

Yocto ADT Plugins in CodeWarrior for ARMv8

1 Introduction

Being able to build a complete software stack is great, but there is also a need to develop software against the stack for being able to compile, run, debug, and profile the software stack as needed. The objective of the Application Development Toolkit (ADT) is to provide developers a way to do this without the need to learn the build system. Ultimately, this lets the developers focus on what they do best - develop.

This Application Note is depicting use of the Yocto ADT Plugins bundled with the CodeWarrior software for creating, building, and debugging a Linux application on the LS2085A EAR SDK.

This Application Note explains:

- How you can setup Yocto sdk for ADT support
- How you can configure Yocto ADT in CodeWarrior
- How you can make a CodeWarrior project Linux application using ADT

2 Setup Yocto SDK for ADT support

To enable the ADT support:

Contents

1	Introduction.....	1
2	Setup Yocto SDK for ADT support.....	1
3	Setup Yocto ADT in the CodeWarrior software.....	2
4	Create and setup a Linux application project using ADT.....	3
4.1	Create the project.....	3
4.2	Setup the project.....	5

Setup Yocto ADT in the CodeWarrior software

1. meta-fsl-toolchain defines gdb preferred version 7.7.1 only for gdb-cross-aarch64, while gdb-cross use the standard (old) gdb 7.6.1. In order to use the preferred fsl 7.7.1 for all other packages (for example meta-ide-support) that use gdb-cross, rename the preferred fsl gdb from gdb-cross-aarch64 to gdb-cross
 - a. vim meta-fsl-toolchain/conf/include/fsl-toolchain-default-version.inc
 - b. Update line 5 to:

```
PREFERRED_VERSION_gdb-cross = "${GDBVERSION}"
```

- c. vim meta-fsl-toolchain/recipes-devtools/gdb/gdb-cross.inc
 - d. Update line 11 to:

```
PN = "gdb-cross"
```

2. Enable the ADT settings in Yocto SDK.
 - a. bitbake meta-ide-support

3 Setup Yocto ADT in the CodeWarrior software

To setup Yocto ADT in the CodeWarrior software:

1. Select **Window > Preferences > Yocto Project ADT**.
2. Browse the **Toolchain Root Location** (build directory) and **Sysroot Location**.

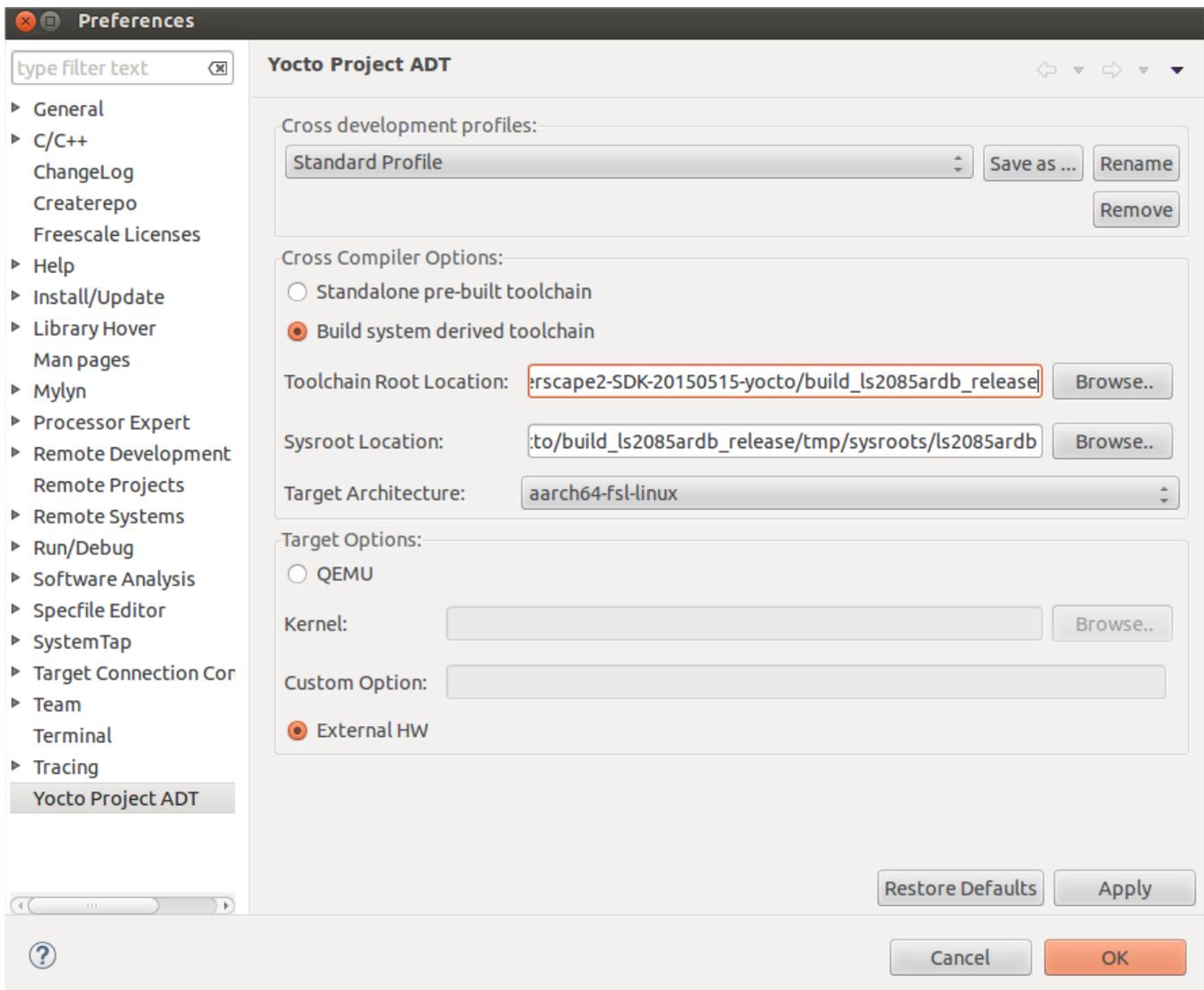


Figure 1. Toolchain root and Sysroot location for LS2085RDB board

3. Click OK.

4 Create and setup a Linux application project using ADT

This section explains:

- [Create the project](#)
- [Setup the project](#)

4.1 Create the project

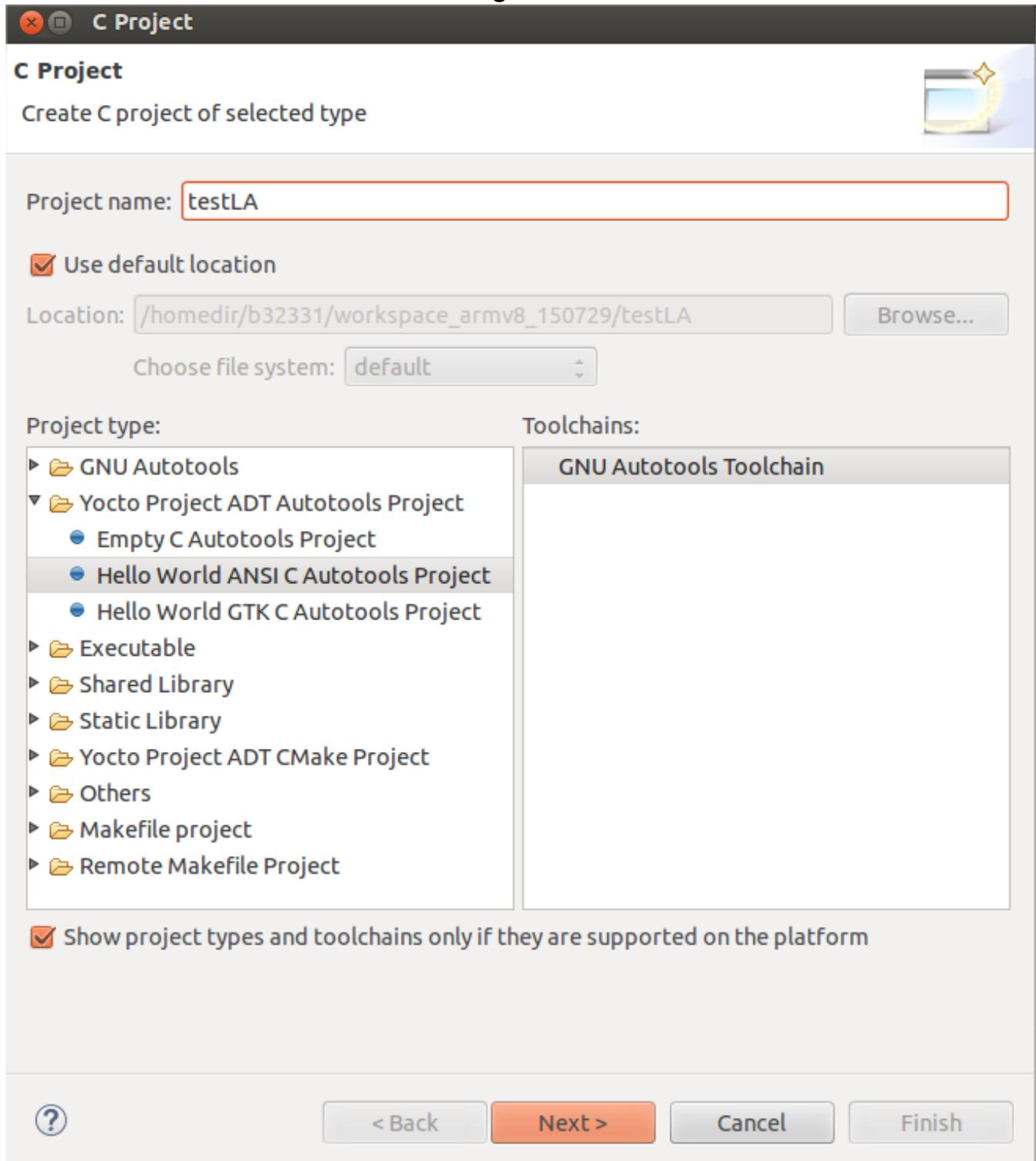
To create a Linux application project using ADT:

1. Select **File > New > C Project**. The **C Project** dialog appears.

Create and setup a Linux application project using ADT

2. Select **Yocto Project ADT Autotools Project > Hello World ANSI C Autotools Project** and click **Next**.

Figure 2.



3. Specify the **Author** in the Basic Settings page and click **Finish**.
4. Right-click the project and select **Build Project**.

4.2 Setup the project

After the project is created, in debug launch the ADT will automatically set **Legacy Remote Create Process Launcher**, which is an old version with smaller feature set, for example OS Resources is not supported.

You can go with legacy process launcher as explained in step 1 or upgrade to use GDB Auto Remote Debugging Launcher as explained in step 2 , but the Debugger tab needs to be updated manually.

1. Go with the Legacy process Launcher:
 - a. In the **Main** tab, select **ScpConnection** from the **Connection** drop-down list. RSE is already created by CodeWarrior. You can edit RSE using the Edit button from left side for editing the IP target Linux). Also, ensure that the `ttypass` field from **ConnectorServices > SSH Settings** is set up to 2.

The screenshot shows the ADT interface. The top window is titled "Properties for ScpConnection". It has a left sidebar with "Connector Services" and "Host" options. The main area is titled "Host" and contains the following fields:

- Resource type: Connection to remote system
- Parent profile: fsr-ub1264-121
- System type: SSH with SCP
- Host name: 192.168.1.2
- Connection name: ScpConnection
- Default User ID: root
- Description: (empty field)

Below this is the "Connector Services" tree view. It has a search filter "type filter text". The tree shows "Connector Services" expanded to "Host". Under "Host", "Available Services" is expanded to "SSH Connector Service", which is further expanded to "SSH Settings".

Below the tree view is a "Properties" table:

Property	Value
keepalive (sec)	300
timeout (sec)	0
ttypass (sec)	2

Create and setup a Linux application project using ADT

- b. In the **Main** tab, update the remote paths and commands as shown below, and click **Debug**.

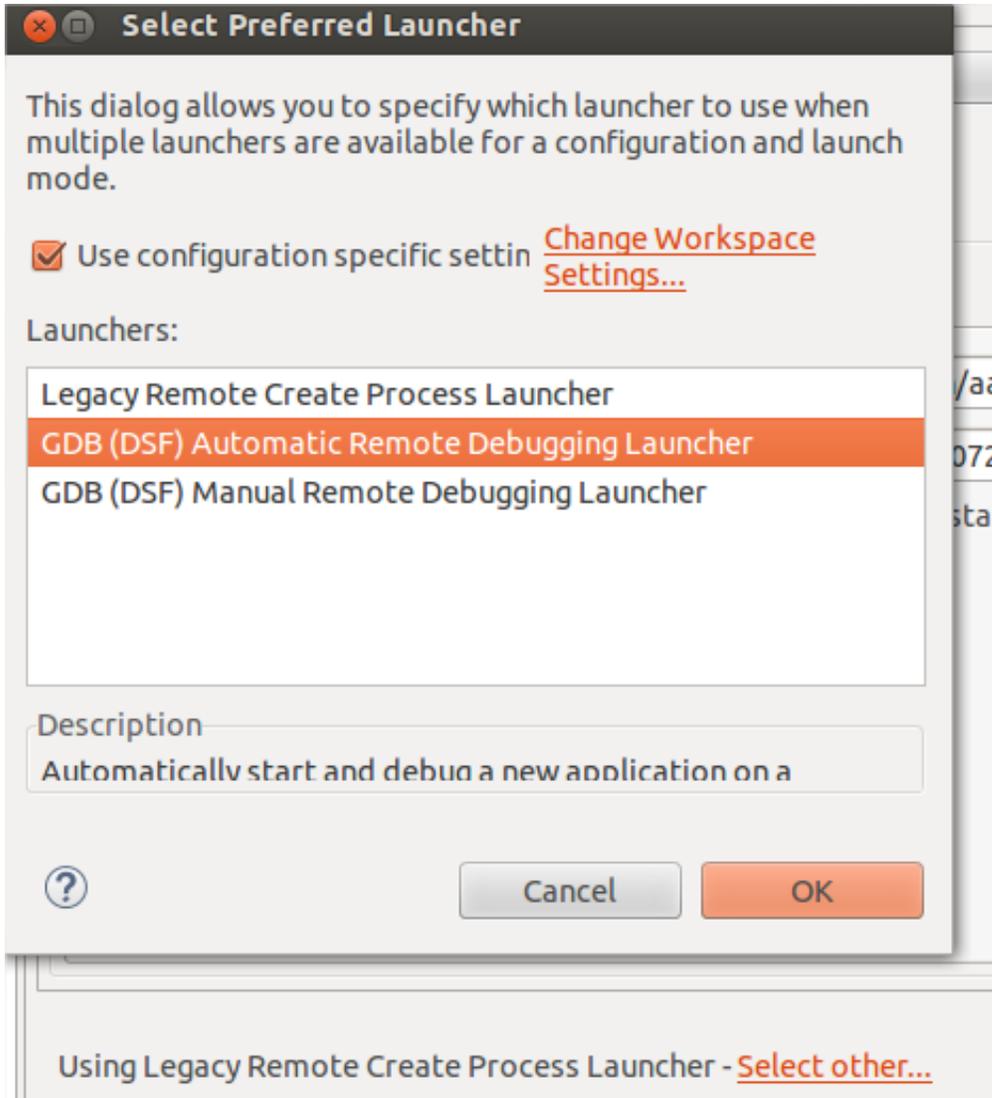
The screenshot shows the 'Main' tab configuration window for a project named 'testLA_gdb_aarch64-fsl-linux'. The configuration includes:

- Connection:** ScpConnection (with buttons for New..., Edit..., and Properties...)
- Project:** testLA (with a Browse... button)
- Build configuration:** Use Active (with a dropdown arrow)
- C/C++ Application:** src/testLA (with buttons for Variables..., Search Project..., and Browse...)
- Remote Absolute File Path for C/C++ Application:** /home/root/testLA (with a Browse... button)
- Commands to execute before application:** chmod +x /home/root/testLA
- Skip download to target path.
- Using Legacy Remote Create Process Launcher - [Select other...](#)** (with Apply and Revert buttons)

At the bottom right, there are buttons for **Close** and **Debug**.

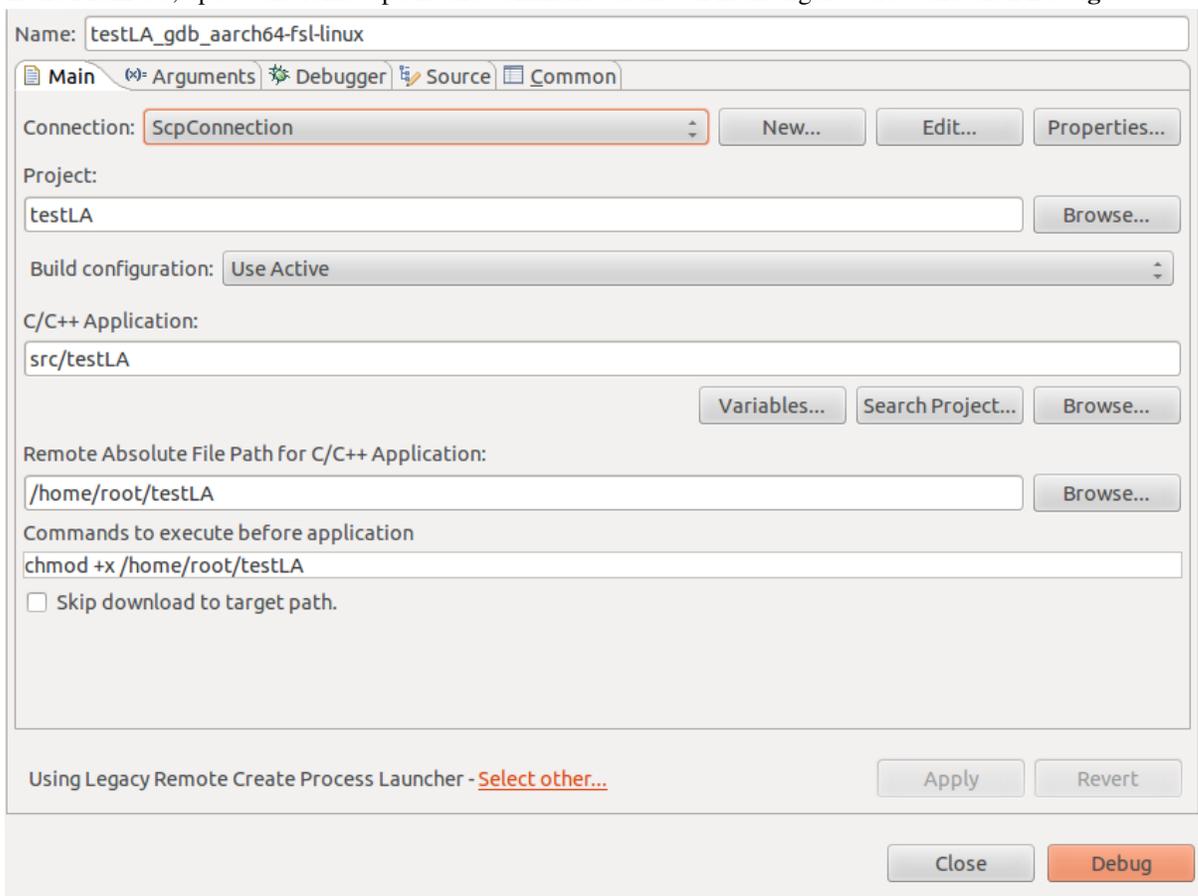
2. Before updating the Process Launcher, save the GDB debugger and GDB command file paths in the **Debugger** tab. These paths will be lost after switching to GDB Auto Remote Debugging Launcher.

- a. Click the **Select other** link and select GDB (DSF) Automatic Remote Debugging Launcher.

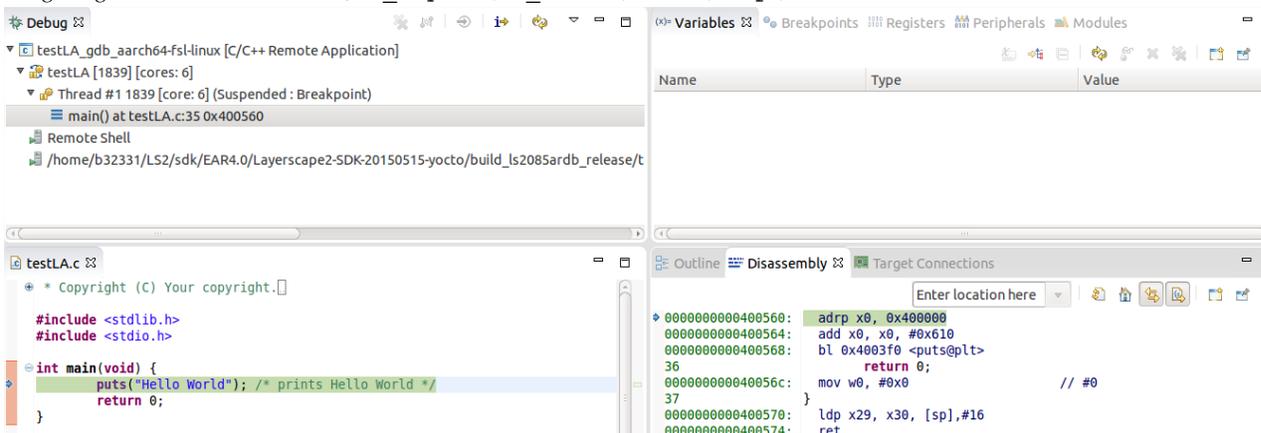


- b. Update the GDB debugger and GDB command file paths in the **Debugger** tab with the ones saved previously.

- c. In the **Main** tab, update the remote paths and commands as shown in the figure below and click **Debug**.



Now, you are in debug with a Linux Application Project set up by the ADT CodeWarrior plugin. For more details about Linux Application Debug capabilities in the CodeWarrior software, refer the "Use Cases" chapter in the *ARMv8 Targeting Manual* available at: `$CW_Layout\CW_ARMv8\ARMv8\Help\PDF`



How to Reach Us:

Home Page:

freescale.com

Web Support:

freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale 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. Freescale reserves the right to make changes without further notice to any products herein.

Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, CodeWarrior, and QorIQ are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Layerscape is trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. ARM, Cortex, Cortex-A53, Cortex-A57, and TrustZone are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2015, Freescale Semiconductor, Inc.