MCU-SMHMI-GSG

Smart HMI Getting Started Guide

Rev. 0 — 26 September 2022 User guide

Document information

Information	Content
Keywords	MCU-SMHMI-GSG, SLN-TLHMI-IOT, Smart home, Human Machine Interface (HMI), IoT
Abstract	This document describes the process of getting up and running with your SLN-TLHMI-IOT board.

1 Plug it in!

Welcome to the SLN-TLHMI-IOT Getting Started Guide! This guide walks you through the process of getting up and running with your SLN-TLHMI-IOT board. This guide takes you through the steps of unboxing your kit, running the out-of-box coffee machine 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 damage, and should you find anything, be sure to report it to your local NXP representative.

1.1 Unbox

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

- A packing list paper
- Fully assembled smart Human Machine Interface (HMI) kit
- USB-C cable

Figure 1. Items inside the box

1.2 Power on

Before we begin, remove the protective film from the RGB and IR camera, as shown in figure 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.

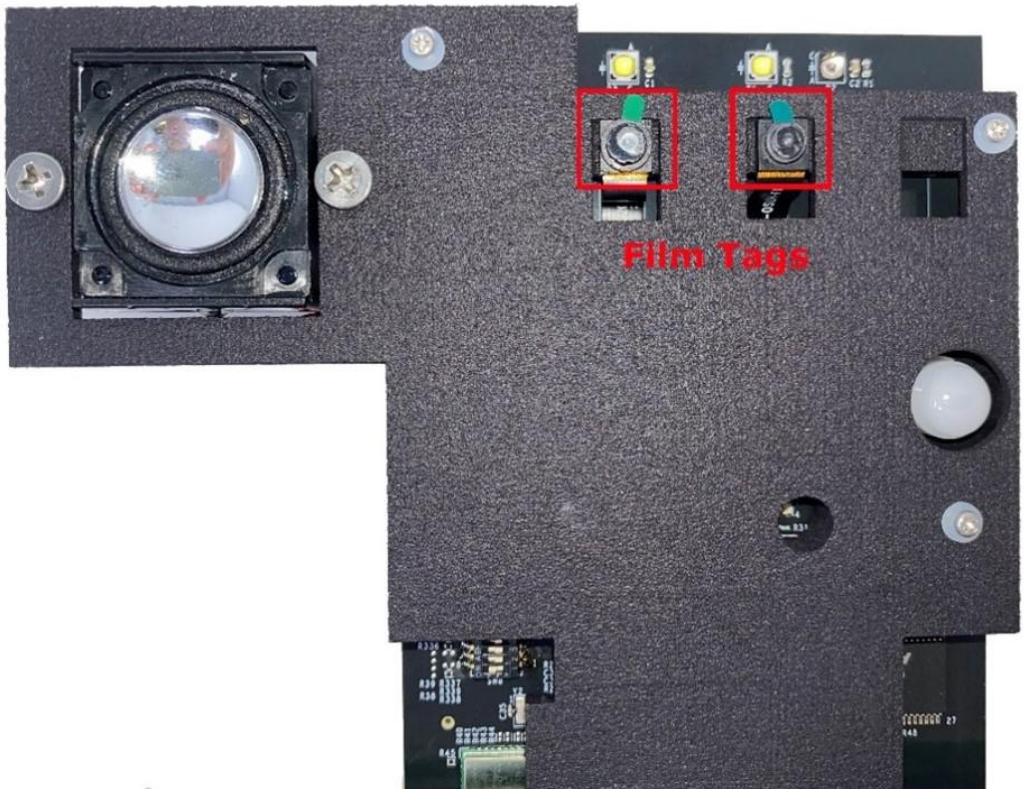


Figure 2. Protective film covering the two cameras: RGB and IR

To get started, follow the steps below:

1. Take the provided USB-C cable and plug one end into your laptop/PC.

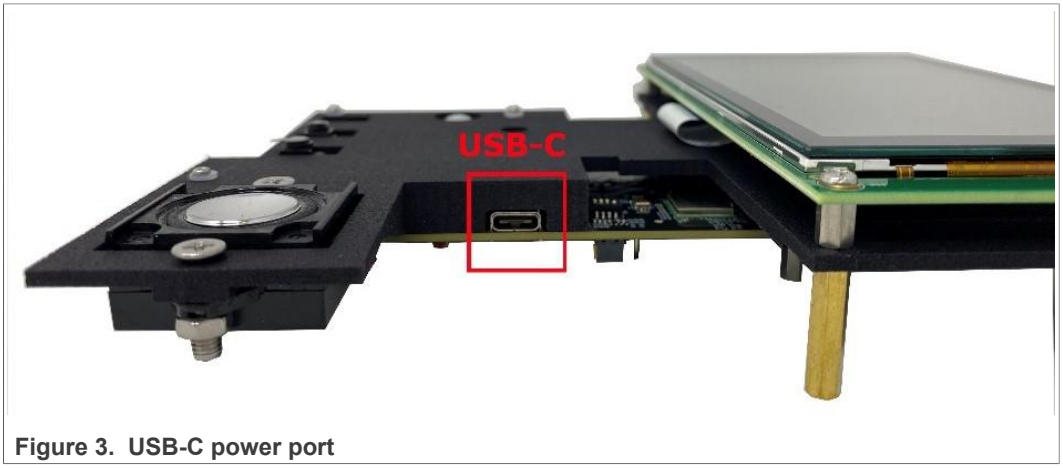


Figure 3. USB-C power port

2. Once successfully plugged in, the display shows a startup screen with the icon for the currently enabled application most prominently displayed. For the purposes of this guide, we are focusing on the coffee machine application.

Figure 4. Startup screen

3. Touch the screen or wait a few seconds and the coffee machine standby screen appears. This screen is used as a screen saver to indicate that the kit is waiting for user interaction. The standby screen is enabled when the board is first turned on, once a coffee is finished brewing, and after a timeout period is reached without any interaction with the kit.

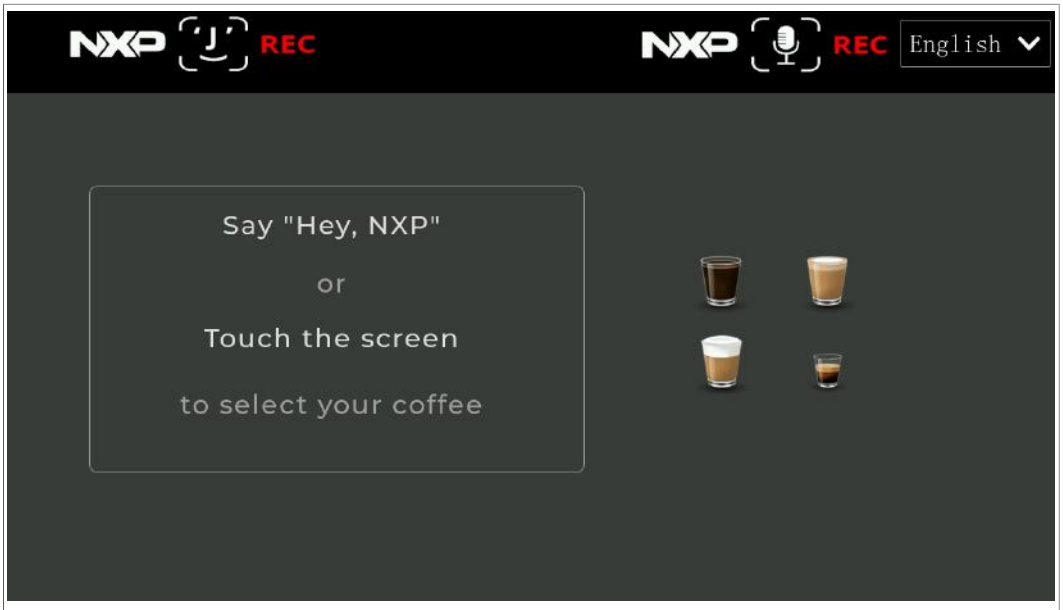


Figure 5. Coffee machine standby screen

4. Once the standby screen is enabled, touch the screen one more time to begin using the coffee machine demo. Touching the screen enables the home screen.

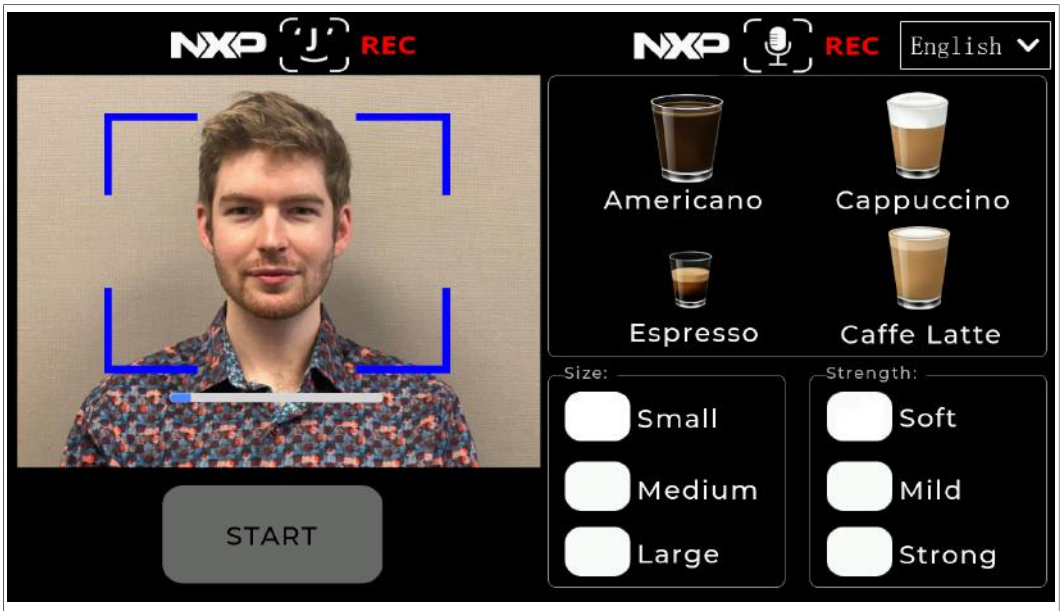


Figure 6. Home screen

Note: *If no voice commands and/or touch events are received, the board automatically reenters the standby screen after a 60 second timeout.*

1.3 Saving a new face

Let us get started with a demonstration of this application's out-of-box features!

To demonstrate the SLN-TLHMI-IOT's ability to create tailored custom experiences for each saved user, we are using our face to save a custom coffee profile.

Before diving in, let us take a quick tour of the home screen layout.

The home screen is divided into two halves: the camera preview pane on the left, and the coffee selection menu on the right.

Additionally, there is a **Start** button in the lower-left corner of the screen and a language selector drop-down menu in the top-right corner of the screen.

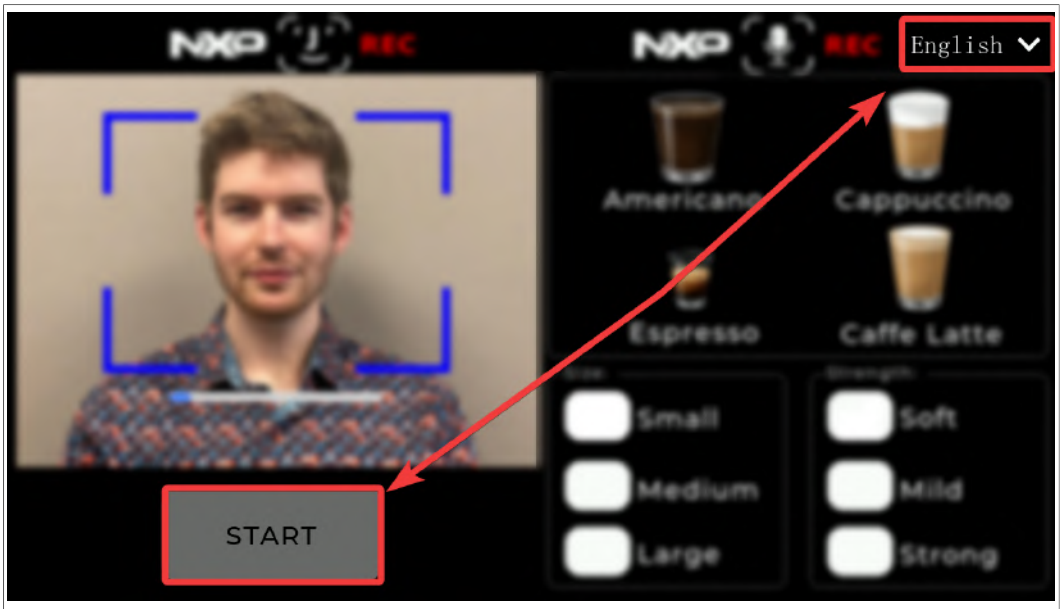


Figure 7. Start button and language drop-down

On the camera preview pane, a bounding box is provided to help you properly align your face to be recognized by the kit.

The different colors of the bounding box are used to indicate face recognition status. By default, the box is colored blue to indicate that no face has been detected.

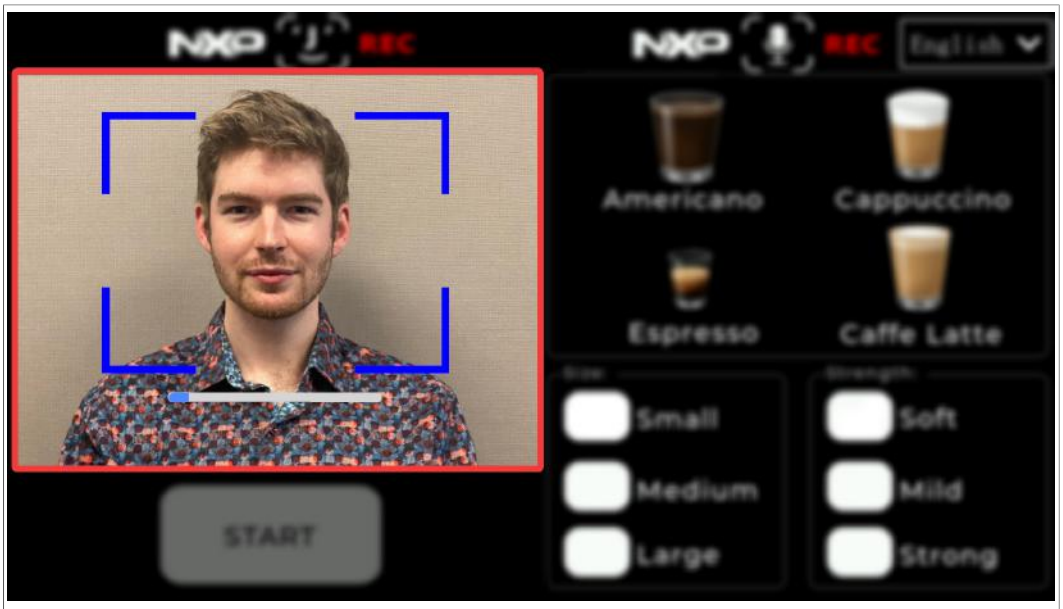


Figure 8. Blue indicates that no face has been detected

The color red is used to indicate that an unrecognized face has been detected. Align your face with the camera until the bounding box turns red.

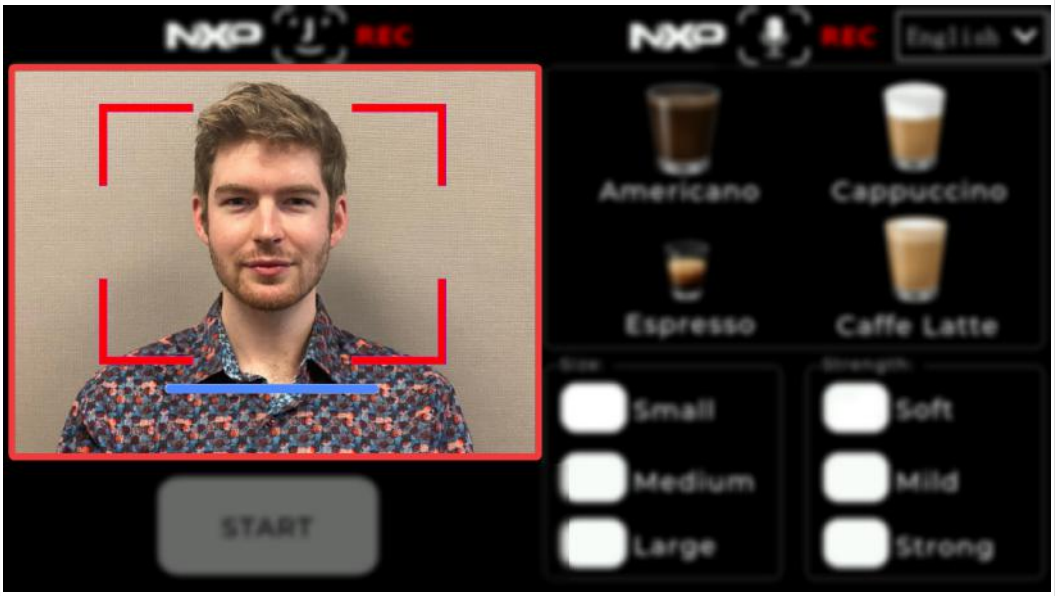


Figure 9. Red indicates that an unrecognized face has been detected

Once the camera detects your face, it can later be used to associate with your preferred coffee selection.

Now that your face has been detected, you are ready to select your coffee preferences. The coffee configuration options can be seen on the right half of the home screen as shown in Figure 10. These options include coffee *type*, *size*, and *strength*.

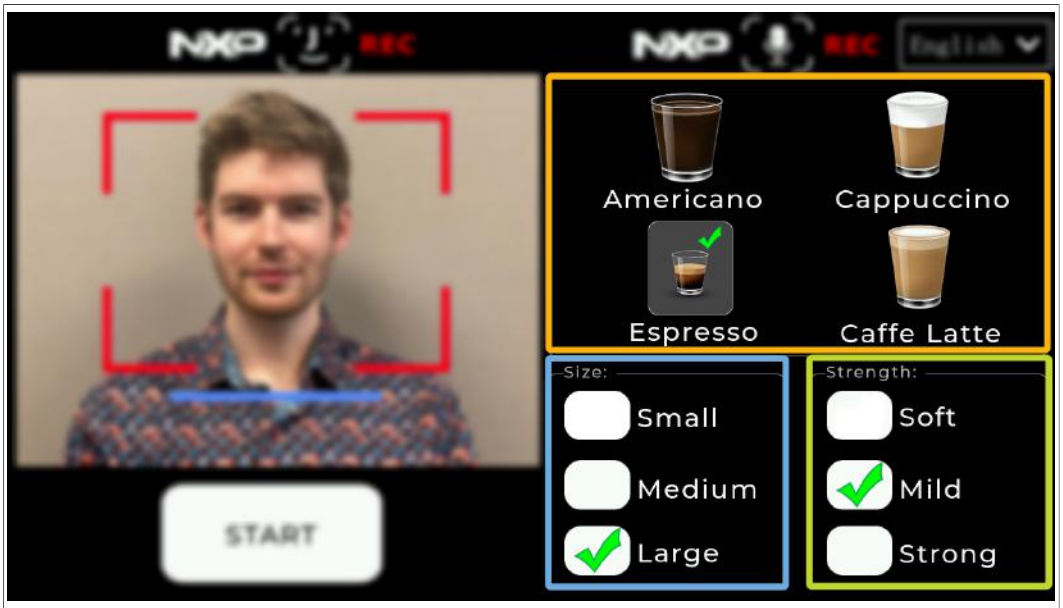


Figure 10. Coffee configuration options: coffee type, size, and strength

To choose your coffee preferences, follow the steps below:

1. Select the coffee type by pressing the icon associated with your preference.
2. Next, click the checkbox selector next to your preferred size.
3. Finally, select the checkbox selector next to your preferred coffee strength.

4. The coffee selection has been customized. Before moving on, ensure that the camera detects your face (indicated by a red bounding box) before pressing the **Start** button to begin brewing the coffee.
   Note: *The start button is grayed out and cannot be pressed until a coffee type, size, and strength have been selected.*

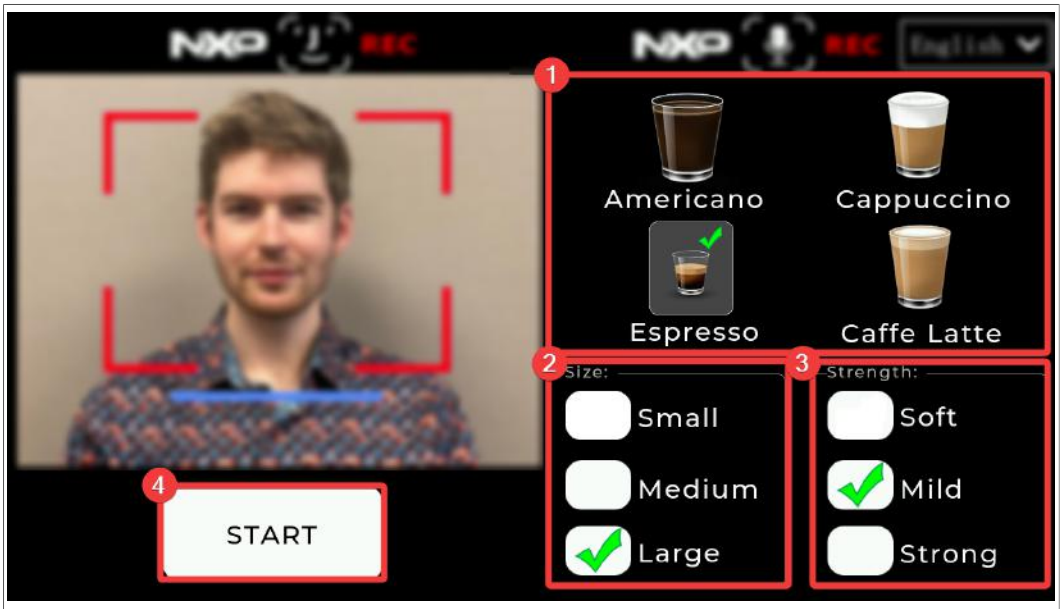


Figure 11. Selecting coffee preferences

5. After pressing the start button, a brewing animation is shown. The brewing screen displays a progress bar and a summary of the chosen selections on the left.

Figure 12. Brewing screen

6. Once the brewing animation is finished, the chosen coffee is shown on the final screen.

Figure 13. Final screen

7. After a brief delay, a voice prompt plays via the onboard speakers, asking "Save your coffee selection? Confirm or Cancel."
   Say "Confirm" to save this order.
   Answering "Confirm" associates the coffee that was brewed with your face. Next time your face is recognized, the coffee machine automatically asks whether you would like to reuse your saved order.

After your order is complete and the prompt has been responded to, the device automatically reenters the standby screen that has been shown at startup.

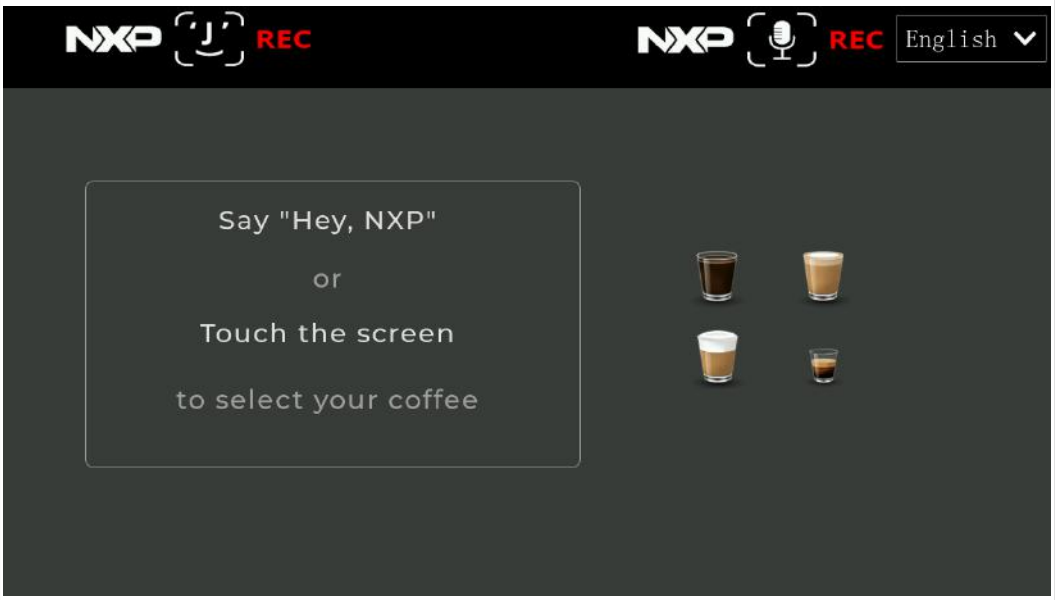


Figure 14. Standby screen

1.4 Modifying a saved face

Now that we have demonstrated the core features using traditional touch-based controls, let us walk through the next generation hands-free voice control capabilities made possible by the processing power of the i.MX RT117H.

First, wake the board out of Standby mode, using the "Hey NXP" wake word.

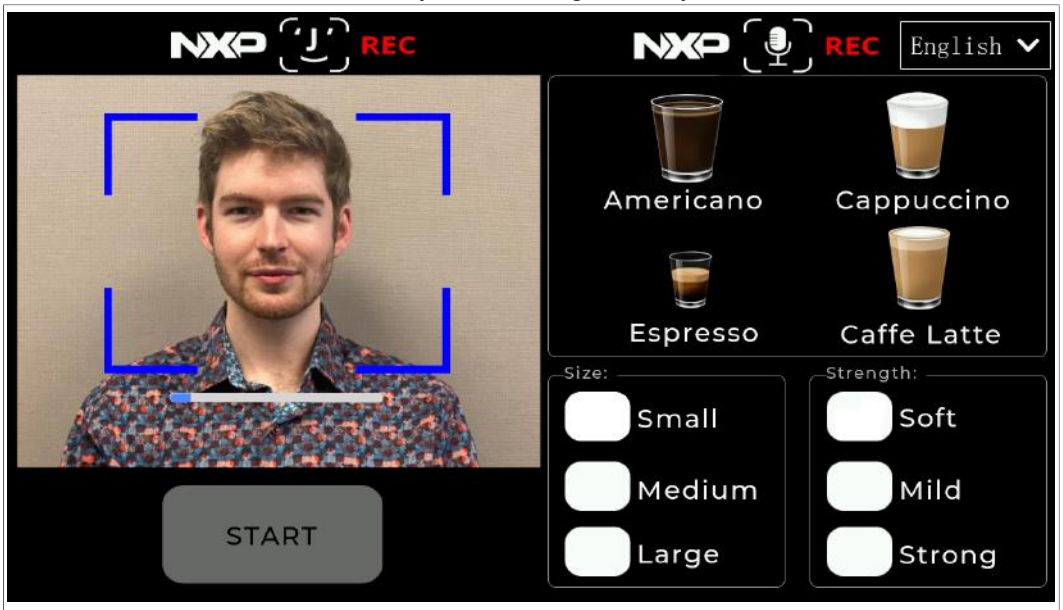


Figure 15. Home screen

Note: *The wake word is used to exit Standby mode but is not necessary to be used prior to each voice command.*

With the board active once more, align your face inside the bounding box until the color of the box turns green[1].

1 If the bounding box turns red, your face is not recognized as a saved user. Assuming you responded with "Confirm" to the registration prompt in the previous step, meaning the application has saved your face and coffee order, you may need to power cycle the kit. Once a face has been recognized or a new face has been detected, the board must be reset or enter standby before it attempts to look for a different face.

Figure 16. Green indicates that your face has been recognized

A green bounding box indicates that you are a recognized user. The coffee machine plays an audio prompt asking if you would like to order "Another Espresso? Confirm or Cancel?"

Responding with "Confirm" immediately takes you to the brewing screen again, while "Cancel" can be used to choose a different coffee selection.

Instead of reusing our saved order, respond to the prompt with the "Cancel" voice command, allowing us to choose something different. This time around, however, we are using hands-free voice commands instead of the touch controls we used previously.

To demonstrate the touchless capabilities of the TLHMI, follow the steps below:

1. Set the coffee type by using the "Cappuccino" voice command. A green LED and brief audio tone indicates successfully recognized voice commands.
   Note: *Each of the supported voice commands matches the words shown on the screen.*

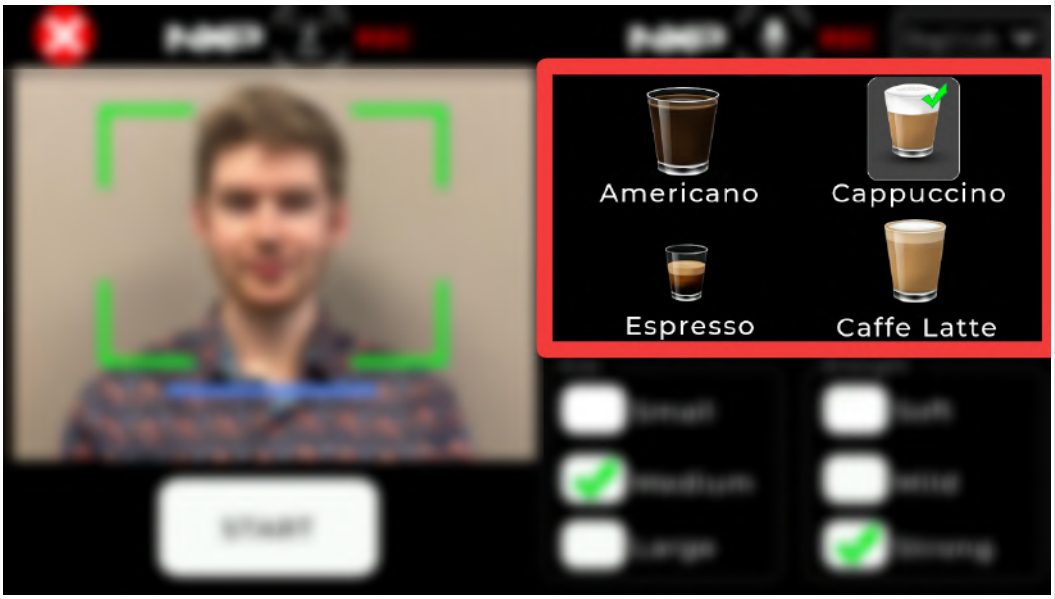


Figure 17. "Cappuccino" voice command updates the coffee type selection

2. Now, update the coffee size using the "Medium" voice command.

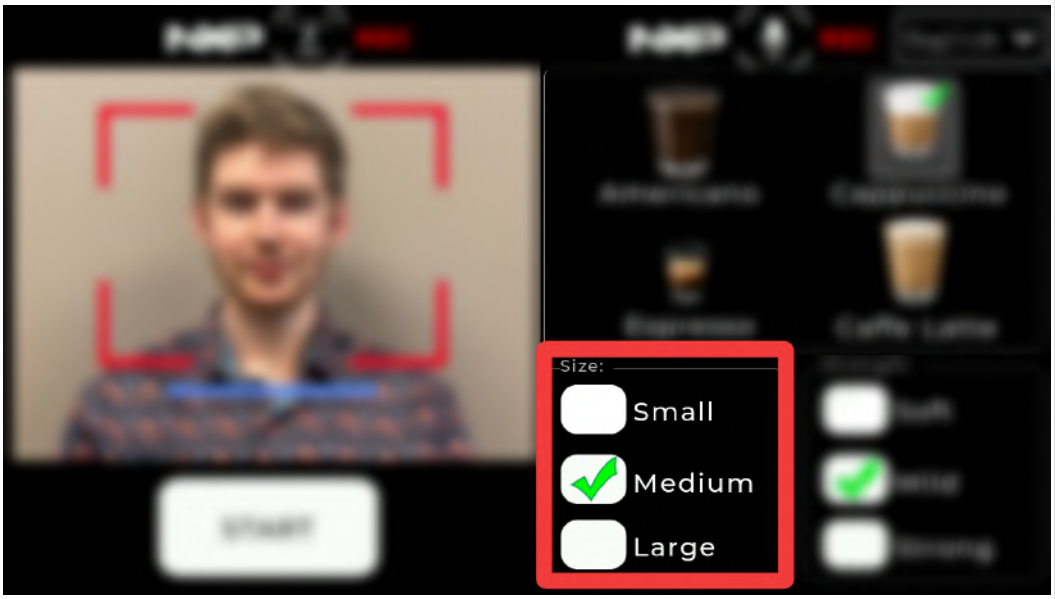


Figure 18. "Medium" voice command updates the coffee size selection

3. Set the strength to the maximum using the "Strong" voice command.

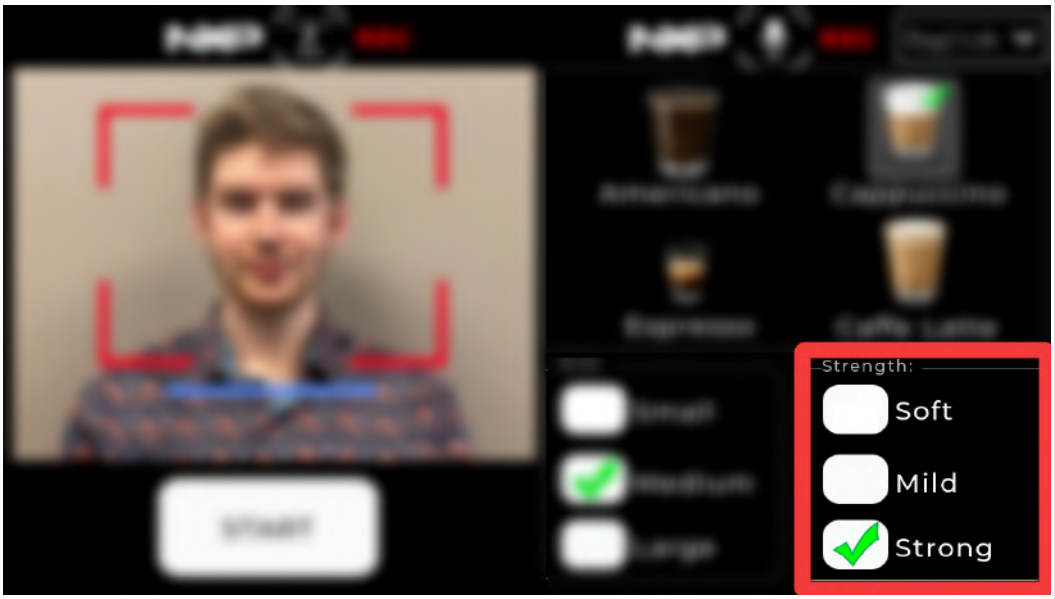


Figure 19. "Strong" voice command updates the coffee strength selection

4. Finally, start the brewing animation again using the "Start" voice command.

Figure 20. Brewing screen displays the updated coffee selection

5. Once the animation is completed, and the final screen is reached, you are prompted again to save your new order (replacing the previous order).

Figure 21. Final screen displays the updated coffee selection

Responding to the prompt with "Confirm" overwrites your previously saved order with your latest selection. Now, the next time your face is recognized, it automatically uses your updated order.

Figure 22. Modified coffee selection

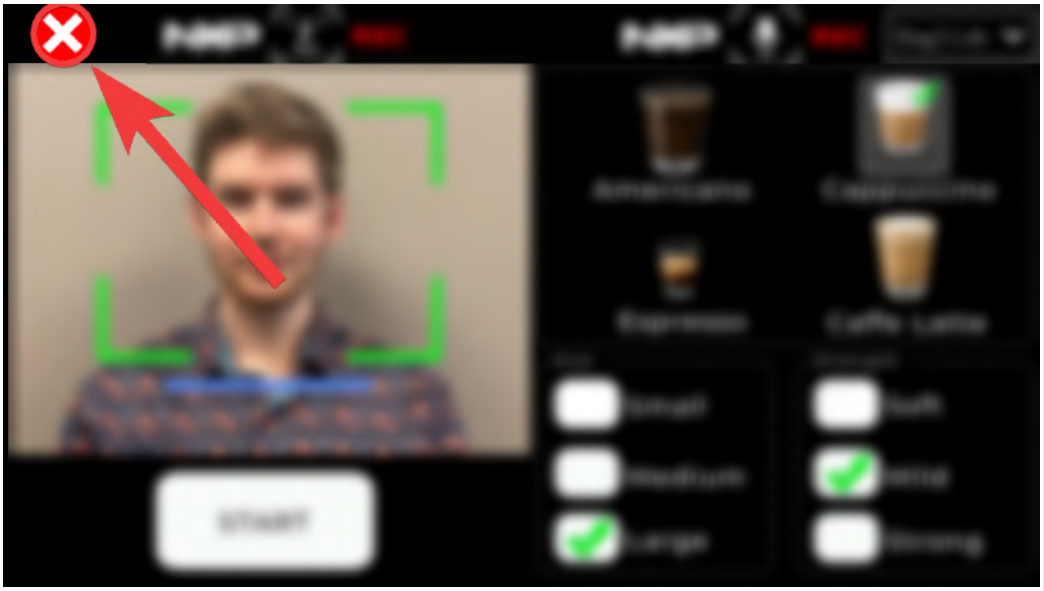
1.5 Deleting a saved face

Now that we have demonstrated saving a face and updating a coffee order, let us finally demonstrate deleting our face from the local face database.

To do so, reawaken the board from Standby mode using the "Hey NXP" wake word and/or by touching the screen.

After the display turns back on, align your face with the onscreen bounding box. Once your face is recognized, use the "Delete User" voice command to delete your face from the database. Alternatively, you can click the red **X** icon in the top-left corner of the screen to remove the recognized face.

Figure 23. Deleting a saved face



After successfully saving a new coffee machine user, modifying, and deleting the order, we have completed our introduction of the coffee machine out-of-box demo experience!

Now that you are familiar with the basics, see the *Smart HMI User Guide* for a comprehensive list of all the out-of-box features available in SLN-TLHMI-IOT. The guide also includes the additional demo applications available such as the smart elevator and smart home touch-panel applications.

2 Setup and installation

2.1 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.

Note: *There is a bug with version 11.5.1 of MCUXpresso IDE which prevents building projects for the SLN-TLHMI-IOT. Therefore, version 11.5.0 is required.*

To download MCUXpresso IDE, follow the steps below:

1. Go to MCUXpresso IDE v11.5.0 [Build 7232] [2022-01-11] and click the **Downloads** button.

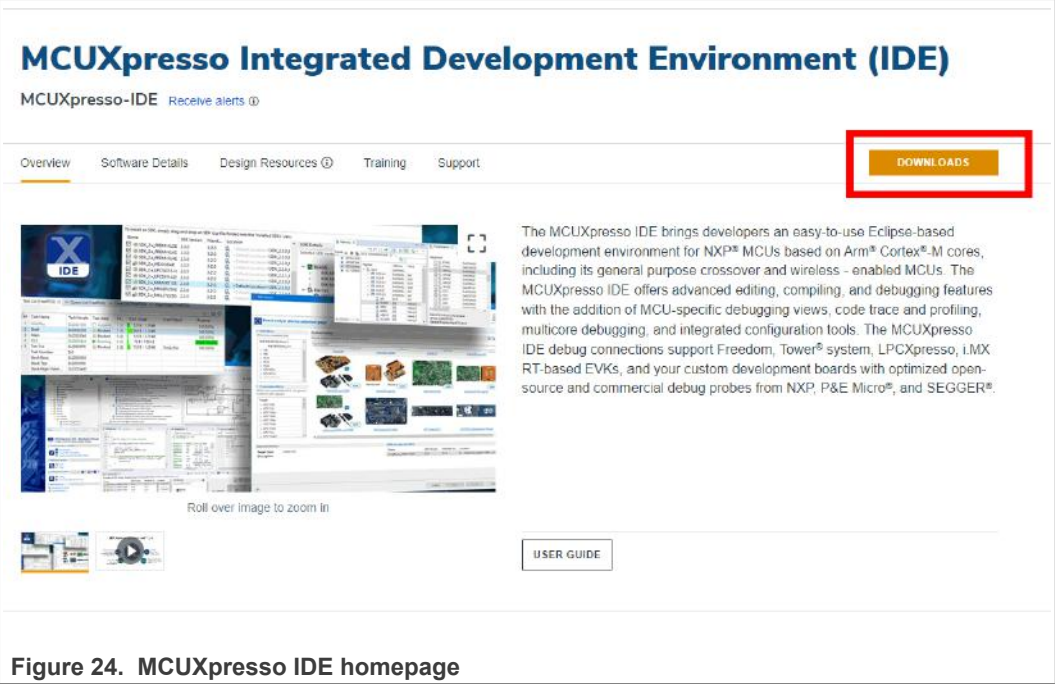


Figure 24. MCUXpresso IDE homepage

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

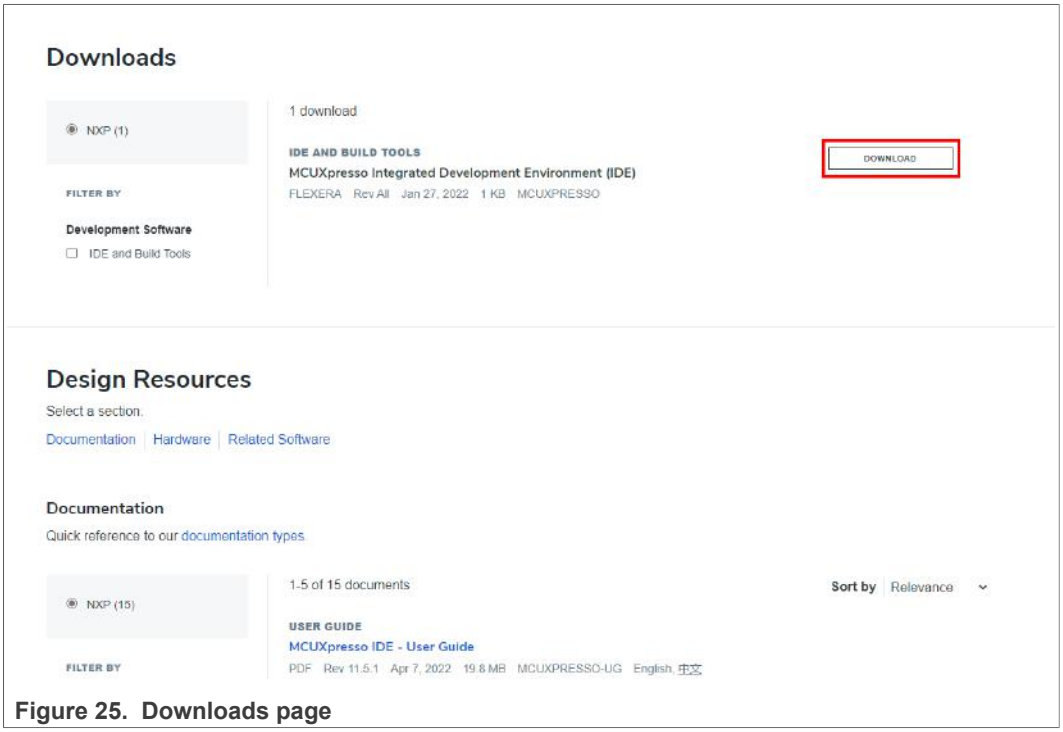


Figure 25. Downloads page

3. The **Sign In** page appears. Login with your credentials.

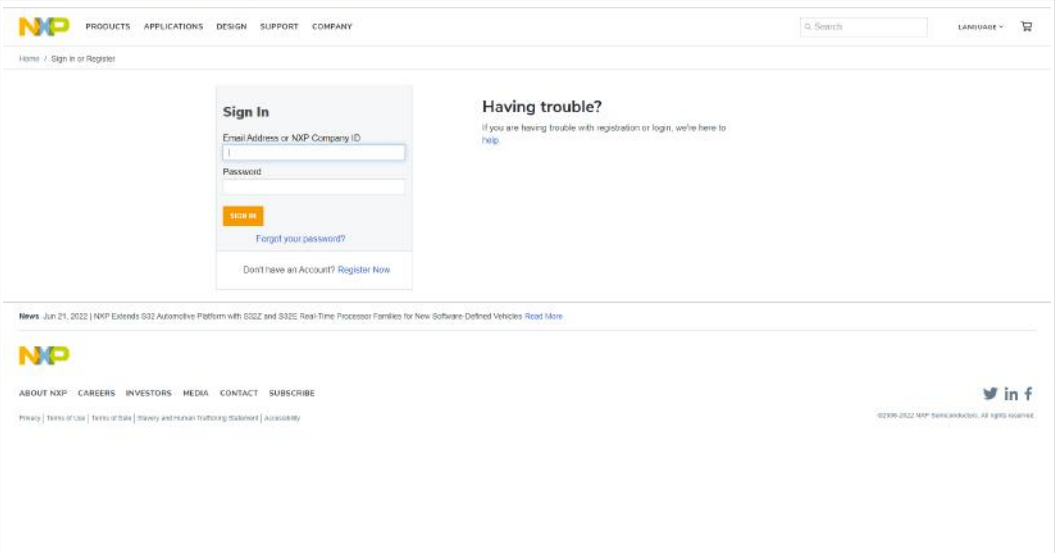


Figure 26. Sign In page

4. Once you have signed in, select the **Previous** tab.

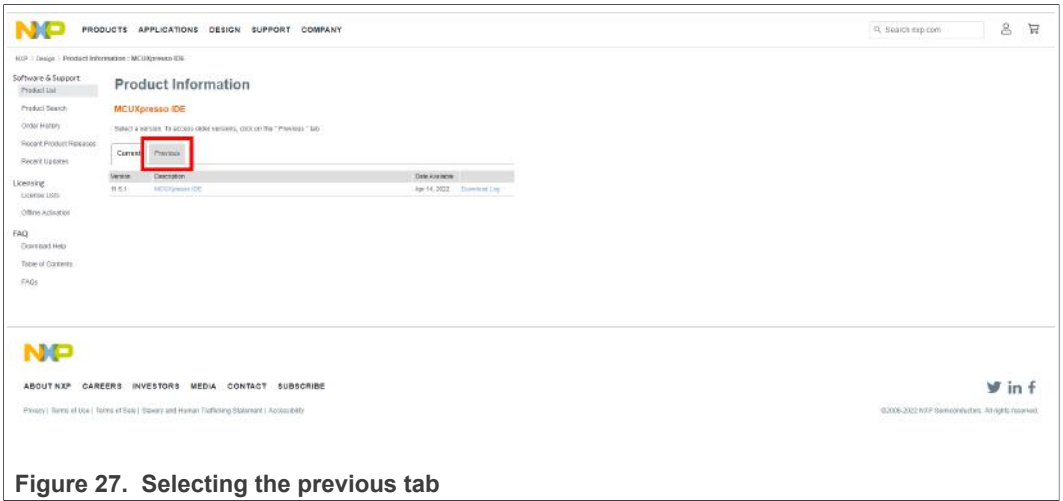


Figure 27. Selecting the previous tab

5. Select **v11.5.0** (See the above note).

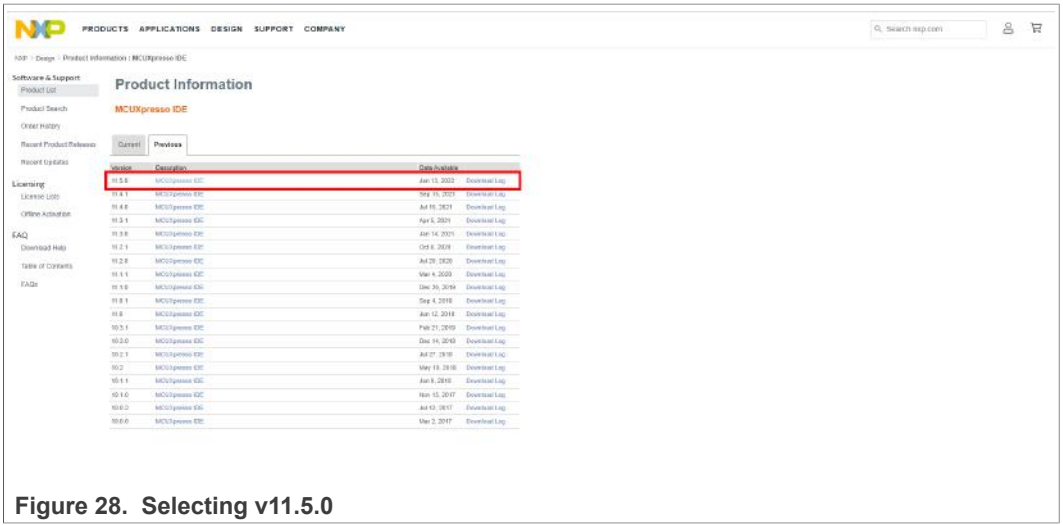


Figure 28. Selecting v11.5.0

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

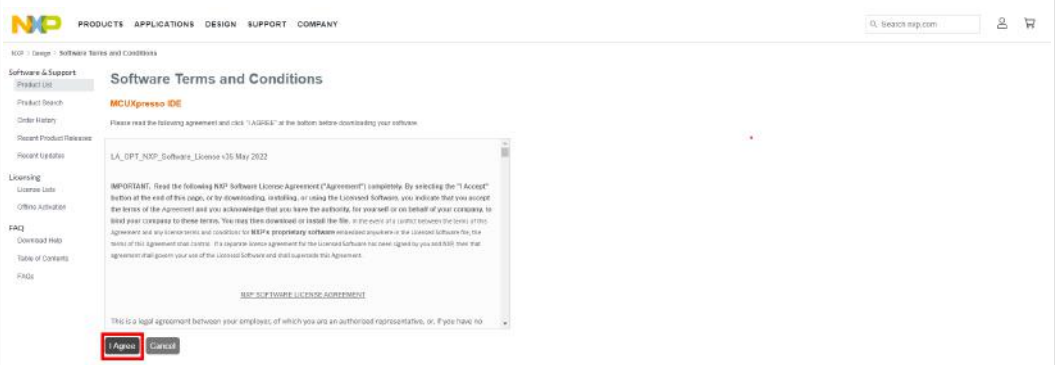


Figure 29. Software terms and conditions page

7. The **Product Download** page appears from where you can download the MCUXpresso IDE. Download the appropriate version for your system.

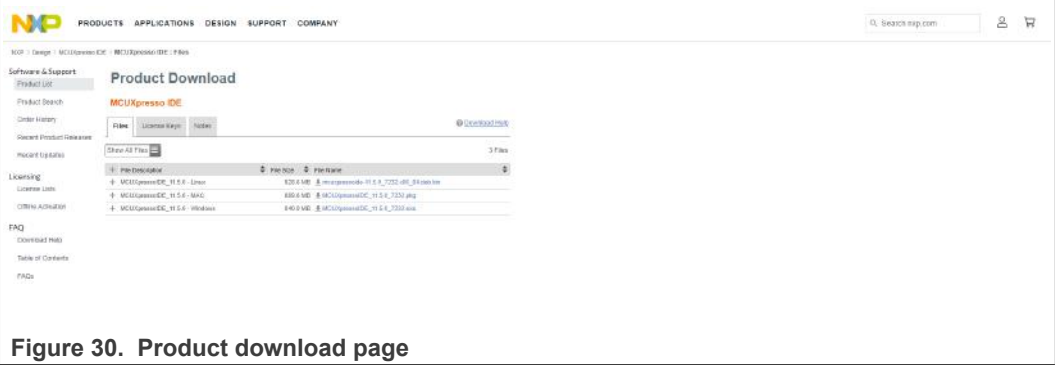


Figure 30. Product download page

8. Open the downloaded application and follow the instructions found in the installer. After successful completion of the download and installation procedure, verify the installation. To do so, select the **About MCUXpresso IDE** from the **Help** tab at the top of the IDE window.

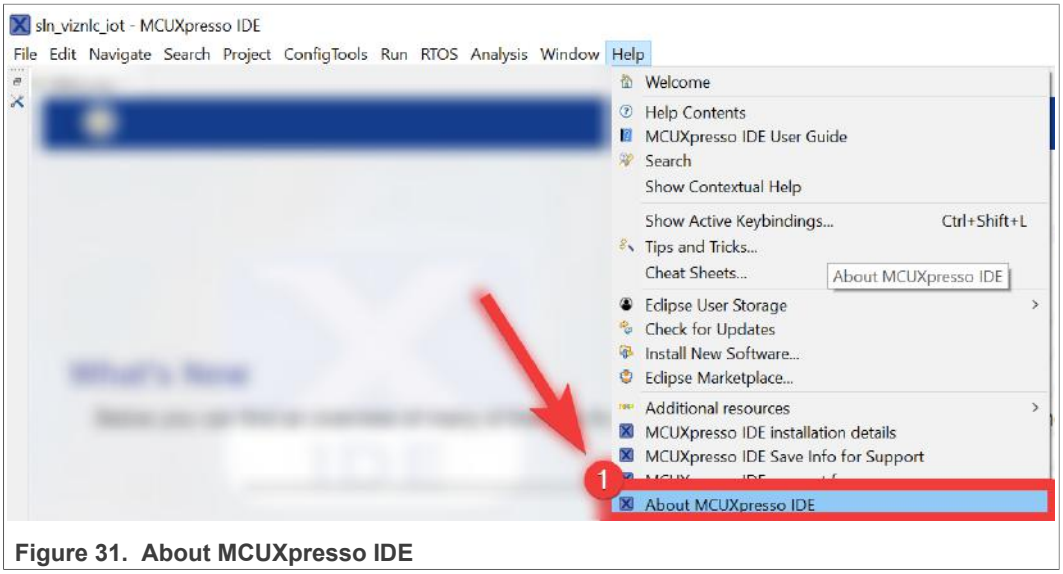


Figure 31. About MCUXpresso IDE

9. The **About MCUXpresso IDE** window appears to show the version information.

Figure 32. MCUXpresso version information

2.2 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-TLHMI-IOT SDK example projects, the target SDK must be imported into MCUXpresso IDE.

To install MCUXpresso SDK, follow the steps below:

1. Open MCUXpresso IDE. Upon opening the application, the MCUXpresso IDE welcome screen appears, as shown in Figure 33.

Figure 33. MCUXpresso IDE welcome screen

Note: *If you have already dismissed the welcome screen and are instead taken directly to the MCUXpresso IDE landing page, you can search for "welcome" from the IDE's search menu, as shown in Figure 34.*

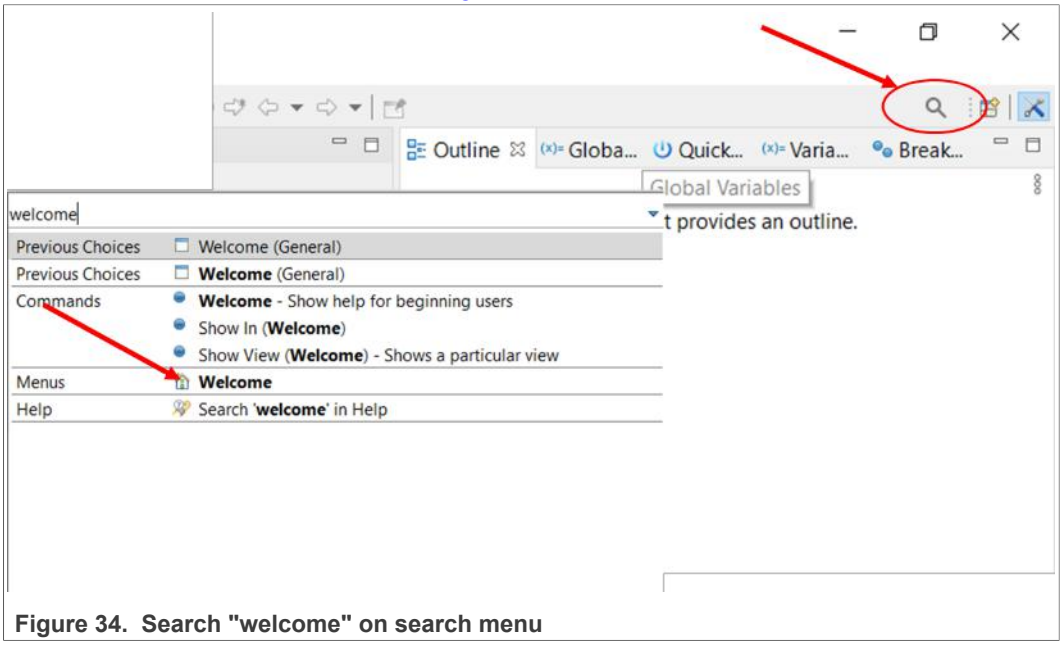


Figure 34. Search "welcome" on search menu

2. Click the **Download and Install SDKs** link.

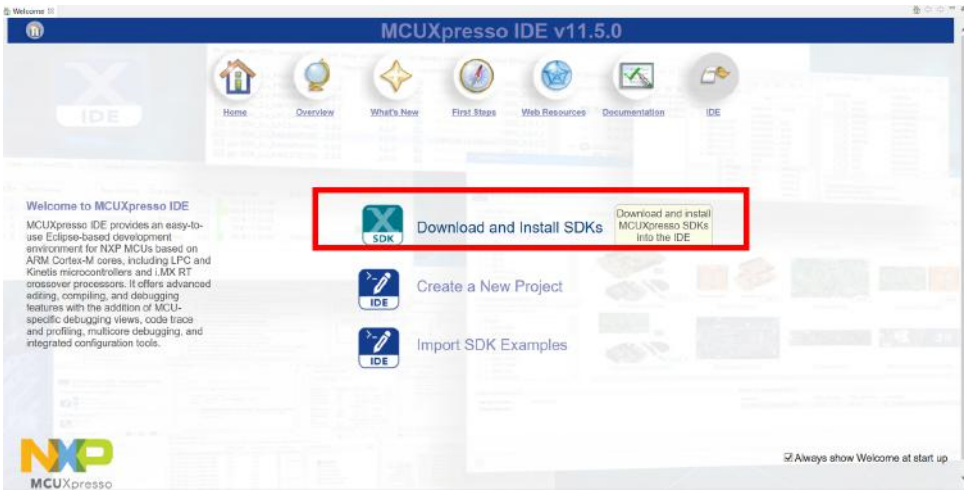


Figure 35. Download SDKs

3. You see a catalog of all the SDKs that can be downloaded through MCUXpresso, as shown in figure Figure 36. These SDKs provide device knowledge, drivers, middleware, and reference example applications for your development board or MCU.

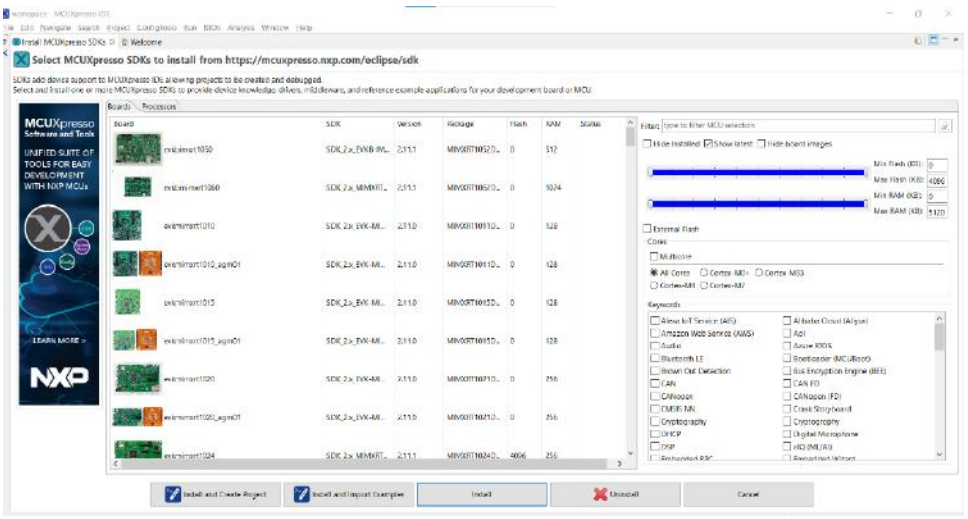


Figure 36. SDK installation popup

4. Type 1170 in the filter section, and select **evkmimxrt1170** as shown in Figure 37.

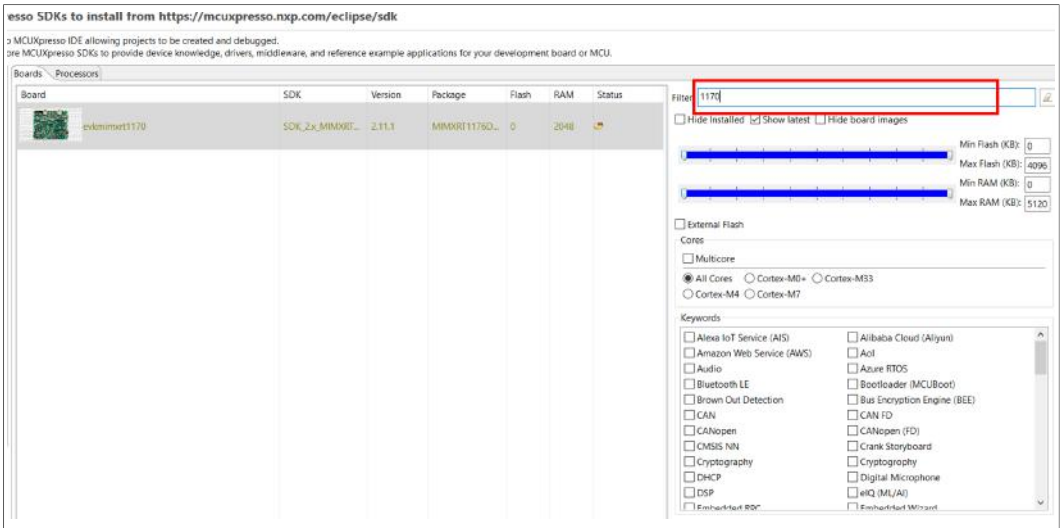


Figure 37. Download evkmimxrt1170 SDK

5. The **Review Licenses** prompt appears. Accept the license agreement and click the **Finish** button.

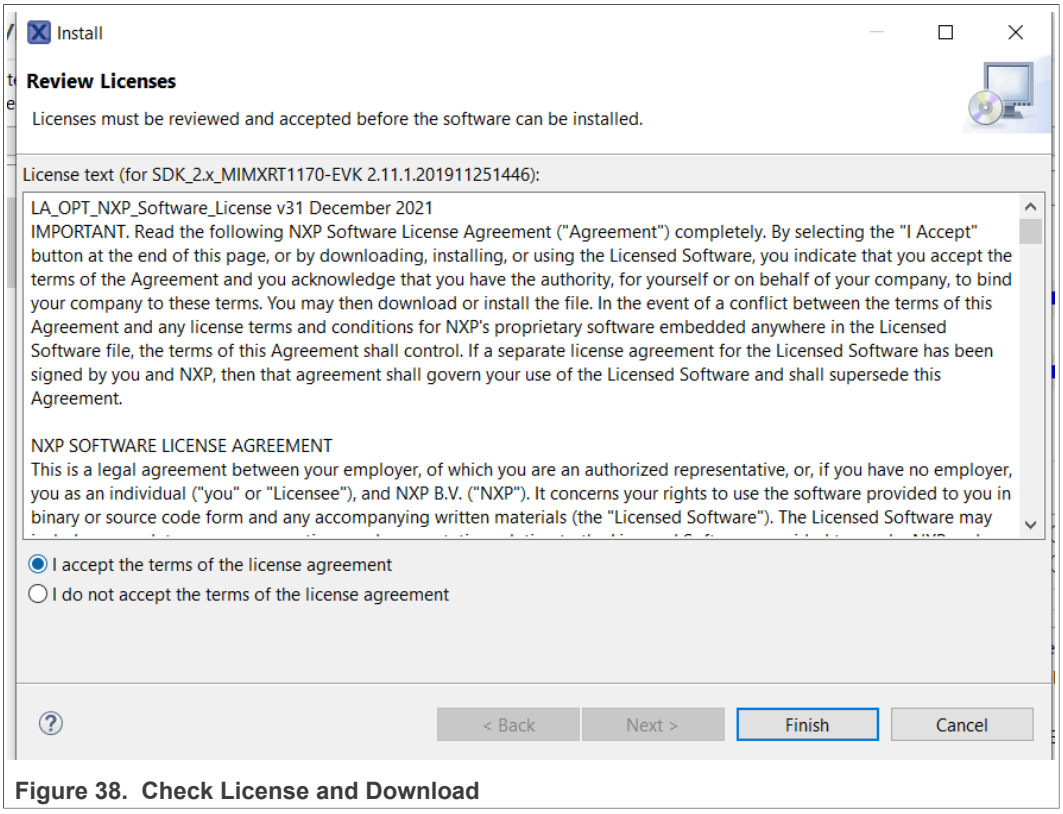


Figure 38. Check License and Download

6. The MCUXpresso proceeds to download the SDK.

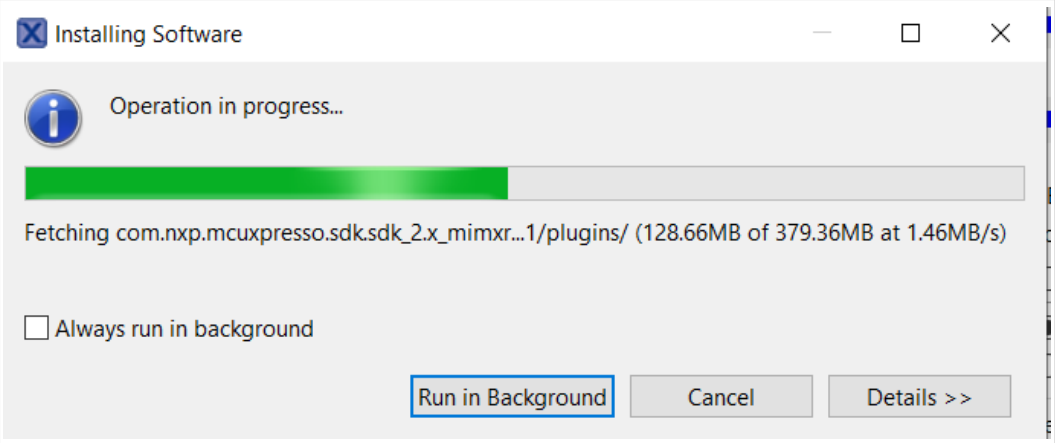


Figure 39. Downloading SDK

2.3 Downloading SLN-TLHMI-IOT projects

The SLN-TLHMI-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 here.

If you are unfamiliar with Git, it is recommended to download the zip folder for now.

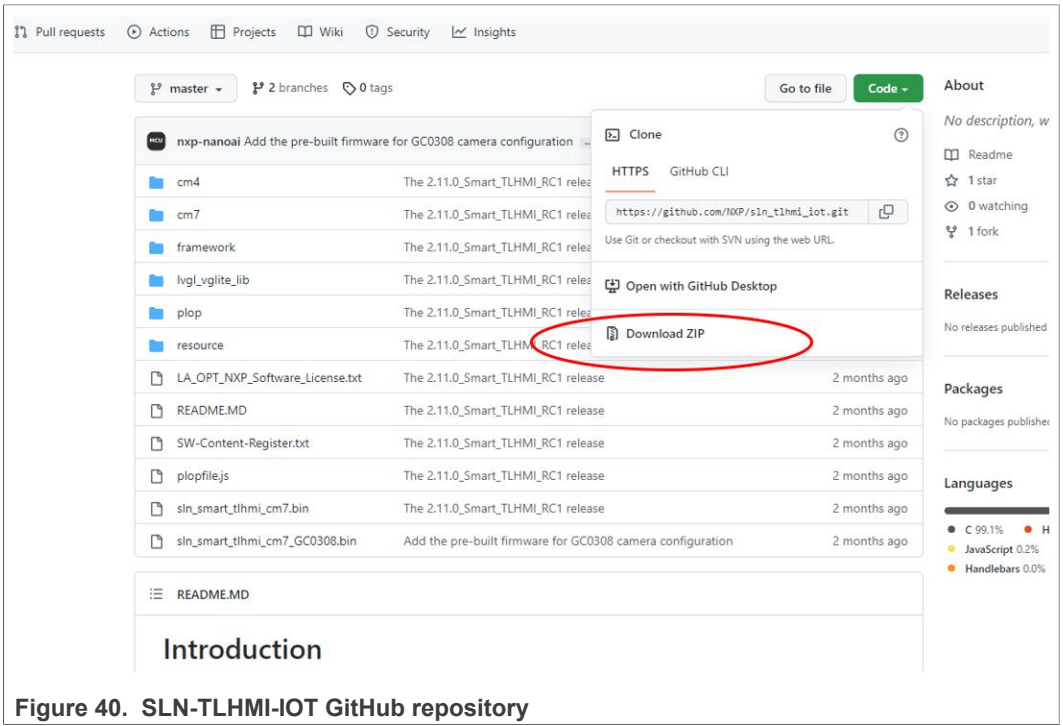


Figure 40. SLN-TLHMI-IOT GitHub repository

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

3 Build and run

3.1 Importing SLN-TLHMI-IOT projects

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

1. Open the **File** menu in the top left corner of the IDE and select **Open Projects from File System…** option.
2. From the screen displayed, navigate to the location where you unzipped the `sln_tlhmi_iot` source code. Now, import the files listed in Figure 41.
   Note: *The `sln_tlhmi_iot` source code repository contains multiple projects including each of the out-of-box demos included with the kit (coffee machine, elevator, and so on). The following sections only focus on flashing and debugging the coffee machine application. The steps for flashing and debugging the other out-of-box demos should be nearly identical.*

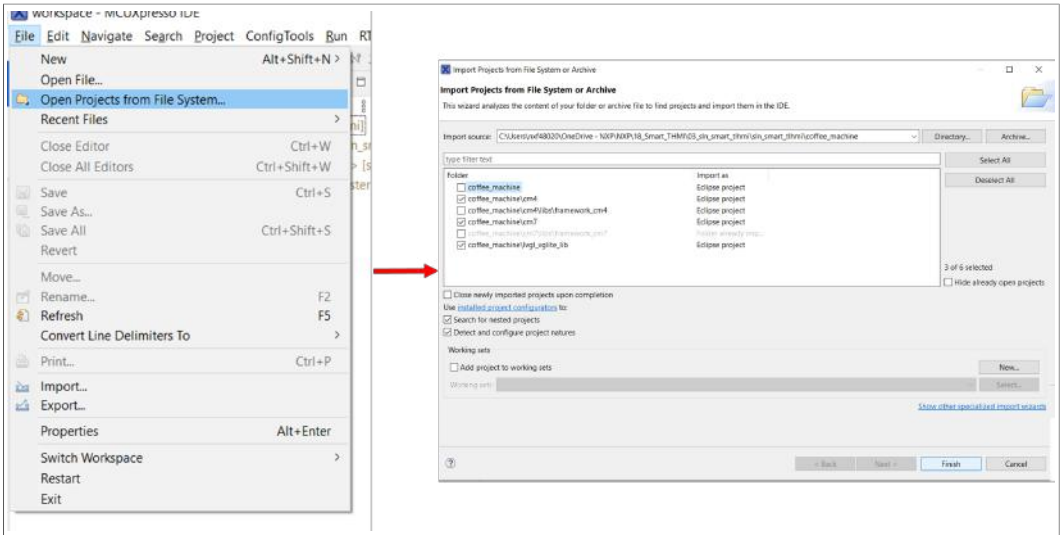


Figure 41. Open SLN-TLHMI-IOT Projects from filesystem

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

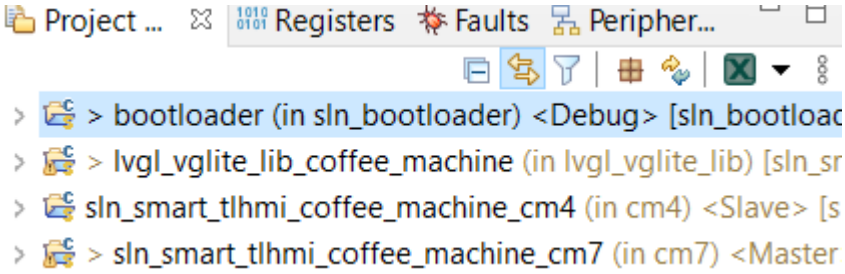


Figure 42. Finish importing SLN-TLHMI-IOT application projects

3.2 Building the SLN-TLHMI-IOT project

The SLN-TLHMI-IOT SDK allows you to build the coffee machine application directly. The application is made up of four subprojects: **bootloader**, dual core **xxx_cm7** and **xxx_cm4** projects for the chosen application demo (coffee machine in our case), and a Light and Versatile Graphics Library (LVGL) library project.

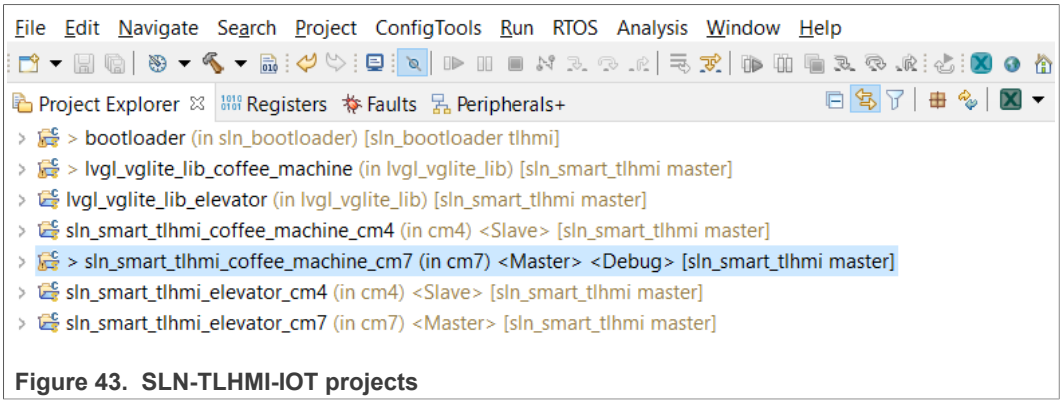


Figure 43. SLN-TLHMI-IOT projects

The **bootloader** project is a first-stage bootloader that manages to jump into the coffee machine and elevator applications. This application can be used for any additional bootloader functionality needed for the product. In the **Project Explorer** pane, select the bootloader project file and navigate to the **QuickStart Panel**. Select the **Build** option to start the compilation and linking of this application.

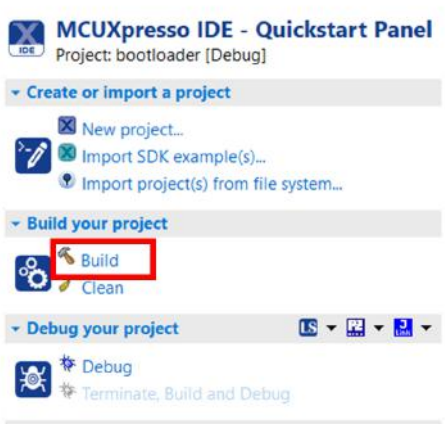


Figure 44. Build

Each of the SLN-TLHMI-IOT applications utilize LVGL for their UI design. Recompiling the code associated with the UI can add significant time and cost to the build process. In MCUXpresso, this process must be done every time the cm7 project is built. This can be especially inconvenient when no UI code has been modified because the rebuilding of LVGL's associated makefiles and source code files takes place regardless of whether code has been modified or not.

To help alleviate this issue, a library project named `lvgl_vglite_lib_coffee_machine` has been created to allow more explicit control over when the UI files should be rebuilt. Essentially, the `lvgl_vglite_lib_coffee_machine` project only needs to be rebuilt whenever changes have been made specifically to the UI code, otherwise, the `lvgl_vglite_lib_xxx` project can safely be ignored. Whenever the library project is rebuilt, MCUXpresso IDE automatically links the newly created UI library code.

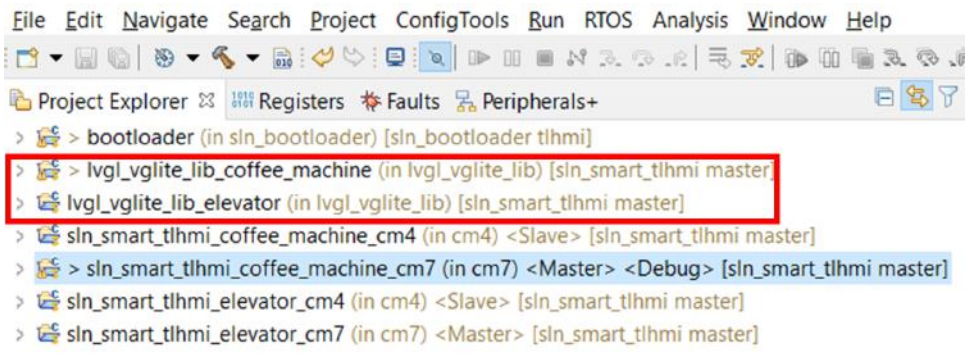


Figure 45. LVGL library projects

Note: *For more information about the software layout and architecture of the SLN-TLHMI-IOT, be sure to check out the Smart HMI Software Development User Guide.*

The `sln_smart_TLHMI_coffee_machine_cm4/cm7` projects are the out-of-box applications that we used earlier to demonstrate the SLN-TLHMI-IOT's face and voice recognition capabilities. These applications (in addition to the bootloader) are flashed onto your SLN-TLHMI-IOT kit by default.

The cm4 project acts as a subproject for the *main* cm7 project and handles things like the UI display, camera control, and more. As a subproject to the cm7 project, building the cm7 master application automatically build the cm4 and create a combined binary containing the code from both projects.

To build the coffee machine application, click the **Build** icon in the **Quickstart Panel**.

Building may take a few minutes to complete, but do not worry, this is normal for applications of this size. Once finished, a message like the following can be seen at the bottom of the IDE.

```
Invoking: MCU C++ Linker
arm-none-eabi-c++ -nostdlib -L"C:\Users\nxf48020\OneDrive - NXP\NXP\18_Smart_THMI\
Memory region         Used Size  Region Size  %age Used
     BOARD_FLASH:     5371488 B         6 MB     85.38%
     BOARD_SDRAM:     2408784 B         4 MB     57.43%
    SRAM_DTC_cm7:      213160 B       448 KB     46.47%
    SRAM_ITC_cm7:       30672 B        64 KB     46.80%
SRAM_OCRAM_CACHED:        240 KB         1 MB     23.44%
SRAM_OCRAM_NCACHED:       12800 B       128 KB      9.77%
    rpmsg_sh_mem:        128 KB       128 KB    100.00%
       fb_sh_mem:     1440000 B         2 MB     68.66%
   NCACHE_REGION:          0 GB         2 MB      0.00%
 BOARD_SDRAM_CM4:      870004 B        13 MB      6.38%
NCACHE_REGION_CM4:           0 GB         4 MB      0.00%
      res_sh_mem:          7 MB         7 MB    100.00%
Finished building target: sln_smart_tlhmi_coffee_machine_cm7.axf
```

Figure 46. Build Successful

3.3 Flashing and debugging SLN-TLHMI-IOT projects

With the SLN-TLHMI-IOT application project compiled, it is now time to program its associated binaries into flash.

Flashing and debugging the SLN-TLHMI-IOT kit requires a SEGGER J-Link with a 9-pin Cortex-M adapter and V7.66c or newer of the J-Link Software and Documentation Pack which can be found on the SEGGER website.

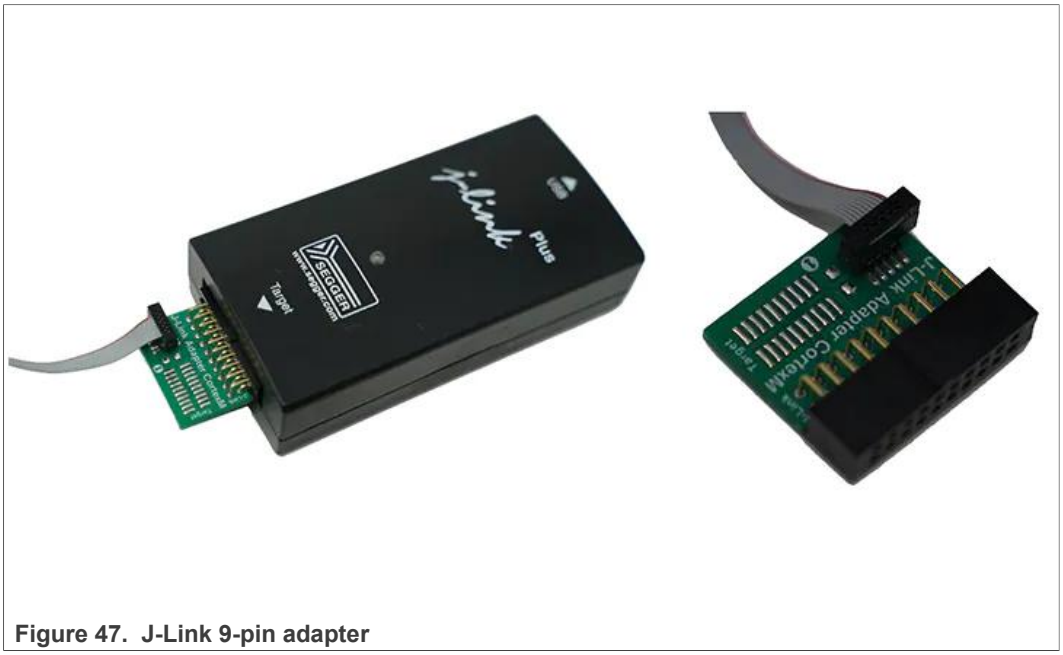


Figure 47. J-Link 9-pin adapter

To flash the kit, follow the steps below:

1. Attach your J-Link debug probe into the J204 header as shown in Figure 48 and connect your J-Link to your computer via USB.

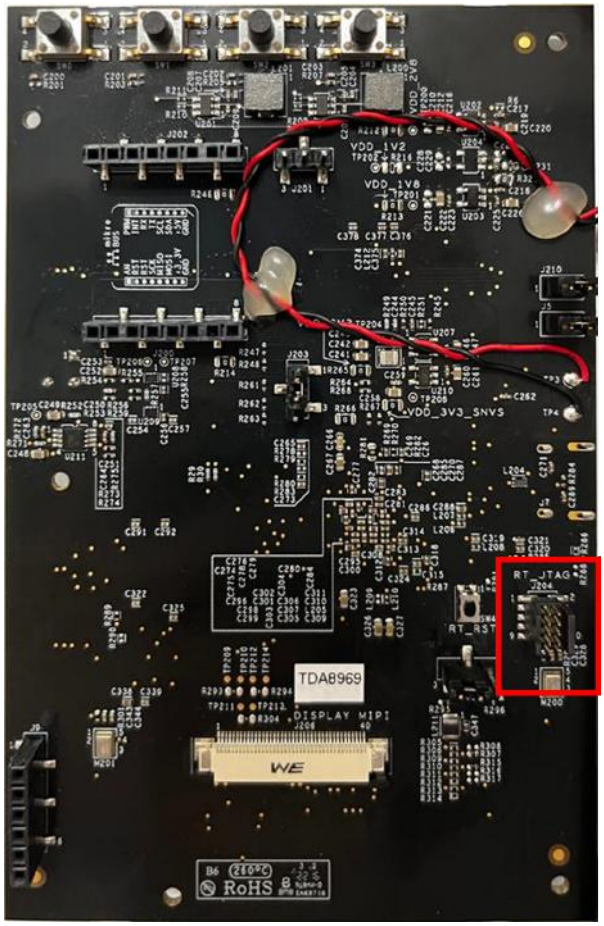


Figure 48. Attach J-Link debugger

2. Next, provide power to the kit by plugging a USB-C cable into the kit's USB-C port and plug the other end into your laptop/PC.
   Note: *If you are upgrading your board's firmware for the first time, it is recommended that you erase the flash completely before updating the firmware image. This helps to ensure that there are no version mismatches with your existing firmware and its associated face database layout. Instructions for erasing the flash using a SEGGER J-Link can be found in J-Link section of the Smart HMI Software Development User Guide.*

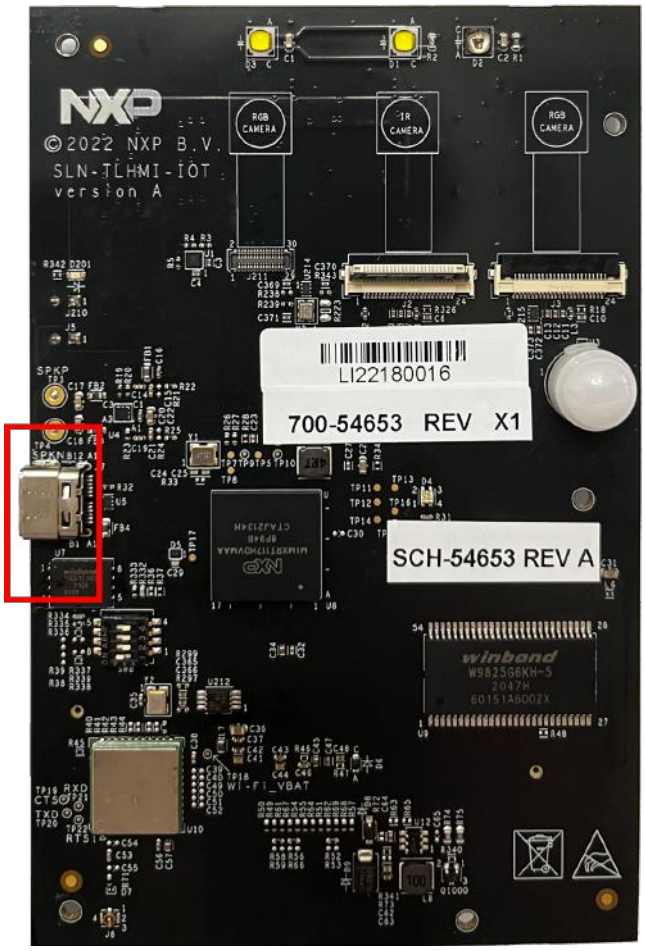


Figure 49. Power on SLN-TLMHI-IOT kit

3. Select the bootloader project in the **Project Explorer** pane.
4. To start the process of loading the binary into flash and begin debugging, choose the **Debug** option in the **QuickStart Panel**.

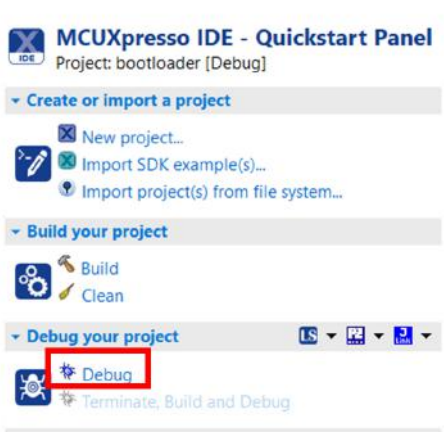


Figure 50. Debug bootloader

5. Select the J-Link probe that is connected to your kit and click the **OK** button.

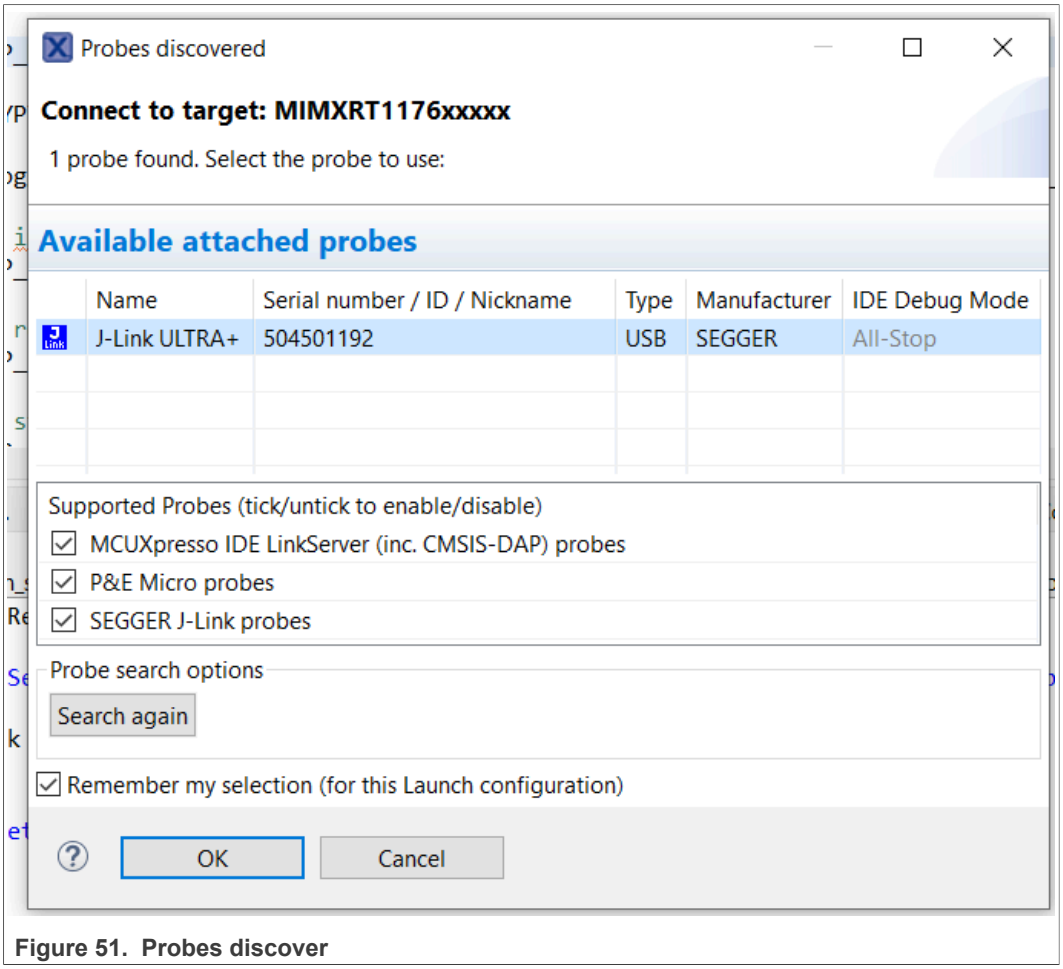


Figure 51. Probes discover

6. This launches the flashing tool. Now, proceed to flash the binary associated with the currently selected project.

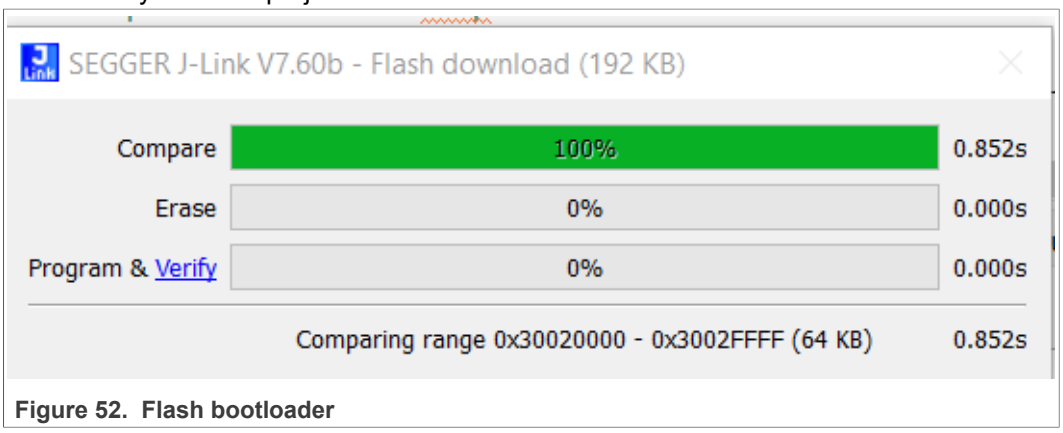


Figure 52. Flash bootloader

7. Once you have flashed the bootloader, repeat the same processes for the `sln_smart_TLHMI_coffee_machine_cm7` project.

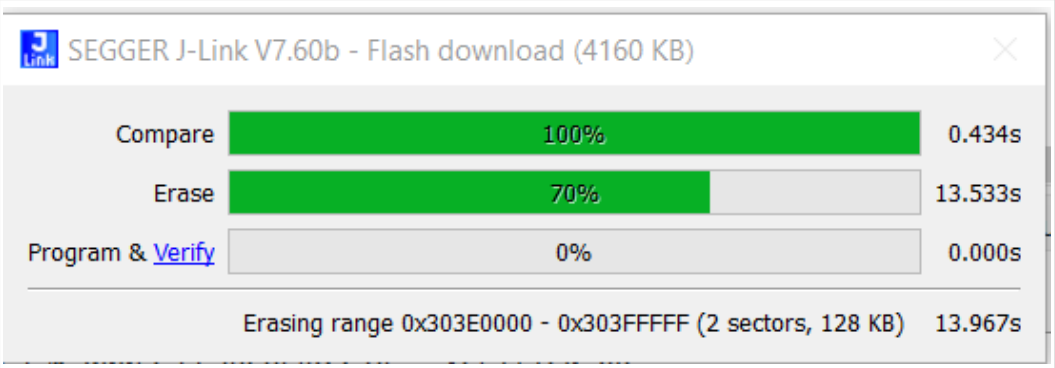


Figure 53. Flash application

8. After successfully debugging the application, the program breaks at the cm7 master project main (found in `sln_smart_TLHMI_cm7.cpp`). The cm7 and cm4 projects then launch into the GDB debug process.

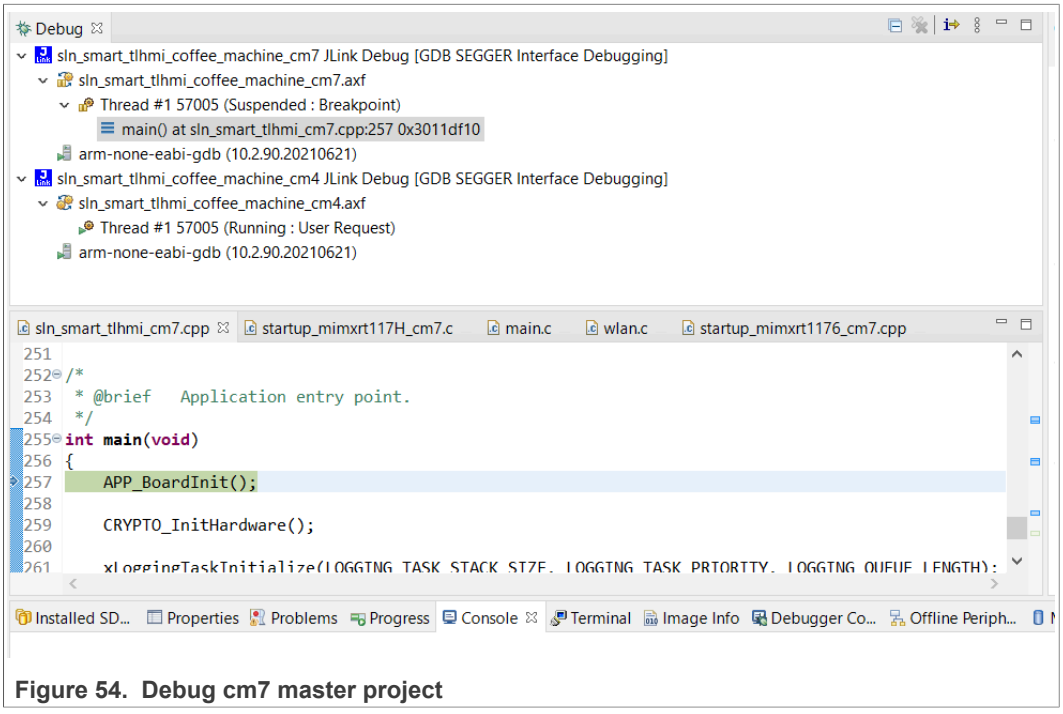


Figure 54. Debug cm7 master project

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

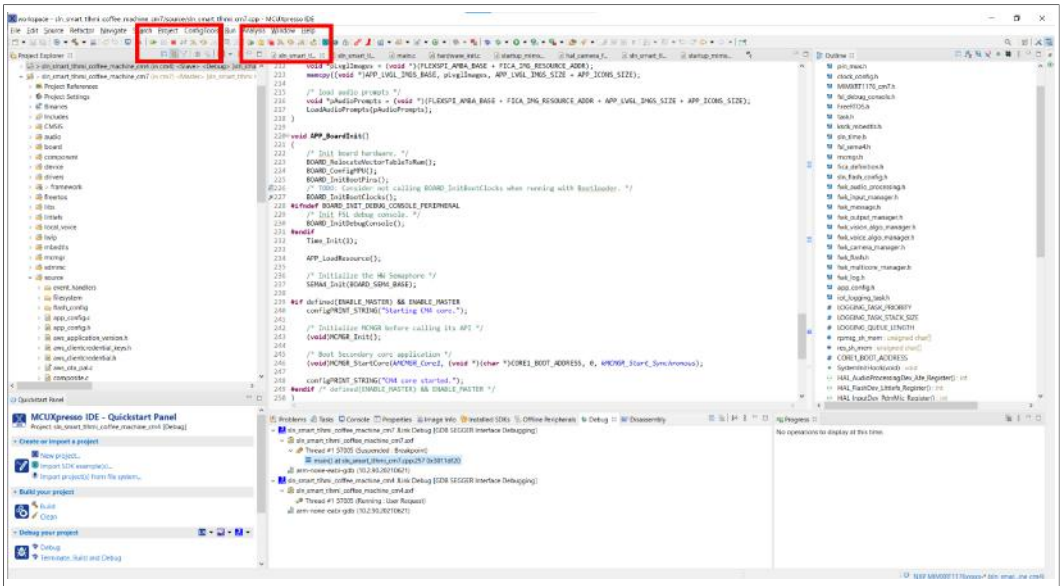


Figure 55. Single versus simultaneous core debugging

10. The highlighted controls in the left box allow you to debug a singular core. Whereas, the controls in the right box allow you to debug both cores simultaneously.

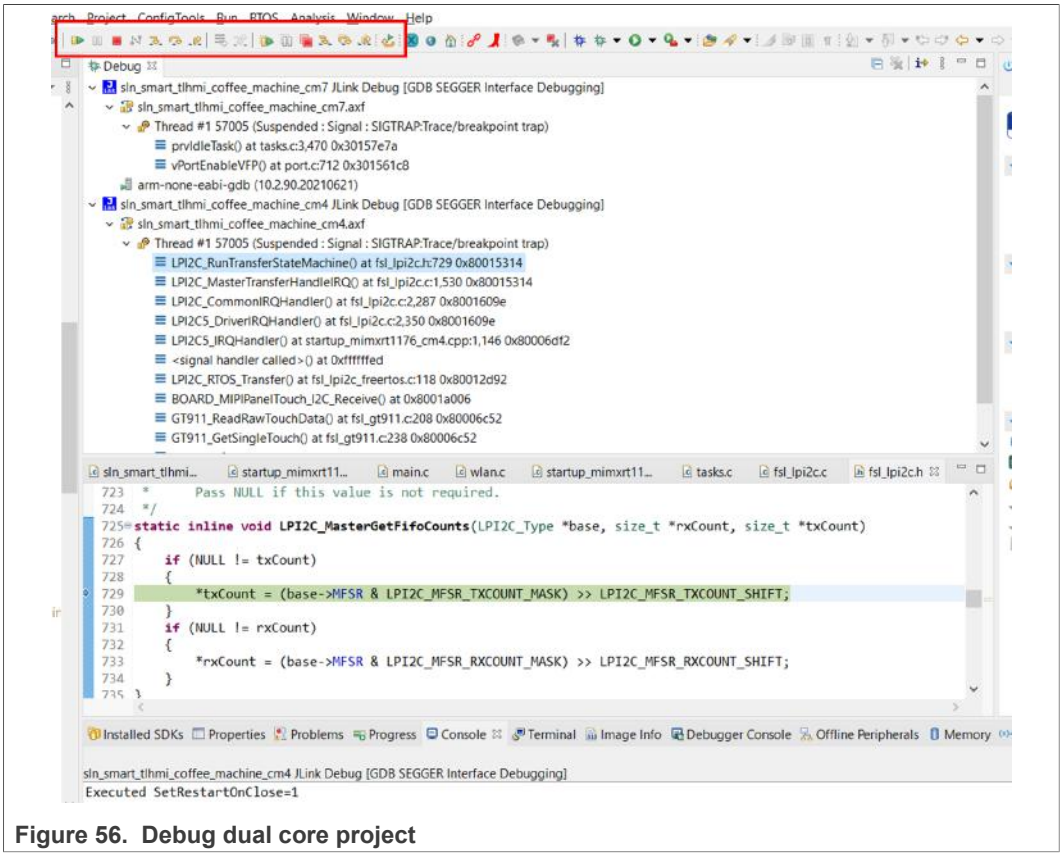


Figure 56. Debug dual core project

Note: *If you notice issues after performing the steps in this section, ensure that you do not have any mismatched FW and/or face database versions remaining. To reset to a known good state, follow the instructions found in the* ***Merged binary*** *section of the Smart HMI Software Development User Guide.*

3.4 Additional resources

By making it this far, you have successfully completed the getting started experience for the SLN-TLHMI-IOT!

For a comprehensive understanding of all the Out-of-Box Experience (OoBE) features, including the additional demo applications that come flashed with the kit, be sure to check out the *Smart HMI User Guide*.

To start building your own applications and learn more about the software architecture, available developer tools, and more, head over to the *Smart HMI Software Development User Guide*.

4 Revision history

Table 1 summarizes the changes done to this document since the initial release.

Table 1. Revision history

Revision number	Date	Substantive changes
0	29 September 2022	Initial release

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.

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.

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.

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 or registered trademarks of Arm Limited (or its subsidiaries) 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.

Contents

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

Date of release: 26 September 2022
Document identifier: MCU-SMHMI-GSG