Freescale Semiconductor, Inc.
User's Guide

Document Number: KSDKLWIPUG
Rev. 1.0.0, 07/2014

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

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-K64F120M Tower System module and the Freescale Freedom FRDM-K64F platforms.

Contents

2 Release scope

2.1 Hardware

- Support for TWR-K64F120M Tower System module and Freescale Freedom FRDM-K64F platform

2.2 Software

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

3 Requirements for running lwIP demos

3.1 Hardware

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

3.2 Software

- Freescale KSDK release package that includes the lwIP TCP/IP package
- IAR Embedded Workbench for ARM® version 7.20.2
- Keil µVision5 Integrated Development Environment Version 5.11 service pack for Kinetis K60
- Kinetis Design Studio Version: 1.1
- Makefiles support with GCC revision 4.8.3 from ARM Embedded

3.3 Board jumper settings

The Ethernet-related jumper settings are described in this document. For other jumper settings, see board-related user's guide.

By default the lwIP stack uses RMII mode, please follow the below hardware configuration:

- TWR-K64F120M
    - TWR-K64F120M Tower System module board

- J32 1-2: Use the external clock from the CLOCKIN0 to keep the synchronization with the external PHY on TWR-SER board.

- 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 the "tcpip/lwip" folder at the root level of the Kinetis_SDK folder.

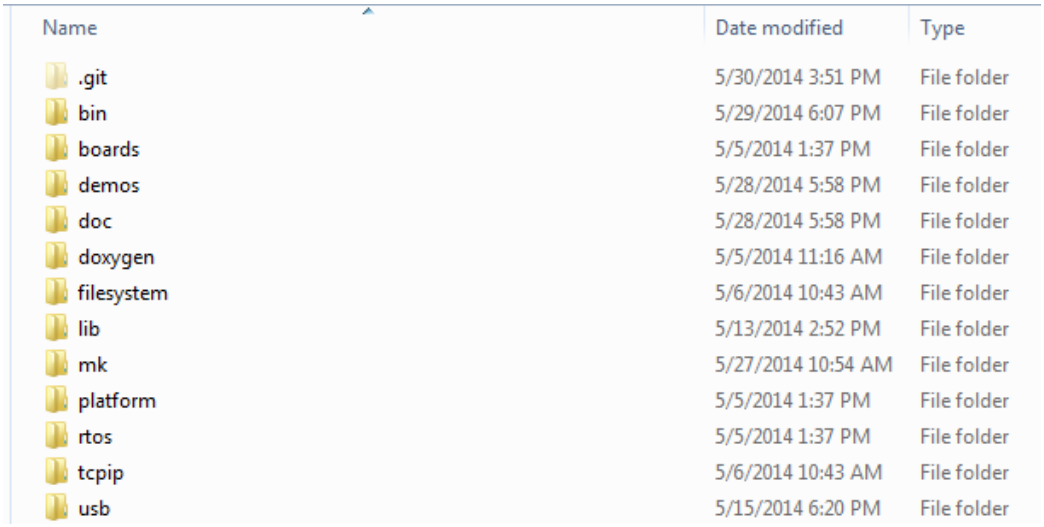

Figure 4-1 SDK folder structure

The lwIP folder includes the source code. There are two subfolders in the lwIP folder as shown in the figure.

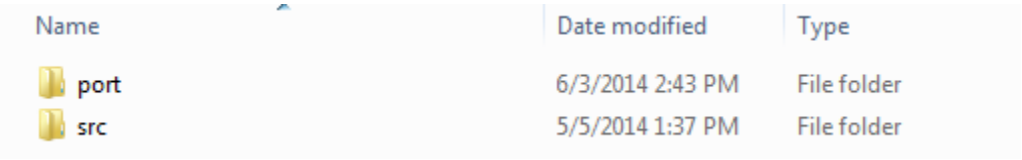

Figure 4-2 lwIP folder structure

- src

  This subfolder includes the lwIP 1.4.1 source code which can be downloaded from this link: download.savannah.gnu.org/releases/lwip/

- 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

1. ENET driver configuration

This release supports both polling and interrupt mode for frame receiving.

In <*install_dir*>*/platform/drivers/enet/fsl_enet_driver.h*, set

#define ENET_RECEIVE_ALL_INTERRUPT 0 to enable polling mode.

Or 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>/demos/lwip_ping_demo/ping_bm/iar/frdmk64f120m or the lwip_ping_demo_freertos.eww on Freescale Freedom FRDM-K64F platform under <install_dir>/demos/lwip_ping_demo/ping_rtos/ping_freertos/iar/frdmk64f120m. These steps use lwip_ping_demo.eww on FRDM-K64F120M as an example.

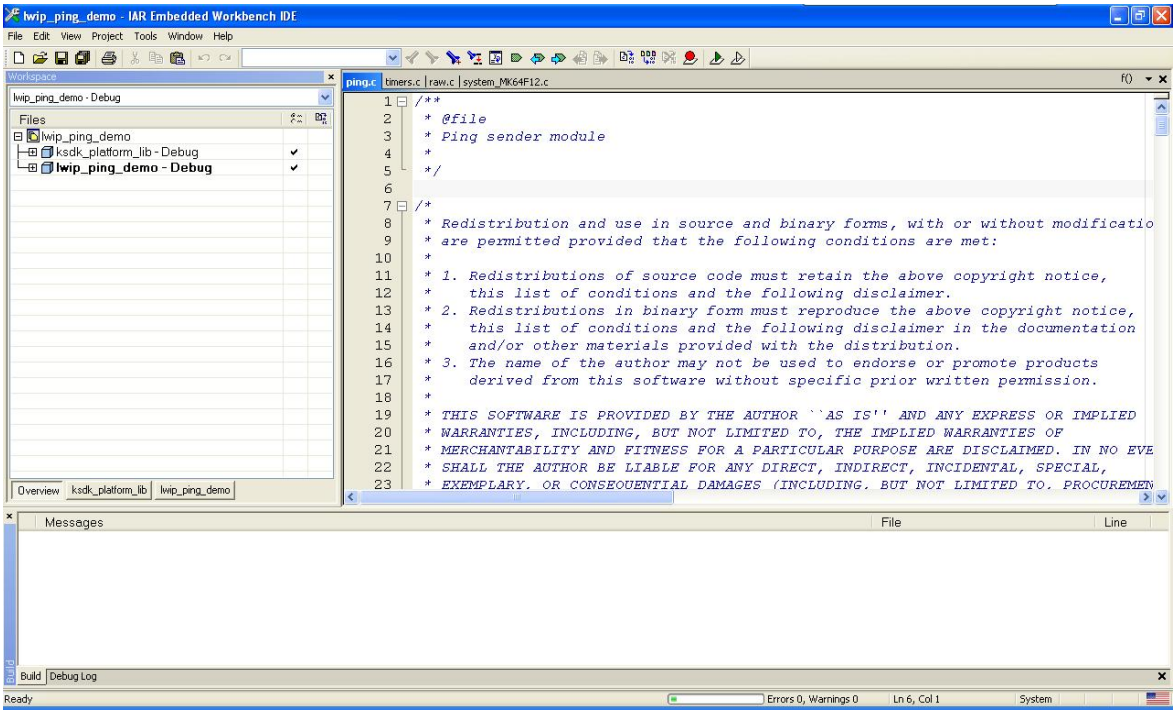

Figure 5-1 Workspace

2. Build the ksdk_platform_lib library.

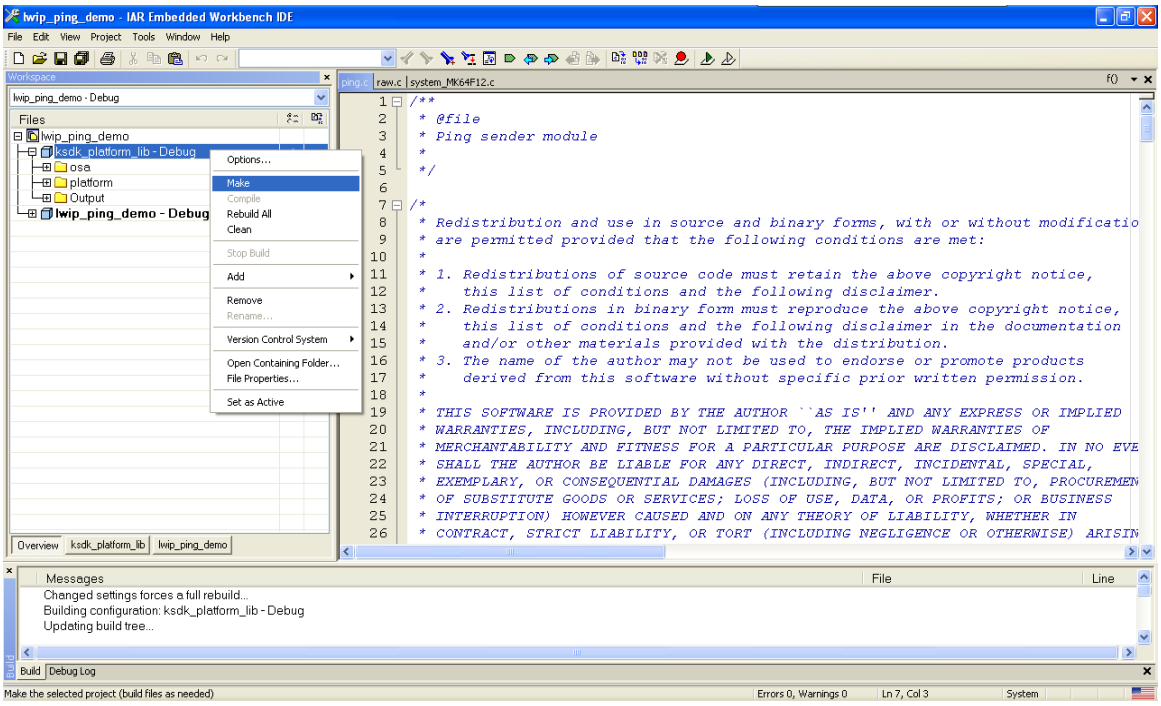

Figure 5-2 ksdk_platform_lib

3. Build the lwip_ping_demo.

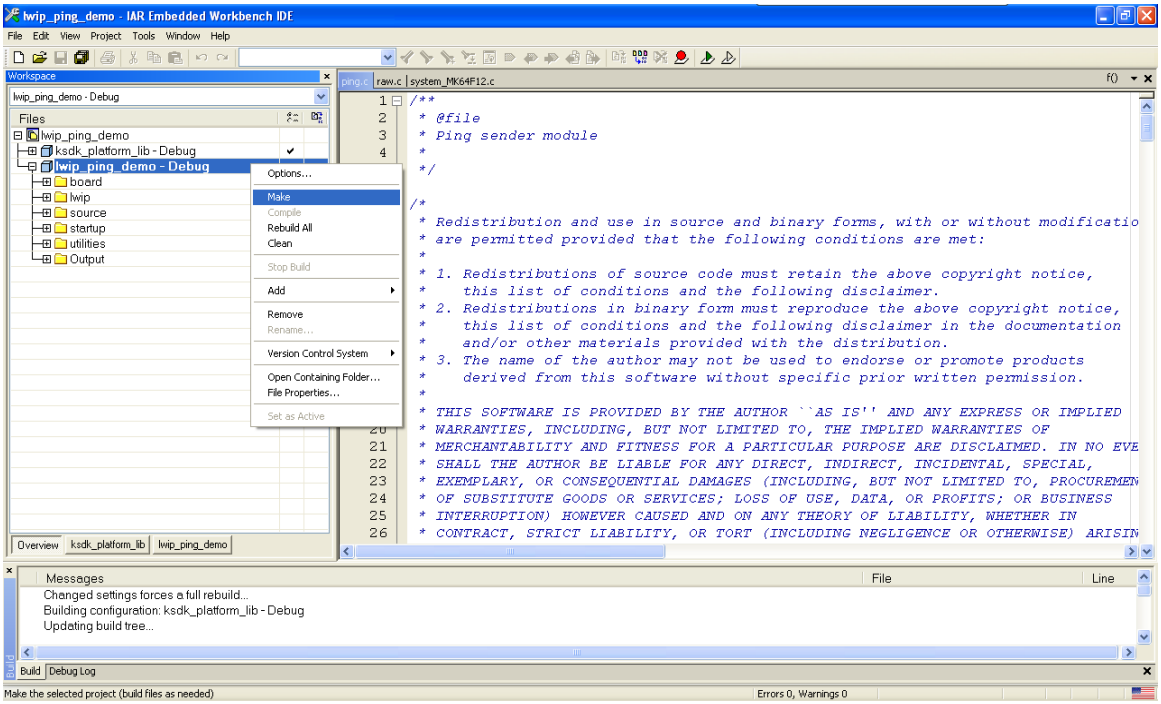

Figure 5-3 lwip_ping_demo

4. Click Download and Debug. 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>/demos/lwip_ping_demo/ping_bm/uv4/frdmk64f120m or the lwip_ping_demo_freertos.uvmpw on Freescale Freedom FRDM-K64F platform under <install_dir>/demos/lwip_ping_demo/ping_rtos/ping_freertos/uv4/frdmk64f120m. These steps take lwip_ping_demo.uvmpw on Freescale Freedom FRDM-K64F platform for an example.

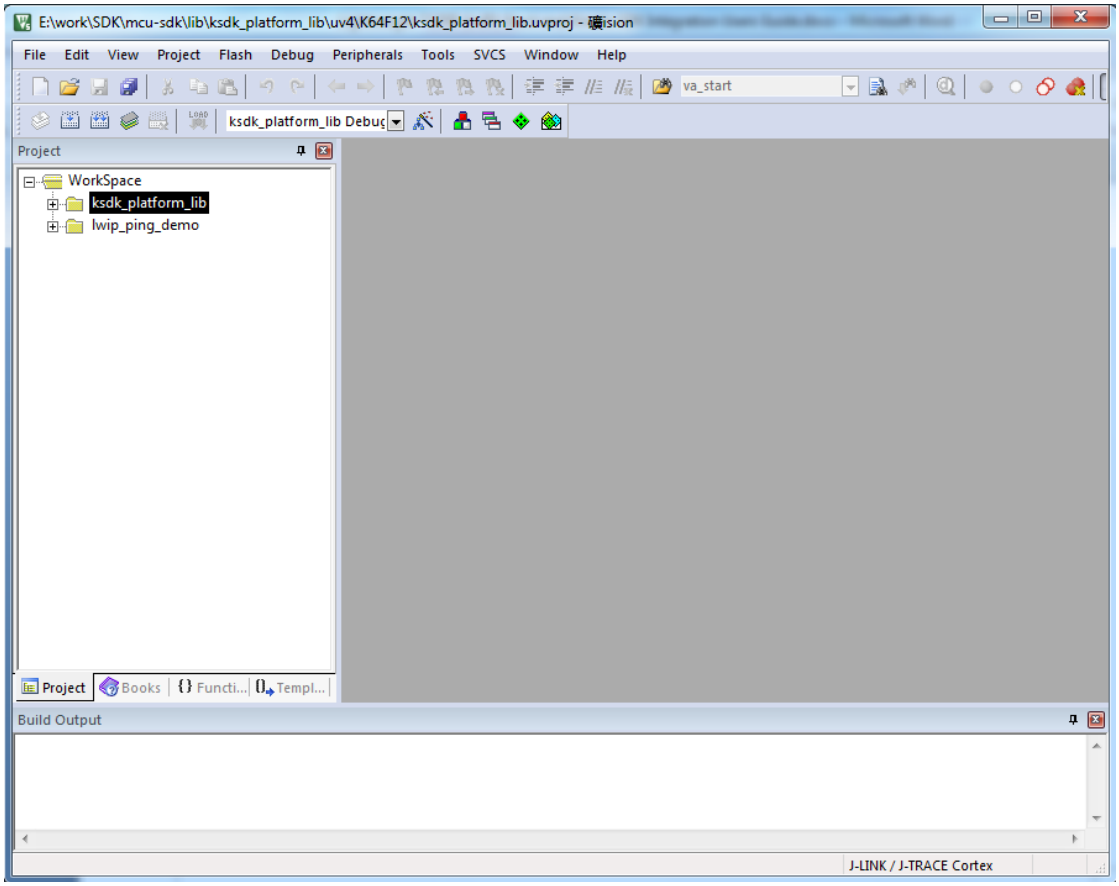

Figure 5-4 Workspace

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 Run to run the demo.

5.4 Step-by-step guide for the Kinetis Design Studio

This section shows how to compile and run demos in the Kinetis Design Studio.

1. The Kinetis Design Studio doesn't have a workspace. Create a workspace and import the lwIP demos and the platform/rtos libraries. 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>/demos/lwip_ping_demo/ping_bm/kds/frdmk64f120m; 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>/demos/lwip_ping_demo/ping_rtos/ping_freertos/kds/frdmk64f120m.

Note

For lwIP and MQX RTOS demos, in addition to the ksdk_mqx_ib_K64F12 and the demo project, import the psp_$ (board), bsp_$ (board) and mqx_stdlib_$ (board).

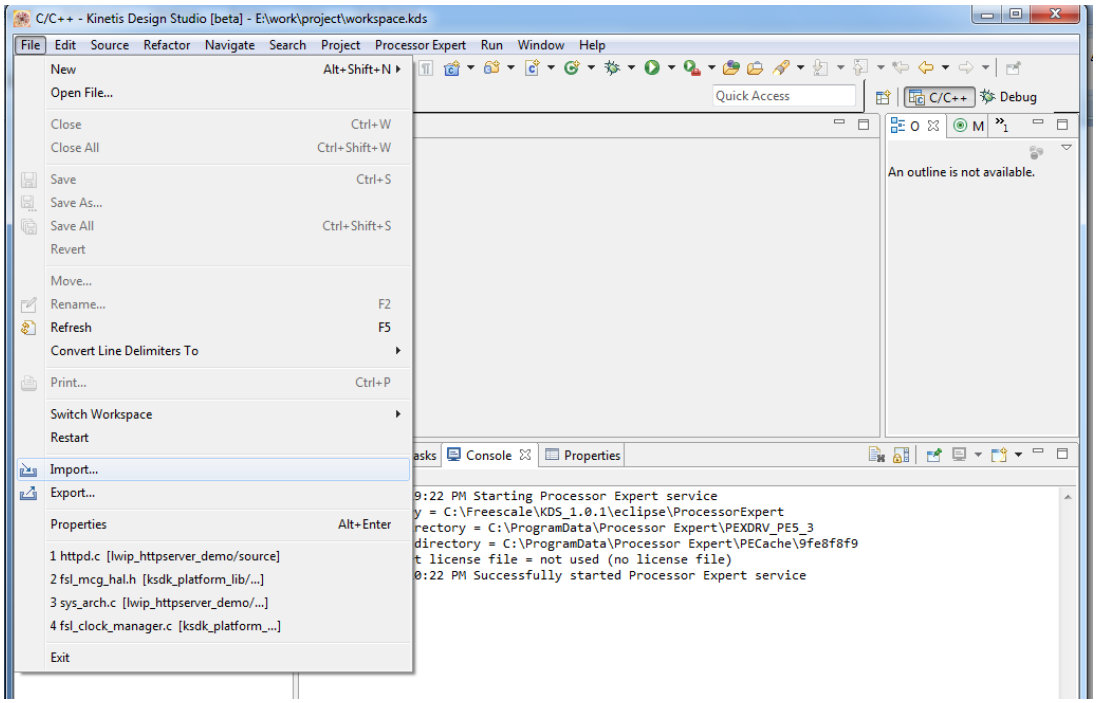

Figure 5-5 Import project -1

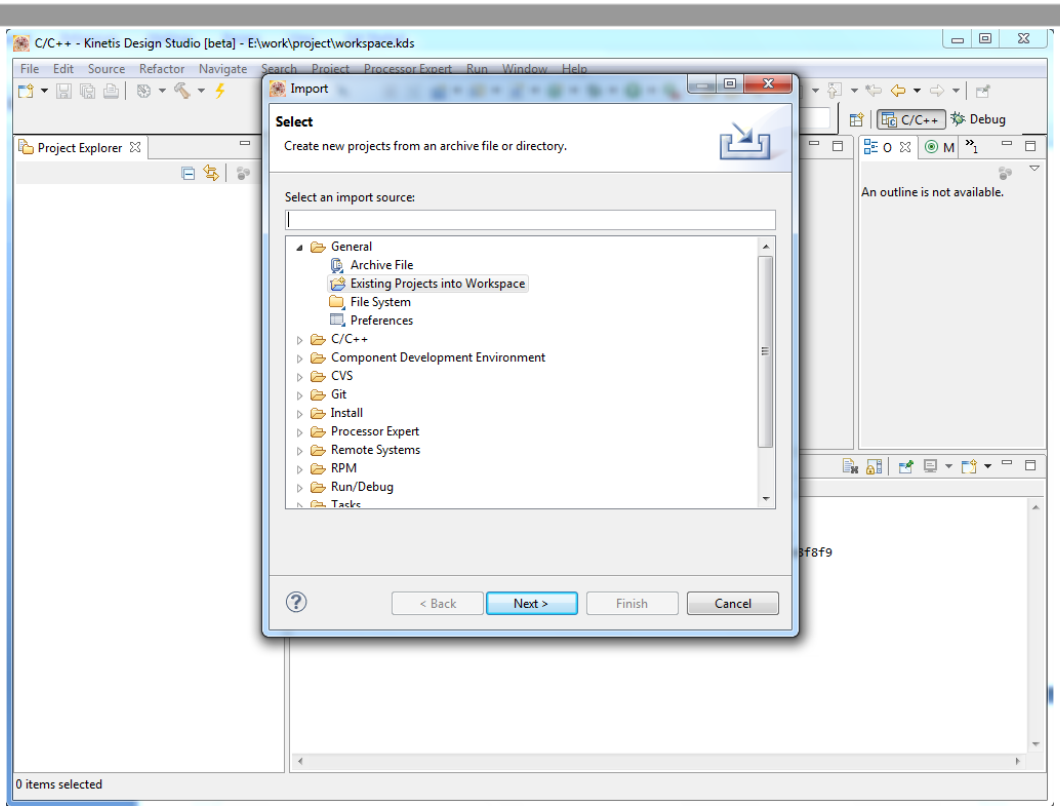

Figure 5-6 Import project - 2

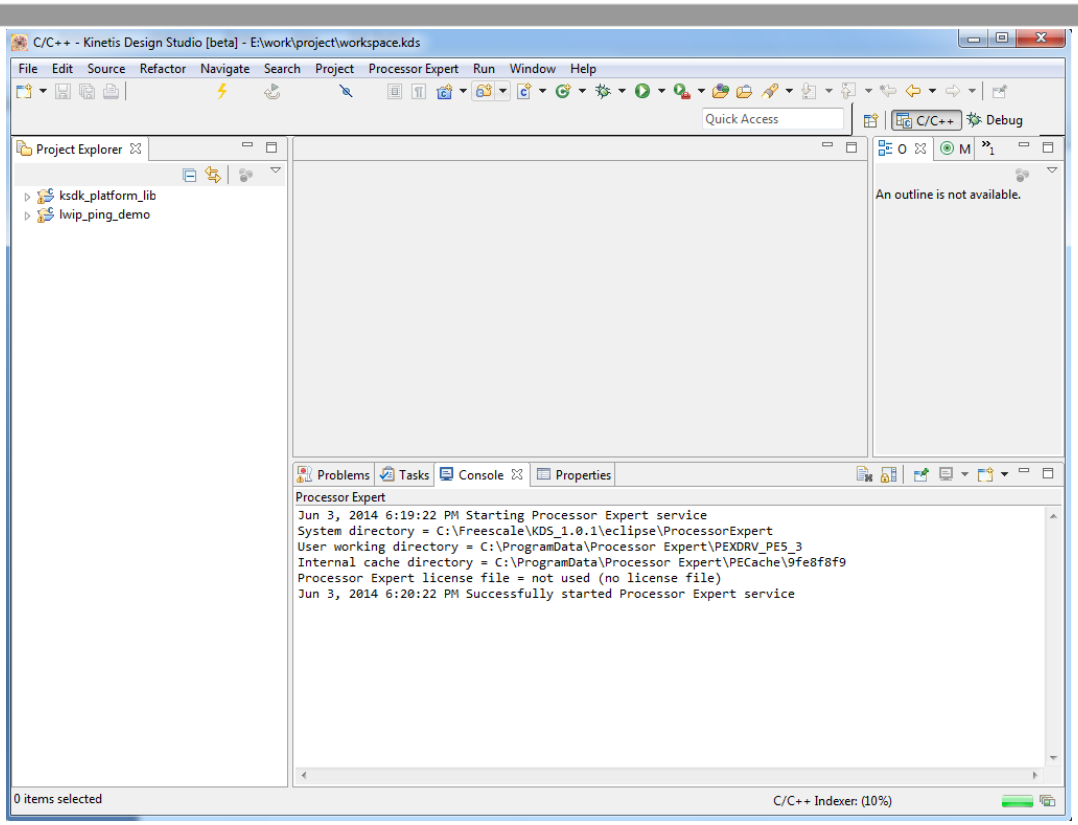

Figure 5-7 Lib project and demo project

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

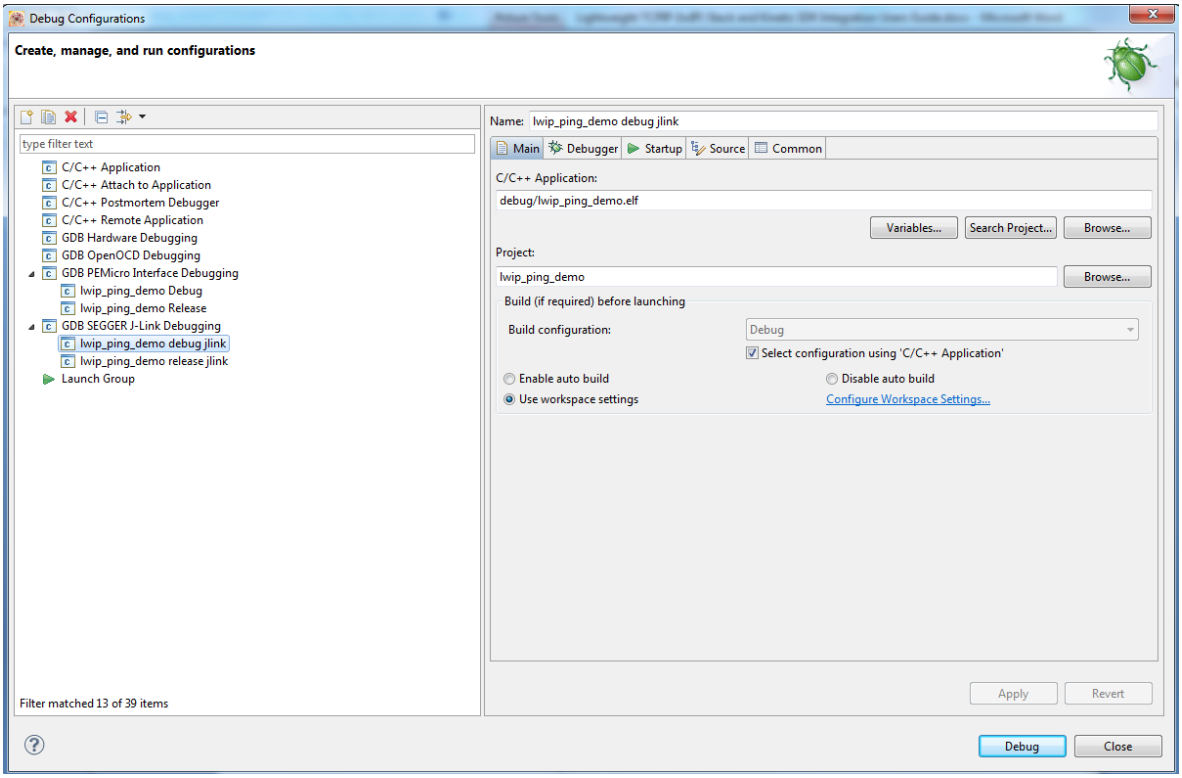

Figure 5-8 Debug Configurations

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

5.5 Step-by-step guide for GCC

1. GCC only supports demos I the lwIP_<demo_name>_demo/<demo_name>_bm/directories.
2. Before building the lwIP demos in the KSDK, the driver library project should be built to generate the library archives: ksdk_platform_lib.a.
3. To build the platform library, change the current directory in GCC Command prompt to:

   <install_dir>/lib/ksdk_platform_lib/gcc/K64F12
4. Run the command mingw32-make build=debug or mingw32-make build=release.
5. Change to the demo directory.

   For example: <install_dir>/demos/lwip_ping_demo/ping_bm/gcc/frdmk64f120m
6. Run the command mingw32-make build=debug or mingw32-make build=release.
7. Go to the Flash_Debug/Flash_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**	**Substantial changes**
1.0.0	7/2014	Initial release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: KSDKLWIPUG
Rev. 1.0.0
07/2014