

IwIP 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 jumper and hardware settings.

2 Release Scope

2.1 Hardware

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

3 Requirements for Running lwIP Demos

1	Overview.....	1
2	Release Scope.....	1
3	Requirements for Running lwIP Demos....	1
4	lwIP code structure.....	2
5	Compiling or Running the lwIP Stack and Demos.....	3
6	Revision History.....	9



3.1 Hardware

- 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 7.50.0
- Keil[®] μ Vision[®] 5 Integrated Development Environment Version 5.17 service pack for Kinetis K60
- Kinetis Design Studio IDE Version: 3.0
- Makefiles support with GCC revision 4.9-2015-q3-update from ARM Embedded
- Atollic[®] TrueSTUDIO[®] 5.4.0

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-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/lwip_1.4.1. 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 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 the Freescale Freedom FRDM-K64F Platform under `<install_dir>/boards/frdmk64f/demo_apps/lwip/lwip_ping/bm/iar/` or the `lwip_ping_demo_freertos.eww` on the Freescale Freedom FRDM-K64F platform under `<install_dir>/boards/frdmk64f/demo_apps/lwip/lwip_ping/freertos/iar/`. These steps use `lwip_ping_demo.eww` on FRDM-K64F as an example.

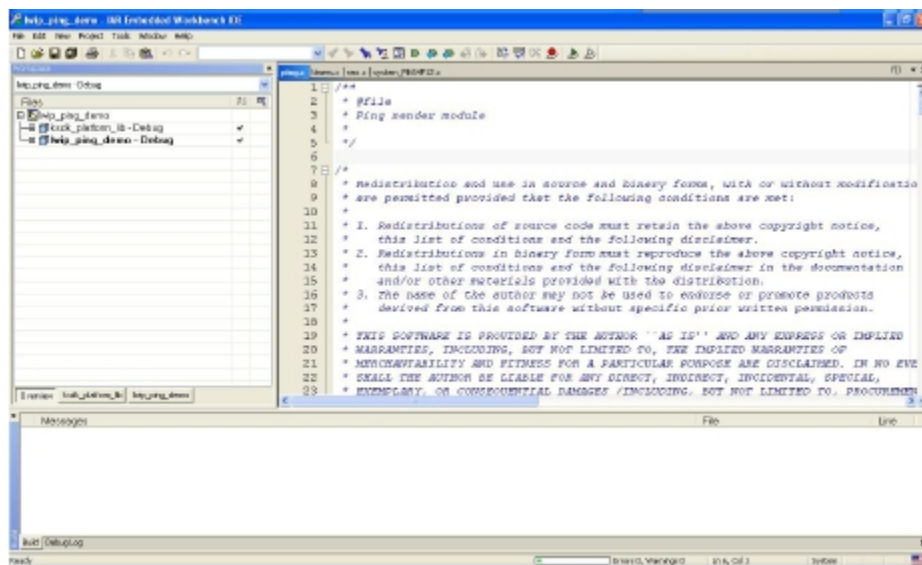


Figure 2. Workspace

2. Build the `ksdk_platform_lib` library.

Compiling or Running the lwIP Stack and Demos

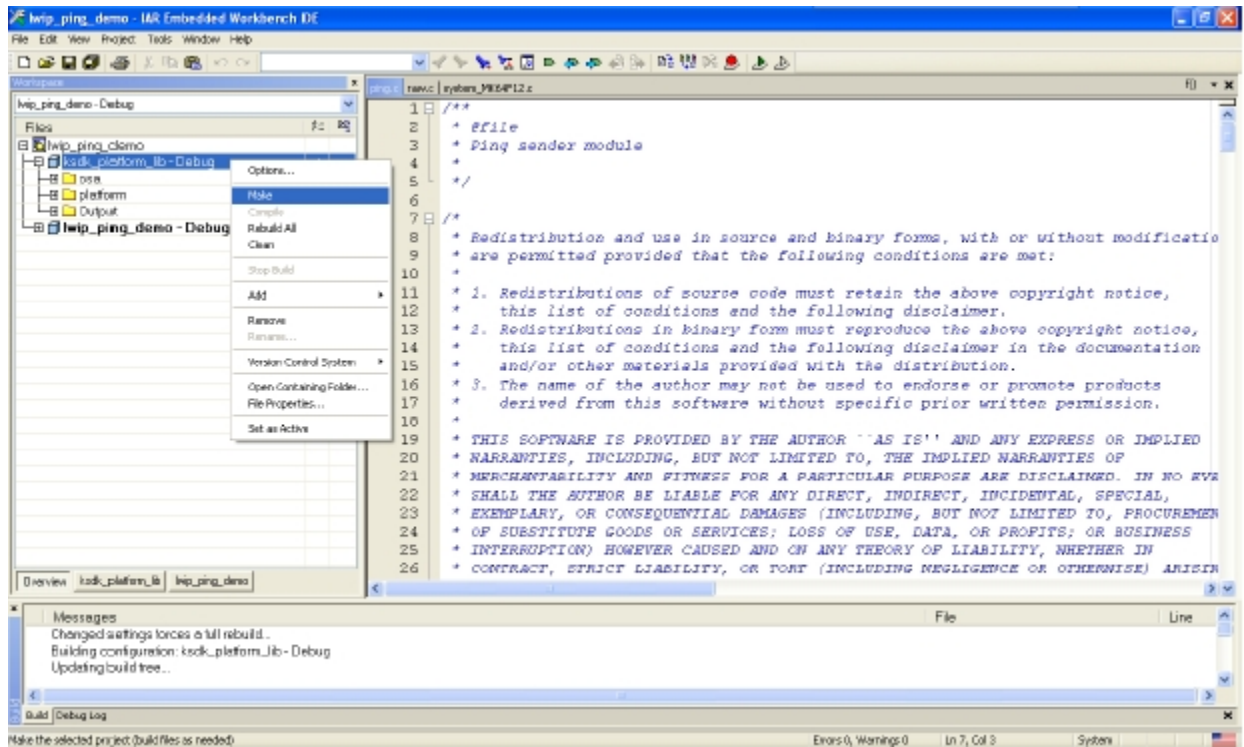


Figure 3. ksdk_platform_lib

3. Build the lwip_ping_demo.

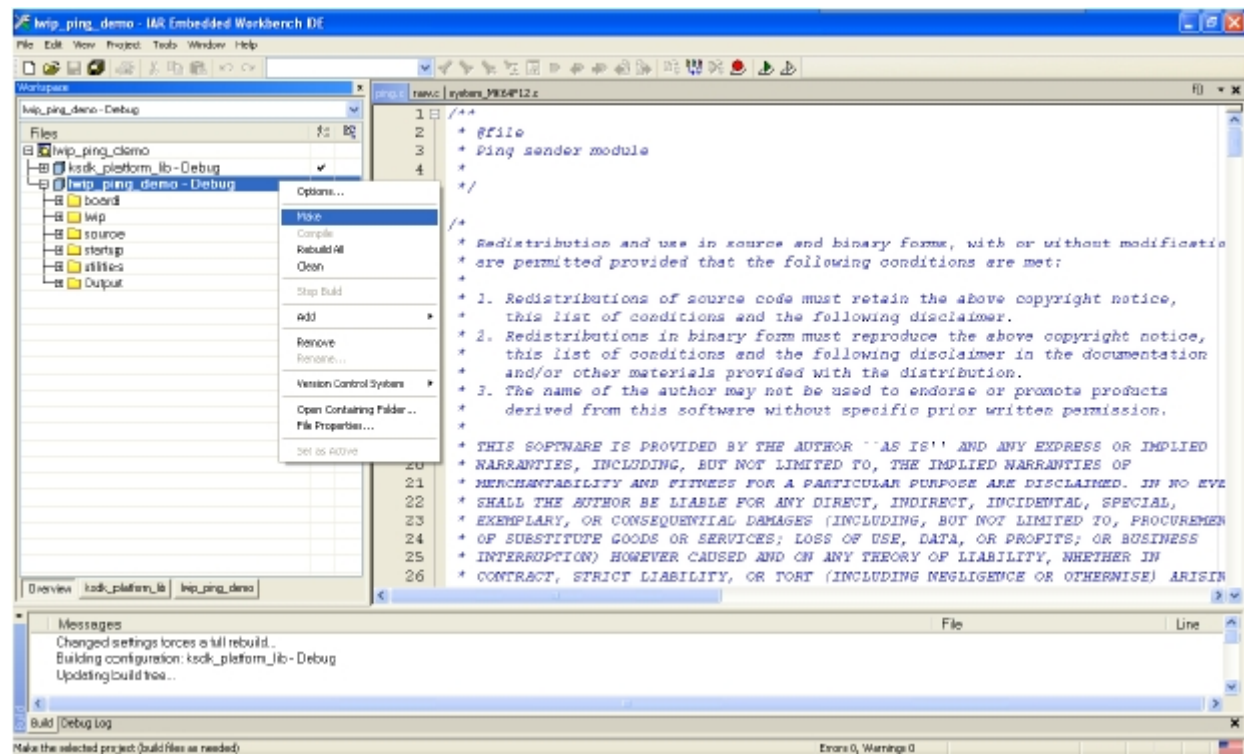


Figure 4. lwip_ping_demo

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

5.2 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 the Freescale Freedom FRDM-K64F platform under `<install_dir>/boards/frdmk64f/demo_apps/lwip/lwip_ping/bm/mdk/` or the `lwip_ping_demo_freertos.uvmpw` on the Freescale Freedom FRDM-K64F platform under `<install_dir>/boards/frdmk64f/demo_apps/lwip/lwip_ping/freertos/mdk/`. These steps take `lwip_ping_demo.uvmpw` on the Freescale Freedom FRDM-K64F platform for an example.

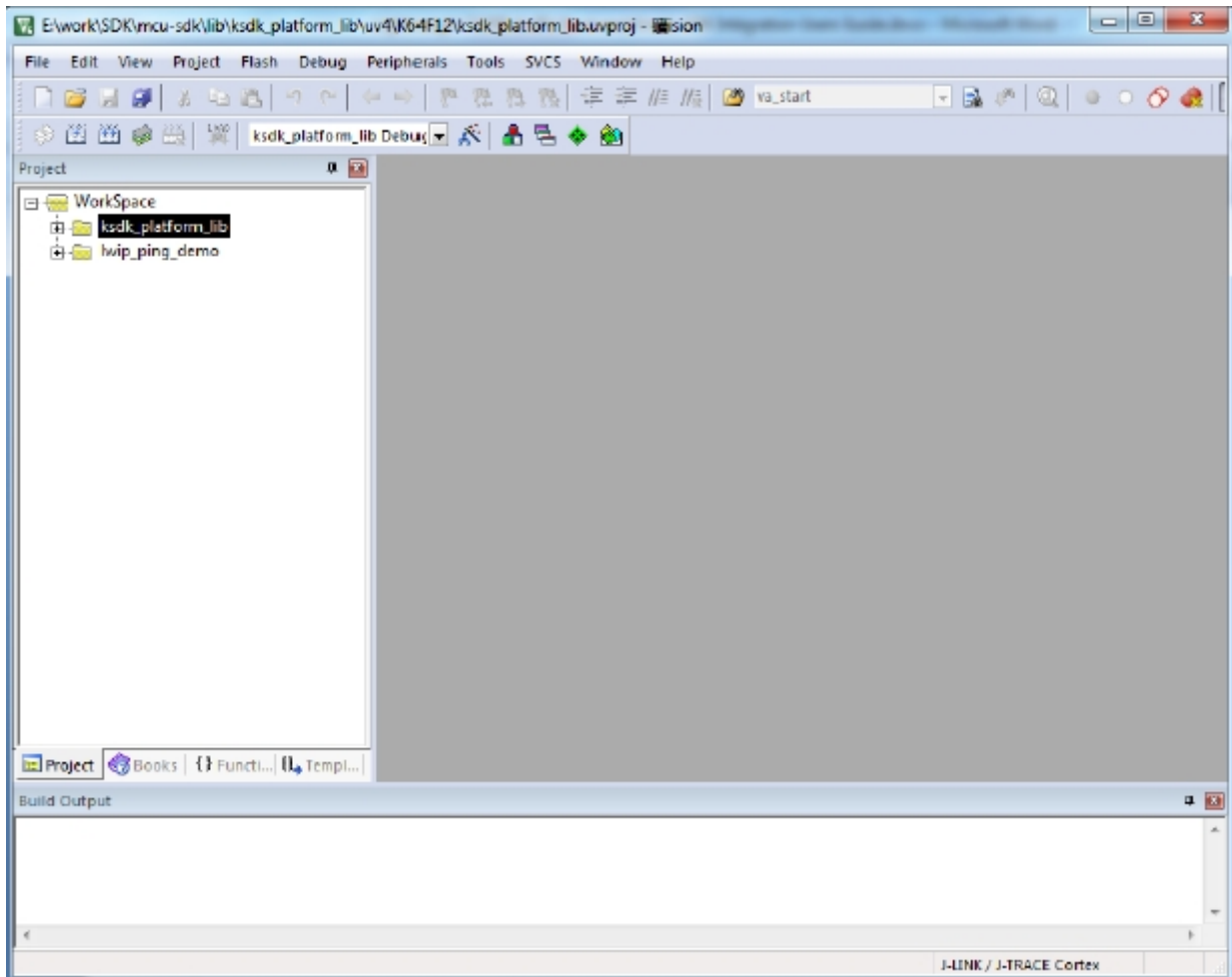


Figure 5. Workspace

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.

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

Compiling or Running the lwIP Stack and Demos

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>/boards/frdmk64f/demo_apps/lwip/lwip_ping/bm/kds/; or lwip_ping_demo_freertos on Freescale Freedom FRDM-K64F platform under <install_dir>/boards/frdmk64f/demo_apps/lwip/lwip_ping/freertos/kds/.

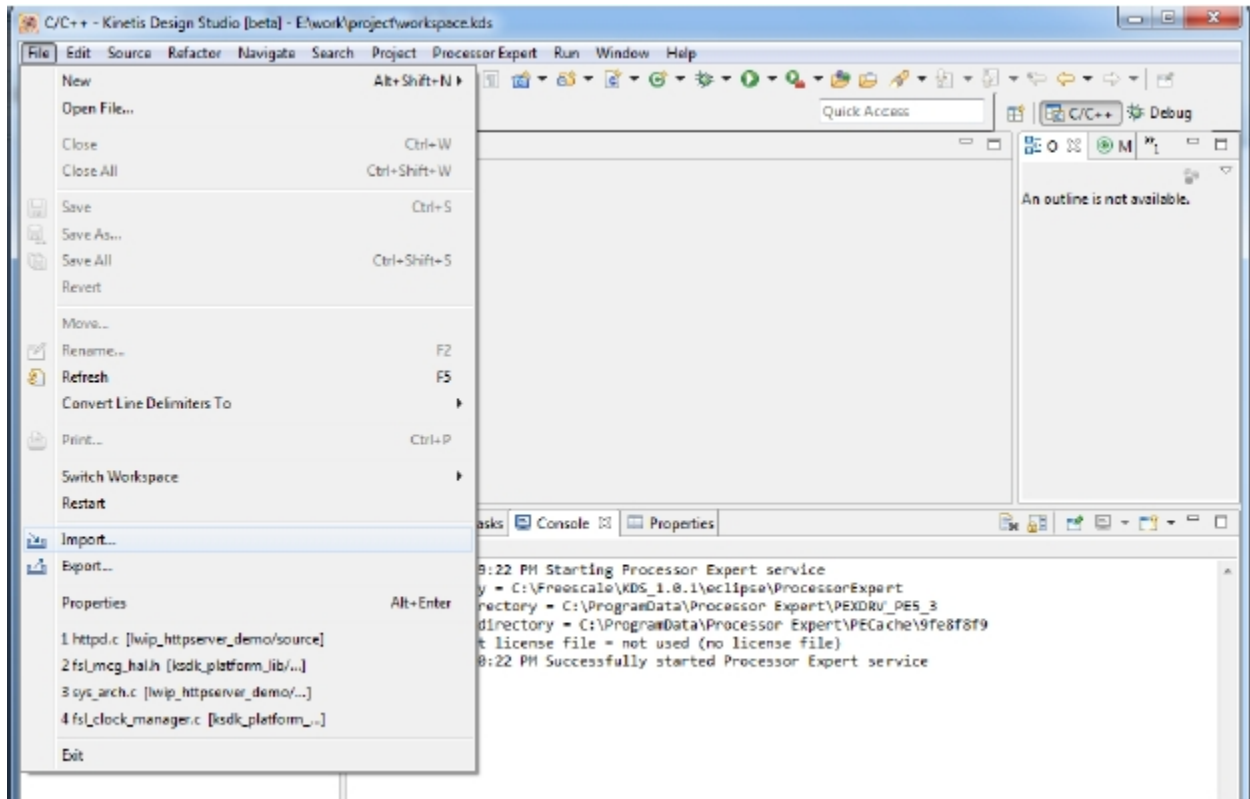


Figure 6. Import project

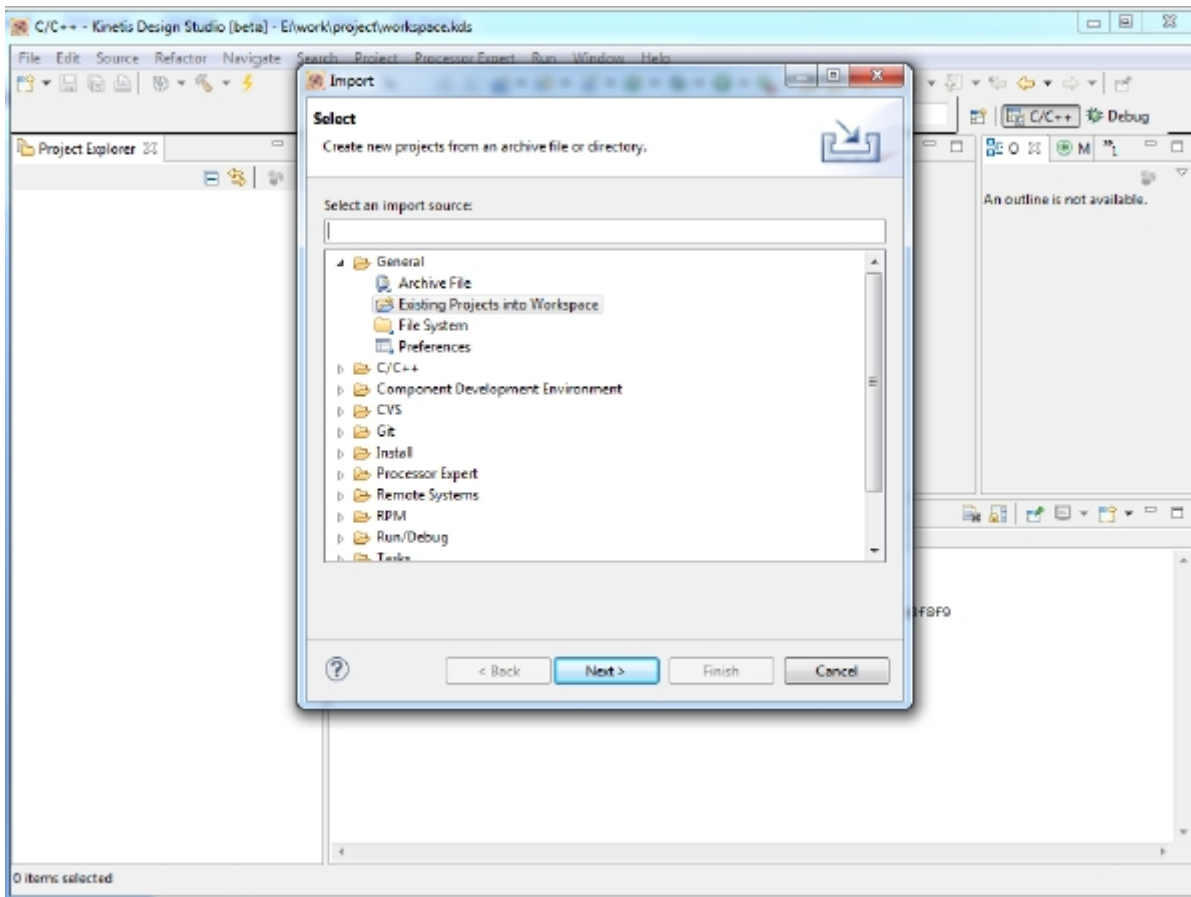


Figure 7. Import project select

Compiling or Running the lwIP Stack and Demos

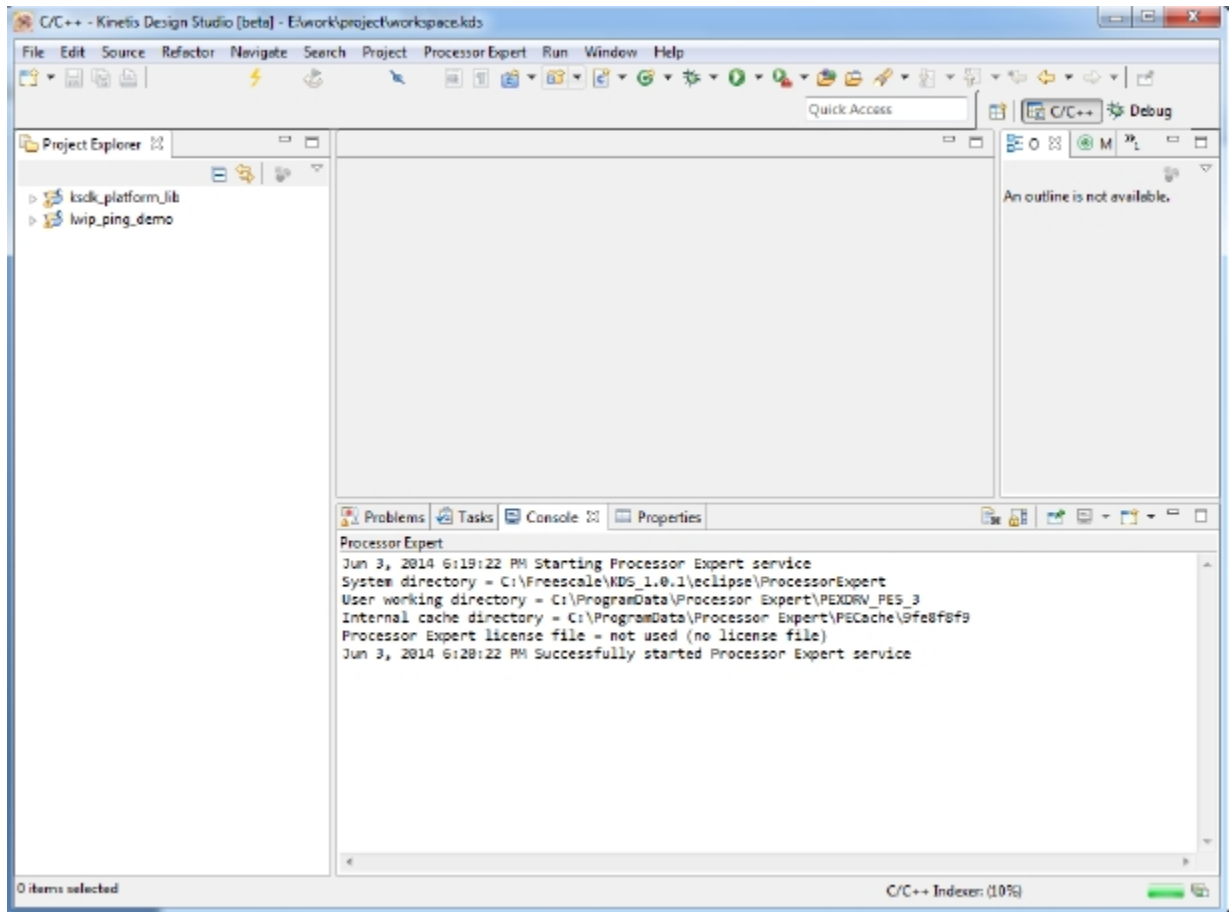


Figure 8. Demo project

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

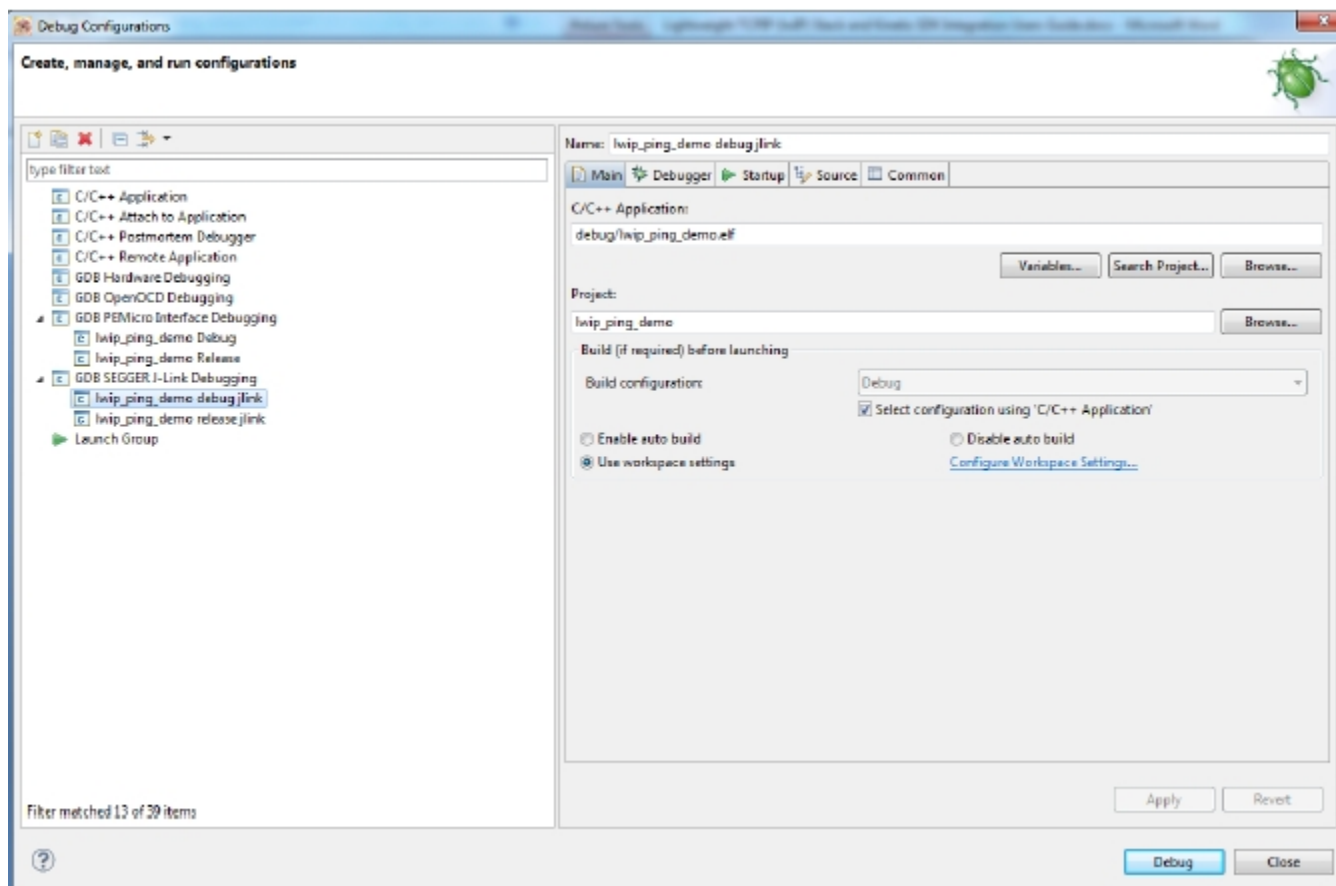


Figure 9. Debug Configurations

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

5.4 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. Change to the demo directory. For example: <install_dir>/boards/frdmk64f/demo_apps/lwip/lwip_ping/bm/armgcc
3. Run build_all.bat to build both debug and release projects.
4. 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.
3	11/2015	Updated Section 1
4	01/2016	Updated Section 1 and Section 3.2

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, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners. ARM, ARM powered logo, Keil, μ Vision, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. mbed is a trademark of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2016 Freescale Semiconductor, Inc.

Document Number KSDKLWIPUG
Revision 4, 01/2016

