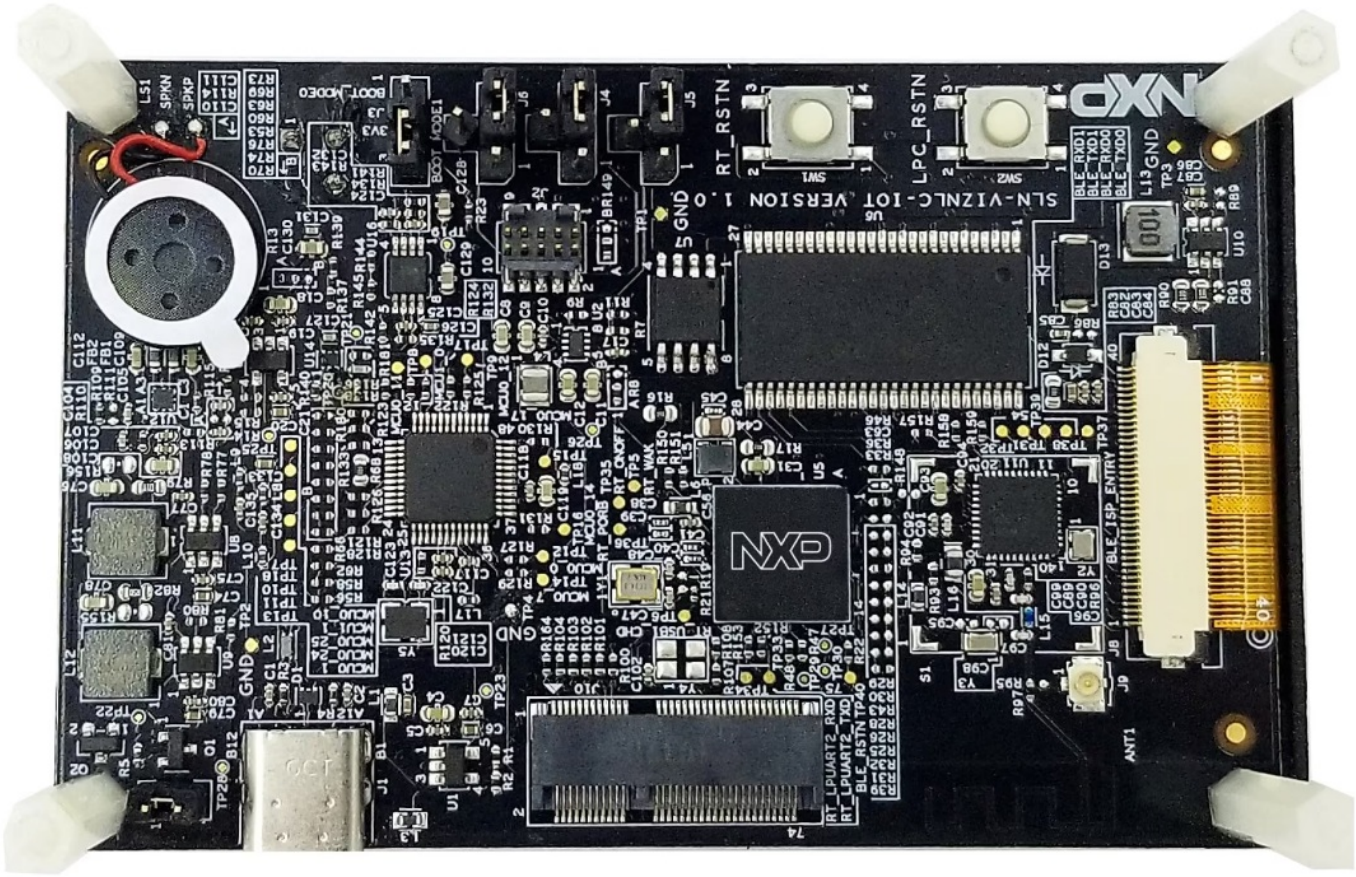
NXP SLN-VIZNLC-IOT Smart Lock User Guide

Contents

NXP SLN-VIZNLC-IOT Smart Lock



Product Information

SLN-VIZNLC-IOT-GSG

The SLN-VIZNLC-IOT-GSG is a Smart Lock IoT board manufactured by NXP Semiconductors. This board comes with an RGB and IR camera that streams video directly to the onboard TFT screen. It also has a GUI overlay that provides information such as registration status and countdown timers. With this board, you can register faces in the local face database of the kit and use the recognition feature to unlock the smart lock.

Product Usage Instructions

Follow the steps below to get started with your SLN-VIZNLC-IOT-GSG board:

1. Unbox the kit and remove the protective film from the RGB and IR camera.
2. Put a jumper on J3 (pins 2 and 3 on the left side) as shown in Figure 35.
3. Take the USB-A > USB-C cable provided inside the kit. Plug the USB-A end into the USB port on your computer and the USB-C end into the USB port of the kit. The USB connector of the kit supplies power to the board and supports data transfer capabilities for Mass Storage Device (MSD) programming and virtual serial port communication.
4. Power on the SLN-VIZNLC-IOT kit. Once the application is ready, your computer detects a new USB COM device and automatically installs the required drivers. A message confirms when the installation is completed.
5. To register a new face, press the Manual Registration button (SW4) on the kit. The guidelines help you align your face correctly during registration, and the countdown timer bar indicates how much longer it takes until the registration process times out and fails.

Document information

Information	Content
Keywords	SLN-VIZNLC-IOT-GSG, Smart Lock, IoT
Abstract	This guide walks you through the process of getting up and running with your SLN-VIZNLC-IOT board

Plug it in

- Welcome to the SLN-VIZNLC-IOT Getting Started Guide.
- This guide walks you through the process of getting up and running with your SLN-VIZNLC-IOT board. This guide takes you through the steps of unboxing your kit, running the out-of-box smart lock demo application, as well as downloading, modifying, and debugging the firmware source code for your kit.
- Before we begin, make sure to check the box your kit came in for any marks or other damages, and should you find anything, be sure to report it to your local NXP representative.

Smart Lock

Unbox

The box your kit arrives in should contain a few different things, including:

- A packing list paper
- Fully assembled VIZNLC kit
- USB-A > USB-C Cable (x1)
- Jumpers (x2)

Figure 1. Items inside the box

Power on

- Before we begin, put a jumper on J3 (pin 2 and 3 on the left side) as shown in the yellow highlighted box in Figure 35. Then remove the protective film from the RGB and IR camera as shown in Figure 2. This protective film is used to protect the lens of each camera during transport. However, failure to remove may cause the image capture not to work correctly.

Figure 2. Protective film covering the two cameras: RGB and IR

To get started, follow the steps below:

1. Take the USB-A > USB-C cable provided inside the kit. Plug the USB-A end into USB port on your computer and the USB-C end into the USB port of the kit. The USB connector of the kit supplies power to the board and supports data transfer capabilities for Mass Storage Device (MSD) programming and virtual serial port communication.

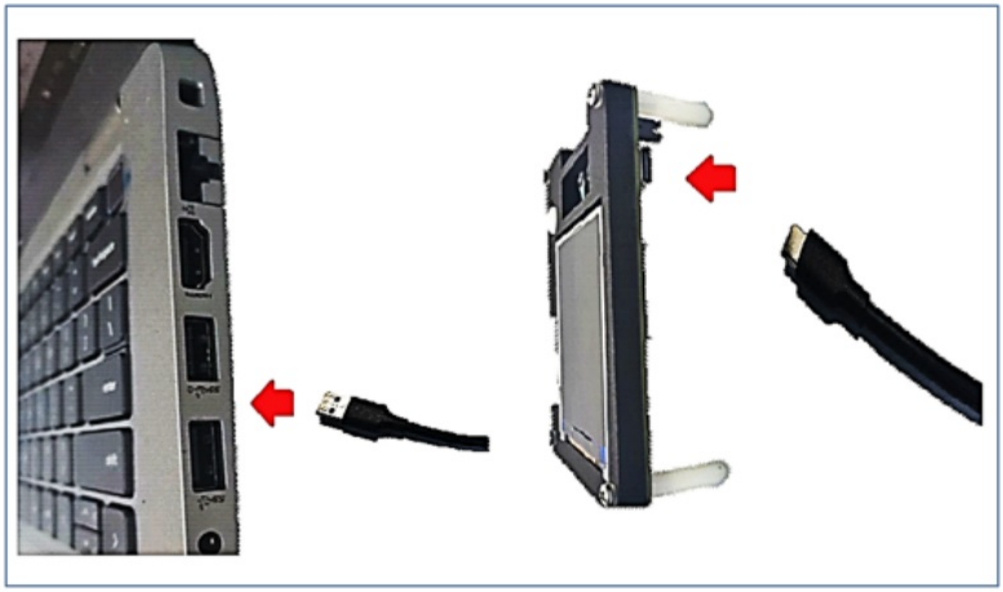


Figure 3. Power on the SLN-VIZNLC-IOT kit

2. Once the application is ready, your computer detects a new USB COM device, and automatically installs the required drivers. A message confirms when the installation is completed.
3. After powering on, the onboard TFT screen streams video directly from the RGB camera alongside a GUI overlay, providing information such as:
    - Locked/Unlocked status whether a face is recognized
    - Current App Type (Smart Lock/Access)
    - ON/OFF status of Wi-Fi and Bluetooth LE
    - Number of registered users

Figure 4. The screen with video preview

Register and recognize a face

- Let us get started with a demonstration of the out-of-box features of this application.
- To use the recognition feature of the SLN-VIZNLC-IOT, a face must be registered in the local face database of the kit. To begin registering a new face, press the Manual Registration button (SW4) on the kit.

Figure 5. Manual Registration button (SW4)

- Once pressed, a message indicating registration is taking place pops up at the top of the screen. The speaker plays an audio message confirming that the registration has started.

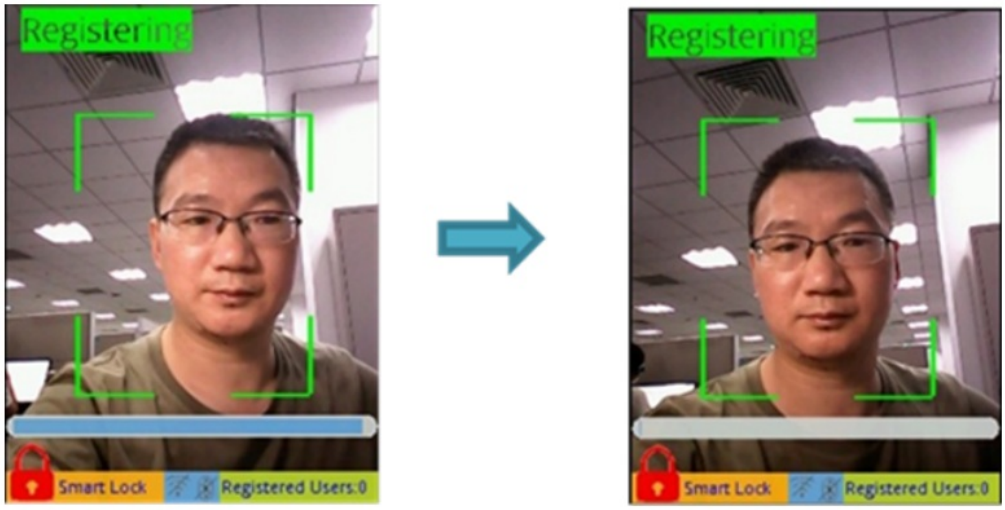


Figure 6. Registering screen

- Figure 6 shows that while registration is taking place, the GUI displays:
- A “Registering” message
- Face alignment guidelines
- A countdown timer bar
- The guidelines help you align your face correctly during registration, and the countdown timer bar indicates how much longer it takes until the registration process times out and fails.

Figure 7. Registration Timeout

- As an additional measure to help with registering your face, the kit even plays a warning audio prompt if too much of the side of your face is exposed during the registration process, saying “Look at Camera”. The kit plays this warning audio until your face is properly pointed toward the camera.
- Should your face fails to register, simply press the SW4 button again to retry.
- Once your face is successfully registered, the kit displays the message “Registration Successful”, and a unique identifier is assigned to your face. The number of registered users is also updated automatically.

Figure 8. Registration Successful

- Once registered, the kit displays the message “Recognition Successful” and plays a corresponding audio file when a recognized face is detected.

Figure 9. Recognition Successful

Liveness detection and anti-spoofing

- The Liveness detection and anti-spoofing features of the SLN-VIZNLC-IOT are switched ON by default. Therefore, enabling the system to distinguish between your actual face and a printout or phone display image of your face.
- This feature helps to defend against some of the most frequent face recognition “spoof” attacks.
- One such spoof attack is when a malicious actor uses a picture of someone to gain access to their face-protected materials. The malicious actor does this spoof by requiring an actual face of the user to unlock the system rather than simply a picture of their face.

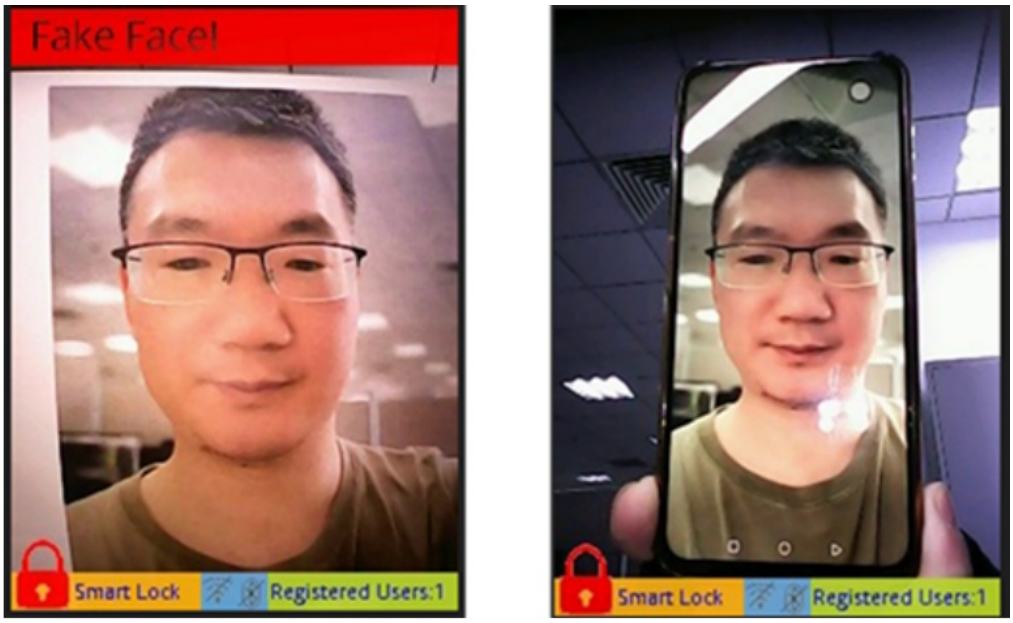


Figure 10. Printed picture and phone display spoof attack

- As shown in Figure 10 using a phone display or a printed picture of a face does not trigger the “Recognition Successful” message.

Connect to serial CLI

- The Smart Lock software installed by default on the SLN-VIZNLC-IOT kit provides a convenient serial-based CLI. This CLI is used to retrieve useful runtime data and configure various application settings. Connecting to the serial-based CLI of the kit can be done using a serial terminal emulator program like PuTTY or Tera Term.
- Before we begin, make sure that you have a serial terminal emulator like PuTTY or Tera Term installed on your computer.

Figure 11. PuTTY and Tera Term

- **Note:** If you are using a Windows machine, we recommend Tera Term for its ability to discover connected COM devices automatically. Additionally, if there is a disconnect, Tera Term reconnects to a device.
- Establish a serial connection with your device by entering the serial settings as shown in Figure 12. Ensure to replace the COM port setting with the COM port associated with your kit.

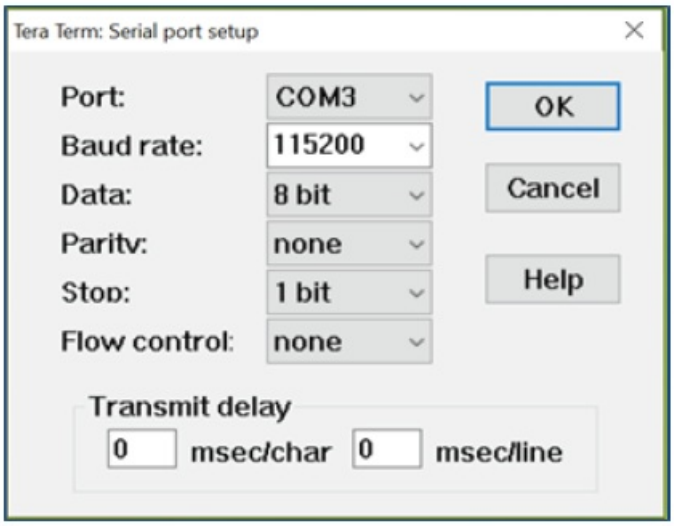


Figure 12. Serial port setup

- Once connected, a blank terminal screen appears that echoes back any characters that you type. Typing the help command prints a list of all available commands and a brief description of their functionalities.

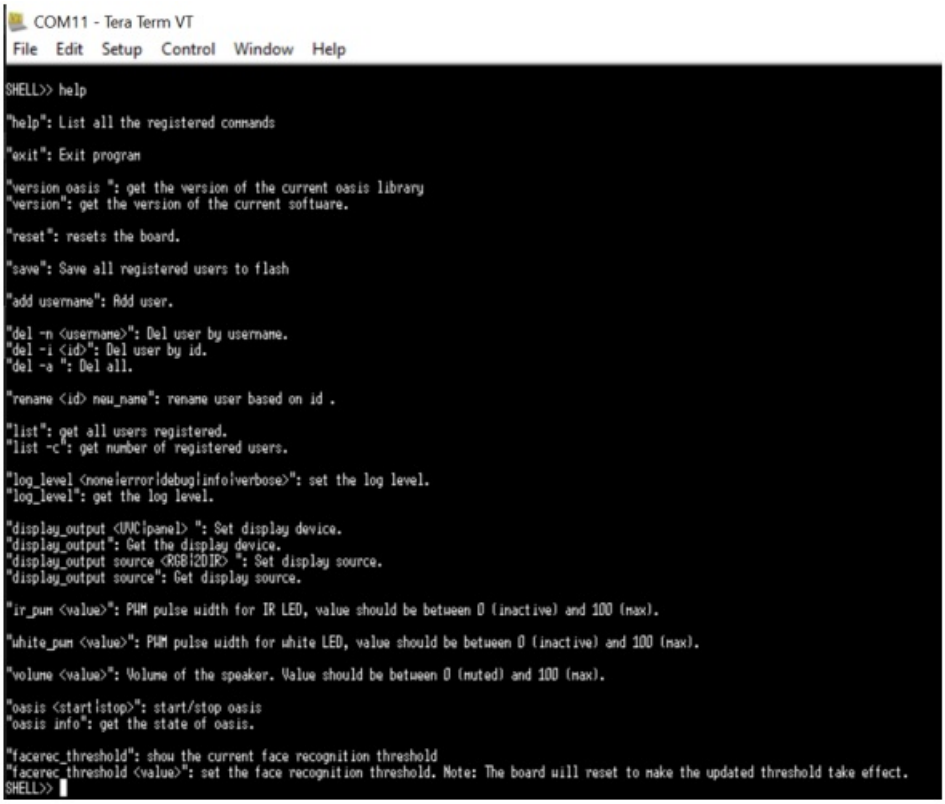


Figure 13. Help menu

Build and run

Getting MCUXpresso IDE

- The MCUXpresso IDE brings developers an easy-to-use eclipse-based development environment for NXP MCUs based on Arm Cortex-M cores, including its general-purpose crossover and wireless-enabled MCUs.
- The MCUXpresso IDE offers advanced editing, compiling, and debugging features. It also offers MCU-specific debugging views, code trace and profiling, multicore debugging, and integrated configuration tools.

To download MCUXpresso IDE, follow the steps below:

1. Go to MCUXpresso IDE homepage and click the Downloads button.

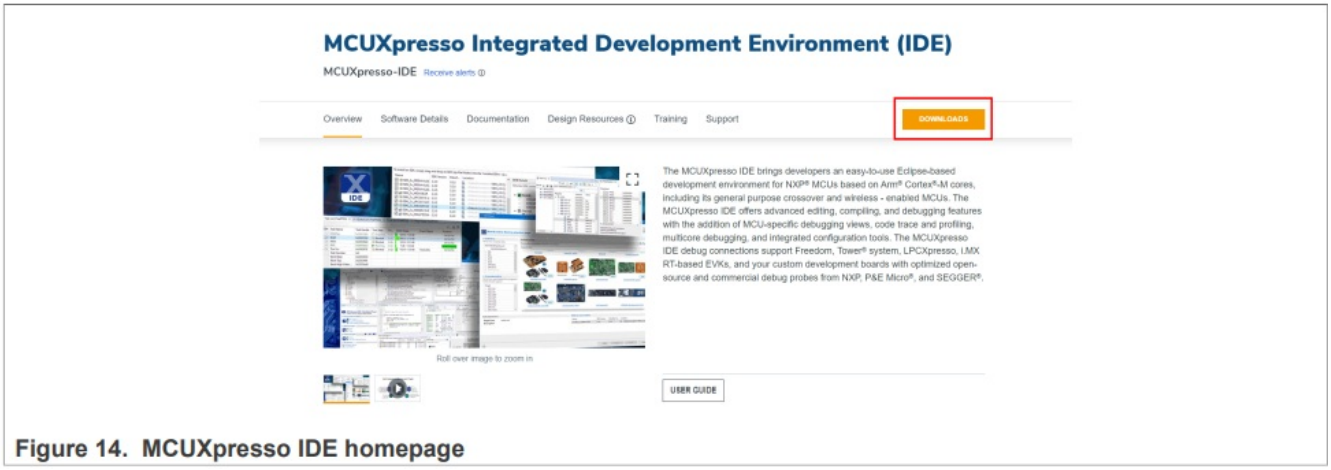


Figure 14. MCUXpresso IDE homepage

2. The Downloads page appears. Next, click the Download button.

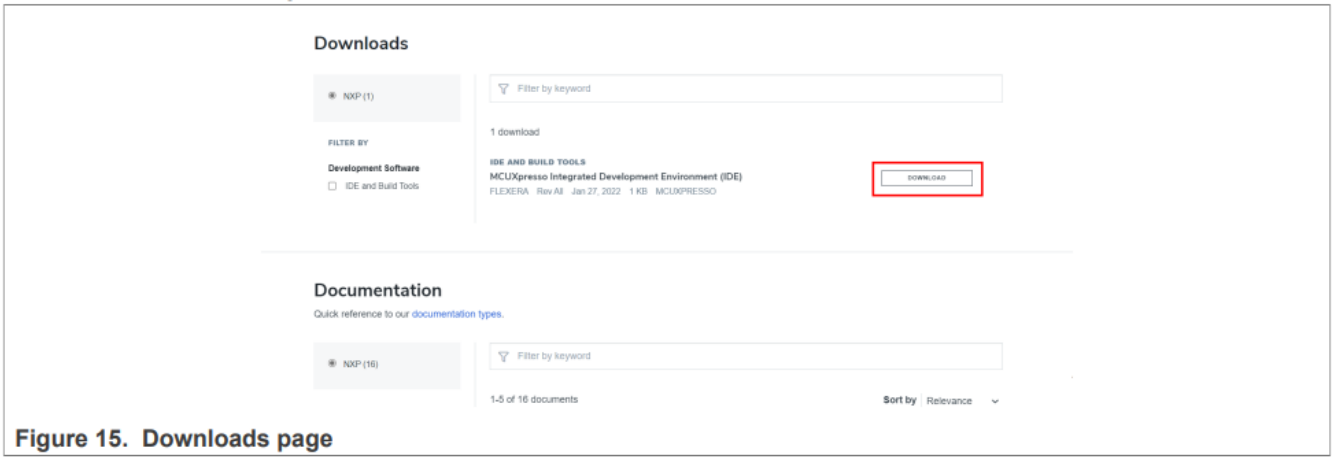


Figure 15. Downloads page

3. The Sign In page appears. Log in with your credentials.

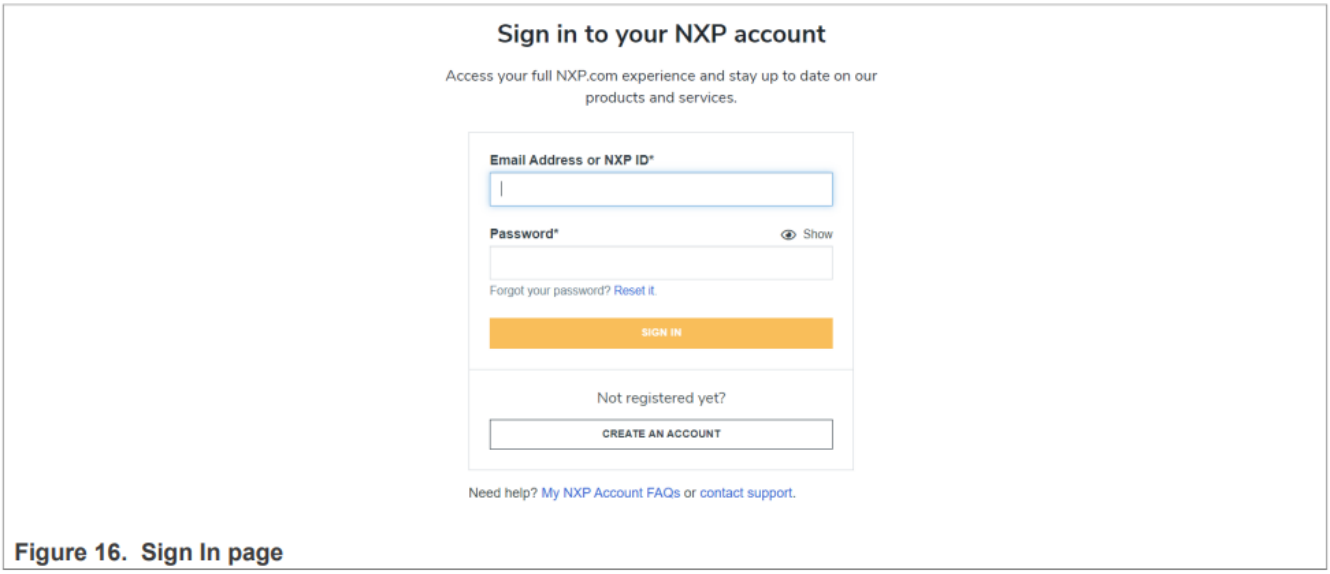


Figure 16. Sign In page

4. Once you have signed in, select Version 11.6.1 or newer from the list.

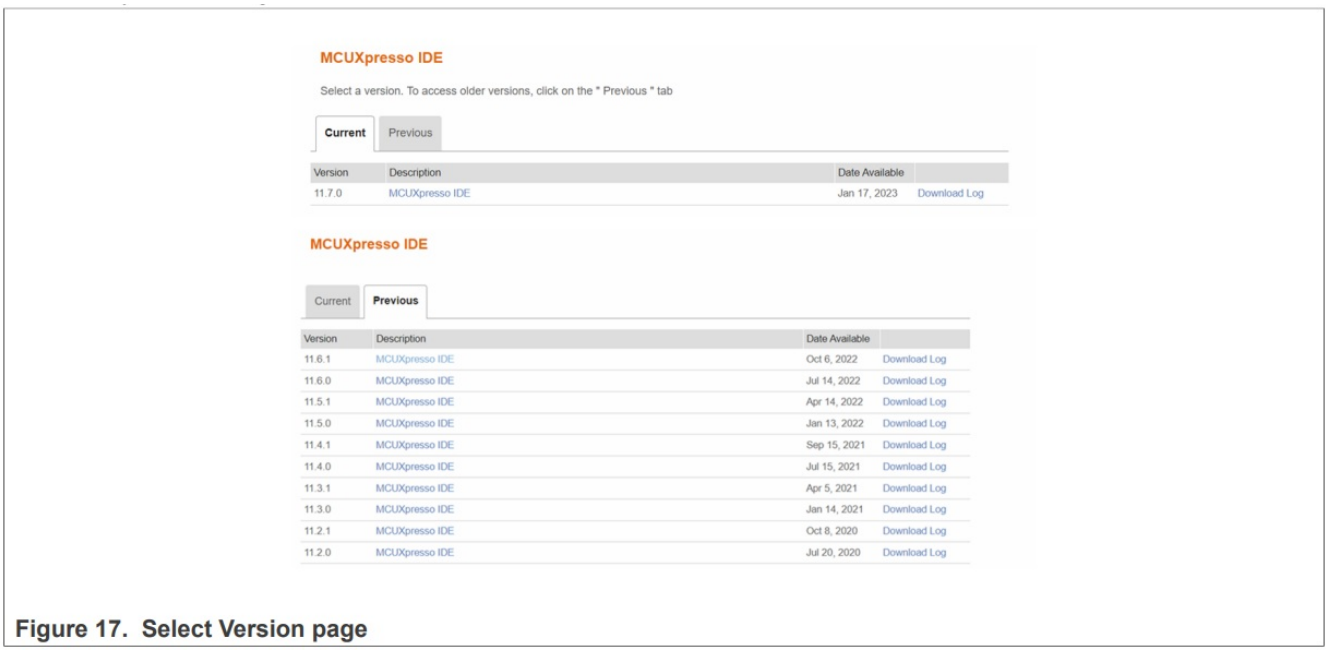


Figure 17. Select Version page

5. The Software Terms and Conditions page appears. Read the conditions and click the I Agree button.

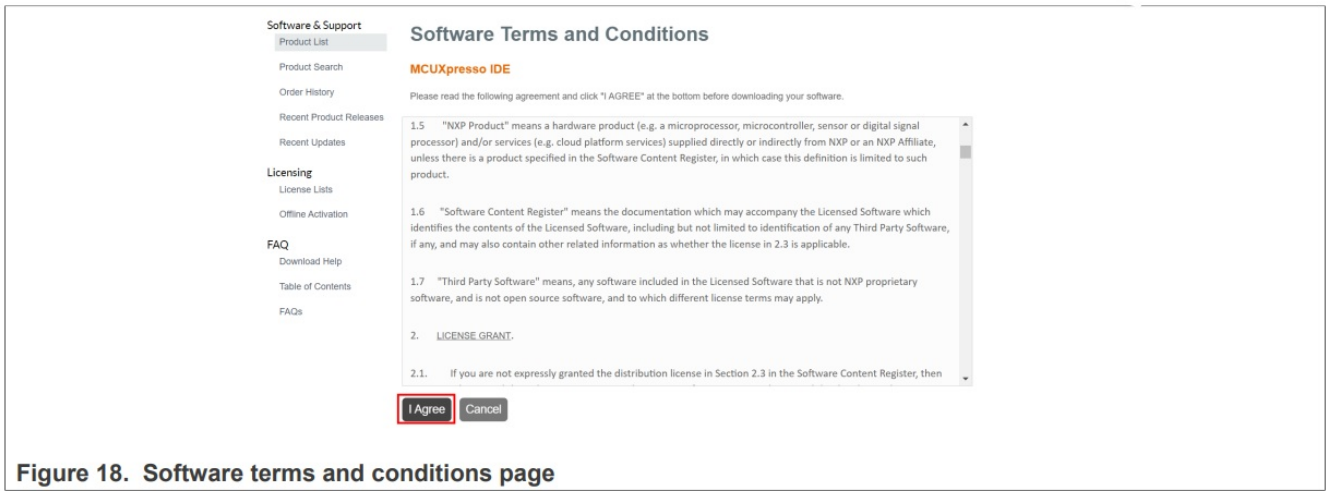


Figure 18. Software terms and conditions page

6. The Product Download page appears from where you can download the MCUXpresso IDE. Download the

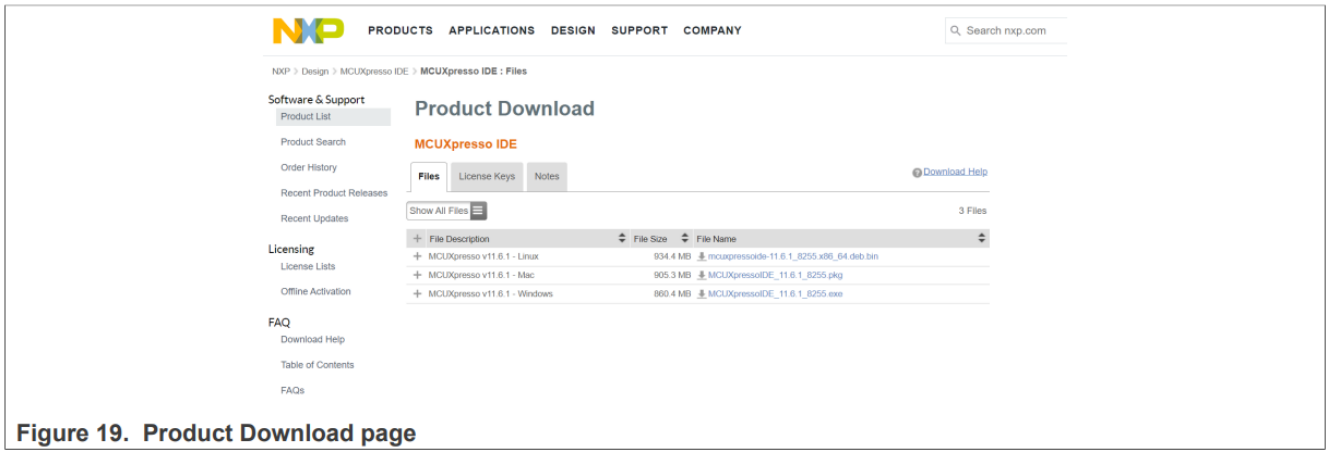


Figure 19. Product Download page

7. Open the downloaded application and follow the instructions found in the installer.

Installing the SDK

- MCUXpresso SDK is a comprehensive software enablement package designed to simplify and accelerate application development with NXP microcontrollers based on Arm Cortex-M cores. The MCUXpresso SDK includes production-grade software with integrated RTOS (optional), stacks and middleware, reference software, and more.

- It is available in custom downloads based on user selections of MCU, evaluation board, and optional software components.
- Before building the SLN-VIZNLC-IOT SDK example projects, the target SDKs (EVK-MIMXRT1060 and K32W061DK6) must be imported into MCUXpresso IDE.
- However, no need to import LPC845 as it is already preinstalled in the IDE.

To install MCUXpresso SDK, follow the steps below:

1. Launch the MCUXpresso SDK. The MCUXpresso SDK Builder welcome screen appears when the application is launched, as shown in Figure 20.

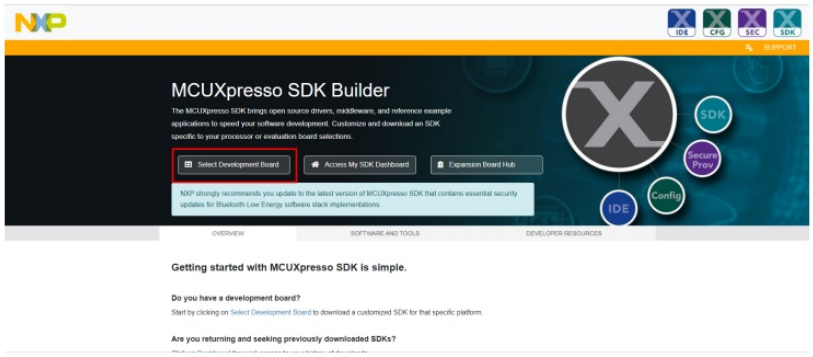


Figure 20. SDK Builder welcome screen

2. Click Select Development Board. After signing in, Select Development Board page appears.

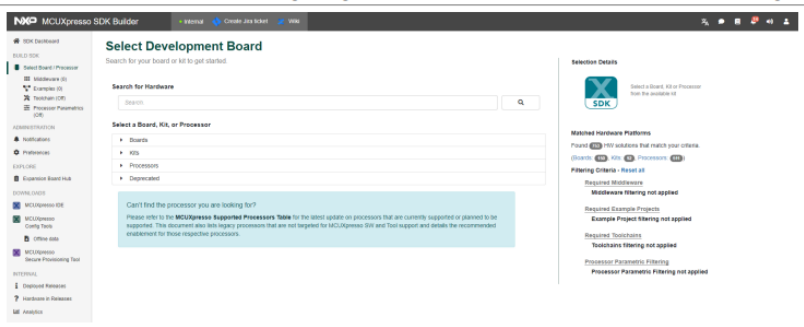


Figure 21. Select Development Board page

3. Search 1060 and select EVK-MIMXRT1060 board. Then click Build MCUXpresso SDK V2.12.1.

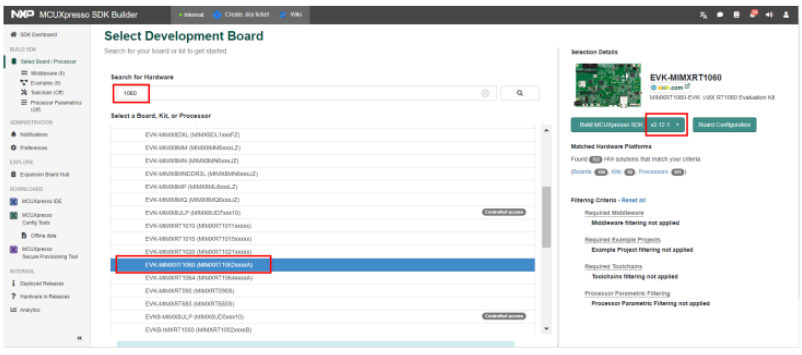


Figure 22. Select EVK-MIMXRT1060 SDK

4. To build SDK, select your Host OS and Toolchain / IDE, and all other necessary SDK components. Then click Download SDK button.

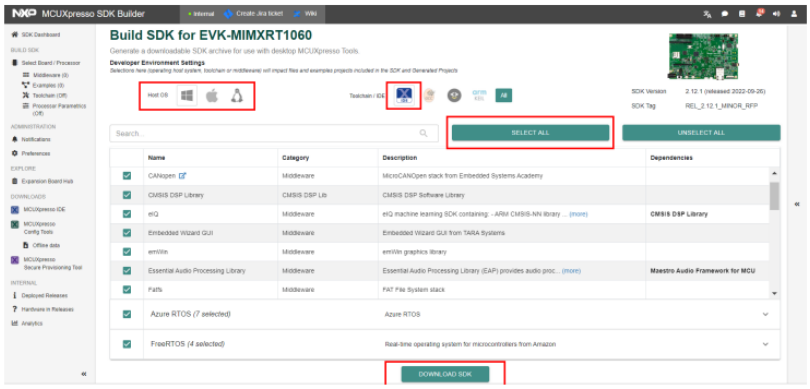


Figure 23. Build SDK and download

5. The dashboard page shows the built SDK. Click Download to download the SDK archive to your PC.

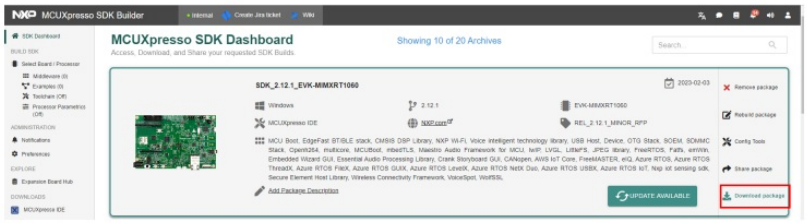


Figure 24. Downloading SDK archive

6. To install, drag and drop the SDK archive file to the IDE Installed SDKs view.

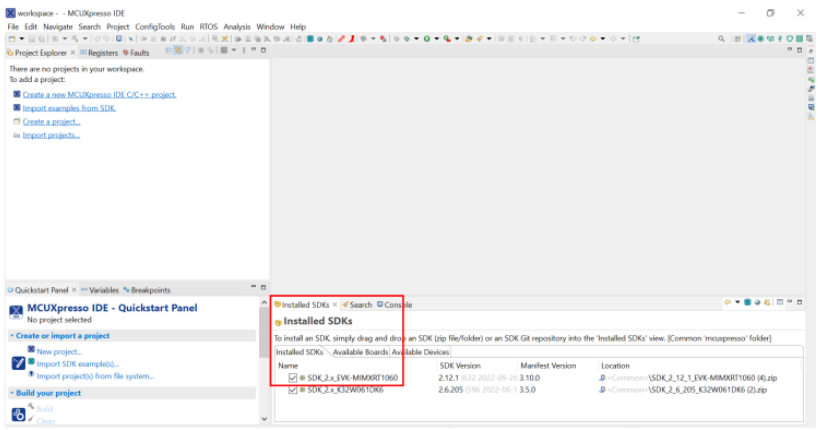


Figure 25. Install SDK

7. Download and install K32W061DK6 SDK V2.6.205 in the same way.

Downloading SLN-VIZNLC-IOT projects

- The SLN-VIZNLC-IOT out-of-box projects are published under the NXP GitHub page. You can either clone the repository using Git or download a zip folder containing the source code from mcu-viznlc.

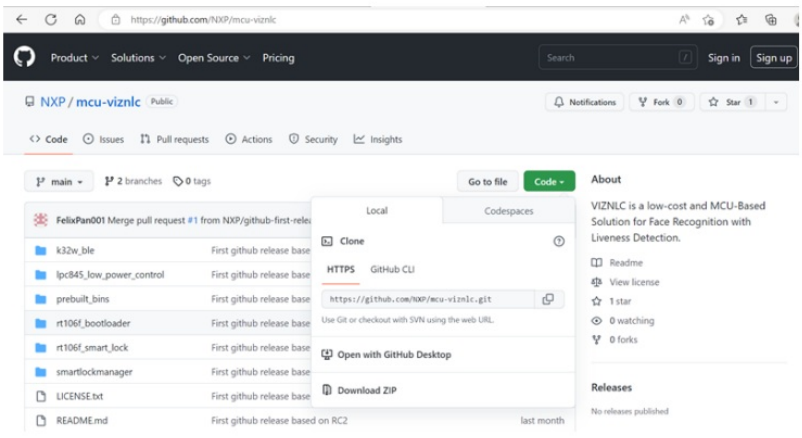


Figure 26. SLN-VIZNLC-IOT GitHub repository

- **Note:** If downloading a zipped archive, ensure to unzip this folder before proceeding to the next step.

Importing SLN-VIZNLC-IOT projects

To import the projects we downloaded into the IDE, follow the steps below:

1. Click Import project(s) from the file system…, then Browse your project path where you unzipped the sln_viznlc_iot source code, and click Next. sln_viznlc_iot source code, and click Next.

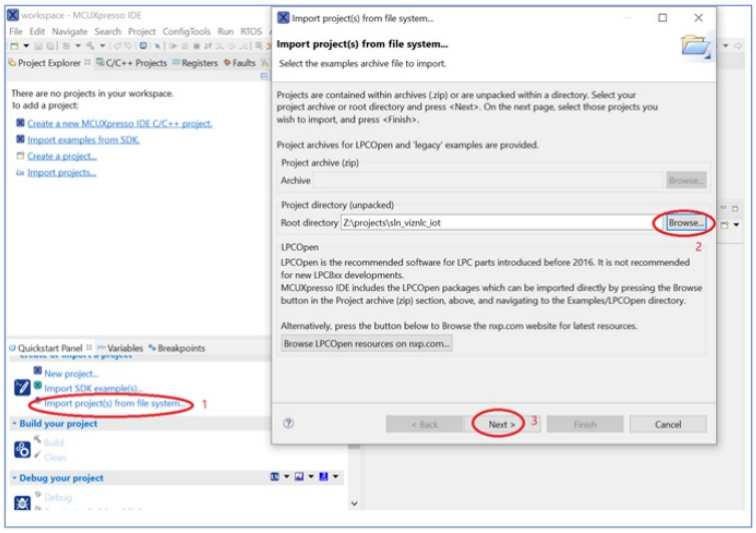


Figure 27. Import project(s) from file system

2. Import the files specified in Figure 28 from the displayed screen.

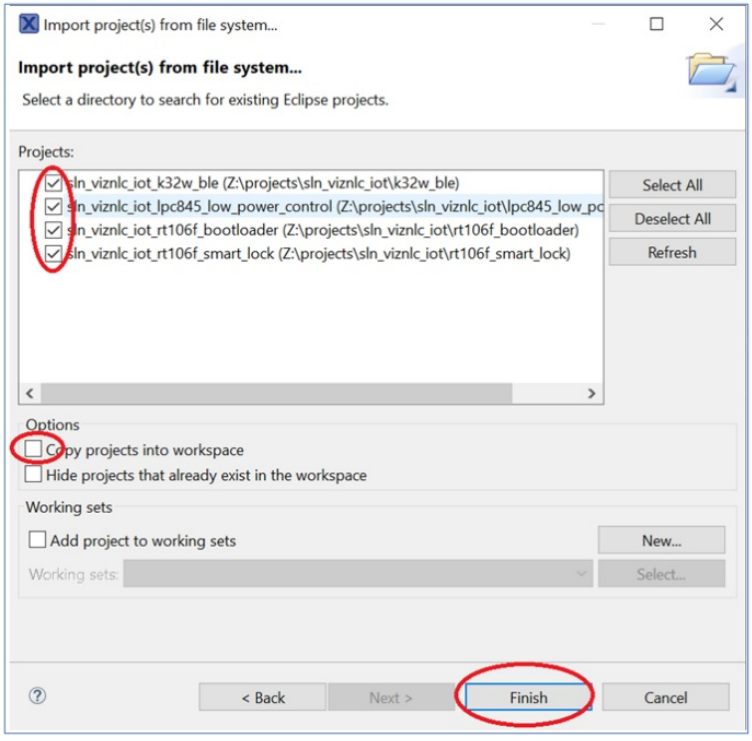


Figure 28. Import project(s) from file system

3. Once successfully imported, you should see projects open in the Project Explorer pane on the left side of the IDE.

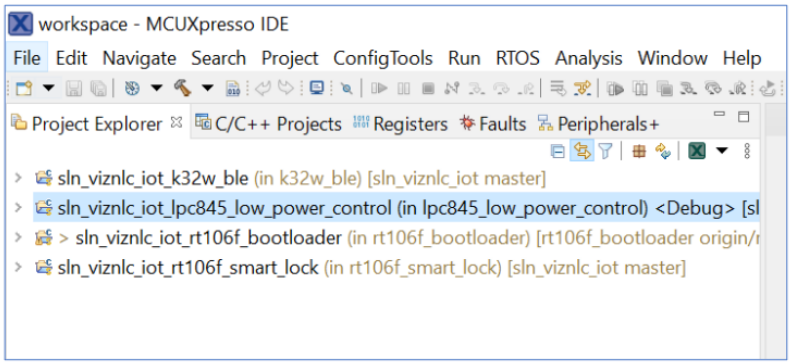


Figure 29. SLN-VIZNLC-IOT projects

Building the SLN-VIZNLC-IOT projects

- The SLN-VIZNLC-IOT SDK allows you to build the smart lock application directly. The application is made up of four subprojects:
- The lpc845_low_power_control project manages the power control of the system. LPC845 works as the host MCU. PIR sensor activates the host MCU and powers the RT106F part.
- The k32w_ble project implements the feature of Bluetooth LE module (UART over Bluetooth LE).
- The rt106f_bootloader and rt106f_smart_lock projects are the out-of-box applications that we used earlier to demonstrate the SLN-VIZNLC-IOT face recognition capabilities. The bootloader manages to jump into the smart lock application.
- These above applications are flashed onto your SLN-VIZNLC-IOT kit by default.
- In the Project Explorer pane, select the project file and navigate to the QuickStart Panel. To start the compilation and linking of this application, click the Build option.

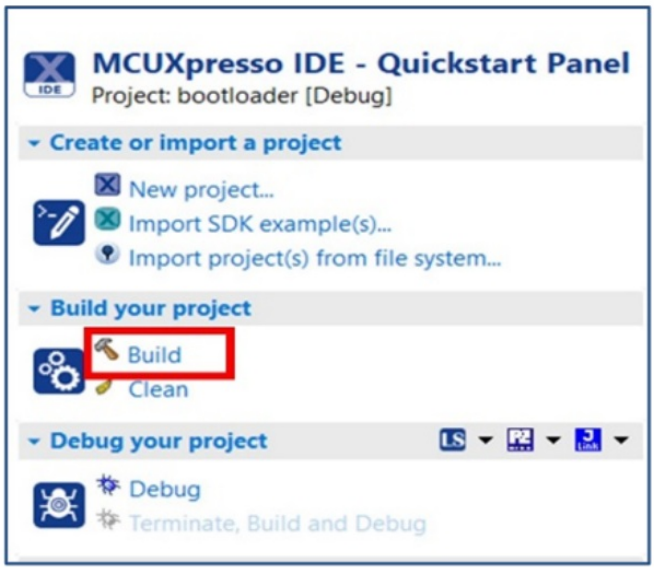


Figure 30. How to build

- The building may take a few minutes to complete but do not worry, as it is normal for applications of this size. Once finished, a message as shown in Figure 31 appears at the bottom of the IDE. Consider the rt106f_smart_lock project as an example.

```
Building target: sln_viznlc_iot_rt106f_smart_lock.axf
Invoking: MCU C++ Linker
arm-none-eabi-c++ -nostdlib -L"Z:\projects\sln_viznlc_iot\rt106f_smart_lock\libs\oasis_2d" -L"Z:\prc
Memory region         Used Size  Region Size  %age Used
     BOARD_FLASH:     4025876 B         8 MB     47.99%
     BOARD_SDRAM:    10879844 B        13 MB     79.81%
    SRAM_DTC_cm7:       14100 B       256 KB      5.38%
    SRAM_ITC_cm7:        5444 B       256 KB      2.08%
   NCACHE_REGION:     3078312 B         3 MB     97.86%
SRAM_OCRAM_CACHED:          0 GB       256 KB      0.00%
SRAM_OCRAM_NCACHED:      38400 B       256 KB     14.65%
Finished building target: sln_viznlc_iot_rt106f_smart_lock.axf

Performing post-build steps
arm-none-eabi-size "sln_viznlc_iot_rt106f_smart_lock.axf"; # arm-none-eabi-objcopy -v -O binary "sln
   text    data     bss     dec     hex filename
4019612    6264 14004352    18030228    1131e94 sln_viznlc_iot_rt106f_smart_lock.axf
```

Figure 31. Build message

Flashing and debugging SLN-VIZNLC-IOT projects

- With the SLN-VIZNLC-IOT application project compiled, it is now time to program associated binaries of this project into flash.
- Flashing and debugging the SLN-VIZNLC-IOT kit requires a SEGGER J-Link with a 9-pin Cortex-M adapter and V7.60d or newer version of the J-Link Software and
- Documentation Pack. This new version of J-Link can be found on the SEGGER website.
  Note: There is a problem to program/debug RT106F using the default J-Link V7.70d in MCUXpresso IDE V11.6.1. J-Link software must be updated, and J-link software V7.60d has been verified.

Figure 32. J-Link 9-pin adapter

- **Note:** The SLN-VIZNLC-IOT kit has one SWD interface connector (J2), which supports programming and debugging the RT106F, LPC845, and K32W061 via a SEGGER J-Link debug probe. To select which MCU to program/debug, the J4, J5, and J6 connectors, each must be set to the position indicated in Figure 33.

Programming	J4&J5&J6	Layout Position
LPC845	1-2(Default)	
RT106F	2-3	
K32W061	2-4	

Figure 33. Switch debug interface

To flash the kit, follow the steps below:

1. Select the programming MCU on J4, J5, and J6 connectors.
2. Attach your J-Link debug probe into the J2 header as shown in Figure 34 and connect your J-Link to your computer via USB.

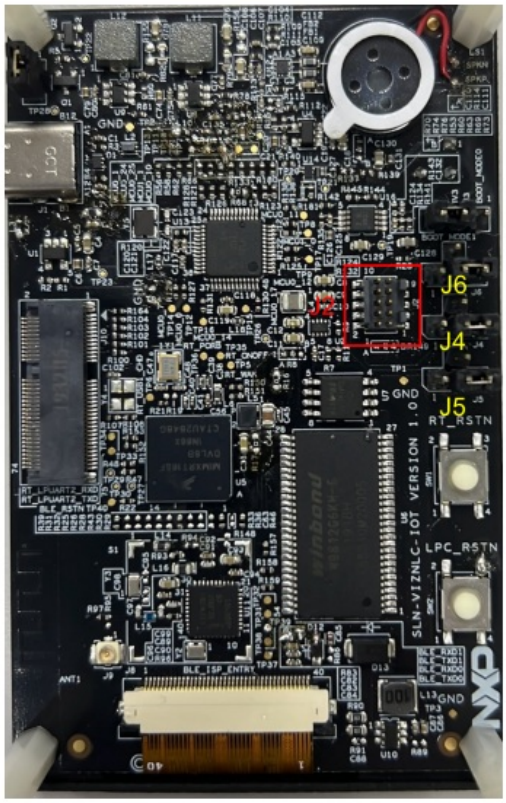


Figure 34. Attach J-Link debugger

3. Next, provide power to the kit by plugging a USB-C cable into the USB-C port of the kit and plug the other end into your laptop/PC.

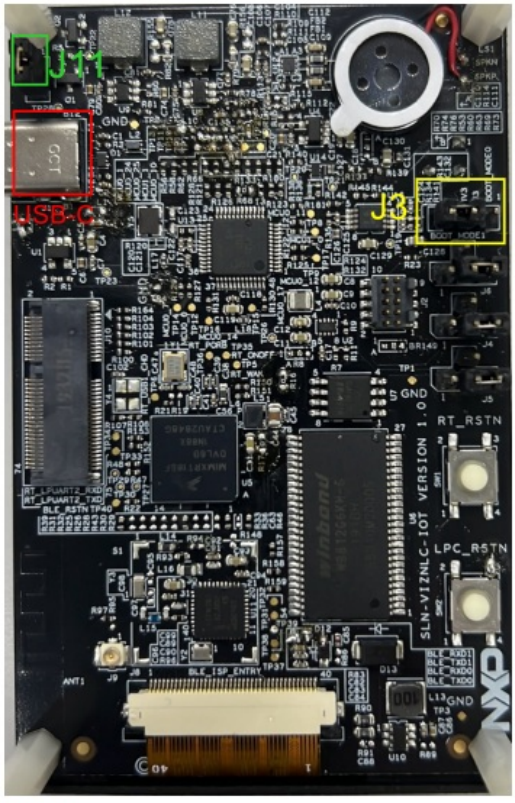


Figure 35. Power on SLN-VIZNLC-IOT kit

- **Note:** LPC845 controls the power supply of RT106F and K32W061. Therefore, during the debug stage, we can bypass it by connecting J11 jumper as shown in the green highlighted box in Figure 35.

4. Select the MCU-related project in the Project Explorer pane. Consider lpc845_low_power_control as an example below.
5. To start the process of loading the binary into flash and begin debugging, choose the Debug option in the

QuickStart Panel.

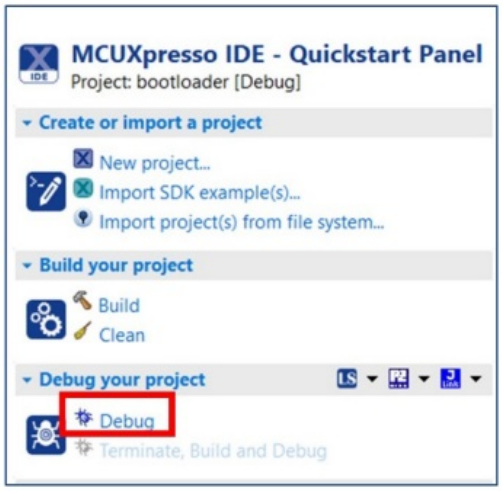


Figure 36. Debug your project

6. Select the J-Link probe that is connected to your kit and click the OK button. Flashing tool is launched.

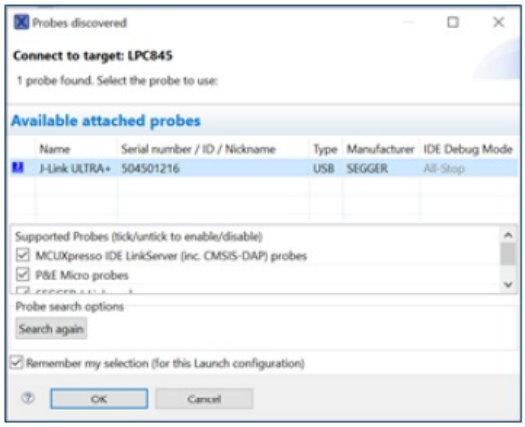


Figure 37. Probes discovered

7. Now, proceed to flash the binary associated with the currently selected project.

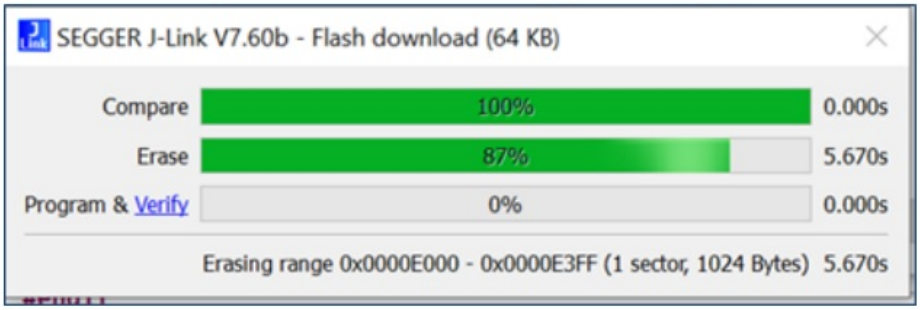


Figure 38. Flash downloading

8. After successfully debugging the application, the program breaks at the project main to start the GDB debug process.

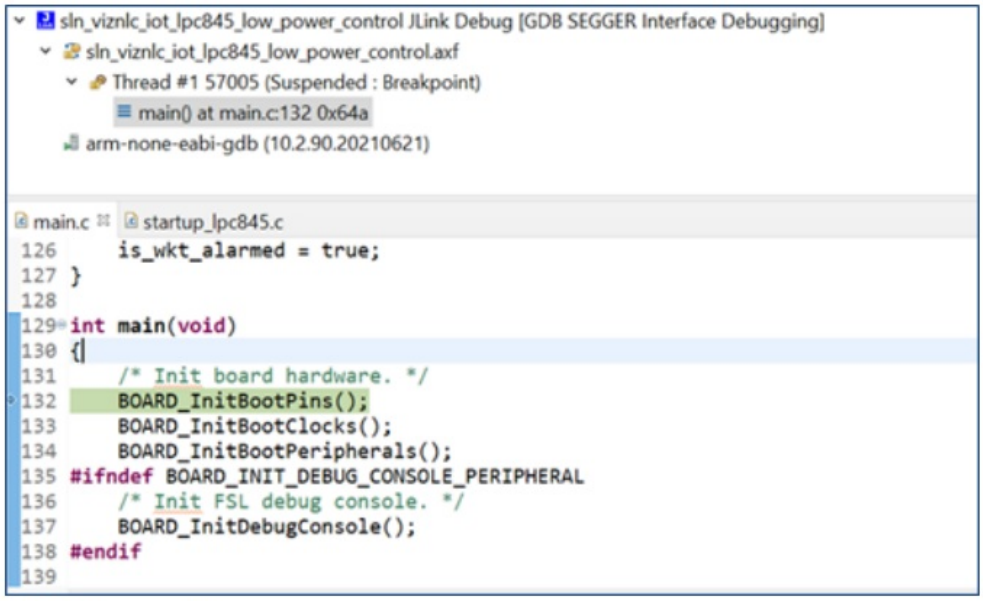


Figure 39. Start to debug project

9. To suspend the debug session, click the pause/play button on the top of the screen.

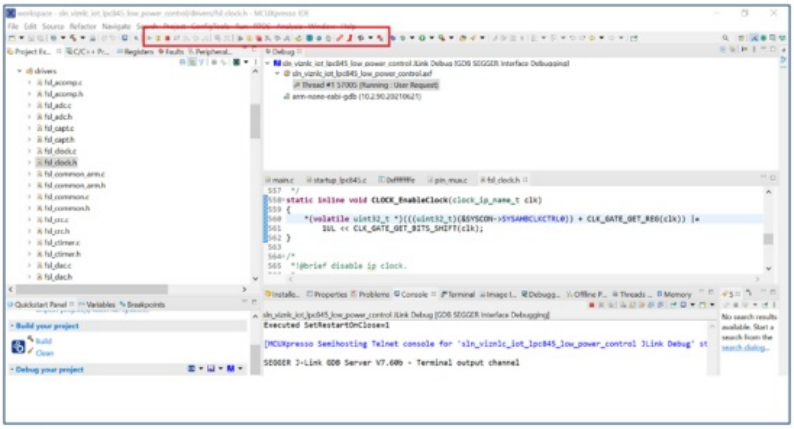


Figure 40. Debugging project

Note: If you want to debug the rt106f_smart_lock project, ensure that you have flashed the bootloader.

Additional resources

- If you have made it to this section, you have successfully finished the SLN-VIZNLC-IOT getting started experience.
- For a comprehensive understanding of all the Out-of-Box Experience (OoBE) features, including the additional demo applications that come flashed with the kit, see SLN-VIZNLC-IOT User Guide.
- To start building your own applications and learn more about the software architecture, available developer tools, and more, head over to the SLN-VIZNLC-IOT Software Developer Guide.

Acronyms

Table 1 lists the acronyms used in this document.

Table 1. Acronyms

Acronym	Definition
TFT	Thin Film Transistor
HAL	Hardware Abstraction Layer
OoBE	Out-of-Box Experience
MSD	Mass Storage Device
VIZNLC	Vision Low Cost
FW	Firmware
SW	Software
HW	Hardware
PIR	Passive InfraRed
IR	InfraRed
GUI	Graphical User Interface

Revision history

Table 2 summarizes the changes done to this document since the initial release.

Table 2. Revision history

Revision history	Date	Substantive changes
0	10 March 2023	Initial release

Legal information

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.

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.
- **Suitability for use in non-automotive qualified products —** Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.
- In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the

product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

- **Translations —** A non-English (translated) version of a document, including the legal information in that 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 vulnerabilities or may support established security standards or specifications with known limitations. 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.

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.
- AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed
- Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.
- Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.
-
- **For more information, please visit: http://www.nxp.com**
-
- **Date of release: 10 March 2023 Document identifier: SLN-VIZNLC-IOT-GSG**

Documents / Resources

	**NXP SLN-VIZNLC-IOT Smart Lock** [pdf] User Guide SLN-VIZNLC-IOT Smart Lock, SLN-VIZNLC-IOT, Smart Lock, Lock

References

- **NXP® Semiconductors Official Site \| NXP Semiconductors**
- **Our Terms And Conditions Of Commercial Sale \| NXP Semiconductors**