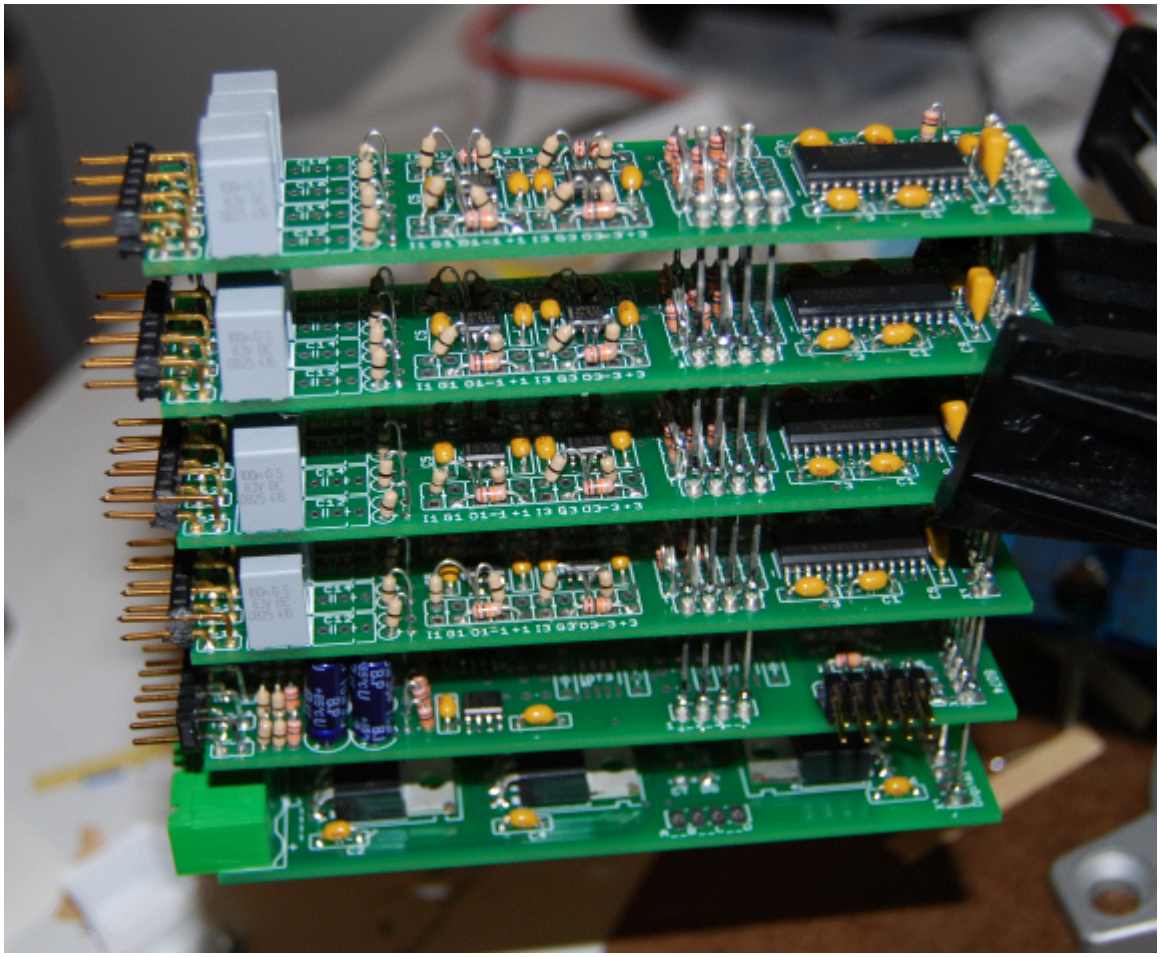
MBMixer Version 2



note: This page is an edit in progress. At this time the change to version 2 is near complete, but I may have missed a few details.

Copyright (C) 2007,2008 Lyle Hazelwood (lylehaze@gmail.com) Licensed for personal non-commercial use only. All other rights reserved.

More Information and
[[http://debtreliefcounselingus.com/credit-card-debt-reduction-strategy.html|credits Here]]

Objectives

Drives an audio mixer built with PGA2311 or PGA4311 volume controls.
Project Purpose and Intent

Description

This project provides a digitally controlled analog audio mixer. Using either an encoder or MIDI control messages, the user may control the volume and balance of attached audio sources. (Ver 2) Now

supports balanced or unbalanced inputs and outputs, optional effects loops, and a wider set of MIDI commands.

Hardware

This project is built around the MIDIBox core module, which is a PIC18F452 based MIDI device. Additionally, a DINX1 may be used (optional) to connect an encoder and pushbutton. A display module may be connected. The one I designed for is 2 lines X 20 characters, but it's not difficult to change. Some quantity of PGA4311 chips are needed to control the audio levels. The 4311 chips support four audio channels. This is two pairs for line mixer, or one input into L,R, FX1, and FX2 for a full mixer). A NE5532 dual op-amp based summing circuit is used to combine the audio sources, and NE5532 based input buffers may also be used to assure that the inputs have a sufficiently low impedance. Finally, the regulator board provides two +5 volt and one -5 volt regulators to supply the PGA chips and audio circuits.

Additional Hardware Info: MIDIboxMixerHardware

The circuit board designs have been provided to SmashTV, and will be available in his web shop. If you are interested, please let him know!

Software

The entire project is designed within MIOS, which is a fantastic open sourced OS for embedded MIDI applications that is written and maintained by Thorsten Klose. The actual code to control the PGA chips was based on code provided by Pilo, who provided it after chatting with him at www.midibox.org/forum/index The MBMixer application can handle up to 16 channels, which may be any combination of Line and Audio types.

The reference design has only a single encoder with pushbutton. The user may select the channel by pressing the button, then turn the encoder to adjust the level. any changes made with the local encoder are also sent out the MIDI out port, so that the control deck can stay synchronized with the actual level.

Software Information: MIDIboxMixerSoftware

Operators Manual

An Operators manual for the MBMixer: MIDIboxMixerManual

Some old Version 1 pics

Version1Pictures

Expansion Limits

How big can it get? How fast does it go? It seems that some people want a LOT of PGA chips chained together. After a discussion in the forums a few days back, I thought I'd put all the expansion notes in one place.

Hardware: The V2 boards can be stacked as deep as you want. The only signal issue will be “fan-out”. The active control lines going into the stack of channel boards are : SCLK, CS, and SDI. The static signals are MUTE and ZCEN, which will probably never move. SDI goes only to the first channel board. That boards SDO goes to the next boards SDI (daisy chained) so no problem with fan-out there. CS and SCLK connect in parallel to ALL the chips in the stack. This is where we will need help. Driving four channel boards should be no problem. Driving eight will probably work, driving sixteen will likely not work, as the signals just aren't strong enough to drive 16 loads at once. The fix? Use some kind of driver chip, buffer, or even a few transistors to make those signals stronger. Because my needs are not as grand, and my budget isn't either, I'm not going to research this. I'll leave that to people who can afford that many chips.  Please post your findings! The power supply board at the bottom of the stack should have plenty of power for channel boards, as long as you don't power your core or display from it. [EDIT] after more investigation, fan out may not be much of a problem at all. I invite the first people who stack deeper than eight to get with me when they do, I beleve we can do this without additional buffering.

Software: The software I provide assumes a 16 channel maximum. That's how MIDI works. I'm using CC7 for Volume, 11 for Expression, 12 for Effects1 and 13 for Effects2. If you want to support more than 16 channels, you'll have to figure out something different and re-write the USER_MPROC_NotifyReceivedEvent stuff. You are welcome to support as many channels as you want. The output table for PGA data (CHANNEL_GAIN) is sized for 64 gain settings. This is the max you can use for 16 channels. (16 PGA4311 chips, Left, Right, FX1, and FX2 per chip). All of the code for MBMixer2 is written in ASM. The code is heavily commented, so you C types should be able to figure it out.

Latency: How long does it take to update the chips? Because of the way the chips are connected, making a change to ANY of the gains means updating the data to all of them. It's reasonable to wonder how many chips you can add before it takes “too long”. recall that 1 second = 1000 ms = 1000000 us. Wikipedia says that 100ms will feel “instantaneous” for most apps. Humans are easy. What about MIOS? Each MIDI byte coming in to MIOS takes about 256 us. If we are gone for longer than that, we may cause MIOS to miss incoming data. Assuming a “full board” of 16 PGA4311 chips, and a 40Mhz FOsc, we can write all 16 chips in about 512 us, or a half-millisecond. We got around this by asking MIOS not to scan DINs or DOUTs while we are writing to the PGA chips, since some of the DIN/DOUT connections are shared with the MBMixer pins. This works great. But please understand, we are sacrificing DIN/DOUT update speed when we are writing to the PGA chips, and that sacrifice will grow with the size of our PGA banks. Just another reason to put the Control Surface on another core.

Downloads

MB Mixer Version 2 source code
MIDI Chart
Config Mixer source code
Parts list from DigiKey

BCR2000 Configuration File
Logarithmic curve spreadsheet
Windows ConfigMixer executable
Linux ConfigMixer executable
OS4 ConfigMixer executable

From:
https://www.midibox.org/dokuwiki/ - **MIDIbox**

Permanent link:
https://www.midibox.org/dokuwiki/doku.php?id=midiboxmixer&rev=1322501889

Last update: **2011/11/28 17:38**