

A DSP/BIOS AD50 Codec Device Driver for the TMS320C5402 DSK

Software Development Systems

ABSTRACT

This document describes the implementation of a DSP/BIOS™ device driver for the TLC320AD50 codec on the TMS320C5402 DSK. This device driver is written in conformance to the DSP/BIOS IOM device driver model. The main functionality of this device driver is to set up the codec. The actual data samples transformation is performed by the generic C5402_DMA_MCBSP part of the driver. This device driver is an interface for the core generic C5402_DMA_MCBSP driver.

Contents

1	Usage	2
1.1	Configuration	3
1.2	Device Parameters	4
1.3	Channel Parameters	4
1.4	Control Commands	4
2	Architecture	4
3	Constraints	5
4	References	5
Appendix A	Device Driver Data Sheet	6
A.1	Device Driver Library Name	6
A.2	DSP/BIOS Modules Used	6
A.3	DSP/BIOS Objects Used	6
A.4	CSL Modules Used	6
A.5	CPU Interrupts Used	6
A.6	Peripherals Used	6
A.7	Interrupt Disable Time	6
A.8	Memory Usage	6

List of Figures

Figure 1	DSP/BIOS IOM Device Driver Model	2
Figure 2	Codec Device Driver Partitioning	3

List of Tables

Table A–1	Device Driver Memory Usage	6
-----------	----------------------------------	---

Trademarks are the property of their respective owners.

1 Usage

The device driver described here is actually part of an IOM mini-driver. That is, it is implemented as the lower layer of a 2-layer device driver model. The upper layer is the class driver, and can be the DSP/BIOS GIO, SIO/DIO, or PIP/PIO modules, or even some other class driver. The upper layer provides an independent and generic set of APIs and services for a wide variety of mini-drivers and allows the application to use a common interface for I/O requests. Figure 1 shows the overall DSP/BIOS device driver architecture. For more information about the IOM device driver model as well as the GIO, SIO/DIO, and PIP/PIO modules, see the References section.

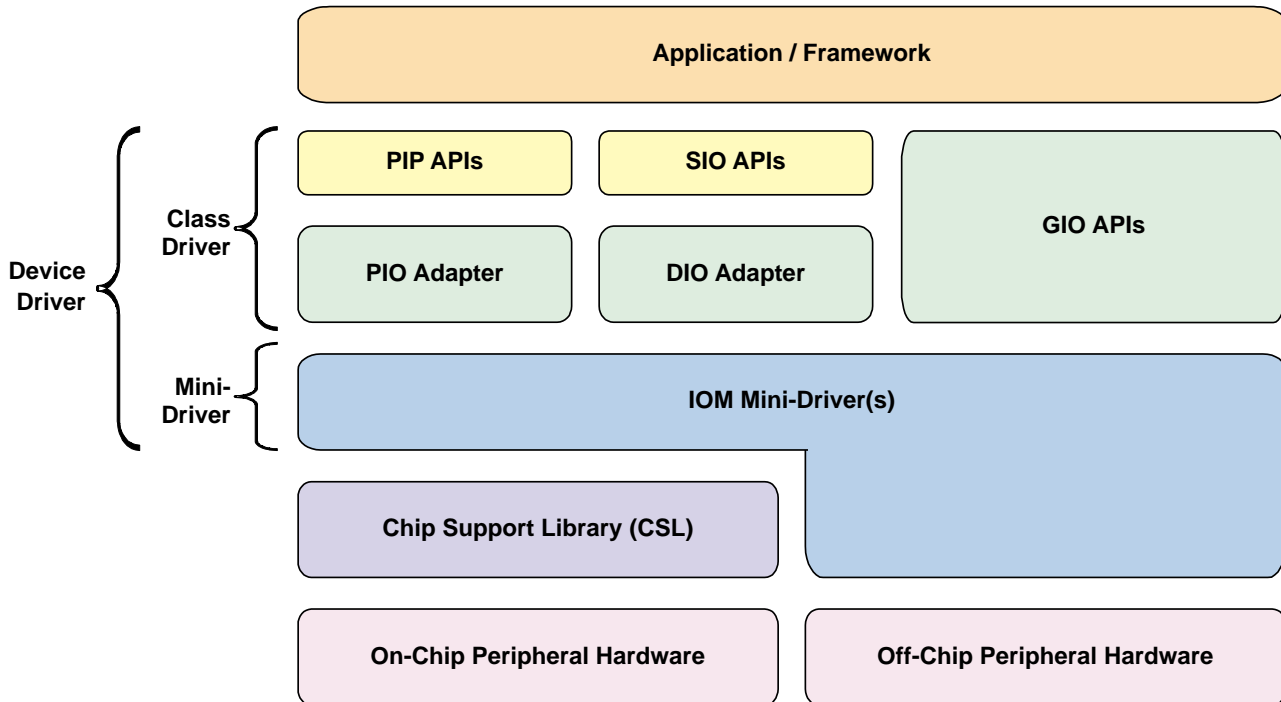


Figure 1. DSP/BIOS IOM Device Driver Model

As shown in Figure 2, mini-drivers are often split into a generic part and a codec sub-driver in order to maximize code reusability. This application note only describes the codec-specific part of the mini-driver, which uses the generic C5402_DMA_MCBSP driver to transfer data to and from the serial ports. Because this codec-specific part is dependent on the generic C5402_DMA_MCBSP part, an application must link both of the libraries in order for it to function correctly. These two libraries are called `dsk5402_dma_ad50.i54` and `c5402_dma_mcbsp.i54`.

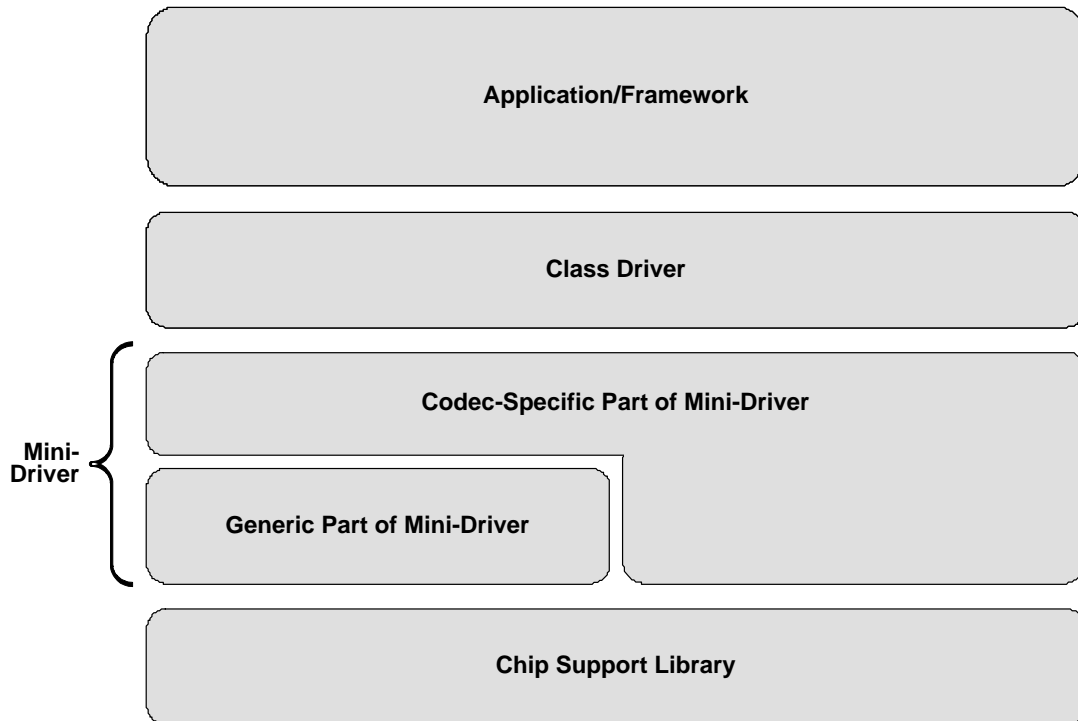


Figure 2. Codec Device Driver Partitioning

1.1 Configuration

To use this device driver, a device entry has to be added and configured in the DSP/BIOS configuration tool. Refer to the DSP/BIOS Device Driver Developer's Guide (SPRU616) for more information on how to use the DSP/BIOS Configuration tool to configure device drivers. This device driver will set up the generic TMS320C5402_DMA_MCBSP driver to meet its needs. The following are the device configuration settings required to use this driver:

- **Init function:** `_DSK5402_DMA_AD50_init`.
- **Function table ptr:** `_DSK5402_DMA_AD50_Fxns`.
- **Function table type:** `IOM_Fxns`.
- **Device id:** This property is ignored by this device driver since there is only one TLC320AD50 codec on the TMS320C5402 DSK and it is hard-coded within the driver.
- **Device params ptr:** An optional pointer to an object of type `DSK5402_DMA_AD50_DevParams`, as defined in the header file `dsk5402_dma_ad50.h`. This pointer will point to a device parameter structure. Set this pointer to `NULL` to use the default parameters. The parameter structure and defaults are described below.
- **Device global data ptr:** N/A, not used by this driver.

1.2 Device Parameters

```

Uns          rxIntrMask;      /* receiver interrupt mask */
Uns          txIntrMask;      /* transmitter interrupt mask */

typedef struct DSK5402_DMA_AD50_DevParams {
    Int versionId;
    Uns rxDmaId;
    Uns txDmaId;
    AD50_Params ad50;
    Uns rxIntrMask;

    Uns txIntrMask;
} DSK5402_DMA_AD50_DevParams;

```

- **versionId:** Version number of the driver.
- **rxDmaId:** Receive DMA channel.
- **txDmaId:** Transmit DMA channel.
- **ad50:** The codec registers setup. If the device params pointer is NULL, the default parameters are used. Here are the default setups for the registers.
 - **Register 1:** Default value is 0x80, which is software reset.
 - **Register 2:** Default value is 0x00.
 - **Register 3:** Default value is 0x00.
 - **Register 4:** This register controls the input gain, output gain, and frequency. Default value is 0xA0, which controls bypass internal DPLL.
- **rxIntrMask:** Receive interrupt mask, set in the ISR.
- **txIntrMask:** Transmit interrupt mask, set in the ISR.

1.3 Channel Parameters

This device driver does not have any channel parameters. Any values that are passed as channel parameters will be ignored. Specifying NULL is suggested.

1.4 Control Commands

This device driver has no run-time control commands.

2 Architecture

This codec-specific portion of the mini-driver is a layer to the generic C5402_DMA_MCBSP part that transfers the data samples, opens the channels, and so on. The codec-specific part uses two basic functions, `mdBindDev()` and `mdCreateChan()`, to bind with the functions from the generic part. The `mdBindDev()` function uses the McBSP port 1 to set up the codec. After setting up the codec, the device driver closes the McBSP port 1 before calling the generic C5402_DMA_MCBSP `mdBindDev()` function to allow this port to be used for data transfer. The `mdCreateChan()` function specifies the mode of the channel (i.e., input or output) to be opened and then calls the generic C5402_DMA_MCBSP `mdCreateChan()` function to carry out the work of creating the channel.

It is important to note that every sample sent to the codec that has the LSB set will be interpreted as a command. This device driver does not strip the LSB when sending the samples to the codec, so the application layer has to strip the LSB from every sample it sent to the codec device driver.

3 Constraints

- Inherits the constraints of the generic C5402_DMA_MCBSP part of the mini-driver.
- Application must strip the least significant bit (LSB) from the data.

4 References

All these documents are available at the TI Developer's Village.

1. *TMS320C54x Chip Support Library API User's Guide* (SPRU420).
2. *TMS320C5000 DSP/BIOS Application Programming Interface (API) Reference Guide* (SPRU404C).
3. *DSP/BIOS Device Driver Developer's Guide* (SPRU616).
4. *TLC320AD50 C/I Sigma-Delta Analog Interface Circuits with Master-Slave Function Data Manual*, SLAS131E.
5. *A DSP/BIOS Generic DMA McBSP Device Driver for TMS320C5000 DSPs* (SPRA858).

Appendix A Device Driver Data Sheet

A.1 Device Driver Library Name

dsk5402_dma_ad50.l54

c5402_dma_mcbasp.l54 is required for building application.

A.2 DSP/BIOS Modules Used

Refer to the generic C5402_DMA_MCBSP documentation.

A.3 DSP/BIOS Objects Used

Refer to the generic C5402_DMA_MCBSP documentation.

A.4 CSL Modules Used

Refer to the generic C5402_DMA_MCBSP documentation.

A.5 CPU Interrupts Used

Refer to the generic C5402_DMA_MCBSP documentation.

A.6 Peripherals Used

McBSP port 1 is used to set up the AD50 codec and data control.

A.7 Interrupt Disable Time

Maximum time that hardware interrupts can be disabled by the driver: refer to the generic C5402_DMA_MCBSP documentation. This measurement is taken using the compiler option `-O3`.

A.8 Memory Usage

Table A–1. Device Driver Memory Usage

	Uninitialized memory	Initialized memory
CODE	—	370 (16-bit words)
DATA	33 (16-bit words)	39 (16-bit words)

NOTE: This data was gathered using the `sectti` command utility.

Uninitialized data: `.bss`

Initialized data: `.cinit + .const`

Initialized code: `.text + .text:init`

The sizes in this table were measured for the near model on the C5402 DSK platform.

Note that to calculate the total driver size overhead, you must include the memory usage of the generic C5402_DMA_MCBSP part of the mini-driver as well. These size overheads are listed in the generic part's application note, which is listed in the References section.

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.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2003, Texas Instruments Incorporated