

Digital Filtering:

TD 01: Introduction to SigmaDSP

Antonin NOVAK

Laboratoire d'Acoustique de l'Université du Maine, Le Mans, France



Goal:

basics of SigmaDSP (ADAU1701)

- know what a SigmaDSP is
- know what an evaluation board is

learn basics of Sigma Studio

- know how to create new project and make it work with a SigmaDSP
- program a simple stereo Input/Output talk-through
- fixed point architecture

work

- parametric equalizer on SigmaDSP
- Self-Boot E2Prom (make the DSP standalone)
- Stereo mixer
- Multiplexer / Demultiplexer
- Index selectable choice



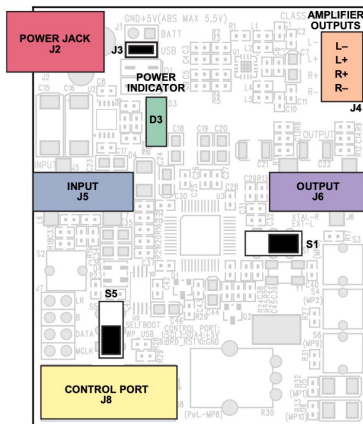
1 Evaluation board - Eval ADAU1701

SigmaDSP - ADAU1701

Sigma Studio

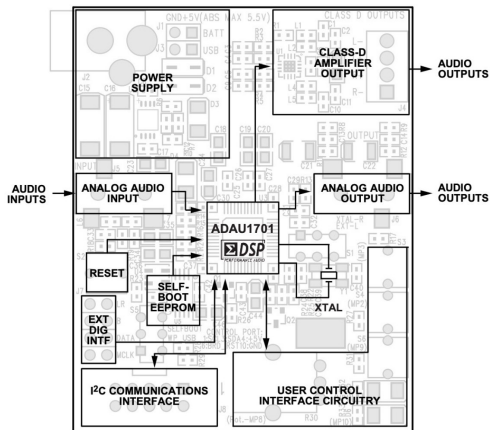
Work

Evaluation board - Eval ADAU1701

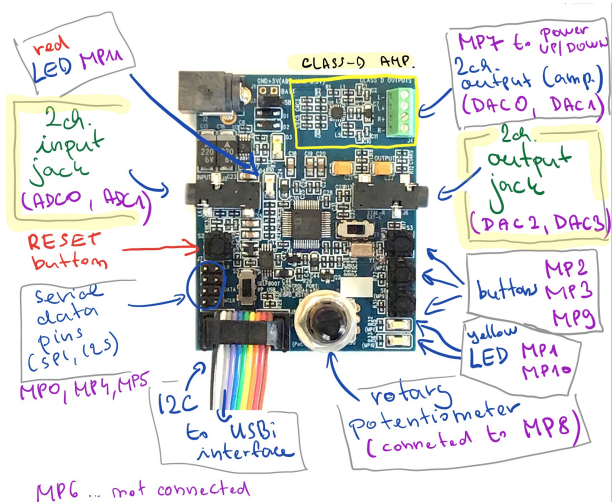


Evaluation board - Eval ADAU1701

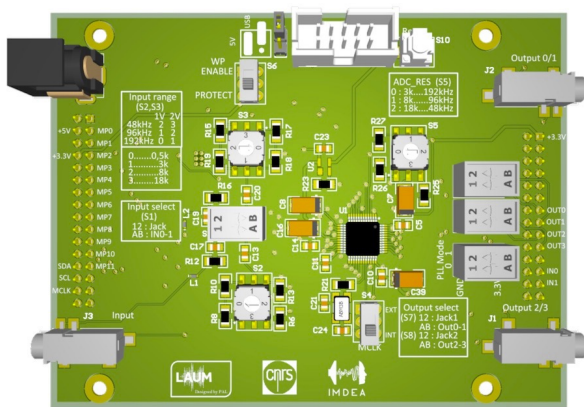
FUNCTIONAL BLOCK DIAGRAM



Evaluation board - Eval ADAU1701



DSP board - LAUM





Evaluation board - Eval ADAU1701

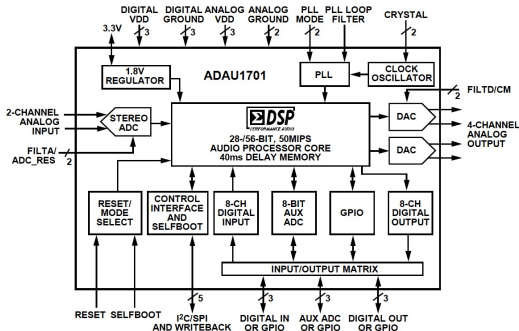
2 SigmaDSP - ADAU1701

Sigma Studio

Work

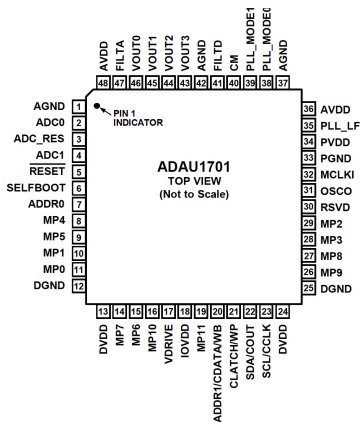
ADAU1701 - SigmaDSP 28-/56-Bit Audio Processor

- 2 ADCs, 4 DACs
- standalone operation with self-boot from serial EEPROM
- auxiliary ADC with 4-input mux for analog control
- GPIOs for digital controls and outputs (multipurpose pins)



ADAU1701 - SigmaDSP 28-/56-Bit Audio Processor

ADAU1701 is just a chip

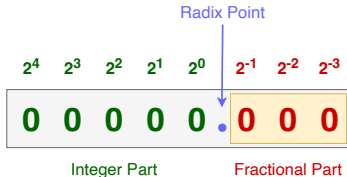


ADAU1701 - SigmaDSP 28-/56-Bit Audio Processor

a SigmaDSP can be programmed only
using Sigma Studio (graphical programming)

Fixed point architecture

In fixed point representation, the number is split into an **integer part (A)** and a **fractional part (B)**.



This 8-bit example would be called 5.3 fixed point, meaning **A=5 bits** for the integer part and **B=3 bits** for the fractional part.

ADAU1701 - 28 bit fixed point architecture

Audio Signal ... 5.23 format



0000 1000 0000 0000 0000 0000 0000 = 1.0

Control Signals and Index Table ... 28.0 format



0000 0000 0000 0000 0000 0000 0100 = 4

ADAU1701 - 28 bit fixed point architecture

Examples of ... 5.23 format

```
0000 0000 0000 0000 0000 0000 0000 = 0.0
0000 0010 0000 0000 0000 0000 0000 = 0.25
0000 0100 0000 0000 0000 0000 0000 = 0.5
0000 1000 0000 0000 0000 0000 0000 = 1.0 (0 dBFS)
0001 0000 0000 0000 0000 0000 0000 = 2.0
0010 0000 0000 0000 0000 0000 0000 = 4.0
0111 1111 1111 1111 1111 1111 1111 = (16.0 - 1 LSB)
```

```
1000 0000 0000 0000 0000 0000 0000 = -16.0
1110 0000 0000 0000 0000 0000 0000 = -4.0
1111 0000 0000 0000 0000 0000 0000 = -1.0
1111 1000 0000 0000 0000 0000 0000 = -1.0
1111 1100 0000 0000 0000 0000 0000 = -0.5
1111 1110 0000 0000 0000 0000 0000 = -0.25
1111 1111 1111 1111 1111 1111 1111 = (1 LSB below 0.0)
```

ADAU1701 - 28 bit fixed point architecture

Examples of ... 28.0 format

```
0000 0000 0000 0000 0000 0000 0000 = 0
0000 0000 0000 0000 0000 0000 0001 = 1
0000 0000 0000 0000 0000 0000 0010 = 2
0000 0000 0000 0000 0000 0000 0011 = 3
0000 0000 0000 0000 0000 0000 0100 = 4
0111 1111 1111 1111 1111 1111 1111 = 134217727 ( $2^{27}$ )
```

```
1000 0000 0000 0000 0000 0000 0000 = -134217728 ( $2^{27}$ )
1111 1111 1111 1111 1111 1111 1100 = -4
1111 1111 1111 1111 1111 1111 1101 = -3
1111 1111 1111 1111 1111 1111 1110 = -2
1111 1111 1111 1111 1111 1111 1111 = -1
```



3

Evaluation board - Eval ADAU1701

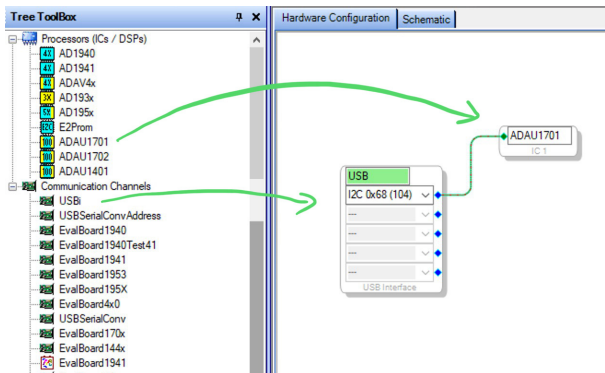
SigmaDSP - ADAU1701

Sigma Studio

Work

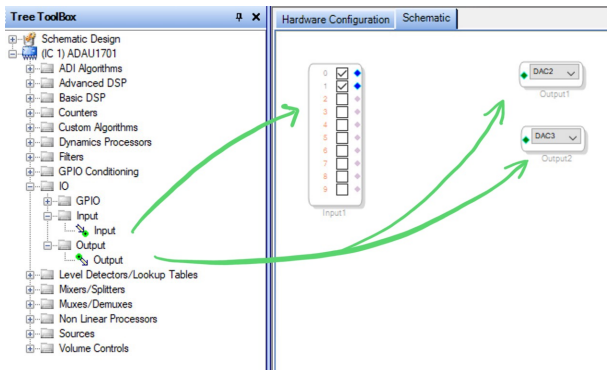
Sigma Studio - starting a project

on *Hardware Configuration* tab :
drag and drop USBi and ADAU1701 and connect them together



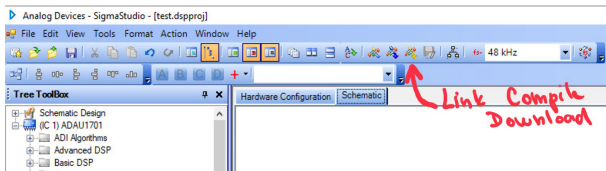
Sigma Studio - starting a project

on *Schematic* tab :
drag and drop Input and Output (in IO folder)
and connect them !!!



Sigma Studio - starting a project

1. Connect inputs to outputs (talk-through)
2. Link, Compile, and Download (one button)
and thats it !!!



3. Connect your phone (music) to input jack and your phones to output jack and check if the talk-through works



Evaluation board - Eval ADAU1701

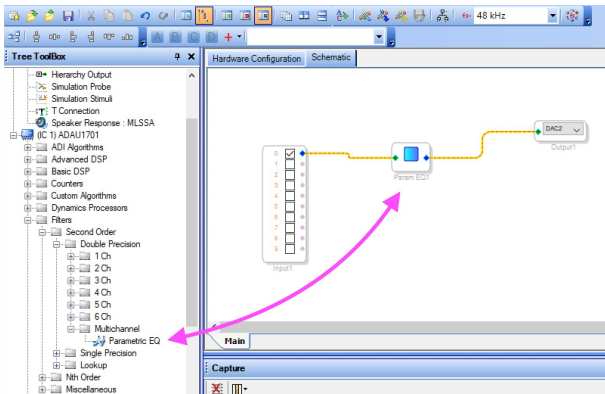
SigmaDSP - ADAU1701

Sigma Studio

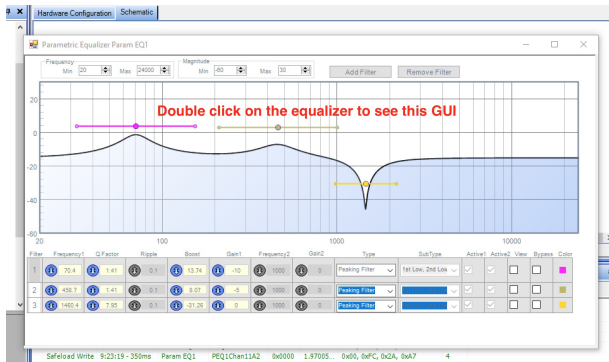
4

Work

Work 1: Parametric Equalizer



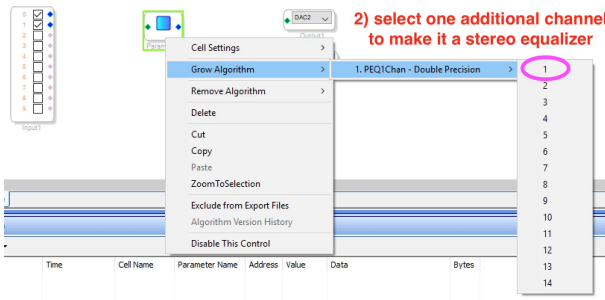
Work 1: Parametric Equalizer



Work 1: Parametric Equalizer

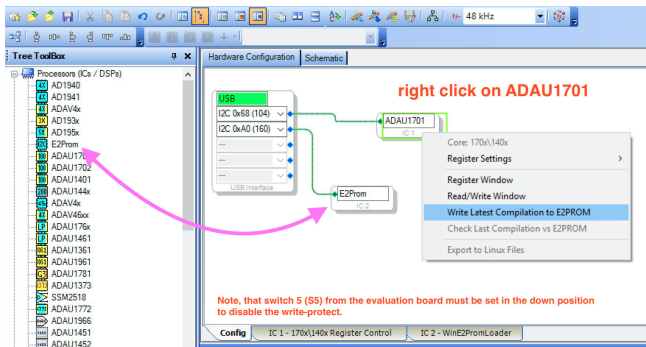
1) right click on the equalizer to open the menu

2) select one additional channel to make it a stereo equalizer



Work 2: Self-Boot E2Prom (make the DSP standalone)

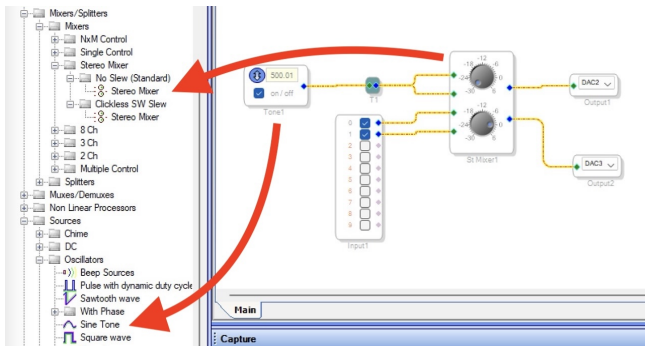
1. Write the compiled project's program data to the E2Prom to make work the board in standalone mode.



2. After writing to E2Prom reset the board (reset button) to verify if the program loads back.
3. Disconnect the board from the programmer and connect to power supply. Your program is now saved on the board (independent of the computer).

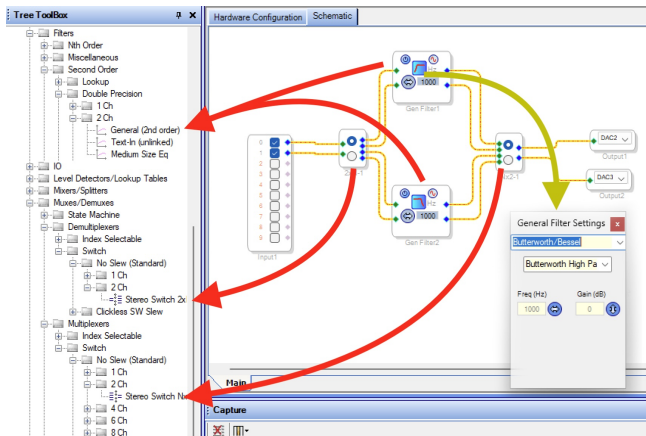
Work 3: Stereo Mixer

- ▶ Mix two audio signals
- ▶ - Input (external) audio
- ▶ - Sine wave



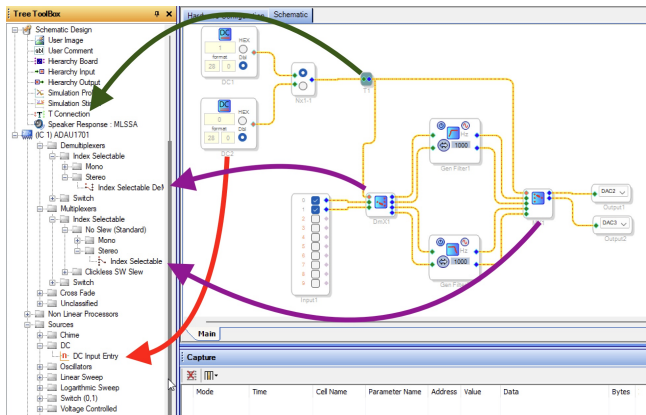
Work 4: Multiplexer and Demultiplexer

- ▶ The goal is to learn how to switch between paths
- ▶ Use two Butterworth filters, one high pass, one low pass.
- ▶ Switch between paths while listening.



Work 5: Synchronized Multiplexer and Demultiplexer

- ▶ The goal is to use just one switch (synchronize the mux. and demux.)
- ▶ Replace the audio switches by index selectable switches.
- ▶ Use just one mono switch to select values 0 or 1 (in 28.0 format).
- ▶ To generate 0 and 1, use DC sources (in 28.0 format).



Work 6: Button with Multiplexer and Demultiplexer

- ▶ Replace the software switch with the GPIO input connected to a button.
- ▶ Use Toggle On-Off block that toggles between zero and one each time a rising edge is detected on the input.
- ▶ Use Interface Read and Interface Write to save the values of the Toggle On-Off block.

