

W60X OpenOCD Debugging Guide V0.1

Beijing Winner Microelectronics Co., Ltd.

Address: 18th Floor, Yindu Building, No.67, Fucheng Road, Haidian District, Beijing , China

Tel: +86-10-62161900

Website: www.winnermicro.com



Document History

Version	Completion Date	Revision Record	Author	Auditor
V0.1	2018-11-16	Initial release	LiLm	



Content

Do	cument	Histor	y	1
Co	ntent			2
1	Introd	luction		4
	1.1	Pı	ırpose	4
	1.2	R	eaders	4
	1.3	Te	erm Definition	4
2	Quick	c Start:	How to use Eclipse+OpenOCD to debug W60X.	5
	2.1	Η	ow to Connect W60X	5
	2.2	In	stal Eclipse	5
	2.3	D	ownload Source Code of W60X_SDK	5
	2.4	In	nport SDK project to Eclipse	5
	2.5	С	onfig OpenOCD to Start Up	. 12
	2.6	С	onfig Eclipse Debugging Function	. 17
	2.7	С	ompiling SDK Source Code with Optimization Level	. 22
	2.8	С	ompile SDK	. 24
	2.9	St	art to Debug	. 26
	2.10	St	op Debugging	. 30
	2.11	С	ompile Release Firmware	. 31
	2.12	U	pdate Firmware through UART Port	. 32
3	Open	OCD U	Jsing Guide	. 32
	3.1	0	penOCD Introduction	. 32
	3.2	In	stall OpenOCD	. 33
		3.2.1	Install libusb	. 34
		3.2.2	Install HIDAPI library	. 34
		3.2.3	Install OpenOCD	. 35
	3.3	D	ebugging with OpenOCD Command Lines	. 36
		3.3.1	Start up OpenOCD with JLINK Simulator	. 36
		3.3.2	Start up OpenOCS with CMSIS-DAP Simulator	. 38
		3.3.3	Host Computer Connects with OpenOCD	. 39
		3.3.4	Common Commands	. 40
		3.3.5	Gdb Debugging Examples	. 41
4	Appe	ndix		. 44



4.1	Install zylincdt to Eclipse	44
-----	-----------------------------	----



1 Introduction

1.1 Purpose

This document introduces how to use GDB to debug with OpenOCD in Eclipse and how can W60X developers deal with single step debugging with OpenOCD.

1.2 Readers

The developers with W60X.

1.3 Term Definition

OpenOCD: Open On-Chip Debugger



2 Quick Start: How to use Eclipse+OpenOCD to debug W60X.

2.1 How to Connect W60X

JTAG simulator should be used with OpenOCD debugging. Choose one type of JTAG simulator (such as JLINK or CMSIS-DAP and so on) and connect the JTAG simulator to W60X. Following figure is the connecting method:

W600	TMS (pb6) TCK (pb7) GND	TMS TCK GND	JTAG仿真器
	GND VDD33	 VREF	

After connected, W60X powers on, and connect JTAG simulator to host computer.

Different JTAG simulator has different driver. For JLINK simulator, users can refer to chapter 3.3.1 to install the driver. For CMSIS-DAP simulator, no driver should be installed. For other JTAG simulators, please install the driver by yourselves.

2.2 Instal Eclipse

- Download the compressed package from <u>http://www.winnermicro.com/html/1/156/158/497.html</u>. It provides Eclipse (integrated zylincdt plug-in unit), Cygwin(OpenOCD has been installed), Cross compiling tools(arm-none-eabi-gcc) etc.
- Discompressed the package and read the ReadMe.txt firstly. Then double-click W60X_IDE.exe to start up configured Eclipse.

2.3 Download Source Code of W60X_SDK

The SDK package can be downloaded from <u>http://www.winnermicro.com</u>. The Version number supported OpenOCD should be G3.1 or higher version.

2.4 Import SDK project to Eclipse

Following figures are importing setps:



File	Edit	Source	Refactor	Navigate	Search	Project	Run	W
	New Open	File				Alt+Sh	ift+N	>
	Close					C	trl+W	
	Close	All				Ctrl+Sh	ift+W	
	Save					C	Ctrl+S	
	Save / Save / Reven	As All t				Ctrl+Sl	nift+S	
-A	Move. Renar	me					F2	
£	Refree	sh					F5	
	Conve	ert Line De	elimiters To	, ,			:	>
8	Print					(Ctrl+P	
	Switch Resta	n Worksp rt	ace				2	>
2	Impor	rt						
2	Expor	t						
	Prope	rties				Alt+	Enter	
	1 wm_	cpu.c [N	/600_SDK_0	63.0Final/]				
	2 star	tup.s [W	500_SDK_G	3.0Final/]				
	3 wm_	main.c [W600_SDK	G3.0Final/	./sys]			
	4 wm_	cpu.c [w	600_sdk/pl	atform/driv	ers]			
	Exit							



💥 Import			—		×
Select					Ľ
Creates a new Make	efile project in a	a directory contain	ing existing code		
<u>S</u> elect an import sou	urce:				
type filter text					
 General C/C++ C/C++ Execute C/C++ Project Existing code at the code of the code o	able : Settings as Autotools pro as Makefile Pro s	oject ject			
?	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cano	el



💥 New Project			-			×
Import Existing Co	ode					
Create a new Make	file project from e	xisting code in	that same direc	tory		
Project Name						
WM_SDK						
Existing Code Locati	on				_	_
C:\workdir\WM_SDI	<				Brov	wse
Languages ☑ C ☑ C++						
Toolchain for Indexe	er Settings					
<none> Cross GCC Cygwin GCC GNU Autotools Too Microsoft Visual C+</none>	lchain + ole toolchains tha	t support this p	blatform			
0	< <u>B</u> ack	<u>N</u> ext >	<u> </u>		Cance	I

After the SDK has imported, the compiler script directory should be redirected to the directory where Makefileis located. Followings are operation steps:



X C/C++ -	Eclipse					
File Edit Source Refactor Navigate Search Project Run Window						
₫ ▼ ☎ ▼ ₫	▼ :::::::::::::::::::::::::::::::::::					
Project E>						
→ [@] WM_SD	New > Go Into					
	Open in New Window					
	 Copy Paste Delete Remove from Context Source Move 					
	Rename F2					
	 Import Export 					
	Clean Project Refresh Close Project Close Unrelated Projects					
	Make Targets >					
	Index > Build Configurations >					
	Show in Remote Systems view Profiling Tools					
	Run As >					
	Debug As >>					
	Restore from Local History					
	Team					
	Compare With					
	Properties Alt+Enter					



X Properties for WM_SDK		_		×		
type filter text	C/C++ Build		< ▼	⇒ ▼ ▼		
 Resource Builders C/C++ Build C/C++ General Linux Tools Path Project References 	Configuration: Default [Active] V	1anage Co	nfigurat	ions		
	Builder Settings Behavior Refresh Policy					
Run/Debug Settings	Builder					
WikiText	$\mathbf{\nabla}$ Use default build command					
	Build command: make		Variabl	es		
	Makefile generation Generate Makefiles automatically Expand Env. Varia	able Refs ir	n Makefi	les		
	Build location					
	Build directory: \${workspace_loc:/WM_SDK}/					
	Workspace File	system	Variable	: S		
	Restore	<u>D</u> efaults	Арр	ly		
0	ОК		Cance	4		



X Folder selection	-		×
Select a folder from workspace:			
 RemoteSystemsTempFiles WM_SDK App Bin Demo Doc Include Lib Platform Src Tools GNU Keil Makeimgsource download.py library.zip makeimg makeimg 			~
(?)	ОК	Cancel	



X Properties for WM_SDK				×		
type filter text	C/C++ Build		(⇒ ▼ 0	⇒ ▼ ▼		
 Resource Builders C/C++ Build C/C++ General 	Configuration: Default [Active] V	∕lanage Co	nfigurati	ons		
Linux Tools Path Proiect References	Builder Settings Behavior Refresh Policy					
Run/Debug Settings	Builder					
> Task Repository	Builder type: External builder			\sim		
WikiText	☑ <u>U</u> se default build command			_		
	Build <u>c</u> ommand: make		Variable	S		
	Makefile generation	able Refs ir	n Makefil	es		
	Build location					
	Build <u>d</u> irectory: \${workspace_loc:/WM_SDK/Tools/GNU}					
	Workspace File	system	Variable	s		
	Restore	<u>D</u> efaults	<u>A</u> ppl	у		
0	OK		Cancel			

2.5 Config OpenOCD to Start Up

Following is how to start up OpenOCD with Eclipse. The advantage of this operation is that users don't have to open a Cygwin window to start OpenOCD.



 \times

C/C++ - Eclipse



Double-click Create:

X External Tools Configurations

Create, manage, and run configurations

Run a program

	Name OpenOCD				
type filter text	Main & Refresh	Build Environment Commo	חנ		
~ 💁 Program	docation:				
⁰ OpenOCD		a) use local bin opened ave			
		Browse Workspace	Brows <u>e</u> File System	Var <u>i</u> ables	
	Working Directory:				
		Browse Workspace	Browse File System	Varia <u>b</u> les	
	Arguments:				
	-f /usr/local/share/c /usr/local/share/ope	penocd/scripts/board/w600_c enocd/scripts/	cmsis-dap.cfg -s	^	
				~	
				Variable <u>s</u>	
	Note: Enclose an argument containing spaces using double-quotes (").				
Filter matched 2 of 2 items			Revert	Apply	
0			Run	Close	



X External Tools Configurations



Create, manage, and run configurations

Run	a	pr	og	ra	m
-----	---	----	----	----	---

type filter text • • Program • OpenOCD	Name: OpenOCD ■ Main ■ Refresh ■ Build ■ Environment ■ Common ■ Build before launch ● The entire workspace ● The project containing the selected resource ● Specific projects	P <u>r</u> ojects
	☐ Include referenced projects	
Filter matched 2 of 2 items	Revert	Apply
3	Run	Close

OpenOCD configuration files should be filled out according to the type of JTAG simulator. Different JTAGE simulator has different configuration files. We have provided with JLINK and CMSIS-DAP.

The configuration files are in the directory of /usr/local/share/openocd/scripts/board/ in Cygwin with the names W60X_jlink.cfg and W60X_cmsis-dap.cfg.

Following example is for using CMSIS-DAP:



X External Tools Configurations



Create, manage, and run configurations

Run a program

	Name: OpenOCD					
type filter text						
• • Program	Location:					
- openoeb	C:\W600_IDE\cygwin\usr	local\bin\openocd.exe				
		Browse Workspace	Brows <u>e</u> File System	Var <u>i</u> ables		
	Working <u>D</u> irectory:					
		Browse Workspace	Browse File Syste <u>m</u>	Varia <u>b</u> les		
	Arguments:					
	-f /usr/local/share/openc /usr/local/share/openocc	ocd/scripts/board/w600_ d/scripts/	cmsis-dap.cfg -s	<		
				Variable <u>s</u>		
	Note: Enclose an argume	nt containing spaces usir	ng double-quotes (").			
Filter matched 2 of 2 items			Re <u>v</u> ert	Apply		
0			Run	Close		

Click Run to start OpenOCD. After starting up successfully, following information will display in console window:



Console
Alternative Action Ac OpenOCD [Program] C:\W600 IDE\cygwin\usr\local\bin\openocd.exe Open On-Chip Debugger 0.10.0+dev-00577-gea41048-dirty (2018-11-21-13:39) Licensed under GNU GPL v2 For bug reports, read http://openocd.org/doc/doxygen/bugs.html adapter speed: 200 kHz adapter speed: 1000 kHz adapter_nsrst_delay: 100 none separate cortex_m reset_config sysresetreq Info : Listening on port 6666 for tcl connections Info : Listening on port 4444 for telnet connections Info : CMSIS-DAP: SWD Supported Info : CMSIS-DAP: JTAG Supported Info : CMSIS-DAP: FW Version = 2.0.0 Info : CMSIS-DAP: Interface Initialised (SWD) Info : SWCLK/TCK = 1 SWDIO/TMS = 1 TDI = 0 TDO = 1 nTRST = 0 nRESET = 1 Info : CMSIS-DAP: Interface ready Info : clock speed 1000 kHz Info : SWD DPIDR 0x2ba01477 Info : w600.cpu: hardware has 6 breakpoints, 4 watchpoints Info : Listening on port 3333 for gdb connections

If the SWD debugging function is disable in W60X SDK, the W60X will be unrecognized during starting OpenOCD. Please modify the firmware and try again.

Then clik following icon to start:



Run	Window Help		
	Resume	F8	🤮 🕶 💁 🕶 😂 🥔 🕶 💷 🔳 🍠 🤤
	Suspend	Ctrl+F2	onsole.h 🗈 wm_main.c 🗈 wm_dem
N.9	Disconnect	Carrie	μ
6 N	Stop Into	55	:puDiv;¤¶
.v.	Step Into	F5 F6	
~	Step Over	F0 E7	<pre>s_reg_read32(HR_CLK_DIV_CTL);</pre>
_0 <u>%</u>	Step Keturn Bus ta Lina	Chulu D	gValue>>4)&0x0F;¤9
-1	Kun to Line	Ctri+K	1XFFFFF000;¤"
-90°	Use Step Filters		1×80000000;
Q,	Run	Ctrl+F11	
楤	Debug	F11	CLK_40M:¤9
₿	Profile		v·=-4;¤¶
	Profile History	>	c;¤9
	Profile As	>	
	Profile Configurations		9 w. =. 2. m
	Due Uletere	、 、	······································
	Run As	· · · · · · · · · · · · · · · · · · ·	
	Run Configurations		(wlanDiv*4/cpuDiv)<<8) (wla
	Kun coniigurations		<pre>32(HR_CLK_DIV_CTL, RegValue);</pre>
	Debug History	>	
	Debug As	>	
	Debug Configurations		
0	Toggle Breakpoint	Ctrl+Shift+B	
0	Toggle Line Breakpoint		
0	Toggle Method Breakpoint		··»This·function·is·used·to·g
65	Toggle Watchpoint		
X	Skip All Breakpoints	Ctrl+Alt+B	syscik» point-to-the-addr-
X	Remove All Breakpoints		• • »None¤¶
	Breakpoint Types	>	
Q	External Tools	>	💁 1 OpenOCD

2.6 Config Eclipse Debugging Function

Right click on the SDK in project management window, choose Debug Configurations:



X C/C++ - Eclipse

<u>File</u> Edit So	ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch	<u>P</u> roject <u>R</u> un <u>W</u> indow	
₫ ▼ 63 ▼ 6	▼ 🔜 🍋 🍕 ▼ 🍬 🕨 II = M 2. O 1.4 ₹ 🕱	\$ ▼ % ▼ <i>A</i> ▼ E m 2 ▼ 9 ▼ + 4	• • •
Project Ex	plorer ¤		
→ [©] WM_SD	ж		
	New	>	
	Go Into		
	Open in New Window		
1	Сору	Ctrl+C	
1	Paste	Ctrl+V	
3	Delete	Delete	
	Remove from Context	Ctrl+Alt+Shift+Down	
	Source	>	
	Nove	52	
	Rename	F2	
n n n n n n n n n n n n n n n n n n n	Import		
	Export		
	Build Project		
a	Refrech	FF	
	Close Project	F3	
	Close Unrelated Projects		
	Make Targets		
	Index	í.	
	Build Configurations	>	
	Show in Romoto Systems view		
	Brofiling Tools		
	Run As		
		× □ 1	Local C/C++ Application
	Profile As		abua Configurations
	Frome As		ebug configurations

Double-click **Zylin Embedded debug (Native)** to create a new debugging configuration:



Beijing Winner Microelectronics Co., Ltd.

X Debug Configurations





C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugger C/C++ Remote Application Launch Group Zylin Embedded debug (Cygwin) Zylin Embedded debug (Native)	 Configure launch settings from this dialog: Press the 'New' button to create a configuration of the selected type. Press the 'Duplicate' button to copy the selected configuration. Press the 'Delete' button to remove the selected configuration. Press the 'Filter' button to configure filtering options. Edit or view an existing configuration by selecting it. Configure launch perspective settings from the 'Perspectives' preference page.
0	Debug Close

The name should be same with the project so as to distinguish other projects:



X Debug Configurations

Create, manage, and run configurations



	Name: WM_SDK Default				
type filter text	🖻 Main 🔅 Debugger 🗖 Cor	mmands 🖗 Source ब Environme	nt "		
$\Box C/C++$ Application	Project (optional):				
$\Box C/C++$ Postmortem Debugger	WM SDK	WM SDK			
© C/C++ Remote Application	C/C++ Application:				
► Launch Group		Search Proiect	Browse		
 R Zylin Embedded debug (Cygwin) Zylin Embedded debug (Native) WM_SDK Default 	Application console				
Filter matched 8 of 9 items		Re <u>v</u> ert	Apply		
0		Debug	Close		

The debugger in use is arm-none-eabi-gdb. The cross compiling tools have been installed in the integrated package with the directory /opt/arm-none-eabi-gcc/ in Cygwin.

Following is the configuration environment:



Beijing Winner Microelectronics Co., Ltd.

X Debug Configurations

Create, manage, and run configurations



	Name: WM_SDK Default				
type filter text	■ Main * Debugger ■ Commands Source ■ Environment □ <u>C</u> ommon				
 C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugger C/C++ Remote Application Launch Group Zylin Embedded debug (Cygwin) Zylin Embedded debug (Native) WM_SDK Default 	Imain * Debugger Commands * Source * Environment * Common Debugger: Embedded GDB Stop on startup at: main Debugger Options Main GDB debugger: arm-none-eabi-gdb Browse Browse GDB command file: Browse (Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) GDB command set: GDB command set: Standard ~ Protocol: mi ~ Verbose console mode Use full file path to set breakpoints	^			
Filter matched 8 of 9 items	Revert Apply Debug Close	~			

💥 Debug Configurations

×	
15-1	
5	
2	

Create, manage, and run configurations

	Name: WM_SDK Default
type filter text C/C++ Application C/C++ Attach to Application C/C++ Postmortem Debugger C/C++ Remote Application Launch Group Zylin Embedded debug (Cygwin) % Zylin Embedded debug (Native) WM_SDK Default	Name: WM_SDK Default Main Debugger Commands Source Environment Common Help/tips on how to setup GDB init script Initialize' commands monitor reset halt monitor fash write_image erase /cygdrive/c/workdir/WM_SDK/Bin/W600_DBG.img 0x08010000 V 'Run' commands
Filter matched 8 of 9 items	Revert Apply Debug Close



The command for initial is:

target remote localhost:3333 monitor reset halt monitor flash write_image erase /cygdrive/c/workdir/WM_SDK/Bin/W60X_DBG.img 0x08010000 file ./Tools/GNU/W60X.elf

Address of W60X's debugging firmware is from 0x08010000 in Flash. So please don't destroy the data in Flash.

The two paths in the command should be filled by actual project paths. The path of W60X_DBG.img should be the absolute path. Symbol table file W60X.elf should be the relative path.

Others can be used with default values, and then click Apply to save the configurations.

2.7 Compiling SDK Source Code with Optimization Level

In order to use symbol tables for single-step debugging, the optimization should be modified from -Os to -g:



C/C++ - Eclipse _ _

<u>File Edit Source Re</u>	efa	c <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>P</u> roje	ect <u>R</u> un	<u>W</u> indow		
@ ▼ @ ▼ @ ▼	«, ·	▼ ≥ D = N D ⊃ D = 3 → 4 = 0	• • 🛷 🔻	■ m 2 ▼ ₩ ▼ ← ▼	⇒ ▼	7
Project Explorer ≈ ✓ ♥ WM_SDK > ⋈ App > ⋈ Bin		B ¢				
> 🖻 Demo > 🖻 Doc		New			>	
> 🛎 Include > 🗠 Lib		Open With				Text Editor
> ≥ Platform > ≥ Src • ≥ Tools		Copy Paste Delete		Ctrl+C Ctrl+V Delete		System Editor In-Place Editor Default Editor
> 🖻 GNU > 🖻 Keil > 🖻 makeimosol	रू द्वी	Remove from Context Mark as Landmark Move	C	Ctrl+Alt+Shift+Down Ctrl+Alt+Shift+Up		Other
download.p		Rename		F2		
 IIbrary.zip makeimg makeimg al 	24 24	Import Export				
makeimg_al	\$	Refresh		F5	_	
makeimg_dl		Make Targets			>	
iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii		Show in Remote Systems v Profiling Tools	iew		>	
makeimg_fis		Run As Debug As			>	
makeimg.py python34.dl	***	Profile As			>	
readme.txt requiremen ⁻	~	Team			>	
li rules.mk li subdir.mk		Compare With Replace With			> >	
test.bin	7	Properties		Alt+Enter	ie ¤	



X C/C++ - WM_SDK/Tools/toolchain.def - Eclipse

 $\underline{\mathsf{F}}{\mathsf{ile}} \ \underline{\mathsf{E}}{\mathsf{dit}} \ \underline{\mathsf{S}}{\mathsf{ource}} \ \mathsf{Refact}{\mathsf{or}} \ \underline{\mathsf{N}}{\mathsf{avigate}} \ \underline{\mathsf{Search}} \ \underline{\mathsf{P}}{\mathsf{roject}} \ \underline{\mathsf{R}}{\mathsf{un}} \ \underline{\mathsf{W}}{\mathsf{indow}}$

(2) ▼ 62 ▼ [2] ♥ [2] % % ▼ [≥ 10 6 2.0 4 = 32	徐 ▼ 🂁 ▼ 永 ▼ 圓 面 全 ▼ 🖗 ▼ 🍄 🗢 ▼
Project Explorer ≅	🖻 🛸 🎽 🗖 🗈 🛸 toolchain.def 🛛
∽ ₽ WM SDK	71
	72 #
> 🖻 Bin	73# Complier options
> 🖻 Demo	74#
> 🖻 Doc	75
> 🖻 Include	76 <mark>CXX_optimization = -g</mark>
> 🖕 Lib	77
> 🖻 Platform	78ifeq (\$(TOOL_GNU),1)
> > Src	79 CFLAGS := -Wall \
✓ @ Tools	80 -DGCC_COMPILE=1 \
> > GNU	81 -mthumb \
> 🖕 Keil	82 \$(CXX_optimization) \
> e makeimosource	83Tunction-sections \
A download py	
library zip	
nakeima	87 -mabi=aapcs
nakeing all	88 -march=armv7-m \
makeing all exe	89 -fno-builtin
makeing_dhexe	90 ARMCFLAGS := -Wall \
makeing_dbg	91 -DGCC_COMPILE=1 -DWM_W600=1 \
makeing_dbg.exc	92 -mthumb \
makeing_abg.py makeing_fls.py	93 \$(CXX_optimization) \
makeing_ns.py	94function-sections \
makeing.exc	95data-sections \
nukeing.py	96 -mcpu=cortex-m3 \
pythons=.un	97 -std=gnu99 \
requirements txt	98 -march=armv7-m \
Requirements.txt	99 -mabi=aapcs \
subdir mk	100 -tno-builtin
a toct hin	101 ASMFLAGS := -Wall \
e teolchain def	102 -mthumb-interwork \
	103 -mtnumb \

Save file to complete the change.

2.8 Compile SDK



Х

X C/C++ - Eclipse <u>File Edit Source Refactor Navigate Search Project Run Window</u> □ 🔄 💝 🔽 🗖 Project Explorer ≈ > 🗗 WM_SDK New > Go Into Open in New Window Copy Ctrl+C Paste Ctrl+V × Delete Delete Remove from Context Ctrl+Alt+Shift+Down Source > Move... Rename... F2 ≥ Import... ⊾ Export... Build Project **Clean Project** W Build Project Building project... Always run in background Run in <u>B</u>ackground Cancel Details >>

Problems □ Console ≅
CDT Build Console [WM_SDK]
CC ../../App/main.c
CC ../../Platform/Boot/gcc/startup_ARMCM3.S
CC ../../Platform/Boot/gcc/retarget_gcc.o
OBJCP ../../Platform/Boot/gcc/retarget_gcc.o
../Bin/W600.bin: 28.7% -- replaced with ../Bin/W600.bin.gz
secboot_len:375c, app_imglen:4e0b8, total:5c0b8
#@./createimg.sh -e cygwin -r
18:56:50 Build Finished (took 1m:13s.247ms)



2.9 Start to Debug

OpenOCD should be running before start to debug and then doing following operation:

₩ C/C++ -	Eclipse		
<u>File</u> <u>E</u> dit <u>S</u> o	ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch	<u>P</u> roject <u>R</u> un <u>W</u> indow	
🔂 🔻 🚳 💌 🖸	▼	☆ ▼ 9₄ ▼ タ ▼ ■ □ 2 ▼ 8	$\bullet \Rightarrow \bullet \bullet \bullet \bullet$
Project Ex	plorer ¤		
→ [©] WM_SD	K		1
	New Co. Into	>	
	Paste	Ctrl+C	
	Delete	Delete	
<u>s</u>	Remove from Context	Ctrl+Alt+Shift+Down	
	Source	>	
	Move		
	Rename	F2	
2	Import		
	Export		
	Clean Project		
8	Refresh	F5	
	Close Project		
	Close Unrelated Projects		
	Make Targets	>	
	Index	>	
	Build Configurations	>	
	Show in Remote Systems view		
	Profiling Tools	>	
	Run As	>	L
	Debug As	>	1 Local C/C++ Application
	Profile As	>	Debug Configurations



Beijing Winner Microelectronics Co., Ltd.

×

💥 Debug Configurations

Create, manage, and run configurations

	Name: WM_SDK Default		
type filter text	🖻 Main 🔅 Debugger 🛎 Commands 🐓 Source 🛎 En	vironment Common	
C/C++ Application	Project (optional):		
© C/C++ Postmortem Debugger	WM_SDK		Browse
C/C++ Remote Application	C/C++ Application:		
Launch Group		Search Project	Browse
 	Application console		
Filter matched 8 of 9 items		Reyert	Apply
0		Debug	Close

Choose Yes when following prompt appears:

X Erro	rs in Workspace	×
	Errors exist in a required project. Continue launch?	
<u> A</u> lv	vays launch without asking <u>Y</u> es <u>N</u> o	
💥 Conf	ïrm Perspective Switch	×
?	This kind of launch is configured to open the Debug perspective when it suspends.	

This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?

Remember my decision

<u>Y</u>es

<u>N</u>o



After the firmware has updated completely, the following prompt will appear:

💥 Debug - Source not found Eclipse		- 🗆 ×
<u>Eile Edit Navigate Search Project Run Window Help</u>		
C ▼ 30 3 \>0808080858(0 ▼ 0 ▼ 9 ▼ 10 √ 10 ▼ 0 ▼ 0 ▼ 0 ▼		Quick Access 🖻 🗟 C/C++ 🎙 Debug
*Debug = %# I+ * *	Variables ** • Breakpoints ** Registers ** Modules	約46日 활동嶺 태려 주요 미
 QpenOCD [Program] C:W600 [DExcygniviusr/local/bin/openocd.exe WM_SDK Default [Zylin Embedded debug (Native)] Embedded GDB (18/12/10 F41:201) (Suspended) Thread [1] (Suspended) 2 <symbol available="" is="" not=""> 0x0000051e</symbol> 1 <symbol available="" is="" not=""> 0x0fffffff</symbol> 	Name	Value
』 arm-none-eabi-gdb (18/12/10 下午12:00)		
	<	>
		~ v
toolchain.def		#Outline #
No source available for ""	,	An outline is not available.
Console # @Tasks & Problems @Executables @ Memory		
WM_SDK Default [Zylin Embedded debug (Native)] arm-none-eabi-gdb (18/12/10 下午12:00)		
target halted due to debug-request, current mode: Thread xPSR: 0x01000000 pc: 0x0000051e msp: 0x20030cd8		^
<pre>monitor riasn write_image erase /cygdrive/c/workdir/WM_SDK/Bin/W600_DBG.img 0x auto erase enabled wrote 32584 bytes from file /cygdrive/c/workdir/WM_SDK/Bin/W600_DBG.img in 70</pre>	.457237s (4.485 KiB/s)	
<pre>file ./Tools/GNU/W600.elf Reading symbols from ./Tools/GNU/W600.elfdone.</pre>		v
K		>

Click F8 to execute Resume command, and the icon will stop at main():



X Debug - WM_SDK/Platform/Sys/wm_main.c - Eclipse		- 🗆 X
Eile Edit Source Refactor Navigate Search Project Run Window Help		
(2) ▼ (3) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2		Quick Access 🕫 🗟 C/C++ 🌣 Debug
Debug =:	■ Variables = % Breakpoints = Registers = Modules Name <	Value
<pre>toolchain.def wwm_main.c # 167 wm_spi_do_config(WM_IO_PB_18); 168 169 170+int main(void) 171 { 172 SystemInit(); 173 174 tls_sys_clk_set(CPU_CLK_80M); 175 176 tls_os_init(NULL); 177 178 /* before use malloc() function, must create mutex used by c_lib */ 179 tls_os_sem_create(&libc_sem, 1); 180 181 182 { 183 tls_os_task_create(NULL, NULL, </pre>	• • • • • • • • • • • • • • • • • • •	ine PRANC
Console # Tasks # Problems 9 Executables Memory		
WM SDK Default [Zylin Embedded debug (Native)] arm-none-eabi-gdb (18/12/10 下午12:10) Reading symbols from ./Tools/GNU/W600.elfdone. b main Breakpoint 1 at 0x8010fc6: file wm_main.c, line 172. Note: automatically using hardware breakpoints for read-only addresses. Breakpoint 1, main () at wm_main.c:172 172 SystemInit();		
<		*

Now users can start to degug.

Other debugging command can be used in the Run menu:



Beijing Winner Microelectronics Co., Ltd.

<u>R</u> un	<u>W</u> indow <u>H</u> elp	
i⇒	Instruction Stepping Mode	
ð	Move to Line (C/C++)	
ъ	Resume at Line (C/C++)	
	Resume	F8
	Suspend	
	Terminate	Ctrl+F2
59	Disconnect	
—0 ⊚⊉	Resume Without Signal	
3	Step Into	F5
P	Step Over	F6
.Ē	Step Return	F7
⇒[Run to Line	Ctrl+R
Ъ,	Use Step Filters	
Q	Run	Ctrl+F11
첺	Debug	F11
	Run History	>
	Run As	>
	Run Configurations	
	Debug History	>
	Debug As	>
Ì	Debug Configurations	
0	Toggle Breakpoint	Ctrl+Shift+B
۲	Toggle Line Breakpoint	
•	Toggle Method Breakpoint	
65	Toggle Watchpoint	
æ	Skip All Breakpoints	Ctrl+Alt+B
X	Remove All Breakpoints	
	Breakpoint Types	>
9	External Tools	>

Users can click on toolbar icon for shortcut operation. Please refer to chapter 2.2 Install Eclipse.

2.10 Stop Debugging

For stopping debugging, users can stop gdb or stop OpenOCD.



X Debug - WM_SDK/Platform/Sys/wm_main.c - Eclipse		
Eile Edit Source Refactor Navigate Search Project Run Wind	ow <u>H</u> elp	
리 두 회원 ()()() · · · · · · · · · · · · · · · ·	$\bullet \odot \bullet \bullet \bullet \bullet \bullet \bullet$	
* Debug ∺		*** * * * • •
Question of the second seco	stop openocd	
 [∞] WM_SDK Default [Zylin Embedded debug (Native)] [∞] Embedded GDB (18/12/10 下午12:11) (Suspended) [∞] Thread [1] (Suspended) ⁼ 1 main() wm_main.c:176 0x08010fd0 [,] arm-none-eabi-gdb (18/12/10 下午12:10) 	stop gdb debug	
		¢

2.11 Compile Release Firmware

The optimization level should be -Os when compiling formal firmware:



₩ C/C++ - WM_SDK/Tools/toolchain.def - Eclipse

<u>File Edit Source Refactor Navigate Search Project Run Window</u>

🖆 🔻 🛱 🔻 📴 🐂 🐘 🗞 💌 🔌 🕨 🖬 🖉 20. o. 1. 🗮 🎇	☆ ▼ % ▼ [∅ ▼]0 □ 10 ▼ 0 ▼ 0 ▼ 0 ▼
Project Explorer ≅	🖻 🛸 🔍 🔍 🗎 Ioolchain.def 🛛
∽ ₽ WM SDK	72#
	73# Complier options
> 🗠 Bin	74 #
> 🖻 Demo	75
> 🖻 Doc	7 <mark>6</mark> CXX_optimization = -Os
> 🖻 Include	77
> 🖻 Lib	78ifeq (\$(TOOL_GNU),1)
> 🔊 Platform	79 CFLAGS := -Wall \
> 🖻 Src	80 -DGCC_COMPILE=1 \
~ ₱ Tools	81 -mthumb \
> 🖻 GNU	82 \$(CXX_optimization) \
> 🖕 Keil	83Tunction-Sections \
> 🖻 makeimgsource	
■ download.py	
■ library.zip	87 -mabi=aapcs
■ makeimg	88 -march=armv7-m \
■ makeimg all	89 -fno-builtin
🖩 makeimg all.exe	90 ARMCFLAGS := -Wall \
■ makeimg dbg	91 -DGCC_COMPILE=1 -DWM_W600=1 \
■ makeimg dbg.exe	92 -mthumb \
■ makeimg_dbg.py	93 \$(CXX_optimization) \
■ makeimg_fls.py	94function-sections \
■ makeimg.exe	95data-sections \
■ makeimg.py	96 -mcpu=cortex-m3 \
■ python34.dll	97 -std=gnu99 \
readme.txt	98 -march=armv7-m \
requirements.txt	99 -mabl=aapcs \
₽ rules.mk	
🕞 subdir.mk	$\frac{101}{102} = \frac{101}{102} = \frac{101}{102}$
🗈 test bin	103 -mthumb \
🗈 toolchain.def	104 - std=gnu99 \
	The sea-grass (

Users can decide to enable or disable SWD function by themselves (call the function wm_swd_config(0)).

2.12 Update Firmware through UART Port

W60X firmware can be updated through UART port. Users can refer to < WM_W60X_Firmware Update Guide.pdf> from <u>http://www.winnermicro.com/html/1/156/158/497.html</u>.

3 OpenOCD Using Guide

3.1 OpenOCD Introduction

OpenOCD is an open source JTAG program in host computer. OpenOCD can connect to embedded device through JTAG simulator and can provide general debugging commands for host computer.





OpenOCD has supported following JTAG simulators:

AICE, ARM-JTAG-EW, ARM-USB-OCD, ARM-USB-TINY, AT91RM9200, axm0432,

BCM2835, Bus Blaster, Buspirate, Chameleon, CMSIS-DAP, Cortino, DENX,

Digilent JTAG-SMT2, DLC 5, DLP-USB1232H, embedded projects, eStick,

FlashLINK, FlossJTAG, Flyswatter, Flyswatter2, Gateworks, Hoegl, ICDI,

ICEBear, J-Link, JTAG VPI, JTAGkey, JTAGkey2, JTAG-lock-pick, KT-Link,

Lisa/L, LPC1768-Stick, MiniModule, NGX, NXHX, OOCDLink, Opendous,

OpenJTAG, Openmoko, OpenRD, OSBDM, Presto, Redbee, RLink, SheevaPlug

devkit, Stellaris evkits, ST-LINK (SWO tracing supported),

STM32-PerformanceStick, STR9-comStick, sysfsgpio, TUMPA, Turtelizer,

ULINK, USB-A9260, USB-Blaster, USB-JTAG, USBprog, VPACLink, VSLLink, Wiggler, XDS100v2, Xverve

OpenOCD has supported following CPU core:

ARM11, ARM7, ARM9, AVR32, Cortex-A, Cortex-R, Cortex-M, LS102x-SAP, Feroceon/Dragonite, DSP563xx, DSP5680xx, EnSilica eSi-RISC, FA526, MIPS EJTAG, NDS32, XScale, Intel Quark

OpenOCD has supported following Flash:

ADUC702x, AT91SAM, ATH79, AVR, CFI, DSP5680xx, EFM32, EM357, eSi-TSMC, FM3, FM4, Kinetis, LPC8xx/LPC1xxx/LPC2xxx/LPC541xx, LPC2900, LPCSPIFI, Marvell QSPI, Milandr, NIIET, NuMicro, PIC32mx, PSoC4, PSoC5LP, SiM3x, Stellaris, STM32, STMSMI, STR7x, STR9x, nRF51; NAND controllers of AT91SAM9, LPC3180, LPC32xx, i.MX31, MXC, NUC910, Orion/Kirkwood, S3C24xx, S3C6400, XMC1xxx, XMC4xxx

3.2 Install OpenOCD

The operations in this chapter are all based on Cygwin environment in Windows. After the Cygwin has been installed, double-click cygwin.bat to open a Cygwin shell window, all the following operations are



based in this shell window.

According to OpenOCD official documentation requirement, compilation must depend on:

```
You'll also need:

- make

- libtool

- pkg-config >= 0.23 (or compatible)

Additionally, for building from git:

- autoconf >= 2.64

- automake >= 1.14

- texinfo
```

So users need to test the Cygwin environment and install unsupported items.

3.2.1 Install libusb

Some simulators for OpenOCD need libsub. So the libsub should be installed before compiling OpenOCD (W60X has provided this lib).

Installation steps for libusb:

- 1. Download source code package from <u>https://libusb.info</u> and decompress.
- 2. Entry source code menu.
- 3. Execute ./*configure --prefix=/usr/* to generate Makefile.
- 4. Execute *make* compiling.
- 5. Execute *make install*.

3.2.2 Install HIDAPI library

Many users use CMSIS-DAP simulator, so W60X has supported this type of simulator. HIDAPI library should be installed for OpenOCD when users use CMSIS-DAP simulator (W60X has provided this lib).

Installation steps for HIDAPI library:

- 1. Execute git clone <u>https://github.com/signal11/hidapi.git</u> to download source code.
- 2. Entry source code menu.



- 3. Execute ./bootstrap to generate configure file.
- 4. Execute ./configure --prefix=/usr to generate Makefile.
- 5. Exectue *make* to compile.
- 6. Execute *make install*.

3.2.3 Install OpenOCD

1. Download OpenOCD Source Code:

Download source code package from <u>http://www.winnermicro.com</u>. Current version is W60X_openocd_0.10.0_r1.

2. Execute ./configure --enable-cmsis-dap --disable-werror to generate Makefile.

If some error occurs, users have to repare the lack of items. The *-enable-cmsis-dap* needn't to be selected if CMSIS-DAP simulator is no use.

Following is the print information after testing:

libjaylink configuration summary: - Package version - Library version	0.2.0-git-86458 0:0:0	345
- Installation pretix	/usr/local	
- Building on	1686-pc-cygwin	
	төөө-рс-судитт	
Enabled transports:		
- USB	ves	
- тср	ýes	
	·	
openocd configuration summary		
MPSSE mode of ETDI based devices	ves (auto)	
ST-Link JTAG Programmer	ves (auto)	
TI ICDI JTAG Programmer	ves (auto)	
Keil ULINK JTAG Programmer	ves (auto)	
Altera USB-Blaster II Compatible	ýes (auto)	
Bitbang mode of FT232R based devices	yes (auto)	
Versaloon-Link JTAG Programmer	yes (auto)	
TI XDS110 Debug Probe	yes (auto)	
OSBDM (JTAG only) Programmer	yes (auto)	
eStick/opendous JTAG Programmer	yes (auto)	
Andes JTAG Programmer	yes (auto)	
USBProg JTAG Programmer	no	
Raisonance RLink JTAG Programmer	no	
OTIMEX ARM-JTAG-EW Programmer	no	
CMSIS-DAP Compiliant Debugger	yes (auto)	
Alterna USB Blaster Compatible	yes (auto)	
ATTERA USB-BLASTER COMPACIBLE	yes (auto)	
OpenITAC Adapter	yes (auto)	
SEGGER 1-Link Programmer	ves (auto)	
Sedder S Ernik Programmer	yes (auto)	



- Execute compiling command to start compilation.
 make
- 4. Execute installation command to install OpenOCD to the system path.

make install

OpenOCD's default installation path is /usr/local/bin, the path for configuration file is /usr/local/share/openocd. A configuration file named W60X.cfg will appeared in the path /usr/local/share/openocd/scripts/target and OpenOCD will use it.

3.3 Debugging with OpenOCD Command Lines

3.3.1 Start up OpenOCD with JLINK Simulator

1. Install driver

JLINK's official driver can't be used for OpenOCD. The usable driver should be download from http://zadig.akeo.ie:





Zadig	– 🗆 X
Device Options Help	
BULK interface (Interface 2)	✓ ☐ Edit
Driver WinUSB (v6. 1. 7600. 16385) WinUSB (v6. 1. 7600. 16385) USB ID 1366 0105 02 WCID ² X	More Information <u>WinUSB (libusb)</u> <u>libusb-win32</u> <u>libusbK</u> <u>WinUSB (Microsoft)</u>
and the contract of the contra	

2. Create configuration file

If the source code package is not downloaded from git, such configuration file may have been installed already in OpenOCD. So following steps can be ignored. Create a new txt file in /usr/local/share/openocd/scripts/board with the name W60X_jlink_cfg.

Open the txt file and input following content:

```
1
    #
    # Example configuration file to hook up an W600 module or board to a JTAG/SWD
 2
 3
    # adapter. Please modify this file to your local setup.
 4
    #
 5
    ŧ
 6
 7
    # Include the configuration for the JTAG adapter. We use the Tian TUMPA here.
 8
    # If you have a different interface, please edit this to include the
 9
10 # configuration file of yours.
    #source [find interface/jlink.cfg]
11
    interface jlink
    # The W600 only supports JTAG.
13
    transport select swd
14
15
16 # The speed of the JTAG interface, in KHz. If you get DSR/DIR errors (and they
    # do not relate to OpenOCD trying to read from a memory range without physical
17
18
    # memory being present there), you can try lowering this.
19
    adapter khz 200
21
    # With no variables set, openood will configure JTAG for the two cores of the W600 and
22
    # will do automatic RTOS detection. This can be be adjusted by uncommenting any of the
   # following lines:
23
24
25 #Source the W600 configuration file
26 source [find target/w600.cfg]
27
```

Save and create a configuration file for OpenOCD connecting JLINK.

3. Execute openocd.exe -f /usr/local/share/openocd/scripts/board/W60X_jlink.cfg -s



/usr/local/share/openocd/scripts/ in shell window of Cygwin. Following information will appear after

OpenOCD starting successfully:

3.3.2 Start up OpenOCS with CMSIS-DAP Simulator

1. Create configuration file

If the source code package is not downloaded from git, such configuration file may have been

installed already in OpenOCD. So following steps can be ignored.

Create a new txt file in /usr/local/share/openocd/scripts/board with the name W60X_cmsis-dap.cfg.

Open the txt file and input following content:

```
1
    #
2
    # Example configuration file to hook up an W600 module or board to a JTAG/SWD
 3
    # adapter. Please modify this file to your local setup.
 4
    #
5
    #
 6
8
   # Include the configuration for the JTAG adapter. We use the Tian TUMPA here.
 9
    # If you have a different interface, please edit this to include the
10 # configuration file of yours.
11 #source [find interface/cmsis-dap.cfg]
   interface cmsis-dap
12
   # The W600 only supports JTAG.
13
14
   transport select swd
15
16
    # The speed of the JTAG interface, in KHz. If you get DSR/DIR errors (and they
17
    # do not relate to OpenOCD trying to read from a memory range without physical
18
    # memory being present there), you can try lowering this.
19
   adapter_khz 200
20
21
    # With no variables set, openocd will configure JTAG for the two cores of the W600 and
2.2
    # will do automatic RTOS detection. This can be be adjusted by uncommenting any of the
23
    # following lines:
24
25
   #Source the W600 configuration file
26 source [find target/w600.cfg]
27
```

Save and create a configuration file for OpenOCD connecting CMSIS-DAP.



4. Execute openocd.exe -f /usr/local/share/openocd/scripts/board/W60X_cmsis-dap.cfg -s

/usr/local/share/openocd/scripts/ in shell window of Cygwin. Following information will appear after OpenOCD starting successfully:

s openord even of /usr/local/share/openord/scripts/board/w600_cmsis_dap_cfg_s_/usr/local/share/op	enocd/scripts/
Dependent are in year of a for an even of the second are the secon	enoed/set ipes/
Licensed under CAUGE CDL VIII (inst)	
bis washtin (/oneodolong/doc/doc/doc/doc/doc/doc/doc/doc/doc/doc	
adapter sport 200 kHz	
adapter speed, 200 kiz a annons accurad	
adapter speed: 1000 km2/e, errors occurred:	
adapter_nsrst_delay: eloo koll//infrd_code/openoco/222/ fibicolle1.4/build/chakee/fies/chakeoucput. To	9 •
none separate	
cortex_m reset_config sysresetreqworkair/inina_coale/openoco/222/inbrdair-1.4/builla	
into : Listening on port 6666 for tcl connections	
into : Listening on port 4444 for teinet connections	
Into:: CMSIS-DAP: SWD/cSupported/workdir/Third_Code/OpenOCD/222/Thbftdil=1.4/build	
Into : CMSIS-DAP: JTAG Supported	
Info : CMSIS-DAP: FW Version = 2.0.0	
Info:: CMSIS-DAP: Interface Initialised (SWD)Code/OpenOCD/222/ hbftdil-1.4/build	
INFO : SWCLK/TCK = 1 SWDIO/TMS = 1 TDI = 0 TDO = 1 NTRST = 0 NRESET = 1	
Info : CMSIS-DAP: Interface ready	
Info:: clock speed 1000 kHzive/e/workdir/Thind_Code/openocD/222/libftdil=1.4/build	
Info : SWD DPIDR 0x2ba01477	
Info : w600.cpu: hardware has 6 breakpoints, 4 watchpoints	
Info:: Listening on port 3333 for gdb/connections/e/openOCD/222/libftdil=1.4/build	

3.3.3 Host Computer Connects with OpenOCD

After starting up successfully, OpenOCD can be connected with telnet or gdb. The port number of telnet is 4444, the port number of gdb is 3333.

Ctrl+C can stop the running of OpenOCD. After OpenOCD starting up, shell window should keep opening to ensure OpenOCD's normal operation and ensure telnet or gdb's normal debugging.

3.3.3.1 Connect by telnet

Windows system comes with telnel client. Users can open a command prompt window and input telnet localhost 4444 or telnet 127.0.0.1 4444 to connect with OpenOCD daemon.





Then users can input commands in command lines:



3.3.3.2 Connect by gdb

The gdb of W60X's cross compiling tool is arm-none-eabi-gdb, and the download address is:

https://launchpad.net/gcc-arm-embedded/4.9/4.9-2014-q4-major.

If user use W60X's integrated package, such tools has already been in the package and can be used directly.

Input command arm-none-eabi-gdb into shell window of Cygwin and the gdb will start up. Then users can

input the debugging commands:



3.3.4 Common Commands

The following table lists some common commands of OpenOCD. Users can also input help command to view all the currently supported commands.

All the commands can be input directly for telnet. For gdb, users should add monitor before the commands.

Target State Handling Commands	
poll	Query current target's status



Beijing Winner Microelectronics Co., Ltd.

halt	Interrupt target's running	
resume [address]	Resume target's running. If the address is specified,	
	programme will run from this address.	
step [address]	Single-step operation. If the address is specified,	
	programme will execute one command from this address.	
reset	Reset target.	
Breakpoint Commands		
bp <addr> <length> [hw]</length></addr>	Set breakpoint at <addr>, command length is <length>,</length></addr>	
	[hw] means hardware breakpoint.	
rbp <addr></addr>	Delete breakpoint at <addr>.</addr>	
Memory Access Commands		
mdw ['phys'] <addr> [count]</addr>	Show the word (4 bytes) counts from physical address.	
mdh ['phys'] <addr> [count]</addr>	Show the short (2 bytes) counts from physical address.	
mdb ['phys'] <addr> [count]</addr>	Show the byte counts from physical address.	
mww ['phys'] <addr> <value></value></addr>	Write a word (4 bytes) to physical address with the	
	<value>.</value>	
mwh ['phys'] <addr> <value></value></addr>	Write a short (2 bytes) to physical address with the	
	<value>.</value>	
mwb ['phys'] <addr> <value></value></addr>	Write a byte to physical address with the <value>.</value>	
flash write_image [erase] [unlock] filename [offset [file_type]]		
Write the firmware's filename to the or	ffset of flash address.	

More detailed commands of OpenOCD can be referred from http://openocd.org/documentation/.

3.3.5 Gdb Debugging Examples

This chapter provides an example with gdb single-step debugging. Let users start to debug with W60X.

Firmware and symbol table are required for single-step debugging. Users can refer to chapter "2.8 Complie

SDK" to generate such files.

In this example, we use CMSIS-DAP simulator to connect with W60X and then start up OpenOCD in Cygwin:





Then create a new Cygwin window and execute arm-none-eabi-gdb:

\$ arm-none-eabi-gdb
GNU gdb (GNU Tools for ARM Embedded Processors) 7.8.1.20141128-c∨s
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http: gnu.org="" gpl.html="" licenses=""></http:>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "host=i686-w64-mingw32target=arm-none-eabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http: bugs="" gdb="" software="" www.gnu.org=""></http:> .
Find the GDB manual and other documentation resources online at:
<http: documentation="" gdb="" software="" www.gnu.org=""></http:> .
For help, type "help".
Type "apropos word" to search for commands related to "word".
(adb)

Then connect the OpenOCD's daemon with target remote localhost:3333:



Input "monitor reset halt" to reset and stop working of CPU:

```
(gdb) monitor reset halt
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000051e msp: 0x20030cd8
(gdb)
```

Then input "monitor flash write_image erase /cygdrive/g/temp/W60X_SDK_G3.0Final/image/W60X.dbg

0x08010000" to update firmware to flash:

```
(gdb) monitor flash write_image erase /cygdrive/g/temp/W600_SDK_G3.0Final/image/w600.dbg 0x08010000
auto erase enabled
wrote 438272 bytes from file /cygdrive/g/temp/W600_SDK_G3.0Final/image/w600.dbg in 82.988007s (5.157 KiB/s)
(gdb) |
```

Then input "file ./image/W60X.out" to load the symbol table:

```
(gdb) file ./image/w600.out
A program is being debugged already.
Are you sure you want to change the file? (y or n) [answered Y; input not from terminal]
Reading symbols from ./image/w600.out...done.
(gdb)
```

Then input "b main" to set a breakpoint, so as to stop at main function:

```
(gdb) b main
Breakpoint 1 at 0x8011c96: file wm_main.c, line 165.
(gdb)
```



At last, input "c" to let programme start running:



Then it will stop at main function:



Then usesrs can input "n" to debug with single-step:

Breakpoint 1, main () at wm_main.c:165	
165 SystemInit();	
(gdb) n	
16/tls_sys_clk_set(CPU_CLK	_80M);
(gdb) n	
109 tis_os_1n1t(NULL);	
(gap) n 172 tls os som croato(&libs som	1).
(adb)	, т),

Input "p libc_sem" and the pointer address will be printed out:

(gdb) p libc_sem \$1 = (tls_os_sem_t *) 0x0

Using "set libc_sem=0x100" to modify the pointer address:



About more detailed debugging skills of gdb, users can refer to the document from http://sourceware.org/gdb/current/onlinedocs/gdb/.



4 Appendix

4.1 Install zylincdt to Eclipse

Using plug-in Zylincdt can support debugging with embedded GDB in Eclipse. The W60X integrated package has provided this plug-in.

This plug-in can be download from http://opensource.zylin.com/zylincdt, following is installation steps:





🖨 Install		_	
Available Software Check the items that you wish to install.			
Work with: http://opensource.zylin.com/zylincdt	d more software by working with the <u>"Availab</u>	∽ Die Software Sites"	Add preferences.
type filter text			
Name	Version		
Select All Deselect All Details			
Zylin Embedded CDT 1.0.0.787S1cKDAnh			ੇ <u>More</u>
Show only the latest versions of available software	Hide items that are already installed		
Group items by category	What is <u>already installed</u> ?		
Show only software applicable to target environment			
☑ Contact all update sites during install to find required software			
?	< Back Next >	Finish	Cancel

Some warning maybe appeare during installation. At last the tools will prompt to reset Eclipse, following the instruction is OK.