

Getting Started with MCUXpresso SDK for MC56F83000-EVK



Contents

Chapter 1 Overview.....	3
Chapter 2 Run a demo application using CodeWarrior.....	4
2.1 Build an example application.....	4
2.2 Run an example application.....	7
Chapter 3 Project template.....	11
3.1 Project name.....	11
3.2 Peripheral driver.....	11
3.3 MCUXpresso config tool support.....	11
Appendix A How to determine COM port.....	12

Chapter 1

Overview

The MCUXpresso Software Development Kit (SDK) provides comprehensive software support for Kinetis, LPC, and DSC Microcontrollers. The MCUXpresso SDK includes a flexible set of peripheral drivers designed to speed up and simplify development of embedded applications. Along with the peripheral drivers, the MCUXpresso SDK provides an extensive and rich set of example applications covering everything from basic peripheral use case examples to full demo applications. The MCUXpresso SDK contains various middleware to support rapid development.

For supported toolchain versions, see *MCUXpresso SDK Release Notes for MC56F83000-EVK* (document MCUXSDKMC56F83000RN).

For more details about MCUXpresso SDK, see [MCUXpresso-SDK](#).

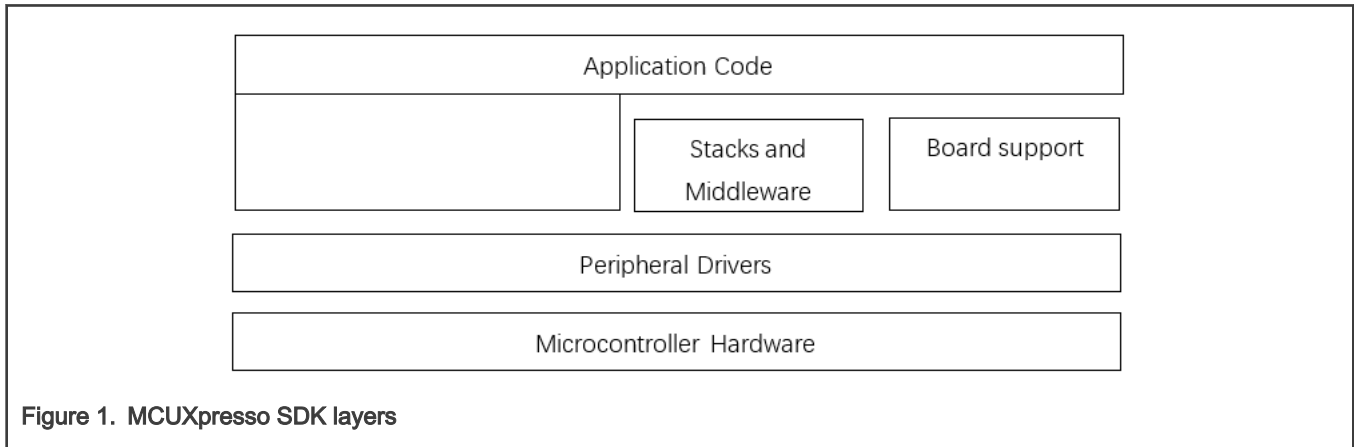


Figure 1. MCUXpresso SDK layers

Chapter 2

Run a demo application using CodeWarrior

This section describes the steps required to build, run, and debug example applications provided in the MCUXpresso SDK.

NOTE

CodeWarrior IDE version 11.1 with update 3 is used as an example to show below steps, and the DSC toolchain should correspond to the latest supported version, as described in *MCUXpresso SDK Release Notes for MC56F83000-EVK* (document MCUXSDKM56F83000RN).

CodeWarrior 11.1 for DSC can be downloaded from [CodeWarrior® for MCUs](#).

Two options to install update 3:

1. Online

Open Codewarrior, click **help** -> **Install New Software**. The settings are as shown in [Figure 2](#). Click **Next** to install.

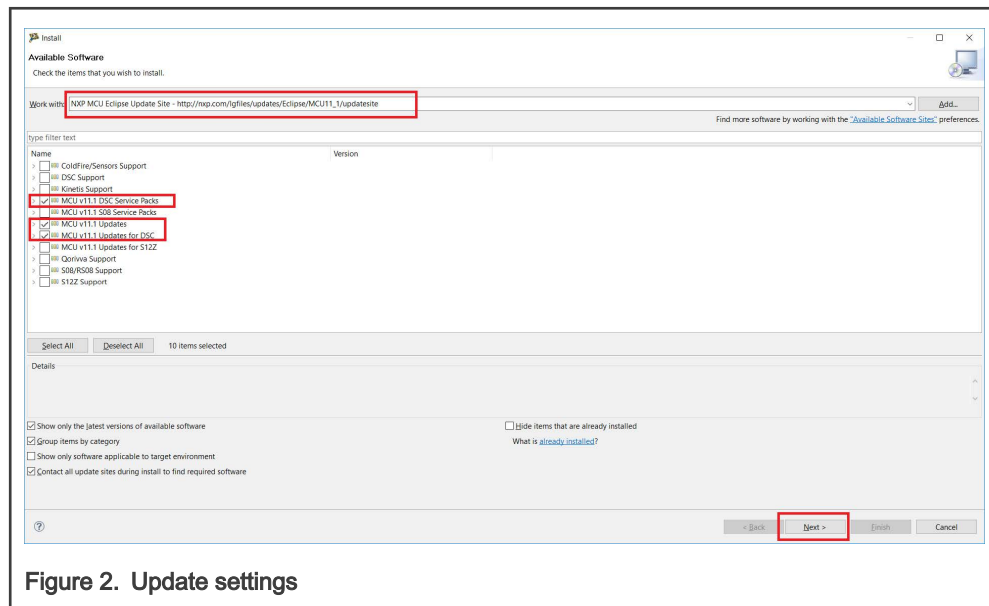


Figure 2. Update settings

2. Offline

Download (via above link) CodeWarrior for MCU 11.1 Update 3 (download the *CW_MCU_11_1_Update3_200717.zip* file).

For details about how to download and install a package, see [Install CodeWarrior 11.1 Updates/ Updates for DSC + MC56F83xxx device support](#).

2.1 Build an example application

Perform the following steps to build the `hello_world` example application.

1. Launch CodeWarrior and you will be asked in the workspace launcher to choose a workspace which holds the projects you will use. Create one if you do not have by entering a workspace folder.

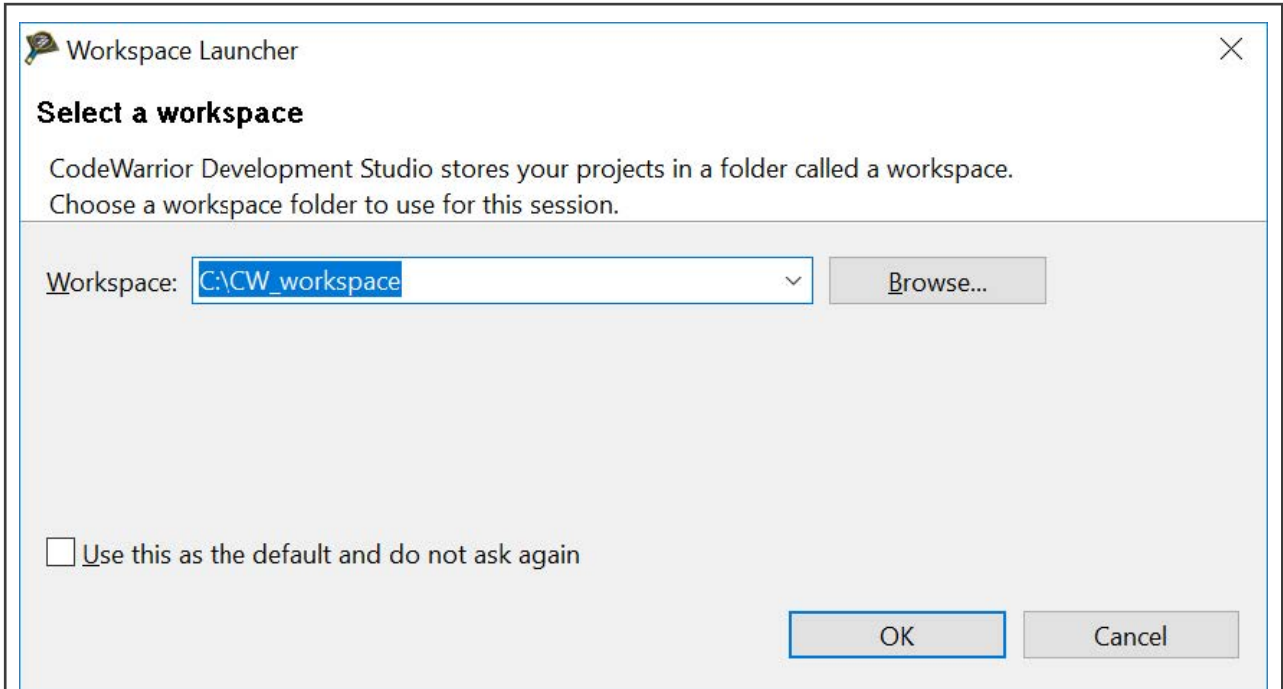


Figure 3. Workspace launcher view

Then the CodeWarrior Development Studio workspace with empty projects appears.

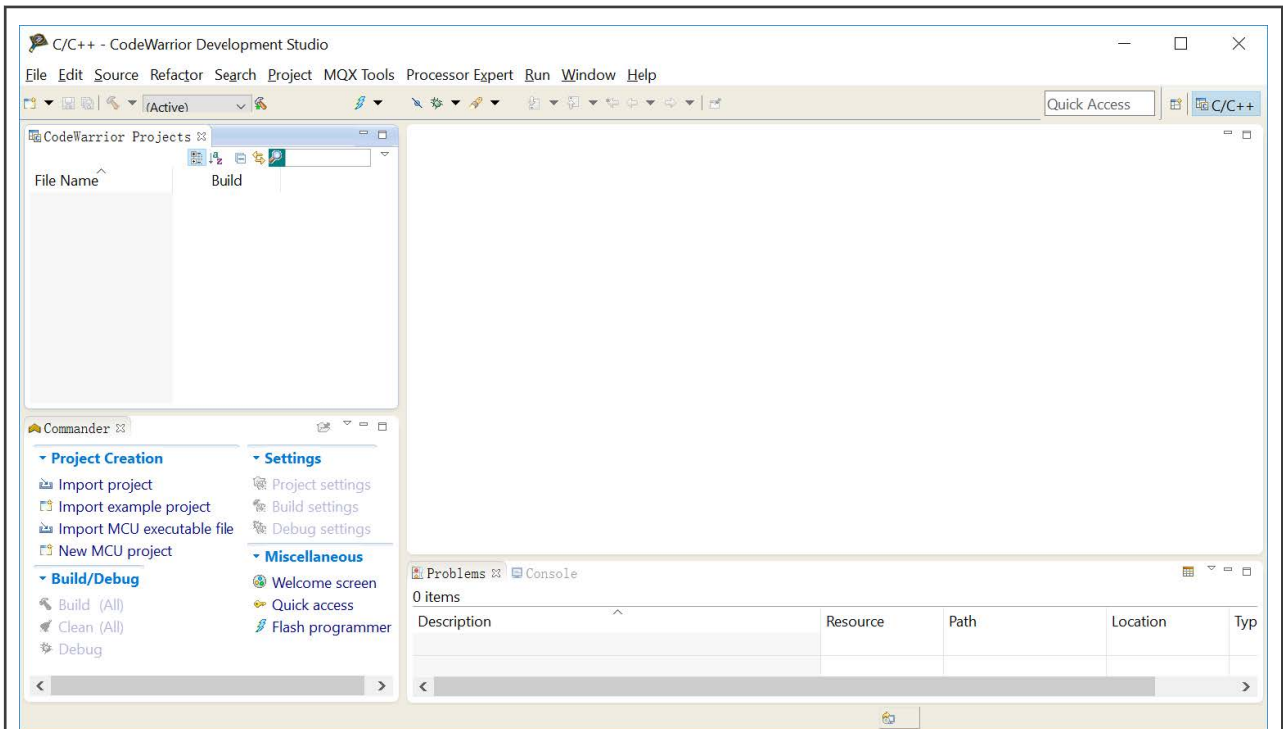


Figure 4. CodeWarrior development studio view

2. Import the project into the workspace.

Click **Import project** in the **Commander** pane. A form is prompted up, and click **Browse** to the SDK install directory.

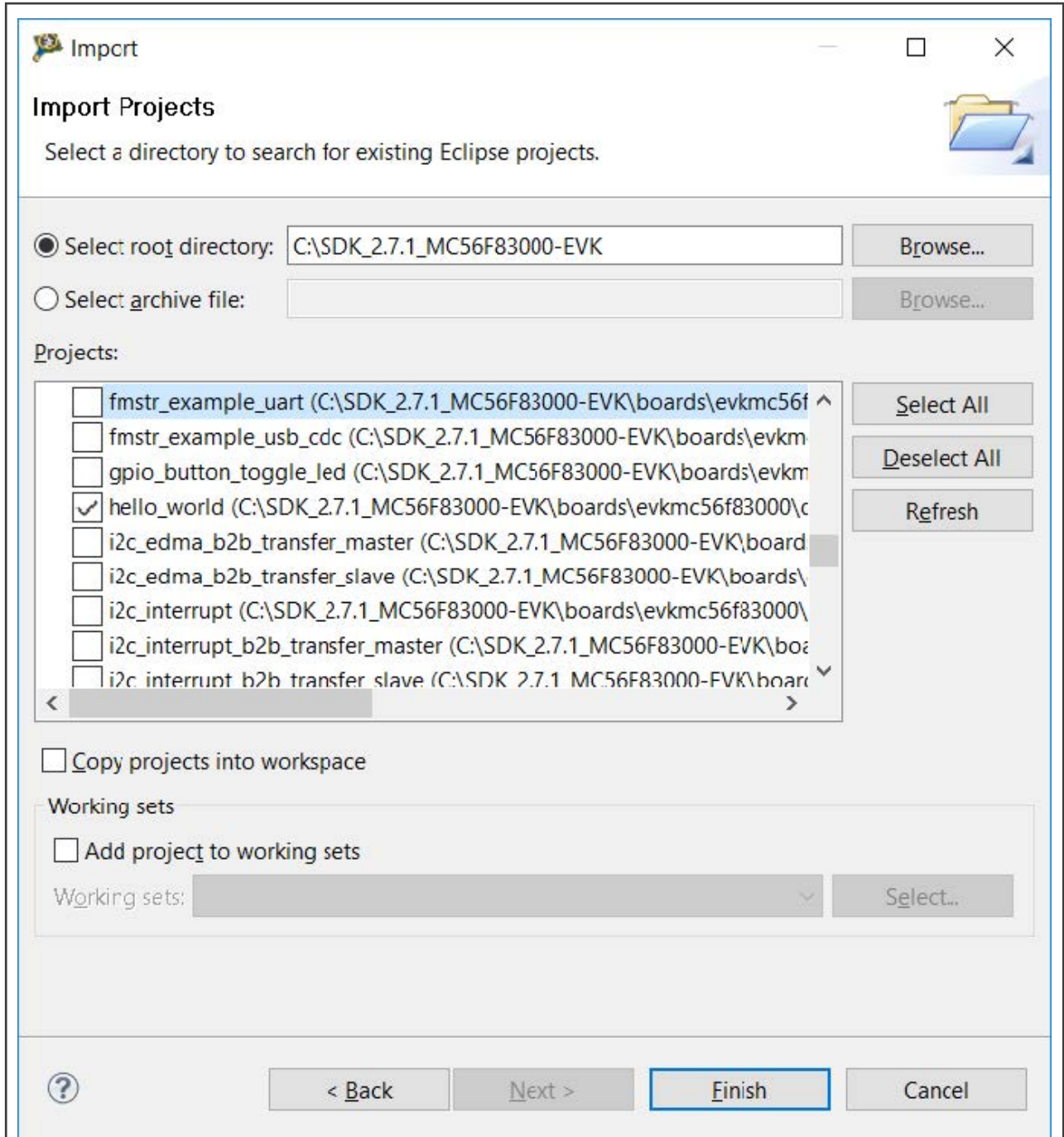


Figure 5. Import projects view

Then all available demo projects are as shown in [Figure 5](#). Select the `hello_world` project in the list, and click **Finish**.

If you already know the project location, you can directly navigate to the folder when click **Browse**, so that you will only see one project.

Most example application workspace files can be located using the following path:

```
<install_dir>/boards/<board_name>/<example_type>/<application_name>/codewarrior
```

Using the MC56F83000-EVK hardware platform as an example, the `hello_world` workspace is located in:

```
<install_dir>/boards/evkmc56f83000/demo_apps/hello_world/codewarrior
```

3. Select the desired build target from the drop-down menu. For this example, select **hello_world – flash_ldm_lpm_debug**, as shown in [Figure 6](#).

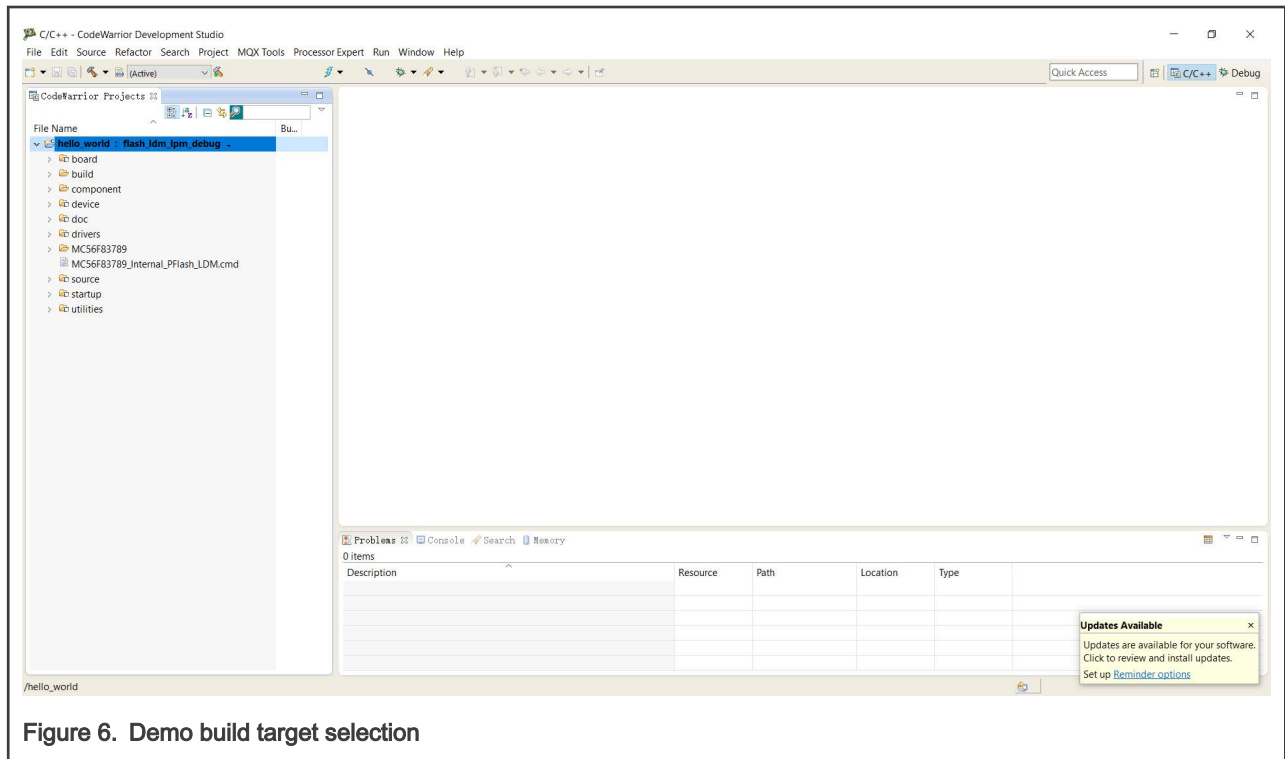


Figure 6. Demo build target selection

4. To build the demo application, click **Build (All)** in the **Commander** pane.
5. The build completes without errors.

2.2 Run an example application

Perform the following steps to download and run the application.

1. Connect the MC56F83000-EVK JM60 USB port, **J8**, to your PC via USB cable within the board package. This USB port is used for on board OSJTAG debugger and USB to UART bridge.
2. Install the OSJTAG driver and USB CDC driver as PC hint if it is the first time you run it on your PC. The OSJTAG and USB CDC driver are provided by Codewarrior by default.

The Codewarrior may prompt to update the JM60 firmware, which requires to connect the **J6** with a jumper on MC56F83000-EVK board and then follow the instruction by Codewarrior to finish the firmware update. The default debug interface is OSJTAG for MC56F83000-EVK board.

3. Open the terminal application on the PC, such as PuTTY or TeraTerm, and connect to the debug COM port (to determine the COM port number, see [How to determine COM port](#)). Configure the terminal with these settings:
 - a. 115200, defined by `BOARD_DEBUG_UART_BAUDRATE` in the `board.h` file
 - b. No parity
 - c. 8 data bits
 - d. 1 stop bit

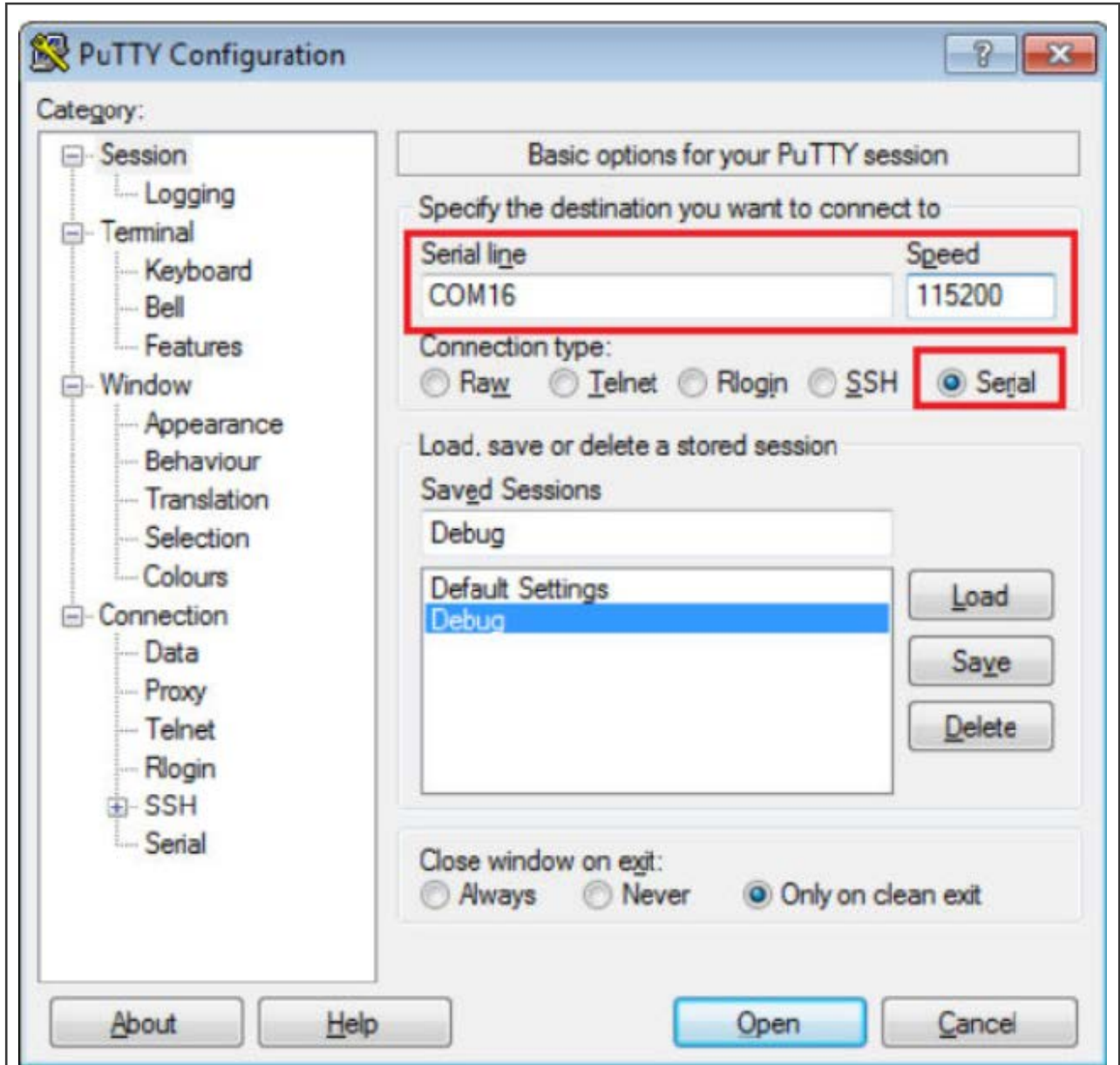


Figure 7. Terminal (PuTTY) configuration

4. For this example, click **Debug** in the **Commander** pane, and select the `hello_world_flash_ldm_lpm_debug_OSJTAG` launch configuration.

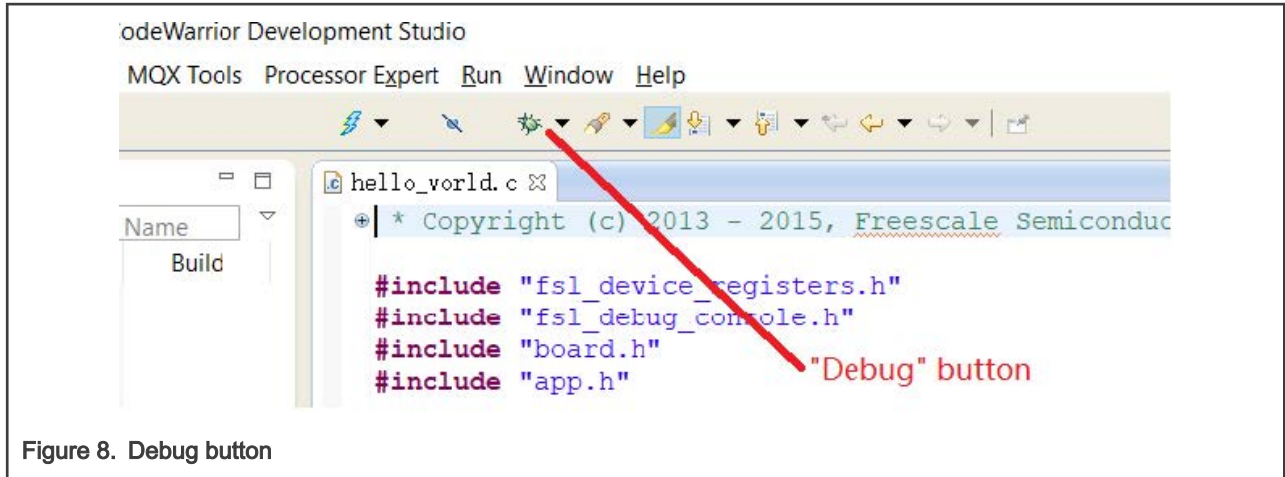


Figure 8. Debug button

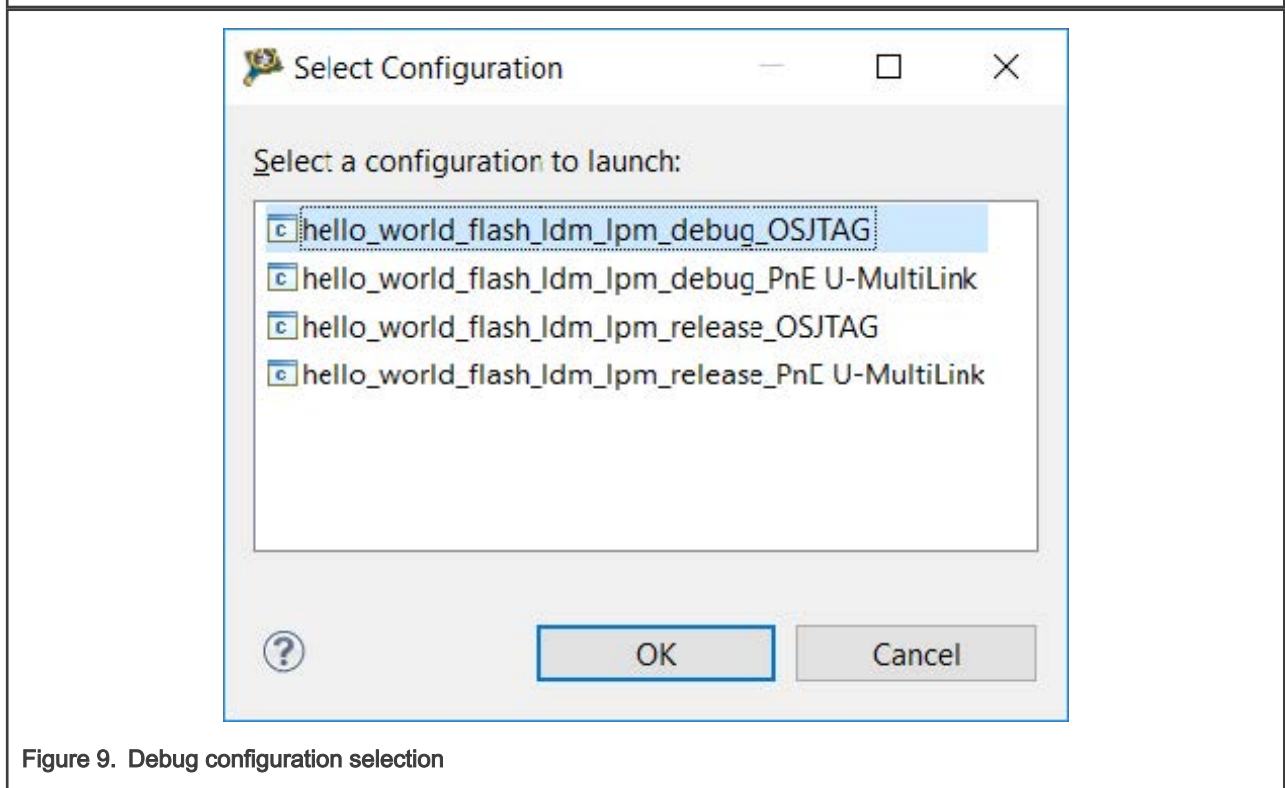


Figure 9. Debug configuration selection

Then the application is downloaded onto target board and automatically runs to the `main()` function.

Click **Run** on the toolbar to run the code.

NOTE

There are two configurations for each SDK project: `flash_ldm_lpm_debug` and `flash_ldm_lpm_release`. `flash_ldm_lpm_debug` uses optimization level 1 and `flash_ldm_lpm_release` uses optimization level 4. When it comes to debug configuration selection, as shown in [Figure 9](#), use the configuration with word **debug** for `flash_ldm_lpm_debug` and the configuration with word **release** for `flash_ldm_lpm_release`.

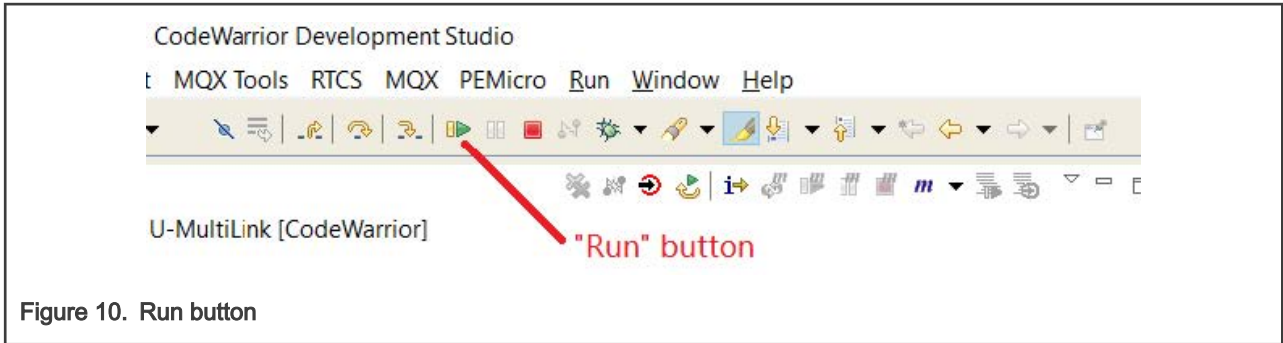


Figure 10. Run button

- 5. The `hello_world` application is now running and a banner is displayed on the terminal, as shown in Figure 11. If it does not appear, check your terminal settings and connections.



Figure 11. Text display of the `hello_world` demo

Chapter 3

Project template

For device with specific part number, the easiest way to set up customer own project based on MCUXpresso DSC SDK peripheral driver, is the project template. The project template package for each MC56F83xxx part number could be downloaded from [MC56F83xxx](#).

The project template provides basic MCUXpresso DSC SDK software framework, including startup, linker file, device header file, debug setting, peripheral driver, FreeMASTER, etc.

3.1 Project name

Codewarrior can import the project template for each part number directly. The default Codewarrior project name is `project_template_{part number}` and customers can change the project name in Codewarrior.

3.2 Peripheral driver

All peripheral drivers files are included in the project template, they are same as the peripheral drivers within SDK package.

If some drivers are not used or required, users may delete them in Codewarrior, or delete them in the `project_template_MC56F83xxx/drivers` folder.

3.3 MCUXpresso config tool support

The project template supports the MCUXpresso Configl Tool. The configuration file, `project_template_{part number}.mex`, under the project template folder, is for MCUXpresso Config Tool usage.

Appendix A

How to determine COM port

This section describes the steps necessary to determine the debug COM port number of your NXP hardware development platform.

1. **Linux:** The serial port can be determined by running the following command after the USB Serial is connected to the host:

```
$ dmesg | grep "ttyUSB"
[503175.307873] usb 3-12: cp210x converter now attached to ttyUSB0
[503175.309372] usb 3-12: cp210x converter now attached to ttyUSB1
```

There are two ports, one is Cortex-A core debug console and the other is for Cortex M4.

2. **Windows:** To determine the COM port open Device Manager in the Windows operating system. Click on the **Start** menu and type **Device Manager** in the search bar.

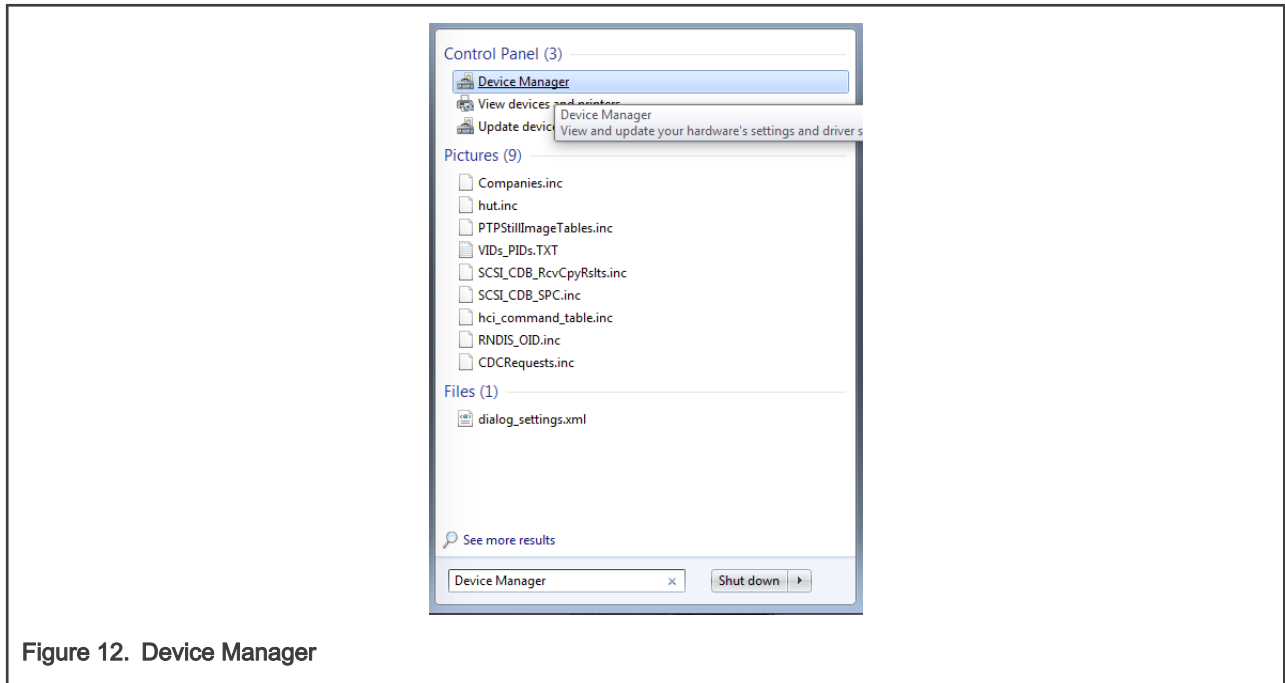


Figure 12. Device Manager

3. In the Device Manager, expand the **Ports (COM & LPT)** section to view the available ports. The COM port names will be different for all the NXP boards.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP 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. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP 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 NXP 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. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QoriQ, QoriQ Converge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, UMEMS, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μ Vision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2020.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: November 2020

Document identifier: MCUXSDKMC58F83000GSUG

