W60X OpenOCD 调试指导

V0.1

北京联盛德微电子有限责任公司 (winner micro)

地址：北京市海淀区阜成路 67 号银都大厦 18 层

电话：+86-10-62161900

公司网址：www.winnermicro.com

文档修改记录

版本	修订时间	修订记录	作者	审核
V0.1	2018-11-16	创建	LiLm	

目录

1 引言

1.1 编写目的

指导如何在 Eclipse 环境中集成 OpenOCD 使用 GDB 调试代码；

指导 W60X 相关的开发人员使用 OpenOCD 对 W60X 进行单步调试；

1.2 预期读者

所有 W60X 相关的开发人员

1.3 术语定义

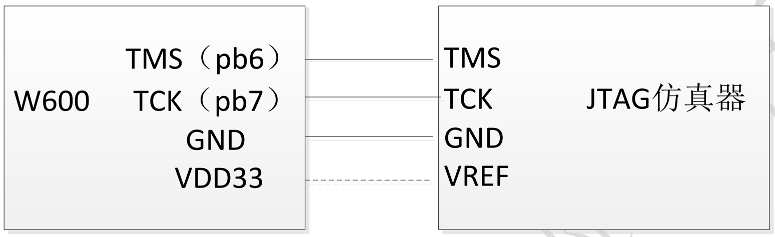
OpenOCD：Open On-Chip Debugger

1.4 参考资料

2 快速上手：使用 Eclipse+OpenOCD 调试 W60X

2.1 连接模块

OpenOCD必须搭配JTAG仿真器使用，选定一款JTAG仿真器（如JLINK、CMSIS-DAP等），使用杜邦线连接JTAG仿真器和W60X模块，其连接方式如下图：



如图所示接线之后，给 W60X 模块上电，JTAG 仿真器连接到上位机（本文所使用的上位机都是 PC 电脑）。

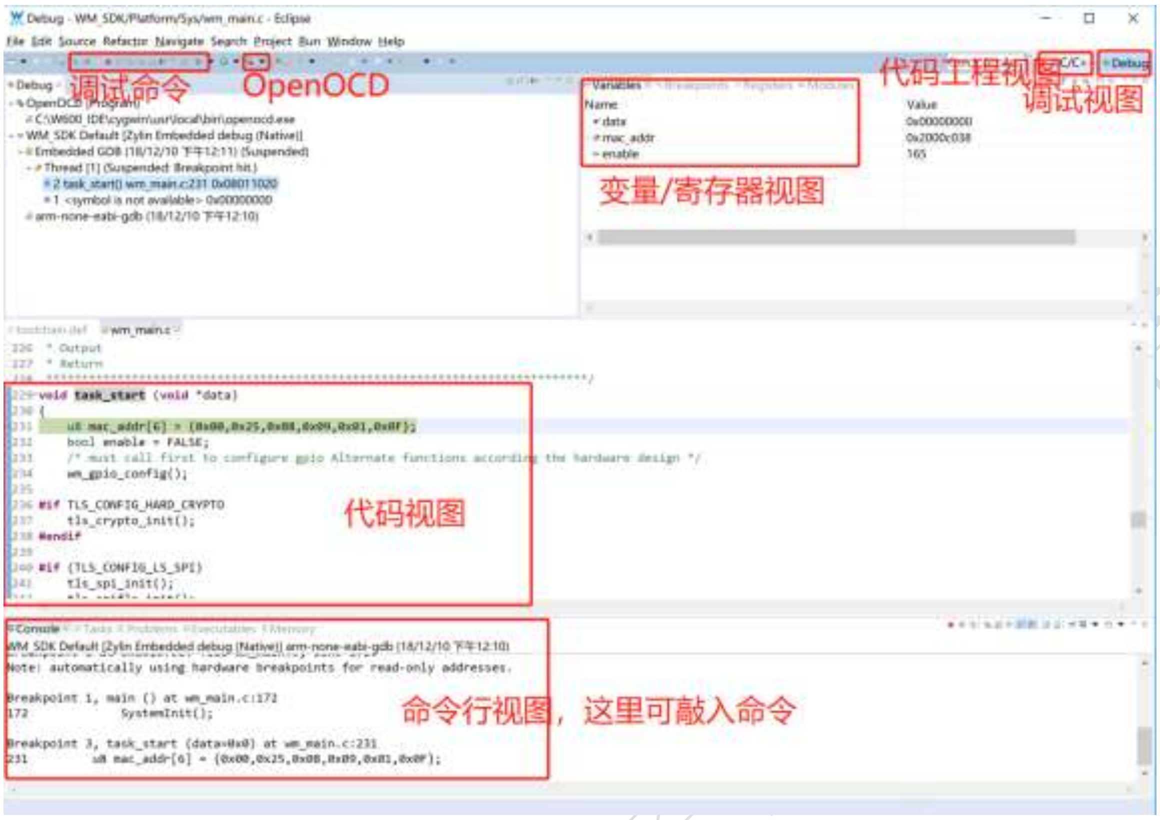
电源接线根据实际情况而定，有些仿真器不需要连接电源线。
不同的 JTAG 仿真器所有使用的驱动不同，如果使用 JLINK 仿真器其驱动安装请参考 3.3.1 章节的安装；如果使用 CMSIS-DAP 仿真器则不需要安装驱动；其他的 JTAG 仿真器请自行安装驱动。

2.2 安装 Eclipse

这里建议使用我们打包配置好的环境，请在 http://www.winnermicro.com/html/1/156/158/497.html 下载，该压缩包里我们提供了 Eclipse（已经集成了 zylincdt 调试插件）、Cygwin（已经安装好了 OpenOCD）、交叉编译工具（arm-none-eabi-gcc）等。

解压压缩包之后，请先阅读压缩包里的 ReadMe.txt 里面的说明，然后通过双击 W60X_IDE.exe 启动我们已经配置好的 Eclipse。

最终工作时的界面将会如下图所示：



```
OpenOCD (Program)
  C:\W600_IDE\cygwin\usr\local\bin\openocd.exe
WM_SDK Default [Zylin Embedded debug (Native)]
  Embedded GDB (18/12/10 下午12:11) (Suspended)
    Thread [1] (Suspended: Breakpoint hit.)
      2 task_start() wm_main.c:231 0x08011020
      1 <symbol is not available> 0x00000000
  arm-none-eabi-gdb (18/12/10 下午12:10)
```

```
 * Output
 * Return
******************************************************************************/
void task_start (void *data)
{
    u8 mac_addr[6] = {0x00,0x25,0x08,0x09,0x01,0x0F};
    bool enable = FALSE;
    /* must call first to configure gpio Alternate functions according the hardware design */
    wm_gpio_config();

#if TLS_CONFIG_HARD_CRYPTO
    tls_crypto_init();
#endif

#if (TLS_CONFIG_LS_SPI)
    tls_spi_init();
```

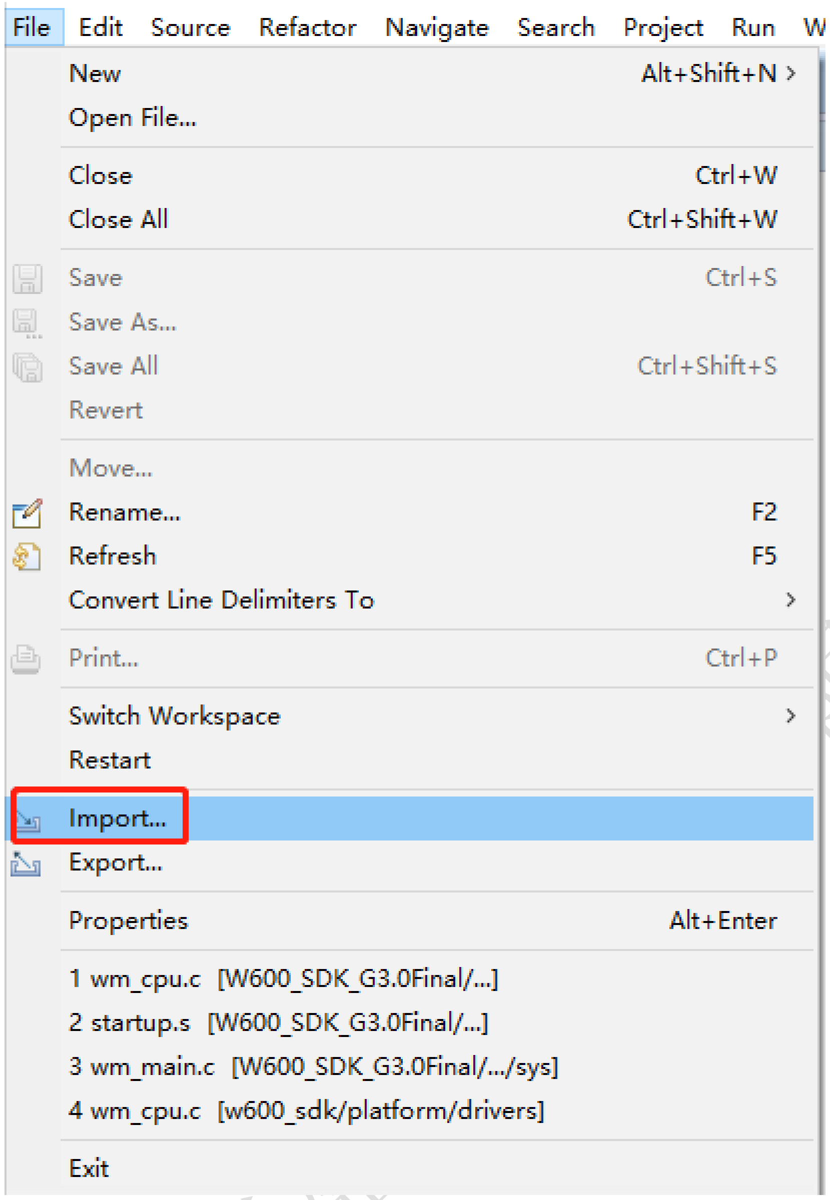
```
WM_SDK Default [Zylin Embedded debug (Native)] arm-none-eabi-gdb (18/12/10 下午12:10)
Note: automatically using hardware breakpoints for read-only addresses.

Breakpoint 1, main () at wm_main.c:172
172             SystemInit();

Breakpoint 3, task_start (data=0x0) at wm_main.c:231
231         u8 mac_addr[6] = {0x00,0x25,0x08,0x09,0x01,0x0F};
```

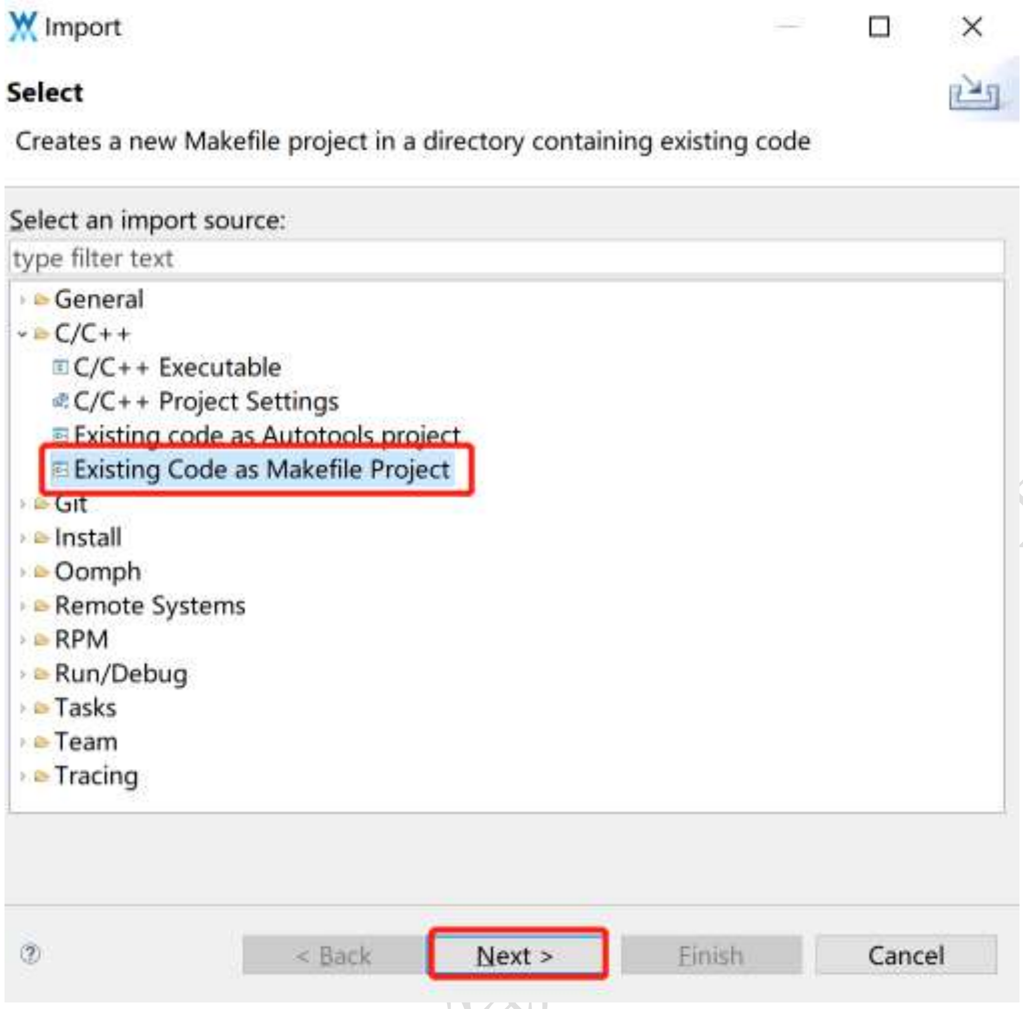
2.3 下载 W60X_SDK 源码

可以在 http://www.winnermicro.com 下载 SDK 包，目前支持 OpenOCD 调试的 SDK 版本为 G3.1 或比 G3.1 更高的版本；

2.4 在 Eclipse 中导入 SDK 工程

导入步骤如下图所示：


File
Edit
Source
Refactor
Navigate
Search
Project
Run
W
New Alt+Shift+N
Open File...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Switch Workspace
Restart
Import...
Export...
Properties Alt+Enter
1 wm_cpu.c [W600_SDK_G3.0Final/...]
2 startup.s [W600_SDK_G3.0Final/...]
3 wm_main.c [W600_SDK_G3.0Final/.../sys]
4 wm_cpu.c [w600_sdk/platform/drivers]
Exit



Import

Select

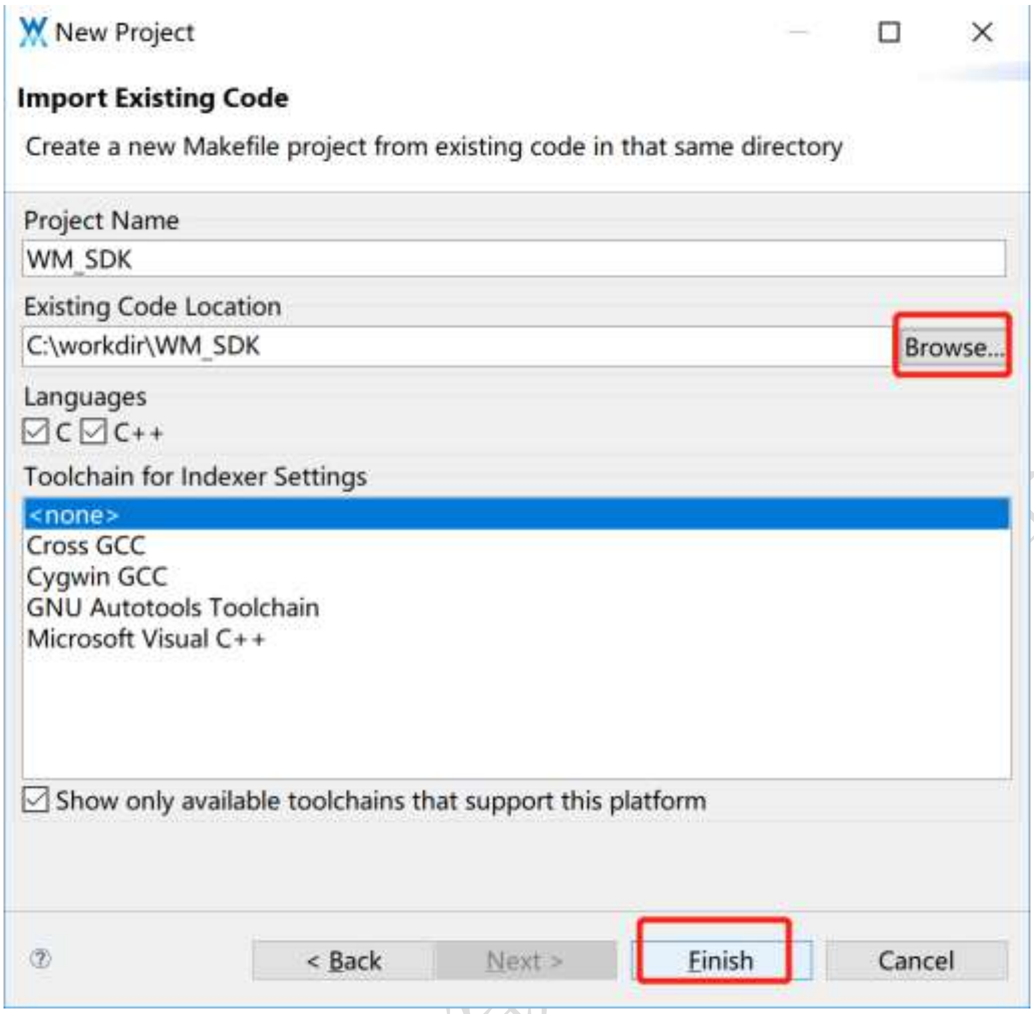
Creates a new Makefile project in a directory containing existing code

Select an import source:

type filter text

- General
- C/C++
  - C/C++ Executable
  - C/C++ Project Settings
  - Existing code as Autotools project
  - Existing Code as Makefile Project
- Git
- Install
- Oomph
- Remote Systems
- RPM
- Run/Debug
- Tasks
- Team
- Tracing

< Back | Next > | Finish | Cancel


New Project

Import Existing Code

Create a new Makefile project from existing code in that same directory

Project Name

WM_SDK

Existing Code Location

C:\workdir\WM_SDK

Languages

☑ C ☑ C++

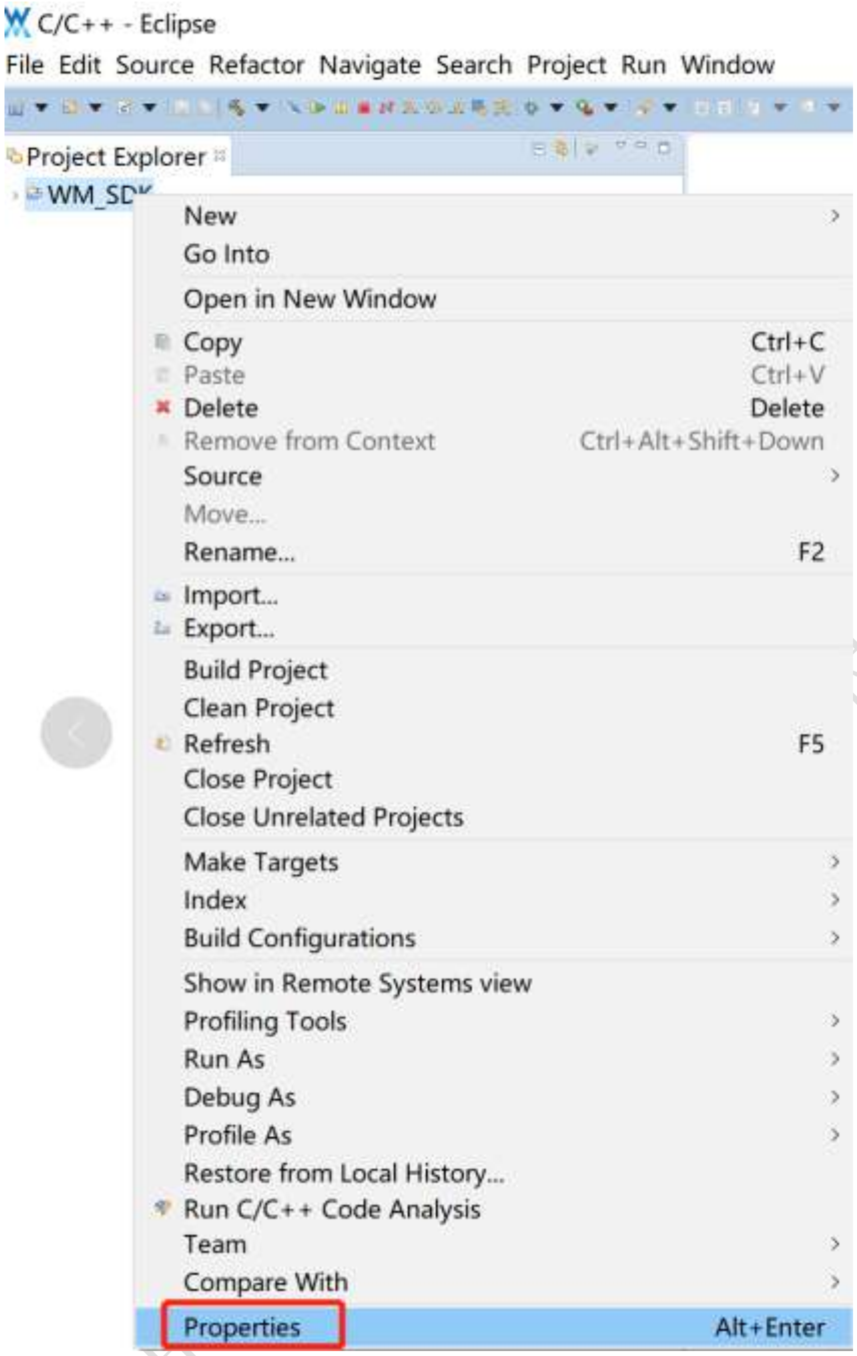
Toolchain for Indexer Settings

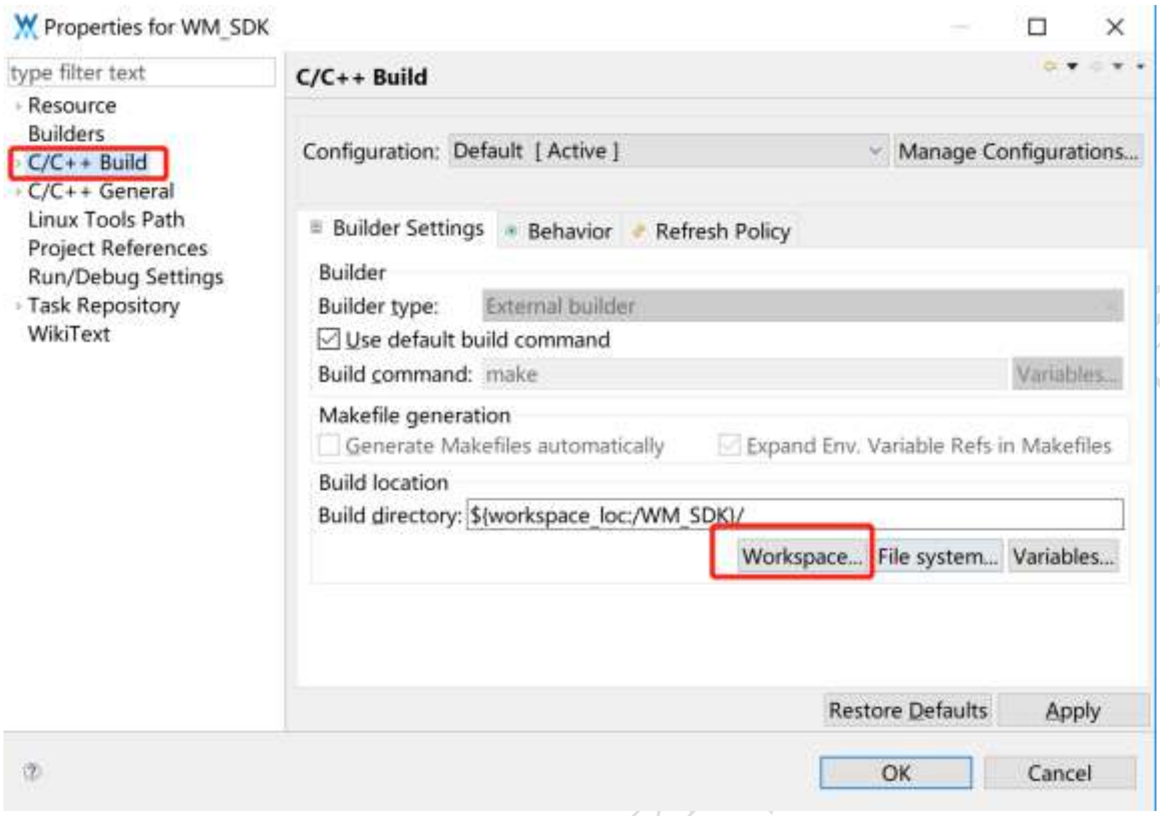
<none>

Cross GCC
Cygwin GCC
GNU Autotools Toolchain
Microsoft Visual C++

☑ Show only available toolchains that support this platform

导入之后，需要将编译脚本目录重定向到 Makefile 所在的目录，请按下图步骤操作：


C/C++ - Eclipse
File Edit Source Refactor Navigate Search Project Run Window
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
Refresh F5
Close Project
Close Unrelated Projects
Make Targets
Index
Build Configurations
Show in Remote Systems view
Profiling Tools
Run As
Debug As
Profile As
Restore from Local History...
Run C/C++ Code Analysis
Team
Compare With
Properties Alt+Enter



Properties for WM_SDK

type filter text

- Resource
- Builders
- C/C++ Build
- C/C++ General
- Linux Tools Path
- Project References
- Run/Debug Settings
- Task Repository
- WikiText

C/C++ Build

Configuration: Default [Active]

Builder Settings | Behavior | Refresh Policy

Builder

Builder type: External builder

☑ Use default build command

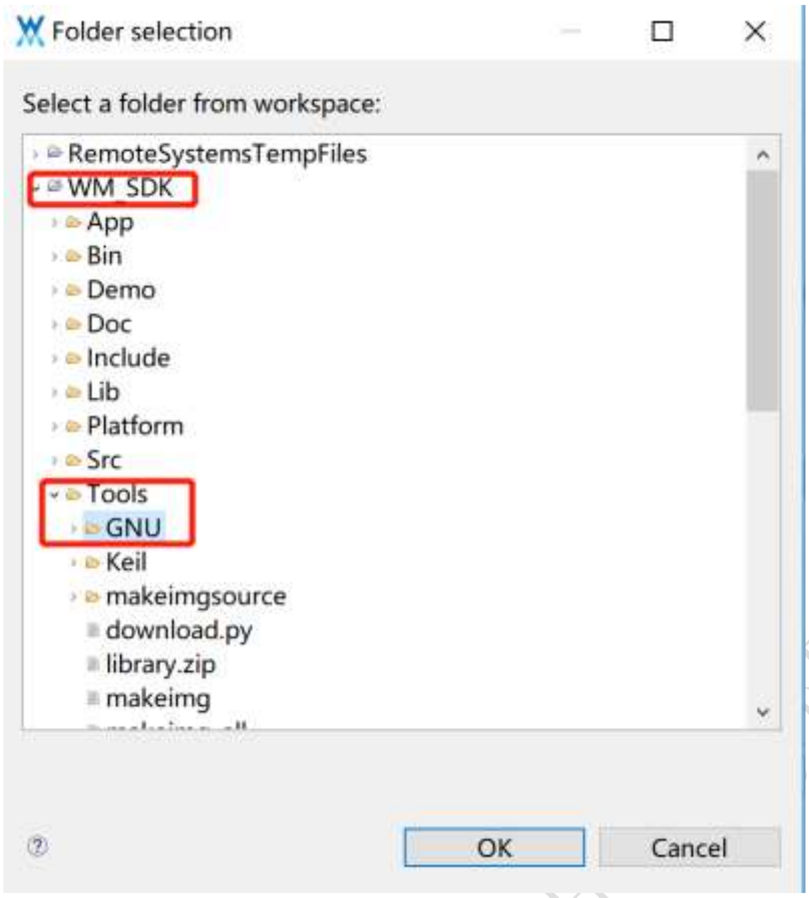
Build command: make

Makefile generation

☐ Generate Makefiles automatically　☑ Expand Env. Variable Refs in Makefiles

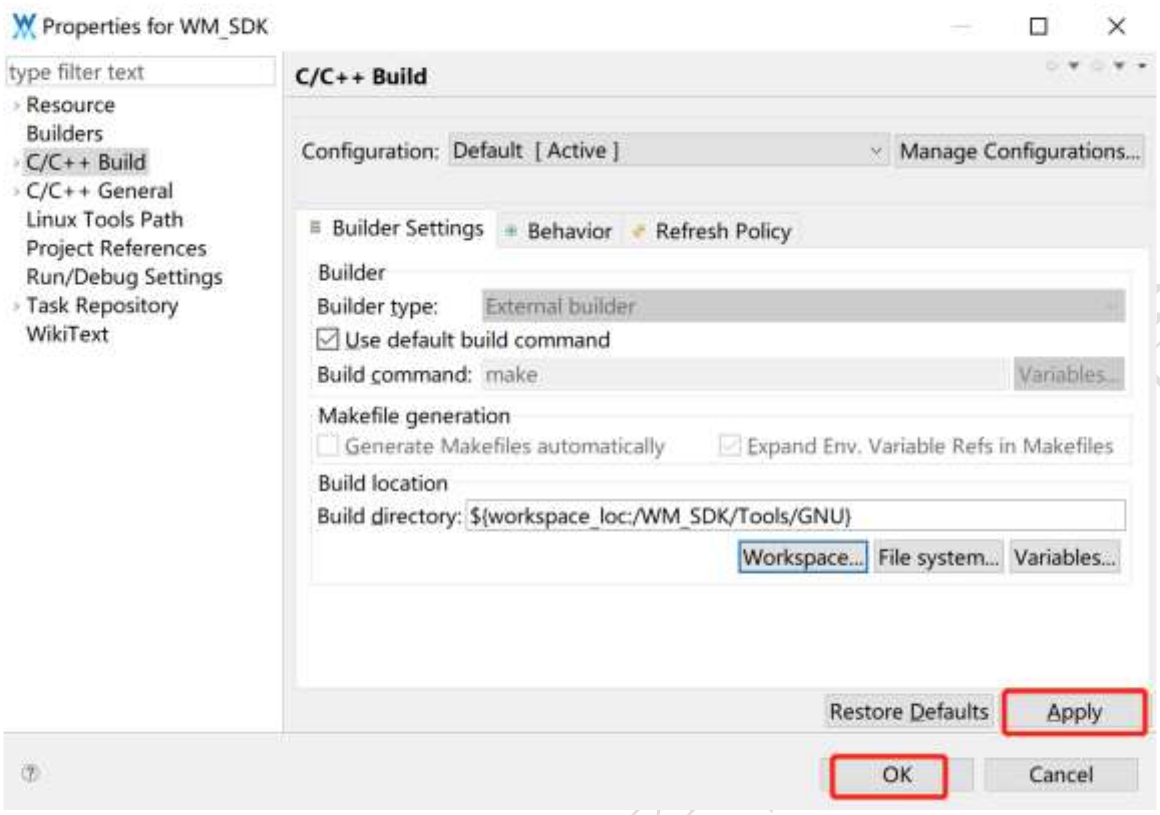
Build location

Build directory: ${workspace_loc:/WM_SDK}/


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


Properties for WM_SDK

type filter text
- Resource
- Builders
- C/C++ Build
- C/C++ General
- Linux Tools Path
- Project References
- Run/Debug Settings
- Task Repository
- WikiText

C/C++ Build

Configuration: Default [Active] Manage Configurations...

Builder Settings | Behavior | Refresh Policy

Builder
Builder type: External builder
☑ Use default build command
Build command: make Variables...

Makefile generation
☐ Generate Makefiles automatically ☑ Expand Env. Variable Refs in Makefiles

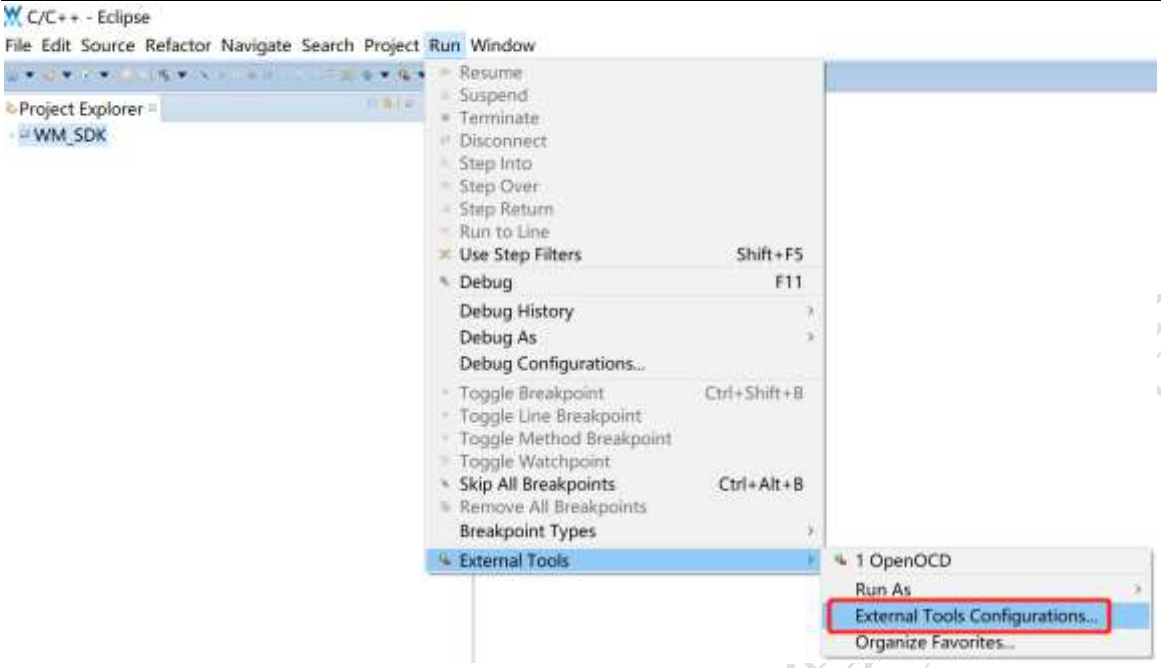
Build location
Build directory: ${workspace_loc:/WM_SDK/Tools/GNU}
Workspace... File system... Variables...

Restore Defaults Apply

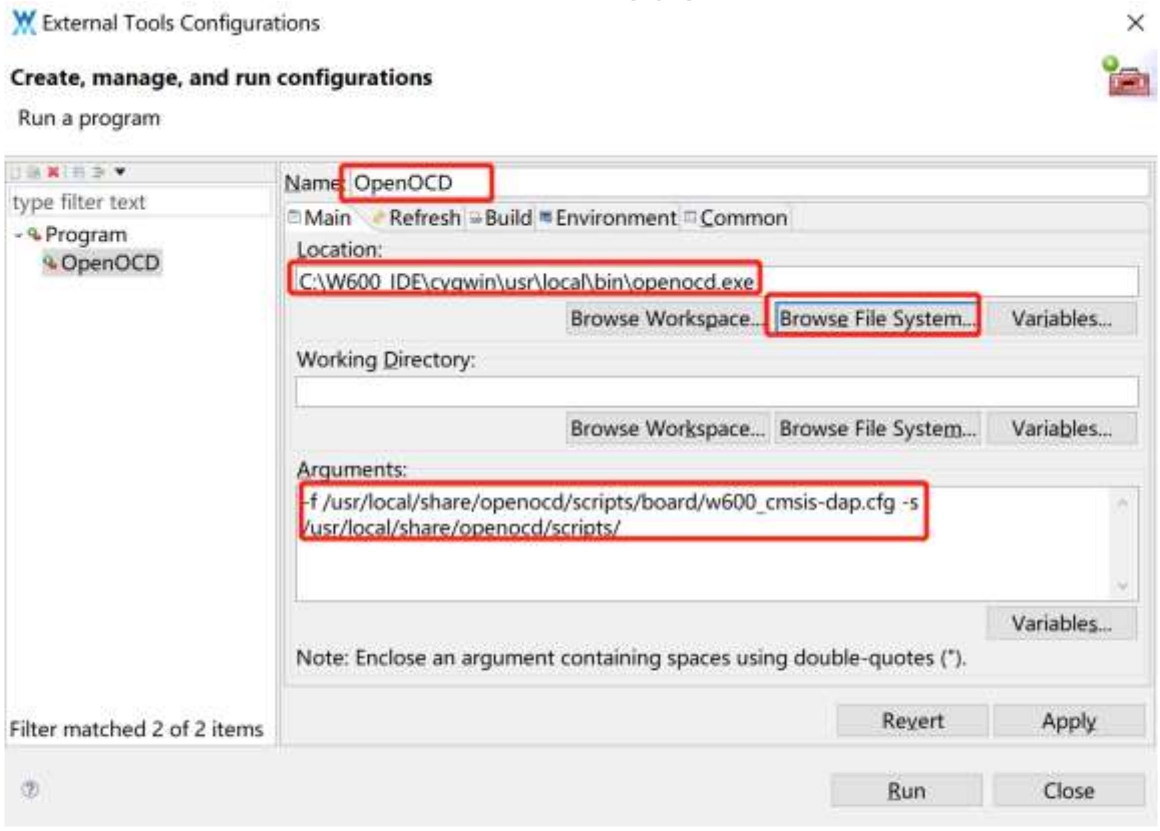
OK Cancel

2.5 配置 OpenOCD 启动

这里配置如何在 Eclipse 中如何启动 OpenOCD，这样操作的好处就是以后就不用再开一个 cygwin 窗口用来启动 OpenOCD。



在左侧双击新建，输入如下图：



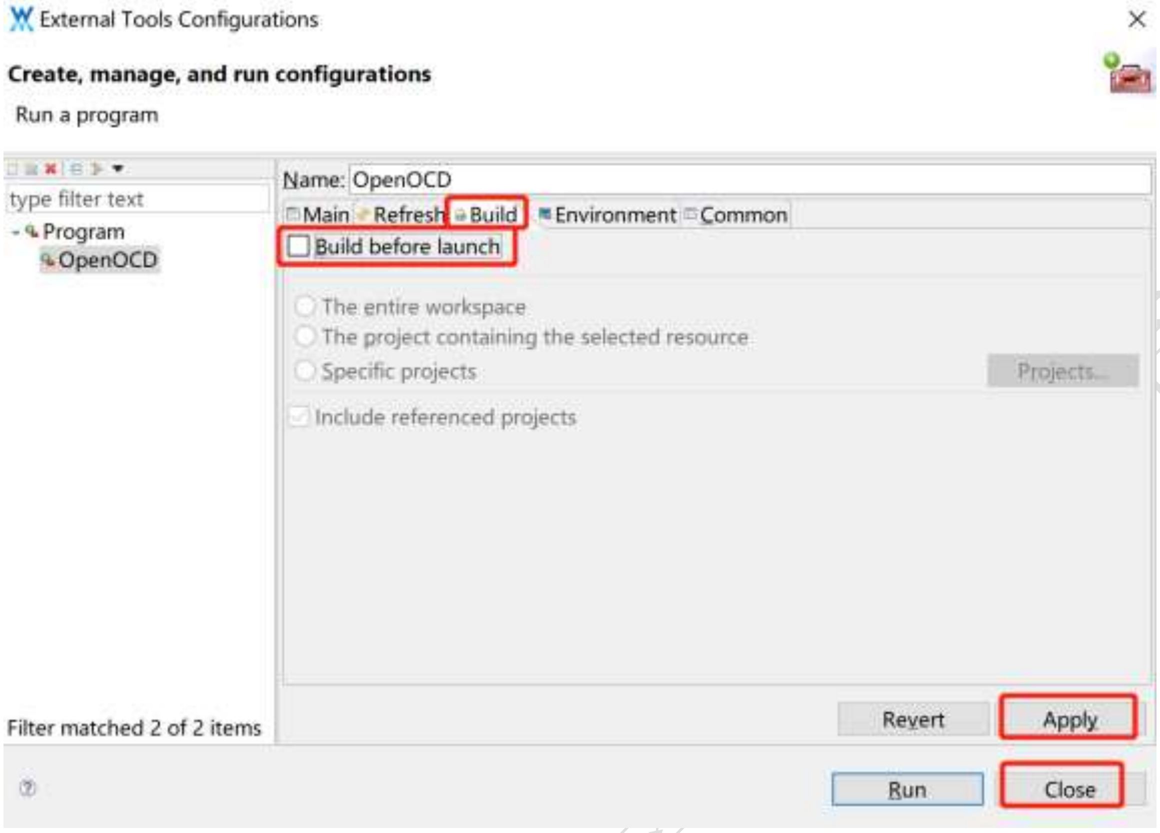
Name: OpenOCD

Location:
C:\W600_IDE\cygwin\usr\local\bin\openocd.exe

Working Directory:

Arguments:
-f /usr/local/share/openocd/scripts/board/w600_cmsis-dap.cfg -s /usr/local/share/openocd/scripts/

Note: Enclose an argument containing spaces using double-quotes (").


External Tools Configurations

Create, manage, and run configurations

Run a program

type filter text

Program

OpenOCD

Filter matched 2 of 2 items

Name: OpenOCD

Main | Refresh | Build | Environment | Common

☐ Build before launch

The entire workspace

The project containing the selected resource

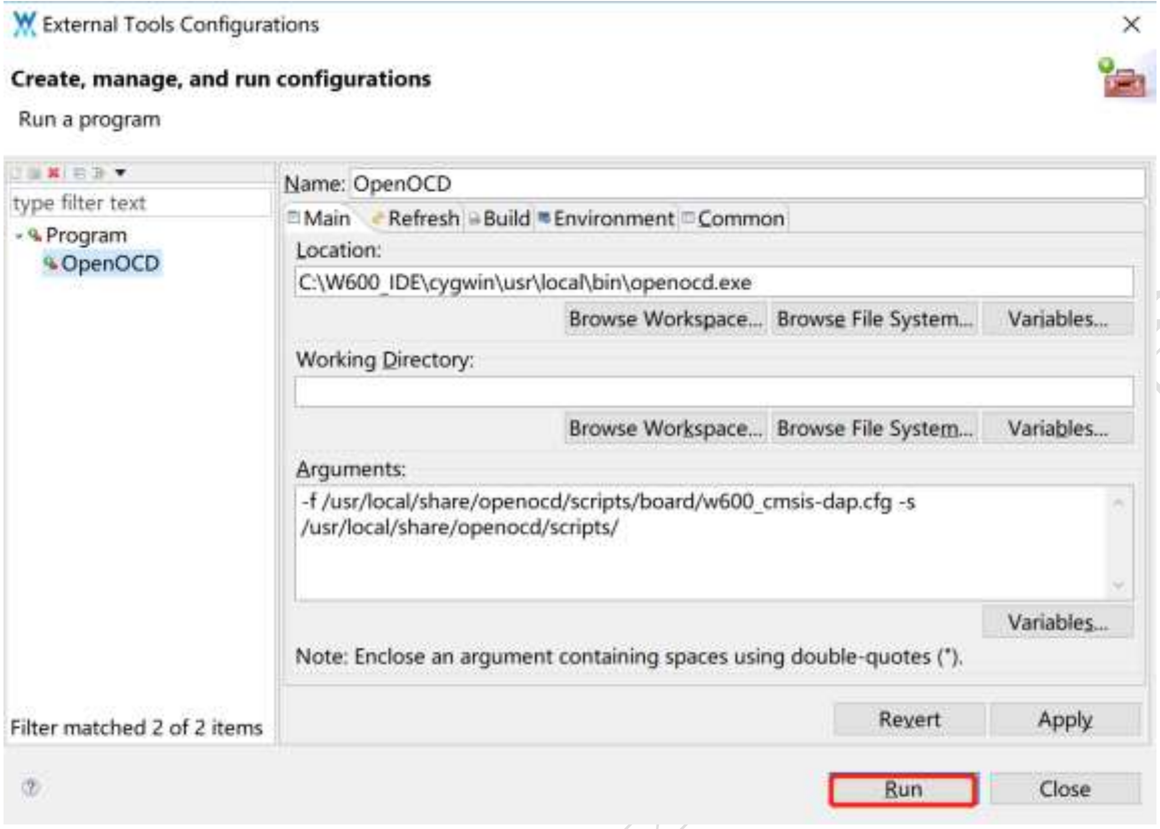
Specific projects

☑ Include referenced projects

OpenOCD 的配置文件要根据实际的 JTAG 仿真器来填写，不同的 JTAG 仿真器使用的配置文件是不同的，我们提供的集成压缩包开发环境中已经附带了 JLINK、CMSIS-DAP 的配置文件。

配置文件在 cygwin 的/usr/local/share/openocd/scripts/board/目录下，配置文件分别为 W60X_jlink.cfg、W60X_cmsis-dap.cfg。

设置时请根据实际使用的 JTAG 仿真器选择对应的配置文件，下图使用了 CMSIS-DAP 举例：


External Tools Configurations

Create, manage, and run configurations

Run a program

Name: OpenOCD

Location:
C:\W600_IDE\cygwin\usr\local\bin\openocd.exe

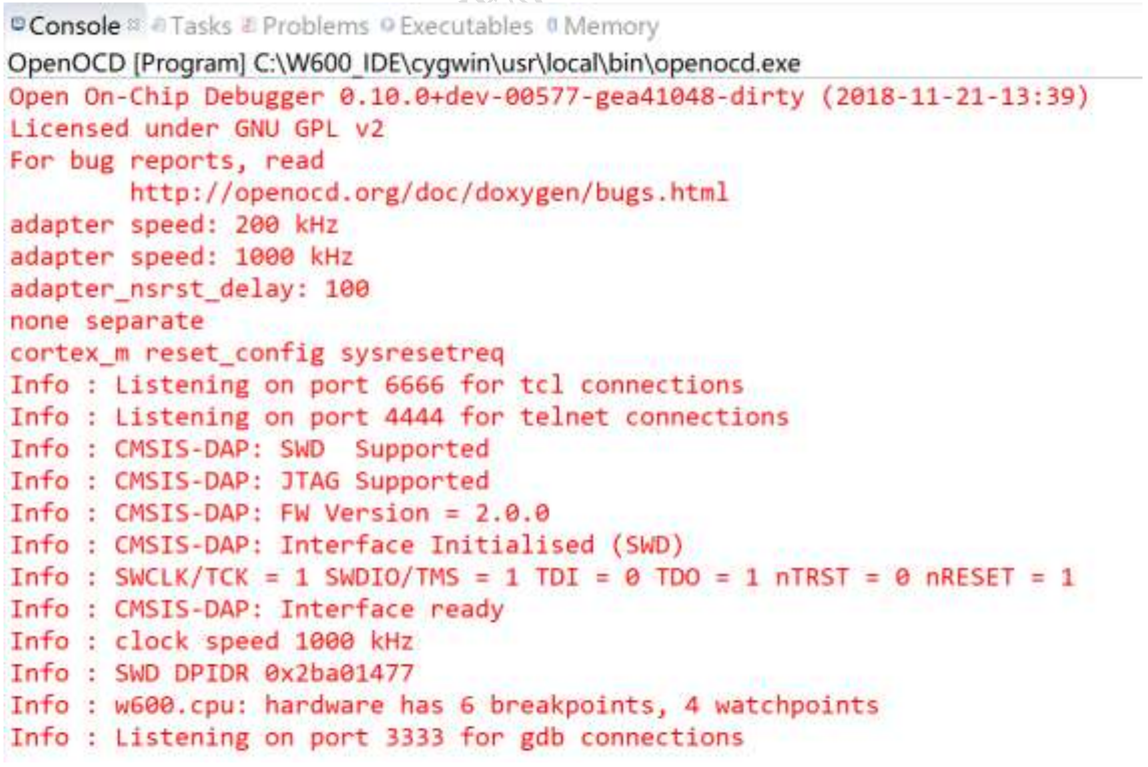
Working Directory:

Arguments:
-f /usr/local/share/openocd/scripts/board/w600_cmsis-dap.cfg -s /usr/local/share/openocd/scripts/

Note: Enclose an argument containing spaces using double-quotes (").

Filter matched 2 of 2 items

配置完成之后点击 Run 启动 OpenOCD，启动成功则会在 console 窗口输出信息：

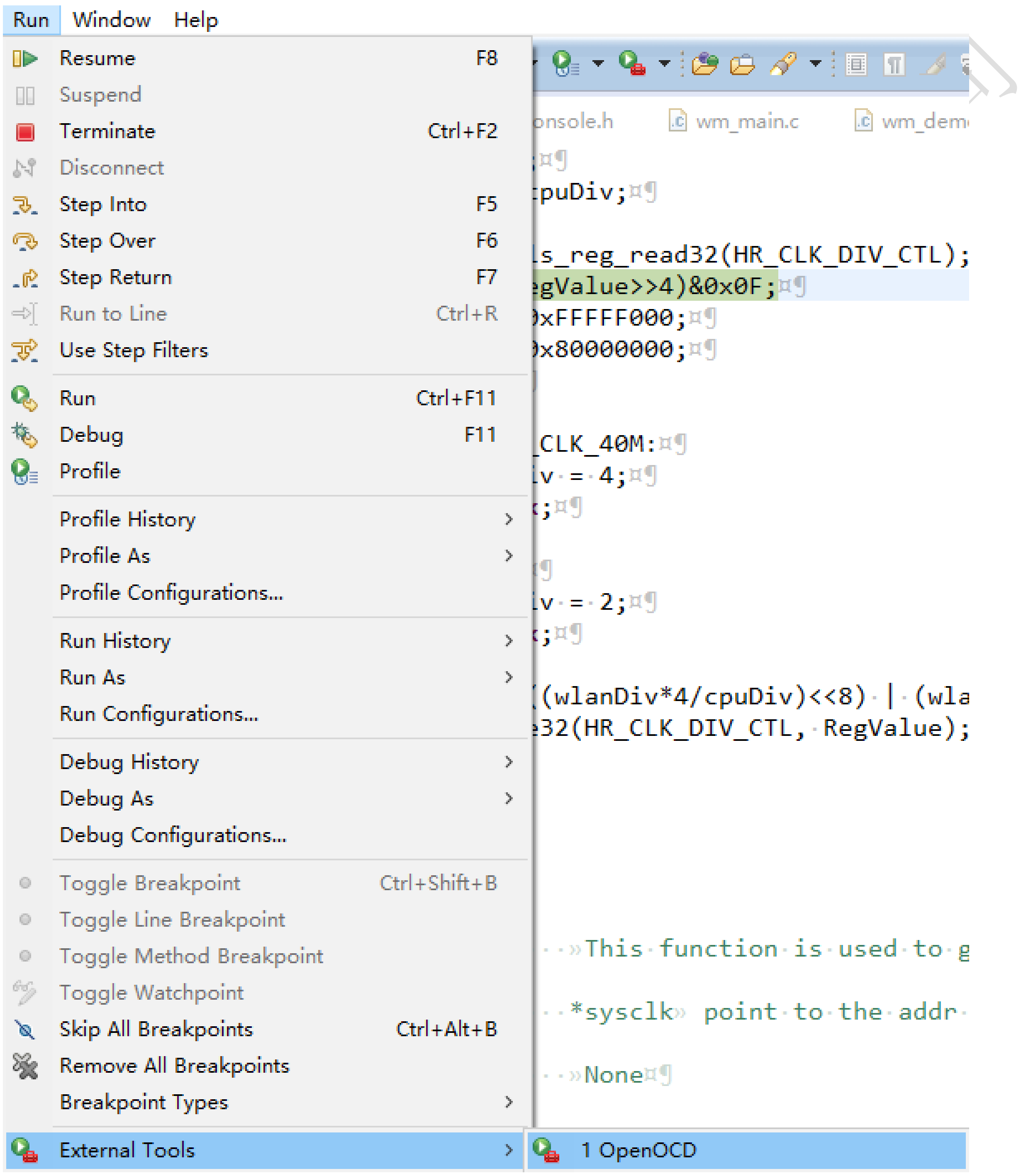


OpenOCD [Program] C:\W600_IDE\cygwin\usr\local\bin\openocd.exe

```
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
```

如果 OpenOCD 启动中发现无法识别 W60X 模块，可能是 W60X 模块当前所用的固件没有打开 SWD 引脚的调试功能，可参考下面的使用串口烧写固件章节烧写固件之后再尝试。

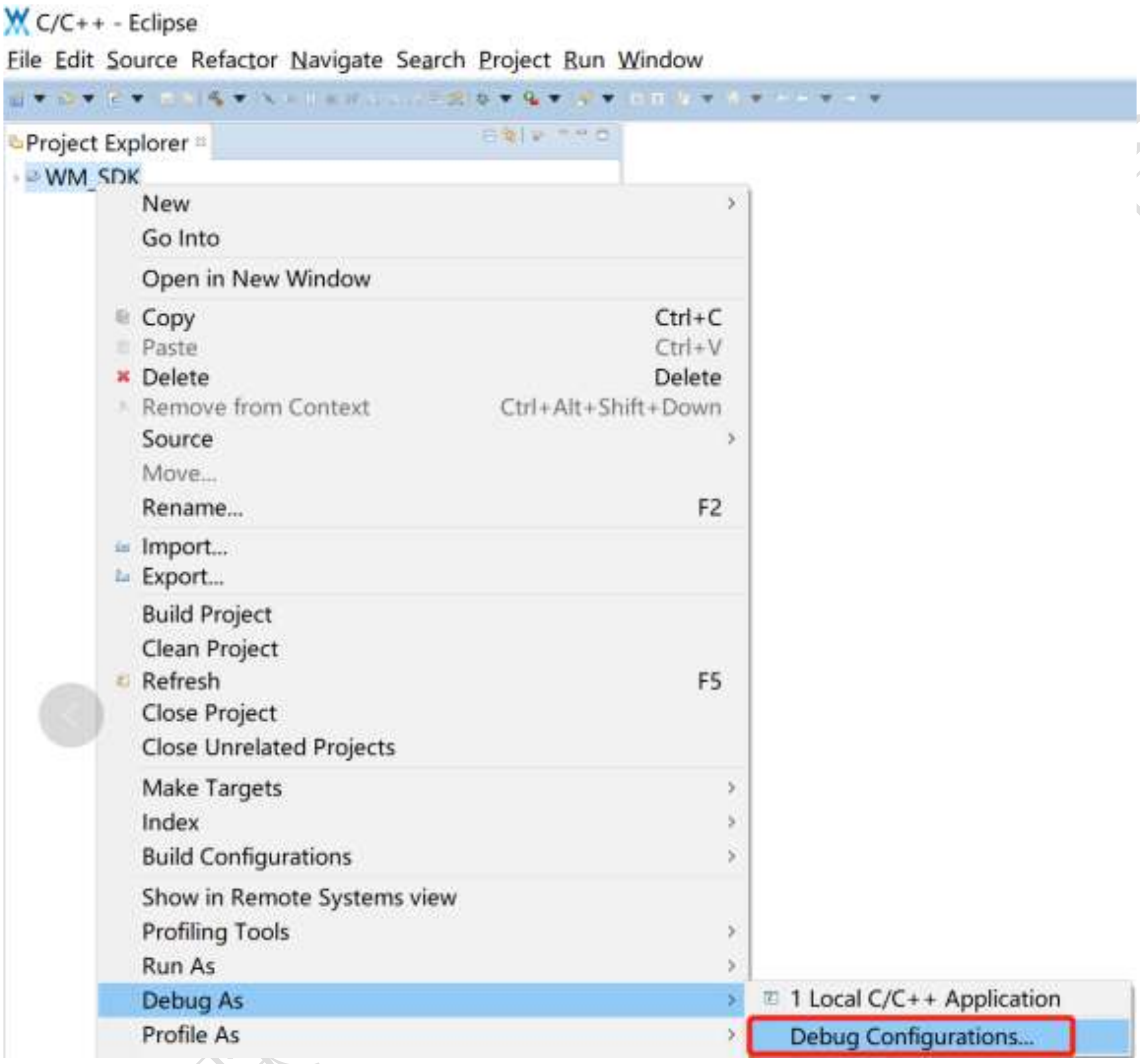
之后的启动也可以通过点击如下图所示来启动：



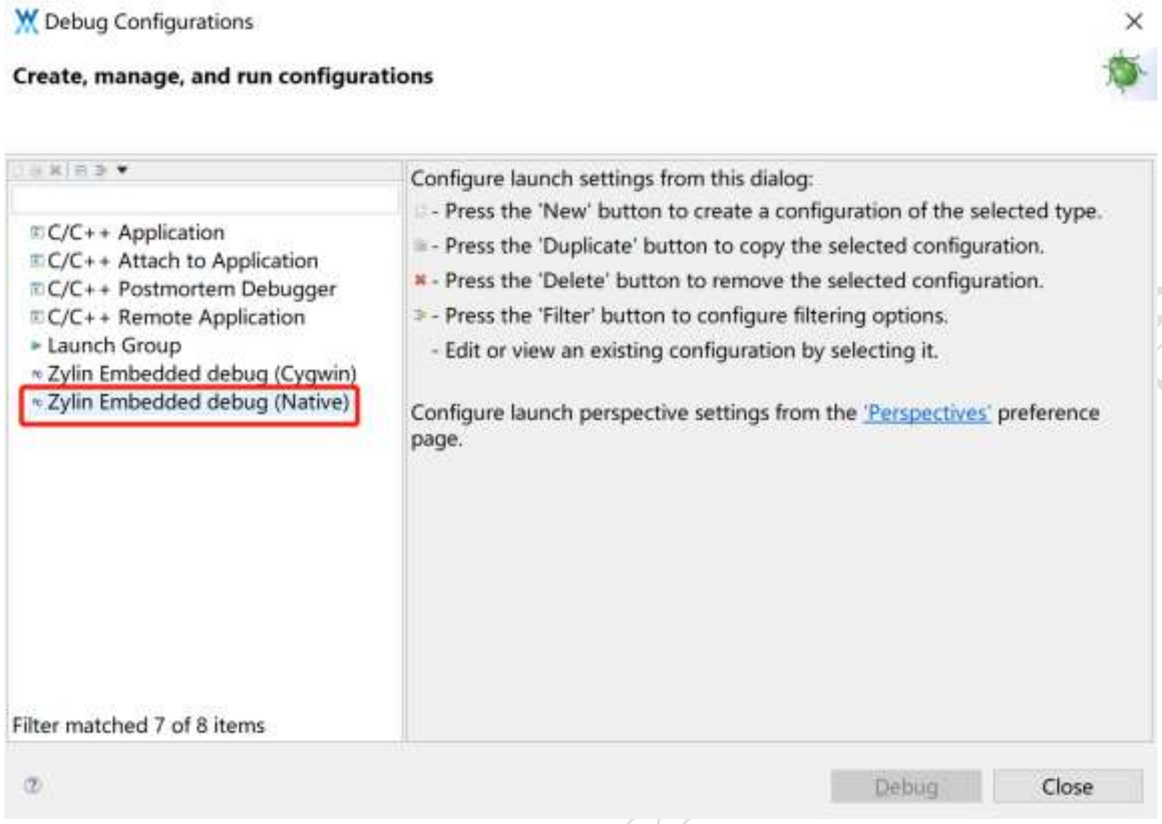
```
puDiv;
s_reg_read32(HR_CLK_DIV_CTL);
gValue>>4)&0x0F;
xFFFFF000;
x80000000;

CLK_40M:
v = 4;
;

v = 2;
;

(wlanDiv*4/cpuDiv)<<8) | (wla
32(HR_CLK_DIV_CTL, RegValue);

  This function is used to g
  *sysclk  point to the addr
  None
```

2.6 配置 Eclipse 调试功能

在工程管理窗口中右键 SDK 项目，选择 DEBUG 配置，如下图：



双击左侧的 Zylin Embedded debug (Native)，新建一个调试配置：


Debug Configurations

Create, manage, and run configurations

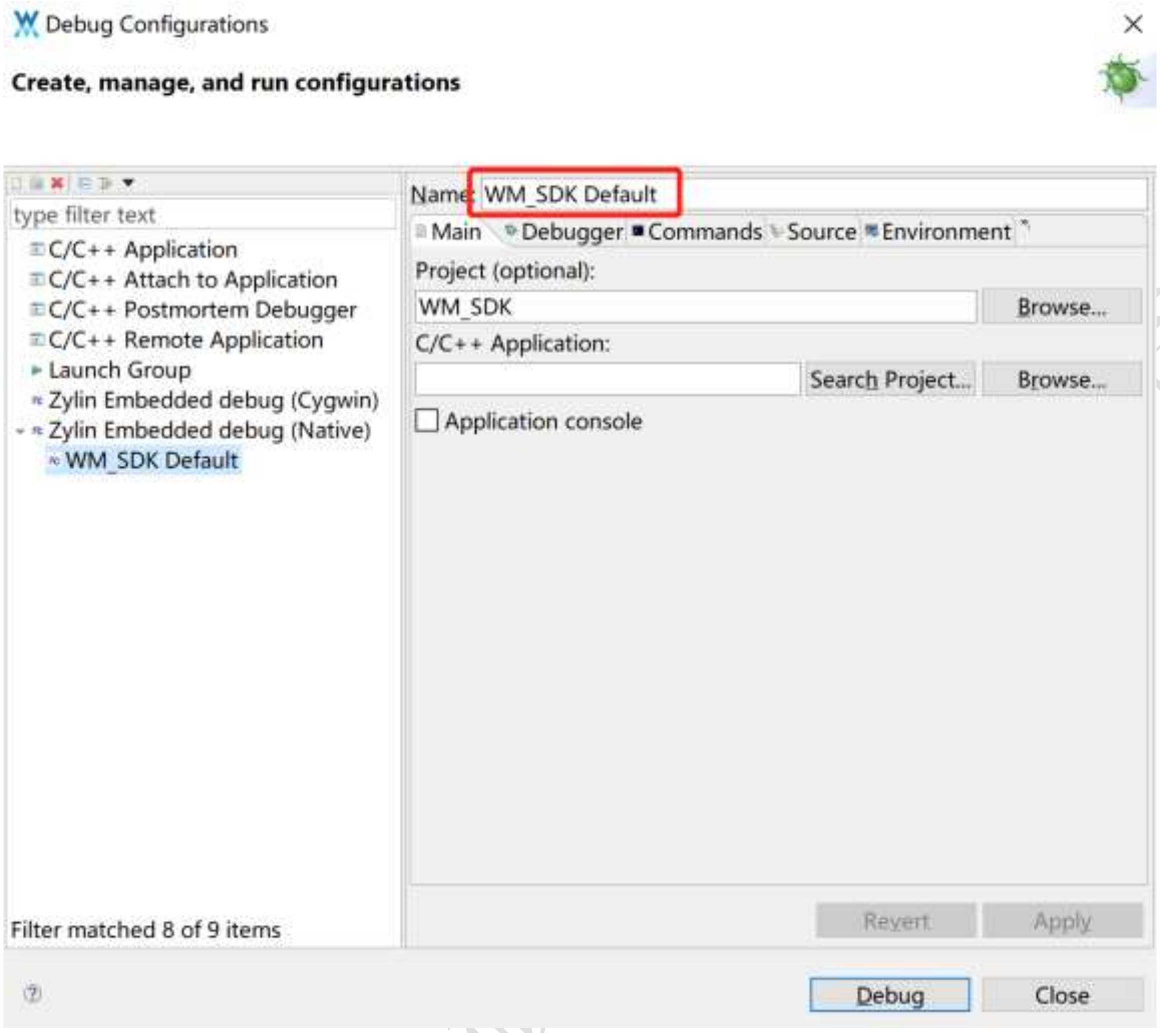
- C/C++ Application
- C/C++ Attach to Application
- C/C++ Postmortem Debugger
- C/C++ Remote Application
- Launch Group
- Zylin Embedded debug (Cygwin)
- Zylin Embedded debug (Native)

Filter matched 7 of 8 items

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.

名称最好保持和工程一致，便于区分其他工程：


Debug Configurations

Create, manage, and run configurations

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

Filter matched 8 of 9 items

Name: WM_SDK Default

Main | Debugger | Commands | Source | Environment

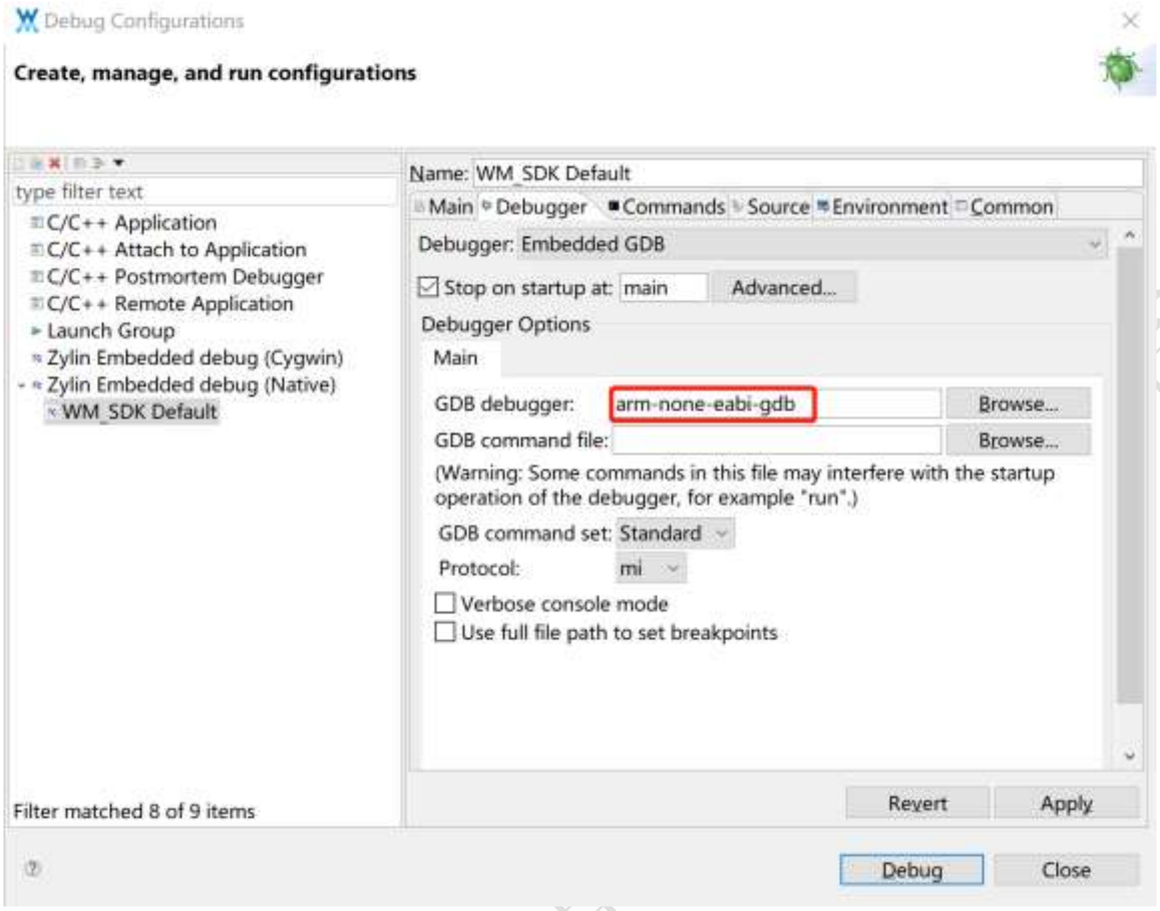
Project (optional):
WM_SDK

C/C++ Application:

☐ Application console

所用的调试器为 arm-none-eabi-gdb，我们提供的集成安装包里已经安装了交叉编译工具，路径在 cygwin 的/opt/arm-none-eabi-gcc/目录下。

我们提供的集成安装包已经配好了环境变量，所以只需要如下图设置即可：



Debug Configurations

Create, manage, and run configurations

Name: WM_SDK Default

Debugger: Embedded GDB

☑ Stop on startup at: main

Debugger Options

Main

GDB debugger: arm-none-eabi-gdb

GDB command file:

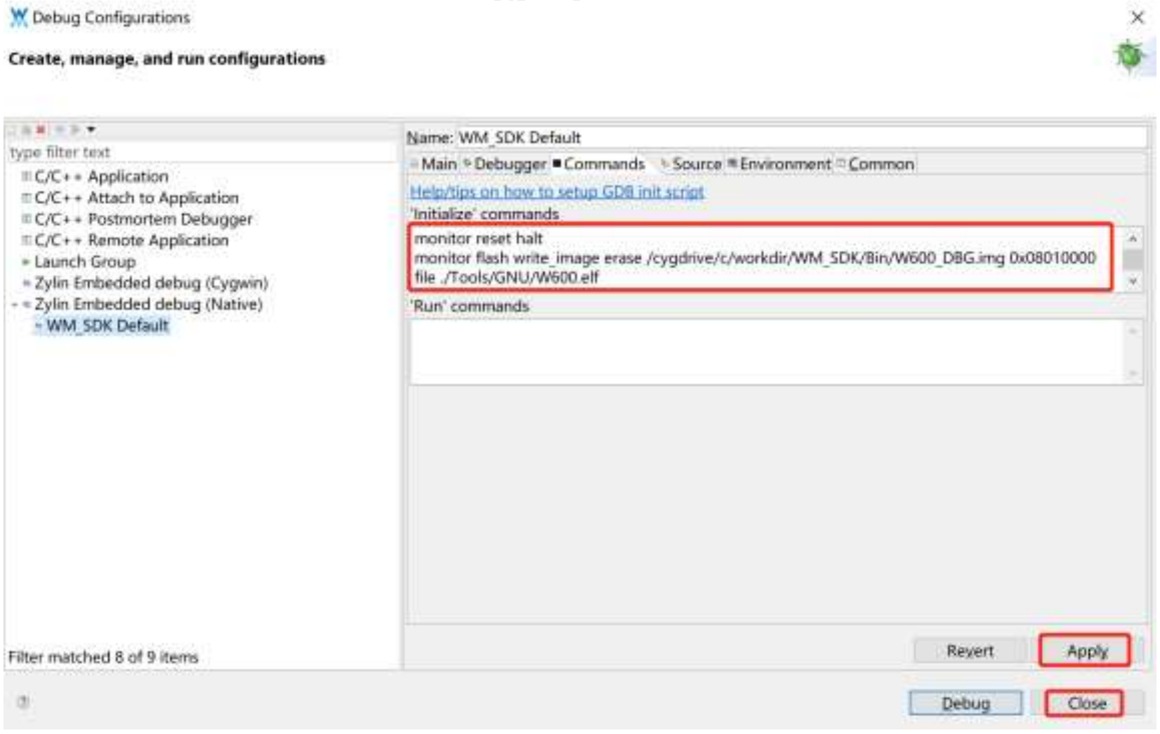
(Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".)

GDB command set: Standard

Protocol: mi

☐ Verbose console mode

☐ Use full file path to set breakpoints



Debug Configurations

Create, manage, and run configurations

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
Filter matched 8 of 9 items
Main
Debugger
Commands
Source
Environment
Common
Help/tips on how to setup GDB init script
Revert
Apply
Debug
Close

Name: WM_SDK Default

'Initialize' commands

```
monitor reset halt
monitor flash write_image erase /cygdrive/c/workdir/WM_SDK/Bin/W600_DBG.img 0x08010000
file ./Tools/GNU/W600.elf
```

'Run' commands

初始化填入的命令为：

```
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
```

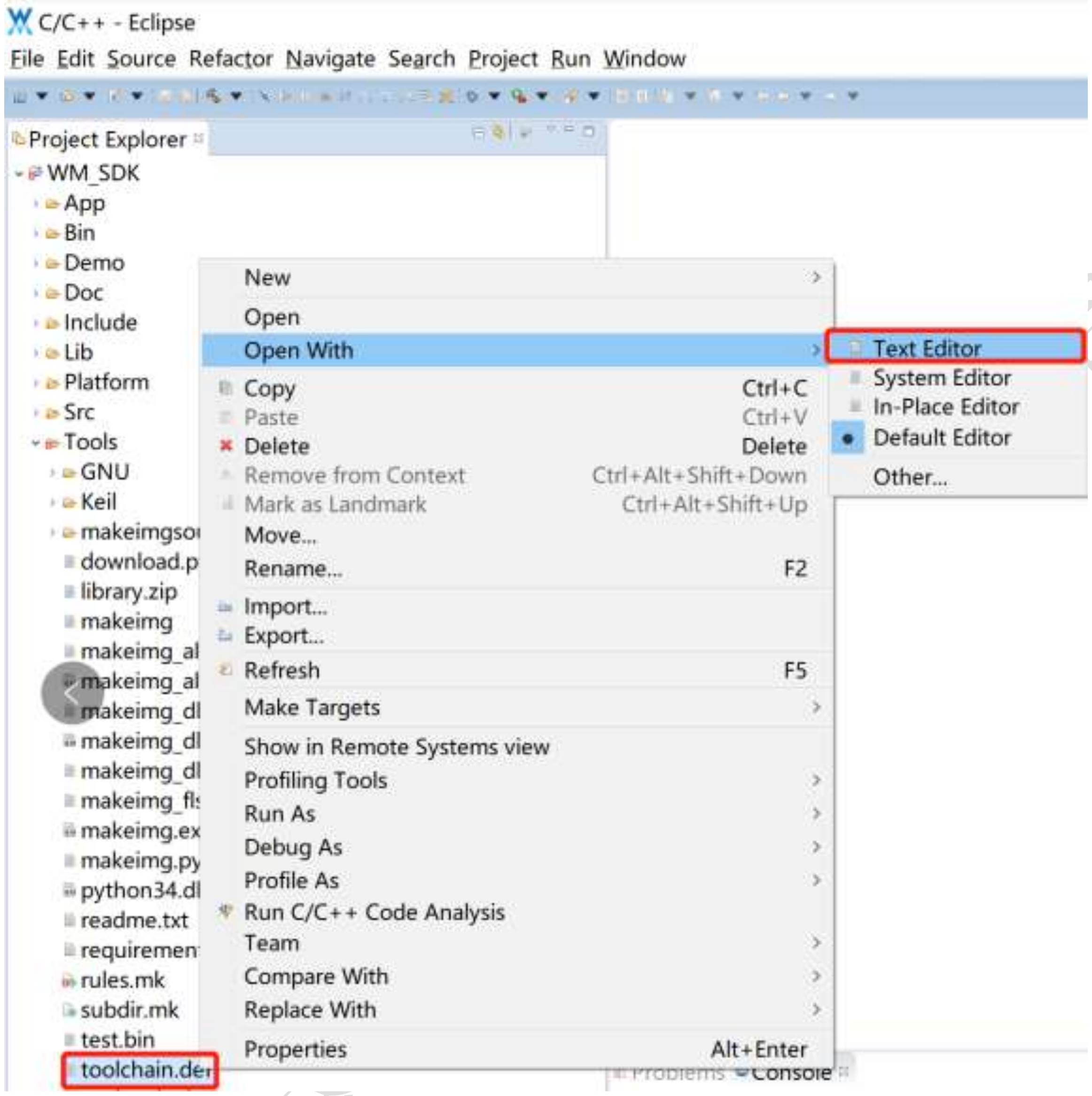
W60X 的调试固件烧写地址为 0x08010000，请勿修改避免造成 flash 损坏而导致模块无法正常工作。命令里的两个路径要根据实际的工程路径填写，其中固件文件 W60X_DBG.img 路径必须要使用绝对路径，符号表文件 W60X.elf 必须使用相对路径，否则会产生找不到文件的错误。

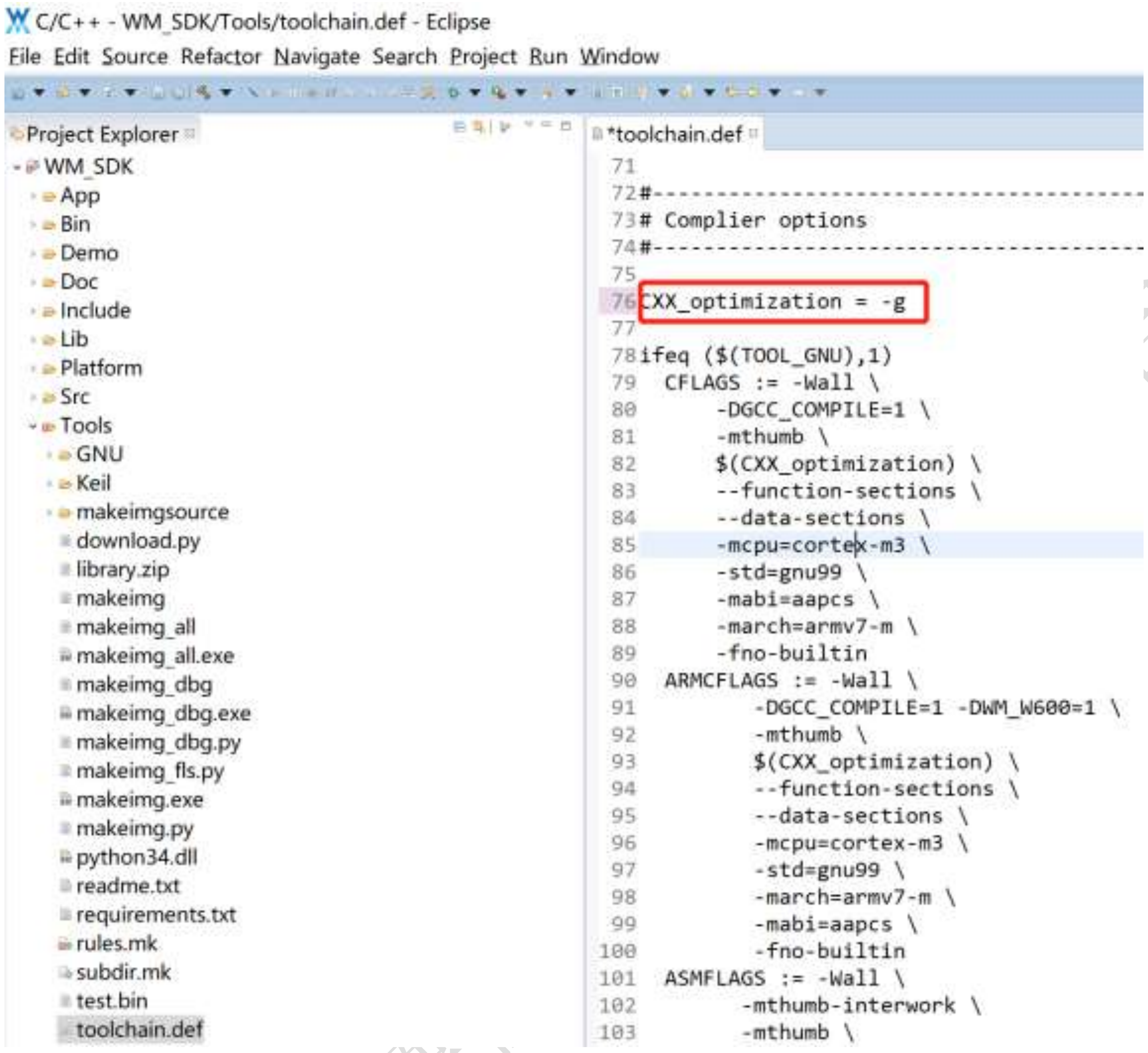
其他的保持默认即可，最后点击 Apply 保存配置。

如果调试中不是每次都需要更新固件，可以去掉 flash write_image 这条命令以减少固件烧写到 flash 的等待时间。

2.7 修改 SDK 源码优化级别

为了单步调试时使用符号表，将编译时的优化-Os 修改为-g 便于生成调试信息：

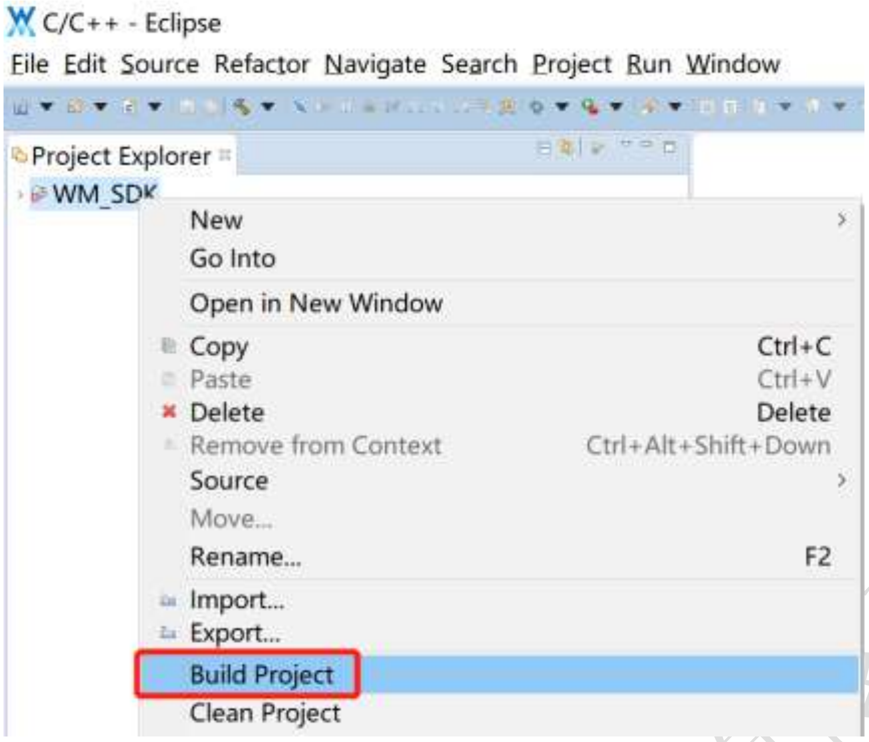

C/C++ - Eclipse
File Edit Source Refactor Navigate Search Project Run Window
Project Explorer
WM_SDK
App
Bin
Demo
Doc
Include
Lib
Platform
Src
Tools
GNU
Keil
makeimgsou
download.p
library.zip
makeimg
makeimg_al
makeimg_al
makeimg_dl
makeimg_dl
makeimg_dl
makeimg_fls
makeimg.ex
makeimg.py
python34.dl
readme.txt
requiremen
rules.mk
subdir.mk
test.bin
toolchain.def
New
Open
Open With
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Mark as Landmark Ctrl+Alt+Shift+Up
Move...
Rename... F2
Import...
Export...
Refresh F5
Make Targets
Show in Remote Systems view
Profiling Tools
Run As
Debug As
Profile As
Run C/C++ Code Analysis
Team
Compare With
Replace With
Properties Alt+Enter
Text Editor
System Editor
In-Place Editor
Default Editor
Other...
Problems Console

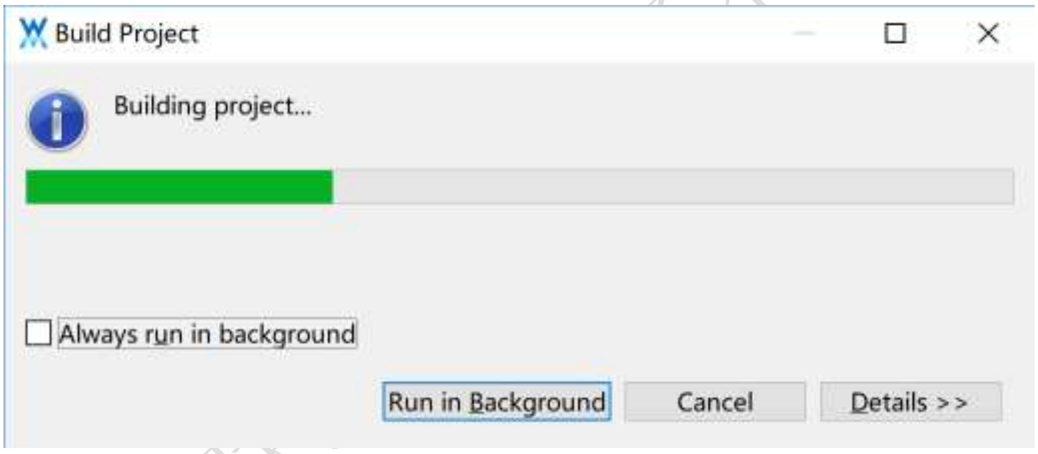


```

#--------------------------------------
# Complier options
#--------------------------------------

CXX_optimization = -g

ifeq ($(TOOL_GNU),1)
  CFLAGS := -Wall \
      -DGCC_COMPILE=1 \
      -mthumb \
      $(CXX_optimization) \
      --function-sections \
      --data-sections \
      -mcpu=cortex-m3 \
      -std=gnu99 \
      -mabi=aapcs \
      -march=armv7-m \
      -fno-builtin
  ARMCFLAGS := -Wall \
         -DGCC_COMPILE=1 -DWM_W600=1 \
         -mthumb \
         $(CXX_optimization) \
         --function-sections \
         --data-sections \
         -mcpu=cortex-m3 \
         -std=gnu99 \
         -march=armv7-m \
         -mabi=aapcs \
         -fno-builtin
  ASMFLAGS := -Wall \
        -mthumb-interwork \
        -mthumb \
```

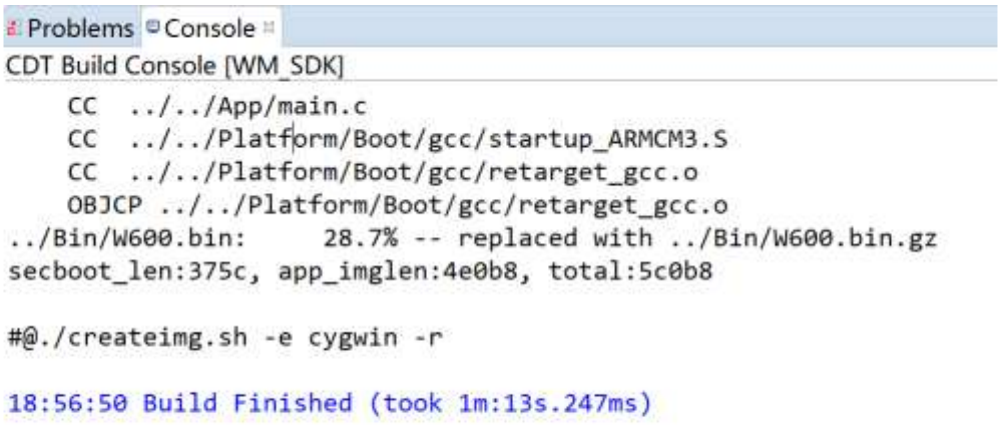
修改后保存文件即可完成修改。

2.8 编译 SDK





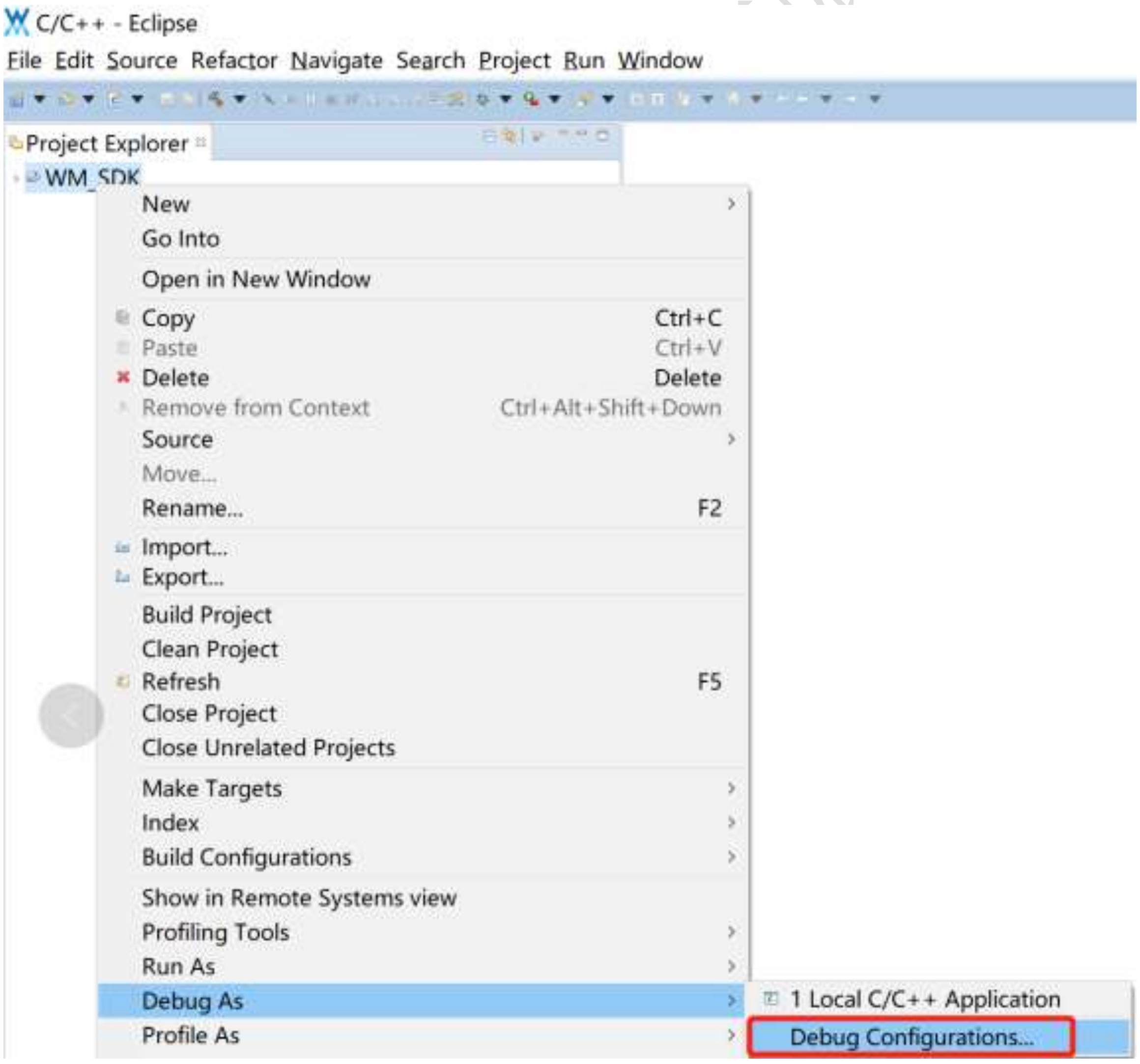
如果没有错误则有如下结果：

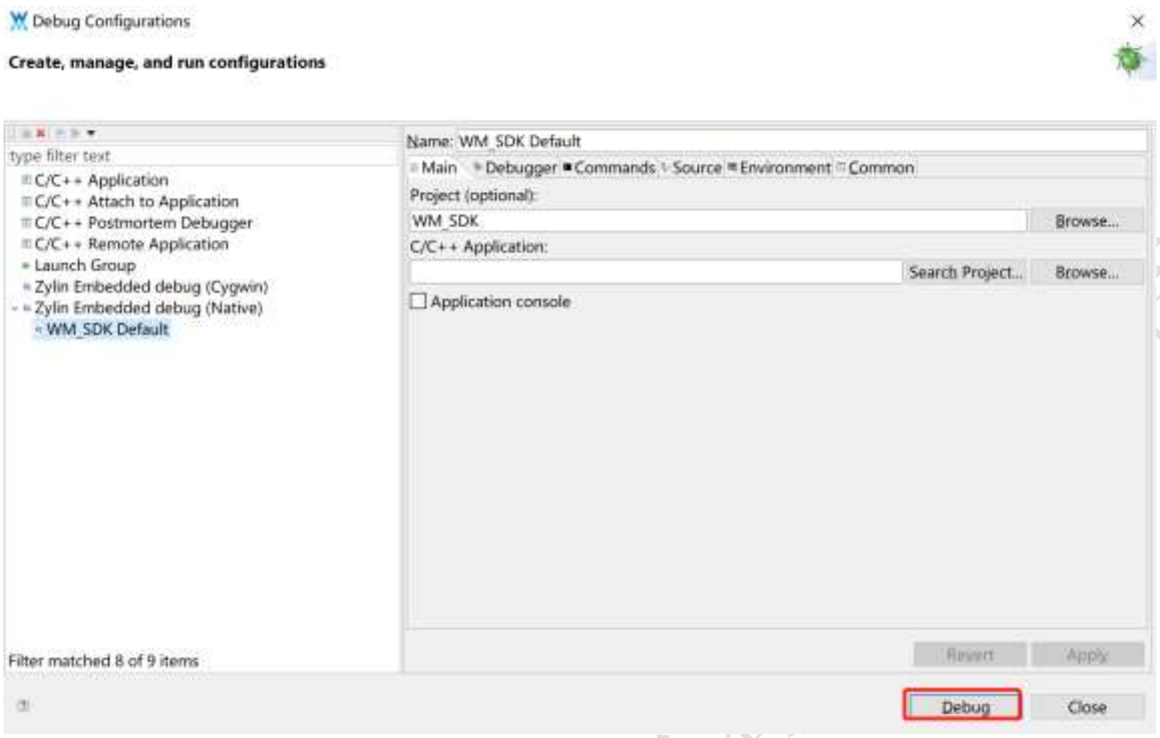


```
    CC  ../../App/main.c
    CC  ../../Platform/Boot/gcc/startup_ARMCM3.S
    CC  ../../Platform/Boot/gcc/retarget_gcc.o
    OBJCP ../../Platform/Boot/gcc/retarget_gcc.o
../Bin/W600.bin:     28.7% -- replaced with ../Bin/W600.bin.gz
secboot_len:375c, app_imglen:4e0b8, total:5c0b8

#@./createimg.sh -e cygwin -r

18:56:50 Build Finished (took 1m:13s.247ms)
```

2.9 启动调试

在启动调试之前需要先运行 OpenOCD（可以使用前面的配置启动 OpenOCD）再执行下面的操作。




Debug Configurations

Create, manage, and run configurations

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

Filter matched 8 of 9 items

Name: WM_SDK Default

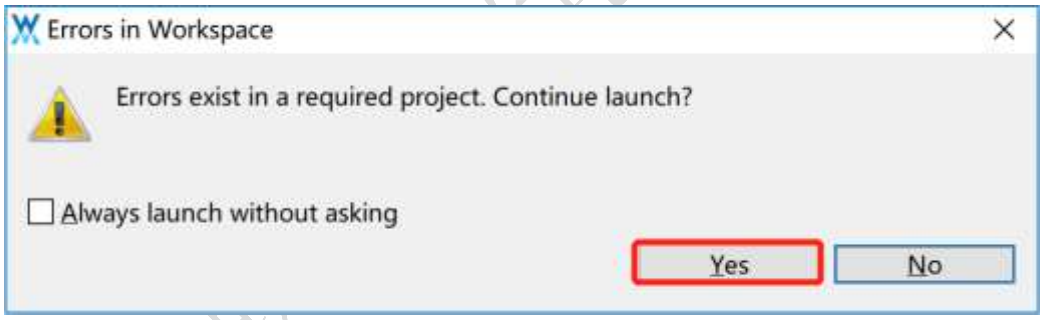
Main | Debugger | Commands | Source | Environment | Common

Project (optional): WM_SDK

C/C++ Application:

Application console

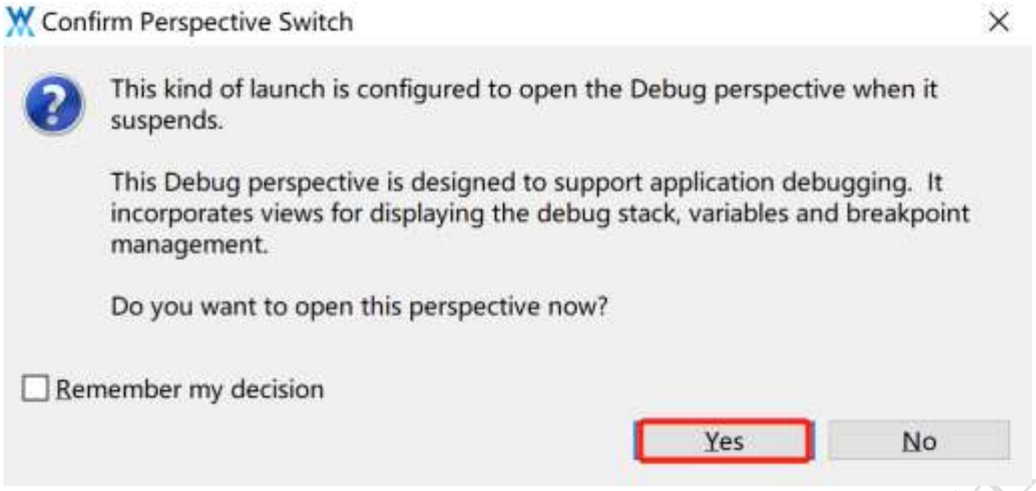
如果有如下提示则选择 Yes 忽略这个提示继续：


Errors in Workspace

Errors exist in a required project. Continue launch?

Always launch without asking

然后会提示是否 debug，确认一下即可：



Confirm Perspective Switch

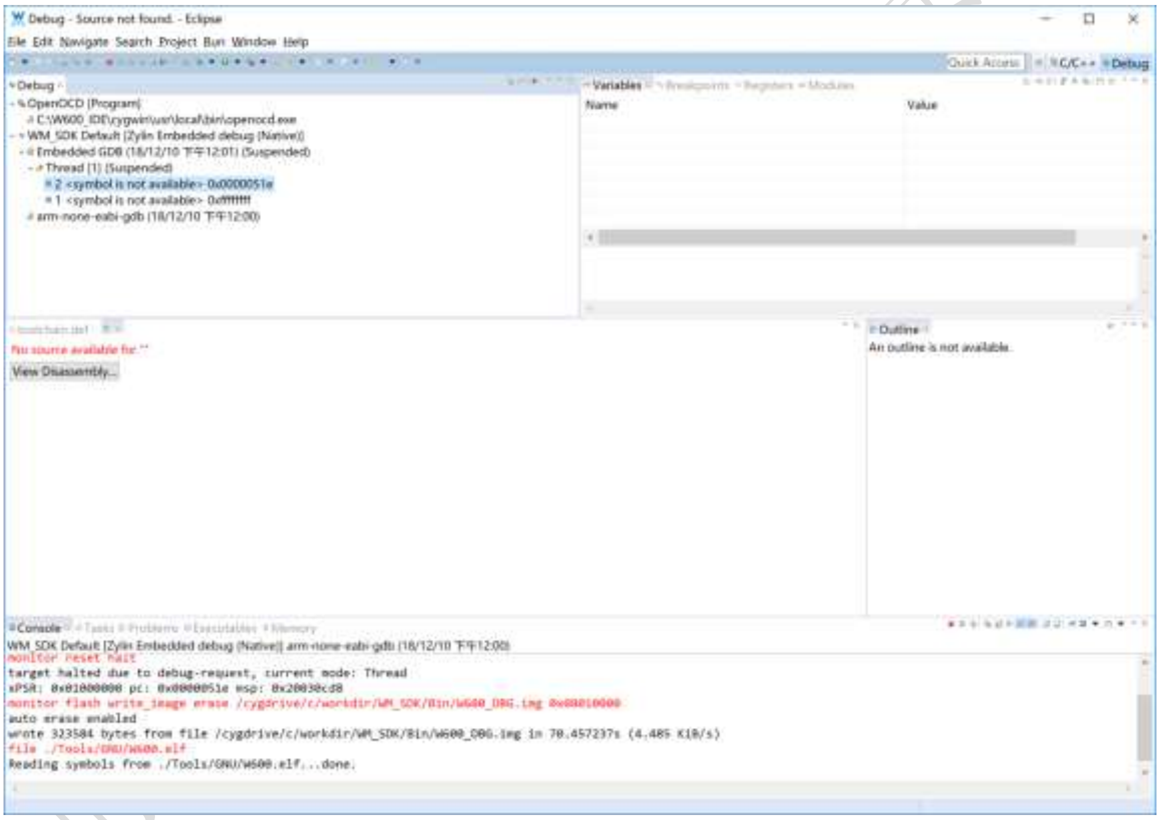
This kind of launch is configured to open the Debug perspective when it suspends.

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

烧写固件到 flash 比较耗时，请耐心等待至固件烧写完成之后，会出现下图提示：



Debug - Source not found. - Eclipse

```
OpenOCD [Program]
  C:\W600_IDE\cygwin\usr\local\bin\openocd.exe
WM_SDK Default [Zylin Embedded debug (Native)]
  Embedded GDB (18/12/10 下午12:01) (Suspended)
    Thread [1] (Suspended)
      2 <symbol is not available> 0x0000051e
      1 <symbol is not available> 0xffffffff
  arm-none-eabi-gdb (18/12/10 下午12:00)
```

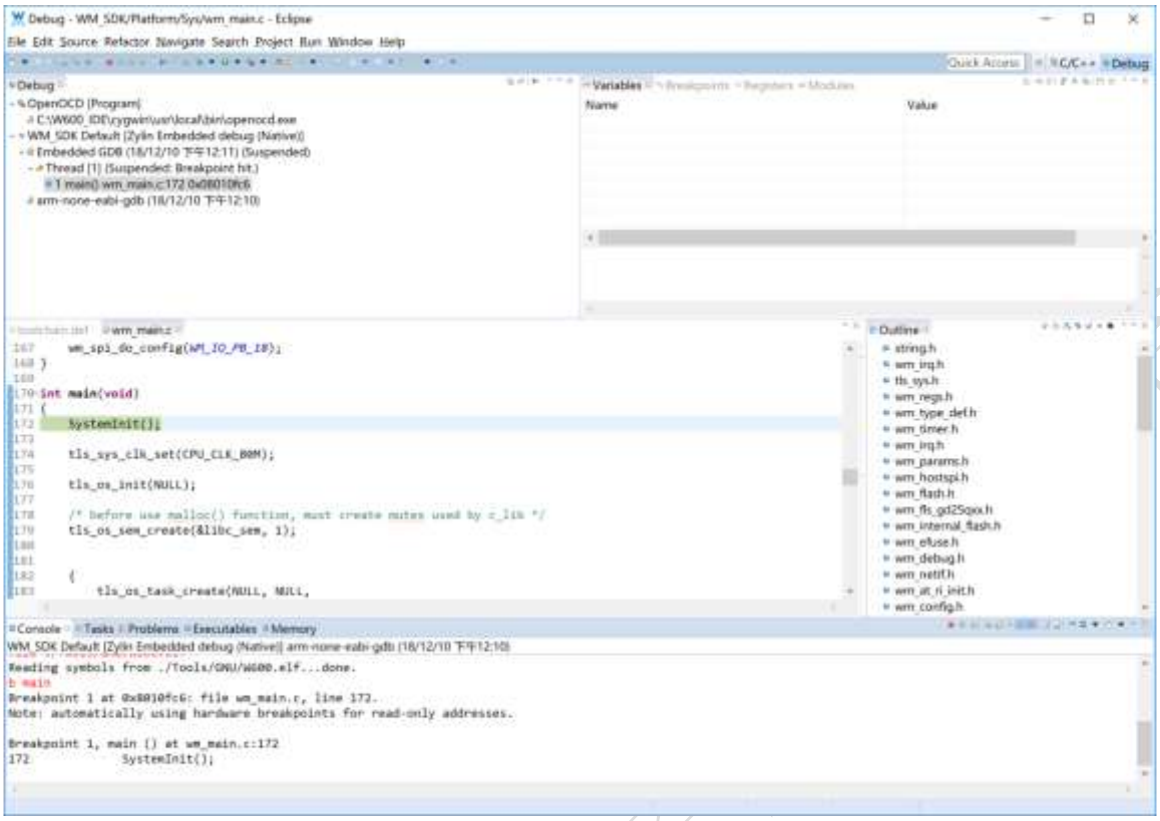
No source available for ""

View Disassembly...

An outline is not available.

```
WM_SDK Default [Zylin Embedded debug (Native)] arm-none-eabi-gdb (18/12/10 下午12:00)
monitor reset halt
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000051e msp: 0x20030cd8
monitor flash write_image erase /cygdrive/c/workdir/WM_SDK/Bin/W600_DBG.img 0x08010000
auto erase enabled
wrote 323584 bytes from file /cygdrive/c/workdir/WM_SDK/Bin/W600_DBG.img in 70.457237s (4.485 KiB/s)
file ./Tools/GNU/W600.elf
Reading symbols from ./Tools/GNU/W600.elf...done.
```

这时候点击 F8 键执行 Resume 命令，即可跑到 main 函数处停住：



Debug - WM_SDK/Platform/Sys/wm_main.c - Eclipse

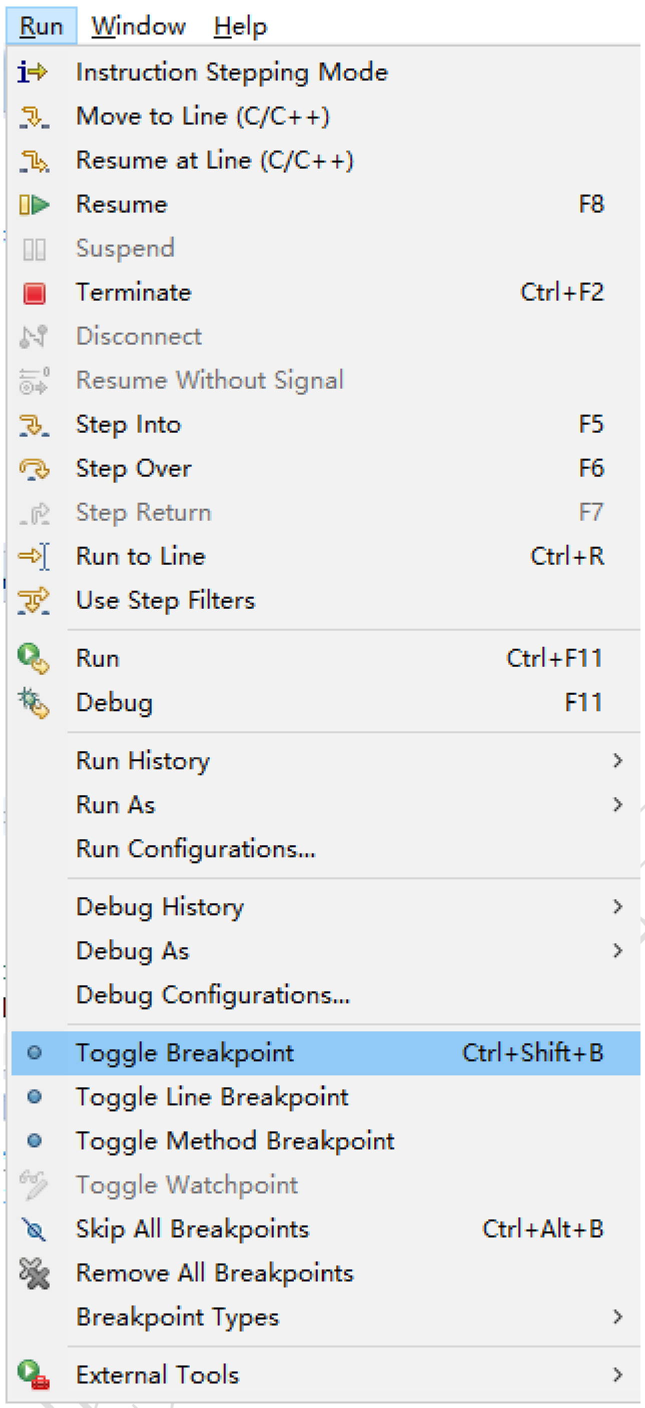
```
OpenOCD [Program]
  C:\W600_IDE\cygwin\usr\local\bin\openocd.exe
WM_SDK Default [Zylin Embedded debug (Native)]
  Embedded GDB (18/12/10 下午12:11) (Suspended)
    Thread [1] (Suspended: Breakpoint hit.)
      1 main() wm_main.c:172 0x08010fc6
  arm-none-eabi-gdb (18/12/10 下午12:10)
```

```
    wm_spi_do_config(WM_IO_PB_18);
}

int main(void)
{
    SystemInit();

    tls_sys_clk_set(CPU_CLK_80M);

    tls_os_init(NULL);

    /* before use malloc() function, must create mutex used by c_lib */
    tls_os_sem_create(&libc_sem, 1);


    {
        tls_os_task_create(NULL, NULL,
```

```
WM_SDK Default [Zylin Embedded debug (Native)] arm-none-eabi-gdb (18/12/10 下午12:10)
Reading symbols from ./Tools/GNU/W600.elf...done.
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();
```

此时就可以开始调试之旅了。

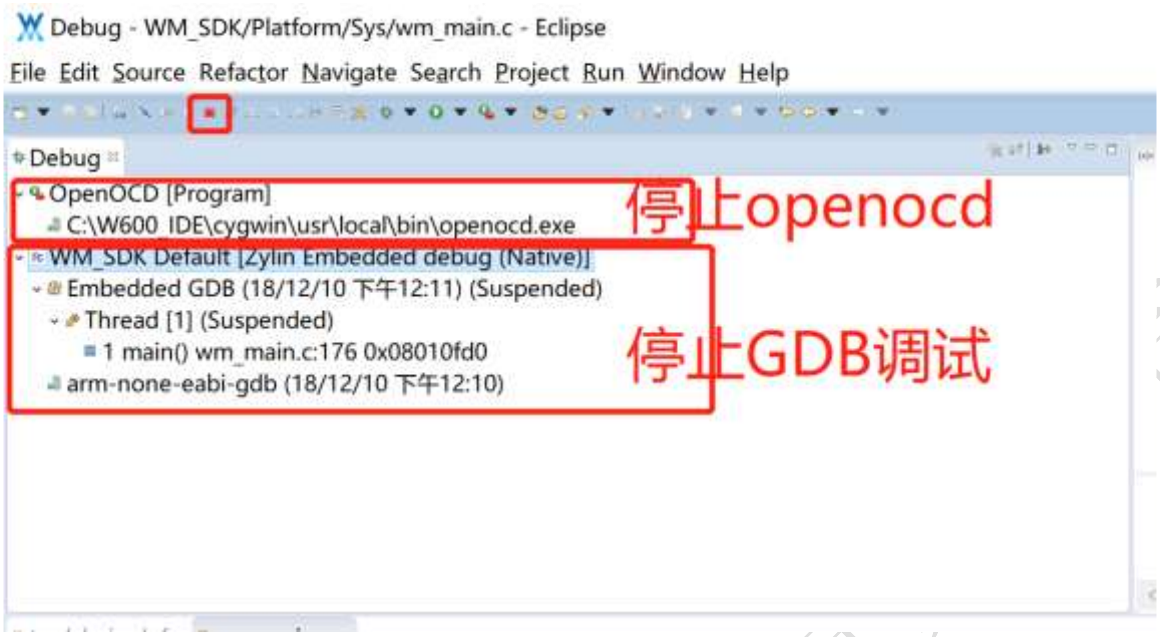
其他的调试命令可以自行在 Run 菜单中点击使用：



调试也可以直接点击工具栏图标进行快捷操作，其图示说明请参阅 Eclipse 安装章节。
如果遇到无法成功进入调试时，可以尝试手动重启 W60X 模块和重启 OpenOCD 之后，再进入调试即可。

2.10停止调试

停止调试时既可以停止 gdb，也选择可以停止 OpenOCD，这里根据情况自行选择。


Debug - WM_SDK/Platform/Sys/wm_main.c - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

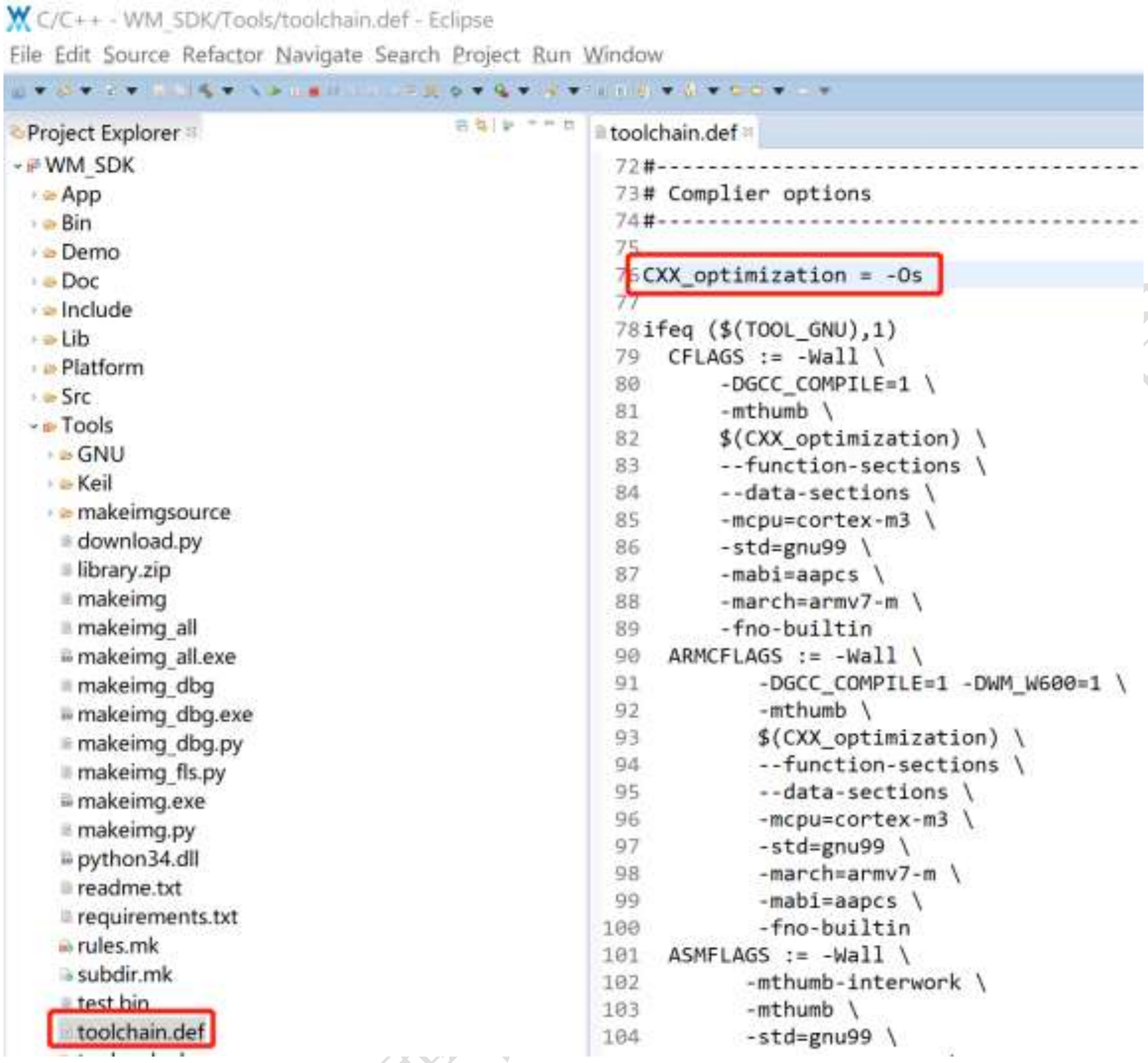
Debug

OpenOCD [Program]
C:\W600_IDE\cygwin\usr\local\bin\openocd.exe
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)

停止调试必须是在 gdb 处于断住的情况下，否则点击停止 debug 是不会产生效果的。如果碰到这种情况请先点击停止 OpenOCD，只要 OpenOCD 停止运行 gdb 也会停止。

2.11编译 Release 固件

编译正式版本的固件时，需要将源码的优化级别改为-Os，如下图所示；



```
#----------------------------------------
# Complier options
#----------------------------------------

CXX_optimization = -Os

ifeq ($(TOOL_GNU),1)
  CFLAGS := -Wall \
        -DGCC_COMPILE=1 \
        -mthumb \
        $(CXX_optimization) \
        --function-sections \
        --data-sections \
        -mcpu=cortex-m3 \
        -std=gnu99 \
        -mabi=aapcs \
        -march=armv7-m \
        -fno-builtin
  ARMCFLAGS := -Wall \
           -DGCC_COMPILE=1 -DWM_W600=1 \
           -mthumb \
           $(CXX_optimization) \
           --function-sections \
           --data-sections \
           -mcpu=cortex-m3 \
           -std=gnu99 \
           -march=armv7-m \
           -mabi=aapcs \
           -fno-builtin
  ASMFLAGS := -Wall \
          -mthumb-interwork \
          -mthumb \
          -std=gnu99 \
```

SWD 复用引脚默认是否关闭由用户自行决定（调用接口函数 wm_swd_config(0)设置即可）。

2.12使用串口下载固件

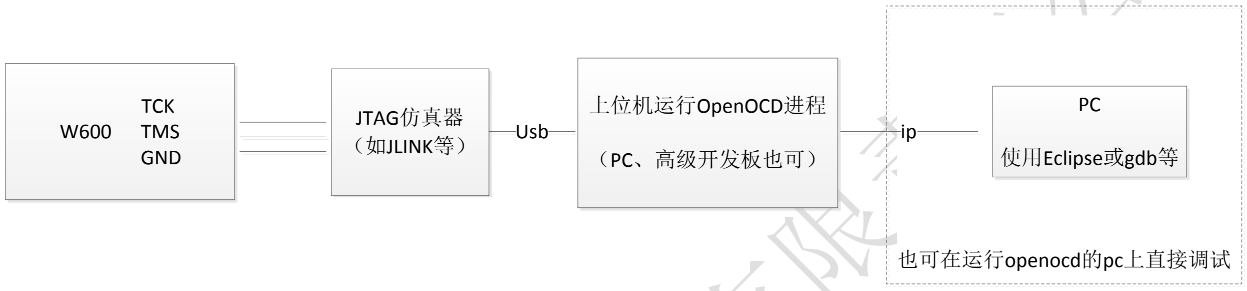
W60X 支持串口烧写固件，具体操作请参阅 http://www.winnermicro.com/html/1/156/158/497.html 提供的文档“WM_W60X_固件升级指导”。

3　OpenOCD 使用进阶

3.1 OpenOCD 简介

在嵌入式开发中，有很多优秀的调试、仿真工具，比如Keil、IAR、Rowley Associates 等，它们的安装、使用都很便利，功能强大，但是价格昂贵(几百美元甚至更多)；还要购买相应的硬件，比如U-Link等USB 到JTAG 的转换盒，这也是一笔不小的开支。对于开发预算有限的工程师来说，完全可以使用免费的开发工具Eclipse、OpenOCD，然后通过一些便宜的JTAG 转接器(比如CMSIS-DAP)就可以达到接近、甚至超越上述商业软件的效果。

OpenOCD是一个开源的JTAG上位机程序，在嵌入式设备调试中起着承上启下的作用：对下使用JTAG仿真器连接嵌入式设备，对上则是为上位机提供通用的调试命令，整个使用的关系如下图所示：



目前 OpenOCD 已经支持的 JTAG 仿真器有如下：

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

目前OpenOCD已经支持调试的架构如下：

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

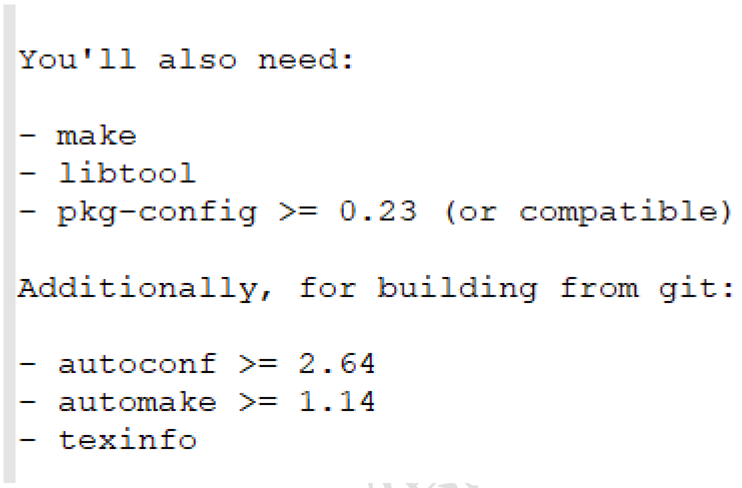
而且OpenOCD还对很多的Flash都提供了支持：

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 编译安装 OpenOCD

本节的操作都是基于 Windows 系统下的 Cygwin 环境，所以首先需要安装 Cygwin，在前面快速上手章节中我们已经提供了一个集成 Cygwin 的压缩包开发环境，建议普通用户直接下载使用打包好的环境以减轻工作量，安装好之后双击 cygwin\Cygwin.bat 则可打开一个 cygwin 的 shell 窗口，之后下面的操作都是在这个 shell 窗口中进行。

按照 OpenOCD 官方文档要求，编译必须依赖项有：



```
You'll also need:

- make
- libtool
- pkg-config >= 0.23 (or compatible)

Additionally, for building from git:

- autoconf >= 2.64
- automake >= 1.14
- texinfo
```

所以接下需要检查 Cygwin 环境是否支持，不支持的依赖项则需要用户自行安装，如果使用的是我们提供的环境则可以跳过这一步。

3.2.1 编译安装 libusb

OpenOCD 部分仿真器需要 libusb 的支持，所以在编译 OpenOCD 之前需要先安装好 libusb，如果已经安装过 libusb 则可跳过这一步（我们提供的环境已经安装好了该库）。

libusb 的编译安装步骤为：

1. 在 https://libusb.info 下载源码包并解压
2. 进入源码目录
3. 执行 *./configure --prefix=/usr/* 生成 Makefile
4. 执行 *make* 编译

5. 执行 *make install* 安装

3.2.2 编译安装 HIDAPI library

考虑到使用 CMSIS-DAP 仿真器的用户比较多，所以这里我们增加对了对其的支持，如果不需要使用 CMSIS-DAP 仿真器则可以跳过这一步。OpenOCD 需要 HIDAPI library 的支持，所以在编译 OpenOCD 之前需要先安装好 HIDAPI library，如果已经安装过 HIDAPI library 则可跳过这一步（我们提供的环境已经安装好了该库）。

HIDAPI library 的编译安装步骤为：

1. 执行 git clone https://github.com/signal11/hidapi.git 下载源码
2. 进入源码目录
3. 执行 *./bootstrap* 生成 configure 文件
4. 执行 *./configure --prefix=/usr* 生成 Makefile
5. 执行 *make* 编译
6. 执行 *make install* 安装

3.2.3 编译安装 OpenOCD

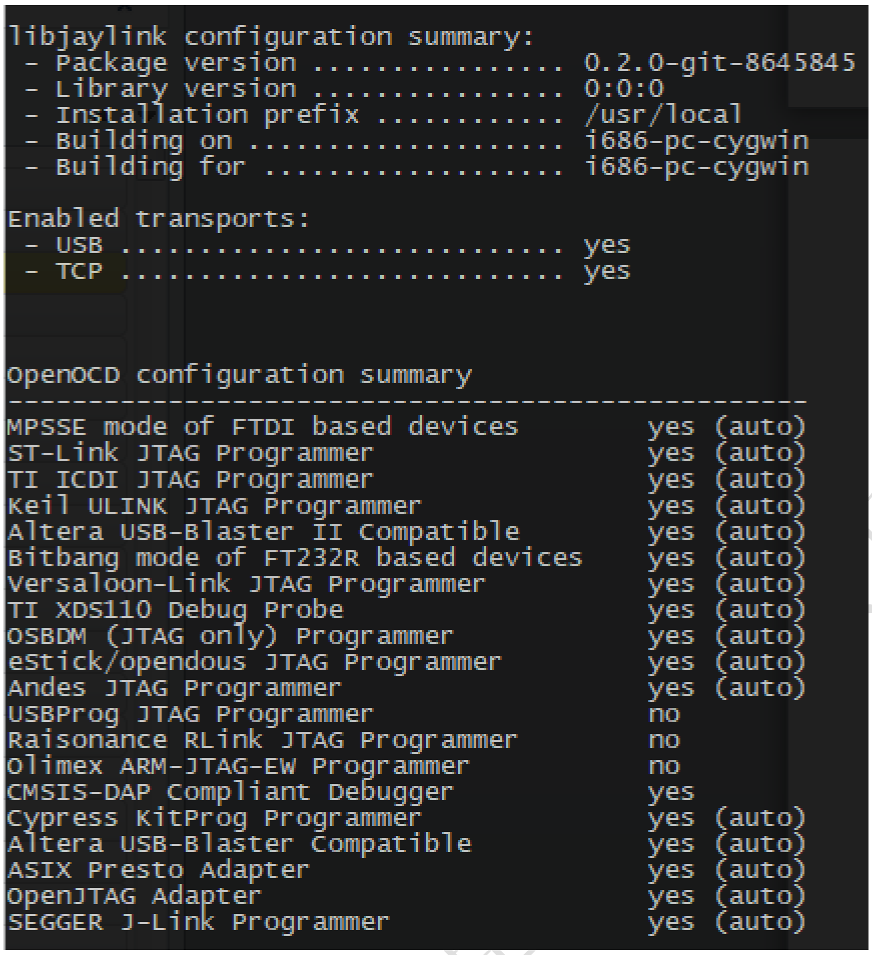
我们提供的环境已经安装好了 OpenOCD，

1. 下载 OpenOCD 源码：
   请从网站 http://www.winnermicro.com/下载源码包，目前发布的版本为 W60X_openocd_0.10.0_r1。

2. 执行命令检查环境依赖项并生成 Makefile
   ./configure --enable-cmsis-dap --disable-werror

   执行之后如果产生错误，那么说明缺少一些依赖，需要使用者自行修复，具体以实际环境为准。如果不使用 CMSIS-DAP 仿真器，则可以不加 *-enable-cmsis-dap* 选项。

   检查完成时的结果如下图所示：



```
libjaylink configuration summary:
 - Package version ................ 0.2.0-git-8645845
 - Library version ................ 0:0:0
 - Installation prefix ............ /usr/local
 - Building on .................... i686-pc-cygwin
 - Building for ................... i686-pc-cygwin

Enabled transports:
 - USB ............................ yes
 - TCP ............................ yes


OpenOCD configuration summary
--------------------------------------------------
MPSSE mode of FTDI based devices        yes (auto)
ST-Link JTAG Programmer                 yes (auto)
TI ICDI JTAG Programmer                 yes (auto)
Keil ULINK JTAG Programmer              yes (auto)
Altera USB-Blaster II Compatible        yes (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
Olimex ARM-JTAG-EW Programmer           no
CMSIS-DAP Compliant Debugger            yes
Cypress KitProg Programmer              yes (auto)
Altera USB-Blaster Compatible           yes (auto)
ASIX Presto Adapter                     yes (auto)
OpenJTAG Adapter                        yes (auto)
SEGGER J-Link Programmer                yes (auto)
```

3. 执行编译命令开始编译

```
make
```

4. 执行安装命令将OpenOCD安装到系统路径下，安装比较耗时请耐心等待至结束

```
make install
```

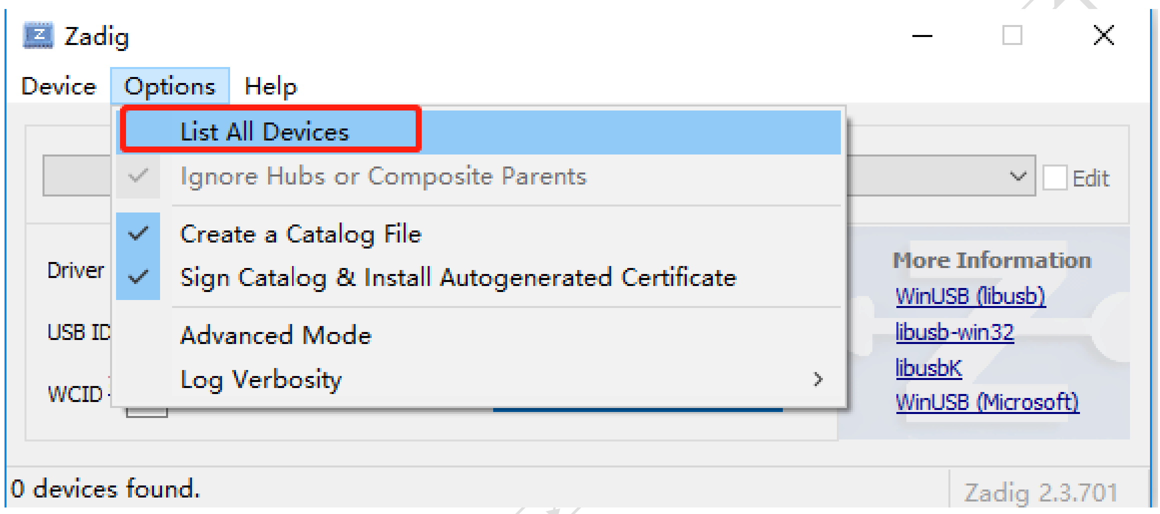
OpenOCD默认的的安装路径为/usr/local/bin,其配置文件默认的安装路径/usr/local/share/openocd。成功安装之后在/usr/local/share/openocd/scripts/target目录下会存在一个名为W60X.cfg的配置文件，之后OpenOCD启动就会用到它。

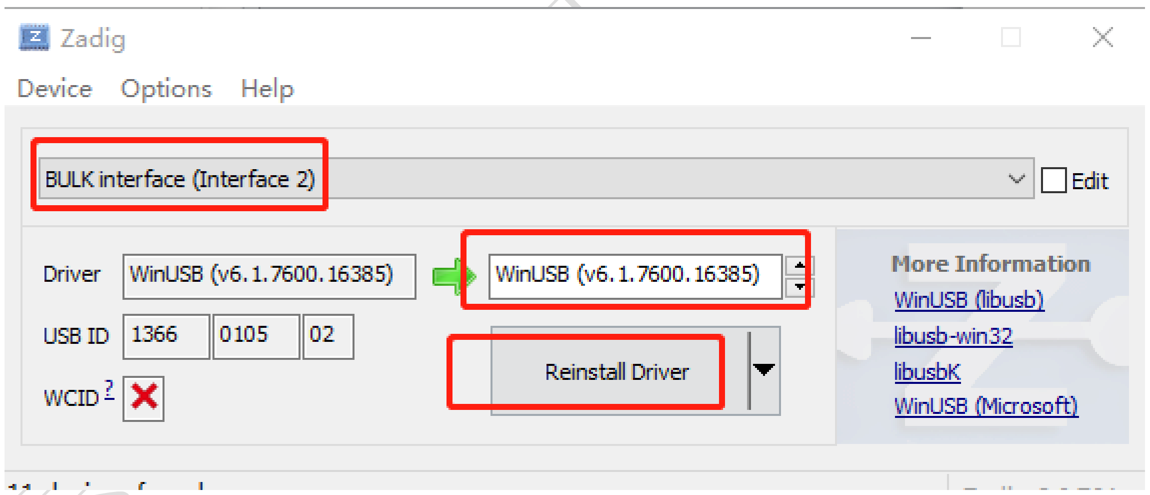
3.3 使用 OpenOCD 命令行调试

3.3.1 使用 JLINK 仿真器启动 OpenOCD

1. 安装驱动

JLINK 官方的驱动不能用于 OpenOCD，按照 OpenOCD 官方文档描述，需要使用 http://zadig.akeo.ie 提供的的驱动，下载安装如下图所示：

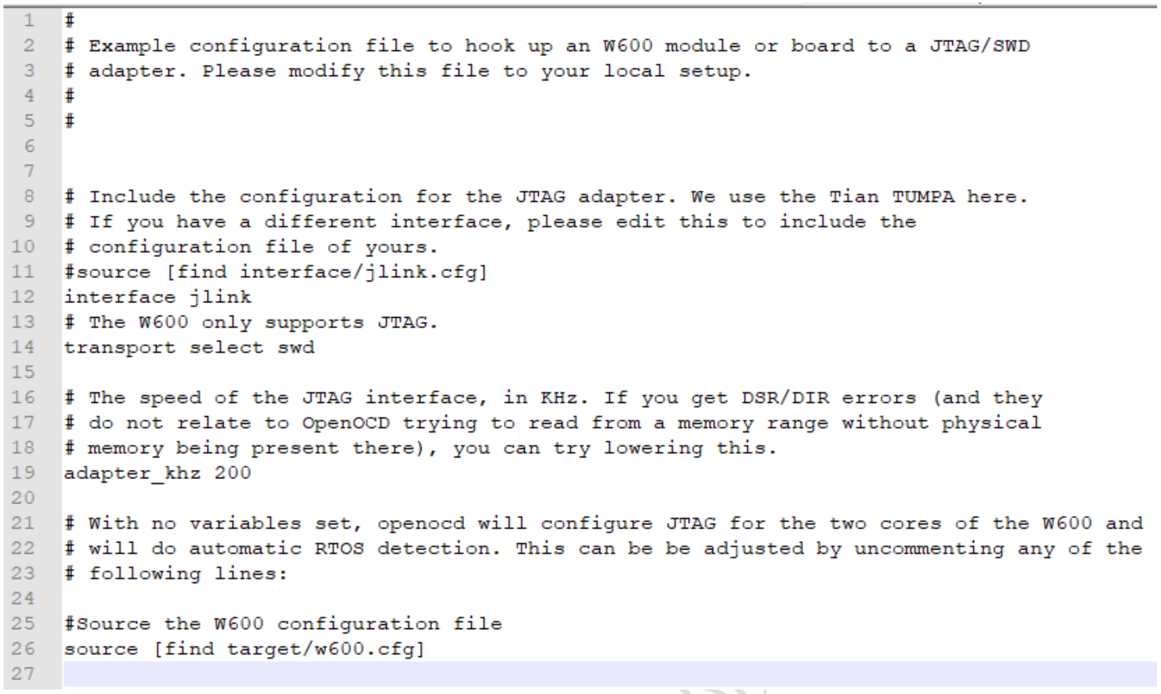




2. 创建配置文件

如果非使用 git 下载的源码包，可能在安装 OpenOCD 时就自动安装了该配置文件，可跳过这一步。

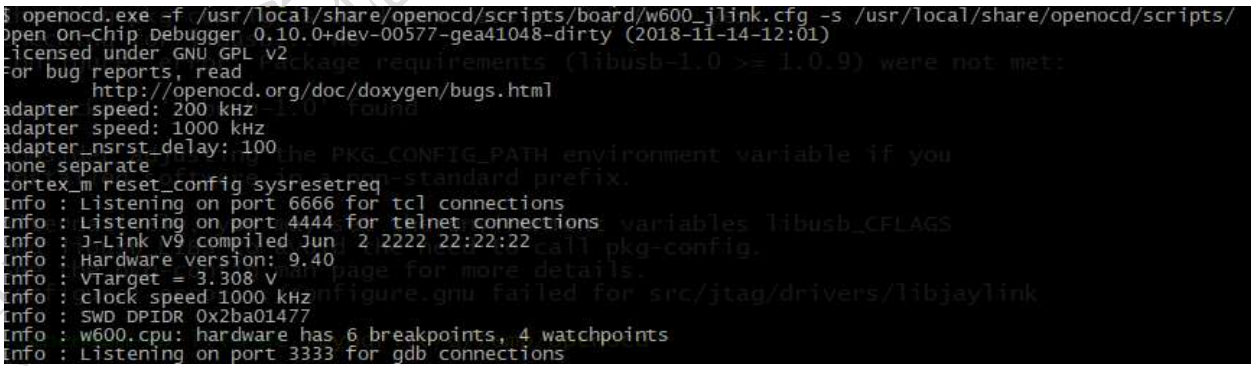
在/usr/local/share/openocd/scripts/board 下新建一个文本文件，取名为 W60X_jlink_cfg，然后用记事本打开编辑，输入内容如下图所示：



```
#
# Example configuration file to hook up an W600 module or board to a JTAG/SWD
# adapter. Please modify this file to your local setup.
#
#


# Include the configuration for the JTAG adapter. We use the Tian TUMPA here.
# If you have a different interface, please edit this to include the
# configuration file of yours.
#source [find interface/jlink.cfg]
interface jlink
# The W600 only supports JTAG.
transport select swd

# 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
# memory being present there), you can try lowering this.
adapter_khz 200

# With no variables set, openocd will configure JTAG for the two cores of the W600 and
# will do automatic RTOS detection. This can be be adjusted by uncommenting any of the
# following lines:

#Source the W600 configuration file
source [find target/w600.cfg]

```

保存后就创建好了 OpenOCD 连接 JLINK 的的配置文件。

3. 在 cygwin 的 shell 窗口中执行 openocd.exe -f /usr/local/share/openocd/scripts/board/W60X_jlink.cfg -s /usr/local/share/openocd/scripts/之后，如果 OpenOCD 没有异常退出提示则说明启动成功，此时结果如下：



```
$ openocd.exe -f /usr/local/share/openocd/scripts/board/w600_jlink.cfg -s /usr/local/share/openocd/scripts/
Open On-Chip Debugger 0.10.0+dev-00577-gea41048-dirty (2018-11-14-12:01)
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 : J-Link V9 compiled Jun  2 2222 22:22:22
Info : Hardware version: 9.40
Info : VTarget = 3.308 V
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
```

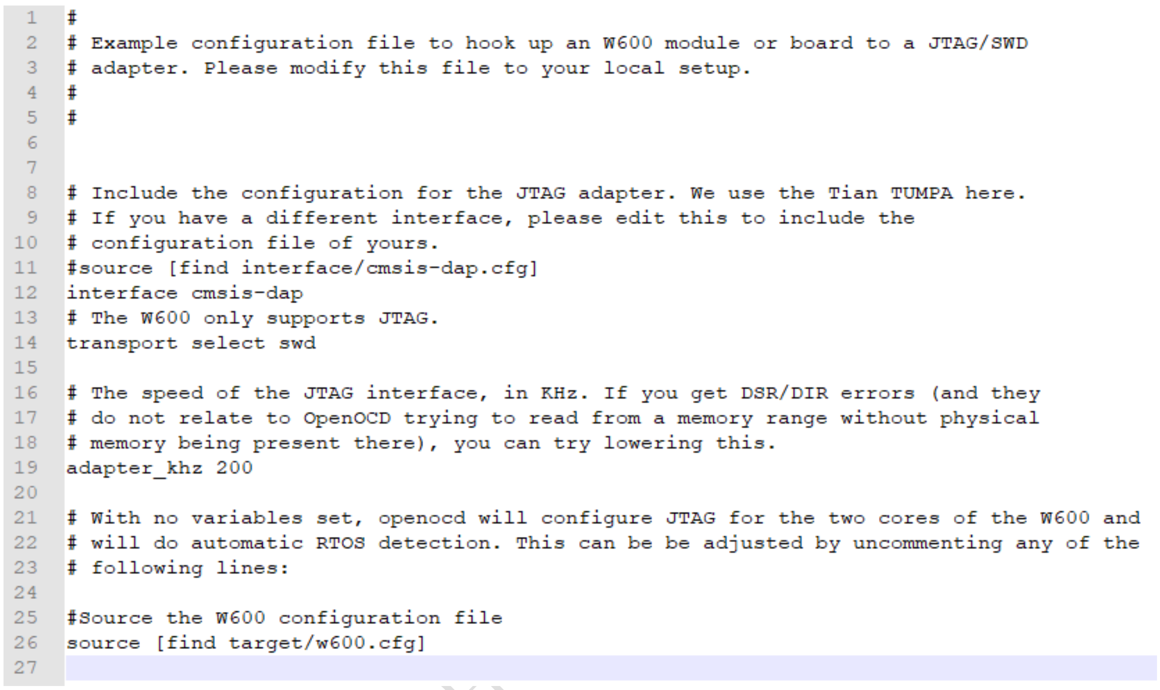
3.3.2 使用 CMSIS-DAP 仿真器启动 OpenOCD

1. 创建配置文件

如果非使用 git 下载的源码包，可能在安装 OpenOCD 时就自动安装了该配置文件，可跳过这一

步。

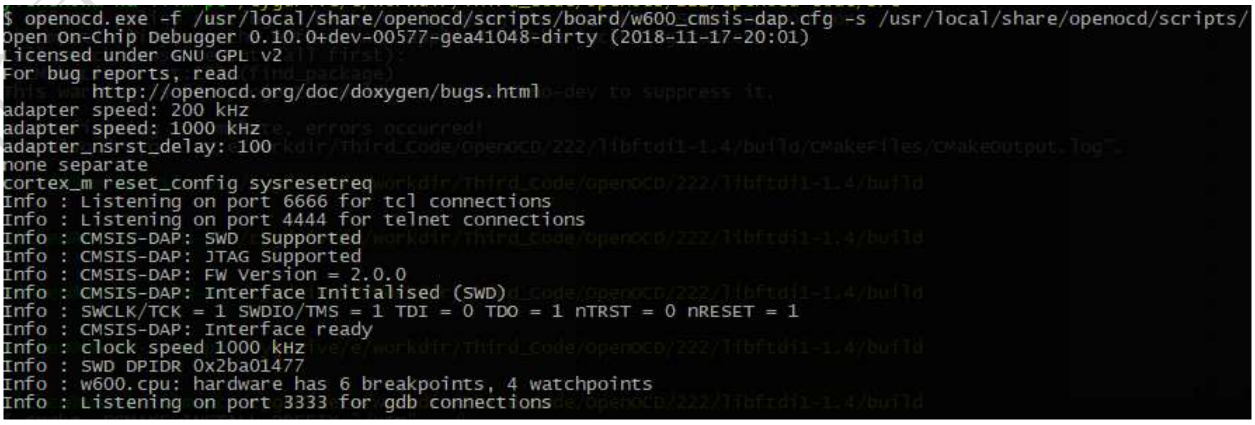
在/usr/local/share/openocd/scripts/board 下新建一个文本文件，取名为 W60X_cmsis-dap.cfg，然后用记事本打开编辑，输入内容如下图所示:



```
#
# Example configuration file to hook up an W600 module or board to a JTAG/SWD
# adapter. Please modify this file to your local setup.
#
#


# Include the configuration for the JTAG adapter. We use the Tian TUMPA here.
# If you have a different interface, please edit this to include the
# configuration file of yours.
#source [find interface/cmsis-dap.cfg]
interface cmsis-dap
# The W600 only supports JTAG.
transport select swd

# 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
# memory being present there), you can try lowering this.
adapter_khz 200

# With no variables set, openocd will configure JTAG for the two cores of the W600 and
# will do automatic RTOS detection. This can be be adjusted by uncommenting any of the
# following lines:

#Source the W600 configuration file
source [find target/w600.cfg]

```

保存后就创建好了 OpenOCD 连接 CMSIS-DAP 的的配置文件。

2. 在 cygwin 的 shell 窗口中执行 openocd.exe -f /usr/local/share/openocd/scripts/board/W60X_cmsis-dap.cfg -s /usr/local/share/openocd/scripts/之后，如果 OpenOCD 没有异常退出提示则说明启动成功，此时结果如下:



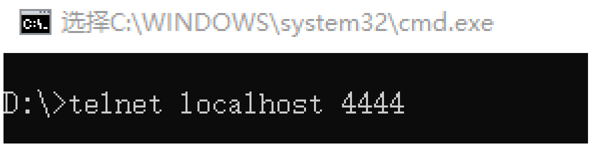
```
$ openocd.exe -f /usr/local/share/openocd/scripts/board/w600_cmsis-dap.cfg -s /usr/local/share/openocd/scripts/
Open On-Chip Debugger 0.10.0+dev-00577-gea41048-dirty (2018-11-17-20:01)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
adapter speed: 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
```

3.3.3 上位机连接 OpenOCD

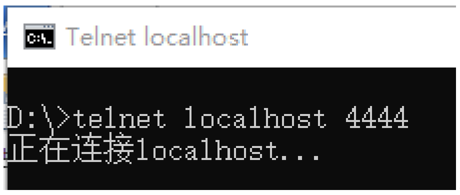
OpenOCD 在启动成功之后就可以通过 telnet 或 gdb 进行连接，tlenet 端口为 4444，gdb 端口为 3333。OpenOCD 在启动之后可以按 ctrl+c 键终止运行，启动之后请不要关闭 shell 窗口以保持 OpenOCD 的正常运行，telnet 或 gdb 才能连接调试。

3.3.3.1 使用 telnet 连接

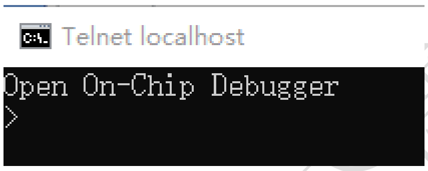
Windows 系统自带 telnel 客户端（如果提示未找到请参考 https://jingyan.baidu.com/article/e73e26c09f6f4724adb6a7de.html 安装），使用时打开一个命令提示符窗口，输入 telnet localhost 4444 或 telnet 127.0.0.1 4444 连接至 OpenOCD 守护进程，连接时效果如下图所示：


选择C:\WINDOWS\system32\cmd.exe

```
D:\>telnet localhost 4444
```

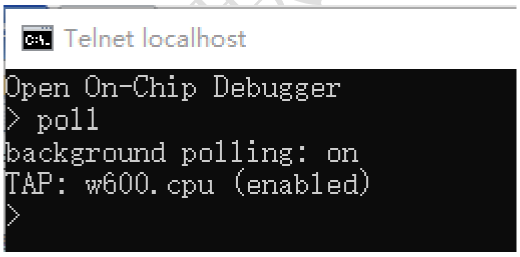

Telnet localhost

```
D:\>telnet localhost 4444
正在连接localhost...
```


Telnet localhost

```
Open On-Chip Debugger
>
```

之后就可以在命令行里键入命令进行调试，如下图：

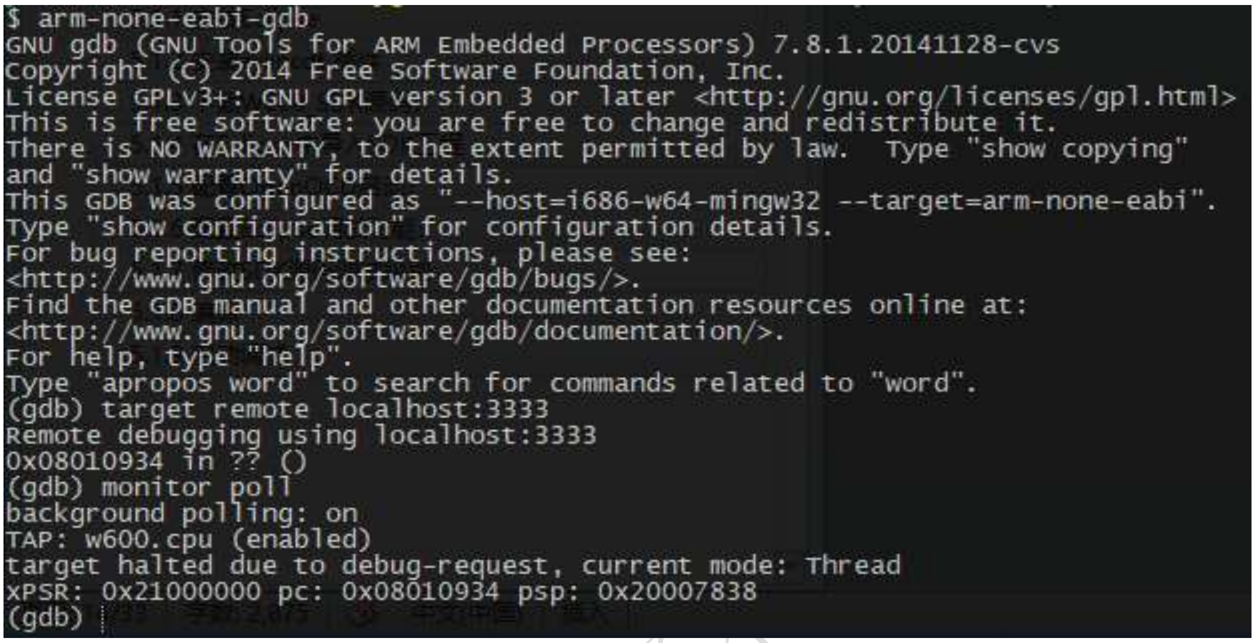

Telnet localhost

```
Open On-Chip Debugger
> poll
background polling: on
TAP: w600.cpu (enabled)
>
```

3.3.3.2 使用 gdb 连接

W60X 使用的交叉编译工具中的 gdb 为 arm-none-eabi-gdb，下载地址为：https://launchpad.net/gcc-arm-embedded/4.9/4.9-2014-q4-major。

如果是下载的我们提供的集成包，那么已经安装好了该工具，可以直接使用，在 cygwin 的 shell 窗口中执行命令 arm-none-eabi-gdb 即可启动 gdb，然后可以在 gdb 中敲入各种调试命令，如下图所示：



```
$ arm-none-eabi-gdb
GNU gdb (GNU Tools for ARM Embedded Processors) 7.8.1.20141128-cvs
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
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-mingw32 --target=arm-none-eabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) target remote localhost:3333
Remote debugging using localhost:3333
0x08010934 in ?? ()
(gdb) monitor poll
background polling: on
TAP: w600.cpu (enabled)
target halted due to debug-request, current mode: Thread
xPSR: 0x21000000 pc: 0x08010934 psp: 0x20007838
(gdb)
```

3.3.4 常用命令

下表列出了一些 OpenOCD 中常用的命令，在使用中也可以敲入 help 命令查看当前支持的所有命令。这些命令在 telnet 终端直接输入，但是在 gdb 终端里面，需要在命令前面添加 monitor。

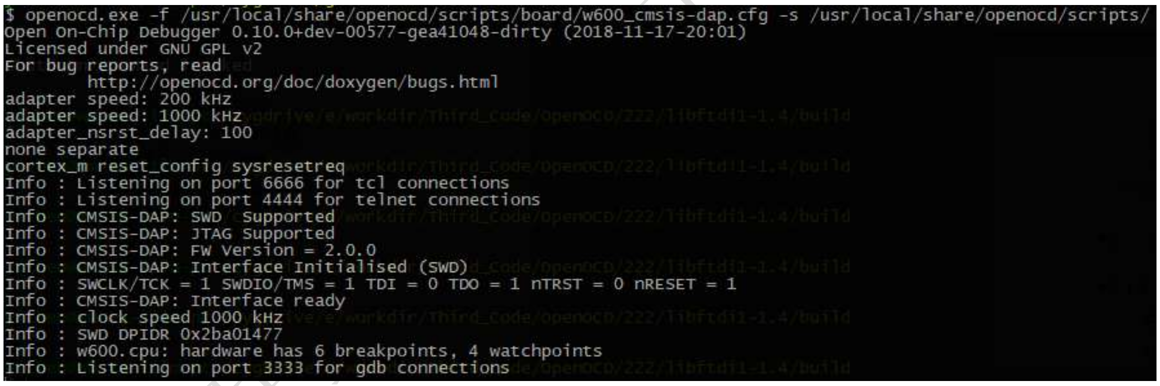
目标板状态处理命令(Target state handling)	
poll	查询目标板当前状态
halt	中断目标板的运行
resume [address]	恢复目标板的运行，如果指定了address，则从address 处开始运行
step [address]	单步执行，如果指定了address，则从address 处开始执行一条指令
reset	复位目标板
断点命令	
bp <addr> <length> [hw]	在地址addr 处设置断点，指令长度为length，hw 表示硬件断点
rbp <addr>	删除地址addr 处的断点
内存访问指令(Memory access commands)	
mdw ['phys'] <addr> [count]	显示从(物理)地址addr 开始的count(缺省是1)个字(4 字节)

mdh [’phys’] <addr> [count]	显示从(物理)地址addr 开始的count(缺省是1)个半字(2 字节)
mdb [’phys’] <addr> [count]	显示从(物理)地址addr 开始的count(缺省是1)个字节
mww [’phys’] <addr> <value>	向(物理)地址addr 写入一个字，值为value
mwh [’phys’] <addr> <value>	向(物理)地址addr 写入一个半字，值为value
mwb [’phys’] <addr> <value>	向(物理)地址addr 写入一个字节，值为value
flash write_image [erase] [unlock] filename [offset [file_type]] 将固件filename烧写到flash的offset地址处	

更加详细的 OpenOCD 指令介绍，可以直接参考其官网的文档，其地址为：
http://openocd.org/documentation/。

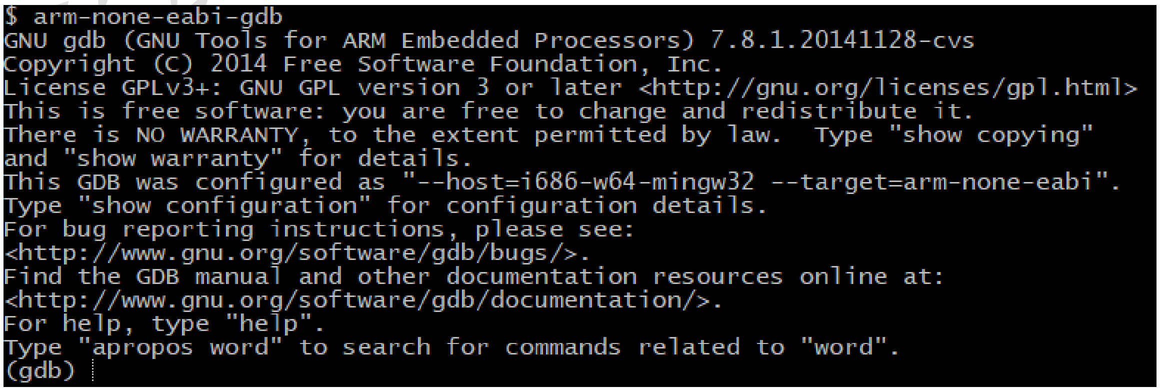
3.3.5 gdb 调试示例

本节使用 gdb 单步调试一段代码作为示例，指导用户如何开启 W60X 的单步调试之旅。
单步调试时需要有调试信息的固件和符号表，如何生成这样的文件请参阅后面章节的 Eclipse 编译 SDK。
这里用 CMSIS-DAP 仿真器连接 W60X，在连接好之后，在 cygwin 中启动 OpenOCD：



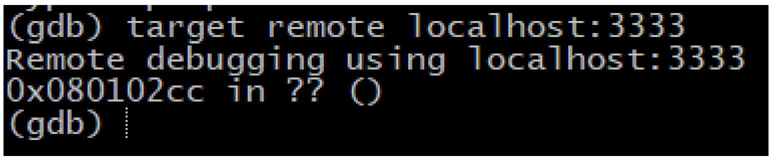
```
$ openocd.exe -f /usr/local/share/openocd/scripts/board/w600_cmsis-dap.cfg -s /usr/local/share/openocd/scripts/
Open On-Chip Debugger 0.10.0+dev-00577-gea41048-dirty (2018-11-17-20:01)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
adapter speed: 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
```

然后再新开一个 cygwin 窗口，执行 arm-none-eabi-gdb 程序：



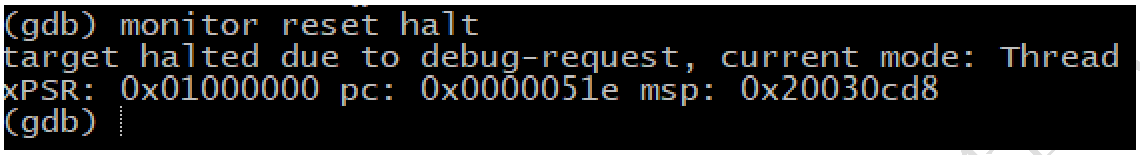
```
$ arm-none-eabi-gdb
GNU gdb (GNU Tools for ARM Embedded Processors) 7.8.1.20141128-cvs
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-w64-mingw32 --target=arm-none-eabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb)
```

然后使用 target remote localhost:3333 命令连接到 OpenOCD 守护进程上：



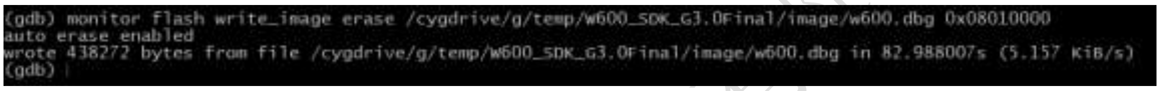
```
(gdb) target remote localhost:3333
Remote debugging using localhost:3333
0x080102cc in ?? ()
(gdb)
```

这时执行 monitor reset halt 命令，复位并停止 cpu 工作：



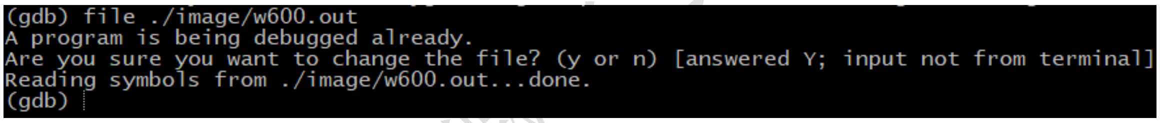
```
(gdb) monitor reset halt
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000051e msp: 0x20030cd8
(gdb)
```

接着执行 monitor flash write_image erase /cygdrive/g/temp/W60X_SDK_G3.0Final/image/W60X.dbg 0x08010000 命令烧写固件到 flash，烧写需要等待一会儿才能完成：



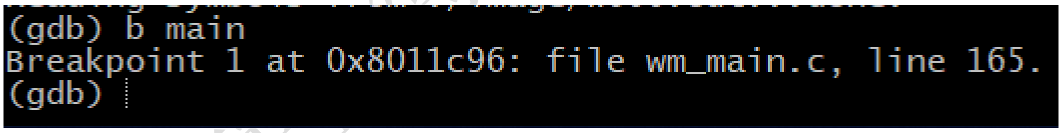
```
(gdb) monitor flash write_image erase /cygdrive/g/temp/w600_SDK_G3.0Final/image/w600.dbg 0x08010000
auto erase enabled
wrote 438272 bytes from file /cygdrive/g/temp/w600_SDK_G3.0Final/image/w600.dbg in 82.988007s (5.157 KiB/s)
(gdb)
```

然后执行 file ./image/W60X.out 加载符号表：



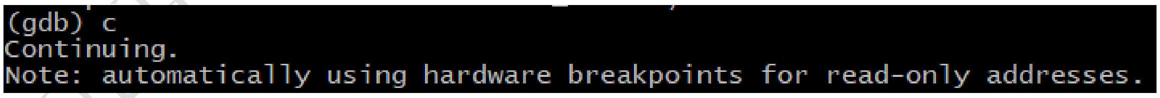
```
(gdb) file ./image/w600.out
A program is being debugged already.
Are you sure you want to change the file? (y or n) [answered Y; input not from terminal]
Reading symbols from ./image/w600.out...done.
(gdb)
```

然后执行 b main 设置一个断点，便于在进入 main 函数时停住：



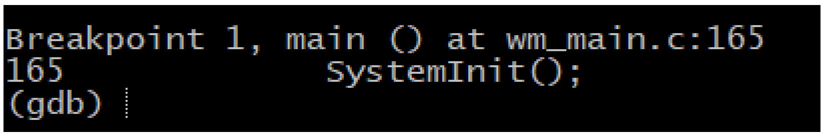
```
(gdb) b main
Breakpoint 1 at 0x8011c96: file wm_main.c, line 165.
(gdb)
```

最后执行 c 命令让程序跑起来：



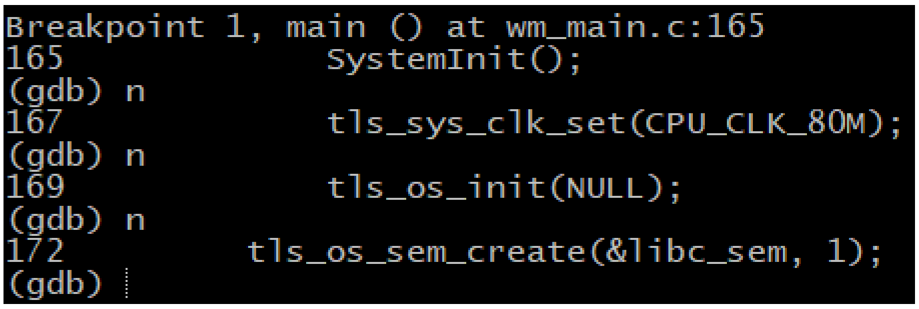
```
(gdb) c
Continuing.
Note: automatically using hardware breakpoints for read-only addresses.
```

很快程序就停在了 main 函数入口处：



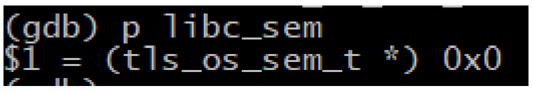
```
Breakpoint 1, main () at wm_main.c:165
165             SystemInit();
(gdb)
```

这时即可使用 n 命令单步调试：



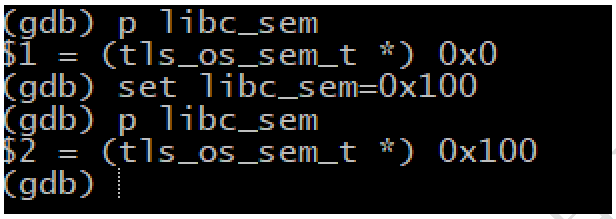
```
Breakpoint 1, main () at wm_main.c:165
165             SystemInit();
(gdb) n
167             tls_sys_clk_set(CPU_CLK_80M);
(gdb) n
169             tls_os_init(NULL);
(gdb) n
172         tls_os_sem_create(&libc_sem, 1);
(gdb)
```

可以使用 p libc_sem 打印出指针地址：



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

使用 set libc_sem=0x100 修改指针地址：



```
(gdb) p libc_sem
$1 = (tls_os_sem_t *) 0x0
(gdb) set libc_sem=0x100
(gdb) p libc_sem
$2 = (tls_os_sem_t *) 0x100
(gdb)
```

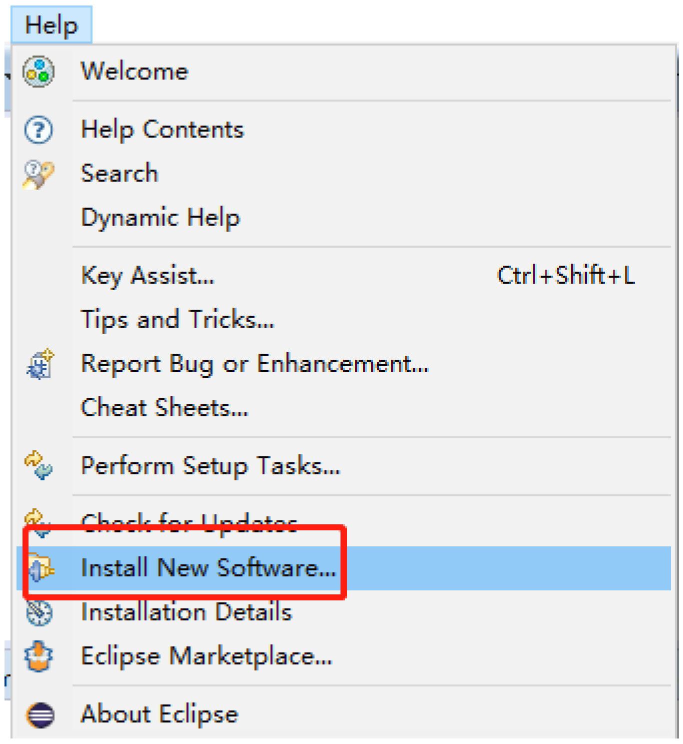
关于更多的 gdb 调试知识请自行参阅其文档 http://sourceware.org/gdb/current/onlinedocs/gdb/。

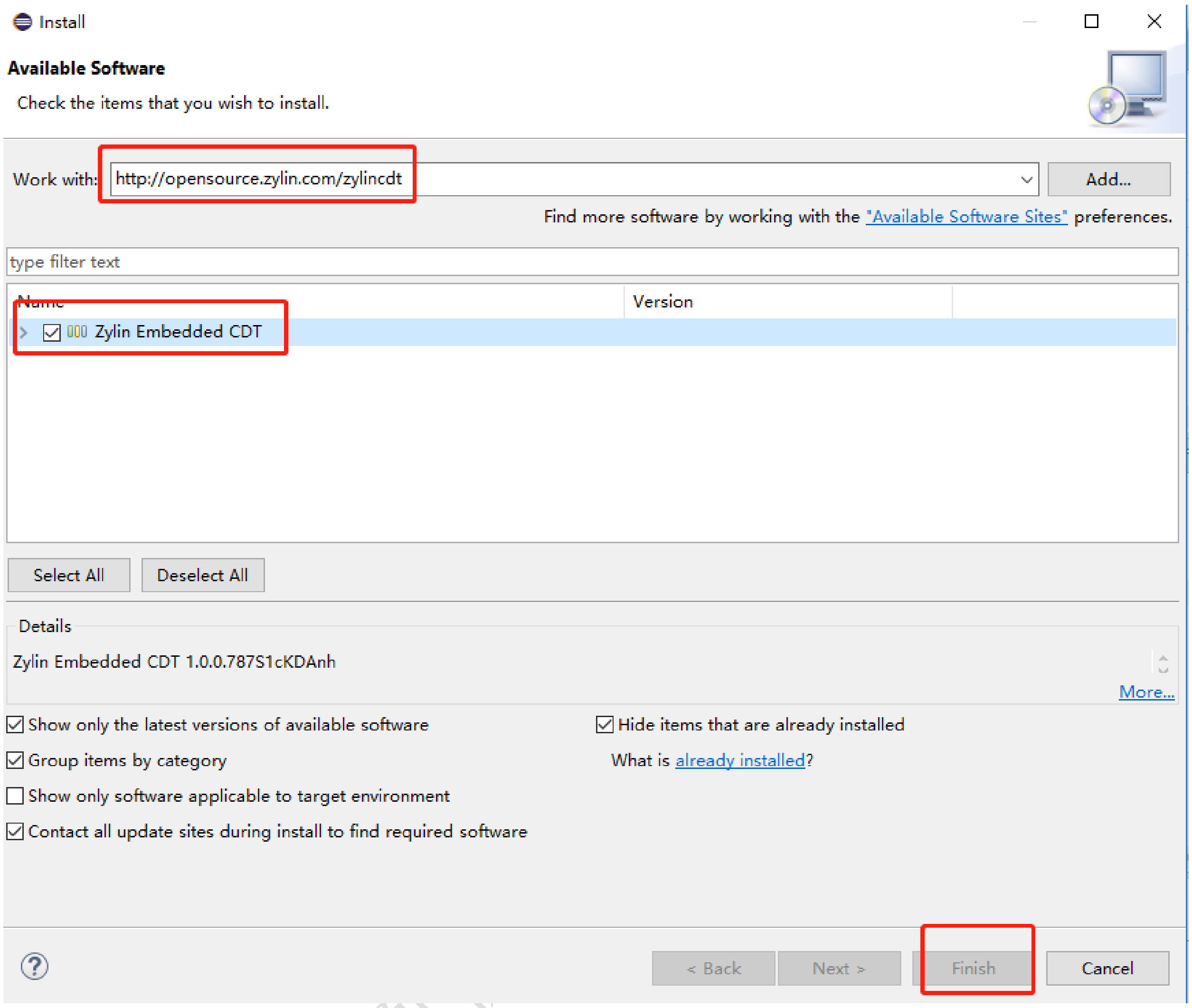
4 附录

4.1 Eclipse 安装 zylincdt 插件

Zylincdt 插件使得 Eclipse 能够支持嵌入式 GDB 在 Eclipse 中调试，我们提供的集成压缩包开发环境中已经默认安装了该插件。

插件更新源为 http://opensource.zylin.com/zylincdt，插件安装方法如下图所示：


Help
Welcome
Help Contents
Search
Dynamic Help
Key Assist... Ctrl+Shift+L
Tips and Tricks...
Report Bug or Enhancement...
Cheat Sheets...
Perform Setup Tasks...
Check for Updates
Install New Software...
Installation Details
Eclipse Marketplace...
About Eclipse


Install

Available Software

Check the items that you wish to install.

Work with: http://opensource.zylin.com/zylincdt

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☑ Zylin Embedded CDT	

Details

Zylin Embedded CDT 1.0.0.787S1cKDAnh

☑ Show only the latest versions of available software
☑ Group items by category
☐ Show only software applicable to target environment
☑ Contact all update sites during install to find required software
☑ Hide items that are already installed
What is already installed?

安装时会有一些警告，最后会提示需要重启 Eclipse，按照提示操作即可。