ADAV4601 System Design Document Rev. 1

INTRODUCTION

The purpose of this document is to help with the design in of the ADAV4601 into a system. The document is detailed as follows

- Section 1 – Getting the Evaluation Board Up and Running
  Outlines the ADAV4601 evaluation board. Gives a brief description of the ways in which the evaluation board and the ADAV4601 can be configured. This evaluation board can be used as a template for the design of the ADAV4601 into a system.

- Section 2 – Powering up the ADAV4601
  Details a specially designed power up sequence for the ADAV4601 which allows the part to be powered up and down with no pops or clicks. This power up sequence should be used when the ADAV4601 has been designed into a system.

- Section 3 – Programming the ADAV4601
  Details the ways in which the ADAV4601 can be programmed and controlled. This section details how the user can control the default audio flow on the ADAV4601 or create a new audio flow from scratch SigmaStudio.

- Section 4 – The Default Flow

- Section 5 – Interfacing the ADAV4601 to a System

- Section 6 – The Application Layer Creation Software
  Only used if the user decides to create their own customized audio flow. The Application Layer allows the user to define their own register map for the audio flow they have created in SigmaStudio; in which they can define controls for filter cut-offs, volumes, third party algorithms etc.

- Appendix A – Detailed Register Descriptions.

- Appendix B – Layout Recommendations.

- Appendix C – ADAV4601 Bill of Materials.

- Appendix D – Schematics

SECTION 1 – GETTING THE EVALUATION BOARD UP AND RUNNING

GENERAL BOARD OVERVIEW

The purpose of this section is to provide a description of the ADAV4601 evaluation board and the various ways that this can be configured in order for the customer to fully evaluate the ADAV4601. This document describes the various sections on the ADAV4601 evaluation board and also the ways in which data can be passed to and taken from the ADAV4601 as well as the several ways in which the device can be clocked. Figure 1 indicates the various sections on the evaluation board.

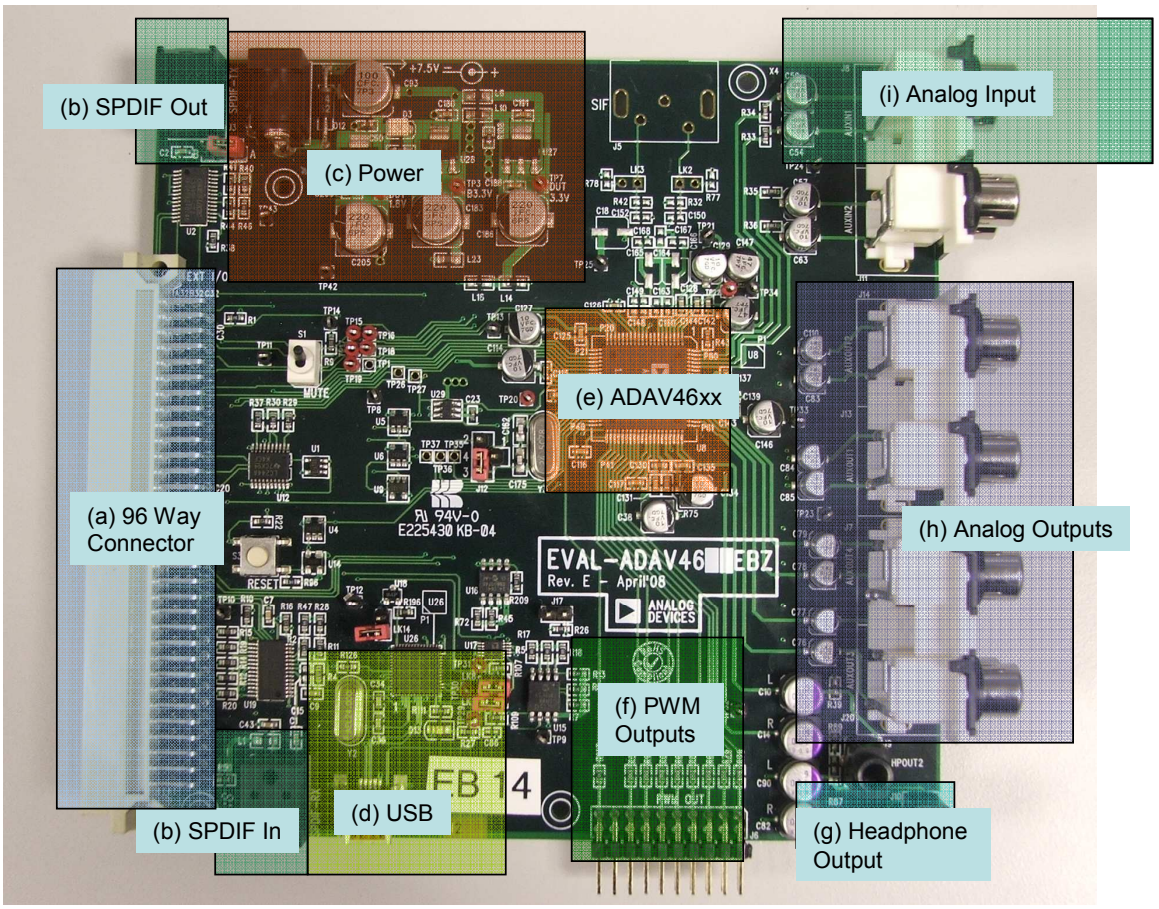


Figure 1: Board Description

The ADAV4601 audio processor deals with mixed signals therefore the board is divided into two separate sections, the left side of the eval board mainly to do with the digital inputs and outputs to the part, the right side of the board dealing mainly with the analog domain. Analog inputs and outputs are handled by way of the Phono connectors. Figure 1 indicating the arrangement of these connectors. Digital inputs and outputs can be handled in several ways on the eval board.

On the left side of the eval board there is a 96-way connector which can be used to interface to other evaluation boards. It is possible to pass digital signals from another board, such as an external MCLK (for clocking the ADAV4601) and also synchronous or asynchronous digital I2S data and framing signals. It is also possible to pass digital signals into the ADAV4601 by way of the SPDIF connectors at the side of the evaluation board.

EVALUATION BOARD DESCRIPTION

The Evaluation board allows the flexible testing of the ADAV4601 audio processor. Both analog and digital I/O can be handled. The various connectors are listed below.

Table 1 Evaluation Board Connectors

Connector Name	Function
J2	7.5V DC Power supply
J3	96-way connector
J6	PWM Outputs
J7	AUXOUT4 Left and Right
J8	AUXIN1 Left and Right
J10	Headphone 1 Output
J12	3 Way Jumper, ADAV4601 Clock Select
J13	AUXOUT1 Left and Right
J18	USB Mini Connector
J20	AUXOUT3 Left and Right

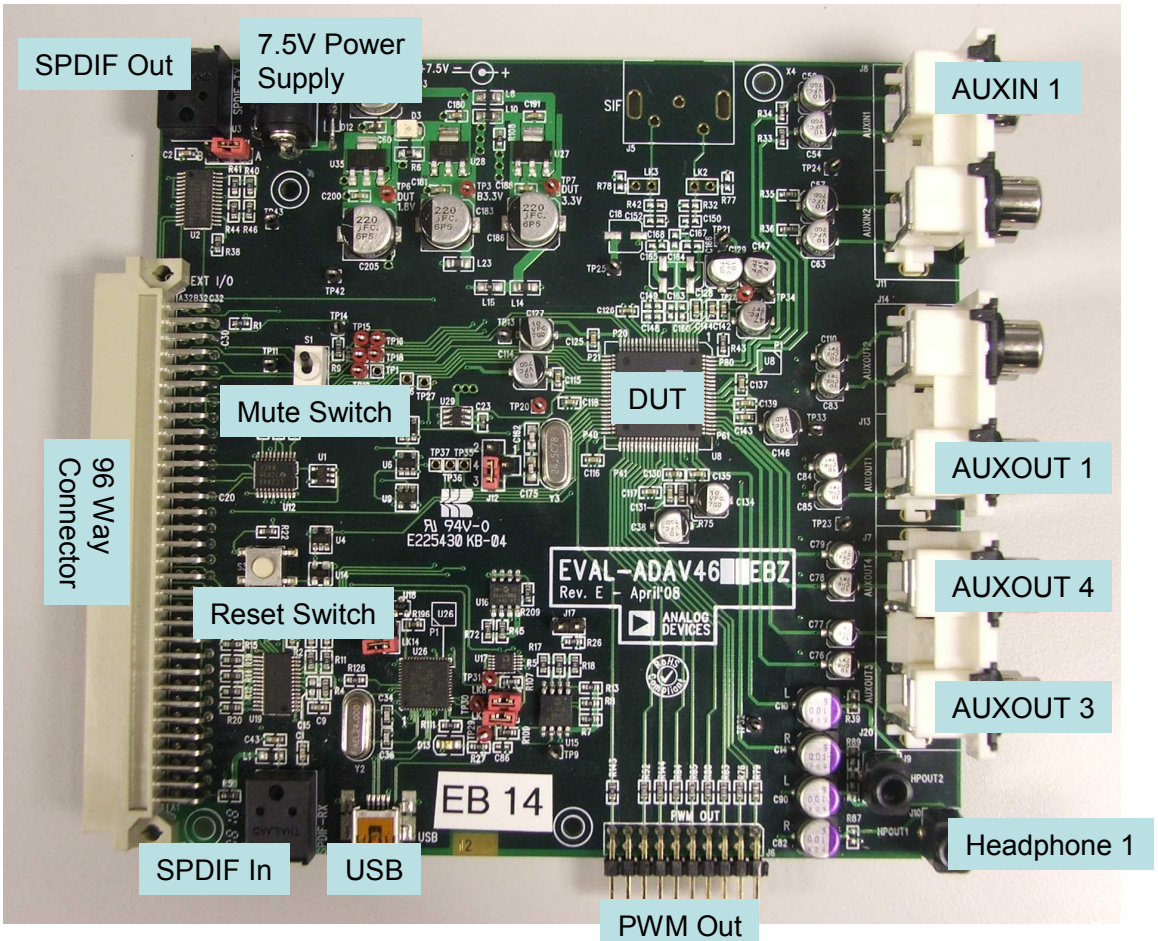


Figure 2: ADAV4601 Rev E Evaluation Board

ADAV4601 EVALUATION BOARD QUICK START

1. Apply power to the board using the included 7.5V Power Supply via connector J2.
2. By default the board is set to use crystal Y3 as clock source. This means jumper J12 should be in position 3-4. If you wish to use a different clock source, please see the section on Providing a Clock Source for the ADAV4601.
3. Connect the Evaluation board to the PC using the provided SigmaStudio USB Adaptor board; refer to Section 3 – Programming the ADAV4601 for more details on programming the ADAV4601.
4. Connect any audio sources and listening/measurement equipment that may be required. Details on the I/O for the board are provided in the Evaluation Board I/O section.
5. For details on powering up the board, and interfacing with the processor, please refer to the Section 2 – Powering up the ADAV4601, this details the register writes required to power up and power down the ADAV4601 evaluation board.

PROVIDING A CLOCK SOURCE FOR THE ADAV4601

The ADAV4601 contains a phase-locked loop (PLL) that generates all of the internal clocks required by the ADAV4601. It is possible to set the master clock (MCLK) frequency to be 64*Fs, 128*Fs, 256*Fs or 512*Fs, where Fs is the sampling frequency and is set at 48 kHz. The ADAV4601 Evaluation Board therefore requires a clock source and there are a number of ways in which to provide this.

a) The clock for the ADAV4601 device can be provided using the crystal Y3. The evaluation board is set to use this clock source by default. Jumper J12 is set in the 4-3 position.
b) The clock for the ADAV4601 can be provided using the 96-way connector. This can be set in one of two ways. The clock will come from the MCLK pin on the 96-way connector (J3-C2), by default the MCLK_OUT function is output on this line so the master slave select must be set here. Jumper J12 is set in the 1-4 position.
c) The third way of clocking the ADAV4601 using the 96-way connector is to use the BCLK1 input. Jumper J12 is set in the 2-4 position.

Table 2: Evaluation Board Clock Modes

Clock Source	Jumper J12 Position
Crystal Y3	4-3
96-Way Connector (MCLK)	1-4
96-Way Connector (BCLK1)	2-4

CLOCKING THE ADAV4601 USING THE ONBOARD CRYSTAL (Y3)

a) To enable the onboard crystal (Y3) to clock the ADAV4601, jumper 12 (J12) must be set to position 4-3. If the user wants to use digital data from the SPDIF receiver in this mode, the data will not be synchronous to the master clock and is therefore handled by the SRC1 of the ADAV4601. The appropriate register writes must be performed to power up and enable the SRCs so as to ensure the correct operation of the device.

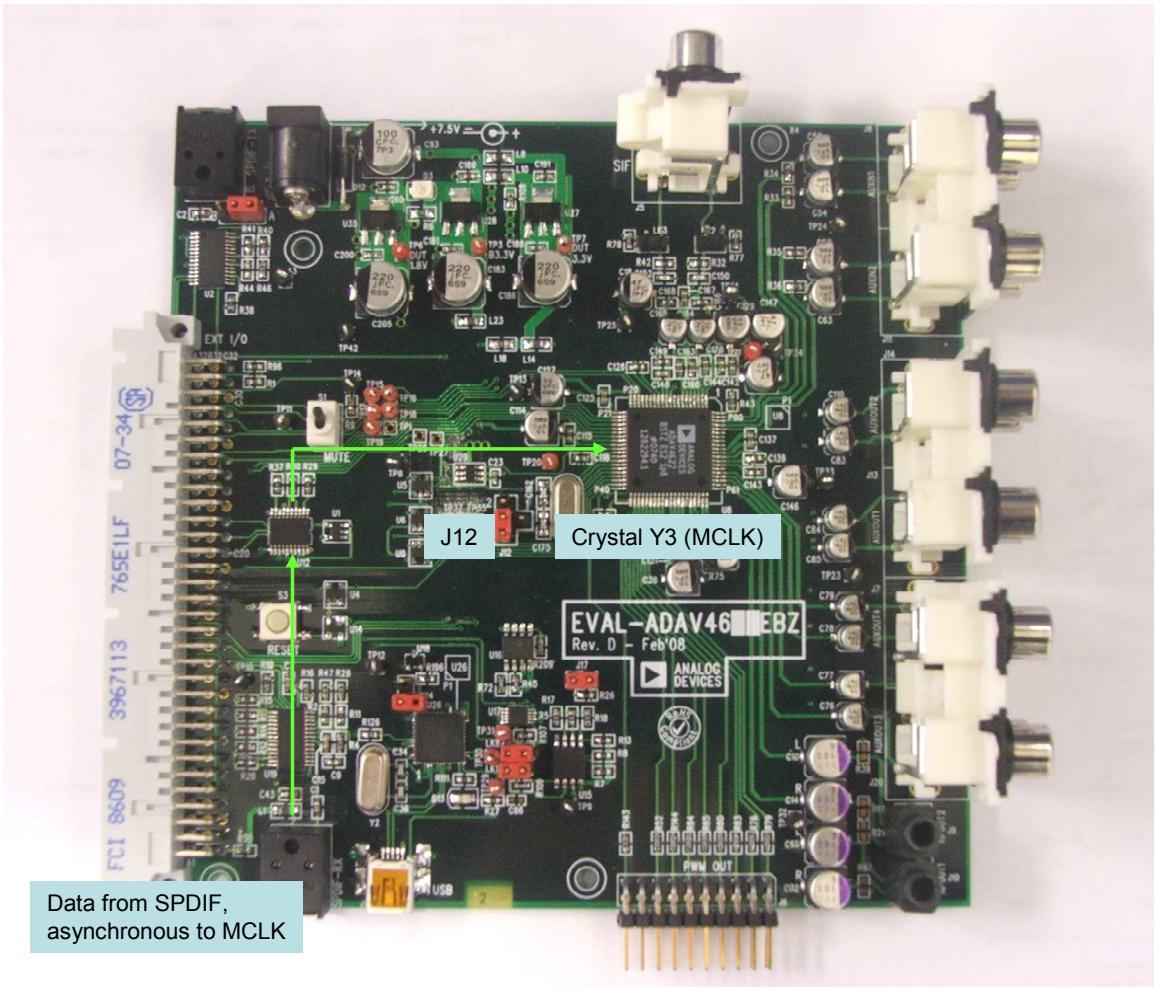


Figure 3. Clocking using the crystal (Y3)

Once the board has been set up to use crystal (Y3) as a clock source the appropriate clock frequency must be programmed in the ADAV4601. This crystal has a clock frequency of 24.576 MHz and therefore the input MCLK frequency to the ADAV4601 must be set to 512*Fs. This can be done by writing directly to the initialisation register in the ADAV4601 using SigmaStudio, please refer to the Appendix A – Detailed Register Descriptionsfor more details on register settings

Note: The ADAV4601 evaluation board is shipped with scripts for controlling the default flow. These scripts include the correct writes for powering up the ADAV4601 and can be tailored using the programming guide to suit the customer's needs.

CLOCKING THE ADAV4601 USING THE 96-WAY CONNECTOR (MCLK)

b) To clock the ADAV4601 using the MCLK from the 96-way connector, jumper 12 (J12) must be set to position 1-4. Data from the 96-way connector in this mode can be synchronous or asynchronous, if there is data is from the SPDIF connector this is asynchronous and can be handled using the SRC1. The appropriate register writes must be performed to power up and enable the SRCs so as to ensure the correct operation of the device.

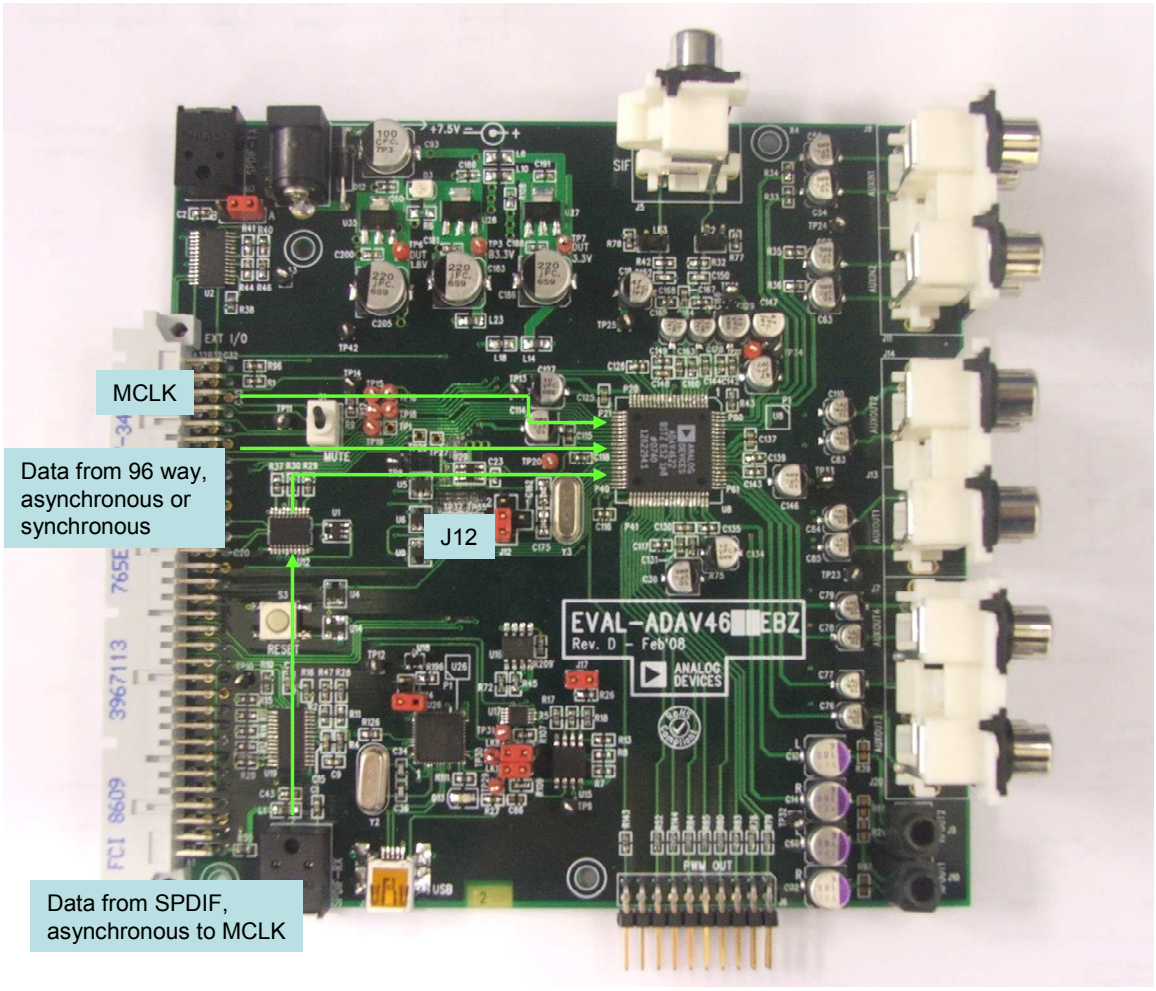


Figure 4. Clocking using the 96-way connector (MCLK)

Once the board has been set up to use the 96-way connector the appropriate clock frequency must be programmed in the ADAV4601. This can be done by writing directly to the initialisation register in the ADAV4601 using SigmaStudio, please refer to the Appendix A – Detailed Register Descriptionsfor more details on register settings.

CLOCKING THE ADAV4601 USING THE 96-WAY CONNECTOR (BCLK1)

c) To clock the ADAV4601 using the BCLK1 from the 96 way connector, jumper 12 (J12) must be set to position 2-4. Data from the 96 way connector in this mode can be synchronous or asynchronous, if there is data is from the SPDIF connector this is asynchronous and can be handled using the SRC1. The appropriate register writes must be performed to power up and enable the SRCs so as to ensure the correct operation of the device.

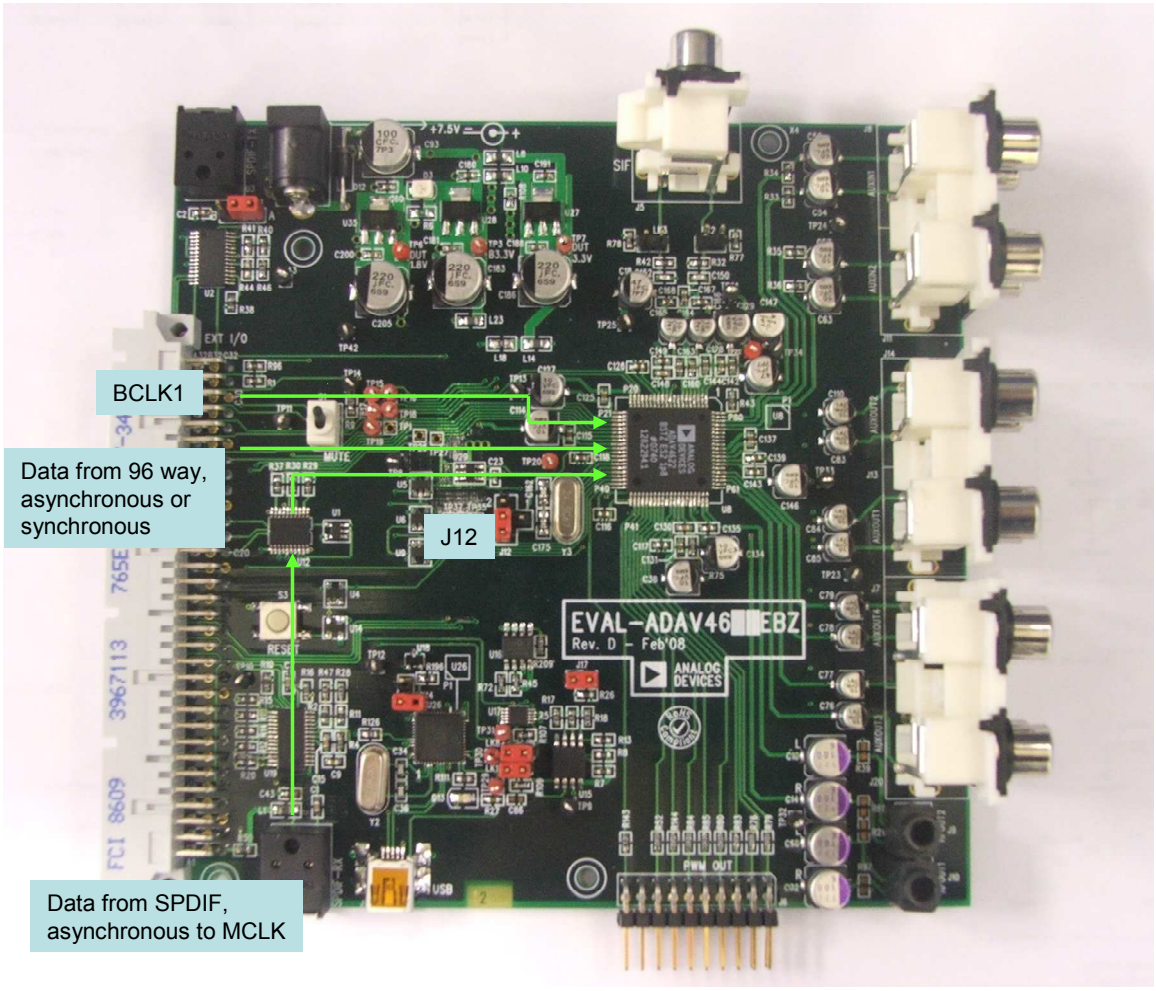


Figure 5. Clocking using the 96-way connector (BCLK1)

Once the board has been set up to use the 96-way connector the appropriate clock frequency must be programmed in the ADAV4601. This can be done by writing directly to the initialization register in the ADAV4601 using SigmaStudio, please refer to the Appendix A – Detailed Register Descriptions for more details on register settings.

EVALUATION BOARD I/O

The Evaluation board provides full I/O for the ADAV4601, including Analog I/O and digital I/O (I2S).

ANALOG I/O

The ADAV4601 evaluation board has 1 line level input, AUXIN1. This input is configured for a 2Vrms maximum input. The three analog outputs (AUXOUT1, AUXOUT3 and AUXOUT4) are line level outputs with a 1Vrms maximum output.

HEADPHONE OUTPUT

The ADAV4601 features 1 stereo headphone out, HPOUT1, which is available on connector J10. This output has an integrated headphone amplifier which is capable of driving a 16ohm load. There is no load on this headphone output by default. The headphone output is only loaded with the inclusion of a headphone jack.

SPDIF I/O

The ADAV4601 Evaluation board has an SPDIF optical receiver (U7), and an SPDIF optical transmitter (U3).

SPDIF RECEIVER

The ADAV4601 does not decode SPDIF streams, so the SPDIF receiver U19 is used to convert the SPDIF stream from the TOSLINK connector (U7) into an I2S digital data stream along with the appropriate clock and framing signals.

When valid data is decoded by the SPDIF receiver, the data is automatically passed through to the SRC1 of the ADAV4601 and is handled as asynchronous data. The reason this data is handled by the SRC is due to the fact that the received data is not synchronised to the MCLK of the ADAV4601 and must be re-synchronised to the internal processor.

Please refer to the Appendix A – Detailed Register Descriptionsfor more details on register settings.

SPDIF TRANSMITTER

The SPDIF transmitter U3 can be fed from one of two sources. The ADAV4601 has both a standard I2S digital output which is SPDIF encoded by the SPDIF encoder U2. The ADAV4601 also has a separate SPDIF output. Either of these SPDIF streams can be output on U3, and selection of the stream is controlled by the link LK11. When in position A, the digital I2S data from the ADAV4601 is converted to an SPDIF stream and output on the SPDIF connector U3. In position B, the SPDIF data transmitted is straight from the ADAV4601.

Note: The SPDIF encoder (U2) requires both a LRCLK and BCLK framing signals for the I2S data as well as an OMCK. The evaluation board is designed to get these signals from the ADAV4601; however the appropriate register writes must be performed to ensure that the SPDIF encoder operates correctly. If the LRCLK1 and BCLK1 are not supplied by the user, the ADAV4601 must be set in master mode which means it can drive the LRCLK1 and BCLK1 lines. Also the MCLK_OUT from the ADAV4601 must be enabled so as to provide the OMCK for the SPDIF receiver. These register writes are provided in the ADAV4601 Power Up script in the default scripts directory. Please refer to the Appendix A – Detailed Register Descriptions for more details on register settings.

PWM OUTPUTS

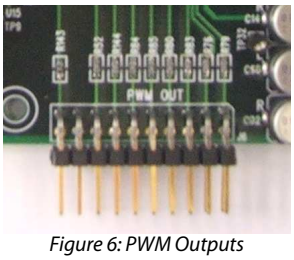


Figure 6: PWM Outputs

The differential PWM outputs from the ADAV4601 are available on header J6 for connection to a Class-D power stage.

Table 3: PWM OUT Signals on J6

Pin	Signal
1	PWM4B
3	PWM4A
5	PWM3B
7	PWM3A
9	PWM2B
11	PWM2A
13	PWM1B
15	PWM1A
19	PWM Ready

All other pins are grounded. Please refer to Appendix A – Detailed Register Descriptionsfor more details on register settings.

LINK/JUMPER CONFIGURATIONS

The links (jumpers) present on the evaluation board are predominantly for in-house evaluation. Table 4below briefly describes each link.

Table 4: Jumper/Link Configurations

Link	Function with jumper in place	Comment
LK14	The USB microcontroller U26 is programmable. Should be Out by default.	ADI Evaluation feature only
J17	The EEPROM U15 is writeable.	ADI Evaluation feature only
LK11	Selects if SPDIF from the ADAV4601 or it the converted I2S data from the part is outputted on the SPDIF transmitter	

TESTPOINTS

Testpoints are distributed on the evaluation board to facilitate the examination of signals. In general a black testpoint should reference a ground signal, while a red testpoint should reference other (non-ground) signals. Table 5 below indicates the corresponding signal for each testpoint.

Table 5: Testpoints

Testpoint	Signal	
TP1	SDIN0	Pin 22 of DUT (Not Stuffed)
TP3	BOARD3.3V	
TP6	DUT_DVDD	Output of voltage regulator U35
TP7	DUT_3.3V	DUT_AVDD, DUT_PLLVDD and DUT_ODVDD Supply
TP8	DGND	
TP9	DGND	
TP10	DGND	
TP11	DGND	
TP12	DGND	
TP13	DGND	
TP14	DGND	
TP15	MUTEB	Pin 15 of DUT
TP16	SDA	Pin 16 of DUT
TP17	SCL	Pin 17 of DUT
TP18	LRCLK2	Pin 18 of DUT
TP19	BCLK2	Pin 19 of DUT
TP20	DUT_DVDD	Pin 14, 21, 31, 40 and 51 of DUT
TP21	AGND	
TP22	VREF	Pin 2 of DUT
TP23	AGND	
TP24	AGND	
TP25	AGND	
TP26	LRCLK0	Pin 26 of DUT (Not Stuffed)
TP27	BCLK0	Pin 27 of DUT (Not Stuffed)
TP29	ADI Testpoint	
TP30	ADI Testpoint	
TP31	ADI Testpoint	
TP32	AGND	
TP33	AGND	
TP34	AGND	
TP35	BCLK1	Pin 35 of DUT (Not Stuffed)
TP36	LRCLK1	Pin 36 of DUT (Not Stuffed)
TP37	SD0	Pin 37 of DUT – I^2C Address select on reset (Not Stuffed)
TP42	DGND	
TP43	DGND	

SECTION 2 – POWERING UP THE ADAV4601

This section explains the recommended sequence of register writes which are required for powering up the device. These register writes are required to ensure that there is no noise at the output of the audio processor while the part is going from the reset state to full power mode. Please refer to the Appendix A – Detailed Register Descriptions for more details on register settings

WRITE SEQUENCE EXPLAINED

The ADAV4601 is controlled via I^2C; therefore the following sequence is given in the following format

34 AAAA DDDD

Where 34 is the I^2C address of the ADAV4601, AAAA is the register address in the ADAV4601 that is being written to and DDDD is the data that is being written.

POWER UP SEQUENCE

After power has been applied initially to the part, nothing in the device is powered up. This means that the ADAV4601 is in reset state.

The recommended power up sequence from this state is therefore as follows.

34 0000 0000

This sets the PLL Frequency to be equal to 512*Fs. Given that the sampling frequency is given as 48kHz, this means the MCLKI frequency is set to be 24.576Mhz. Depending on the clock frequency required, this register must be set accordingly.

34 0006 0200

This register write powers up the PLL block in the ADAV4601. This bit must be set to ensure the correct operation of the device.

34 000A 0801

Setting this register to this enables the PLL for use. Like the above register write, this bit must be set to ensure the correct operation of the device. There should be a delay of 15ms (worst-case scenario) after this write to ensure that the PLL has locked to the MCLKI frequency.

34 008D 0628

This register write disables the clamp on the headphone output. By default, the headphone output is connected directly to ground. Once this is disabled, the headphone is in normal operation. This register write also disables the voltage reference, which is enabled by default. When enabled and the DACs powered up, Vref is at 1.5V. This sharp transition from 0V to 1.5V causes a pop at the output of the part. See Figure 7 for more details.

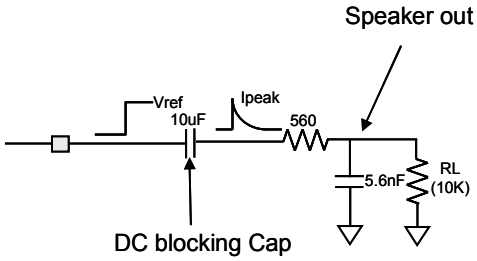


Figure 7: Vref powered up instantaneously

This is disabled in this register write, and the Vref is powered up at a gradual rate to ensure no pop is audible at the output.

34 0005 A06F

This register write powers up the Auxiliary DACs; AUXDAC1 and AUXDAC3. It also powers up the DAC internal reference buffer and puts all DACs into normal mode.

34 0006 023F

This register write powers up the headphone 1 amplifiers as well as the headphone 1 DACs.

Note: AUXDAC4 and headphone 1 are connected together. Therefore, to use AUXDAC4, headphone 1 must be powered up.

Note: Theses two writes power up all available DACs on the ADAV4601. In power critical applications or depending on the number of DACs required, the user can power up the minimum number of DACs needed for operation.

34 000B 2020

This write disables the tri-state on the headphone. By default the headphone outputs are tri-stated. This register write enables the headphone outputs for normal use.

34 0329 0400

This register write adjusts the current to the Vref charger circuit. By writing this, it changes the initial change in the Vref charge profile to a much more gradual increase, which further decreases the chances of hearing any pops on the output of the ADAV4601. For more information on the Vref charger circuit, refer to Figure 9.

34 008D 06BA

This register write, enables the Vref charger circuitry which when set allows the gradual charging of the Vref voltage from 0V up to 1.5V in an S shaped profile, which ensures that there are no sharp current spikes at the output of the part, which would otherwise cause noise. See Figure 8 for more details.

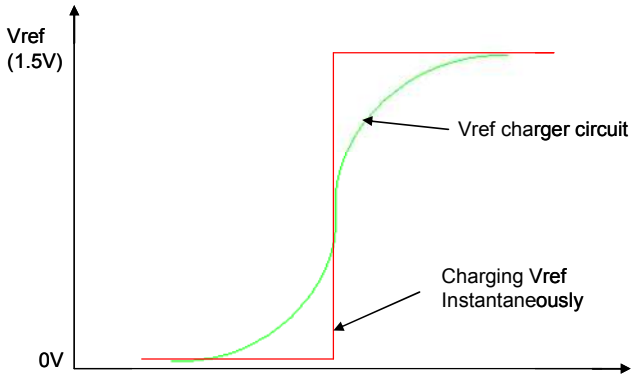


Figure 8: Vref Charger Circuit

This register write also powers up the iDAC buffer which is used to isolate the iDAC from the I/V converter. This ensures that the current source drain voltages in the iDAC are fully charged up before they are applied to the current to voltage converter. This ensures that the current source drain voltages follow the Vref charger circuit and when transferred from the iDAC buffer to the current to voltage converter, they are both at the same potential, which means that there is no sharp voltage transition at the output of the DAC. This register also enables the charge iDAC function. When this bit is set to 1, this allows the instantaneous charging of the current DAC. This helps to ensure that the DACs are powered up quicker.

34 0305 0100

This register write improves the DAC performance. It is therefore recommended that this be written to the part. There should be a delay of at least 100ms after this write. This ensures that the Vref charger has had ample time to charge the Vref voltage from 0V to 1.5V

34 008D 06A0

Once the Vref charger has charged the Vref voltage to 1.5V (after approximately 100ms). This register write disables the Vref charger circuitry and enables the Vref voltage straight from the Vref core. See Figure 9 for more details.

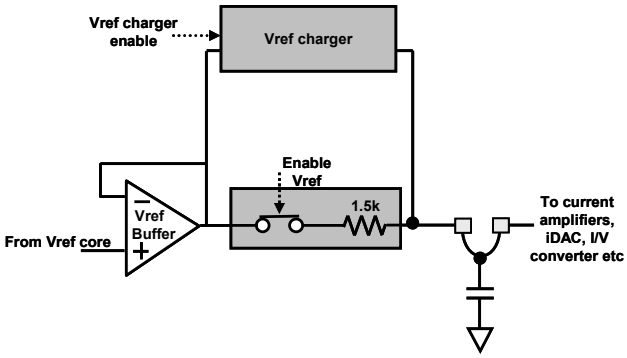


Figure 9: Vref charger circuitry

There should be no DC level jump which would otherwise cause a noise on the output of the DAC. This register write also disables the charge iDAC function. As the current DAC is now fully charged, there is no reason to have this enabled anymore.

34 0000 001B

This register write now enables the Audio Processor for the ADAV4601. It also globally powers up all parts of the ADAV4601 that have not already been powered up, this will include the SRCs, the PWM section, the ADC and DAC engine (as well as all the ADCs which had not been previously powered up) and the audio processor. This register write also enables the both the SRC1 and SRC2 for use.

Note: This register write powers up all parts of the ADAV4601 that were not previously powered up. In power critical applications this may not be desired. In this case the ADAV4601 can have the various digital blocks powered up individually using the digital power management register.

Note: This register was written to previously to set the PLL frequency, for this example the PLL frequency is assumed to be 512*Fs. This must be kept consistent throughout the power up sequence.

34 000B 2121
34 000B 2222
34 000B 2323
34 000B 2424
34 000B 2525
34 000B 2626
34 000B 2727
34 000B 2828
34 000B 2929
34 000B 2A2A
34 008D 07A0

This sequence of writes must be written in succession so as to decrease the gain of the headphone amplifiers from 0dB to-15dB in 1.5dB steps. The headphone amplifier gain by default is set to 0dB which means that the outputs reference voltage is set to 1.5V, by decreasing this to -15dB means that the reference voltage is reduced to 0.267V, at which there is a much lower chance of hearing any spurious responses caused by any possible DC level changes while enabling the iDAC. It is at this point that the iDAC is connected to the current to voltage converter. The profile of the reference voltage is shown in Figure 10.

34 000B 2929
34 000B 2828
34 000B 2727
34 000B 2626
34 000B 2525
34 000B 2424
34 000B 2323
34 000B 2222
34 000B 2121
34 000B 2020

Once the iDAC and the current to voltage converter have been connected there is no longer any potential issue about hearing a click at the output of the headphone as there are no longer any DC level changes and therefore the headphone amplifier gain can be increased again from 0.267V up to 1.5V. These register writes increase the gain of the headphone amplifiers in 1.5dB steps from -15dB to 0dB. The profile of the reference voltage is shown in Figure 10.

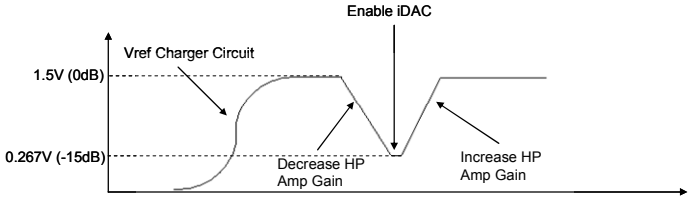


Figure 10: Vref Profile at headphone output

34 008D 0720

As the iDAC and the current to voltage converter are now connected, there is no longer any need for the iDAC buffer. This register write therefore powers it down.

Note: This is the final register write, which is used to power up the DAC. At this point in the power up sequence, the DACs are fully powered up and ready to be used.

34 0081 0202
34 0086 0202

Once the SRC detects invalid data, the outputs of the SRC are slewed to a muted state. However in the default audio flow there is a slew which controls the mute and un-mute function on the output from the SRC. As it would be undesirable to have two slews controlling the same thing, these two register writes disable the internal SRC mute slew so that the only one controlling that mute and un-mute function on the SRC output is that which is in the default audio flow.

34 0082 1843
34 0087 1843

These register writes set the delay for the SRC un-mute signal from both the SRC1 and SRC2 to 7.1ms from when the output of the SRC is un-muted. Refer to Figure 11 for more details regarding this un-muting sequence.

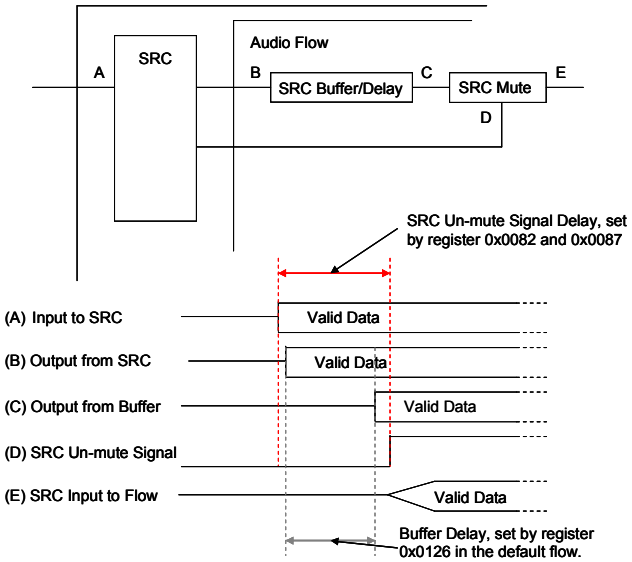


Figure 11: SRC Unmute Sequence

Once there is valid audio in the SRC, the SRC un-mutes immediately, however there is a buffer in the default audio flow, which stores samples from the output of the SRC. This buffer must be full of valid data before it is in turn un-muted and data sent to the audio flow. Therefore, the un-mute signal from the SRC is delayed by a time, which is just marginally longer than that of the time taken to fill the audio flow buffer. If full of valid data, this is then un-muted and data is sent to the audio flow.

34 001F D890

This register is the PWM control register 2. By setting this register to this value, the PWM outputs are re-synced at the output pads by a clock from the digital core of the ADAV46xx. By default all 4 PWM outputs from the ADAV4601 are phase shifted by 45°. By resyncing them at the output, they are all re-aligned. This register write enables the PWM ready early signal. This PWM ready output from the ADAV4601 can be used to indicate to an external microcontroller that an external power stage can be un-muted. Refer to Figure 12 for more information on how the PWM ready signal relates to the PWM outputs.

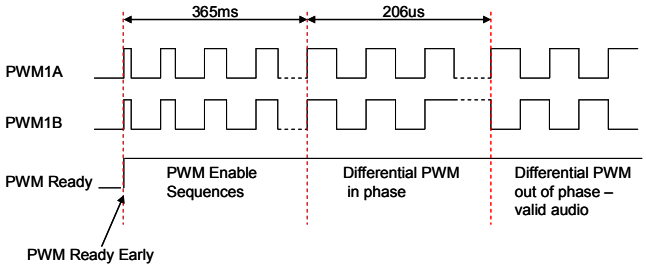


Figure 12: PWM Ready Signal

34 001E 818A

This register setting is used to enable the PWM low signal clamping, if the input to the PWM block from the audio processor goes below the low-level threshold also set by this write, the output of the PWM generator is muted, i.e. PWM generators produce a 50/50 duty cycle.

34 008F 000F

This register sets the PWM clamp release time. This controls the number of consecutive samples that the PWM will stay muted after the signal that is inputted to the PWM block was outside the threshold limit set by the previous register write. This ensures that low level spurious responses are kept muted until valid audio is ready to be outputted. This register write sets the clamp time to be 11 samples. This means that once the input goes above the threshold set by the previous register write, no sound will appear on the output of the PWM generator for at least 11 samples.

34 0126 0150

This register write is used to initialize the SRC delay buffer in the default audio flow. Once the SRC starts outputting valid audio, this is stored in the audio flow buffer. Once the un-mute signal for the audio flow has been set, the buffer in the audio flow starts outputting valid audio. The one requirement for this is to ensure that the SRC un-mute signal delay (set by register 0082 and 0087) is marginally longer than the audio flow SRC buffer delay (set by register 0126). Refer to Figure 11 for more details on setting the SRC delay buffer.

34 000A 5F81

This register is the miscellaneous control register. By setting this register to this value, the four PWM channels are enabled and also the PWM enable and disable sequences. These are specially constructed patterns which are used to bring the PWM channels from a zero condition to a 50/50 duty cycle. Refer to Figure 12 for more details on the PWM enable sequences. This register is also used to enable the MCLK_OUT function; this means the part can output a clock on the MCLK_OUT pin which can be used to clock another device. Should this function be required this bit must be set.

Note: This register was written to previously to enable the PLL. Therefore when writing to this register care must be taken not to disable the PLL as this will affect the operation of the device.

34 0100 7077
34 0101 7777

These registers control the input multiplexers in the default audio flow. These writes send the data that is on the SRC1 channel to all the available outputs: Tweeter, Woofer, AUXDAC1, AUXDAC3, SPDIF and SDO0. Once these registers have been set the part must then be un-muted.

34 0121 00FF

This register write, un-mutes the ADAV4601. If the there is data on the SRC1 channel this should now be available on all outputs.

Note: By default this power up sequence puts data from SRC1 out on all outputs of the ADAV4601. This will have to be changed depending on the flow used.

POWER DOWN SEQUENCE

From the powered up state that was described in the previous section. The following is the recommended sequence when powering down the ADAV4601.

34 0121 0000

Mute the audio flow using the fast mute control register. This mutes all outputs on the ADAV4601. It is recommended to wait for 100ms after performing this write before performing the next I^2C write.

34 008D 07A0

Power up the iDAC buffer, this is used to disconnect the iDAC and the DAC current to voltage converter. It is recommended to wait for 10ms after this write so as to ensure that the iDAC buffer has been fully powered up.

34 000B 2121
34 000B 2222
34 000B 2323
34 000B 2424
34 000B 2525
34 000B 2626
34 000B 2727
34 000B 2828
34 000B 2929
34 000B 2A2A

These register writes decrease the gain of the headphone amplifier in steps of -1.5dB from 0dB to -15dB. This is an additional precaution to ensure that no pops or clicks are heard at the headphone outputs.

34 008D 06A0

This register write disconnects the iDAC from the current to voltage converter. This means that any large current flow that may occur on the iDAC will not be heard at the output.

34 0329 0400

This register write adjusts the current to the Vref discharger circuit. By writing this, it changes the initial change in the Vref charge profile to a much more gradual decrease, which further decreases the chances of hearing any pops on the output of the ADAV4601.

34 008D 06A9

This register write, enables the Vref discharger circuitry which when set allows the gradual discharging of the Vref voltage from 1.5V down to 0V in an S shaped profile, which ensures that there are no sharp current spikes at the output of the part, which would otherwise cause noise. It is recommended that there is delay of at least 500ms so as to be sure the capacitor at the Vref pin has discharged fully. Refer to Figure 13 for more details on the output of the Vref discharger circuit.

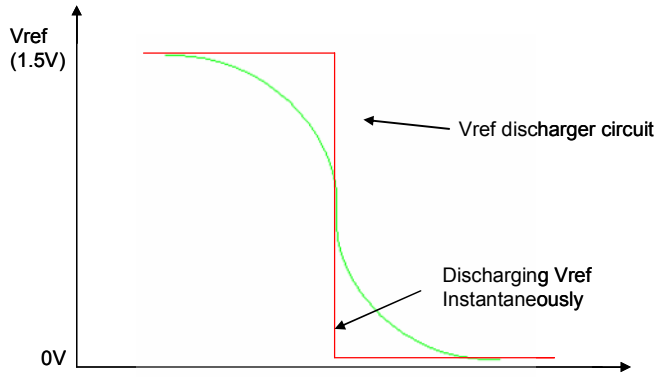


Figure 13: Vref Discharger Circuit

34 0005 866F

This register write powers down the Vref buffer. The reference voltage Vref has now been powered down to 0V which means that the input to every amplifier is already at 0V. Therefore there is no longer any need for the Vref buffer to be powered up.

34 008D 02A1

This register write enables Vref which effectively connects the output of the Vref buffer to ground and also clamps the headphone outputs to ground. It is recommended to wait for 1 second after performing this write to ensure that all residual charge is gone from the Vref capacitor.

34 000B 0000

This write tri-states the headphone outputs which sets the outputs of the headphones to high-impedance.

34 000A 0F81

This write disables the PWM channels. It is recommended to wait for 2 seconds after this write before performing an additional I^2C write.

34 0006 0200
34 0005 0000

These register writes power down the headphone amplifier, headphones DACs and all the Auxiliary DACs as well as putting all DACs in low power mode.

34 0000 0000

This register write powers down the rest of the device. This disables all the digital section on the ADAV46xx as well as disabling the processor and the PLL.

SECTION 3 – PROGRAMMING THE ADAV4601

DESIGNING AND DEVELOPING

This section describes how it is possible to control the ADAV4601 audio processor and evaluation board with the software supplied.

If the customer wishes to develop their own custom audio flow. Analog Devices, Inc. offers an award-winning graphical programming tool (SigmaStudio™) that allows custom flows to be quickly developed and evaluated. This allows the creation of customer-specific audio flows, including use of the Analog Devices library of third-party algorithms. This means that the customer can tailor the audio processing to their own specific needs and requirements.

Alternatively, the ADAV4601, by default, loads a dedicated TV audio flow that incorporates full matrix switching (any input to any output), automatic volume control that compensates for volume changes during advertisements or when switching channels, dynamic bass, a multi-band equalizer, and up to 200 ms of stereo delay memory for audio-video synchronization. This audio flow is loaded by default when the part is powered on. To support and control this audio flow, it is possible to use the SigmaStudio programming tool to perform I²C writes to the ADAV4601 to enable the device and control the audio flow registers.

Figure 14 shows both methods of controlling the audio processing, using either the default flow on the chip or developing a new unique flow. Instructions for both methods are included in this section.

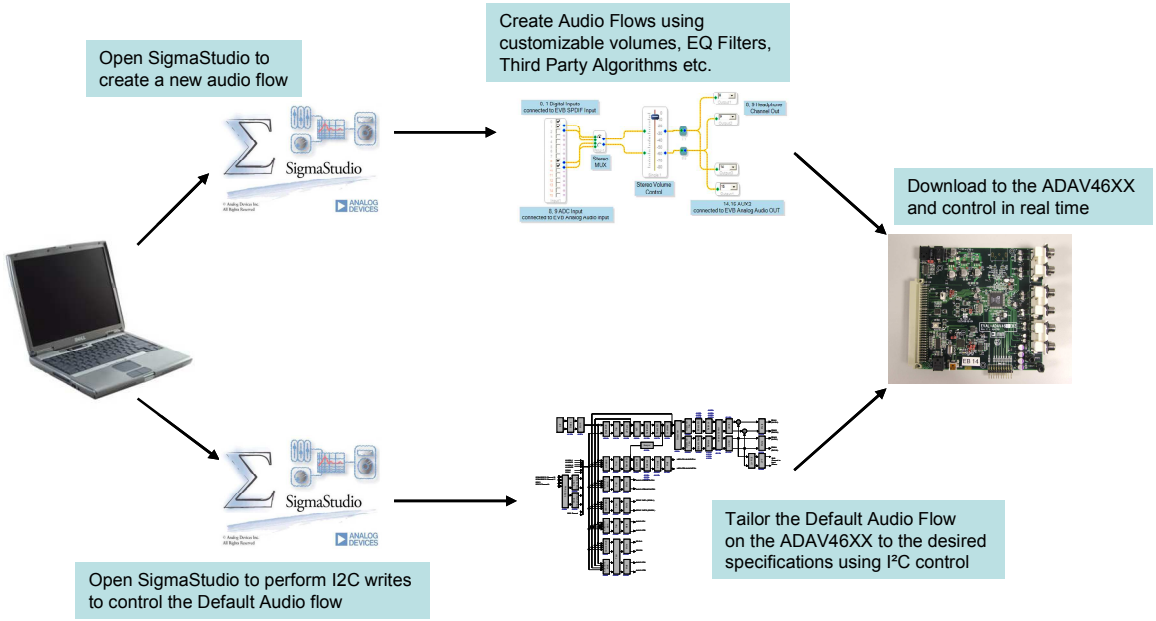


Figure 14: Controlling the Evaluation Board

CONNECTING THE HARDWARE

The ADAV4601 evaluation kit ships with a USB to serial converter and adaptor board which allows the graphical programming suite SigmaStudio to communicate with the ADAV4601 evaluation board. The boards should be connected as shown in Figure 15. The SigmaStudio USB Serial converter and adaptor board should be connected to the ADAV4601 Evaluation board using the 96-way connector (J3).

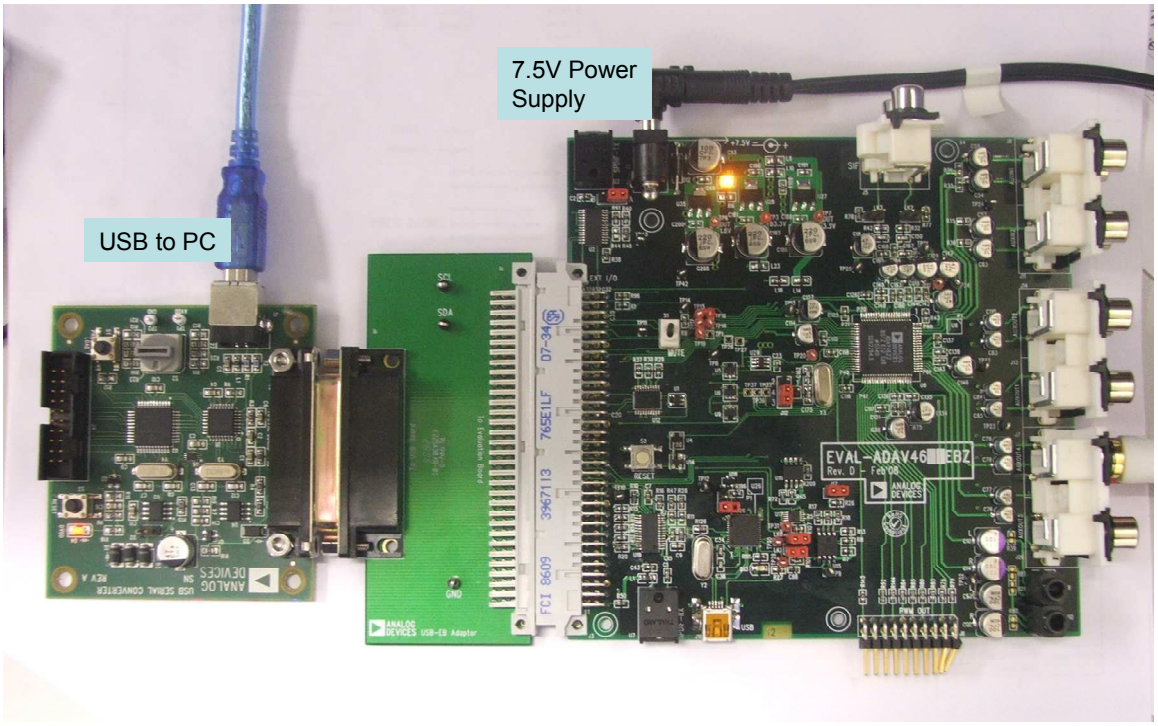


Figure 15: SigmaStudio Hardware Setup

Upon first connecting the USB to serial converter board you may be required to install the correct drivers.

Note: Ensure that SigmaStudio is closed each time when connecting the USB Serial Converter board to the PC. This ensures that the correct driver is loaded for the USB to serial converter.

INSTALLING SIGMASTUDIO

The Installation CD that shipped with the ADAV4601 automatically installs SigmaStudio. If SigmaStudio has not been installed please, remove the CD and re-insert this into the PC. SigmaStudio should install automatically, if SigmaStudio does not launch, select the **setup.exe** file on the CD and follow the instructions.

INSTALLING ADDITIONAL DLL FILES

Before launching SigmaStudio the user must add additional dll files to the SigmaStudio installation directory located on the local drive at ***C:\Program Files\Analog Devices Inc\SigmaStudio***. These additional files can be found at the following location ***C:\Program Files\Analog Devices\ADAV46xx Documentation\Additional dlls***.

Once these have been added to this directory the SigmaStudio can then be launched by going to **Programs-> Analog Devices SigmaStudio -> SStudio.exe.** The user should then see the following window.

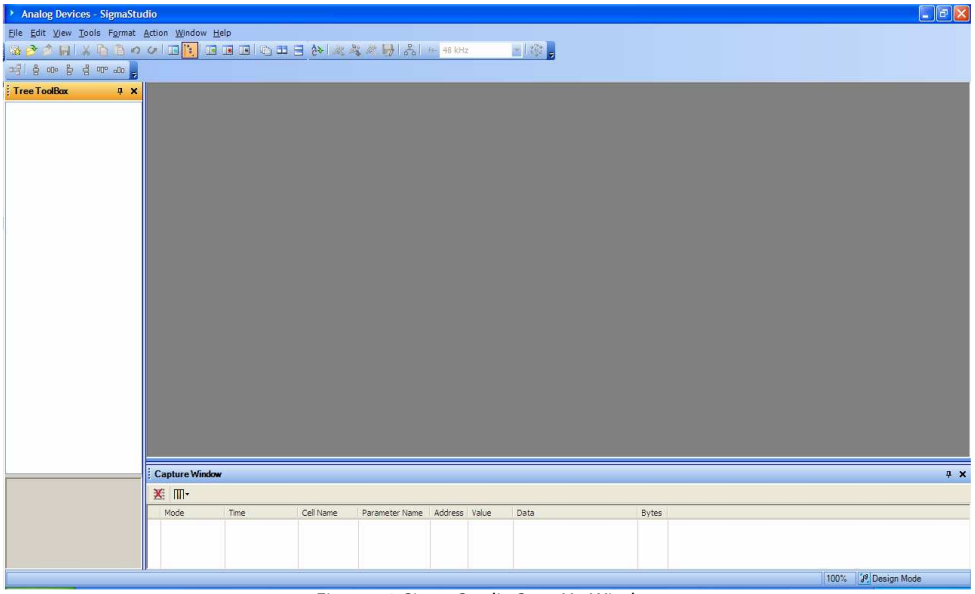


Figure 16: SigmaStudio Start Up Window

Once this window has launched, the additional **dll** files can be activated by going to **Tools -> Add-Ins Browser**.

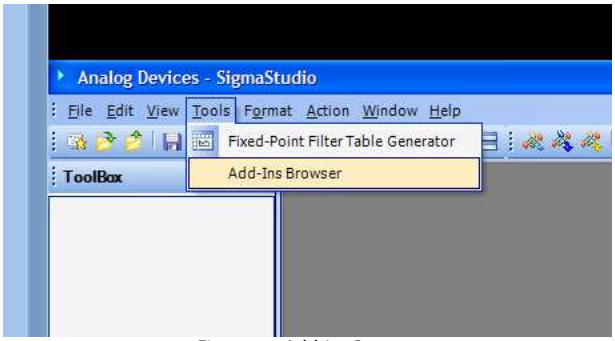


Figure 17: Add-Ins Browser

Once selected this opens the Add-Ins Browser Window which can be seen underneath. The **Add-Ins Browser** is used to include other options in the SigmaStudio Program. When this is opened the **dlls** that have been copied to the SigmaStudio directory on the C drive should be selected here. These **dll** files are now available to use in SigmaStudio.

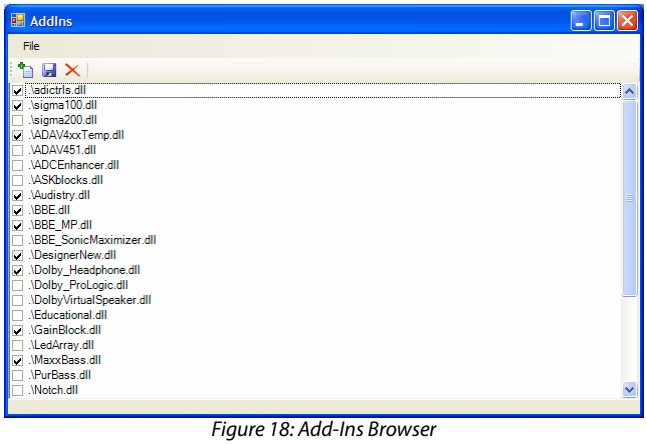


Figure 18: Add-Ins Browser

EVALUATING THE ADAV4601 USING SIGMASTUDIO

An example SigmaStudio project is supplied with the ADAV4601 evaluation board. Once SigmaStudio has been launched, select **File -> Open**, and then navigate to the following folder ***C:\Program Files\Analog Devices\ADAV46xx Documentation\Sample Audio Flow***. In this folder select the project files entitled, Sample Audio Flow.dspproj. This is the Evaluation Board Test flow.

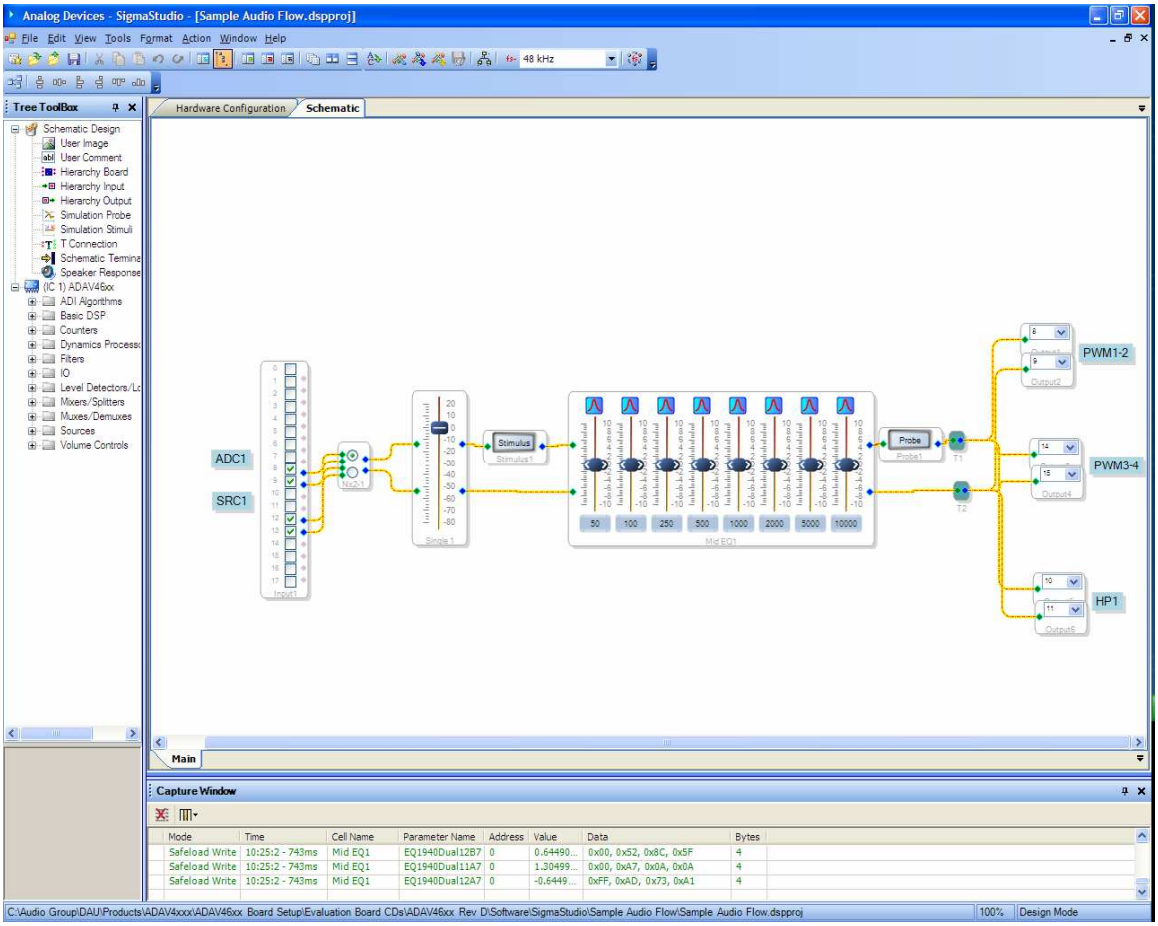


Figure 19: SigmaStudio Test Flow

This is the schematic window of SigmaStudio in which the audio flow can be created. This simple audio flow consists of two inputs, one from the analog input ADC1 and the second from the asynchronous data input SRC1. These are then connected to a 2 to 1 stereo mux which allows the user to select the input to the audio flow. This input is then routed to a single stereo volume control and then put through an eight band equaliser. This is then outputted on the PWM channels, the Headphone 1 outputs and the AUXOUT1 outputs.

Once this project is open the user may be able to edit this by dragging additional blocks from the toolbox which is to the left of the schematic window. From this toolbox it is possible to add various other audio flow related blocks.

Note: To develop a new flow the evaluation board does not need to be connected to your PC. It only needs to be connected when the flow is ready for download and testing. However it is useful to have the evaluation board present as this allows the user to test their audio flow while they are creating it.

POWERING UP THE ADAV4601 EVALUATION BOARD

Once this project is open, the next step is to power up the evaluation board. This can be done by going to the Hardware Configuration Window Tab and then to the **IC 1 – IIC Script** as indicated in the Figure 20 below.

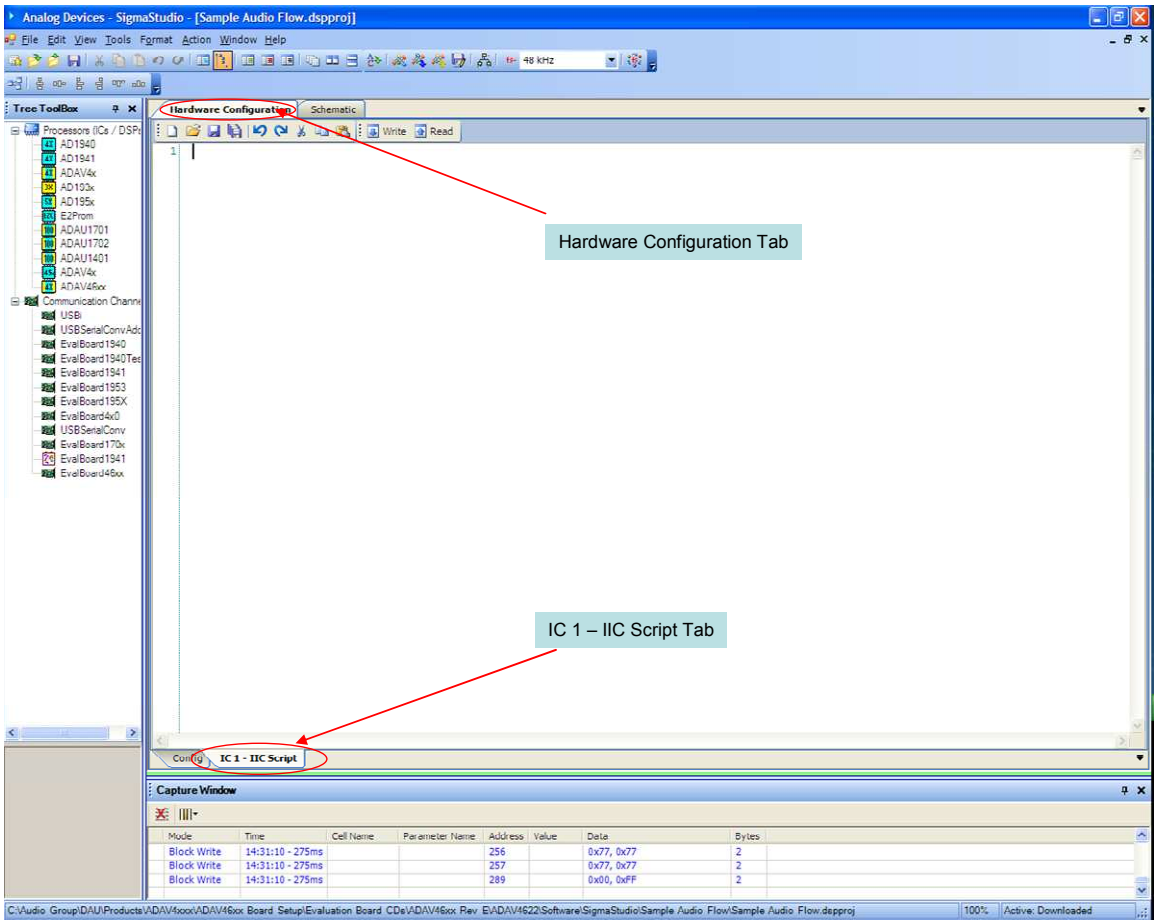


Figure 20: Powering Up Evaluation Board

This **IIC Script tab** is used for performing I²C writes to the ADAV4601. To power up the ADAV4601, select **File -> Open** and then navigate to the folder in which the SigmaStudio Test Flow is located; ***C:\Program Files\Analog Devices\ADAV46xx Documentation\Sample Audio Flow***.

I²C writes are performed to the part in the format:

Device Address	Register Address	Data
34	0000	001B

To write to the device, select **Write**.

Note: This power up sequence is simply designed to power up and enable the audio processor. When designing the ADAV4601 into a system, it is recommended to use the power up sequence documented in Section 2 – Powering up the ADAV4601.

Note: To ensure that SigmaStudio is communicating correctly with the ADAV4601 it is possible to read back from the registers you have written to by writing the following sequence:

Device Address	Register Address	Data
34	0000	r2

Where r2, reads back 2 bytes of data. This can be included as part of the same power up script. To read back from the device, select **Read**. If all registers have been programmed properly, the following dialog box will open displaying the readback values.
If the values read back from SigmaStudio do not match the register settings below, the ADAV4601 has not been programmed correctly. Please reset the board and download the power up sequence again.

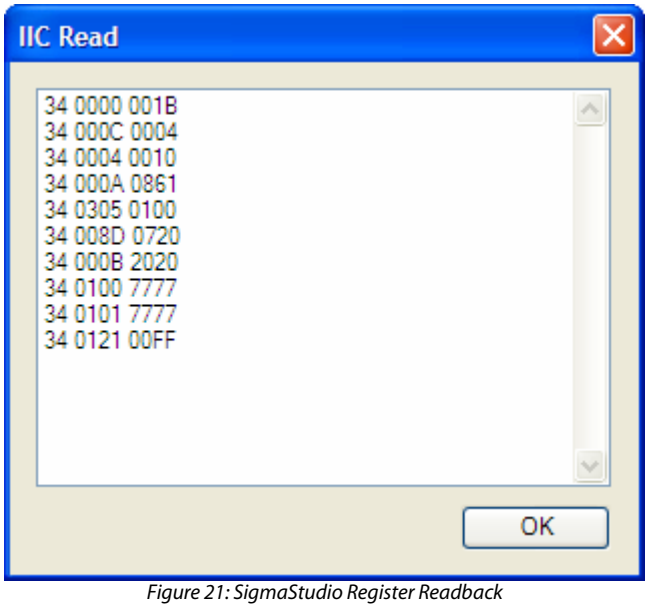


Figure 21: SigmaStudio Register Readback

DOWNLOADING AND COMPILING AN AUDIO FLOW

Once the Audio Flow is created in SigmaStudio and the ADAV4601 is fully powered up, all that remains to be done is for the audio flow created in SigmaStudio to be compiled and downloaded.

To do this, click on the **Link Compile Download** button in SigmaStudio. See Figure 22 for more details.

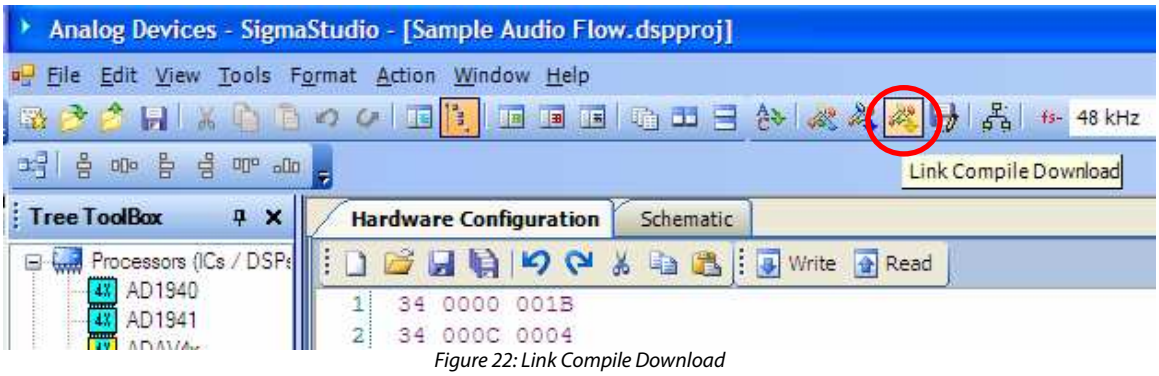


Figure 22: Link Compile Download

SigmaStudio will now compile, and if no errors are found, will download the flow to the ADAV4601. The status bar below the schematic window will turn green once the schematic has been downloaded.

Note: The green status bar only indicates if the audio flow has compiled properly, this is not an indication that the flow has downloaded properly to the ADAV4601, therefore it is important to read back from the registers when powering up the ADAV4601 to ensure that SigmaStudio is communicating properly with the USB adaptor board.

The graphical blocks in your flow can now be used to control the audio processing in real time. Sound should also appear at the output of the device depending on the setup of the Audio flow i.e. provided the input mux is set correctly and none of the outputs are muted.

Note: if you change the order of the blocks, or rewire any part of the audio flow, then you must recompile and download the flow. There is no need to recompile if you simply adjust parameters. If you make any changes that require a recompile, the status bar will turn blue again.

CREATING A NEW AUDIO FLOW

The previous section explains how to evaluate the ADAV4601 evaluation board with SigmaStudio using the provided Sample Audio Flow. However if the user wants to create a new audio flow from scratch the procedure is as follows;

1. In SigmaStudio create a new flow by clicking on **File -> New Project**. You should be presented with the following window.

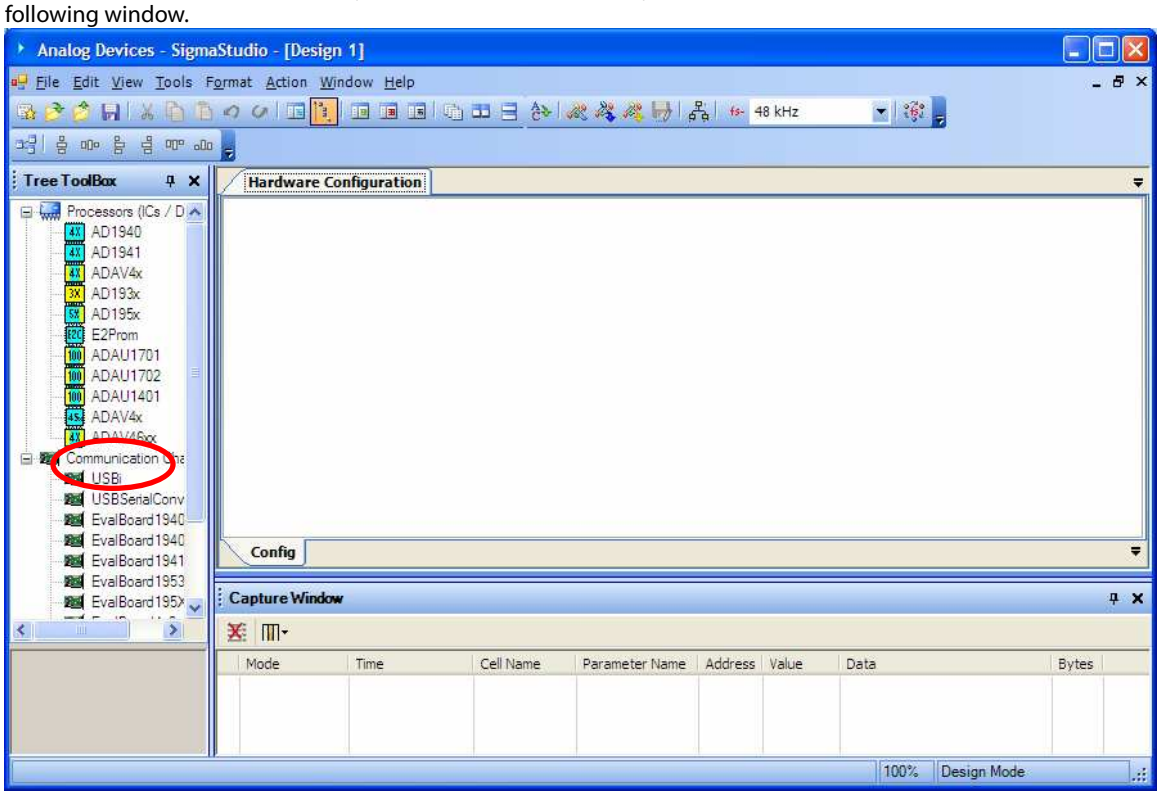


Figure 23: Selecting the Processor

2. First, select the Correct Processor from the Processors (ICs/DSPs) List. Drag the icon labeled **ADAV46xx** from the window on the left into the Hardware Configuration window on the right. This tells SigmaStudio which processor you are using.
3. Next you must select the correct hardware for communicating with the evaluation board. Click on the Communications Channel tab and drag the icon labeled **EvalBoard46xx** into the main window.
4. Connect the **EvalBoard46xx block** by clicking on the blue output pin to the green input pin on the **ADAV46xx block**, as shown.

Note: When the communication channel is selected and connected to the eval board, the icon for the communication channel changes from orange to white. This is provided the USB adaptor board shipped with the evaluation kit is connected to the PC. This indicates that the adaptor board is using the correct driver. **The adaptor board should always be connected before SigmaStudio is launched.**

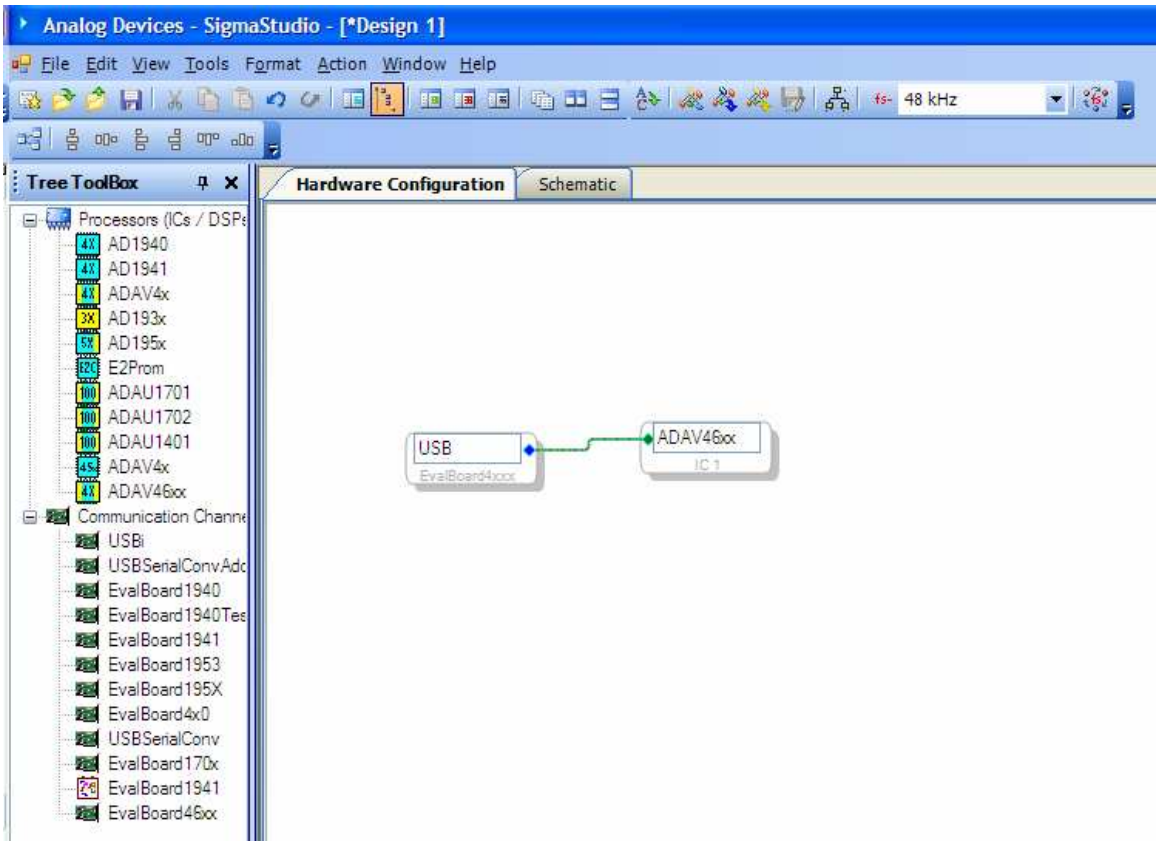


Figure 24: Connecting the Eval Board to the Adaptor Board

5. SigmaStudio is now configured to use the ADAV4601 Evaluation Board. To begin creating a new flow click on the **Schematic tab**.

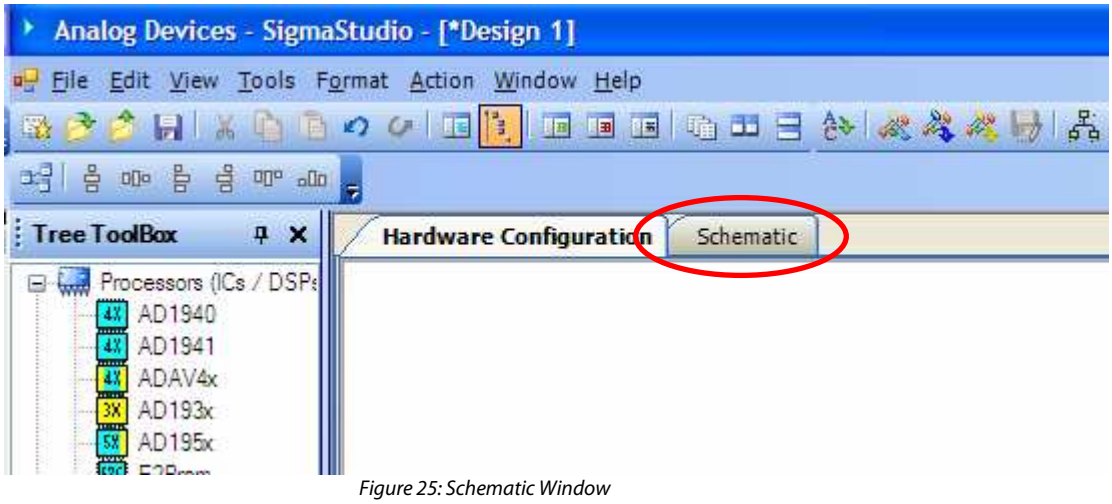


Figure 25: Schematic Window

6. Algorithms are added from the **Tree ToolBox** on the left to develop your audio flow, using the same drag and drop method as above.

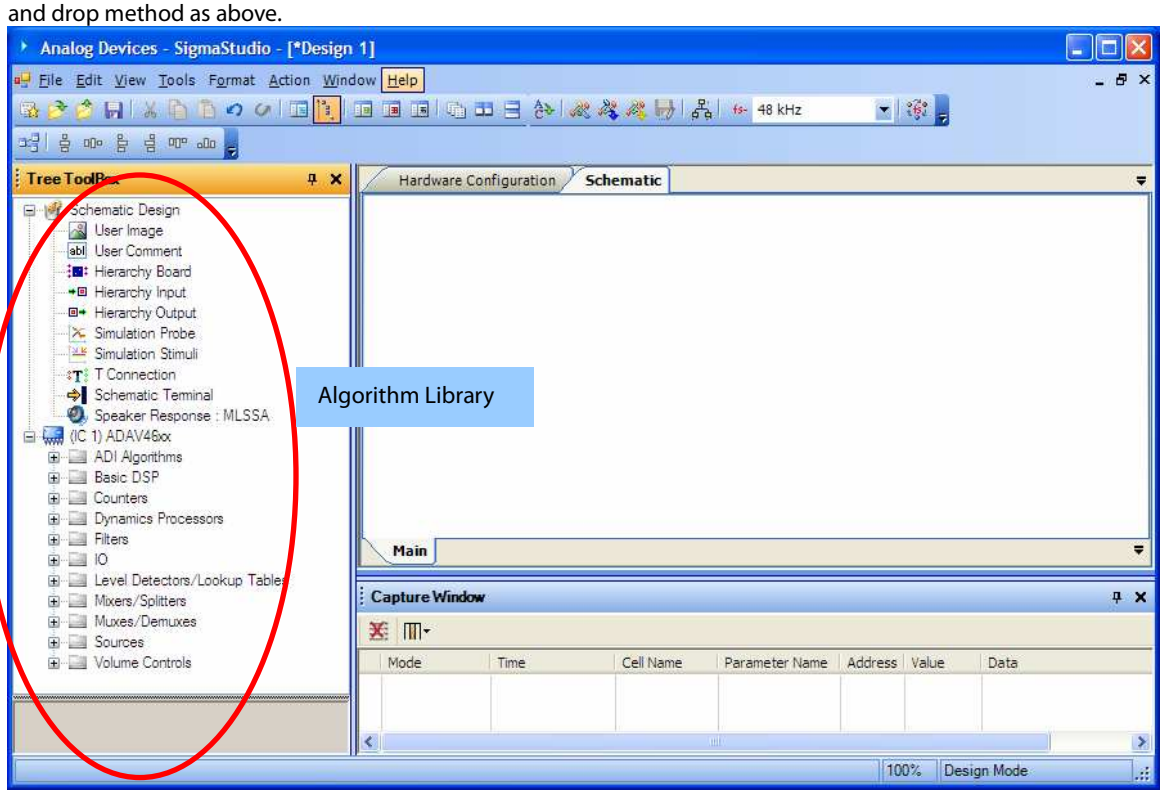


Figure 26: Schematic ToolBox

Note: All input and output pins on each algorithm block must be connected. Pins that are left unconnected will cause an error when compiled.

In order to pass data from the various hardware blocks on the ADAV4601, such as ADCs, DACs and SRCs to the Audio flow, each input and output on the ADAV4601 has a input pin assignment in SigmaStudio. The pin assignments and how they correspond with the appropriate inputs and outputs is shown in Figure 27.

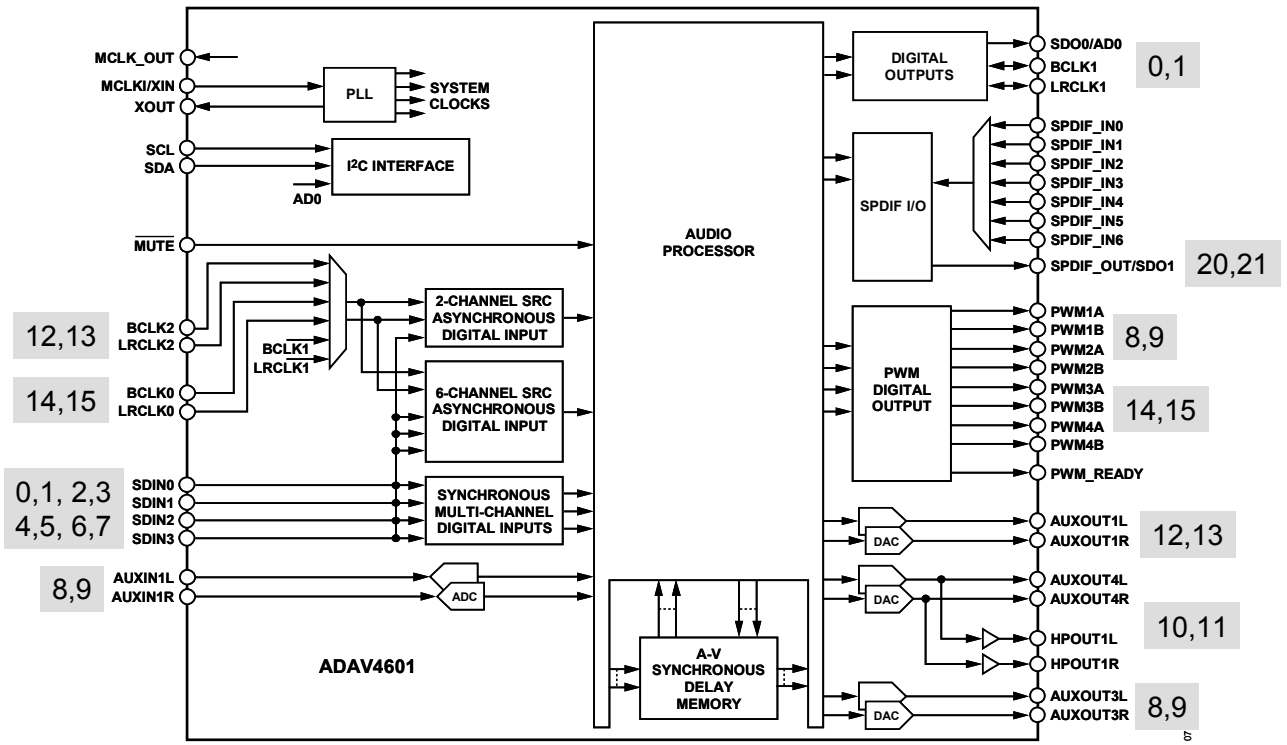


Figure 27: SigmaStudio IO

Table 6: ADAV4601 SigmaStudio Pin Assigments

Input Channels	
0	SDINL0
1	SDINR0
2	SDINL1
3	SDINR1
4.	SDINL2/SRC2BL
5	SDINR2/SRC2BR
6	SDINL3/SRC2CL
7	SDINR3/SRC2CR
8	AUXIN1L
9	AUXIN1R
10	NA
11	NA
12	SRC1L
13	SRC1R
14	SRC2AL
15	SRC2AR

Output Channels	
0	SDOL0
1	SDOR0
2 – 7	No Connect
8	PWM1/ AUXOUT3L
9	PWM2/ AUXOUT3R
10	AUXOUT4L/Headphone 1L
11	AUXOUT4R/Headphone 1R
12	AUXOUT1L
13	AUXOUT1R
14	PWM3
15	PWM4
16	NA
17	NA
18	NA
19	NA
20	SPDIF OUTL
21	SPDIF OUTR

For example, in your audio flow, to use incoming data on AUXIN1L and AUXIN1R, inputs 8 and 9 on your SigmaStudio Input block are used. If you wished to feed them to PWM1 and 2, you would use outputs 8 and 9, as shown in Figure 28 below.

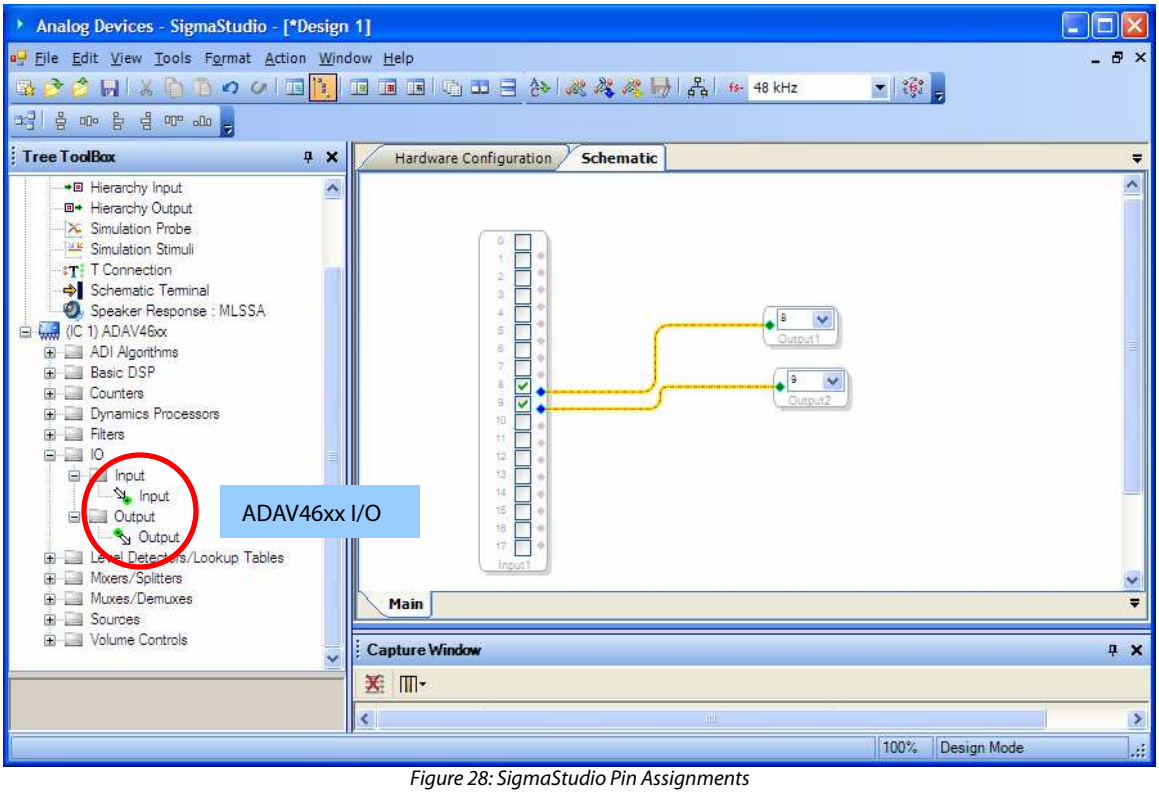


Figure 28: SigmaStudio Pin Assignments

Note: An important point to note is that the size and functionality of the Audio Flow is limited by the number of instructions available in the Program ROM of the ADAV4601. This is only critical for very large designs so therefore care should be taken when designing large audio flows.

The ADAV4601 internally operates at multiples of a sampling frequency (Fs) of 48kHz. When creating audio flows in SigmaStudio it is important to set the sampling frequency of the audio flow to 48kHz. This ensures that all filters coefficients are calculated with a sampling frequency of 48kHz. See Figure 29 for more details on setting the sampling frequency of the audio flow.

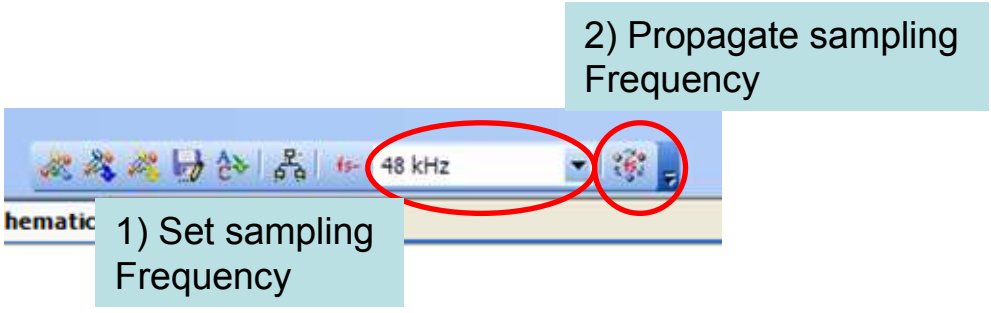


Figure 29: Setting SigmaStudio Sampling Rate

Once the sampling frequency has been set to 48kHz, the propagate sampling frequency should be selected. This ensures that all blocks within the audio flow use this new sampling frequency.

USING SIGMASTUDIO TO CONTROL THE DEFAULT AUDIO FLOW

On the reset of the part (or just after power has been applied) a default audio flow is loaded into the ROMs of the ADAV4601. This flow contains the most common TV audio processing blocks and allows the user to quickly implement the ADAV4601 in their system. The ADAV4601 allows full customization of this flow via a detailed Register Map, which is I²C controlled.

The default flow which is available in the device is shown underneath. The register map used to control the default flow can be found in the Default Audio Flow Registers section.

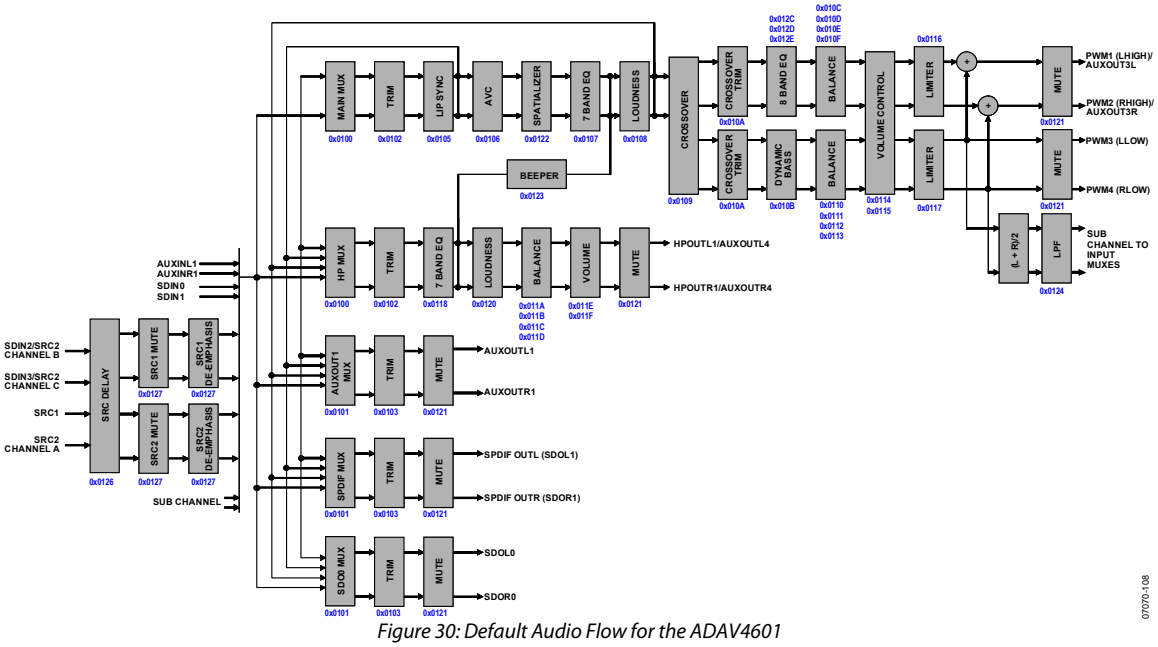


Figure 30: Default Audio Flow for the ADAV4601

The default audio flow in the ADAV4601 is a comprehensive solution which could be utilized in many TV applications. The flow incorporates full matrix switching (any input to any output) and offers a couple of different Analog Devices patented algorithms such as an AVC (Automatic Volume Control) and Spatializer. As well as this the default audio flow offers a number of generic blocks such as volume controls, EQ filters and compressors. Each block can be enabled and disabled as required.

This default audio flow can be completely controlled by I²C writes. Similar to powering up the ADAV4601 using SigmaStudio it is possible to use the **IC 1 – IIC Script tab** in SigmaStudio to control the default audio Flow.

Open SigmaStudio and **Select File -> New Project**. This will open up a new design window. Then select the ADAV4601 Processor and ADAV4601 Evaluation Board as before. Refer to Figure 24 for more details on this.

Now Select the **IC1 – IIC Script tab** as before. Refer to Figure 20 for the **IC1 – IIC Script tab**. This window where it is possible to create I²C writes to the ADAV4601. To Power up the default flow, select file open and navigate to the ***C:\Program Files\Analog Devices\ADAV46xx Documentation\Default Audio Flow Scripts*** directory and select the **ADAV4601 PowerUp.iic** Script File. This can be seen in Figure 31 below.

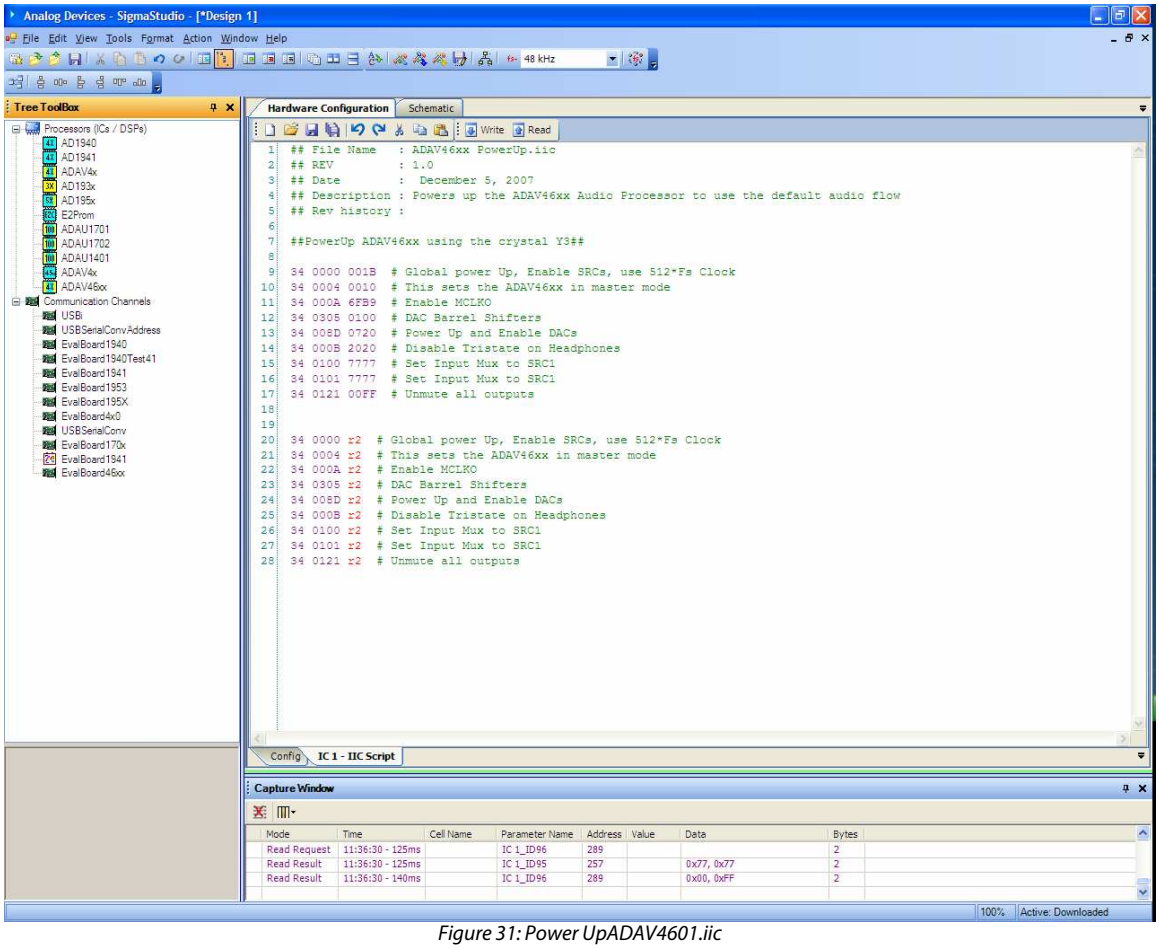


Figure 31: Power UpADAV4601.iic

To Power Up the ADAV4601, select the **Write** button, this will write the power up sequence to the device. Once written, the values written can be read back using the **Read** button. This will return the following dialog box.

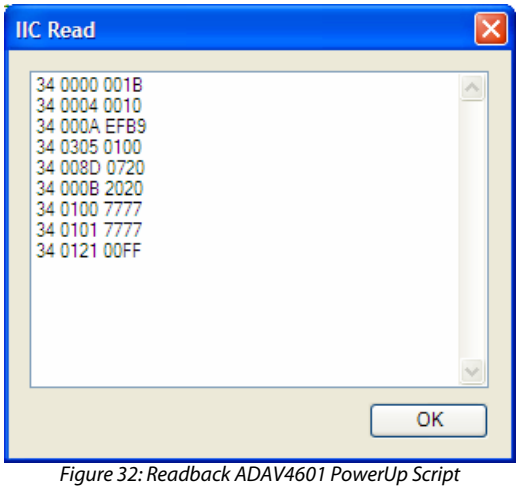


Figure 32: Readback ADAV4601 PowerUp Script

Note: If the dialog box does not display these values, the part has not been programmed properly and will not operate correctly. Reset the evaluation board and write the power up script to the evaluation board again. To reset the evaluation board, please refer to Section 1 – Getting the Evaluation Board Up and Running for more details.

Note: This is a simplified power up sequence to get the ADAV4601 fully operational, the recommended power up sequence provided in Section 2 – Powering up the ADAV4601 should be used when designing the part into a system.

Now the ADAV4601 evaluation board should be completely operational and if there is a digital source connected to the SPDIF receiver, this should be available on all the ADAV4601 outputs. Now referring to the programming guide supplied, it should be possible to tailor the default audio flow to meet the user's needs.

To control the Default Audio flow, a comprehensive set of scripts has been supplied with the evaluation kit. For example to set the Main Channel Volume, on the **IC1 – IIC Script tab**, select Open and select the **Reg 0114 Main Volume.iic** file. This now opens in the Script Window.

Is the user wants to set the main channel volume to -6dB, they must first comment out the default value and then uncomment the -6dB I^2C write. See Figure 33 for more details.

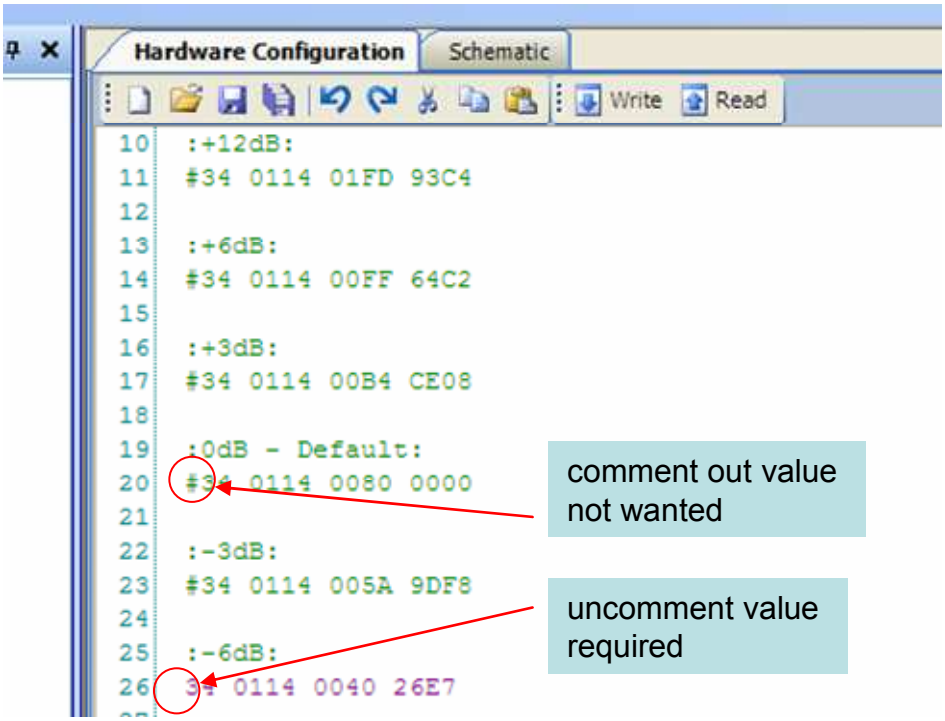


Figure 33: Setting Main Volume

This can then be written to the part as described previously. Likewise all other Default Audio Flow settings can be controlled in a similar way allowing the user full control.

For more details on the default audio flow loaded in the ADAV4601, please refer to Appendix A – Detailed Register Descriptions for more details.

ADAV4601 SELF BOOT

If a custom flow has been designed, it can be stored on an external EEPROM. The ADAV4601 features a self boot option which allows the part to copy ROMS from an EEPROM to its own internal memory. In this way, the user can develop their own custom flows and have them loaded to the ADAV4601 on start up. During the transfer, the ADAV4601 is in master mode, meaning it controls the I2C bus. It controls the copying of the ROMS, easing the load on the microcontroller. During the transfer, it is important nothing else accesses the I2C bus, or conflicts can occur. Once it has completed, the ADAV4601 releases the I2C bus, and becomes a slave again. The time to transfer the ROMS is approximately 1.2sec.

Boot Sequence

1. MCU Powers up the ADAV4601 (Reg 0000h, bit 0) and enables the DSP (Reg 0000h, bit 1) with outputs muted via I2C.
2. MCU writes to ADAX4xyz to enable EEPROM booting via I2C.
3. After 2.5 us time the ADAV4601 will become an I2C master and start reading the EEPROM contents.
4. After 1.16 s the ADAV4601 will be finished booting and release I2C bus, The ADAV4601 is now an I2C slave device.
5. MCU will unmute the ADAV4601 outputs via I2C: ~ 7 ms
6. MCU can continue configuring the rest of the system via I2C.

Note: In order to successfully copy the data from the EEPROM, the ADAV4601 must be powered up and the DSP enabled.

Enabling EEPROM Booting

1. Register 0316h contains the EEPROM chip address. It is set to a default of 0x50. This register must be programmed with the correct EEPROM address.
2. Register 0317h contains the EEPROM start address. It is set to a default of 0x0000. This register must be programmed with the correct EEPROM start address.
3. Register 0200h bit 16 is the Enable Self booting from an external EEPROM bit. Once this bit is set to 1, the Self booting process begins. The ADAV4601 becomes a master on the I2C bus and initiates the data transfer.

EEPROM Data Structure

The following is the general data structure for the EEPROM.

Table 7. EEPROM Data Structure

Message Byte
Length
Length
Device Address
Address
Address
Data 1
Data 2
..
..
..
Data X
Message Byte

Message Byte:	0x01: Block Message Followed by data 0x00: End of Data stream
Length:	Number of blocks to be read (addresses + data) (For example above, Length would be 0009h)
Device Address:	Chip Address of ADAV4601
Address:	Target Memory address in ADAV4601
Data :	Data to be written
Message Byte:	0x01: Block Message Followed by data 0x00: End of Data stream

Boot Time

The EEPROM has 4 blocks which need to be copied to the part.

- SigmaDSP program 3072 x 6 bytes = 18432 bytes
- SigmaDSP parameters 1024 x 4 bytes = 4096 bytes
- Application layer instructions 640 x 5 bytes = 3200 bytes
- Application layer lookup table 6144 x 4 bytes = 24567 bytes

The time given to copy each block is given below:

#1: Sigma DSP Program
18432 bytes + 6 bytes to set address pointer = 18438 bytes

$$t_1 = bytes * 9\frac{bits}{bytes} * \frac{1}{f_{I2C\_clk}}$$

$$t_1 = 18438 * 9 * \frac{1}{390KHz}$$

$$t_1 = 425.49mS$$

#2: SigmaDSP Parameter
4096 bytes + 6 bytes to set the address pointer = 4102 bytes

$$t_1 = bytes * 9\frac{bits}{bytes} * \frac{1}{f_{I2C\_clk}}$$

$$t_1 = 4102 * 9 * \frac{1}{390KHz}$$

$$t_1 = 94.66mS$$

#3: Application layer Instructions
3200 bytes + 6 bytes to set the address pointer = 3206 bytes

$$t_1 = bytes * 9\frac{bits}{bytes} * \frac{1}{f_{I2C\_clk}}$$

$$t_1 = 3206 * 9 * \frac{1}{390KHz}$$

$$t_1 = 73.99mS$$

#4: Application layer lookup table
24567 bytes + 6 bytes to set address pointer = 24573 bytes

$$t_1 = bytes * 9\frac{bits}{bytes} * \frac{1}{f_{I2C\_clk}}$$

$$t_1 = 24573 * 9 * \frac{1}{390KHz}$$

$$t_1 = 567.07mS$$

Total time for 4 sections:

$$t_{total} = t_1 + t_2 + t_3 + t_4$$

$$t_{total} = 1.16s$$

The part should be muted for the duration of the transfer, to avoid unwanted noises on the output.
Once the transfer of data is complete, the ADAV4601 returns to slave mode. The microcontroller can now access the I2C bus again and continue programming the ADAV4601 and other devices.

SECTION 4 – THE DEFAULT FLOW

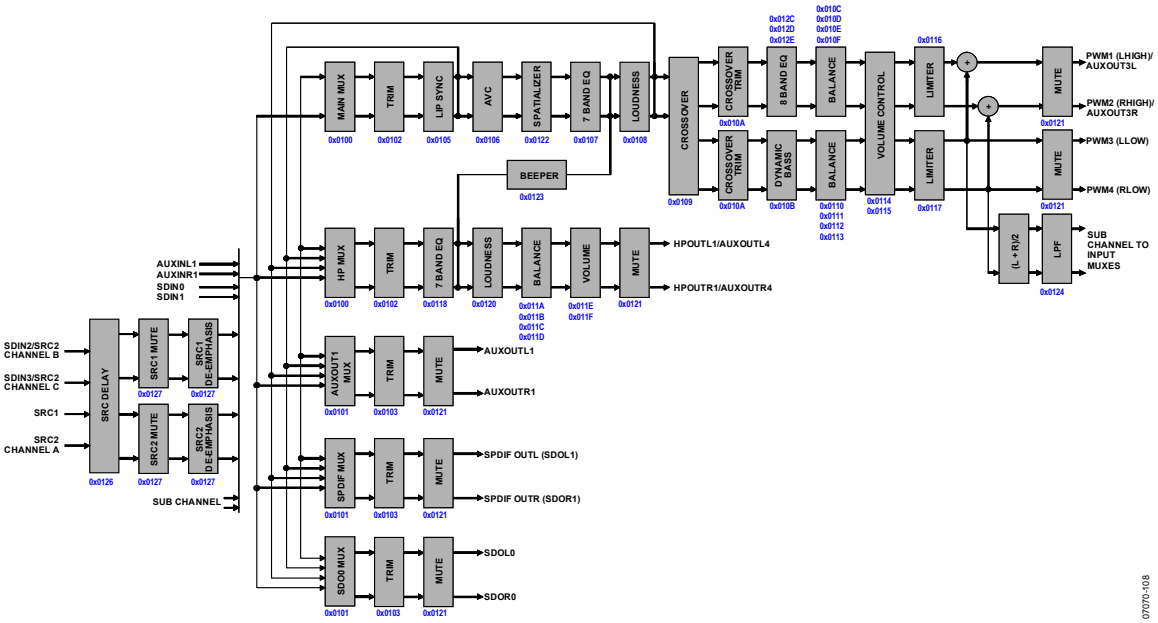


Figure 34: Default Audio Flow for the ADAV4601

DEFAULT FLOW STRUCTURE

The Default Flow consists of five parallel channels all of which are muted by default;
Main
Headphone
AUXOUT1
S/PDIF
SDO0

Main Channel

The core of the Default Flow is the Main Channel which provides the main audio processing; Volume, Lip Sync, AVC, Spatailizer, EQ, Loudness, Dynamic Bass.

The Main Channel contains a crossover which splits the signal into a tweeter channel and a woofer channel. PWM1, PWM2 output the tweeter. PWM3 and PWM4 output the woofer. By default AUXOUT3 outputs just the tweeter. It can be configured using Bit 5 in register 0x0122 to output the sum of the tweeter and woofer.

The sub channel is created by adding the left and right woofer channels and diving by two and low pass filtering. The woofer channel is unaffected by this. The sub channel is then looped back to the input muxes for output on another channel.

The Main Channel provides a loop back at two points (after Lip Sync and after Loudness) where some of the signal is then available as an input to any of the other muxes thereby providing a degree of processing to all other channels.

Headphone Channel

This has some basic processing; Volume, EQ and Loudness. It also simultaneously outputs on AUXOUT4.

AUXOUT1 Channel

This has a Volume control.

S/PDIF Channel

This has a Volume control. It can also be configured by changing Bit 9 in register 0x000C to output on SDO1.

SDO0 Channel

This has a Volume control.

INPUTS

Each of the five parallel channels has an output mute and an input mux; Main Mux, HP Mux, AUXOUT1 Mux, S/PDIF Mux, SDO0 Mux.

Each of these Muxes have the following ten inputs available;

ADC1
SDIN0
SDIN1
SDIN2/SRC2 Channel B
SDIN3/SRC2 Channel C
SRC1
SRC2 Channel A
Main Channel after Lip Sync (Not Available to Main Mux)
Main Channel after loudness (Not Available to Main Mux)
Sub Channel (Not Available to Main Mux)

The default for each Mux is no input.

SCENARIOS

The Default Flow has five channels, some with different output options and ten inputs per channel. There are a multitude of routing possibilities with this flow. Below there are a few examples of common scenarios.
All I^2C writes assume part is already fully powered up with relevant sections enabled and clocks correctly configured.

Analog In to Analog Out with no audio processing
34 0101 0010 #Route AUXIN1 to AUXOUT1
34 0121 0002 #Unmute AUXOUT1 output

Analog In to Analog Out with audio processing
34 0100 1000 #Route AUXIN1 to Main
34 0122 FDC1 #Enable AVC, Lip Sync, EQ, Bass, Loudness, Limiter, Crossover.
34 0122 FDE1 #Add the tweeter and woofer channels together
34 0121 0020 #Unmute Main Tweeter/AUXOUT3 output

Analog In to Analog Out with audio processing and with Sub Channel coming to Analog Out

```
34 0100 1000        #Route AUXIN1 to Main
34 0122 FDC1        #Enable AVC, Lip Sync, EQ, Bass, Loudness, Limiter, Crossover.
34 0122 FDE1        #Add the tweeter and woofer channels together
34 0101 00E0        #Route Subchannel to AUXOUT1
34 0121 0022        #Unmute AUXOUT1 output and Main Tweeter/AUXOUT3 output
```

Analog In to PWM Out with audio processing

```
34 0100 1000        #Route AUXIN1 to Main
34 0122 FDC1        #Enable AVC, Lip Sync, EQ, Bass, Loudness, Limiter, Crossover.
34 0121 00C0        #Unmute Tweeter and Woofer
```

Digital In (I^2S) to Digital Out (I2S) with audio processing

```
34 0100 3000        #Route SDIN0 to Main
34 0122 FC01        #Enable AVC, Lip Sync, EQ, Bass, Loudness
34 0101 D000        #Route Main (after loudness) to SDO
34 0121 0008        #Unmute SDO output
```

Digital In (I^2S) to Digital Out (S/PDIF) with no audio processing

```
34 0101 0300        #Route SDIN0 to S/PDIF
34 0121 0004        #Unmute S/PDIF output
```

Analog In to Digital Out (S/PDIF) with no audio processing

```
34 0101 0100         #Route AUXIN1 to S/PDIF
34 0121 0004        #Unmute S/PDIF output
```

Digital In (I^2S) to Analog Out with no audio processing

```
34 0101 0030        #Route SDIN0 to AUXOUT1
34 0121 0002        #Unmute AUXOUT1 output
```

CHANGING THE VOLUME

The volume works by multiplying the decimal contents of the volume control register by the signal. The volume control is a 28-bit word in 5.23 two's complement format. The ADAV4601 uses 16-bit registers therefore the volume control must be spread over two registers. The figure below graphically illustrates how this is arranged.

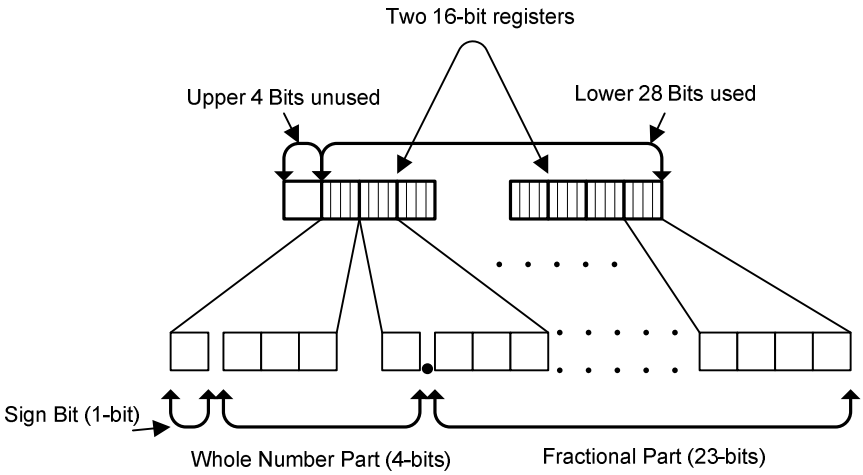


Figure 35. 28-Bit word across two 16-bit registers

To simplify updating the volume control, the I²C address pointer auto-increments when writing and reading. This means that the volume control register can be updated in a single I²C block write. Therefore, it is recommended that the volume control is updated using the following I²C write format:

<device address> <0114> <32-bit data transfer>

Note that the volume control is a 32-bit parameter; therefore, both Register 0x0114 and Register 0x0115 must be written to. Writing anything less than the 32 bits to these registers will not update the parameter.

The default volume is 1 or 0dB which cause no change to the signal. The I²C write for this is:
34 0114 00800000

EXAMPLE: INCREASE THE MAIN VOLUME TO +6DB OVER DEFAULT

1. The volume works by multiplying the decimal contents of the register by the signal. Therefore in order to increase by 6dB, the multiplication factor equivalent to 6dB must be found. Using some algebra this can be easily found.

$$20Log_{10}x = 6dB$$

$$Log_{10}x = \frac{6}{20}$$

$$x = 10^{\frac{6}{20}}$$

$$x = 1.995262315$$

2. Convert this decimal number to X.23 format by multiplying by 2^{23}. (This simplifies the conversion to Hex that would otherwise be difficult due to the fraction)
   $1.995262315*2^{23} = 16737473$

3. Then simply convert this 5.23 decimal number to Hexadecimal.
   0x00FF64C1

4. Write this value to the registers.
   34 0114 00FF64C1

The same idea applies to other 28-bit words that are spread over two registers.

SECTION 5 – INTERFACING THE ADAV4601 TO A SYSTEM

There are three options for interfacing the ADAV4601 to a system.

1. Using the Default Flow
2. Using a Custom Flow
3. Using a Custom Flow with the Application Layer

USING THE DEFAULT FLOW

The ADAV4601 has a default audio flow on board and was previously shown in Figure 34. The register map for this flow is in the Default Audio Flow Registers section in Appendix A – Detailed Register Descriptions. The default flow is very simple to control requiring only one I²C write to change parameters of filters, volumes, gains, etc.
The ADAV4601 is controlled via I²C; therefore the following sequence is given in the following format

34 AAAA DDDD

Where 34 is the I²C address of the ADAV4601, AAAA is the register address in the ADAV4601 that is being written to and DDDD is the data that is being written.

Default Flow Example

Change the low and high frequency of the crossover (Address 0x0109 Crossover Register (Default: 0x0505)) to 110Hz. The I²C write required is

34 0109 0606

No safe loading to registers, which prevents pops, clicks and glitches, needs to be considered as this is all taken care of automatically.

USING A CUSTOM FLOW

Interfacing a custom flow requires a number of steps and are illustrated in the figure below.

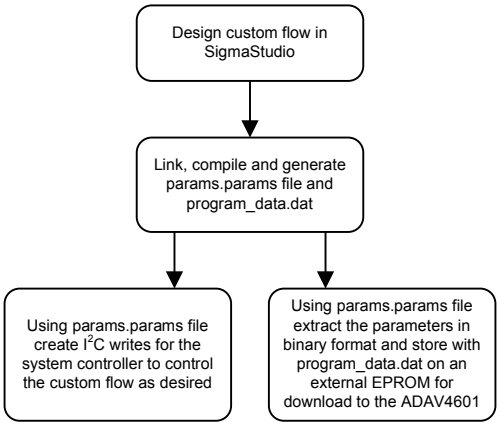


Figure 36. Using a Custom Flow

It is important to note that when updating parameters or slews in the flow to use safe loading. An example of this is given below.

Every block (volume, filter, mux, etc) available in SigmaStudio is a cell and is defined in the **params.params** file generated by SigmaStudio. An example of a cell defined in **params.params**, in this case a general 2nd order low pass filter with a cut off of 1kHz, is shown below.
Where:

Cell Name:	Indicates that the following parameter information belongs to that cell.
Parameter Name:	A cell may have more than one parameter associated with it and this identifies it.
Parameter Address:	The location of the parameter in parameter memory in decimal.
Parameter Value:	The value of the parameter in decimal.
Parameter Data:	The value of the parameter in hexadecimal 5.23 format.

Cell Name = Gen Filter1
Parameter Name = EQ1940Single10B1
Parameter Address = 0
Parameter Value = 0.00408828258514404
Parameter Data :
0X00, 0X00, 0X85, 0XF7,

Cell Name = Gen Filter1
Parameter Name = EQ1940Single11B1
Parameter Address = 1
Parameter Value = 0.00817668437957764
Parameter Data :
0X00, 0X01, 0X0B, 0XEF,

Cell Name = Gen Filter1
Parameter Name = EQ1940Single12B1
Parameter Address = 2
Parameter Value = 0.00408828258514404
Parameter Data :
0X00, 0X00, 0X85, 0XF7,

Cell Name = Gen Filter1
Parameter Name = EQ1940Single11A1
Parameter Address = 3
Parameter Value = 1.8951700925827
Parameter Data :
0X00, 0XF2, 0X94, 0XEF,

Cell Name = Gen Filter1
Parameter Name = EQ1940Single12A1
Parameter Address = 4
Parameter Value = -0.911523461341858
Parameter Data :
0X0F, 0X8B, 0X53, 0X33,

Custom Flow Example:

The example that follows goes through the steps required to change the filter cut off to 2kHz.

1. The system controller provides the filter coefficients, b_0 b_1 b_2 a_1 a_2, in 5.23 format.
   b_0 = 0.0156048536300659 -> 0x0001FF57
   b_1 = 0.0312097072601318 -> 0x0003FEAE
   b_2 = 0.0156048536300659 -> 0x0001FF57
   a_1 = 1.76945173740387 -> 0x00E27D65
   a_2 = -0.831871271133423 -> 0x0F95853E

2. The filter coefficients will have to be safe loaded. See the Safe Loading to Parameter RAM and Target/Slew RAM section for more details. Move the coefficient value into the safeload parameter registers.

34 2040 0001FF57
34 2041 0003FEAE
34 2042 0001FF57
34 2043 00E27D65
34 2044 0F95853E

3. Move the addresses of the respective filter coefficients into the safeload addresses registers.

34 2045 0001
34 2046 0002
34 2047 0003
34 2048 0004
34 2049 0005

4. Initiate the safe load by setting Bit 4 in Register 0x0200 to 1.

34 0200 0010

ROMS and Registers

When the ADAV4601 is configured for use with a custom flow it contains two ROMS: program and parameter. Both can be stored externally on an EEPROM and can be loaded after power-up.

Program ROM

Program ROM is **42-bits wide** and occupies **Address 0x1400 to Address 0x1FFF**. This is where the audio flow generated in SigmaStudio is stored.

Parameter ROM

Parameter ROM is **28-bits wide** and occupies **Address 0x1000 to Address 0x13FF**. Default parameters for default flow and custom flow are stored here.

Safe Loading to Parameter RAM and Target/Slew RAM

Up to five safe load registers can be loaded with parameter RAM address data. The data is transferred to the requested address when the RAM is idle. It is recommended to use this method for dynamic updates during run time. For example, a complete update of one biquad section can occur in one audio frame. This method is not available for writing to the program RAM or control registers.

There are ten safe load registers operating in pairs of five, where five of them store addresses and five of them store data. To safe load a register, move its address into a safe load address register and move its data into the corresponding safe load data register. If it is a parameter RAM, set **Bit 4 in Register 0x0200 to 1** to initiate the safe load. If it is a target/slew RAM, set **Bit 5 in Register 0x0200 to 1** to initiate the safe load.

The safe load data registers are located from **Address 0x2040 to Address 0x2044** and are five-bytes wide.

The safe load address registers are located from **Address 0x2045 to Address 0x2049** and are two-bytes wide.

The last five instructions of the program RAM are used for the safe load process; therefore, the program length should be limited to 2555 cycles (2560 – 5). It is guaranteed that the safe load occurs within one LRCLK period (21 µs at f_S = 48 kHz) of the initiate safe transfer bit being set. Safe load only updates those safe load registers that have been loaded with new data since the last safe load operation. For example, if only two parameters or target RAM locations are updated, it is only necessary to load two of the safe load registers; the other safe load registers are ignored because they contain old data.

USING A CUSTOM FLOW WITH THE APPLICATION LAYER

We strongly recommend this approach. The advantage of using the Application Layer is that it automatically takes care of safeloading. It dramatically reduces the amount of I²C writes to the part freeing up the I²C bus in your system and simplifying microcontroller code.

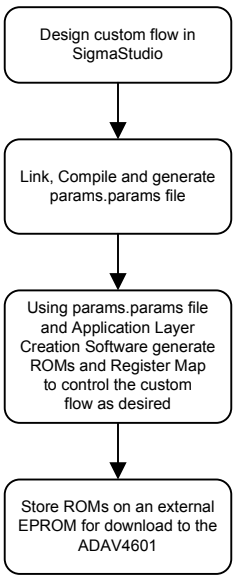


Figure 37. Using a Custom Flow with the Application Layer

Custom Flow with Application Layer Example

A general 2nd order low pass filter will have a number of filter cut off frequencies and an address defined by the user in the Application Layer Software. In order to change the cut off frequency simply write to address of the register and change it to one of the pre-defined cut offs. For this example assume the address is 0x1000 and a cut off of 2kHz is assigned to a value of 0xF, then write required to change the filter cut off is

34 1000 000F

This only requires one I²C write compared to the previous example which took eleven I²C writes.

ROMS and Registers

When the ADAV4601 is configured for use with the Application Layer it contains four ROMS: program, instruction, parameter, and LUT. All can be stored externally on an EEPROM and can be loaded after power-up.

Program ROM

Program ROM is **42-bits wide** and occupies **Address 0x1400 to Address 0x1FFF**. This is where the audio flow generated in SigmaStudio is stored.

Instruction ROM

Instruction ROM is **33-bits wide** and occupies **Address 0x3000 to Address 0x327F**. This is where the application layer register map is stored.

Parameter ROM

Parameter ROM is **28-bits wide** and occupies **Address 0x1000 to Address 0x13FF**. Default parameters for custom flow are stored here.

LUT ROM

LUT ROM is **28-bits wide** and occupies **Address 0x4000 to Address 0x57FF**. This contains the remaining predefined parameters for the custom flow.

READ/WRITE DATA FORMATS

The read/write formats of the control port are designed to be byte oriented. This allows easy programming of common microcontroller chips. To fit into a byte-oriented format, 0s are appended to the data fields before the MSB to extend the data-word to eight bits. For example, 28-bit words written to the parameter RAM are appended with four leading 0s to equal 32 bits (4 bytes); 40-bit words written to the program RAM are not appended with 0s because they are already a full five bytes. These zero-padded data fields are appended to a 3-byte field consisting of a 7-bit chip address, a read/write bit, and an 16-bit RAM/register address. The control port knows how many data bytes to expect based on the address given in the first three bytes.

The total number of bytes for a single location write command can vary from five bytes (for a control register write) to eight bytes (for a program RAM write). Burst mode can be used to fill contiguous register or RAM locations. A burst mode write begins by writing the address and data of the first RAM or register location to be written to. Rather than ending the control port transaction (by issuing a stop command in I^2C mode), as would be done in a single-address write, the next data-word can be written immediately without specifying its address.

SECTION 6 – THE APPLICATION LAYER CREATION SOFTWARE

THE APPLICATION LAYER

The purpose of the Application Layer is to simplify system design and reduce time to market. It achieves this by drastically reducing I²C writes while still maintaining full control over a customised audio flow in the ADAV4601. What this means to the user is the chance to have complicated functions in the audio flow and yet simple register writes to control these functions. This will drastically simplify the microcontroller code for the user and also means that the I²C bus will not be "clogged" with unnecessary writes.
Physically the Application Layer is a hardware module on-chip that interprets the output from the Application Layer Creation Software and allows the control described above.

APPLICATION LAYER CREATION SOFTWARE

The aim of the Application Layer Creation Software is to create an upper layer of registers, simpler than the register map created by SigmaStudio. Once a project is finished in SigmaStudio, it gives a full register map to control every block in the project. The blocks are controlled by the SigmaDSP parameters, in the Parameter RAM. Dealing with this is usually too complex and not easily readable. Here is where the Application Layer deals with this complexity, offering a simple custom register map that will perform all the complex operations needed. All the reads and writes are done through the I²C bus. See Figure 38.

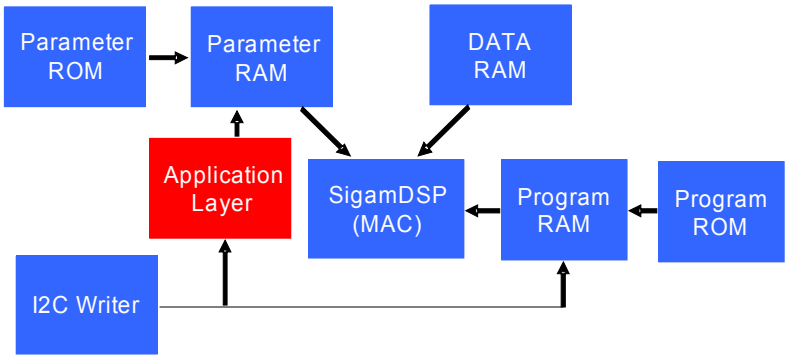


Figure 38. ADAV4601 Hardware Block Diagram

In order to define the Application Layer, the Application Layer Creation Software can import any SigmaStudio flow, detect the blocks present in the project, and then the user can define a custom register map for them. Once this is done, the software can compile the code and create the Application Layer ROMs. These ROMs can be downloaded directly to the Application Layer RAM from the software provided (SigmaStudio), and in a system design can be booted an external I²C EEPROM memory.

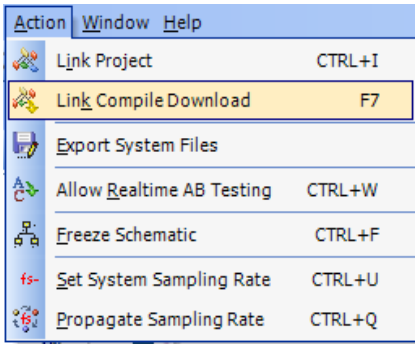
In addition to the ROMs, the Application Layer Creation software will produce:

- HTML Register map documentation. Useful to know how to control the register map that has been created.
- Standard 'C' code library. Ready to be embedded in any system with an I²C master controller, facilitating the control (read and write) of the registers present in the map. 'C' functions for every field, in every register, are produced.

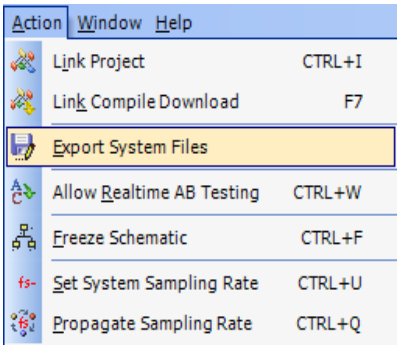
LINKING THE APPLICATION LAYER SOFTWARE TO YOUR AUDIO FLOW

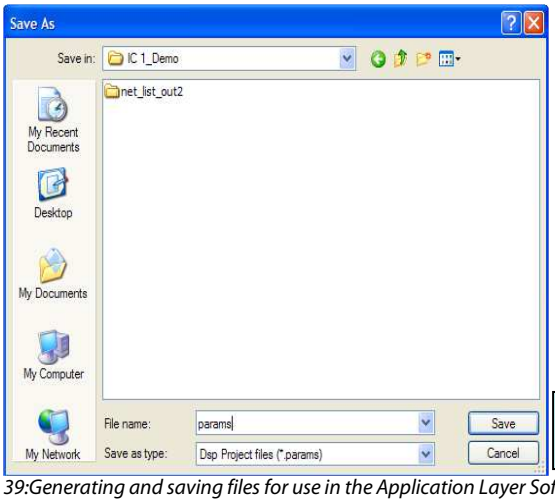
To create an Application Layer for the new SigmaStudio Audio Flow, the user must generate a number of files from SigmaStudio. These files are used by the Application Layer Software.

Generate the project data files in /project_folder/IC 1_name_of_folder



Generate params.params file. Save it in /project_folder/IC1_name_of_folder





Input "params" as file name. Save it in /project_folder/IC1_name_of_folder

Figure 39:Generating and saving files for use in the Application Layer Software

APPLICATION LAYER SOFTWARE GUI

To start the program, click ***Start -> All Programs -> Analog Devices -> Analog Devices Application Layer -> Application Layer***. Once loaded, the software GUI looks like Figure 40 below. For more information about each function, please refer to the User Manual section.

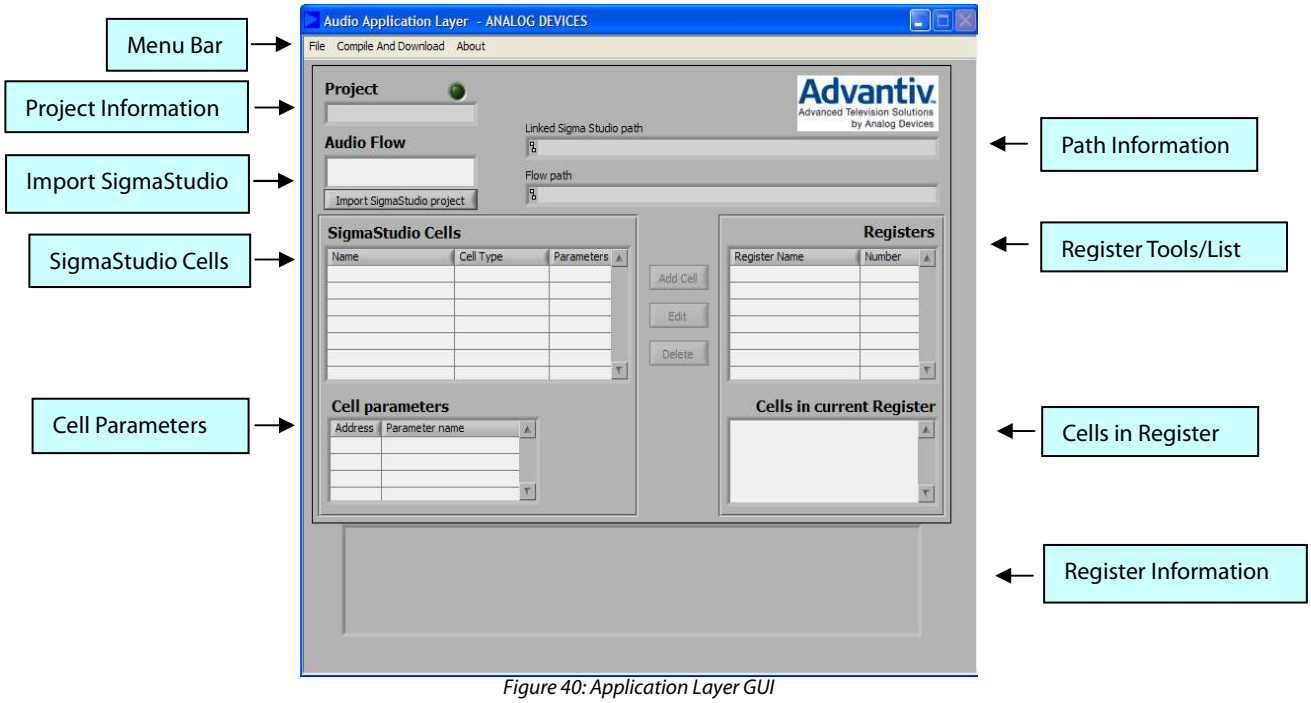


Figure 40: Application Layer GUI

Table 8: Application Layer GUI

Menu Bar	Project functions (load, save, compile, download, etc.)
Project Information	Current project.
Import SigmaStudio	Allows a link from a SigmaStudio project to the Application Layer.
SigmaStudio Cells	Interactive list with all the cells automatically extracted from the SigmaStudio project.
Cell Parameters	Information window with the parameters related to the cell selected in the listbox above.
Path Information	Path where the project is saved, and path to the linked SigmaStudio project.
Register Tools	Tools to manipulate the registers.
Registers List	Interactive list of the current registers defined in the project
Cells in Register	Interactive list of the cells defined in the selected register.
Register Information	Displays information of the selected register.

USER MANUAL

Menu Bar: File Menu

This menu is related to the project and files management.

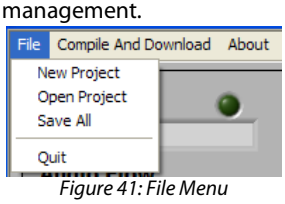


Figure 41: File Menu

New Project

Creates a new project set of files. By default, projects are blank. The next step will be importing a SigmaStudio set of files. A workspace where the project folder will be created and a project name will be requested.

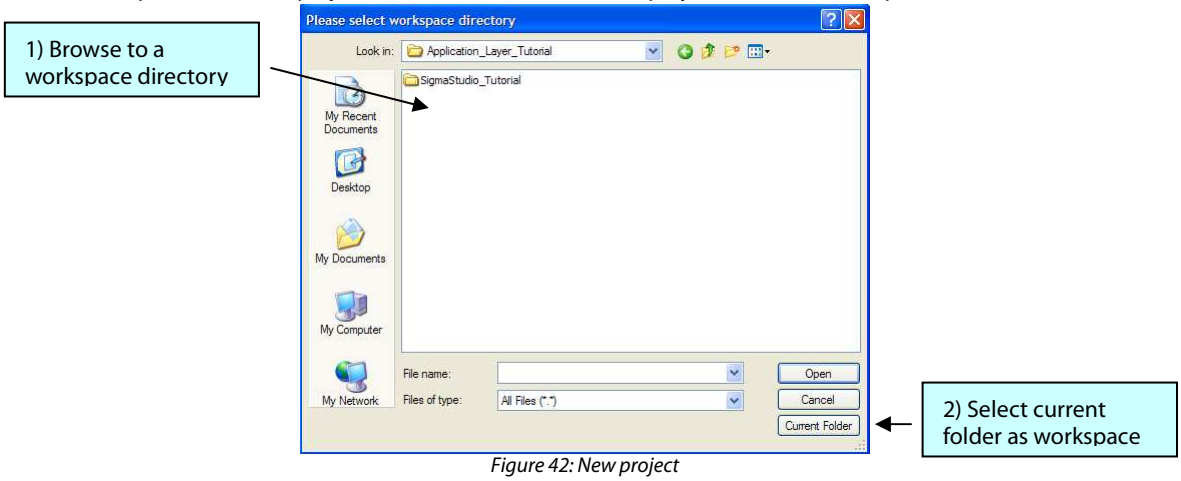


Figure 42: New project

Open Project

Loads a previous project from disk. Application Layer GUI project files have the **.grp** extension.

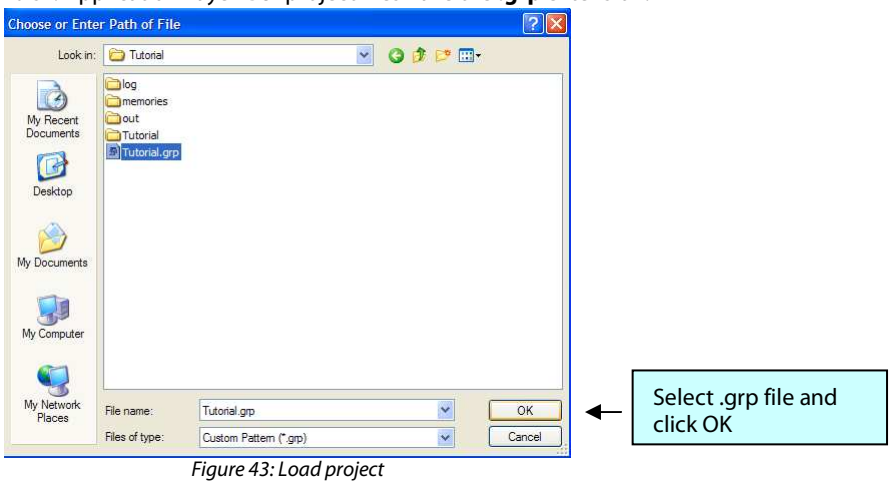


Figure 43: Load project

Save All

Saves current project to disk (auto-save mode by default).

Quit

Exits from the application.

Compile and Download

The functions related to the project data output generation are here.

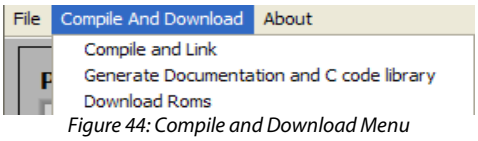


Figure 44: Compile and Download Menu

Compile and Link

Compiles and links the project. This will generate the ROM files matching the audio flow created, they will be placed on ***/project_directory/memories***.

Generate Documentation and C Code Library

Automatically generates documentation in HTML format, and standard C code library. Files are placed on ***/project_directory/Register_map.html*** and ***/project_directory/audio_hal_adi.c.***

Download ROMs

Download the ROMs to the on-chip Application Layer memories, through the USB cable. This will automatically take the ROMS from the ***/project_directory/memories*** folder. Please note that ROMS must have been generated before this step.

Project Information

Once a project is open, the project's flow name is displayed in this list, see Figure 45. Clicking over the **Audio Flow** name (**Tutorial** in the example below), the flow information windows will be populated (**SigmaStudio Cells** and **Registers**). There are some file and version checks behind this function. If the GUI doesn't pop up any message, and populates the **SigmaStudio Cells** and **Registers** lists, the current set of files is correct and ready to use.

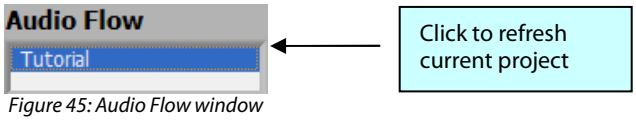


Figure 45: Audio Flow window

Note: Possible error messages are:

- Newer program_data file found on linked SigmaStudio folder : a newer version of the SigmaStudio project has been detected on the Linked SigmaStudio Path, it's recommended to Import SigmaStudio again, so the files keep synchronized. This happens when the project is recompiled in SigmaStudio, after being imported to the Application Layer GUI.
- Newer parameters file found on linked SigmaStudio folder : a newer version of the *params.params* file was found on the *Linked SigmaStudio Path*, it's recommended to Import SigmaStudio again, so the files keep synchronized. This happens when a *params.params* file is regenerated after the project was imported to the Application Layer GUI.
- No registers defined: the project is blank, no registers have been created.
- Files missing. Please import SigmaStudio project to flow: the SigmaStudio set of files is missing, please import them as shown in Figure 46.

Import SigmaStudio

The first step after creating the basic structure of the Application Layer project is to import a SigmaStudio project to the Application Layer GUI. This might be needed as well, if any updates on the SigmaStudio project are done. As shown in the previous section, after updating the project details, there are some checks to determine if any user action should be taken.

Update the project information (see previous section) and click **Import SigmaStudio Project**:

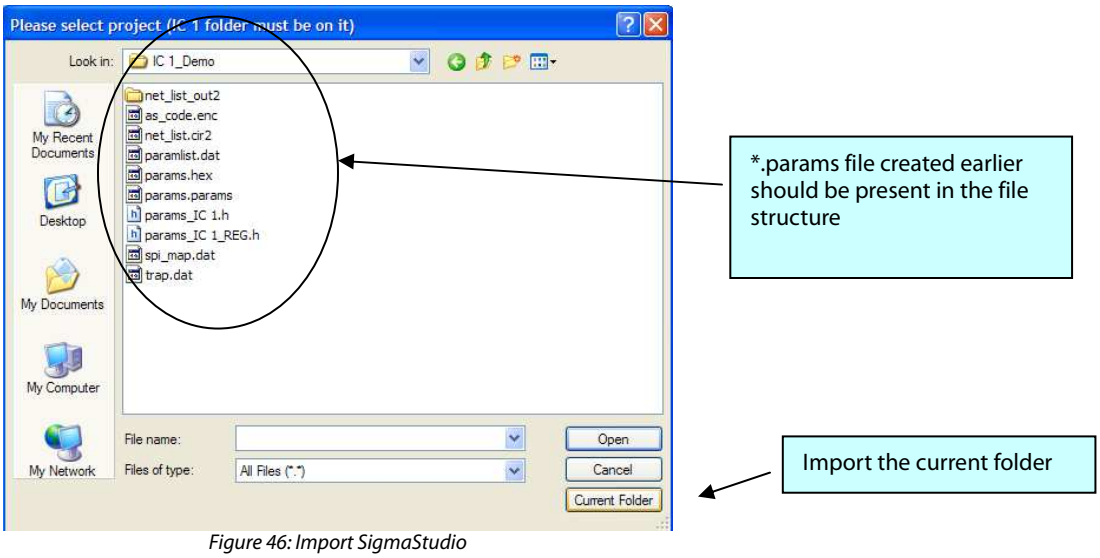


Figure 46: Import SigmaStudio

The possible list of messages, and action to take:

- Error checking files . Please verify that *.params is on IC1 folder : the file with the parameters definition can't be found, a SigmaStudio project was imported but the file was missing. Generate the *.params files as shown in the first section, and reimport.

- *Parameters file is older than program.data file. Please regenerate params file from SigmaStudio :* the *params.params* file is older than the *program.data*, which means that the SigmaStudio project was recompiled after the latest parameters file was generated. This might cause a mismatch between both files. Regenerate the parameters files from SigmaStudio as shown in the first section, and reimport.

Application Layer Register Structure

A register created in the Application Layer GUI is able to control one or more SigmaStudio cells. Registers have a unique name and address, and are always 16-bits long. Bits are grouped into inputs. An input is a field in the register, which calls the Application Layer when is written to. Register name, and input names, are given by the user when creating the register map.

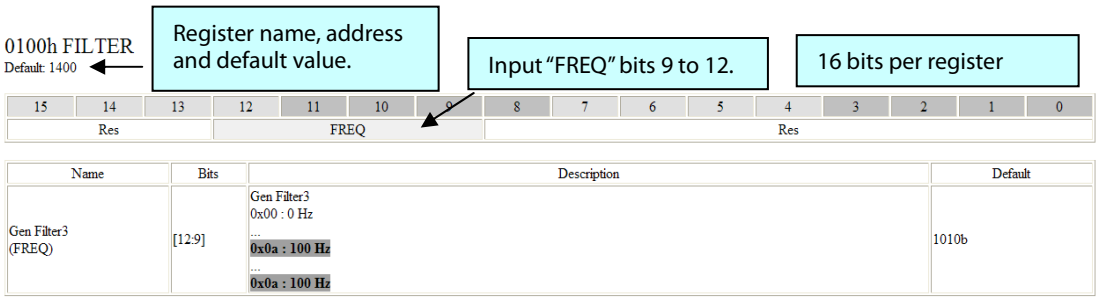


Figure 47

SigmaStudio Cells and Cell Parameters

This interactive list has the SigmaStudio Cells extracted from the SigmaStudio project. A cell is a block in the SigmaStudio project, which is controlled by a set of parameters. Figure 48 below shows an example of a filter.

Clicking the cells in the list, the **Cell parameters** window is updated with the parameters related to that cell. Once a cell is selected, it can be added to a register using the **Add cell to register** function.

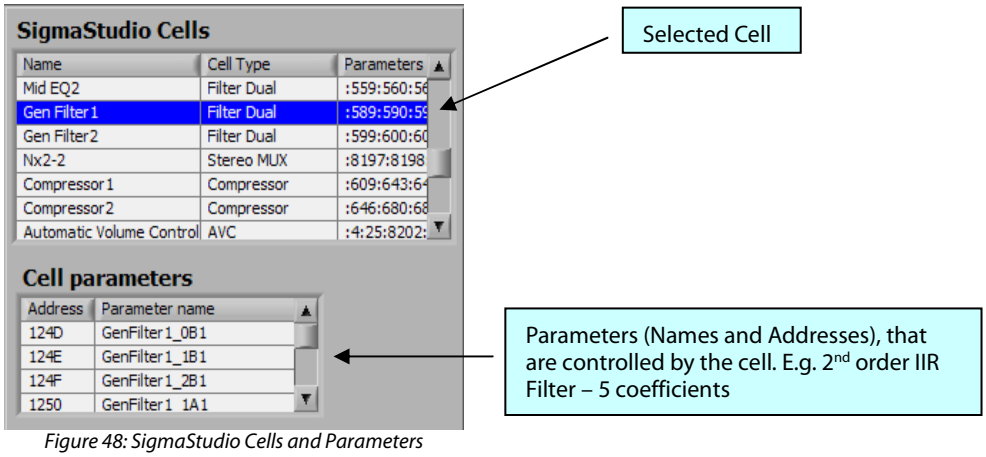


Figure 48: SigmaStudio Cells and Parameters

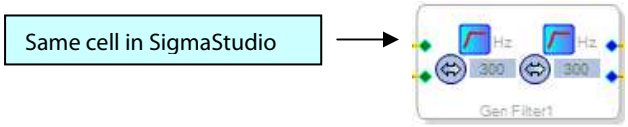


Figure 49: The cell as it appears in SigmaStudio

SigmaStudio Cells list has three elements:

***Name*:** Name of the cell, matches the SigmaStudio cell name, and is set by the user when developing the audio flow in SigmaStudio. In the example above, *Gen Filter1* is the name of the highlighted cell. ***Note*:** Just like variables when programming in C or any other programming language, Cell names in SigmaStudio should be given names descriptive of their purpose. This makes it easier for the user when creating their Application Layer.

***Cell Type*:** Detected cell type. This value is used by the register creation wizards, so depending on the type, a wizard matching it will be loaded. The selected cell above (Figure 48), will load a Filter Wizard, since its type is *Filter Dual*.

***Parameters*:** List of parameters related to the cell (internal SigmaStudio numbers). The *Cell parameters* window below the *SigmaStudio Cells* windows (Figure 48), shows more details about them.

Cell Parameters list has two elements:

***Address*:** Actual parameter memory address in the parameter memory map (hexadecimal value). Writing to this value directly would change the parameter value. This should be only done for development and debug purposes, and is not supported from the Application Layer GUI.

Parameter Name: Name of the selected parameter.

Path Information

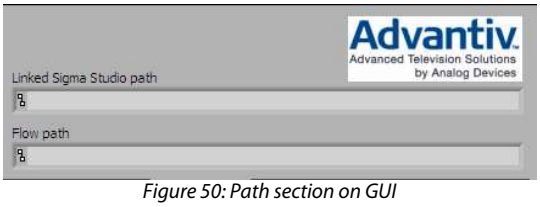


Figure 50: Path section on GUI

Linked SigmaStudio path

Path where the SigmaStudio master project is. Please note that when importing, a local copy of this folder is created. If any change is done in the master project, the Application Layer GUI will detect it.

Flow path

Path where the Application Layer GUI project files are saved.

Register Tools

There are the three basic operations that the user can do to create and manipulate the custom audio flow registers.

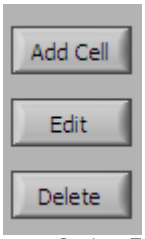


Figure 51: Register Tools

Add Cell to a Register

This is the basic function to add a cell to a register. Select the cell to be added, and the destination register. This function will call the associated Wizard, depending on the detected *cell_type*. The destination register can be either a new register, or one previously created.

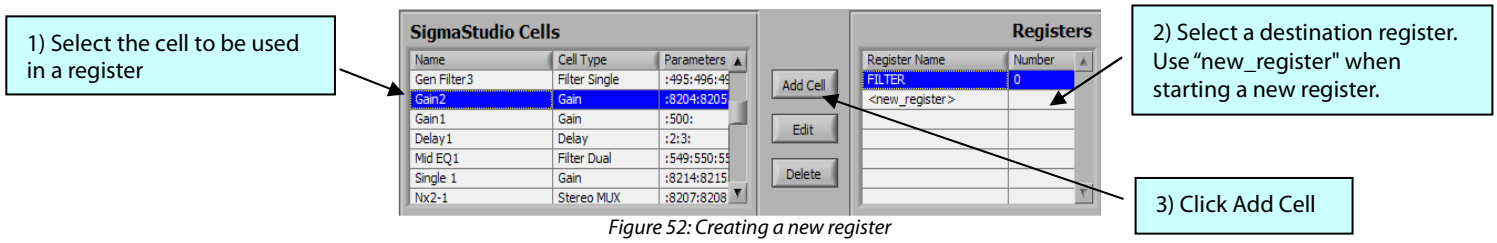


Figure 52: Creating a new register

When a cell is added to a *new_register*, the software will request a Register Name (name in the register map), and an Input Name (input name that will control the SigmaStudio cell). Input names will always be fields inside the register, where the maximum size is 16 bits per register. By default, the new register number will be the lowest available, and it's assigned automatically. It can be changed, see the Edit Register section for more information about the registers structure, see Application Layer Register Structure section.

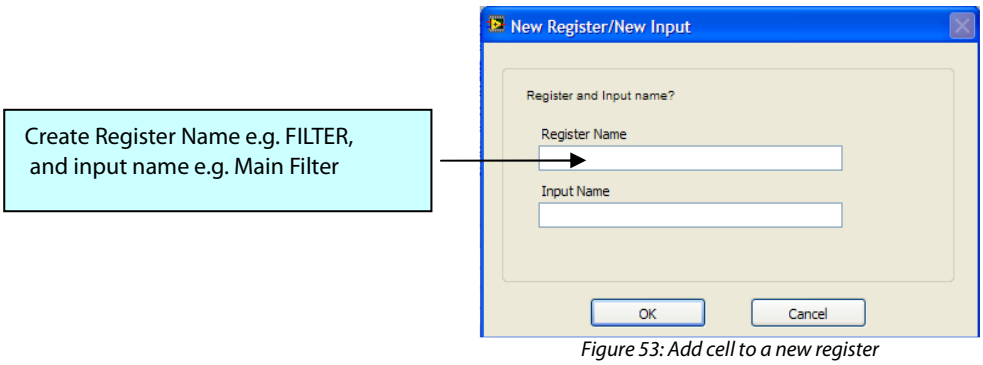


Figure 53: Add cell to a new register

When the cell is added to a previously created register, the user should ensure there is enough free space for the new field. The GUI will display a window with the current register definition. Any available space is named as "UNASSIGNED". After this, please input a name for the Input field.

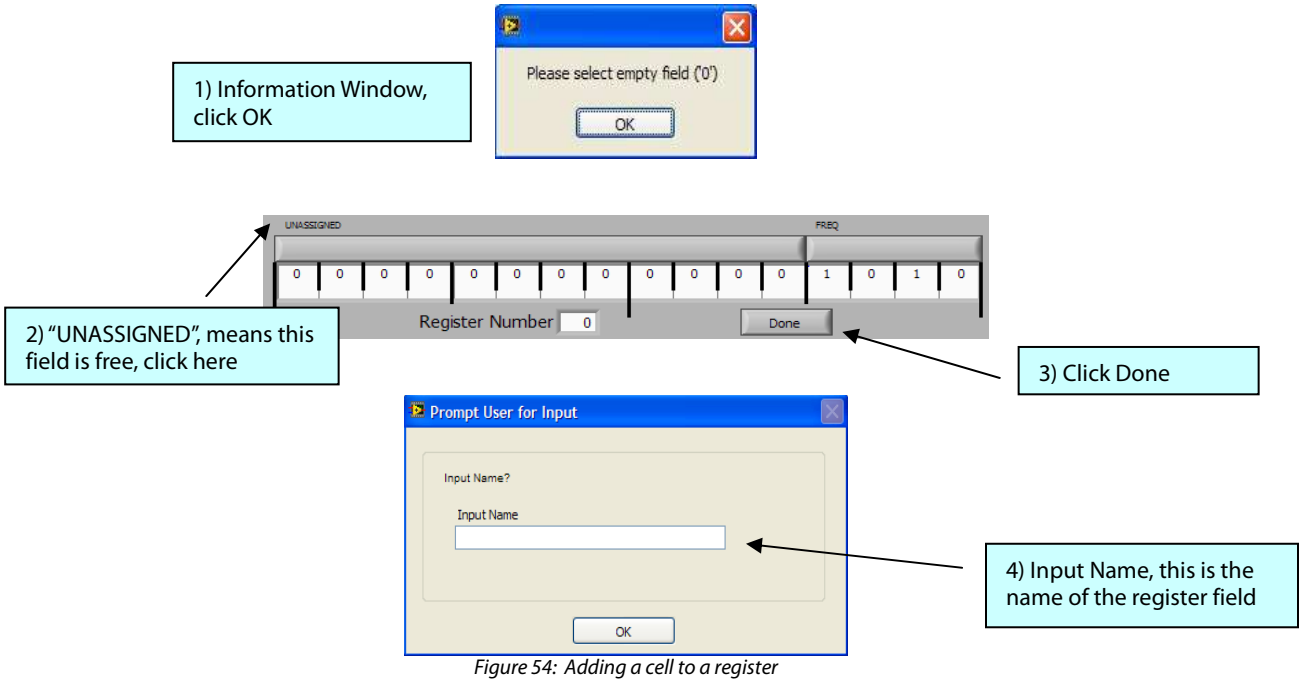


Figure 54: Adding a cell to a register

After these steps, the wizard associated with the cell type will be called. A single register has a size of 16 bits. The wizard will manage the available free bits in the register (UNASSIGNED bits). When creating a control in a "new register", the available number of bits will be 16. Depending on the configuration of the cell control, the number of bits needed, will change. The wizard will know how many bits are available for the input to be created, and in case it doesn't fit it will show an error.

Edit Register

From this command the user can edit an already created register. This allows the user to change the register number, and to shift the register fields within the register. Select one of the registers, and click *Edit,* which will launch the register editing window.

To change the register number, enter the new one in the Register Number field. To shift a field, click over it, and then click over the MSB of the desired destination position. Shifting fields may be useful before adding new fields to predefined registers.

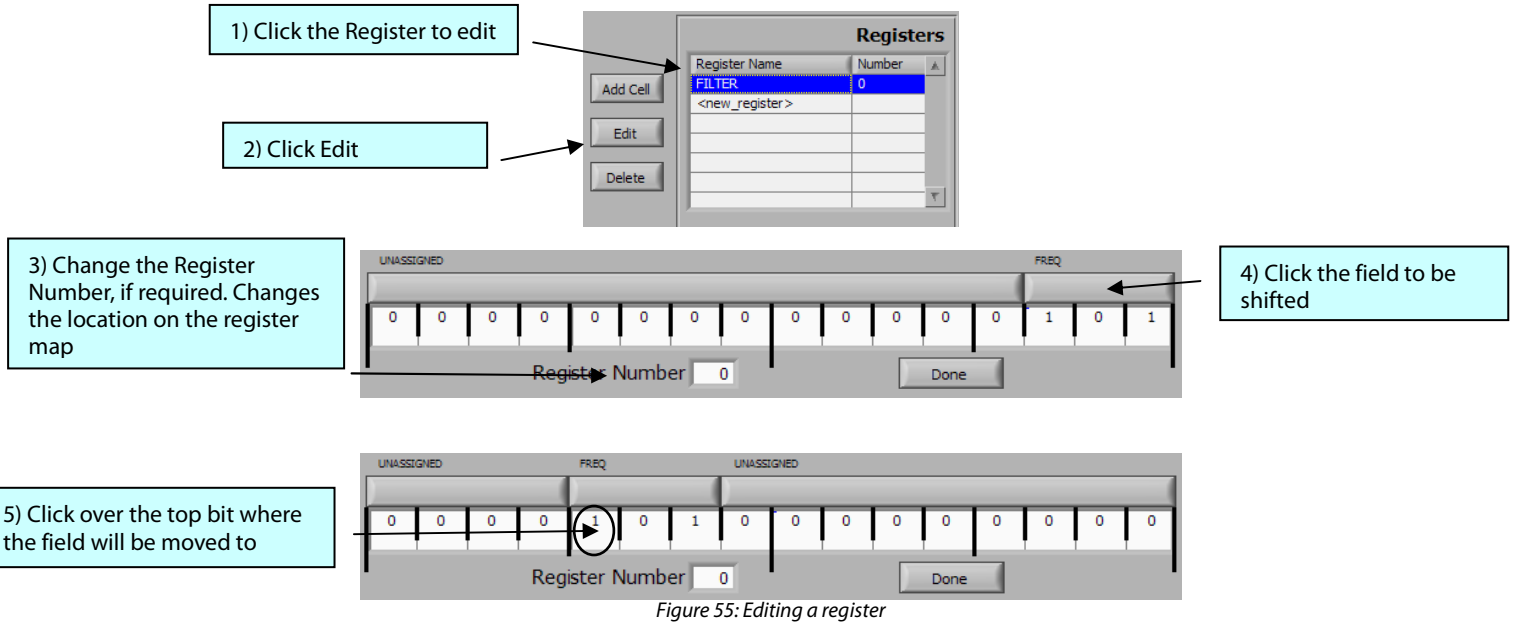


Figure 55: Editing a register

Delete Register

Delete the selected register from the register map. The GUI will ask for confirmation before deleting any register.

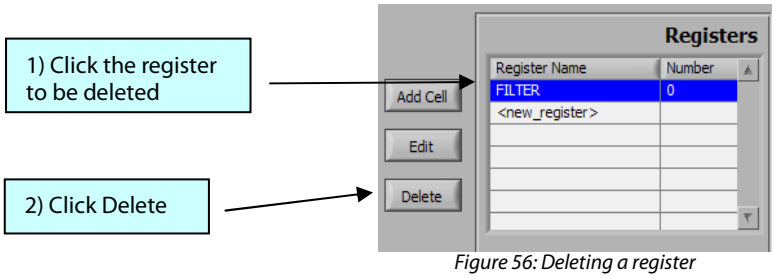


Figure 56: Deleting a register

Registers List and Cells in Current Register

These interactive windows control the register map, and the cells that each register controls. Clicking over *Registers list* will change the selected register. Each register has a register name, and a register number. The numbers are automatically assigned (lowest available), and can be changed as described in the last section.

Editing a Cell Previously Defined

Clicking a register, updates the "*Cells in current Register*" list. It is possible to edit a cell by double-clicking. This will call the wizard associated to the cell type. The wizard will load the values that the cell had last time it was modified. For more information about the wizards, refer to The Cell Wizards section.

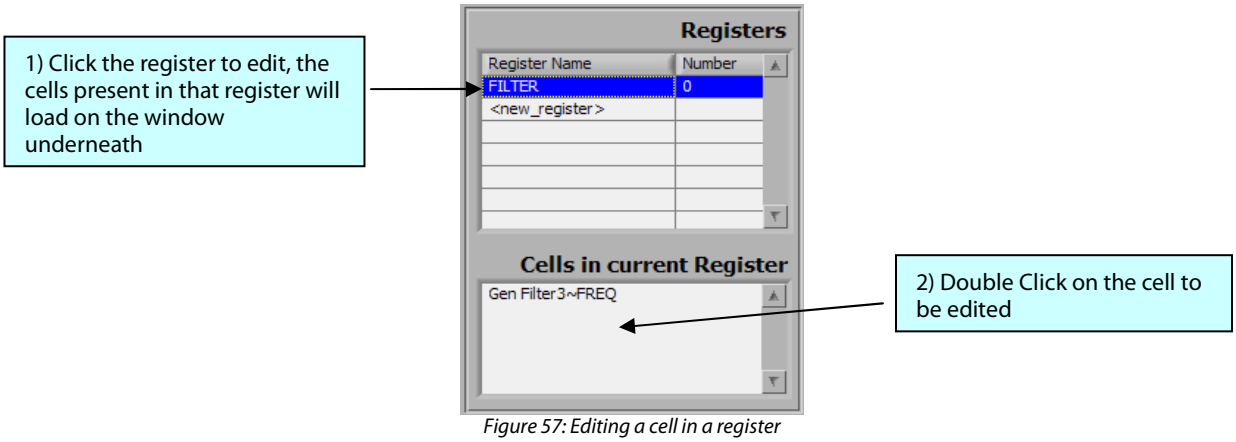


Figure 57: Editing a cell in a register

Register Information

This window displays the current status of a register. It displays the register fields, and the default bit values. A field named "UNASSIGNED" means the field is unused. Click a register in the *Register list* and it will display it automatically.

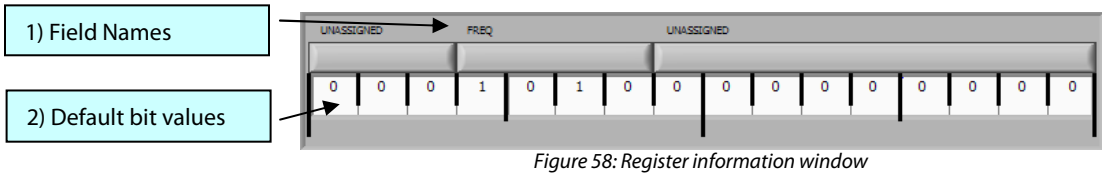


Figure 58: Register information window

THE CELL WIZARDS

The Cell Wizard allows the user to define the range of values for each parameter controllable in their audio flow. For example the cell Wizard for a filter will allow the user to specify the range of cut off frequencies and gains for that specific filter. Each cell type in SigmaStudio has a wizard associated with it in the Application Layer Software. The GUI will recognise the cell type, and it will run the wizard that matches this cell type.
The wizards allow the user to create new registers and to edit previously defined ones. They also auto-generate the information that is needed to generate the HTML documentation and the standard C library. Field size is automatically managed by the GUI, if the cell control size exceeds the number of unassigned bits, the GUI will report an error message. Wizards have some basic information value boxes:

Cell name: Name of the SigmaStudio cell selected.
Register name: Name of the register, given by the user.
Bits: Number of bits needed by the field, this value will depend on the configuration of the cell control.

Every Wizard will set a default value once a register is created. This default value is selected by the user, except in the *Import Code* blocks.

Note: It is absolutely essential that the default value for a certain cell in the SigmaStudio Audio flow should match the default value set by the user in the Application Layer software. This will ensure consistency in the users design.

Volume / Gain Wizard

This wizard controls the *Volume* and *Gain* cells created in SigmaStudio. It supports both single and multiple inputs (SigmaStudio blocks can control single or multiple inputs within the same block). The control will operate over all the volume parameters present in the cell.

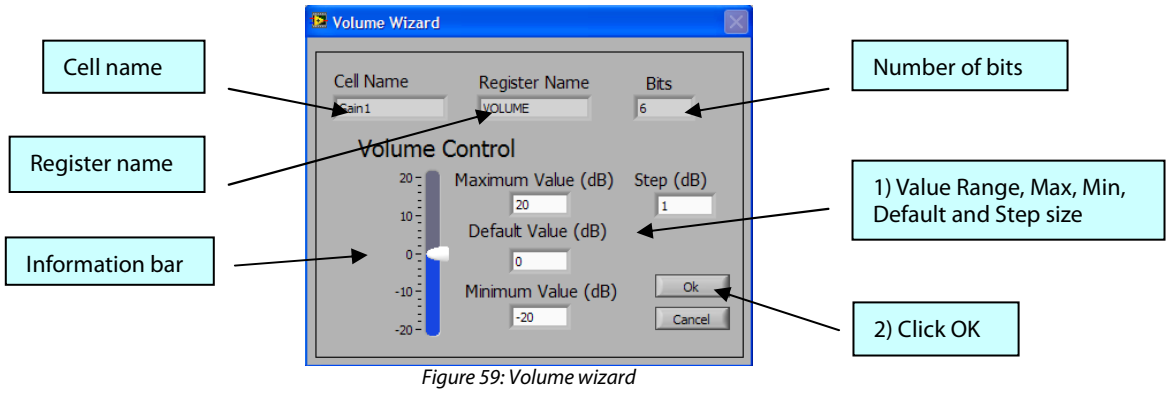


Figure 59: Volume wizard

Note: When selecting the volume or gain block in SigmaStudio, choose the shared volume control. This uses a single parameter for the volume control. This is more efficient and easier to control using the Application Layer. The other volume controls are not supported.

Filters Wizard

This wizard controls any *filter* cell created in SigmaStudio. It automatically supports single and multiple filter cells. The filters can be controlled by cut off frequency (gain will be a fix value for the frequency range) or by gain (cut off frequency will be fix for the gain range).

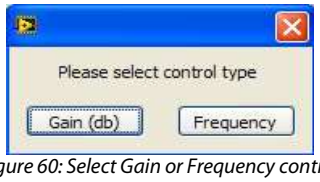


Figure 60: Select Gain or Frequency control

The filter selected (either gain or frequency controlled) should have as well a filter shape or type.

Filter Wizard: Gain Control

A filter controlled by gain will have all the frequency related parameters fixed. Writing to the gain value field, the gain response of the filter will change.

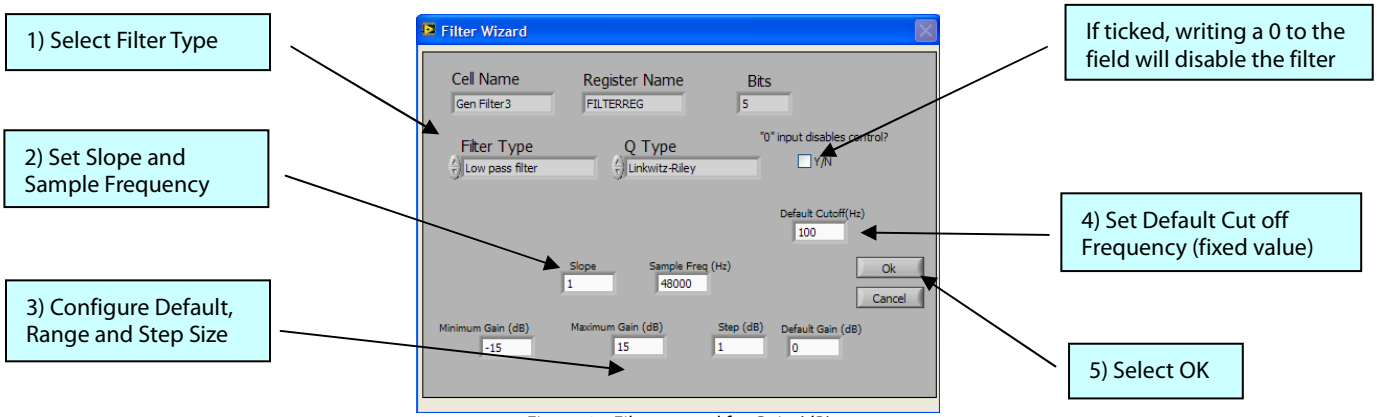


Figure 61: Filter control for Gain (dB)

Filter Wizard: Frequency Control

A filter controlled by frequency will have all the gain related parameters fixed. Writing to the cut off frequency value field, the cut off frequency response of the filter will change.

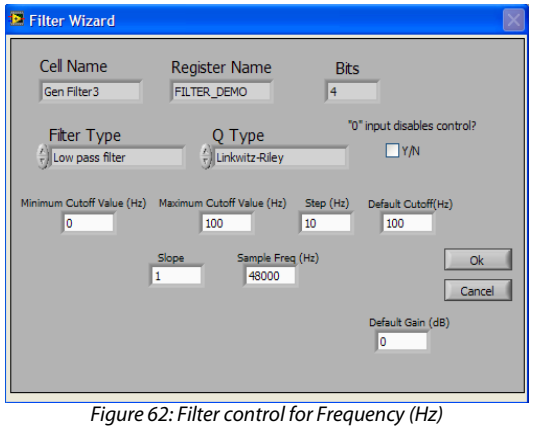


Figure 62: Filter control for Frequency (Hz)

Filter Type Selection

Irrespective of the filter control chosen (gain or frequency), the filter type has to be configured. This will basically dictate the filter shape and the Q-type that will define the behaviour of the filter response. To do so, select the values from the drop menu:

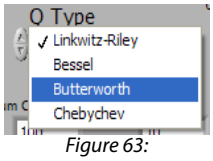


Figure 63:

Multiplexers Wizard

This wizard allows control over a mux. The Application Layer GUI supports the muxes in SigmaStudio under the TV Algorithms library.

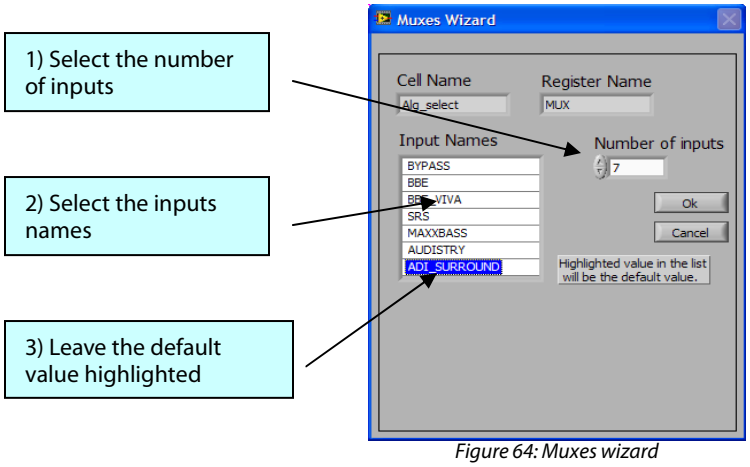


Figure 64: Muxes wizard

Delay Wizard

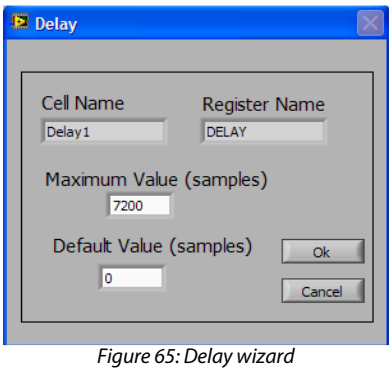


Figure 65: Delay wizard

The delay wizard creates a single or multiple delay block control.

Loudness Wizard

The loudness wizard creates a loudness block control. This control will have two fields in the register, one to control the cutoff frequency and one to control the level. The wizard will automatically create these two fields, and assign the suffixes *_LFC* and *_LLEVEL* to generate the two inputs.

1) Select the maximum, default and minimum values along with step sizes

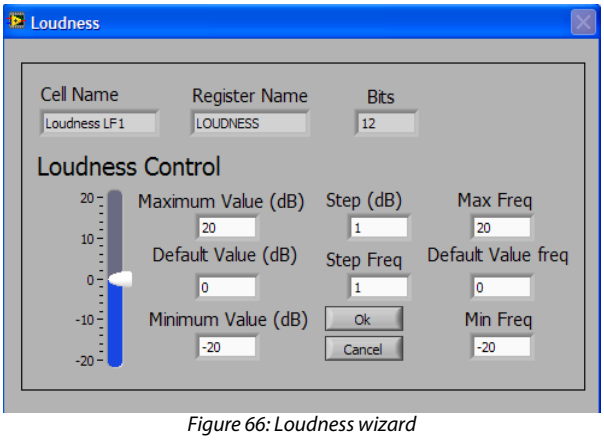


Figure 66: Loudness wizard

Clipper Wizard

Controls the clipper blocks. This wizard is very similar to the volume / gain wizard.

1) Select the maximum, default and minimum values along with step sizes

Figure 67: Clipper wizard

Mute Wizard

Controls the mute blocks.

1) If ticked, a 0 will mute the output. If unticked a 1 will mute the output.

2) If ticked, by default the block will be muted

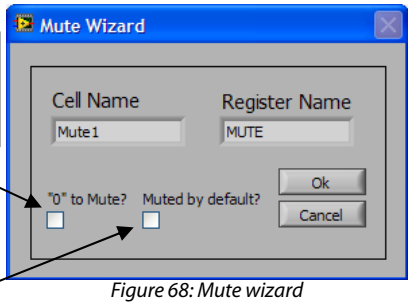


Figure 68: Mute wizard

Enable Wizard

Controls the enable block.

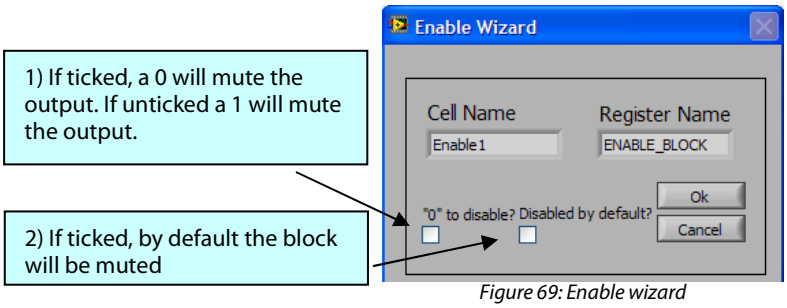


Figure 69: Enable wizard

Imported Registers

Some of the cell types create the registers importing a code file (**.cod*). The GUI will request the **.cod* file, and it will check that the cell type in the project matches the cell type defined in the .cod file.
Please note that these registers will not be editable, and they will be exclusive to the cell added to them. This means, it's not possible to add a cell to a previously created register, if this one was an imported register.

We provide a set of *.COD* files for the following cell types:

- ADI Surround
- AVC
- BBE
- BBE VIVA
- BassEnhance
- Compressor

The *.cod* files can be found on the *AppLayer / import_code_files /* directory. Once the code file is selected, a register like the example in Figure 70 (AVC block) will be automatically added to the register map.

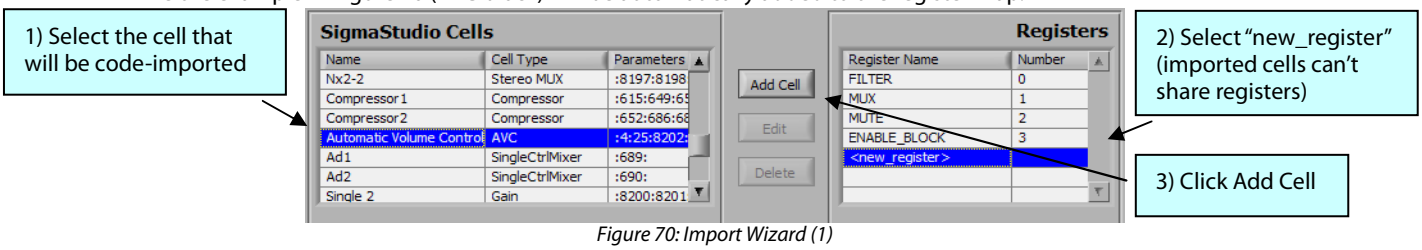


Figure 70: Import Wizard (1)

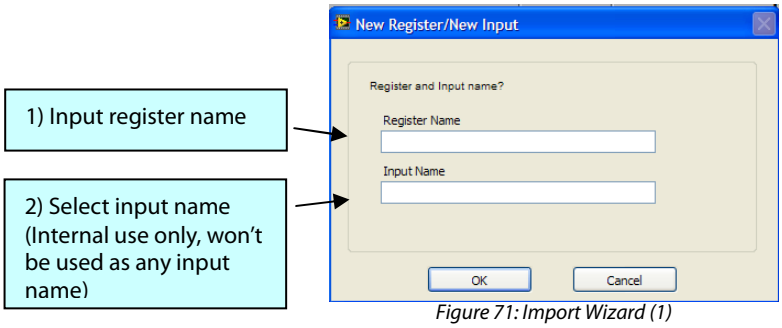


Figure 71: Import Wizard (1)

Open the *.cod file

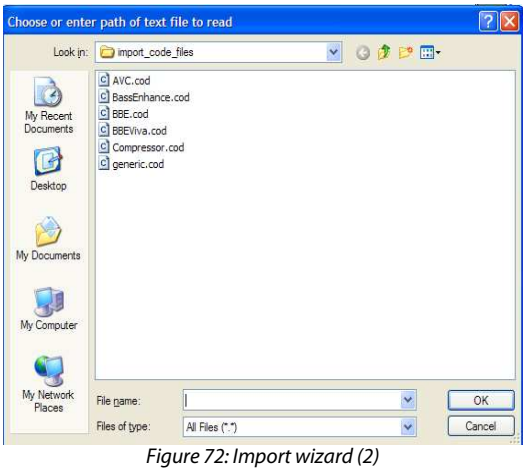


Figure 72: Import wizard (2)

0104h AVC
Default: 1358

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_GAIN				AVC_DECAY				MAX_ATTEN				AVC_LEVEL			

Figure 73: Import wizard example : AVC register created

To add or update these ***.cod*** code files, please copy them to this path, or install the latest release of the Application Layer GUI.

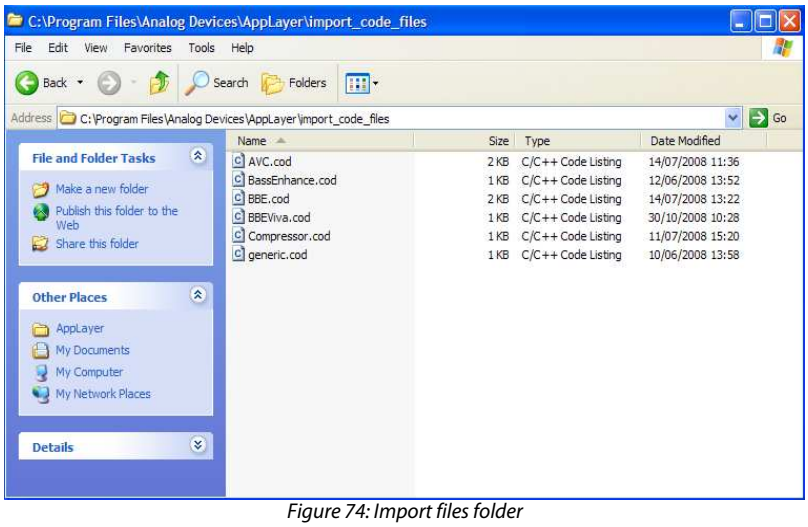


Figure 74: Import files folder

STEP-BY-STEP SAMPLE PROJECT CREATION

Here are the steps to follow in order to create a sample Application Layer project from scratch.

Application Layer Project Creation from SigmaStudio Project.

The user needs to create a project folder for the Application Layer, and then link it to a previously created SigmaStudio project.

1. Create a new project
2. Import a SigmaStudio project
3. Add cells to create new registers
    a. Depending on the cell, a different wizard will load
4. Edit / delete registers if it's needed
5. Compile and link the project
6. Generate HTML and C code data
7. Download the ROMs

APPENDIX A – DETAILED REGISTER DESCRIPTIONS

This section consists of a detailed description of the ADAV4601 registers. It is broken into 2 sections Default Audio Flow Registers and Main Control Registers. If a user has created their own custom audio flow the Application Layer Software will generate a HTML register map.

DEFAULT AUDIO FLOW REGISTERS

Address 0x0100 Mux Select 1 Register (Default: 0x0000)

Table 9.

Bit No.	Bit Name	Description	Default
Bits[15:12]	Main source mux	Source for main channel. 0x0 = reserved 0x1 = ADC1 0x2 = reserved 0x3 = SDIN0 0x4 = SDIN1 0x5 = SDIN2/SRC2 Channel B 0x6 = SDIN3/SRC2 Channel C 0x7 = SRC1 0x8 = SRC2 Channel A 0x9 = reserved 0xA = reserved	0000
Bits[11:8]	Reserved	Always write as 0 if writing to this register.	0000
Bits[7:4]	Headphone 1/AUXOUT4 output	Source for the Headphone 1/AUXOUT4 output. 0x0 = reserved 0x1 = ADC1 0x2 = reserved 0x3 = SDIN0 0x4 = SDIN1 0x5 = SDIN2/SRC2 Channel B 0x6 = SDIN3/SRC2 Channel C 0x7 = SRC1 0x8 = SRC2 Channel A 0x9 = reserved 0xA = reserved 0xB = reserved 0xC = delayed main input 0xD = main input after loudness 0xE = subchannel	0000
Bits[3:0]	AUXOUT3 output mux	Source for the AUXOUT3 output. 0x0 = reserved 0x1 = ADC1 0x2 = reserved 0x3 = SDIN0 0x4 = SDIN1 0x5 = SDIN2/SRC2 Channel B 0x6 = SDIN3/SRC2 Channel C 0x7 = SRC1	0000

		0x8 = SRC2 Channel A 0x9 = reserved 0xA = reserved 0xB = reserved 0xC = delayed main input 0xD = main input after loudness 0xE = subchannel	

Address 0x0101 Mux Select 2 Register (Default: 0x0000)

Table 10.

Bit No.	Bit Name	Description	Default
Bits[15:12]	SDO0 output	Source for the SDO0 output channel. 0x0 = reserved 0x1 = ADC1 0x2 = reserved 0x3 = SDIN0 0x4 = SDIN1 0x5 = SDIN2/SRC2 Channel B 0x6 = SDIN3/SRC2 Channel C 0x7 = SRC1 0x8 = SRC2 Channel A 0x9 = reserved 0xA = reserved 0xB = reserved 0xC = delayed main input 0xD = main input after loudness 0xE = subchannel	0000
Bits[11:8]	SPDIF output	Source for the SPDIF output. 0x0 = reserved 0x1 = ADC1 0x2 = reserved 0x3 = SDIN0 0x4 = SDIN1 0x5 = SDIN2/SRC2 Channel B 0x6 = SDIN3/SRC2 Channel C 0x7 = SRC1 0x8 = SRC2 Channel A 0x9 = reserved 0xA = reserved 0xB = reserved 0xC = delayed main input 0xD = main input after loudness 0xE = subchannel	0000
Bits[7:4]	AUXOUT1 output	Source for the AUXOUT1 output. 0x0 = reserved 0x1 = ADC1 0x2 = reserved 0x3 = SDIN0 0x4 = SDIN1 0x5 = SDIN2/SRC2 Channel B	0000

		0x6 = SDIN3/SRC2 Channel C 0x7 = SRC1 0x8 = SRC2 Channel A 0x9 = reserved 0xA = reserved 0xB = reserved 0xC = delayed main input 0xD = main input after loudness 0xE = subchannel	
Bits[3:0]	Reserved	Always write as 0 if writing to this register.	0000

Address 0x0102 Main and Headphone 1/AUXOUT4 Input Trim Register (Default: 0x0E0E)

Table 11.

Bit No.	Bit Name	Description	Default
Bits[15:14]	Reserved	Always write as 0 if writing to this register.	00
Bits[13:8]	Main input trim	These register bits are used to gain or attenuate the input to the main channel processing path from −14 dB to +14 dB in +1 dB steps. 0x00 = +14 dB 0x01 = +13 dB … 0x07 = +7 dB … 0x0E = 0 dB … 0x15 = −7 dB 0x1B = −13 dB 0x1C = −14 dB	001110
Bits[7:6]	Reserved	Always write as 0 if writing to this register.	00
Bits[5:0]	Headphone 1/AUXOUT4 input trim	These register bits are used to gain or attenuate the input to the Headphone 1/ AUXOUT4 channel processing path from −14 dB to +14 dB in +1 dB steps. 0x00 = +14 dB 0x01 = +13 dB … 0x07 = +7 dB … 0x0E = 0 dB … 0x15 = −7 dB … 0x1B = −13 dB 0x1C = −14 dB	001110

Address 0x0103 SDO0/AUXOUT3 and SPDIF Input Trim Register (Default: 0x0E0E)

Table 12.

Bit No.	Bit Name	Description	Default
Bits[15:14]	Reserved	Always write as 0 if writing to this register.	00
Bits[13:8]	SDO0/AUXOUT3 input trim	These register bits are used to gain or attenuate the input to the SDO0 and the AUXOUT3 processing path from −14 dB to +14 dB in +1 dB steps. 0x00 = +14 dB 0x01 = +13 dB … 0x07 = +7 dB … 0x0E = 0 dB … 0x15 = −7 dB 0x1B = −13 dB 0x1C = −14 dB	001110
Bits[7:6]	Reserved	Always write as 0 if writing to this register.	00
Bits[5:0]	SPDIF input trim	These register bits are used to gain or attenuate the input to the SPDIF processing path from −14 dB to +14 dB in +1 dB steps. 0x00 = +14 dB 0x01 = +13 dB … 0x07 = +7 dB … 0x0E = 0 dB … 0x15 = −7 dB … 0x1B = −13 dB 0x1C = −14 dB	001110

Address 0x0104 AUXOUT1 Input Trim Register (Default: 0x0E0E)

Table 13.

Bit No.	Bit Name	Description	Default
Bits[15:14]	Reserved	Always write as 0 if writing to this register.	00
Bits[13:8]	AUXOUT1 input trim	These register bits are used to gain or attenuate the input to the AUXOUT1 channel processing path from −14 dB to +14 dB in +1 dB steps. 0x00 = +14 dB 0x01 = +13 dB … 0x07 = +7 dB … 0x0E = 0 dB … 0x15 = −7 dB 0x1B = −13 dB 0x1C = −14 dB	001110

Bit No.	Bit Name	Description	Default
Bits[7:0]	Reserved	Always write as 0 if writing to this register.	00000000

Address 0x0105 Main Delay Register (Default: 0x0000)

Table 14.

Bit No.	Bit Name	Description	Default
Bits[15:13]	Reserved	Always write as 0 if writing to this register.	000
Bits[12:0]	Main delay	These register bits are used to specify the lip synchronization delay for the main channel. 0x0000 = 20.83 µs (1 sample delay at 48 kHz) 0x0001 = 20.83 µs (1 sample delay) 0x0002 = 41.66 µs (2 sample delay) … 0x1C20 = 150 ms (7200 samples delay)	0000000000000

Address 0x0106 Automatic Volume Control (Default: 0x350C)

Table 15.

Bit No.	Bit Name	Description	Default
Bits[15:12]	AVC maximum gain	Used to control the maximum gain in the range of 0 dB to 15 dB. This is the maximum gain that can be applied to the input signal in order to reach the desired output level. 0x0 = 15 dB 0x1 = 14 dB 0x2 = 13 dB 0x3 = 12 dB 0x4 = 11 dB 0x5 = 10 dB ... 0xF = 0 dB	0011
Bits[11:8]	AVC decay time	Used to control the decay time in the range of 20 ms to 12 sec. The decay time corresponds to the time required for the output to reach the desired level. 0x0 = 20 ms 0x1 = 100 ms 0x2 = 200 ms 0x3 = 500 ms 0x4 = 1 sec 0x5 = 2 sec 0x6 = 3 sec 0x7 = 4 sec 0x8 = 5 sec 0x9 = 6 sec 0xA = 7 sec 0xB = 8 sec 0xC = 9 sec 0xD = 10 sec	0101

Bit No.	Bit Name	Description	Default
		0xE = 11 sec 0xF = 12 sec	
Bits[7:4]	AVC maximum attenuation	Used to control the maximum attenuation in the range of −18 dB to −3 dB in +1 dB steps. This is the maximum attenuation that can be applied to the input signal in order to reach the desired output level. 0x0 = −18 dB 0x1 = −17 dB … 0xE = −4 dB 0xF = −3 dB	0000
Bits[3:0]	AVC output level	Used to control the required output level of the AVC block in the range of −3 dBFS to −18 dBFS in −1 dB steps. If the input signal is greater than the output level that has been set by these bits, the gain is automatically reduced. 0x0 = −18 dBFS 0x1 = −17 dBFS … 0xC = −6 dBFS 0xD = −5 dBFS 0xE = −4 dBFS 0xF = −3 dBFS	1100

Address 0x0107 Main Sever Band EQ Control Register (Default: 0x0018)

Table 16.

Bit No.	Bit Name	Description	Default
Bits[15:11]	Reserved	Always write as 0 if writing to this register.	00000
Bits[10:8]	Main EQ band	These register bits control the frequency band of the equalizer. 0x0 = 120 Hz 0x1 = 200 Hz 0x2 = 500 Hz 0x3 = 1200 Hz 0x4 = 3000 Hz 0x5 = 7500 Hz 0x6 = 12000 Hz	000
Bits[7:6]	Reserved	Always write as 0 if writing to this register.	00
Bits[5:0]	Main EQ gain	These register bits are used to control the required gain of the equalizer. The gain of the equalizer is changed in 0.5 dB steps. 0x00 = +12 dB 0x01 = +11.5 dB … 0x0A = +7 dB … 0x18 = 0 dB …	011000

Bit No.	Bit Name	Description	Default
		0x26 = −7 dB … 0x2F = −11.5 dB 0x30 = −12 dB	

Address 0x0108 Main Channel Loudness Register (Default: 0x0000)

Table 17.

Bit No.	Bit Name	Description	Default
Bits[15:7]	Reserved	Always write as 0 if writing to this register.	000000000
Bits[6:5]	Cutoff frequency	These register bits are used to control the cutoff frequency of the loudness. 00b – 10 Hz 01b – 30 Hz 10b – 50 Hz 11b – 70 Hz	00
Bits[4:0]	Loudness level	These register bits are used to control the required level of loudness. 0x00 = 0 dB 0x01 = +1 dB … 0x0E = +14 dB 0x0F = +15 dB	00000

Address 0x0109 Crossover Register (Default: 0x0505)

Table 18.

Bit No.	Bit Name	Description	Default
Bits[15:14]	Reserved	Always write as 0 if writing to this register.	00
Bits[13:8]	Low crossover frequency	The low crossover frequency is the cutoff frequency of the crossover low-pass filter. This means that only the frequencies under this frequency are sent to the woofer output. The frequency changes in 10 Hz steps. 0x00 = 50 Hz 0x01 = 60 Hz 0x02 = 70 Hz 0x03 = 80 Hz 0x04 = 90 Hz 0x05 = 100 Hz 0x06 = 110 Hz … 0x2D = 500 Hz 0x2E = 510 Hz	000101
Bits[7:6]	Reserved	Always write as 0 if writing to this register.	00
Bits[5:0]	High crossover frequency	The high crossover frequency is the cutoff frequency of the crossover high-pass filter. This means that only the frequencies above this frequency are sent to the tweeter output. 0x00 = 50 Hz 0x01 = 60 Hz 0x02 = 70 Hz	000101

Bit No.	Bit Name	Description	Default
		0x03 = 80 Hz 0x04 = 90 Hz 0x05 = 100 Hz 0x06 = 110 Hz ... 0x3B = 640 Hz 0x3C = 650 Hz	

Address 0x010A Crossover Trim Register (Default: 0x0E0E)

Table 19.

Bit No.	Bit Name	Description	Default
Bits[15:14]	Reserved	Always write as 0 if writing to this register.	00
Bits[13:8]	Tweeter trim	These register bits are used to gain or attenuate the input to the tweeter outputs from −14 dB to +14 dB in +1 dB steps. 0x00 = +14 dB 0x01 = +13 dB … 0x07 = +7 dB … 0x0E = 0 dB … 0x15 = −7 dB … 0x1B = −13 dB 0x1C = −14 dB	001110
Bits[7:6]	Reserved	Always write as 0 if writing to this register.	00
Bits[5:0]	Woofer trim	These register bits are used to gain or attenuate the input to the woofer outputs from −14 dB to +14 dB in +1 dB steps. 0x00 = +14 dB 0x01 = +13 dB … 0x07 = +7 dB … 0x0E = 0 dB … 0x15 = −7 dB … 0x1B = −13 dB 0x1C = −14 dB	001110

Address 0x010B ADI Bass Control Register (Default: 0x0062)

Table 20.

Bit No.	Bit Name	Description	Default
Bits[15:9]	Reserved	Always write as 0 if writing to this register.	0000000
Bits[8:4]	Boost value	The boost ranges from 0 dB to 31 dB and it controls the maximum dynamic gain applied to the algorithm. 0x00 = 0 dB 0x01 = 1 dB 0x02 = 2 dB 0x03 = 3 dB 0x04 = 4 dB 0x05 = 5 dB 0x06 = 6 dB 0x07 = 7 dB … 0x1E = 30 dB 0x1F = 31 dB	00110
Bits[3:0]	Boost frequency	The boost frequency ranges from 20 Hz to 320 Hz, and it designates the center frequency for the boosting filter. The frequency is increased in 20 Hz steps. 0x0 = 20 Hz 0x1 = 40 Hz 0x2 = 60 Hz 0x3 = 80 Hz … 0xE = 300 Hz 0xF = 320 Hz	0010

Address 0x010C and Address 0x010D Tweeter Left Balance Control Registers (Default: 0x0080, 0x0000)

These two registers (0x010C and 0x010D) control the balance for the left tweeter output.

The balance control words are 28-bit words in twos complement and a 5.23 format. This means there are five bits to the left of the decimal point of which the most significant bit is the sign bit.

To simplify updating the tweeter left balance control, the I^2C address pointer auto-increments when writing and reading. This means that the balance control register can be updated in a single I^2C block write. Therefore, it is recommended that the tweeter left balance control is updated using the following I^2C write format:

<dev addr><010C><32-bit data transfer>

Note that the tweeter left balance control is a 32-bit parameter; therefore, both Register 0x010C and Register 0x010D must be written to. Writing anything less than the 32 bits to these registers will not update the parameter.

Table 21.

Bit No.	Bit Name	Description	Default
Bits[15:12]	Reserved	Always write as 0 if writing to this register.	0000
Bits[11:0]	Tweeter left balance control register[27:0]	0x010C Bits[11:0] = tweeter left balance control register[27:16]	000010000000
Bits [15:0]	Tweeter left balance control register[27:0]	0x010D Bits[15:0] = tweeter left balance control register[15:0]	0000000000000000

Address 0x010E and Address 0x010F Tweeter Right Balance Control Registers (Default: 0x0080, 0x0000)

These two registers (0x010E and 0x010F) control the balance for the right tweeter output.
The balance control words are 28-bit words in twos complement and a 5.23 format. This means there are five bits to the left of the decimal point of which the most significant bit is the sign bit.

Table 22.

Bit No.	Bit Name	Description	Default
Bits[15:12]	Reserved	Always write as 0 if writing to this register.	0000
Bits[11:0]	Tweeter right balance control register[27:0]	0x010E Bits[11:0] = tweeter right balance control register[27:16]	000010000000
Bits [15:0]	Tweeter right balance control register[27:0]	0x010F Bits [15:0] = tweeter right balance control register[15:0]	0000000000000000

Address 0x0110 and Address 0x0111 Woofer Left Balance Control Registers (Default: 0x0080, 0x0000)

These two registers control the balance for the left woofer output.
The balance control words are 28-bit words in twos complement and a 5.23 format. This means there are five bits to the left of the decimal point of which the most significant bit is the sign bit.

Table 23.

Bit No.	Bit Name	Description	Default
Bits[15:12]	Reserved	Always write as 0 if writing to this register.	0000
Bits[11:0]	Woofer left balance control register[27:0]	0x0110 Bits[11:0] = woofer left balance control register[27:16]	000010000000
Bits [15:0]	Woofer left balance control register[27:0]	0x0111 Bits[15:0] = woofer left balance control register[15:0]	0000000000000000

Address 0x0112 and Address 0x0113 Woofer Right Balance Control Registers (Default: 0x0080, 0x0000)

These two registers control the balance for the right tweeter output.
The balance control words are 28-bit words in twos complement and a 5.23 format. This means there are five bits to the left of the decimal point of which the most significant bit is the sign bit.

Table 24.

Bit No.	Bit Name	Description	Default
Bits[15:12]	Reserved	Always write as 0 if writing to this register.	0000
Bits[11:0]	Woofer right balance control register[27:0]	0x0112 Bits[11:0] = woofer right balance control register[27:16]	000010000000
Bits [15:0]	Woofer right balance control register[27:0]	0x0113 Bits [15:0] = woofer right balance control register[15:0]	0000000000000000

Address 0x0114 and Address 0x0115 Main Volume Control Registers (Default: 0x0080, 0x0000)

These two registers control the volume for the main channel output.
The volume control words are 28-bit words in twos complement and a 5.23 format. This means there are five bits to the left of the decimal point of which the most significant bit is the sign bit.

Table 25.

Bit No.	Bit Name	Description	Default
Bits[15:12]	Reserved	Always write as 0 if writing to this register.	0000
Bits[11:0]	Woofer right balance control register[27:0]	0x0114 Bits[11:0] = main volume Bits[27:16]	000010000000
Bits [15:0]	Woofer right balance control register[27:0]	0x0115 Bits[15:0] = main volume Bits[15:0]	0000000000000000

Address 0x0116 Tweeter Peak Limiter Control Register (Default: 0x0F00)

This register controls the peak limiter for the main output tweeter.

Table 26.

Bit No.	Bit Name	Description	Default
Bits[15:13]	Reserved	Always write as 0 if writing to this register.	000
Bits[12:8]	Post gain	These register bits control the post gain in the range of +15 dB to −15 dB in +1 dB steps. 0x00 = +15 dB 0x01 = +14 dB … 0x08 = +7 dB … 0x0F = +0 dB … 0x16 = −7 dB … 0x1D = −14 dB 0x1E = −15 dB	01111
Bits[7:5]	Hold time	These register bits control the hold time for the limiter, which is the time in ms that the limiter holds the attenuated level after the current input to the limiter function falls below the limiter threshold. 0x0 = 0 ms 0x1 = 10 ms 0x2 = 20 ms 0x3 = 30 ms 0x4 = 40 ms 0x5 = 50 ms 0x6 = 60 ms 0x7 = 70 ms	000

Bit No.	Bit Name	Description	Default
Bits[4:0]	Decay time	These register bits control the decay time for the limiter in the range of 10 dB/s to 320 dB/s in 10 dB/s steps. 0x00 = 10 dB/s (~870 ms) 0x01 = 20 dB/s (~435 ms) 0x02 = 30 dB/s (~289 ms) 0x03 = 40 dB/s (~217 ms) 0x04 = 50 dB/s (~173 ms) 0x05 = 60 dB/s (~144 ms) 0x06 = 70 dB/s (~124 ms) 0x07 = 80 dB/s (~108 ms) 0x08 = 90 dB/s (~96 ms) 0x09 = 100 dB/s (~86 ms) 0x0A = 110 dB/s (~78 ms) 0x0B = 120 dB/s (~72 ms) 0x0C = 130 dB/s (~66 ms) 0x0D = 140 dB/s (~62 ms) 0x0E = 150 dB/s (~57 ms) 0x0F = 160 dB/s (~54 ms) 0x10 = 170 dB/s (~51 ms) 0x11 = 180 dB/s (~48 ms) 0x12 = 190 dB/s (~45 ms) 0x13 = 200 dB/s (~43 ms) 0x14 = 210 dB/s (~41 ms) 0x15 = 220 dB/s (~39 ms) 0x16 = 230 dB/s (~37 ms) 0x17 = 240 dB/s (~36 ms) 0x18 = 250 dB/s (~34 ms) 0x19 = 260 dB/s (~33 ms) 0x1A = 270 dB/s (~32 ms) 0x1B = 280 dB/s (~31 ms) 0x1C = 290 dB/s (~29 ms) 0x1D = 300 dB/s (~28 ms) 0x1E = 310 dB/s (~28 ms) 0x1F = 320 dB/s (~27 ms)	0000

Address 0x0117 Woofer Peak Limiter Control Register (Default: 0x0F00)

Table 27.

Bit No.	Bit Name	Description	Default
Bits[15:13]	Reserved	Always write as 0 if writing to this register.	000
Bits[12:8]	Post gain	These register bits control the post gain in the range of +15 dB to −15 dB in +1 dB steps. 0x00 = +15 dB 0x01 = +14 dB … 0x08 = +7 dB …	01111

Bit No.	Bit Name	Description	Default
		0x0F = 0 dB … 0x16 = −7 dB … 0x1D = −14 dB 0x1E = −15 dB	
Bits[7:5]	Hold time	These register bits control the hold time for the limiter, which is the time in ms that the limiter holds the attenuated level after the current input to the limiter function falls below the limiter threshold. 0x0 = 0 ms 0x1 = 10 ms 0x2 = 20 ms 0x3 = 30 ms 0x4 = 40 ms 0x5 = 50 ms 0x6 = 60 ms 0x7 = 70 ms	000
Bits[4:0]	Decay time	These register bits control the decay time for the limiter in the range of 10 dB/s to 320 dB/s in 10 dB/s steps. 0x00 = 10 dB/s (~870 ms) 0x01 = 20 dB/s (~435 ms) 0x02 = 30 dB/s (~289 ms) 0x03 = 40 dB/s (~217 ms) 0x04 = 50 dB/s (~173 ms) 0x05 = 60 dB/s (~144 ms) 0x06 = 70 dB/s (~124 ms) 0x07 = 80 dB/s (~108 ms) 0x08 = 90 dB/s (~96 ms) 0x09 = 100 dB/s (~86 ms) 0x0A = 110 dB/s (~78 ms) 0x0B = 120 dB/s (~72 ms) 0x0C = 130 dB/s (~66 ms) 0x0D = 140 dB/s (~62 ms) 0x0E = 150 dB/s (~57 ms) 0x0F = 160 dB/s (~54 ms) 0x10 = 170 dB/s (~51 ms) 0x11 = 180 dB/s (~48 ms) 0x12 = 190 dB/s (~45 ms) 0x13 = 200 dB/s (~43 ms) 0x14 = 210 dB/s (~41 ms) 0x15 = 220 dB/s (~39 ms) 0x16 = 230 dB/s (~37 ms) 0x17 = 240 dB/s (~36 ms) 0x18 = 250 dB/s (~34 ms) 0x19 = 260 dB/s (~33 ms) 0x1A = 270 dB/s (~32 ms) 0x1B = 280 dB/s (~31 ms)	00000

Bit No.	Bit Name	Description	Default
		0x1C = 290 dB/s (~29 ms) 0x1D = 300 dB/s (~28 ms) 0x1E = 310 dB/s (~28 ms) 0x1F = 320 dB/s (~27 ms)	

Address 0x0118 Headphone 1/AUXOUT4 Seven Band EQ Control Register (Default: 0x0018)

Table 28.

Bit No.	Bit Name	Description	Default
Bits[15:11]	Reserved	Always write as 0 if writing to this register.	00000
Bits[10:8]	Headphone 1/ AUXOUT4 EQ band	These register bits control the frequency band of the equalizer. 0x0 = 120 Hz 0x1 = 200 Hz 0x2 = 500 Hz 0x3 = 1200 Hz 0x4 = 3000 Hz 0x5 = 7500 Hz 0x6 = 12000 Hz	000
Bits[7:6]	Reserved	Always write as 0 if writing to this register.	00
Bits[5:0]	Headphone 1/ AUXOUT4 EQ gain	These register bits are used to control the required gain of the equalizer. The gain of the equalizer is changed in 0.5 dB steps. 0x00 = +12 dB 0x01 = +11.5 dB … 0x0A = +7 dB … 0x18 = 0 dB … 0x26 = −7 dB … 0x2F = −11.5 dB 0x30 = −12 dB	001110

Address 0x011A and Address 0x011B Headphone 1/AUXOUT4 Left Balance Control Registers (Default: 0x0080, 0x0000)

These two registers control the balance for the Left Headphone 1 and AUXOUT 4 output.
The balance control words are 28-bit words in twos complement and a 5.23 format. This means there are five bits to the left of the decimal point of which the most significant bit is the sign bit.

Table 29.

Bit No.	Bit Name	Description	Default
Bits[15:12]	Reserved	Always write as 0 if writing to this register.	0000
Bits[11:0]	Woofer left balance control register[27:0]	0x011A Bits[11:0] = Headphone 1/AUXOUT 4 left balance control register[27:16]	000010000000

Bit No.	Bit Name	Description	Default
Bits [15:0]	Woofer left balance control register[27:0]	0x011B Bits [15:0] = Headphone 1/AUXOUT 4 left balance control register[15:0]	0000000000000000

Address 0x011C and Address 0x011D Headphone 1/AUXOUT4 Right Balance Control Registers (Default: 0x0080, 0x0000)

These two registers control the balance for the Right Headphone 1 and AUXOUT 4 output.
The balance control words are 28-bit words in twos complement and a 5.23 format. This means there are five bits to the left of the decimal point of which the most significant bit is the sign bit.

Table 30.

Bit No.	Bit Name	Description	Default
Bits[15:12]	Reserved	Always write as 0 if writing to this register.	0000
Bits[11:0]	Woofer left balance control register[27:0]	0x011C Bits[11:0] = Headphone 1/AUXOUT 4 right balance control register[27:16]	000010000000
Bits [15:0]	Woofer left balance control register[27:0]	0x011D Bits [15:0] = Headphone 1/AUXOUT 4 right balance control register[15:0]	0000000000000000

Address 0x011E and Address 0x011F Headphone 1/AUXOUT4 Volume Control Registers (Default: 0x0080, 0x0000)

These two registers control the volume for the headphone and AUXOUT4 outputs.
The volume control words are 28-bit words in twos complement and a 5.23 format. This means there are five bits to the left of the decimal point of which the most significant bit is the sign bit.

Table 31.

Bit No.	Bit Name	Description	Default
Bits[15:12]	Reserved	Always write as 0 if writing to this register.	0000
Bits[11:0]	Headphone 1 volume control[27:0]	0x011E Bits[11:0] = Headphone 1/AUXOUT 4 volume Bits[27:16]	000010000000
Bits [15:0]	Headphone 1 volume control[27:0]	0x011F Bits [15:0] = Headphone 1/AUXOUT 4 volume Bits[15:0]	0000000000000000

Address 0x0120 Headphone 1/AUXOUT4 Channel Loudness Register (Default: 0x0000)

Table 32.

Bit No.	Bit Name	Description	Default
Bits[15:7]	Reserved	Always write as 0 if writing to this register.	000000000
Bits[6:5]	Cutoff frequency	These register bits are used to control the cutoff frequency of the loudness. 00b = 10 Hz 01b = 30 Hz 10b = 50 Hz 11b = 70 Hz	00
Bits[4:0]	Loudness level	These register bits are used to control the required level of loudness. It can be changed in 1 dB steps.	00000

Bit No.	Bit Name	Description	Default
		0x00 = 0 dB 0x01 = 1 dB … 0x0E = 14 dB 0x0F = 15 dB	

Address 0x0121 Mute Control Register (Default: 0x0000)

Table 33.

Bit No.	Bit Name	Description	Default
Bits[15:8]	Reserved	Always write as 0 if writing to this register.	00000000
Bit[7]	Mute tweeter output	Mutes the tweeter output. 0b = mutes 1b = unmutes	0
Bit[6]	Mute woofer output	Mutes the woofer output. 0b = mutes 1b = unmutes	0
Bit[5]	Mute AUXOUT3 output	Mutes the AUXOUT3 output. 0b = mutes 1b = unmutes	0
Bit[4]	Mute HP1/AUXOUT4 output	Mutes the Headphone 1 and AUXOUT4 output. 0b = mutes 1b = unmutes	0
Bit[3]	Mute SDO0 output	Mutes the SDO0 output. 0b = mutes 1b = unmutes	0
Bit[2]	Mute SPDIF output	Mutes the SPDIF output. 0b = mutes 1b = unmutes	0
Bit[1]	Mute AUXOUT1 output	Mutes the AUXOUT1 output. 0b = mutes 1b = unmutes	0
Bit[0]	Reserved	Always write as 0 if writing to this register.	0

Address 0x0122 Audio Flow Control Register (Default: 0x8001)

Table 34.

Bit No.	Bit Name	Description	Default
Bit[15]	Reserved	Always write a 1 if writing to this register.	1
Bit[14]	Enable AVC	When set to 1, it enables the AVC function. 0b = disabled 1b = enabled	0
Bit[13]	Enable main delay	When set to 1, it enables the lip synchronization delay. 0b = disabled 1b = enabled	0
Bit[12]	Enable main EQ	When set to 1, it enables the seven band equalizer. 0b = disabled	0

Bit No.	Bit Name	Description	Default
		1b = enabled	
Bit[11]	Enable ADI bass	When set to 1, it enables the ADI bass. 0b = disabled 1b = enabled	0
Bit[10]	Enable main loudness	When set to 1, it enables the loudness. 0b = disabled 1b = enabled	0
Bit[9]	Enable main beeper	When set to 1, it leaves only the beeper on the main channel. By default, this register bit is set to 0, which means the main channel input is added to the beeper. 0b – beeper and channel 1b – beeper only	0
Bit[8]	Enable main limiter	When set to 1, it enables the tweeter and woofer peak limiters. 0b = disabled 1b = enabled	0
Bit[7]	Enable speaker EQ	When set to 1, it enables the eight band speaker equalizer for the tweeter output. 0b = disabled 1b = enabled	0
Bit[6]	Enable crossover bypass	When set to 1, it enables the crossover low-pass and high-pass filters for the main channel. 0b = disabled 1b = enabled	0
Bit[5]	Tweeter output control	When set to 1, the tweeter and woofer outputs are added together and output on the tweeter output. 0b = tweeter only 1b = tweeter and woofer	0
Bit[4]	Enable subchannel LPF	Used to control the LPF on the subchannel. 0b = enabled 1b = disabled	0
Bit[3]	Enable HP1 EQ	When set to 1, it enables the seven band equalizer for the Headphone 1 channel. 0b = disabled 1b = enabled	0
Bit[2]	Enable HP1 loudness	When set to 1, it enables the loudness for the Headphone 1 channel. 0b = disabled 1b = enabled	0
Bit[1]	Enable HP1 beeper	When set to 1, it leaves only the beeper on the Headphone 1 channel. By default, this bit is 0, which means the Headphone 1 channel input is added to the beeper. 0b = beeper and channel 1b = beeper only	0
Bit[0]	Enable spatializer	When set to 1, it enables the ADI spatializer. 0b = disabled 1b = enabled	1

Address 0x0123 Main Beeper Control Register (Default: 0x0005)

This register controls the main beeper block.

Table 35.

Bit No.	Bit Name	Description	Default
Bits[15:12]	Reserved	Always write as 0 if writing to this register.	0000
Bits[11:8]	Volume gain	These register bits control the volume of the beeper in the range of −38 dB to −10 dB in +2 dB steps. 0x0 = Off 0x1 = −38 dB 0x2 = −36 dB ... 0xE = −12 dB 0xF = −10 dB	0000
Bits[7:6]	Reserved	Always write as 0 if writing to this register.	00
Bits[5:0]	Frequency	These register bits control the frequency of the beeper in the range of 0 Hz (beeper off) to 11812.5 Hz in 187.5 Hz steps. 0x00 = 0 Hz 0x01 = 187.5 Hz 0x02 = 375 Hz 0x03 = 562.5 Hz 0x04 = 750 Hz 0x05 = 937.5 Hz 0x06 = 1125 Hz … 0x2E = 8625 Hz 0x2F = 8812.5 Hz	000101

Address 0x0124 Low-Pass Filter (Subchannel) Register (Default: 0x0003)

This register is used to control the low pass filter cutoff frequency for the subwoofer channel.

Table 36.

Bit No.	Bit Name	Description	Default
Bits[15:4]	Reserved	Always write as 0 if writing to this register.	000000000000
Bits[3:0]	LPF sub cutoff	These register bits control the cutoff frequency of the low-pass filter of the subwoofer channel. The range of values is 60 Hz to 340 Hz in 20 Hz steps. 0x0 = Off 0x1 = 60 Hz 0x2 = 80 Hz 0x3 = 100 Hz 0x4 = 120 Hz ... 0xE = 320 Hz 0xF = 340 Hz	0011

Address 0x0126 SRC Delay Register (Default: 0x0000)

This register is used to set the delay for the SRC channel.

Table 37.

Bit No.	Bit Name	Description	Default
Bits[15:12]	Reserved	Always write as 0 if writing to this register.	0000
Bits[11:0]	SRC delay	The range of values is 20.83 µs to 42 ms in 20.83 µs (1 sample delay) steps. 0x000 = 20.83 µs (1 sample delay) 0x001 = 41.66 µs (2 sample delay) ... 0x7E1 = 42 ms (2017 sample delay)	000000000000

Address 0x0127 SRC Control Register (Default: 0x0030)

This register is used to enable the DSP mute circuit for SRC1 and SRC2. If SRC is enabled and detects an error, it will mute the output of the SRC. It also bypasses de-emphasis filters of the SRC.

Table 38.

Bit No.	Bit Name	Description	Default
Bits[15:6]	Reserved	Always write as 0 if writing to this register.	0000000000
Bit[5]	SRC1 de-emphasis filter bypass	This bypasses the SRC1 de-emphasis filter. 0b = bypass 1b = enabled	0
Bit[4]	SRC2 de-emphasis filter bypass	This bypasses the SRC2 de-emphasis filter. 0b = bypass 1b = enabled	0
Bit[3]	SRC1 DSP mute circuit bypass	If an error is detected, it mutes the output of SRC1. 0b = enabled 1b = disabled	0
Bit[2]	SRC2 Channel A DSP mute circuit bypass	If an error is detected, it mutes the output of SRC2. 0b = enabled 1b = disabled	0
Bit[1]	SRC2 Channel B DSP mute circuit bypass	If an error is detected, it mutes the output of SRC2. 0b = enabled 1b = disabled	0
Bit[0]	SRC2 Channel C DSP mute circuit bypass	If an error is detected, it mutes the output of SRC2. 0b = enabled 1b = disabled	0

MAIN CONTROL REGISTERS

Address 0x0000 Initialization Control Register (Default: 0x0080)

Table 39.

Bit No.	Bit Name	Description	Default
Bits[15:13]	Reserved	Always write as 0 if writing to this register.	000
Bit[12]	Slew mute	When set to 1, all slew parameters ramp to zero. It is recommended to set this bit to 1 prior to downloading a new program to reduce any risk of pops or clicks on the output. 0b = unmuted 1b= muted	0
Bit[11]	Reserved	Always write as 0 if writing to this register.	0
Bits[10:9]	MCLKI frequency select	Used to select the MCLKI pin frequency. 00b = 512 × frequency sample (FS) (24.576 MHz) 01b = 256 × FS (12.288 MHz) 10b = 128 × FS (6.144 MHz) 11b = 64 × FS (3.072 MHz)	00
Bits[8:7]	SRC2 Channel A input select	Used to select the source for SRC2 channel A. 00b = SDIN0 01b = SDIN1 10b = SDIN2 11b = SDIN3	01
Bits[6:5]	SRC1 input select	Used to select the source for SRC1. 00b = SDIN0 01b = SDIN1 10b = SDIN2 11b = SDIN3	00
Bit[4]	SRC2 Channel A enable	Used to enable SRC2 Channel A. 0b – disabled 1b – enabled	0
Bit[3]	SRC1 enable	Used to enable SRC1 Channel A. 0b = disabled 1b = enabled	0
Bit[2]	GSB enable	When set to 1, the ADAV4601 enters standby mode. 0b = disable standby or not in standby 1b = enable standby or not in standby	0
Bit[1]	Audio processor enable	Set to 1 to enable the audio processor. 0b = disabled 1b = enabled	0
Bit[0]	GPU	Globally powers up all parts of the device. If read back, it indicates the status of the global power-up. 0b = use power management register 1b = global power-up	0

Address 0x0004 Serial Port Control 1 Register (Default: 0x0000)

Table 40.

Bit No.	Bit Name	Description	Default
Bits[15:13]	Reserved	Always write as 0 if writing to this register.	000
Bits[12:11]	SRC2 serial mode	Use to select the format of the data for SRC2. 00b = I²S 01b = left-justified 10b = right-justified 11b = not applicable	00
Bits[10:9]	SRC2 word width	Use to specify the word width of the data. 00b = 24 bits 01b = 20 bits 10b = 16 bits 11b = not applicable	00
Bits[8:7]	SRC1 serial mode	Use to select the format of the data for SRC1. 00b = I²S 01b = left-justified 10b = right-justified 11b = not applicable	00
Bits[6:5]	SRC1 word width	Use to specify the word width of the data. 00b = 24 bits 01b = 20 bits 10b = 16 bits 11b = not applicable	00
Bit[4]	Sync master slave	Used to set master or slave mode for the synchronous input. In slave mode, LRCLK1 and BCLK1 are provided by another source. In master mode, the ADAV46xx provides LRCLK1 and BCLK1. 0b = slave 1b = master	0
Bits[3:2]	Sync serial mode	Used to select the format of the data for the inputs used by the synchronous serial input block. 00b = I²S 01b = left-justified 10b = right-justified 11b = not applicable	00
Bits[1:0]	Sync word width	Used to specify the word width of the data for the synchronous digital inputs. 00b = 24 bits 01b = 20 bits 10b = 16 bits 11b = not applicable	00

Address 0x0005 Analog Power Management 1 Register (Default: 0x8000)

Table 41.

Bit No.	Bit Name	Description	Default
Bit[15]	DAC standby disable	Set to 1 after reset, which means all DACs are in normal mode but are still powered down. 0b = DACs in low power mode 1b = DACs in normal mode	1
Bit[14]	Reserved	Always write as 0 if writing to this register.	0
Bit[13]	REF BUF	Provides the voltage reference for the analog core. 1b = block powered up 0b = block powered down	0
Bit[12]	Reserved	Always write as 0 if writing to this register.	0
Bit[11]	Reserved	Always write as 0 if writing to this register.	0
Bit[10]	ADC1 right	Powers on the ADC1 right channel. 1b = block powered up 0b = block powered down	0
Bit[9]	ADC1 left	Powers on the ADC1 left channel. 1b = block powered up 0b = block powered down	0
Bit[8:7]	Reserved	Always write as 0 if writing to this register.	00
Bit[6]	AUXDAC3 right	Powers on the AUXDAC3 right channel. 1b = block powered up 0b = block powered down	0
Bit[5]	AUXDAC3 left	Powers on the AUXDAC3 left channel. 1b = block powered up 0b = block powered down	0
Bit[4]	Reserved	Always write as 0 if writing to this register.	0
Bit[3]	Reserved	Always write as 0 if writing to this register.	0
Bit[2]	Reserved	Always write as 0 if writing to this register.	0
Bit[1]	AUXDAC1 right	Powers on the AUXDAC1 right channel. 1b = block powered up 0b = block powered down	0
Bit[0]	AUXDAC1 left	Powers on the AUXDAC1 left channel. 1b = block powered up 0b = block powered down	0

Address 0x0006 Analog Power Management 2 Register (Default: 0x0000)

Table 42.

Bit No.	Bit Name	Description	Default
Bits[15:10]	Reserved	Always write as 0 if writing to this register.	000000
Bit[9]	PLL	Powers on the PLL. 1b = block powered up 0b = block powered down	0
Bits[8:6]	Reserved	Always write as 0 if writing to this register.	000
Bit[5]	Reserved	Always write as 0 if writing to this register.	0
Bit[4]	Reserved	Always write as 0 if writing to this register.	0
Bit[3]	HP1DAC right	Powers on the HP1 DAC right channel.	0

Bit No.	Bit Name	Description	Default
		1b = block powered up 0b = block powered down	
Bit[2]	HP1DAC left	Powers on the HP1 DAC left channel. 1b = block powered up 0b = block powered down	0
Bit[1]	HP1 AMP right	Powers on the HP1 AMP right channel. 1b = block powered up 0b = block powered down	0
Bit[0]	HP1 AMP left	Powers on the HP1 AMP left channel. 1b = block powered up 0b = block powered down	0

Address 0x0007 Digital Power Management Register (Default: 0x0000)

Table 43.

Bit No.	Bit Name	Description	Default
Bits[15:8]	Reserved	Always write as 0 if writing to this register.	00000000
Bit[7]	PWM	Powers on the PWM channels. 1b = block powered up 0b = block powered down	0
Bit[6]	S/PDIF TX	Powers on the S/PDIF transmitter. 1b = block powered up 0b = block powered down	0
Bit[5]	Reserved	Always write as 0 if writing to this register.	0
Bit[4]	SRC2	Powers on SRC2. 1b = block powered up 0b = block powered down	0
Bit[3]	SRC1	Powers on SRC1. 1b = block powered up 0b = block powered down	0
Bit[2]	Reserved	Always write as 0 if writing to this register.	0
Bit[1]	ADC/DAC engine	Powers on the ADC/DAC engine. 1b = block powered up 0b = block powered down	0
Bit[0]	Audio processor	Powers on the audio processor core. 1b = block powered up 0b = block powered down	0

Address 0x0009 SPDIF Transmitter Control Register (Default: 0x0000)

Table 44.

Bit No.	Bit Name	Description	Default
Bits[15:8]	Reserved	Always write as 0 if writing to this register.	00000000
Bits[14:12]	SPDIF output select	Selects the source for the SPDIF output. 000b = output internally generated SPDIF 001b = output SPDIF_IN2	000

Bit No.	Bit Name	Description	Default
		010b = output SPDIF_IN1 011b = output SPDIF_IN0 100b = output SPDIF_IN3 101b = output SPDIF_IN4 110b = output SPDIF_IN5 111b = output SPDIF_IN6	
Bit[11]	SPDIF disable	Enable or disables the SPDIF transmitter. 0b = enabled 1b = disabled	0
Bit[10]	PRE Edge	Sets the edge to be used for the preamble. 0b = rising edge 1b = falling edge	0
Bit[9]	Validity polarity	Used to indicate to the receiver if the data in the transmitted stream is valid audio data. 0b = valid data sent 1b = invalid data sent	0
Bit[8]	Copy flag	Used to indicate to the receiver if the data is copyright material. 0b = copyright 1b = not copyright	0
Bits[7:0]	Channel status	Used to specify the type of equipment in use. Not applicable when the SPDIF Mux Bits[14:12] are set to anything other than 000b.	00000000

Address 0x000A Misc Control Register (Default: 0x8000)

Table 45.

Bit No.	Bit Name	Description	Default
Bit[15]	PWM ready flag (read-only)	Indicates the current status of the PWM ready pin. When PWM ready is low, the PWM is not enabled. When PWM ready is high, the PWM is enabled and stable. 0b = PWM ready pin low 1b = PWM ready pin high	0
Bit[14]	Enable selected PWM channels	When enabled, all PWM channels selected by Bits[10:7] can be used. 0b = all PWM channels disabled 1b = selected PWM channels enabled	0
Bit[13]	MCLK_OUT CLK type select	Used to configure the MCLK_OUT pin. 0b = crystal frequency on MCLK_OUT 1b = internally generated clocks on MCLK_OUT	0
Bit[12]	PWM enable/disable patterns	Enables the enable/disable patterns for the PWM block. 0b = enable/disable pattern not used 1b = Use enable/disable pattern	0
Bit[11]	DAC mod offset	Adds dc offset to the DAC Σ-Δ modulator to eliminate idle tones. It is recommended that this bit is disabled before the ADC/DAC engine is powered up in Control Register 0x0007[1]. 0b = enabled 1b = disabled	1

Bit No.	Bit Name	Description	Default
Bit[10]	PWM Enable 4	The PWM outputs are disabled by default, which means that the outputs are at GND. When this bit is set to 1 and Bit[14] of this register is set to 1, then the PWM Enable 4 channel is enabled. 0b = disabled 1b = enabled	0
Bit[9]	PWM Enable 3	The PWM outputs are disabled by default, which means that the outputs are at GND. When this bit is set to 1 and Bit[14] of this register is set to 1, then the PWM Enable 3 channel is enabled. 0b = disabled 1b = enabled	0
Bit[8]	PWM Enable 2	The PWM outputs are disabled by default, which means that the outputs are at GND. When this bit is set to 1 and Bit[14] of this register is set to 1, then the PWM Enable 2 channel is enabled. 0b = disabled 1b = enabled	0
Bit[7]	PWM Enable 1	The PWM outputs are disabled by default, which means that the outputs are at GND. When this bit is set to 1 and Bit[14] of this register is set to 1, then the PWM Enable 1 channel is enabled. 0b = disabled 1b = enabled	0
Bit[6]	SRC2 lock (read-only)	Set to 1 when the sample rate converter (SRC) locks to the incoming data, indicating the data is valid. 0b = not locked 1b = locked	0
Bit[5]	SRC1 lock (read-only)	Set to 1 when the sample rate converter (SRC) locks to the incoming data, indicating the data is valid. 0b = not locked 1b = locked	0
Bit[4]	MCLK_OUT Enable	Enables the clock chosen by Bit[13] and Bits[3:1] to be output on the MCLK_OUT pin. 0b = MCLK_OUT function disabled 1b = MCLK_OUT function enabled	0
Bits[3:1]	Select internally generated clock	Selects the frequency of the internally generated clock to be output on the MCLK_OUT pin. 000b = crystal clock from internal PLL 001b = audio processor clock (122.88 MHz/2560 × FS) 010b = engine clock 49.152 MHz/1024 × FS) 011b = SRC clock/2 (24.576 MHz/512 × FS) 1xxb = modulator clock (6.144 MHz/128 × FS)	000
Bit[0]	PLL enable	Enables the PLL. 0b = PLL bypassed 1b = PLL enabled	0

Address 0x000B Headphone Control Register (Default: 0x0000)

Table 46.

Bit No.	Bit Name	Description	Default
Bits[15:8]	Reserved	Always write as 0 if writing to this register.	00000000
Bit[7]	HP1 mute	When set to 1, mutes the headphone output immediately without ramping. 0b = unmuted 1b = mute	0
Bit[6]	HP1 short-circuit protect	Enable the short-circuit protection for the headphone amplifier. 0b = disabled 1b = enabled	0
Bit[5]	HP1 tristate	Disables tristating of the headphone amplifier. 0b = enabled 1b = disabled	0
Bits[4:0]	Headphone 1 gain/attenuation	Used to apply analog attenuation to the headphone amplifier. 00000b = 0 dB 00001b = −1.5 dB 00010b = −3 dB … 11101b = −43.5 dB 11110b = −45 dB 11111b = +1.5 dB	00000

Address 0x000C Serial Port Control 2 Register (Default: 0x8004)

It should be noted that the SDIN3, LRCLK0, BCLK0, LRCLK1, BCLK1, LRCLK2, and BCLK2 can also be used as SPDIF_IN0, SPDIF_IN1, SPDIF_IN2, SPDIF_IN3, SPDIF_IN4, SPDIF_IN5, and SPDIF_IN6.

Table 47.

Bit No.	Bit Name	Description	Default
Bits[15:14]	SCR2 clock select	Used to select the serial clocks used for the input to SRC2. 00b = uses LRCLK0 and BCLK0 01b = uses LRCLK1 and BCLK1 10b = uses LRCLK2 and BCLK2 11b = Reserved	10
Bits[13:12]	SRC1 clock select	Used to select the serial clocks used for the input to SRC1. 00b = uses LRCLK0 and BCLK0 01b = uses LRCLK1 and BCLK1 10b = uses LRCLK2 and BCLK2 11b = Reserved	00
Bit[11]	Reserved	Always write as 0 if writing to this register.	0
Bit[10]	Digout Enable 1	Used to change the function of the PWM1A and PWM1B pins to additional serial digital outputs, SDO2 and SDO3. 0b = PWM1A and PWM1B in normal operation 1b = PWM1A and PWM1B used as SDO2 and SDO3	0
Bit[9]	Digout Enable 2	Used to change the function of SPDIF output to serial digital output SDO1. 0b = SPDIF output normal operation 1b = SPDIF output used as SDO1	0

Bit No.	Bit Name	Description	Default
Bits[8:7]	BCLK frequency (master)	Used to set the BCLK frequency when the synchronous serial port is in master mode. 00b = 64 × frequency sample, FS, (3.072 MHz) 01b = 128 × FS (6.144 MHz) 10b = 256 × FS (12.288 MHz) 11b = reserved	00
Bits[6:5]	Reserved	Always write as 0 if writing to this register.	00
Bit[4]	Dither enable	When set to 1, it performs dithering on the digital output when the word width is set to 20 bits or 16 bits. This reduces the effect of truncation noise. 0b = disabled 1b = enabled	0
Bits[3:2]	Synchronous port clock select	Used to select the serial clocks used for the synchronous digital inputs. 00b = uses LRCLK0 and BCLK0 01b = uses LRCLK1 and BCLK1 10b = uses LRCLK2 and BCLK2 11b = reserved	0
Bit[1]	8ch TDM enable	When set to 1, Time Division Multiplexing mode is enabled. 0b = disabled 1b = enabled	0
Bit[0]	Reserved	Always write as 0 if writing to this register.	0

Address 0x000D Reserved (Default: 0x0721)

Address 0x0018 Audio Mute Control 1 Register (Default: 0x7F00)

Table 48.

Bit No.	Bit Name	Description	Default
Bits[15:8]	PWM output latency	Set the delay from the 50/50 duty-cycle square wave to zero on GND when the output is muted and Bit[5] is set to 1. 0x00 = 1.066 ms 0x01 = 2.133 ms … 0x5F = 101.33 ms … 0xFE = 270.93 ms 0xFF = 272 ms	01011111
Bits[7:6]	Reserved	Always write as 0 if writing to this register.	00
Bit[5]	PWM zero enable	Used to specify the final condition of the PWM channels after a mute. 0b = PWM not zeroed after audio mute 1b = PWM zeroed after audio mute	0
Bit[4]	Mute clear select	Mute clear select bit. When the mute pin is used to mute the device, the part can be unmuted in two ways, depending on the condition of this bit. 0b = mute pin rising edge clears mute bit 1b = mute clear gated by clear mute bit	0

Bit No.	Bit Name	Description	Default
Bit[3]	Audio mute	Used to control the software mute. 0b = unmute 1b = mute	0
Bit[2]	Clear mute	Set to 1 to unmute the ADAV4601 when the external pin has been used to mute the part and the mute clear select bit is set. Having performed the required action, this bit automatically resets to 0. 0b = no change 1b = clear pin mute	0
Bit[1]	Mute status (read-only)	Displays the status of the ADAV4601 mute. 0b = currently unmuting or unmuted 1b = currently muting or muted	0
Bit[0]	Mute flag (read-only)	Set to 1 when the ADAV4601 is in mute. 0b = unmuted 1b = muted	0

Address 0x0019 PWM Status Register (Default: 0x0000)

Table 49.

Bit No.	Bit Name	Description	Default
Bits[15:4]	Reserved	Always write as 0 if writing to this register.	000000000000
Bit[3]	PWM4 status	Set when the PWM4 outputs have gone from zero to 50/50 duty cycle. 0b = PWM4 disabled 1b = PWM4 enabled	0
Bits[2]	PWM3 status	Set when the PWM3 outputs have gone from zero to 50/50 duty cycle. 0b = PWM3 disabled 1b = PWM3 enabled	0
Bits[1]	PWM2 status	Set when the PWM2 outputs have gone from zero to 50/50 duty cycle. 0b = PWM2 disabled 1b = PWM2 enabled	0
Bit[0]	PWM1 status	Set when the PWM1 outputs have gone from zero to 50/50 duty cycle. 0b = PWM1 disabled 1b = PWM1 enabled	0

Address 0x001E Reserved (Default: 0x3205)

Address 0x001F PWM Control Register (Default: 0x1070)

Table 50.

Bit No.	Bit Name	Description	Default
Bits[15:14]	PWM Ready Configure	00b = PWM ready forced low 01b = PWM ready forced high 10b = PWM ready late 11b = PWM ready early	00
Bit[13]	Reserved	Always write as 0 if writing to this register.	0

Bit No.	Bit Name	Description	Default
Bit[12]	Reserved	Always write a 1 if writing to this register	1
Bits[11:7]	Reserved	Always write as 0 if writing to this register.	00000
Bits[6:4]	Reserved	Always write a 1 if writing to this register.	111
Bits[3:0]	Reserved	Always write as 0 if writing to this register.	0000

Address 0x0080 Reserved (Default: 0x00FE)

Address 0x0081 Reserved (Default: 0x8202)

Address 0x0085 Reserved (Default: 0x00FE)

Address 0x0086 Reserved (Default: 0x8202)

Address 0x008A SRC Configuration 3 Register (Default: 0x0032)

Table 51.

Bit No.	Bit Name	Description	Default
Bits[15:7]	Reserved	Always write as 0 if writing to this register.	000000000
Bit[6]	SRC2 Channel C enable	Used to enable Channel C of SRC2. 0b = disabled 1b = enabled	0
Bits[5:4]	SRC2 Channel C Input Select	Used to select the serial data input for SRC2 Channel C. 00b =SDIN0 01b = SDIN1 10b = SDIN2 11b = SDIN3	11
Bit[3]	Reserved	Always write as 0 if writing to this register.	0
Bit[2]	SRC2C Channel B enable	Used to enable Channel B of SRC2. 0b = disabled 1b = enabled	0
Bits[1:0]	SRC2C Channel B input select	Used to select the serial data input for SRC2 Channel B. 00b =SDIN0 01b = SDIN1 10b = SDIN2 11b = SDIN3	10

Address 0x008C DAC Standby Register (Default: 0x0000)

Table 52.

Bit No.	Bit Name	Description	Default
Bits[15:5]	Reserved	Always write as 0 if writing to this register.	0000000000
Bit[4]	HP1 Standby	Set to 0 after reset, which means the HP amplifier is in normal mode but is still powered down. 0b = Amplifer in normal mode	0

Bit No.	Bit Name	Description	Default
		1b = Amplifer in low power mode	
Bits[3]	Reserved	Always write as 0 if writing to this register.	0
Bit[2]	AUXDAC1 Standby	Set to 0 after reset, which means the DAC is in normal mode but is still powered down. 0b = DAC in normal mode 1b = DAC in low power mode	0
Bit[1]	AUXDAC3 Standby	Set to 0 after reset, which means the DAC is in normal mode but is still powered down. 0b = DAC in normal mode 1b = DAC in low power mode	0
Bits[0]	HP1 DAC Standby	Set to 0 after reset, which means the HP DAC is in normal mode but is still powered down. 0b = DAC in normal mode 1b = DAC in low power mode	0

Address 0x008D Reserved (Default: 0x0220)

Address 0x008E SPDIF Transmitter Control 2 Register (Default: 0x002D)

Table 53.

Bit No.	Bit Name	Description	Default
Bits[15:8]	Reserved	Always write as 0 if writing to this register.	00000000
Bits[7:4]	Channel status sampling frequency	Used to set the channel status sampling rate in the SPDIF transmitted stream. Should not change the sample rate but only the status bits. 0x0 = 44.1 kHz 0x2 = 48 kHz 0x3 = 32 kHz	0010
Bit[3]	SPDIF TX word length field size	Selects the maximum SPDIF transmitter word length. 0b = 20 bits maximum 1b = 24 bits maximum	1
Bits[2:0]	Transmitter word length	Used to select how many of the bits set by Bit[3] carry valid data. If 24-bit maximum: 0x5 = 24 bits 0x4 = 23 bits 0x2 = 22 bits 0x6 = 21 bits 0x1 = 20 bits If 20-bit maximum: 0x5 = 20 bits 0x4 = 19 bits 0x2 = 18 bits 0x6 = 17 bits 0x1 = 16 bits	101

Address 0x008F Reserved (Default: 0x000A)

Address 0x0200 EEPROM Self Boot Control Register (Default: 0x0000)

Table 54.

Bit No.	Bit Name	Description	Default
Bits[15]	Self boot enable	It initiates a self boot from the external EEPROM. 0b = normal operation 1b = initiates self boot	0
Bits[14:6]	Reserved	Always write as 0 if writing to this register.	000000000
Bit[5]	Safe load target/slew RAM	Initiates a safe load to target/slew RAM. Cleared when safe load completed. 0b = finished 1b = safe load request	0
Bit[4]	Safe load parameter RAM	Initiates a safe load to parameter RAM. Cleared when safe load completed. 0b = finished 1b = safe load request	0
Bits[3:0]	Reserved	Always write as 0 if writing to this register.	0000

Address 0x0316 EEPROM Device Address Register (Default: 0x0050)

Table 55.

Bit No.	Bit Name	Description	Default
Bits[15:7]	Reserved	Always write as 0 if writing to this register.	000000000
Bits [6:0]	EEPROM device address	0x50 = sets external EEPROM address	1010000

Address 0x0317 EEPROM Data Address Register (Default: 0x0000)

Table 56.

Bit No.	Bit Name	Description	Default
Bits[15:0]	EEPROM start address	0x0000 = default address	0000000000000000

APPENDIX B – LAYOUT RECOMMENDATIONS

DECOUPLING

The following pins must be decoupled to ground with the following priority

- VREF
- FILTA
- FILTD
- PLL_LF
- SUPPLIES (AVDD, DVDD, ODVDD)

All the Ceramic Bypass Capacitors must be decoupled to the corresponding GND pins before seeing the ground plane. See Figure 75 for more details.

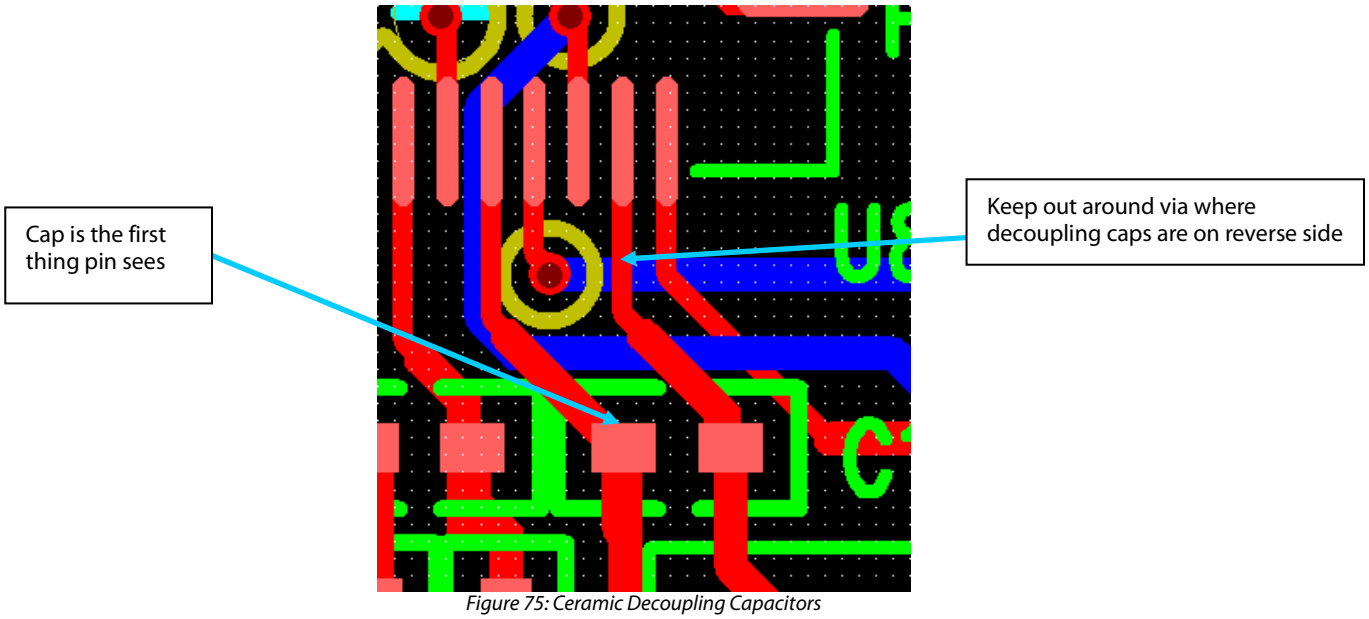


Figure 75: Ceramic Decoupling Capacitors

The ceramic capacitors should be placed as close to the ADAV4601 as possible, governed by the priority listing above. The larger electrolyte capacitors should then placed as closed to the ADAV4601 as possible in the same priority as listed above.
In addition, the 1nF and 100nF ceramic capacitors for the PLL Loop Filter should both be placed as close as possible to the ADAV4601.

Note: If a ceramic capacitor has to be placed on the underside of the evaluation board, ensure there is a keep out put around the via to ensure the pin sees the cap first. Do not leave the pin see the ground plane first. See the above figure for more details regarding this.

CRYSTAL OSCILLATOR CIRCUIT

Traces for the oscillator circuit should be kept as short as possible to avoid any possible fluctuations in the crystal frequency due to parasitic capacitance. In addition, avoid long board traces connected to any of these components because such traces may affect crystal start-up and operation.

PWM OUTPUTS

All PWM Output differential pairs should be matched in length, i.e. PWM1A = PWM1B, PWM2A = PWM2B.

GROUND PLANE

A split ground plane should be used in the layout of the ADAV4601 with the Analog and Digital Grounds connected underneath the ADAV4601 using a single link. This is to avoid possible ground loop currents in the Analog and Digital ground planes. Components in an analog signal path should be placed away from the digital ground plane and vice versa.

APPENDIX C – ADAV4601 BILL OF MATERIALS

The table below is the bill of materials for the ADAV4601 evaluation board. The minimum BOM required can be found by just using the parts highlighted in green. These correspond to the components shown in Figure 76: Typical Application Diagram.

Table 57. ADAV4601 Bill of Materials

Part	Value	Description	ADI/Farnell/Digikey Order Codes
U16		IC, SM EEPROM SERIAL 64K	FEC 9758070
U15		IC, SM EEPROM SERIAL 512K	FEC 9758038
J12		T header (3-pin SIP header&1SIP header)	FEC150411
U12		IC, SM LOGIC OCTAL BUFFER/DRIVER	FEC 1236353
U8		ADAV4601	ADAV4601BSTZ
U17		ADG736	ADG736BRMZ
U18		Microprocessor Supervisory Circuit	ADM809SARTZ
U14		IC SUPVSR W/RESET 2.63V SOT143-4	ADM811-3RARTZ
U35		IC REG LDO 1A PREC 1.8V SOT-223	ADP3338AKCZ-1.8
U27		IC REG LDO 1A PREC 3.3V SOT-223	ADP3338AKCZ-3.3
U28		IC REG LDO 1A PREC 3.3V SOT-223	ADP3338AKCZ-3.3
L18	0 Ohm	FERRITE BEAD, 0805 600Z	FEC 9331662
L1	600Z	FERRITE BEAD, 0805 600Z	FEC 1193423
L2	600Z	FERRITE BEAD, 0805 600Z	FEC 1193423
L8	600Z	FERRITE BEAD, 0805 600Z	FEC 1193423
L10	600Z	FERRITE BEAD, 0805 600Z	FEC 1193423
L14	600Z	FERRITE BEAD, 0805 600Z	FEC 1193423
L17	600Z	FERRITE BEAD, 0805 600Z	FEC 1193423
L23	600Z	FERRITE BEAD, 0805 600Z	FEC 1193423
C37	0.1uF	SMD Capacitor	FEC 3019482
C53	0.1uF	SMD Capacitor	FEC 3019482
C75	0.1uF	SMD Capacitor	FEC 3019482
C154	0.1uF	SMD Capacitor	FEC 3019482
C156	0.1uF	SMD Capacitor	FEC 3019482
C158	0.1uF	SMD Capacitor	FEC 3019482
C91	0.1uF	SMD Capacitor	FEC 3019482
C92	0.1uF	SMD Capacitor	FEC 3019482
C96	0.1uF	SMD Capacitor	FEC 3019482
C98	0.1uF	SMD Capacitor	FEC 3019482
C43	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C2	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C3	0.1uF	SMD Capacitor	FEC 3019482
C4	0.1uF	SMD Capacitor	FEC 3019482
C5	0.1uF	SMD Capacitor	FEC 3019482
C6	0.1uF	SMD Capacitor	FEC 3019482
C11	0.1uF	SMD Capacitor	FEC 3019482
C26	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C31	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C32	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C35	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C40	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989

C39	0.1uF	SMD Capacitor	FEC 3019482
C60	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C180	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C181	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C188	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C191	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C200	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C23	0.1uF	SMD Capacitor	FEC 3019482
C25	0.1uF	SMD Capacitor	FEC 3019482
C33	100nF	CAPACITOR, 0805 100NF 50V	FEC 3019949
C8	10nF	CAP CER 10000PF 50V 5% COG 0805	Digikey 490-1642-1-ND
C20	10nF	CAP CER 10000PF 50V 5% COG 0805	Digikey 490-1642-1-ND
C24	10nF	CAP CER 10000PF 50V 5% COG 0805	Digikey 490-1642-1-ND
C29	10nF	CAP CER 10000PF 50V 5% COG 0805	Digikey 490-1642-1-ND
C41	10nF	CAP CER 10000PF 50V 5% COG 0805	Digikey 490-1642-1-ND
C42	10nF	CAP CER 10000PF 50V 5% COG 0805	Digikey 490-1642-1-ND
C34	12pF	CAPACITOR, 0603 12PF 50V	FEC 721979
C36	12pF	CAPACITOR, 0603 12PF 50V	FEC 721979
C162	18pF	CAPACITOR, 0603 18PF 50V; NPO	FEC 721992
C175	18pF	CAPACITOR, 0603 18PF 50V; NPO	FEC 721992
C97	2.2uF	SMD Capacitor	FEC 9402101
C99	2.2uF	SMD Capacitor	FEC 9402101
C1	10nF	CAPACITOR, 0603 10NF 25V	FEC 3019561
C7	10nF	CAPACITOR, 0603 10NF 25V	FEC 3019561
C15	1nF	CAPACITOR, 0603 1NF NPO 50V	FEC 317202
C9	22nF	CAPACITOR, 0603 22NF 50V	FEC 3019755
C82	100uF	CAPACITOR, CASE C 100UF 6.3V	FEC 9188800
C90	100uF	CAPACITOR, CASE C 100UF 6.3V	FEC 9188800
C93	100uF	CAPACITOR, CASE E 100UF 16V	FEC 9694404
C16	10uF	CAPACITOR, CASE B 10UF 16V	FEC 9753893
C38	10uF	CAPACITOR, CASE C 10UF 35V	FEC 9694471
C54	10uF	CAPACITOR, CASE C 10UF 35V	FEC 9694471
C59	10uF	CAPACITOR, CASE C 10UF 35V	FEC 9694471
C114	10uF	CAPACITOR, CASE C 10UF 35V	FEC 9694471
C127	10uF	CAPACITOR, CASE C 10UF 35V	FEC 9694471
C129	10uF	CAPACITOR, CASE C 10UF 35V	FEC 9694471
C134	10uF	CAPACITOR, CASE C 10UF 35V	FEC 9694471
C146	10uF	CAPACITOR, CASE C 10UF 35V	FEC 9694471
C76	10uF	CAPACITOR, CASE B 10UF 16V	FEC 9696792
C77	10uF	CAPACITOR, CASE B 10UF 16V	FEC 9696792
C78	10uF	CAPACITOR, CASE B 10UF 16V	FEC 9696792
C79	10uF	CAPACITOR, CASE B 10UF 16V	FEC 9696792
C84	10uF	CAPACITOR, CASE B 10UF 16V	FEC 9696792
C85	10uF	CAPACITOR, CASE B 10UF 16V	FEC 9696792
C183	220uF	CAPACITOR, CASE E 220UF 6.3V	FEC 9694340
C186	220uF	CAPACITOR, CASE E 220UF 6.3V	FEC 9694340
C205	220uF	CAPACITOR, CASE E 220UF 6.3V	FEC 9694340
C145	47uF	CAPACITOR, CASE C 47UF 6.3V	FEC 9694323
C147	47uF	CAPACITOR, CASE C 47UF 6.3V	FEC 9694323

C115	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C116	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C117	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C118	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C125	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C126	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C128	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C131	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C132	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C135	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C137	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C139	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C142	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C143	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C144	0.1u	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
C130	1000pF	CAP CERAMIC 1000PF 50V NP0 0603	Digikey PCC2151CT-ND
C133	1000pF	CAP CERAMIC 1000PF 50V NP0 0603	Digikey PCC2151CT-ND
C28	0.1uF	CAPACITOR, 0603 100NF 50V; Y5V	FEC 431989
J3		DIN41612 PCB CONNECTOR 96-PIN MALE	FEC 1097922
J2		SOCKET, PCB DC POWER 2.1MM	FEC 224-959 (PK10)
U2		IC, SM TRANSMITTER DIGITAL AUDIO	FEC 1023448
U19		IC, SM RECEIVER DIGITAL AUDIO	FEC 1023452
U26		USB Microcontroller	Digikey 428-1680-ND
D12		DIODE, 1A 400V; 1N4004	FEC 9565027
J6		HEADER, 2 ROW VERT 20WAY	FEC 9733744
J17		HEADER, 1 ROW 2WAY & JUMPER SOCKET	FEC 1022247 & 150411
LK1		HEADER, 1 ROW 2WAY & JUMPER SOCKET	FEC 1022247 & 150411
LK8		HEADER, 1 ROW 2WAY & JUMPER SOCKET	FEC 1022247 & 150411
LK14		HEADER, 1 ROW 2WAY & JUMPER SOCKET	FEC 1022247 & 150411
LK11		HEADER, 1 ROW 3WAY	FEC 1022248 & 150-411
D3		LED, SMD PLCC-2 YELLOW	FEC 1058412
D13		LED, SMD, GREEN	FEC 1318243
U1		IC, SM NC7S TINY LOGIC	FEC 1014136
U9		IC, SM NC7S TINY LOGIC	FEC 1014136
U4		Buffer/Line Driver	FEC 1013811
U5		Buffer/Line Driver	FEC 1013811
U6		Buffer/Line Driver	FEC 1013811
J7		SOCKETS, 4X PHONO, PCB	FEC 1280670
J8		SOCKETS, 4X PHONO, PCB	FEC 1280670
J13		SOCKETS, 4X PHONO, PCB	FEC 1280670
J20		SOCKETS, 4X PHONO, PCB	FEC 1280670
R108	0R	RESISTOR, 0603 0R0	FEC 9331662
R143	0R	RESISTOR, 0603 0R0	FEC 9331662
R196	100k	SMD Resistor (0.063W)	FEC 9330402
R26	10k	RESISTOR, 0603 10K	FEC 9330399
R111	10k	RESISTOR, 0603 10K	FEC 9330399
R1	10k	RESISTOR, 0603 10K	FEC 9330399
R96	10k	RESISTOR, 0603 10K	FEC 9330399
R27	1k	RESISTOR, 0603 1K	FEC 9330380

R106	1k	RESISTOR, 0603 1K	FEC 9330380
R33	20k	Thick Film Resistor Series:MC0603	FEC 1127448
R34	20k	Thick Film Resistor Series:MC0603	FEC 1127448
R43	20k	Thick Film Resistor Series:MC0603	FEC 1127448
R45	2k	RESISTOR, 0603 2K	FEC 9330763
R72	2k	RESISTOR, 0603 2K	FEC 9330763
R107	2k	RESISTOR, 0603 2K	FEC 9330763
R109	2k	RESISTOR, 0603 2K	FEC 9330763
R75	2k	RES 2.0K OHM 1/10W 5% 0603 SMD	Digikey P2.0KGCT-ND
R209	2k2	RESISTOR, 0603 2K2	FEC 9330810
R6	330R	RESISTOR, 0805 330R	FEC 1099797
R52	33R	RES 33 OHM 1/10W 5% 0603 SMD	Digikey P33GCT-ND
R76	33R	RES 33 OHM 1/10W 5% 0603 SMD	Digikey P33GCT-ND
R79	33R	RES 33 OHM 1/10W 5% 0603 SMD	Digikey P33GCT-ND
R80	33R	RES 33 OHM 1/10W 5% 0603 SMD	Digikey P33GCT-ND
R83	33R	RES 33 OHM 1/10W 5% 0603 SMD	Digikey P33GCT-ND
R84	33R	RES 33 OHM 1/10W 5% 0603 SMD	Digikey P33GCT-ND
R85	33R	RES 33 OHM 1/10W 5% 0603 SMD	Digikey P33GCT-ND
R144	33R	RES 33 OHM 1/10W 5% 0603 SMD	Digikey P33GCT-ND
R50	47k	RESISTOR, 0603 47K	FEC 9331255
R28	0R	RESISTOR, 0603 0R0	FEC 9331662
R41	0R	RESISTOR, 0603 0R0	FEC 9331662
R46	0R	RESISTOR, 0603 0R0	FEC 9331662
R47	0R	RESISTOR, 0603 0R0	FEC 9331662
R2	10k	RESISTOR, 0603 10K	FEC 9330399
R11	10k	RESISTOR, 0603 10K	FEC 9330399
R29	10k	RESISTOR, 0603 10K	FEC 9330399
R30	10k	RESISTOR, 0603 10K	FEC 9330399
R37	10k	RESISTOR, 0603 10K	FEC 9330399
R38	10k	RESISTOR, 0603 10K	FEC 9330399
R40	10k	RESISTOR, 0603 10K	FEC 9330399
R44	10k	RESISTOR, 0603 10K	FEC 9330399
R4	3k	RESISTOR, RC21 0603 3K	FEC 9330976
R10	47k	RESISTOR, 0603 47K	FEC 9331255
R12	47k	RESISTOR, 0603 47K	FEC 9331255
R14	47k	RESISTOR, 0603 47K	FEC 9331255
R15	47k	RESISTOR, 0603 47K	FEC 9331255
R16	47k	RESISTOR, 0603 47K	FEC 9331255
R19	47k	RESISTOR, 0603 47K	FEC 9331255
R20	47k	RESISTOR, 0603 47K	FEC 9331255
R22	47k	RESISTOR, 0603 47K	FEC 9331255
R9	4k7	SMD Resistor (0.063W)	FEC 9331247
R3	0R	RESISTOR, 0603 0R0	FEC 9331662
R5	0R	RESISTOR, 0603 0R0	FEC 9331662
R17	0R	RESISTOR, 0603 0R0	FEC 9331662
R18	0R	RESISTOR, 0603 0R0	FEC 9331662
R126	0R	RESISTOR, 0603 0R0	FEC 9331662
R7	10k	RESISTOR, 0603 10K	FEC 9330399
R8	10k	RESISTOR, 0603 10K	FEC 9330399

R13	10k	RESISTOR, 0603 10K	FEC 9330399
U29		SN74LVC1T45 LOGIC, Buffer/Level Shifter	FEC 1470916
J10		Vertical Stereo Jack	FEC 1243242
S3		SWITCH, SPNO SMD	FEC 177807
S1		Sealed Ultra Miniature SPST Toggle Switch	C&K GT12MCKE
TP3		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP6		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP7		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP15		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP16		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP17		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP18		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP19		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP20		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP22		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP29		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP30		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP31		TERMINAL, PCB RED PK100	FEC 8731144 (Pack 100)
TP8		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP9		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP10		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP11		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP12		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP13		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP14		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP21		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP23		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP24		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP25		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP32		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP33		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP34		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP42		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
TP43		TERMINAL, PCB BLACK PK100	FEC 8731128 (Pack 100)
U7		OPTICAL RECEIVER 3.3V	FEC 1225769
U3		OPTICAL TRANSMITTER 3.3V	FEC 1225776
J18		SOCKET, USB MINI-AB SMT + CABLE	FEC 9786490 + FEC 1308879
Y3	24.576MHz	CRYSTAL 24.576 MHZ HC49/US	Digikey 300-8512-ND
Y2	24MHz	CRYSTAL 24MHZ	FEC 9509640

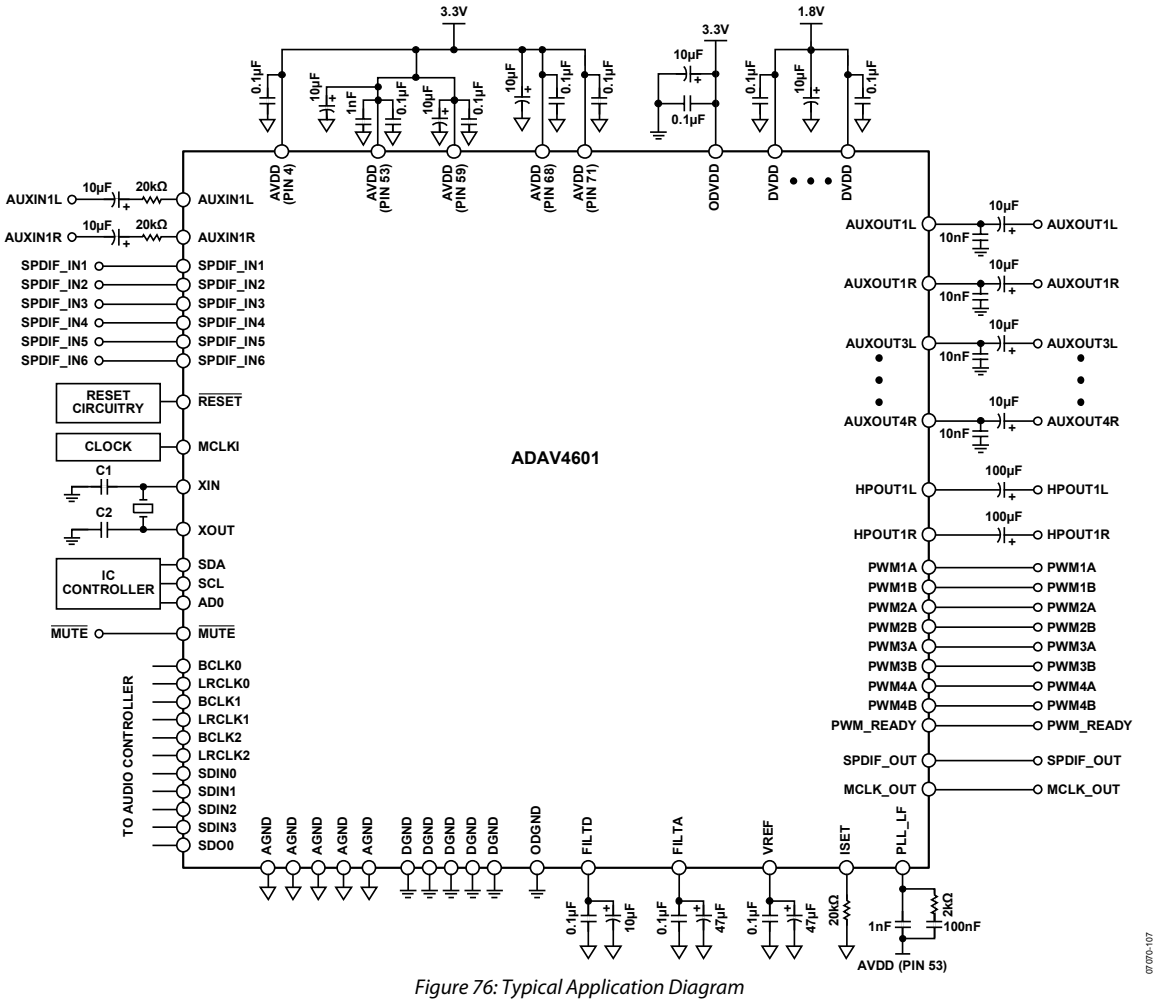
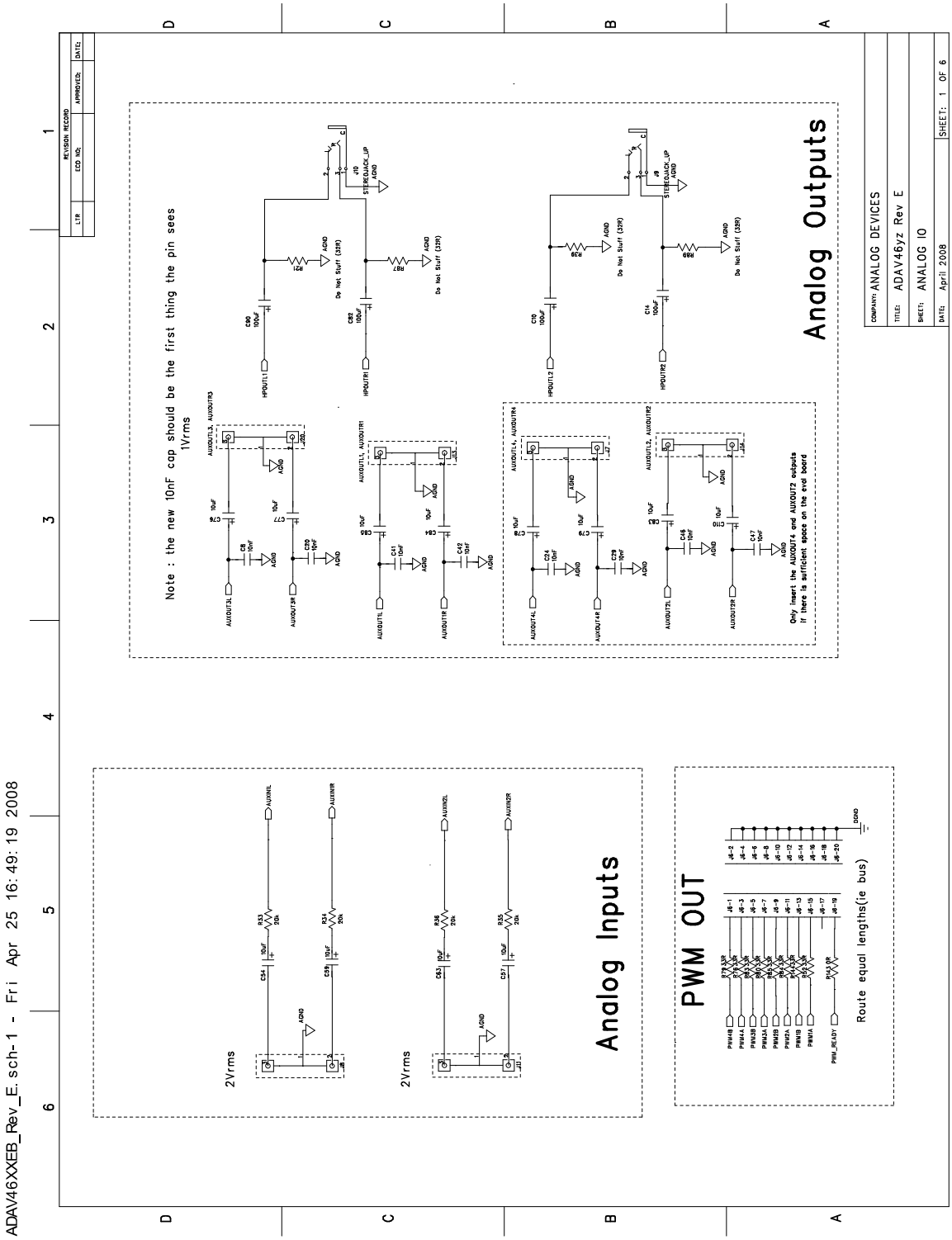
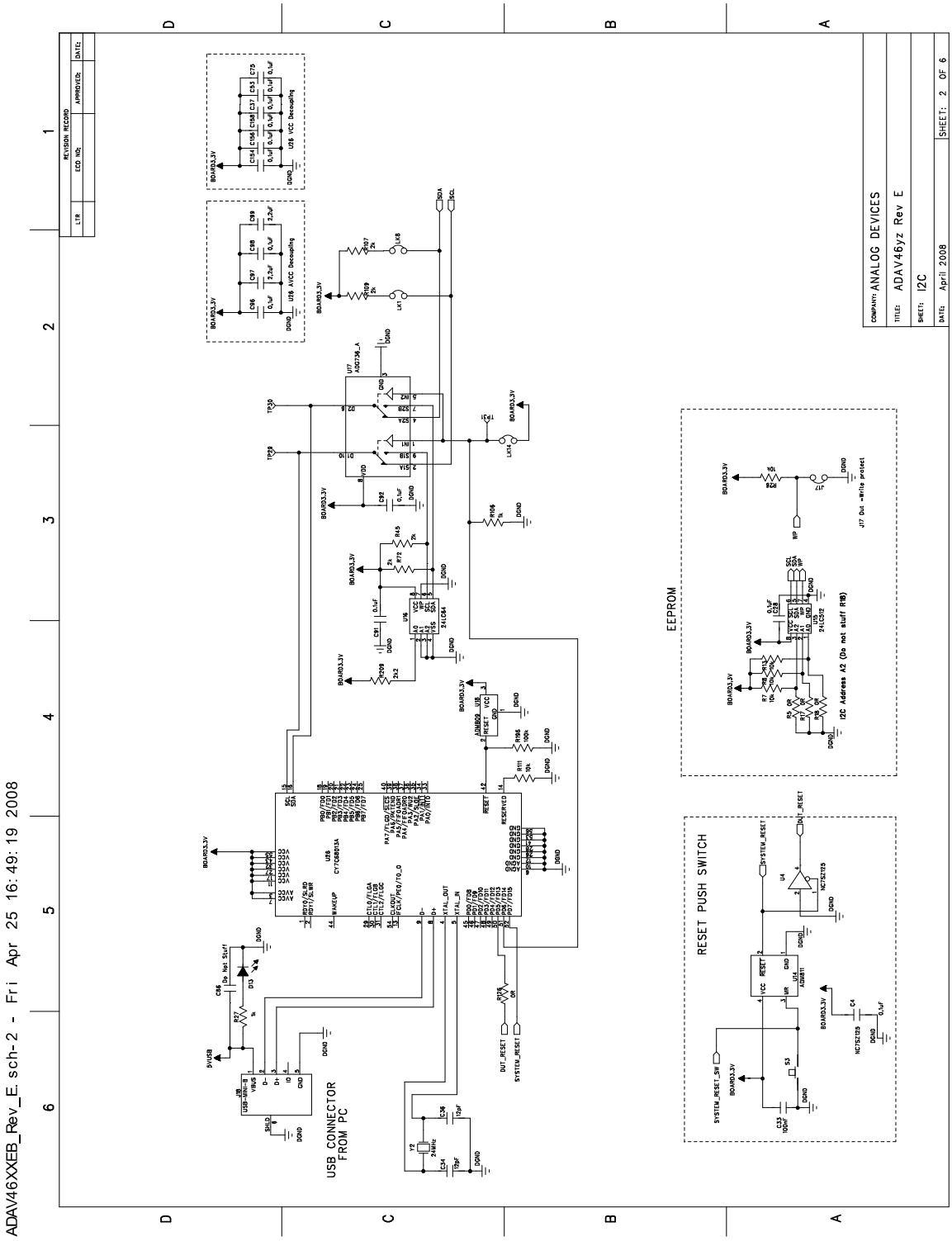
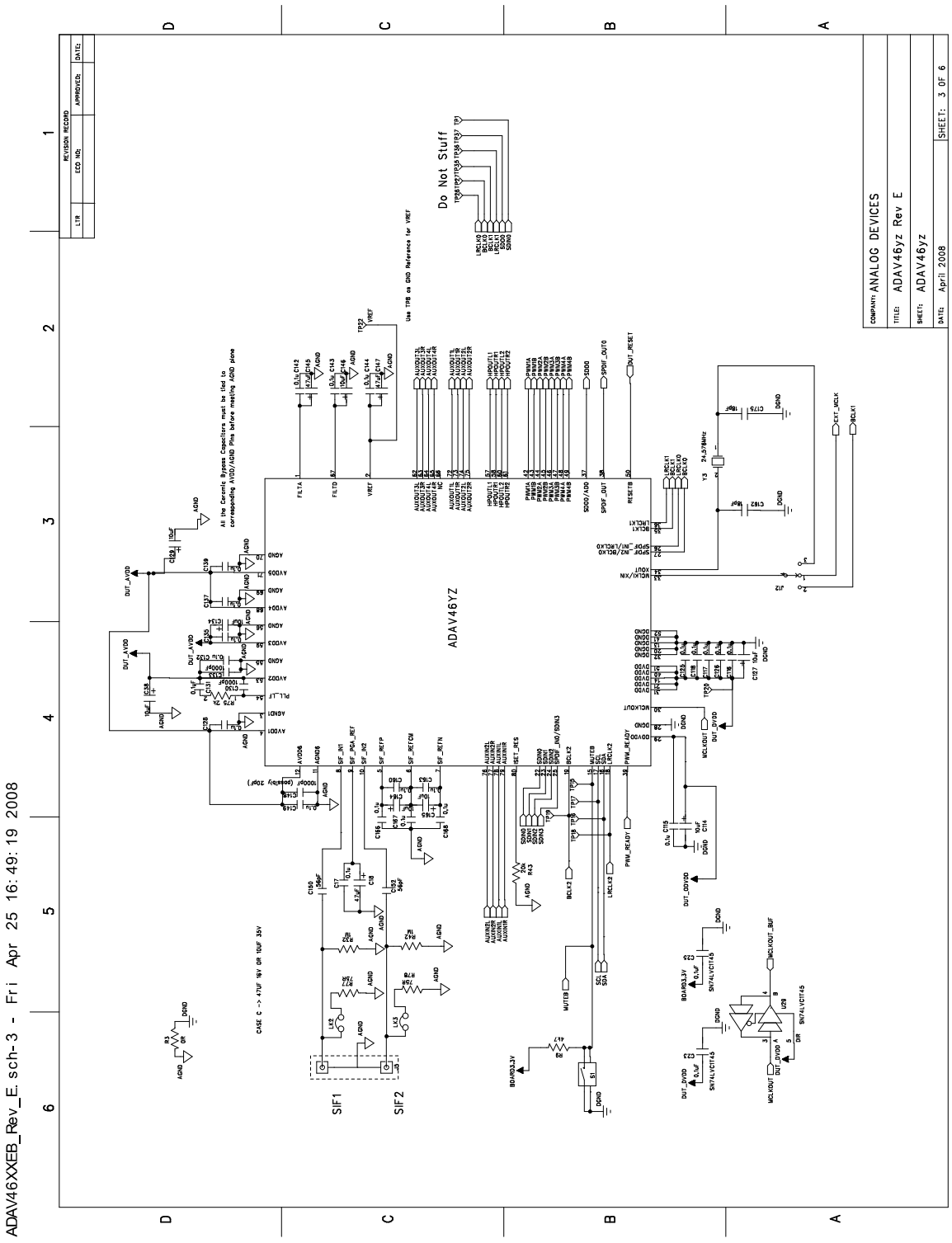


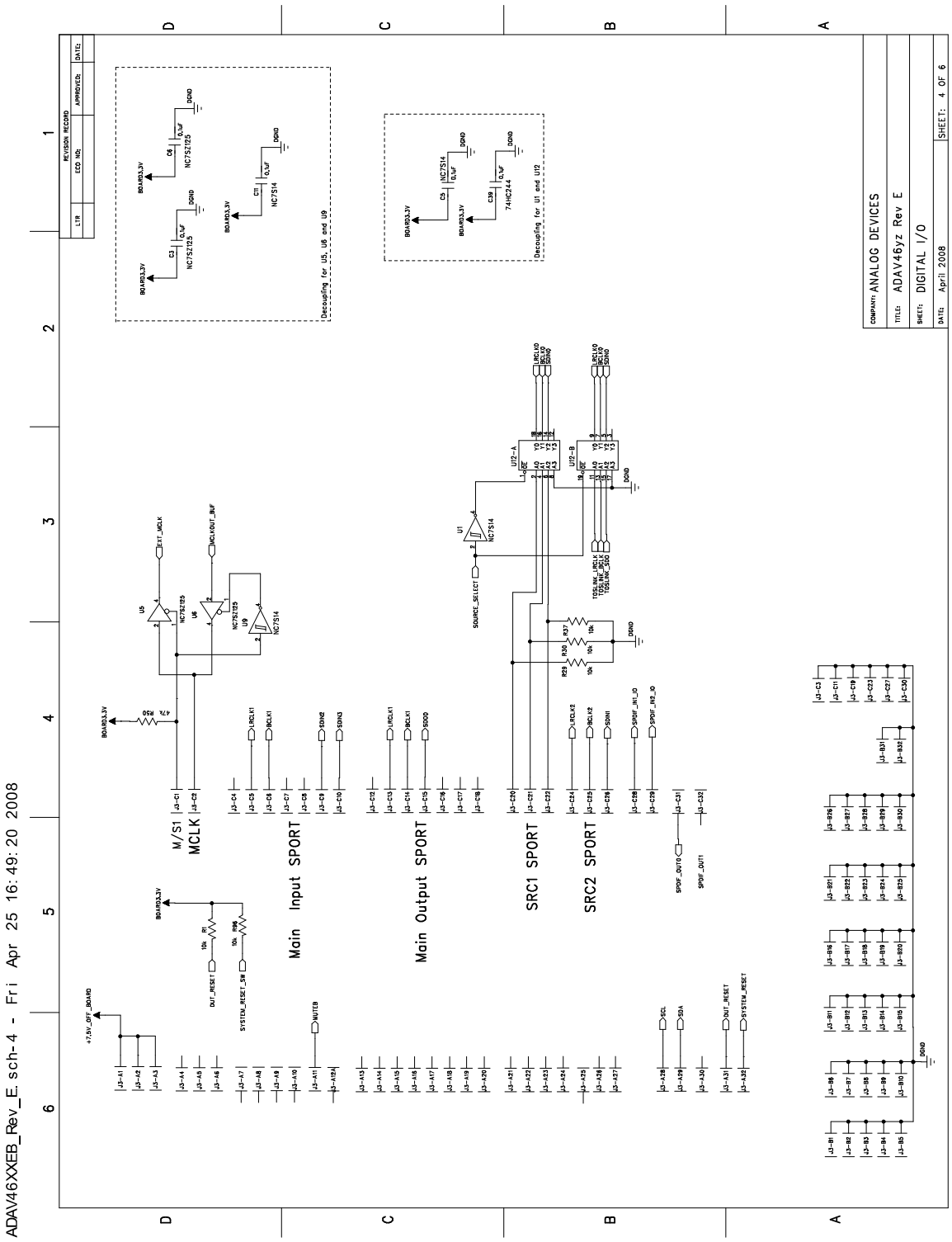
Figure 76: Typical Application Diagram

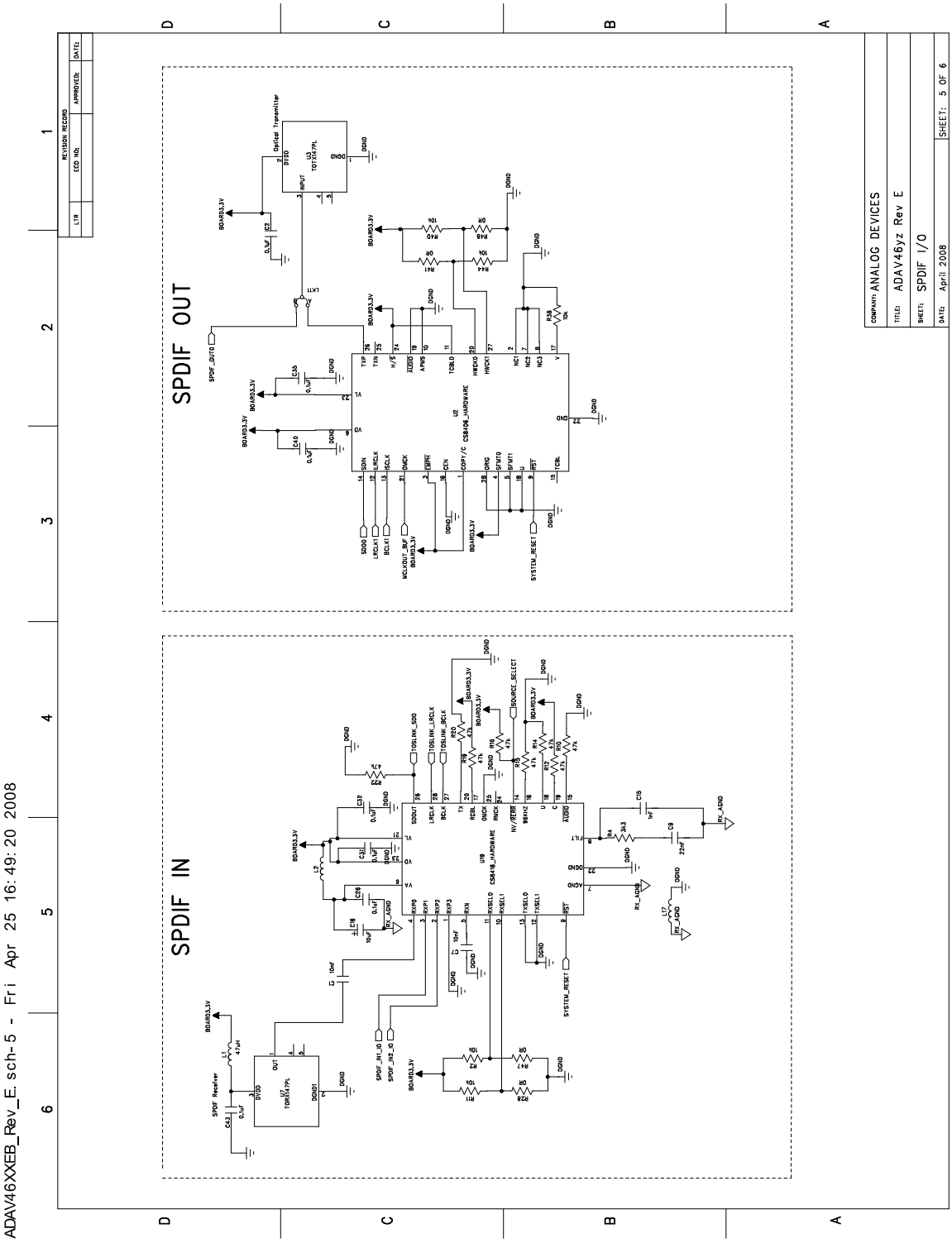
APPENDIX D – SCHEMATICS

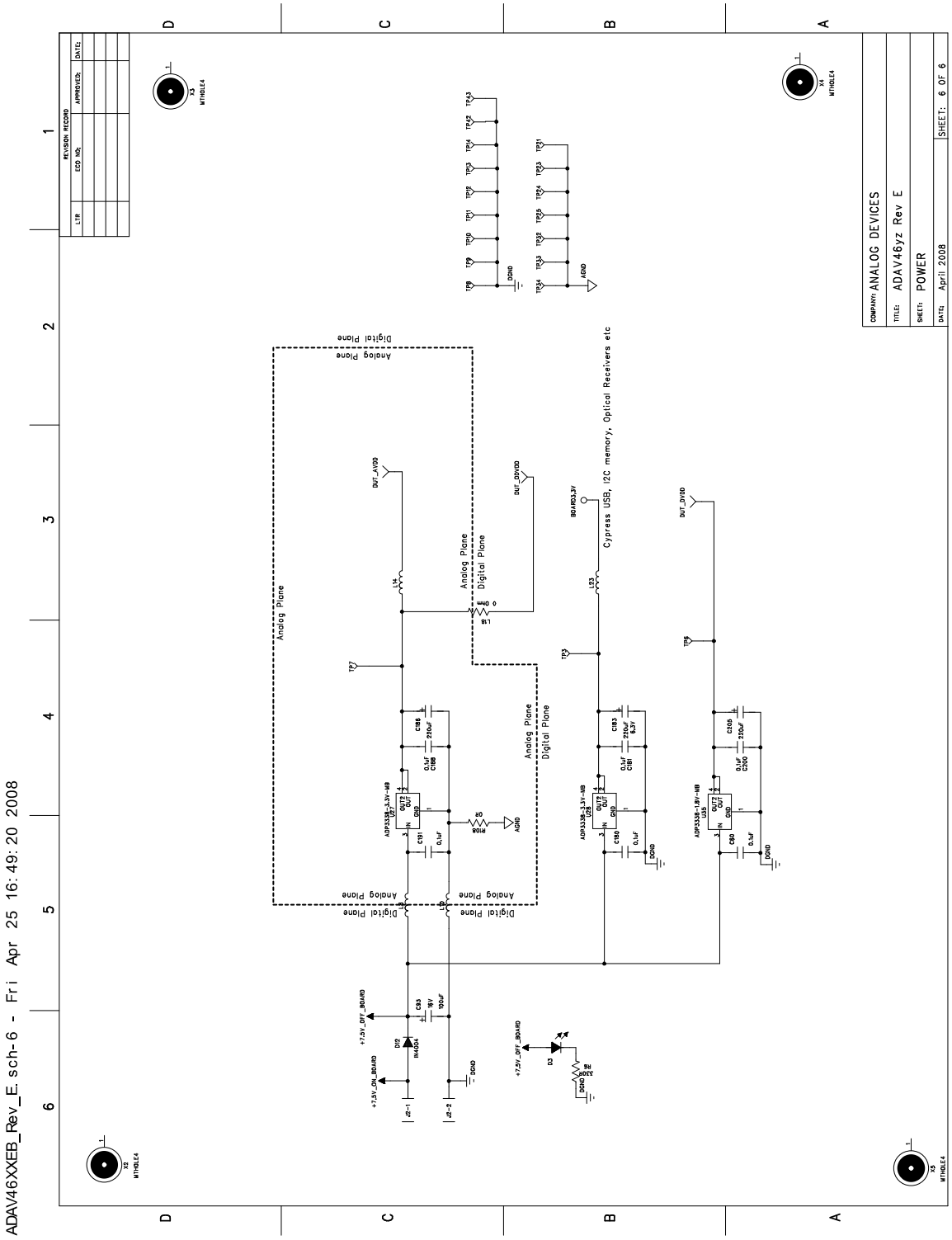



ADAV46XXEB_Rev_E.sch-2 - Fri Apr 25 16:49:19 2008
USB CONNECTOR FROM PC
EEPROM
RESET PUSH SWITCH
COMPANY: ANALOG DEVICES
TITLE: ADAV46yz Rev E
SHEET: I2C
DATE: April 2008
SHEET: 2 OF 6


ADAV46XXEB_Rev_E.sch - 3 - Fri Apr 25 16:49:19 2008
ADAV46YZ
Do Not Stuff
SIF1
SIF2
COMPANY: ANALOG DEVICES
TITLE: ADAV46yz Rev E
SHEET: ADAV46yz
DATE: April 2008
SHEET: 3 OF 6


ADAV46XXEB_Rev_E.sch-4 - Fri Apr 25 16:49:20 2008
Main Input SPORT
Main Output SPORT
SRC1 SPORT
SRC2 SPORT
M/S1
MCLK
COMPANY: ANALOG DEVICES
TITLE: ADAV46yz Rev E
SHEET: DIGITAL I/O
DATE: April 2008
SHEET: 4 OF 6


SPDIF OUT
SPDIF IN
ADAV46XXEB_Rev_E.sch-5 - Fri Apr 25 16:49:20 2008
COMPANY: ANALOG DEVICES
TITLE: ADAV46yz Rev E
SHEET: SPDIF I/O
DATE: April 2008
SHEET: 5 OF 6


ADAV46XXEB_Rev_E.sch-6 - Fri Apr 25 16:49:20 2008
COMPANY: ANALOG DEVICES
TITLE: ADAV46yz Rev E
SHEET: POWER
DATE: April 2008
SHEET: 6 OF 6
REVISION RECORD
LTR
ECO NO:
APPROVED:
DATE:
Analog Plane
Digital Plane
Cypress USB, I2C memory, Optical Receivers etc
DUT_AVDD
DUT_DOVDD
DUT_DVDD
BOARD3.3V
DGND
AGND
+7.5V_OFF_BOARD
+7.5V_ON_BOARD
MTHOLE4