

Gowin_EMPU(GW1NS-4C) IDE 软件



IPUG928-1.1,2021-02-08

版权所有© 2021 广东高云半导体科技股份有限公司

未经本公司书面许可,任何单位和个人都不得擅自摘抄、复制、翻译本文档内容的部分或全部,并不得以任何形式传播。

免责声明

本文档并未授予任何知识产权的许可,并未以明示或暗示,或以禁止发言或其它方式授予任 何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外,高云半导体 概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和/或使用不作任何 明示或暗示的担保,包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知 识产权的侵权责任等,均不作担保。高云半导体对文档中包含的文字、图片及其它内容的准 确性和完整性不承担任何法律或非法律责任,高云半导体保留修改文档中任何内容的权利, 恕不另行通知。高云半导体不承诺对这些文档进行适时的更新。

版本信息

日期	版本	说明
2020/04/20	1.0	初始版本。
2021/02/08	1.1	 支持外部设备 AHB PSRAM Memory Interface; 支持外部设备 AHB HyperRAM Memory Interface; 支持外部设备 APB SPI_Nor_Flash; GPIO 支持多种端口类型配置; I²C 支持多种端口类型配置; 升级软件版本以及软件开发工具包。

目录

目	录i
图	目录
1	ARM Keil MDK 软件1
	1.1 软件安装1
	1.2 工程模板1
	1.2.1 创建工程1
	1.2.2 配置编译选项
	1.2.3 编译
	1.2.4 下载
	1.2.5 在线调试
	1.3 参考设计12
2	GOWIN MCU Designer13
	2.1 软件安装13
	2.2 工程模板13
	2.2.1 创建工程13
	2.2.2 配置编译选项
	2.2.3 编译
	2.2.4 下载
	2.2.5 在线调试
	2.3 参考设计

图目录

图 1-1 创建工程	1
图 1-2 配置器件	2
图 1-3 配置 ROM 和 RAM	3
图 1-4 配置输出文件格式	4
图 1-5 配置头文件路径	5
图 1-6 配置 Flash 选项	5
图 1-7 编译	6
图 1-8 器件 GW1NS-4C/GW1NSR-4C 下载选项配置	7
图 1-9 器件 GW1NSER-4C 下载选项配置	7
图 1-10 器件 GW1NS-4C/GW1NSR-4C MCU JTAG 模式切换配置	9
图 1-11 器件 GW1NSER-4C MCU JTAG 模式切换配置	9
图 1-12 配置仿真器类型	10
图 1-13 配置调试接口类型	11
图 1-14 启动软件调试	12
图 2-1 创建工程	14
图 2-2 选择平台类型配置	14
图 2-3 选择工具链和路径	15
图 2-4 配置 Target Processor 选项	16
图 2-5 配置 Cross ARM GNU Assembler > Preprocessor 选项	17
图 2-6 配置 Cross ARM C Compiler > Includes 选项	17
图 2-7 配置 Cross ARM C Linker 选项	18
图 2-8 配置 Cross ARM GNU Create Flash Image 选项	19
图 2-9 配置 Devices 选项	19
图 2-10 编译	20

图 2-11 下载软件 Programmer 选项	21
图 2-12 器件 GW1NS-4C/GW1NSR-4C 下载选项配置	
图 2-13 器件 GW1NSER-4C 下载选项配置	
图 2-14 配置软件调试等级	24
图 2-15 建立软件调试配置选项	24
图 2-16 配置 Main 选项	25
图 2-17 配置 Debugger 选项	
图 2-18 器件 GW1NS-4C/GW1NSR-4C MCU JTAG 模式切换配置	27
图 2-19 器件 GW1NSER-4C MCU JTAG 模式切换配置	27
图 2-20 启动软件调试	

1 ARM Keil MDK 软件

1.1 软件安装

请参考 ARM Keil MDK(V5.26 及以上版本)网站提供的 <u>Getting Started</u> <u>with MDK</u>。

1.2 工程模板

使用 ARM Keil MDK 软件进行 Gowin_EMPU(GW1NS-4C)软件编程设计,需要创建工程、配置编译选项、编写代码、编译、下载和在线调试。

1.2.1 创建工程

双击打开 ARM Keil MDK 软件,选择菜单栏 "Project > New uVision Project...",创建工程,如图 1-1 所示。

🐺 μVision File Edit View Project Flash Debug Peripherals Tools SVCS Window Help New µVision Project.. 🗋 💕 🖌 🧭 New Multi-Project Workspace... ۵ 🔛 🔛 🧳 Open Project... Project Close Project Export Manage Select Device for Target ... Remove Item N Options... Clean Targets Build Target Rebuild all target files Batch Build Batch Setup... S Translate... 🗒 Stop build

图 1-1 创建工程

1.2.2 配置编译选项

配置器件

Gowin_EMPU(GW1NS-4C)内置 ARM Cortex-M3 MCU 内核, 配置器 件为 "ARM Cortex M3 > ARMCM3", 如图 1-2 所示。

图 1-2 配置器件

Device Target Output Listing Vser	C/C++ Asm Linker Debug Vtilities
Software Packs Vendor: ARM Device: ARMCM3 Toolset: ARM Search:	Software Pack Pack: ARM.CMSIS.5.6.0 URL: <u>http://www.keil.com/pack/</u>
ARM ARM Cortex M0 ARM Cortex M0 plus ARM Cortex M1 ARM Cortex M23 ARM Cortex M3 ARM Cortex M33 ARM Cortex M33 ARM Cortex M35P ARM Cortex M4	The Cortex-M3 processor is an entry-level 32-bit Am Cortex processor designed for a broad range of embedded applications. It offers significant benefits to developers, including: - simple, easy-to-use programmers model - highly efficient ultra-low power operation - excellent code density - deterministic, high-performance interrupt handling - upward compatibility with the rest of the Cortex-M processor family.
ОК	Cancel Defaults Help

配置 ROM 和 RAM

配置 ROM(指令存储器, FLASH)和 RAM(数据存储器, SRAM)的 起始地址和 Size,如图 1-3 所示。

ROM 配置:

- 起始地址: 0x0000000
- Size: 0x8000 (32KB)

RAM 配置:

- 起始地址: 0x2000000
- Size: 与 Gowin 云源软件的 IP Core Generator 中的 SRAM Size 配置保持一致,可以配置为 2KB、4KB、8KB 或 16KB
- 以软件开发工具包 Keil_RefDesign 参考设计为例, RAM Size 配置为 0x4000 (Gowin 云源软件的 IP Core Generator > Gowin_EMPU(GW1NS-4C) > SRAM Size 配置为 16KB)

Device	[arget]	Output List	ing Vser	C/C++ A	sm I	inker :	Debug Vtili	ties	
ARM ARM	ICM3		_		-Code G ARM	eneration	Use default	compiler versio	on 5 🔻
_			<u>X</u> tal (MHz): 1	2.0					
Operating	system:	None		-	🗆 🗆 U:	se Cross-N	Module Optimizat	tion	
System Vi	iewer File:				U:	se MicroL	IB L	Big Endian	
🔲 Use (Custom Fil	e							
Read/0	Only Memo	ry Areas ——			-Read/	Write Men	nory Areas		
default	off-chip	Start	Size	Startup	default	off-chip	Start	Size	NoInit
	ROM1:			0		RAM1:			
	ROM2:			- C		RAM2:			
	ROM3:					RAM3:			
	on-chin	,	,			on-chin	, 	,	
	IROM1:	0×0	0×8000	¢		IRAM1:	0x20000000	0x4000	
	IROM2:			0		IRAM2:			
			OK	Can	cel	Dei	faults		Help

图 1-3 配置 ROM 和 RAM

配置输出文件格式

Gowin 云源软件的 Programmer 下载软件,支持软件编程设计二进制 BIN 文件下载,配置 Gowin_EMPU(GW1NS-4C)输出文件格式为二进制 BIN 文件格式。

User 命令行选项中,使用格式转换工具 fromelf 转换输出文件 axf 格式为 bin 格式,调用方法如图 1-4 所示。

fromelf 格式转换命令为: fromelf.exe --bin -o bin-file axf-file。例如, C:\Keil v5\ARM\ARMCC\bin\fromelf.exe --bin -o led.bin .\Objects\led.axf。

格式转换工具 fromelf 的路径,请根据用户本地 ARM Keil MDK 安装路 径进行修改。

图 1-4 配置输出文件格式

Device Target Output Listin	g User C/C++ Asm Linker Debug Ut	ilit	ies	
Command Items	User Command		Stop on Exi	S
-Before Compile C/C++ File				
Run #1		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	D:\Keil_v5\ARM\ARMCC\bin\fromelf.exebin	2	Not Specified	
Run #2		2	Not Specified	
Run 'After-Build' Conditionally				
✓ Beep When Complete	Start Debugging			
	OK Cancel Defaults		Не	elp

配置头文件路径

配置 C 头文件路径,编译过程中用来调用不同路径的 C 头文件,配置 如图 1-5 所示。

以软件开发工具包 Keil_RefDesign 参考设计为例, C 头文件路径配置, 如下所示。

- "..\CORE"
- "..\PERIPHERAL\Includes"
- "..\SYSTEM"
- "..\USER"

图 1-5 配置头文件路径

Device Target Output Listing User	C/C++ Asm Linker Debug Utili	ties	
Preprocessor Symbols			
Define:			
U <u>n</u> define:			
Language / Code Generation			
Execute-only Code	Strict ANSIC Warnings:	All Warnings 🗨	
Optimization: Level 0 (-00)	Enum Container always int	Thum <u>b</u> Mode	
Optimize for Time	Plain Char is Signed	No Auto Includes	
Split Load and Store Multiple	□ Read-Only Position Independent ✓ C99 Mode		
✓ One ELF Section per Function	<u>R</u> ead-Write Position Independent	GNU extensions	
Include\CORE;\PERIPHERAL\Inclu Paths	ides;\SYSTEM;\USER		
Controls			
Compilerc99 -ccpu Cortex-M3 -li -g -C	00apcs=interworksplit_sections -I/CORE -I	^	
string	STSTEM -1/USER	~	
ОК	Cancel Defaults	Help	

配置 Flash 选项

请关闭 "Utilities > Update Target before Debugging"选项,如图 1-6 所示。

图 1-6 配置 Flash 选项

Device Target Output Listing Vser C/C++	Asm Linker Debug Vtilities			
Configure Flash Menu Command				
Use Target Driver for Flash Programming	Vse Debug Driver			
Use Debug Driver	Settings Update Target before Debugging			
Init File:	Edit			
O Use External Tool for Flash Programming				
Command:				
Arguments:				
🗖 Run Independent				
Configure Image File Processing (FCARM):				
Output File:	Add Output File to Group:			
	CORE			
Image Files Root Folder:				
OK	Cancel Defaults Help			

1.2.3 编译

完成代码编写和选项配置后,单击工具栏 "Build" (☑) 或 "Rebuild" (☑),编译产生二进制 BIN 文件,如图 1-7 所示。

图 1-7 编译

🖉 🛄 🖾 🎯 🕶 📖 🗰 Ied		
Project 🛛 🕂 🗵	main.c	•
E ⁴ ³ Project: led ▲	13 */	
🖻 🔛 led		
I CORE	15 /* Includes	S7
😑 🦢 STARTUP	17	
startup_gw1ns4c.s	18 /* Declarations	*/
🖨 🦢 SYSTEM	<pre>19 void delay_ms(IO uint32_t delay_ms);</pre>	
	20 void GPIOInit(void);	
😑 🦢 PERIPHERAL	21	
m aw1ns4c apio.c	22 /> functions	- ^ /
e awins4c i2c.c		
core cm3 h	25 SystemInit(); //Initializes system	
gwinsis h	26 GPIOInit(); //Initializes GPIO	
gwinstein	27	
gwins4c_contai	28 while (1)	
gwins4c_gpio.n	30 GPTO ResetBit (GPTOD GPTO Pin 0) · //IED1 on	
gwIns4c_i2c.h	31 delay ms(500);	
gw1ns4c_misc.h	32	
gw1ns4c_rtc.h	33 GPIO_SetBit(GPIO0,GPIO_Pin_0); //LEDl off	
gw1ns4c_spi.h	34 delay_ms(500);	
gw1ns4c_syscon.	35 - }	
aw1ns4c timer.h	37	
	39 //Thirislizes GDTO	
E Project 🕝 Books {} Func 🛛 🕁 Temp	<	>
uild Output		ą
ompiling gwlns4c_spi.c		
compiling gwlns4c_timer.c		
ompliing gwins4c_syscon.c		
compiling gwlms4c wdog.c		
compiling gwlns4c it.c		
ompiling gwlns4c uart.c		
compiling main.c		
linking		
frogram Size: code=812 RO-data	=224 KW-data=16 21-data=1632	
<pre>.\Objects\led.axf" = 0 Error(</pre>	s). O Warning(s).	
Muild Time Elapsed: 00:00:02	-//	

1.2.4 下载

使用 Gowin 云源软件的 Programmer 下载软件,下载软件编程设计二进制 BIN 文件。

在 Gowin 云源软件中或安装路径下,打开下载软件 Programmer。

单击 Programmer 菜单栏 "Edit > Configure Device" 或工具栏 "Configure Device" (歐), 打开 Device configuration。

如果选用器件 GW1NS-4C/GW1NSR-4C,下载选项配置,如图 1-8 所示。

- Access Mode 下拉列表,选择"MCU Mode"选项。
- Operation 下拉列表,选择"Firmware Erase, Program"选项或"Firmware Erase, Program, Verify"选项。

🙀 Device configuration		?	\times	
Device Operation				
Access Mode:	MCV Mode		-	
Operation:	Firmware Erase, Program		•	
Firmware Erase, Progra	л			
Programming Options				
File name: E:/desktop/mcu_test/m3_4c/gowin_empu_QN48G.fs				
User Flash Initialization				
- FW/MCU/Binary Input Options				
Firmware/Binary File: [E:/desktop/mcu_test/m3_4c/led.bin			
	Save	Cano	el :	

图 1-8 器件 GW1NS-4C/GW1NSR-4C 下载选项配置

如果选用器件 GW1NSER-4C,下载选项配置,如图 1-9 所示。

- Access Mode 下拉列表,选择"SecureFPGA Mode"选项。
- Operation 下拉列表, 选择"Firmware Erase, Program Securely"选项。

图 1-9 器件 GW1NSER-4C 下载选项配置

🙀 Device configuration

×

?

Device Operation				
Access Mode:	e: SecureFPGA Mode 💌			
Operation:	Firmware Erase, Program Securely 🔹			
Firmware Erase, Program, Verify with Security(One Time Programmable For Key Authentication Code)				
- Programming Options	Programming Options			
File name: C:/Users/Desktop/secure_fpga/impl/pnr/secure_fpga.fs				
User Flash Initialization				
FW/MCU/Binary Input Options				
Firmware/Binary File: ER_4C/Debug/bk_with_no_enroll_GW1NSER_4C.bin				
	Save Cancel			

- "FW/MCU/Binary Input Options > Firmware/Binary File"选项,导入
 Gowin EMPU(GW1NS-4C)软件编程设计二进制 BIN 文件。
- 单击 "Save", 完成下载选项配置。

注!

"Programming Options > File name"选项,导入 Gowin_EMPU(GW1NS-4C)硬件设计码 流文件,请参考 IPUG932, Gowin_EMPU(GW1NS-4C)硬件设计参考手册。

完成 Device configuration 后,单击 Programmer 工具栏 "Program/Configure" (事),完成 Gowin_EMPU(GW1NS-4C)的下载。

1.2.5 在线调试

完成 Gowin_EMPU(GW1NS-4C)软件下载后,如果用户软件设计出现问题,可以连接开发板与 U-LINK 或 J-LINK 仿真器,在线调试 MCU 软件设计。

Gowin_EMPU(GW1NS-4C)软件调试流程,包括:

- MCU JTAG 模式切换
- MCU JTAG 接口切换
- 配置软件调试选项
- 连接调试仿真器
- 启动软件调试

MCU JTAG 模式切换

使用下载软件 Programmer,将 Gowin_EMPU(GW1NS-4C)的 MCU JTAG 模式,由下载模式切换到调试模式。

在 Gowin 云源软件中或安装路径下,打开下载软件 Programmer。

单击 Programmer 菜单栏 "Edit > Configure Device"或工具栏 "Configure Device" (☞),打开 Device configuration。

如果选用器件 GW1NS-4C/GW1NSR-4C, MCU JTAG 模式切换配置, 如图 1-10 所示。

- Access Mode 下拉列表,选择"MCU Mode"选项。
- Operation 下拉列表,选择 "Connect to JTAG of MCU"选项。

🎄 Device configurati	on	?	×
-Device Operation			
Access Mode:	MCV Mode		•
Operation:	Connect to JTAG of MCU		•
Connect to JTAG o	£MCV.		

图 1-10 器件 GW1NS-4C/GW1NSR-4C MCU JTAG 模式切换配置

如果选用器件 GW1NSER-4C, MCU JTAG 模式切换配置, 如图 1-11 所示。

- Access Mode 下拉列表,选择"SecureFPGA Mode"选项。
- Operation 下拉列表,选择 "Connect to JTAG of MCU"选项。

图 1-11 器件 GW1NSER-4C MCU JTAG 模式切换配置

💸 Device configuratio	n	? >	<
- Device Operation			٦
Access Mode:	SecureFPGA Mode	•	
Operation:	Connect to JTAG of MCU	•	
Connect to JTAG of	Ê MCU.		
	Save	Cancel	

● 单击 "Save", 完成 MCU JTAG 模式切换配置。

完成 Device configuration 后,单击 Programmer 工具栏 "Program/Configure" (事),完成 MCU JTAG 模式切换。

MCU JTAG 接口切换

以软件开发工具包 DK-START-GW1NSR4C-QN48G V1.1 开发板为例。

手动将开发板上 SW3、SW4、SW5、SW6 拨码开关(JTAG 接口 TCK、 TDO、TDI 和 TMS)由 "FT232"(Download)切换为 "J-LINK"(Debug)。

注!

• 在 MCU JTAG 模式切换和 MCU JTAG 接口切换过程中,必须保持开发板上电状态, 不可断电。 • 如果中途开发板断电,重新上电后自动还原为 MCU JTAG 下载模式。

配置软件调试选项

双击打开 ARM Keil MDK 软件,选择工具栏 Options for Target…"▲", 打开 Options for Target, 配置"Debug"选项。

● 配置仿真器类型

单击 Debug 仿真器类型下拉列表,配置所用仿真器类型,如图 1-12 所示。

- U-LINK 仿真器 如果选择使用 U-LINK 仿真器,则选择 "ULNK2/ME Cortex Debugger"。
- J-LINK 仿真器 如果选择使用 J-LINK 仿真器,则选择"J-LINK/J-TRACE Cortex"。

图 1-12 配置仿真器类型

Device Target Output Listing User C/C++ .	Asm Linker Debug Utilities
C Use Simulator with restrictions Settings	G Use: J-LINK / J-TRACE Cortex ▼ Settings
Limit Speed to Real-Time	ULINK2/ME Cortex Debugger
✓ Load Application at Startup ✓ Run to main()	ULINKplus Debugger
Initialization File:	Initializatic J-LINK / J-TRACE Cortex
Edit	Medele Cortex M Debugger
Parters Dahus Castien Callines	PENicro Debugger
	NULink Debugger
I Breakpoints I Ioolbox	I → Bra Stellans ICD1
✓ Watch Windows & Performance Analyzer	Vivatch Windows
I♥ Memory Display I♥ System viewer	J♥ Memory Display J♥ System viewer
CPU DLL: Parameter:	Driver DLL: Parameter:
SARMCM3.DLL -MPU	SARMCM3.DLL -MPU
Dialog DLL: Parameter:	Dialog DLL: Parameter:
DCM.DLL pCM3	TCM.DLL PCM3
Wam if outdated Executable is loaded	Warn if outdated Executable is loaded
Manage Component Vi	ewer Description Files
OK Car	ncel Defaults Melp

● 配置调试接口类型

单击 Settings,如果选用 J-LINK,则打开 Cortex JLink/JTrace Target Driver Setup,如果选用 U-LINK,则打开 ULINK2/ME Cortex-M Target Driver Setup。

选用 J-LINK 仿真器,如图 1-13 所示。

- JTAG 调试接口

Gowin_EMPU(GW1NS-4C)支持 JTAG 调试接口, 配置 Port 选项为 "JTAG"。

- SW 调试接口

Gowin_EMPU(GW1NS-4C)不支持 SW 调试接口,请不要选用 SW 调试接口。

图 1-13 配置调试接口类型

ortex JLink/JTrace Target Driver Setup	
Debug Trace Flash Download	
J-Link / J-Trace Adapter	_JTAG Device Chain
SN: 805251123 -	IDCODE Device Name IR len Move
Device: J-Link ARM	TDO Ox4BA00477 ARM CoreSight JTAG-DP 4
HW : V8.00 dll : V6.94	TDI Covm
FW : J-Link ARM V8 compiled No Port: Max	Automatic Detection ID CODE:
Auto Clk	Add Delete Update IR len:
Connect & Reset Options Connect: Normal Reset: N <u>Reset after Connect</u>	ormal ▼ Cache Options Download Options Cache <u>C</u> ode Cache <u>C</u> ode Cache <u>M</u> emory Download to <u>F</u> lash
C USB O TCP/IP C USB O TCP/IP Scan State: ready	Settings Port (Auto: Autodetect 0 0 1 0 1 : 0 Ping
	确定 取消 应用(A)

Download Options 配置选项, 请关闭 "Verify Code Download" 选项和 "Download to Flash" 选项。

如果调试选项配置成功,在 MCU JTAG 模式切换、MCU JTAG 接口切换以及连接仿真器后,JTAG Device Chain 可以正确显示 Gowin_EMPU(GW1NS-4C)的 IDCODE、Device Name 等信息。

启动软件调试

完成软件调试选项配置后,连接 U-LINK 或 J-LINK 调试仿真器。

单击工具栏 Debug "Q",进入软件调试状态。

可以进行断点设置、单步调试、复位和运行等操作,如图 1-14 所示。

图 1-14 启动软件调试

<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>I</u>	roject Fl <u>a</u> sh	Debug Peripherals Tools SVCS Window Help	
🗋 💕 🖬 🖉 🗌	x 🖻 🛍 🛛	9 ℃ ← → 整 整 整 後 字 症 //2 //2 @ GPIO 🛛 🗟 🖋 🔍 🔹 🔗 🔗 🍓 🖬 • 🔦	
👫 📃 🚳 🖓	0+ {}+ *()		
Registers	д 🗵	Disassembly	д 🗙
Registers Register Core Ri Ri	0 2 Yalue 0x2000374 0x0000000 0x0000000 0x0000000 0x0000000 0x0000000 0x0000000 0x0000000 0x0000000 0x00000000 0x0000000 0x00000000 0x0000010 0x00000000 0x0000100 0x00000100 0x0000100 0x00000100 0x0000100 0x00000100 0x00000100 0x00000100 0x00000100 Thread Privilseed MSP 981459 0.08814590 0.08814590	<pre>Disssembly Dissembly Dissembly Disssembly Dissembly</pre>	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Im Project Regist Command Load "C:\\User * JLink Info: F	ers //liukai/ll	36 } 37 //Initializes GPI0 39 void GPI0Init(void) 40 CHI0 TaitTurol (chi0 TaitTuro) 41 CHI0 TaitTurol (chi0 TaitTuro) 41 CHI0 TaitTurol (chi0 TaitTuro) 41 CHI0 TaitTurol (chi0 TaitTuro) 5 Call Stack + Locals Name Location/Value Type • Occore after reset via DENCR.VC_COREREST. • • • main	> 2 2
* JLink Info: H	leset: Reset	t device via AIRCR.SYSRESETREQ.	
<		>	
>			
ASSIGN BreakDis	able Break	Enable BreakKill BreakList BreakSet BreakAccess 🛛 🚱 Call Stack + Locals 🗖 Memory 1	
		J-LINK / J-TRACE Cortex t1: 0.09814590 sec L:26 C:1 C	AP NUM SCRL OVR I .:

1.3 参考设计

Gowin_EMPU(GW1NS-4C)支持 ARM Keil MDK(V5.26 及以上版本) 软件环境的参考设计,通过此链接获取如下参考设计: <u>cdn.gowinsemi.com.cn/Gowin_EMPU_V1.1.zip</u>

 $Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign$

$2_{\text{GOWIN MCU Designer}}$

2.1 软件安装

高云半导体网站提供 GOWIN MCU Designer (V1.1 及以上版本) 软件 安装包下载 <u>http://www.gowinsemi.com.cn/prodshow.aspx</u>。

注!

GOWIN MCU Designer (V1.1 及以上版本) 软件安装与配置,请参考 <u>SUG549</u>, *GOWIN MCU Designer 用户指南*。

2.2 工程模板

使用 GOWIN MCU Designer 软件进行 Gowin_EMPU(GW1NS-4C)软件 编程设计,需要创建工程、配置编译选项、编写代码、编译、下载和在线调 试。

2.2.1 创建工程

创建工程

选择菜单栏 "File > New > C Project",如图 2-1 所示。

- 建立项目名称和位置;
- 选择项目类型 Empty Project;
- 选择工具链 ARM Cross GCC。

图 2-1 创建工程

W C Project			_		×
C Project Create C project of selected typ	e				\$
<u>P</u> roject name: gowin_led ☑ Use <u>d</u> efault location					
Location: C:\GMD\workspace\g	gowin_led			B <u>r</u> owse.	
Choose file system: o Project type: Executable Empty Project Hello World ARM C Hello World RISC-V Show project types and tool	Project C Project c hains only if the	Toolchains: ARM Cross GCC RISC-V Cross GC	C e platform		
?	< <u>B</u> ack	<u>N</u> ext >	Einish	Cance	1

选择平台配置类型

选择平台配置类型"Debug"和"Release",如图 2-2 所示。

图 2-2 选择平台类型配置

W C Project							\times
Select Configura	ations						\diamond
Select platforms	and configura	tions you wish	to deploy on				
Project type:	Executable						
Toolchains:	ARM Cross G	CC					
Configurations:							
🗹 🛞 Debug					Sele	ect all	
M 🛞 Release					Dese	lect all	
					Advanced	d setting	ys
Use "Advanced Additional confi	settings" buttor gurations can b	n to edit proje be added after	ct's properties.				
ose manage d	onigurations c	outtons either	on toolbar or on	property p	ages.		
?		< <u>B</u> ack	<u>N</u> ext >	<u>F</u> ini	sh	Cance	el

选择工具链和路径

选择交叉编译工具链 arm-none-eabi-gcc 及其所在路径,默认配置,如 图 2-3 所示。

图 2-3 选择工具链和路径

🐝 C Project			×
GNU ARM Cross	Toolchain ain and configure path		\$
Toolchain name:	GNU MCU Eclipse ARM Embedded GCC (arm-none-eabi-gcc)		~
Toolchain path:	C:\GMD\toolchain\ARM_toolchain\bin	 Brov	vse
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish	Cance	1

建立项目工程

完成工程创建后,选择 GOWIN MCU Designer 工作空间 workspace 下 新建的项目工程,添加工程结构和代码,导入软件编程设计。

以软件开发工具包 GMD_RefDesign 参考设计为例,软件编程设计项目 工程结构及代码如下所示。

- CORE: ARM Cortex-M3 MCU 内核定义
- PERIPHERAL: 外部设备驱动函数库
- STARTUP: MCU 内核引导启动文件
- SYSTEM: 寄存器定义、系统初始化和系统时钟定义
- USER: 用户应用设计
- gw1ns4c_flash.ld: GMD Flash 链接器

完成项目工程结构建立后,在 GOWIN MCU Designer 的 Project Explorer 视图中选择当前工程,右键选择 "Refresh"选项,自动更新当前项目工程的结构和代码。

2.2.2 配置编译选项

GOWIN MCU Designer 的 Project Explorer 视图中,选择当前工程,右

键选择 "Properties > C/C++ Build > Settings" 选项,配置当前工程的参数 选项。

配置 Target Processor 选项

配置"Target Processor"选项,该选项配置,如图 2-4 所示。

以软件开发工具包 GMD_RefDesign 参考设计为例, Target Processor 选项配置,如下所示。

- Arm family: cortex-m3
- Architecture: Toolchain default
- Instruction set: Thumb (-mthumb)
- Endianness: Toolchain default
- Unaligned access: Toolchain default

图 2-4 配置 Target Processor 选项

Tool Settings 🛞 Toolchains 📕 Devices	🎤 Build Steps 🖳	Build Artifact 🗟 Binary Parsers 🔄
🖄 Target Processor	ARM family	cortex-m3 v
🖄 Optimization 🆄 Warnings	Architecture	Toolchain default
Debugging	Instruction set	Thumb (-mthumb)
Weight Cross AKM GNU Assembler Preprocessor	Thumb interwork	k (-mthumb-interwork)
🖄 Includes 🖄 Warnings	Eloat ABI	Toolchain default
Miscellaneous	FPU Type	Toolchain default
Preprocessor	Unaligned access	Toolchain default
🖉 Includes 🖄 Optimization	AArch64 family	Generic (-mcpu=generic)
🖉 Warnings	Feature crc	Toolchain default
 Wiscellaneous Wiscellaneous Cross ARM C Linker 	Feature crypto	Toolchain default
🖄 General	Feature fp	Toolchain default
Miscellaneous	Feature simd	Enabled (+simd)
* Scross ARM GNU Create Flash Image Beneral	Code model	Small (-mcmodel=small)
Cross ARM GNU Print Size	Other target flags	rrict-align)
😁 General	5 -5-	L

配置 Cross ARM GNU Assembler > Preprocessor 选项

配置 "Cross ARM GNU Assembler > Preprocessor > Defined symbols" 选项,该选项配置为 "___STARTUP_CLEAR_BSS",如图 2-5 所示。

🛞 Tool Settings 🛞 Toolchains 🔳 Devices	🎤 Build Steps 🚇 Build Artifact 🗟 Binary Parsers 💽
 Target Processor Optimization Warnings Debugging Cross ARM GNU Assembler Preprocessor Includes Warnings Miscellaneous Cross ARM C Compiler Preprocessor Includes Optimization Warnings Marinings Miscellaneous 	✓ Use preprocessor □ Do not search system directories (-nostdinc) □ Preprocess only (-E) □ Defined symbols (-D) ② ③ ④ ④ ② ⑤ ② STARTUP_CLEAR_BSS
 Stores ARM C Linker General Libraries Miscellaneous Cross ARM GNU Create Flash Image General Cross ARM GNU Print Size General 	Undefined symbols (-U) 🕢 🕢 🖗 🎘 🖢

图 2-5 配置 Cross ARM GNU Assembler > Preprocessor 选项

配置 Cross ARM C Compiler > Includes 选项

配置 "Cross ARM C Compiler > Includes > Include paths"选项,该选项配置为当前工程 C 头文件引用路径,如图 2-6 所示。

图 2-6 配置 Cross ARM C Compiler > Includes 选项

🛞 Tool Settings 🚷 Toolchains 📕 Devices ,	🎤 Build Steps 🚇 Build Artifact 🗟 Binary Parsers 💽
 Target Processor Optimization Warnings Debugging Cross ARM GNU Assembler Preprocessor Includes Warnings Miscellaneous Cross ARM C Compiler Preprocessor 	Include paths (-I) Image: Constant of the system paths (-isystem) *\${workspace_loc:/\${ProjName}/PERIPHERAL/Includes}" *\${workspace_loc:/\${ProjName}/SYSTEM}" *\${workspace_loc:/\${ProjName}/USER}" *\${workspace_loc:/\${ProjName}/USER}" Include system paths (-isystem) Image: Constant of the system
 Includes Optimization Warnings Miscellaneous Cross ARM C Linker General Libraries Miscellaneous Cross ARM GNU Create Flash Image General General Cross ARM GNU Print Size General 	Include files (-include) 🕢 🕢 🖗

以软件开发工具包 GMD_RefDesign 参考设计为例,C 头文件路径配置, 如下所示。

- "\${workspace_loc:/\${ProjName}/CORE}"
- "\${workspace_loc:/\${ProjName}/PERIPHERAL/Includes}"
- "\${workspace_loc:/\${ProjName}/SYSTEM}"
- "\${workspace_loc:/\${ProjName}/USER}"

配置 Cross ARM C Linker 选项

配置 "Cross ARM C Linker > General > Script files"选项,该选项配置为 Flash 链接器 "gw1ns4c_flash.ld",如图 2-7 所示。

图 2-7 配置 Cross ARM C Linker 选项

🛞 Tool Settings 🛞 Toolchains 📕 Devices	🎤 Build Steps 🚇 Build Artifact	🗟 Binary Parsers 🚺
 Tool Settings Toolchains Devices Target Processor Optimization Warnings Debugging Cross ARM GNU Assembler Preprocessor Includes Warnings Miscellaneous Cross ARM C Compiler Preprocessor Includes Optimization Warnings Miscellaneous Cross ARM C Linker General Miscellaneous 	Build Steps Build Artifact Script files (-T) \${workspace_loc:/\${ProjName}/g	Binary Parsers (· · 원 원 양 상 상 w1ns4c_flash.ld)*
 Miscellaneous Cross ARM GNU Create Flash Image General Cross ARM GNU Print Size General 		

以软件开发工具包GMD_RefDesign参考设计为例,Flash链接器配置,如下所示。

"\${workspace_loc:/\${ProjName}/gw1ns4c_flash.ld}"

配置 Cross ARM GNU Create Flash Image 选项

配置 "Cross ARM GNU Create Flash Image > General > Output file format"选项,该选项配置为"Raw binary",如图 2-8 所示。

Target Processor	Output file format (O) Bau binant		
 Target Processor Optimization Warnings Debugging Cross ARM GNU Assembler Preprocessor Includes Warnings Miscellaneous Cross ARM C Compiler Preprocessor Includes Optimization Warnings Miscellaneous Cross ARM C Linker General Libraries Miscellaneous 	Output file format (-O) Raw binary Section: -j .text Section: -j .data Other sections (-j)	2 £ 2 7 I	✓
 ➢ General ➢ Cross ARM GNU Print Size 			

图 2-8 配置 Cross ARM GNU Create Flash Image 选项

配置 Devices 选项

配置"Devices > Device selection"选项,配置器件为"ARM Cortex M3 > ARMCM3",如图 2-9 所示。

图 2-9 配置 Devices 选项

Name		Details					
, and the second s	ARM Cortex M0	Eamily (12)	R LR RAM 25				
>	ARM Cortex M0 plus	Eamily (12)	8 kB RAM 25	5 kB ROM)			
>	ARM Cortex M1	Family (12)	8 kB RAM 25	5 kB ROM)			
>	ARM Cortex M23	Family (25)	6 kB RAM, 40	96 kB ROM)			
~	ARM Cortex M3	Family (12)	8 kB RAM, 25	6 kB ROM)			
	ARMCM3	Device (Co	ortex-M3, Rev	r2p1, 10 MHz)			
>	> ARM Cortex M33 Family (256 kB RAM, 4096 kB ROM)						
>	ARM Cortex M4	Family (12)	8 kB RAM, 25	6 kB ROM)			
evice cor 1emory r RMCM3	re: Cortex-M3 nap (Warning: Not yet	used to gener	rate the linker	scripts!)			
Section	Start	Size	Startup				
	0~2000000	0x00020000	0				
IRAM1	0x2000000						

2.2.3 编译

完成工程选项配置和代码编写后,编译当前工程,单击工具栏编译按钮 "≤》",或者右键单击当前工程,单击"Build Project"选项,编译产生 Gowin_EMPU(GW1NS-4C)软件编程设计二进制BIN文件,如图2-10所示。

图 2-10 编译



2.2.4 下载

使用下载软件 Programmer,下载软件编程设计二进制 BIN 文件。

单击 GOWIN MCU Designer 菜单栏 "Run > Programmer"或工具栏 Programmer "₩",打开下载软件 Programmer,如图 2-11 所示。

🔆 C/C++ - cm3_led/USER/main.c	- GOWIN MCU Designer			– 🗆 ×
File Edit Source Refactor Na	avigate Sear <u>ch Pr</u> oject Ru	n Window Help		
	🔪 💩 : 📫 : 🏥 : 🎄 🗸 🕌	Programmer	⇒ -	Ouick Access
		P 811		
🎦 Project Explorer 🛛 📃 🗖	🖸 main.c 🛛 🤎	Programmer		🗄 O 🛛 🐾 👘 🗖
📄 🔩 🗢	1	Debug History >	^	🖻 🕂 😿 🖋 🛛 🗰 🔻
cm3_free_rtos	5⊕ * Copyri 14	Debug As >	gy Co.,Ltd.	gw1ns4c.h
cm3_hyper_ram	15 /* Includes	Debug Configurations		++ delay_ms(volatile uint:
📋 cm3_i2c	16 #include "gw1n:	Toggla Brasknoist (trl+Chift+P		+ GPIOInit(void) : void
cm3_int_priority	1/ 18 /* Declaration	Tabala Lias Presidentiat		main(void) : int
cm3_keyscan	19 void delay_ms(Toggle Line Breakpoint		 GPIOInit(void) : void
✓ 25 cm3_led	20 void GPIOInit(loggle Method Breakpoint		delay_ms(volatile uint:
> 🖑 Binaries	21 22 /* Functions -	Toggle Watchpoint		
> 🗊 Includes	23⊖ int main(void) 🄌	Skip All Breakpoints Ctrl+Alt+B		
> 🗁 CORE	24 {	Remove All Breakpoints		
> 👝 Debug	25 Systeminit 26 GPIOInit()	Breakpoint Types >		
> 🗁 PERIPHERAL	27		,	
> 👝 STARTUP	28 while(1)			
> 👝 SYSTEM	30 GPTO ResetBit	(GPTO0.GPTO Pin 0): //LED1 on		
🗸 🗁 USER	31 delay_ms(500);		
> h gw1ns4c_conf.h	32			
> 🖻 gw1ns4c_it.c	34 delay ms(500,GPI0_PIN_0); //LEDI OTT		
> h gw1ns4c_it.h	35 }	/;		
> 🖻 main.c	36 }			
gw1ns4c_flash.ld	37 38 //Initializes GPT	3		
Cm3_mm	39 void GPIOInit(voi	4)	~	
Cm3_psram	<		>	< >
cm3_retarget	Roblems R Tasks	Concolo 💥 🔲 Proportion 🖓 Dobugger Concol		
Cm3_rtc	Tasks			
👕 cm3_spi		-	4 6 🔁 🔢	🔊 = 🖻 🖂 🖬 🖌 🖂 🔺
cm3_spi_nor_flash	CD1 Build Console [cm3_lec	ij		
Cm3_systick	Finished building: cm3	_led.bin		^
cm3_timer	Invoking: Cross ARM GN	J Print Size		
Cm3_uart	arm-none-eabi-sizefe	ormat=berkeley "cm3_led.elf"		
cm3_uart0_int	text data bs:	s dec hex filename		
📋 cm3_ucos_iii	Finished building: cm3	led.siz		
Cm3_wdog	a	-		
DigitalSeg	1			, v
~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~				,
🗠 cm3_led				

图 2-11 下载软件 Programmer 选项

单击下载软件 Programmer 菜单栏 "Edit > Configure Device" 或工具 栏 Configure Device "☞",打开 Device configuration。

如果选用器件 GW1NS-4C/GW1NSR-4C,下载选项配置,如图 2-12 所示。

- Access Mode 下拉列表,选择"MCU Mode"选项。
- Operation 下拉列表,选择"Firmware Erase, Program"选项或"Firmware Erase, Program, Verify"选项。

🙀 Device configuration		?	×
Device Operation			
Access Mode:	MCU Mode		-
Operation:	Firmware Erase, Program		-
Firmware Erase, Progra	Л		
- Programming Options			
File name: E:/desktop/	mcu_test/m3_4c/gowin_empu_QN48G.fs		
🔄 User Flash Initiali:	ation		
-FW/MCU/Binary Input Op	tions		
Firmware/Binary File: [E:/desktop/mou_test/m3_4c/led.bin		
	Save	Canc	el

图 2-12 器件 GW1NS-4C/GW1NSR-4C 下载选项配置

如果选用器件 GW1NSER-4C,下载选项配置,如图 2-12 所示。

- Access Mode 下拉列表,选择 "SecureFPGA Mode"选项。
- Operation 下拉列表, 选择"Firmware Erase, Program Securely"选项。

图 2-13 器件 GW1NSER-4C 下载选项配置

🙀 Device configuration	? ×
-Device Operation	
Access Mode:	SecureFPGA Mode 👻
Operation:	Firmware Erase, Program Securely 🔹
Firmware Erase, Progra Key Authentication Co	am,Verify with Security(One Time Programmable For de)
- Programming Options	
File name: C:/Users/De	esktop/secure_fpga/impl/pnr/secure_fpga.fs
User Flash Initiali	zation
-FW/MCU/Binary Input Op	tions
Firmware/Binary File:	ER_4C/Debug/bk_with_no_enroll_GWINSER_4C.bin
	Save Cancel

- "FW/MCU/Binary Input Options > Firmware/Binary File"选项,导入
 Gowin EMPU(GW1NS-4C)软件编程设计二进制 BIN 文件。
- 单击 "Save", 完成下载选项配置。

注!

"Programming Options > File name"选项,导入 Gowin_EMPU(GW1NS-4C)硬件设计码 流文件,请参考 IPUG932, Gowin_EMPU(GW1NS-4C)硬件设计参考手册。

完成 Device configuration 后,单击 Programmer 工具栏 "Program/Configure" (事),完成 Gowin_EMPU(GW1NS-4C)的下载。

2.2.5 在线调试

完成 Gowin_EMPU(GW1NS-4C)软件下载后,如果用户软件设计出现问题,可以连接开发板与 J-LINK 仿真器,在线调试当前 MCU 软件设计(在线调试的 MCU 软件设计必须与下载到芯片中的 MCU 软件设计保持一致)。

Gowin_EMPU(GW1NS-4C)软件调试流程,包括:

- 配置软件调试等级
- 配置软件调试选项
- MCU JTAG 模式切换
- MCU JTAG 接口切换
- 连接调试仿真器
- 启动软件调试

配置软件调试等级

Project Explorer 视图中,选择当前调试项目工程的"Properties > C/C++ Build > Settings > Tool Settings > Debugging > Debug level"选项,建议配 置调试等级为 Default(-g)或 Maximum(-g3),如图 2-14 所示。

🛞 Tool Settings 🛞 Toolchains 📕 Devices	🎤 Build Steps 🚇 Bui	d Artifact 🗟 Binary 🔹 🕨
🖄 Target Processor	Debug level	Maximum (-g3)
Optimization Warnings	Debug format	Toolchain default 🛛 🗸
Debugging	Generate prof inform	ation (-p)
✓	Generate gprof infor	mation (-pg)
🖄 Preprocessor	Other debugging flags	
🖄 Includes		
🖄 Warnings		
Miscellaneous		
V 🛞 Cross ARM C Compiler		
Preprocessor		
2 Includes		
Wornings		
Miscellaneous		
X S Cross ARM C Linker		
General		
Libraries		
Miscellaneous		
✓		
🖉 General		
✓		
🖄 General		
	1	

图 2-14 配置软件调试等级

配置软件调试选项

选择菜单栏 "Run > Debug Configurations > GDB SEGGER J-Link Debugging",右键选择 "New"选项,建立当前工程的软件调试配置选项, 如图 2-15 所示。

图 2-15 建立软件调试配置选项

🗱 Debug Configurations Create, manage, and run configurations 📑 🗈 🗶 🖻 🌩 🗕 Configure launch settings from this dialog: type filter text 📑 - Press the 'New' button to create a configuration of the selected type. C/C++ Application 📄 - Press the 'Duplicate' button to copy the selected configuration. C/C++ Attach to Application 🔀 - Press the 'Delete' button to remove the selected configuration. C/C++ Postmortem Debugger 🔆 - Press the 'Filter' button to configure filtering options. C/C++ Remote Application C GDB Hardware Debugging - Edit or view an existing configuration by selecting it. C GDB OpenOCD Debugging GDB PyOCD Debugging Configure launch perspective settings from the 'Perspectives' preference page. GDB OEMU Debuaaina C GDB SEGGER J New Launch Group 🗶 Delete Filter matched 10 of 61 items ? Debug Close

IPUG928-1.1

选择已建立的调试选项。

1. 选择"Main"选项,配置当前工程的输出映像文件等信息,如图 2-16 所示。

冬	2-16	配置	Main	选项
---	------	----	------	----

🐳 Debug Configurations	×
Create, manage, and run conf	igurations
Image: Second state st	Name: cm3_led Debug Main Is Debugger Startup Source Common SVD Path Project: cm3_led Browse C/C++ Application: Debug\cm3_led.elf Browse Debug\cm3_led.elf Yariables Search Project Browse Build (if required) before launching Build Configuration: Select Automatically ✓ O Enable auto build O Disable auto build O Disable auto build ●
< >> Filter matched 11 of 62 items	Re <u>v</u> ert Apply
?	<u>D</u> ebug Close

2. 选择 "Debugger" 选项, 配置 Debugger 器件名称和调试接口等信息, 如图 2-17 所示。

以软件开发工具包 GMD_RefDesign 参考设计为例, Debugger 选项配置, 如下所示。

- Device name: Cortex-M3
- Endianness: Little
- Connection: USB
- Interface: JTAG
- Connect to running target: Close
- Initial speed: Fixed 1000KHz
- GDB port: 2331
- SWO port: 2332
- Telnet: 2333
- Verify downloads: Open
- Initialize registers on start: Open

- Local host only: Open
- Other options: -singlerun –strict –timeout 0 –nogui
- Command: set mem inaccessible-by-default off

Gowin_EMPU(GW1NS-4C)仅支持 JTAG 调试接口,不支持 SWD 调试接口。

禁用 Connect to running target 选项。

图 2-17 配置 Debugger 选项

🗎 Mair 隊 Debugger 🔪 > Startup 🦃 Source 🔲 Common 🔀 SVD Path							
J-Link GDB Server	Setup					_	
Start the J-Link	GDB server	locally		Conne	ct to running ta	rget	
Executable path:	\${jlink_path)/\${jlink_gdbse	erver}			Browse	e Variables
Actual executable:	C:/Program	n Files (x86)/SE	GGER/JLink/.	ILinkGDBServe	rCL.exe		
	(to change i	it use the <u>glob</u> a	<u>al or worksp</u>	<u>ace</u> preference	es pages or the	project pro	perties page)
Device name:	Cortex-M3					Support	<u>ed device names</u>
Endianness:	Little	OBig					
Connection:	USB	OIP			(USB serial or	IP name/ad	dress)
Interface:	⊖ swd	JTAG					
Initial speed:	OAuto	OAdaptive	Fixed	1000 kHz			
GDB port:	2331				-		
SWO port:	2332]		🗹 Verif	fy downloads [🗸 Initialize r	egisters on start
Telnet port:	2333]		✓ Loca	I host only [Silent	
Log file:							Browse
Other options:	-singlerun	-strict -timeout	0 -nogui				
✓ Allocate console	e for the GD	B server		Allocate co	onsole for semil	hosting and	SWO
GDB Client Setup							
Executable name:	\${cross_pre	efix}gdb\${cross	_suffix}			Browse	e Variables
Actual executable:	arm-none-	eabi-gdb					
Other options:							
Commands:	set mem in	accessible-by-	default off				~
							\sim

MCU JTAG 模式切换

使用下载软件 Programmer,将 Gowin_EMPU(GW1NS-4C)的 MCU JTAG 模式,由下载模式切换到调试模式。

单击 GOWIN MCU Designer 菜单栏 "Run > Programmer" 或工具栏 Programmer "↓ ",打开 Programmer 下载软件,如图 2-11 所示。

单击 Programmer 菜单栏 "Edit > Configure Device"或工具栏 "Configure Device" (), 打开 Device configuration。

如果选用器件 GW1NS-4C/GW1NSR-4C, MCU JTAG 模式切换配置, 如图 2-18 所示。

- Access Mode 下拉列表,选择"MCU Mode"选项。
- Operation 下拉列表,选择 "Connect to JTAG of MCU"选项。

图 2-18 器件 GW1NS-4C/GW1NSR-4C MCU JTAG 模式切换配置

🙀 Device configuratio	on	?	\times
-Device Operation			
Access Mode:	MCV Mode		-
Operation:	Connect to JTAG of MCU		-
Connect to JTAG of	£ MCV.		
		Save Ca	ncel

如果选用器件 GW1NSER-4C, MCU JTAG 模式切换配置, 如图 2-19 所示。

- Access Mode 下拉列表,选择"SecureFPGA Mode"选项。
- Operation 下拉列表,选择 "Connect to JTAG of MCU"选项。

图 2-19 器件 GW1NSER-4C MCU JTAG 模式切换配置

👾 Device configurat	tion	?	×
-Device Operation-			
Access Mode:	SecureFPGA Mode		-
Operation:	Connect to JTAG of MCU		-
Connect to JTAG	of MCU.		
	Sav	e Can	cel

● 单击 "Save",完成 MCU JTAG 模式切换配置。

完成 Device configuration 后,单击 Programmer 工具栏 "Program/Configure" (事),完成 MCU JTAG 模式切换。

MCU JTAG 接口切换

以软件开发工具包 DK-START-GW1NSR4C-QN48G V1.1 开发板为例。

手动将开发板上 SW3、SW4、SW5、SW6 拨码开关(JTAG 接口 TCK、 TDO、TDI 和 TMS)由 "FT232"(Download)切换为 "J-LINK"(Debug)。

注!

• 在 MCU JTAG 模式切换和 MCU JTAG 接口切换过程中,必须保持开发板上电状态, 不可断电。 • 如果中途开发板断电,重新上电后自动还原为 MCU JTAG 下载模式。

启动软件调试

连接 J-LINK 仿真器。

单击工具栏 Debug "☆ ▼",进入调试状态,可以进行断点设置、单步 调试、复位和运行等操作,如图 2-20 所示。

图 2-20 启动软件调试

W Debug - cm3_led/USER/main.c - GOWIN MCU Design File Edit Source Refactor <u>N</u> avigate Search <u>P</u> roje	er ect <u>R</u> un <u>W</u> indow	/ Help					-		×
📑 🗝 🔚 🐘 🔪 🕪 🗉 🔳 🙌 🎝 🖓 🔐 H	• 🗟 🗷 🕹 🖣	↓ 日 参 ▼ (D 🕶 🍅 🔗	- 🥖 🐓	• 🚰 • 🐐	- 🔶 🗸	⇒ ▼		
					Qui	ck Acces	s 🛛 🖻	🖬 🐐	*
₩ Debug 🛛	🍇 i i i 🗸 🖓 🖓	(x)= Variab	🛛 💊 Brea	ak 388 Reg	ist 🕆 Pe	riph 1	\Lambda Modu	iles 🗖	
✓						K.	⇒ti ⊟	11 1	\bigtriangledown
✓ [™] cm3_led.elf		Name		Туре			Value		
✓									
main() at main.c.su 0x18a									
arm-none-eabi-gdb									
📕 Semihosting and SWV		<							>
									$\widehat{}$
		<							>
i main.c ⊠				- 8	🗄 🗄 Outli	ne 🛙			
23⊖ int main(void)				^		E 🖡	🛛 🔍 🖋	● 牂	\bigtriangledown
24 1 25 SystemInit(); //Initializes system					9 B	gw1ns4	c.h		
26 GPIOInit(); //Initializes GPIO					++	delay_n	ns(volatil	e uint32_	t) : vc
28 while(1)						main(vo	oid) : int	void	
29 { 230 GPIO_ResetBit(GPIO0,GPIO_Pin_0); //I	LED1 on				•	GPIOIni	t(void) :	void	
31 delay_ms(500);					•	delay_n	ns(volatil	e uint32_	t) : vc
<pre>33 GPI0_SetBit(GPI00,GPI0_Pin_0); //</pre>	LED1 off								
34 delay_ms(500); 35 }									
36 }				×					
	-			- M D				• -	_
E Console 🛛 🖉 Tasks 🦹 Problems 🚺 Executables	Memory			K 🔆 🖹 k 🛓		2	₽ - (<u>≯</u> ▼ "	
cm3_led Debug [GDB SEGGER J-Link Debugging] JLinkGD Reading all registers	BServerCL.exe								^
Performing single step									- 1
Reading all registers									
Removing breakpoint @ address 0x0000018A, Size = Read 4 bytes @ address 0x0000018A (Data = 0x4620	= 2 02101)								
<									>
	Writable	Smart Insert	30 : 1						6

2.3 参考设计

Gowin_EMPU(GW1NS-4C)支持 GOWIN MCU Designer (V1.1 及以上 版本)软件环境的参考设计,通过此链接获取如下参考设计: cdn.gowinsemi.com.cn/Gowin EMPU V1.1.zip

Gowin_EMPU\ref_design\MCU_RefDesign\GMD_RefDesign

