

Gowin_EMPU_M1 IDE Software **Reference Manual**

IPUG536-2.3.1E, 07/18/2025

Copyright © 2025 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

GOWIN is a trademark of Guangdong Gowin Semiconductor Corporation and is 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	
02/18/2019	1.0E	Initial version published.	
07/18/2019	1.1E	 MCU hardware design and software programming design support extended peripherals: CAN, Ethernet, SPI-Flash, RTC, DualTimer, TRNG, I²C, SPI, SD-Card. MCU supports off-chip SPI-Flash downloading startup. 	
08/18/2019	1.2E	 MCU hardware design and software programming design support extended peripheral: DDR3 Memory. Fixed known issues of ITCM, DTCM Size and IDE. 	
09/27/2019	1.3E	Updated and optimized MCU programming software and the interface and functions of Gowin MCU Designer.	
01/16/2020	1.4E	 MCU hardware design and software programming design supports PSRAM. MCU compiling software GMD V1.0 updated. RTOS reference design updated. Hardware and software reference design of AHB2 and APB2 extension bus interface added. 	
03/10/2020	1.5E	GW2A-18C/GW2AR-18C/GW2A-55C devices added.	
06/12/2020	1.6E	 MCU supports external instruction memory. MCU supports external data memory. Extension of 6 AHB bus interfaces. Extension of 16 APB bus interfaces. GPIO supports multiple interface types. I²C supports multiple interface types. 	
07/16/2021	1.7E	MCU version updated.	
10/12/2021	1.8E	ITCM and DTCM Size of GW2AN-9X/GW2AN-18X modified.	
05/11/2023	1.9E	Arora V FPGA products supported.	
07/21/2023	2.0E	Tested software version updated.	
03/07/2024	2.1E	 GW5AT-60 Version A products supported. Reference example for IDE software updated. 	
07/12/2024	2.2E	 GW5ART-15 Version A products supported. Reference example for IDE software updated. 	
01/17/2025	2.3E	The note on online software debugging added.	
07/18/2025	2.3.1E	The link of GMD software installation package in "2.1 Software Installation" updated.	

Contents

C	ontentsi
Li	ist of Figuresii
1	ARM Keil1
	1.1 Software Installation1
	1.2 Project Template1
	1.2.1 Create a New Project1
	1.2.2 Configuration Option2
	1.2.3 Build
	1.2.4 Download
	1.2.5 Software Online Debug9
	1.3 Reference Design 10
2	GMD Software10
	2.1 Software Installation 11
	2.2 Project Template 11
	2.2.1 Create a Project 11
	2.2.2 Configuration Option
	2.2.3 Build
	2.2.4 Download
	2.2.5 Software Online Debug19
	2.3 Reference Design

List of Figures

Figure 1-1 Create a New Project	. 1
Figure 1-2 Device Configuration	. 2
Figure 1-3 ROM and RAM Configuration	. 4
Figure 1-4 Output File Format Configuration	. 5
Figure 1-5 Header File Path Configuration	. 5
Figure 1-6 JTAG Debug Interface Configuration	. 6
Figure 1-7 SW Debug Interface Configuration	. 7
Figure 1-8 Flash Configuration	. 8
Figure 1-9 Debug Initialization File Configuration	. 8
Figure 1-10 Project Compiling	. 9
Figure 1-11 Start Debug	. 10
Figure 2-1 Creat a New Project	. 12
Figure 2-2 Select Platforms and Configurations	. 12
Figure 2-3 Select Configuration Toolchain and Path	. 13
Figure 2-4 Target Processor Configuration	. 14
Figure 2-5 Cross ARM GNU Assembler > Preprocessor Configuration	. 14
Figure 2-6 Cross ARM C Compiler > Includes Configuration	. 15
Figure 2-7 Cross ARM C Linker Configuration	. 17
Figure 2-8 Cross ARM GNU Create Flash Image Configuration	. 17
Figure 2-9 Devices Configuration	. 18
Figure 2-10 Build	. 18
Figure 2-11 Create Software Debugging Configurations Option	. 19
Figure 2-12 Main Option Configuration	. 20
Figure 2-13 Debugger Option Configuration	. 21
Figure 2-14 Software Debugging Level Configuration	. 22
Figure 2-15 Software Online Debugging Start-up	. 23

1 ARM Keil

1.1 Software Installation

For the detailed, please refer to <u>Getting Started with MDK</u> provided by ARM Keil MDK website.

Note!

ARM Keil MDK (V5.26 and above) is recommended.

1.2 Project Template

ARM Keil MDK can be used for Gowin_EMPU_M1 software programming. The steps include project creation, configuration, coding, compilation, downloading and debugging.

1.2.1 Create a New Project

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

🐺 µVision Х File Edit View Project Flash Debug Peripherals Tools SVCS Window Help 📄 📄 🚽 🗿 New μVision Project... 🎦 🏡 🔃 🚝 🎼 /版 💆 RTOS 💦 🔍 🔍 🔹 New Multi-Project Workspace... 🔊 🕆 🖓 🖶 🖓 🎃 Open Project... Project Close Project ۲ Export Manage ۲ Select Device for Target ... Remove Item N Options... Alt+F7 Clean Targets Build Target F7 Rebuild all target files 🍪 🛛 Batch Build Batch Setup... 🖻 Pr... 🧑 Bo... | · Translate... Ctrl+F7 Build Output џX Stop build Create a new µVision project

Figure 1-1 Create a New Project

1.2.2 Configuration Option

Device Configuration

ARM Cortex-M1 is embedded in Gowin_EMPU_M1, so the device type is configured as " ARM Cortex M1 > ARMCM1", as shown in Figure 1-2.



Select Device for	Target 'Target 1'	;	×
Device			
Softwa	re Packs	•	
Vendor: ARM Device: ARMC Toolset: ARM Search:	M1		
,		Des <u>cr</u> iption:	
ARM ARM ARM ARM ARM ARM ARM ARM	Cortex M0 Cortex M0 plus Cortex M1 RMCM1 Cortex M23 Cortex M3 Cortex M33 Cortex M35P Cortex M4	The ARM Cortex-M1 FPGA processor is intended for deeply embedded applications that require a small processor integrated into an FPGA. The ARM Cortex-M1 processor implements the ARMv6-M architecture profile.	
		OK Cancel Help	

ROM and RAM Configuration

Gowin_EMPU_M1 internal instruction memory or external instruction memory is the ROM.

Gowin_EMPU_M1 internal data memory or external data memory is the RAM.

1. Configure the initial address and the size of ROM (Internal Instruction Memory) and RAM (Internal Data Memory).

ROM initial address and size configuration:

- Off-chip SPI-Flash memory download startup
 - ROM initial address: 0x400
 - ROM Size: Set according to the actural configuration of the hardware design ITCM Size. For example, if ITCM Size 32KB, ROM size is set to 0x7C00.
- On-chip ITCM initialization value download startup
 - ROM initial address: 0x00000000.
 - ROM Size: Please set according to the actural configuration of the hardware design ITCM Size. For example, if ITCM Size

32KB, ROM size is set to 0x8000.

RAM initial address and Size configuration:

- RAM initial address: 0x20000000
- RAM Size: Please set according to the actual configuration of the hardware design DTCM Size. For example, if DTCM Size 32KB, RAM size is set to 0x8000.

Limited by the on-chip memory resource, the size configuration of ITCM and DTCM cannot exceed the max. on-chip memory size.

- For GW1N/R-9, ITCM or DTCM can be configured up to 32KB. If ITCM or DTCM has been configured to 32KB, the other can only be configured up to 16KB.
- For GW2AN-9X/18X, ITCM or DTCM can be configured up to 32KB. If ITCM or DTCM has been configured to 32KB, the other can only be configured up to 16KB.
- For GW2A/R/NR-18, ITCM or DTCM can be configured up to 64KB. If ITCM or DTCM has been configured to 64KB, the other can only be configured up to 16KB.
- For GW2A/N-55, ITCM or DTCM can be configured up to 256KB. If ITCM or DTCM has been configured to 256KB, the other can only be configured up to 16KB.
- For GW5A/T/ST/S-138, ITCM or DTCM can be configured up to 512KB. If ITCM or DTCM has been configured to 512KB, the other can only be configured up to 218KB.
- For GW5AT-75, ITCM or DTCM can be configured up to 256KB. If ITCM or DTCM has been configured to 256KB, the other can only be configured up to 256KB.
- For GW5A/R/S-25, ITCM or DTCM can be configured up to 64KB. If ITCM or DTCM has been configured to 64KB, the other can only be configured up to 32KB.
- For GW5A/T-60, ITCM or DTCM can be configured up to 128KB. If ITCM or DTCM has been configured to 128KB, the other can only be configured up to 64KB.
- For GW5AT/RT-15, ITCM or DTCM can be configured up to 64KB. If ITCM or DTCM has been configured to 64KB, the other can only be configured up to 8KB.

The configuration of ROM (Internal Instruction Memory) and RAM (Internal Data Memory) is as shown in Figure 1-3.

Using DK_START_GW5A-LV25UG324 V2.0 development board reference design an instance, the initial address of ROM is 0x400 and the "Size" is 0x7C00. The initial address of RAM is 0x20000000 and the Size is 0x8000.

0 0	/					
🕅 Options for Target 'led'	×					
Device Target Output Listing User C/C++ Asm Linker Debug Utilities						
ARM ARMCM1 Code Generation						
<u>X</u> tal (MHz): 12.0	ARM Compiler: Use default compiler version 5					
Operating system: None 🔽 🔽 Use Cross-Module Optimization						
System Viewer File:	Use MicroLIB II Big Endian					
🔲 Use Custom File						
Read/Only Memory Areas	Read/Write Memory Areas					
default off-chip Start Size Sta	artup default off-chip Start Size Nolnit					
ROM1:	C 🛛 RAM1: 🗖 🗖					
ROM2:	C 🗆 RAM2:					
П ROM3:	C 🗆 RAM3: 🔽 🗖					
on-chip	on-chip					
IROM1: 0x400 0x7C00						
	C □ IRAM2: □ □ □					
OK	Cancel Defaults Help					

Figure 1-3 ROM and RAM Configuration

2. Configure the initial address and the size of ROM (External Instruction Memory) and RAM (External Data Memory).

ROM initial address and Size configuration:

- ROM initial address: 0x00000000.
- ROM Size: Please set according to the actual Size of the hardware design.

RAM initial address and Size configuration:

- RAM initial address: 0x20100000.
- RAM Size: Please set according to the actual Size of the hardware design.

Output File Format Configuration

Gowin_EMPU_M1 outputs BIN file, so axf file format should be converted to BIN file format.

If executable program file is used as the initial value of ITCM, the BIN file should be converted to four hex files, itcm0, itcm1, itcm2, and itcm3 using Gowin script make_hex.exe.

The usage of calling the file format tool with user command line is as shown in Figure 1-4.

- Run #1: fromelf.exe --bin -o bin-file axf-file
- Run #2: make_hex.exe bin-file

ommand Items	User Command		Stop on Exi	S
Before Compile C/C++ File				
		2	Not Specified	
Run #2		2	Not Specified	
Before Build/Rebuild				
🗌 Run #1		2	Not Specified	
Run #2		2	Not Specified	
After Build/Rebuild				
🔽 Run #1	$\label{eq:linear} D:\Keil_v5\ARM\ARMCC\bin\fromelf.exebin\$	2	Not Specified	\Box
Run #2	make_hex.exe led.bin	2	Not Specified	\Box
<u>R</u> un 'After-Build' Conditionally				

Figure 1-4 Output File Format Configuration

Header File Path Configuration

The C code header file configuration is used to call the C code header file during building. The configuration is as shown in Figure 1-5.

Figure 1-5 Header File Path Configuration

🖏 Options for Target 'led'	×
Device Target Output Listing User C/C++ Asm Linker Debug Utilities	
Preprocessor Symbols	
Undefine:	
Language / Code Generation	
Execute-only Code Strict ANSI C Warnings: All Warnings	•
Optimization: Level 0 (-O0) ▼ □ Enum Container always int □ Thumb Mode	
Coptimize for Time	udes
☐ Split Load and Store Multiple ☐ Read-Only Position Independent	
Image: Provide a state of the sta	ons
Include Paths <u>Mi</u> sc Controls	·
Compiler -c99 -c -cpu Cortex-M1 -li -g -00apcs=interworksplit_sections -l //.ibrary/libraries/cmsis/cm1/core_support/mdk -l	Ŷ
OK Cancel Defaults	[elp

Debug Configuration

- Configure the Emulator
 - 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".
- Configure the Debug Interface
 - JTAG Debug Interface If it is configured as the JTAG debug interface, the configuration method is as shown in Figure 1-6.

Figure	1-6 ITAG	Dehug	Interface	Configur	ation
Inguit	1-0 J1110	Debug	include	Comigui	ation

CN 1/20/06/205		IDCODE	Davisa Nama	IP Ion Move
Device: J-Link ARM	TDO	⊙ 0x4BA00477	ARM CoreSight JTAG-DP	4 Up
HW : V8.00 dll : V6.94 FW : J-Link ARM V8 compiled No	TDI	<		> Down
Port: Max JTAG	C Manu	matic Detection Jal Configuration	ID CODE: Device Name:	
Auto Clk	Add	Delete Up	date IR len:	
Connect & Reset Options Connect: Normal Reset: Nor Reset after Connect	mal	Cach	e Options Download ache <u>C</u> ode ache <u>M</u> emory Down	d Options Code Download load to <u>F</u> lash
Connect & Reset Options Connect: Normal Reset: Nor Reset after Connect Interface CUSB C TCP/IP	mal	Cach	e Options Download ache <u>C</u> ode ache <u>M</u> emory Down	d Options Code Download load to <u>F</u> lash Misc

- SW Interface

If it is configured as the SW debug interface, the configuration is as shown in Figure 1-7.

	k/JTrace Target Driv	er Setup			
ebug T	Trace Flash Downlo	bad			
J-Link /	J-Trace Adapter	SW Dev	vice		
SN:	4294967295	•	IDCODE	Device Name	Move
Device:	J-Link ARM	A SW	DI 0x2BA01477	ARM CoreSight SW-DP	Up
HW :	V8.00 dll :	V6.94			Dawn
FW:	J-Link ARM V8 cor	npiled No			
Po	ort:	Max @ Au	omatic Detection	ID CODE:	
S	SW 👻 5 MH	lz 🔹 C Ma	nual Configuration	Device Name:	
		Auto Clk	d Delete Ut	odate IR len:	
	-				
-Conne	act & Resat Ontions			he Options Download	Ontions
Conne Conne	ect & Reset Options ect: Normal 💌 eset after Connect	Reset: Normal	Cad	he Options — Download Cache <u>C</u> ode — Verify Cache <u>M</u> emory — Downl	l Options Code Download oad to <u>F</u> lash
Conne Conne I <u>R</u> Interfac	ect & Reset Options ect: Normal v eset after Connect ce B © TCP/IP	Reset: Normal TCP/IP Network Settings		he Options Download Cache <u>C</u> ode Cache <u>M</u> emory Downlo	I Options Code Download oad to Elash Misc
Conne Conne I <u>R</u> i	ect & Reset Options ect: Normal v eset after Connect ce B O TCP/IP	Reset: Normal TCP/IP Network Settings IP-Addres:	Port (/	he Options Download	l Options Code Download oad to <u>F</u> lash Misc JLink Info
Conne Conne I R Interfac C US	ect & Reset Options ect: Normal v eset after Connect ce B O TCP/IP Scan eady	Reset: Normal TCP/IP Network Settings IP-Addres: 127 . 0 . 0	▼ Cacl	he Options Download Cache <u>C</u> ode Cache <u>M</u> emory Download	I Options Code Download oad to Elash Misc JLink Info JLink Cmd
Conne Conne I R Interfac C US	ect & Reset Options ect: Normal v eset after Connect ce B C TCP/IP Scan eady	Reset: Normal TCP/IP Network Settings IP-Addres: 127 . 0 . 0	■ Cacl	he Options Download	l Options Code Download oad to Elash Misc JLink Info JLink Cmd

Figure 1-7 SW Debug Interface Configuration

In the Debug Interface Type Configuration option:

- Please do not select the "Download Options > Verify Code Download" option.
- Please do not select the "Download Options > Download to Flash" option.

Flash Configuration

If online debugging is required, "Update Target before Debugging" cannot be selected, as shown in Figure 1-8.

B Options for Target 'led'	×
Device Target Output Listing User C/C++ Asm Linker Debug Utilities	
Configure Flash Menu Command	
Use Debug Driver Settings 🗌 Update Target before Debugg	ging
Init File:	
C Use External Tool for Flash Programming	
Command:	
Arguments:	
🗖 Run Independent	
Configure Image File Processing (FCARM):	
Output File: Add Output File to Group:	
STARTUP	-
Image Files Root Folder:	
OK Cancel Defaults	Help

Figure 1-8 Flash Configuration

Debug Initialization File Configuration

If selecting off-chip SPI-Flash memory download startup, it needs to load debug initialization file when debugging online. Select ext_debug.ini in "Initialization File" option as shown in Figure 1-9.

"ext_debug.ini" file is located in the "library\debug".

Figure 1-9 Debug Initialization File Configuration

😗 Options for Target 'led'	×
Device Target Output Listing User C/C++ A	Asm Linker Debug Vtilities
C Use Simulator with restrictions Settings □ Limit Speed to Real-Time	
Load Application at Startup Initialization File: Edit	Load Application at Startup Initialization File:
Restore Debug Session Settings Image: Breakpoints Image: Toolbox Image: Breakpoints Image: Toolbox	Restore Debug Session Settings
CPU DLL: Parameter: SARMCM3.DLL	Driver DLL: Parameter: SARMCM3.DLL
Dialog DLL: Parameter: DARMCM1.DLL pCM1	Dialog DLL: Parameter: TARMCM1.DLL -pCM1
Warn if outdated Executable is loaded Manage Component Vie	Warn if outdated Executable is loaded
OK Car	wel Defaults Help

1.2.3 Build

After encoding and configuration, click Build " or Rebuild " click "Project > Build Target" or "Project > Rebuild all target files" on the menu bar to build the project to generate software design BIN file and four hex files of itcm0, itcm1, itcm2, and itcm3, as shown in Figure 1-10.





1.2.4 Download

After compiling Gowin_EMPU_M1software programming design, for the downloading, please refer to <u>IPUG532, Gowin_EMPU_M1Download</u> <u>Reference Manual.</u>

1.2.5 Software Online Debug

After completing the download of the hardware design bitstream files generated by the hardware design and the software design BIN files generated by the software programming design, if there are any issues, you can use the U-LINK and J-LINK to debug online.

You can download and debug the software, no recompilation required.

Note!

Gowin_EMPU_M1 does not support automatic downloading during debugging with ARM Keil software. Prior to each debugging session, please use the Programmer tool to download the Binary file of the software design you intend to debug. Then, start debugging to ensure that the software project being debugged is the current one.

1. Connect Emulator

Connect J-LINK or U-LINK according to the Debug Access Port (JTAG: JTAG_3~JTAG_18, VCC and GND; or SWD: JTAG_7, JTAG_9, VCC and GND) location constrained to FPGA IO in the hardware design.

2. Start Debug

Connect the U-LINK or J-LINK Emulator. Click the Debug button " on the tool bar, or click "Debug > Start/Stop Debug Session" on the menu bar to start debug. You can perform operations of breakpoint setting, single-step debug, reset and run, as shown in Figure 1-11.

Figure 1-11 Start Debug



1.3 Reference Design

Gowin provides reference design in ARM Keil MDK (tested software version V5.26) software environment. Click this <u>link</u> to get following reference design:

...\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo、 cm1_fatfs、cm1_freertos、cm1_rtthread_nano、cm1_tcpip、 cm1_ucos_iii

2_{GMD} Software</sub>

2.1 Software Installation

GMD software installation package is available at Gowinsemi website.

For the software installation and configuration of GMD, please refer to <u>SUG549, GOWIN MCU Designer User Guide</u>.

Note!

GOWIN MCU Designer (V1.2 and above) is recommended.

2.2 Project Template

Using GMD for Gowin_EMPU_M1 software programming design, it involves projects creation, option configuration, code writing, building, downloading, and online debug.

2.2.1 Create a Project

Create a New Project

Click "New" (\square) on the tool bar or select "File > New > C Project" on the menu bar, as shown in Figure 2-1.

- 1. Create a project name and location.
- 2. Select "Empty Project" type.
- 3. Select "ARM Cross GCC" building tool chain.

Figure 2-	1 Creat a	New Pro	oject
-----------	-----------	---------	-------

🐳 C Project	-	
C Project		
Create C project of selected type		
Project name: gowin_led		
✓ Use <u>d</u> efault location		
Location: D:\GMD_workspace\workspace	ce_cm1\gowin_led	B <u>r</u> owse
Choose file system: default	×	
Project type:	Toolchains:	
✓ ➢ Executable	ARM Cross GCC	
Empty Project	RISC-V Cross GCC	
<		
Show project types and toolchains on	ly if they are supported o	on the platform
? < <u>B</u> ack <u>N</u> e	ext > <u>F</u> inish	Cancel

Select Platforms and Configurations

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

Figure 2-2 Select Platforms and Configurations

🐳 C Project	_		×
Select Configurations Select platforms and configurations you wish to deploy on			\$
Project type: Executable Toolchains: ARM Cross GCC Configurations:	5	Select all	
Use "Advanced settings" button to edit project's properties. Additional configurations can be added after project creation Use "Manage configurations" buttons either on toolbar or o	on.	erty page	s.
? ≤ <u>B</u> ack <u>Next</u> > <u>Finit</u>	sh	Canc	el

Select Configuration Toolchain and Path

Select "arm-none-eabi-gcc" as the cross compiling toolchain and import its path. It is recommended that Toolchain name and Toolchain path be configured by default, as shown in Figure 2-3.

```
Figure 2-3 Select Configuration Toolchain and Path
```

关 C Project			_	
GNU ARM Cross Toolchain				
Select the toolchain and configure path				
Toolchain name:	GNU MCU Eclipse ARM Embedded (iCC (arm-none	-eabi-gcc)	~
Toolchain path:	D:\GMD\toolchain\ARM_toolchain\bi	n		Browse
?	< <u>B</u> ack	ext >	<u>F</u> inish	Cancel

Create a Project

After project creation, select the created project in Project Explorer view, add engineering structure and import the software programming design.

Using GMD_RefDesign for an instance, the software programming design projects and codes are listed as follows.

Select the current project in Project Explorer view, and right-click "Refresh" option to automatically update the structure and code of the current project.

2.2.2 Configuration Option

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

Target Processor Configuration

Select "Target Processor > ARM family " and configure the option to "cortex-m1", as shown in Figure 2-4.

Figure 2-4 Target Processo	Configura	tion
🛞 Tool Settings 🛞 Toolchains 🔳 Devices	🎤 Build Steps 🖳	Build Artifact 🗟 Binary Parsers 🚺
Processor	ARM family	cortex-m1 ~
🖉 Optimization 🖄 Warnings	Architecture	Toolchain default \lor
🖄 Debugging	Instruction set	Thumb (-mthumb)
🗸 🛞 Cross ARM GNU Assembler	Thumb interworl	k (-mthumb-interwork)
🖄 Preprocessor	E	
🖄 Includes	Englanness	
🖉 Warnings	Float ABI	Toolchain default \sim
🖄 Miscellaneous	EPI I Turbe	Toolchain default
V 🛞 Cross ARM C Compiler	In o type	
Preprocessor	Unaligned access	Toolchain default ~
2 Includes	AArch64 family	Generic (-mcpu=generic)
Werninge		
Missellaneous	Feature crc	loolchain default
× S Cross ARM C Linker	Feature crypto	Toolchain default \sim
General	Eo oturo fo	Toolchain default
Libraries	reature ip	
Miscellaneous	Feature simd	Enabled (+simd)
🗸 🛞 Cross ARM GNU Create Flash Image	Code model	Small (-mcmodel=small)
🖄 General	Ctrict align (mot	rist alian)
🗸 🛞 Cross ARM GNU Print Size	Strict align (-mst	rict-align)
🖄 General	Other target flags	
	<	3
		/

... • 4 **T** (D 0 0 ...

Cross ARM GNU Assembler > Preprocessor Configuration

Select " Cross ARM GNU Assembler > Preprocessor > Defined symbols (-D)" to configure the option to "__STARTUP_CLEAR_BSS" as shown in Figure 2-5.

Figure 2-5	Cross ARM	GNU	Assembler >	> Pre	processor	Configu	ratior



Cross ARM C Compiler > Includes Configuration

Select "Cross ARM C Compiler > Includes > Include paths (-I)" to configure the C header file path, as shown in "\${workspace_loc:/\${ProjName}/src/library/libraries/cmsis/cm1/core_suppo rt/gmd}"

- "\${workspace_loc:/\${ProjName}/src/library/libraries/cmsis/cm1/device_ support}"
- "\${workspace_loc:/\${ProjName}/src/library/libraries/drivers/inc}"
- "\${workspace_loc:/\${ProjName}/src/library/middlewares/delay}"
- "\${workspace_loc:/\${ProjName}/src/library/middlewares/dmm}"
- "\${workspace_loc:/\${ProjName}/src/library/middlewares/gpio}"
- "\${workspace_loc:/\${ProjName}/src/library/middlewares/uart}"
- "\${workspace_loc:/\${ProjName}/src/project}"

Figure 2-6.

For example, in the software programming reference design "GMD_RefDesign\cm1_demo", the configuration of the C header file reference path is described as below.

- "\${workspace_loc:/\${ProjName}/src/library/libraries/cmsis/cm1/core_s upport/gmd}"
- "\${workspace_loc:/\${ProjName}/src/library/libraries/cmsis/cm1/device_ support}"
- "\${workspace_loc:/\${ProjName}/src/library/libraries/drivers/inc}"
- "\${workspace_loc:/\${ProjName}/src/library/middlewares/delay}"
- "\${workspace_loc:/\${ProjName}/src/library/middlewares/dmm}"
- "\${workspace_loc:/\${ProjName}/src/library/middlewares/gpio}"
- "\${workspace_loc:/\${ProjName}/src/library/middlewares/uart}"
- "\${workspace_loc:/\${ProjName}/src/project}"

 Target Processor Optimization Warnings Debugging GNU ARM Cross Assembler Preprocessor Includes Warnings Warnings 	Include paths (-1) *\$(workspace_loc:/\$(ProjName)/src/library/libraries/cmsis/cm1/cd *\$(workspace_loc:/\$(ProjName)/src/library/libraries/cmsis/cm1/d *\$(workspace_loc:/\$(ProjName)/src/library/middlewares/delay)* *\$(workspace_loc:/\$(ProjName)/src/library/middlewares/dmm)* *\$(workspace_loc:/\$(ProjName)/src/library/middlewares/gpio)* *\$(workspace_loc:/\$(ProjName)/src/library/middlewares/gpio)* *\$(workspace_loc:/\$(ProjName)/src/library/middlewares/uart)* *\$(workspace_loc:/\$(ProjName)/src/library/middlewares/uart)*	evice	supr s_su	Sort ppo	신] /gm rt}"	₽ nd}*
 Winderlandeds Warnings Miscellaneous GNU ARM Cross C Compiler Preprocessor Includes Optimization Warnings Miscellaneous GNU ARM Cross C Linker General Libraries Miscellaneous 	Include system paths (-isystem)		(1 ₃₀)		<u>ها</u>	Ð
 SGNU ARM Cross Create Flash Image General SGNU ARM Cross Print Size General 	Include files (-include)		1. 		<u>ې</u>	Ŷ

Figure 2-6 Cross ARM C Compiler > Includes Configuration

Cross ARM C Linker Configuration

Select "Cross ARM C Linker > General > Script files (-T)" to configure "GOWIN_M1_flash_burn.ld" or "GOWIN_M1_flash_xip.ld" as GMD Flash linker, as shown in Figure 2-7.

Using software programming reference design GMD_RefDesign\cm1_demo for an instance, the Flash link is configured as below.

"\${workspace_loc:/\${ProjName}/src/library/libraries/cmsis/cm1/device _support/startup/gmd/linker/GOWIN_M1_flash_burn.ld}"

The GMD Flash linker Flash initial address "FLASH ORIGIN" setting is shown below:

- Internal Instruction Memory:
 - GOWIN_M1_flash_xip.ld: FLASH ORIGIN: 0x00000000, ITCM Initialization download running
 - GOWIN_M1_flash_burn.ld: FLASH ORIGIN: 0x00000400, off-chip SPI-Flash memory download boot
- External Instruction Memory:
 - GOWIN_M1_flash_xip.ld: FLASH ORIGIN: 0x00000000.

Target Processor	Script files (-T)	🔊 🎣 🜚 🏹 ·
Optimization	*\${workspace_loc:/\${ProjName}/src/library/libraries/cmsis/cm1/device_support/s	tartup/gmd/linker/GOWIN_M1_flash_burn.ld
2 Warnings		
2 Debugging		
GNU ARM Cross Assembler		
2 Preprocessor		
2 Includes		
Warnings		
Miscellaneous		
GNU ARM Cross C Compiler		
2 Preprocessor		
Includes		
Detimization		
Warnings		
Miscellaneous		
General		
() Libraries		
Miscellaneous		
GNU ARM Cross Create Flash Image		
@ General		
GNU ARM Cross Print Size		
A General		
	Do not use standard start files (-nostartfiles)	
	Do not use default libraries (-nodefaultlibs)	
	No startup or default libs (-nostdlib)	
	Remove unused sections (-Xlinkerqc-sections)	
	Print removed sections (-Xlinkerprint-gc-sections)	
	Omit all symbol information (-s)	

Figure 2-7 Cross ARM C Linker Configuration

Cross ARM GNU Create Flash Image Configuration

Select " Cross ARM GNU Create Flash Image > General > Output file format (-O)" to configure the option as "Raw binary" and generate software programming design BIN file, as shown in Figure 2-8.

Figure 2-8 Cross ARM GNU Create Flash Image Configuration



Devices Configuration

Select " Devices > Devices" and configure the option as "ARM Cortex M1 > ARMCM1", as shown in Figure 2-9.

Figure 2-9 Devices (Configuration
----------------------	---------------

Tool Settir	ngs 🛞 Toolchains	; 📕 Devices	🎤 Build Steps	🚇 Build Artifact	🗟 Binary Parsers	4
Device sele	ction (Used by deb	ug. Not yet use	ed during build!)			
Name		Details				^
>	ARM Cortex A5	Family (2	048 kB RAM, 204	8 kB ROM)		
>	ARM Cortex A7	Family (2	048 kB RAM, 204	8 kB ROM)		
>	ARM Cortex A9	Family (2	048 kB RAM, 204	8 kB ROM)		
>	ARM Cortex M0	Family (1	28 kB RAM, 256	kB ROM)		
>	ARM Cortex M0 pl	us Family (1	28 kB RAM, 256	kB ROM)		
~	ARM Cortex M1	Family (1	28 kB RAM, 256	kB ROM)		
	ARMCM1	Device (C	Cortex-M1, Rev r1	p0, 10 MHz)		
>	ARM Cortex M23	Family (2	56 kB RAM, 4096	kB ROM)		۷
Device core Memory ma ARMCM1	:: Cortex-M1 ap (Warning: Not y	et used to gen	erate the linker se	cripts!)		
Section	Start	Size	Startup			
IRAM1	0x20000000	0x00020000	0			
IROM1	0x00000000	0x00040000	1			
Edit						

2.2.3 Build

After project option configuration and coding, click "Build" ($^{\circ}$) or "Build All" ($^{\circ}$) on the tool bar, or select "Project > Build Project" or "Project > Build All" on the menu bar to generate software design BIN file, as shown in Figure 2-10.

Figure 2-10 Build



2.2.4 Download

After building Gowin_EMPU_M1software programming design, for the downloading, see <u>IPUG532, Gowin_EMPU_M1Download Reference</u> <u>Manual.</u>

2.2.5 Software Online Debug

After downloading the Gowin_EMPU_M1 software programming design BIN file, if there is a problem with your software design, you can connect the development board to the J-LINK emulator and debug the current software design online (the online debugging software design must be consistent with the software design downloaded to the chip).

Note!

Gowin_EMPU_M1 does not support automatic downloading during debugging with ARM Keil software. Prior to each debugging session, please use the Programmer tool to download the Binary file of the software design you intend to debug. Then, start debugging to ensure that the software project being debugged is the current one.

Gowin_EMPU_M1 software online debugging process includes:

- Configure software debugging options
- Configure software debugging levels
- Connect debugging emulators
- Start software online debugging

Software Debugging Configurations

 As shown in Figure 2-11, select "Run > Debug Configurations > GDB SEGGER J-Link Debugging > New" to create the debug configuration option of current project.

Figure 2-11 Create Software Debugging Configurations Option

🐳 Debug Configurations	×
Create, manage, and run configurations	to.
Image: State St	Configure launch settings from this dialog:
 C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugger C/C++ Remote Application GDB Hardware Debugging GDB OpenOCD Debugging 	 Press the 'Duplicate' button to copy the selected configuration. Press the 'Delete' button to remove the selected configuration. Press the 'Filter' button to configure filtering options. Edit or view an existing configuration by selecting it.
C GDB PyOCD Debugging C GDB QEMU Debugging C GDB SEGGER J-Link Debugging	Configure launch perspective settings from the <u>'Perspectives'</u> preference page.
Filter matched 10 of 16 items	e
(?)	Debug Close

2. Select "Main" option in the created software debugging options to configure "Project" and "C/C++ Application" options of current debugging project, as shown in Figure 2-12.

Figure	2-12	Main	Ontion	Config	iration
riguie	2-12	wiam	Option	Comig	aration

🐳 Debug Configurations			×
Create, manage, and run config	urations		Ť.
Image: Second Secon	Name: cm1_demo Debug	Browse	
	Build (if required) before launching Build Configuration: Select Automatically O Enable auto build © Use workspace settings	Variables Search Project	B <u>r</u> owse
Filter matched 11 of 17 items		Re <u>v</u> ert	Apply
?		Debug	Close

- 3. Select the "Debugger" option of the created software debugging options to configure the J-Link and GDB options of the current debugging project, as shown in Figure 2-13.
 - Device Name: Cortex-M1
 - Interface: JTAG or SWD
 - Endianness: Little
 - Connection: USB

1 🗈 🗶 🖪 🏇 🗸	Name ant dama D	ahua							
vpe filter text		ebug	.) 5-0						
C/C++ Application	Main >> Debugg	jer 🔰 📂 Sta	irtup 🐶 Sour	ce 🔟 <u>C</u> om	imon 🚠 SVD Pa	ath			
C/C++ Attach to Applica	Start the Llink	GDB server	locally		Connect	to running tar	tet		
C/C++ Postmortem Deb		ACT 1				to running targ			
C/C++ Remote Applicati	Executable path:	\${jlink_path	}/\${jlink_gdbse	erver}			Brow		
C GDB Hardware Debuggi	Actual executable:	C:/Program	Files (x86)/SE	GGER/JLink	c/JLinkGDBServer	rCL.exe			
GDB OpenOCD Debuggi		(to change it use the <u>global</u> or <u>workspace</u> preferences pages or the <u>pr</u>							
GDB QEMU Debugging	Device name:	Cortex-M1					Suppo		
GDB SEGGER J-Link Deb	Endianness:	Little	OBig						
💽 cm1_demo Debug	Connection:	USB	OIP			(USB serial or	IP nam		
Launch Group	Interface:	SWD	⊖ JTAG						
	Initial speed:	Auto	OAdaptive	• Fixed	1000 kHz				
	GDB port:	2331							
	SWO port:	2332			✓ Verify do	ownloads 🖂 II	nitialize		
	Telnet port:	2333			∠ Local ho	st only 🗌 S	ilent		
	Log file:								
	Other options: -singlerun -strict -timeout 0 -nogui								
	Allocate consol		Allocate cons	sole for semiho	osting a				
	<						>		
>									

Figure 2-13 Debugger Option Configuration

Software Debugging Level Configuration

In the Project Explorer view, select "Properties > C/C++ Build > Settings > Debugging > Debug level" option of the current debugging project, and recommend configuring the debugging level as Default(-g) or Maximum(-g3), as shown in Figure 2-14.

e ®	Tool	Settings	No Toolch	ains	Devices	Build Steps	en Buil	d Artifact	Binary P	a + +		
_			•						<u> </u>			
	🖄 Target Processor					Debug level		Maximum	(-g3)	\sim		
	Optimization					Debug format		Toolchain	default	\sim		
	22	Warnings	;									
	23	Debuggir	ng			Generate prof information (-p)						
		Cross AR	IVI GINU ASS	empler		Generate gpi	rot intorr	nation (-p	g)			
		Prepro	ocessor			Other debuggir	ig flags					
		🖉 includ	les									
		Misco	Ilanaour									
			M C Compil	or								
		Drenn	ocessor									
		🖄 Includ	les									
		🖄 Optim	nization									
		🖄 Warni	ings									
		🖄 Misce	llaneous									
	/ 🛞	Cross AR	M C Linker									
		🖄 Gener	ral									
		🖄 Librar	ies									
		🖄 Misce	llaneous									
	/ 📎	Cross AR	M GNU Cre	ate Flasl	h Image							
		🖄 Gener	ral									
1	/ 🛞	Cross AR	M GNU Prin	t Size								
		🖄 Gener	ral									

Figure 2-14 Software Debugging Level Configuration

Software Online Debugging Start-up

According to the physical constraints location of JTAG debugging interface (JTAG: JTAG_3~JTAG_18, VCC and GND; or SWD: JTAG_7, JTAG_9, VCC and GND) in the hardware design, connect the J-LINK emulator and the development board.

Click "Debug" button in the tool bar to drop the list "* ", select the current project Debug configuration, click to enter the debug state, perform breakpoint settings, single-step debugging, reset and run, etc., as shown in Figure 2-15.

inguie - 10 solutione simile Desag	8-	ing start up					
workspace_cm1 - Debug - cm1_demo/src/project/led/led_demo.c - GOW		ICU Designer			- 0		Х
File Edit Source Refactor Navigate Search Project Run Window	He	lp					
. 🗂 🕶 🔚 🐚 👪 💌 💷 🛤 🗛 👁 . 🖉 🍻 🔜 🗷 🕹 ! 🐂	Ļ	🎋 • 🔘 • 🍅 🛷 •	1 2 -	🖗 • 🌤 🔶 • 🔿	-		
				Quick Acc	ess 🕴 😭		ゃ
🎋 Debug 🛛 🦌 🙀 🗸 🖻		(x)= Variab 🛛 💁 Breal	k 1999 Regis	st 🕆 Periph 🛋	Modules	-	
✓		Name	Туре	∛iii ⇒ Value	i 🗆 📬	ď	▽
Thread #1 57005 (Suspended : Step)							
main() at main s(64.0x5a2							
							t,
	~	<					>
€ led_demo.c ⊠ € main.c			- 8	🗄 Outline 🖾			
24 25 int led_demo(void) 26 { 27 SystemInit(); //Initializes system clock 28 gpio_init(); //Initializes GPIO0 29 delay_init(); //Initializes delay functions 30 while(1) 31 while(1) 32 { 33 GPIO_WriteBits(GPIO0,0xE); //led1 34 delay_sec(1); //Delay 1 second 35 GPIO_WriteBits(GPIO0,0xD); //led2 36 delay_sec(1); 37 GPIO_WriteBits(GPIO0,0xB); //led3 38 delay_sec(1); 39 delay_sec(1); 30 delay_sec(1); 31 delay_sec(1); 32 delay_sec(1); 33 delay_sec(1); 34 delay_sec(1); 35 delay_sec(1); 36 delay_sec(1); 37 delay_sec(1); 38 delay_sec(1); 39 delay_sec(1); 30 delay_sec(1); 31 delay_sec(1); 33 delay_sec(1); 34 delay_sec(1); 35 delay_sec(1); 36 delay_sec(1); 37 delay_sec(1); 38 delay_sec(1); 39 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 31 delay_sec(1); 33 delay_sec(1); 34 delay_sec(1); 35 delay_sec(1); 36 delay_sec(1); 37 delay_sec(1); 38 delay_sec(1); 39 delay_sec(1); 39 delay_sec(1); 39 delay_sec(1); 39 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 31 delay_sec(1); 32 delay_sec(1); 33 delay_sec(1); 34 delay_sec(1); 35 delay_sec(1); 36 delay_sec(1); 37 delay_sec(1); 38 delay_sec(1); 39 delay_sec(1); 39 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 31 delay_sec(1); 32 delay_sec(1); 33 delay_sec(1); 34 delay_sec(1); 35 delay_sec(1); 36 delay_sec(1); 37 delay_sec(1); 38 delay_sec(1); 39 delay_sec(1); 39 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 31 delay_sec(1); 32 delay_sec(1); 33 delay_sec(1); 34 delay_sec(1); 35 delay_sec(1); 36 delay_sec(1); 37 delay_sec(1); 38 delay_sec(1); 38 delay_sec(1); 39 delay_sec(1); 39 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 30 delay_sec(1); 31 delay_sec(1); 31 delay_sec(1); 32 delay_sec(1); 33 delay_sec(1); 34 delay_sec(1); 35 delay_sec(1); 36 delay_sec(1); 37 delay_sec(1); 37 delay_sec(1); 38 delay_s				 □ [a] 2 □ demo.h □ gpio.h □ delay.h ○ led_demoi 	R ≥ ^S ●	推	~
😑 Console 🛛 🖉 Tasks 💦 Problems 🕡 Executables 📋 Memory			८ 🔆 🖳 🛓	I 🛯 🖓 💭 🛃 🖬) - 📑 -	-	
cm1_demo Debug [GDB SEGGER J-Link Debugging] JLinkGDBServerCL.exe							
Removing breakpoint @ address 0x00000588, Size = 2 Removing breakpoint @ address 0x00000CB4, Size = 2 Reading 64 bytes @ address 0x20007FC0 Read 4 bytes @ address 0x20007E2 (Data = 0x00182300)							
<							>
			1				:6

Figure 2-15 Software Online Debugging Start-up

2.3 Reference Design

Gowin_EMPU_M1 provides reference design in GMD (tested software version V1.2) software environment. Click this <u>link</u> to get following reference design:

...\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo、 cm1_fatfs、cm1_freertos、cm1_rtthread_nano、cm1_ucos_iii

