

Gowin_EMPU_M1 IDE **软件参考手册**

IPUG536-1.3, 2019-09-27

版权所有[©]2019 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
2019/02/18	1.0	初始版本。
2019/07/18	1.1	 MCU 硬件设计与软件编程设计支持扩展外部设备 CAN、 Ethernet、SPI-Flash、RTC、TRNG、DualTimer、I2C、 SPI、SD-Card;
		● MCU 支持片外 SPI-Flash 下载启动方式。
2019/08/18	1.2	 MCU硬件设计与软件编程设计支持扩展外部设备DDR3 Memory; 修复已知 ITCM、DTCM Size 和 IDE 问题。
2019/09/27	1.3	更新优化 MCU 编程软件 GOWIN MCU Designer 界面和功能。

目录

目录i
图目录ii
1 ARM Keil 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1
1.2.3 编译
1.2.4 软件调试9
1.3 参考设计11
2 GOWIN MCU Designer12
2.1 软件安装与配置12
2.2 工程模板12
2.2.1 工程创建12
2.2.2 工程配置15
2.2.3 编译19
2.2.4 调试19
2.3 参考设计21

i

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 JTAG 调试接口类型	6
图 1-7 配置 SW 调试接口类型	6
图 1-8 配置 Flash 选项	7
图 1-9 配置调试初始化文件	8
图 1-10 工程编译	9
图 1-11 仿真器连接	10
图 1-12 启动调试	11
图 2-1 新建工程	13
图 2-2 选择平台配置类型	14
图 2-3 选择编译工具链和路径	14
图 2-4 配置 Target Processor	15
图 2-5 配置 Cross ARM GNU Assembler Preprocessor	16
图 2-6 配置 Cross ARM C Compiler Includes	16
图 2-7 配置 Cross ARM C Linker	17
图 2-8 配置 Cross ARM GNU Create Flash Image	18
图 2-9 配置 Devices	18
图 2-10 建立调试配置选项	19
图 2-11 配置 Image 路径	20
图 2-12 配置 Debugger	20
图 2-13 启动调试	21

1 ARM Keil 软件

1.1 软件安装

请参考 ARM Keil MDK 官网提供的 Getting Started with MDK.

1.2 工程模板

使用 ARM Keil MDK 进行 Gowin_EMPU_M1 软件编程设计,需要创建工程、配置选项、编写代码、编译和调试。

1.2.1 创建工程

打开 ARM Keil MDK,选择菜单栏 Project 中 New uVision Project..., 创建工程,如图 1-1 所示。

图 1-1 创建工程

File Edit View	Proj	iect Flash	n Debu	g P	eripherals	Tools	SVCS	Window	Help	
🗋 🖆 🖼 🗿 🗌		New µVis	ion Proje	.t						
I I I I I I I I I I I I I I I I I I I		New Mult	i-Project	Nork	space					
Project		Open Pro	ject ject							
		Export Manage))
		Select Dev Remove It	vice for Ta	rget .						
	Ň	Options							Alt	+F7
		Clean Tar	gets							
		Build Targ	jet II target f	iles						F7
		Batch Bui	Id							
	8	Translate. Stop buil	 d						Ctrl	+F7

1.2.2 配置

配置器件

Gowin_EMPU_M1 内置 ARM Cortex-M1 内核,所以器件选择 ARM Cortex M1 的"ARMCM1",如图 1-2 所示。

👿 Options for Target 'led'				X
Device Target Output Li Software Packs Vendor: ARM Device: ARMCM1 Toolset: ARM Search:	sting User	C/C++ Asm	Linker Debug tware Pack Pack: ARM.CMSIS.5.4 URL: <u>http://www.ke</u>	Utilities
ARM ARM Cortex M0 ARM Cortex M0 ARM Cortex M1 ARM Cortex M1 ARM Cortex M2 ARM Cortex M3 ARM Cortex M3 ARM Cortex M3 ARM Cortex M4	plus 3 3 3 (MPS3)	The ARM Cort embedded ap an FPGA. The ARM Cort architecture pr	ex-M1 FPGA processor is lications that require a sm ex-M1 processor implemen ofile.	intended for deeply nall processor integrated into Ints the ARMv6-M
[OK	Cancel	Defaults	Help

图 1-2 配置器件

配置 ROM 和 RAM

Gowin_EMPU_M1 的 ITCM 作为 ROM。

Gowin_EMPU_M1的 DTCM 作为 RAM。

配置 ROM(ITCM)和 RAM(DTCM)的起始地址和容量大小。

如果选择片外 SPI-Flash 下载启动方式,则 Gowin_EMPU_M1 的 ITCM 的起始地址为 0x400,容量根据硬件实际配置设定,软件开发工具包的参考 设计中配置为 0x7C00。

如果选择片内 ITCM 初始值下载启动方式,则 Gowin_EMPU_M1 的 ITCM 起始地址为 0x0000000,容量可以配置为 1KB 或 2KB 或 4KB 或 8KB 或 16KB 或 32KB 或 64KB 或 128KB 或 256KB,参考设计默认 32KB。请参 考 IPUG531, Gowin_EMPU_M1 硬件设计参考手册 ITCM Size 设定的值。

Gowin_EMPU_M1 的 DTCM 起始地址为 0x20000000,容量可以配置 为1KB或2KB或4KB或8KB或16KB或32KB或64KB或128KB或256KB, 参考设计默认 32KB。请参考 IPUG531, Gowin_EMPU_M1 硬件设计参考手 册中 DTCM Size 设定的值。

受片内存储资源限制,ITCM 和 DTCM 的容量配置不能超过片内最大存储容量。

GW1N-9系列,ITCM或DTCM最大可配置为32KB,如果ITCM或DTCM 某个存储器已配置为32KB,则另一个存储器最大只能配置为16KB; GW1NR-9 系列,ITCM 或 DTCM 最大可配置为 32KB,如果 ITCM 或 DTCM 某个存储器已配置为 32KB,则另一个存储器最大只能配置为 16KB;

GW2A-18 系列, ITCM 或 DTCM 最大可配置为 64KB, 如果 ITCM 或 DTCM 某个存储器已配置为 64KB,则另一个存储器最大只能配置为 16KB;

GW2AR-18 系列, ITCM 或 DTCM 最大可配置为 64KB, 如果 ITCM 或 DTCM 某个存储器已配置为 64KB,则另一个存储器最大只能配置为 16KB;

GW2A-55 系列, ITCM 或 DTCM 最大可配置为 256KB, 如果 ITCM 或 DTCM 某个存储器已配置为 256KB,则另一个存储器最大只能配置为 16KB。

ROM 和 **RAM** 的配置如图 **1-3** 所示,以 **DK-START-GW2A18** V2.0 开发 板参考设计为例, **ROM** 和 **RAM** 分别配置为 **32KB**。

图 1-3 配置 ROM 和 RAM

Options for Target 'led'	<u> </u>
Device Target Output Listing Vser C/C++ A	sm Linker Debug Utilities
ARM ARMCM1	
Xtal (MHz): 12.0	ARM Compiler:
Operating system: None	
System Viewer File:	Use Cross-Module Optimization
	🗌 Use MicroLIB 📃 Big Endian
Use Custom File	
Read/Only Memory Areas	Read/Write Memory Areas
default off-chip Start Size Startup	default off-chip Start Size NoInit
□ ROM1: □ C	🗖 RAM1:
□ ROM2: ○	□ RAM2: □
□ ROM3: □ C	□ RAM3: □ □
on-chip	on-chip
IROM1: 0x0 0x8000 €	IRAM1: 0x20000000 0x8000 □
□ IROM2: O	IRAM2:
, ,	,
OK Can	cel Defaults Help

配置输出文件格式

- Gowin_EMPU_M1 软件编程设计输出文件格式使用 BIN 格式,所以需要将 axf 文件格式转换为 BIN 文件格式
- 如果使用 BIN 文件作为 ITCM 的初始值,需要使用 make_hex.exe 工具 将 BIN 文件转换为四个十六进制文件 itcm0、itcm1、itcm2 和 itcm3
- User 命令行选项中文件格式转换工具调用方法如图 1-4 所示
 - Run #1
 - fromelf.exe --bin -o bin-file axf-file
 - Run #2
 - make_hex.exe bin-file

				5
Options for Target 'led'				
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				
		2	Not Specified	
		2	Not Specified	
After Build/Rebuild				
🔽 🔽 Run #1	C:\Keil_v5\ARM\ARMCC\bin\fromelf.exebin -o	led.	bin .\Objects\le	d.axf
✓ Run #2	make_hex.exe led.bin	2	Not Specified	\Box
Run 'After-Build' Conditionally				
Eeep When Complete	Start Debugging			
	OK Cancel Defaults		Не	lp

图 1-4 配置输出文件格式

配置头文件路径

配置头文件路径,编译过程中用来调用头文件,配置如图 1-5 所示。

Device Target Output Listing User	C/C++ Asm Linker Debug	Utilities
Preprocessor Symbols		·
Define:		
Undefine:		
Language / Code Generation		
	Strict ANSI C	Warnings:
Optimization: Level 0 (-00)	Enum Container always int	All Warnings 🔍
Coptimize for Time	Plain Char is Signed	🗖 Thumb Mode
Split Load and Store Multiple	Read-Only Position Independent	No Auto Includes
✓ One ELF Section per Function	Read-Write Position Independent	C99 Mode
Include\CORE;\PERIPHERAL\inc;	\STARTUP;\SYSTEM;\USER	
Controls		
Compiler control string	00apcs≕interworksplit_sections -I/C0 I/USER	RE -I/PERIPHERAL/inc
		1

图 1-5 配置头文件路径

配置调试选项

- 配置仿真器类型
 - U-LINK 仿真器 如果选择使用 U-LINK 仿真器,则 Debug 选项配置为"ULNK2/ME Cortex Debugger"。
 - J-LINK 仿真器 如果选择使用 J-LINK 仿真器,则 Debug 选项配置 为"J-LINK/J-TRACE Cortex。

• 配置调试接口类型

调试接口类型配置可以配置为 JTAG 或 Serial Wire。

JTAG 接口
 如果配置为 JTAG 接口,如图 1-6 所示。

Cortex-M Target Driver Setup	a 100 m 100 Mag 011-1	X
Debug Trace Flash Download		
ULINK USB - JTAG/SW Adapter	TAG Device Chain	
Serial No: Any	IDCODE Device Name IF	Rien Move
ULINK Version: ULINK2	TDO Ox4BA00477 ARM CoreSight JTAG-DP	4 Up
Device Family: Cortex-M	ты	Down
Firmware Version: V2.03	Automatic Detection ID CODE:	
SWJ Port: JTAG -	C Manual Configuration Device Name:	
Max Clock: 1MHz 💌	Add Delete Update IR len:	AP: 0x00
Debug Connect & Reset Options	Cache Options Download	d Options
Connect: Normal 🗨 Reset:	utodetect 💌 🗹 Cache Code 🗌 Verify	Code Download
Reset after Connect	IV Cache Memory Down	load to Hash
Log Debug Accesses		
[OK Cancel	Help

图 1-6 配置 JTAG 调试接口类型

- SW 调试接口

如果配置为 SW 调试接口,如图 1-7 所示。

图 1-7 配置 SW	调试接口类型
-------------	--------

Cortex-M Target Driver Setup	that [1000] and [Long Manage [Millions]]	×
Debug Trace Flash Download		
ULINK USB - JTAG/SW Adapter Serial No: Any	SW Device	Move Up
Device Family: Cortex-M Firmware Version: V2.03	Automatic Detection ID CODE: Manual Configuration Device Name:	Down
Max Clock: 1MHz	Add Delete Update AP	: 0x00
Debug Connect & Reset Options Connect: Normal ✓ Reset after Connect ✓ Log Debug Accesses	t: Autodetect Cache Options Code Code Code Code Code Code Code Code	wnload Iash
	OK Cancel	Help

配置 Flash 选项

如果需要在线调试,请不要选择"Update Target before Debugging"选项, 如图 1-8 所示。

		x
options for far		
Device Target	Output Listing User C/C++ Asm Linker Debug Utilities	
Configure Flash I	Menu Command	٦
Use Target I	Driver for Flash Programming	
	- Use Debug Driver Settings Update Target before Debugging	
Init File:	Edit	
O Use Externa	al Tool for Flash Programming	
Command:		
Arguments:		
Γ	Run Independent	
Configure Image	File Processing (FCARM):	
Output File:	Add Output File to Group:	
	STARTUP	
Image Files Roo	ot Folder: 🗌 🗖 Generate Listing	
	OK Cancel Defaults Help	

图 1-8 配置 Flash 选项

配置调试初始化文件

如果选择片外 SPI-Flash 下载方式,在线调试时需要加载调试初始化文件, "Initialization File"选项中选择加载初始化文件 ext_debug.ini,如图 1-9 所示。

🕼 Options for Target 'led'				
Device Target Output Listing User C/C++ Asm Linker Debug Jtilities				
O Use Simulator with restrictions Settings				
✓ Load Application at Startup ✓ Run to main()	☑ Load Application at Startup ☑ Run to main()			
Initialization File:	Initialization File:			
Edit	.\ext_debug.ini Edit			
Restore Debug Session Settings	Restore Debug Session Settings			
I Breakpoints I Toolbox I Breakpoints I Toolbox				
Vatch Windows & Performance Analyzer	Watch Windows			
I Memory Display I System Viewer I Memory Display I System Viewer				
CPU DLL: Parameter:	Driver DLL: Parameter:			
SARMCM3.DLL SARMCM3.DLL				
Dialog DLL: Parameter: Dialog DLL: Parameter:				
DARMCM1.DLL PCM1	TARMCM1.DLL pCM1			
Manage Component Viewer Description Files				
OK Cancel Defaults Help				

图 1-9 配置调试初始化文件

1.2.3 编译

完成代码编写和工程配置后,编译输出 BIN 格式文件和四个十六进制格式文件 itcm0、itcm1、itcm2 和 itcm3,如图 1-10 所示。



图 1-10 工程编译

1.2.4 软件调试

完成硬件设计输出和软件编程设计输出的下载后,如果用户软件设计出现问题,可以连接 U-LINK 或 J-LINK 仿真器在线调试。用户可以下载、调试软件,无需重新编译硬件设计。

1. 连接仿真器

按照硬件设计中约束到 FPGA IO 的 Debug Access Port 的位置,连接 J-LINK 或 U-LINK。

以参考设计和 DK-START-GW2A18 V2.0 开发板为例,连接方式如图 1-11 所示。



图 1-11 仿真器连接

2. 启动调试

连接 U-LINK 或 J-LINK 仿真器,选择工具栏 Debug 按钮,开始调试, 如图 1-12 所示。

File Edit View Project Elash	Debug Peripherals Tools SVCS Window Help
	7 (~ ← → ♥ № № № ≇ # <i> </i>
0 9 9 9 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
Registers 🛛 🖡 📔	
Register Value	0x000004A2 0000 MOVS r0,r0
- Core	23: SystemInit();
b0 0-0-01170740 H1 0-20000270 H2 0-22000270 H4 0-0000000 H5 0-20000270 H4 0-0000000 H5 0-20000270 H6 0-20000270 H5 0-20000270 H6 0-20000270 H7 0-20000270 H8 0-20000270 H8 0-20000270 H1 0-20000500 H1 0-20000570 H14 0-20000570 H15 0-20000570 H14 0-20000570 H15 0-20000570 H16 0-20000570 H17 0-20000570 H18 0-20000570 H19 0-20000570 H19 0-20000570 H19 0-20000570 H9 Thread M0-0 Thread M0-0 Thread M0-0 Thread State 1050010500 Sec	<pre>24: 0x0000044 F7FFFEDC BL.W SystemInit (0x00000460) 25: Lcd Pin_Config(): 26: Voc00048 F7FFFES2 BL.W Lcd Pin_Config (0x0000350) * * * * * * * * * *</pre>
	<pre>28 Lod Write_String(" * Gowin GWINS-2C * ",0); 29 Lod_Write_String(" Hello World!",1); 30 while(1); 32 }</pre>
	33 L
🖻 Project 🛛 🧱 Registers	€ [
Command	a 🖪 Call Stack + Locals
Load "D:\\user-bak\\Users	\\root\\Desktop\\type_c\\1ns_2c\\Examples\\KEIL\\LCD-Display\\1cd\\PROJECT\\0 ^ Name Location/ Type
	● m 0x00000000 int f0

图 1-12 启动调试

1.3 参考设计

Gowin_EMPU_M1 支持 ARM Keil MDK 软件环境的参考设计: Gowin_EMPU_M1\ref_design\MCU_RefDesign\Keil_RefDesign

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

2.1 软件安装与配置

GOWIN MCU Designer 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

2.2.1 工程创建

创建工程

选择菜单栏 File 中的 New 下的 C Project,如图 2-1 所示。

- 1. 建立项目名称
- 2. 选择项目类型 Empty Project
- 3. 选择工具链 ARM Cross GCC

图 2-1 新建工程

🐝 C Project	
C Project Create C project of selected type	
Project name: led_cm1	
✓ Use <u>d</u> efault location	
Location: D:\GOWIN_MCU_Designer\wor	kspace\led_cm1 Browse
Choose file s <u>y</u> stem: default 💌]
Project type:	Toolchains:
🔺 🚘 Executable	ARM Cross GCC
Empty Project	MinGW GCC
Hello World ARM C Project	RISC-V Cross GCC
Hello World RISC-V C Project	
Hello World ANSI C Project	
Shared Library	
Static Library	
Makefile project	
۰ III ا	
Show project types and toolchains only	y if they are supported on the platform

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 选择平台配置类型

🖗 C Project	
Select Configurations	
Select platforms and configurations you wish to deploy on	
Project type: Executable Toolchains: ARM Cross GCC Configurations:	
 ✓ Some Debug ✓ Some Release 	Select all Deselect all
	Advanced settings

选择编译工具链和路径

选择交叉编译工具链 arm-none-eabi-gcc 和其所在路径,如图 2-3 所示。

图 2-3 选择编译工具链和路径

🙀 C Project		- • •
GNU ARM Cross	Toolchain ain and configure path	
Toolchain name:	GNU MCU Eclipse ARM Embedded GCC (arm-none-eabi-gcc)	•
Toolchain path:	D:\GOWIN_MCU_Designer\toolchain\ARM_toolchain\bin	Browse

导入软件设计

完成新建工程后,选择工作空间 workspace 下新建的项目工程,导入软件编程设计。

以 GNU_RefDesign 参考设计为例,软件编程设计目录及代码如下所示。

- CORE: ARM Cortex-M1 内核定义
- PERIPHERAL: 外设驱动库
- **STARTUP**: 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: 用户设计
- GOWIN_M1_flash.ld: GNU 工具链的 Flash 链接脚本

2.2.2 工程配置

GOWIN MCU Designer 中,选择当前工程,右键选择 Properties,选择 选项卡 C/C++ Build,选择 Setting 选项,配置工程参数选项。

配置 Target Processor

选择 Target Processor, 配置 ARM family 为 cortex-m1, 如图 2-4 所示。

图 2-4 配置 Target Processor

S Tool Settings S Toolchains Devices	🎤 Build Steps 🚇	Build Artifact 🗟 Binary Pa 🛨	
🖄 Target Processor	ARM family	cortex-m1	
🖉 Optimization 🎘 Warnings	Architecture	Toolchain default	
🖉 Debugging	Instruction set	Thumb (-mthumb)	
Cross ARM GNU Assembler	Thumb interwork (-mthumb-interwork)		
🐸 Preprocessor 灅 Includes	Endianness	Toolchain default	
🖄 Warnings	Float ABI	Toolchain default	
Miscellaneous ARM C Compiler	FPU Type	Toolchain default	
🖄 Preprocessor	Unaligned access	Toolchain default	
🖄 Includes 🎽 Optimization	AArch64 family	Generic (-mcpu=generic)	
🖄 Warnings	Feature crc	Toolchain default	
Miscellaneous Scross ARM C Linker	Feature crypto	Toolchain default	
🖄 General	Feature fp	Toolchain default	
🖄 Libraries 🎽 Miscellaneous	Feature simd	Enabled (+simd)	
a 🛞 Cross ARM GNU Create Flash Image	Code model	Small (-mcmodel=small)	
🖄 General	Strict align (-mstrict-align)		
Cross ARM GNU Print Size General	Other target flags		

配置 Cross ARM GNU Assembler Preprocessor

选择 Cross ARM GNU Assembler 下的 Preprocessor 选项, 配置汇编宏 定义___STARTUP_CLEAR_BSS, 如图 2-5 所示。

Tool Settings	🛞 Toolchains	Devices	🎤 Build Steps	🚇 Build Artifact	🗟 Binary Pa 🕚
p Target Pi	rocessor		Use preproce	essor	
🖄 Optimiza	tion		Do not searc	h system directorie	es (-nostdinc)
🖄 Warnings	5		Preprocess o	nly (-E)	
🖄 Debuggi	ng		Defined symbol	ols (-D)	े 🔊 🧟 है। स
🔺 🛞 Cross AR	M GNU Assemb	er			
🖄 Prepr	ocessor		STARTUP_CLE	AR_BSS	
🖄 Incluc	les				
🖄 Warn	ings				
🖄 Misce	llaneous				
4 🛞 Cross AR	M C Compiler				
🖄 Prepr	ocessor				
🖄 Includ	les				
🖄 Optin	nization				
🖄 Warn	ings				
└ 配置(Cross ARM C	Compiler	Includes		I

选择 Cross ARM C Compiler 下的 Includes 选项,配置 C 头文件路径, 如图 2-6 所示。

以 GNU_RefDesign 参考设计为例,头文件路径配置如下所示。

"\${workspace_loc:/\${ProjName}/CORE}"

"\${workspace_loc:/\${ProjName}/PERIPHERAL/inc}"

"\${workspace_loc:/\${ProjName}/SYSTEM}"

"\${workspace_loc:/\${ProjName}/USER}"

图 2-6 配置 Cross ARM C Compiler Includes

🛞 Tool Settings 🛞 Toolchains 📕 Devices	🎤 Build Steps 🙅 Build Artifact 🗟 Binary Pa 🔹 🔸
 Target Processor Optimization Warnings Debugging Cross ARM GNU Assembler 	Include paths (-I) * * * * * * * * * * * * * * * * * * *
 Preprocessor Includes Warnings Miscellaneous Cross ARM C Compiler 	
Includes Optimization Werninge	Include system paths (-isystem) 🛛 👻 🛞

图 2-5 配置 Cross ARM GNU Assembler Preprocessor

配置 Cross ARM C Linker

选择 Cross ARM C Linker 下的 General 选项, 配置 Flash 链接脚本 GOWIN_M1_flash.ld, 如图 2-7 所示。

以 GNU_RefDesign 参考设计为例, Flash 链接脚本如下所示。

"\${workspace_loc:/\${ProjName}/GOWIN_M1_flash.ld}"

如果 Gowin_EMPU_M1 软件编程设计使用片外 SPI-Flash 下载启动方 式,请在 GOWIN_M1_flash.ld 中修改 Flash 起始地址"FLASH ORIGIN"为 0x00000400。



图 2-7 配置 Cross ARM C Linker

配置 Cross ARM GNU Create Flash Image

选择 Cross ARM GNU Create Flash Image 中的 General 选项,配置 Gowin_EMPU_M1 输出文件格式为 RAW binary,如图 2-8 所示。

🛞 Tool Settings 🛞 Toolchains 🔳 Devices	🎤 Build Steps 🚇 Build Artifact 🗟 Binary Pa 💶
Target Processor	Output file format (-O) Raw binary
Warrings	Section: -j .text
Warnings Debugging	🔲 Section: -j .data
Cross ARM GNU Assembler	Other sections (-j) 🗿 👔 🙀
Preprocessor	
Includes	
🖉 Warnings	
Miscellaneous	
Cross ARM C Compiler	
🖄 Preprocessor	
🖄 Includes	
🖄 Optimization	
🖄 Warnings	
🖄 Miscellaneous	
Cross ARM C Linker	
🖄 General	
🖉 Libraries	
Cross AKM GNU Create Flash Image	
	J
	11

图 2-8 配置 Cross ARM GNU Create Flash Image

配置 Devices

选择 Devices 选项卡,选择器件 ARM Cortex M1 "ARMCM1",如图 2-9 所示。

图 2-9 配置 Devices

Tool Sett	ings 🛞 Toolchains	📕 Devices 🎤 Build Steps 😤 Build Artifact	🗟 Binary Pa
Device sel	ection (Used by debug.	Not yet used during build!)	
Name		Details	
⊳	ARM Cortex A5	Family (2048 kB RAM, 2048 kB ROM)	
⊳	ARM Cortex A7	Family (2048 kB RAM, 2048 kB ROM)	
⊳	ARM Cortex A9	Family (2048 kB RAM, 2048 kB ROM)	:
⊳	ARM Cortex M0	Family (128 kB RAM, 256 kB ROM)	
\triangleright	ARM Cortex M0 plus	Family (128 kB RAM, 256 kB ROM)	
4	ARM Cortex M1	Family (128 kB RAM, 256 kB ROM)	
	ARMCM1	Device (Cortex-M1, Rev r1p0, 10 MHz)	
	ARM Cortex M23	Family (256 kB RAM, 4096 kB ROM)	

2.2.3 编译

完成工程配置和代码编写后,编译工程,选择工具栏编译按钮,开始编译。

2.2.4 调试

配置调试选项

1. 选择菜单栏 Run 中的 Debug Configurations 下的 GDB SEGGER J-Link Debugging 选项卡, 右键选择 New, 建立工程的调试配置选项, 如图 2-10 所示。

图 2-10 建立调试配置选项

🐝 Debug Configurations

Create, manage, and run configurations

type filter text
C/C++ Application
C/C++ Attach to Application
C/C++ Postmortem Debugger
C/C++ Remote Application
C GDB Hardware Debugging
C GDB OpenOCD Debugging
C GDB PyOCD Debugging
C GDB QEMU Debugging
Launch Group
Duplicate
💥 Delete

2. 选择 Main 选项卡, 配置输出文件路径, 如图 2-11 所示。

图 2-11 配置 Image 路径

₩ Debug Configurations	
Create, manage, and run configurations	
😣 [Debugger]: Device name?	
Image: Second Secon	Name: cm1_led Debug Main Debugger Startup Source Project: cm1_led C/C++ Application: Debug\cm1_led.elf Build (if required) before launching Build Configuration: Select Automatically © Enable auto build @ Use workspace settings

- 3. 选择 Debugger 选项卡, 配置 Debugger 器件名称为 Cortex-M1, 调试接口为 JTAG 或 Serial Wire, 如图 2-12 所示。
 - Device Name, Cortex-M1
 - Interface, JTAG 或 SWD
 - Endianness, Little
 - Connection, USB

图 2-12 配置 Debugger

J-Link GDB Server S	Setup						
Start the J-Link	GDB server	locally		Co	onnect to runnir	ng target	
Executable path:	\${jlink_pat	k_path}/\${jlink_gdbserver}					Variables
Actual executable:	C:/Program Files (x86)/SEGGER/JLink_V632i/JLinkGDBServerCL.exe						
	(to change	it use the glob	al or <u>works</u>	pace pref	ferences pages	or the <u>project</u> pro	perties page
Device name:	Cortex-M1			Supported device names			
Endianness:	Little	🔘 Big					
Connection:	OSB	O IP			(USB se	rial or IP name/ad	dress)
Interface:	⊘ SWD	JTAG					
Initial speed:	O Auto	O Adaptive	Fixed	1000	kHz		
GDB port:	2331						
SWO port:	2332	1		V	erify download:	s 🔽 Initialize regi	sters on start
Telnet port:	2333			V Lo	ocal host only	🔲 Silent	
Log file:					-		Browse
Other options:	-singlerun	-strict -timeou	t 0 -nogui				

调试

连接 J-LINK 仿真器与开发板,选择工具栏 Debug 按钮,开始调试,如



Gowin_EMPU_M1 支持 GOWIN MCU Designer 软件环境的参考设计: Gowin_MCU_M1\ref_design\MCU_RefDesign\GNU_RefDesign

