

How-To and Troubleshooting Guide for PRU-ICSS PROFIBUS

Paula Carrillo

ABSTRACT

This troubleshooting guide focuses on how to test and debug PRU-ICSS PROFIBUS. Tips, tricks and debugging techniques are offered.

Contents

1	PRU-ICSS PROFIBUS Overview	2
2	Demo Setup	3
3	Hardware Debug Tips	5
4	PRU-ICSS Debug Via Code Composer Studio (CCS)	5
5	Common Issues	6
6	FAQ	7
7	References	7

List of Figures

1	Software Image	3
2	Siemens TIA Portal	7

List of Tables

1	PRU-ICSS Resources Used for PROFIBUS	3
2	Tx Enable Pins	4
3	ISO MINI EVM as PRU-ICSS PROFIBUS Transceiver	4
4	Slave DP Status Information	5

Trademarks

Sitara, Code Composer Studio are trademarks of Texas Instruments. All other trademarks are the property of their respective owners.



1 PRU-ICSS PROFIBUS Overview

TI's Sitara[™] System-on-Chip (SoC) family offers Programmable-Realtime Unit Industrial Communications Subsystem (PRU-ICSS), which enables the integration of real-time industrial communications protocols and eliminates the need for an external ASIC or FPGA.

TI's offers several PRU-ICSS Industrial software protocols, which are adds-on to TI's Processor SDK. PROFIBUS slave and master are among them.

PROFIBUS is a well stablish technology, with more than 50 million nodes installed worldwide. The PROFIBUS industrial field bus is used to connect controllers to remote input/output units, sensors, actuators and inter-networking components.

1.1 **PROFIBUS Components**

Each PROFIBUS node has three components: the electrical layer, the data link layer and an application layer. The electrical layer is implemented using RS-485 or fiber-optic.

The data link layer is called Fieldbus Data Link (FDL) and it implements master-slave communication. The application layer is one of DP-V0, DP-V1, or DP-V2. And it supports messaging between the PROFIBUS nodes.

It is important to identify these layers for trouble shooting an issue. Always check that the connector and cables are functional before further debugging.

1.2 PRU-ICSS PROFIBUS Software Architecture Overview

The PROFIBUS real-time frame handler (Fieldbus Data Link or FDL) is encapsulated in the Sitara programmable real-time unit (PRU) subsystem. 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 7, 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.



www.ti.com



Figure 1. Software Image

For more details on TI's PROFIBUS implementation, see the *PROFIBUS on Sitara Processors White Paper*.

1.3 PRU-ICSS Resources Used for PROFIBUS

Table 1. PRU-ICSS Resources Used for PROFIBUS

Resources Used for Implementing PROFIBUS	Function
PRUx	Host Communication
PRUy	UART Communication
UART (ICSS)	TX/RX PROFIBUS messages

2 Demo Setup

PRU-ICSS PROFIBUS slave and PRU-ICSS PROFIBUS master are add-on packages to TI-RTOS Processor SDK (PSDK). Download the recommended TI-RTOS PSDK version on PROFIBUS slave system requirements or PROFIBUS master system requirements in order to assure correct functionality. Mix and match between TI-RTOS PSDK and PRU-ICSS PROFIBUS version could lead to incorrect behavior.

2.1 ICEv2 Board Setup

For selecting PROFIBUS, set jumper on J7, J8 (pin 1 and 2) and J10. And remove Jumper on J6, if any.



2.2 Enabling PROFIBUS Slave Support in Boards Without Inbuilt RS485 Connector

An external RS485 isolated transceiver can be added to validate PROFIBUS support on the boards without inbuilt connector. Currently, those boards are AM437x IDK, K2G ICE and AMIC11x. Connections from the transceiver are made to the prx_uart0_rxd, prx_uart0_txd and prx_pru0_gpoy via expansion connectors. prx_pru0_gpoy is used as Tx Enable pin. Tx enable pin is configurable through shared memory (L3 RAM) via parameter PruTxGpioNum at offset 0x1D94. Additionally, "g_pL2GlobVar->sel_GPO_GPIO" selects between PRU-ICSS GPO and SoC GPIO. Users should be aware there is a potential limitation at 12Mbaud when using SoC GPIO mode, due to increased access latency from PRU. Table 2 has the list of Tx Enable pins used by the applications.

Table 2. Tx Enable Pins

Board	GPO Used	Ball Number
AMIC11xICE	PR1_PRU0_PRU_R30_5	C13
AM437x IDK ICSS0	PR0_PRU0_GP01	N22
AM437x IDK ICSS1	PR1_PRU0_GP06	C20
K2G ICE	PR0_PRU0_GP07	D4

2.3 ISO MINI EVM as PRU-ICSS PROFIBUS Transceiver

ISO MINI EVM enables PROFIBUS on TI's supported boards, which lack of an inbuilt RS485 connector. The ISO MINI EVM details can be found here ISO1176T Small EVM. TB1 is the logic side of the Transceiver, and should be connected to the board. Table 3 gives a description of the TB1 pins. TB2 corresponds to PROFIBUS interface, pins A and B should be connected to a PROFIBUS network.

Table 3. ISO MINI EVM as PRU-ICSS PROFIBUS Transceiver

Pin	Description
Vin	3.3 V from the board
GND1	Ground from the board
R	UART Rx pin
De	Tx Enable pin
D	UART Tx pin

2.4 How to Build and Run PRU-ICSS PROFIBUS Slave

Use the following links with instructions on how to build and run PROFIBUS demo:

- Code Composer Studio[™] (CCS) Project creation: *Industrial Protocol Package Getting Started Guide* wiki
- Running and configuring PROFIBUS slave wiki

Inputs can be changed from the slave side by pressing 'I' on UART console.

2.5 How to Build and Run PRU-ICSS PROFIBUS Master

The following links provide instructions on how to build and run PROFIBUS demo:

- CCS Project creation: Industrial Protocol Package Getting Started Guide wiki
- Running and configuring PROFIBUS Master wiki

You can select baud rate, and configure the number of slaves. Note that current PROFIBUS master demo application (evaluation version) supports only TI PROFIBUS slaves and a maximum number of three slaves.

www.ti.com

2.6 **PROFIBUS** Diagnostics

- PROFIBUS slave application sends static diagnostic data to the master using the UART console option 'S'. Option 'E' sends extended diagnostic message. Console option 'C' clears diagnosis.
- Use 'Getdiag' command in PROFIBUS master for getting diagnostics of all the slaves

2.7 Slave DP Status Information

DP Status indicates the operational state and communication data exchange status, between master and slave. 'D' gives the DP status - 0x80 or 0x81 indicates the data exchanges, and 0x0 indicates that data exchange is stopped. The DP status value can be decoded as shown in Table 4.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Slave is in data exchange	Output data received	Slave is in sync mode	Slave is in freeze mode	Baud rate found	Leave master, Slave has leaved the data exchange state set only flag, which must be reseted by the user	State of Clear_Data Bit of the laste received Global_Control (State of the master: CLEAR or OPERATE)	Timeout (DP Watchdog expired)

Table 4. Slave DP Status Information

3 Hardware Debug Tips

- You can connect a scope in order to measure activity of RX, Tx and Tx enable. Also, baud rate can be checked with the help of a scope.
- Another check point is to validate the voltage supply of the RS-485 transceiver.
- It is also important to make sure the termination resistors are added to the PROFIBUS line according to the PROFIBUS standard. For more information, see *PROFIBUS Electrical Layer solution*.
- PROFIBUS trace hardware tools such as ProfiTrace or PROCENTEC are good diagnostic tools for testing cables, measuring signals and checking the network.

4 PRU-ICSS Debug Via Code Composer Studio (CCS)

Instructions on how to load and debug and ARM application in CCS are shared in Industrial Protocol Package Getting Started Guide - CCS debug section. PRU-ICSS PROFIBUS firmware is distributes as a binary. No source code is included in PRU-ICSS PROFIBUS software package. Debugging PRU-ICSS PROFIBUS firmware would need TI's experts help. E2E forum is a great place to get help. However, in some cases, you might need to gather PRU's information in order for TI's experts to help.

The following steps show how to connect and debug PRU cores.

1. Right Click on the PRU_0 or PRU_1 and select "Connect Target". Then suspend and view Disassembly.

▲ P Texas Instruments XDS100v2 USB Debug Probe_0/PRU_0 (Suspended) ≡ 0x0000C0 (no symbols are defined)

▲ P Texas Instruments XDS100v2 USB Debug Probe_0/PRU_1 (Suspended)
■ 0x000068 (no symbols are defined)

TEXAS INSTRUMENTS

www.ti.com

2. Disassembly view in CCS. Step through can be used

IIII Registers	E Disassembly	/ 😂 🛷 Search 💊 Breakpoints
		Enter location h
000048:	090597F9	LSL R25, R23.w0, 5
00004c:	244000E4	LDI R4, 16384
000050:	00E4F9F9	ADD R25, R25, R4
000054:	00FAF9F9	ADD R25, R25, R26
000058:	240350E2	LDI R2, 848
00005c:	240001E1	LDI R1, 1
000060:	0897E1F8	LSL R24, R1, R23.w0
000064:	E0E23A98	SBBO &R24.b0, R26, R2, 4
♦ 000068:	C91FFF00	QBBC 0, R31, 31

- 3. PRU-ICSS code example. Below source code maps Disassembly code with source code.
 - **NOTE:** PRU-ICSS PROFIBUS assembly source code is TI internal, no distributed. It is shown here only as a reference.

	: //***********************************
	: // MAIN LOOP
	: //**********************
	: main_loop:
	: // Are Events pending? (Timer, RX, TX, Host)
0xc91fff00	: QBBC main_loop, EventReg, EventBitPos
	:
	: //**********************
	: // EVENT HANDLER
	: //***********************************
	: EVENT_HANDLER: // Jumped to once an interrupt has occured in R31.
	: //This will retrieve the interrupt number
	: // Initialize pointer to priority index register location in the INTC
	: // Retrieve interrupt number from the INTC registers
0xf1003680	: LBBO R0, HOSTINTPRIIDX1, 0, 4

5 Common Issues

Baudrate not locking. *timeout_timer()* is used to check StartDelimiterCounter variable in L3 Shared memory structure. PRU FDL increments this counter whenever it sees a valid Start Delimiter (SD) on the serial bus. If StartDelimterCounter is zero then this means PRU FDL did not lock to the baudrate set on the bus by the master.

The following procedure is done in order to check next baudrate option:

- 1. Disable PRU0 and PRU1
- 2. Set the next baud rate using PROFIBUS driver API
- 3. Re-enable PRU1
- 4. Set Slave Address, Min TSDR and Current Baudrate at PRU FDL using CMD_SET_VALUE_REQ
- 5. Re-enable PRU0
- 6. Restart timer

6

Users, who has access to PROFIBUS stack from Molex, can fix baud rate in timeout_timer() function. timeout_timer() is located in fdl.c file, and uses set_value() API to set Baud rate.



www.ti.com

6 FAQ

• Which master can be used for testing?

- Any PROFIBUS master can be used, however, for a quick sanity test and demo setup users can implement and run TI's PRU-ICSS PROFIBUS master on any of supported platforms. At the time writing this document those supported platforms are AM57x and AM335x.
- Additionally, you can use the PROFIBUS tool kit analyzers, such as ProfiTrace analyzer. Siemens also offers different PROFIBUS controllers. One popular combination is S7 PLC, plus a PPROFIBUS communication module. Simatic STEP7 is used for programing Siemens devices. There are some tutorials used to get familiar with Siemens TIA Portal.



Figure 2. Siemens TIA Portal

• PROFIBUS slave hang. How to debug it?

- Connecting JTAG CCS can help to debug both ARM and PRU-ICSS issues. For general tips and tricks, TI-RTOS tips and tricks.
- Check in which part of ARM application the code hangs. Also, check if PRUs code are in a forever loop. For further details, see Section 4.
- What is supported/ not supported?
 - For a list of supported features, fixed issues among other information, see the PRU-ICSS PROFIBUS slave Release Notes wiki and the PRU-ICSS PROFIBUS master Release Notes wiki.
- How to detect if host CPU crash?
 - PROFIBUS Slave Application uses Watchdog timer to detect the CPU crash. This support is enabled by deafult. User can disable the feature by removing the macro "ENABLE_WATCHDOG_SUPPORT" in the file "profi_misc.h". The time out can be configured by changing the value of the macro "WATCHDOG_EXPIRE_TIME". Watchdog task is created if the macro ENABLE_WATCHDOG_SUPPORT is enabled. The task will continuously reset the PDI Watchdog Timer in the PRUSS periodically so that the time out does not occur. The profibus firmware will be polling the timeout event and once the time out is detected, the Data exchange will be disabled.

7 References

- PROFIBUS on Sitara Processors White Paper
- Industrial Protocol Package Getting Started Guide wiki
- Running and configuring PROFIBUS Master wiki
- PROFIBUS Electrical Layer solution
- PRU-ICSS PROFIBUS slave Release Notes wiki
- PRU-ICSS PROFIBUS master Release Notes wiki

IMPORTANT NOTICE AND DISCLAIMER

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

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2018, Texas Instruments Incorporated