

# Table of Contents

<b>Using a Polycom Soundstructure as a Digital Audio Mixer</b> .....	2
<b>Introduction</b> .....	2
<b>The motivation</b> .....	2
<b>The Capabilities</b> .....	2
Input signal chain .....	4
Matrix setup .....	4
Output chain .....	5
<b>Poking Around</b> .....	5
<b>Using SoundStructure Studio</b> .....	7
Using as a sound mixer .....	9
<b>Controlling Externally</b> .....	12
<b>Hardware Adaptations</b> .....	12
<b>Downloads</b> .....	13

# Using a Polycom Soundstructure as a Digital Audio Mixer

## Introduction

So I found that you can get your hands on a “teleconference processor” such as the Polycom Soundstructure C16 for surprisingly cheap. I only happen to have one since a seller threw it in when I was buying a Crestron DMPS-300 for cheap to use as an HDMI video switcher...

I just assumed it was going to be some proprietary nonsense only usable for Polycom VoIP equipment, and so I let it sit for a while. Until today after dinner where the thought randomly occurred to me that - hey wait a second - this thing has 16 inputs and 16 outputs, right?

I wonder if those inputs have preamps that would let me hook up mics? If so, it probably has an EQ and given it's bound to be digital, maybe has flexible bands?

## The motivation

You see, an “entry” tier digital audio mixers like a Behringer XR18 (or smaller cousins XR16 or XR12) are more flexible than their analog counterparts, but come with a steep price tag.

I didn't have to look far at all since the documentation for this thing can be found shockingly easily - like yeah you can find the Crestron tools on very shady FTP sites, but these were just public. I'll attach the things I found later on in this page.

So anyway, I started digging into it - and it turns out I severely underestimated the capabilities of this thing, and just how open and easy the control interface is, it's shockingly flexible.

## The Capabilities

Take a look at this:

	SoundStructure	
	C-Series	SR-Series
<b>Input Processing</b>		
Up to 8th order highpass and lowpass	✓	✓
1st or 2nd order high shelf and low shelf	✓	✓
10-band parametric equalization	✓	✓
Acoustic echo cancellation, 20-22kHz 200 msec tail-time, monaural or stereo	✓	✓
Automatic gain control: +15 to -15dB	✓	✓
Dynamics processing: gate, expander, compressor, limiter, peak limiter	✓	✓
Feedback Eliminator: 10 adaptive filters	✓	✓
Noise cancellation: 0-20dB noise reduction	✓	✓
Automixer: gain sharing or gated mixer	✓	✓
Signal fader gain: +20 to -100 dB	✓	✓
Signal delay to 1000 msec	✓	✓
<b>Output Processing</b>		
1st or 2nd order high shelf and low shelf filters	✓	✓
10-bands of parametric or 31-band graphic equalizer	✓	✓
Dynamics processing: gate, expander, compressor, limiter, peak limiter	✓	✓
Signal fader gain: +20 to -100 dB	✓	✓
Cross over equalization up to 8th order highpass and lowpass filters, 1st order horn equalization	✓	✓
Crossover delay: up to 100 msec	✓	✓
Signal delay: up to 1000 msec	✓	✓
<b>Submix Processing</b>		
Up to 8th order highpass and lowpass filters	✓	✓
1st or 2nd order high shelf and low shelf filters	✓	✓
10-bands of parametric equalization	✓	✓
Dynamics processing: gate, expander, compressor, limiter, peak limiter	✓	✓
Signal fader gain: +20 to -100 dB	✓	✓
Signal delay: up to 1000 msec	✓	✓
<b>Telco Processing</b>		
Line echo cancellation, 80-3300Hz, 32msec tail-time	✓	✓
Dynamics processing: gate, expander, compressor, limiter, peak limiter on telco transmit and receive	✓	✓
Up to 8th order highpass and lowpass filters	✓	✓
1st or 2nd order high shelf and low shelf filters	✓	✓
10-bands of parametric equalization on telco transmit and receive	✓	✓
Call progress detection	✓	✓
Signal fader gain: +20 to -100 dB	✓	✓
Automatic gain control: +15 to -15dB on telco receive	✓	✓
Signal delay on telco transmit and receive: up to 1000 msec	✓	✓
Noise cancellation: 0-20dB noise reduction on telco receive	✓	✓

- You've 10 band parametric EQ on the inputs
- You've got gating, compression and limiters
- AEC that we can disable and never touch
- Feedback elimination - need to see if this is any good (the Behringer FeedbackDestroyer is AKA the flute destroyer)
- Auto gain and auto mix can be turned off, I need to try it and see if they are any use

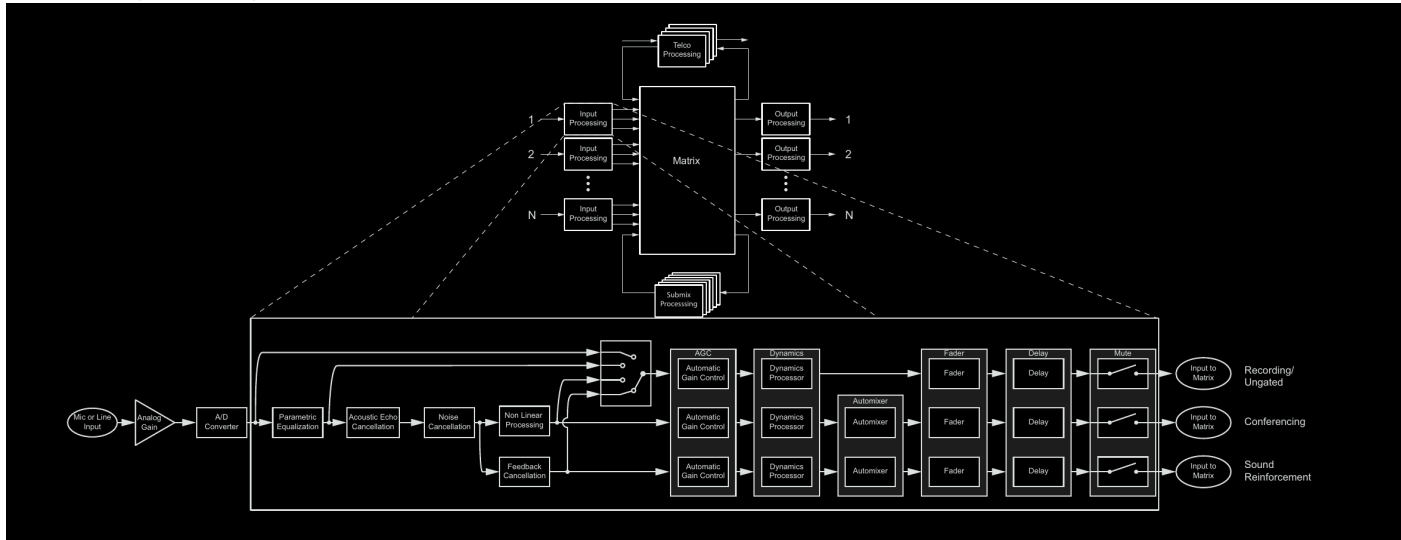
Things not on here

- 60 dB of analog gain headroom before the DAC from what I can tell
- There is per-port phantom power on all 16 inputs
- Each input is balanced or unbalanced for mic or line
- Each input has the same signal chain

Also apparently you can buy up to three of these and chain them via firewire, to use any input across the units with any output. Seems easy way to expand to 32 ins and 32 outs.

# Input signal chain

Here is a good diagram from the PDF:



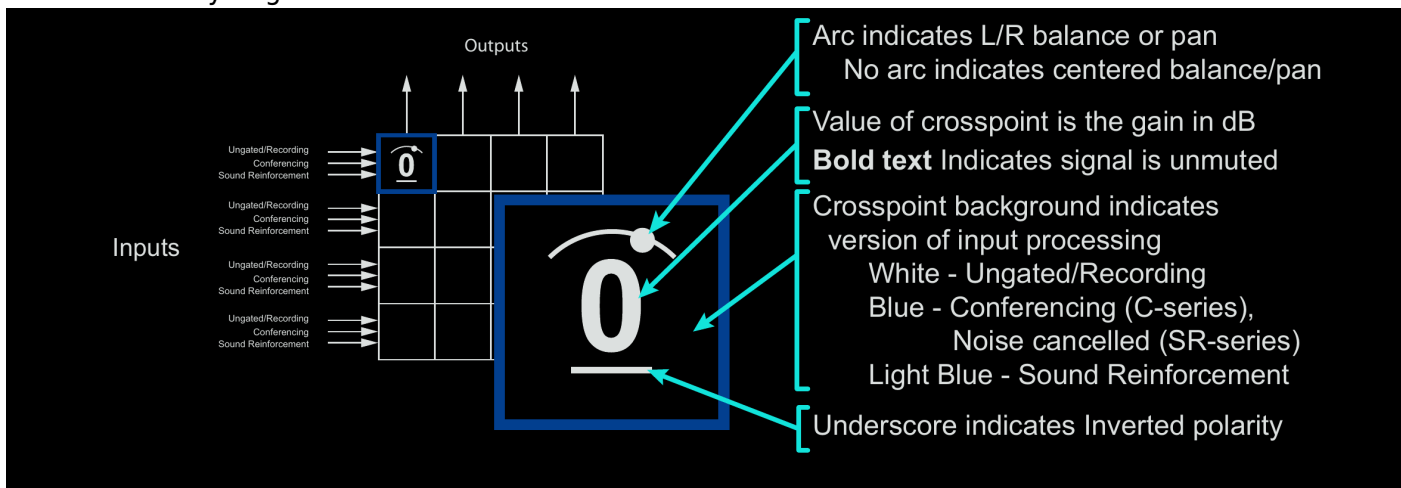
This shows the exact order of operations.

The most interesting thing on here is the mux that produces the recording, conference and reinforcement outputs, since this is something different than a normal digital mixer.

- Each signal is split, and when setting up the in-out matrix, you can pick which of these three feeds to give each output.
- The ungated is useful since you can grab it post-EQ but before any “conference” specialty things
- I'm also testing the sound reinforcement out, since you can enable/disable any of the steps in this chain on the fly.

## Matrix setup

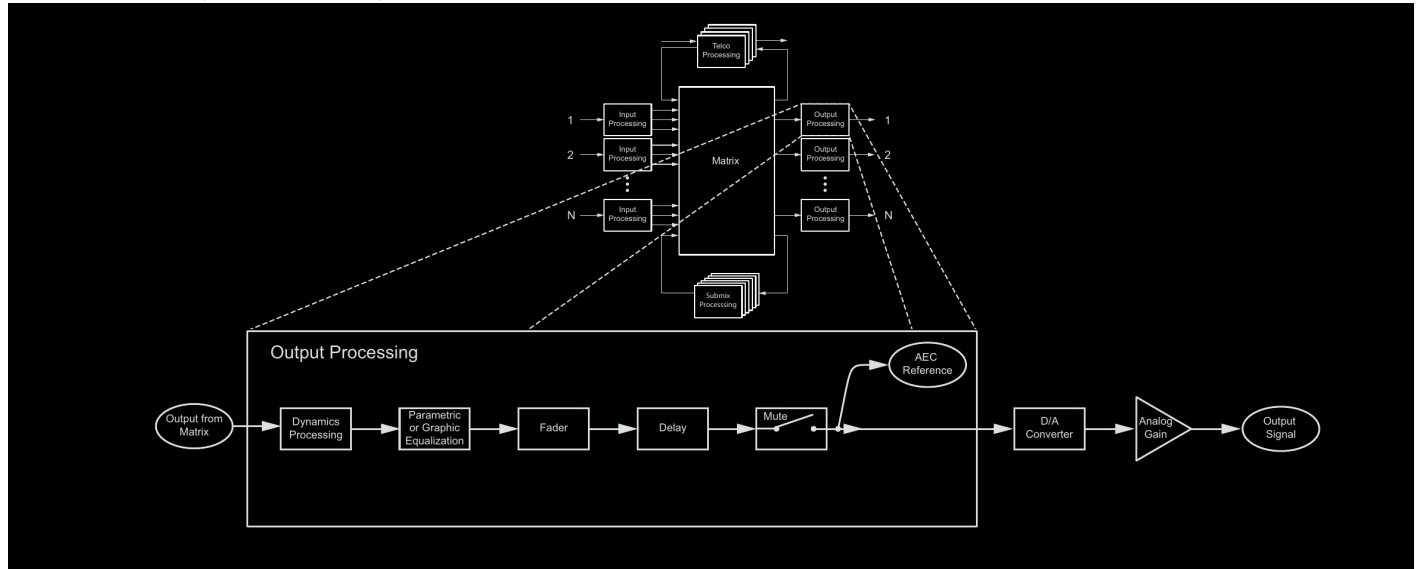
Here is control you get:



On each output, you can pick levels of each input, pan (if stereo out), and which point to get.

## Output chain

Here is the steps on the output:



- Each output has its own EQ and fader, which means beyond having a master out, this can be used for all the monitor outputs while giving a custom EQ to each.
- You can also just make a direct out, give source audio for say an FX SEND.

## Poking Around

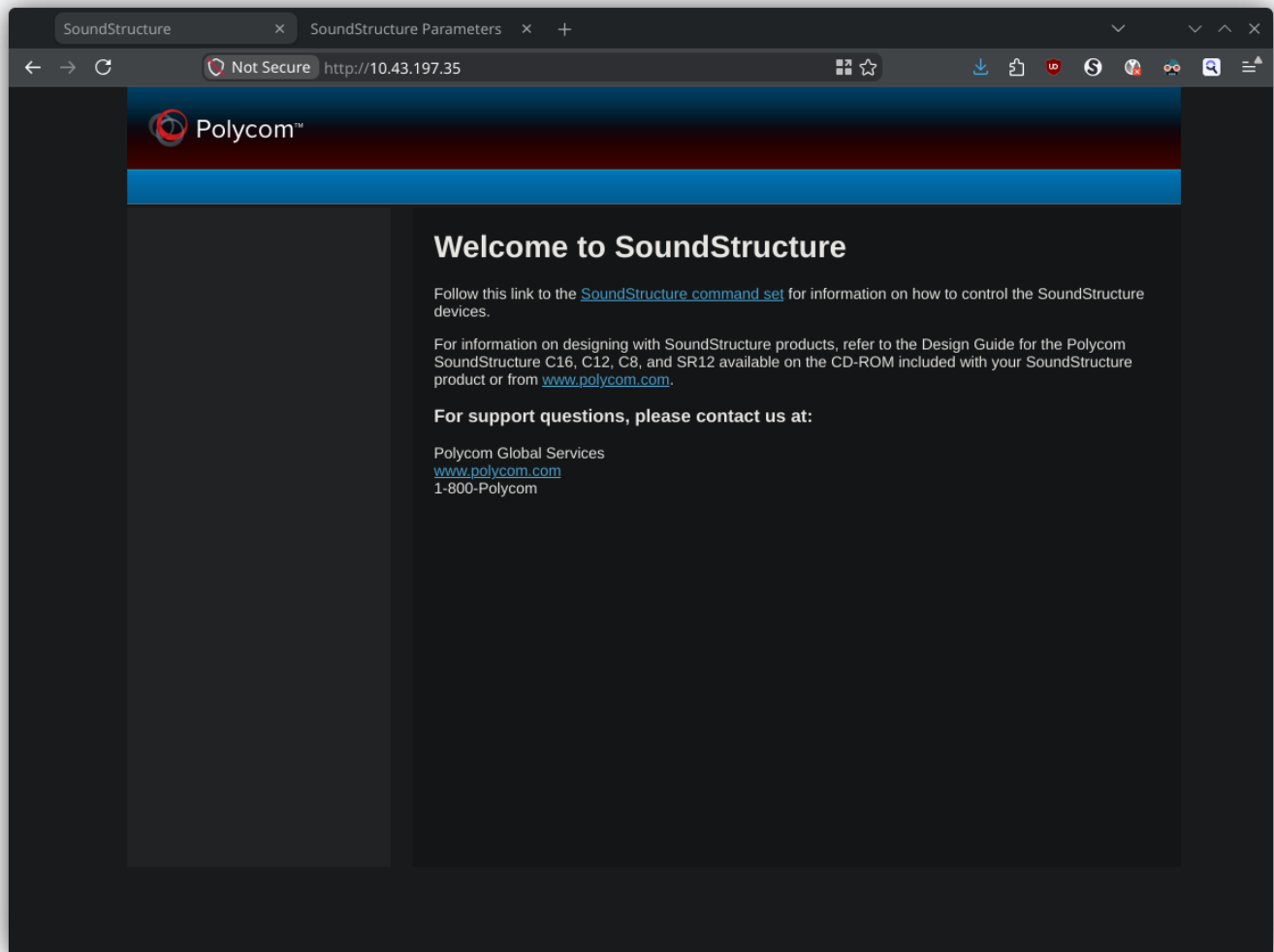
My first step was figure out what the IP address was on my unit - easy to wireshark and wait for a packet. Then I checked what ports exist:

```

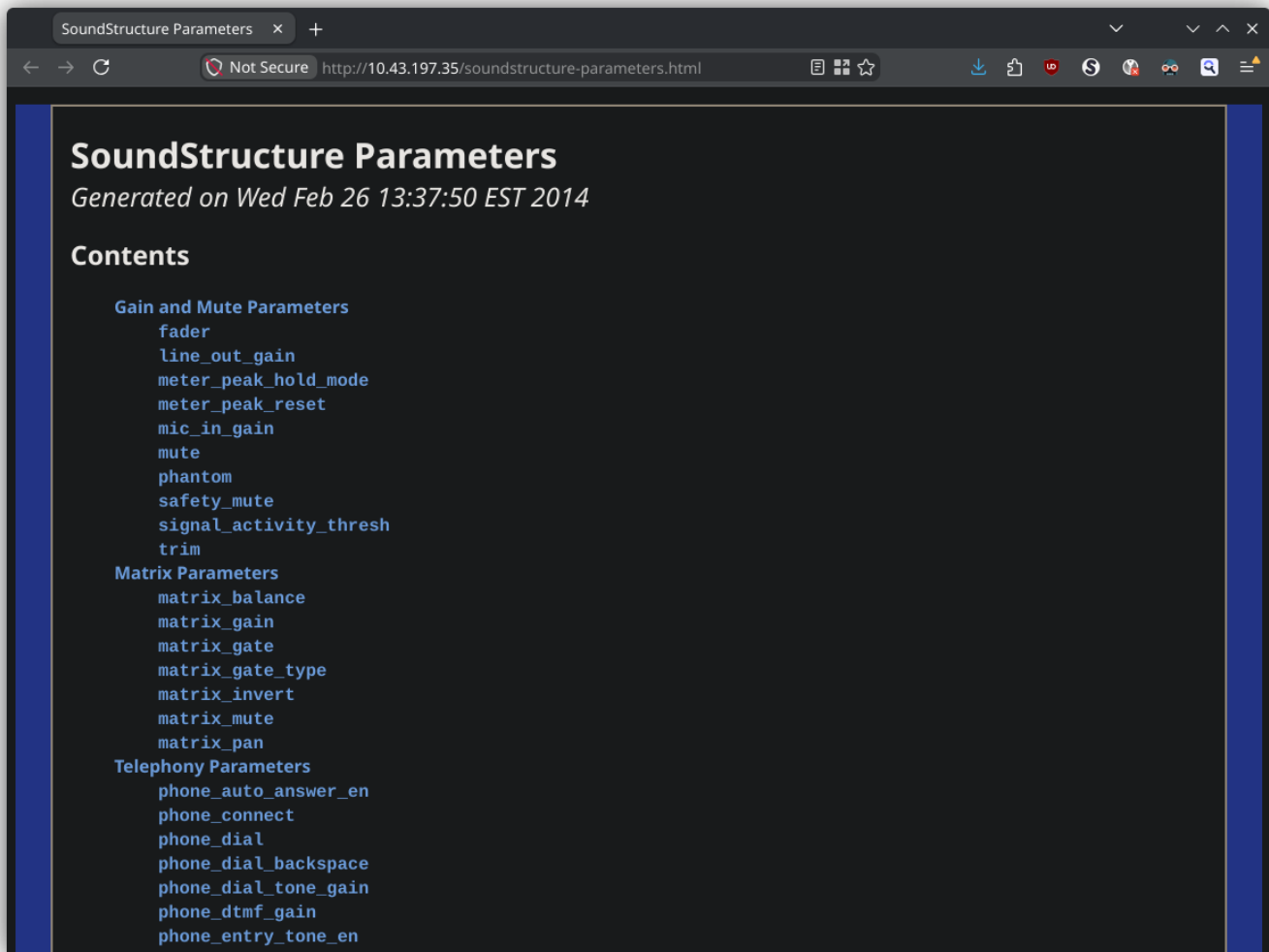
PORT      STATE SERVICE
21/tcp    open  ftp
23/tcp    open  telnet
79/tcp    open  finger
80/tcp    open  http
111/tcp   open  rpcbind
113/tcp   open  ident
513/tcp   open  login
514/tcp   open  shell

```

I figured I may as well go to the HTTP port and we get a little welcome page:



Amazingly, if you click the link, it goes to a local copy of all the commands and parameters!



By the way all the commands and parameters listed here are also available in the PDF.

Through this process I'm kind-of amazed at how simple it is to use? Like, I'm so used to having to reverse-engineer the command set or hack something together to be able to use hardware for unintended purposes, that just giving me the API feels wrong almost.

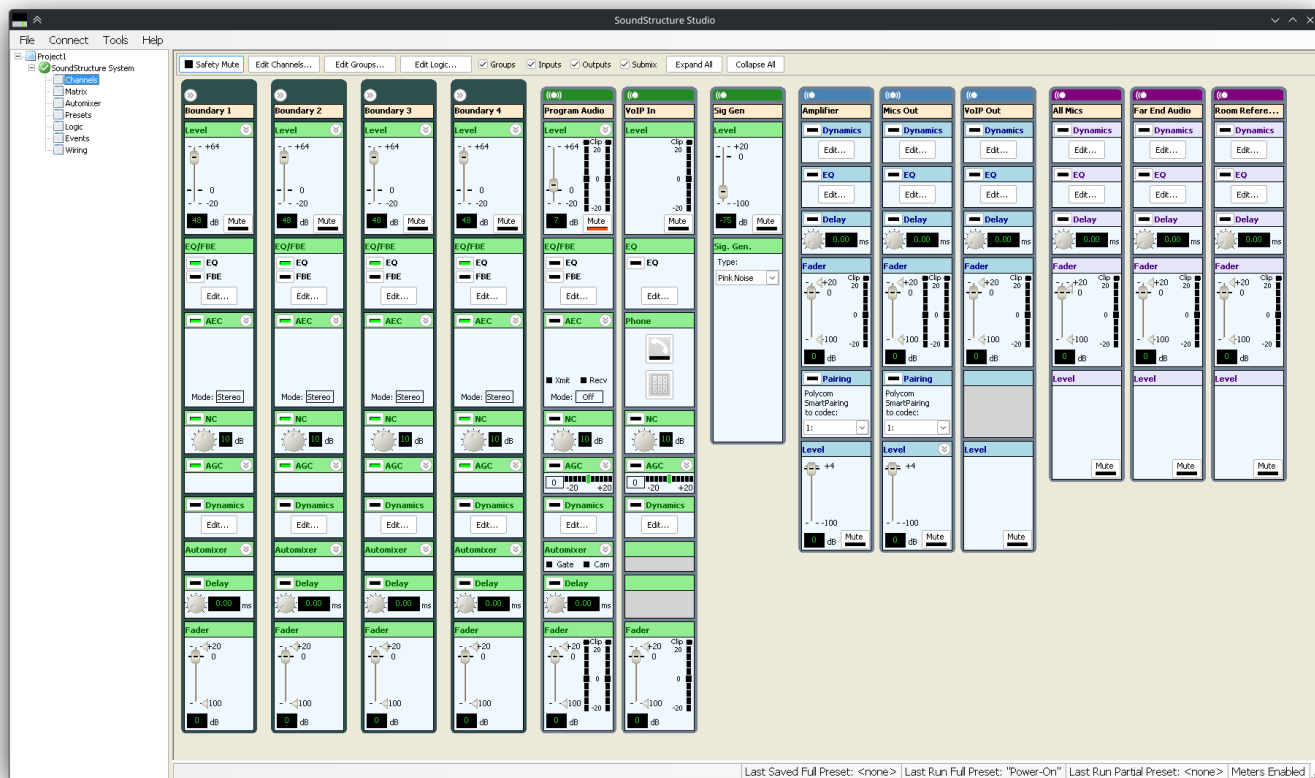
Anyway, if you look through the list, there's controls for basically everything you then see in their software. The one bit I don't know yet is presets. If presets work, than this becomes even more useful as I can bake presets for each group performing, and jump between them.

## Using SoundStructure Studio

I was able to find the software on the HP website, and it didn't require any login which is again a nice change for once. It works just fine in WINE on Linux, and was able to access the SoundStructure over the LAN connection.

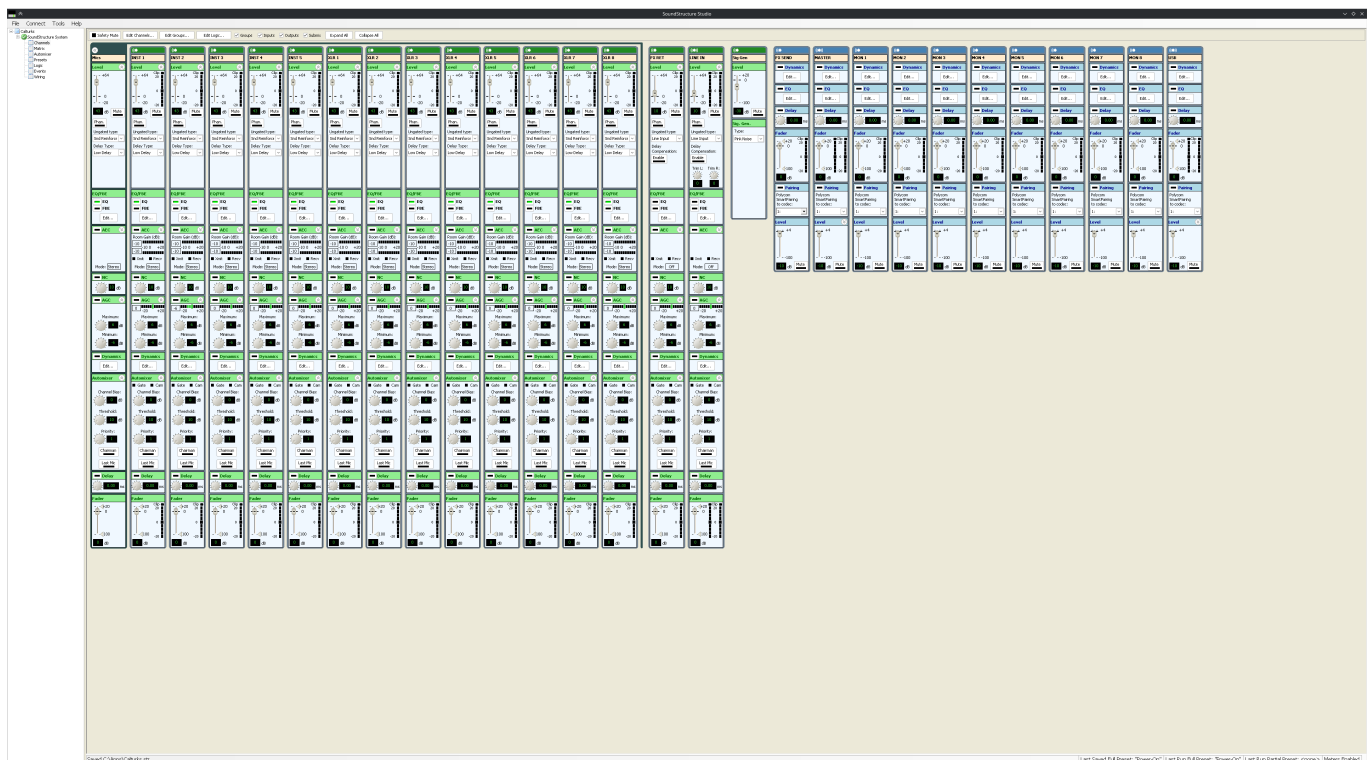
I may as well point out you can use LAN over their control interface, OR you can telnet, OR you can use the serial port. Both the SW and you can control it either way.

The software is *\_fine\_*, it's not the best UI but they have some presets, and the UI is actually usable. Here is the default preset they start you with:



It's both similar and different than say the XR18 UI.

So I of course immediately wanted to throw that out and test how far I can push it.



and here you go!

- 13 channels of mic/instrument inputs
- 1 channel of effect return
- 2 channels of line in (stereo)
- 1 main output
- 1 effect send
- 8 monitor outputs
- 1 stereo pair for a USB interface (and we're not using up all the outputs yet)
- You can make subgroups as you see fit (I think up to 16) so you can do mics, instrument, drums like you would on an actual sound board

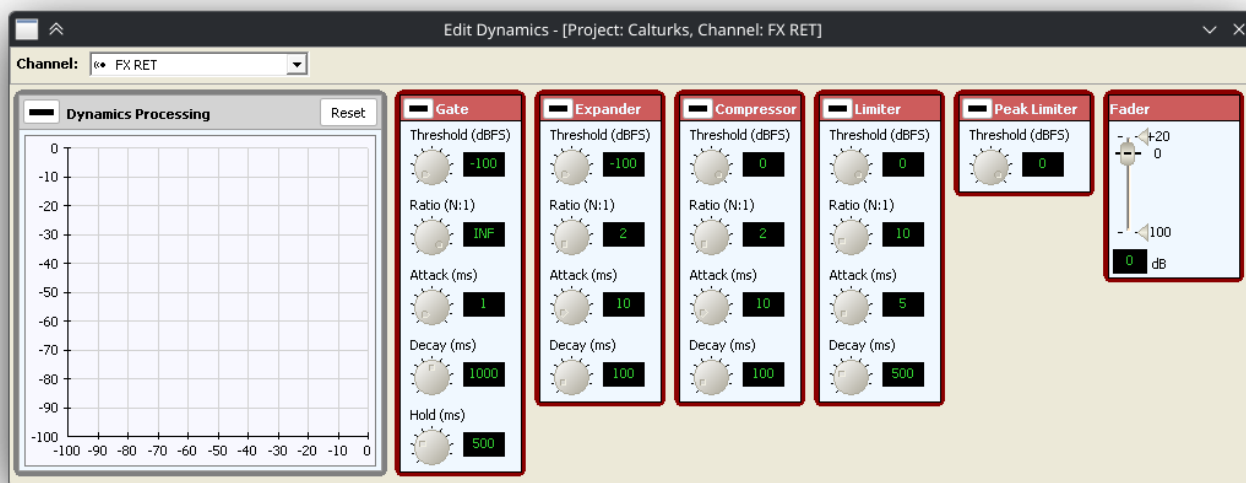
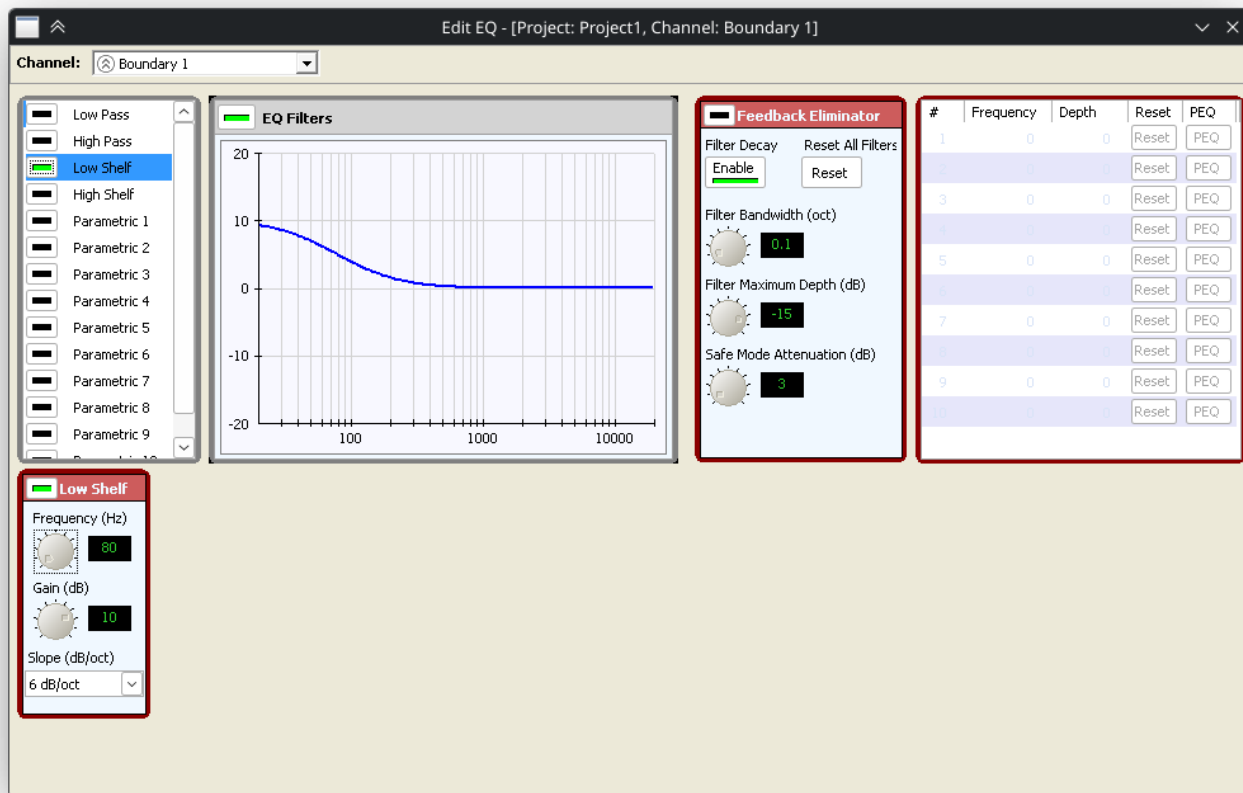
## Using as a sound mixer

- It's also worth noting this channel assignment and settings are saved to EEPROM so you don't have to set up your channel configuration, you can start with this, then just use the commands to jump between the settings on these. (so you could implement "presets" just at a higher level of abstraction)
- Similarly a way for better external control might be make all inputs/outputs here uniform, then control them all via the commands instead of using this software (since eg channel names are used in the API, so maybe name in/out, then have your higher level external program name them based on a json for example).

\* On each input here, you can see the gain, mute, phantom power, then which tap-point from the signal chain to use. Then EQ and feedback suppression, then echo cancel, noise cancel, gain control, dynamics,

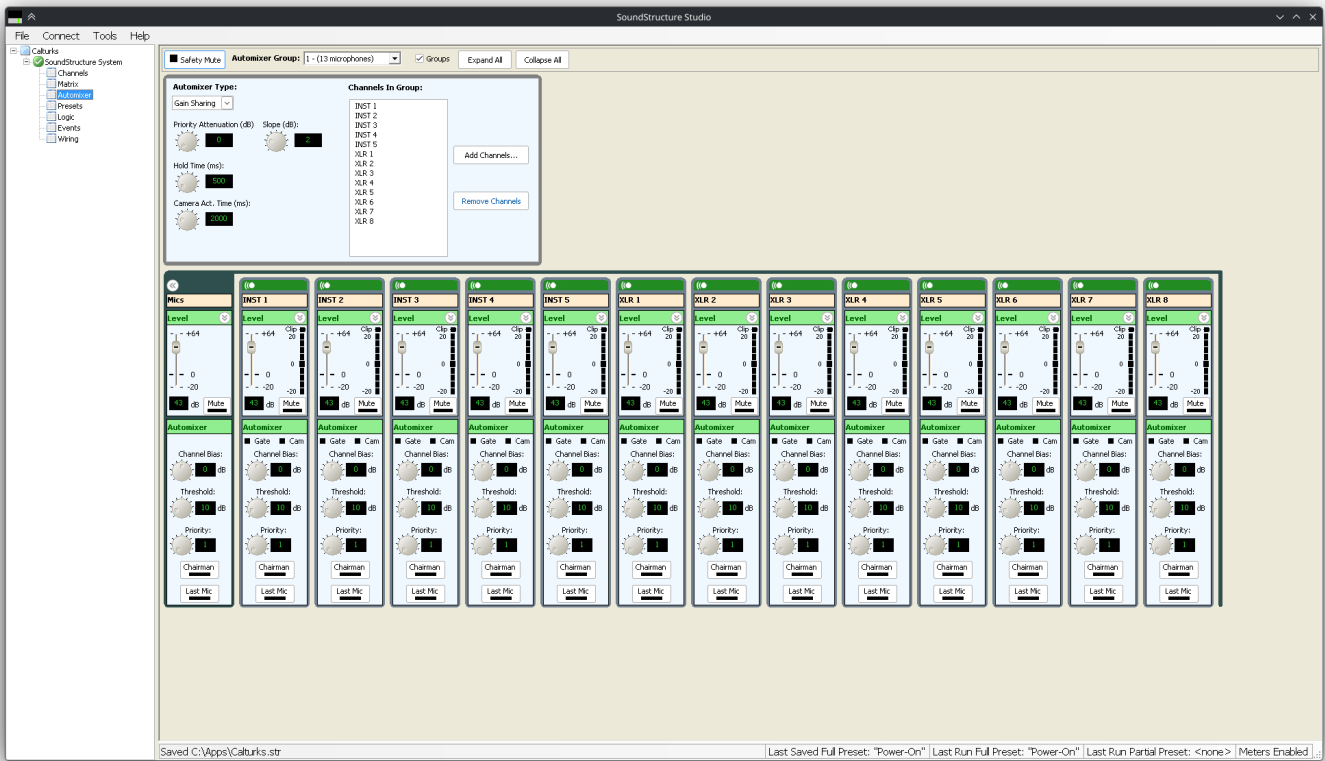
automix, delay and fader.

\* I took a look at what the options on the EQ and dynamics are, and it's not bad!



\* here is what you get in automix, seems they're just doing a way for you to keep all channels open and it

picks them as it needs. maybe not for live music but for say a panel this might be useful



\* finally here is the matrix configuration. again, you get level, pan and the tap-point to use.



## Side note

- One thing I'm used to seeing of course is that the fader on the input would really be the control for the main bus + fx (which are post-fader), but not the monitors, which are pre-fader
- On this "board", main is treated like any other input, so it has a fader in this matrix, that is after the fader in the input.
- and by that logic, all the monitor outs are post-fader, after the fader on the input.

This lends itself to kinda two modes of operation:

- Keep the input fader at 0, then control the actual mix level from input to everywhere from the matrix
  - You then use the matrix for the monitor as well, but it's on you to match FX SEND level to main
  - To get away with this, you almost need to implement external control so you can bring the fader dials for all the sends on the same page as say the EQ control
- You set and forget the matrix to main at 0, use the fader on the input channel
  - This is easier in this native program, but makes your monitors post fader

## Controlling Externally

- To build external control, is somewhat re-implementing their software and I haven't figured out if I take it on.
- I figure I can make a rust main server that talk to this, then make my UI with egui or kotlin to use a tablet (or web to serve both...)
- We can re-invent a lot of what their software has, our own form or presets, heck we could monitor levels and do our own adjustments even
- Same with adjusting if things are mic vs line, so on. The easiest way might be telnet, ie, open a socket to port 23 and make commands
- My main issue is going to be queueing commands and keeping the UI synced... And no I don't want to vibe code this

## Hardware Adaptations

- Really this is the other big issue, this uses terminal connectors for permanent install, not XLR or 1/4".
- You can buy an XLR patch panel but then you've spent more money on that than this board.
- Then you need to solder like 32 XLR connectors... and wire them to screw terminals
- (holup I'm starting to see how Behringer can charge \$300 now)

Still might be a fun weekend project though :)

## Downloads

Get the SoundStructure Studio software from Polycom directly here:

<https://downloads.polycom.com/voice/SoundStructure/studiosetup-1-9-1.zip> Get the design guide PDF

and software manual (where the signal chain charts came from) here:

[https://polycom-moscow.ru/pdf/SoundStructure\\_Design\\_Guide.pdf](https://polycom-moscow.ru/pdf/SoundStructure_Design_Guide.pdf)

From:

<https://wiki.tonytascioglu.com/> - **Tony Tascioglu Wiki**

Permanent link:

[https://wiki.tonytascioglu.com/articles/using\\_a\\_polycom\\_soundstructure\\_as\\_an\\_audio\\_mixer?rev=1776928661](https://wiki.tonytascioglu.com/articles/using_a_polycom_soundstructure_as_an_audio_mixer?rev=1776928661)

Last update: **2026-04-23 07:17**

