Freescale Semiconductor
Application Note

Document Number: AN4979

Linux application debug using CodeWarrior for QorIQ LS series – ARM V7 ISA and AppTRK

Contents

1. Introduction

This document describes the steps required for Linux application debugging using the CodeWarrior for QorIQ LS series – ARM V7 ISA and AppTRK.

This document includes the following sections:

- Build the Linux sources.
- Build AppTRK.
- Perform Linux application debug using CodeWarrior for QorIQ LS series – ARM V7 ISA and AppTRK.

2. Preliminary background

The following are the steps required to compile Linux image that include AppTRK agent for the ARMv7 board.

2.1. Downloads

In order to debug a Linux application using AppTRK and CodeWarrior, download LS1021A SDK ISO from:
`http://linux.freescale.net/labDownload2/viewDownloads.php?Filter=LS1021A+SDK+V1.0+QDS&field=PL&Action=Filter`

2.2. Install the SDK with Yocto

To install Yocto on the host machine, perform these steps:

1. Mount ISO on your machine

   ```
   $ sudo mount -o loop LS1021A-SDK-<version>-<target>-<yyyymmdd>-yocto.iso /mnt/cdrom
   ```

2. As a non-root user, perform the following steps to install Yocto:

   ```
   $ cd /mnt/cdrom
   $ ./install
   ```

3. When you are prompted to input the install path, ensure that the current user has correct permission for the install path.

NOTE	There is no uninstall script. To uninstall Yocto, you can remove the `<yocto_install_path>/LS1021A-SDK-<version>-<target>-<yyyymmdd>-yocto` directory manually.

2.3. Host Environment

Yocto requires some packages to be installed on host. The following steps are used for preparing Yocto:

1. `$ cd <yocto_install_path>`
2. `$ ./scripts/host-prepare.sh`
3. `$source ./fsl-setup-poky –m <machine>`

NOTE	For example, for LS1021AQDS board, the above command will be: `$ source ./fsl-setup-pocky –m ls1021aqds –j 4 –t 4`, where `–j` is the number of jobs to spawn during the compilation stage and `–t` is the number of BitBake tasks that can be issued in parallel.

2.4. Builds

AppTRK is the target resident application for debugging Linux applications using CodeWarrior. There are two ways to have AppTRK running on the target board:

2.4.1 Deploy AppTRK into rootfs image

1. To add AppTRK to <image-target>, edit:

   `<yocto_install_path>/meta-fsl-networking/images/<image-target>.bb`
   and add
   `apptrk` to `IMAGE_INSTALL`

2. Build the Linux image using BitBake command:

```
$ cd <yocto_install_path>/
$ source ./build_<machine>_release/SOURCE_THIS
$ bitbake <image-target>
```

2.4.2 Deploy AppTRK directly into running Linux

1. SDK includes an “apptrk” recipe
2. Build the apptrk using BitBake command:

```
$ cd <yocto_install_path>/
$ source ./build_<machine>_release/SOURCE_THIS
$ bitbake apptrk
```

3. Check
   `<yocto_install_path>/build_<machine>_release/tmp/work/cortexa7hf-vfp-neon-fsl-linux-gnueabi/apptrk/git-r0/git` for the elf file

4. Secure Copy (SCP) the AppTRK executable to the target board

NOTE	CodeWarrior for ARMv7 includes the source archive and elf for apptrk. Make sure AppTRK was built with the same toolchain and the same floating point ABI (Application Binary Interface) as the libraries from the ramdisk and also as the Linux kernel image.

2.4.3 Start AppTRK

After AppTRK executable was deployed into the root file system, it can be started using the following command:

```
root@ls1021aqds:~# apptrk :12345 &
[1] 1026
```

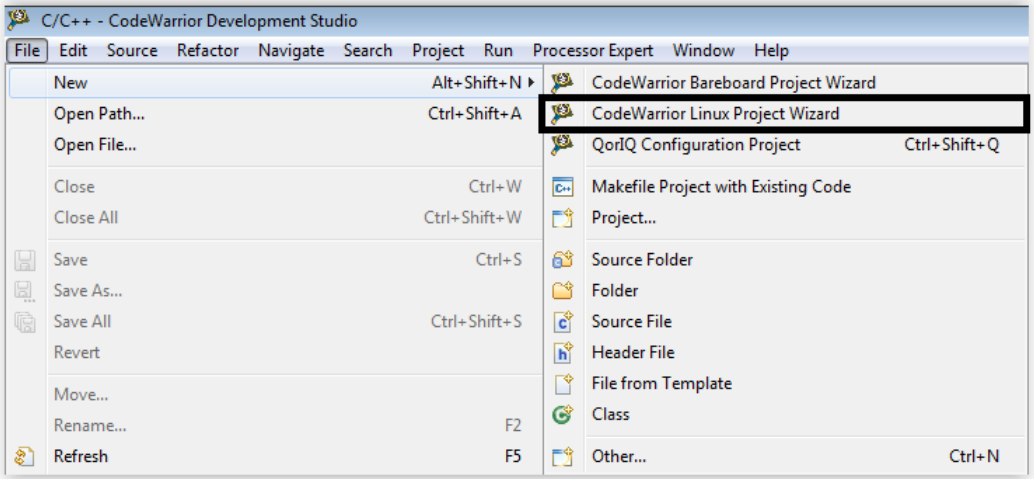
NOTE	Run apptrk in background to be able to input commands in the same console where AppTRK was started and use 12345 as communication port to align with the CodeWarrior wizard.

3. Create ARMv7 Linux Application project

To create an ARMv7 project for Linux Application, follow these steps:

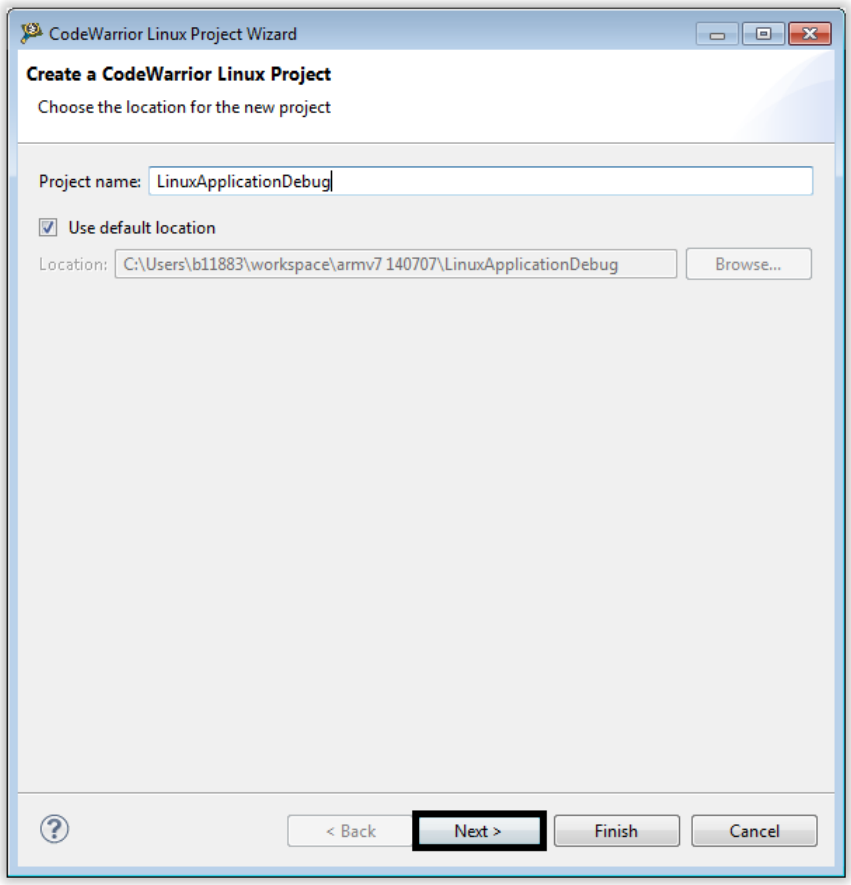
1. Start CodeWarrior for QorIQ LS series – ARM V7 ISA.
2. Choose **File** > **New > CodeWarrior Linux Project Wizard** to create a new Linux project.

Figure 1. CodeWarrior File menu option



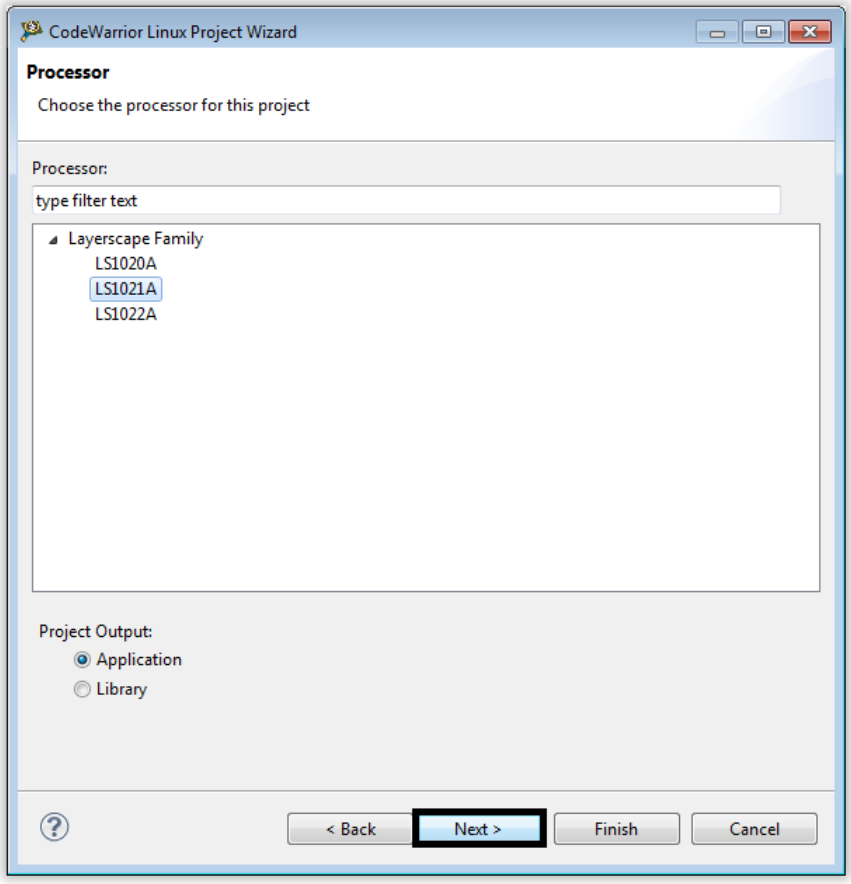
3. Specify **Project name** and **Location**, or use the default location and click **Next**.

Figure 2. CodeWarrior Linux Project wizard



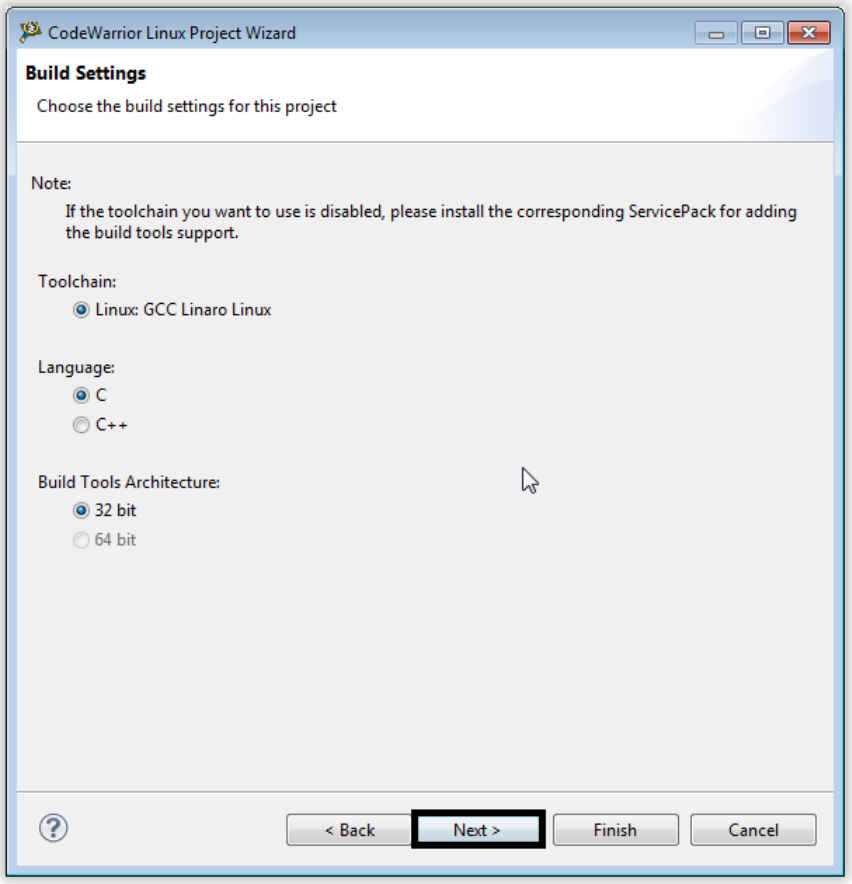
4. Select **Processor** type, **Project Output** and click **Next**.

Figure 3. Processor page



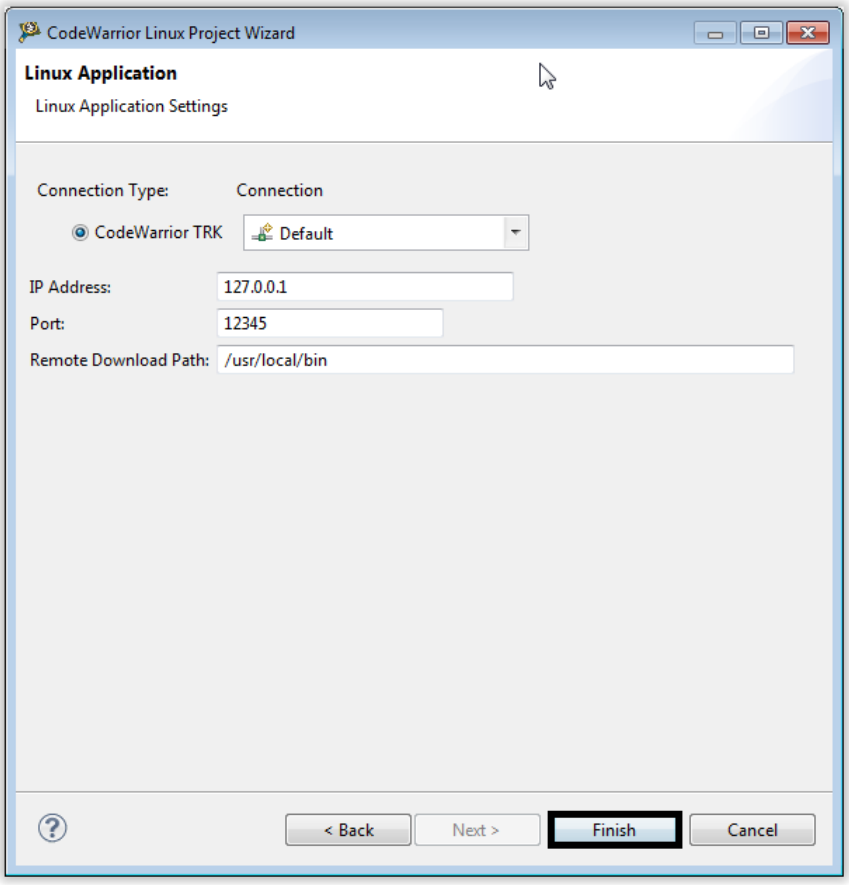
5. Configure **Build Settings** and click **Next**.

Figure 4. Build Settings page



6. Configure connection details and click **Finish** to create the Linux Application project.

Figure 5.Linux Application page



NOTE Make sure **Remote Download Path** exists on target Linux.

4. Linux Application debug

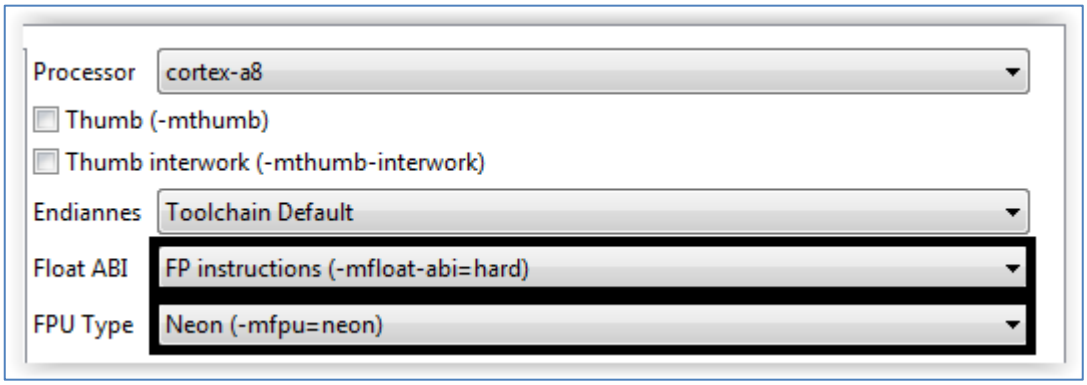
Before starting a debug session, it is mandatory to ensure that the Linux Application project is built with the same toolchain version and floating point ABI as the rootfs and AppTRK were built.

Toolchain version used by SDK is gcc-linaro 4.8 (Aarch32) 4.8-2013.12 and by default, FP is set to `-mfloat-abi=hard -mfpu=neon`.

To use the same settings for your Linux Application project, follow these steps:

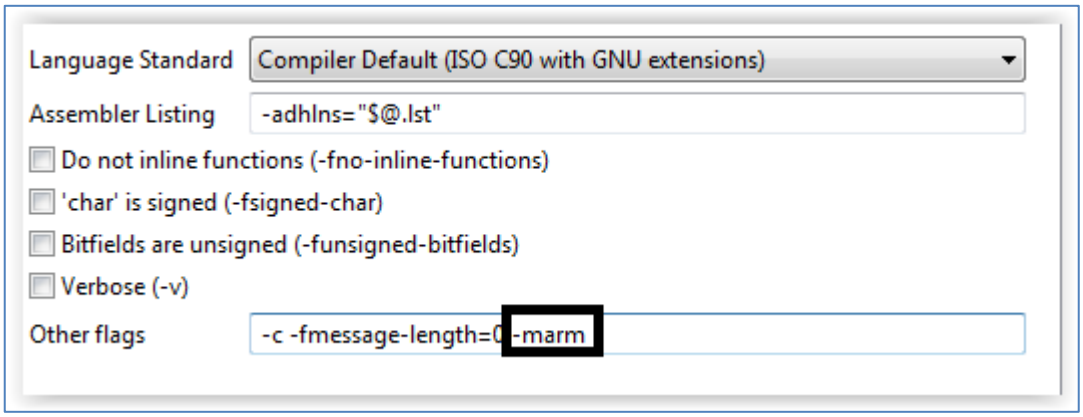
1. Go to **Project Properties > C/C++ Build > Settings > Target Processor**, and select the options as shown below.

Figure 6.Target Processor options



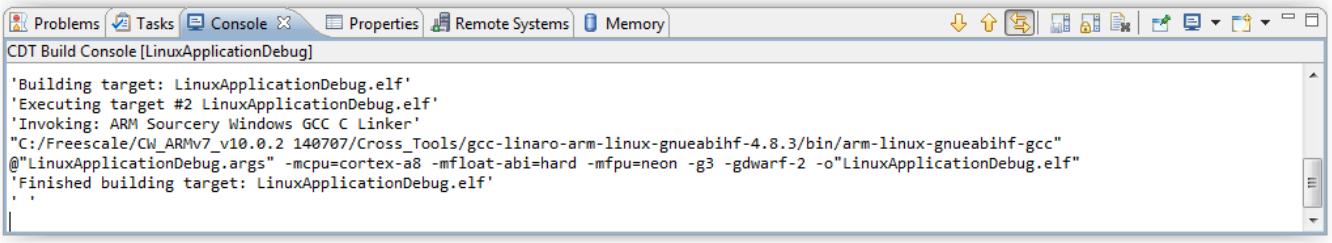
2. Go to **Project Properties > C/C++ Build > Settings > ARM Sourcery Windows GCC C Compiler**, and add the `-marm` option, as shown below.

Figure 7.ARM Sourcery Windows GCC C Compiler options



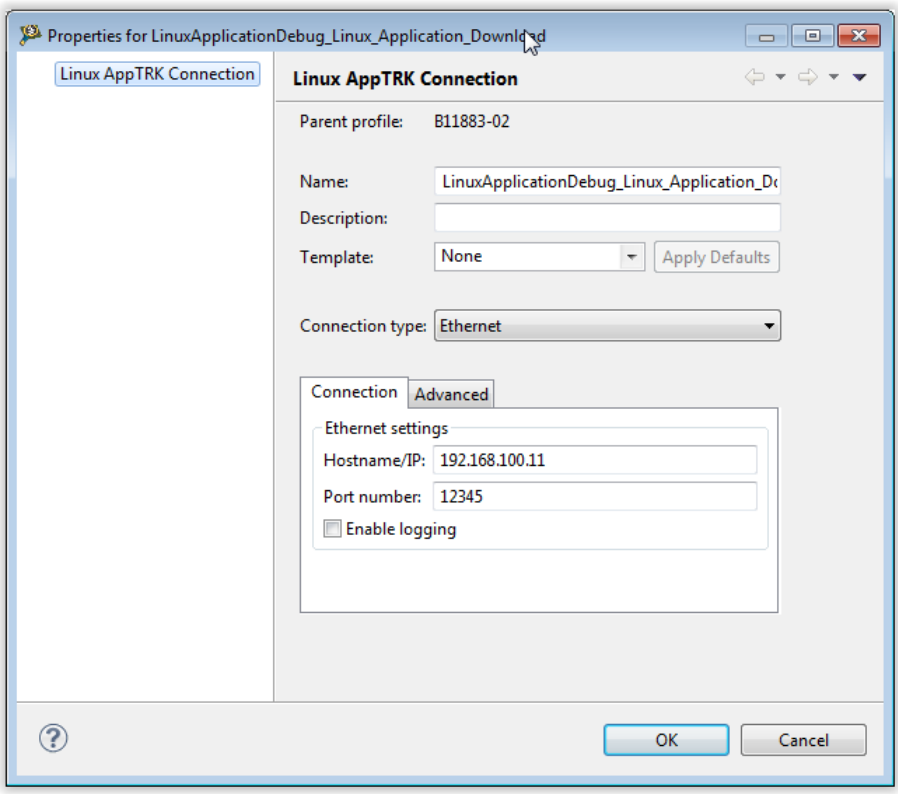
3. Build the project. The **Console** view will display the finish building target message when the build is complete.

Figure 8.Build output in Console view



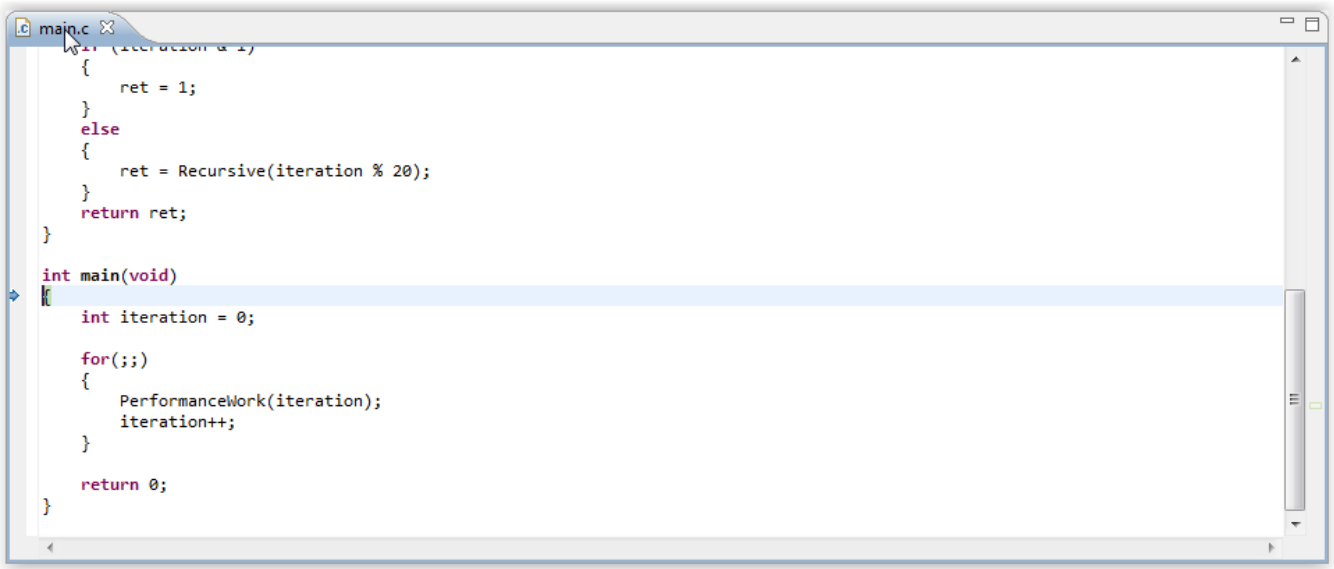
4. Check the debug configuration, **Hostname/IP** and **Port number** and then start debugging.

Figure 9. Connection details



5. Using **Download Launch**, the Linux application will be downloaded in the folder specified in the Remote download path from the Debug configurations window under **Debugger -> Remote** tab. The program will stop at `main` function.

Figure 10. Linux application stopped at main function.



6. Having the target application stopped at 'main', run control, setting/removal of breakpoints, reading/writing memory/registers can be performed.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

Document Number: AN4979

28 July 2014