



# LPC55S69 Quick Start

## *Hands-on 1: [hello\\_world](#)*

Getting Started Guide and Lab Hand Out

---

## Contents

<b>1 Lab Overview</b> .....	<b>2</b>
<b>2 Prerequisites</b> .....	<b>2</b>
<b>3 Software</b> .....	<b>2</b>
3.1 Download MCUXpresso LPCXpresso55S69 SDK .....	3
3.2 Get the latest MCUXpresso IDE.....	4
3.3 Tera Term Tutorial .....	4
3.4 Update Link2 Debugger Firmware.....	5
<b>4 Hardware preparation</b> .....	<b>6</b>
<b>5 SDK Example: <a href="#">hello_world</a></b> .....	<b>8</b>

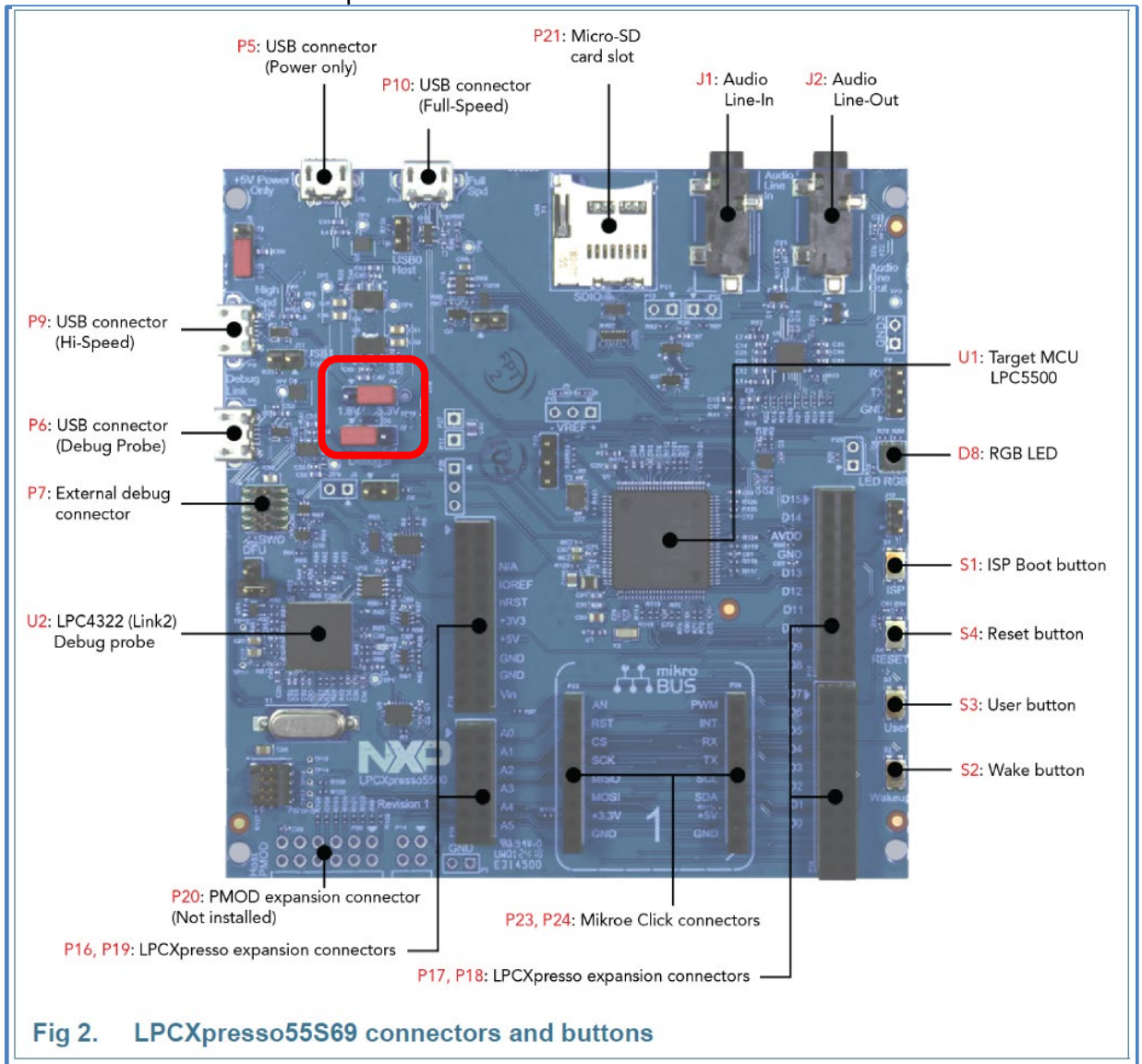
# 1 Lab Overview

This lab will use LPC55S69-EVK and the MCUXpresso SDK example [hello\\_world](#). We will run the example code from flash to demonstrate how SDK projects can be used, and the Hands-on will also confirm that the necessary tools are installed correctly. Sometimes the EVK is referred to by its alternative name LPCxpresso55S69. This document will refer to LPC55S69-EVK (or just “EVK”).

# 2 Prerequisites

The following items are needed to complete this hands-on lab:

- **Board**
  - **LPC55S69-EVK and user-provided micro USB cable.**



# 3 Software

*For Tech Days this software is already installed. – skip to section 4.  
You can use the following instructions to install the software on your own computer.*

### 3.1 Download MCUXpresso LPCXpresso55S69 SDK

1. Open a web browser and navigate to the MCUXpresso homepage [mcuxpresso.nxp.com](http://mcuxpresso.nxp.com)
2. Click on **Select Development Board** to download the LPCXpresso55S69 SDK with AWS component



3. After being redirected to nxp.com login page. Enter your account information or register for a new account.
4. Back on the Select Development Board page, search the **LPCXpresso55S69**, select it and click on **Build MCUXpresso SDK**

#### Select Development Board

Search for your board or kit to get started.



**Search by Name**

**Select a Device, Board, or Kit**

- Boards
  - LPCXpresso55S69
- Kits
  - LPCXPRESSO55S69 with AGM01
  - LPCXpresso55S69-OM13790HOST
- Processors
  - LPC55S66
  - LPC55S69

**Name your SDK**

#### Hardware Details

Board	LPCXpresso55S69
Device	LPC55S69
Core Type / Max Freq	Cortex-M33 / 100MHz Cortex-M33 / 100MHz
Device Memory Size	608 KB Flash 304 KB RAM

#### Additional Links

🔗 If using MCUXpresso IDE, please note that LPCXpresso55S69 requires MCUXpresso IDE v10.3.1 or later

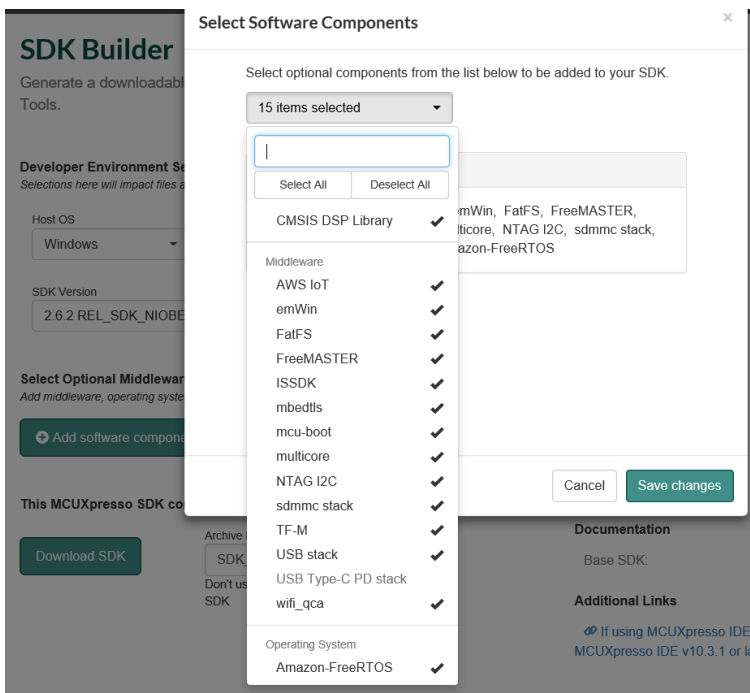
#### Actions

**Build MCUXpresso SDK**

🔧 Explore selection with Pins tool

🕒 Explore selection with Clocks tool

5. At the Select Optional Middleware section, click the + **Add software component**, a window pop-up with a list of components, **Select All** then click **Save Changes**



6. Select **Download SDK**








Note: you may see “Request to Build” instead, so click on that.

7. After the package gets built, there will be a new download in the SDK Dashboard section. Click on the “Download SDK archive and documentation” button.

## MCUXpresso SDK Dashboard




Access, Download, and Share your requested SDK Builds.

My Recent SDKs (66 total archives) [SHOW ALL](#) [REMOVE ALL](#)

SDK Archive Details	Actions
 <p><a href="#">SDK_2.6.2_LPCXpresso55S69</a> </p> <p>Build Date: <b>2019-08-13</b>, Board: <b>LPCXpresso55S69</b>            OS: <b>Windows</b>, Toolchain: <b>All Toolchains</b>            Components: emWin, ISSDK, USB stack, sdmmc stack, Amazon-FreeRTOS, FatFS, FreeMASTER, wifi_qca, mcu-boot, mbedtls, AWS IoT, multicore, TF-M, CMSIS DSP Library, NTAG I2C            SDK Version: <b>2.6.2</b> (2019-06-26)            SDK Tag: REL_SDK_NIOBE4_2.6.2_RFP2.0_RC1</p>	    

## 3.2 Get the latest MCUXpresso IDE

8. Open [mcuxpresso.nxp.com](http://mcuxpresso.nxp.com)
9. Select **Software and Tools**; the click Learn More about MCUXpresso IDE

OVERVIEW	SOFTWARE AND TOOLS	DEVELOPER RESOURCES
 <p><b>MCUXpresso SDK</b></p> <p>An open-source software development kit (SDK) built specifically for your processor and evaluation board selections.</p> <p><a href="#">Learn More &gt;</a></p>	 <p><b>MCUXpresso IDE</b></p> <p>An easy-to-use integrated development environment (IDE) for creating, building, debugging, and optimizing your application.</p> <p><a href="#">Learn More &gt;</a></p>	 <p><b>MCUXpresso Config Tools</b></p> <p>A comprehensive suite of system configuration tools, including pins, clocks, SDK builder and more.</p> <p><a href="#">Learn More &gt;</a></p>

10. Download & Install MCUXpresso IDE 11.0.0

Here are some links for you:

- MCUXpresso IDE, version **MCUXpressoIDE\_11.0.0** from [here](#), and installed onto your host, assume Windows 10.
- MCUXpresso **SDK\_2.6.2\_LPCXpresso55S69.zip** from [here](#).
- A terminal emulator, such as TeraTerm or PuTTY. This lab describes the use of TeraTerm.

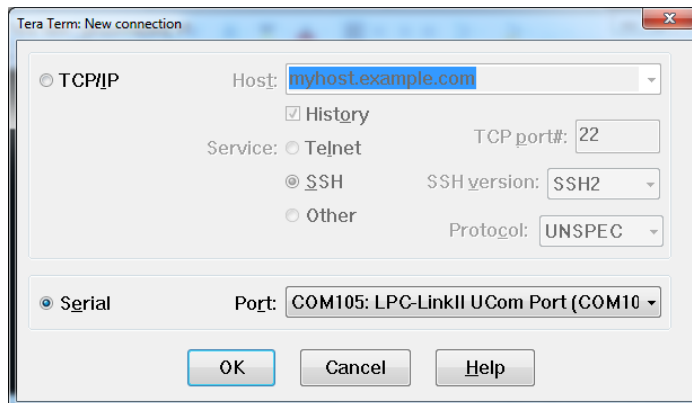
## 3.3 Tera Term Tutorial

Tera Term is a very popular open source terminal emulation application. This program can be used to display information sent from your NXP development platform's virtual serial port.

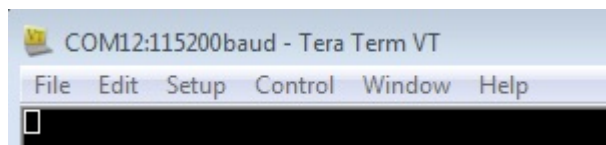
1. **Download** Tera Term from SourceForge. After the download, run the installer and then return to this webpage to continue.

[Download](#)

2. **Launch** Tera Term. The first time it launches, it will show you the following dialog. Select the serial option. Assuming your board is plugged in, there should be a COM port automatically populated in the list.



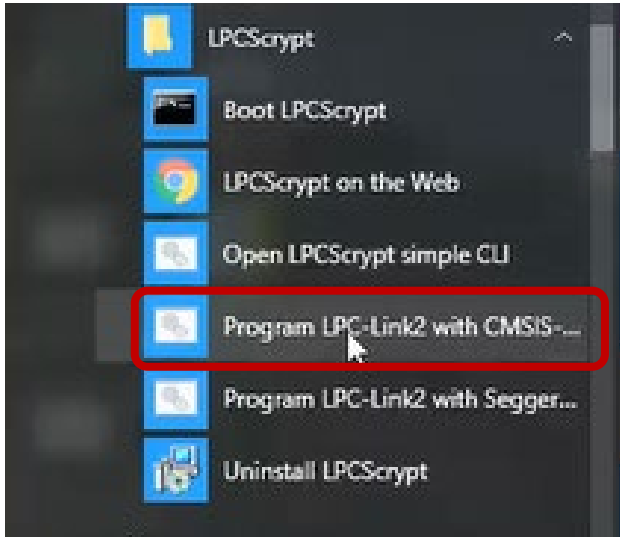
3. **Configure** the serial port settings (using the COM port number identified earlier) to 115200 baud rate, 8 data bits, no parity and 1 stop bit. To do this, go to Setup -> Serial Port and change the settings.
4. **Verify** that the connection is open. If connected, Tera Term will show something like below in its title bar.



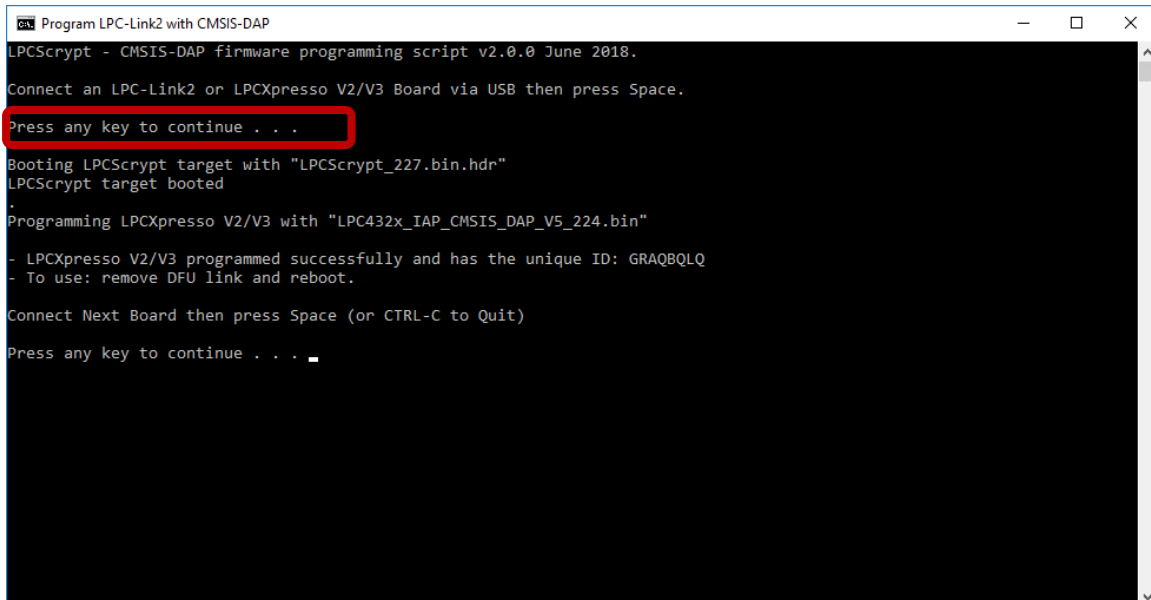
5. **You're ready to go**

### 3.4 Update Link2 Debugger Firmware

1. NXP periodically update the debug firmware. Some of the early shipments of LPC55S69-EVK boards contained an old version of the Link2 debugger firmware. It is convenient to configure the board so that the LPCSCRYPT tool can update the firmware and install communication drivers.
2. The Link2 microcontroller will be configured for Device Firmware Update (DFU) mode. This mode is enabled by fitting the DFU jumper on header J4 on the left-hand edge of the board adjacent to the Samtec FSTH 10-pin debug header.
3. Now power cycle the EVK by unplugging and re-plugging the USB cable. Note that the red LED **DS2** is illuminated and does not flash. The EVK is ready for Link2 firmware update.
4. Open the application LPCSCRYPT CMSIS update by clicking on the Windows Start button and scrolling to find LPCScrypt. Expand the tools and choose Program LPC-Link2 with CMSIS-Dap



5. A command window will open. Press any key to start the update. When the script stops you are done.



6. Remove the DFU jumper and power cycle the part by unplugging and re-plugging the USB cable.

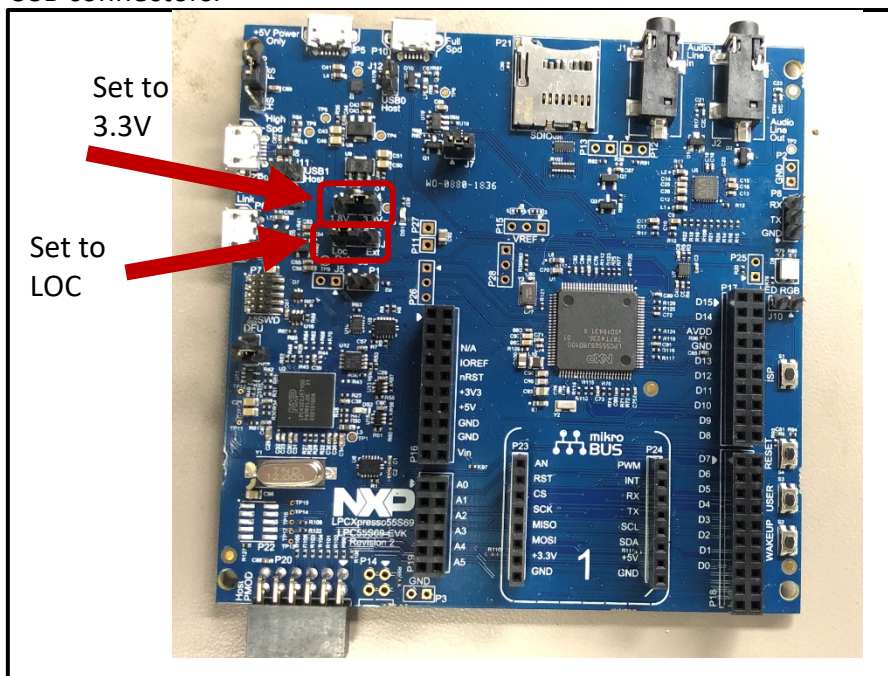
## 4 Hardware preparation

1. **Power Supply:** LPC55S69-EVK can be powered from one of four USB sources:
  - a. USB Debug connector **P6** (left-hand side of board) (this connection also generates 2.5 volts for the onboard Link2 Debug probe),

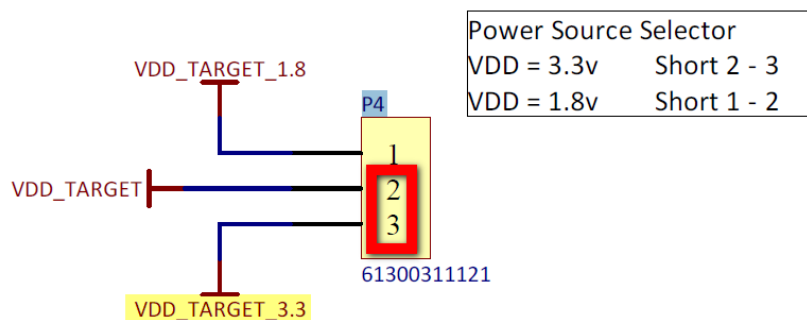
- b. USB High Speed connector **P9** (left hand side of the board) when the peripheral is in 'Device' mode,
- c. USB +5V power only USB connector **P5** (leftmost connector on the top edge of board), or
- d. USB Full Speed connector **P10** (adjacent to the SD socket on the on the top edge of board) when the peripheral is in 'Device' mode.

Each of these 4 USB- power sources are connected via protection diodes and more than one can be connected to your host PC at any time. Unless notified otherwise, we will always power the board via the USB Debug connector **P6**.

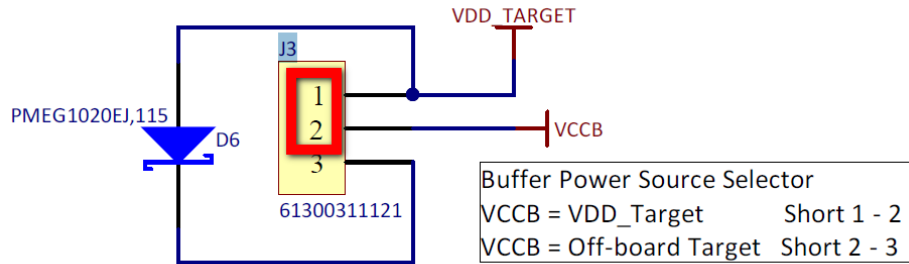
2. The EVK has a low jumper/header count. There are only 2 important jumpers to configure, and these should be the installed default when you receive the board. In the photograph of the board on the previous page you'll see the two important jumpers highlighted in red to the right of the USB connectors.



**P4** configures the operating voltage for the LPC5569 (either 3.3 volts or 1.8 volts). The jumper should be installed on the **[2-3]** setting, adjacent to the legend '**3.3V**' on the PCB. If you don't do this step you will have issues with the AWS WIFI lab.



**J3** configures the board for onboard local debugger, or external debugger (for example Segger JLink). The jumper should be installed on the **[1-2]** setting, adjacent to the legend '**Loc**' on the PCB.




3. Plug the USB cable into your PC.
4. Plug micro B end of USB cable into the **P6** Debug Link connector on the left-hand side of the EVK.  
**NOTE:** This will power the EVK and you will see red LED DS1 illuminate to signal that the EVK is powered.
5. Wait until windows on your PC install all required drivers

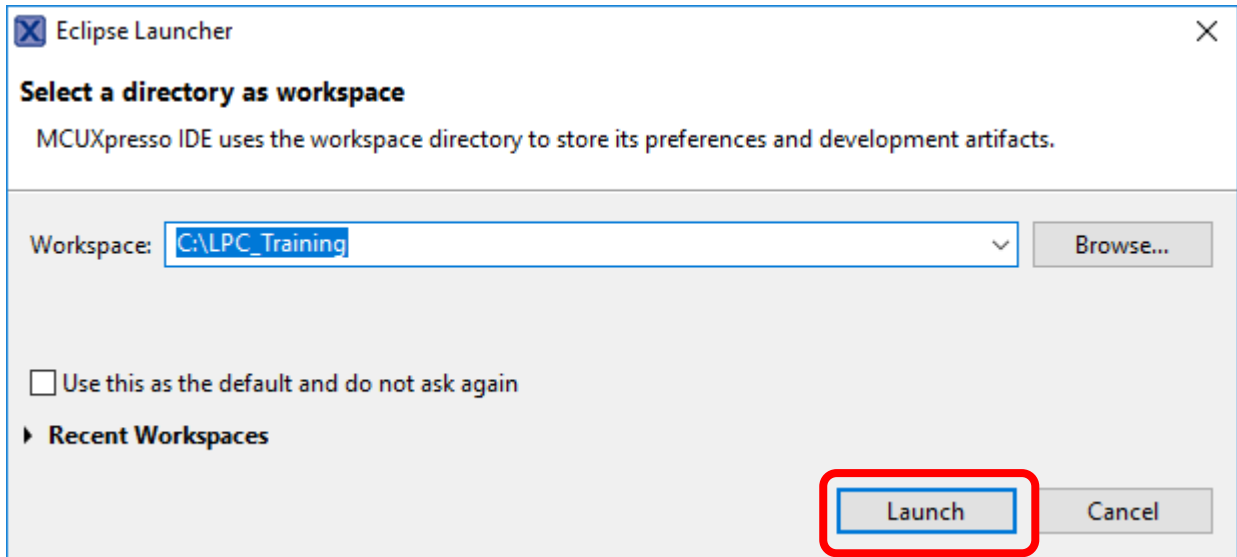
*After installation is complete you will see a red LED DS2 blinking at 1 Hz. Windows will have installed:*

- a. Link2 debugger, with CMSIS-DAP debug protocol pre-programmed;
- b. LPC-Link2 UCOM port for serial communication.

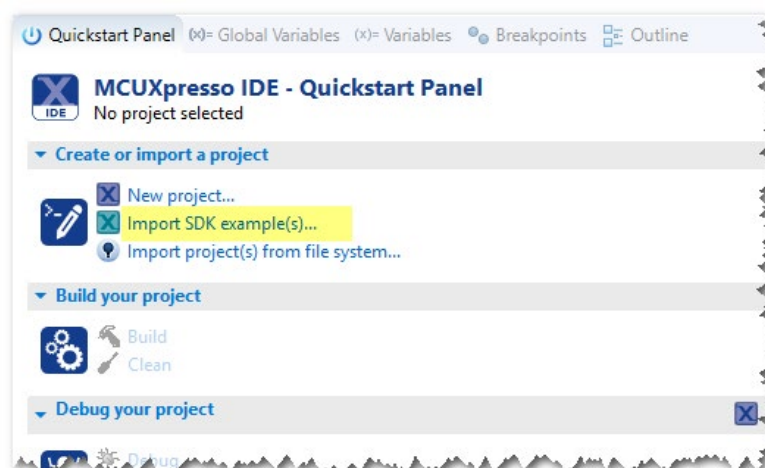
## 5 SDK Example: **hello\_world**

1. It is now time to start using both MCUXpresso IDE and MCUXpresso SDK. The SDK – Software Development Kit – is a collection of drivers, middleware and software examples. MCUXpresso SDK provides all of the necessary software to support all newly-released NXP microcontrollers.
2. MCUXpresso SDK for LPC55S69-EVK is available to download free of charge from [www.nxp.com](http://www.nxp.com). A demonstration will be provided in the seminar, and a video screen-grab of the process [Get Software for LPC55S69-EVK](#) is available on-line. You'll can also download a zip file of the SDK: [SDK\\_2.6.2\\_LPCXpresso55S69.zip](#).
3. We have assumed that [MCUXpresso IDE v11.0.0 \[Build 2516\] \[2019-06-05\]](#) is installed on your host PC. Start the IDE from the desktop, toolbar icons, or by running the program from the install location C:\NXP\MCUXpressoIDE\_11.0.0\_2516. 
4. The IDE will first prompt you for a workspace to store preferences, source code and development artifacts. It does not really matter where this workspace is located, nor the name that you provide. For simplicity, give the workspace the name **LPC\_Training**, and then click **[Launch]**. Ignore these instructions if you know what you are doing and would prefer a more convenient location.



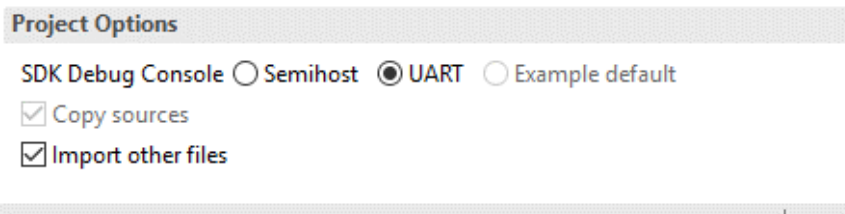
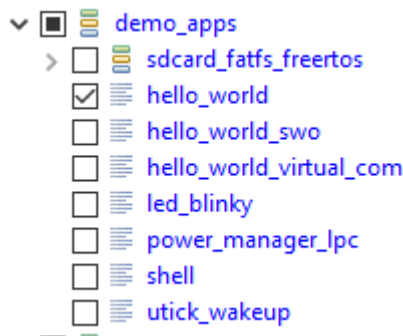


5. MCUXpresso opens with the **Develop** perspective and observe the **Quickstart Panel** view and the **Installed SDKs** view at the lower edge of the perspective.
6. Our first step is to import the SDK into MCUXpresso IDE with a 'drag and drop' operation. Locate the SDK for LPC55S69-EVK that you've been provided - [SDK\\_2.6.2\\_LPCXpresso55S69.zip](#) in Windows Explorer, then drag-and-drop it into the **Installed SDKs** view in MCUXpresso IDE.
7. Depending upon the host permission on your account, this process may, or may not work. If the drag and drop fails, you can manually copy the ZIP file into the following folder. Note that it is in your User account on your host PC.  
C:\Users\\mcuxpresso\SDKPackages
8. You should end up with the SDK installed into MCUXpresso IDE (you may need to restart MCUXpresso IDE). The SDK will be visible in the Installed SDKs view and we can now start to use the example software.
9. This is a hello\_world hands-on, and so we will import this SDK example into the workspace. From the **Quickstart Panel** view, select the wizard **Import SDK example(s)**...

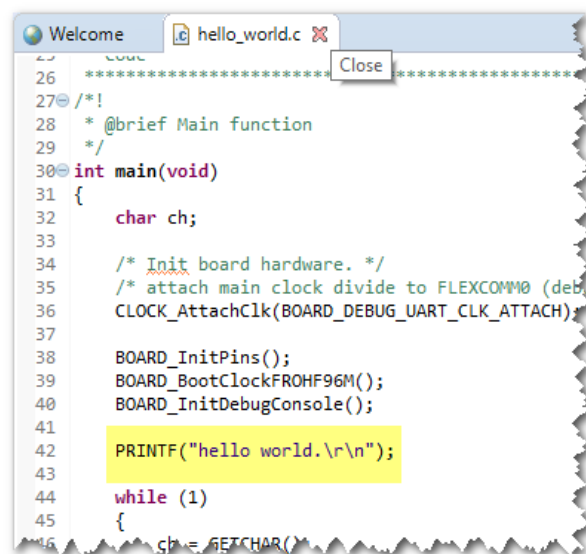
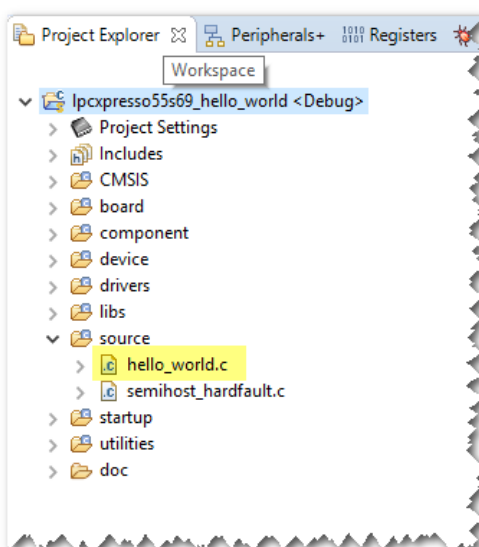


10. You can now work through the SDK Import Wizard screens, making the following selections:

- a. First, select the SDK that you wish to use by clicking on the **photograph** of the board : **lpcxpresso55s69**, then click **[Next >]**
- b. Then select the project(s) that you wish to import. We just need the **hello\_world** example under the **demo\_apps** group. Select the **hello\_world** project then in the top right select **UART** in the project options



- c. Close the Wizard with the **[Finish]** button. MCUXpresso IDE will import the SDK example project into our default workspace and the project can be browsed in the Project Explorer view.
11. Open the source module **hello\_world.c** for editing (double-click). Browse the very trivial code and note the macro **PRINTF** that enables us to redirect the print to either the console UART, or to a semi-hosted console via the debugger and into MCUXpresso IDE.



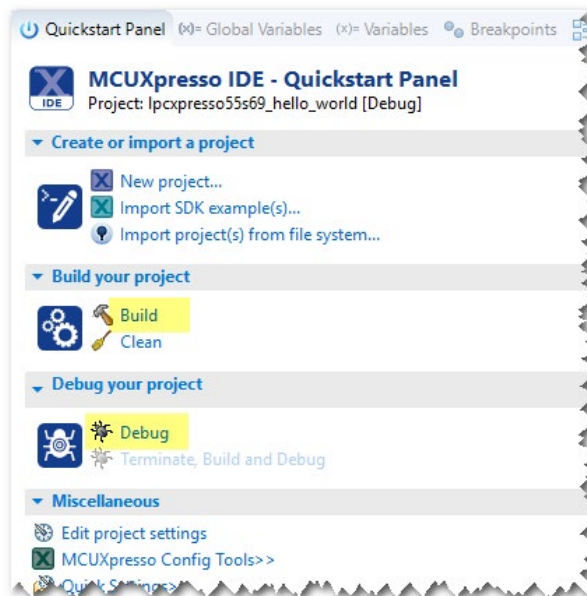
12. Modify the **PRINTF** string so that it prints a customized message. Use whatever string you want, for example **hello world from LPC55S69** then save the edit with hotkey **[Ctrl-S]**.

```

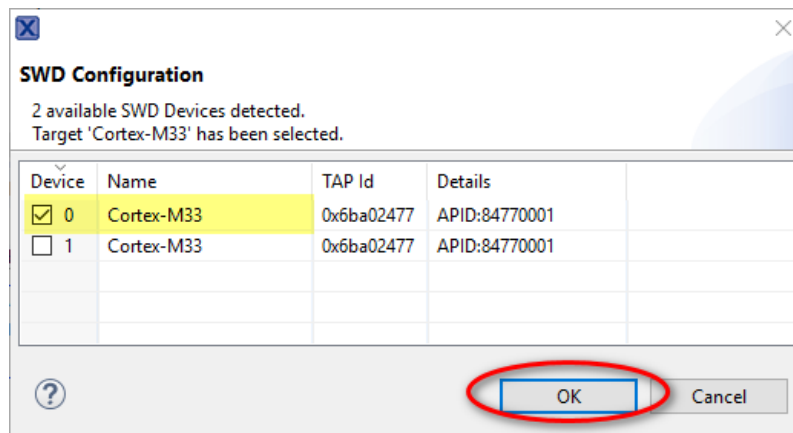
hello_world.c Welcome
29 lpcpresso55s69_hello_world/source/hello_world.c
30 int main(void)
31 {
32     char ch;
33
34     /* Init board hardware. */
35     /* attach main clock divide to FLEXCOMM0 (debug console) */
36     CLOCK_AttachClk(BOARD_DEBUG_UART_CLK_ATTACH);
37
38     BOARD_InitPins();
39     BOARD_BootClockFR0HF96M();
40     BOARD_InitDebugConsole();
41
42     PRINTF("hello world from LPC55S69.\r\n");
43
44     while (1)
45     {

```

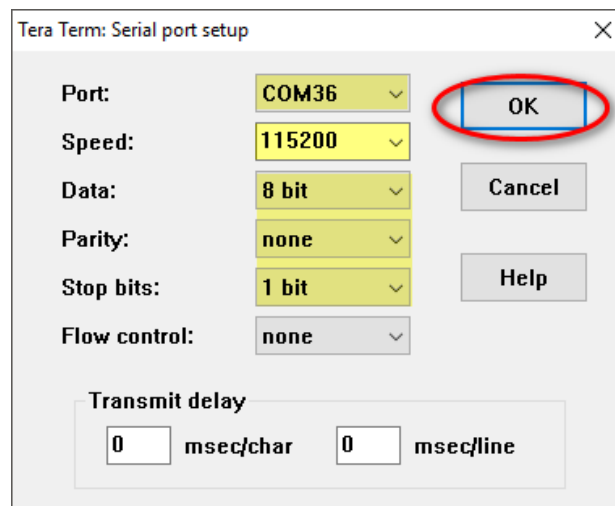
12. Connect the LPC55S69-EVK to host PC with a micro USB cable into **Debug Link P6**, if you have not already done so. Then select the project name **lpcpresso55s69\_hello\_world** in the Project Explorer view.
13. Use the Quickstart Panel view to **Build** the code example, and then the **Debug** option to start a debug session.



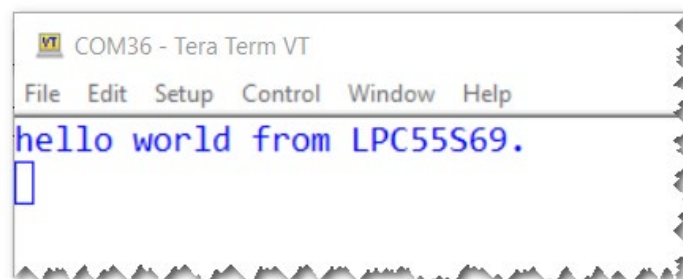
14. MCUXpresso IDE will undertake a probe discovery, and will report that it finds an **LPC\_LINK2 CMSIS-DAP V5.2** debug probe connected to the host. It will use this probe as the default, and so click [OK] to initialize the debug session.
15. The debugger session will find 2 SWD Devices (one for each core on the LPC55S69). This application is written for Device0 / Core0 and so please accept the default SWD Configuration by pressing [OK].



16. The debugger will load the image into internal flash on the LPC55S69, and then run the code up to `main()` where it will stop on the first line.
17. Run your terminal emulator (this lab uses TeraTerm) and open the COM port that was identified in Step 35. Speed / Format is 115,200 8N1:



18. Back in MCUXpresso IDE, run the application under the debugger using keyboard shortcut **F8 Resume**. The application will run, and you will see the `printf()` output string in TeraTerm:



19. For simplicity, resume the code with hotkey [F8] and observe that the PRINTF sends the string 'hello world' via the Tera Term Console
20. There is not much more that we can do, and so terminate the debug session with hotkey [Ctrl]-[F2] to return to the editing mode
21. Please stop here, Hands-On1 is now complete. Your tools are installed correctly.