Getting Started with PDK (Platform Development Kit) in J7

12th March 2019

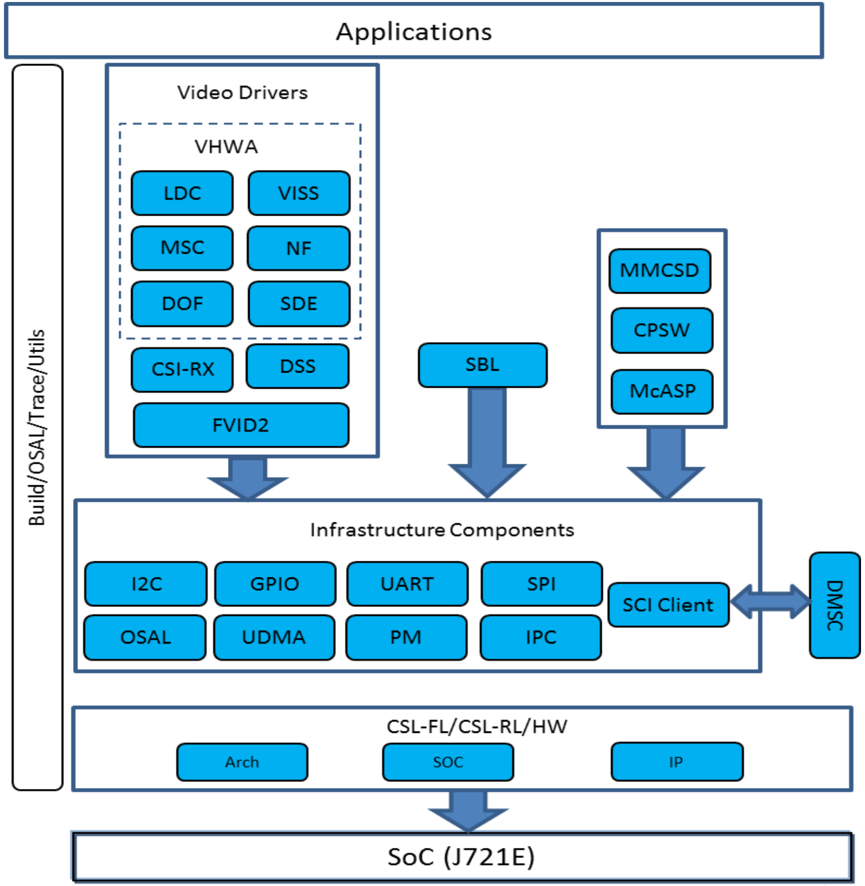
Agenda

- PDK Overview
- Directory Structure
- Prerequisite & Build System
- Software Modules
- Documentation
- Q&A

PDK Overview

- Supported SoC: J721E
- Supported OS: TI-RTOS, Baremetal
- Supported CPUs: R5, A72, C6x, C7x
- Misra-C compliant
- C based OS agnostic code
- Provides stand-alone examples and UT’s
- Makefile based build system

PDK Software Architecture



PDK Overview – Not Supported

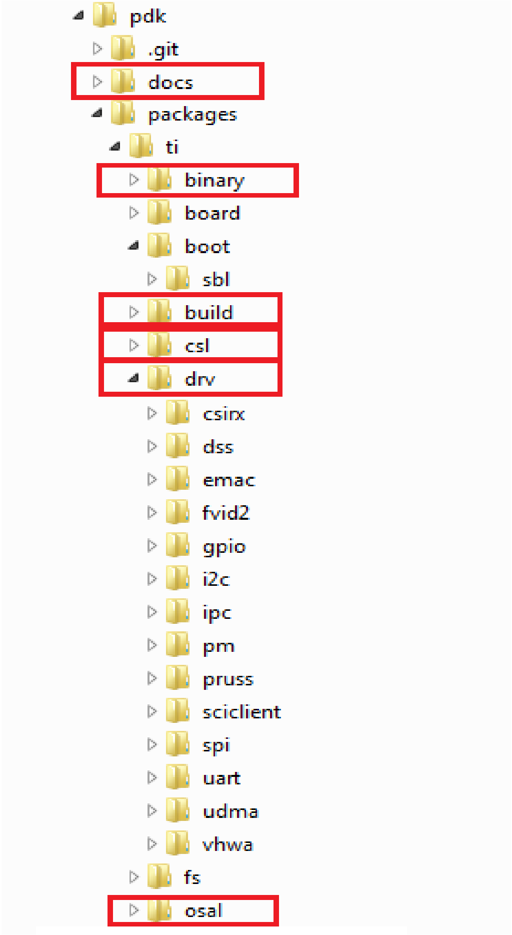
- Full featured software for peripherals (async, state machine, queuing)
- Depend upon any OS services (e.g. thread, locks, memory management)
- Peripheral “IP version specific” software
- Code with dependency (like algorithms, OS) for examples, demos
- PDK is not a middleware complete solution

Agenda

- PDK Overview
- Directory Structure
- Prerequisite & Build System
- Software Modules
- Documentation
- Q&A

Directory Structure

- Examples and sample applications are part of corresponding module
- Docs – Contains api guide, release notes and test reports
- Binary – Contains generated application binary
- Build – Contains makefile build system
- CSL – Baremental API’s for multiple IP’s/Cores
- OSAL – Interface layer for OS and NonOS
- Drv – Drivers for different modules



```
pdk
  .git
  docs
  packages
    ti
      binary
      board
      boot
        sbl
      build
      csl
      drv
        csirx
        dss
        emac
        fvid2
        gpio
        i2c
        ipc
        pm
        pruss
        sciclient
        spi
        uart
        udma
        vhwa
      fs
      osal
```

Agenda

- PDK Overview
- Directory Structure
- Prerequisite & Build System
- Software Modules
- Documentation
- Q&A

Pre-requisite & Build System

- Building and running PDK examples requires
  - Code Composer Studio (v 8.0. or above)
  - Compiler and Tool chains
  - Emulator – XDS560v2
- Makerules for different cores are specified in “<pdk>/packages/ti/build/makerules”
- Environment variables are specified in “<pdk>/packages/ti/build/makerules/env.mk”
- All build command’s the should be issued from “<pdk>/packages/ti/build” directory

Pre-requisite & Build System

Module/Example	Command	Description
All cores	gmake -j allcores BOARD=j721e	Command to build all example and libraries for J721E soc
All examples/Library	gmake -j all BOARD=j721e CORE=<core>	Command to build all examples and libraries for a given core of J721E
Example/Library	gmake <module> BOARD=j721e CORE=<core>	Command to build an examples and libraries for a given core of J721E
Help	gmake help	Help Menu

<core> *can be mcu1_0, mcu1_1, mcu2_0, mcu2_1, mcu3_0, mcu3_1, mpu1_0, c7x, c66xdsp_1,   c66xdsp_2*

Agenda

- PDK Overview
- Directory Structure
- Prerequisite & Build System
- Software Modules
- Documentation
- Q&A

Software Modules

Module	Description	Path
CSL	Chip Support Library provides functional layer API’s to configure and access peripherals	<pdk>/packages/ti/csl
SBL	Secondary boot loader initializes the execution environment and load’s multicore image	<pdk>/packages/ti/boot/sbl
CSIRX	CSI2RX library provides FVID2 API’s to configure the and capture sensor data stream using CSI2 protocol	<pdk>/packages/ti/drv/csirx
DSS	Display Sub System library provides set of FVID2 API’s to configure and control DSS engine to output video frames stored in the memory to the external display interfaces	<pdk>/packages/ti/drv/dss
VHWA	Vision Hardware Accelerator library provides FVID2 API’s to configure and used hardware accelerators such as NF, LDC, MSC, VISS, DOF and SDE	<pdk>/packages/ti/drv/vhwa

Software Modules

Module	Description	Path
PM	Power Management Module provides API to configure Power domain and Clock domain.	<pdk>/packages/ti/drv/pm
UDMA	Unified DMA is a versatile system DMA which provides multiple TX Channels and multiple RX Channels to transfer data using either TR based or Packet oriented control structures	<pdk>/packages/ti/drv/udma
SPI	SPI driver provides API’s to configure and use McSPI and Octal SPI	<pdk>/packages/ti/drv/spi
IPC	IPC used mailbox to provide two way communication between multiple host cores	<pdk>/packages/ti/drv/ipc
I2C	I2C library provides interface to configure and use multiple instance I2C	<pdk>/packages/ti/drv/i2c

Software Modules

Module	Description	Path
SCI Client	An Interface to the TI-SCI protocol for RTOS and non-OS based applications to exposes the core message details, valid module/clock IDs to the higher level software and abstracts the communication with the firmware based on the TI-SCI protocol	<pdk>/packages/ti/drv/sciclient
EMAC	Ethernet Media Access Control provides half- or full-duplex operation to send or receive data over the Ethernet.	<pdk>/packages/ti/drv/emac
UART	UART library provides interface to read and write data on UART port	<pdk>/packages/ti/drv/uart
OSAL	OSAL provides an simplified interface layer to use access RTOS feature	<pdk>/packages/ti/osal
Board	Provides API to initialize the board environment including clock, PLL, DDR and uart	<pdk>/packages/ti/board

Software Modules – CSL Module

- Chip Support Library contains
  - Logical group of register reads and writes to get a functional layer for a given IP
  - Stateless functions; blocking and runs to completion
  - System Configuration Code:
    - Provides functions specific to SoC level System Configuration.
    - Provides the following functionalities:
      - Start-up Code
      - Interrupt-vector initialization
      - Low level CPU specific code
    - May involve assembly code and hence can be tools specific
  - Sub module to further abstract and group the functionality.

Software Modules – CSL Module

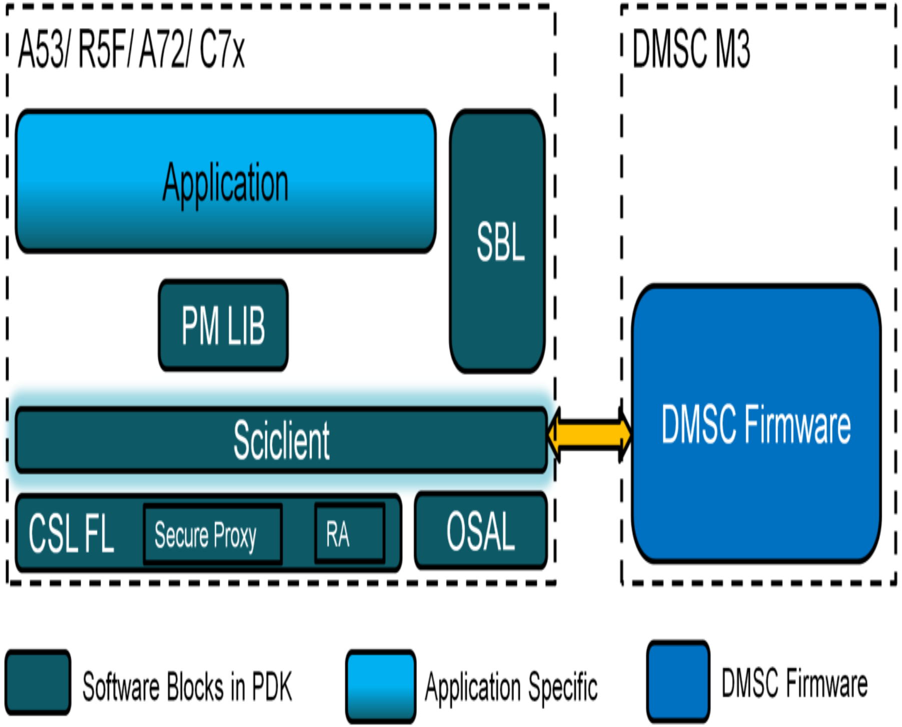
Sub Modules	Description
Arch	This sub module contains the core specific functionality and initialization code for the system and cores. The interrupt controller API's to configure and enable/disable the interrupt is also part of this sub module. Source code is available under 'csl/arch' folder.
SoC	SoC specific defines are abstraction under this sub module. SoC specific base address, macros and cross bar configuration code is available under this sub module. Source code is available under 'csl/soc' folder.
IP Specific Code	API to use the functionality of different IP's is available as part of this sub module. Source code is available under 'csl/src/ip' folder.

Software Modules – SBL

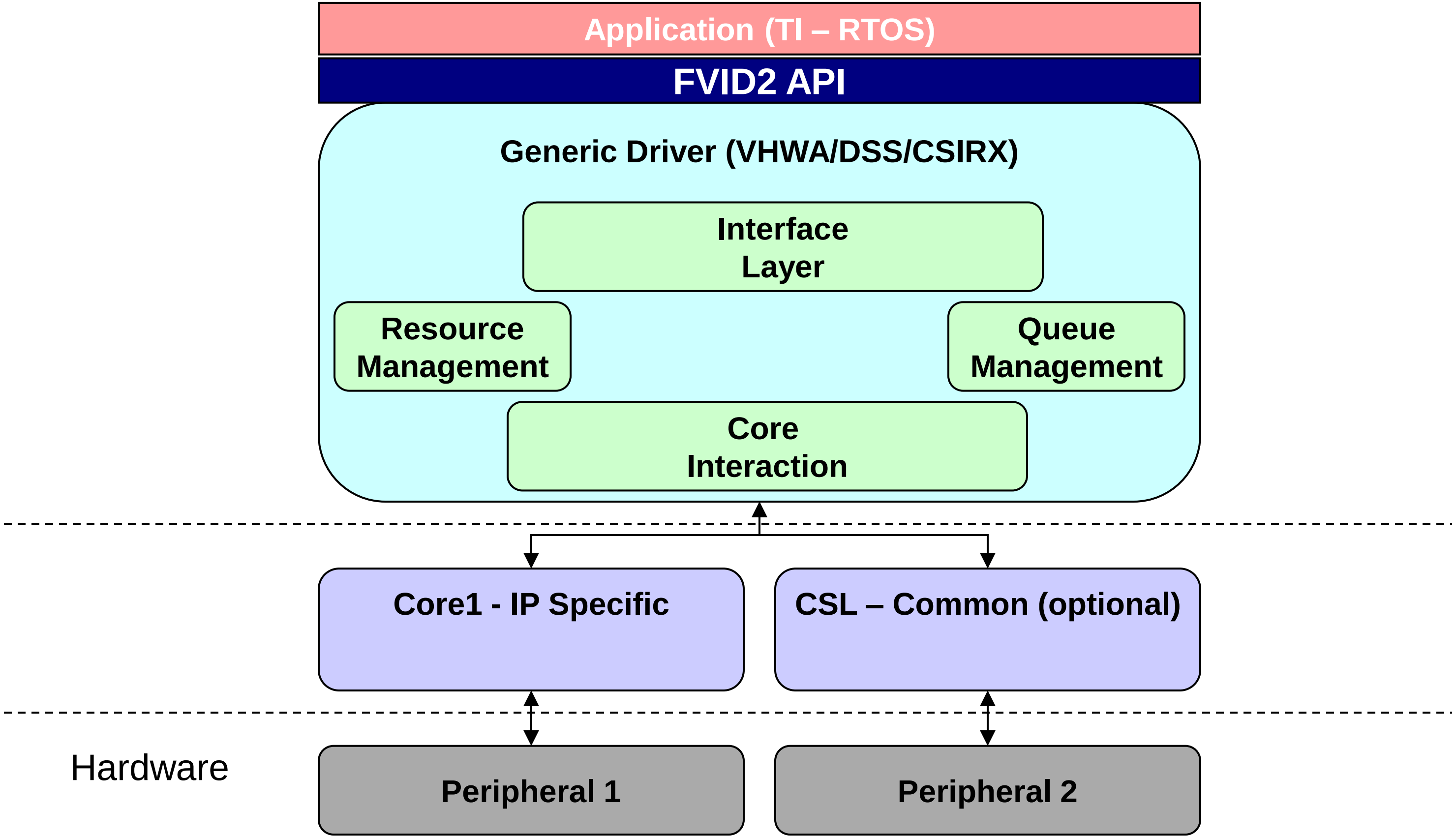
- SBL
  - Initializes the execution environment for multi-core application
  - Configure the Voltage rails
  - Initialize the Interrupt Controller
  - Setup the ADPLL values
  - Power on I/O peripherals
  - Power on slave cores
  - Configure required PADs
  - Initialize DDR
  - Load the multicore application image into DDR
  - Bring slave cores out of Reset
  - Boot Modes: MMCSD, OSPI and Uniflash (UART)

Software Modules – PM

- PM Module
  - Provides API to turn on/off the modules in the device
  - Set clock rate for different module
  - Supports Dynamic Power Management for cores
  - Provides API to detects device temperature and alerts on over temp
  - Interacts with System firmware via SCI client for Power management

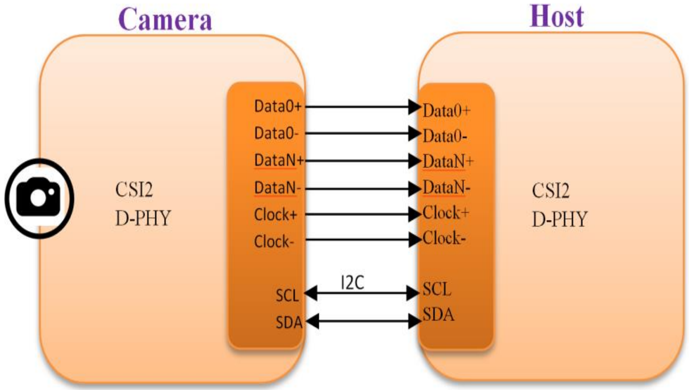


Generic FVID2 Driver Stack



Software Modules – CSI2RX

- Uses FVID2 API interface
- Up to 32 capture channels
- Supported Data formats
  - RAW8/10/12/14/16/20
  - YUV422-10 bit
  - RGB888
- OTF mode and loop-back mode to support re-transmission through CSI2TX
- 2.5 Gbps per lane and 4 independent lane in total
- FIFO handling
- Error Handling and Recovery
- D-PHY configuration
- Frame drop buffer programming



Software Modules – VHWA

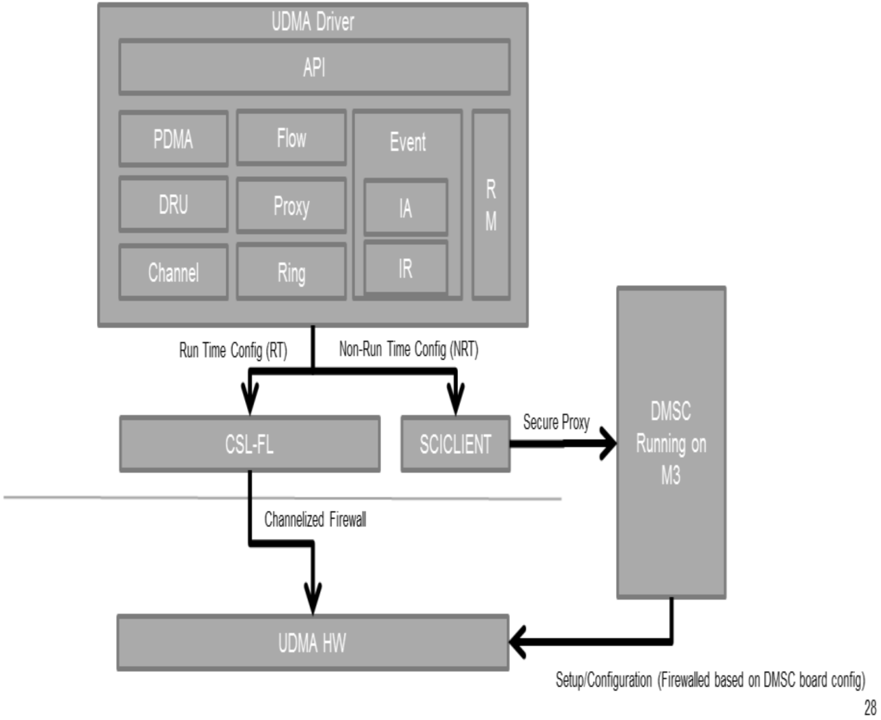
- Uses FVID2 API interface
- Consists of
  - Lens Distortion Correction (LDC)
  - Multi-Scalar (MSC)
  - Vision Imaging Sub System(VISS)
  - Noise Filter (NF)
  - Dense Optical Flow (DOF)
  - Stereo Disparity Engine (SDE)
- Independent drivers for supported IP's
- Supports 8bit Packed, 12bit Packed and 12 bit unpacked format
- Wide input/ouput resolution
  - Up to 2K for DOF and SDE
  - Up to 8K for LDC, NF, MSC and VISS

Software Modules – DSS

- Single instance of display controller driver and one instance of display driver for each of the forward pipes
- Supports both 8/10-bit YUV222 and 24/30/36-bit RGB input data formats
- Different output interfaces: DPI, eDP, DSI and OLDI
- Supports Scaling, Color Space Conversion, Blending, Color Keying
- Supports display sharing with other cores/OS
- Support up to four simultaneous 4K display outputs
- CRC generation/comparison and freeze frame detection
- Hardware based connection check

Software Modules – UDMA

- TR based and Packet oriented control structure
- Multiple TX and RX channels
- M:N mux to generate interrupt on various cores
- Event based trigger mechanism
- Up to 64K system events
- Supports polling and interrupt mode



Software Modules – I2C

- Compliant with Philips I2C specification version 2.1
- Supports a standard mode (up to 100 kbps) and fast mode (up to 400 kbps)
- Supports HS mode for transfer up to 3.4 Mbps
- 7-bit and 10-bit device addressing modes
- Supports both master and slave mode
- Built-in configurable FIFOs
- Programmable multi-slave channel

Agenda

- PDK Overview
- Directory Structure
- Prerequisite & Build System
- Software Components
- Documentation
- Q&A

Documentation

Document	Directory/Path	Description
API Guide	<pdk>/docs/pdk_introduction.html	Contains the API description for all drivers/libraries of PDK
User Guide	<pdk>/docs/pdk_introduction.html	Detailed used guide with steps to build and run each module
Release Notes	<pdk>/docs/pdk_introduction.html	Release note for PDK release
Test Report	<pdk>/docs/pdk_introduction.html	Test report for the PDK release
Traceability Matrix	<pdk>/docs/pdk_introduction.html	Requirement – Design –Code traceability matrix

Q&A