# MCUXSDKLPC551XGSUG Getting Started with MCUXpresso SDK for LPCXpresso55S16

Rev. 2.10.1 — 09 September 2021

User Guide

## 1 Overview

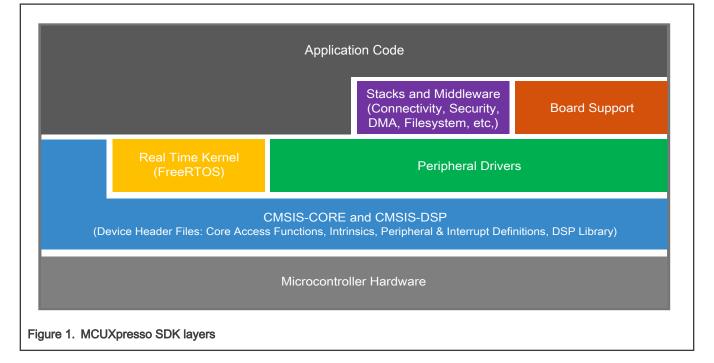
The NXP MCUXpresso software and tools offer comprehensive development solutions designed to optimize, ease and help accelerate embedded system development of applications based on general purpose, crossover and Bluetooth<sup>™</sup>-enabled MCUs from NXP. The MCUXpresso SDK includes a flexible set of 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 optional RTOS integrations such as FreeRTOS and Azure RTOS, a USB host and device stack, and various other middleware to support rapid development.

For supported toolchain versions, see *MCUXpresso SDK Release Notes* (document MCUXSDKRN). *MCUXpresso SDK Release Notes for LPCXpresso55S16* (document MCUXSDKLPC551XRN).

For more details about MCUXpresso SDK, see MCUXpresso Software Development Kit (SDK).

#### Contents

1	Overview1
2	MCUXpresso SDK board support
	package folders2
3	Run a demo using MCUXpresso IDE
4	Run a demo application using IAR24
5	Run a demo using Keil <sup>®</sup> MDK/µVision
6	Run a demo using Arm GCC35
7	MCUXpresso Config Tools50
8	MCUXpresso IDE New Project
	Wizard50
9	How to determine COM port51
10	How to define IRQ handler in CPP
	files52
11	Default debug interfaces 53
12	Updating debugger firmware55
13	Revision history56





## 2 MCUXpresso SDK board support package folders

MCUXpresso SDK board support package provides example applications for NXP development and evaluation boards for Arm<sup>®</sup> Cortex<sup>®</sup>-M cores including Freedom, Tower System, and LPCXpresso boards. Board support packages are found inside the top level boards folder and each supported board has its own folder (an MCUXpresso SDK package can support multiple boards). Within each <br/>
<code>board\_name></code> folder, there are various sub-folders to classify the type of examples it contain. These include (but are not limited to):

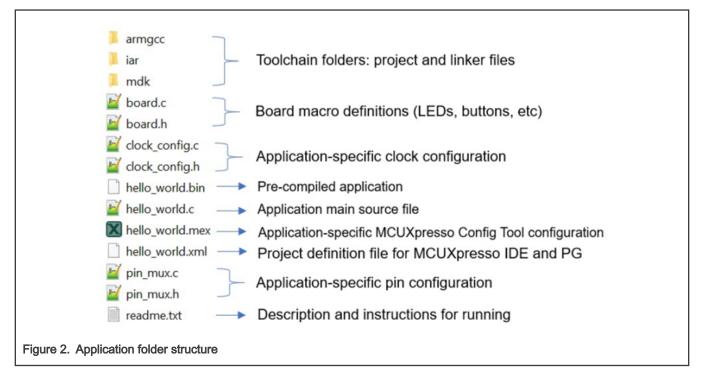
- cmsis driver examples: Simple applications intended to show how to use CMSIS drivers.
- demo\_apps: Full-featured applications that highlight key functionality and use cases of the target MCU. These applications typically use multiple MCU peripherals and may leverage stacks and middleware.
- driver\_examples: Simple applications that show how to use the MCUXpresso SDK's peripheral drivers for a single use case. These applications typically only use a single peripheral but there are cases where multiple peripherals are used (for example, SPI conversion using DMA).
- emwin\_examples: Applications that use the emWin GUI widgets.
- rtos\_examples: Basic FreeRTOS<sup>TM</sup> OS examples that show the use of various RTOS objects (semaphores, queues, and so on) and interfaces with the MCUXpresso SDK's RTOS drivers
- usb\_examples: Applications that use the USB host/device/OTG stack.

## 2.1 Example application structure

This section describes how the various types of example applications interact with the other components in the MCUXpresso SDK. To get a comprehensive understanding of all MCUXpresso SDK components and folder structure, see *MCUXpresso SDK API Reference Manual*.

Each <board\_name> folder in the boards directory contains a comprehensive set of examples that are relevant to that specific piece of hardware. Although we use the hello\_world example (part of the demo\_apps folder), the same general rules apply to any type of example in the <board\_name> folder.

In the hello\_world application folder you see the following contents:



All files in the application folder are specific to that example, so it is easy to copy and paste an existing example to start developing a custom application based on a project provided in the MCUXpresso SDK.

#### 2.2 Locating example application source files

When opening an example application in any of the supported IDEs, a variety of source files are referenced. The MCUXpresso SDK devices folder is the central component to all example applications. It means the examples reference the same source files and, if one of these files is modified, it could potentially impact the behavior of other examples.

The main areas of the MCUXpresso SDK tree used in all example applications are:

- devices/<device name>: The device's CMSIS header file, MCUXpresso SDK feature file and a few other files
- devices/<device name>/cmsis drivers: All the CMSIS drivers for your specific MCU
- devices/<device\_name>/drivers: All of the peripheral drivers for your specific MCU
- devices/<device\_name>/<tool\_name>: Toolchain-specific startup code, including vector table definitions
- devices/<device\_name>/utilities: Items such as the debug console that are used by many of the example applications
- devices/<devices\_name>/project: Project template used in CMSIS PACK new project creation

For examples containing middleware/stacks or an RTOS, there are references to the appropriate source code. Middleware source files are located in the middleware folder and RTOSes are in the rtos folder. The core files of each of these are shared, so modifying one could have potential impacts on other projects that depend on that file.

## 3 Run a demo using MCUXpresso IDE

NOTE

Ensure that the MCUXpresso IDE toolchain is included when generating the MCUXpresso SDK package.

This section describes the steps required to configure MCUXpresso IDE to build, run, and debug example applications. The hello\_world demo application targeted for the LPCXpresso55S16 hardware platform is used as an example, though these steps can be applied to any example application in the MCUXpresso SDK.

#### 3.1 Select the workspace location

Every time MCUXpresso IDE launches, it prompts the user to select a workspace location. MCUXpresso IDE is built on top of Eclipse which uses workspace to store information about its current configuration, and in some use cases, source files for the projects are in the workspace. The location of the workspace can be anywhere, but it is recommended that the workspace be located outside of the MCUXpresso SDK tree.

#### 3.2 Build an example application

To build an example application, follow these steps.

1. Drag and drop the SDK zip file into the **Installed SDKs** view to install an SDK. In the window that appears, click **OK** and wait until the import has finished.

ල්) Installed SDKs ස	📄 Properties 😑 Console	Problems (	Memory	🚯 Instruction
To install an SDK, simpl	y drag and drop an SDK (zip		ne 'Installed S	SDKs' view.
Name	Version	Location		
Figure 3. Install an SDK				

2. On the Quickstart Panel, click Import SDK example(s)....

<ul> <li>MCUXpresso IDE - Quickstart Panel No project selected</li> <li>Create or import a project</li> <li>New project.</li> <li>Import SDK example(s)</li> <li>Build your project (s) from file system</li> <li>Build your project</li> <li>Clean</li> <li>Debug</li> <li>Terminate, Build and Debug</li> <li>Miscellaneous</li> <li>Gdtt project seltings</li> <li>Quick Settings&gt;</li> <li>Export project(s) to archive (p)</li> <li>Export project(s) to archive (p)</li> <li>Export project(s) to archive (p)</li> <li>Build and references to archive (p)</li> <li>Build and references to archive (p)</li> <li>Build and references to archive (p)</li> </ul>	Create or import a project     New project     New project     Import SDK example(s)     Import project(s) from file system      Build your project     Build     Solution     Clean      Debug your project     Solution     Pobug     Pobug     Pobug     Pobug     Miscellaneous     Clean     Solution     Clean     Solution     Clean     Solution     Clean     Clean     Solution     Clean     Cl	🙂 Quickstart Panel 🗰 Global Variables 🕫 Variables 🕫 Breakpoints 🕃 Outline	
<ul> <li>New project</li> <li>Import SDK example(s)</li> <li>Import project(s) from file system</li> <li>Build your project</li> <li>Clean.</li> <li>Debug your project</li> <li>Clean.</li> <li>Debug</li> <li>Miscellaneous</li> <li>Edit project settings</li> <li>Quick Settings&gt;&gt;</li> <li>Export project(s) to archive (zip)</li> </ul>	<ul> <li>New project</li> <li>Import SDK example(s)</li> <li>Import project(s) from file system</li> <li>Build your project</li> <li>Build your project</li> <li>Clean</li> <li>Debug your project</li> <li>Pebug</li> <li>Terminate, Build and Debug</li> <li>Miscellaneous</li> <li>Edit project settings</li> <li>Quick Settings&gt;</li> <li>Export project(s) and references to archive (zip)</li> </ul>	MCUXpresso IDE - Quickstart Panel No project selected	
<ul> <li>Import SDK example(s)</li> <li>Import project(s) from file system</li> <li>Build your project</li> <li>Build Clean</li> <li>Debug your project</li> <li>Debug</li> <li>Miscellaneous</li> <li>Filt project settings</li> <li>Quick Settings&gt;&gt;</li> <li>Export project(s) to archive (zip)</li> <li>Export project(s) and references to archive (zip)</li> </ul>	<ul> <li>Import SDK example(s)</li> <li>Import project(s) from file system</li> <li>Build your project</li> <li>Build your project</li> <li>Clean</li> <li>Debug your project</li> <li>P Debug</li> <li>Terrinate, Build and Debug</li> <li>Miscellaneous</li> <li>Edit project settings</li> <li>Quick Settings&gt;</li> <li>Export project(s) to archive (zip)</li> <li>Export project(s) and references to archive (zip)</li> </ul>	* Create or import a project	
<ul> <li>Build</li> <li>Clean</li> <li>Debug</li> <li>Terminate, Build and Debug</li> <li>Miccellaneous</li> <li>Edit project settings</li> <li>Quick Settings&gt;&gt;</li> <li>Export project(s) to archive (zip)</li> <li>Export project(s) and references to archive (zip)</li> </ul>	<ul> <li>Build</li> <li>Clean</li> <li>Clean</li> <li>Debug</li> <li>Perminate, Build and Debug</li> <li>Miscellaneous</li> <li>Edit project settings</li> <li>Quick Settings&gt;</li> <li>Export project(s) to archive (zip)</li> <li>Export project(s) and references to archive (zip)</li> </ul>	🚧 🔤 Import SDK example(s)	
Debug your project			
<ul> <li>Debug</li> <li>Terminate, Build and Debug</li> <li>Miscellaneous</li> <li>Edit project settings</li> <li>Quick Settings&gt;&gt;</li> <li>Export project(s) and references to archive (zip)</li> </ul>	<ul> <li>Debug</li> <li>Terminate, Build and Debug</li> <li>Miscellaneous</li> <li>Edit project settings</li> <li>Quick Settings&gt;</li> <li>Export project(s) to archive (zip)</li> <li>Export project(s) and references to archive (zip)</li> </ul>	ති <sup>5</sup> <sup>build</sup>	
<ul> <li>Miscellaneous</li> <li>Edit project settings</li> <li>Quick Settings &gt;</li> <li>Export project(s) to archive (zip)</li> <li>Export project(s) and references to archive (zip)</li> </ul>	<ul> <li>Miscellaneous</li> <li>Edit project settings</li> <li>Quick Settings &gt;&gt;</li> <li>Export project(s) to archive (zip)</li> <li>Export project(s) and references to archive (zip)</li> </ul>		🛛 🕶 🔛 👻 🛃 🕶
<ul> <li>Miscellaneous</li> <li>Edit project settings</li> <li>Quick Settings &gt;</li> <li>Export project(s) to archive (zip)</li> <li>Export project(s) and references to archive (zip)</li> </ul>	<ul> <li>Miscellaneous</li> <li>Edit project settings</li> <li>Quick Settings &gt;&gt;</li> <li>Export project(s) to archive (zip)</li> <li>Export project(s) and references to archive (zip)</li> </ul>	Debug * Terminate, Build and Debug	
Quick Settings>>     Project(s) to archive (zip)     Export project(s) and references to archive (zip)	<ul> <li>Quick Settings&gt;&gt;</li> <li>Export project(s) to archive (zip)</li> <li>Export project(s) and references to archive (zip)</li> </ul>	* Miscellaneous	
		<ul> <li>Quick Settings&gt;&gt;</li> <li>Export project(s) to archive (zip)</li> <li>Export project(s) and references to archive (zip)</li> </ul>	
		Figure 4. Import an SDK example	

3. In the window that appears, expand the LPC55xx folder and select LPC55S16. Then, select LPC55S16 and click Next.

SDK Import Wizard			— 🗆 X
Importing project(s) for	device: LPC55S16 using board: LPCXpre	sso55516	
Board and/or D	Device selection page		
- SDK MCUs	Available boards		Jª îª   a
MCUs from installed SDKs	Please select an available board for yo	our project.	
Please visit	Supported boards for device: LPC55	516	
mcuxpresso.nxp.com to	NKO ST		Hard and the second second
obtain additional SDKs.	<b>9</b>		
NXP LPC55S16			
✓ LPC55xx LPC55S16	Ipcxpresso55s16	Docxpresso55s16 agm01	sso55s16 om13790host
	<		
Selected Device: LPC555	16 using board: LPCXpresso55S16	SDKs for selected MCU	
Target Core: cm33		Name SDK Ve.	Manife Location
	xx are ARM Cortex-M33 based rollers for embedded applications.	SDK_2.x_board_LP( 2.6.0	3.5.0 🖳 <common>\board_LF</common>
0		< Back Next >	Finish Cancel
Figure 5. Select LPCX	presso55S16 board		

4. Expand the  $demo_apps$  folder and select  $hello_world$ . Then, select UART as SDK Debug Console and click Next.

SDK Import Wizard				x
The source from the SDK will be copied into the workspace. If you want to use linked files, please unzip the 'SDK_2.x_FRDM-K64F' SDK.	N	P	T <sup>C</sup>	
Import projects				
Project name prefix: frdmk64f_			a	
✓ Use <u>d</u> efault location				
Location: C:\Users\b59906\Documents\MCUXpressoIDE_10.0.0_299_beta\workspace\frdmk64f_			B <u>r</u> owse	ן
Project Type	Project Options			
C Project C++ Project C Static Library C++ Static Library	Copy sources			
		^^	1	_
Examples	è	a 🗹 🦓		
Name	Version	l	-	
Cmsis_driver_examples     demo_apps				
▷ □ □ □ lwip				
▶ 🔲 🚪 mbedtls				
wifi_qca				
wolfssl				
adc16_low_power				
🔲 🧮 bubble				
ac_adc			-	
E ecompass     E ftm_pdb_adc16     E			-	
$ \equiv \underline{ftm} \underline{pub} \underline{aucto} $				ľ
power_manager				
power_mode_switch				
m ≣ rtc_func				
m ≡ chell III				
III			,	
?	<u>N</u> ext > <u>Einis</u>	ih	Cancel	
jure 6. Select hello_world				

🔀 SDK Import Wizard		— 🗆 X
A The source from the SDK will be copied into the	e workspace.	
If you want to use linked files, please unzip the		
🔀 Import projects		
Project name prefix: lpcxpresso55s16	Project name suffix:	bi di seconda di second
✓ Use default location		
Location: C:\Users\nxf55082\Documents\MCUXp	ressoIDE_11.1.0_3169_prc3\workspace\lpcxpresso	55s16 Browse
Project Type	Project Options	
● C Project ○ C++ Project ○ C Static Library		Semihost
	Copy sources Import other files	
Examples		🔟 🖉 🔌 🖽 E
type to filter		
Name	Description	Version
> 🔄 🗏 FreeMASTER_examples		
Comparison in the second se		
✓ ■ Ξ demo_apps		
▶ hello_world	The Hello World demo application pro	
■ <b>hello_world_swo</b>	The Hello World SWO demo prints th	
□ ■ hello_world_virtual_com	The Hello World demo application pro	
□ ≡ led_blinky □ ≡ power_manager_lpc	The LED Blinky demo application prov The power_manager_lpc application s	2
Shell	The Shell Demo application demonstr	
□ = shell □ ≡ utick_wakeup	The purpose of this demo is to show	
>	···· F ··· F ··· F ··· · · · · · · · ·	······
?	< Back Next >	Finish Cancel
Figure 7. Select hello world		

5. Ensure Redlib: Use floating point version of printf is selected if the example prints floating point numbers on the terminal for demo applications such as adc\_basic, adc\_burst, adc\_interrupt. Otherwise, it is not necessary to select this option. Then, click Finish.

🔀 SDK Import Wizard								$\times$
						N		E
X Advanced Set	tings				 			
• C/C++ Library Settin	ngs							
Set library type (and h	osting variant) Redlib (nohost-nf)	N						
Redlib: Use floating	point version of printf r rather than string based printf			wlibNano: Use wlibNano: Use				
Redirect SDK "PRINT	FF" to C library "printf" ardFault handler			direct printf/sc direct printf/sc				
<ul> <li>Hardware settings</li> </ul>								
Set Floating Point type	FPv5-SP-D16 (Hard ABI)							
• MCU C Compiler								
Language standard	GNU C99 (-std=gnu99)							~
TrustZone Project Type	None							$\sim$
• MCU Linker								
Link application to F	RAM		istZone P one	roject Type				~
<ul> <li>Memory Configuration</li> </ul>	ion							
Memory details								
Default LinkServer Flas	h Driver						Browse	e
?		< Back		Next >	Finish		Canc	el
Figure 8. Select ⊍se	floating point version o	of printf						

#### 3.3 Run an example application

For more information on debug probe support in the MCUXpresso IDE, see community.nxp.com.

To download and run the application, perform the following steps:

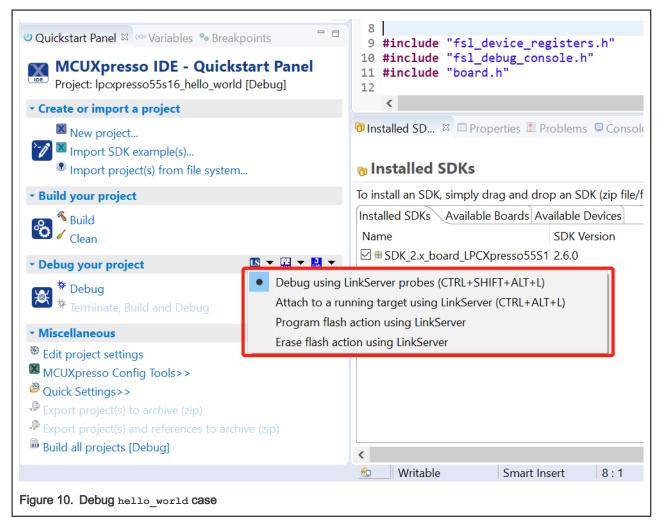
- 1. See the table in Default debug interfaces to determine the debug interface that comes loaded on your specific hardware platform. For LPCXpresso boards, install the DFU jumper for the debug probe, then connect the debug probe USB connector.
  - For boards with a P&E Micro interface, see PE micro to download and install the P&E Micro Hardware Interface Drivers package.
  - For the MRB-KW01 board, see www.nxp.com/USB2SER to download the serial driver. This board does not support
    the OpenSDA. Therefore, an external debug probe (such as a J-Link) is required. The steps below referencing the
    OpenSDA do not apply because there is only a single USB connector for the serial output.
  - If using J-Link with either a standalone debug pod or OpenSDA, install the J-Link software (drivers and utilities) from www.segger.com/jlink-software.html.

- For boards with the OSJTAG interface, install the driver from www.keil.com/download/docs/408.
- 2. Connect the development platform to your PC via a USB cable.
- 3. Open the terminal application on the PC, such as PuTTY or TeraTerm, and connect to the debug serial port number (to determine the COM port number, see How to determine COM port. Configure the terminal with these settings:
  - a. 115200 or 9600 baud rate, depending on your board (reference <code>BOARD\_DEBUG\_UART\_BAUDRATE</code> variable in <code>board.h</code> file)
  - b. No parity
  - c. 8 data bits
  - d. 1 stop bit

Г

egory:						
Session	Basic options for yo	Basic options for your PuTTY session				
Logging	Specify the destination you w	ant to connect to				
- Terminal - Keyboard	Serial line	Speed				
- Bell	COM16	115200				
Features Window Appearance Behaviour	Connection type: Raw © Telnet © Rk Load, save or delete a stored	ogin () <u>S</u> SH () Serja I session				
Translation	Saved Sessions					
- Selection	Debug					
Colours	Default Settings	Load				
Connection	Debug					
Proxy		Save				
- Telnet		Delete				
Flogin						
⊕- SSH Serial	Close window on exit: Always Never	Only on clean exit				
Ab						
About	elp	Open <u>C</u> ancel				

4. On the Quickstart Panel, click on Debug lpcxpresso55s16\_hello\_world [Debug] to launch the debug session.

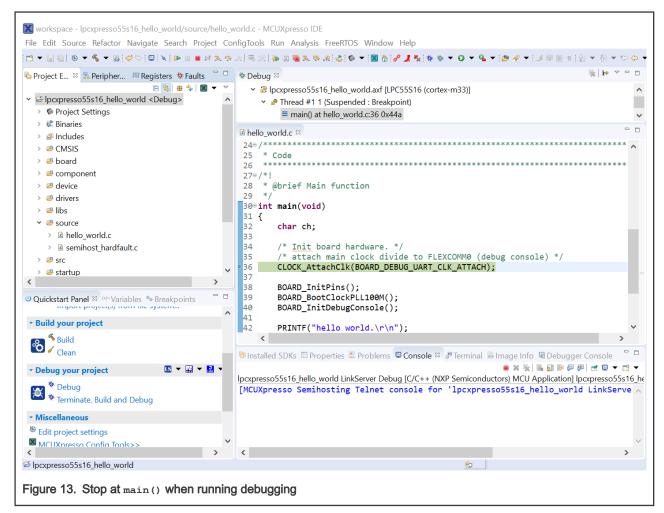


5. The first time you debug a project, the **Debug Emulator Selection** dialog is displayed, showing all supported probes that are attached to your computer. Select the probe through which you want to debug and click **OK**. (For any future debug sessions, the stored probe selection is automatically used, unless the probe cannot be found.)

Pr	obes discovered			-				
	Connect to target: MK64FN1M0xxx12         1 probe found. Select the probe to use:							
Ava	Available attached probes							
	Name	Serial number/ID	Туре	Manu	IDE Debug Mode			
12	USB1 - OpenSDA (7A790E49	7A790E49	USB1	P&E Mi	All-Stop			
	ported Probes (tick/untick to MCUXpresso IDE LinkServer P&E Micro probes		robes					
	SEGGER J-Link probes							
	arch again							
🔽 Re	Remember my selection (for this Launch configuration)							
?	)			ОК	Cancel			
	hed Probes: debug emulator							

Name	Serial number/ID	Туре	Manufa	IDE Debug Mo	ode
LPC-LINK2 CMSIS-DAP V5.182	BRAUBQER	LinkSer	NXP Se	Non-Stop	
			1		
•					
	CMSIS-DAP) probes				
be search options					
arch again					
ł	LPC-LINK2 CMSIS-DAP V5.182 ported Probes (tick/untick to ena MCUXpresso IDE LinkServer (inc. P&E Micro probes SEGGER J-Link probes pe search options	LPC-LINK2 CMSIS-DAP V5.182 BRAUBQER ported Probes (tick/untick to enable/disable) MCUXpresso IDE LinkServer (inc. CMSIS-DAP) probes P&E Micro probes SEGGER J-Link probes pe search options	LPC-LINK2 CMSIS-DAP V5.182       BRAUBQER       LinkSer         LinkSer       LinkSer         ported Probes (tick/untick to enable/disable)         MCUXpresso IDE LinkServer (inc. CMSIS-DAP) probes         P&E Micro probes         SEGGER J-Link probes         be search options	LPC-LINK2 CMSIS-DAP V5.182 BRAUBQER LinkSer NXP Se LinkSer NXP Se LinkSer NXP Se LinkSer NXP Se InkSer NXP Se	LPC-LINK2 CMSIS-DAP V5.182 BRAUBQER LinkSer NXP Se Non-Stop

6. The application is downloaded to the target and automatically runs to  ${\tt main}\left(\cdot\right)$  .



7. Start the application by clicking Resume.

	Project Pup Window
	1 = N
igure 14. Resume button	

The hello\_world application is now running and a banner is displayed on the terminal. If this is not the case, check your terminal settings and connections.



### 3.4 Build a TrustZone example application

This section describes the steps required to configure MCUXpresso IDE to build, run, and debug TrustZone example applications. The trustzone version of the hello\_world example application targeted for the LPCXpresso55S16 hardware platform is used as an example, though these steps can be applied to any TrustZone example application in the MCUXpresso SDK.

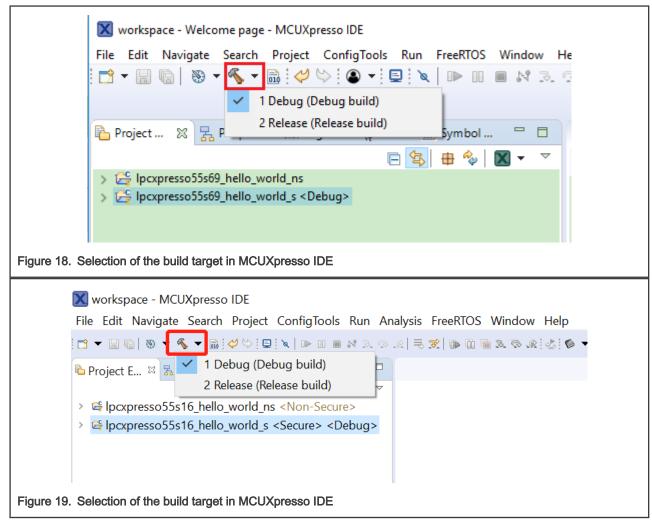
1. TrustZone examples are imported into the workspace in a similar way as single core applications. When the SDK zip package for LPCXpresso55S16 is installed and available in the **Installed SDKs** view, click **Import SDK example(s)**... on the Quickstart Panel. In the window that appears, expand the LPC55xx folder and select LPCXpresso55S16. Then, select LPCXpresso55S16 and click Next.

SDK Import Wizard					$\times$
<sup>①</sup> Importing project(s) for a	levice: LPC55S16 using board: LPCX	presso55S16		NP	6
🔀 Board and/or D	evice selection page				
- SDK MCUs	Available boards			ļ	az taz   🖉
MCUs from installed SDKs.	Please select an available board for	r your project.			
Please visit	Supported boards for device: LPC	55S16			
<u>mcuxpresso.nxp.com</u> to obtain additional SDKs.		Note the second			
NXP LPC55S16					
✓ LPC55xx					
LPC55S16					
	SDK III	SDK		SDK	
	lpcxpresso55s16	lpcxpresso55s16 agm01 lpc	<u>xpresso55s16_om</u>	<u>13790host</u>	
	<				>
Selected Device: LPC55S1	6 using board: LPCXpresso55516	SDKs for selected MCU			
Target Core: cm33		Name SDI	K Ve Manife	Location	
Description: The LPC55x	x are ARM Cortex-M33 based	#SDK 2.x board LP( 2.6.	.0 3.5.0	Common>\bd	oard LPC
microcontr	ollers for embedded applications.				_
?		< Back Next >	Finish	Canc	ما
		Next >	TITIST	Caric	ei
Figure 16. Select the L	PCXpresso55S16 board				

2. Expand the trustzone\_examples/ folder and select hello\_world\_s. Because TrustZone examples are linked together, the non-secure project is automatically imported with the secure project, and there is no need to select it explicitly. Then select UART as SDK Debug Console. Then, click **Finish**.

ard_LPCXpresso55S16' SDK. The advanced options page		2
rd_LPCXpresso55S16' SDK. The advanced options page		
2 Project name suffix:		
Project name suffix:		
1.0_3169_prc3\workspace\lpcxpresso55s16	Br	owse
Project Options		
c Library SDK Debug Console O Semihost O UART	) Example de	fault
✓ Copy sources		
✓ Import other files		
	2	🕀
Description	Version	
	-	
The Secure Faults demo application demonstrates handlin		
The Secure GPIO dome application demonstrates using of		
The Secure GPIO demo application demonstrates using of .		
The Secure GPIO demo application demonstrates using of . The Secure GPIO demo application demonstrates using of .		
	Project Options         c Library         SDK Debug Console () Semihost () UART ()         Copy sources         Import other files	Project Options         c Library         SDK Debug Console ○ Semihost ● UART ○ Example de         ○ Copy sources         ☑ Import other files         ☑ @ @ @         Description         Version         The Hello World demo application provides a sanity check         The Hello World demo application provides a sanity check         The Secure Faults demo application demonstrates handlin

3. Now, two projects should be imported into the workspace. To start building the TrustZone application, highlight the lpcxpresso55s16\_hello\_world\_s project (TrustZone master project) in the Project Explorer. Then, choose the appropriate build target, **Debug** or **Release**, by clicking the downward facing arrow next to the hammer icon, as shown in Figure 18. For this example, select the **Debug** target.



The project starts building after the build target is selected. It is requested to build the application for the secure project first, because the non-secure project needs to know the secure project since CMSE library when running the linker. It is not possible to finish the non-secure project linker when the secure project since CMSE library is not ready.

#### NOTE

When the **Release** build is requested, it is necessary to change the build configuration of both the secure and non-secure application projects first. To do this, select both projects in the Project Explorer view by clicking to select the first project, then using shift-click or control-click to select the second project. Right click in the Project Explorer view to display the context-sensitive menu and select **Build Configurations > Set Active >Release**. This is also possible by using the menu item of **Project > Build Configuration >Set Active >Release**. After switching to the **Release** build configuration. Build the application for the secure project first.

🔀 workspace - Welco	me p	age - MCUXpresso IDE					
File Edit Navigate	Sear	ch Project ConfigTools Run	FreeRTOS V	Vindow H	Help		
📬 🗕 🖪 👘 🗎 😁 🗸	- 6	- 🗟 🞺 😂 - 📃 🔌		N 3.	3 .e   🗟 🔊   9	Þ Ü 🖷 3	R R 🖗 🕇 🕹 💰
🎦 Project 🕱 🚼	Perip	her 👭 Registers 🔺 Faults 🛔	💭 Symbol		🕥 Welcome 🐹		
		日 🕏	⊞%	<b>•</b> •	⇔ ⇒ ■	file:///C:/nxp/l	MCUXpressolDE_10.3.0_219
> 😂 lpcxpresso55s69		o_world_ns <debug></debug>					
> 😂 lpcxpresso55s69		New	>	1			
		Go Into					
		Show in Local Terminal	>				To install as SDK, simpl
	D	Сору	Ctrl+C				
	Ē	Paste	Ctrl+V			_	
	×	Delete	Delete				
		Source	>				D = 0v 1. mail
		Move	F2				
) Quickstar (×)= Glo		Rename	F2				ACUXpresso IDE provides a Cortex®-M cores, including I
Quiciotanii ( ) olo	<u>121</u>	Import				6	editing, compiling, and debu nulticore debugging, and inf
	4	Export		^			our installation of MCUXpre
MCUXpres Project: Ipcxpr		Build Project				r	equired.
<ul> <li>Create or import a</li> </ul>		Clean Project Refresh	F5		nstalled SDKs	Properties	📃 Console 🐹 🖹 Pro
-		Close Project					
New proje		Close Unrelated Projects					lo_world_s JLink Debug [0
🔚 💽 Import pr		Build Configurations	>	Set A		1 Debug	lnet console for 'l (Debug build)
<ul> <li>Build your project</li> </ul>		Build Targets	>	Man	age	2 Release	e (Release build)
💽 🐔 Build		Index	>	Build	I All	world. al world!	
💟 🖌 Clean		Validate		Clear	n All	printed fro	om normal world!
Debug your project		Run As	>	Build	Selected	tring as a	callback to normal
		Debug As	>		String 2: Test2 Both strings ar		1
Terminate		Profile As	>		2001 201 2062 01	e noe equa	
<ul> <li>Miscellaneous</li> </ul>		Restore from Local History Launch Configurations	>		[Closed Telnet	Session]	
<ul> <li>miscendheous</li> </ul>		counter configurations				-	
gure 20. Switchi	ng T	rustZone projects into the	Release bu	ild config	guration		

ile Edit Navigate Search		New		>	Window Help
s • 🔚 🐚 📎 • 🔦 • 🗎		Go Into			3. 3. 2. 2. 3. 4. ▼ IM 1: 8 1 8 1 8 4. ▼ 0 ▼ 4. ▼ 19 8
Project E 🛛 🔀 Peripher.		Show in Local Terminal		>	
a Project E ~ as Penpher.		Сору	Ct	rl+C	
😂 lpcxpresso55s16_hellc		Paste		rl+V	
😂 lpcxpresso55s16_hellc	×	Delete	De	elete	
		Source		>	
		Move		50	
		Rename		F2	
		Import			
		Export			
		Build Projects			
	କ	Clean Project Refresh		F5	
	ŝ.	Close Projects		FD	
		Close Unrelated Project			
		Build Configurations		>	Set Active > ✓ 1 Debug (Debug build)
		Build Targets		>	Manage 2 Release (Release build)
		Index		>	Build All
Quickstart Panel 🖾 🙁 Var	0	Run As		>	Clean All
MCUXpresso II	*	Debug As		>	Build Selected
Project: lpcxpresso55:		Profile As		>	build Selected
Create or import a proj		Restore from Local History			
New project		Launch Configurations		>	🗖 Properties 🖹 Problems 📮 Console 🛛 🦑 Terminal 🗟 Image Ir
New project Import SDK example		Utilities		>	presso55s16_hello_world LinkServer Debug [C/C++ (NXP Semic
<ul> <li>Import sold examp</li> <li>Import project(s) fi</li> </ul>		Tools		>	pressource repug [C/C++ (NXP Semic
		Validate			
Build your project	**	Run C/C++ Code Analysis		>	t Session]
🗞 🔨 Build 🖌 Clean		Team Compare With		>	
<ul> <li>Clean</li> </ul>		Compare with Configure		>	
2 items selected		Source		>	
2 items selected		Source		1	

## 3.5 Run a TrustZone example application

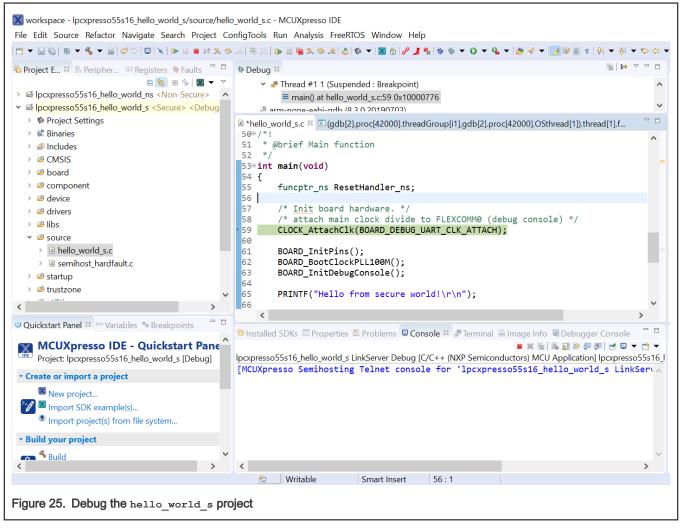
To download and run the application perform all steps as described in *Section 3.3, "Run an example application"*. These steps are common for single core, dual-core, and TrustZone applications, ensuring both sides of the TrustZone application are properly loaded and started secure application. However, there is one additional dialogue that is specific to TrustZone examples. See the following figures as reference.

workspace - lpcxpresso55s16_hello_world_ns/source/hello_ File Edit Source Refactor Navigate Search Project Conf	·
	R   ₹ \$\$   \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$\$ \$
<ul> <li>Project Expl E Peripherals + IIII Registers * Faults</li> <li>Project Expl Peripherals + IIII Registers * Peripherals + IIII Registers + Peripherals + Peripherals + IIII Registers + Peripherals + IIII Registers + Peripherals + Peripherals + IIII Registers + Peripherals + IIII Registers + Peripherals + IIII Registers + Peripherals + Perip</li></ul>	<pre>     Phello_world_ns.c ☆     1⊕/*     2 * Copyright 2018 NXP     3 *     4 * SPDX-License-Identifier: BSD-3-Clause     5 */     6     7 #include "fsl_device_registers.h"     8 #include "fsl_debug_console.h"     9 #include "veneer_table.h"     10 #include "pin_mux.h"     11 #include "board.h"     12 #include "clock config.h" </pre>
<ul> <li>O Quickstart Panel ≅ ∞ Variables • Breakpoints</li> <li>MCUXpresso IDE - Quickstart Panel Project: lpcxpresso55s16_hello_world_ns [Debug]</li> <li>Create or import a project</li> <li>New project</li> <li>New project</li> <li>Import SDK example(s)</li> <li>Import project(s) from file system</li> <li>Build your project</li> <li>Build Clean</li> <li>Debug your project</li> </ul>	<ul> <li>□ 13⊕ /************************************</li></ul>
<ul> <li>Debug</li> <li>Terminate, Build and Debug</li> <li>Miscellaneous</li> <li>Edit project settings</li> <li>MCUXpresso Config Tools&gt;&gt;</li> </ul>	LinkServer probes (CTRL+SHIFT+ALT+L) nning target using LinkServer (CTRL+ALT+L) action using LinkServer LinkServer 0 warnings. (took 6s
Quick Settings>> © Ipcxpresso55s16_hello_world_ns	
Figure 22. Load lpcxpresso55s16_hello_world_ns	4 <u></u> 4

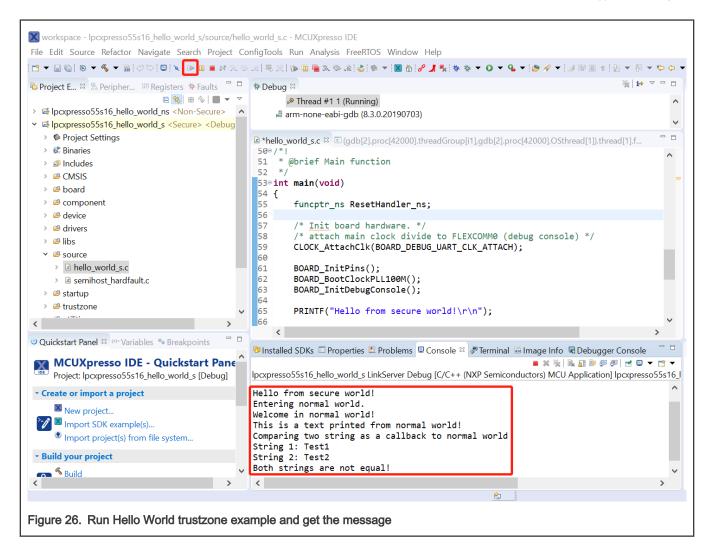
۸	allable attacked weekse					
AV	ailable attached probes	Serial number/ID	Туре	Manufa	IDE Debug Mode	
LS	LPC-LINK2 CMSIS-DAP V5.182	BRAUBQER	LinkSer	NXP Se		
_						
C						
Sup	pported Probes (tick/untick to ena MCUXpresso IDE LinkServer (inc	-				
$\overline{\checkmark}$		. Civisis-DAFJ probes				
	SEGGER J-Link probes					
	be search options					
	earch again					
36	aich agairt					

After loading the non-secure application, press **RESET** on board to release the device connect. Then, highlight the lpcxpresso55s16\_trustzone\_examples\_hello\_world\_s project (TrustZone master project) in the Project Explorer. In the Quickstart Panel, click lpcxpresso55s16\_trustzone\_examples\_hello\_world\_s [Debug] to launch the second debug session.

<ul> <li>Workspace - Source not found MCUXpresso IDE</li> <li>File Edit Navigate Search Project ConfigTools Run Analysis Fr</li> <li>Image: Image: Image:</li></ul>	·
① Quickstart Panel          ◎ Wariables          ● Breakpoints         □         □         MCUXpresso IDE - Quickstart Panel         Project: lpcxpresso55s16_hello_world_s [Debug]         □         □         □	
<ul> <li>Create or import a project</li> <li>New project</li> <li>Import SDK example(s)</li> <li>Import project(s) from file system</li> </ul>	
	Installed S □ Properties ڲ Problems □ Console ≅ Prominal □ Image ↓ ↑ CDT Build Console [lpcxpresso55s16_hello_world_s] Finished building target: lpcxpresso55s16_hello_world_s
Debug voir project     Debug using Lin     Attach to a runn     Program flash av	<pre>kServer probes (CTRL+SHIFT+ALT+L) ing target using LinkServer (CTRL+ALT+L) ction using LinkServer n using LinkServer 15:39:13 Build Finished. 0 errors, 0 warnings. (took 8s </pre>
Figure 24. Debug lpcxpresso55s16_hello_world_s case	Э



Start the application by clicking **Resume**. The hello\_world TrustZone application then starts running, and the secure application starts the non-secure application during run time.



# 4 Run a demo application using IAR

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

NOTE

IAR Embedded Workbench for Arm version 8.32.3 is used in the following example, and the IAR toolchain should correspond to the latest supported version, as described in the *MCUXpresso SDK Release Notes*.

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

NOTE

IAR Embedded Workbench for Arm version 8.32.1 is used in the following example, and the IAR toolchain should correspond to the latest supported version, as described in the *MCUXpresso SDK Release Notes* (document ID: MCUXSDKRN).

### 4.1 Build an example application

Do the following steps to build the hello world example application.

1. Open the desired demo application workspace. Most example application workspace files can be located using the following path:

<install\_dir>/boards/<board\_name>/<example\_type>/<application\_name>/iar

Using the LPCXpresso55S16 hardware platform as an example, the hello world workspace is located in:

<install\_dir>/boards/lpcxpresso55s16/demo\_apps/hello\_world/iar/hello\_world.eww

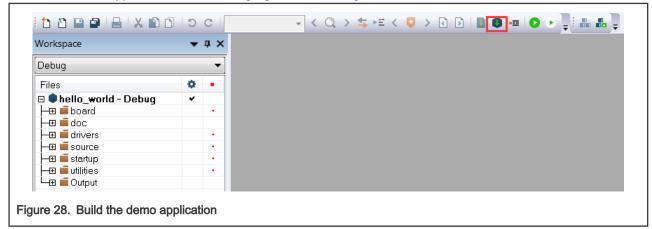
Other example applications may have additional folders in their path.

2. Select the desired build target from the drop-down menu.

For this example, select hello\_world - debug.

Workspace		×
Debug		-
Debug		
Release		
🛛 🖂 🗇 hello_world - Deb	×	
🛛 🛏 🖸 board		
doc		
🚽 🛏 🖬 drivers		
source		
📕 🛏 🖬 🔂 startup		
📕 🛏 🖬 🗀 utilities		
📗 🖵 🔂 Output		
gure 27. Demo build target selection		

3. To build the demo application, click Make, highlighted in red in Figure 28.



4. The build completes without errors.

#### 4.2 Run an example application

To download and run the application, perform these steps:

- 1. Connect the development platform to your PC via USB cable.
- 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 or 9600 baud rate, depending on your board (reference <code>BOARD\_DEBUG\_UART\_BAUDRATE</code> variable in the <code>board.h</code> file)
- b. No parity
- c. 8 data bits
- d. 1 stop bit

	Basic options for	your PuTTY session
Logging	Specify the destination you	want to connect to
E- Terminal	Serial line	Speed
Bell	COM16	115200
Features	Connection type:	
Window	○ Raw ○ Telnet ○	Rlogin © <u>S</u> SH 💿 Serial
Appearance Behaviour	Load, save or delete a stor	red session
Translation	Saved Sessions	
Selection	Debug	
Colours	Default Settings	Load
- Connection	Debug	
Proxy		Save
Telnet		Delete
Riogin		
i⊞- SSH Serial	10-	
Jona	Close window on exit: Always Never	Only on clean exit
	O Aways O Never	Only on clean exit
About	elp	Open <u>C</u> ancel

3. In IAR, click the **Download and Debug** button to download the application to the target.



4. The application is then downloaded to the target and automatically runs to the main() function.

Vorkspace Debug Files E <b>● hello_world - Debuq</b>	₹ † ×	hello_world.c ×
Files	-	
	· ·	41
🗆 🔍 hello 🛛 world - Debug	۰ ب	42 🖂 /******
	*	43 * Prototypes
- 🕀 🛑 board		44 L
—⊞ 🖬 doc —⊞ 🖬 drivers		45 46 🖂 /***********************************
He source		47 * Code
- I i startup		48 L ***********************************
🕀 🖬 utilities		
🖵 🖬 Output		50 * @brief Main function 51 +/
		⇒ 52 int main (void)
		53 🖵 🕻
		54 char ch;
		55
		56 /* Init board hardware. */ 57 /* attach 12 MHz clock to FLEXCOMM0 (debug console) */
		58 CLOCK AttachClk(BOARD DEBUG UART CLK ATACH);
		59
		60 BOARD_InitPins();
		61 BOARD_BootClockFROHF48M();
		62 BOARD_InitDebugConsole();
t 8 🖴 🔿 🖴 IX 🗈 6 I	5 C	- < Q > \$ ⊨ < ♥ > < ▷   ■ ● = 'Θ ⊂ 8 ↓ ∩ ∓ ⋫ + + ● □ 5 -↓
	5 С ▼ <b>म</b> ×	
Vorkspace	▲ ů ×	hello_world.c ×
Vorkspace		hello_world.c × main()
Vorkspace Debug	▲ ů ×	hello_world.c ×
Vorkspace Debug Files 3 <b>• hello_world - Debug</b>	▼ ‡ ×	hello_world.c         X           main()         41           42         /************************************
Vorkspace Debug Files I • hello_world - Debug	• • ×	hello_world.c         X           main()         41           42         +           43         +           44         +
Vorkspace Debug Files Debulo_world - Debug Deboard Deboard Deboard Deboard	• • ×	hello_world.c         ×           main()         41           42         +           43         +           44         +
Vorkspace Debug Files • hello_world - Debug • board • board • doc • doc	• • ×	hello_world.c         X           main()         41           42         +           43         +           44         +
Vorkspace Debug Files • hello_world - Debug • board • board • doc • doc • drivers • • • source	• • ×	hello_world.c ×         main()         41         42         + Prototypes         44         45         46         47         * Code
Vorkspace Debug Files • hello_world - Debug • board • board • doc • doc	• • ×	hello_world.c X         main()         41         42         + Prototypes         43         + Code         44         + Code         43         + Code         43         + 1
Vorkspace Debug Files • hello_world - Debug • board • board • doc • doc • divers • divers • source • source	• • ×	hello_world.c X         main()         41         42         43         * Prototypes         44         45         46         * Code         47         * Code         48         49         * brief Main function
Vorkspace Debug Files Phello_world - Debug Debord Debord Debord Debord Debord Debord Debug Source Debug Debu	• • ×	$ \begin{array}{c c}                                    $
Vorkspace Debug Files Phello_world - Debug Deboard Deboard Deboard Deboard Deboard Deboard Deboard Debug Debu	• • ×	hello_world.c X         main()         41         42         43         * Prototypes         44         45         46         * Code         47         * Code         48         49         * brief Main function
Vorkspace Debug Files Phello_world - Debug Debord Debord Debord Debord Debord Debord Debug Source Debug Debu	• • ×	<pre>     hello_world.c X     main()     41     42 - /***********************************</pre>
Vorkspace Debug Files Phello_world - Debug Debu	• • ×	<pre>     hello_world.c X     main()     41     42      43     * Prototypes     44     45     46      /*     * Code     * Code     *     * Brief Main function     51     */     * 52     int main(void)     53      {</pre>
Vorkspace Debug Files Phello_world - Debug Debu	• • ×	<pre>     hello_world.c X      main()      41     42      43     * Prototypes     *********************************</pre>
B i doc     Grivers     Source     Startup     Startup	• • ×	<pre>     hello_world.c X     main()     41     42 - /***********************************</pre>
Vorkspace Debug Files Files Files Generation Files	• • ×	<pre>     hello_world.c X     main()     41     42 - /***********************************</pre>
Vorkspace Debug Files Files Files Generation Files	• • ×	<pre>kello_world.c X main()  41 42 43 44 45 46 44 45 46 7 47 6 7 * Code 48 44 45 45 45 46 7 * Code 48 47 47 50 47 * Code 52 knt main(void) 53 54 55 55 56 55 55 56 55 5 5 5 5 5 5 5 5</pre>
Vorkspace Debug Files Phello_world - Debug Debu	• • ×	<pre>     hello_world.c X      main()      41     42      43     * Prototypes     *********************************</pre>

5. Run the code by clicking the **Go** button.



6. The hello\_world application is now running and a banner is displayed on the terminal. If it does not appear, check your terminal settings and connections.

	Putty	
	hello world.	
		I
Figure 34. Text display of the hello	_world demo	

### 4.3 Build a TrustZone example application

This section describes the particular steps that need to be done in order to build and run a TrustZone application. The demo applications workspace files are located in this folder:

```
<install_dir>/boards/<board_name>/trustzone_examples/<application_name>/<core_type>/iar/<application_name> ns/iar
```

<install\_dir>/boards/<board\_name>/trustzone\_examples/<application\_name>/<core\_type>/iar/<application\_name> s/iar

Begin with a simple TrustZone version of the Hello World application. The TrustZone Hello World IAR workspaces are located in this folder:

```
<install_dir>/boards/lpcxpresso55s16/trustzone_examples/hello_world/hello_world_ns/iar/
hello world ns.eww
```

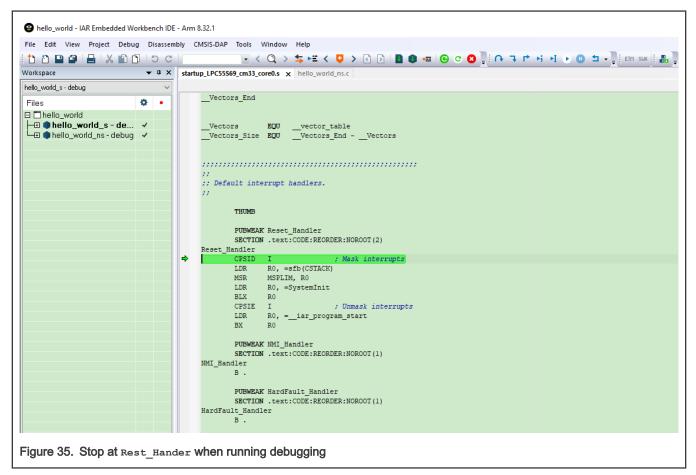
```
<install_dir>/boards/lpcxpresso55s16/trustzone_examples/hello_world/hello_world_s/iar/ hello world s.eww
```

<install dir>/boards/lpcxpresso55s16/trustzone examples/hello world/hello world s/iar/hello world.eww

This project hello\_world.eww contains both secure and non-secure projects in one workspace and it allows the user to easily transition from one project to another. Build both applications separately by clicking **Make**. It is requested to build the application for the secure project first, because the non-secure project needs to know the secure project, since the CMSE library is running the linker. It is not possible to finish the non-secure project linker with the secure project since CMSE library is not ready.

#### 4.4 Run a TrustZone example application

The secure project is configured to download both secure and non-secure output files, so debugging can be fully managed from the secure project. To download and run the TrustZone application, switch to the secure application project and perform steps 1-4 as described in *Section 4.2, Run an example application*. These steps are common for both single core, dual-core, and TrustZone applications in IAR. After clicking **Download and Debug**, both the secure and non-secure image are loaded into the device flash memory, and the secure application is executed. It stops at the Rest Hander function.



Run the code by clicking Go to start the application.

< Q, > ⇆ += < 😳 > R 🖻 🔳 🜒 += I 🕞 🗢 🕄 🍹 (+ 🥆 + H 🕨 💷 🖕 📰 🖦 📮 Figure 36. Go button

The TrustZone hello\_world application is now running and a banner is displayed on the terminal. If this is not true, check your terminal settings and connections.

Putty	_	×
Hello from secure world!		~
Entering normal world.		
Welcome in normal world!		
This is a text printed from normal world!		
Comparing two string as a callback to normal world		
String 1: Test1		
String 2: Test2		
Both strings are not equal!		
Figure 37. Text display of the trustzone hello_world application		

# 5 Run a demo using Keil<sup>®</sup> MDK/µVision

This section describes the steps required to build, run, and debug example applications provided in the MCUXpresso SDK. The hello\_world demo application targeted for the LPCXpresso55S16 hardware platform is used as an example, although these steps can be applied to any demo or example application in the MCUXpresso SDK.

### 5.1 Install CMSIS device pack

After the MDK tools are installed, Cortex<sup>®</sup> Microcontroller Software Interface Standard (CMSIS) device packs must be installed to fully support the device from a debug perspective. These packs include things such as memory map information, register definitions, and flash programming algorithms. Follow these steps to install the appropriate CMSIS pack.

1. Open the MDK IDE, which is called µVision. In the IDE, select the Pack Installer icon.

File Edit Vie	w Project Flash	Debug Pe	ripherals To	ols SVCS	Window
🗋 🐸 🖬 🕯	1 x 4a 🕰	9 10 4	-> P 7	自己的	律律》
8 🖻 🖽 (			- 8	66	* * 🖄

2. After the installation finishes, close the Pack Installer window and return to the  $\mu$ Vision IDE.

#### 5.2 Build an example application

1. Open the desired example application workspace in:

<install\_dir>/boards/<board\_name>/<example\_type>/<application\_name>/mdk

The workspace file is named as <demo\_name>.uvmpw. For this specific example, the actual path is:

<install dir>/boards/lpcxpresso55s16/demo apps/hello world/mdk/hello world.uvmpw

2. To build the demo project, select Rebuild, highlighted in red.

	Ś	) 🖾 🖼 🧼	hello_world Debug	• 🔊
Figure 39. Build the demo				

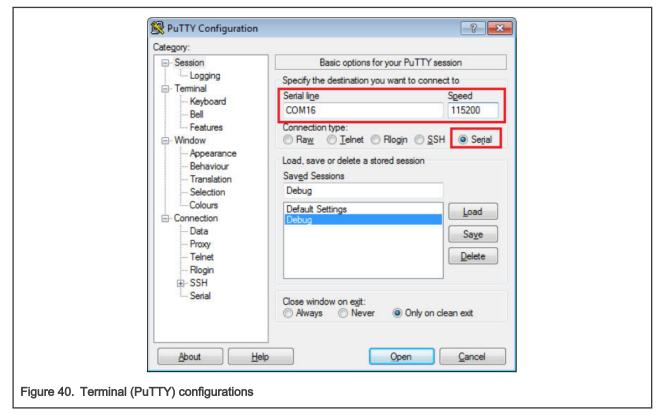
3. The build completes without errors.

#### 5.3 Run an example application

To download and run the application, perform these steps:

- 1. See the table in Default debug interfaces to determine the debug interface that comes loaded on your specific hardware platform.
  - For boards with the CMSIS-DAP/mbed/DAPLink interface, visit mbed Windows serial configuration and follow the instructions to install the Windows operating system serial driver. If running on Linux OS, this step is not required.
  - The user should install LPCScrypt or MCUXpresso IDE to ensure LPC board drivers are installed.

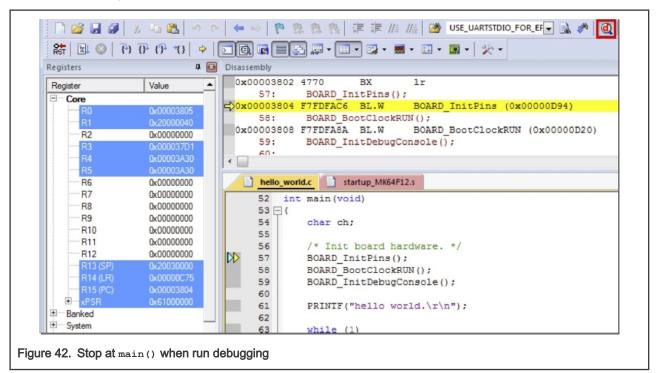
- For boards with a P&E Micro interface, visit www.pemicro.com/support/downloads\_find.cfm and download and install the P&E Micro Hardware Interface Drivers package.
- If using J-Link either a standalone debug pod or OpenSDA, install the J-Link software (drivers and utilities) from www.segger.com/jlink-software.html.
- If using J-Link either a standalone debug pod or J-link firmware programmed into the on-board debug probe, install the J-Link software (drivers and utilities) from www.segger.com/jlink-software.html.
- For boards with the OSJTAG interface, install the driver from www.keil.com/download/docs/408.
- 2. Connect the development platform to your PC via USB cable using OpenSDA USB connector.
- 3. Open the terminal application on the PC, such as PuTTY or TeraTerm and connect to the debug serial port number (to determine the COM port number, see How to determine COM port). Configure the terminal with these settings:
  - a. 115200 or 9600 baud rate, depending on your board (reference <code>BOARD\_DEBUG\_UART\_BAUDRATE</code> variable in the <code>board.h</code> file)
  - b. No parity
  - c. 8 data bits
  - d. 1 stop bit



4. In µVision, after the application is built, click the Download button to download the application to the target.

🖉 🖾 🕮 駴 🙀	hello_world Debug	- 🔊
Project	д 💌	
🖃 🚰 WorkSpace		
🗄 🔧 Project: hello_world		
gure 41. Download button		

5. After clicking the **Download** button, the application downloads to the target and is running. To debug the application, click the **Start/Stop Debug Session** button, highlighted in red.



6. Run the code by clicking the Run button to start the application.

	Registers Run (F5) Register Start code execution Core R0 0x00000966 R1 0x1FFF0440	
Figure 43. Go button		

The hello\_world application is now running and a banner is displayed on the terminal. If this does not appear, check your terminal settings and connections.

	Putty	
	hello world.	
Figure 44. Text display of the hello_	world demo	

### 5.4 Build a TrustZone example application

This section describes the particular steps that need to be done in order to build and run a TrustZone application. The demo applications workspace files are located in this folder:

<install\_dir>/boards/<board\_name>/trustzone\_examples/<application\_name>/iar/<application\_name>\_ns/mdk

<install\_dir>/boards/<board\_name>/trustzone\_examples/<application\_name>/iar/<application\_name>\_s/mdk

Begin with a simple TrustZone version of the Hello World application. The TrustZone Hello World Keil MSDK/µVision <sup>®</sup> workspaces are located in this folder:

```
<install_dir>/boards/lpcxpresso55s16/trustzone_examples/hello_world/hello_world_ns/iar/
hello_world_ns.eww
<install_dir>/boards/lpcxpresso55s16/trustzone_examples/hello_world/hello_world_s/mdk/
```

<install dir>/boards/lpcxpresso55s16/trustzone examples/hello world/hello world s/iar/hello world.eww

This project hello\_world.uvmpw contains both secure and non-secure projects in one workspace and it allows the user to easily transition from one project to another.

Build both applications separately by clicking **Rebuild**. It is requested to build the application for the secure project first, because the non-secure project needs to know the secure project since CMSE library is running the linker. It is not possible to finish the non-secure project linker with the secure project because CMSE library is not ready.

#### 5.5 Run a TrustZone example application

hello world s.uvmpw

The secure project is configured to download both secure and non-secure output files so debugging can be fully managed from the secure project.

To download and run the TrustZone application, switch to the secure application project and perform steps as described in Run a TrustZone example application. These steps are common for single core, dual-core, and TrustZone applications in µVision. After clicking **Download and Debug**, both the secure and non-secure image are loaded into the device flash memory, and the secure application is executed. It stops at the Rest Hander function.

		Peripherals Tools SVCS Window Help
	5 🖻 🚨 🤊 🤊	
👫   🗄 🚳   २२	0 + 10   ⇒	
Registers	ф <mark>Б</mark>	Disassembly
Register	Value	119: CPSID I ; Mask interrupts
R2	0x00000000	0x10000214 B672 CPSID 1
R3	0x10002141	120: LDR RO, = Image\$\$ARM_LIB_STACK\$\$ZI\$\$Base
R4	0x10002578	0x10000216 4846 LDR r0,0x10000330
R5	0x10002578	121: MSR MSPLIM, R0
R6	0x0C00F301	0x10000218 F380880A MSR MSPLIM,r0
R7	0x5AC3C35A	<
	0x55AACC33	
	0xC33CC33C	startup_LPC55S16.s bello_world_s.c
R10	0x5AC3C35A	113 Reset Handler PROC
R11	0x0000000	114 EXPORT Reset Handler [WEAK]
R12	0x0000002	115 IMPORT SystemInit
R13 (SP)	0x30008000	116 IMPORT main
R14 (LR)	0x10000521	117 IMPORT Image\$\$ARM LIB STACK\$\$2I\$\$Base
R15 (PC)	0x1000206C	118
± xPSR	0×6900000	119 CPSID I ; Mask interrupts
+ Banked		120 LDR R0, = Image\$\$ARM_LIB_STACK\$\$ZI\$\$Base
± Secure		121 MSR MSPLIM, R0
Non-Secure		122 LDR RO, =SystemInit
Internal		123 BLX R0
Mode	Secure Thread	124 CPSIE I ; Unmask interrupts
Privilege	Privileged	125 LDR R0, =main
Stack	MSP	126 BX R0
States	703	127 ENDP
Sec	0.00007030 -	
🖻 Project 🛛 🚟 Regist	ers	Text Editor Configuration Wizard

Run the code by clicking Run to start the application.

RST	¥. 📀	<del>{</del> }	0	{}•	*{}
Register	S EL D	ın (F5		п	1
Registe		art co		ecut	ion
E Co	re				
	R0		0x0	0000	965
	R1		0x1	FFF04	440

#### Figure 46. Go button

The hello\_world application is now running and a banner is displayed on the terminal. If this is not the case, check your terminal settings and connections.

×

Putty COM57 - Putty

3	
Hello from secure world!	
Entering normal world.	
Welcome in normal world!	
This is a text printed from normal world!	
Comparing two string as a callback to normal world	
String 1: Test1	
String 2: Test2	
Both strings are not equal!	
Figure 47. Text display of the trustzone hello_world application	

# 6 Run a demo using $\operatorname{Arm}^{\mathbb{R}}$ GCC

This section describes the steps to configure the command line Arm<sup>®</sup> GCC tools to build, run, and debug demo applications and necessary driver libraries provided in the MCUXpresso SDK. The hello\_world demo application is targeted for the LPCXpresso55S16 hardware platform which is used as an example.

#### NOTE

ArmGCC version 7-2018-q2 is used as an example in this document. The latest GCC version for this package is as described in the *MCUXpresso SDK Release Notes for LPCXpresso55S16* (document MCUXSDKLPC551XRN).

#### 6.1 Set up toolchain

This section contains the steps to install the necessary components required to build and run an MCUXpresso SDK demo application with the Arm GCC toolchain, as supported by the MCUXpresso SDK. There are many ways to use Arm GCC tools, but this example focuses on a Windows operating system environment.

#### 6.1.1 Install GCC Arm Embedded tool chain

Download and run the installer from GNU Arm Embedded Toolchain. This is the actual toolset (in other words, compiler, linker, and so on). The GCC toolchain should correspond to the latest supported version, as described in *MCUXpresso SDK Release Notes*.

#### 6.1.2 Install MinGW (only required on Windows OS)

The Minimalist GNU for Windows (MinGW) development tools provide a set of tools that are not dependent on third-party C-Runtime DLLs (such as Cygwin). The build environment used by the MCUXpresso SDK does not use the MinGW build tools, but does leverage the base install of both MinGW and MSYS. MSYS provides a basic shell with a Unix-like interface and tools.

- 1. Download the latest MinGW mingw-get-setup installer from MinGW.
- 2. Run the installer. The recommended installation path is C: \MinGW, however, you may install to any location.

**NOTE** The installation path cannot contain any spaces.

3. Ensure that the mingw32-base and msys-base are selected under Basic Setup.

Installation Package Settings						
Basic Setup	Packag	je	Class	Installed Version	Repository Version	Description
All Packages	mingw	-developer-tool	bin		2013072300	An MSYS Installation for MinGW Developers (meta
	🐑 mingw	32-base	bin		2013072200	A Basic MinGW Installation
	mingw	32-gcc-ada	bin		4.8.1-4	The GNU Ada Compiler
	mingw	32-gcc-fortran	bin		4.8.1-4	The GNU FORTRAN Compiler
	mingw	32-gcc-g++	bin		4.8.1-4	The GNU C++ Compiler
	mingw	32-gcc-objc	bin		4.8.1-4	The GNU Objective-C Compiler
	🧐 msys-t	base	bin		2013072300	A Basic MSYS Installation (meta)

4. In the Installation menu, click Apply Changes and follow the remaining instructions to complete the installation.

Installation Package S	ettings				
Update Catalogue		Package			
Mark All Upgrades		mingw-developer-tool			
Apply Changes		mingw32-base			
	700005	mingw32-gcc-ada			
Quit	Alt+F4	mingw32-gcc-fortran			
		mingw32-gcc-g++			
		mingw32-gcc-objc msys-base			
	11	🐑 msys-base			

5. Add the appropriate item to the Windows operating system path environment variable. It can be found under **Control Panel->System and Security->System->Advanced System Settings** in the **Environment Variables...** section. The path is:

#### <mingw\_install\_dir>\bin

Assuming the default installation path, C:\MinGW, an example is shown below. If the path is not set correctly, the toolchain will not work.

#### NOTE

If you have C:MinGWMsysX.xbin in your PATH variable (as required by Kinetis SDK 1.0.0), remove it to ensure that the new GCC build system works correctly.

S	ystem Properties
Į	Computer Name Hardware Advanced System Protection Remote
	Environment Variables
	Edit System Variable
	Variable name: Path
	Variable value: 2gram Files (x86) \CMake\bin;C: \MinGW\bin
	OK Cancel
	System variables
	Variable Value
	OS Windows_NT Path C:\Program Files (x86)\Parallels\Parallel
	PATH C: program Piles (xob) parallels parallels PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;
	PROCESSOR_A AMD64
	New Edit Delete
_	OK Cancel
re 50. Add Path to systems er	vironment

#### 6.1.3 Add a new system environment variable for ARMGCC\_DIR

Create a new *system* environment variable and name it as ARMGCC\_DIR. The value of this variable should point to the Arm GCC Embedded tool chain installation path. For this example, the path is:

```
C:\Program Files (x86)\GNU Tools Arm Embedded\8 2018-q4-major
C:\Program Files (x86)\GNU Tools Arm Embedded\7-2018-q2
```

See the installation folder of the GNU Arm GCC Embedded tools for the exact path name of your installation.

Short path should be used for path setting, you could convert the path to short path by running command for %I in (.) do echo %~sI in above path.

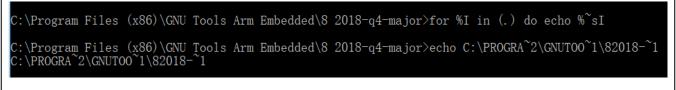


Figure 51. Convert path to short path

Environment Variables	×
User variables for	
Variable	Value
OneDrive	C:\Users\ \OneDrive - NXP
OneDriveConfimercial	C:\Users\ \OneDrive - NXP
Path	C:\Ruby24-x64\bin;C:\Users\nxa07599\AppData\Local\Micros
PATHEXT	.COM;:EXE;:BAT;.CMD;:VBS;:VBE;JS;JSE;:WSF;:WSH;:MSC;:RB;:RB
TEMP	C:\Users\ \AppData\Local\Temp
TMP	C:\Users\ \AppData\Local\Temp
New User Variable	×
	C_DIR RA~2\GNUTOO~1\82018-~1 Browse File C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.2 C:\Program Files (x86)\SEGGER\JLink_V640 C:\Keil_v5\UV4 No
	New Edit Delete
RMGCC_DIR system varia	

#### 6.1.4 Install CMake

- 1. Download CMake 3.0.x from www.cmake.org/cmake/resources/software.html.
- 2. Install CMake, ensuring that the option Add CMake to system PATH is selected when installing. The user chooses to select whether it is installed into the PATH for all users or just the current user. In this example, it is installed for all users.

	CMake 3.0.2 Setup		
	Install Options Choose options for installing CMake 3.0.2		
	By default CMake does not add its directory to the system PATH.		
	Do not add CMake to the system PATH O Add CMake to the system PATH for all users Add CMake to the system PATH for current user		
	Create CMake Desktop Icon		
	Nullsoft Install System v2.46	Cancel	
Figure 53. Install CMake			

- 3. Follow the remaining instructions of the installer.
- 4. You may need to reboot your system for the PATH changes to take effect.
- 5. Make sure sh.exe is not in the Environment Variable PATH. This is a limitation of mingw32-make.

### 6.2 Build an example application

To build an example application, follow these steps.

1. Open a GCC Arm Embedded tool chain command window. To launch the window, from the Windows operating system **Start** menu, go to **Programs >GNU Tools Arm Embedded <version>** and select **GCC Command Prompt**.

	GNU Tools for ARM Embedded Process
	Documentation
	🚳 GCC Command Prompt
	🎯 Uninstall GNU Tools for ARM Embec
Figure 54. Launch command prompt	

2. Change the directory to the example application project directory which has a path similar to the following:

<install\_dir>/boards/<board\_name>/<example\_type>/<application\_name>/armgcc

For this example, the exact path is:

<install\_dir>/examples/lpcxpresso55s16/demo\_apps/hello\_world/armgcc

NOTE

To change directories, use the  ${\tt cd}$  command.

3. Type **build\_debug.bat** on the command line or double click on **build\_debug.bat** file in Windows Explorer to build it. The output is as shown in Figure 55.

<pre>[ 842] Building C object CMakeFiles/hello_world.elf.dir/C_/nxp/SDK_2.0_FRDM-K64F /devices/MK64F12/drivers/fsl_smc.c.obj [ 922] Building C object CMakeFiles/hello_world.elf.dir/C_/nxp/SDK_2.0_FRDM-K64F /devices/MK64F12/drivers/fsl_clock.c.obj [100%] Linking C executable debug\hello_world.elf [100%] Built target hello_world.elf [100%] Built target hello_world.elf C:\nxp\SDK_2.0_FRDM-K64F\boards\frdmk64f\demo_apps\hello_world\armgcc&gt;IF "" == " " (pause ) Press any key to continue</pre>
[ 85%] Building C object CMakeFiles/hello_world.elf.dir/C_/Users/nxf55082/Downloads/board_LPCXpresso55S16/components/ser
ial_manager/serial_manager.c.obj [ 90%] Building C object CMakeFiles/hello_world.elf.dir/C_/Users/nxf55082/Downloads/board_LPCXpresso55S16/components/lis ts/generic list.c.obj
[96%] Building C object CMakeFiles/hello_world.elf.dir/C_/Users/nxf55082/Downloads/board_LPCXpresso55S16/components/ser ial_manager/serial_port_uart.c.obj [100%] Linking C executable debug\hello_world.elf [100%] Built target hello_world.elf
C:\Users\nxf55082\Downloads\board_LPCXpresso55S16\boards\lpcxpresso55s16\demo_apps\hello_world\armgcc>IF "" == "" (pause
Préss any key to continue
Figure 56. hello_world demo build successful

### 6.3 Run an example application

This section describes steps to run a demo application using J-Link GDB Server application. To update the on-board LPC-Link2 debugger to Jlink firmware, see Updating debugger firmware.

NOTE J-Link GDB Server application is not supported for TFM examples. Use CMSIS DAP instead of J-Link for flashing and debugging TFM examples.

After the J-Link interface is configured and connected, follow these steps to download and run the demo applications:

- Connect the development platform to your PC via USB cable between the LPC-Link2 USB connector (may be named OSJTAG for some boards) and the PC USB connector. If using a standalone J-Link debug pod, connect it to the SWD/JTAG connector of the board.
- 2. Open the terminal application on the PC, such as PuTTY or TeraTerm, and connect to the debug serial port number (to determine the COM port number, see How to determine COM port). Configure the terminal with these settings:
  - a. 115200 or 9600 baud rate, depending on your board (reference <code>BOARD\_DEBUG\_UART\_BAUDRATE</code> variable in <code>board.h</code> file)
  - b. No parity
  - c. 8 data bits
  - d. 1 stop bit

- Session	Basic options for you	Ir PuTTY session
Logging	Specify the destination you wa	ant to connect to
- Terminal	Serial line COM16	Speed 115200
Bell Features Window	Connection type:	
Appearance Behaviour Translation	Load, save or delete a stored Saved Sessions	session
Selection	Debug	
Colours	Default Settings Debug	Load
- Data	Debug	Save
Proxy Telnet		Delete
		<u></u>
Serial	Close window on exit: Always Never	Only on clean exit

#### NOTE

Make sure the board is set to FlexSPI flash boot mode (ISP2: ISP1: ISP0 = ON, OFF, ON) before use GDB debug.

- Open the J-Link GDB Server application. Assuming the J-Link software is installed, the application can be launched by going to the Windows operating system Start menu and selecting Programs -> SEGGER -> J-Link <version> J-Link GDB Server.
- 4. After it is connected, the screen should look like this figure:

#### Run a demo using Arm<sup>®</sup> GCC

	SEGG	ER J-Link GDB Serve	r V6.46g		- 🗆 X	
	File Help	)				
	GDB	Waiting for connection	on		Stay on top	
	J-Link	Connected	SWD	4000 kHz	Show log window	
	Device	L3A60xxx_M4 (Halte	ed) 3.29V	little endian	Generate logfile	
					Verify download	
	Clear L	og				
	Hardwar S/N: 36 Checkin Target Listeni Connect Connect	re: J-Link Lite-F re: V1.00 51000738 ug target voltage voltage: 3.29 V ing on TCP/IP por ing to target red to target g for GDB connect	9 rt 2331	Jun 25 2012 16:40:07	~	
	0 bytes do	wnloaded		Conne	cted to target	
	1.00					
Figure 58. SEGGEF	R J-Link GI	DB Server sci	reen after su	ccessful connectio	n	
	-	nk GDB Server V6	.54c		—	×
File	e Help		.54c		_	×
File	e Help GDB Waiting	g for connection		4000 1417	Stay on top	
File G J-	e Help GDB Waiting J-Link Connec	g for connection	SWD	4000 kHz	Stay on top	low
File G J-	e Help GDB Waiting	g for connection		4000 kHz little endian	Stay on top	low
File G J-	e Help GDB Waiting J-Link Connec	g for connection	SWD		Stay on top	low
File G J- D	e Help GDB Waiting J-Link Connec	g for connection	SWD		Stay on top	low
File G J- D S/ Fe Ch Ta Li C O CO	GDB Waiting GDB Waiting D-Link Connec Device Unspec Clear Log /N: 60011193 eature (s): 1 hecking targ arget voltag istening on onnecting to	34 RDI, FlashBP, get voltage ge: 3.30 V TCP/IP port 2 o target	SWD 3.30V FlashDL, JF1 2331	little endian	Stay on top	low
File G J- D S/ Fe Ch Ta Li Co Co Wa	GDB Waiting GDB Waiting I-Link Connec Device Unspec Clear Log /N: 6001119: eature (s): I hecking targarget voltag istening on onnecting to onnected to aiting for (	g for connection ted ified (Halted) 34 RDI, FlashBP, get voltage ge: 3.30 V TCP/IP port 2 o target target GDB connection	SWD 3.30V FlashDL, JF1 2331	ash, GDB	Stay on top	low le hd
File G J- D S/ Fe Ch Ta Li Co Co Wa	GDB Waiting GDB Waiting D-Link Connec Device Unspec Clear Log /N: 60011192 eature (s): I hecking tarr arget voltag istening on onnecting to onnecting to onnecting for ( ytes download	g for connection tted tified (Halted) 34 RDI, FlashBP, get voltage ge: 3.30 V TCP/IP port 2 o target target GDB connection ded	SWD 3.30V FlashDL, JFl 2331 n	ash, GDB	Stay on top	low le hd

5. If not already running, open a GCC Arm Embedded tool chain command window. To launch the window, from the Windows operating system Start menu, go to Programs -> GNU Tools Arm Embedded <version> and select GCC Command Prompt.

	GNU Tools for ARM Embedded Process Documentation
	🚳 GCC Command Prompt
	🗑 Uninstall GNU Tools for ARM Embec
Figure 60. Launch command promp	t

6. Change to the directory that contains the example application output. The output can be found in using one of these paths, depending on the build target selected:

<install\_dir>/boards/<board\_name>/<example\_type>/<application\_name>/armgcc/debug

<install\_dir>/boards/<board\_name>/<example\_type>/<application\_name>/armgcc/release

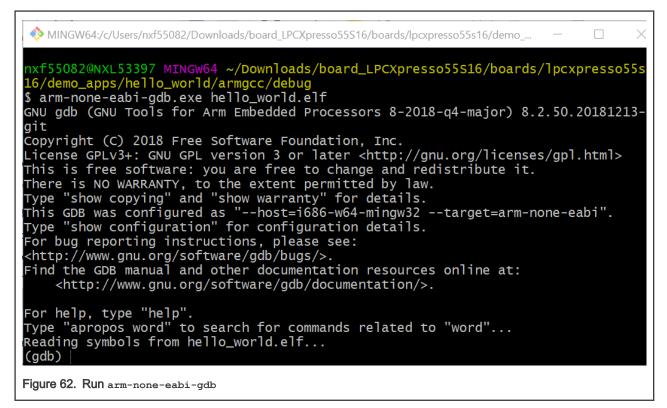
#### For this example, the path is:

<install\_dir>/boards/frdmk32l3a6/demo\_apps/hello\_world/cm4/armgcc/debug

<install dir>/boards/lpcxpresso55s16/demo apps/hello world/armgcc/debug

7. Run the arm-none-eabi-gdb.exe <application\_name>.elf command. For this example, it is arm-none-eabi-gdb.exe hello world.elf.

GCC Command Prompt - arm-none-eabi-gdb.exe C:\Users\nxa12829\Desktop\k32I3\boards\frdmk32I3a6\demo_apps\hello_world\cm4\armgcc\debu — 🛛 🗙
C:\Program Files (x86)\GNU Tools ARM Embedded\8 2018-q4-major>arm-none-eabi-gdb.exe C:\Users\nxa12829\Desktop\k3213\boar ds\frdmk3213a6\demo_apps\hello_world\cm4\armgcc\debug\hello_world_demo_cm4.elf C:\Program Files (x86)\GNU Tools ARM Embedded\8 2018-q4-major\bin\arm-none-eabi-gdb.exe: warning: Couldn't determine a p ath for the index cache directory. GNU gdb (GNU Tools for Arm Embedded Processors 8-2018-q4-major) 8.2.50.20181213-git Copyright (C) 2018 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <http: gnu.org="" gpl.html="" licenses=""> This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law. Type "show configured as "host=i686-w64-mingw32target=arm-none-eabi". Type "show configuration" for configuration details. For bug reporting instructions, please see: <http: bugs="" gdb="" software="" www.gnu.org=""></http:>. Find the GDB manual and other documentation/&gt;.</http:>
For help, type "help". Type "apropos word" to search for commands related to "word" Reading symbols from C:\Users\nxa12829\Desktop\k3213\boards\frdmk3213a6\demo_apps\hello_world\cm4\armgcc\debug\hello_wor ld_demo_cm4.elf (gdb) Figure 61. Run arm-none-eabi-gdb



#### 8. Run these commands:

- **a**. target remote localhost:2331
- $b. \ \text{monitor reset}$
- $\boldsymbol{C}.$  monitor halt
- **d**. load
- e. monitor reset
- 9. The application is now downloaded and halted at the watch point. Execute the monitor go command to start the demo application.

The hello\_world application is now running and a banner is displayed on the terminal. If this does not appear, check your terminal settings and connections.

COM3 - PuTTY	
hello world. T	
Figure 63. Text display of the hello_world demo	

### 6.4 Build a TrustZone example application

This section describes the steps to build and run a TrustZone application. The demo application build scripts are located in this folder:

```
<install_dir>/boards/<board_name>/trustzone_examples/<application_name>/<core_type>/iar/<application_name>_ns/armgcc
```

<install\_dir>/boards/<board\_name>/trustzone\_examples/<application\_name>/iar/<application\_name>\_ns/ armgcc

<install\_dir>/boards/<board\_name>/trustzone\_examples/<application\_name>/<core\_type>/iar/<application\_name>\_s/armgcc

<install\_dir>/boards/<board\_name>/trustzone\_examples/<application\_name>/iar/<application\_name>\_s/ armgcc

Begin with a simple TrustZone version of the Hello World application. The TrustZone Hello World GCC build scripts are located in this folder:

```
<install_dir>/boards/lpcxpresso55s16/trustzone_examples/hello_world/hello_world_ns/iar/
hello_world_ns.eww
```

<install\_dir>/boards/lpcxpresso55s16/trustzone\_examples/hello\_world/hello\_world\_s/iar/ hello\_world\_s.eww

<install dir>/boards/lpcxpresso55s16/trustzone examples/hello world/hello world s/iar/hello world.eww

Build both applications separately, following steps for single core examples as described in *Section 6.2, "Build an example application"*. It is requested to build the application for the secure project first, because the non-secure project needs to know the secure project, since CMSE library is running the linker. It is not possible to finish the non-secure project linker with the secure project because the CMSE library is not ready.

	_	_	
C:\WINDOWS\system32\cmd.exe	_		$\times$
			^
[ 47%] Building ASM object CMakeFiles/hello_world_s.elf.dir/2dae3411939747f3d06776b994f98cc9/LPC55S69/gcc/st 69 cm33 core0.S.obj		p_LPC	:558
[52%] Building C object CMakeFiles/hello_world_s.elf.dir/4579822925539bldclc691983e2db5fe/devices/LPC55S69/ usart.c.obj			`s1_
usaru.c.opj [ 56%] Building C object CMakeFiles/hello_world_s.elf.dir/4579822925539b1dc1c691983e2db5fe/devices/LPC55S69/			-61
flexcomm.c.obj			
[ 60%] Building C object CMakeFiles/hello world s.elf.dir/4579822925539b1dc1c691983e2db5fe/devices/LPC55S69/			s1_
gpio.c.obj □[[65%] Building C object CMakeFiles/hello_world_s.elf.dir/4579822925539b1dc1c691983e2db5fe/devi			5869
/utilities/fsl_assert.c.obj			
[ 69%] Building C object CMakeFiles/hello_world_s.elf.dir/4579822925539b1dc1c691983e2db5fe/devices/LPC55S69/			/st
r/fsl_str.c.obj [ 73%] Building C object CMakeFiles/hello_world_s.elf.dir/93b83a68a14fbf36e09b516617c5bee5/utilities/debug_c			d a
[ 75%] burruing c object cmakerires/nerro_worru_s.err.urr/95063a08a14r0r50e09051001(c50ee3/utrifities/debug_c		16/15	u_u
[78%] Building C object CMakeFiles/hello_world_s.elf.dir/C_/Users/nxf05192/Downloads/LPCXpresso55S69_227/co	mpon	ents/	uar
/usart_adapter.c.obj			
[ 82%] Building C object CMakeFiles/hello_world_s.elf.dir/4579822925539b1dc1c691983e2db5fe/components/serial			ser
al_manager.c.obj			
[ 86%] Building C object CMakeFiles/hello_world_s.elf.dir/4579822925539bldclc691983e2db5fe/components/serial			ser
ial_port_uart.c.obj [ 91%] Building C object CMakeFiles/hello world s.elf.dir/C /Users/nxf05192/Downloads/LPCXpresso55569 227/cc			1.5 -
ts/generic list.c.obj	Juipon		118
[95] Building C object CMakeFiles/hello world s.elf.dir/4579822925539b1dc1c691983e2db5fe/devices/LPC55S69/	driv	ers/f	s1
reset.c.obj			
[100%] Linking C executable debug\hello_world_s.elf			
[100%] Built target hello_world_s.elf			
C:\Users\nxf05192\Downloads\LPCXpresso55869_227\boards\1pcxpresso55s69\trustzone_examples\hello_world\cm33_c orld s\armgcc>IF "" == "" (pause )	orev	\ne11	.o_w
Press any key to continue			- U

### Figure 64. hello\_world\_s example build successful

C:\WINDOWS\system32\cmd.exe	—		$\times$
[ 47%] Building C object CMakeFiles/hello_world_s.elf.dir/C_/Users/nxf55082/Downloads/board_LPCXpresso555	6/devi	ices/Ll	PC5 ^
5S16/system_LPC55S16.c.obj			
[ 52%] Building ASM object CMakeFiles/hello_world_s.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/devices/LPC5	S16/gc		rtu
p_LPC55S16.S.obj [ 56%] Building C object CMakeFiles/hello world s.elf.dir/C /Users/nxf55082/Downloads/board LPCXpresso55S1		i.coc/L	005
5316/drivers/fsl usart.c.obj		ICES/LI	00
[ 60%] Building C object CMakeFiles/hello_world_s.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/devices/LPC55S		vers/f:	s1
flexcomm.c.obj			
[ 65%] Building C object CMakeFiles/hello_world_s.elf.dir/C_/Users/nxf55082/Downloads/board_LPCXpresso5551		ices/Ll	PC5
5S16/drivers/fsl_gpio.c.obj		1242	18-
[ 69%] Building C object CMakeFiles/hello_world_s.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/devices/LPC55S1 assert.c.obj	6/ut11	lities,	IS
[73%] Building C object CMakeFiles/hello world s.elf.dir/aff6f96f781e03199b967c8ecf81a7b8/LPC55S16/utilit			ons
ole/fsl_debug_console.c.obj			
[ 78%] Building C object CMakeFiles/hello_world_s.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/devices/LPC55S		lities,	/st
r/fsl str. c. obj			,
[ 82%] Building C object CMakeFiles/hello_world_s.elf.dir/C_/Users/nxf55082/Downloads/board_LPCXpresso5551 art/usart adapter.c.obj	6/comp	ponent:	3/ U
[ 86%] Building C object CMakeFiles/hello_world_s.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/components/seri			ser
ial manager. c. obj			
[ 91%] Building Č object CMakeFiles/hello_world_s.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/components/seri			$\operatorname{ser}$
ial_port_uart.c.obj			1.0
[ 95%] Building C object CMakeFiles/hello_world_s.elf.dir/C_/Users/nxf55082/Downloads/board_LPCXpresso555]	.6/com		\$/1
ists/generic_list.c.obj [100%] Linking C executable debug\hello world s.elf			
[100%] Built target hello world s. elf			
$\label{eq:c:users_nxf550g2} Downloads board_LPCX pressof5516 boards lpcx pressof5516 trust zone_examples hello_world hello_w$	110_wc	orld_s	\ar
mgcc>IF "" == "" (pause )			
Press any key to continue			$\sim$
Figure 65. hello_world_s example build successful			

C:\WINDOWS\system32\cmd.exe -	Π	×
	tup_LP	
S69_cm33_core0.S.obj [ 47%] Building C object CMakeFiles/hello_world_ns.elf.dir/4579822925539b1dc1c691983e2db5fe/devices/LPC55S69/dr		fel
_usart.c.obj[ 52%]		
Building C object CMakeFiles/hello_world_ns.elf.dir/4579822925539bldclc691983e2db5fe/devices/LPC55S69/drivers/f mm.c.obj		xco
[ 57%] Building C object CMakeFiles/hello_world_ns.elf.dir/4579822925539b1dc1c691983e2db5fe/devices/LPC55S69/dr		fs1
_gpio.c.obj [ 61%] Building C object CMakeFiles/hello_world_ns.elf.dir/4579822925539b1dc1c691983e2db5fe/devices/LPC55S69/ut	ilitie	es/f
sl_assert.c.obj [ 66%] Building C_object CMakeFiles/hello world ns.elf.dir/4579822925539b1dc1c691983e2db5fe/devices/LPC55S69/ut		
tr/fsl_str.c.obj	IIICIE	575
[71%] Building C_object CMakeFiles/hello_world_ns.elf.dir/93b83a68a14fbf36e09b516617c5bee5/utilities/debug_con		s1_
debug_console.c.obj [ <b>76%]</b> Building C object CMakeFiles/hello_world_ns.elf.dir/4579822925539b1dc1c691983e2db5fe/components/uart/usa		pte
r.c.obj [ 80%] Building C object CMakeFiles/hello_world_ns.elf.dir/4579822925539bldc1c691983e2db5fe/components/serial_m		/se
rial_manager.c.obj [ 85%] Building C object CMakeFiles/hello world ns.elf.dir/4579822925539b1dc1c691983e2db5fe/components/serial π		:/se
rial port_uart.c.obj [ 90%] Building C object CMakeFiles/hello world ns.elf.dir/4579822925539b1dc1c691983e2db5fe/components/lists/ge	neric	110
t.c.obj		
[ 95%] Building C object CMakeFiles/hello_world_ns.elf.dir/4579822925539bldc1c691983e2db5fe/devices/LPC55S69/dr Freset.c.obj		fsl
[100%] Linking C executable debug\hello_world_ns.elf [100%] Built target hello_world_ns.elf		
C:\Users\nxf05192\Downloads\LPCXpresso55S69_227\boards\1pcxpresso55s69\trustzone_examples\hello_world\cm33_core orld_ns\armgcc>IF "" == "" (pause )	0\he11	.o_w

#### Press any key to continue . .

#### Figure 66. hello\_world\_ns example build successful

C:\WINDOWS\system32\cmd.exe	- [	$\square$ $\times$	
[ 42%] Building C object CMakeFiles/hello_world_ns.elf.dir/C_/Users/nxf55082/Downloads/board_LPCXpresso55	16/devi	ces/LPC	^
55S16/system_LPC55S16.c.obj		- / - + +	
[ 47%] Building ASM object CMakeFiles/hello_world_ns.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/devices/LPC3 up LPC55S16.S.obj	5210/gc	c/start	
[ 52%] Building Č object CMakeFiles/hello_world_ns.elf.dir/C_/Users/nxf55082/Downloads/board_LPCXpresso55		ces/LPC	
55516/drivers/fsl_usart.c.obj		10.1	
[ 57%] Building C object CMakeFiles/hello_world_ns.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/devices/LPC55S flexcomm.c.obj		ers/isl	
[ 61%] Building C object CMakeFiles/hello_world_ns.elf.dir/C_/Users/nxf55082/Downloads/board_LPCXpresso55	16/devi	ces/LPC	
55816/drivers/fsl_gpio.c.obj			
[66%] Building C object CMakeFiles/hello_world_ns.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/devices/LPC55	16/util	ities/f	
sl_assert.c.obj [ 71%] Building C object CMakeFiles/hello_world_ns.elf.dir/aff6f96f781e03199b967c8ecf81a7b8/LPC55S16/util:		biig con	
sole/fsl debug console.c.obj		24 <u>8</u> _00m	
[ 76%] Building C object CMakeFiles/hello_world_ns.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/devices/LPC55		ities/s	
tr/fsl_str.c.obj [ 80%] Building C object CMakeFiles/hello world ns.elf.dir/C /Users/nxf55082/Downloads/board LPCXpresso55	16/comp	onents/	
uart/usart_adapter.c.obj			
[85%] Building C object CMakeFiles/hello_world_ns.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/components/ser		ager/se	
rial_manager.c.obj <b>[ 90%]</b> Building C object CMakeFiles/hello world ns.elf.dir/2d39d49956e612f966ff1b9d5ddcfbcd/components/se	ial man	ager/se	
rial_port_uart.c.obj			
[ 95%] Building C object CMakeFiles/hello_world_ns.elf.dir/C_/Users/nxf55082/Downloads/board_LPCXpresso55		onents/	
lists/generic_list.c.obj [100%] Linking C executable debug\hello world ns.elf			
[100%] Built target hello world ns.elf			
C:\Users\nxf55082\Downloads\board_LPCXpresso55S16\boards\lpcxpresso55s16\trustzone_examples\hello_world\he rmgcc>IF "" == "" (pause )	llo_wor	ld_ns∖a	
Press any key to continue			~
Figure 67. hello_world_ns example build successful			

### 6.5 Run a TrustZone example application

When running a TrustZone application, the same prerequisites for J-Link/J-Link OpenSDA firmware, and the serial console as for the single core application, apply, as described in *Section 6.3, "Run an example application"*.

To download and run the TrustZone application, perform steps 1 to 10, as described in *Section 5.3, "Run an example application"*. These steps are common for both single core and trustzone applications in Arm GCC.

#### Then, run these commands:

- 1. arm-none-eabi-gdb.exe
- 2. target remote localhost:2331
- 3. monitor reset
- monitor halt
- 5. monitor exec SetFlashDLNoRMWThreshold = 0x20000
- 6. load <install\_dir>/boards/lpcxpresso55s16/trustzone\_examples/hello\_world/hello\_world\_ns/armgcc/ debug/hello\_world\_ns.elf
- 7. load <install\_dir>/boards/lpcxpresso55s16/trustzone\_examples/hello\_world/hello\_world\_s/armgcc/ debug/hello\_world\_s.elf

8.

The application is now downloaded and halted at the watch point. Execute the monitor go command to start the demo application.

arm-none-eabi-gdb.exe	_		×
<pre>C:\Program Files (x80)\GNU Tools Arm Embedded\7 2018-q2-update&gt;arm=none=eabi-gdb.exe GNU gob (GNU Tools for Arm Embedded Processors 7-2018-q2-update) 8.1.0.20180315-git Copyright (C) 2018 Free Software Foundation, Inc. License CPLv3+: GNU GPL version 3 or later Chttp://gnu.org/licenses/gpl.html&gt; There is NO WARKANT, to the extent permitted by law. Type "show copying" and "show warranty" for details. This GDB was configured as "-host=1636~w64-mingw32target=arm=none-eabi". Type "show configuration for configuration details. For bug reporting instructions, please see: Chttp://www.gnu.org/software/gdb/bugs/&gt; Find the GDB manual and other documentation resources online at: Chttp://www.gnu.org/software/gdb/bugs/&gt; Find the GDB manual and other documentation /&gt;. For help. type "help". Type "apropos word" to search for commands related to "word". (gdb) target remote localhost:2331 warning: No executable has been specified and target does not support determining executable automatically. Try using the "file" command. 0x0001055 in ?? () (gdb) monitor reset Resetion .interrupts, size 0x140 inm 0x10000 Loading section .interrupts, size 0x140 inm 0x10000 Loading section .interrupts, size 0x41 mm 0x11880 Loading section .int_array, size 0x41 mm 0x10800 Loading section .int_array, size 0x41 mm 0x10800 Loading section .int_array, size 0x41 mm 0x10800 Loading section .int_array, size 0x41 mm 0x100001 Loading section .int_array, size 0x41 mm 0x1000036a Loading section .int_arra</pre>			
P COM57 - PuTTY -	Г	1	×
<u> </u>		-	<u> </u>
Hello from secure world! Entering normal world.			
Welcome in normal world!			
This is a text printed from normal world! Comparing two string as a callback to normal world			
String 1: Testl			
String 2: Test2			
Both strings are not equal!			
Figure 69. Text display of the trustzone hello_world application			

# 7 MCUXpresso Config Tools

MCUXpresso Config Tools can help configure the processor and generate initialization code for the on chip peripherals. The tools are able to modify any existing example project, or create a new configuration for the selected board or processor. The generated code is designed to be used with MCUXpresso SDK version 2.x.

Table 1 describes the tools included in the MCUXpresso Config Tools.

Table 1. MCUXpresso Config Tools

Config Tool	Description	Image
Pins tool	For configuration of pin routing and pin electrical properties.	
Clock tool	For system clock configuration	
Peripherals tools	For configuration of other peripherals	<b>(</b>
TEE tool	Configures access policies for memory area and peripherals helping to protect and isolate sensitive parts of the application.	$\bigcirc$
Device Configuration tool	Configures Device Configuration Data (DCD) contained in the program image that the Boot ROM code interprets to setup various on-chip peripherals prior the program launch.	<b>*</b>

MCUXpresso Config Tools can be accessed in the following products:

- Integrated in the MCUXpresso IDE. Config tools are integrated with both compiler and debugger which makes it the easiest way to begin the development.
- Standalone version available for download from www.nxp.com/mcuxpresso. Recommended for customers using IAR Embedded Workbench, Keil MDK µVision, or Arm GCC.
- Online version available on mcuxpresso.nxp.com. Recommended to do a quick evaluation of the processor or use the tool without installation.

Each version of the product contains a specific *Quick Start Guide* document MCUXpresso IDE Config Tools installation folder that can help start your work.

# 8 MCUXpresso IDE New Project Wizard

MCUXpresso IDE features a new project wizard. The wizard provides functionality for the user to create new projects from the installed SDKs (and from pre-installed part support). It offers user the flexibility to select and change multiple builds. The wizard also includes a library and provides source code options. The source code is organized as software components, categorized as drivers, utilities, and middleware.

To use the wizard, start the MCUXpresso IDE. This is located in the **QuickStart Panel** at the bottom left of the MCUXpresso IDE window. Select **New project**, as shown in Figure 70.

U Quickstart Panel	🕪 = Global Variables 🔅 = Variables 💁 Breakpoints 🗄 = Outline	
	esso IDE (Free Edition)	*
▼ Start here		
🛛 New project		
🔀 Import SDK e	xample(s)	
🖲 Import projec	ct(s) from file system	=
🔏 Build " []		
🖌 Clean " []		
🎋 Debug " []		
🎋 Terminate, Bu	ild and Debug " []	
🕲 Edit '' project	settings	
🖉 Quick Setting	\$>>	-
MCUXpresso IDE Quicksta	art Panel	

For more details and usage of new project wizard, see the *MCUXpresso\_IDE\_User\_Guide.pdf* in the MCUXpresso IDE installation folder.

# 9 How to determine COM port

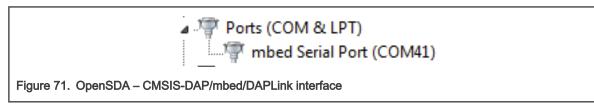
This section describes the steps necessary to determine the debug COM port number of your NXP hardware development platform. All NXP boards ship with a factory programmed, on-board debug interface, whether it's based on OpenSDA or the legacy P&E Micro OSJTAG interface. To determine what your specific board ships with, see #unique\_37.

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.
- 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.
  - a. OpenSDA CMSIS-DAP/mbed/DAPLink interface:



b. OpenSDA - P&E Micro:

Ports (COM & LPT)

Figure 72. OpenSDA – P&E Micro

c. OpenSDA – J-Link:

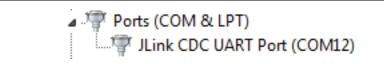


Figure 73. OpenSDA – J-Link

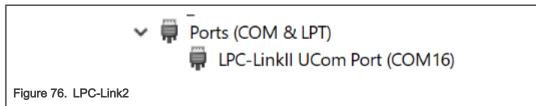
d. P&E Micro OSJTAG:

Ports (COM & LPT) OSBDM/OSJTAG - CDC Serial Port (http://www.pemicro.com/osbdm, http://www.pemicro.com/opensda) (COM43) Figure 74. P&E Micro OSJTAG

e. MRB-KW01:



f. LPC-Link2:



### 10 How to define IRQ handler in CPP files

With MCUXpresso SDK, users could define their own IRQ handler in application level to

override the default IRQ handler. For example, to override the default <code>PIT\_IRQHandler</code> define in <code>startup\_DEVICE.s</code>, application code like app.c can be implement like:

```
c
void PIT_IRQHandler(void)
{
    // Your code
}
```

When application file is CPP file, like app.cpp, then extern "C" should be used to ensure the function prototype alignment.

```
cpp
extern "C" {
    void PIT IRQHandler(void);
```

Getting Started with MCUXpresso SDK for LPCXpresso55S16, Rev. 2.10.1, 09 September 2021

}

```
void PIT_IRQHandler(void)
{
    // Your code
}
```

# 11 Default debug interfaces

The MCUXpresso SDK supports various hardware platforms that come loaded with a variety of factory programmed debug interface configurations. Table 2 lists the hardware platforms supported by the MCUXpresso SDK, their default debug interface, and any version information that helps differentiate a specific interface configuration.

NOTE The OpenSDA details column in Table 2 is not applicable to LPC.

Table 2.	Hardware	platforms	supported	by I	MCUXpresso SDK
----------	----------	-----------	-----------	------	----------------

Hardware platform	Default interface	OpenSDA details
EVK-MC56F83000	P&E Micro OSJTAG	N/A
EVK-MIMXRT595	CMSIS-DAP	N/A
EVK-MIMXRT685	CMSIS-DAP	N/A
FRDM-K22F	CMSIS-DAP/mbed/DAPLink	OpenSDA v2.1
FRDM-K28F	DAPLink	OpenSDA v2.1
FRDM-K32L2A4S	CMSIS-DAP	OpenSDA v2.1
FRDM-K32L2B	CMSIS-DAP	OpenSDA v2.1
FRDM-K32W042	CMSIS-DAP	N/A
FRDM-K64F	CMSIS-DAP/mbed/DAPLink	OpenSDA v2.0
FRDM-K66F	J-Link OpenSDA	OpenSDA v2.1
FRDM-K82F	CMSIS-DAP	OpenSDA v2.1
FRDM-KE15Z	DAPLink	OpenSDA v2.1
FRDM-KE16Z	CMSIS-DAP/mbed/DAPLink	OpenSDA v2.2
FRDM-KL02Z	P&E Micro OpenSDA	OpenSDA v1.0
FRDM-KL03Z	P&E Micro OpenSDA	OpenSDA v1.0
FRDM-KL25Z	P&E Micro OpenSDA	OpenSDA v1.0
FRDM-KL26Z	P&E Micro OpenSDA	OpenSDA v1.0
FRDM-KL27Z	P&E Micro OpenSDA	OpenSDA v1.0
FRDM-KL28Z	P&E Micro OpenSDA	OpenSDA v2.1
FRDM-KL43Z	P&E Micro OpenSDA	OpenSDA v1.0
FRDM-KL46Z	P&E Micro OpenSDA	OpenSDA v1.0
FRDM-KL81Z	CMSIS-DAP	OpenSDA v2.0

Table continues on the next page ...

	,	
Hardware platform	Default interface	OpenSDA details
FRDM-KL82Z	CMSIS-DAP	OpenSDA v2.0
FRDM-KV10Z	CMSIS-DAP	OpenSDA v2.1
FRDM-KV11Z	P&E Micro OpenSDA	OpenSDA v1.0
FRDM-KV31F	P&E Micro OpenSDA	OpenSDA v1.0
FRDM-KW24	CMSIS-DAP/mbed/DAPLink	OpenSDA v2.1
FRDM-KW36	DAPLink	OpenSDA v2.2
FRDM-KW41Z	CMSIS-DAP/DAPLink	OpenSDA v2.1 or greater
Hexiwear	CMSIS-DAP/mbed/DAPLink	OpenSDA v2.0
HVP-KE18F	DAPLink	OpenSDA v2.2
HVP-KV46F150M	P&E Micro OpenSDA	OpenSDA v1
HVP-KV11Z75M	CMSIS-DAP	OpenSDA v2.1
HVP-KV58F	CMSIS-DAP	OpenSDA v2.1
HVP-KV31F120M	P&E Micro OpenSDA	OpenSDA v1
JN5189DK6	CMSIS-DAP	N/A
LPC54018 IoT Module	N/A	N/A
LPCXpresso54018	CMSIS-DAP	N/A
LPCXpresso54102	CMSIS-DAP	N/A
LPCXpresso54114	CMSIS-DAP	N/A
LPCXpresso51U68	CMSIS-DAP	N/A
LPCXpresso54608	CMSIS-DAP	N/A
LPCXpresso54618	CMSIS-DAP	N/A
LPCXpresso54628	CMSIS-DAP	N/A
LPCXpresso54S018M	CMSIS-DAP	N/A
LPCXpresso55s16	CMSIS-DAP	N/A
LPCXpresso55s28	CMSIS-DAP	N/A
LPCXpresso55s69	CMSIS-DAP	N/A
MAPS-KS22	J-Link OpenSDA	OpenSDA v2.0
MIMXRT1170-EVK	CMSIS-DAP	N/A
TWR-K21D50M	P&E Micro OSJTAG	N/AOpenSDA v2.0
TWR-K21F120M	P&E Micro OSJTAG	N/A
TWR-K22F120M	P&E Micro OpenSDA	OpenSDA v1.0
TWR-K24F120M	CMSIS-DAP/mbed	OpenSDA v2.1

Table 2. Hardware platforms supported by MCUXpresso SDK (continued)

Table continues on the next page ...

Hardware platform	Default interface	OpenSDA details
TWR-K60D100M	P&E Micro OSJTAG	N/A
TWR-K64D120M	P&E Micro OpenSDA	OpenSDA v1.0
TWR-K64F120M	P&E Micro OpenSDA	OpenSDA v1.0
TWR-K65D180M	P&E Micro OpenSDA	OpenSDA v1.0
TWR-K65D180M	P&E Micro OpenSDA	OpenSDA v1.0
TWR-KV10Z32	P&E Micro OpenSDA	OpenSDA v1.0
TWR-K80F150M	CMSIS-DAP	OpenSDA v2.1
TWR-K81F150M	CMSIS-DAP	OpenSDA v2.1
TWR-KE18F	DAPLink	OpenSDA v2.1
TWR-KL28Z72M	P&E Micro OpenSDA	OpenSDA v2.1
TWR-KL43Z48M	P&E Micro OpenSDA	OpenSDA v1.0
TWR-KL81Z72M	CMSIS-DAP	OpenSDA v2.0
TWR-KL82Z72M	CMSIS-DAP	OpenSDA v2.0
TWR-KM34Z75M	P&E Micro OpenSDA	OpenSDA v1.0
TWR-KM35Z75M	DAPLink	OpenSDA v2.2
TWR-KV10Z32	P&E Micro OpenSDA	OpenSDA v1.0
TWR-KV11Z75M	P&E Micro OpenSDA	OpenSDA v1.0
TWR-KV31F120M	P&E Micro OpenSDA	OpenSDA v1.0
TWR-KV46F150M	P&E Micro OpenSDA	OpenSDA v1.0
TWR-KV58F220M	CMSIS-DAP	OpenSDA v2.1
TWR-KW24D512	P&E Micro OpenSDA	OpenSDA v1.0
USB-KW24D512	N/A External probe	N/A
USB-KW41Z	CMSIS-DAP\DAPLink	OpenSDA v2.1 or greater

Table 2. Hardware platforms supported by MCUXpresso SDK (continued)

# 12 Updating debugger firmware

### 12.1 Updating LPCXpresso board firmware

The LPCXpresso hardware platform comes with a CMSIS-DAP-compatible debug interface (known as LPC-Link2). This firmware in this debug interface may be updated using the host computer utility called LPCScrypt. This typically used when switching between the default debugger protocol (CMSIS-DAP) to SEGGER J-Link, or for updating this firmware with new releases of these. This section contains the steps to re-program the debug probe firmware.

#### NOTE

If MCUXpresso IDE is used and the jumper making DFUlink is installed on the board (JP5 on some boards, but consult the board user manual or schematic for specific jumper number), LPC-Link2 debug probe boots to DFU mode, and MCUXpresso IDE automatically downloads the CMSIS-DAP firmware to the probe before flash memory programming (after clicking **Debug**). Using DFU mode ensures most up-to-date/compatible firmware is used with MCUXpresso IDE.

NXP provides the LPCScrypt utility, which is the recommended tool for programming the latest versions of CMSIS-DAP and J-Link firmware onto LPC-Link2 or LPCXpresso boards. The utility can be downloaded from www.nxp.com/lpcutilities.

These steps show how to update the debugger firmware on your board for Windows operating system. For Linux OS, follow the instructions described in LPCScrypt user guide (www.nxp.com/lpcutilities, select LPCScrypt, and then the documentation tab).

- 1. Install the LPCScript utility.
- 2. Unplug the board's USB cable.
- 3. Make the DFU link (install the jumper labelled DFUlink).
- 4. Connect the probe to the host via USB (use Link USB connector).
- 5. Open a command shell and call the appropriate script located in the LPCScrypt installation directory (<LPCScrypt install dir>).
  - a. To program CMSIS-DAP debug firmware: <LPCScrypt install dir>/scripts/program\_CMSIS
  - b. To program J-Link debug firmware: <LPCScrypt install dir>/scripts/program\_JLINK
- 6. Remove DFU link (remove the jumper installed in Step 3).
- 7. Re-power the board by removing the USB cable and plugging it in again.

### 13 Revision history

This table summarizes revisions to this document.

#### Table 3. Revision history

Revision number	Date	Substantive changes
0	February 2018	Initial Release
1	June 2019	Updated for MCUXpresso SDK v2.8.0
2	15 January 2021	Updated for MCUXpresso SDK v2.9.0
2.10.0	10 July 2021	Updated for MCUXpresso SDK v2.10.0
2.10.1	09 September 2021	Updated for MCUXpresso SDK v2.10.1

How To Reach Us
Home Page:
nxp.com
Web Support:
nxp.com/support

Limited warranty and liability — 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.

**Right to make changes** - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX,EMBRACE, GREENCHIP, HITAG, 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, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorlQ, QorlQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, elQ, 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-PLUS, ULINKpro, µVision, 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. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2018-2021.

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: 09 September 2021 Document identifier: MCUXSDKLPC551XGSUG

# arm