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

May 1, 2017
## Agenda

<table>
<thead>
<tr>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are the changes from Industrial SDK 1.x and 2.x to PRU-ICSS-Industrial SW plus Processor SDK</td>
</tr>
<tr>
<td>Migrating an application from ISDK to PRU-ICSS-Industrial Processor SDK</td>
</tr>
<tr>
<td>Example of moving a starterware application to use an LLD</td>
</tr>
</tbody>
</table>
At a high level – What is changing

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

- Easier migration to custom application platforms because platform 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.
  - Refer: Processor SDK RTOS Migration Guide history
Each Industrial communications protocol will be contained in a separate package.

**Industrial SDK**
(AM335x, AM437x)

- Industrial Application
- API
- Peripheral Drivers
- Protocol Stack
- Protocol Driver
- Protocol Firmware
- TI-RTOS
- Bootloader, Tools

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

- Protocol Stack
- API
- Protocol Driver
- Protocol Firmware

**Processor SDK**

- TI-RTOS
- Bootloader
- Tools
- Peripheral Drivers

**Third Party or Customer**
- TI
Each Industrial communications protocol will be contained in a separate package

- Instead of all of the protocols and the common libraries being delivered in a single Industrial SDK
  - Each protocol is an individual PRU-ICSS-Industrial-SW package
  - Each PRU-ICSS-INDUSTRIAL-SW package runs on top of the Processor SDK
  - Available at [www.ti.com/tool/PRU-ICSS-INDUSTRIAL-SW](http://www.ti.com/tool/PRU-ICSS-INDUSTRIAL-SW)

**SYSBIOSSDK-IND-SITARA**: SYSBIOS Industrial Software Development Kit (SDK) for AM335x and AM437x will continued to be delivered until all industrial protocols for the AM335x and AM437x are migrated to PRU-ICSS-Industrial-SW
  - TI will no longer provide SYSBIOSSDK-IND-AM335x after 31 March 2017, and will no longer provide SYSBIOSSDK-IND-SITARA after 30 June 2017
Why Change? - Processor SDK RTOS: Maximize Software Reuse

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

Software may be different, but API remains the same (CSL, LLD, etc.)
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.
The Processor SDK hardware interface uses the Chip Support Library and Low Level Drivers in place of Starterware.

- Most peripheral and SOC specific configuration is done using the LLDs.
- The LLDs are built on top of the CSL layer and provide an interface to operate and control the peripherals on the device.
- The CSL contains low level register level definitions and basic functionality to configure cores and registers on the device.
  - 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)
  - In the case of the AM335x and AM437x some of the CSL draws from starterware

- However - Starterware is still provided in the AM335x and AM437x Processor SDKs for legacy support.
Changes from Industrial SYSBIOSSDK-IND-AM335x to SYSBIOSSDK-IND-SITARA (SYSBIOS Industrial 1.1 to 2.1)
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
Changes from SYSBIOSSDK-IND-SITARA to PRU-ICSS-Industrial and Processor SDK
Overview of Industrial Software Stack

SYSBIOS Industrial SDK

Few Changes in Protocol and Application

Improved partitioning, generality & uniformity

PRU-ICSS-Industrial & Processor SDK
File Structure from Industrial SDK to Processor SDK

For more information

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

Examples, Tests and Diagnostics have been removed for clarity
Processor SDK RTOS – Porting to a custom board

- May be used "as is" or customer can implement value-add modifications
- Needs to be modified or replaced with customer version

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
Processor SDK RTOS: Functional View

Components that will definitely need modification
Components that may need modification
Overview

What are the changes from Industrial SDK 1.x and 2.x to PRU-ICSS-Industrial SW plus Processor SDK

Moving an application from ISDK to PRU-ICSS-Industrial Processor SDK

Example of moving a starterware application to use an LLD
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 the modifications can be small – 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\`
    
    - 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](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\`
    
    - Note this is very similar to the ISDK
      
      Drive:`\ti\sysbios_ind_sdk_VERSION\sdk\board\source\`
    
    - 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
Overview

What are the changes from Industrial SDK 1.x and 2.x to PRU-ICSS-Industrial SW plus Processor SDK

Moving an application from ISDK to PRU-ICSS-Industrial Processor SDK

Example of moving a starterware application to use an LLD
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 BUILD_CFG_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 larger 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
Appendix
Previous AM355x AM437x Processor SDK & PRU-ICSS Industrial Software Stack

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
- Profinibus master stack
- Profinibus slave stack
- HSR/PRP driver

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)