Industrial Software Development Kit

Overview - AM335x and AM437x 2.1 Industrial Software Development Kit
Oct 2, 2015
SDK Software

- The SYSBIOS Industrial SDK combines all the software components and tools needed to begin development of SYS/BIOS-based applications on the ARM for the:
  - AM437x Industrial Development Kit (IDK)
  - AM335x Industrial Communications Engine (ICE)v2.

- The SDK includes the following:
  - Open source SYS/BIOS Real Time Operating System (RTOS)
    - Library of peripheral drivers integrated with SYSBIOS
  - Starterware with the device abstraction layer, device, and board APIs
    - Bootloader for AM437x IDK and AM335x ICEv2
  - Sample applications demonstrating peripheral use cases
  - Evaluation industrial communications protocol stacks for EtherCAT, Ethernet/IP, Profinet, ProfiBus, and SNMP
  - Sample industrial I/O applications over communication protocols such as EtherCAT
  - Motor and Drive feedback capability with interfaces for EnDat, Sigma Delta Decimation Filter, and ADC

- The SDK runs under the Code Composer Studio integrated development environment (IDE) v.6.1/
Industrial SDK (ISDK) Architecture
Industrial SDK – Typical Usage Model

Customer Software

- Industrial Application for end product (customer value add)
- BSP for customer boards/RTOS

Third Party Software

- Free Evaluation version of protocol stack
- For production, customers license stack from 3P, or develop/use own stack

TI Software

- Protocol firmware, protocol and peripheral drivers, example slave application, SYSBIOS bootloader, tools
- Some customers use these as is for production, some use as reference for their port
SDK Industrial Communications Software

- **EtherCAT Slave Controller**
  - Beckhoff Slave Stack Version 5.11
  - EtherCAT Commands
  - 8 FMMU support in Firmware
  - 8 SM support in Firmware (stack support only 4)
  - 8KB process data RAM
  - Distributed clocks
    - 64-bit DC
    - SYNC0 out generation single shot and cyclic mode support
    - SYNC1 out generation - SYNC1 cycle time multiple of SYNC0 cycle time
    - Latch0 and Latch1 inputs
  - DL Loop Control
    - Using PRU-ICSS MDIO state m/c (slow)
    - Using MII_RX_LINK (fast - depending on PHY link loss detection latency)
    - Enhanced link detection in EtherCAT firmware
  - EEPROM Emulation
    - QSPI flash non-volatile storage support
    - Provisions to use volatile storage for debug and development
  - Interrupts – AL/ECAT events
  - Watchdog – PDI and SM
  - Error Counters – all key counters
  - LED – Run, Error and Port0/1 activity based on firmware feedback
  - CoE, EoE, FoE and CiA402
  - Management Interface for PHY over EtherCAT
  - PHY address configuration and host side PRU-ICSS MDIO API for PHY programming
  - Redundancy support validated using PG2.x silicon
  - Application upgrade over FoE complying to SEMI TWG guidelines (not resetting PRU-ICSS)
SDK Industrial Communications Software

- **EtherNet/IP adapter**
  - EtherNet/IP adapter implementation on low latency ICSS cut-thru switch
  - Beacon based DLR ring node capability with ICSS based implementation
    - Flush Table frame support has been added (New in this release)
  - Integrated with Molex Ethernet/IP adapter stack Version 3.3.x supporting the following features
    - Ethernet/IP Capabilities
      - Explicit Messaging and Implicit Messaging
      - UCMM
      - Class 1 and Class 3 Connection
    - Supported CIP Objects
      - Identity
      - Message Router
      - Assembly
      - Connection Manager
      - Ethernet Link
      - TCP/IP Interface
      - DLR (on ICSS implementation)
      - QoS
      - Interface and Media counters in Ethernet Link Object (New in this release)
  - Supports Exclusive Owner, Input Only and listen Only Connections
  - Supports Address Conflict Detection
  - Compliant with ODVA EtherNet/IP Protocol Conformance Test Tool (Rev CT-12)
  - IEEE1588 End to End Transparent Clock support in firmware (New in this release)
SDK Industrial Communications Software

• **Profinet RT/IRT device**
  – Profinet I/O RT/IRT Device(slave) conforms to PROFINET Conformance classes A,B and C functionality.
  – Integrated with Molex PROFINET stack
  – PROFINET I/O RT/IRT Device Features:
    • Supports minimum cycle time of 250 us
    • Integrated two-port cut-through switch, 100 Mb/s Full Duplex
    • Profinet Quality of Service(QoS)
    • Upto 8 Application Relations (ARs)
    • Data Hold Timer
    • DCP Identify Filter
    • PTCP
    • 1 ms buffering per port
    • 8 IOCR with 40..1440 Bytes
    • 802.1d learning bridge for received source MAC addresses
    • PNIO static routing and custom FDB for multicast addresses
    • Interrupt Pacing
  – Compliant with Profinet IO Tester Version 2.3.5.25.4430
  – Compliant with Profinet IRT SPIRTA Module Version 2.3.5.25.402
SDK Field Bus and Ethernet Software

- **PROFIBUS DP Slave Controller**
  - Evaluation PROFIBUS DP slave stack library file from TMG
  - Baud rates 12M, 6M, 3M, 1.5M, 500k, 187.5k, 93.75k, 45.45k 19.2k, 9.6k
  - Telegram size Up to 244 bytes
  - DPv0 support Cyclic exchange of data and diagnosis
  - DPv1 support Acyclic/cyclic data exchange and alarm handling
  - Response time 11-bit minimum TSDR response time
  - Watchdog 10ms time base
  - Get diagnostics and extended diagnostics

- **Ethernet MAC**
  - Ethernet MAC via PRU-ICSS
  - Independent EMAC instance per PRU
SDK Motor Control Software

• **EnDat Master**
  – EnDat 2.2 & 2.1 command set
  – Interrupted and continuous clock mode
  – Clock configuration up to 16MHz
  – Cable length up to 100m @8MHz
  – Propagation delay compensation
  – Automatic estimation of propagation delay
  – Receive on-the-fly CRC verification of position, parameters and additional information
  – Two modes of operation - host trigger and periodic trigger

• **Motor Control**
  – PMSM Field Oriented Control (FOC)
  – EnDat position feedback
  – Current sense via Sigma Delta Decimation Filter or On-chip ADC
  – Closed Current control
  – Closed Speed control
  – Closed Position control
  – Current loop (PWM period) verified up to 33.3KHz
  – Speed/Position loop at one-tenth of current loop

• **Sigma Delta Decimation Filtering (SDDF)**
  – PRU Firmware source examples of SINC2 and SINC3 filter
  – SINC3 filter for Normal current and voltage measurement
  – SINC2/SINC3 filter for overcurrent detection
  – Two modes of operation - trigger based sampling and continuous sampling
  – Signaling via DIGIO upon detecting overcurrent
SDK Software Components

• **SDK Software**
  – [Code Composer Studio](#) version CCS 6.1.0.00104
  – [Industrial SDK](#) version 2.1.0.1
  – Compiler GNU v4.8.4 (Linaro)

• **Required Additions**
  – SYS/BIOS 6.41.4.54 Real Time Operating System
  – XDC Tool 3.31.2.38_core
  – NDK 2.24.2.31
  – Serial console terminal application (like Teraterm, minicom, HyperTerminal)

  – IA_SDK_HOME
    • System Variable to location where the SDK is installed. Typically C:\ti\sysbios_ind_sdk_2.1.0.1\sdk
    • This can also be defined ether as a project or a CCS user defined variable.
SDK Directories

sdk
  board
    docs
    include
    lib
    source

control
  foc
  math_blocks

docs

interfaces
  endat_master
  pru_onchip_adc_sampling
  sddf

os_drivers
  docs
  include
  lib
  lld
    emac
    pruss
    source
    ICSS

- **board** - This folder contains source code board specific interface implementations of digital I/Os, Flash and API documentation. *Draws upon Starterware definitions and APIs.*

- **control** - This folder contains source code required for Motor control (FOC and IQMATH)

- **docs** - This folder contains the Release Notes, User Guide and Getting Started guide

- **interfaces** - This folder contains EnDat Master, ADC sampling and Sigma Delta Decimation Filter motor control support functions interface source, header and firmware

- **os_drivers** - This folder contains source code and include files for SYS/BIOS drivers for AM335 and AM437x with pre-built libraries and API documentation. *Draws upon Starterware definitions and APIs.*
SDK Directories

**protocols**
- ethercat_slave
- ethernetip_adapter
- ethernet_mac
- profibus_slave
- profinet_slave
- ptp
- snmp

**starterware**

**tools**
- bin2header
- flasher
- gel
- isdk_image
- post_build

**examples** (Utilizes Starterware definitions and APIs)
- endat_diagnostic
- ethercat_slave
- ethernetip_adapter
- ethernet_mac
- example_utils
- motor_control
- profibus_slave
- profinet_slave
- sddf_gui_client

- **protocols** - This folder contains supported industrial automation protocol libraries and/or source codes and their project files.

- **starterware** - This contains StarterWare code which provides Device Abstraction Layer libraries and peripheral/board level sample/demo examples.

- **tools** - This contains tools supporting application development SPI flashing and GEL files.

- **examples** - This folder contains examples, their project files, source and include files. Users can build these applications in CCS. *Draws upon Starterware definitions and APIs.*
Industrial SDK Drivers

The Industrial SDK provides driver libraries for accessing device peripheral modules from a SYS/BIOS application. Some of the primary industrial drivers are:

**Board**
- The Industrial SDK provides a driver library for accessing device peripheral modules from an application. This has implementations for Digital Inputs, Digital Outputs, Flash memory (QSPI and SPI), Rotary Switch, TLK100 Ethernet Phy, integrated matrix monochrome display. There are different build configurations present for AM3x and AM4x.
- Additional information available at {IA_SDK_HOME}\board\docs

**PRU-ICSS NDK NIMU**
- NDK Driver interfaces the Network Developers Kit (NDK) with underlying architecture. Network Development Kit (NDK) serves as a rapid prototype platform for the development of network and packet-processing applications.
- IPv6 and IPv4 compliant TCP/IP stack
- Layer 3 & 4 network protocols
- High-level network applications including HTTP server and DHCP
- This driver interfaces with both Industrial Communication SubSystem (ICSS) and Common Platform Ethernet Switch (CPSW).
Industrial SDK Low Level Drivers

PRU ICSS LLD
- PRU ICSS Low Level Driver enables application/drivers to interface with the PRU Subsystem for the AM335x and AM437x.
- This module contains APIs which can be used to initialize PRU, load firmware and configure the PRU-ICSS features.
- Additional information available at {IA_SDK_HOME}\os_drivers\docs

ICSS EMAC LLD
- ICSS EMAC LLD (ICSS EMAC Driver) is an ethernet driver implementation using the PRU-ICSS.
- Contains APIs which enable communication between any High Level Driver/Application and EthernetIP, Profinet and ethernet mac firmware.
- ICSS EMAC Driver supports both EMAC mode and Switch mode depending on the firmware loaded to PRU-ICSS.
- For more information please refer {IA_SDK_HOME}\os_drivers\docs.
What is Starterware

- C based NO-OS code
- Can be used stand-alone
- Peripheral programming interface - Easy to use SW interface
- Software portability across devices for a given peripheral
- Stand-alone examples
- Not a middleware complete solution
- Product is designed to scale for each SoC. Single package is used to support more than one SoC. The product packaging/folder takes care of scalability for new IPs of new SoCs.
StarterWare Block Diagram

- StarterWare provides a no-OS platform support for ARM/DSP based SoCs.
- StarterWare provides Device Abstraction Layer (DAL) libraries and peripheral/board level samples and demo examples.
SDK Starterware Directories

**starterware**

**board**

- **Board module** (boards + devices): Abstracts out board specific information through APIs / header files.
  - Auto-detection of board type and definition
  - Generic pinmux configuration APIs
  - Defines board specific data
    - Pinmux, clock source, memories, power
    - On-board devices list, IP, mux, IO expanders

**dal**

- **DAL module** (Device Abstraction Layer) contains the APIs to configure and access the device peripherals

**device**

- **Device module** contains documentation & source for the onboard devices.

**soc**

- **SOC module**: Abstracts out SoC specific information through APIs / header files.
  - Power, Reset, and Clock Management APIs
  - Interrupt, Cache, MMU APIs
  - Device Specific Data – ID, revision, IPs contained, Base addresses
SDK Starterware Directories

**tools**
- ais_gen
- binToC
- bmp2c
- bmpToRaster
- enet_client
- ewarm
- flash_writer
- gel
- makefsfile
- out2rpnc
- pinmux_config
- pnmtoc
- pru_image
- qspi
- ti_image
- usb_inf

- **Tools module:**
  - Contains various SW tools like flash tools
  - E.g. Nor flash Writer is used to write data or erase the NOR flash

**utils**

- **Utils module:**
  - UART Console
  - I2C utilities
SDK Starterware Directories

**Library**
- grlib
- mmcsd
- nandlib
- norlib
- qspilib
- third_party
  - fatfs
  - ff9b
  - lwip-1.3.2
  - lwip-1.4.0
  - lwip-1.4.1
  - neonMathLib
  - xmodem
- usblib
- usb_lib

**Library module:**
- Provides higher level API for certain complex peripherals
- Uses driver abstraction layer
- Provide buffer management and manage state of the device

**Bootloader**
- Boot loader performs:
  - Initializes the board
  - Configure the peripheral I/O
  - Initialize DDR
  - Load multicore RPRC image and brings the slave cores out of reset
- Boot Modes: QSPI and MMCSD
SDK Starterware Directories

**Examples**

- adc
- cache_mmu
- cap_tsc
- dcan
- ddr
- dmtimer
- dss
- enet
- epwm
- example_utils
- gcc
- gpio
- grlib_demo
- lcdc
- mcspi
- mmcsd
- nand
- pm
- qspi
- rtc
- test
- uart
- usb
- vpfe
- wdt

**Examples module**
- Examples organized based on IP / use-case rather than EVM
- Common across SoCs / EVMs
  - Use SoC & Board data to keep the example code generic
  - Enables user to get examples working on custom board by defining their board data.
- Example Utils: Extract out common code needed by multiple examples
  - IP example focuses on the details of IP specific programming rather than subsidiary modules needed to create a use-case.
  - Utils include
    - MMU configuration
SDK Pre-Built Libraries

• The SDK installation comes with following pre-built libraries. These libraries can be used to build user applications quickly.

• Communications
  – libecat_slave_stack_gcc.a - EtherCAT Slave Stack library including PRU Firmware.
  – libethernetip_stack.a - EtherNet/IP Slave stack library
  – libprofibus.a - Profibus Slave stack library
  – libprofinet_slave_stack_gcc.a - Profinet Slave stack library
  – libsnmp_core_gcc.a - SNMP core stack library

• SysBios
  – libsys_bios_driver.a - SYS/BIOS specific implementation for accessing peripheral devices and EVM functionality

• Starterware
  – libboard.a - StarterWare library for board details
  – libdal.a - Starterware DAL specific initialization library
  – libdevice.a - StarterWare library for peripheral devices
  – libsoc.a - StarterWare SOC library
  – libutils.a - Starterware Utils library
Pre-Built Example Libraries

The pre-built binaries of the SDK examples can be downloaded separately. These are binary files that can be loaded on SD card or QSPI/SPI flash. To boot from SD card you will need to use the boot loader. All of the pre-built binaries with the exception of the bootloader are board independent.

- **endat_diagnostic** - Sample application demonstrating EnDat master functionality.
- **ethernet_mac** - Sample application demonstrating Ethernet MAC functionality over PRU-ICSS. The example uses two PRU's to implement two independent MAC's with two different MAC addresses and two different IP addresses.
- **example_utils** - Sample application demonstrating different peripherals on the board.
- **motor_control** - Sample application demonstrating motor control.
- **ethercat_slave** - Sample application demonstrating EtherCAT slave functionality.
- **profibus_slave** - Sample application demonstrating Profibus slave functionality.
- **profinet_slave** - Sample application demonstrating Profinet slave functionality.
- **sddf_gui_client** - Sample application demonstrating Sigma Delta functionality.
SDK Reference Documents

- Industrial SDK Getting Started Guide
- Industrial SDK 2.1.0.1 Release notes
- SYS/BIOS Getting Started Guide
- SYS/BIOS User Guide
- API Guides
  - EtherCAT Firmware API Guide
  - {IA_SDK_HOME}/board/docs
  - {IA_SDK_HOME}/interfaces/endat_master/doc
  - {IA_SDK_HOME}/interfaces/sddf/doc
  - {IA_SDK_HOME}/os_drivers/docs
  - {IA_SDK_HOME}/protocols/ethernetip_adapter/docs
  - {IA_SDK_HOME}/protocols/profibus_slave/doc
  - {IA_SDK_HOME}/protocols/profinet_slave/Docs

- Migration Guide from 1.1 to 2.1