AN15104

Porting Guide for Industrial Ethernet Protocols

Rev. 1.0 — 31 July 2026 Application note

Document information

Information	Content
Keywords	AN15104, FRDM-IMXRT1186, i.MX RT1180, porting guide, industrial Ethernet protocols
Abstract	This application note introduces the method to enable the industrial Ethernet protocols like Profinet, EtherCAT, and EtherNet/IP on a new board like FRDM-IMXRT1186.

1 Introduction

This application note explains how to port NXP industrial Ethernet protocol projects, including EtherCAT, PROFINET, and EtherNet/IP, from the MIMXRT1180-EVK to the FRDM-IMXRT1186 platform. It covers board-level adaptations, PHY and flash configuration updates, project configuration changes, and protocol-specific modifications, and includes reference patches to simplify migration to custom hardware designs.

2 Overview

NXP supports a series of industrial network protocols based on the i.MX RT1180. On the i.MX RT1180 website, customers can download two types of resources for evaluating industrial networks.

The first resource is a binary file for the MIMXRT1180-EVK board. Download the binary file directly to the EVK board, and then follow the guide to test the application. The second resource is a VS Code project that contains the GOAL library. The project also targets the MIMXRT1180-EVK board by default.

You can compile the project to generate an ELF file for debugging. For customers who want to develop their own boards and use PORT protocols, the second resource serves as a good reference. You only need to modify a few files and configurations to enable the protocols on your own board.

Figure 1 shows the structure of the project that NXP currently provides. The green sections represent the source files included in the project. The yellow sections represent the libraries included in the project. Therefore, the porting effort focuses on the code in the green sections.

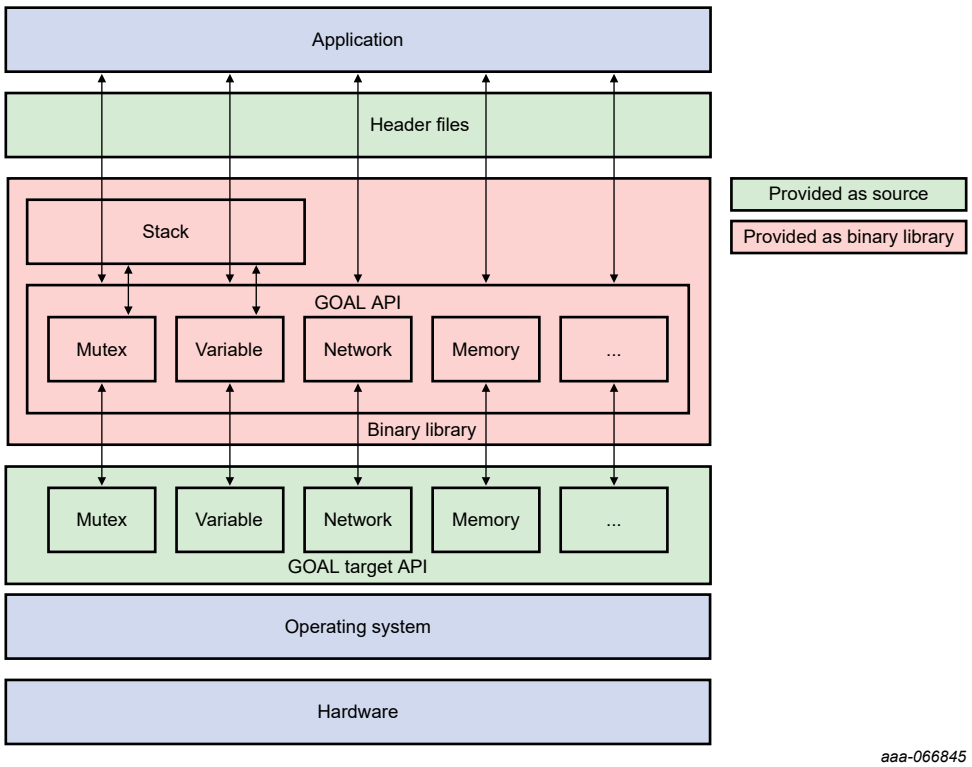

Figure 1. Project Structure Diagram

2.1 Migration strategy

The imported VS Code project path still contains the `mimxrt1180_evk` naming. This naming convention is intentional during the porting process. The port maintains the existing include and source paths. It replaces the

board-specific content under those paths with FRDM-MIMXRT1186 BSP files. This approach reduces project churn. It also isolates functional changes from large-scale, rename-only modifications. The porting did not change the SDK version that the EVK used, and the SDK used now is 25.9.0.

The migration follows these principles:

- Use the FRDM-MIMXRT1186 hardware as the source of truth for pin assignments, clock configuration, PHY reset, and external flash topology.
- Keep the upper-layer GOAL, EtherCAT, PROFINET, and EtherNet/IP application entry points unchanged.
- Defer path renaming until the migrated project is functionally validated.

3 EtherCAT porting

This section introduces how to port EtherCAT application from MIMXRT1180-EVK board to FRDM-MIMXRT1186 board.

3.1 Change summary

Table 1. EtherCAT porting changes summary

Directory	Main changes	Reason
`plat/board/nxp/evkmimxrt1180`	Replaced `board.c/h`, `clock_config`, and `pin_mux` content	Keep the source path stable while using the FRDM BSP implementation.
`plat/drv/nvs/sflash/fsl/rt1180`	Moved NVS from FLEXSPI1 to FLEXSPI2 and updated AMBA base handling.	The FRDM flash uses the different FLEXSPI interface from the EVK.
`protos/ethercat/drivers/board/nxp/evkmimxrt1180`	Changed EtherCAT ESC initialization from RMII to MII and updated PHY reset GPIO.	The FRDM EtherCAT PHY interface is wired differently from the EVK.
`projects\goal_ecat_rpc_lib\00_cc\vsc\nxp\mimxrt1180_evk_cm33_rtos`	Updated board ID/device part number, CPU macros, linker scripts, and TRDC.	Generate RT1186 startup, headers, drivers, and memory layout.
`projects\goal_ecat_rpc\04_led_button_eoe\vsc\nxp\mimxrt1180_evk_cm7_rtos`	Updated RT1186 CPU macros and memory map.	Keep the secondary image consistent with the CM33 sysbuild flow.

3.2 Code change

Board-level configuration

The board-level configuration consists of two parts. The first part involves board-level resource updates, primarily in `board.c/h`, `clock_config.c/h`, and `pin_mux.c/h`. The second part involves GOAL target updates, primarily in `goal_target_board.c/h`. For `board.c/h` and `clock_config.c/h`, use the content from the SDK `hello_world` demo to replace the content in the original files.

For `pin_mux.c/h`, enable the ECAT MII pin multiplexing and other ECAT-related pins, such as I2C and MDIO. You can obtain all the required changes from the patch included with this application note.

For `goal_target_board.c/h`, remove the NETC-related code because EtherCAT does not require these components. One important consideration is that the GOAL library may contain dependencies on the Ethernet interface. Therefore, do not disable the `GOAL_CONFIG_ETHERNET` macro solely to remove NETC-related configuration.

Platform configuration

The board layout also affects the platform configuration. The GOAL library uses NVS for certain operations. The EVK board connects the external flash through FlexSPI1. The FRDM board connects the external flash through FlexSPI2. Update the code to accommodate this difference.

EtherCAT interface configuration

The EtherCAT interface update is the most important hardware-specific change. The EVK implementation configures RMII pins and selects RMII mode in `ECAT_MISC_CFG`. The FRDM board requires MII pin groups instead. Therefore, update the pin initialization functions and clear the RMII selection bits. Also update the PHY reset pin and LED pin configurations to match the FRDM board.

3.3 Project configuration change

The project metadata and CMake definitions were updated together. This approach is necessary because the board ID, device part number, CPU macros, linker scripts, XIP boot headers, and TRDC components collectively determine the generated image. Updating only one of these elements can still produce a buildable image. However, the image may boot with an incorrect startup configuration or memory layout.

Some key changes are listed below:

```
"boardId": "frdmimxrt1186"
"deviceId": "MIMXRT1186xxxxx"
-DCPU_MIMXRT1186CVJ8C_cm33
-DMIMXRT1186_cm33_SERIES
-DXIP_BOOT_HEADER_ENABLE=1
CONFIG_MCUX_COMPONENT_driver.trdc=y
```

Table 2. EtherCAT project configuration changes

Area	Reason
CPU and series macros	Select correct RT1186 headers, startup, IRQ names, and register definitions.
XIP boot header	Required for FRDM flash boot flow.
TRDC component	Needed for RT1186 access control in multi-core/peripheral use cases.
Linker file	Boot flash change interface for i.MX RT1186.

3.4 Run the example

To run the example project, perform the following steps:

1. After the code and project configuration changes are applied, rebuild the project and download the generated image to the FRDM-IMXRT1186 board. The example used for the validation is the `goal_ecat_rpc 04_led_button_eoe` demo, in which the CM33 core runs the GOAL EtherCAT stack, and the CM7 core runs the LED/button application.
2. Connect the EtherCAT port0 of the FRDM-IMXRT1186 board to a PC running ICE, which acts as the EtherCAT MainDevice. After the board is powered up, the debug console prints the "EtherCAT CM33".
3. In ICE, scan the EtherCAT network. The FRDM-IMXRT1186 board is discovered as an EtherCAT SubDevice device. The device can be switched from the INIT state through PREOP and SAFEOP to the OP state. When the device is in the OP state, the cyclic process data exchange works normally: writing the LED output data in ICE can let the user LED light in different color on the board.

The ported example behaves the same on the FRDM-IMXRT1186 board as the original example on the MIMXRT1180-EVK board, which proves that the EtherCAT porting is successful.

4 Profinet and EtherNet/IP porting

Profinet and EtherNet/IP both rely on the NETC peripheral. As a result, the required porting changes are identical for both protocols.

4.1 Change summary

Table 3. Profinet and EtherNet/IP porting changes summary

Directory	Main changes	Reason
`plat/board/nxp/evkmimxrt1180`	Replaced board support content and updated startup, NETC ports, PHY reset, and NVS registration.	Keep source paths stable while using an FRDM-MIMXRT1186 BSP implementation.
`plat/drv/phy`	Replaced the Realtek RTL8201/RTL8211 path with a Motorcomm YT8531 PHY driver.	The FRDM PHY ID and link-status register model differ from Realtek devices.
`plat/drv/nvs/sflash/fsl/rt1180`	Moved NVS from FLEXSPI1 to FLEXSPI2 and updated AMBA base handling.	The FRDM serial flash window visible to CM33 differs from the EVK mapping.
`projects\goal_pnio_rpc_lib\00_cc\vsc\nxp\mimxrt1180_evk_cm33_rtos`	Updated board ID/device part number, CPU macros, linker scripts, TRDC, and XIP boot header.	Generate RT1186 startup, headers, drivers, and memory layout.
`projects\goal_pnio_rpc\31_led_button\vsc\nxp\mimxrt1180_evk_cm7_rtos`	Updated RT1186 CPU macros, TRDC, and memory map.	Keep the secondary image consistent with the CM33 flow.

4.2 Code change

Board-level configuration

The board-level configuration consists of two parts. The first part involves board-level resource updates, primarily in `board.c/h`, `clock_config.c/h`, and `pin_mux.c/h`. The second part involves GOAL target updates, primarily in `goal_target_board.c/h`.

For `board.c/h` and `clock_config.c/h`, use the content from the SDK `hello_world` demo to replace the content in the original files. For `pin_mux.c/h`, enable the NETC Switch Port 0 and Switch Port 2 pin multiplexing. Also, enable other NETC-related pins, such as MDIO.

You can obtain all the required changes from the patch included with this application note.

For `goal_target_board.c/h`, update the NETC-related code. The FRDM board supports only two switch ports. Therefore, remove the definitions and initialization code for the other three ports used on the EVK board. Initialize only Switch Port 0 and Switch Port 2. The FRDM board uses Switch Port 0 in RGMII 1000 Mbit/s mode. Update the MAC configuration accordingly. The FRDM board uses the YT8531 PHY, which differs from the RTL8211 PHY used on the EVK board. Therefore, update the PHY driver to support the YT8531 device.

Platform configuration

The platform configuration is also related to the board layout. The GOAL library uses NVS for several operations. The EVK board connects the external flash through FlexSPI1. The FRDM board connects the external flash through FlexSPI2. Update the code to accommodate this difference.

4.3 Project configuration change

Same as EtherCAT, the project metadata and CMake definitions were updated together. Update all the items listed below to configure the imported project to use the i.MX RT1186 SDK components and support the FRDM-MIMXRT1186 board.

Some key changes are listed below:

```
"boardId": "frdmimxrt1186"
"deviceId": "MIMXRT1186xxxxx"
-DCPU_MIMXRT1186CVJ8C_cm33
-DMIMXRT1186_cm33_SERIES
-DXIP_BOOT_HEADER_ENABLE=1
CONFIG_MCUX_COMPONENT_driver.trdc=y
```

Table 4. Profinet and EtherNet/IP project configuration changes

Area	Reason
CPU and series macros	Select correct RT1186 headers, startup, IRQ names, and register definitions.
XIP boot header	Required for FRDM flash boot flow.
TRDC component	Needed for RT1186 access control in multi-core/peripheral use cases.
Linker file	Boot flash change interface for i.MX RT1186.

4.4 Run the example

To run the example project, perform the following steps:

1. After the code and project configuration changes are applied, rebuild the project and download the generated image to the FRDM-IMXRT1186 board. The FRDM board uses different PHY for ECAT and NETC. Therefore, before powering up the FRDM board, change J12 and J18 to 1-2 connected to enable the NETC phys. The example used for the validation is the `goal_pnio_rpc 31_led_button` demo. Because EtherNet/IP shares the same NETC-related changes, the EtherNet/IP example can be verified in the same way.
2. Connect NETC Switch Port 0 or Switch Port 2 of the FRDM-IMXRT1186 board to the PC running the ICE. After the board is powered up, the debug console shows "Profinet CM33".
3. For Profinet, use ICE together with the GSDML file provided with the example. The controller discovers the device through DCP, assigns the device name and the IP address, and then establishes the application relation (AR). After the connection is established, the LED on the board can be controlled by the output data of the controller and the other use case can also get the same result as EVK run.

For EtherNet/IP, use ICE with the EDS file provided with the example. The scanner establishes a Class 1 (implicit) connection with the device, and the LED data is exchanged in the same way as in the Profinet example.

The ported examples behave the same on the FRDM-IMXRT1186 board as the original examples on the MIMXRT1180-EVK board, which proves that the Profinet and EtherNet/IP porting is successful.

5 Porting files

All changes required for the porting are included in the attached patch (AN15104SW.zip). Applying the patch makes it easier to run the evaluation project on the FRDM-IMXRT186 board. The ZIP file contains the patches and instructions for applying them.

6 Acronyms and abbreviations

Table 5. Acronyms and abbreviations

Acronym	Expansion
AMBA	Advanced Microcontroller Bus Architecture
BSP	Board Support Package
CMake	Cross-Platform Make
ECAT	EtherCAT
ELF	Executable and Linkable Format
ESC	EtherCAT Slave Controller
EVK	Evaluation Kit
FRDM	Freedom Development Platform
GOAL	Generic Open Abstraction Layer
GPIO	General-Purpose Input/Output
I2C	Inter-Integrated Circuit
IRQ	Interrupt Request
MAC	Media Access Control
MDIO	Management Data Input/Output
MII	Media Independent Interface
NVS	Non-Volatile Storage
PHY	Physical Layer Device
RGMII	Reduced Gigabit Media Independent Interface
RMII	Reduced Media Independent Interface
RTL	Realtek (device prefix used for PHY devices RTL8201/ RTL8211)
SDK	Software Development Kit
TRDC	Trusted Resource Domain Controller
XIP	Execute In Place

7 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2026 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

8 Revision history

Table 6 summarizes the revisions to this document.

Table 6. Revision history

Document ID	Release date	Description
AN15104 v.1.0	31 July 2026	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at 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.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

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