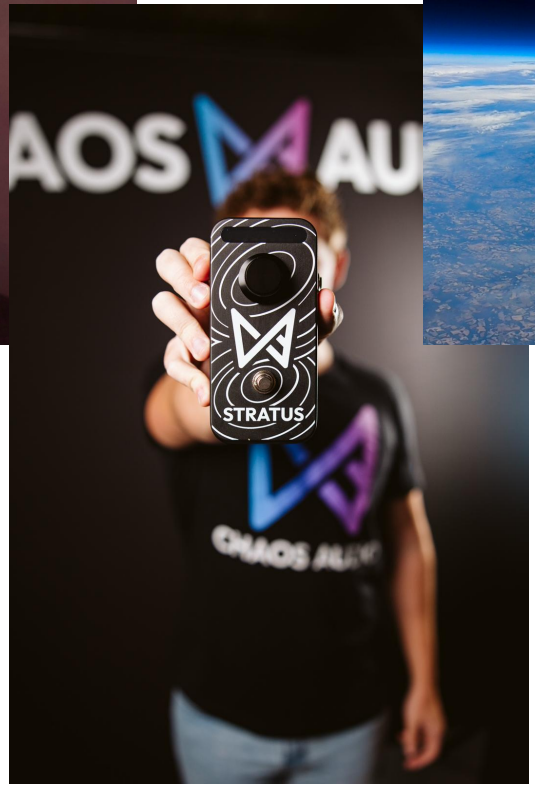
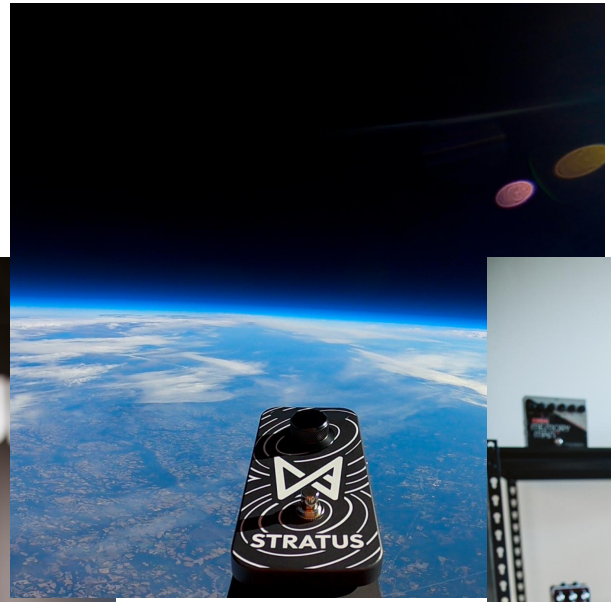
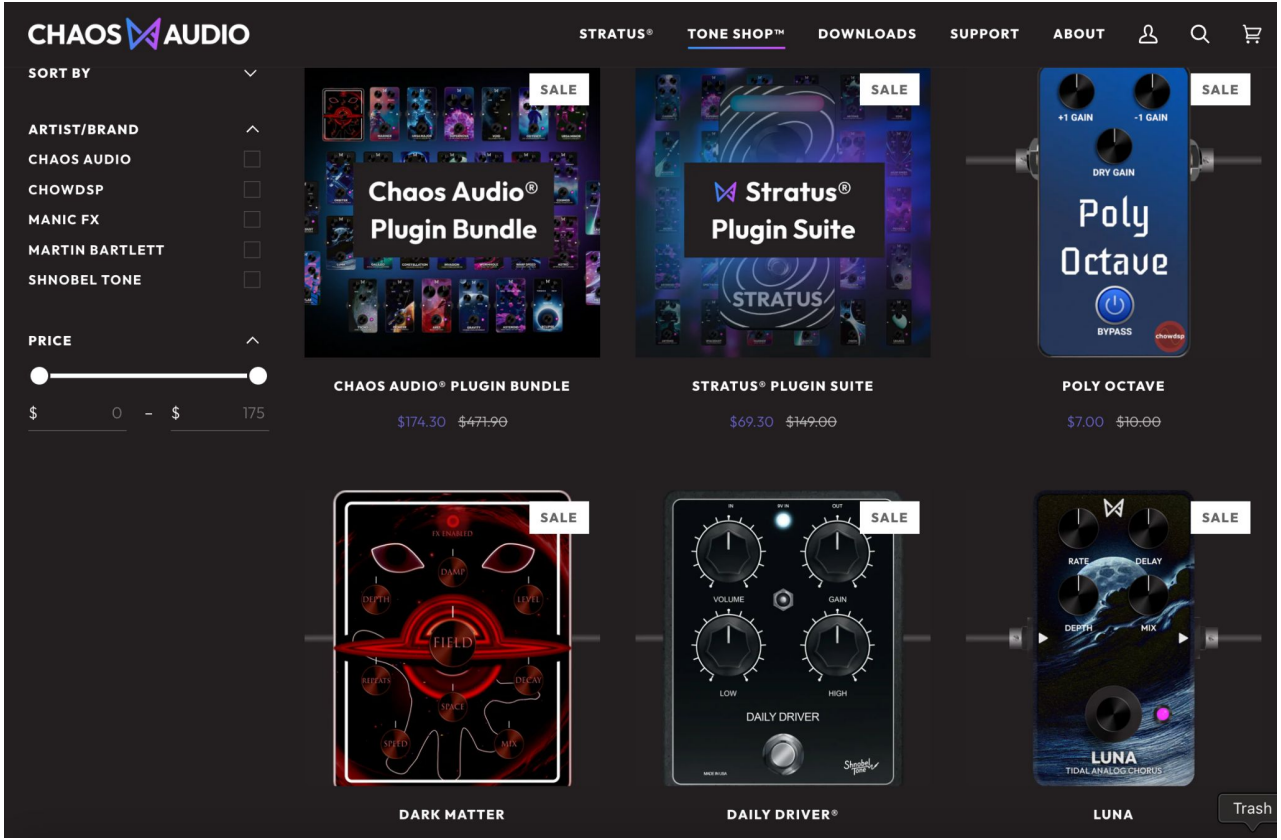
FAUST *and* CHAOS AUDIO

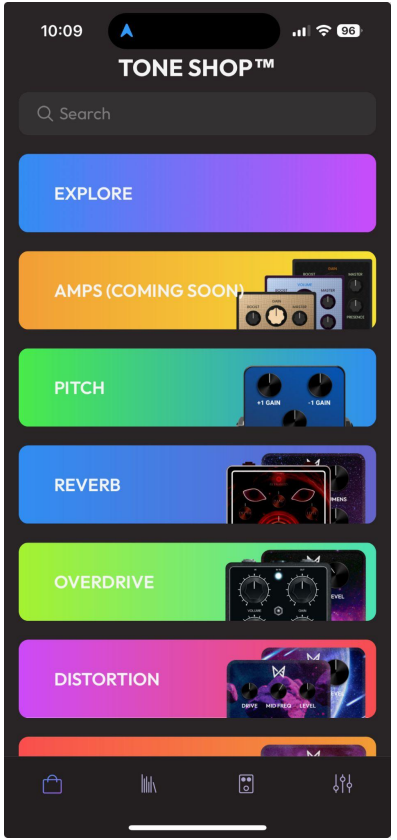
(Sounds a little frightening!)


STRATUS


STRATUS

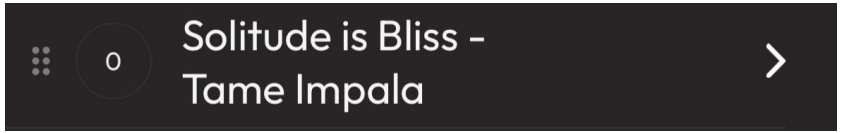

STRATUS





Anyone can develop for Tone Shop; cross-platform between Stratus and PC

(DEMO) Chain of plugins, all created using Faust:



Before 2024:

FAUST

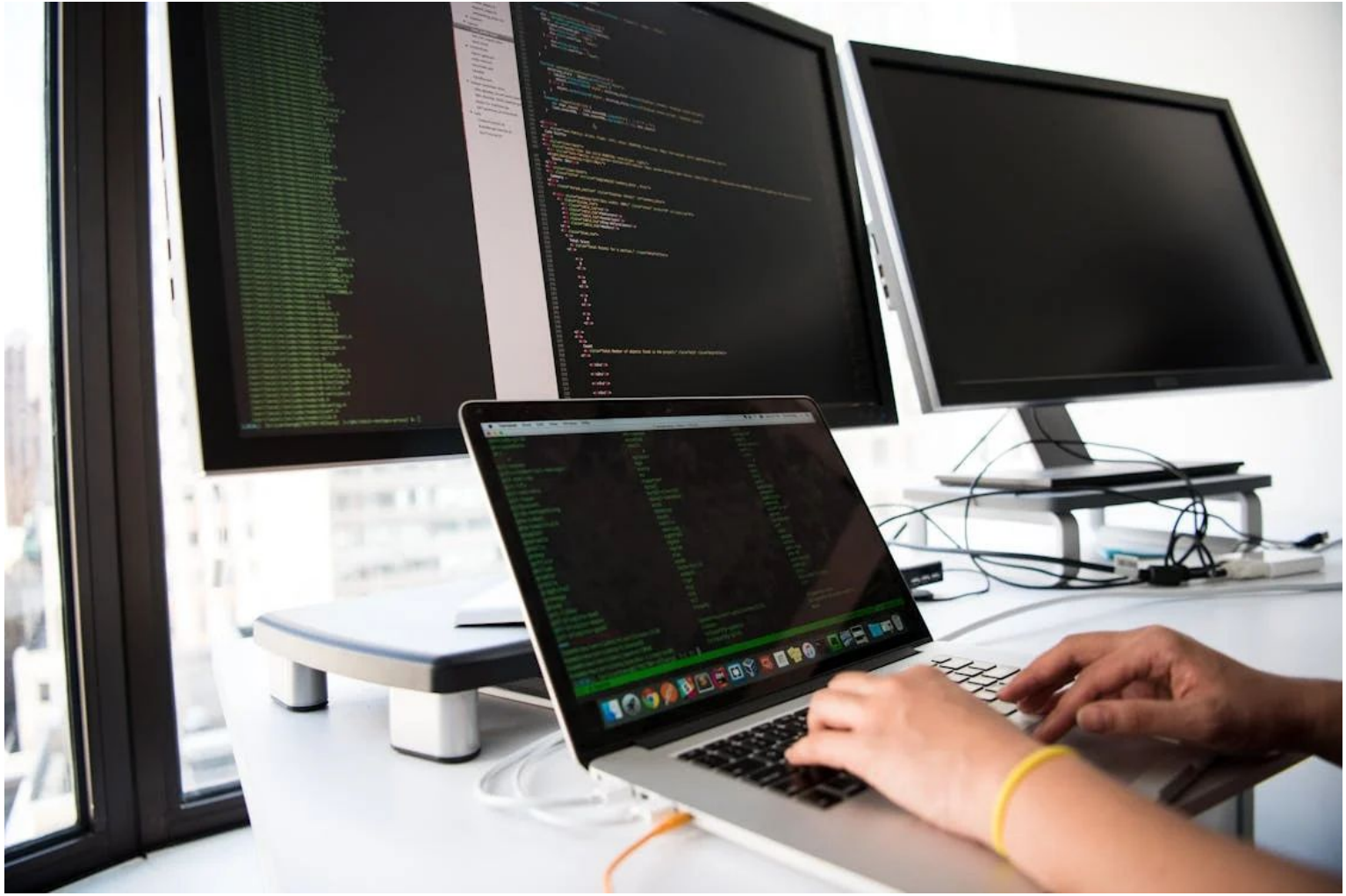
Generated C++ interface

Hacky manual tweaks

C++ interface for Stratus plugins


WORLD'S
OKAYEST
BASS PLAYER


I PLAY THE
BASS
BECAUSE
I LIKE IT
NOT BECAUSE
I'M GOOD AT IT

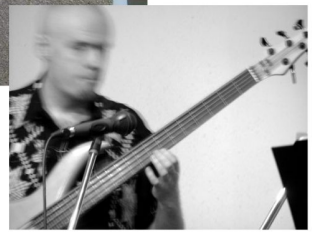
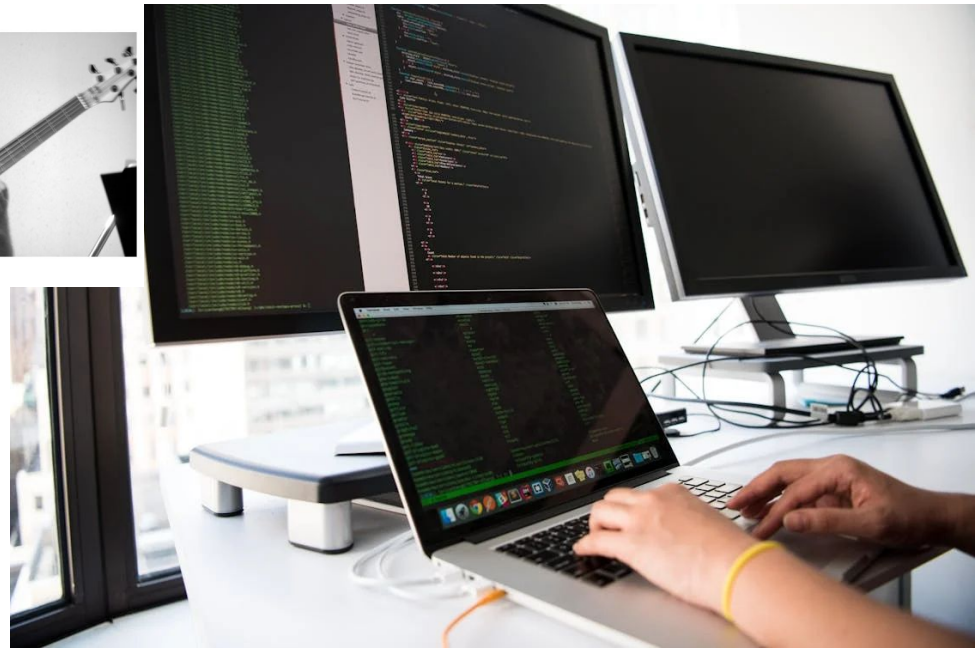



DD-20 DIGITAL DELAY
TIME
FEEDBACK
DELAY LEVEL
LOOPER LEVEL
LOOPER
DELAY
Record
Play
Once
Undo
A
B
C
TAP
RATE
DEPTH
OUTPUT
INPUT
BYPASS
TU-2
Clean / OD
Boost
KEY
SCALE
MIX
HARMONY
VOLUME
TONE
COLOR
DIST
STOP
PLAY


CHECK
E.LEVEL
LOW FILTER
RATE
DEPTH
MIN MAX
CUT FLAT
MIN MAX
MIN MAX
OUTPUT B
OUTPUT A
(MONO)
INPUT
Bass
Chorus
CEB-3
BOSS
50 120 400 500 800 4.5k 10k Level
+15
0
-15
OUTPUT
INPUT
Bass
Equalizer
GEB-7
BOSS
GAIN REDUCTION
RELEASE
ATTACK
RATIO
OUTPUT
MXR
BASS INNOVATIONS
bass
compressor


CHECK
LEVEL
Astounding
Licks
Genuine
Modesty
OUTPUT
INPUT
Talent
Simulator
BOSS



KICKSTARTER

Search projects, creators, and categories

Start a project

Art Comics Crafts Dance Design Fashion Film Food Games Journalism Music Photography Publishing Technology Theater Discover

You backed this project.

Mark as received View pledge

YOUR VOICE MATTERS

How do you feel about this project so far?

Your response may be shared anonymously with this creator.

Stratus: The Super Smart Stompbox

The versatile and affordable guitar pedal from the future. It's never been easier to unleash your creativity.

Download the App

Created by

Chaos Audio

582 backers pledged $132,983 to help bring this project to life.

Last updated October 25, 2024


CHECK
E.LEVEL
LOW FILTER
RATE
DEPTH
MIN MAX
CUT FLAT
MIN MAX
MIN MAX
OUTPUT B
OUTPUT A
(MONO)
INPUT
Bass
Chorus
CEB-3
BOSS
50 120 400 500 800 4.5k 10k Level
+15
0
-15
OUTPUT
INPUT
Bass
Equalizer
GEB-7
BOSS
GAIN REDUCTION
RELEASE
ATTACK
RATIO
OUTPUT
MXR
BASS INNOVATIONS
bass
compressor


THRESHOLD
LEVEL
RATIO
BLEND
GRAVITY
DYNAMIC MASS COMPRESSOR
RATE
DELAY
DEPTH
MIX
LUNA
TIDAL ANALOG CHORUS
100Hz
200Hz
400Hz
800Hz
1.6kHz
3.2kHz
SPECTROMETER
HEX BAND EQ


Orange F 11:54
PEDALBOARD
CLOSE
REMOVE
THRESHOLD
LEVEL
RATIO
BLEND
GRAVITY
DYNAMIC MASS COMPRESSOR


OUT
STRATUS


THRESHOLD
LEVEL
RATIO
BLEND
GRAVITY
DYNAMIC MASS COMPRESSOR


100Hz
200Hz
400Hz
800Hz
1.6kHz
3.2kHz
SPECTROMETER
HEX BAND EQ


RATE
DELAY
DEPTH
MIX
LUNA
TIDAL ANALOG CHORUS


THRESHOLD
LEVEL
RATIO
BLEND
GRAVITY
DYNAMIC MASS COMPRESSOR


100Hz
200Hz
400Hz
800Hz
1.6kHz
3.2kHz
SPECTROMETER
HEX BAND EQ


MARINER
URSA MAJOR
SUPERNOVA
VOID
ODYSSEY
URSA MINOR
ORBITER
URANUS
PLUTO
ARTEMIS
GEMINI
PEGASUS
COSMOS
SPACEDUST
LAUNCH
LUNA
GALILEO
CONSTELLATION
INVASION
WORMHOLE
WARP SPEED
ASTRO
SKYLAR
METEOR
TYCHO
PIONEER
ARES
GRAVITY
ASTEROID
ECLIPSE


RATE
DELAY
DEPTH
MIX
LUNA
TIDAL ANALOG CHORUS


OUT
STRATUS

Dev-Portal

Public

Edit Pins | Unwatch 3 | Fork 0 | Starred 5

e1d7315 | 4 Branches | 0 Tags | Go to file t | Code

landon-chaos Added additional information to README — e1d7315 · 7 months ago — 8 Commits

examples	Clarified example FAUST code	7 months ago
resources	Cleaned up folder name	7 months ago
README.md	Added additional information to README	7 months ago

README

Dev-Portal

Everything you need to get started on developing effects for Stratus® and Tone Shop™.

Required Hardware

- Stratus multi-effects pedal

Documentation

FAUST - A useful DSP programming language. When using FAUST to generate C++ code, it will already follow the

About

Everything you need to get started on developing effects for Stratus® and Tone Shop™.

Readme
Activity
Custom properties
5 stars
3 watching
0 forks
Report repository

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 3

Developer Portal - Access

Inbox ×

Landon McCoy <landon@chaosaudio.com>
to me

Hey Martin,

I see you're interested in developing effects for Stratus! Or at least poking around out of curiosity. 🙂

Here is a link to the GitHub repository with some basic instructions and example code:
https://github.com/chaosaudio/Dev-Portal

In order to build effects for Stratus, you'll need access to Stratus' root Linux user. Here is the password for that user:

Feel free to change this password after logging in.

*** *NOTE: This is sensitive information and should not be shared with anyone outside of your development team!* ***

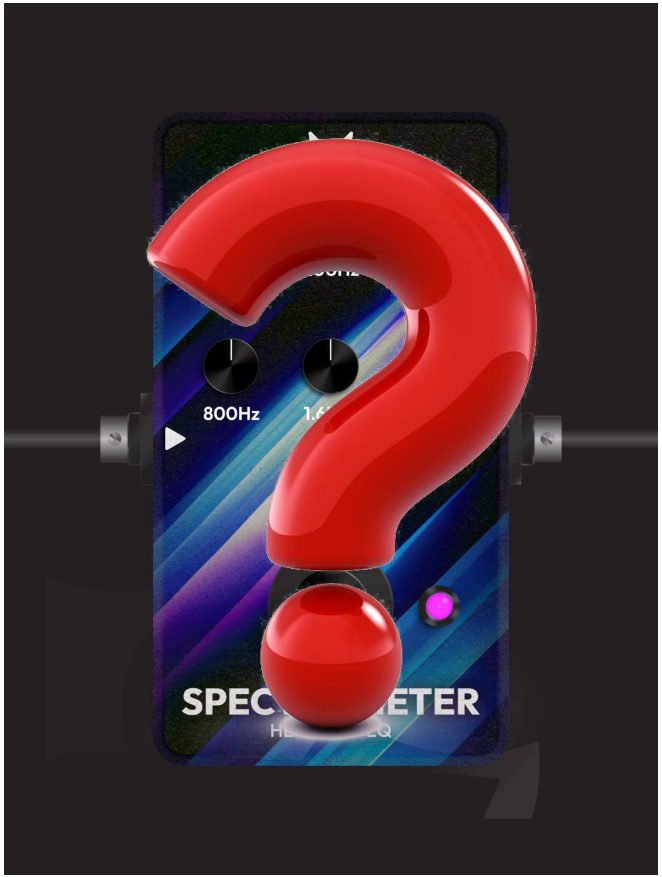
Let me know if you have any questions!

All the best,

In order to build effects for Stratus, you'll need access to Stratus' root Linux user. Here is the password for that user:

[redacted]

Feel free to change this password after logging in.


800Hz

README

Dev-Portal

Everything you need to get started on developing effects for Stratus® and Tone Shop™.

Required Hardware

- Stratus multi-effects pedal

Documentation

FAUST - A useful DSP programming language. When using FAUST to generate C++ code, it will already follow the general format necessary for Stratus.

Usage

You must include the provided `dsp.hpp` file in any algorithms you compile. You can find this header file in the `resources` folder.

```
import "dsp.hpp"

class example_effect : public dsp {

}
```

Compilation

SSH into Stratus, copy your files to Stratus' local filesystem, and build your algorithm with the following command:

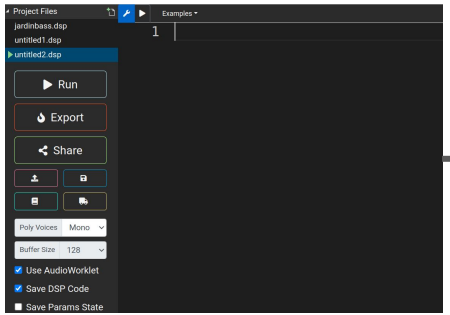
```
  g++ -fPIC -shared -O3 -g -march=armv7-a -mtune=cortex-a8
-mfloat-abi=hard -mfpu=neon -ftree-vectorize -ffast-math "EFFECT_NAME".cpp -o "EFFECT_NAME".so
```

These flags will ensure that the most optimized binary is generated.

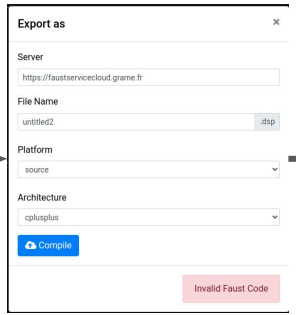
You can log into Stratus via your terminal with the following command (Mac and Linux only):

```
ssh root@stratus.local
```

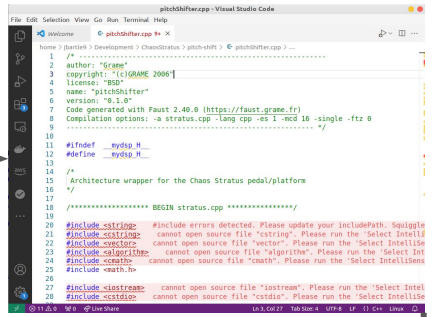
The Initial Development Process For Stratus!



Design in Faust



Export the C++ source



HEAVILY massage it to fit the Chaos interface

Copy to the right place with the right name

Logon to the pedal and compile the source

Copy to the pedal

And THEN

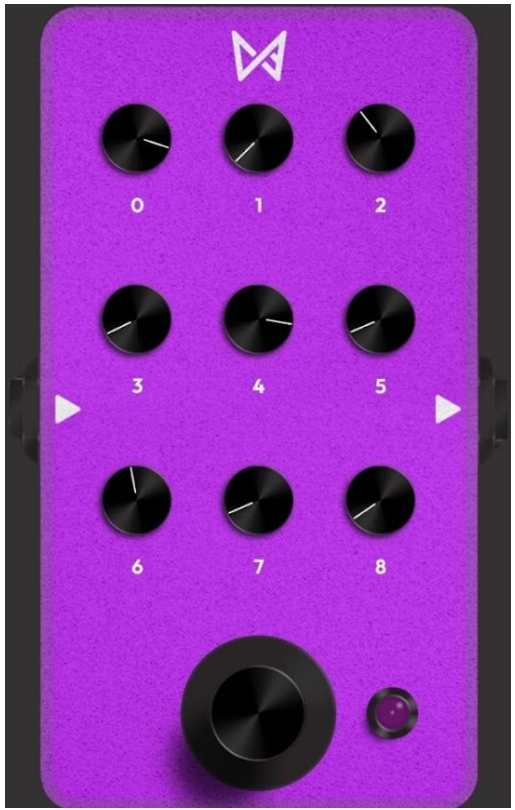
- Build a UI for it in the development portal ... using a tool that wasn’t QUITE usable at the time...

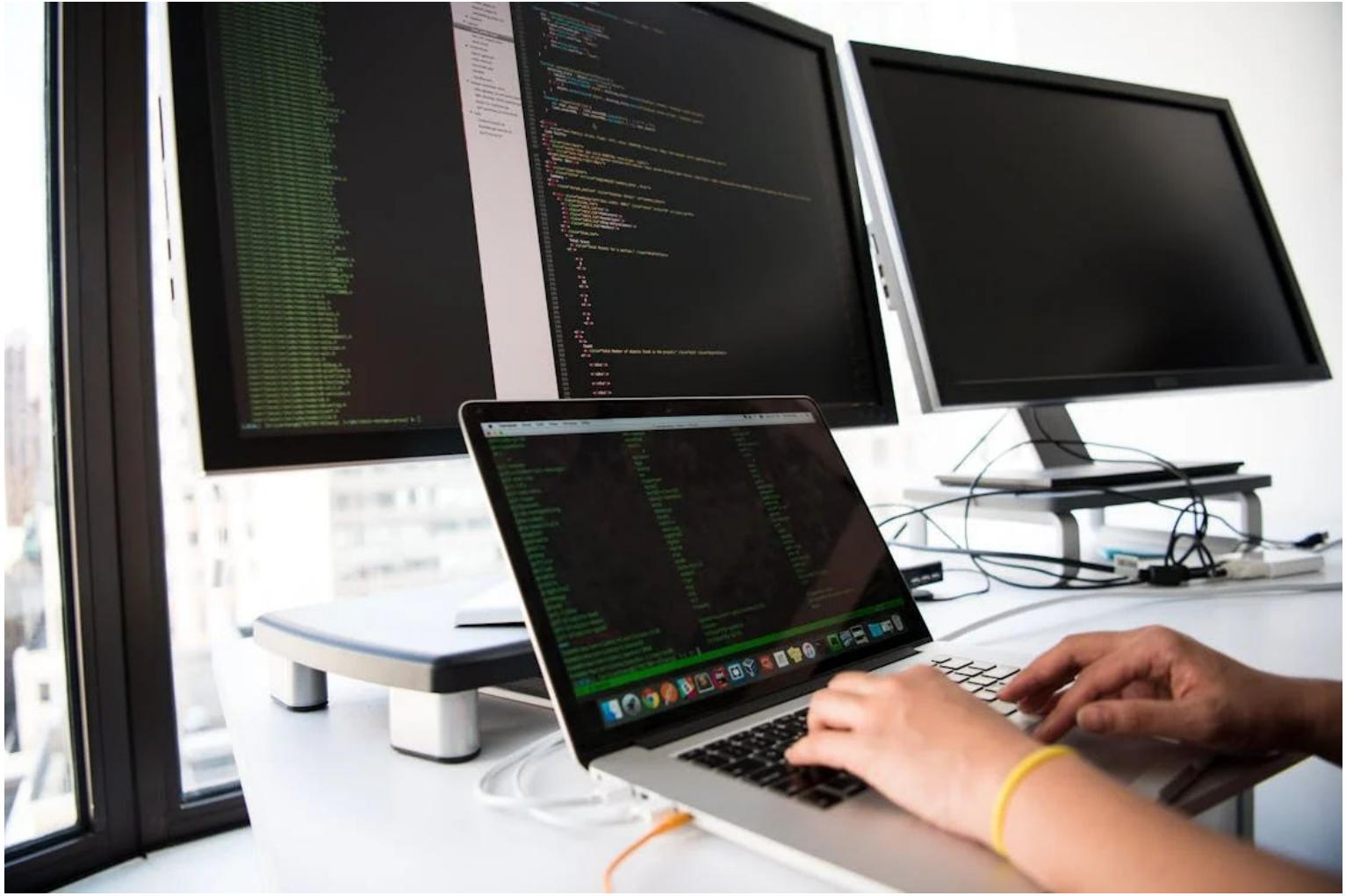
Hmmmmm

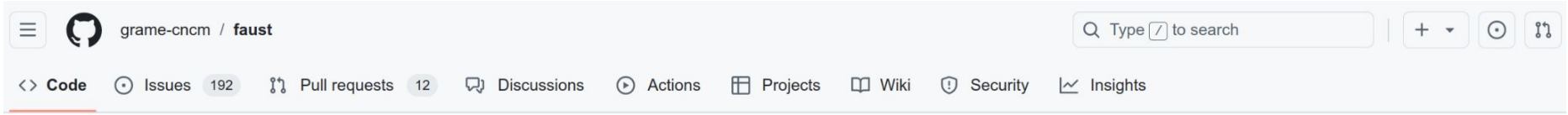

THRESHOLD
LEVEL
RATIO
BLEND
GRAVITY
DYNAMIC MASS COMPRESSOR


TIME
REPEATS
TONE
MIX
TRANSIT
BINARY DATA REPEATER

The 9-Knob Tester - my first suggestion







faust

Public

Edit Pins | Watch 89 | Fork 323 | Star 2.6k

master-dev | 49 Branches | 79 Tags | Go to file t | Add file | Code

sletz Typos. ✓ 2189f61 · 3 days ago | 13,689 Commits

.github/workflows	Set version to 2.76.0.	2 weeks ago
architecture	Set version to 2.76.0.	2 weeks ago
benchmark	Cleanup benchmarks.	4 years ago
build	Set version to 2.76.0.	2 weeks ago
compiler	Typos.	3 days ago
debian	Debian package fixes for Ubuntu 16.04.	6 years ago
documentation	Set version to 2.76.0.	2 weeks ago
embedded	Formatting.	2 months ago
examples	Add test29.dsp and vital_rev.dsp example.	4 months ago

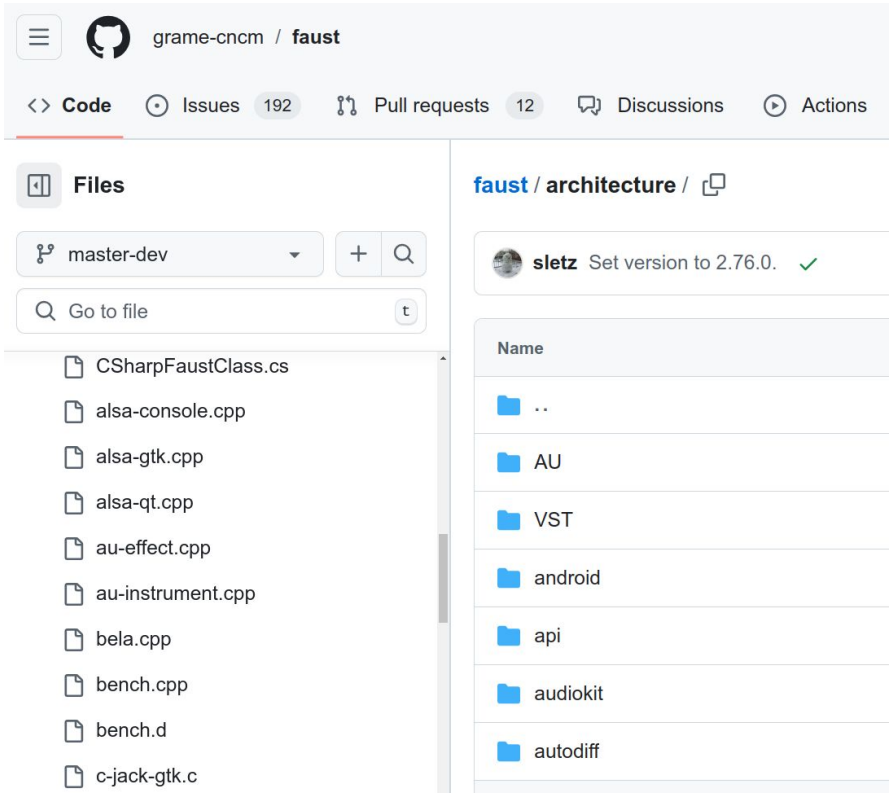
About

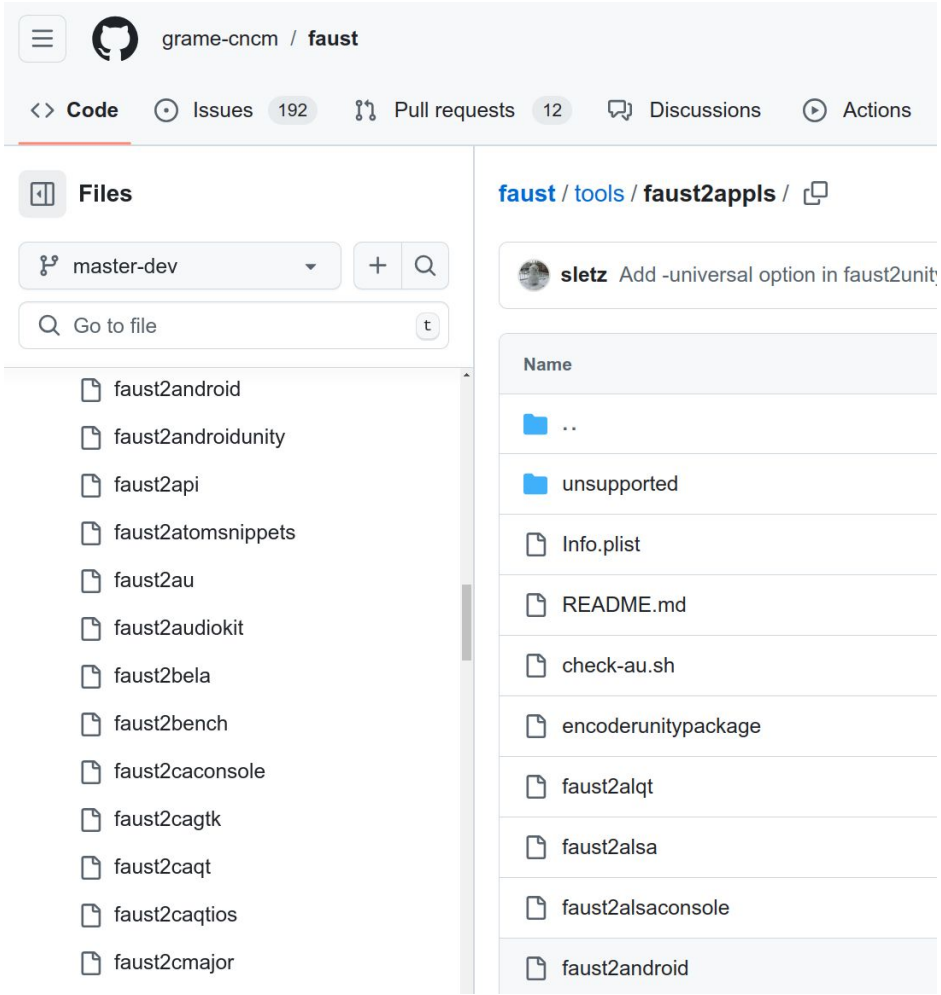
Functional programming language for signal processing and sound synthesis

faust.grame.fr

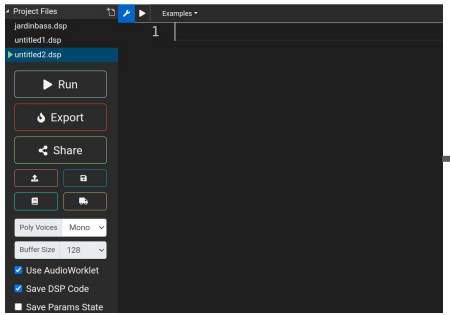
audio | c | rust | c-plus-plus | csharp | compiler | cpp | functional-programming | julia | dsp | llvm | dlang | julia-language | wasm | faust | jsfx | jax | cmajor | rnbo

Readme
View license
Activity
Custom properties
2.6k stars
89 watching

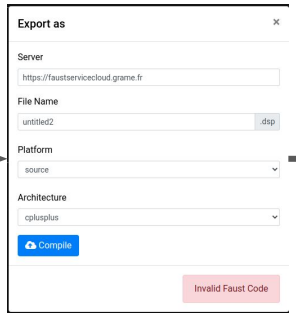

grame-cncm / faust
Code
Issues 192
Pull requests 12
Discussions
Actions
Files
master-dev
Go to file t
CSharpFaustClass.cs
alsa-console.cpp
alsa-gtk.cpp
alsa-qt.cpp
au-effect.cpp
au-instrument.cpp
bela.cpp
bench.cpp
bench.d
c-jack-gtk.c
faust / architecture /
sletz Set version to 2.76.0.
Name
..
AU
VST
android
api
audiokit
autodiff


grame-cncm / faust
Code
Issues 192
Pull requests 12
Discussions
Actions
Files
master-dev
Go to file t
faust2android
faust2androidunity
faust2api
faust2atomsnippets
faust2au
faust2audiokit
faust2bela
faust2bench
faust2caconsole
faust2cagtk
faust2caqt
faust2caqtios
faust2cmajor
faust / tools / faust2appls /
sletz Add -universal option in faust2unity
Name
..
unsupported
Info.plist
README.md
check-au.sh
encoderunitypackage
faust2alqt
faust2alsa
faust2alsaconsole
faust2android

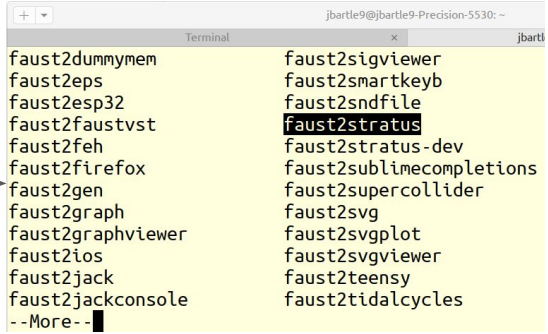
The Next Iteration Development Process For Stratus!



Design in Faust



Save the DSP source



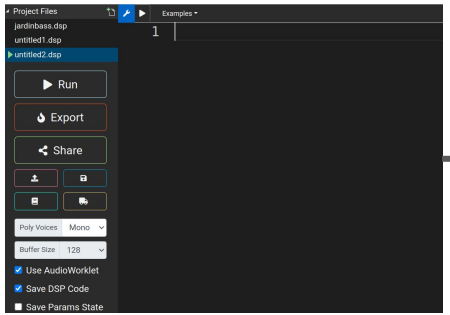
Run faust2startus to gen the C++

Copy to the pedal

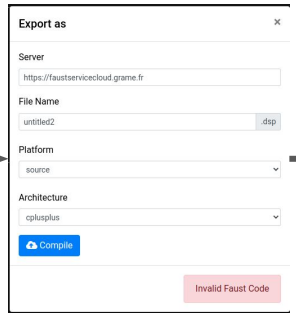
Logon to the pedal and compile the source

Copy to the right place with the right name

Once the new Faust version was deployed for the UI



Design in Faust

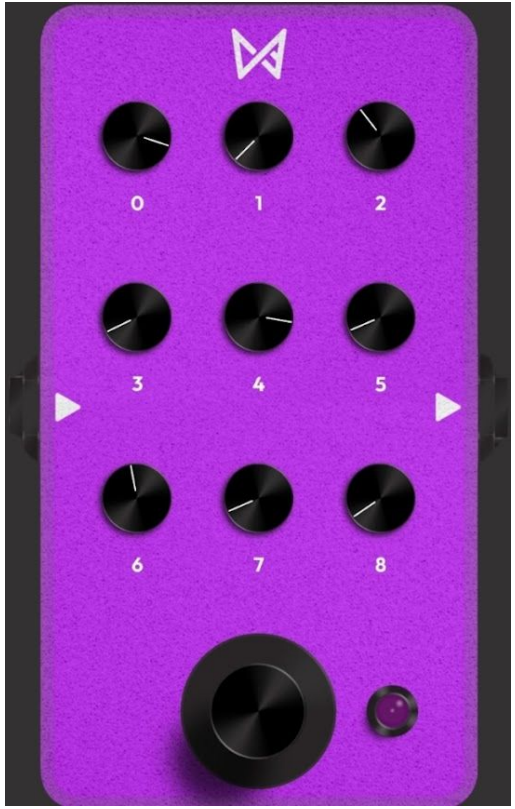


Pick chaos-stratus as your architecture and download

Copy to the pedal

Logon to the pedal and compile the source

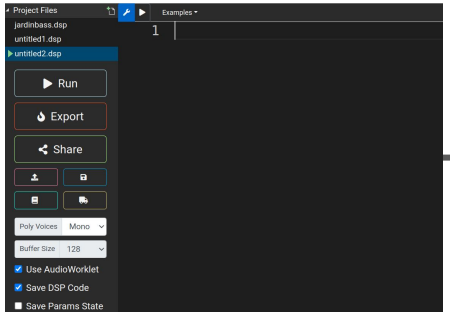
Copy to the right place with the right name


0
1
2
3
4
5
6
7
8

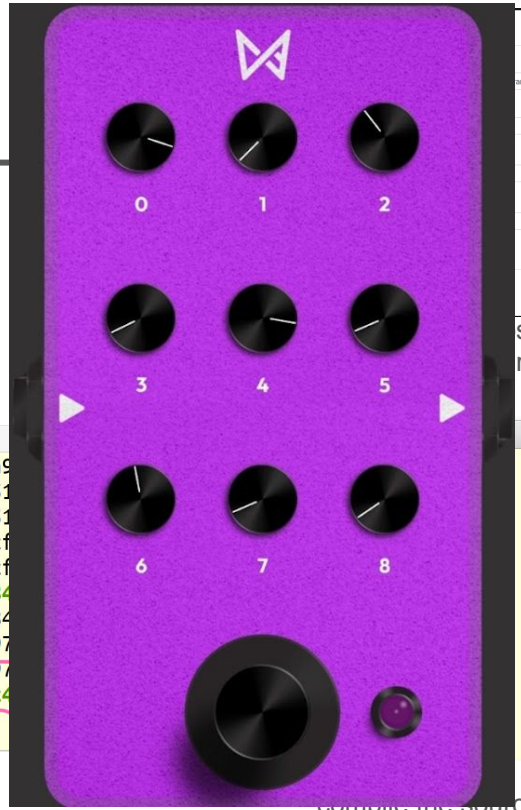
We added support for declare values for the effect UUID and version strings

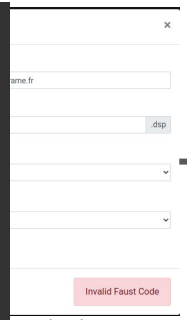
We introduced support for the “stratus:nn” metadata on a knob!

```
// Stratus dec      s
declare stratu id "55631e3a-94f7-42f8-8204-f5c6c11c4a21";
declare stratusVersion "0.1.0";
declare filename "semiparambasseq.dsp";
declare name "semiparambasseq";

// The ubiquitous import
import("stdfaust.lib");

// Slider (Knob) functions when this has its own UI
live_eq_slider(0,L,C,H)  = hslider("[%0]Freq[style:knob][stratus:%0]",C,L,H,1) ;
live_level(0) = hslider("[%0]Gain[style:knob][stratus:%0]", 0, -15, 15, 1);

// Slider (Knob) functions while we use the 9KNOB test effect UI
knob9_eq_slider(0,L,C,H) = hslider("[%0]Freq[style:knob][stratus:%0]",5,0,10,0.1) : *((C-L)/5) : +(L);
knob9_level(0) = hslider("[%0]Gain[style:knob][stratus:%0]", 5,0,10,0.1) : -(5) : *(3);

// Which of the above we actually want to use at the moment
eq_slider(0,L,C,H) = knob9_eq_slider(0,L,C,H);
level(0) = knob9_level(0);
```

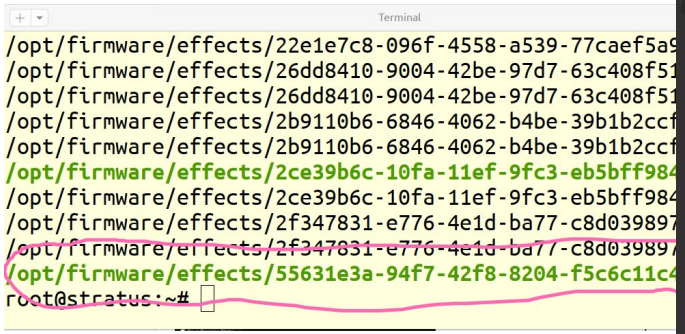


Design in Faust

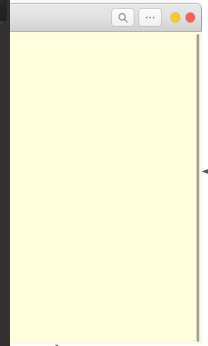




s-stratus as your
re and download



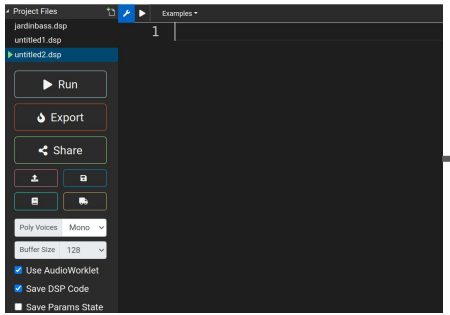
Copy to the right place with the right name



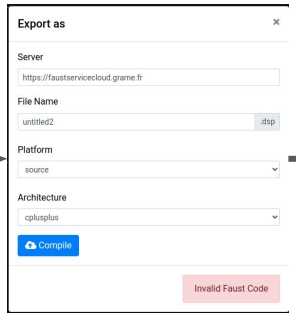
and
compile the source

Copy to the pedal

What if you could get the Faust IDE *itself* to build it!



Design in Faust



Pick chaos-stratus as your architecture, build and download directly to the pedal

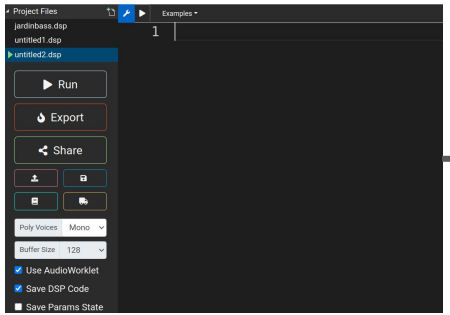
Installed to the pedal

SLETZ WON’T LET ME!

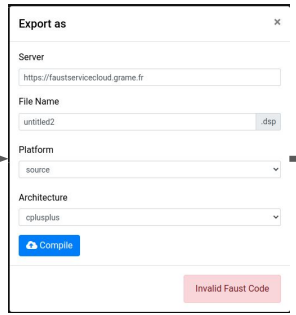
For good reason:

- It requires a specific Dockerfile pulled from “somewhere” and launched within the Faust service
- We don’t want an ad-hoc solution just for Stratus
- BUT how do you ensure that some malevolent Fauster doesn’t create an architecture that builds insidious things that compromise a user’s devices?

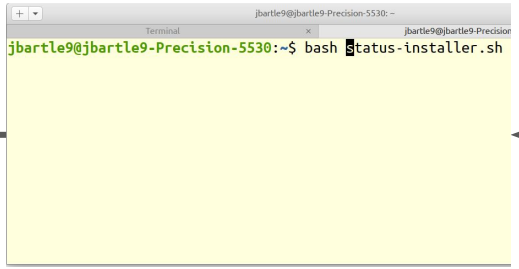
OK, what if you could get an executable *installer* from the IDE



Design in Faust



Pick chaos-stratus as your architecture,
click compile and download



Run the installer

Installs to the pedal

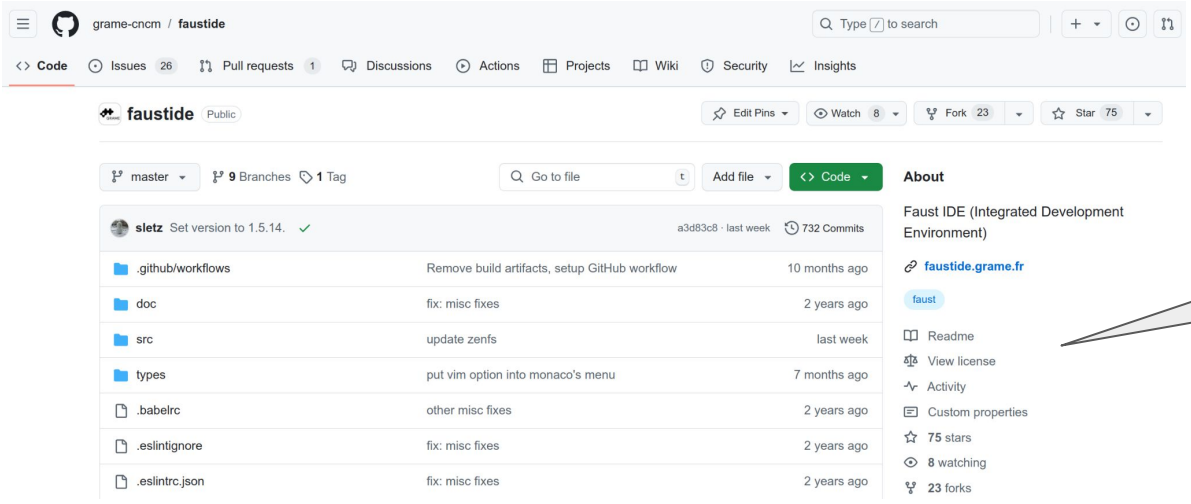
What stops us fully doing *that*

The IDE only expects to build zips

That means you have to

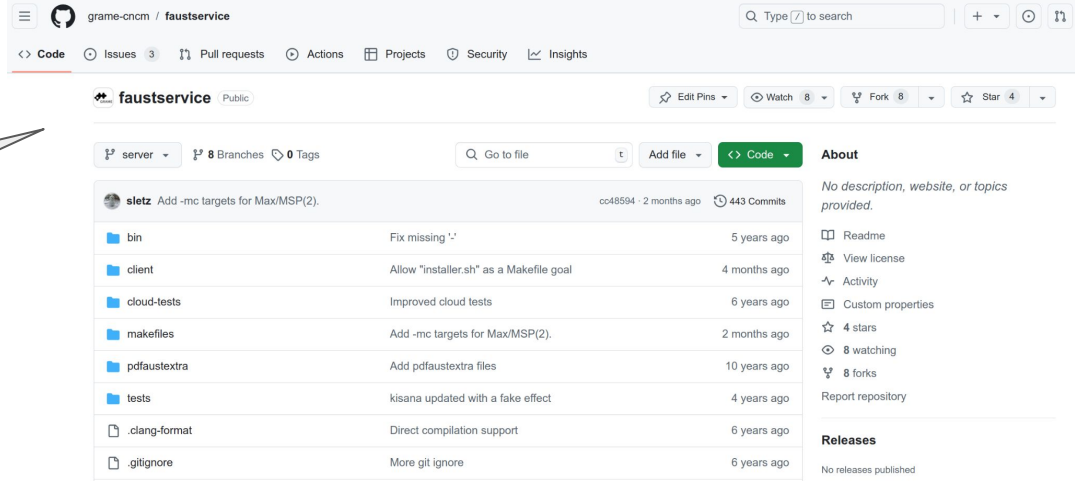
- Download the zip
- Unzip the zip SOMEWHERE
- Run the installer

Which is *OK* but ... *WHY* does the IDE only allow zips?



The IDE itself

The service that builds stuff



Use the Location header if it is available

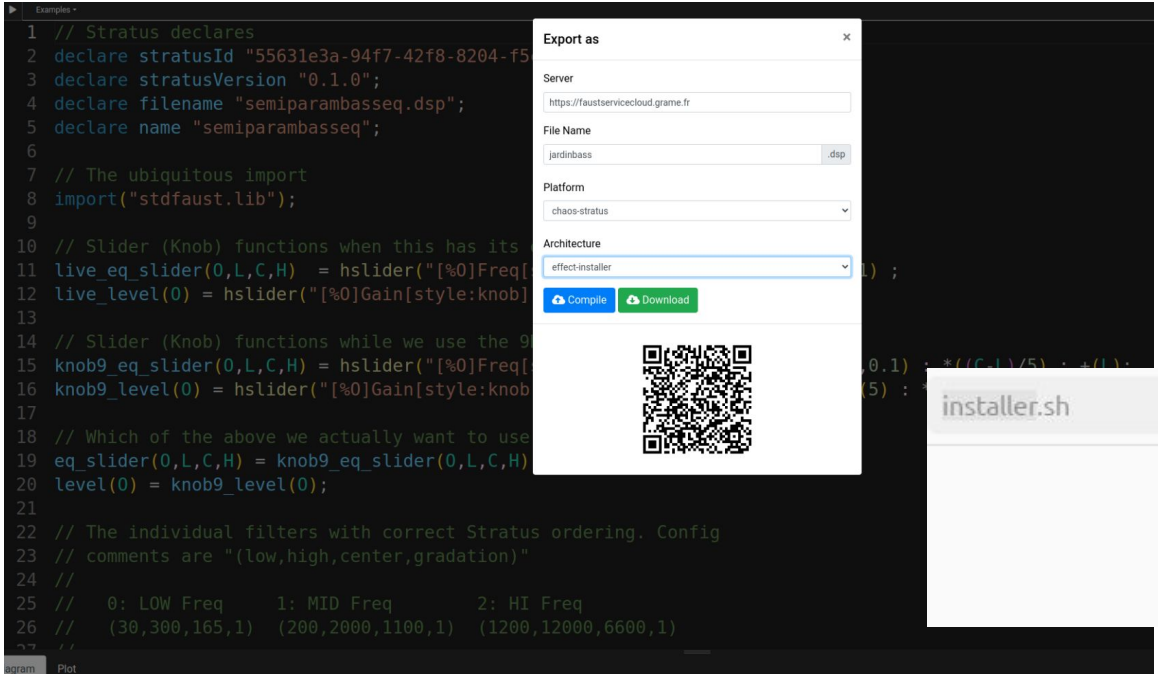
#96 by bassmanitram was closed on Jul 6

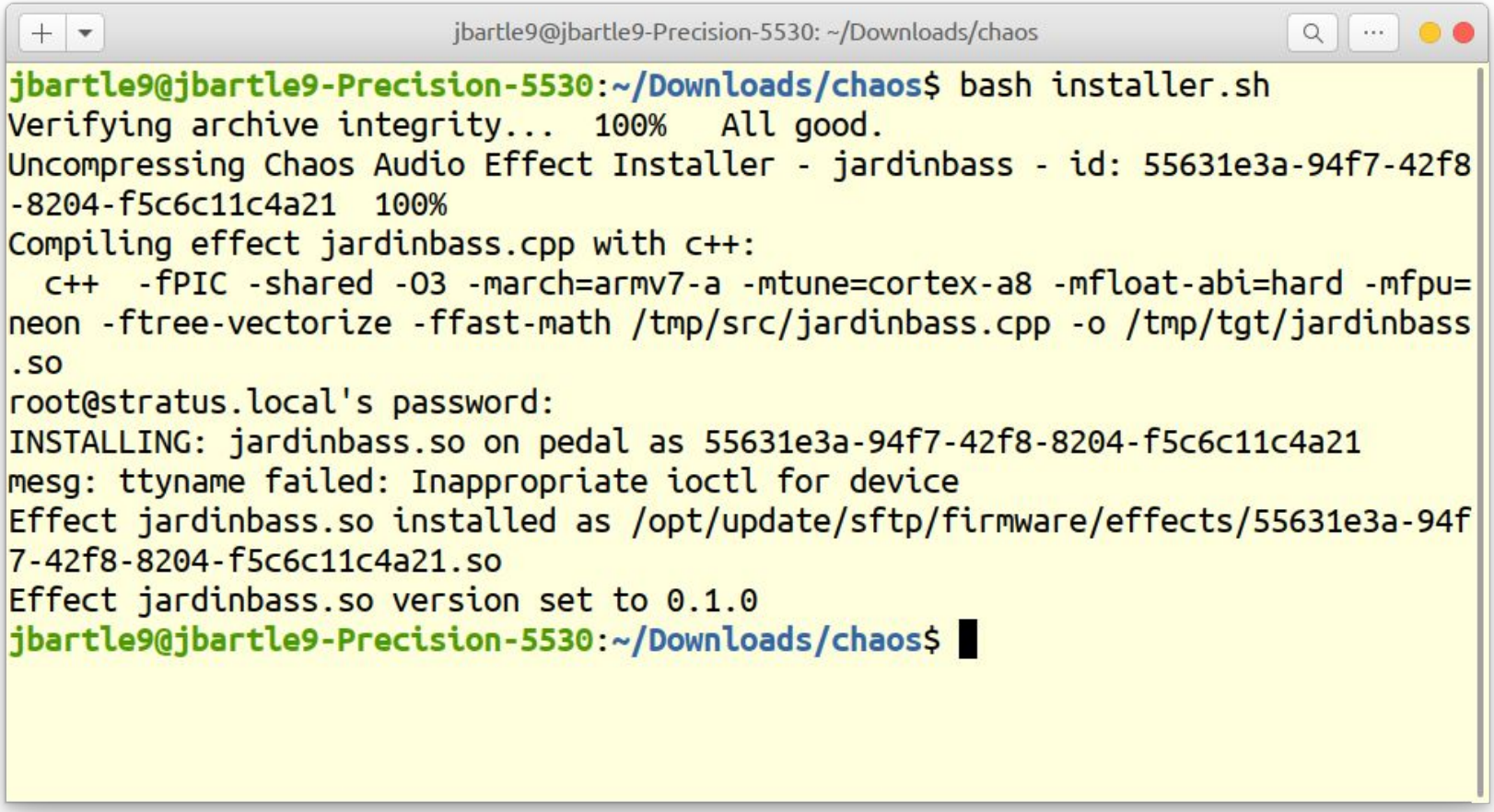
The PR that lets the IDE know what the *service* has generated rather than fixating on .zip files.

The PR that lets the service generate non .zip files and tell the IDE what has been generated.

Modify the faustservice to facilitate building self-extracting installers

#11 by bassmanitram was closed on Jul 6



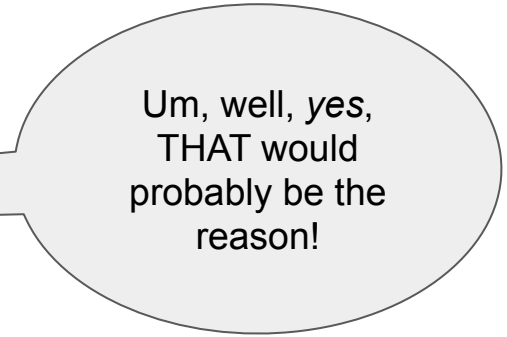

jbartle9@jbartle9-Precision-5530: ~/Downloads/chaos
jbartle9@jbartle9-Precision-5530:~/Downloads/chaos$ bash installer.sh
Verifying archive integrity... 100%   All good.
Uncompressing Chaos Audio Effect Installer - jardinbass - id: 55631e3a-94f7-42f8
-8204-f5c6c11c4a21 100%
Compiling effect jardinbass.cpp with c++:
  c++ -fPIC -shared -O3 -march=armv7-a -mtune=cortex-a8 -mfloat-abi=hard -mfpu=
neon -ftree-vectorize -ffast-math /tmp/src/jardinbass.cpp -o /tmp/tgt/jardinbass
.so
root@stratus.local's password:
INSTALLING: jardinbass.so on pedal as 55631e3a-94f7-42f8-8204-f5c6c11c4a21
mesg: ttyname failed: Inappropriate ioctl for device
Effect jardinbass.so installed as /opt/update/sftp/firmware/effects/55631e3a-94f
7-42f8-8204-f5c6c11c4a21.so
Effect jardinbass.so version set to 0.1.0
jbartle9@jbartle9-Precision-5530:~/Downloads/chaos$

Now Let’s Fix Stuff

Making use of all this to get bass effects

Fixing the compressor

```
...
process = _ <: (_:*(1-mix)),(
fi.low_shelf(-4,250):
compressor(ratio,thresh,att,rel,kneeAtt):*(mix)) :> _:*(gain);
```



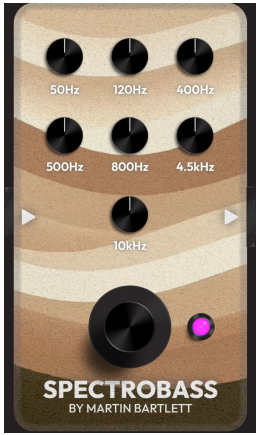
The original code (obtained under NDA... errr... so... um... sorry Landon?)

```
...
process = _ <: (_:*(1-mix)),(
compressor(ratio,thresh,att,rel,kneeAtt):*(mix)) :> _:*(gain);
```

NOW it’s a bass compressor!

A 7-band bass EQ!

```
eq1 = vslider(  "50hz[stratus:0]",  0, -15, 15, 0.1);
eq2 = vslider( "120hz[stratus:1]",  0, -15, 15, 0.1);
eq3 = vslider( "400hz[stratus:2]",  0, -15, 15, 0.1);
eq4 = vslider( "500hz[stratus:3]",  0, -15, 15, 0.1);
eq5 = vslider( "800hz[stratus:4]",  0, -15, 15, 0.1);
eq6 = vslider("4.5Khz[stratus:5]",  0, -15, 15, 0.1);
eq7 = vslider( "10Khz[stratus:6]",  0, -15, 15, 0.1);

fc1 = 50;
fc2 = 120;
fc3 = 400;
fc4 = 500;
fc5 = 800;
fc6 = 4500;
fc7 = 10000;

bw1 = 50;
bw2 = 120;
bw3 = 100;
bw4 = 120;
bw5 = 300;
bw6 = 1800;
```




```
filt1 = fi.peak_eq(eq1,fc1,bw1);
filt2 = fi.peak_eq(eq2,fc2,bw2);
filt3 = fi.peak_eq(eq3,fc3,bw3);
filt4 = fi.peak_eq(eq4,fc4,bw4);
filt5 = fi.peak_eq(eq5,fc5,bw5);
filt6 = fi.peak_eq(eq6,fc6,bw6);
filt7 = fi.high_shelf(eq7,fc7);

process = filt1: filt2: filt3: filt4: filt5: filt6: filt7;
```

- Basically the stock EQ with an extra knob
- Parameters based upon the freq response charts of the Boss GEB-7 Bass EQ
- UI by Landon!
- Published (and free)

A tutorial

chaosaudio / **Dev-Portal**

Type / to search

Code | Issues | Pull requests | Actions | Projects | **Wiki** | Security | Insights

Faust and the Stratus - a basic tutorial

Edit | New page

Martin Bartlett edited this page on May 28 · 7 revisions

Introduction

It sounds so COOL - build your OWN effects for your Stratus!!! Using "Faust" no less!!!

But what does it really mean?

Well, here I'm going to take you through...

- the creation of a Stratus-compatible effect algorithm in Faust,
- installing it on your pedal
- testing it with the **9 Knob** tester effect

... without you having to pre-install *anything* else on your computer or your pedal.

Then, in a more advanced section, I'll take you through installing Faust and building and testing your effect locally before committing it to your pedal.

We *won't* be covering UI design here - we'll have a separate page for that in the future.

The sections will appear as I go through the steps myself, so if you get here early you may not have all the steps yet

Prerequisites

There are a few prerequisites to getting the most out of this WiKi:

- You need to be using UN*X in some form, because the effect installer is a UNX* shell script. Linux is the obvious choice, but the installer also works on Macs (because Macs are actually UN*X-based), and even Windows has Linux support now.

  If you got this far, you probably know already that even your Stratus pedal is running Linux ... it *is* feasible (but not yet fully supported) to run the installer script on the Stratus itself! We'll cover that in the advanced section!

- You need to have signed up to the Chaos Stratus developer program, by which means you will have been given the

Pages 2

Find a page...

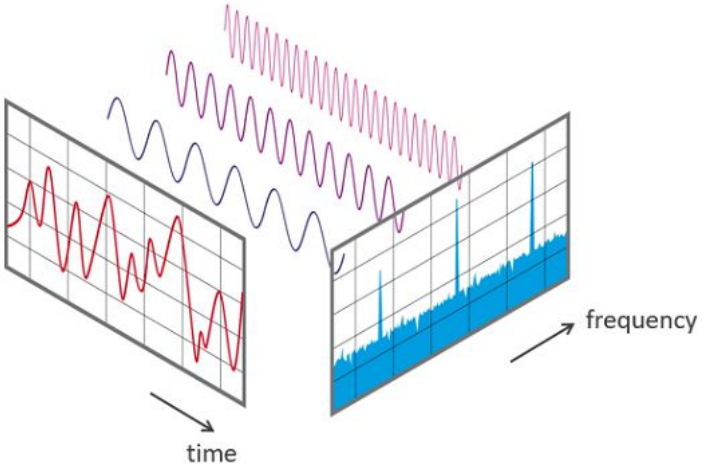
- Home
- Faust and the Stratus - a basic tuto...
  - Introduction
  - Prerequisites
  - The Effect
    - The DSP Code
    - Explanation
      - The declares at the top
      - The import statement
      - The process at the bottom
      - The EQ
      - The parts of the EQ
        - That odd/weird integer
      - A bit of function indirection
      - The Stratus controls
  - Building/Installing - the EASY way
    - The ... "do something with my DSP" ... panel in the IDE
    - Installation
      - Installation directly on the pedal

+ Add a custom sidebar

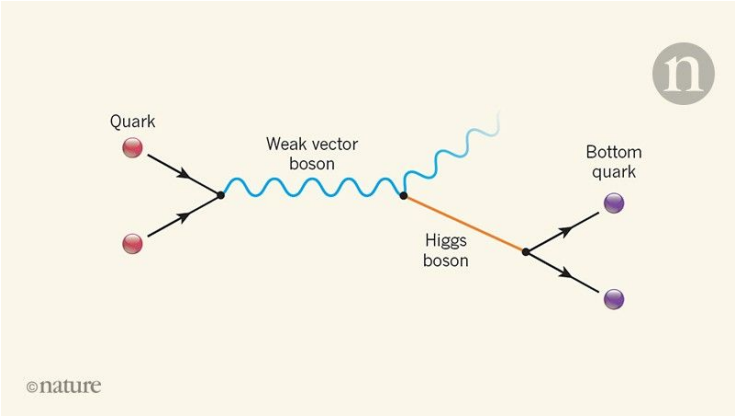
Emulates this with the 9-Knob test effect

https://github.com/chaosaudio/Dev-Portal/wiki/Faust-and-the-Stratus-%E2%80%90-a-basic-tutorial

I can now tell the difference between the Fourier Transform and the Higgs Boson



VS



To do

- Work with SLETZ on *some* scheme that might let the build occur in the Faust service
  - Build on the pedal is slow if the effect is complex
  - Build on a computing device requires Docker to be installed
- Sort out the Faust IDE code!
  - Lets just say it needs some love
  - Implement a logarithmic spectrometer option - I have a working implementation :)
- Oh and ...


CHECK
LEVEL
Astounding
Licks
Genuine
Modesty
OUTPUT
INPUT
Talent
Simulator
BOSS

FUN DEMO TIME!

AIDA DSP and CHAOS AUDIO

AIDA DSP is bringing their ML amp modeling to Stratus, powered by their AIDA-X modeling system.

Faust plugins, AIDA amp plugins, and any other plugins can be chained together – just like a real pedalboard!

https://aida-x.cc/

Martin Bartlett’s

FAUST

Stratus toolkit

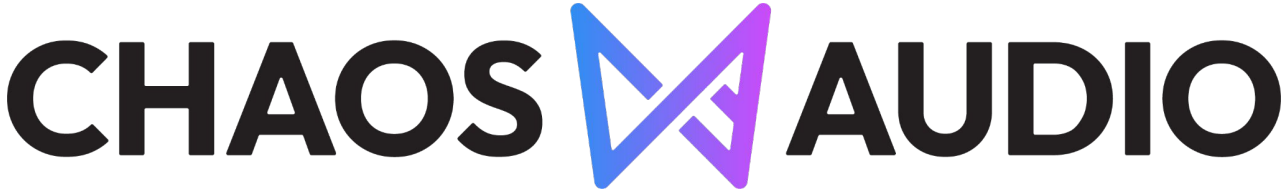
=

So much potential for

NEW PLUGINS!

HUGE THANK YOU TO MARTIN!!!

(Literally a huge “thank you”) (Everyone claps virtually!)



Use code “**STRATUSIFC24**” to get $20 off Stratus!

Use code “**PLUGINSIFC24**” to get 20% off all the plugins!

https://chaosaudio.com/

Thank You