Resource Manager (RM)

Release Notes

Applies to Product Release: 02.02.00.03
Publication Date: July 17, 2018

Contributors to this document

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

Contents

Release Notes

RM version 02.02.00.03

Overview

This document provides the release information for the latest Resource Manager (RM) which can be used by applications that want to manage device resources.

RM includes:

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

Dependencies

RM 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 2.2.0.3

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

Release 2.2.0.2

- Resolved PRSDK-442
- Resolved PRSDK-747

Release 2.2.0.1

- Added support for Keystone II Gauss (K2G) device

Release 2.2.0.0

- Added support for Keystone I C6657 and C6678 devices
- Added A15 library build for Keystone II ARM RTOS use cases
- Added device-dependent library support

Release 2.1.2.0

- Improved RM Server resource request/response latency by ~90%. Moved all RM Policy DTB parsing to instance init. The policy is parsed during instance init and all policy info is stored in binary search trees for quicker access when checking the permissions of a request.
- Added new unspecified allocation permission. Any RM instance assigned an "e", or "E" for a set of resources in the policy will not be allocated the affected resources via a resource request with an UNSPECIFIED base value.
- RM Server socket is now accessible from RM Clients running in a non-root user process. The RM Server socket can be assigned a group permission when the server is started with the optional –g, or –group, [GROUP_NAME] input allowing access to non-root user's with the same group ID.
- RM server socket specified in rm_server_if.h public API. The RM Server socket file system path was buried in the sockrmmsg.h header which was replicated by any user-space application that wanted to connect to the RM Server. This was not sustainable or backwards compatible. Moving the RM Server socket to a public API allows any app that wishes to connect to the RM Server to reference the RM_SERVER_SOCKET_NAME #define in the rm_server_if.h API.
- Add resources for Cppi_open's register writes (as *-hw-open).
- QMSS LLD: Mask out queues (4095, 8191, 12287, 16383) because they are reserved as return queues from CPPI HW.
- Resolved all "-Wall –Wextra" GCC compiler warnings, except those found in the libfdt source code.
- Source code formatting cleanup and minor bug fixes
- Synced K2E and K2L DSP+ARM policy files with latest Linux Kernel DTB

Release 2.1.0.8

- Added new single DSP core, multi-task, test program for all K2 devices which exercises the request serialization capabilities of RM Clients when the Multi-Threaded Critical Section OSAL fucntions are implemented.
- Coverity fixes for Linux test examples.
- Resolved SDOCM00114533

Release 2.1.0.7

- Fixed potential memory leak when creating a resource allocator
- Fixed linker error when compiling ARM-DSP test's DSP client application for secure devices
- Synchronized qpend queues with latest linux dts
- Added srio RXU related resources.

- Added new Multi-Threaded Critical Section OSAL function. A multi-threaded semaphore object can be provided as part of the RM instance initialization. The RM instance will use the semaphore object to serialize any usages of the instance as long as the object is non-NULL.
- Added Multi-Threaded ARM Linux test case.

Release 2.1.0.6

- Fixed memory leak when extracting used resources from a Linux DTB.
- Granted GIC queue access to a new valid instance, "RM_Client_NETAPI", used by the NETAPI ipsecmgr daemon. This allows the daemon to open the GIC queues while still restricting access to the rest of the system.
- Reserved more QMSS infrastructure queues, CPPI QM1 rx channels, CPPI QM1 tx channels, and CPPI QM1 rx flows for the Kernel based on the formula "8 channels for data usage irrespective of device + 1 channel per DSP core"

Release 2.1.0.5

- Restricted access to netcp-rx/tx channels for K2L and K2E platforms.
- Gave full access to all PA user stats for all platforms.
- Updated QMSS Linking RAM indices used by Linux on all devices
- Fixed heap memory leak when checking resource request permissions
- Synchronized QMSS InfraDMA channels with queues (12 are used by linux).
- Set entire DDR3 region to be non-cached in all rmK2xArmv7LinuxDspClientTestProject RTSC .cfg files. The ARM-DSP test project uses IPC's RPMSG transport to send RM messages between ARM and DSP. The RPMSG transport makes use of a CMA region that must be non-cached. This CMA region is allocated at runtime. The entire DDR3 region is made non-cached as a workaround since the CMA region is unknown at compile time.

Release 2.1.0.4

- Removed duplicated rmServer source code in the test/ directory. The rmServer runs on ARM but, otherwise, is device independent. The rmServer source code has been relocated to the test/armv7/linux/ directory.
- Fixes for issues flagged by Coverity
- See Table 1 for resolved IRs

Release 2.1.0.3

- Sync with 2.0.0.9

Release 2.1.0.2

- Sync with 2.0.0.8

Release 2.1.0.1

- Sync with 2.0.0.6
- K2E and K2L support

Release 2.0.0.9

- Added two new transport APIs:
    - Rm_receiveGetPktServiceSrcName – Returns the RM instance name that originated the service request encapsulated within the RM packet.
    - Rm_receiveGetPktSrcName – Returns the RM instance name that originated the RM packet.
- Modified the Linux rmServer application so that it can be run as a daemon.

Release 2.0.08

- Added new resources for user stats counters in PA_LLD: 32bUsrStats and 64bUsrStats

Release 2.0.07

- Added shared object library support

Release 2.0.0.6

- See Table 1 for resolved IRs

Release 2.0.0.5

- Add Global resource list and Policy to support Qmss_QueueType_INTC_SET[234]_QUEUE to add all INTC(CIC) accessible queues.
- Service responses now contain the number of times the requesting instance has allocated the resource.
- Update policies and global resource lists to latest linux DTS.
- See Table 1 for resolved IRs

Release 2.0.0.4

- Added example that has Client running on DSP that communicates with a Server running in Linux user-space. Communication between ARM and DSP is over IPC MessageQ. Instructions for running the test application can be found in the RM User Guide: http://processors.wiki.ti.com/index.php/MCSDK_UG_Chapter_Developing_System_Mgmt#Running_RM_Test_Projects
- See Table 1 for resolved IRs

Release 2.0.0.3

- Added high priority accumulation and starvation counter queues to QM2.
- Extensive bug fixes and cleanup of RM source and test code
- Added ability to request a resource's reference count
- Added ability for Client Delegates to locally manage resources

- Added Server process for Linux
- Added Linux process to process test project that exchanges messages between a Client process and the latter Server process
- See list of Resolved IRs

Release 2.0.0.2

- Added Shared Server and Shared Client instance types that use shared memory to communicate internally. No transport code is needed for these instance types to communicate. Intended for use in DSP only, time-critical applications where, non-blocking APIs are required.
    - Includes example showing how to setup and use Shared Server/Client
- Added RM instance cleanup APIs
- Extensive bug fixes and cleanup of RM source

Release 2.0.0.1

- 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

IR Parent/ Child Number	Severity Level	IR Description
SDOCM00112301	S2-Major	Need way to use IPC RPMSG with Linux without marking the entire DDR3 as uncached

IR Parent/ Child Number	Severity Level	IR Description

Licensing

Please refer to the SDK licensing document for the details.

Delivery Package

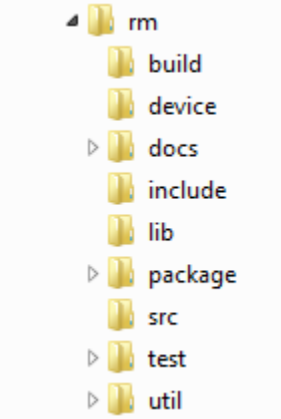
There is no separate delivery package. RM is being delivered as part of PDK.

Installation Instructions

RM 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 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 RM. 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 files* Header files which are provided by RM and should be used by the application developers for customization and usage.
`ti/drv/rm/build`	The directory contains internal XDC build related files which are used to create the RM package.

`ti/drv/rm/device`	The directory contains the device specific RM global resource list and policy files.
`ti/drv/rm/docs`	The directory contains the RM documentation.
`ti/drv/rm/include`	The "include" directory has private RM 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 RM along with their *code/data size information*.
`ti/drv/rm/package`	Internal RM package files.
`ti/drv/rm/src`	Source code for the RM.
`ti/drv/rm/test`	The "`test`" directory in RM has unit test cases which are used by the development team to test RM.
`ti/drv/rm/util`	The "util" directory has BSD-licensed, open source components used by RM.

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/rmDocs.chm