NXP Semiconductors
User's Guide

Document Number: MCSDK220GSUG
Rev. 4, 03/2017

Getting Started with Multicore SDK (MCSDK) v2.2.0

Contents

1 Overview

Multicore Software Development Kit (MCSDK) is a Software Development Kit that provides comprehensive software support for NXP dual/multicore devices. The MCSDK is combined with the MCUXpresso SDK to make the software framework for easy development of multicore applications.

The following figure highlights the layers and main software components of the MCSDK.

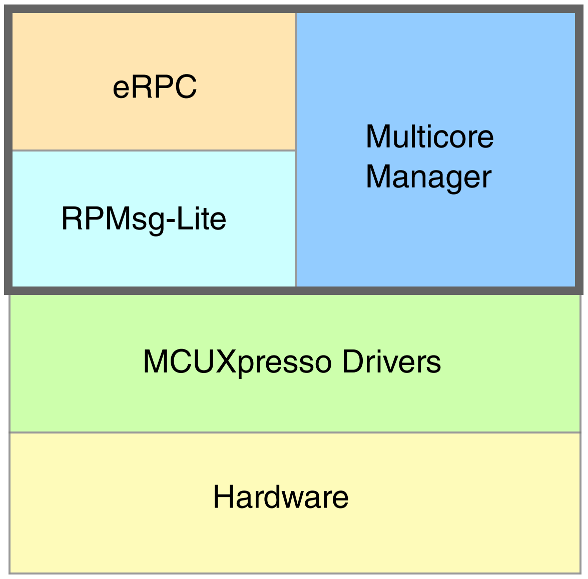

Figure 1. MCSDK layers

All the MCSDK-related files are located in *<MCUXpressoSDK_install_dir>/middleware/multicore_<version>* folder.

For supported toolchain versions, see the *Multicore SDK v.2.2.0 Release Notes* (document MCSDK220RN). For the latest version of this and other MCSDK documents, visit www.nxp.com.

2 MCSDK Components

The MCSDK consists of following software components:

- **Embedded Remote Procedure Call (eRPC)**:

  This component is a combination of a library and code generator tool that implements a transparent function call interface to remote services (running on a different core).

- **Multicore Manager (MCMGR)**:

  This library maintains information about all cores and starts up secondary/auxiliary core(s).

- **Remote Processor Messaging - Lite (RPMsg-Lite)**:

  Inter Processor Communication library.

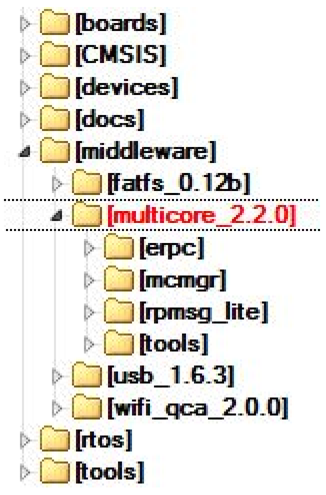

Figure 2. Multicore folder structure

2.1 Embedded Remote Procedure Call

eRPC (Embedded Remote Procedure Call) is the RPC system created by NXP. The RPC is a mechanism used to invoke a software routine on a remote system via a simple local function call.

When a remote function is called by the client, the function's parameters and an identifier for the called routine are marshalled (or serialized) into a stream of bytes. This byte stream is transported to the server through a communications channel (IPC, TPC/IP, UART, etc). The server unmarshalls the parameters, determines which function was invoked, and calls it. If the function returns a value, it is marshalled and sent back to the client.

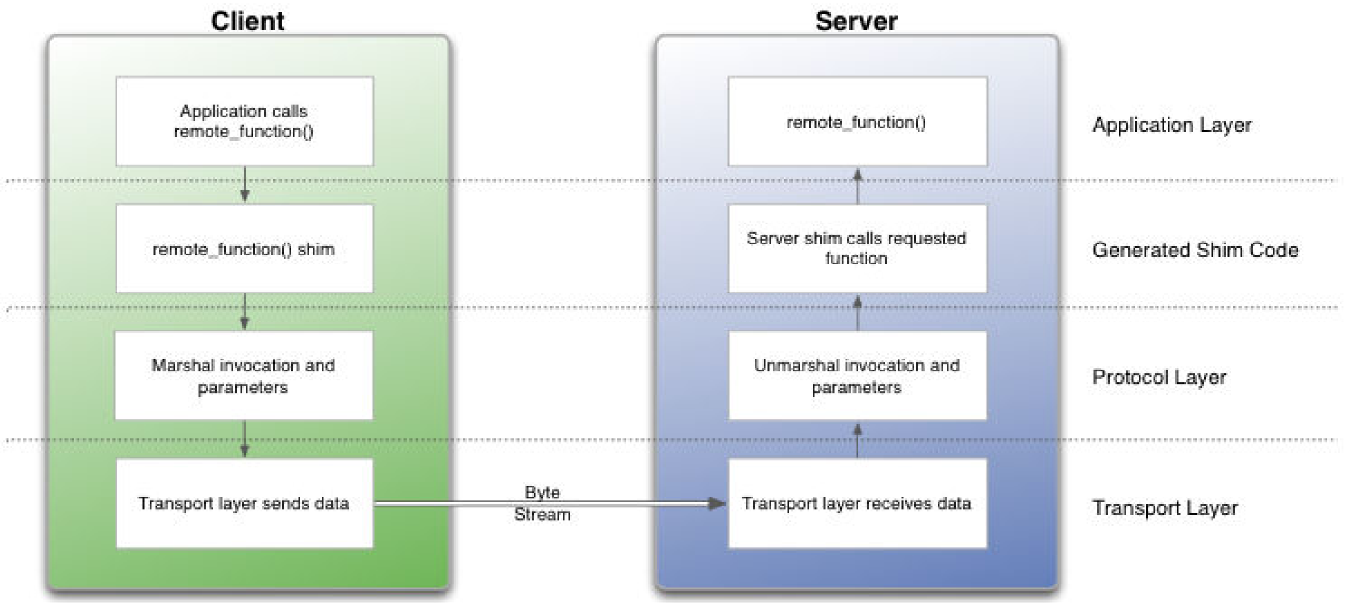
Client
Server
Application calls remote_function()
remote_function()
Application Layer
remote_function() shim
Server shim calls requested function
Generated Shim Code
Marshal invocation and parameters
Unmarshal invocation and parameters
Protocol Layer
Transport layer sends data
Byte Stream
Transport layer receives data
Transport Layer

Figure 3. eRPC Block diagram

RPC implementations typically use a combination of a tool (erpcgen) and IDL (interface definition language) file to generate source code to handle the details of marshalling a function's parameters and building the data stream.

Main eRPC features:

- Scalable from bare metal to Linux® OS - configurable memory and threading policies.
- Focus on embedded systems - intrinsic support for C, modular, and lightweight implementation.
- Abstracted transport interface - RPMsg is the primary transport for multicore, UART, or SPI-based solutions can be used for multichip.

The eRPC library is located in the *<MCUXpressoSDK_install_dir>/middleware/multicore_<version>/erpc* folder. For detailed information about the eRPC, see the documentation available in the *<MCUXpressoSDK_install_dir>/middleware/multicore_<version>/erpc/doc* folder.

2.2 Multicore Manager

The Multicore Manager (MCMGR) software library provides a number of services for multicore systems.

Main MCMGR features:

- Maintains information about all cores in system.
- Loading of code for secondary core (if code is located in RAM).
- Secondary/auxiliary core(s) startup and shutdown.
- Maintains table of address mapping for shared memory regions.

The MCMGR library is located in the *<MCUXpressoSDK_install_dir>/middleware/multicore_<version>/mcmgr* folder. For detailed information about the MCMGR library, see the documentation available in the *<MCUXpressoSDK_install_dir>/middleware/multicore_<version>/mcmgr/doc* folder.

2.3 Remote Processor Messaging Lite (RPMsg-Lite)

RPMsg-Lite is a lightweight implementation of the RPMsg protocol. The RPMsg protocol defines a standardized binary interface used to communicate between multiple cores in a heterogeneous multicore system. Compared to the legacy OpenAMP implementation, RPMsg-Lite offers a code size reduction, API simplification, and improved modularity.

Main RPMsg protocol features:

- Shared memory interprocessor communication.
- Virtio-based messaging bus.
- Application-defined messages sent between endpoints.
- Portable to different environments/platforms.
- Available in upstream Linux OS.

The RPMsg-Lite library is located in the *<MCUXpressoSDK_install_dir>/middleware/multicore_<version>/rpmsg-lite* folder. For detailed information about the RPMsg-Lite, see the *RPMSG-Lite User's Guide* located in the *<MCUXpressoSDK_install_dir>/middleware/multicore_<version>/rpmsg_lite/doc* folder.

3 MCSDK demo applications

Multicore and multiprocessor example applications are stored together with other MCUXpresso SDK examples, in the dedicated multicore subfolder.

Table 1. Multicore example applications

Location	Folder
Multicore example projects	<MCUXpressoSDK_install_dir>/boards/<board_name>/multicore_examples/<application_name>/<core_type>/<toolchain>
Multiprocessor example projects	<MCUXpressoSDK_install_dir>/boards/<board_name>/multiprocesor_examples/<application_name>/<core_type>/<toolchain>

See the *Getting Started with MCUXpresso SDK* (document MCUXSDKGSUG) and *Getting Started with MCUXpresso SDK for XXX Derivatives* documents for more information about the MCUXpresso SDK example folder structure and the location of individual files that form the example application projects. These documents also contain information about building, running, and debugging multicore demo applications in individual supported IDEs. Each example application also contains a readme file that describes the operation of the example and required setup steps.

4 Revision history

This table summarizes revisions to this document.

Table 2. Revision history

Revision number	Date	Substantive changes
0	09/2015	Initial release
1	03/2016	Updated for the SDK 2.0.0 and the MCSDK 1.1.0
2	09/2016	Updated for the MCSDK 2.0.0 and the LPCXpresso54114 support
3	09/2016	Updated for the MCSDK 2.1.0
4	03/2017	Updated for the MCSDK 2.2.0

Document Number MCSDK220GSUG
Revision 4, 03/2017