

130232 Audio DSP Board

Note: connectors & headers have been renamed to conform to Elektor style

Hardware

- DC power plug: center pin is GND.
- DC power: from 5 to 12 V
- ADC0 and ADC1 cinch connectors are analog audio inputs
- DAC0, DAC1, DAC2 and DAC3 cinch connectors are analog audio outputs.
- P1, P2, P3, P4 are analog inputs for the DSP. They deliver a voltage from 0V to a maximum level defined by trimmer P5.
- S1 is the Reset switch for the DSP.
- JP1: Place a jumper on JP1 to program the EEPROM. The DSP will not boot in this case. Remove JP1 and press S1 to make the DSP boot.
- K10 is for programming the EEPROM using a special programmer.
- K12 allows using the Elektor FTDI-BOB module 110553 to program the EEPROM using the Elektor application S.Studio to EEPROM Converter. This is rather slow.
- K9 is for programming the EEPROM using custom hardware like an Arduino board. Its pins accept 5V signal levels, and are protected by 3V3 zener diodes.
- K11 gives access to all GPIO pins of the DSP. Four of them (2, 3, 8, 9) are also used by the potentiometers P1-4.
- Trimmer P5 sets the maximum level for the potentiometers.

Software

- Software may become unresponsive during some operations, always wait before re-clicking.

Tutorial

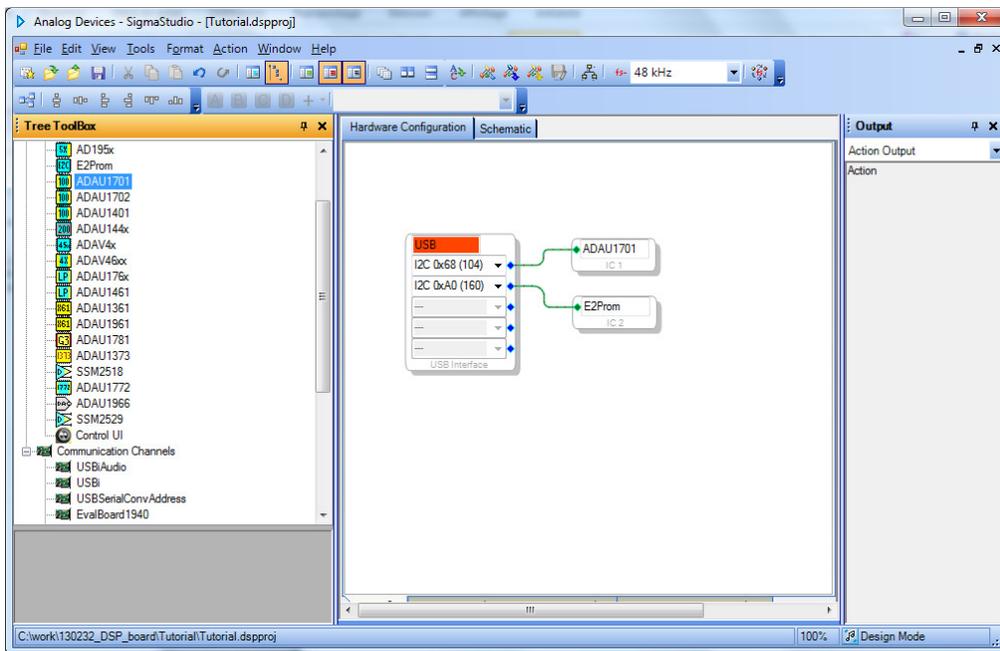
Prerequisites:

- Windows 7 x86/x64 or Windows Vista or Windows XP Professional or Home Edition with SP2
- at least one com port or a USB-to-serial adapter
- .net framework 4 + 3.5
- Sigma Studio 3.9 (download at : http://www.analog.com/en/dsp-software/ss_sigst_02/sw.html, free but account needed)

- Elektor S.Studio to EEPROM Converter utility or an Arduino Uno or Mega board or an I²C EEPROM programmer

1. Create a SigmaStudio project

- Start SigmaStudio
- Select “File” → “New Project” and wait for the project to be created. You will see elements in the toolbox on the left; drag and drop “ADAU1701”, “USBi” and “E2Prom” blocks to the main window named Hardware Configuration.
- Now link one of the blue dots of the block USBi to the green dot of the ADAU1701, link the other blue dot of USBi to the green dot of the E2Prom. The result should look like this:

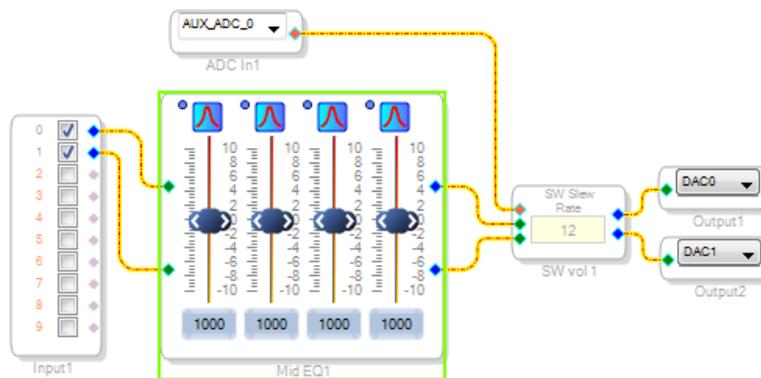


- Save the project (f.i. as “Tutorial”), in a folder of the same name.

Click on the “Schematic” tab. Notice the change of toolbox at the left.

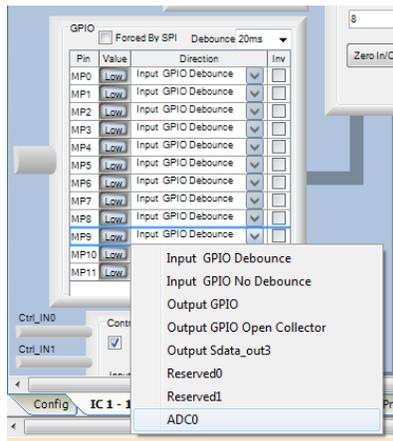
- Under IO → Input, drag and drop an “Input” element.
- Under IO → Output, drag and drop an “Output” element twice.
- Under GPIO → Input, drag and drop the “Auxiliary ADC Input” element.
- Under Volume Controls → Adjustable Gain → Ext Control → Clickless SW Slew, drag and drop the “Single slew ext vol”
- Right click on the “SW vol 1” block, select Grow Algorithm → 1. Ext vol (SW slew) → 1. Block “SW vol 1” should now have 3 inputs and 2 outputs.

- From Filters → Second order → Single precision → 2 Ch drag and drop “Medium Size Eq.” Right click on the “Mid EQ 1” block, select Grow Algorithm → 1. 2 Channel – Single Precision → 3. Block “Mid EQ 1” should now have 4 sliders.
- Connect the blocks according like this:



Now we need to configure GPIO9 as an auxiliary ADC input:

- Return to the “Hardware Configuration” tab and then click “IC1 -170x\140x Register Control” tab at the bottom of the window.
- Select MP9 In the GPIO block and change “Input GPIO Debounce” to “ADC0”, then return to “Config” tab:



Return to the Schematic and change the values of the slider frequencies in the equalizer and set the gain to your taste.

The project is ready, save it.

Press F7 or select from the menu Action → Link Compile Download. You will see a Comms error because you do not have the official programmer. At some point during the creation of the schematic you will see a message mentioning USB problems. You can safely ignore this message.

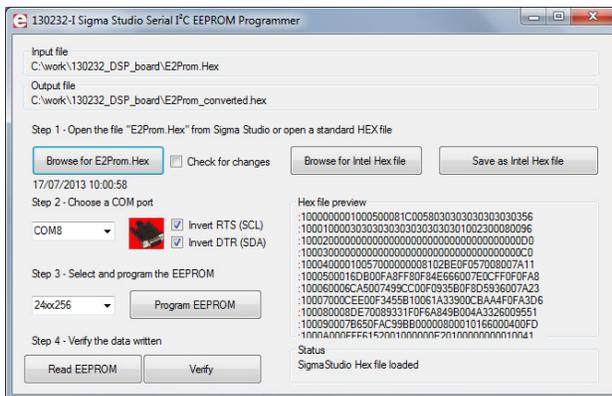
Return to the Hardware Configuration Tab, right click on IC1 and select “Write Latest Compilation to EEPROM”. This will create an executable file that we can program into the EEPROM.

2. Programming the EEPROM

Several possibilities exist:

- Programming over the serial port using the Elektor Sigma Studio Serial I²C EEPROM Programmer utility (slow)
- Use an Arduino board as a programmer (**recommended**)
- Use an EEPROM programmer

Ad a.



- Connect a stereo audio source to ADC0 and ADC1, and headphones or an amplifier to DAC0 and DAC1.
- Connect the DTR, RTS & GND lines of a serial port to K12 using either a real RS-232 port or a USB-to-serial converter like the Elektor BOB. Note that depending on the type of port you use you may need to invert or not the RTS and DTR signals. Checkboxes to do this are provided (default values are for BOB)
- Place a jumper on JP1
- Power the DSP board
- Launch the utility and open the file E2Prom.Hex that you created with SigmaStudio. A preview in Intel Hex format will be shown. You can also load a file in Intel Hex format if you have such a file
- Select the serial port in the utility
- Select the right EEPROM size and click the “Program EEPROM” button. Programming will start and may take several minutes depending on the size of the file. The slowness is due to the serial port driver.
- To verify if all went well first click “Read EEPROM” (this will again take a while), then click “Verify”.

- If all went well, remove the jumper from JP1 and press the Reset pushbutton on the DSP board. Verify that you can adjust output volume by turning P4.

Note 1 : This tool allows the conversion of an E2Prom.Hex file in proprietary Analog Devices format into a standard Intel Hex file that can be used with a commercial EEPROM programmer. Use the button “Save as Intel Hex File” to do this.

Note 2 : This tool can also be used to read an EEPROM and save its contents as an Intel Hex file. To do so start the tool without loading a Hex file. Then click “Read EEPROM” (this will again take a while), followed by a click on the button “Save as Intel Hex File”. BTW, it will read no more than 10000 bytes as the maximum possible ADAU1701 program size is about 9200 bytes.

Ad b.

To speed up programming we wrote an Arduino sketch that accepts a file in E2Prom format and that will program it into the EEPROM. To make things even better we added automatic control of jumper JP1 and the Reset pushbutton. Connector K9 allows you to connect the Arduino board. Here is the default pin-out. The sketch is of course available for download.

Signal	Arduino pin
SDA	10
SCL	11
WP	12
Reset	13
GND	GND

Use a serial port terminal program to send the E2Prom.Hex file to the Arduino board (option “Send file” in Tera Term). The baud rate is 19200 baud. Faster rates will not work well.

Ad c.

Refer to the manual of your programmer. You can use the Elektor utility mentioned under point a to create useable files. Connector K10 is available for direct connection to the EEPROM’s I²C pins.

More examples

A few more examples like a stereo chorus, short delay and a state variable filter with three outputs have been created and are available for download.