

Application Note

MSPM0 Design Flow Guide



Eason Zhou

ABSTRACT

The application note is written in a flow, in order to give clear steps to every engineer in a project when developing MSPM0 MCUs. It can help to find the related materials following the project progress. All of the instructions are in steps; tips will be given at the same time.

Table of Contents

1 Overview	2
2 Step1: MSPM0 Selection	4
3 Step2: MSPM0 Software Evaluation	6
3.1 Hardware Setup	6
3.2 MSPM0 Software Setup	7
3.3 MCU Function Evaluation	9
4 Step3: Customized PCB Board Design	11
4.1 Get MSPM0 Package	12
4.2 Fix Pin Functions	13
4.3 Schematic and PCB Generation	13
5 Step 4: Mass Production	14
5.1 Generate Production Files	14
5.2 Program Software and Tools Selection	15
6 Appendix	16
6.1 Light a LED and CCS Quick Introduction	16
6.2 Steps to Generate PCB Library	21
6.3 MSP-GANG Quick Introduction	24
Revision History	26

List of Figures

Figure 1-1. MSPM0 Design Flow	3
Figure 2-1. MSPM0 Device List	4
Figure 2-2. MSPM0 Important Document List	4
Figure 2-3. Device Comparison Table	5
Figure 2-4. Ordering and Quality Part View	5
Figure 3-1. MSPM0G3507 Launchpad	6
Figure 3-2. SDK Structure	7
Figure 3-3. MSPM0 SDK	8
Figure 3-4. Document Overview	9
Figure 3-5. Sysconfig Overview	10
Figure 3-6. CCS Project Overview	10
Figure 3-7. Head File and Start File Porting	11
Figure 3-8. E2E online	11
Figure 4-1. Ultra Librarian tool entrance	12
Figure 4-2. Sysconfig MCU view	13
Figure 4-3. MSPM0 Minimum System	13
Figure 4-4. MSPM0 Schematic Other Attentions	14
Figure 5-1. Create Program Files	14
Figure 5-2. Program Software and Tool	15
Figure 5-3. Pin Connection of Standalone XDS110	15
Figure 5-4. XDS110 on Board	16
Figure 6-1. CCS Installation	17

Figure 6-2. MSPM0 Support Selection.....	17
Figure 6-3. J-link selection.....	18
Figure 6-4. Hardware Setup.....	18
Figure 6-5. Choose CCS Workspace.....	19
Figure 6-6. Import Project.....	19
Figure 6-7. Remove Duplicated Project.....	19
Figure 6-8. Debug Code.....	20
Figure 6-9. Common Used Project Settings.....	20
Figure 6-10. Common Used Debug Functions.....	20
Figure 6-11. Ultra Librarian Tool Entrance.....	21
Figure 6-12. Ultra Librarian Tool Device Selection.....	21
Figure 6-13. Ultra Librarian Tool CAD Download.....	22
Figure 6-14. Run Altium Designer Script.....	22
Figure 6-15. Generate Library.....	23
Figure 6-16. Select Footprint.....	23
Figure 6-17. Import Library.....	23
Figure 6-18. MSP-GANG Pin Assignment.....	24
Figure 6-19. Download Code Using MSP-GANG with GUI.....	24
Figure 6-20. Enable Non-main Programming.....	25
Figure 6-21. Generate and Save Image.....	25
Figure 6-22. Change Mode.....	25
Figure 6-23. Offline Porgramming.....	26

List of Tables

Table 3-1. MSPM0 Debugger Compare.....	6
Table 3-2. MSPM0 Related IDE Compare.....	7
Table 3-3. MSPM0 Example Coverage.....	8
Table 3-4. Empty Project Description.....	8

Trademarks

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

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

All trademarks are the property of their respective owners.

1 Overview

The document is written in a flow, in order to give clear steps to every engineer in a project when developing MSPM0 MCUs. Besides, a list the related documents are provided for further reference. The appendix section uses an example – light an LED to show the basic usage method of software Code Composer Studio™ (CCS) , guide you on how to generate a PCB footprint files and how to use MSP-GANG (Mass production tool).

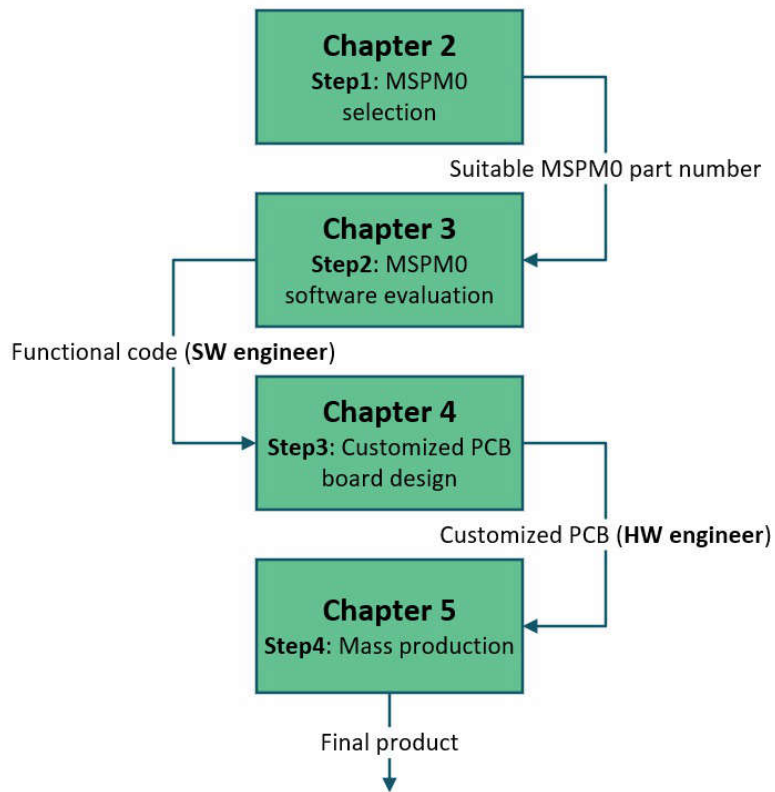


Figure 1-1. MSPM0 Design Flow

DRAFT
TI Confidential – NDA R

2 Step1: MSPM0 Selection

This step shows you how to find a suitable MSPM0 orderable number.

Here is the [link](#) for the MSPM0 device list. After you go to this page, you can use the filter on the left to do Initial screening based on your MCU peripheral requirement or you can directly go to the device page through the left search textbox.

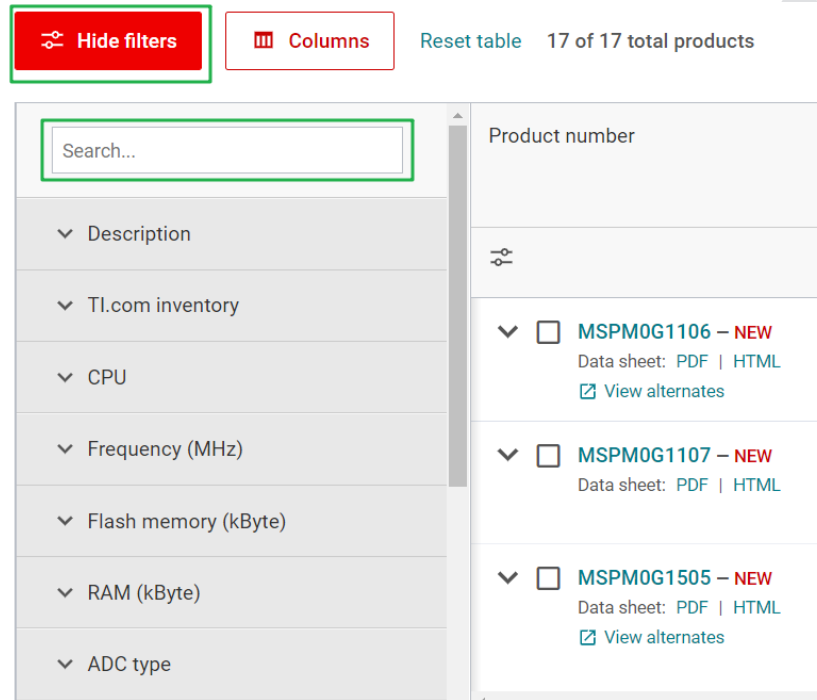


Figure 2-1. MSPM0 Device List

After you go to the **device page**, you can find more spec or functions details for this product. 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 of dedicated MSPM0. The device-specific TRM introduces the application method and characteristics of a series MSPM0. The device-specific errata introduces corrigendum description of MSPM0 related series or versions.

NEW
MSPM0L1306 PREVIEW

32-MHz Arm® Cortex®-M0+ MCU with 64-KB flash, 4-KB SRAM, 12-bit ADC, comparator, OPA

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

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

ERRATA [MSPM0 Microcontrollers Errata](#)

Figure 2-2. MSPM0 Important Document List

In the *Device Comparison* table in the device-specific data sheet, you can easily do a further screening by a quick compare between different part numbers.

5 Device Comparison

Table 5-1. Device Comparison

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

Figure 2-3. Device Comparison Table

For the orderable part number and its reference price, see the **Ordering & quality** part of the **device page**.

MSPM0L1306 PREVIEW Data sheet Order now

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

Ordering & quality

Part number ↓↑	Buy	Ti.com inventory ↓↑	Qty Price (USD) ↓↑	Package qty Carrier ↓↑
XMSM0L1306SDGS20R ACTIVE	Enter quantity Add to cart Limit: 5	93	1ku ▼	1 LARGE T&R
XMSM0L1306SDGS28R ACTIVE	Enter quantity Add to cart Limit: 10	170	1ku ▼	5,000 LARGE T&R

Figure 2-4. Ordering and Quality Part View

3 Step2: MSPM0 Software Evaluation

This step shows you how to setup a software evaluation environment for MSPM0. To get a step-by-step instruction based on CCS and launchpad, see [Section 6.1](#).

3.1 Hardware Setup

3.1.1 Debugger Selection

Here is a feature summary of different debugger examples supporting MSPM0. For XDS110 and XDS100 On-Board, they are TI's own debuggers. They will support more functions, like EnergyTrace™ (an energy measurement technology) and BSL than general debuggers. For XDS110 On-Board, it is the cheapest debugger solution, integrated into TI Launchpad. For more information, please refer to the next chapter [Section 3.1.2](#).

Table 3-1. MSPM0 Debugger Compare

Features	XDS110	XDS110 On-Board	J-Link
cJTAG (SBW)	√	√	√
BSL tool	√	√	
Backchannel UART	√	√	
Power supply	1.8~3.6 V	3.3/5 V	5 V
IDE: CCS	√	√	√
IDE: 3rd party	IAR/Keil	IAR/Keil	IAR/Keil

3.1.2 Launchpad Introduction

It is suggested to start MSPM0 development with a launchpad first. [Figure 3-1](#) shows an overview on the launchpad. It contains the MCU and a XDS110 debugger. You can also use other debuggers like J-Link to debug the MCU after removing the jumpers.

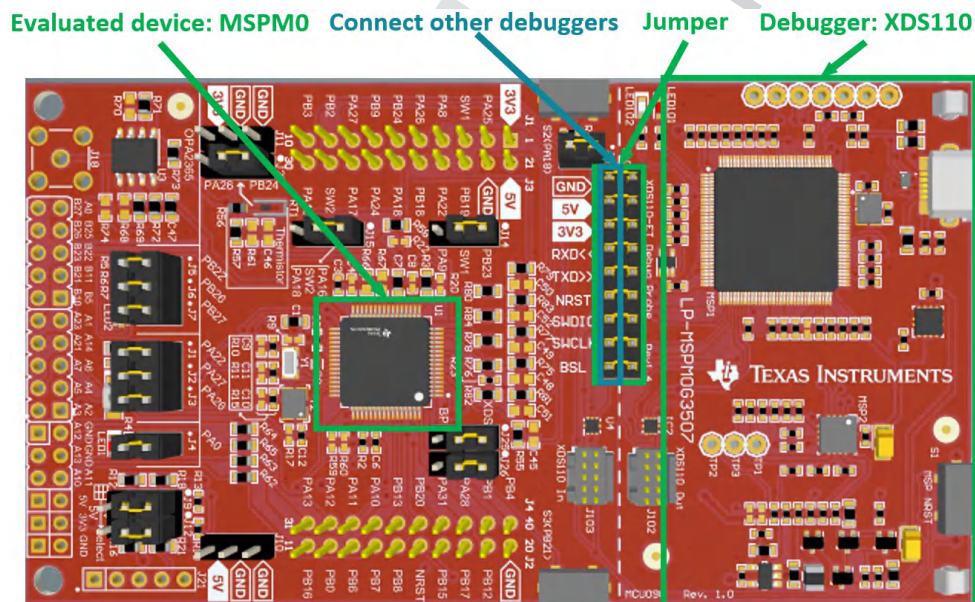


Figure 3-1. MSPM0G3507 Launchpad

Here is the orderable launchpad for a sub family MSPM0 evaluation. You can also find the device-specific EVM User's Guide with **schematic** under these links:

- LP-MSPM0L1306: [Landing page](#), [MSPM0L1306 LaunchPad Development Kit User's Guide](#)
- LP-MSPM0G3507: [Landing page](#), [MSPM0G3507 LaunchPad Development Kit User's Guide](#)

3.2 MSPM0 Software Setup

3.2.1 Install IDE

Table 3-2 shows a quick compare for different IDEs supporting MSPM0. They are CCS, IAR and Keil. CCS is TI's own IDE, we would suggest customers to use [CCSTUDIO](#), which is more mature than CCSTUDIO-THEIA.

Table 3-2. MSPM0 Related IDE Compare

IDEs	CCSTUDIO / CCSTUDIO-THEIA	IAR	Keil
License	Free	Charged	Charged
Compiler	TI Arm® Clang / GCC	IAR C/C++ Compiler for Arm	Arm Compiler Version 6
XDS110	Supported	Prepare to support	Prepare to support
J-Link	Supported	Supported	Supported

Here are the related guides for different IDEs. It shows how to do setup and build projects on the related IDE.

- [CCS quick start guide](#)
- [CCS IDE guide for MSPM0](#)
- [IAR quick start guide](#)
- [IAR IDE guide for MSPM0](#)
- [Keil quick start guide](#)
- [Keil IDE guide for MSPM0](#)

For quick start guides, they are under the sub document "MSPM0 SDK Quick Start Guides", which will show how to setup the environment and run the first example on the related IDE.

For IDE guides, they will show the related development knowledge beyond the quick start guides. We would suggest you to read the quick start guide first and then go through IDE guide for high level MSPM0 development knowledge.

3.2.2 Install SDK

Here is the structure of [MSPM0 SDK \(offline\)](#) version, which will be shown in the folder type.

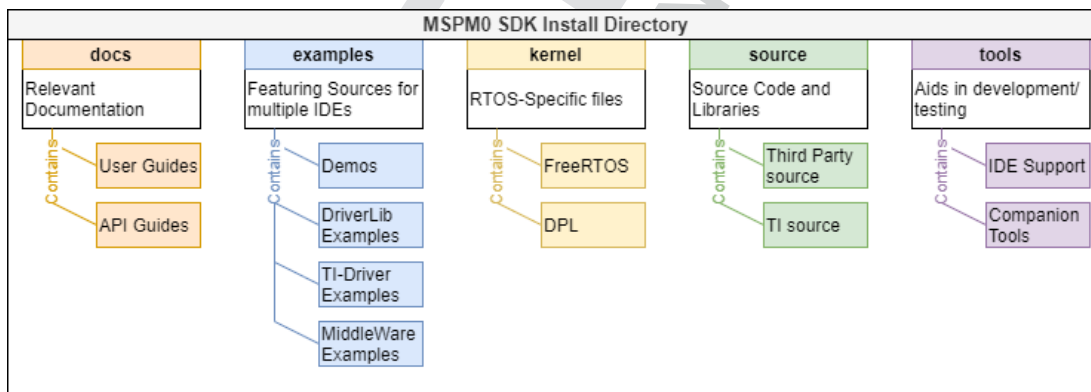


Figure 3-2. SDK Structure

The mostly used folders are examples folder and docs folder. In this part we will focus on introduce them, as shown in [Figure 3-3](#).

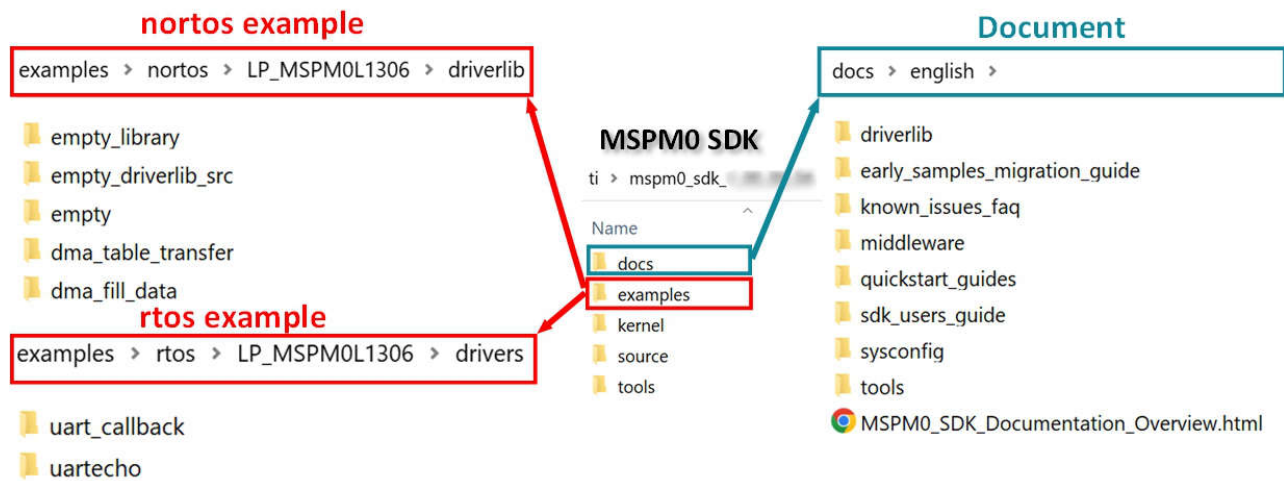


Figure 3-3. MSPM0 SDK

TI will make a launchpad for one MSPM0 sub family, with a superset MSPM0 on board. For this MSPM0 sub family, they will reuse the same example code under the address "mspm0_sdk_x_x_x_x \ examples \ nortos \ LP_MSPM0xxxx" for nortos example and "mspm0_sdk_x_x_x_x \ examples \ rtos \ LP_MSPM0xxxx" for rtos example. Table 3-3 shows all the example code type.

Table 3-3. MSPM0 Example Coverage

Supported by SDK	Platform 1		Platform 2	Platform 3
IDE	CCS		Keil	IAR
Compilers	TI Arm-Clang	GNU Arm (GCC)	Arm/Keil Compiler	IAR Arm compiler
RTOS	FreeRTOS			
Code examples	Driverlib (Low level driverlib)/TI Drivers(High level driverlib)			

In the rtos example level, there are two folders. The most important folder is drivers, which demo the peripheral control based on TI-Driver.

In the nortos example level, there are many folders. The most important folder is driverlib folder, which contains the peripheral example code based on Driverlib, as shown in Figure 3-3. For demo folder, it contains the example codes which use above one MCU peripherals. For other folders, they are all belong to the middle ware and contains solution for typical application. In nortos examples, you can also find four empty examples for users to build their own project. Here is the difference.

Table 3-4. Empty Project Description

Example	Type	Language	Use Sysconfig	Library Files in Project
empty	Project	C++	Yes	No
empty_cpp	Project	C	Yes	No
empty_library	Static library (.lib file in Debug folder after debugging)	C	No	No
empty_driverlib_src (Suggested)	Project	C	No	Yes

For docs folder, its structure and the important documents are shown in Figure 3-4. We strongly suggest the users to read these documents first when meet any questions about SDK .

MSPM0 SDK Documentation Overview.html

MSPM0 SDK Documentation

- Release Notes
- Quick Start Guides
- MSPM0 SDK User's Guide
- MSPM0 Tools Guide
- Manifest

SDK Components

DriverLib Documentation

- Driverlib Guide

Middleware Documentation

- Motor Control Library
- GUI Composer Library
- IQMath
- LIN Library

MSPM0 SDK Quick Start Guides

The following guides provide step-by-step instructions to get started quickly using MSPM0:

- MSPM0 SDK QuickStart Guide for CCS Theia
- MSPM0 SDK QuickStart Guide for CCS
- MSPM0 SDK QuickStart Guide for IAR
- MSPM0 SDK QuickStart Guide for Keil

MSPM0 SDK User Guide

- MSPM0 SDK Examples Guide
 - 1. Introduction
 - 2. MSPM0 SDK Examples
- SDK Overview
 - 1. SDK Components and Architecture
 - 2. SDK Structure
 - 3. Getting Started

MSPM0 Tools Guide

- IDEs and Compilers
 - Code Composer Studio
 - IAR Embedded Workbench for ARM
 - ARM Keil MDK
 - TI Arm Clang Compiler
 - Arm GCC Toolchain
- Code Generation and Ease-of-Use
 - SysConfig
- Debugging and Programming Tools
 - XDS-110
 - Segger
 - UniFlash

DriverLib Common APIs

Functions

<code>__STATIC_INLINE void DL_Common_updateReg (volatile uint32_t</code>	Writes value to specified register - retaining b
<code>void DL_Common_delayCycles (uint32_t cycles</code>	Consumes the number of CPU cycles specifi

Figure 3-4. Document Overview

Here are the online links for the most important documents:

- [MSPM0 SDK User Guide](#)
- [MSPM0 Tools Guide](#)
- [Driverlib API Guide](#)

3.3 MCU Function Evaluation

After the hardware, IDE and SDK is prepared. MSPM0 evaluation can started.

3.3.1 Fix Pin Functions and Generate Setting Code With Sysconfig

Sysconfig is a tool that can help to generate the peripheral setting code shown in [Figure 3-5](#). Here is the brief use flow:

- Add the wanted peripherals in “peripheral usage”
- Set its parameters in “Peripheral setting”, paired with TRM.
- After debugging, it can generate C code directly.

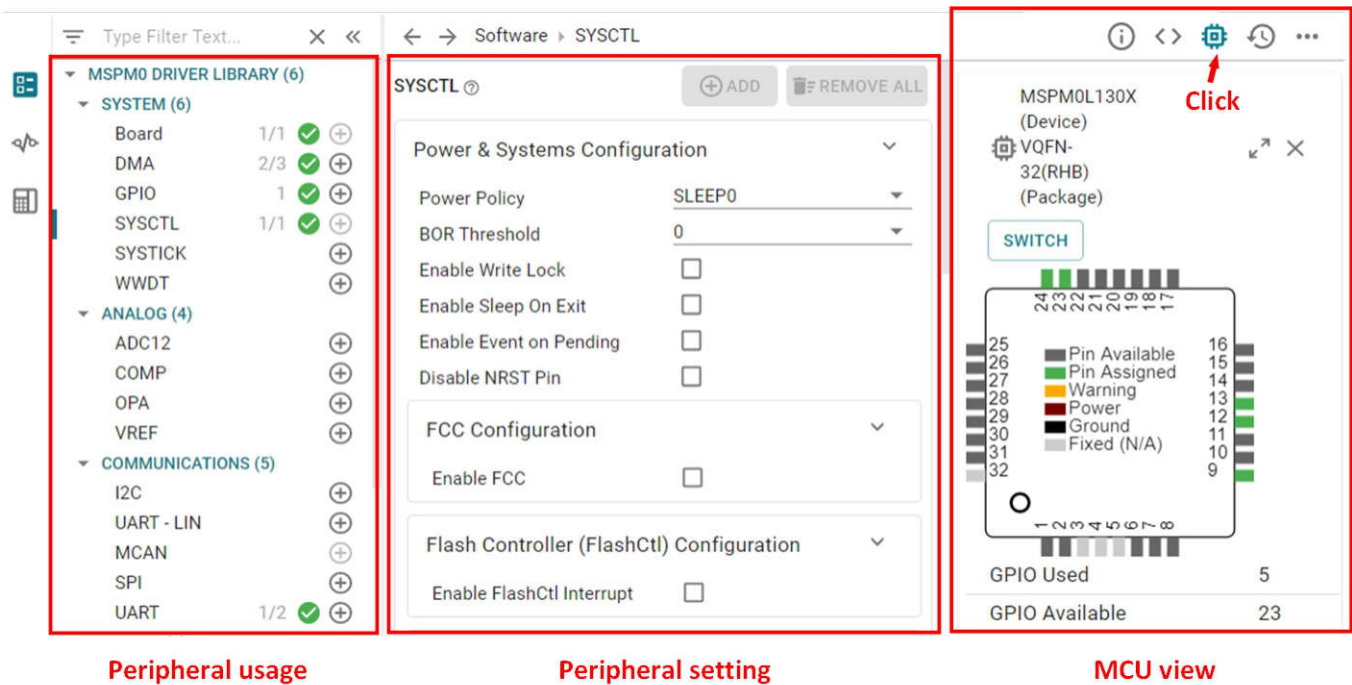


Figure 3-5. Sysconfig Overview

Besides of code generation, another advantage of sysconfig is that it can help check the peripheral resource and the function usage confliction. Under MCU view, paired with data sheet, it can easily help users to fix MCU pin functions.

For more details to use Sysconfig under different IDE, see the following document:

- [SysConfig Guide for MSPM0](#)

3.3.2 Coding and porting

This section provides a description of the MSPM0 example code based on CCS. The most important files are in red.

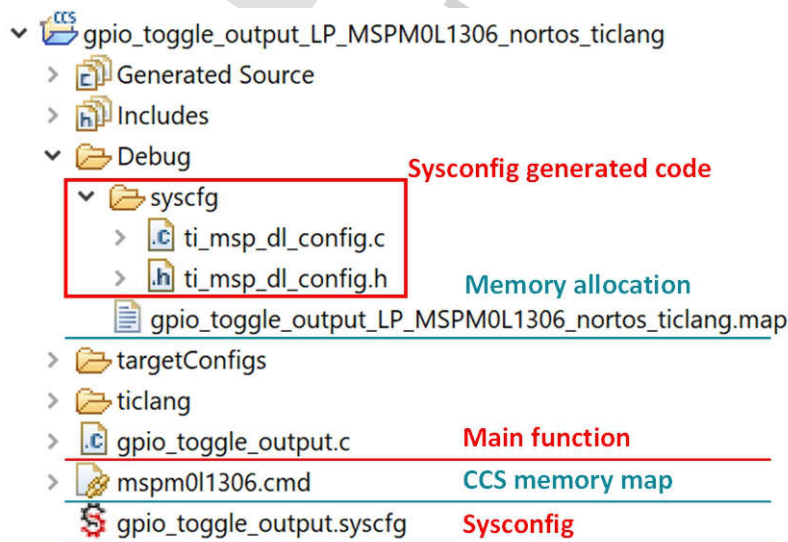


Figure 3-6. CCS Project Overview

Please pay attention that syconfig only generates the peripheral setting code. For peripheral usage, see the rtos / nortos code example in SDK, TRM, and [MSPM0 driverlib API guide](#).

Here is the common questions about CCS, when first use, for you to refer:

- **Can't set break point:** Change optimization level under this address (Your project > Properties > Build > Arm Compiler > Optimization level)
- **Output Hex / Bin files:** Please refer to [Section 5.1](#)
- **Can't find the .c / .h files in a folder:** Add the folder into the compile search path (Your project > Properties > Build > Arm Compiler > Include Options)
- **Can't import a project or its name becomes gray:** Please delete the project with the same name under the workspace.
- **Enable Non-main change:** Change the erase type under this address (Your project > Properties > Debug > MSPM0 Flash Setting > Erase MAIN and NONMAIN)

When you finish the basic code development and want to move the project from launchpad to the dedicated MSPM0 under the same sub family. You first need to click the "Switch" button in MCU view, as shown in [Figure 3-5](#). Then you may need to copy the related linker file or start up file into the project from SDK, as shown in [Figure 3-7](#).

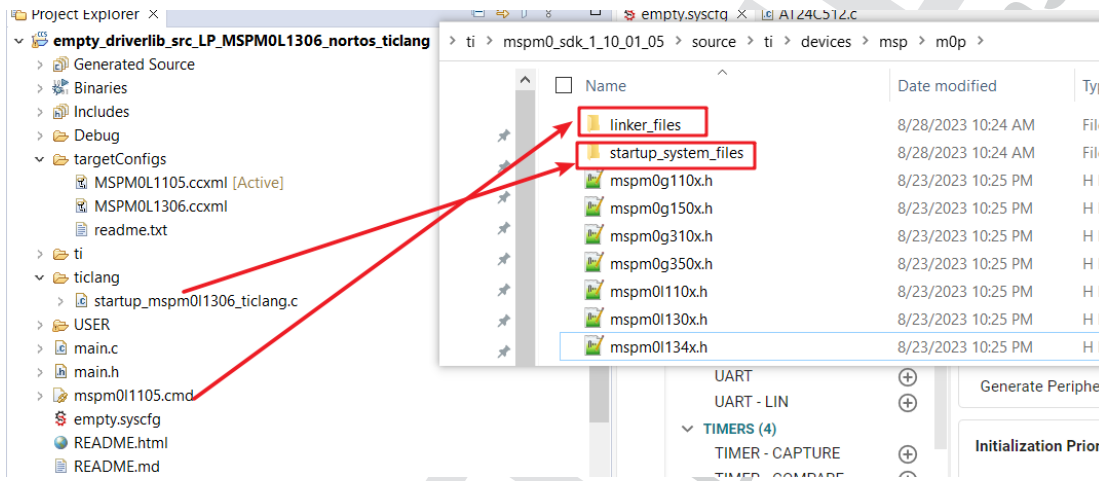
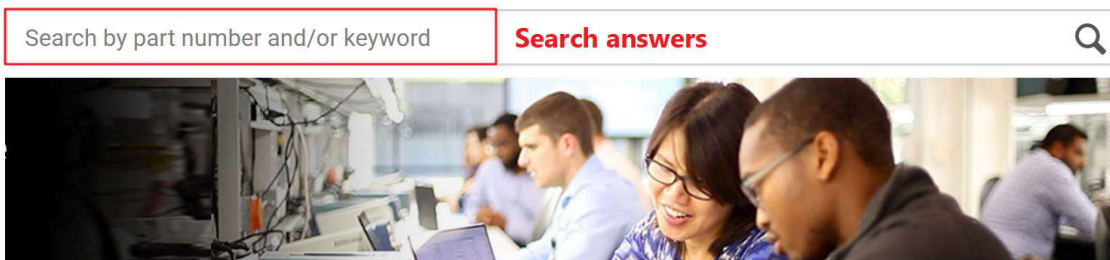


Figure 3-7. Head File and Start File Porting

For other questions, you can check whether the MSPM0 metrials summery ([MSPM0 MCUs Quick Reference Guide](#)) can give some help. You can also get support from online technical support forum: [E2E](#). TI engineers will give response in 24 hours on this online support platform.



Ask a new question

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

[Ask a new question](#) **Ask questions**

Get quality, packaging or ordering support

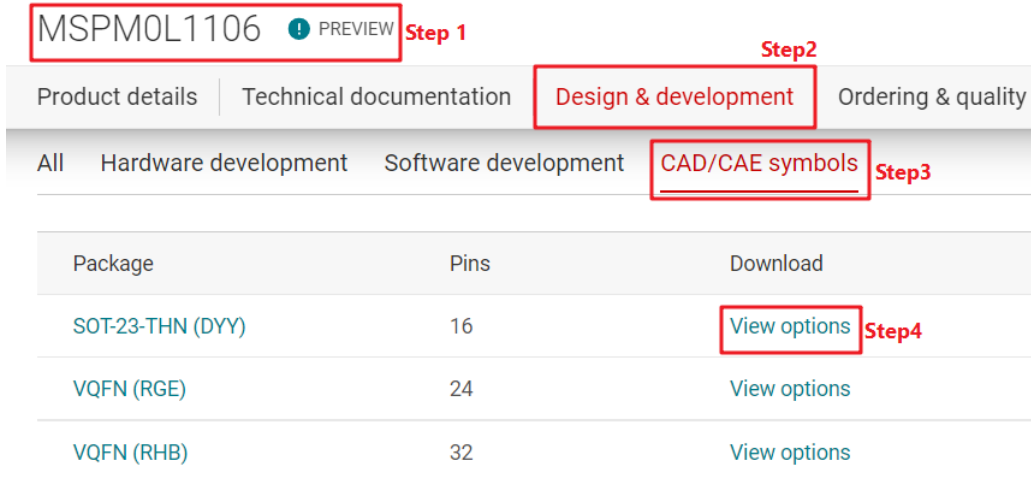
For non-design-related questions such as ordering semiconductor parts and tools, contact our customer support center where you can open a support ticket, chat with us 24 hours a day, Monday through Friday; or call the TI support team.

Figure 3-8. E2E online

4 Step3: Customized PCB Board Design

4.1 Get MSPM0 Package

The easy way to get the MSPM0 package is to use the Ultra Librarian tool on TI.com, as shown in [Figure 4-1](#). For the detailed instruction, see [Section 6](#).



MSPM0L1106 PREVIEW **Step 1**

Product details | Technical documentation | **Design & development** **Step 2** | Ordering & quality

All | Hardware development | Software development | **CAD/CAE symbols** **Step 3**

Package	Pins	Download
SOT-23-THN (DYY)	16	View options Step 4
VQFN (RGE)	24	View options
VQFN (RHB)	32	View options

Figure 4-1. Ultra Librarian tool entrance

DRAFT

TI Confidential – NDA Restriction

4.2 Fix Pin Functions

It is suggested to use MCU view of sysconfig to help you fix the pin function fist with software engineer.

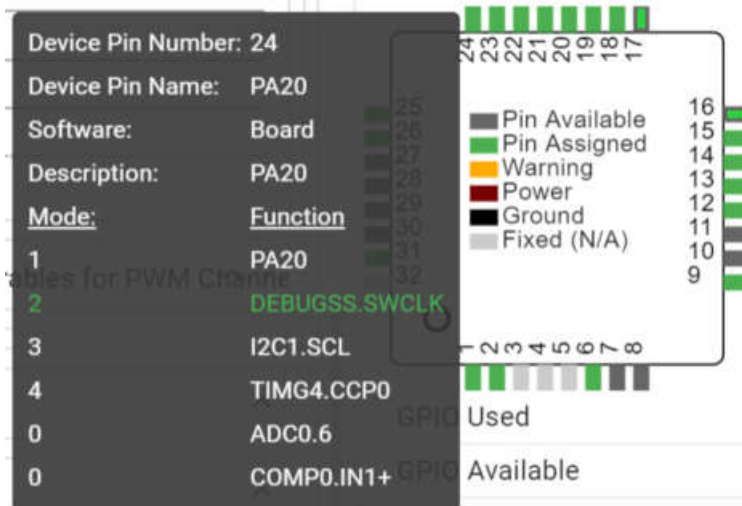


Figure 4-2. Sysconfig MCU view

4.3 Schematic and PCB Generation

Figure 4-3 shows the minimum requirement (Power + Reset + Vcore) with suggested values for MSPM0 hardware setup.

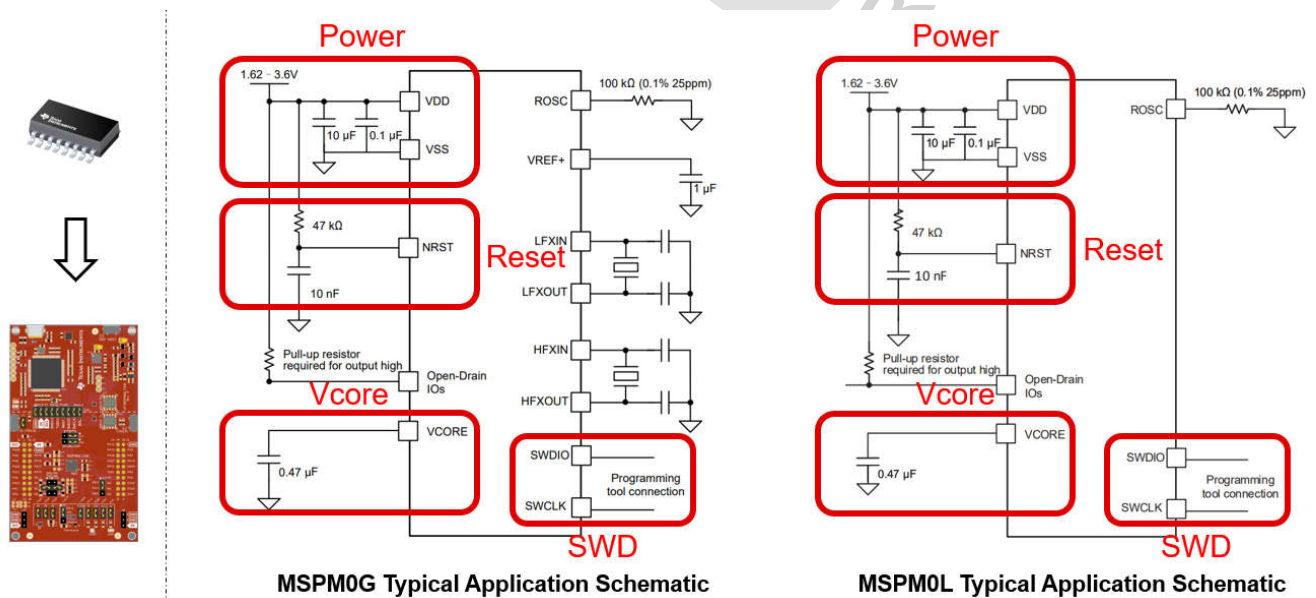


Figure 4-3. MSPM0 Minimum System

Figure 4-4 shows other attentions when designing the schematic in [Unable to auto-generate link text](#).

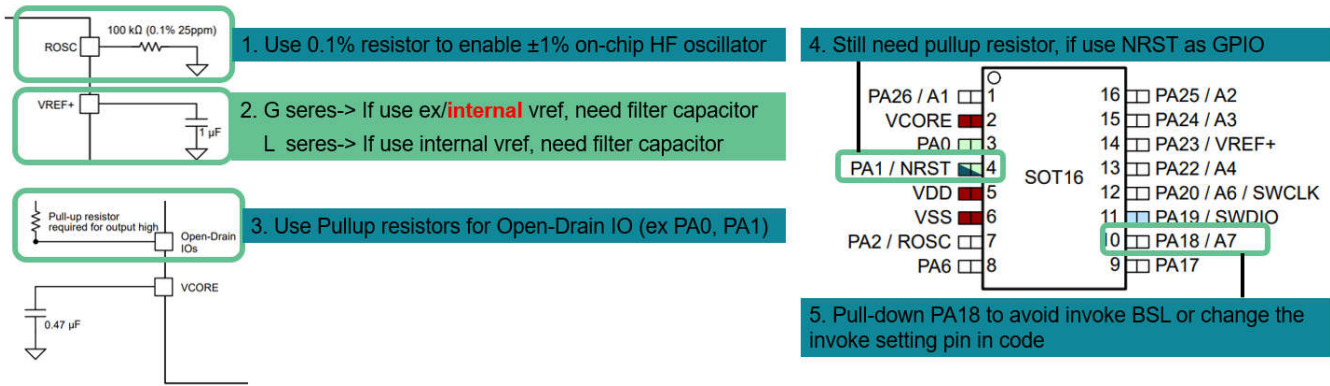


Figure 4-4. MSPM0 Schematic Other Attentions

For further schematic or PCB design reference, see:

- [MSPM0 L-Series MCUs Hardware Development Guide](#)
- [MSPM0 G-Series MCUs Hardware Development Guide](#)
- Device-specific MSPM0's Launchpad EVM user's guide
- Device-specific MSPM0's Data Sheet

5 Step 4: Mass Production

5.1 Generate Production Files

Figure 5-1 illustrates the steps to generate production files (.bin, .txt, .TI_TXT) based on CCS.

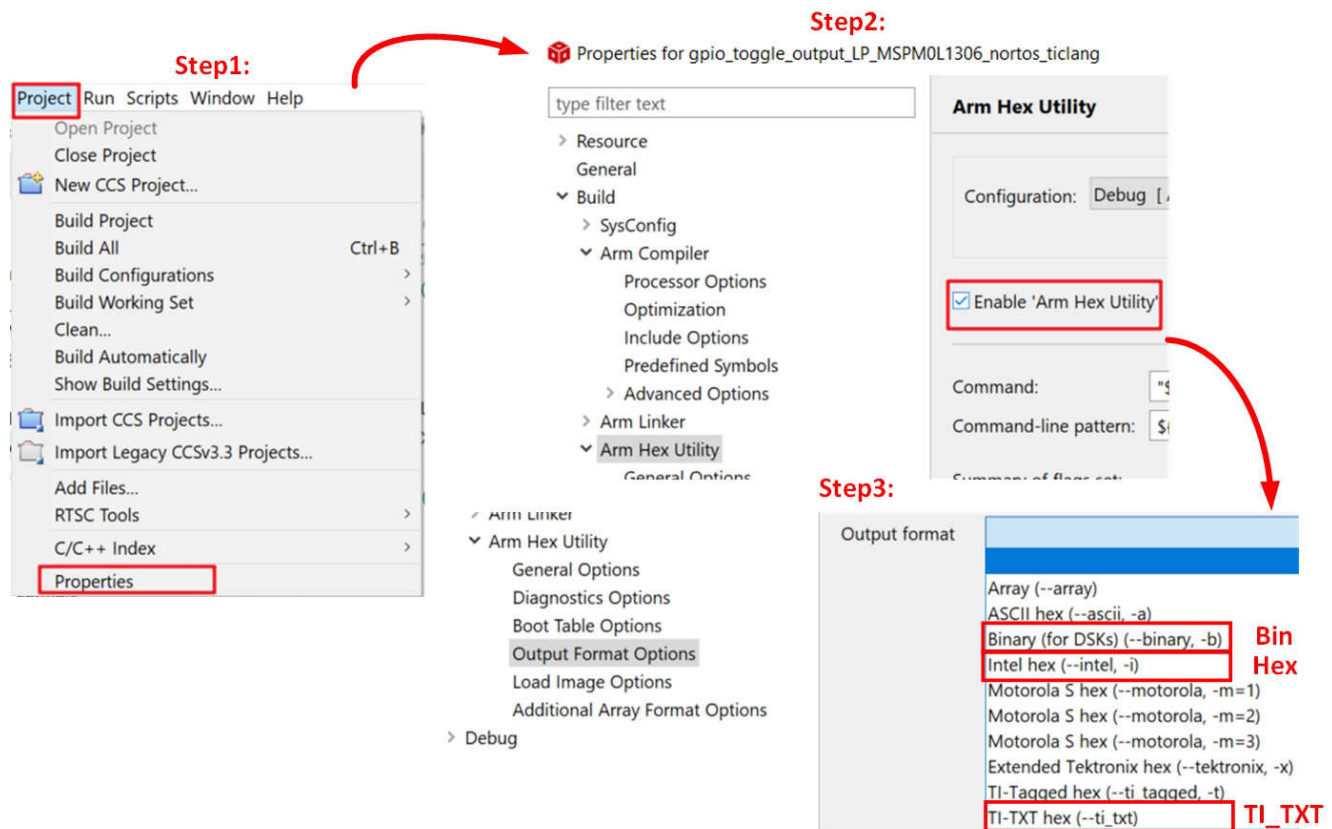


Figure 5-1. Create Program Files

5.2 Program Software and Tools Selection

Figure 5-2 shows an overview of the program software and tools. The available interface is JTAG (SWD) and Bootloader (BSL).

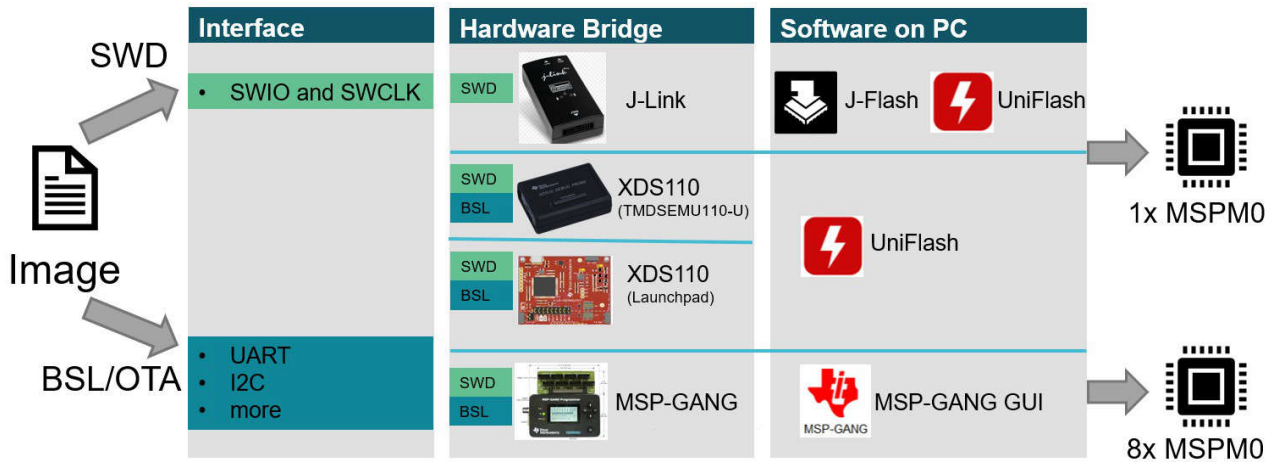


Figure 5-2. Program Software and Tool

For J-Link, it only support SWD. You can use J-Flash or UniFlash to pair with it to download code.

For standalone XDS110, it is TI own hardware bridge. Here is the used pin for you to refer. It only supports UniFlash.

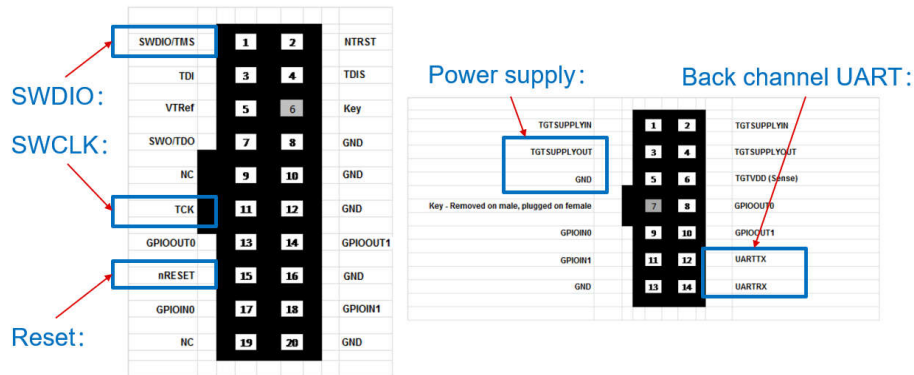


Figure 5-3. Pin Connection of Standalone XDS110

For XDS110 on lauchpad as shown in Figure 5-4, it keeps the basic programming function compared with standalone XDS110. That means you can also use it for mass production.

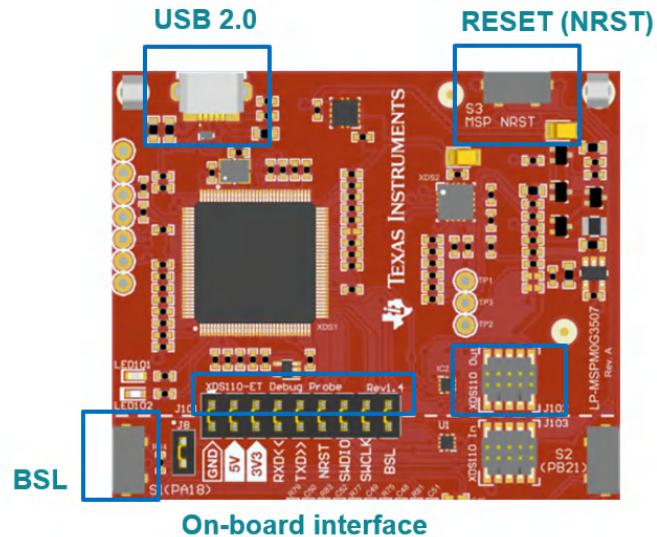


Figure 5-4. XDS110 on Board

MSP-GANG is a production solution, combining a GUI and a hardware bridge. It can program up to eight MSP devices at one time without PC GUI control. You can refer to [Section 6.3](#) for how to use it. Here is the user guide link for further reference.

- [MSP-GANG product page](#)

For Uniflash, it is TI own program tools. It is a standalone tool used to program on-chip flash memory on TI MCUs. Here are the references:

- Uniflash chapter in [MSPM0 Tools Guide](#)

6 Appendix

6.1 Light a LED and CCS Quick Introduction

This sections shows how to light an LED based on CCS at beginning. A short description to CCS is also provided, in order to let you ramp up with the tool as soon as possible.

6.1.1 Install CCS and SDK

Here are the important steps and tips for CCS installation.

1. Download [CCS \(above 12.2 version\)](#) and start installation, and keep pressing next.

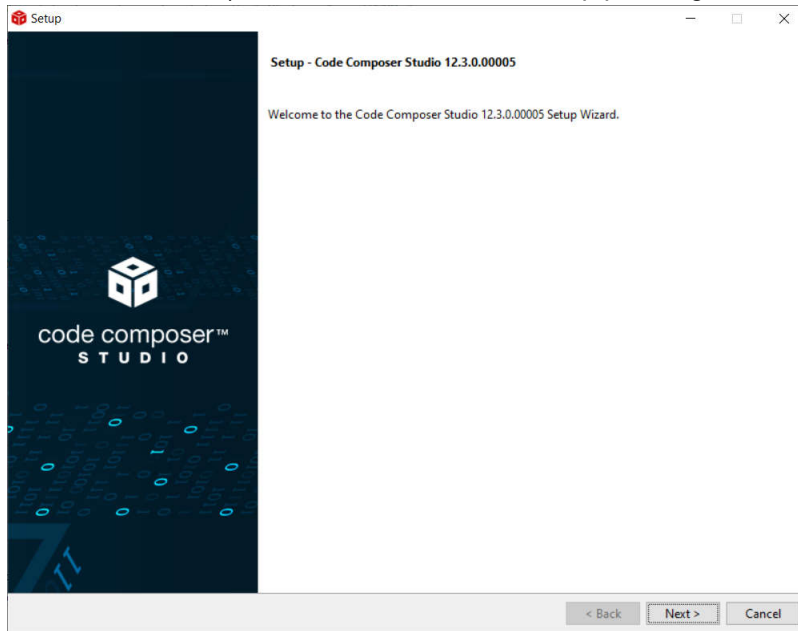


Figure 6-1. CCS Installation

2. Select MSPM0 support component.

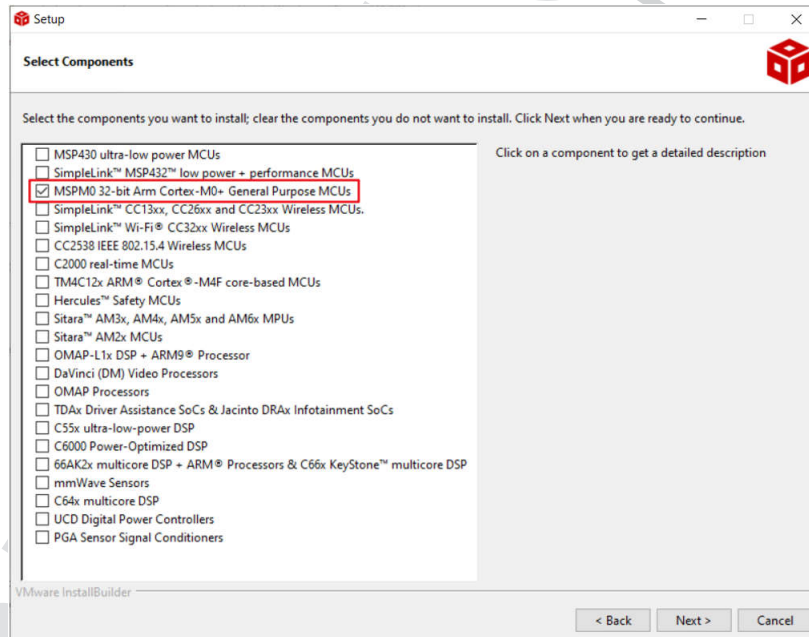


Figure 6-2. MSPM0 Support Selection

3. Select J-link if needed.

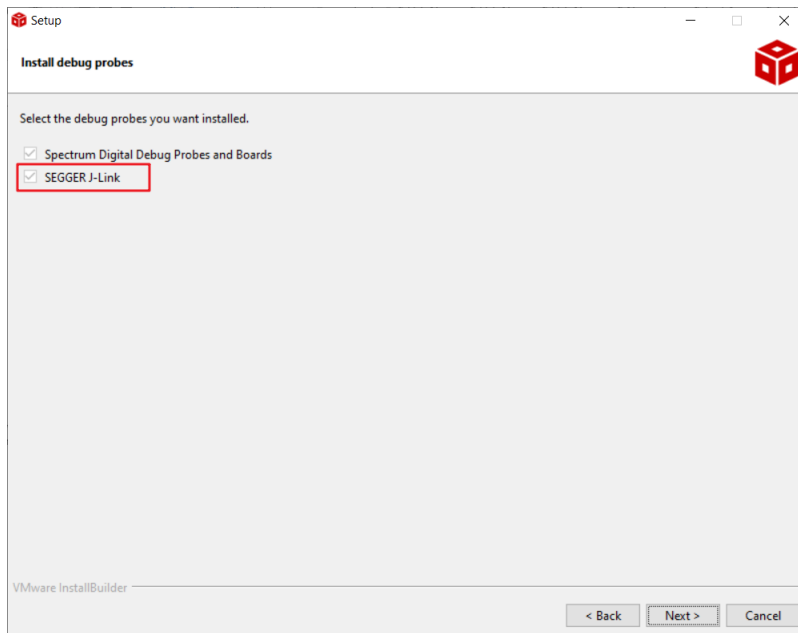


Figure 6-3. J-link selection

4. Install MSPM0 SDK.

6.1.2 Hardware Setup

Get a launchpad and plug in the computer.

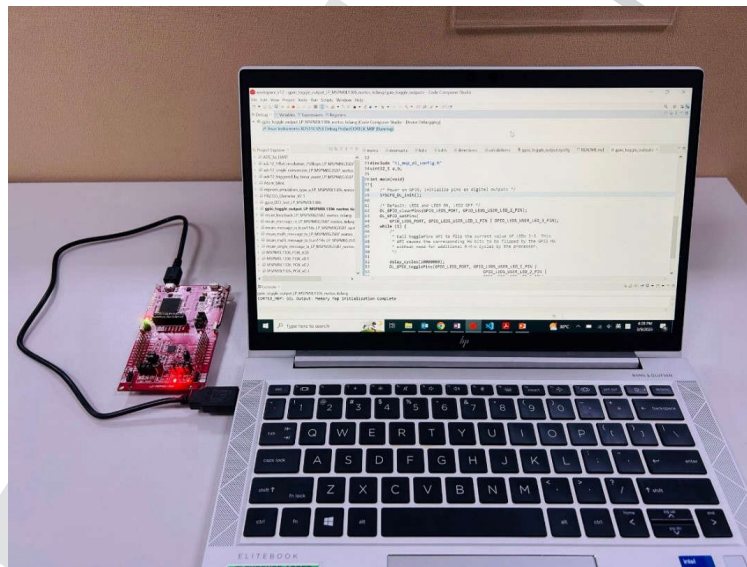


Figure 6-4. Hardware Setup

6.1.3 Code Import

1. Open CCS. The workspace means the address where to copy your imported project.

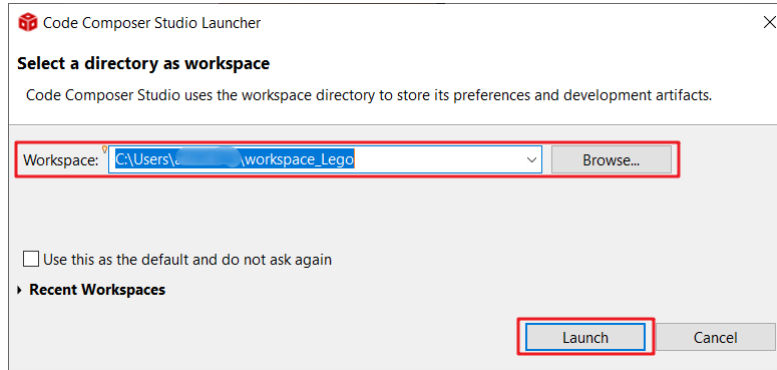


Figure 6-5. Choose CCS Workspace

2. Import the general-purpose input/output (GPIO) toggle project with the TI-Clang compiler.

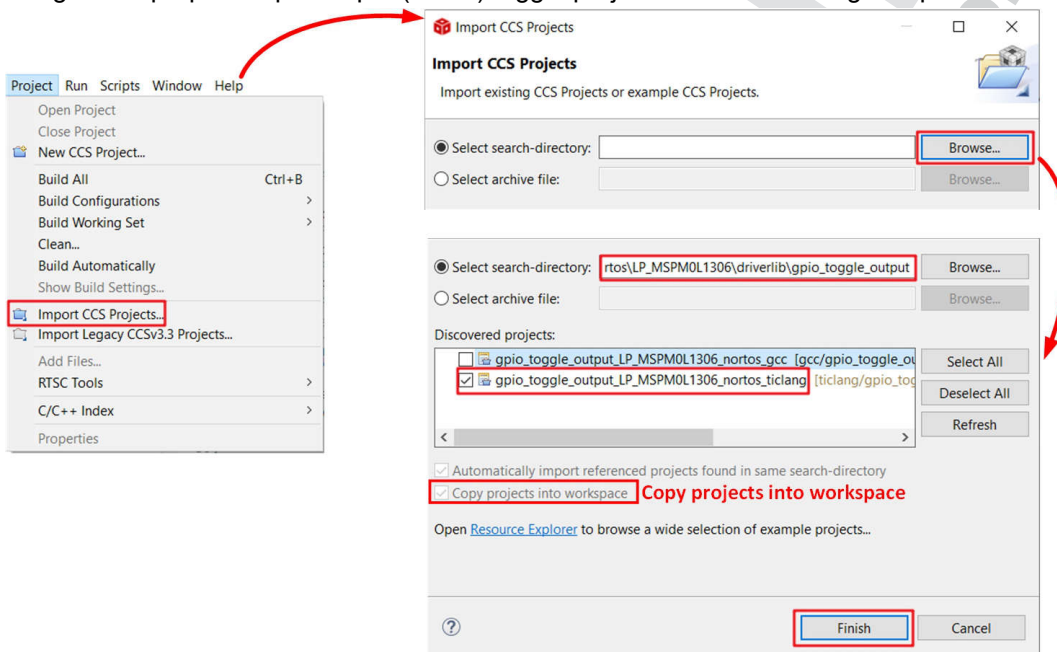


Figure 6-6. Import Project

3. If it cannot be imported, delete the same name project under workspace.

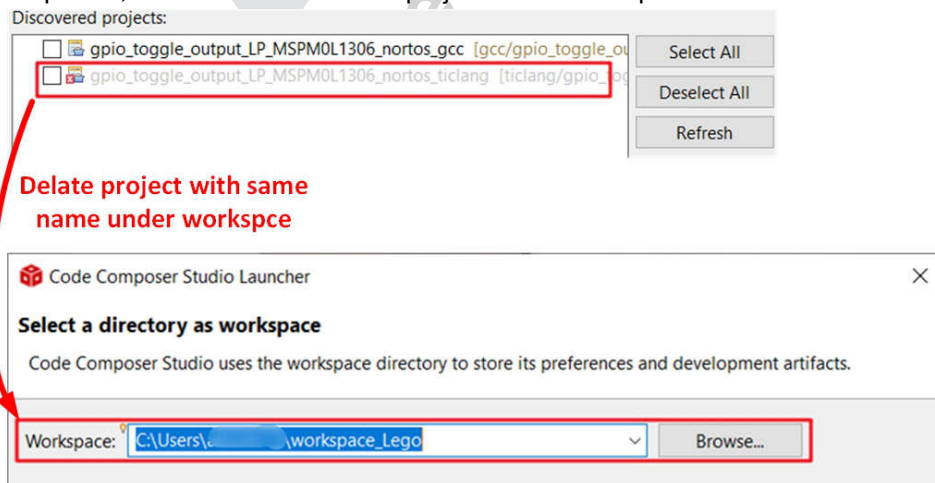


Figure 6-7. Remove Duplicated Project

6.1.4 Debug and CCS Quick Introduction

1. Start debug, then you can see GPIO toggle on the LP.

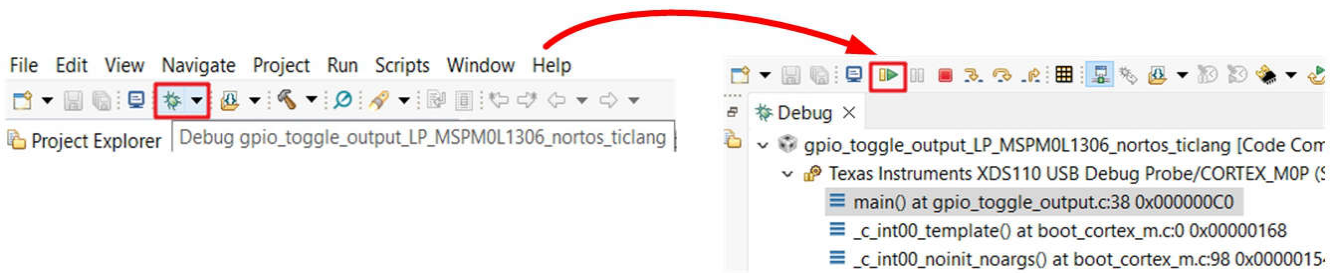


Figure 6-8. Debug Code

2. Here we give a quick introduction to CCS functions.

a. Project properties common used settings:

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

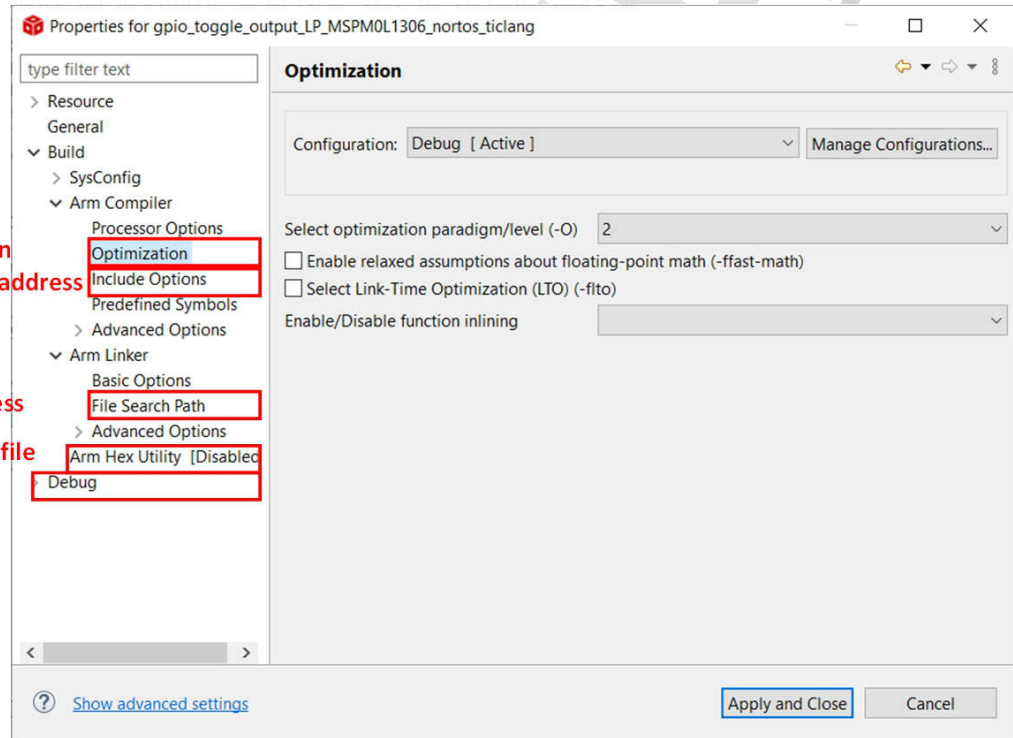


Figure 6-9. Common Used Project Settings

b. Debug common used functions.

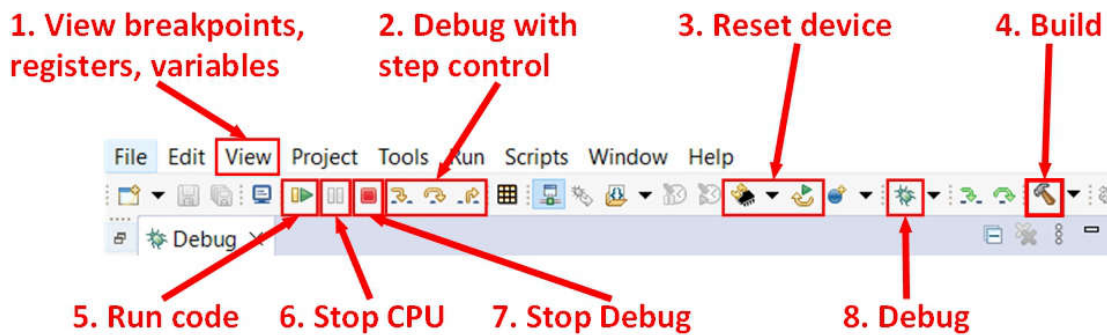


Figure 6-10. Common Used Debug Functions

6.2 Steps to Generate PCB Library

1. Go to the entrance of Ultra Librarian tool under the MSPM0 device page using the steps shown in Figure 6-11.

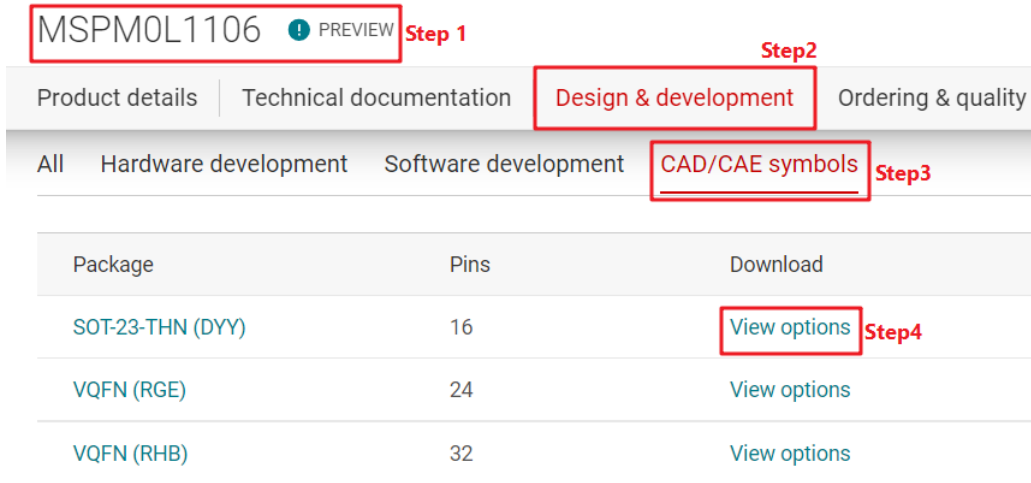


Figure 6-11. Ultra Librarian Tool Entrance

2. Select your wanted CAD format and Pin ordering, then you can get the Altium design lib file.

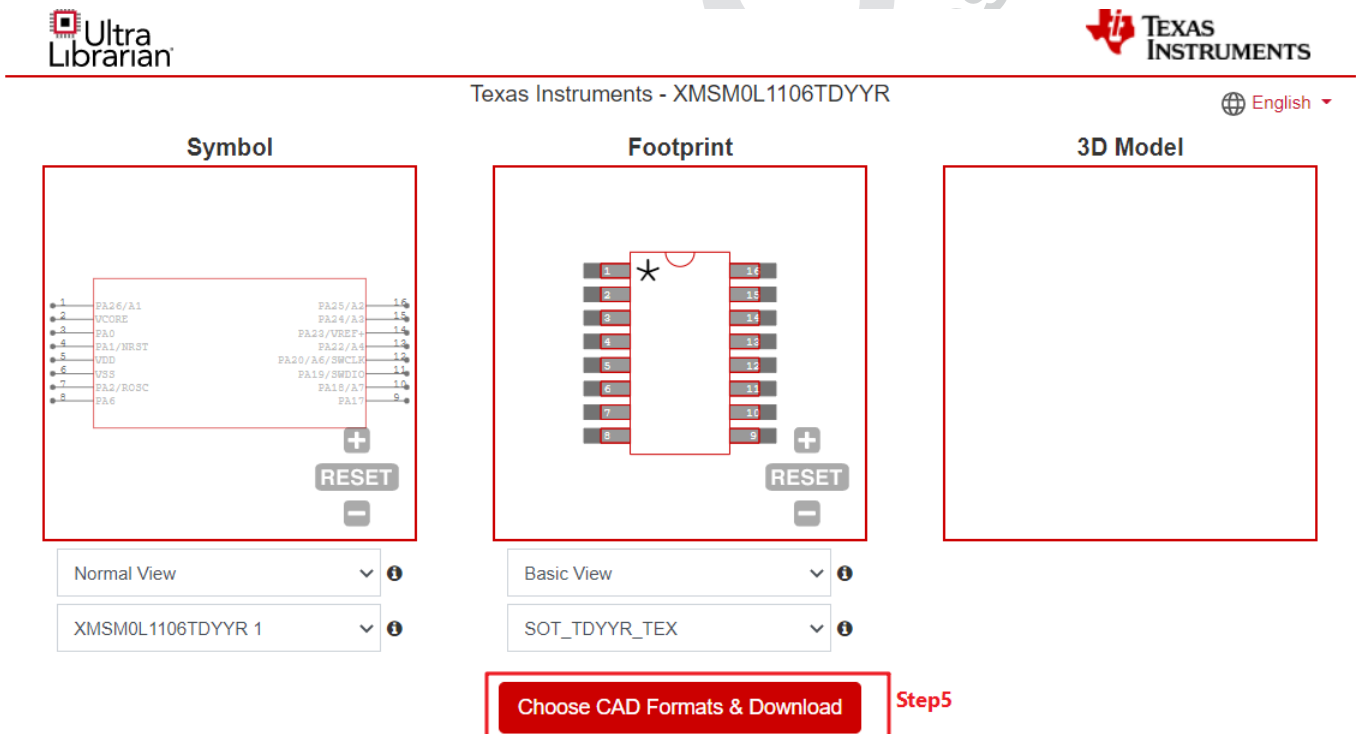


Figure 6-12. Ultra Librarian Tool Device Selection

3. Here, Altium Designer lib is used as an example.

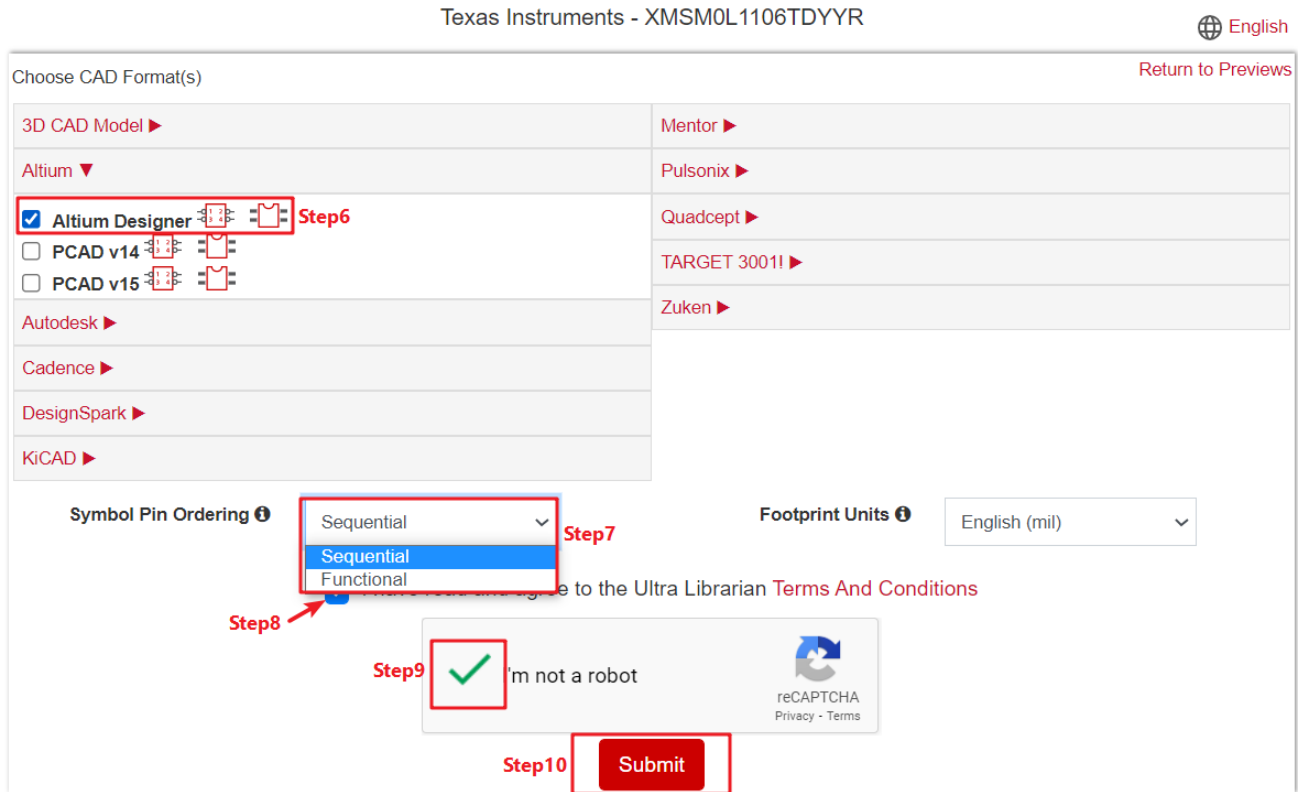


Figure 6-13. Ultra Librarian Tool CAD Download

4. Run Altium Designer script as shown in Figure 6-14.

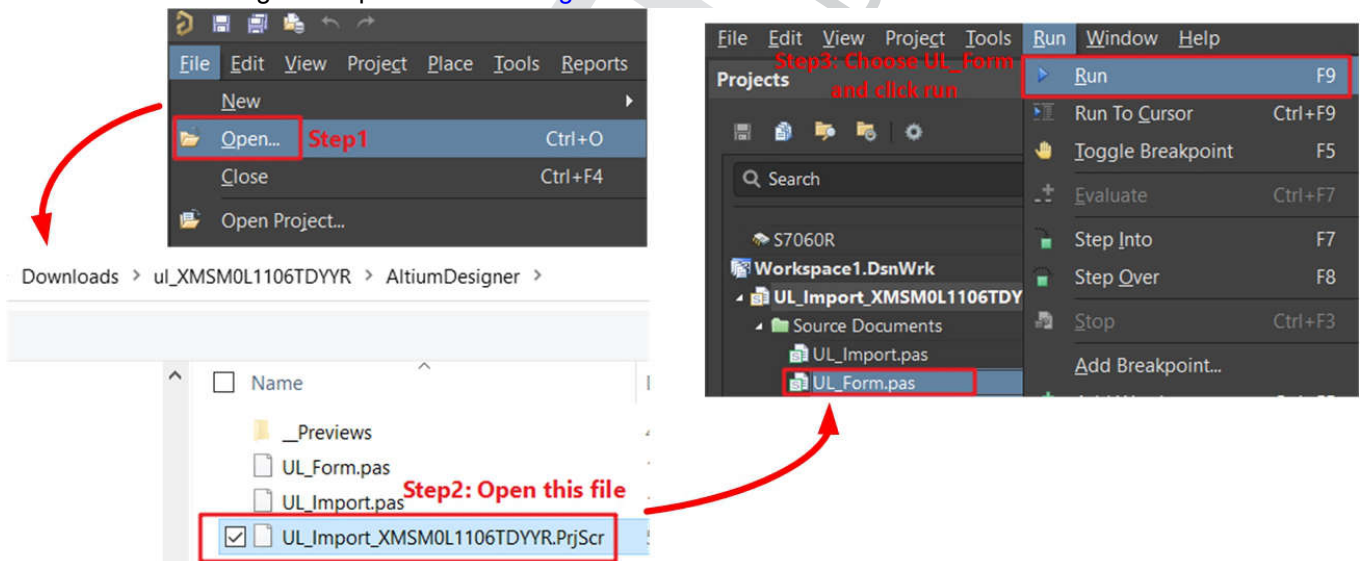


Figure 6-14. Run Altium Designer Script

5. Generate PCB library and schematic library as shown in Figure 6-15.

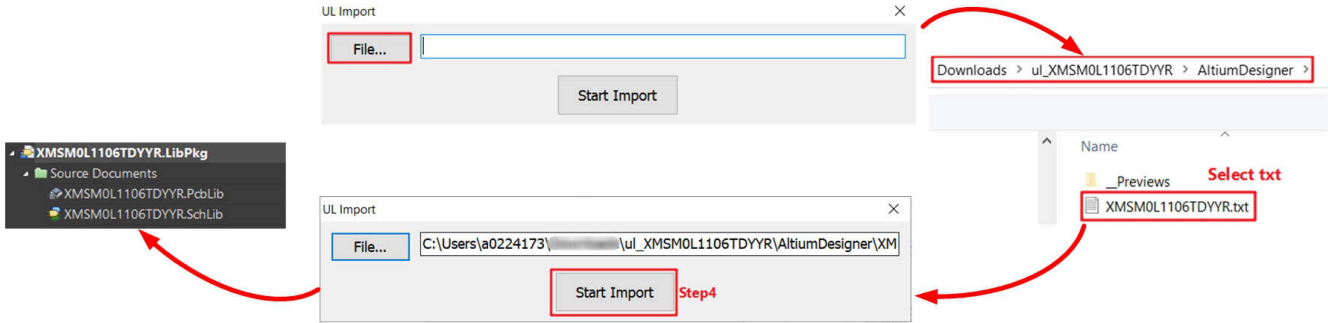


Figure 6-15. Generate Library

6. Choose suitable footprint under PCB Library.

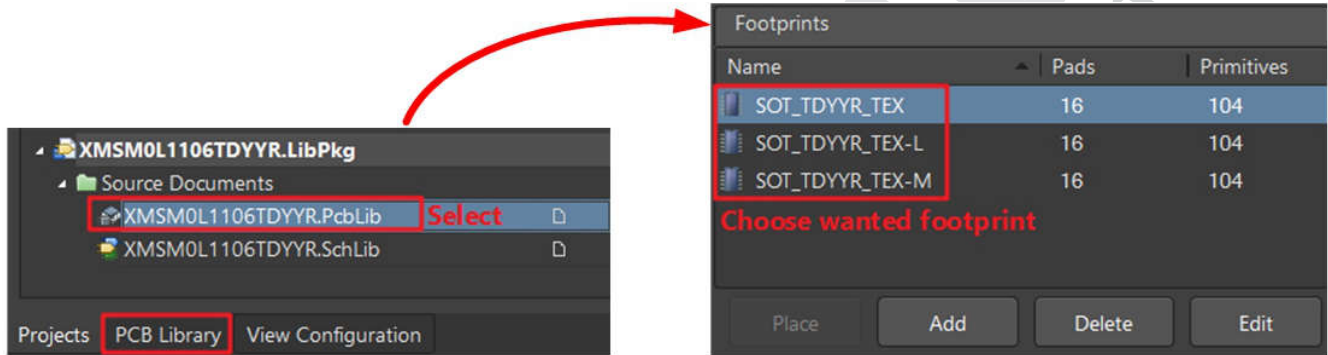


Figure 6-16. Select Footprint

7. Import PCB library and schematic library.

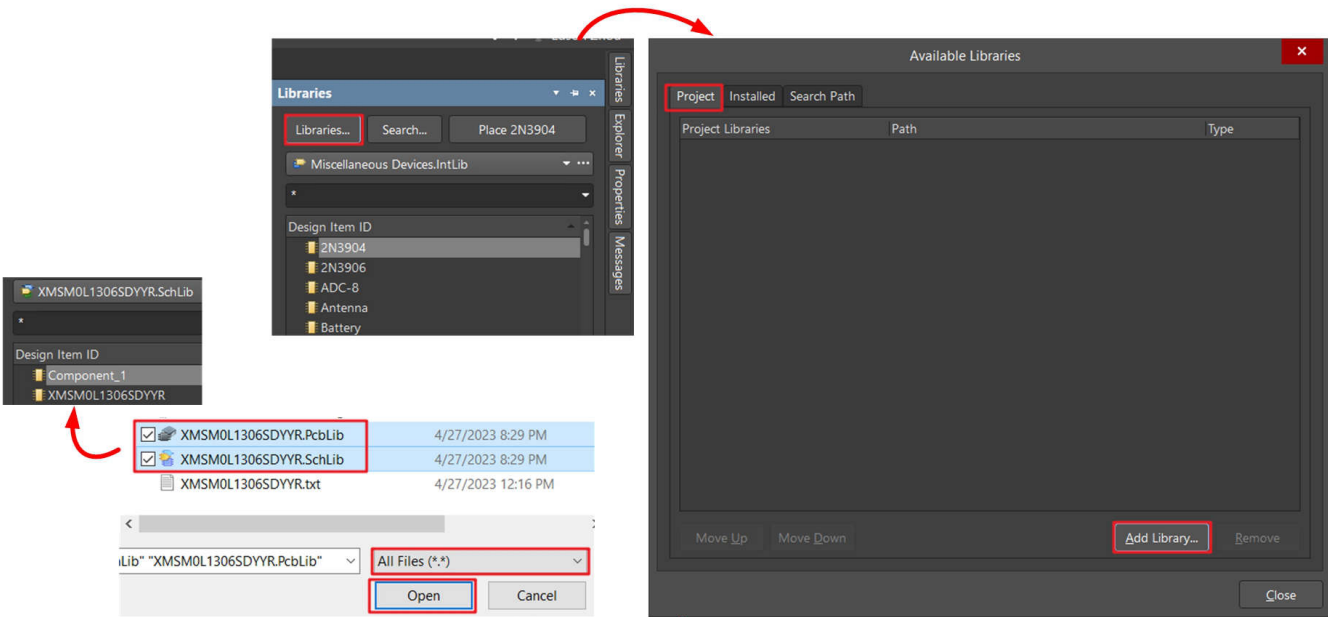


Figure 6-17. Import Library

6.3 MSP-GANG Quick Introduction

In this part we will show how to use MSP-GANG to offline program MSPM0. At the beginning we will show how to use MSP-GANG with its GUI to program a MSPM0.

1. First you need to finish the pin connection used for SWD. Here is a quick reference.

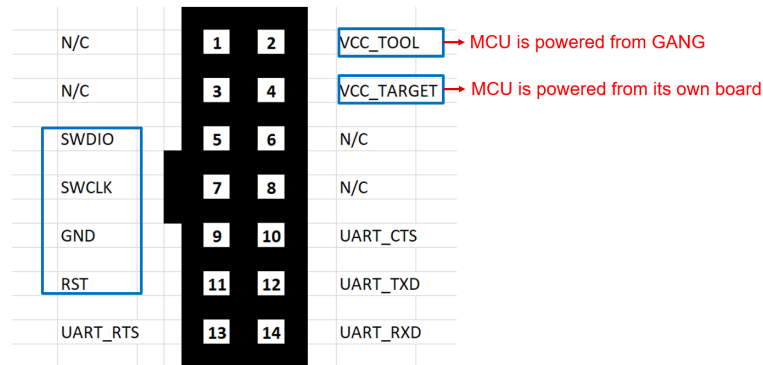


Figure 6-18. MSP-GANG Pin Assignment

After you finish the hardware setup, you can follow the steps to do programming. For Step2, you can refer to Section 5.1 to generate the code file. For Step4, the enable target is related to the hardware port used, which is signed in number near the port.

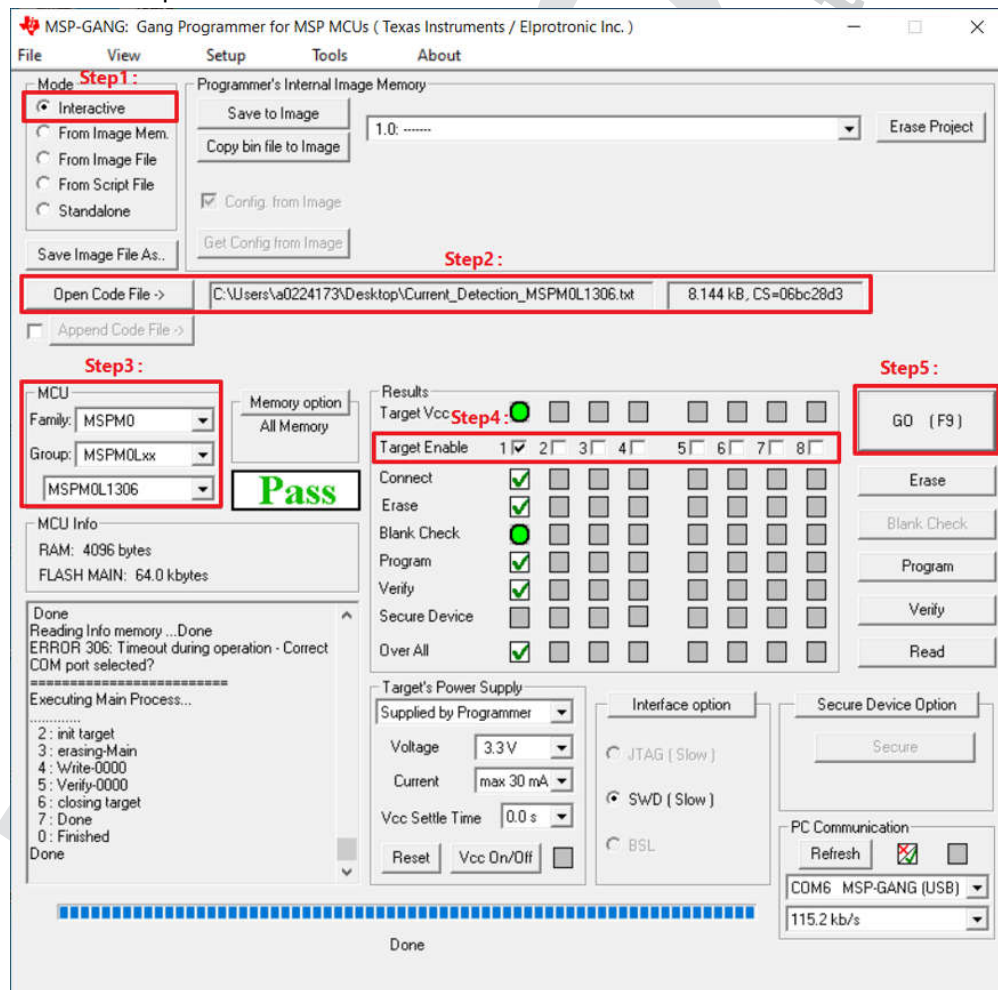


Figure 6-19. Download Code Using MSP-GANG with GUI

1. If your code file need to change Non-main (SWD and BSL configure flash area), you need to enable this function first.

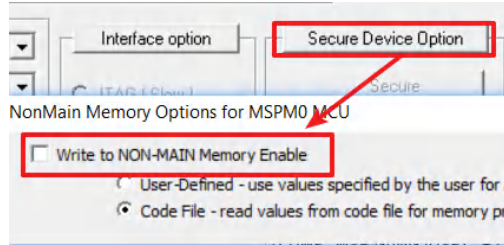


Figure 6-20. Enable Non-main Programming

Next, we will show how to use MSP-GANG to program a MSPM0 in stand alone mode.

1. First you need to save the code file and settings into MSP-GANG. You can give a project name to this image. Then click "Save to Image".

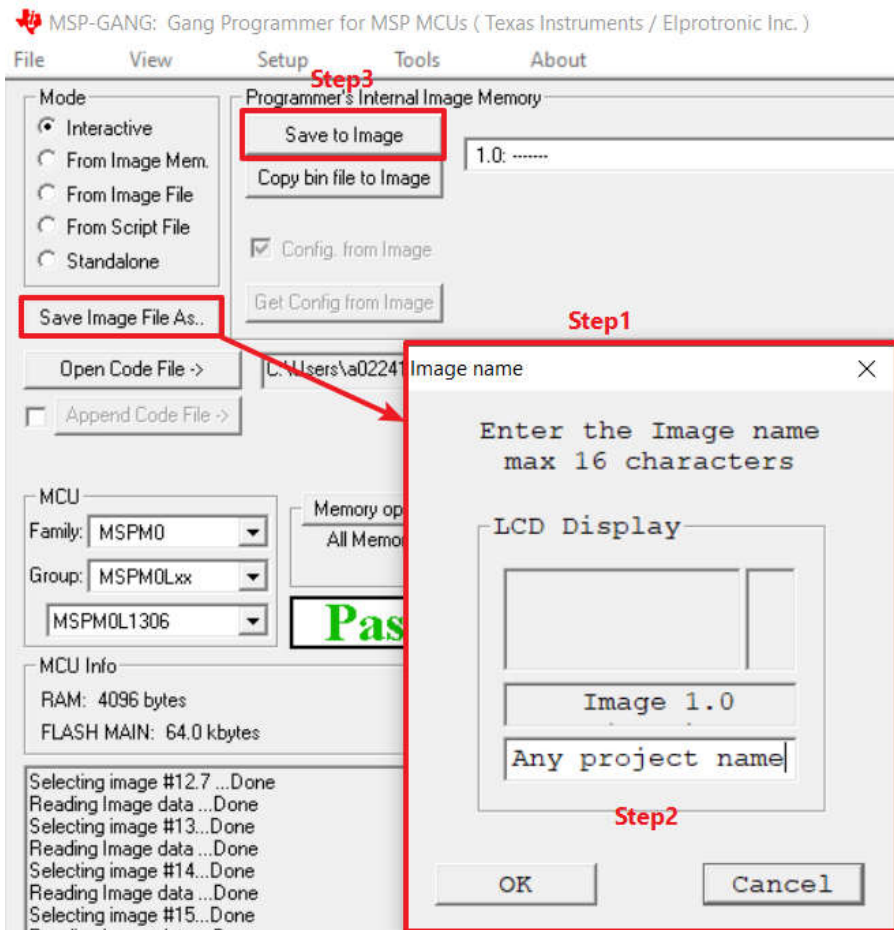


Figure 6-21. Generate and Save Image

2. Then you can change the mode to standalone or directly close the GUI.

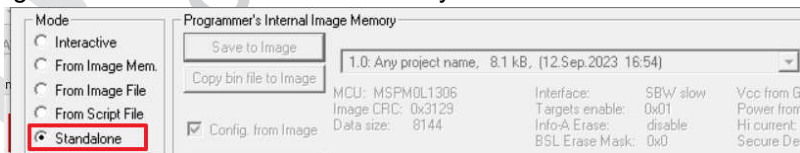


Figure 6-22. Change Mode

3. As we only save one image into the MSP-GANG. You can directly click "Go" to do the programming. If you have more than one image, you need to switch to the right one first.

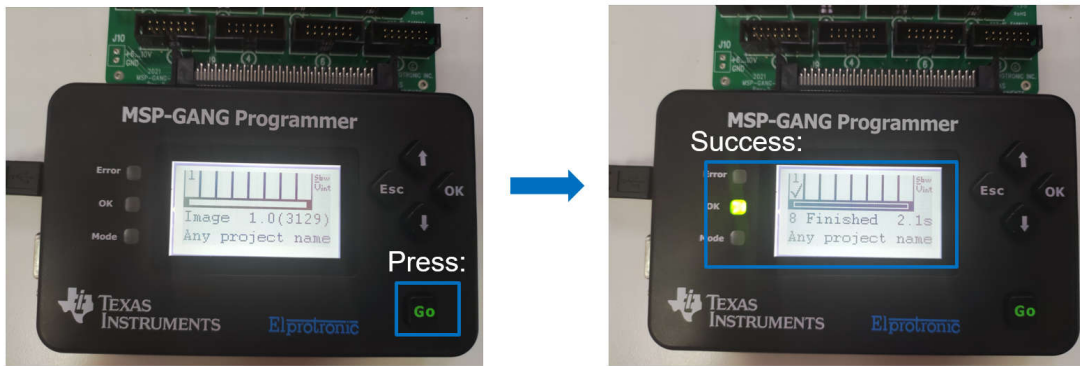


Figure 6-23. Offline Programming

Revision History

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

Changes from Revision * (May 2023) to Revision A (June 2023)

	Page
• Updated the numbering format for tables, figures and cross-references throughout the document.....	2
• Updated Section 3.1.1	6
• Updated Section 3.1.2	6
• Updated Section 3.2.2	7
• Updated Section 3.3.2	10

DRAFT ONLY

TI Confidential – NDA Restrictions

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](https://www.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 © 2023, Texas Instruments Incorporated

DRAFT
TI Confidential – NDA Restrictions