

PDK/PDK FAQ

[Return to PDK Software Developer Guide](#)

Contents

Generic

- What does PDK stand for?
- What do I get with PDK, and how do I get started?
- How do I maintain my changes over PDK? How to pick up the latest bug fixes from a newer PDK release?

Board Porting

- How to change UART instance in VSDK and PDK?

Example Porting

- How to change IP instance for an example in PDK?

Build

- Can I build PDK on a Linux machine?
- How do I build PDK library for a particular Platform?
- Can I selective build VIP driver only without DSS and VPE drivers?
- How can I build PDK examples while using Processor SDK Vision (AKA Vision SDK)?
- What are the common issues faced during the build?
 - Windows Permissions
 - Compiler Versions
 - Compiler Paths
 - Building with -j

CPU Architecture

- Where is the source code for different CPUs supported by PDK?
- What are the different levels of cache supported by M4?
- What are the different levels of MMU supported by M4?
- What are the recommended AMMU settings for M4?
- What are the different levels of cache supported by A15?
- What are the different levels of MMU supported by A15?
- Is there an example on A15 MMU and cache?

Interrupt Issues

- How to configure the interrupts in PDK for bare metal?
- How to configure the interrupts in PDK for RTOS?
- How to configure the IRQ Crossbar?
- What are common issues faced while integrating PDK CSL or baremetal example code in Vision SDK?

Secondary Boot loader

- How to get time stamp before a core is started by SBL?
- What are some of the common mistakes done while using QSPI boot mode?
- What are the common issues faced while using CCS and SBL at the same time for development?
- How to change I2C setting for adapting to custom PMIC in the SBL?

EDMA3LLD

- How to fix resource allocation failure?

CAN Controller(DCAN/MCAN)

- What is CAN?
- How is bit-rate calculated?
- What is sampling point and how is it calculated?
- What is CAN bus termination?
- What is Transceiver Delay Compensation Value(TDCV) and how is it calculated?
- Things to consider/check before sending a message over CAN bus
- CAN communication failed, where to start debug?

Firewall

- How to program firewall when EMIF is programmed in interleaved mode?

Generic

What does PDK stand for?

PDK stands for Platform Development Kit containing TI RTOS and baremetal drivers for various peripherals present in TI devices

What do I get with PDK, and how do I get started?

Each PDK release is available as a full source release including:

- Complete source code for all drivers, with associated build system
- Sample applications and unit test applications demonstrating the usage of all modules
- Release documents: Release Notes, User Guide

To get started with PDK, you can refer to the release documents mentioned in the release notes in the docs folder of the product. Refer to [PDK Userguide](#) for high level information. The doxygen API documentation present in docs folder provides details on the APIs, and the sample examples provided with PDK demonstrate usage of the APIs. The sample application is present in examples folder of each of the components within PDK.

How do I maintain my changes over PDK? How to pick up the latest bug fixes from a newer PDK release?

Kindly refer to http://processors.wiki.ti.com/index.php/PDK/Maitaining_Custom_Changes_Over_TI_Software_Release. This covers:

1. How to maintain custom changes over a TI software release
2. If in case you need to migrate to a different version of the same TI Software
3. How to take in Bug fixes from the latest and greatest TI Software release without really having to migrate to the latest version. (Note: This is available from the PDK 01.09.XX.XX version)

Board Porting

How to change UART instance in VSDK and PDK?

Change the UART1 pinmux setting in function BSP platform file as per your platform. For example for TDA2xx, change in Bsp_platformTda2xxSetPinMux() in file ti_components/drivers/pdk_<VERSION>/packages/ti/drv/vps/src/platforms/src/bsp_platformTda2xx.c per the UART pin that is needed. Below is an example to change to UART3

```
Bsp_platformSetPinmuxRegs((UInt32) 0, (UInt32) CTRL_CORE_PAD_UART3_TXD,
    BSP_PLATFORM_IOPAD_CFG_INPUTENABLE_OUT |
    BSP_PLATFORM_IOPAD_CFG_PULLUDENABLE_ENABLE |
    BSP_PLATFORM_IOPAD_CFG_PULLTYPESELECT_UP);
Bsp_platformSetPinmuxRegs((UInt32) 0, (UInt32) CTRL_CORE_PAD_UART3_RXD,
    BSP_PLATFORM_IOPAD_CFG_INPUTENABLE_BI |
    BSP_PLATFORM_IOPAD_CFG_PULLUDENABLE_DISABLE |
    BSP_PLATFORM_IOPAD_CFG_PULLTYPESELECT_DOWN);
```

In case of VSDK application, change the UART driver create in VSDK to point to the required UART instance in file vision_sdk\links_fw\src\rtos\utils_common\src\utils_uart.c in fxn System_uartInit()

In case of PDK application, do the similar change in ti_components/drivers/pdk_<VERSION>/packages/ti/drv/vps/examples/utility/src/bsputils_uart.c

Similar query can be found in the E2E forum https://e2e.ti.com/support/arm/automotive_processors/f/1021/p/665277/2446737

Example Porting

How to change IP instance for an example in PDK?

TDA2/TDA3 SoCs have multiple instances of the same IP. User needs to take care of the below items while porting the example:

PRCM: While porting the example to a new instance, say porting the example from UART1 to UART5; in order to make sure that the particular module is powered on user needs to take care of PRCM. PDK contains full fledged PM driver that can be used to enable PRCM. The APIs for setting clock domain and module enable are PMHALCMSetCdClockMode and PMHALModuleModeSet respectively. Alternatively user can choose to write the PRCM registers. Registers to be searched in TRM are CM_<Clock domain>_CLKSTCTRL and CM_<Power domain>_<Module>_CLKCTRL.

Pin Mux: Different instances of an IP are connected on different pins. User needs to make sure that he makes appropriate changes in the Pin Mux configuration. Pin mux registers are defined in Control Module.

Build

Can I build PDK on a Linux machine?

Yes. PDK build is based on makefiles. Edit the pdk/packages/ti/build/Rules.make file under #Paths for linux machine section to point to the tools path. Then give the following command from prompt

```
make -s all
```

Notes: "-j" option is also support by make. One could use this to do parallel build and speed up the build time.

How do I build PDK library for a particular Platform?

The build platform can be provided through command line which is illustrated below

```
make -s all BOARD=tda2xx-evm
```

The supported platforms are given in the top level pdk/packages/ti/build/makefile.

Can I selective build VIP driver only without DSS and VPE drivers?

Yes. It is possible to build only VIP or only DSS or only VPE or any two combination of above three so that other drivers are compiled out resulting in reduction in memory.

The following command could be used

For VIP only build:

```
make -s pdk_libs PACKAGE_SELECT=vps-vip-only
```

For DSS only build:

```
make -s pdk_libs PACKAGE_SELECT=vps-dss-only
```

For VPE only build:

```
make -s pdk_libs PACKAGE_SELECT=vps-vpe-only
```

How can I build PDK examples while using Processor SDK Vision (AKA Vision SDK)?

PDK's build folder is different from the build folder of Vision SDK. In order to build PDK examples change the current directory to <vstdk_version>/ti_components/drivers/<pdk_version>/packages/ti/build. Modify paths for tool chains and PDK_INSTALL_PATH appropriately. The following command could be used to build examples:

```
make -s examples
```

For more details on build commands, refer to http://processors.wiki.ti.com/index.php/PDK/PDK_TDA_Building_The_Package

What are the common issues faced during the build?

Windows Permissions

Sometimes there are issues with windows permissions. User should make sure that he has full permissions/admin rights while trying the build. User can use `ls -l` command to check the permissions and use `chmod` command to change the permissions.

Compiler Versions

PDK supports multiple CPUs and a particular PDK release is validated with a particular versions of compilers for different CPUs. It is not guaranteed that PDK examples/source code will work with different version of compilers. It is recommended that user refers to PDK documentation for supported compiler versions and use the same in order to avoid issues.

Compiler Paths

Different compiler paths are defined in Rules.make. User should make sure that appropriate paths are set in case the make is not able to find compiler executable while building. Also Vision SDK releases do not include the A15 compiler. The same needs to be downloaded from the internet and installed. User should make sure that they have installed A15 compiler in case they are getting createprocess error.

Building with -j

Sometimes the build fails while using the make with -j option. This command is used to speed up the builds by invoking multiple builds at once and is not 100% reliable depending on the build machine. In case the build fails user should try to build without giving -j switch in the build command.

CPU Architecture

Where is the source code for different CPUs supported by PDK?

PDK supports multiple CPU architectures. The source code for different architecture is present here: `pdk/packages/ti/csl/arch/<cpu_arch>`

What are the different levels of cache supported by M4?

Although IPU subsystem has two M4 cores, both cores share a single cache called IPUx_UNICACHE.

What are the different levels of MMU supported by M4?

IPU subsystem has single MMU called IPUx_UNICACHE_MMU or AMMU.

What are the recommended AMMU settings for M4?

The recommended AMMU settings are given in `pdk/packages/ti/boot/sbl_auto/sbl_utils/src/tda3xx/sbl_utils_tda3xx_ammu_config.h`. The same is used by TDA3xx SBL. Gel files also contain the recommended AMMU settings. Look at `C:\ti\<ccsv8>\ccs_base\emulation\gel\<TDAxx>\<TDAxxx>_multicore_reset.gel`

What are the different levels of cache supported by A15?

A15 supports two level of cache L1 and L2. At L1 level, there are two caches L1D (for data) and L1P (for program). Each L1 cache has size 32 KB. At L2 level there is a unified cache.

What are the different levels of MMU supported by A15?

MMU allows the A15 processor to map a 32-bit virtual address to a 40-bit physical address (supports Large physical address extensions). The MMU hardware supports upto 3-levels of translation tables but PDK only supports the first 2 levels of translation tables. The level1 translation table is a 32 byte descriptor table comprising of four 64bit entries. There can be upto 4 level2 translation tables, each 4KB in size and comprising of 512 64bit entries.

1. Each entry in the level1 table either gives the base address and defines the attributes of a memory area of size 1GB or gives the address of the next level of translation table and some attributes for that translation.
2. Each entry in the level2 table either gives the base address and defines the attributes of a memory area of size 2MB or gives the address of the next level of translation table and some attributes for that translation.

Is there an example on A15 MMU and cache?

Example for A15 MMU and cache is here: `pd\packages\ti\csl\example\mmu\mmu_a15_data_validation\mmu_a15_data_validation_app_main.c` and look at http://processors.wiki.ti.com/index.php/PDK/PDK_TDA_CSL_User_Guide#A15_MMU_Example for example description

Interrupt Issues

How to configure the interrupts in PDK for bare metal?

APIs to configure interrupts are defined in `pd\packages\ti\csl\arch<cpu_name>\interrupt.h`. User can refer to `TimerIntcInit` API in `pd\packages\ti\csl\example\timer\timer_app\main<cpu_name>.c` on how to configure interrupts.

How to configure the interrupts in PDK for RTOS?

RTOS drivers use OSAL for configuring interrupts and these APIs are defined in `pd\packages\ti\drv\vps\include\osal\bsp_osal.h`. User can refer to VPS drivers on how to configure interrupts using OSAL APIs.

How to configure the IRQ Crossbar?

In TDA2/TDA3 family there are multiple interrupt sources and all interrupt sources are not directly connected to CPUs. Hence user might need to configure interrupt crossbar. The APIs to configure crossbar in bare-metal/CSL context are defined in `pd\packages\ti\csl\soc<soc_name>\csl_device_xbar.h`. APIs to configure crossbar in RTOS context are defined in `pd\packages\ti\drv\vps\include\osal\bsp_osal.h`.

What are common issues faced while integrating PDK CSL or baremetal example code in Vision SDK?

One common mistake while integrating PDK code in Vision SDK is interrupts. Interrupt APIs from baremetal code cannot be used in Vision SDK. E.g. `Intc_Init` API will reset the CPU's interrupt controller and lead to hang in Vision SDK. Hence user should take care of not calling any `Intc_<name>` APIs in Vision SDK. User should use OSAL APIs for registering interrupts and configuring interrupt crossbar when using CSL in Vision SDK context.

Secondary Boot loader

How to get time stamp before a core is started by SBL?

TDA SoCs have 32K Timer which is clocked by the 32-KHz system clock. The same counter can be read by calling the API `SBLUtilsRead32KTimer()`. User can use this API to get the time stamp before the core boot up in SBL. User can refer to TDA2xx SBL (`pd\packages\ti\boot\sbl_auto\sbl_app\src\tda2xx\sbl_tda2xx_main.c`) to see how 32K timer is used to get the time stamp before IPU1_0 is started by SBL.

What are some of the common mistakes done while using QSPI boot mode?

There is one common issue with flashing and booting SBL in QSPI boot mode: Flashwriter will leave QSPI flash in either QSPI-1 or QSPI-4 mode after completion depending on user selection at the time of flashing. User has to make sure SYSBOOT setting is set to use the corresponding QSPI-1 or QSPI-4 boot mode otherwise SBL won't come up.

What are the common issues faced while using CCS and SBL at the same time for development?

1. Ideally when the user wants to load SBL using debugger gel files should not be present in the target configuration. In case gel files are present and SBL is loaded using debugger it might initialize the IPU's AMMU in such a way that it clashes with gel file configuration which leads to exceptions. To avoid this gel files should be modified to have same AMMU configuration as SBL. Same is applicable for EVE MMU.
2. Many times the dip switches i.e. SYSBOOT is set in a way where some pre-flashed SBL boots. This can be avoided by setting SYSBOOT to a reserved/debug boot mode. Please refer to SBL user guide for DIP switch settings for debug boot mode.

How to change I2C setting for adapting to custom PMIC in the SBL?

User needs to define the new PMIC Voltage regulator table for the custom PMIC. E.g. `gJ6Eco17x17LP8733RegulatorTable` is defined in `pdh/packages/ti/boot/sbl_auto/sbl_utils/src/tda2xx/sbl_utils_tda2xx.c`. Also care should be taken care that the corresponding I2C instance is enabled by PRCM while doing I2C initialization. For more details refer to section 3.2 "Taking Care of Differences in Device Voltage Rail and PMIC Integration" of <http://www.ti.com/lit/an/sprac22/sprac22.pdf>.

EDMA3LLD

How to fix resource allocation failure?

The EDMA resources like DMA Channels, PaRAMs, QDMA Channels are shared between applications running on all cores. So in EDMA3LLD the resources are statically allocated to of the each cores. The "EDMA3_DRV_open" function takes a parameter of type "EDMA3_DRV_InstanceInitConfig", which specifies the resources owned by the core. Please check the bit corresponding to the resource no is set. Ex to allocate the DMA channel number 5, make sure the bit 5 in EDMA3_DRV_InstanceInitConfig->ownDmaChannels[0] is set.

In case of Vision SDK the EDMA resources are partitioned in files present in "vision_sdk\links_fw\src\rtos\utils_common\src\dma_cfg"

CAN Controller(DCAN/MCAN)

What is CAN?

CAN is:

1. a multi-master broadcast serial bus standard for connecting electronic control unit
2. a serial asynchronous bus protocol, message-based protocol designed automotive applications but also used for industrial applications.
3. a differential pair signal and could be twisted without issue.

How is bit-rate calculated?

Formula for bit-rate is:

$$\text{bit rate (bits per second)} = (\text{CAN clock in Hz}) / \text{BRP} / (1 + \text{TSEG1} + \text{TSEG2})$$

where: CAN clock is functional clock of CAN module

BRP: Bit rate pre-scalar value

TSEG1, TSEG2: Time segments used to define sampling point for the bit.

What is sampling point and how is it calculated?

Sampling point is the point of time at which the bus level is read and interpreted as the value at that respective time. Typical value of sampling point is between 75-90%.

Sampling point is calculated as:

$$\text{Sampling Point(\%)} = (1 + \text{TSEG1}) / (1 + \text{TSEG1} + \text{TSEG2})$$

TSEG1, TSEG2: Time Segments expressed in terms of number of T_q (Time Quantum)

T_q : the period of the CAN module functional clock.

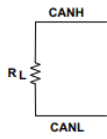
What is CAN bus termination?

CAN bus termination is needed to minimize the reflection, to reduce the noise. The transmission lines must be terminated to make sure noise does not cause communication failure.

There are two types of CAN termination:

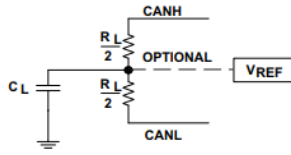
1. Standard termination:

The traditional 120 Ohm resistor(R_t) is used on each end of the bus as shown in the following figure.



2. Split Termination

Two 60 Ohm($R_L/2$) resistors along with capacitor(C_L) is used to suppress the noise at both ends of the bus. Typical value of C_L for 1.1 Mbps is 4.7 nF. It is shown in the following figure



NOTE: R_L and C_L values are dependent on a signaling rate.

What is Transceiver Delay Compensation Value(TDCV) and how is it calculated?

CAN is a broadcast protocol, every node on the bus sees the message on the bus even if it is Tx node. Tx node uses this data on Rx line for integrity and error checks locally.

Overview of bit flow through different stages in communication is:

CAN module core → CAN module Tx line → SoC Tx Pin/PAD → Transceiver Out → CAN Bus → Transceiver In → SoC Rx Pin/PAD → CAN module Rx line → CAN module core

Above whole path introduces latency between data going out from Tx line to data coming in on Rx line. Major part of this latency comes from Transceiver. Hence it is called Transceiver Delay Compensation Value.

This needs to be programmed as controller has to adjust its Secondary Sampling Point(SSP) for sampling Rx line at the correct time.

- MCAN provides functionality to measure delay between transition on Tx line and same transition for Rx line. User need not to measure transceiver loop-back delay or refer transceiver documentation for the same. User just needs to program SSP position which could be same as Sampling Point(SP).

Things to consider/check before sending a message over CAN bus

Make sure following is done/checked before attempting communication on bus:

- ISO-11898 compatible transceivers have to be connected to a CAN bus.
- CAN bus should have proper termination.
- CAN_L lines from nodes shall be connected together and same goes for CAN_H and GND lines as well.
- Make sure all connections are firm and all pins properly connected.
- All nodes on the bus are configured for same bit rate.
- In case of CAN FD, make sure Transceiver Delay Compensation value is correctly programmed.
- CAN module is powered up and configured properly.
- SoC pins/PADs for CAN are configured properly.

CAN communication failed, where to start debug?

When any CAN message transmission/reception fails, controller updates LEC(Last Error Code) and DLEC(Data Phase Last Error Code) fields in status register('MCAN_PSR' in case of MCAN and 'DCAN_ES' in case of DCAN).

LEC: this field is updated when error happened in arbitration phase in case of CAN FD message or during full message for classic CAN message.

DLEC: this field is updated when error happened in data phase in case of CAN FD message.

LEC/DLEC fields have following errors:

1. Stuff Error

- Node type:** Rx
- Description:** More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
- Check if all the nodes on the bus have same bit-rate.

2. Form Error

- Node type:** Rx
- Description:** A fixed format part of a received frame has the wrong format.
- This could happen if receiver node has drift in CAN functional clock momentarily or bus has interference.

3. AckError

- Node Type:** Tx
- Description:** The message transmitted by the MCAN module was not acknowledged by another node.
- CAN bus should have more than one node (apart from transmitting node) up and running (not in sleep or power down state).

4. Bit1Error and Bit0Error

- 9/25/2020PDK/PDK FAQ - Texas Instruments Wiki
- *Node Type:* Tx
 - *Description:*
 - *Bit1Error:* During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value 1), but the monitored bus value was dominant.
 - *Bit0Error:* During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value 0), but the monitored bus value was recessive.
 - Tx and Rx lines are sampled at Sampling Point(SP) and Secondary Sampling Point(SSP) respectively have different values. Position of SSP is configurable.
 - This can happen in following cases:
 - Check if tx and rx pins are configured properly. These pins should be pulled high by default i.e. when bus is idle(no communication is going on).
 - Check if CAN transceiver is enabled.
 - *For LEC:*
 - If configured SP is too large for given node.
 - Segment after SP(TSEG2) should be large enough to compensate for Transceiver Delay.
 - *For DLEC:*
 - Configured Transceiver Delay Compensation value is less/more than required.

2. CRCError

- *Node type:* Rx
- *Description:* The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data.
- This could happen if receiver node have drift in CAN functional clock momentarily or bus have interference.

Firewall

How to program firewall when EMIF is programmed in interleaved mode?

When operated in interleaved mode, both EMIFs have half of the address space.
While programming EMIF(operating in interleaved mode) FW, application need to program the offset to the half of it's original value. So,

```
start offset = (start address - base address)

end offset  = (end address - base address)

start address of the region: (start offset / 2)

end address of the region: (end offset / 2)
```

For example, to protect address region from 0x81000000 to 0x8108000. Parameters provided to L3FW LIB API(FWLIB3ConfigRegion) will be as follows:

```
regionCfgPrms.targetBaseAddr  = 0x80000000;

regionCfgPrms.regionStartAddr = 0x80800000; /* 0x80800000 = 0x80000000 + (0x100000 / 2) */

regionCfgPrms.regionEndAddr   = 0x80804000; /* 0x80804000 = 0x80000000 + (0x1008000 / 2) */
```

Note: Normally, the regions in firewall are configurable with a size of 4-KiB granularity. But in case of interleaved mode, this granularity becomes 8-KiB due to '/2' factor. 8-KiB = 0x2000.
Note: This **offset correction** (of '/2' factor) is only needed if **EMIF** is operating in **interleaved** mode. In case of **non-interleaved** mode of operation, this correction **should not be applied**.

Keystone=

1. switchcategory:MultiCore=

- For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **PDK/PDK FAQ** here.

For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum

For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **PDK/PDK FAQ** here.

C2000=For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article **PDK/PDK FAQ** here.

DaVinci=For technical support on DaVincoplease post your questions on The DaVinci Forum. Please post only comments about the article **PDK/PDK FAQ** here.

MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article **PDK/PDK FAQ** here.

OMAP35x=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **PDK/PDK FAQ** here.

OMAPL1=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **PDK/PDK FAQ** here.

MAVRK=For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article **PDK/PDK FAQ** here.


For technical support on MAVRK please post your questions at <http://e2e.ti.com>. Please post on comments about article **PDK/PDK FAQ** here.

Links

[Processors](#)

https://processors.wiki.ti.com/index.php/PDK/PDK_FAQ

7/8

	Amplifiers & Linear	DLP & MEMS	<ul style="list-style-type: none">▪ ARM Processors▪ Digital Signal Processors (DSP)▪ Microcontrollers (MCU)▪ OMAP Applications Processors	Switches & Multiplexers
	Audio	High-Reliability		Temperature Sensors & Control ICs
	Broadband RF/IF & Digital Radio	Interface		Wireless Connectivity
	Clocks & Timers	Logic		
	Data Converters	Power Management		

Retrieved from "https://processors.wiki.ti.com/index.php?title=PDK/PDK_FAQ&oldid=236614"

This page was last edited on 26 April 2019, at 06:54.
Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.