RM LLD

Release Notes

TEXAS INSTRUMENTS

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

Contents

RM LLD version 01.00.00.11

Overview

This document provides the release information for the latest RM Low Level Driver which can be used by applications that want to manage QMSS, CPPI, and PA LLD resources.

RM LLD module includes:
- Pre-compiled library for DSP (Big and Little) Endian of RM LLD.
- Source code.
- API reference guide

LLD Dependencies

LLD is dependent on following external components delivered in PDK package:
- None

New/Updated Features and Quality

This is an **engineering release**, tested by the development team.

Release 1.0.0.11

- **SDOCM00095409** – Flow Id and Queue Numbers shared between ARM and DSP are not provided by RM

Release 1.0.0.10

- Updated Appleton resource table to reserve new resources used by ARM.

- **SDOCM00093407** – (Child) QM, CPPI, RM, SRIO LLD: osal defines memcpy and so on by itself
- **SDOCM00093494** – rm_testproject project gets an exception on termination due to corruption caused by non-cache aligned RM resource table
- **SDOCM00093554** – PDK C6614: The RM project's Qmss_setQueueThreshold test fails
- **SDOCM00094622** – RM LLD: Resource table PRAGMA for cache alignment needs to be modified for C++ compiler

Release 1.0.0.9

- **SDOCM00092636** – Appleton Resource Table file does not share Infrastructure Resources between DSP & ARM
- **SDOCM00093373** – Appleton resource table needs to be updated to reserve more channels for ARM UDMA
- **SDOCM00093485** – RM does not track all QMSS Accumulator channels

Release 1.0.0.8

- **SDOCM00092304** – RM checks in QM LLD functions (AckInterrupt/Eoi)

Release 1.0.0.7

- **SDOCM00092010** – PDK C6657: The RM project does not build because it is relying on PA which does not exist in the C6657 PDK

Release 1.0.0.6

- **SDOCM00091798** – RM LLD Global Data Structures Could cause data corruption due to no pad

Release 1.0.0.5

- **SDOCM00090437** – Appleton RM LLD test application's resource table must account for resources used by Linux

Release 1.0.0.4

- **SDOCM00090650** – PDK (6670 & 6678): RM project does not build

Release 1.0.0.3

- **SDOCM00090406** – RM LLD must switch off support for AIF, FFTC A, FFTC B, FFTC C, and BCP DMAs for Shannon

Release 1.0.0.2

- **SDOCM00090356** – rm_public_lld.h causes circular dependency when building standalone PA and QMSS LLDs

Release 1.0.0.1

- **SDOCM00089999** – c6670 RM LLD Missing CPPI FFTC C Resources

Release 1.0.0.0

- Initial release of low level driver

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
SDOCM00095409	S2 - Major	Flow Id and Queue Numbers shared between ARM and DSP are not provided by RM

Known Issues/Limitations

IR Parent/ Child Number	Severity Level	IR Description

Licensing

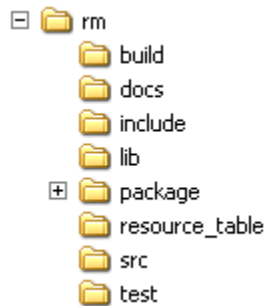Please refer to the SDK licensing document for the details.

Delivery Package

There is no separate delivery package. The RM 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

The following is the directory structure after the RM LLD package has been installed:

The following table explains each individual directory:

Directory Name	Description
ti/drv/rm	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 RM 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 RM package. 3. *Exported Driver header file*

	Header files which are provided by the RM low level driver and should be used by the application developers for driver customization and usage.
`ti/drv/rm/build`	The directory contains internal XDC build related files which are used to create the RM low level driver package.
`ti/drv/rm/docs`	The directory contains the RM low level driver documentation.
`ti/drv/rm/include`	The "include" directory has private RM low level driver header files. These files should not be used by application developers.
`ti/drv/rm/lib`	The "`lib`" folder has pre-built Big and Little Endian libraries for the RM low level driver along with their *code/data size information*.
`ti/drv/rm/package`	Internal RM low level driver package files.
`ti/drv/rm/resource_table`	The "resource_table" directory contains a device specific default resource table definition that assigns all DSPs full permissions to all resources on the device.
`ti/drv/rm/src`	Source code for the RM low level driver.
`ti/drv/rm/test`	The "`test`" directory in the RM low level driver has unit test cases which are used by the development team to test the RM low level driver.
`eclipse`	The "eclipse" directory has files required to integrate RM low level driver documentation with Eclipse IDE's Help Menu.

Customer Documentation List

Table 2 lists the documents that are accessible through the **/docs** folder on the product installation CD or in the delivery package.

Table 2   **Product Documentation included with this Release**

Document #	Document Title	File Name
1	API documentation (generated by Doxygen)	docs/rmlldDocs.chm