

PRU-ICSS Getting Starting Guide on Linux

*Jason Reeder**Catalog Processors*

ABSTRACT

This application report provides a getting started guide for the PRU-ICSS of the Sitara™ family of devices while running Linux on the Arm® cores.

Contents

1	Overview	2
2	PRU-ICSS Getting Started With Linux	3
3	More Information	4
4	Frequently Asked Questions	5
5	References	6

Trademarks

Sitara, Code Composer Studio are trademarks of Texas Instruments.

Arm is a registered trademark of Arm Limited.

EtherCAT is a registered trademark of Beckhoff Automation. GmbH, Germany.

PROFIBUS, PROFINET are registered trademarks of PROFIBUS International.

All other trademarks are the property of their respective owners.

1 Overview

1.1 PRU-ICSS Description

The Programmable Real-time Unit and Industrial Communication SubSystem (PRU-ICSS) consists of dual 32-bit RISC cores (Programmable Real-time Units, or PRUs), shared data, instruction memories, internal peripheral modules, and an interrupt controller (INTC). The programmable nature of the PRU, along with its access to pins, events and all system-on-chip (SoC) 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 SoC.

Figure 1 shows the details of the PRU-ICSS found in the AM335x 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. The interaction between processors is determined by the nature of the firmware loaded into the PRU's instruction memories.

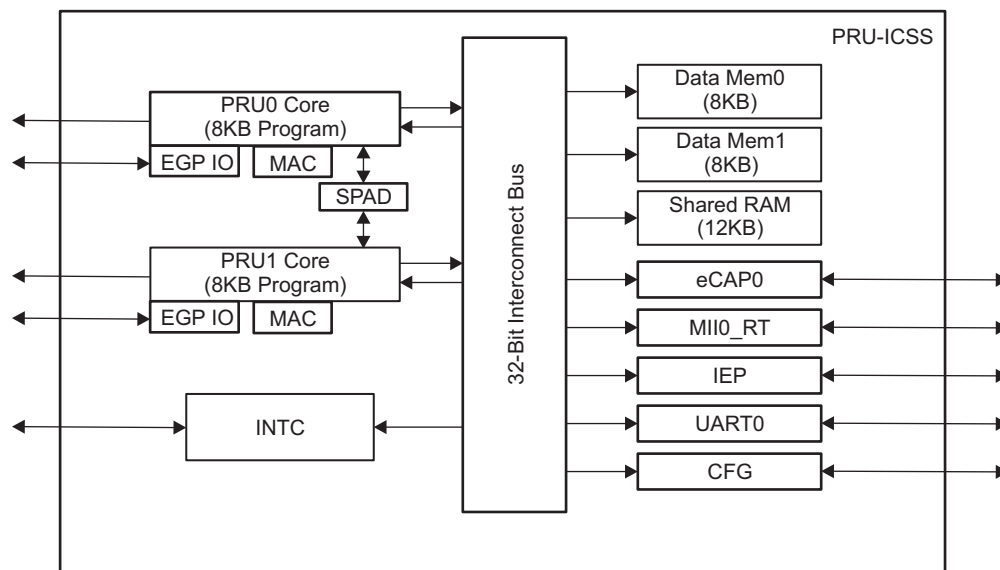


Figure 1. PRU-ICSS Block Diagram

1.2 Devices Containing the PRU-ICSS

The PRU-ICSS can be found throughout the Sitara family of devices from Texas Instruments (TI). There are slight variations in the subsystem across devices so this table was created to show the capabilities of each device's subsystem in one place.

The table mentioned above is a great reference for the following information (and more):

- The number of PRU-ICSS's in a device (some devices have more than one)
- The size of the internal memories (instruction RAM, data RAM, and shared RAM)
- The number of general purpose input/output pins available externally

1.3 What Can the PRU-ICSS Do?

1.3.1 Real-Time Industrial Communication

The PRU-ICSS was designed with real-time industrial communications in mind. Ethernet, Fieldbus, interface, and redundancy protocols are available for the PRU-ICSS, including certified solutions for EtherCAT®, PROFIBUS®, PROFINET®, HSR and PRP among others. The PRU-ICSS can support multiple protocols on the same hardware using firmware supplied by TI and stacks from industry partners.

For industrial communication protocols, TI, or one of its industry partners, will provide the PRU firmware to be loaded. In this type of use case, the PRU-ICSS appears to the high level operating system (Linux or RTOS) as another Ethernet peripheral.

1.3.2 General Purpose Use Cases

The PRU cores were created to be non-pipelined, single-cycle execution engines. This means that all instructions (except for memory reads and writes) complete in a single cycle. Since there is no pipeline, the PRU cores are completely deterministic: the PRU cores will not rearrange instructions during execution. If the PRU firmware has a critical loop that was written to complete in 15 cycles, it will complete in 15 cycles every single time.

The PRU-ICSS can also read from or write to external pins in a single cycle using a PRU core register. With a PRU core running at 200 MHz, this means that an external signal change can be detected in as little as 5 nanoseconds! This also means that the PRU can toggle external pins with a granularity of 5 nanoseconds.

The combination of deterministic PRU cores along with direct access to GPIO means that the PRU-ICSS opens up a lot of interesting uses cases to Sitara SoCs.

1.4 PRU-ICSS Usage Environments

In the context of the SoC, there are three environments in which the PRU-ICSS is used: Code Composer Studio™ (CCS), TI RTOS running on the Arm core(s), or Linux running on the Arm core(s).

1.4.1 Code Composer Studio

Code Composer Studio (CCS) is an Integrated Development Environment (IDE) that supports TI's Microcontroller and Embedded Processors portfolio. Using CCS, the PRU-ICSS can be loaded, run, and fully debugged just like any other core in the SoC. This environment is very useful during the development and debug stages of creating a stand-alone PRU firmware.

1.4.2 TI RTOS

TI RTOS, based on SYS/BIOS, is Texas Instrument's Real-Time Operating System (RTOS) that can be used as the high-level operating system on the processors of Sitara devices (Cortex-A or C66x DSP). In this environment, an RTOS driver is provided to load the PRU-ICSS memories and run the PRU cores. The PRU firmware files are expected to be built and then converted into an array format that the [RTOS driver](#) can use.

1.4.3 Linux

TI creates its own Linux distribution and packages it into the [Processor SDK Linux](#). In this environment, the [Linux RemoteProc](#) framework treats the PRU cores inside the PRU-ICSS as [remote processors](#). The PRU firmware files are expected to be built into elf binaries and placed into the Linux file system. The remainder of this application report will focus on using the PRU-ICSS in the Linux environment.

2 PRU-ICSS Getting Started With Linux

As mentioned previously, there are two types of use cases for the PRU-ICSS while running Linux on the Arm core:

- The first type of use case utilizes TI-provided PRU firmware for industrial communication protocols, such as HSR or PRP.
- The second type of use case utilizes building block examples to create general purpose applications that take advantage of the strengths of the PRU-ICSS.

Both of these types of use cases require the [Processor SDK Linux](#).

2.1 TI Provided PRU-ICSS Firmware

In the Processor SDK Linux, TI provides prebuilt PRU-ICSS firmware to implement 100Mbps Ethernet ports as well as the redundancy protocols HSR and PRP. In addition to the PRU firmware provided, TI has also created and provided the Linux drivers necessary to use the PRU Ethernet ports from user space applications.

To learn more about the PRU Ethernet firmware and how to test it, see [PRU-ICSS Ethernet](#).

To learn more about the HSR and PRP redundancy implementation, see [Processor SDK Linux HSR PRP](#).

2.2 General Purpose Firmware

Before creating general purpose Linux PRU firmware, you must understand the basics of how the Linux RemoteProc framework works. The RemoteProc Linux driver is responsible for taking the PRU firmware from the Linux filesystem, parsing it for any resources that it must provide for the PRU (for example, interrupts or shared memory), loading the firmware into PRU instruction memory and data memory, and then running the PRUs.

For more information on the RemoteProc driver and how it works, see [RemoteProc](#). A key thing to understand is that the resource table is necessary for the RemoteProc driver to work, even if it is empty.

User applications can require communication between the PRUs and the Arm core. Linux provides a method for this communication called RemoteProc Messaging (RPMMsg). RPMMsg fits into the RemoteProc framework. TI provides a RPMMsg library and getting started examples to demonstrate how to use RPMMsg. These examples showcase the PRU firmware using its resource table to request specific interrupt mappings and shared memory to implement the communication channel.

To learn more about how RPMMsg works, see [RPMMsg](#).

For a quick start guide that walks through the RPMMsg out-of-box example as well as how to rebuild/reload the example, see [RPMMsg Quick Start Guide](#) wiki.

Once RemoteProc and RPMMsg are understood, a good next step is to check out the TI-provided [PRU-ICSS Hands-on Labs](#). These labs are written for a BeagleBone Black with a PRU Cape attached, but the concepts apply to the PRU-ICSS across the Sitara family. Labs 1-3 require CCS and do not use Linux. If CCS is of no interest, Labs 1-3 can be skipped and you can start from [Lab4](#).

The PRU Software Support Package is the central location for all the previously mentioned general purpose firmware source code and libraries. The package is located in the 'example-applications/pru-icss-x.y.z' folder of the Processor SDK Linux installation. The git repository for the package is also available at [PRU Software Support Package Git Repository](#). This package contains building block PRU examples for Sitara devices along with the necessary header files and libraries. The examples in the PRU Software Support Package may be used as a starting point for developing new general purpose firmware.

3 More Information

3.1 PRU C Compiler and Assembler

PRU firmware can be written using C code or assembly. C code is more approachable to new users, but it may not get the absolute best performance achievable. The recommendation is to use C code in your firmware as much as possible, and then see how well the provided C compiler optimizes your code. If further optimizations are necessary, then assembly code can also be used.

TI provides the PRU C compiler in the PRU Code Generation Tools package available in the 'linux-devkit/sysroots/x86_64_arago-linux/usr/share/ti/cgt-pru/' directory of the Processor SDK Linux. The C compiler is available as an add-on package for CCS. The compiler may also be downloaded from this link: [PRU Code Generation Tools](#).

The PRU C compiler user guide is available at: [PRU Optimizing C/C++ Compiler v2.2 User's Guide](#).

The PRU assembler user guide is available at: [PRU Assembly Language Tools v2.2 User's Guide](#).

The PRU assembly instructions list is available at: [PRU Assembly Instruction User's Guide](#).

3.2 TI Designs

TI Designs are system-level hardware and software references for specific use cases. The following TI Designs are PRU-ICSS examples where Linux is running on the Arm core.

3.2.1 Industrial Protocol TI Designs (TI Provided Firmware)

[High-Availability Seamless Redundancy Ethernet Reference Design for Substation Automation for Linux®](#) – This reference design implements a solution for high-reliability, low-latency network communications for substation automation equipment in Smart Grid transmission and distribution networks. It supports the high-availability seamless redundancy (HSR) specification in the IEC 62439 standard.

[Parallel Redundancy Protocol Ethernet Reference Design for Substation Automation on Linux](#) – This Linux®-based reference design demonstrates high-reliability, low-latency network communications for substation-automation equipment in smart grid transmission and distribution networks. This reference design supports the Parallel Redundancy Protocol (PRP) specification in the IEC 62439 standard.

[OPC UA Data Access Server for AM572x Reference Design](#) – OPC UA is an industrial machine-to-machine protocol designed to allow interoperability and communication between all machines connected under Industry 4.0.

3.2.2 General Purpose TI Designs

[Flexible Interface \(PRU-ICSS\) Reference Design for Simultaneous, Coherent DAQ Using Multiple ADCs](#) – This TI design showcases the PRUs ability to implement parallel SPI ports to connect to (6) 8-channel ADCs simultaneously.

[Simple PID Control Reference Design With PRU®-ICSS Through Web Interface](#) – This design implements a simple PID loop for a small motor in the PRU. The data (set point, current speed, constants, etc.) is communicated to the Arm core through RPMsg and displayed on a webpage.

[Thermal Printing with the PRU-ICSS on the BeagleBone Black Reference Design](#) – A thermal printer driver is made using the GPIO of the PRU.

3.3 E2E Forum Support

Any questions not addressed by this PRU-ICSS Getting Started Guide for Linux can be posted on the [Sitara E2E Forums](#).

4 Frequently Asked Questions

This section includes a few of the most common questions while using the PRU-ICSS with Linux on the Arm core. For a comprehensive list of Frequently Asked Questions on the PRU-ICSS, see the [PRU-ICSS FAQ](#) wiki.

4.1 Do you Support the BeagleBoard Community Linux Distribution, the PASM Tool, or the Linux UIO Driver for PRU General Purpose Use Cases?

No, TI does not support these.

The BeagleBoard Linux Distribution is supported by the BeagleBoard community on their forums: [BeagleBoard Community Forums](#).

The PASM tool has been deprecated in favor of the PRU Code Generation Tools mentioned earlier.

The UIO driver for general purpose use cases has also been deprecated and is replaced by the RemoteProc driver.

4.2 Does the PRU-ICSS Support Interrupts?

Yes, but not in the same way that most cores support interrupts.

The PRU-ICSS contains an interrupt controller that can map 64 system events down to two flags that are set in a PRU core register (bits 30 and 31 in core register R31). The PRU core can then check each of these flags in a single cycle to see if an event has occurred. These flags can either be polled upon or checked periodically (dependent on what makes the most sense for the use case).

The PRU-ICSS interrupt controller does not support jumping the program counter of the PRU core to a pre-determined function when an event occurs.

4.3 What are the Read and Write Latencies to Different Memories?

The latencies associated with reading from and writing to different SoC memories are found in the [PRU Read Latencies](#) wiki.

5 References

- [BeagleBoard Community Forums](#)
- [PRU Optimizing C/C++ Compiler v2.2 User's Guide](#)
- [PRU Assembly Language Tools v2.2 User's Guide](#)
- [PRU Assembly Instruction User's Guide](#)
- [Flexible Interface \(PRU-ICSS\) Reference Design for Simultaneous, Coherent DAQ Using Multiple ADCs](#)
- [Parallel Redundancy Protocol Ethernet Reference Design for Substation Automation on Linux](#)
- [Thermal Printing with the PRU-ICSS on the BeagleBone Black Reference Design](#)
- [Flexible Interface \(PRU-ICSS\) Reference Design for Simultaneous, Coherent DAQ Using Multiple ADCs](#)
- [Simple PID Control Reference Design With PRU®-ICSS Through Web Interface](#)
- [Thermal Printing with the PRU-ICSS on the BeagleBone Black Reference Design](#)
- [PRU-ICSS FAQ](#) wiki
- [PRU Read Latencies](#) wiki

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 that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such 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 non-compliance 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/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

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