HYPLNK Low Level Driver

Release Notes

TEXAS INSTRUMENTS

Contents

HYPLNK Low Level Driver version 02.01.00.08

Overview

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

HYPLNK LLD module includes:
- Compiled library (Big and Little) Endian of HYPLNK Low Level Driver.
- Sources, examples and unit test code.
- API reference guide
- Design Documentation

LLD Dependencies

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

New/Updated Features and Quality

Release 2.1.0.8

- Updated buildlib.xs to add RULES_MAKE macro to support build based on custom Rules.make.

Release 2.1.0.7

- Fixed Hyplnk reboot sequence.

Release 2.1.0.6

- Added EDMA (Read/Write) operation over hyperlink.

- Added QMSS DMA (Packet transfer) operation over hyperlink.

Release 2.1.0.5

- Support both Keystone I and Keystone II devices from a single code base.

- Use Processor SDK's defines for device (SOC_### instead of DEVICE_###) and test project naming convention.

Release 2.1.0.4

- Modified Hyperlink serdes initialization sequence in the LLD example.

- Added new example to demonstrate generation and reception of cic interrupts

- Fixed linux mmap() of port 1

Release 2.1.0.3

- Changed device-dependent shared library to use common soname, libhyplnk_device.so.1

- Remove .created from libraries *.a.

Release 2.1.0.2

- Updated the default Serdes setup procedure at the Hyper Link common example per CSL serdes changes.

Release 2.1.0.1

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

- Added serialStop code to hyplnkExamplePeriphSetup() to prevent lockups.

Release 2.1.0.0

- Added k2e device

Release 2.0.0.10

- Added ARMv7 libraries that include precompiled hyplnk_device.c.

  - usr/lib/libhyplnk.a/so do not contain precompiled hyplnk_device.c. User application should compile hyplnk_device.c with any necessary customizations and use this library.

  - usr/lib/libhyplnk_k2[hk].a/so contains precompiled hyplnk_device.c for the specified device. This should not be used together with user-compiled hyplnk_device.c in order to avoid duplicate symbols from linker.

  - Similar functionality will be added to DSP at later date.

Release 2.0.0.9

- Added library support for ARMv7

- LLD example to include support for ARMv7

- LLD example to include a reset function at end of run

Release 2.0.0.8

- LLD example modification to support 312.5Mhz reference clock
- LLD example including support for SERDES debug API usage

Release 2.0.0.7

- Enhancement in LLD to support multiple ports
- Update in examples for multiple port testing
- Verification complete for LLD for 156.25 Mhz

Release 2.0.0.6

- Updated CSL name from GEM to C66X_COREPAC

Release 2.0.0.5

- Updated the interrupt name in the example project as per the change in CSL device interrupts file.

Release 2.0.0.4

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

Release 2.0.0.3

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

Release 2.0.0.2

- Modification for examples to support single library to work for all LLDs. Default location of C66x libraries to lib\c66x inside component directory

Release 1.0.1.3:

- Resolved Linux host compilation issue with example project

Release 1.0.1.2:

- Standalone delivery packaging
- SDOCM00084196: Project warning for hyplnkLLDPlatCfg.h

Release 1.0.1.1:

- SDOCM00082831: Minor cleanups in HyperLink LLD

Release 1.0.1.0:

- SDOCM00082831: Minor cleanups in HyperLink LLD
- SDOCM00082828: Two device HyperLink example

Release 1.0.0.4:

- SDOCM00081879: hyplnk_read_ECCErrors_reg() doesn't work. It accesses the wrong HW register

Release 1.0.0.3

- Added makefile support
- Simplified and automated process of LLD version update

Release 1.0.0.2

- 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 1.0.0.1

- Changes for limiting doxygen requirement only during the release
- Copyright modification to TI BSD

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
PRSDK-2194	NA	Add RTOS Installer script to autoset SDK_INSTALL_PATH

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 HYPLNK LLD is being delivered as part of PDK.

Patches and Modifications to Tools

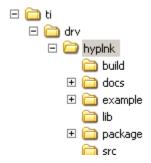N/A

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 HYPLNK LLD has the following directory structure:

The following table explains each individual directory:

Directory Name	Description
ti/drv/hyplnk	The top level directory contains the following:- 1. *Environment configuration batch file* The file "setupenv.bat" is used to configure the build environment for the HYPLNK low level driver. 2. *XDC Build and Package files* These files (config.bld, package.xdc etc) are the XDC build files which are used to create the HYPLNK package. 3. *Exported Driver header file* Header files which are provided by the HYPLNK low level driver and should be used by the application developers for driver customization and usage.
ti/drv/hyplnk/build	The directory contains internal XDC build related files which are used to create the HYPLNK low level driver package.
ti/drv/hyplnk/docs	The directory contains the HYPLNK low level driver documentation.
ti/drv/hyplnk/example	The "example" directory in the HYPLNK low level driver contains an

	example using bidirectional memory access with the peripheral via CPU, EDMA and InfraDMA operation. This also serves as the unit test.
`ti/drv/hyplnk/lib`	The "`lib`" folder has pre-built Big and Little Endian libraries for the HYPLNK low level driver along with their _code/data size information_.
`ti/drv/hyplnk/package`	Internal HYPLNK low level driver package files.
`ti/drv/hyplnk/src`	Source code for the HYPLNK 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/ hyplnkDocs.chm
2	Release Notes (this document)	docs/ReleaseNotes_HYPLNK_LLD.pdf
3	Software Manifest document	docs/HYPLNK_LLD_1_0_SoftwareManifest.pdf