GigaDevice Semiconductor Inc.

Arm[®] Cortex[®]-M3/4/23/33 32-bit MCU

Application Note AN041



Table of Contents

Tab	le o	f Contents	.2
List	of	Figures	.4
List	of	Tables	. 6
		volonment environment	7
1.	Dev		. /
2.	Pro	ject development	. 8
2.	1.	New project	. 8
2.2	2.	New project folder and add files	12
	2.2.1	1. Ceate folders and add files manually	12
	2.2.2	2. Ceate folders and add files by "Refresh"	23
2.3	3.	Project configurations	25
	2.3.1	1. Target Processor option configuration	26
	2.3.2	2. Optimization option configuration	27
	2.3.3	3. GNU Arm Cross C Compiler configuration	28
	2.3.4	4. GNU Arm Cross C Linker configuration	30
	2.3.5	5. Build Steps configuration-generate bin file	32
2.4	4.	Build project	33
2	5	Use .I-l ink to download and debug the project	34
	2.5.1	Debug configuration interface.	34
	2.5.2	2. Main tab	35
	2.5.3	3. Debugger tab	36
	2.5.4	4. SVD Path tab	36
2	6.	Use GD-I ink to download and debug the project	37
	2. 6.1	1. Debug configuration interface	37
	2.6.2	2. Main tab	38
	2.6.3	3. Debugger tab	38
	2.6.4	4. SVD Path tab	39
2.	7.	Debug interface	39
	2.7.1	Toolbar introduction	41
	2.7.2	2. Registers view	42
	2.7.3	3. Peripherals view	43
	2.7.4	4. Memory view	45
	2.7.5	5. Expressions view	46
	2.7.6	6. Disassembly view	47
	2.7.7	7. Exit the Debug perspective	48
3.	Imp	oort an existing project	49



List of Figures

Figure 2-1. New ARM C project	8
Figure 2-2. Select C Managed Build	9
Figure 2-3. Create new ARM project name and select project storage path	. 10
Figure 2-4. Select ARM cross toolchain path	11
Figure 2-5. Project perspective	11
Figure 2-6. New project folder	. 12
Figure 2-7. New virtual sub-folder	. 13
Figure 2-8. ARM project view	. 14
Figure 2-9. Add files	. 15
Figure 2-10. Select files to be imported	. 16
Figure 2-11. Import files to the Application folder	. 17
Figure 2-12. Import files to the CMSIS folder	. 18
Figure 2-13. Import files to the Doc folder	. 18
Figure 2-14. Import files to the LD folder	. 19
Figure 2-15. Import files to the Peripherals folder	. 20
Figure 2-16. Import files to the Startup folder	. 21
Figure 2-17. Import files to the Utilities folder	. 22
Figure 2-18. Final ARM project view	. 23
Figure 2-19. Project folder structure	. 24
Figure 2-20. Refresh the project	. 24
Figure 2-21. Project structure in Eclipse IDE	. 25
Figure 2-22. Project properties configurations	. 26
Figure 2-23. Target Processor configuration	. 27
Figure 2-24. Optimization configuration	. 28
Figure 2-25. GNU Arm Cross C Compiler -> Preprocessor configuration	. 29
Figure 2-26. GNU Arm Cross C Compiler -> Includes configuration	. 30
Figure 2-27. GNU Arm Cross C Linker -> General configuration	. 31
Figure 2-28. GNU Arm Cross C Linker -> Miscellaneous configuration	. 32
Figure 2-29. Build Steps configuration	. 33
Figure 2-30. Build project	. 33
Figure 2-31. Build ARM project completed	. 34
Figure 2-32. Enter Debug Configuratios interface	. 35
Figure 2-33. GDB SEGGER J-Link Debugging-Main tab	. 35
Figure 2-34. GDB SEGGER J-Link Debugging-Debugger tab	. 36
Figure 2-35. GDB SEGGER J-Link Debugging-SVD Path tab	. 37
Figure 2-36. Enter Debug Configuratios interface	. 37
Figure 2-37. GDB OpenOCD Debugging-Main tab	. 38
Figure 2-38. GDB OpenOCD Debugging-Debugger tab	. 39
Figure 2-39. GDB OpenOCD Debugging-SVD Path tab	. 39
Figure 2-40. Enter Debug perspective -1	. 40

AN041 GD32 ARM MCU Eclipse development tutorial for Windows

Windows	tutorial for	development	Eclipse	MCU	ARM	GD32	GigaDevice
40			ective -2	ig persp	ter Debu	e 2-41. E	Figur
41				spective	bug per	e 2-42. D	Figur
42			w	sters vie	en Regi	e 2-43. O	Figur
43				/iew	gisters v	e 2-44. R	Figur
44			'iew	oherals v	en Perip	e 2-45. O	Figur
45				s view	ripherals	e 2-46. P	Figur
46				ory view	en Mem	e 2-47. O	Figur
46				ew	emory vie	e 2-48. N	Figur
47			view	essions	en Expr	e 2-49. O	Figur
47				ns view	pressior	e 2-50. E	Figur
47			view	ssembly	en Disa	e 2-51. O	Figur
48				ly view	sassemb	e 2-52. D	Figur
48			spective	bug pers	it the De	e 2-53. E	Figur
49			roject - 1	cisting p	ort an ex	e 3-1. Im	Figur
50			roject - 2	cisting p	ort an ex	e 3-2. Im	Figur
51		igging in RAM	when debug	ory map	ile mem	e 4-1. Ld	Figur
51	g in RAM	ble when debugging	pt vector tal	e interrup	ocate the	e 4-2. Re	Figur
52		bugging in RAM	ns when deb	iguration	ug confi	e 4-3. De	Figur
53		ging in RAM	vhen debug	pective w	ug pers	e 4-4. De	Figur
55		uration	lata configu	g point d	t floating	e 5-1. Pri	Figur



AN041 GD32 ARM MCU Eclipse development tutorial for Windows

List of Tables

Table 6-1	. Revision	history		56
-----------	------------	---------	--	----



1.

Development environment

- Development boards: GD32 MCU boards
- Hardware Debugger: J-Link V9/V10 or GD-Link
- Operating system: WIN7 64-bit OS
- IDE: eclipse-embedcpp-2021-03-R-win32-x86_64
- Cross toolchains: xpack-arm-none-eabi-gcc-10.2.1-1.1-win32-x64
- Build Tools: gnu-mcu-eclipse-windows-build-tools-2.12-20190422-1053-win64
- GDB server: OpenOCD / J-Link GDB Server CL V7.54b

2. **Project development**

2.1. New project

Open Eclipse, LAUCH eclipse-workspace. Under "File->New" option, uesr can choose to create a new C/C++ Project and select C Managed Build option.

e e	clipse-workspace - Eclipse IDE			
File	Edit Source Refactor Navigate	Search Projec	t Ru	ın Window Help
	New	Alt+Shift+N ▶	C++	Makefile Project with Existing Code
	Open File		ت	C/C++ Project
	Open Projects from File System			Project Create a new C or C++ project
	Recent Files	+	C++	Convert to a C/C++ Autotools Project
	Close Editor	Ctrl+W	C++	Convert to a C/C++ Project (Adds C/C++ Natur
	Close All Editors	Ctrl+Shift+W	62	Source Folder
	Save	Ctrl+S	C	Folder
	Save As		C	Source File
Ð	Save All	Ctrl+Shift+S	h	Header File
	Revert			File from Template
	Move		G	Class
Ŕ	Rename	F2		Example
\$	Refresh	F5		Other
	Convert Line Delimiters To	+	Г	
Ð	Print	Ctrl+P		
\geq	Import			
4	Export			
	Properties	Alt+Enter		
	Switch Workspace	+		
	Restart			
	Exit		а т.	estas 🖻 Concela 🗖 Dana anti-es

Figure 2-1. New ARM C project



Figure 2-2. Select C Managed Build

New C/C++ Project									
Templates for	New C/C++ Project								
All CMake Make	C Managed Build A C Project build using the CDT's managed build system.								
Meson	C++ Managed Build A C++ Project build using the CDT's managed build system. ■								
	CMake Project A CMake project with a Hello World executable to get started.								
	Empty or Existing CMake Project Create a CMake project with no files. Can be used to create one over existing content.								
	<								
?	< Back Next > Finish Cancel								

Enter the "Project name" and configure the project type. For convenience, it is recommended to put the project in the FW directory. The compilation chain is selected as ARM Cross GCC.



Figure 2-3. Create new ARM	project name and select	project storage path
J		

		×						
C Project ① Directory with specified name already exists.		Ď						
Project name: Project Use default location Location: F:\eclipse-workspace\GD32F10x_Firmware_Li Choose file system: default	brary_V2.2.1\Template\Eclipse_project	Browse						
Project type: → GNU Autotools → Executable → Empty Project → Hello World ANSI C Project → Hello World Arm C Project → Hello World RISC-V C Project → Hello World RISC-V C Project → ADuCM36x C/C++ Project → Hello World Arm Cortex-M C/C++ Project → Freescale Processor Expert C/C++ Project → Freescale Processor Expert C/C++ Project → Freescale Kinetis KLxx C/C++ Project → SiFive RISC-V C/C++ Project → STM32F1xx C/C++ Project → STM32F1xx C/C++ Project → STM32F3xx C/C++ Project → STM32F3xx C/C++ Project → STM32F3xx C/C++ Project → STM32F7xx C/C++ Project → STM32F7xx C/C++ Project → Static Library → Makefile project	Toolchains: Arm Cross GCC Cross GCC MinGW GCC RISC-V Cross GCC							
Show project types and toolchains only if they are supported on the platform								
? < B	ack Next > Finish	Cancel						

If the Eclipse IDE has set the ARM Toolchains Path correctly, the path will be automatically selected here. If the Eclipse IDE has not set the ARM Toolchains Path, user can also select the absolute path to the ARM Toolchains here.



Figure 2-4. Select ARM cross toolchain path

•	in August that Medica (Meg	
GNU Arm Cross	Foolchain ain and configure path	Ď
Toolchain name:	xPack GNU Arm Embedded GCC (arm-none-eabi-gcc)	•
Toolchain path:	<pre>\${eclipse_home}\eclipse_toolchain\xpack-arm-none-eabi-gcc-10.2.1-1.1\bin</pre>	Browse
?	< Back Next > Finish	Cancel

Click "Finish" until the display interface is shown in *Figure 2-5. Project perspective*. At this point, the establishment of the Project is completed.

Figure 2-5. Project perspective

eclipse-workspace - Ec	lipse IDE									x
File Edit Source Ref	actor Navigate	Search Pr	roject Run	Window	Help					
	🖌 🕶 🔜 🔯 🔻	🚳 🔻 💽 🔻	6 - *	- 🜔 -	🧛 👻 💁 👻	10 10 1	🝷 🔛 🗉 п 📃	🗣 🔍 🕹		
b • A • ↔ ↔ ↔	• 🗢 • 🖻								۹ 🗄	
陷 Proje 🛛 🗖 🗖							- 6	1 🗄 o 🛙	🖲 B 🔓 D 👘	
🗆 🛱 🏹 🕴										8
▲ E Project (in Eclipse_)								There is n	o active editor t	hat
▲ B Includes								provides a	an outline.	
E:/eclipse/										
E:/eclipse/										
	🖹 Problems 🛛	🛛 🧔 Tasks	📮 Console	🔲 Prope	erties				78	
	0 items									
	Description	~			Resource	Path	Location	Туре		_
4										
Project									- m 👝	2 6
					1					



2.2. New project folder and add files

2.2.1. Ceate folders and add files manually

Right-click the project name and select "new->Folder".

Figure 2-6. New project folder

	10	E.P. 1 (A)			
4 🔊 In		New	۱.	2	Project
⊳ 🕒		Go Into		Ľ	File
⊳ 🕒		Open in New Window		₽	File from Template
⊳ 🕒		Show In	Alt+Shift+W > Ctrl+C Ctrl+V Delete F2	8 6 6 6 6 6 6 6	Folder
		Show in Local Terminal			Class
		Сору			Header File
	Ē	Paste			Source File
	×	Delete			Source Folder
		Source			C/C++ Project
		Move			Convert to a C/C++ Project (Adds C/C++ Nature)
		Rename		_	
	n des	Import			Example
					Other Other

Create a virtual folder "Peripherals".



New Folder	100	
Folder		
Create a new tolder resource.		
Enter or select the parent folder:		
Project		
🖄 🗇 🖒		
▷ 🚰 Project		
Folder name: Peripherals		
Ccc Advanced		
Use default location	tual Caldar)	
Folder is not located in the life system (viii) Cathering to alternate location (Linked Folder)	rtual Folder)	
	Browse	Variables
Choose file system: default 💌		
Resource Filters		
?	Finish	Cancel

Create the Application, CMSIS, Doc, Ld, Startup and Utilities folders in the same way.



Figure 2-8. ARM project view



Right-click "Application" and select the "Import" option to import the file.



Figure 2-9. Add files



Import select "File System". Select the path of the file to be imported, and tick the file to be imported.



Figure 2-10. Select files to be imported





E Import		
File system Import resources from the local file system.		
From directory: F:\eclipse-workspace\GD32F10x_Firmwar	re_Library_V2.2.1\Examples\GPIO\Running_led 👻	Browse
Running_led	Image: gd32f10x_it.c Image: gd32f10x_it.h Image: gd32f10x_libopt.h Image: gd32f10x_l	
Filter Types Select All Deselect All		Province
Options Overwrite existing resources without warning Create top-level folder << Advanced		Diowsen
 ✓ Create links in workspace ✓ Create virtual folders ✓ Create link locations relative to: PROJECT_LOC 	•	
?	< Back Next > Finish	Cancel

In the same way, import the required files into the "CMSIS", "Doc", "Ld", "Peripherals", "Startup" and "Utilities" folders.



Figure 2-12. Import files to the CMSIS folder

E Import		
File system Import resources from the local file system.		
From directory: F:\eclipse-workspace\GD32F10x_Firmware_Library_V	/2.2.1\Firmware\CMSIS\GD\GD32F10x\Source •	Browse
D 🔲 🔁 Source	🔽 🖻 system_gd32f10x.c	
Filter Types Select All Deselect All		
Into folder: Project/CMSIS		Browse
Options		
Overwrite existing resources without warning Create top-level folder		
<< Advanced		
♥ Create links in workspace		
Create virtual folders		
Create link locations relative to: PROJECT_LOC •		
?	< Back Next > Finish	Cancel

Figure 2-13. Import files to the Doc folder

Carl Import		
File system Import resources from the local file system.		
From directory: F:\eclipse-workspace\GD32F10x_Firmware_	Library_V2.2.1\Examples\GPIO\Running_led	Browse
Filter Types Select All Deselect All	C main.h C readme.txt C systick.c	- -
Into folder: Project/Doc Options Overwrite existing resources without warning Create top-level folder << Advanced		Browse
 ✓ Create links in workspace ✓ Create virtual folders ✓ Create link locations relative to: PROJECT_LOC 		
?	< Back Next > Finish	Cancel



Figure 2-14. Import files to the LD folder

File system Import resources from the local file system.		
From directory: Ft/eclipse-workspace\GD32F10x_Firmware_Library_V	/2.2.1\Firmware\CMSIS\GD\GD32F10x\Source\GCC ✔	Browse
	 S startup_gd32f10x_cl.S S startup_gd32f10x_hd.S 	=
	S startup_gd32f10x_md.S	-
Filter Types Select All Deselect All		
Into folder: Project/Ld		Browse
Ontions		
Covers to a level felder		
<< Advanced		
 ✓ Create links in workspace ✓ Create virtual folders ✓ Create link locations relative to: PROJECT_LOC ▼ 		
3	< Back Next > Finish	Cancel



E Import		
File system Import resources from the local file system.		
From directory: :\eclipse-workspace\GD32F10x_Firmware_Library_V2	2.1\Firmware\GD32F10x_standard_peripheral\Source 👻	Browse
Filter Types Select All Deselect All Into folder: Project/Peripherals Options Overwrite existing resources without warning Create top-level folder << Advanced Ø Create links in workspace Ø Create links in workspace Ø Create link locations relative to: PROJECT_LOC	Image: Constraint of the state of the s	Browse
0	r Park Not a Finish	Cancel
\odot	Pinish	Cancer



Figure 2-16. Import files to the Startup folder

Import		
File system Import resources from the local file system.		
From directory: :\eclipse-workspace\GD32F10x_Firmware_Library_V2.2.1 Image: GCC Filter Types Select All Deselect All Into folder: Project/Startup Options Overwrite existing resources without warning Create top-level folder << Advanced	l\Firmware\CMSIS\GD\GD32F10x\Source\GCC ▼ gd32f10x_flash.ld S startup_gd32f10x_cl.S S startup_gd32f10x_hd.S S startup_gd32f10x_md.S S startup_gd32f10x_xd.S	Browse
Create link locations relative to: PROJECT_LOC	< Back Next > Finish	Cancel



C Import	
File system Import resources from the local file system.	
From directory: F:\eclipse-workspace\GD32F10x_Firmware_Library_V2.2.1\Utilities 👻	Browse
Image: box with the second state of	
Filter Types Select All Deselect All	
Into folder: Project/Utilities	Browse
Options Overwrite existing resources without warning Create top-level folder	
<< Advanced Create links in workspace Create virtual folders Create link locations relative to: PROJECT_LOC	
Pack Next > Finish	Cancel



Figure 2-18. Final ARM project view

eclipse-workspace - Project/Applicatio	on/main	c - Eclipse IDE	1000		100	2.4	100		۲	
File Edit Source Refactor Navigate	e Sear	ch Project Run	Window Help							
📑 🗝 🔚 🕞 📎 🕶 🇞 🖛 📷 💣 🕶	· 🚳 •	🖸 🕶 🎯 🕶 🔅	- 🜔 - 🚱 - 💁 -	1 🥭	🖻 🔗 🔻 🗆	/ R/ 🗉 👖	📮 🏪 🍬	188	Ŧ	<u>ې</u> ا
Project Explorer 💥 📃 🗖	1 🕞	nain.c 🕅								
	3 28	WARRANTIES OF	MERCHANTABILITY AND	FIT	IESS FOR A P	ARTICULAR P	URPOSE ARE D	DISCLAIME	o.	
▲ I ^{CC} Project (in Eclipse project)	29	IN NO EVENT SH	ALL THE COPYRIGHT H	IOLDER	OR CONTRIB	UTORS BE LI	ABLE FOR ANY	Y DIRECT,	_	
A M Includes	30	INDIRECT, INCI	DENTAL, SPECIAL, EX D. PROCUREMENT OF SU	(EMPLA JBSTTT	ARY, OR CONS TUTE GOODS O	EQUENTIAL D	AMAGES (INCL LOSS OF USE	LUDING, B F. DATA.	JT DR	
E:/eclipse/eclipse toolchain/	32	PROFITS; OR BL	SINESS INTERRUPTION	 HOW 	IEVER CAUSED	AND ON ANY	THEORY OF L	LIABILITY	,	
E:/eclipse/eclipse toolchain/	33	WHETHER IN CON	ITRACT, STRICT LIABI	LITY,	OR TORT (I	NCLUDING NE	GLIGENCE OR	OTHERWIS	E)	
E:/eclipse/eclipse toolchain/	34	OF SUCH DAMAGE	WAY OUT OF THE USE	: 01- 1	HIS SOFTWAR	E, EVEN IF /	ADVISED OF	THE POSSI	SILI	
	36	*/								
▶ ad32f10x it.c	37	#414- "-470	510							
⊳ Ramain.c	239	#include gd32	110x.eval.h"							
▷ R systick.c	40	<pre>#include "syst</pre>	ick.h"							
CMSIS	41	<pre>#include <stdi< pre=""></stdi<></pre>	o.h>							
4 🕞 Doc	42	∋ /*!								
🔜 readme.txt	44	\brief	main function							
4 🕞 Ld	45	\param[in]	none							
🙀 gd32f10x_flash.ld	47	\retval	none							
4 🚰 Peripherals	48	*/								
⊳ 🙀 gd32f10x_adc.c	49	⊖int main(void)								
⊳ 🙀 gd32f10x_bkp.c	51	د systick_co	onfig();							
⊳ 🛃 gd32f10x_can.c	52	gd_eval_le	<pre>ed_init(LED2);</pre>							
gd32f10x_crc.c	53	gd_eval_le	<pre>ed_init(LED3); ed_init(LED4);</pre>							=
⊳ 🛃 gd32f10x_dac.c	55	55 gd eval led init(<i>LED5</i>);								
⊳ 🛃 gd32f10x_dbg.c 🕴	≡ 56									
⊳ 🛃 gd32f10x_dma.c	58	/* tur	n on led2, turn off	led5	; */					
gd32f10x_enet.c	59	gd_eva	l_led_on(LED2);							
gd32f10x_exmc.c	60	gd_eva	<pre>il_led_off(LED5);</pre>							
⊳ 属 gd32f10x_exti.c	62	/* tur	n on led3, turn off	led2	2 */					
gd32f10x_fmc.c	63	gd_eva	l_led_on(<i>LED3</i>);							
gd32f10x_fwdgt.c	64	gd_eva	<pre>il_led_off(LED2); imc(1000);</pre>							
⊳ 🙀 gd32f10x_gpio.c	66	/* tur	n on led4, turn off	led3	*/					
⊳ 🙀 gd32f10x_i2c.c	67	gd_eva	l_led_on(<i>LED4</i>);							
⊳ 属 gd32f10x_misc.c	68	gd_eva	<pre>il_led_off(LED3);</pre>							
⊳ 🔒 gd32f10x_pmu.c	05	<	_1ms(1000),						•	*
⊳ 🛃 gd32f10x_rcu.c										
⊳ 🔒 gd32f10x_rtc.c		Problems 🛛 🖉 T	asks 🖳 Console 🔲 I	Prope	rties					
⊳ 🔒 gd32f10x_sdio.c	0 ite	ns								
⊳ 🙀 gd32f10x_spi.c	Des	cription			Resource	Path	Loca	ation	Тур	e
▷ 🙀 gd32f10x_timer.c										
⊳ 🙀 gd32f10x_usart.c										
⊳ 🙀 gd32f10x_wwdgt.c										
⊿ 🔁 Startup										
▷ 🔓 startup_gd32f10x_cl.S										
⊿ 🔁 Utilities										
▷ 🙀 gd32t10x_eval.c	*									

2.2.2. Ceate folders and add files by "Refresh"

In addition to the above-mentioned method of creating folders and importing corresponding files manually, user can also put the files that need to be imported together with its folders in the folder at the same level as the created .cproject file. In the Eclipse IDE, right-click the project name and select "Refresh" to import the folders and files into the project directly.



Figure 2-19. Project folder structure

白你
퉬 .settings
퉬 Debug
퉬 gd_libs
퉬 inc
퉬 ldscripts
퉬 src
.cproject
.project
📋 readme.txt

Figure 2-20. Refresh the project

📑 🗕 🖬 🖬) 📮 🏪 🔍 🕪 💷 🔤 🖓 :	3. 13. 12 i → i
Project Explore	r 🛛 🕞 🕏 🍸	° □ 🕸
▷ C gd32f10 ▲ C Project (▷ \ Binari	New Go Into	Þ
 ▷ Includ ▷ Applic ▷ CMSI: ▷ Debut 	Open in New Window Show In Show in Local Terminal	Alt+Shift+W ►
 Debug Doc Doc<td>Copy Paste Delete</td><td>Ctrl+C Ctrl+V Delete</td>	Copy Paste Delete	Ctrl+C Ctrl+V Delete
	Move Rename	F2
	Export Build Project	
\$	Refresh Refresh Close Project	F5



Figure 2-21. Project structure in Eclipse IDE



Note: The files and folders created in the "Refresh" method are all real, and once a file is deleted in the Eclipse IDE, the file will be deleted from the disk directly.

2.3. Project configurations

Right-click the project and select the "Properties" option to open it.



Project (in Eclipse_project) 35 OF SUCH 4 🗊 Incl New > 🕒 E Go Into > 🕒 E Open in New Window > 🕒 E Show In Alt+Shift+W ▶ 4 🗁 Apr Show in Local Terminal Þ 🛃 g ۲ 🖻 🖳 r 📄 Copy Ctrl+C ⊳ 🛃 s Ctrl+V Ē. Paste ⊳ 🔁 CM 🗙 Delete Delete 4 🚘 Doc Source . 🖹 r Move... 4 🔁 Ld Rename... F2 🖳 g 🛯 🔁 Peri 🚵 Import... 🖻 🗟 🧧 🗾 Export... Þ 🛃 g Build Project Þ 🖳 🤆 Þ 🖳 g Clean Project 🖻 🖳 g 👔 🛛 Refresh E5 Þ 🗟 🤆 Close Project Þ 🖳 g Close Unrelated Project Þ 🖳 🤆 Build Targets Open Properties Dialog Þ 🛃 g Index Þ 🖳 🤆 **Build Configurations** Þ 🛃 🤆 ► ⊳ 🖳 d Profiling Tools ۲ ▷ Run As ▷ Run As ▷ Run As ▷ Run As ۲ Profile As Þ 🛃 🤆 Þ 🛃 🤤 Restore from Local History... 🖻 🗟 ç 💖 🛛 Run C/C++ Code Analysis Þ 🛃 🤆 Team ۲ ⊳ <u> </u>c Compare With ь 🖻 🗟 🤄 🔽 🛛 Validate Þ 🖳 🤤 Configure ۲ Þ 🛃 g 🔺 🔁 Star Source ► ⊳ <u> s</u> s Properties Alt+Enter a 🔁 Utili

Figure 2-22. Project properties configurations

2.3.1. Target Processor option configuration

"C/C++ Build->Settings->Tool Settings->Target Processor" option configurations:

According to the core of the target chip, select cortex-m3, cortex-m4, cortex-m23 or cortex-m33. In this guide, select cortex-m3.





Figure 2-23. Target Processor configuration

2.3.2. Optimization option configuration

Configure the optimization level in the "C/C++ Build->Settings->Tool Settings->Optimization" option, with options -O0, -O1, -O2, -O3, -Os, -Ofast, -Og.



- • • × Properties for Project $(\neg \neg \neg \neg \neg$ type filter text Settings Resource Builders Configuration: Debug [Active] Manage Configurations... ⊿ C/C++ Build **Build Variables** Environment 🛞 Tool Settings 🛞 Toolchains 📕 Devices 🏮 Container Settings 🎤 Build Steps 🙅 B 🔹 🖻 Logging Settings 🖄 Target Processor Optimization Level Optimize size (-Os) Lool Chain Editor 🖄 Optimization ☑ Message length (-fmessage-length=0) ▷ C/C++ General 🖄 Warnings ✓ 'char' is signed (-fsigned-char) Linux Tools Path 🖄 Debugging ⊳ MCU Function sections (-ffunction-sections) 4 🛞 GNU Arm Cross Assembler Project Natures Data sections (-fdata-sections) Preprocessor **Project References** No common unitialized (-fno-common) 🖄 Includes Run/Debua Settinas Do not inline functions (-fno-inline-functions) 🖄 Warnings Task Repository 🖄 Miscellaneous Assume freestanding environment (-ffreestandin Task Tags 4 🛞 GNU Arm Cross C Compiler 🔲 Disable builtin (-fno-builtin) Validation Preprocessor Single precision constants (-fsingle-precision-co WikiText 🖄 Includes Position independent code (-fPIC) 🖄 Optimization Link-time optimizer (-flto) A Warnings Disable loop invariant move (-fno-move-loop-in 🖄 Miscellaneous Other optimization flags A 🛞 GNU Arm Cross C Linker 🖄 General 🖄 Libraries Miscellaneous a 🛞 GNU Arm Cross Create Flash Image 🖄 General 4 🛞 GNU Arm Cross Print Size 🖄 General ? Apply and Close Cancel

Figure 2-24. Optimization configuration

2.3.3. **GNU Arm Cross C Compiler configuration**

Configure Cross C compilation options in the "C/C++ Build->Settings->Tool Settings->GNU Arm Cross C Compiler" option.

In this guide, add USE_STDPERIPH_DRIVER and GD32F10X_CL pre-compiled macros in the "Preprocessor->Defined symbols' option.

8







Add the header file paths required by the project in the "includes->Include paths" option. Add in this guide:

"\${ProjDirPath}/../../Firmware/CMSIS/GD/GD32F10x/Include"

"\${ProjDirPath}/../../Firmware/CMSIS"

"\${ProjDirPath}/.././Firmware/GD32F10x_standard_peripheral/Include"

"\${ProjDirPath}/../../Template"

"\${ProjDirPath}/../../Utilities"

Note: The header file path added in this guide is a relative path. User can also add the absolute path directly here.



Figure 2-26. GNU Arm Cross C Compiler -> Includes configuration



2.3.4. **GNU Arm Cross C Linker configuration**

Configure Cross C link options in "C/C++ Build->Settings->Tool Settings->GNU Arm Cross C Linker".

Add in the "General ->Script files" option:

"\${ProjDirPath}/../../Firmware/CMSIS/GD/GD32F10x/Source/GCC/gd32f10x_flash.ld"

The linker script is responsible for telling the linker how to configure memory for the compiled executable file. The ld script used should conform to the FLASH and SRAM size of the target chip and the memory configuration required by the customer.

Note: The ld file path added in this guide is a relative path. User can also add the absolute path directly here.



Properties for Project type filter text (- - -) Settings -Resource Configuration: Debug [Active] Manage Configurations... Builders ⊿ C/C++ Build Build Variables 🛞 Tool Settings 🛞 Toolchains 🔳 Devices 🏮 Container Settings 🎤 Build Steps 😤 Build Artifact 🗟 Binary Parsers 🔹 🗠 Environment Target Processor Logging Script files (-T) 🗟 🔊 🗟 衍 👌 Settings Tool Chain Editor Optimization Warnings ⊳ C/C++ General Debugging Linux Tools Path a 🛞 GNU Arm Cross Assembler ▷ MCU Preprocessor Project Natures 🖄 Includes Project References 🖉 Warnings Run/Debug Settings Miscellaneous Task Repository 4 🛞 GNU Arm Cross C Compiler Task Tags Preprocessor ▷ Validation 🖄 Includes WikiText 🖄 Optimization 🖄 Warnings Aliscellaneous 4 🛞 GNU Arm Cross C Linker 🐸 General 🖄 Libraries Miscellaneous a 🛞 GNU Arm Cross Create Flash Image 🖄 General a 🛞 GNU Arm Cross Print Size 🖄 General • • ? Apply and Close Cancel

Figure 2-27. GNU Arm Cross C Linker -> General configuration

In the "Miscellaneous" option, check "Use newlib-nano" and "Do not use syscalls". (The code size can be optimized)







2.3.5. Build Steps configuration-generate bin file

In "C/C++ Build->Settings-> Build Steps", user can add commands to generate bin/hex files.

Add in this guide:

arm-none-eabi-objcopy -O binary "Project.elf" "Project.bin"; arm-none-eabi-objdump -D "Project.elf"> "Project.dump"



Figure 2-29. Build Steps configuration

Properties for Project	
type filter text	Settings 🗢 🕆 🗄
 kesource Resource C/C++ Build Build Variables Environment Location Settings Tool Chain Editor C/C++ General Linux Tools Path MCU Project Natures Project References Run/Debug Settings Task Repository Task Rapository Task Tags Validation WikiText 	Settings Image: Configuration: Configuration: Debug [Active] Image: Build Artifact Binary Parsers Image: Build Artifact Binary Parsers Image: Pre-build steps Image: Pre-build steps Command: Image: Pre-build steps Image: Post-build steps Image: Project.elf Image: Post-build steps Image: Project.elf Image: Project.elf Project.el
?	Apply and Close Cancel

2.4. Build project

Select "Project->Build Project" to compile the current project.

Note: "Build Project" is to compile the current project, and "Build All" is to compile all the projects in the current workspace.



Figure 2-30. Build project

Note: User need to save the current project before compiling each time, otherwise the compiling is the last project. After modification, in order to ensure the correctness, please



clean the project first and then build.

After compiling, it can be seen that the corresponding elf, hex and bin files have been generated.



Figure 2-31. Build ARM project completed

2.5. Use J-Link to download and debug the project

2.5.1. Debug configuration interface

In the menu bar, click "Run->Debug Configurations" to enter the Debug configuration interface.



Figure 2-32. Enter Debug Configuratios interface

eclipse-workspace - Project/Applicat	ion/main.c - Eclipse IL)E		
File Edit Source Refactor Naviga	te Search Project	Run	Window Help	
📑 🕶 🔚 🐚 😻 🕶 🗞 🕶 🔜 📃	轴 🔌 🕹 💣 🔻	Q	Run Last Launched	Ctrl+F11
		检	Debug Last Launched	F11
Project Explorer 🛛 🗖 🗖	i main.c ⊠	8	Profile Last Launched	
🖻 🔄 🏹 🕴	49⊖ int main(voi		Profile History	+
a 🚰 Project (in Eclipse_project)	50 { 51 systick		Profile As	+
⊳ 🖑 Binaries	52 gd_eval		Profile Configurations	
⊳ 🗊 Includes	53 gd_eval	-		
D 🔂 Application	54 gd_eval	-	Run History	•
🛛 🔁 CMSIS	56		Run As	+
🗅 📂 Debug	57 while(1)		Run Configurations	
🛛 🚰 Doc	58 /* 1 59 gd e		Debug History	
🛛 🔁 Ld	60 gd_e	-bla	Dobug Ac	
🗅 🚘 Peripherals	61 dela	r -	Debud As	r
🖻 🚰 Startup	62 /* t		Debug Configurations	
🛛 🔁 Utilities	Galarian (1997) (19		Breakpoint Types	•
	🖹 Problems 🧔 Ta	۰	Toggle Breakpoint	Ctrl+Shift+B
	CDT Build Console [F	۰	Toggle Line Breakpoint	
		65	Toggle Watchpoint	
	makeno-print-d	0	Toggle Method Breakpoint	
	arm-none-eabi-obj	jo.	Skip All Breakpoints	Ctrl+Alt+B
	Froject.dump	520	Remove All Breakpoints	
	Invoking: GNU Arm	-26	nonovo na proakpointa	
	arm-none-eabi-obj	9	External Tools	•

Use J-Link GDBServerCL as the GDB Server, and use the GDB tool in the GCC tool chain as the GDB Client.

Double-click GDB SEGGER J-Link Debugging to create a new set of J-Link configuration options.

2.5.2. Main tab

Figure 2-33. GDB SEGGER J-Link Debugging-Main tab

Debug Configurations	-				_			X
Create, manage, and run configurati	ons						Ŕ	Š.
	Name: Project Debug							
type filter text	📄 Main 隊 Debugger	🕨 Startup 🗄	🖉 Source 🔲 Comr	mon 🖁	SVD Path			
C/C++ Application C/C++ Attach to Application	Project:							_ îl
C/C++ Container Launcher	Project	_					Browse	
C/C++ Postmortem Debugge	C/C++ Application:							
C/C++ Remote Application	Debug\Project.elf							
GDB Hardware Debugging					Variables	Search Project	Browse	
C GDB OpenOCD Debugging	Build (if required) bef	ore launching						
C GDB PyOCD Debugging		- i - i						
GDB QEMU Debugging	Build Configuration:	Select Automa	tically				•	
GDB SEGGER J-Link Debuggir	© Enable auto build			🔘 Disa	able auto build			
Project Debug	Use workspace set	ings		<u>Config</u>	ure Workspace	Settings		-
						Revert	Apply	
Filter matched 13 of 13 items								
?						Debug	Clo	se



In the "Main" tab, select the current project, usually the elf file under the current project will be added automatically. If not, user can click "Browse" to add the elf file manually.

Note: If user have compiled multiple models before, user need to select the corresponding executable elf file. For convenience, user can also create a new set of "Debug configuration" for each chip model.

2.5.3. Debugger tab

In the "Debugger" tab, fill in the device name of the target chip model, which is GD32F107VC in this guide.

If the J-Link path has been configured correctly when setting up the Eclipse environment, it will be recognized automatically here. If user have not configured it correctly before, user can also select the absolute path of J-Link GDBServerCL in the "Executable path" column.

Note: The chip model filled in "Device name" column must be supported by the J-Link driver which is selected here.

Debug Configurations		x
Create, manage, and run configuration	1	5
ype filter text C C/C++ Application C C/C++ Attach to Application	me: Project Debug Main 梦 Debugger ▶ Startup ☞ Source □ Common 문 SVD Path -Link GDB Server Setup ♥ Start the J-Link GDB server locally □ Connect to running target	
C C/C++ Container Launcher C C/C++ Postmortem Debugge C C/C++ Remote Application Cii C/C++ Unit	Executable path: \$(jlink_path)/\$(jlink_gdbserver) Browse, Variables Actual executable: D:\Program Files\SEGGER\/Link/JLinkGDBServerCLexe (to change it use the global or workspace preterences pages or the project properties page)	E
C GDB Hardware Debugging GDB OpenOCD Debugging GDB PyOCD Debugging	Device name: GD32F107VC Supported device names Endianness: Little Big	
© GDB QEMU Debugging © GDB SEGGER J-Link Debuggir	Connection: USB IP (USB serial or IP name/address) Interface: SWD JTAG Interface: Interface: Interface: Interface: Interface: Interface: Interface: Interface: Interface: Interface: Inter	
Launch Group	Initial speed: O Auto Adaptive Fixed 1000 kHz GDB port: 2331	
	SWO port: 2332 Initialize registers on start Telnet port: 2333 Local host only Silent	•
← III → Filter matched 13 of 13 items	Revert Apply	
?	Debug	

Figure 2-34. GDB SEGGER J-Link Debugging-Debugger tab

2.5.4. SVD Path tab

In the "SVD Path" tab, select the SVD file required by the target chip.



Figure 2-35. GDB SEGGER J-Link Debugging-SVD Path tab

Debug Configurations	Autochany Mape III	
Create, manage, and run configurati	ons	Ť.
♥ ● ▼ type filter text € C/C++ Application € C/C++ Attach to Application € C/C++ Attach to Application € C/C++ Postmortem Debugge € C/C++ Nemote Application € C/C++ Unit € GDB Hardware Debugging € GDB OpenOCD Debugging € GDB QEMU Debugging € GDB QEGGER J-Link Debuggirg € Project Debug € Launch Group	Name: Project Debug Main 参 Debugger ▶ Startup Source □ Common So SVD Path SVD file (used by the peripheral registers viewer; may be .svd.zip) File path: F:\eclipse-workspace\GD32F10x_Firmware_Library_V2.2.1\GD32F10x_CL.svd	Browse Variables
 ✓ IIII → Filter matched 13 of 13 items 		Revert Apply
0		Debug Close

2.6. Use GD-Link to download and debug the project

2.6.1. Debug configuration interface

In the menu bar, click "Run->Debug Configurations" to enter the Debug configuration interface.



Figure 2-36. Enter Debug Configuratios interface

Use OpenOCD as the GDB Server, and use the GDB tool in the GCC tool chain as the GDB



Client.

Double-click GDB OpenOCD Debugging to create a new set of OpenOCD configuration options.

2.6.2. Main tab

Figure 2-37. GDB OpenOCD Debugging-Main tab

Debug Configurations		
Create, manage, and run configuration	15	TOT -
Image: Second Secon	Name: Project Debug (1) Main	Durce Common Commo Common Common Comm
?		Debug Close

In the "Main" tab, select the current project, usually the elf file under the current project will be added automatically. If not, user can click "Browse" to add the elf file manually.

Note: If user have compiled multiple models before, user need to select the corresponding executable elf file. For convenience, user can also create a new set of "Debug configuration" for each chip model.

2.6.3. Debugger tab

If the OpenOCD path has been configured correctly when setting up the Eclipse environment, it will be recognized automatically here. If user have not configured it correctly before, user can also select the absolute path of OpenOCD in the "Executable path" column.

In the "Config options" column, fill in the cfg file used. In this guide:

-f \${eclipse_home}\eclipse_toolchain\OpenOCD\scripts\target\openocd_gdlink_gd32f10x.cfg

The cfg file of OpenOCD provides information such as debugger, debugging protocol, target chip identification and target chip programming algorithm selection.



Figure 2-38. GDB OpenOCD Debugging-Debugger tab

reate, manage, and run configuration	ns 🔅
Ye is in X in Y → type filter text C (/C++ Application C (/C++ Attach to Application C (/C++ Container Launcher C (/C++ Postmortem Debugger C (/C++ Remote Application C (/C++ Unit C GDB Hardware Debugging C GDB PyOCD Debugging C GDB PyOCD Debugging C GDB PyOCD Debugging C GDB StGGER J-Link Debugging C Project Debug C Project RISCV Debug C Project Debug C Launch Group	Name: Project Debug (1) Main * Debugger Startup * Source Common * SVD Path OpenOCD Setup Start OpenOCD locally Browse Variables Actual executable path: \${openocd_path}/\${openocd_executable} Browse Variables Actual executable: Executable change it use the global or workspace preferences pages or the project properties page) GDB port: 3333 Telnet port: 4444 Config options: -f \${eclipse_toolchain\OpenOCD\scripts\target\openocd_gdlink_gd32f10x.cfg
Filter matched 15 of 16 items	Revert Apply

2.6.4. SVD Path tab

In the "SVD Path" tab, select the SVD file required by the target chip.

Figure 2-39. GDB OpenOCD Debugging-SVD Path tab

 Debug Configurations Create, manage, and run configuration 	s
Ype filter text © C/C++ Application © C/C++ Application © C/C++ Application © C/C++ Container Launcher © C/C++ Remote Application Cü C/C++ Remote Application Cü C/C++ Unit © GDB OpenOCD Debugging © DP opject Debug (1) © GDB PyOCD Debugging C GDB OPENDED Filter matched 15 of 16 items	Name: Project Debug (1) Main 参 Debugger Startup Source SVD file (used by the peripheral registers viewer; may be .svd.zip) File path: F:\eclipse-workspace\GD32F10x_Firmware_Library_V2.2.1\GD32F10x_CL.s Browse Tariables Revert Apply
?	Debug Close

2.7. Debug interface

After the debug configurations is completed, click "Debug" to enter the Debug perspective.



Figure 2-40. Enter Debug perspective -1

Debug Configurations	
Create, manage, and run configur	ations
	Name: Project Debug
C/C++ Application C/C++ Attach to Application C/C++ Container Launchei	Main Startup Source Common Startup Source Common Startup A Source Common Startup A Source Common Startup A Common Startup A Source A Sour
 C/C++ Postmortem Debug C/C++ Remote Application Ciji C/C++ Unit GDB Hardware Debugging GDB OpenOCD Debugging GDB PyOCD Debugging GDB QEMU Debugging GDB QEGER J-Link Debug Project Debug Launch Group 	Load Symbols and Executable Load symbols
< >	Use file: Workspace File System Executable offset (hex):
Filter matched 13 of 13 items	Revert Apply
	Debug Close

Switch to Debug perspective.

Figure 2-41. Enter Debug perspective -2

Con	firm Perspective Switch		
?	This kind of launch is configured to open the Debug perspective when it suspends.		
This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints.			
	Switch to this perspective?		
Remember my decision			
	Switch No		



Figure 2-42. Debug perspective

eclipse-workspa	ace - Project/Application/ma	in.c <mark>- Ec</mark> lipse IDE	100		- C	-		. • X
File Edit Source	e Refactor Navigate Se	arch Project Run V	Window Help					
i 🔿 🕶 🖪 🕼 🛛	a 💷 🐘 🔍 🕩 🗉 🔳	IN 3 3 8 i≯	表 定 🕹 🎄 🕶 () - 💁 - 🍅 📁	🛷 🕶 🚺 🐓 🛉	- *;> =;* <> - =	> - =1	
			0 O A				٩	😰 🖥 🔯
🔓 Р 🖾 🗖 🗖	🏇 Debug 🖾					🙀 itə 🖇 🗖 🗖	(x)= V ⊠ ¥4	- 0
F 🔄 🍸 🕴	▲ C Project Debug [GDB	SEGGER J-Link Debugg	ing]				10 📑 🗆	
⊿ 😂 Project (in E	4 🔐 Project.elf						Name	Туре
⊳ 🖑 Binaries	a 🖗 Thread #1 5700	05 (Suspended : Breakp	point)					21
⊳ 🗊 Includes	🔳 main() at ma	in.c:51 0x80003f4						
🛛 🗁 Applicati	📕 JLinkGDBServerCL	.exe						
🛛 🕞 CMSIS	📕 arm-none-eabi-go	łb						
🛛 🗁 Debug	📕 Semihosting and S	SWV						
👂 🚰 Doc								
🖻 🚰 Ld	14 Variat ma	in function						
👂 🚰 Peripher	44 (brief ma 45 \param[in] no	one				*		
b 🔁 Startup	46 \param[out] no	one						
D Calification	4/ \retval no 48 */	one						
	49⊖ int main(void)							
	50 {	-().						
	52 gd eval led ir	nit(<i>LED2</i>);					< III	4
	53 gd_eval_led_ir	<pre>nit(LED3);</pre>						*
	54 gd_eval_led_ir	nit(<i>LED</i> 4); nit(<i>LED</i> 5):						
	56							
	57 while(1){	lod2 turn off los	45 */					
	59 gd eval le	ed on(LED2);	/ נ			_		
	CO ed avol 10	sa sterreney.					*	
	E o 1 92 1010 p 1 v			C 0 M				
	Project Dobug IGDR SEGGE	P Llink Dobugging]	xecutables and Debug	ger console 🕕 we	mory 🔳 🚜 🤉	(=x 🔛 🛩 🖃 🕻		<u> </u>
	SEGGER J-Link GDB Serve	r V7.54b Command Li	ine Version					
	JLinkARM.dll V7.54b (DL	L compiled Sep 14 2	2021 16:11:46)					=
	Command line: -if swd -	device GD32F107VC -	endian little -spe	ed 1000 -port 233	1 -swoport 2332 -te	lnetport 2333 -vd	-ir -localho	stonly 1 -
	GDB Server start s	ettings						
	GDB Server Listening po	ort: 2331						
	SWO raw output listenin	ng port: 2332						
	Accept remote connection	on: localhost o	only					
	Generate logfile:	off	,					
	Verity download:	on						
	Silent mode:	off						
	Single run mode:	on O me						
	J-Link related se	ettings						
	J-Link Host interface:	USB						
	J-Link script: J-Link settings file:	none						
	Target related se	ttings						
	Target device:	GD32F107VC						
	Target interface sneed.	1000kH7						-
۰ <u>۱</u>	•	Law 5-11		52 20 2172				•
		Writable	Smart Insert	53:28:21/2	1		: 🔊 🚺	1 🎓 🎽 🛇

2.7.1. Toolbar introduction





2.7.2. Registers view

In the menu bar, select "Window->Show view->Registers" option, open it to view the value of general-purpose registers.

Figure 2-43. Open Registers view





1010 Registers 🖾	to 🕫 🗖 📩 😁
Name	Value
1010 rO	0x8000000 (Hex)
10101 r1	0x8000000 (Hex)
1010 r2	0xa0000000 (Hex)
1010 r3	0x8000175 (Hex)
1010 r 4	0x2000060c (Hex)
10101 r5	0x20000664 (Hex)
¹⁰¹⁰ r6	0x20000664 (Hex)
¹⁰¹⁰ r7	0x484c0440 (Hex)
1010 r8	0x185690 (Hex)
1010 r9	0xc1436 (Hex)
1010 r 10	0x2180634 (Hex)
1010 r 11	0x40902612 (Hex)
¹⁰¹⁰ ₀₁₀₁ r12	0x20000590 (Hex)
¹⁰¹⁰ sp	0x20002000 (Hex)
1111 Ir	0x800022f (Hex)
10101 pc	0x80003f4 (Hex)
¹⁰¹⁰ xpsr	0x61000000 (Hex)
¹⁰¹⁰ msp	0x0 (Hex)
¹⁰¹⁰ psp	0x3b5b260 (Hex)
1000 primask	0x0 (Hex)
¹⁰¹⁰ basepri	0x0 (Hex)
1010 0101 faultmask	0x0 (Hex)
1919 control	0

Figure 2-44. Registers view

2.7.3. Peripherals view

In the menu bar, select "Window->Show view->Peripherals" option, open to view the value of the peripheral registers.



- ccipse-workspa	ecipse-workspace - Project/Application/mainte - Ecipse IDE							
File Edit Source	e Refactor Navigate Search Project Ru	Window Help						
📑 🗝 🖪 🕼 🖡) 💷 🎭 🔍 🕨 💷 🔳 🕺 🌫 🕫	New Window	9	🖌 🕶 🗁 🥖 🔻 🗳	/ 包 - 行 - や			
		Editor	• -					
Px 🗆 🗆	🏇 Debug 🛛 📄 💥 🎼	Appearance	•		£ 🛤 🕒			
🖻 😫 🍸 🖇	🔺 💽 Project Debug [GDB SEGGER J-Link Debu	Show View	•	Breakpoints	Alt+Shift+Q, B			
a 👺 Project (in E	4 🔐 Project.elf	Perspective	۱ 🗉	Console	Alt+Shift+Q, C			
⊳ 🖑 Binaries	🛯 🖗 Thread #1 57005 (Suspended : Brea	Novigation	. 🎋	Debug				
⊳ 🗊 Includes	main() at main.c:51 0x80003f4	Navigation	í 🖳	Debugger Console				
🛛 🗁 Applicati	JLinkGDBServerCL.exe	Preferences	ø	Debug Sources				
CMSIS	arm-none-eabi-gdb	1010 r4		Disassembly				
Debug	Seminosting and SWV	1919 r5	0	Error Log	Alt+Shift+Q, L			
Doc		1010 r 6	0	Executables				
Deriphor		1010 r7	<u>60</u>	Expressions				
Periphen		1010 -0	•	Memory				
District Control Provide Statistics		1010 r10	n	Memory Browser				
		100 r11	-	Modules				
		1010 r12	믕	Outline	Alt+Shift+Q, O			
		1010 sp	E	Peripherals				
		1919 Ir		Problems	Alt+Shift+Q, X			
		1010 pc		Progress				
		1010 xpsr	- B	Project Explorer				
		1010 msp	1010	Registers				



Figure 2-46. Peripherals view

	indonino 🐮 exter	essions 🛋 Modules 🚡	Peripherals 🛛 🏀				
Peripheral	Address	Description		E 8			
	0x40028700	Ethernet: Precision time protocol					
	0xA0000000	External memory contro	ller				
	0x40010400	External interrupt/event controller					
	0x40022000	EMC	controller	ΤC			
E B FWDGT	0x40003000	Free watchdog timer					
	0x40010800	General-purpose I/Os					
	0x40010C00	General-purpose I/Os					
	0x40011000						
	0x40011400						
	0x40011800						
C REAL	0x40011C00						
	0-40012000						
GPIOA: 0x40010800 ᢄ	े 🕂 New Rend	erings	Value				
		Address	value				
		0040010800					
		0~/0010800	0~000000	_			
MD0		0x40010800	0x0000000				
MD0		0x40010800 [1:0]	0x0000000 0x0				
CTL0		0x40010800 [1:0] [3:2] [5:4]	0x00000000 0x0 0x0 0x0				
MD0 CTL0 MD1 CTL1		0x40010800 [1:0] [3:2] [5:4] [7:6]	0x00000000 0x0 0x0 0x0 0x0 0x0				
MD0 CTL0 MD1 CTL1 MD2		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8]	0x00000000 0x0 0x0 0x0 0x0 0x0 0x0 0x0				
MD0 CTL0 MD1 CTL1 MD2 CTL2		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10]	0x00000000 0x0 0x0 0x0 0x0 0x0 0x0 0x0				
MD0 CTL0 MD1 CTL1 MD2 CTL2 MD3		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10] [13:12]	0x00000000 0x0 0x0 0x0 0x0 0x0 0x0 0x0				
MD0 CTL0 MD1 CTL1 CTL2 MD3 CTL3		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10] [13:12] [15:14]	0x0000000 0x0				
MD0 CTL0 MD1 CTL1 MD2 CTL2 MD3 CTL3 MD4		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10] [13:12] [15:14] [17:16]	0x0000000 0x0				
MD0 CTL0 MD1 CTL1 MD2 CTL2 MD3 CTL3 MD4 CTL4		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10] [13:12] [15:14] [17:16] [19:18]	0x00000000 0x0				
MD0 CTL0 MD1 CTL1 CTL2 MD3 CTL3 MD4 CTL4 MD5		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10] [13:12] [15:14] [17:16] [19:18] [21:20]	0x00000000 0x0				
MD0 CTL0 MD1 CTL1 MD2 CTL2 MD3 CTL3 MD4 CTL4 MD5 CTL5		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10] [13:12] [15:14] [17:16] [19:18] [21:20] [23:22]	0x00000000 0x0				
MD0 CTL0 MD1 CTL1 MD2 CTL2 MD3 CTL3 MD4 CTL4 MD5 CTL5 MD6		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10] [13:12] [15:14] [17:16] [19:18] [21:20] [23:22] [25:24]	0x00000000 0x0				
MD0 CTL0 MD1 CTL1 MD2 CTL2 MD3 CTL3 MD4 CTL4 MD5 CTL5 MD6 CTL5		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10] [13:12] [15:14] [17:16] [19:18] [21:20] [23:22] [25:24] [27:26]	0x00000000 0x0				
MD0 CTL0 MD1 CTL1 MD2 CTL2 MD3 CTL3 MD4 CTL4 MD5 CTL5 MD6 CTL6		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10] [13:12] [15:14] [17:16] [19:18] [21:20] [23:22] [25:24] [27:26] [29:28]	0x00000000 0x0				
MD0 CTL0 MD1 CTL1 MD2 CTL2 MD3 CTL3 CTL4 MD5 CTL5 MD6 CTL6 MD7		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10] [13:12] [15:14] [17:16] [19:18] [21:20] [23:22] [25:24] [27:26] [29:28] [31:30]	0x00000000 0x0				
MD0 Image: CTL0 Image: CTL1 Image: CTL2 Image: CTL3 Image: CTL3 Image: CTL4 Image: CTL5 Image: CTL6 Image: CTL7 Image: CTL7		0x40010800 [1:0] [3:2] [5:4] [7:6] [9:8] [11:10] [13:12] [15:14] [17:16] [19:18] [21:20] [23:22] [25:24] [27:26] [29:28] [31:30] 0x40010804	0x00000000 0x0 0x0				

2.7.4. Memory view

In the menu bar, select "Window->Show view->Memory" option, and click the "+" sign above the "Memory" window to open the corresponding memory address.



Figure 2-47. Open Memory view

n	Window	Help						
i	New	Window		9	- 🤌 🗁	🔗 👻 🎿	월 ▼ 월 ▼	*5 0
	Edite	or	•					
	Арр	earance	•		□ (x)= V	ariab 鸟 B	reak 🙀 Ex	pre
k C	Sho	w View	×	•	Breakpoints		Alt+Shift+Q,	В
	Pers	pective	•		Console		Alt+Shift+Q,	С
: 1	Nav	Navigation		苓	Debug			
)3t				R	Debugger Console			
	Pret	erences		e	Debug Sour	ces		
					Disassembly	/		
				0	Error Log		Alt+Shift+Q,	L
				0	Executables			
				6 %ੂ	Expressions			
				0	Memory			
				0	Memory Bro	owser		
				=1	Modules			

Figure 2-48. Memory view

📃 Console <u> Pr</u> oblems	🜔 Executat	oles 🛛 🙀 Debu	gger Console	Memory	x				
Monitors 🛛 🖶 🞇 🎇	0x40022010)x40022010 : 0x40022010 <hex> 🛛 🕂 New Renderings</hex>							
0x40022010	Address	0 - 3	4 - 7	8 - B	C - F				
	40022010	80000000	00000000	00000000	FCFFFF03				
	40022020	FFFFFFF	00000000	00000000	00000000				
	40022030	00000000	00000000	00000000	0000000				
	40022040	00000000	00000000	00000000	0000000				
	40022050	8000000	00000000	00000000	0000000				
	40022060	0000000	0000000	0000000	0000000				

2.7.5. Expressions view

In the menu bar, select "Window->Show View->Expressions" and click the "+" sign in the "Expressions" window to add and view the value of the corresponding variable.



Figure 2-49. Open Expressions view

Run	Window Help							
R 🗗	New Window		9	• 🕈 🗁 🗁 🔸	🍠 🛃 👻 😽 🗸 🏷			
	Editor	►						
	Appearance	+	-	°□ (x)= Variables	💁 Breakpoints 🙀 E			
-Link [Show View	•	•	Breakpoints	Alt+Shift+Q, B			
	Perspective	÷		Console	Alt+Shift+Q, C			
nded :	Navigation	•	*	Debug				
ystick.		G		Debugger Console				
>() at	Preferences		ø	Debug Sources				
C:/1 UX	60004a4		<u></u>	Disassembly				
00042	0		0	Error Log	Alt+Shift+Q, L			
_			0	Executables				
🖒 sys	tick.c 🛛 🐴		s <u>k</u> î	Expressions				
			0	Memory				
			0	Memory Browser				

Figure 2-50. Expressions view

(x)= Variables 💁 Breakpo	oints 🙀 Expressions 🛛	À Modules 🛛 🗆
		🕂 🗙 💥 📑 🖻 🕴
Expression	Туре	Value
(×)⊧ delay	volatile uint32_t	998
🖕 Add new expressio		

Note: Ecplise can only view the value of the variable when the code is not running. It is temporarily unable to update the value of the variable in real time.

2.7.6. Disassembly view

Select the "Instruction Stepping Mode button" in the debug toolbar to open the disassembly window.

Figure 2-51. Open Disassembly view



In the disassembly window, breakpoints can be enabled, assembly instructions can be executed in single step, etc.



Figure 2-52. Disassembly view

	,			
eclipse-workspace -	Project/Application/main.c - Eclipse IDE			
File Edit Source R	efactor Navigate Search Project Run Window H	lelp		
📑 🗝 🖬 👘 🗎 🖬	2 🕸 🗴 🖡 🗉 🔳 🕅 🧏 🤕 🖉 🖉 =	b 🕸 • O • 💁 • 1	😂 🕼 🛷 🕶	♪ 월 ▼ 得 ▼ ♥ ♥ ♥ ♥ ♥ ▼ ฮ
				٩
눱 Pr 🛛 🗖 🗖	🏘 Debug 🛛	🖻 💥 🖬 🕴 🗖 🗖	(x)= Varia	🗣 Break 🙀 Expre 🛋 Modul 🎬 Disass
🖻 🔄 🏹 🕴	Project Debug [GDB SEGGER J-Link Debugging]			Enter location here 🛛 👻 👔 🛐 🔯
P 🕁 Project (in Eclips			080003f4: 51 080003f6: 52 080003fa: 080003fc: 53 08000400:	<pre>main: push {r3, lr} systick_config(); bl</pre>
	🖻 main.c 🛛 🖻 systick.c 💽 main() at ma 🎽		08000402: 54	<pre>bl 0x8000190 <gd_eval_led_init> gd_eval_led_init(LED4);</gd_eval_led_init></pre>
	<pre>51 systick_config(); 52 gd_eval_led_init(LED2); 53 gd_eval_led_init(LED3); 54 gd_eval_led_init(LED4); 55 gd_eval_led_init(LED5); 56 57 while(1){ 58 /* turn on led2, turn off led5 */ 59 gd_eval_led_on(LED2);</pre>	•	08000406: 08000408: 55 0800040c: 0800040e: 59 08000412: 08000414: 60	<pre>movs r0, #2 bl 0x8000100 <gd_eval_led_init> gd_eval_led_init(LED5); movs r0, #3 bl 0x8000100 <gd_eval_led_init> gd_eval_led_on(LED2); movs r0, #0 bl 0x80001c8 <gd_eval_led_on> gd_eval_led_off(LED5); </gd_eval_led_on></gd_eval_led_init></gd_eval_led_init></pre>

2.7.7. Exit the Debug perspective

Click the "Stop debugging" button, and then click "C/C++" to enter the project perspective.

Figure 2-53. Exit the Debug perspective

eclipse-workspace -	Project/Application/systick.c - Eclipse IDE					
File Edit Source R	efactor Navigate Search Project Run Wi	ndow Help				
🛛 🛨 🖃 🕼 🗋 🖬	2 🏪 🔌 🕪 💷 📕 🕺 🌫 🖘 .ete 🖬 🗏	5 🗷 🕹 🕸 🕶 🔘 🕶 🎙	-	😂 🗀 🛷 🕶 🌛 🖢 👻	취 🔹 🏷 ᅷ 🔶 🔹 🖒	- 🛃 👝
	U					🔍 🗄 🖬 🔯
陷 Pr 🛛 🗖 🗖	🎋 Debug 🛛	🖻 💥 it 🗧		^{(x)=} Variables [©] _© Breakp	ooints 🙀 Expressions 🛛	🛋 Modules 👘 🗖
E 🕏 🍸 🕴	▲ 💽 Project Debug [GDB SEGGER J-Link Debu	igging]	-		ž. 🎫 🖻	💠 🗶 💥 📑 🖻 🕴
🛛 🏕 🎏 Project (in Eclips	4 證 Project.elf			Expression	Туре	Value
b 🎇 Binaries	🏿 🕐 Thread #1 57005 (Suspended : Brea	akpoint)		(×)⊧ delay	volatile uint32_t	998
⊳ 🔊 Includes	delay_decrement() at systick.c:83	0x80004b0	=	👍 Add new expressi	o	
Application	signal handler called>() at 0xfff	HH9				
⊳ 🛃 gd32f10x_i	delay_1ms() at systick.c:71 0x800	04a4				
⊳ 🛃 main.c	main() at main.c:b1 0x8000426					
⊳ 🛃 systick.c	JLinkGDBServerCL.exe		*			
D CMSIS	🖻 main.c 💽 main() at ma 🖻 systick	.c 🛛 🐂 👘				
Debug	78 \param[out] none		*			
	80 */					
P Perinherals	81⊖ void delay_decrement(void)					
> Ga Startup	82 { 2083 if (011 = delay){					
▷ (Utilities	84 delay;					
	85 }					
	87		Ŧ			
	•	•				
	Console 🛛 🕄 Problems 🕥 Executables	R Debugger Console	/lemo	orv 🔳 🕷	×	
	Project Debug [GDB SEGGER J-Link Debugging]					



3. Import an existing project

In addition to new projects, user can also import existing Eclipse projects directly. In the menu bar, click "File->Import", select "General->Exisiting Projects into Workspace" to import the existing project, and click "Next".



E Import	
Select Create new projects from an archive file or directory.	Ľ
Select an import wizard:	
type filter text	×
 General Archive File Existing Projects into Workspace File System Preferences Projects from Folder or Archive C/C++ Git Finstall Oomph RPM RPM Run/Debug Teelee 	
? < Back Next > Finish	Cancel

Select the path of an existing project file, Eclipse will recognize all the projects under this path. Select the corresponding project, and click "Finish" to import the existing project.



Figure 3-2. Import an existing project - 2

Import	- • ×
Import Projects Select a directory to search for existing Eclipse projects.	
Select root directory: F:\eclipse-workspace\GD32F10x_Firmware_Library_V2.2.1\Template Select archive file: Projects:	Browse
Project (F:\eclipse-workspace\GD32F10x_Firmware_Library_V2.2.1\Template\Eclipse_proje	Select All Deselect All
Options Search for nested projects Copy projects into workspace Close newly imported projects upon completion Hide projects that already exist in the workspace	
Working sets Add project to working sets Working sets:	New Select
? < Back Next > Finish	Cancel



4. Debug in RAM

Step 1: Modify the linker script, for example, as shown in *Figure 4-1. Ld file memory map when debugging in RAM*.

Figure 4-1. Ld file memory map when debugging in RAM

/* memory map */ MEMORY {		
FLASH (rx)	: ORIGIN = 0x20000000,	LENGTH = 8K
RAM (vrw)	: ORIGIN = 0x20002000.	LENGTH = 8K

Step 2: Relocate the interrupt vector table to SRAM. Recompile the project after completing steps 1 and 2.



45			\param[in] none	
46			\param[out] none	
47			\retval none	
48	*/			
49⊝	in	t	main(void)	
50	{	-		
51	Ľ		<pre>nvic_vector_table_set(NVIC_VECTTAB_RAM, 0);</pre>	
52			<pre>systick_config();</pre>	
53			<pre>gd_eval_led_init(LED2);</pre>	
54			<pre>gd_eval_led_init(LED3);</pre>	
55			<pre>gd eval led init(LED4);</pre>	
56			<pre>gd_eval_led_init(LED5);</pre>	
57				
5.2			while(1){	

Step 3: In the "Debug Configurations->Startup" option, check "RAM application".



Figure 4-3. Debug configurations when debugging in RAM

reate, manage, and run configu	rations
Image: Second	Name: Project Debug Imain 参 Debugger Startup Source Common and S Initialization Commands Initialization Commands Initial Reset and Halt Type: Low speed: 1000 KHz JTAG/SWD Speed: Initial Reset and Halt Type: Low speed: 1000 KHz Image: SWD Speed: Image: Switch and the set of the s
C Project Debug	Load Symbols and Executable Load symbols
Iter matched 13 of 13 items	Runtime Options RAM application (reload after each reset/restart) Run/Restart Commands

Step 4: Enter the Debug perspective when debugging in RAM, as shown in the figure below.





eclipse-workspace	- Project/Application/main.c - Eclipse IDE				
File Edit Source	Refactor Navigate Search Project Run Window Help				
: 🖻 🗕 🖪 🖪 :	🗉 🐏 🔍 🗈 🗉 🖷 🕺 🧟 👁 . R 🖬 🗟 🗷 🐇 🏕 🗸 🔾	- Q	+ i 🍅 👝 🛷	/ - : 씨: 님 - 웹 - や 라 수 - 라 - ㅋ	
				0	
🛅 Pr 🛛 🗖 🗖	🎋 Debug 🔀 📃 👘 🕴		(x)= Varia	🖲 Brea 👾 Expr 🛋 Modu 🎬 Disas 🛛	
	C Project Debug [GDB SEGGER J-Link Debugging]			Enter location here 🔹 🔹 🟠 😫	🗂 🖻 🖇
4 📂 Project (in Eclip	A 📸 Project.elf		6	main:	^
⊳ 👯 Binaries	Thread #1 57005 (Suspended : Breakpoint)		✓ 20000514: 51	systick config():	
⊳ 🗊 Includes	main() at main.c:51 0x200003f4		200003f6:	<pre>bl 0x20000464 <systick_config></systick_config></pre>	
Application	JLinkGDBServerCL.exe		52	<pre>gd_eval_led_init(LED2);</pre>	
CMSIS	arm-none-eabi-gdb		200003Ta: 200003fc:	bl 0x20000190 <gd eval="" init="" led=""></gd>	
🛛 🗁 Debug	Semihosting and SWV	53	gd_eval_led_init(LED3);	=	
Doc 🕞 🕞			20000400:	movs r0, #1	
⊿ 🚰 Ld			54	<pre>gd eval led init(LED4);</pre>	
🖳 gd32f10x	© 0x80003fa		20000406:	movs r0, #2	
Peripherals	<pre>\$51 systick_config(); 52 ad aval lad init/(502);</pre>	^	20000408:	<pre>bl 0x20000190 <gd_eval_led_init> gd_eval_led_init(LEDE);</gd_eval_led_init></pre>	
Eartup	53 gd eval led init(<i>LED3</i>);		2000040c:	movs r0, #3	
Dilities	54 gd_eval_led_init(LED4);		2000040e:	<pre>bl 0x20000190 <gd_eval_led_init></gd_eval_led_init></pre>	
	<pre>55 gd_eval_led_init(LED5); 56</pre>		59	gd_eval_led_on(LED2);	
	57 while(1){		20000412:	bl 0x200001c8 <gd_eval_led_on></gd_eval_led_on>	
	58 /* turn on led2, turn off led5 */		60	<pre>gd_eval_led_off(LED5);</pre>	
	59 gd_eval_led_on(LED2); 60 gd_eval_led_off(LED5);	E	20000418: 2000041a:	movs r0, #3 bl 0x200001e0 <gd eval="" led="" off=""></gd>	
	61 delay_1ms(1000);		61	delay_1ms(1000);	
	62 /* turn on led3, turn off led2 */	-	2000041e:	mov.w r0, #1000 ; 0x3e8	-
		F.		٠	Þ
	E Console 🕺 🕄 Problems 🕡 Executables 🗟 Debugger Console	1 Me	mory		
	Project Debug (GDB SEGGER J-Link Debugging)	0			
	R0 = 00004813, R1 = 00000000, R2 = E000E000, R3 = E000ED00				
	R4 = 2000060C, R5 = 20000664, R6 = 20000664, R7 = 48480440				
	R8 = 00185690, R9 = 000C1436, R10= 02180634, R11= 40902612				



5. **Printing with printf**

5.1. Use steps

Step 1: Add the syscall.c file, and add the following _write function definition to the file.

```
int _write(int file, char *ptr, int len)
{
    int Dataldx;
    for (Dataldx = 0; Dataldx < len; Dataldx++)
        {
            ___io_putchar( *ptr++ );
        }
    return len;
}</pre>
```

Step 2: Redirect usart to the __io_putchar function.

```
int __io_putchar(int ch)
{
    usart_data_transmit(EVAL_COM0, (uint8_t) ch );
    while(RESET == usart_flag_get(EVAL_COM0, USART_FLAG_TBE)){
    };
    return ch;
}
```

Step 3: Use the printf function to print normally.

printf("Running led test!\r\n");

5.2. Print floating point data configuration

Print floating point data configuration:

check the "-u _prinft_float" option in the project "Properties->C/C++ Build->Settings->Tool Settings->GNU Arm Cross C Linker->Miscellaneous" option.



Figure 5-1. Print floating point data configuration



Note: 1. When using printf function, user need to add "\r\n" at the end of the printed content, for example, printf("Running led test!\r\n"). 2. Using printf function in GCC will greatly increase the size of the code. If it is an occasion that requires a high codesized size, printf function is not recommended.



6. Revision history

Table 6-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Nov.30, 2021



Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the "Product"), is owned by the Company under the intellectual property laws and treaties of the People's Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it's suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it's suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.

© 2021 GigaDevice – All rights reserved