Freedom Studio User Manual

2019-05-0

Table of Contents

Introduction **5**

Product Overview 5

Eclipse 5

Setting Up Freedom Studio **6**

Download and Install 6

Windows Installation 6

MacOS Installation 8

Linux Installation 8

Contents 9

Tools Setup 9

The Freedom Studio Environment **10**

Workspaces 10

Eclipse Perspectives 10

The SiFive Perspective 11

Project Explorer 12

Editor, Outline, Disassembly 13

Terminal 14

Breakpoints 15

Registers 16

Expressions 16

Getting Started **17**

The First Run Dialog **17**

Flashing Your Arty FPGA Board 19

Before you continue 19
Flash the Arty Board 19
Windows Only 20
Create a Freedom E SDK Project 21
Freedom Studio HOWTO Guides 24
Register List Management 24
A Quick Example 24
Creating Register List Files 26
Commenting the Register List File 26
Specifying Register Names 26
Single Registers 26
Built-in Macros 26
Include File 26
Register Ordering 27
Using Register List Files 27
Where To Specify a Register List File 27
Managing Hardware Breakpoint Resources 30
Option 1: Add a gdb initialization command 31
Option 2: Set a preference or project property 32
Valid settings 33
Setting the Global Preference 33
Setting the Workspace Preference 34
Setting the Project Property 35
Setting the Launch Configuration Attribute 36
Conditional Optimization 37
Known Issues 42

When the debugger first connects I receive message saying "No source available for address" 42

Upon starting a debug connection, the Console prints out a lot of text in red colored font 42

Trouble Shooting **42**

Linux USB Permission Issues 42

Target Board Setup **42**

Windows Board Setup **42**

Windows JLink USB Driver 43

macOS Board Setup **43**

Linux OS Board Setup **44**

Required Libraries 44

Let’s Check Our Dependencies 44

Enable Access to USB Devices 46

SiFive Copyright Notice **48**

Software Licenses **48**

SiFive End User License Agreement **48**

Eclipse Public License - v 2.0 **48**

GNU GENERAL PUBLIC LICENSE, V2 **54**

GNU GENERAL PUBLIC LICENSE, V3 **62**

Introduction

Freedom Studio is an integrated development environment which can be used to write and debug software targeting SiFive based processors. Freedom Studio is based on the industry standard Eclipse platform and is bundled with a pre-built RISC-V GCC Toolchain, OpenOCD, and the freedom-e-sdk. The freedom-e-sdk is a complete software development kit targeting SiFive bare metal processors.

Product Overview

This section will describe the individual components used in a release.

Eclipse

The major versions of the Eclipse feature plugins are as follows:

- Eclipse Oxygen
- Eclipse C/C++ Development Tools
- Git Integration for Eclipse (eGit)
- Terminal View Core
- SiFive RISC-V Cross Compiler
- SiFive OpenOCD Debugging

- SiFive J-LINK Debugging
- SiFive QEMU Debugging
- SiFive freedom-e-sdk Project Template

Setting Up Freedom Studio

Download and Install

Freedom Studio can be downloaded from the SiFive website at the following address:

https://www.sifive.com/boards/#software

Downloads are provided for Windows, MacOS, and Linux.

Windows Installation

Important

Rule #1
It is important that you choose an installation path that does not contain spaces. Freedom Studio will check the installation path when started and will warn you if it detects a path that contains any space characters.

Rule #2
You must enable Windows Long Path support. You must do this before extracting the product archive. The Freedom Studio installation folder contains paths that are deep enough to exceed the "legacy" MAX_PATH (=260) character limit imposed by Windows. This limit is still enabled by default, but Windows 10 (starting with version 1607) allows for disabling this limit by installing a specific register key/value using the Windows regedit tool:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem
LongPathsEnabled REG_DWORD = 0x1

To simplify this process you can download the following registry file and double-click it to install this key automatically:

https://static.dev.sifive.com/dev-tools/FreedomStudio/misc/EnableLongPaths.reg

If you still have problems extracting the archive after enabling Long Path Support contact support@sifive.com

More information on this topic can be found here:

https://docs.microsoft.com/en-us/windows/desktop/fileio/naming-a-file#paths

Now that we have those important notes are out of the way…

You can install multiple versions of Freedom Studio on your system, and use all of them.

We recommend that you keep the installation path as short as possible. We suggest creating a folder at the root of your installation drive called "FreedomStudio" (no spaces).

Then inside that folder you can install multiple versions of Freedom Studio into subfolders. Like:

```
c:\FreedomStudio
  |
  +- FreedomStudio-2019.03
  +- FreedomStudio-2019.05
```

The product zip archive extracts to a long folder name (for instance `FreedomStudio-4.7.2.2019-03-4-win32.win32.x86_64`). We recommend that you shorten the folder name using a naming scheme similar (or identical) to the one shown above.

We recommend using a tool like 7-Zip to handle large zip archives on Windows. Unzip the downloaded zip archive to a directory on your PC by right-clicking on the zip file and selecting "Extract All". After unzipping the bundle, you can open Freedom Studio by double-clicking on FreedomStudio.exe in the installation directory.

For more information for setting up SiFive development platforms, please consult the platform's User Guide and Windows Board Setup.

MacOS Installation

Important

It is important that you choose an installation path that does not contain spaces. Freedom Studio will check the installation path when started and will warn you if it detects a path that contains any space characters.

Extract FreedomStudio.tar.gz to the desired folder by double clicking the bundle. Freedom Studio is not currently a signed macOS application and therefore may present an error when running. Therefore in order to run Freedom Studio on macOS it may be necessary to open Freedom Studio for the first time as described in this URL: https://support.apple.com/kb/PH25088?locale=en_US

It is also possible to execute this command line to remove the extended attribute marking the .app file for quarantine:

```
$ xattr -d com.apple.quarantine FreedomStudio.app
```

Start Freedom Studio by clicking on FreedomStudio.app found in the FreedomStudio folder which was just extracted.

For setting up SiFive development platforms, please consult the platform's User Guide and macOS Board Setup.

Linux Installation

Important

It is important that you choose an installation path that does not contain spaces. Freedom Studio will check the installation path when started and will warn you if it detects a path that contains any space characters.

Extract FreedomStudio.tar.gz to the desired folder using the following command:

```
tar -xzf /path/to/FreedomStudio.tar.gz
```

For setting up SiFive development platforms, please consult the platform's User Guide and Linux OS Board Setup.

Contents

The directory contents are as follows:

FreedomStudio
The installation root directory

FreedomStudio(.exe)(.app)
The OS specific executable to open

SiFive
SiFive files

SiFive/doc
The documentation delivered with Freedom Studio.

SiFive/Licenses
Open Source Licenses.

SiFive/Misc
Directory containing miscellaneous files such as OpenOCD config files, and Linux OpenOCD udev rules

SiFive/openocd
Directory containing the bundled OpenOCD

SiFive/toolchain
Directory containing the RISC-V GCC toolchain

Build Tools (Windows Only)
Tools which allow eclipse CDT to function in a Windows environment such as make, echo, etc…

jre (Windows and Linux Only)
The Java Run Time Environment (JRE). On macOS the JRE is located under the FreedomStudio.app bundle.

Tools Setup

Freedom Studio will automatically detect its installation path on the first run and configure itself to use the bundled tools described in Section <Contents>>. If, for any reason, Freedom Studio was not able to detect the bundled tools, it will prompt the user to enter the tool paths directly with the dialog shown in Figure ???. If prompted, be sure to select the "bin" directory which contains the tool binaries. These paths will set the global defaults used by Freedom Studio.

The tool paths can be changed at anytime by clicking the following:

Windows and Linux - **Window – Preferences – Freedom Studio**

MacOS - **Freedom Studio – Preferences – Freedom Studio**

Global RISC-V Toolchain Paths - for SiFive toolchains, select the default toolchain **RISC-V GCC/Newlib** and use the browse button to select the toolchain directory

Global OpenOCD Paths - for SiFive OpenOCD distributions, set the OpenOCD executable to "openocd" and use the browse button to select the OpenOCD directory

The tool path preferences can be set at 3 different scopes: Global, Workspace, and Project. Global scope sets the default for the installation and is the lowest priority. Workspace scope allows you to set the toolchain preferences specific a a given Workspace, and will override the Global setting. Project scope, which can be set by right clicking a project in your workspace and selecting **Properties – Freedom Studio**, allows you to set preferences on a per-project basis. Project scope always takes priority over Global and Workspace.

This flexibility allows the user to easily work with a number of different tools installed on the same system, such as one built from source using Freedom-E-SDK, while still maintaining project portability.

The Freedom Studio Environment

Workspaces

Eclipse uses workspaces to group together a set of related projects. Eclipse workspaces allow for a lot of flexibility in how one organizes their projects. For example, it is possible to have a workspace which contains only a single project. It is also possible to have a workspace which contains multiple related projects such as a library project and an application which depends on that library.

Switching workspaces is accomplished by selecting **File – Switch Workspace**.

When starting Freedom Studio, Eclipse will prompt you to select a workspace. Freedom Studio will remember the locations of previously selected workspaces.

Important

When choosing a workspace location do not chose a location that contains spaces in the path.

Eclipse Perspectives

Eclipse uses perspectives to group windows together which are collectively useful for a given task.

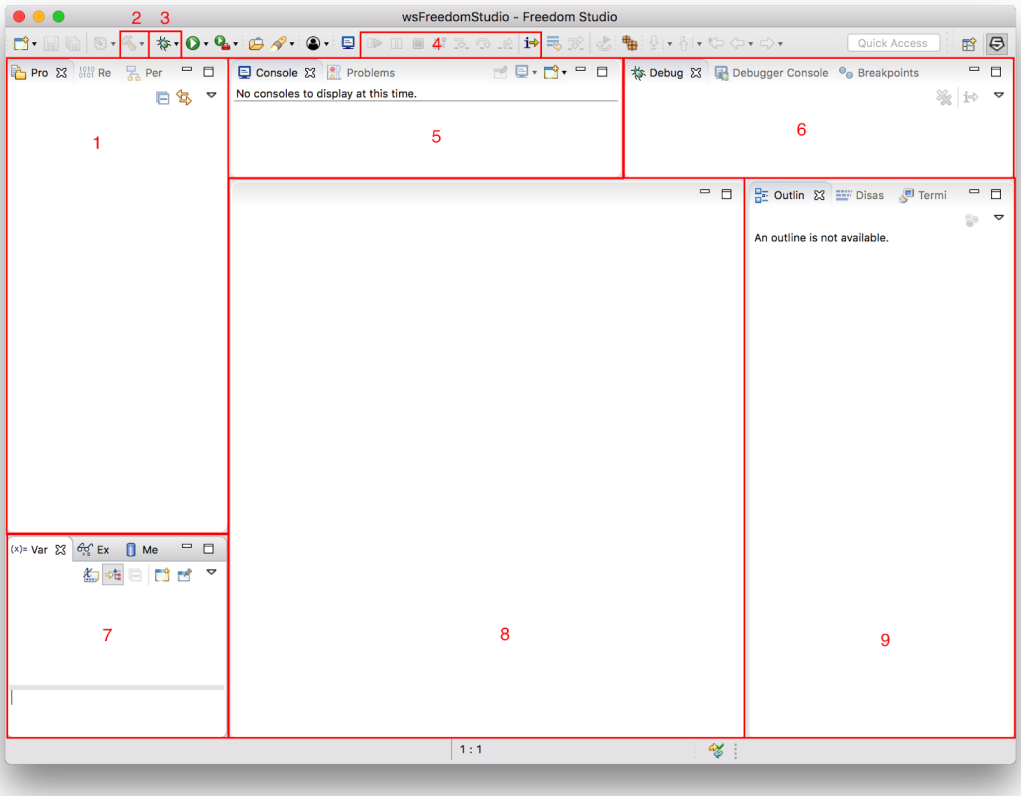
Freedom Studio ships with its own SiFive perspective which can be useful for both programming and debugging. Please see Section ??? for a detailed description of the SiFive Eclipse perspective.

Freedom Studio also ships with the standard Eclipse perspectives: C/C++ , Debug, and Git. From Eclipse, you can change perspectives by clicking **Window – Perspectives – Open Perspective**.

Perspectives are user customizable and persistent to a workspace.

The SiFive Perspective

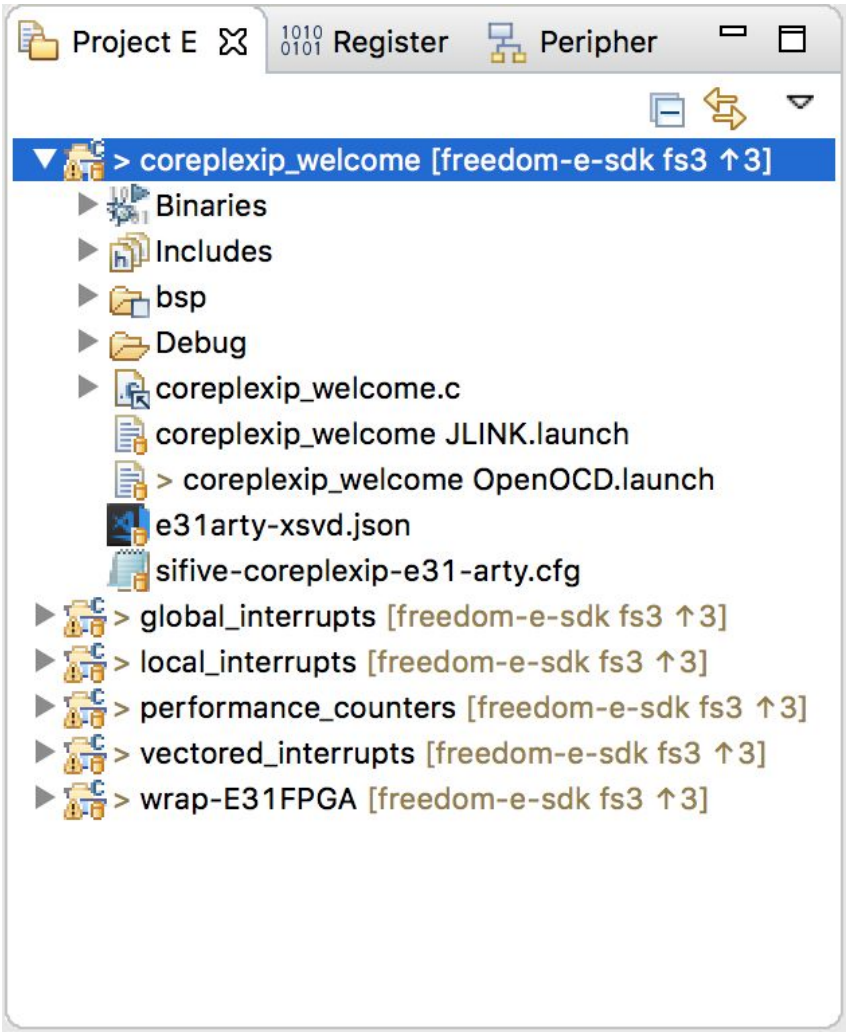
The SiFive Perspective.



1. Project Explorer, Register, and Peripheral Views. These views are described below.
2. Build Toolbar Button. Pressing this button will build (compile) the active project.
3. Debug Toolbar Button. The down arrow next to the bug lets you pick a specific configuration.
4. Debug Control Toolbar Buttons. These buttons are used for debug run, halt, and stepping control.

5. Console. These views display useful information when building applications.
6. Breakpoint and Debug Views displays useful information when debugging applications.
7. Variable, Expression, and Memory Views. These views are described below.
8. Editor View is used to edit source code.
9. Outline, Disassembly, and Terminal Views are described below.

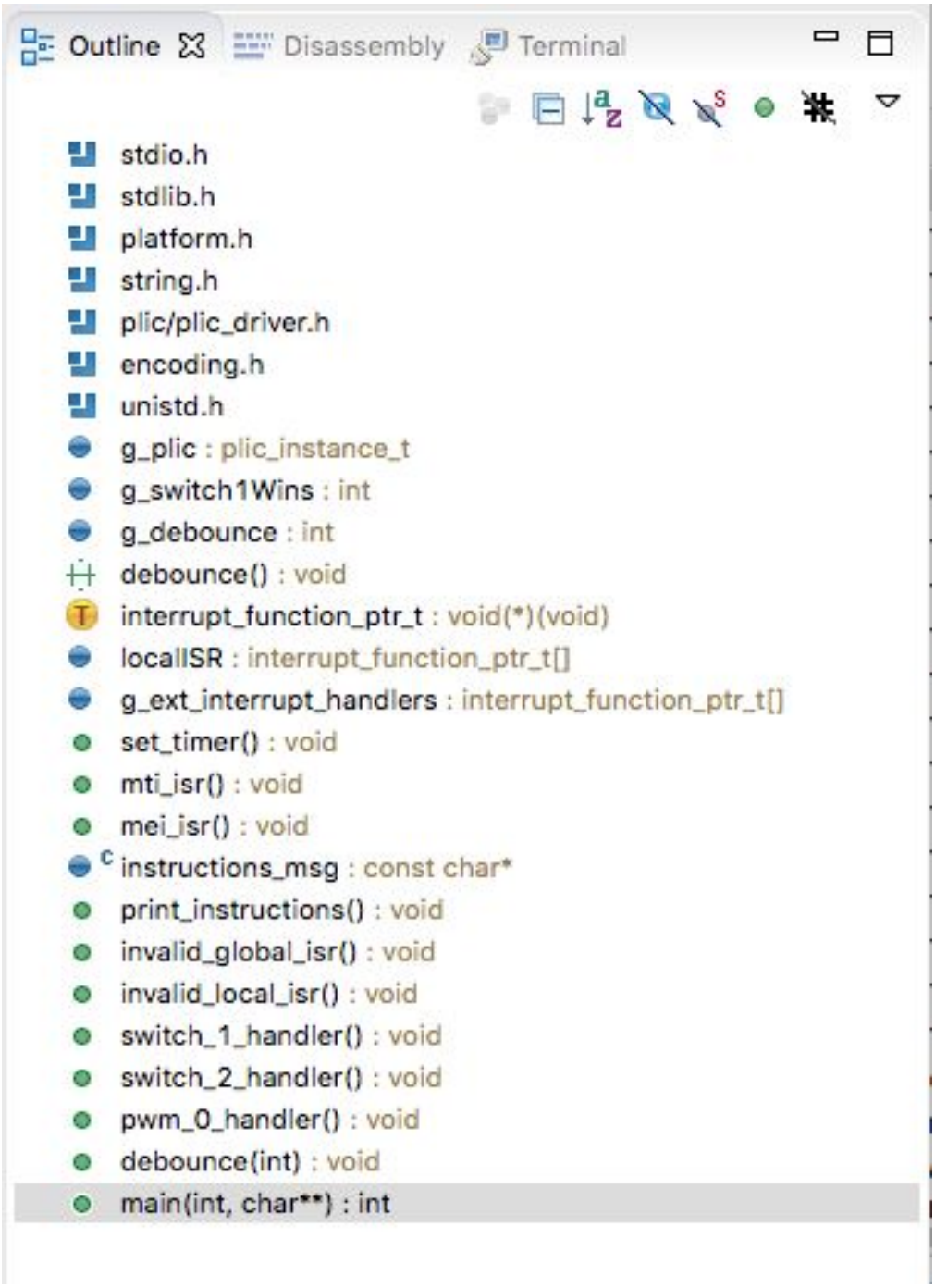
Project Explorer



The Project Explorer view displays projects in the workspace. Use this view for opening, editing, and creating new project source files. If a project contains files under revision control, Project Explorer will also display information regarding the repositories and branches.

Editor, Outline, Disassembly

The Editor and Outline views are used to write and navigate code. The Editor also provides useful contextual information for your code. Hovering the mouse over statements will reveal pop-ups which expand macros, evaluate variables and structures, provide function definitions, etc… Double-clicking a line number in the editor will set a breakpoint at that line.

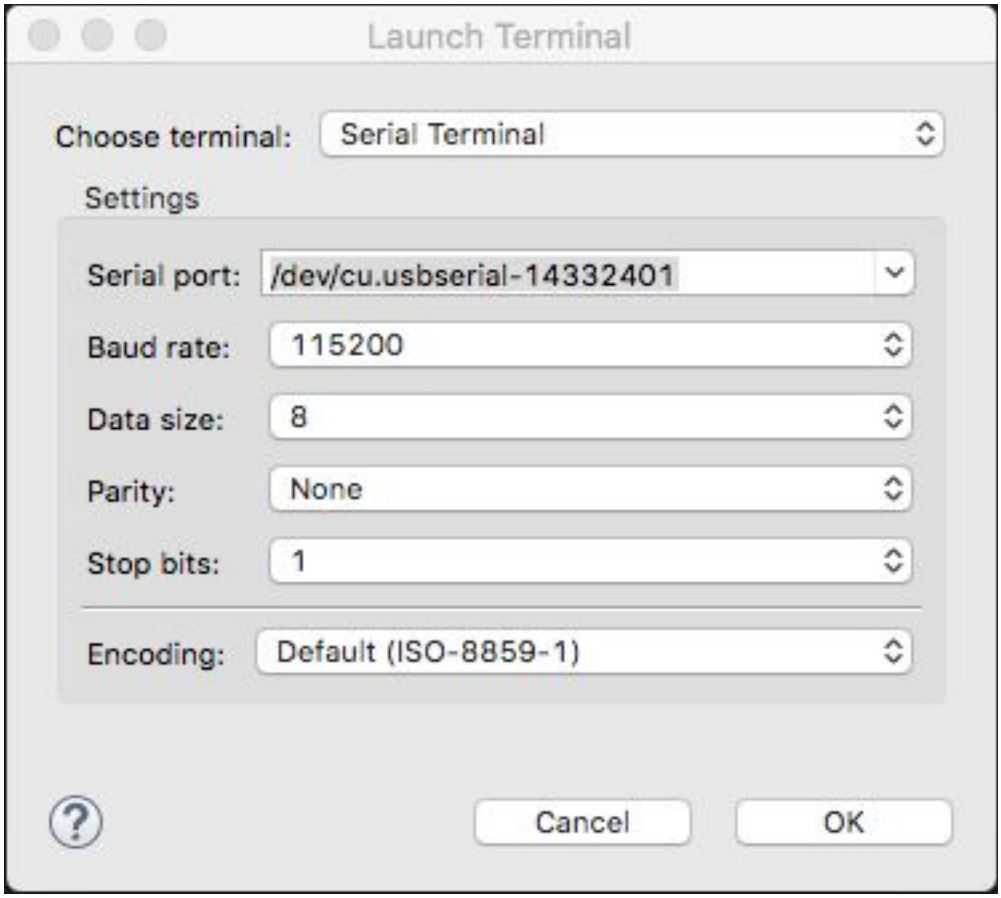


The Outline view, shown in Figure ???, gives a "top-level" view of the active file in the editor including functions, types, constants, etc… Clicking on an item in the Outline view will take you to that items location in the source code.

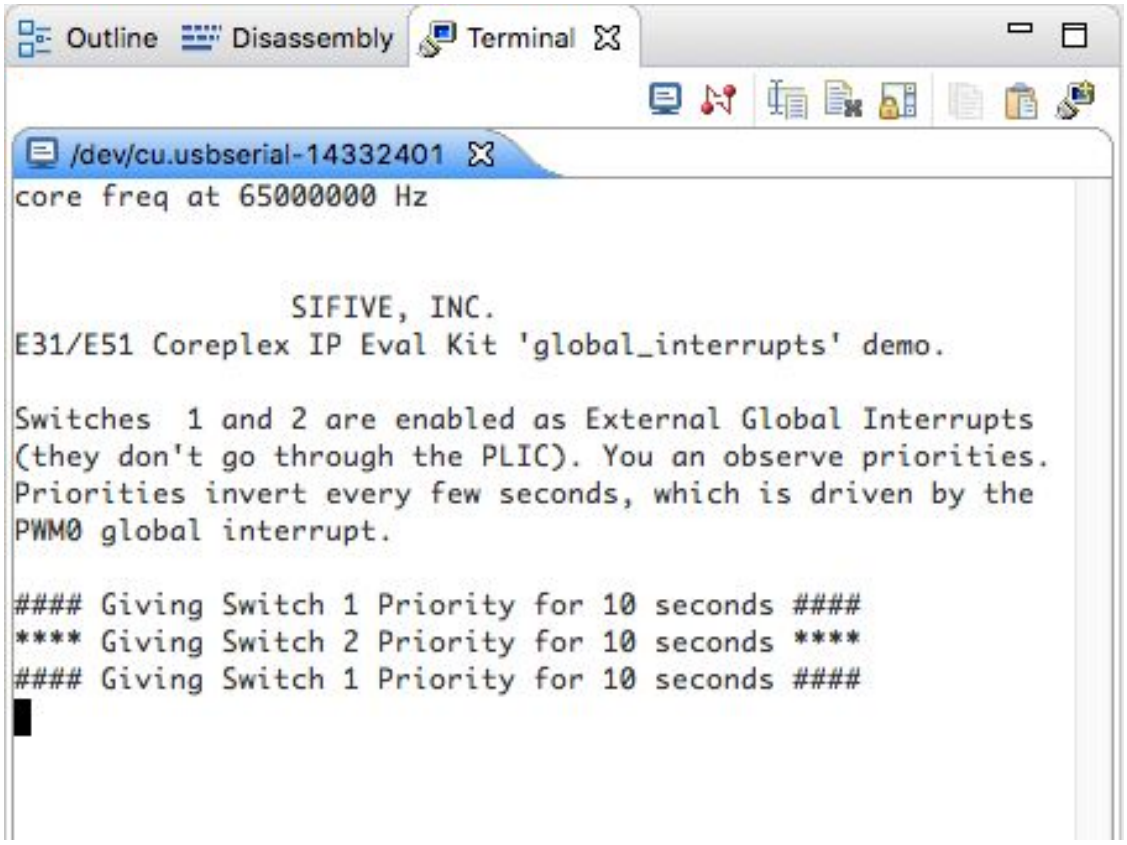
Terminal

The Terminal view, shown in Figure ???, can be used to display a local terminal, a serial terminal, or ssh into a remote machine. The serial terminal allows the user to view serial output, such as that from a SiFive development board, without leaving the development environment. On Windows platforms this view negates the need for an external serial terminal program. On MacOS and Linux platforms, it is possible to open serial port directly, or open a local terminal and run GNU Screen.

To open a serial terminal, open the Terminal view and select the "Launch Terminal" button which resembles a screen. In the **Choose Terminal** menu, select **Serial Terminal**. It is then possible to select the desired serial settings such as baud rate and encodings.



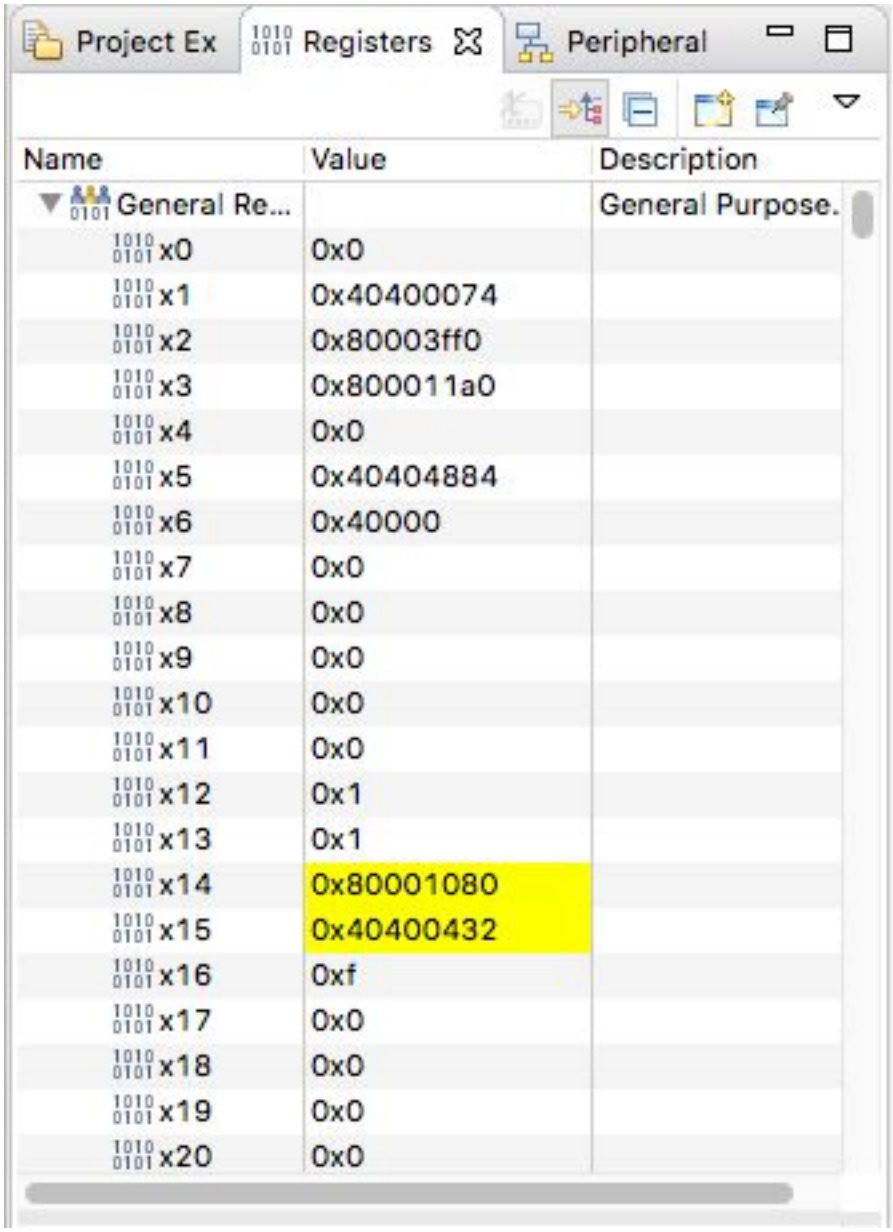
If the text in the serial terminal is displayed incorrectly, make sure that the correct baud rate is selected. SiFive example projects default to 115200 baud. Changing the encoding to UTF-8 might also help.



Breakpoints

The Breakpoints view allows for creating, enabling, and disabling of breakpoints. You can set a breakpoint's properties by right-clicking on a breakpoint and selecting "Properties". From the properties menu, you can set properties such as breakpoint type (hard, soft), and ignore count.

Registers

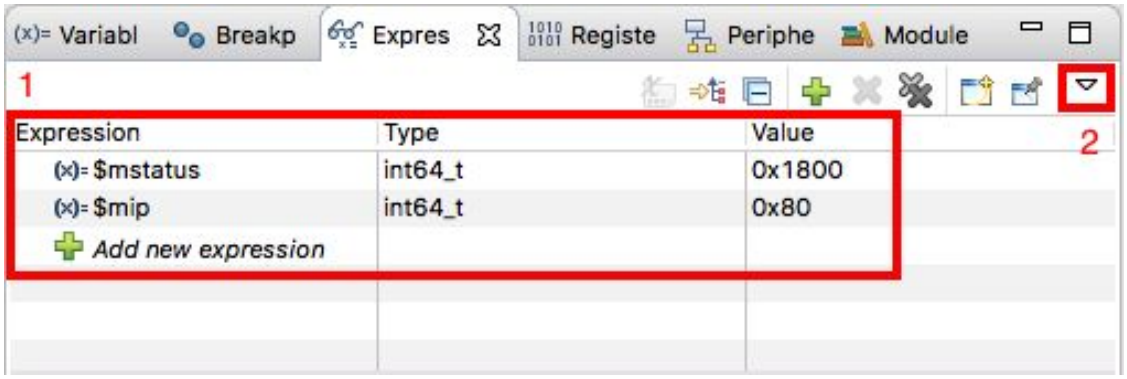


The Registers view displays the integer and floating point register files. It is possible to write to registers by double-clicking their value field. While stepping through code, the Registers view will highlight registers as they change.

Expressions

The Expression view allows you to view any variable within scope. In addition to variables, it is possible to use this view to see the current value of CSRs on your device. The Expression view, along with other eclipse views which display variables and memory,

allows for changing the value format (for example to hexadecimal). The format can be changed by clicking the down arrow marked with "2" in screenshot:



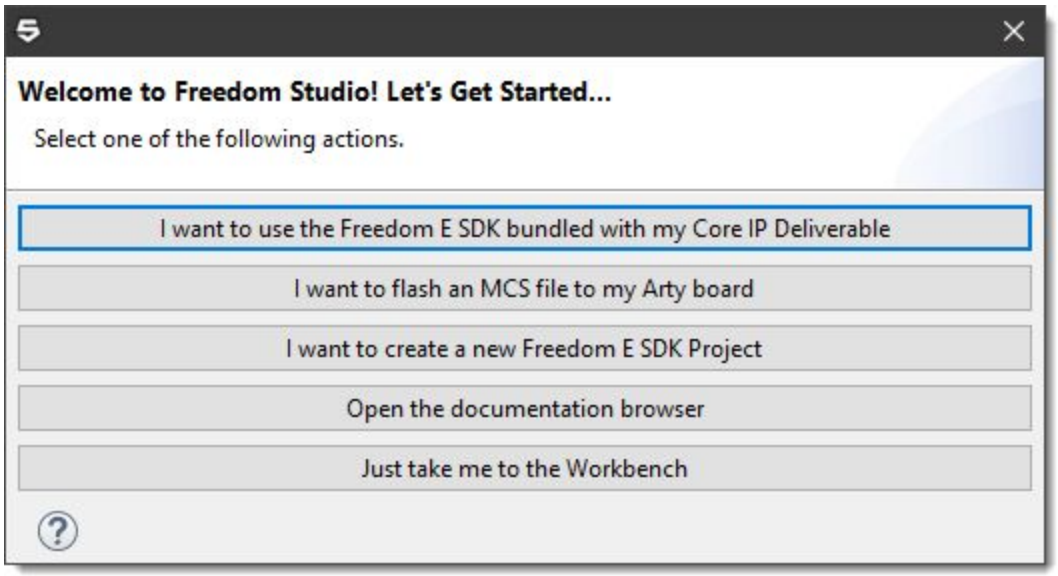
Getting Started

This section walks step-by-step through creating and debugging a freedom-e-sdk project.

Starting with version 19.05, Freedom Studio includes a brand new dedicated Freedom E SDK project wizard.

The First Run Dialog

When you start Freedom Studio with a new workspace you will be presented with the First Run Dialog. This dialog is simply an easy way to get started with common first time tasks.



- **I want to use the Freedom e SDK bundled with my Core IP Deliverable**
  Choose this option if you have a core IP deliverable that you'd like to start working with. With this option you will tell Freedom Studio where the core IP deliverable

freedom-e-sdk folder is. Freedom Studio will then use this bundled SDK when creating new projects.

- **I want to flash an MCS file to my Arty board**
  Choose this option if you want to get started by flashing an MCS file to your Arty board. You will also have the choice to jump right into creating a project at the end of the flashing process.

- **I want to create a new Freedom E SDK project**
  If you have a HiFive series board or an Arty FPGA board already flashed with core IP and want to jump straight to creating a project, select this option.

- **Open the documentation browser**
  Choose this option to open the documentation browser. From here you can dig into all the documentation bundled with Freedom Studio.

- **Just take me to the workbench**
  If you don't want to start with any of the options listed above, choose this option and you'll be taken to your new clean workspace.

Flashing Your Arty FPGA Board

Before you continue

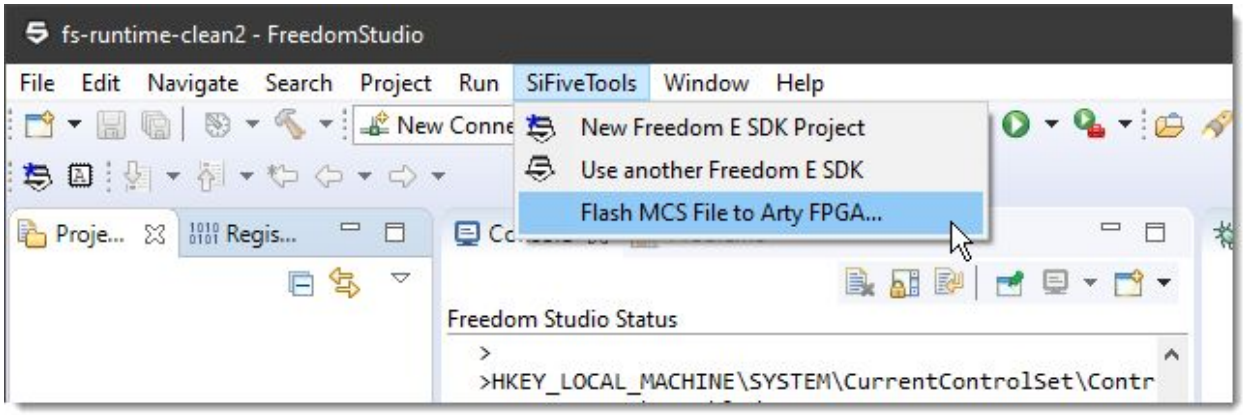
Before continuing with this section please review the Target Board Setup instructions to ensure that everything is properly configured and all host dependencies have been installed.

Flashing the Arty requires both the Olimex probe and the Arty board USB connector be connected to the host PC. Both USB connection are used during the process. Do not simply connect the Arty USB to a power supply when flashing.

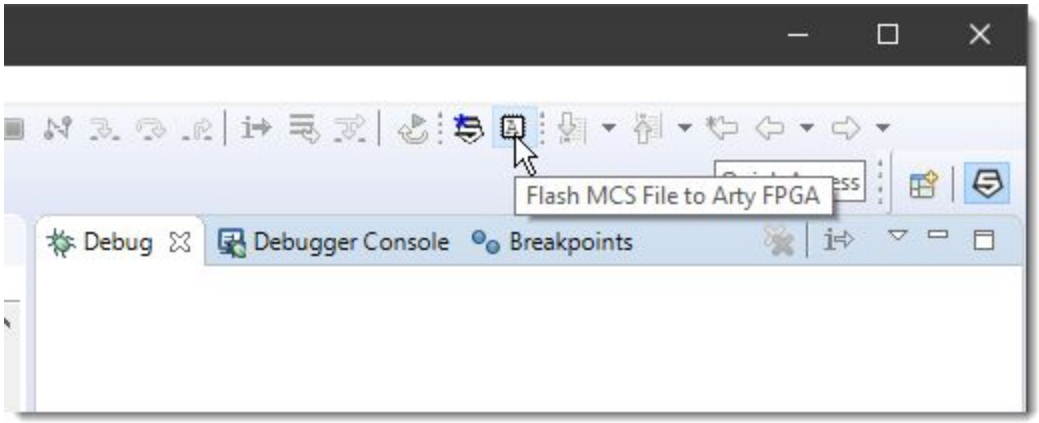
Flashing an Arty Board using a JLink connection is not supported at this time.

Flash the Arty Board

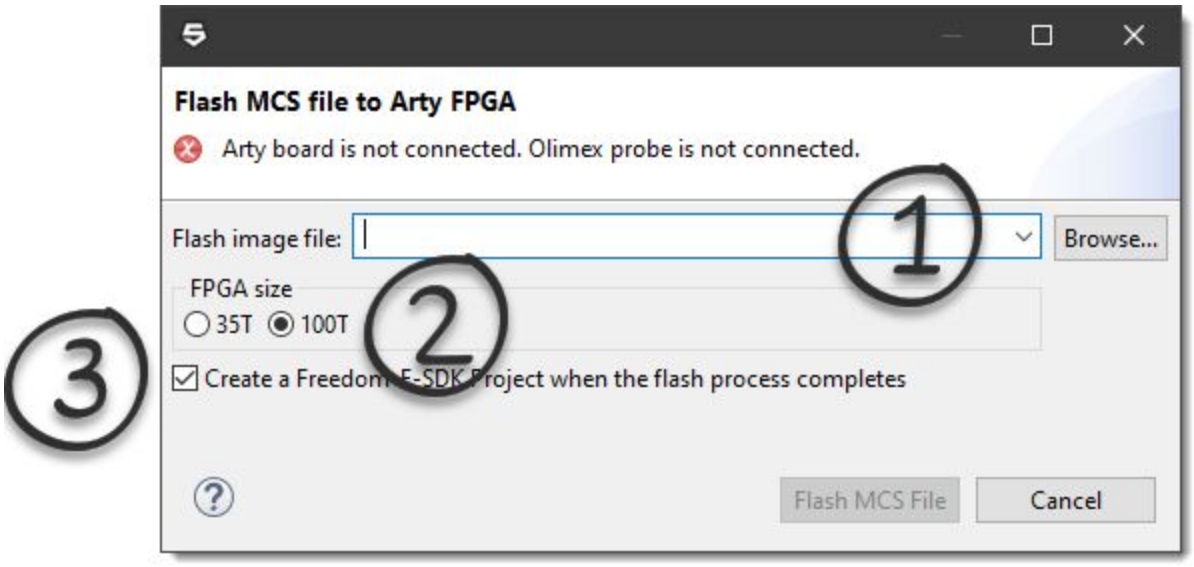
First open the Arty Flash Dialog. This dialog can be opened from the main menu by selecting SiFiveTools → Flash MCS File to Arty FPGA…



or by clicking the Arty Flash icon on the main toolbar:



Selecting either of these will open the Arty Flash Dialog. This dialog will look a little different on each host platform:



1. First select the MCS file you want to flash. MCS files are available in Core IP deliverables; in downloaded evaluation packages, or directly from Sifive, or may be created in your flow.
2. Important: Make sure you select the correct FPGA configuration for your MCS file and Arty board. Freedom Studio will attempt to select the correct setting, but if it cannot be determined heuristically,, no default selection is made and you will have to choose. Choose wisely.
3. Check this box if you want to open the New Freedom E SDK Project Wizard when the flashing process is completed.

Once you've made your selections, click the **Flash MCS File** button to start the flashing process. See the notes below regarding Windows hosts.

The flash process can take several minutes to complete. When it is complete Freedom Studio will prompt you to press the PROG button on the Arty board. You must do this in order to load and use the newly flashed MCS file.

Windows Only

On Windows host platforms Freedom Studio can monitor the connection status of the Olimex probe and the Arty Digilent connection. The Arty Flasher will report the status (as shown above) and the **Flash MCS File** button will not be enabled unless both are detected as connected.

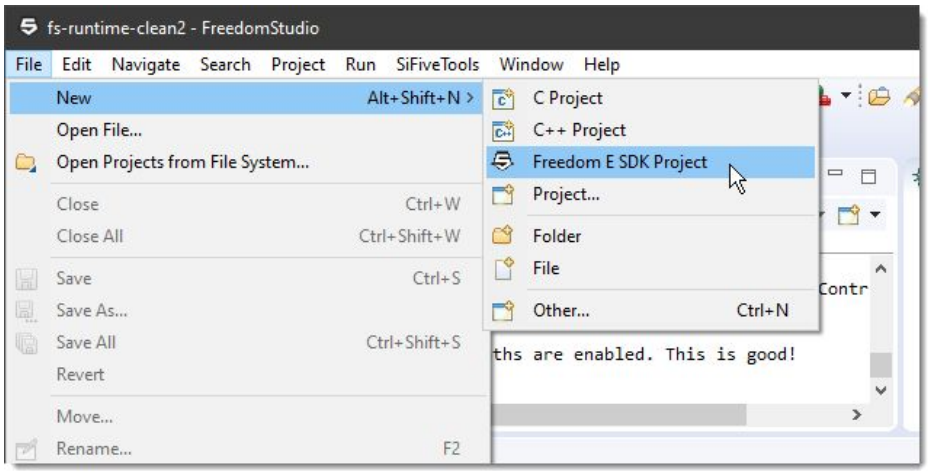
Freedom Studio also monitors the driver status for both devices and will install required drivers as parts of the flashing process. You may have to authorize the driver installation if

Windows displays a UAC prompt. Flashing will not succeed unless you authorize the driver installation.

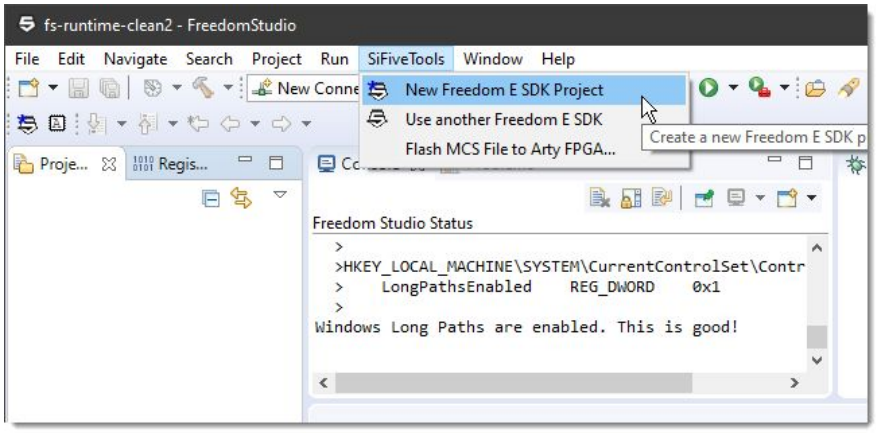
Create a Freedom E SDK Project

Creating a new Freedom E SDK Project is very simple. There are multiple ways to start:

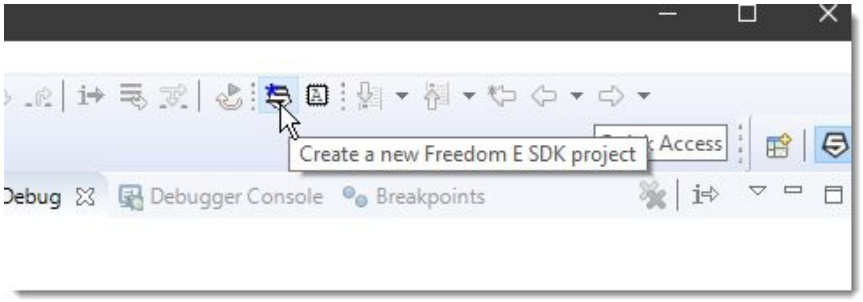
- From the main menu, select **File → New → Freedom E SDK Project**, as shown below:



- From the SiFiveTools menu, select **New Freedom E SDK Project**, as shown:



- On the main application toolbar, click the “New Freedom E SDK Project” icon, as shown:



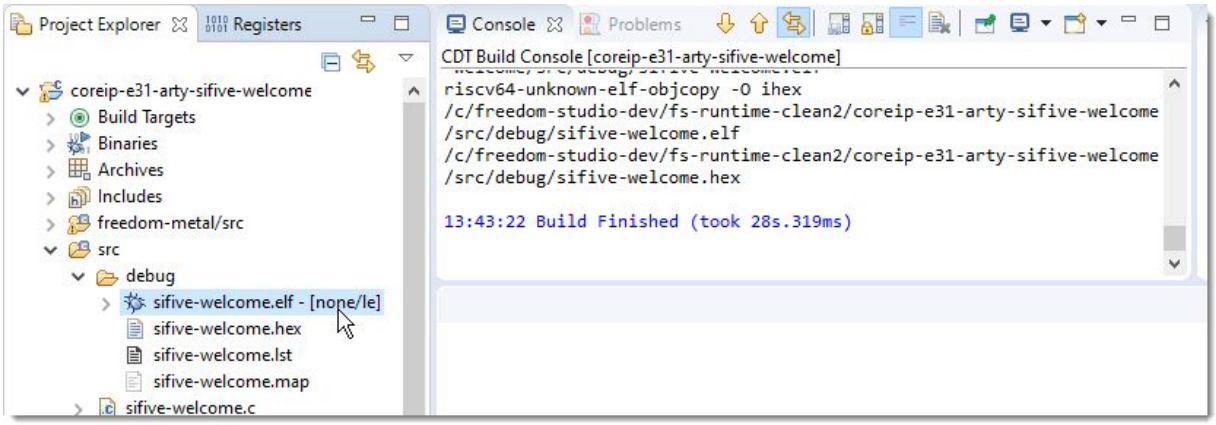
Selecting any of these will open the Freedom E SDK New Project Wizard. The first page of this wizard is shown below:

1. When you first open the wizard the target selection box might be empty. You need to select a target from the options in the drop-down. You should select the target that matches your core and target platform of choice. For instance if you have an Arty Board and are using an E31 core you will want to choose "coreip-e31-arty".
2. Select an example program. Several examples are provided and each one demonstrates different features sets of the core.
3. The project name is automatically generated based on your target and example selections. If you do not like the generated name you can change it on the next wizard page.
4. Finally, you can choose to automatically create a debug launch configuration for your new project. Select the type of launch as determined by your debugger probe.

Choose “OpenOCD” if you are using an Olimex probe, and “JLink” if you are using a JLink probe or a target with a built-in JLink OB device, and “QEMU” if you are using one of the QEMU targets. Selecting certain targets will automatically select the best option for that target.

That’s really all there is to creating a new Freedom E SDK project. If you are satisfied with your choices, go ahead and click the **Finish** button. If you would like to change the project name, click the **Next** button and give your project a new name on the following page.

When you click the **Finish** button, Freedom Studio will create your new project and build it. When the build is complete Freedom Studio will reveal the built ELF file in the project explorer, as shown:



If you chose to create a debug launch configuration when creating your project the Debug Launch Configuration Dialog will automatically open after the ELF file is revealed.

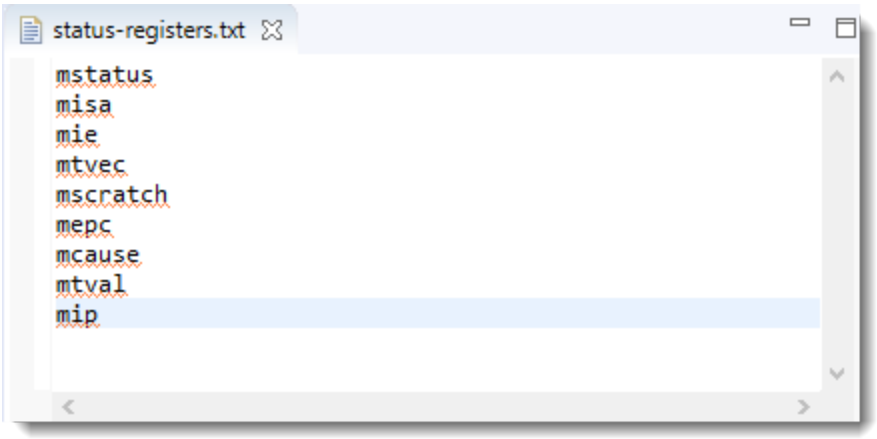
Freedom Studio HOWTO Guides

Register List Management

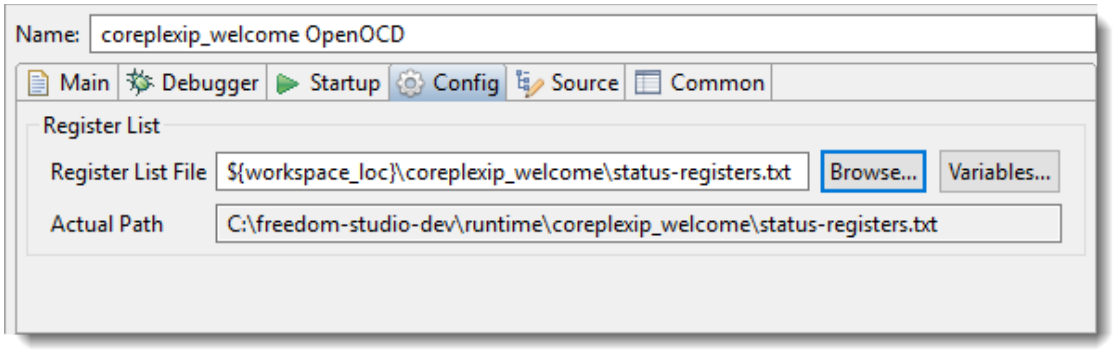
This document provides an overview on how to use and customize the list of registers displayed in the Freedom Studio IDE Registers View. This feature is primarily intended to give you control over what registers are displayed. You may want to use this, for example, when you do not want to see a complete list of all target registers. Or alternately, you may want to specify registers that are not included in the default list of registers.

A Quick Example

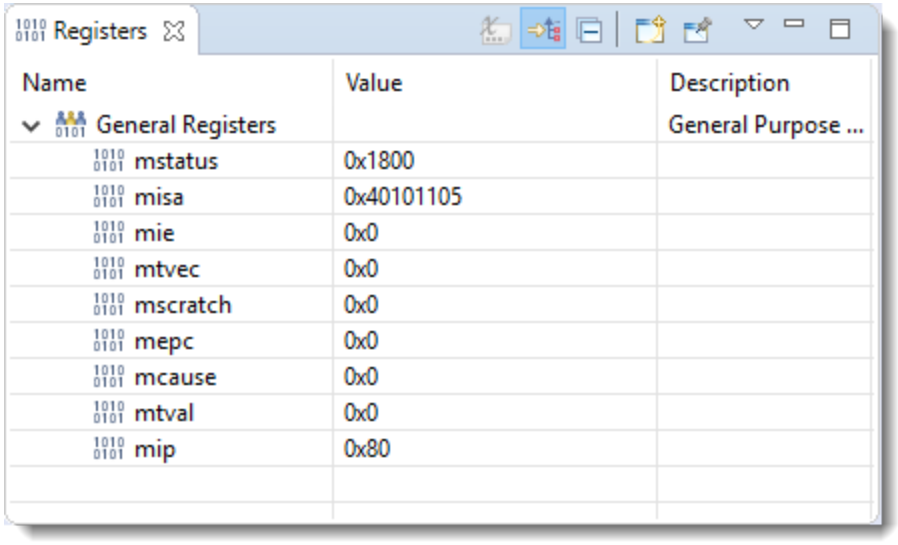
Let’s assume you have a register list file called 'status-registers.txt'. The content of the file looks like:



Now specify that file as a register list in the debug launch configuration:



When you launch your debug session and open the Register View you will see this list:



Creating Register List Files

The Register List File is a text file that, at it's simplest form, lists a single register name on each line. Each listed register will be displayed in the Registers View in the order specified.

Commenting the Register List File

The register list file treats any line that starts with a # (hash) character as a comment line. It is ignored by the parser. The # character can be preceded by whitespace.

Only the first word of a line is treated as a register name. Any additional words are ignored.

Specifying Register Names

Single Registers

Any register can be specified by putting the name of the register as the first word on a line.

Built-in Macros

The following macros can be used to specify multiple related registers without having to list each register individually

Built-in Register List Macros

Macro Name	Description
general_registers	The 32 General Purpose Registers plus PC
machine_registers	The machine status registers
perfmon_registers	Performance Monitor Control and Data Registers
fpu_registers	Floating Point Registers

Include File

You can create several register list files, for example, building your lists of related registers, and then build a master register list by including these files in a composite register list file. To include another register list simply use:

```
#include <register-list-file>
```

The `#include` directive can be used multiple times in a single file.

Nested `#include` directives are supported. An `#include` file may `#include` additional files.

Where `register-list-file` is either an absolute or relative path. Relative paths are relative to the folder containing the current register list file being parsed. Keep this in mind

if you are using nested `#include` directives and your register list files live in different folders.

Register Ordering

Registers are displayed in the Register View in the same order as they are specified in the Register List File.

Using Register List Files

Now that you have created one or more register list files you may want to use them with Freedom Studio. This section explains your options for specifying how to use your register list files.

Where To Specify a Register List File

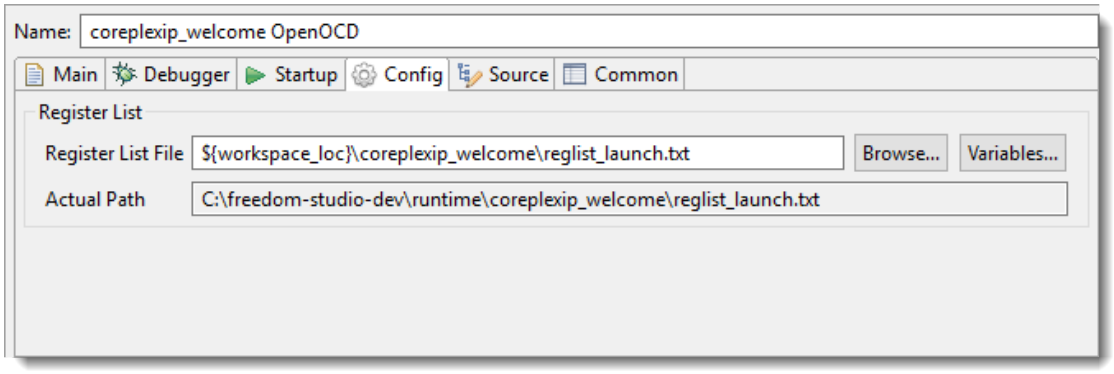
A register list file can be specified in 4 place. These four locations are prioritized such that a specification in a higher priority location will override any specification in a lower priority location. The four locations are, in descending priority order (highest priority first):

Prioritized Register List Specification Locations

Location	Description
Debug Launch Config	Specify a register list file for each individual launch configuration
Project Property	Specify a register list file for each project
Workspace Preference	Specify a register list file for each workspace
Global Preference	Specify a global register list file, for all Freedom Studio workspaces

Debug Launch Configuration

This is the highest priority option for specifying a register list file. You will find the controls to specify the debug launch register list file on the `Config` tab of the Debug Launch Configuration Dialog:

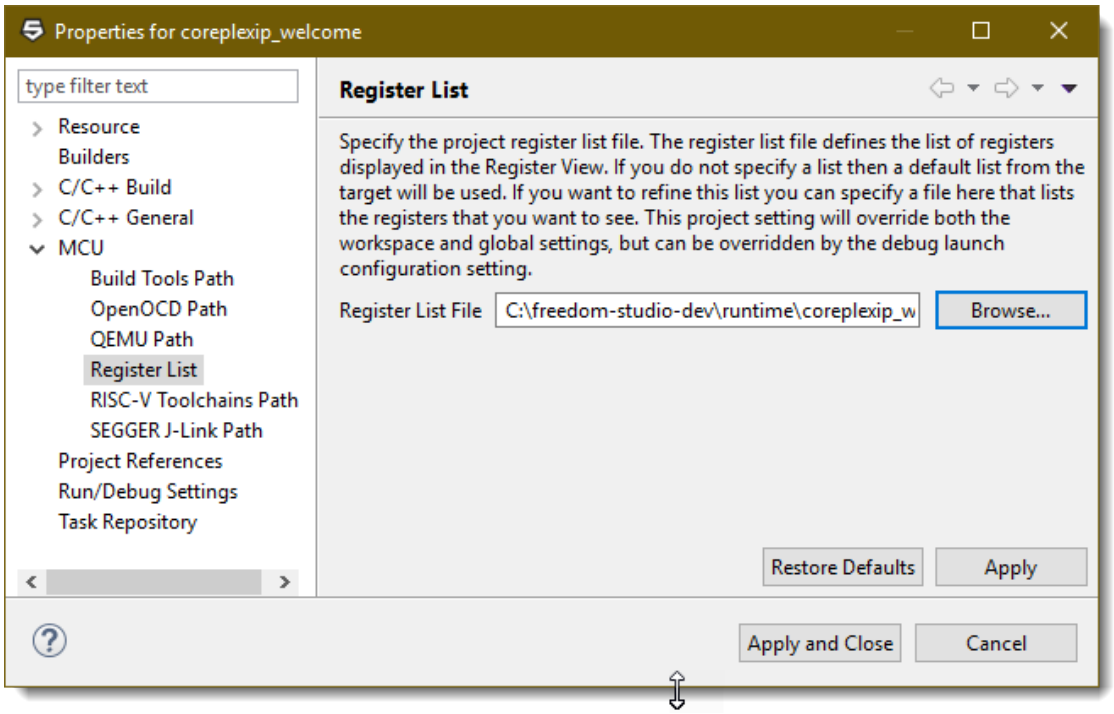


When you specify a register list file in a debug launch configuration the path displayed in the `Actual Path` box will always reflect the fully resolved path to the register list file. If you are not specifying a register list file here then the `Actual Path` may display a path to another register list file if one has been specified using a lower priority specifier.

Project Property

Specifying a register list file as a project property will cause that register list to be used with all launch configuration created for the project, overriding any global or workspace preferences. Each launch can override the project specification by using the launch configuration option to specify a register list file.

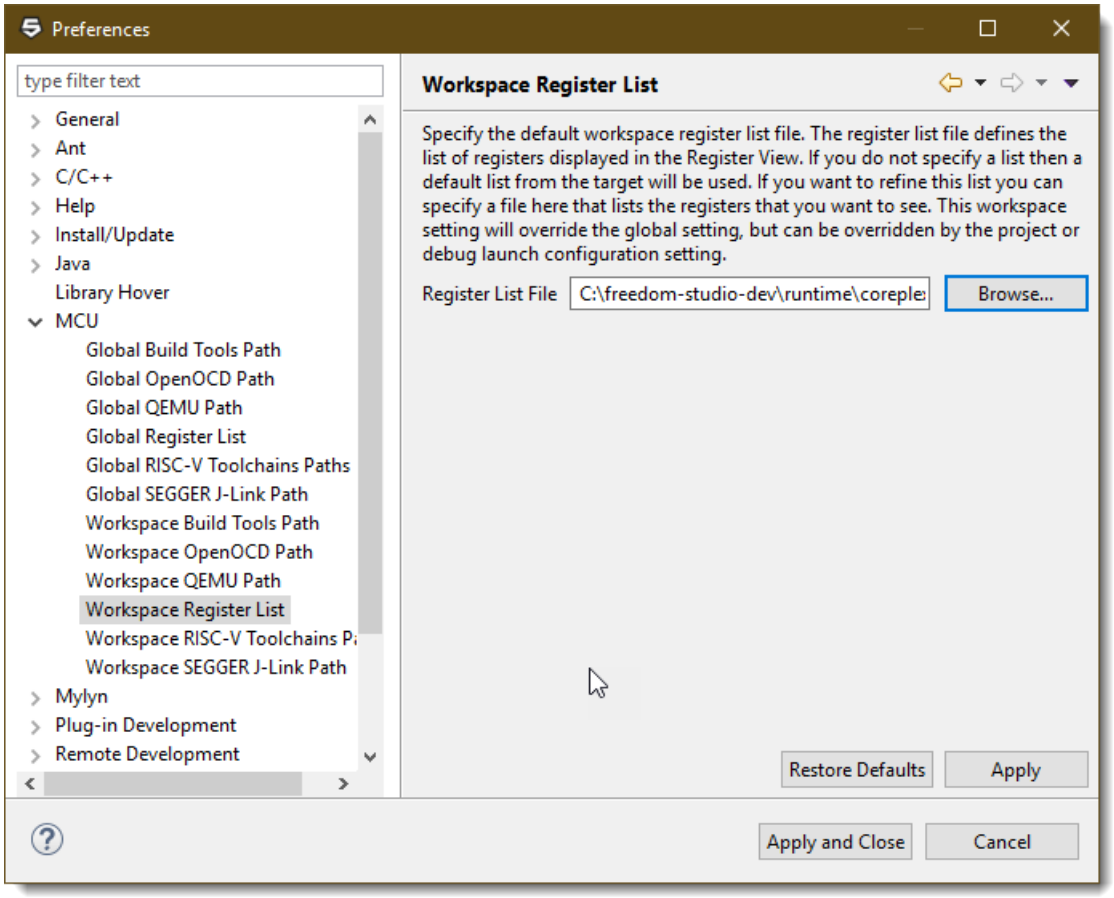
You can setup the project register list file specification by opening the Project Properties dialog and navigating to the `MCU → Register List` property page:



Workspace Preferences

Specifying a workspace register list file will cause that file to be used for all projects within the workspace unless a project overrides the setting by specifying a register list in the project properties or a debug launch configuration.

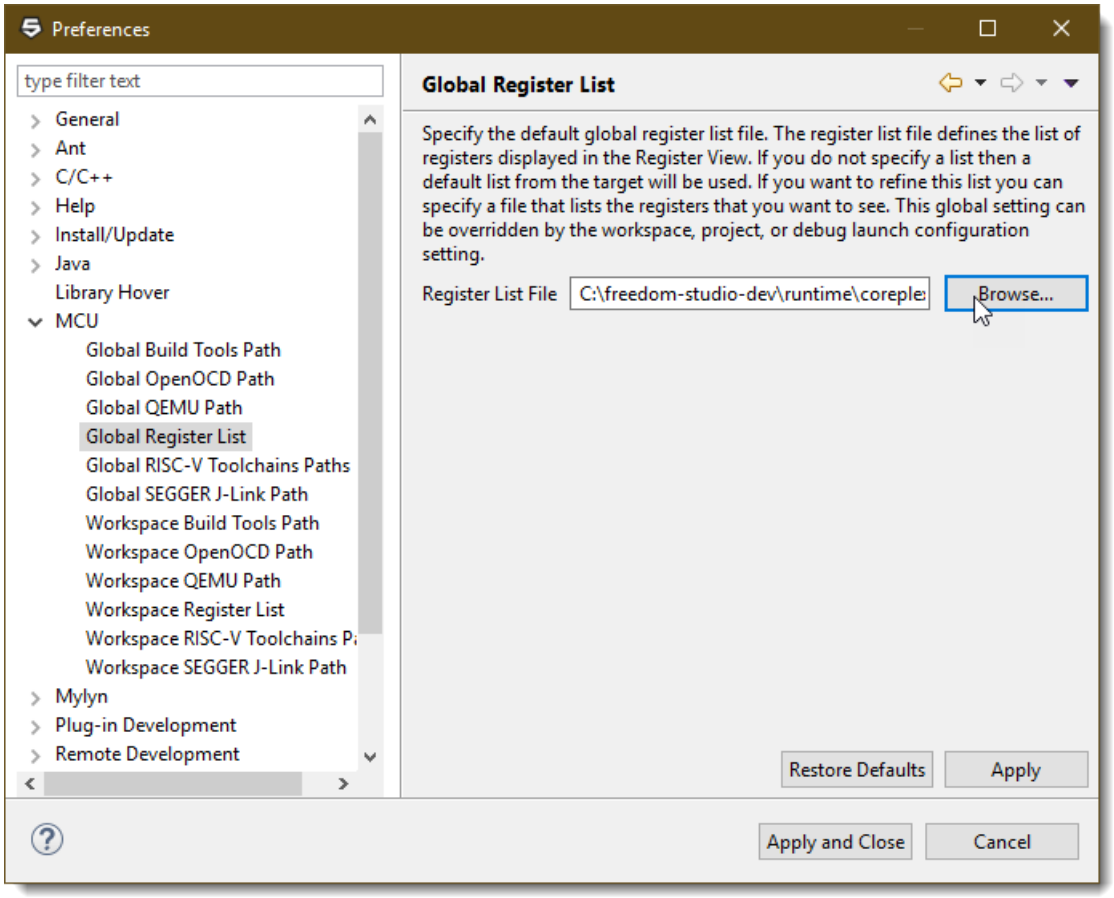
You can specify the workspace preference by opening the Freedom Studio Preference Window and navigating to the `MCU → Workspace Register List` page:



Global Preferences

Specifying a global register list file will cause that file to be used for all Freedom Studio workspaces unless a workspace, project, or debug launch overrides the setting.

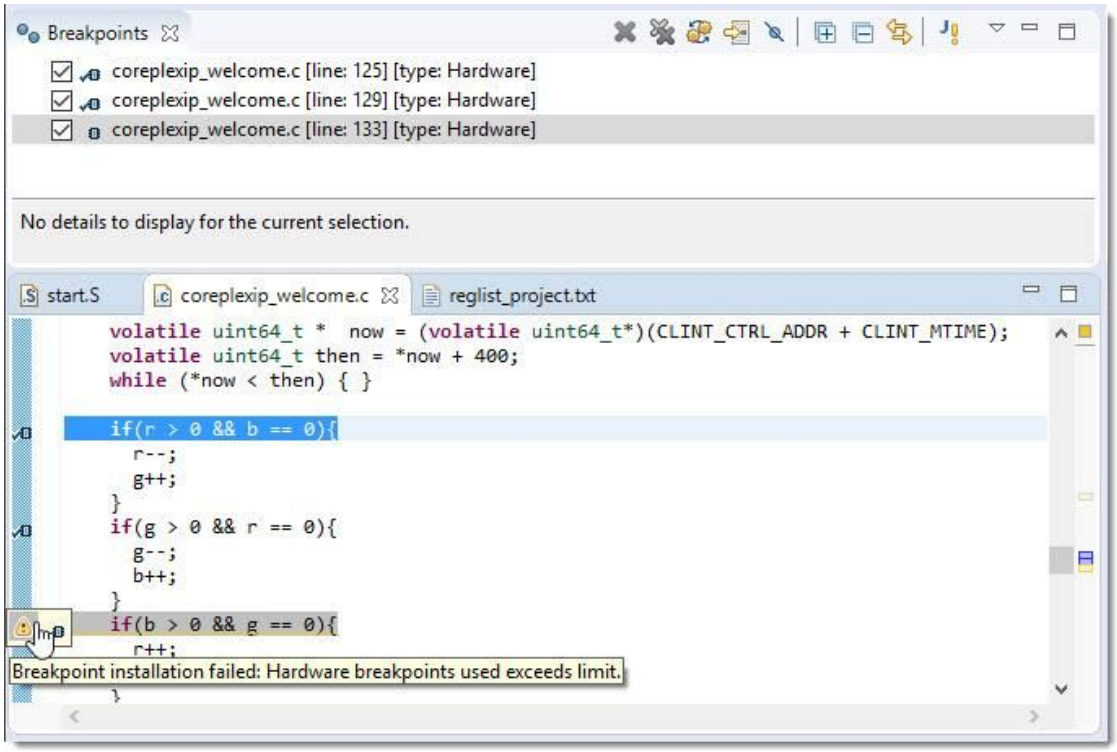
You can specify the global preference by opening the Freedom Studio Preference Window and navigating to the `MCU → Global Register List` page:



Managing Hardware Breakpoint Resources

This document summarizes how to manage hardware breakpoint resources on a target system. Different cores have different numbers of hardware breakpoints. It is important for GDB to know how many hardware breakpoints exist on a target. Attempting to use more breakpoints than exist on the target will cause unpredicable debugger problems.

When GDB knows how many hardware breakpoints exist on the target, you can create as many hardware breakpoints as you need, but only the number that exist will be enabled. Freedom Studio will indicate which breakpoints cannot be enabled due to lack of resources. You can then manage the enablement of each breakpoint to ensure that the breakpoint you need is enabled (by disabling breakpoints that you do not need). This screenshot shows how Freedom Studio indicates that too many hardware breakpoints have been enabled.



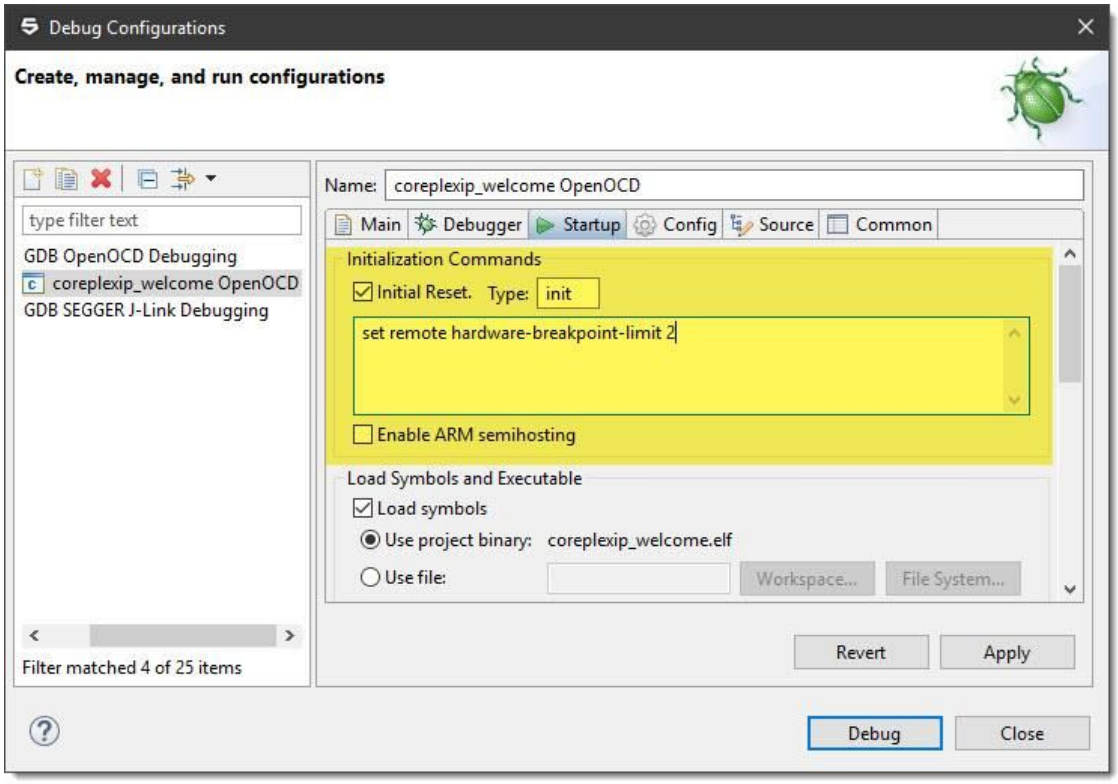
Too Many Hardware Breakpoints

Freedom Studio cannot automatically determine the number of hardware breakpoints present on the system. We plan to add this ability in a future release.

GDB needs to know the number of hardware breakpoints on the target. There are two ways to do this.

Option 1: Add a gdb initialization command

Add the 'set remote hardware-breakpoint-limit' command to the Initialization Commands section of a launch configuration. You must do this for every new launch configuration.



Note

Setting this setting using Option 1 takes precedence over Option 2 (described below). If you find that your preference setting is not being applied, check to make sure that you do not have this command specified in the Initialization Commands.

Option 2: Set a preference or project property

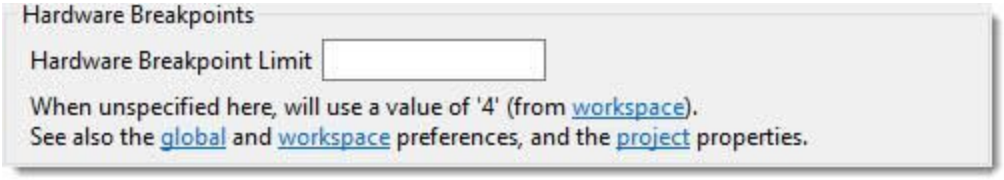
You can set global and workspace preferences to define the number of hardware breakpoints on your target system. You can also set this in your project properties and in a launch configuration.

Finer-grain settings take priority over courser-grain settings. The priority, from highest to lowest is:

- Debug Launch Configuration
- Project Property
- Workspace Preference
- Global Preference

Each new launch configuration will use the highest priority setting that exists. If no setting exists, then Freedom Studio will use the hard-coded default of '2'.

The launch configuration dialog always describes the setting used and where the setting originates. For instance, the screenshot below shows the setting is 4 and originates from the workspace preference setting. This implies that the project property setting has not been defined (it is blank). Clicking on any of the underlined setting scopes will open the corresponding settings page where you can change the setting if desired.



Setting value description

Valid settings

The following table shows the valid setting values.

Valid Setting Values

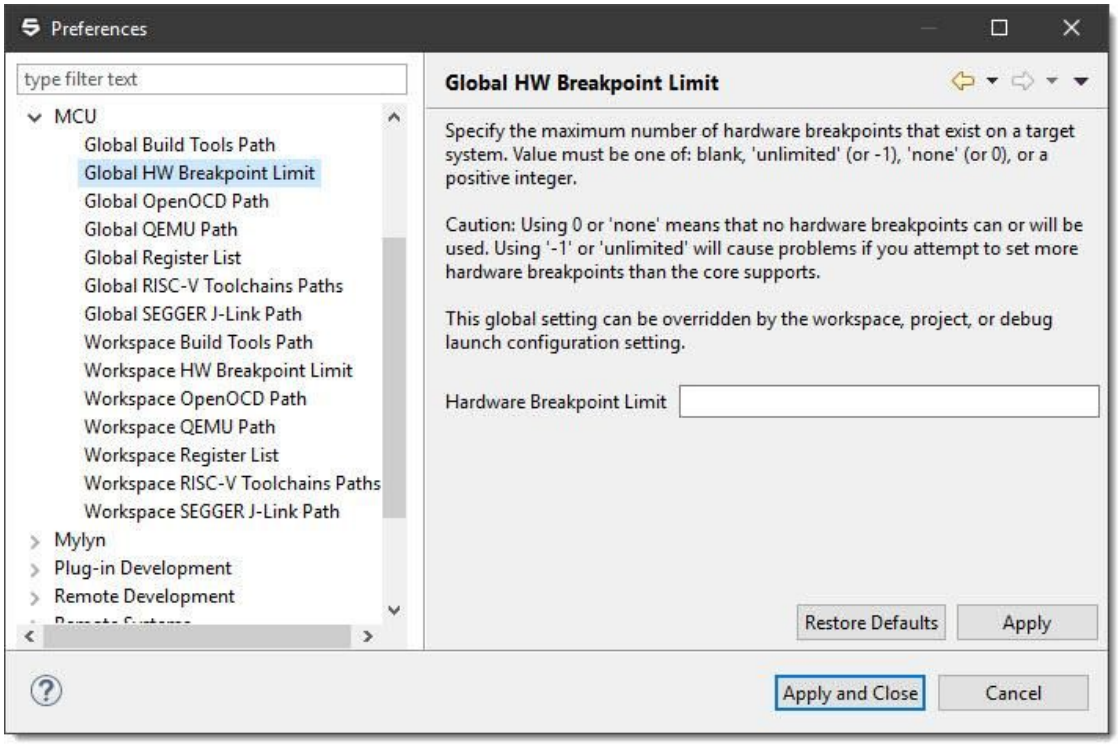
Value	Description
blank	Leave the setting blank and it will not be used.
'unlimited', or -1	Tells GDB that you have unlimited hardware breakpoints.
'none', or 0	Tells GDB that you have no hardware breakpoints.
x, a positive integer	Tells GDB that you have *x* hardware breakpoints.

GDB defaults to 'unlimited'. Freedom Studio overrides this default and uses '2'. Using 'unlimited' allows you to set more hardware breakpoints than may exist on the target. GDB will attempt to set all of them. This leads to unpredictable debugger behavior. We do not recommend using 'unlimited', but we won't stop you from doing so.

Setting the Global Preference

We recommend setting the hardware-breakpoint-limit globally when you have a single target system. This ensures that the setting applies in all workspaces, projects, and launch configurations. If you ever need to use a different target that has a different number of hardware breakpoints you can easily override the global setting using any of the higher priority settings.

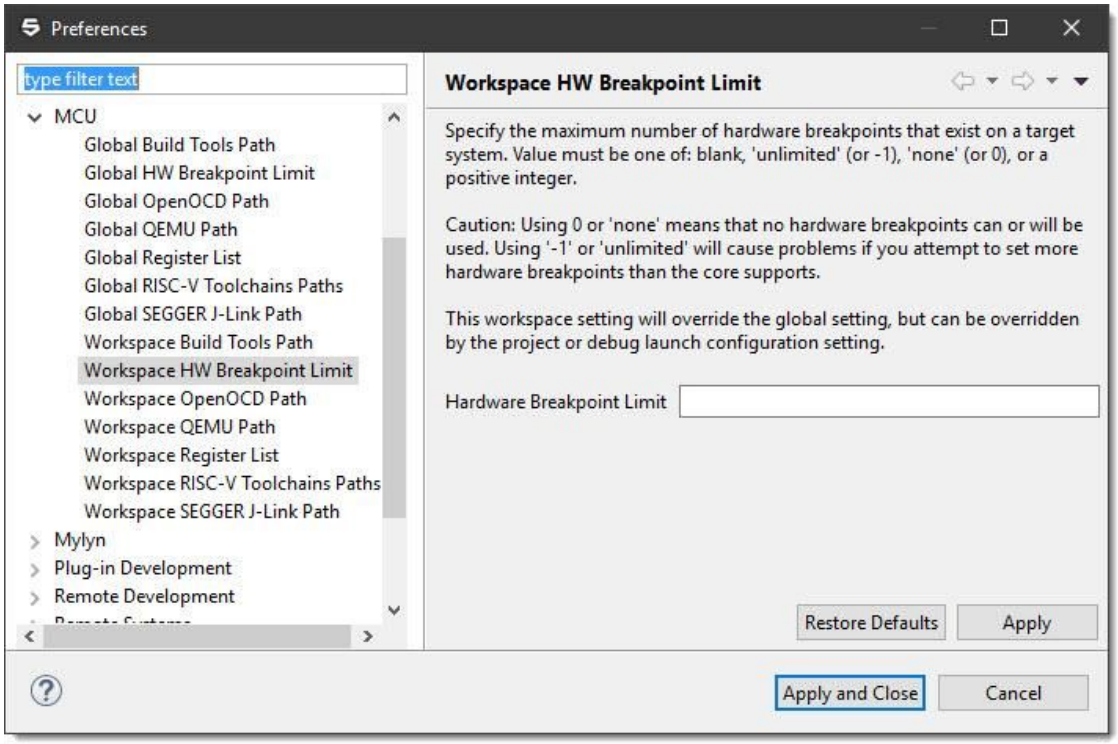
Set the global preference by opening the Preferences Dialog (`Windows -→ Preferences`) and navigating to the `MCU | Global HW Breakpoint Limit` page.



Setting the Workspace Preference

We recommend using the Workspace Preference when you have multiple target systems and want to create a workspace for each target system. This ensures the setting is correctly applied for the target used in each workspace.

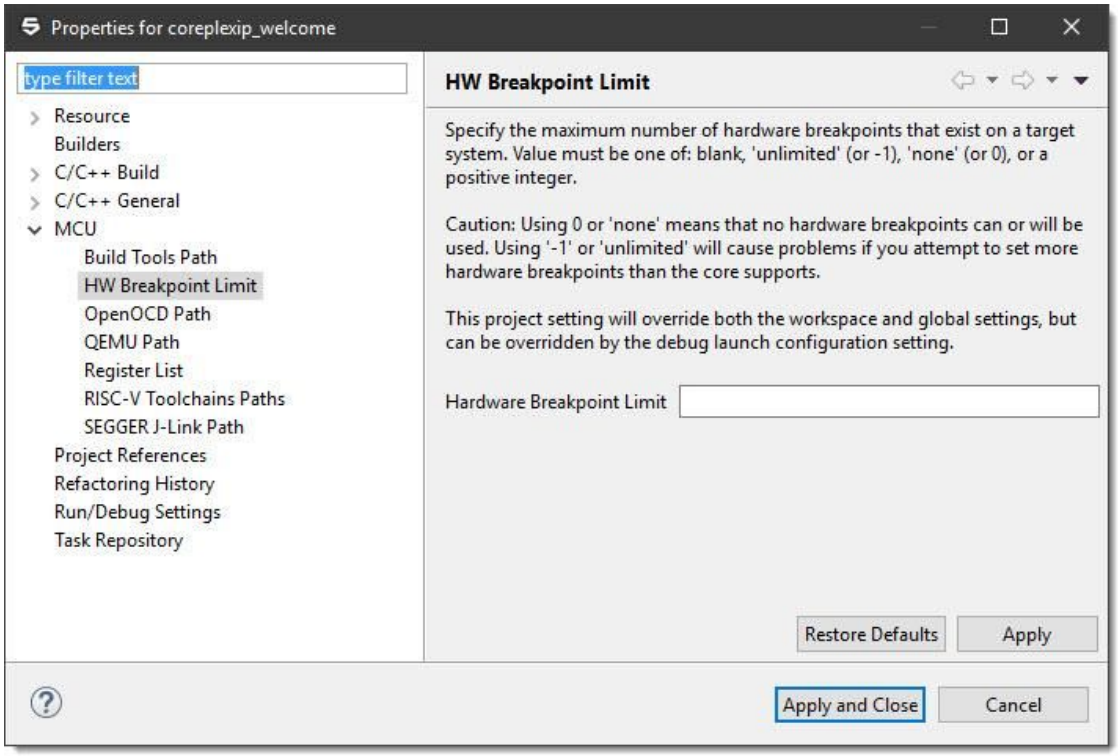
Set the workspace preference by opening the Preferences Dialog (`Windows -→ Preferences`) and navigating to the `MCU | Workspace HW Breakpoint Limit` page.



Setting the Project Property

We recommend using the Project Property setting when you have multiple target system and want to work on all of them within a single Workspace. This ensures the setting is correctly applied for the target used in each project.

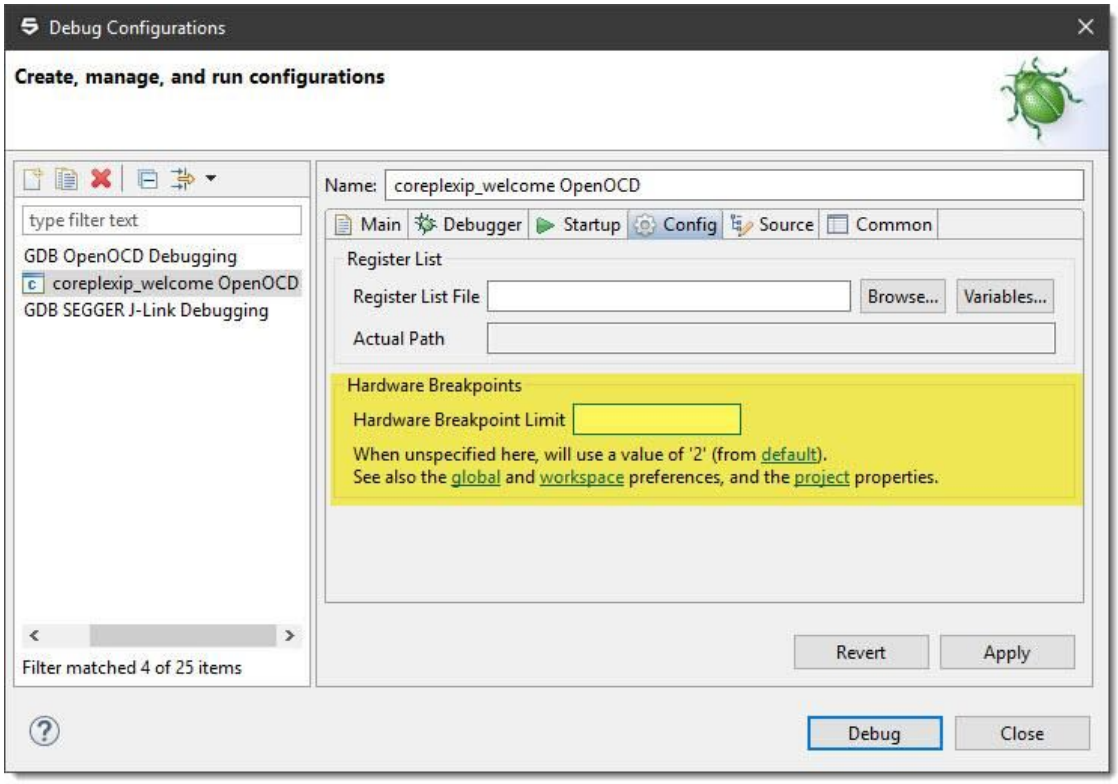
Set the project property by opening the Project Properties Dialog (`Project -→ Properties`) and navigating to the `MCU | HW Breakpoint Limit` page.



Setting the Launch Configuration Attribute

We recommend using the launch configuration attribute setting for target connections that you do not use often. This ensures the setting is not applied to oft-used targets (that are better served using a more broadly applied setting from the project, workspace, or global settings).

Set the launch configuration attribute by opening the launch configuration dialog, navigating to the 'Config' tab, where the breakpoint count can be set for this single launch config.



Conditional Optimization

This section describes how to apply compiler optimization conditionally within a source file.

Debugging optimized code can be complicated because the optimizer will change the order of the code and optimize out variables. When single-stepping through the code the source line indication can jump around erratically. You will not be able to examine variable values that have been optimized away.

The normal solution is to turn off optimizations for the entire project when you need to debug something. Sometimes this is not desirable (or even possible). In these cases you can turn off optimization for just the code that needs to be debugged using compile-time `#pragma` statements.

The comments in the following source example explain how, when, and when not do use the `#pragma` statements.

Example source code.

```
// See LICENSE for license details.

#include <stdint.h>
#include <stdbool.h>
#include <stdatomic.h>
```

```
#include "encoding.h"
#include <platform.h>

#ifndef _SIFIVE_COREPLEXIP_ARTY_H
#error 'coreplexip_welcome' demo only supported for Coreplex IP Eval Kits
#endif

void pwm(uint16_t r, uint16_t g, uint16_t b);
uint16_t option0(uint16_t p1, uint16_t p2);
uint16_t option1(uint16_t p1, uint16_t p2);
uint16_t option2(uint16_t p1, uint16_t p2);

static const char sifive_msg[] = "\n\r\
\n\r\
                SIFIVE, INC.\n\r\
\n\r\
         55555555555555555555555\n\r\
        5555                   5555\n\r\
       5555                     5555\n\r\
      5555                       5555\n\r\
     5555       555555555555555555555\n\r\
    5555       5555555555555555555555\n\r\
   5555                             5555\n\r\
  5555                               5555\n\r\
 5555                                 5555\n\r\
555555555555555555555555555          55555\n\r\
 55555           555555555           55555\n\r\
   55555           55555           55555\n\r\
     55555           5           55555\n\r\
       55555                   55555\n\r\
         55555               55555\n\r\
           55555           55555\n\r\
             55555       55555\n\r\
               55555   55555\n\r\
                 555555555\n\r\
                   55555\n\r\
                     5\n\r\
\n\r\
";

#if __riscv_xlen == 32
  static const char welcome_msg[] = "\n\r\
\n\r\
Welcome to the E31 Coreplex IP FPGA Evaluation Kit!\n\r\
\n\r";
#else
static const char welcome_msg[] = "\n\r\
```

```
\n\r\
Welcome to the E51 Coreplex IP FPGA Evaluation Kit!\n\r\
\n\r";
#endif

static void _putc(char c) {
  while ((int32_t) UART0_REG(UART_REG_TXFIFO) < 0);
  UART0_REG(UART_REG_TXFIFO) = c;
}

int _getc(char * c){
  int32_t val = (int32_t) UART0_REG(UART_REG_RXFIFO);
  if (val > 0) {
    *c =  val & 0xFF;
    return 1;
  }
  return 0;
}

static void _puts(const char * s) {
  while (*s != '\0'){
    _putc(*s++);
  }
}

/*
 * Enable maximum optimization for the main() function.  The pragma directive
for
 * optimization act on a function scope. You cannot place these pragmas
around code
 * inside of a function (it will generate a compiler error).
 */
#pragma GCC push_options
#pragma GCC optimize ("3")
int main (void){

  // 115200 Baud Rate at (65 / 2) MHz
  UART0_REG(UART_REG_DIV) = 282;
  UART0_REG(UART_REG_TXCTRL) = UART_TXEN;
  UART0_REG(UART_REG_RXCTRL) = UART_RXEN;

  // Wait a bit because we were changing the GPIOs
  volatile int i=0;
  while(i < 10000){i++;}

  _puts(sifive_msg);
```

```
  _puts(welcome_msg);

/*
 * These pragma, if uncommented, will generate compiler errors because this
 * only works when used outside of functions.  Optimization is performed
 * on function blocks, not on individual code lines.
 */
//#pragma GCC push_options
//#pragma GCC optimize ("0")
  uint16_t r=0x3F;
  uint16_t g=0;
  uint16_t b=0;
//#pragma GCC push_options

  PWM0_REG(PWM_CFG)   = 0;
  PWM0_REG(PWM_CFG)   = (PWM_CFG_ENALWAYS) | (PWM_CFG_ZEROCMP) |
(PWM_CFG_DEGLITCH);
  PWM0_REG(PWM_COUNT) = 0;

  // The LEDs are intentionally left somewhat dim.
  PWM0_REG(PWM_CMP0)  = 0xFE;

  while(1){
    volatile uint64_t *  now = (volatile uint64_t*)(CLINT_CTRL_ADDR +
CLINT_MTIME);
    volatile uint64_t then = *now + 400;
    while (*now < then) { }

    if(r > 0 && b == 0){
      r--;
      g++;
    }
    if(g > 0 && r == 0){
      g--;
      b++;
    }
    if(b > 0 && g == 0){
      r++;
      b--;
    }

    pwm(r,g,b);

    g = option0(r, b);
    b = option1(r, g);
    r = option2(g, b);
```

```
  }// While (1)
}
#pragma GCC pop_options

/*
 * This function uses the project setting for optimization
 */
void pwm(uint16_t r, uint16_t g, uint16_t b)
{
    PWM0_REG(PWM_CMP1)  = 0xFF - (r >> 2);
    PWM0_REG(PWM_CMP2)  = 0xFF - (g >> 2);
    PWM0_REG(PWM_CMP3)  = 0xFF - (b >> 2);
}

/*
 * Enable maximum optimization.  The 'result' variable will be optimized out.
 */
#pragma GCC push_options
#pragma GCC optimize ("3")
uint16_t option0(uint16_t p1, uint16_t p2) {
    int result = p1 * p2;
    return result;
}
#pragma GCC pop_options

/*
 * Turn off all optimization. The 'result' variable is not optimized out.
 */
#pragma GCC push_options
#pragma GCC optimize ("0")
uint16_t option1(uint16_t p1, uint16_t p2) {
    int result = p1 * p2;
    return result;
}
#pragma GCC pop_options

/*
 * Enable maximum optimization.  The 'result' variable would normally be
optimized out.
 */
#pragma GCC push_options
#pragma GCC optimize ("3")
uint16_t option2(uint16_t p1, uint16_t p2) {
    /*
     * Use 'volatile' keyword to ensure variable does not get optimized out.
     */
    volatile int result = p1 * p2;
```

```
    return result;
}
#pragma GCC pop_options
```

Known Issues

Freedom Studio is still considered Beta software and as such there are a number of known, but minor, issues which are described below.

If you come across other issues not reported here, please let us know on our forum: https://forums.sifive.com/.

When the debugger first connects I receive message saying "No source available for address"

This occurs when instructing the debugger to halt immediately after connecting to the target. It is safe to ignore this message. Stepping/Running the target will work as expected from this point.

Upon starting a debug connection, the Console prints out a lot of text in red colored font

While red font can be scary, it is generally benign debugging output. This happens because OpenOCD output status message through stderr and Freedom Studio renders stderr in red. You can change this color in the Freedom Studio Preference, but be aware that this will affect all consoles that accept and display stderr text.

Trouble Shooting

Linux USB Permission Issues

By default, some Linux distributions do not give users permissions to access USB devices. The HiFive1 and FPGA getting started guides describe the process to grant your user the correct permissions. For your convenience the *99-openocd.rules* file is included with in the *FreedomStudio/SiFive/Misc* directory.

Correcting Terminal Output

When using the Terminal View in Freedom Studio you may see terminal output from a target UART that does not properly handle “carriage returns”. You may see output that looks like :

To resolve this, open a command window and issue the following command:

```
stty -F <tty-device-name> onlcr inlcr
```

You can do this while connected to the terminal in Freedom Studio. You should see immediate results. You may need to adjust other stty settings depending on your environment.

Target Board Setup

Windows Board Setup

This section will describe how to connect SiFive development boards to your Windows computer.

Digilent (on Arty boards) and Olimex devices require specific device drivers to function properly with Freedom Studio. Starting with Freedom Studio 2019.05 these device drivers are automatically installed when needed. There is no need to manually install any device drivers. When a driver is installed you may be prompted by Windows UAC to authorize the installation of the driver.

The device driver for the Digilent USB connection is only installed when you use the Arty Flashing utility within Freedom Studio. If you intended to use Vivado to update your FPGA images then we suggest you use Vivado only to update your FPGA. Vivado and Freedom Studio use two different device drivers for updating the FPGA image.

If you have used Freedom Studio to update or install and FPGA image and then decide to use Vivado, you will need to manually uninstall the device driver installed by Freedom Studio (via the Device Manager) before Vivado will recognize the target again.

Windows JLink USB Driver

Note

If you have installed JLink software independently of Freedom Studio then the USB driver is already installed.

If you are using a HiFive1-revB board (which has a JLink interface built-in), or if you intend to use a JLink Probe you need to ensure that the JLink USB device driver is installed.

Freedom Studio, at this time, does not install this driver automatically. The driver installation file is located at:

```
<install-folder>/SiFive/jlink/jlink<version-info>/USBDriver/x64/dpinst_x64.exe
```

Run the installer and accept the default choices.

macOS Board Setup

By default, macOS has the standard FTDI driver installed while OpenOCD expects to communicate over USB using libusb. In order to allow OpenOCD to communicate with the SiFive development boards, it is necessary to unload the FTDI driver from macOS.

The procedure to unload the driver is available through the *SiFiveTools -> Setup OpenOCD FTDI Access* menu entry or by typing it manually at the command prompt:

- Open *Applications/Utilities/Terminal*
- Paste in the following command:
  `sudo kextunload -p -b com.apple.driver.AppleUSBFTDI`
- Paste in the following command:
  `sudo kextutil -b com.apple.driver.AppleUSBFTDI -p AppleUSBEFTDI-6010-1`

Note: This is not a permanent solution and after logging out of you computer it is necessary to issue the above commands above.

To avoid having to issue these commands on every log-in, it is possible to add the above commands to your user's *.17ex/.bash_profile*. By doing so, the above commands will be issued automatically every time your user logs in.

To switch back to standard Apple FTDI Access the *SiFiveTools -> Restore Apple FTDI Access* menu entry can be used or again it can be typed manually at the command prompt:

- Open *Applications/Utilities/Terminal*
- Paste in the following command:
  `sudo kextunload -p -b com.apple.driver.AppleUSBFTDI`
- Paste in the following command:
  `sudo kextutil -b com.apple.driver.AppleUSBFTDI`

Linux OS Board Setup

Required Libraries

For Arty board and Olimex support The following libraries need to be installed on the host system:

- libftdi1
- libusb

These can be installed on Ubuntu with the following command:

```
>sudo apt-get install libftdi1-2 libusb-0.1-4 libusb-1.0
```

And on CentOS 7 with the following command:

```
>sudo yum install libftdi libusb
```

And on Fedora 29 with the following command:

```
>sudo yum install libftdi-1.3-12.fc29.x86_64 libusb-1:0.1.5-13.fc29.x86_64
```

Let's Check Our Dependencies

The two programs that require these libraries are `OpenOCD` and `xc3sprog`. You can check that all dependencies are satifsied using the `ldd` utility.

For instance, on Ubuntu:

```
$ cd ~/FreedomStudio/SiFive/xc3sprog/xc3sprog-0.1.2-2019.04.1
$ ldd xc3sprog
    linux-vdso.so.1 =>  (0x00007ffed35f8000)
    libftdi1.so.2 => not found
    libusb-0.1.so.4 => not found
    libstdc++.so.6 => /usr/lib/x86_64-linux-gnu/libstdc++.so.6
(0x00007f395565f000)
    libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007f3955447000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f395507d000)
    libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f3954d74000)
    /lib64/ld-linux-x86-64.so.2 (0x00007f3955a42000)
$ cd
~/FreedomStudio/SiFive/riscv-openocd/riscv-openocd-0.10.0-2019.05.0-RC1/bin
$ ldd openocd
    linux-vdso.so.1 =>  (0x00007ffe3cadd000)
    libusb-1.0.so.0 => /lib/x86_64-linux-gnu/libusb-1.0.so.0
(0x00007fe58b0b1000)
    libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007fe58ada8000)
    librt.so.1 => /lib/x86_64-linux-gnu/librt.so.1 (0x00007fe58aba0000)
    libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fe58a99c000)
    libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0
```

```
(0x00007fe58a77f000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fe58a3b5000)
    libudev.so.1 => /lib/x86_64-linux-gnu/libudev.so.1 (0x00007fe58b4b2000)
    /lib64/ld-linux-x86-64.so.2 (0x00007fe58b2c9000)
```

While OpenOCD looks good, we can see the we need to install libusb (version 0.1) and libftdi in order to satisfy dependencies for xc3sprog, so let's do that:

```
$ sudo apt-get install libftdi1-2 libusb-0.1-4
  <not showing all the ouput here>
$ cd ~/FreedomStudio/SiFive/xc3sprog/xc3sprog-0.1.2-2019.04.1
$ ldd xc3sprog
    linux-vdso.so.1 =>  (0x00007ffc051b5000)
    libftdi1.so.2 => /usr/lib/x86_64-linux-gnu/libftdi1.so.2
(0x00007fbded75d000)
    libusb-0.1.so.4 => /lib/x86_64-linux-gnu/libusb-0.1.so.4
(0x00007fbded554000)
    libstdc++.so.6 => /usr/lib/x86_64-linux-gnu/libstdc++.so.6
(0x00007fbded171000)
    libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007fbdecf59000)
    libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fbdecb8f000)
    libusb-1.0.so.0 => /lib/x86_64-linux-gnu/libusb-1.0.so.0
(0x00007fbdec977000)
    libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007fbdec66e000)
    /lib64/ld-linux-x86-64.so.2 (0x00007fbded96b000)
    libudev.so.1 => /lib/x86_64-linux-gnu/libudev.so.1 (0x00007fbdedb54000)
    libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0
(0x00007fbdec451000)
    librt.so.1 => /lib/x86_64-linux-gnu/librt.so.1 (0x00007fbdec249000)
```

That looks good! Now both openocd and xc3sprog are ready to go.

Enable Access to USB Devices

By default, most Linux distributions do not give users permissions to access USB devices. One either needs root access or to be given the appropriate permissions.

Below are steps you can follow to access your development kit without sudo permissions (although sudo permissions are required for the initial setup):

Step 1: With your board's debug interface connected, make sure your device shows up with the lsusb command:

```
+

$ lsusb
  .
  .
  .
```

With your devices connected, check the output of the `lsusb` command to see that your devices are visible to the system. Use the table below to determine which entry you should see for your devices.

lsusb identifiers

Device	USB Identifier
Arty USB	Bus XXX Device XXX: ID 0403:6010 Future Technology Devices International, Ltd FT2232C Dual USB-UART/FIFO IC
HiFive2	Bus 001 Device 019: ID 0403:6011 Future Technology Devices International, Ltd FT4232H Quad HS USB-UART/FIFO IC
Olimex	Bus XXX Device XXX: ID 15ba:002a Olimex Ltd. ARM-USB-TINY-H JTAG interface.
HiFive1 RevB	Bus XXX Device XXX: ID 1366:1051 SEGGER
JLink Probe	Bus XXX Device XXX: ID 1366:0101 SEGGER J-Link PLUS

Step 2: Set the udev rules to allow the device to be accessed by the plugdev group:

Note

For your convenience a *99-freedomstudio.rules* file is included with Freedom Studio in the *FreedomStudio/SiFive/Misc* directory. You can install this file with this command:

```
$ sudo cp 99-freedomstudio.rules /etc/udev/rules.d/
```

The 99-freedomstudio.rules files installs rules that recognize the following USB devices and adds them to the plugdev group:

- Olimex ARM_USB_TINY_H
- HiFive2
- Arty Digilent USB

Step 3: See if your board shows up as a serial device belonging to the plugdev group. For instance with the Arty Board USB connector connected and an Olimex probe connected you should see something like

```
$ ls -l /dev/ttyUSB*
.
.
crw-rw-r-- 1 root plugdev 188, 0 Jun  7 11:01 /dev/ttyUSB0
crw-rw-r-- 1 root plugdev 188, 1 Jun  7 11:01 /dev/ttyUSB1
crw-rw-r-- 1 root plugdev 188, 2 Jun  7 11:07 /dev/ttyUSB2
```

.
.

But how do you know which serial port belongs to which device? You cannot tell from the output above. In fact, there is no simple way to do it, so we have provided a handy shell script called *listusb.sh* located in the *FreedomStudio/SiFive/Misc* directory.

Running that script yields much enlightenment:

```
$ ./listusb.sh
/dev/ttyUSB1 - Digilent_Digilent_USB_Device_210319A92CC9
/dev/ttyUSB0 - Digilent_Digilent_USB_Device_210319A92CC9
/dev/ttyUSB2 - 15ba_Olimex_OpenOCD_JTAG_ARM-USB-TINY-H_OL150D61
```

Note

If you have other serial devices or multiple boards attached, you may have more devices listed.

The ID (ttyUSB *X*) is assigned dynamically and is dependent on the order in which you connect your devices. Their assignment will change if you disconnect and reconnect in a different order. (But as long as you do not disconnect a device, its assigned ID will not change.)

Note

If your device present more than a single UART you will always want to select the higher number of the pair. In the example above you would want to use /dev/ttyUSB1

Note

The tty/USB device provided by the Olimex probe cannot be used as a UART. You can ignore this device.

Step 4: Add yourself to the plugdev group. You can use the whoami command to determine your user name.

```
> sudo usermod -a -G plugdev `whoami`
```

1. Log out and log back in, then check that you're now a member of the plugdev group:

```
$ groups
... plugdev ...
```

If you are not part of the plugdev group, perform a full reset.

Now you should be able to access the serial (UART) and debug interface without sudo permissions.

SiFive Copyright Notice

SiFive expressly disclaims all warranties, representations and conditions of any kind, whether expressed or implied, including, but not limited to, the implied warranties or conditions of merchantability, fitness for a particular purpose and non-infringement.

SiFive does not assume any liability rising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation indirect, incidental, special, exemplary, or consequential damages.

SiFive reserves the right to make changes without further notice to any products herein.

Software Licenses

Portions of Freedom Studio are governed by the following licenses

SiFive End User License Agreement

```
This be the placeholder!  Aarg!
```

Eclipse Public License - v 2.0

```
    THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE
    PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION
    OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

  a) in the case of the initial Contributor, the initial content
     Distributed under this Agreement, and

  b) in the case of each subsequent Contributor:
     i) changes to the Program, and
     ii) additions to the Program;
  where such changes and/or additions to the Program originate from
  and are Distributed by that particular Contributor. A Contribution
  "originates" from a Contributor if it was added to the Program by
  such Contributor itself or anyone acting on such Contributor's behalf.
  Contributions do not include changes or additions to the Program that
  are not Modified Works.

"Contributor" means any person or entity that Distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which
are necessarily infringed by the use or sale of its Contribution alone
or when combined with the Program.
```

"Program" means the Contributions Distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement or any Secondary License (as applicable), including Contributors.

"Derivative Works" shall mean any work, whether in Source Code or other form, that is based on (or derived from) the Program and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship.

"Modified Works" shall mean any work in Source Code or other form that results from an addition to, deletion from, or modification of the contents of the Program, including, for purposes of clarity any new file in Source Code form that contains any contents of the Program. Modified Works shall not include works that contain only declarations, interfaces, types, classes, structures, or files of the Program solely in each case in order to link to, bind by name, or subclass the Program or Modified Works thereof.

"Distribute" means the acts of a) distributing or b) making available in any manner that enables the transfer of a copy.

"Source Code" means the form of a Program preferred for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Secondary License" means either the GNU General Public License, Version 2.0, or any later versions of that license, including any exceptions or additional permissions as identified by the initial Contributor.

2. GRANT OF RIGHTS

  a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, Distribute and sublicense the Contribution of such Contributor, if any, and such Derivative Works.

  b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in Source Code or other form. This patent license shall apply to the combination of the Contribution and the Program if, at

the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to Distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

e) Notwithstanding the terms of any Secondary License, no Contributor makes additional grants to any Recipient (other than those set forth in this Agreement) as a result of such Recipient's receipt of the Program under the terms of a Secondary License (if permitted under the terms of Section 3).

3. REQUIREMENTS

3.1 If a Contributor Distributes the Program in any form, then:

a) the Program must also be made available as Source Code, in accordance with section 3.2, and the Contributor must accompany the Program with a statement that the Source Code for the Program is available under this Agreement, and informs Recipients how to obtain it in a reasonable manner on or through a medium customarily used for software exchange; and

b) the Contributor may Distribute the Program under a license different than this Agreement, provided that such license:
   i) effectively disclaims on behalf of all other Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness

for a particular purpose;

ii) effectively excludes on behalf of all other Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) does not attempt to limit or alter the recipients' rights in the Source Code under section 3.2; and

iv) requires any subsequent distribution of the Program by any party to be under a license that satisfies the requirements of this section 3.

3.2 When the Program is Distributed as Source Code:

a) it must be made available under this Agreement, or if the Program (i) is combined with other material in a separate file or files made available under a Secondary License, and (ii) the initial Contributor attached to the Source Code the notice described in Exhibit A of this Agreement, then the Program may be made available under the terms of such Secondary Licenses, and

b) a copy of this Agreement must be included with each copy of the Program.

3.3 Contributors may not remove or alter any copyright, patent, trademark, attribution notices, disclaimers of warranty, or limitations of liability ("notices") contained within the Program from any copy of the Program which they Distribute, provided that Contributors may add their own appropriate notices.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not

apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be Distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to Distribute the Program (including its Contributions) under the new version.

Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved. Nothing in this Agreement is intended to be enforceable by any entity that is not a Contributor or Recipient. No third-party beneficiary rights are created under this Agreement.

Exhibit A - Form of Secondary Licenses Notice

"This Source Code may also be made available under the following Secondary Licenses when the conditions for such availability set forth in the Eclipse Public License, v. 2.0 are satisfied: {name license(s), version(s), and exceptions or additional permissions here}."

  Simply including a copy of this Agreement, including this Exhibit A is not sufficient to license the Source Code under Secondary Licenses.

  If it is not possible or desirable to put the notice in a particular file, then You may include the notice in a location (such as a LICENSE file in a relevant directory) where a recipient would be likely to look for such a notice.

  You may add additional accurate notices of copyright ownership.

GNU GENERAL PUBLIC LICENSE, V2

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

  The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

  When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you

distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest

your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is

implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

  8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

  9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

  10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

  11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS

TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the program's name and a brief idea of what it does.>
> Copyright (C) <year> <name of author>
>
> This program is free software; you can redistribute it and/or modify
> it under the terms of the GNU General Public License as published by
> the Free Software Foundation; either version 2 of the License, or
> (at your option) any later version.
>
> This program is distributed in the hope that it will be useful,
> but WITHOUT ANY WARRANTY; without even the implied warranty of
> MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
> GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along
> with this program; if not, write to the Free Software Foundation, Inc.,
> 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this

when it starts in an interactive mode:

```
    Gnomovision version 69, Copyright (C) year name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
  `Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

GNU GENERAL PUBLIC LICENSE, V3

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The GNU General Public License is a free, copyleft license for
software and other kinds of works.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
```

GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could

make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

  The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

  To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

  A "covered work" means either the unmodified Program or a work based on the Program.

  To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited

permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,

and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product

(including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation

into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions.

Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this

License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However,

nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free

patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you

to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program. It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year> <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program> Copyright (C) <year> <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show
w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License. Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library. If this is what you want to do, use the GNU Lesser General

Public License instead of this License. But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.