Freescale Semiconductor
User's Guide

Document Number: KSDKLWIPUG
Rev. 2, 09/2015

lwIP TCP/IP Stack and Kinetis SDK Integration User's Guide

Contents

1 Overview

This document describes how to compile and run the lwIP TCP/IP stack examples. This document also provides the board-specific information related to the TWR-K60D100M, TWR-K64F120M, and TWR-K65F180M Freescale Tower System modules and the FRDM-K64F platform Freescale Freedom Development Platform.

2 Release Scope

2.1 Hardware

- Support for TWR-K64F120M, TWR-K65F180M, and TWR-K60D100M Tower System modules and FRDM-K64F Freescale Freedom Development Platform

2.2 Software

- Contains PING, TCP, UDP, and HTTP demos
- Bare Metal and RTOS are both supported

3 Requirements for Running lwIP Demos

3.1 Hardware

- TWR-K60D100M
- TWR-K64F120M/ Freescale Freedom FRDM-K64F platform
- TWR-SER and elevator
- TWR-K65F180M
- USB cable
- Ethernet cable

3.2 Software

- Freescale KSDK release package that includes the lwIP TCP/IP package
- IAR Embedded Workbench for ARM® version 7.40.3
- Keil® µVision® 5 Integrated Development Environment Version 5.15 service pack for Kinetis K60
- Kinetis Design Studio IDE Version: 3.0
- Makefiles support with GCC revision 4.9-2015-q1-update from ARM Embedded
- Atollic® TrueSTUDIO® 5.3.1

3.3 Board jumper settings

The Ethernet-related jumper settings are described in this document. For other jumper settings, see the board-specific user's guide. By default the lwIP stack uses RMII mode. Follow, the below hardware configuration:

- TWR-K60D100M
  - J10 2-3: Use the external clock from the CLOCKIN0 to keep the synchronization with the external PHY on TWR-SER Tower System module.
- TWR-K64F120M
  - J32 1-2: Use the external clock from the CLOCKIN0 to keep the synchronization with the external PHY on TWR-SER Tower System module.
- TWR-K65F180M
  - No jumper specifications.
- TWR-SER
  - J2 3-4: Ethernet PHY Clock Select 50 MHz, RMII mode. Cut off other connections on this jumper.
  - J3 2-3: Route 50 MHz clock to CLOCKIN0. Cut off other connections on this jumper.
  - J12 9-10: Ethernet PHY Configuration, pull-up CONFIG0, RMII select. Cut off other connections on this jumper.
- Freescale Freedom FRDM-K64F platform
  - No jumper specifications.

4 lwIP code structure

The lwIP code is located in this folder: <KSDK install_dir>/middleware/tcpip/lwip. The lwip folder includes the source code. There are two subfolders in the lwip folder as shown in the figure.

Figure 1. lwIP folder structure

- src
  - This subfolder includes the lwIP 1.4.1 source code which can be downloaded from this link: savannah.gnu.org
- port
  - This subfolder includes the adapter files which can make the lwIP stack run on the KSDK and different RTOSes.

5 Compiling or Running the lwIP Stack and Demos

5.1 Configuration

ENET driver configuration

This release supports both polling and interrupt mode for frame receiving. In <install_dir</platform/drivers/inc/fsl_enet_driver.h, set #define ENET_RECEIVE_ALL_INTERRUPT 0 to enable polling mode.

Alternatively, set #define ENET_RECEIVE_ALL_INTERRUPT 1 to enable interrupt mode.

5.2 Step-by-step guide for IAR

This section shows how to compile and run demos in IAR.

1. Open the workspace corresponding to different demos and different boards. For example, the lwip_ping_demo.eww on Freescale Freedom FRDM-K64F Platform under <install_dir>/examples/frdmk64f/demo_apps/lwip/lwip_ping_demo/ping_bm/iar/ or the lwip_ping_demo_freertos.eww on Freescale Freedom FRDM-K64F platform under <install_dir>/examples/frdmk64f/demo_apps/lwip/lwip_ping_demo/ping_rtos/ping_freertos/iar/. These steps use lwip_ping_demo.eww on FRDM-K64F as an example. <install_dir>/examples/frdmdk64f/demo_apps/lwip/lwip_ping_demo/ping_bm/iar/

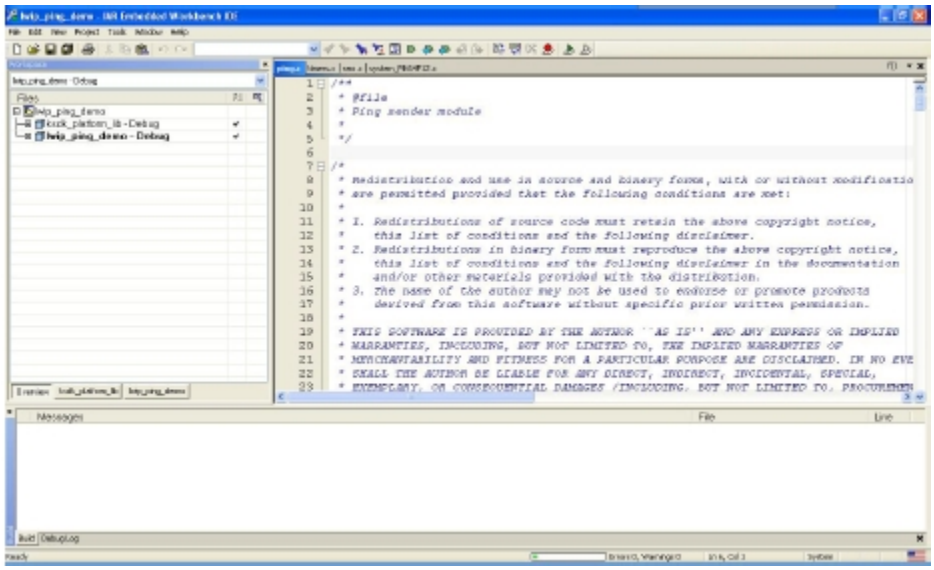

Figure 2. Workspace

2. Build the ksdk_platform_lib library.

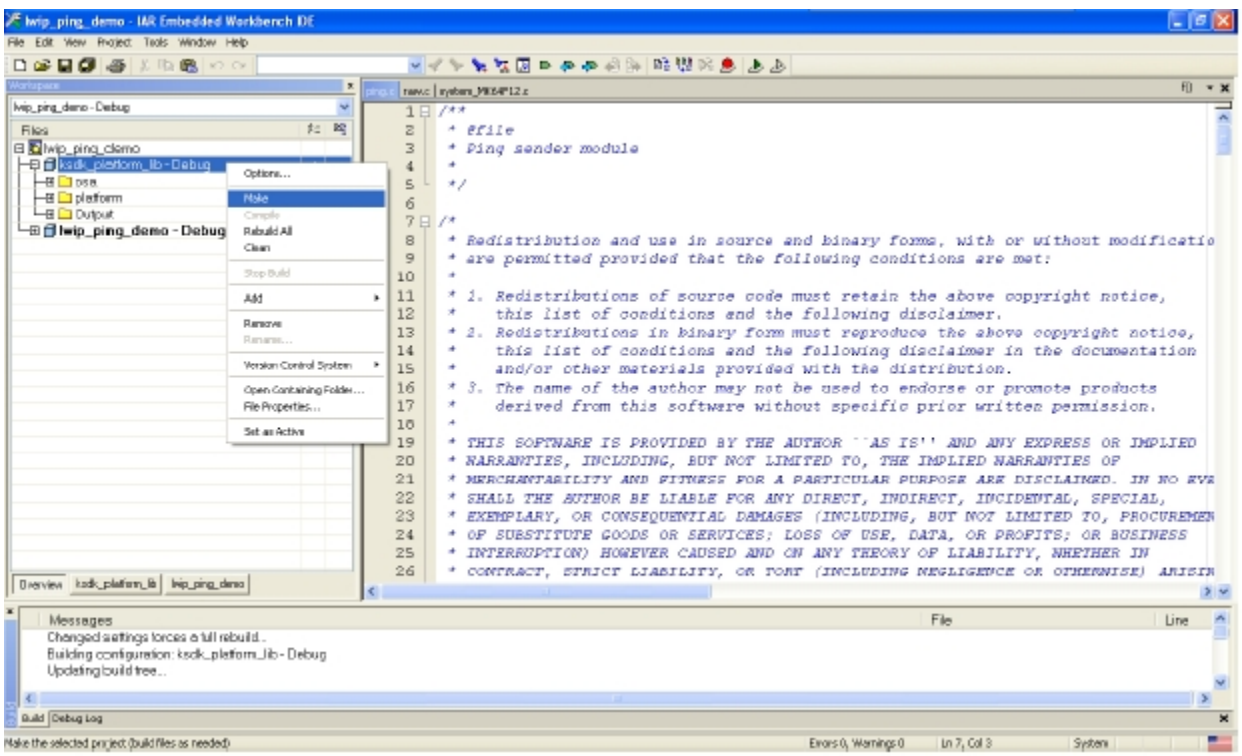

Figure 3. ksdk_platform_lib

3. Build the lwip_ping_demo.

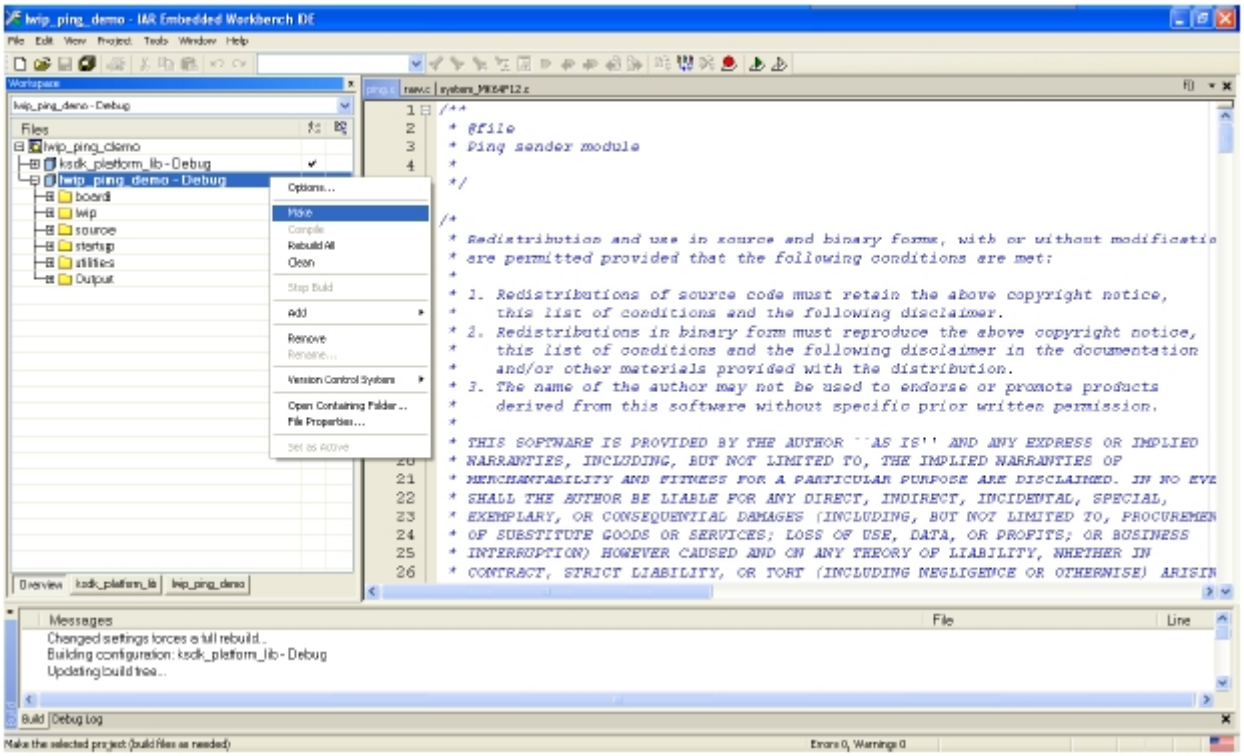

Figure 4. lwip_ping_demo

4. Click the “Download and Debug” button. Wait for the download to finish.
5. Click the “Go” button to run the demo.

5.3 Step-by-step guide for Keil

This section shows how to compile and run demos in Keil.

1. Open the workspace corresponding to different demos and different boards. For example, the lwip_ping_demo.uvmpw on Freescale Freedom FRDM-K64F platform under <install_dir>/examples/frdmk64f/demo_apps/lwip/lwip_ping_demo/ping_bm/mdk/ or the lwip_ping_demo_freertos.uvmpw on Freescale Freedom FRDM-K64F platform under <install_dir>/examples/frdmk64f/demo_apps/lwip/lwip_ping_demo/ping_rtos/ping_freertos/mdk/. These steps take lwip_ping_demo.uvmpw on Freescale Freedom FRDM-K64F platform for an example.

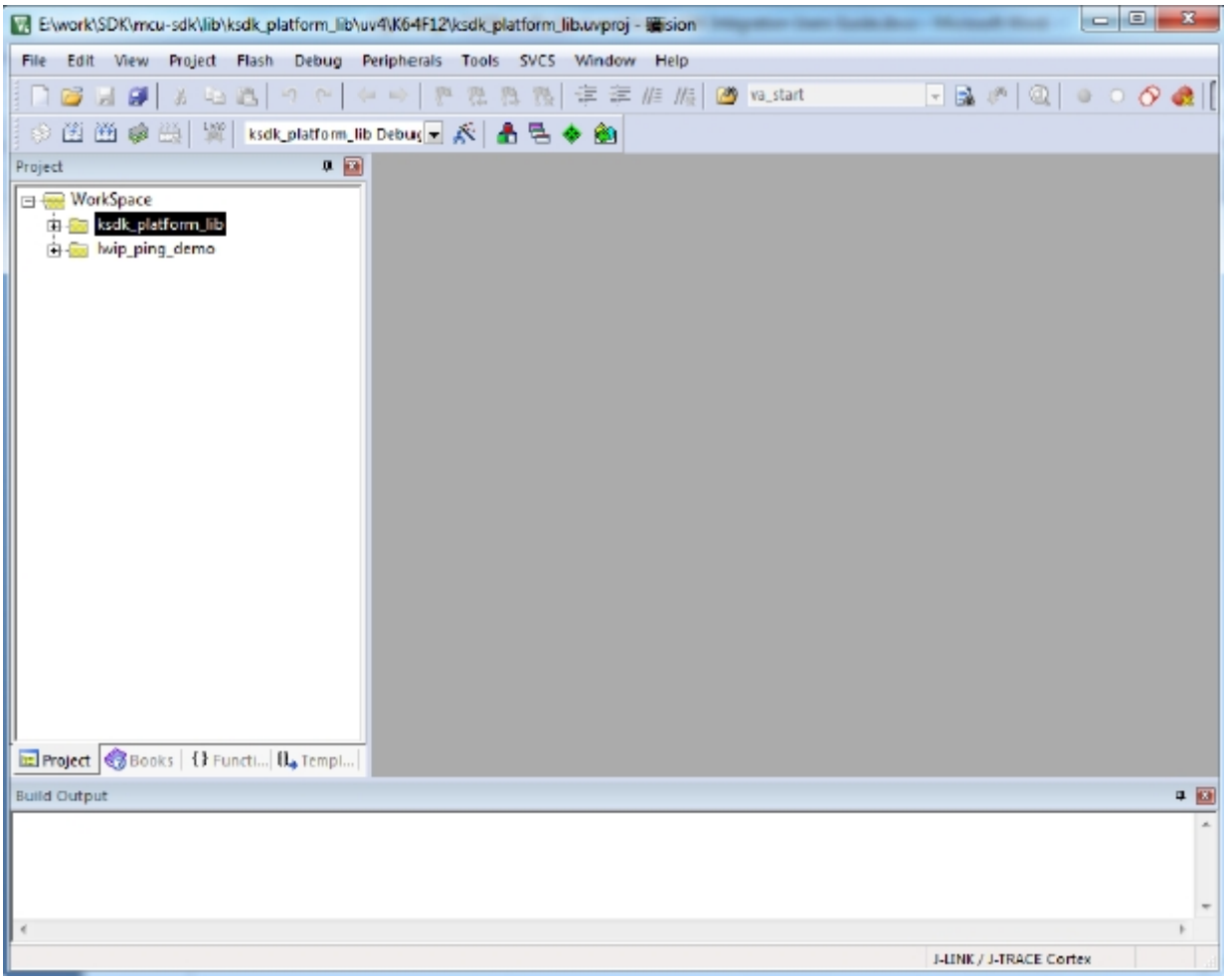

Figure 5. Workspace

2. Build the ksdk_platform_lib library.
3. Build the lwip_ping_demo.
4. Click Start/Stop Debug Session. Wait for the download to finish.
5. Click the “Run” button to run the demo.

5.4 Step-by-step guide for the Kinetis Design Studio IDE and Atollic TrueSTUDIO

This section shows how to compile and run demos in the Kinetis Design Studio IDE. The steps are identical for Atollic TrueSTUDIO.

1. The Kinetis Design Studio doesn’t have a workspace. Create a workspace and import the platform/rtos libraries and the lwIP demos. For example, ksdk_platform_lib under <install_dir>/lib/ksdk_platform_lib/kds/K64F12 and .cproject for lwip_ping_demo on Freescale Freedom FRDM-K64F platform under <install_dir>/examples/frdmk64f/demo_apps/lwip/lwip_ping_demo/ping_bm/kds/; ksdk_freertos_lib under <install_dir>/lib/ksdk_freertos_lib/kds/K64F12 and .cproject for lwip_ping_demo_freertos on Freescale Freedom FRDM-K64F platform under <install_dir>/examples/frdmk64f/demo_apps/lwip/lwip_ping_demo/ping_rtos/ping_freertos/kds/.

NOTE

For lwIP and MQX™ RTOS demos, in addition to the ksdk_mqx_lib_K64F12 and the demo project, import the mqx _<board_name> under <install_dir>/

rtos/mqx/mqx/build/kds and mqx_stdlib_<board_name> under <install_dir>/rtos/mqx/mqx_stdlib/build/kds.

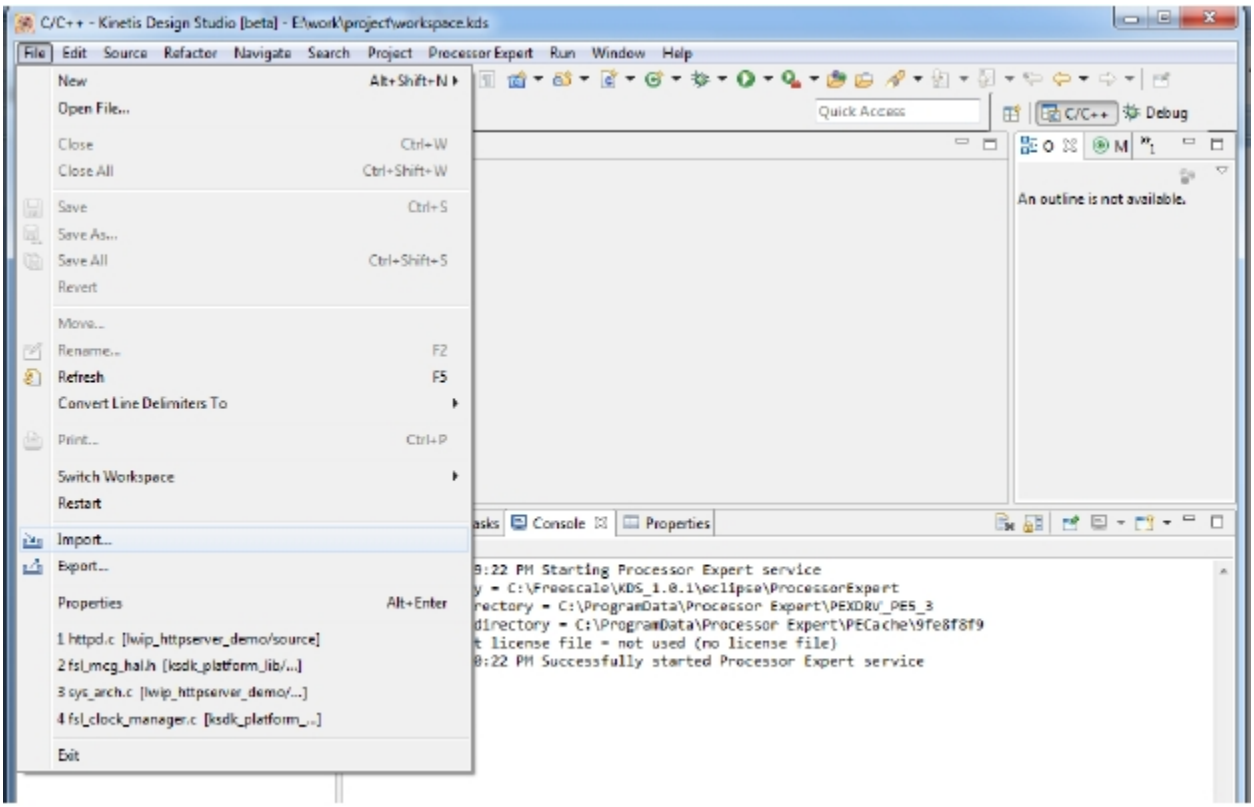

Figure 6. Import project

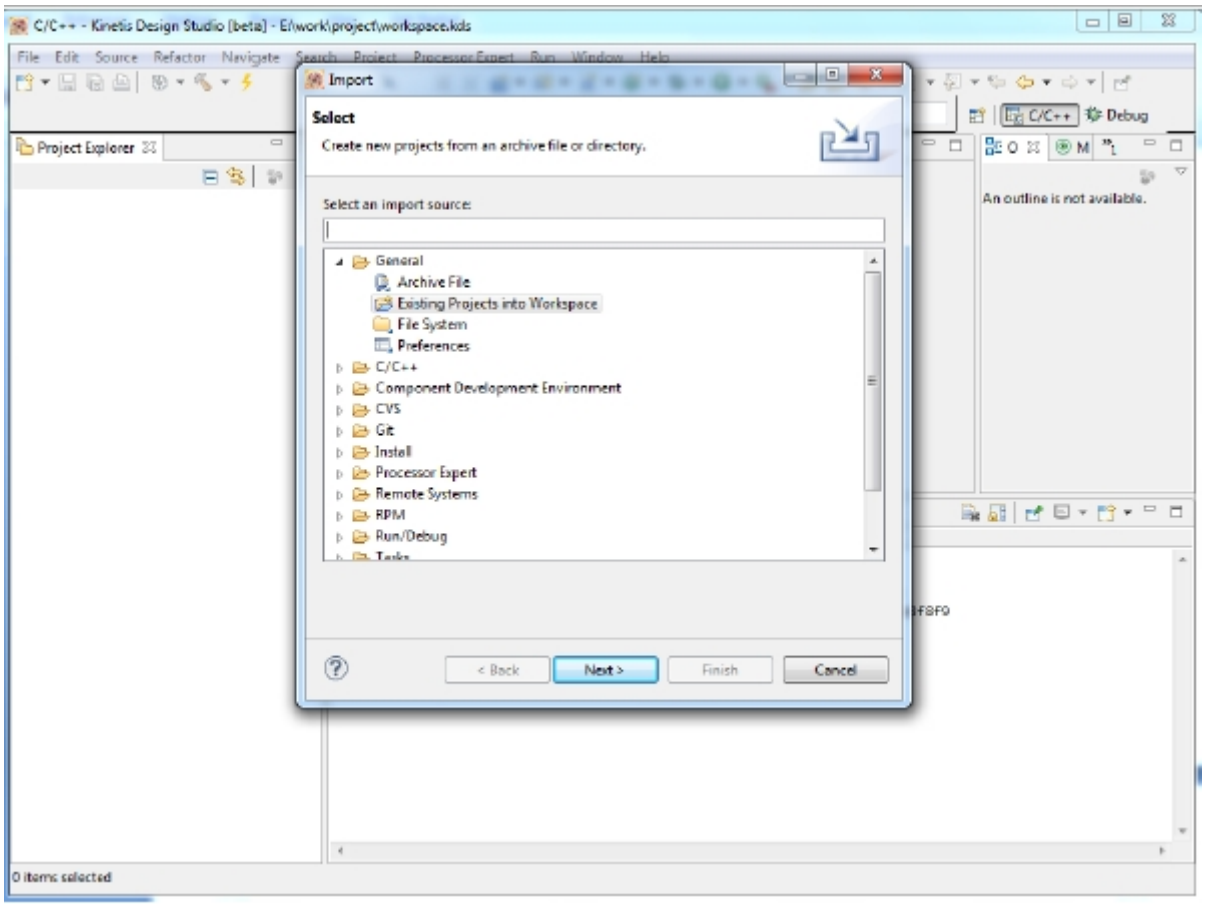

Figure 7. Import project select

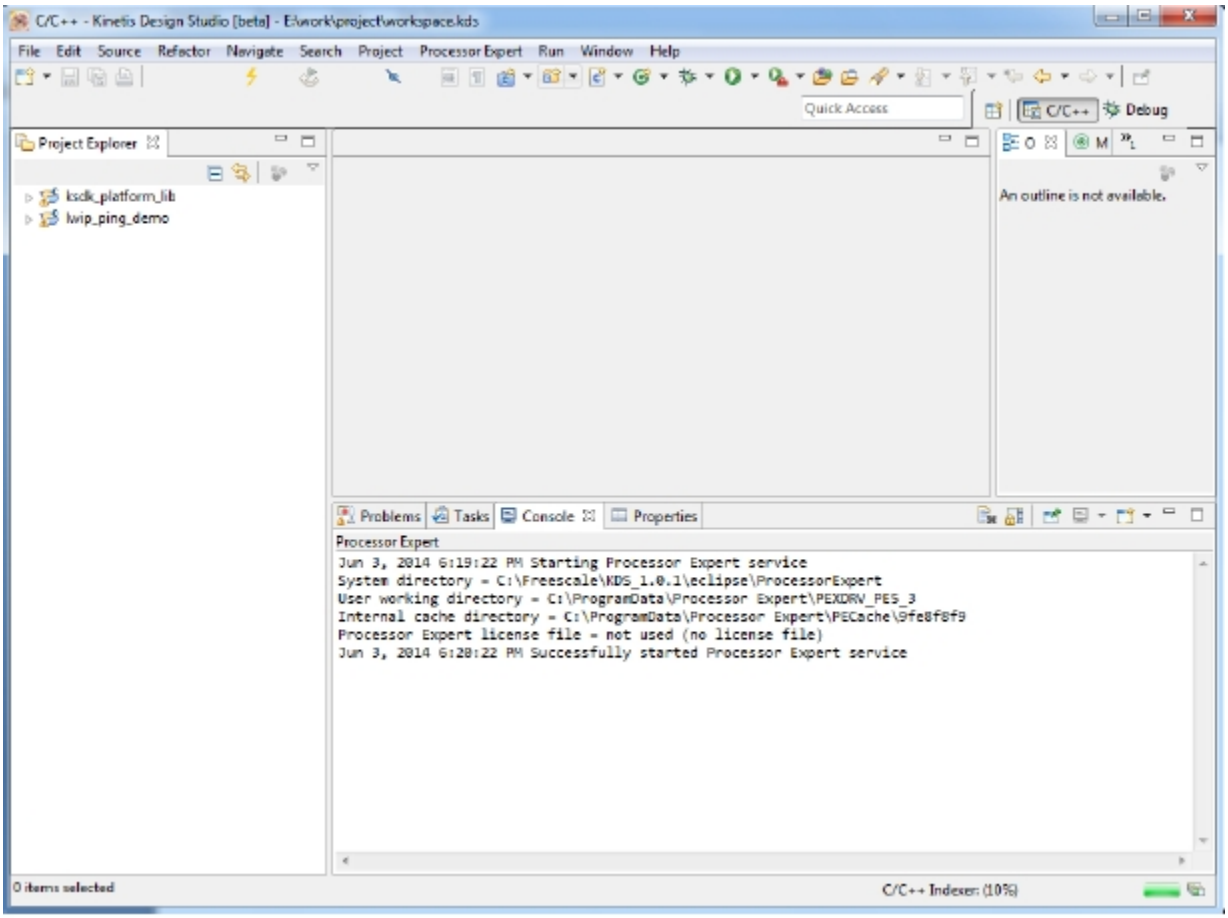

Figure 8. Lib project and demo project

2. Build the ksdk_platform_lib library.
3. Build the lwip_ping_demo.
4. Open debug configurations and choose J-Link Debugging.

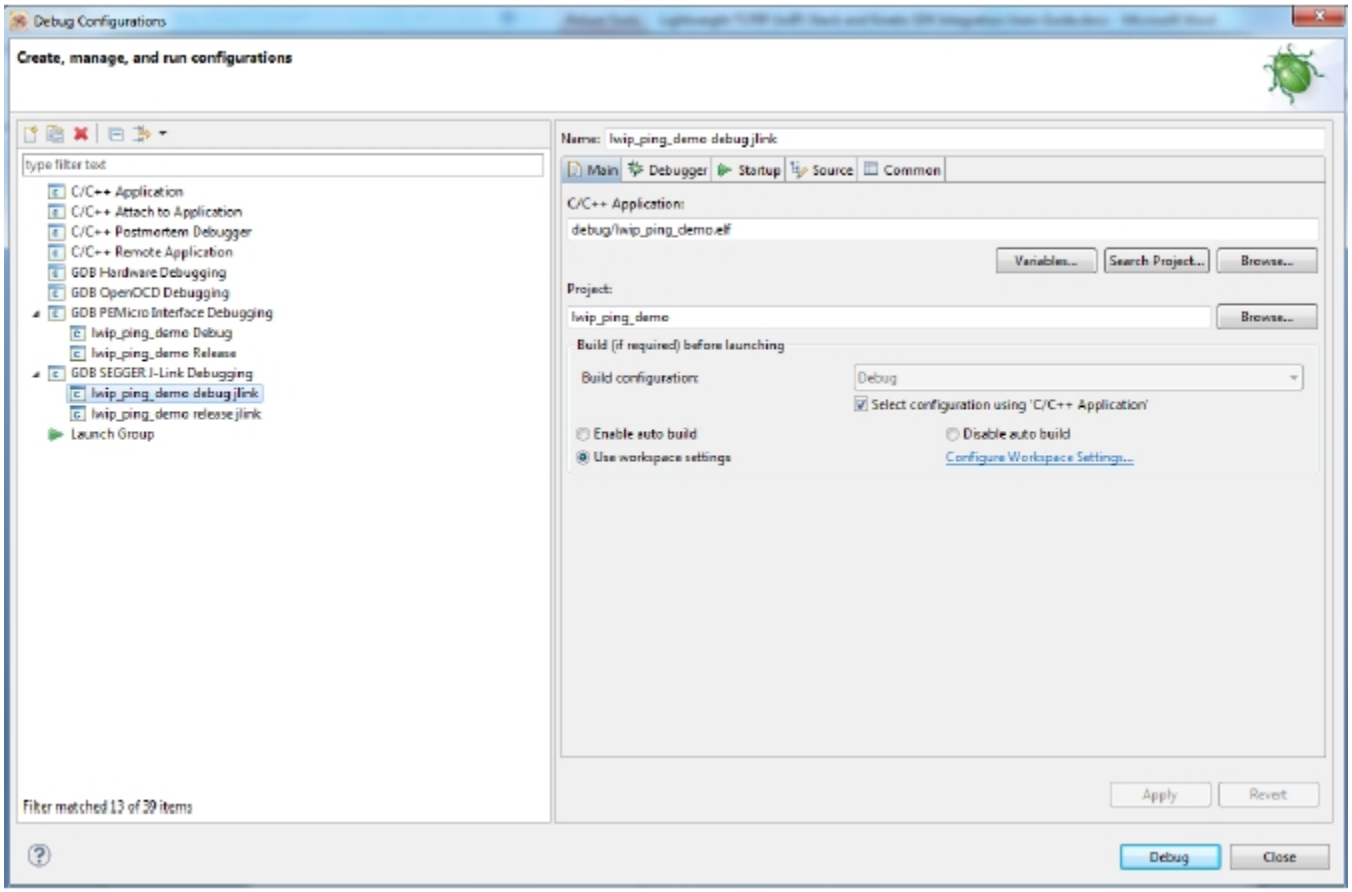

Figure 9. Debug Configurations

5. Click the “Debug” button. Wait for the download to finish.
6. Click the “Resume” button to run the demo.

5.5 Step-by-step guide for ARMGCC and KDSGCC

1. ARMGCC and KDSGCC both use cMake to generate makefiles. Run the batch file (in the Windows® operating system) or sh file (in Linux® operating system) to build projects. These steps use ARMGCC as an example.
2. Before building the lwIP demos in the KSDK, the driver library project should be built to generate the library archives:
   - libksdk_platform.a
   - libksdk_platform_freertos.a
   - libksdk_platform_ucosii.a
   - libksdk_platform_ucosiii.a
   - libksdk_platform_mqx.a
   - lib_mqx.a
   - lib_mqx_stdlib.a
3. To build the platform library, change the current directory to <install_dir>/lib. libksdk_platform.a for TWR-K64F120M as an example under ksdk_platform_lib/ armgcc/K64F12, run build_all.bat to build both debug and release lib. For lib_mqx.a, change the directory to <install_dir>/rtos/mqx/mqx/build/armgcc/mqx_<board_name>. Separately, run the build_debug.bat and build_release.bat to build debug and release libs. For lib_mqx_stdlib.a, change the directory to <install_dir>/rtos/mqx/mqx_stdlib/build/armgcc/mqx_stdlib_<board_name> and separately run the build_debug.bat and build_release.bat to build debug and release libs.
4. Change to the demo directory. For example: <install_dir>/examples/frdmk64f/demo_apps/lwip/lwip_ping_demo/ping_bm/armgcc

5. Run build_all.bat to build both debug and release projects.
6. Go to the debug/release directory to download and run the elf file using gdb.

6 Revision History

This table summarizes revisions to this document.

Table 1. Revision history

Revision number	Date	Substantive changes
2	09/2015	Updated tool versions in Section 3.2 and updated Section 5.2.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number KSDKLWIPUG
Revision 2, 09/2015