D2AUDIO APPLICATION NOTE AN11

Using The X-Series Module Control Interface

1 Introduction

The self-contained audio power amplifer modules from D2Audio™, with their high-efficiency and high-performance, provide the perfect solution for today's higher perfomance audio amplifiers, receivers, and powered speakers. These modules provide the customer with a drop-in class-D amplifier solution. Their flexibility and powerful signal-processing features can only be harnessed by utilizing the module control interface. This application note describes the Module Control Interface (MCI) on the X-series amplifier modules and discusses how to use it.

2 Interface Description

The MCI is a two-wire serial interface used to access the advanced features of the X-series modules. These features include volume control, tone control, equalizers, compressor-limiters, and other advanced signal-processing algorithms. 1 to 4 channel modules will have a single interface while 5 to 8 channel modules offer 2 interfaces. Each MCI provides access to internal registers that configure the signal-processing features.

The MCI uses a master-slave protocol. All X-series module control interfaces behave as slave devices and require an external microcontroller to act as the master device.

A transaction can be broken into 4 distinct phases: Start condition, Device Address and Read/Write (R/W) bit, Data transfer, and Stop condition. Bytes are transmitted most significant bit (MSB) first and least significant bit (LSB) last. With each byte transmitted, the receiver performs a 1 bit acknowledge (ACK) identifier to confirm that the transfer occurred correctly. A transaction is started with a Start condition and ended with a Stop condition. Start and Stop conditions are only issued by the master device. Slave devices, like the X-series amplifier module, cannot issue Start and Stop conditions. At certain times, it may be more efficient to use a Repeat Start condition. Repeat Start conditions are special-case conditions which occur when a Stop condition would have immediately been followed by a Start condition. Repeat Start conditions are more efficient than using separate Stop and Start conditions due to the fact that they drop the Stop condition, allowing higher throughput.

The Device Address is an 8-bit value that allows the master to uniquely identify the target MCI in a control transaction. The Device Address least significant bit is the read/write bit and determines whether a read or write transaction is occuring. The Device Address assigned to a MCI is a function of the module and its configuration and is specified in datasheets with the least significant bit set to 0. Please refer to the associated amplifier module datasheet to determine the MCI Device Address.

3 MCI Hardware

The MCI is accessable by the SCL and SDA pins on 1 to 4 channel modules and the SCL[1:0] and SDA[1:0] pins on 5 to 8 channel modules. The MCI is an open-drain design with the SCL and SDA pins having 10k ohm pull-up resistors to 3.3v internal to the module. The module drives the data line with an open-drain driver and it is expected that external master devices will use open drain-drivers for both SCL and SDA pins. The X-series MCI internal block diagram is shown in Figure 1.

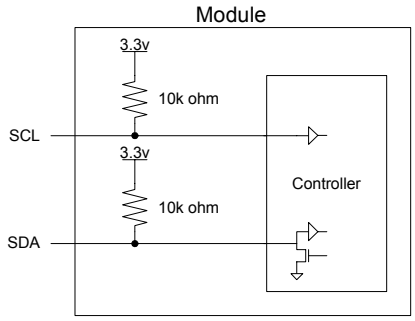

FIGURE 1: MCI Internal Structure

It is anticipated that an external microcontroller that is acting as the master device will connect directly to the SCL and SDA pins on the X-series module. This is shown in Figure 2.

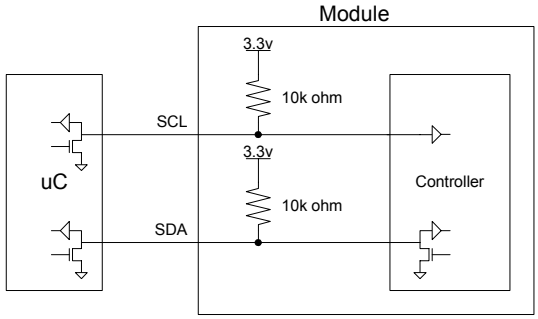

FIGURE 2: Connecting MCI

If two control interfaces are present on a module, it is permissable to connect the multiple SCL pins together as well as the multiple SDA pins together for the purposes of connecting to a single microcontroller as shown in Figure 3.

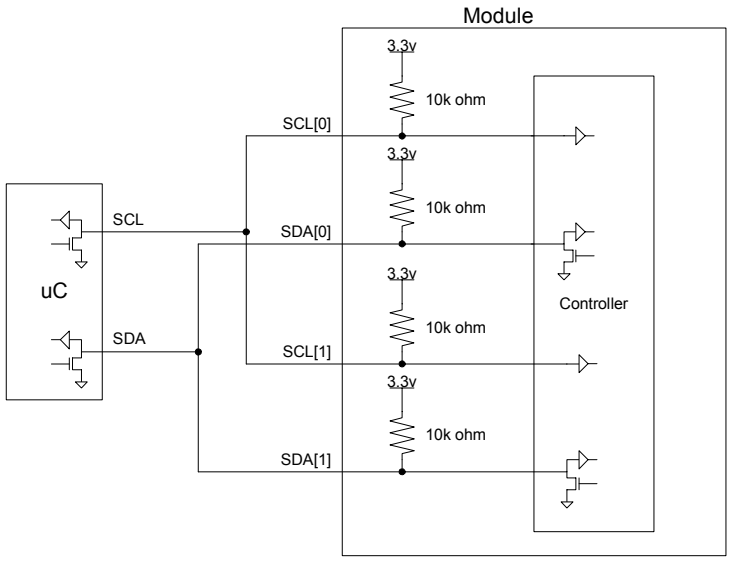

FIGURE 3: Connecting Multiple MCI

Care should be taken to verify that the external microcontroller can properly sink the current from the paralleled pull-up resistors when multiple MCIs are connected together. The microcontroller drivers should be able to sink 1mA per attached MCI.

4 Electrical Waveforms

To better enable debugging and understanding of the module control interface, it is important to understand the electrical signals that are transmitted on the SCL and SDA wires during transactions. A Start Condition is identified by the SDA transitioning from high-to-low while SCL is high. A Stop Condition is identified by by SDA transitioning from low-to-high while SCL is high. For all other data communications including device address, ACKs, and payload data, SDA

transitions while SCL is low, and data is latched on the SCL rising edge. Repeat Start conditions are electrically identical to Start conditions.

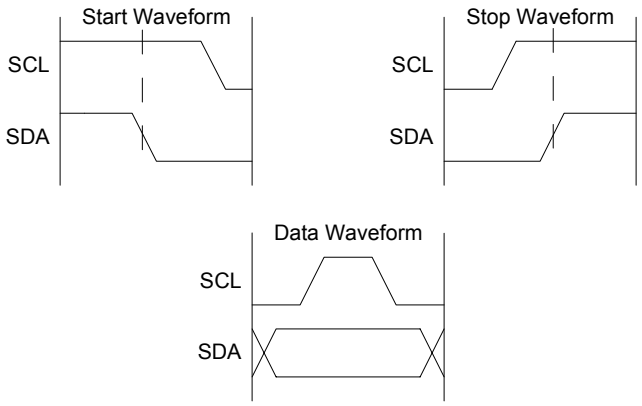

FIGURE 4: MCI Waveforms

The clock signal is always the responsibility of the master. This is the pacing signal of the interface. The interface should be limited to bit-rates less than 100kbps. Operation at speeds

greater than this are not guaranteed. The electrical waveform timings are shown in Figure 5.

TABLE 1: MCI Waveform Timings

Symbol	Description	Min	Max	Unit
fSCL	SCL frequency		100	kHz
t_{buf}	Bus free time between transmissions	4.7		us
t_wSCL	SCL clock low	4.7		us
t_wSCL	SCL clock high	4.0		us
t_sSTA	Setup time for a (repeated) Start	4.7		us
t_hSTA	Start condition Hold time	4.0		us
t_hSDA	SDA hold from SCL falling	0		us
t_sSDA	SDA setup time to SCL rising	250		ns
t_dSDA	SDA output delay time from SCL falling		3.5	us
t_r	Rise time of both SDA and SCL		1	us
t_f	Fall time of both SDA and SCL		300	ns
t_sSTO	Setup time for a Stop condition	4.7		us

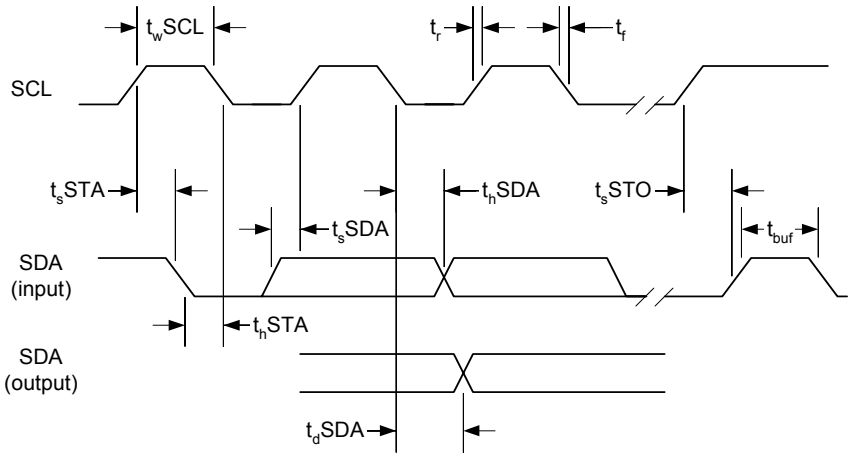

FIGURE 5: MCI Waveform Timing Diagram

5 Writing Control Registers

All write operations to an X-series amplifier module control register must begin with the Start condition, followed by the Device Address byte (with read/write bit set low), 3 Register Address bytes, 3 Data bytes and a Stop condition. The amplifier module slave controller acknowledges each byte by pulling SDA low on the bit immediately following each byte. The Register Address

bytes uniquely define the control register within the module that is targeted by the transaction. The Data bytes are written to the internal register by the transaction. Register Addresses and valid Data values are described in the module's datasheet. The operation is shown in Table 2.

TABLE 2: Controller Register-Write Byte-Sequence Description

Byte	Read/Write	Name	Description
Start Condition			
0	W	Device Address + Read/ Write Bit	Device Address of channel group, with R/W bit cleared
1	W	Register Address [23:16]	Upper 8 bits of register address
2	W	Register Address [15:8]	Middle 8 bits of register address
3	W	Register Address [7:0]	Lower 8 bits of register address
4	W	Data[23:16]	Upper 8 bits of register data to write
5	W	Data[15:8]	Middle 8 bits of register data to write
6	W	Data[7:0]	Lower 8 bits of register data to write
Stop Condition			

A sequence diagram for the write operation is shown in Figure 6. This diagram helps illustrate the relationship between ACKs and bytes of data.

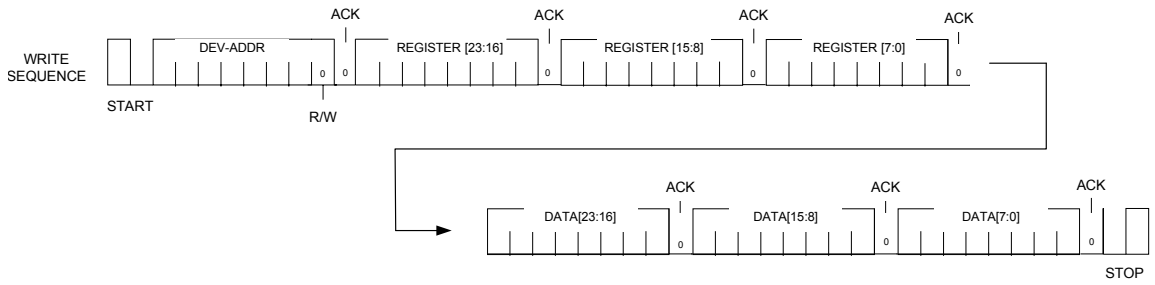

FIGURE 6: Write Command Sequence Diagram

6 Reading Control Registers

All read operations from the X-series amplifier module controller registers require two transactions. During the first transaction, the master must send the Start condition, followed by the Device Address byte (with read/write bit cleared), and 3 Register Address bytes. The amplifier module slave controller acknowledges each byte in the first step by pulling SDA low on the bit immediately following each byte.

Immediately following the first transaction, the master must send a Repeat Start condition, followed by the Device Address byte (with read/write bit set). The amplifier module slave controller will acknowledge the Device Address byte by pulling SDA low on the bit immediately following the Device Address. The next 3 bytes are sent by the amplifier module slave controller

to the master, and contain the three Data bytes to be read. The master must acknowledge the first two Data bytes by pulling SDA low on the bit immediately following the Device Address. After the third Data byte, the master should send a negative acknowledge (NACK). At the end of the second transaction, the master must send the Stop condition. It should be noted that while a Repeat Start condition is used to signify the start of the second transaction, it is valid to perform a Stop condition followed by a Start condition.

TABLE 3: Controller Register-Read Byte-Sequence Description

Step	Byte	Read/Write	Name	Description
1	Start Condition			
	0	W	Device Address + Read/ Write Bit	Device Address of channel group, with R/W bit cleared
	1	W	Register Address [23:16]	Upper 8 bits of register address
	2	W	Register Address [15:8]	Middle 8 bits of register address
	3	W	Register Address [7:0]	Lower 8 bits of register address
2	Repeat Start Condition			
	4	W	Device Address + Read/ Write Bit	Device Address of channel group, with R/W bit set
	5	R	Data[23:16]	Upper 8 bits of register data to read
	6	R	Data[15:8]	Middle 8 bits of register data to read
	7	R	Data[7:0]	Lower 8 bits of register data to read
	Stop Condition			

A sequence diagram for the control write sequence is shown in Figure 7. This diagram helps illustrate the relationship between ACKs and bytes of data.

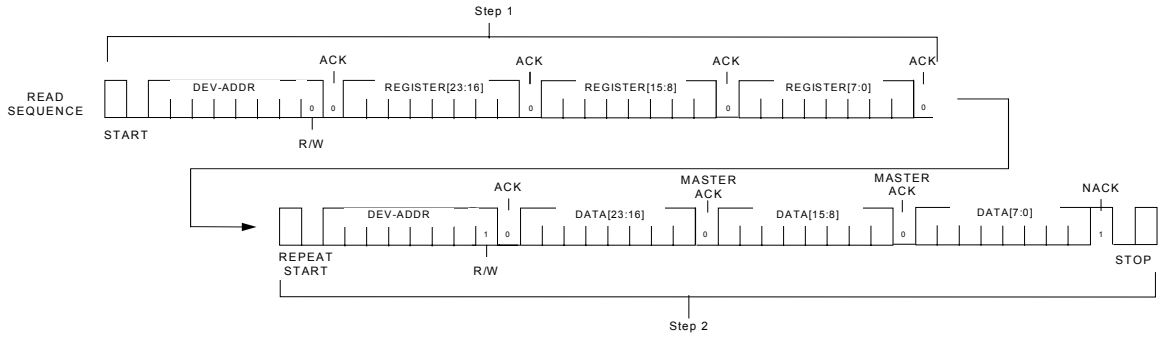

FIGURE 7: Read Command Sequence Diagram

7 Communication Errors

When using the Module Control Interface, there is the potential for a communication error. The most obvious indication that an error occurred is that the ACK bit is not low. One can also verify a control register write by rereading the control register and verifying it against the expected value.

If an error occurs, it is suggested to reattempt the failed operation. If the error continues to occur, there are a number of possible suggestions to aid in the debugging process.

1. Verify that the amplifier module is generating an ACK after the Device Address and Read/Write Bit. If this does not occur, the Device Address may be incorrect. Keep in mind that there is more than one Device Address on modules with 6, 7, or 8 channels.

2. Verify that each byte is being sent most significant bit (MSB) first. Data will be corrupted if this is not occuring.

3. Verify that the SCL clock frequency has not exceeded 100 kbps. If this has occurred, the master device is running faster than specified.

4. Verify the waveform timings shown in Table 1 are being observed. Data corruption may occur if the timings are not being observed.

8 Example: Changing Master Volume

A typical usage for the Module Control Interface includes adjusting the volume. The following steps illustrate how to set the amplifier module's master volume to maximum.

1. Determine Device Address

Please refer to a specific amplifier module's datasheet to determine the device address. For the purposes of this example, 0xB2 will be used as the Device Address.

2. Determine Register Address

The master volume is accessed through the MasterVolumeParameter register. For illustrative purposes, 0x123456 will be used as the MasterVolumeParameter address in this example. Please refer to a specific amplifier module's datasheet to determine the correct register address for the MasterVolumeParameter.

3. Determine Value To Write

Each parameter that can be accessed via the MCI has an associated description in the module documentation. This description includes the register address and a discussion of how the parameter should be adjusted. Equations and safe limits are given where appropriate. For the MasterVolumeParameter, the maximum value is 0x7FFFFF.

4. Perform Write Transaction

A write operation will be performed to device address 0xB2 that will set internal register 0x123456 to 0x7FFFFF. The resulting write transaction is shown in Figure 8. This write transaction should now have set the internal register 0x123456 to 0x7FFFFF.

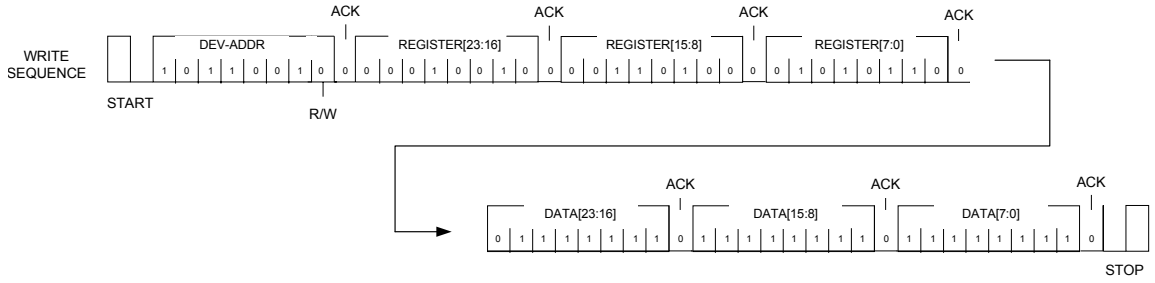

FIGURE 8: Master Volume Change Write Sequence Diagram

5. Verify Register Write

A read operation can be performed to verify that the master volume was properly changed. Please note that the R/W bit is set the second time the device address is sent. The resulting read transaction is shown in Figure 9. If the read transaction does not return the anticipated 0x7FFFFF, there was an error.

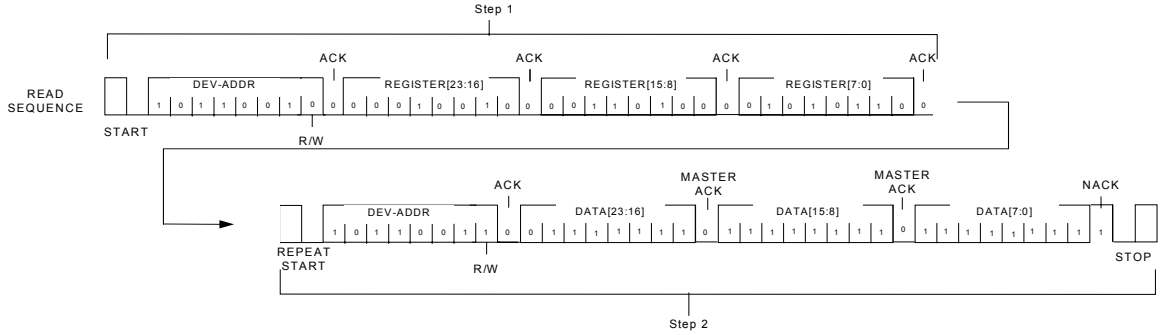

FIGURE 9: Master Volume Change Read Sequence Diagram

Trademarks

D2Audio, D2, the D2Audio logo, D2A, and Audio Canvas are trademarks of D2Audio Corporation. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies

Nuclear and Medical Applications

D2Audio products are not authorized for use as critical components in life support systems, equipment used in hazardous environments or nuclear control systems without the express written consent of D2Audio Corporation.