GigaDevice Semiconductor Inc.

GD-Link V2 Adapter

User Guide

Revision 1.1

(Jan. 2024)



Table of Contents

Table of	^c Contents	2
List of F	igures	3
List of T	ables	5
1. Int	roduction	6
2. Ha	Irdware introduction	7
2.1. F	Pin definitions and wiring methods	7
	Button, LEDs and Buzzer	
	Dutput voltage	
	ftware features	
	-irmware updates	
	Programming function	
3.2.1.	IDE programming	14
3.2.2.	GD-Link Programming	20
3.2.3.	Offline programming	22
3.2.4.	Machine singal triggered programming	24
3.2.5.	Virtual USB disk drag and drop programming	25
3.3. C	Debug function	28
3.3.1.	SWD /JTAG debugging	28
3.3.2.	SWO function	29
3.4. V	/irtual serial port printing	33
4. Re	evision history	35
	₽	



List of Figures

Figure 2-1. GD-Link V2 pinout diagram	. 7
Figure 2-2. SWD interface connection diagram	. 8
Figure 2-3. JTAG interface connection diagram	. 8
Figure 2-4. SWD + SWO interface connection diagram	. 9
Figure 2-5. Serial interface connection diagram	. 9
Figure 2-6. GD-Link V2 adapter hardware	10
Figure 2-7. GD-Link V2 output voltage select	11
Figure 3-1. GD-Link V2 firmware update step 1	12
Figure 3-2. GD-Link V2 firmware update step 2	13
Figure 3-3. GD-Link V2 firmware update step 3	13
Figure 3-4. GD-Link V2 firmware update step 4	14
Figure 3-5. KEIL debug configuration	15
Figure 3-6. KEIL utilities configuration	15
Figure 3-7. KEIL Download Icon	16
Figure 3-8. Build output window - programming successful	16
Figure 3-9. IAR debugger configuration	16
Figure 3-10. IAR CMSIS DAP configuration	17
Figure 3-11. IAR download button	18
Figure 3-12. IAR download progress bar	18
Figure 3-13. Access the "Debug Configurations" interface	
Figure 3-14. Configure the "Eclipse Debug" tab	19
Figure 3-15. Enter the debugging interface in Eclipse	20
Figure 3-16. GD-Link programmer programming options configuration	20
Figure 3-17. Connecting the target chip in GD-Link Programmer	21
Figure 3-18. GD-Link Programmer burns target chip	
Figure 3-19. GD-Link V2 offline download parameter configuration	22
Figure 3-20. GD-Link V2 offline download file update configuration	23
Figure 3-21. Offline download file updated to GD-Link V2	23
Figure 3-22. Simultaneously adding BOOT+APP offline download file update to GD-Link V2.	23
Figure 3-23. Offline download configuration option byte feature	24
Figure 3-24. Machine signal programming pin distribution schematic diagram	25
Figure 3-25. USB mass storage device	26
Figure 3-26. Virtual USB drive	26
Figure 3-27. KEIL debugging interface	28
Figure 3-28. IAR debugging interface	29
Figure 3-29. SWO configuration step 1 in KEIL	30
Figure 3-30. SWO configuration step 2 in KEIL	30
Figure 3-31. Debug (printf) viewer window in KEIL	
Figure 3-32. Logical Analyzer window in KEIL	33
Figure 3-33. USB serial device	34



Fig	ure 3-34.	USB	virtual	serial	printing		34



List of Tables

Table 2-1. GD-Link V2 pin function definitions	7
Table 2-2. Working status of GD-Link V2	10
Table 3-1. Machine signal programming pin function definition	25
Table 3-2. CONFIG.TXT file content	26
Table 3-3. Drag-and-Drop programming configuration parameter definitions	27
Table 3-4. Trace mode enable	31
Table 3-5. Printf retarget	31
Table 4-1. Revision history	35



1. Introduction

GD-Link V2 is a rich-featured, easy-to-use, and portable debugging and programming tool developed by GigaDevice for GD32 series MCU, which has the following characteristics:

- USB2.0 high-speed interface
- Provide 5V or 3.3V power supply for the target chip
- Support firmware update through the GD-Link Programmer software
- Support SWD / JTAG debugging and programming interface
- Support GD32 ARM / RISC-V core full series of chips
- Support KEIL / IAR / Eclpise debugging and programming
- Support offline programming
- Support virtual USB disk drag and drop programming
- Support SWO function
- Support virtual serial port printing

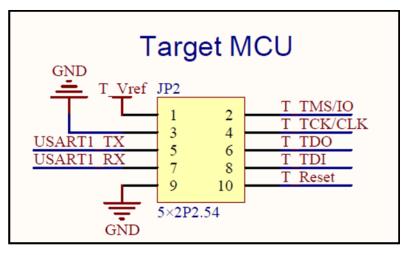


2. Hardware introduction

2.1. Pin definitions and wiring methods

To enable programming, debugging, serial communication, and printing functions, connect the GD-Link V2 pins to the SWD (SWO), JTAG, or USART interface of the target chip using DuPont wires or ribbon cables. The pinout of GD-Link V2 is illustrated in *Figure 2-1. GD-Link V2 pinout diagram*.

Figure 2-1. GD-Link V2 pinout diagram



The functions of each GD-Link V2 pin are described as shown in <u>Table 2-1. GD-Link V2 pin</u> function definitions.

Pin Number	Pin Name	Description		
1	T_Vref	Target chip power supply, providing 3.3V / 5V		
2	T_TMS/IO	JTAG TMS pin / SWD SWDIO pin		
3	GND	Power ground		
4	T_TCK/CLK	JTAG TCK pin / SWD CLK pin		
5	USART1_TX	Serial transmission pin		
6	T_TDO	JTAG TDO pin / SWO pin		
7 USART1_RX		Serial reception pin		
8 T_TDI		JTAG TDI pin		
9 GND 1		Power ground		
10 T_Reset JTAG / SWD target chip reset pin				

The diagram of GD-Link V2 hardware connection to the target chip is illustrated in <u>Figure 2-2</u>. <u>SWD interface connection diagram</u>, <u>Figure 2-3</u>. <u>JTAG interface connection diagram</u>, <u>Figure 2-4</u>. <u>SWD + SWO interface connection diagram</u> and <u>Figure 2-5</u>. <u>Serial interface</u> <u>connection diagram</u>.



Figure 2-2. SWD interface connection diagram

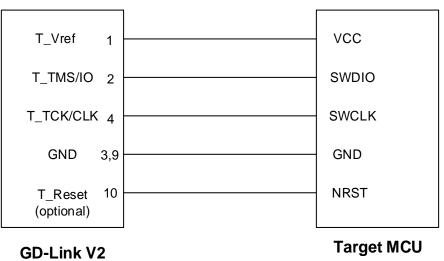
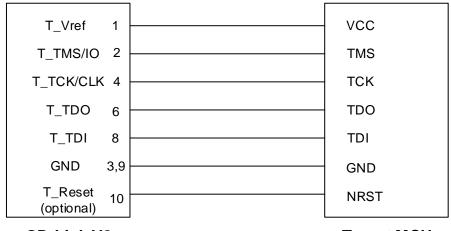


Figure 2-3. JTAG interface connection diagram



GD-Link V2

Target MCU





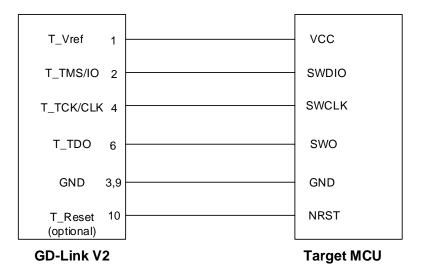
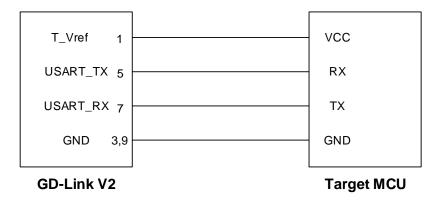


Figure 2-5. Serial interface connection diagram

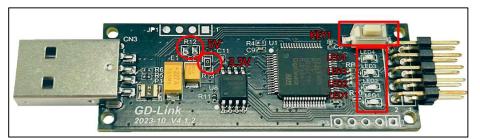


2.2. Button, LEDs and Buzzer

GD-Link V2 features a single button (K1), a buzzer (BZ1) and four LEDs (LED1/2/3/4) as indicators. The physical representation of GD-Link V2 is shown in *Figure 2-6. GD-Link V2 adapter hardware*. The button K1 is used for firmware updates and offline programming. For specific usage instructions, please refer to the firmware update and offline programming section.



Figure 2-6. GD-Link V2 adapter hardware



During offline programming and drag-and-drop programming from a virtual USB disk, when the target chip has been successfully programmed with the desired file, the buzzer will beep, indicating a successful programming status. The on-off and blinking of the LED indicate different working states of GD-Link V2. *Table 2-2. Working status of GD-Link V2* provides a description of the different status of these LEDs which indicate the status of programming and debugging tool.

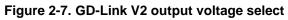
LED	LED status	GD-Link V2 working status		
	alwaya bright	Offline programming or drag-and-drop		
LED1	always bright	programming successful		
LEDI	flashing	performing offline programming or drag-and-		
	liasining	drop programming		
LED2	flashing fast	USB connection successful		
LED2	flashing slow	USB not connected		
LED3	always bright	Firmware update status		
LED4	always bright	Power supply is normal		

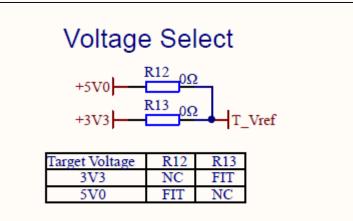
Table 2-2. Working status of GD-Link V2

2.3. Output voltage

The debugger provides 5V and 3.3V output voltages for users to choose. The output voltage can be modified by short-connecting the R12 and R13 resistors in the hardware through the 0Ω resistor. The schematic diagram of the voltage selection is shown in *Figure 2-7. GD-Link V2 output voltage select*. Reference *Figure 2-6. GD-Link V2 adapter hardware*, when the 0Ω resistor is welded at R13, the T_Vref voltage is 3.3V, when the 0Ω resistor is welded at R13, the T_Vref voltage is 5V.









3. Software features

3.1. Firmware updates

GD-Link V2 provides firmware update functionality. Firmware updates are used to:

- Support the latest MCUs released by GD32.
- Fix issues present in the firmware.

GD-Link V2 can be updated using the GD-Link Programmer software. Users can visit the GD32MCU official website to obtain the latest version of the GD-Link Programmer software, unzip it after downloading, and follow these firmware update steps:

- 1. Disconnect GD-Link V2 from the computer's USB port.
- 2. While holding down button K1, plug GD-Link V2 back into the computer's USB port. At this time, LED3 is always on, indicating that the programmer is in firmware upgrade mode.
- 3. Release button K1 and click the "GD-Link" menu in the GD-Link Programmer software. Choose "Update Firmware" to start the firmware update process.
- 4. A progress bar will pop up in the GD-Link Programmer software, indicating the progress of the update. Wait for it to reach 100% and show a successful update message.

Refer to *Figure 3-1. GD-Link V2 firmware update step 1*, *Figure 3-2. GD-Link V2 firmware update step 2* and *Figure 3-3. GD-Link V2 firmware update step 3* for visual guidance on the firmware update process.

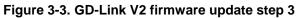
Figure 3-1. GD-Link V2 firmware update step 1





Figure 3-2. GD-Link V2 firmware update step 2

📸 GI	D-Link Progr	rammer 4.	7.14.1907	6					_	
) Target(I)				View(V)	Help(H)				
Output Shor is	ni ti al infor	ration an	d softwar	• inform	ation her	76	GD-Link Programmer Please make sure DO NOT disconnect from PC in upgrade process Confirm to upgrade to Firmware version 17 夏(竹)	-	penties "index of the second s	ink) Unknown SWD Unknown Unknown Unknown Yes Unknown Yes Unknown Unknown Unknown
Ready	H Statu	<u>s</u> /								CAP NUM SCRL -

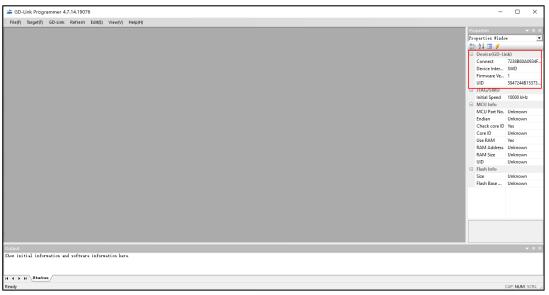


B-Link Programmer 4.7.14.19076	-	
File(F) Target(T) GD-Link Refresh Edit(E) View(V) Help(H)		
	Properties Properties Windo	~ 0 ×
		··· ·
	8E 24 💷 🗲	
	Device(GD-Lin	
	Connect Device Inter	Unknown
	Firmware Ve	
		Unknown
	JTAG/SWD	Unknown
	Initial Speed	10000 kHz
	MCU Info	rever ni la
GD-Link Progress X	MCU Part No.	Unknown
		Unknown
Operation: Upgrading GD-Link firmware	Check core ID	Yes
	Core ID	Unknown
RealTime: Upgrading GD-Link firmware Successfully!	Use RAM	Yes
Time Cost: 5.828 s	RAM Address	Unknown
		Unknown
100%		Unknown
OK	Flash Info	
UN		Unknown
	Flash Base	Unknown
Dufput	_	→ 0 ×
Computer Show initial information and software information here.		
H 4 b H Status		
Ready		CAP NUM SCRL

After the update is completed, user can check the current firmware version number in the properties pane, as shown in *Figure 3-4. GD-Link V2 firmware update step 4*.



Figure 3-4. GD-Link V2 firmware update step 4



Note: During the firmware update process, do not unplug GD-Link V2 from the computer's USB port.

3.2. Programming function

3.2.1. IDE programming

Programming with KEIL (version 5.27 and above):

Connect GD-Link V2 to the target chip according to the hardware connection described in <u>Pin</u> <u>definitions and wiring methods</u> section. Connect the USB interface of GD-Link V2 to the PC, and wait for LED2 to enter rapid blinking mode. Open KEIL software, in the KEIL Debug tab, select "CMSIS-DAP Debugger" or "CMSIS-DAP ARMv8-M Debugger" in the "Debug" option, as shown in <u>Figure 3-5. KEIL debug configuration</u>.



Figure 3-5. KEIL debug configuration

Options for Target 'GD32F10X_ Device Target Output Listing		1 C/C++ Asm Linker Debug Utilities	×			
○ Use Simulator with restricts □ Limit Speed to Real-Time	ions	Settings CMSIS-DAP Debugger	Settings			
CMSIS-DAP Cortex-M Target Drive			×			
CMSIS-DAP - JTAG/SW Adapter	-SW Dev SWDIO	Vice Device Name IDCODE Device Name Ox1BA01477 ARM CoreSight SW-DP	Move			
Firmware Version: 0255		omatic Detection ID CODE: nual Configuration Device Name:	Down			
Add Delete Update AP: [0x00 Debug Connect & Reset Options Download Options Connect: Nomal Reset: Autodetect Image: Cache Options Download Options Image: Connect: Nomal Reset: Autodetect Image: Cache Code Image: Cache Code <td< td=""></td<>						
	(DK Cancel	Help			

In the "Utilities" tab, select "Use Debug Driver" and click the "Setting" button to choose the MCU download algorithm and configure the erase mode and other settings, as shown in *Figure 3-6. KEIL utilities configuration*.

Figure 3-6. KEIL utilities configuration

Options for Target 'GD32F	10X_XD'		1	×
Device Target Output List:	ng User C	/C++ Asm Li	nker Debug Vtilities	
Configure Rash Menu Command © Use Target Driver for Rash A Use Debug Dri		3 Settings	2 ☑ Use Debug Driver ☑ Update Target before Debug	gging
CMSIS-DAP Cortex-M Target D	river Setup			×
Debug Trace Flash Downloa Download Function 4 Control Crase Full Chip Crase Full Chip Crase Sectors C Do not Erase Programming Algorithm	d │ ✓ Program ✓ Verify ✓ Reset and F		lgorithm x20000000 Size: 0x00001000	
Description	Device Size	Device Type	Address Range	
GD32F10x Extra-density FMC	3M	On-chip Flash	08000000H - 082FFFFFH	
	5			
		Start:	Size:	
	Add	Remove		
	OK	Cance	1	Help

Click the "Download" icon in the KEIL menu bar. In the "Build Output" window, the programming progress can be monitored, as shown in *Figure 3-7. KEIL Download Icon* and



Figure 3-8. Build output window - programming successful.

Figure 3-7. KEIL Download Icon

🗋 📬 🛃	🦪 🐰 🖻 🖺	º (º ← ⇒	1 1 1 1 1	医律	i≡ //≞ //§
🔮 🕮 🏙	🐝 🕶 🧱 🚺	GD32F10X_XD	V 🔊 占	Ē. ♦	🐡 💩

Figure 3-8. Build output window - programming successful

```
Full Chip Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 16:06:29
```

Programming with IAR (version 8.50 and above):

Connect GD-Link V2 to the target chip according to the the hardware connection described in <u>Pin definitions and wiring methods</u> section. Connect the USB interface of GD-Link V2 to the PC, and wait for LED2 to enter rapid blinking mode. Open IAR software. In the IAR "Project" menu, choose "Options." In the "Debugger" tab, choose "CMSIS-DAP" as the debugger driver, as shown in <u>Figure 3-9. IAR debugger configuration</u>. In the "Setup" tab, choose the MCU type, download algorithm, and other configurations according to the target chip's requirements, as shown in <u>Figure 3-10. IAR CMSIS DAP configuration</u>.

Options for node "GD32103B_B	EVAL"	×
Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Did totrage	EVAL* Factory Setti Factory Setti Factory Setti Factory Setti]
	OK Cancel	

Figure 3-9. IAR debugger configuration



Figure 3-10. IAR CMSIS DAP configuration

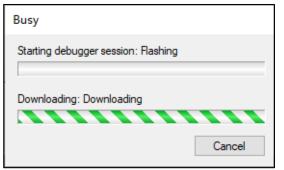
In the menu bar "Project" drop-down option "Download", click "Download active application" and wait for the progress bar to complete the burning, as shown in <u>Figure 3-11. IAR</u> <u>download button</u> and <u>Figure 3-12. IAR download progress bar</u>.



Figure 3-11. IAR download button

Add Files Add Group Import File List Add Project Connection Edit Configurations Remove de "gd32f10x.h" G IN ANY WAY OUT OF THE USE OF
Import File List Add Project Connection Edit Configurations H DAMAGE. Remove de "gd32fl0x.h" G I' ANY WAY OUT OF THE USE OF THE US
Add Project Connection IG IN ANY WAY OUT OF THE USE OF THE US
Edit Configurations H DAMAGE. Remove de "gd32f10x.h" Create New Project de "gd32f103b_eval.h" Add Existing Project de "gd32f103b_eval.h" Options Alt+F7 rief main function aram[in] none aram[out] none aram[out] none febuild All f8 clean f8 Batch build f8
Remove de "gd32f10x.h" Create New Project de "gd32f103b_eval.h" Add Existing Project de "gd32f103b_eval.h" Options Alt+F7 version Control System aram[in] none aram[out] none aram[out] none Compile Ctrl+F7 Rebuild All clean Batch build F8 enable the LEDs clock */ unceriph clock enable/PUI GEDCO
de "gd32f10x.h" Create New Project Add Existing Project Options Alt+F7 version Control System Make F7 compile Chrl+F7 Rebuild All Clean Batch build F8
Create New Project de "gd32f103b_eval.h" Add Existing Project de "systick.h" Options Alt+F7 Version Control System aram[in] none aram[out] none aram[out] none Compile Ctrl+F7 Rebuild All stick_config(); Batch build F8 enable the LEDs clock */ uncertiph clock enable/PUL GEDCO
Add Existing Project Options Alt+F7 Version Control System Make F7 Compile Ctrl+F7 Rebuild All Clean Batch build F8 enable the LEDs clock */ Unperiph clock enable/PEULGEDCO
Version Control System rief main function Version Control System aram[in] none aram[out] none aram[out] Make F7 etval none Compile Ctrl+F7 in (void) Rebuild All f8 enable the LEDs clock */ Uncertain Cook enable the LEDs clock */
Make F7 etval none etval none for etval none in (void) rebuild All Clean Batch build F8 enable the LEDs clock */ un priph clock enable(PCII GEDCO)
Make F7 etval none Compile Ctrl+F7 in (void) Rebuild All (stick_config(); Batch build F8 enable the LEDs clock */
Rebuild All in (void) Clean stick_config(); Batch build F8 enable the LEDs clock */ up refine clock enable (PCU GEDCO)
Rebuild All Clean Stick_config(); Batch build F8 enable the LEDs clock */
Batch build F8 enable the LEDs clock */
enable the LEDs clock */
C-STAT Static Analysis u_periph_clock_enable (RCU_GPIOC) ;
Stop Build Ctrl+Break configure LEDs GPIO port */ io init(GPIOC, GPIO MODE OUT PP,
Download and Debug Ctrl+D
Debug without Downloading Debug without Downloading io_bit_reset (GPIOC, GPIO_PIN_6
Attach to Running Target
Make & Restart Debugger Ctrl+R Iile(1) { /* turn on LED2 */ }
C Restart Debugger Ctrl+Shift+R mio bit set (GPTOC. GPTO PIN 6)
Download Download active application Alt+
SFR Setup Download file Erase memory
CMSIS-Pack Manager gpio bit set (GPIOC, GPIO PIN 7)
Open Device Description File /* insert 200 ms delay */
Save List of Registers delay_lms (200) ;

Figure 3-12. IAR download progress bar



Programming with Eclipse

Connect GD-Link V2 to the target chip according to the hardware connection described in the <u>Pin definitions and wiring methods</u> section. Connect the USB interface of GD-Link V2 to the PC, and wait for LED2 to enter rapid blinking mode. Open the Eclipse software and click "RUN" menu and select the dropdown option "Debug Configurations..." to enter the "Debugger" tab, as shown in <u>Figure 3-13. Access the "Debug Configurations" interface</u>. Configure the OpenOCD path correctly and fill in the cfg file to be used in the "Config options" section, as demonstrated in <u>Figure 3-14. Configure the "Eclipse Debug" tab</u> in the Eclipse Debug Configuration interface.

After completing the configuration, click the "Apply" button to save the settings. Then, select



the "Debug" button, and when the "Confirm Perspective Switch" window appears, click "YES" to confirm. This will initiate the code download and take to the debugging interface, as illustrated in *Figure 3-15. Enter the debugging interface in Eclipse*.

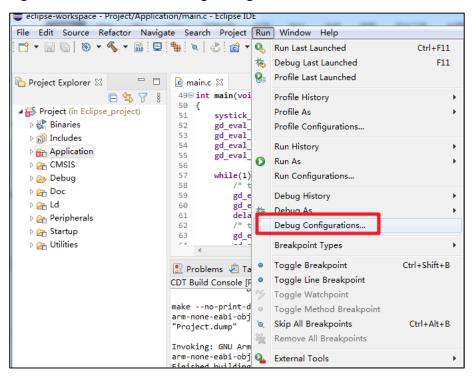
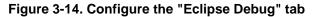


Figure 3-13. Access the "Debug Configurations" interface



Debug Configurations		x
Create, manage, and run configuratior		r
type filter text C/C++ Application C/C++ Attach to Application C/C++ Container Launcher C/C++ Postmortem Debugger C/C++ Remote Application Ci /C++ Unit GDB Hardware Debugging C GDB OpenOCD Debugging C GDB OpenOCD Debugging C GDB QEMU Debugging C GDB SEGGER J-Link Debugging C Project Debug C Project RISCV Debug C Project RISCV Debug Launch Group	Iame: Project Debug (1) Main Image: Debugger Startup OpenOCD Setup Image: Start OpenOCD locally Image: Start OpenOCD locally Image: Start OpenOCD locally Executable path: \$(openocd_path)/\$(openocd_executable) Actual executable: E:\eclipse_toolchain\OpenOCD\bin/openocd_exe Image: Config options: Image: Start OpenOCD locally Image: Config options: -f \$(eclipse_toolchain\OpenOCD\scripts\target\openocd_gdlink_gd32f10x.cfg Image: Config options: -f \$(eclipse_toolchain\OpenOCD\scripts\target\t	
Filter matched 15 of 16 items	Revert Apply	
?	Debug Close	



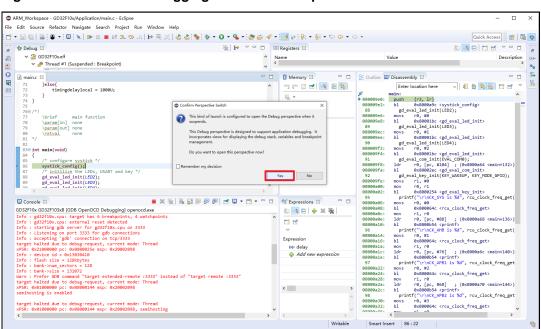
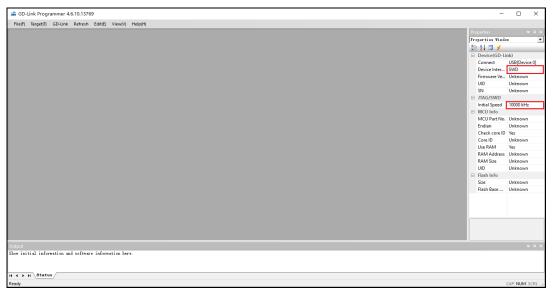


Figure 3-15. Enter the debugging interface in Eclipse

3.2.2. GD-Link Programming

Connect GD-Link V2 to the target chip according to the hardware connection described in Pin definitions and wiring methods section. Connect the USB interface of GD-Link V2 to the PC, and wait for LED2 to enter rapid blinking mode. Open the GD-Link Programmer software and select the JTAG / SWD programming interface and configure the communication speed in the "Properties" window. Refer to Figure 3-16. GD-Link programmer programming options configuration for an illustration of GD-Link Programmer programming options.

Figure 3-16. GD-Link programmer programming options configuration



Click the "Target" dropdown menu and choose the "Connect" option. Check the "Output" window for a message indicating "Connection successful." At the same time, the detailes



information about the connected target chip, including its specific type are listed in the "Properties" window. Refer to *Figure 3-17. Connecting the target chip in GD-Link Programmer* for an illustration of GD-Link Programmer successfully connecting to the target chip.

📸 GD-Link	Programmer 4.6.10.13769		-	\square X
E File(F) Tar	get(T) GD-Link Refresh Edit	(E) View(V) Help(H)		
	Connect	R		
	Disconnect		Properties Windo	· •
	Security	,	🔠 24 🖃 🗲 👘	
			Device(GD-Line)	nk)
	Insecurity			USB[Device 0]
	Configure OptionBytes	F6	Device Inter	
	Mass Erase	F4	Firmware Ve	
	Page Erase	F3		5546D40E93931
				0000002700000
	Check Blank		JTAG/SWD	
	Compare data with opened file			10000 kHz
	Program	F7	MCU Info	GD32L233CCT6
	Continuous Program	2		Little Endian
			Check core ID	
	Read Data	•		0xBF11477
	Run App	P9		Yes
			RAM Address	0x20000000
			RAM Size	32KB
				38 4F 36 36 4B 3
			Flash Info	
				256KB
			Flash Base	0x8000000
Output				
Show initial	information and software inf successful!	Corner on here 3		
Connection s Getting opti A5 5A FF 00	ruccessful! ion bytes successfully! FF OD FF OD FF OD FF OD FF OD	F7 00		
нчэн	Status			
Ready				CAP NUM SCRL

Figure 3-17. Connecting the target chip in GD-Link Programmer

Drag and drop the binary file, "xxx.bin" or the executable file "xxx.hex" into the GD-Link Programmer software. When using the "xxx.bin" file for programming, a dialog will appear on the host computer's software, prompting to enter the starting address for the download. After entering the correct download address, click the "OK" button. Then, select the "Target" dropdown menu and choose the "Program" option. The software will start downloading the program to the target chip. Wait for the progress bar to reach 100%, and a message will confirm the successful download, as shown in *Figure 3-18. GD-Link Programmer burns target chip*.

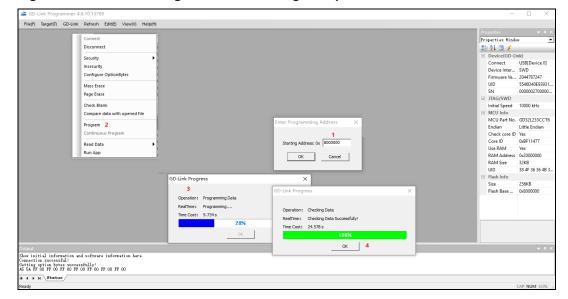


Figure 3-18. GD-Link Programmer burns target chip



3.2.3. Offline programming

Connect the USB interface of GD-Link V2 to the PC, and wait for LED2 to enter rapid blinking mode. Open the GD-Link Programmer software. Click "GD-Link" menu bar and then choose "Configuration" to configure the parameters of offline programming, referring to *Figure 3-19. GD-Link V2 offline download parameter configuration*. The following configurations can be performed using this interface:

- Whether to enable read protection after offline programming completion.
- Erase method selection: full chip erase or page erase.
- Limit the number of offline programming downloads.

Click the "OK" button in the offline programming parameter configuration interface to save the settings. After configuration, in the menu bar, click "GD-Link" and then "Update File" to enter the file update interface. Referring to *Figure 3-20. GD-Link V2 offline download file update configuration*. Select the specific part number of the target MCU, add the xxx.bin file, specify the download address to the target chip, and click the "Update" button. Wait for the progress bar to reach 100% to complete the file update, as shown in *Figure 3-21. Offline download file update file update to GD-Link V2*.

File updating supports one-time burning for BOOT+APP functionality. The user can continue to click the "Add" button to add a second bin file, specify the burning address. The file update allows to add a maximum of 8 bin files. The addition process is illustrated in <u>Figure 3-22</u>. <u>Simultaneously adding BOOT+APP offline download file update to GD-Link V2</u>.

For GD32W515 series MCU, offline programming also supports option byte configuration. When selecting the MCU part number as GD32W515 series MCU in the "GD-Link Update File Configuration" window, click the "Configure OptionBytes" button. In the pop-up window, perform the option byte configuration. After configuration, click the "OK" button to save the relevant settings, as shown in *Figure 3-23. Offline download configuration option byte feature*.

GD-Link Programmer 4.6.10.13769		
File(F) Target(T) GD-Link Refresh Edit(E) View(V) Help	(H)	
Update File		
Configuration 1		
Update Firmware		
	GD-Link Configuration X	
	Offline-Programming Configuration	
	Secure After Programming Erase Option Chip	
	Program Limit Count 501 2	
	Online-Programming Configuration	
	Run After Programming	
	Product SN	
	Write SN SN Address:0x 00000000	
	Next SN: 1 SN Increment; 1	
	OK Cancel	

Figure 3-19. GD-Link V2 offline download parameter configuration



Figure 3-20. GD-Link V2 offline download file update configuration

GD-Link Programmer 4.6.10.13769		
File(F) Target(T) GD-Link Refresh Edit(E) View(V) Help(H) Update File 1 Configuration Update File 2		
Update Firmware	· 编 打开	×
	← → ✓ ↑ _ ≪ 新加巻 (E:) → Test ✓ ひ 搜索"Te	est" p
GD-Link Update File Configuration X	组织 ▼ 新建文件夹	🖽 🔻 🛄 🕜
Part No. : GD32F103RGT6 - 2	G fw32101 ★ ▲ 名称 ▲	修改日期 类型
	💻 此电脑 🔂Template 2	2020/7/31 14:14 文件夹
File Index File Path File Size Address	3D 对象 📃 test_led.bin 2	2021/4/16 9:52 BIN 文件
Enter Programming Address X Starting Address: 0x 9000000 4 OK Cancel Delete Add 3 Cancel Update	 · 很須 · 回片 · 回片 · 文档 · 下載 · 音乐 · 重 · 重 · 重	hex v
	打开	Ŧ(O) 取消

Figure 3-21. Offline download file updated to GD-Link V2

GD-Link Programmer 4.6.10.13769	
File(F) Target(T) GD-Link Refresh Edit(E)	View(V) Help(H)
GD-Link Update File Configuratio	GD-Link Programmer × Continu to do this, please click "Yes"; Otherwise please click "No".
Part No. : GD32F1032ET6 Download File Index GD-Link Pro 1	▼ 3 <u>是(r)</u> <u>哥(N)</u>

Figure 3-22. Simultaneously adding BOOT+APP offline download file update to GD-



Link V2

📸 GD-Link Programmer 4.6.10.13769	■ 打开 Enter Programming Address × ×
: File(F) Target(T) GD-Link Refresh Edit(E) View(V) Help(H)	← → 小 ▲ 新加 3 提索"Test" ♪ 组织 ▼ 新建文件夹
GD-Link Update File Configuration	O fw32101 ★ ▲ OK Cancel 修改日期 类型
Part No.: GD32F1032ET6 -	■ 此电脑 2020/7/31 14:14 文件夹 3 D 对象 副 APP_0x08040000.bin 2021/4/16 9:52 BIN 文件 ■ and BOOT 0x08000000.bin 2021/4/16 9:52 BIN 文件
Download File Index File Path File Size Address	■ 视频
	文件名(N): BOOT_0x0800000.bin v *.biry*.hex v 2 打开(0) 取消
	m 打开 Enter Programming Address X X X
	← → ↑ ▲ 《 新加 Starting Address: 0x 8004000 程気" Test" ク
Delete Add	OK Cancel OK Cancel 例 例 例 例
Cancel Update	■ 此理論Template 2020/7/31 14:14 文件共 ③ 3D 対象 ④ APP_0x08040000.bin 2021/4/16 9:52 BIN 文件
Download	■ 视频
File Index File Path File Size ddress	
File Index File Path File Size Eddress 1 E:\Test\B00T_0x08000000.bin 0x1004 fx8000000 2 7 E:\Test\APP_0x08040000.bin 0x1004 0x8004000	文件名(N): APP_0x08040000.bin ・biry*.hex ・ 打开(O) 取消

Figure 3-23. Offline download configuration option byte feature

🐞 GD-Link Programmer 4.3.7.9536	-				_	o ×
File(F) Target(T) GD-Link Refresh Edit(E) View(V) Help(H)						
					Properties Properties Windo	× 0 ×
	GD-Link Option Bytes Configuration			×	Device(GD-Li Connect Device Inter Firmware Ve	USB[Device 0] SWD
GD-Link Update File Configuration X	E FMC OBR	0x000080AA			UID	Unknown
2	TZEN	2	Trust zone enable bit		SN	Unknown
Part No. : GD32W515PIQ6 • 1 Configure OptionBytes	SRAM1 RST		SRAM1 reset enable bit		JTAG/SWD	
Download Configuration	SPC	0xAA	Flash security protection value		Initial Speed	500 kHz
File Index File Path File Size Address	FMC_OBUSR	0x7FFFFFFF			 MCU Info MCU Part No. 	
File Index File Path File Size Address	USER	0x7FFFFFFF	Option byte USER value		MCU Part No. Endian	Unknown
	E FMC_SECMCFG0	0x003F0000			Check core ID	
	SECM0_EPAGE	0x3F	End page of secure mark area 0		Core ID	Unknown
	SECM0 SPAGE	0x0	Start page of secure mark area 0		Use RAM	Yes
	E FMC DMP0	0x00000000			RAM Address	Unknown
	DMPOEN		DMP area 0 enable		RAM Size	Unknown
	DMP0 EPAGE	0x0	End page of DMP area 0		UID	Unknown
				<u> </u>	Flash Info	
	End page of secure mark area 0				Size	Unknown
Delete Add					Flash Base	Unknown
Cancel Update	Tips: please refer to the chip User Mar	nual to modify the		Cancel		
Output						▼ 0 ×
Show initial information and roftware information here. Wydating offline=rycgramming file: — Traing complete: — Updating complete:						
H 4 > H Status						
Beady						CAP NUM SCRL -:

After updating the offline burning file to GD-Link V2, refer to the <u>Pin definitions and wiring</u> <u>methods</u> section related to hardware connection with the target chip. Manually press the K1 button, if LED1 entering rapid blinking mode, indicating that the offline burning process is ongoing. When the buzzer beeps, it signifies the completion of the offline burning. At this time, LED1 is always bright. If the buzzer does not beep, and LED1 is turned off after blinking, it indicates an offline burning failure.

3.2.4. Machine singal triggered programming

GD-Link V2 offers machine-triggered programming functionality. The signal interface pinout diagram is shown in *Figure 3-24. Machine signal programming pin distribution* <u>schematic diagram</u>. The functions of each pin for the machine-triggered programming interface are described in <u>Table 2-1. GD-Link V2 pin function definitions</u>. After updating



GD-Link V2 Adapter User Guide

the programming file into the programmer as described in the offline programming section, users can initiate the programming process by providing a 100ms low-level pulse signal to the T_START pin. During the programming process, the T_BUSY pin remains at a low-level signal. When the programming is successful, the T_GOOD pin generates a low-level signal, while a low-level signal on the T_NG pin indicates a programming failure.

Figure 3-24. Machine signal programming pin distribution schematic diagram

GND <u>—</u> JP3 <u>T NG</u> <u>T BUSY</u> <u>T GOOD</u> <u>T START</u> <u>5</u> <u>4×1P2.54</u>

Table 3-1. Machine signal programming pin function definition

Pin Number	Pin Name	Description
1	GND	Power ground
2	T_NG	Defaults to a high level. When burning fails, this pin goes to a low level.
3	T_BUSY	Defaults to a high level. When burning is in progress, this pin goes to a low level.
4	T_GOOD	Defaults to a high level. When burning is successful, this level goes to a low level.
5	T_START	Defaults to a high level. When this pin receives a low-level signal with a width of 100ms, burning starts.

3.2.5. Virtual USB disk drag and drop programming

Insert the GD-Link V2 USB into the PC port. There will be a USB mass storage device in the PC device manager, and a GigaDevice disk with the GD logo will appear in the local disk. As shown in *Figure 3-25. USB mass storage device* and *Figure 3-26. Virtual USB drive*.



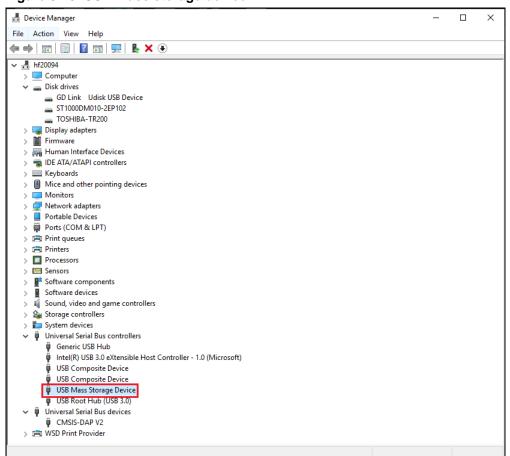
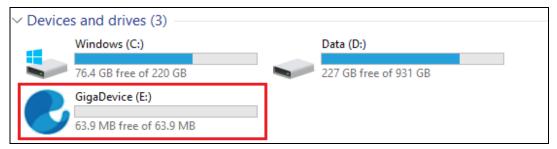


Figure 3-25. USB mass storage device

Figure 3-26. Virtual USB drive



Double-click to open the disk. Inside the disk, there is a CONFIG.TXT file. By modifying the content of this file and saving it, the initialize the programming parameter can be configured. The content of the CONFIG.TXT file is shown in <u>Table 3-2. CONFIG.TXT file content</u>.

Table 3-2. CONFIG.TXT file content

# Program config:	
Target CPU: ARM	
Program Flash start address: 0x08000000	
Erase method: Page	
Reset method: Software	
Read protection: Disable	
Sram start address: 0x20000000	



Note: Keep the format of this TXT UTF-8. Please configure programming parameters follow the format before programming. E.g: # Program config: Target CPU: ARM Program start address: 0x08000000 Erase method: Chip Reset method: Software Read protection: Disable Sram start address: 0x2000000

The options and descriptions for each parameter configuration are as shown in <u>Table 3-3.</u> <u>Drag-and-Drop programming configuration parameter definitions</u>.

Parameter	Options	Description		
	ARM	Select ARM as the target chip core		
Target MCU core architecture	RISC-V	Select RISC-V as the target chip core		
	0x08XXXXXX	Program flash start address		
Program flash start address	0x00^^^^	0x08XXXXXX		
	0x0CXXXXXX	Program flash start address 0x0CXXXXX		
Erase method	Page	Flash erasing method is page erasing		
Erase method	Chip	Flash erasing method is full chip erasing		
Reset method	Software	Reset method after completing chip download		
		is software reset		
Reset method	Hardware	Reset method after completing chip download		
		is hardware reset		
	0x2XXXXXXX	Target chip's SRAM start address is		
Sram start address	0.2	0x2XXXXXXX		
Sidili sidit dudiess	0x3XXXXXXX	Target chip's SRAM start address is		
		0x3XXXXXX		
Dahus istarfaas	SWD	Select SWD as the download interface(only		
	300	for ARM)		
Debug interface	JTAG	Select JTAG as the download interface(only		
	JIAG	for RISC-V)		

Table 3-3. Drag-and-Drop programming configuration parameter definitions

After configuring the programming parameters, save and close the file. Refer to the <u>Hardware</u> <u>introduction</u> section, connect GD-Link V2 to the target chip via SWD (GD Cortex-M core MCU) or JTAG interface (GD RISC-V core MCU) correctly, then copy or drag the binary xxx.bin or executable file xxx.hex generated by the IDE or compiler toolchain to the recognized GigaDevice disk device. The programmer will automatically identify the target chip and complete the file programming.

After programming is complete, the virtual USB device will unmount and then remount from the disk. Once mounting is complete, open the GigaDevice disk. If the disk contains only the



CONFIG.TXT file, it indicates a successful file programming. If a FAIL.TXT file appears in the disk, it indicates a programming failure. Double-click to open FAIL.TXT and check the reason for the programming failure.

Note:

1. When the debugger loses power and is unplugged and reconnected, the previous programming parameters will revert to default values.

2. The binary xxx.bin file should be generated by the compiler and the corresponding download target address should be filled correctly, otherwise, programming failure may occur.

3.3. Debug function

3.3.1. SWD /JTAG debugging

Debugging with KEIL (version 5.27 and above):

Complete the KEIL configuration according to <u>IDE programming</u> chapter, click the icon button of "Start/Stop Debug Session" in the KEIL menu bar to enter the debugging interface, as shown in <u>Figure 3-27. KEIL debugging interface</u>.

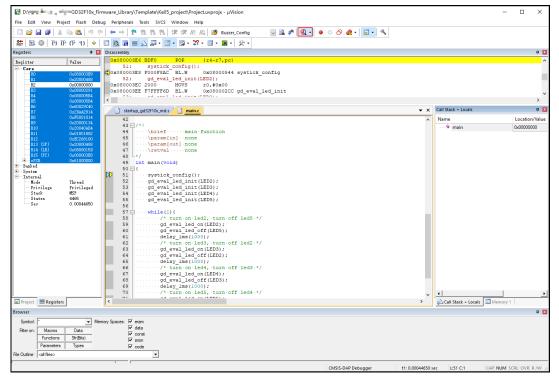


Figure 3-27. KEIL debugging interface

Debugging with IAR (version 8.50 and above):

Complete the IAR configuration according to IDE programming chapter, click the icon button



of "Download and Debug" in the IAR menu bar to enter the debugging interface, as shown in *Figure 3-28. IAR debugging interface*.

Figure 3-28. IAR debugging interface

GD32103B_EVAL - IAR Embedded Wo	rkbench IDE - Arn	a 8.50.9					- 0	×
File Edit View Project Debug Disass	embly CMSIS-DAP	Tools Window Help						
) H	▶ 00 ±				
	K main.c X			Disassembly				v 0
Debug	-		fo		~ Memory	~	PA I	
	42	\brief main function	^		- memory			
Files 🌣 •	43	\param[in] none		Disassembly				
□ ● GD32103B_EVAL - D ✓		\param[out] none		delay-			P0 (P1)	
	45 46 4/	\retval none		0x800'038a 0x800'038c		LDR SUBS	RO, [R1] RO, RO, #1	
		main (void)		0x800'038e		STR	R0, [R1]	
-B GD32F10x_EVAL	48 🚍 {			}				
-B 🖬 GD32F10x_Peripherals		systick_config();		0x800'0390		BX	LR	
- 🖽 🖷 Startup	50 51	/* enable the LEDs clock */		0x800'0392		MOAR	R0, R0	
🛏 🛋 Output		rcu periph clock enable (RCU GPIOC);			0xe000'ed18	DC32 DC32	0xe000'ed18	
	53	real_bright_brook_chapic(hos_bright))			0xe000'e400 0xe000'e014	DC32 DC32	0xe000'e400 SYST RVR	
	54	/* configure LEDs GPIO port */			0xe000'e014	DC32 DC32	0xe000'e018	
		gpio_init(GPIOC, GPIO_MODE_OUT_PP, GPIO_OSPEED_50MHZ, GPIO_PIN_6 GPIO_PIN	_7		0xe000'e010	DC32	0xe000'e010	
	56 57	/* reset LEDs GPIO pin */		0x800'03a8	0x2000'0000	DC32	SystenCoreC	lock
		<pre>gpio bit reset(GPIOC, GPIO PIN 6 GPIO PIN 7 GPIO PIN 8 GPIO PIN 9);</pre>			0x2000'0004	DC32	delay	
	59			int main(void))			
		while(1){		{				
	61 62	/* turn on LED2 */ opio bit set(GPIOC, GPIO PIN 6);		0x800'03b0	main:	PUSH	{R4, LR}	
	63	<pre>gplo_bit_set(GridC, Grid_FIN_6); /* insert 200 ms delay */</pre>		systick co		1031	JNA, TNJ	
	64	delay 1ms(200);			0xf7ff 0xffce	BL	systick_con	fig
	65				_clock_enable(
	66	/* turn on LED3 */			0xf240 0x6004		RO, #1540	
	67 68	<pre>gpio_bit_set(GPIOC, GPIO_PIN_7); /* insert 200 ms delay */</pre>			0xf000 0xf837		rcu_periph_	
	69	delay ims(200);		gpio_init 0x800'03be	GPIOC, GPIO_MC	DE_OUT_PP, LDR.N	GPIO_OSPEED_ R4. [PC. #0:	
	70	/(//			: 0x4414 0x7370		R4, [PC, #0 R3, #960	x68]
	71	/* turn on LED4 */		0x800'03c4		MOVS	R2, #3	
	72	<pre>gpio_bit_set(GPIOC, GPIO_PIN_8); /* insert 200 ms delay */</pre>		0x800'03c6		MOVS	R1, #16	
	73	<pre>/* insert 200 ms delay */ delay ims(200);</pre>		0x800'03c8	0x0020	MOVS	R0, R4	
	75	acra3_100 (2007)			0xf7ff 0xff19		gpio_init	
	76	/* turn on LED5 */			reset(GPIOC, GF			GPI
	77	<pre>gpio_bit_set(GPIOC, GPIO_PIN_9);</pre>		0x800'03ce 0x800'03d2	0xf44f 0x7170	MOV.V MOVS	R1, #960 R0, R4	
	78	/* insert 200 ms delay */			0xf7ff 0xff7a		qpio bit re	-
GD321038_EVAL	<		> v	<				
ebug Log								v 0
Log								^
Wed Nov 09, 2022 09:35:18: 1496 byte	s verified (292.19	Kbytes/sec)						
Ward Nov 09, 2022 09:35:18: Downloa								×
ulid								• Q
Messages								
Building configuration: GD32103B_EV	/AL - Debug							
Updating build tree	5							
<								>
leady					Ln 47, Col 1	System	大写 数字 改写	

Debugging with Eclipse

Complete the Eclipse configuration and debugging according to *IDE programming* chapter.

3.3.2. SWO function

The Serial Wrie Output (SWO) function uses the ITM (Instrumentation Trace Marcrocell) module in the Cortex-M kernel to output debugging information in the kernel through the SWO pin of the chip. The connection mode between the burner and the chip is referred to *Figure 2-4. SWD* + *SWO interface connection diagram*.

Note: For details about whether the chip supports the SWO function, see the corresponding user manual and datasheet.

SWO configuration in KEIL

Select the Debug tab in Options for Target and select CMSIS-DAP ARMv8-M Debugger from the drop-down list, referring to *Figure 3-29. SWO configuration step 1 in KEIL*.



🕅 Options for Target 'GD32E503Z_EVAL'	×			
Device Target Output Listing User C/C++ (Ad	C6) Asm Linker Debug Utilities			
C Use Simulator with restrictions Settings □ Limit Speed to Real-Time Image: Setting to the set of the s	 ✓ Use: CMSIS-DAP ARMv8-M Debugg ▼ Settings ULINKplus Debugger J-LINK / J-TRACE Cortex ST-Link Debugger Initializatic Pemicro Debugger ULINK Pro ARMv8-M Debugger ULINK Pro ARMv8-M Debugger ILINK2/MF ARMv8-M Debugger Models ARMv8-M Debugger IF Breakpoints ✓ Toolbox ✓ Watch Windows Tracepoints ✓ Memory Display ✓ System Viewer 			
CPU DLL: Parameter: Dialog DLL: Parameter:	Driver DLL: Parameter: SARMV8M.DLL -MPU Dialog DLL: Parameter: TCM.DLL -pCM33 Wam if outdated Executable is loaded			
Manage Component Viewer Description Files				
OK Cancel Defaults Help				

Figure 3-29. SWO configuration step 1 in KEIL

Select the Trace tab in Settings, and the configuration interface is shown in Figure 3-30. SWO configuration step 2 in KEIL.

igure 3-30. SWO configuration	n step 2 in KEIL	
CMSIS-DAP ARMv8-M Target Driver	Setup	
Debug Trace Flash DownLoad Core Clock: 180.000000 MHz Trace Clock: 180.000000 MHz Trace Pot Serial Wire Output - UART/NRZ SWO Clock Prescaler: 57 Autodetect SWO Clock: 3.157894 MHz	✓ Trace Enable 2 ✓ Use Core Clock Timestamps 3 ✓ Enable Prescaler: 1 ▼ PC Sampling Prescaler: 1024*16 ▼ Periodic Periodic ○ n Data R/W Sample	Trace Events CPI: Cycles per Instruction EXC: Exception overhead SLEEP: Sleep Cycles LSU: Load Store Unit Cycles FOLD: Folded Instructions EXCTRC: Exception Tracing
-ITM Stimulus Ports Enable: 0x00000001 31 P	Port 24 23 Port 16 15	4 Port 8 7 Port 0

Port 31..24 🔽

OK

Privilege: 0x0000008

The Trace pin is enabled in the code. For an MCU with Trace mode configuration, the Trace mode needs to be configured as asynchronous mode, as shown in Table 3-4. Trace mode enable. For details about how to enable Trace mode, see the Debug chapter in the user

Port 23..16

Cancel

Port 15..8

 \times

Help

Port 7..0



manuals of each series of MCUs.

Table 3-4. Trace mode enable

DBG_CTL |= DBG_CTL_TRACE_IOEN;

In the code, the serial printf output is redirected to the ITM output, and the added code is shown in *Table 3-5. Printf retarget*.

Table 3-5. Printf retarget

#define ITM_Port8(n)	efine ITM_Port8(n) (*((volatile unsigned char *)(0xE0000000+4*n)))			
#define ITM_Port16(n)	(*((volatile unsigned short*)(0xE0000000+4*n)))			
#define ITM_Port32(n)	n) (*((volatile unsigned long *)(0xE0000000+4*n)))			
#define DEMCR	(*((volatile unsigned long *)(0xE000EDFC)))			
#define TRCENA	0x01000000			
int fputc(int ch, FILE *f)				
{				
if (DEMCR & TRCEN	IA)			
{				
while (ITM_Port32(0) == 0) {};				
$ITM_Port8(0) = ch;$				
}				
return(ch);				
}				

Enter the debugging interface, select "View" -> "Serial Windows" -> "Debug(printf)Viewer", open the serial port printing interface, run the code at full speed, and the printed information will be displayed in the Debug(printf)Viewer window. The Debug(printf)Viewer window in is shown in *Figure 3-31. Debug (printf) viewer window in KEIL*.



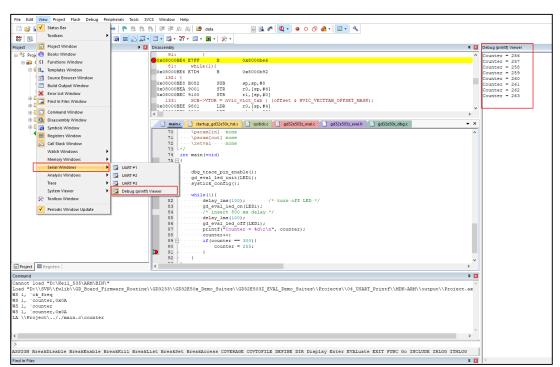
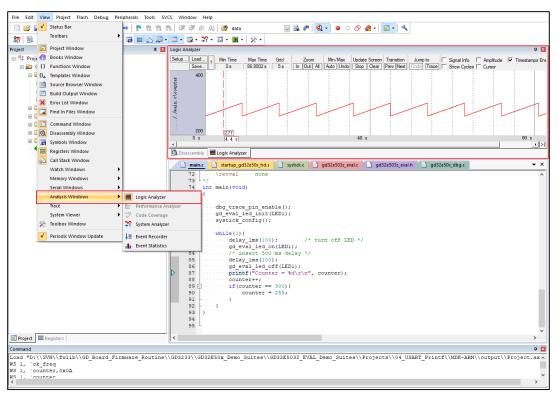


Figure 3-31. Debug (printf) viewer window in KEIL

Enter the debugging interface, choose "View" -> "Analysis Windows" -> "Logic Aanlyzer", open the logic analyzer window, add the variables to be observed, run the code at full speed, and the value of the variable will be displayed in the logic analyzer window through the waveform. The Logic Aanlyzer window is shown in *Figure 3-32. Logical Analyzer window in KEIL*.



Figure 3-32. Logical Analyzer window in KEIL



3.4. Virtual serial port printing

When the GD-Link V2 USB is inserted into the PC port, a USB serial device will appear on the PC Device Manager port (COM and LPT) interface (there is no driver for WIN10 system, and the corresponding driver should be installed for win7 system), as shown in *Figure 3-33. USB serial device*, refer to *Figure 2-5. Serial interface connection diagram*. Connect GD-Link V2 to the serial port pin hardware of the target chip, configure the correct serial port baud rate and other information in the serial port debugging assistant, and write the data to be sent to the target MCU serial port receiver through the serial port debugging assistant. The target MCU can also print the information to be printed through the USB port of the burner to the upper computer interface of the serial debugging assistant through the serial port transmitter and display it, as shown in *Figure 3-34. USB virtual serial printing*.



Figure 3-33. USB serial device

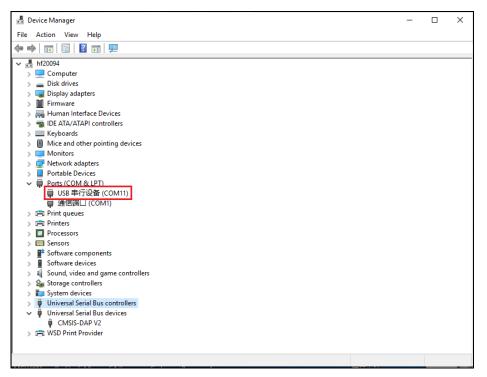
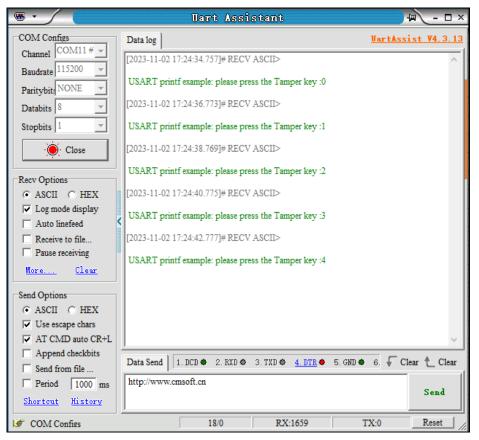


Figure 3-34. USB virtual serial printing





4. Revision history

Table 4-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Nov.1 2023
1.1	Add the output voltage section	Jan.2 2023



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.

© 2024 GigaDevice - All rights reserved