TI Designs

Multi-Protocol Industrial Ethernet Detection With PRU-ICSS for Industrial Automation

Design Overview

Industrial Ethernet for industrial automation applications exist in more than 20 industrial standards. Some of the well-established, real-time Ethernet protocols, such as EtherCAT®, EtherNet/IP™, PROFINET, Sercos III, and PowerLink require dedicated MAC hardware support in terms of field programmable gain arrays (FPGAs) or application specific integrated circuits (ASICs). The Programmable Real-time Unit and the Industrial Communication Subsystem (PRU-ICSS), which exists as a hardware block inside TI's Sitara™ family of processors, replaces FPGAs or ASICs with a single-chip solution. A firmware in the PRU-ICSS allows detection of the type of industrial Ethernet protocol and loads the appropriate industrial application during run-time into the Sitara processor. This TI design describes the multi-protocol, Industrial Ethernet detection firmware for the PRU-ICSS.

Design Resources

TIDEP0032	Tool Folder Containing Design Files
TMDSICE3359	Tool Folder
AM3359	Product Folder
TIDEP0001	Tool Folder
TIDEP0003	Tool Folder
TIDEP0008	Tool Folder
TIDEP0010	Tool Folder
TIDEP0028	Tool Folder

TI E2E™ Community

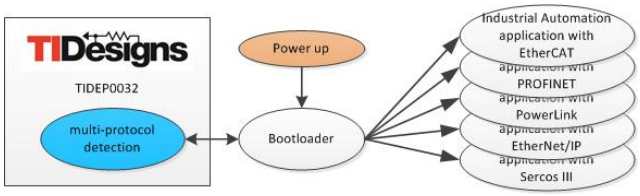
ASK Our E2E Experts
WEBENCH® Calculator Tools

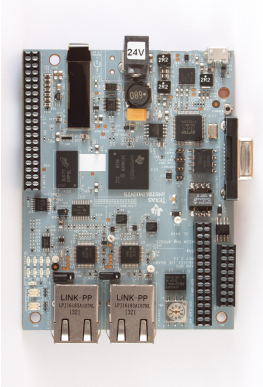
Design Features

- Multi-Protocol Industrial Ethernet Detection Firmware for PRU-ICSS
- Supports Detection of the Top Industrial Ethernet Protocol Standards Including EtherCAT, EtherNet/IP, PROFINET, Sercos III, and PowerLink.
- Source Code for Multi-Protocol Industrial Ethernet Detection Firmware for PRU-ICSS Available for Customization and Adding More Industrial Ethernet Protocol Standards
- PRU-ICSS Firmware is Tested on TI's Multi-Protocol Demonstrator Panel (Shown at Electronica and SPS IPC Drives Trade Show) and Contains Firmware Source Code and Design Implementation

Featured Applications

- Programmable Logic Control Systems (PLC)
- Industrial Drives
- Industrial Sensors and I/O Modules
- Industrial Communication Gateways





1 Key System Specifications

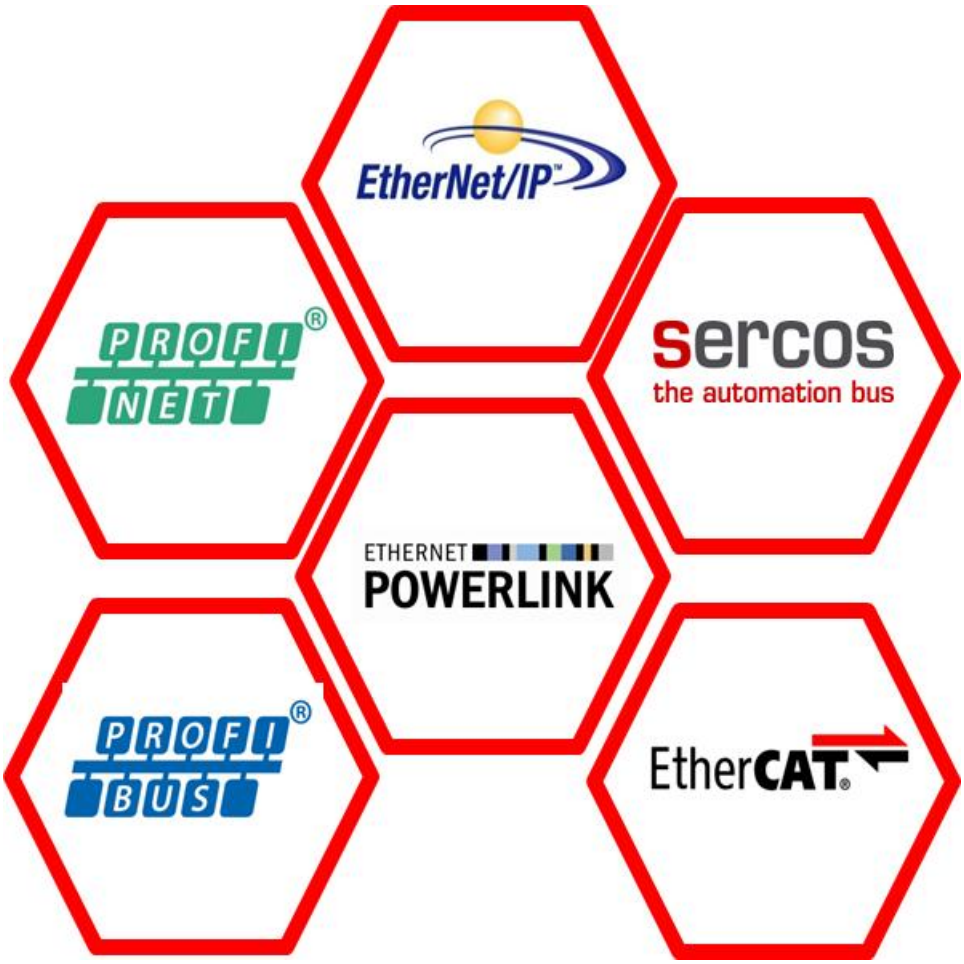


Figure 1. Multi-Protocol Detection of Various Industrial Ethernet Standards With PRU-ICSS

At the time of this writing, more than 20 Industrial Ethernet protocols have been defined as standards and can be found in factories equipped with industrial automation solutions. Table 1 lists the Industrial Ethernet protocols that are supported by this TI design using the multi-protocol detection firmware for PRU-ICSS. This TI design also allows customers to differentiate their products by adding the option to detect additional Industrial Ethernet standards by writing the PRU-ICSS firmware source code.

Table 1. Real-Time Ethernet Types and Detection Methods

PROTOCOL NAME	DETECTION TYPE	PROTOCOL ID (STORED IN SHARED MEMORY)
EtherCAT	EtherType field	0xA488
PROFINET	EtherType field	0x9288
Sercos III	EtherType field	0xCD88
PowerLink	EtherType field	0xAB88
EtherNet/IP	EtherType field and MAC address match	0x0002 and 0xE49069*****

2 System Description

New types of multi-core embedded processors and systems-on-a-chip (SoC) that feature profoundly greater programmability are changing the adopted strategies that many manufacturers of industrial networking equipment use for their product lines. Teaming these new processors with software-based protocol execution can provide a scalable platform approach. Such a flexible architecture is capable of supporting multiple industrial communications protocols across a wide range of system types. In the process, faster, more powerful processing capabilities are delivered to users of industrial networking equipment while manufacturers reduce their system costs, simplify their assembly processes, optimize feature differentiation, and streamline field support.

2.1 Deterministic Industrial Ethernet Protocols

Each of the Industrial Ethernet (or real-time Ethernet) protocols has its own benefits within the industrial automation and control market.

Most of the slave devices require specific hardware support because the standard hardware for Ethernet medium access control (MAC)cannot handle the real-time constraints. Some features, such as on-the-fly processing, are required to maintain a low latency of Ethernet frames. This on-the-fly processing means that while the MAC hardware is still receiving the frame, the protocol-specific MAC extracts or inserts bytes (or both) from the stream of incoming bytes and then later recalculates and attaches the new Ethernet frame checksum.

The following Table 2 provides some information about the most prominent industrial Ethernet protocols that include EtherCAT®, EtherNet/IP™, PROFINET, POWERLINK, and Sercos III.

Table 2 also lists the organization that keeps the protocol up-to-date in terms of specification updates or testing requirements. View more information about the protocols at each organization's website.

Table 2. Industrial Ethernet Standards

CRITERIA	PROFINET	PowerLink	EtherNet/IP	EtherCAT	Sercos III
Organization	PNO	EPSG	ODVA	ETG	SERCOS International
www.	profibus.com	ethernet-powerlink.org	odva.org	ethercat.org	sercos.org

2.1.1 EtherCAT

EtherCAT is a MAC layer protocol that is transparent to higher-level protocols like TCP/IP. EtherCAT has been executed mostly in ASICs and FPGAs. EtherCAT adds minimum overhead to data transmissions to ensure low-latency slave response times and supports as many as 65,535 nodes on an industrial Ethernet network.

For more details on EtherCAT technology with TI processors visit the TIDEP0001 tool folder.

2.1.2 EtherNet/IP

EtherNet/IP is an application-layer protocol that executes on top of TCP/IP and employs Common industrial Protocol (CIP) for industrial networks. An unlimited number of nodes can be supported in an EtherNet/IP network because standard Ethernet and Ethernet switches are utilized by EtherNet/IP.

For more details on EtherNet/IP technology with TI processors visit the TIDEP0003 tool folder.

2.1.3 PROFINET

PROFINET has three performance classes that provide different throughput speeds and various degrees of low latency response times. Network topologies that PROFINET support are usually branch and star architectures.

For more details on PROFINET technology with TI processors visit the TIDEP0008 tool folder.

2.1.4 PowerLink

PowerLink employs a polling and time-slicing mechanism to ensure real-time network performance. PowerLink operates on top of IEEE 802.3 Ethernet so that it supports a variety of network topologies, such as cross connect and hot plug.

For more details on PowerLink technology with TI processors visit the TIDEP0028 tool folder.

2.1.5 Sercos III

This is the third generation of Serial Real-time Communication System (Sercos). Sercos III supports ring and line network topologies. In ring network topology, Sercos III provides redundancy to the extent that the failure of one slave does not disable communications to the other slave nodes on the network.

For more details on Sercos III technology with TI processors visit the TIDEP0010 tool folder.

2.2 Practical Uses for Multi-Protocol Industrial Ethernet Detection

At the time of this writing, products for industrial automation solutions are developed by manufacturers for one specific industrial Ethernet protocol. The devices are programmed at the manufacturer's assembly line for that specific industrial Ethernet protocol. Due to hardware constraints (ASIC/FPGA), most of the products for industrial automation solutions do not support multi-protocol industrial Ethernet and cannot load the industrial Ethernet protocol on-the-fly.

PRU-ICSS-based products allow the support of multi-protocol industrial Ethernet applications, such as the industrial Ethernet firmware, which is loaded during device startup.

Figure 2 shows the execution flow of a multi-protocol industrial Ethernet detection with a PRU-ICSS-enabled device from power-up. First the bootloader is loaded, which performs initialization of the system peripherals like memory interfaces, system PLLs, general purpose input and output (GPIOs) pins, and others. Because the bootloader must determine the type of industrial Ethernet standard that it receives from the Ethernet frame, the bootloader loads the multi-protocol industrial Ethernet detection application (TIDEP0032) first. The PRU-ICSS firmware analyzes the incoming Ethernet frames on the real-time Ethernet ports. As soon as the industrial Ethernet standard has been determined, the protocol type stores in non-volatile memory and the bootloader restarts, again. Following this restart, the bootloader reads the non-volatile memory and therefore knows the type of industrial Ethernet standard. The bootloader can then load the appropriate industrial automation application with the appropriate industrial Ethernet support (such as EtherCAT, PROFINET, PowerLink, EtherNet/IP, or Sercos III).

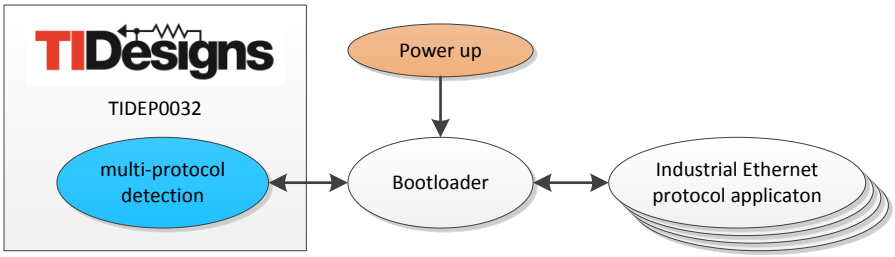


Figure 2. TI Design TIDEP0032 Multi-Protocol Industrial Ethernet With PRU-ICSS

The TIDEP0032 TI design covers the multi-protocol detection firmware for PRU-ICSS. The design describes the reception of Ethernet frames and elaborates on the detection of the different types of industrial Ethernet standards. The design also explains how to add additional detection methods for other industrial Ethernet standards.

2.3 PRU-ICSS Peripheral

2.3.1 PRU-ICSS System Block Diagram

The PRU-ICSS consists of two programmable RISC cores with dedicated RAM and ROM (see Figure 3). The RISC cores operate at 200 MHz with a single-cycle instruction execution (5 ns) that enables real-time deterministic programming. For more information on the PRU-ICSS, view the TI Wiki page at http://processors.wiki.ti.com/index.php/PRU-ICSS.

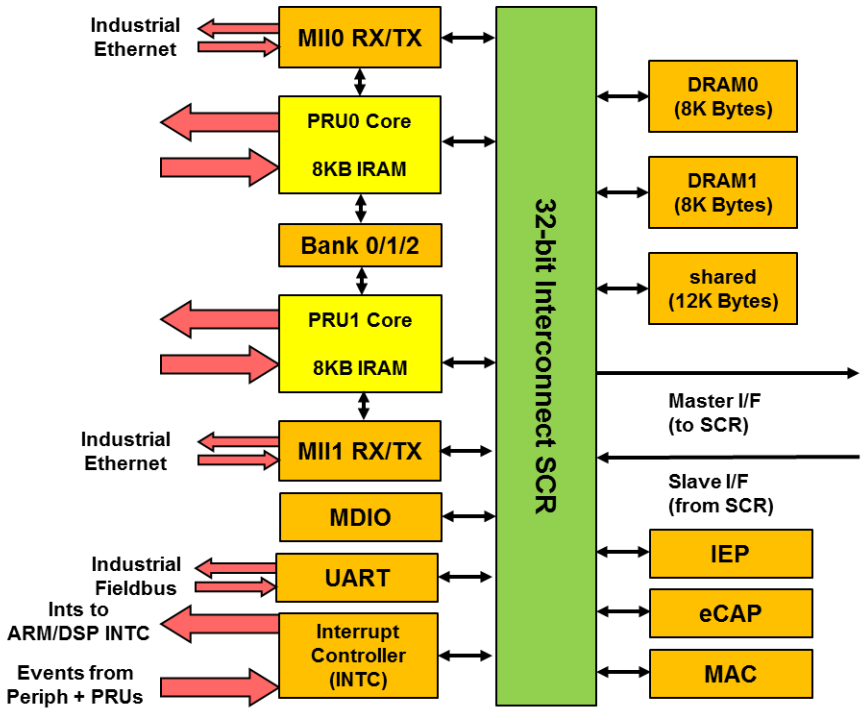


Figure 3. PRU-ICSS System Block Diagram in AM335x Processors

2.3.2 MII Interfaces

The PRU-ICSS has two dedicated media-independent interfaces (MIIs) to receive Ethernet frames. Each PRU has direct access to one of the MII blocks. The multi-protocol detection firmware uses the “RX L2” mode to receive Ethernet frames within a 64-byte circular ring buffer. While the MII interface block writes into this ring buffer, the PRU must read the buffer in chunks of 32-bytes before the write pointer begins to overwrite the data in the circular L2 receive buffer. Because the PRU executes instructions in a deterministic real-time manner (5 ns per instruction), the following Equation 1 and Equation 2 are used to calculate the instructions cycle budget. These calculations assume that the user implements a 100-Mbps Ethernet, where one byte is received every 80 ns.

64 bytes × 80 ns per byte = 5.12 µs (1)

5.12 µs / (5 ns per PRU instruction) = 1024 PRU instructions (2)

As the user can calculate, the PRU can execute 1024 instructions before the RX L2 buffer begins to wrap around, which overwrites the already-received bytes of the frame. A value of 32 bytes is received after 2.56 µs or 512 PRU instructions.

The PRU handles 32 bytes of the Ethernet frames simultaneously. To handle this amount, the XIN assembly instruction maps 32 bytes of RX L2 data into register R2 through R9 in a single PRU instruction cycle. The PRU then stores this 32-byte data into the shared memory to assemble the chunks of 32-byte Ethernet frame fragments. This process repeats until the Ethernet frame has been completely received; only then does the PRU begin analyzing the received Ethernet frame to determine the type of industrial Ethernet standard.

2.3.3 PRU RAM and Shared Memory

The PRU-ICSS has 12KB of shared memory in addition to 8KB per PRU core. The PRU0 local RAM is used to store a complete Ethernet frame. As soon as the Ethernet frame has been analyzed and the type of industrial Ethernet standard has been determined, the protocol type is stored in the shared memory. The ARM application only uses the protocol type for further processing.

2.3.4 PRU-ICSS Interrupt Controller

The PRU issues an event through the interrupt controller (INTC) as soon as the industrial Ethernet standard has been detected by the PRU-ICSS firmware. The ARM interrupt controller receives this event as an interrupt. After sending the event, the PRU continues to analyze Ethernet frames while the PRU updates the PRU RAM location continuously after analyzing every frame. The ARM specifically must decide how many multiple events of the same type of industrial Ethernet standard can be received before the application restarts the bootloader. Figure 4 shows the INTC configuration. System event 18 (SYS_EVENT_18) is mapped to Channel-2, which leads into the Host-2 event. The Host-2 event is then received by the ARM interrupt controller.

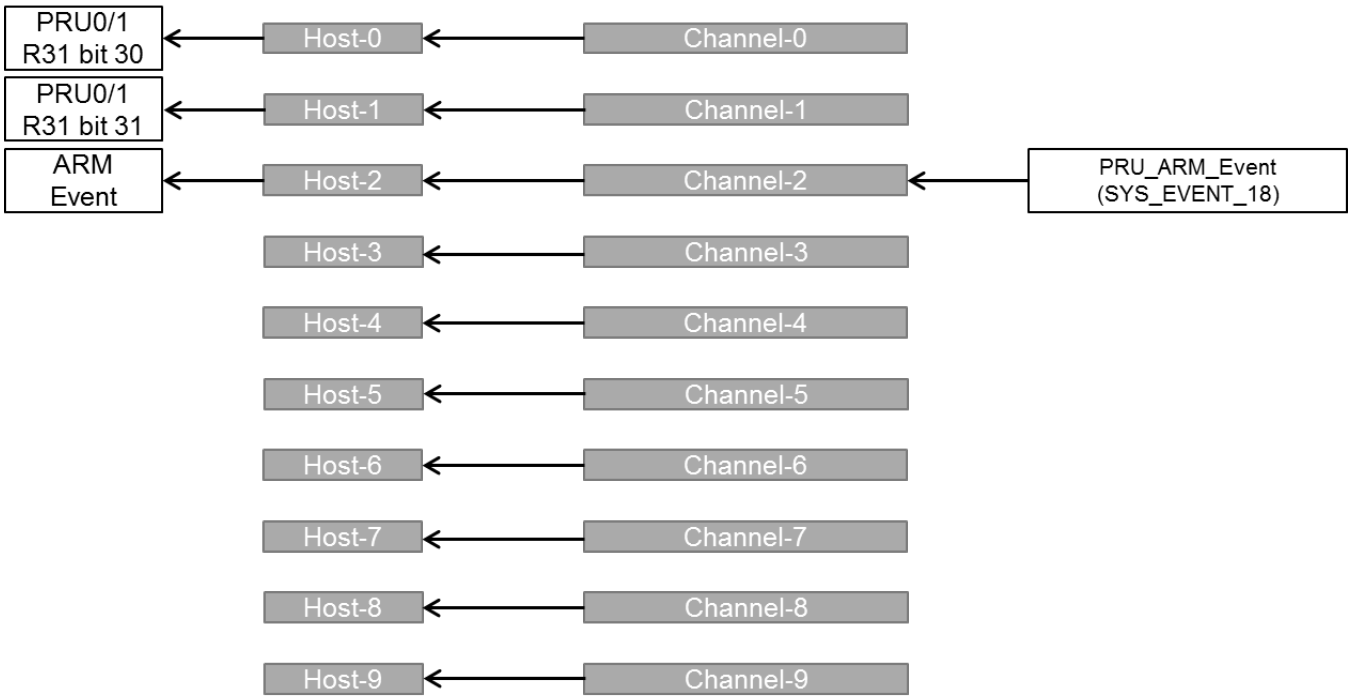


Figure 4. INTC Configuration in PRU-ICSS

2.3.5 PRU Assembler Programming

The PRU-ICSS firmware is programmed using an RISC assembler for this TI design. This programming method improves the implementation feasibility for real-time critical applications. Note that TI's Code Composer Studio™ (CCSv6) also allows the user to program the PRU in "C" language; however, the "C" language is not used for this TI design to meet the real-time constraints.

View the *PRU Assembly Instructions* manual on the TI Wiki: http://processors.wiki.ti.com/index.php/PRU_Assembly_Instructions.

3 Block Diagram

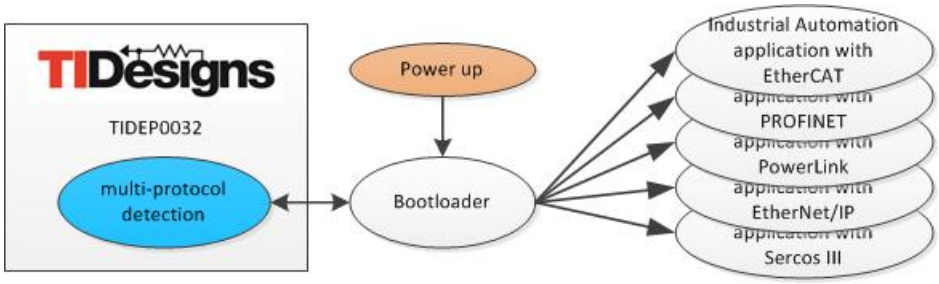


Figure 5. Multi-Protocol Industrial Ethernet Detection With PRU-ICSS

3.1 Highlighted Products

3.1.1 AM335x Processor

Up to 1-GHz Sitara™ARM® Cortex®-A8 32-bit RISC processor

- NEON™ SIMD coprocessor
- 32KB of L1 Instruction and 32KB of data cache with single-error detection (parity)
- 256KB of L2 cache with error correcting code (ECC)
- 176KB of on-chip boot ROM
- 64KB of dedicated RAM
- Emulation and debug - JTAG
- Interrupt controller (up to 128 interrupt requests)

Programmable Real-Time Unit Subsystem and Industrial Communication Subsystem (PRU-ICSS)

- Supports protocols such as EtherCAT®, PROFIBUS, PROFINET, EtherNet/IP™, and more
- Two programmable real-time units (PRUs)
- 32-bit load and store RISC processor capable of running at 200 MHz
- 8KB of instruction RAM with single-error detection (parity)
- 8KB of data RAM with single-error detection (parity)
- Single-cycle 32-bit multiplier with 64-bit accumulator
- Enhanced GPIO module provides shift-in or shift-out support and parallel latch on external signal
- 12KB of shared RAM with single-error detection (parity)
- Three 120-byte register banks accessible by each PRU
- INTC for handling system input events
- Local interconnect bus for connecting internal and external masters to the resources inside the PRU-ICSS
- Peripherals inside the PRU-ICSS:
  - One universal asynchronous receiver and transmitter (UART) port with flow control pins, supports up to 12 Mbps
  - One enhanced capture (eCAP) module
  - Two MII Ethernet ports that support industrial Ethernet, such as EtherCAT
  - One management data input and output (MDIO) port

On-chip memory (shared L3 RAM)

- 64KB of general-purpose on-chip memory controller (OCMC) RAM
- Accessible to all masters

External memory interfaces (EMIF)

- mDDR(LPDDR), DDR2, DDR3, and DDR3L controller:
  - mDDR: 200-MHz clock (400-MHz data rate)
  - DDR2: 266-MHz clock (532-MHz data rate)
  - DDR3: 400-MHz clock (800-MHz data rate)
  - DDR3L: 400-MHz clock (800-MHz data rate)
  - 16-bit data bus
  - 1GB of total addressable space
  - Supports one x16 or two x8 memory device configurations
- General-purpose memory controller (GPMC)
  - Flexible 8-bit and 16-bit asynchronous memory interface with up to seven chip selects (NAND, NOR, Muxed-NOR, or SRAM)
  - Uses BCH code to support 4-, 8-, or 16-bit ECC
  - Uses hamming code to support 1-bit ECC

3.1.2 TMDSICE3359 Industrial Communication Engine EVM

Hardware specification

- AM3359 ARM Cortex-A8
- DDR3, NOR flash, and serial peripheral interface (SPI) flash
- Organize light-emitting diode (OLED) display
- TPS65910 power management
- 24-V power supply
- USB cable for JTAG interface and serial console

Software and tools

- SYS/BIOS real-time operating system (OS)
- Starterware base port
- TI's Code Composer Studio™ (CCS) integrated development environment (IDE)
- Application stack for industrial communication protocols
- Sample industrial applications

Connectivity

- PROFIBUS interface
- CANOpen
- EtherNet/IP
- PROFINET
- Sercos III
- Digital inputs and outputs (I/O)
- SPI
- UART
- JTAG

See the TMDSICE3359 website for complete list of features and design resources: www.ti.com/tool/tmdsice3359.

4 System Design Theory

One way to determine the type of industrial Ethernet standard is to compare specific fields to known values inside the 802.3 Ethernet frame structure.

The PRU-ICSS firmware compares the Ethertype field against a known value in four cases out of the five supported industrial Ethernet standards. In addition, the PRU-ICSS firmware analyzes the source MAC address for the EtherNet/IP standard.

4.1 Ethernet Frame Format

The following Table 3 shows the format structure of an 802.3 Ethernet frame. The seven “Preamble” octets (bytes) and the “Start of Frame Delimiter” (SFD) octet are used by the Ethernet PHY and MAC to synchronize the receive logic to the incoming frame. The MII block within the PRU-ICSS is configured to remove the 8 bytes of frame header from the incoming frame. Therefore, the MAC destination address is the first information that is stored in the RX L2 ring buffer, followed by the MAC source address. After the MAC source address, in situations where there is no 802.1Q tag supported by the Ethernet standard, the next two bytes that are received are two octets of the Ethertype field. These two octets are at an offset of 12 bytes from the start of the MAC destination address. The Ethertype information is used to determine the type of the industrial Ethernet standard.

Table 3. Ethernet Packet and Frame Structure

LAYER	PREAMBLE	START OF FRAME DELIMITER	MAC DESTINATION	MAC SOURCE	802.1Q TAG (OPTIONAL)	ETHERTYPE (ETHERNET II) OR LENGTH (IEEE 802.3)	PAYLOAD	FRAME CHECK SEQUENCE (32-BIT CRC)	INTERPACKET GAP
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46(42)[1]–1 500 octets	4 octets	12 octets
Layer 2 Ethernet frame			← 64–1518(1522) octets →						
Layer 1 Ethernet frame	← 72–1526(1530) octets →								

(1) From IEEE 802.3-2005 Clause 3.5

4.2 Industrial Ethernet Standard Specifics

Each industrial Ethernet standard has its own characteristics in the Ethertype field. Table 4 lists the expected Ethertype values of each individual industrial Ethernet standard. As soon as the frame has been received, the PRU firmware loads 2 bytes of Ethertype field from the PRU0 local RAM (located at 12 bytes offset from the MAC destination address, as the preceding Section 4.1 details). Compare the frame’s Ethertype value against the value that Table 4 shows to determine the type of industrial Ethernet standard.

Table 4. Industrial Ethernet Standard Specific Ethertype Values

PROTOCOL NAME	ETHERTYPE FIELD VALUE (2 BYTES)	COMMENT
EtherCAT	0x88AE	
PROFINET	0x8892	
Sercos III	0x88CD	
PowerLink	0x88AB	
EtherNet/IP	0x8000	IP protocol

Before the PRU-ICSS completes the analysis, the PRU-ICSS also checks whether the MAC source address matches within the MAC address range for Rockwell Automation. The PRU-ICSS makes this check only for the EtherNET/IP, which has the IP protocol value as Ethertype.

4.3 Multi-Protocol Industrial Ethernet Detection—Function Sequence

The following Figure 6 shows a flow chart of the generic function block. The highlighted blue boxes are part of this TIDEP0032 TI design.

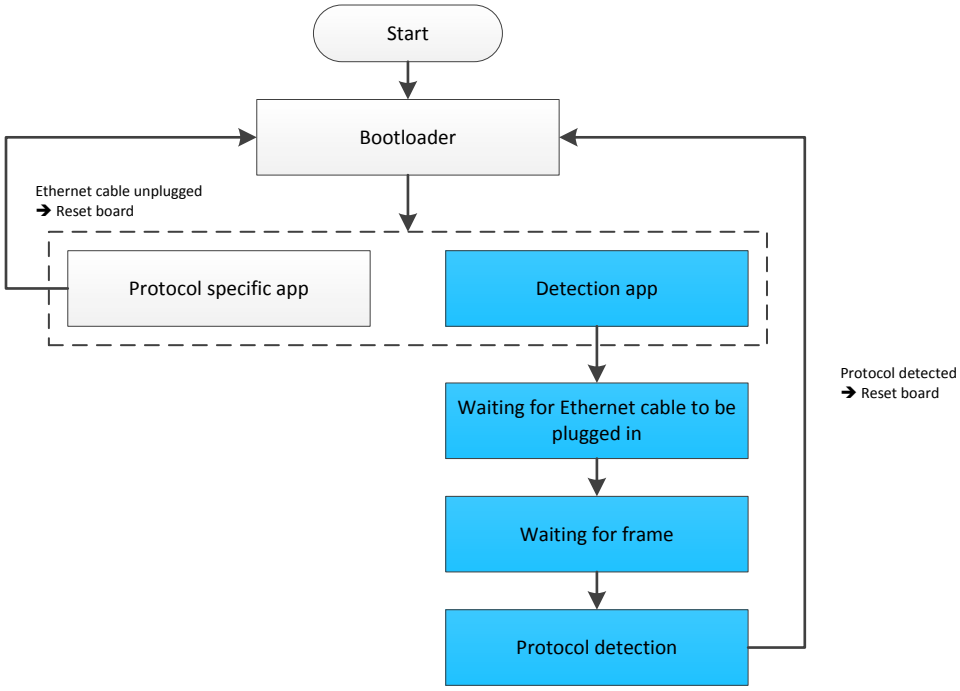


Figure 6. Multi-Protocol Application Flow Chart

After the bootloader loads the detection application binary into the ARM processor, the ARM initializes the required SoC function blocks including PRU-ICSS, MII, and Ethernet PHYs. Then the application waits for the user to plug in an Ethernet cable to establish a PHY link with the PLC. After establishing this connection, the detection application waits to receive an Ethernet frame. As soon as an Ethernet frame has been received, the application analyzes the frame content to determine the industrial Ethernet standard. After the industrial Ethernet standard has been determined, the application generates a reset to reload the bootloader.

4.3.1 ARM Application Description

The ARM application consists of a main() function, a task, and an interrupt service routine (ISR). The application uses the Texas Instruments SYS/BIOS™ real-time operating system (RTOS).

The main() function performs the following tasks:

- Initializing SoC device
  - Memory management unit (MMU)
  - Type of board detection based on information stored in the on-board EEPROM
  - Configuration of processor pins
  - SPI, GPIO, LED, and UART peripheral
- Initializing PRU-ICSS, including configuration of RX L2 ring buffer within the MII block
- Registering an interrupt handler that receives events generated by PRU-ICSS
- Registering task reset_task()
- Displaying message onto the OLED display
- Starting SYS/BIOS scheduler

The ISR waits to receive an interrupt generated by the PRU-ICSS firmware. The interrupt is issued when an Ethernet frame has been received and analyzed. The ISR checks the type of industrial Ethernet standard and writes this information into the RTC_SCRATCH0_REG register. This register maintains the data integrity if the device performs a cold start (reset), allowing the information stored in this register to be evaluated by the bootloader.

The following Table 5 and Table 6 are references from the AM3359 *Technical Reference Manual (TRM)* [4] and provide further information about the register RTC_SCRATCH0_REG and PRM_RSTCTRL.

Table 5. RTC_SCRATCH0_REG Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
RTCSCRATCH0																														
R/W-0h																														

The RTC_SCRATCH0_REG is located at memory location 0x44E3E060.

Table 6 shows the PRM_RSTCTRL register and Table 7 describes the fields. The global software features cold and warm reset control. This register is auto-cleared and only “Write 1” is possible; a read only returns a “0”.

Table 6. RTC_SCRATCH0_REG Register Field Descriptions

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8
RESERVED																							
Rreturns0s-0h																							
7					6					5				4				3				2				1	
RESERVED																RST_GLO BAL_ SW COLD_				RST_GLOBAL_ WARM_SW			
Rreturns0s/W-0h																Rreturns0s/W-0h				Rreturns0s/W-0h			

Table 7. PRM_RSTCTRL Register Field Descriptions

BIT	FIELD	TYPE	RESET	DESCRIPTION
31-2	RESERVED	Rreturns0s	0h	
1	RST_GLOBAL_COLD_ SW	Rreturns0s/W	0h	Global COLD software reset control. This bit is reset only upon a global cold source of reset. Read returns 0. 0h = 0x0 : Global COLD software reset is cleared. 1h = 0x1 : Asserts a global COLD software reset. The software must ensure the SDRAM is properly put in self-refresh mode before applying this reset.
0	RST_GLOBAL_WARM_ SW	Rreturns0s/W	0h	Global WARM software reset control. This bit is reset upon any global source of reset (WARM and COLD). Read returns 0. 0h = 0x0 : Global WARM software reset is cleared. 1h = 0x1 : Asserts a global WARM software reset.

The *task_reset()* waits until the *g_reset* parameter value equals ”1”; otherwise, the *task reset()* resumes sleep mode. As soon as the *g_reset* parameter equals “1”, it sets the *RST_GLOBAL_COLD_SW* bit into the *PRM_RSTCTRL* register at the 0x44E00F00 memory location.

Setting the *RST_GLOBAL_COLD_SW* bit triggers a SoC cold-start software reset that causes the bootloader to reload. After reloading, the bootloader can determine the kind of industrial Ethernet standard by reading out *RTC_SCRATCH0_REG*. The bootloader must then clear the *RTC_SCRATCH0_REG* register before the bootloader loads the industrial application software with the appropriate industrial Ethernet standard.

4.3.2 PRU-ICSS Firmware Description

The PRU firmware first checks (through MDIO access) whether the Ethernet PHY has established a link, which means the Ethernet cable has been plugged into the RJ45 connector and a link with the PLC has been established.

As soon as the link has been established, the PRU-ICSS firmware waits for the reception of a new Ethernet frame. The PRU-ICSS firmware stores the Ethernet frame into shared memory. Upon the complete reception of the frame, the PRU-ICSS firmware analyzes the type of industrial Ethernet based on the Ethertype field inside the Ethernet frame (located at an offset of 12 bytes from the frame start). After analyzing the Ethertype field, the PRU-ICSS firmware stores the type of detected standard in PRU local memory and generates an interrupt to the ARM through the interrupt controller. After generating the interrupt, the PRU-ICSS firmware returns to wait and receive the next frame.

4.4 Extend PRU-ICSS Firmware for Industrial Ethernet Protocols

Engineers can extend the PRU-ICSS firmware to analyze additional industrial Ethernet standards. To achieve such an extension, the engineer must know the characteristics of the industrial Ethernet standard. One example of a characteristic or identification parameter is the Ethertype field, which is used with EtherCAT. If the Ethertype matches with 0x88AE, the PRU-ICSS firmware determines that this is an EtherCAT frame. Note that in the PRU-ICSS register the 2 bytes are swapped, which means that the firmware checks for a match with 0xAE88. The same program flow can be used for additional Ethertype values, as determined by the engineer.

If checking additional fields in the Ethernet frame is necessary, the engineer can access the complete frame, which is stored in PRU0 local data RAM (memory location 0x00000), using the PRU-ICSS firmware.

5 Getting Started

The following hardware and software is required to demonstrate and evaluate the multi-protocol industrial Ethernet detection with PRU-ICSS.

5.1 Hardware Requirements

- TMDSICE3359 ICE EVM (TMDSICE3359)
- Programmable logic controllers that support the following industrial Ethernet standards: EtherCAT, PROFINET, EtherNet/IP, Sercos III, and PowerLink

5.2 Software Requirements

- CCSv6 (http://processors.wiki.ti.com/index.php/Download_CCS)
- AM335x SYS/BIOS Industrial SDK 01.01.00.05
- SYS/BIOS 6.41.0.26 and XDCtools 3.30.5.60
- ARM Compiler TI V5.1.10
- PRU C-Compiler v2.1.1

The TIDEP0032 design consists of an ARM and a PRU project for CCSv6. The user must import and build each of the two projects in CCS, individually. The PRU project generates a firmware header file (*.h*) during the build process. The ARM project includes the PRU firmware header files during its build process, which means that the user must first build the PRU project before he or she builds the ARM project. Note that if a user modifies the PRU-ICSS firmware, he or she must also rebuild the PRU and ARM project for the PRU-ICSS firmware changes to take effect.

5.3 CCS Import, Build, and Download

Locate the ARM project in the subfolder *...\detection_app* (from the default installation pathway).

Locate the PRU project in the subfolder *...\detection_firmware_ccs*.

First import both projects into CCS through the *Import CCS Eclipse Projects* window (see Figure 7 and Figure 8).

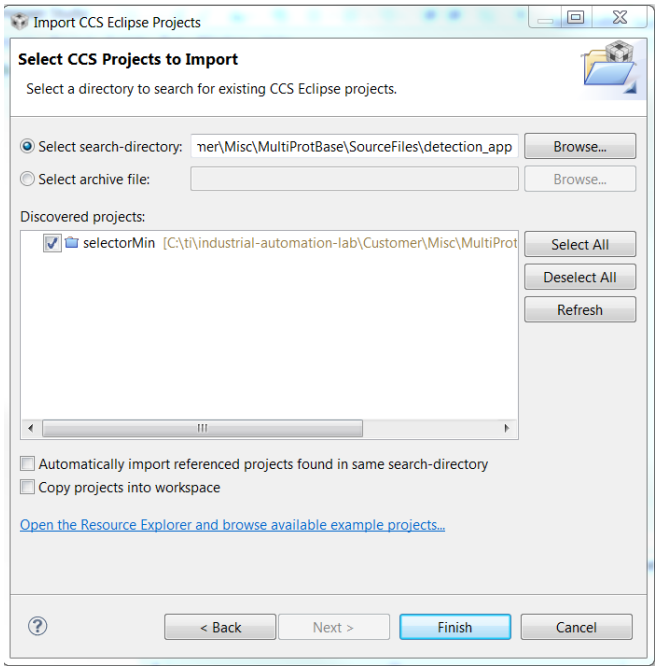


Figure 7. CCS Import Dialog for ARM Project

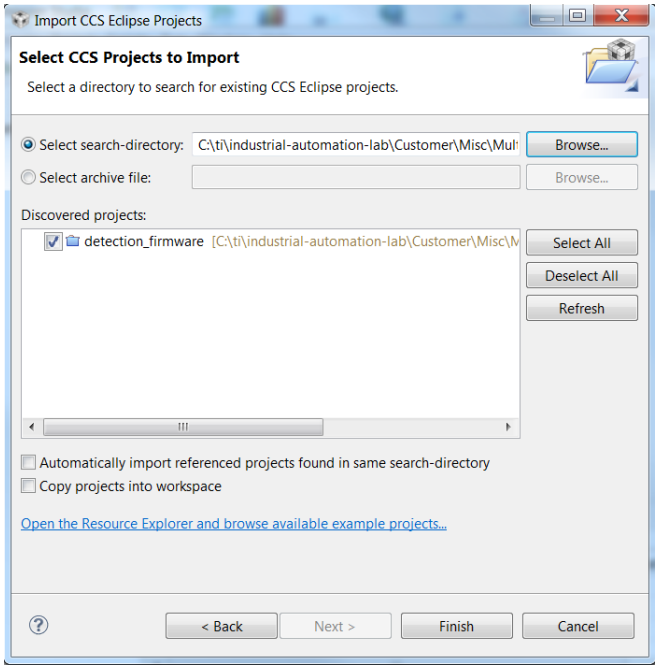


Figure 8. CCS Import Dialog for PRU Project

After importing the projects, the project explorer shows the PRU project *detection_firmware* and the ARM project *selectorMin* (see Figure 9).

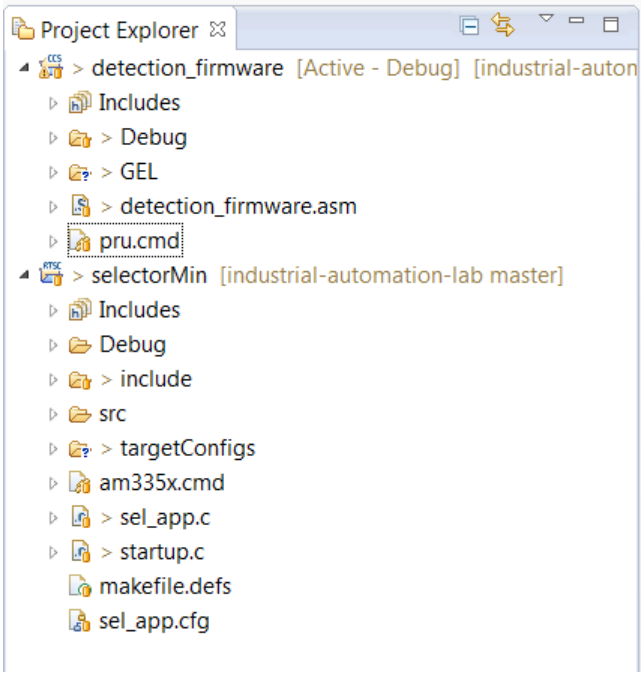


Figure 9. CCS Project Explorer View

The next step is to rebuild the PRU project:

- In the *Project Explorer* window select the *detection_firmware* folder with the cursor
- From the menu bar, select: *Project→ Build Project*
- This process builds the PRU-ICSS project and generates the file *detection_firmware.out*

Open a new *CMD* window to generate the *.h* file from the *.out* file (see Figure 10). The *.h* file is used by the *selectorMin* application and contains the PRU-ICSS firmware.

- Navigate to the *detection_firmware_css\Debug* folder
- Execute the file *build_header.bat*
- Executing this file generates the protocol_detection_bin.h and copies it into the *selectorMin\include* folder

NOTE: The user may require changing the file path inside the *.bat* file to accommodate the development paths of the specific PC in use.

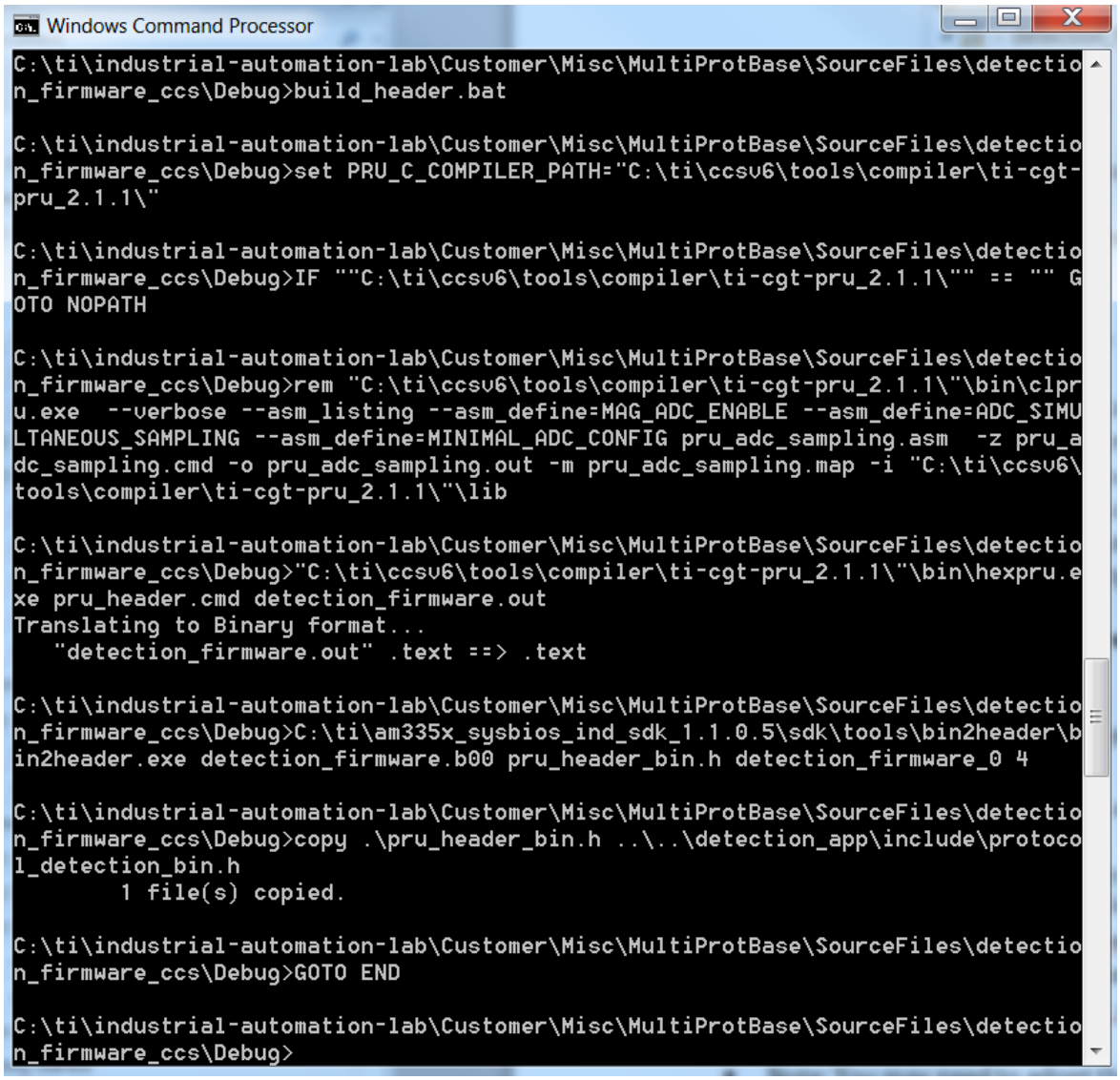


Figure 10. Generating Firmware Header File

The next step is to generate a target configuration file (if it does not exist already), see Figure 11. Generating this file allows the CCS software to connect through the JTAG with the TMDSICE3359 ICE EVM.

- From the menu bar, select *File* → *New* → *Target Configuration*
- In the basic panel:
  - Under the *All Connections* panel, select: "Texas Instruments XDS100v2 USB Emulator"
  - In *Board* or *Device*, select: AM3359
- In the cascaded advanced panel:
  - Select *CortexA8*
  - Select the GEL file *TMDXICE3359_v2_1A.gel* from the SDK in the folder *sdk\tools\gel\ICE*
  - Click the *Save* button

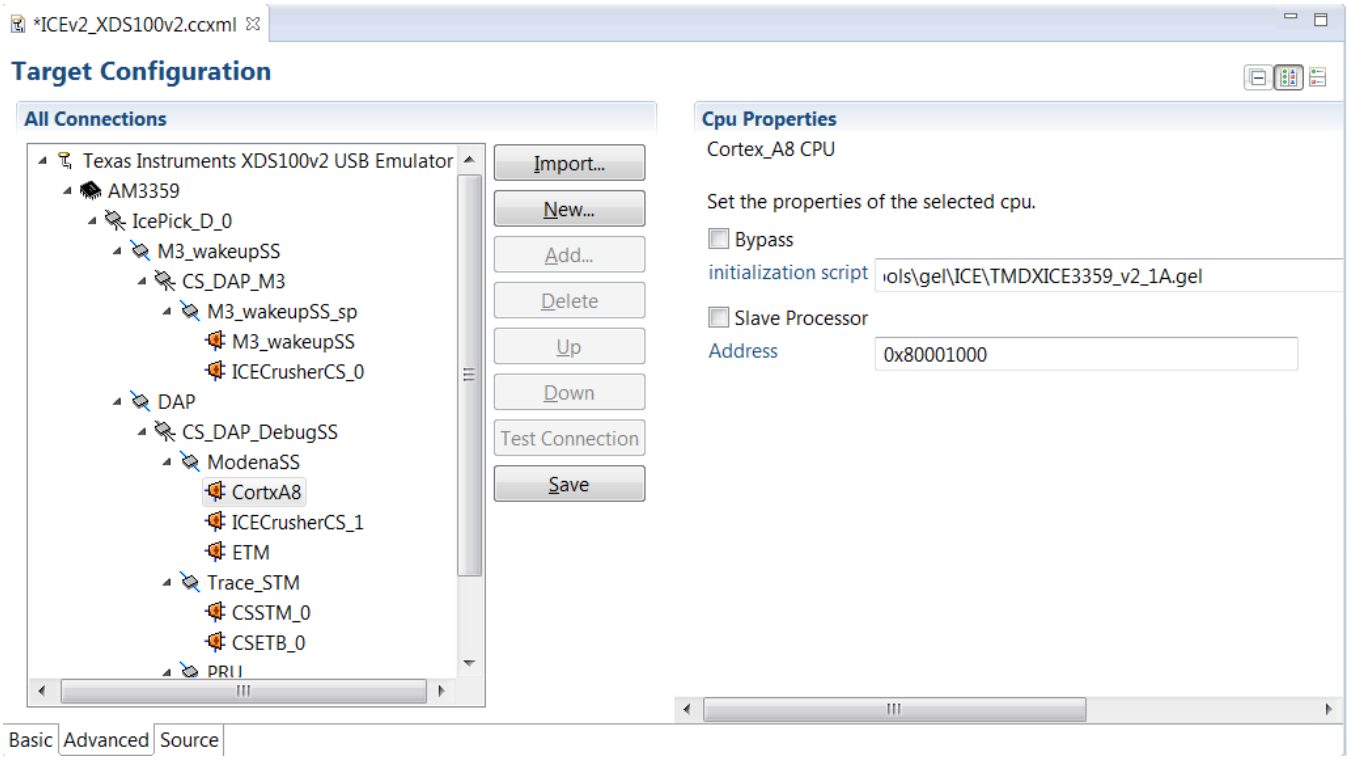


Figure 11. Target Configuration File

Connect the TMDSICE3359 ICE EVM to a 24-V power supply; connect a USB cable between the ICE board and the PC. Next, launch the target configuration file to set up the JTAG debug connection between CCSv6 and the ICE EVM.

- In the *Debug* window, select the *CortexA8* entry
- From the menu bar, select: *Run → Connect Target*
- Select: *Run → Reset → CPU Reset (HW)*
- Select: *Scripts → AM335x System Initialization → AM3359_ICE_INITIALIZATION*
- Select: *Run → Load → Load Program*
- Use the *Browse project* button to select the file *selectorMin.out* and press the *OK* button two times
- Now the program is downloaded into the ARM processor. As soon as the download completes, the *Program Counter* stops at the main() function (see Figure 12).

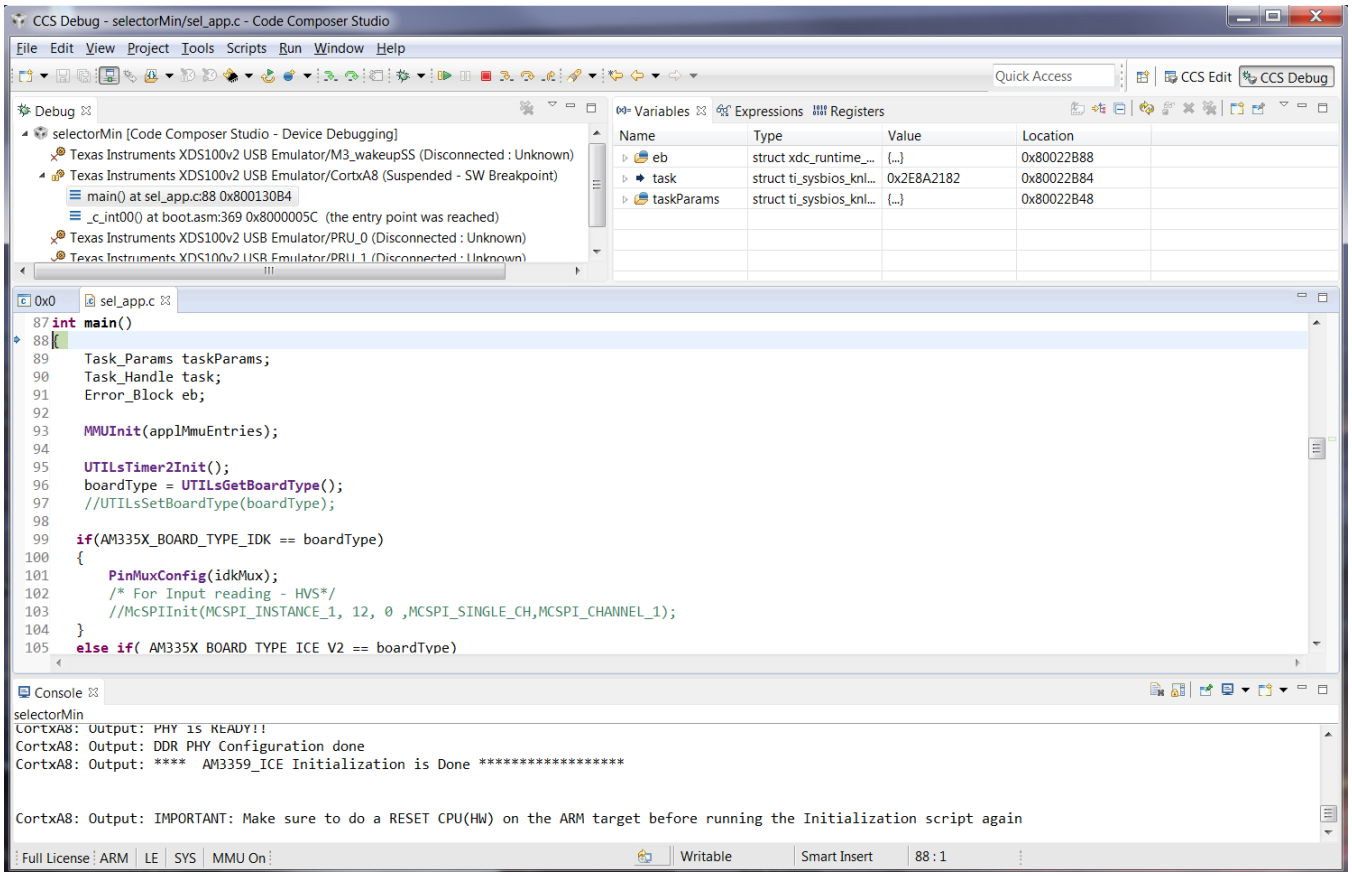


Figure 12. CCS Debug View After Program Download

For testing and debug purposes, TI recommends to add a breakpoint into the task function *task_reset* to prevent the application from resetting the SoC. If the application resets the SoC, then CCS loses the JTAG connection to the ICE EVM (Figure 13). Alternatively, the user can also comment out the HWREG line to prevent the overall generation of a reset.

```
239 void task_reset(UArg arg0, UArg arg1) {
240
241     while(1) {
242         if(g_reset) {
243             HWREG(0x44E00F00) = 0x2;
244         }
245         Task_sleep(100);
246     }
247 }
```

Figure 13. Add Breakpoint Into task_reset() to Prevent SoC Reset

Now start the ARM application:

- From the menu bar, select: *Run → Resume*
- Plug the Ethernet cable, which is connected to the PCL, into the RJ45 connector (J2)
- As soon as the PLC sends a frame and the protocol has been determined, the application must be stopped at the breakpoint in the function *task_reset()*

6 Test Setup

The TIDEP0032 TI design has been validated with PLCs from Beckhoff (EtherCAT), Siemens (PROFINET), Automata (Sercos III), B&R (PowerLink), and Rockwell Automation/Allen-Bradley (EtherNet/IP).

The following Figure 14 shows the test setup and Figure 15 shows the physical demonstrator platform.

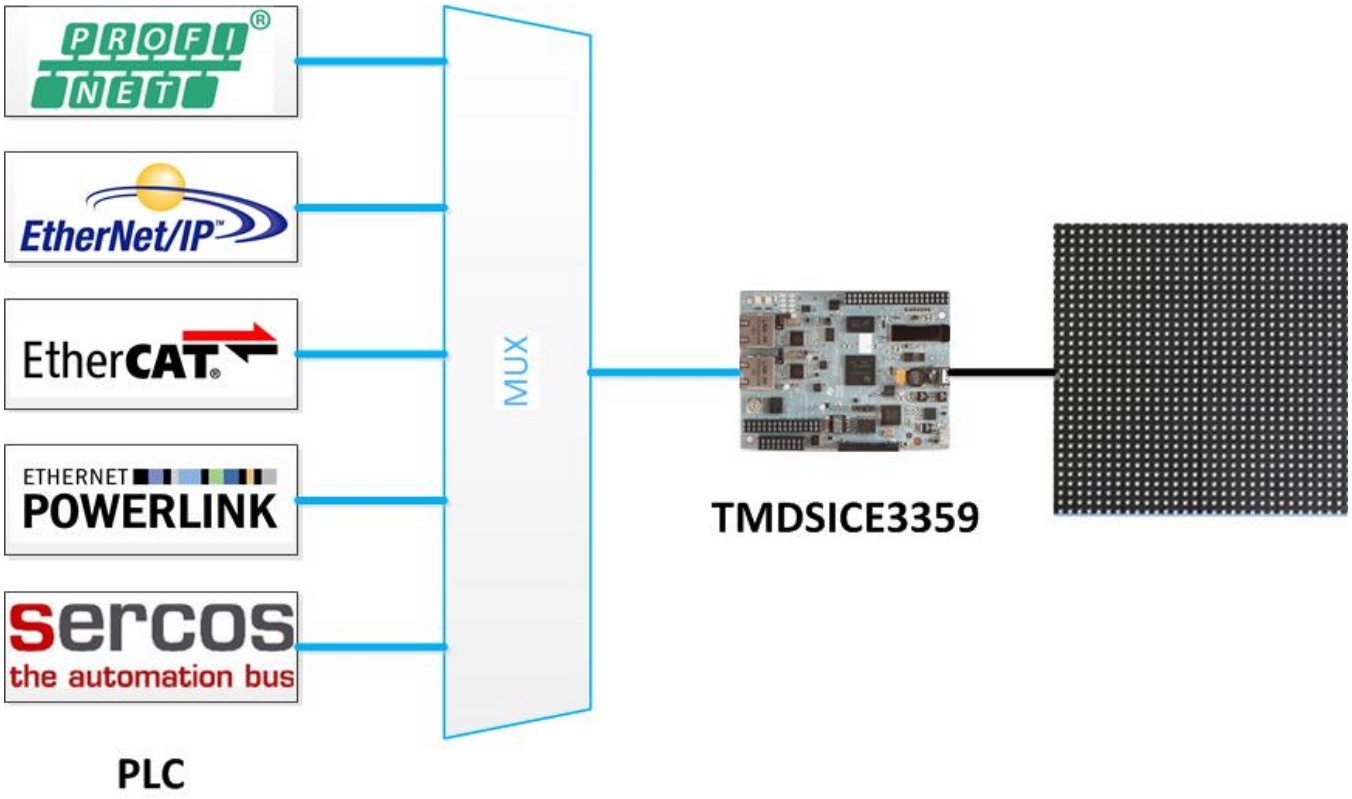


Figure 14. Multi-Protocol Test Setup

Figure 15. Multi-Protocol Demonstrator as Shown at Electronica 2014 and SPS IPC Drives 2014

7 Design Files

7.1 Schematics

To download the schematics, see the design files at TIDEP0032.

7.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDEP0032.

7.3 PCB Layout

To download the PLC layout, see the design files at TIDEP0032.

7.4 Gerber Files

To download the Gerber files, see the design files at TIDEP0032.

7.5 Physical TMDSICE3359 EVM

Purchase the TMDSICE3359 ICE EVM from the TI online shop. Please refer to: TMDSICE3359.

8 Software Files

To download the software files for this design, see the design files at TIDEP0032.

9 References

1. Texas Instruments, *Texas Instruments Industrial Communications*, Demonstration Video on Multi-Protocol Detection Panel from Electronica 2014 (http://bit.ly/1Kh2Qaf)
2. Texas Instruments, *PRU-ICSS*, TI Wikipedia Page (http://processors.wiki.ti.com/index.php/PRU-ICSS)
3. Texas Instruments, *PRU Assembly Instructions*, TI Wikipedia Page (http://processors.wiki.ti.com/index.php/PRU_Assembly_Instructions)
4. Texas Instruments, *AM335x Sitara ™ Processors*, AM335x Technical Reference Manual (SPRUH73)

10 Terminology

CCS— Code Composer Studio

ICSS— Industrial Communication Subsystem

ISR— Interrupt Service Routine

MII— Media Independent Interface

PLC— Media Independent Interface

PRU— Programmable Real-time Unit

RTOS— Real-time Operating System

SoC— System-on-a-chip

TRM— Technical Reference Manual

11 About the Author

THOMAS MAUER is a System Applications Engineer in the Factory Automation and Control Team at Texas Instruments Freising, where he is responsible for developing reference design solutions for the industrial segment. Thomas brings to this role his extensive experience in industrial communications like Industrial Ethernet and fieldbuses and industrial applications. Thomas earned his Electrical Engineering degree (Dipl. Ing. (FH)) at the University of Applied Sciences in Wiesbaden, Germany.

FABIAN FISCHER is a Master student at the RWTH Aachen University in Electrical Engineering and Information Technology. As a part of the Factory Automation and Control Team at Texas Instruments in Freising, he is developing solutions for the industrial area.

IMPORTANT NOTICE FOR TI REFERENCE DESIGNS

Texas Instruments Incorporated ("TI") reference designs are solely intended to assist designers (“Buyers”) who are developing systems that incorporate TI semiconductor products (also referred to herein as “components”). Buyer understands and agrees that Buyer remains responsible for using its independent analysis, evaluation and judgment in designing Buyer’s systems and products.

TI reference designs have been created using standard laboratory conditions and engineering practices. **TI has not conducted any testing other than that specifically described in the published documentation for a particular reference design.** TI may make corrections, enhancements, improvements and other changes to its reference designs.

Buyers are authorized to use TI reference designs with the TI component(s) identified in each particular reference design and to modify the reference design in the development of their end products. HOWEVER, NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY THIRD PARTY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT, IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI REFERENCE DESIGNS ARE PROVIDED "AS IS". TI MAKES NO WARRANTIES OR REPRESENTATIONS WITH REGARD TO THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, EXPRESS, IMPLIED OR STATUTORY, INCLUDING ACCURACY OR COMPLETENESS. TI DISCLAIMS ANY WARRANTY OF TITLE AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT, QUIET POSSESSION, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS WITH REGARD TO TI REFERENCE DESIGNS OR USE THEREOF. TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY BUYERS AGAINST ANY THIRD PARTY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON A COMBINATION OF COMPONENTS PROVIDED IN A TI REFERENCE DESIGN. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ON ANY THEORY OF LIABILITY AND WHETHER OR NOT TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, ARISING IN ANY WAY OUT OF TI REFERENCE DESIGNS OR BUYER’S USE OF TI REFERENCE DESIGNS.

TI reserves the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques for TI components are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

Reproduction of significant portions of TI information in TI data books, data sheets or reference designs is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards that anticipate dangerous failures, monitor failures and their consequences, lessen the likelihood of dangerous failures and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in Buyer’s safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed an agreement specifically governing such use.

Only those TI components that TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components that have ***not*** been so designated is solely at Buyer's risk, and Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265