

# AN12398

## EdgeLock SE05x Quick start guide with Visual Studio project examples

Rev. 2.5 — 12 September 2022  
534625

Application note

### Document information

Information	Content
Keywords	EdgeLock SE05x, EdgeLock SE05x Plug & Trust Middleware, Visual Studio projects
Abstract	This document explains how to get started with EdgeLock SE05x Plug & Trust middleware using the Visual Studio project examples. It provides detailed instructions to run the Microsoft Visual Studio projects using the CMake-based build system included in the EdgeLock SE05x Plug & Trust middleware.



## Revision history

### Revision history

Revision number	Date	Description
1.0	2019-06-08	First document release
1.1	2019-06-20	Update of board figures
2.0	2019-11-25	Update of CMake build system materials
2.1	2019-12-17	Corrected OM-SE05xARD J14 jumper setting
2.2	2020-11-19	Update for EdgeLock SE051
2.3	2020-12-07	Updated to the latest template and fixed broken URLs
2.4	2022-06-02	<p>Add EdgeLock SE050E and EdgeLock A5000 product variants.</p> <p>Update <a href="#">Table 1</a>, <a href="#">Figure 1</a>, <a href="#">Figure 2</a>, <a href="#">Figure 3</a>, <a href="#">Figure 5</a>, <a href="#">Figure 6</a>, <a href="#">Figure 7</a>, <a href="#">Figure 8</a>, <a href="#">Figure 9</a>, <a href="#">Figure 10</a>, <a href="#">Figure 11</a>, <a href="#">Figure 12</a>, <a href="#">Figure 13</a>, <a href="#">Figure 14</a>, <a href="#">Figure 15</a>, <a href="#">Figure 16</a>, <a href="#">Figure 17</a>, <a href="#">Figure 18</a>, <a href="#">Figure 19</a>, <a href="#">Figure 20</a>, <a href="#">Figure 21</a>, <a href="#">Figure 22</a>, <a href="#">Figure 23</a>, <a href="#">Figure 31</a>, <a href="#">Figure 32</a>, <a href="#">Figure 33</a>, <a href="#">Figure 34</a>, <a href="#">Figure 35</a>, <a href="#">Figure 36</a>, <a href="#">Figure 43</a>, <a href="#">Figure 44</a> and <a href="#">Figure 45</a>.</p> <p>Add Section <a href="#">Section 5</a> Product specific CMake build settings.</p> <p>Add Section <a href="#">Binding EdgeLock SE05x to a host using Platform SCP</a> Binding EdgeLock SE05x to a host using Platform SCP.</p> <p>Update tool versions in <a href="#">Section 7</a>, <a href="#">Section 8</a> and <a href="#">Section 9</a>.</p> <p>Moved section "Update FRDM-K64F board with DAPLink firmware" into <a href="#">Section 10</a>.</p>
2.5	2022-07-04	<p>Update to EdgeLock SE Plug &amp; Trust Middleware version 04.02.xx.</p> <p>Update Section <a href="#">Section 5</a> Product specific CMake build settings.</p> <p>Update Section <a href="#">Binding EdgeLock SE05x to a host using Platform SCP</a> Binding EdgeLock SE05x to a host using Platform SCP.</p>

## 1 How to use this document

The Plug & Trust middleware package is delivered with the CMake files that include the set of directives and instructions describing the project's source files and targets. The CMake architecture allows developers to build files for their platform and native build environment and run exactly the same project example on PC/Windows/Linux and embedded targets.

This document provides detailed instructions to run Visual Studio examples provided in the Plug & Trust middleware using FRDM-K64F and OM-SE05xARD boards. The main body of this document should be used in this sequence:

1. Order board samples. [Section 2](#) contains the ordering details of the boards required in this document
2. Setup your boards. [Section 3](#) describes how to setup the OM-SE05xARD and FRDM-K64F boards.
3. Run project examples. Go to [Section 4](#) for instructions to import and run EdgeLock SE05x Visual Studio project examples.

Supplementary material has been provided in the appendices.

## 2 Hardware required

The EdgeLock SE05x works as an auxiliary security device attached to a host controller, communicating with through an I<sup>2</sup>C interface. To follow the instructions provided in this document, you need an EdgeLock SE05x development board and a FRDM-K64F MCU board, acting as a host controller.

### EdgeLock SE05x development boards ordering details

The EdgeLock SE05x and EdgeLock A5000 product support packages are providing development boards for evaluating EdgeLock SE05x and EdgeLock A5000 features. Select the development board of the product you want to evaluate. [Table 1](#) details the ordering details of the EdgeLock SE05x and EdgeLock A5000 development boards.

**Table 1. EdgeLock SE05x development boards.**

Part number	12NC	Description	Picture
<a href="#">OM-SE050ARD-E</a>	9354 332 66598	SE050E Arduino® compatible development kit	
<a href="#">OM-SE050ARD-F</a>	9354 357 63598	SE050 Arduino® compatible development kit	
OM-SE050ARD	9353 832 82598	SE050F Arduino® compatible development kit	
<a href="#">OM-SE051ARD</a>	9353 991 87598	SE051 Arduino® compatible development kit	
<a href="#">OM-A5000ARD</a>	9354 243 19598	A5000 Arduino® compatible development kit	

**Note:** The pictures in this guide will show SE050E, but all boards in [Table 1](#) can be used as well with the same hardware configuration.

### FRDM-K64F MCU board ordering details

[Table 2](#) details the ordering details for the FRDM-K64F board.

**Table 2. FRDM-K64F details**

Part number	12NC	Content	Picture
<a href="#">FRDM-K64F</a>	935326293598	Freedom development platform for Kinetis K64, K63 and K24 MCUs	 The image shows the FRDM-K64F development board, which is a blue printed circuit board (PCB) featuring a central Kinetis K64 microcontroller, various connectors, and component packages.

### 3 Boards setup

This section explains how to prepare the OM-SE05xARD boards and FRDM-K64F board to run the Plug & Trust middleware project examples. This consists of:

1. [OM-SE05xARD jumper configuration](#).
2. [OM-SE05xARD and FRDM-K64F board connection](#).

**Note:** If your FRDM-K64F board does not already contain the DAPLink firmware, you need to update the FRDM-K64F board as described in [Section 10](#).

#### 3.1 OM-SE05xARD jumper configuration

The OM-SE05xARD boards have jumpers that allow you to configure the I<sup>2</sup>C interface of EdgeLock SE05x secure elements via the Arduino header. Configure the jumper settings as shown in [Figure 1](#) to enable this option.

**Note:** For more information about the jumper settings, refer to [AN13539 OM-SE05xARD hardware overview](#).

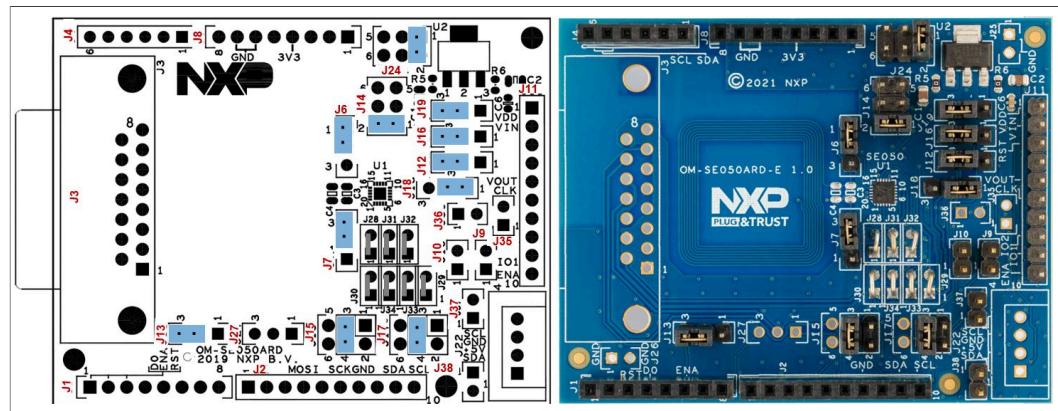


Figure 1. Jumper configuration for FRDM-K64F

#### 3.2 OM-SE05xARD and FRDM-K64F board connection

The OM-SE05xARD boards and FRDM-K64F board can be directly connected using the Arduino connectors. The OM-SE05xARD boards come with male connectors while the FRDM-K64F board comes with female headers.

Mount any OM-SE05xARD board on top of the FRDM-K64F as shown in [Figure 2](#):

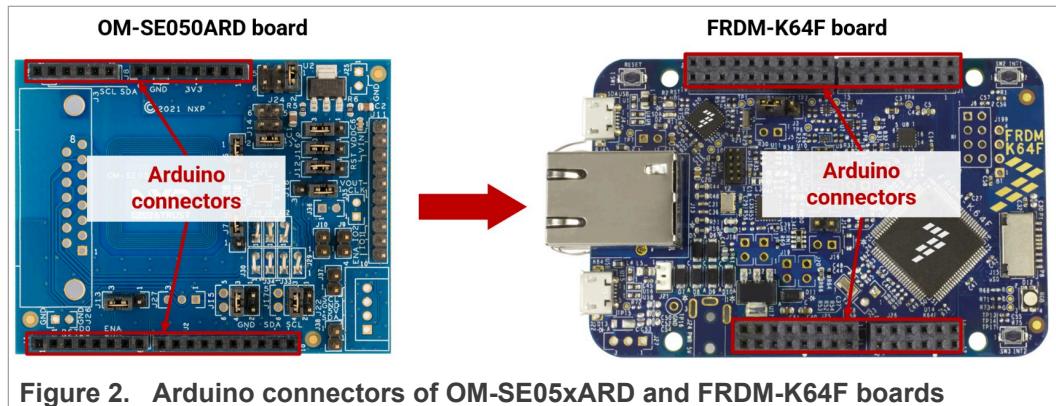


Figure 2. Arduino connectors of OM-SE05xARD and FRDM-K64F boards

Double check that the two boards are connected as shown in [Figure 3](#):

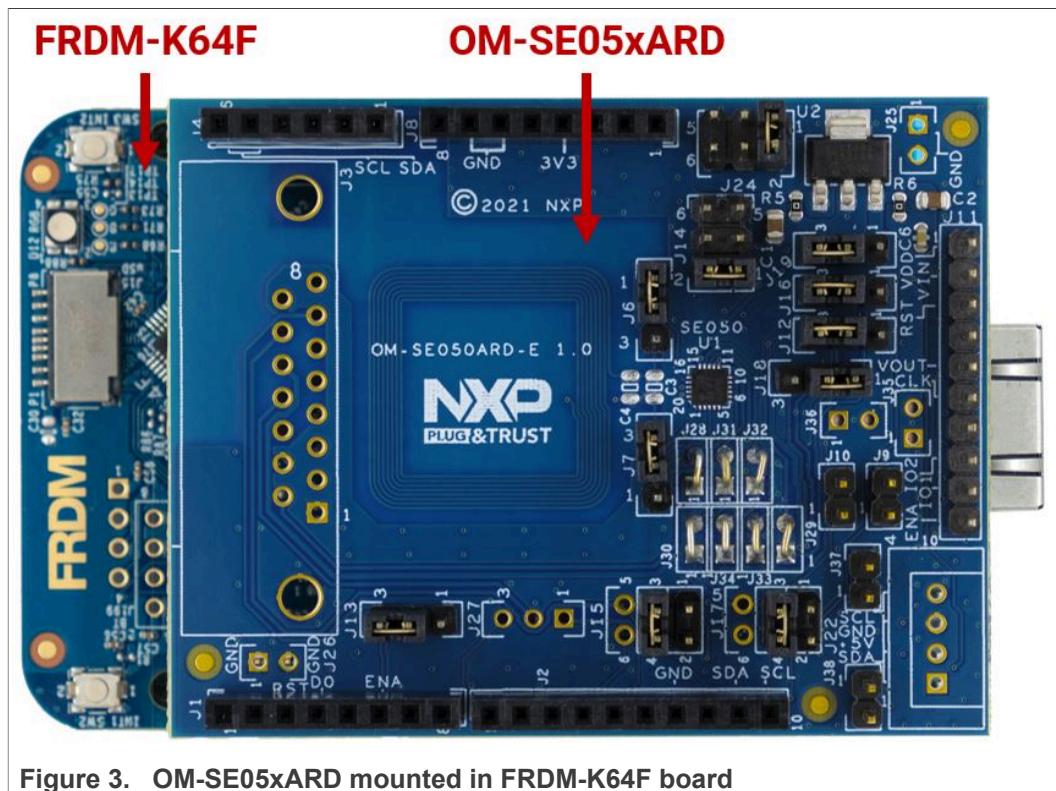


Figure 3. OM-SE05xARD mounted in FRDM-K64F board

**Note:** Refer to [Figure 1](#) for OM-SE05xARD jumper configuration.

### 3.3 Flash FRDM-K64F with VCOM software

The VCOM software allows the FRDM-K64F board to be used as a bridge between the Windows machine and the EdgeLock SE05x and enables the execution of the EdgeLock SE05x `ssscli` tool and other utilities from the laptop. To flash the VCOM software into the FRDM-K64F, follow these steps:

## EdgeLock SE05x Quick start guide with Visual Studio project examples

1. Unplug and plug again the USB cable to the openSDA USB port as shown in [Figure 4](#):

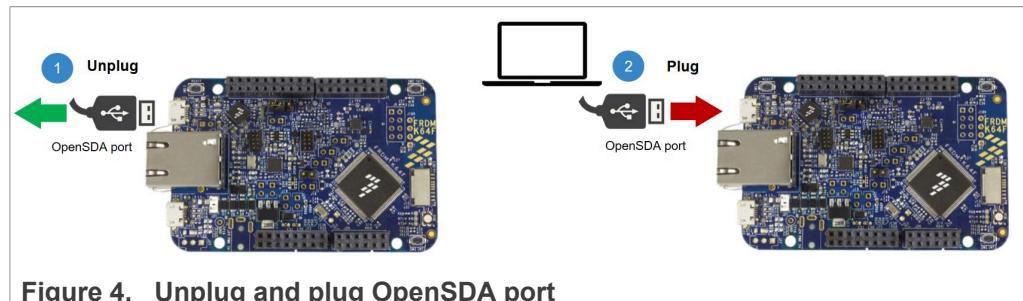


Figure 4. Unplug and plug OpenSDA port

2. When you plug the board, your laptop should recognize the board as an external drive as shown in [Figure 5](#):

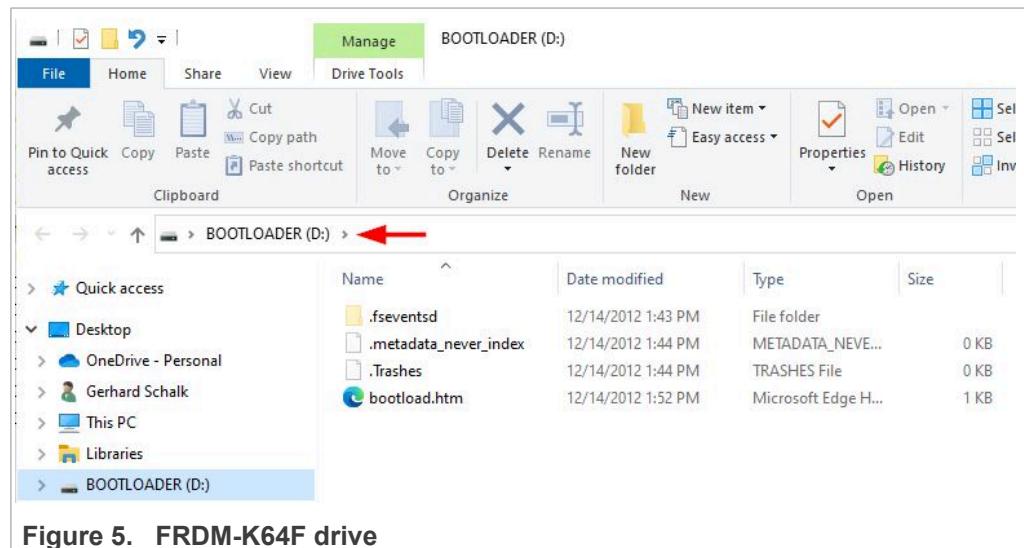


Figure 5. FRDM-K64F drive

## EdgeLock SE05x Quick start guide with Visual Studio project examples

3. Flash the VCOM software to FRDM-K64F. The VCOM software binary can be found in the Plug & Trust middleware package, inside the `simw-top\binaries` folder as shown in [Figure 6](#):

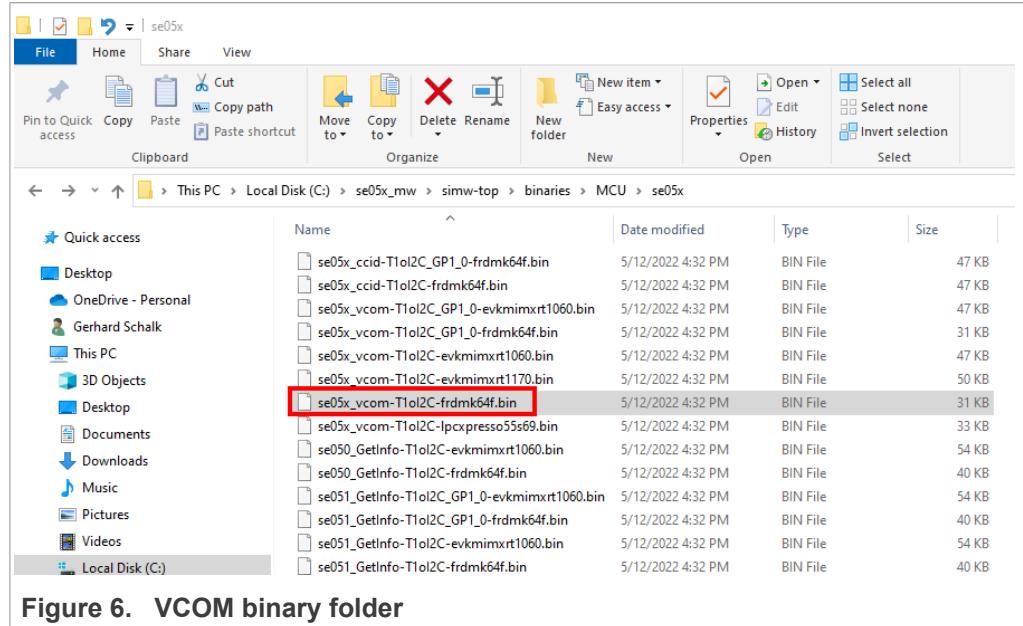


Figure 6. VCOM binary folder

4. Drag and drop or copy and paste the `a7x_vcom-T1oI2C-frdmk64f-SE050x.bin` file into the FRDM-K64F drive from your computer file explorer as shown in [Figure 7](#):

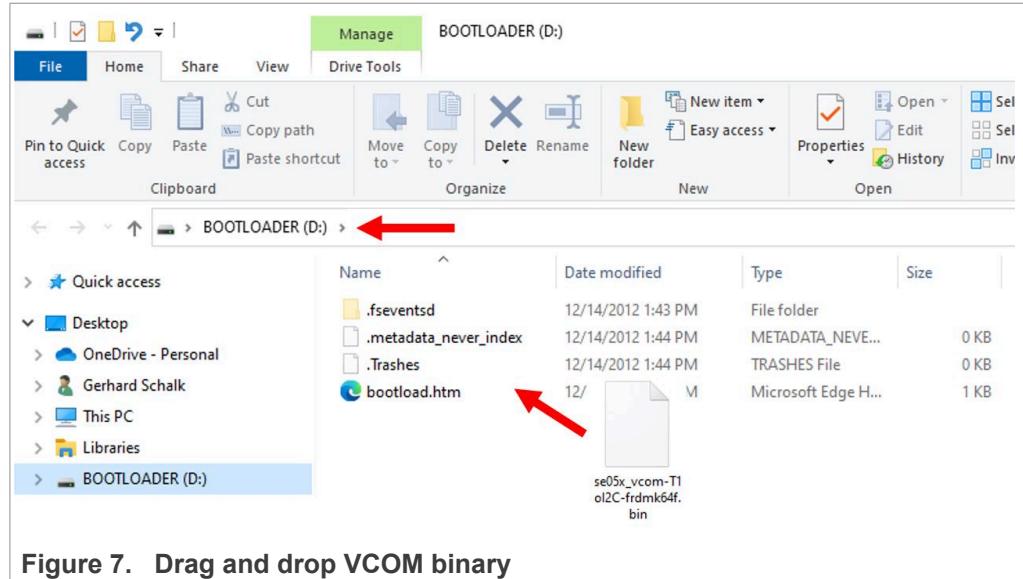


Figure 7. Drag and drop VCOM binary

5. The serial and VCOM ports should be recognized by your Device Manager. To check that the ports are recognized, follow the steps indicated in [Figure 8](#):
- Unplug the USB cable from the OpenSDA USB port.
  - Plug the USB cable to the OpenSDA USB port.
  - Check that the serial port is recognized in the category **Ports (COM & LPT)**. In this document, it is recognized as **USB Serial Device (COM7)** but this naming

## EdgeLock SE05x Quick start guide with Visual Studio project examples

might change depending on your computer. Therefore, it is important that you identify which device is recognized at the moment you plug the SDA USB port to the computer.

- d. Plug the USB cable to the K64F USB port.
- e. Check that the VCOM port is recognized in the category **Ports (COM & LPT)**. In this document, it is recognized as *Virtual Com Port (COM8)* but this naming might change depending on your computer (e.g. It could also appear named as *USB Serial Device*). Therefore, it is important that you identify which device is recognized at the moment you plug the K64F USB port to the computer.

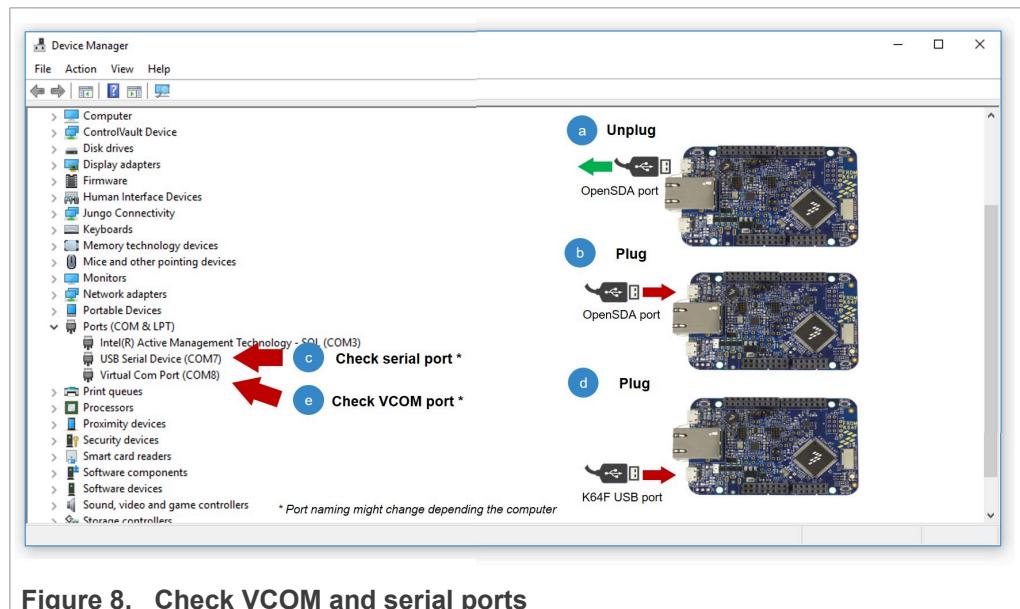


Figure 8. Check VCOM and serial ports

## 4 Run Plug & Trust middleware Visual Studio project examples

This section explains how to run Plug & Trust middleware Visual Studio project examples using the CMake-based build system.

### 4.1 Prerequisites

The following tools are required to run the Plug & Trust middleware Visual Studio projects:

1. Install Visual Studio  $\geq$  2017 version, or higher, in your laptop. For reference, [Section 7](#) illustrates how Visual Studio 2022 version can be installed, but the same procedure can be applied for more recent versions.
2. Install CMake. For reference, [Section 8](#) illustrates the detailed installation instructions.
3. Install Python Python  $\geq$  3.7.x and  $\leq$  3.9.x 32-bit version, in your laptop. For reference, [Section 9](#) illustrates how Python 3.7.x 32-bit version can be installed, but the same procedure can be applied for more recent versions.

### 4.2 Download Plug & Trust middleware

Follow these steps to download the Plug & Trust middleware in your local machine:

1. Download Plug & Trust middleware from the [NXP website](#).
2. Create a folder called **se05x\_mw** in C: directory as shown in [Figure 9](#):

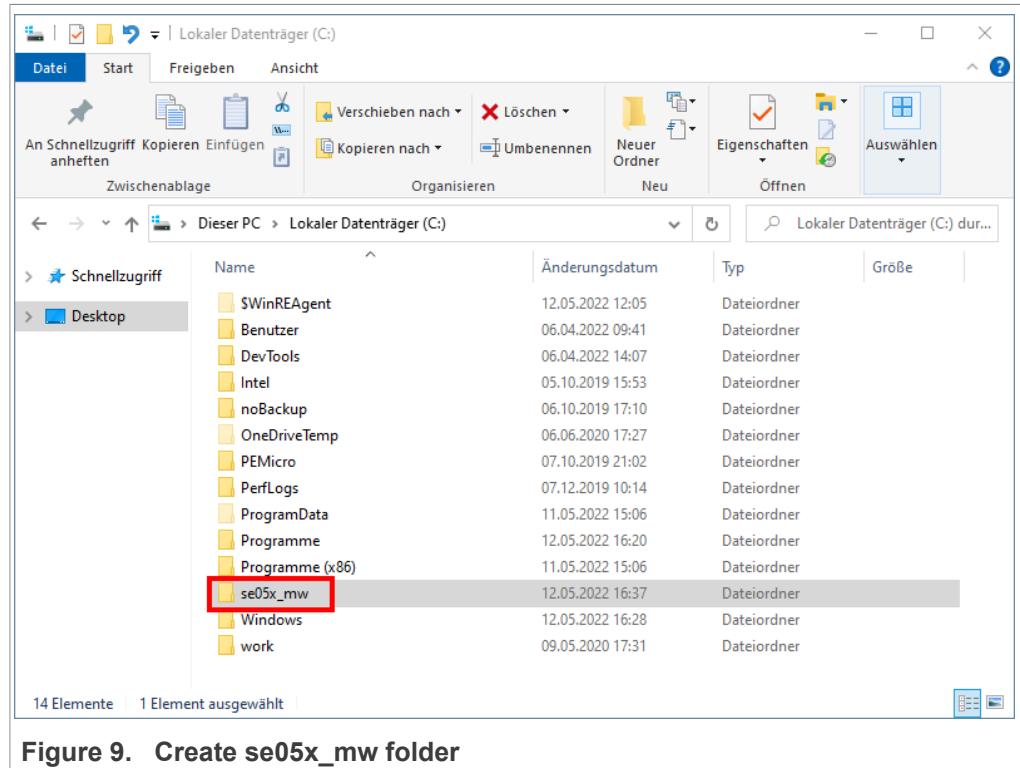


Figure 9. Create se05x\_mw folder

3. Unzip the Plug & Trust middleware inside the `se05x_mw` folder. After unzipping, you will see a folder called `simw-top` created. The contents of the `simw-top` directory should look as shown in [Figure 10](#):

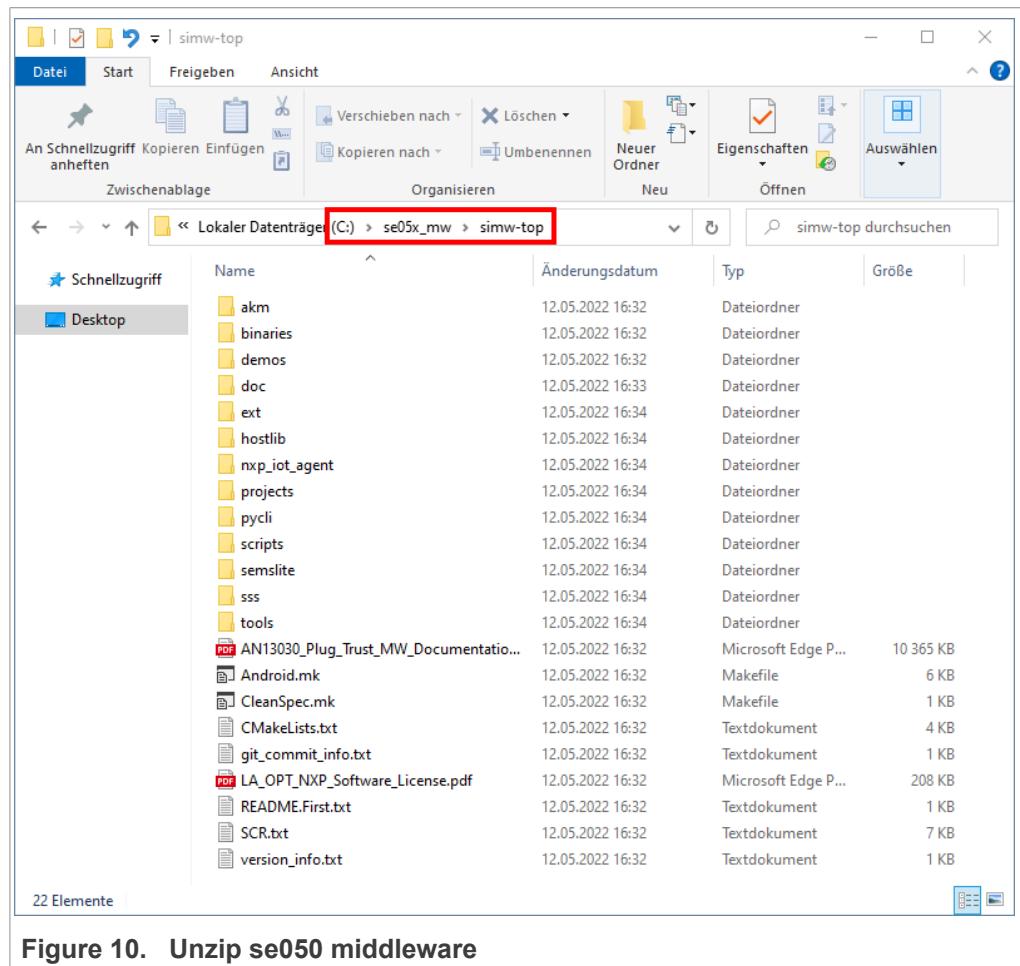


Figure 10. Unzip se050 middleware

**Note:** It is recommended to keep `se05x_mw` with the shortest path possible and without spaces in it. This avoids some issues that could appear when building the middleware if the path contains spaces.

#### 4.3 Build Plug & Trust middleware project examples

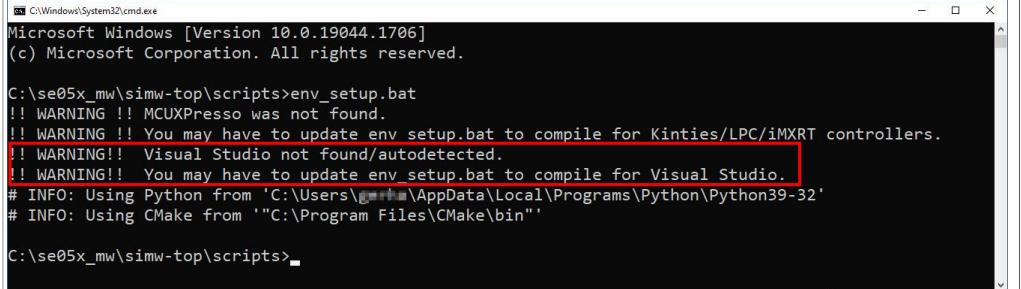
The Plug & Trust middleware uses CMake for building the project examples into your local machine. To build Plug & Trust middleware, open a Command Prompt and use the following steps as shown in [Figure 12](#):

1. Go to folder with the unzipped SE050 middleware:  
(1) Send >> cd C:\se05x\_mw\simw-top\scripts

## EdgeLock SE05x Quick start guide with Visual Studio project examples

## 2. Define the environment:

(2) Send &gt;&gt; env\_setup.bat



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

C:\se05x_mw\simmw-top\scripts>env_setup.bat
!! WARNING !! MCUXPresso was not found.
!! WARNING !! You may have to update env setup.bat to compile for Kinties/LPC/iMXRT controllers.
!! WARNING!! Visual Studio not found/autodetected.
!! WARNING!! You may have to update env setup.bat to compile for Visual Studio.
# INFO: Using Python from 'C:\Users\...\AppData\Local\Programs\Python\Python39-32'
# INFO: Using CMake from '"C:\Program Files\CMake\bin"'

C:\se05x_mw\simmw-top\scripts>
```

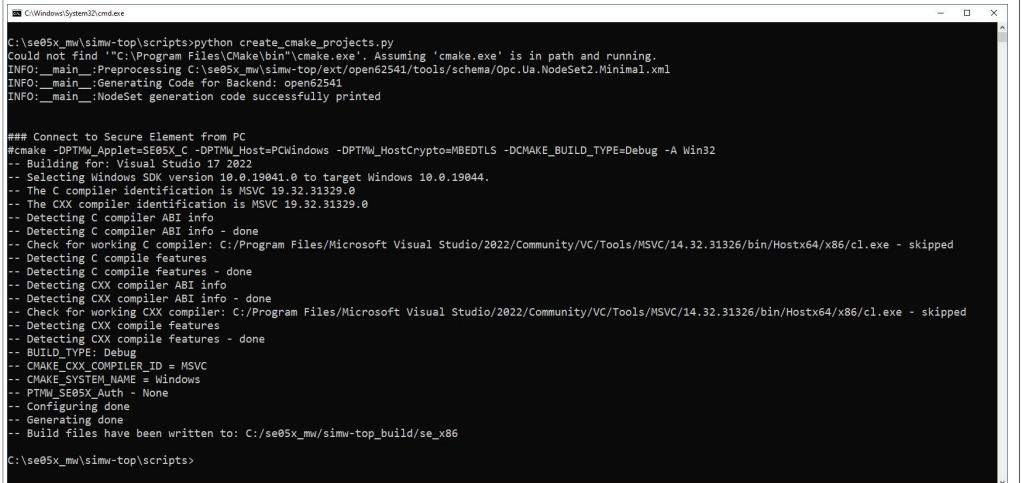
Figure 11. Generate Plug &amp; Trust middleware define the environment

Depending on your PC installation you may need to update the application file locations within the env\_setup.bat file.

## 3. Generate the Plug &amp; Trust middleware project examples:

(3) Send &gt;&gt; python create\_cmake\_projects.py

**Note:** This command may take a few seconds to complete.



```
C:\Windows\System32\cmd.exe
C:\se05x_mw\simmw-top\scripts>python create_cmake_projects.py
Could not find '":\Program Files\CMake\bin"\cmake.exe'. Assuming 'cmake.exe' is in path and running.
INFO: main :Preprocessing C:\se05x_mw\simmw-top\ext\open62541\tools\schema\Opc.Ua.NodeSet2.Minimal.xml
INFO: __main__ :Generating Code for Backend: open62541
INFO: __main__ :NodeSet generation code successfully printed

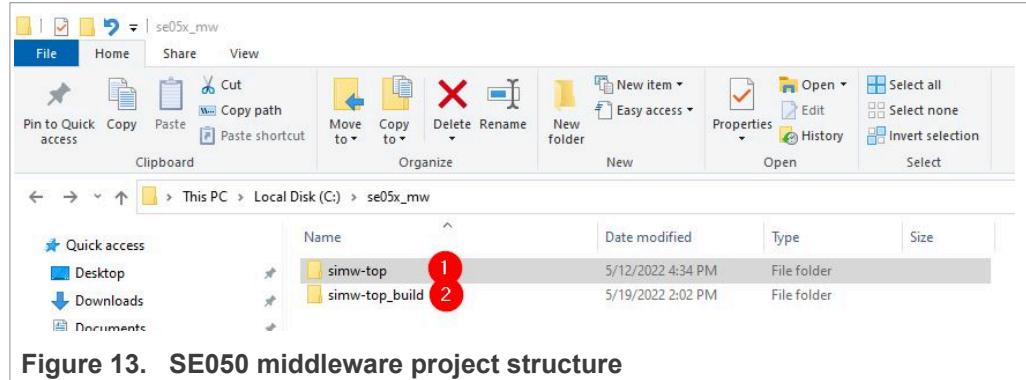
## Connect to Secure Element from PC
#cmake -DPTMW_Applet=SE05X_C -DPTMW_Host=PCWindows -DPTMW_HostCrypto=MBEDTLS -DCMAKE_BUILD_TYPE=Debug -A Win32
-- Building for: Visual Studio 17 2022
-- Selecting Windows SDK version 10.0.19041.0 to target Windows 10.0.19044.
-- The C compiler identification is MSVC 19.32.31329.0
-- The CXX compiler identification is MSVC 19.32.31329.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: C:/Program Files/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC/14.32.31326/bin/Hostx64/x86/cl.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: C:/Program Files/Microsoft Visual Studio/2022/Community/VC/Tools/MSVC/14.32.31326/bin/Hostx64/x86/cl.exe - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- BUILD_TYPE: Debug
-- CMAKE_CXX_COMPILER_ID = MSVC
-- CMAKE_SYSTEM_NAME = Windows
-- PTMW_SE05X.Auth - None
-- Configuring done
-- Generating done
-- Build files have been written to: C:/se05x_mw/simw-top_build/se_x86

C:\se05x_mw\simmw-top\scripts>
```

Figure 12. Generate Plug &amp; Trust middleware project examples

## EdgeLock SE05x Quick start guide with Visual Studio project examples

4. Your project directory should now contain two folders: a (1) `simw-top` folder and a (2) `simw-top_build` folder as shown in [Figure 13](#):



**Figure 13.** SE050 middleware project structure

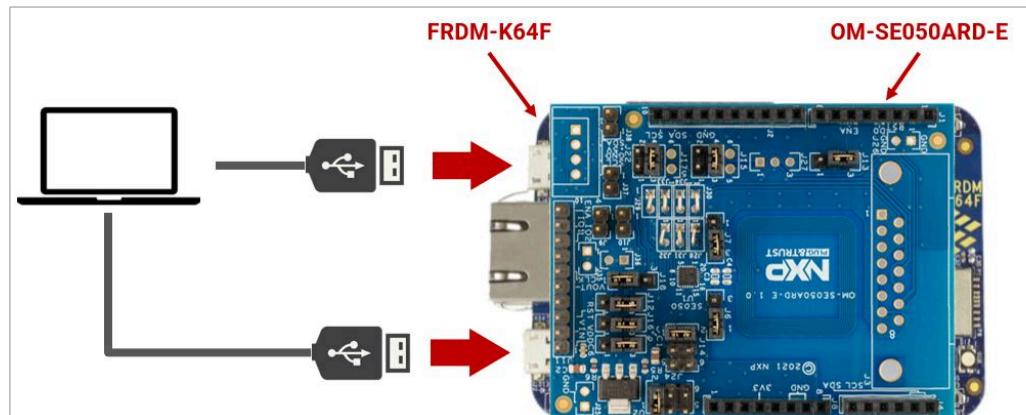
#### 4.4 Execute EdgeLock SE05x Visual Studio project examples

This section explains how to run the Plug & Trust middleware project example called `se05x_minimal`. The `se05x_minimal` project outputs the memory left in EdgeLock SE05x security IC.

**Note:** The execution of the `se05x_minimal` project is shown as an example. The steps detailed in this section can be replicated to run any other project example included as part of the Plug & Trust middleware.

To execute the `se05x_minimal` test example, follow these steps:

1. Connect the FRDM-K64F board to your laptop as shown in [Figure 14](#). Check that your Windows Device Manager recognizes FRDM-K64F board as shown in [Figure 8](#). Refer to [Figure 1](#) for OM-SE050ARD jumper configuration.



**Figure 14.** Plug OpenSDA and USB FRDM-K64F ports.

## EdgeLock SE05x Quick start guide with Visual Studio project examples

2. Open the CMake configuration menu as shown in [Figure 15](#) and ensure that all the flags are set properly for your use case.
  - a. Open a command prompt and go to the directory where the Plug & Trust middleware is built.  
Send: cd C:\se05x\_mw\simw-top\_build\se\_x86
  - b. Open the cmake configuration interface.  
Send: cmake-gui .

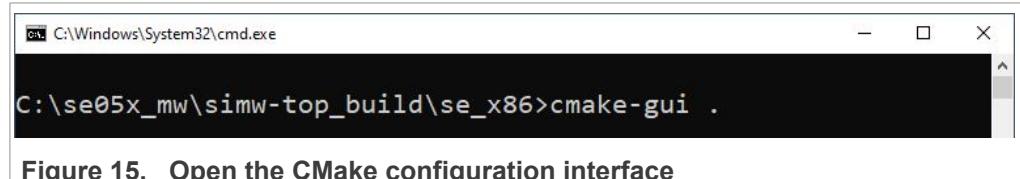


Figure 15. Open the CMake configuration interface

This step allows the user to customize the compilation options. The default build configuration of the EdgeLock SE05x Plug & Trust middleware  $\geq$  v04.02.0x generates code for the OM-SE050ARD-E development board. You need to adapt the CMake settings in case you are using a different EdgeLock secure element

## EdgeLock SE05x Quick start guide with Visual Studio project examples

development board or a different secure element product IC. The settings are described in [Section 5](#). The default SE050E settings are shown in [Figure 16](#). In this example we use plain communication. Plain communication for the example execution is enabled by selecting the following options:

- Select `None` for the CMake option `PTMW_SE05X_Auth`.
- Select `None` for the CMake option `PTMW_SCP`.

How to enable Platform SCP is described in [How to enable Platform SCP](#).

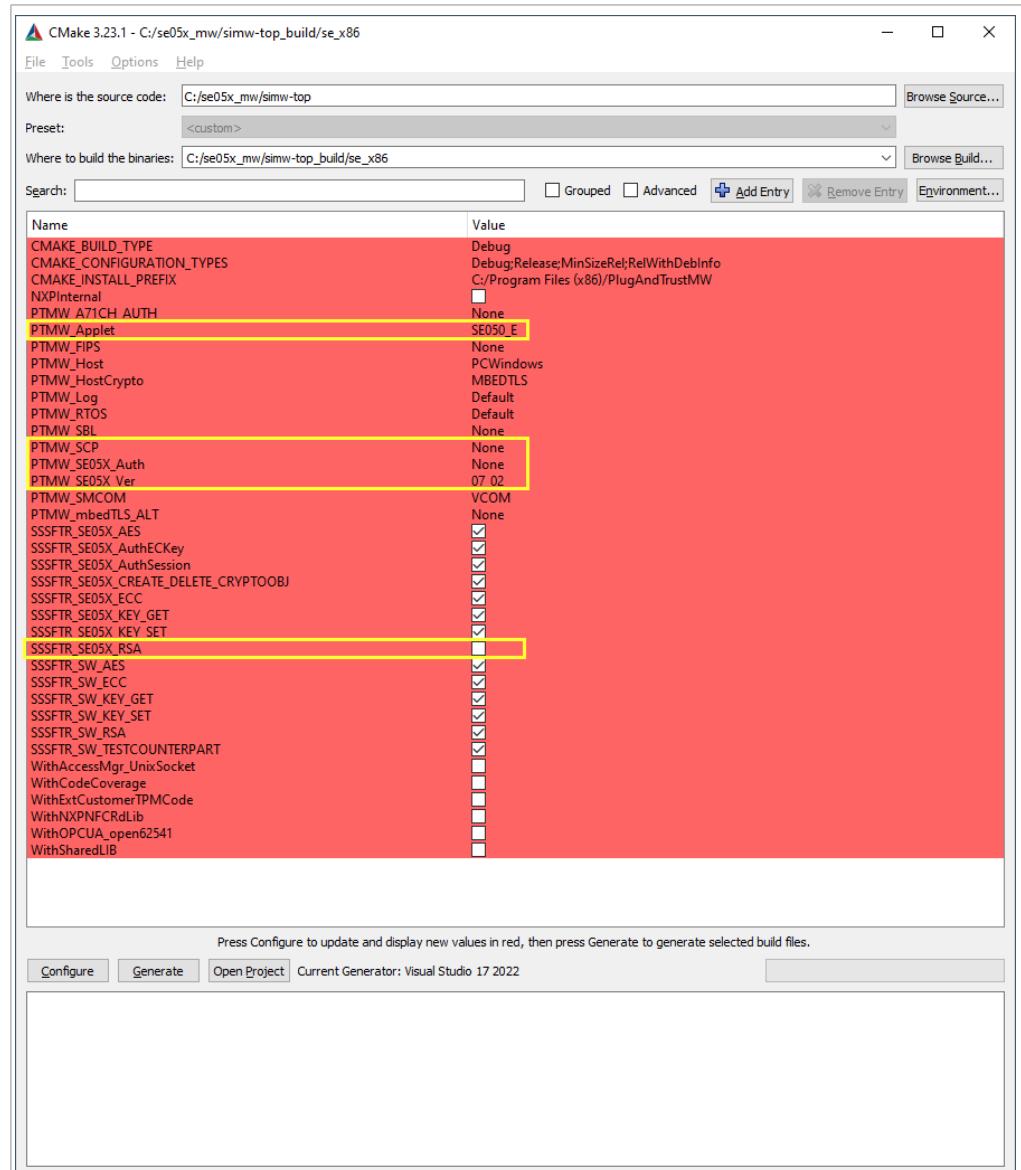


Figure 16. Review CMake configuration

If you have edited any of the parameters in the menu, before exiting press the buttons **Configure** and **Generate** to apply the changes.

For more information about the CMake options please refer the CMake section of Plug & Trust middleware documentation: `simw-top/doc/scripts/cmake_options.html`

## EdgeLock SE05x Quick start guide with Visual Studio project examples

3. Go to the C:\se05x\_mw\simw-top\_build\se\_x86 directory. Double click the **PlugAndTrustMW.sln** Visual Studio project solution as shown in [Figure 17](#):

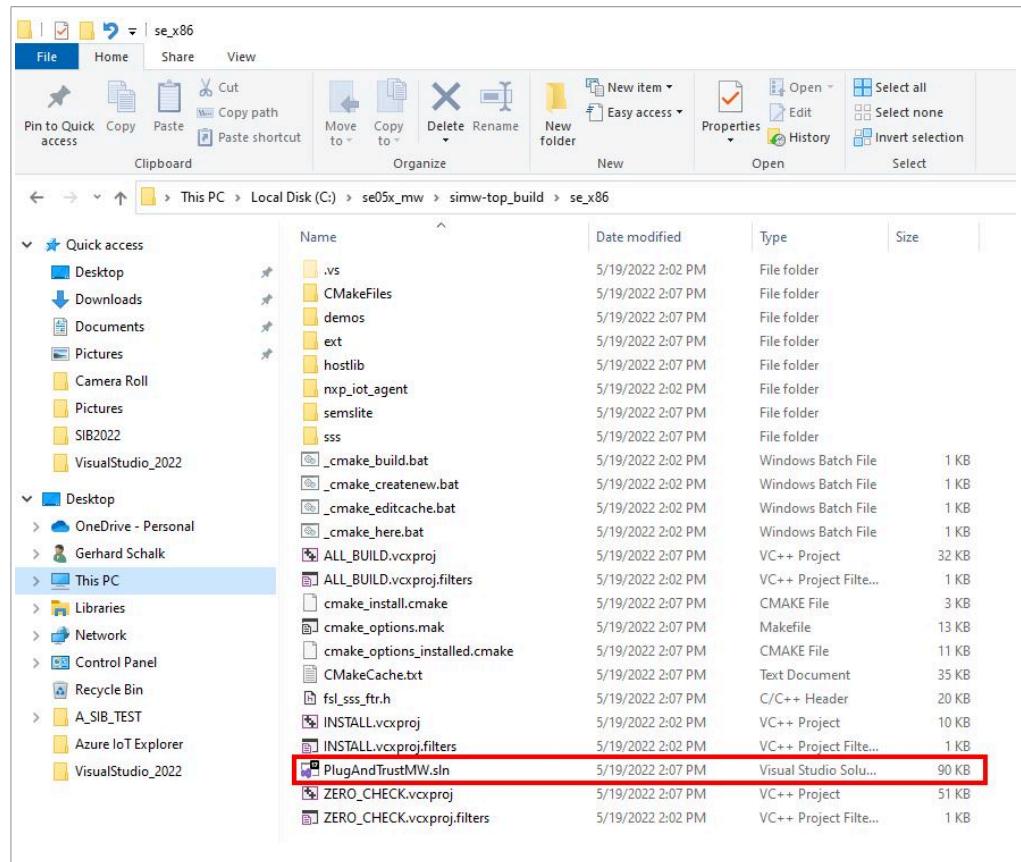
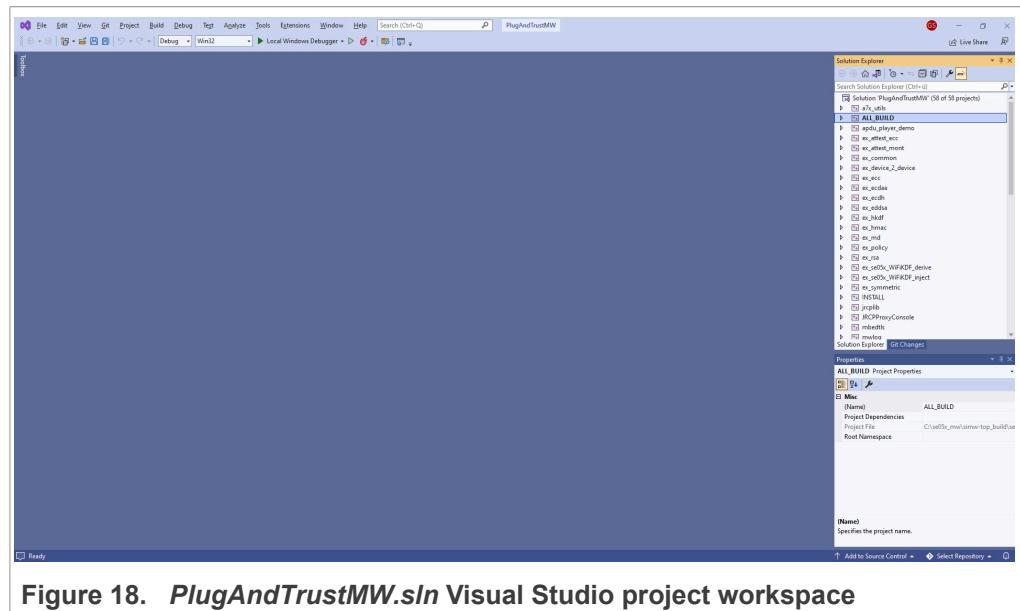


Figure 17. Open **PlugAndTrustMW.sln** Visual Studio project solution

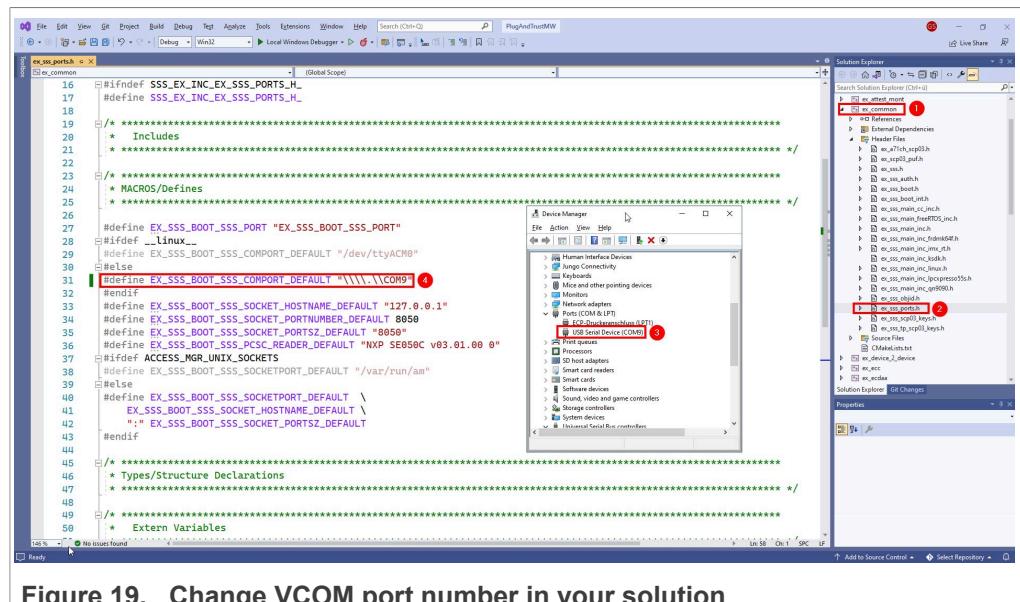
## EdgeLock SE05x Quick start guide with Visual Studio project examples

4. The Visual Studio IDE will open with the Plug & Trust middleware project examples included in the workspace as can be seen in [Figure 18](#):



**Figure 18.** *PlugAndTrustMW.sln* Visual Studio project workspace

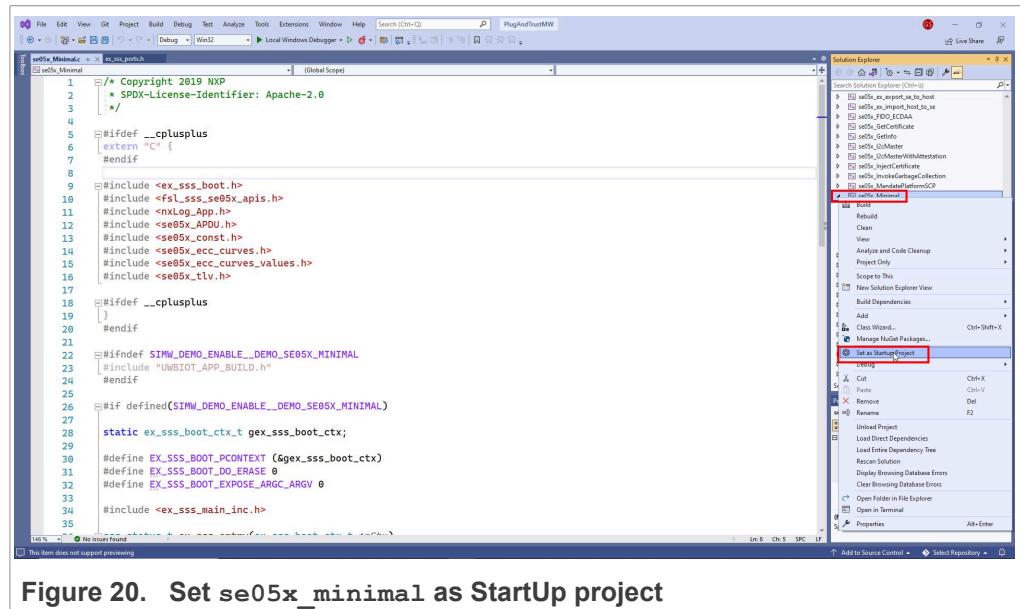
5. Change the VCOM port number in the Plug & Trust middleware project. To do so, follow the instructions shown in [Figure 19](#):
- Go to the `ex_common` project and open the `ex_sss_ports.h` file inside the `headers` directory.
  - Change `#define EX_SSS_BOOT_SSS_COMPORT_DEFAULT "\\\\.\\"COMx"` with the port COM number your laptop assigned to your FRDM-K64F. In this setup, the COM number is COM9.



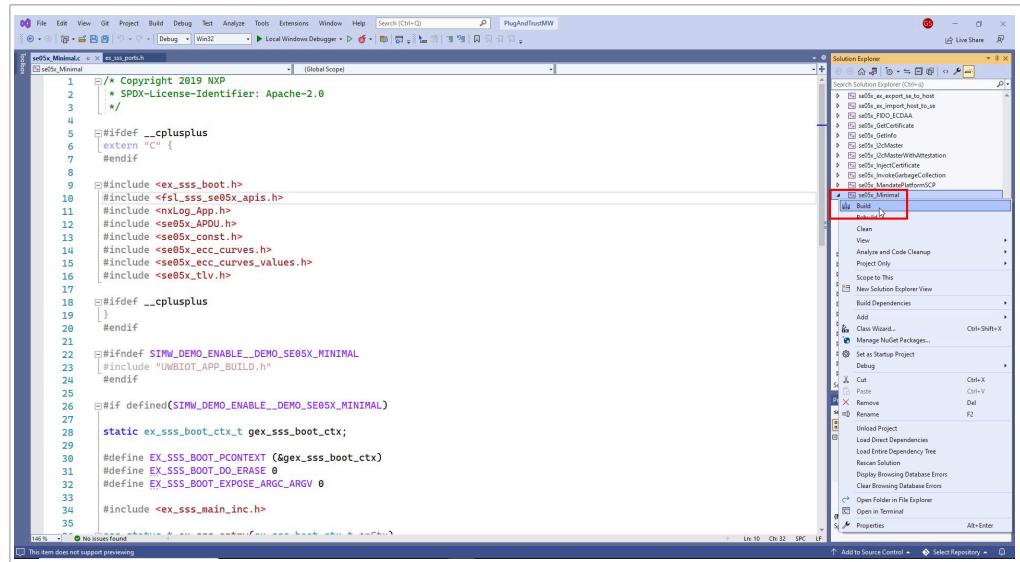
**Figure 19.** Change VCOM port number in your solution

## EdgeLock SE05x Quick start guide with Visual Studio project examples

6. Select the se05x\_minimal project from the Solution Explorer window located on the right-hand side of the Visual Studio IDE. Do right-click on the project and click on **Set as StartUp project** as shown in [Figure 20](#):

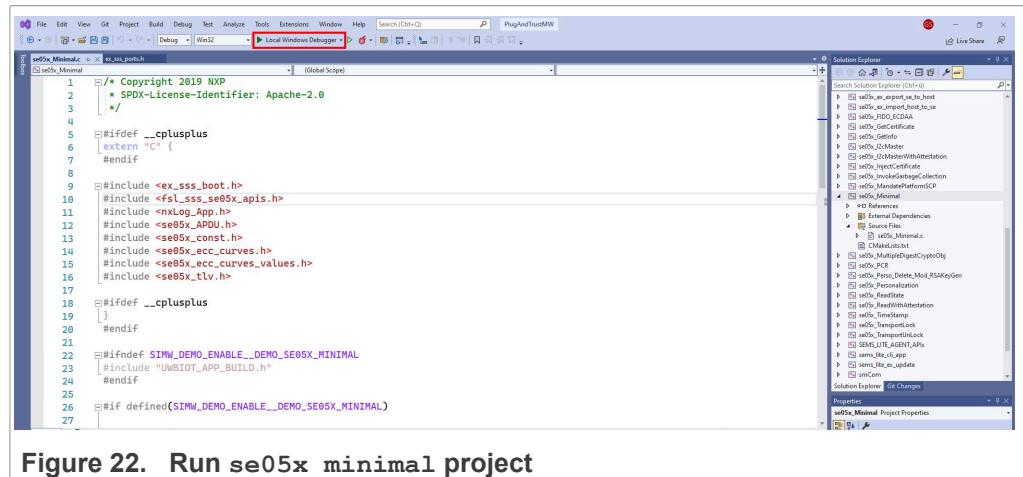
**Figure 20.** Set se05x\_minimal as StartUp project

7. Right click on the se05x\_minimal project and build it by clicking the build option as shown in [Figure 21](#).

**Figure 21.** Build se05x\_minimal project

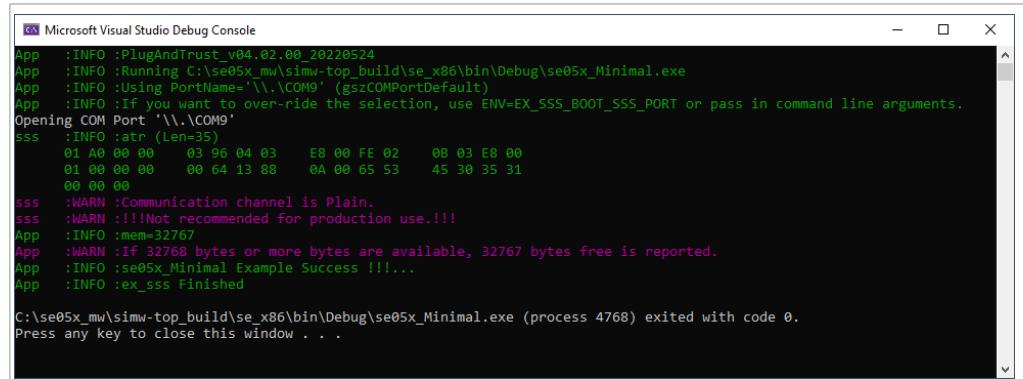
## EdgeLock SE05x Quick start guide with Visual Studio project examples

8. Click on **Local Windows Debugger** button in the top menu to run se05x\_minimal project as shown in [Figure 22](#). The project will be executed after the project building process has finished.



**Figure 22. Run se05x\_minimal project**

9. If the se05x\_minimal project runs successfully, a Console window will be opened. The logs in this Console window indicate the available memory in EdgeLock SE05x security IC (in this case, 592) as can be seen in [Figure 23](#):



**Figure 23. Verify that se05x\_minimal project is running**

10. The same operation can be repeated with any of the other Plug & Trust middleware project examples.

## 5 Product specific CMake build settings

The NXP Plug & Trust middleware supports the SE05x Secure Elements, the A5000 Secure Authenticator, and the legacy A71CH products.

The EdgeLock Plug & Trust middleware is delivered with CMake files that include the set of directives and instructions describing the project's source files and the build targets. The CMake files are used to select a dedicated EdgeLock product IC and the corresponding IoT applet or Authenticator application.

The SE050 product identification can be obtained as described in [AN12436 chapter 1 Product Information](#). [AN12973](#) describes the same procedure for the SE051 product family.

The following tables show the required `PTMW` CMake options to build a dedicated product variant. The `SSSFTR_SE05X_RSA` CMake option is used to optimize the memory footprint for product variants that do not support RSA.

**Table 3. CMake Settings for SE050E product variants**

Variant	OEF ID	PTMW_Applet	PTMW_FIPS	PTMW_SE05X_Ver	PTMW_SE05X_Auth	PTMW_SC	SSSFTR_SE05X_RSA
SE050E Dev. Board OM-SE050ARD-E	A921	SE050_E	None	07_02	any option	None or SCP03_SSS	disabled
SE050E2	A921						

**Table 4. CMake Settings for SE050F product variants**

Variant	OEF ID	PTMW_Applet	PTMW_FIPS	PTMW_SE05X_Ver	PTMW_SE05X_Auth	PTMW_SC	SSSFTR_SE05X_RSA
SE050F Dev.Board OM-SE050ARD-F	A92A	SE05X_C	SE050	03_XX	PlatfSCP03 or UserID_PlatformSCP03 or AESKey_PlatformSCP03 or ECKey_PlatformSCP03	SCP03_SSS	enabled
SE050F2	A92A						

**Table 5. CMake Settings for SE050 Previous Generation product variants**

Variant	OEF ID	PTMW_Applet	PTMW_FIPS	PTMW_SE05X_Ver	PTMW_SE05X_Auth	PTMW_SC	SSSFTR_SE05X_RSA
SE050A1	A204	SE05X_A	None	03_XX	any option	None or SCP03_SSS	disabled
SE050A2	A205						
SE050B1	A202	SE05X_B	None	03_XX	any option	None or SCP03_SSS	enabled
SE050B2	A203						

**Table 5. CMake Settings for SE050 Previous Generation product variants...continued**

Variant	OEF ID	PTMW_Applet	PTMW_FIPS	PTMW_SE05X_Ver	PTMW_SE05X_Auth	PTMW_SC	SSSFTR_SE05X_RSA
SE050C1	A200	SE05X_C	None	03_XX	any option	None or SCP03_SSS	enabled
SE050C2	A201						
SE050 Dev Board OM-SE050ARD	A1F4						
SE050F2	A77E <sup>[1]</sup>	SE05X_C	SE050	03_XX	PlatfSCP03 or UserID_PlatformSCP03 or AESKey_PlatformSCP03 or ECKey_PlatformSCP03	SCP03_SSS	enabled

[1] All SE050F2 with variant A77E have date code in year 2021. All the SE050F2 with date code in the year 2022 have the variant identifier A92A.

**Table 6. CMake Settings for SE051 product variants**

Variant	OEF ID	PTMW_Applet	PTMW_FIPS	PTMW_SE05X_Ver	PTMW_SE05X_Auth	PTMW_SC	SSSFTR_SE05X_RSA
SE051A2	A920	SE05X_A	None	07_02	any option	None or SCP03_SSS	disabled
SE051C2	A8FA	SE05X_C	None	07_02	any option	None or SCP03_SSS	enabled
SE051W2	A739	SE05X_C	None	07_02	any option	None or SCP03_SSS or SCP03_SSS	enabled
SE051A2	A565	SE05X_A	None	06_00	any option	None or SCP03_SSS	disabled
SE051C2	A564	SE05X_C	None	06_00	any option	None or SCP03_SSS	enabled

**Table 7. CMake Settings for A5000 product variants**

Variant	OEF ID	PTMW_Applet	PTMW_FIPS	PTMW_SE05X_Ver	PTMW_SE05X_Auth	PTMW_SCN	SSSFTR_SE05X_RSA
OM-A5000ARD	A736	AUTH	None	07_02	any option	None or SCP03_SSS	disabled
A5000	A736						

## 5.1 Example: SE050E CMake build settings

The following images show the configuration for the SE050E development board OM-SE05ARD-E.

- Select `SE05X_E` for the CMake option `PTWM_Applet`.
- Select `None` for the CMake option `PTWM_FIPS`.
- Select `07_02` for the CMake option `PTWM_SE05X_Ver`.
- Disable the CMake option `SSSFTR_SE05X_RSA`.

In this example we use plain communication. Plain communication for the example execution is enabled by selecting the following options:

- Select `None` for the CMake option `PTMW_SE05X_Auth`.
- Select `None` for the CMake option `PTMW_SCN`.

How to enable Platform SCP is described in [How to enable Platform SCP](#).

## EdgeLock SE05x Quick start guide with Visual Studio project examples

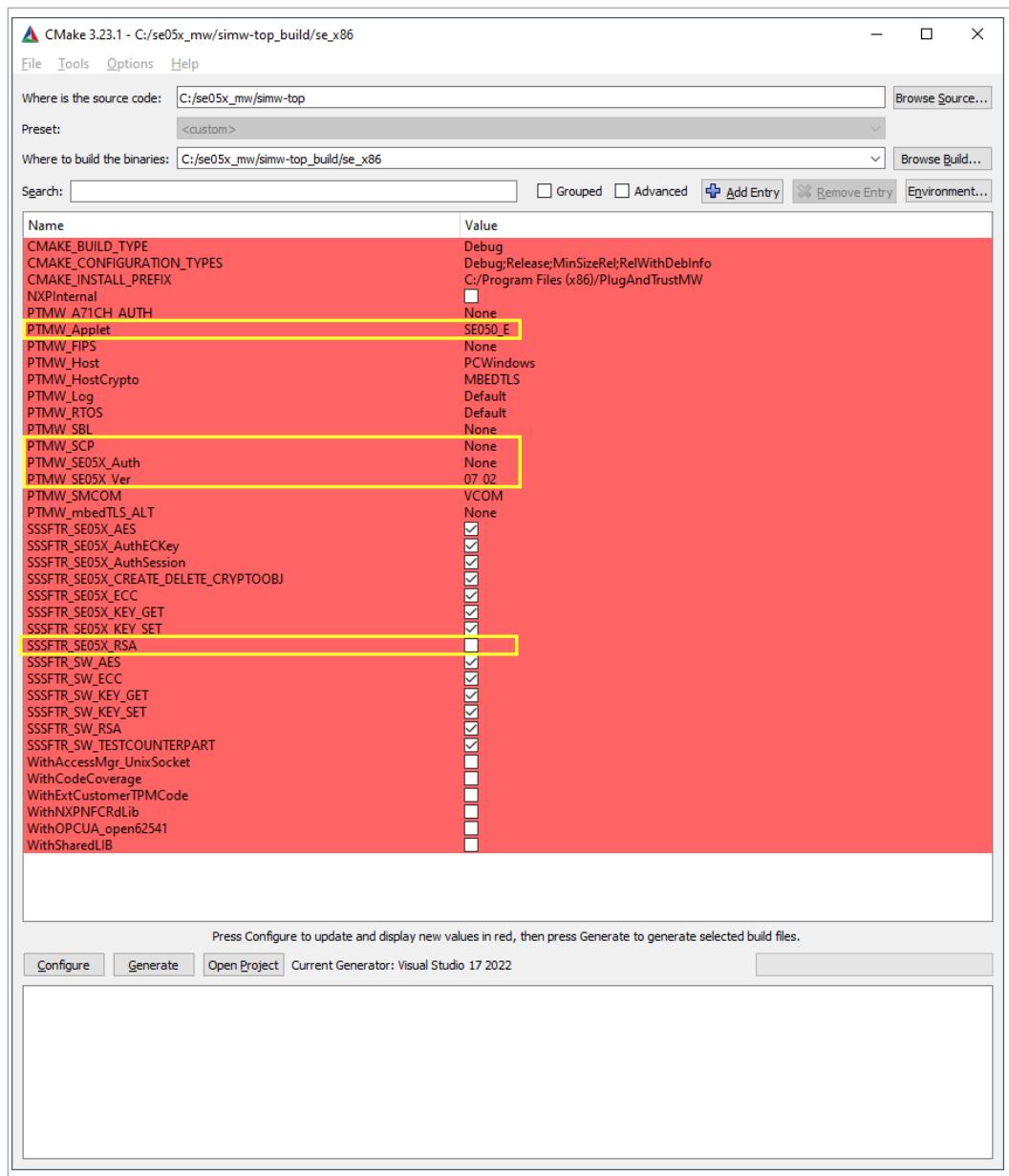


Figure 24. SE050E CMake Settings - Plain communication

## 6 Binding EdgeLock SE05x to a host using Platform SCP

Binding is a process to establish a pairing between the IoT device host MPU/MCU and EdgeLock SE05x, so that only the paired MPU/MCU is able to use the services offered by the corresponding EdgeLock SE05x and vice versa.

A mutually authenticated, encrypted channel will ensure that both parties are indeed communicating with the intended recipients and that local communication is protected against local attacks, including man-in-the-middle attacks aimed at intercepting the communication between the MPU/MCU and the EdgeLock SE05x and physical tampering attacks aimed at replacing the host MPU/MCU or EdgeLock SE05x .

EdgeLock SE05x natively supports Global Platform Secure Channel Protocol 03 (SCP03) for this purpose. PlatformSCP uses SCP03 and can be enabled to be mandatory.

This chapter describes the required steps to enable Platform SCP in the middleware for EdgeLock SE05x.

The following topics are discussed:

- [Section 6.1 Introduction to the Global Platform Secure Channel Protocol 03 \(SCP03\)](#)
- [Section 6.2 How to configure the Platform SCP keys](#)
- [Section 6.3 How to enable Platform SCP](#)

### 6.1 Introduction to the Global Platform Secure Channel Protocol 03 (SCP03)

The Secure Channel Protocol SCP03 authenticates and protects locally the bidirectional communication between host and EdgeLock SE05x against eavesdropping on the physical I2C interface.

EdgeLock SE05x can be bound to the host by injecting in both the host and EdgeLock SE05x the same unique SCP03 AES key-set and by enabling the Platform SCP feature in the Plug & Trust middleware. The [AN12662 Binding a host device to EdgeLock SE05x](#) describes in detail the concept of secure binding.

SCP03 is defined in [Global Platform Secure Channel Protocol '03' - Amendment D v1.2](#) specification.

SCP03 can provide the following three security goals:

- **Mutual authentication (MA)**
  - Mutual authentication is achieved through the process of initiating a Secure Channel and provides assurance to both the host and the EdgeLock SE05x entity that they are communicating with an authenticated entity.
- **Message Integrity**
  - The Command- and Response-MAC are generated by applying the CMAC according NIST SP 800-38B.
- **Confidentiality**
  - The message data field is encrypted across the entire data field of the command message to be transmitted to the EdgeLock SE05x, and across the response transmitted from the EdgeLock SE05x.

The SCP03 secure channel is set up via the EdgeLock SE05x Java Card OS Manager using the standard ISO7816-4 secure channel APDUs.

## EdgeLock SE05x Quick start guide with Visual Studio project examples

The establishment of an SCP03 channel requires three static 128-bit AES keys shared between the two communicating parties: Key-ENC, Key-MAC and Key-DEK. These keys are stored in the Java Card Secondary Security Domain (SSD) and not in the secure authenticator applet.

Key-ENC and Key-MAC keys are used during the SCP03 channel establishment to generate the session keys. Session Keys are generated to ensure that a different set of keys are used for each Secure Channel Session to prevent replay attacks.

Key-ENC is used to derive the session key S-ENC. The S-ENC key is used for encryption/decryption of the exchanged data. The session keys S-MAC and R-MAC are derived from Key-MAC and used to generate/verify the integrity of the exchanged data (C-APDU and R-APDU).

Key-DEK key is used to encrypt new SCP03 keys in case they get updated.

**Table 8. Static SCP03 keys**

Key	Description	Usage	Key Type
Key-ENC	Static Secure Channel Encryption Key	Generate session key for Decryption/Encryption (AES)	AES 128
Key-MAC	Static Secure Channel Message Authentication Code Key	Generate session key for Secure Channel authentication and Secure Channel MAC Verification/Generation (AES)	AES 128
Key-DEK	Data Encryption Key	Sensitive Data Decryption (AES)	AES 128

The session key generation is performed by the Plug & Trust middleware host crypto.

**Table 9. SCP03 session keys**

Key	Description	Usage	Key Type
S-ENC	Session Secure Channel Encryption Key	Used for data confidentiality	AES 128
S-MAC	Secure Channel Message Authentication Code Key for Command	Used for data and protocol integrity	AES 128
S-RMAC	Secure Channel Message Authentication Code Key for Response	User for data and protocol integrity	AES 128

**Note:** For further details please refer to [Global Platform Secure Channel Protocol '03' - Amendment D v1.2](#).

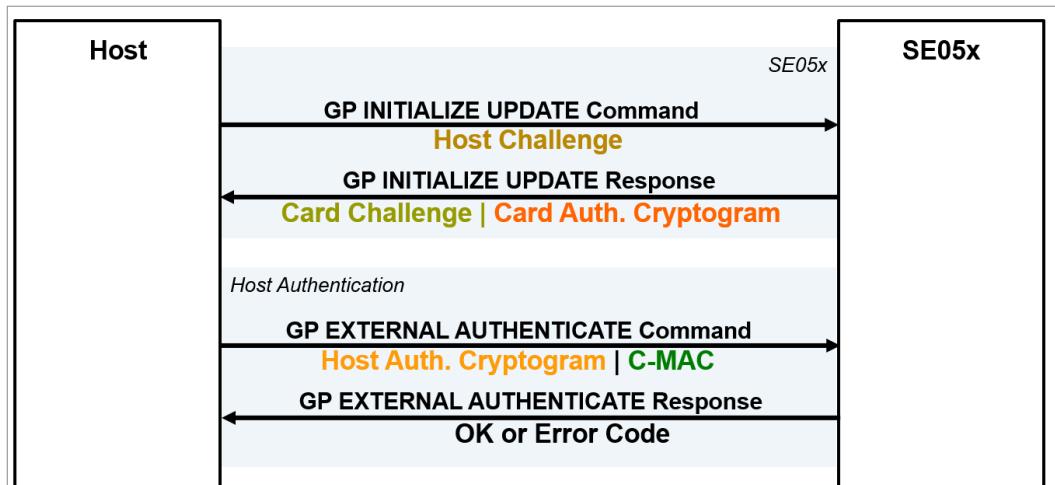
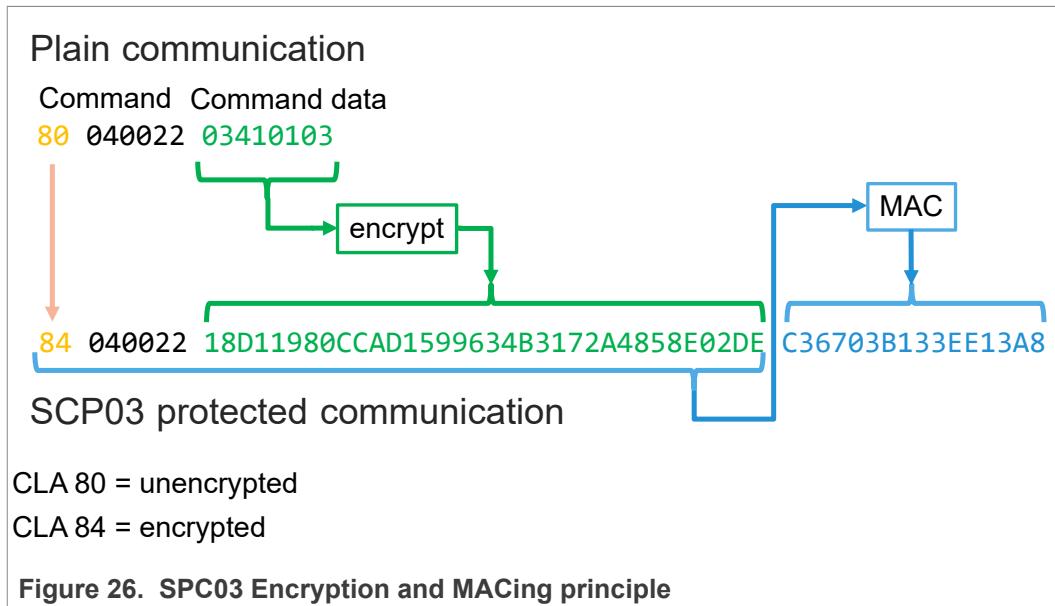


Figure 25. SPC03 mutual authentication – principle



## 6.2 How to configure the product specific default Platform SCP keys

The default Platform SCP key values are described for the EdgeLock SE05x product variants in [AN12436](#) and for the EdgeLock SE051 variants in [AN12973](#).

The Platform SCP keys can be defined in the Plug & Trust middleware source code.

The Plug & Trust middleware header file `ex_sss_tp_scp03_keys.h` contains the default values of all EdgeLock SE05x, EdgeLock SE051, A5000 and A71CH product variants.

The `ex_sss_tp_scp03_keys.h` header file can be found in the following location: C:\\se05x\_mw\\simw-top\\sss\\ex\\inc

## EdgeLock SE05x Quick start guide with Visual Studio project examples

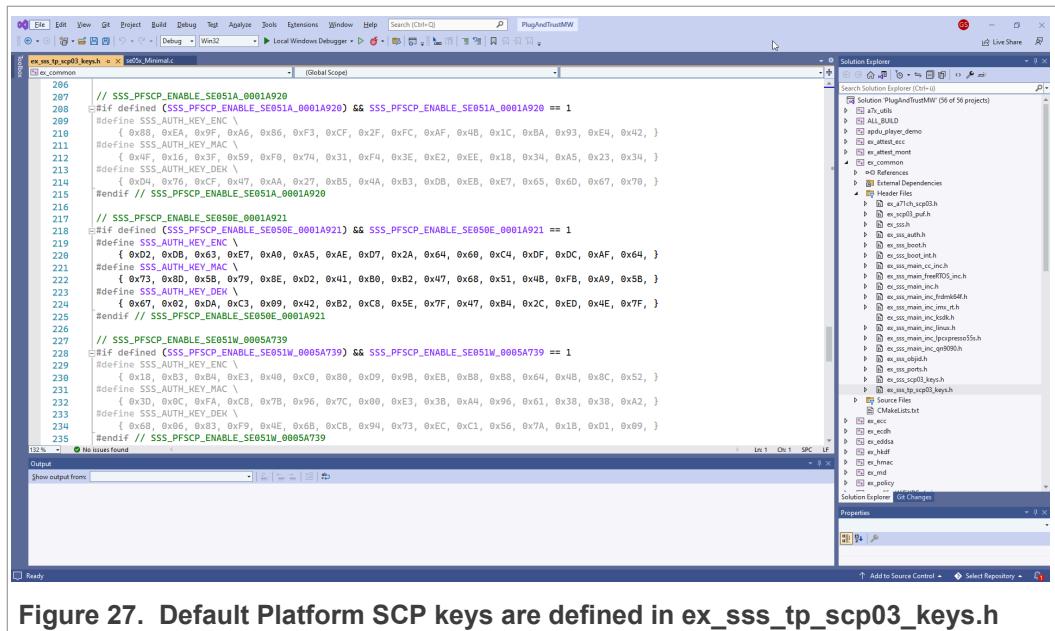


Figure 27. Default Platform SCP keys are defined in ex\_sss\_tp\_scp03\_keys.h

The fsl\_sss\_ftr.h.in file includes options to select one of the predefined default Platform SCP keys. This file is located in: C:\se05x\_mw\simw-top\sss\inc

Select the desired value of the compilation option by setting exclusively the corresponding C-preprocessor define SSS\_PFSCP\_ENABLE\_xx to 1 (enable). All other values for the same option (represented by C-preprocessor defines SSS\_PFSCP\_ENABLE\_xx) must be set to 0.

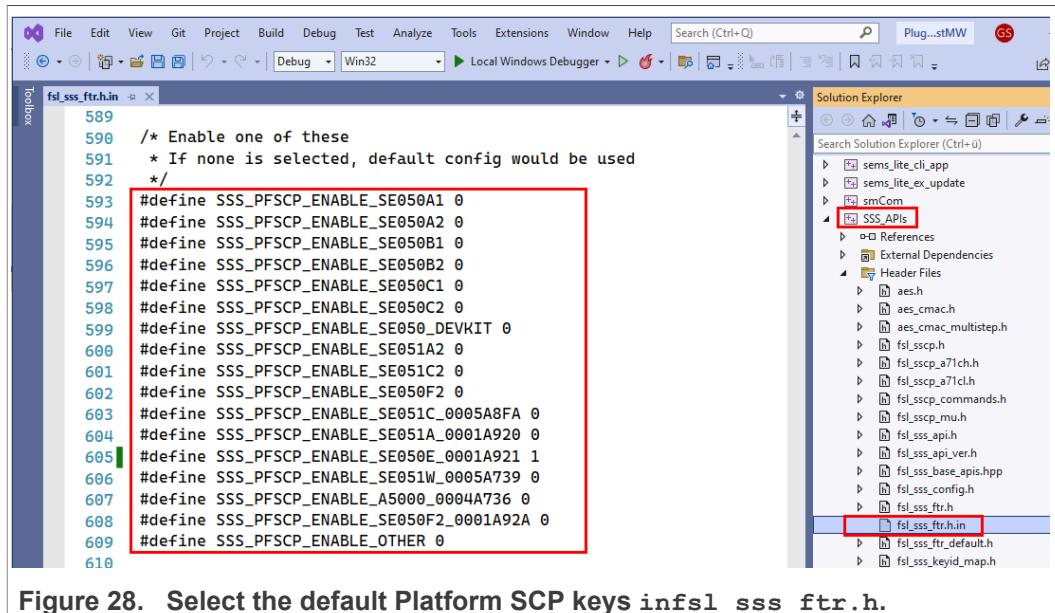


Figure 28. Select the default Platform SCP keys in fsl\_sss\_ftr.h.in

The Plug & Trust Middleware uses a feature file to select/detect used/enabled features within the middleware stack. The file fsl\_sss\_ftr.h is automatically generated into the used build directory. CMake is overwriting the fsl\_sss\_ftr.h file every time CMake is invoked. CMake is using the SCP key settings of the fsl\_sss\_ftr.h.in file as input to generate the the fsl\_sss\_ftr.h file. You do not have to manually edit

## EdgeLock SE05x Quick start guide with Visual Studio project examples

the `fsl_sss_ftr.h` feature file. Selections from CMake edit cache would automatically make relevant updates into the generated feature file.

**Note:** The Platform SCP key selection in the `fsl_sss_ftr.h.in` CMake input file is persistent.

The location of the generated `fsl_sss_ftr.h` feature header file is: `C:\se05x_mw\simw-top_build\se_x86`

The following tables contains the the Platform SCP key header file define to be set to 1 (enable) for the different secure element and secure authenticator product variants.

**Table 10. Platform SCP key define prefix for SE050E product variants**

Variant	OEF ID	Platform SCP key define to be set to '1'
SE050E Dev. Board OM-SE050ARD-E	A921	<code>SSS_PFSCP_ENABLE_SE050E_0001A921</code>
SE050E2	A921	<code>SSS_PFSCP_ENABLE_SE050E_0001A921</code>

**Table 11. Platform SCP key define prefix for SE050F product variants**

Variant	OEF ID	Platform SCP key define to be set to '1'
SE050F Dev. Board OM-SE050ARD-F	A92A	<code>SSS_PFSCP_ENABLE_SE050F2_0001A92A</code>
SE050F2	A92A	<code>SSS_PFSCP_ENABLE_SE050F2_0001A92A</code>

**Table 12. Platform SCP key define prefix for SE050 Previous Generation product variants**

Variant	OEF ID	Platform SCP key define to be set to '1'
SE050A1	A204	<code>SSS_PFSCP_ENABLE_SE050A1</code>
SE050A2	A205	<code>SSS_PFSCP_ENABLE_SE050A2</code>
SE050B1	A202	<code>SSS_PFSCP_ENABLE_SE050B1</code>
SE050B2	A203	<code>SSS_PFSCP_ENABLE_SE050B2</code>
SE050C1	A200	<code>SSS_PFSCP_ENABLE_SE050C1</code>
SE050C2	A201	<code>SSS_PFSCP_ENABLE_SE050C2</code>
SE050 Dev Board OM-SE050ARD	A1F4	<code>SSS_PFSCP_ENABLE_SE050_DEVKIT</code>
SE050F2	A77E <sup>[1]</sup>	<code>SSS_PFSCP_ENABLE_SE050F2</code>

[1] All SE050F2 with variant A77E have date code in year 2021. All the SE050F2 with date code in the year 2022 have the variant identifier A92A.

**Table 13. Platform SCP key define prefix for SE051 product variants**

Variant	OEF ID	Platform SCP key define to be set to '1'
SE051A2	A920	<code>SSS_PFSCP_ENABLE_SE051A_0001A920</code>
SE051C2	A8FA	<code>SSS_PFSCP_ENABLE_SE051C_0005A8FA</code>
SE051W2	A739	<code>SSS_PFSCP_ENABLE_SE051W_0005A739</code>
SE051A2	A565	<code>SSS_PFSCP_ENABLE_SE051A2</code>
SE051C2	A564	<code>SSS_PFSCP_ENABLE_SE051C2</code>

**Table 14. Platform SCP key define prefix for A5000 product variants**

Variant	OEF ID	Platform SCP key define to be set to '1'
A5000 Dev. Board OM-A5000ARD	A736	SSS_PFSCP_ENABLE_A5000_0004A736
A5000	A736	SSS_PFSCP_ENABLE_A5000_0004A736

### 6.3 How to enable Platform SCP

To enable Platform SCP is required to rebuild the SDK with the following CMake options:

- Select SCP03\_SSS for the CMake option PTMW\_SCP.
- Select PlatfSCP03 for the CMake option PTMW\_SE05X\_Auth.

The following images show the configuration for the SE050E development board OM-SE05ARD-E.

1. Open a command prompt and go to the directory where the Plug & Trust middleware is built.  
Send: cd C:\se05x\_mw\simw-top\_build\se\_x86
2. Open the cmake configuration interface.  
Send: cmake-gui .

## EdgeLock SE05x Quick start guide with Visual Studio project examples

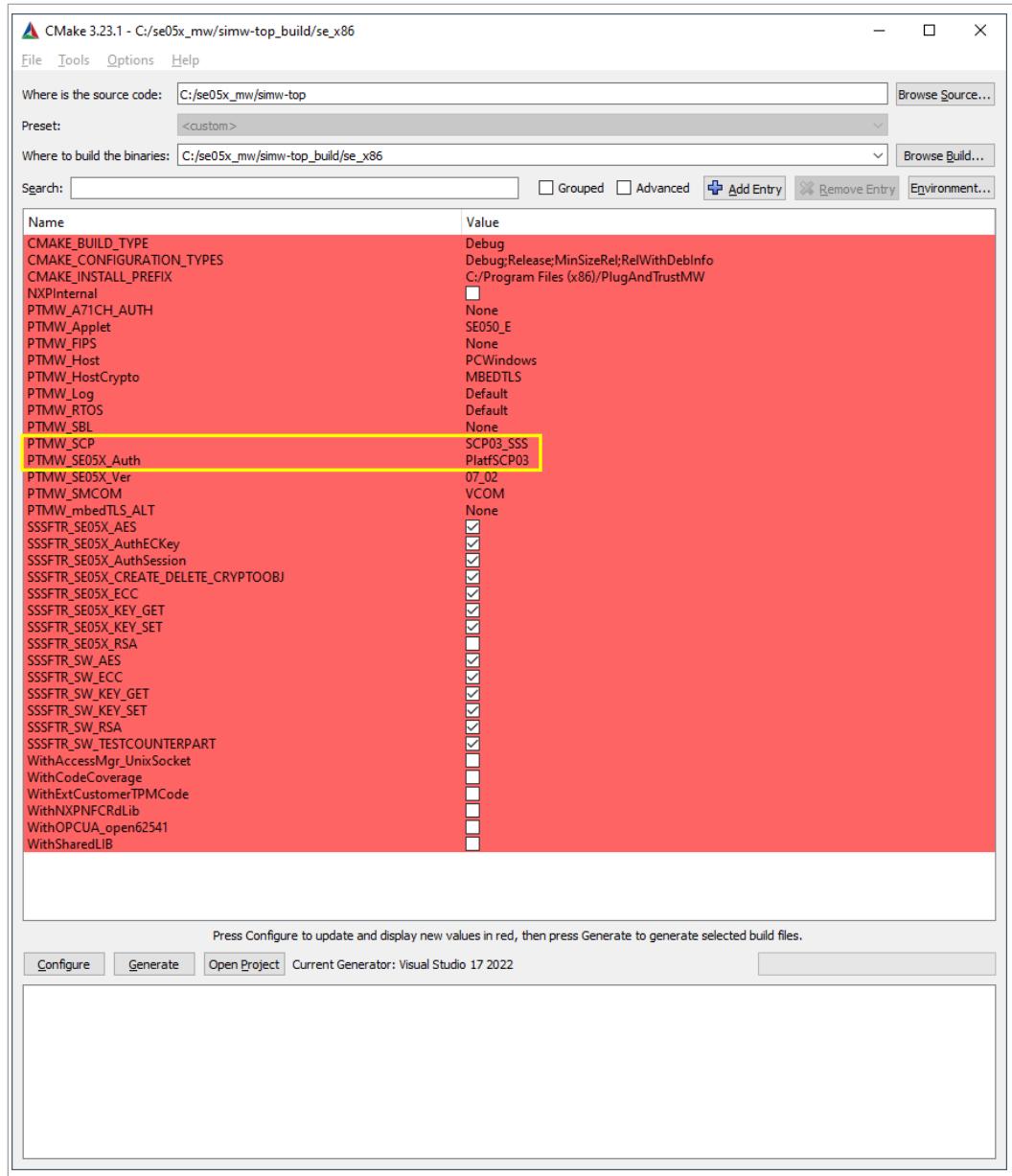


Figure 29. SE050E CMake Settings - PlatformSCP enabled

If you have edited any of the parameters in the menu, before exiting press the buttons **Configure** and **Generate** to apply the changes. In the next step we need to rebuild the Visual Studio solution. Finally, we can verify if we successfully enabled Platform SCP. For this purpose we run again the se05x\_minimal example as described in [Section 4.4](#).

## EdgeLock SE05x Quick start guide with Visual Studio project examples

```
Microsoft Visual Studio Debug Console
App :INFO :PlugAndTrust_v84.02.00_20220524
App :INFO :Running C:\se05x_mw\simw-top_build\se_x86\bin\Debug\se05x_Minimal.exe
App :INFO :Using PortName='\\.\COM9' (gszComPortDefault)
App :INFO :If you want to over-ride the selection, use ENV=EX_SSS_ROOT_SSS_PORT or pass in command line arguments.
App :INFO :Using default PlatformSCP03 keys. You can use keys from file using ENV=EX_SSS_BOOT_SCPO3_PATH.
Opening COM Port '\\.\COM9'
sss :INFO :atr (Len=35)
    01 A0 00 00    03 96 04 03    E8 00 FE 02    08 03 E8 00
    01 00 00 00    00 64 13 88    0A 00 65 53    45 30 35 31
    00 00 00
App :INFO :mem=32767
App :WARN :IF 32768 bytes or more bytes are available, 32767 bytes free is reported.
App :INFO :se05x_Minimal Example Success !!!...
App :INFO :ex_sss Finished

C:\se05x_mw\simw-top_build\se_x86\bin\Debug\se05x_Minimal.exe (process 2868) exited with code 0.
Press any key to close this window . . .
```

Figure 30. Verify that se05x\_minimal project is running with Platform SCP enabled

The Plug & Trust Middleware provides the following additional examples to rotate the PlatformSCP Keys and to mandate Platform SCP.

- **SE05X Rotate PlatformSCP Keys example:** Showcases authentication with default Platform SCP03 keys and the rotation (update) of those keys with user defined keys. The example documentation is available in the EdgeLock SE05x Plug & Trust Middleware documentation ([C:/se05x\\_mw/simw-top/doc/demos/se05x/se05x\\_RotatePlatformSCP03Keys/Readme.html](C:/se05x_mw/simw-top/doc/demos/se05x/se05x_RotatePlatformSCP03Keys/Readme.html)). The example source code is available at [C:/se05x\\_mw/simw-top/demos/se05x/se05x\\_RotatePlatformSCP03Keys](C:/se05x_mw/simw-top/demos/se05x/se05x_RotatePlatformSCP03Keys).
- **SE05X Mandate SCP example:** Showcases how to make Platform SCP03 authentication mandatory in EdgeLock SE05x. The example documentation is available in the EdgeLock SE05x Plug & Trust Middleware documentation ([C:/se05x\\_mw/simw-top/doc/demos/se05x/se05x\\_MandatePlatformSCP/Readme.html](C:/se05x_mw/simw-top/doc/demos/se05x/se05x_MandatePlatformSCP/Readme.html)). The example source code is available at [C:/se05x\\_mw/simw-top/demos/se05x/se05x\\_MandatePlatformSCP](C:/se05x_mw/simw-top/demos/se05x/se05x_MandatePlatformSCP).
- **SE05x AllowWithout PlatformSCP example:** This project demonstrates how to configure SE05X to allow without platform SCP. The example documentation is available in the EdgeLock SE05x Plug & Trust Middleware documentation ([~/se\\_mw/simwtop/doc/demos/se05x/se05x\\_AllowWithoutPlatformSCP/Readme.html](~/se_mw/simwtop/doc/demos/se05x/se05x_AllowWithoutPlatformSCP/Readme.html)). The example source code is available at [~/se\\_mw/simw-top/demos/se05x/se05x\\_AllowWithoutPlatformSCP](~/se_mw/simw-top/demos/se05x/se05x_AllowWithoutPlatformSCP).

## 7 Appendix A: Install Visual Studio 2022

Visual Studio is Microsoft's fully-featured IDE for Android, iOS, Windows, web, and cloud. Visual Studio 2022 introduces rich support for CMake, including cross-platform CMake projects.

This section explains how to install Visual Studio 2022 version, but the same procedure can be applied for more recent versions.

1. Go to [Visual Studio site](#).
2. Select (1) **Download** and click on **Community 2022** in the *Download Visual Studio* button as shown in [Figure 31](#) and [Figure 32](#):

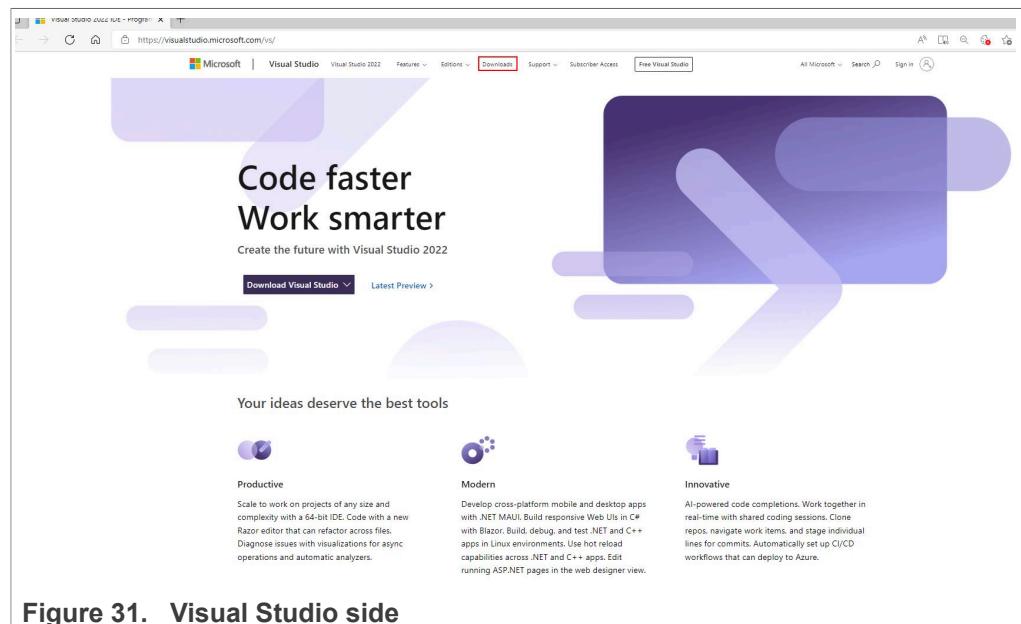


Figure 31. Visual Studio side

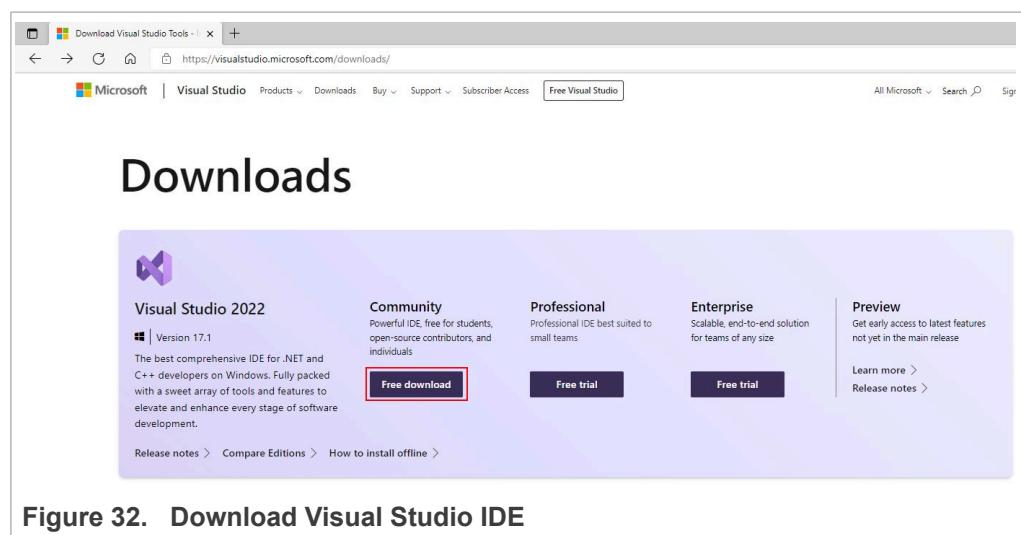


Figure 32. Download Visual Studio IDE

## EdgeLock SE05x Quick start guide with Visual Studio project examples

3. An \*.exe installer will download to your laptop. Double click on the installer file and follow the setup wizard until the installation is completed. This process might take a few minutes. [Figure 33](#) shows Visual Studio installation wizard as an example:

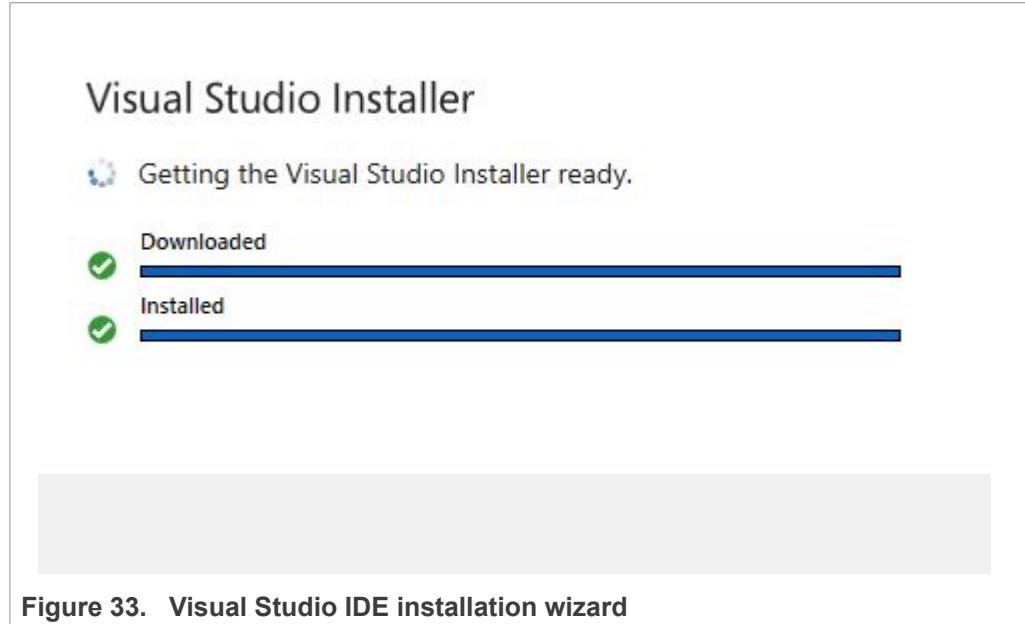


Figure 33. Visual Studio IDE installation wizard

4. As part of the Visual Studio setup, it is mandatory that you enable the installation of **Desktop development with C++**. Select (1) **Desktop development with C++** and (2) click install as shown in [Figure 34](#):

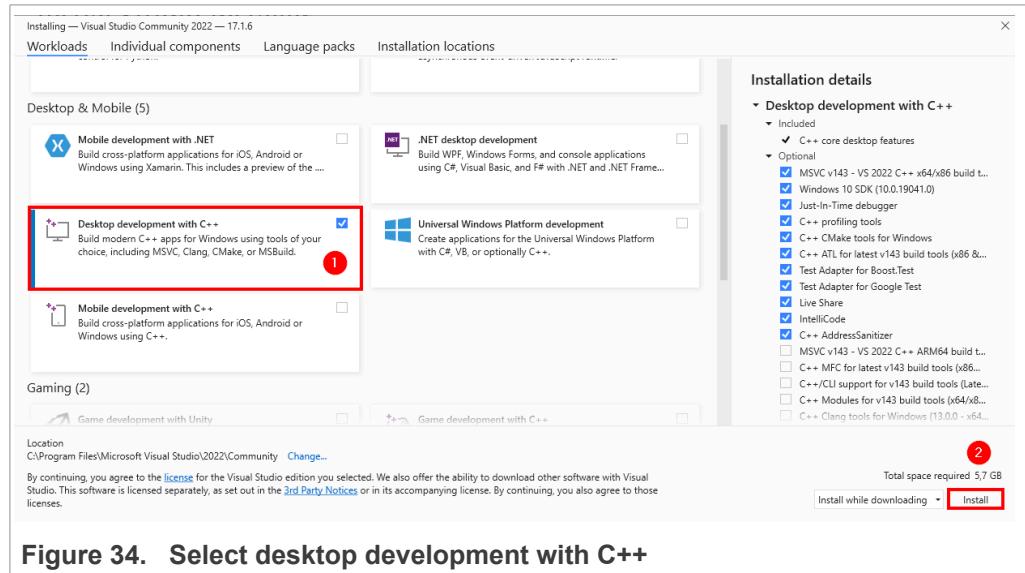
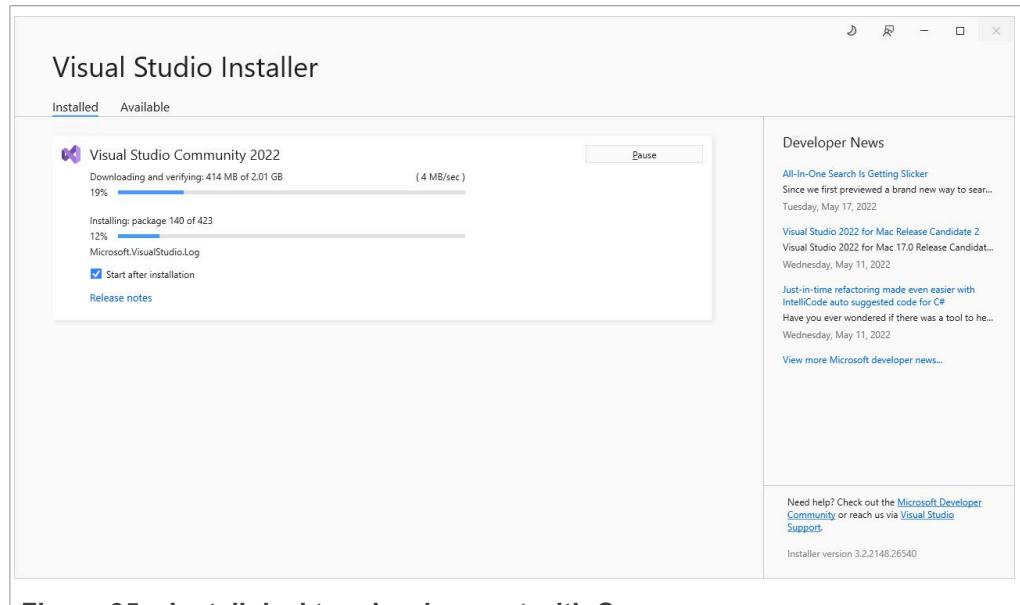


Figure 34. Select desktop development with C++

**EdgeLock SE05x Quick start guide with Visual Studio project examples**

5. Visual C++ Tools for CMake is installed by default as part of the Desktop development with C++ workload. This process might take several minutes. [Figure 35](#) shows Visual Studio installation wizard as an example:

**Figure 35. Install desktop development with C++**

6. After the installation is completed, you might be asked to reboot your system.

## 8 Appendix B: Install CMake

CMake is an open-source, cross-platform family of tools that helps you build C/C++ projects on multiple platforms using a compiler-independent method. It has minimal dependencies, requiring only a C++ compiler on its own build system. SE05x middleware leverages on CMake to generate native makefiles and workspaces that can be used in the compiler environment of your choice.

To install CMake:

1. Go to CMake downloads page: <https://cmake.org/download/>
2. Scroll down and select your binary distribution. For this guide, the binary distribution is Windows as shown in [Figure 36](#):

The screenshot shows the CMake download page. At the top, there's a navigation bar with links for About, Services, Resources, Download, and a search icon. Below the navigation, it says "Latest Release (3.22.3)". A note below this says: "The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gzipped tar files. To install a .sh file, run it with /bin/sh and follow the directions. The OS-machine.tar.gz files are gziped tar files of the install tree. The OS-machine.tar.Z files are compressed tar files of the install tree. The tar file distributions can be untared in any directory. They are prefixed by the version of CMake. For example, the linux-x86\_64 tar file is all under the directory cmake-linux-x86\_64. This prefix can be removed as long as the share, bin, man and doc directories are moved relative to each other. To build the source distributions, unpack them with zip or tar and follow the instructions in README.rst at the top of the source tree. See also the [CMake 3.22 Release Notes](#)".

Below this, there are two tables: "Source distributions" and "Binary distributions".

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.22.3.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.22.3.zip

Platform	Files
Windows x64 Installer: <b>Installer tool has changed. Uninstall CMake 3.4 or lower first!</b>	cmake-3.22.3-windows-x86_64.msi
Windows x64 ZIP	cmake-3.22.3-windows-x86_64.zip
Windows i386 Installer: <b>Installer tool has changed. Uninstall CMake 3.4 or lower first!</b>	cmake-3.22.3-windows-i386.msi
Windows i386 ZIP	cmake-3.22.3-windows-i386.zip
macOS 10.13 or later	cmake-3.22.3-macos-universal.dmg
	cmake-3.22.3-macos-universal.tar.gz
macOS 10.10 or later	cmake-3.22.3-macos10.10-universal.dmg
	cmake-3.22.3-macos10.10-universal.tar.gz
Linux x86_64	cmake-3.22.3-linux-x86_64.sh
	cmake-3.22.3-linux-x86_64.tar.gz
Linux aarch64	cmake-3.22.3-linux-aarch64.sh
	cmake-3.22.3-linux-aarch64.tar.gz

Figure 36. Download CMake

## EdgeLock SE05x Quick start guide with Visual Studio project examples

3. Double click on the downloaded installer file. Windows Defender SmartScreen might pop-up the wizard shown in [Figure 37](#):



Figure 37. Execute CMake installer

4. If this is your case: Click (1) on **More info** and then (2) click on **Run anyway** as shown in [Figure 38](#):

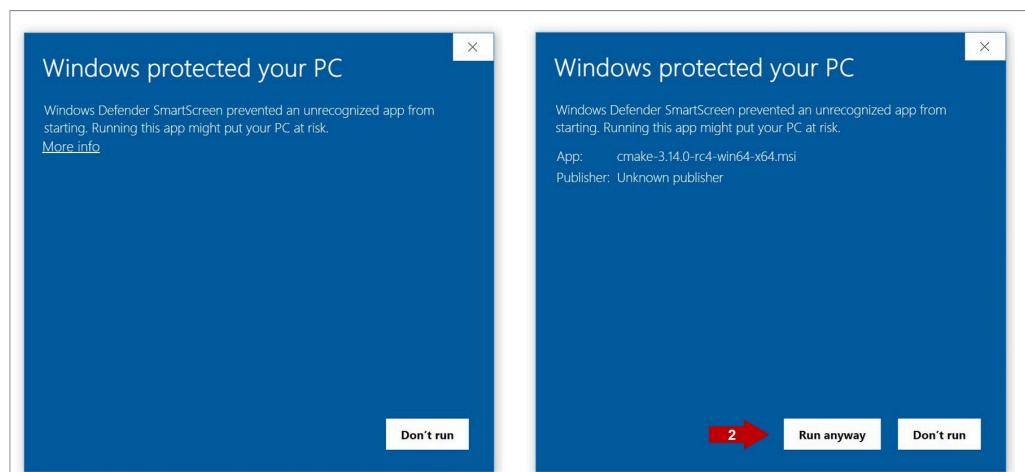


Figure 38. Run the CMake installer (II)

## EdgeLock SE05x Quick start guide with Visual Studio project examples

5. The CMake installation wizard will open. Click (1) **Next** and (2) **accept** the End-User License Agreement as shown in [Figure 39](#):

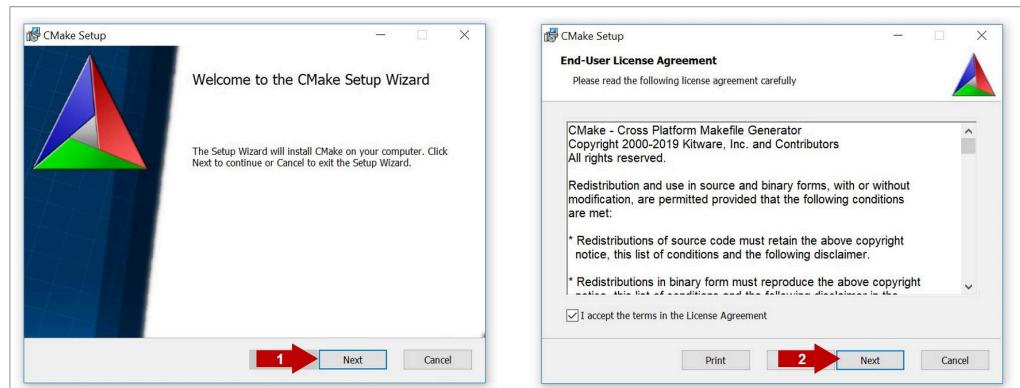


Figure 39. CMake installation wizard

6. As part of the CMake setup, (1) **Add Cmake to the system PATH for all users** and (2) click **Next** as shown in [Figure 40](#):

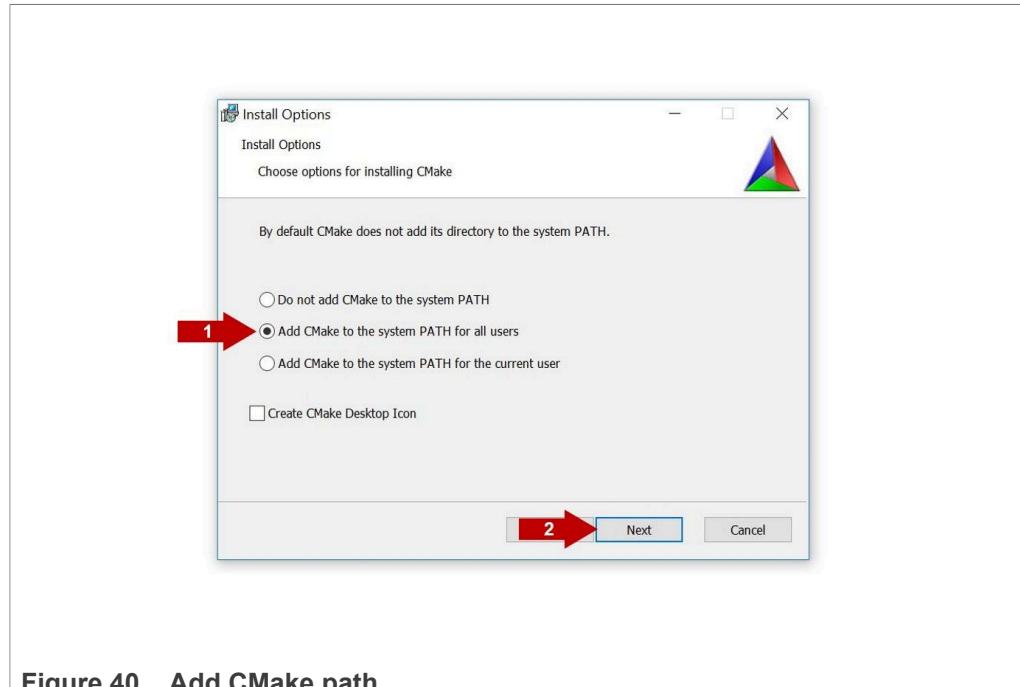


Figure 40. Add CMake path

## EdgeLock SE05x Quick start guide with Visual Studio project examples

7. Select a destination folder, (1) click **Next** and then (2) click **Install** as shown in [Figure 41](#):

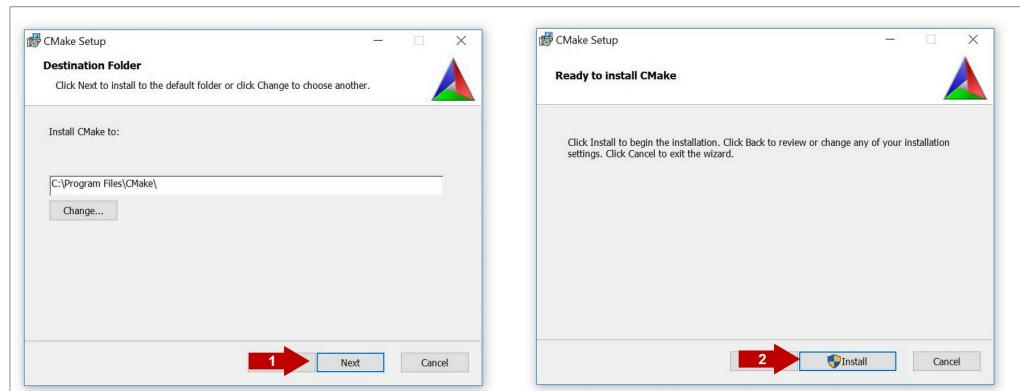


Figure 41. Install CMake

8. Wait a few seconds until the installation is completed and click **Finish** as shown in [Figure 42](#):

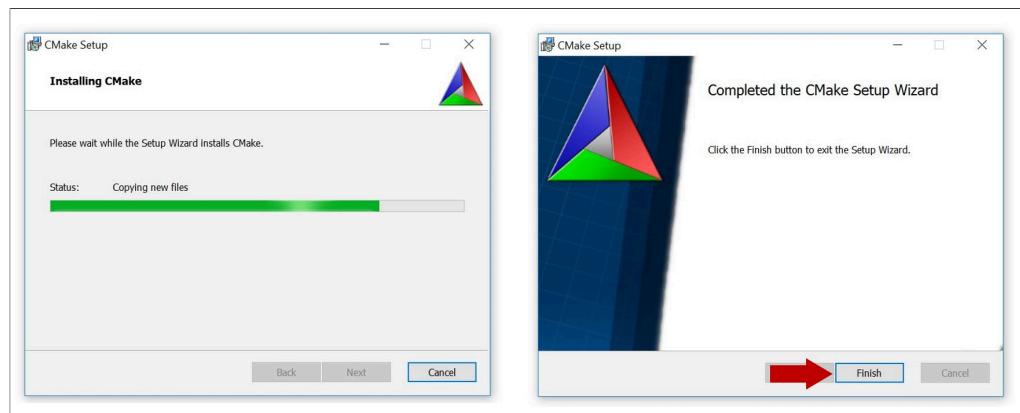


Figure 42. Complete CMake installation

## 9 Appendix C: Install Python

This section explains how to install Python  $\geq 3.7.x$  and  $\leq 3.9.x$  32-bit version, but the same procedure can be applied for more recent versions. Follow these steps to install Python in your local machine:

1. Go to <https://www.python.org/downloads> and download **Python  $\geq 3.7.x$  and  $\leq 3.9.x$  32-bit version**. Make sure you download the Python 32 bit version.

Files						
Version	Operating System	Description	MD5 Sum	File Size	PGP	
Gzipped source tarball	Source release		1440acb71471e2394befdb30b1a958d1	25800844	SIG	
XZ compressed source tarball	Source release		e754c4b2276750fd5b4785a1b443683a	19154136	SIG	
macOS 64-bit Intel-only installer	macOS	for macOS 10.9 and later, deprecated	2714cb9e0241cf7e2f9022714a55d27a	30395760	SIG	
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	c2393ab11a423d817501b856ab5da9f	38217233	SIG	
Windows embeddable package (32-bit)	Windows		c1d2af96d9f3564f57f35fc3c1006eb	7671509	SIG	
Windows embeddable package (64-bit)	Windows		b8e8bfba8e56edcd654d15e3bcd2e29a	8509821	SIG	
Windows help file	Windows		784020441c1a25289483d8771a8215	9284044	SIG	
<b>Windows installer (32-bit)</b>	Windows		<b>457d648dc8a71b6bc32da30a7805c55b</b>	<b>27767040</b>	<b>SIG</b>	←
Windows installer (64-bit)	Windows	Recommended	747ac35ae667f4ec1ee3b001e9b7dbc6	28909456	SIG	

Figure 43. Download Python 3.9.x 32 bit version

2. Double click on the downloaded installer file. Select the "*Install launcher for all users*" and "*Add Python 3.7 to Path*" options and click *Install Now* as indicated in Figure 44:

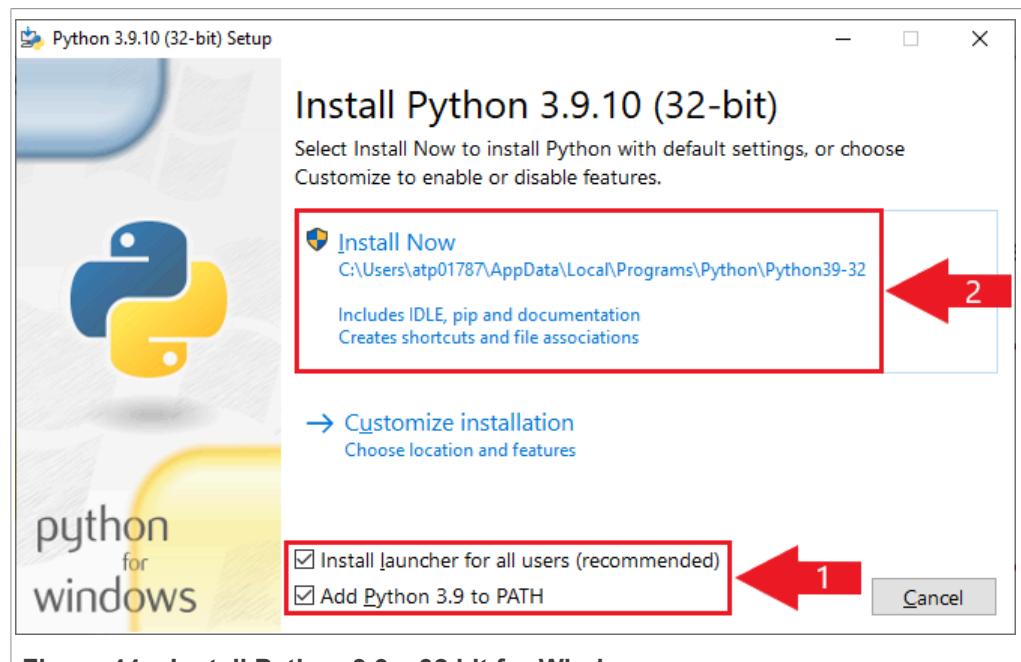
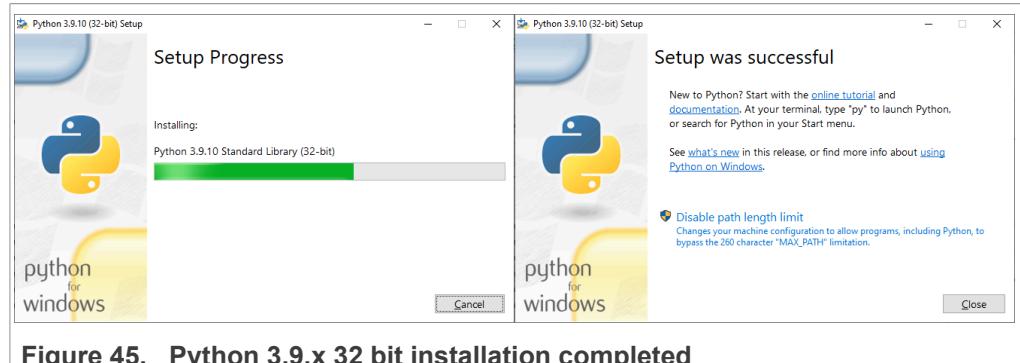


Figure 44. Install Python 3.9.x 32 bit for Windows

**EdgeLock SE05x Quick start guide with Visual Studio project examples**

3. Wait a few seconds until the installation is completed as indicated in [Figure 45](#)



**Figure 45. Python 3.9.x 32 bit installation completed**

## 10 Appendix D: Update FRDM-K64F board with DAPLink firmware

Arm Mbed DAPLink is an open-source software project that enables programming and debugging application software running on Arm Cortex CPUs. DAPLink runs an open-source bootloader and enables developers with drag-and-drop programming, a serial port and CMSIS-DAP based debugging.

**Note:** To debug MCUXpresso project examples, we need to flash FRDM-K64F with DAPLink firmware. If your FRDM-K64F board already includes DAPLink firmware, you can skip these steps.

To flash DAPLink firmware, follow these steps:

1. Go to [NXP OpenSDA](#) site
2. Scroll down and select FRDM-K64F board from the **Download - OpenSDA bootloader and application** drop down list as indicated in [Figure 46](#):

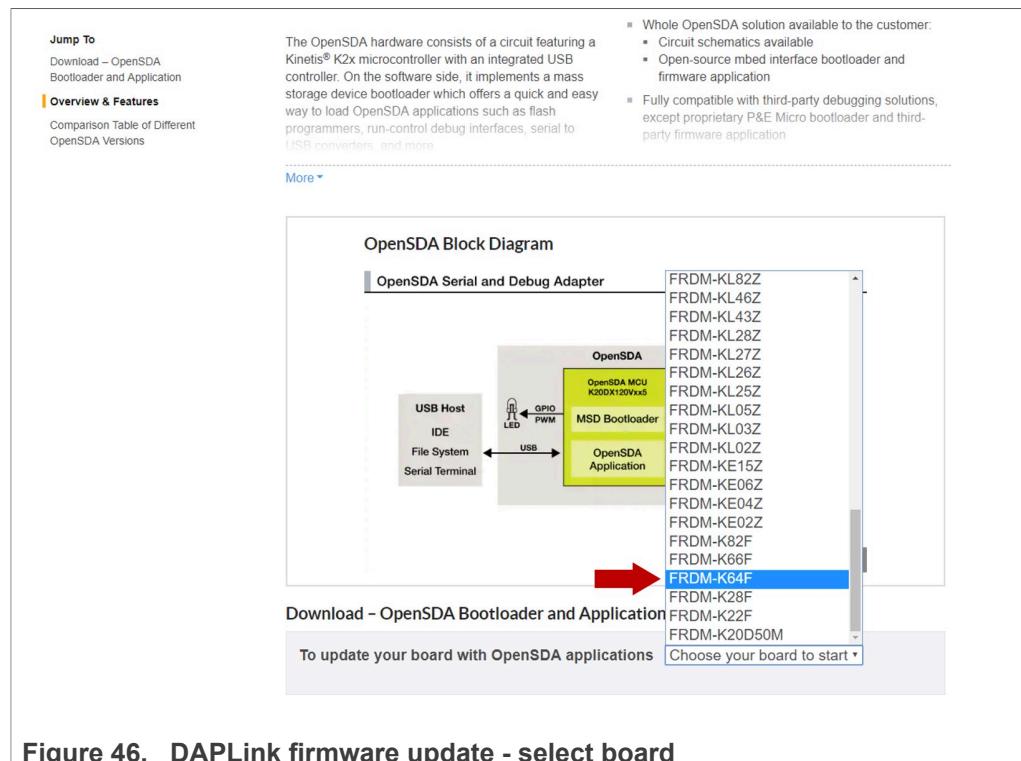


Figure 46. DAPLink firmware update - select board

## EdgeLock SE05x Quick start guide with Visual Studio project examples

3. Download the latest DAPLink firmware version as shown in [Figure 47](#):

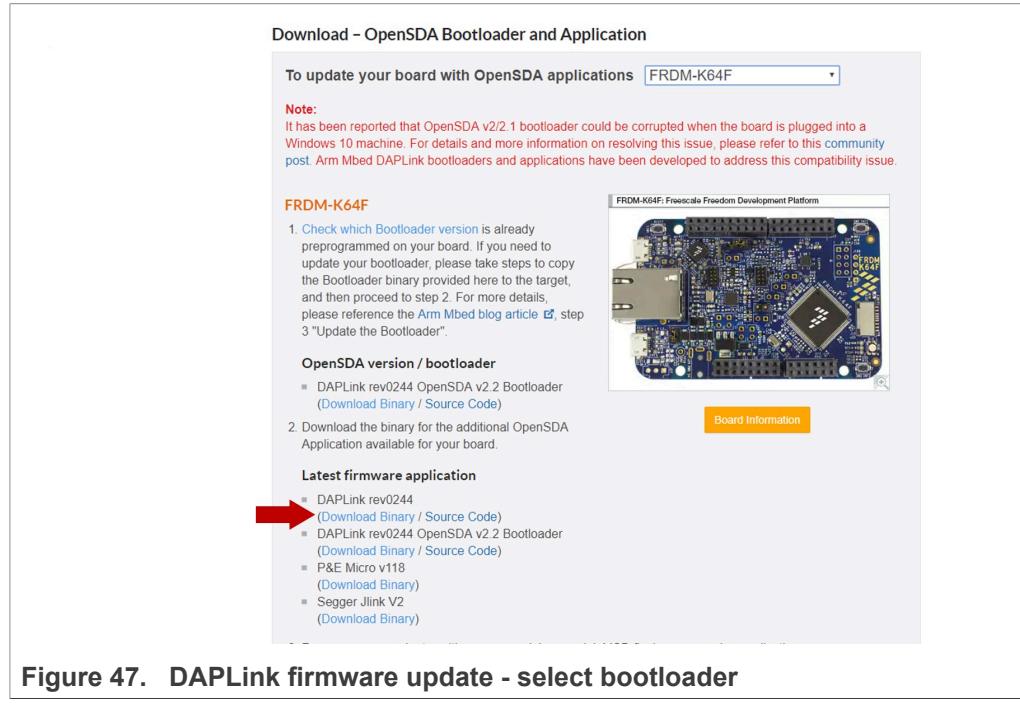


Figure 47. DAPLink firmware update - select bootloader

4. Start the board's bootloader mode. To do so, (1) keep reset button pressed while (2) connecting the USB cable to the SDA USB port and release it after 1s ([Figure 48](#)):

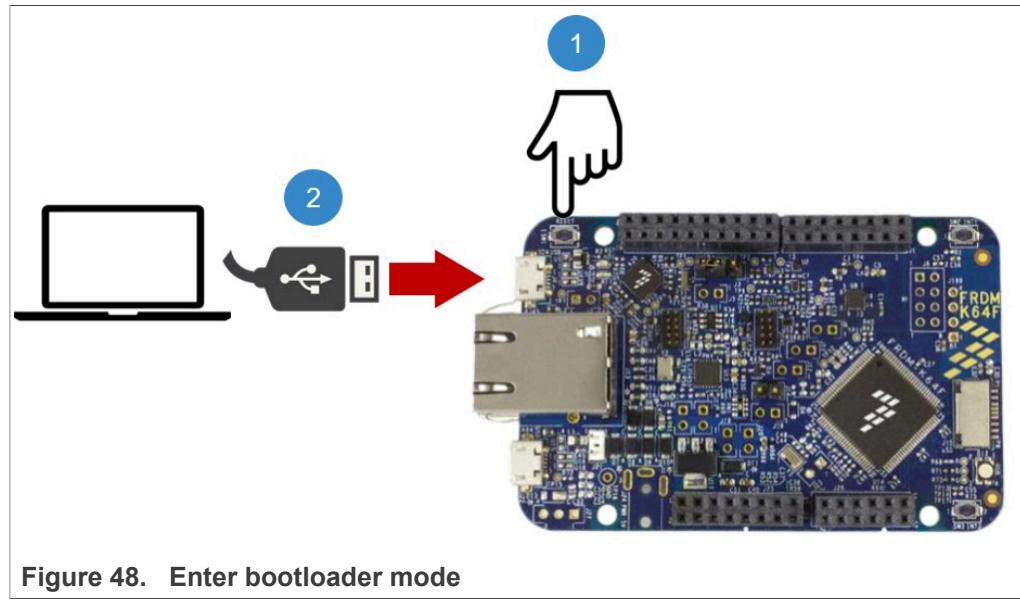


Figure 48. Enter bootloader mode

## EdgeLock SE05x Quick start guide with Visual Studio project examples

5. Drag and drop or copy and paste the binary file into the BOOTLOADER drive from your computer file explorer as shown in [Figure 49](#). The FRDM-K64F will automatically un-mount after the drag and drop operation.

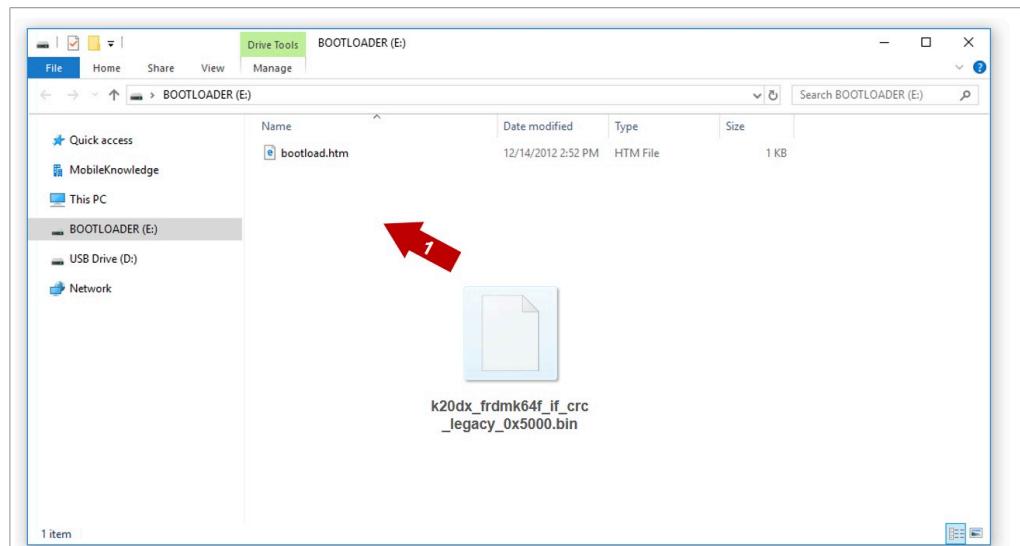


Figure 49. Enter bootloader mode

6. Un-plug and re-plug the USB cable from the SDA USB port **without** keeping reset button pressed.
7. Check the category Ports (COM & LPT) from your computer Device Manager to ensure that new devices have been properly detected and their driver correctly installed by your computer OS.



Figure 50. Enter bootloader mode

**Note:** In case the device does not show up in your Device Manager, please download the latest bootloader version, as shown in [Figure 47](#), or check / exchange the USB cables used.

## 11 Legal information

### 11.1 Definitions

**Draft** — A draft status on a document indicates that 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 in a draft version of a document and shall have no liability for the consequences of use of such information.

### 11.2 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.

**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.

**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.

**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.

### 11.3 Trademarks

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

## Tables

Tab. 1.	EdgeLock SE05x development boards .....	4	Tab. 9.	SCP03 session keys .....	26
Tab. 2.	FRDM-K64F details .....	5	Tab. 10.	Platform SCP key define prefix for SE050E product variants .....	29
Tab. 3.	CMake Settings for SE050E product variants .....	21	Tab. 11.	Platform SCP key define prefix for SE050F product variants .....	29
Tab. 4.	CMake Settings for SE050F product variants .....	21	Tab. 12.	Platform SCP key define prefix for SE050 Previous Generation product variants .....	29
Tab. 5.	CMake Settings for SE050 Previous Generation product variants .....	21	Tab. 13.	Platform SCP key define prefix for SE051 product variants .....	29
Tab. 6.	CMake Settings for SE051 product variants ....	22	Tab. 14.	Platform SCP key define prefix for A5000 product variants .....	30
Tab. 7.	CMake Settings for A5000 product variants ....	23			
Tab. 8.	Static SCP03 keys .....	26			

## Figures

Fig. 1.	Jumper configuration for FRDM-K64F .....	6
Fig. 2.	Arduino connectors of OM-SE05xARD and FRDM-K64F boards .....	7
Fig. 3.	OM-SE05xARD mounted in FRDM-K64F board .....	7
Fig. 4.	Unplug and plug OpenSDA port .....	8
Fig. 5.	FRDM-K64F drive .....	8
Fig. 6.	VCOM binary folder .....	9
Fig. 7.	Drag and drop VCOM binary .....	9
Fig. 8.	Check VCOM and serial ports .....	10
Fig. 9.	Create se05x_mw folder .....	11
Fig. 10.	Unzip se050 middleware .....	12
Fig. 11.	Generate Plug & Trust middleware define the environment .....	13
Fig. 12.	Generate Plug & Trust middleware project examples .....	13
Fig. 13.	SE050 middleware project structure .....	14
Fig. 14.	Plug OpenSDA and USB FRDM-K64F ports .....	14
Fig. 15.	Open the CMake configuration interface .....	15
Fig. 16.	Review CMake configuration .....	16
Fig. 17.	Open PlugAndTrustMW.sln Visual Studio project solution .....	17
Fig. 18.	PlugAndTrustMW.sln Visual Studio project workspace .....	18
Fig. 19.	Change VCOM port number in your solution .....	18
Fig. 20.	Set se05x_minimal as StartUp project .....	19
Fig. 21.	Build se05x_minimal project .....	19
Fig. 22.	Run se05x_minimal project .....	20
Fig. 23.	Verify that se05x_minimal project is running ...	20
Fig. 24.	SE050E CMake Settings - Plain communication .....	24
Fig. 25.	SPC03 mutual authentication – principle .....	27
Fig. 26.	SPC03 Encryption and MACing principle .....	27
Fig. 27.	Default Platform SCP keys are defined in ex_sss_tp_scp03_keys.h .....	28
Fig. 28.	Select the default Platform SCP keys infsl_sss_ftr.h .....	28
Fig. 29.	SE050E CMake Settings - PlatformSCP enabled .....	31
Fig. 30.	Verify that se05x_minimal project is running with Platform SCP enabled .....	32
Fig. 31.	Visual Studio side .....	33
Fig. 32.	Download Visual Studio IDE .....	33
Fig. 33.	Visual Studio IDE installation wizard .....	34
Fig. 34.	Select desktop development with C++ .....	34
Fig. 35.	Install desktop development with C++ .....	35
Fig. 36.	Download CMake .....	36
Fig. 37.	Execute CMake installer .....	37
Fig. 38.	Run the CMake installer (II) .....	37
Fig. 39.	CMake installation wizard .....	38
Fig. 40.	Add CMake path .....	38
Fig. 41.	Install CMake .....	39
Fig. 42.	Complete CMake installation .....	39
Fig. 43.	Download Python 3.9.x 32 bit version .....	40
Fig. 44.	Install Python 3.9.x 32 bit for Windows .....	40
Fig. 45.	Python 3.9.x 32 bit installation completed .....	41
Fig. 46.	DAPLink firmware update - select board .....	42
Fig. 47.	DAPLink firmware update - select bootloader .....	43
Fig. 48.	Enter bootloader mode .....	43
Fig. 49.	Enter bootloader mode .....	44
Fig. 50.	Enter bootloader mode .....	44

## Contents

<b>1</b>	<b>How to use this document .....</b>	<b>3</b>
<b>2</b>	<b>Hardware required .....</b>	<b>4</b>
<b>3</b>	<b>Boards setup .....</b>	<b>6</b>
3.1	OM-SE05xARD jumper configuration .....	6
3.2	OM-SE05xARD and FRDM-K64F board connection .....	6
3.3	Flash FRDM-K64F with VCOM software .....	7
<b>4</b>	<b>Run Plug &amp; Trust middleware Visual Studio project examples .....</b>	<b>11</b>
4.1	Prerequisites .....	11
4.2	Download Plug & Trust middleware .....	11
4.3	Build Plug & Trust middleware project examples .....	12
4.4	Execute EdgeLock SE05x Visual Studio project examples .....	14
<b>5</b>	<b>Product specific CMake build settings .....</b>	<b>21</b>
5.1	Example: SE050E CMake build settings .....	23
<b>6</b>	<b>Binding EdgeLock SE05x to a host using Platform SCP .....</b>	<b>25</b>
6.1	Introduction to the Global Platform Secure Channel Protocol 03 (SCP03) .....	25
6.2	How to configure the product specific default Platform SCP keys .....	27
6.3	How to enable Platform SCP .....	30
<b>7</b>	<b>Appendix A: Install Visual Studio 2022 .....</b>	<b>33</b>
<b>8</b>	<b>Appendix B: Install CMake .....</b>	<b>36</b>
<b>9</b>	<b>Appendix C: Install Python .....</b>	<b>40</b>
<b>10</b>	<b>Appendix D: Update FRDM-K64F board with DAPLink firmware .....</b>	<b>42</b>
<b>11</b>	<b>Legal information .....</b>	<b>45</b>

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.