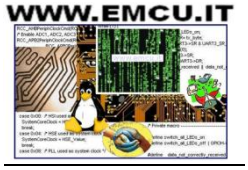
# Eclipse + Codesourcy + OpenOCD + GCC, for develop and debug on STM32Fo family using LINUX.

**By:**
**Nicolas Fillon**
**ST** FAE (USA) – Santa Clara (California).



## INTRODUCTION

This guide's intent and purpose is to allow a user with minimal Linux experience to successfully setup an **absolutely free** development environment to program the STM32Fo microcontroller with full debugging capability. Other STM32F families can be substituted.

## PREREQUISITES

1. A Linux distribution, this example uses Linux Mint 13 (Ubuntu works nicely too)
2. An Internet connection
3. At least 2GB of spare hard drive space
4. Familiarity with TERMINAL
5. An STM32Fo Discovery Board!

## PART 1 – Install Codesourcy

If you are using 64bit Linux, install ia32-libs with TERMINAL by inputing:

*sudo apt-get install ia32-libs*

Download and install Sourcery G++ Lite. In this example, we use Sourcery G++ Lite 2011.03-42 for ARM EABI. Newer versions might not work properly.
https://sourcery.mentor.com/sgpp/lite/arm/portal/subscription?@template=lite

*sh arm-2011.03-42-arm-none-eabi.bin*

If the error message "ERROR: DASH shell not supported as system shell" is displayed, use this command to change the shell:

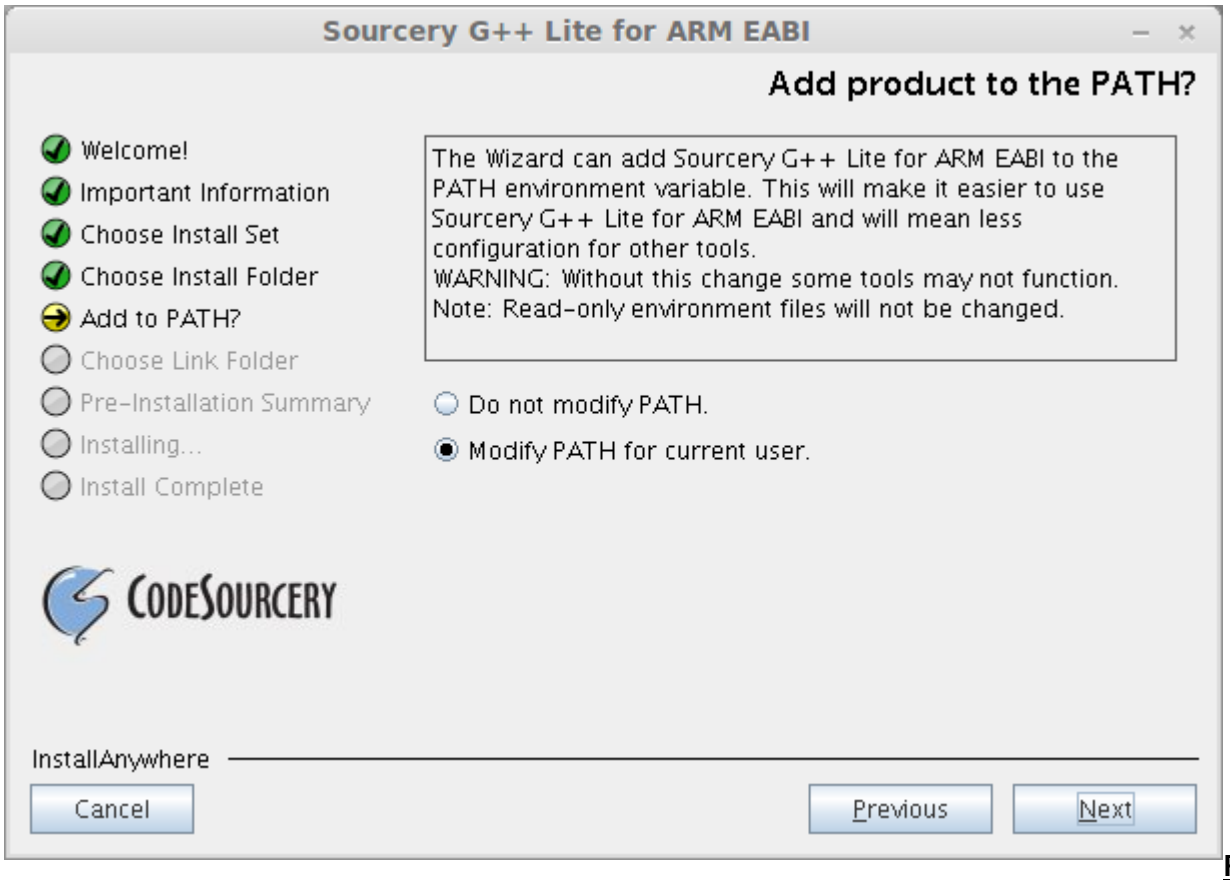
*sudo dpkg-reconfigure -plow dash*

It can now be installed by using the command:

*~/Desktop/arm-2011.03-42-arm-none-eabi.bin*

Follow the prompts and configure as listed below

1. Choose install set
   - Minimal Installation
2. Choose Install Folder
   - Default folder (/home/zach/CodeSourcery/Sourcery_G++_Lite)
3. Add to Path?
   - Check *Modify path for user*
4. Choose Link Folder
   - Check *Don't create links*




**PAR**

**T 2 – Install OpenOCD**

Open a terminal and run each command.

*sudo apt-get install build-essential*
*sudo apt-get install git*
*sudo apt-get install libtool*
*sudo apt-get install libftdi1*
*sudo apt-get install texinfo*
*git clone http://repo.or.cz/r/openocd.git/*
*cd '/home/zach/openocd/'*
*./configure --prefix=/usr --enable-maintainer-mode --enable-stlink^C*
*make*
*git clone https://github.com/szczys/stm32fo-discovery-basic-template.git*
*git clone git://github.com/texane/stlink.git*

This should create a directory named *openocd*, a directory named *stm32fo-discovery-basic-template*, and *stlink* all in the local directory (/home/zach/).

Install the udev rules for our st-link device, so that a device "/dev/stlink" is accessible non-root. The needed rules are part of the sources that come with the file "10-stlink.rules".

*cd ~/stlink/*

*sudo install -m 644 49-stlinkv2.rules /etc/udev/rules.d/49-stlinkv2.rules*

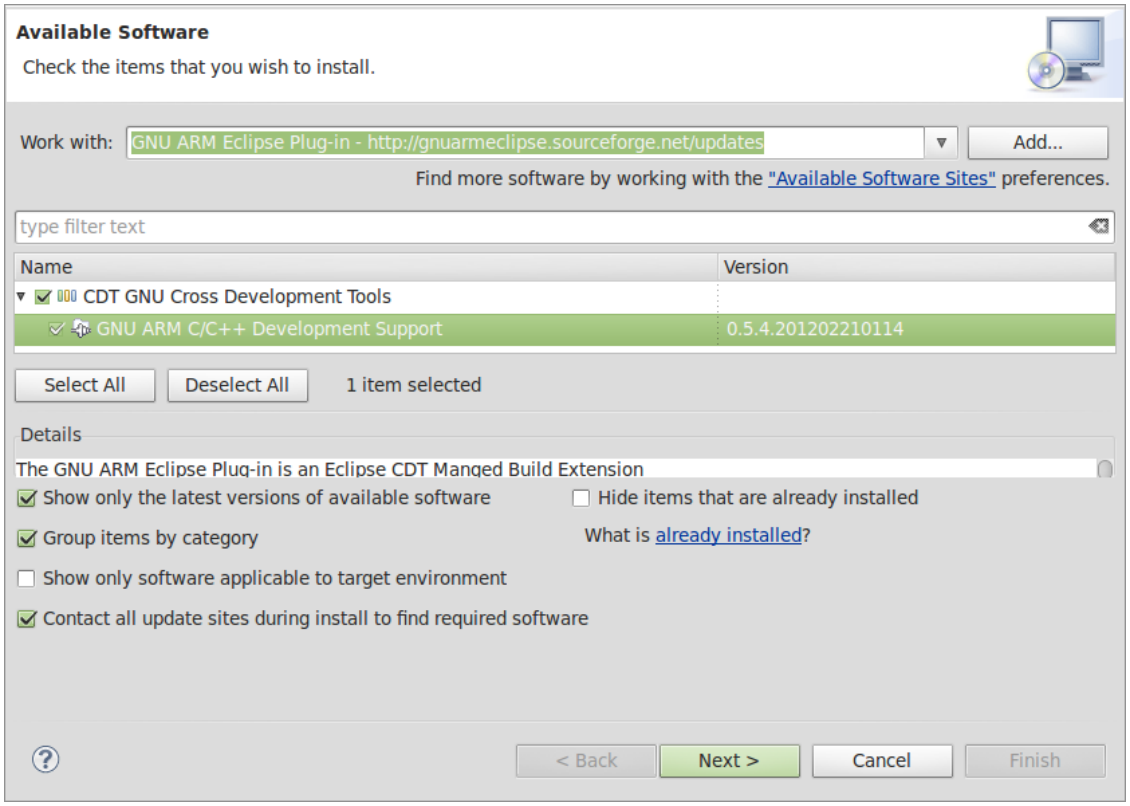
To activate the new rules either reboot, or:

*sudo udevadm control --reload-rules*

## PART 3 – Install Eclipse

1. Download Eclipse C/C++ Development Edition . The specific edition used here is:
   Eclipse IDE for C/C++ Developers Version: Indigo Release Build id: 20110615-0604

2. Download/Install GNU ARM Eclipse plug-in, go to indigo "Help"->"Install New software", Work with: GNU ARM Eclipse Plug-in - http://gnuarmeclipse.sourceforge.net/updates




## PART 4 – Setup File Directory

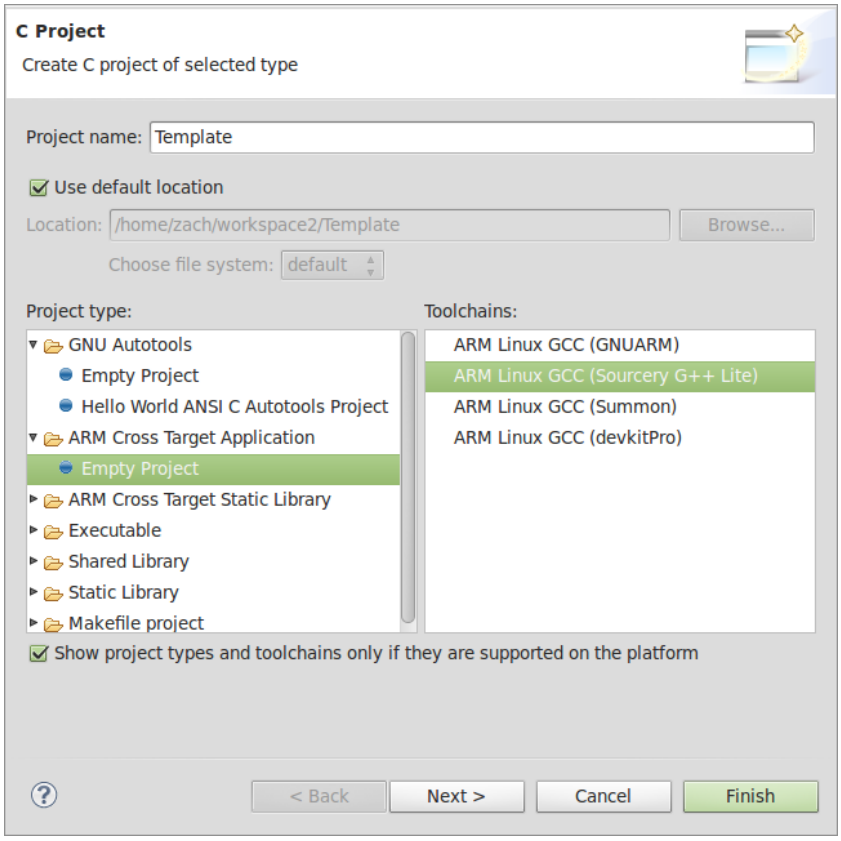
1. Download the STM32FoDiscovery Firmware package http://www.st.com/internet/com/SOFTWARE_RESOURCES/SW_COMPONENT/FIRMWARE/stm32fodiscovery_fw.zip
2. Extract the contents and copy the top level, Libraries folder to the workspace/project folder i.e /home/zach/workspace/Template/
3. Copy *stm32foxx_conf.h* from an example project in the STM32fo firmware package
4. In the Template folder create a new folder called *src*. Save all your source files in that location.
5. Copy and paste the files located in /home/zach/stm32fo-discovery-basic-template/Device/ldscripts to your project's src folder (~/workspace/Template/src/)
6. Open the *stm32fo.ld* file which now resides in your src folder, and modify the blue highlighted lines to match below

```
/*
Default linker script for STM32F051R8T6 64k 8k
Copyright RAISONANCE S.A.S. 2007

!!! This file is automatically generated by RIDE !!!
Do not modify it, as it will be erased at every link.
You can use, copy and distribute this file freely, but without any warranty.
*/

/* include the memory spaces definitions sub-script */
INCLUDE "/home/zach/workspace/Template/src/stm32fodiscovery_def.ld"

/* include the sections management sub-script for FLASH mode */
INCLUDE "/home/zach/workspace/Template/src/sections_flash.ld"
```

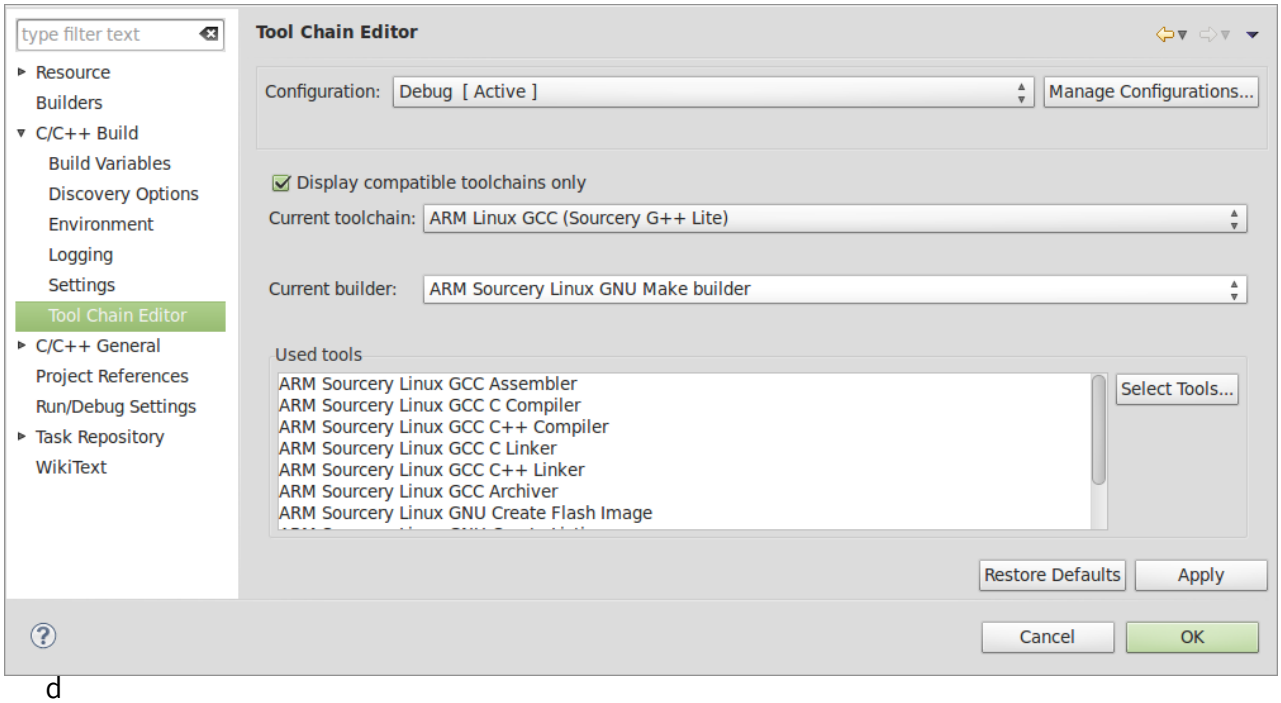
7. Rename *startup_stm32foxx.s*, located in ~/stm32fo-discovery-basic-template/and change the .s to .S
8. Copy the renamed file and paste it into the *src* directory.
9. Copy the ~/stm32fo-discovery-basic-template/extra/ to ~/workspace/Template/

## PART 5 – Setup Eclipse

1. Create an empty C Project

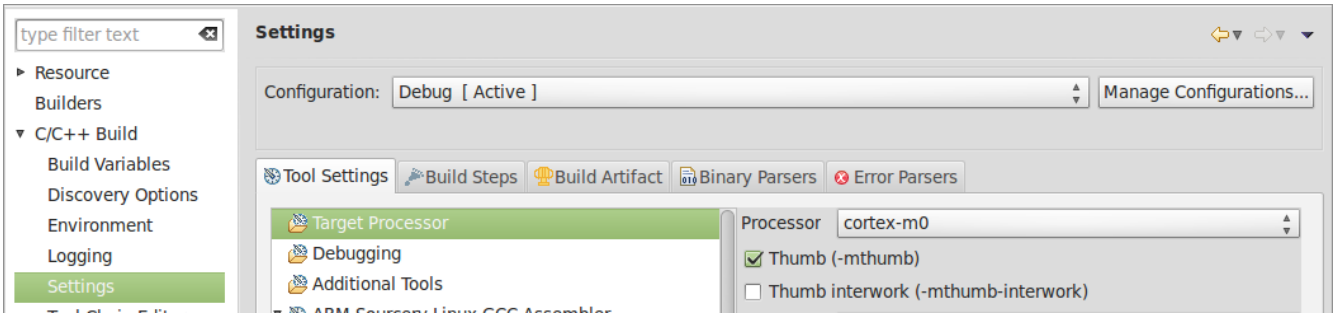



2. Set Tool chain

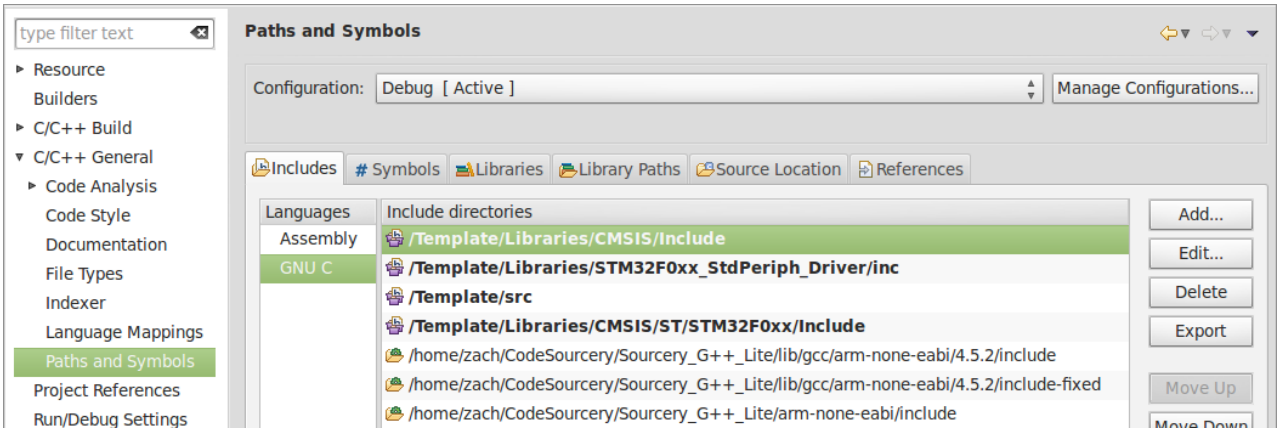



d

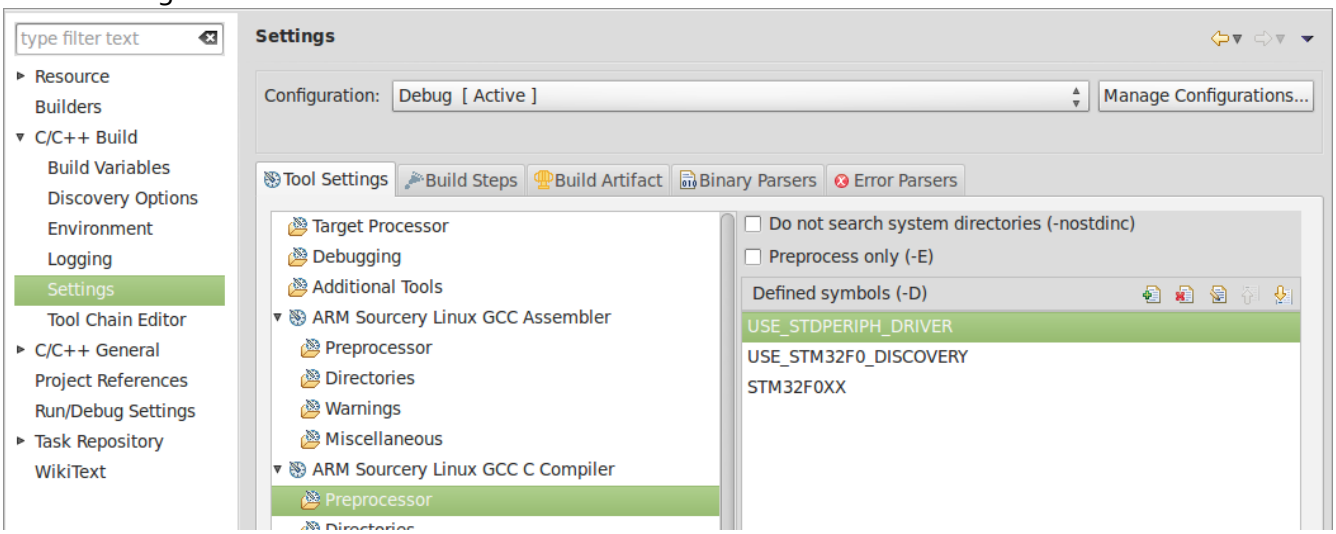
. Change Processor to cortex-mo



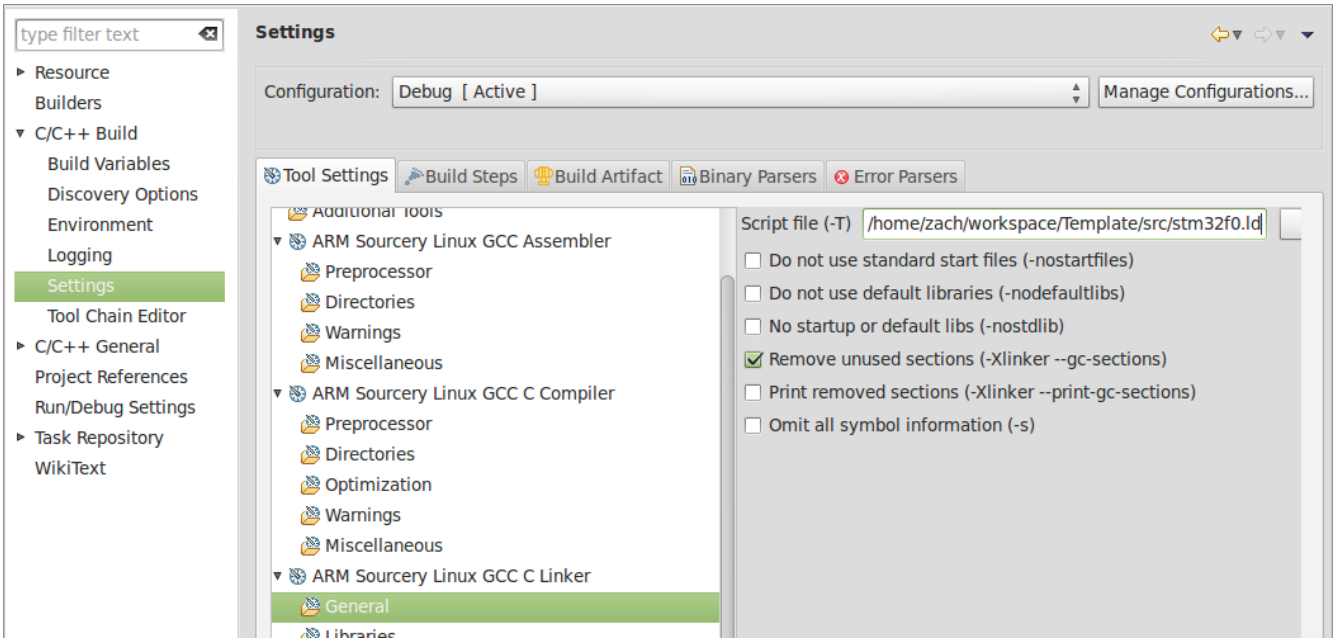

4. Paths and Symbols (Set for both Assembly SourceFile & GNU C). To add each path quickly, add every path as a workspace path
   /Template/Libraries/CMSIS/Include
   /Template/Libraries/STM32Foxx_StdPeriph_Driver/inc
   /Template/src
   /Template/Libraries/CMSIS/ST/STM32Foxx/Include




5. Symbols in Settings
   USE_STDPERIPH_DRIVER
   USE_STM32F0_DISCOVERY
   STM32F0XX

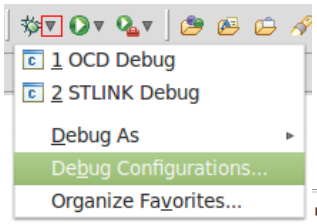



6. Linker setting. Check the Remove unused sections. Change the textbox so as to match the location of your stm32f0.ld file: /home/zach/workspace/Template/src/stm32f0.ld .

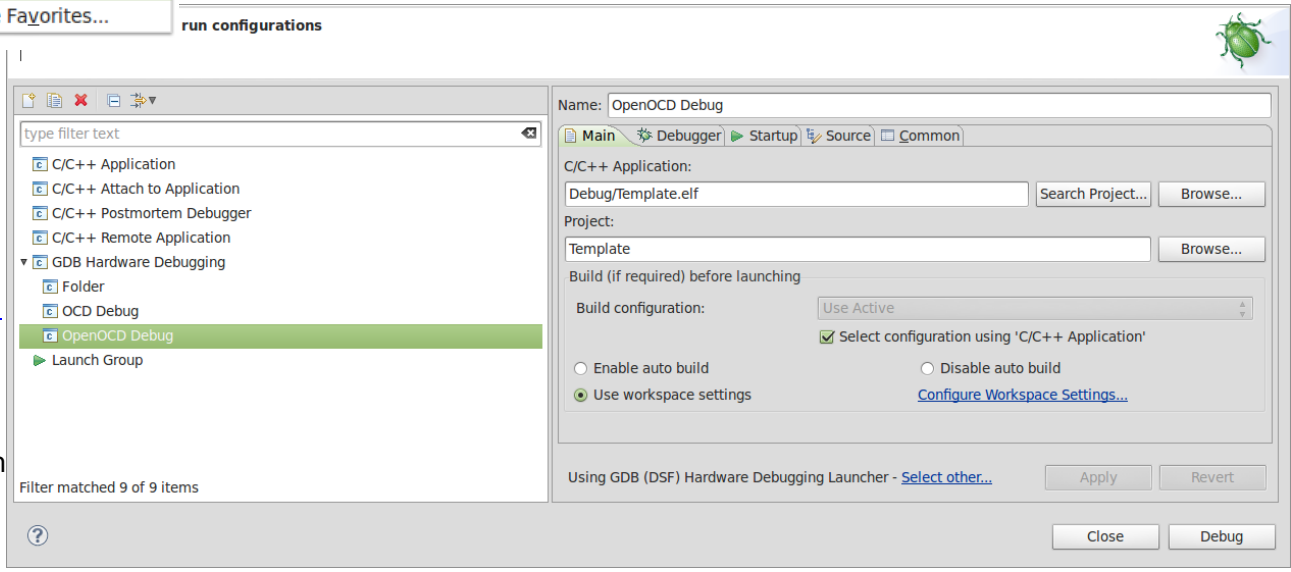



## PART 6 – Configure the gdb/OpenOCD

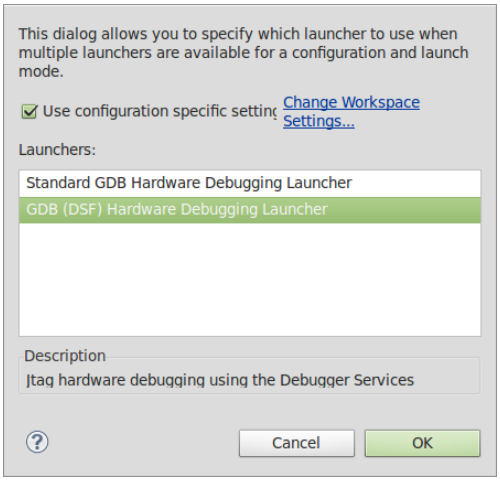
1. Open Debug Configurations… by clicking on the down arrow of the green bug, and selecting Debug Configurations.



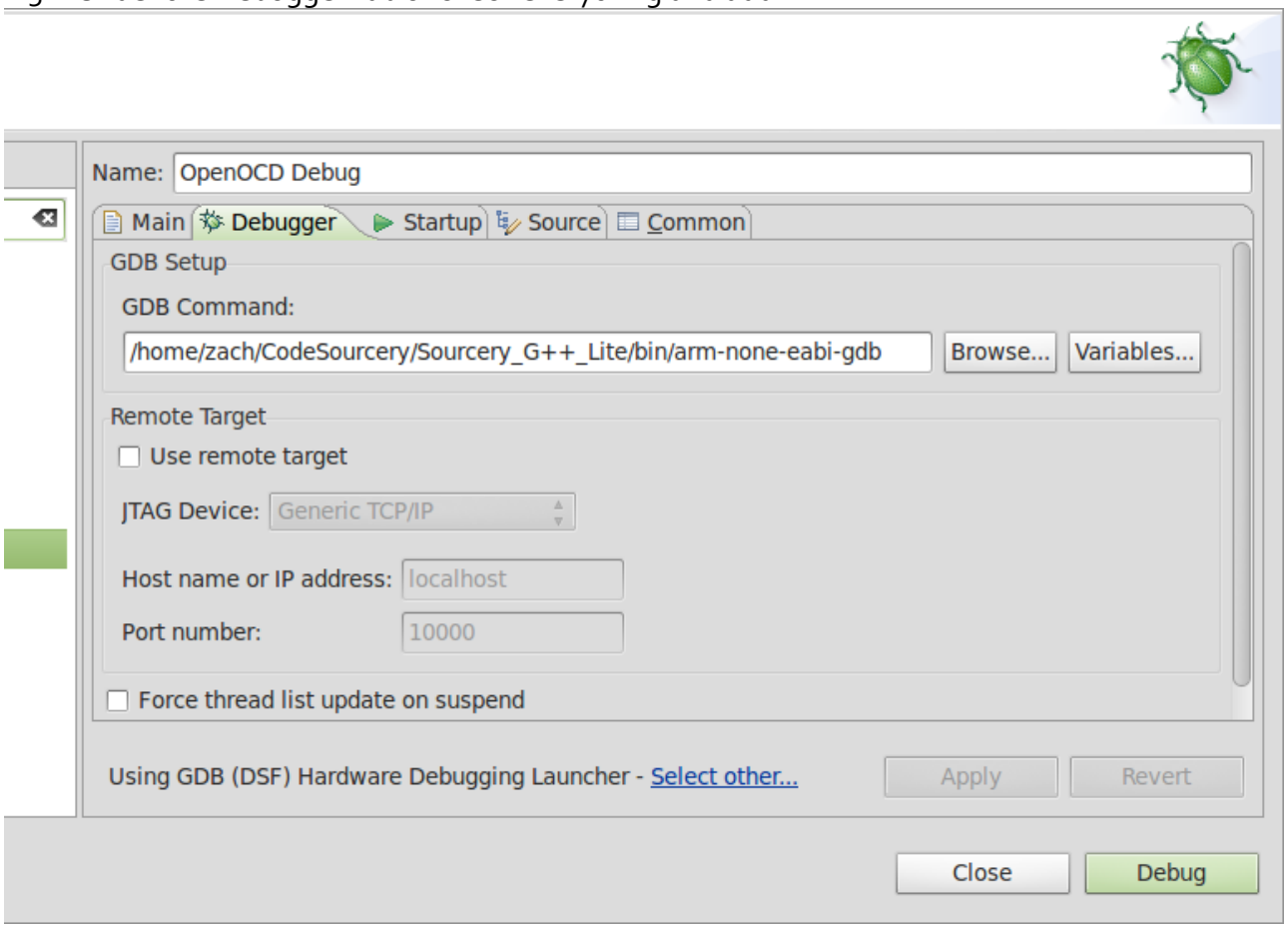

2. On the pop-up window, create a new launch configuration by double clicking on the text GDB Hardware Debugging
3. Change Template Debug to OpenOCD Debug and make sure the other values match the ones below




4. On the same tab click Select other… which resides on the bottom on the page. Select GDB (DSF) Hardware Debugging Launcher. Click OK.



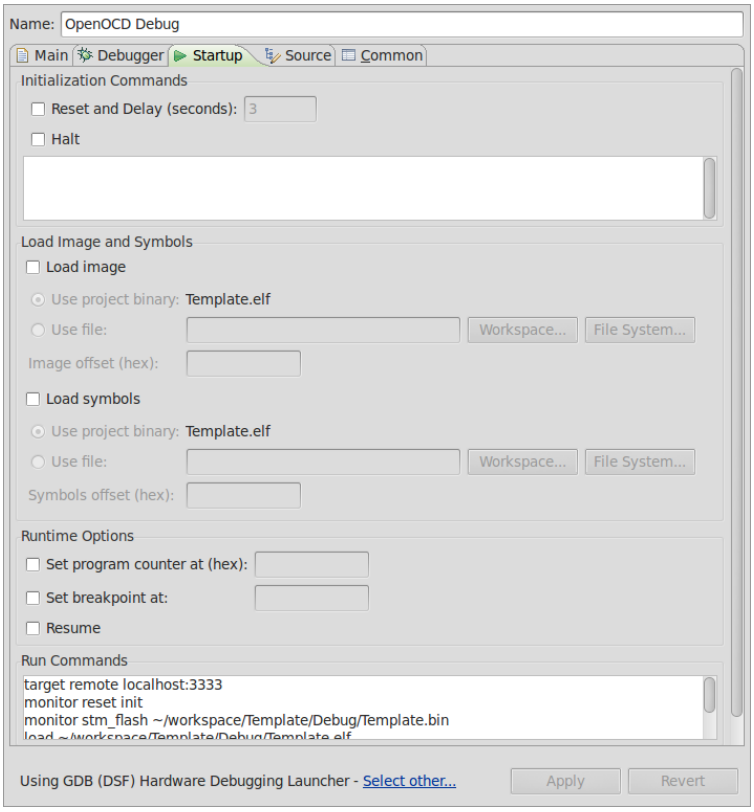

5. Under the Debugger Tab uncheck everything and add



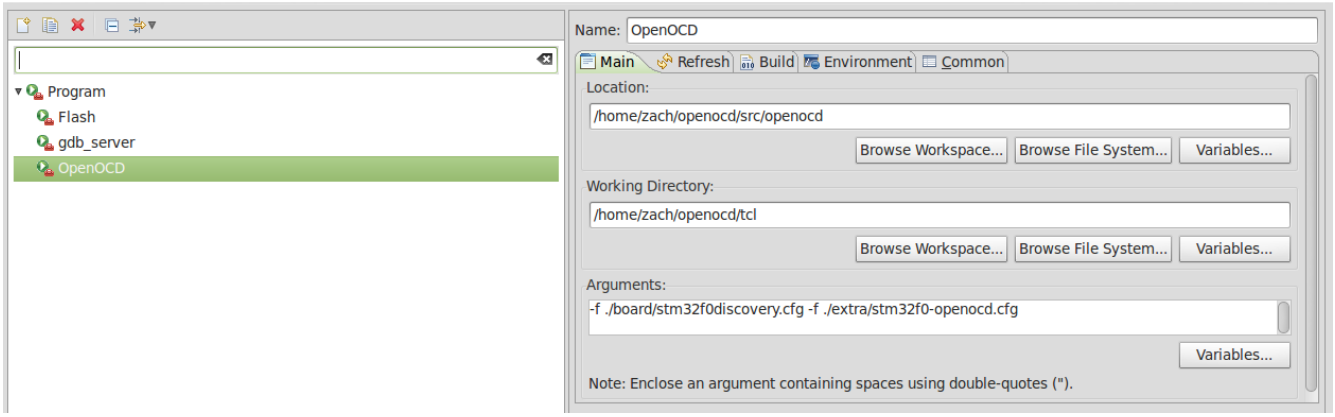

GDB Command: /home/zach/CodeSourcery/Sourcery_G++_Lite/bin/arm-none-eabi-gdb

6. Under the Startup Tab uncheck everything. In the Run Commands: you should have something similar. target remote localhost:3333

monitor reset init
monitor stm_flash ~/workspace/Template/Debug/Template.bin
load ~/workspace/Template/Debug/Template.elf
symbol-file ~/workspace/Template/Debug/Template.elf
cont

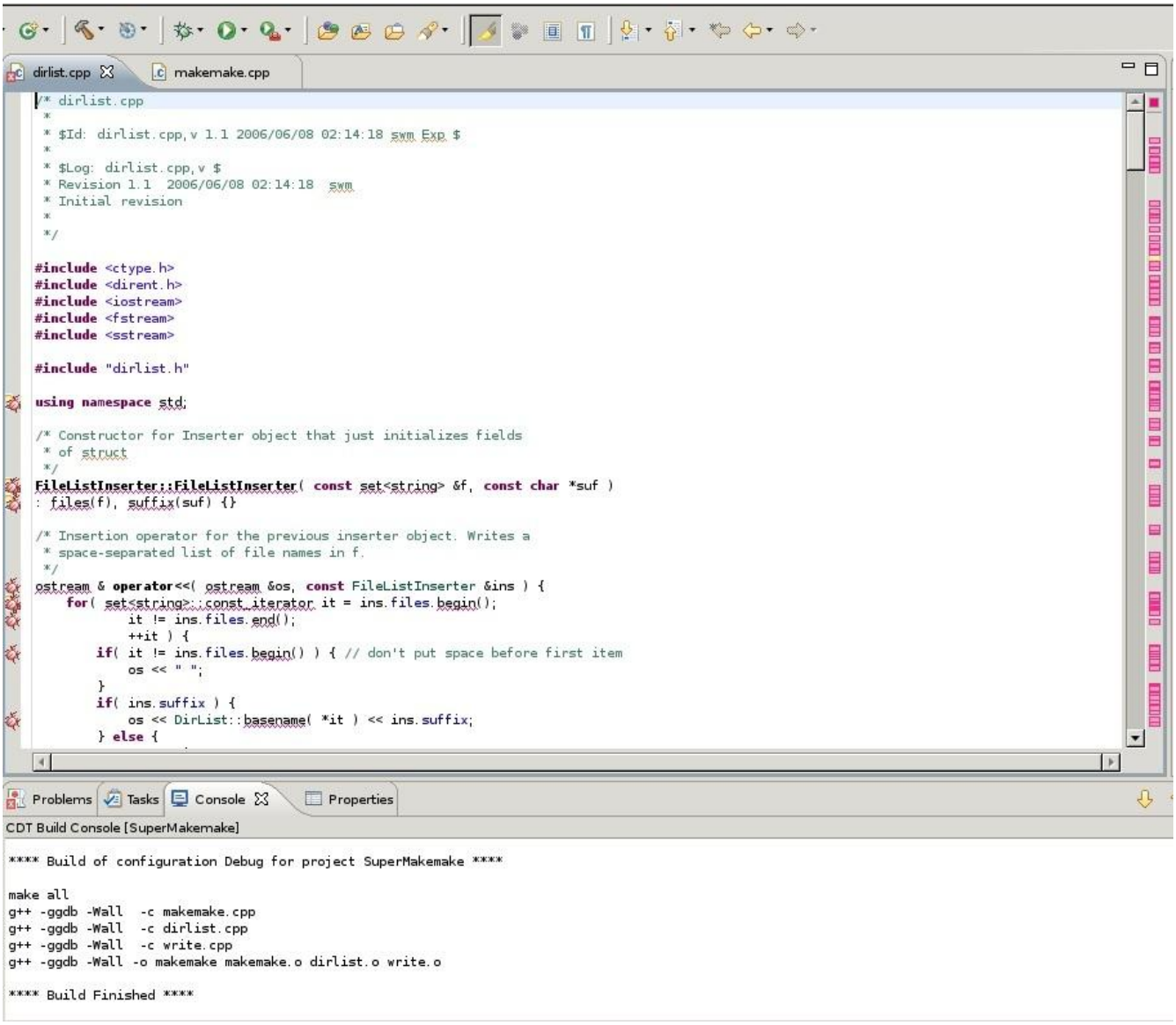



7. Press Apply. Under External Tools Configurations, add the
Location:/home/zach/openocd/src/openocd
Working directory: /home/zach/openocd/tcl
Arguments: -f ./board/stm32fodiscovery.cfg -f ./extra/stm32fo-openocd.cfg




## APPENDIX A – Erroneous Errors

If for some reason you build too soon, errors will always appear in your project even though your project compiles fine. CDT is unfortunately at fault and this problem can easily be fixed.




(Example of error-just not with gcc)

Right click on your Template project
Highlight index
Click Rebuild

If this doesn't work delete your errors on the problems tab (right click, select delete) and try the Appendix A process again.

**LICENSE**

*You may not sell this file without the original author's permission
*Accreditation must be given to the original author/authors when republishing this document, or modified versions of this document
*The original author/authors retain ownership of this document
*Unless required by applicable law or agreed to in writing, this guide is distributed on an "AS IS"
*BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.