Migration Guide
Industrial SDK to
PRU-ICSS-Industrial & Processor SDK

Oct 20, 2017
Agenda

At a high level - What are the changes from Industrial 2.x to PRU-ICSS-Industrial SW plus Processor SDK

Processor SDK Overview

Changes from SYSBIOSSDK-IND-SITARA to PRU-ICSS-Industrial and Processor SDK

Migrating an Application from ISDK to PRU-ICSS-Industrial Processor SDK

Examples
  - Moving a starterware application to use an LLD
  - Modifying Board Library to Change UART Instance on AM335x
What is changing

Industrial package is modularized so that each industrial protocol has a separate add-on package that can be used with the Processor SDK RTOS SDK for any given SOC.

- Easier migration to custom boards because changes are consolidated in board software.
- Incremental software changes when moving to newer TI platforms like AM57xx and K2G.
- Improved maintenance of existing protocols
- Simplifies the addition of new protocols

Hardware interface through Chip Support Library and Low Level drivers in place of Starterware.

- Low Level Drivers allow for simpler integration with TI RTOS using OS abstraction layer (OSAL)
- Common API interface enables ease of migration for TI MCU customers moving to TI ARM or DSP platforms by maintaining the same API interface for common IPs like SPI, I2C, UART, etc.
- Continued support for AM335x and AM437x Starterware is provided for legacy development projects.
  - From starterware from 2.1.2.3 to the Processor SDK there are only a few changes few changes.
  - In the Processor SDK, starterware is built with makefiles instead of CCS projects.
  - Some AM335x and AM437x CSL is supported by starterware
  - Refer: [Processor SDK RTOS Migration Guide history](#)
Each Industrial communications protocol will be contained in a separate package

**Industrial SDK**
(AM335x, AM437x)

**PRU-ICSS-SW Industrial Library Model**
(AM335x, AM437x, AM57x)

Available at [www.ti.com/tool/PRU-ICSS-INDUSTRIAL-SW](http://www.ti.com/tool/PRU-ICSS-INDUSTRIAL-SW)
Why Change? - PRU-ICSS-SW Industrial Software Delivery

- Separate Industrial protocol packages provide:
  - Better support
  - Ease of migration to other platforms
  - Improved maintenance of existing protocols
  - Simplifies the addition of new protocols

- This change is transparent for stack engagement with third parties. No change is required from third parties.
Agenda

What are the changes from Industrial 2.x to PRU-ICSS-Industrial SW plus Processor SDK

**Processor SDK Overview**

What has moved where from ISDK 1 -> ISDK 2 -> Processor SDK & PRU-ICSS-Industrial

Migrating an Application from ISDK to PRU-ICSS-Industrial Processor SDK

Examples

- Moving a starterware application to use an LLD
- Modifying Board Library to Change UART Instance on AM335x
Processor SDK Overview
WHAT Is the Processor SDK?

- It is a unified software platform for TI embedded processors
  - Common API's interfaces across all supported platforms
  - Consolidates all platform specific software into a single SW component
- It provides out-of-the-box (OOB) benchmarks and demos.
- Releases are consistent across TI’s broad portfolio.
  - This allows software to be reused across devices.
- Supported Operating Systems (OS) are Linux and TI-RTOS.
- It provides well-defined APIs abstraction layers.
- Available as free download
  - Download Page for AM335x
  - Download page for AM437x
  - Download page for AM57x
Processor SDK RTOS SUPPORTS MULTIPLE SoCs

• Contains Chip Support Libraries (CSL), drivers, and basic board-support utilities.
• Includes source code and prebuilt libraries.
• Contains a basic networking stack.
• Contains bootloaders and boot utilities.
• Compiler: Linaro GNU compiler collection (GCC) tool chains.
Processor SDK RTOS Components

OS Software

- Applications: Implemented on top of the OS, and may be architecture dependent.
- Operating System dependent components: TI-RTOS kernel, OSAL, tools, utilities, drivers.

Non-OS Software

- Core-specific / OS-independent components: Optimized libraries.
- SoC-dependent / OS-independent components: Device and platform drivers.
SoC components: The basic low level components of the software consists of the CSL and LLD which are OS agnostic.

LLD: builds on top of the CSL layer and provides an interface to operate and control the peripherals on the device.

CSL: contains low-level register-level definitions and basic functionality to configure cores and registers on the device.
### Example Applications

- EtherCAT Slave
- Profinet RT/IRT Slave
- Ethernet/IP
- Profibus Master
- Profibus Slave
- HSR/PRP

### Industrial Libraries

- EtherCAT slave stack
- Profinet RT/IRT slave stack
- Ethernet/IP slave stack
- ProfiBus master stack
- ProfiBus slave stack
- HSR/PRP driver
- Industrial Board Lib

### PROCESSOR SDK

**OS Software**

- Inter-processor communication
- Framework Components
- OS ABSTRACTION LAYER (OSAL)
- NETWORK I/F Mgmt Unit (NIMU)

**SOFTWARE FRAMEWORK COMPONENTS**

- XDC Tools

**ALGORITHM LIBRARIES**

- DSPLIB
- IMGLIB
- MATHLIB

**LOW LEVEL DRIVERS (LLD)**

- EDMA3
- iCSS-EMAC
- PCIe
- PRUSS
- 12C
- EMAC
- USB
- SPI
- GPIO
- UART
- SD/MMC

**CHIP SUPPORT LIBRARY (CSL)**

**HOST**

- Device/platform-dependent
- OS-dependent
- OS Independent
- N/A for AM3 & 4

**HARDWARE**

- Texas Instruments

### Notes

- Not Applicable to AM335x and AM437x devices
- C66x optimized libraries.
- C-code source is provided so user can re-build libraries on other cores

---

**Non-OS Software**

- TCP/IP NETWORKING (NDK)
- TI - RTOS

**PLATFORM/EVM SOFTWARE**

- SECONDARY BOOTLOADER
- FATFS

**BOARD LIBRARY**

- DIAGNOSTICS
Bare metal code for configuring platforms and interacting with board specific components on EVM platforms.
The TI RTOS components include SYSBIOS kernel that provides all the OS services.

Framework Components which can be used to manage resources such as EDMA memory between algorithms or tasks.

OSAL, that acts as a bridge between the OS and the bare-metal code underneath.

Network stack (NDK) which provides network services and a transport layer.
The PRU-ICSS-SW Industrial adds applications that provide a starting point for developers. The industrial Communications, Drive, and Board libraries.

The PRU-ICSS-SW Industrial adds applications that provide a starting point for developers. The industrial Communications, Drive, and Board libraries.

**Example Applications**
- EtherCAT Slave
- Profinet RT/IRT slave
- Ethernet/IP
- Profibus Master
- Profibus Slave
- HSR/PRP

**Industrial Libraries**
- EtherCAT slave stack
- Profinet RT/IRT slave stack
- Ethernet/IP slave stack
- ProfiBus master stack
- ProfiBus slave stack
- HSR/PRP driver
- Industrial Board Lib

**SOFTWARE FRAMEWORK COMPONENTS**
- Inter-processor communication Framework Components
- OS ABSTRACTION LAYER (OSAL)
- NETWORK I/F Mgmt Unit (NIMU)

**ALGORITHM LIBRARIES**
- DSPLIB
- IMGILIB
- MATHLIB

**LOW LEVEL DRIVERS (LLD)**
- EDMA3
- ICSS-EMAC
- PCIe
- PRUSS
- 12C
- EMAC
- USB
- SPI
- GPIO
- UART
- SD/MMC

**CHIP SUPPORT LIBRARY (CSL)**

**PLATFORM/EVM SOFTWARE**
- SECONDARY BOOTLOADER
- FATFS
- BOARD LIBRARY
- DIAGNOSTICS

**RTOS-NETWORK**
- TCP/IP NETWORKING (NDK)

**Non-OS Software**
- XDC Tools

**OS Software**
- Processor SDK

**Device/platform-dependent**
- OS-dependent
- OS Independent
- N/A for AM3 & 4
PSDK-RTOS simplifies the porting process by consolidating all HW specific SW into the board library.

Modifying the board library is the major step required while porting the software to the custom platform.

Aside from the Board library - the SBL and diagnostics components may need to be modified.

Application level changes are limited as all of the high level drivers and application relies on this board library for hardware configuration.
Processor SDK : Additional Detail - CHIP SUPPORT LIBRARY (CSL)

- CSL constitutes a set of well-defined APIs that abstract low-level details of the SoC device.
- Users configure, control (start/stop, etc.) and have read/write access to peripherals avoiding register bit-field details.
- The CSL has two layers:
  - The first layer assigns a standard name to the Memory Mapped Registers. Uses macros and type definitions.
  - The second layer is a set of “C” functions to manipulate these registers.
- The AM335x and AM437x Processor SDKs contain Starterware for legacy support.
- Some AM335x and AM437x CSL draws from starterware
- For more information see - [http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_CSL](http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_CSL)
Low Level Drivers (LLD) hide the details of CSL from the application.

Applications are recommended to use the LLD APIs.

There are a few IP and peripherals that do not have LLDs. In these cases, the application uses the CSL Functional layer directly.

In a few cases, LLDs access the hardware directly (not via CSL).

Agenda

What are the changes from Industrial 2.x to PRU-ICSS-Industrial SW plus Processor SDK

Processor SDK Overview

Changes from SYSBIOSSDK-IND-SITARA to PRU-ICSS-Industrial and Processor SDK

Migrating an Application from ISDK to PRU-ICSS-Industrial Processor SDK

Examples
  - Moving a starterware application to use an LLD
  - Modifying Board Library to Change UART Instance on AM335x
Changes from SYSBIOSSDK-IND-SITARA to PRU-ICSS-Industrial and Processor SDK
Industrial Software Stack Changes

SYSBIOS Industrial SDK vs PRU-ICSS-Industrial & Processor SDK

Few Changes in Protocol and Application

Improved partitioning, generality & uniformity
There are a number of Starterware modifications that are needed to move to both the ISDK 2.0 and PRU-ICSS-Industrial and Processor SDK. These are described in the Starterware migration guide [http://processors.wiki.ti.com/images/5/55/Migration_guide.pdf](http://processors.wiki.ti.com/images/5/55/Migration_guide.pdf).

Examples, Tests and Diagnostics have been removed for clarity.
File Structure from Industrial SDK 2 to Processor SDK

For more information
File Structure going from Industrial SDK 1 to PRU-ICSS EtherCAT

Examples, Tests and Diagnostics have been removed for clarity
File Structure going from Industrial SDK 2 to PRU-ICSS EtherCAT

Industrial SDK

- ISDK 2.1.3.2
  - board
    - include
      - gpio, i2c, mcspi, mux, phy, platform, support, tlkphy
        - icev2AM335x
          - lcd, rotaryswitch, spiflash
        - idkAM437x
          - qspi
    - lib
    - source
  - examples
    - ethtool
  - os_drivers
    - include
      - edma, emac, epwm, mddio, ndkdeviceconfig
    - lib
    - source
  - protocols
    - ethtool
      - docs
    - ecat_appl
      - EcatStack
      - esi
      - patch
      - firmware
      - include
      - stack_lib
  - tools
    - bin2header

PRU-ICSS-EtherCAT

- docs
- examples
  - board
    - dp83822, misc, board_rotary_switch, tlk105
      - tlk110, tlkphy, delay_us, icss_emac_osal
    - mddio_drv, oled_drv, osdrv_ndkdeviceconfig
      - IceAM335x
        - gpioLed, i2cLed, mcspi, oled, phy, spi
        - idkAM437x
      - emac
  - ethercat_slave
    - esi
    - src
      - firmware
      - include
      - projects
      - pdk_patches
    - EtherCAT Protocol
      - docs
        - ecat_appl
      - EcatStack
      - esi
      - icss_emac_osal
      - mdio_drv
      - oled_drv
      - osdrv_ndkdeviceconfig
      - EtherCAT Protocol
      - tools
        - bin2header

Board

- Examples, Tests and Diagnostics have been removed for clarity

EtherCAT Protocol

- Examples, Tests and Diagnostics have been removed for clarity
When migrating from ISDK 1.x two sets of modifications are required:

1) There are a number of Starterware modifications that are needed to move to both the ISDK 2.0 and PRU-ICSS-Industrial and Processor SDK. These are described in the Starterware migration guide:


2) A few API’s change when moving from ISDK1.x moving to ISDK 2.x. These are described in:

http://processors.wiki.ti.com/index.php/SYSBIOS_Industrial_SDK_Migration_guide_from_1.1_to_2.1

Examples have been removed for clarity.
Agenda

What are the changes from Industrial 2.x to PRU-ICSS-Industrial SW plus Processor SDK

Processor SDK Overview

Changes from SYSBIOSSDK-IND-SITARA to PRU-ICSS-Industrial and Processor SDK

Migrating an Application from ISDK to PRU-ICSS-Industrial Processor SDK

Examples
- Moving a starterware application to use an LLD
- Modifying Board Library to Change UART Instance on AM335x
Migrating an Application from ISDK to PRU-ICSS-Industrial Processor SDK
Processor SDK RTOS – Porting to a custom board

TI Application on TI Evaluation Platform

- Tools (UIA)
- EDMA, Etc
- LLD
- IPC
- CSL
- Demo Application
- Network Dev Kit
- TI Board

TI Application on Customer Platform

- Tools (UIA)
- EDMA, Etc
- LLD
- IPC
- CSL
- Demo Application
- Network Dev Kit
- Custom Board

Platform Migration

- No modifications required
- May be used “as is” or customer can implement value-add modifications
- Needs to be modified or replaced with customer version
PSDK-RTOS simplifies the porting process by consolidating all HW specific SW into the board library.

Modifying the board library is the major step required while porting the software to the custom platform.

Aside from the Board library - the SBL and diagnostics components may need to be modified.

Application level changes are limited as all of the high level drivers and application relies on this board library for hardware configuration.
Moving a project from ISDK to PRU-ICSS-Industrial SW & Processor SDK

Preparation

• Identify the changes / additions that were made to:
  – ISDK starterware, board, and bootloader libraries
  – ISDK protocol and other industrial libraries (control & interface)
• Identify any BIOS configuration modifications.
• Identify the changes / additions that were made to the application
Moving a project from ISDK to PRU-ICSS-Industrial SW & Processor SDK

Migration

• Incorporate in the changes / additions that were made to the starterware, board library and bootloader into the Processor SDK

  – Starterware changes can be used directly or the modifications can be implemented using the LLDs.

  – The advantage of using the LLDs is that their added integration can reduce development effort – For example - Configuring the LLD and enabling the pinmux.

  – Additional information is in the LLD section of

    https://training.ti.com/sites/default/files/docs/Processor_SDK_RTOS_P2_Slides_0.pdf
Moving a project from ISDK to PRU-ICSS-Industrial SW & Processor SDK

Migration

- Incorporate the board library modifications into the board libraries of the SOC and the PRU-ICSS-Industrial

  - DRIVE:\ti\pdk_SOC_Vers\packages\ti\board\src\SOC_Board\n
    - Additional information is available in the Board Library section of https://training.ti.com/sites/default/files/docs/Processor_SDK_RTOS_P2_Slides_0.pdf

  - DRIVE:\ti\PRU-ICSS-PRTOTOCOL_Slave_VERSION\examples\board\SOC_Board\n
    - Note this is very similar to the ISDK
      Drive:\ti\sysbios_ind_sdk_VERSION\sdk\board\source\n
    - Incorporate the bootloader modifications
Moving a project from ISDK to PRU-ICSS-Industrial SW & Processor SDK

Migration

• Incorporate the changes / additions that were made to the ISDK protocol and other industrial libraries (board, control and interface)
  
  – The protocol sections in the PRU-ICSS-Industrial SW are very similar to the Industrial SDK
    
    – However, because PRU-ICSS-Industrial SW supports a broad range of devices - #ifdef SOC_NAME is used to support the SOCs unique characteristics in defines and assignments

• Incorporate any TI RTOS configuration modifications.

• Incorporate any application changes / additions
  
  • The application sections in the PRU-ICSS-Industrial SW are very similar to the Industrial SDK
Agenda

What are the changes from Industrial 2.x to PRU-ICSS-Industrial SW plus Processor SDK

Processor SDK Overview

Changes from SYSBIOSSDK-IND-SITARA to PRU-ICSS-Industrial and Processor SDK

Migrating an Application from ISDK to PRU-ICSS-Industrial Processor SDK

Examples
- Moving a starterware application to use an LLD
- Modifying Board Library to Change UART Instance on AM335x
Examples

Moving a starterware application to use an LLD
Modifying Board Library to Change UART Instance on AM335x
Example of moving a Starterware Application to use the Processor SDK LLD

If we desired to change the Console from UART3 to UART 1

- **ISDK Implementation**
  - Not all UART instances are supported in the Industrial SDK Starterware - it is necessary to
    - Add code to enable the uart instance in PRCMModuleEnable in
      `{IA_SDK_HOME}\starterware\soc\am335x\am335x_prcm.c` and
    - Change the BUILDCFG_MOD_UART to specify UART 1 in
      `{IA_SDK_HOME}\starterware\board\am335x\am335x_icev2.c`
    - Add code to configure the pin mux settings in ConsoleUtilsUartPinMuxSetup in
      `{IA_SDK_HOME}\starterware\utils\console_utils_uart.c`
    - We then rebuild the Starterware soc, board, utils and the sysbios board_support library.
Example of moving a Starterware Application to use the Processor SDK LLD

If we desired to change the Console from UART3 to UART 1

In the Processor SDK – there is a lager number of UARTs already supported in the library

- **Already done - PRCM domain:** In the board library - define the enable for the UART in
  DRIVE:\ti\pdk_am335x_1_0_x\packages\ti\board\src\<board_name>\board_name.c

- **Already done - PINMUX setting:**
  - Add Pinmux definition for UART in
    DRIVE:\ti\pdk_am335x_1_0_5\packages\ti\starterware\board\am335x\am335x_icev2_pinmux_data.c
    static pinmuxModuleCfg_t gUartPinCfg[]
  - Add the pinmux module configuration for UART in
    DRIVE:\ti\pdk_am335x_1_0_x\packages\ti\board\src\<board_name>\board_name_pinmux.c
    Board_STATUS Board_pinmuxConfig (void)

- In DRIVE:\ti\pdk_SOC_VER\packages\ti\board\src\SOC_BOARD\include\board.cfg.h
  change  #define BOARD_UART_INSTANCE  to  1

- In DRIVE:\ti\pdk_am335x_1_0_x\packages\ti\drv\uart\soc\am335x\UART_soc.c,
  edit uartInitCfg[CSL_UART_PER_CNT] interrupt number for the UART instance

- Rebuild: starterware library, board library and UART library
This document describes the procedure to modify the default UART0 example in the AM335x Processor SDK RTOS package to enable UART1.

On the BeagleBone Black (BBB) P9 header, pins 24(TX) and 26(RX) are connected to UART1.

This procedure shows a test to verify that UART1 is enabled on the BBB.

Appendix
AM355x AM437x Processor SDK & PRU-ICSS Industrial Software Stack

Example Applications
- EtherCAT Slave
- Profinet RT/IRT slave
- Ethernet/IP
- Profinet Master
- Profinet Slave
- Profibus Slave
- HSR/PRP
- ...

Industrial Libraries
- EtherCAT slave stack
- Profinet RT/IRT slave stack
- Ethernet/IP slave stack
- Profinet Master stack
- Profinet Slave stack
- HSR/PRP driver
- Industrial Board Lib

Software Framework Components
- Inter-processor communication
- Framework Components
- OS Abstraction Layer
- NIMU

Network
- NDK or 3rd Party Stack
- TI RTOS kernel

TI RTOS/ Low Level Drivers
- EDMA3
- ICSS EMAC
- PCIe
- PRUSS
- I2C
- EMAC
- USB
- McSPI/QSPI
- GPIO
- UART
- MMCSD
- ...

Platform/EVM Software
- Secondary bootloader
- FATFS
- Board Library
- Diagnostics

Chip Support Library

Hardware

For more Information: [http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_Software_Stack](http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_Software_Stack)
Why Change? to Processor SDK RTOS:
Maximize Software Reuse

- **TI Demo Application on TI Evaluation Platform**
  - Demo Application
  - Tools (UIA)
  - EDMA, Etc
  - Industrial Protocol & NDK
  - LLD
  - IPC
  - CSL

- **TI Demo Application on Customer Platform**
  - Demo Application
  - Tools (UIA)
  - EDMA, Etc
  - Industrial Protocol & NDK
  - LLD
  - IPC
  - CSL

- **Customer Application on Customer Platform**
  - Custom Application
  - Tools (UIA)
  - EDMA, Etc
  - Industrial Protocol & NDK
  - LLD
  - IPC
  - CSL

- **Custom App on Next Generation TI SOC Platform**
  - Custom Application
  - Tools (UIA)
  - EDMA, Etc
  - Industrial Protocol & NDK
  - LLD
  - IPC
  - Next Gen TI Platform

---

**Platform Migration**
- No modifications required
- May be used “as is” or customer can implement value-add modifications
- Needs to be modified or replaced with customer version

**Application Migration**

**Future Proof**

Software may be different, but API remains the same (CSL, LLD, etc.)

Texas Instruments