AN12135

A71CH Quick start guide for OM3710A71CHARD and Kinetis

Rev. 1.0 — 09 July 2018
492310

Application note
COMPANY PUBLIC

Document information

Info	Content
Keywords	Security IC, A71CH, OM3710A71CHARD, FRDM-K64F, FRDM-K82F, FRDM-KW41Z
Abstract	This document provides a detailed guide for getting started with OM3710A71CHARD and the FRDM-K64F, FRDM-K82F or FRDM-KW41Z development platforms for K64, K82 and KW41Z MCUs respectively.

Revision history

Rev	Date	Description
1.0	20180709	First release

Contact information

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

1. Introduction

This document explains how to get started with the OM3710A71CHARD development kit and the FRDM-K64F, FRDM-K82F or FRDM-KW41Z development platforms for K64, K82 and KW41Z MCUs, respectively. This guide provides an overview of the hardware used, followed by detailed instructions for setting up the software development environment. Finally, it describes how to run A71CH application examples using FRDM-K64F, FRDM-K82F or FRDM-KW41Z, acting as the platform host MCU or acting as USB-to-I²C adapter.

2. A71CH Overview

The A71CH is a ready-to-use solution, enabling ease-of-use security for IoT device makers. It is a secure element capable of securely storing and provisioning credentials, securely connecting IoT devices to public or private clouds and performing cryptographic device authentication.

The A71CH solution provides basic security measures protecting the IC against many physical and logical attacks. It can be integrated with various host platforms and operating systems to secure a broad range of applications. In addition, it is complemented by a comprehensive product support package, offering easy design-in with plug & play host application code, easy-to-use development kits, documentation and IC samples for product evaluation.

3. System description

The A71CH evaluation setup presented in this document consists of an A71CH security IC connected to the FRDM-K64F, FRDM-K82F or FRDM-KW41Z development platforms through the OM3710A71CHARD Arduino compatible kit.

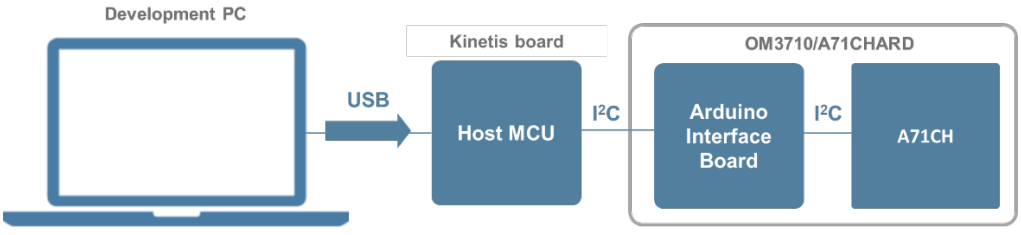


Fig 1. System architecture diagram

This getting-started guide is divided in three parts:

- **Hardware overview and setup**: It describes the FRDM-K64F, FRDM-K82F, FRDM-KW41Z development platforms and the A71CH Arduino compatible kit (OM3710/A71CHARD) as well as how to mount them together.
- **Software setup**: It describes how to configure the development environment and how to import the required software packages.

- **A71CH application examples execution**: It describes how to run the A71CH application examples contained in the A71CH Host software package.

Note: From now on, the term 'Kinetis board' will be used in this guide to avoid redundancy and to improve readability. 'Kinetis board' refers to the FRDM-K64F, FRDM-K82F and FRDM-KW41Z supported models.

4. Hardware overview

This setup uses a Kinetis board as a host MCU while the A71CH security IC acts as the secure element. The following two boards are needed:

1. The A71CH Arduino compatible development kit (OM3710/A71CAHRD).
2. The FRDM-K64F, FRDM-K82F or FRDM-KW41Z evaluation board.

4.1 A71CH Arduino compatible development kit (OM3710/A71CHARD)

The OM3710/A71CHARD is an Arduino development kit containing two items as well as:

1. An A71CH Mini PCB board (OM3710/A71CHPCB)
2. An Arduino interface board, allowing the user to connect the A71CH to any host featuring an Arduino compatible header (e.g., many LPC, Kinetis and i.MX boards in the industry).

4.1.1 A71CH Mini PCB board (OM3710/A71CHPCB)

The OM3710/A71CHPCB board is a small PCB containing the A71CH solution and a set of jumpers for the I²C or SPI host interface selection (Note that only the I²C driver is available. SPI support may be added in future revisions).

Fig 2 shows an image of the MiniPCB. It features two connectors that can be used depending on the communication interface employed. The figure shows the jumpers configuration that enables the use of the A71CH I²C interface.

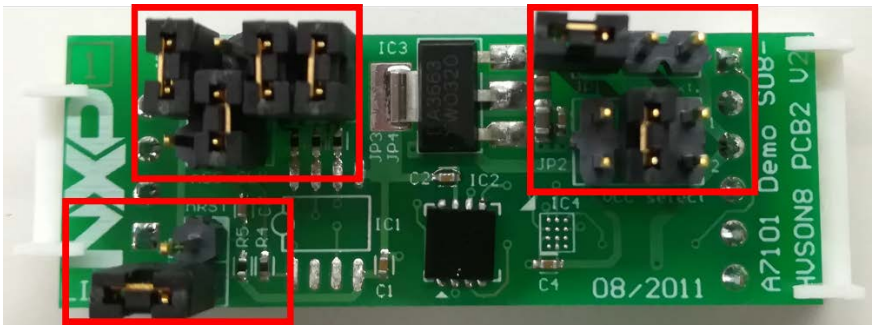


Fig 2. A71CH mini PCB OM3710/A71CHPCB

To enable the I²C communication protocol, it is necessary to configure JP5/6 according to Table 1. JP2 connects the A71CH to the on-board 3.3V voltage regulator on the

MiniPCB board. The jumpers JP3 and JP4 enable the I²C SDA/SCL pull-up resistors. JP7 can be used to connect the A71CH reset signal.

Table 1. Default OM3710/A71CHPCB Jumper settings

Jumper	Setting	Usage
JP1	Not set	External VCC connection
JP2	3-4	Connect A71CH to 3.3V regulator on MiniPCB
JP3	Set	Connect I²C SDA pull-up resistor
JP4	Set	Connect I²C SCL pull-up resistor
JP5	1-2	Use I²C address 0x92/0x93
	2-3 (Default)	Use I²C address 0x90/0x91
JP6	1-2	Activate I²C interface
JP7	Not set (Default)	A71CH operates
	Set	A71CH IC reset

The board schematic and layout are shown in Fig 3 and Fig 4.

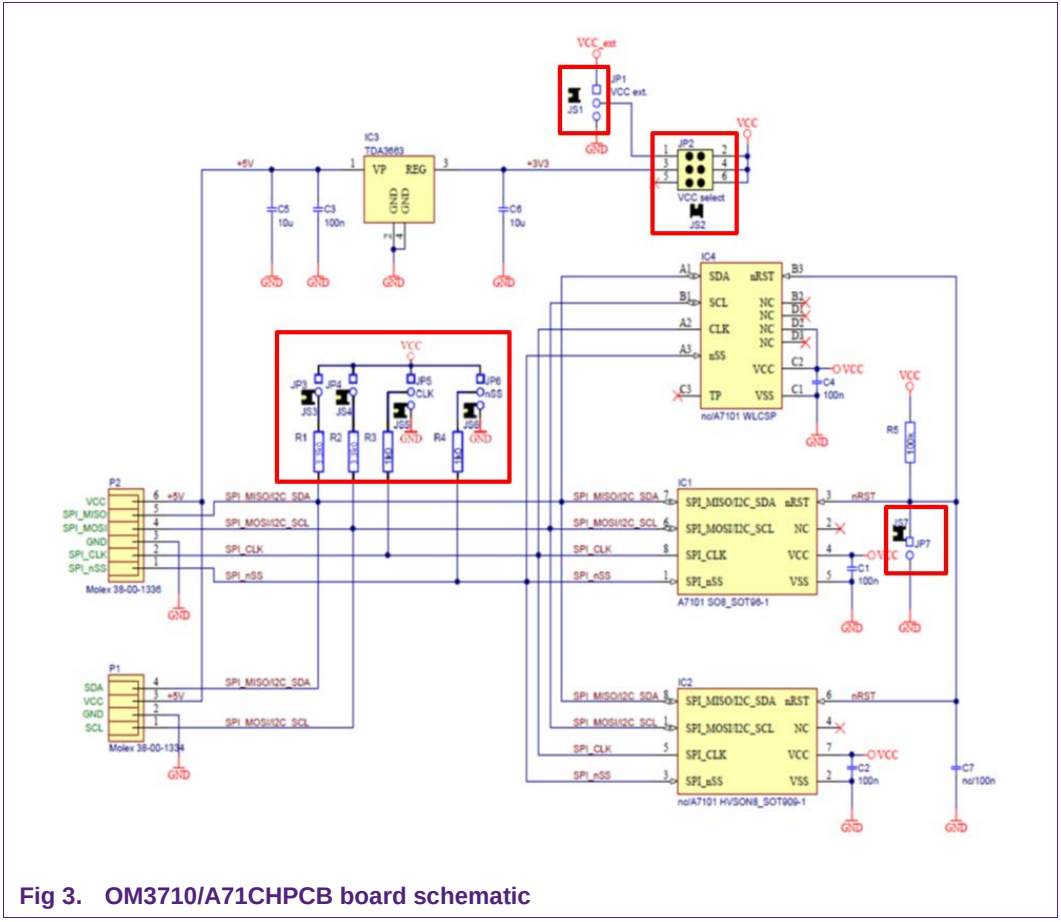


Fig 3. OM3710/A71CHPCB board schematic

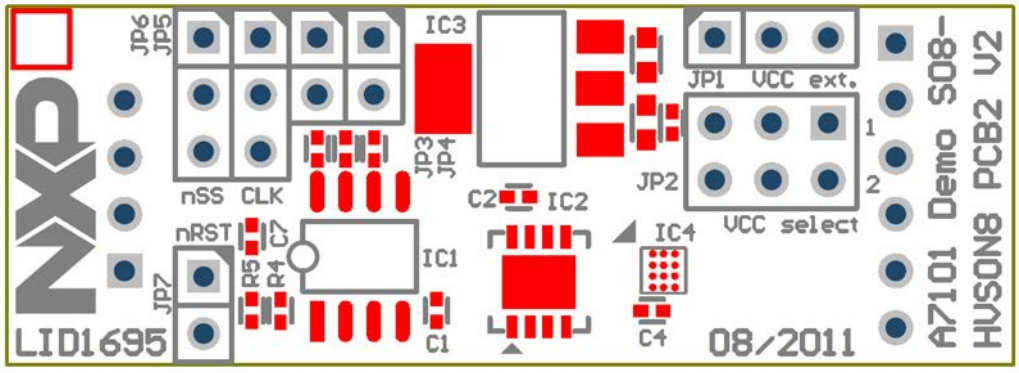
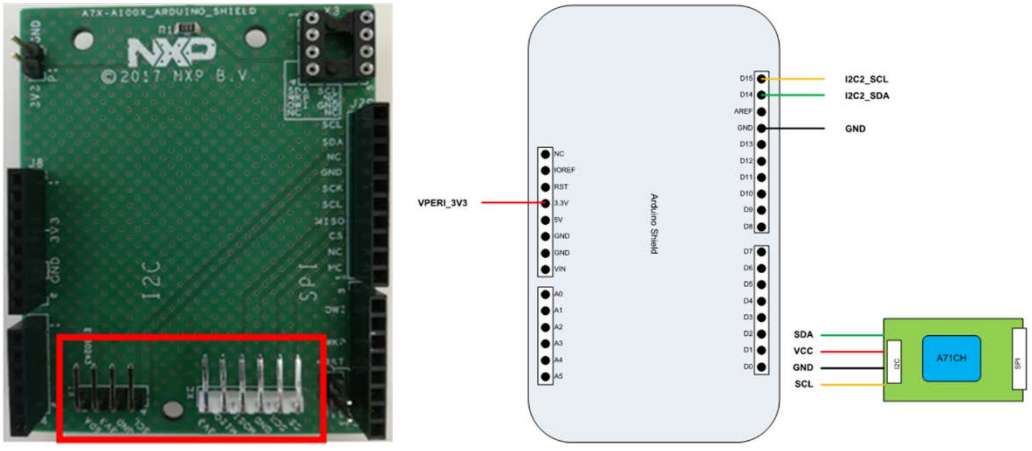


Fig 4. OM3710/A71CHPCB board silkscreen

4.1.2 Arduino interface board

The Arduino header board permits the user to interface the A71CH OM3710/A71CHPCB with the Kinetis board. Fig 5 shows the board pinout.



(1) Connection from the perspective of a matching Arduino Shield (top view)

Fig 5. A71CHARD Arduino header

In addition, the A71CHARD provides dedicated male connectors to mount the A71CHPCB via I^2C or SPI without any hardware modification.

4.2 Freedom development platforms for Kinetis

The section details the Freedom development platforms for Kinetis supported by the A71CH product support package.

4.2.1 FRDM-K64F

The Kinetis FRDM-K64F [FRDM_K64F] development platform is a simple, yet sophisticated design, featuring a Kinetis K64 series microcontroller, built on the ARM® Cortex®-M4 core. The FRDM-K64F can be used to evaluate the K64, K63, and K24 Kinetis K series devices. It features the MK64FN1M0VLL12 MCU, which boasts the maximum operation frequency of 120 MHz, 1 MB of flash, 256 KB RAM, a full-speed USB controller, Ethernet controller, secure digital host controller, and analog and digital peripherals.

The FRDM-K64F hardware is form-factor compatible with the Arduino R3 pin layout, providing a broad range of expansion board options. The onboard interface includes a six-axis digital accelerometer & magnetometer, RGB LED, SDHC, add-on Bluetooth module, add-on RF module, Ethernet and OpenSDAv2, the NXP open-source hardware embedded serial and debug adapter running an open-source bootloader.

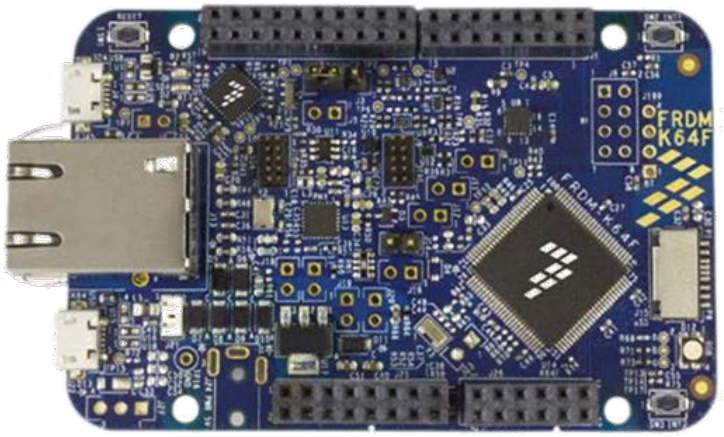


Fig 6. FRDM-K64F Freedom development platform for Kinetis K64, K63 and K24 MCUs

4.2.2 FRDM-K82F

The Freescale Freedom K82 hardware [FRDM-K82F] is a simple yet sophisticated design featuring a Kinetis K series microcontroller built on the ARM® Cortex®-M4 core which features a floating-point unit (FPU).

The FRDM-K82F can be used to evaluate the K80, K81, and K82 Kinetis K series devices. The FRDMK82F board features the K82FN256VLL15 MCU, which boasts a maximum operation frequency of 150 MHz, 256 KB of flash, a 256 KB RAM, a full-speed USB controller with available crystal-less operation, and analog and digital peripherals.

Fig 7. FRDM-K82F Freedom development platform for Kinetis K80, K813 and K82 MCUs

4.2.3 FRDM-KW41Z

The FRDM-KW41Z Freedom development board [FRDM_KW41Z] is a small, low-power, and cost-effective evaluation and development board for application prototyping and demonstration of the KW41Z/31Z/21Z (KW41Z) family of devices. The KW41Z integrates a radio transceiver operating in the 2.36 GHz to 2.48 GHz range (supporting a range of FSK/GFSK and O-QPSK modulations) and an ARM Cortex-M0+ MCU into a single package.

The FRDM-KW41Z development board consists of the KW41Z device with a 32 MHz reference oscillator crystal, RF circuitry (including antenna), 4-Mbit external serial flash, and supporting circuitry in the popular Freedom board form-factor. The board is a standalone PCB and supports application development with NXP's Bluetooth Low Energy, Generic FSK, and IEEE Std. 802.15.4 protocol stacks including Thread.

Fig 8. FRDM-KW41Z Freedom development platform for Kinetis KW41Z/31Z/21Z MCUs

5. Hardware setup

The hardware setup consists of mounting the different boards together. Two simple steps are required. First, plug the A71CH Mini PCB board (OM3710/A71CHPCB) to the I2C adaptor of the Arduino interface board.

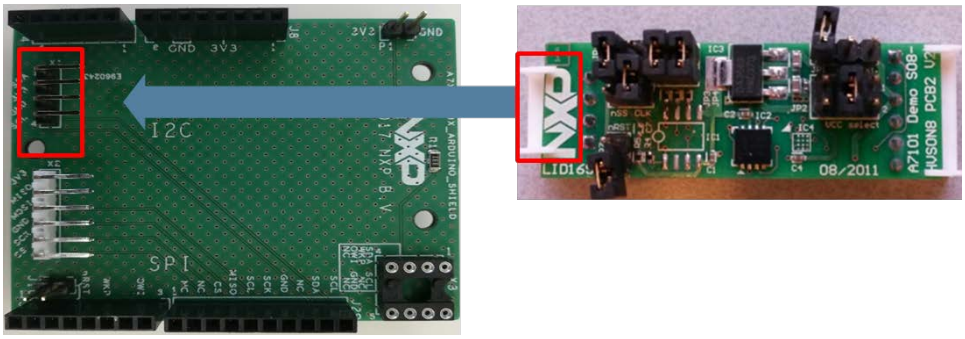


Fig 9. A71CH Mini PCB board (OM3710/A71CHPCB) mounting on the I2C adaptor of the Arduino interface board

Second, plug the A71CH into the Kinetis board using the Arduino adaptors. Please note the Arduino shield board comes with male connectors below. If the Kinetis board does not come with the Arduino headers assembled by default, the user can easily solder them.

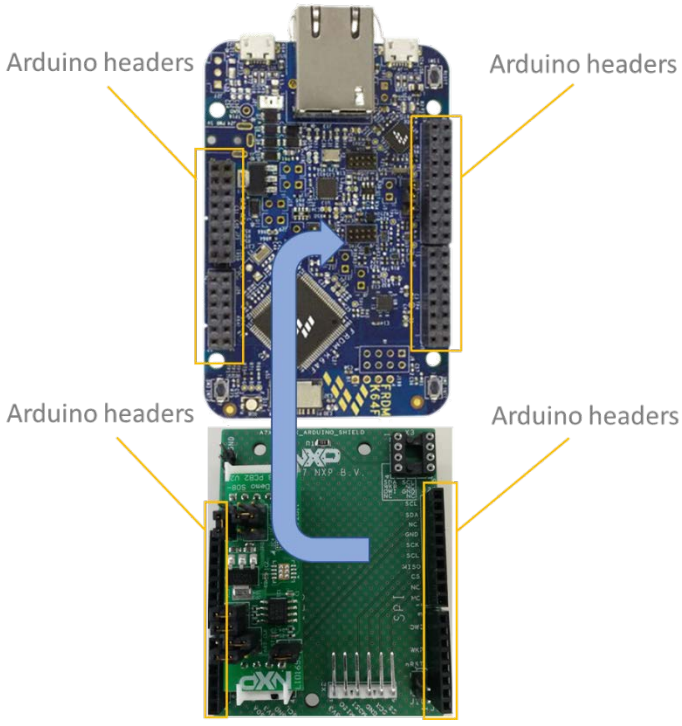


Fig 10. A71CH Arduino kit mounted on Kinetis FRDM-K64F board

Then, the A71CH security IC is connected to the Kinetis board through the Arduino interface board (Fig 11).

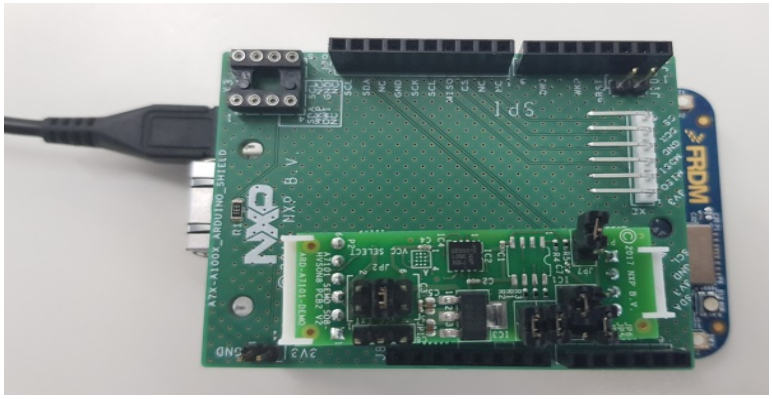


Fig 11. Arduino interface board connected to the Kinetis board (FRDM-K64F)

As can be observed, there are two USB connectors in the Kinetis boards FRDM-K64F and FRDM-K82F (Fig 12). The USB connector highlighted in red corresponds to OpenSDA serial port. This port will be used by the development PC to flash the A71CH examples into the Kinetis MCU. On the other hand, the USB connector highlighted in

yellow corresponds to the virtual COM connector port. In case of the FRDM-KW41Z board, the virtual COM port is not available.

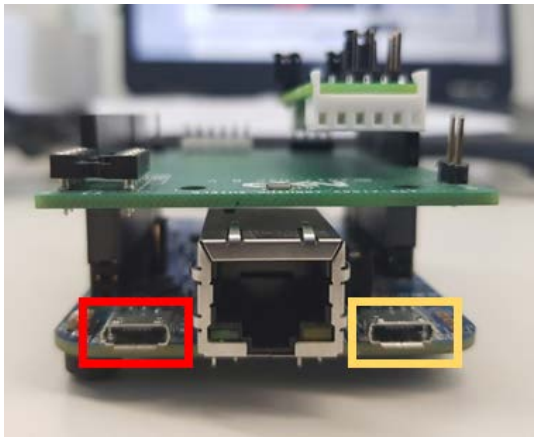


Fig 12. Red USB indicates OpenSDA serial port. Yellow USB indicates virtual COM connector port (FRDM-K64F)

6. Software setup

This section details the required steps to complete the software setup for A71CH security IC and FRDM-K64F, FRDM-K82F or FRDM-KW41Z Freedom development platforms.

Note: This section details the software setup for A71CH security IC and FRDM-K64F, FRDM-K82F or FRDM-KW41Z Freedom development platforms based on A71CH HostLib v1.4.0. If you are using a different A71CH HostLib version, the screenshots or project names indicated in this section may differ.

6.1 MCUXpresso IDE installation

MCUXpresso IDE is a fully featured software development environment for NXP's ARM-based MCUs, and includes all the tools necessary to develop high-quality embedded software applications in a timely and cost-effective fashion.

MCUXpresso IDE is based on the Eclipse IDE and includes the industry standard ARM GNU toolchain. It brings developers an easy-to-use and unlimited code size development environment for NXP MCUs based on Cortex-M cores (LPC, Kinetis and i.MX RT). The IDE combines the best of the widely popular LPCXpresso and Kinetis Design Studio IDEs, providing a common platform for all NXP Cortex-M microcontrollers.

MCUXpresso IDE is a free toolchain providing developers with no restrictions on code or debug sizes. It provides an intuitive and powerful interface with profiling, power measurement on supported boards, GNU tool integration and library, multicore capable debugger, trace functionality and more. MCUXpresso IDE debug connections support Freedom, Tower, EVK, LPCXpresso and custom development boards with industry

leading open-source and commercial debug probes including LPC-Link2, P&E and SEGGER.

The fully featured debugger supports both SWD and JTAG debugging, and features direct download to on-chip and external flash memory

The installation file of MCUXpresso can be found in [MCUXPRESSO_IDE]. The setup wizard will guide the user through the process of installing MCUXpresso correctly. Since MCUXpresso requires extra drivers during the installation, check all the items on the list to allow the drivers to be installed. Make sure the checkbox for installing the NXP debug drivers is activated (Fig 13).

Note: Please, install MCUXpresso IDE version 10.2.0 or higher

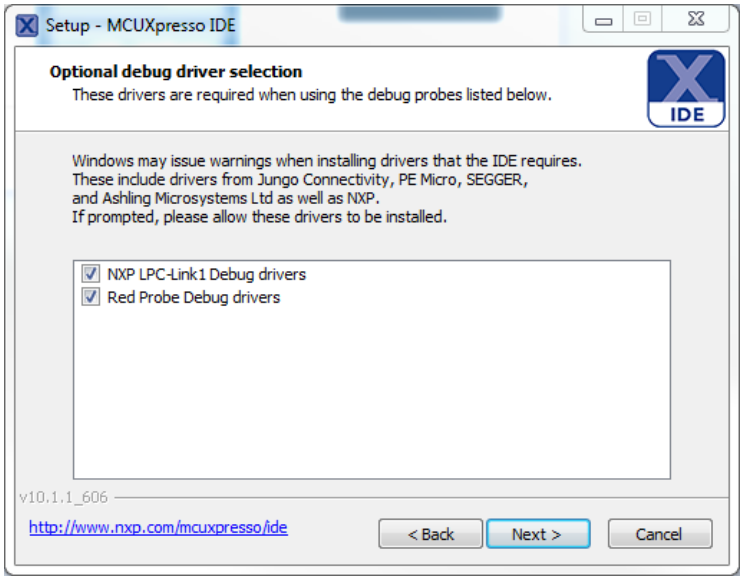


Fig 13. MCUXpresso install wizard

6.2 OpenSDA configuration

OpenSDA is a serial and debug adapter built into the Kinetis board. It provides a bridge between the development PC and the Kinetis MCU, which can be used for debugging, flash programming and serial communication all over USB.

Note: This section explains how to install the correct OpenSDA bootloader firmware version to the Kinetis FRDM board. This needs to be done for debugging, flash programming, and serial communication over a single USB connection between a host and an embedded target processor. If this section is not followed carefully, it is possible the examples will not be executed.

To configure OpenSDA into the Kinetis FRDM board, an OpenSDA bootloader (.bin file) should be downloaded from OpenSDA website [OPENSDA_FIRMWARE]. Scroll down the page to section '*Compatible Evaluation Boards'* and search for the target Kinetis FRDM board. In this case, Fig 14 depicts the OpenSDA bootloader version defined for the Kinetis FRDM K64F: version 2.0.

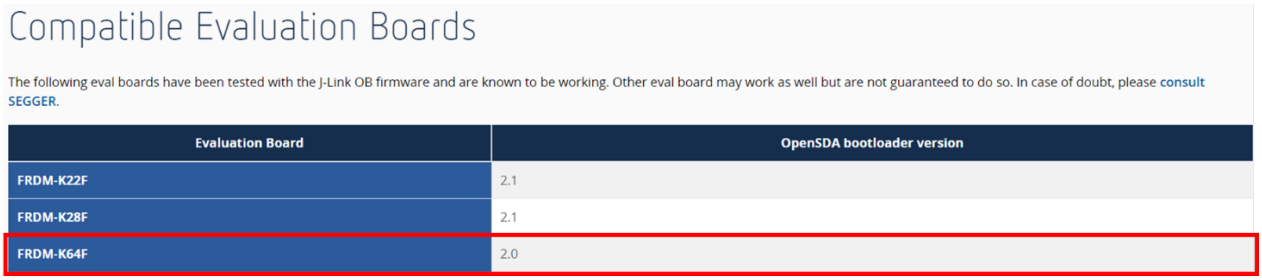


Fig 14. OpenSDA bootloader version for the Kinetis FRDM-K64F

Once the OpenSDA bootloader version is identified, click in the '*Downloads'*, scroll down until '*J-Link OpenSDA – Generic Firmwares'* appears and download the desired version.

Fig 15 illustrates the process; in this case *OpenSDA V2 Bootloader* has been selected, according to the compatible evaluation boards table previously mentioned.

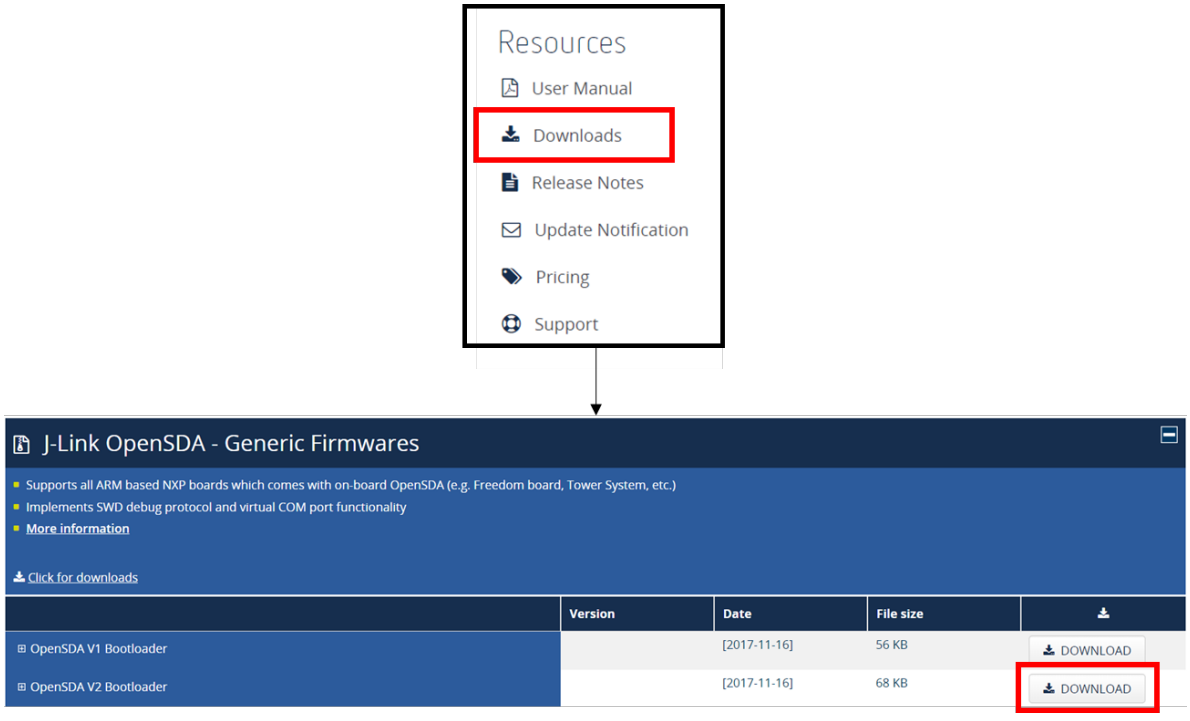


Fig 15. Desired firmware for the Kinetis FRDM-K64F

To write the downloaded firmware into the Kinetis FRDM board, the bootloader mode should be enabled. For this, press '*Reset'* button and, while holding down the button, connect a USB cable to the Kinetis board (e.g., FRDM K64, Fig 16).

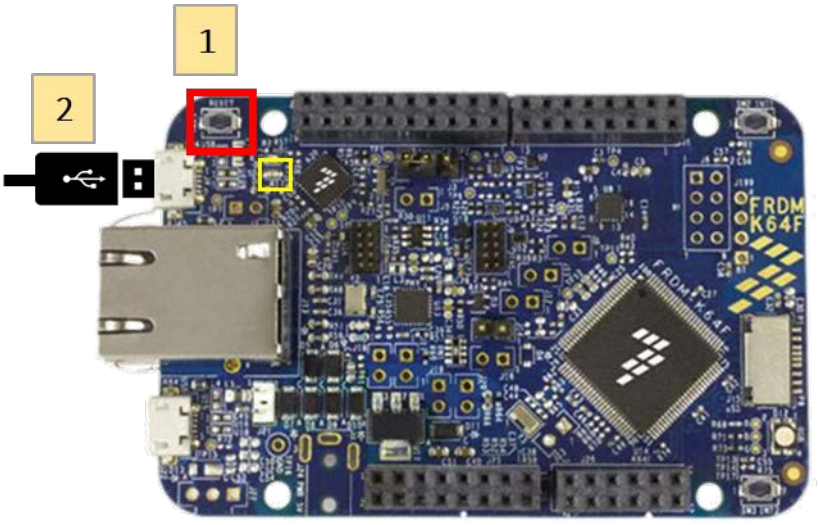


Fig 16. Enabling bootloader mode

After connecting the USB cable to the Kinetis board, the green led located inside the yellow square will start blinking and the development PC will show a new drive called '*BOOTLOADER'*.

Drag the downloaded firmware directly into the drive (Fig 17). Once the file is copied inside the '*BOOTLOADER'* drive, unplug the Kinetis board and plug it again. The green led remains still, thus indicating that the OpenSDA bootloader firmware has been configured correctly.

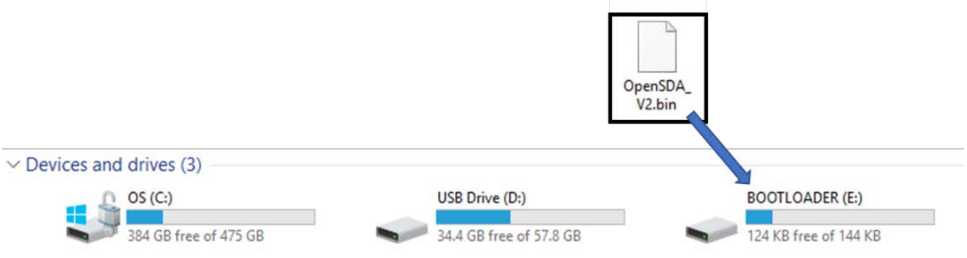


Fig 17. Copying the firmware into the Kinetis board in 'BOOTLOADER' mode

6.3 Kinetis SDK package for A71CH

To generate and download your customized SDK for your Kinetis FRDM board, you can enter the MCUXpesso SDKBuilder website [SDKBuilder] and follow these steps:

1. Select your Kinetis FRDM board and click on '*Build MCUXpresso SDK*'; in this case the selected board is the *FRDM-K64F* (Fig 18).

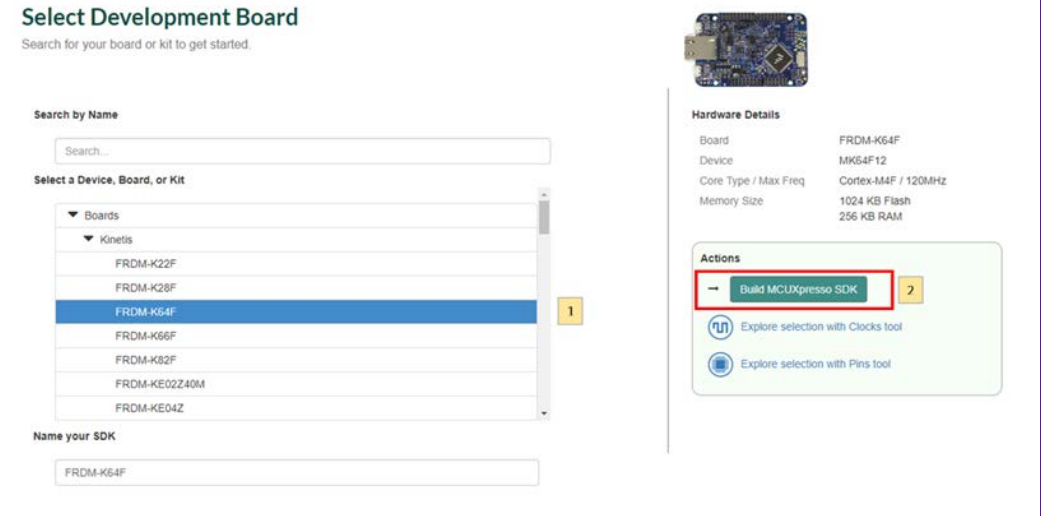


Fig 18. Kinetis board selection

In the next screen, select the software components (Fig 19):

2. Select '*Add software component'*.
3. Select the middleware; choose options '*FatFS*', '*USB stack*', '*lwIP*', '*mbedtls*', '*Secure Element*', '*Amazon-Freertos Kernel*' and '*AWS IoT*'.
4. Click on '*Save changes*'.
5. Finally, click on '*Download SDK*'.

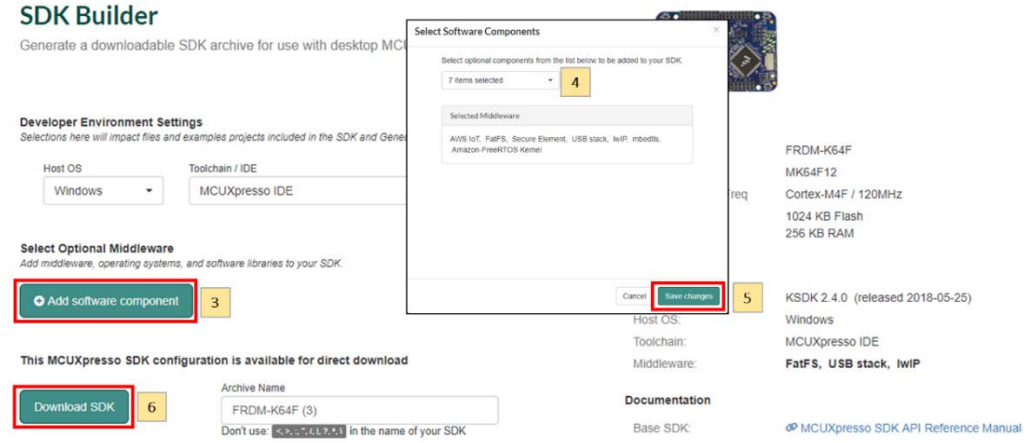


Fig 19. Kinetis SDK for A71CH configuration

The downloaded SDK should be imported in MCUXpresso IDE. To import the SDK into MCUXpresso IDE, drag and drop the SDK file inside the red square (‘*Installed SDKs*’) and then click ‘*OK*’ to confirm the operation (Fig 20).

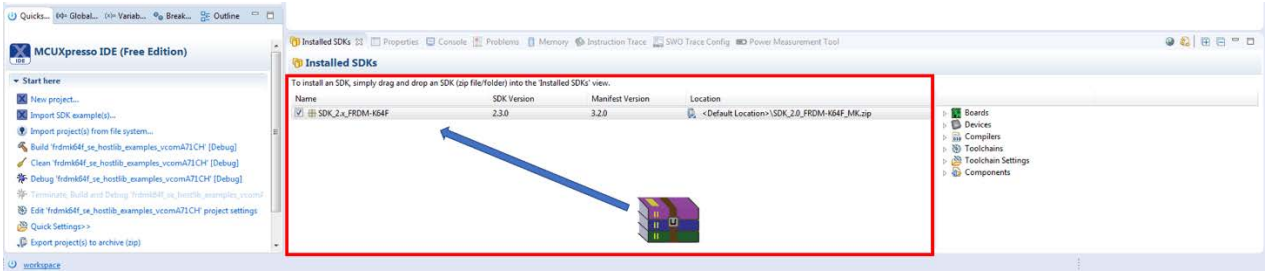


Fig 20. Installing the downloaded Kinetis SDK into MCUXpresso

6.4 Importing A71CH example projects in MCUXpresso IDE

There are two possible ways to import A71CH project examples in MCUXpresso IDE, depending if we are using the MCUXpresso project files bundled with the A71CH Host Software package installer or if the installed SDK package already contains the A71CH middleware:

- Importing the A71CH example projects from the installed SDK.
- Importing the A71CH example projects from local drive (included in the A71CH Host Library installer).

6.4.1 Importing the A71CH example projects from the installed SDK

The first option is to import the A71CH example projects from the installed SDK:

1. Select ‘*Import SDK example(s)’…* to import available example projects to the workspace.

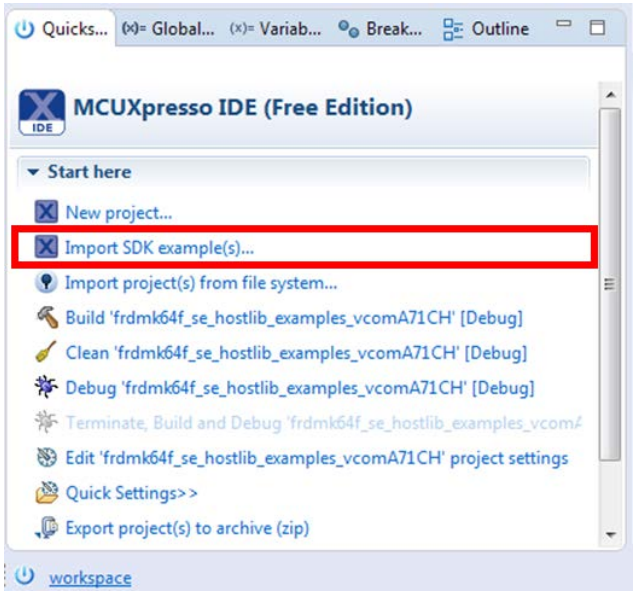


Fig 21. Importing project from SDK

An SDK Wizard window will pop-up:

2. Select '*frdmk64f*' from Available boards and then click the next button (Fig 22).

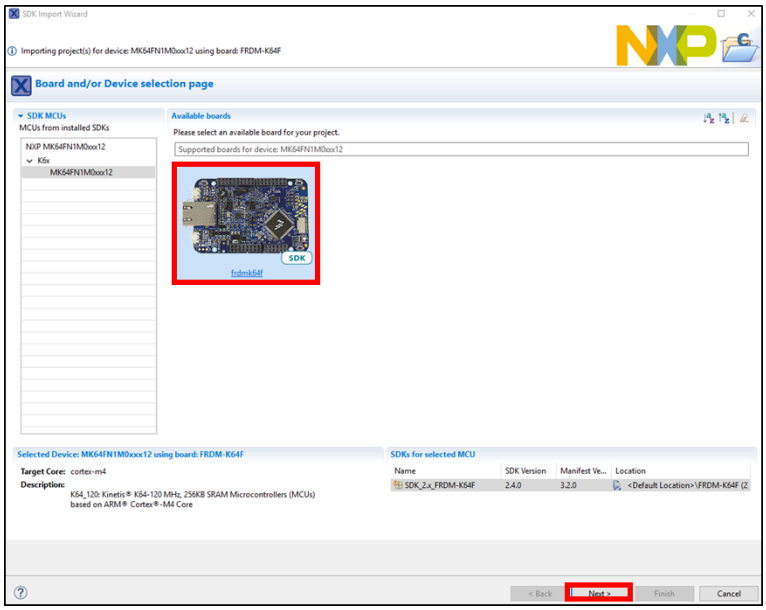


Fig 22. Importing project from SDK. Board selection page.

3. A list with different elements included in the SDK will appear; click on '*se-hostlib-examples*' and '*Finish'* (Fig 23).

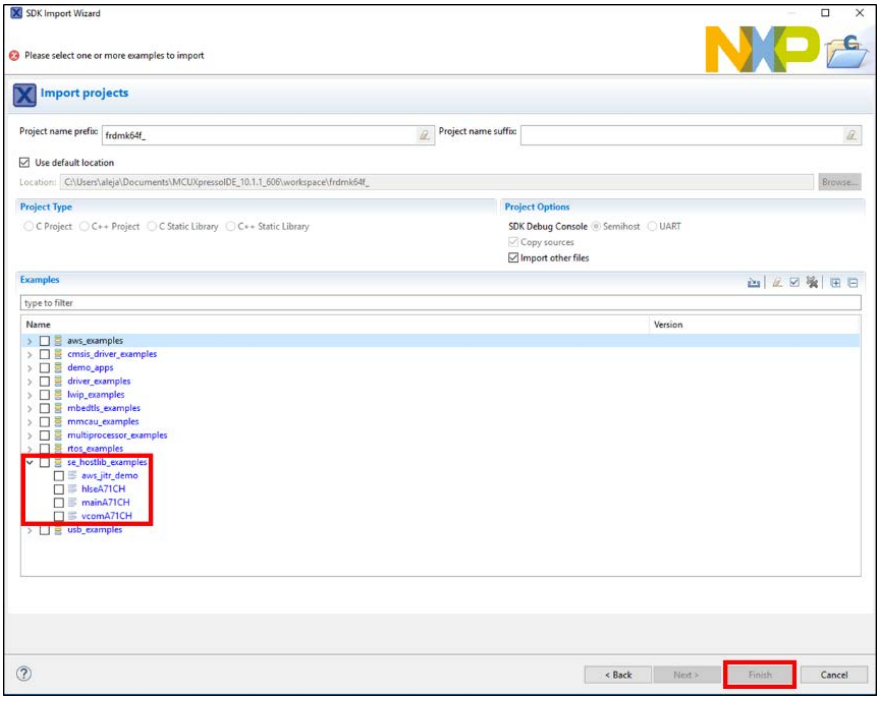


Fig 23. Importing project from SDK. Select A71CH example projects.

The imported examples will appear in the workspace window

6.4.2 Importing A71CH example projects from local drive (bundled with installer)

Alternatively, the project files bundled with the A71CH Host Software package installer can be used. The A71CH Host Software Package can be downloaded from [A71CH_HOST_SW].

For instance, these are in '*A71CH_v<libversion>/frdmk64f_projects'* in the case of the FRDM K64F board. The content of this folder is illustrated in Fig 24. As can be seen, there are three example projects:

- A71CH Host API usage project: demonstrates the usage of various functionalities of the A71CH in combination with mbedTLS cryptographic library.
- VCOM project: allows the Kinetis board to be used as a bridge between the PC and the A71CH and enables the execution of the A71CH Configure tool and other utilities from the PC.
- AWS JITR demo project: performs the connection of the Kinetis board to Amazon Web Service (AWS) cloud, preparing the board for the Just-In-Time Registration (JITR) procedure.

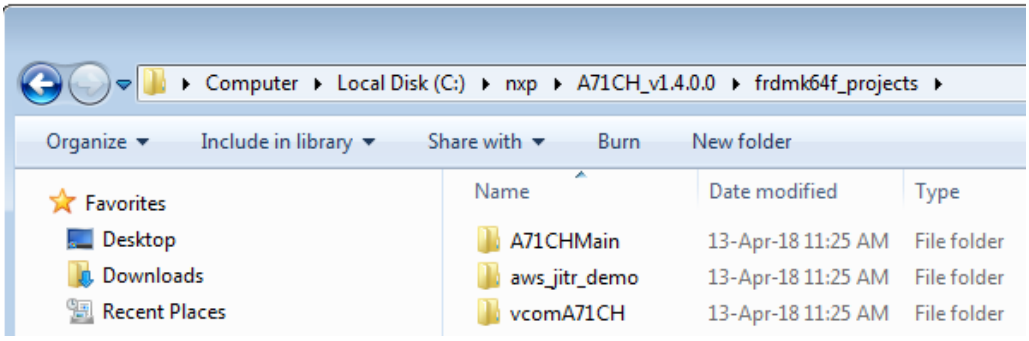


Fig 24. Content of 'frdmk64f_projects' folder

To import a project from file system, click on '*Import project(s) from file system…*' in the '*Quick start Panel'* located in the bottom left (Fig 25).

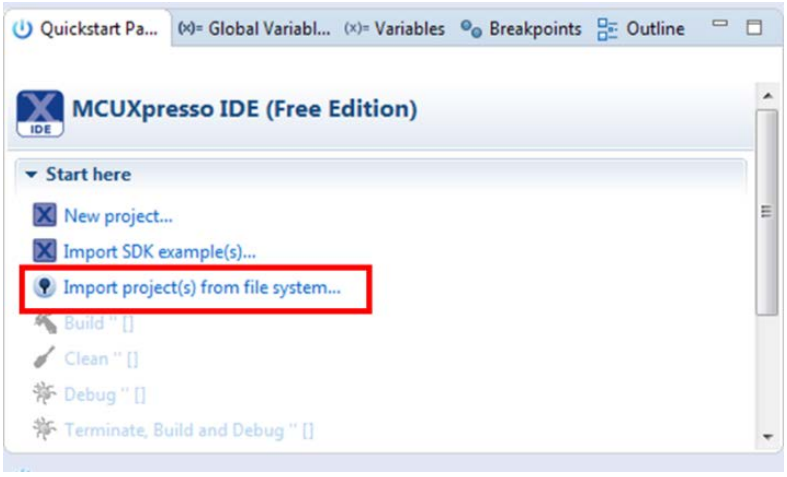


Fig 25. Import project from file system

After clicking the import option, a new pop-up will open. In the '*Project directory (unpacked)*' field, browse and point to the correct project directory (Fig 26). Then, click on '*Next*'.

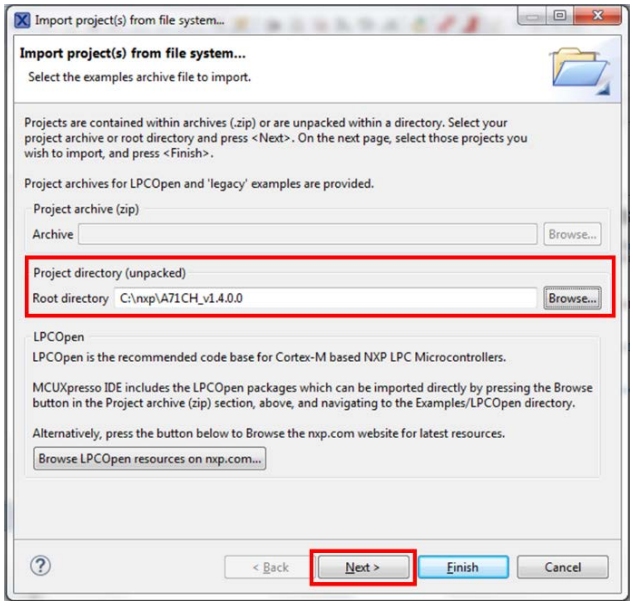


Fig 26. Project directory to import

Finally, select all the available A71CH example projects and then click on '*Finish*'.

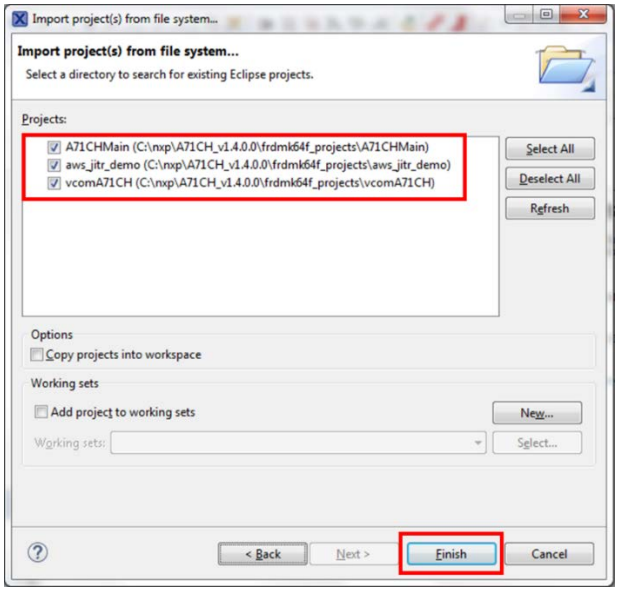


Fig 27. Project directory to import

6.5 Microsoft Visual Studio IDE installation

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as web sites, web apps, web services and mobile apps.

The download of this IDE is not mandatory. It is used to compile the tools and example projects, but they are already supplied as binaries as well.

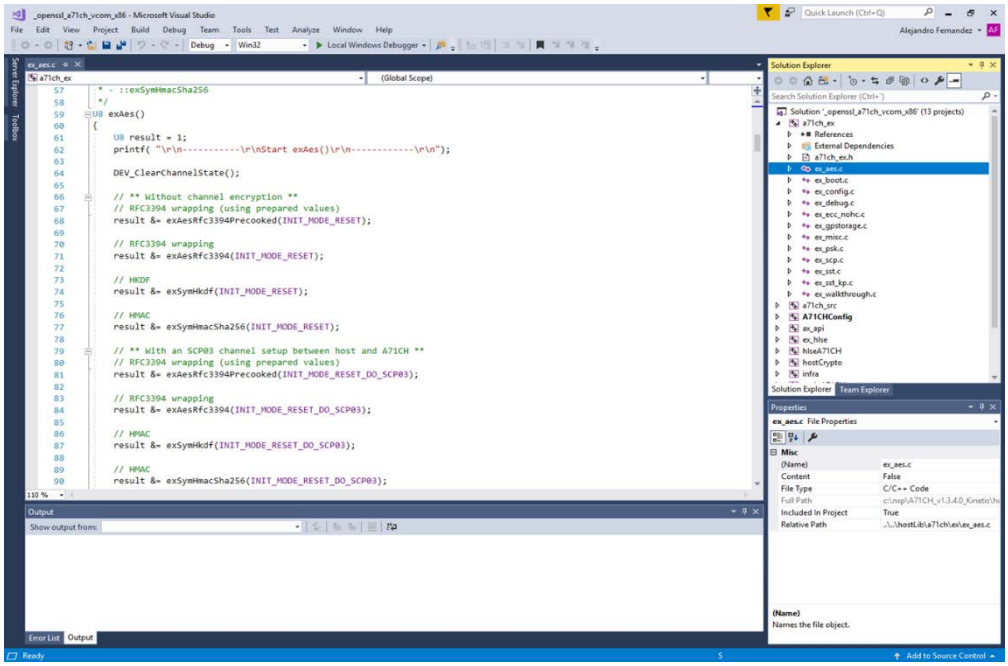


Fig 28. Microsoft Visual Studio 2017

The available A71CH projects for Microsoft Visual Studio support Microsoft Visual Studio 2010, 2012, 2015 and 2017 versions. The Microsoft Visual Studio IDE installation process can be found as part of the Microsoft online documentation.

6.6 Terminal setup

A terminal application must be executed from the development PC to interact with the Kinetis board. Any terminal supporting a serial port interface can be used.

In this document, Tera Term is used and can be downloaded from [TERA_TERM]. The setup wizard will guide the user through the installation. The standard installation can be chosen for this purpose. Once it is finished, Tera Term can be started.

7. A71CH application examples execution

The A71CH Host software package [A71CH_HOST_SW] includes three different application examples:

- **A71CH Host API usage example**: A sample project including a set of source code examples oriented to show the A71CH Host Library usage.
- **A71CH Configure Tool**: A command line tool that supports the injection of credentials into the A71CH.
- **mbedTLS examples**: A set of examples that demonstrate the integration of mbedTLS software stack with A71CH.

Note that, FRDM-KW41Z is not compatible with the A71CH Configure tool and the mbedTLS examples because it cannot be configured as a VCOM port.

Table 2. Kinetis development boards and A71CH supported examples

	Host API usage	A71CH Configure Tool	VCOM Mode	mbedTLS examples
FRDM-K64F	Yes	Yes	Yes	Yes
FRDM-K82F	Yes	Yes	Yes	Yes
FRDM-KW41Z	Yes	No	No	No

Note: This section details the A71CH application examples execution based on A71CH HostLib v1.4.0. If you are using a different A71CH HostLib version, the screenshots or project names indicated in this section may differ.

7.1 Running A71CH Host API usage examples

Fig 29 shows the setup that will be used to run the A71CH Host API usage example. The A71CH security IC is connected to the Kinetis board through the Arduino interface board and the user will employ MCUXpresso IDE (installed in the Windows PC) to program the Kinetis MCU, so that it executes the A71CH Host API usage example application.

The execution output can be seen either in the MCUXpresso IDE console or by using Tera Term. Both cases are explained step by step in this section.

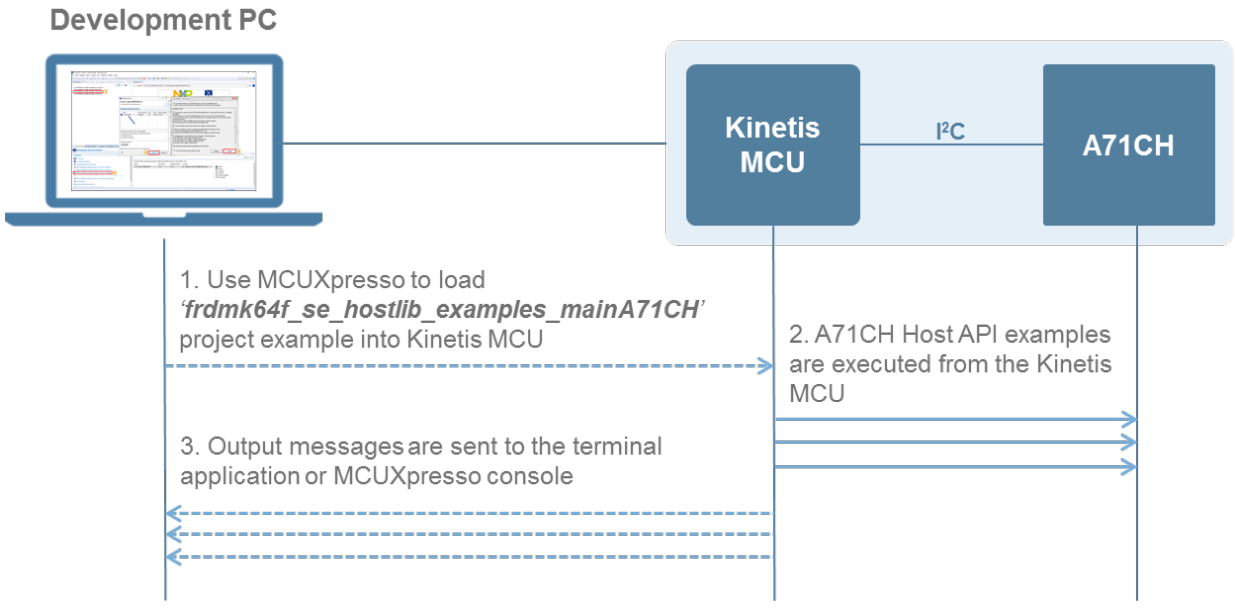


Fig 29. Running A71CH Host API system setup (e.g. FRDM-K64F)

7.1.1 Printing the output with MCUXpresso IDE

Assuming that the SDK has already been installed, and the project examples have been imported as explained in section 6.4, open MCUXpresso IDE and take the following steps (Fig 30):

1. Select '*frdmk64f_se_hostlib_examples_mainA71CH*' project.
2. Click on Debug '*frdmk64f_se_hostlib_examples_mainA71CH*' [Debug] (Note that the name might be slightly different in future versions of the Host software package).
3. Select J-Link OpenSDA probe and click on 'OK'. Make sure the OpenSDA serial port is connected to the Windows platform (Fig 12, highlighted in red)
4. If a 'Terms of use' pop-up appears, check 'Do not show this message again for today' box and click 'Accept'.

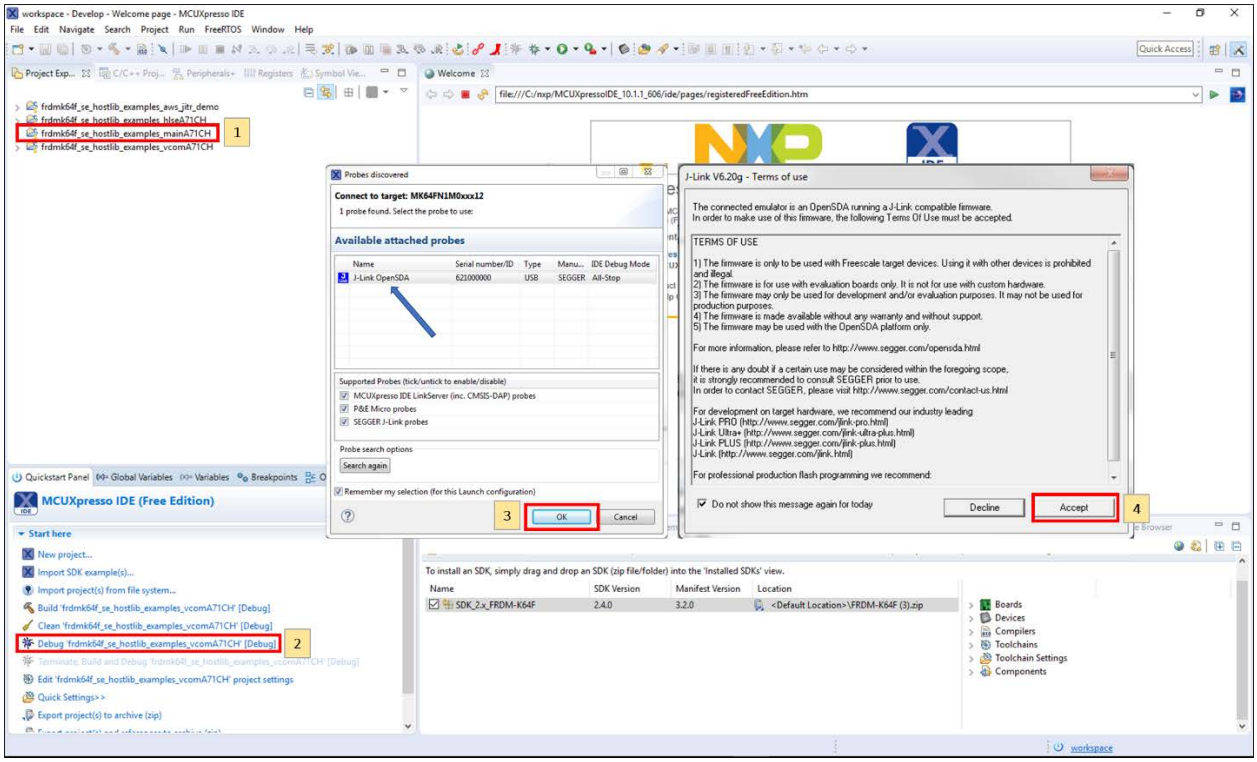


Fig 30. Configuration steps to debug using MCUXpresso IDE Console

After that, the project will start to compile and execute automatically. Once the process is finished, the user should press the '*F8'* key to run the program. Fig 31 shows the output in the console tab.

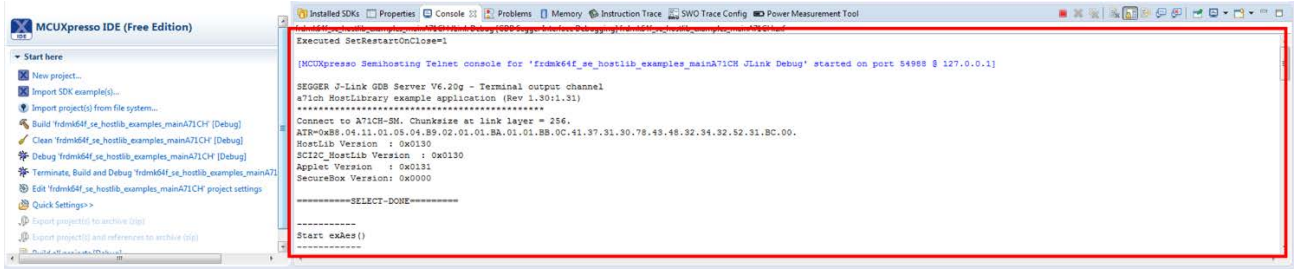


Fig 31. Generated output printed in MCUXpresso IDE Console

7.1.2 Printing the output using Tera Term

The output of the A71CH Host API usage example execution can be seen by using Tera Term. For this, the SDK Debug Console should be configured to be in '*UART Console'* mode. Fig 32 illustrates the process; simply click on '*Quick Settings'*, 'SDK Debug Console' and finally choose '*UART Console'*.

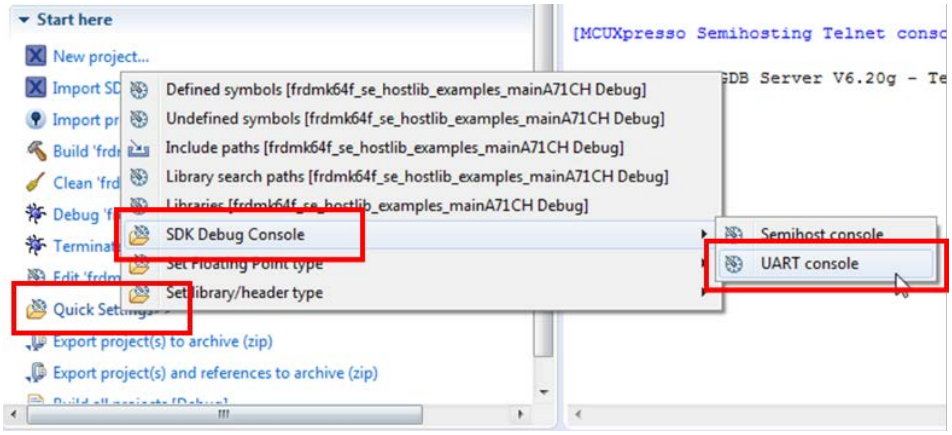


Fig 32. SDK Debug Console in UART Console mode

Once the SDK Debug Console is set to UART Console, it should be configured correctly (Fig 33):

1. Right click on the project and click on '*Properties'*.
2. Click on '*Settings'* (2.1) and '*Preprocessor'* (2.2). The variable '*SDK_DEBUGCONSOLE'* should be set to '0'. If it is already '0', skip this step. On the contrary, if it is set to '1', double click the variable and a new Edit dialog will appear. Change the value from '1' to '0' and finally click '*OK'*. A new pop-up will warn the user that the project should be rebuilt to correctly apply the changes.
3. A new variable must be mentioned. Following the same steps described in the previous step, click on '*Add*…' (3.3). '*Enter value'* dialog will appear. Write '*SDK_DEBUGCONSOLE*_UART' and click '*OK'*.
4. Click on '*Apply'* or '*OK'* to save the changes. A new pop-up will warn the user that the project has to be rebuilt to correctly apply the changes.

Note: If the option UART console or Semihost console cannot be clicked, make sure that there is not a running session. If there is a running session (a running thread inside '*Debug'* window), click on the *'Terminate all debug sessions'* button.

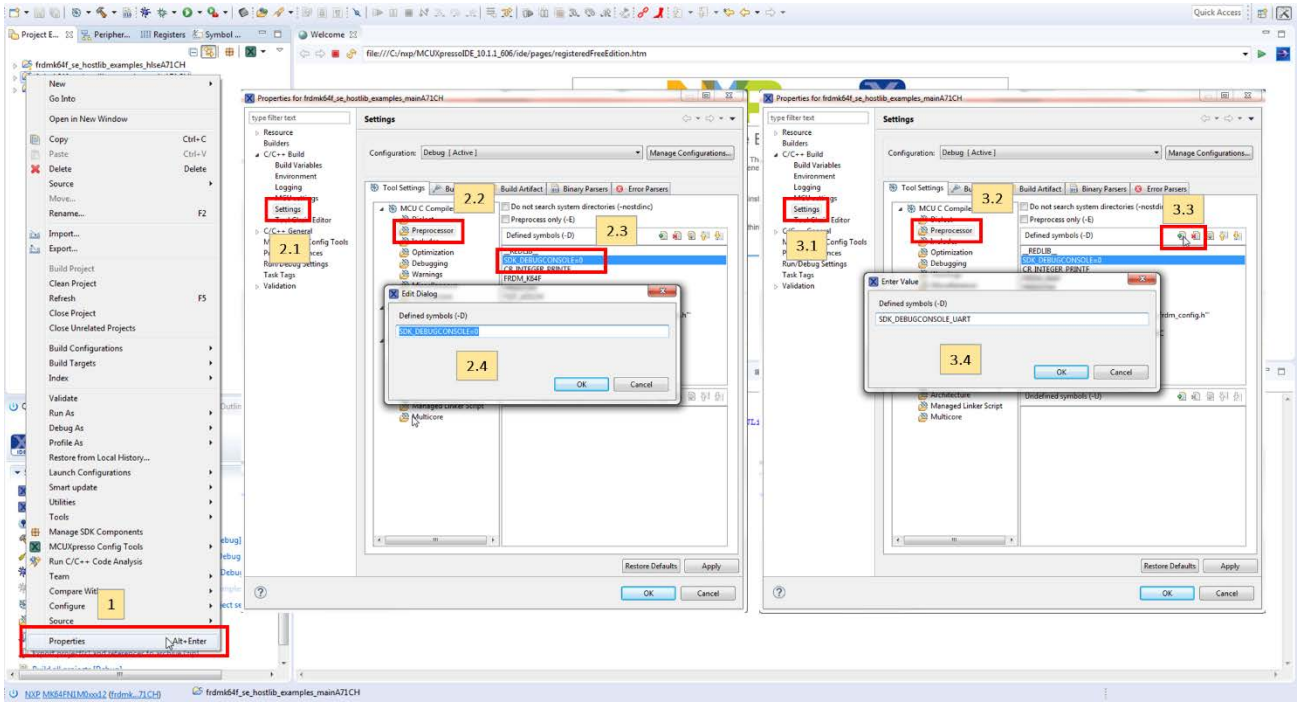


Fig 33. MCUXpresso UART console configuration

Finally, to run the example project:

1. First, click on '*Build* '<*project name*>' [Debug]'.
2. Then, click on Debug '<*project name*>' [Debug]' to run the project in the Kinetis board.

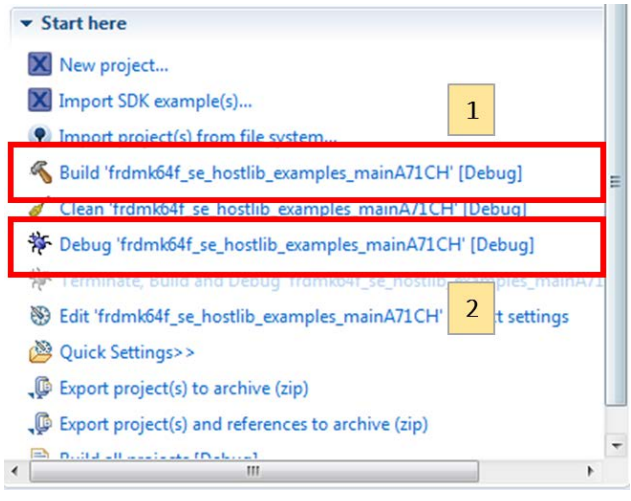


Fig 34. Build and Debug options

On Tera Term, the first thing that should be configured is a new connection (Fig 35). The user should choose a Serial connection and a port. This port can be checked in the Window's device manager under "*Ports (COM & LPT)*" menu.

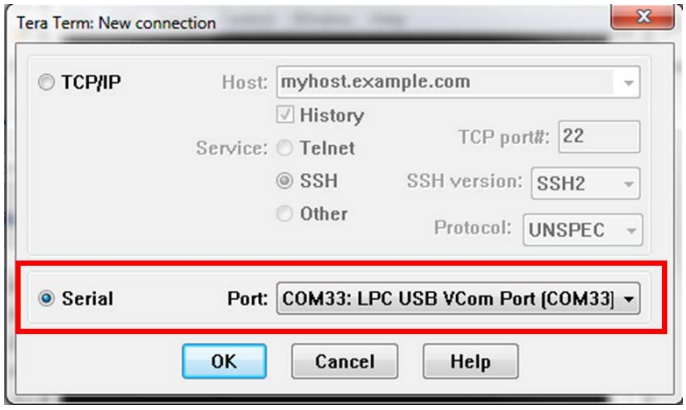


Fig 35. Tera Term new connection

Then, the selected port should be set as shown in Fig 36.

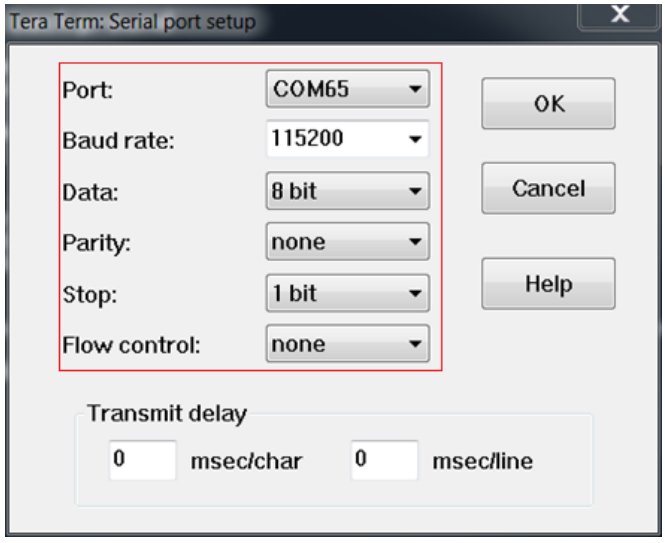


Fig 36. Serial port setup

The terminal window should be configured as shown in Fig 37.

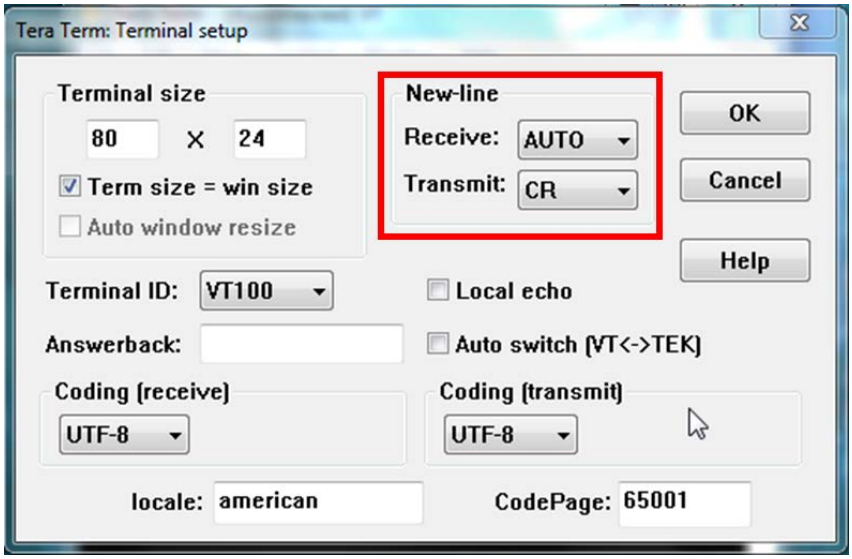


Fig 37. Terminal setup

Finally, Tera Term will print the output generated by the A71CH Host API example applications (Fig 38).

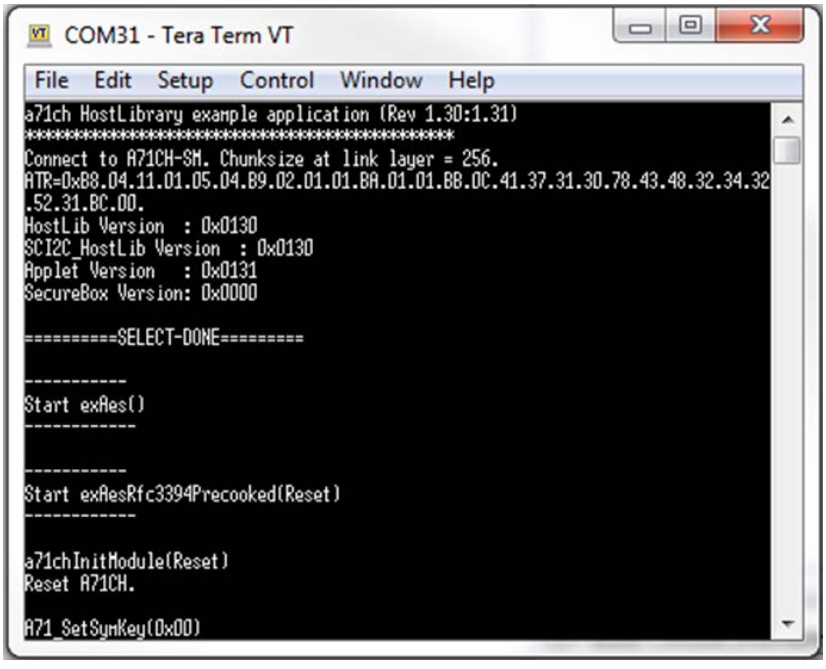


Fig 38. Generated output printed in Tera Term

7.2 Running A71CH Configure tool

In order to run the A71CH Configure tool, the system architecture should be modified. In this case, the Kinetis board will be programmed to behave as a USB to I²C adapter and the A71CH Configure tool application will be executed from a development PC; e.g., a Windows platform.

Note: In this case, Kinetis board refers to the FRDM-K64F and FRDM-K82F as the FRDM-KW41Z cannot be configured as VCOM port

Fig 39 illustrates the system architecture of this scenario.

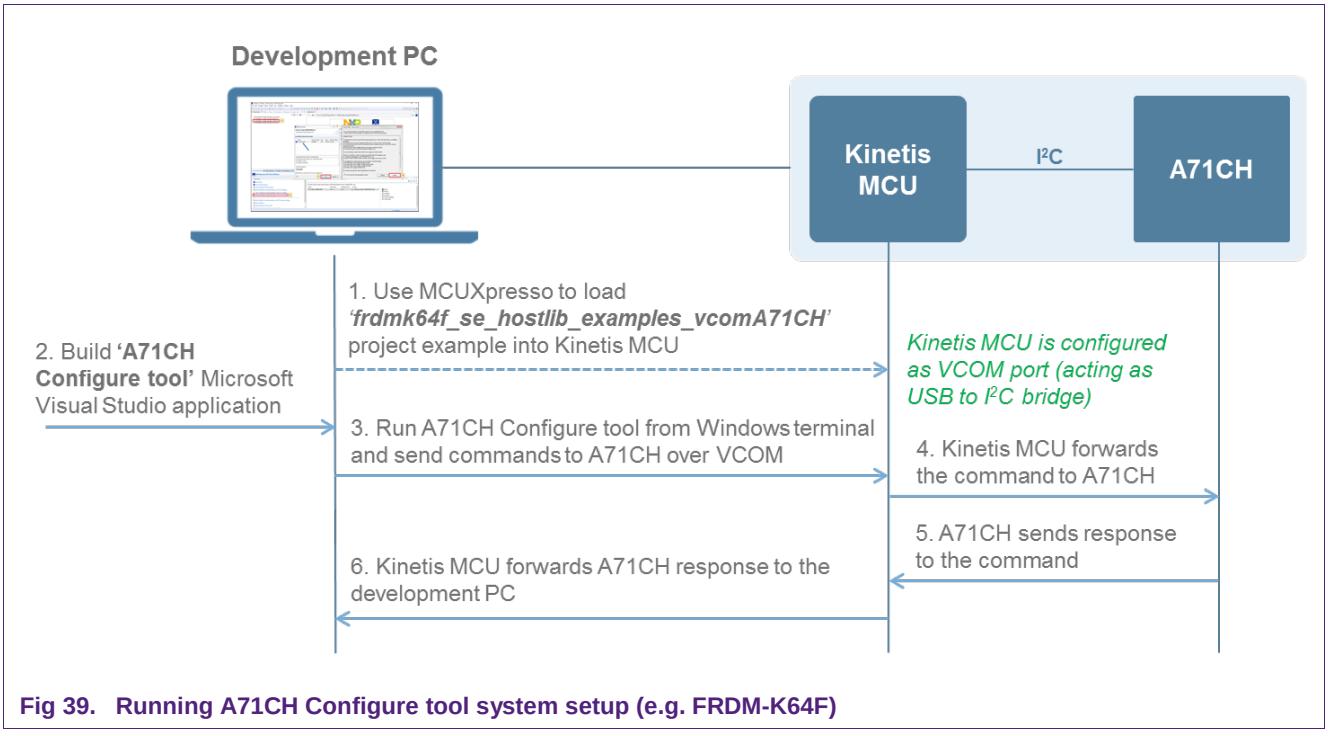


Fig 39. Running A71CH Configure tool system setup (e.g. FRDM-K64F)

The A71CH Configure tool application is built using Microsoft Visual Studio and it is launched from a terminal console; e.g., Windows PowerShell console.

The A71CH Configure tool application is based on the Host library and, therefore, will use the functions contained in the A71CH Host API or generic API HLSE, the APDU layer and the smComSerial communication layer to exchange APDUs between the Windows platform and the Kinetis board over USB.

On the other side, the Kinetis board will be programmed with the '*xxx_se_hostlib_examples_vcomA71CH'* project to act as a USB to I²C adapter, thus receiving the incoming APDUs from the USB and sending these to the A71CH security IC over I²C interface. Again, the Host library will provide the required layers to establish this communication.

Taking everything into account, the following steps must be followed in order to prepare a Windows-based device for running the A71CH Configure tool application:

- Load '*xxx_se_hostlib_examples_vcomA71CH*' program into the Kinetis using MCUXpresso.
- Connect the Kinetis to the Windows platform over USB
- Build the A71CH Configure tool application with Microsoft Visual Studio.
- Start the built application from a Windows terminal passing the virtual COM port address as input argument.

7.2.1 Set the Kinetis board as virtual COM port

Similarly, as in section 0, the '*xxx_se_hostlib_examples_vcomA71CH*' project can be flashed into the Kinetis MCU by opening the MCUXpresso IDE, selecting it from the list of imported projects, and debugging it. Fig 40 shows the MCUXpresso IDE screen; the '*xxx_se_hostlib_examples_vcomA71CH*' project and the 'debug' option have been highlighted in red. If the user selects the '*Debug'* option, the project is flashed to the memory of the board and it will work standalone.

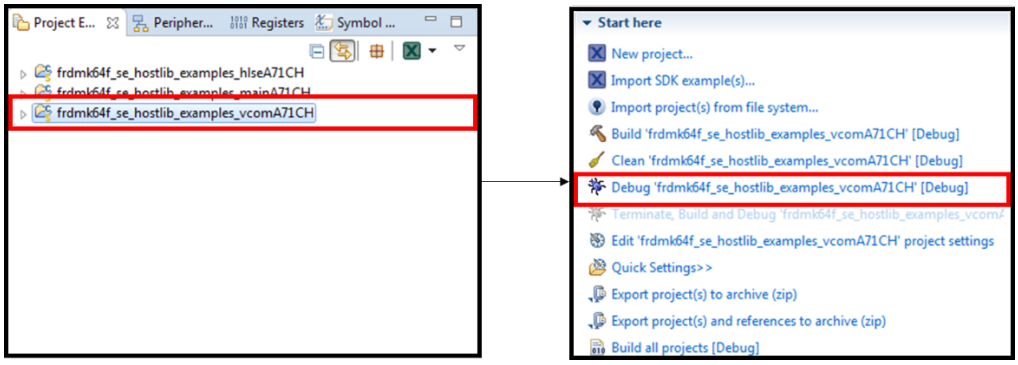


Fig 40. Loading VCOM project into the Kinetis board

7.2.2 Connect the Kinetis board to the Windows platform over USB

Once the Kinetis board has been configured as a virtual COM port, connect it to the Windows-based platform through the USB port highlighted in yellow in Fig 12.

It is possible that the development PC does not recognize the Kinetis board. Check Appendix in section 8 for drivers troubleshooting.

7.2.3 Build the A71CH Configure tool application with Microsoft Visual Studio

The A71CH Configure tool will be executed from the Windows platform. Fig 41 indicates the path to the A71CH Configure tool files (depending on the Microsoft Visual Studio installed version). Open the project by double-clicking '*_openssl_a71ch_vcom_x86.sln*'. Microsoft Visual Studio will automatically open.

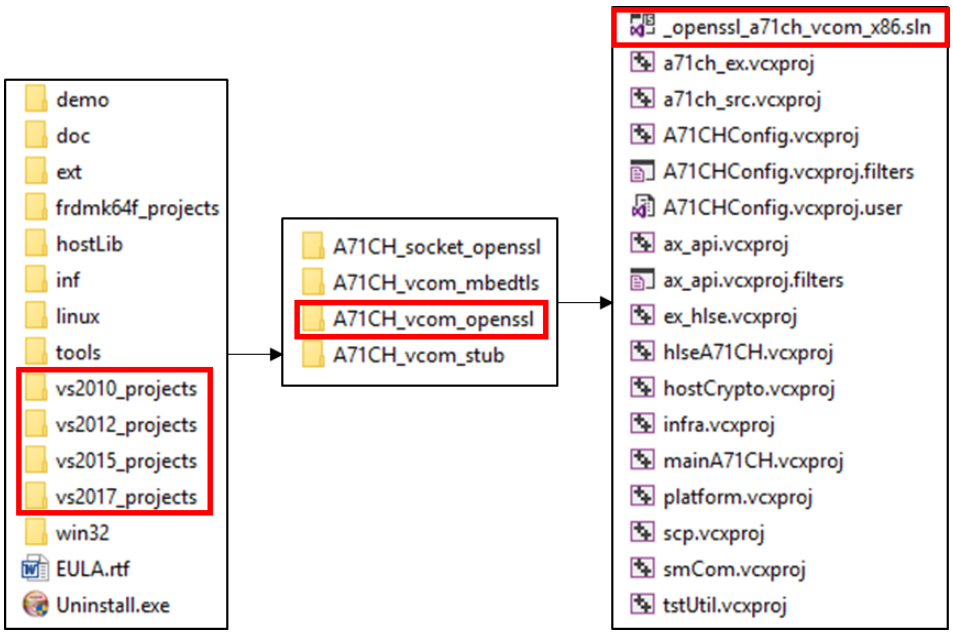


Fig 41. Localization of the Visual Studio A71CH projects in the Host Library

To configure the virtual COM address into the a71chConfig project, do the following:

1. Right-click '*A71CHConfig*' in the '*Solution Explorer'* tab and open the '*Properties'* window.
2. Then, in '*Configuration Properties – Debugging'* tab make sure the virtual COM address is set in the '*Command Arguments'* field.
3. Once the virtual COM port has been defined, invoke menu Build - Build Solution to create the executable.

Fig 42 illustrates the above-mentioned steps to build the project in Microsoft Visual Studio. The 'Build Solution' option has been highlighted in red, while the '*Properties'* and '*Command Arguments'* field have been highlighted in green. In addition, each step has been numbered in the figure.

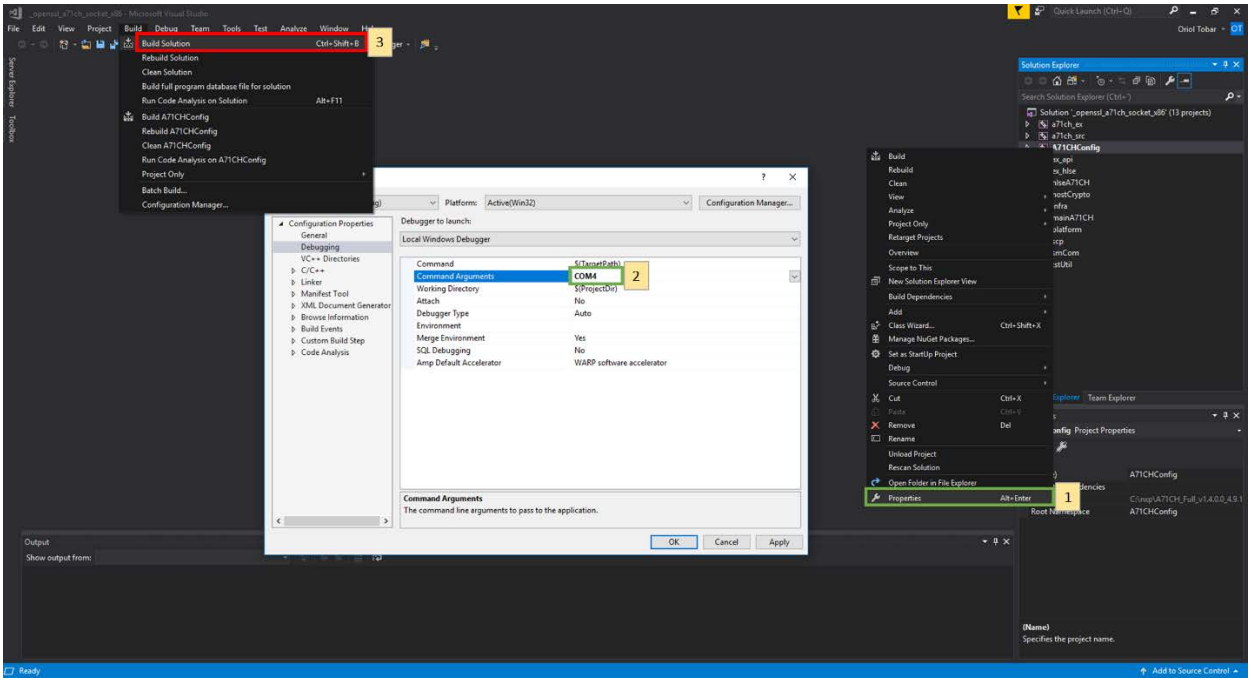


Fig 42. Visual Studio 2017. Building the a71chHostLib project

The resulting executable will be named '*A71CHConfig_vcom.exe*' and will be in the '*tools'* folder. In order to run the obtained executable, open a PowerShell window in Debug folder. This can be easily done by right clicking the folder while pressing the shift key.

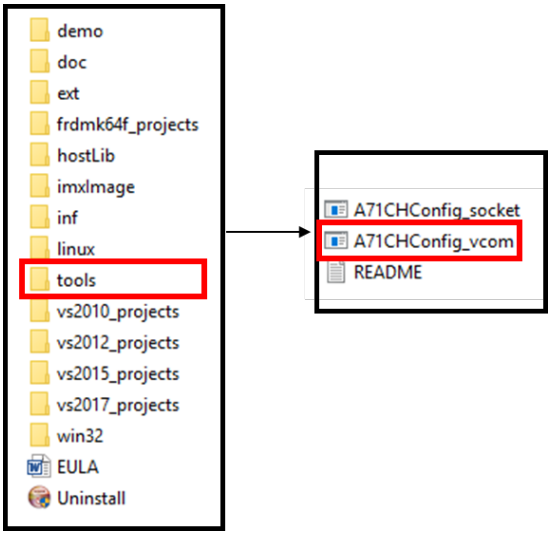


Fig 43. Win32-vs2015 folder internal structure. Open a PowerShell window in a71chHostLib\Debug folder

Once in the PowerShell window, it is possible to search and list all the existing .exe files (executables) with the following command:

```
dir *.exe
```

As shown in Fig 44, the previously built A71CHConfig project generated an executable inside the tools folder.

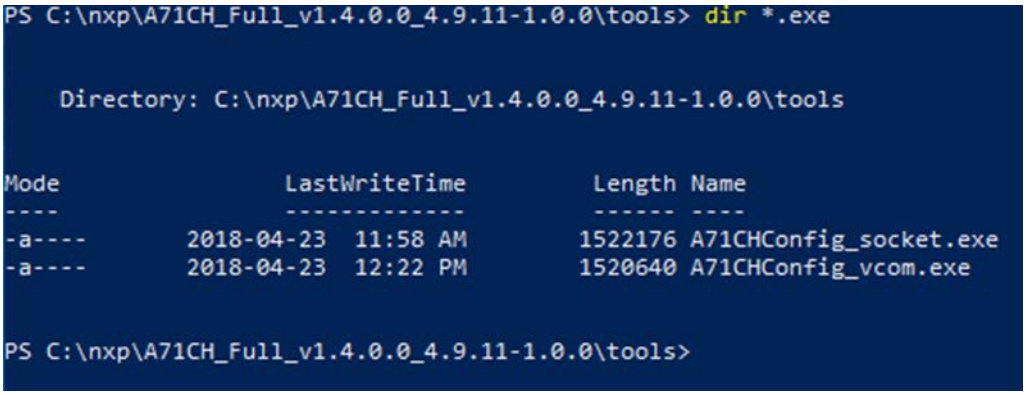


Fig 44. Executable file path

Finally, the A71CH Configure tool can be run with the following command:

```
.\a71chConfig_vcom.exe COM4 <input_arguments>
```

Where ‘COM4’ is the virtual COM port direction. The A71CH Configure tool will be launched and the connection between the Windows platform and the A71CH over the Kinetis board will be established.

7.3 Running mbedTLS scripts

The mbedTLS scripts are a set of examples that demonstrate the integration of mbedTLS software stack with A71CH. This setup uses the Kinetis board as USB to I²C adapter and the development PC configured both as a standalone mbedTLS server and mbedTLS client. Fig 45 illustrates the system architecture of this scenario.

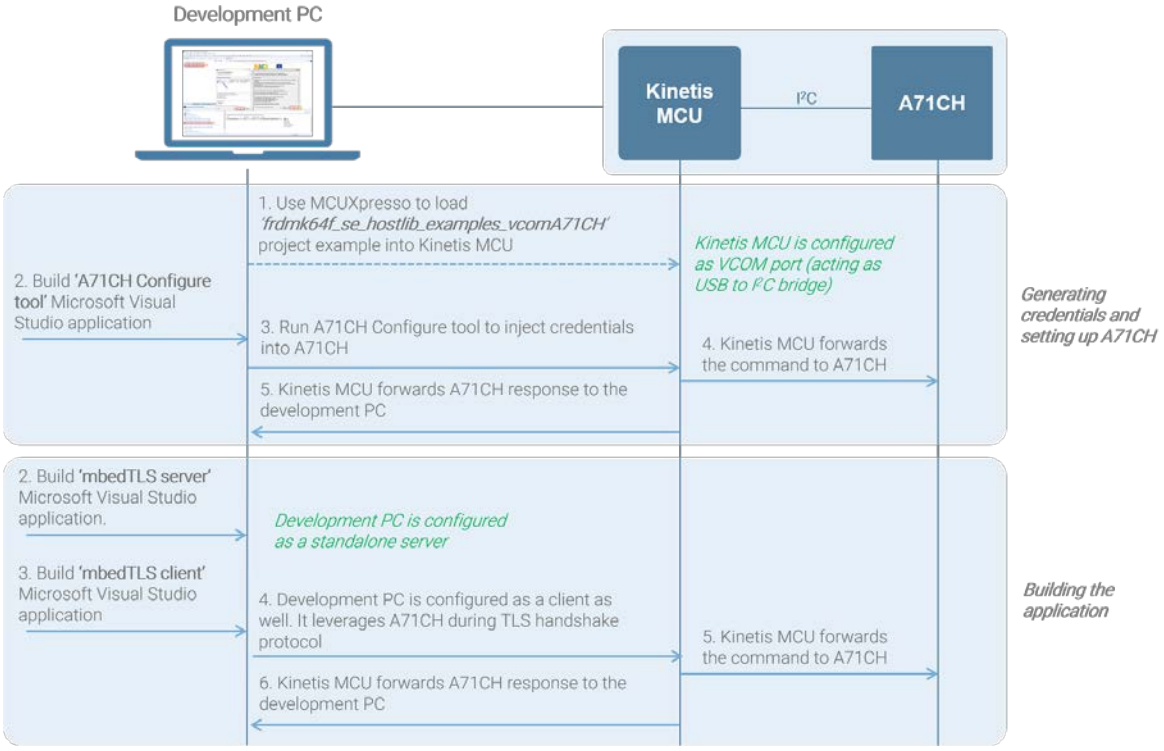


Fig 45. Running mbedTLS scripts system setup (e.g. FRDM-K64F)

Note: In this case, Kinetis board refers to the FRDM-K64F and FRDM-K82F as the FRDM-KW41Z cannot be configured as VCOM port

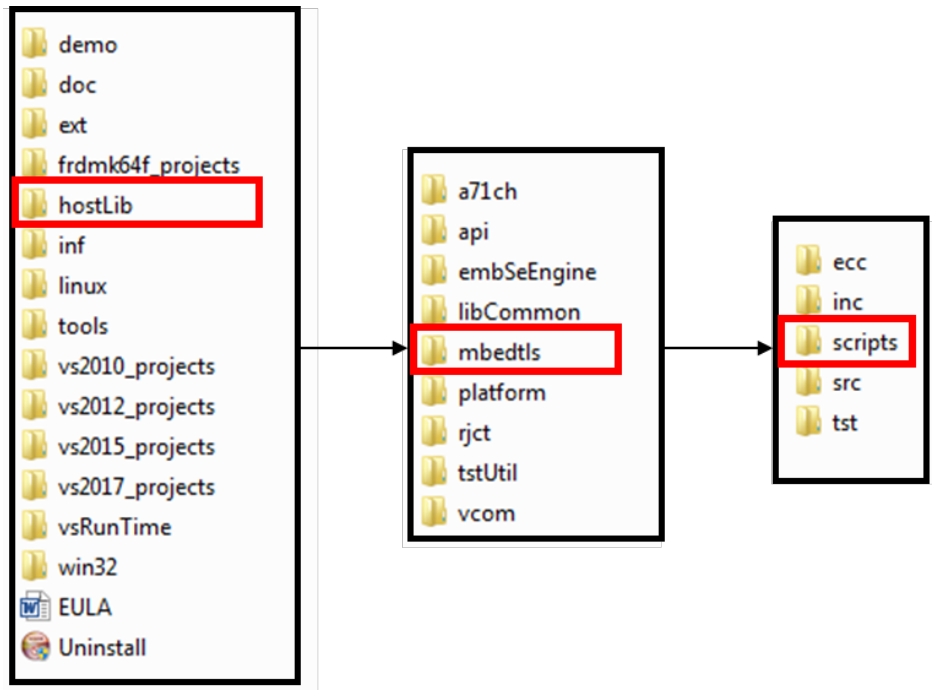


Fig 46. Scripts folder location

The examples contained in the A71CH Host software package are separated into two parts:

- **Stand Alone examples**: These examples are standalone examples and do not use the A71CH at all. These examples depend on command line arguments to get the necessary credentials for the demonstration. These examples have '*_sa_*' in their file name
- **A71CH examples**: These examples depend on A71CH for cryptographic operations. These examples do not use keys or certificates from any file system, but rather use keys at specific indexes in the A71CH or certificates in the GP storage of the A71CH. These examples have '*_ax_*' in their file name.

In order to execute these examples there is a series of prerequisites to be met. The following steps are needed to prepare the credentials and the A71CH to run the mbedTLS examples; the necessary scripts are in '*scripts'* folder inside the '*mbedtls'* folder, as can be seen in Fig 46:

1. Generate the involved credentials using the OpenSSL commands in the development PC.
2. Program the Kinetis to behave as a VCOM.
3. Inject the created credentials into the A71CH running the A71CH Configure tool in the development PC.

The following section explains each one of the prerequisite steps.

7.3.1 Generate credentials

The first step is to generate the credentials. By default, there isn't any key or certificate injected in the A71CH.Run the file '*RunOnce_CreateCertificates.bat*' (double click). After finishing the execution of this file; inside the '*ecc'* folder, new files will appear (Fig 47):

- The parameters of the ECC curve used.
- Self-signed CA credentials and public key.
- Certificate and public key for the client
- Certificate and public key for the server

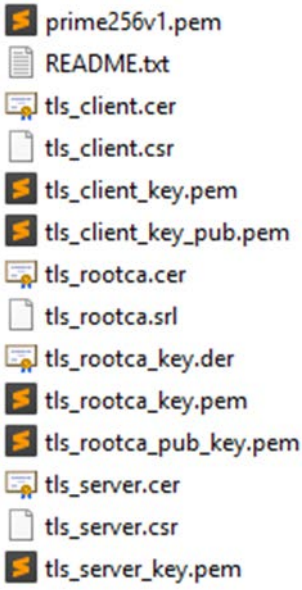


Fig 47. Generated files after executing 'RunOnce_CreateCertificates.bat'

7.3.2 Connect Kinetis and program it to behave as VCOM

To set up the A71CH and the Kinetis board to behave as a VCOM port, follow the steps in section 7.2.1.

7.3.3 Inject credentials into the A71CH

In order to inject the created credentials into the A71CH, it is required to build the A71CH Configure tool application example as explained in section 7.2.3. Assuming we already have the A71CH Configure tool executable, launch the .bat file '*ResetAndUpdateA71CH.bat*' as seen in Fig 48, indicating the VCOM port configured in section 7.3.2. This .bat file initiates the A71CH Configure tool in '*script'* mode; i.e., the script name must be passed as input argument. Then, the commands contained in that script are executed. More concretely, it calls the script '*ResetAndUpdateA71CH.script.txt*' which contains the following A71CH Configure tool commands:

- Set pair

- Set pub
- Set gp
- Info pair
- Info pub

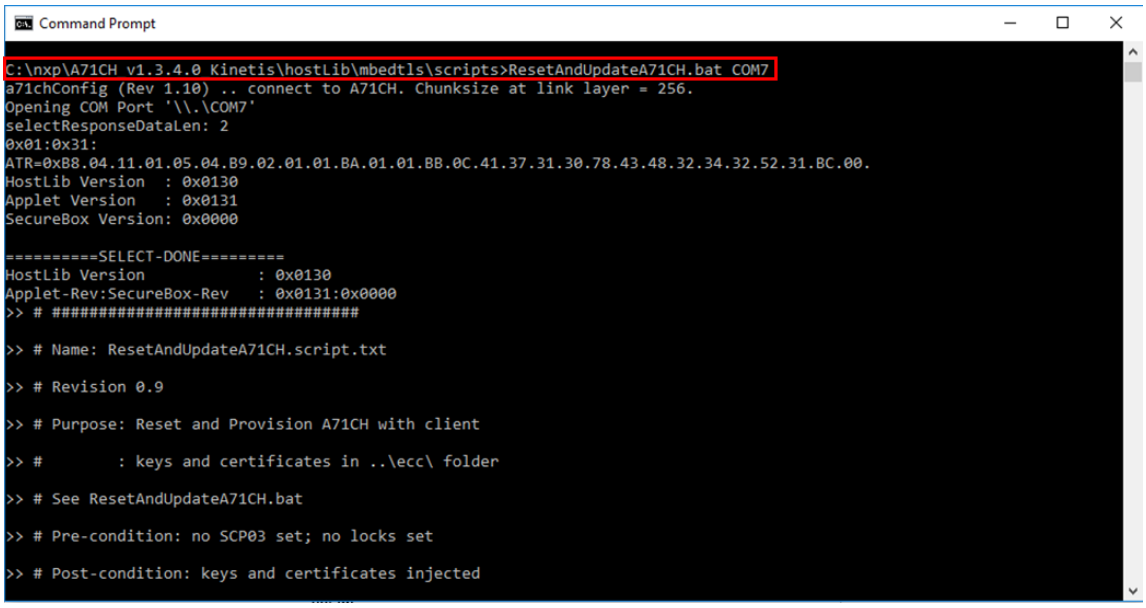


Fig 48. Credentials injection using 'ResetAndUpdateA71CH.bat'

At this point, the A71CH will be provisioned and we will be able to run the mbedTLS examples.

7.3.4 mbedTLS examples

The following examples demonstrate the integration between the A71CH and mbedTLS:

- TLS server/client
- DTLS server/client
- ECDSA verify

Now the example of TLS server/client is explained. For further information about all the available mbedTLS examples, please refer to the Doxygen documentation included in the [A71CH_HOST_SW].

7.3.5 mbedTLS server/client

The mbedTLS server/client example application is a demonstration of how to establish a TLS-based connection between a server and a client. In this case, the A71CH will contain the client credentials and will be involved in the connection establishment.

Fig 49 shows the contents of the '*script*' folder. To run the TLS server/client example, two scripts are required:

- start_a71ch_SSL2_client.bat
- start_standalone_SSL2_server.bat

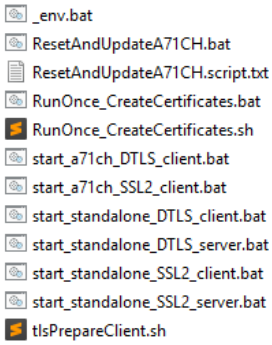


Fig 49. Contents of 'scripts' folder

In addition, the following Visual Studio projects should be built:

- mbedTLS_sa_ssl_server2
- mbedTLS_ax_ssl_client2

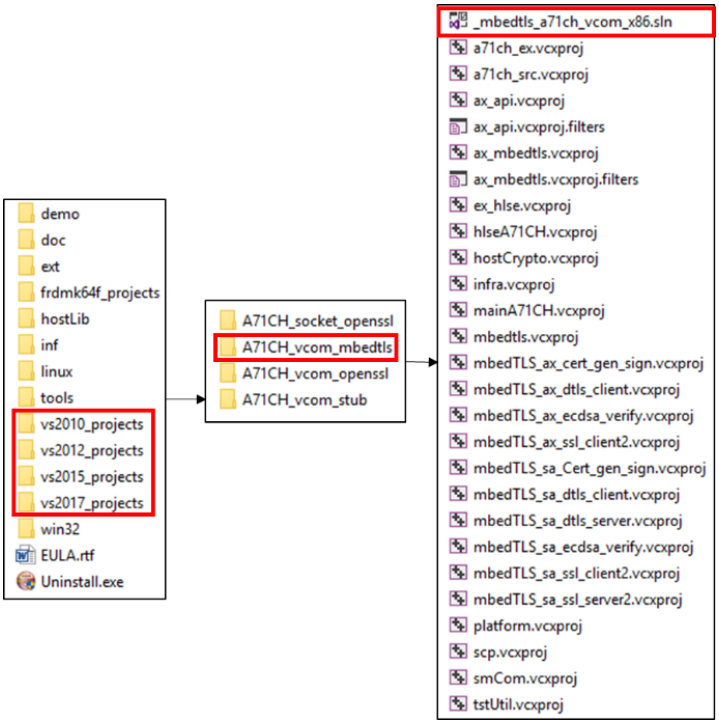


Fig 50. Contents of the 'A71CH_vcom_mbedtls' folder

Fig 50 shows the location of the '*_mbedtls_a71ch_vcom_x86.sln*' Visual Studio project. Once opened, we will be able to select the applications to be built. Remember to make sure the virtual COM address is set in the '*Command Arguments'* field of the '*Configuration Properties – Debugging'* tab as explained in section 7.2.3.

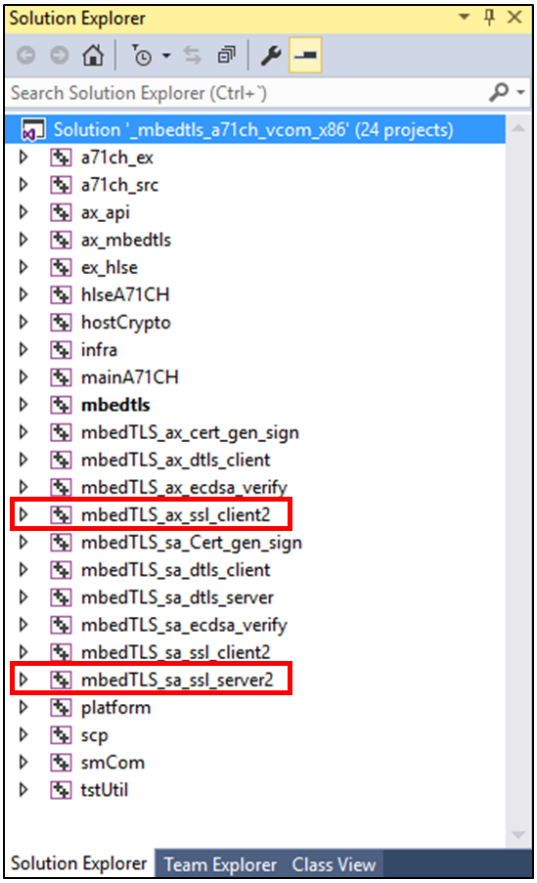


Fig 51. Visual Studio 2017 Solution Explorer tab

Finally, both the server and the client can be started. To start the server, run the '*start_standalone_SSL2_server.bat*' file. A new terminal window will open, and the server will start to wait for remote connections as depicted in Fig 52.

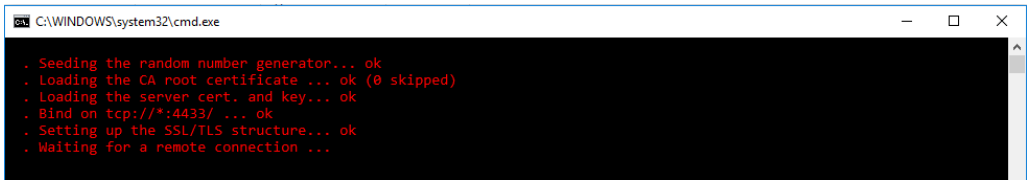


Fig 52. Standalone SSL2 server

Then, the client can be started by running the '*start_a71ch_SSL2_client.bat*' file from a terminal. Note that in this case the virtual COM port number should be passed as input

argument (Fig 53). The client application will try to establish a TLS connection with the server using mbedTLS. The A71CH will be involved in the TLS handshake protocol, providing the required credentials. A simple string '*ABCDE…*' will be sent to the server just to check that the connection has been properly established.

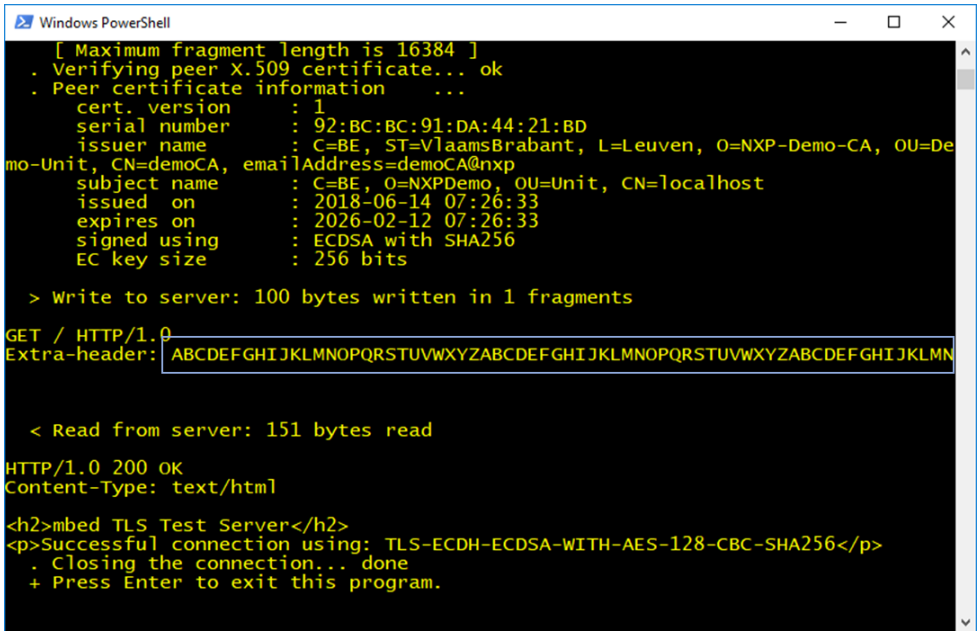


Fig 53. A71CH SSL2 client

Fig 54 shows when the server application receives the '*ABCDE…*' string and prompts it in the terminal.

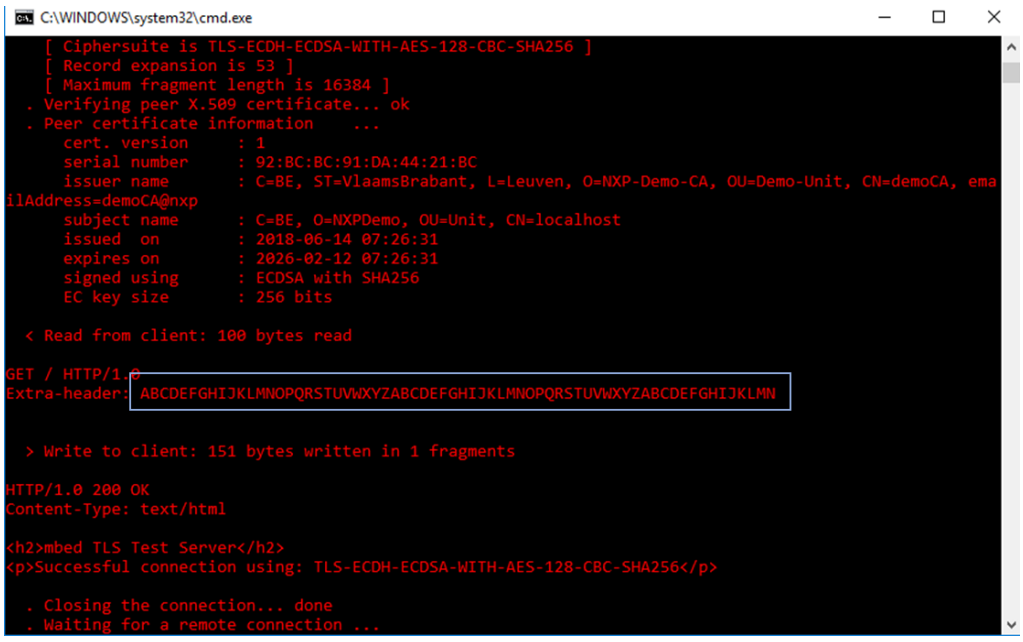


Fig 54. Standalone SSL2 server

8. Appendix: VCOM driver installation troubleshooting

Once the Kinetis board has been configured to act as a USB to I²C adaptor, the Windows platform will be able to detect and assign it a virtual COM port number. For this, the Kinetis board should be connected to the Windows platform as depicted in Fig 55.

Fig 55. Left USB connection is used to flash the program from MCUXpresso IDE. Right USB port connection acts as a virtual COM port

To ensure that the Kinetis is correctly recognized, open the '*Device Manager'* control panel. The Kinetis board should be detected and labeled as *'…VCOM Port (COMX)*' within the '*Ports (COM & LPT)'* drop-down (Fig 56).

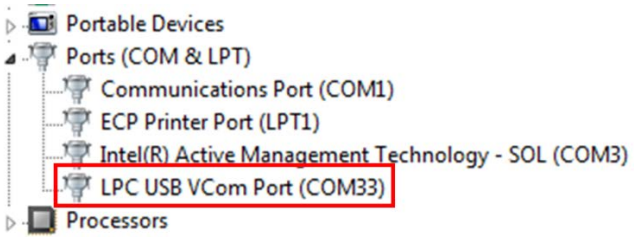


Fig 56. Drivers correctly installed

However, it is possible that Windows will not detect the Kinetis board or that its drivers need to be updated. In these cases, it would be labeled as '*MCU VIRTUAL COM DEMO'* within '*Other devices'* as shown in Fig 57.

Fig 57. Device not detected correctly

In order to update the drivers, do the following:

1. Right-click on '*MCU VIRTUAL COM DEMO'*.
2. Click on Update Driver.
3. '*Browse my computer for driver software'*.
4. '*Let me pick from a list of device drivers on my computer'*.
5. Select Ports (COM & LPT).
6. Un-check '*Show compatible hardware'*.
7. Select '*NXP'* and '*LPC USB VCOM Port'*.
8. Ignore the warning message and click on '*Yes'*.

9. References

All the references contained in this document are listed in the following table:

Table 3. References

[A71CH_HOST_SW]	**A71CH Host Software Package (Bash installer for Windows) –** DocStore, document number sw4673xx[1], Version 01.04.00 (or later), available on www.nxp.com/A71CH **A71CH Host Software Package (Bash installer for Linux) –** DocStore, document number sw4672xx[1], Version 01.03.00 (or later), available on www.nxp.com/A71CH
[AN_A71CH_HOST_SW]	**AN12133 A71CH Host software package documentation** – Application note, document number 4643**[1]
[QUICK_START_WIN]	**AN12134 Quick start guide for Windows** – Application note, document number 4644**[1]
[TERA_TERM]	**Tera Term terminal** - https://osdn.net/projects/ttssh2/releases/
[MCUXPRESSO_IDE]	**MCUXpresso IDE -** https://www.nxp.com/support/developer-resources/software-development-tools/mcuxpresso-software-and-tools/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE
[OPENSDA_FIRMWARE]	**OpenSDA / OpenSDA V2 website -** https://www.segger.com/products/debug-probes/j-link/models/other-j-links/opensda-sda-v2/
[MBED_TLS]	**mbedTLS website -** https://tls.mbed.org/
[SDKBUILDER]	**MCUXPresso SBKBuilder website -** https://mcuxpresso.nxp.com/en/select
[FRDM_K64F]	**Kinetis FRDM-K64F -** https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/kinetis-cortex-m-mcus/k-seriesperformancem4/k2x-usb/freedom-development-platform-for-kinetis-k64-k63-and-k24-mcus:FRDM-K64F
[FRDM_K82F]	**Kinetis FRDM-K82F -** https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/kinetis-cortex-m-mcus/k-seriesperformancem4/k8x-secure/freedom-development-platform-for-kinetis-k82-k81-and-k80-mcus:FRDM-K82F
[FRDM_KW41Z]	**Kinetis FRDM-KW41Z -** https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/kinetis-cortex-m-mcus/w-serieswireless-conn.m0-plus-m4/freedom-development-kit-for-kinetis-kw41z-31z-21z-mcus:FRDM-KW41Z

[1] **… document version number

10. Legal information

10.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

10.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

10.1 Licenses

ICs with DPA Countermeasures functionality

NXP ICs containing functionality implementing countermeasures to Differential Power Analysis and Simple Power Analysis are produced and sold under applicable license from Cryptography Research, Inc.

10.2 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

FabKey — is a trademark of NXP B.V.

I²C-bus — logo is a trademark of NXP B.V.

11. List of figures

12. List of tables

13. Contents

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

Date of release: 09 July 2018 492310

Document identifier: AN12135