# PCIe Low Level Driver

# Release Notes

Applies to Product Release: 02.02.00.04
Publication Date: March 16, 2016



**Contributors to this document**



Texas Instruments, Incorporated
20450 Century Boulevard
Germantown, MD 20874 USA

# Contents

**Release Notes**

# PCIe Low Level Driver version 02.02.00.04

## Overview

This document provides the release information for the latest PCIe LLD which should be used by drivers and application that interface with PCIe.

PCIe LLD module includes:

- Compiled library (Big and Little) Endian of PCIe Low Level Driver.
- Sources, examples and unit test code.
- API reference guide

## LLD Dependencies

LLD is dependent on following external components delivered in PDK package:

- CSL

## New/Updated Features and Quality

**Release 2.2.0.4**

- Added support for K2G.
- Added ARM test/example support for applicable devices.
- Added new example for EDMA (Read/Write) operation from RC to EP and vice versa.
- Fixed Klocwork/Misra-C warnings

**Release 2.2.0.3**

- Use newly added CSL defines for MSI and PM functions, remove direct register pokes from example.
- Added lib and test/example support for Keystone I
- Added benchmarking support
- Fixed Klocwork/Misra-C warnings

- Use board_init() in test/example and change the printf() to uart instead of CCS console

**Release 2.2.0.2**

- Double each example such that one uses the soc file pcie_soc.c (wSocFile) while the other uses the soc library (wSocLib) which already contains pcie_soc.o. This ensures both libraries get tested, but is no otherwise necessary to look at both versions.
- Remove CSL redefines since they were fixed in CSL.
- Adjust delay count in pcie_sample_board.c:PlatformPCIESS1CtrlConfig() to match TRM. This shouldn't change results, previous sequence was not seen to fail.

**Release 2.2.0.1**

- Add MSI/Legacy interrupt to example.
- Route example printf() to uart instead of ccs console.

**Release 2.2.0.0**

- Add support for AM572X/AM571X.

**Release 2.1.0.2**

- Updated serdes init sequence in pcie_sample.c.

**Release 2.1.0.1**

- Added device library support (precompiled pcie_device.c) into DSP libraries (lib/k2*/c66/ti.drv.pcie.a*).

**Release 2.1.0.0**

- Added k2e and k2l.
- Example NOT extended to support both pcie.

**Release 2.0.0.4**

- Add a pcie_device.c for each device, and remove cslr_device.h from the LLD. This allows support for more than one interface (up to 4), and allows one library to be used on more than one device. The user may compile device/k2h/src/pcie_device.c or device/k2k/src/pcie_device.c or define Pcie_InitCfg themselves.
- Note: Pcie_setMode configures all devices. Change to Pcie_setInterfaceMode to configure one interface.
- Note: LLD no longer touches kicker. Application should unlock kicker once before calling Pcie_setMode and leave unlocked

**Release 2.0.0.3**

- Renamed the device specific folders as per new naming conventions.
- Support for TCI6636K2H device (k2h).

**Release 2.0.0.2**:

- Updates for using auto-generated cslr_device.h and csl_device_interrupt.h files.

**Release 2.0.0.1**:

- Modification for single LLD library to work for all platforms. Moved the default location of C66x libraries to lib\c66x inside component directory

**Release 1.0.0.3**:

- Resolved Linux host compilation issue with example projects

**Release 1.0.0.2**:

- Added makefile support
- Simplified and automated process of LLD version update
- Complete functional API for all PCIe registers.
- Enable the -dpcie_DEBUG flag to bounds-check all input parameters. The default is disabled.
- Enable the configuration of all the registers other than the BAR registers via a single API call. The registers which took an index plus a value are expanded into an array of values.
- Enhance doxygen to cover descriptions of all register fields.

**Release 1.0.0.1**:

- Deprecated support for C64P ELF and COFF. Only C66 ELF is supported now
- In the example, block coherent API for L1D, L1P and L2 have been modified to use CACHE_FENCE_WAIT enumeration. This enumeration internally uses the C66 mfence instruction which is recommended for all block coherence cache operations.

**Release 1.0.0.0**:

- Initial Release

## Resolved Incident Reports (IR)

Table 1 provides information on IR resolutions incorporated into this release.

**Table 1 Resolved IRs for this Release**

| IR Parent/ Child Number | Severity Level | IR Description |
|---|---|---|
| SDOCM00121387 | Major | PCIE lld is DOA for k1/k2 platform since mcsdk merge |
| SDOCM00121689 | Major | RTOS: PCIE ARM unit test projects get an exception when run on AM571x-IDK |

| IR Parent/ Child Number | Severity Level | IR Description |
|---|---|---|
| SDOCM00121575 | Minor | Pcie EDMA Arm projects fails to finish succesfully every 1 out 20 times. |

## Known Issues/Limitations

**Table 2 Known Issue IRs for this Release**

| IR Parent/ Child Number | Severity Level | IR Description |
|---|---|---|
| | | |

## Licensing

Please refer to the software Manifest document for the details.

## Delivery Package

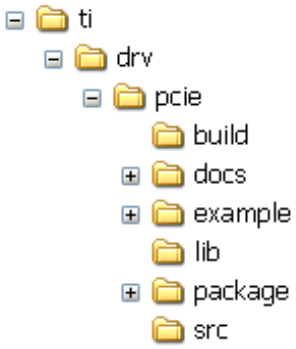
There is no separate delivery package. The PCIe LLD is being delivered as part of PDK.

## Installation Instructions

The LLD is currently bundled as part of Platform Development Kit (PDK). Refer installation instruction to the release notes provided for PDK.

### Directory structure

After installation, the PCIe LLD has the following directory structure:




The following table explains each individual directory:

| Directory Name | Description |
|---|---|
| `ti/drv/pcie` | The top level directory contains the following:- |

| | |
|---|---|
| | 1. *Environment configuration batch file*<br>The file "`setupenv.bat`" is used to configure the build environment for the PCIe low level driver.<br>2. *XDC Build and Package files*<br>These files (`config.bld, package.xdc` etc) are the XDC build files which are used to create the PCIe package.<br>3. *Exported Driver header file*<br>Header files which are provided by the PCIe low level driver and should be used by the application developers for driver customization and usage. |
| `ti/drv/pcie/build` | The directory contains internal XDC build related files which are used to create the PCIe low level driver package. |
| `ti/drv/pcie/docs` | The directory contains the PCIe low level driver documentation. |
| `ti/drv/pcie/example` | The "`example`" directory in the PCIe low level driver contains a simple example with edma enabled. |
| `ti/drv/pcie/lib` | The "`lib`" folder has pre-built Big and Little Endian libraries for the PCIe low level driver along with their *code/data size information*. |
| `ti/drv/pcie/package` | Internal PCIe low level driver package files. |
| `ti/drv/pcie/src` | Source code for the PCIe low level driver. |

## Customer Documentation List

Table 3 lists the documents that are accessible through the **/docs** folder on the product installation CD or in the delivery package.

**Table 3 Product Documentation included with this Release**

| Document # | Document Title | File Name |
|---|---|---|
| 1 | API documentation (generated by Doxygen) | docs/pcieDocs.chm |
| 2 | Release Notes (this document) | docs/ReleaseNotes_PCIe_LLD.pdf |