Release Notes

BIOSPSP DA830 02.00.01

June 06, 2009

This page has been intentionally left blank.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third–party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments

Post Office Box 655303

Dallas, Texas 75265

This page has been intentionally left blank.

TABLE OF CONTENTS

1 Introduction

PSP Package serves as fundamental software platform for development, deployment and execution of application software on DA830 based platform. PSP abstracts the functionality provided by the DA830[1].

PSP package is intended for the DSP that runs DSP/BIOS™ (user guide that came along with this release details the system requirements).

1.1 Release Category

This PSP release versioned 02.00.01 is a GA release for EVM DA830. Please refer to section Drivers/Components for this release for details of this release.

1.2 Text Conventions

O	This bullet indicates important information. Please read such text carefully.
q	This bullet indicates additional information.

[1] PSP may not provide abstraction for all the features provided by DA830.

2 Out-of-Box Contents

BIOSPSP_02_00_01_DA830_Setup.exe contains following:

1.	– Source code for driver and other necessary abstractions. – Project files (DSP/BIOS™), CCS4 build files and XDC package.bld files, to build PSP package in host environment. Please note that at this point of time CCSV4 does not support building xdc based libraries. Hence only option available as of now is to build the libraries through command prompt xdc command (Please refer the user guide available with this release for detailed build instructions) – The above mentioned items are located inside the <installation dir>\pspdrivers_02_00_01\packages\ti directory – Please note that the CCS setup files and GEL files are _NOT_ provided with this release and this would be available with latest CCSv4 releases or from EVM manufacturer.
2.	Release Notes (this document) providing an overview of this release.
3.	User Guide that provides information on package usage and each drivers usage.
4.	CDOC based driver API (generated) documentation for all the drivers inside package.
5.	High level design document of each driver (Complete list of the documents is available in the User Guide).

3 Drivers/Components for this release

This section lists the driver / components that are delivered as part of this release.

- DSP/BIOS™ 6.20.00.37 Drivers
  - Serial (UART, I2C and SPI)
  - Audio (McASP, AUDIO interface and CODEC)
  - GPIO
  - PSC (that helps to turn the clock on/off for the modules)
  - LCDC Raster controller driver
  - LCDC LIDD controller driver
  - MMCSD, NAND and Block Media drivers
  - CSLr and examples of CSLr usage
- Sample applications that demonstrate the usage of above drivers
- ERTFS File System is required if one wants to maintain a filesystem on Storage Media. Same can be downloaded from following link:

https://www-a.ti.com/downloads/sds_support/targetcontent/bios_file_system/index.html

Please note that at this point of time the drivers does not have any abstraction for the OS APIs and they use the OS (DSP/BIOS™ 6.20.00.37) inside the drivers.

3.1 Changes from previous release (Version 02.00.00.06)

1) This is version of BIOS PSP is compatible with the DSP/BIOS™ 6.20.00.37, XDC Tools version 3.15.00.50 and IPC 1.00.00.40. The eclipse help functionality and product view feature have been enabled in this release as plugins for use with CCSv4.
2) Implementation of IRs (Known issues and Bugs), raised on the previous release. Please refer to “Fixed in this release” section for a list of IRs fixed in this release

3.2 Other changes

None

4 Known Issues

4.1.1 MISRA C (All components)

- SDOCM00052184: MISRA C check was not run on the code base as we had issues in running MisraC tools on this package. (Tool had issues to scan the XDC header files and we are currently working on it).

  Workaround: None.

4.1.2 Instrumentation (All components)

- SDOCM00052185: Instrumentation code is not yet implemented as the XDC/CCSv4 version at the point of development did not support RTA and instrumentation APIs

  Workaround: None

4.1.3 LCDC Raster Device Driver (DSP/BIOS™)

- SDOCM00052150: LCDC raster driver sample application abruptly hangs while displaying frames continuously. This issue is being investigated.

  Workaround: Disable the palette loaded interrupt. Do not specify the interrupt mask for palette loaded interrupt in the raster configuration structure which is passed as channel parameters during channel creation.

- SDOCM00053815: The sample image displayed by the raster driver sample application has slight flickering. This issue is being analyzed

  Workaround: None

4.1.4 SPI Device Driver (DSP/BIOS™)

- SDOCM00058729: The SPI read operation may sometimes yield an erroneous byte during multiple read operations. This is being analyzed.

  Workaround: None.

4.1.5 McASP Device Driver (DSP/BIOS™)

- SDOCM00058757: The MUTE on/off IOCTL does not work as expected. While the audio is muted by issuing IOCTL the audio mutes. However, un-muting the same via IOCTL does not actually un-mute the audio. This issue is being investigated.
- SDOCM00058759: The loop job modification IOCTL does not work as expected. When the application configures the loop job with user supplied data for loop job via this IOCTL the same is not played. This issue is being investigated.
- SDOCM00058823: The Aic31.h file is not present in the package. This issue will be fixed in future releases. The applications may complain about this while compiling with this file included.

  Workaround: Please compile (run xdc command from command line) the codec library. This will generate the Aic31.h header file.

5 Limitations

5.1 I2C (DSP/BIOS™) Device Driver

- Loopback is not supported in interrupt mode
- In case of time bound IO requests, on timeout the driver is not able to perform any operations on the peripheral. (This peripheral limitation is documented in the technical reference manual of I2C under ICMDR register).
- I2C driver does not support slave mode of operation in polled mode. Only interrupt and DMA interrupt mode of operation are supported. The slave mode operation is tested successfully 100,200 and 400 kHz I2C clock frequency.
- (a) I2C slave application need to take care of the data (application level) protocol on when and what to receive and send by/from slave side.
- (b)This driver provides a generic bus communication path for slave.
- (c) Application protocol also needs to consider the latency caused by software slave implementation.
- (d) The driver does not support “0” no of byte transfer and the slave driver would not function properly if master issues a STOP condition immediately after a START condition.

5.2 SPI (DSP/BIOS™) Device Driver

- Slave mode of operation is tested at 2MHz. Because of the wired EVM to EVM connectivity, signal integrity was not good to test on further higher frequencies.
- Application protocol also needs to consider the latency caused by software slave implementation. (b) The driver does not support “0” no of byte transfer.

5.3 UART (DSP/BIOS™) Device Driver

- Loopback is not supported in DMA mode of operation.
- UART Baud rates greater than 115200 are not supported. This is due to high error percentage observed for baud rates greater than 115200. Please refer UART datasheet sprufm6, section 2.1 (Table 1) for more details.

5.4 GPIO (DSP/BIOS™) Device Driver

- The GPIO driver now supports only one instance, fixed number of banks (eight) and fixed number of pins per bank (sixteen). This is a limitation, as there are issues in getting initialization done for variable length arrays (inside structures, instance parameters etc) through the XDC framework.
- The GPIO driver provides the APIs, Gpio_(get/set)PinUseStatus and Gpio_(get/set)BankUseStatus for checking if the pin or bank is in use (as a functional pin and hence not available as GPIO). These, APIs should be used before calling any GPIO module APIs on setting data or status for the pins/banks. Though, GPIO driver shall make explicit check for use status for individual pin operations, it does not do it for group (or all pins in a bank) operations since it becomes an overkill every time, especially if the group of pins is used for data transfer etc. Hence, the application should make this check at least once before use of the required GPIO pins and then can proceed.

- SDOCM00048745: It is observed that even if the GPIO bank pin interrupts are enabled for rising edge only, the interrupt from the pin occurs at both rising edge and falling edge of the incoming signal at the pin. This issue is because of bouncing of signal at the GPIO pin when the SW3-1 state is changed. This bouncing of signal at the pin causes falling edges at the pin and hence triggers the falling edge interrupt. The user could use software de-bouncing techniques (like delay and read the steady state of the pin to check if it is spurious – falling edge – interrupt).

5.5 McASP (DSP/BIOS™) Device Driver

- Mcasp Driver does not support switching from DIT mode to TDM mode dynamically.

5.6 LCDC LIDD Controller Device Driver (DSP/BIOS™)

- The driver supports only character displays, up to 4 lines. Other STN displays are not supported.
- The driver supports only 8-bit data interface to the character displays.

5.7 evmDA830 EVM Limitation

- None

6 Fixed in this release

6.1 SPI (DSP/BIOS™) Device Driver

- SDOCM00048829 – The SPI driver was not tested in slave mode. This feature has been tested in this release. Please see the limitation section for details on limited feature tested in this mode.

  Files Modified: Spi.c, Spi_edma.c

- SDOCM00055031 – The SPI polled mode transfer would fail at all frequencies. This was due to improper handling of instance handler pointer. This issue has been fixed.

  Files Modified: Spi.c

6.2 I2C (DSP/BIOS™) Device Driver

- SDOCM00048829 – The I2C driver was not tested in the slave mode. This feature has been tested in this release. Please see the limitation section for details on limited feature tested in this mode

  Files Modified: I2c.c, I2c_edma.c

6.3 MMCSD (DSP/BIOS™) Device Driver

- SDOCM00057619 –The sample application would fail immediately (during initialization) if card was not inserted. This has been fixed. The card detection function does not return error now and the sample application can continue with other initialization.

  Files Modified: dda_mmcsdBios.c

- SDOCM00057621 - The clock divider values were being updated while the clock is running. This has been fixed in this release

  Files Modified: llc_mmcsd.c

6.4 BLKMEDIA (DSP/BIOS™) Device Driver

- SDOCM00057623 – The unregistration function would fail. This was because the resources were not properly freed in the completion callbacks. This is fixed int his release.

  Files Modified: blkmedia.c

- SDOCM00056285 – The block media support for two USB instances has been added in this release.

  Files Modified: blkmedia.c

6.5 McASP (DSP/BIOS™) Device Driver

- SDOCM00055111 – Dynamic codec configuration would fail when .

  Files Modified: Mcasp.c

- SDOCM00055113 – The McASP channel deletion would fail. This was because the order in which the Frame Sync and serializers were stopped was wrong in the channel deletion function. This has been fixed in this release.

Files Modified: Mcasp.c

- SDOCM00055497 – The submit function was trying to restart the EDMA transfer every time which was causing the events to get cleared because of which the transfer was stopping. The EDMA transfer function is now called only once during the start of the transfer

  Files Modified: Mcasp.c

6.6 Audio (DSP/BIOS™) Device Driver

- SDOCM00057458 - Critical section protection in the audio driver channel creation was not handled properly. This has been fixed in this release

  Files Modified: Audio.c

6.7 UART (DSP/BIOS™) Device Driver

- SDOCM00058583 – In DMA mode of operation FIFO under run errors were not handled by the UART interrupt handler. This would result in UART not sending any events to the EDMA in case of new reception. This is fixed in this release.

Special notes

- Please refer the user guide for installation, build instructions and tool version information.
- The resource allocation is the responsibility of application (system integrator)

7 Technical Support BIOS PSP

To submit questions about issues with this BIOS PSP drivers release please go to the external forums at http://community.ti.com/ or to http://support.ti.com for email support.

8 Release History

8.1 Release 02.00.00.06

This was a GA release for DA830

8.1.1 New in the release

- This release supports the DSP/BIOS™ 6.20 and XDC 3.15.00.50.
- Implementation of IRs (Known issues and Bugs) raised on the previous release. Please refer to "Fixed in this release" section for a list of IRs fixed in this release.

8.1.2 Fixed in this release

8.1.2.1 LCDC Raster Controller (DSP/BIOSTM) Device Driver

- SDOCM00055114 – The user supplied segment Id for dynamic allocation of raster frame buffers was not updated during channel open into channel object. This however, would not result in failure since it would take the default memory instance of the system. This issue has been fixed.

  Files Modified: Raster.c

8.1.2.2 MMCSD (DSP/BIOSTM) Device Driver

- SDOCM00055115 – The interrupt status bit for the GPIO pin used for card detection was not cleared in the interrupt handler registered by the sample application. This issue has been fixed.

  Files Modified: mmcsdSample_io.c

8.1.2.3 McASP (DSP/BIOSTM) Device Driver

- SDOCM00055116 – The calculation for submit buffer size in case of multiple serializers was wrong. This issue has been fixed.

  Files Modified: Mcasp.c

8.1.3 Known Issues

8.1.3.1 MISRA C (All components)

- SDOCM00052184: MISRA C check was not run on the code base as we had issues in running MisraC tools on this package. (Tool had issues to scan the XDC header files and we are currently working on it).

  Workaround: None.

8.1.3.2 Instrumentation (All components)

- SDOCM00052185: Instrumentation code is not yet implemented as the XDC/CCSv4 version at the point of development did not support RTA and instrumentation APIs

  Workaround: None

8.1.3.3 LCDC Raster Device Driver (DSP/BIOSTM)

- SDOCM00052150: LCDC raster driver sample application abruptly hangs in debug mode. This issue is being investigated. However, this issue is not observed in release mode.

  Workaround: None

  SDOCM00053815: The sample image displayed by the raster driver sample application has slight flickering. This issue is being analyzed

  Workaround: None

8.1.3.4 SPI Device Driver (DSP BIOSTM)

- SDOCM00051115: The SET_POLLEDMODETIMEOUT ioctl call can be used to set data transfer timeout values for polled mode of operation. If the timeout value is so less that the timeout actually occurs, before completion of data transfer, the driver raises an error and updates the error status in the io-packet. However, after this the subsequent Stream calls on the driver are failing because the stream aborts with the bad mode. This ideally should not occur and seems like an issue with the Stream layer. This issue has been raised to the Stream/DSP BIOS™ team as an issue (SDOCM00054581) and is being analyzed.

  Workaround: Use sufficiently large timeout values that do not actually cause a timeout.

- SDOCM00055031 – The read/write operation via SPI driver in polled mode in release profile build. This error is due to write buffer corruption in the driver. This issue is being analyzed and will be fixed in future releases.

  Workaround: Use interrupt or DMA mode of operation

8.1.3.5 I2C Device Driver (DSP BIOSTM)

- SDOCM00048829: The slave mode of configuration of the device driver is yet to be tested

  Workaround: None

- SDOCM00051115: The SET_POLLEDMODETIMEOUT ioctl call can be used to set data transfer timeout values for polled mode of operation. If the timeout value is so less that the timeout actually occurs, before completion of data transfer, the driver raises an error and updates the error status in the io-packet. However, after this the subsequent Stream calls on the driver are failing because the stream aborts with the bad mode. This ideally should not occur and seems like an issue with the Stream layer. This issue has been raised to the Stream/DSP BIOS™ team as an issue (SDOCM00054581) and is being analyzed.

  Workaround: Use sufficiently large timeout values that do not actually cause a timeout

8.1.3.6 McASP Device Driver (DSPBIOSTM)

- SDOCM00055111: After dynamic instantiation of the codec instance, the channel creation for receive and transmit on this dynamically instantiated instance fails. Note that this is observed only with dynamically instantiated instance of the codec. Channel creation on a statically instantiated codec instance works fine. This issue is being analyzed.

  Workaround: Use static instantiation for the codec instance

- SDOCM00055113: McASP transmit channel deletion fails. This is caused by under run error while deleting the channel. This issue is being analyzed.

  Workaround:None

8.2 Release 02.00.00.05

This was an internal release for DA830

8.2.1 New in the release

- This release contains the MMCSD, NAND and Block Media drivers in addition the drivers contained in the previous release.
- Implementation of platform initialization library, which contains the platform specific initialization functions separated from the sample application. Please refer to the User Guide for more details on the same.
- Implementation of IRs (Known issues and Bugs) raised on the previous release. Please refer to "Fixed in this release" section for a list of IRs fixed in this release

8.2.2 Fixed in this release

8.2.2.1 SPI Device Driver (DSP/BIOS™)

- SDOCM00053787: The SPI read/write test was failing on some sectors of the on board SPI flash.

  This issue has been fixed.

  Files modified: Spi.c

- SDOCM00053814: The functionality setting for SCS0 pin, as GPIO or CS, was not taking care of pin mode setting (3/4/5 pin mode).

  This issue has been fixed.

  Files modified: Spi.c

- SDOCM00052186: Use of GPIO pin as SPI Chip select is not tested.

  This feature has been tested.

  Files modified: Spi.c, Spi.xdc, Spi_edma.c

8.2.2.2 LCDC Raster Controller Device Driver (DSP/BIOS™)

- SDOCM00053819: The default image provided in the LCDC raster sample application needs to be replaced

  This issue was raised on the original quality of the default image supplied. Since, then the image has been replaced with an RGB stripe image.

  Files modified: rasterSample_image16bpp.h

- SDOCM00052578: The LCDC Raster driver does not return the proper sub-panel position, using Raster_IOCTL_GET_RASTER_SUBPANEL_CONF IOCTL, set previously by the Raster_IOCTL_SET_RASTER_SUBPANEL_POS IOCTL.

  This issue has been fixed to return the recently set value, by updating the subpanel information structure in the driver appropriately in the IOCTL.

  Files modified: Raster.c

- SDOCM00052575: The LCDC Raster driver does not return the proper sub-panel default data value, using Raster_IOCTL_GET_RASTER_SUBPANEL_CONF IOCTL, set previously by the Raster_IOCTL_SET_RASTER_SUBPANEL_DATA IOCTL.

  This issue has been fixed to return the recently set value, by updating the subpanel information structure in the driver appropriately in the IOCTL.

  Files modified: Raster.c

- SDOCM00052573: The LCDC Raster driver does not return the proper lines per pixel threshold value, using Raster_IOCTL_GET_RASTER_SUBPANEL_CONF IOCTL, set previously by the Raster_IOCTL_SET_RASTER_SUBPANEL_LPPT IOCTL.

  This issue has been fixed to return the recently set value, by updating the subpanel information structure in the driver appropriately in the IOCTL.

  Files modified: Raster.c

- SDOCM00052543: The image displayed by the LCDC Raster sample application contains artifacts.

  This issue has been fixed by correctly configuring the Pixel Clock polarity value, passed by the sample application via channel parameters

  File modified: rasterSample_main.c

- SDOCM00052487: Successive creation and deletion of raster channel fails.

  This issue has been fixed by appropriately initializing the instance state variable. This was applicable only for static instantiation.

  Files modified: Raster.xs

- SDOCM00052471: Dynamic allocation of frame buffers by the driver on application's behalf via Raster_IOCTL_ALLOC_FB, fails. The memory is not allocated for the frame buffers.

  This issue has been fixed by increasing the default heap instance size in the rasterSample.cfg. This is used for allocation of buffers.

  Files modified: rasterSample.cfg

8.2.2.3 GPIO Device Driver (DSP/BIOS™)

- SDOCM00051379: The PINMUX settings for GPIO0_8 and GPIO0_12, used in the GPIO sample application are wrong.

  This issue has been fixed. The correct values have been included in the platform library for GPIO.

  Files modified: gpioSample_main.c, gpioSample_io.c

- SDOCM00051167: The index used for referencing GPIO Pin configuration settings during initialization was wrong. It was referencing the index used for bank configuration.

  This issue has been fixed and the correct index is used.

  Files modified: Gpio.c

8.2.2.4 McASP Device Driver (DSP/BIOS™)

- SDOCM00054324: McASP doesn't initialize FIFOs correctly when operating with multiple serializers.

  This issue has been fixed and the FIFOs are initialized with correct values for multiple serializers

  Files modified: Mcasp.c

- SDOCM00054147: The multi-slot one serializer non-interleaved format is not supported by the driver

  This issue has been fixed and support of multi-slot one serializer non-interleaved has been added. Also, the support for more buffer formats has been added. Please refer to the user guide for more details.

  Files modified: Mcasp.c, Mcasp_edma.c, Mcasp.xdc

- SDOCM00054082: The calculation of length for default loop job buffer was wrong for multi-slot formats.

  This issue has been fixed.

  File modified: Mcasp.c, Mcasp_edma.c

- SDOCM00052890: The sample rate change IOCTL was not implemented for master mode of McASP.

  The support has been added.

File modified: Mcasp_ioctl.c

8.2.3 Known issues

8.2.3.1 MISRA C (All components)

- SDOCM00052184: MISRA C check was not run on the code base as we had issues in running MisraC tools on this package. (Tool had issues to scan the XDC header files and we are currently working on it).

  Workaround: None.

8.2.3.2 Instrumentation (all components)

- SDOCM00052185: Instrumentation code is not yet implemented as the XDC/CCSv4 version at the point of development did not support RTA and instrumentation APIs

  Workaround: None

8.2.3.3 LCDC Raster Device Driver (DSP BIOS™)

- SDOCM00052150: LCDC raster driver sample application abruptly hangs in debug mode. This issue is being investigated. However, this issue is not observed in release mode.

  Workaround: None

  SDOCM00053815: The sample image displayed by the raster driver sample application has slight flickering. This issue is being analyzed

  Workaround: None

8.2.3.4 *UART Device Driver (DSP BIOS™)*

- SDOCM00050476: It is observed that when the UART driver is configured to operate in polled mode and compiled in debug profile via XDC (command line compilation), the sample application hangs. This issue is being investigated. However, this issue is not observed in other modes and also with CCSv4 compilation.

  Workaround: None

8.2.3.5 *GPIO Device Driver (DSP BIOS™)*

- SDOCM00048745: It is observed that even if the GPIO bank pin interrupts are enabled for rising edge only, the interrupt from the pin occurs at both rising edge and falling edge of the incoming signal at the pin. This issue is being investigated. This issue is also observed with the RCSL example for GPIO.

  Workaround: None

8.2.3.6 *SPI Device Driver (DSP BIOS™)*

- SDOCM00048829: The slave mode of configuration of the device driver is yet to be tested

  Workaround: None

- SDOCM00054146: The SPI sample application fails at frequencies lower than 5MHz. This issue is being analyzed.

  Workaround: None

- SDOCM00051115: The SET_POLLEDMODETIMEOUT ioctl call can be used to set data transfer timeout values for polled mode of operation. If the timeout value is so less that the timeout actually occurs, before completion of data transfer, the driver raises an error and updates the error status in the io-packet. However, after this the subsequent Stream calls on the driver are failing because the stream aborts with the bad mode. This ideally should not occur and seems like an issue with the Stream layer. This issue has been raised to the Stream/DSP BIOS™ team as an issue (SDOCM00054581) and is being analyzed.

  Workaround: Use sufficiently large timeout values that do not actually cause a timeout.

8.2.3.7 *I2C Device Driver (DSP BIOS™)*

- SDOCM00048829: The slave mode of configuration of the device driver is yet to be tested

  Workaround: None

- SDOCM00051115: The SET_POLLEDMODETIMEOUT ioctl call can be used to set data transfer timeout values for polled mode of operation. If the timeout value is so less that the timeout actually occurs, before completion of data transfer, the driver raises an error and updates the error status in the io-packet. However, after this the subsequent Stream calls on the driver are failing because the stream aborts with the bad mode. This ideally should not occur and seems like an issue with the Stream layer. This issue has been raised to the Stream/DSP BIOS™ team as an issue (SDOCM00054581) and is being analyzed.

  Workaround: Use sufficiently large timeout values that do not actually cause a timeout

8.3 Release 02.00.00.04

This was the BETA release for DA830.

8.3.1 New in the release

- This release contains the LCDC Raster controller driver and the LCDC LIDD controller driver and their usage examples in addition to the previous release components.
- SPDIF mode support for McASP, in McASP driver, is implemented and tested in this release
- Usage of Hardware FIFO in McASP is implemented, in McASP driver, is implemented and tested in this release
- Implementation of IRs (Known issues and Bugs) raised on the previous release. Please refer to “Fixed in this release” section for a list of IRs fixed in this release

8.3.2 Fixed in this release

8.3.2.1 SPI (DSP/BIOS™) Device Driver

- SDOCM00048831 - The SPI driver need to have "use a GPIO" as chip select feature. This IR is fixed in this release. However, this feature could not be tested using current EVM

  Files Modified: Spi.c, Spi.xdc.

- SDOCM00049818 - SPI sample application stream creates fails in debug profile. This IR is fixed in this release.

  Files Modified: spiSample_io.c

- SDOCM00051114 - Spi_IOCTL_CANCEL_PENDING_IO ioctl returns failure. This IR is fixed in this release.

  Files Modified: Spi.c.

8.3.2.2 I2C (DSP/BIOS™) Device Driver

- SDOCM00049900 - I2C Set bit rate IOCTL commands fails. This IR is fixed in this release.

  Files Modified: I2c.c

- SDOCM00050946 - Driver hangs for I2C write to invalid slave address in Interrupt and DMA mode. This IR is fixed in this release.

  Files Modified: I2c.c.

- SDOCM00050948 – I2C Driver Deviation from RDD. This IR is fixed in this release

  Files Modified: I2c.c.

8.3.2.3 UART (DSP/BIOS™) Device Driver

- SDOCM00050475 - UART Baud rates greater than 115200 are not supported. This IR is fixed in this release.

Files Modified: none.

- SDOCM00050477 - UART Multiple submit calls (transmit) results in failure in dma mode. This IR is fixed in this release.

  Files Modified: Uart_edma.c, Uart.xdc.

- SDOCM00050835 - UART sample application fails in interrupt mode, when a file greater than 1000 bytes is given as input. This IR is fixed in this release.

  Files Modified: uartSample_io.c.

8.3.2.4 Audio (DSP/BIOS™) Device Driver

- SDOCM00049862 – "The audio interface driver does not implement some IOCTLS" is fixed. All the audio driver IOCTLs are implemented.

  Files modified Audio.c.

8.3.2.5 McASP (DSP/BIOS™) Device Driver

- SDOCM00050686 - "SPDIF mode is not tested for McASP driver". This feature is implemented in this release.

  Files modified Mcasp.c and Mcasp_edma.c.

- SDOCM00049904 – "McASP Mute on/off IOCTL commands fails" is fixed in this release.

  Files modified Mcasp_ioctl.c

- SDOCM00049851 – "Usage of hardware FIFO in Mcasp driver is not implemented". This feature is implemented in this release.

  Files modified mcasp.c, mcasp.xdc

- SDOCM00049850 – "Mcasp clock failure error handling is not implemented". This feature is implemented in this release.

  Files modified Mcasp.c

8.3.2.6 Aic3106 Codec driver (DSP/BIOS™)

- SDOCM00050490 – "The left channel of the HPOUT is muted for aic31 codec driver" is fixed in this release.

  Files modified Aic31.c, Aic31.xdc.

- SDOCM00049855 – "AIC31 driver does not support some IOCTLS" is fixed.

  Files Modified Aic31.c, Aic31.xdc,

- SDOCM00049853 – "Aic31 driver does not support more than one codec instance" is fixed in this release.

  Files modified Aic31.c, Aic31.xdc

8.3.3 Known Issues

8.3.3.1 MISRA C (All components)

- SDOCM00052184: MISRA C check was not run on the code base as we had issues in running MisraC tools on this package. (Tool had issues to scan the xdc based header files and we are currently working on it).

  Workaround: None.

8.3.3.2 Instrumentation (all components)

- SDOCM00052185: Instrumentation code is not yet implemented as the XDC/CCSv4 version at the point of development did not support RTA and instrumentation APIs

  Workaround: None

8.3.3.3 SPI (DSP/BIOS™)

- SDOCM00048829: Slave mode is not tested, due to absence of on-board SPI master. The plan is to test the slave mode using board to board communication in future releases.

  Workaround: None.

- SDOCM00052186: Use of GPIO pin as SPI Chip select is not tested.

  Workaround: None

8.3.3.4 I2C (DSP/BIOS™)

- SDOCM00048829: Slave mode is not tested, due to absence of on-board I2C master. The plan is to test the slave mode using board to board communication in future releases.

  Workaround: None.

8.3.3.5 UART (DSP/BIOS™)

- SDOCM00050476: UART driver fails during data transfer when built for Polled mode in debug profile using command line compilation.

  Workaround: None

- SDOCM00048830: UART Hardware flow control is not tested. This test is planned for future release.

  Workaround: None

8.3.3.6 McASP (DSP/BIOS™)

- SDOCM00052187:McASP is not tested in the master mode.

  Workaround: None.

- SDOCM00052285: The audio output contains noise at increased input (to the board) audio volume levels

  Workaround: Adjust input volume to the board to optimum levels

8.3.3.7 GPIO (DSP/BIOS™)

- SDOCM00048745 :The GPIO sample application configures the GPIO interrupt to be generated at rising edge. However the interrupt is generated at rising edge as well as falling edge. This issue is being analyzed.

  Workaround: None.

8.3.3.8 Audio Interface driver (DSP/BIOS™)

- SDOCM00052188: The IOCTL for the audio sample rate configuration is not working properly.

  Workaround: None

8.3.3.9 Aic3106 Codec driver (DSP/BIOS™)

- None

8.3.3.10 LCDC Raster Controller (DSP/BIOS™)

- SDOCM00052150: The LCDC Raster controller driver sample application hangs randomly when built in debug profile.

  Workaround: None

- SDOCM00052578: The LCDC Raster driver does not return the proper sub-panel position, using Raster_IOCTL_GET_RASTER_SUBPANEL_CONF IOCTL, set previously by the Raster_IOCTL_SET_RASTER_SUBPANEL_POS IOCTL.

  Workaround: None

- SDOCM00052575: The LCDC Raster driver does not return the proper sub-panel default data value, using Raster_IOCTL_GET_RASTER_SUBPANEL_CONF IOCTL, set previously by the Raster_IOCTL_SET_RASTER_SUBPANEL_DATA IOCTL.

  Workaround: None

- SDOCM00052573: The LCDC Raster driver does not return the proper lines per pixel threshold value, using Raster_IOCTL_GET_RASTER_SUBPANEL_CONF IOCTL, set previously by the Raster_IOCTL_SET_RASTER_SUBPANEL_LPPT IOCTL.

  Workaround: None

- SDOCM00052543: The image displayed by the LCDC Raster sample application contains artifacts. The issue is being analysed.

  Workaround: None

- SDOCM00052487: Successive creation and deletion of raster channel fails.

  Workaround: None

- SDOCM00052471: Dynamic allocation of frame buffers by the driver on application's behalf via Raster_IOCTL_ALLOC_FB, fails. The memory is not allocated. For the frame buffers is not allocated. The issue is being analyzed.

  Workaround: Use statically created buffers (frame buffer arrays)

8.4 Release 02.00.00.03

This was the EA2 release for DA830.

8.4.1 New in the release

- Drivers for UART, SPI, I2C, McASP, Audio Interface, Codec, GPIO and PSC.

8.4.2 Fixed in this release

- None

8.4.3 Known Issues

8.4.3.1 MISRA C (All components)

- MISRA C check was not run on the code base as we had issues in running MisraC tools on this package. Tool had issues to scan the xdc based header files and we are currently working on it). Workaround: None.

8.4.3.2 Instrumentation (all components)

- Instrumentation code is not yet implemented as the XDC/CCSv4 version at the point of development did not support RTA and instrumentation APIs

8.4.3.3 SPI (DSP/BIOS™)

- In SPI sample application “stream_create” fails in debug profile of the application. We are analyzing this issue. Workaround: To have the application profile as “whole_program_debug”.
- Slave mode is not tested, due to absence of on-board SPI master. The plan is to test the slave mode using board to board communication in future releases. Workaround: None.
- SPI driver fails in interrupt mode and configured for loopback (DLB). Workaround: None.
- SPI data transfer works for 8 and 16 bits character length only. Workaround: None.

8.4.3.4 I2C (DSP/BIOS™)

- Slave mode is not tested, due to absence of on-board I2C master. The plan is to test the slave mode using board to board communication in future releases. Workaround: None.
- IOCTL command for setting the bit rate for data transfer fails. Workaround: None.

8.4.3.5 UART (DSP/BIOS™)

- UART driver fails in EDMA mode and configured for loopback (DLB). Workaround: None.
- SDOCM00050477. Multiple read/write fails in the UART sample applciation, when the driver is configured in EDMA mode. Workaround: None. This will be fixed in the next release
- SDOCM00050476. The release mode executable for UART sample application in this release package fails. Workaround: None

8.4.3.6 McASP (DSP/BIOS™)

- McASP FIFO support is not yet implemented and is planned for future releases.

  Workaround: None.

- SPDIF mode is not tested and planned for future release.

  Workaround: None.

- McASP clock failure error handling is not yet implemented.

  Workaround: None.

- IOCTL command for MUTE on/off is not working yet. This is being analyzed. Workaround: None

8.4.3.7 GPIO (DSP/BIOS™)

- The GPIO sample application configures the GPIO interrupt to be generated at rising edge. However the interrupt is generated at rising edge as well as falling edge. This issue is being analyzed.

  Workaround: None.

8.4.3.8 Audio Interface driver (DSP/BIOS™)

IOCTLs for the following are not yet implemented.

- Configuring the receive channel sample rate for a given Audio configuration
- Configuring the transmit channel sample rate for a given Audio configuration

8.4.3.9 Aic3106 Codec driver (DSP/BIOS™)

IOCTLs for the following are not yet implemented.

- Select the input Audio mode Line-In or Mic-In.
- Select the output Audio mode Line-Out or Speaker-Out
- Increase or Decrease the input volume

8.5 Release 02.00.00.02

- This was the EA1 release for DA830 that contained the rCSLs and their examples.