TI Designs Industrial Communications Gateway PROFINET[®] IRT to PROFIBUS[®] Reference Design

TEXAS INSTRUMENTS

Overview

PROFINET® is becoming the leading industrial Ethernet protocol in automation due to its high-speed, deterministic communications, and enterprise connectivity; however, as the world's most popular fieldbus, PROFIBUS®'s importance and usage will continue for many years due to legacy investment protection. Recognizing the hybrid attribute in process plants, this TI Design presents an integral solution for the migration of existing fieldbus technology into the real-time Ethernet network, which is based on the programmable real-time unit and industrial communication subsystem (PRU-ICSS) of TI's Sitara™ AM57x processor. The TI Design demonstrates PROFIBUS master and PROFINET IRT device on AM572x with protocol stacks simultaneously running on an ARM® Cortex®-A15 core while an entire PROFINET switch and PROFIBUS frame handler are executed on the two PRU-ICSSes respectively. The migration scheme is generally also suitable for other fieldbus and industrial protocols.

Resources

TIDEP-0075 AM572x TMDXIDK5728 PRU-ICSS-INDUSTRIAL-SW PROCESSOR-SDK Design Folder Product Folder Tools Folder Tools Folder Tools Folder

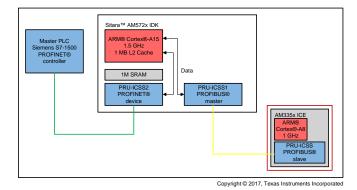


Features

- PROFIBUS:
 - Cyclic PROFIBUS-DP (DPV0) and Enhancements to DP (DPV1) for Acyclic Services
 - Transmission Speed: 9.6 Kbaud to 12 Mbaud
- PROFINET IRT:
 - Conformance Classes A, B, and C Functionality
 - Minimum Cycle Time of 250 µs
- PROFIBUS Master and PROFINET IRT Device Simultaneously Supported
- Simplified Protocol Translation Between PROFIBUS and PROFINET IRT
- Leverage PRU-ICSS Industrial Software for Sitara Processors Built on top of Processor-SDK-RTOS
- Tested on TMDXIDK5728 Board and Includes Documentation, Software, Demonstration Application, and Hardware Design Files

Applications

- Factory Automation & Controls: PLC, DCS & PAC: Communication Module
- Factory Automation & Controls: PLC, DCS & PAC: CPU (PLC Controller)



1



An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

1 System Overview

PROFIBUS (process fieldbus) is a standard for automation technology, which was first developed in 1989 in Germany. Today, there are over 35 million PROFIBUS nodes installed. The PROFIBUS industrial field bus is used to connect controllers to remote input and output units, sensors, actuators, and internetworking components. The applications where PROFIBUS is deployed include factory automation, drives and motion control, process automation, and safety-critical applications[1].

PROFINET is a real-time Ethernet standard for the high-speed, deterministic communications required for a wide range of industrial applications including factory automation, process automation, and building automation. Originally developed under the leadership of Siemens GmbH as a network extension to the popular PROFIBUS fieldbus technology, PROFINET is now supported by PROFIBUS and PROFINET International, the largest fieldbus organization in the world. Standard Ethernet is not efficient for the small amounts of data exchange that are typical in factory automation, for example, and the lack of determinism means Ethernet cannot support the real-time operation necessary for motor drive control and other industrial real-time applications. PROFINET is a leading example of solving these challenges so that Ethernet can be adapted for real-time industrial applications, such as those found in factory automation environments.[2].

PROFINET encompasses many features from PROFIBUS. PROFINET's unprecedented performance and enterprise-wide connectivity help realize great operational benefits and compelling business advantages. Market research indicates there is high level of usage of PROFIBUS and the market is starting to transition to PROFINET. While PROFIBUS and PROFINET remain popular in the market, new PROFINET installations are now overtaking PROFIBUS installations. However, the transition is likely to be a gradual process due to legacy investment protection and conservative nature of manufacturing industry, especially for process and factory automation. Current installations in factories and plants include many end nodes communicating over PROFIBUS including PLCs, motion controllers, HMIs, drives, and IO modules. As manufacturers look to include newer PROFINET-enabled equipment in their factories, they will need to bridge the new PROFINET networks to the older PROFIBUS networks in a seamless way. Industry 4.0 gateways, communications modules, or PLCs and other equipment capable of communicating over both protocols will be critical to helping overcome this change.

In this TI Design, an integrated approach for the migration of existing fieldbus technology PROFIBUS into the real time Ethernet network PROFINET, which is based on PRU-ICSS industrial software for the Sitara processors AM572x is proposed and implemented. The PRU-ICSS industrial software packages are built to use on top of Processor-SDK-RTOS, a unified software development platform, and provide optimized PRU-ICSS firmware, a corresponding PRU-ICSS driver for the ARM processor and example applications to enable real-time industrial communications for TI Sitara processors.

1.1 System Description

2

This TI Design leverages the PROFIBUS master example and PROFINET IRT device example provided in PRU-ICSS industrial software for Sitara processors. The design integrates and prioritizes the tasks of each protocol and manages cache and interrupt resources to allow two protocols run simultaneously. In addition, a simplified protocol translation scheme is implemented to make a seamless connection within a distributed heterogeneous industrial enterprise environment possible.

PROFIBUS master frames exchanged with AM335x ICE slave are processed by PRU-ICSS1.

PROFINET IRT device packets transferred with the master programmable logic controller (PLC) (Siemens S7-1500) PROFINET controller are processed by PRU-ICSS2.

The two protocol stacks run simultaneously on one Cortex-A15 processor and are translated on the host processor.



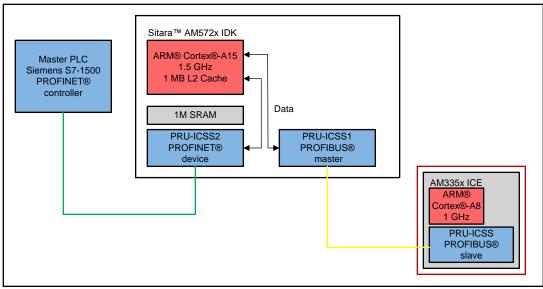
1.2 Key System Specifications

Table 1	. Key System	Specifications
---------	--------------	----------------

PARAMETER	SPECIFICATIONS
PROFIBUS services	Cyclic PROFIBUS-DP (DPV0) and enhancements to DP (DPV1) for acyclic services
PROFIBUS transmission speed	9.6 Kbaud to 12 Mbaud
PROFINET IRT conformance	Class A, B, and C
PROFINET IRT minimum cycle time	250 µs
Protocol translation	PROFIBUS and PROFINET

1.3 Block Diagram

The system block diagram (Figure 1) illustrates the multiprotocol setup on AM572x and communication with respective master and slave devices.



Copyright © 2017, Texas Instruments Incorporated

Figure 1. System Block Diagram

1.4 Design Considerations

- PROFIBUS protocol implementation
- PROFINET IRT protocol implementation
- Protocol integration and translation
 - Tasks and priorities
 - DDR cache-ability
 - Interrupt mapping

3



1.5 Highlighted Products

1.5.1 AM572x

The AM572x Sitara ARM processor is built to meet the complex processing needs of modern embedded industrial products. The AM572x provides high-processing performance through the flexibility of a fully-integrated mixed processor solution with two ARM Cortex-A15 cores, two ARM Cortex-M4 cores, and two TI C66x digital signal processor (DSP) cores as well as two industrial communication subsystems (PRU-ICSS), which can be used for real-time communications and IO applications, such as PROFINET, PROFIBUS, EtherCAT[®], and more.

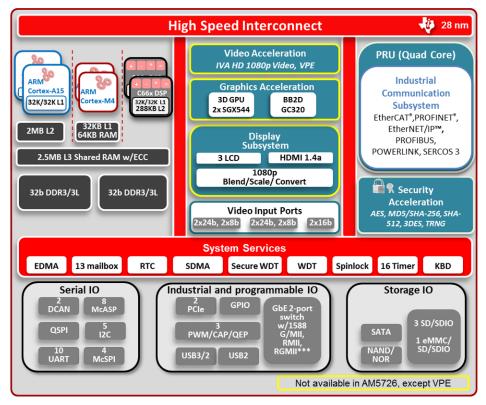


Figure 2. AM572x Architecture

1.5.2 PRU-ICSS

The PRU-ICSS consists of:

- Two 32-bit load and store RISC CPU cores:
 - Programmable real-time units (PRUs): PRU0 and PRU1
- Data RAMs per PRU core
- Instruction RAMs per PRU core
- Shared RAM
- Peripheral modules
- Interrupt controller (PRUSS_INTC)

The programmable nature of the PRU cores, along with their access to pins, events, and all device resources, provides flexibility in implementing fast real-time responses, specialized data handling operations, custom peripheral interfaces, and in offloading tasks from the other processor cores of the device. The PRU cores are programmed with a small, deterministic instruction set. Each PRU can operate independently or in coordination with each other and can also work in coordination with the device-level host CPU. This interaction between processors is determined by the nature of the firmware loaded into the PRU's instruction memory.

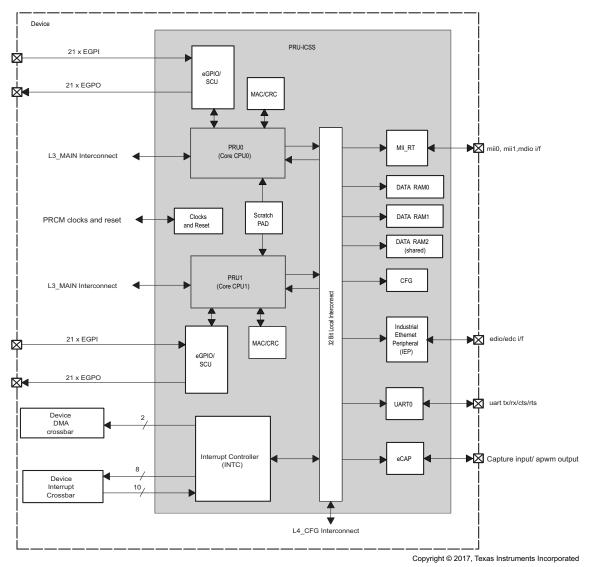


Figure 3. AM572x PRU-ICSS Architecture



System Overview

www.ti.com

Among the interfaces supported by the PRU-ICSS are real-time industrial protocols used in master and slave mode, such as:

- EtherCAT
- PROFINET
- EtherNet/IP™
- Ethernet POWERLINK
- SERCOS III
- HSR and PRP
- EnDat 2.2

More details about the processor features refer to AM572x Sitara[™] Processors Silicon Revision 2.0, 1.1[1].

1.6 System Design Theory

The PRU-ICSS industrial software for Sitara processors provides the foundation for the multiprotocol integration and data translation. The software includes firmware for the protocols, drivers, hardware initialization routines, adaptation layer for the stack API, protocol stack, and the application itself. (Disclaimer: The protocol stacks are from third-party vendors and validated on Sitara devices. The copy of the stack libraries included is solely for evaluation, development and test purposes).

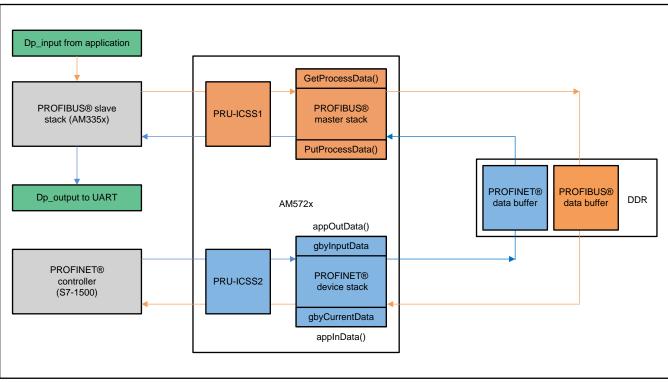
To integrate the PROFINET and PROFIBUS from two separate TI-RTOS Code Composer Studio[™] (CCS) projects in the PRU-ICSS industrial software package, a few aspects need to be taken into consideration. These include memory management, cache-ability, interrupt mapping, and real-time task scheduling and synchronization.

The protocol translation is implemented through buffer lists. Each protocol owns a buffer list with a *length* field as flag to indicate data availability while the other protocol polls the field. The timing of data transfer is controlled by the application that aligns with each protocol's internal state machine.

The data initially is from dp_input buffer from the PROFIBUS slave application then received by the PROFIBUS master stack from the GetPRocessData() API through PRU-ICSS1 and stored in the PROFBUS data buffer in DDR. The PROFINET device polls the PROFIBUS data buffer and sends it to the PROFINET controller. A *move* module is added in the PROFINET controller, which will route the data back to the PROFINET device. Once the PROFINET device receives the data from the PROFINET controller, the PROFINET device stores the data into shared data buffer. PROFIBUS sends the data from the PROFINET data buffer to slave through PutProcessData(). Finally, the data received in Output_buffer of PROFIBUS slave is transmitted to the universal asynchronous receiver-transmitter (UART) console.



Figure 4 shows the protocol translation scheme.



Copyright © 2017, Texas Instruments Incorporated

Figure 4. Protocol Translation Scheme



System Overview

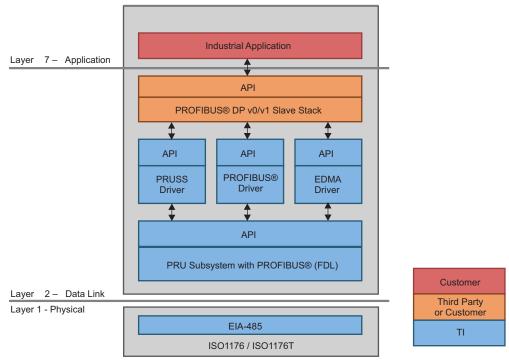
8

1.6.1 PROFIBUS[®] Implementation on AM572x

The PROFIBUS software components comprise the integrated PROFIBUS solution:

- Microcode that implements FDL functionality in the PRU
- PROFIBUS-DP protocol that runs on the ARM MPU
- Industrial application

Figure 5 shows the PROFIBUS component software architecture.



Copyright © 2017, Texas Instruments Incorporated

Figure 5. PROFIBUS® Software Architecture

The PROFIBUS real-time frame handler (fieldbus data link or FDL) is encapsulated in the PRU-ICSS1. The PRUs implement real-time PROFIBUS message transmission, frame validation, and communication with the ARM processor. Interrupts are used to communicate with the ARM where the PROFIBUS stack (layer seven, DP-protocol) and the industrial application are run. All process data handling like cyclic, acyclic, and service access point (SAP) between the PROFIBUS stack on ARM and the PRU is through the internal memory. One of the PRUs controls the integrated on-chip UART that is designated for PROFIBUS communication at up to 12-Mbaud data rate. The industrial application and the PROFIBUS DP-protocol (layer seven) are operated on the ARM. The solution can be completed with an RS-485 transceiver suitable for harsh environments, such as TI's ISO 1176T PROFIBUS transceiver[1].



1.6.2 **PROFINET® IRT Implementation on AM572x**

The PROFINET slave software components comprise the integrated PROFINET solution:

- Microcode that implements layer two functionality in the device's PRU
- PROFINET slave stack that runs on the ARM processor
- Industrial application

Figure 6 shows the PROFINET software architecture.

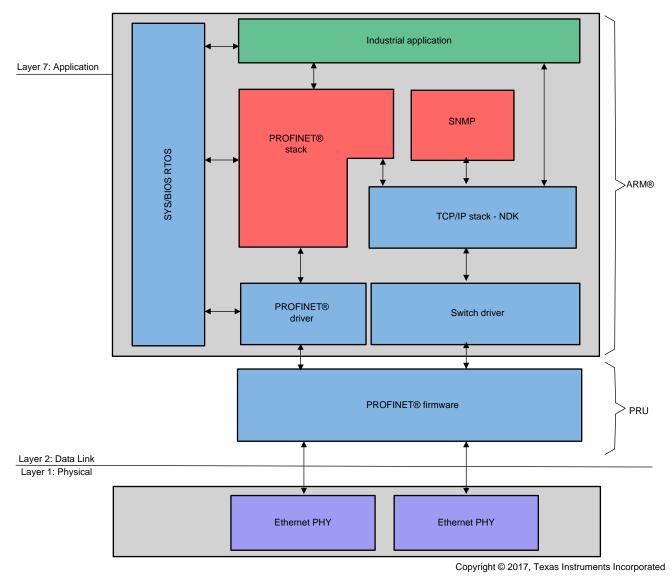


Figure 6. PROFINET[®] Software Architecture

The PROFINET slave layer two functionalities are implemented in PRU-ICSS2.

CPM and PPM processing CPM frames are parsed by the PRU-ICSS firmware to check whether the host processor should be notified. Firmware stores the CPM frame in predefined buffer so that data is completely ready for host consumption before it is interrupted.

PPM frames for all the active IOCRs are generated or produced by the firmware running on PRUs. At the start of every phase, firmware determines which PPM frames must be transmitted in that particular phase, and the firmware transmits them out without any interference from host.

9



System Overview

Data hold timer (DHT) is implemented in the PRU firmware, and whenever the DHT expires, an interrupt is raised to the ARM processor running the PROFINET stack.

To reduce the number of DCP identify request frames reaching the ARM processor, DCP identify filter is implemented in the PRU firmware to filter out all the DCP identify frames which are not meant for the host.

A two-port, cut-through switch is integrated in PROFINET firmware. The switch handles the non-real-time traffic and can be interfaced with PROFINET, TCP, and IP stacks running on the host[2][3].

1.6.3 Protocol Integration and Translation

1.6.3.1 Key PROFIBUS[®] and PROFINET[®] APIs, Global Variables, and Buffers

The following high-level APIs are listed to provide an overview of PROFIBUS and PROFINET stack functionality.

PROFIBUS:

- · Get the process data of required length from the given offset.
 - short GetProcessData (short DataOffset, short DataLen, BYTE *ProcessDataPtr, short UserLock);
- Put the process data of required length from the given offset.
 - short PutProcessData (short DataOffset, short DataLen, BYTE
 *ProcessDataPtr, short UserLock);
- Transfer the process data.

```
- short TransferProcessData (TPDTransferPrm* PDPrm);
```

PROFINET:

• Stack input data buffer.

```
- APP_BYTE gbyInput_Data[MAX_DATA_FRAME_IO];
```

- Stack output data buffer.
 - APP_BYTE gbyOutputData[MAX_DATA_FRAME_IO];
- Stack AR status information.
 - APP_BYTE gbyOutputAPDU[MAX_ARD_PER_DEVICE];
- Stack user ALARM flag.
 - volatile unsigned char appAlarmActive;

PROFIBUS data are processed in tasks - appInData and appOutData with same priority level 2.

```
Task_Params_init(&taskParams);
```

```
taskParams.priority = 2;
```

```
taskParams.instance->name = "appInData";
taskParams.stackSize = 2048;
Task_create(appInData; &taskParas, NULL);
Task_Params_init(&taskParams);
taskParams.priority = 2;
taskParams.instance->name = "appOutData"
taskParams.stackSize = 2048;
```

Task_create(appInData; &taskParas, NULL);

1.6.3.2 Tasks and Priories

PROFIBUS example main function is integrated into PROFINET project as a subfunction, and the tasks in each protocol are reprioritized to ensure the functions are performed in required process time and critical functions can preempt lower priority threads.

PROTOCOLS	TASKS	PRIORITY	DESCRIPTION
PROFINET	taskLedBlink	2	To control LED blink
PROFINET	appInData	2	To get input data from PROFIBUS master and forward to PROFINET controller
PROFINET	appOutData	2	To send data received from PROFINET controller to PROFIBUS master
PROFINET	taskPruss	3	To initialize all software components and run the stack
PROFINET	RxTask	10	To get packets from ICSS_EMAC and process
PROFIBUS	Uarttask	1 To check the keyboard input for UART key driven n	
PROFIBUS	mainTask	2	To initialize PROFIBUS and handle input and output

1.6.3.3 DDR Cache-ability

PROFIBUS requires 1-MB memory DDR region to be uncached for PRU driver memory mapping. The DDR memory cache-ability is configured in am572x_app.cfg file, where serval memory sections with various cache-ability attributes will be defined. The following is the code snippet for uncached DDR memory region:

```
Mmu.initDescAttrsMeta(attrs0);
attrs0.type = Mmu.DescriptorType_BLOCK; //
                                                    BLOCK descriptor
attrs0.shareable = 2; //
                                 shareable
attrs0.attrIndx = 1; // Non-cache, device
                                                   memory
// Set the descriptor for each entry in
                                                 the address range
for (var i = 0x9000000; i <</pre>
                                     0 \times A0000000; i = i + 0 \times 00200000) 
// Each 'BLOCK' descriptor entry spans a
                                                  2MB address range
Mmu.setSecondLevelDescMeta(i, i,
                                         attrs0);
}
```

1.6.3.4 Interrupt Mapping

AM572x has a large number of interrupts to service the requirements of the device's many peripherals and subsystems. The MPU, DSP (x2), IPU (x2), EVE (x4), and PRU-ICSS (x2) subsystems are capable of servicing these interrupts through their integrated interrupt controllers. In addition, each processor's interrupt controller is preceded by an interrupt controller crossbar (IRQ_CROSSBAR) that provides flexibility in mapping the device interrupts to processor interrupt inputs. The IRQ_CROSSBAR module is controlled by registers in the CTRL_MODULE_CORE submodule. The IRQ_CROSSBAR is able to map any of its input signals to any of its outputs. Each output of the IRQ_CROSSBAR module is connected only to one interrupt line of certain interrupt controller (INTC). Thus, the device IRQs are mapped to the device INTCs through the IRQ_CROSSBAR.

By default, CTRL_CORE_MPU_IRQ_120_121 is configured in both PROFIBUS on PRU-ICSS instance one and PROFINET on PRU-ICSS instance two from crossbar. To get the two protocols concurrently run, PROFIBUS interrupt is remapped through CTRL_CORE_MPU_IRQ_118_119 while PROFINET interrupt mapping remains in CTRL_CORE_MPU_IRQ_120_121.

#define	INTERRUPT_MAP_TO_CHANNEL3	119+32
#define	INTERRUPT_MAP_TO_CHANNEL2	118+32



Getting Started Hardware and Software

PRUICSS_registerIrqHandler(pruHandle, CHANNEL3, INTERRUPT_MAP_TO_CHANNEL3, INTERRUPT_MAP_TO_CHANNEL3, 1, (PRUSSDRV_IRQ_HANDLER)&pruevtout1_isr); PRUICSS_registerIrqHandler(pruHandle, CHANNEL2, INTERRUPT_MAP_TO_CHANNEL2, 1, (PRUSSDRV_IRQ_HANDLER)&pruevtout0_isr);

1.6.3.5 Protocol Translation

The protocol translation is implemented at the application level.

Two buffer lists PnetGbyInput_Data and PbusInputData are created to exchange the process data from each protocol, thus perform protocol translation.

```
struct buffer_list {
uint8_t *address;
uint32_t length;
};
APP_BYTE     PnetGbyInput_Data[NUM_PN_BUFS][MAX_DATA_FRAME_IO]; /* buffer
list for PROFINET slave inputs */
BYTE     PbusInputData[NUM_PB_BUFS][MAX_DATA_LEN]; /* buffer list for
PROFIBUS master inputs */
```

The buffer lists are updated whenever new data is received or transmitted in each protocol. Followed by GetProcessData() in PROFIBUS, PbusInputData is filled by new data received from PROFIBUS slave and indicates its ready for sending to PROFINET device; meanwhile prior to PutProcessData(), the data received from the PROFINET controller is put into the PROFIBUS output data buffer and then the data ready indicator is cleared. Likewise in PROFINET, PnetGbyInput_Data is filled by new data received from the PROFINET controller and ready for sending to PROFINET matching to PROFINET controller and is indicated as ready for sending to PROFIBUS master in appInData task. Meanwhile, data received from PROFIBUS master is sent to PROFINET controller and the data ready indicator is cleared.

The buffer list update is processed in critical section to avoid concurrent accesses to the shared resources that lead to unexpected or erroneous behavior. To ensure exclusive use of critical sections, a synchronization mechanism is implemented at the entry and exit of the buffer list update. The buffer list is a circular buffer and updated in a wraparound manner. Each buffer list is tracked by its read and write index to ensure valid data access.

2 Getting Started Hardware and Software

2.1 Hardware

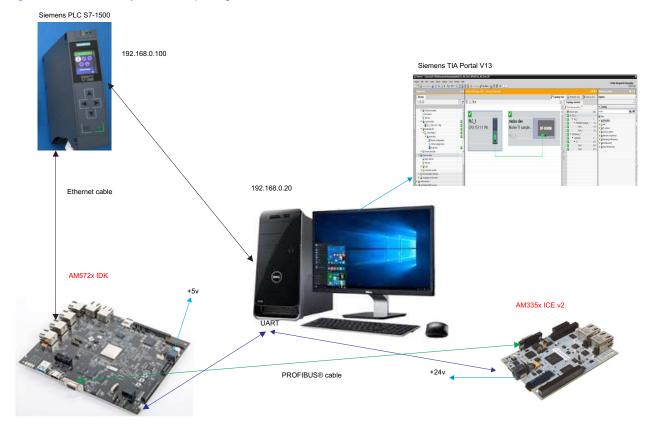
Required hardware includes:

- 1. AM572x IDK
- 2. AM3359 ICE v2
- 3. Siemens S7-1500 PLC
- 4. PROFIBUS connector or cable
- 5. Ethernet cables
- 6. Two 1GB+ SD card
- 7. 5-V DC power supply
- 8. 24-V DC power supply
- 9. Two mini USB cables
- 10. XDS-560M JTAG emulator (optional)



2.1.1 Hardware Connections

Figure 7 shows the system setup diagram.



Copyright © 2017, Texas Instruments Incorporated

Figure 7. Hardware Connections

The AM572x industrial development kit (IDK) is a development platform for evaluating the industrial communication and control capabilities of Sitara AM572x processors for applications in factory automation, drives, robotics, grid infrastructure, and more.

Note the power supply for the AM572x IDK is 5-V DC while other Sitara EVMs may use 12-V DC.

The FTDI USB port is used as UART console port (baud 115200, 8N1). XDS560-M JTAG emulator is optional as the image can be booted up from SD card.

The AM3359 industrial communications engine (ICE) is a development platform targeted for systems that specifically focus on the industrial communications capabilities of the Sitara AM335x ARM Cortex-A8 processors. The AM335x ARM Cortex-A8 processors also integrate the PRU-ICSS, which is discussed in detail in the beginning of this guide for AM572x processors. Unlike AM57x, the AM335x processors only have one instance of the PRU-ICSS, so they are able to support one industrial protocol at a time (PROFIBUS or Ethernet-based protocols are both supported for AM335x). AM335x enables low-footprint designs with minimal external components and with best-in-class, low-power performance.

Siemens S7-1500 PLC provides PROFINET IO (two-port switch) as standard interface.



2.2 Software

Required software includes:

- 1. CCS Linux version 6.1.2
- 2. Processor SDK RTOS 2.0.2.11
- 3. PRU-ICSS industrial software for Sitara processors 1.0.0.0
 - PRU-ICSS-PROFINET-SLAVE
 - PRU-ICSS-PROFIBUS-MASTER
- 4. ProfibusMaster_ProfinetSlave_01.00.00.00
- 5. SYSBIOSSDK-IND-SITARA 02_01_03_02
- 6. TIA Portal v13+

2.2.1 Install Code Composer Studio™

CCS version 6.1.2 is available at the *Download CCS* Wiki page.

2.2.2 Install Processor SDK RTOS

Processor SDK TI-RTOS is available at PROCESSOR-SDK-RTOS-AM7X 02_00_02_11.

2.2.3 Install PRU-ICSS Industrial Software

PRU-ICSS industrial software is available at PRU-ICSS-INDUSTRIAL-SW.

- 1. Install PROFIBUS master from PRU-ICSS-PROFIBUS-MASTER.
- 2. Install PROFINET slave from PRU_ICSS_PROFINET_Slave.

2.2.4 Install and Build ProfibusMaster_ProfinetSlave CCS Project

- 1. Download ProfibusMaster_ProfinetSlave_01.00.00.00 from http://git.ti.com. Note the future releases of Processor SDK RTOS will include the software.
- 2. Copy ProfibusMaster_ProfinetSlave_01.00.00.00\profibus_master_update\Profibus_AM57x_bsp.c to PRU-ICSS-Profibus_Master_01.00.00.00\examples\profibus_master directory.
- 3. Copy ProfibusMaster_ProfinetSlave_01.00.00\profibus_master_update\swpruss.h to PRU-ICSS-Profibus_Master_01.00.00.00\protocols\profibus_master\include\directory.
- 4. Import ProfibusMaster_ProfinetSlave_01.00.00.00 to CCS and build.
- Follow Processor SDK RTOS BOOT AM57x Wiki page to create a bootable image, and copy to the SD card. A JTAG emulator may also be used to download the ProfibusMaster_ProfinetSlave_01.00.00.out from ProfibusMaster_ProfinetSlave_01.00.00.00\Release.

2.2.5 Install TIA Portal for Siemens S7-1500 PLC Configuration

TIA Portal software from Siemens is used to configure S7-1500 PROFINET PLC. A trial version and service pack is available at:

- SIMATIC STEP 7 (TIA Portal) V13 TRIAL Download
- Service Pack 1 for SIMATIC STEP 7 V13 including PLCSIM (TIA Portal)

Refer to the Getting Started SIMATIC S7-1500 TIA Portal guide.

2.2.6 Install PROFIBUS® Slave Prebuilt Image for AM3359 ICEv2 Board

Install SYSBIOSSDK-IND-SITARA 02_01_03_02. The PROFIBUS slave prebuilt image is available in the software from SYSBIOSSDK-IND-SITARA.

14



3 Test Setup

Figure 8 shows the hardware connection diagram to setup the test environment.

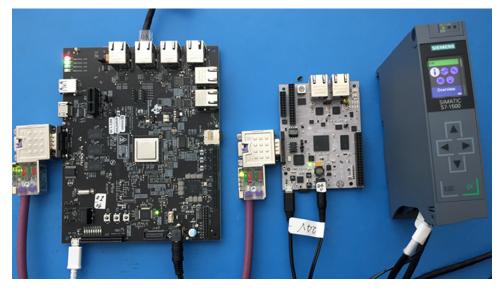


Figure 8. Test Setup

3.1 Setup PROFIBUS® Slave on AM3359 ICE v2 Board

Follow the Wiki pages below to set up PROFIBUS slave on AM3359 ICE v2 board with the prebuilt image installed in section 2.2.6

- SYSBIOS Industrial SDK 02.01.03 User Guide Wiki page
- PROFIBUS DP slave demo setup on AM335x Wiki page

NOTE: Set jumpers between pins 2 and 3 on J18 and J19 to select ICSS mode. Set jumper on J7, J8 (pin 1 and 2), and J10. Remove jumper on J6.

3.2 Configure Siemens S7-1500 PLC

Refer to the System overview SIMATIC S7-1500 for S7-1500 PLC setup details.

NOTE: Because PROFINET uses TCP/IP for configuration, parametrization, and diagnostics, an IP address needs to be programmed in the PLC. For real-time data process, PROFINET uses physical MAC address for more deterministic performance.



In addition, a MOVE module is added into the PROFINET communication path between controller and device.

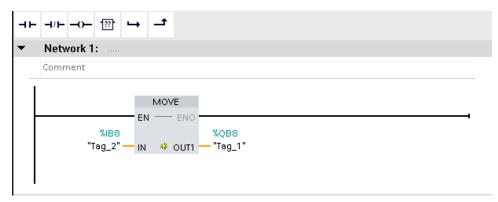


Figure 9. MOVE Module in TIA Portal

The PROFINET device description file (GSDML-V2.31-MOLEX_TI-AM57xxProfinet_SDK-20150723.xml) is under PRU-ICSS-Profinet_Slave_01.00.00.00\examples\profinet_slave\GSD.

3.3 Demonstration

- 1. Power on PROFIBUS slave: AM3359 ICE v2.
- Power on AM572x IDK with image loaded from JTAG emulator or SD card. Figure 10 shows the UART console log when the board is booted up from SD card. Wait until PROFINET connection is established then press any key to continue.

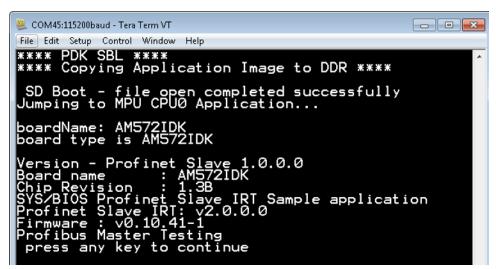


Figure 10. AM572x SD Boot Log



3. Check PROFINET connection status change from TIA Portal. Figure 11 shows the PROFINET controller offline before the communication is established between PROFINET controller and device.

👫 Siemens - C:\garrett\industrial\AM5	7xx_PN_Slave_IRTVAM57xx_PN_Slave_IRT	
Project Edit View Insert Online	Options Tools Window Help	Total
📑 隆 🗐 Save project 🚊 🖌 🖬 🛱	🗙 🐂 🖞 (주 🗄 🔃 🕼 🖳 💭 🕼 💋 Go online 🖉 Go offline 🍶 🖪 📑 💥 🚍 🛄	Tutai
Project tree 🛛 🕻 🕻	AM57xx PN_Slave_IRT > Devices & networks	_ @ =×
Devices		🚽 Topology view 🔒 Network view 📑 Device view
BOO E 🔤 🖻	Network 🖞 Connections HMI connection 👻 😇 🤑 🕰 😫	Network overvit 🗧 🕨
¥		A V Device
AMS7xx_PN_Slave_IRT		V Device V PLC.1
Add new device		= + PLC_1
devices & networks	PLC_1 molex-dev CPU 1511-1 PN Molex-Ti sample pp.nogm	GSD device_1
PLC_1 [CPU 1511-1 PN]	CPU 1511-1 PN Molex ⁻¹¹ sample DP-NORM PLC_1	molex-dev
Device configuration		
S Online & diagnostics		
 Program blocks Add new block 	PN/E_1	
Add new block		
Technology objects		
External source files		
PLC tags		
PLC data types		
Watch and force tables		-
Online backups		<u>*</u>
Traces		
📴 Program info		
Device proxy data		
PLC alarms		
Text lists		
Local modules Distributed I/O		
Gommon data		
Documentation settings		
Languages & resources		
Conignages are sources		
Card Reader/USB memory		
		×
	< II	> 100% 🔹 🕂 🗧 < 🗉 >

Figure 11. PROFINET[®] Controller Offline

Figure 12 shows PROFINET controller online after the communication is established between PROFINET controller and device.

M Siemens - C:\garrett\industrial\AM57	xx_PN_Slave_IRTVAM57xx_PN_Slave_IRT	
Project Edit View Insert Online O	Jptions Tools Window Help	Total
📑 🎦 🗔 Save project 💻 🗶 🛍 📬	🗙 🖒 ± (# ± 🐁 🗓 📓 📓 💋 Go online 🖉 Go offine 👔 🖪 🕞 🛠 😑 🕕	rotan
Project tree 🛛 🕄 📢	AM57∞ PN Slave IRT → Devices & networks	_ 2 = ×
Devices		Topology view 🔥 Network view 🕅 Device view
BOO E 🗄	💦 Network 🖞 Connections HMI connection 👻 🖾 🤫 🖽 🔍 ±	Network overvit 4 >
2		A Device
AMS7xx_PN_Slave_IRT		Vevice
Add new device		V + rtt_1
Devices & networks	PLC_1 molex-dev CPU 1511-1 PN Molex-II sample ne wnaw	GSD device_1
🔋 💌 🚰 PLC_1 [CPU 1511-1 PN] 🛛 🗹		✓ ► molecdev
Device configuration	PLC_1	
Online & diagnostics	—	
👻 🙀 Program blocks 🛛 🔵	PINE 1	
Add new block	(1111-1)	
🖀 Main [081] 🛛 🔵		
Technology objects		
External source files		
🕨 🎝 PLC tags 🛛 🔍		7
PLC data types		-
Watch and force ta		1
Online backups		-
Traces		
Program info		
Device proxy data		
PLC alarms		
Local modules In Distributed I/O		
Online card data		
Common data		
Documentation settings		
Languages & resources		
Online access		
Card Reader/USB memory		×
	K III > 100%	V 🔄 K II 🔹 V

Figure 12. PROFINET[®] Controller Online



Test Setup

- 4. Configure PROFIBUS master. Figure 13 shows how to configure PROFIBUS master.
 - Select baud rate, 1 through 8.
 - Enter 1 for number of slave station.
 - Enter 9 for slave station address.

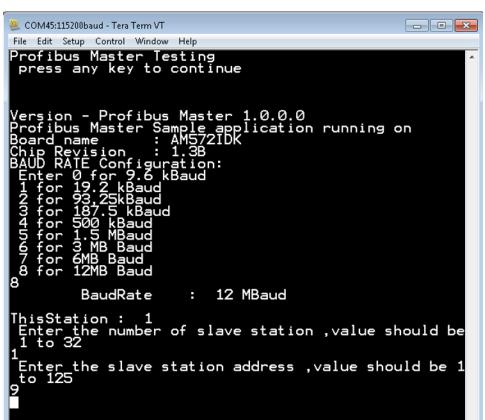


Figure 13. Configure PROFIBUS® Master on AM572x



🖲 COM45:115200baud - Tera Term VT File Edit Setup Control Window Help Profibus Master Testing press any key to continue Version - Profibus Master 1.0.0.0 Profibus Master Sample application running on Board name : AM572IDK Chip Revision : 1.38 BAUD RATE Configuration: Enter 0 for 9.6 kBaud 1 for 19.2 kBaud 2 for 93,25kBaud 3 for 187.5 kBaud 4 for 500 kBaud 1212412020 for for for kBaud MBaud 1.5 MBaud 3 MB Baud 6MB Baud 12MB Baud for for Ω 12 MBaud BaudRate ThisStation : 1 Enter the number of slave station ,value should be 1 to 32 Enter the slave station address ,value should be 1 to 125The slave station 1 address is 9 SlaveAdd : 9, 0, 0 connecting board... Result of ConnectBoard:0x00 LoadBusParameter:0K Event: IN-RING InitDPMaster:OK LoadSlaveParameters:OK InitDPMaster:OK Event: IN-RING StartDataExchange:OK OutputsAreOK : OK (0x00) Profinet Slave IRT fc:0f:4b:9c:21:e0 Slave in state DataExchange: 9

Figure 14. PROFIBUS® Master State



Test Setup

www.ti.com

6. Turn on monitoring of programmable module: MOVE and check the data update in both input and output.

	157	'x>	<u>د</u> ا	PN_Slave_IRT → PLC_1	CPU 1511-1 PN] 🔸	Program blocks	► Main [OB1]
ð	Å.	ð	1	\$ # % E 🗄 💳 🕏) 🗷 ± 🚨 ± 🖃 😥	¢° 💊 🖑 🐖	1 🕸 📭 📲 🚱 🕎 🔢
	M						· · · · · · · · · · · · · · · · · · ·
		N	lar	me	Data type	Default value	Comment
		•	•	Input			
		•	•	Initial_Call	Bool		Initial call of this OB
				Remanence	Bool		=True, if remanent data are available
		•	•	Temp			
				«Add new»			
		•	•	Constant			
		ł		<add new=""></add>			
_	-	-					
•	-	-1	-	-o- 🕾 🛶 🛨			
,		ы.		work 1:			
		Co	mr	ment			
	ī.						
	L			MOVE			
	F	_		EN EN	0		
	L			16#A3	16#A3		
	L			%IB8	%QB8		
	L			"Tag_2" — IN 🛛 🗰 OUT	1 — "Tag_1"		
	L				_		
	L						
	۰.						



7. Verify the received data update in PROFIBUS slave on AM3359 ICE v2.

Figure 16. PROFIBUS® Slave Data Update



4 Design Files

4.1 Bill of Materials

To download the Bill of Materials (BOM) for each board, see the design files at TIDEP-0075.

4.2 Software Files

To download the Assembly Drawings for each board, see the design files at TIDEP-0075.

5 Related Documentation

- 1. Texas Instruments, *AM572x Sitara™ Processors Silicon Revision 2.0, 1.1,*, Technical Reference Manual (SPRUHZ6)
- 2. Texas Instruments, *PROFINET on TI's Sitara™ AM335x processors*, Marketing White Papers (SPRY252)
- 3. Texas Instruments, *Certified PROFINET IRT v2.3 Device With 1-GHz ARM Application Processor,*, TI Design (TIDUAK0)
- 4. Texas Instruments, PROFIBUS on Sitara Processors, Marketing White Papers (SPRY155)

5.1 Trademarks

Sitara, Code Composer Studio are trademarks of Texas Instruments, Inc.. ARM, Cortex are registered trademarks of ARM Limited. EtherCAT is a registered trademark of Beckhoff Automation GMbH. EtherNet/IP is a trademark of ODVA, Inc.. PROFINET, PROFIBUS are registered trademarks of PROFIBUS and PROFINET International (PI). All other trademarks are the property of their respective owners.

6 About the Author

GARRETT DING is a software applications engineer for the embedded processing group at Texas Instruments, where he is responsible for developing reference design solutions and providing technical support to customers for the industrial segment. Garrett earned his Master of Science in Electrical Engineering (MSEE) from NanJing University of Science and Technology, China.

ELLEN KOU is a product marketing manager at Texas Instruments, where she is responsible for applications of ARM processors in industrial automation market segments. Ellen earned her Master of Engineering in Electrical Engineering (MEn) from Texas A&M University in College Station, Texas.

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY 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 products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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 RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your noncompliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/stdterms.htm), evaluation

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2017, Texas Instruments Incorporated