

# **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-K60D100M, TWR-K64F120M, and TWR-K65F180M Tower System modules and the Freescale Freedom FRDM-K64F platform.

## **Contents**

1	Overview .....	1
2	Release scope.....	2
3	Requirements for running lwIP demos .....	2
4	lwIP code structure.....	3
5	Compiling or running the lwIP stack and demos .....	4
6	Revision history .....	14

## 2 Release scope

### 2.1 Hardware

- Support for TWR-K64F120M, TWR-K65F180M, and TWR-K60D100M Tower System module and Freescale Freedom FRDM-K64F 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<sup>®</sup> version 7.40.2
- Keil<sup>®</sup>  $\mu$ Vision<sup>®</sup> 5 Integrated Development Environment Version 5.14 service pack for Kinetis K60
- Kinetis Design Studio IDE Version: 3.0
- Makefiles support with GCC revision 4.8-2014-q3-update from ARM Embedded
- Atollic<sup>®</sup> TrueSTUDIO<sup>®</sup> 5.3.0

### 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-K60D100M
  - J10 2-3: Use the external clock from the CLOCKIN0 to keep the synchronization with the external PHY on TWR-SER board.
- TWR-K64F120M
  - J32 1-2: Use the external clock from the CLOCKIN0 to keep the synchronization with the external PHY on TWR-SER board.
- 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 4-1 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/](http://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/inc/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>/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/twrk64f120m/demo_apps/lwip/lwip_ping_demo/ping_rtos/iar/`

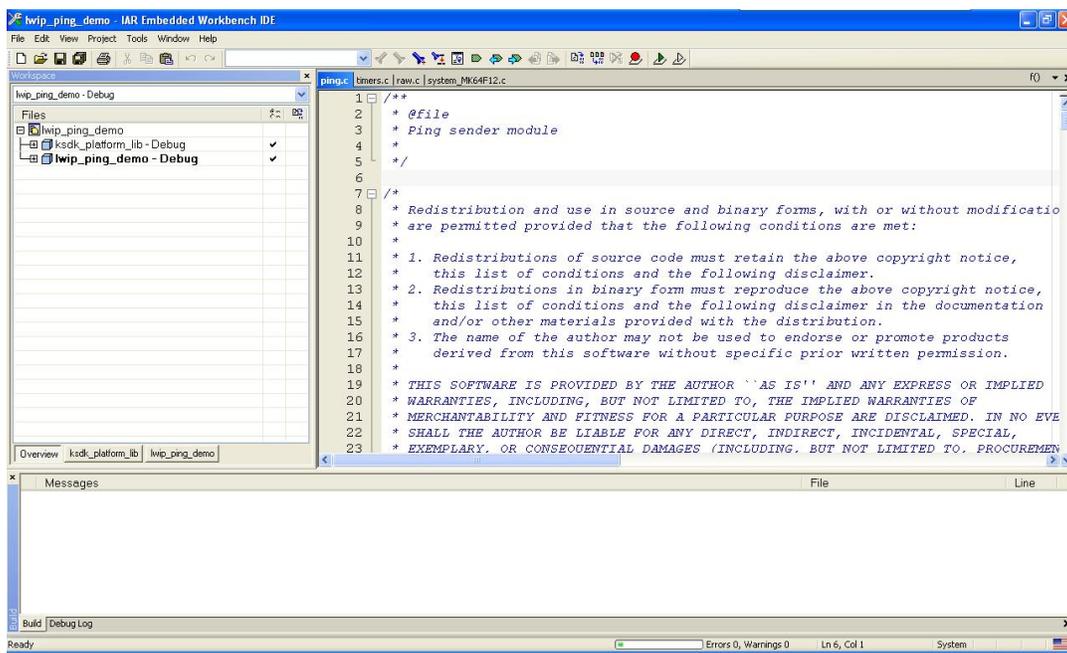


Figure 5-1 Workspace

## 2. Build the ksdk\_platform\_lib library.

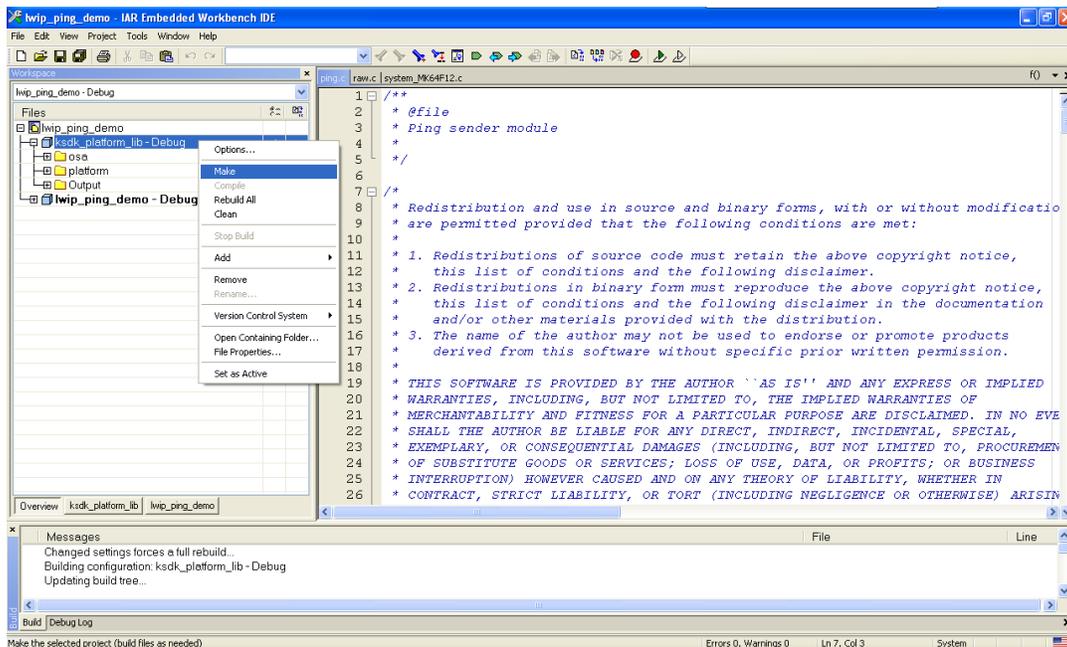


Figure 5-2 ksdk\_platform\_lib

## 3. Build the lwip\_ping\_demo.

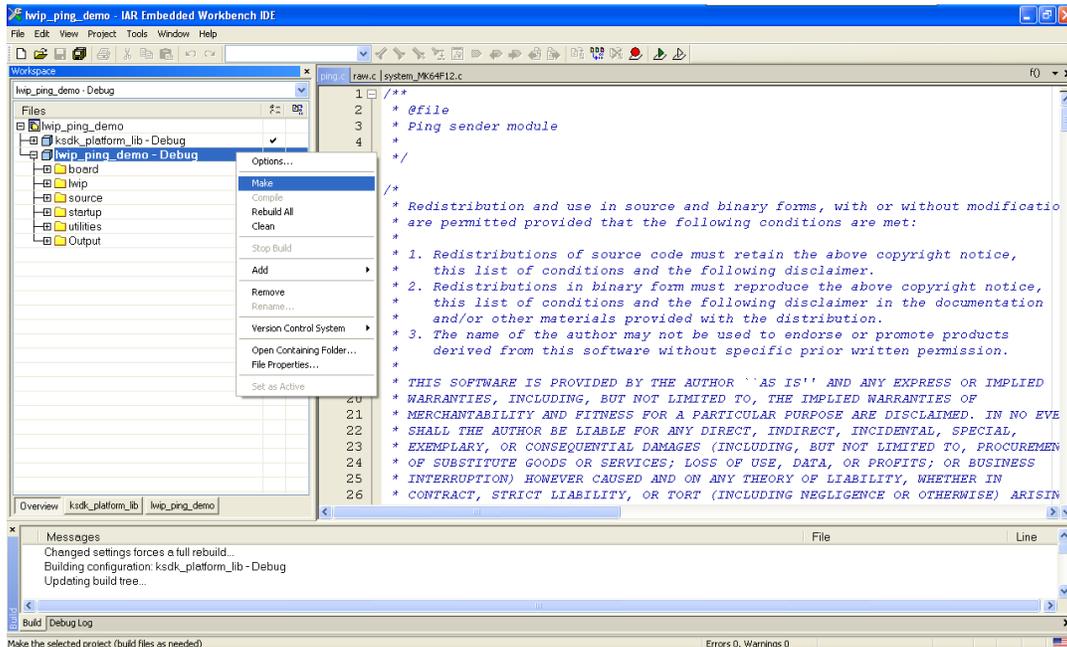


Figure 5-3 lwip\_ping\_demo

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>/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.

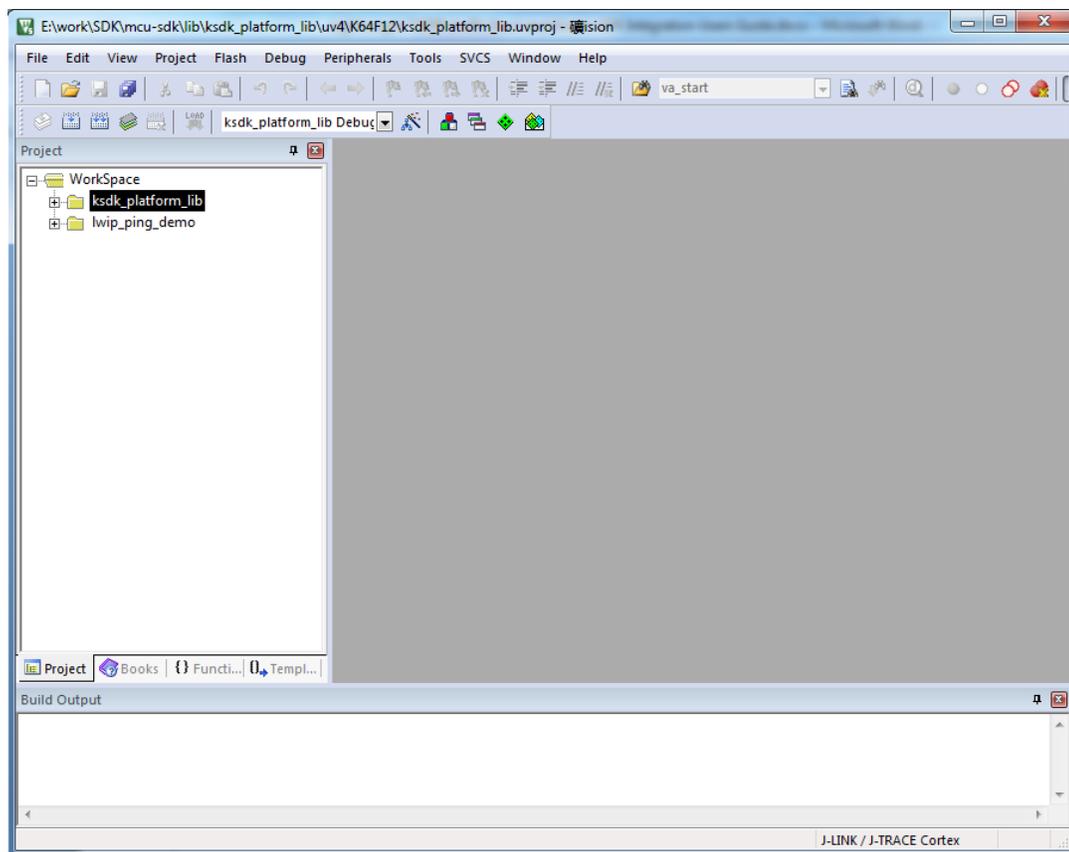


Figure 5-4 Workspace

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 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.

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

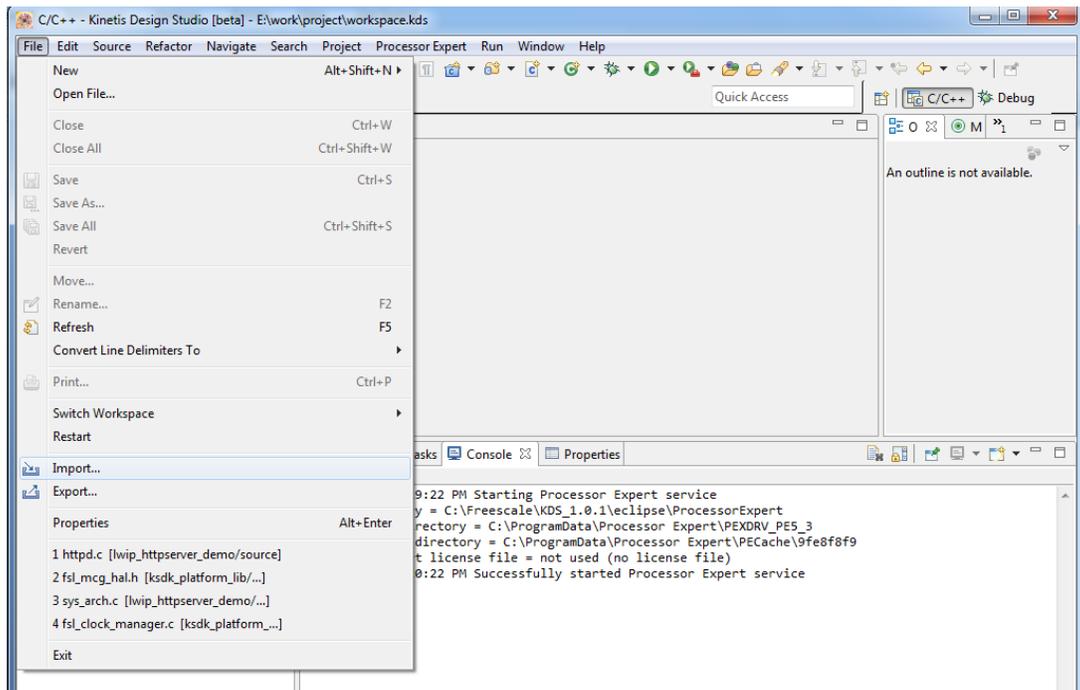
`<install_dir>/lib/ksdk_freertos_lib/kds/K64F12`

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/.`

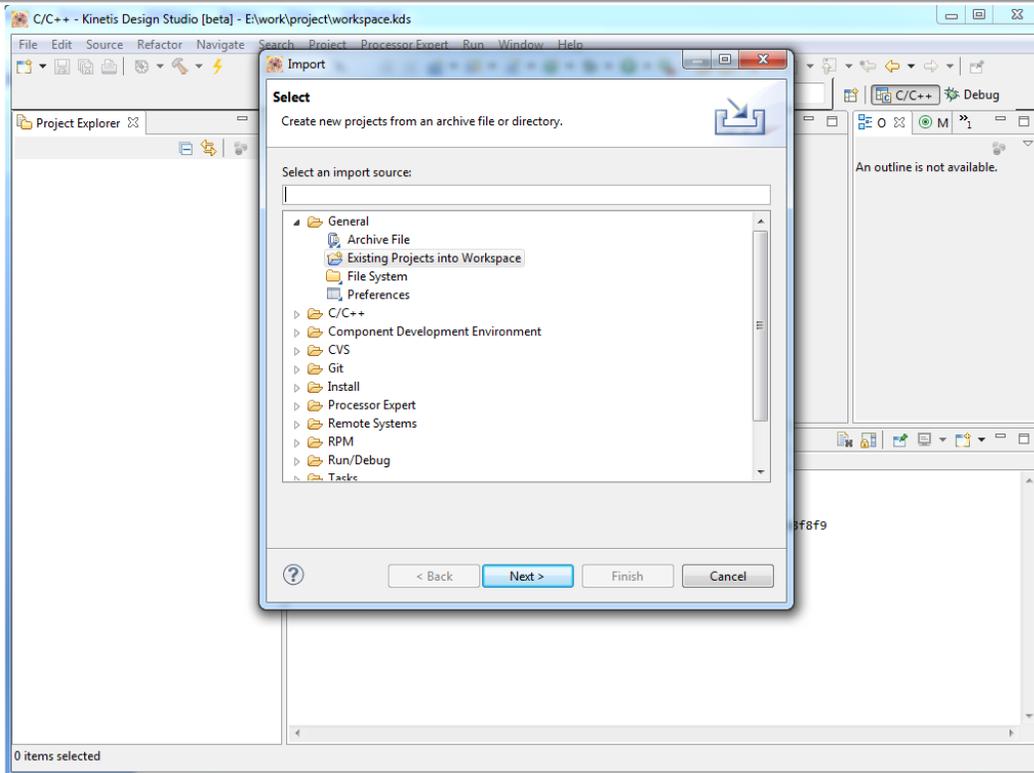
### Note

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`.

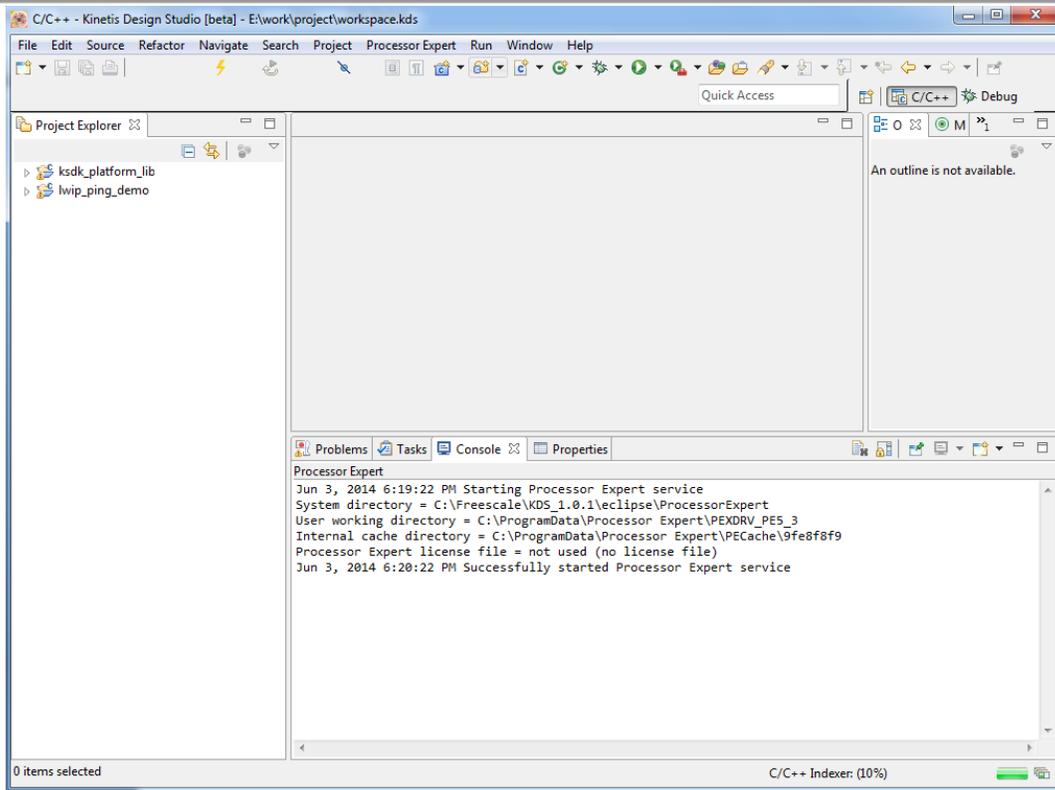


---

**Figure 5-5 Import project -1**

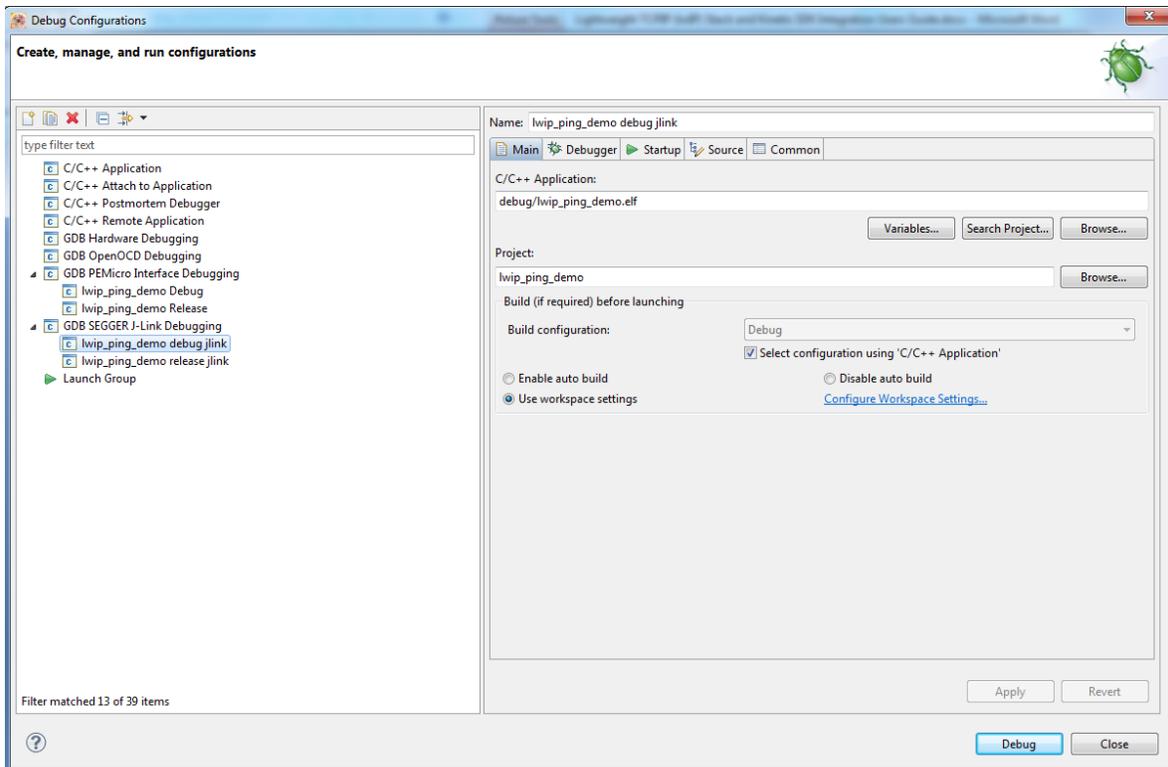


**Figure 5-6 Import project - 2**



**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 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 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 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.

<b>Table 1 Revision History</b>		
<b>Revision number</b>	<b>Date</b>	<b>Substantial changes</b>
1	04/2015	Kinetis SDK 1.2.0 release
0	12/2014	Kinetis SDK 1.1.0 release

---

**How to Reach Us:**

**Home Page:**

[freescale.com](http://freescale.com)

**Web Support:**

[freescale.com/support](http://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](http://freescale.com/SalesTermsandConditions).

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

© 2015 Freescale Semiconductor, Inc.

