AN13644

Getting started with i.MX RT Industrial Drive Development Platform

Rev. 1.2 — 8 April 2024 Application note

Document information

Information	Content
Keywords	i.MX, RT1170, multi-motor control, TSN
Abstract	This document explains how to get started with the NXP i.MX RT Industrial Drive Development Platform. It provides detailed instructions to assemble the hardware and program and run the NXP i.MX RT Industrial Drive Development Platform demo application. This allows users to control and monitor up to 4 PMSM motors using either FreeMASTER or a TSN master device.

1 Abbreviations

Table 1 summarizes the abbreviations used in this document.

Table 1. Abbreviations

Acronym	Description
ACIM	AC Induction Motor
BLDC	Brushless DC
EVK	Evaluation Kit
GUI	Graphic User Interface
IDE	Integrated Development Environment
PMSM	Permanent Magnet Synchronous Motor
PSB	Power Stage Board
RTC	Real Time Clock
SDK	Software Development Kit
SNVS	Secure Non-Volatile Storage
SWD	Serial Wire Debug
TSN	Time Sensitive Networking
TTL	Transistor-Transistor-Logic
UART	Universal Asynchronous Receiver/Transmitter

2 Introducing the i.MX RT Industrial Drive Development Platform

The **i.MX RT Industrial Drive Development Platform** is a flexible modular-board kit that speeds up the development, evaluation, and validation of complex multi-motor control applications for industrial robots, mobile robotics, multi-axis machinery, digital manufacturing, and many other industrial use cases.

The i.MX RT Industrial Drive Development Platform demonstrates how an **NXP i.MX RT1170 crossover MCU** can be applied to simultaneously control up to four Permanent Magnet Synchronous Motors (PMSMs) while handling advanced functionalities such as data logging, fault detection, deterministic connectivity (Ethernet TSN) and complex user interfaces. By leveraging an NXP EdgeLock SE05x secure element for the secure storage of keys and credentials, the i.MX RT1170 MCU also supports strong cybersecurity based on the latest, most secure cryptographic algorithms and protocols, therefore opening the path to achieving some of the highest security levels of the *ISA/IEC 62443-4-2* industrial standard.

The i.MX RT Industrial Drive Development Platform comes with a fully-featured hardware and software package that allows users to start developing multi-motor control and other industrial applications quickly:

- The i.MX RT Industrial Drive Development Platform **hardware package** consists of a daughter card integrating the i.MX RT1170 crossover MCU, a digital board to expand the interfaces available to the daughter card and up to four power stage boards to transform control commands into power signals for driving up to four servo motors. All the boards can be configured and adapted to meet the specific requirements of the application that is being developed. More information on the hardware package can be found in *i.MX RT Industrial Drive Development Platform hardware overview* (document AN13642).
- The **i.MX RT Industrial Drive Development Platform software package** consists of a reference demo application and API that demonstrate how to take advantage of the i.MX RT Industrial Drive Development Platform hardware's capabilities to develop a secure, robust, and reliable multi-motor control system that meets the requirements, standards, and best practices required by industrial products. This significantly

reduces the effort required to develop multi-motor control applications, in turn reducing the time-to-market of the product. More information on the software package can be found in *i.MX RT Industrial Drive Development Platform software overview* (document AN13643).

2.1 How to use this document

This document provides step-by-step instructions to assemble the i.MX RT Industrial Drive Development Platform hardware and run the demo application. At the end of this document, you should be able to control and monitor four motors simultaneously using NXP's run-time debugging tool (FreeMASTER) and also be able to receive motor control commands from a TSN master device. The document is structured as follows:

- Section 3 lists the hardware and software material required to run the i.MX RT Industrial Drive Development Platform demo;
- Section 4 describes how to assemble the different hardware components of the i.MX RT Industrial Drive Development Platform;
- Section 5 describes how to install in the PC the tools required to compile and debug the i.MX RT Industrial Drive Development Platform demo;
- Section 6 describes how to import, compile, flash, and execute the i.MX RT Industrial Drive Development Platform demo application;
- Section 7 describes how to control the motors connected to the i.MX RT Industrial Drive Development Platform using the FreeMASTER tool;
- Section 8 describes how to control the motors connected to the i.MX RT Industrial Drive Development Platform using a TSN master device.

3 Hardware and software required

This section lists the hardware and software material needed for executing the i.MX RT Industrial Drive Development Platform demo application. The required hardware components are listed in Table 2. The required software tools are listed in Table 3.

Note: *Additional hardware components are required to control motors using a TSN master device. See Section 8 for more information.*

Table 2. Hardware material

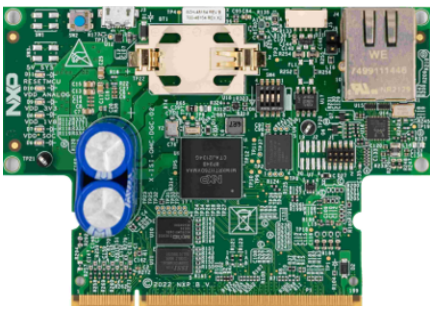
Component name	Description	Picture
Daughter card *(ISI-QMC-DGC02)*	The daughter card provides an i.MX RT1176 dual-core crossover MCU featuring a 1 GHz (800 MHz in industrial qualified version) Arm Cortex-M7 core and a 400 MHz Arm Cortex-M4 core.	

Table 2. Hardware material*...continued*

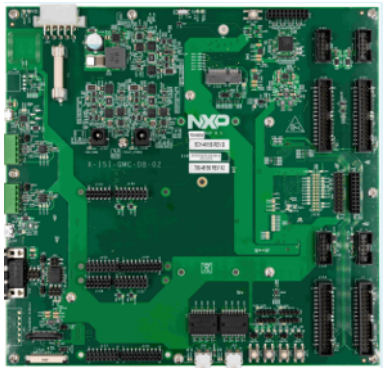
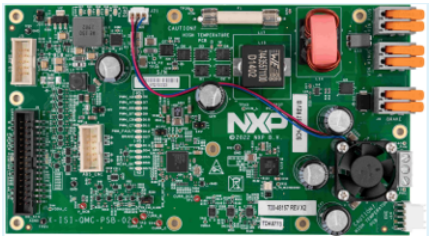
Component name	Description	Picture
Digital board *(ISI-QMC-DB02)*	The digital board works as an external platform to prototype multi-motor control applications. It includes widely used industrial communication and peripheral interfaces.	
Power stage board *(ISI-QMC-PSB02 or ISI-QMC-PSB02B)*	The power stage board includes a 3-phase inverter with the gate driver for the control and connection of the motors. Up to 4 power stage boards can be connected to the digital board to control up to 4 motors through the i.MX RT1170 crossover processor. The power stage board must be supplied through an external power source using connectors. The maximum voltage allowed is 55 V, and the Over Current (OC) peak level is 14.9 A. This board houses the 3-phase MOSFET-based inverter with Gate driver NXP GD3000 gate driver IC, connections port for incremental encoder, and Analog Front End IC 13388 for temperature monitoring.	
3-phase PM Synchronous Motors (PMSM	i.MX RT Industrial Drive Development Platform is able to control up to 4 PMSM 3-phase motors. In this document *Teknic Industrial-Grade NEMA 23 motors* (*M-2310P-LN-04K*) are used. Other motors can be used as well; however, it is recommended to use motors with similar characteristics (maximum voltage 30 V, maximum power 200 W, incremental TTL encoder). ***Note:*** *It is not included in the i.MX RT Industrial Drive Development Platform hardware package.*	
MCU-Link Pro debug probe *(or Segger J-Link debug probe)*	The MCU-Link Pro is a fully featured debug probe that can be used with the MCUXpresso IDE. Alternatively, it is also possible to use Segger J-Link debug probes. ***Note:*** *It is not included in the i.MX RT Industrial Drive Development Platform hardware package.*	

Table 2. Hardware material*...continued*

Component name	Description	Picture
Power supply	A power supply that can provide enough voltage and current to power the motors connected to the power stage boards. **It is recommended not to go beyond a maximum voltage of 30 V to power the motors**. In this document we apply a voltage of 24 V and limit the current to 3 A. ***Note:*** *It is not included in the i.MX RT Industrial Drive Development Platform hardware package.*	
CR2032 coin cell battery *(Optional)*	A CR2032 coin cell battery is used for powering the SNVS_LP (low power) of the i.MX RT1176 MCU in the daughter card for RTC backup. ***Note:*** *It is not included in the i.MX RT Industrial Drive Development Platform hardware package.*	
Cables and wires	Cable and wires to connect different hardware components of the i.MX RT Industrial Drive Development Platform: • **30-pin flat cables** and **10-pin flat cables** to connect the power stage boards to the digital board. These cables are included in the power stage board package; • A **micro-USB cable** to connect the digital board to the PC; • An **SWD cable** to connect the debug probe to the daughter card. This cable is typically included in the debug probe package; • Wires to connect motors to power stage boards. These cables are not included in the i.MX RT Industrial Drive Development Platform; • Wires to connect the power supply to power stage boards. These cables are not included in the i.MX RT Industrial Drive Development Platform.	
LCD	The digital board supports the connection of an MIPI DSI LCD that can be used as a user interface to display data and, if touch functionality is supported, as an input interface as well. Two MIPI DSI connectors (J132 and J44, respectively) are provided to connect the 5.5" LCD panel (NXP Semiconductors module or a Raspberry Pi compatible LCD). The J53 jumper is used to connect the Raspberry Pi compatible LCD to the 5-V external power supply.	

Table 3. Software tools

Name	Description
MCUXpresso IDE	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. **The minimum supported version of MCUXpresso is 11.7.1**.
FreeMASTER	FreeMASTER is a user-friendly real-time debug monitor and data visualization tool that enables runtime configuration and tuning of embedded software applications. This tool is used to control and monitor the motors connected to the i.MX RT Industrial Drive Development Platform. **The minimum supported version of FreeMASTER is 3.2**.

Note: *A Windows PC has been used to run the software tools listed in Table 3. Other operating systems might be supported as well.*

4 Hardware preparation

This section describes how to connect different hardware components of the i.MX RT Industrial Drive Development Platform. After completing this section you should be able to power up the system which consists of a daughter card, a digital board and up to four power stage boards, each one controlling a PMSM motor.

- Prepare the daughter card and connect it to the digital board
- Connect the power stage board to the digital board
- Connect the motors and power up the system
- Connect LCD with digital board

4.1 Prepare the daughter card and connect it to the digital board

The daughter card is the core of the i.MX RT Industrial Drive Development Platform as it provides the i.MX RT1176 crossover MCU that controls the whole system. The daughter card can control up to four motors simultaneously. To do this, the daughter card first must be connected to the digital board that contains provision for peripheral connections. Up to four power-stage boards can then be connected to the digital board, each controlling one motor.

To prepare the daughter card and connect it to the digital board, follow these steps :

1. *(Optional)* Place a **CR2032 coin cell battery** in the coin cell holder of the daughter card as shown in Figure 1. The coin cell battery is required to power the *SNVS_LP* (Low Power) of the i.MX RT1176 MCU for RTC backup.
   Note: *This step is optional. The daughter card can operate without a coin cell battery.*

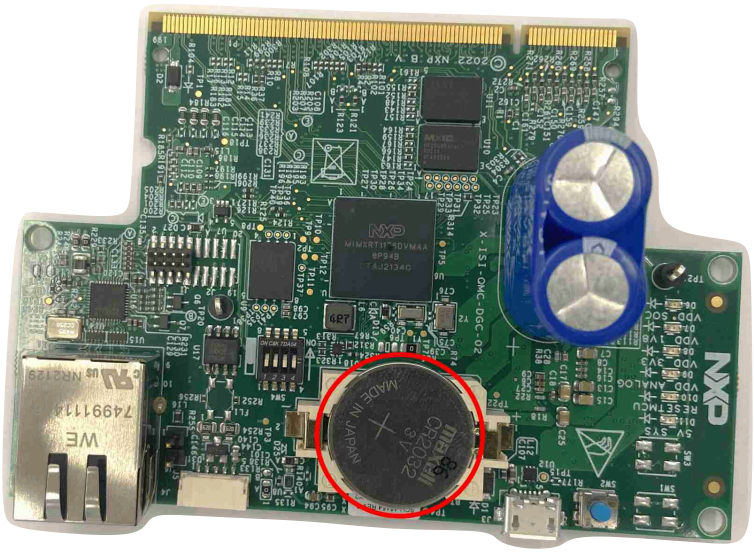


Figure 1. Insert a CR2032 coin cell battery in the coin cell holder of the daughter card

2. Check that the DIP switch configuration of the daughter card is the one shown in Figure 2 (*SW4.1: ON; SW4.2: OFF; SW4.3: OFF; SW4.4: OFF* – Serial Downloader Mode (SDP)):

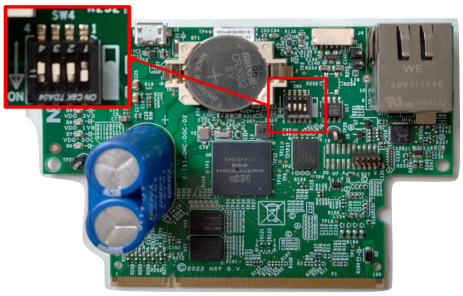


Figure 2. Daughter card DIP switch configurations

3. To check that the daughter card is properly working, connect it to a 5V power supply (for example, a laptop) using its micro-USB connector (*J3*) as shown in Figure 3. Status LEDs *D6, D7, D8, D9, and D11* should turn green if the board is correctly supplied.
   Note: *The daughter card integrates a superCAP that can power the daughter card for a few seconds when power is cut off (for example, to save important data before shutting down the system). It takes approximately 10 seconds for the superCAP to charge completely. Discharge of the superCAP takes approximately 30 seconds (all LEDs will turn off).*
   Note: *The micro-USB connector must be used to power the board ONLY when the daughter card is used as a standalone device.*

Figure 3. Daughter card supplied using the micro-USB connector (standalone operation)

4. Remove the USB cable from the daughter card, then connect it to the bottom side of the digital board using the *SODIMM-200* connector as shown in Figure 4:

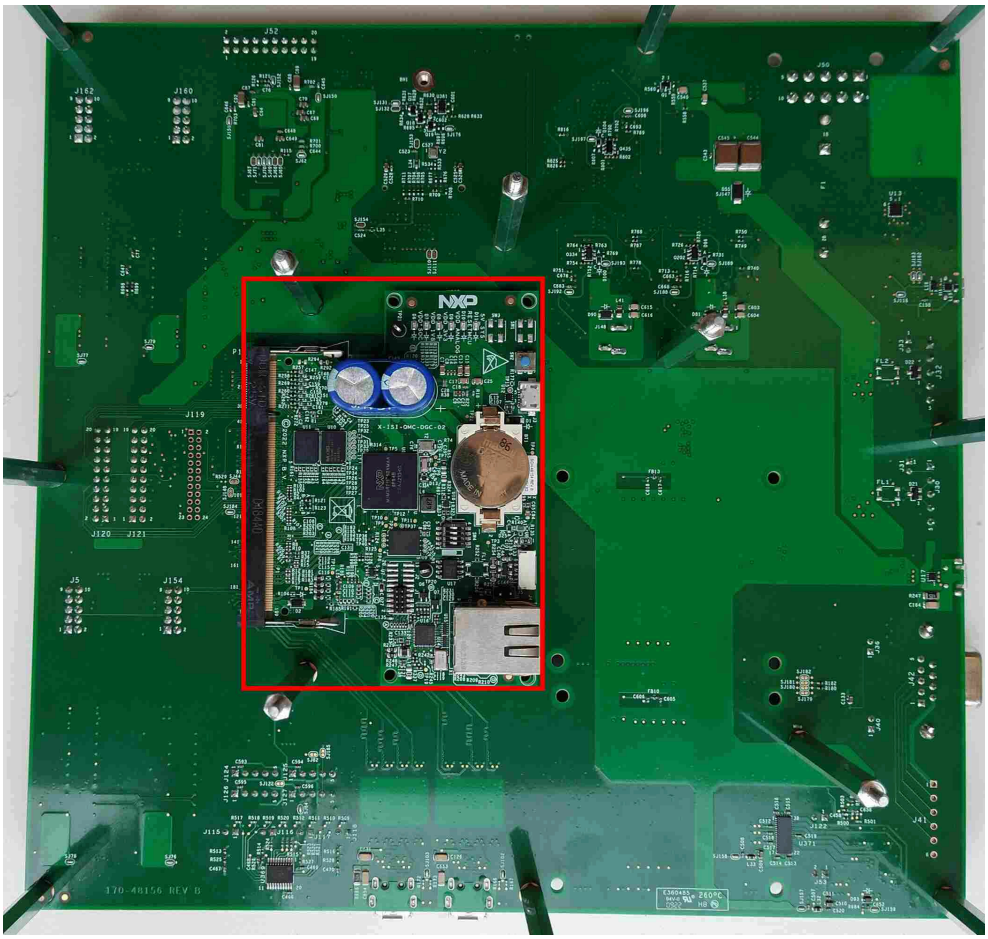


Figure 4. Connect the daughter card to the digital board

5. Check that the daughter card is properly connected to the digital board as shown in Figure 5:
   (1) Connect the digital board to a 5V power supply (for example, a laptop) through a micro USB cable (*J48* connector);
   (2) Turn the switch *SW1* into the *ON* position;
   (3) If the digital board is correctly supplied, the *D94* LED turns on;
   (4) If the daughter card is correctly connected and supplied, the status LEDs *D6, D7, D8, D9, and D11* in the daughter card turn on as well.
   Note: *The micro-USB connector must be used to supply the digital board only when no power stage board is connected. If at least a power stage board is connected to the digital board, then the power supply will come from the individual stages connected to the digital board.*
   Note: *The J48 micro-USB connector will also be used for FreeMASTER communication.*

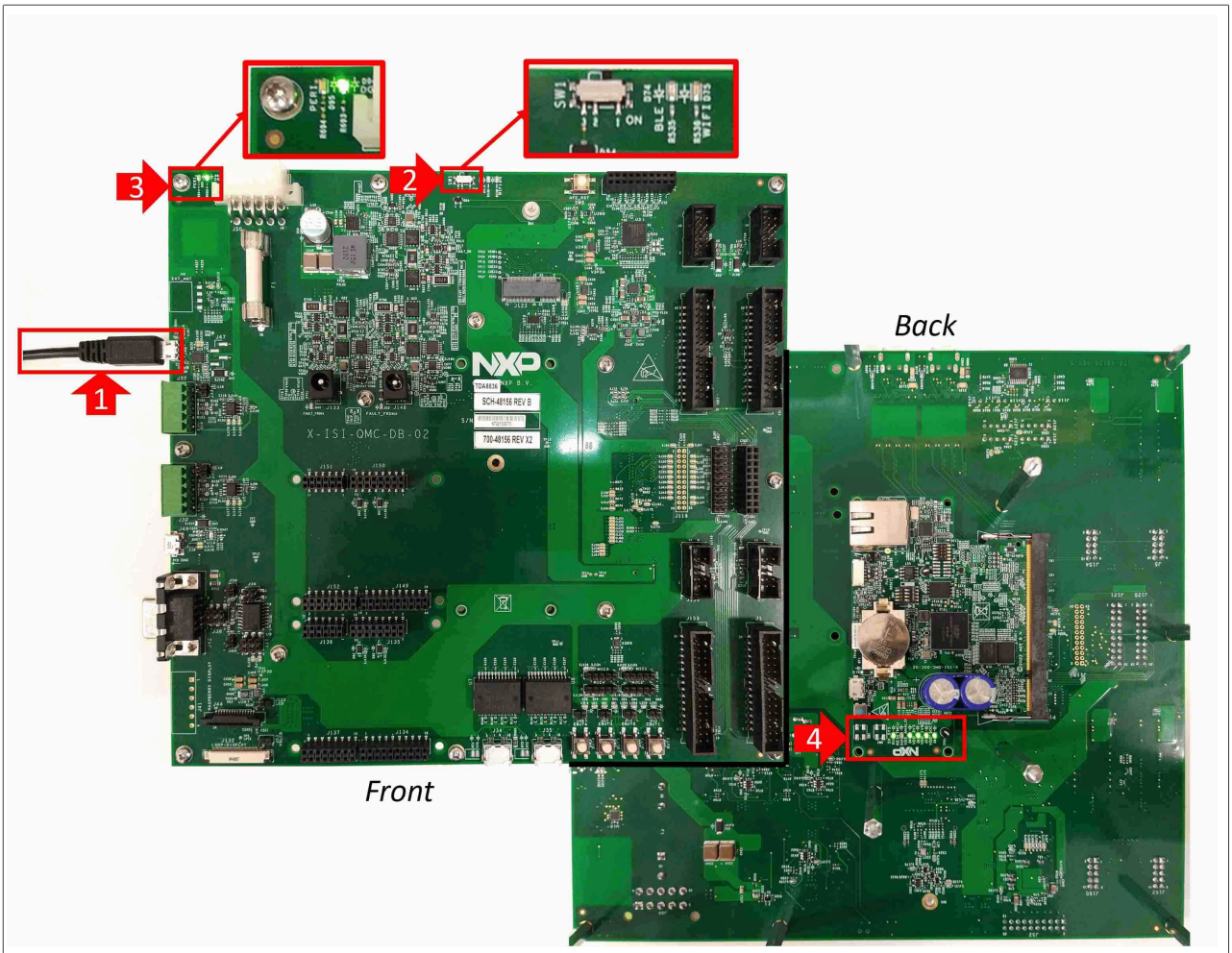


Figure 5. Digital board supplied using the micro-USB connector (standalone operation)

4.2 Connect the power stage board to the digital board

Up to 4 power stage boards can be connected to the digital board, each one supporting the connection and control of a 3-phase motor. Follow these instructions to connect one or more power stage boards to the digital board:

1. The digital board provides four pairs of 30-pin and 10-pin connectors. The 30-pin connectors *(J1, J153, J159, J161)* are used for signal interconnection between the digital board and the power stage boards. The 10-pin connectors (*J5, J154, J160, J162*) are used to supply the digital board and the daughter card with power provided by the power stage boards. Connectors are shown in Figure 6:
   (1) Connectors for the power stage board controlling motor 1 (*J153, J154*);
   (2) Connectors for the power stage board controlling motor 2 (*J161, J162*);
   (3) Connectors for the power stage board controlling motor 3 (*J1, J5*);
   (4) Connectors for the power stage board controlling motor 4 (*J159, J160*).

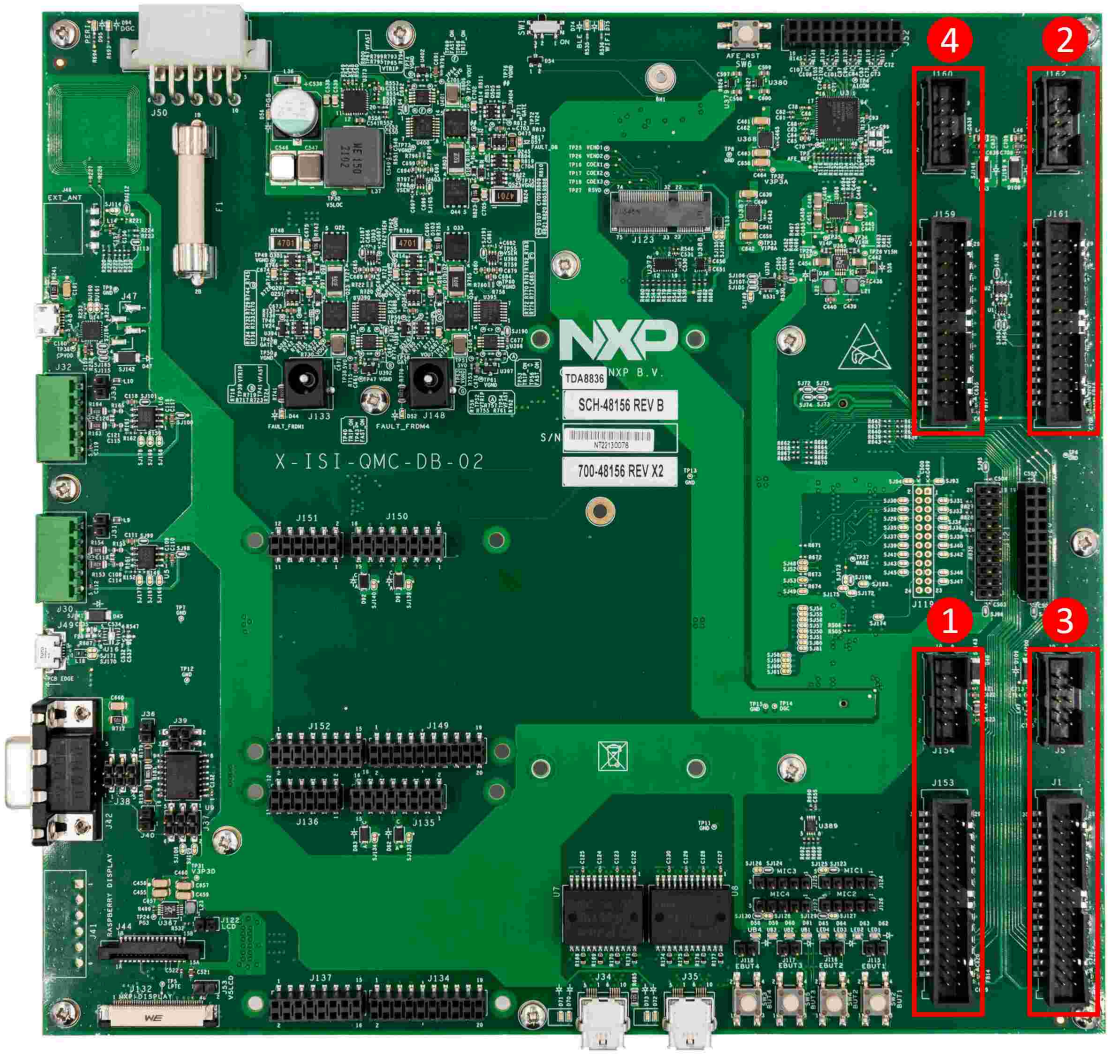


Figure 6. Power stage board connectors in the digital board

2. Connect the power stage boards to the digital board using the connectors described in the previous step. Figure 7 shows as an example the connection of the power stage board controlling *motor 1*:
(1) Use a 10-pin flat cable to connect the *J154* connector in the digital board to the *J5* connector in the power stage board;
(2) Use a 30-pin flat cable to connect the *J153* connector in the digital board to the *J1* connector in the power stage board.

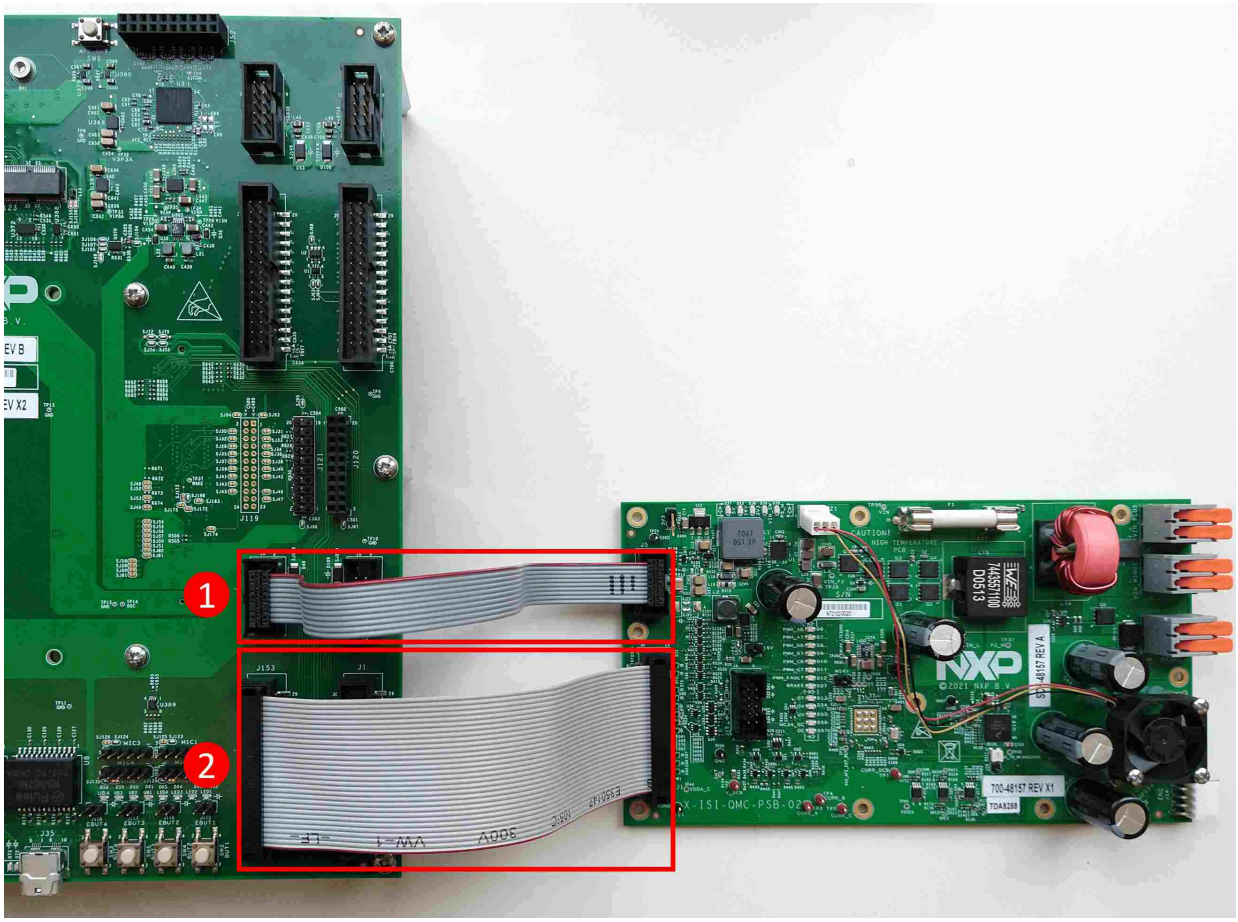


Figure 7. Connect a power stage board to the digital board

3. Repeat the above step for all the power stage boards that you want to connect to your system. You can stack two power stage boards on top of one another using the standoffs and screws included in the i.MX RT Industrial Drive Development Platform hardware package. If you connect all power stage boards, the result must be similar to Figure 8:

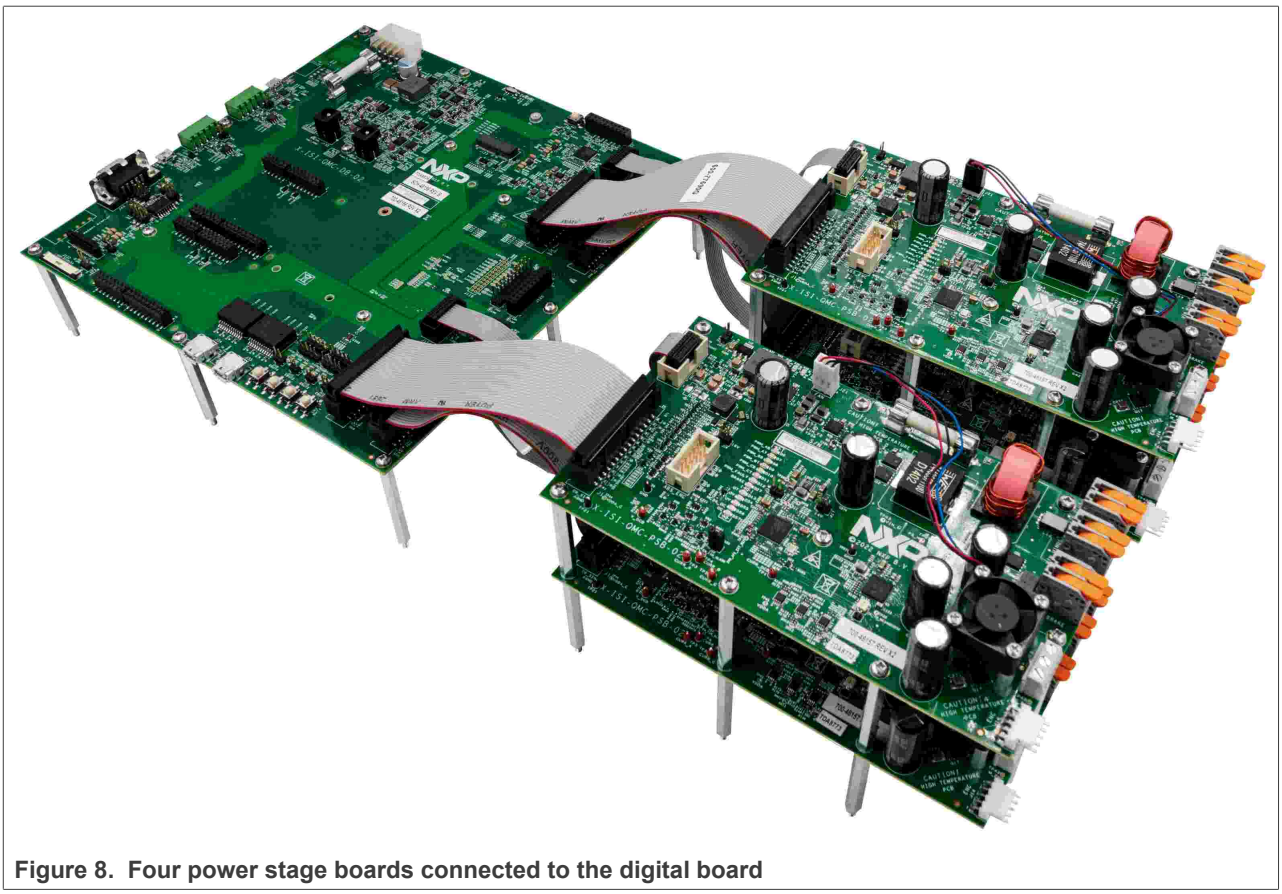


Figure 8. Four power stage boards connected to the digital board

4.3 Connect the motors and power up the system

To connect the PMSM motors to the power stage boards and supply and power up the system, follow these steps:

1. Identify in the motor connector the *incremental encoder signals* (*5V, GND, ENC_A, ENC_B, ENC_I*) and the *motor phase signals* (*R, S, T* or *A, B, C*). These signals must be routed to the power stage board using the proper wires based on the motor manufacturer recommendation for wire gauges. The *Teknic Industrial-Grade NEMA 23 motor* (*M-2310P-LN-04K*) used as reference in this document has the pinout shown in Figure 9:

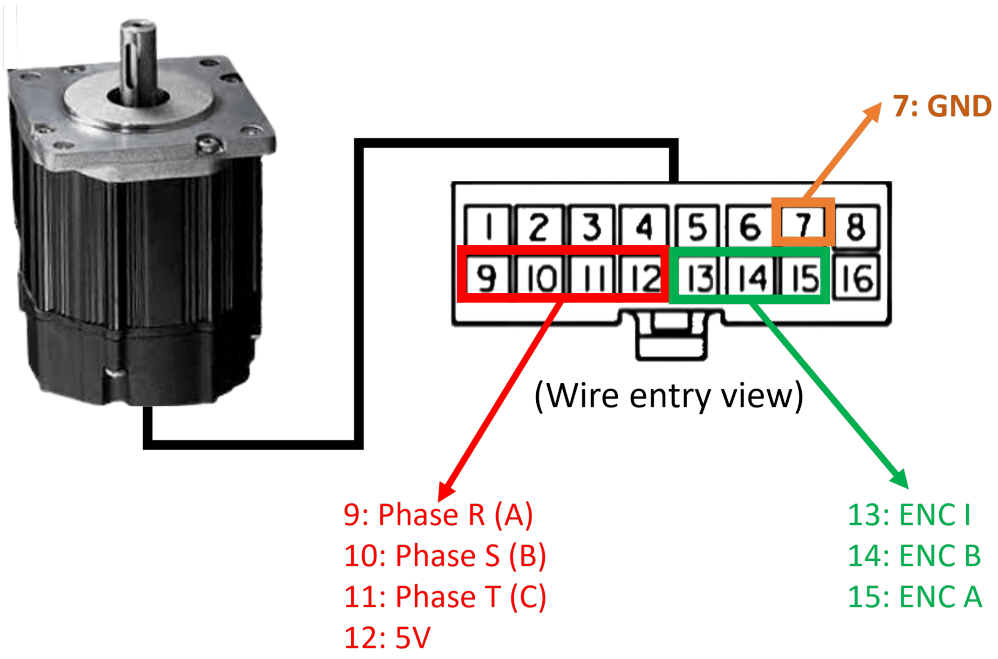


Figure 9. Teknic Industrial-Grade NEMA 23 motor - Connector pinout

2. Connect the motors to the power stage boards as shown in Figure 10:
(1) Encoder signals of the motor must be routed to the 5-pin *J14* connector of the power stage board as described in Table 4:

Table 4. Mapping of J14 pins to incremental encoder signals

J14 pin	Signal
1	5VDC_IN
2	GND
3	ENC_A
4	ENC_B
5	ENC_I

(2) The three phases of the motor must be routed to the *J13* connector of the power stage board as described in Table 5:
Note: *Loosen the screws of the connector before inserting the wires, then tighten them again.*

Table 5. Mapping of J13 pins to phase signals

J13 pin	Signal
1	PHASE R (A)
2	PHASE S (B)
3	PHASE T (C)

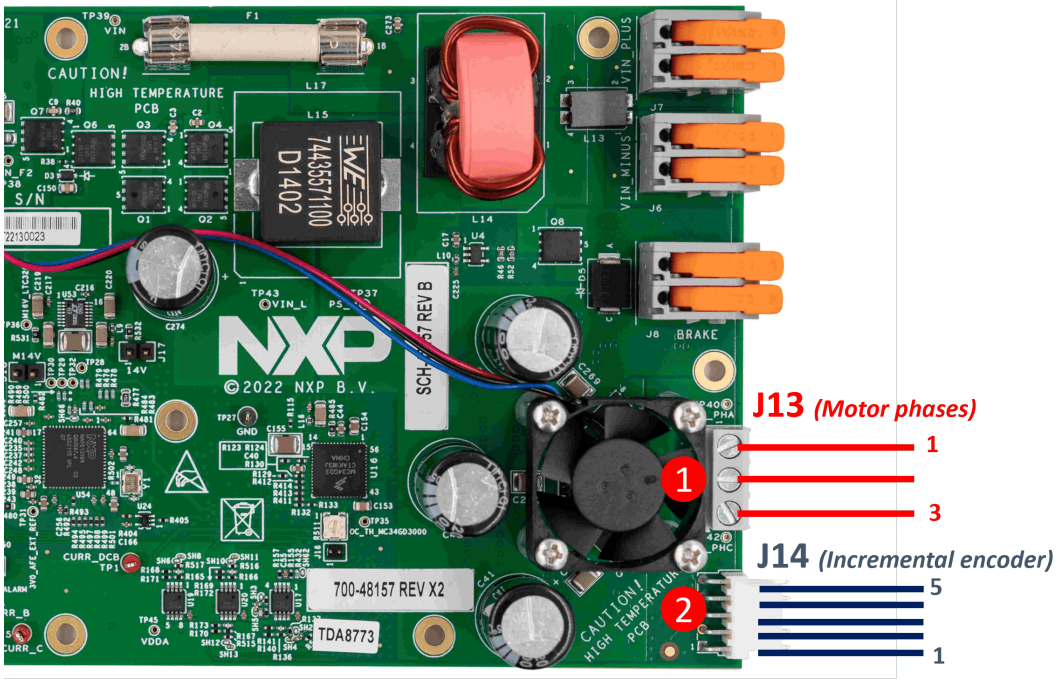


Figure 10. Connect a motor to the power stage board (incremental encoder and phase signals)

3. Connect the power supply to the power stage boards as shown in Figure 11:
   (1) Connect the positive wire of the power supply to any of the two pins of the *J7* connector of all connected power stage boards. See Figure 11;
   (2) Connect the ground wire of the power supply to any of the two pins of the *J6* connector of all connected power stage boards. See Figure 11;
   (3) Set the maximum output voltage of the power supply unit to *24 V* and the maximum current output to *0.5 A*, then turn on the power supply unit;
   (4) If power stage boards are correctly supplied, the status LEDs of the power stage boards should turn green and fans should start spinning. The digital board and the daughter card should turn on as well if the power stage boards are correctly connected to the digital board;
   Note: *Make sure that the ON/OFF switch (SW1) of the digital board is in the ON position; otherwise power will not be supplied to the digital board.*

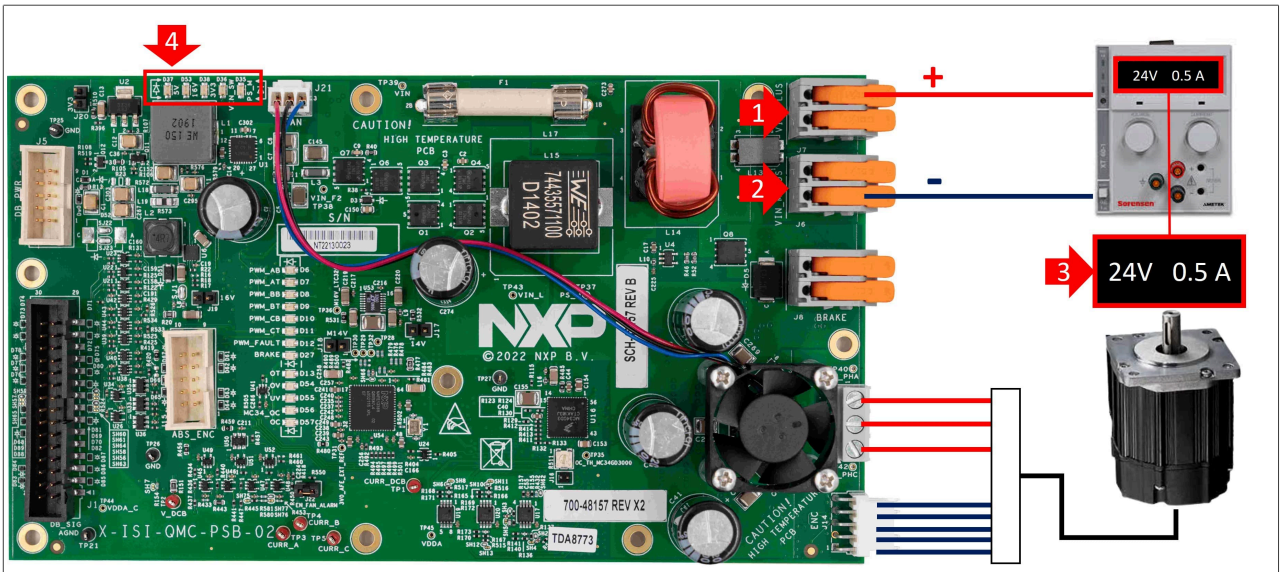


Figure 11. Connect the power supply to a power stage board

4. If the power stage boards are correctly supplied and no fault has occurred, increase the maximum output current of the power supply unit to *3 A*. At the end, the whole system must look as shown in Figure 12.

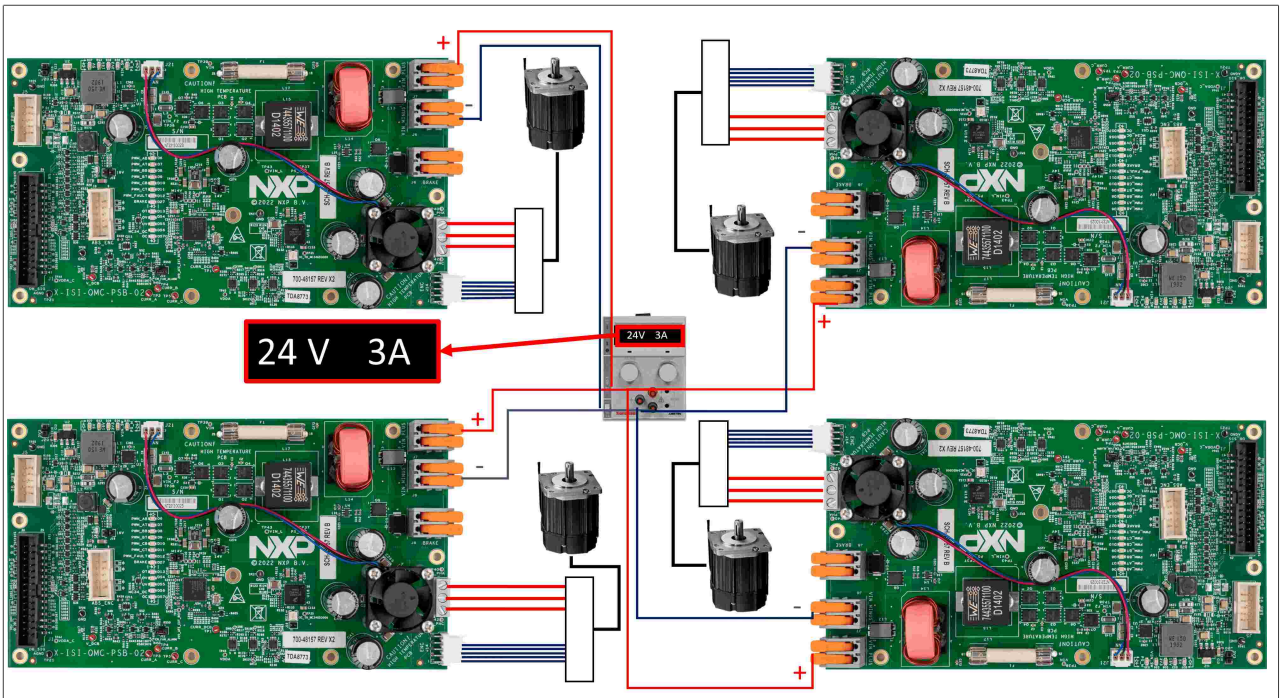


Figure 12. Connect the power supply to 4 power stage boards

4.4 Connect LCD with digital board

The digital board supports the connection of a MIPI DSI LCD that can be used as a user interface to display data and, if touch functionality is supported, as an input interface as well. Two MIPI DSI connectors (J132 and J44, respectively) are provided to connect the RK055HDMIPI4MA0 NXP LCD module or a Raspberry Pi

compatible LCD. The J53 jumper is used to connect the Raspberry Pi compatible LCD to the 5-V external power supply.

To enable the J132 connector, the J122 jumper must be in the open position. This enables the usage of the RK055HDMIPI4MA0 NXP LCD module.

Note: *There is a known issue with the display output. Due to the framebuffers being stored in the Octal RAM and the logs in the Octal FLASH, writes and reads to both share the same Octal bus. Race conditions can lead to display flickering and "rolling" effects. This is purely cosmetic and once the display content stabilizes, the application continues running without any issues.*

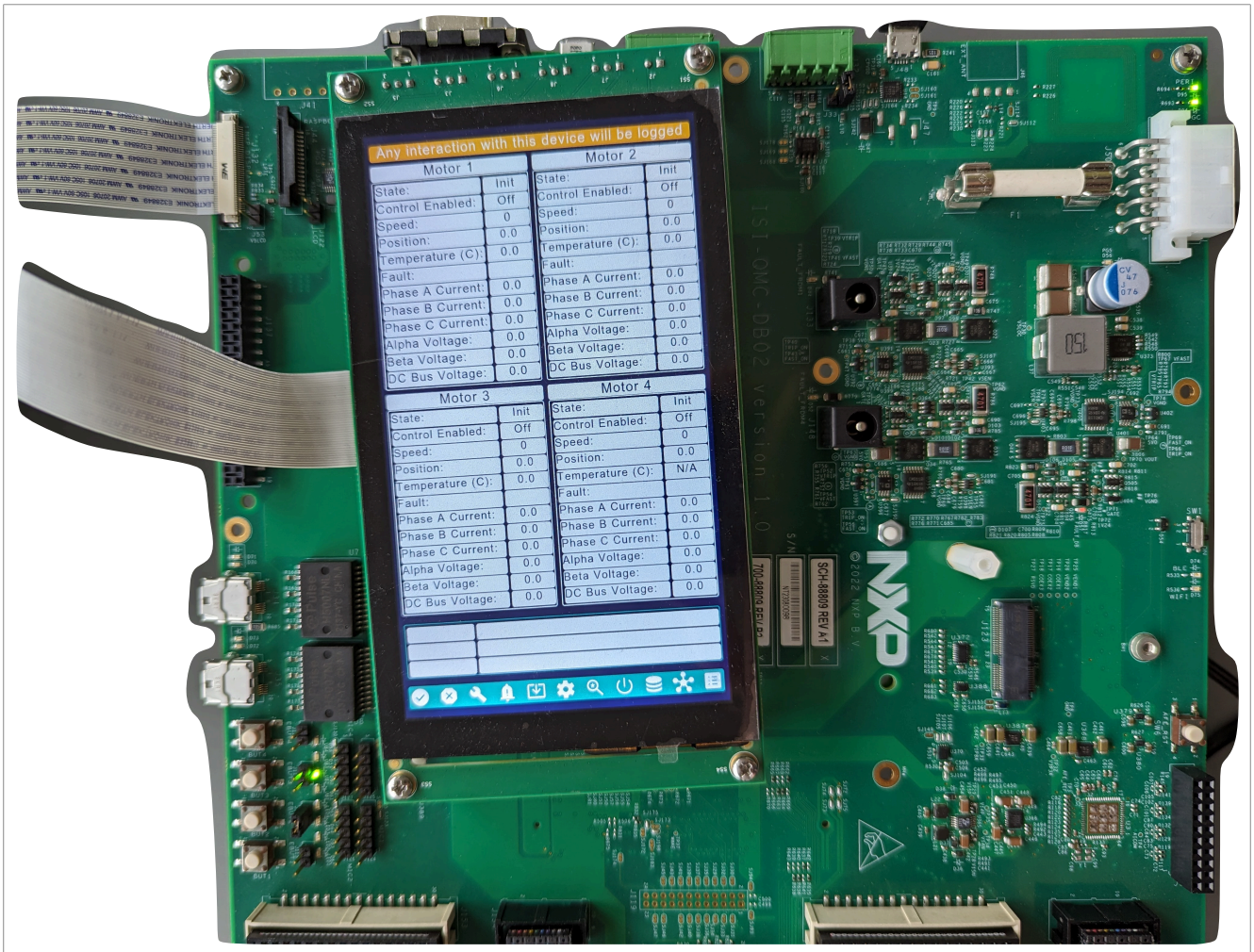


Figure 13. Connect the LCD

5 Software environment preparation

This section describes how to install in the PC the tools required to compile and debug the i.MX RT Industrial Drive Development Platform demo:

- Install the ***MCUXpresso IDE*** as described in Section 5.1;
- Install the ***FreeMASTER*** tool as described in Section 5.2;
- Configure the ***MCU-Link Pro*** debug probe as described in Section 5.3;
- Install the ***USB to UART drivers*** as described in Section 5.4.

5.1 Install the MCUXpresso IDE

MCUXpresso is an Eclipse-based development environment for NXP MCUs based on Arm Cortex-M cores offering advanced editing, compiling, and debugging features. In this document, MCUXpresso is used to build and run the i.MX RT Industrial Drive Development Platform software project. To install MCUXpresso, follow the steps below:

1. Go to the MCUXpresso webpage and click the *downloads* button as shown in Figure 14:
   Note: *To download MCUXpresso, you need an NXP account.*

Figure 14. Download MCUXpresso IDE - homepage

2. Download the latest available version of MCUXpresso (under the *current* tab) as shown in Figure 15:
   Note: *The minimum recommended version of MCUXpresso is 11.7.1. It is recommended to always use the latest available version.*

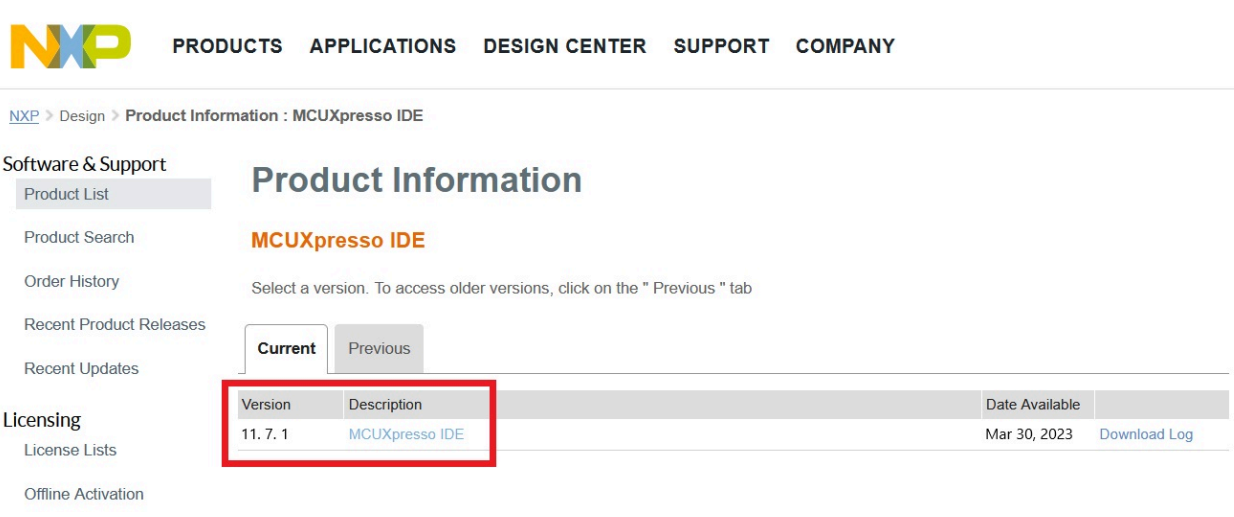


Figure 15. Download MCUXpresso IDE - Select version

3. Run the downloaded installer file and follow the setup wizard until the MCUXpresso installation is completed. Make sure you allow the installation of the additional drivers required by MCUXpresso during the installation as shown in Figure 16. Detailed information on how to use the MCUXpresso IDE can be found in the *MCUXpresso IDE User Guide* (document MCUXPRESSO-UG).

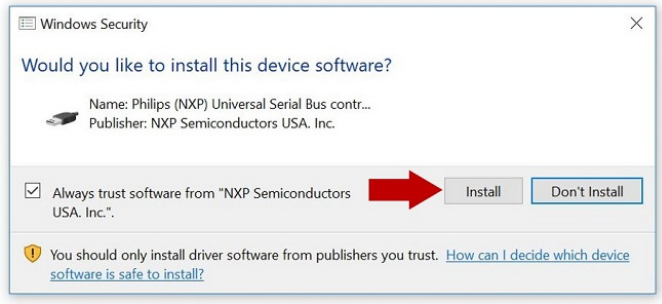


Figure 16. Install MCUXpresso IDE - Drivers

5.2 Install FreeMASTER

FreeMASTER is a user-friendly, real-time debug monitor, and data visualization tool that enables runtime configuration and tuning of embedded software applications. This tool is used to control and monitor the motors connected to the i.MX RT Industrial Drive Development Platform. To install FreeMASTER in your PC, follow the steps below:

1. Go to the FreeMASTER webpage and click the *DOWNLOADS* button, as shown in Figure 17:

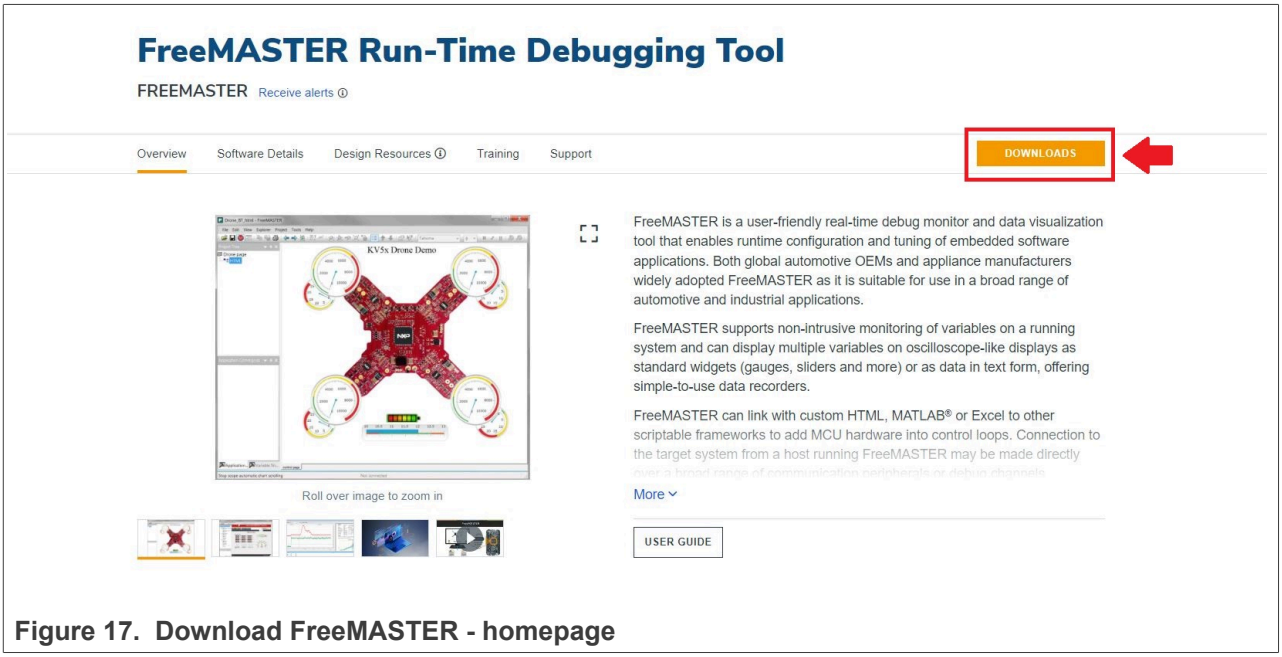


Figure 17. Download FreeMASTER - homepage

2. Click the *DOWNLOAD* button next to the FreeMASTER version for your operating system as shown in Figure 18. The FreeMASTER Windows version is used in this document. Once you have downloaded the installer, double-click the executable file to start the installation wizard.
Note: *Accept the license agreement before you can download the installer. Also, log in with your NXP account.*

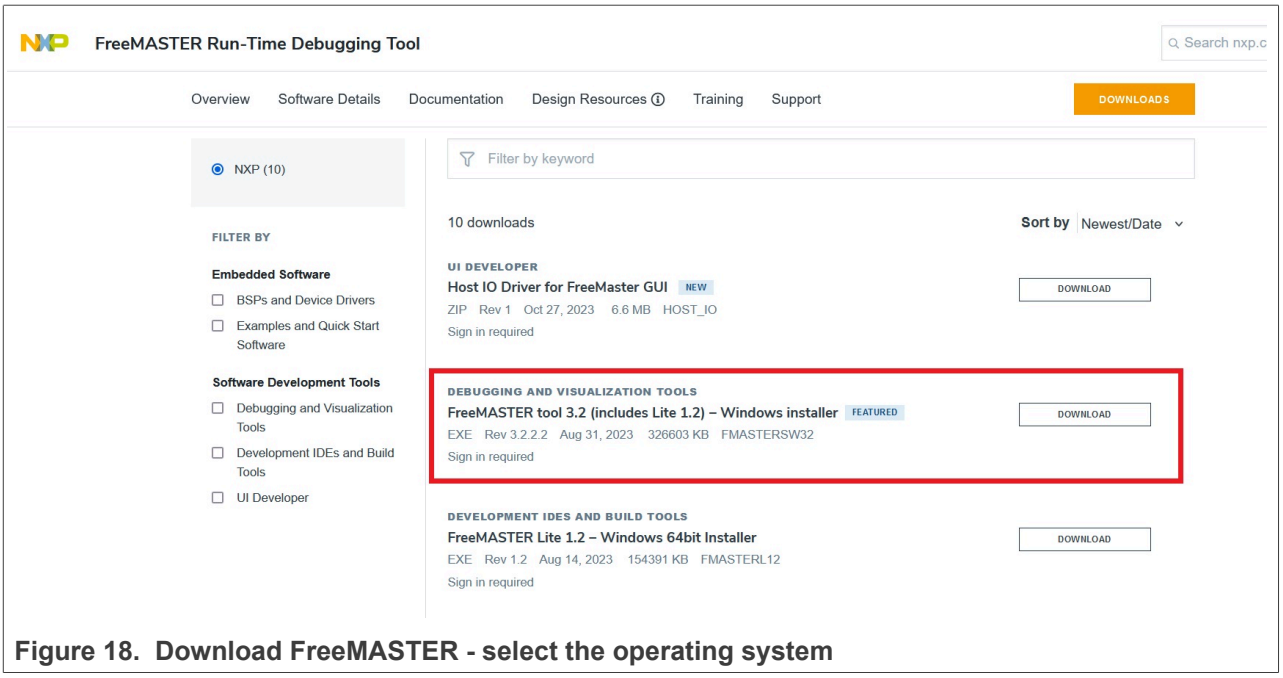


Figure 18. Download FreeMASTER - select the operating system

3. During installation, an activation code is requested to activate FreeMASTER Lite. Retrieve the activation code as shown in Figure 19:
   (1) Click the *yes* button when you are prompted to visit the NXP licensing portal to retrieve the code;
   (2) The activation key is in the *license keys* tab of the website. Insert the activation key when prompted, then continue with the installation.
   Note: *The software licensing webpage can also be accessed from this link.*

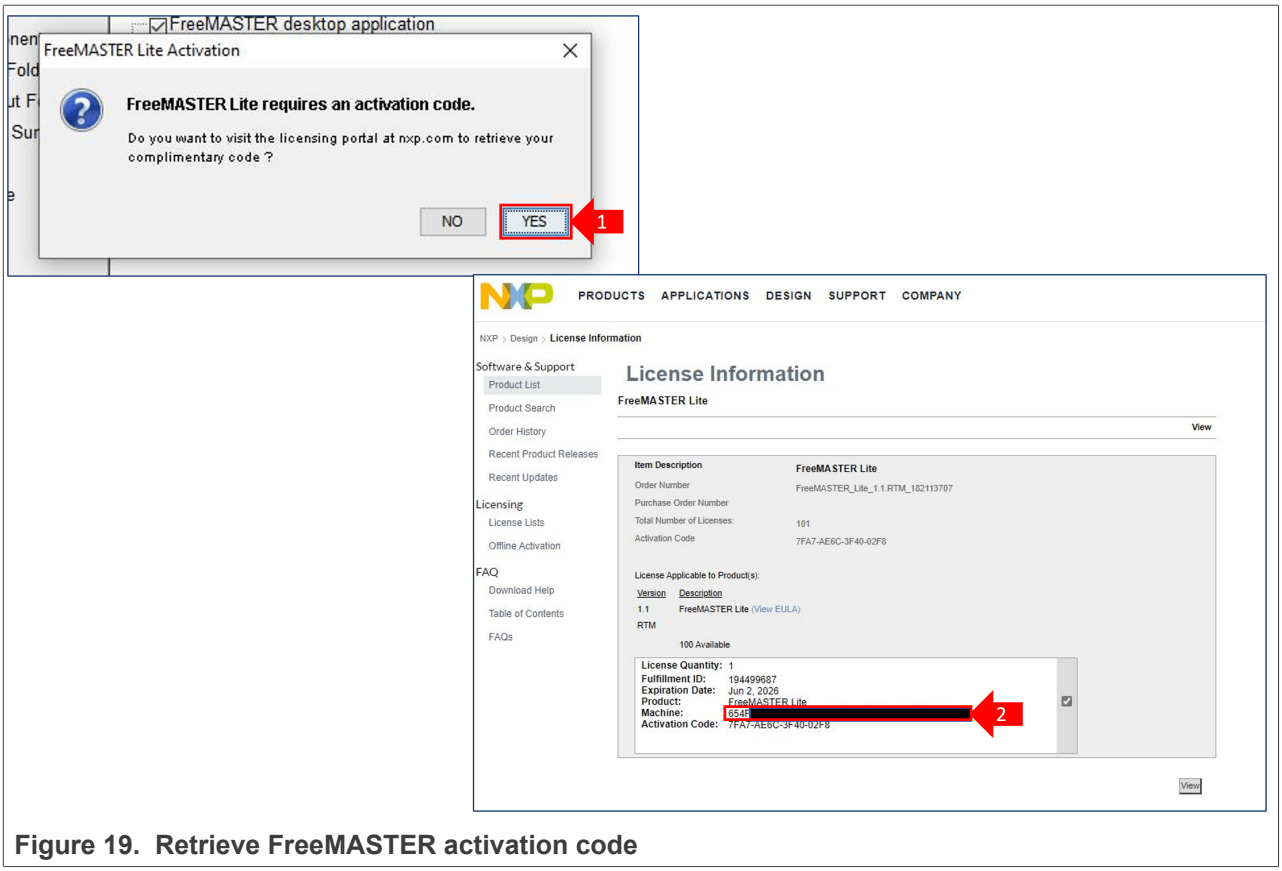


Figure 19. Retrieve FreeMASTER activation code

5.3 Configure MCU-Link Pro for J-Link debugging

The MCU-Link Pro is a fully featured debug probe that can be used with MCUXpresso IDE or other IDEs supporting CMSIS-DAP and/or J-Link protocols. Flashing the i.MX RT Industrial Drive Development Platform demo software in the daughter card requires the user to configure the MCU-Link Pro for J-Link debugging.

Follow these instructions to configure the MCU-Link Pro for J-Link debugging:

1. First, you must flash the J-Link firmware in the MCU-Link Pro. To do this, go to the MCU-Link Pro website and download the *MCU-Link installer* software for your operating system as shown in Figure 20.
   Note: *The Windows installer is used in this document.*

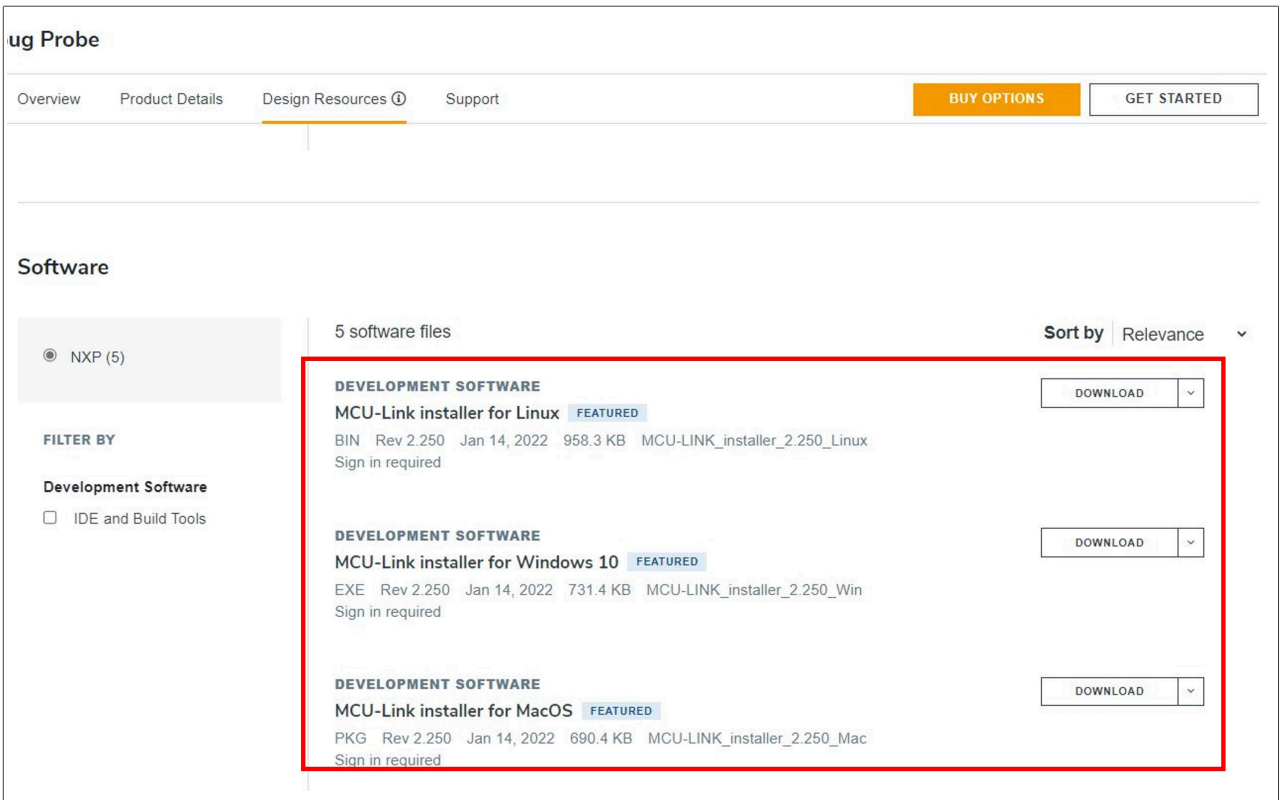


Figure 20. Flash J-Link firmware in MCU-Link Pro - Download *MCU-Link installer*

2. Run the *MCU-Link installer* executable and follow the installation wizard. By default, the installer creates an *MCU-LINK_installer* folder in *C:\nxp* as shown in Figure 21.
   Note: *The name of the folder might change depending on the version of the installer downloaded.*

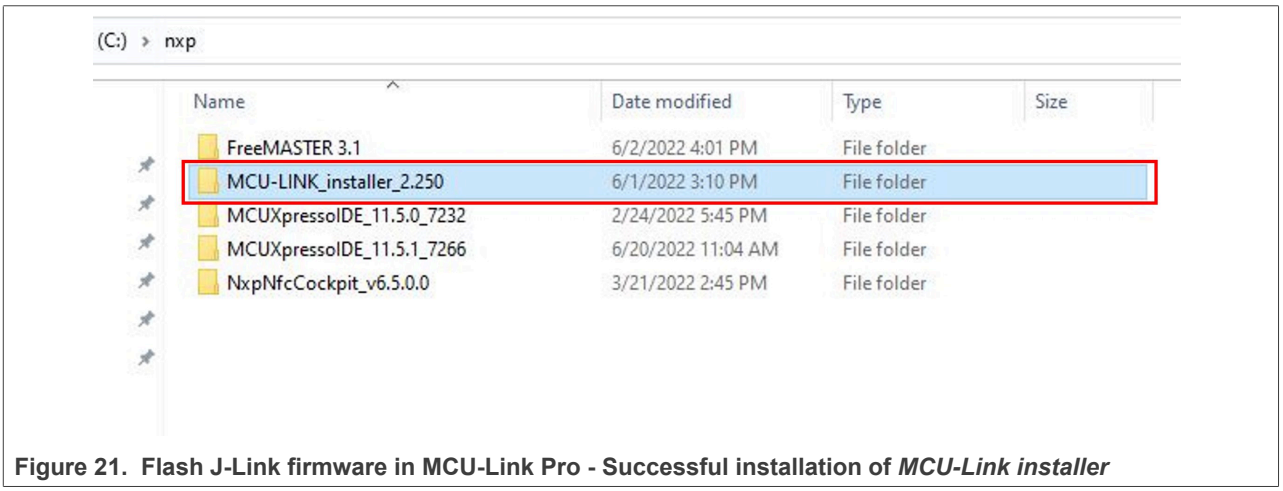


Figure 21. Flash J-Link firmware in MCU-Link Pro - Successful installation of *MCU-Link installer*

3. Follow the next steps to configure the jumpers of the MCU-Link Pro debug probe for flashing the J-Link firmware. See Figure 22:
   (1) Close the *J4* jumper of the MCU-Link Pro;
   (2) Connect the MCU-Link Pro to the PC using the micro-USB connector (*J1*).

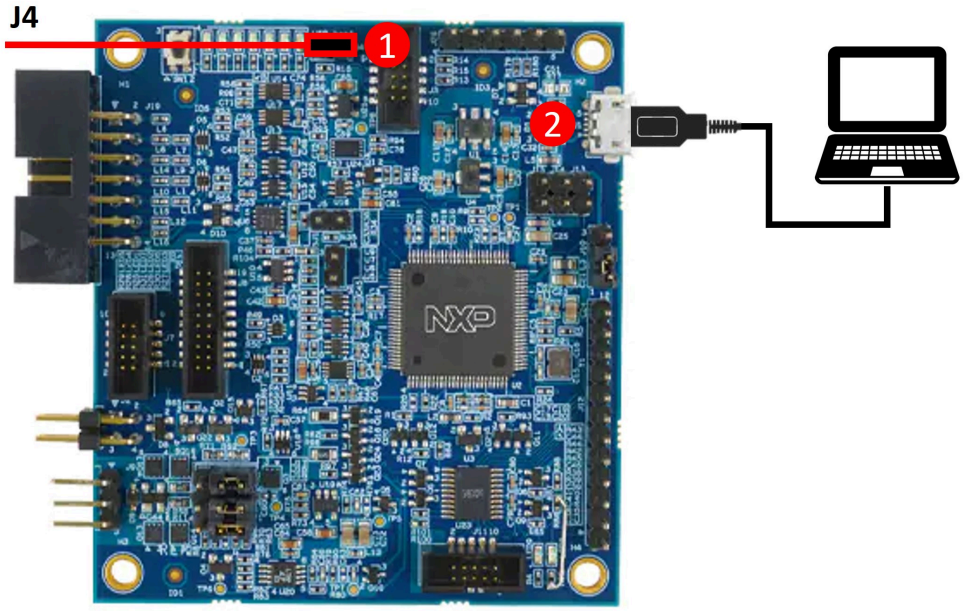


Figure 22. Flash J-Link firmware in MCU-Link Pro - Configure MCU-Link Pro debug probe

4. Flash the J-Link firmware in the MCU-Link Pro debug probe. Open a command prompt in the folder where the *MCU-Link installer* has been already installed (by default *C:\nxp\MCU-LINK_installer_2.250*), then send the command below to program the J-Link firmware:

```
.\scripts\program_JLINK
```

If the flashing is successful, you should see the message shown in Figure 23. Then, the MCU-Link PRO is ready to be used for J-Link debugging and flashing.

```
Programming "Firmware_J-Link-MCU-Link_210930.s19"

Programmed successfully - To use: remove ISP jumper and reboot.
```

Figure 23. Flash J-Link firmware in MCU-Link Pro

Note: *Once the script has been flashed, the J4 jumper must be removed from the MCU-Link Pro debug probe.*

5.4 Install USB to UART drivers

USB to UART drivers are required to detect the COM ports when the digital board is connected to the PC. Follow these instructions to download and install the USB to UART drivers in your PC:

1. Go to the following website and download the *CP210x Windows Drivers* software package as shown in Figure 24:

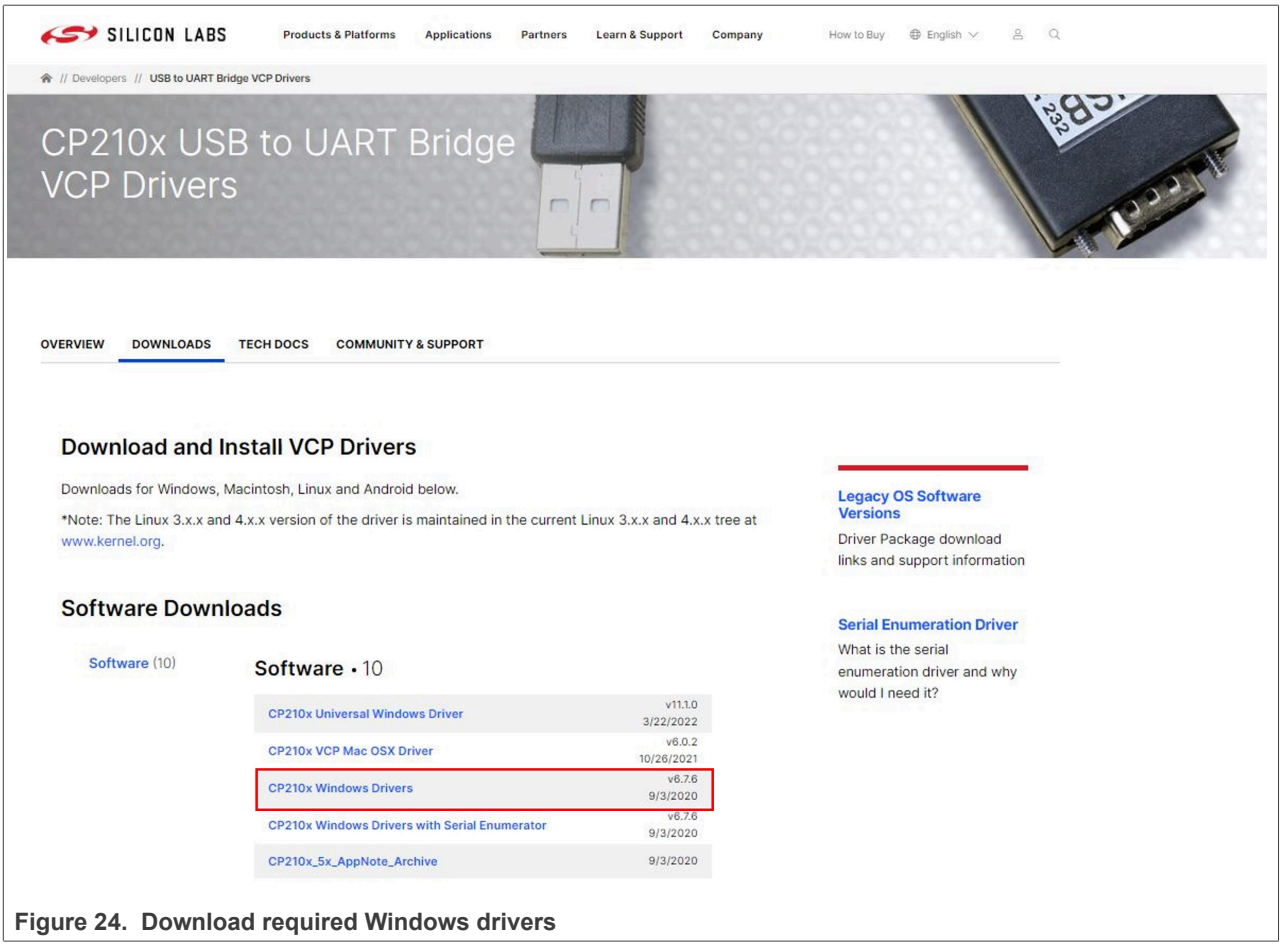


Figure 24. Download required Windows drivers

2. Run the executable and follow the steps of the installation wizard to install the drivers. Click the *finish* button to complete the installation as shown in Figure 25:

Figure 25. Complete installation of USB to UART Bridge Driver

3. Connect the digital board to the PC using the *J1* micro-USB connector, then use the Windows device manager to check that the two COM ports of the digital board are correctly detected as shown in Figure 26. The *Enhanced COM port* is used for FreeMASTER communication, while the *Standard COM port* is used for printing debug messages.

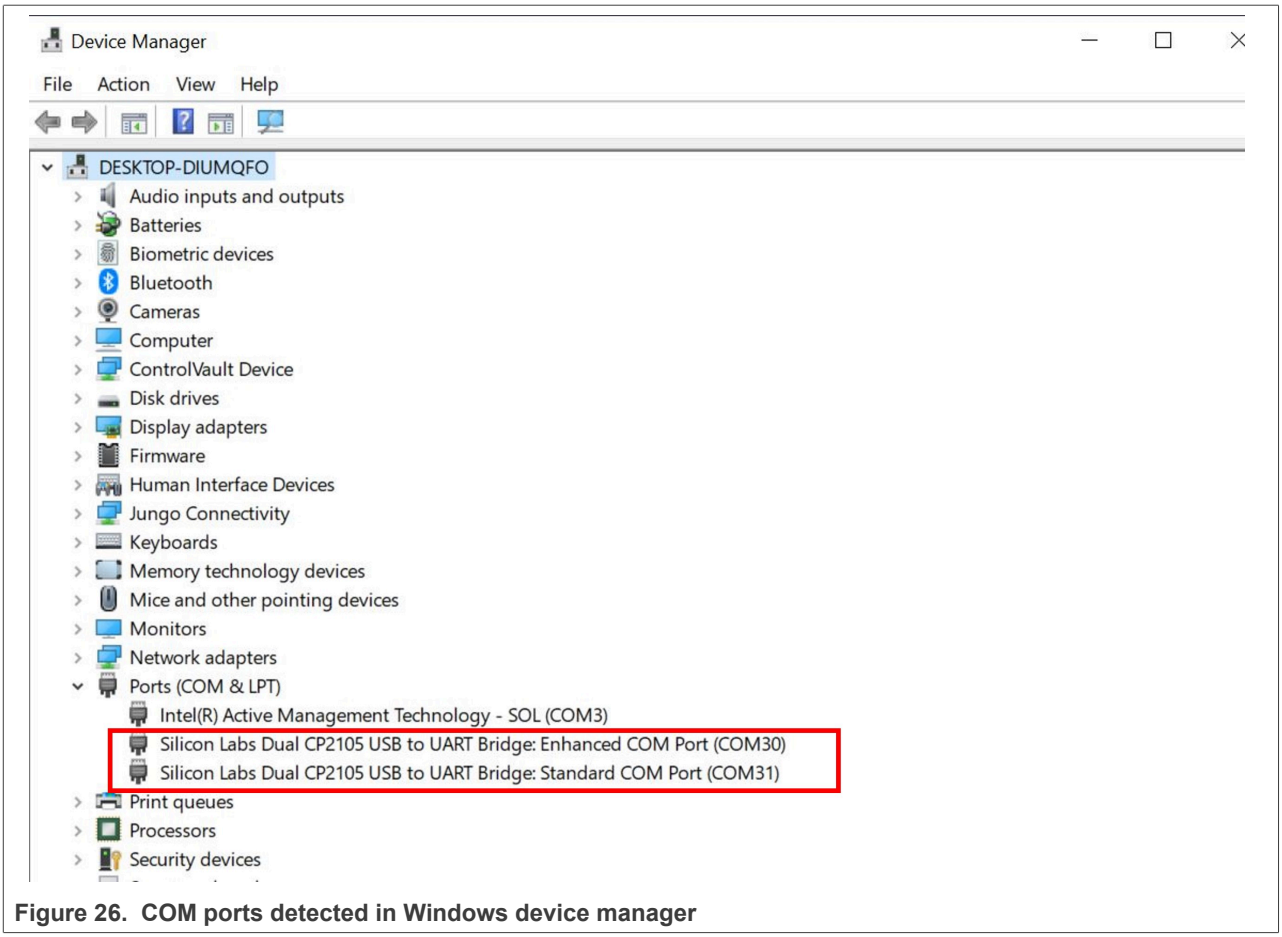


Figure 26. COM ports detected in Windows device manager

6 Software preparation

The i.MX RT Industrial Drive Development Platform software package includes a demo application that demonstrates how to monitor and control up to four PMSM motors connected to the system. Follow the instructions in this section to provision the secure element through the provisioning tool, import, and configure the i.MX RT Industrial Drive Development Platform demo application in MCUXpresso, compile it, flash it in the daughter card, and execute it. The following subsections provide details on software preparation:

- Provision the secure element
- Import the i.MX RT Industrial Drive Development Platform demo application
- Configure the i.MX RT Industrial Drive Development Platform demo application
- Compile the i.MX RT Industrial Drive Development Platform demo application
- Install the J-Link software and configure the flash loader .
- Flash the i.MX RT Industrial Drive Development Platform demo application and execute it
- Control the platform though a web interface

6.1 Provision the secure element

To provision the onboard Secure Element, use the NFC reader included in the digital board package. Provisioning takes several steps, as described below. To use the full security features provided with the application, obtain an ACS ACR1252 1S CL Reader PICC 0, open */tools/Release/Qmc2gProvisioningTool.*

ini, comment out the default CARD READER, uncomment the ACS ACR1252, and skip steps 10 to 14 of this subchapter. There is an alternative path described at the end of the subchapter.

Figure 27. Default GMMC Pocket NFC CCID 0 reader included in the HW package

1. Power off the HW to enable provisioning via NFC.
2. Insert the NFC reader into your PC and let the installation happen. After successful installation, the LED blinks red/blue/white color.
3. Go into the *tools* folder at the *root* directory and unzip the *ProvisioningTool.zip* archive into the *tools* folder. The zip password is “123”.
4. Run *ProvisioningTool.exe*. All files are placed in and taken from the working folder. Change the working folder according to your preference (copy *image_enc.exe* into your customized working folder).
5. Go to the *Create Identities* tab and click *Quick Create and Issue All Identities*. All generated content is placed into your working folder.

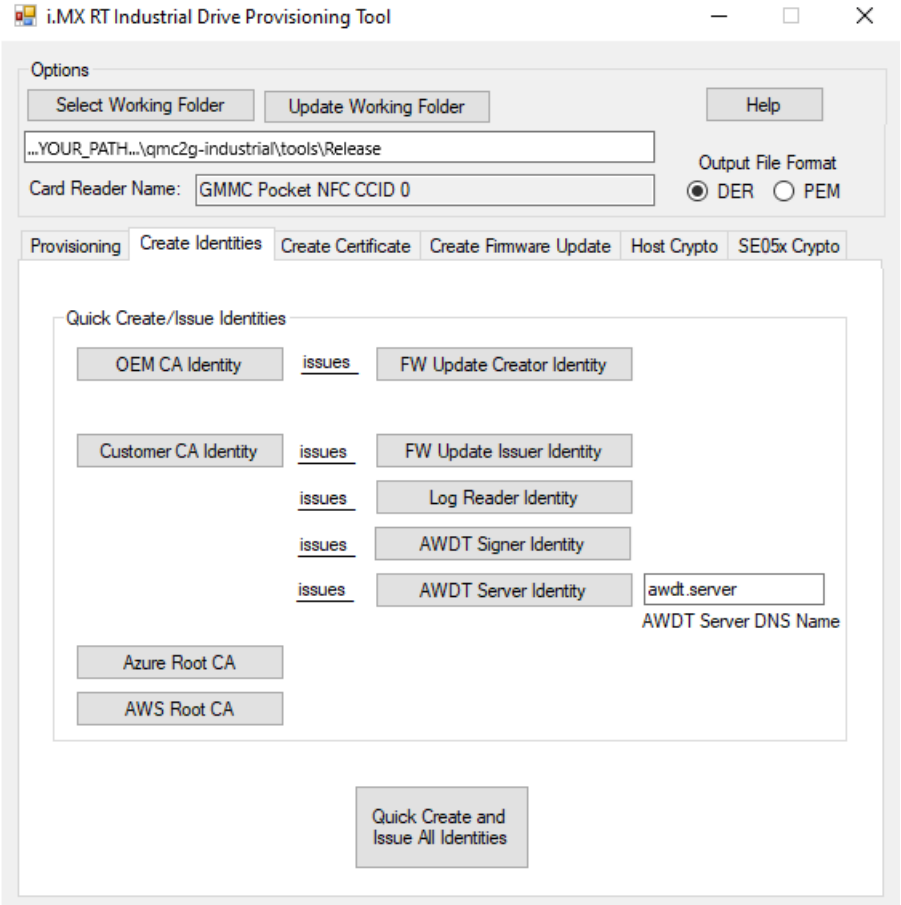


Figure 28. Create identities in the Provisioning Tool

6. Go to the *SE05x_Crypto* tab.

7. Take the unpowered Digital board and place the board by the NFC antenna below the NFC reader. The LED color changes to white once a connection is established. To ensure a strong and uninterrupted connection, make sure that the NFC reader is positioned properly on top of the digital board's NFC antenna.

Figure 29. Place the NFC reader on the Digital Board NFC antenna

8. In the provisioning tool, click the *Factory reset* button while having the NFC reader connected to the Secure Element on the Digital Board.

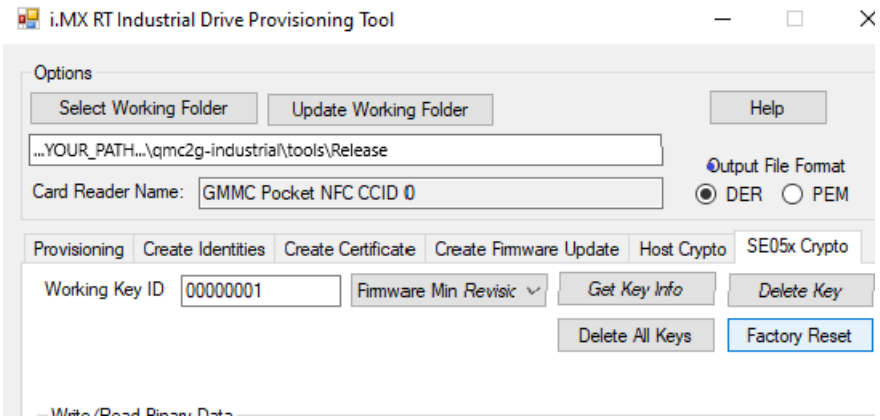


Figure 30. Perform a Factory Reset of the Secure Element

9. To provision the Secure Element with security assets, go to the *Provisioning* tab. The provisioning steps must be done as described below as policies for secure objects have not been fully defined yet.
10. Apply Policies by clicking the *Yes* radio button and select *Auth Objects*. Click the *Provision Selected* button while holding the NFC Reader connected to the Secure Element on the Digital Board.

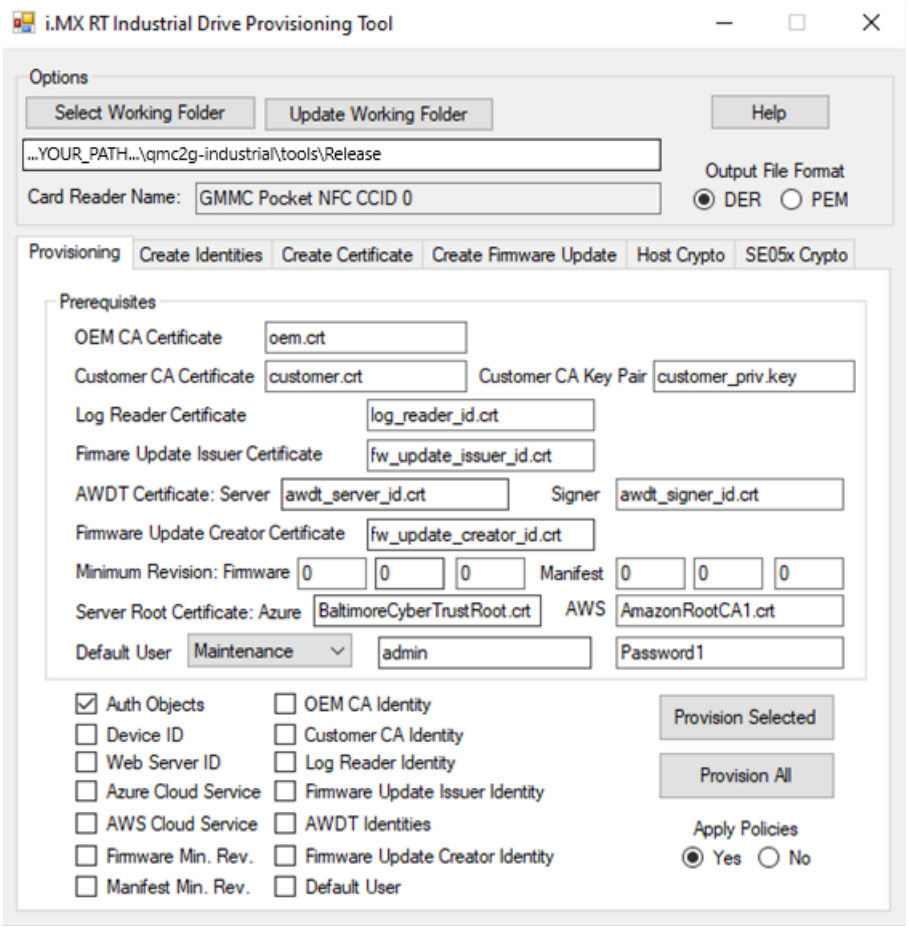


Figure 31. Provision the Secure Element with Auth Objects and Apply Policies enabled

11. Deselect *Auth Objects* and set Apply Policies to *NO*.
12. Select all remaining checkboxes.
13. Click the *Provision Selected* button again, while still holding the NFC reader connected to the Secure Element on the Digital Board. It takes a few seconds to complete the operation.

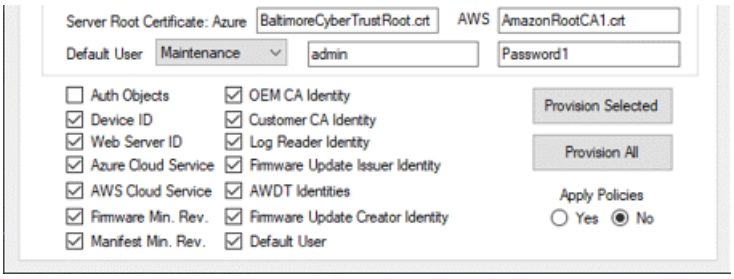


Figure 32. Provision the Secure Element with the rest of the options and Apply Policies disabled

14. The Secure Element on the Digital board is provisioned and you are ready to program your board and try out the application.

Note: *Optional steps for the ACS ACR1252 1S CL Reader PICC 0:*

10. Apply Policies by clicking the Yes radio button and click the Provision All button while holding the NFC Reader connected to the Secure Element on the Digital Board.

11. The Secure Element on the Digital board is provisioned and you are ready to program your board and try out the application.

6.2 Import the i.MX RT Industrial Drive Development Platform demo application

1. Launch the MCUXpresso IDE. When you are prompted to select the workspace, do as shown in Figure 33:
   (1) Choose an existing workspace where you will import the project or create a new workspace.
   Note: *The name of your folder may be different from the one shown in Figure 33.*
   (2) Click the **Launch** button to open the selected workspace.

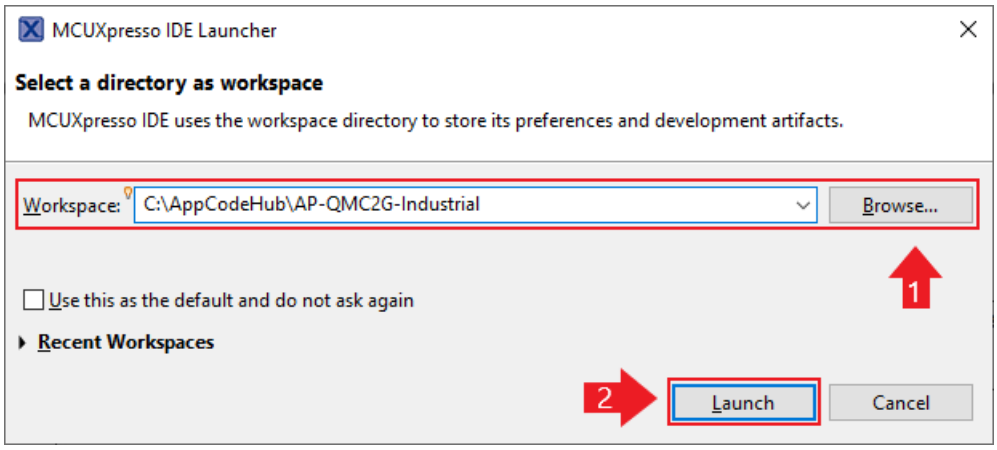


Figure 33. Select the project workspace

2. There are two ways of downloading and installing the i.MX RT Industrial Drive Development Platform software package. You can use the command line and the West utility or let MCUXpresso download and install the package directly. The package is distributed as an application software pack. West and Git must be installed and configured into your PATH variable for the next steps.

6.2.1 Option 1: Get the App Software Pack with MCUXpresso IDE

1. Open the MCUXpresso IDE and select a workspace location in an empty directory.
2. Right-click the blank area of the *Installed SDKs* panel at the bottom and select *Import remote SDK Git repository*….

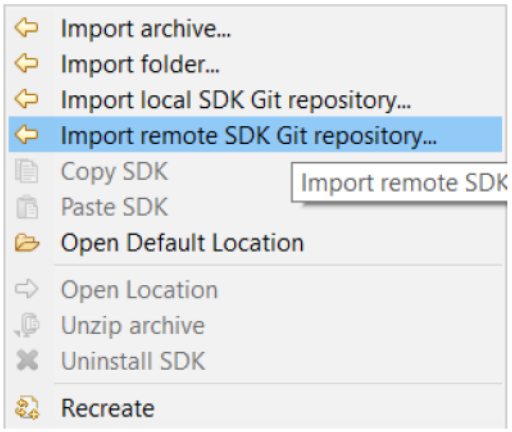


Figure 34. Import remote SDK Git repository

3. In the dialog box that comes up:
   a. In the *Location* field, click the *Browse* button and create an empty directory named "ap-qmc2g-industrial" for the application software pack to be downloaded to. Make note of this location as it will be used throughout this lab.
   b. In the *Repository* field, write "https://github.com/nxp-appcodehub/ap-qmc2gindustrial".
   c. In the *Revision* field, write "main". Then click *OK* to download the application software pack.

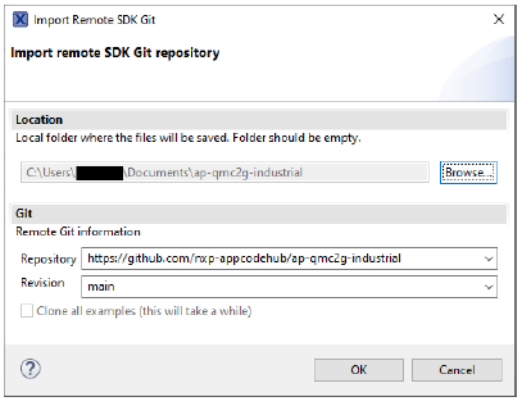


Figure 35. Import Remote SDK Git

4. Once imported, the *Installed SDKs* tab looks as shown in Figure 36:

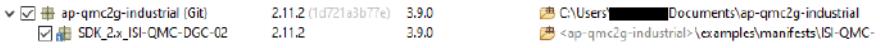


Figure 36. Installed SDKs tab

6.2.2 Option 2: Use the command line

1. Open the Windows Command Prompt and execute the following:

```
west init -m https://github.com/nxp-appcodehub/ap-qmc2g-industrial --mr main
 ap-qmc2gindustrial
cd ap-qmc2g-industrial
west update
```

2. Open the MCUXpresso IDE and select a workspace location in an empty directory.

3. Drag and drop the "ap-qmc2g-industrial directory" that was created in the previous step into the *Installed SDKs* window located in a tab at the bottom of the screen named *Installed SDKs*. Click *OK* for pop-ups shown in Figure 37and Figure 38:

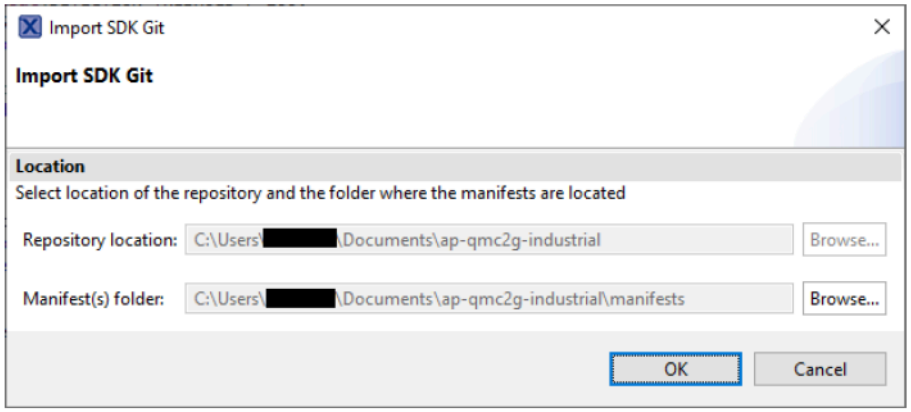


Figure 37. Import SDK Git via command line

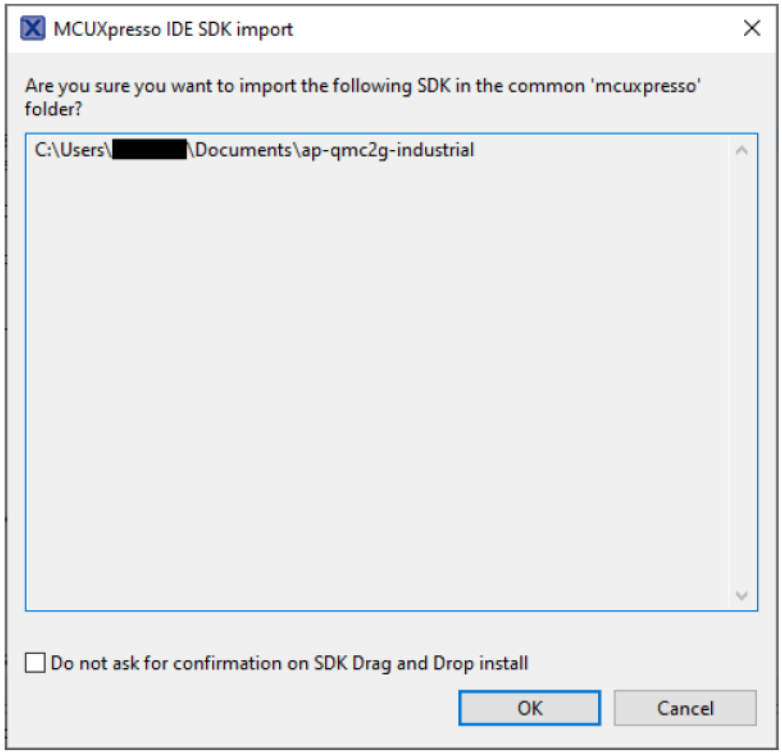


Figure 38. Import SDK to MCUXpresso IDE

4. Once imported, the *Installed SDKs* panel looks as shown in Figure 39

Figure 39. Installed SDKs tab

5. Select the i.MX RT Industrial Drive Development Platform demo projects to import, as shown in Figure 40:
   (1) Select the *Import SDK examples* option from the *Quickstart Panel*.
   (2) Choose the *isi_qmc_dgc board* icon.
   (3) Click the *next* button.

(4) Select all three given projects under *Industrial Application*:

- Industrial_app_master_cm7
- Industrial_app_slave_cm4
- Industrial_bootloader

6. Click the *Finish* button to import the projects to MCUXpresso.
   Note: *"Industrial" is the internal identifier of the i.MX RT Industrial Drive Development Platform.*

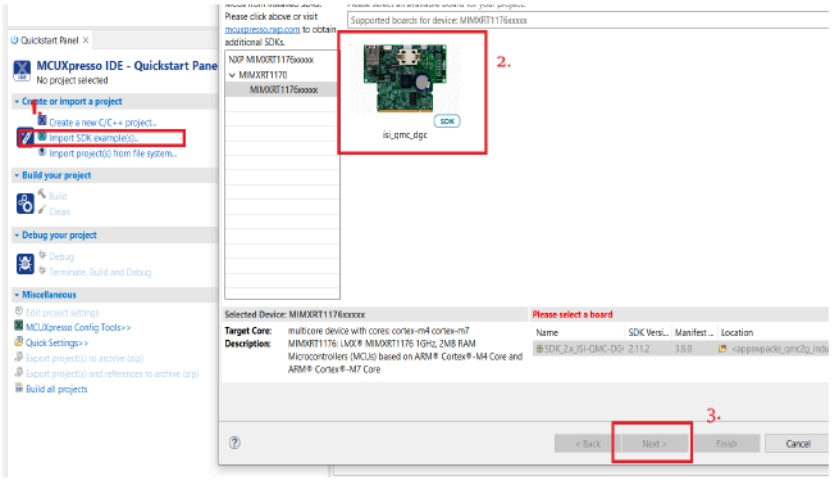


Figure 40. Import SDK example board selection

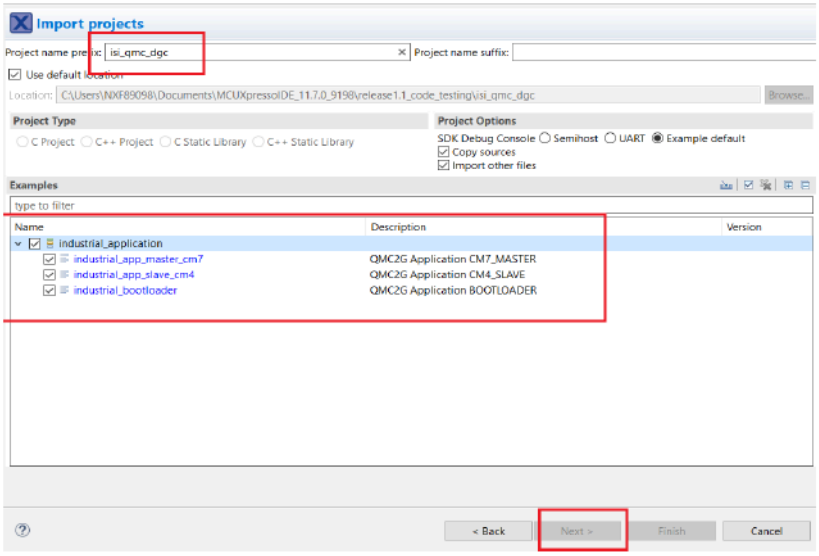


Figure 41. Import SDK Git via command line

6.3 Configure the application

The application can be configured through the *source/qmc_features_config.h* files in the CM4 and CM7 projects. Keep the configurations that share the name in sync between the projects. The most important things to configure before the first run are in the CM7 project on lines 38 and 75-78: Depending on how many Power Stage Boards are connected, the same or lower number must be configured in *MC_MAX_MOTORS*. There are four connectors on the Digital Board: Motor 1 (J153), Motor 2 (J161), Motor 3 (J1), and Motor 4 (J159). The application assumes that if only one Power Stage Board is connected, it is connected to the Motor 1 (J153)

connector on the Digital Board. Likewise, if two are connected, it is assumed that they are connected to Motor 1 (J153) and Motor 2 (J161), and so on.

```
#define MC_MAX_MOTORS        (4)          /* Number of supported motors */
```

Depending on the type of Power Stage Board, it either does or does not have the NAFE chip soldered on it. These four macros must be configured accordingly. Otherwise, the application does not run correctly. If fewer than 4 power stage boards are used, set the unused ones to 0 as well.

```
#define MC_HAS_AFE_MOTOR1 (1) /* Defines if PSB1 has the AFE soldered on. 1
 means the AFE is soldered, 0 means the AFE is missing. */
#define MC_HAS_AFE_MOTOR2 (1) /* Defines if PSB2 has the AFE soldered on. 1
 means the AFE is soldered, 0 means the AFE is missing. */
#define MC_HAS_AFE_MOTOR3 (1) /* Defines if PSB3 has the AFE soldered on. 1
 means the AFE is soldered, 0 means the AFE is missing. */
#define MC_HAS_AFE_MOTOR4 (1) /* Defines if PSB4 has the AFE soldered on. 1
 means the AFE is soldered, 0 means the AFE is missing. */
```

The other configurations can control TSN and Secure Element debug information, whether the FreeMASTER connection should be supported by the application, various thresholds, delays, sizes and other configuration values, the GUI color scheme, pin usage, and more.

6.4 Compile the i.MX RT Industrial Drive Development Platform demo application

There are several build targets for all three of the projects. The CM7 and CM4 projects have the Debug, Release, Debug_SBL, and Release_SBL targets.

The Debug and Release targets require a debug probe to be connected to the board. When the application is built this way, it must be initialized by the debug probe and started manually through the IDE. However, it does not need the Bootloader project to be compiled and is simpler to get started with for testing and debugging purposes.

Note: *The Release target has some debug symbols disabled and the debugging experience will be affected.*

The Debug_SBL and Release_SBL targets require the Bootloader project to be compiled as well. To run the application when it is built with one of these targets, refer to the Provisioning and Secure Bootloader User Guide document in the *tools/ folder.* It is more complicated to set up but the application is able to boot on its own.

The Bootloader project has the Debug_Non_Secure, Debug, and Release targets. The Debug_Non_Secure is the simplest version, which does not enable any of the security features and must only be used for testing and debugging purposes. The Debug target enables most of the security features but not FW encryption and does not need all security-related fuses burned. The Release target is the production target that requires some critical security fuses to be burned. It is irrevocable and can negatively influence debugging and testing of the application.

1. Build the *isi_qmc_dgc_industrial_app_slave_cm4* project, as shown in Figure 42:
   (1) Select the *isi_qmc_dgc_industrial_app_slave_cm* project from the left pane.
   (2) Click the arrow next to the hammer icon in the top bar. Select the *Debug (Debug build)* option. Wait a few seconds to finish the compilation.

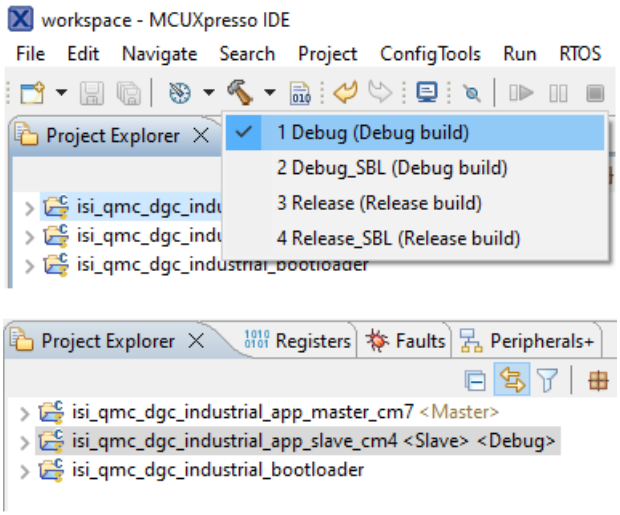


Figure 42. Build the isi_qmc_dgc_industrial_app_slave_cm4 project

2. Build the *isi_qmc_dgc_industrial_app_master_cm7* project, as shown in Figure 43:
   (1) Select the *isi_qmc_dgc_industrial_app_master_cm7* project from the left pane.
   (2) Click the arrow next to the hammer icon in the top bar. Select the *Debug (Debug build)* option. Wait a few seconds for the project to compile.
   Note: *it is important to compile the isi_qmc_dgc_industrial_app_slave_cm4 project before compiling the isi_qmc_dgc_industrial_app_master_cm7 project.*

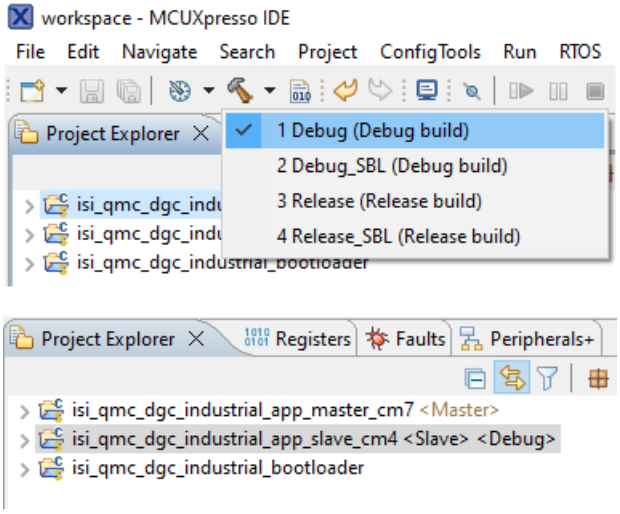


Figure 43. Build isi_qmc_dgc_industrial_app_master_cm7 project

6.5 Install the J-Link software and configure the flashloader

Before you can flash the i.MX RT Industrial Drive Development Platform demo application, install the J-Link software and configure the J-Link flashloader. To do it, follow these steps:

Note: *Before following the instructions in this section, make sure you have installed MCUXpresso on your PC. Installing MCUXpresso after completing the instructions in this section might override some of the configurations applied.*

1. Go to the SEGGER J-Link website and download the J-Link installer for your operating system as shown in Figure 44. Run the executable and follow the installation wizard until the installation is completed

successfully. Use the following link for J-Link installations: https://wiki.segger.com/UM08001_J-Link_/_J-Trace_User_Guide

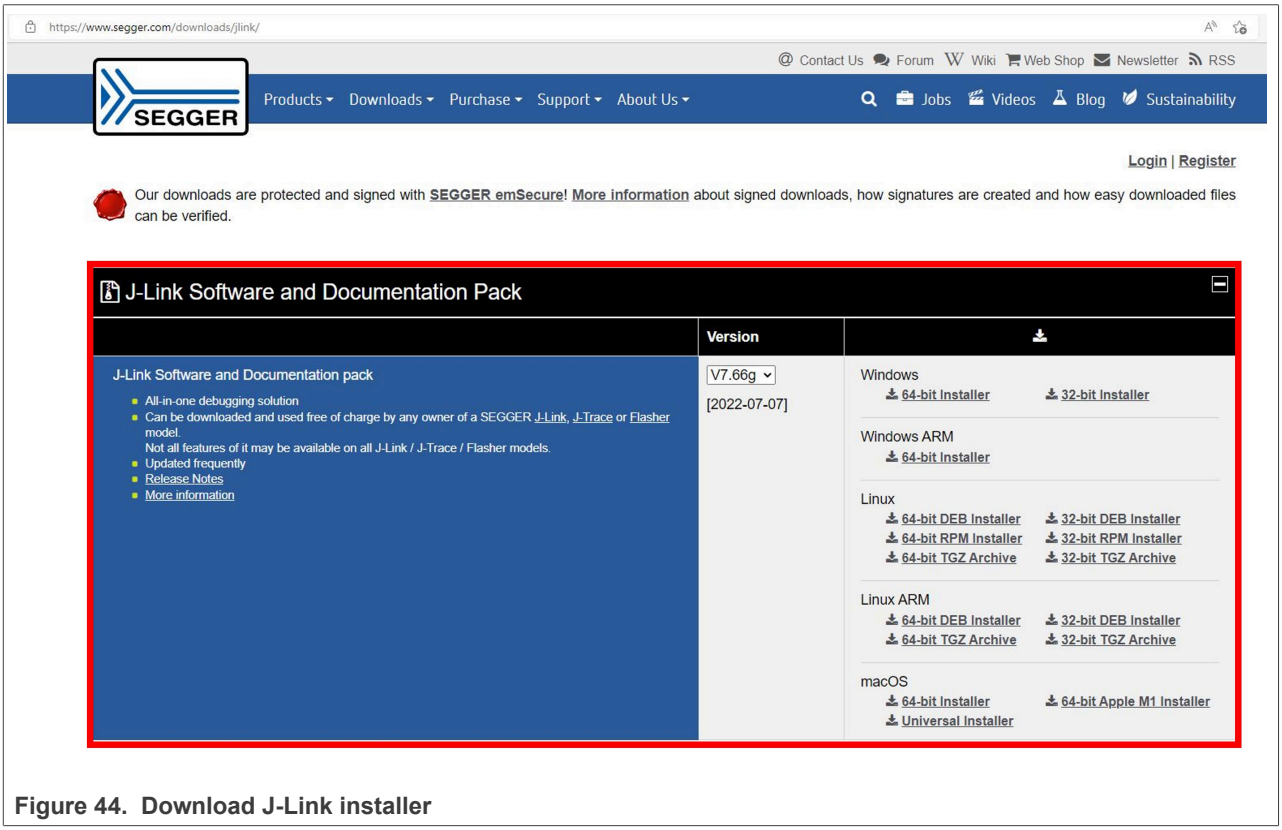


Figure 44. Download J-Link installer

2. During the installation process, a J-Link configurations folder is created in "%APPDATA%/SEGGER/". Copy the JLinkDevices folder from the "tools" folder of the application software pack into the "%APPDATA%/SEGGER/" folder, as shown in Figure 45.

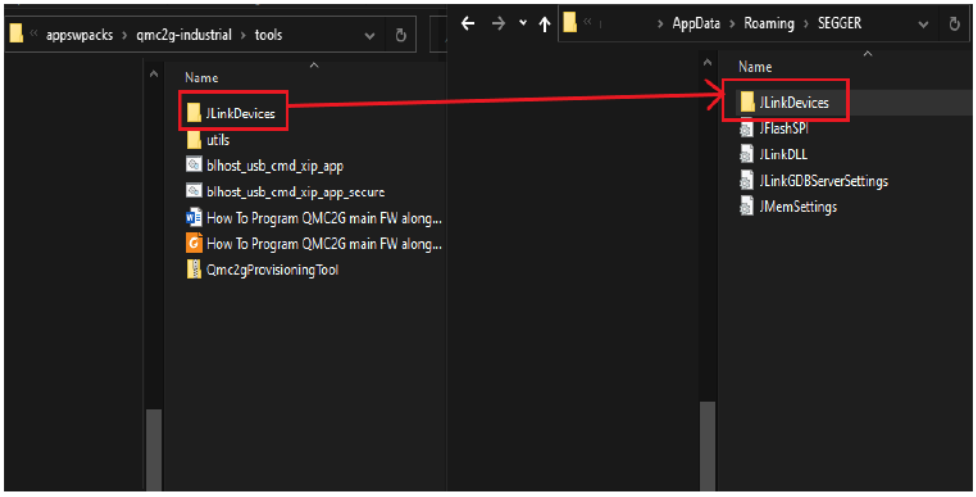


Figure 45. Update the SEGGER flashloader configuration

Note: *For a different OS, the J-Link device file must be updated in different locations. Use the following locations:*

Table 6. Device file locations

OS	Location
Windows	*C:\Users\<USER>\AppData\Roaming\SEGGER\JLink Devices*
Linux	*$HOME/.config/SEGGER/JLinkDevices*
macOS	*$HOME/Library/Application Support/SEGGER/JLinkDevices*

6.6 Flash the i.MX RT Industrial Drive Development Platform demo application and execute it

Follow the instructions provided below to flash the i.MX RT Industrial Drive Development Platform demo application in the daughter card and execute it:

Note: *In this section, the MCU-Link Pro debug probe is used to flash the software. However, Segger J-Link debug probes can also be used in a similar way to flash the demo application in the daughter card.*

1. Connect the *MCU-Link Pro* debug probe to the system as shown in Figure 46:
   (1) First, connect the MCU-Link Pro to the PC using a micro-USB cable. Make sure that the *J4* jumper (firmware update jumper) of the MCU-Link Pro is not placed;
   (2) Use a *10-pin to 10-pin Cortex SWD* cable to connect the MCU-Link Pro SWD connector (*J7*) to the daughter card SWD connector (*J1*). Make sure that the SWD cable is inserted in the daughter card SWD connector with the same orientation shown in the picture.
   Note: *The red cable of the SWD cable must be connected to pin number 2 of the SWD connector in the daughter card.*
   (3) Make sure that the daughter card is correctly supplied, so that the status LEDs are turned on. If not, make sure that the system is correctly supplied and that the *SW1* switch of the digital board is in the ON position.

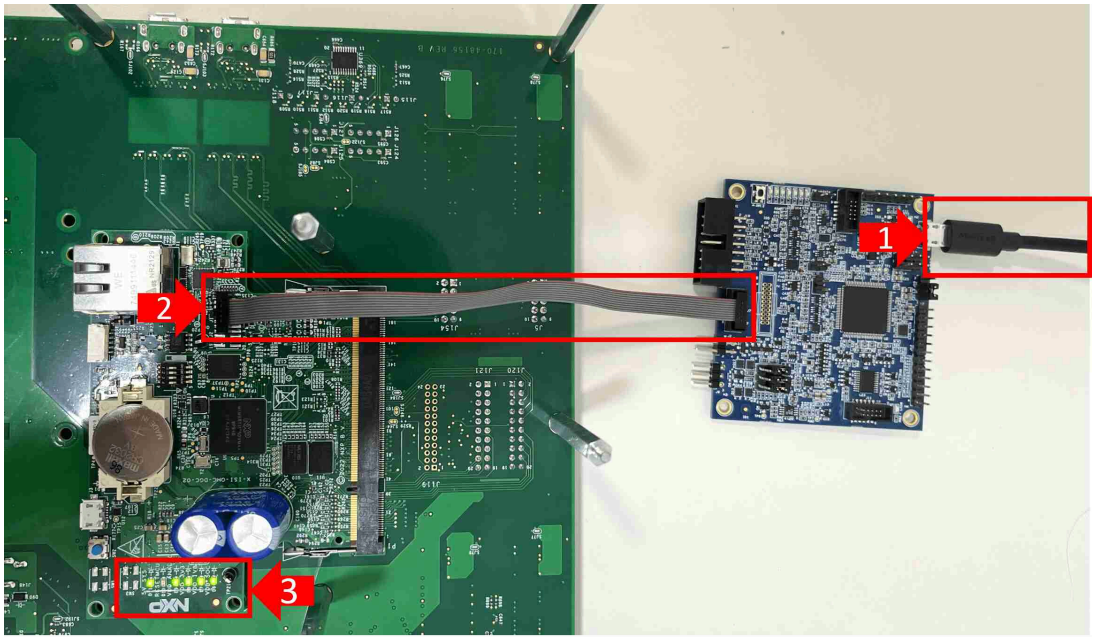


Figure 46. Connect the MCU-Link Pro to the daughter card

2. If you have not done it already, connect the digital board to the PC using the *J48* connector as shown in Figure 47. If you open the Windows device manager, you should see two *Silicon Labs Dual CP2105 USB*

to UART bridge ports: the *enhanced COM port* will be used for the FreeMASTER communication, while the *standard COM port* will be used to print debug logs in the terminal. Take note of the two COM port numbers.
Note: *If you cannot see the two UART ports, make sure you correctly installed the USB to UART drivers as described in Section 5.4.*

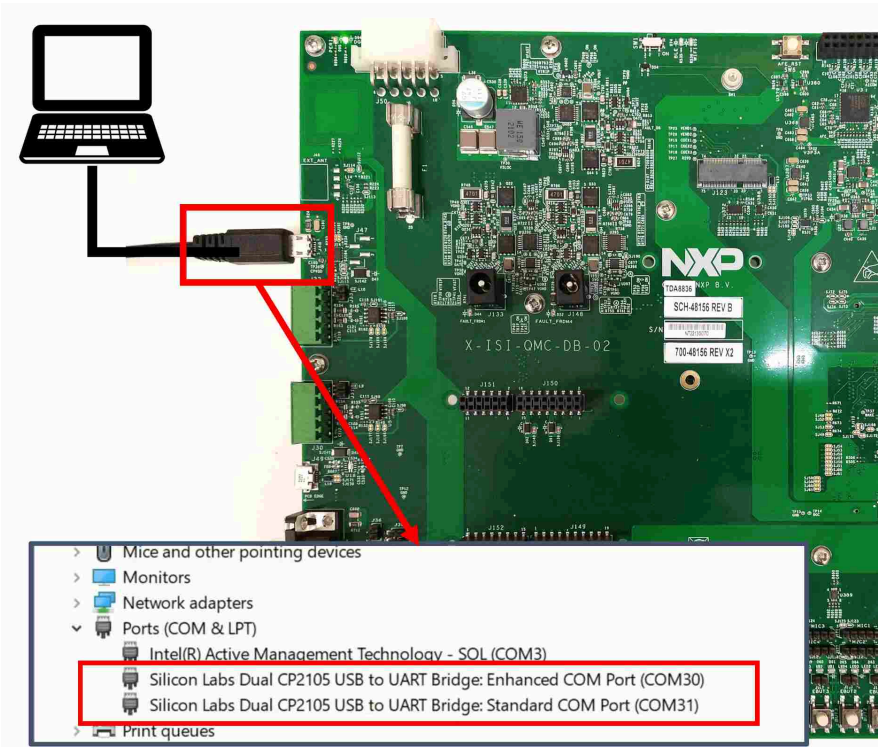


Figure 47. Connect the digital board to the PC

3. In MCUXpresso, open a serial terminal window as shown in Figure 48. The serial terminal window will be used to see debug logs when the software is running.
   (1) Click the *terminal* tab on the bottom pane of MCUXpresso IDE;
   (2) Click the *open a new terminal* icon;
   (3) Select *serial terminal*;
   (4) Select the serial communication port number from the dropdown menu. This corresponds to the *standard COM port* retrieved in the previous step;
   (5) Set the baud rate to 115200;
   (6) Leave the default value for the other settings (*data size: 8, parity: none, stop bits: 1, encoding: default*);
   (7) To open the serial terminal window, click the *OK* button.
   Note: *Alternatively, use your preferred terminal application to print debug logs. Make sure to use the same settings as in Figure 48.*

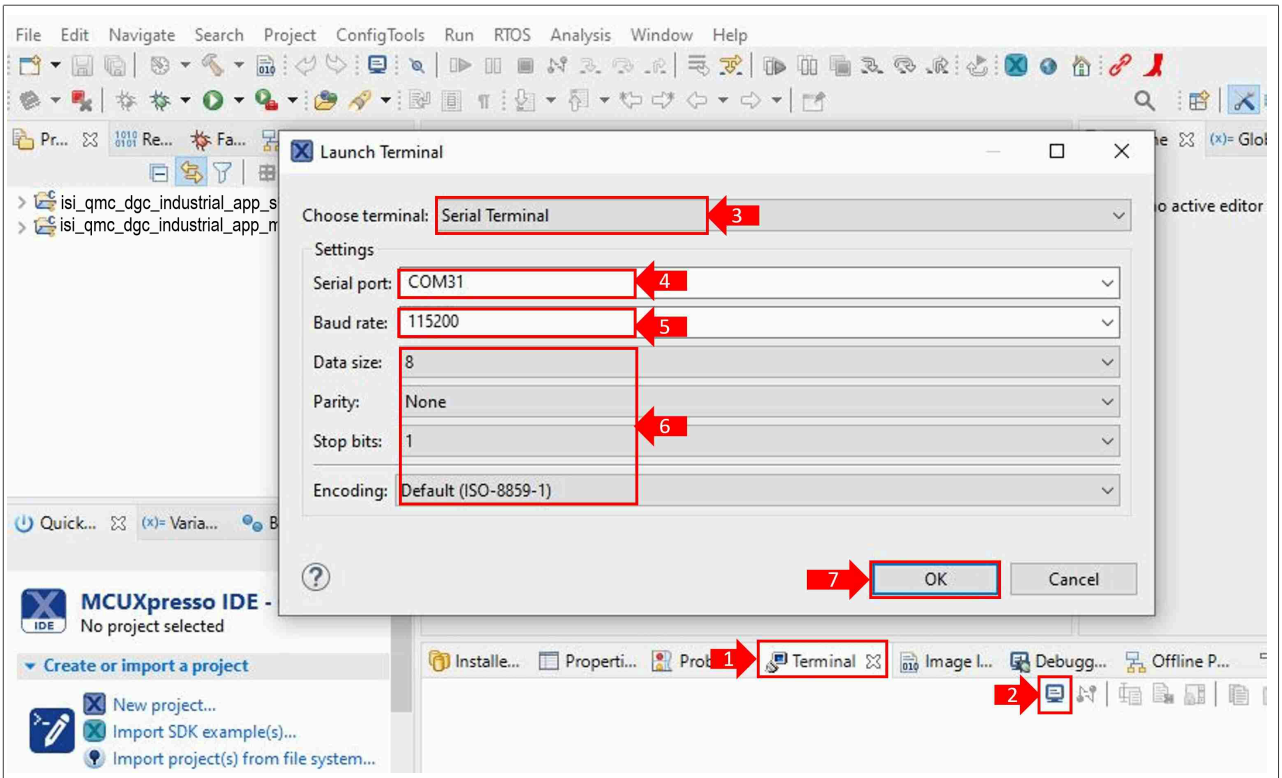


Figure 48. Open a serial terminal window in MCUXpresso

4. Start debugging the *isi_qmc_dgc_industrial_app_master_cm7* project as shown in Figure 49:
   (1) Select the *isi_qmc_dgc_industrial_app_master_cm7* project;
   (2) Click the arrow next to the green *debug* icon;
   (3) Select *debug configurations*.

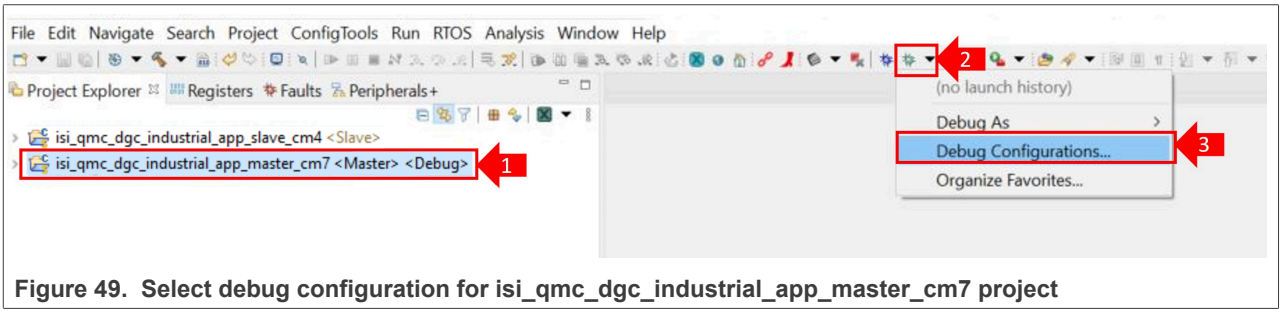


Figure 49. Select debug configuration for isi_qmc_dgc_industrial_app_master_cm7 project

5. Follow the steps shown in Figure 50 to start the debugging process:
   (1) Select the *isi_qmc_dgc_industrial_app_master_cm7* Debug configuration under *GDB SEGGER Interface debugging*;
   (2) Open the *J-Link debugger* tab;
   (3) Check that the *J-Link interface settings* are set as in Figure 50. In particular, the *device* field should have the value *MIMXRT1170_FLEXSPI2_UFL*;
   (4) Make sure to uncheck the *Reset before running* option.
   Note: *If the device field is not set correctly, check that you correctly applied the Segger flashloader configuration as described in Section 6.5.*
   (5) Click the *debug* button to start debugging the software. Wait a few seconds for MCUXpresso to load the software image in the daughter card.

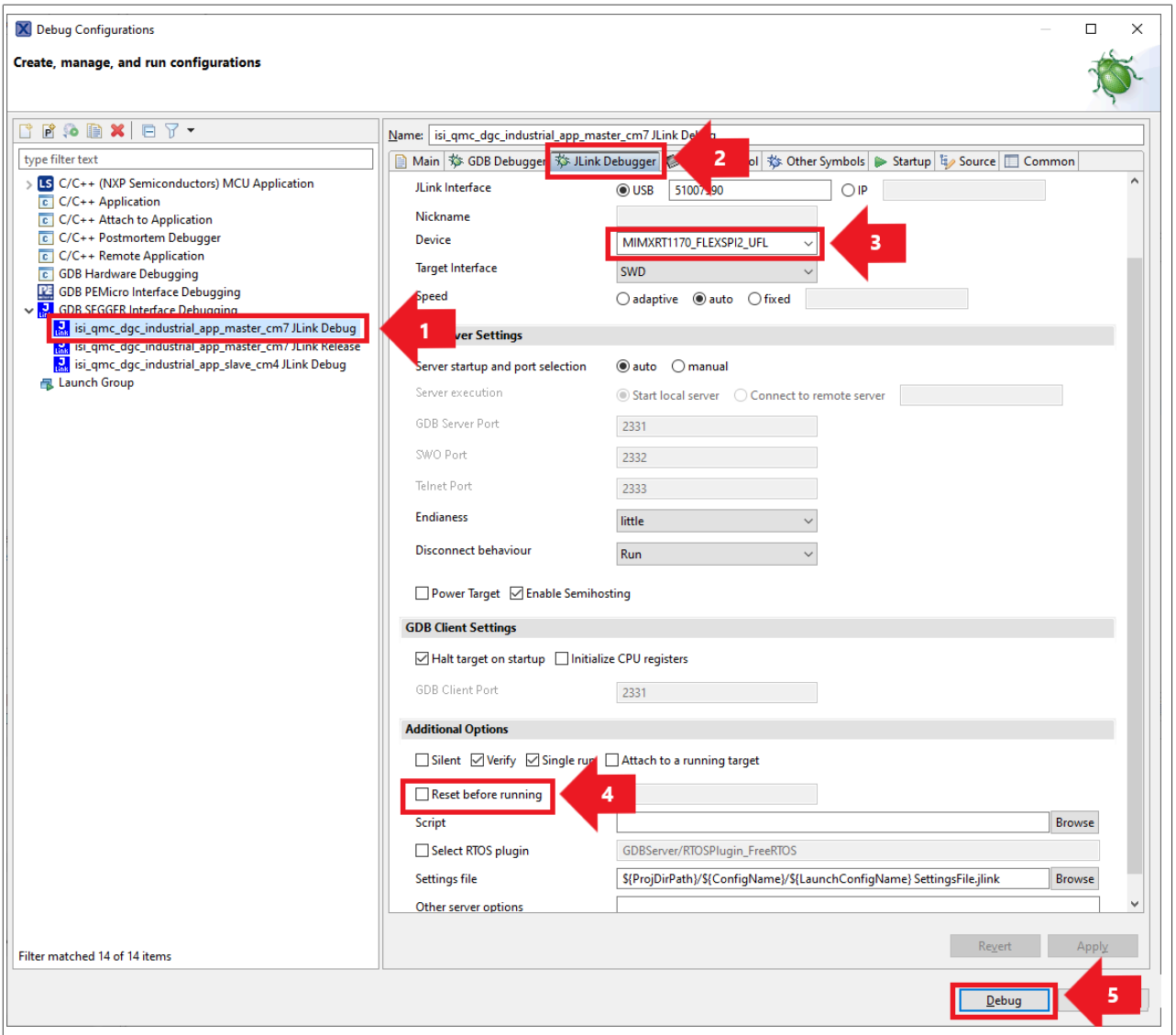


Figure 50. Debug configurations for isi_qmc_dgc_industrial_app_master_cm7

6. The software starts running on the daughter card and will stop at a preconfigured breakpoint. Set a second breakpoint as shown in Figure 51:
   (1) Double-click on the *main()* function running in *thread #1*;
   (2) Set a breakpoint in line 184 (*CLOCK_GetFreqFromObs(CCM_OBS_ADC1_CLK_ROOT);*), just after the *BOARD_InitBootPeripheral()* function. To do this, double-click the blue line next to the line number where you want to set the breakpoint. A small blue dot must appear indicating that the breakpoint has been correctly set;
   (3) Click the *resume* button to run the software until it reaches the breakpoint in line 184.

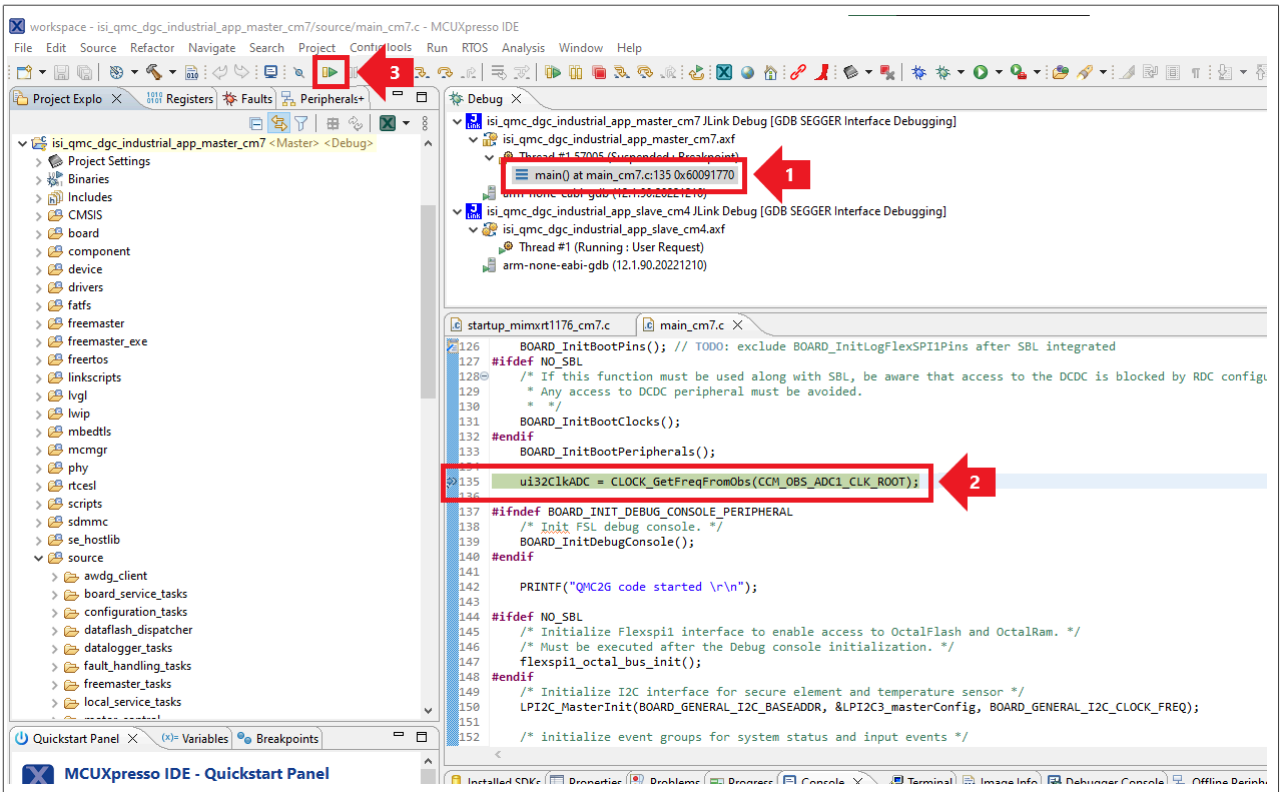


Figure 51. Execute Software: set breakpoint in main() ofisi_qmc_dgc_industrial_app_master_cm7

7. Start a debugging session for *isi_qmc_dgc_industrial_app_slave_cm4* following the same steps described for *isi_qmc_dgc_industrial_app_master_cm7* (Step 4). Select the appropriate debugging configuration as shown in Figure 52:
   (1) Select the *isi_qmc_dgc_industrial_app_slave_cm4 J-Link Debug* configuration under *GDB SEGGER Interface debugging*;
   (2) Open the *J-Link debugger* tab;
   (3) Check that the *J-Link interface settings* are set as in Figure 52. In particular, the *device* field should have the value *MIMXRT1176xxxA_M4*; Uncheck the "Attach to a Running Target" option.
   (4) Click the *Debug* button to start debugging the software. Wait a few seconds for MCUXpresso to load the software image in the daughter card.

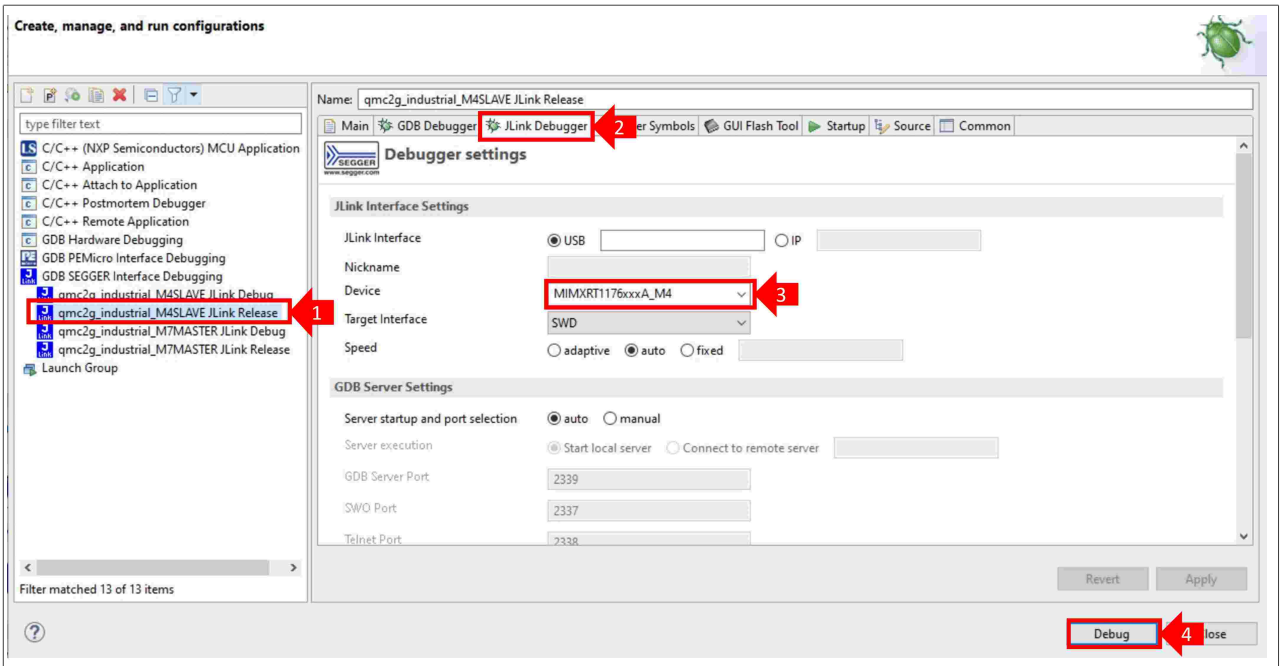


Figure 52. Start debugging session for isi_qmc_dgc_industrial_app_slave_cm4

8. Once the software is loaded (1), double-click the *main()* function of the *isi_qmc_dgc_industrial_app_slave_cm4*, (2) then click the *resume* button as shown in Figure 53:

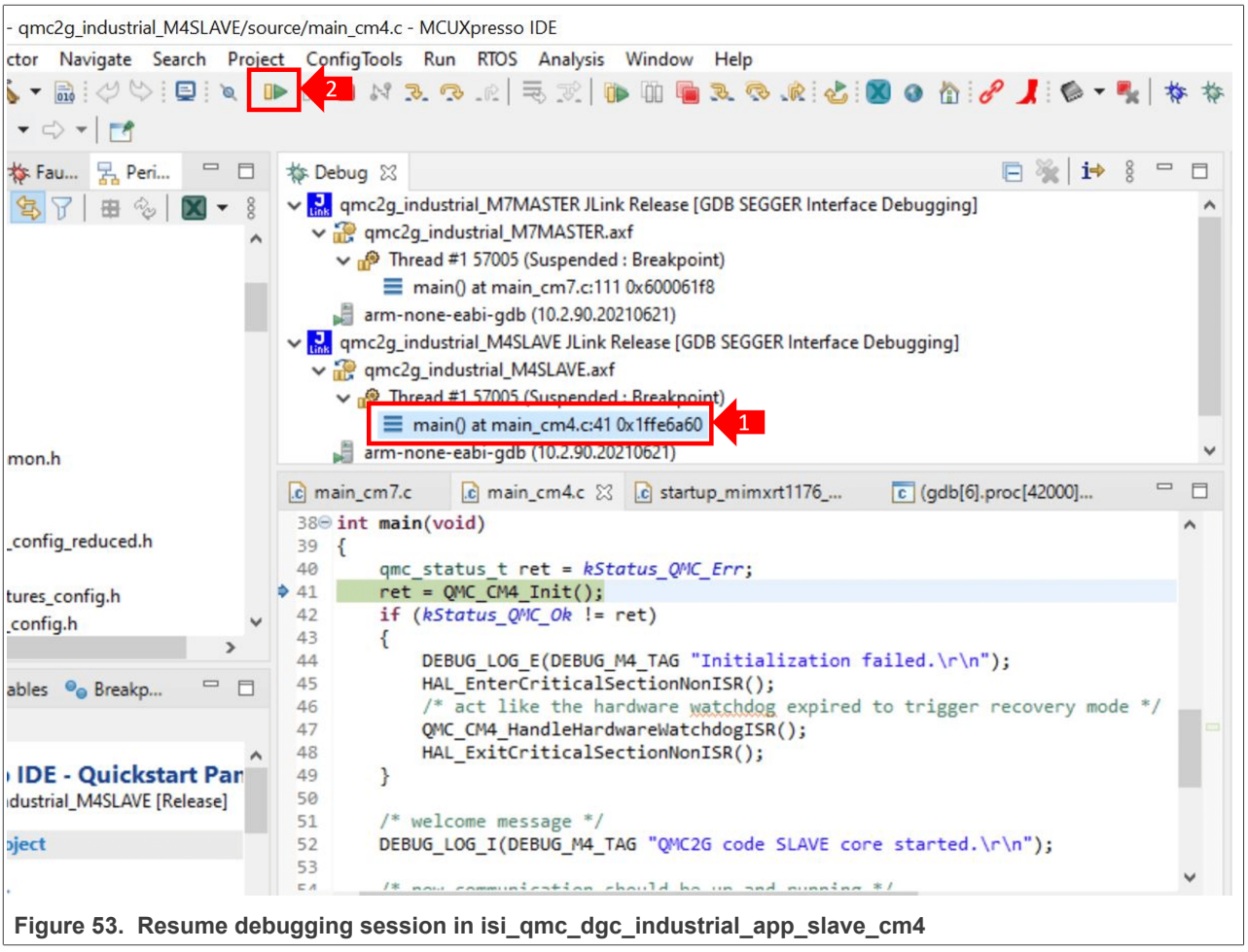


Figure 53. Resume debugging session in isi_qmc_dgc_industrial_app_slave_cm4

9. Go back to the *main*() function of *isi_qmc_dgc_industrial_app_master _cm7* and click the *resume* button to resume the execution of the software after the breakpoint. If the software is running correctly on the daughter card, you should see the message shown in Figure 54 in the terminal window that was opened in Step 3. You can now control the motors using either FreeMASTER or a TSN master device.
Note: *Other logs might be shown right after the QMC2G code started message.*

Figure 54. i.MX RT Industrial Drive Development Platform software is correctly running

6.7 Control the platform through a web interface

The application can be controlled through a web API used to communicate with a web server running on the i.MX RT Industrial Drive Platform. To set this up, a host machine is required with an Ethernet connection to the

platform. The host machine can be either a separate PC running a Linux or Windows OS or it can be the same PC used to program the boards. If you use the same PC, set up a dual network to allow the server connection to work correctly. A simple way is to use a static IP for your PC as described below:

1. Make sure that the application is running on the platform.
2. Connect an Ethernet cable between the daughter card and your host machine.
3. The web server IP is set as 10.42.0.10 by default. Set up the host machine IP within the same subnet IP range. Navigate to the IPV4 Properties and configure them as in the figure below:

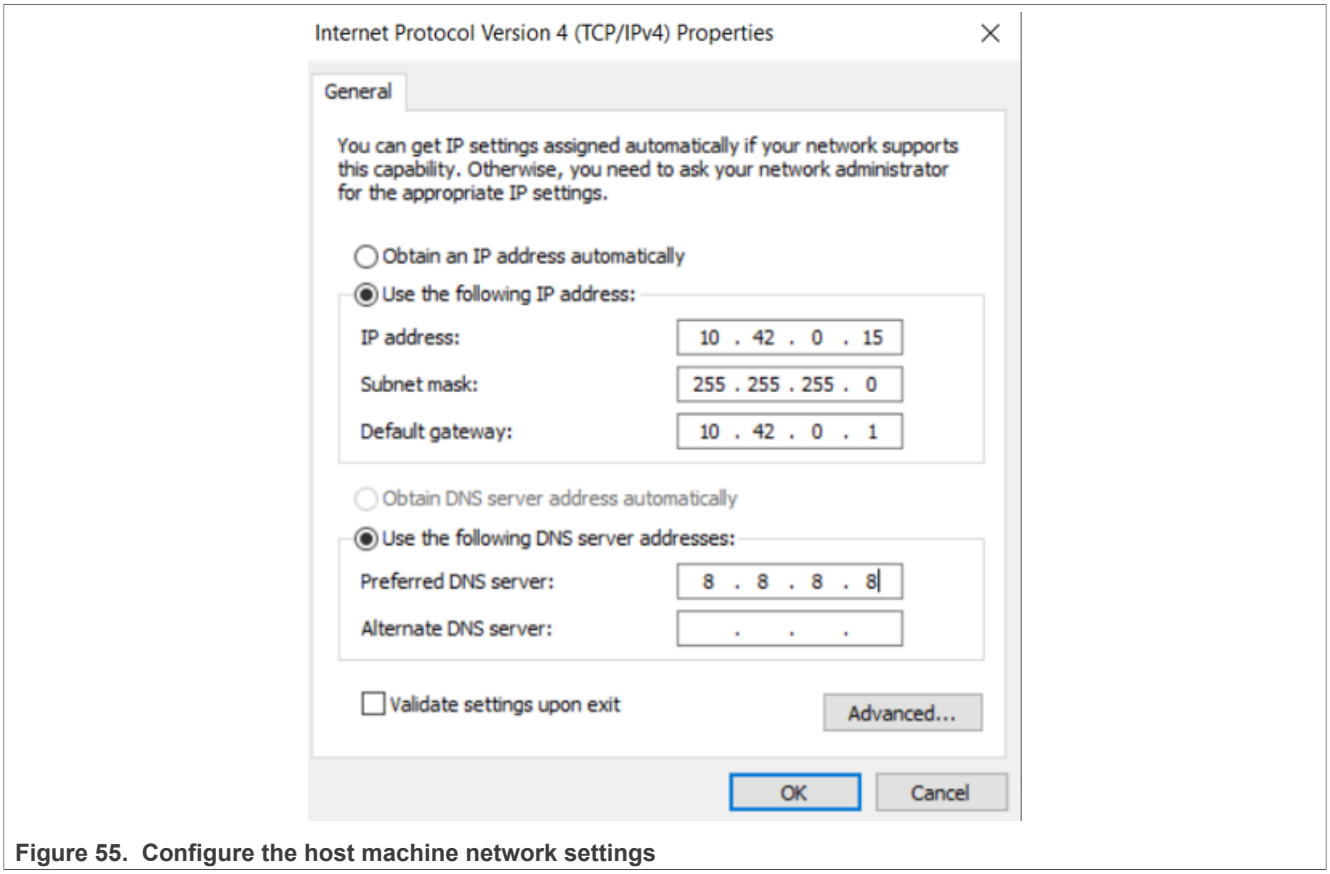


Figure 55. Configure the host machine network settings

4. Optional: Test the connection with a ping to confirm it is ok.
5. Open *10.42.0.10* in your web browser. The following webpage should appear:

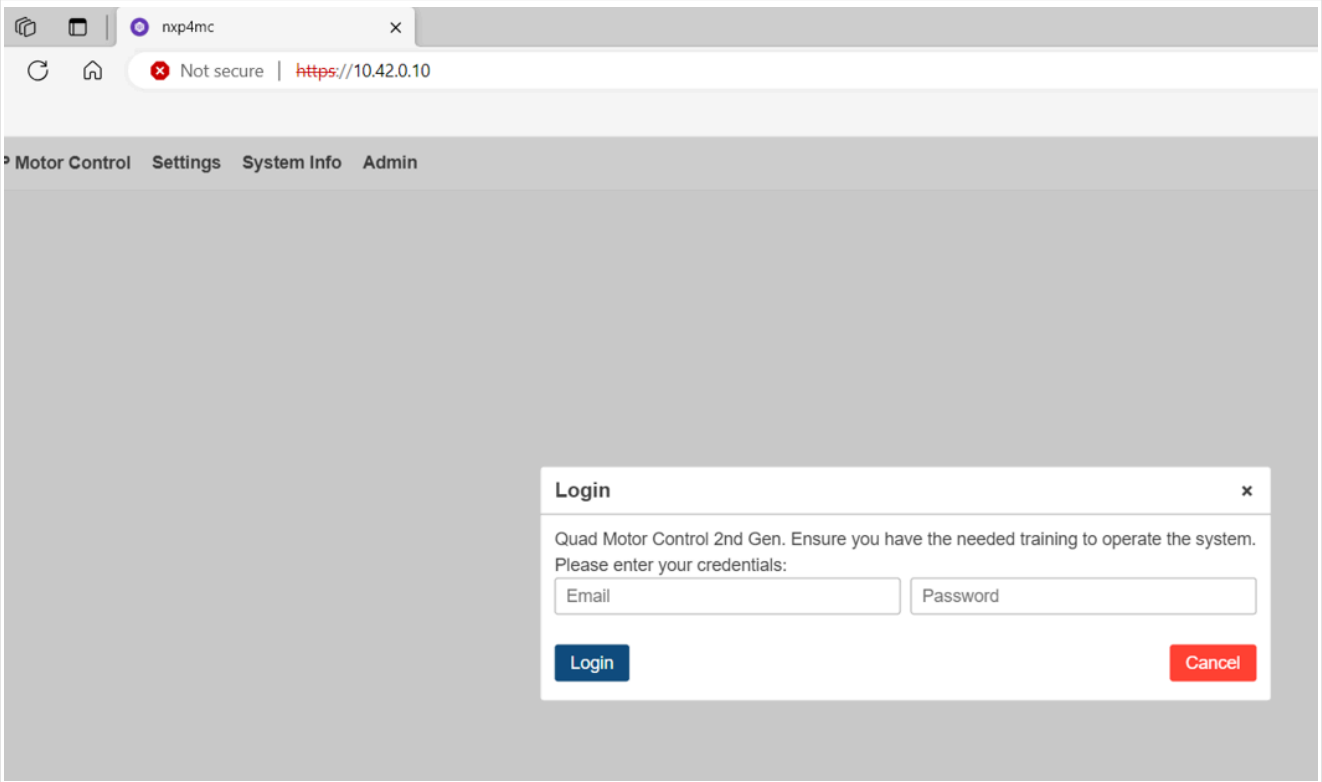


Figure 56. Login screen

6. The login credentials were configured during the provisioning steps. The default login for an administrator is *admin – Password1*.

Figure 57. Login credentials in the Provisioning Tool

7. Once logged in, the motor control dashboard must be visible. In this dashboard, configure and send motor commands and control the motors similarly to how it is done in FreeMASTER.

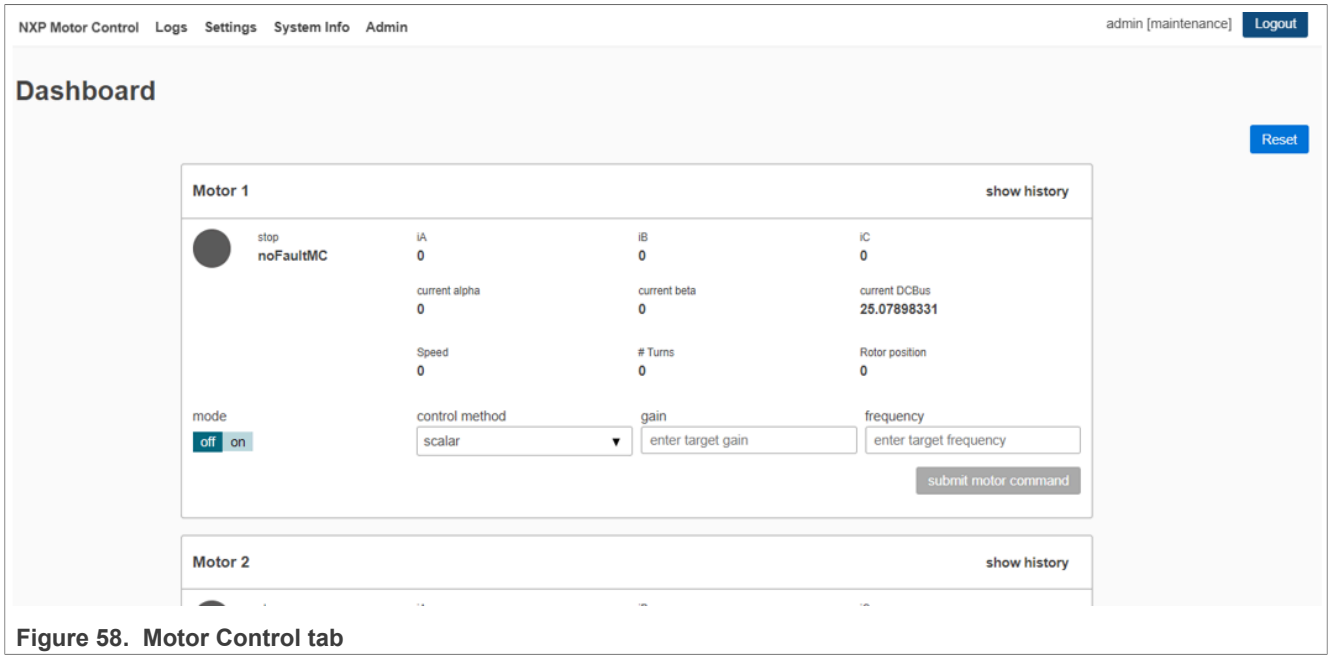


Figure 58. Motor Control tab

8. The Logs tab shows the latest logs reported by the application.
9. The Settings tab can be used to reconfigure various device settings:

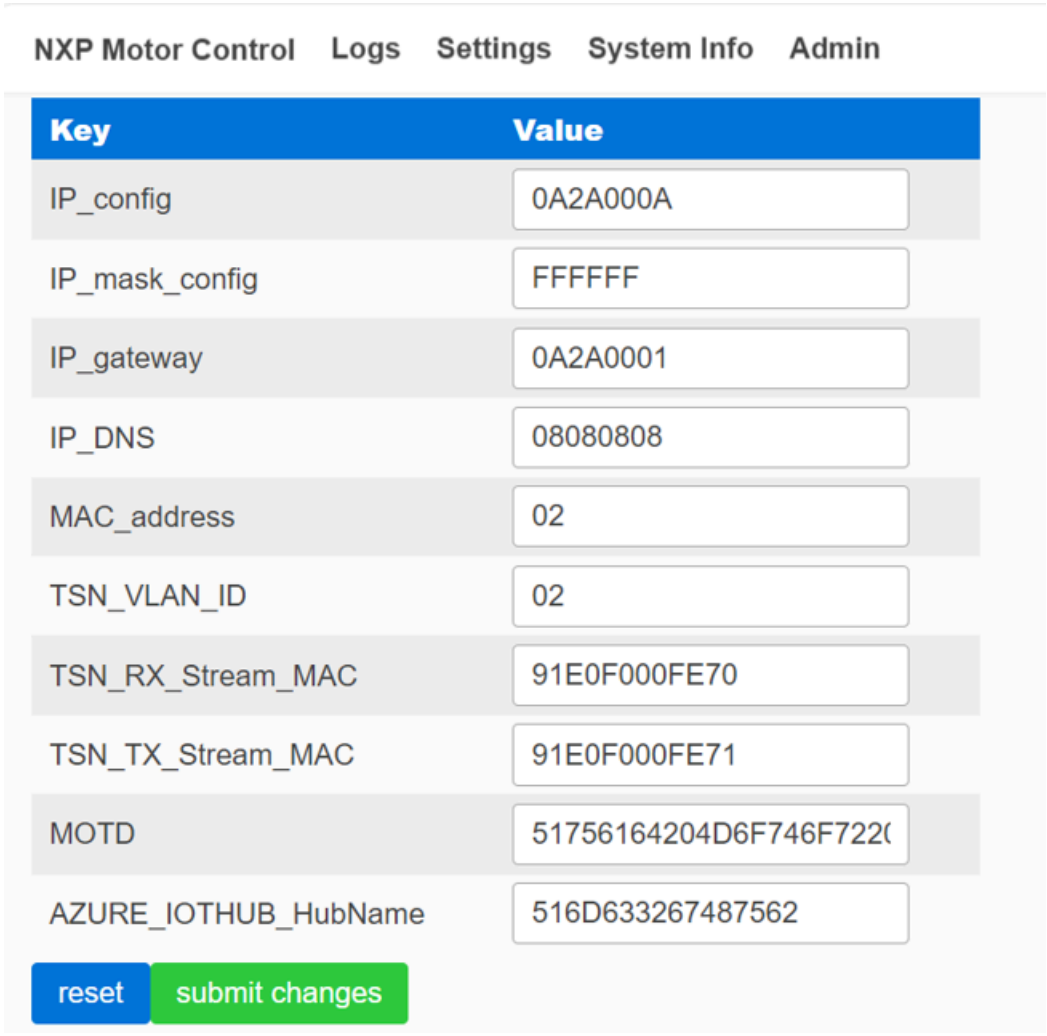


Figure 59. Settings tab

10. The System Info tab is used for the system time and lifecycle state configuration and FW updates:

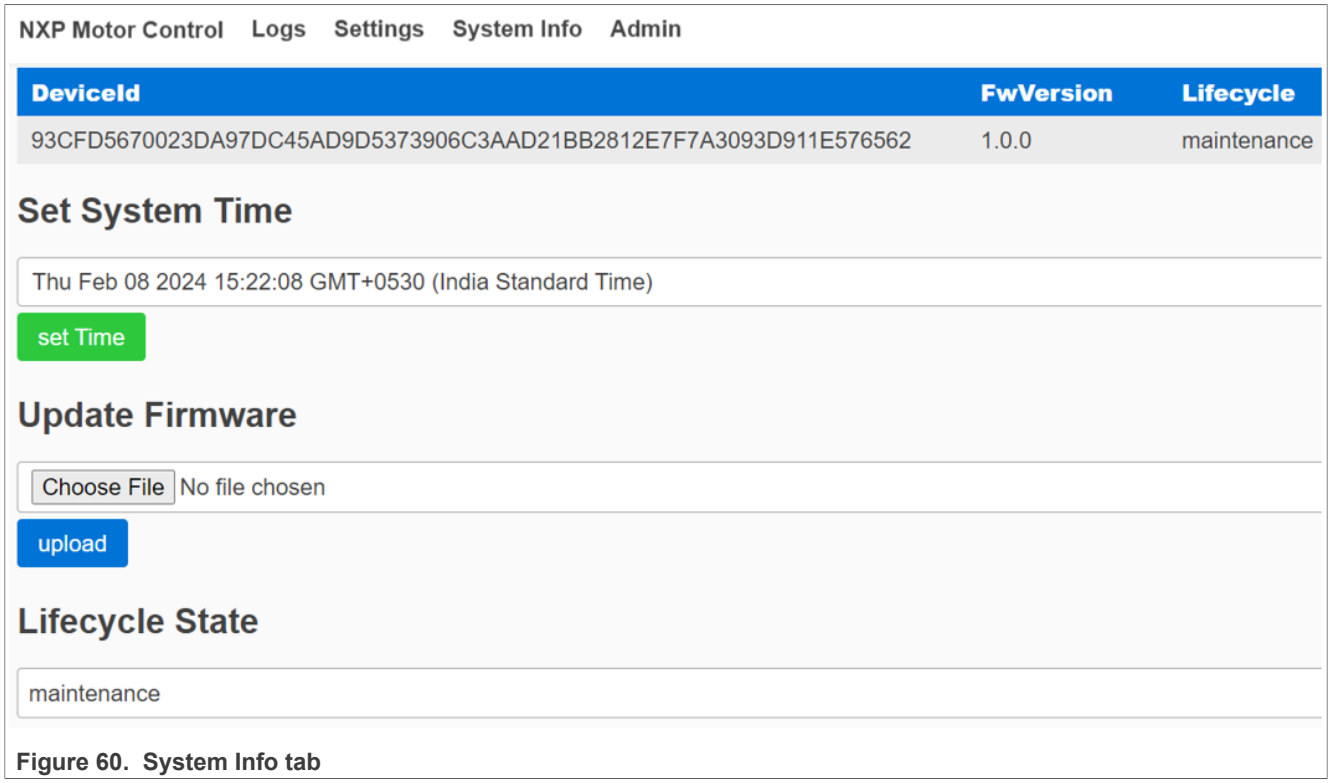


Figure 60. System Info tab

11. The Admin tab is for user management. Its features are only available to administrators:

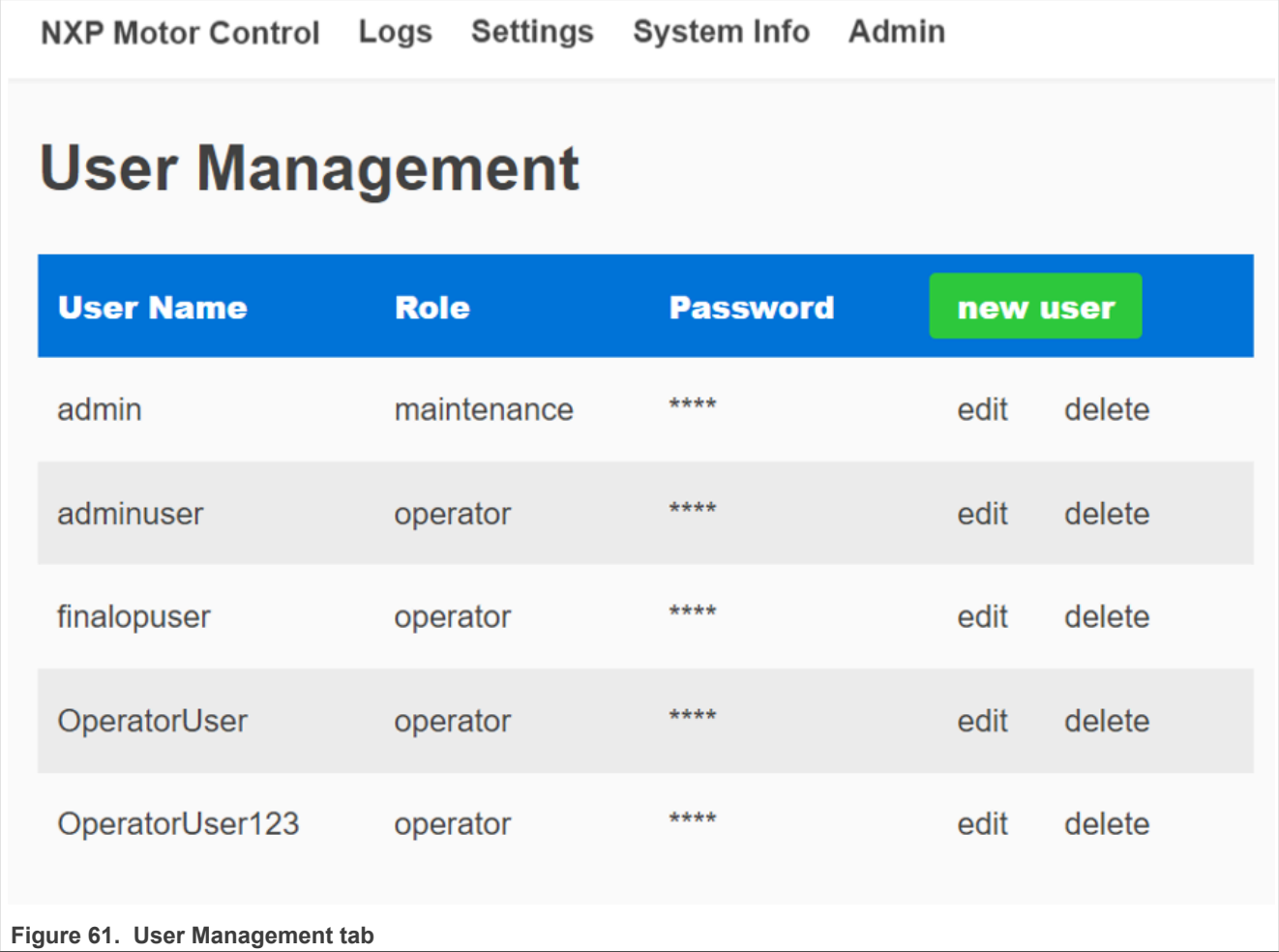


Figure 61. User Management tab

7 Spin the motors with FreeMASTER

This section describes how to spin the motors connected to the i.MX RT Industrial Drive Development Platform using the FreeMASTER GUI. If you have not installed FreeMASTER, follow the instructions in Section 5.2 to download the tool and install it. Make sure that the hardware has been correctly assembled (see Section 4) and the software is running correctly in the i.MX RT Industrial Drive Development Platform as described in Section 6.

1. In MCUXpresso, double-click the *pmsm_demo_QUAD_release.pmp* file located in the *freemaster_exe* folder of the *isi_qmc_dgc_industrial_app_master _cm7* project, as shown in Figure 62. FreeMASTER should open and display the interface shown in Figure 63.
   Note: *The error message next to each motor will disappear once FreeMASTER connects to the system in the next step.*

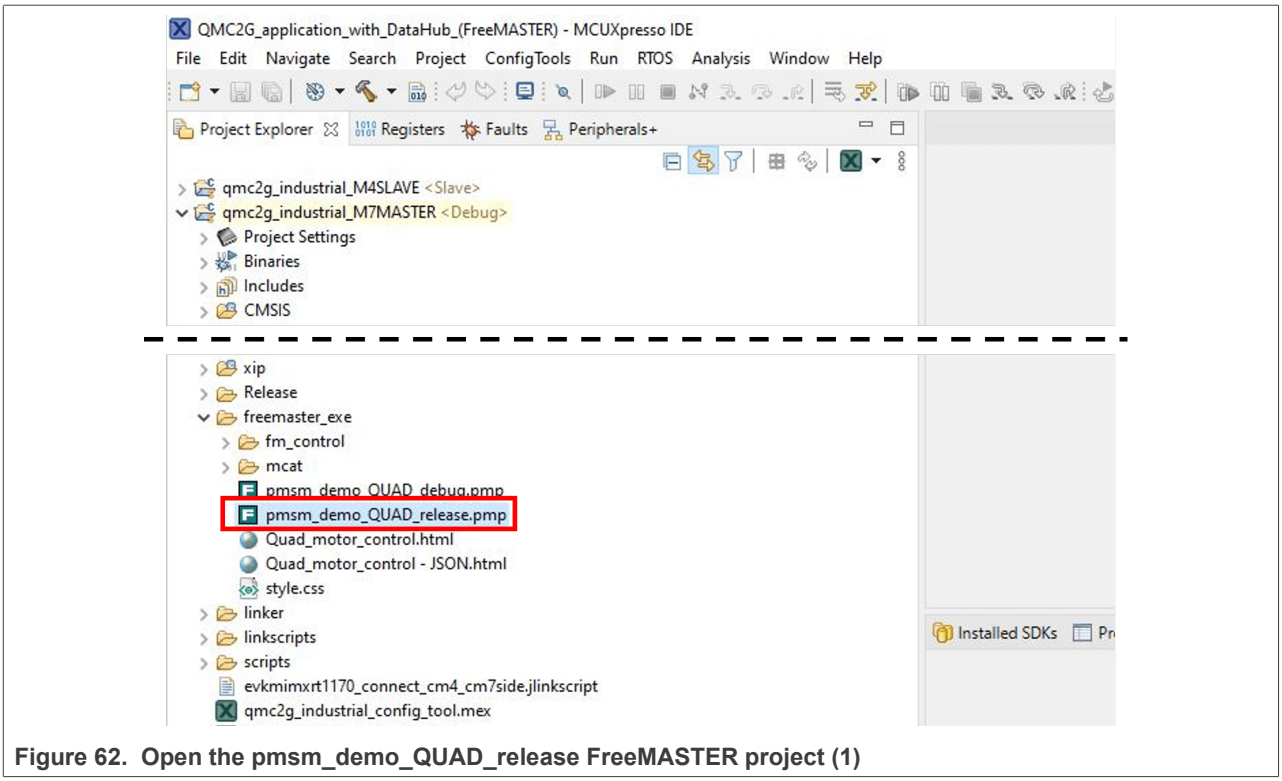


Figure 62. Open the pmsm_demo_QUAD_release FreeMASTER project (1)

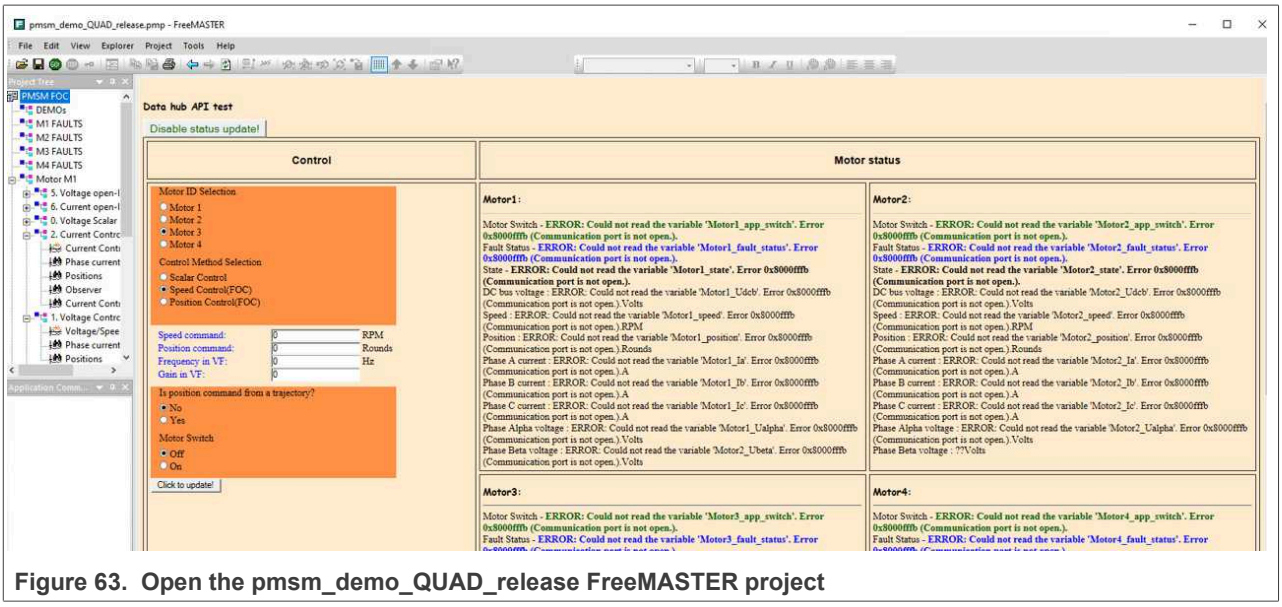


Figure 63. Open the pmsm_demo_QUAD_release FreeMASTER project

2. Open the FreeMASTER communication with the daughter card as shown in Figure 64. Before executing this step, make sure that the digital board is connected to the PC using the *J48* micro-USB connector (see Figure 47).
   (1) Click the *start communication* button in the top bar;
   (2) If the communication is successfully established, the state of the motors changes from *Error 0x8000fffb (Communication port is not open)* to *kMC_NoFaultMC* (fault status), *kMC_App_Off* (motor switch) and *kMC_Stop* (state). If fewer than four motors are connected, faults still appear for the missing motors but it does not prevent the connected motors to run.

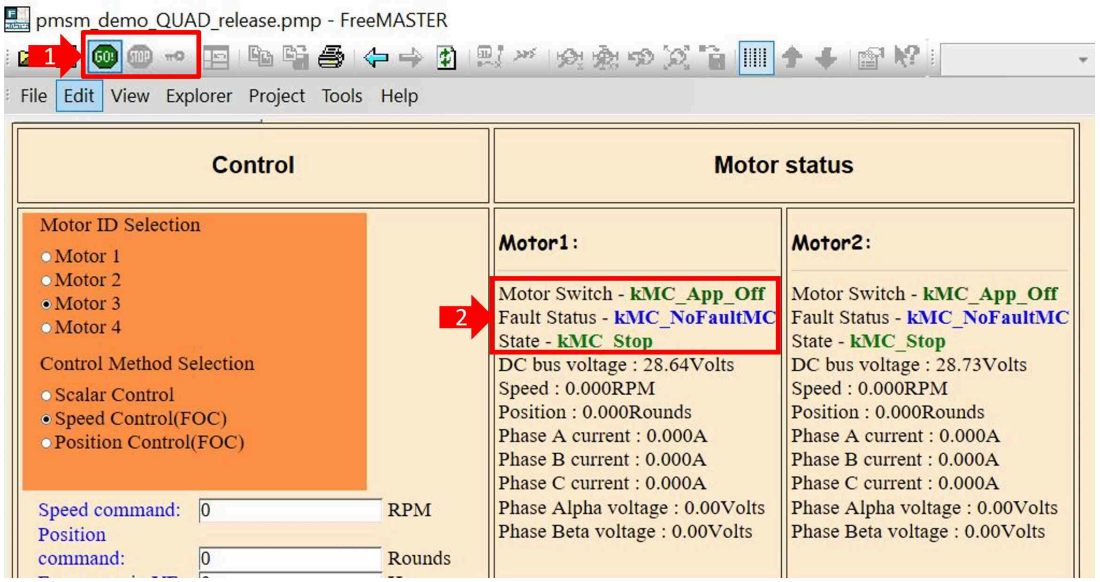


Figure 64. Start FreeMASTER communication

3. You can now control the motors using the FreeMASTER GUI as shown in Figure 65:
   (1) Select the motor to which you want to send a command;
   (2) Select the control method. If you select *speed control*, the motor will be controlled by speed commands (in RPM). If you select *position control*, the motor will be controlled by position commands (in rounds). If you select *scalar control*, the motor will be controlled by *frequency in VF* (Hz) and *gain in VF*;
   (3) Set the speed command value (if you set *speed control*), the position command value (if you set *position control*) or the *frequency in VF* (Hz) and *gain in VF* (if you set *scalar control*);
   (4) If *position control* has been selected, indicate if the position command is from a trajectory or not;
   (5) Indicate if you want to turn the selected motor on or off;
   (6) Click *click to update* button to send the motor control command to the system;
   (7) The selected motor should start spinning and the motor control data shall be displayed in real-time in FreeMASTER.
   Note: *Make sure to set only motor control values that do not exceed the limits of the motors you are controlling. Check the motor data sheet or contact the motor manufacturer for more information.*
   Note: *In the currently released version the buttons are disabled. The system assumes that they are under a locked cover and direct access without opening the cover would result in a tampering event. To enable the buttons and test this behavior, the FEATURE_HANDLE_BUTTON_PRESS_EVENTS macro in qmc_features_config.h must be set to 1.*

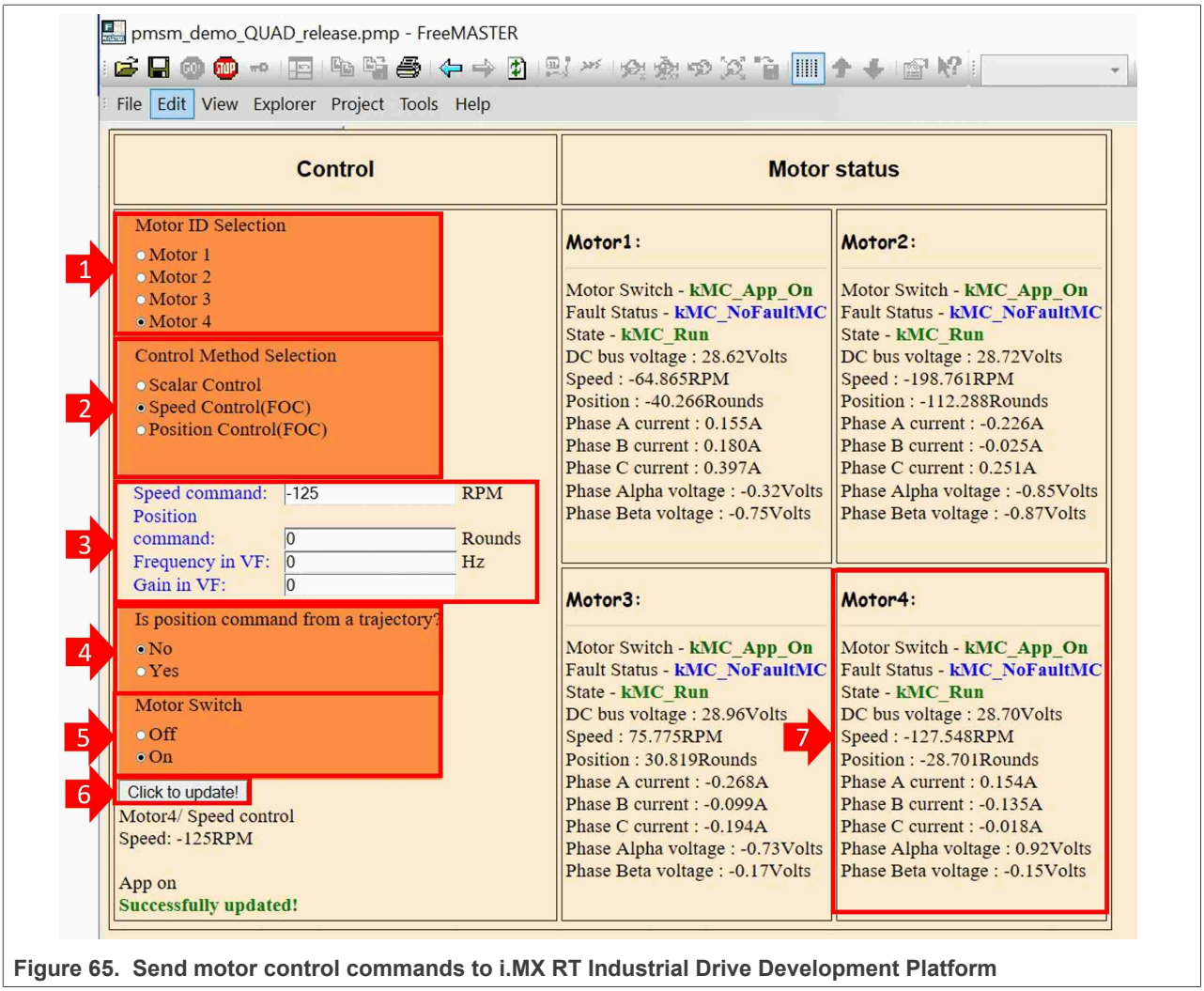


Figure 65. Send motor control commands to i.MX RT Industrial Drive Development Platform

8 Spin the motors with a TSN master device

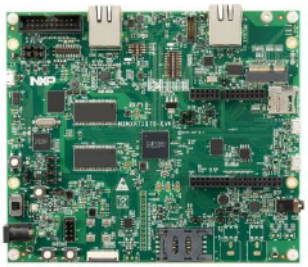
This section describes how to control the motors connected to the i.MX RT Industrial Drive Development Platform using a TSN master device. The TSN master device is simulated using the i.MX RT1170 EVK board. The example application sends a predefined sequence of motor control commands using the Ethernet connection. This section consists of the following subsections:

- Additional Hardware required
- Setting up the TSN master device with i.MX RT1170 EVK
- Flash the TSN master device and spin the motors

8.1 Additional hardware required

This section lists additional hardware material required to set up the TSN master device and run the example application. The required hardware components are listed in Table 7:

Table 7. Hardware required (TSN master demo)

Component Name	Description	Picture
i.MX RT1170 EVK	The i.MX RT1170 EVK integrates the i.MX RT1170 crossover MCU and provides a high-performance solution in a highly integrated board. The i.MX RT1170 EVK acts as a TSN master device.	
RJ45 Ethernet cable	An RJ45 Ethernet cable is required for connecting the i.MX RT1170 EVK (TSN master device) to the i.MX RT Industrial Drive Development Platform.	

8.2 Setting up the TSN master device with i.MX RT1170 EVK

This section explains how to prepare the i.MX RT1170 EVK board and connect it to the i.MX RT Industrial Drive Development Platform using the Ethernet cable:

1. Power up the using the 5V DC input as shown in Figure 66:
   (1) Set the power selection jumper (*J38*) in the *1-2* position;
   (2) Set the DIP switch (*SW1*) to internal boot mode: *SW1.1: OFF; SW1.2: OFF; SW1.3: ON; SW1.4: OFF*;
   (3) Connect the board to the power supply using the *5V DC IN* connector (*J45*);
   (4) Use the *SW5* ON-OFF switch to turn on the board. The *D16* LED right next to the switch should turn on indicating that the board is correctly supplied.

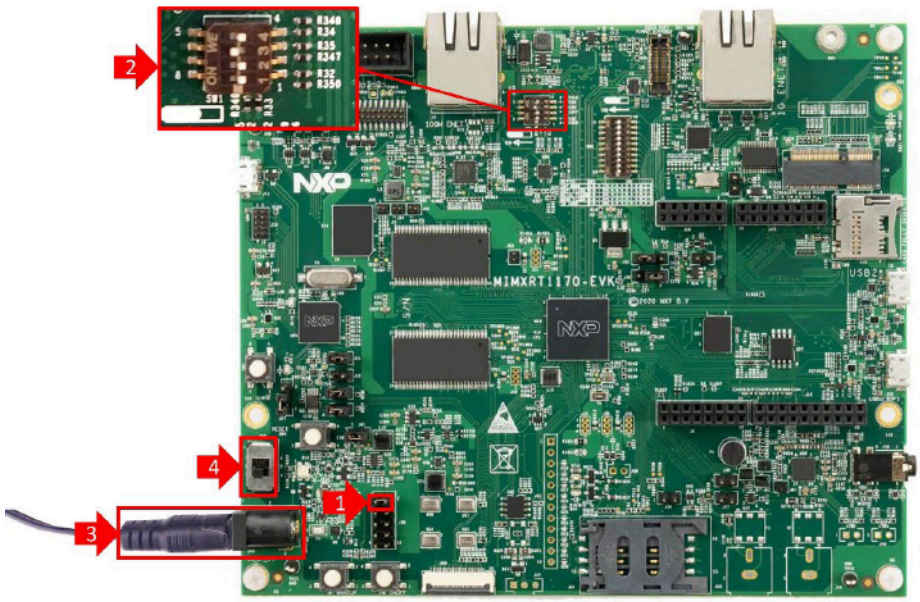


Figure 66. Power up the i.MX RT1170 EVK board

2. Connect the i.MX RT1170 EVK to the i.MX RT Industrial Drive Development Platform using an RJ-45 Ethernet cable as shown in Figure 67:
(1) Connect one end of the RJ-45 Ethernet cable to the Gigabit Ethernet connector of the i.MX RT1170 EVK board (*J3*);
(2) Connect the other end of the RJ-45 Ethernet cable to the TSN Ethernet connector of the daughter card (*J4*).

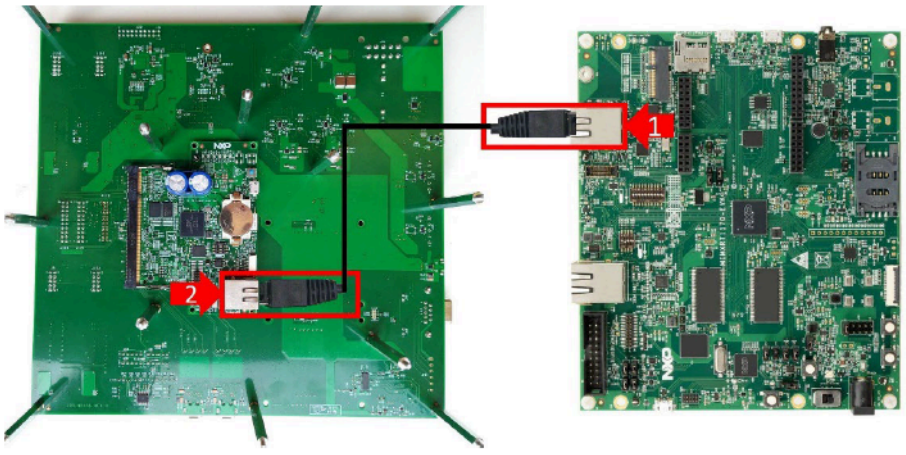


Figure 67. Connect the i.MX RT1170 EVK board to the daughter card using Ethernet cable

8.3 Flash the TSN master device and spin the motors

Follow the steps outlined in this section to flash the demo application in the i.MX RT1170 EVK. After flashing the board, the software starts running and sends a pre-defined sequence of motor control commands to the i.MX RT Industrial Drive Development Platform. When motor control commands are received by the daughter card, the motors connected to the system start spinning.

Note: *Make sure that the i.MX RT Industrial Drive Development Platform demo application is running as described in Section 6 before executing the steps described in this section.*

1. Connect the i.MX RT1170 EVK board to the PC using a micro-USB cable as shown in Figure 68:
   (1) Connect the micro-USB cable to the *J11* connector of the i.MX RT1170 EVK board;
   (2) If you open the Windows device manager, you should see a new *USB Serial Device*.

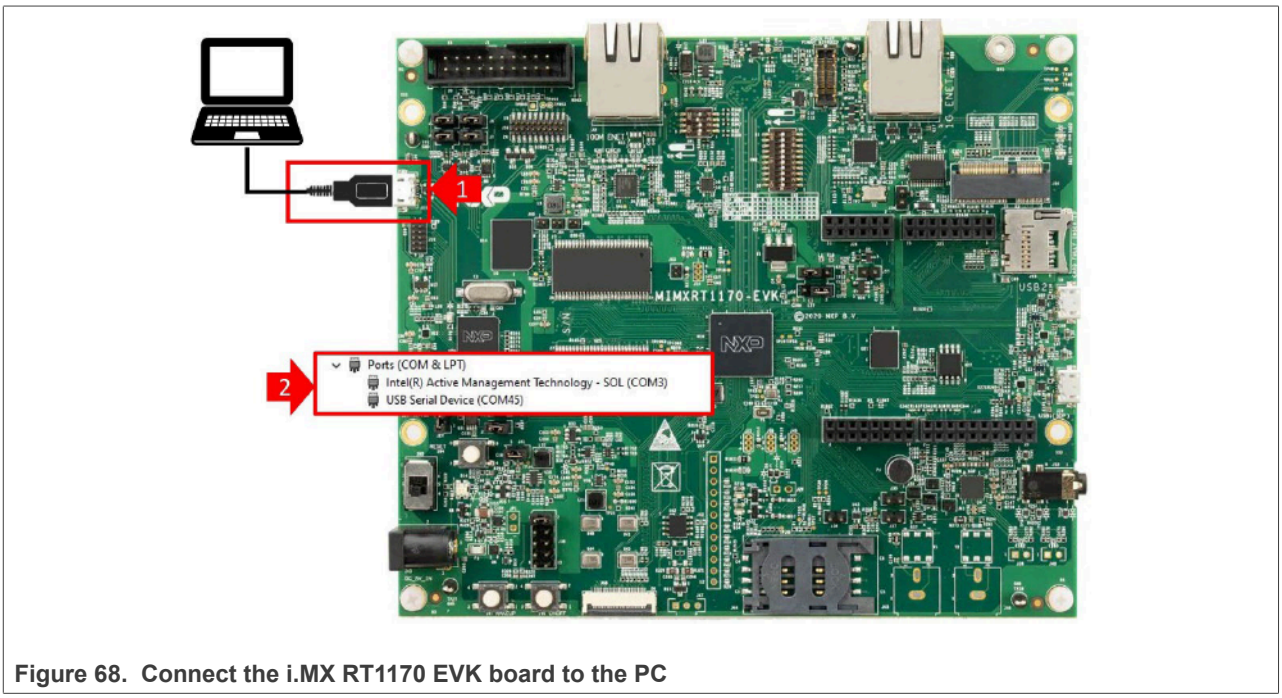


Figure 68. Connect the i.MX RT1170 EVK board to the PC

2. Flash the *imxrt1170evk_industrial_app_tsn_motion_controller.bin* file containing the demo application in the i.MX RT1170 EVK board using the DAP-Link interface of the board as shown in Figure 69. The binary file for i.MX RT1170 EVK is delivered with the i.MX RT Industrial Drive Development Platform software package. It can be found in the *tools/ folder*.
   (1) Open the *RT1170-EVK* drive from the Windows file explorer;
   (2) Drag and drop the *imxrt1170evk_industrial_app_tsn_motion_controller.bin* binary file in the drive. The window will close for a moment and reopen again after the software has been successfully flashed.

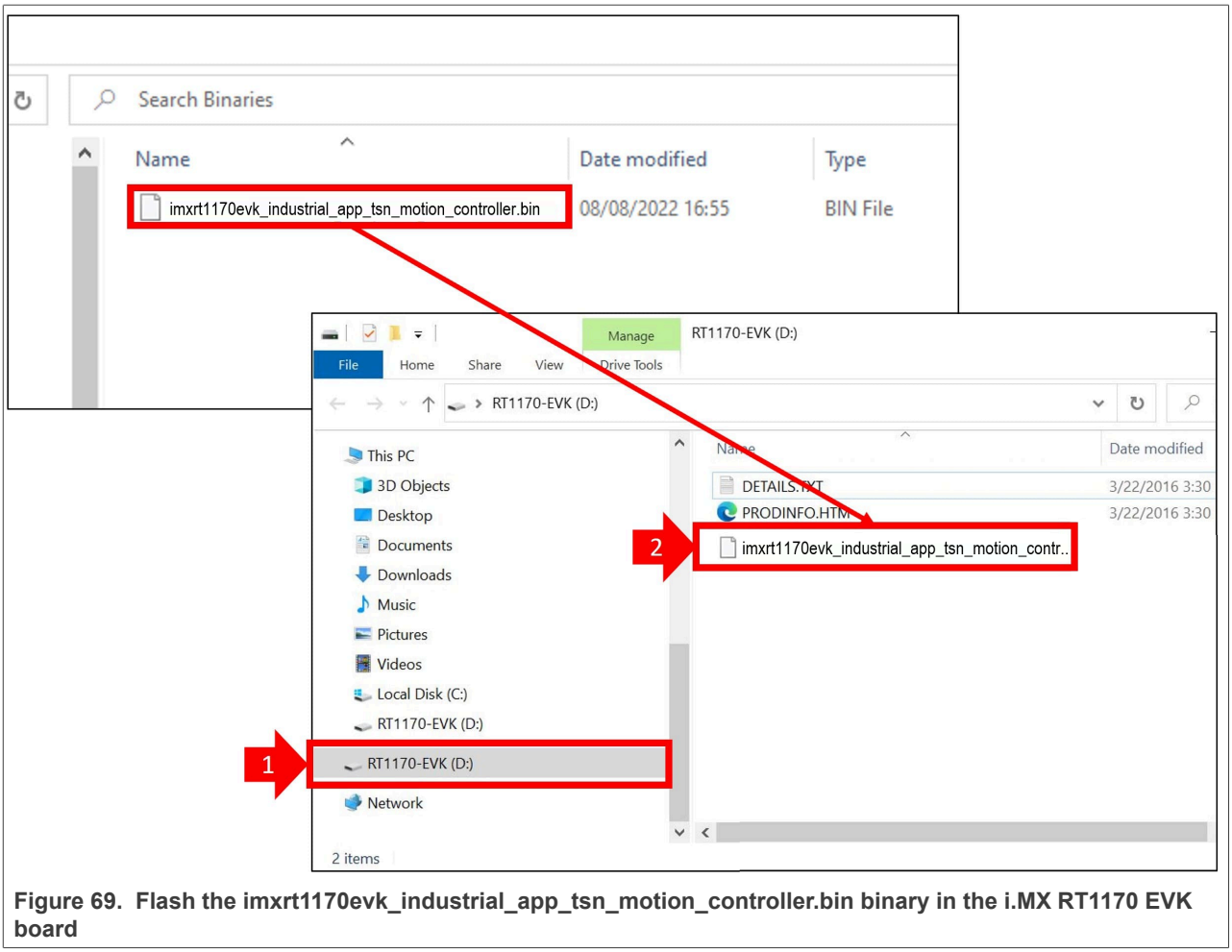


Figure 69. Flash the imxrt1170evk_industrial_app_tsn_motion_controller.bin binary in the i.MX RT1170 EVK board

3. Power cycle the i.MX RT1170 EVK board using the ON-OFF switch (*SW5*). The TSN master demo application will start running and will send motor control commands to the i.MX RT Industrial Drive Development Platform. If the demo is working correctly, you should see the motors spinning following a predefined sequence.
4. You can see the logs of the i.MX RT1170 EVK demo application by opening a serial connection to the board using any terminal application (in this document the Tera Term application has been used):
   (1) Select from the dropdown menu the serial port number obtained in Step 1;
   (2) Set the baud rate to *115200*;
   (3) You can leave the default value for the other settings (*data: 8 bit, parity: none, stop bits: 1 bit, flow control: none*);
   (4) Then, click the *New open* button. You should see logs as in Figure 71.

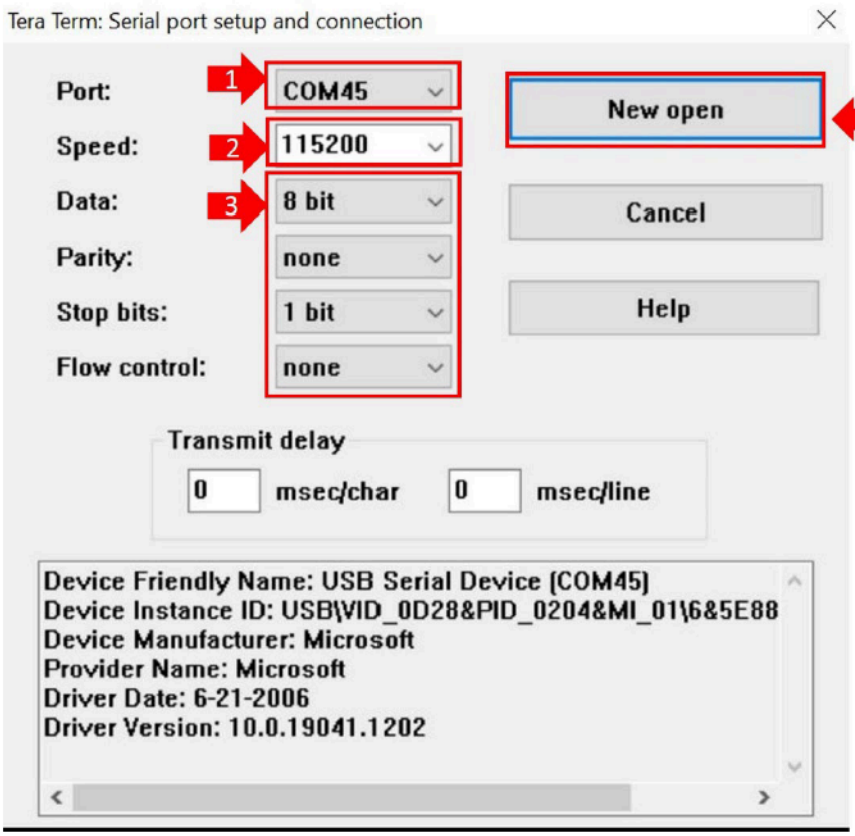


Figure 70. Launch serial terminal in MCUXpresso

Figure 71. Logs of i.MX RT1170 EVK demo application

9 Note about the source code in the document

Example code shown in this document has the following copyright. The source code is licensed with LA_OPT_NXP_Software_License v53 December 2023. More details in the License.txt and SW-Content-Register.txt files.

10 Revision history

Table 8. Revision history

Document ID	Release date	Description
AN13644_Rev.1.2	08 April 2024	Updated with changes belonging to application software pack release v1.2
AN13644_Rev.1.1	1 November 2023	Updated with changes belonging to application software pack release v1.1
AN13644_Rev.1.0	8 September 2023	Initial version

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 https://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 document 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.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell 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.

Contents

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

Date of release: 8 April 2024
Document identifier: AN13644