AN12449

Sensor data protection with EdgeLock^{TM} SE05x

Rev. 1.3 — 6 July 2021 Application note
546813

Document information

Information	Content
Keywords	EdgeLock SE05x, Sensor data protection
Abstract	This application note describes how to leverage EdgeLock SE05x for guaranteeing sensor data protection against remote attacks. It gives insights into the integration of EdgeLock SE05x from a hardware and software perspective for this use case. It also provides detailed instructions to run a code example that demonstrates how to leverage EdgeLock SE05x to protect data from a security-sensitive sensor.

Revision history

Revision history

Revision number	Date	Description
1.3	2021-07-06	• Reformulating introduction in Section 1. • Change Figure 4 and Figure 5
1.2	2020-12-07	Updated to latest template and fixed broken URLs
1.1	2019-11-25	Updated section 5 referring to MW v02.11.03
1.0	2019-10-18	First document release

1 EdgeLock SE05x for sensor data protection use case

As IoT becomes more and more connected, data are captured everywhere through different kinds of sensors. These data are analyzed in the backend or cloud and potentially trigger actions on different actuators. Often these sensors are in distributed sensor networks without direct human control; for instance, a wind turbine that stops in case sensors detect a wind becoming too strong. As such, these sensors and sensor data become an interesting entry point for hackers through manipulating sensor data or even remotely manipulating the sensor itself. Therefore, companies need to take care that these data cannot be manipulated and that a remote attack on the sensor data is detected for reliability.

Unprotected sensor data passed and handled inside an MCU is a potential security thread as the sensor data can be manipulated when there are no cryptographic integrity measures. We can majorly improve on this aspect with the use of a dedicated security IC like the EdgeLock SE05x.

The EdgeLock SE05x is designed to be used as a companion chip to any type of MCU or MPU and sensors can be directly connected to it using an I^2C master interface as depicted in Figure 1:

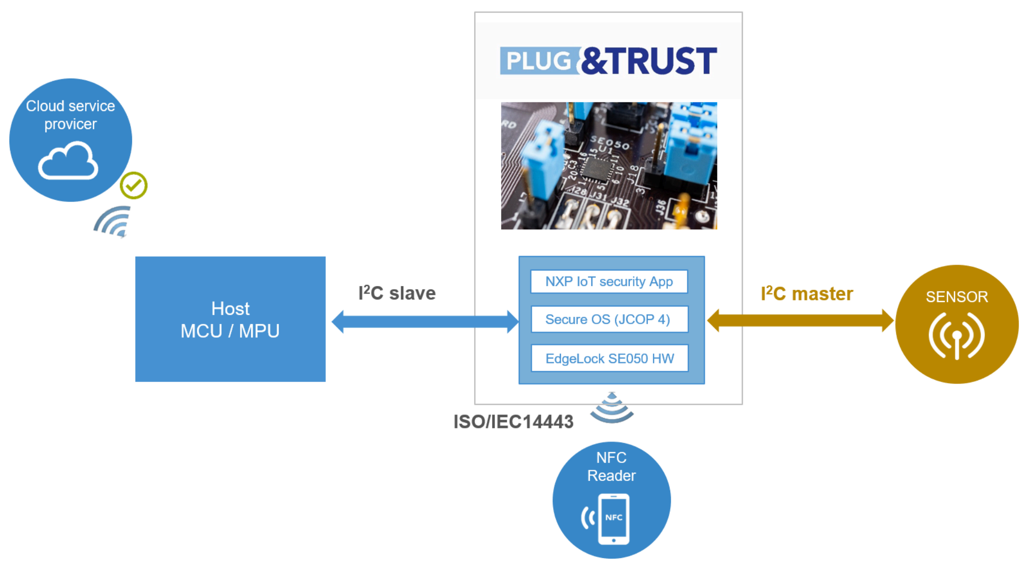


Figure 1. EdgeLock SE05x block diagram

The EdgeLock SE05x allows you to set up a secure, end-to-end connection from the secure element connected directly to the sensor or actuator to your local IoT gateway or cloud-based service, protecting the interface between the sensor and the security IC. As such, EdgeLock SE05x helps you to provide a higher level of security in your IoT system by:

- **Preventing remote data manipulation**: The data extracted by the sensor is collected directly by the secure element which adds a cryptographic integrity protection layer.
- **Authenticating the sensor:** The system authenticates the sensor data and adds a proof of origin for the data to come from a specific IoT device.

- **Providing end-to-end security on the communication channel**: The data collected from the sensor can be encrypted and securely transferred to your gateway or cloud for further treatment and analysis.

2 EdgeLock SE05x hardware integration

The EdgeLock SE05x works as an auxiliary security IC attached to a host MCU using an I^2C interface. The host MCU represents the I^2C master and the EdgeLock SE05x is the slave in the I^2C bus.

Besides the mandatory connection to the host MCU, a sensor node or similar element can be connected to EdgeLock SE05x using an additional I^2C interface. In this case, the EdgeLock SE05x is the I^2C master while the sensor node must operate as the slave in the I^2C bus. This interface features a maximum clock rate of 400 kHz.

This section gives insights into the integration of EdgeLock SE05x in your IoT sensor nodes from a hardware perspective. It includes:

- EdgeLock SE05x pinout description
- EdgeLock SE05x application circuit
- How to connect an external sensor using the OM-SE050ARD board.

2.1 EdgeLock SE05x pinout description

The EdgeLock SE05x is delivered in a HX2QFN20 flat package (SOT1969-1) of 20 pins and dimensions of 3x3x033 millimeters. The SE_IO1 and SE_IO2 pins are used as the SDA and SCL lines respectively for the I^2C master interface. Figure 2 shows the EdgeLock SE05x package pinout description and highlights in green the location of the SE_IO1 and SE_IO2 pins.

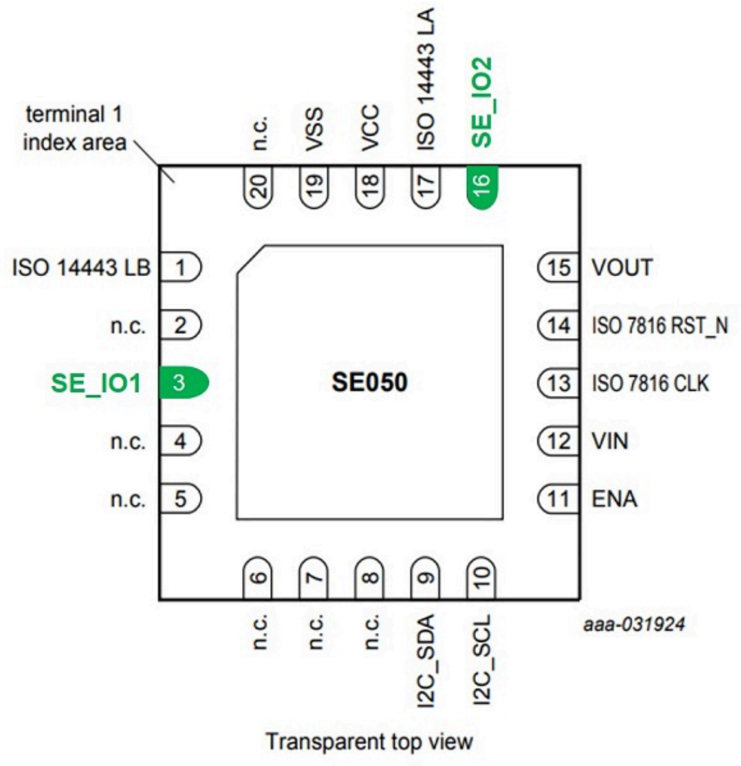


Figure 2. EdgeLock SE05x pinout description

Table 1 provides a description of the pins in the EdgeLock SE05x package. The pins related to the I^2C master interface used for sensor connection have been intentionally highlighted in bold.

Table 1. EdgeLock SE05x pin description (HX2QFN20)

Pin	Symbol	Description
1	ISO 14443 LB	ISO14443 antenna connection
3	**SE_IO1**	**ISO 7816 IO1, GPIO or I^2C Master SDA**
9	I^2C_SDA	I^2C slave SDA
10	I^2C_SCL	I^2C slave SCL
11	ENA	Input for power switch between Vin and Vout (high=on)
12	VIN	Power supply for power switch, IO2, SDA and SCL.
13	ISO 7816 CLK	ISO 7816 clock input (not active in I^2C mode)
14	ISO 7816 RST_N	ISO 7816 reset input low active (not active in I^2C mode)
15	VOUT	Power switch output (connect to Vcc)
16	**SE_IO2**	**ISO7816 IO2, GPIO pad or I^2C Master SCL**
17	ISO 14443 LA	ISO14443 antenna connection
18	VCC	Core power supply
19	VSS	Ground

2.2 EdgeLock SE05x application circuit

EdgeLock SE05x has two power supply domains, referred to as the *core domain* and the *IO domain* and two supply pins, *Vcc* and *Vin*. The *Vcc* supplies the *core domain* and *Vin* supplies the *IO domain* as depicted in Figure 3. The EdgeLock SE05x voltage range is 1.8V - 3.6V (it is operable up to 5V but not fully characterized).

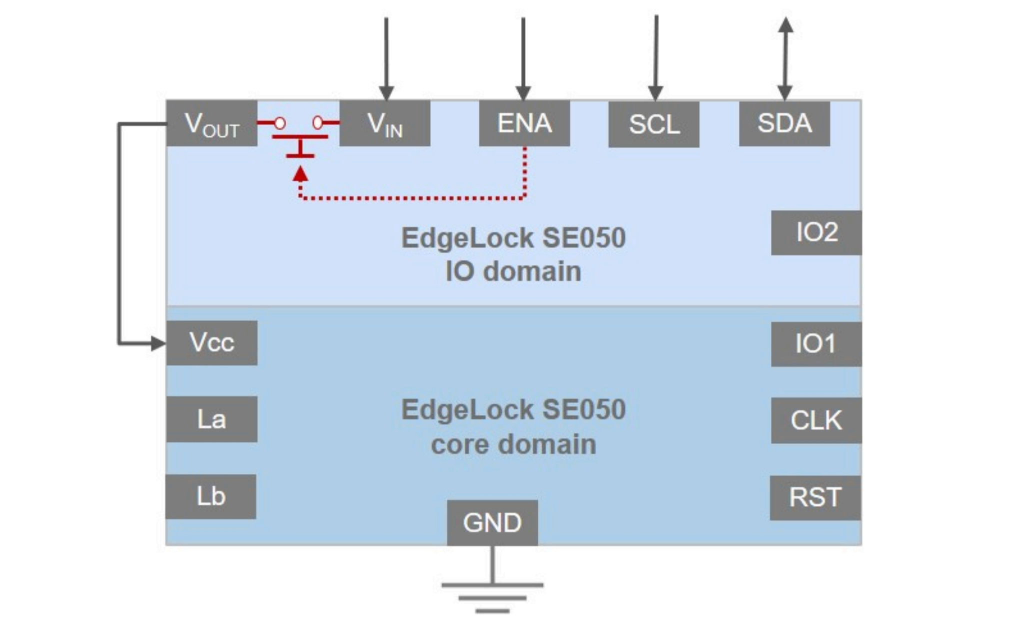


Figure 3. EdgeLock SE05x power supply domains

Figure 4 describes the application circuit to integrate EdgeLock SE05x into your IoT sensor. The EdgeLock SE05x is connected to the host MCU using the I^2C slave interface (SDA and SCL lines). The host MCU provides the clock and pull-up supply on the I^2C bus and controls the supply switch via the ENA pin. The EdgeLock SE05x *Vin* pin is supplied by the main supply source of the system (V_{DD}). *Vcc* is supplied directly by *Vout* and the sensor is connected to the IO1 and IO2 pins with external pull-up resistors, used as SDA and SCL lines towards EdgeLock SE05x.

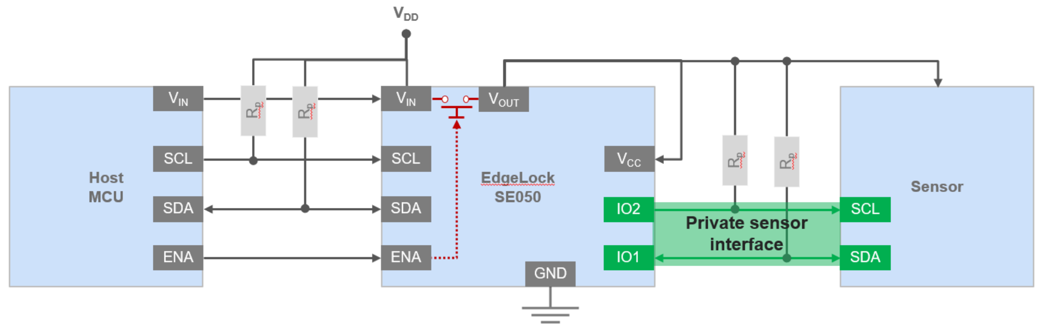


Figure 4. EdgeLock SE05x application circuit for sensor connection

Figure 5 shows another possible architecture where the sensor subsystem is directly connected to the host MCU and to the EdgeLock SE05x using two different interfaces. In this architecture, the interface between the host MCU and the sensor can be used for high data throughput while the interface between the sensor and the EdgeLock SE05x can be used as the sensor configuration interface. In such case, we can achieve a low latency data link while still having attested sensor configuration via the secure I^2C interface with the EdgeLock SE05x.

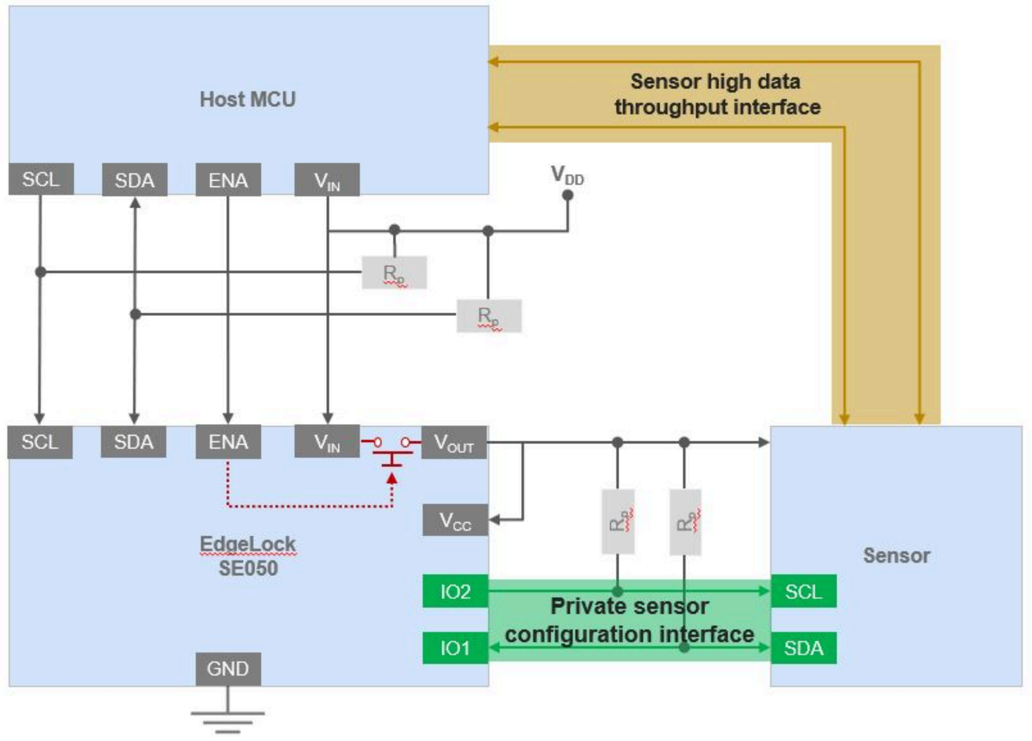


Figure 5. EdgeLock SE05x application circuit for sensor with two I^{2}C interfaces

2.3 Sensor connection using the OM-SE050ARD board

The OM-SE050ARD comes with headers and connectors that allow us to access the EdgeLock SE05x interfaces, including the I^{2}C master lines to connect a sensor node. Therefore, we can use this flexible and easy-to-use development kit to evaluate the EdgeLock SE05x features, build a proof of concept or prototype our IoT sensor solution before going to production.

The EdgeLock SE05x I^{2}C master lines are accessible via the OM-SE050ARD J11 jumper. The J11 jumper is a 10-pin header with male connectors soldered by default in OM-SE050ARD. Figure 6 indicates in red the location of the I^{2}C master lines to connect an external sensor to the OM-SE050ARD.

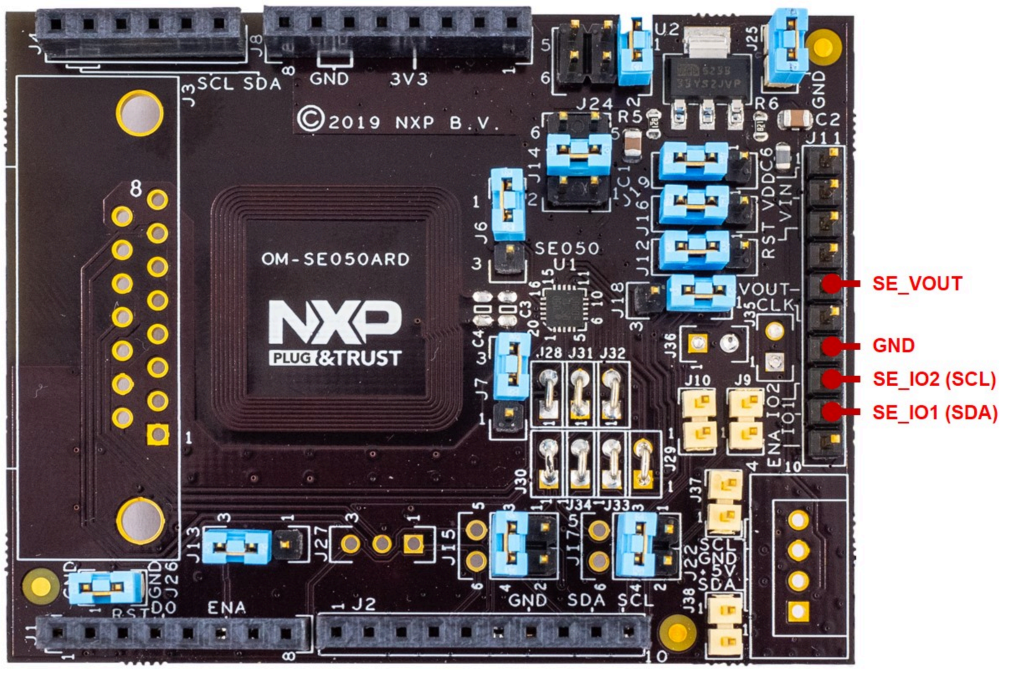


Figure 6. OM-SE050ARD I2C master interface connection

Note: *J18 must be set to position 1-2 to access SE_IO2 from J11 header.*

Note: *IO1 and IO2 lines can also be accessed from the mounting holes for the DB15 connector.*

Note: *IO2 can also be accessed from Arduino-R3 header J2:8 if J18 is set in posiition 2-3.*

3 Software integration with EdgeLock SE05x Plug & Trust Middleware

The EdgeLock SE05x Plug & Trust Middleware is a single software stack designed to facilitate the integration of EdgeLock SE05x into your host MCU software. This middleware has built-in cryptographic and device identity features, abstracts the APDU commands and communication interface exposed by the EdgeLock SE05x, and it is directly accessible from stacks like OpenSSL, mbedTLS or other cryptographic libraries. It also comes with support for various NXP MCU / MPU platforms and can be ported to multiple host platforms and host operating systems. Figure 7 is a simplified representation of the layers and components which EdgeLock SE05x Plug & Trust Middleware is made of:

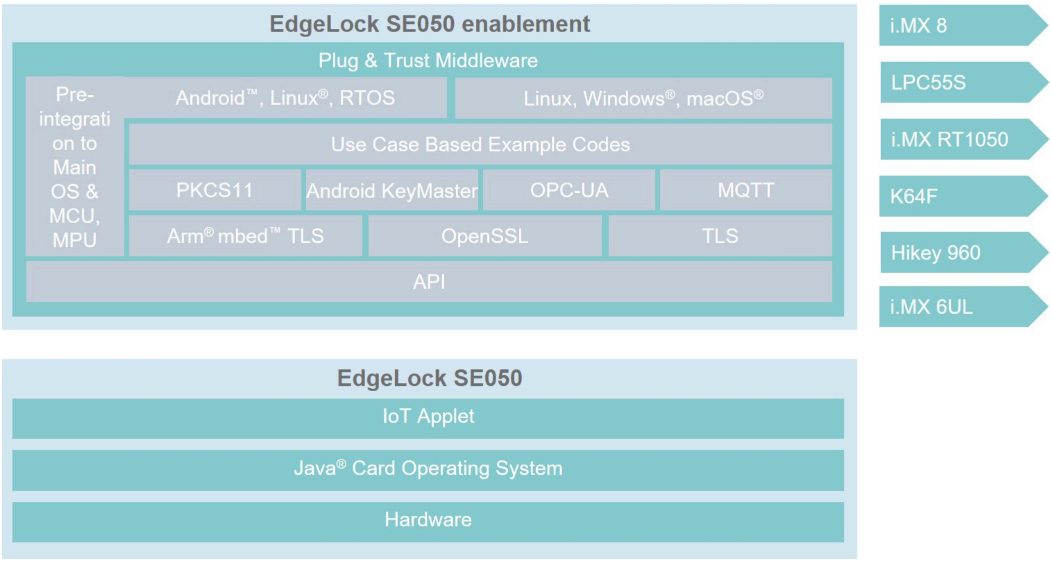


Figure 7. NXP Plug & Trust middleware block diagram

This section gives insights into the integration of EdgeLock SE05x from a software perspective. It includes a short overview of the useful functions included in the EdgeLock SE05x Plug & Trust Middleware to read and write data from the EdgeLock SE05x I^2C master interface.

3.1 EdgeLock SE05x Plug & Trust Middleware I^2C master API

The EdgeLock SE05x Plug & Trust Middleware provides two functions to read and write data from a sensor connected over the I^2C master interface, with or without data attestation. These two functions are:

- `smStatus_t Se05x_i2c_master_txn();`
- `smStatus_t Se05x_i2c_master_attst_txn();`

The `smStatus_t Se05x_i2c_master_txn()` function allows us to send a number of bytes to read and to be written from the connected sensor over the EdgeLock SE05x I^2C master interface.

The `smStatus_t Se05x_i2c_master_attst_txn()` function allows us to read data with attestation, guaranteeing that data is not manipulated by an attacker. Using this function, data collected from the sensor is signed with one of the secret keys stored in EdgeLock SE05x. This is the recommended function to ensure that sensor is not manipulated.

Both the `smStatus_t Se05x_i2c_master_txn()` and the `smStatus_t Se05x_i2c_master_attst_txn()` expect as a function parameter an array of commands in a TLV format. This array of commands, or command set, is constructed as a sequence of TLV instructions described in Table 2

Table 2. EdgeLock SE05x I^{2}C master command set TLVs

Instruction	TLV-type	Description
CONFIGURE	0x01	This TLV type is used to configure the I^{2}C slave address or the SCL clock (100kHz or 400 Khz)
WRITE	0x03	This TLV type is used to write a number of bytes in the I^{2}C master interface
READ	0x04	This TLV tpye is used to read a number of bytes from the I^{2}C master interface

The EdgeLock SE05x autonomously executes the list of commands received from the host MCU and responds with the read bytes for the READ TVLs and a return code for the CONFIGURE and WRITE TLVs commands. In addition, if data is read with attestation, it responds with a timestamp, a random, the chip unique ID, and a signature over the previous values concatenated.

Go to Section 4 to learn how to run a source code example that leverages the EdgeLock SE05x Plug & Trust Middleware I^{2}C master functions to securely read data from a sensor.

3.2 EdgeLock SE05x Plug & Trust Middleware I^{2}C master API documentation

You can refer to the code documentation provided as part of the EdgeLock SE05x Plug & Trust Middleware for full details about the I^{2}C master API. To open the HTML documentation:

1. Go to the directory where you unzipped EdgeLock SE05x Plug & Trust Middleware
2. Go to simw-top\doc folder,in your EdgeLock SE05x Plug & Trust Middleware package
3. Double click in the `index.html` file.

4. A browser with the documentation landing page will be opened. Navigate to ***3. Plug and Trust MW Stack*** and ***3.5 I2CM / Secure sensor*** section as shown in Figure 8

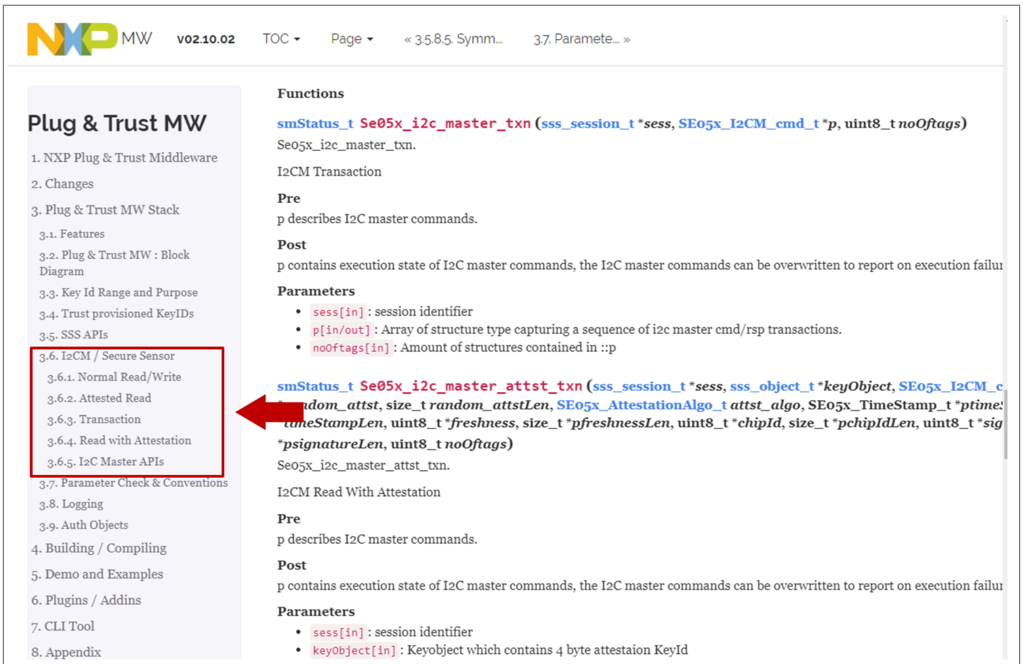


Figure 8. EdgeLock SE05x Plug & Trust Middleware I^{2}C master API documentation

On the other hand, if you are interested in the low level APDU commands to operate the I^{2}C master interface of EdgeLock SE05x, you can refer to AN12412 - SE050 APDU specification document.

4 Running the EdgeLock SE05x secure sensor demo

The EdgeLock SE05x Plug & Trust Middleware package includes a project example that demonstrates how to leverage EdgeLock SE05x to protect data from a security-sensitive sensor. In this demo, we use one OM-SE050ARD board and two FRDM-K64F boards. One of two FRDM-K64F is used as the host MCU towards the EdgeLock SE05x, the second FRDM-K64F board is used as an accelerometer sensor as shown in Figure 9.

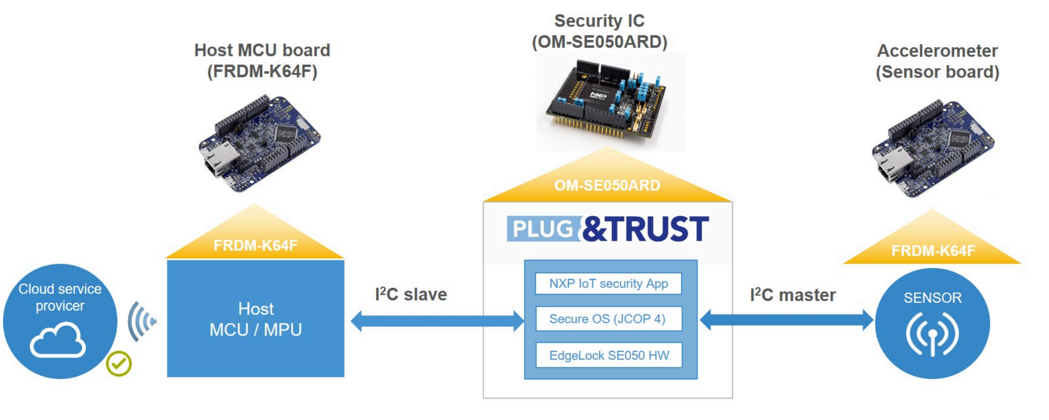


Figure 9. EdgeLock SE05x secure sensor demo setup

The steps required to run the project example that demonstrates how to read data from a sensor connected to EdgeLock SE05x security IC are:

1. Get the required hardware.
2. Download FRDM-K64F SDK.
3. Flash the software in the sensor board (FRDM-K64F).
4. Configure OM-SE050ARD jumper settings.
5. OM-SE050ARD connection with the host MCU board (FRDM-K64F).
6. OM-SE050ARD connection with the sensor board board (Accelerometer).
7. Import the project example in MCUXpresso workspace.
8. Build and run the secure sensor project example.

4.1 Hardware required

The hardware required to run the project example that demonstrates how to read data from a sensor connected to EdgeLock SE05x security IC is:

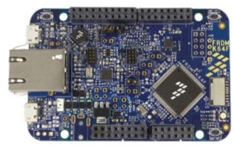
1. One OM-SE050ARD board

Table 3. OM-SE050ARD development kit details

Part number	12NC	Content	Picture
OM-SE050ARD	935383282598	EdgeLock SE050 development board	

2. Two FRDM-K64F boards

Table 4. FRDM-K64F details

Part number	12NC	Content	Picture
FRDM-64F	935326293598	Freedom development platform for Kinetis K64, K63 and K24 MCUs	

4.2 Download and install the FRDM-K64F SDK

The sensor data protection project example is included as part of the FRDM-K64F SDK. Install it to your workspace as shown in Figure 10

1. Download the FRDM-K64F SDK, publicly available from the NXP website.
2. Drag and drop the FRDM-K64F SDK zip file in the *Installed SDKs* section in the bottom part of the MCUXpresso IDE.
3. Check that the FRDM-K64F SDK is installed successfully.

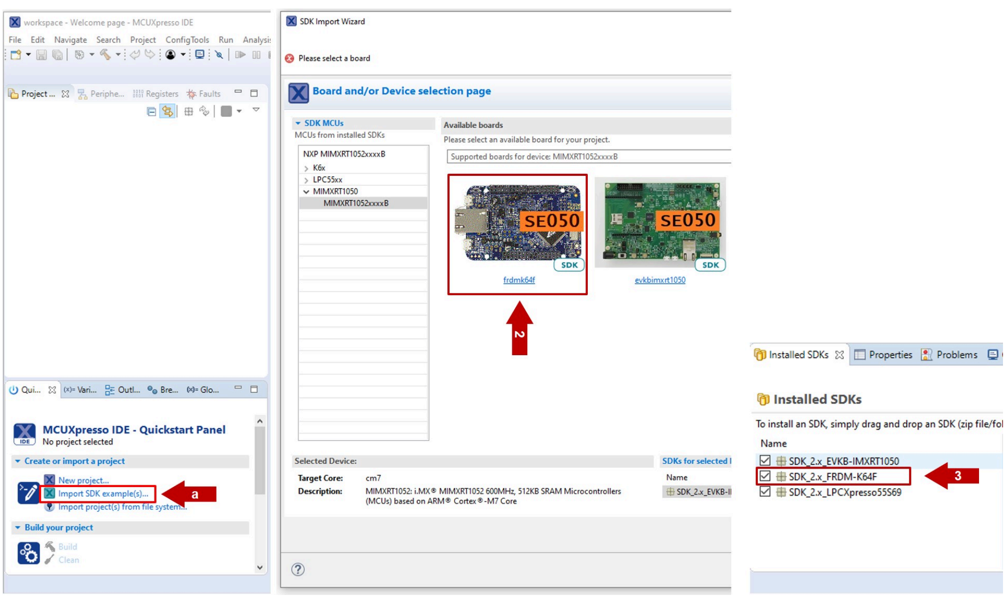


Figure 10. Import FRDM-K64F SDK version

Note: *For more detailed instructions on how to install it the FRDM-K64F into our MCUXpresso workspace, refer to AN12396 - Quick start guide with FRDM-K64F.*

4.3 Flash the software in the sensor board (Accelerometer)

We use the I^2C interface of the FRDM-K64F board to read out data from the accelerometer sensor. Therefore, we need to make sure that the FRDM-K64F is not running any firmware that may make use of the I^2C bus. The EdgeLock SE05x Plug & Trust Middleware includes a binary file we can flash into FRDM-K64F to make sure that the FRDM-K64F board behaves as expected for this demo. To flash this binary file, follow these steps:

1. Unplug and plug again the USB cable to the OpenSDA USB port as shown in Figure 9:

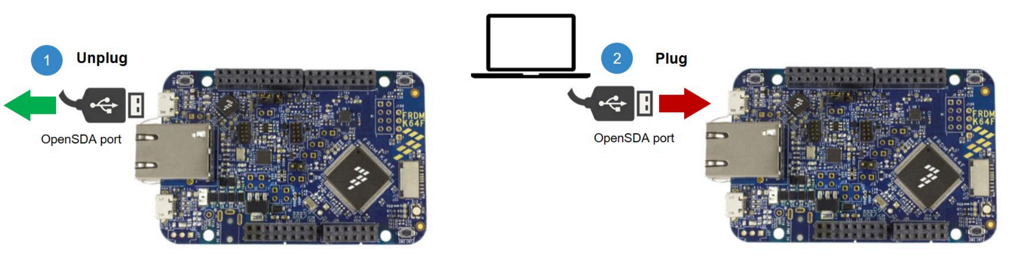


Figure 11. Unplug and plug OpenSDA port of FRDM-K64F

2. When you plug the board, your laptop should recognize the board as an external drive as shown Figure 12:

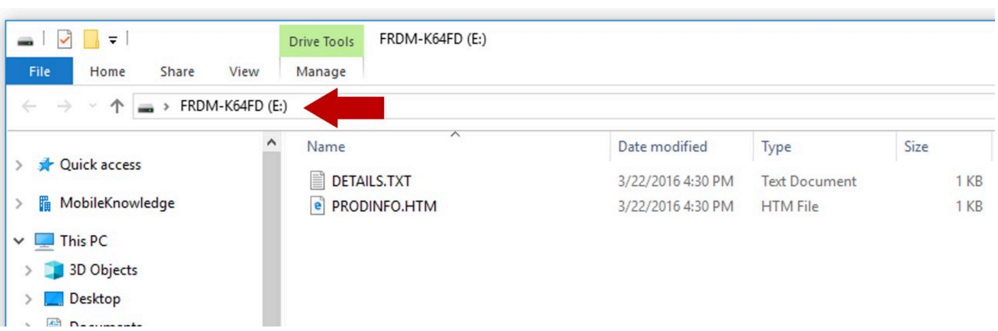


Figure 12. FRDM-K64F drive

3. Go the folder with the binary file to be flashed for this demo. This can be found in `simw-top\demos\se05x\ex_i2cMaster` folder of the EdgeLock SE05x Plug & Trust Middleware as show in Figure 13:

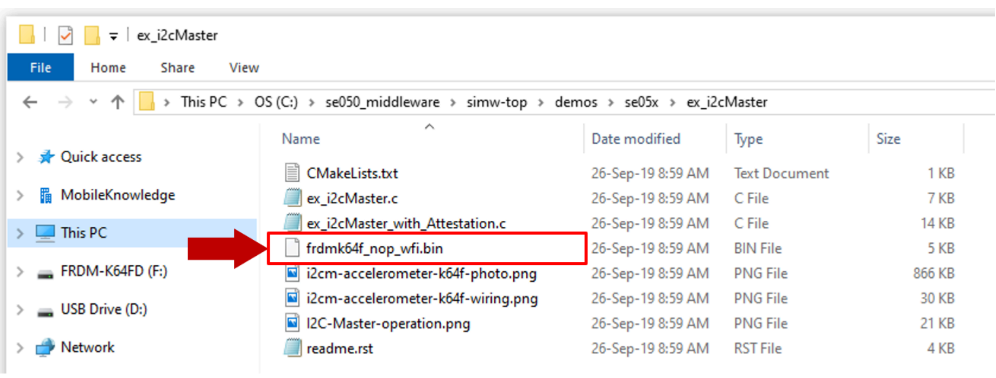


Figure 13. Find the binary file for the sensor board

4. Drag and drop or copy and paste the `frdmk64f_nop_wfi.bin` binary the FRDM-K64F drive from your computer file explorer as shown Figure 14:

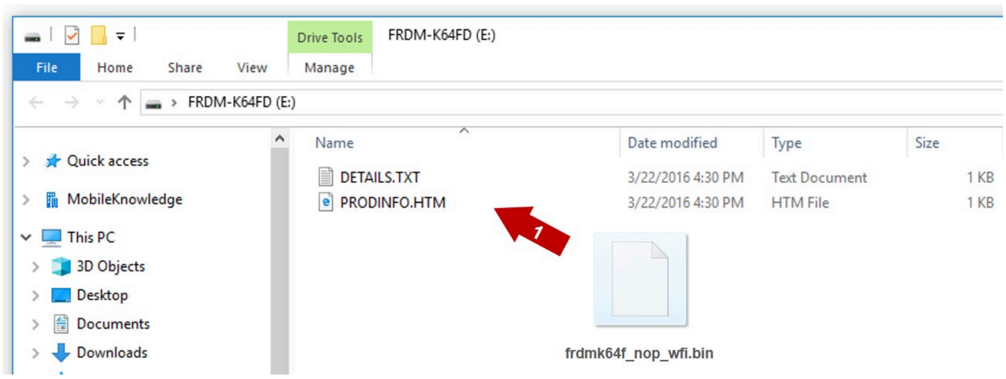


Figure 14. Drag and drop `frdmk64f_nop_wfi.bin` binary file for the sensor board

4.4 Configure OM-SE050ARD jumper settings

We use the Arduino headers to connect the host MCU board to the OM-SE050ARD and the J11 header to connect the sensor board in our setup. The jumper settings to enable the I^{2}C slave interface over the Arduino header and the I^{2}C master interface over the J11 header are shown in Figure 15:

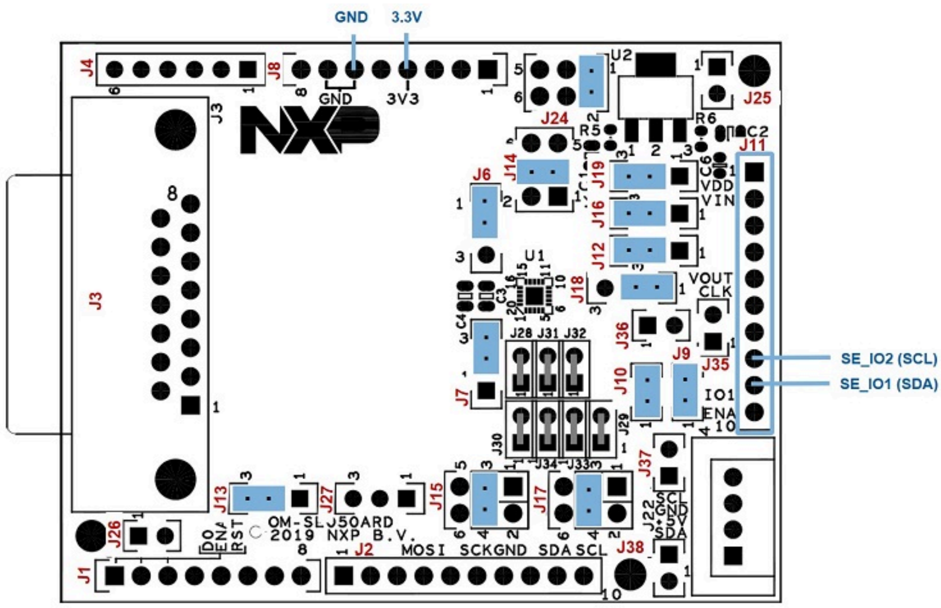


Figure 15. Demo OM-SE050ARD jumper configuration

Table 5 details the jumper settings for the configuration of the OM-SE050ARD I^{2}C master interface in J11 header.

Table 5. Jumper settings for EdgeLock SE05x in I²C master mode

Jumper	Configuration	Comment
J6	Set to 1-2 (Default)	Contactless operation disabled
J7	Set to 2-3 (Default)	Contactless operation disabled
J9, J10	**Set to "Closed"**	**Set to “Closed” to enable pull-up resistors for I²C master signals SE_IO1 and SE_IO2 *(if IOT sensor board not already provides pull-up resistors).***
J12	Set to 2-3 (Default)	SE_RST routed to ARD_RST on J1:3
J13	Set to 2-3 (Default)	SE_ENA set to ARD_ENA on J1:6
J14	Set to 1-2 (Default)	Routed to V_{DD} supply voltage (Default)
J15	Set to 3-4 (Default)	I²C_SDA routed to ARD_SDA_R3 (J2:9)
J16	Set to 2-3 (Default)	V_{DD} as SE_V_{IN}
J17	Set to 3-4 (Default)	I²C_SCL routed to ARD_SCL_R3 (J2:10)
J18	Set 1-2 (Default)	SE_IO2 to pin 9 of header J11
J19	Set to 2-3 (Default)	V_{DD}=3.3V supply voltage from Arduino-R3 voltages
J24	Set to 1-2 (Default)	No input LDO
J25, J26	Do not care	Dummy jumpers
J37, J38	Set to “Open” (Default)	3k3 pull-up resistor for I²C standard mode

4.5 OM-SE050ARD connection with the host MCU board (FRDM-K64F)

One of the FRDM-K64F boards is used as host MCU with the EdgeLock SE05x as a companion security IC attached to it. We can connect the OM-SE050ARD board on top of the FRDM-K64F board using the Arduino connectors available in both boards as shown in Figure 16.

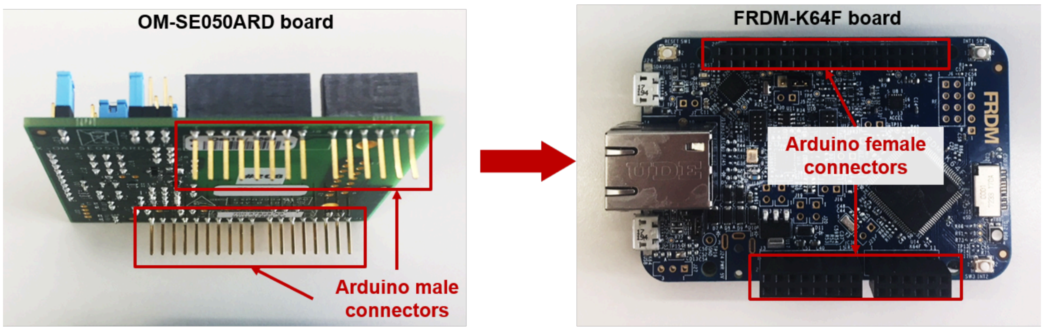


Figure 16. Mounting the OM-SE050ARD with the host MCU board

4.6 OM-SE050ARD connection with the sensor board (Accelerometer)

We use the accelerometer embedded in the FRDM-K64F as the sensor for this demo. As such, we need to connect the second FRDM-K64F board to the I^{2}C master interface of the OM-SE050ARD. For that, we wire the FRDM-K64F and OM-SE050ARD boards as shown in Figure 17

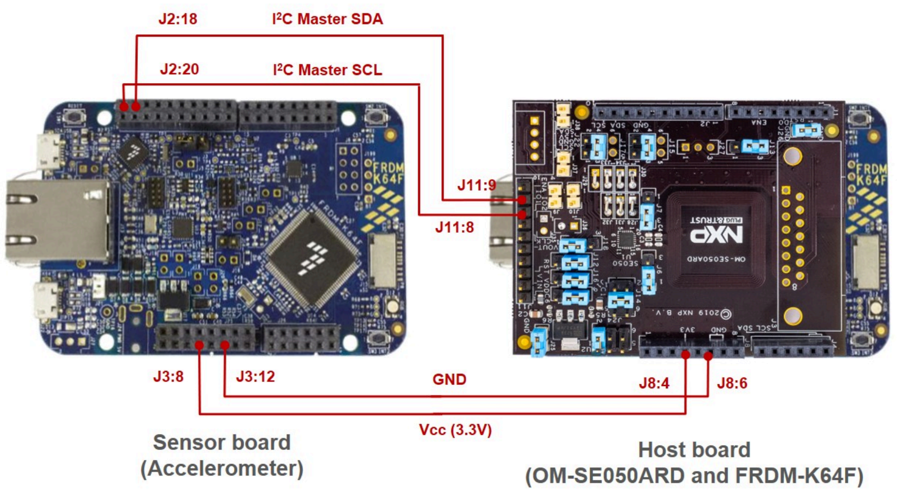


Figure 17. OM-SE050ARD connection with the sensor board

Table 6 shows the detailed connection between the OM-SE050ARD and the FRDM-K64F acting as a sensor.

Table 6. List of necessary connections between boards

Description	OM-SE050ARD board	FRDM-K64F (Sensor
I^{2}C Master SDA	J11:9	J2:18
I^{2}C Master SCL	J11:8	J2:20
Vcc (3.3V)	J8:4	J3:8
GND	J8:6	J3:12

4.7 Import the project example in MCUXpresso workspace

The FRDM-K64F SDK includes two source project examples called `ex_i2cMaster` and `ex_i2cMaster_with_Attestation` that read the accelerometer data via the EdgeLock SE05x I²C master interface with or without data attestation respectively.

Import any of the two to your MCUXpresso workspace as shown in Figure 18:

1. Click *Import SDK examples* from the MCUXpresso IDE quick start panel.
2. Select `se_hostlib_se05x_ex_i2cMaster` project example and click the *Finish* button (or alternatively, select `se_hostlib_se05x_ex_i2cMaster_with_Attestation`).
3. Check the project is now visible in your MCUXpresso workspace

Note: *For detailed instructions on how to import project examples from FRDM-K64F SDK, check* AN12396 - Quick start guide with Kinetis K64F

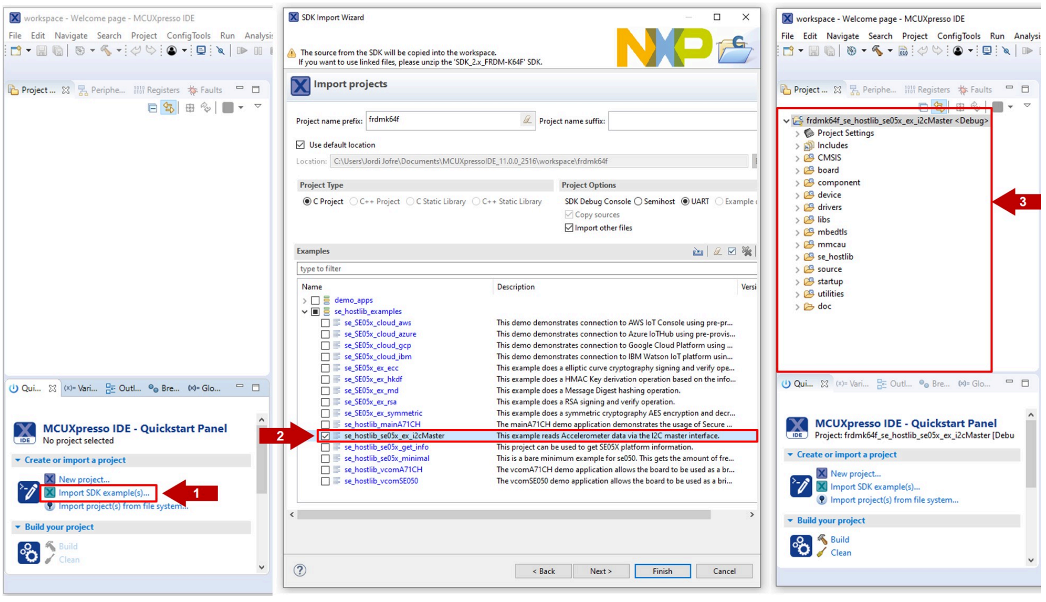


Figure 18. Import sensor data protection project example

4.8 Build and run the secure sensor project example

The EdgeLock SE05x Plug & Trust Middleware includes two source code examples called `ex_i2cMaster_with_Attestation` and `ex_i2cMaster` that read the accelerometer data via the I²C master interface with or without data attestation respectively. To configure the `cmake_project_frdmk64f` project to execute these source code examples, follow these steps:

1. Connect the Host board (OM-SE050ARD and FRDM-K64F) the computer through the OpenSDA debug USB port as shown in Figure 19.

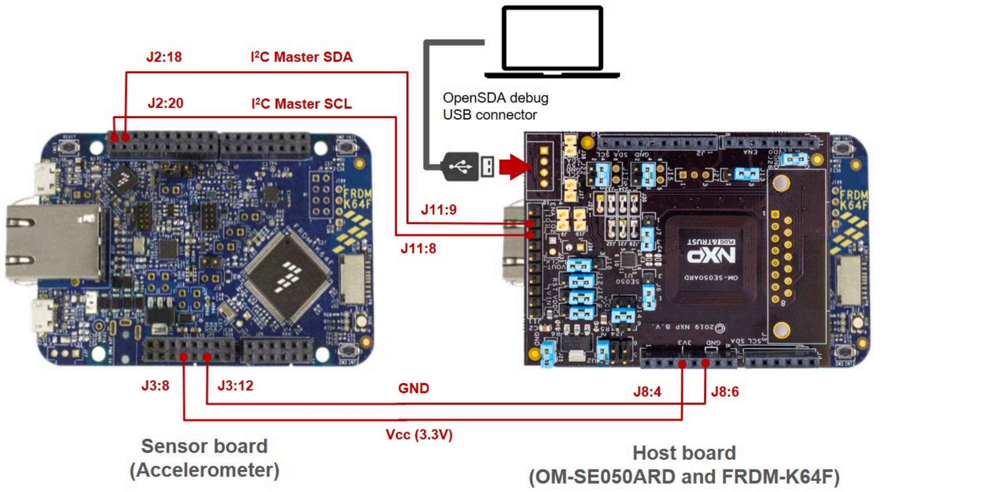


Figure 19. Connect the Host board to the computer.

2. Open TeraTerm and establish a serial port connection, as shown in Figure 20. When we execute the project, the data read from the accelerometer will be displayed in this TeraTerm serial port.

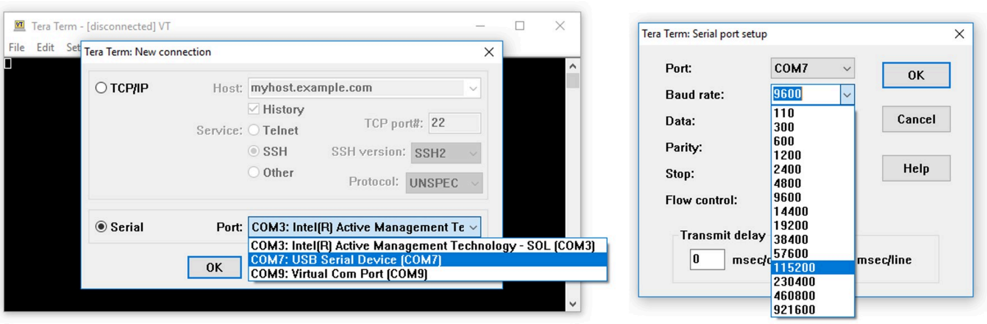


Figure 20. Configure Tera Term.

3. Go to the MCUXpresso Quickstart Panel and click *Debug* button, wait a few seconds until the project executes and click on Resume to allow the software to continue its execution as shown in Figure 21

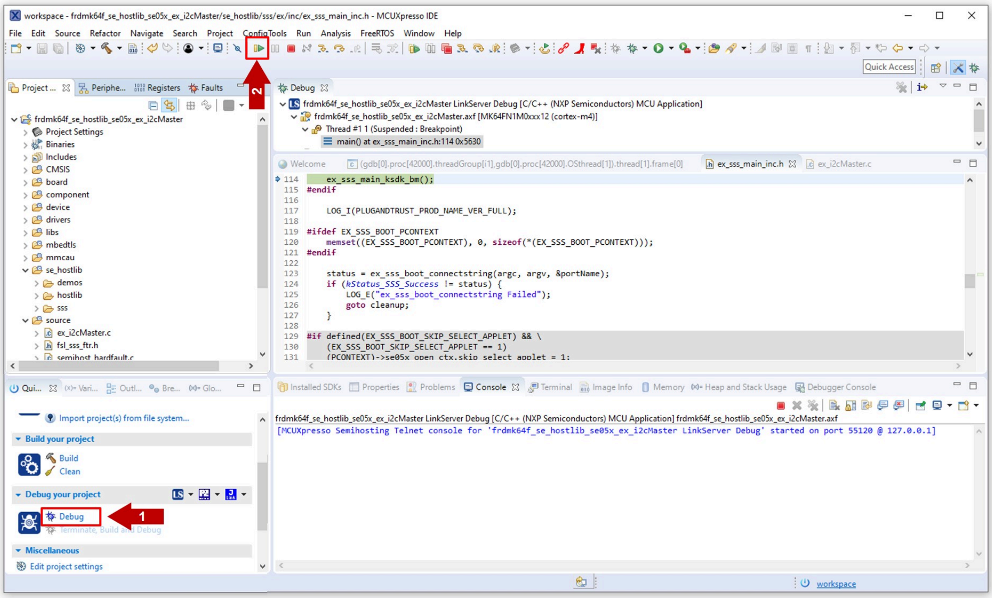


Figure 21. Run sensor data protection project example

4. The accelerometer data is now displayed in the TeraTerm window. Move the FRDM-K64F board acting as a sensor in different directions to observe how the accelerometer coordinates change in the TeraTerm window, as shown in Figure 22

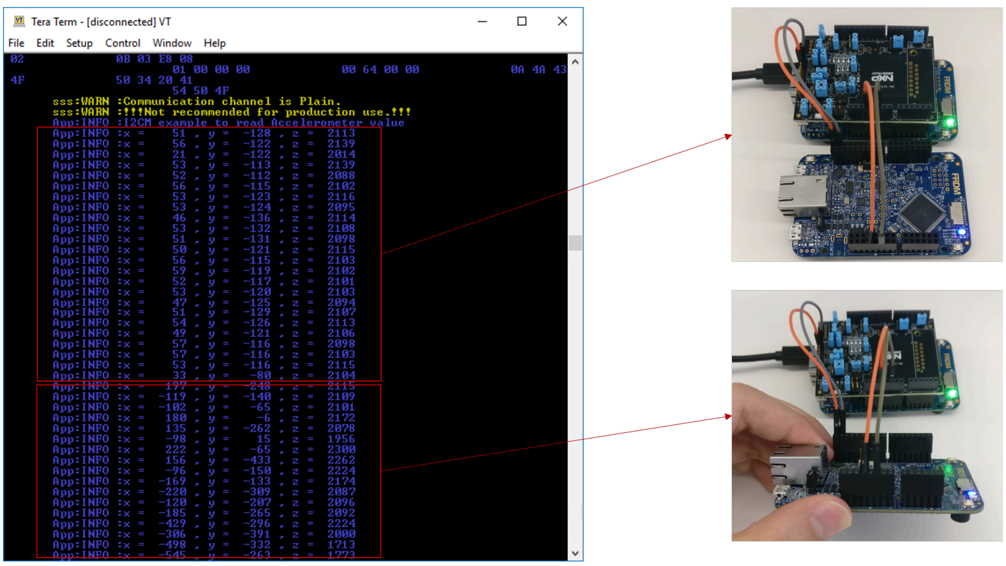


Figure 22. Tera Term logs reflect the change in the orientation of the Sensor FRDM-K64F.

5 Legal information

5.1 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.

5.2 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.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

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.

5.3 Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

Contents

Date of release: 6 July 2021
Document identifier: AN12449
Document number: 546813