Application Note

Multi-Protocol Industrial Communication Design Based on TI Sitara PRU-ICSS

Wayne Li, Zekun Bai

Abstract

Industrial automation systems impose increasingly stringent requirements for deterministic, multi-protocol real-time communication. Conventional gateway designs rely on dedicated protocol ASICs or FPGAs, resulting in higher cost and limited protocol flexibility. TI's Sitara™ processors integrate the Programmable Real-Time Unit and Industrial Communication Subsystem (PRU-ICSS), embedding deterministic 32-bit RISC cores alongside the Arm® application processor on a single SoC, with single-cycle non-pipelined execution and independent firmware loading per protocol. This document describes the PRU-ICSS architecture, the technical mechanisms of EtherCAT, PROFINET, and EtherNet/IP, and multi-protocol design examples, providing industrial equipment manufacturers with a practical multi-protocol communication design reference based on PRU-ICSS.

Table of Contents

Trademarks

1 Introduction

With the rapid development of Industry 4.0 and intelligent manufacturing, industrial Ethernet protocols have gradually become the mainstream communication infrastructure for industrial automation. Multi-protocol coexistence is a common phenomenon in industrial environments. North American markets favor EtherNet/IP, European markets widely adopt PROFINET and EtherCAT, motion control applications tend to prefer EtherCAT or PROFINET IRT, while OEM manufacturers serving global supply chains often need to support multiple protocols simultaneously. This presents a practical challenge for equipment manufacturers: how to use a unified hardware platform to cover the protocol requirements of different regions and industries. TI has addressed this challenge by integrating the PRU-ICSS subsystem into Sitara processors, implementing a flexible and efficient multi-protocol communication engine and providing industrial equipment manufacturers with an integrated design.

2 PRU Feature Comparison

The Programmable Real-Time Unit (PRU) is a 32-bit deterministic RISC core. The PRU delivers the real-time behavior of an FPGA while retaining the programmability and lower complexity of a CPU. The PRU is specifically designed for ultra-low-latency control of I/O pins (Figure 2-1):

- Single cycle execution (5ns, non pipelined CPU)
- Broadside interface (1024 bit data bus)
- Multi-core real-time processing (5ns parallel sync)
- Two cycle interrupt latency (10ns task switch)
- Two cycle arbitration (10ns spinlock)

The PRU is embedded within select TI processors and microcontrollers. Because the PRU has access to device pins (I/O), events, and all relevant SoC resources, it is capable of:

- Performing high-speed real-time response: on-the-fly data processing
- Executing specialized data processing operations, such as computing CRC for custom polynomials, FFT, FIR, Sin…
- Implementing custom peripheral interfaces through bit-banging, such as parallel ADC/DAC data acquisition interfaces, multi-channel ADC SPI timing sequences, 1-Wire protocol, QEP, PWM, and LVDS
- Offloading tasks from the SoC CPU: for example, servicing high-speed UARTs, maintaining large buffers, and handing data off to the main Arm core to maintain no data loss

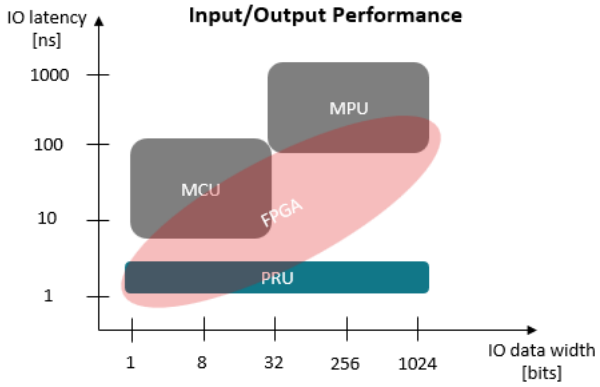

Figure 2-1. PRU Input/Output Performance

The PRU subsystem (Figure 2-2) consists of two real-time cores, each with a fixed 5ns per-instruction execution time and no instruction pipeline, verifying strictly deterministic real-time performance. Each PRU core has a dedicated instruction and data memory, and collaborates with the Arm cores through shared memory, enabling flexible task load distribution.

The subsystem connects to system resources through a 32-bit interconnect bus and is equipped with dedicated peripherals, including MII, MDIO, UART, an interrupt controller, a IEP, and a MPY/MAC unit, all of which access the real-time cores directly to minimize latency.

Each PRU supports up to 30 inputs and 32 outputs, allowing precise monitoring of and response to external events within a predictable time window, making it preferred for real-time applications such as Industrial Ethernet and custom I/O interfaces.

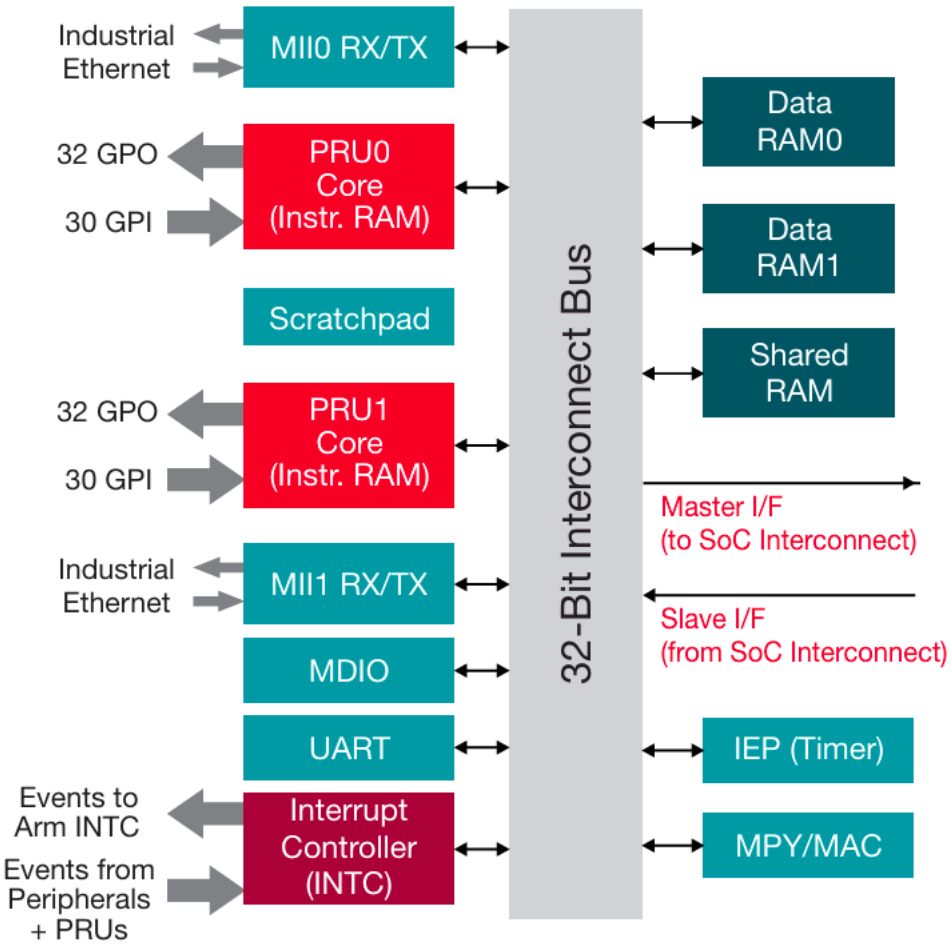

Figure 2-2. The TI PRU Subsystem

2.1 PRUSS: The Programmable Real-time Unit Subsystem

The Programmable Real-time Unit Subsystem (PRUSS) is the foundational member of the PRU subsystem family. The PRUSS comprises two 32-bit RISC PRU cores, shared memory, an Interrupt Controller (INTC), and direct access to I/O pins and SoC resources. The programmable nature of the PRU cores enables PRUSS to efficiently handle real-time processing, implement custom peripheral interfaces, and offload time-critical tasks from other processor cores.

PRUSS does not support industrial communication subsystem functionality — Ethernet-related signals (MII/MDIO) are not exposed.

Supported devices include the AM62x.

2.2 PRU-ICSS: The Programmable Real-time Unit and Industrial Communication Subsystem

The Programmable Real-time Unit and Industrial Communication Subsystem (PRU-ICSS) includes all PRUSS features and adds an integrated industrial communication subsystem for industrial networking applications.

PRU-ICSS supports EtherCAT, PROFINET, EtherNet/IP, PROFIBUS, EnDat 2.2, and 100Mbit industrial Ethernet communication.

Supported devices include the AM261x, AM263x, and AM263Px, as well as older devices AM335x, AM437x, and AM57x.

2.3 PRU_ICSSG: The Programmable Real-time Unit and Industrial Communication Subsystem - Gigabit

The Programmable Real-time Unit and Industrial Communication Subsystem - Gigabit (PRU_ICSSG) is the highest-end version of the PRU subsystem family. PRU_ICSSG includes all the features of PRU-ICSS, adding two Auxiliary Programmable Real-Time Unit (RTU) cores, two Transmit PRU (TX_PRU) cores, broadside memories, improved event management with the task manager, data processing and data movement accelerators, and new peripherals such as PWM.

PRU_ICSSG is capable of handling high-performance scenarios such as gigabit industrial communications, advanced real-time processing, and multi-axis motor control, making it the most fully featured design across the three generations of subsystems.

Supported devices include the AM64x and AM243x.

2.4 PRU Subsystem Feature Comparison

Figure 2-3 summarizes and compares the differences among the three PRU subsystems.

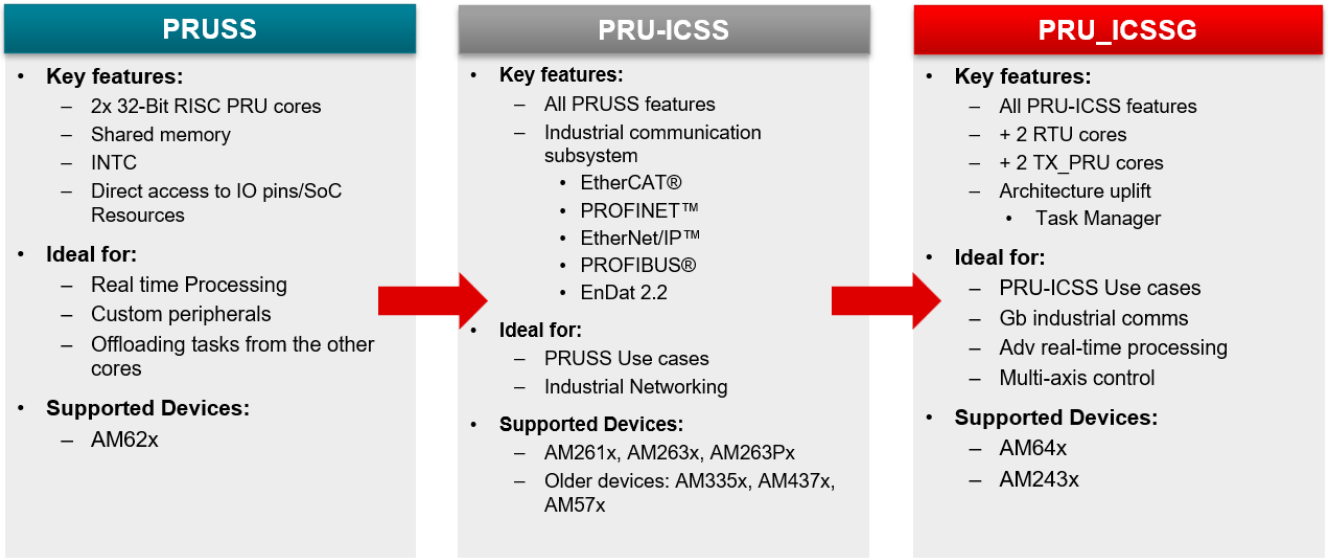

Figure 2-3. PRU Subsystem Comparison

Table 2-1 provides an overview of the key features across the PRU subsystems.

Table 2-1. PRU Subsystem Feature Comparison

Category	Feature	PRUSS	PRU-ICSS	PRU_ICSSG
General PRU Specifications	PRU cores	Yes	Yes	Yes
	RTU_PRU cores	No	No	Yes
	TX_PRU cores	No	No	Yes
	IRAM (per core)	Yes	Yes	Yes
	DRAM (2 DRAMs per PRU-ICSS/ PRU_ICSSG)	Yes	Yes	Yes
	Shared DRAM	Yes	Yes	Yes
	INTC	Yes	Yes	Yes
General Purpose Inputs	Direct Input	Yes	Yes	Yes
	16-bit Parallel Capture	Yes	Yes	Yes
	28-bit Shift	Yes	Yes	Yes
	3 Ch. Peripheral Interface (EnDAT)	No	Device dependent	Yes
General Purpose Outputs	Direct Output	Yes	Yes	Yes
	Shift out	Yes	Yes	Yes
Accelerators: Data Processing	MPY/MAC	Yes	Yes	Yes
	CRC 16/32	Yes	Device dependent	Yes
	Scratch Pad	Yes	Yes	Yes
	Broadside RAM	No	No	Yes
	Task Manager	No	No	Yes
	Spinlock	No	No	Yes
Accelerators: Data Movement	XFR2VBUS	Yes	No	Yes
	XFR2TR	No	No	Yes
Peripherals	UART	Yes	Yes	Yes
	IEP	Yes	Yes	Yes
	MII_RT or MII_G_RT	No	Yes	Yes
	MDIO	No	Yes	Yes
	PWM	No	No	Yes
Application	Real-time Processing	Yes	Yes	Yes
	Custom Peripherals	Yes	Yes	Yes
	Industrial Communication	No	Yes	Yes
	Multi-axis Control	No	No	Yes
Supported	Representative Devices	AM62x	AM261x, AM263x, AM335x, AM437x, AM57x	AM64x, AM243x

3 Mainstream Industrial Protocols

3.1 EtherCat: Ethernet for Control Automation Technology

Figure 3-1 illustrates a typical EtherCAT network topology, where an EtherCAT Master communicates with multiple device nodes, including Drive, Sensor, Analog I/O, and Digital I/O, in a daisy-chain manner. A key feature of EtherCAT is the *Processing on the Fly* mechanism: as a data frame passes through each device node, the device directly extracts the command data addressed to it from the data stream and simultaneously inserts its response data back into the same frame, enabling extremely low-latency, real-time communication across the network.

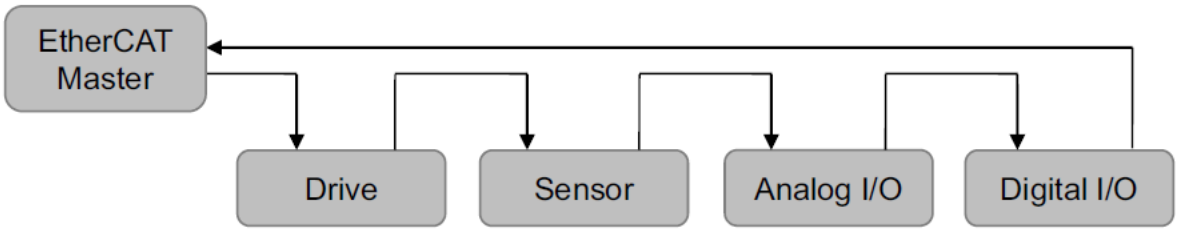

Figure 3-1. EtherCAT Network Example

EtherCAT telegram (Figure 3-2) is embedded in the payload of an Ethernet frame and may contain one or more EtherCAT datagrams destined for EtherCAT devices. Each datagram consists of a Header, Data, and a Working Counter. The Header and Data specify the operation the device must perform, while the Working Counter is incremented by each device that successfully processes the command, allowing the master to confirm that all intended devices have handled the datagram correctly.

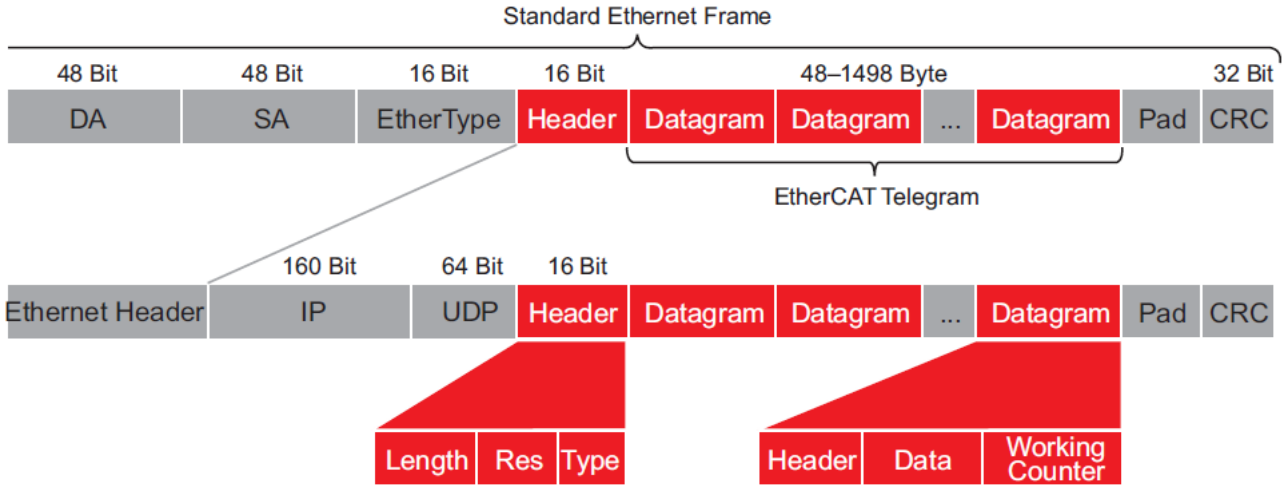

Figure 3-2. EtherCAT Telegram

EtherCAT firmware architecture (Figure 3-3) is divided into two PRU cores: PRU0 handles port RX1/TX0, the Host Interface, Sync Managers, and FMMU (Fieldbus Memory Management Unit), while PRU1 is responsible for port RX0/TX1, Distributed Clocking, and Error Handling. The two cores share access to EtherCAT Registers, Shared Memory, and Digital I/O resources, enabling coordinated real-time processing. The entire firmware stack is managed through a PRU Subsystem Driver and Host API layer, over which the EtherCAT Device Stack operates, interfacing with the hardware through MII, MDIO, and Digital I/O physical interfaces.

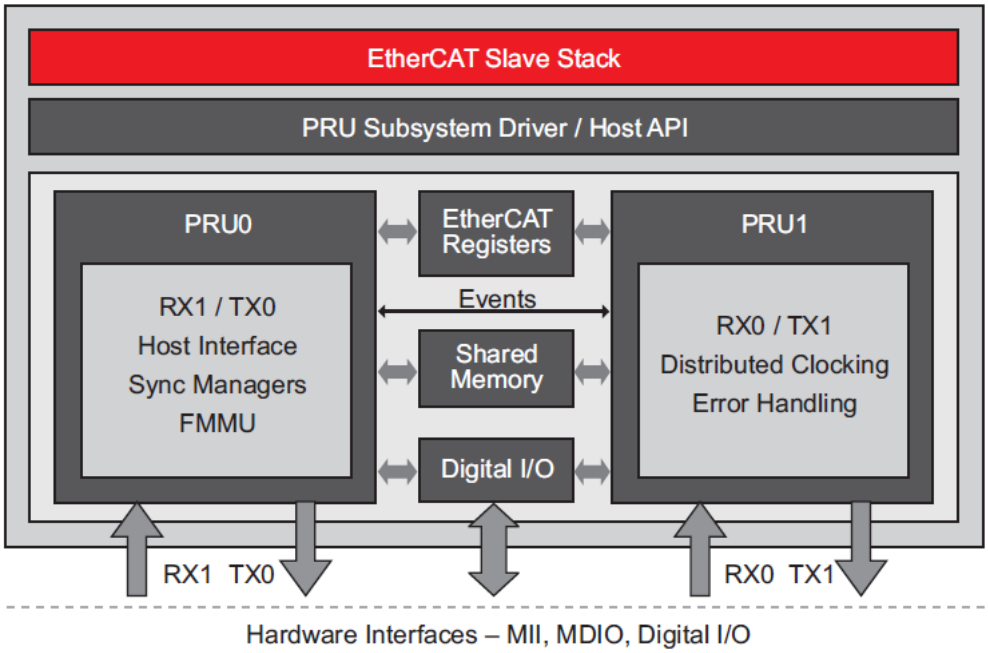

Figure 3-3. EtherCAT Firmware Architecture on PRU-ICSS

3.2 PROFINET: Real-Time Ethernet Standard

PROFINET is a real-time Ethernet standard for the high-speed, deterministic communications required for a wide range of industrial applications including factory automation, process automation, and building automation.

PROFINET standard defines four conformance classes that build upon one another and are oriented to serve different applications and use-cases:

- Conformance Class A (CC-A): can be implemented with standard Ethernet hardware and supports basic PROFINET functions such as cyclic Real-Time (RT) communication with update times between 1ms to 512ms.
- Conformance Class B (CC-B): extends CC-A with network diagnostics via SNMP, system redundancy and network redundancy through Media Redundancy Protocol (MRP).
- Conformance Class C (CC-C): extends CC-B with the support of cyclic Isochronous Real-Time (IRT) communication with update times less than 250µs.
- Conformance Class D (CC-D): this is the most advanced PROFINET class. It utilizes Time-Sensitive Networking (TSN) capabilities such as Time-Aware Shaper (TAS) and frame preemption to achieve very high determinism and low latency.

PROFINET standard defines three types of devices: IO Controller, IO-Device and IO-Supervisor. An example of a simplified network architecture inside a factory plant is provided in Figure 3-4.

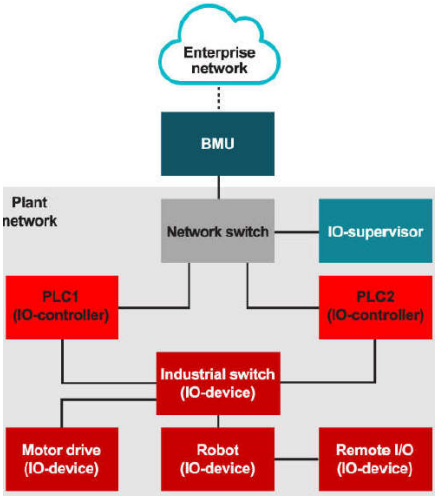

Figure 3-4. PROFINET Network Example

The PROFINET device implementation integrated on Sitara processors (Figure 3-5) has three major software components. The first is microcode that implements Layer 2 functionality in the device's PRU-ICSS; the second is the PROFINET device stack that runs on the Arm core; and, the third is the industrial application. TI provides additional components such as the protocol adaptation layer and device drivers in the software development kits that support Sitara processors.

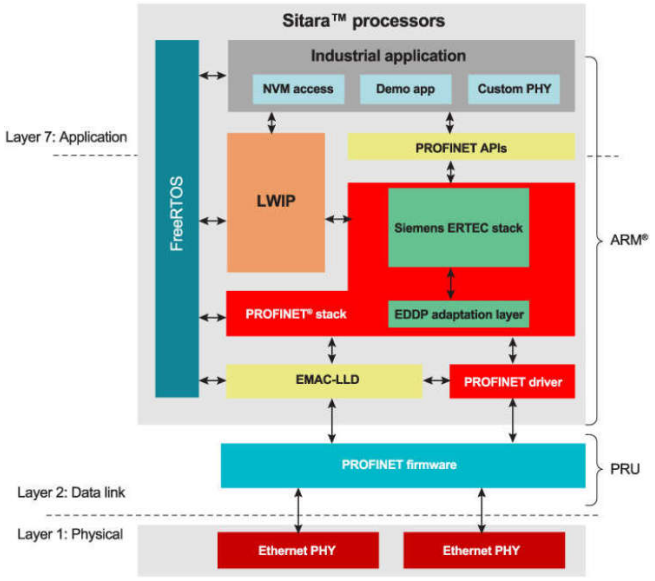

Figure 3-5. PROFINET Software Architecture on Sitara Processors

As shown in Figure 3-6, the PRU-ICSS integrates complete PROFINET Layer 2 functions, including CPM/PPM processing, DHT, DCP filtering, ARP filtering, cut-through switching, and error detection. Its internal shared memory provides a PROFINET register space, and combined with the PRU's deterministic real-time processing capability, verifies consistent and predictable PROFINET frame processing latency.

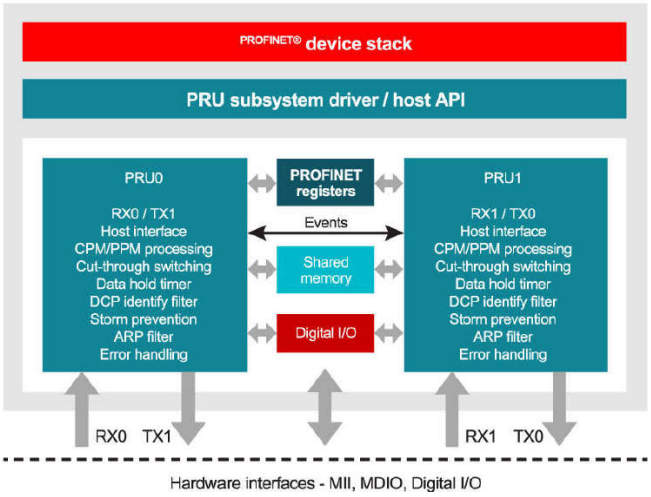

Figure 3-6. PROFINET Firmware Architecture on PRU-ICSS

3.3 EtherNet/IP: Common Industrial Protocol

EtherNet/IP is a member of a family of network protocols that implements the Common Industrial Protocol (CIP) at its upper layers. EtherNet/IP is the name given to CIP when it is implemented over standard Ethernet as defined by IEEE 802.3. Figure 3-7 illustrates the three software components required for EtherNet/IP device implementations on TI Sitara devices: PRU-based Layer 2 firmware, an Arm-hosted EtherNet/IP device stack, and the industrial application.

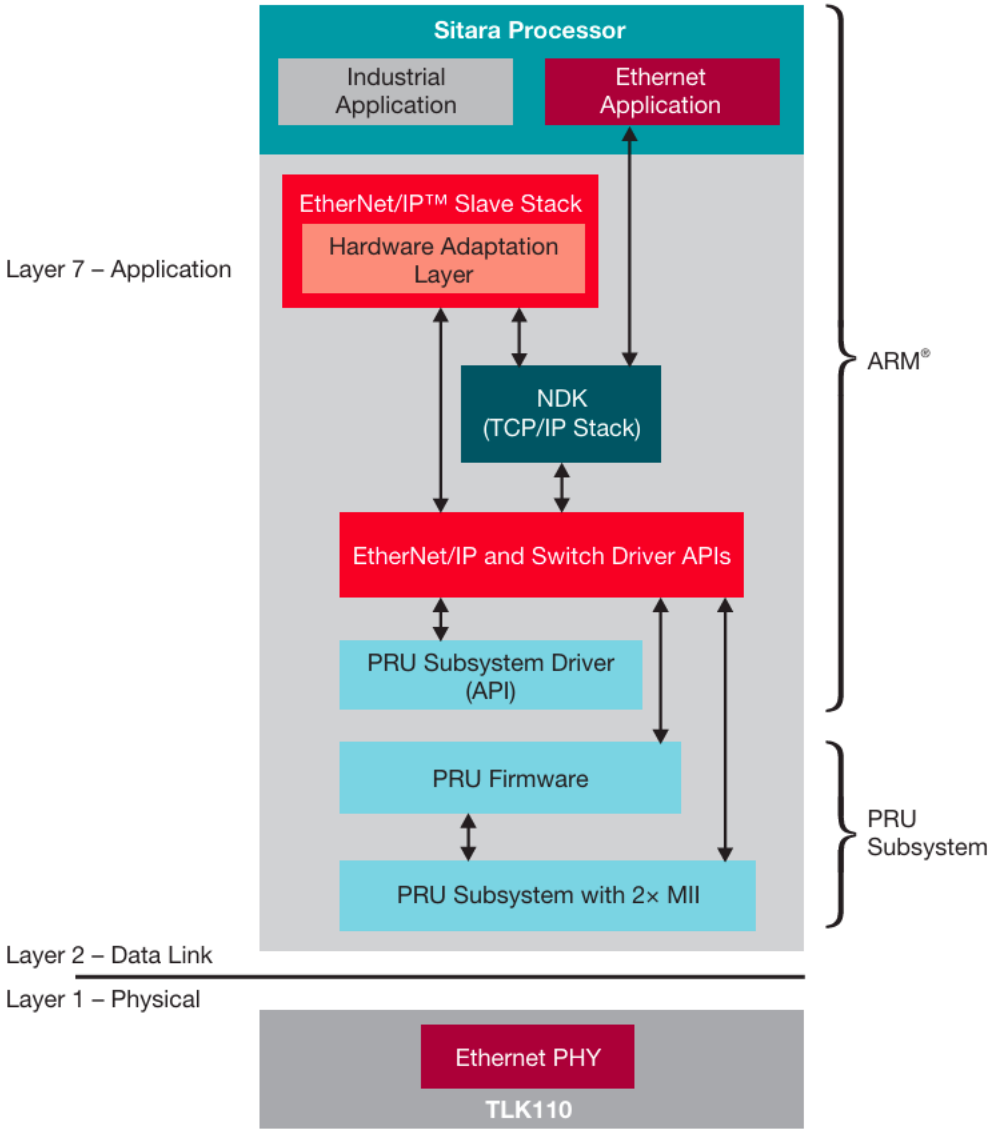

Figure 3-7. EtherNet/IP Software Architecture on Sitara Processors

The PRU-ICSS firmware architecture for EtherNet/IP integration on TI Sitara devices is shown in Figure 3-8. The PRU-ICSS handles core Ethernet switch functions, including MAC learning, storm prevention, and packet statistics. Each of the two PRU cores independently manages one physical port, handling a dedicated RX/TX pair. Inter-core coordination is achieved via specialized instructions and shared memory. The architecture supports low-latency store-and-forward switching with configurable parameters, and allows PRUs to interrupt the Arm core in real time for deterministic processing.

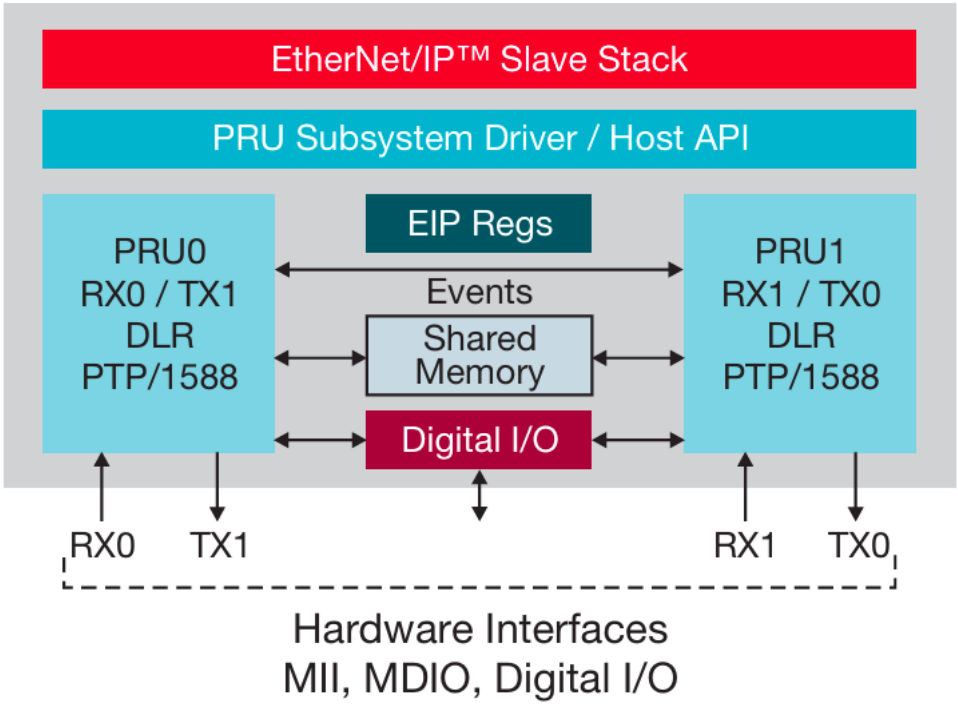

Figure 3-8. EtherNet/IP Firmware Architecture on PRU-ICSS

4 Multi-Protocol Industrial Communication Architecture Design Based on PRU-ICSS

4.1 Hardware Architecture

TI's PRU form the basis for a real-time subsystem on the processor. As shown in Figure 4-1, each PRU runs at 200MHz with single-cycle, non-pipelined execution, giving the SoC direct and deterministic access to external I/O. Each PRU has a dedicated Instruction RAM and Data RAM, so real-time firmware runs without memory contention or unpredictable delays. A shared RAM region allows efficient data exchange between the PRU cores and the Arm core. Incoming interrupts are delivered directly to the PRU through the integrated INTC, bypassing the multi-layer interconnect path and keeping response latency bounded and predictable.

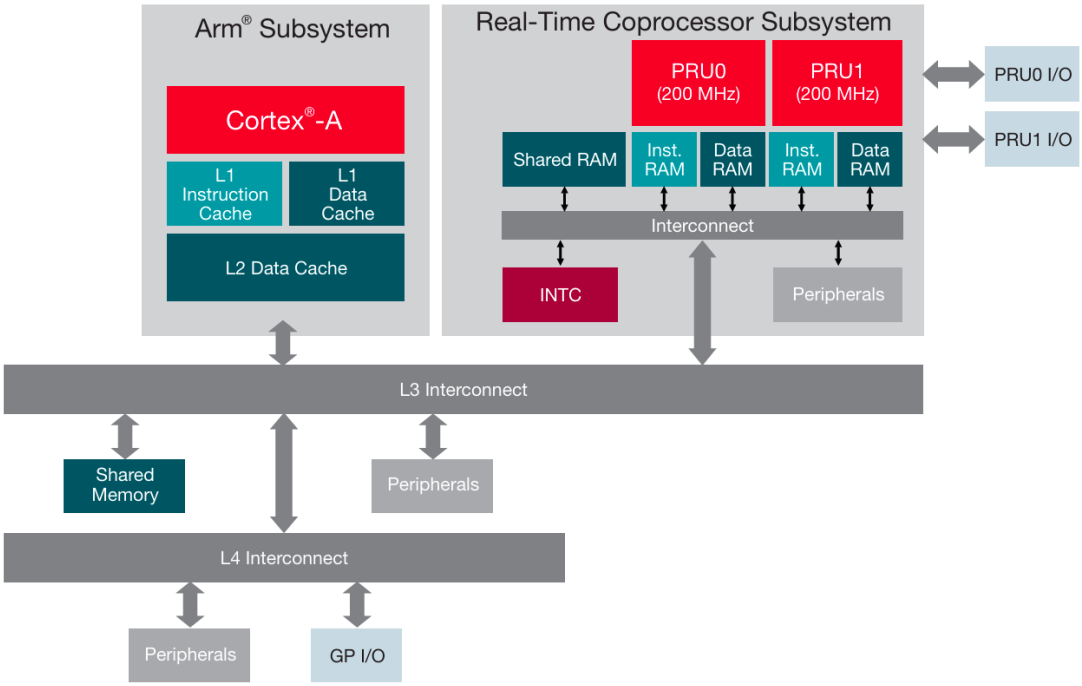

Figure 4-1. PRU-ICSS Multi-Protocol Communication Hardware Architecture

4.2 Software Architecture

The software architecture (Figure 4-2) tightly couples the Arm core and the PRU subsystem through a layered Linux driver stack. A Linux kernel running on the Arm core provides two driver mechanisms for PRU communication: RemoteProc and rpmsg. RemoteProc serves as the basic control mechanism: it reads the device tree blob (DTB) passed from U-Boot, loads PRU firmware into instruction memory, and starts PRU execution. Once the PRU is running, rpmsg handles bidirectional message passing between the PRU firmware and user space applications through a shared vring buffer in PRU data memory, exposed through the /dev/rpmsg-pru interface. Sitting above these core Linux drivers are two PRU-specific client drivers: RemoteProc-PRU and rpmsg-PRU which abstract the generic framework into a unified API for the protocol stack above.

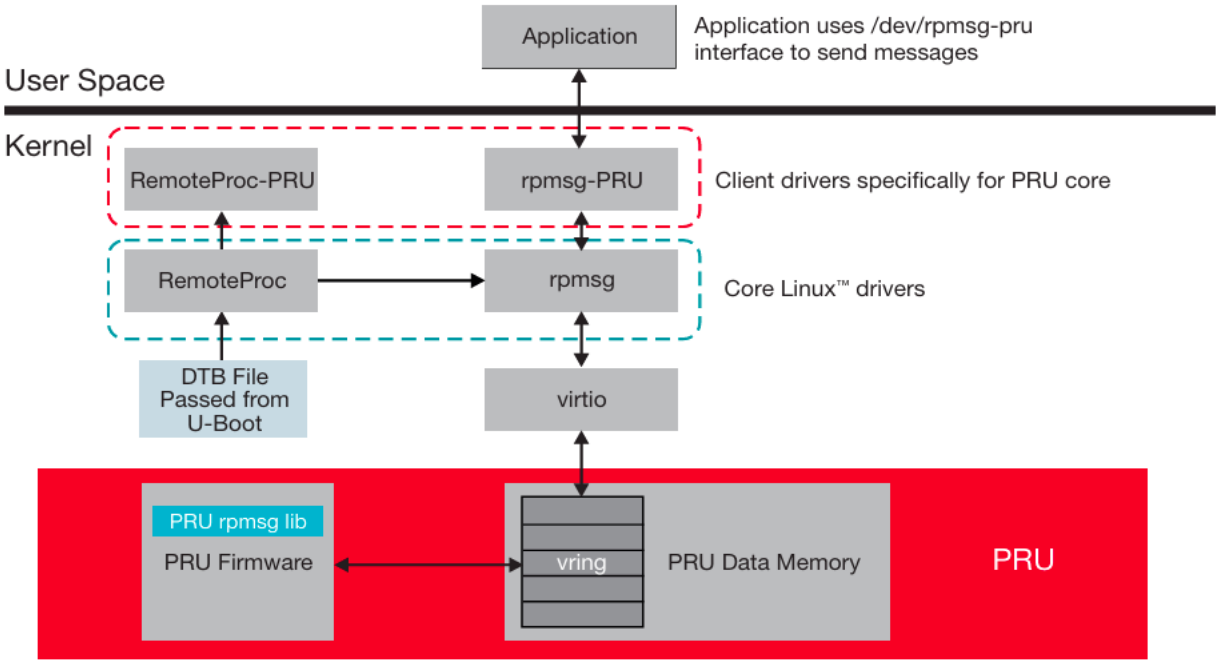

Figure 4-2. Sitara Arm/PRU Software Architecture

Figure 4-3 maps this generic stack to a concrete EtherCAT Device implementation on the AM335x. The PRU Subsystem Driver (API) in Figure 3-3 corresponds to the client driver layer in Figure 3-2: it exposes a unified interface that hides the underlying firmware loading and message passing mechanisms from the layers above. The Protocol Adaptation Layer, EtherCAT Device Stack, and Industrial Application running on the Arm core collectively correspond to the user space application in Figure 4-2, exchanging data with the PRU through the /dev/rpmsg-pru interface. The PRU firmware handles Layer 2 data link functions: frame processing and MII interface control. This layered design allows different industrial protocol firmware to be loaded onto the PRU independently, enabling multi-protocol operation without modifications to the application-layer software.

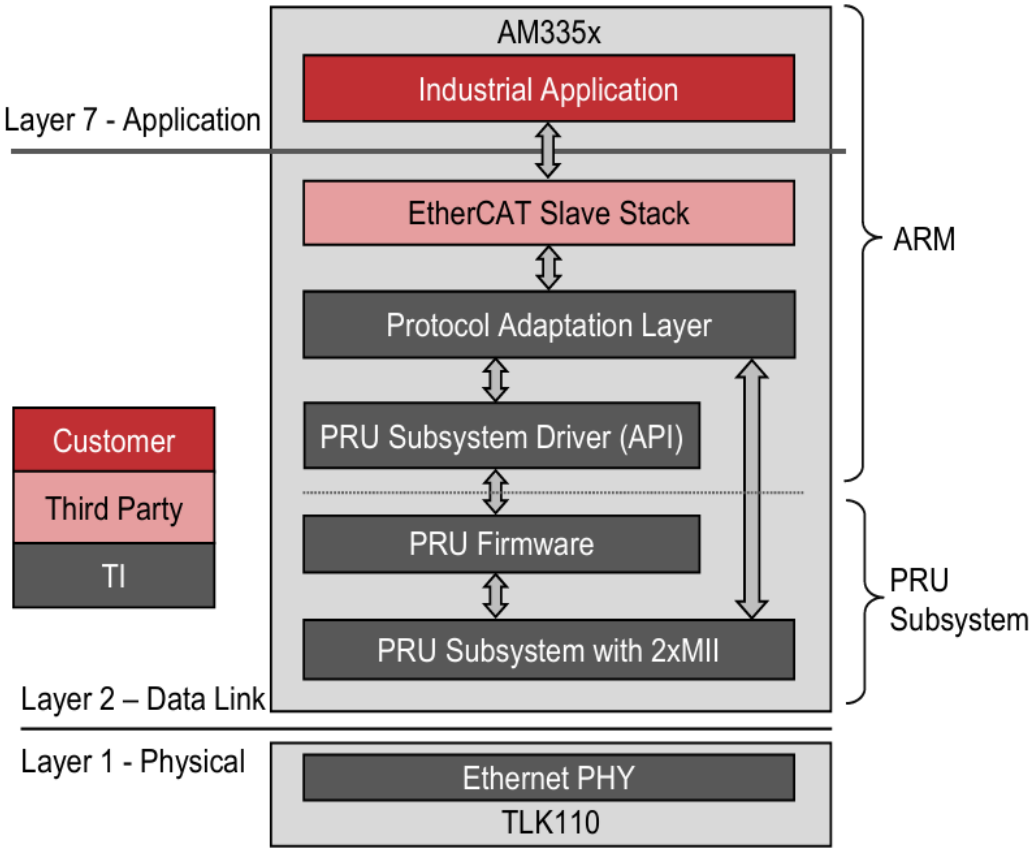

Figure 4-3. EtherCAT Device Software Architecture on PRU-ICSS

4.3 Application Examples

Figure 4-4 shows a representative industrial automation topology built on TI Sitara processors. The AM62Px/AM62x serves as the top-level PROFINET RT Controller running ladder logic while simultaneously driving a 1080p HMI display. The AM64x acts as both a PROFINET RT Device and an EtherCAT Controller, commanding downstream AM243x-based VFD nodes over EtherCAT, with a PCIe Wi-Fi module providing wireless connectivity. Each node loads its respective PRU firmware independently, enabling PROFINET, EtherCAT, and IO-Link to coexist across the network without application-layer modifications.

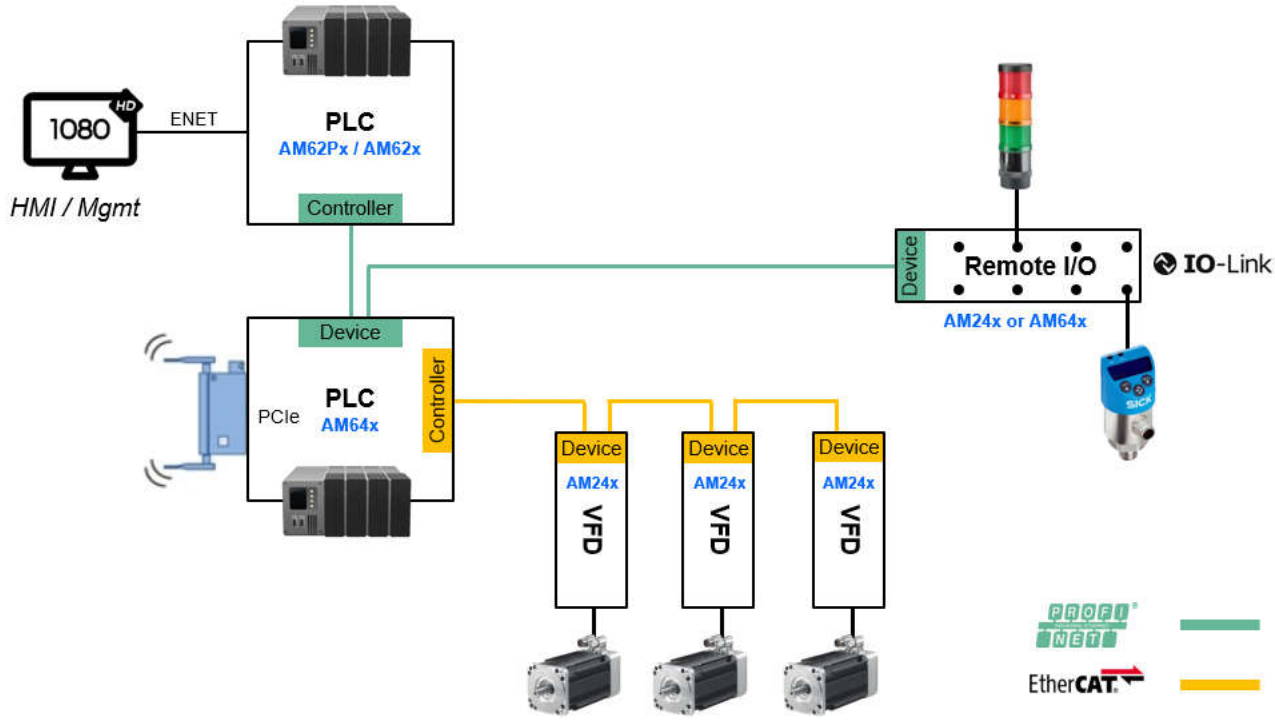

Figure 4-4. TI Sitara Processors Multi-Protocol Industrial Communication System

5 Industrial Protocol Support Matrix

Table 5-1. Industrial Protocol Support Matrix

Feature	AM62x	AM261	AM263	AM243	AM64x
Subsystem	1x PRUSS	2x PRU-ICSS	1x PRU-ICSS	2x PRU_ICSSG	2x PRU_ICSSG
R5F/A Cores	4x A53	2x R5F	4x R5F	4x R5F	4x R5F + 2x A53
Processing Capacity	16.8k DMIPS	2K DMIPS	3.2K DMIPS	6.4k DMIPS	12.4k DMIPS
Controller Support	Yes	Yes	Yes	Yes	Yes
Device Support	No	Yes	Yes	Yes	Yes
Multi-Protocol Support	No	Yes	Yes	Yes	Yes
Gigabit Ethernet Ports	2	2	2	Up to 5	Up to 5
PCIe/USB3.0	2x USB2	1x USB2	No	1x lane/USB3	1x lane/USB3

6 Summary

This document describes the PRU-ICSS architecture, the technical mechanisms of EtherCAT, PROFINET, and EtherNet/IP, and multi-protocol design examples on TI Sitara processors. Three generations of PRU subsystems, PRUSS, PRU-ICSS, and PRU_ICSSG, offer increasing feature sets, with PRU_ICSSG being the highest-end version. Each PRU core runs with single-cycle, non-pipelined execution and loads protocol-specific firmware independently, implementing Layer 2 functions for EtherCAT, PROFINET, and EtherNet/IP; this layered design enables multi-protocol operation without modifications to the application-layer software. A representative system example demonstrates AM62x, AM64x, and AM243x devices coexisting within a single industrial network, each running its designated protocol firmware. The protocol support matrix further confirms multi-protocol capability across AM261x, AM263x, AM243x, and AM64x.

7 References

1. Texas Instruments, PRU Subsystem Features Comparison, application note
2. Texas Instruments, Introductory PRU Guide, seismic
3. Texas Instruments, Ensuring real-time predictability: Leveraging TI's Sitara™ processors programmable real-time unit, marketing white paper
4. Texas Instruments, AM261x Sitara™ Microcontrollers, datasheet
5. Texas Instruments, AM263x Sitara™ Microcontrollers with Real-Time Control, datasheet
6. Texas Instruments, Industrial Communication Protocols Supported on TI Processors and MCUs, application note
7. Texas Instruments, Multi-protocol industrial communication: empowering next-generation industrial automation, webinar.
8. Texas Instruments, EtherCAT® on Sitara™ Processors , marketing white paper.
9. Texas Instruments, PROFINET® on TI's Sitara™ processors, marketing white paper.
10. Texas Instruments, EtherNet/IP™ on TI's Sitara™ processors, marketing white paper.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025