Lattice CertusPro-NX PROFINET Stack

Reference Design

FPGA-RD-02303-1.0

January 2025

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
AHBL	Advanced High-performance Bus-Lite
ADI	Analog Devices
APB	Advanced Peripheral Bus
API	Application Programming Interface
AXI	Advanced Extensible Interface
CPU	Central Processing Unit
DCP	Data Consistency Protocol
DMA	Direct Memory Access
FIFO	First-In-First-Out
FPGA	Field Programmable Gate Array
FreeRTOS	Free Real Time Operating System
GPIO	General Purpose Input Output
GSRD	Golden Software Reference Design
ICMP	Internet Control Protocol Message
IPK	Lattice IP Package
ISR	Interrupt Service Routines
LPDDR	Low Power Double Data Rate Generation 4
LPDDR4	Low Power Double Data Rate Generation 4
LwIP	Light weight Internet Protocol
MCB	Motion Control Board
MPMC	Multi-Port Memory Controller
PHY	Physical Network Interface
PLC	Programmable Logic Controller
PNIO	PROFINET I/O
PNIO-CM	PROFINET I/O Configuration Manager
PNIO-PS	PROFINET I/O Protocol Stack
PROFINET	Process Field Network
QSPI	Quad Serial Peripheral Interface
RISC-V	Reduced Instruction Set Computer-V
RTL	Register-Transfer Level
SGDMA	Scatter-Gather Direct Memory Access
TIA	Totally Integrated Automation
TSEMAC	Tri-Speed Ethernet Media Access Controller
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Data gram Protocol

1. Introduction

Process Field Network (PROFINET) is an industrial Ethernet standard that facilitates real-time data exchange between controllers such as PLCs and devices such as sensors and actuators for precise control in automation. It supports various network topologies and offers safety, energy management, and IT integration features. The PROFINET architecture is scalable and allows easy system expansion, making it ideal for diverse industrial applications.

Field-Programmable Gate Array (FPGA) devices enable PROFINET by providing the necessary hardware acceleration for real-time data processing. FPGA devices handle high-speed deterministic data exchanges to ensure minimal latency and precise timing. FPGA devices can be configured to implement various PROFINET protocols. Additionally, FPGAs can integrate with standard Ethernet controllers, supporting various network topologies and ensuring seamless communication between devices. The reconfigurable nature of FPGAs makes it easier to scale and adapt PROFINET systems to different industrial applications.

This reference design demonstrates an implementation of a PROFINET subsystem implemented on a Lattice CertusPro™-NX FPGA device.

1.1. Quick Facts

The reference design files are packaged together with the PROFINET reference design.

Table 1.1. Summary of the Reference Design

General	Target Devices	LFCPNX-100
	Source code format	C Language
Simulation	Functional simulation	Not performed
	Timing simulation	Not performed
	Test bench	Not available
	Test bench format	Not available
Software Requirements	Software tool and version	FreeRTOS, Lattice Propel™ 2023.2, Lattice Propel Builder 2023.2
	IP version (if applicable)	• ADIN1200 MII (Available in the design as Verilog) • ADIN1300 RGMII (Available in the design as Verilog) • SGDMA 2.1.1 • TSE MAC 1.5.0 • AXI Interconnect 1.2.2 • RISCV 2.3.0 • AHBL Interconnect 1.2.0 • LPDDR4 2.1.0 • GPIO 1.6.2 • QSPI 1.1.1 • MPMC 1.0.0
Hardware Requirements	Board	ADI Motion Control Board (MCB)
	Cable	Lattice Programming Cable, Cat 6 LAN Cable

1.2. Features

Key features of the PROFINET reference design include:

- PROFINET (RT-Labs p-net stack) functionality on ADI MCB with Lattice CertusPro-NX device GSRD
- Ethernet communication with ICMP packet response and reply
- Supports 100 Mbps Ethernet communication with an ADIN1200 module

For more information on the GSRD features, refer to the Golden System Reference Design Demo User Guide (FPGA-UG-02205).

For more information on the ADI MCB, refer to ADI MCB system.

For more information on the PROFINET p-net stack, refer to RT-Labs PROFINET Stack.

For more information on the ADIN1200 module, refer to ADIN1200.

1.3. PROFINET Overview

PROFINET is a modern industrial communication protocol designed to replace traditional fieldbus systems in factory automation. Fieldbuses have been crucial for industrial communication, but PROFINET, built on the IEEE 802.3 Ethernet standard, provides significant advantages such as higher data speeds, better scalability, and seamless integration with IT systems. By using Ethernet, PROFINET can leverage the familiar infrastructure while adding industrial-grade reliability and determinism for real-time applications.

In terms of protocols, PROFINET supports both standard IP-based communication (TCP/IP and UDP/IP) for non-time-critical data and a specific EtherType (0x8892) for real-time communication, which bypasses standard IP layers to achieve lower latency. This combination allows PROFINET to be highly flexible, supporting both general network traffic and high-performance real-time data exchange in the same network.

The adoption of PROFINET is widespread across industries such as automotive, manufacturing, process industries, and energy, due to its robust performance, flexibility, and compatibility with Industry 4.0 standards. The protocol has become a popular choice for industrial automation due to its support for complex, scalable systems while maintaining high-speed, reliable communication needed for modern industrial processes.

1.3.1. PROFINET Real-Time Communication and Cyclic Data Exchange

In PROFINET, real-time communication is essential for the deterministic behavior required in industrial automation, such as controlling machines and processes. To achieve this, PROFINET uses specialized mechanisms to set up real-time connections and handle cyclic data exchange with minimal delays.

1.3.2. Establishing a Real-Time Connection

When a PROFINET system is initialized, the I/O Controller (typically a PLC or industrial PC) configures communication with the I/O devices (sensors and actuators) based on the application requirements. This process involves the following steps:

- Device Discovery and Assignment – The I/O Controller scans the network for connected I/O devices. Each I/O device has a unique Device Name and IP address assigned, which helps the controller to identify and communicate with it.
- Communication Setup – After identifying the devices, the I/O Controller configures the I/O devices by setting up cyclic data exchange intervals. The communication parameters, such as update rates, priority, and real-time class, are defined during this step.
- Real-Time Channel Creation – PROFINET uses a specific EtherType (0x8892) to set up a real-time (RT) channel.

1.3.3. Cyclic Exchange of Real-Time Data

Once the connection is established, the cyclic data exchange begins. This process involves the continuous transmission of real-time process data between the I/O Controller and I/O devices at predefined intervals.

- Cyclic Data Frames – PROFINET employs periodic data frames for sending and receiving real-time information. Each device process data (such as sensor readings or actuator commands) is exchanged within these frames.
- PROFINET RT Real-Time Class – This class ensures that real-time communication takes place within the standard Ethernet structure by reducing the delays caused by IP layer processing. Cycle times typically range from 1 to 10 ms, making RT suitable for standard automation tasks

2. Directory Structure and Files

Figure 2.1 shows the directory structure.

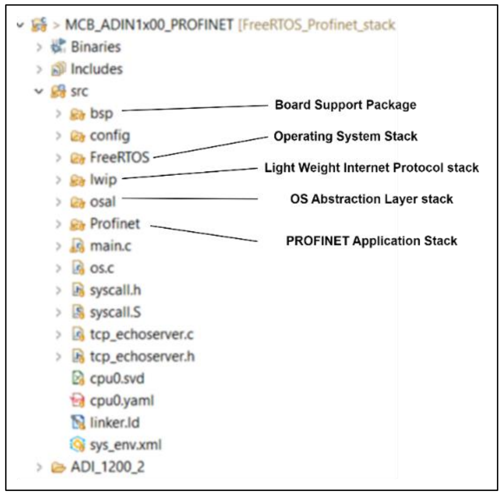

Figure 2.1. PROFINET Stack on FreeRTOS Directory Structure

For the FPGA GSRD stack, refer to the Golden System Reference Design Demo User Guide (FPGA-UG-02205).

The ADIN1200 mii1, 1200_mii2 or ADIN1300 IP is available in the design release folder below. Except for the IP mentioned above, the rest are similar with the GSRD design published for 2023.2.

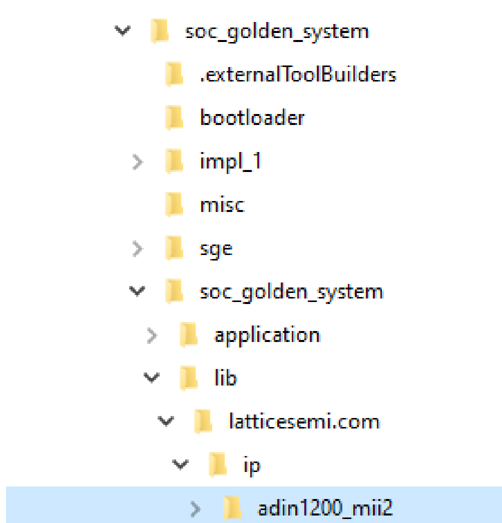

Figure 2.2. ADINI200 PHY Verilog File Location

Apart from the GSRD for each different ADIN connector, there are different IP supporting it. Table 2.1 shows the list of files included in the reference design package for ADIN1200.

Table 2.1. ADIN1200 File List

Attribute	Description
adin1200_mii1_inst.ipx	This file contains the information on the files associated to the generated IP for ETHERCAT port 1
adin1200_mii2_inst.ipx	This file contains the information on the files associated to the generated IP for ETHERCAT port 2
adin1200_mii1.cfg	This file contains the parameter values used in IP configuration for ETTHERCAT port 1

Attribute	Description
adin1200_mii2.cfg	This file contains the parameter values used in IP configuration for ETTHERCAT port 2
component.xml	Contains the ipxact:component information of the IP.
design.xml	Documents the configuration parameters of the IP in IP-XACT 2014 format.
rtl adin1200_mii1.v	This file provides an example RTL top file that instantiates the module. For ETHERCAT port 1
rtl/adin1200_mii2.v	This file provides an example RTL top file that instantiates the module. For ETHERCAT port 2
rtl/adin1200_mii1_bb.v	This file provides the synthesis closed box for ETHERCAT port 1
rtl/adin1200_mii2_bb.v	This file provides the synthesis closed box for ETHERCAT port 2
misc/adin1200_mii1_tmpl.v misc/adin1200_mii1_tmpl.vhd	These files provide instance templates for ETHERCAT port 1
misc/adin1200_mii2_tmpl.v misc/adin1200_mii2_tmpl.vhd	These files provide instance templates for ETHERCAT port 2

Table 2.2 shows the list of files included in the reference design package for ADIN1300.

Table 2.2. ADIN1300 File List

Attribute	Description
adin1300_mii_inst.ipx	This file contains the information on the files associated to the generated IP.
adin1300_mii.cfg	This file contains the parameter values used in IP configuration.
component.xml	Contains the ipxact:component information of the IP.
design.xml	Documents the configuration parameters of the IP in IP-XACT 2014 format.
rtl adin1300_mii.v	This file provides an example RTL top file that instantiates the module. For ETHERCAT port 1.
rtl/adin1300_mii_bb.v	This file provides the synthesis closed box for ETHERCAT port 1.
misc/adin1300_mii_tmpl.v misc/adin1300_mii_tmpl.vhd	These files provide instance templates for ETHERCAT port 1.

3. Functional Description

3.1. PROFINET Architecture Overview

PROFINET is an industrial Ethernet standard used for communication between automation devices. The FPGA design to demonstrate PROFINET is based on the Lattice Golden System Reference Design with the inclusion of an ADIN1200 Soft IP wrapper, which is ADIN1200_mii2, from the Analog Devices to communicate with the external ADIN1200 PHY module**.**

The ADI MCB board has two other PHY modules available for PROFINET: *ADIN1200* and *ADIN1300*. However, in this document, the reference design use case focuses on the ADIN1200 PHY module. Similar configuration methods discussed in this document can be directly applied to the other two interfaces using its respective wrapper design ADIN1200_mii1 and ADIN1300_rgmii. The specific configuration methods of these interfaces are further explained in the ADIN and TSE MAC Integrated Design section. The choice of the interface is for purpose of demonstration only. The other interface designs provided can be setup similarly based its corresponding requirements and dependencies.

The PROFINET software stack runs on FreeRTOS executed using a soft RISC-V core. Lattice developed BSP drivers facilitate communication between the hardware elements on the FPGA and FreeRTOS software. During boot up, these drivers initialize and configure FPGA peripherals to establish effective coordination with the RISC-V processor. For details on the boot flow, refer to the Golden System Reference Design Demo User Guide (FPGA-UG-02205).

The ADIN1200_mii2 and ADIN1200_mii1 wrappers allow PROFINET to perform communication over Lattice Triple Speed Ethernet (TSE) MAC Soft IP with the ADIN1200 PHY module. This applies to the ADIN1300 and ADIN1300 PHY module as well.

The SGDMA IP transfers data between the system LPDDR4 memory and the ADIN1200 TSE interface.

The PROFINET software stack execution in this reference design is built on top of the lwIP (lightweight IP) stack. The lwIP Stack is an open-source TCP/IP stack designed for embedded systems, focusing on reducing memory and processing requirements. The lwIP stack sends and retrieves network data using the SGDMA IP firmware API and diverts all PROFINET related protocol packets to the PROFINET software stack.

The QSPI controller is required to copy the PROFINET software stack binary from QSPI flash to the LPPDR4 memory where the software stack is executed from. System memory is used for bootloader operations.

Figure 3.2 shows the FPGA design used to run the PROFINET software stack on an LFCPNX-100 device with an ADIN IP variant connected to the ADIN PHY module.

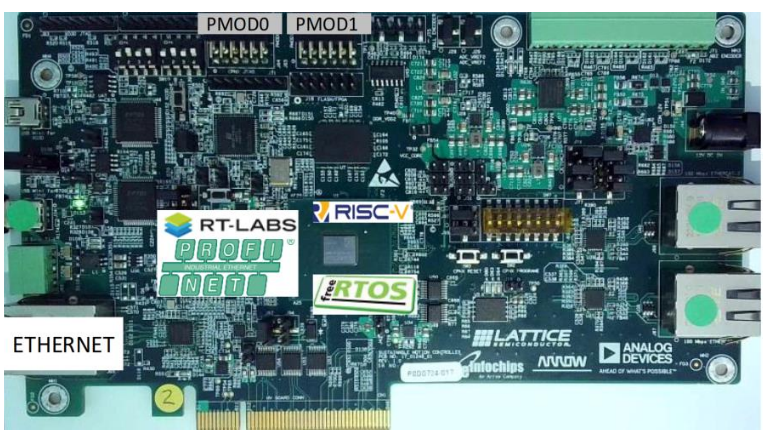

Figure 3.1. ADI MCB Evaluation Board

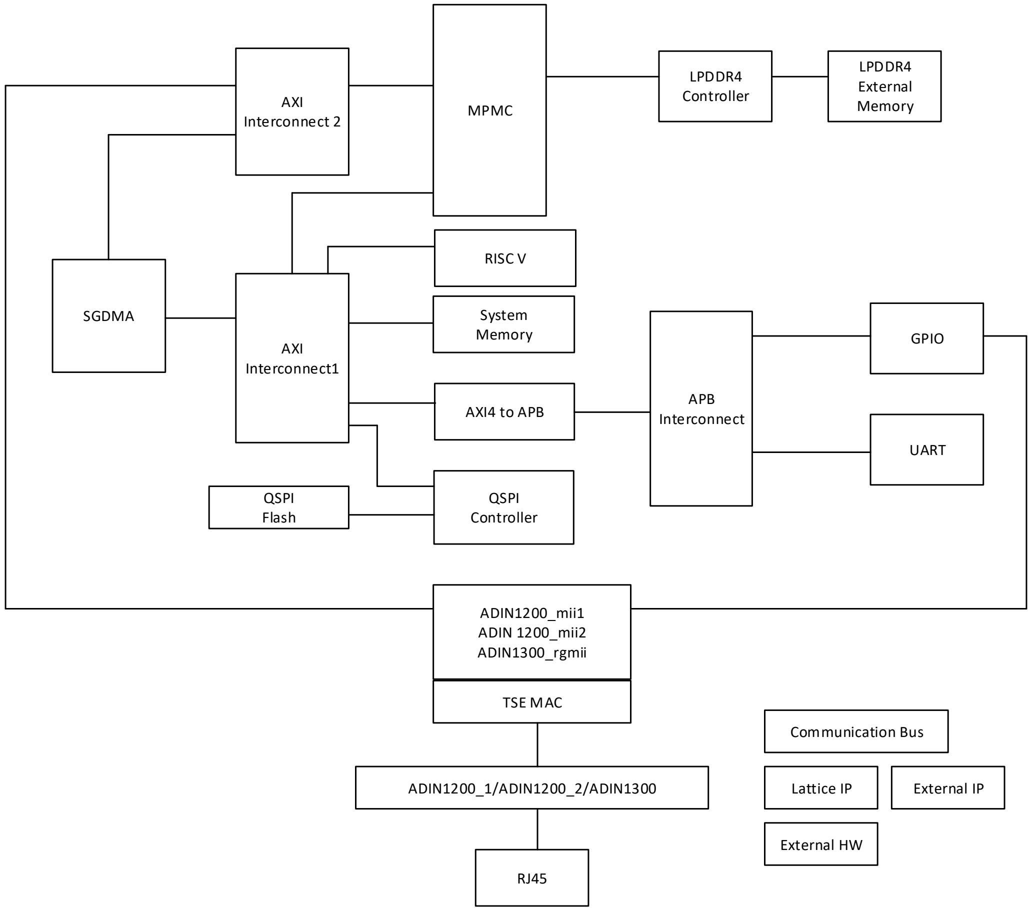

Figure 3.2. PROFINET Stack with ADIN PHY Solution Based on Lattice GSRD

3.2. PROFINET System Components

The GSRD architecture includes the following IP blocks and hardware connections in the PROFINET design.

3.2.1. Soft IP Data Transmission Blocks

- AXI Interconnect 2 – Manages the communication between different IP blocks within the system.
- APB Interconnect – Facilitates communication between the Advanced Peripheral Bus (APB) and other system components.
- MPMC – Controls access to multiple memory ports.
- AXI4 to Memory Map Bridge – Converts AXI4 protocol transactions to memory-mapped transactions.

3.2.2. Soft IP Control Blocks

- SGDMA – Manages data transfer between memory and peripherals.
- RISCV V – CPU register access for system configuration and control.
- LPDDR4 Controller – Manages the interface and communication with LPDDR4 memory.
- GPIO – Interfaces for general-purpose digital signals.
- UART – Serial communication interface.
- ADIN1200_mii1/ADIN1200_mii2/ ADIN1300_rgmii – Non-Lattice Ethernet physical layer transceiver for network connectivity.
- TSE MAC – Manages Ethernet data transmission and reception.
- System Memory – Stores the boot firmware run-time memory

3.2.3. Hardware Components

- QSPI Flash – Provides non-volatile storage for the system.
- LPDDR4 Memory – Main External Memory
- ADIN1200 PHY/ADIN 1300 PHY module – Ethernet communication device

3.3. PROFINET Stack System Flow

This section describes the boot process and Ethernet data flow from an external source (Siemens TIA).

The PROFINET software binary is stored in the external SPI flash. During boot, the boot loader copies the instruction code from external flash to LPDDR. Boot loader process sets the ISR function pointer to a pre-determined LPDDR4 memory address through the memory controller.

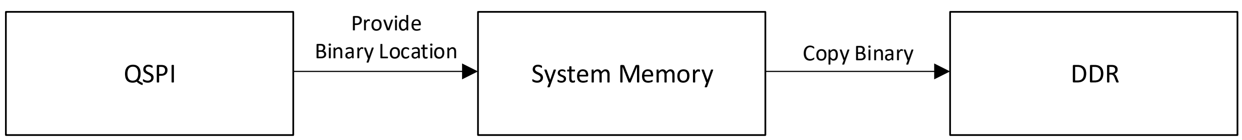

Figure 3.3. PROFINET System Flow Diagram

The SGDMA IP transfers incoming AXI streams from the TSE MAC IP to the LPDDR4 through AXI interconnect2 as shown in Figure 3.4. When data transfer is complete for an Ethernet packet transaction, SGDMA sets the transaction complete bit within the Buffer Descriptor block located in the LPDDR4 memory.

The MPMC IP arbitrates between multiple sources to route data to the LPDDR4 memory controller.

The PROFINET software stack running on the RISC-V module periodically polls the SGDMA buffer through AXI interconnect1 to determine if there is a network packet to be processed.

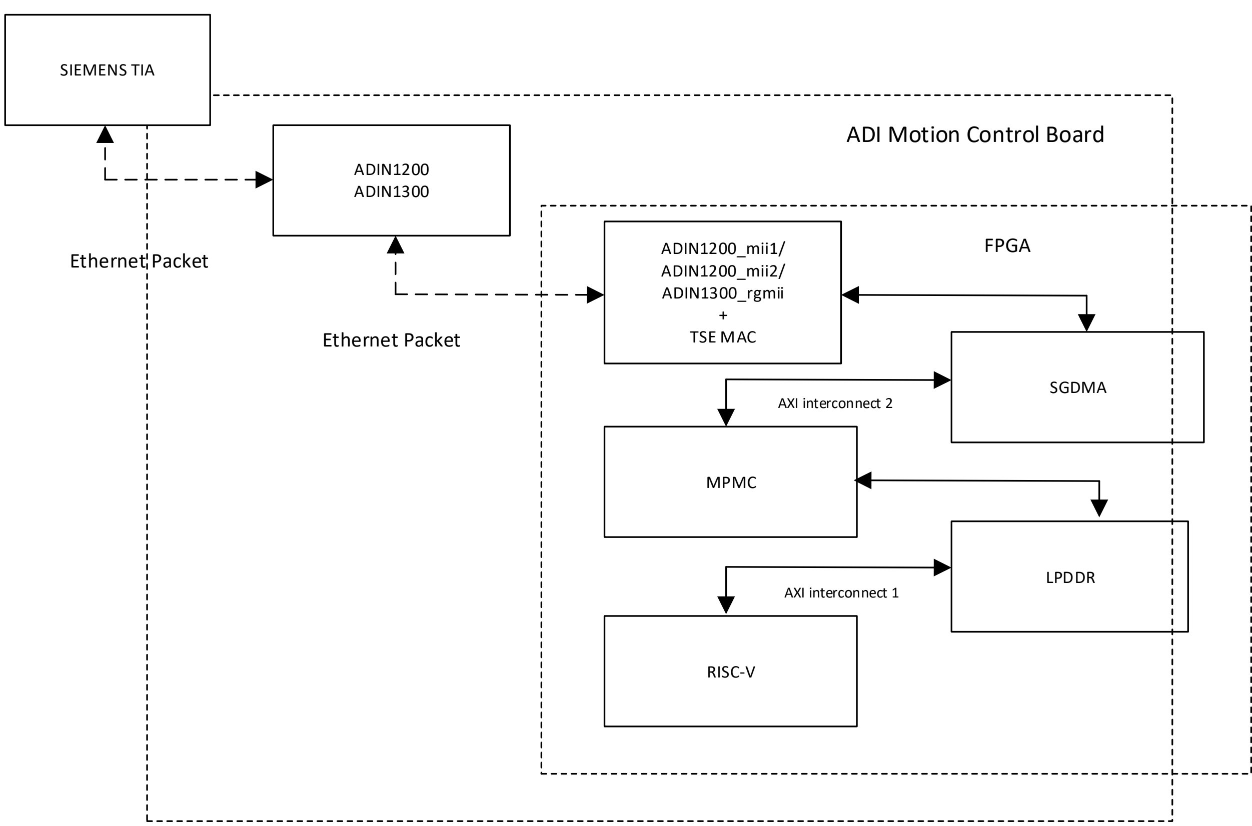

Figure 3.4. ADI Motion Control Block

3.3.1. PROFINET System Memory Map

The PROFINET system is based on the Lattice GSRD implementation. Table 3.1 shows the memory location of each of the IP blocks accessible by the OS.

Table 3.1. PROFINET Stack Memory Map

Base Address	End Address	Range (Bytes)	Range (Bytes in hex)	Size (KB)	Block
00300000	0037FFFF	512000	80000	512	SPI FLASH CONTROLLER
00000000	0003FFFF	256000	40000	256	CPU Data Ram
10000000	10000FFF	4096	1000	4	GPIO
10001000	10004FFF	16384	4000	16	TSE MAC
10090000	10090FFF	4096	1000	4	UART
10091000	10091FFF	4096	1000	4	GPIO RST IRQ
10092000	10092FFF	4096	1000	4	LPDDR4 APB
10093000	10093FFF	4096	1000	4	SGDMA
1009B000	1009BFFF	4096	1000	4	FPGA CONFIG APB
10110000	5010FFFF	1073741824	40000000	1048576	LPDDR4 AXI
F2000000	F20FFFFF	1048576	100000	1024	CLINT (CPU)
FC000000	FC3FFFFF	4194304	400000	4096	PLIC (CPU)
F0000400	FFFFFFFF	262144000	FA00000	256000	RESERVED (CPU)

4. PROFINET Reference Design Parameter Description

This section describes the IP configuration changes and software components added to the Lattice GSRD to enable the PROFINET software stack. Using GSRD as the base design, the configuration changes for SGDMA IP, such as buffer depth, are necessary for system performance optimization. The TSE MAC and PHY are also changed to support interfacing with the ADIN1200 PHY module.

Further information on PROFINET software stack is found in RT-Labs PROFINET Stack. The subsequent sections in the document describe in detail the changes made to the GSRD.

4.1. Scatter Gather DMA IP for PROFINET Stack

Ethernet communication for PROFINET stack requires the SGDMA IP to move data from TSE MAC to and from LPDDR4 through AXI Interconnect2, as shown in Figure 4.1. The FIFO depth of the SGDMA IP block is configured to be 4096 bytes. The PROFINET stack data processing utilizes lwIP stack as the network driver for FreeRTOS.

The lwIP stack configures the SGDMA IP parameters upon power up through the SGDMA driver API. The lwIP stack also allocates in LPDDR4 memory, transmit and receive buffers, as well as buffer descriptors for SGDMA operations. Once configuration for SGDMA operation has completed, the lwIP stack continuously polls data from the SGDMA buffer descriptor field on LPDDR4 to determine the readiness of receive or transmit data. When either receive or transmit data is available for transfer, the lwIP stack retrieves the Buffer Descriptor Address Pointer from LPDDR4, which identifies where the buffered data are located.

When receiving packet from TSE MAC, the transfer operation from SDGMA to LPDDR4 is initiated by lwIP stack (see the *Single Buffer Descriptor per Request* section of the SGDMA Controller IP Core User Guide (FPGA-IPUG-02131) document) by triggering the SGDMA streaming request. Once the data streaming is completed, the Descriptor Status Complete bit in the Buffer Descriptor is set to 1 (Bit 31 in the Buffer Descriptor, in the *S2MM_STATUS (Offset 0x0C)* section, of the SGDMA Controller IP Core User Guide (FPGA-IPUG-02131)). The lwIP stack copies the data from the Buffer Descriptor and push it upstream to the PROFINET stack for processing.

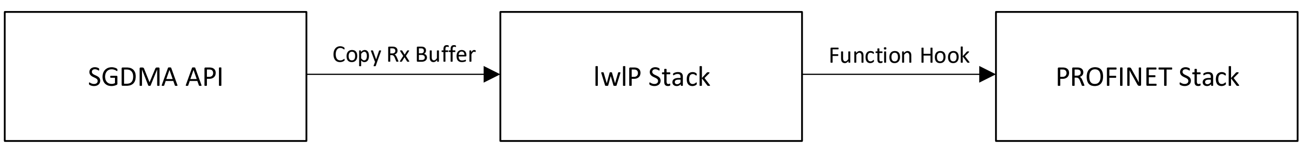

Figure 4.1. SGDMA Driver API Receive

When transmitting packet to TSE MAC, the lwIP stack copies the data from the PROFINET stack to the transmit buffer through a callback function. The transfer operation to SDGMA from LPDDR4 is initiated by lwIP stack (see the *Single Buffer Descriptor per Request* section of the SGDMA Controller IP Core User Guide (FPGA-IPUG-02131) document) by triggering the SGDMA streaming request. The SGDMA driver API streams the data out to ADIN1200_mii2 directly.

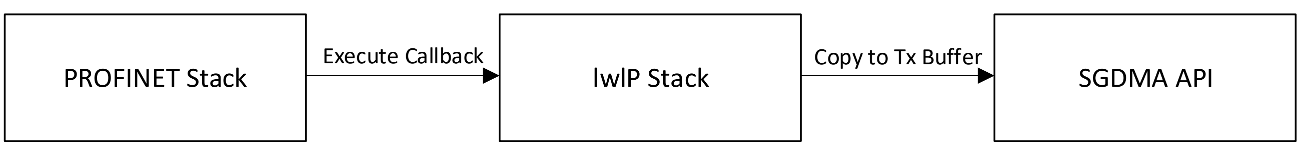

Figure 4.2. SGDMA Driver API Transmit

For more details, refer to the SGDMA Controller IP Core User Guide (FPGA-IPUG-02131).

4.2. ADIN and TSE MAC Integrated Design

The ADI Motion Control Board (MCB) includes three ports with two ADIN1200 PHY module and a single ADIN1300 PHY module. The ports are ADIN1200 for port 1, ADIN1200 for port 2, and ADIN1300 port. The ADIN1200 ports are labelled as ETHERCAT-1 and ETHERCAT-2 ports on the board as shown in Figure 4.3 and ADIN1300 port is shown as Ethernet port in Figure 4.4. Depending on which ADIN1200 port is selected, the corresponding FPGA bit stream file and software configuration binary to be used is described in the PROFINET Application Description section. In this reference design, the ADIN1200 port 2 is used.

Figure 4.3. ADIN1200 Ports Located on the ADI MCB

Figure 4.4. ADIN1300 Ports Located on the ADI MCB

The ADIN1200 port selection on the ADI MCB board is configured in the GPIO setting. Depending on which port is chosen, the FreeRTOS BSP driver function is used to configure GPIO RST IRQ Address offset 7 for ADIN1200 port 1 (ETHERCAT-1) and GPIO RST IRQ Address offset 6 for ADIN1200 port 2 (ETHERCAT-2) accordingly.

For ADIN1300 to work on PROFINET some changes are required on the FreeRTOS:

- Change the PHY Register Control bit 9 to 0 to disable 1 Gbps advertising
- Change the PHY Register for Auto Negotiation to value 0x1e1
- Configure GPIO RST IRQ Address offset to 1 upon power up

The lwIP stack configures the IEEE 802.3 Clause 22 registers in the ADIN1200 PHY module and ADIN1300 PHY module through the individual interface driver APIs. The designation of these registers is found in the related documentation for the individual PHY modules

For more details on the ADIN1200 module, refer to the ADIN1200 page.

For more details on the ADIN1300 module, refer to the ADIN1300 page.

For more information on the ADI MCB, refer to the ADI MCB system page.

For more information on the TSEMAC IP, refer to the Tri-Speed Ethernet MAC IP User Guide (FPGA-IPUG-02084).

5. PROFINET Application Description

5.1. PROFINET Reference Design Build Configuration

To build the FreeRTOS PROFINET reference design, select the appropriate ADIN interface. In the example shown in Figure 5.1, ADIN1200_2 is selected.

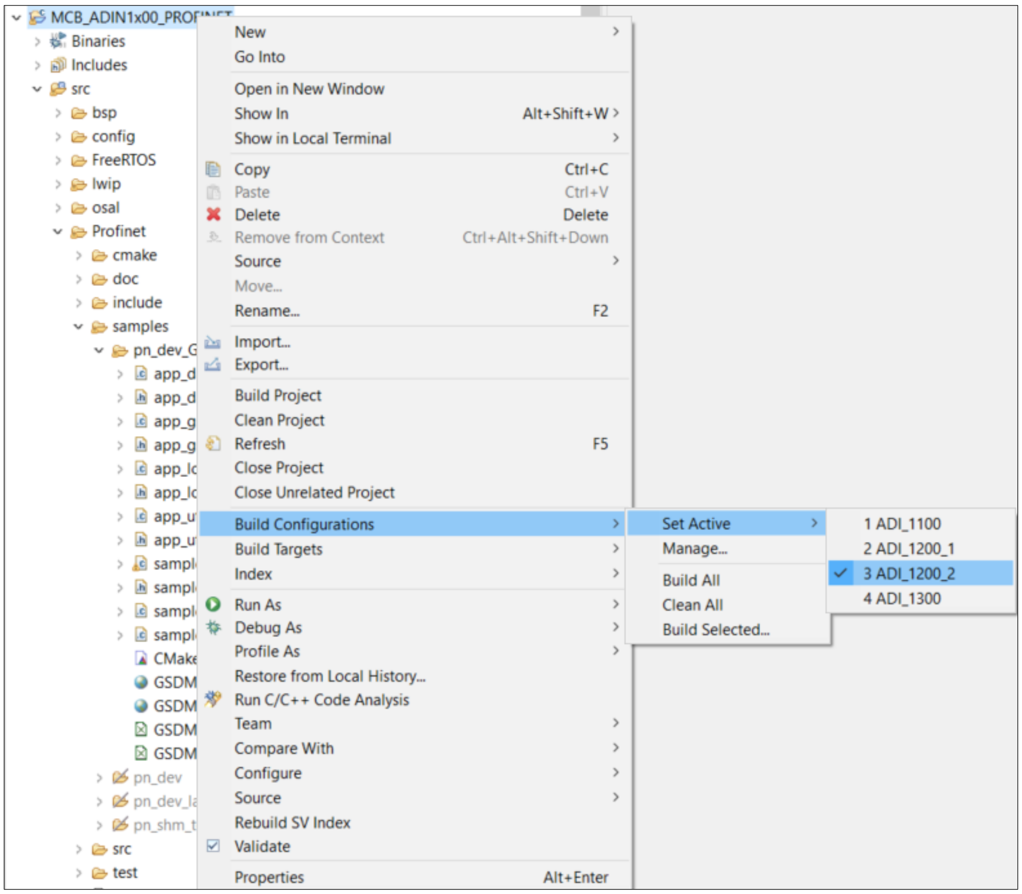

Figure 5.1. Build Configuration for PROFINET FreeRTOS Reference Design

5.2. PROFINET Software Stack Functional Description

5.2.1. lwIP Stack Functions

The Ethernet and Internet Protocol stacks used in the PROFINET reference design is developed using the lwIP Stack . The implementation is ported into the FreeRTOS framework. The lwIP stack has function hooks that are adapted to the PROFINET network function

The current network address is set to 192.168.0.50. You can change the IP address in *ethernet_config.h* and recompile to change the network address.

5.2.2. ADIN1200 and ADIN1300 Driver API Functions

ADIN1200 and ADIN1300 driver performs several key functions.

5.2.2.1. ADIN_init()

This API is used to identify the PHY interface for the ADIN1200 or ADIN1300 module.

5.2.2.2. phy_write()

```
phy_write (tsemac_handle_t *handle,unsigned int phy_LPDDR,unsigned int reg_offset,unsigned int data)
```

This API writes the respective ADIN module parameters for IEEE clause 22 registers. Refer to ADIN1200 or ADIN1300 for more details.

5.2.2.3. phy_read()

```
phy_read(tsemac_handle_t *handle,unsigned int phy_addr, unsigned int reg_offset)
```

This API reads the respective ADIN module parameters for IEEE clause 22 registers. Refer to ADIN1200 or ADIN1300 for more details.

5.2.2.4. adin_GetLinkStatus()

```
adin_GetLinkStatus(adinPhy_DeviceHandle_t hDevice, adi_phy_LinkStatus_e *status)
```

This API detects thelink status of the respective ADIN module. This function is used in the lwIP stack low level Ethernet tasks that run continuously on the FreeRTOS.

5.2.3. SGDMA Driver API functions

The functions that are required to implement SGDMA controls are based on SGDMA Controller IP Core User Guide (FPGA-IPUG-02131).

5.2.3.1. sgdmaAppInit()

This API initializes the DMA with TX and RX buffers.

5.2.3.2. sgdmaGetRxLength()

This API gets the length of the DMA using the SGDMA driver s2mm_buf_desc_dma() and returns the length of the buffer.

5.2.3.3. sgdmaReadFromMAC()

```
sgdmaReadFromMAC(unsigned int *pBuffer,unsigned short readlen)
```

This API reads from the buffer pointer and copies it into the network buffer to be sent upstream.

5.2.3.4. sgdmaWritetoMAC()

```
sgdmaWritetoMAC(unsigned int *OutputBuffer,  int total_bytes)
```

This API copies the buffer sent from the upstream network stack to be pushed out into the network.

5.2.4. UDP Stack Functions

The UDP stack pushes the UDP communication up the lwIp stack. The UDP receive function stack is shown in Figure 5.2.

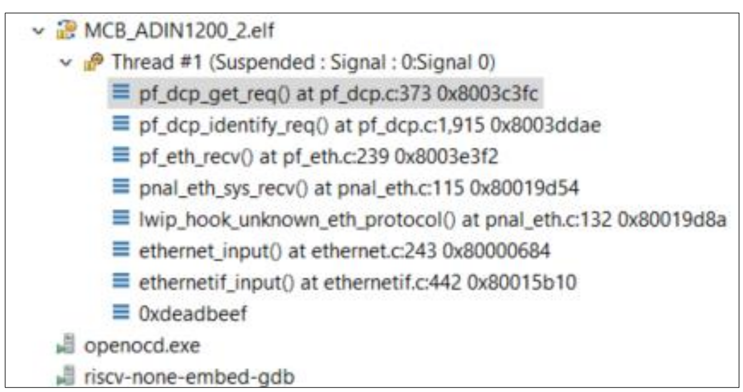

Figure 5.2. UDP Receive Stack

The ethernetif_input() function constantly polls the PROFINET UDP packet buffered in the SGDMA IP block. When it acquires the buffer, the Ethernet pushes the packet from the SGDMA buffer through the lwip_hook_unknown_eth_protocol() function for further upstream processing required by the PROFINET Ethernet application APIs. The UDP transmit function stack is shown below in Figure 5.3.

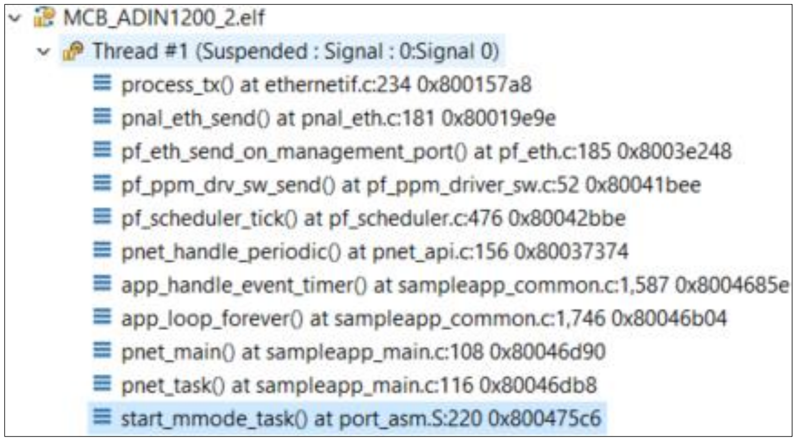

Figure 5.3. UDP Transmit Stack

To transmit data, the PROFINET stack pushes the packet out using the TCP messaging thread in the LWIP stack. The TCP thread sends the packet out by pushing the buffer into the UDP Ethernet stack to be transmited through process_tx().

5.2.5. PROFINET Stack Basic Functions

The PROFINET stack implemented in this reference design is provided by RT-Labs. The stack is located as an open-source repository: https://github.com/rtlabs-com/p-net .

Below are some key points that is required to be configured in the ADI MCB implementation for PROFINET:

- The PROFINET initialization requires a heap memory stack of 256 KB on the DDR RAM
- The default configuration for sample applications is derived from pn_dev as developed by RT-Labs
- The network interfaces are configured by app_utils_pnet_cfg_init_netifs() to ensure that the available interfaces are configured

The functions performed in pnet_init are as follows:

- pf_cpm_init(net)
    - Cyclic Process Management (CPM) is a core concept in the PROFINET protocol that enables the synchronized execution of tasks across a network of devices. It provides a deterministic and efficient mechanism for controlling the timing and sequence of operations in industrial automation systems.
- pf_ppm_init(net)
    - Process Parameter Mapping (PPM) maps process variables between different devices or systems. It involves defining the correspondence between the variables, including their names, data types, units, and other relevant attributes.
- pf_alarm_init(net)
    - Alarms in PROFINET are critical components for monitoring and managing industrial processes. They are used to alert operators or systems of abnormal conditions or potential issues that may affect the safety, efficiency, or quality of operations.
- pf_eth_init(net, p_cfg)
    - Ethernet initialization to enable the Ethernet stack.
- pf_bg_worker_init(net)
    - PROFINET background worker is used to save the configuration during run-time.

- pf_cmina_init(net)
  - Communication Module Interface for Network Access (CMINA) is a fundamental component in the PROFINET protocol that provides a standardized interface for connecting communication modules to the network. It defines the physical and electrical characteristics, as well as the communication protocols, required for modules to interact with PROFINET devices.
- pf_dcp_exit(net) and pf_dcp_init(net)
  - Diagnostic Communication Profile (DCP) is a standardized communication profile within the PROFINET protocol that provides a framework for exchanging diagnostic information between devices on the network. It enables devices to monitor their own health, detect faults, and provide detailed diagnostic data to help troubleshoot and maintain the system.
- pf_port_init(net)
  - PROFINET port initialization and interface configuration
- pf_lldp_init(net)
  - Link Layer Discovery Protocol (LLDP) is a network protocol used in PROFINET to discover and exchange information about neighboring devices on the network. It provides a mechanism for devices to automatically configure themselves and establish connections, simplifying network management and troubleshooting.
- pf_pdport_init(net)
  - PROFINET physical device management port initialization.

5.2.5.1. Tasks

There are five events to be handled by the PROFINET task:

- APP_EVENT_READY_FOR_DATA
  - Event to handle application data sent from the PLC
- APP_EVENT_ALARM
  - Alarm events that occur within the stack and to be sent to the PLC
- APP_EVENT_SM_RELEASED
  - Sub module events from the I/O module to be sent to the PLC
- APP_EVENT_ABORT
  - Abort event in the case of error sent from I/O module to the PLC
- APP_EVENT_TIMER
  - Timer event scheduled periodically by the I/O module to be sent to the PLC or consumed internally

5.2.5.2. Application Control

The application control for the PROFINET stack resides in the *sampleapp* framework. The *sampleapp* can be used to toggle five LEDs and receive six inputs based on the GSDML file and the Siemens TIA project settings provided in this reference design.

5.3. Communication Between PLC and PROFINET I/O System

This connection is implemented using the Ethernet interface. Most of the messages should be in ASCII to facilitate debugging using a terminal program on the host system.

5.3.1. Messages from PLC to ADI MCB System

- PNIO-DCP
- LLDP Hello packet
- PNIO-CM
- PNIO-PS
- PMIO

5.3.2. P-DCP

P-DCP stands for PROFINET Data Consistency Protocol. It is a crucial component of the PROFINET protocol that guarantees consistent and reliable transmission of data between devices in a PROFINET network

5.3.3. Link Layer Discovery Protocol (LLDP)

PROFINET devices use LLDP to discover other network devices, such as switches and routers, that are connected to the same physical network.

5.3.4. PNIO-CM

PNIO-CM stands for PROFINET I/O Configuration Manager. It is a crucial component of the PROFINET protocol that plays a vital role in configuring and managing I/O devices in a PROFINET network.

5.3.5. PNIO-PS

PNIO-PS stands for PROFINET I/O Protocol Stack. It is the underlying communication protocol that governs the exchange of data between I/O devices and controllers in a PROFINET network.

5.3.6. PNIO

PNIO stands for PROFINET I/O. It is the core protocol responsible for communication between I/O devices, such as sensors and actuators, and a central controller in a PROFINET network.

For more information on PROFINET protocols, refer to the PROFINET Specification page.

6. Running the Reference Design

6.1. Compiling the Reference Design C Code in the Propel Software

Refer to the Lattice Propel Training for the basic Propel compilation steps.

The C code for the running firmware must be compiled based on the ADIN interface used for the project, which is slightly different from the GSRD.

The active build selection is either ADIN1300, ADIN1200_2, or ADIN1200_1. Examples shown in this document are based on ADIN1200_2. For the bootloader firmware, the active build selection must be *Boot*.

The build artifact location is shown in Figure 6.1.

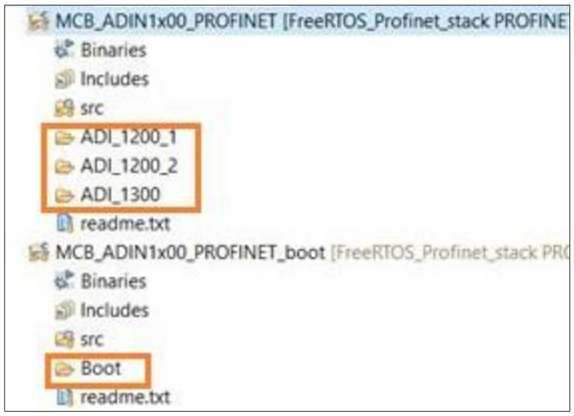

Figure 6.1. PROFINET Build Artifact Location in Propel

6.2. Generating the Bitstream File in the Lattice Radiant™ Software

To generate the bitstream file, refer to Golden System Reference Design User Guide (FPGA-RD-02283).

7. Testing the Application with PROFINET Demo Setup

7.1. Demo Overview

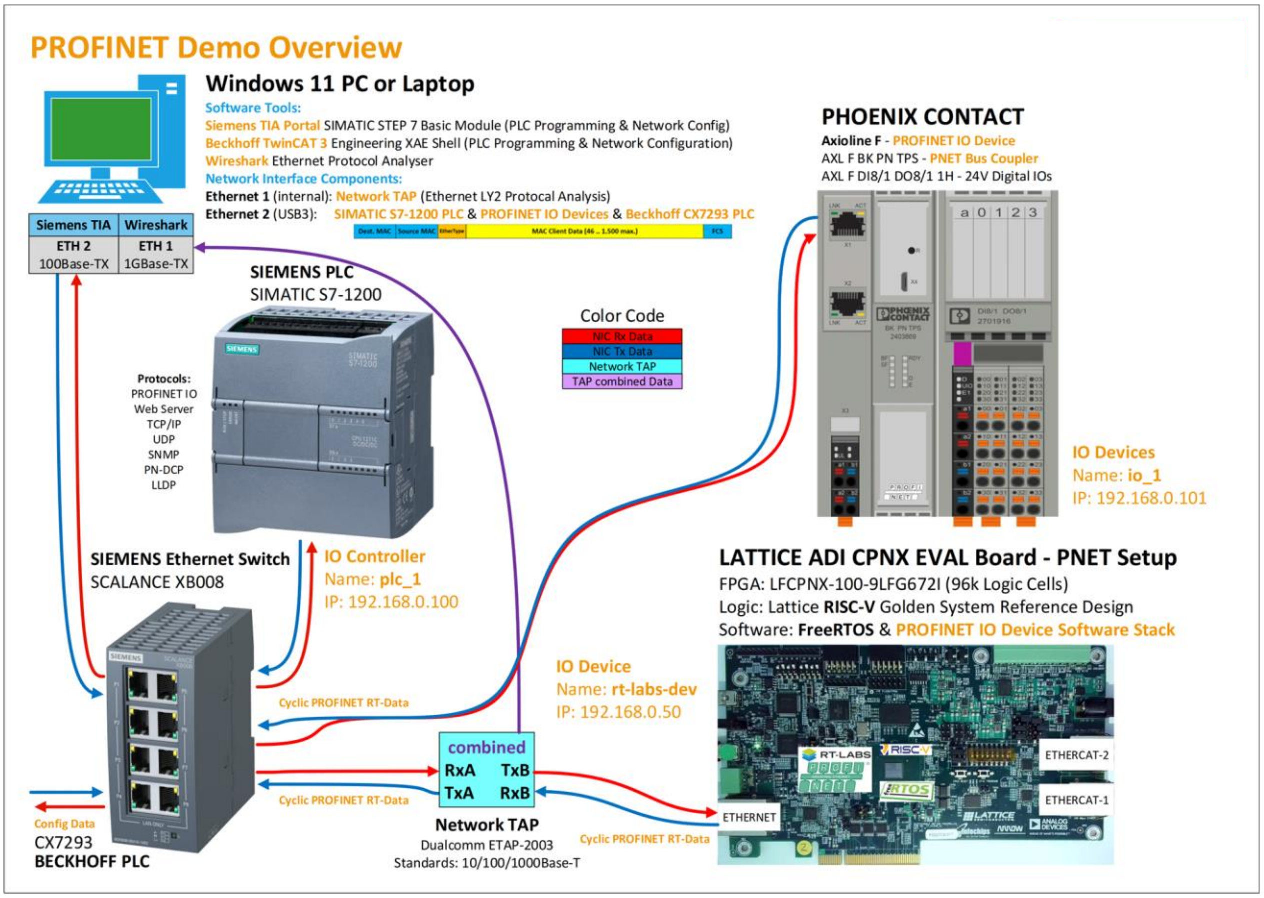

Figure 7.1. PROFINET Demo Setup Overview

7.1.1. Hardware Components

7.1.1.1. Windows 11 PC or Laptop

- Ethernet Requirements
  - Option 1: Two RJ45 Ports (10/100/1000BASE-T)
  - Option 2: One RJ45 Port (most Laptops today) and additional external USB-Ethernet Adapter
- USB-Ethernet Adapter (optional for Laptops with only one Ethernet Port)
  - For example, Amazon Basics USB3 Gigabit Ethernet Adapter
    - Standards: 10/100/1000BASE-T
    - PHY Chip: ASIX AX88179

Figure 7.2 shows the overview of the PROFINET Devices: PLC, Ethernet Switch and I/O Device.

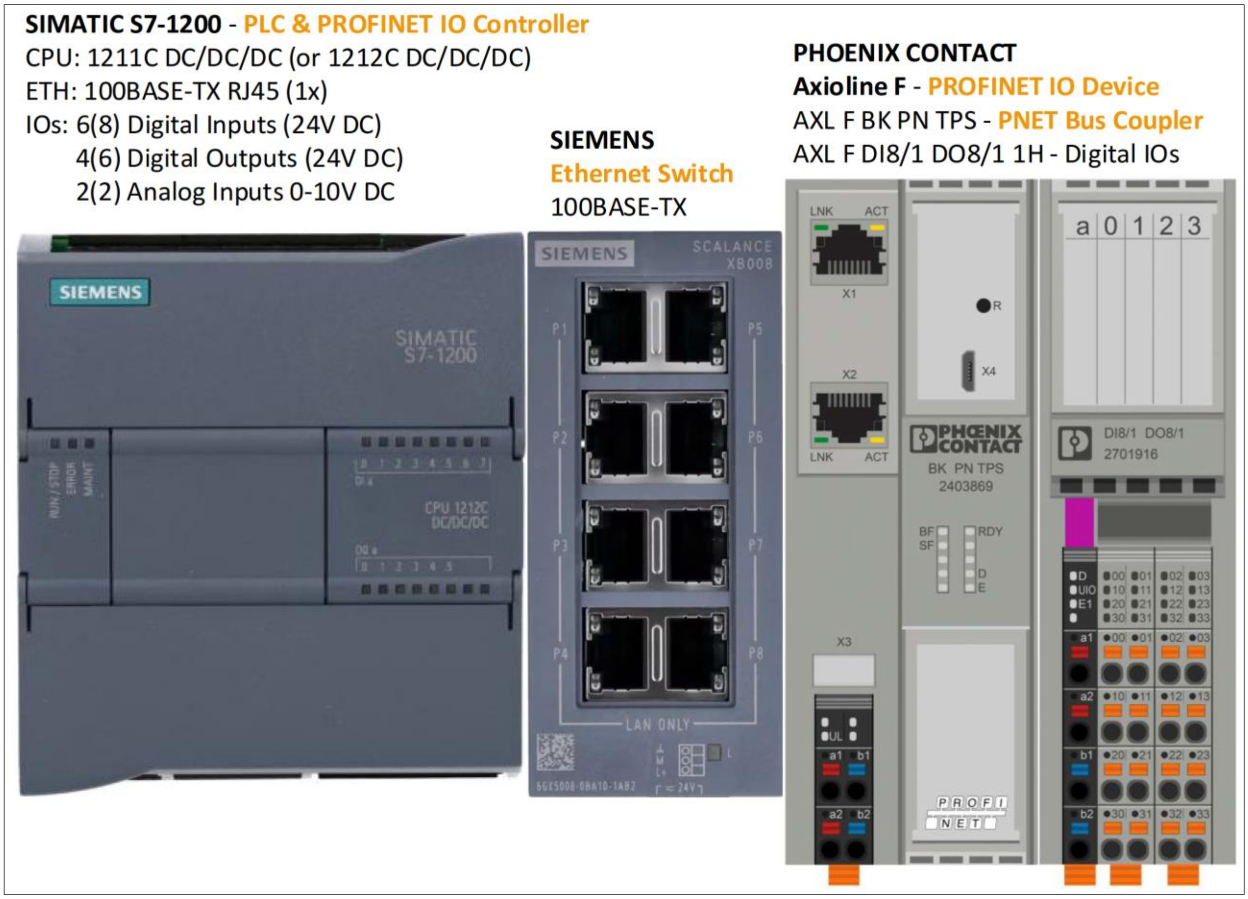

Figure 7.2. PROFINET Demo Hardware Setup

7.1.1.2. Siemens Simatic S7-1200 PLC and PROFINET I/O Controller

The Siemens Simatic S7-1200 Programmable Logic Controller (PLC) is a compact, modular controller designed for small to mid-sized automation tasks. As a PROFINET I/O Controller, it manages communication with distributed I/O devices over a PROFINET network, ensuring real-time data exchange and precise control. The S7-1200 also supports running PLC programs for testing and demonstration purposes, making it ideal for system development, testing, troubleshooting and validation.

The following are the details of the PLC and PROFINET I/O Controller:

- Manufacturer: Siemens
- Description: PLC and PROFINET I/O Controller
- Type: SIMATIC S7-1200
- Version: CPU 1211C DC/DC/DC (optional with CPU 1212C DC/DC/DC)
- Part Number: 6ES7211-1AE40-0XB0 (6ES7212-1AE40-0XB0)

7.1.1.3. Siemens 100BASE-TX Ethernet Switch

The following are the details of the Ethernet Switch:

- Manufacturer: Siemens
- Description: Ethernet Switch (unmanaged industrial 10/100BASE-T Network Switch)
- Type: SCALANCE XB008 (8 RJ45 Ports)
- Part Number: 6GK5008-0BA10-1AB2

7.1.1.4. PHOENIX CONTACT PROFINET I/O Device

This PROFINET Bus Coupler with 24 V Digital I/O is a reliable, *known good device* is essential for setting up the demo network controlled by the Siemens PLC before adding the Lattice CertusPro-NX evaluation board as new PROFINET I/O device. This ensures the proper functionality of the existing I/O controller and PROFINET network, enabling us to identify and analyze any issues that may arise during software development.

The following are the details of the PROFINET I/O device:

- Manufacturer: PHOENIX CONTACT
- Description: PROFINET I/O Device (Known Good Reference Device)
- Type: Axioline F PROFINET Bus coupler and 24 V Digital I/O Module (8x Input, 8x Output)
- Part Numbers: AXL F BK PN TPS (Article # 2403869) and AXL F DI8/1 DO8/1 1H (Article # 2701916)

7.1.1.5. Ethernet Network TAP

A Network TAP (Test Access Point) or Ethernet TAP, serves as a crucial hardware component inserted inline within a network link to effectively capture real-time data traffic traversing the network. This captured data is subsequently forwarded to a monitoring tool (such as Wireshark) for analysis.

The Dualcomm TAP provides the following features.

- Tested at a full 1 Gbps data throughput with no single packet loss
- Blocks ingress traffic of the monitor port to prevent a monitoring device such as a computer from injecting packets into the network

The following are the details of the Ethernet Network TAP:

- Manufacturer: Dualcomm Technology, Inc.
- Description: 10/100/1000Base-T Network TAP (active store-and-forward TAP)
- Model: ETAP-2003

7.1.1.6. Cat 6 UTP Network Cables

- Option 1: Cat 6 UTP Network Cable 0.3 m black
- Usage: PLC, PROFINET I/O Devices and Network TAP Connections
- Option 2: Cat 6 UTP Network Cable 2m white
- Usage: Ethernet Connections to Laptop

7.1.1.7. Siemens Simatic S7-1200 Input Switch Adapter

This adapter module enables manually control of the 24 V inputs on Siemens PLCs. Using the module switches, input[0:7] can be toggled on or off. It is designed to connect directly to the PLC input terminal block.

The following are the details of the input switch adapter:

- Manufacturer: Siemens
- Description: Input Simulator Module with Switches
- Type: SIM 1274, 8 DI (eight Input Switches)
- For PLC: SIMATIC S7-1200
- With CPU: 1211C (6 DI 24 V DC) or 1212C (8 DI 24 V DC)
- Part Number: 6ES7274-1XF30-0XA0

7.1.1.8. ADI Motion Control Board

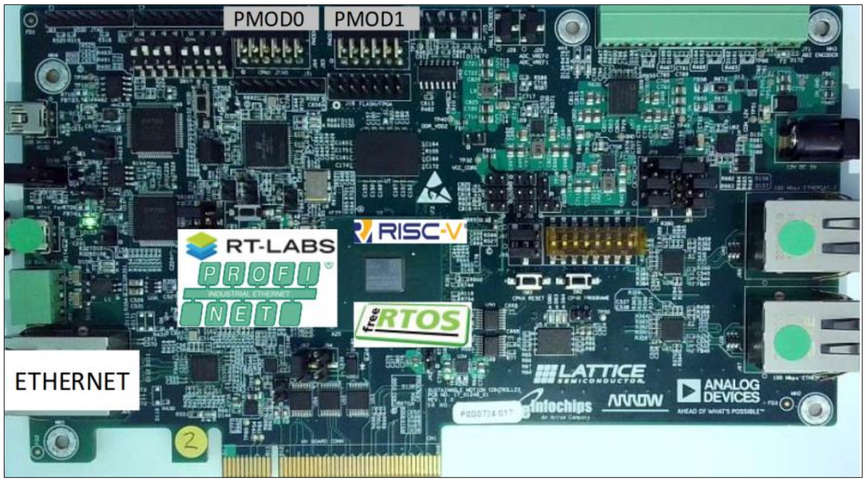

Figure 7.3. ADI CertusPro-NX Motion Control Evaluation Board

Figure 7.4 shows the CertusPro-NX motion control block diagram.

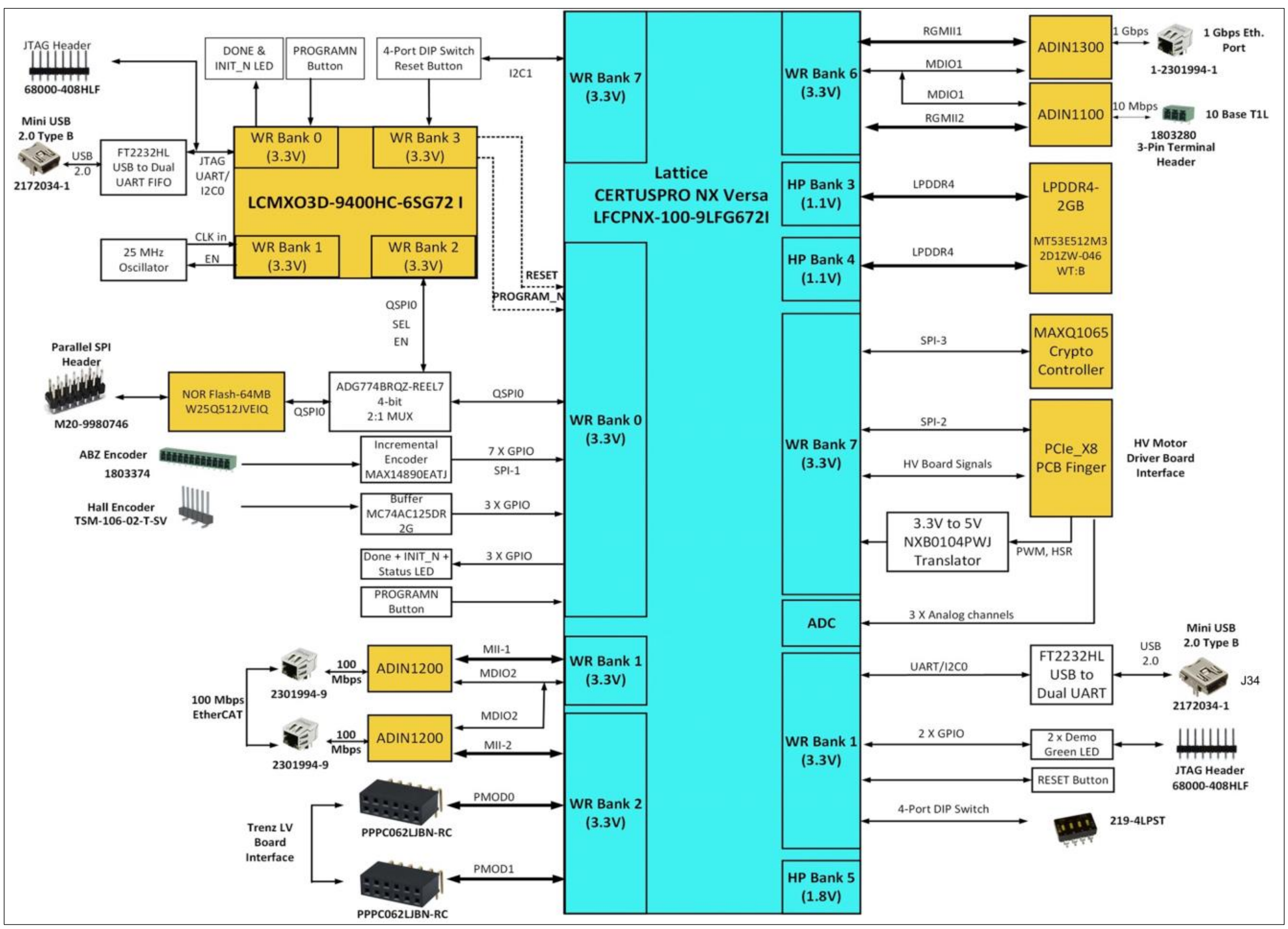

Figure 7.4. ADI CertusPro-NX Motion Control Block Diagram

7.1.1.9. PMOD Test Headers and Modules

The PMOD interface (peripheral module interface) is an open standard defined by Digilent in the PMOD Interface Specification for connecting the peripheral modules to the FPGA and Microcontroller development boards.

The following are the details of the test headers and modules:

- Manufacturer: Digilent
  - Option 1: PMOD TPH2 (12-pin Test Header)
    - Description: 12 external pins to easily test the GPIO signals passing through each of the PMOD pins.
  - Option 2: PMOD 8LD (Eight High-brightness LEDs)
    - Description: Eight LEDs with driver, each LED can be individually illuminated from a logic high signal.
- Manufacturer: 1BitSquared
  - Description: PMOD DIP-Switch V1.1a (8 Inputs switching to +3.3 V with 10K pull-down resistors)

7.1.1.10. PMOD Configuration Options

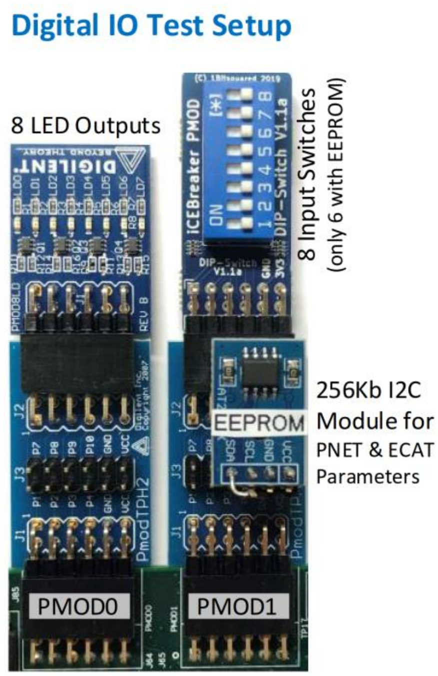

Figure 7.5. DIGITAL I/O Setup for LED

7.1.2. Software Components

7.1.2.1. Lattice Radiant Programmer – FPGA Fabric and System Memory Configuration Tool

The Lattice Radiant software offers all the tools and features to help you develop your FPGA applications. The Radiant Programmer is part of the Radiant Design software, but a standalone version is also available.

- **Functions**:
  - Download of bitstream (BIT) configuration files to setup the FPGA fabric and system memory directly or to store the configuration data in an external SPI Flash memory to enable FPGA configuration and initialization of internal system memory during power-up
  - Download of binary (BIN) data files to the external SPI Flash memory to enable the bootloader software moving the data to the external LP-DDR4 RISC-V code memory
- **License**: Free Node-locked License available with Lattice Semiconductor account

7.1.2.2. Siemens PRONETA – Diagnostics Tool for PROFINET Networks

PRONETA is a PC-based commissioning and diagnostics tool for PROFINET Networks. The free basic version is used in this Demo for scanning the PROFINET Network. A licensed version with additional functionalities is also available for a simple commissioning of the network and an easy check of the distributed I/O of connected I/O devices, without PLC.

- Demo Functions: Scanning the network for accessible devices, setup of IP addresses and device names
- License Options: free Software (basic version) and licensed version with more functions
- Download Links: PRONETA Basic 3.8 (or higher)
- Npcap 1.80 installer for Windows (free software)

To install the Siemens PRONETA, perform the following:

1. Download the ZIP File from Siemens website
2. Unpack the ZIP and store the PRONETA.exe file
3. No Installation is required, but the NPCAP driver is needed to run PRONETA.
4. Install the NPCAP version 1.80 or higher before starting PRONETA the first time.

   NPCAP allows the Windows software to capture raw network traffic (such as the Ethernet) using a simple, portable API. NPCAP allows you to send raw packets as well. The Mac and Linux systems already include the NPCAP API, so NPCAP allows popular software such as Wireshark to run on all these platforms with a single codebase.

7.1.2.3. Siemens TIA Portal – Engineering and Diagnostics Tool (SIMATIC STEP 7 Basic Module)

The Siemens TIA Portal (Totally Integrated Automation Portal) is a powerful engineering software platform designed for configuring and programming industrial automation systems, with a focus on PROFINET networks and I/O device integration. It simplifies PLC programming, network setup, and device configuration with a unified interface for managing PROFINET-enabled devices.

The following are the details of the Siemens TIA Portal:

- Function: PLC Engineering Software for Network Configuration and PLC Programming
- Type: TIA Portal
- Module: SIMATIC STEP 7 Basic
- Version: V19 (or higher)

The following are the guidelines and notes to installing and testing the Siemens TIA with the reference design:

- Image File: C:\Tools\Siemens\TIA_Portal_STEP7_Prof_Safety_WinCC_V19.iso
  Note: The ISO Image is shown with Disk Symbol (see Figure 7.7).

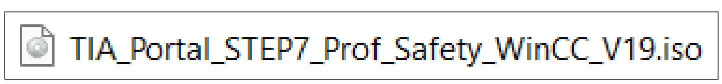

Figure 7.6. ISO Image for TIA Portal

To install the Siemens TIA portal, perform the following:

1. Double-click on the mounted DVD Drive to open the DVD disk image.
2. Click **start.exe**. Install Option is *Minimal* (not *Typical*, not *user defined*).

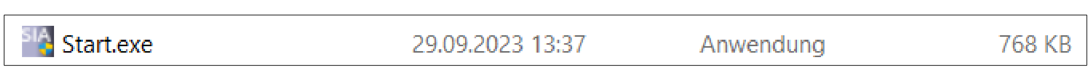

Figure 7.7. Start Button to Install

3. After the reboot, open the DVD Disk Image again after the *no DVD Drive detected* message.

7.1.2.4. Wireshark Network Protocol Analyzer

Wireshark is a powerful network analyzer tool used for capturing and analyzing network traffic in real time. It provides detailed insights into Ethernet Layer two frames, allowing you to inspect protocols (identified by a unique EtherType value) like IPv4 (UDP and TCP), ARP, LLDP and PROFINET.

For PROFINET analysis, Wireshark decodes the protocol's real-time communication between I/O controllers and I/O devices, offering precise diagnostics and troubleshooting for industrial automation networks.

The following are the details of the Wireshark Network Protocol Analyzer:

- Functions: Capturing of the Ethernet LY2 frames and analysis of IP and PROFINET protocols
- License: free software
- Download: Wireshark tool

7.2. Evaluation Board Configuration with Lattice Radiant Programmer

You must configure the Lattice ADI Evaluation Board with the provided PROFINET Demo Example before you can use it properly. The following components must be programmed first using the Lattice Radiant Programmer software tool. Once configured, the board can be fully tested to verify the functionality of the PROFINET setup and the Evaluation Board itself. This process ensures the board is ready for further testing with the customer development project.

The following are the description for the binaries used in the Reference Design

- FPGA Logic Fabric with the Lattice Hardware Reference Design
  This is the content of the project bitstream programming file (for example, with RISC-V MCU IP and Ethernet MAC IP).
- Bootloader Software that is part of the Lattice Software Reference Design
  This is the content of the project bitstream programming file to setup the FPGA system memory.
- PROFINET Software Stack, TCP/IP Stack and Application Software
  This is the content of the project Binary programming file, to enable the bootloader to move the software code to the external LP-DDR4 memory.

The *.bit* and *.bin* files are programmed separately to different locations of the FPGA SPI Flash configuration memory as described below.

To program the bit file to Start Address 0x0000 0000, perform the following:

1. Download and install the *Lattice Radiant Programmer Software* in the Lattice website.
2. Connect the USB port aside the green connector on the left side of the board to your PC or Laptop.
3. Power-up the evaluation board.
4. Start the Radiant Programmer. Create a new project or open the *existing project* within the demo zip file. Figure 7.8 is displayed after clicking on *Detect Cable,* if everything is fully functional.
5. Click **OK** to go back to the default screen.
6. Repeat the *Detect Cable* process in case of any error messages.

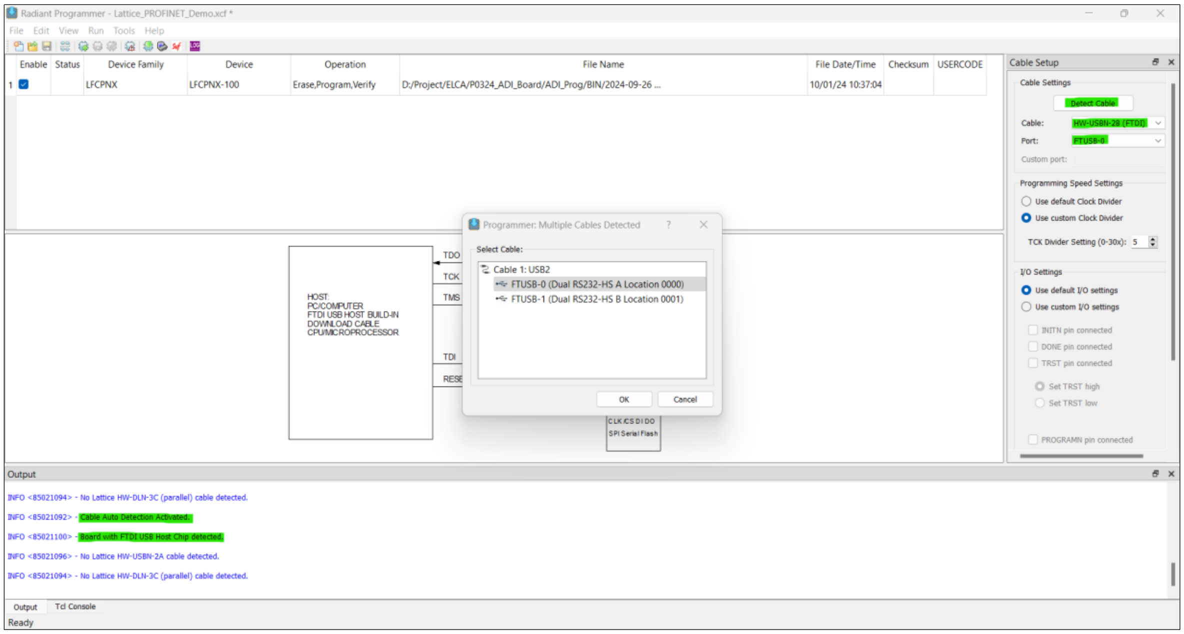

Figure 7.8. Bit File Programming for ADI MCB

7. Enter the bitstream file name.
8. Double-click on the *Erase/Program/Verify* field or any other mode that may be displayed to open the Device Properties window.
9. Apply the settings for **Device Operation**, **Programming Options (Programming file)** and **SPI Flash Options** as shown in Figure 7.9.
    a. Set the operation mode to **Erase/Program/Verify**.
    b. Make sure that the following addresses are correct:
        - Start Address (Hex) – **0x00000000**
        - End Address (Hex) – **0x00240000**

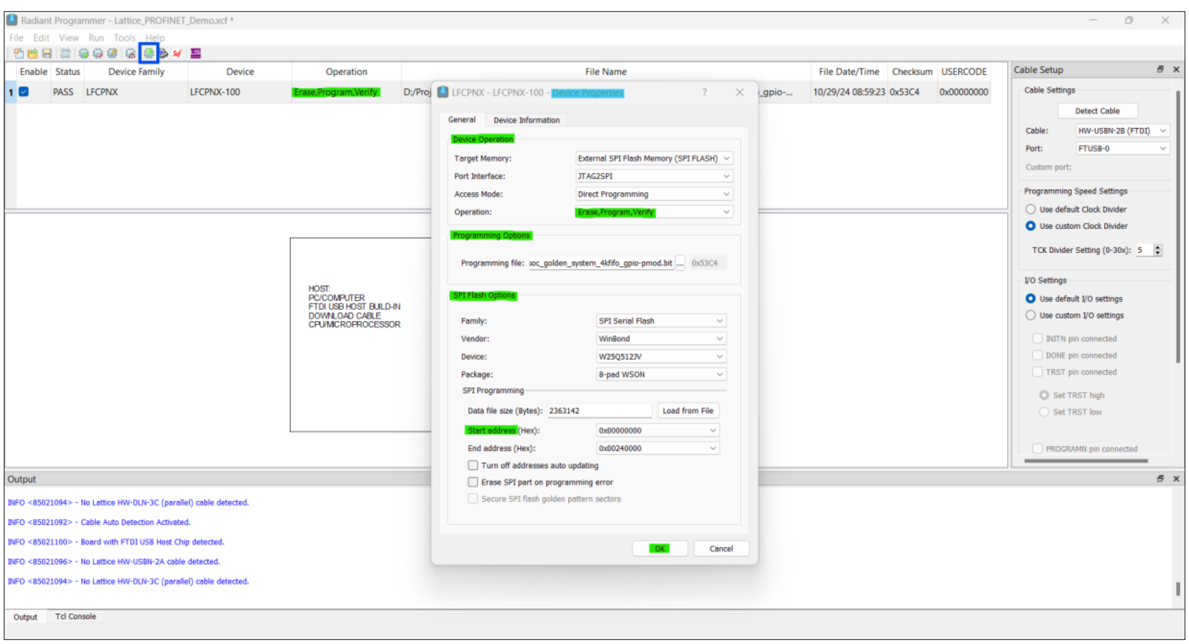

Figure 7.9. Configuring Bit File Programming for ADI MCB

10. Click **OK** to close the Device Properties window
11. Click the **Program Device** icon in the main menu to start the programming process.
12. The bitstream file is programmed to **Start Address 0x0000 0000** of the SPI Flash configuration memory.
13. The following sequence is displayed on the **Output** window once the operation is successful.

```
INFO <85021096> - No Lattice HW-USBN-2A cable detected.
INFO <85021094> - No Lattice HW-DLN-3C (parallel) cable detected.
INFO <85021074> - Check configuration setup: Start.
INFO <85021077> - Check configuration setup: Successful (Ignored JTAG Connection
Checking).
INFO <85021294> - Device1 LFCPNX-100: LFCPNX-100: Refresh Verify ID
INFO <85021298> - Operation Done. No errors.
INFO <85021294> - Device1 LFCPNX-100: W25Q512JV: Erase,Program,Verify
Initializing...   IDCode Checking...
Enter 4-Byte mode...   Enabling...   Erasing...   Disabling...
Enabling... Programming... Disabling... Verifying...
INFO <85021399> - Execution time: 01 min : 18 sec
INFO <85021371> - Elapsed time: 01 min : 24 sec
INFO <85021373> - Operation: successful.
```

To program the bin file to Start Address 0x020A 0000, perform the following:

1. Enter the binary file name.
2. Double-click on the *Erase/Program/Verify* field or any other mode that may be displayed to open the *Device Properties* window.
3. Check the settings for Device Operation, Programming Options (Programming file) and SPI Flash Options as shown in Figure 7.9.
    a. Set the operation mode to **Erase/Program/Verify**.
    b. Make sure that the following addresses are correct:
        - Start Address (Hex) –**0x00240000**
        - End Address (Hex) – **0x00000000**

    The new end address is automatically calculated using the end address after loading as offset.
4. Click **OK** to close the Device Properties window.
5. Click the **Program Device** icon in the main menu to start the programming process.
6. The bitstream file is programmed to Start Address 0x020A0000 of the SPI Flash configuration memory.
7. The following sequence is displayed on the Output window in case of a successful operation.

7.3. Scanning the Network for Accessible PROFINET Devices with Siemens PRONETA

Before adding a new I/O device to a PROFINET network, use the PRONETA scanning tool to confirm that the device is functioning and accessible. If the device cannot be detected or accessed, the configuration process with the TIA Portal engineering tool fails. PRONETA ensures that the device is connected and ready, making the subsequent configuration process successful.

PRONETA is a quick and easy solution for verifying a PROFINET I/O device setup without the need to install the more complex TIA Portal software.

Note: Set the local IP address of the used Ethernet Adapter to 192.168.0.1 (subnet: 255.255.255.0).

To scan the network, perform the following:

1. Start the **PRONETA.exe** and click *Network Analysis*.
2. Click **Online – Refresh**.
3. After the successful scan process, the accessible devices are listed in the **Device Table**.
4. Select the device within the table to view the details. The information of the selected device is displayed in the *Device Details* window.

7.4. Configuring the PROFINET Device Names and IP Addresses with Siemens PRONETA

Note: This an easy and direct approach to prepare the I/O Device Names and IP Addresses before using the Siemens TIA Portal software.

To configure the PROFINET device name and IP address, perform the following:

1. Scan the network as described in the Scanning the Network for Accessible PROFINET Devices with Siemens PRONETA section.
2. In the **Network Analysis - Online - Graphical View** window, select **device io_1** to show the device details.
3. Double-click on the device (for example, io_1) to open the Set Network Parameters window.
4. Set the **Device Name**, **Static IP Address**, and **Subnet Mask**.
5. Select **Apply settings permanently** and click the **Set** button to change the device settings.

7.5. Quick Start with the Siemens TIA Sample Project and Programming the FPGA Device

The Siemens TIA sample project is available that directly works with the hardware setup shown in the PROFINET Demo Overview diagram in the Demo Overview section.

The TIA sample project enables a quick start of the PROFINET Demo by following the guidelines below.

1. Order the PROFINET demo network hardware components.
    - Review the manufacturer product links and ordering information in the Hardware Components section.
2. Build the PROFINET demo hardware setup.
    - Review the demo overview diagram in the Demo Overview section.
    - Follow the manufacturer product datasheets and installation instructions.
3. Download and install the *Lattice Radiant Programmer* software.

   Note: The Radiant Programmer is part of the Radiant Design software.
    - Sign-In or Register for a Lattice Semiconductor account.
    - Download the Windows version of the Radiant Design software.
    - Follow the Installation Guides or download and install the Programmer Standalone 2024.1.1 64-bit for Windows.
    - Request a free Node-locked License.
4. Run Radiant Programmer to program the qSPI Flash on the Lattice ADI MCB evaluation board
    - Open the programmer sample project of the demo ZIP file (PROFINET Demo.xcf).
    - Program the bitstream file (FPGA fabric and bootloader software) to Start Address 0x0000 0000.
    - Program the binary file (PROFINET stack and application software) to Start Address 0x020A 0000.
5. Download and Install the *Siemens PRONETA* software tool.
    - Review the product information and download link in the Software Components section.
    - Follow the guidelines described in the Software Components section.

6. Run PRONETA to scan and setup the network.
    - Scan the network for accessible devices as mentioned in the Scanning the Network for Accessible PROFINET Devices with Siemens PRONETA section.
    - The Siemens PLC, Phoenix Contact I/O device, and rt-labs-dev I/O device appears.
    - Configure the PROFINET names and IP addresses according to the overview diagram in the Demo Overview section.
    - Follow the guidelines in the Testing of basic PROFINET I/O Device Functions section for any changes, and then scan the network again to check the status.
7. Download and install the *Siemens TIA Portal* software tool.
    - Review the product information and download link in the Software Components section.
    - Follow the guidelines described in the Software Components section.
8. Run TIA Portal with the PROFINET demo sample project.
    - Open the TIA sample project of the demo zip file.
9. Check current device name and IP address configuration
    - Use the guidelines in the Checking the Current Device Names and IP Addresses section.
    - The settings must be the same as defined in the PROFINET overview diagram.
10. Compile and load the PLC configuration and control software.
    - Follow the guidelines in the Compiling and Loading the PROFINET System Configuration section.
11. Go online and check the PROFINET network status.
    - Follow the guidelines in the Checking the System Status Online section.
    - Check the status symbols in the project tree window.
    - All status symbols should be green as in screenshot in the Checking the System Status Online section.
    - Switch on the local inputs [0:3] of the Siemens PLC device.
    - This sets the local outputs [0:3] of the PLC device to the active HI state (LEDs on).
    - For io_1 and rt-labs-dev device tests, refer to the Testing of basic PROFINET I/O Device Functions section.

7.6. Testing of basic PROFINET I/O Device Functions

The PROFINET I/O functionality of the example application implemented on the Lattice ADI evaluation board (see the configuration in the Evaluation Board Configuration with Lattice Radiant Programmer section) can be tested with a Siemens TIA Portal demo project provided by Lattice. For general guidance on using the TIA software, a complete tutorial on setting up a project from scratch is available in the Setting Up a New Demo Project Using Siemens TIA section.

The TIA demo project includes the following functions:

- Verification of Siemens PLCs PROFINET I/O Controller functionality with a simple PLC control program
- Verification of PROFINET I/O Reference Device functionality by using its digital inputs and outputs
- Testing and demonstration of PROFINET I/O user device functionality on the Lattice ADI Evaluation Board

The TIA demo project provides the following features for testing the Lattice PROFINET sample application:

- Testing of one digital system output on PMOD0[5] (PROFINET *Flash LED* diagnostic feature)
- Testing of five digital user outputs on PMOD0[0:4]
- Testing of six digital user inputs on PMOD1[0:5], PMOD1[6:7] is reserved for an I2C EEPROM module

7.6.1. Connection of PMOD Test Modules to the Lattice ADI Evaluation Board

For I/O testing, connect the PMOD modules as shown in the *Digital I/O Test Setup* below. This allows verification of the output status on PMOD0[0:7] through LEDs and enables switching of the inputs on PMOD1[0:7]. The *Motor Control Setup* is an example for a Dual PMOD module with a connector distance specified in the PMOD Specifications.

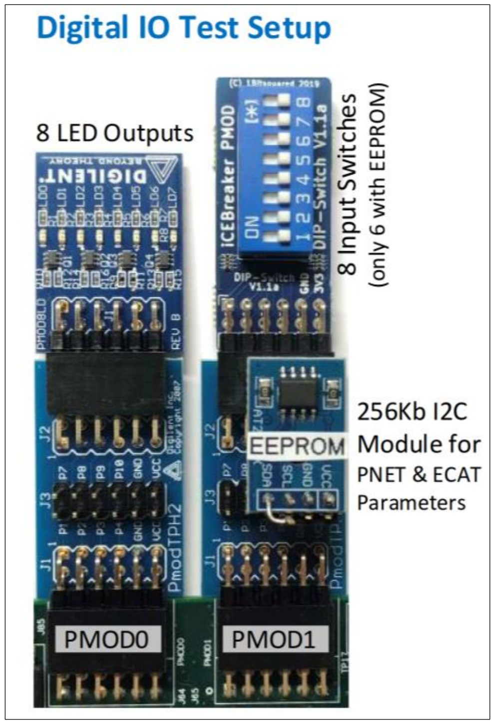

Figure 7.10. Demo PMOD Modules for LED I/O

7.6.2. Testing the Digital System Output (Flashing LED) on PMOD0 Peripheral Connector

The flashing LED function on the PROFINET I/O devices is primarily used for visual identification and diagnostics. In networks with multiple devices, it enables maintenance engineers to locate specific devices easily within complex setups or large installations. This mandatory feature, which can be triggered remotely through the Siemens TIA Portal (engineering and maintenance tool), simplifies device identification and troubleshooting.
Activate and verify the Flash LED function on the Lattice evaluation board:

1. Open the Siemens TIA Portal software tool.
2. Open the Lattice demo example project.
3. Go to *Project view* and click the **Accessible devices** button in the toolbar and then click **Start search**.
4. The following status output appears in the Accessible devices window when all PROFINET devices according to the overview slide can be accessed during the search process.

   Note: To verify that the test setup (built according to the overview slide) is functioning correctly, you must test the *Flash LED* feature on each network device, starting with the PLC. This ensures that all hardware connections are properly established and that, on devices with blinking LEDs, the PROFINET protocol stack is working. Additionally, you can ping the IP addresses listed on the *Accessible nodes* table, this should be possible for each device with no issues.
5. Select **plc_1** and check the Flash LED box to activate the flashing LED on the Siemens PLC device.
6. Select **io_1** and activate the flashing LED on the PHOENIX CONTACT Reference I/O device.
7. Select **rt-labs-dev** and activate the flashing LED on the Lattice evaluation board I/O device implementation.

7.6.3. Testing of Digital User Outputs and Inputs on PMOD Peripheral Connectors

The digital user outputs of the FPGA on the Lattice ADI board, connected to the PMOD port 0, can be easily tested using an LED test module and a simple PLC control program. The inputs of the IO_1 reference device are used to switch and toggle the outputs on the Lattice evaluation board. To set the status of the 24V inputs on the PHOENIX CONTACT I/O device, connect manual switches as shown below.

The digital user inputs of the FPGA on the Lattice ADI board, connected to the PMOD port 1, can be easily tested using a DIP switch test module and the same PLC control program. The inputs of the Lattice ADI board are tested by switching the distributed outputs on the io_1 reference device. No other connections are needed.

If this works, it not only confirms that the user I/O are functioning correctly but also indicates that all hardware and software components are operating properly. This includes the PLC controlling the PROFINET network and processing the control program, the io_1 device and the PROFINET stack with the application software on the evaluation board (rt-labs-dev).

Figure 7.11 shows the testing of the rt-labs-dev user outputs [0:4] with switches on the io_1 device and the PMOD0 LED module.

Note: User output [5] is reserved for the *Flash LED option* and cannot be switched.

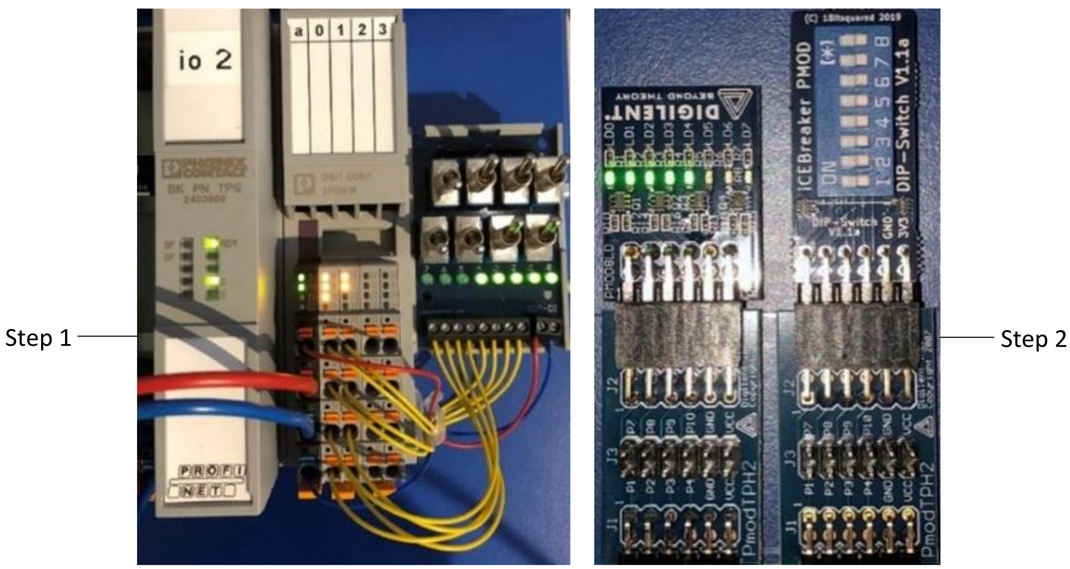

Figure 7.11. Testing Methods for Output

Figure 7.12 shows the testing of the rt-labs-dev user inputs [0:5] with PMOD1 DIP switch module and output LEDs on io_1 device.

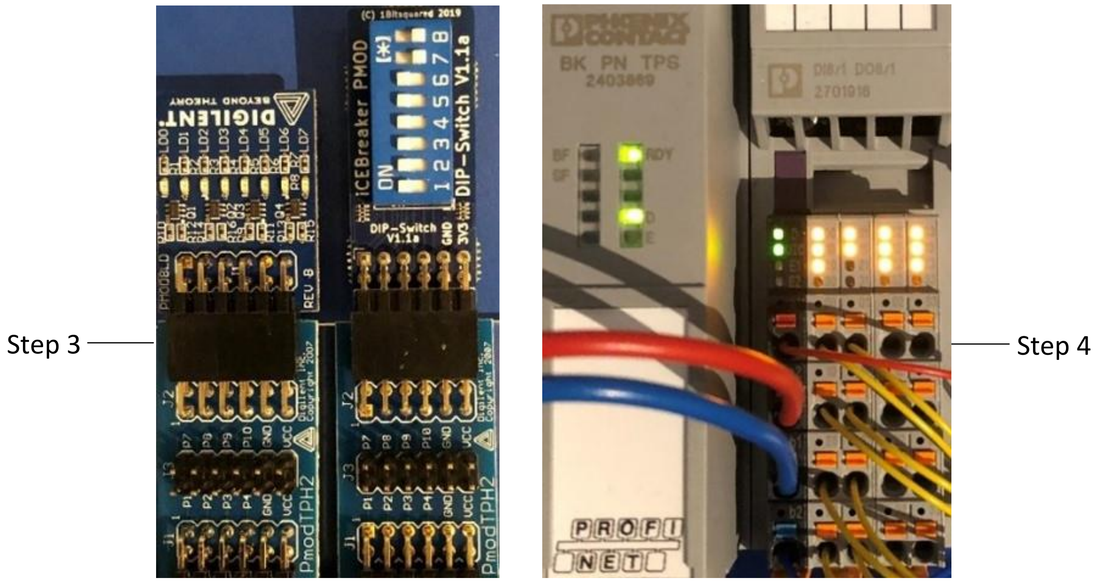

Figure 7.12. Testing Methods for Input

7.7. Capturing and Analyzing the PROFINET Cyclic Ethernet Data Transfers

Using Wireshark, you can capture Ethernet Layer II traffic and decode I/O packets based on the PROFINET specification, identified by the Ethertype value 0x8892. Cyclic PROFINET I/O data exchanged between the I/O Controller (PLC) and I/O Devices is encapsulated within these Ethernet frames. Wireshark enables detailed analysis of parameters like cycle time, status information and diagnostics data, essential for troubleshooting and optimization of PROFINET I/O communication (see examples below for the rt-labs-dev sample application).

Figure 7.13 shows the PROFINET I/O cyclic data transfer from PLC (I/O Controller) to the Lattice ADI Board (I/O device).

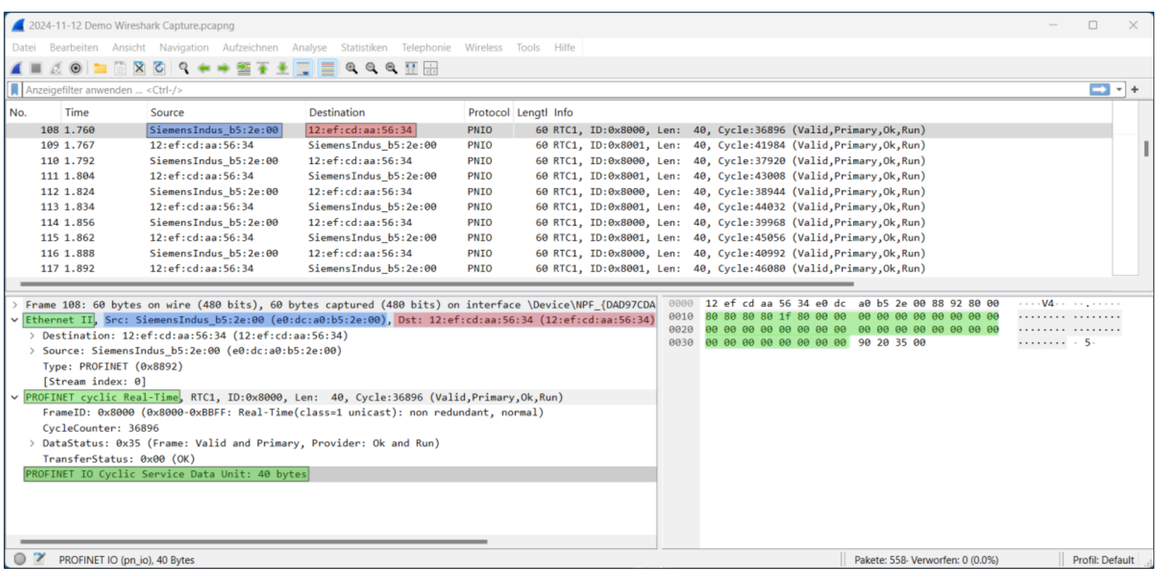

Figure 7.13. PROFINET Cycle Data Transfer from PLC to ADI MCB

Figure 7.14 shows the PROFINET I/O cyclic data transfer from the Lattice ADI board (I/O device) to PLC (I/O Controller).

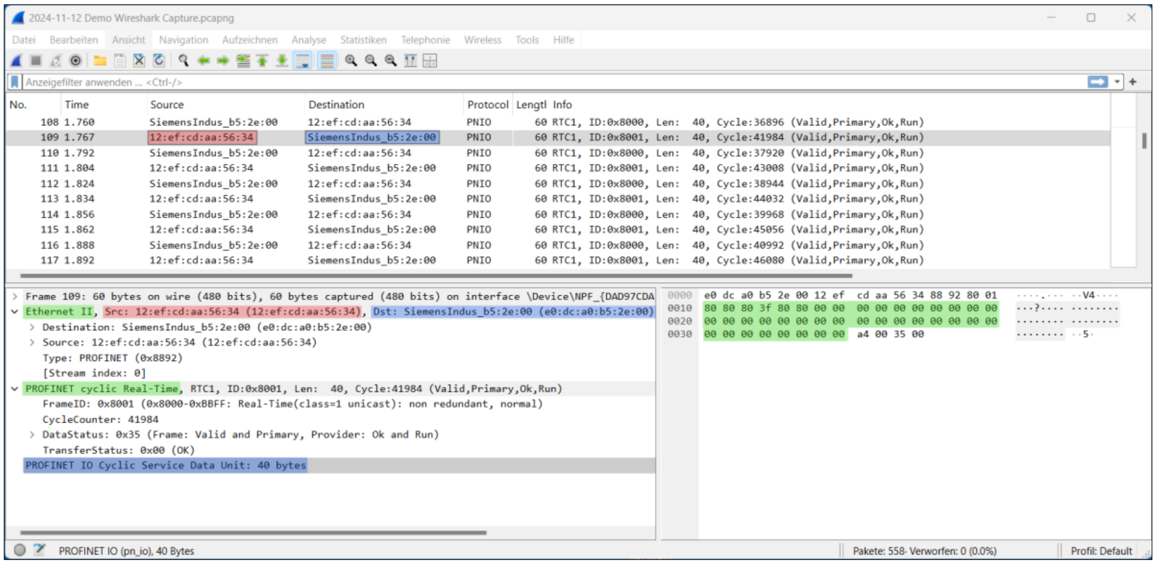

Figure 7.14. PROFINET Cycle Data Transfer from ADI MCB to PLC

7.8. Setting Up a New Demo Project Using Siemens TIA

7.8.1. Creating a New Project

To create a new project, perform the following:

1. Open the TIA Portal.
2. In the Portal view, select **Create new project** and enter project path, name, and version.
3. Click **Create** and go to the *Project* view with the button located on the bottom left corner of the screen.
4. The Project view screen appears.

7.8.2. Adding the Siemens PLC Device

To add the Siemens PLC device, perform the following:

1. In the **Project tree - Device** window, double-click on the **Add new device**.
2. In the **Add new device** window, select the correct Siemens PLC type and click **OK**.
3. Configure the **PLC Security** settings.
4. Run through all topics and choose no protection features for the demo.
5. Click **Finish**.
6. The **PLC_1 device** is shown in the **Device view** window.

7.8.3. Adding the GSDML Files to the Current Project

The GSDML file (General Station Description in XML format) contains technical details about the device, such as manufacturer information, device type, and version numbers. It also describes the available input and output channels of an I/O device, defining what data the device sends and receives. Additionally, the GSDML file is used in a PROFINET engineering tool (like Siemens TIA Portal) to integrate the device into the system (for example, for adding a device to a PROFINET Network), allowing the proper configuration and programming.

To add the GSDML file to the current project, perform the following:

1. Create **sub-folder GSD** within the **Additional Files** folder of the project structure.
2. Load **GSDML file** from product website (for example, PHOENIX CONTACT).
3. Store the **GSDML file** in the GSD sub-folder.
4. Open the **Project view - Options - Manage general station description files (gsd)** window.

5. Select the GSD sub-folder as Source path. The **GSDML** files are displayed.
6. Select the GSDML file (if not yet installed) and click **Install**. The *hardware catalog* is updated automatically.
7. Double-check the installed GSDs.

7.8.4. Adding the Phoenix Contact I/O Device as Reference

To add the Phoenix contact I/O device, perform the following:

1. In the **Project tree - Devices** window, double click on **Devices and networks**. The *Devices and networks - Network view* windows appear in the middle of the screen.
2. The **Hardware catalog** window appears on the right side of the screen.
3. In the **Catalog** window, extend the *Other field devices* product tree to **PROFINET I/O > I/O > Phoenix Contact > Axioline**.
4. Click the product type **AXL F BK PN TPS**. Drag and drop it to the *Devices and networks – Network view*. The *Phoenix Contact I/O Device* appears in the *Devices and networks – Network view* window. The new I/O device is added but not yet configured and assigned to a PLC device. The device also appears in the *Ungrouped devices* section of the **Project tree - Devices** window.

   This section always provides direct access to the Device configuration and Online and diagnostics features.

7.8.5. Adding the RT-Labs GSDML File and I/O Device

To add the RT-Labs GSDML file and I/O device, perform the following:

1. Download the RT-Labs GSDML file from the *Github* website.
2. Store the **GSDML file** in the GSD sub-folder.
3. Open the **Project view - Options - Manage general station description files (gsd)** window.
4. Select the GSD sub-folder as source path.
5. Double-check the **Installed GSDs**. If it is not yet installed, select the GSDML file and click **Install**. The hardware catalog is updated automatically.
6. In the **Catalog** window, extend the *Other field devices* product tree to **PROFINET I/O > I/O > RT-Labs > P-Net Samples**.
7. Click the product type P-Net multi-mode sample app. The **RT-Labs I/O Device** appears in the *Devices and networks – Network view* window. The new I/O device is added but is not yet configured and assigned to the PLC device. The device also appears in the *Ungrouped devices* section of the **Project tree - Devices** window.

   This section always provides direct access to the Device configuration and online and diagnostics features.

7.8.6. Adding the I/O Module of the Phoenix Contact I/O Device Head Module

To add the I/O module of the Phoenix Contact I/O device, perform the following:

1. Double click the **io_1** symbol in the *Devices and networks – Network view* window.
2. To enter the Ungrouped devices – io_1 [AXL F BK PN TPS] - Device view window, select the **I/O digital – AXL F DI8/1 DO8/1 1H** module in the *Hardware catalog* window.
3. Drag and drop it to the *Device overview* window.
4. Check the **Ungrouped devices – io_1 [AXL F BK PN TPS] - Device overview** window. The selected digital I/O module type with the I/Q addresses appears.

7.8.7. Adding the I/O Module to the RT-Labs I/O Device Head Module

To add the I/O Module to the RT-Labs I/O device head module, perform the following:

1. Double-click the **rt-labs-dev** symbol in the *Devices and networks – Network view* window.
2. To enter the Ungrouped devices - rt-labs-dev[P-Net multi-mode sample app] - Device view window, select the DI 8xLogicLevel and DO 8xLogicLevel modules in the *Hardware Catalog* window.
3. Drag and drop it to the **Device overview** window (one by one).

4. Check the **Ungrouped devices – rt-labs-dev[P-Net multi-mode sample app] - Device overview** window. The types of the selected digital I/O modules with the I/Q addresses appear.

7.8.8. Checking the Current Device Names and IP Addresses

To check the current device names and addresses, perform the following:

1. In the Devices and networks – Network view window, click the Show address labels icon.
2. The **device names** and **IP addresses** can be changed with the PRONETA tool. The *name* and *IP address* of the rt-labs-dev is fixed and cannot be changed.

7.8.9. Assigning the I/O Devices to the Siemens PLC Device

To assign the I/O devices to the Siemens PLC device, perform the following:

1. Drag and drop a connection line to assign the **io_1** I/O device to **plc_1 ()**.
2. Drag and drop a connection line to assign the **rt-labs-dev I/O** device to **plc_1**.

7.8.10. PLC and I/O Device Configuration

To configure the Siemens PLC I/O Controller (Minimum Cycle Time), perform the following:

1. In the **Project tree - Devices** window, double-click on **plc_1 [CPU 1211C DC/DC/DC] - Device configuration**. The **plc_1 - Device view** window appears.
2. In the **Device view - Properties - General** window, set the Minimum cycle time to 10 ms.

To configure the PHOENIX CONTACT I/O device (update the Time and Watchdog Cycles), perform the following:

1. In the **Project tree - Devices** window, double-click on **Ungrouped devices - io_1 - Device configuration**. The *Ungrouped devices - io_1 [AXL F BK PN TPS] - Device view* window appears.
2. In the **Device view - Properties - General** window, select the real time settings for the I/O cycle.
3. Set the update time to **2 ms**.
4. Set the accepted update cycles without I/O data parameter to **3 cycles**. The Watchdog time is calculated automatically and set to the update time value multiplied by 3.

To configure the Lattice RT-LABS I/O device (Update Time and Watchdog Cycles), perform the following:

1. In the **Project tree - Devices** window, double-click on **Ungrouped devices - rt-labs-dev - Device configuration**. The *Ungrouped devices - rt-labs-dev - Device view* window appears.
2. In the **Device view - Properties - General** window, select the Real time settings for the I/O cycle.
3. Set the update time to **32 ms**.
4. Set the accepted update cycles without I/O data parameter to **3 cycles**. The Watchdog time (ms) calculated automatically and set to the update time multiplied by 3.

7.8.11. Checking the Device I/O Addressing within the PLC System Memory

7.8.11.1. Introduction

If access is made to the inputs and outputs through the operands *%I* or *%Q* in the user program, then there is no direct access to the local or distributed input/output modules. In this case, access is made to a memory area in the CPU system memory. This area is called the process image of the inputs (PII) and the process image of the outputs (PIQ). The contents of the process image do not reflect the actual values of the inputs/outputs, but the values at the time the process image was updated.

7.8.11.2. Processing of the PLC program

The Main[OB1] process image of the PLC program is updated cyclically before the OB1 is processed.

The program is processed cyclically in the following order:

- Processing of internal jobs of the CPU operating system.
- Writing of states from the OB1 process image of the outputs to the outputs of the modules.
- Reading of the input states from the modules into the OB1 process image of the inputs.

- Processing of the user program in OB1.
- Continue with Step 1.

The memory area types for absolute addressing of the S7-1200 process image:

- %I (Process image input)
  The CPU copies the state of the physical inputs to *I memory* at the beginning of each cycle.
- %Q (Process image output)
  The CPU copies the state of *Q memory* to the physical outputs at the beginning of each cycle.

7.8.11.3. PLC Tags and Symbolic Addressing of the Image Process

Step 7 Basic symbolic programming for tagging the process allows you to create the symbolic names or tags for the addresses of the data, whether as PLC tags relating to memory addresses and I/O points or as local variables used within a code block. To use these tags in your user program, enter the tag name for the instruction parameter.

7.8.11.4. Absolute Addressing Referenced by the PLC Tags

The CPU provides a variety of specialized memory areas, including inputs (I), outputs (Q), bit memory (M), data block (DB), and local or temporary memory (L). The user program accesses (reads from and writes to) the data stored in these memory areas. Each different memory location has a unique address. The program uses these addresses to access the information in the memory location.

Note: Whether you use a tag (such as *Start* or *Stop*) or an absolute address (such as *%I0.3* or *%Q1.7*), a reference to the input (I) or output (Q) memory areas accesses the process image and not the physical output. It is important to check that the absolute addresses (%Ix.x or %Ox.x) referenced by the user defined *PLC tags* are consistent with the configured system memory area configured for the PROFINET I/O devices.

For more information, refer to the *Getting Started with S7-1200, 11/2009, A5E02486791-01 Siemens* document.

7.8.11.5. TIA Portal – Reviewing Configured Device I/O Addressing

To review the configured device I/O Addressing, perform the following:

1. Double-click on the **rt-labs-dev symbol** in the *Devices and networks – Network* tab.
2. To enter the **Ungrouped devices - rt-labs-dev[P-Net multi-mode sample app] - Device view** window, place the **Device overview** window on the right side of the **Device view** window.
3. Select the **DI 8xLogicLevel** module in the **Device overview** window.
4. In the **DI 8xLogicLevel_1 - Properties - General** window, check the Input address entries. The **start** and **end** addresses must have the same value because the output capability is one byte. The value for the **rt-labs-dev** I/O device **Input addresses** must be *2*.
5. The PLC device start/end address is 0, the reference I/O device start/end address is 1 followed by the rt-labs-dev. It is important that there are no conflicting device settings.
6. Select the **DO 8xLogicLevel** module in the **Device overview** window.
7. Follow the same steps as above for the **Output addresses.** The value for the **rt-labs-dev** I/O device **Output addresses** must be *2* as well.

7.8.11.6. TIA Portal – Verifying Configured Device I/O Addresses and Review of Device Names and IP Addresses

To verify the configured device I/O Addressing, perform the following:

1. Click the **PN/IE_1 network connection** in the *Devices and networks – Network view* window.
2. To open the **PN/IE_1[Industrial Ethernet] – Properties – General** window to access the properties section.
3. In section **plc_1.PROFINET I/O System**, review the I/O address settings in sub-section **Overview of addresses**.
4. Check if I/O addressing is equal to the previous setting.
5. Check if device names and IP addresses are correctly set according to the demo overview slide.

7.8.12. Setting Up the PLC Tags to Enable Symbolic Addressing of the Process Image

This section explains how to create symbolic names, or *tags*, to simply address digital inputs and outputs in a PLC control program by using these tag names to reference elements in the PLC process image memory.

To set up the PLC tags, perform the following:

1. In the **Project tree - Devices** window, double-click on **plc_1 - PLC tags - Show all tags** to open the view.
2. In the **plc_1[CPU 1211C DC/DC/DC] - PLC tags – Tags** window, add the following entries to enable the symbolic addressing for the PLC local I/O and the remote I/O on the PROFINET I/O devices.

PLC Local Input Tags

LOCAL_IN0	Default tag table	Bool	%I0.0	☐	☑	☑	☑
LOCAL_IN1	Default tag table	Bool	%I0.1	☐	☑	☑	☑
LOCAL_IN2	Default tag table	Bool	%I0.2	☐	☑	☑	☑
LOCAL_IN3	Default tag table	Bool	%I0.3	☐	☑	☑	☑
LOCAL_IN4	Default tag table	Bool	%I0.4	☐	☑	☑	☑
LOCAL_IN5	Default tag table	Bool	%I0.5	☐	☑	☑	☑

PLC Local Output Tags

LOCAL_OUT0	Default tag table	Bool	%Q0.0	☐	☑	☑	☑
LOCAL_OUT1	Default tag table	Bool	%Q0.1	☐	☑	☑	☑
LOCAL_OUT2	Default tag table	Bool	%Q0.2	☐	☑	☑	☑
LOCAL_OUT3	Default tag table	Bool	%Q0.3	☐	☑	☑	☑
LOCAL_OUT4	Default tag table	Bool	%Q0.4	☐	☑	☑	☑
LOCAL_OUT5	Default tag table	Bool	%Q0.5	☐	☑	☑	☑

Phoenix Contact Reference Device Input Tags

AXIOLINE_IN0	Default tag table	Bool	%I1.0	☐	☑	☑	☑
AXIOLINE_IN1	Default tag table	Bool	%I1.1	☐	☑	☑	☑
AXIOLINE_IN2	Default tag table	Bool	%I1.2	☐	☑	☑	☑
AXIOLINE_IN3	Default tag table	Bool	%I1.3	☐	☑	☑	☑
AXIOLINE_IN4	Default tag table	Bool	%I1.4	☐	☑	☑	☑
AXIOLINE_IN5	Default tag table	Bool	%I1.5	☐	☑	☑	☑
AXIOLINE_IN6	Default tag table	Bool	%I1.6	☐	☑	☑	☑
AXIOLINE_IN7	Default tag table	Bool	%I1.7	☐	☑	☑	☑

Phoenix Contact Reference Device Output Tags

AXIOLINE_OUT0	Default tag table	Bool	%Q1.0	☐	☑	☑	☑
AXIOLINE_OUT1	Default tag table	Bool	%Q1.1	☐	☑	☑	☑
AXIOLINE_OUT2	Default tag table	Bool	%Q1.2	☐	☑	☑	☑
AXIOLINE_OUT3	Default tag table	Bool	%Q1.3	☐	☑	☑	☑
AXIOLINE_OUT4	Default tag table	Bool	%Q1.4	☐	☑	☑	☑
AXIOLINE_OUT5	Default tag table	Bool	%Q1.5	☐	☑	☑	☑
AXIOLINE_OUT6	Default tag table	Bool	%Q1.6	☐	☑	☑	☑
AXIOLINE_OUT7	Default tag table	Bool	%Q1.7	☐	☑	☑	☑

Lattice Evaluation Board RT-LABS Input Tags

RT-LABS_IN0	Default tag table	Bool	%I2.0	☐	☑	☑	☑
RT-LABS_IN1	Default tag table	Bool	%I2.1	☐	☑	☑	☑
RT-LABS_IN2	Default tag table	Bool	%I2.2	☐	☑	☑	☑
RT-LABS_IN3	Default tag table	Bool	%I2.3	☐	☑	☑	☑
RT-LABS_IN4	Default tag table	Bool	%I2.4	☐	☑	☑	☑
RT-LABS_IN5	Default tag table	Bool	%I2.5	☐	☑	☑	☑
RT-LABS_IN6	Default tag table	Bool	%I2.6	☐	☑	☑	☑
RT-LABS_IN7	Default tag table	Bool	%I2.7	☐	☑	☑	☑

Lattice Evaluation Board RT-LABS Output Tags

RT-LABS_OUT0	Default tag table	Bool	%Q2.0	☐	☑	☑	☑
RT-LABS_OUT1	Default tag table	Bool	%Q2.1	☐	☑	☑	☑
RT-LABS_OUT2	Default tag table	Bool	%Q2.2	☐	☑	☑	☑
RT-LABS_OUT3	Default tag table	Bool	%Q2.3	☐	☑	☑	☑
RT-LABS_OUT4	Default tag table	Bool	%Q2.4	☐	☑	☑	☑
RT-LABS_OUT5	Default tag table	Bool	%Q2.5	☐	☑	☑	☑
RT-LABS_OUT6	Default tag table	Bool	%Q2.6	☐	☑	☑	☑
RT-LABS_OUT7	Default tag table	Bool	%Q2.7	☐	☑	☑	☑

Figure 7.15. Siemens TIA Portal PLC Tags Setup for Demo

7.8.13. Setting Up the PLC Program Block in Main [OB1]

The test program on this page enables to proof the basic functionality of the PROFINET I/O communication between the I/O Controller Device (PLC) and the connected I/O devices.

Testing Digital Outputs on PMOD0 connector of the RT-LABS I/O device implementation on the Lattice Evaluation Board, utilizing the Digital Inputs on the PHOENIX CONTACT AXIOLINE Reference I/O device.

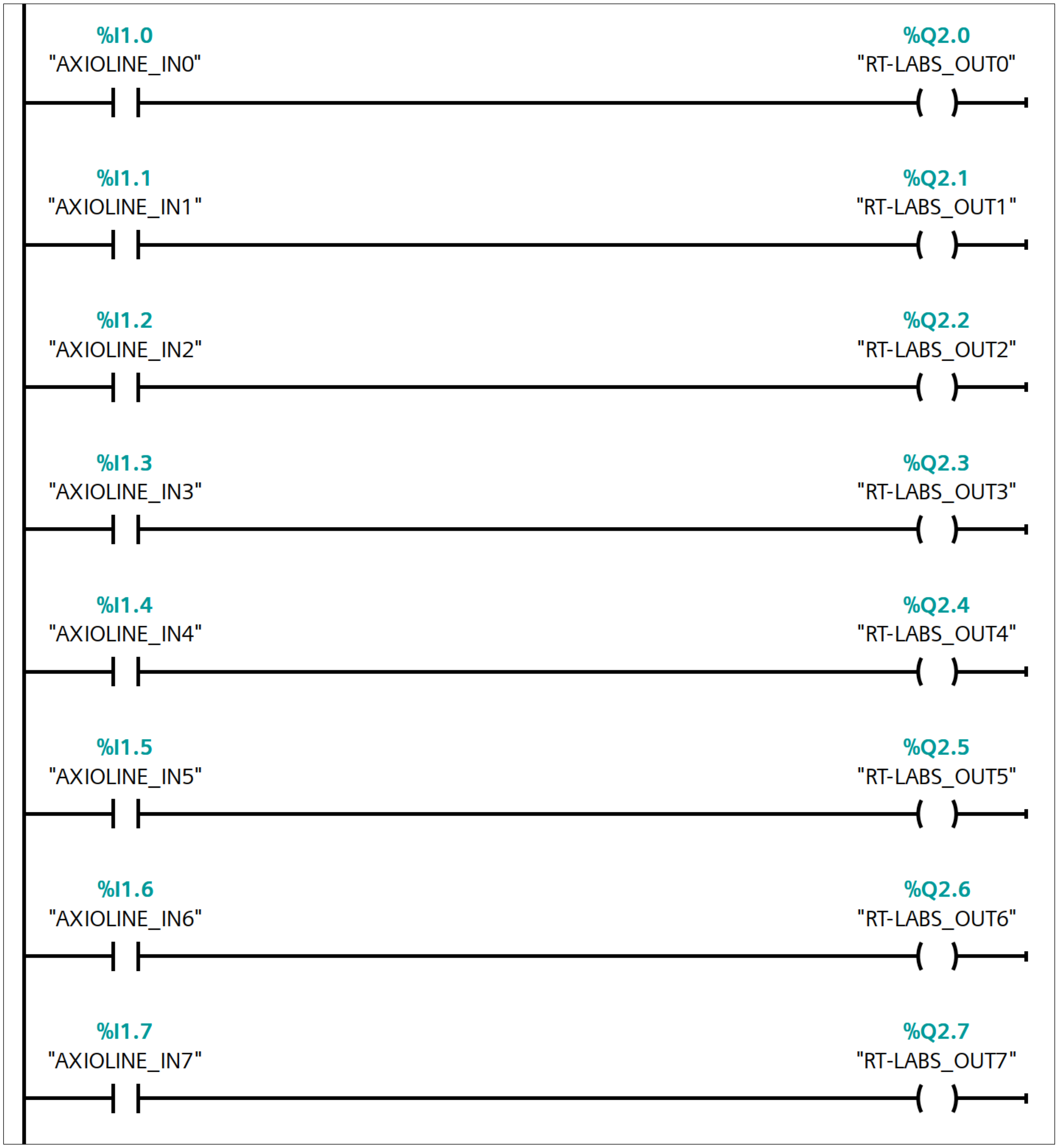

Figure 7.16. Siemens TIA Portal Main Block Setup for Demo for Output

Testing Digital Inputs on the PMOD1 connector of the RT-LABS I/O Device Implementation on the Lattice Evaluation Board, utilizing the Digital Outputs on the PHOENIX CONTACT AXIOLINE Reference I/O device.

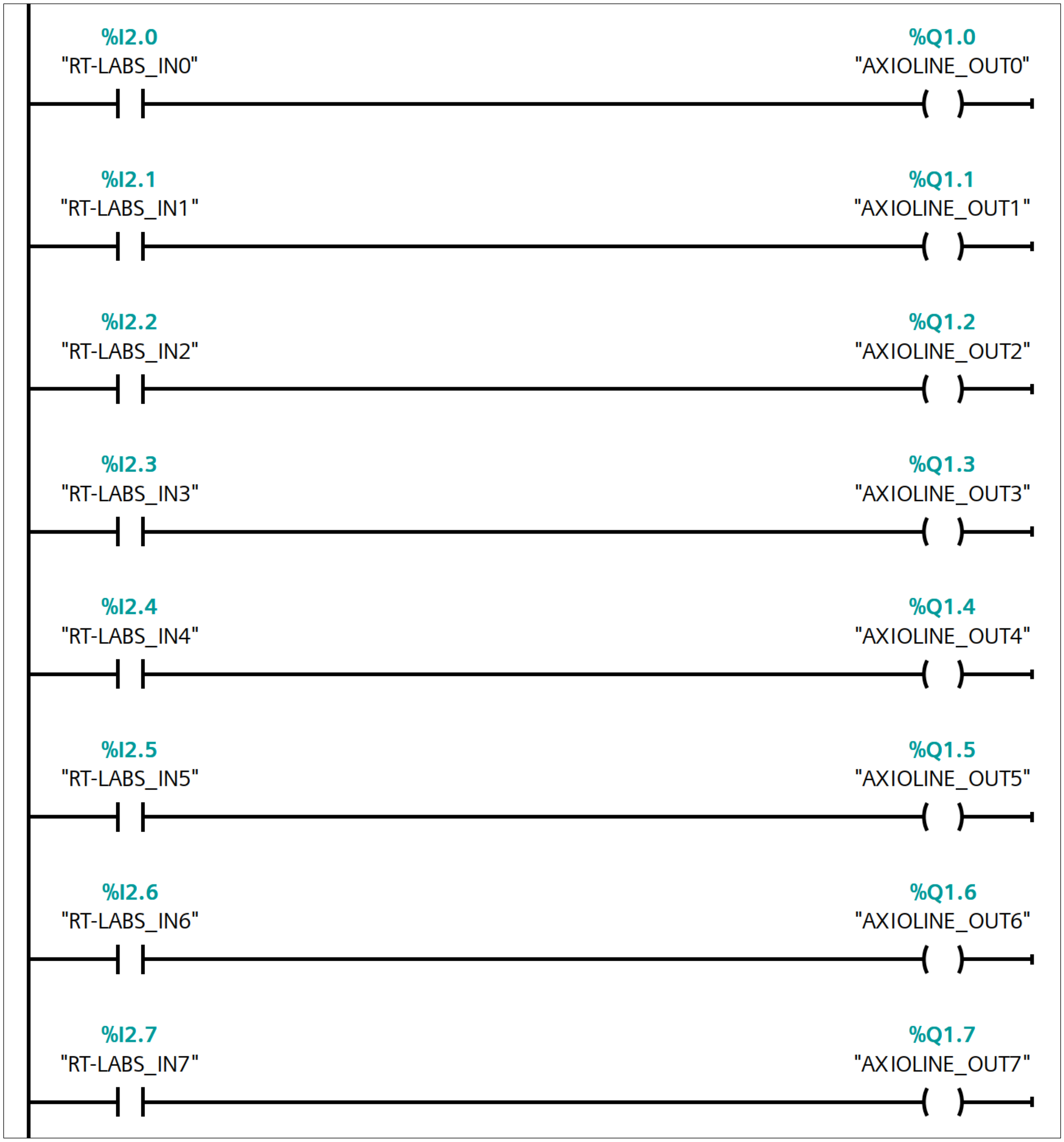

Figure 7.17. Siemens TIA Portal Main Block Setup for Demo for Input

7.8.14. Compiling and Loading the PROFINET System Configuration

This step is mandatory to setup the PLC device with all configured settings and the PLC program to run the demo and must be repeated after any changes.

To compile and load the PROFINET system configuration, perform the following:

1. In the *Project tree – Devices* window, select **plc_1**.
2. Click the **Compile** button with this icon  in the TIA Portal top menu toolbar. The system configuration is processed and then it is ready to be downloaded to the PLC device.
3. Click **Download to device** button with this icon  in the top menu toolbar.
4. The system configuration is downloaded to the PLC device.
5. Follow the instructions and check the status of the messages within the Load results windows.

7.8.15. Checking the System Status Online

To check the system status online, perform the following:

1. In the Project tree - Devices window, double-click on the **plc_1 - Online and diagnostics**. The **plc_1[CPU 1211C DC/DC/DC] - Online access** windows appear.
2. Select the interface/subnet **PN/IE_1** and the correct **Ethernet Adapter** of your PC Interface, and then click **Go online**.
3. If the system setup is successful, the screen shows your PC is successfully connected (online) with the Siemens PLC and the I/O devices. For testing, go offline and try again to go online.

7.9. I/O Cycle Timing Constraints

As the implementation is fully software based, the optimal setting for an I/O cycle is 8 ms with 10 watchdog time outs. The current performance is limited at 10 watchdog timeouts or 80 ms for 8 ms I/O. Any value lower than 10 watchdog timeouts can cause the connection to the PLC to be unstable as the RISC-V CPU is capped at 100 MHz on the LFCPNX-100 FPGA device.

8. Resource Utilization

The resource utilization for the ADI MCB is shown in the tables below.

Table 8.1. Main System Resource Utilization

	LUT4 Logic	LUT4 Distributed RAM	LUT4 Ripple Logic	PFU Registers	I/O Registers	I/O Buffers	DSP MULT	EBR	Large RAM
soc_golden_system	44794(1)	13908(0)	7896(0)	41918(4)	22(4)	118(49)	26(0)	128(0)	4(0)
adin1200_mii2_inst	3458(0)	4608(0)	546(0)	1943(0)	0(0)	0(0)	0(0)	4(0)	0(0)
adin1200_mii2_wrapper_inst	3458(12)	4608(0)	546(0)	1943(8)	0(0)	0(0)	0(0)	4(0)	0(0)
U1	3446(0)	4608(0)	546(0)	1935(0)	0(0)	0(0)	0(0)	4(0)	0(0)
mac_phy_top_inst	3446(0)	4608(0)	546(0)	1935(0)	0(0)	0(0)	0(0)	4(0)	0(0)
genblk1.tse_mac	3446(3446)	4608(4608)	546(546)	1935(1935)	0(0)	0(0)	0(0)	4(4)	0(0)
ahbl_interconnect_inst	87(0)	0(0)	0(0)	6(0)	0(0)	0(0)	0(0)	0(0)	0(0)
lscc_ahbl_interconnect_inst	87(0)	0(0)	0(0)	6(0)	0(0)	0(0)	0(0)	0(0)	0(0)
ahbl_lite_bus.u_lscc_ahbl_bus	87(0)	0(0)	0(0)	6(0)	0(0)	0(0)	0(0)	0(0)	0(0)
u_lscc_ahbl_decoder	4(2)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
genblk1[0].genblkl.u_lscc_ahbl_decoder_sel	2(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
genblk1[0].u_lscc_ahbl_decoder_comp	2(2)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
u_lscc_ahbl_default_slv	10(10)	0(0)	0(0)	2(2)	0(0)	0(0)	0(0)	0(0)	0(0)
u_lscc_ahbl_multiplexor	73(73)	0(0)	0(0)	4(4)	0(0)	0(0)	0(0)	0(0)	0(0)
apb_interconnect0_inst	88(0)	0(0)	0(0)	6(0)	0(0)	0(0)	0(0)	0(0)	0(0)
lscc_apb_interconnect_inst	88(0)	0(0)	0(0)	6(0)	0(0)	0(0)	0(0)	0(0)	0(0)
apb_bus.u_lscc_apb_bus	88(0)	0(0)	0(0)	6(0)	0(0)	0(0)	0(0)	0(0)	0(0)
u_lscc_apb_decoder	24(15)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
genbik1[0].u_Isec_apb_decoder_sel	5(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
genblk1[0].u_Isec_apb_decoder_comp	5(5)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
genbiki[i].u_Isec_apb_decoder_sel	3(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
genblic1[0].u_lsec_apb_decoder_comp	3(3)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
genbiki[3].u_Isec_apb_decoder_sel	1(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
genblic1[0].u_Isec_apb_decoder_comp	1(1)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
u_lsee_apb_multiplexor	64(64)	0(0)	0(0)	6(6)	0(0)	0(0)	0(0)	0(0)	0(0)
axi2ahbl_inst	524(0)	0(0)	44(0)	206(0)	0(0)	0(0)	0(0)	0(0)	0(0)
Isec_axilahb_lite_inst	524(524)	0(0)	44(44)	206(206)	0(0)	0(0)	0(0)	0(0)	0(0)

Table 8.2. PROFINET Total Resource Utilization

	LUT4 Logic	LUT4 Distributed RAM	LUT4 Ripple Logic	PFU Registers	I/O Registers	I/O Buffers	DSP MULT	EBR	Large RAM
soc_golden_system	44794(1)	13908(0)	7896(0)	41918(4)	22(4)	118(49)	26(0)	128(0)	4(0)

9. Debugging

This section describes the debugging methods to use for PROFINET on the ADI MCB. For code debugging tools, refer to relevant Lattice Propel user guide.

The Wireshark analyzer is the recommended tool to debug the PROFINET packet format and implementation.

9.1. Packet Inspection

Use the Wireshark analyzer tool to inspect PROFINET packets. You can download the Wireshark tool from https://www.wireshark.org/download.html.

Perform the following steps to run packet inspection:

1. Open the Wireshark tool and select the network (Ethernet).
2. Click on the Ethernet network.
3. Click on the **Run** ( ) button.

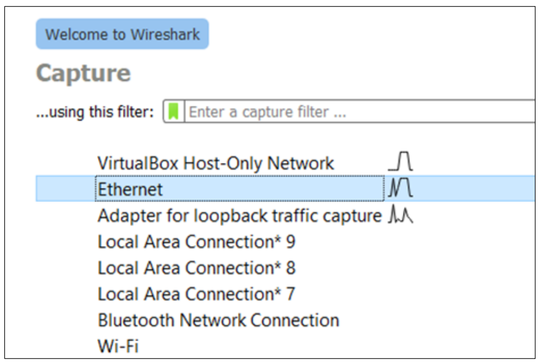

Figure 9.1. Wireshark Tool: Ethernet Selection

Figure 9.2 shows the PNIO-CM packet format**.**

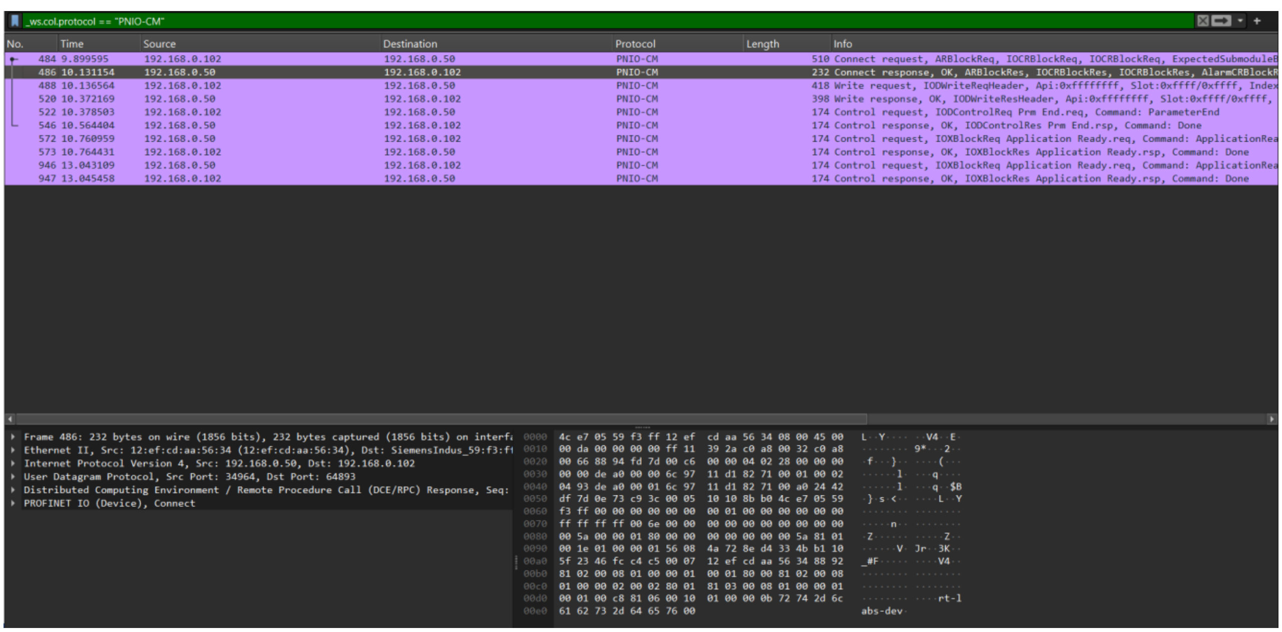

Figure 9.2. Wireshark Tool Filter _ws.col.protocol == "PNIO"

Figure 9.3 shows the PNIO-DCP packet format.

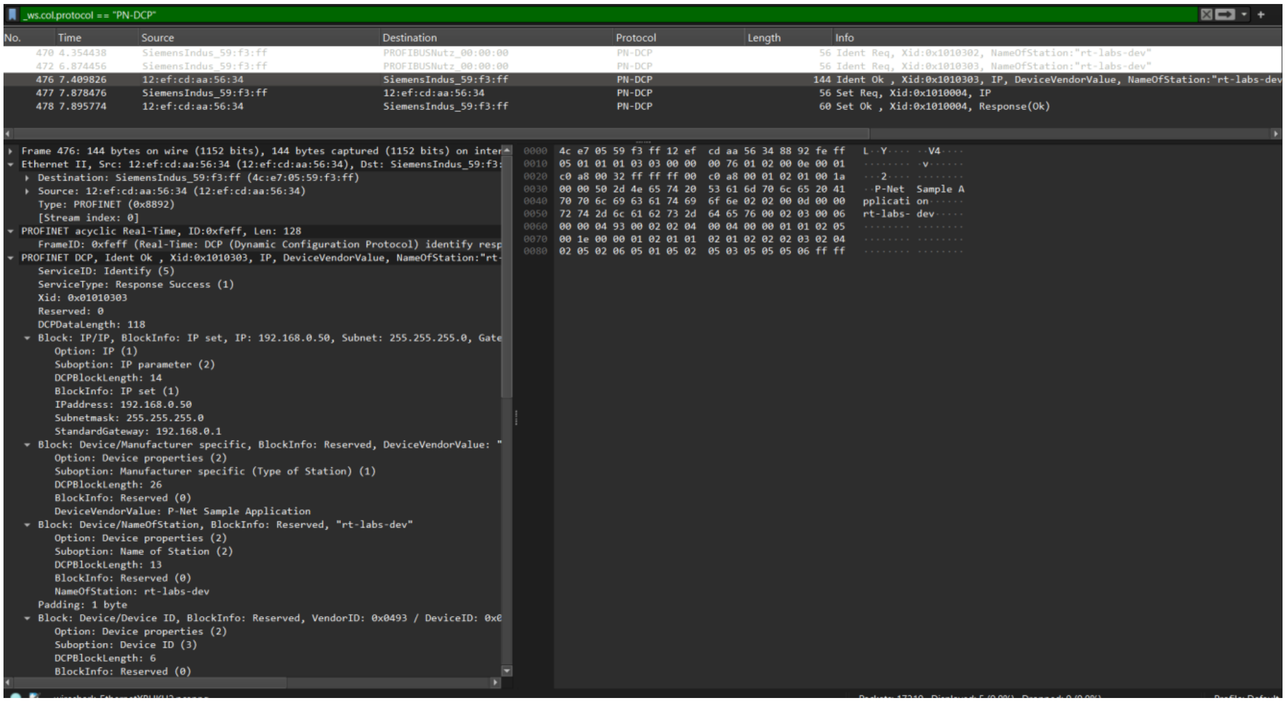

Figure 9.3. Wireshark Tool Filter _ws.col.protocol == "PN-DCP"

Figure 9.4 shows the LLDP packet format.

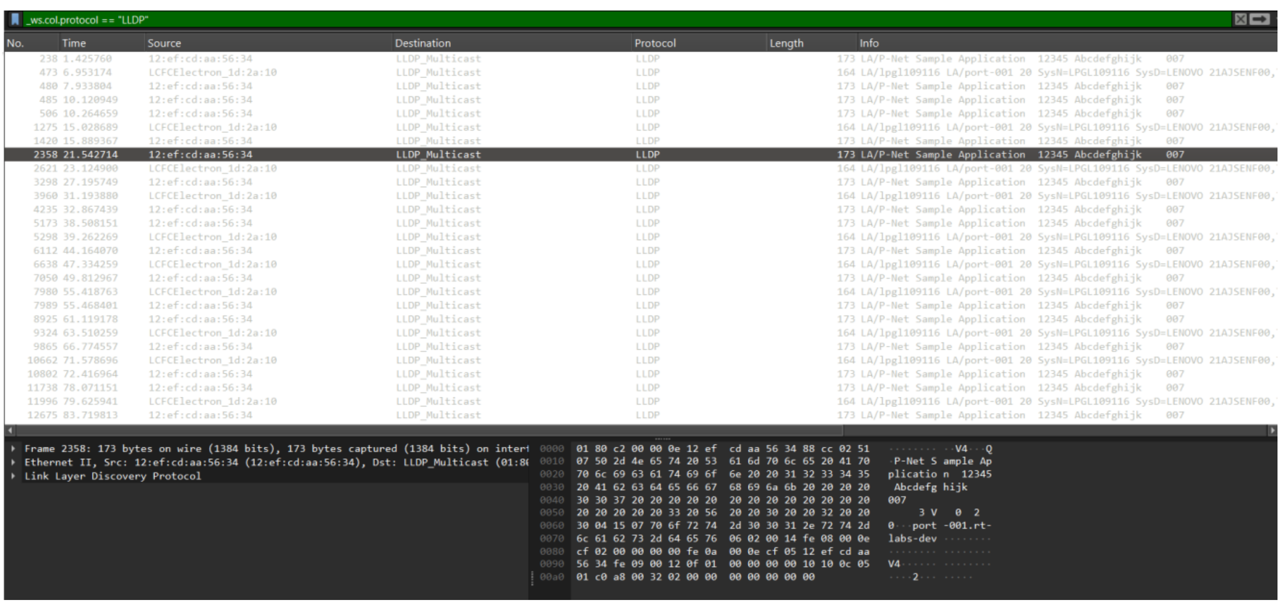

Figure 9.4. Wireshark Tool Filter " _ws.col.protocol == "LLDP"

Figure 9.5 shows the PNIO-PS packet format.

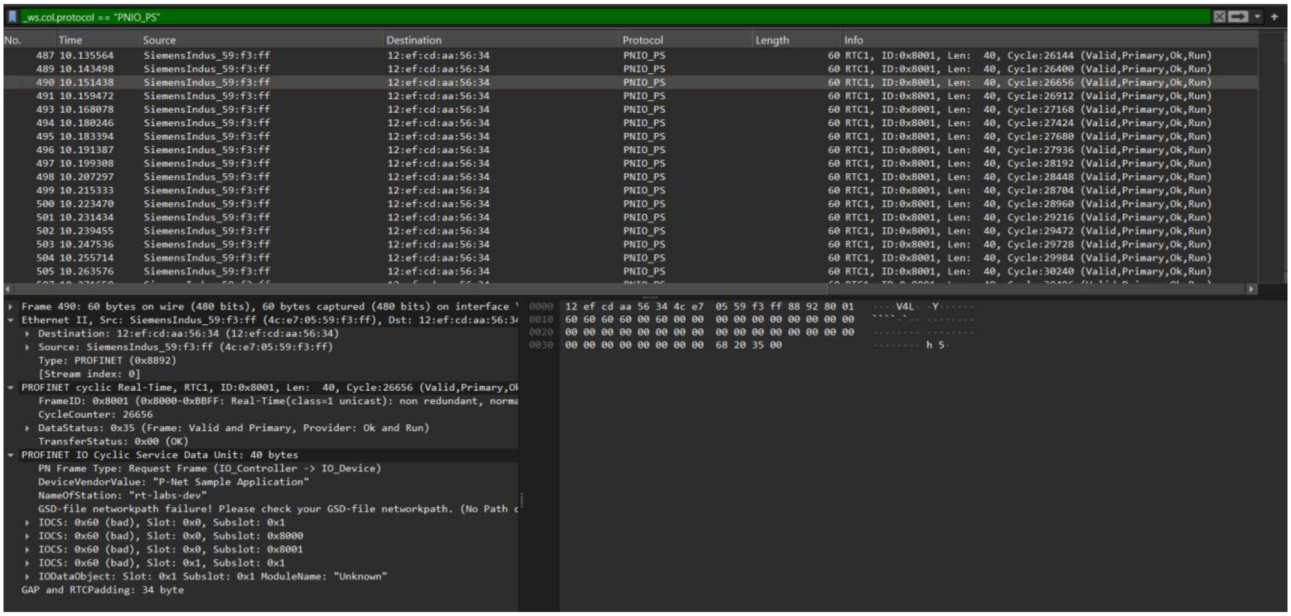

Figure 9.5. Wireshark Tool Filter " _ws.col.protocol == "PNIO-PS"

Figure 9.6 shows the PNIO packet format.

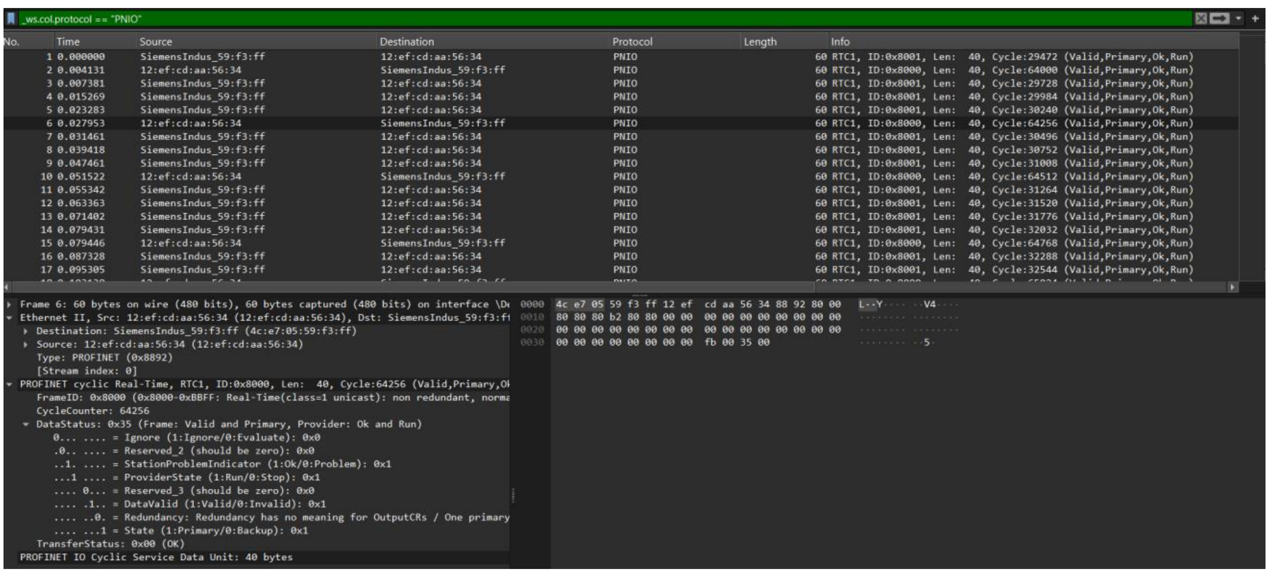

Figure 9.6. Wireshark Tool Filter " _ws.col.protocol == "PNIO-PS"

References

- Golden System Reference Design User Guide (FPGA-RD-02283)
- Golden System Reference Design Demo User Guide (FPGA-UG-2205)
- SGDMA Controller IP Core User Guide (FPGA-IPUG-02131)
- Tri-Speed Ethernet MAC IP User Guide (FPGA-IPUG-02084)
- CertusPro-NX web page
- PROFINET Specification page
- RT-Labs PROFINET Stack
- Lattice Radiant FPGA design software
- Lattice Solutions Reference Designs web page
- Lattice Solutions IP Cores web page
- Lattice Propel Design Environment web page
- Lattice Radiant Software User Guide
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, please refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.0, January 2025

Section	Change Summary
All	Initial release.

www.latticesemi.com