Effective Field Transmitter Applications, application note
- Texas Instruments, MSPM0 G-Series MCUs Hardware Development Guide, application note
- Texas Instruments, MSPM0 G-Series MCUs Power Optimization Guide, application note
- Texas Instruments, MSPM0 Gauge L1 Solution Guide, application note
- Texas Instruments, MSPM0 Gauge L2 Solution Guide, application note
- Texas Instruments, MSPM0 L-Series MCUs Hardware Development Guide, application note
- Texas Instruments, MSPM0 L-Series MCUs Power Optimization Guide, application note
- Texas Instruments, MSPM0 Live Firmware Update (LFU) Bootloader Implementation, application note
- Texas Instruments, MSPM0 MCUs Quick Reference Guide, application note
- Texas Instruments, MSPM0 MCUs: More Options, Unlimited Possibilities, product overview
- Texas Instruments, MSPM0 Motor Control, application note
- Texas Instruments, MSPM0 Sensored FOC Tuning, user's guide
- Texas Instruments, MSPM0 Sensorless FOC Tuning Guide, user's guide
- Texas Instruments, MSPM0 Universal FOC Tuning, user's guide
- Texas Instruments, MSPM0-Based Low-Cost Single-Chip Pulse Oximeter Reference Design, product overview
- Texas Instruments, MSPM0-Based Medical Alarm Design, application brief
- Texas Instruments, MSPM0: Idea to Product With Easy-to-Use Tools, Software, and Academy, product overview
- Texas Instruments, MSPM0C: A New Standard 32-Bit MCU for 8-Bit and 16-Bit MCU Applications, product overview
- Texas Instruments, MSPM0Cx- Toothbrush and Shaver, application brief
- Texas Instruments, MSPM0G3507 Low Power Test and Guidance, application note
- Texas Instruments, MSPM0L or MSPM0G: How to Pick the Right MSP Microcontroller for Your Application, application note
- Texas Instruments, Functional Safety Manual for MSPM0L130x-Q1, functional safety information
- Texas Instruments, MSPM0Lx22x Microcontrollers Enabling Low-Power Display and Security Designs, product overview
- Texas Instruments, Make System Design Easy With MSPM0 Precision Analog, application note
- Texas Instruments, Migration Guide From Microchip to MSPM0, application note
- Texas Instruments, Migration Guide From NXP to MSPM0, user's guide
- Texas Instruments, Migration Guide From Renesas RL78 to Arm-Based MSPM0, application note
- Texas Instruments, Migration Guide From STM8 to MSPM0, application note
- Texas Instruments, Operating Time of MSPM0 Powered by a Capacitor, application note


- Texas Instruments, Optimize Automotive Body Electronics Designs With AEC-Q100 MSPM0 MCUs, application brief
- Texas Instruments, Optimized H-Bridge Driver Control for Stepper and BDC Motors Using MSPM0 MCUs, application brief
- Texas Instruments, Optimizing Field Sensor and Transmitter Applications With MSPM0 MCUs, application brief
- Texas Instruments, PGA460 Control Based on MSPM0 for Distance Detection, application note
- Texas Instruments, PIR Motion Detection With MSPM0, application note
- Texas Instruments, Realizing HVAC FAN Control Design with MSPM0 MCU, product overview
- Texas Instruments, Realizing Low-Power and High-Scalability OBC Wake-up Design with MSPM0 MCU, product overview
- Texas Instruments, Realizing UWB Passive Entry Passive Start (PEPS) Design with MSPM0 MCU, product overview
- · Texas Instruments, Scalable Battery Backup Subsystem With Adjustable Output, product overview
- Texas Instruments, Sensored Brushed DC Motor Control Based on MSPM0, application note
- Texas Instruments, Simplifying Design in True Wireless Stereo Control With a Low-Cost MSPM0 MCU, application brief
- Texas Instruments, Simplifying Pulse Oximeter Designs With Low-Cost Highly Integrated MSPM0 MCUs, application brief
- Texas Instruments, Software Defined Glass LCD Solution Based on MSPM0 MCUs, application note
- Texas Instruments, Streamlining Smoke Detector Designs With Highly Integrated MSPM0 MCUs, application brief
- Texas Instruments, TI's Smallest M0+ MCU Package Enables Room to do More in Your Design, application brief
- Texas Instruments, TPS929xxx LED Driver Control Using MSPM0 Through UART Over CAN, application note
- Texas Instruments, Ture Wireless Stereo (TWS) Charging Case Design Based on MSPM0L1105, application note
- Texas Instruments, Understanding the MSPM0 Debug Subsystem, application note
- Texas Instruments, Using MSPM0 MCUs to Design Trapezoidal-Based BLDC Motor Controllers, application brief

10 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision D (October 2024) to Revision E (March 2025)		Page	
•	Updated Overview section to include MSPM0 device overview and ecosystem introduction	4	
•	Added a link to Section 7 in Software Development Instructions section.	7	
•	Added a link for the LaunchPad product page in LaunchPad Introduction section	8	
•	Added a description of MSPSDK in the MSPM0-SDK Setup section	9	
•	Removed MSPM0-SDK Example figure	10	
•	Added the code example figure in the Examples Folder Introduction section.	10	
•	Updated the CCS Quick Start section to include a different CCS version description	<mark>21</mark>	
•	Updated the CCS installation picture due to CCS version update in CCS Installation section	<mark>21</mark>	
•	Updated <i>Import a SDK Example</i> section with the CCS example importing picture due to CCS version update.	22	
•	Updated <i>Example Download and Debug</i> section with the Debugger Selection picture due to CCS version update	23	
•	Updated the debugger selection picture due to CCS version update in Generate Hex Files section	25	
•	Updated the Nonmain programming picture due to CCS version update in <i>Program NONMAIN</i> section Updated <i>Obtaining a MSPM0 Package</i> section to give more detailed instruction for MSPM0	26	
	package generation	43	
•	Added C-GANG information to Mass Production Instructions section	48	
•	Changed the product introduction to C-GANG in Section 5.2.3 section	52	
•	Added more detailed instructions and images to Program Hardware Quick Start section	57	
•	Change the unlock picture due to CCS version change in the Unlock Through CCS section	63	
•	Added more CCS reference links in CCS Common Questions section	67	
•	Changed the picture due to CCS version update in Change the Optimization Level section	<mark>68</mark>	
•	Added Technical Documentation Resources section	70	



IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2025, Texas Instruments Incorporated