
SoftConsole v5.0

Release Notes

Table of Contents

Table of Contents	2
SoftConsole v5.0	5
Overview	5
Document conventions	5
Supported platforms	5
Known issues	6
Other resources	6
Installing SoftConsole	6
Before installation	6
Installation.....	7
After installation	7
Running SoftConsole	8
Creating a new project.....	9
Configuring project properties	9
Building a project	9
Debugging a project.....	9
Built-in serial terminal view	10
Other useful Documentation	10
Free/Open Source Packages	11
Packages Used.....	11
Licensing.....	11
Documentation.....	11
Product Support	13
Customer Service	13
Customer Technical Support Center	13
Technical Support	13
Website	13

Contacting the Customer Technical Support Center.....	13
Email.....	13
My Cases.....	13
Outside the U.S.	14
ITAR Technical Support	14

SoftConsole v5.0

Overview

SoftConsole v5.0 supports development and debugging of software targeting RISC-V CPU based SoCs on Microsemi FPGA devices.

Document conventions

1. This document assumes that SoftConsole is installed into the default location `$HOME/Microsemi_SoftConsole_v5.0` and some instructions refer to this directory. If SoftConsole is installed elsewhere then substitute the actual install directory for the default.
2. The installer executable is referred to as `Microsemi-SoftConsole-v5.0.0.x-Linux-x86-Installer` where 'x' means the build number. Substitute the actual build number for 'x'.
3. Some commands that must be run in a command line terminal shell are outlined below. In some cases these require root privileges in which case they are shown executed using `sudo`. If `sudo` does not work – for example, because the user does not have `sudo` (sudoers) privileges – then use `su` instead.

Supported platforms

SoftConsole has been tested on and currently supports the following host platforms:

1. Ubuntu 14.04.5 LTS Desktop 32 and 64 bit
2. Ubuntu 16.04.1 LTS Desktop 32 and 64 bit
3. CentOS/Red Hat Enterprise Linux 6.8 Desktop 32 and 64 bit
4. CentOS/Red Hat Enterprise Linux 7.2 Desktop 64 bit
5. openSuse Leap 42.1 [Gnome/Kde desktop] 64 bit

The following host platforms are not yet supported:

1. Linux distributions/versions other than those listed above
2. Microsoft Windows
3. Virtual machines – SoftConsole will install and can compile/link projects on VMware and Oracle VM VirtualBox virtual machines but, due to a problem with the FlashPro Fpcommwrapper and FTDI support libraries, FlashPro JTAG debugging is unreliable and may not work at all. As there is no fix planned for this problem only physical rather than virtual machines are recommended and supported for debugging.

The following hardware has been used for testing (running RISC-V based SoCs):

1. Microsemi SmartFusion2 M2S025 Creative Development Kit board
<http://www.microsemi.com/products/fpga-soc/design-resources/dev-kits/smartfusion2/future-creative-board>
2. Microsemi SmartFusion2 M2S090 Security Evaluation Kit board
<http://www.microsemi.com/products/fpga-soc/design-resources/dev-kits/smartfusion2/sf2-evaluation-kit>
3. Microsemi SmartFusion2 M2S150 Advanced Development Kit board
<http://www.microsemi.com/products/fpga-soc/design-resources/dev-kits/smartfusion2/smartfusion2-advanced-development-kit>
4. FlashPro5 JTAG programmer
<http://www.microsemi.com/products/fpga-soc/design-resources/programming/flashpro#hardware>

Known issues

1. This version of SoftConsole does not support Cortex-M targets. Use SoftConsole v4.x for Cortex-M development and debugging.
2. The target board must be manually reset or power cycled each time before establishing a debug connection to ensure reliable connection. If this is not done then OpenOCD errors may result and the debug connection may fail.
3. When debugging there may be a noticeable pause between initiating the debug connection and the program downloading to the target.
4. Development of RISC-V 32 bit software is currently supported. Development of RISC-V 64 bit software is not yet supported.
5. Several RISC-V project properties presented are not applicable/compatible with the Microsemi RISC-V implementation – for example, hardware float options, 64 bit options etc.

Other resources

1. The Microsemi RISC-V github repository contains example hardware and firmware projects, RISC-V HAL (Hardware Abstraction Layer) firmware support, example RISC-V firmware projects etc. <https://github.com/RISCV-on-Microsemi-FPGA>. The github repository contains a number of sub-folders each of which can be retrieved using git. For example to get the riscv-junk-drawer contents:

```
git clone https://github.com/RISCV-on-Microsemi-FPGA/riscv-junk-drawer
```

Installing SoftConsole

Before installation

SoftConsole is a 32 bit application. Before it can be installed or run a number of 32 bit packages/libraries must be installed.

Ubuntu 64 bit

1. `sudo dpkg --add-architecture i386`
2. `sudo apt-get update`
3. `sudo apt-get install -y libgtk2.0-0:i386`
4. `sudo apt-get install -y libxtst6:i386`
5. `sudo apt-get install -y libusb-1.0-0:i386`
6. `sudo apt-get install -y lib32ncurses5`
7. `sudo apt-get install -y libstdc++6:i386`

CentOS/Red Hat Enterprise Linux 6.8 64 bit

1. `sudo yum install -y gtk2.i686`
2. `sudo yum install -y libXtst.i686`
3. `sudo yum install -y libusb1.i686`
4. `sudo yum install -y ncurses-libs.i686`

CentOS/Red Hat Enterprise Linux 7.2 64 bit

1. `sudo yum install -y gtk2.i686`
2. `sudo yum install -y libXtst.i686`
3. `sudo yum install -y libusbx.i686`
4. `sudo yum install -y ncurses-libs.i686`

Ubuntu/CentOS/Red Hat Enterprise Linux 32 bit

- All required packages already installed by default

openSUSE Leap 42.1 64 bit

1. `sudo zypper install -y gtk2-tools-32bit`
2. `sudo zypper install -y libXtst6-32bit`
3. `sudo zypper install -y libusb-1_0-0-32bit`
4. `sudo zypper install -y libncurses5-32bit`
5. `sudo zypper install -y libgthread-2_0-0-32bit`

Notes:

1. The `apt-get/yum install` commands above can be merged into one line specifying all relevant packages for convenience but are listed separately above for clarity.
2. It is recommended that the host Linux platform is up to date with all updates available through the system's package manager.
3. It is not a problem if any of the packages listed are reported as already installed.
4. On CentOS/Red Hat Enterprise Linux 64 bit if the following error occurs when attempting to install any of the 32 bit packages

```
Error: Protected multilib versions ...
```

then first upgrade the relevant 64 bit package and then try again to install the 32 bit package. For example, if the error occurred when installing `gtk2.i686` then do this:

```
sudo yum upgrade -y gtk2
sudo yum install -y gtk2.i686
```

Installation

1. Download the installer executable named `Microsemi-SoftConsole-v5.0.0.x-Linux-x86-Installer`.
2. If necessary mark the installer as executable (assuming that the installer executable has been downloaded to the `$HOME/Downloads` directory):

```
cd ~/Downloads
chmod +x Microsemi-SoftConsole-v5.0.0.x-Linux-x86-Installer
```

3. Run the installer::

```
./Microsemi-SoftConsole-v5.0.0.x-Linux-x86-Installer
```

4. Follow the installation GUI wizard instructions on screen. If it does not appear then double check that the required dependent packages/libraries were installed as explained previously.

After installation

By default USB devices are only accessible with root privileges. In order to debug using SoftConsole and FlashPro5 as a non root user some additional steps must be taken.

1. The user running SoftConsole must be a member of the `plugdev` group. Ubuntu creates the `plugdev` group and adds new users to it by default. If, for some reason, neither of these applies then the `plugdev` group must be created and the user account under which SoftConsole will be run must be made a member

of the `plugdev` group. This is the case on CentOS/Red Hat Enterprise Linux.

2. Copy the OpenOCD udev rules file and tell udev to load it:

```
cd $HOME/Microsemi_SoftConsole_v5.0/openocd/share/openocd/contrib
sudo cp 99-openocd.rules /etc/udev/rules.d
sudo udevadm trigger
```

3. Connect a FlashPro5 programmer to the target board which is programmed with a RISC-V SoC design. Connect the FlashPro5 programmer to the computer. Power on the target board. Check that the FlashPro device is visible to the operating system:

```
lsusb
```

```
Bus 001 Device 004: ID 1514:2008 Actel
```

If the FlashPro5 device (id 1514:2008 or vendor id 0x1514, product id 0x2008) does not appear then double check that the previous instructions were carried out correctly.

4. Run OpenOCD from the command line and check that it can connect to the target via the FlashPro5:

```
cd ~/Microsemi_SoftConsole_v5.0/openocd/bin
export LD_LIBRARY_PATH=`pwd`
./openocd -f board/microsemi-riscv.cfg
```

The following indicates that everything is working as expected and OpenOCD can be terminated using Ctrl-C. Note that the specific FlashPro port/id will vary.

```
Info : FlashPro ports available: S201Z7LB20
Info : FlashPro port used: S201Z7LB20
Info : clock speed 6000 kHz
Info : JTAG tap: FPGA.tap tap/device found: 0x1f8071cf (mfg: 0x0e7 (GateField),
part: 0xf807, ver: 0x1)
microsemi_flashpro tunnel_jtag_via_ujtag on
Info : JTAG tap: FPGA.tap disabled
Info : JTAG tap: FPGA.dap enabled
Info : RISC-V IDCODE = 0x10e31913
Info : Examined RISC-V core; XLEN=32, misa=0x40902223
halted at 0x60000700 due to debug interrupt
```

The following or any other errors indicates that there is still a problem in which case double check that the previous instructions were carried out correctly. Also try a reboot and try again in case the udev rules did not reload at step 2.

```
Info : FlashPro ports available: none
Info : FlashPro port used: usb
Error: InitializeProgrammer(usb) failed : Can not connect to the programmer
```

Running SoftConsole

Run SoftConsole from the shortcut installed on the desktop or from the system menu. By default it will open the example workspace in `~/Microsemi_SoftConsole_v5.0/extras/workspace.examples` which contains one or more example projects for illustrative purposes. Refer to the Microsemi github repository for more examples and other relevant resources. New workspaces can be created if required.

Creating a new project

To create a new RISC-V C project:

1. Select *File > New > C Project* from the SoftConsole main menu.
2. Enter a *Project name*
3. Under *Project type* select *RISC-V Embedded Application > Empty Project*
4. Under *Toolchains* ensure that *RISC-V GCC/Newlib Toolchain* is selected
5. Click Finish

Notes:

1. RISC-V C++ projects can also be created by varying the instructions above
2. *Project type > RISC-V Embedded Static Library > Empty Project* can be selected to create a static library project.
3. *Project type > RISC-V Embedded Application > Hello World C Project* can be selected to create a simple "Hello world" project.
4. New application projects will normally benefit from the addition of the Microsemi RISC-V Hardware Abstraction Layer firmware which provides an abstraction layer on which Microsemi DirectCore peripheral IP and other firmware drivers sit.

Configuring project properties

Most project properties are set to acceptable defaults although can be changed to suit the specific needs of the user/application.

1. Select the project in the SoftConsole *Project Explorer*.
2. Right click on the project and select *Properties* from the context menu or select the *Project > Properties* from the SoftConsole main menu.
3. Select *C/C++ Build > Settings > Tool Settings* to see or change the various project and tool settings.
4. Select *Configuration = Debug* or *Active* or *[All Configurations]* depending on the desired scope of any settings that are entered/changed.
5. Changes commonly made to a project include:
 - a. *RISC-V GCC/Newlib C/C++ Compiler*
 - i. *Preprocessor* – symbols to be defined/undefined at preprocessing time.
 - ii. *Directories* – include paths to be searched at preprocessing time
 - b. *RISC-V GCC/Newlib C/C++ Linker*
 - i. *General > Script file (-T)* – linker script to be used at link time.

Building a project

Once the required source files (including Microsemi RISC-V HAL, Microsemi DirectCore IP core firmware driver source files, application code etc.) have been added to the project and any necessary project properties have been configured appropriately the project can be built.

1. Select the project in the SoftConsole *Project Explorer*.
2. From the SoftConsole main menu select *Project > Build Project* or one of the other build options. The SoftConsole GUI provides several other ways to trigger a project build.
3. Errors/warnings, if any, are displayed in the *Console* and *Problems* views where they can be reviewed and addressed.
4. When any errors are fixed the project will compile and link.

Debugging a project

Once a project has compiled and linked it can be debugged on hardware. Ensure that the necessary post installation tasks to ensure that FlashPro5/OpenOCD work have been successfully completed. Refer to the example project(s) in the example workspace for examples of how a project and debug launch configuration can be configured.

To create a debug launch configuration for a new project:

1. Select the project in the SoftConsole *Project Explorer*.
2. Right click on the project and select *Debug As > Debug Configurations...*
3. In the *Debug Configurations* dialog right click on *GDB OpenOCD Debugging* and select *New*.
4. Most of the debug launch configuration settings are set correctly by default – as long as the defaults were not changed when previously creating a debug launch configuration. Certainly with a new/clean workspace the default settings will be set appropriately. However at least one setting must be changed for each project:
 - a. *Debugger > GDB Client Setup > Other options* – this must always be manually set to the name of the executable program ELF file to be debugged – e.g. *Debug/systick-blinky.elf*. This file is passed to GDB and ensures that the RISC-V GDB operates in the appropriate mode (32 or 64 bit). Failure to pass the correct executable name will result in a size mismatch debugging error.
5. Other debug launch configuration settings are correct by default and generally do not need to be changed. In particular the following options must be left unchecked for now:
 - a. *Startup > Initialization Commands > Initial Reset*
 - b. *Startup > Initialization Commands > Enable ARM Semihosting*
 - c. *Startup > Runtime Options > Debug in RAM*
 - d. *Startup > Run/Restart Commands > Pre-run/Restart reset*
6. By default debug launch configurations are stored in the workspace but it is often convenient to configure them to be stored in the project instead. To do this change *Common > Save as > Local file to Shared file*.

Built-in serial terminal view

SoftConsole includes a built-in serial terminal view which obviates the need to run a separate serial terminal emulator when connecting to a target board using a UART. The plug-ins used to implement this view are pre-installed. Refer to this blog post for information on how to show and configure the terminal view (but skip the parts dealing with plug-in installation as this is already done):

<http://mcuoneclipse.com/2015/04/20/serial-terminal-view-in-eclipse-luna/>

Other useful Documentation

1. Erich Styger's "MCU on Eclipse" blog (<http://mcuoneclipse.com/>): Useful tips and tricks for using Eclipse/CDT, GNU ARM Eclipse, GNU Tools for ARM Embedded Processors, OpenOCD etc. The [Compendium page](#) is a good place to find posts/articles relevant to Eclipse, OpenOCD etc.
2. The websites and documentation links for the various open source components used in SoftConsole are also useful references. These are listed below.

Free/Open Source Packages

Packages Used

Microsemi SoftConsole v5.0 uses a number of free and/or open source packages. Microsemi acknowledges and thanks those organizations and developers who work on these packages and make them available to others for reuse under the relevant license conditions.

Package	Version
Oracle Java SE	8u102
Eclipse/CDT	Eclipse 4.4.2 (Luna SR2) CDT 8.6.0 for Eclipse Luna with Microsemi modified starter.exe (on Windows) to allow for graceful termination of OpenOCD.
GNU ARM Eclipse plugins	2.12.1-201604190915
Roa Logic BV RISC-V Eclipse plugins	2016.02.0.qualifier with Microsemi modifications to some project property defaults
RISC-V GCC/GDB/Binutils/Newlib development tools	GC 6.1.0
OpenOCD	0.10.0 with SiFive RISC-V support and Microsemi enhancements
InstallJammer (Linux only)	1.2.15
RXTX Java library and Eclipse plug-ins	2.1-7r4

Licensing

Microsemi SoftConsole v5.0 uses a number of free and/or open source packages whose use is governed by the specified license agreements.

Package	License
Oracle Java SE	Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX
Eclipse/CDT	Eclipse Public License
GNU ARM Eclipse plugins	Eclipse Public License
Roa Logic BV RISC-V Eclipse plugins	Eclipse Public License
RISC-V GCC/GDB/Binutils/Newlib development tools	https://github.com/riscv/riscv-gnu-toolchain/blob/master/LICENSE
OpenOCD	GNU General Public License v3
InstallJammer	GNU General Public License with exception
RXTX Java library and Eclipse plug-ins	LGPL v2.1 + Linking Over Controlled Interface

Documentation

Package	Documentation
Microsemi RISC-V github	https://github.com/RISCV-on-Microsemi-FPGA
Eclipse/CDT	Eclipse documentation CDT documentation
GNU ARM Eclipse plugins	Documentation

Roa Logic BV RISC-V Eclipse plugins	Github
OpenOCD	User's Guide
RXTX Java library and Eclipse plug-ins	Documentation

Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060**

From the rest of the world, call **650.318.4460**

Fax, from anywhere in the world **408.643.6913**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

For Microsemi SoC Products Support, visit <http://www.microsemi.com/products/fpga-soc/design-support/fpga-soc-support>.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group [home page](http://www.microsemi.com/soc/), at <http://www.microsemi.com/soc/>.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to [My Cases](#).

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Visit [About Us](#) for [sales office listings](#) and [corporate contacts](#).

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within [My Cases](#), select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the [ITAR web page](#).



Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

© 2016 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,600 employees globally. Learn more at www.microsemi.com.

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.