

# **ARM DS-5 Tools and Avnet ZED Series**

**#5**

## **Debug a Linux Application using DS-5 and Avnet ZedBoard or MicroZed**



Sept 2013  
Version 01

## Table of Contents

ARM DS-5 Tools and Avnet ZED Series .....	3
Required Installations .....	4
Technical Support .....	5
Debugging a Linux Application.....	6
Configure a Remote System Explorer (RSE) Connection .....	7
Import the Hello World Linux Example Project into DS-5 .....	11
Debug Hello World on Linux .....	16
Revision History .....	18
Resources .....	18

## ARM DS-5 Tools and Avnet ZED Series

This tutorial is one in a series of step by step instruction manuals. Together they document the procedures necessary to utilize the ARM Development Studio 5 (DS-5™) Software Suite and the DSTREAM Debugging tools with the Avnet Zynq Evaluation and Development (ZED) boards. These tutorials can be used on their own, or in combination with Avnet online videos and OnRamp Technical Session™.

The ARM software and hardware tools provide a powerful debugging suite for processor-based systems built around the dual Cortex-A9 cores present in the Xilinx Zynq SoC, at the heart of the Avnet ZED boards. A Linux software developer can simultaneously debug applications and kernel module code, with separate control over each thread. You can step through Linux boot code, first stage bare metal boot code, and bare metal applications. When used in concert with the Xilinx Vivado tools for FPGA fabric development, the ARM debugger and Internal Logic Analyzer (ILA) IP can be cross-triggered to stop on software and hardware breakpoints, or when a hardware event occurs. For difficult-to-isolate intermittent faults, DS-5 provides access to the Cortex-A9 on-chip Trace facility. Once your embedded system is running correctly, DS-5 uses Streamline, a graphical system profiler, to identify performance bottlenecks in your design to ensure top-shelf operation.

This tutorial series begins with the most basic tool configuration and board connection. It takes you all the way through to the most complex aspects of hardware/software co-debugging to root out design errors that are otherwise apparent only in very complex use cases, or worse, after a product is released. Together the ARM DS-5 tools, Xilinx Vivado and Avnet ZED boards provide an unparalleled combination to compress design timelines, cut project costs and optimize your product for the marketplace.

## Required Installations

### Software

The recommended software for this tutorial series is:

- ARM Development Studio 5 (Exact version used is 5.14, build 1702)
- Xilinx ISE WebPACK 14.5 (Free license and download from Xilinx website)
- Cypress CY7C64225 USB-to-UART Bridge Driver (for ZedBoard serial output)
- Silicon Labs CP2104 USB-to-UART Bridge Driver (for MicroZed serial output)
- Tera Term (Exact version used is V4.75)
- Xilinx Software Development Kit, version 14.5
- For hardware/software co-debugging, Xilinx Vivado 2013.2

### Hardware

The targeted hardware consists of the following:

- PC workstation with at least 5 GB RAM, 30GB free hard disk space, Windows 7 64-bit operating system, and a wired GB Ethernet connection
- Available SD card slot on PC or external USB-based SD card reader
- One of:
  - Avnet ZedBoard Kit (**AES-Z7EV-7Z020-G**)
    - USB cable (Type A to Micro-USB Type B)
    - 4GB SD card
    - 12v Power supply
  - Avnet MicroZed Kit (**AES-Z7MB-7Z010-G**)
    - USB cable (Type A to Micro-USB Type B)
    - 4GB SD card
- Avnet ZedBoard Debug Adapter Kit (**AES-ZBDB-ADPT-G**)
  - 14-pin Xilinx PC4 ribbon cable
- ARM DSTREAM unit and Keil pod with wide cable connector
  - 20-pin JTAG ribbon cable
  - USB cable (Type A to Printer)
  - 5v Power supply
- CAT-5 Ethernet cable

## Technical Support

For technical support with any of the instructions, please contact your local Avnet/Silica FAE or visit the support forums:

<http://www.zedboard.org/forum>

<http://www.microzed.org/forum>

Additional technical support resources are listed below.

*ZedBoard Kit/MicroZed Kit support page with Documentation and Reference Designs*

<http://www.zedboard.org/content/support>

<http://www.microzed.org/content/support>

For Xilinx technical support, you may contact your local Avnet/Silica FAE or Xilinx Online Technical Support at [www.support.xilinx.com](http://www.support.xilinx.com) . On this site you will also find the following resources for assistance:

- Software, IP, and Documentation Updates
- Access to Technical Support Web Tools
- Searchable Answer Database with Over 4,000 Solutions
- User Forums
- Training - Select instructor-led classes and recorded e-learning options

Contact your Avnet/Silica FAE or Avnet Support for any additional questions regarding the reference designs, kit hardware, or if you are interested in designing any of the kit devices into your next design.

<http://www.em.avnet.com/techsupport>

For ARM technical support, you may contact your local Avnet/Silica FAE or ARM Online Technical Support at [www.arm.com/support](http://www.arm.com/support) .

## Debugging a Linux Application

In this tutorial we will boot the Avnet ZED target with a standard Linux kernel that was generated in the Avnet Linux for ZedBoard Speedway course, and then download and debug an application. The DSTREAM hardware and Avnet Debug Adapter board are not required.

You will need SD card files to complete these instructions. The files are part of the download package that included this tutorial. Browse to the location on your host where the download package was decompressed, and look for the following folder:

**<Download package folder>\Support05**

For the purposes of this tutorial we will assume the files exist in folder:

**C:\OnRamps\Support05**

### For ZedBoard:

To begin the procedure, set the ZedBoard Boot Mode to SD boot using jumpers JP11 to JP7 set to the following:

	<b>JP11</b>	<b>JP10</b>	<b>JP9</b>	<b>JP8</b>	<b>JP7</b>
Position	SIG-GND	3V3-SIG	3V3-SIG	SIG-GND	SIG-GND

### For MicroZed:

To begin the procedure, set the MicroZed Boot Mode to JTAG only using jumpers JP3 to JP1 set to the following:

	<b>JP3</b>	<b>JP2</b>	<b>JP1</b>
Position	2-3	2-3	1-2

Connect an Ethernet cable between the target and your host computer. The IP Address of your host must be on the same subnet as target address **192.168.1.10**.

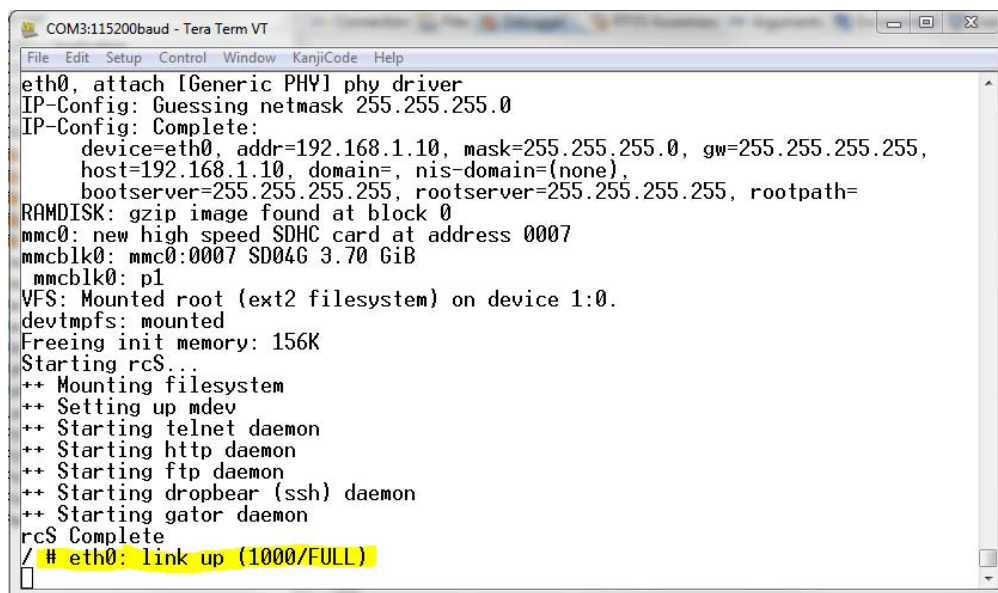
To prepare to boot the target, copy all the files in the following folder to the root folder of your SD card:

**For ZedBoard: C:\OnRamps\Support05\ZedBoard**

**For MicroZed: C:\OnRamps\Support05\MicroZed**

**For Both: C:\OnRamps\Support05\Common**

Insert the card into the SD card slot on the underside of the target. Power the target and start a Tera Term session on your host using the COM port assigned to the UART bridge. You will see the target boot into Linux on the Tera Term console and should also see an indication that the Ethernet connection to the host has been detected.



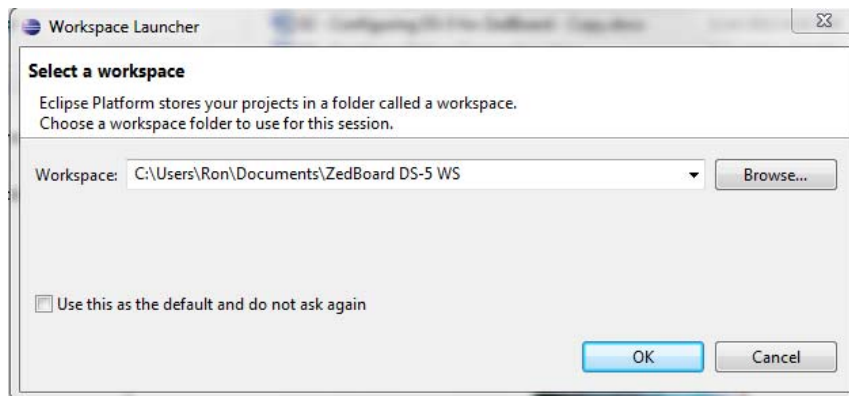
```
COM3:115200baud - Tera Term VT
File Edit Setup Control Window KanjiCode Help
eth0, attach [Generic PHY] phy driver
IP-Config: Guessing netmask 255.255.255.0
IP-Config: Complete:
    device=eth0, addr=192.168.1.10, mask=255.255.255.0, gw=255.255.255.255,
    host=192.168.1.10, domain=, nis-domain=(none),
    bootserver=255.255.255.255, rootserver=255.255.255.255, rootpath=
RAMDISK: gzip image found at block 0
mmc0: new high speed SDHC card at address 0007
mmcblk0: mmc0:0007 SD04G 3.70 GiB
    mmcblk0: p1
VFS: Mounted root (ext2 filesystem) on device 1:0.
devtmpfs: mounted
Freeing init memory: 156K
Starting rcS...
++ Mounting filesystem
++ Setting up mdev
++ Starting telnet daemon
++ Starting http daemon
++ Starting ftp daemon
++ Starting dropbear (ssh) daemon
++ Starting gator daemon
rcS Complete
/ # eth0: link up (1000/FULL)
```

### Linux Boot on ZedBoard in Tera Term

## Configure a Remote System Explorer (RSE) Connection

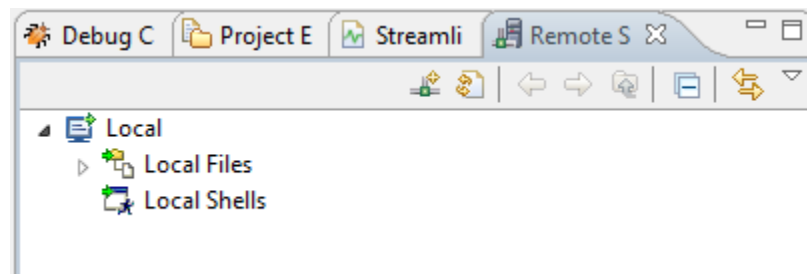
Debug communication with the ZED target will be done over an Ethernet cable between the target and your host. Before we can do this we must inform DS-5 how to configure the connection.

1. Open the DS-5 IDE on your host PC. For the purposes of this tutorial series, we will continue to use the same workspace created in an earlier lesson, named **ZedBoard DS-5 WS**.



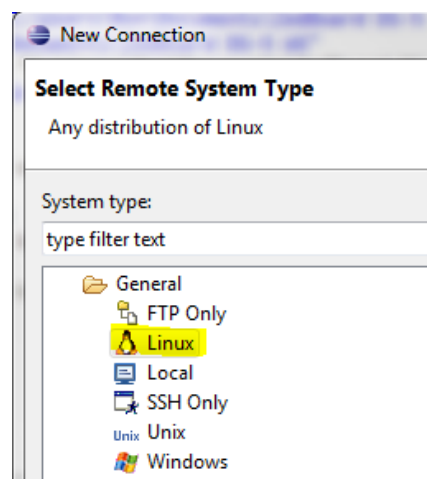
### Select a DS-5 Workspace

2. If you do not have a Remote System tab in the panel with Project Explorer, from the main menu select **Window | Show View | Other | Remote Systems | Remote Systems**. Select the **Remote Systems** tab.



### DS-5 Remote Systems Tab

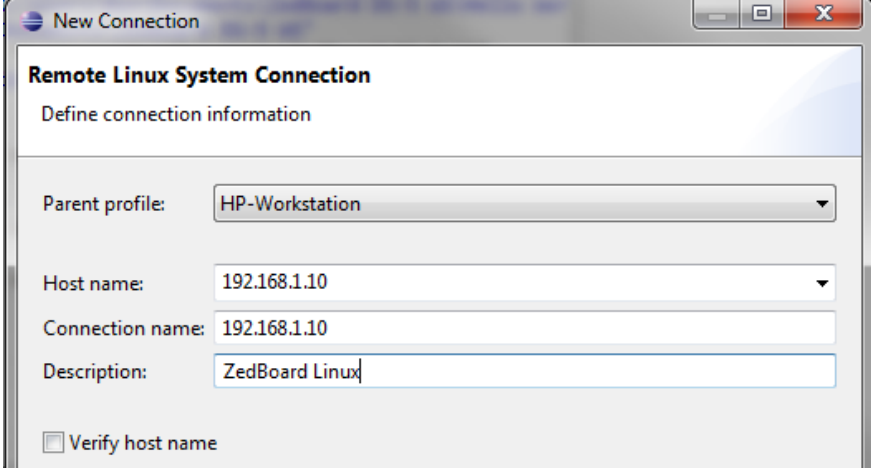
3. Anywhere inside the panel, right click and select **New Connection**. In the **New Connection** window, select **Linux** and click the **Next** button.



### Create a Remote System Connection for Linux



4. For the host name, enter the ZED target IP address of **192.168.1.10**. This will be copied into the Connection name field. You can optionally enter a description if you wish. The **Parent Profile** name will be dependent on your host platform, but leave it unchanged. Click the **Next** button.

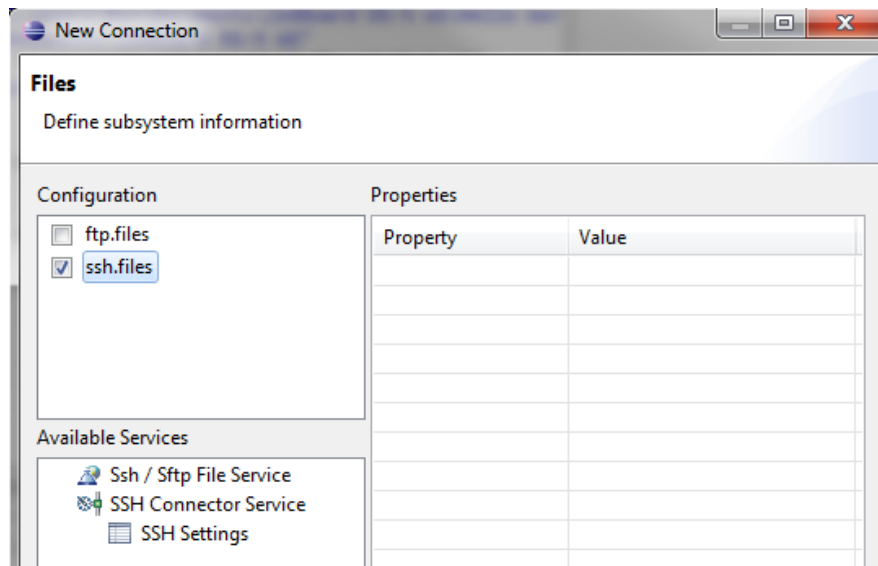


The screenshot shows a window titled "New Connection" with a subtitle "Remote Linux System Connection". Below the subtitle is the instruction "Define connection information". The form contains the following fields:

- Parent profile:** A dropdown menu with "HP-Workstation" selected.
- Host name:** A text field containing "192.168.1.10".
- Connection name:** A text field containing "192.168.1.10".
- Description:** A text field containing "ZedBoard Linux".
- Verify host name:** An unchecked checkbox.

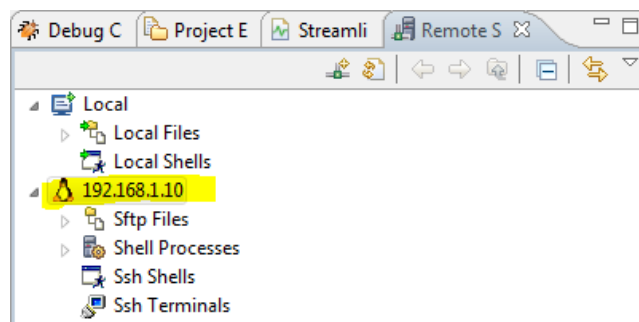
#### RSE Connection Parameters for ZED Target

5. Select the **ssh.files** configuration (Secure Shell) and click the **Finish** button.



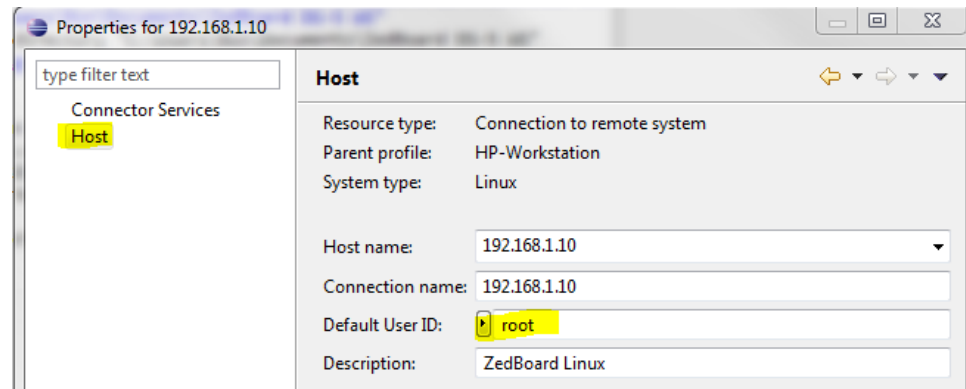
### Secure Shell Configuration

6. In the Remote Systems tab, right-click on the new entry for the ZED target IP address and select Properties.



### New Remote System Configuration for ZED Target

7. Select the **Host** entry and click the arrow next to the **Default User ID**. Enter **root** for the ID and click the **OK** button.

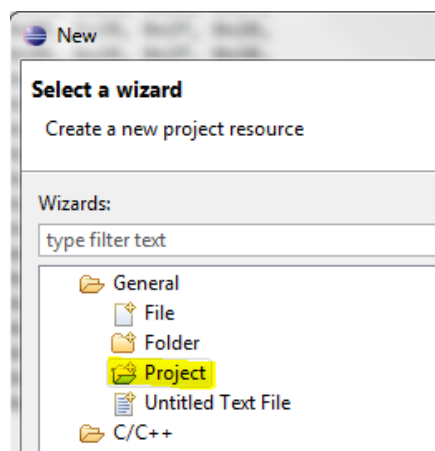


### Create root Login for ZED Target

Now the Remote Systems configuration for the ZED target is complete, we can use this in the future to establish a debug connection for any Linux application project in DS-5.

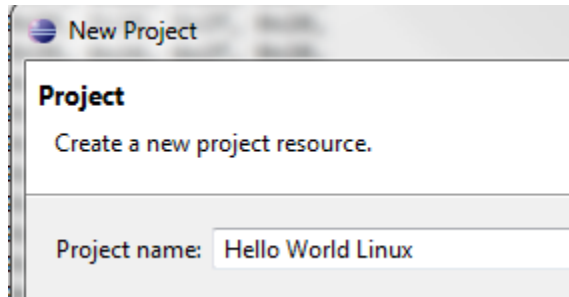
## Import the Hello World Linux Example Project into DS-5

1. First, we will create a new, empty project as the container for our files. In the C/C++ Perspective, right-click in the **Project Explorer** window and select **New | Other**.
2. Expand the **General** entry, and select **Project** from the list. Click the **Next** button.



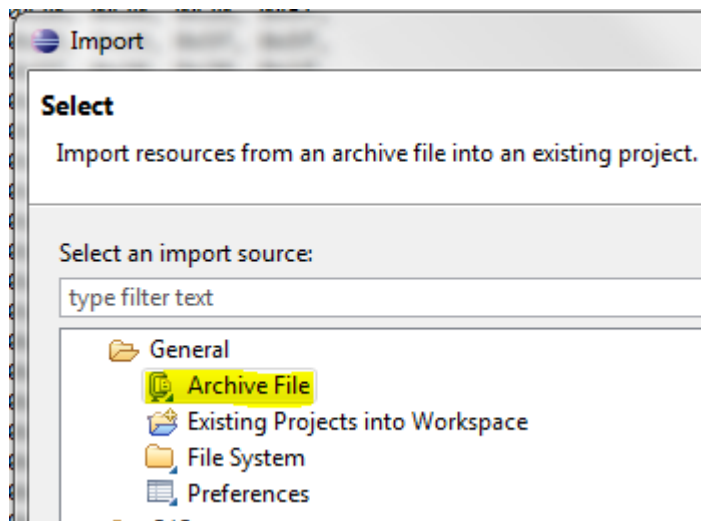
### Create a New DS-5 Project

3. Enter **Hello World Linux** for the Project Name and click the **Finish** button. The new project will appear in the Project Explorer window.



#### Name the DS-5 Project

4. Right-click on **Hello World Linux** in the Project Explorer window and select **Import** from the drop-down menu. Expand the **General** folder and select **Archive File**. Click the **Next** button to continue.

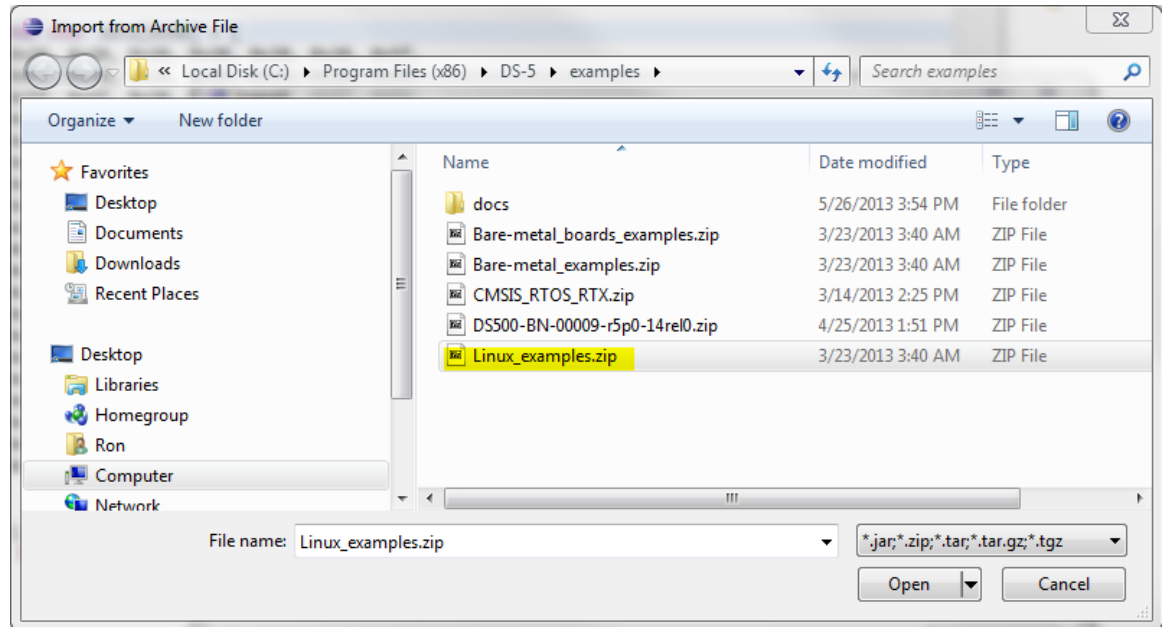


#### Import Project from Archive

5. Browse to the Linux examples archive file that was loaded as part of your DS-5 installation. If you accepted the default installation location, the file will be located at:

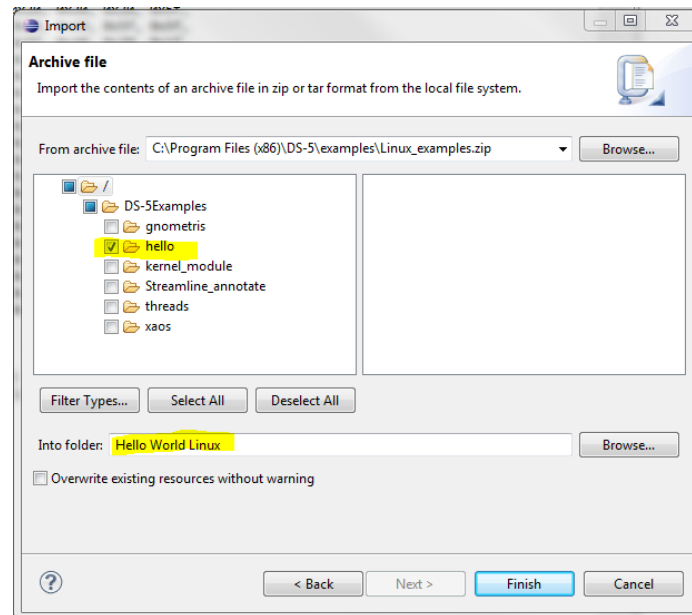
**C:\Program Files (x86)\DS-5\examples**

Select the **Linux\_examples.zip** file and click the Open button.



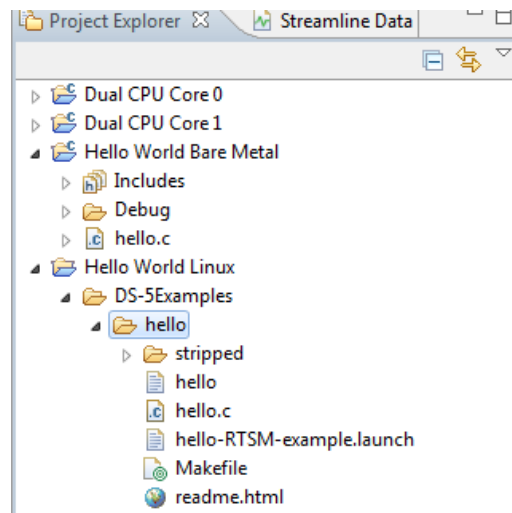
**DS-5 Linux Examples Archive**

6. In the Import window, expand the entries until you can see all the DS-5 Examples as shown below. Click the **Deselect All** button to clear all the checkboxes, and then select only the **hello** project. **Hello World Linux** will already be shown as the target folder for the import. Click the **Finish** button.



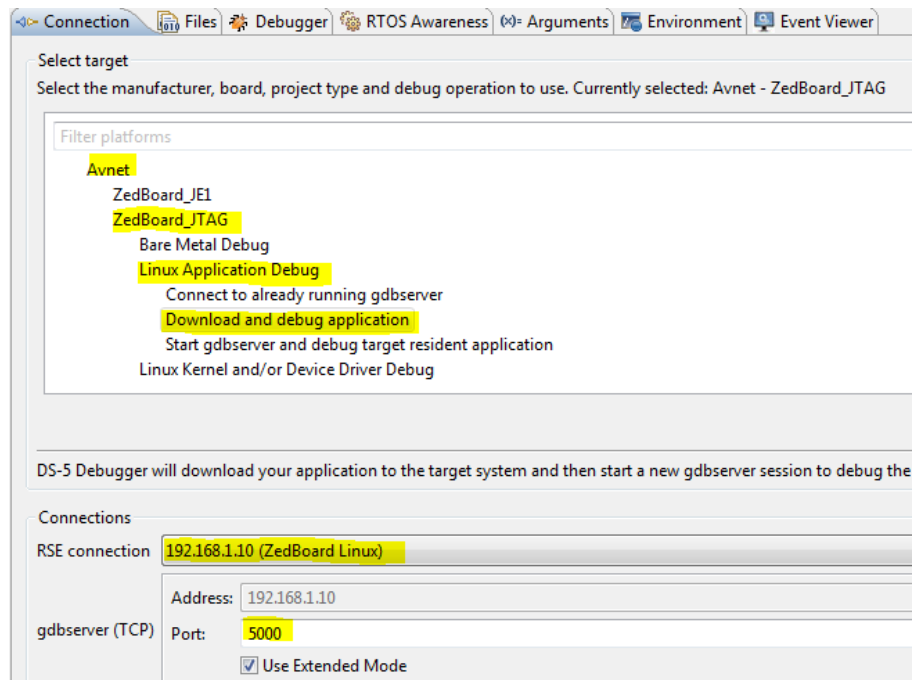
### Linux Hello World Example

7. The example is already pre-compiled, so we don't need to build it at this time. Expand the **Hello World Linux** project and right-click on the **hello** folder. Select **Debug As | Debug Configurations** from the drop-down menu.



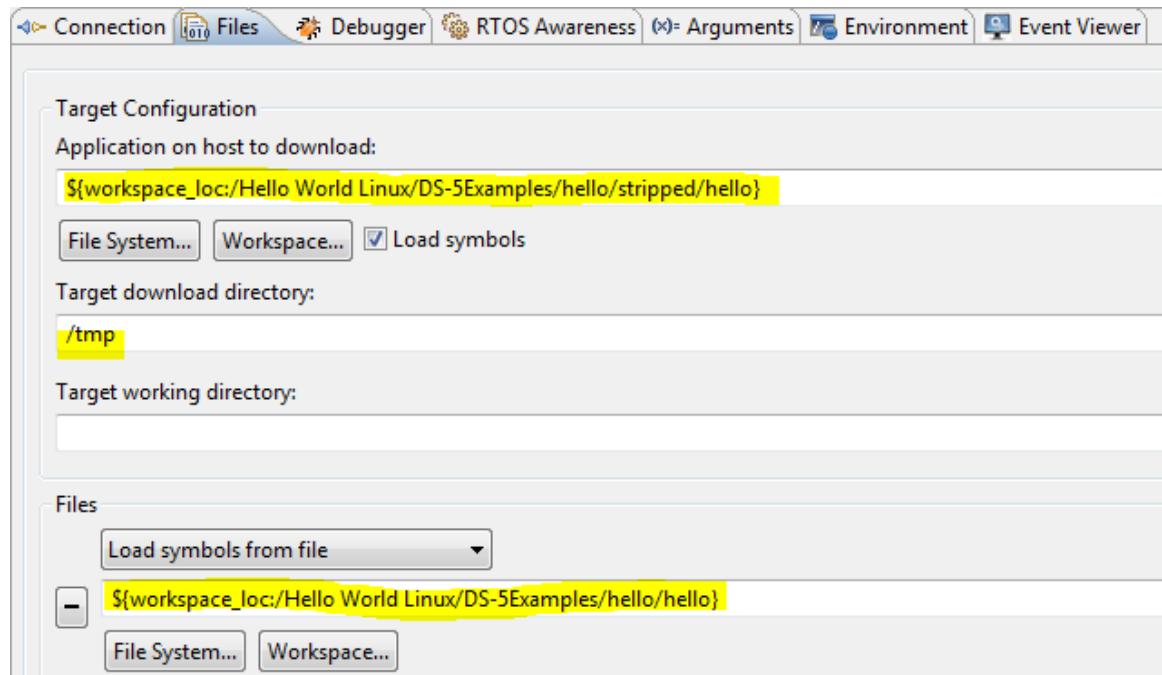
### Linux Hello World Example Imported

8. Create a new Debug Configuration and enter **Hello World Linux** for the name. We will be connecting to a ZED target that is already running Linux, so we need to set parameters that will allow us to talk to gdbserver on the Linux side and download our application to the board. In the **Connection** tab, select **Avnet | ZedBoard\_JTAG | Linux Application Debug | Download and debug application**. Under **Connections**, you should see that the ZedBoard Linux Remote Systems (RSE) connection we created has been selected for target communications (if you happen to have multiple connections, select the 192.168.1.10 entry here). The standard gdbserver port 5000 is used by default.



**Hello World Linux Debug Connection Parameters**

9. Select the **Files** tab. We can use the stripped (no debug information) version of our compiled hello program as the application to download to the board. Click the upper **Workspace** button, and select the stripped hello file from the Hello World Linux project, as shown. Enter **/tmp** for the **Target Download Directory**, and click the lower **Workspace** button to select a symbol file. For this file we will use the debug version of our compiled source.



### Linux Application Files for Debug

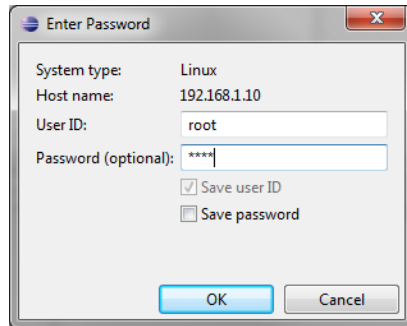
10. Select the **Debugger** tab. Ensure the Run control is set to **Debug from symbol main**. Click the **Apply** button to save the new configuration.

## Debug Hello World on Linux

From the previous steps, in DS-5 you should be in the **Debugger** tab for the **Hello World Linux** configuration. Click the **Debug** button. If you are asked for a Perspective change, click the **Yes** button.

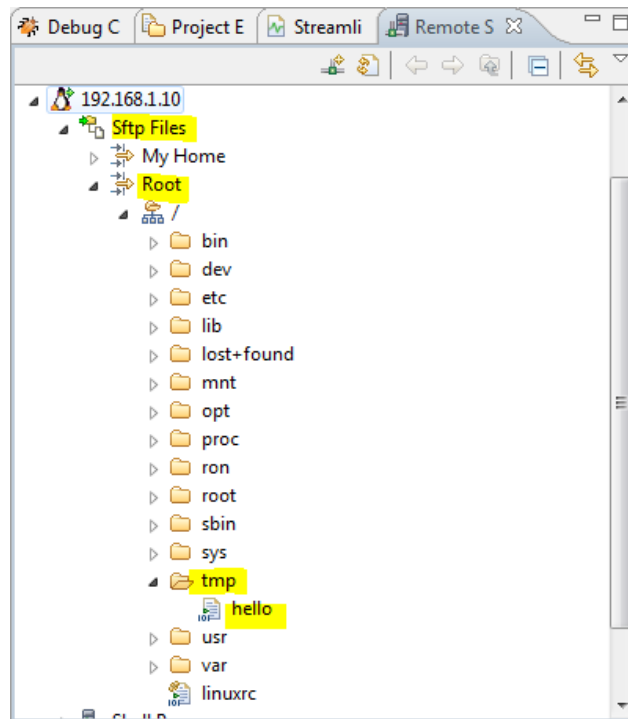
1. If you are prompted for a password, enter **root**. Click the **OK** button.





### Enter ZED target Linux root Password

2. You are now debugging a Linux application on the target. You can use this procedure to debug any Linux application in the future. Before we terminate the debug session, click on the Remote System tab and expand the **Sftp | Root** files entry. This allows access to the root file system on the target, and is a very convenient way of moving files between your host and the target during debugging sessions. If you expand the **tmp** directory, you will see the hello program, because this is the location we selected (**/tmp**) in the Debug Configuration for download.



### Root File System on ZedBoard

3. You may familiarize yourself with the standard debug controls at this point to verify the debug operation if you wish.



When you are done, select the Debug Control tab and terminate  the debug connection.

## Revision History

Date	Version	Revision
28 May 13	00	Initial Draft
20 Sept 13	01	Release

## Resources

<http://www.zedboard.org>

<http://www.xilinx.com/zyng>

<http://www.arm.com/products/tools/software-tools/ds-5/index.php>