

# CWARMv8GS

## CodeWarrior Development Studio for QorIQ LS series - ARM V8 ISA Hardware Board Getting Started Guide

Rev. 11.5.0 — 06/2020

User's Guide

### 1 Introduction

This document explains how to install the CodeWarrior Development Studio for QorIQ LS series - ARM V8 ISA.

Additionally, it describes how to prepare the boards, use the CodeWarrior to create, build, and debug a simple project.

This chapter contains:

- [System requirements](#)
- [Supported devices](#)

#### 1.1 System requirements

This section provides the operating platform requirements for installing the CodeWarrior software.

**Table 1. System Requirements**

|                  |  |
|------------------|--|
| Processor        | Windows® OS: Intel® Pentium® 4 processor, 2 GHz or faster, Intel® Xeon™, Intel® Core™, AMD Athlon™ 64, AMD Opteron™, or later<br><br>Linux® OS: 1.8 GHz Intel® Pentium® class processor (or better). 64-bit host OS required.  |
| Hardware         | <ul style="list-style-type: none"><li>• CD-ROM drive for CD installation</li><li>• Microsoft® Mouse compliant pointing device</li><li>• Internet connectivity for web downloads and update access</li></ul>  |
| Operating System | <ul style="list-style-type: none"><li>• Microsoft® Windows 7 64-bit</li><li>• Microsoft® Windows 10 64-bit</li><li>• RHEL 7.8, 8.2 64-bit</li><li>• CentOS 7.8, 8.2 64-bit</li><li>• Ubuntu 16.04 64-bit</li><li>• Ubuntu 18.04 64-bit</li><li>• Ubuntu 20.04 64-bit</li><li>• Fedora 31 64-bit</li><li>• Mint 19.3 64-bit</li></ul> |

*Table continues on the next page...*

### Contents

|   |          |
|---|----------|
| <b>1 Introduction.....</b>  | <b>1</b> |
| 1.1 System requirements....   | 1        |
| 1.2 Supported devices.....  | 2        |
| <b>2 Configuring Target Hardware.....</b>   | <b>2</b> |
| 2.1 Preparing LS1088ARDB board.....   | 2        |
| 2.2 Preparing LS1046ARDB board.....   | 3        |
| 2.3 Preparing LS1012ARDB board.....   | 5        |
| 2.4 Preparing LS2088RDB and board.....  | 6        |
| 2.5 Preparing LS1043ARDB board.....   | 7        |
| <b>3 Installing CodeWarrior Software and Creating, Building, and Debugging a Bareboard Project.....</b> | <b>8</b> |
| 3.1 Installing and registering CodeWarrior software....   | 8        |
| 3.2 Working with bareboard application..  | 10       |
| 3.2.1 Creating, building bareboard project..  | 10       |
| 3.2.2 Debugging bareboard project..   | 12       |
| 3.3 Working with Linux application.....   | 15       |
| 3.3.1 Creating simple Linux application project.  | 15       |
| 3.3.2 Updating remote connection.....   | 16       |
| 3.3.3 Using sysroot.....  | 18       |
| 3.3.4 Debugging Linux application project.....  | 20       |



**Table 1. System Requirements (continued)**

|        |  |
|--------|--|
| Memory | At least 2 GB of RAM<br>At least 3 GB of free disk space |
|--------|--|

**NOTE**

CodeWarrior installation on 64-bit Linux hosts requires presence of dependent 32-bit libraries in the system. For supported distributions, the installer detects missing dependencies and provides options to install them automatically. It is recommended (although not required) that all product maintenance operations are performed with administrative privileges. When running the installer in GUI mode with admin privileges, please use the graphical sudo command recommended by your Linux distribution (ie. gksu/gksudo).

## 1.2 Supported devices

The following table lists the device families and the targets supported by CodeWarrior for ARMv8.

**Table 2. Supported target devices**

| Device Family        | Description   |
|----------------------|---|
| LS2088A              | Supports generation of multi-core target projects for LS2088ARDB, LS2088AQDS. |
| LS1012A <sup>1</sup> | Supports generation of multi-core target projects for LS1012ARDB, LS1012AQDS. |
| LS1043A              | Supports generation of multi-core target projects for LS1043ARDB, LS1043QDS.  |
| LS1088A              | Supports generation of multi-core target projects for LS1088ARDB, LS1088QDS.  |
| LS1046A              | Supports generation of multi-core target projects for LS1046ARDB, LS1046QDS.  |

1. Not hot pluggable card. Please ensure that board is OFF while inserting the cards.

## 2 Configuring Target Hardware

This chapter explains how to configure the boards supported by the CodeWarrior for QorIQ LS series software.

You can configure board using an integrated development environment (IDE), such as Freescale's CodeWarrior IDE.

### 2.1 Preparing LS1088ARDB board

This topic explains steps to configure LS1088ARDB board.

1. Ensure that all hardware kit contents are present in your package by using the list enclosed with the LS1088ARDB system.
2. Check and verify the LS1088RDB default switch, connector, and jumper settings. Establish the working environment for LS1088ARDB.

For more details, see LS1088ARDB User Guide.

3. For QSPI boot, use the following switch settings:

**Table 3. QSPI boot**

| Switch            | Value           |
|-------------------|-----------------|
| <b>LS1088ARDB</b> |                 |
| SW1               | 0x31 = 00110001 |
| SW2               | 0x40 = 01000000 |

*Table continues on the next page...*

**Table 3. QSPI boot (continued)**

| Switch | Value           |
|--------|-----------------|
| SW3    | 0xF2 = 11110010 |
| SW4    | 0x93 = 10010011 |
| SW5    | 0x70 = 01110000 |

Where '1' = up/ON

The settings above configure the board for:

- RCW\_SRC = 9b'0x62 - QSPI Port "A"
- SYSCLK = 100Mhz (Differential)
- DDRCLK = 100 MHz (Fixed)

4. Optionally, connect the CodeWarrior TAP to the LS1088ARDB.
5. Install the AC power cord from the chassis to the local mains outlet. If the power supply has a switch, turn it to ON (NOTE: Not all ATX power supplies will have an AC switch.) Status LED D2 ("3V3\_SB") will be on when the standby power is ready.

After the system has been properly set up, follow these steps to switch on the board and use it:

1. Turn ON the power switch.  
The green power supply monitors and the blue PASS LED will be on. Status LED will show an idle pattern.
2. If any red or amber LED is on, there may be a problem with the configuration or software.
3. Execute Tera Term.
4. Select the Serial option in Tera Term and ensure that mbed serial port is selected.
5. Click OK.
6. Select Setup > Serial port and configure the host computer's serial port with the following settings:
  - Baud rate: 115200 bit/s
  - Number of data bits: 8
  - Stop bit: 1
  - Parity: None
  - Flow control: None
7. Click OK.
8. Press the Reset switch and the board boots up.
9. The boot log appears on the Console window.

## 2.2 Preparing LS1046ARDB board

This topic explains steps to configure LS1046ARDB board.

1. For QSPI boot, use the following settings:

**Table 4. QSPI boot**

| Switch | Value           |
|--------|-----------------|
| SW3    | 0x46 = 01000110 |

*Table continues on the next page...*

**Table 4. QSPI boot (continued)**

| Switch | Value           |
|--------|-----------------|
| SW4    | 0x3B = 00111011 |
| SW5    | 0x22 = 00100010 |

Where '1' = UP/ON

These settings above configure the board for:

- RCW\_SRC = 9b'0x44 Serial NOR (QSPI)
  - SYSCLK = 100 MHz (Differential)
  - DDRCLK = 100 MHz (Derived internally from differential SYSCLK)
  - SD1REFCLK = 156.25 MHz
2. Connect the 12 V DC power adaptor to the wall outlet and turn on the power supply.
  3. Optionally, connect the CodeWarrior TAP to the LS1046ARDB.
  4. Optionally, connect the Ethernet cable if you want to connect your board to the network, for example, for obtaining the latest board software and updating board images.
  5. Make a console port connection from the board to see console output:
    - For Standalone mode (board setup with only one board):
      - a. Ensure that SW4[4] is ON (default value).
      - b. Using the USB 2.0 Type-A to Micro-B cable provided with the hardware kit, connect the port with the "CONSOLE" silk on the chassis front panel to a USB port on the host machine, as shown in the figure below. The D8 LED on the PCB turns ON.
    - For board farm:
      - a. Turn SW4[4] to OFF.
      - b. Using the RJ45-to-DB9F cable provided with the hardware kit, connect the port with the "UART1" silk on the chassis back panel to the host machine serial port, USB-to-DB9 convertor, or other devices, such as the CodeWarrior TAP, as shown in the figure below.

Next steps in this section are based on the Standalone mode

6. Turn ON the power switch (SW2). The power LED (D3) on the printed circuit board (PCB) and the LED with "POWER" silk on the chassis front panel turn green.

**NOTE**

The D7 (ASLEEP) LED on the PCB remains OFF if correct RCW is fetched.

7. Install the mbed Windows serial port driver on the host machine to use the USB serial port. See Software requirements for details.

**NOTE**

This is a one-time activity. Skip this step if you have already installed the mbed driver on the host machine. If this driver is installed, then the host machine automatically installs the USB driver and detects the USB device.

8. Execute Tera Term.
9. Select the **Serial** option in Tera Term and ensure that mbed serial port is selected.
10. Click **OK**.
11. Select **Setup > Serial port** and configure the host computer's serial port with the following settings:

- Baud rate: 115200 bit/s
- Number of data bits: 8
- Stop bit: 1
- Parity: None
- Flow control: None

12. Click **OK**.
13. Press the Reset switch (SW1) and the board boots up.  
The console shows the U-Boot messages.

### 2.3 Preparing LS1012ARDB board

This topic explains steps to configure LS1012ARDB board.

1. For QSPI boot, use the following settings:

**Table 5. QSPI boot**

| Switch | Value           |
|--------|-----------------|
| SW1    | 0xA6 = 10100110 |
| SW2    | 0x00 = 00000000 |

Where '1' = UP/ON

These settings above configure the board for:

- RCW\_SRC = QSPI

The hard coded switch settings are:

**Table 6. Hard coded switch settings**

| Switch | Value           |
|--------|-----------------|
| SW1    | 0x26 = 00100110 |
| SW2    | 0x00 = 00000000 |

Where '1' = UP/ON

2. Connect the 12 V DC power adaptor to the wall outlet.
3. Turn on the power supply. The 12 V LED (D4) and the power status LED (D19) turn green. Also, PORST Red LED (D103) will turn on and turn off immediately.
4. Optionally, the board can also be powered from PoE or a USB port (with battery backed option).
5. Plug the type-A connector of the USB cable into the USB port of your PC and connect type-B connector to the onboard USB 2.0 AB (Debug) connector, J13.

**NOTE**

Your PC will automatically install the USB driver and detect the USB device.

6. Install the mbed Windows serial port driver that you downloaded, as described in the Prerequisites section.

**NOTE**

This is a one time activity, please ignore if you have already installed mbed driver on your system (PC or laptop).

7. Optionally, connect the CodeWarrior TAP to the LS1012ARDB. The board also contains an onboard low-speed debugger (CMSIS-DAP) accessible from the USB debug connector, J13.
8. Optionally, connect the Ethernet cable if you want to connect your board to the network, for example, for obtaining latest board software and updating board images.
9. Execute Tera Term.
10. Select the Serial option in Tera Term and ensure that mbed serial port is selected.
11. Click OK.
12. Select Setup > Serial port and configure the host computer's serial port with the following settings:
  - Baud rate: 115200 bit/s
  - Number of data bits: 8
  - Stop bit: 1
  - Parity: None
  - Flow control: None
13. Click OK.
14. Press the Reset switch (SW3) and the board boots up.
15. The board boots up, The console shows the U-Boot messages.

## 2.4 Preparing LS2088RDB and board

This topic explains steps to configure LS2088RDB board.

Perform the following steps to configure a LS2088ARDB board:

1. Verify that all hardware kit contents are present.
2. Check and verify the LS2088ARDB default switch, connector, and jumper settings. Establish the working environment for LS2088ARDB.

For more details, see LS2088ARDB User Guide.

3. For NOR boot use the following switch settings:

**Table 7. NOR boot - LS2088ARDB**

| Switch            | Value           |
|-------------------|-----------------|
| <b>LS2088ARDB</b> |                 |
| SW4               | 0xFF = 11111111 |
| SW3               | 0x12 = 00010010 |
| SW5               | 0xFF = 11111111 |
| SW7               | 0x42 = 01000010 |
| SW8               | 0x5F = 01011111 |
| SW9               | 0x40 = 01000000 |
| SW6               | 0xFF = 11111111 |

Where '1' = DOWN/ON

These settings above configure the board for:

- RCW\_SRC = 9b'0x25 - NOR Flash 16-bit

- SYSCLK = 100 MHz
  - DDRCLK = 133 MHz
  - IFC - CS0 - NOR (vBank:0), CS2 - NAND, CS3 - CPLD
4. Connect the CodewarriorTAP. For details, see the board User Guide.
  5. Install the AC power cord from the chassis to the local mains outlet. If the power supply has a switch, turn it to ON (NOTE: Not all ATX power supplies will have an AC switch.) Status LED "LED6" will be on when the standby power is ready.
  6. Press the power switch (SW5) momentarily.
  7. The power switch will blink green, then amber, then become solid green when the system is stable. The status LEDs on the PCB will run through various patterns while powering up and resetting the board. When done, the bright green VDD\_POK (LED10) and GVDD\_POK (LED11) power LEDs will be on. LED3 will be on while the boot code is fetched. LED2/LED3 will show an idle pattern.

## 2.5 Preparing LS1043ARDB board

This topic explains steps to configure LS1043ARDB board.

Perform the following steps to configure a LS1043ARDB board:

1. Verify that the LS1043ARDB hardware kit contains all items mentioned in the kit contents list provided with the LS1043ARDB packing list.
2. Verify the LS1043ARDB dual inline package (DIP) switch settings.
3. For NOR boot use the following switch settings:

**Table 8. NOR boot**

| Switch | Value           |
|--------|-----------------|
| SW3    | 0x92 = 10010010 |
| SW4    | 0x12 = 00010010 |
| SW5    | 0xA0 = 10100000 |

Where '1' = UP/ON

These settings above configure the board for:

- RCW\_SRC = 9b'0x25 - NOR flash 16-bit
  - SYSCLK = 100 MHz (Differential)
  - DDRCLK = 100 MHz (Derived internally from differential SYSCLK)
  - IFC - CS0 - NOR (vBank:0), CS1 - NAND, CS2 - CPLD
4. For NAND boot use the following switch settings:

**Table 9. NAND boot**

| Switch | Value           |
|--------|-----------------|
| SW3    | 0x92 = 10010010 |
| SW4    | 0x83 = 10000011 |
| SW5    | 0x20 = 00100000 |

Where '1' = UP/ON

These settings above configure the board for:

- RCW\_SRC = 9b'0x106 - NAND flash 8-bit, 2 KB page, 64 page/block
  - SYSCLK = 100 MHz (Differential)
  - DDRCLK = 100 MHz (Derived internally from differential SYSCLK)
  - IFC - CS0 - NAND, CS1 - NOR, CS2 - CPLD
5. Verify the LS1043ARDB jumper settings.
  6. Attach the required devices to the LS1043ARDB connectors.
  7. Optionally, connect the CodeWarrior TAP to the LS1043ARDB.

---

**NOTE**

For details about switch configurations, jumper settings, connectors, and setting up CodeWarrior TAP, see LS1043ARDB Quick Start.

---

8. Install the DC power cord from the desktop power supply to the board power inlet.

After the system has been properly set up, follow the steps below to power up and use the LS1043ARDB:

1. Turn ON the power switch (SW2).
2. The power LED (D1) on the printed circuit board (PCB) and the left LED (D13) of the chassis become green when the board is powered up. D3 LED on the PCB remains OFF while the boot code is fetched.
3. Insert the universal serial bus (USB) cable to the mini-USB port in the front panel used for console. You need to install the USB drivers on the host machine before using the serial terminal. You can download the latest drivers from <https://developer.mbed.org/handbook/Windows-serial-configuration>.
4. If you are using the RJ45 port, insert the RJ45 to UART cable into the J4 bottom port and set SW3[3] to OFF.  
The boot log appears on the Console window.

## 3 Installing CodeWarrior Software and Creating, Building, and Debugging a Bareboard Project

This chapter explains how to install the CodeWarrior Development Studio for QorIQ LS series - ARM V8 ISA software.

Additionally, it describes how to use the software to create, build, and debug a demonstration multi-core processor project.

This chapter explains:

- [Installing and registering CodeWarrior software](#)
- [Working with bareboard application](#)
- [Working with Linux application](#)

### 3.1 Installing and registering CodeWarrior software

This section provides the steps required to install the CodeWarrior software on both Windows and Linux operating systems.

#### Microsoft® Windows OS installation:

Administrator rights are required to install CodeWarrior on Microsoft Windows operating system, since the installer copies files into the System and Program Files folders. CodeWarrior service packs are installed with the **Eclipse Updater**. The updater must also be run using administrator rights. To start the Eclipse Updater, select **'Help > Install new software'** in the menu.

The default CodeWarrior installation folder is `C:\Freescale`. To protect against malware, Windows operating system do not allow normal processes to change files in the *Program Files* folder; therefore, you must have administrator rights to install and run CodeWarrior software from this location.

Your project workspace needs to be setup in any folder that you can fully access.

1. Run the installer.



The install wizard appears.

2. Follow the wizard's on-screen instructions to install the CodeWarrior software and the desired GCC toolchains.

When installation completes, the **InstallShield Wizard Completed** page appears.

3. Click **Finish**.

The wizard closes.

4. Restart CodeWarrior and install e200 service pack. This can be installed using even online or offline mode. Check Installing service pack for CodeWarrior tools chapter for details.

Once you reached this point, you have successfully installed CodeWarrior Development Studio for QorIQ LS series - ARM V8 ISA.

### Linux OS installation

The CodeWarrior installer must run from a root account to install any required dependencies. If running from a non-root account, the installer will warn you regarding any un-fulfilled product dependencies but will continue to run. You will have to manually add the dependencies by referring to README.txt in the layout.

CodeWarrior service packs are installed with the **Eclipse Updater**. To start the Eclipse Updater select '**Help > Install new software**' in the menu.

Eclipse needs read/write access to the installation folder. Make sure the eclipse installation folder has the appropriate permissions for all users. Make sure your project workspace has read and write permissions.

1. If you have a CodeWarrior Development Studio installation CD, insert it into the Linux host computer's CD-ROM drive and mount the CD-ROM media on the Linux file system.
2. On the host computer, open a new terminal window.

A shell session starts.

3. If you have an installation CD, change the working directory to the CD-ROM mount directory.

#### NOTE

See README.txt file in the mount directory. This file contains installation instructions of different Linux distributions.

4. If you have an archive instead of an installation CD, unpack it and change the working directory to the unpacked folder.
5. If you have logged in with a non-root account, issue the command `xhost +`.
6. Issue the command `./setuplinux.sh`.

The install wizard starts and displays its welcome page.

7. Follow the wizard's on-screen instructions to install the CodeWarrior software and the desired GCC toolchains.

When the software installation completes, the wizard displays its installation summary page.

8. Click **Finish**.

9. Restart CodeWarrior and install e200 service pack. This can be installed using even online or offline mode. Check Installing service pack for CodeWarrior tools chapter for details.

The CodeWarrior software installation is now complete.

### Licensing

Both Windows and Linux installers generate an Evaluation license for the Architect edition that is valid for 15 days. The license is generated regardless of other product versions that may have been installed on the same host. The certificate generated is valid only on the machine where the product has been installed. During the installation, the user is informed that the evaluation key is node-locked and number of days the key is valid. Node-locking element is by default Ethernet ID. In case no Ethernet ID is found, the disk ID is used.

To obtain a suitable license, refer the [CodeWarrior Development Studio for QorIQ LS series for ARM V8 ISA](#) product summary page.

## 3.2 Working with bareboard application

This section explains the steps required to create, build, and debug an ARMv8 bareboard project.

### 3.2.1 Creating, building bareboard project

This topic explains how to create and build a bareboard project.

To create, build a bareboard project:

1. Launch the CodeWarrior IDE.

On Windows:

- Select **Start > All Programs > Freescale CodeWarrior > CW4NET { $\{CW\}$  Version} > CodeWarrior for ARMv8**.

The **Workspace Launcher** dialog appears.

On Linux:

- Open a new terminal window and change the working directory to:  $\{\$CWInstall\}/CW\_ARMv8/$ , where  $\{\$CWInstall\}$  is the path to your CodeWarrior installation.

- Issue the command `./fsl_eclipse.sh`

The **Workspace Launcher** dialog appears.

- a. If you wish to change the location of your project's workspace, click **Browse** to select the new path.

The **Select Workspace Directory** dialog appears.

- b. Select the required folder. Alternatively, to create a new workspace directory:

- On Windows, click **Make New Folder**.
- On Linux, click **Create Folder**.

- c. Click **OK**.

The **Select Workspace Directory** dialog closes.

- d. Click **OK** to store the project at the specified location.

The CodeWarrior IDE launches and displays the **Welcome** page.

#### NOTE

The **Welcome** page is displayed when the CodeWarrior IDE is launched for the first time. You can always return to the Welcome page by selecting **Help > Welcome** from the CodeWarrior IDE menu bar.

- e. Click **Go to the workbench**, on the Welcome page.

The workbench window appears.

2. Create a new project.

- a. From CodeWarrior IDE menu bar, select **File > New > CodeWarrior Stationery**.

- b. In **Available stationaries**, select **ARMv8 > Bareboard > Hello World C Project**.

- c. In the **Project name** text box, type `FirstProjectTest`.

**NOTE**

The **Location** text box shows the default workspace location. To change this location, deselect the **Use default location** checkbox and click **Browse** to select a new location. Ensure to append the name of your project to the end of selected path. Until you do so, the message "An existing directory cannot be specified for the project location." appears at the top of the current wizard page, and you cannot go to the next page.

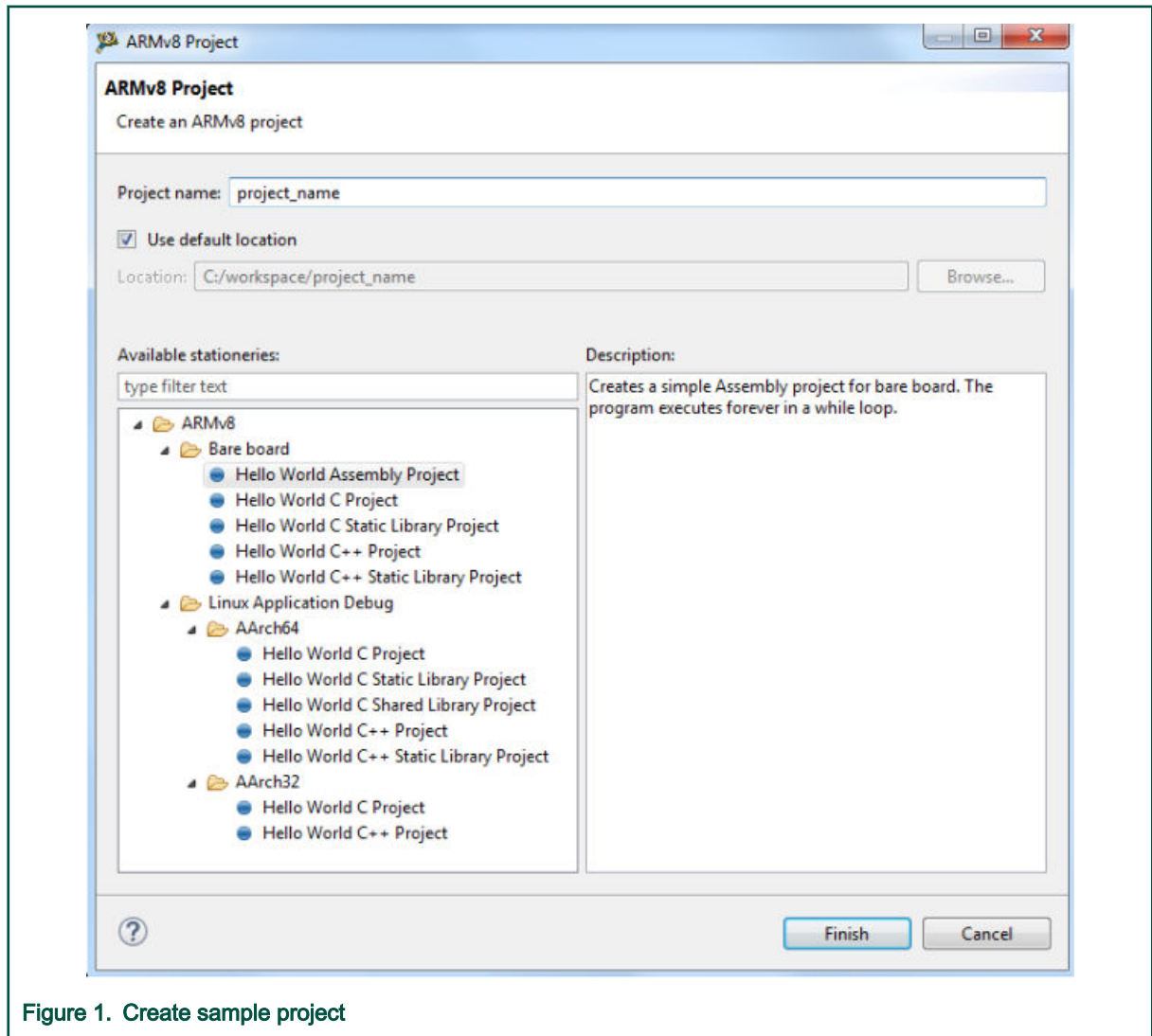


Figure 1. Create sample project

- d. Click **Finish**.

The new project appears in the **Project Explorer** view.

3. Build the project.

- a. Select the newly created project in the **Project Explorer** view.
- b. Select **Project > Build Project** to build the project. Alternatively, right-click the project in the **Project Explorer** view and select **Build Project** from the context menu that appears.

The IDE compiles the project's source code files and links resulting object code into an ELF-format executable file.

**NOTE**

You can use the `ecd.bat/ecd.sh` script to script Eclipse Workspace building from the command line. The syntax is:

```
ecd -data ${WORKSPACE} -import ${PROJECT_PATH}
- build [all | ${PROJECT_NAME}[/${PROJECT_CONFIG}]].
```

Ecd is a wrapper on top of the CDT command-line builder so it accepts all the options specific to this tool as arguments.

### 3.2.2 Debugging bareboard project

This topic explains how to debug a bareboard project.

To debug a CodeWarrior bareboard project:

1. Prepare to debug the program.
  - a. Launch the CodeWarrior IDE.
  - b. Select **Window > Show View> Other....**  
The **Show View** dialog appears.
  - c. Select the **Debug> Target Connection Configuration** option.

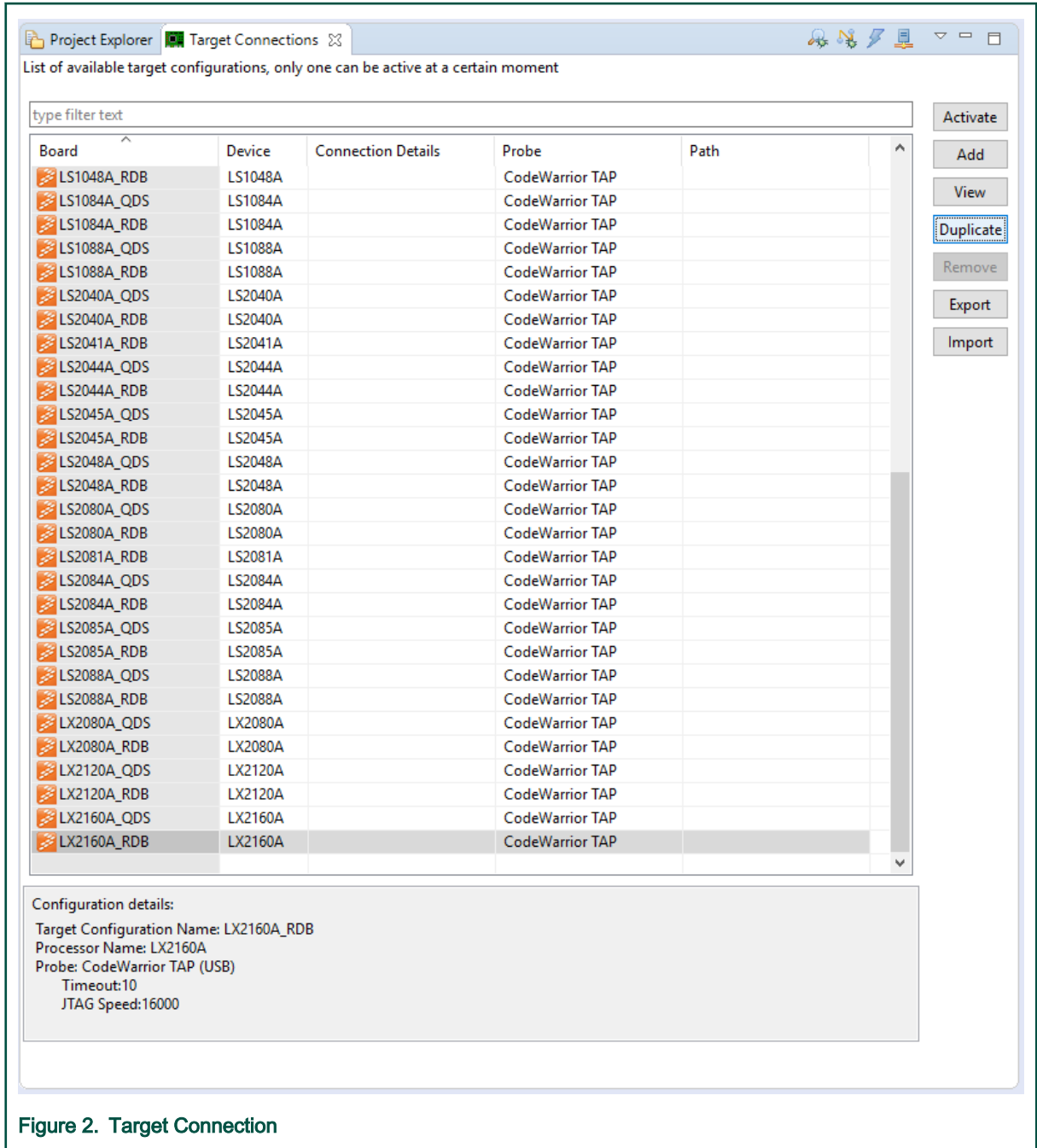


Figure 2. Target Connection

- d. Create a Target configuration.
  - Select the required target connection.
  - Click the **Duplicate** button. The **Duplicate target configuration** dialog appears.
  - Specify the name for the custom configuration.
  - Select the custom configuration and click **Edit**.
  - In the **CodeWarrior TAP Connection** section, specify the **Hostname/IP** value.

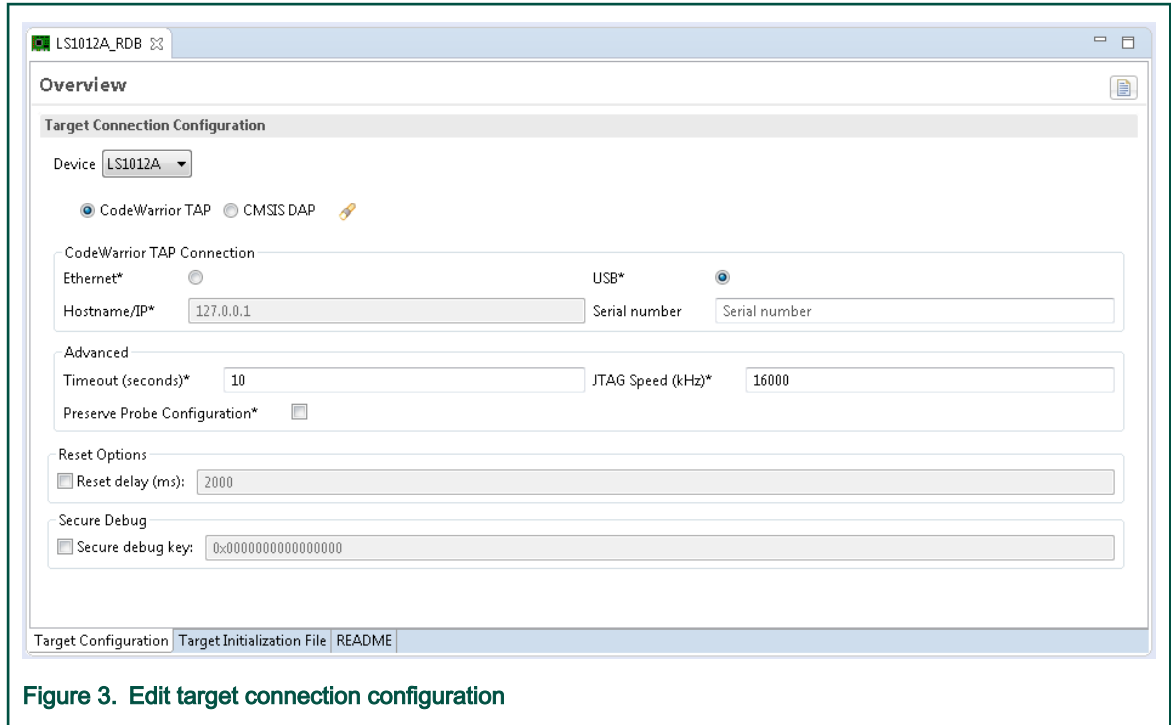


Figure 3. Edit target connection configuration

- e. Select **File > Save**.
- f. Select **Run > Debug Configurations**.
- g. Expand Select **GDB Hardware Debugging**.
- h. Select the launch configuration with the name `${ProjectName}`.
- i. Click the **Debugger** tab.
- j. Select the core to debug (the default value should be "CortexA57#0").
- k. Click the **Startup** tab.
- l. If you do not want to reset the board after running the init script, ensure that the **Reset and Delay (seconds)** and **Halt** checkboxes are deselected.
- m. Click **Apply** in the **Debug Configurations** dialog.

The IDE saves your settings.

2. Debug the program.

- a. Click **Debug**.
- b. The IDE switches to the **Debug** perspective. The debugger downloads your program to the target board and halts execution at the first statement of `main()`.
- c. Click a thread in the **Debug** view.

The program counter icon



(on the marker bar) points to the next statement to be executed.

- d. In the **Debug** view, click **Step Over**



The debugger executes the current statement and halts at the next statement.

3. Set breakpoint and execute the program up to the breakpoint.

- a. In the editor area, scroll to a statement.
- b. Double-click the marker bar next to the statement.

A breakpoint indicator



appears next to the statement.

- c. In the **Debug** view, click **Resume**



.

The debugger executes all statements up to, but not including the breakpoint statement.

4. Control the program:

- a. In the **Debug** view (top-left of perspective), click **Step Over**



.

The debugger executes the current statement and halts at the next statement.

- b. In the **Debug** view, click **Resume**



.

- c. In the **Debug** view, click **Terminate**



.

The program terminates and the debug session ends.

5. Select **File > Exit**.

The CodeWarrior IDE window closes.

### 3.3 Working with Linux application

This section explains the steps required to create, build, and debug a Linux application project.

#### 3.3.1 Creating simple Linux application project

This topic explains steps to create a ARMv8 Linux application project.

To create a new ARMv8 Linux application project:

1. Open CodeWarrior Development Studio for QorIQ LS series - ARM V8 ISA.
2. Select **File > New > ARMv8 Stationery > Linux Application Debug > Hello World C Project**.
3. Specify a project name.
4. Click **Finish**.
5. Select the newly created Linux application project in the **Project Explorer** view.
6. Select **Project > Build project**.

To create a ARMv8 Linux application project using an existing Linux application image, see [#unique\\_17](#).

### 3.3.2 Updating remote connection

This topic explains how to change the settings in a default remote connection.

The IP/hostname and the SSH port of the Linux target must be set to the correct values.

To change the default values perform the following steps:

1. Click **Run > Debug Configurations**.

The **Debug Configurations** dialog appears.

2. Select the Linux application project in the left-hand panel, under **C/C++ Remote Application**.

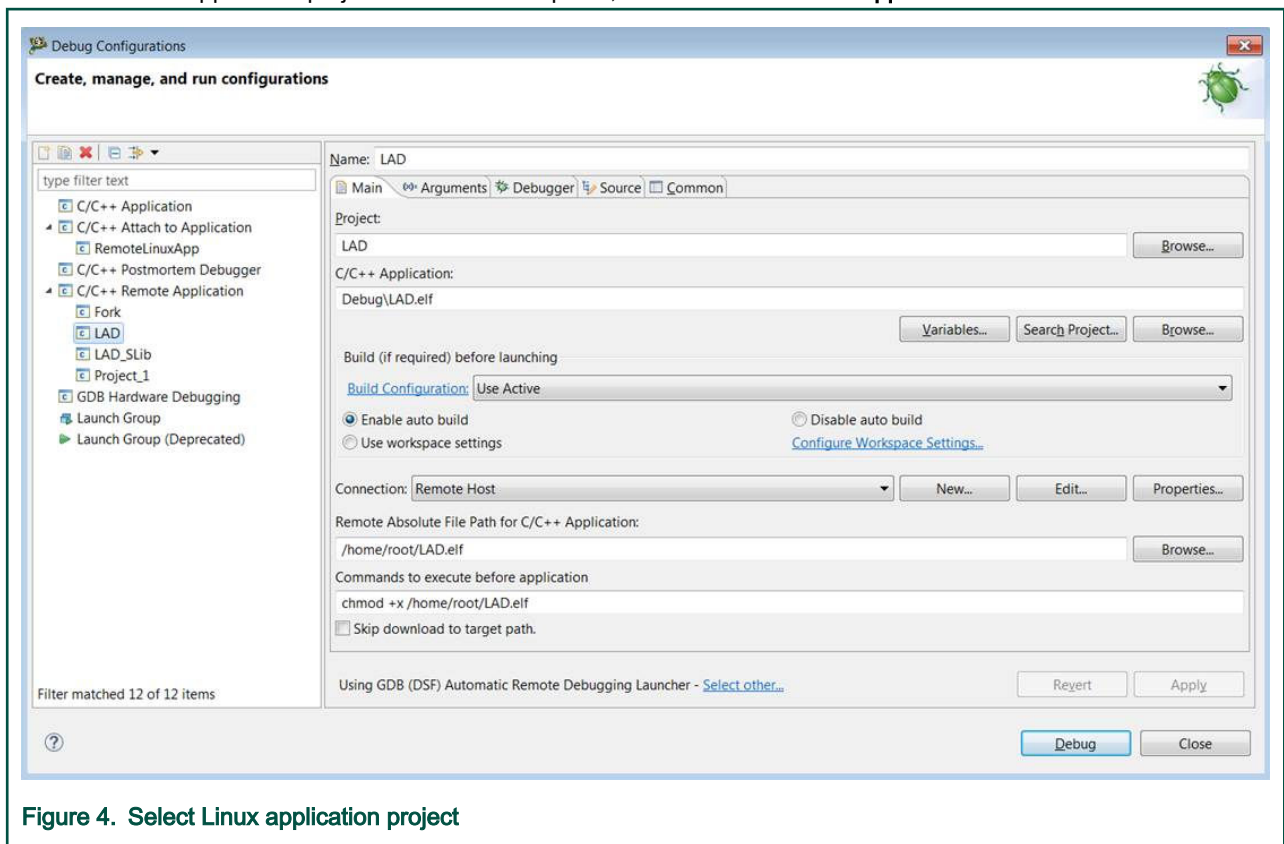


Figure 4. Select Linux application project

3. Click the **Edit** button right next to the default **Remote Host** connection. The **Edit Connection** dialog appears.



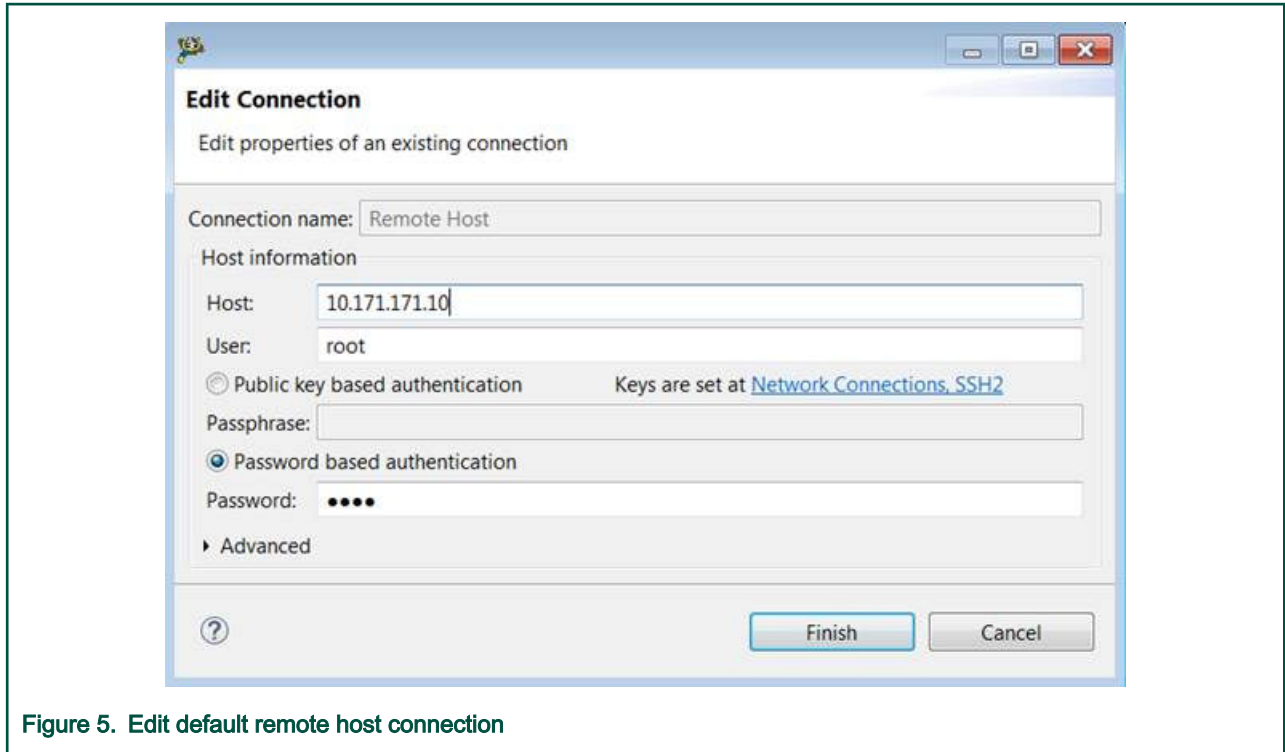


Figure 5. Edit default remote host connection

4. Specify the IP of the Linux target in the **Host** text box, specify the user name in the **User** text box and specify the password in the **Password** text box.
5. Expand the **Advanced** section and configure the SSH port and other connection settings, as needed.

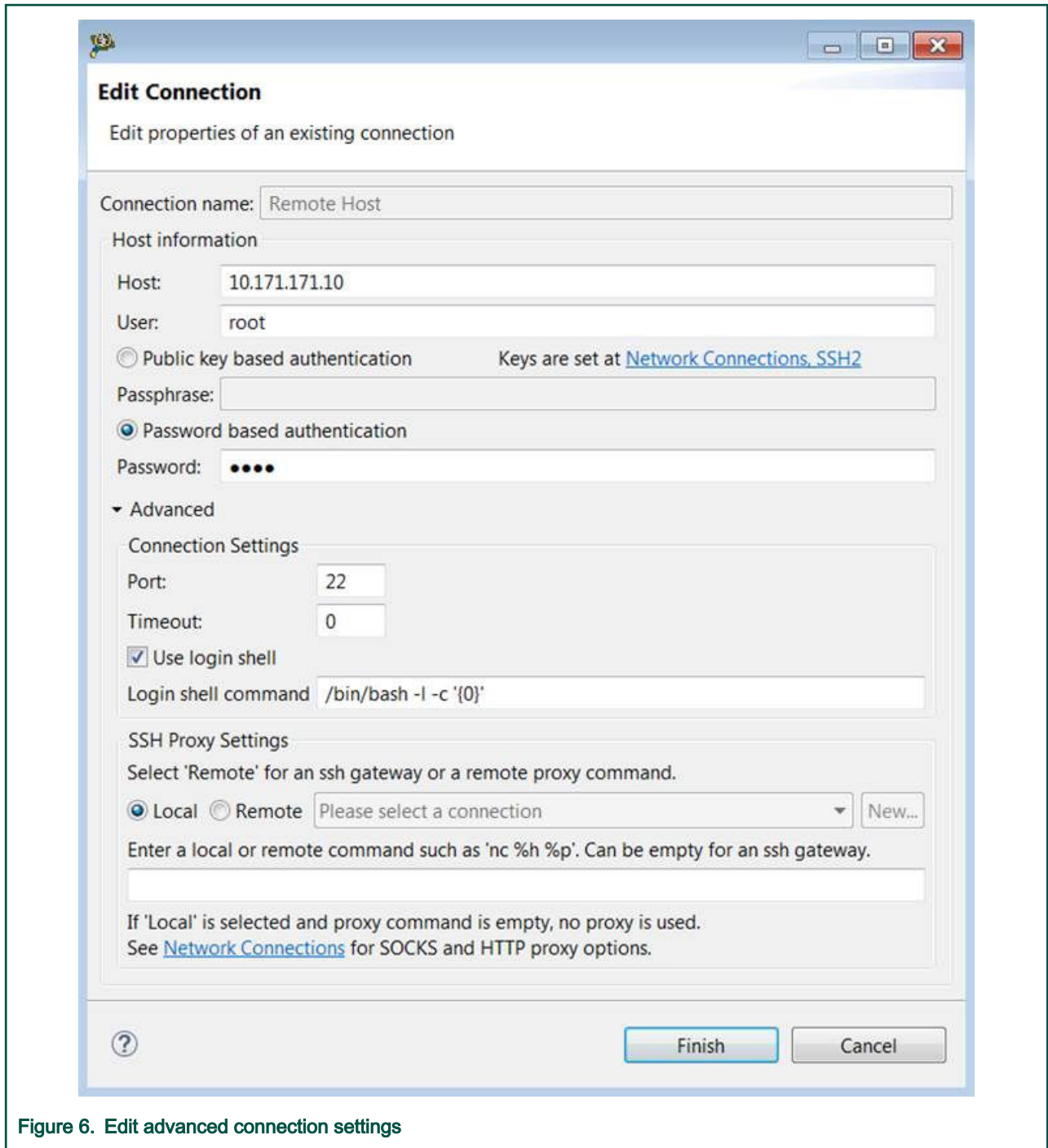


Figure 6. Edit advanced connection settings

- Click **Finish**.

### 3.3.3 Using sysroot

This section is required only if you want to enable full debug support (inside target system libraries) for the Linux application project.

#### NOTE

Before you proceed, ensure that you have completed all the steps in [Updating remote connection](#).

To enable full debug support for a Linux application project, perform these steps:

1. GDB should be configured to use the target system libraries.
  - a. On the host PC, create a folder `rootfs` and a sub-directory `lib`.
  - b. Copy the following libraries: `libc`, `ld`, `libpthread` into to the `rootfs/lib/` folder. Use the full library name as you see it on the target, for example `libpthread.so.0`, `ld-linux-aarch64.so.1`, `libc.so.6`.
  - c. Create a `*.gdbinit` file on the file system. For example, `test.gdbinit`
  - d. Add following content in the `.gdbinit` file:

```
set sysroot <host_path_to_rootfs>
```

For example, set `sysroot` `C:\Users\u12345\Desktop\rootfs`

#### NOTE

If you are running the CodeWarrior software on the same Linux machine where you have compiled the SDK package, you can directly set up the `sysroot` from that location in the `gdbinit` file. For example, the path to `sysroot` on a yocto-based sdk would be:

```
set sysroot /home/u12345/Desktop/SDK_Setup/QorIQ-SDK-V2.0-20160527-yocto/  
build_ls2088ardb/tmp/sysroots/ls2088ardb/
```

2. Add missing settings in launch configuration file.
  - a. Right-click the project and select **Debug As > Debug Configurations**.  
The **Debug Configurations** dialog appears.
  - b. Expand **C/C++ Remote Application**, select the launch configuration for the Linux application project you want to debug.
  - c. Click the **Main** sub tab in the **Debugger** tab.
  - d. Browse to `*.gdbinit` path in **GDB command file** field.

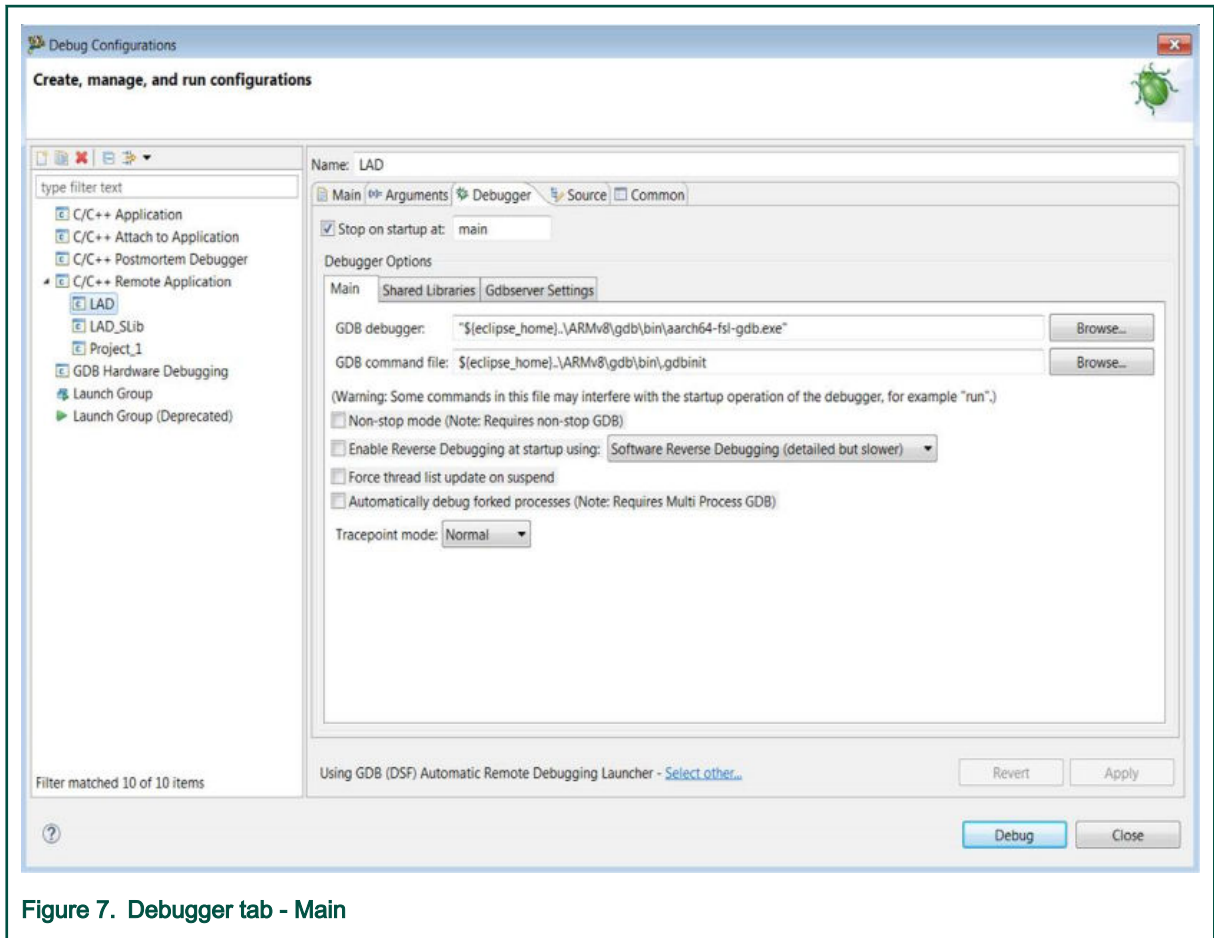


Figure 7. Debugger tab - Main

- e. Click **Apply**.

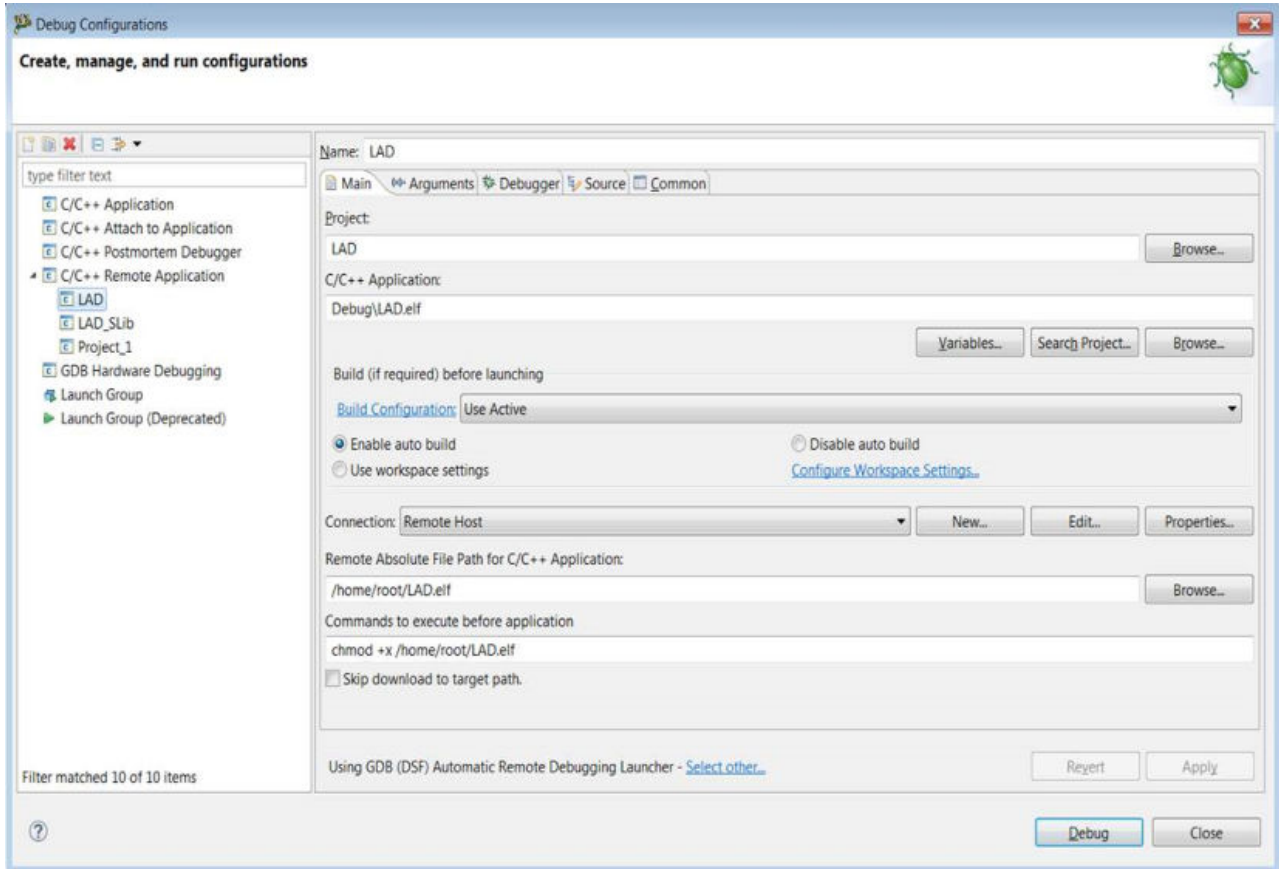
### 3.3.4 Debugging Linux application project

This topic explains steps to debug a Linux application project.

To debug a Linux application project:

1. From the CodeWarrior IDE menu bar, select **Run > Debug Configurations**.

- In the **Debug Configuration** dialog, expand **C/C++ Remote Application** and select the launch configuration for the Linux application project you want to debug.



- Click **Debug**.

### ***How To Reach Us***

#### **Home Page:**

[nxp.com](http://nxp.com)

#### **Web Support:**

[nxp.com/support](http://nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, Freescale, the Freescale logo, and QorIQ are trademarks of are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, Cortex are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

© NXP B.V. 2020.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 06/2020

Document identifier: CWARMv8GS