




# Gowin\_EMPU\_M3 IDE Software **Reference Design**

IPUG919-1.1E, 07/16/2021

**Copyright © 2021 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.**

**GOWIN**, , Gowin, and GOWINSEMI are trademarks of Guangdong Gowin Semiconductor Corporation and are registered in China, the U.S. Patent and Trademark Office, and other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of GOWINSEMI.

#### **Disclaimer**

GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. All information in this document should be treated as preliminary. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

### Revision History

Date	Version	Description
04/03/2020	1.0E	Initial version published.
07/16/2021	1.1E	The version of MCU software updated.

# Contents

<b>Contents .....</b>	<b>i</b>
<b>List of Figures .....</b>	<b>ii</b>
<b>1 ARM Keil MDK .....</b>	<b>1</b>
1.1 Software Installation .....	1
1.2 Project Template .....	1
1.2.1 Create a New Project .....	1
1.2.2 Configuration Options .....	2
1.2.3 Build .....	7
1.2.4 Download .....	8
1.2.5 Software Debugging .....	9
1.3 Reference Design .....	9
<b>2 GOWIN MCU Designer .....</b>	<b>10</b>
2.1 Software Installation .....	10
2.2 Project Template .....	10
2.2.1 Create a New Project .....	10
2.2.2 Configuration Options .....	12
2.2.3 Build .....	16
2.2.4 Download .....	16
2.2.5 Debug .....	17
2.3 Reference Design .....	19

# List of Figures

Figure 1-1 Create a New Project .....	1
Figure 1-2 Device Configuration .....	2
Figure 1-3 ROM and RAM Configuration .....	3
Figure 1-4 Output File Format Configuration .....	3
Figure 1-5 C Header File Path Configuration .....	4
Figure 1-6 Configure Emulator Type .....	5
Figure 1-7 Debugging Interface Type Configuration .....	6
Figure 1-8 Flash Configuration .....	6
Figure 1-9 Debugging Initialization File Configuration .....	7
Figure 1-10 Build .....	7
Figure 1-11 Download .....	8
Figure 1-12 Start Debugging .....	9
Figure 2-1 Create a New Project .....	11
Figure 2-2 Select Platform and Configuration .....	11
Figure 2-3 Select Configuration Path and Toolchain .....	12
Figure 2-4 Target Processor Configuration .....	13
Figure 2-5 GNU ARM Cross Assembler Configuration .....	13
Figure 2-6 GNU ARM Cross C Compiler Configuration .....	14
Figure 2-7 GNU ARM Cross C Linker Configuration .....	14
Figure 2-8 GNU ARM Cross Create Flash Image Configuration .....	15
Figure 2-9 Devices Configuration .....	15
Figure 2-10 Build .....	16
Figure 2-11 Download .....	17
Figure 2-12 Create Debugging Configuration Options .....	17
Figure 2-13 Main Configuration .....	18
Figure 2-14 Debugger Configuration .....	18
Figure 2-15 Start Debugging .....	19

# 1 ARM Keil MDK

## 1.1 Software Installation

For the detailed information, please refer to [Getting Started with MDK](#) provided by ARM Keil MDK website

**Note!**

It is recommended ARM Keil MDK V5.26 and above.

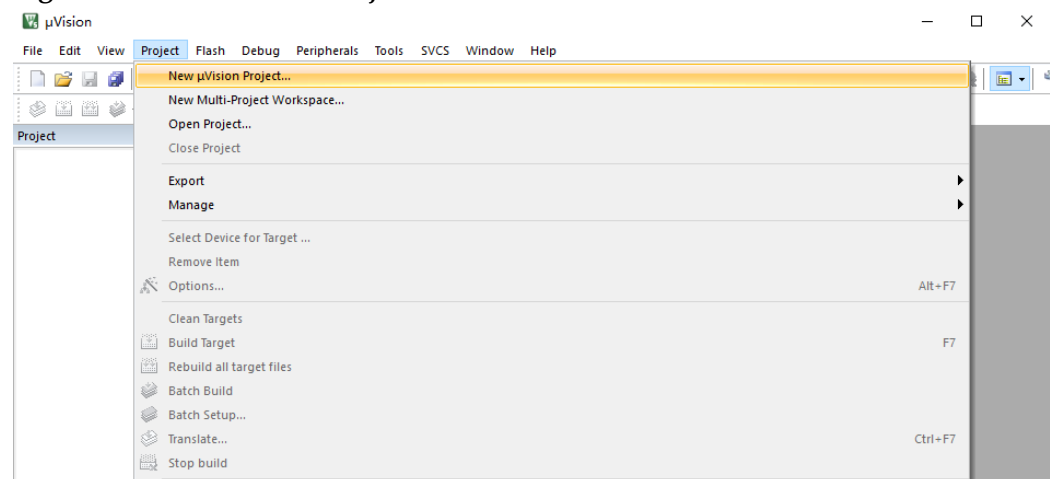
## 1.2 Project Template

ARM Keil MDK can be used for Gowin\_EMPU\_M1 software programming. The steps include project creation, configuration, coding, compilation, and debugging.

### 1.2.1 Create a New Project

Open ARM Keil MDK and select "Project > New uVision Project..." to create a new project, as shown in Figure 1-1.

**Figure 1-1 Create a New Project**

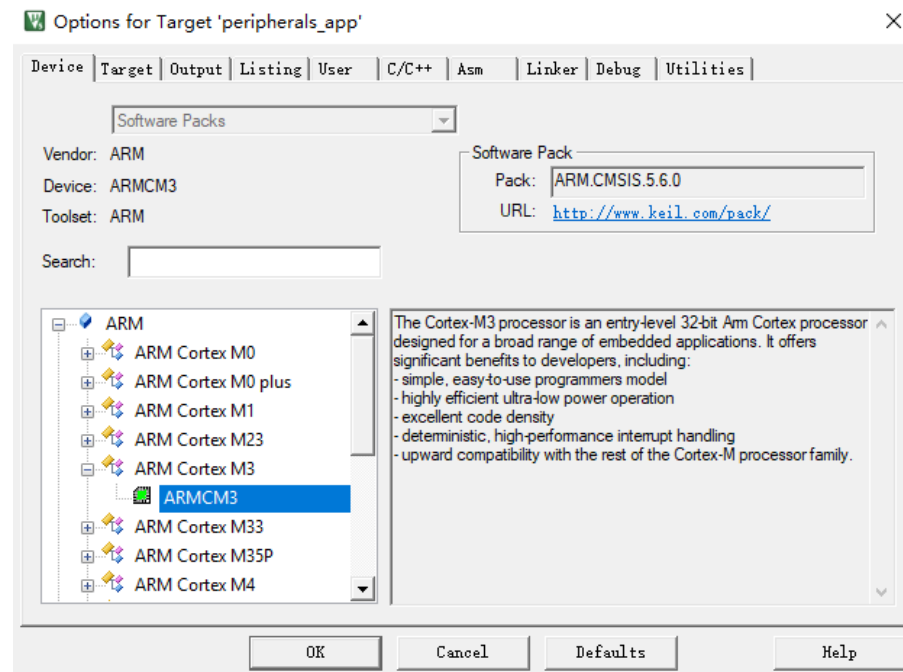


## 1.2.2 Configuration Options

### Device Configuration

ARM Cortex-M3 is embedded in Gowin\_EMPU\_M3. Select "ARM Cortex M3 > ARMCM3", as shown in Figure 1-2.

Figure 1-2 Device Configuration



### ROM and RAM Configuration

Instruction Memory of Gowin\_EMPU\_M3 is used as ROM.

Data Memory of Gowin\_EMPU\_M3 is used as RAM.

Configure the start address and size of ROM and RAM, as shown in Figure 1-3.

#### ROM Configuration

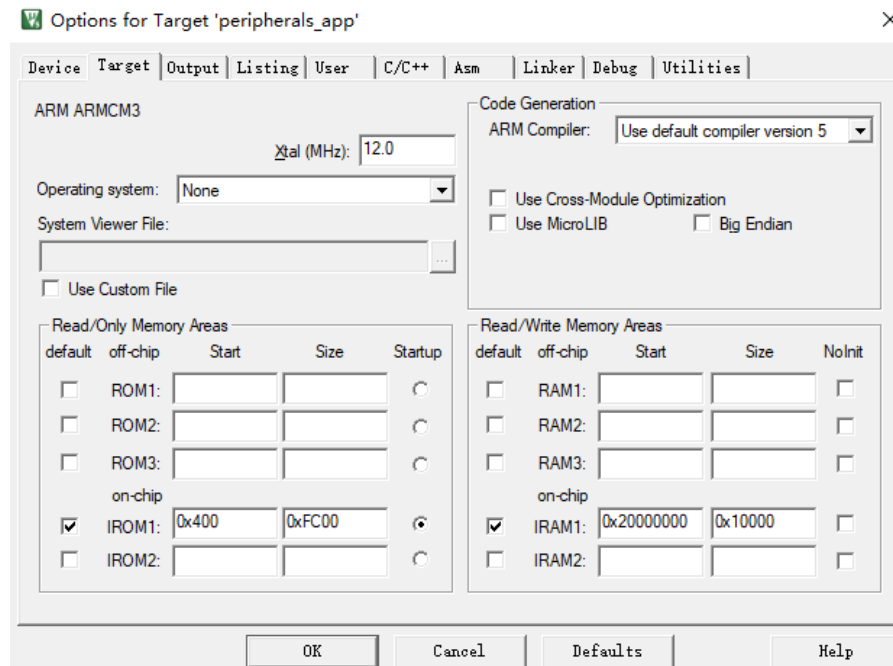
- Off-chip SPI-Flash boot mode is selected and the start address is 0x400.
- The size can be configured as 16KB, 32KB, 64KB, or 128KB according to the configuration of Gowin IP Core Generator > Gowin\_EMPU\_M3 > Instruction Memory Size.
- Take Keil\_RefDesign reference design in SDK as an example. ROM size is configured as 0xFC00 (Gowin IP Core Generator > Gowin\_EMPU\_M3 > Instruction Memory Size is configured as 64KB).

#### RAM Configuration

- The start address of RAM is 0x20000000.
- The size can be configured as 16KB, 32KB, 64KB, or 128KB according to the configuration of Gowin IP Core Generator > Gowin\_EMPU\_M3 > Data Memory Size.
- Take Keil\_RefDesign reference design in SDK as an example. RAM

size is configured as 0x10000 (Gowin IP Core Generator > Gowin\_EMPU\_M3 > Data Memory Size is configured as 64KB).

**Figure 1-3 ROM and RAM Configuration**

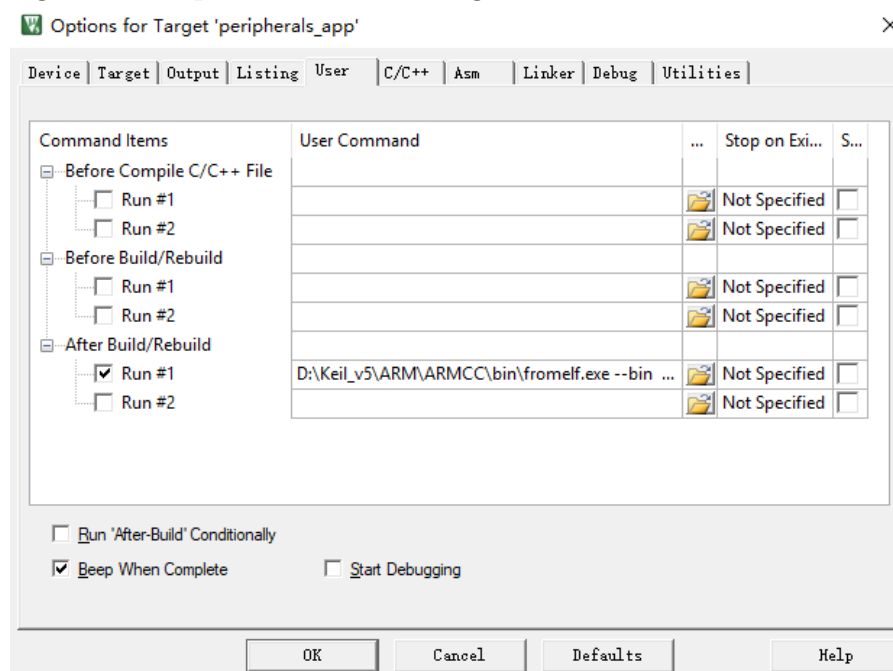


## Output File Format Configuration

Gowin Programmer supports BIN files downloading. Configure the output file format as BIN.

- The call method of file format conversion tool in the User option is as shown in Figure 1-4.
- Format conversion command: Run #1: fromelf.exe --bin -o bin-file axf-file

**Figure 1-4 Output File Format Configuration**





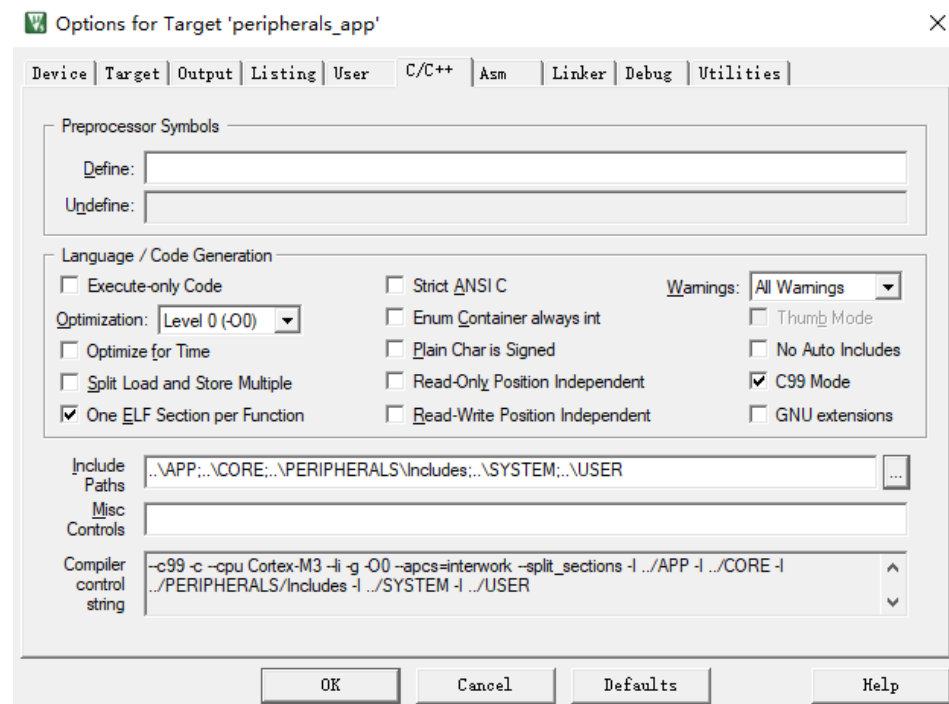
## Header File Path Configuration

Configure C header file path and different header file C paths are called during building as shown in Figure 1-5.

Take Keil\_RefDesign reference design in SDK for an instance, the the C header file paths are listed as follows.

- "..\CORE"
- "..\PERIPHERALS\Includes"
- "..\SYSTEM"
- "..\USER"
- "..\APP"

**Figure 1-5 C Header File Path Configuration**

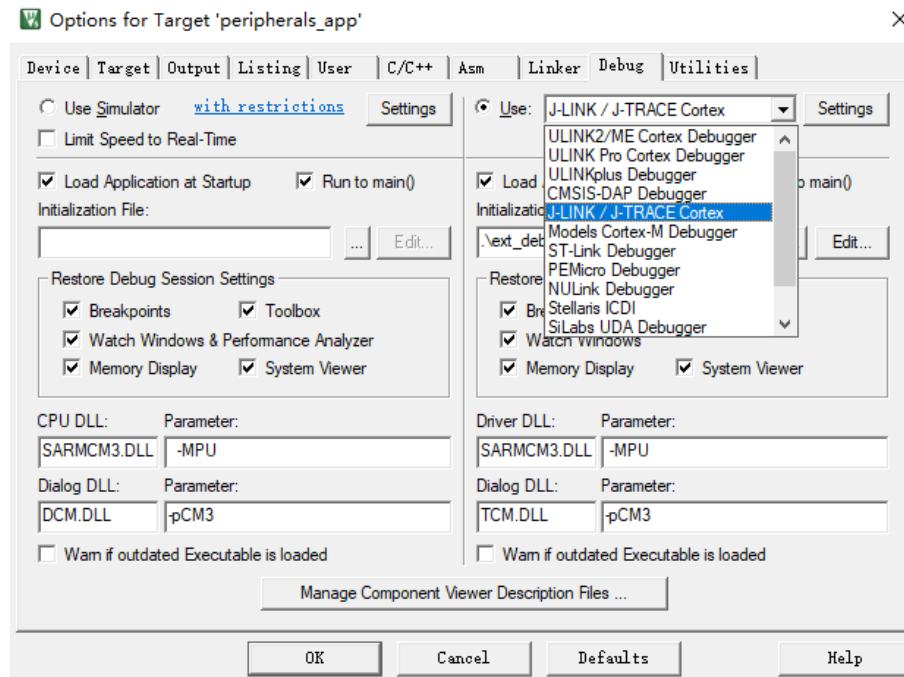


## Configure Debug Options

- Configure the Emulator

Click the Debug emulator drop-down list and select the type, as shown in Figure 1-6.

- U-LINK Emulator  
If the U-LINK emulator is selected, use ULNK2/ME Cortex Debugger.
- J-LINK Emulator  
If the J-LINK emulator is selected, use J-LINK/J-TRACE Cortex.

**Figure 1-6 Emulator Type Configuration**

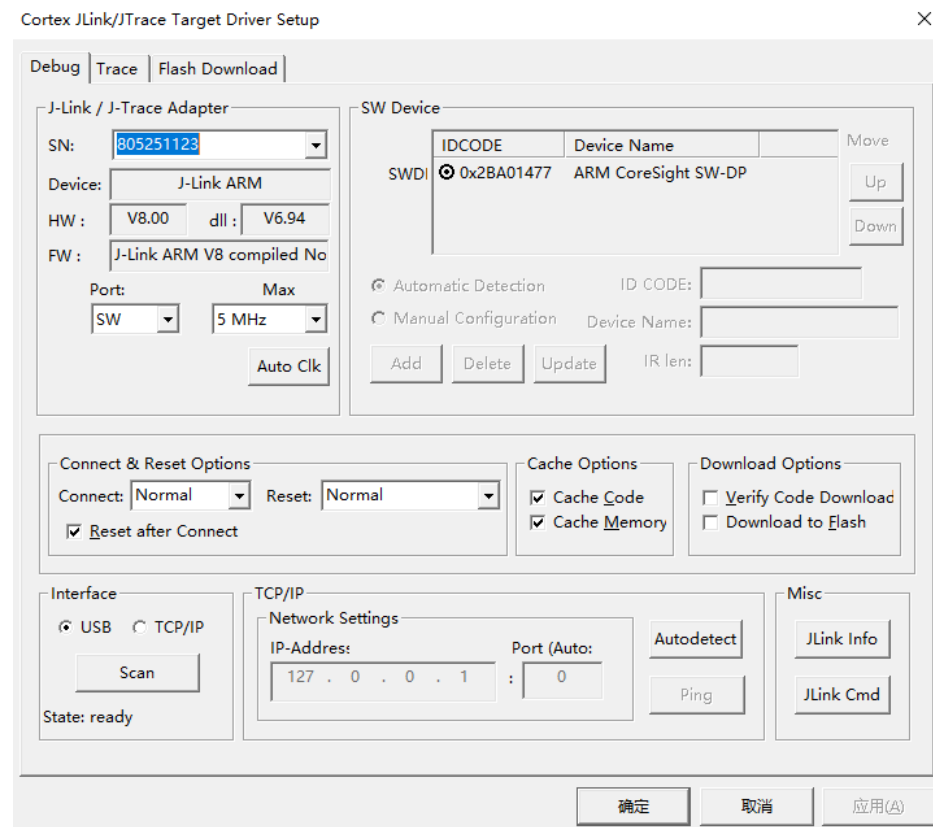
- **Configure Debug Interface**

For example, select J-LINK/J-TRACE Cortex. Click “Settings” to open Cortex JLink/JTrace Target Driver Setup and select the type of debug interface.

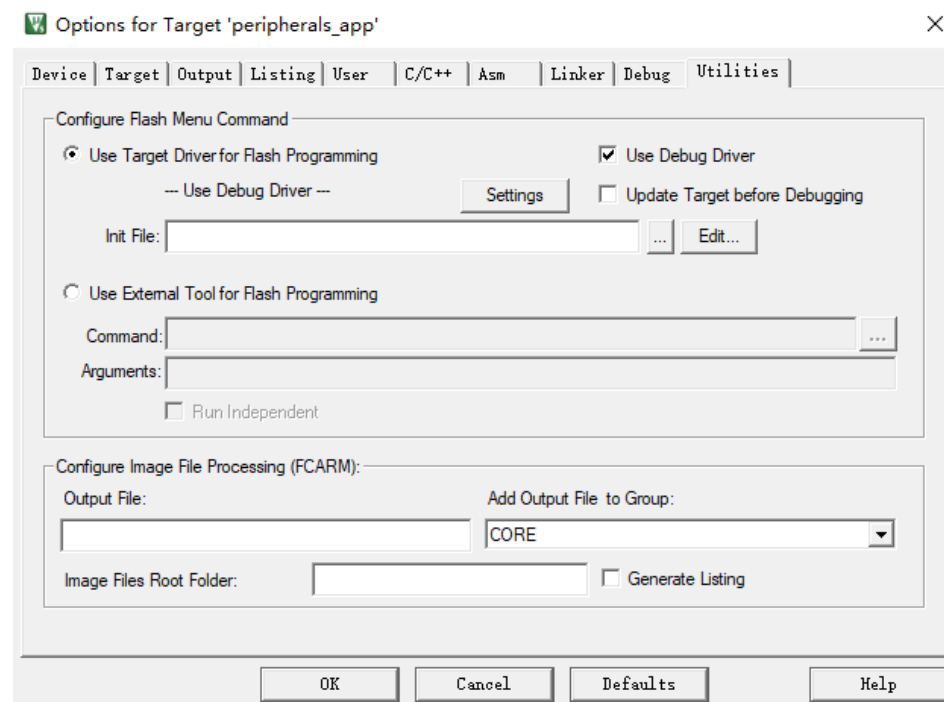
- JTAG interface  
If Gowin IP Core Generator > Gowin\_EMPU\_M3 > Debug Interface enables JTAG, then configure SWJ Port as JTAG.
- SW interface  
If Gowin IP Core Generator > Gowin\_EMPU\_M3 > Debug Interface disables JTAG, then configure SWJ Port as the SW interface.

Do not select "Verify Code Download" and "Download to Flash" in Download Options.

If the debug options are configured successfully, after connecting J-LINK or U-LINK emulator, the “JTAG Device Chain” can display the IDCODE, Device Name and other information of Gowin\_EMPU\_M3, as shown in Figure 1-7.

**Figure 1-7 Debugging Interface Type Configuration****Flash Configuration**

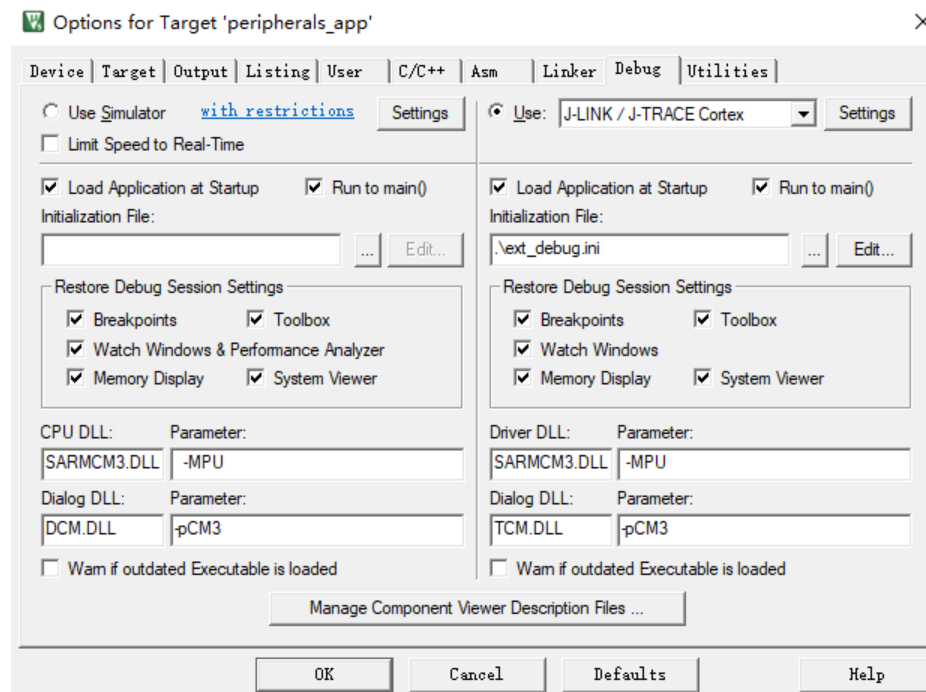
Do not select "Utilities > Update Target before Debugging" , as shown in Figure 1-8.

**Figure 1-8 Flash Configuration**

## Debugging Initialization File Configuration

When debugging the Gowin\_EMPU\_M3, load the debugging Initialization File. Click "Debug > Initialization File" to load "ext\_debug.ini" file, as shown in Figure 1-9.

Figure 1-9 Debugging Initialization File Configuration



## 1.2.3 Build



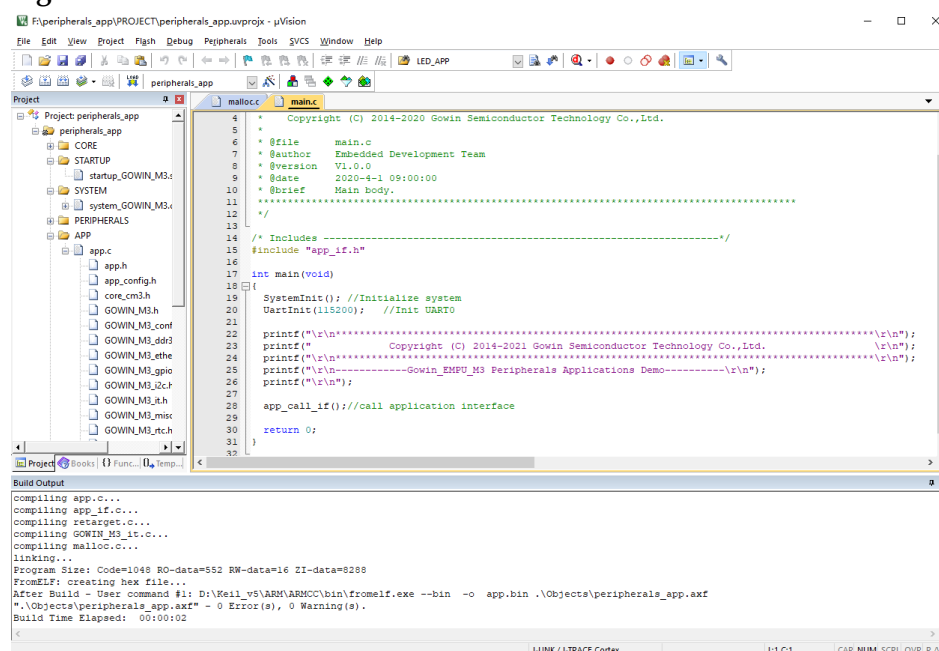
After encoding and configuration, click "Build" (  ) or "Rebuild" (  ) in tool bar to generate software programming BIN File in binary format, as shown in Figure 1-10.

Figure 1-10 Build



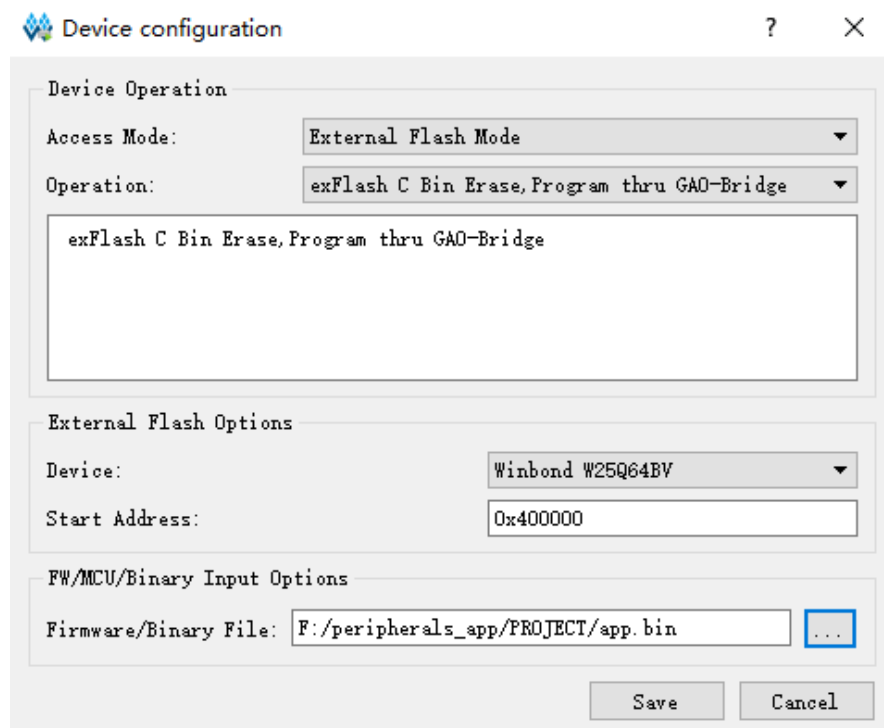
## 1.2.4 Download

After building, use Gowin Programmer, the download tool, to download the software programming BIN file in binary format.

Run Gowin Programmer, click "Edit > Configure Device" or Configure Device "🔧" in the tool bar to open the "Device configuration" dialog box.

- Select "External Flash Mode" in "Access Mode" drop-down list;
- Select "exFlash C Bin Erase, Program thru GAO-Bridge" or "exFlash C Bin Erase, Program, Verify thru GAO-Bridge" in "Operation" drop-down list.
- Select "FW/MCU Input Options > Firmware/Binary File" to import the software programming BIN File in binary format to download.
- Select based on the on-board Flash in "External Flash Options > Device" (such as Winbond W25Q64BV);
- Configure the start address as "0x400000" in "External Flash Options > Start Address".
- Click "Save" as shown in Figure 1-11.

Figure 1-11 Download



After device configuration, click "Program/Configure" (🔧) in the Programmer tool bar to complete downloading of Gowin\_EMPU\_M3 software programming BIN files in binary format.

## 1.2.5 Software Debugging

After downloading, if there are any design issues, you can use the U-LINK and J-LINK to debug on-line.

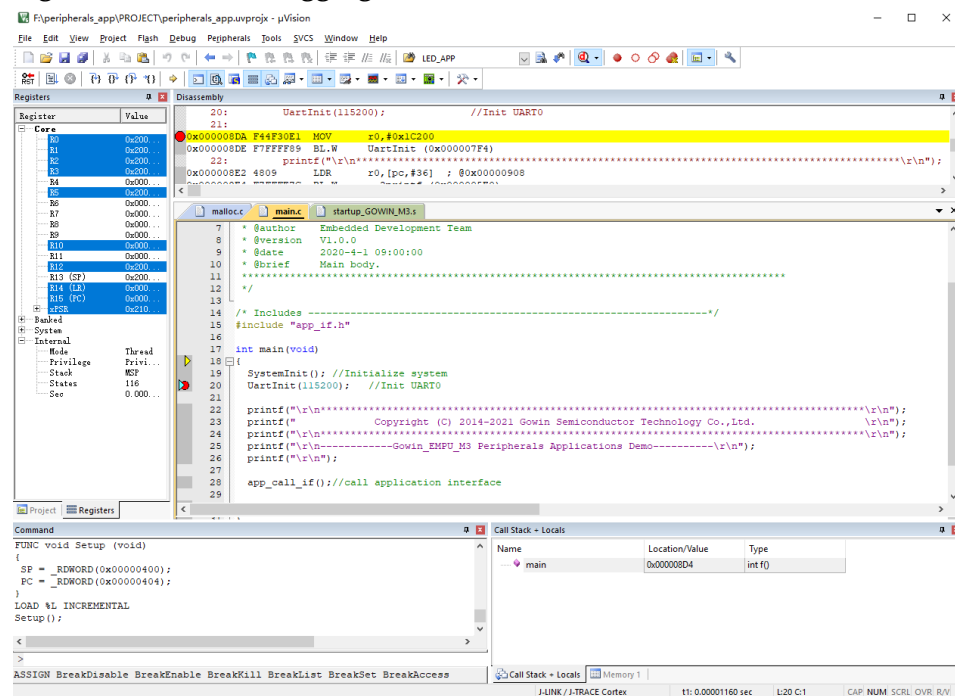
### 1. Connect the Emulator

Connect J-LINK or U-LINK according to the Debug Access Port location constrained to FPGA IO in the hardware design.

### 2. Start Debugging

Connect U-LINK or J-LINK emulators and click the "Debug" button in the tool bar to debug. Users can perform operations of breakpoint setting, single-step debugging, reset, run, etc. as shown in Figure 1-12.

**Figure 1-12 Start Debugging**



## 1.3 Reference Design

Gowin\_EMPU\_M3 provides reference design in ARM Keil MDK (V5.26 and above).

Gowin\_EMPU\_M3\ref\_design\MCU\_RefDesign\Keil\_RefDesign

# 2 GOWIN MCU Designer

## 2.1 Software Installation

The installation package of GOWIN MCU software Design is available at website <http://www.gowinsemi.com.cn/prodshow.aspx>.

For the software installation and configuration of Gowin MCU Designer, please refer to [SUG549](#), GOWIN MCU Designer User Guide.

**Note!**

It is recommended GOWIN MCU Designer (V1.1 and above).

## 2.2 Project Template

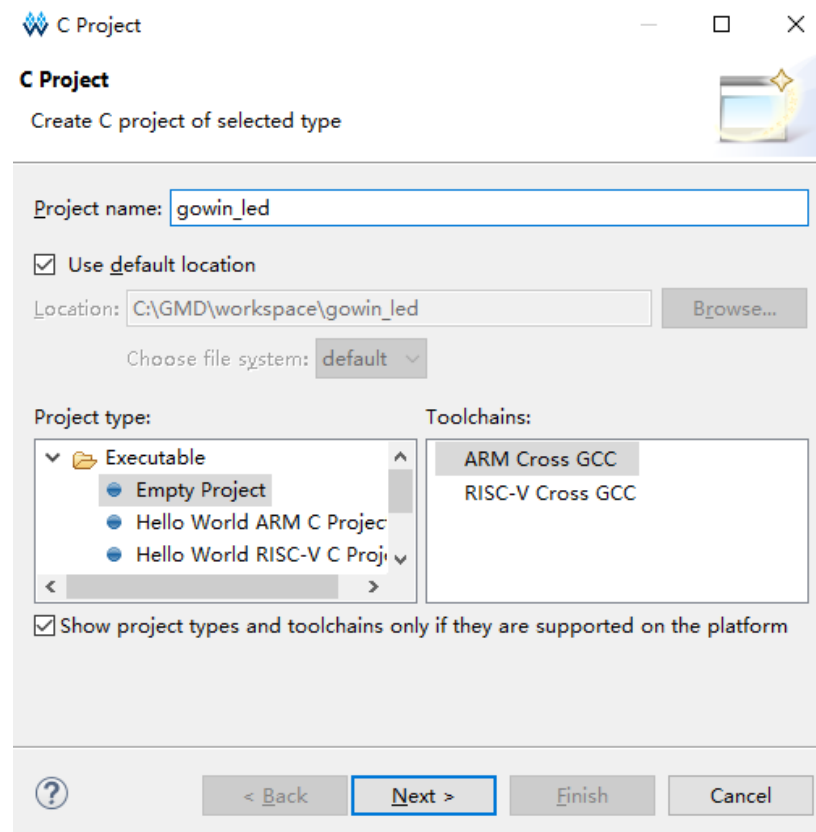
ARM GOWIN MCU Designer can be used for Gowin\_EMPU\_M3 software programming. The steps include project creation, configuration, coding, build and debug.

### 2.2.1 Create a New Project

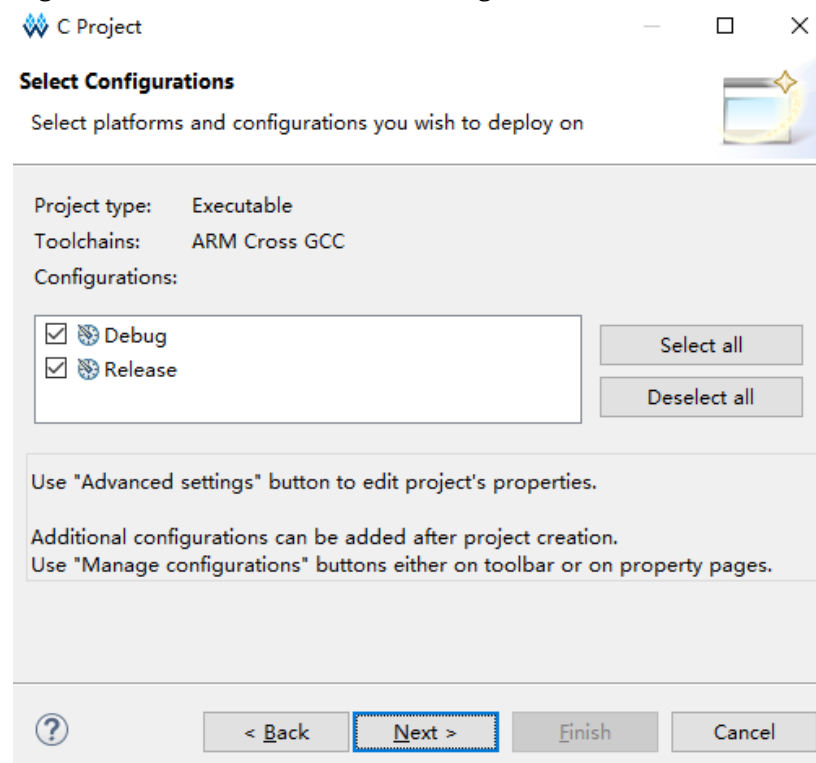
#### Create a New Project

Click "File > New > C Project" on the menu bar, as shown in Figure 2-1.

1. Create a project name and location;
2. Select the "Empty Project" type;
3. Select the "Cross ARM GCC" toolchain.

**Figure 2-1 Create a New Project****Select Platform and Configuration**

Select "Debug" and "Release" as the platform and configuration, as shown in Figure 2-2.

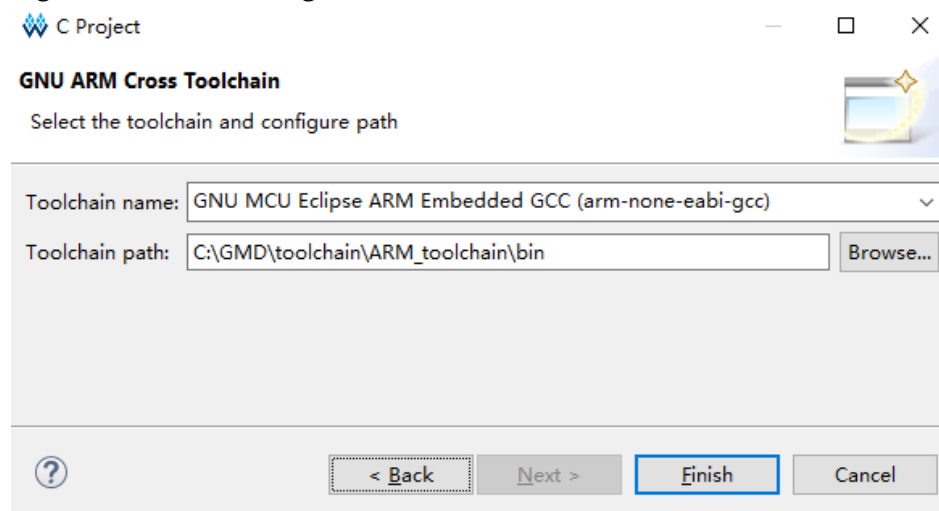
**Figure 2-2 Select Platform and Configuration**



## Select Configuration Path and Tool Chain

Select "arm-none-eabi-gcc" as the toolchain and its path, as shown in Figure 2-3.

Figure 2-3 Select Configuration Path and Toolchain



## Create a Project

After the new project creation, select the new projects in GOWIN MCU Designer workspace and add project item and code.

Take GMD\_RefDesign reference design for an instance, the software programming design contents and codes are listed as follows.

- CORE: ARM Cortex-M1 register definition
- PERIPHERAL: Peripheral driving function library
- STARTUP: Startup files
- SYSTEM: Register definition, system Initialization, and system clock definition
- USER: User application design
- Script/GOWIN\_M3\_flash.Id: Flash linker

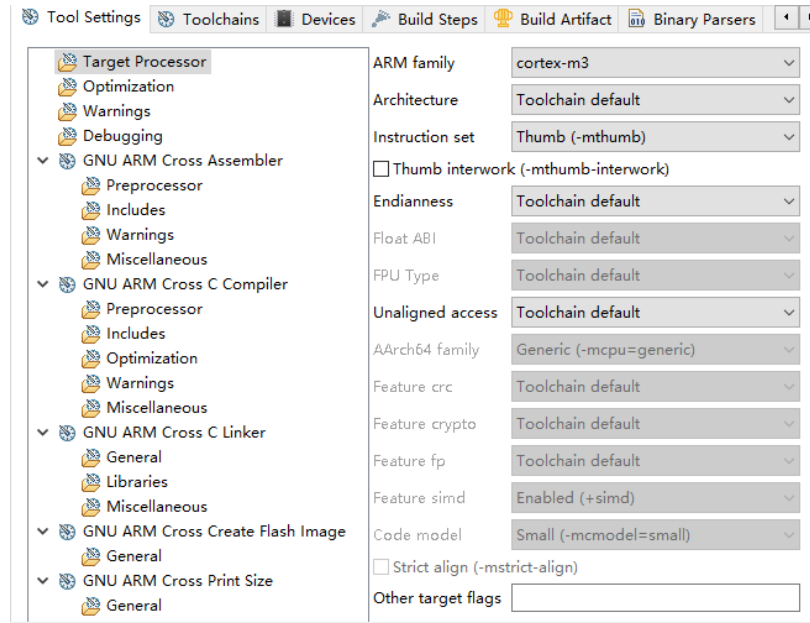
After the project architecture is created, select the current project in the Project Explorer view of GOWIN MCU Designer, right-click and select the "Refresh" to automatically update the architecture and code of the current project.

## 2.2.2 Configuration Options

In the Project Explorer view of GOWIN MCU Designer, select the current project, right click and select "Properties > C/C++ Build > Settings" to configure the project parameters.

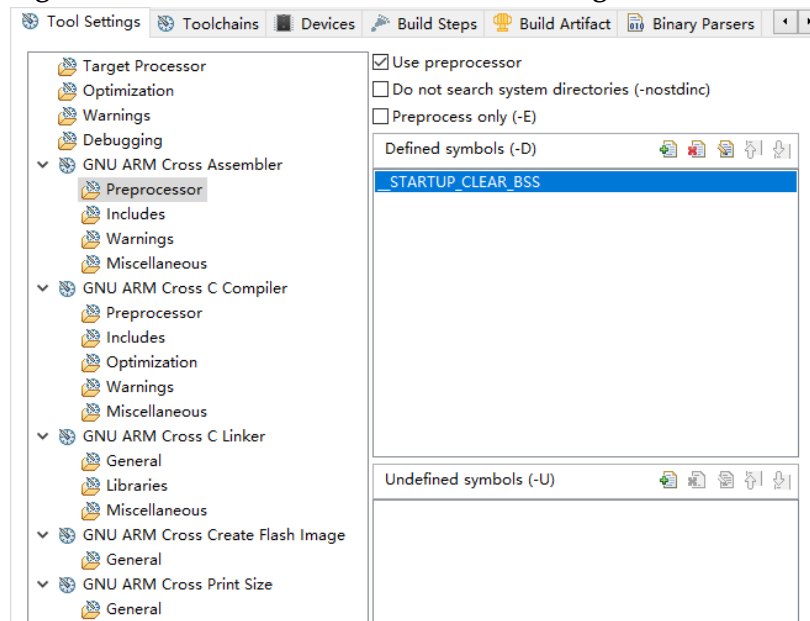
### Target Processor Configuration

Configure "Target Processor > ARM family" as "cortex-m3" as shown in Figure 2-4.

**Figure 2-4 Target Processor Configuration**

### Configure GNU ARM Cross Assembler

Configure "GNU ARM Cross Assembler > Preprocessor > Defined symbols (-D)" as "\_\_STARTUP\_CLEAR\_BSS", as shown in Figure 2-5.

**Figure 2-5 GNU ARM Cross Assembler Configuration**

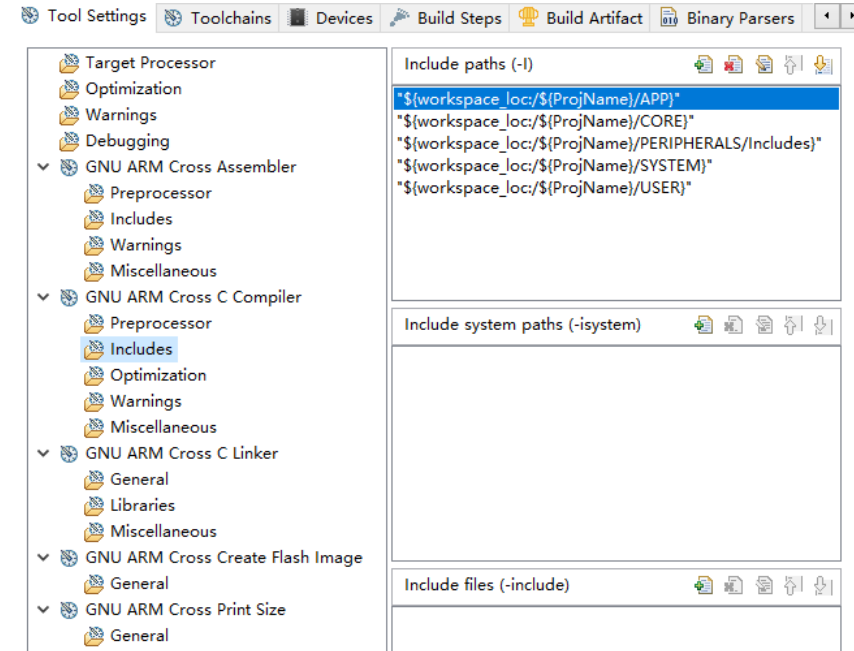
### Configure GNU ARM Cross C Compiler

Select "GNU ARM Cross C Compiler > Includes > Include paths (-I)" to configure C header file path of the current project, as shown in Figure 2-6.

Take GMD\_RefDesign reference design in SDK for an instance, the C header file paths are listed as follows.

- "\${workspace\_loc}/\${ProjName}/CORE"
- "\${workspace\_loc}/\${ProjName}/PERIPHERALS/Includes"

- "\${workspace\_loc}/\${ProjName}/SYSTEM}"
- "\${workspace\_loc}/\${ProjName}/USER}"
- "\${workspace\_loc}/\${ProjName}/APP}"

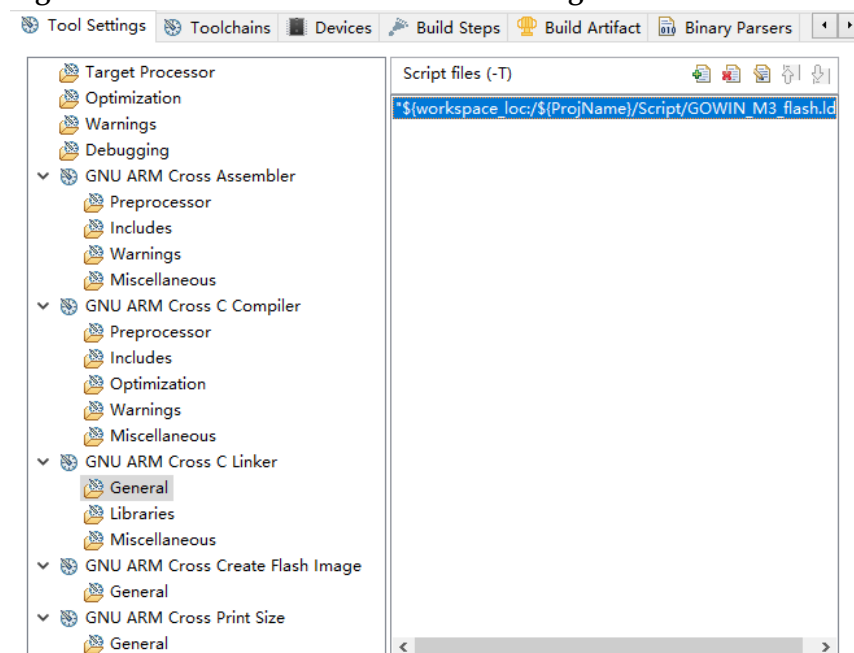
**Figure 2-6 GNU ARM Cross C Compiler Configuration**

### Configure GNU ARM Cross C Linker

Select "GNU ARM GNU C Linker > General > Script files (-T)" to configure Flash linker as "GOWIN\_M3\_flash.ld", as shown in Figure 2-7.

Take GMD\_RefDesign reference design in SDK for an instance, the Flash linker configuration is as follows.

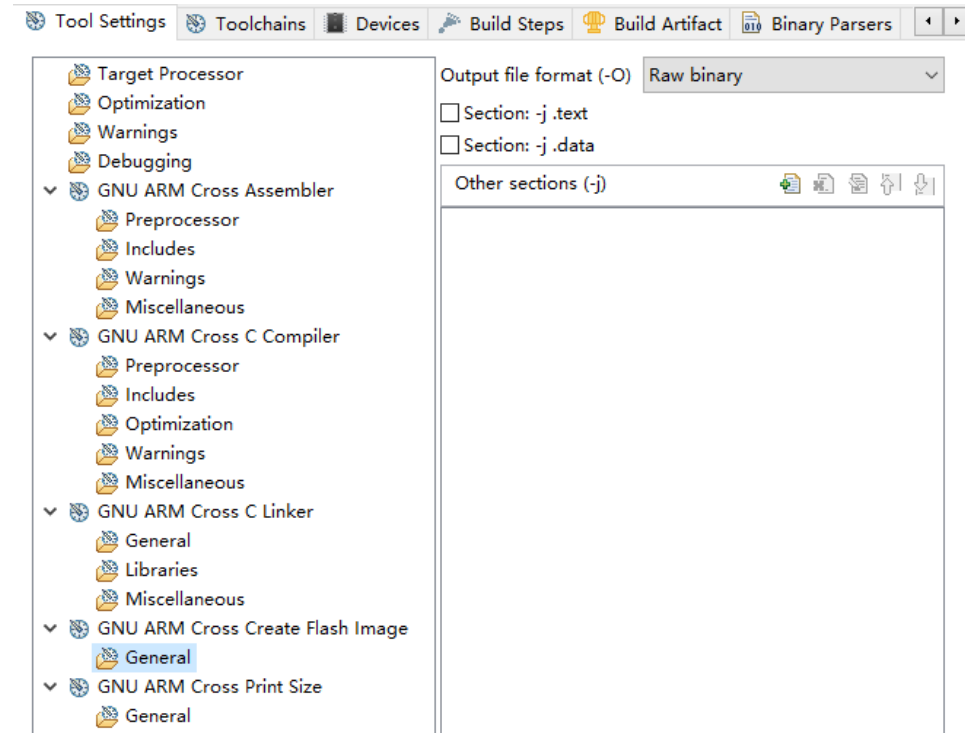
"\${workspace\_loc}/\${ProjName}/Script/GOWIN\_M3\_flash.ld}"

**Figure 2-7 GNU ARM Cross C Linker Configuration**

## Configure GNU ARM Cross Create Flash Image

Configure “GNU ARM Cross Create Flash Image > General > Output file format (-O)” as “RAW binary”, as shown in Figure 2-8.

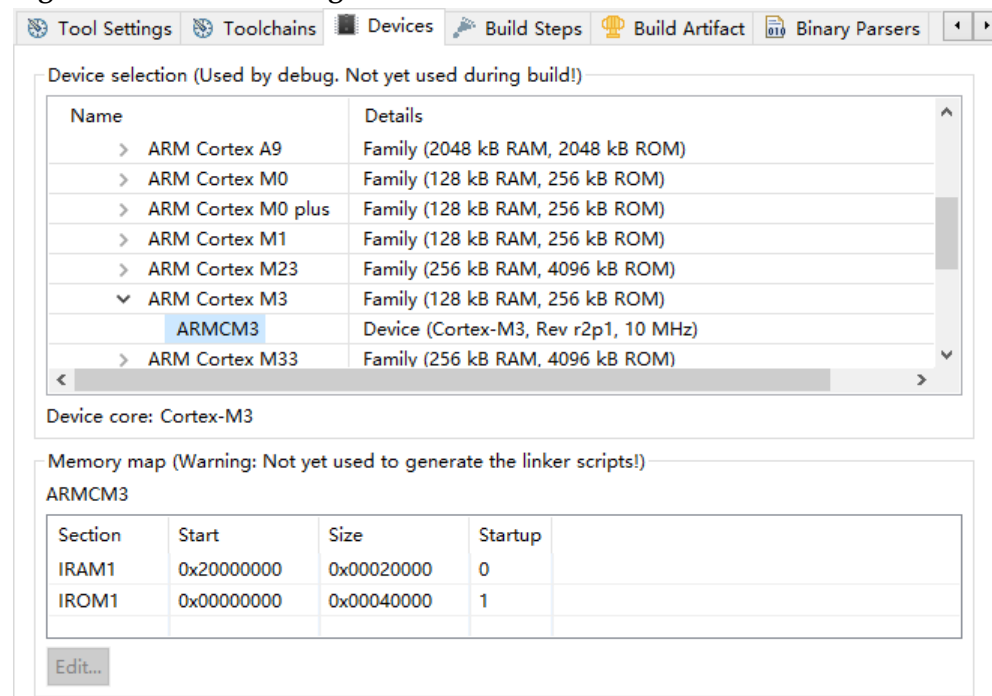
Figure 2-8 GNU ARM Cross Create Flash Image Configuration



## Configure Devices

Configure device as “ARM Cortex M3 > ARMCM3” in “Devices > Device selection” option, as shown in Figure 2-9.

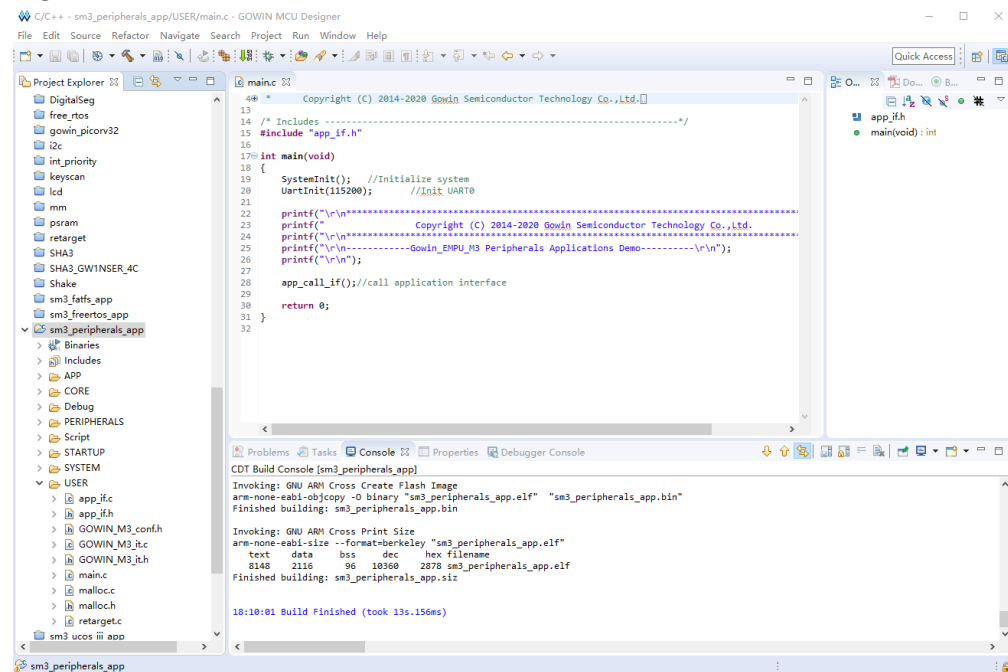
Figure 2-9 Devices Configuration



## 2.2.3 Build

After project configuration and encoding, select the "🔧" build button in the tool bar to generate Gowin\_EMPU\_M3 software programming BIN file in binary format, as shown in Figure 2-10.

**Figure 2-10 Build**



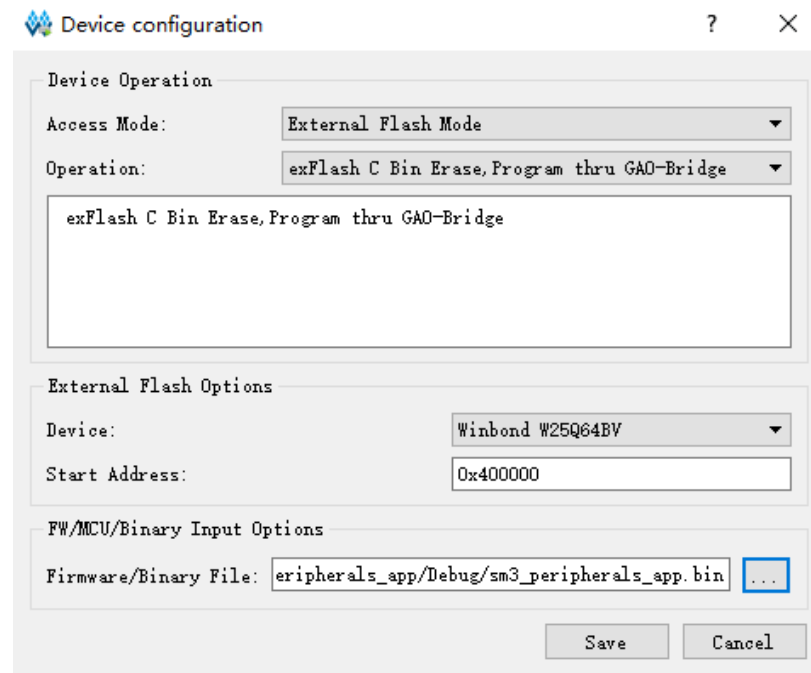
## 2.2.4 Download


After building, use Gowin Programmer, the download tool, to download the software programming BIN file in binary format.

In GOWIN MCU Designer, Click "Run/Programmer" in the menu bar or "🔧" in the tool bar to open Programmer.

Click "Edit > Configure Device" in the menu bar or "Configure Device" (🔧) in the tool bar to open the "Device configuration".

- Select "External Flash Mode" in "Access Mode" drop-down list;
- Select "exFlash C Bin Erase, Program thru GAO-Bridge" or "exFlash C Bin Erase, Program, Verify thru GAO-Bridge" in "Operation" drop-down list.
- Select "FW/MCU Input Options > Firmware/Binary File" to import the software programming BIN File in binary format to download.
- Select based on the on-board Flash in "External Flash Options > Device" (such as Winbond W25Q64BV);
- Configure the start address as "0x400000" in "External Flash Options > Start Address".
- Click "Save" as shown in Figure 2-11.

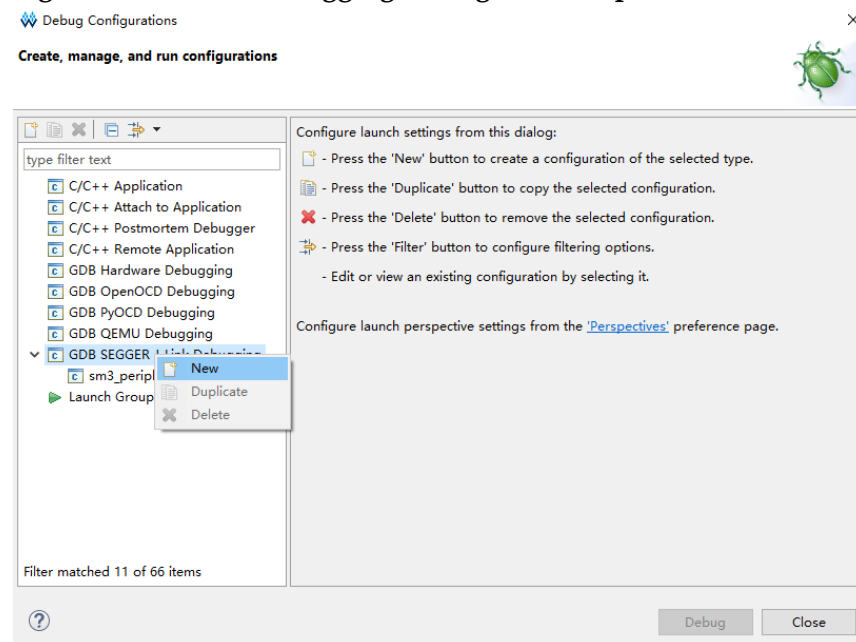
**Figure 2-11 Download**

After device configuration, click "Program/Configure" (  ) in the Programmer tool bar to complete downloading of Gowin\_EMPU\_M3 software programming BIN files in binary format.

## 2.2.5 Debug

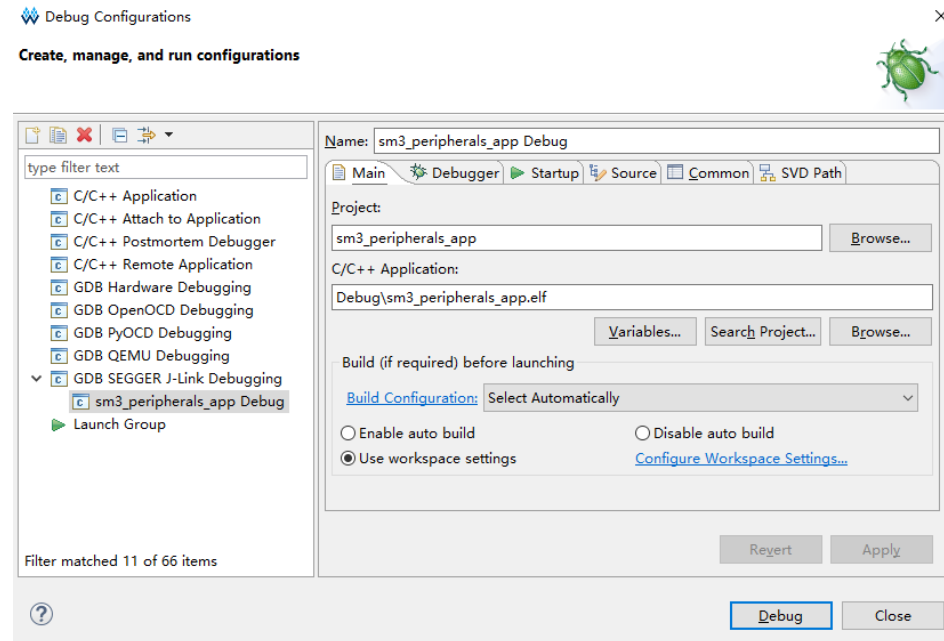
### Configure Debug Options

1. select "Run > Debug Configurations > GDB SEGGER J-Link Debugging > New" to create the project debugging configuration options, as shown in Figure 2-12.

**Figure 2-12 Create Debugging Configuration Options**

2. Select "Main" to configure the output image file of current project, as shown in Figure 2-13.

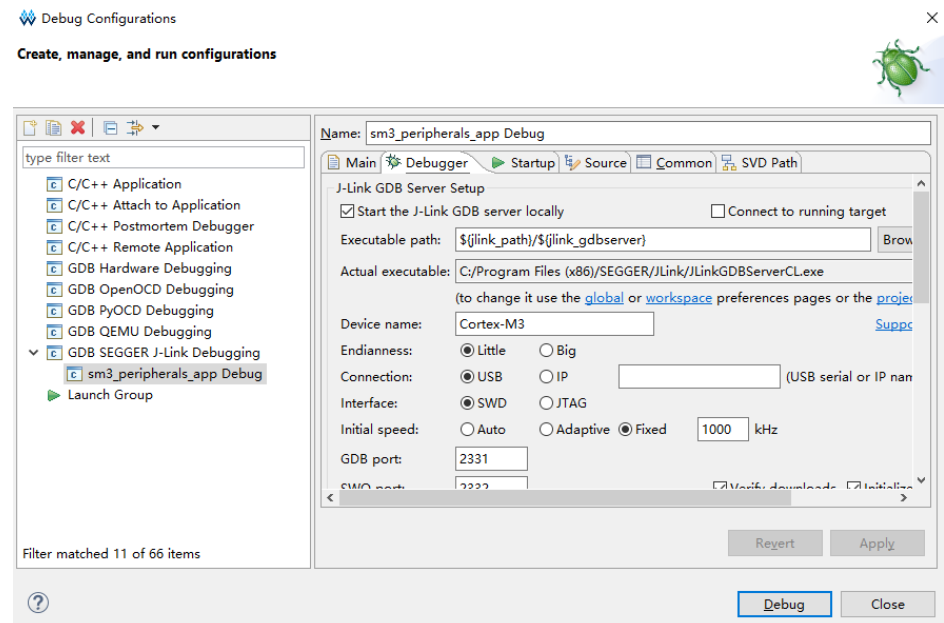
Figure 2-13 Main Configuration



3. Select "Debugger" to configure Debugger name and interface, as shown in Figure 2-14.

- Device Name: Cortex-M3
- Interface: JTAG or SWD
- Endianness: Little
- Connection: USB

Figure 2-14 Debugger Configuration

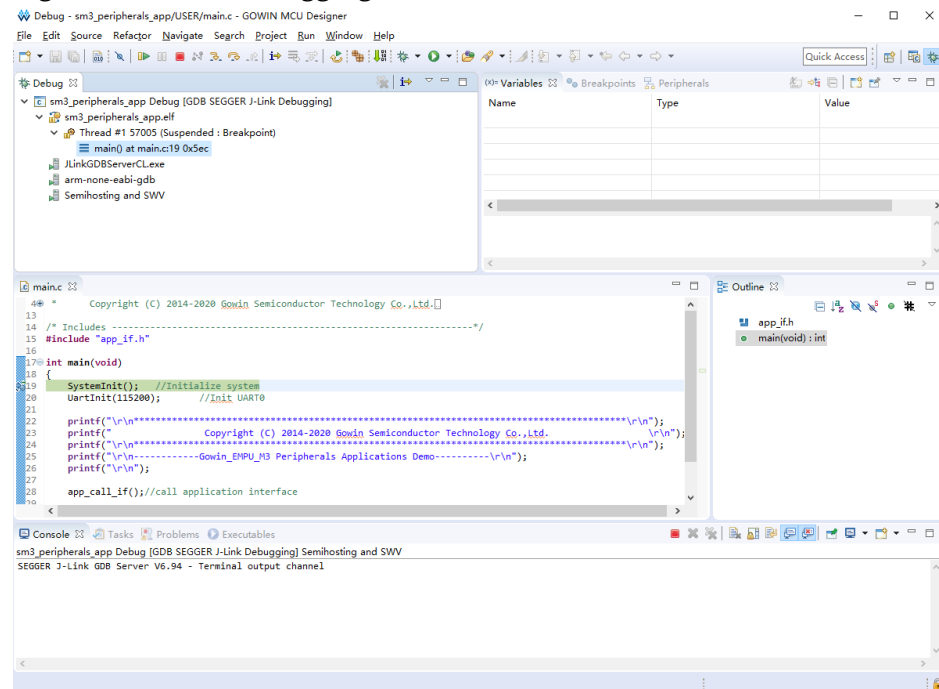


## Software Debugging

Connect J-LINK emulator to development board and click the "Debug" button in the tool bar to debug. Users can perform operations of breakpoint

setting, single-step debugging, reset, run, etc. as shown in Figure 2-15.

**Figure 2-15 Start Debugging**



## 2.3 Reference Design

Gowin\_EMPU\_M3 provides reference design in GOWIN MCU Designer (V1.1 and above).

Gowin\_MCU\_M3\ref\_design\MCU\_RefDesign\GMD\_RefDesign



