AB001 (AD2428) AND AB0331 (AD2437) A²B MODULE AND EVM KIT USER MANUAL

Rev 1

18-Feb-24

DRAFT

Signal Processing

http://clk.works/

Products covered:

AB0001 AD2428 based A²B to I²S module

AB0331 AD2437 based A²B to I²S module

AB0020 EVM for A²B module

1. TABLE OF CONTENTS

1 INTRODUCTION

The Clockworks A²B modules are intended to allow developers to quickly integrate Analog Device's A²B capability in to a prototyping environment. For OEMs it provides a quick time to market solution without need to design and debug the A²B portion.

The modules can operate as either an A²B root (first node) or client (peripheral downstream nodes).

The AB0001 is intended to be locally powered, i.e. it does not support drawing power from an A²B bus with phantom power. For AD2428 applications the AB0003 module can be used for phantom powered systems. The module does provide power to downstream A²B slave nodes using A²B's CFG0 phantom power feature.

For AD2437 systems with the AB0331 module the phantom power answer is a bit more complex. A modified version of the AB0331 can be used with the appropriate circuitry on the carrier to create a phantom powered node. Please contact Clockworks for more information about that configuration.

The AB0001 module is a small 2 layer PCB with two 12 pin 2.54 mm (0.1") connectors on the bottom for easy mating to other hardware. The tan connectors indicate CFG0 (up to 9V) phantom power can be used.

Figure 1 AB0001 AD2428 based A²B module

The AB0003 module is a small 4 layer PCB with 16 pin and 24 pin 2.54 mm (0.1") connectors on the bottom for easy mating to other hardware. The red connectors indicate CFG4 (up to 24V) phantom power can be used.

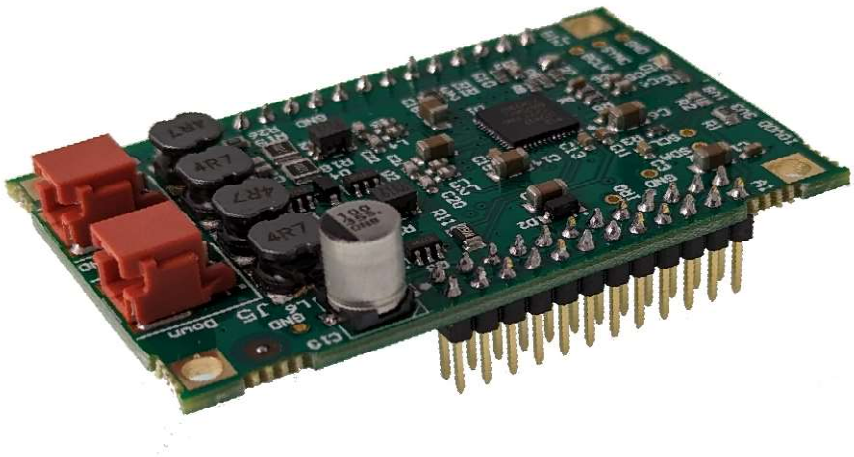


Figure 2 AB0331 AD2437 based A²B module

1.1 MODULE EVM

For system developers there is a need to have A²B devices that can stand in for other hardware, or to monitor audio streams that are being sent between other devices in the system. The EVM also provides a known good reference for operating the module and a design that can be copied if needed.

The EVM uses the Analog Device's (ADI) ADAU1761 Sigma DSP so that the EVM can perform local processing on the audio input and output, which is a common practice in many A²B applications.

For root node/system development an Analog Devices USBi (or equivalent) will be needed to use the SigmaStudio+ A²B development environment[1].

1.2 SOFTWARE SUPPORT

At present all A²B software is supplied by ADI directly. Normally this software package includes both an add on for SigmaStudio+ that allows an A²B network to be described graphically, and a library with a standard API that can be used by the (host) processor that is connected to the A²B root (first) device in an actual system.

The A²B software API information can be found on ADI's website and Wiki pages.

Functionally the Clockworks module and EVM combination in slave mode operate the same as the ADI WBZ eval board, except that the Clockworks system is locally powered.[2]

When using ADI's tools for A²B configuration and operation remember to set the A²B device type to match the node type(s) being used in the system.

Please see section 6 for an example with the A²B add-on for Sigma Studio.

2 GETTING STARTED WITH THE EVM

[1] SigmaStudio 4.7 can be used for AB0001 AD2428 systems. Software development for the ADAU1761 Sigma DSP is not supported in SigmaStudio+ and the SigmaStudio 4.7 must be used.
[2] The Clockworks EVM setup matches the original ADI 2425WBZ design. The newer ADI 2428WBZ design adds two microphones and has a few other minor I/O changes. From the perspective of configuring the hardware in the ADI SigmaStudio+ for A²B tool they are all very similar.

The setup steps are similar between the two version (root or client) of the EVM board.

The EVM board and the modules are ESD sensitive and must be handled following good ESD practices with a proper and safe grounding of the work area and anyone handling the boards.

The EVM kit ships with a DC power supply that has universal input. The North America version uses the two blade plug, whereas the international version of the kit has multiple adapters that mate with the supply.

The specific supply in an EVM kit depends on the module choice. For the AB0001 AD2428 module, which provides CFG0 phantom power, a 12V 6W supply is used; the board locally regulates that down to the 9V CFG0 standard. For AD2437 systems that use CFG4 power, a 24V 60W supply is used.

If using the international version of the kit please attach the correct AC mains adapter to the supply.

The AC adapter provides power via a standard (center positive) via 2.1mm barrel connector. Alternately the board can be powered from a bench supply providing 12 – 24V DC by connecting wires to the Phoenix block connector.

2.1 MIXED CFG0 AND CFG4 SYSREMS

Applying 24V CFG4 power to a AD2428 CFG0 system would be a very bad thing to do.

Please contact Clockworks for options to isolate phantom power between CFG0 and CFG4 systems.

2.2 CONNECTION STEPS- CLIENT NODE EVM CONFIGUATION

The EVM kit is shipped with the A²B module installed. The A²B module is configured for IO_{Vdd} of 3.3V

- Set SW1 (power) to the off position (slide button at board edge).
- Attach the 12-24 VDC supply to input jack J1 and plug the supply in to the AC mains.
- Connect an audio source to J8 (Line in) if planning on using the EVM for stereo audio acquisition
- Connect headphones or amplified speakers to the Headphone out jack J4. Alternately amplified speakers or other line level device can be connected to J5 (Line out) if you change the default software for the ADAU1761.

- Connect an A²B cable to lower module jack ("From upstream") to the A²B root or A²B client located electrically closer to the A²B root.
- Set the MCLK source jumper to the right side to select the A²B module as the MCLK source.

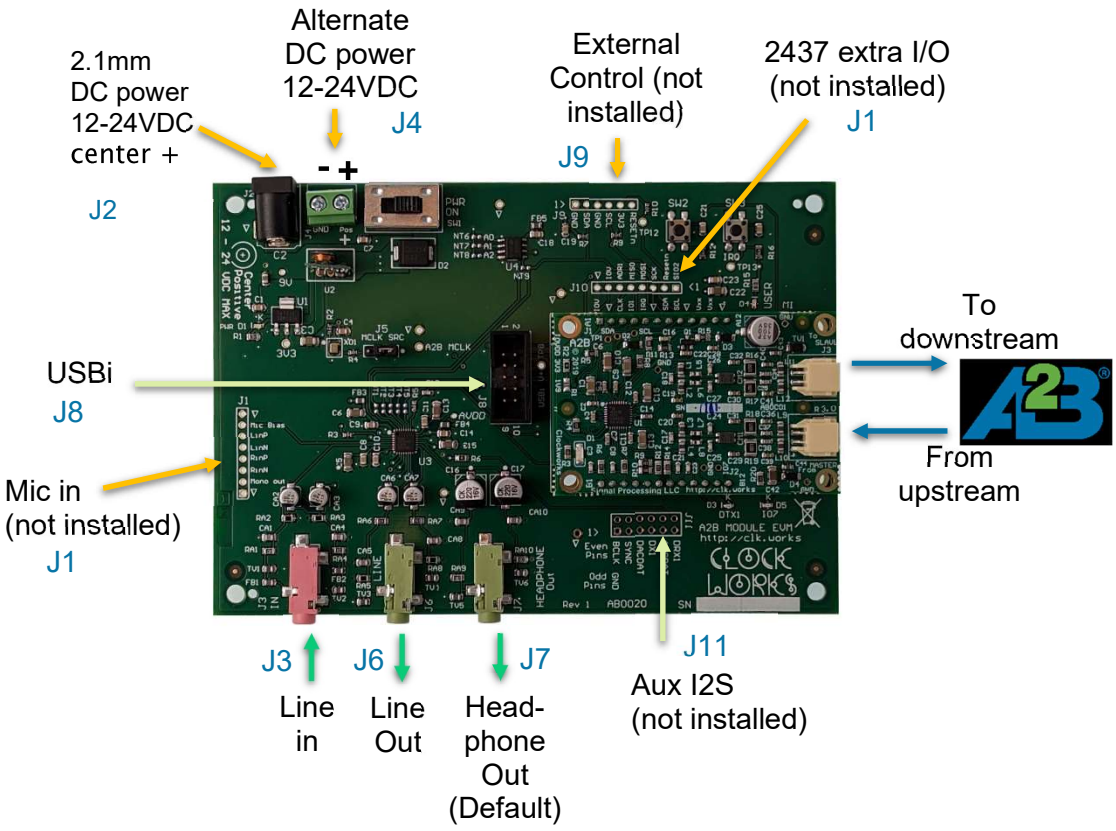


Figure 3 EVM external connectors

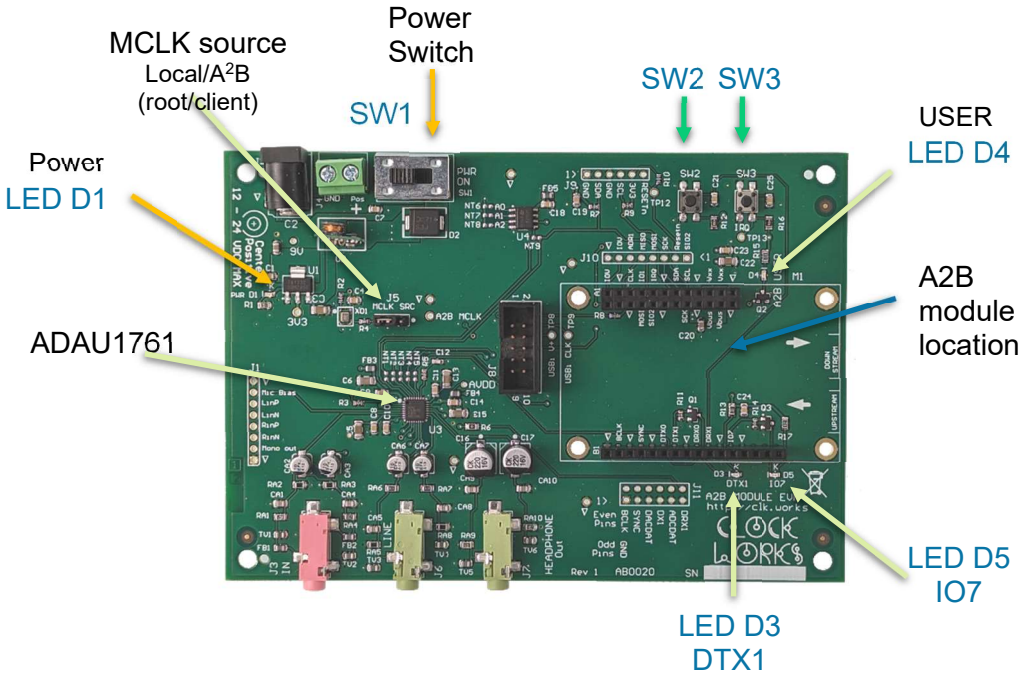


Figure 4 EVM Components

The EVM can be powered up by switching SW1 to the "ON" position. The power indicator on the EVM and on the A²B module should be light.

Once the A²B host is operational and the A²B network established – which with the provided software examples includes programming the ADAU1761, the User LED (D4) will light.

If the user LED fails to light or the board is not operational for passing audio check that the *Download on initialization* check box is enabled if using the SigmaStudio A²B tools. If developing applications programmatically verify the ADAU1761 code is being correctly processed and downloaded.

By default that EEPROM is not programmed. Some A²B software examples or AD2437 systems with CFG4 power may expect certain contents in the EEPROM.

2.3 CONNECTION STEPS – ROOT EVM CONFIGURATION

The steps are basically the same as the hardware is almost the same.

- Set the MCLK source jumper to the left side to select the EVM's 12.288 MHz oscillator as the MCLK source.

We'll illustrate the connections using one root EVM and one client EVM board using AD2428 based AB0001 modules.

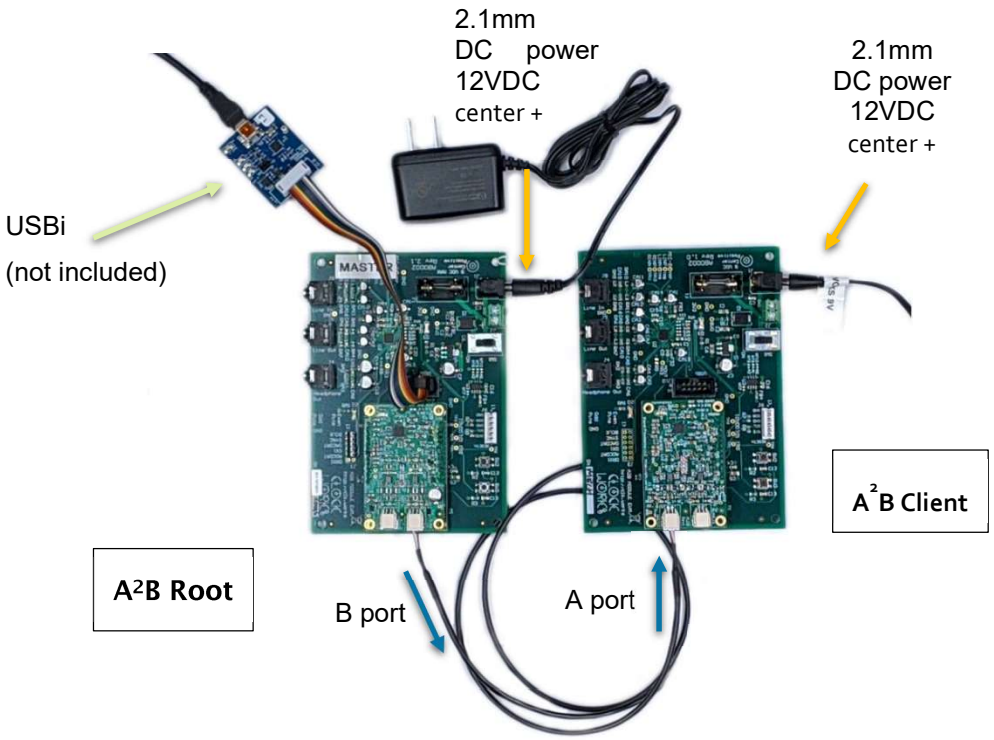


Figure 5 Setting up two EVMs

An A²B cable is used to connect the first (root) EVM's downstream port (B) to the upstream port (A) on the client node EMV. If you have more nodes to connect they would connect to the downstream (B) port of the slave node.

Once the hardware is connected you can power up the boards, the order that they are powered up in does not matter. Please see the software section for information about configuring the boards.

2.4 PUSHBUTTON SWITCHES

The EVM includes two pushbutton switches that connect to the A²B module. Specific software must be used to define interactions with the switches. A small capacitor is used to aid in the debounce of the switch but software should verify a clean switch closure and release.

SW2 is connected to the DRX1 signal, which is also GPIO 6 on the AD2428 device.

SW3 is connected to the IRQ pin, which can also be used as GPIO 0 on the AD2428 device.

2.5 USE

2.5.1 ANALOG OUT

The default DSP program uses the headphone output jack. To use the line out jack the default program must be modified to route in the I²S input to the DAC and enable both the headphone and line out jack in the chip register setup.

220uF capacitors are used on the headphone output for AC coupling. With 32 ohm headphones Fc is 22 Hz; 16 ohm headphones Fc is 45 Hz.[3]

For the line out 10 uF capacitors are used, with typical 10 kohm input impedances of an amplifier Fc is 1.5 Hz.

3 EVM DESIGN INFORMATION

For the EVM schematics (pdf with right click for part details) and a 3D view, along with gerbers and .STEP files are available for download from the product page. Source files in Altium format are available on request.

For the modules a 3D view and .STEP file are available for download. The module schematics follow the ADI reference designs and those should be referred to before undertaking your own custom design.

[3] So yes, if you have low impedance headphones and want a good experience you probably want an external buffer amplifier if you wanted to do some sort of listening beyond "*yup there's signal here.*"

3.1 POWER INPUT

The EVM design include a diode on the DC input to prevent damage from a reverse connected input supply.

The AD2428 maximum supply voltage is 9V. The AB0001 module design includes a Schottky diode as suggested by ADI's design guide for a local powered A2B module, meaning there is another 0.4V or so drop on the A²B module and this the actual voltage to the AD2428 on the module is about 8.6V.

The only possible impact from the lower supply voltage is if A²B phantom power feature is used to power multiple downstream nodes; each node has a 0.4V drop. For more details please see the full AD2428 datasheet for a discussion on phantom power of nodes.

The EVM does not use the IOVDD module output. The A²B module must be jumpered for 3.3V IO when used with the EVM.

3.2 FEATURES FOR MODIFYING THE EVM

There may be cases where a change to the default wiring is needed. There are a couple of provisions in the design to attempt to simplify making modifications. Please note that Clockworks can not support modified boards nor take any responsibility for your modifications making you late for dinner or something worse like burning down the building.

3.2.1 TESTPOINTS

Testpoints are available on some of the signals to facilitate probing or modifying the board. Please see the schematic for their (electrical) location.

The unused analog I/O pins of the ADAU1761 are also brought out to a test point; please see the ADAU1761 datasheet for how to use them as no circuitry is provided on the EVM.

In addition to signals a few GND test points are provided. These are connected to the internal ground plane of the PCB. There is a single ground plane in this design.

3.2.2 NT COMPONENTS

NT (Net Tie) components are 0603 footprints with copper trace between them. By cutting the traces the two net pieces can be isolated and connected to different places. They are available on the ADAU1761's I²S signals and the 3 address bits for the EEPROM..

3.2.3 MICROPHONE INPUT

The EVM provides test point holes on a 0.1" grid for the microphone pins of the ADAU1761. Please see the ADAU1761 datasheet for suggested circuit configurations for using these inputs.

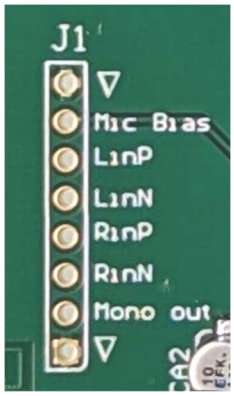


Figure 6 ADAU1761 microphone input signal breakout

3.3 J9 EXTERNAL CONTROL CONNECTOR

A 6 pin connector can be installed here to control the board over I²C. It follows the same pinout as used on some other Clockworks products. Pin assignments are as follows:

1	GND
2	SDA
3	GND
4	SCL
5	3.3V
6	RESETn

The RESETn is normally defined as an active low signal that is driven by open collector logic where the default driving source includes a pullup. On the EVM board and the Ab0001 AD2428 A²B

module there is no reset capability, the parts have built in brown-out and reset. Therefor this signal serves no purpose for the normal EVM use.

For the AD2437 AB0003 this signal connects to the module's RESET in which does connect to the AD2437 reset input.

If using the I²C bus on a client (non-root) node keep in mind that the AD2428 or AD2437 device on the A²B module is normally the I²C master.

3.4 J3 I²S CONNECTOR

This 12 pin dual row connector (2.54mm/0.1" spacing) provides access to the I²S signals. It is not installed on the EVM you can insert whatever connector may be needed.

2	BCLK
4	SYNC
6	DACDATA (DTX0)
8	DTX1
10	ADCDAT (DRX1)
12	DRX1 (IO6 - SW2)
1,.., 11	GND (odd pins)

4 USE OF THE EVM WITHOUT AN A²B MODULE

The EVM Board can be used without an A²B module for ADAU1761 development using ADI's Sigma Studio tools. The ADAU1761 receives its clock on the MCLK pin, it will not operate without that signal. The default software expects a 12.288 MHz clock which corresponds to 256 Fs for an Fs of 48 kHz.

Place the MLCK jumper in the left position to select the local oscillator.

Figure 7 MCLK jumper for local oscillator

4.1 USBI CONNECTION

For use in root (A²B bus master) mode, or to develop software for the ADAU1761 Analog Device's Sigma Studio tools are used. An emulator (ADI calls it *USBi*) is required for downloading and interacting with the ADAU1761. For more information please see:

http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-adusb2ebz.html#eb-documentation

5 AB0001 MODULE INFORMATION

All module functions are determined by the AD2428 device, please refer to that device's datasheet and technical reference manual for detailed operating information.

5.1 AB0001 CONNECTORS

Pinout – A Connector.

Pin	Name	Notes
1	IOVDD	Jumper selects between AD2428 internal regulator voltages, default is 3.3V (vs. 1.8V)
2	GND	
3	ADR2	AD2428 ADR2/IO2 line with 10K pulldown. For client nodes this is CLKOUT2, which is used as MCLK
4	ADR1	AD2428 ADR1/IO1 line with 10K pulldown
5	IRQ	AD2428 IRQ/IO0 line
6	GND	
7	SDA	I2C data
8	SCL	I2C clock
9	GND	
10,11	Vxx	Power 5-8V suggested range, 4V min, 9V max.
12	GND	

Pinout – B Connector.

Pin	Name	Notes
1	GND	
2	BCLK	AD2428 bit clock (output from client module, input as root)
3	GND	
4	SYNC	AD2428 frame sync (output from client module, input as root)
5	GND	
6	DTX0	AD2428 DTX0 with 10K pulldown (output)
7	DTX1	AD2428 DTX1 with 10K pulldown (output)
8	GND	
9	DRX0	AD2428 DRX0
10	DRX1	AD2428 DRX1
11	GND	
12	IO7	AD2428 IO7/PDMCLK pin[4]

[4] Original rev 1 modules were built with AD2425W and this pin needed to be grounded to indicate a slave device. There is a jumper on the EVM that can be cut to remove the ground connection on this pin if needed (the EVM is however only capable of being an A²B slave)

5.2 MODULE MECHANICAL INFORMATION

Top View.

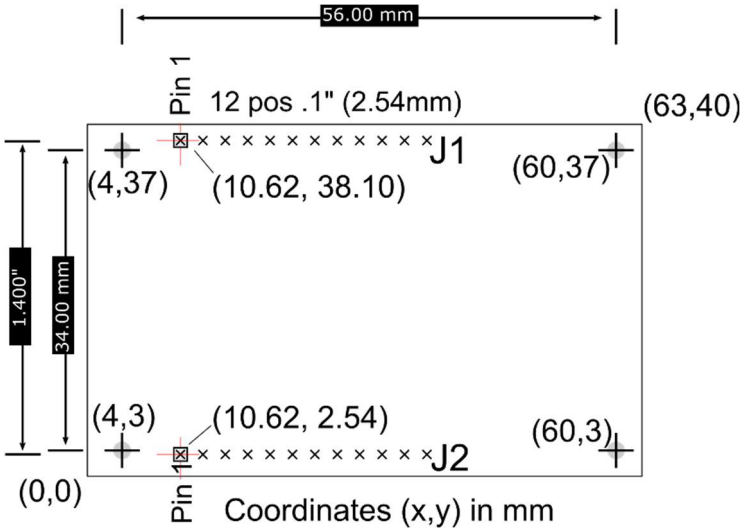


Figure 8 AB0001 A²B module dimensions

The board is slightly asymmetrical relative to the mounting holes, which are 4mm from the back edge and 3mm from the front (A²B connector) edge.

For the AB0001 all electrical components are on the top side of the board. The two 12 pin connectors are located on the bottom side of the board.

For the AB0003 electrical components are on both sides of the board. One 16 pin single row and a dual row 12 position (i.e. 24 pins total) connectors are located on the bottom side of the board.

Component heights are nominally as follows for the figure shown below. Component substitutions during assembly may sometimes be required; this may alter these values.

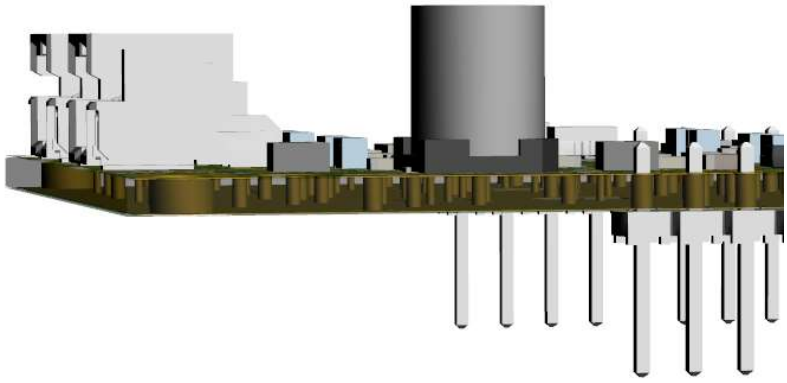


Figure 9 A²B module side view cut-away

Board thickness 1.6 mm(.062”).

Plastic on bottom connectors thickness 2.29mm (.090”).

Pin length 6.1mm (.24”).

DuraClik connector height 6.4mm (.252”).

C24 (bulk 100 uF cap) height 7.7mm (.303”)

5.3 AB0001 (AD2428) INTERFACE DETAIL

The following two schematic sections from the A²B module are provided to aid in understanding the A²B module interface.

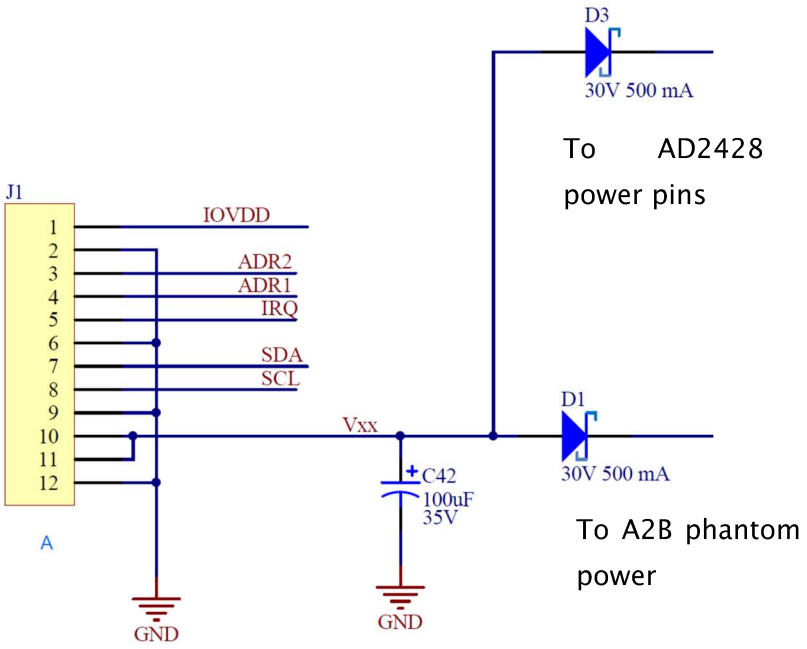


Figure 10 Module A connector wiring[5]

C42 is required by the ADI design rules and should be kept in mind for the load capacitance on the external supply.

The other connector carries the I²S signals. Analog Devices suggests series termination and a small 27 pF capacitor to reduce EMI issues and the module implements those suggestions.

[5] RefDes for Rev 2 module PCB shown unless noted otherwise.

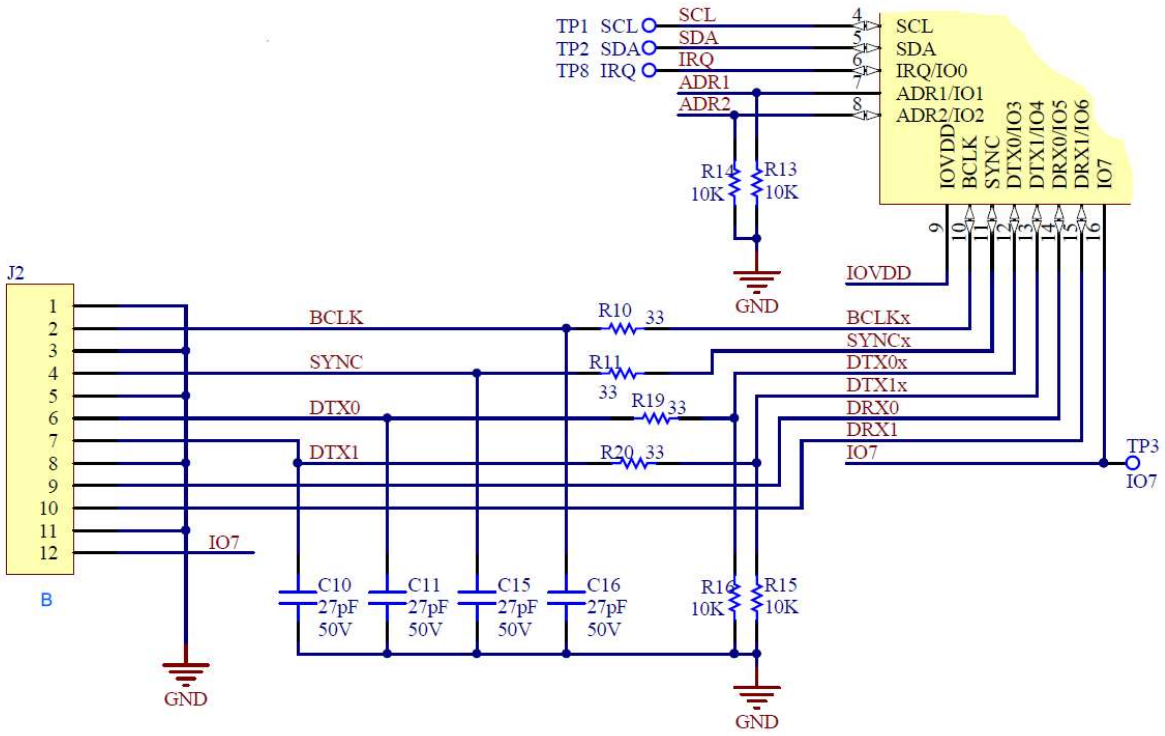


Figure 11 Module B connector wiring. On AD2428 IO7 can also be PDMCLK out pin.[6]

The remainder of the module schematic corresponds to the ADI recommended design for locally powered nodes, as that guidance is subject to change please see the ADI documentation for further details.

Unused module pins may be left floating except for the I²C pins. (see 5.3.3).

5.3.1 IOVDD

This pin is normally an output from the module. It can be configured for 3.3V or 1.9V[7] operation.

If neither voltage selection jumper is installed then IOVdd can be set externally, however all of the AD2428 rules about supply sequencing must be strictly adhered to.

The I²C bus must match the IOVdd, i.e. the pullup resistor's connection to the supply.

[6] RefDes for Rev 2 module PCB shown unless noted otherwise.
[7] The datasheet for the AD2428W specified the range of voltage from that supply as 1.7 to 1.98 with 1.9 being the typical value. In a few places though it's referred to as the 1.8V supply.

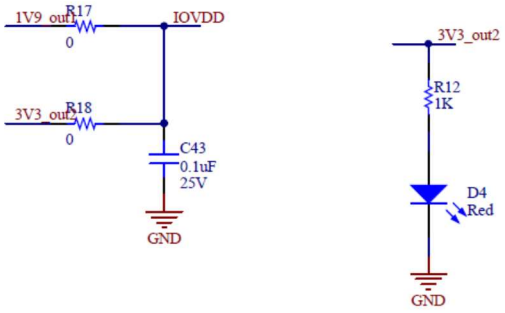


Figure 12 Rev 2 PCB IOVdd connections. 1V9 is connected to the AD2428 VOUT1, pin 32. 3V3 is connected to the AD2428 VOUT2, pin 29.

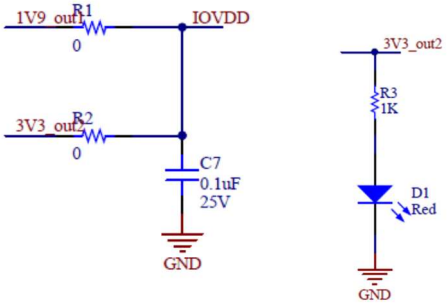


Figure 13 Rev 3 PCB IOVdd connections. 1V9 is connected to the AD2428W VOUT1, pin 32. 3V3 is connected to the AD2428 VOUT2, pin 29. (different RefDes #s)

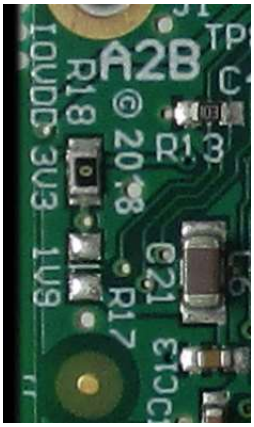


Figure 14 Rev 2: IOVdd jumper resistors. 3.3V is the factory default

5.3.2 VXX

The AD2428 datasheet shows that this pin should not exceed 9V, though the Schottky diode in the module input supply connection does mean that 9.3V is closer to the absolute maximum limit of the module. The minimum voltage is 4.2V for the module to operate, however no phantom power can be supplied at that voltage.

5.3.3 SCL, SDA

As stated on the AD2428 datasheet, these two pins should be grounded if not used by system this board plugs in to.

If I²C is used then the main board must provide the I²C pullups and correctly match the IOVdd.

6 AB0331 MODULE

All module functions are determined by the AD2437 device, please refer to that device's datasheet and technical reference manual for detailed operating information.

6.1 AB0331 CONNECTORS

Pinout – A Connector.

Notes	Name	Pin	Pin	Name	Notes
Connected to EVM reset line	/RESET	2	1	IOVDD	1.8V or 3.3V. Jumper selects between AD2437 internal regulator voltages, defaults to 3.3V
SPI Master in/Slave Out Data	MISO	4	3	GND	
	GND	6	5	ADR2	AD2437 ADR2/IO2 line with 10K pulldown
SPI Master out/Slave In Data	MOSI	8	7	ADR1	AD2428 ADR1/IO1 line with 10K pulldown (on EVM connects to LED)
AD2437 SIO2 pin. Use as SPI select.	SIO2	10	9	IRQ	AD2437 IRQ/IO0 line (on EVM can connect to pushbutton or GPIO9 of the RP2040 board)
	GND	12	11	GND	
Reserved	-	14	13	SDA	I2C data
	GND	16	15	SCL	I2C clock
SPI data clock	SCK	18	17	GND	
A2B bus power (24V 2 amp max) input	VBus	20	19	Vxx	Power 5-9V suggested range, 4V min, 9.2V max.
	VBus	22	21	Vxx	
	GND	24	23	GND	

Pinout – B Connector.

Pin	Name	Notes
1	GND	
2	BCLK	AD2437 bit clock (output from client module, input as root)
3	GND	
4	SYNC	AD2437 frame sync (output from client module, input as root)
5	GND	
6	DTX0	AD2437 DTX0 with 10K pulldown (output)
7	DTX1	AD2437 DTX1 with 10K pulldown (output)
8	GND	
9	DRX0	AD2437 DRX0
10	DRX1	AD2437 DRX1
11	GND	
12	IO7	AD2428 IO7/PDMCLK pin[8]
13	GND	
14,15	Vbout	CFG4 phantom power from upstream connection (24V typical)
16	GND	

6.2 MODULE MECHANICAL INFORMATION

Top View.

[8] Original rev 1 modules were built with AD2425W and this pin needed to be grounded to indicate a slave device. There is a jumper on the EVM that can be cut to remove the ground connection on this pin if needed (the EVM is however only capable of being an A²B slave)

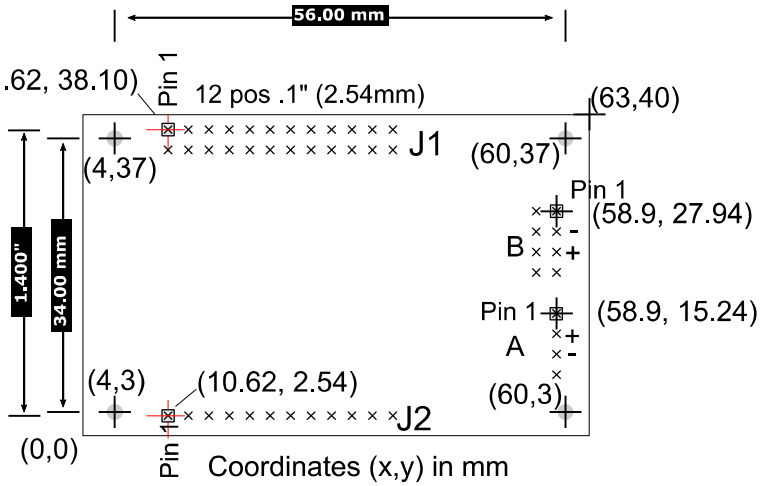
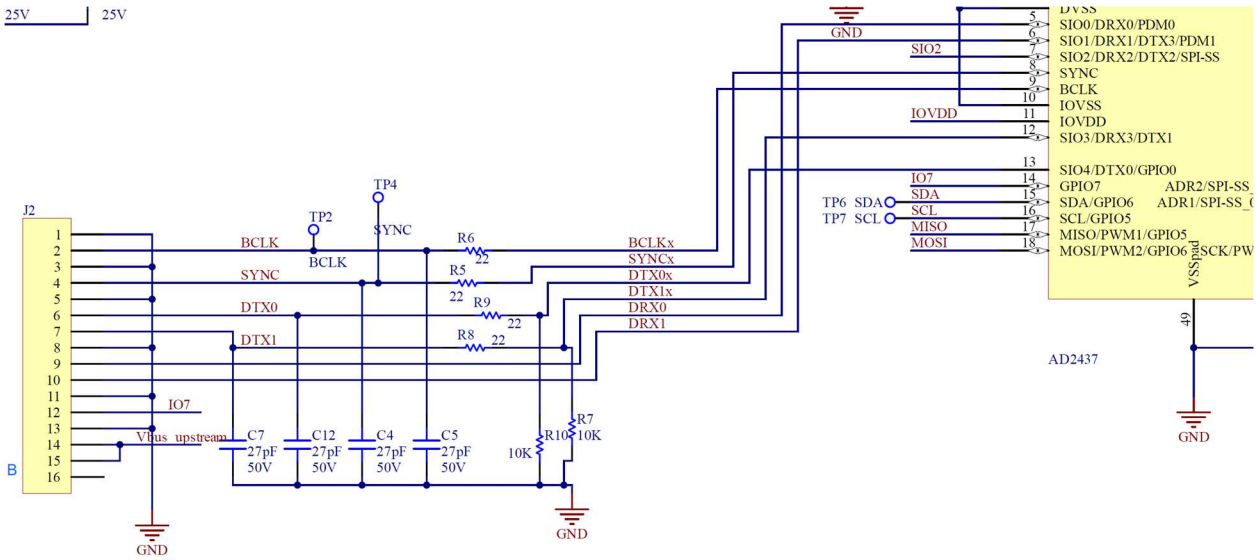
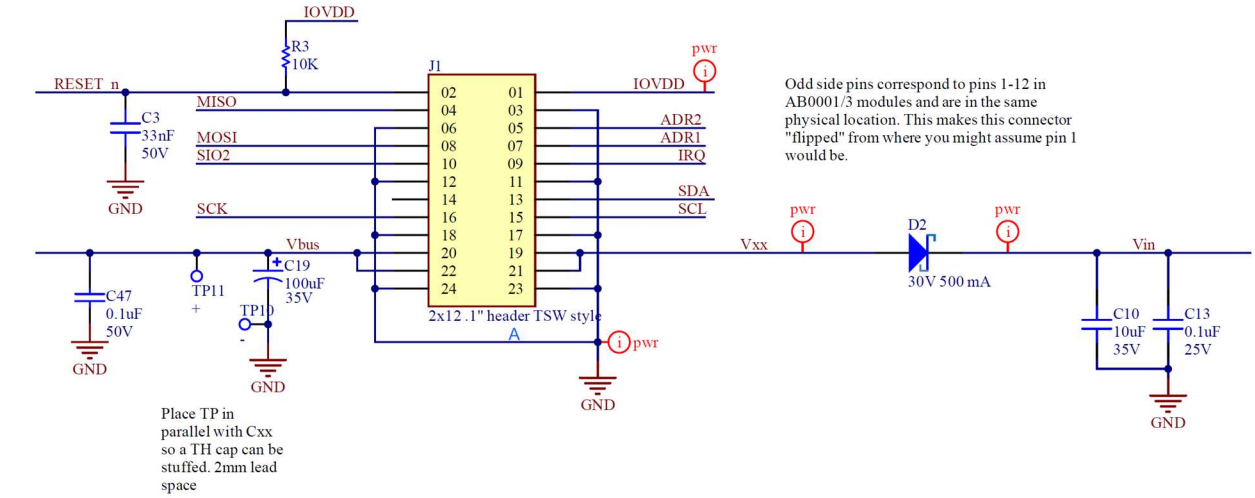


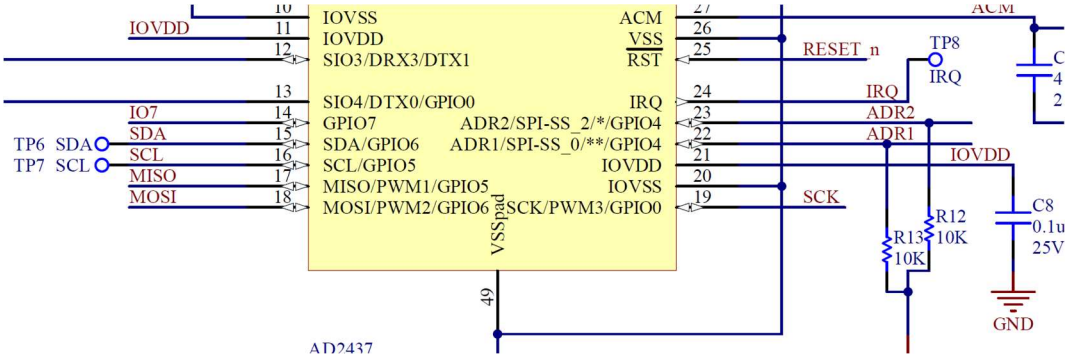
Figure 15 AB0331 module

6.3 AB0331 (AD2437) INTERFACE DETAIL

The following schematic sections from the A²B module are provided to aid in understanding the A²B module interface.




IOVDD
R3
10K
RESET_n
MISO
C3
33nF
50V
GND
MOSI
SIO2
SCK
Vbus
C19
100uF
35V
TP11
TP10
C47
0.1uF
50V
J1
2x12 .1" header TSW style
A
pwr
IOVDD
ADR2
ADR1
IRQ
SDA
SCL
Odd side pins correspond to pins 1-12 in AB0001/3 modules and are in the same physical location. This makes this connector "flipped" from where you might assume pin 1 would be.
Vxx
D2
30V 500 mA
Vin
C10
10uF
35V
C13
0.1uF
25V
Place TP in parallel with Cxx so a TH cap can be stuffed. 2mm lead space


IOVSS
IOVDD
SIO3/DRX3/DTX1
SIO4/DTX0/GPIO0
GPIO7
SDA/GPIO6
SCL/GPIO5
MISO/PWM1/GPIO5
MOSI/PWM2/GPIO6
ACM
VSS
RST
IRQ
ADR2/SPI-SS_2/*/GPIO4
ADR1/SPI-SS_0/**/GPIO4
IOVDD
IOVSS
SCK/PWM3/GPIO0
VSSpad
AD2437
IO7
SDA
SCL
MISO
MOSI
TP6 SDA
TP7 SCL
RESET_n
TP8
IRQ
ADR2
ADR1
SCK
R12
10K
R13
10K
C8
0.1u
25V

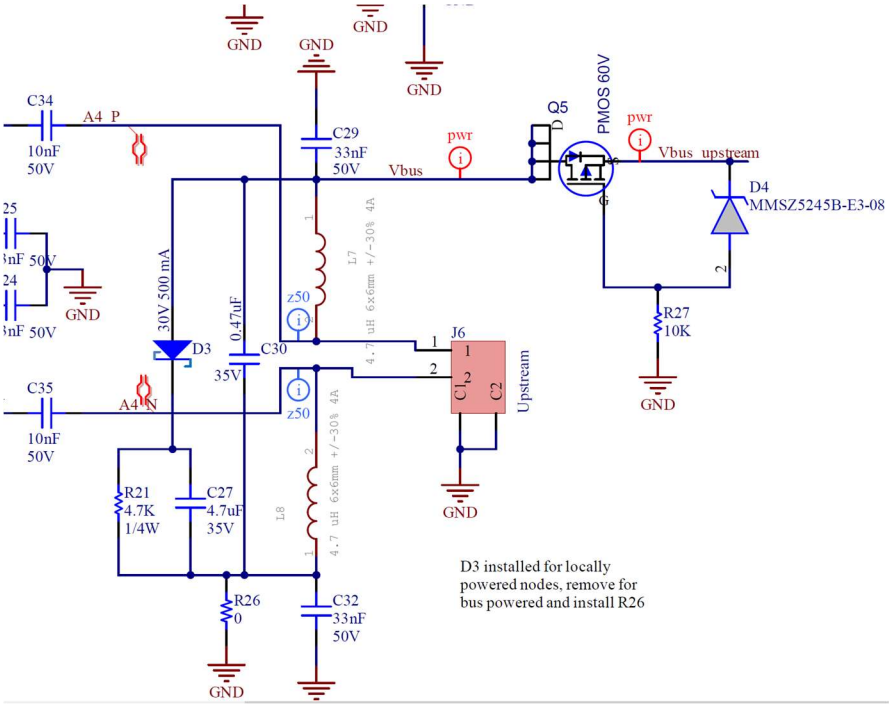

C34
A4_P
10nF
50V
C29
33nF
50V
Vbus
pwr
Q5
PMOS 60V
Vbus_upstream
D4
MMSZ5245B-E3-08
R27
10K
D3
30V 500 mA
0.47uF
C30
35V
J6
Upstream
C35
A4_N
10nF
50V
R21
4.7K
1/4W
C27
4.7uF
35V
R26
0
C32
33nF
50V
D3 installed for locally powered nodes, remove for bus powered and install R26

Figure 16 Schematic sections for external connectivity

7 CONFIGURATION EXAMPLE WITH SIGMASTUDIO

Set up and operation of SigmaStudio+ is basically the same as described here for SigmaStudio 4.7. The settings dialogs look different but have all of the same controls shown here.

The AB0331 AD2437 module can only be used with SigmaStudio+, please see the next chapter for details.

This section outlines the steps for using the Clockworks EVM with the Analog Devices A²B add-on for Sigma Studio. ADI includes tutorial examples with the software add-on and this section shows how to use the example

..\Schematics\BF\A2BSchematics\adi_a2b_3NodeSampleDemoConfig.dspproj

You can also create a new example on a blank sheet SigmaStudio sheet following the procedures in the ADI documentation. When first starting out it may be easier to edit a provided example for your host (and save it as a new file!) than to get all of the settings in place.

If using the SC589 based SHARC Audio Module please also see the next section as that board as that support has been updated from when this document was developed. While the concepts are the same as working with the ADI EVMs there are a few differences that must be considered.

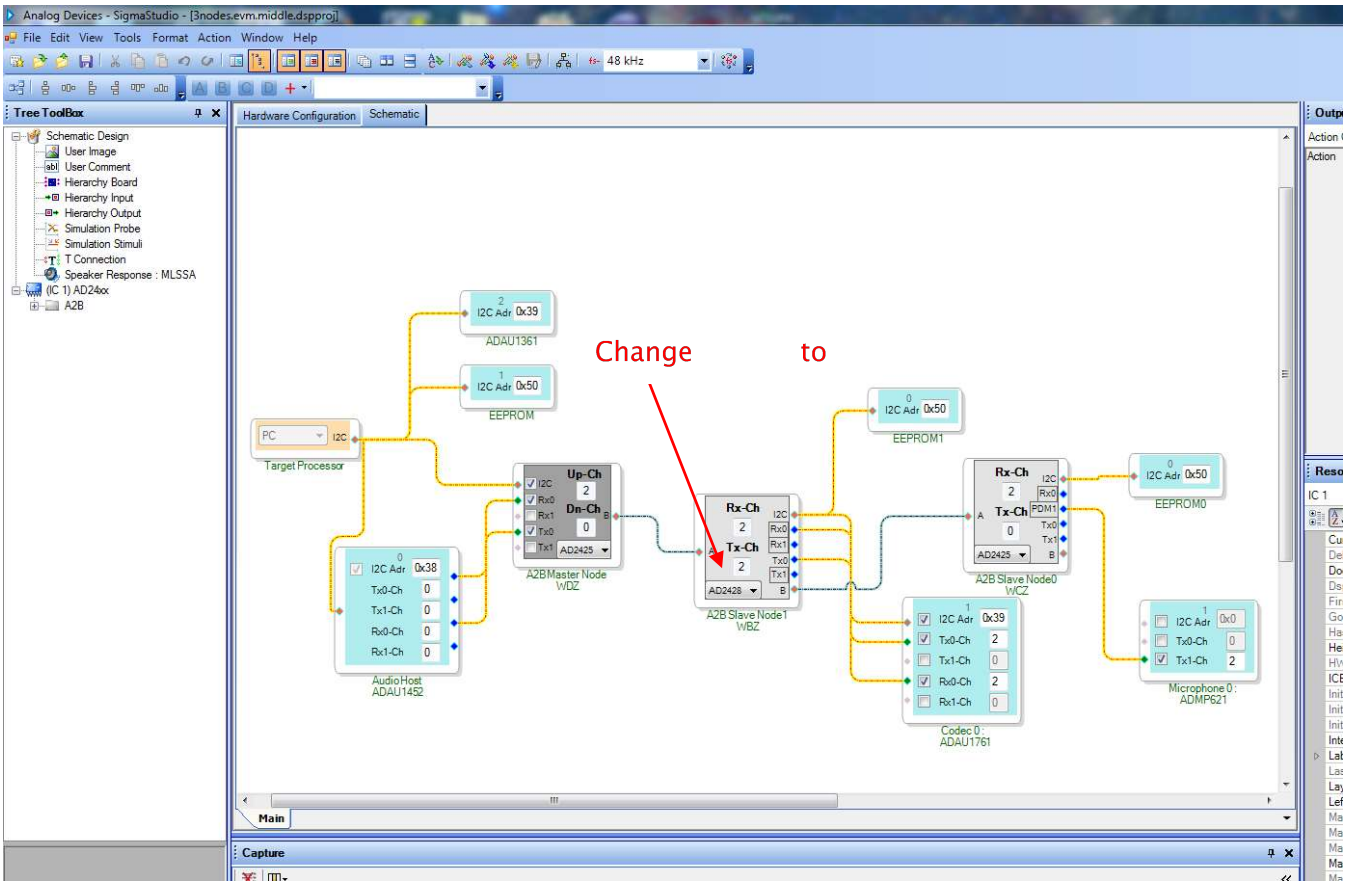


Figure 17 Sigma Studio 3 node example

Figure 6 shows one of the ADI default example with the (ADI WDZ board) ADAU1452 master and WBZ and WCZ slaves. The Clockworks EVM replaces the WBZ slave in this example. To do that change the chip type on the 2nd node to AD2428 as shown by the red arrow.

If you don't have the WCZ microphone board you can delete it from the example.

In this example the root node peripherals must be programmed, which is enabled by selecting its properties and checking the *Program during discovery* box. The path to the XML file with the ADAU1452 must be valid and if just trying the system for the first time should be the default pass through program.

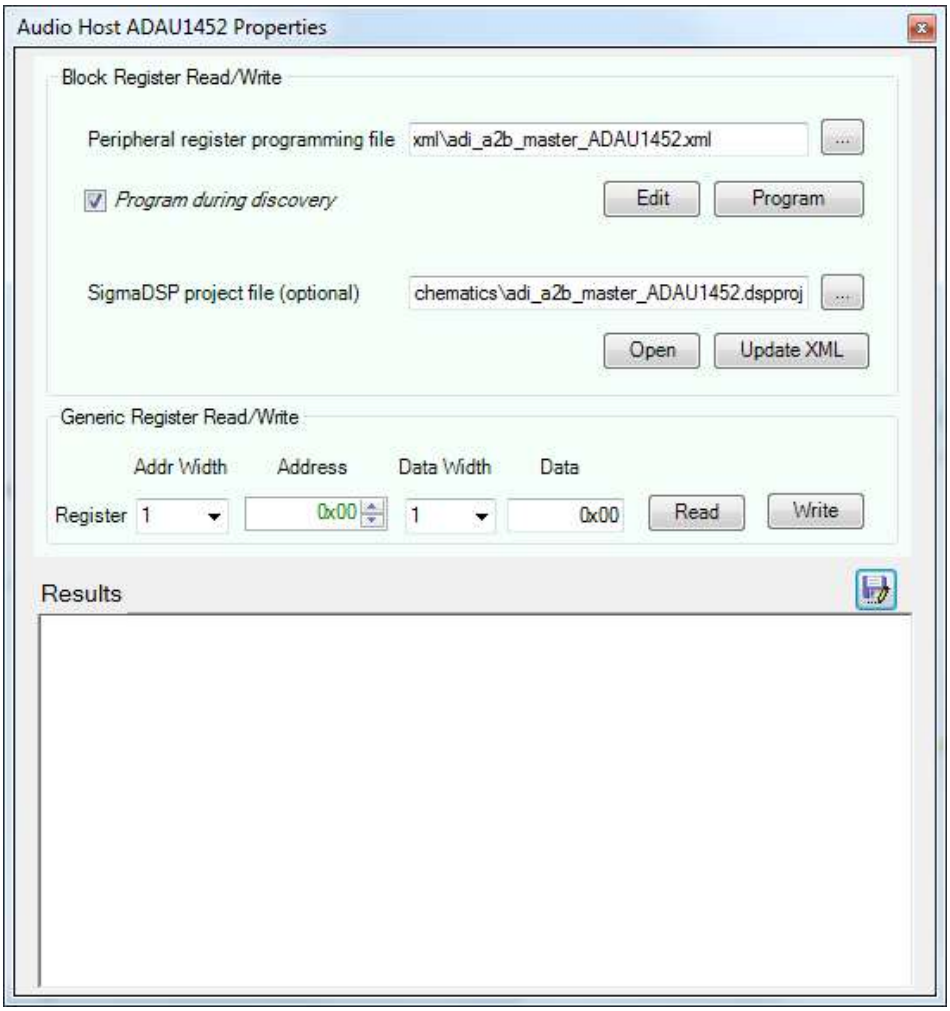


Figure 18 WDZ Root node properties

The Clockwork's EVM board ADAU1761 must likewise be programmed, and the default program used with the WBZ board can be used, as shown in Figure 8. As with the root node the *Program during discovery* box must be checked.

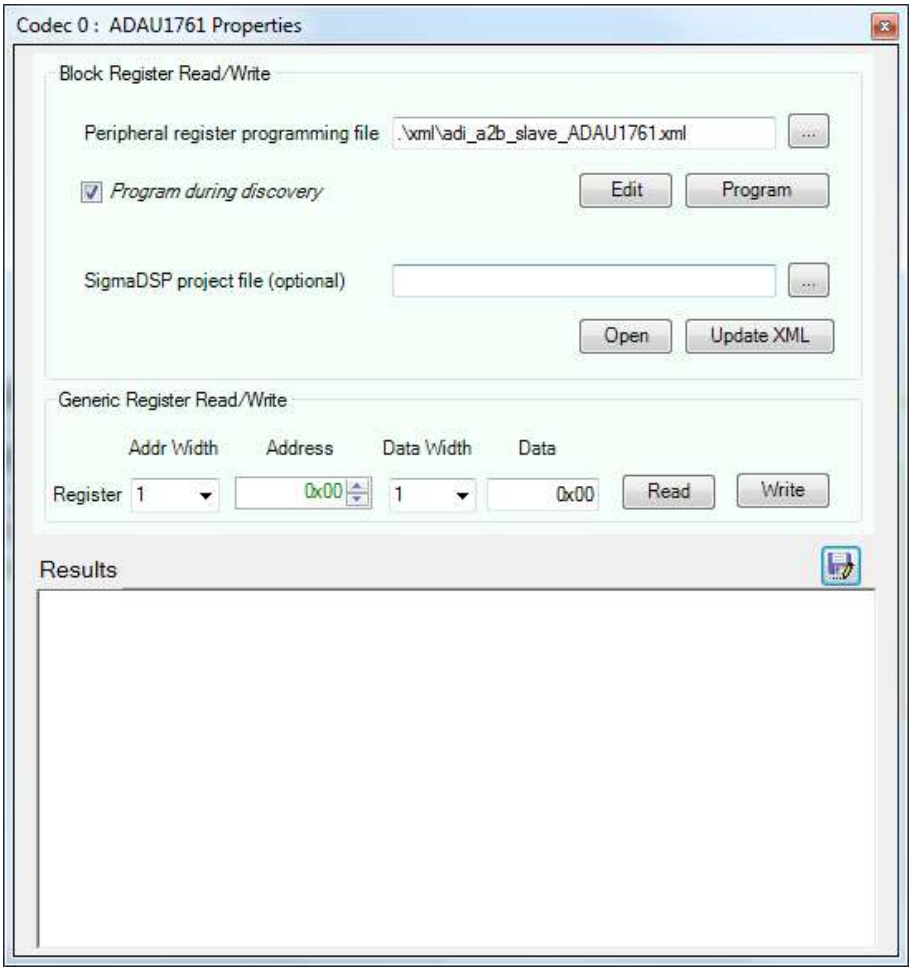


Figure 19 ADAU1761 programming on Clockworks EVM board (as a client node)

It should not be necessary to change the AD2428 properties, the I²C setup screen (properties home page) is shown in Figure 9.

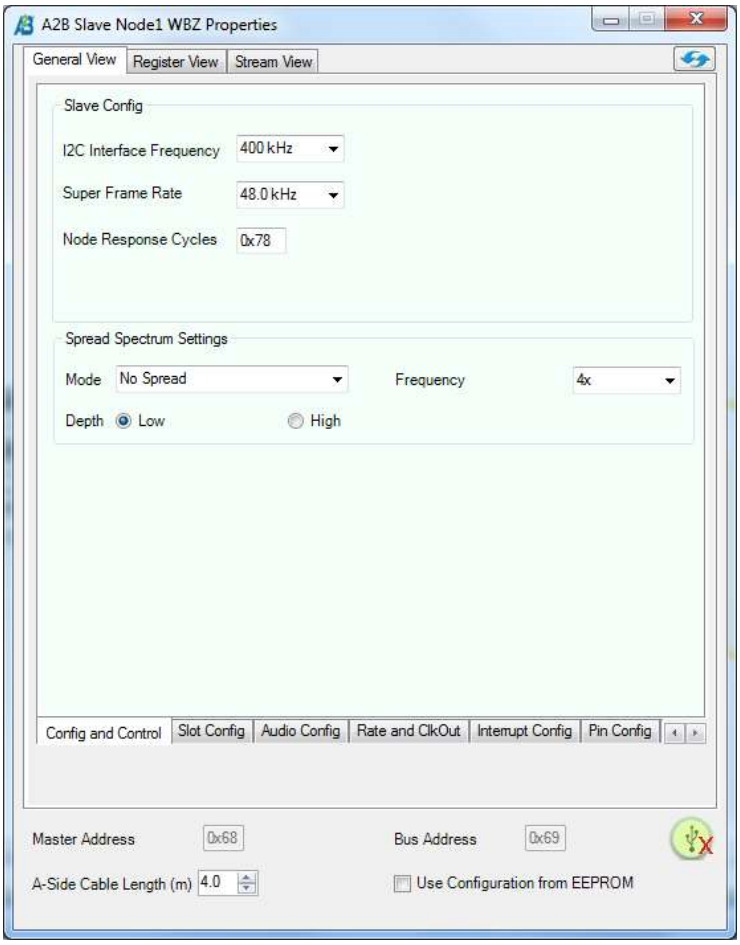
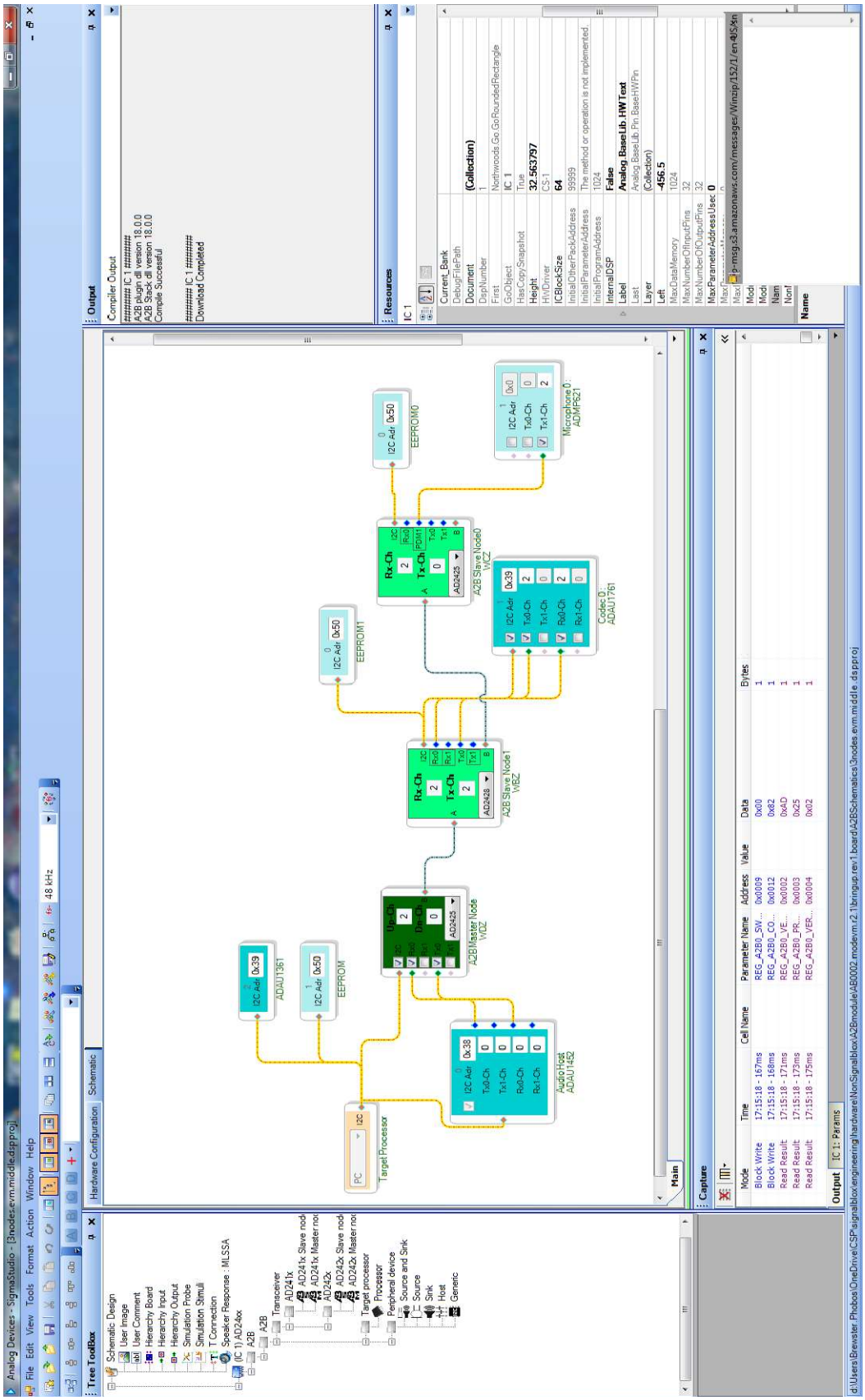


Figure 20 AD2428 Properties for Clockworks EVM based on defaults for ADI WBZ board.

Once all of the settings are checked hit the Link Compile Download button and after a few seconds the A²B nodes should turn green to indicate the networks has been discovered, as shown in the figure on the next page.



This particular ADI example takes the audio from the 2nd client node (a WCZ mic board) and plays it out the headphone jack of the 1st client node (the Clockworks EVM here, a WBZ board in the original example). The line in to the 1st client node is sent to the headphone out of the A2B root node.

7.1.1 I²S SETUP

If using the ADI provided example all of the settings are correct for the AD2428 to exchange audio with the ADAU1761 when it runs the corresponding default program. If inserting a A²B transceiver block and peripheral block (instead of copying) all of the settings need to match that used for the SigmaDSP's I²S interface. Failure modes include no audio, very badly distorted audio, or full scale modulated noise.

Figure 21 and Figure 22 show the two relevant settings for use with the default ADAU1761 program.

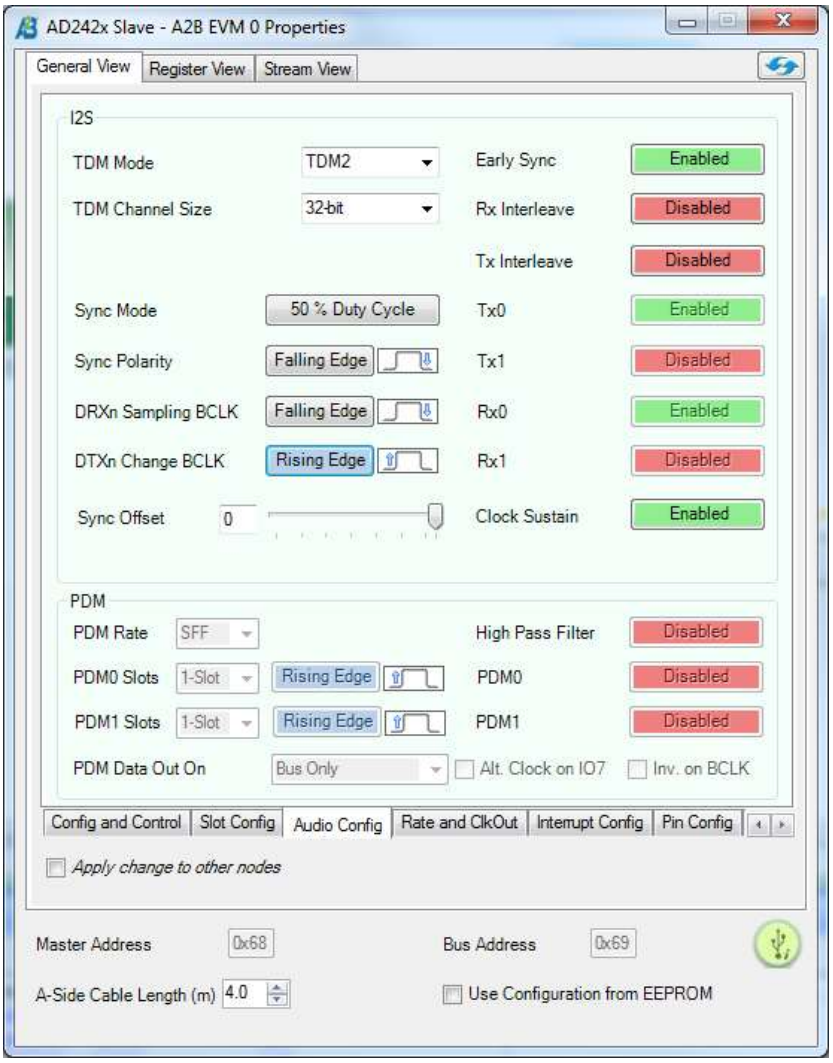


Figure 21 Audio Config tab settings for ADAU1761 default EVM program

If any of the edge or Sync setting are wrong highly distorted audio will result. These settings correspond in part with the `I2SCFG` register (0x42), which should have a value of 0x91 if the bits have been set properly.

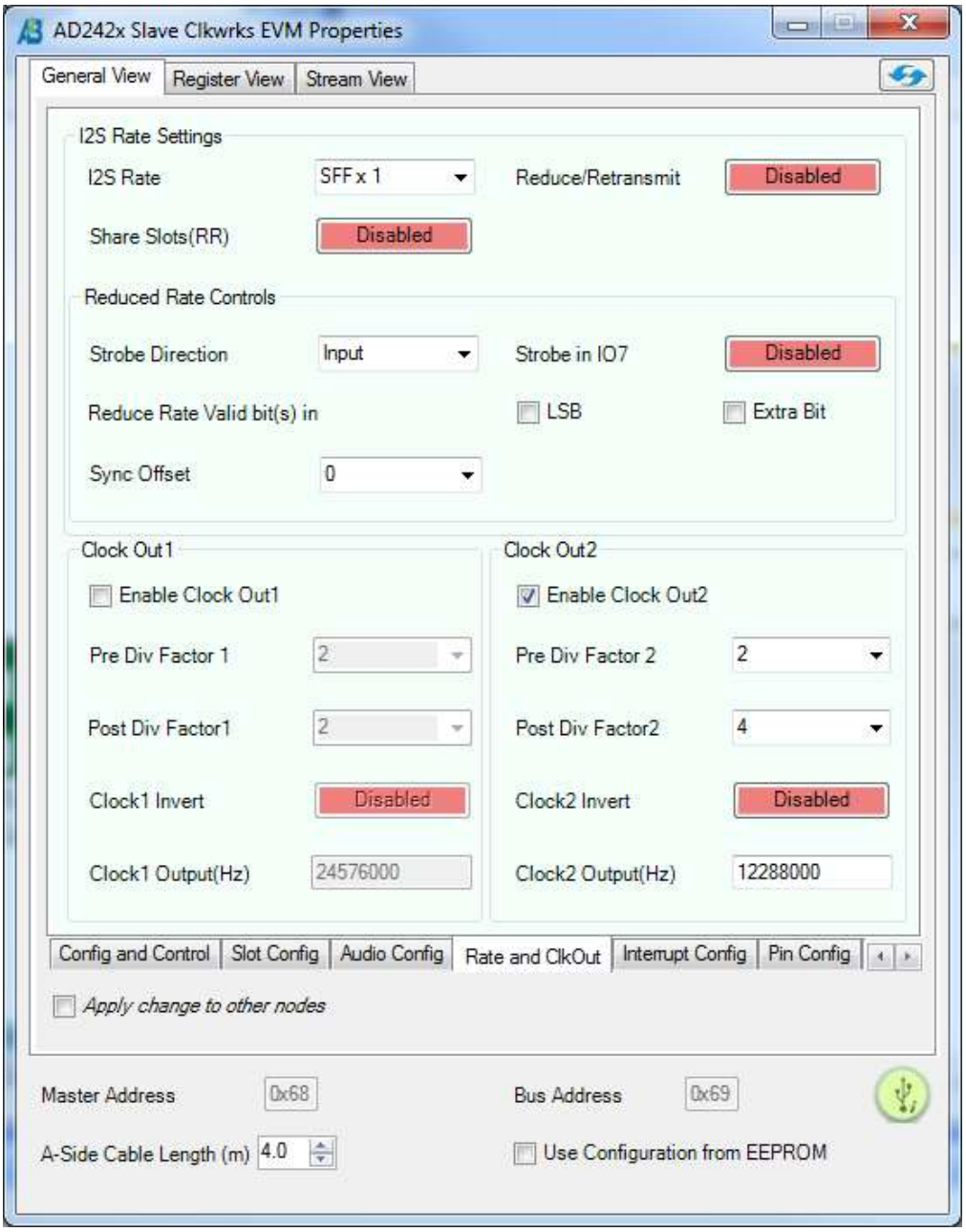


Figure 22 Rate and ClkOut tab settings for ADAU1761 default EVM program (slave node)

7.2 ACCESSING THE LED AND SWITCHES FROM SIGMA STUDIO

GPIO can be accessed from the AD2428 block in Sigma Studio by right clicking on the block associated with the EVM board. There are two ways to do this, one at a low level of direct register access, like what a program would do, and the other via the more convenient "General View," though that second method only helps for output, i.e. turning the LED on and off.

7.2.1 THE HARD WAY

Select Register View and the IO tab. For details of the registers and their bit maps please see the AD242x Programmers Reference Manual.

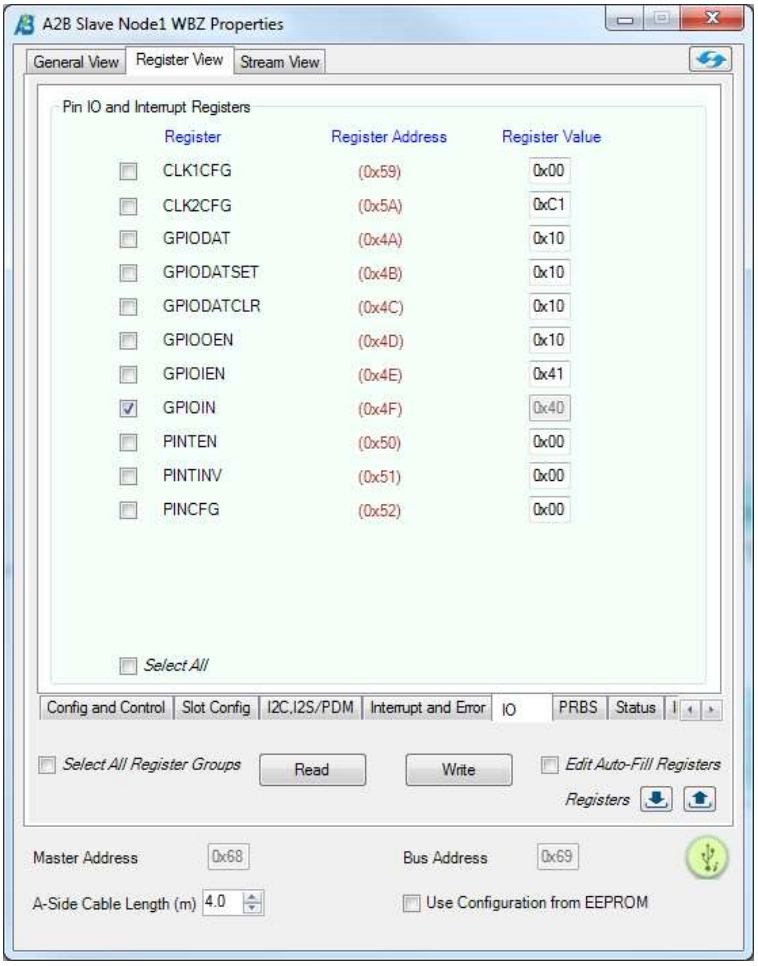


Figure 23 IO tab in SigmaStudio

Top operate the LED first write 0x10 to the GPIOOEN register (0x4D). Then place 0x10 in the GPIODATSET or GPIODATCLR register and write that to turn the LED on and off. The LED is connected (through the EVM to A2B module connector) to DTX1/IO4 pin of the AD2428.

To read the switches for write 0x41 to GPIOIEN register (0x4E). SW2 is connected to DRX1/IO6 (bit map 0x4)0and SW3 to IRQ/IO0 (bit map 0x01). Select GPIOIN and hit the Read button to see the switch values.

7.2.2 THE EASY WAY

This will allow turning the EVM's LED on and off, which might be useful to confirm the node you've selected is the one you think it is. On the General View tba select the Pin Config dialog and set GPIO4 to be an output. High turns the LED on, Low shuts it off.

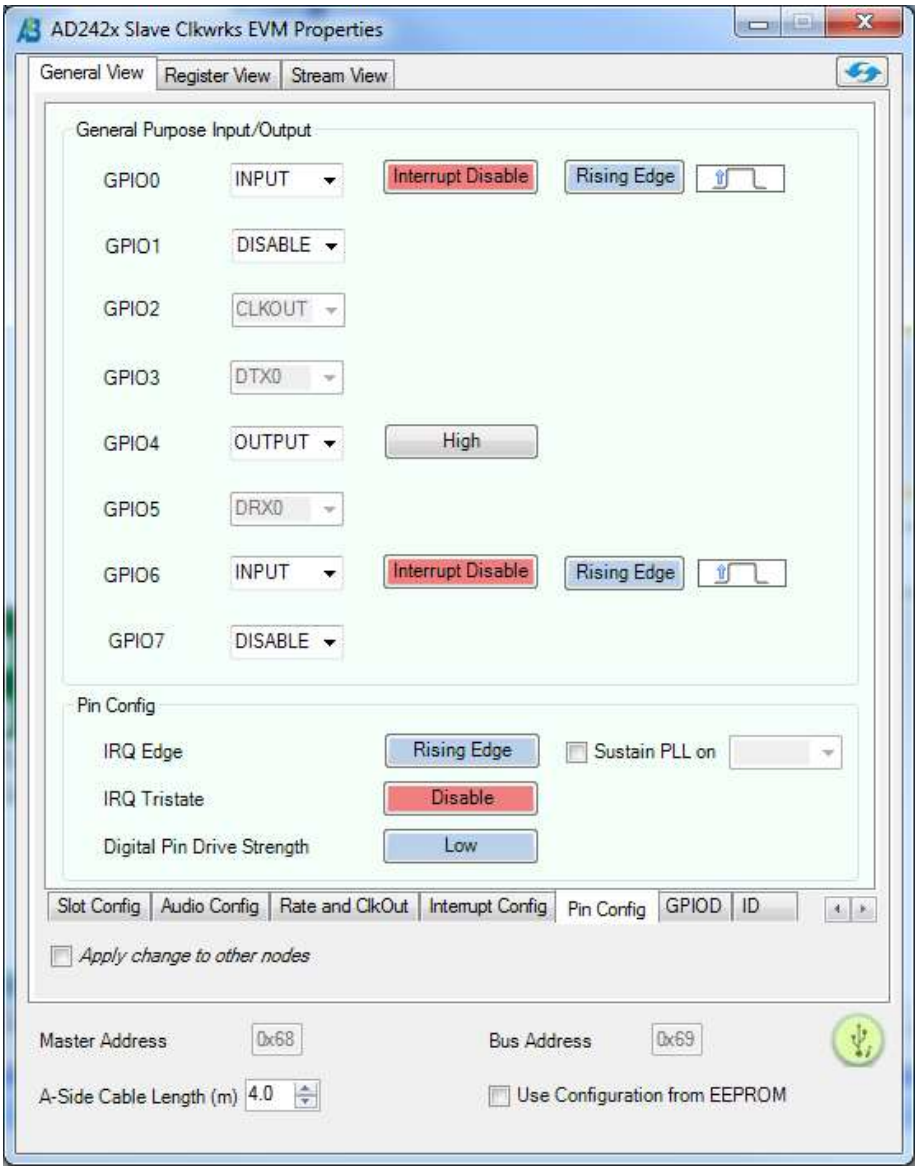


Figure 24 EVM User LED (GPIO 4) control

7.3 CONFIGURING A ROOT MODE EVM

Start with the supplied example (`EVM_mstr_EVM_slave_WCZ.dsproj`) or create a new project and add the host, AD2428, and ADAU1761 blocks as shown. If creating from scratch make sure you use the AD242x root node (i.e. MCLK jumper on left pins) for the first one and client nodes (i.e. MCLK jumper on right pins) for the rest.

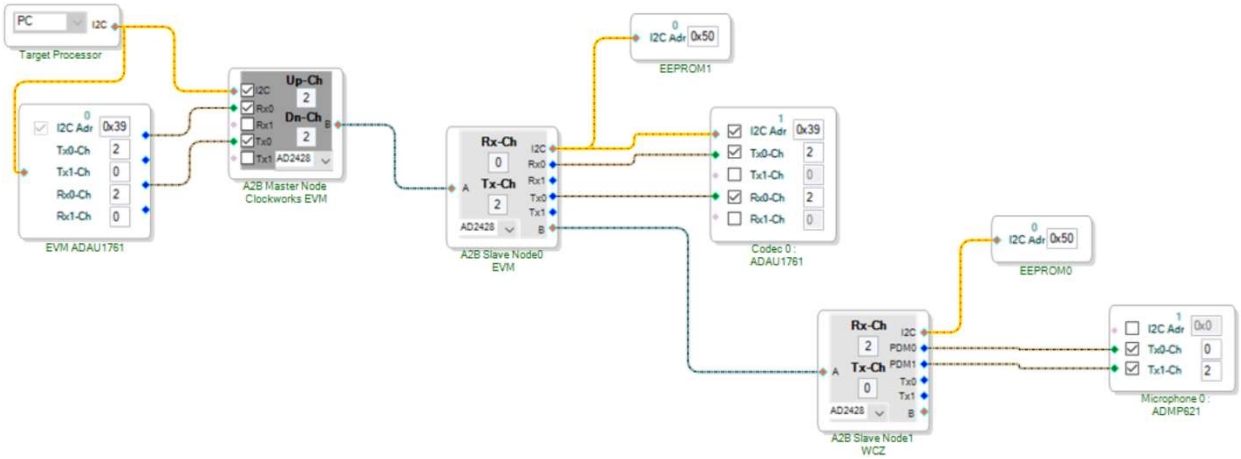


Figure 25 EVM root with EVM client example (with ADI WCZ node at end)

For the client node follow the directions in the prior section, the remainder of this section will just look at the root node setup.

The AD2428 and the ADAU1761 are both controlled over I²C from the host running SigmaStudio. When a *Link Compile Download* is performed the AD2428 is initialized and the ADAU1761 program is also loaded as the *Program during discovery* is enabled in the ADAU1761 properties:

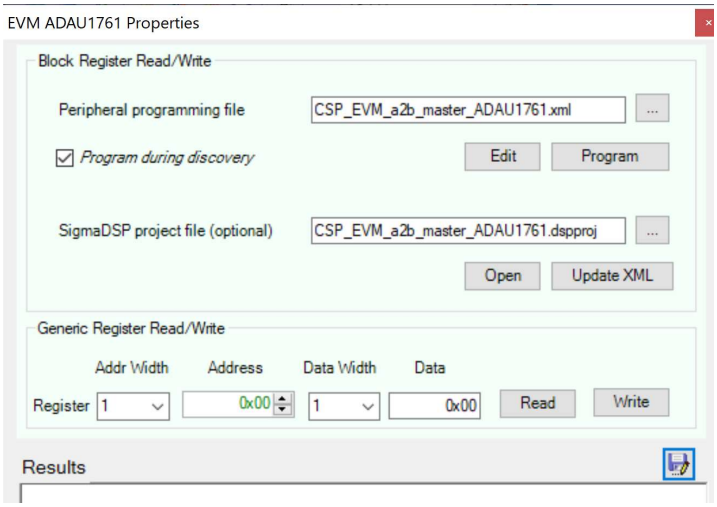


Figure 26 Root node 1761 file setup

The root node has slightly different properties than the slave node.

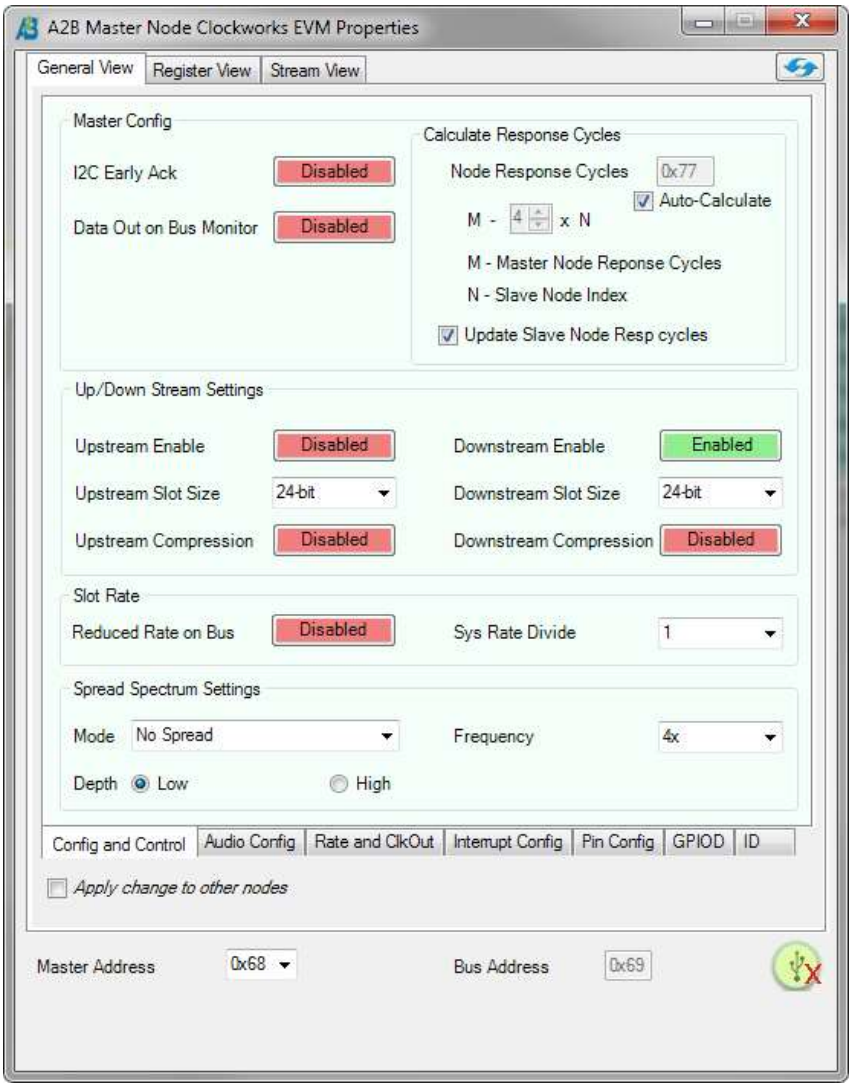


Figure 27 Root node properties

In the provided example the I²S lines between the AD2428 and the ADAU1761 are operated in TDM2 mode (more commonly called stereo). The ADAU1761 only has one I²S in and one I²S out line. On the EVM root node the ADAU1761 generates the I²S BLCK and SYNC signals; the opposite is true on an EVM slave node.

For the AD2428 this can be set directly in its properties:

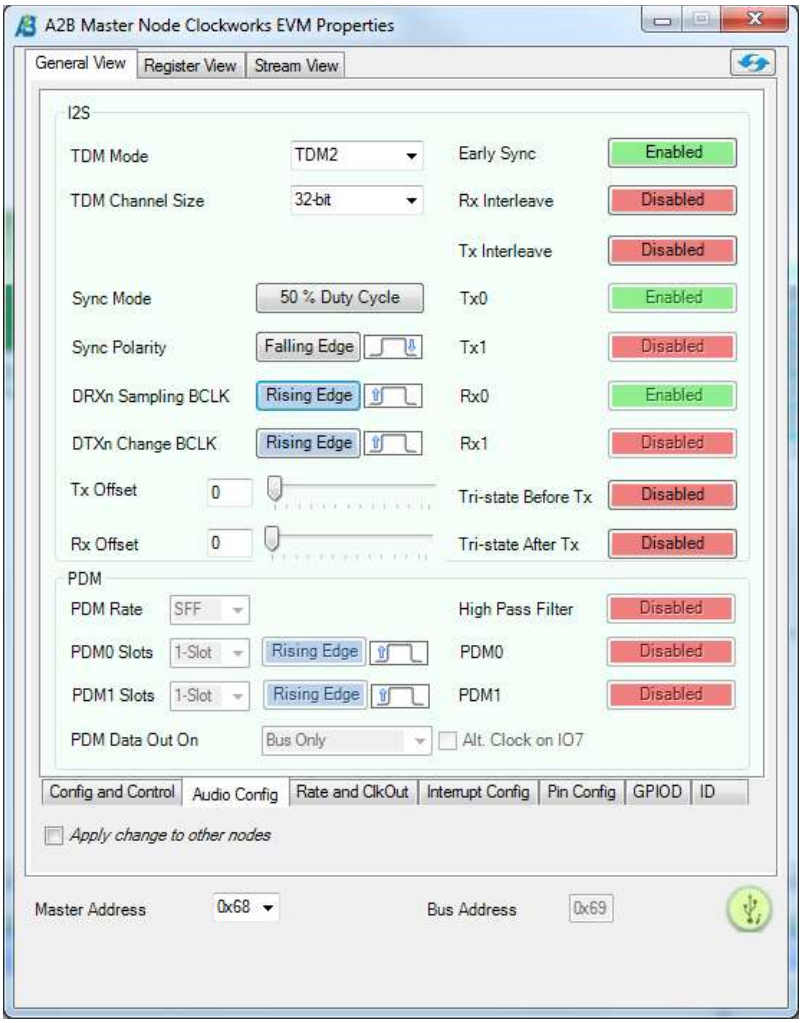


Figure 28 Root node settings for TDM2 over I²S

To set up the ADAU1761 the SigmaStudio schematic for a separate project must be updated. Do not mix up the root node's ADAU1761 SigmaStudio schematic with the client node's, they need to operate differently due to the different I²S clocking.

The provided root node ADAU1761 example is copied from one that ADI uses in their examples. Don't worry if the example(s) you have look different. You will want to edit these or create new ones to match your processing configuration.

The ability to create test tones is useful when debugging why nothing seems to be working the way you wanted.

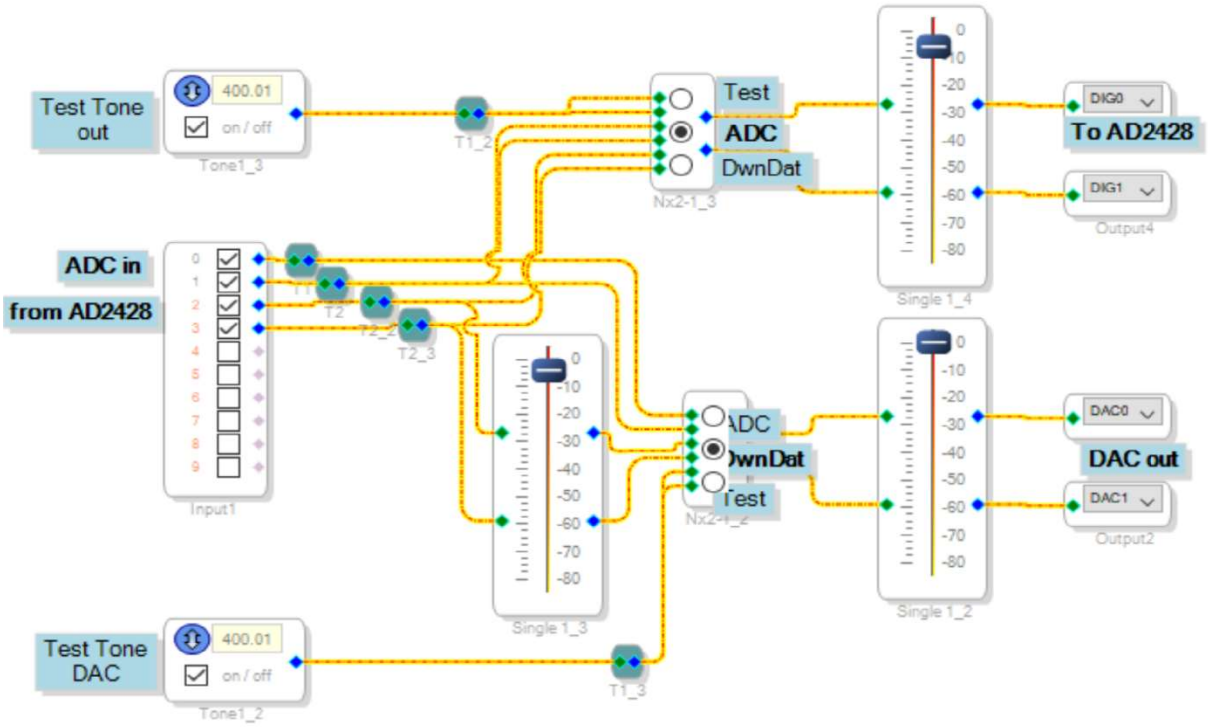


Figure 29 Root node ADAU1761 example (`CSP_EVM_a2b_master_ADAU1761.dspproj`)

The ADAU1761 is set up with the correct divider for the 12.288 MHz MCLK, which is 256 times Fs (Fs = 48 kHz).

On the root node the AD2428 uses the BCLK as an input, along with the FSYNC (or FLCK). For correct AD2428 the frame sync is the important clock, it multiples that clock up by 1024 to create the A²B network clock.

The bit clock on the root node is only used for clocking I²S data in and out of theAD2428.

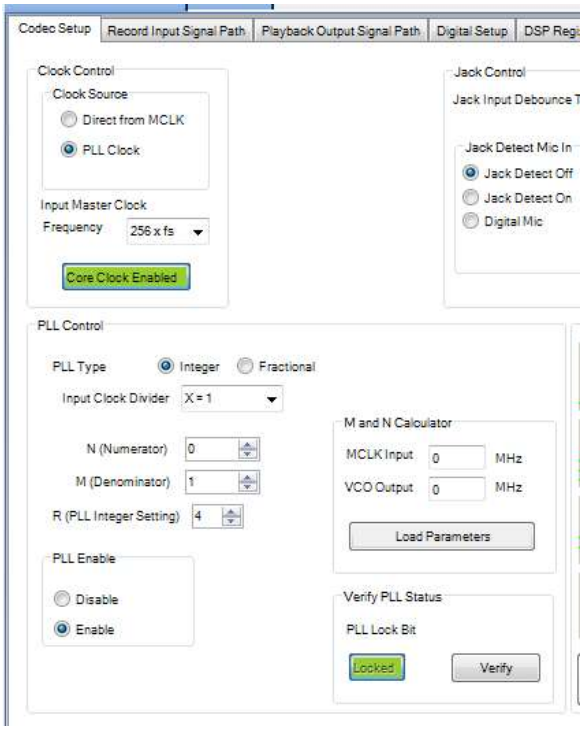


Figure 30 Root node ADAU1761 clock setup

For complete details on the ADAU1761 setup please see the datasheet and user guide. For the I²S interface setting it to Master mode is the key setting.

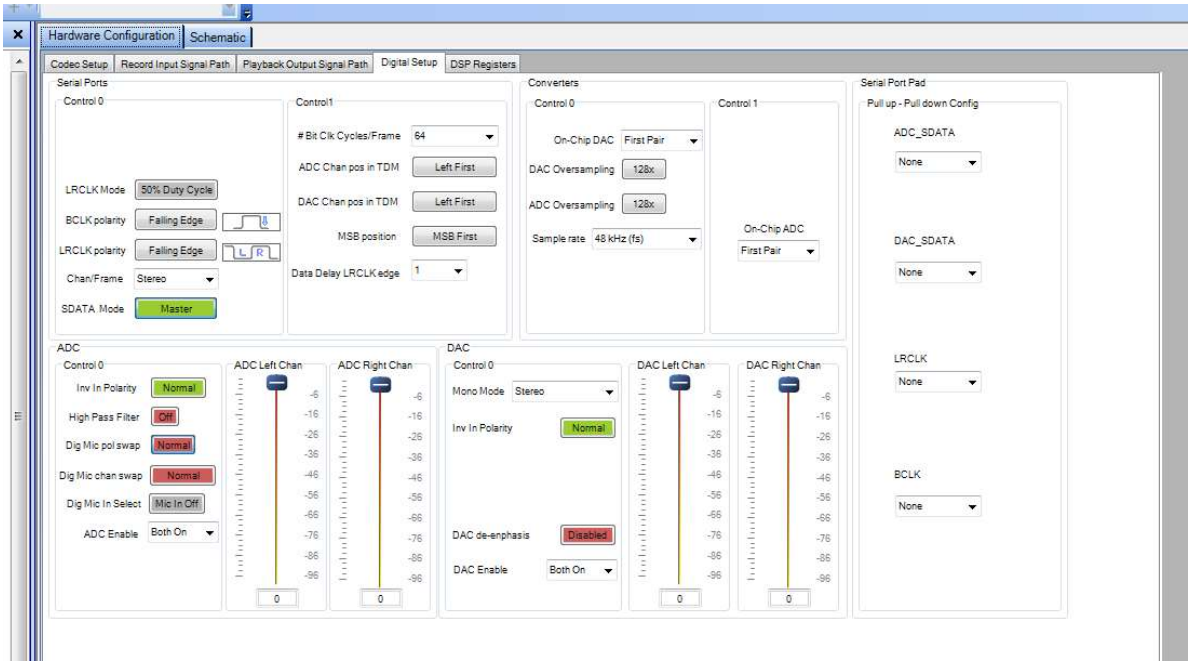


Figure 31 ADAU1761 serial port setting on EVM root node

Once the ADAU1761 has been set up the way you want you need to create the XML file that is used to program the ADAU1761 at boot up. Full details are in the SigmaStudio guide(s), but the summary is

- *Link Compile Download* (hardware doesn't need to be active)
- Copy the Capture window left side contents to the right side (hit the << button on the right if it's not visible)
- Save the right side window to an XML file
- Return to the A²B schematic and set the root node ADAU1761 to use the XML file

If you haven't already, use the Streams configuration to route audio data between the nodes in your system.

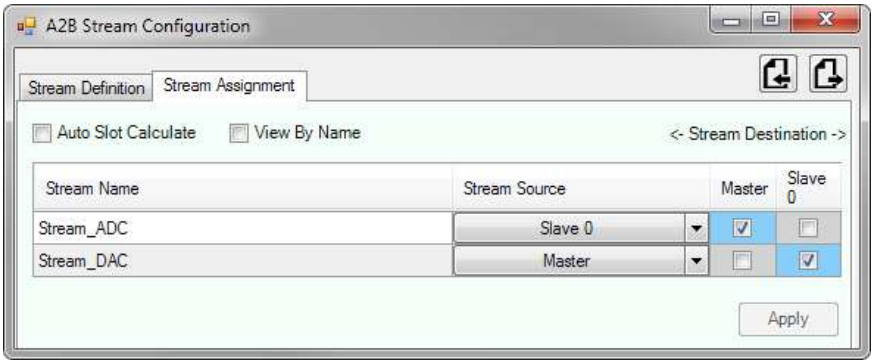


Figure 32 Simple stereo stream routing for two EVM example

At this point the A²B schematic is ready to run. If you're using the canned example you would have skipped all of the above. Hit the *Link Compile Download* and the two AD2428 nodes should turn green. If you get an error message then check that power is on to all of the boards and they are cabled correctly. If you made new diagrams then recheck your work.

7.3.1 THINGS THAT MIGHT GO WRONG

If there is a bug in your ADAU1761 SigmaStudio schematic for the root node then it probably won't create the I²S clock(s), and the root node AD2428 will be unhappy. You may want to run your ADAU1761 schematic first and probe the I²S signal to make sure you set up I²S correctly.

Once the A²B schematic is running you can open the ADAU1761 schematic and dynamically change its controls and have them reflected in the running hardware. If opening the ADAU1761 schematic for the first time you should hit the *Link Compile Connect* button.

You can also work with a client node's ADAU1761 in a similar manner to adjust its settings on the fly.

Very occasionally the USBi and/or SigmaStudio may get a little lost and not make your changes, or sometimes report it can't find the USBi. A power cycle of the A2B node and unplugging the USBi from the host PC usually clears that up, sometime restarting SigmaStudio is needed too. This does not happen very often (we see it maybe once in a 100 runs where we're changing things).

It's also been noticed that if the A²B application is running and the host PC goes to sleep it can sometimes shut down the A²B target. Exit from SigmaStudio if you want to leave things going for extended periods.

8 SHARC AUDIO MODULE (SAM) AS A²B ROOT

The SC589 based SAM is a low cost development board from Analog Devices. ADI has introduced newer SHARC based modules as well as moved support for A2B with them into the SigmaStudio+ tool.

This section is coped from the older Clockworks manual as it may be useful until better SHARC specific documentation can be developed.

While the procedures described in ADI provided A²B software package documents theoretically explain the process of using SC589 with A²B, in real life it can be a bit, hmmm, let's call it obtuse.

The diagrams and software described in this section are available from Clockworks, if you don't find them on the support site please contact us.

Your first stop should be the ADI Wiki to gather up the latest things that they have made available.

https://wiki.analog.com/resources/tools-software/sharc-audio-module

8.1 IF YOU DON'T HAVE THE A²B SOFTWARE PACKAGE…

you won't be able to actually develop A²B based applications. You can however run the precooked A²B network configurations that ADI has made available for SAM, as well as the ones that Clockworks has created.

These precooked configurations can be used with your own custom SC589 software that you develop, you are restricted in that you can't change the topology, sample rates, routing, etc.

When the SAM Bare Metal package runs it simply loads a predefined sequence of I²C commands in to the master A²B node located on the SAM. This sets up all of the topology as well as programming other I²C devices like the commonly used ADAU1761 SigmaDSP or I²C devices likes the SSM3582 Class D amplifier chip used on the quad channel SAM FIN.

Once the A²B network is running the SAM software simply reads/writes data from/to the I²S port connected to the AD2425W A²B transceiver chip. For applications that don't need the sophisticated fault and error detection that A²B offers, this may be more than adequate.

If the hardware configuration might change – which after all is the whole point of the flexibility offered by A²B – you will need the tools[9].

8.2 USING THE ADI A²B TOOLS WITH SAM – HARDWARE AND SOFTWARE SETUP

The steps are as follows for a basic SAM plus Clockworks A²B EVM setup. Audio is just exchanged between the two boards in this example.

8.2.1 HARDWARE SETUP SUMMARY

Details of the hardware and boards can be found in their respective user manuals. Three USB ports are needed, keep that in mind if your laptop is USB challenged. The tools run on Windows 7 and later.

- Connect the ICE1000 to the SAM JTAG port and your PC
- Connect the USBi to the SAM USBi port (careful, it's just a header, so verify where pin 1 is) and your PC.
- Connect a UART to USB adapter to the UART header on SAM. Normal baud rate is 115,200.
- Connect audio sources to the SAM Line in jack and the A²B EVM line in jacks.
- Connect amplified speakers to the SAM line out jack.
- Connect headphones or amplified speakers to the A²B EVM headphone out jack.

[9] Or hire developers that use the tools to provide that piece of your product's software.

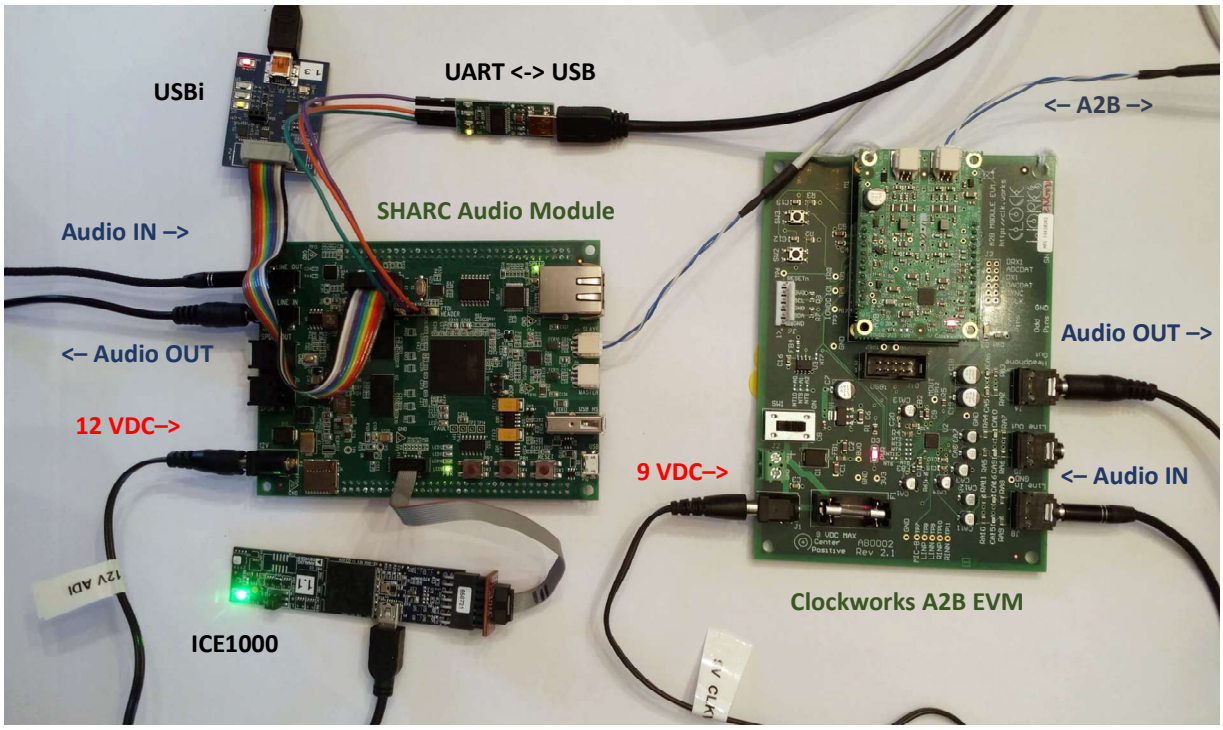


Figure 33 SAM and A2B EVM hardware setup for the prior EVM; the new one uses a 12V supply

8.2.2 SOFTWARE SETUP SUMMARY

There are a number of tools that are used in the process of creating A²B based applications. This manual is not the place to learn about them.[10]

- CCES. When used with ICE1000 and SAM the free version can be used.
- SigmaStudio
- A2B add on for SigmaStudio (including API)
- Terminal emulator for the UART connection
- SAM Bare Metal Framework
- A pre-cooked SC589 bypass program or one created from the wizard (requires editing)

[10] Are we lazy? Err, maybe. Seriously though the tools are always getting updated and if the details were copied in to here we would be forever updating this document, which would seriously cut in to the time available for sampling more IPAs.

8.3 BASIC SAM EXAMPLE WITH A2B EVM SETUP IN SIGMA STUDIO

This example exchanges audio between the two boards shown in the setup of Figure 33.

8.3.1 SIGMASTUDIO SCHEMATIC

Open the file `SAM-EVM.dsppproj` in SigmaStudio[11] or create the equivalent on a new blank sheet.[12]

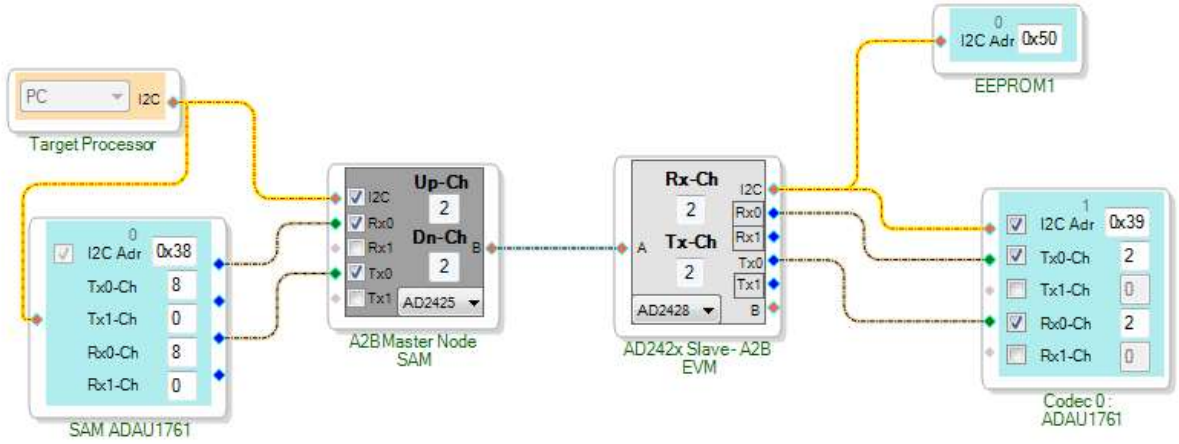
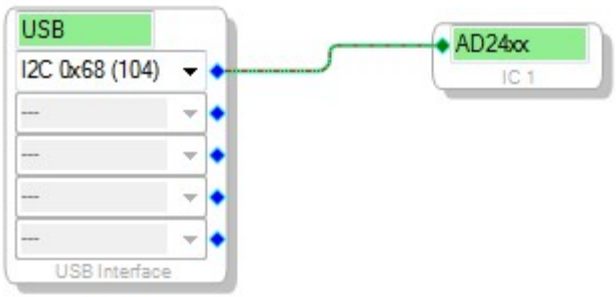


Figure 34 SAM with A2B EVM schematic

If starting from a blank schematic add the USBi connection on the Hardware Configuration tab. For SAM it looks like Figure 35.

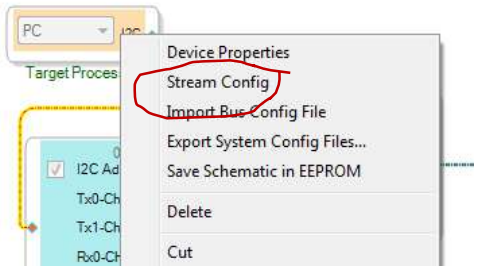


[11] A²B extensions must have been previously installed.
[12] If starting from scratch please see the information about A²B chip setup in Section 7.1.1.

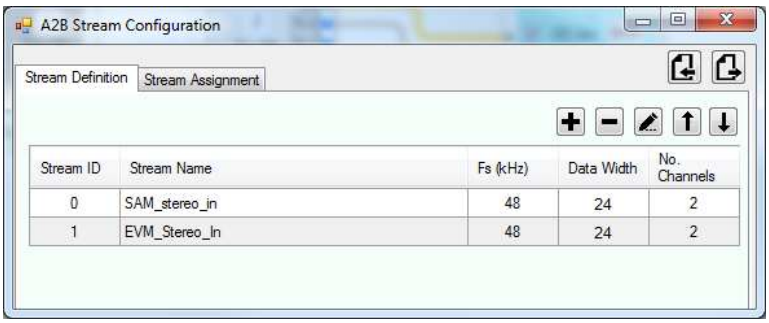
Figure 35 Sigma Studio Hardware Configuration tab for use with SAM and USBi

The Stream capability of the later chips makes it easier to set up and manage audio data than using the manual configuration of the first generation chips. These next steps explain the process; the provided example already has this set up this way.

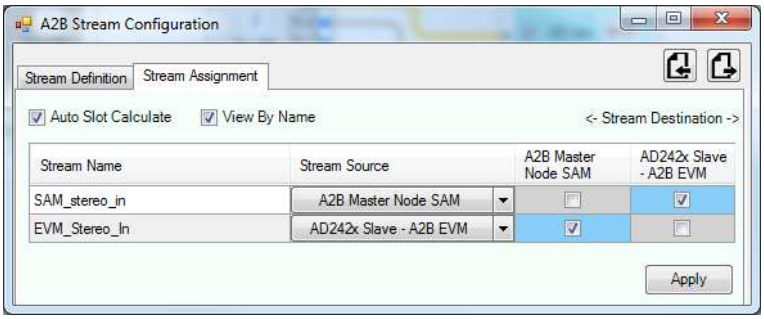
Right click on the Target processor block and select Stream Config:



Select the + button on the A2B Stream Configuration dialog and add two streams with 2 channels each:

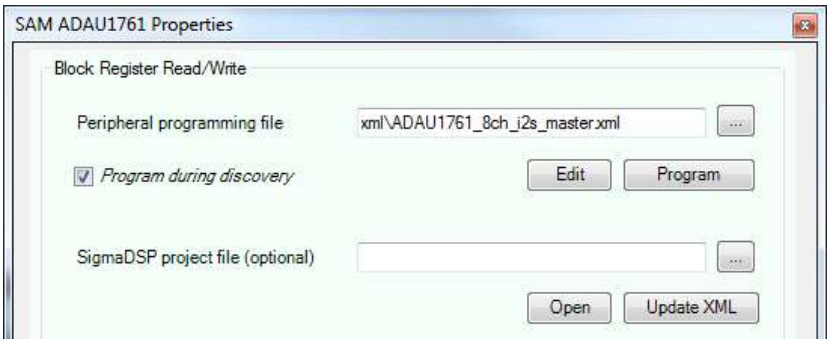


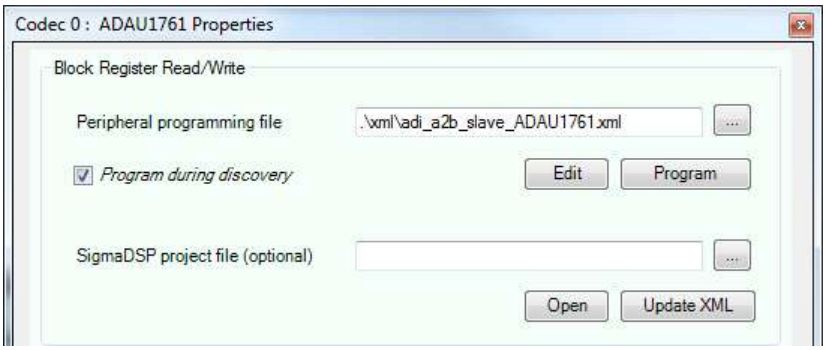
Then select the stream assignment tab:



Make sure Auto Slot Calculate is selected, and View by Name[13] is usually more informative then viewing by ID. In this example we're swapping audio between the two nodes.

Next set up the programs that get run on the two ADAu1761 in the system (one on SAM, one on EVM). They run different programs. The process of creating the `.xml` files from SigmaStudio is covered in the SigmaStudio documentation as well as in the A²B specific documents in the A²B add-on for SigmaStudio. The latter document also explains running and debugging ADAU1761 programs from within this setup.





If building an example from scratch check the I2S related settings for the EVM as described in Section 7.1.1.

Before doing the Compile Download & Run you must first load a pass through program on the SC589.

8.3.2 SAM SC589 BYPASS PROGRAM

There are some ADI provided examples that can be used for this step, or the SAM Wizard can be used to generate a basic pass through program.

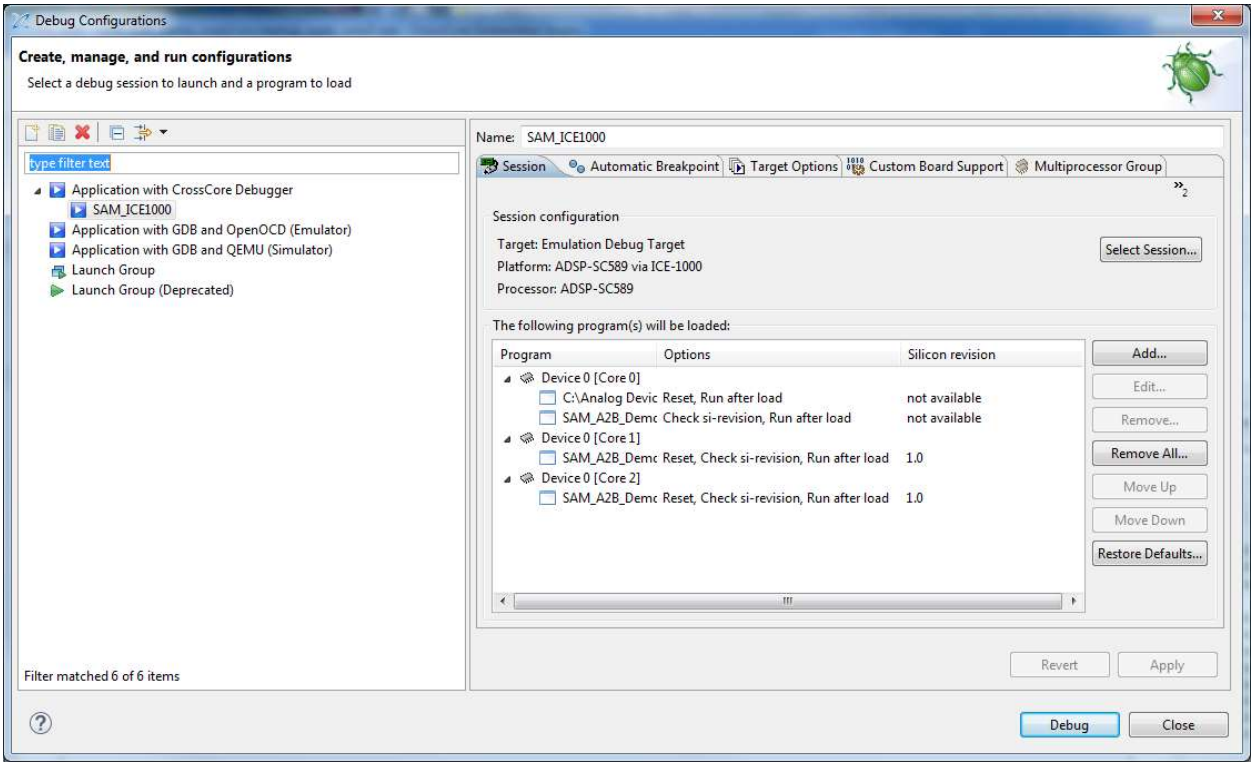
[13] If the names of the AD24xx nodes are changed it's been noticed that this dialog can be a little slow to refresh the names. Clicking on a Stream Source seems to sync it back up.

To edit the bare metal framework program, open up `common/audio_system_config.h` and make these changes[14]:

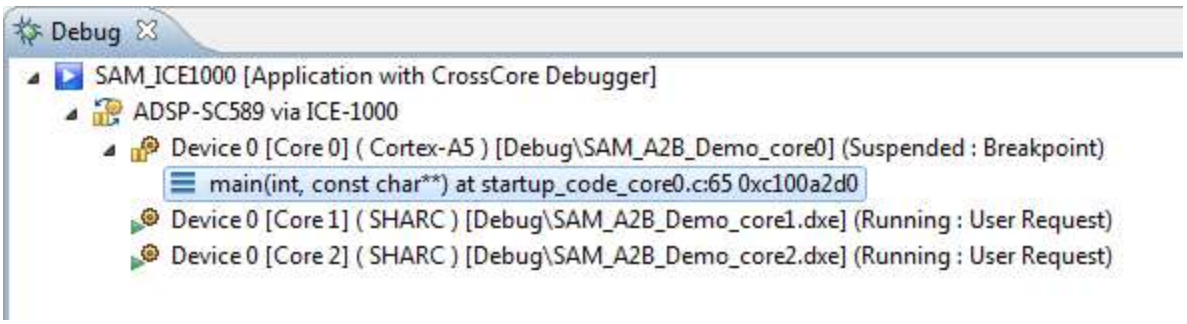
```
/*****************************************************************************
 * 3. Select an audio processing framework to use (only select one)
 *****************************************************************************/

// Standard audio processing framework (SHARC Audio Module
// and SHARC Audio Module + Audio Project Fin)
#define AUDIO_FRAMEWORK_8CH_SAM_AND_AUDIOPROJ_FIN          FALSE

// Audio processing framework for use with the automotive daughter board
#define AUDIO_FRAMEWORK_16CH_SAM_AND_AUTOMOTIVE_FIN        FALSE

// Bypasses the ADSP-SC589 so I2S signals routed directly between ADAU1761 and
// A2B controller (GPIO4 = LED on this board)
#define AUDIO_FRAMEWORK_A2B_BYPASS_SC589                   TRUE   <--- If you're using SS, set this
to TRUE to bypass the SHARC

/*****************************************************************************
 * 4. Select whether or not to enable A2B in the framework
 *****************************************************************************/

#define ENABLE_A2B                                         TRUE

#if (ENABLE_A2B)
/**
 * If A2B is enabled, select the role that this SHARC Audio Module board
 * will play (TRUE = master node, FALSE = slave node)
 */
    #define A2B_ROLE_MASTER                                TRUE   <--- If this SAM board is a slave
node on A2B bus, set to FALSE
```

If importing an existing project in to a CCES workspace verify that the Debug configuration is set up correctly[15].

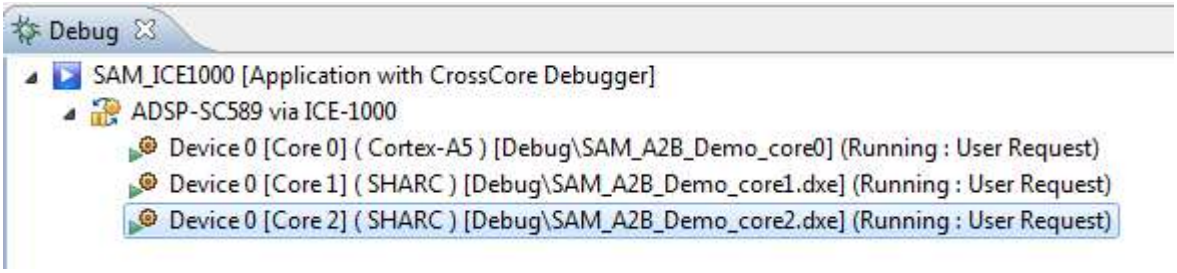
[14] Thanks go to Dan Ledger at https://www.pathcollaborative.com/ for these code snippets.
[15] We skipped a lot of steps here, see the tutorials on SAM and CCES if this looked like magic to you. The good news is CCES is more or less standard Eclipse. The bad news is CCES is more or less standard Eclipse.



After you select Debug the code will be compiled (if needed) and if running with the stock CCES behaviors the Debug perspective will open, followed by the code loading on the three SC589 cores and then each stopping at the entry breakpoint:



For each core select it and hit the run button, after doing that will all 3 it will look like:



And depending on the code being run the ADI framework startup messages will be displayed:

```
00:00:00.000 [INFO - ARM]  Baremetal Framework (version 2.0.0) for the ADI SHARC Audio Module

00:00:00.000 [INFO - ARM]  System Configuration:

00:00:00.000 [INFO - ARM]    Processor cores running at 450.00 MHz

00:00:00.000 [INFO - ARM]    Audio sample rate set to 48.00 KHz

00:00:00.000 [INFO - ARM]    Audio block size (per channel) set to 32 samples / frame

00:00:00.000 [INFO - ARM]  Configuring the SRU - ADAU1761 is the I2S master

00:00:00.000 [INFO - ARM]  Configuring the ADAU1761
```

The SC589 and the USBi both want to be I²C bus masters. Therefor the pass through program should not attempt I²C transactions as it will lead to something going south.

The system is now ready to have A²B operation be controlled from Sigma Studio over the USBi connection.

8.3.3 CONTROLLING FROM SIGMA STUDIO

Turn down the output level on your audio output device as a mistake in setting can result in full scale output noise. Set the source output level to around 50%, assuming some sane line level source is being used.

Select the Compile Link Download button in SigmaStudio. All of the AD242x node blocks should turn green. If you get an error message recheck your USBi setup, A²B wiring, and that the boards have power.

SigmaStudio should indicate its communications with the SAM board with a few messages/second. If that's happening but you get no audio confirm that the CCES session is still active[16] as if the SC589 isn't running then audio data will not be moved around.

When in doubt power cycle everything and start over. Very occasionally the USBi will go off in to the weeds and need to be unplugged from the PC and plugged back in. When in doubt check Device Manager for the USBi device (in the USB section).[17]

Bring up your audio output device volume. With two audio sources and two audio monitoring devices you should hear both sources. If one or both is highly distorted review the I²S settings as discussed in Section 7.1.1. If you hear clipping on loud parts your source is most likely overloading the inputs, turn down the audio source output level.

At this point you've now verified that your A²B topology works.

8.4 CREATE A PROGRAM THAT SETS UP THE A2B OPERATION THE SAME AS A SIGMASTUDIO SCHEMATIC.

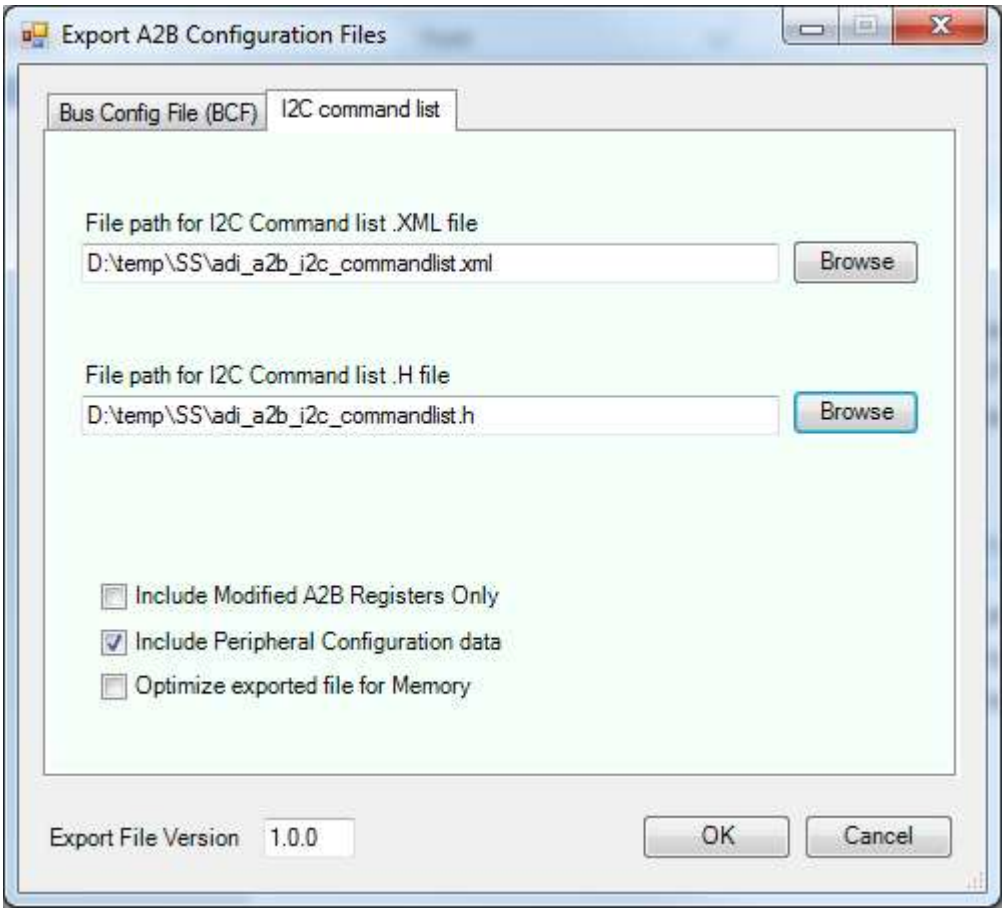
At some point in the far future, but before our Sun becomes a red giant, the SAM Project Wizard will have an option to generate a Bare Metal Framework project that allows direct selection of the Clockworks A²B EVM board as part of the A²B network. Until then though you'll need to slog through the details of all of this.

8.4.1 EXPORT FROM SIGMASTUDIO

Right click on the TargetProcessor icon and select the export option and the I²C tab. Fill in the target file locations; this can be pretty much anywhere as you'll copy them in to a CCES created Bare Metal Project Wizard created project. Select the Include Peripheral Data checkbox.

The dialog is a little deceiving, it needs a complete path and filename, not just the path (you'll get an access denied or similar message if you just provide a path). Best way to avoid issues is to use the browse button.

[16] Actually there's no real easy way to know this as you can quit CCES and the SC589 would still keep running. If the loaded program is printing to the console or flashing LEDS then that would be a clue its running. When in doubt restart. We should note we've never seen the SAM quit – it's more like a "oops I unplugged it for a second without realizing it" type thing.
[17] The appearance in device manager is the same if you use the USBi clone available from SushiBits on Tindie. Both ADI's USBi and the clone via Tindie were used when this manual was developed.



9 SIGMASTUDIO+ OPERATION WITH THE EVM

When the AB0003 module, which is based on the AD2437, is used then SigmaStudio+ with the A²B add on is required for development.

Generally the same steps and setting as described for SigmaStudio and the AD2428 apply to SigmaStudio+ and the AD2437.

One important difference is the AD2437 I/O assignment and a chip errata that affects MCLK output on the ADR2 pin. There is an unexpected interaction between SPI settings and that pin.

9.1 TWO AB0331 EVM EXAMPLE

This can be found in the file EVM_EVM.ssprj in the AD2437 based examples from the SigmaStudio+ downloads on the product page.

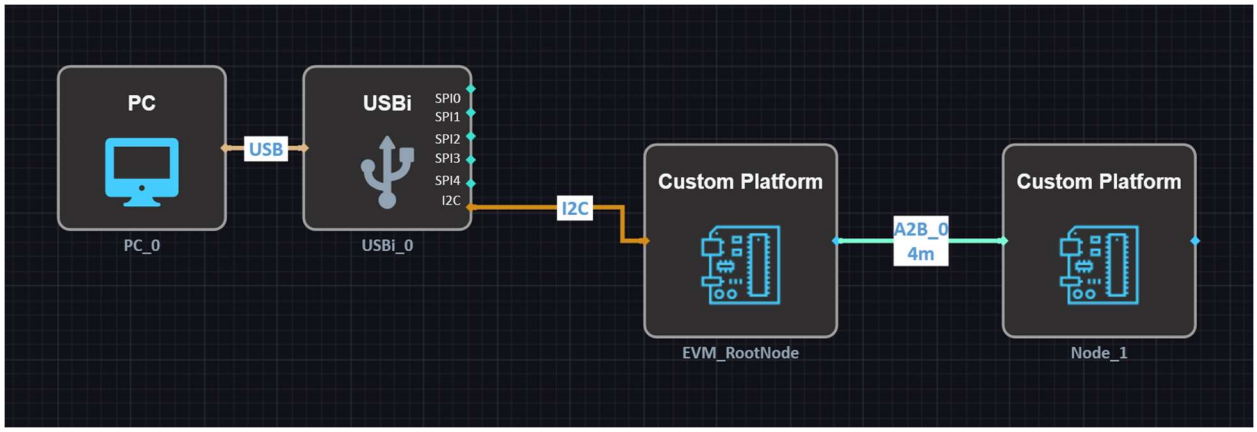
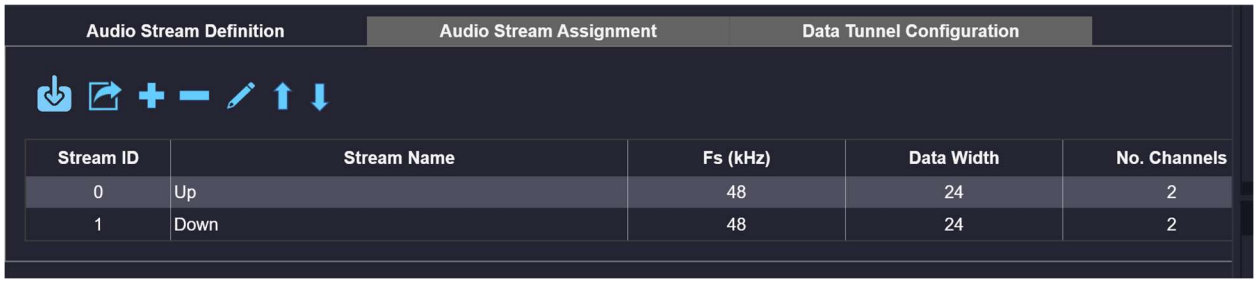


Figure 36 Simple two EVM example

This example sends stereo audio between the two boards.



Stream ID	Stream Name	Fs (kHz)	Data Width	No. Channels
0	Up	48	24	2
1	Down	48	24	2

Figure 37 Stream definition settings

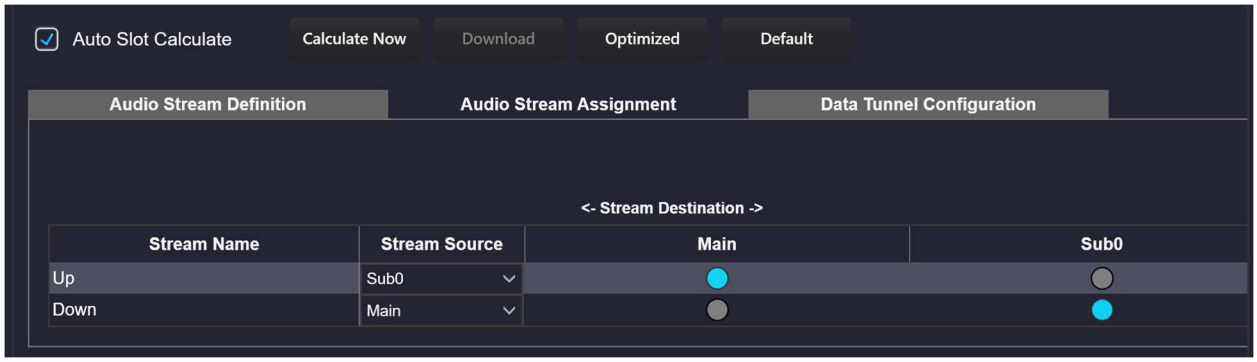


Figure 38 Stream routing

If modifying or creating your own streams make sure the *Auto Slot Calculate* is set.

One difference between AD2428 and AD2437 is the I/O mapping.

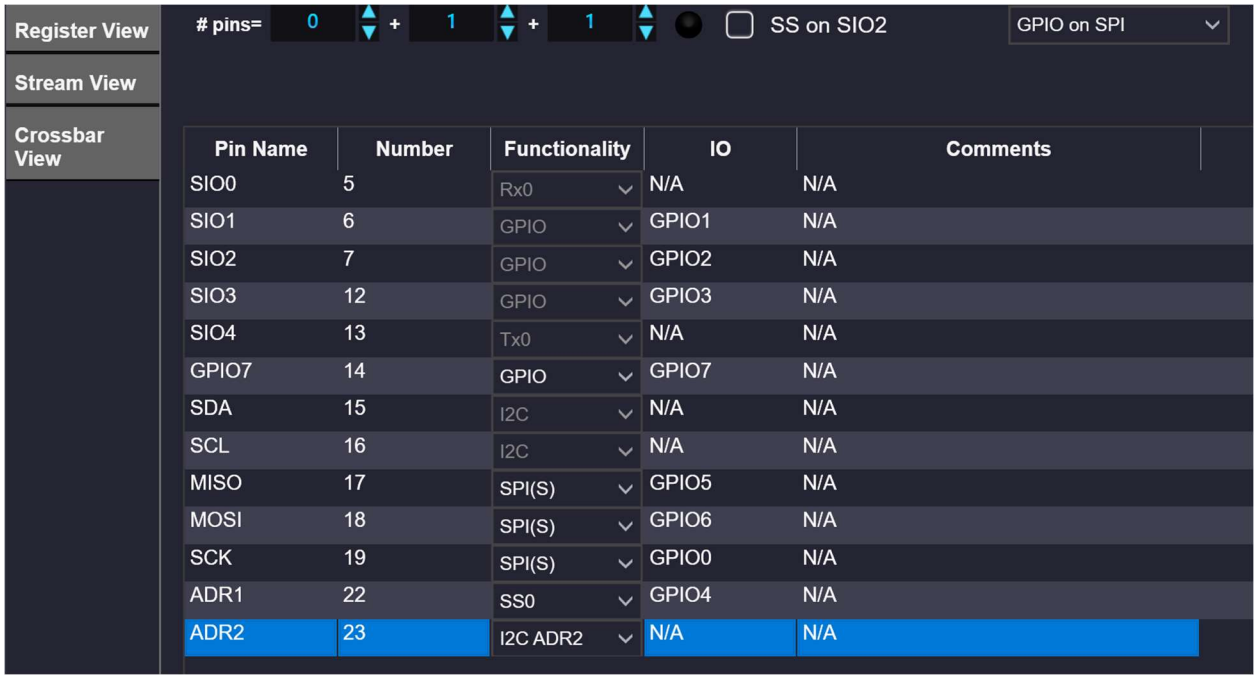


Figure 39 root node I/O setup

MCLK on the root node is created by a local oscillator so the AD2437 clock output on ADR2 can be left off.

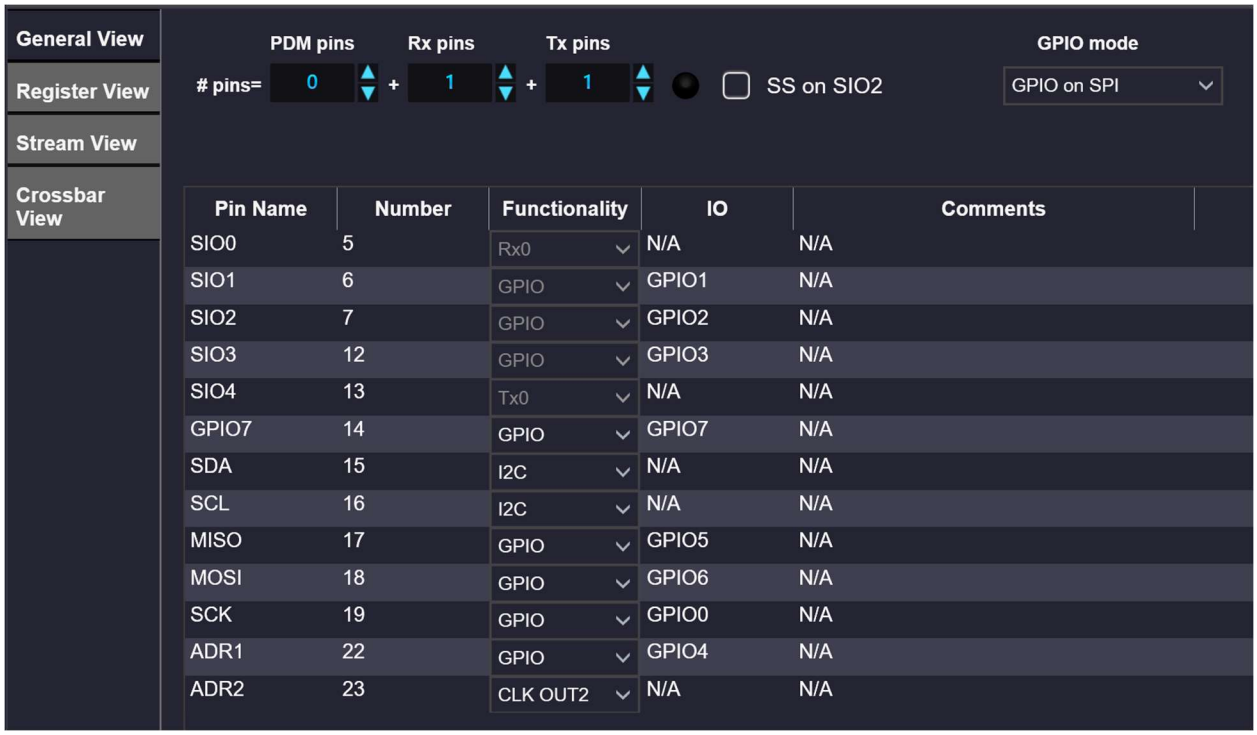


Figure 40 I/O mapping for client node. Please check the chip errata for SPI settings

The I²S settings are the same as the ones described in the SigmaStudio 4.7 example in the prior chapter; the AD2437 can have more I2S lines so the dialog is slightly different than the AD2428 one.

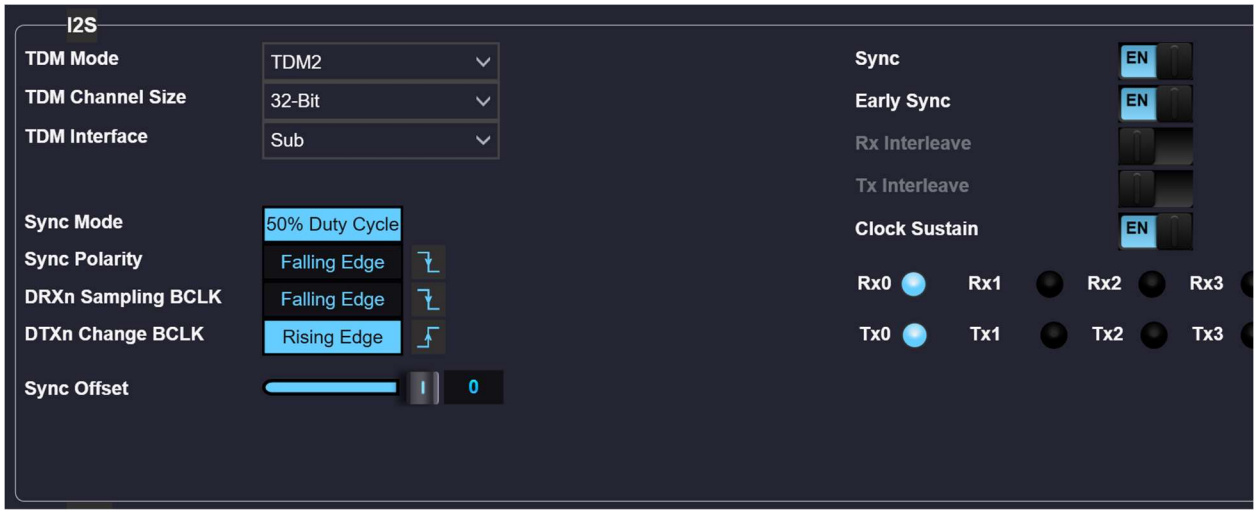


Figure 41 Client node I2S settings

The I²S settings assume the ADAU1761 settings that are defined by the .xml file included with this example.