# ToDo

move Filter settings to config prefs

rework Settings menu a bit

## Will changes to patches outside of the editor be wiped out?

need to figure out checksum (again)

need to be able to write patches from my program

## Ascertain, once and for all, the BEHAVIOR of the controller with regards to:

### “Hardware” versus “Basic” modes (boot with UP pressed, or not)

is the case that practically speaking, hardware mode just allows you to change patches, but basic mode works of the patch in slot 0?

can you have different poly\_modes per hardware patch?

why the mystery around the up-down-left-right buttons vs modes

why the mystery around the slider button vs modes

### Confirm that “splits” (fret and string settings) affect behavior of notes

pgm\_change (number) and bank\_lsb/msb are ONLY sent out by Editor program?!?

midi\_volume and midi\_reverb are ONLY sent out by Editor program?!?!

PITCH\_BEND is affected

TRANSPOSE is affected

DYN\_SENSE (range) is affected

DYN\_OFFSET is affected.

### Figure out if the Pedal affects controller behavior

Manual says: “Send CC 66 value 127, 0 on release, for hold/loop"

Can you send these CCs to the controller to affect behavior?

### NOTING THAT ANYTHING I DO COULD CHANGE IN A FIRMWARE UPDATE

### and there is a pending “Software” update.

## Figure out how “tempo” works in Quantiloop and Audiobus

how does Quantiloop determine the “tempo” of the “first track recorded”

and how does that relate to the range of the tempos that AudioBus sends out

which appear to go from 60 to something like 60+7f = 187?

# Pre-design notes

## Guidelines

The Tuner is NOT an oft used first priority.

Nor is the Sensitivity window.

The FTP Settings window becomes part of editing a “patch”.

- Hence why I call them “configuration” instead of patches.

Brightness still doesn’t have an integer editor, and does not need to be first in the list.

## Change “patch” back to “config” ??

Whats another word for these things, as “patches” will now be a sub-part of a single rig.

“rig” ???

## Pedal behavior for compatability with “OldRig”

I want a whole new set of pedal behaviors.

Pedals should keep working while in configSystem, so …

Pedal behavior has to be set globally by the “last selected rig”

and modally, perhaps, while IN the rig.

## Other backwards compatability issues

oldRig existed before there was midi control over AudioBus.

With midi control over Audiobus, I can achieve relative loop and synth volumes and use AudioBus for the “master” control, obviating the need for a “volume pedal” in Tonestack.

For backwards compatability, there ToneStack and AudioBus should have “oldRig” and “newRig” thingees.

AND it is possible to automatically change between them …

# New Rig Discussion

A MORE WHOLISTIC APPROACH to controlling the iPad and the apps on it.

The newRig configuration, new approach, involves more than just the pedal blindly sending out control changes and stateless midi-messages.

The newRig configuration will be very specific to the FTP, Quantiloop, and SampleTank, and will maintain state, and provide modes.

The idea is to cram as much into a “real time” pedal as possible, while still falling short of a script based approach.

A second possible configuration IS such a script based approach.

## New Capabilities

### Switch Programs

Seems like a top level button.

Switch programs automatically on certain things (like setting patch volumes).

What else can I do to, or get from, SampleTank.

What else can I do to, or get from, Quantiloop.

Do I want to go as far as to have individual control over guitar effects, or should they remain “global”, as set on iPad touch screen?

### Tempo Changes

## Enhanced Capabilities

### Relative Volumes for loope/synth tracks

Can be achieved by using AudioBus for “master” volumes.

Have (uneeded?) added redundancy in Sample Tank “Volume” vs “Master Volume”

#### Alternative B

The rig/system will maintain a set of relative volumes for the four Quantiloop tracks.

There will be a rapid way to change the relative volumes.

The overall volume will scale these relative volumes.

### “Latching” Foot pedal (continuous) controllers.

One big issue is achieving a consistent with the pedals state when changing patches.

I currently sort-of achieve this by making sure all SampleTank patches have a volume of zero, so that when I switch to them, they don’t make any noise. But still, then, as soon as I touch the pedal, they “jump” to the pedal value.

Changing patches will automatically set the volume to zero, THEN change the patch.

Each patch \*should\* have a “default” volume that is easily achievable.

After a patch change, the synth volume pedal should do nothing unti it is returned to zero.

Perhaps on the first “up” movement (upto or past the “default” volume), should only go to the default volume.

So there is this generalized idea that you have to “latch” a pedal by moving past a value, and then returning to it.

Will require evident UI that the latch is engaged, or the pedal is in an unlatched state.

Only “good” display device is the screen. Hard to imagine changing button colors or intensities that have other semantics.

### Damped Control Movments

Controls should have selectable “damping” ..

Perhaps from 0 to 255 .. as the number of milliseconds, minimum, for the control to send out an incremental change of one unit.

Particularly important for Tempo.

### FTP Configuration on a per-patch basis

#### Per patch

Poly Mode

Touch Sensitivity

Pedal Mode? – does this affect behavior? How? When and how is split(0) used?

##### Fret Ranges

apparently default values defines ONE split (split(1))

if off defaults define “active” splits

###### fret\_range\_low\_up\_12

the range of frets, from the lowest fret (open position) for hardware patches 1 & 2

- so default is 34 indicating that split(1) takes all 34 frets?

###### fret\_range\_high\_down\_34

the range of frets from 1f down to this number, for hardware patches 3 and 4.

- default 0?

###### string\_range\_12

the strings covered by the red hardware patch1 (as opposed to yellow hardware patch 2)

- seems like it should be the opposite, so that ZERO means all strings belong to patch1 (split1)

- does it default to 6?

###### string\_range\_34

default?

the strings covered by the blue hardware patch3 (as opposed to green hardware patch 4)

- these only come into play IF fret\_range\_high\_down\_34 is nonzero?

#### Per Split

Bend Mode

Touch Sensitivity

-

### Patch Banks

I envision allowing 32 patches in four banks of 8, or something like that, to get more patches into less buttons.

For example, using two rows of four buttons, with one button for the bank selector, I could get multiples of 8 in different colors.

Knowing where I am, and what patches are in each bank, is difficult, requires memory at this time.

Screen display is important.

### Timeout Modes

The new rig will have modes of operation (i.e. selected patch bank)

Some of these modes may be accessible for 2 seconds or something like that, with the system reverting to the “main” mode automatically.

i.e. Press the “loop relative volume mode” button, and for 2 seconds you get a chance to start modifying the relative volumes of the four loop channels.

### “Special Buttons”

#### End Song – slow tempo down as much as possible, prepare loop “stop” button to bring all volumes to zero.

#### Init Song – clear the looper, reset stuff to a known state.

#### Memory – remember the state of the sound to return to it later

#### Recall – recall the memorized state of the sound

#### Tempo Up/Tempo Down – have to work this out

how does Quantiloop determine the “tempo” of the “first track recorded”

and how does that relate to the range of the tempos that AudioBus sends out

which appear to go from 60 to something like 60+7f = 187?

# New Rig

## Assumptions

For now the use of the splits on the controller itself are too complicated to bring into play.

I am going to use SPLIT(1) settings for BEND\_MODE, DYN\_RANGE, and DYN\_OFFSET

and assume the the PGM\_CHANGE, BANK\_LSB&MSB, MIDI\_VOLUME, and MIDI\_REVERB are not sent by the controller, or not useful.