MCBSP LLD

Release Notes

Applies to Product Release: 01.00.00.02:
Publication Date: July 10, 2013

Document License

Contributors to this document

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

Contents

Release Notes

MCBSP LLD version 01.00.00.02

Overview

This document provides the release information for the latest MCBSP Low Level Driver which should be used by drivers and application that interface with MCBSP IP.

MCBSP LLD module includes:

- Pre-compiled library for DSP (Big and Little) Endian of MCBSP LLD
- Source code
- API reference guide
- Design Documentation

LLD Dependencies

LLD is dependent on following external components delivered in CSL/LLDs package:

- CSL
- EDMA3 LLD

New/Updated Features and Quality

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

Release 1.0.0.2

- Added support for ping pong mechanism using initially submitted frames.
- Added Example project for testing MCBSP_LOOPJOB_ENABLE with external clock
- Added Example project to test MCBSP_LOOP_PING_PONG with external clock
- Bug fixes (see Resolved IR section below)

Release 1.0.0.1

- Bug fixes (See IR section below)

Release 1.0.0.0

- Initial release of MCBSP 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
SDOCM00097283	Major	MCBSP LLD: The example project fails when using 8-bits and 16-bits sample size in Big Endian mode
SDOCM00097806	Minor	MCBSP LLD: Multichannel mode doesn't work if only selected timeslots are enabled
SDOCM00098654	Major	MCBSP LLD: Provide support for using ping/pong buffers for EDMA transfers
SDOCM00099497	Minor	Bug in mcbsp lld : bitReversal not set correctly
SDOCM00099500	Minor	Mcbsp example code: Incorrect FPGA setting to set the AMC_EN bits to route clocks to test header

Known Issues/Limitations

IR Parent/ Child Number	Severity Level	IR Description

Licensing

Please refer to the software Manifest document for the details.

Delivery Package

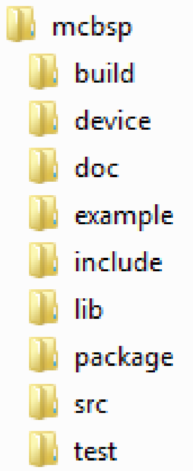
There is no separate delivery package. The MCBSP LLD is being delivered as part of CSL/LLDs package.

Installation Instructions

The LLD is currently bundled as part of CSL/LLDs package. Refer installation instruction to the release notes provided for CSL/LLDs.

Directory Structure

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



```
mcbsp
  build
  device
  doc
  example
  include
  lib
  package
  src
  test
```

The following table explains each individual directory:

Directory Name	Description
`ti/drv/mcbsp`	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 MCBSP 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 MCBSP package. 3. *Exported Driver header file* Header files which are provided by the MCBSP low level driver and should be used by the application developers for driver customization and usage.
`ti/drv/mcbsp/build`	The directory contains internal XDC build related files which are used to create the MCBSP low level driver package.
`ti/drv/mcbsp/device`	The directory contains the device specific files for the MCBSP low level driver.
`ti/drv/mcbsp/docs`	The directory contains the MCBSP low level driver documentation.
`ti/drv/mcbsp/example`	The "`example`" directory in the MCBSP low level driver has the Digital Loopback example.
`ti/drv/mcbsp/include`	The "include" directory has private MCBSP low level driver header files. These files should not be used by application developers.
`ti/drv/mcbsp/lib`	The "`lib`" folder has pre-built Big and Little Endian libraries for the MCBSP low level driver along with their *code/data size information*.
`ti/drv/mcbsp/package`	Internal MCBSP low level driver package files.
`ti/drv/mcbsp/src`	Source code for the MCBSP low level driver.

Example Sample Application

The section documents information about the example code located in the MCBSP LLD package.

MCBSP Digital Loopback Example

This sample application demonstrates the use of the MCBSP driver in digital loopback mode. The MCBSP driver supports only DMA mode of operation. The Mcbsp sample application has a project, called `MCBSP_DigLpbkExampleProject`.

This project (sample application) is used to configure the MCBSP in loopback mode and configures the required clocks, etc. The driver along with the required component modules are configured in `mcbspMasterDigLpbk.cfg` file. The required memory for the heap is also created here. The `mcbspMasterDigLpbk.cfg` file contains the remaining BIOS configuration like the configuration of the event combiner etc. This helps to map the MCBSP events to the CPU interrupts.

The `mcbspStartDigLpbkApp` function running in `mcbspDigLpbkAppTask` exercises the Mcbsp driver. The application uses MCBSP internal clock to drive frame sync and data bit clocks. The example application opens 2 channels (one RX and one TX) for the data transfer. A source buffer filled with known pattern is transferred from CPU memory to MCBSP peripheral (to DX pin - TX path) with the help of EDMA controller. The digital loopback mode in MCBSP loops this data back to the DR pin (RX path) and EDMA controller receives this data and stores in the destination buffer in CPU memory. The sample application sends and receives configured number of data bytes for multiple iterations and each time verifies the data integrity by comparing the transmitted and received data buffers.

MCBSP External Clock with LoopJob Example

This sample application demonstrates the use of the MCBSP driver with MCBSP_LOOPJOB_ENABLE with external clock. The Mcbsp sample application has a project, called `MCBSP_ExtClkLoopJobExampleProject`.

This does not use digital loopback. The application continually receives data frames from the Receive channel of Mcbsp and loops the data back to the transmit side. The data integrity needs to be checked externally by using external TDM cards and running a BERT test.

MCBSP External Clock with Ping pong Example

This sample application demonstrates the use of the MCBSP driver with MCBSP_LOOPJOB_ENABLE & MCBSP_LOOP_PING_PONG features with external clock. The Mcbsp sample application has a project, called `MCBSP_ExtClkPingPongExampleProject`.

Again does not use digital loopback. The application continually receives data frames from the Receive channel of Mcbsp and loops the data back to the transmit side. The data integrity needs to be checked externally by using external TDM cards and running a BERT test.

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/mcbsplldDocs.chm
2	Design Document	docs/MCBSP_LLD_SDS.pdf
3	Software Manifest	docs/MCBSP_LLD_SoftwareManifest.pdf