

GW1NS-2C MCU IDE

软件参考手册

IPUG519-1.2.1,2019-08-06

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
2018/08/21	1.0	初始版本。
2018/11/21	1 1	● 优化 Eclipse 安装过程
2010/11/21	1.1	● 增加工程实例操作
2010/04/12	1.0	● 更新软件编程库
2019/04/12	1.2	● 更新软件编程参考设计
2019/08/06	1.2.1	修复已知 SPI 和 ADC 问题。

目录

目录i
图目录ii
1 ARM Keil MDK 软件1
1.1 软件安装1
1.2 工程模板1
1.2.1 配置选项1
1.2.2 工程配置2
1.2.3 工程编译6
1.2.4 工程下载7
1.2.5 工程调试7
1.3 参考设计9
2 GOWIN MCU Designer10
2.1 软件安装10
2.2 工程模板10
2.2.1 工程创建10
2.2.2 工程配置14
2.2.3 工程编译21
2.2.3 工程编译
2.2.3 工程编译

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	3
图 1-5 配置宏定义和头文件路径	4
图 1-6 配置下载选项	4
图 1-7 配置调试选项	5
图 1-8 调试接口类型	5
图 1-9 工程编译	6
图 1-10 MCU 下载	7
图 1-11 JTAG 模式切换命令	8
图 1-12 启动调试	9
图 2-1 图新建工程	11
图 2-2 选择平台类型配置	12
图 2-3 选择工具链和路径	13
图 2-4 工程结构	13
图 2-5 选择 Properties	14
图 2-6 配置 Cross ARM GNU Assembler Preprocessor	15
图 2-7 配置 Cross ARM GNU Assembler Includes	16
图 2-8 配置 Cross ARM C Compiler Preprocessor	17
图 2-9 配置 Cross ARM C Compiler Includes	18
图 2-10 配置 Cross ARM C Linker	19
图 2-11 配置 Cross ARM GNU Create Flash Image	20
图 2-12 配置 Devices	21
图 2-13 编译工程	21
图 2-14 工程下载	22
图 2-15 建立调试配置选项	23
图 2-16 配置 Image 路径	23

图 2-17 配置 Debugger	24
图 2-18 JTAG 接口切换	25
图 2-19 启动调试	25

1 ARM Keil MDK 软件

1.1 软件安装

请参考 ARM Keil MDK 官网提供的<u>《MDK Getting Started》</u>和<u>《uVision</u> <u>User's Guide》</u>。

1.2 工程模板

使用 ARM Keil MDK 软件进行 Gowin_EMPU 软件编程设计,需要创建工程、配置工程选项、编码、编译、下载和调试。

1.2.1 配置选项

工程创建

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

File Edit View	Proj	ject Flash Debug Peripherals Tools SVCS Window Help	
i 🗋 💕 🖌 💋		New µVision Project	
8 X X 8		New Multi-Project Workspace	
Project		Close Project	
		Export Manage)
		Select Device for Target Remove Item	
	Ň	Options	Alt+F7
		Clean Targets	
		Build Target	F7
		Rebuild all target files	
	۲	Batch Build	
	٢	Translate	Ctrl+F7
		Stop build	

图 1-1 创建工程

1.2.2 工程配置

配置器件

Gowin_EMPU for GW1NS-2C 内置 ARM Cortex-M3 内核,所以器件选择 ARM Cortex-M3 的"ARMCM3",如图 1-2 所示。

图 1-2 配置器件

Device Target Output Listing User	C/C++ Asm Linker Debug Utilities
Vendor: ARM	Pack
Toolset: ARM	URL: <u>http://www.keil.com/pack/</u>
Search:	
ARM ARM Cortex M0 ARM Cortex M0 plus ARM Cortex M23 ARM Cortex M3 ARM CONSK_CM3 CMSDK_CM3 CMSDK_CM3 ARM Cortex M33 ARM Cortex M33 ARM Cortex M33 ARM Cortex M33 (MPS3)	The Cortex-M3 processor is an entry-level 32-bit ARM 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.
OK	Cancel Defaults Help

配置 ROM 和 RAM

配置 ROM 和 RAM 的起始地址和容量大小。

GW1NS-2C ROM 起始地址为 0x00000000, 容量为 128K Byte。

GW1NS-2C RAM 起始地址为 0x2000000,容量为 2KB、4KB 或 8KB, 如图 1-3 所示。

Xtal (MHz):	12.0	Code C	Generation Compiler:	Use defaul	t compiler version	on 5 💌
Operating system: None	-					
System Viewer File:			se Cross-M	odule Optimiza	ation	
DS_CM3.svd			se MicroLII	B	🗖 Big Endian	
Use Custom File						
- Read/Only Memory Areas		-Read/	Write Mem	ory Areas		
default off-chip Start Size	Startup	default	off-chip	Start	Size	NoInit
ROM1:	- C		RAM1:			
ROM2:	- c		RAM2:			
□ ROM3:	- o		RAM3:		<u> </u>	
on-chip			on-chin		1	
✓ IROM1: 0x0 0x20000	· ·		IRAM1:	0x20000000	0x2000	
IROM2:	0		IRAM2:			
, ,			,			

图 1-3 配置 ROM 和 RAM

配置输出文件格式

下载工具 Programmer 支持 BIN 二进制文件下载格式,所以配置输出文件格式为 BIN 格式。

User 命令行选项中转换*.axf 文件为*.bin 文件,如图 1-4 所示。

图 1-4 配置输出文件格式

Command Items			Stop on Evi	
	oser command		Stop on Exi	J
Before Compile C/C++ File		-		_
Run #1			Not Specified	
Run #2		Ê	Not Specified	
Before Build/Rebuild				
Run #1		2	Not Specified	
Run #2		2	Not Specified	
After Build/Rebuild		_		
🔽 Run #1	C:\Keil_v5\ARM\ARMCC\bin\fromelf.exebin -o	.\Ob	jects\led.bin .\(Dbjects\led.
Run #2			Not Specified	
🔲 Run 'After-Build' Conditionally				
✓ Beep When Complete	☐ Start Debugging			
	OK Cancel Defaults		н	alp

命令格式为:

- Run #1

• fromelf.exe --bin -o bin-file axf-file

配置宏定义和头文件路径

配置宏定义"USE_STDPERIPH_DRIVER",用来调用标准外设。 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。

图 1-5 配置宏定义和头文件路径

Device Target Output Listing User	C/C++ .sm Linker Debug	Utilities
Prenrocessor Symbols Define: USE_STDPERIPH_DRIVER		
Language / Code Generation Execute-only Code Optimization: Level 0 (O0) Optimize for Time Split Load and Store Multiple One ELF Section per Function	Strict ANSI C Enum Container always int Plain Char is Signed Read-Only Position Independent Read-Write Position Independent	Wamings: All Wamings Thumb Mode No Auto Includes C99 Mode
Include Paths\PERIPH\Includes\ Misc Controls Compiler	STARTUP;\SYSTEM;\USER apcs=interwork -split_sections -I/CORE USER	-1/PERIPH/includes -1
OK	Cancel Defaults	Help

配置下载工具

如果使用 Keil 软件内部下载方法,则配置 Flash 下载工具为外部工具 programmer_cli,如图 1-6 所示。

图 1-6 配置下载选项

Device Target Output Listing User C/C++ Asm Linker Debug Utilities
Configure Rash Menu Command
C Use Target Driver for Flash Programming 😥 Use Debug Driver
Use Debug Driver Settings 🔽 Update Target before Debugging
Init File:
Use External Tool for Flash Programming
Command: D:\Gowin\1.8\programmer2\bin\programmer_cli.exe
Arguments: -device GW1NS-2C -run 22 f D:\user-bak\Users\root\Desktop\ed\PROJECT\Objects\ed.bin
✓ Run Independent
Configure Image File Processing (FCARM):
Output File: Add Output File to Group:
STARTUP
Image Files Root Folder:
OK Cancel Defaults Help

Command 为 Programmer 路径。

Arguments 为 Programmer 命令选项,如--device GW1NS-2C --run 22

-f bin-file .

配置调试选项

如图 1-7 所示:

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

图 1-7 配置调试选项

Device Target Output Listing User C/C++ A	sm Link <mark>r</mark> Debug Vilities
O Use Simulator with restrictions Settings	Use: ULINK2/ME Contex Debugger Y Settings
Limit Speed to Real-Time	ULINK2/ME Cortex Debugger
C Load Application at Startup Initialization File: Edit Edit	CMSIS DaP Debugger CMSIS DaP Debugger CMSIS DaP Debugger Debugger PEMicro Debugger PEMicro Debugger PEMicro Debugger
Restore Debug Session Settings	Restore Stellaris ICDI
Image: Breakpoints Image: Toolbox Image: Watch Windows & Performance Analyzer Image: Memory Display Image: System Viewer	Image: First Stabs UDA Debugger Image: Altera Blaster Cortex Debugger Image: First Stabs Image: Altera Blaster Cortex Debugger Image: Altera Blaster C
CPU DLL: Parameter:	Driver DLL: Parameter:
SARMCM3.DLL -MPU	SARMCM3.DLL -MPU
Dialog DLL: Parameter:	Dialog DLL: Parameter:
DCM.DLL -pCM3	TCM.DLL pCM3
Manage Component Vie	swer Description Files
OK Can	cel Defaults Help

调试接口类型配置为 JTAG, 如图 1-8 所示。

图 1-8 调试接口类型

Debug Trace Flash Download				
ULINK USB - JTAG/SW Adapter		Device Chain		
Serial No: Any		IDCODE	Device Name	IR len Move
ULINK Version: ULINK2	TDO	⊙ 0x4BA00477	ARM CoreSight JTAG-DP	4 Up
Device Family: Cortex-M	TDI			Down
Firmware Version: V2.03	• Au	, tomatic Detection	ID CODE:	
SWJ Port: JTAG 👻	O Ma	anual Configuration	Device Name:	
Max Clock: 1MHz	Add	d Delete U	odate IR len:	AP: 0x00
Connect & Reset Options Connect: Normal Reset For Reset after Connect	: Autode	tect 💌	Cache Options ──── Dow ✓ Cache Code □ ✓ Cache Memory □	vnload Options Verify Code Download Download to Flash
LI LOG DEDUG Accesses I S	top after F			Halp

1.2.3 工程编译

完成编码和工程配置后,编译生成 Gowin_EMPU 二进制 BIN 文件,如 图 1-9 所示。

图 1-9 工程编译

le Edit View Project Flash De	bug Peripherals Tools SVCS Window Help
1 G B B 5 2 8 10	요 수 요 🖻 🎊 🔃 連 /// /// 🧭 read_rx_bmc_decoder_or 🚽 🖈 🧖 🍳 🗕 🔹 🔗 🍓 🛄 🔫
🕮 🖽 👂 🖂 🙀 Ied	
d 🔹 😰	i main.c
S Project: led	1
🖻 闷 led	2日/*
🕀 🦢 STARTUP	3 • • • • • • • • • • • • • • • • • • •
I CORE	4 *
SYSTEM	S Copyright (C) 2014-2018 Gowin Semiconductor Technology Co., Ltd.
DEPIDHERAL	7 Bfile main.c
O CON LICED	8 * Sauthor Embedded Development Team
B OSER	9 * Sversion V1.0.0
a gwinszk_it.c	10 * 8date 2018-5-1 09:00:00
I main.c	11 * @brief Main program body.
	12
	13 7/
	15 /* Includes
	16 #include "gwins2k.b"
	17 #include <atdio.h></atdio.h>
	18 #include "gwlns2k_gpio.h"
	19 #include <stdlib.h></stdlib.h>
	20 #include <atring.h></atring.h>
	21 VOID DELASY(D ULINESZ_C RECOURD)//258 18 = 6555000
	<pre>23 for(: nCount != 0: nCount);</pre>
	24 1
	25
	26 int main (vold)
	27 🗐 (
	28 Systeminit();
	30 GRIOD-DONTENSET = OVEREFEET.
	31 while (1)
	32 📋 (
	33 GPI0_SetBit(GPI00,GPI0_Pin_1)GPI0_Pin_2)GPI0_Pin_3);
	34 GPIO_ResetBit(GPIO0, GPIO_Pin_0);
	35 //GPICO->DATACUT = 0xle;
	36 Delay(8333000); 37 GETO SetBir (BETO Sin DigETO Bin 21GETO Bin 3);
	38 GPID ResetEit (GPID0.GPID Pin 1):
	39 //GPIOO->DATAOUT = 0xd;
	40 Delay (8333000):
	41 GPI0_SetBit(GPI00,GPI0_Pin_1)GPI0_Pin_0)GPI0_Pin_3);
	42 GPIO_ResetBit(GPIO0, GPIO_Pin_2);
	43 //GPICO-DERIACUT = 0x10; 44 Delaw(232000);
т 🕜 Ва () Ри (Ц. Те	
Output	
iling main.c	
king	
gram Size: Code=580 RO-data=	224 RW-data=16 2I-data=1632
r suiid - User command #1: 4	::\vell_vb\AWY\AWX\C\DIN\TromeIT.exeDIN -0 .\UDJects\ied.DIN .\UDJects\ied.DXf 0 Name/ch
d Time Elanced: 00.00.02	V. as man unsulf (also

1.2.4 工程下载

完成工程配置和编译后,可以使用两种下载方式:

- 第一种方法:点击 ARM Keil MDK 工具栏下载按钮,调用上述已经配置的外部命令行工具 programmer_cli,下载 Gowin_EMPU 二进制 BIN 文件。
- 第二种方法:打开 Programmer 工具,配置器件访问模式为 MCU Mode, 操作选项选择 Firmware Erase, Program, Firmware File 导入
 Gowin_EMPU 二进制 BIN 文件,进行下载,如图 1-10 所示。

图 1-10 MCU 下载

ò Device co	onfiguration	ζ
-Device Op	eration	٦l
Access Mo	le: MCV Mode 🗸	
Operation	Firmware Erase, Program 👻	
Firmware	Erase, Program	
- FW Option	5	ור
Firmware	 File: D:/user-bak/Users/root/Desktop/systick_bin	
TTIMMATE.		
	Save	

1.2.5 工程调试

完成 Gowin_EMPU 二进制 BIN 文件下载后,如果用户设计出现问题,可以连接 U-LINK 或 J-LINK 仿真器使用调试功能。

JTAG 模式切换

如图 1-11 所示,使用 Programmer 下载软件将 JTAG 模式由下载模式 切换到调试模式。

打开 Programmer,器件访问配置为 MCU Mode,操作模式配置为 Connect to JTAG of MCU,完成 JTAG 模式切换。

图 1-11 JTAG 模式切换命令

🙀 Device configuratio	n	8 X
Device Operation		
Access Mode:	MCV Mode	-
Operation:	Connect to JTAG of MCV	•
Connect to JTAG of	MCV.	
		[[
	Save	Lancel

JTAG 接口切换

如果使用开发板"DK-EVAL-GW1NS2 V1.1", 需要手动将 JTAG 接口 TMS、TCK、TDI 和 TDO 跳线帽由 FDTI 下载切换到 ARM 下载,即由(1,3) 端口切换到(3,4)端口。

如果使用开发板"DK-START-GW1NS2 V1.1",需要手动将 JTAG 接口 TMS、TCK、TDI 和 TDO 拨码开关由 FPGA 下载切换到 ARM 下载。

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。

图 1-12 启动调试



1.3 参考设计

Gowin_EMPU for GW1NS-2C 支持 ARM Keil MDK 软件环境的参考设计:

Gowin_EMPU\ref_design\MCU_RefDesign\Keil_RefDesign

2GOWIN MCU Designer

2.1 软件安装

高云半导体官网提供 GOWIN MCU Designer 软件安装包下载。

GOWIN MCU Designer 软件安装与配置请参考 SUG549,<u>《GOWIN</u> <u>MCU Designer 用户指南》</u>。

2.2 工程模板

2.2.1 工程创建

新建工程

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

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

图 2-1 图新建工程



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。

Select Configura Select platforms		
Project type: Toolchains: Configurations:	Executable Cross ARM GCC	
 ♥ Debug ♥ Release 		Select all Deselect all
		Advanced settings
Use "Advanced Additional confi Use "Manage co	settings" button to edit project's properties. gurations can be added after project creation. onfigurations" buttons either on toolbar or on property	pages.
?	< Back Next > Fin	ish Cancel

图 2-2 选择平台类型配置

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,如图 2-3 所示。

图 2-3 选择工具链和路径

🖨 C Project		_ D X
Cross GNU ARM	Toolchain ain and configure path	
Toolchain name: Toolchain path:	GNU Tools for ARM Embedded Processors (arm-none-eabi-gcc) D:\Gowin_MCU_Designer\toolchain\GNU Tools ARM Embedded\5.4 2016q2\bin	▼ Browse

建立工程结构

完成新建工程后,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,如图 2-4 所示。

- CORE: ARM Cortex-M3 内核定义
- PERIPHER: 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
- gw1ns2c_flash.ld: Flash 链接脚本

图 2-4 工程结构

퉬 .settings	
퉬 CORE	
퉬 Debug	
)) PERIPHER	
STARTUP	
SYSTEM	
🌗 USER	
.cproject	
.project	
gw1ns2c_flash.ld	

导入工程

选择当前工程,右键选择 Refresh,自动导入如图 2-4 所示的工程结构和代码。

2.2.2 工程配置

选择当前工程,右键选择 Properties,选择选项卡 C/C++ Build,选择 Setting 选项,如图 2-5 所示。

图 2-5 选择 Properties

Properties for timer			
type filter text	Settings		← → ⇒ ▼
Resource			
Builders			
▲ C/C++ Build	Configuration: Debug [Active]		 Manage Configurations
Build Variables			
Environment	ND Test Continue IND as a training and	× • • • •	
Logging	Toolchains Devices	P Build Steps	Build Artifact
Settings	Target Processor	ARM family	cortex-m3
Tools Daths	Optimization	A 12 A	T 11 1 1 6 h
XI C/C++ Compiler	🖄 Warnings	Architecture	l ooichain defauit
C/C++ General	🖄 Debugging	Instruction set	Thumb (-mthumb)
Linux Tools Path	A Son Cross ARM GNU Assembler	Thumb interwor	k (-mthumb-interwork)
Project References	Preprocessor	Endianness	Toolchain default
Run/Debug Settings	2 Includes		
> Task Repository	2 Warnings	Float ABI	Toolchain default
WikiText	Miscellaneous	FPU Type	Toolchain default
	Preprocessor	Upaligned accord	Toolchain default
	A Includes	Unaligned access	
	Optimization	AArch64 family	Generic (-mcpu=generic)
	🖉 Warnings	Feature crc	Toolchain default
	🖄 Miscellaneous	Eastura covota	Toolchain dafault
	Scross ARM C Linker	reature crypto	
	A General	Feature fp	Toolchain default
	Libraries	Feature simd	Enabled (+simd)
	Miscellaneous		
	General		Small (-mcmodel=small)
	A 🐼 Cross ARM GNU Print Size	Strict align (-mst	trict-align)
	🖄 General	Other target flags	
			4
			Restore Defaults Apply
?			OK Cancel

配置 Cross ARM GNU Assembler Preprocessor

选择 Cross ARM GNU Assembler 中的 Preprocessor, 配置汇编宏定义 ___STARTUP_CLEAR_BSS, 如图 2-6 所示。



图 2-6 配置 Cross ARM GNU Assembler Preprocessor

配置 Cross ARM GNU Assembler Includes

选择Cross ARM GNU Assembler 中的 Includes, 配置汇编引用文件路径, 如图 2-7 所示。



图 2-7 配置 Cross ARM GNU Assembler Includes

以 GNU_RefDesign 参考设计为例,Assembler 头文件路径配置如下所示。

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

配置 Cross ARM C Compiler Preprocessor

选择 Cross ARM C Compiler 中的 Preprocessor, 配置 C 宏定义 USE_STDPERIPH_DRIVER, 如图 2-8 所示。



图 2-8 配置 Cross ARM C Compiler Preprocessor

配置 Cross ARM C Compiler Includes

选择 Cross ARM C Compiler 中的 Includes, 配置 C 引用文件路径, 如 图 2-9 所示。



图 2-9 配置 Cross ARM C Compiler Includes

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}"

配置 Cross ARM C Linker

选择 Cross ARM C Linker 中的 General, 配置 Flash 链接脚本 gw1ns2c_flash.ld, 如图 2-10 所示。



图 2-10 配置 Cross ARM C Linker

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

"\${workspace_loc:/\${ProjName}/gw1ns2c_flash.ld}"

配置 Cross ARM GNU Create Flash Image

选择 Cross ARM GNU Create Flash Image 中的 General, 配置 Gowin_EMPU 映像文件格式为 RAW binary, 如图 2-11 所示。



图 2-11 配置 Cross ARM GNU Create Flash Image

配置 Devices

选择 Devices 选项卡, 选择器件 ARM Cortex-M3 "ARMCM3", 如图 2-12 所示。

Name		Details				
⊿ ARI	M	Vendor				
\triangleright	ARM Cortex M0	Family (12	kB RAM, 256 kB ROM)			
\triangleright	ARM Cortex M0 p	us Family (12	kB RAM, 256 kB ROM)			
⊳	ARM Cortex M1	Family (12	KB RAM, 256 KB ROM)			
۵	ARM Cortex M3	Family (12	Family (128 kB RAM, 256 kB ROM)			
	ARMCM3	Device (Co	ex-M3, r2p1, 10 MHz)			
⊳	ARM Cortex M4	Family (12	Family (128 kB RAM, 256 kB ROM) Family (128 kB RAM, 256 kB ROM)			
⊳	ARM Cortex M7	Family (12				
ь	ARM Cortex M23	Family (25)	LB RAM 4096 LB ROM)			
evice core	: Cortex-M3					
Memory ma ARMCM3	ap (Warning: Not y	et used to gener	te the linker scripts!)			
Section	Start	Size	Startup			
IRAM1	0x20000000	0x00020000	0			

图 2-12 配置 Devices

2.2.3 工程编译

完成工程配置和编码后,编译工程,选择工具栏编译按钮,开始编译, 如图 2-13 所示。

图 2-13 编译工程



2.2.4 工程下载

完成工程配置和编译后,可以使用两种下载方法:

- 第一种方法:点击 GOWIN MCU Designer 工具栏下载按钮,调用已经 配置的下载工具 download,下载 Gowin_EMPU 二进制 BIN 文件。
- 第二种方法:打开 Programmer 工具,配置器件访问模式为 MCU Mode, 操作选项选择 Firmware Erase, Program, Firmware File 导入 Gowin_EMPU 二进制 BIN 文件,下载 Gowin_EMPU 二进制 BIN 文件, 如图 2-14 所示。

图 2-14 工程下载

🙀 Device co	onfiguration	8 ×
-Device Op	eration	
Access Mo	le:	MCV Mode 🔹
Operation	Fi	rmware Erase, Program 💌
Firmware	Erase, Program	
-FW Uption	.5	
Firmware	File: D:/user=b	ak/Users/root/Desktop/systick.bin
		Save Cancel

2.2.5 工程调试

配置调试选项

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 2-15 所示。

图 2-15 建立调试配置选项

选择已建立的调试配置选项。

1. 选择 Main 选项卡, 配置映像文件路径, 如图 2-16 所示。

图 2-16 配置 Image 路径

Debug Configurations					
Create, manage, and run configurations					
Image: System of the syste	Name: Icd Debug Main Bebugger Startup Source Common Project: Icd C/C++ Application: C/C++ Application:				
Cij C/C++ Unit DSF PDA Application C GDB Hardware Debugging C GDB OpenOCD Debugging C GDB PyOCD Debugging C GDB QEMU Debugging C GDB SEGGER J-Link Debugging	Build (if required) before launching Build Configuration: Select Automatically Enable auto build Use workspace settings				
C Icd Debug ► Launch Group Qt Qt Local Application	Se workspace settings				

2. 选择 Debugger 选项卡, 配置 Debugger 器件名称和调试接口为 JTAG, 如图 2-17 所示。

- Device Name: Cortex-M3
- Interface: JTAG
- Other options: -singlerun -timeout
- 图 2-17 配置 Debugger

📄 Main 🗇 Debugger 🔪 🕨 Startup 🦆 Source 🔲 Common							
	J-Link GDB Server Setup						
	✓ Start the J-Li	nk GDB server locally					
	Executable:	\${jlink_path	}/\${jlink_gdbse	erver}			
	Device name:	Cortex-M3					
	Endianness:	Little	🔘 Big				
	Connection:	USB	⊚ IP			(USB serial or IP name/address)	
	Interface:	🔘 SWD	JTAG				
	Initial speed:	Auto	O Adaptive	Fixed	1000 kH	Hz	
	GDB port:	2331					
	SWO port:	2332					
	Telnet port:	2333					
	Log file:			_			
	Other options:	-singlerun ·	-timeout				
	Allocate con	sole for the	GDB server	-			
	GDB Client Setup						
	Executable:	{cross_pre	efix}gdb\${cross	s_suffix}			
	od	ns:					
	Other options:						
	Commands:	set mem inaccessible-by-default off					

JTAG 模式切换

完成调试模式配置后,可以使用两种方法完成 JTAG 模式切换:

- 第一种方法:调用上述软件配置中已配置的 JTAG 模式切换工具 switchmode,完成 JTAG 软件模式由下载模式切换到调试模式。
- 第二种方法:打开 Programmer,配置器件访问模式为 MCU Mode,操 作模式为 Connect to JTAG of MCU,完成 JTAG 软件模式由下载模式切 换到调试模式。

JTAG 接口切换

如果使用开发板"DK-EVAL-GW1NS2 V1.1", 需要将开发板上的 JTAG 接口 TMS、TCK、TDI、TDO 四个跳线帽由(1,3)端切换到(3,4)端口, 完成 JTAG 接口切换, 如图 2-18 所示。

如果使用开发板"DK-START-GW1NS2 V1.1",需要将开发板上的 JTAG 接口 TMS、TCK、TDI、TDO 四个拨码开关由 FPGA 下载切换到 ARM 下载。



图 2-18 JTAG 接口切换

调试

连接 J-LINK 仿真器,选择工具栏 Debug 按钮,开始调试,如图 2-19 所示。

图 2-19 启动调试

Poebug 🕄	🙀 😥 🕾 🗖 😳 Variables 🕄 🧌	Broakpoints 🧏 Peripherals	실 ·4 B [1] 번 조 ·
C Icd Debug [GDB SEGGER J-Link Debugging]	Name	Type	Value
🔺 🔡 lcd.elf			
a 🧬 Thread #1 57005 (Suspended : Breakpoint)			
main() at main.c:23 0x15c			
JinkGDBServerCLexe			
arm-none-eabi-gdb			
j∐ Semihosting and SWV			
	4		
			Design of the second back a second
main.c 61		- 0	Ge Outline 23 2 Visualizer & El 1% of % • 4
50 Copyright (C) 2014-2018 Gowin Semiconductor Technology Co.,Ltd.		*	gwlns2k.h
15 /* Includes*/		1	gwins2k_lcd1602.h
16 #include "gwlns2k.h"			1 stdio.h
17 #include _gwinszk_icd1602.h* 18 #include /stidio h>			stdib.h
19 #include <stdlib.h></stdlib.h>			main() : int
20			
21= int main()			
<pre>23 SystemInit();</pre>			
24		1	
<pre>25 Lcd_Fin_Config(); 26 Lcd_Init();</pre>			
27			
<pre>28 Lcd_Write_String(" " Gowin GwINS-2C " ",0);</pre>			
38 CCG_WITE_SCING(HELLO HOTEL (1))			
<pre>31 while(1);</pre>			
32 }		UI.	
			1
Console 21 🕘 Tasks 🛣 Problems 🕡 Executables			- X X X X P (10)
d Debug (GDB SEGGER J-Link Debugging) JLinkGDBServerCL.exe			
tarting target CPU			
ereakpoint reached g address exeeeeisc eading all registers			
ead 4 bytes 🖗 address 0x0000015C (Data = 0x47804809)			
enoving breakpoint 8 address 0x0000015C. Size = 2			

2.3 参考设计

Gowin_EMPU for GW1NS-2C 支持 GOWIN MCU Designer 软件环境的 参考设计:

Gowin_EMPU\ref_design\MCU_RefDesign\GNU_RefDesign

