第一章 Eclipse IDE for C/C++ Developers (2018-12 版) 的安装

- 一、硬件准备
- 1. 带有 windows 10 x64 操作系统的笔记本
- 2. 实验板 STM32F405RG 核心板 (自制实验板,用于验证)
- 3. J-Link V9 (用于验证)
- 二、安装步骤
- 1. 安装 jre-8u202-windows-i586.exe、jre-8u202-windows-x64.exe 。
- 2. 将 eclipse 复制到 C 盘,打开 eclipse。

3. 在线安装 GNU MCU Eclipse Plug-ins,如"在线安装(推荐的方式)"文件所示,网速必须 要快,最好是用 VPN 加速。

4. 复制 ARM Embedded GCC 到 C:/User/xxx/AppData/Roming/GNU MCU Eclipse 目录,GNU MCU Eclipse 目录是新建的。

5. 复制 Build Tools 到 C:/User/xxx/AppData/Roming/GNU MCU Eclipse 目录。

- 6. 复制 OpenOCD 到 C:/User/xxx/AppData/Roming/GNU MCU Eclipse 目录。
- 7. 复制 QEMU 到 C:/User/xxx/AppData/Roming/GNU MCU Eclipse 目录。
- 8. 在线安装 System Workbench for STM32 Bare Metal Edition (Ac6),详见第二章。
- 三、需要下载的文件列表
- 1. jre-8u202-windows-i586.exe
- 2. jre-8u202-windows-x64.exe
- eclipse-cpp-2018-12-R-win32-x86_64.zip https://www.eclipse.org/downloads/packages/
- 4. gnu-mcu-eclipse-arm-none-eabi-gcc-8.2.1-1.4-20190214-0604-win64.zip
- 5. gnu-mcu-eclipse-build-tools-2.11-20180428-1604-win64.zip
- 6. gnu-mcu-eclipse-openocd-0.10.0-11-20190118-1134-win64.zip
- 7. gcc-arm-none-eabi-8-2018-q4-major-win32.exe

其中 7 为可选安装。

百度网盘下载地址:

https://pan.baidu.com/s/1EgtBxMQaSFh1QCPxgZv_3Q

第二章 System Workbench for STM32 和 J-Link 驱动的安装

一、System Workbench for STM32 的安装

You must then create a new update site, to be able to install System Workbench for STM32 - Bare Metal Edition:

1. You should click on "Add:"

2. Then give a name to the update site (System Workbench for STM32 - Bare Machine edition) and set the location to

http://ac6-tools.com/Eclipse-updates/org.openstm32.system-workbench.update-site-v2

3. Then click "OK" to create the update site

Then you should select all of the OpenSTM32 tools and click "Next>"

You will then just have to accept the license and System Workbench for STM32 - Bare Metal edition will install itself in your Eclipse setup.

Note that you will probably have to restart Eclipse for the tools to be available; Eclipse will automatically suggest the restart.

二、J-Link 驱动的安装

🔜 SEGGER J-Link DLL Updater V6.44b	×
3 applications found that can be updated to V6.44b of the J-Link software:	
 □ Keil MDK-ARM (DLL V6.40 in "C:\KeiL_v5\ARM\Segger") □ NXP S32 Design Studio (DLL V6.42a in "C:\NXP\S32DS_ARM_v2018.R1\Drivers\Segger") □ Atollic TrueSTUDIO for STM32 V9.3.0 (DLL V6.42b in "C:\Program Files (x86)\Atollic\TrueSTUDIO for STM32 9.3.0\ 	\Servers\J-Link_gdbserver'')
Select All Select None Select the ones you would like to replace by this version. The previous version will be renamed and kept in the same folder, allowing manual "undo". In case of doubt, do not replace existing DLL(s). You can always perform this operation at a later time via start menu.	Ok Cancel

图 2-1 J-Link 驱动的安装 在安装 J-Link 驱动时,选择不覆盖,如图 2-1 所示。 第三章 在 System Workbench for STM32 中使用 J-Link 的方法 上一章已经安装了 J-Link 驱动,由于 J-Link 应用广泛,较为重要,这里重点介绍。

1.在 Preferences\MCU 对应 Item 设置 J-Link 的全局路径,如图 3-1 所示。

Preferences	— 🗆 X
type filter text	Global SEGGER J-Link Path 🗢 👻 🗸
 > C/C++ ChangeLog > Docker > Help > Install/Update > Library Hover > MCU Global ARM Toolchains Paths Global Jumper Path Global Jumper Path Global OpenOCD Path Global QEMU Path Global RISC-V Toolchains Paths Global SEGGER J-Link Path Workspace ARM Toolchains Paths 	Configure the location where SEGGER J-Link is installed. The values are stored within Eclipse. Unless redefined more specifically, they are used for all projects in all workspaces. After installing SEGGER updates, restart Eclipse for the defaults to be re-evaluated and use the Restore Defaults button to configure the new location. Executable: JLinkGDBServerCL.exe Folder: C:\Program Files (x86)\SEGGER\JLink_V644b Browse Restore Defaults Apply
2 2 4 0	Apply and Close Cancel

图 3-1 J-Link 全局路径设置

2.在 Debug Configuration \ Executable path 框设置 JlinkGDBServerCL.exe 的路径。

设置字符串为: "\${jlink_path}/\${jlink_gdbserver}",如图 3-2 所示。

 Debug Configurations Create, manage, and run configurations 	5			- • ×
📑 🖻 🎭 📄 🗙 🖻 🐎 🔹 ,	Name: test Debug]
Create, manage, and run configurations Image: and run config	Browse Variables			
Launch Group (Deprecated)	SWO port:	2332 ✓ Verify downloads ✓ In	- □ × A state of the	
🚱 ST's STM32 MPU Debugging	Telnet port:	2333 🗹 Local host only	Immon 20 SVD Path Connect to running target U644b//LinkGDBServerCLexe pace preferences pages or the project properties page) Supported device names (USB serial or IP name/address) 1000 kHz 2 Verify downloads [Initialize registers on start 2 Local host only [Silent] Browse Yariables Local host only [Silent] Browse Allocate console for semihosting and SWO one-eabi-gdb Browse Variables tabols.arm-none.win32_1.17.0.201812190825\tools\compiler\bin\arm-none-eabi-gdb No Revert Apply Debug Close	
	Log file:			Browse
	Other options:	-singlerun -strict -timeout 0 -nogui		
	Allocate console	e for the GDB server Allocate console for semihosting and SWO		
	GDB Client Setup			
	Executable name:	\${openstm32_compiler_path}\arm-none-eabi-gdb	Browse	Variables
	Actual executable:	C:\eclipse\plugins\fr.ac6.mcu.externaltools.arm-none.win32_1.17.0.201812190825\tools\compiler\bin\arm-non	ie-eabi-gdb	
	Other options: Commands:	set mem inaccessible-by-default off		^ ~
	Remote Target			~
< >> Filter matched 17 of 18 items			Revert	Apply
?			Debug	Close

图 3-2 System Workbench for STM32 中使用 J-Link 驱动的设置

3.在 Device name 编辑框设置芯片名称,如 STM32F405RG,不分大小写,如图 3-2 所示。 4.在 GDB Client Setup\Executable name 框,设置 GDB 客户端名称。

设置字符串为: "\${openstm32_compiler_path}\arm-none-eabi-gdb",如图 3-2 所示。

第三章 STM32CubeMX 导出的 SWS4STM32 工程的调试设置

- 1. 选择 SWS4STM32 工程,同时去掉 Generate Under Root 的选择(默认打勾)。 选择不打勾,工程文件会集中放在一个子目录中。
- 2. 选择使用最新的库,这里是 STM32Cube_FW_F4V1.24.0 版本。如图 4-1 所示。

STM32 CubeMX	File	Window	Help	12	F 🕒 🔰	\mathbf{X}	57
Home > STM32F405	RGTx 🔰 test.ioc -	· Project Manager >		GENER	RATE CODE		
Pinout & Config	juration	Clock Configuration	Project Manager		То	ols	
Project	Project Name Project Location H:\stm32f405r\STM32F40 Application Structure	5RGT6_Test\STM32F405RG_HAL\					
Code Generator	Basic foolchain Folder Location H:\stm32f405r\STM32F40 foolchain / IDE SW4STM32	SRGT6_Test\STM32F405RG_HAL\	not generate the main() nerate Under Root				
Advanced Settings	inker Settings Minimum Heap Size Minimum Stack Size	0x200 0x400					
	Acu and Firmware Packa Mcu Reference STM32F405RGTx Firmware Package Name STM32Cube FW_F4 V1.2	ge and Version 4.0					
	✓ Use Default Firmware	Location pository/STM32Cube_FW_F4_V1.24 (0	Browse			

图 4-1 利用 STM32CubeMX 生成 SWS4STM32 工程 3.可以使用 J-Link 进行程序调试。上一章有说明,确认,如图 4-2 所示。



图 4-2 利用 STM32CubeMX 生成 SWS4STM32 工程,使用 J-Link 单步调试

4. 可以使用 ST-Link 进行程序调试。如图 4-3 所示。

eclipse-workspace - test/Application/User/main.c - Eclipse IDE

rile Edit Source Relactor Navigate Search Project Run	window help		
🔨 🐐 🔳 🎋 Debug 🗸 🔽 test_ac6	✓ Image: A the second seco	3	🕹 🗄
🗱 Debug 🕴 🎦 Project Explorer 🛛 🕹 🛛 💥 🖬 🗢 🗢 🗖	i main.c ☆ c 0x80006a8	- 8	(X):
✓ c test_ac6 [GDB OpenOCD Debugging]	76 /* Reset of all peripherals, Initializes the Flash interface and the <u>Systick</u> . */	^	
v 🥮 test.elf	77 HAL_Init();		
W - @ Thread #1 (Sucponded : Step)	78		IN
• m au # (suspendeu : step)	79 7* USER CODE BEGIN LOIT *7		
main() at main.c:95 0x8000690	00 91 /* USED CODE END To:+ */		
🔎 openocd.exe			
🚚 arm-none-eabi-gdb	83 /* Configure the system clock */		
	84 SystemClock_Config();		
	85		
	86 /* USER CODE BEGIN SysInit */		
	87		
	88 /* USER CODE END SysInit */		
	09 90 /* Initialize all configured peripherals */		
	90 / Initialize all configured peripherals /		
	92 MX TIM2 Init():		
	93 /* USER CODE BEGIN 2 */		
	94 HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_0);		
	95 HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_0);		
	96		
	97 HAL_GPI0_TogglePin(GPI0B, GPI0_PIN_1);		
	98 HAL_GPIO_TOgglePIN(GPIOB, GPIO_PIN_1);		
	100 June Code END 2 17		
	101 /* Infinite loop */		
	102 /* USER CODE BEGIN WHILE */		
	103 while (1)	~	
	<	>	<
	🖳 Console 🛿 🔠 Registers 🖹 Problems 🕡 Executables 📓 Debugger Console 🕕 Memory		
	test ac6 [GDB OpenOCD Debugging] openocd.exe		
	225 42 (/64) 929999999999999999999		
	(26) d3 (/64): 0x0000000000000		
	(27) d4 (/64): 0x000000000000000		
	(28) d5 (/64): 0x0000000000000000		
	(29) d6 (/64): 0x000000000000000		
	(30) d7 (/64): 0x000000000000000		
图 4-3 利用 STM32Cubel	MX 生成 SWS4STM32 工程,使用 ST-Link 单步调试		

5.GDB OpenOCD Debugging \ Config Options 参数设置,如图 4-4 所示。

Debug Configurations				- 🗆 X
Create, manage, and run configurations	i			Ť
Image: String Strin	Iame: test_ac6 Main ≸ Debugg OpenCCD Setup Start OpenOCD Executable path: Actual executable: GDB port: Telnet port: Tcl port: Config options:	er Startup & Source Common & SVD Path locally Stopenocd_path/Stopenocd_executable) Cr/Users/wjandcf/AppData/Reaming/GNU MCU Eclipse/OpenOCD/0.10.0-11-20190118-1134/bin/openocd.exe to change it use the global or workspace preferences pages or the project properties page) 3333 4444 6666 -f "interface/stlink-v2.cfg"	Browse	Variables
	Allocate console GDB Client Setup Start GDB sessi Executable name: Actual executable: Other options: Commands:	- r Target/stm32Hx.ctg* e for OpenOCD Allocate console for the telnet connection on \$(openstm32_compiler_path)\arm-none-eabi-gdb Ct\cetlipse\plugins\fr.ac6.mcu.externaltools.arm-none.win32_1.17.0.201812190825\tools\compiler\bin\arm-none- set mem inaccessible-by-default off	Browse eabi-gdb	Variables

图 4-4 GDB OpenOCD Debugging Config Options 参数设置

在 STM32CubeMX 生成的工程中,笔者测试使用 Ac6 STM32 Debugging 调试不成功(以往版本都是可以用的)。GDB OpenOCD Debugging 则没有问题, Config Options 填入如下参数: -f "interface/stlink-v2.cfg"

-f "target/stm32f4x.cfg"

Executable name 默认字符串为 "\${cross_prefix}gdb\${cross_suffix}", 替换为 "\${openstm32_compiler_path}\arm-none-eabi-gdb"。

6. Global OpenOCD Path 设置,如图 4-5 所示。

Preferences				
type filter text	Global Ope	nOCD Path	<	┝ ▾ ⇔ ▾ ▾
 MCU Global ARM Toolchains Paths Global Build Tools Path Global Jumper Path 	Configure th used for all After installi	e location where GNU MCU Eclipse OpenOCD is installed. The values are stored within Eclipse. Unler projects in all workspaces. ng OpenOCD updates, restart Eclipse for the defaults to be re-evaluated and use the Restore Default	is redefined more speci	fically, they are ne new location.
Global OpenOCD Path	Executable:	openocd.exe		
Global pyOCD Path Global OEMU Path	Folder:	C:/Users/wjandcf/AppData/Roaming/GNU MCU Eclipse/OpenOCD/0.10.0-11-20190118-1134/bin	Browse	xPack
Global RISC-V Toolchains Paths Global SEGGER J-Link Path Workspace ARM Toolchains Paths Workspace Build Tools Path Workspace Jumper Path Workspace OpenOCD Path Workspace gyOCD Path Workspace QEMU Path Workspace RISC-V Toolchains Paths Workspace SEGGER J-Link Path				
> Oomph				
> Remote Development				
> Remote Systems			Restore Defaults	Apply
? 🕍 🖆 🛞			Apply and Close	Cancel

图 4-5 Global OpenOCD Path 设置

openocd.exe

C:/Users/wjandcf/AppData/Roaming/GNU MCU Eclipse/OpenOCD/0.10.0-11-20190118-1134/bin 7. 可以使用 cmsis-dap 进行程序调试。如图 4-6、图 4-7 所示。

	No. Internet de	2			
Image: Start Gog Image: Start Gogg Image: Start Gogg Image: Start Gogg Image: Start Goggg <th>Name: [test_cmsis_dat Main Debugg OpenOCD Setup Start OpenOCD Executable path: Actual executable: GDB port: Telnet port: Tid port: Discutable: Setup of the setup of the setup</th> <th>p er e Startup Goenocd_path//Slope Cr/Users/vsjandef/AppD (to change it use the gls 3333 4444 6666</th> <th>urce Common St SVD Path</th> <th>Browse</th> <th>Variables.</th>	Name: [test_cmsis_dat Main Debugg OpenOCD Setup Start OpenOCD Executable path: Actual executable: GDB port: Telnet port: Tid port: Discutable: Setup of the setup	p er e Startup Goenocd_path//Slope Cr/Users/vsjandef/AppD (to change it use the gls 3333 4444 6666	urce Common St SVD Path	Browse	Variables.
	Coming options: Allocate console GDB Client Setup Start GDB sessis Executable name: Actual executable: Other options: Commands:	-f "iterget/stm32(4xcfg -f "target/stm32(4xcfg -for OpenOCD on Stopenstm32_compiler, C(veclipse\plugins\fr.ac set mem inaccessible-b		Browse	Variables



Config options 字符串接口修改为 -f "interface/cmsis-dap.cfg" Executable name 默认字符串为 "\${cross_prefix}gdb\${cross_suffix}",替换为 "\${openstm32_compiler_path}\arm-none-eabi-gdb"。



图 4-7 使用 cmsis-dap 进行程序单步调试

由于 OpenOCD 这两年改动比较大, GNU MCU Eclipse 也是 GNU ARM Eclipse 改名而来的, 笔者奋斗了一天,才设置好 OpenOCD 的调试,但 J-Link 的设置没多大变化。路径中的 wjandcf 是笔者的当前电脑中的用户名。

开源软件的力量是强大的,也许现在你们不用 GNU MCU Eclipse,但将来不一定。

对于目前主流的 STM32/S32K 软件开发, J-Link 是使用最广泛的,但正版价格昂贵,正版 ST-LinkV3 是 USB2.0 高速通信,性价比很高,由于笔者主要搞汽车电子,ST-LinkV3 没有买 来实验。

参考信息: https://gnu-mcu-eclipse.github.io/debug/openocd/ http://www.openstm32.org/HomePage

作者: wjandsq@163.com 日期: 2019/03/20