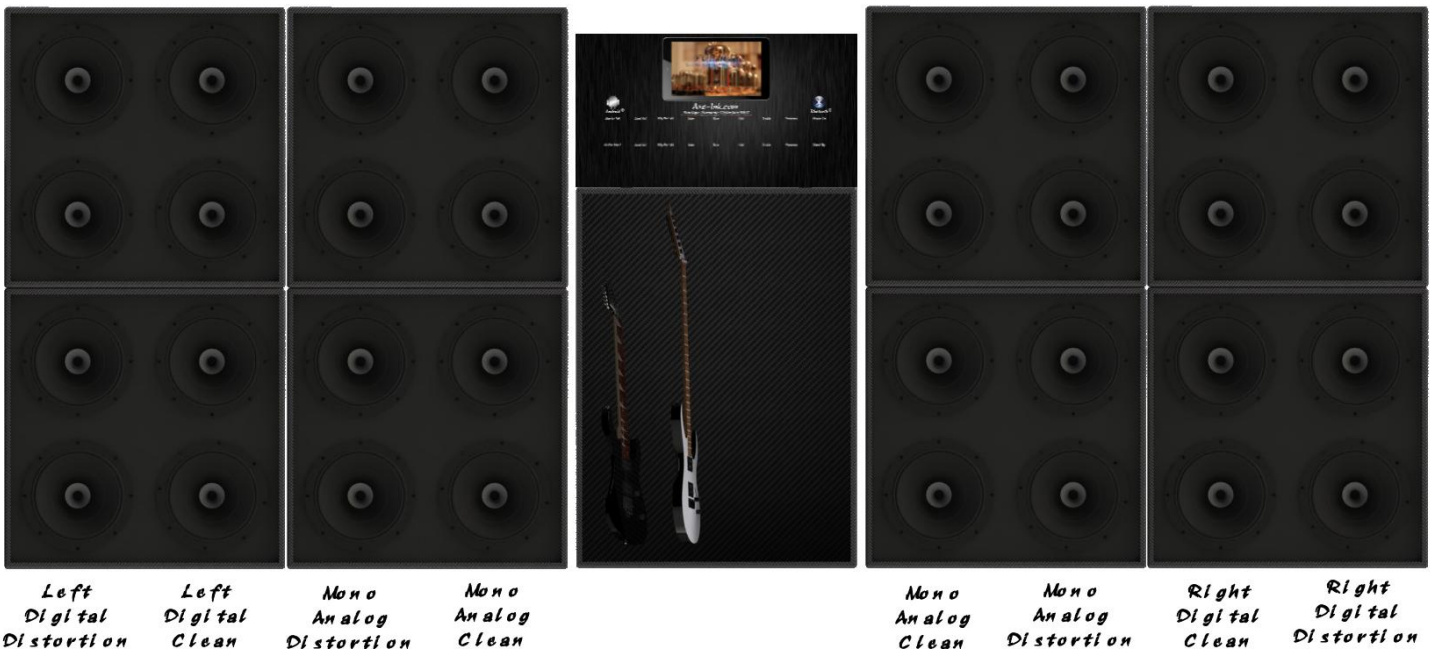

PODhead

Android Operator Interface Project

William Jordan

Axe-Ink.com

Boutique Harmonic Distortion R&D

616-848-8526

research@DTC-Intl.com

The objective of this project is to create an Android Operator Interface to a two channel tube amp that has full-time active clean & distortion channels. These analog pristine channels are mono and intended to be the base to which two full-time digital stereo clean & distortion channels will be added.

Initial design direction is to develop an android application for a Nexus 7 pad that will be removable from the amp, leveraging the visual & tactile features to provide a pleasant operator interface. Bluetooth transmission of effect presets will control the digital signal paths through a FPGA Main board.

A elaborate footboard consisting of multiple expression pedals will give real-time operator interface to preset primary parameters, for example Delay has two primary parameters of delay time & feedback.

A rack style final product will house transformers, tubes, analog & digital boards with aesthetic front & rear panels for cabling and controls.

Schedule

Due to the nature of these research & new product development activities the schedule is undetermined, but there are significant time to market concerns. Project Milestone compensation is planned to keep this project a priority.

Software Development :: Android

Swiping Screens with selectable graphical backgrounds & themes will provide the base to which Sliders and other interfaces can be added to make it easy to change preset parameters.

Persistent Preset Storage, Save to new preset and Download / Upload of Presets from Axe-Ink.com will set the stage for forums.

Read / Write of Presets via Bluetooth are the technology points and the following Preset Parameters should be padded with spare variable types and other planning for future growth.

Each effect will have its own screen(s) and a dropdown menu will permit instant access to all effects, settings and options.

Preset Parameters

preset=1

presetname=default

signalpath=stereo

Noise Gate

Sets a low signal threshold to clamp the signal to zero

private boolean noisegateon=true;

private int thld=-3;

private int atten=2;

private int attack=5;

private int release=2;

Distortion

Digital distortion

private boolean distortonon = true;

private float again=3f;

private float alvl=10;

private float bgain=8.5f;

private float blvl=10;

private float atreble=5f;

private float amid=5f;

private float abass=5f;

private float btreble=5f;

private float bmid=5f;

private float bbass=5f;

private int diga=50;

private int digb=50;

Pitch Shift

Raises or Lowers the audio interval and combines it with the original signal to create a harmony.

```
private boolean pitchon=true;

private int fx=15;

private int lvl=100;

private int bal=0;

private int shfta=4;

private int dtna=10;

private int shftb=4;

private int dtnb=10;

private int outa=100;

private int outb=100;

private int pana=30;

private int panb=30;
```

Detune

Similar to pitch shift and chorus in that the audio signal is lowered and several voices are combined with the original to fatten the sound.

```
private boolean detuneon=true;

private int lvl=25;

private int bal=0;

private int dtna=10;

private int dtnb=10;

private int dtnc=10;

private int dtnd=10;

private int outa=50;

private int outb=50;

private int outc=50;

private int outd=50;

private int spread=1;


private boolean choruson=true;
```

```
private int fx=40;
private int lvl=70;
private float speed=.3;
private float depth=4
private float delay=40;
```

Flanger

Delays the original signal and modulates it with sweeping time periods then combines it with original

```
private boolean flangeon=true;
private int fx=25;
private float lvl=10;
private int balance=0;
private float speed=.3f;
private float depth=30f;
private int fdback=50;
```

Delay

Delays original signal with original to produce an echo

```
private Boolean delayon=true;
private int fx=25;
private int dlvl=25;
private int bal=0;
private int time=333;
private int fdbck=25;
private int tapit=0;
private int smear=0;
private int lpf=0;
private int freq=10000;
```

```
private boolean phaseron=true;
private int fx=25;
```

```
private int lvl=25;
private int bal=0;
private float speed=.1f;
private int fdback=50;
private int outa=50;
private int outb=50;
private int pana=0;
private int panb=0;
private int outl=10;
private int outr=10;
```

Compressor

Sets a upper threshold to reduce the volume of loud sounds

```
private boolean compressoron=true;
private int lvl=25;
private int thld=-5;
private int ratio=2;
private int gain=50;
private int attack=10;
private int release=5;
```

Equalizer

User selectable frequencies x several screens of adjustable controls that boost or cut signal

```
private boolean equalizeron=true;
private int lvl=10;
private int band1=300;
private int band1lvl=0;
private int band2=300;
private int band2lvl=0;
.....
private int bandx=300;
```

```
private int bandxlvl=0;
```

Rotary Cabinet

Modulates & Amplifies the treble / bass signal to produce the effect of a rotating speaker like a Leslie Speaker Cabinet

```
private boolean rotaryon=true;
private int fx=25;
private int lvl=25;
private int bal=0;
private int spread=10;
private int phorn=10;
private int protor=10;
private int hlvl=10;
private int rlvl=10;
private float hspeed=.1f;
private float rspeed=.1f;
private int hdoppler=1;
private int rdoppler=1;
private int hdepth=20;
private int rdepth=20;
private int xover=10000;
private int hacc=1;
private int racc=1;
```

Reverb

Multiple delayed signals (reflections) are added to the original and then delay over time

```
private boolean reverbon=true;
private int fx=25;
private int dlvl=100;
private int bal=0;
```

```
private int density=80;
private int diff=30;
private int dis=7;
private int xover=300;
private int freq=20000;
private int damp=2;
private float delay=2.54f;
private int size=3;
private int reflect=8;
private int outl=25;
private int outr=25;
private int poutl=25;
private int poutr=25;
```

Wah Pedal

Sweeps a filter response up & down according to a foot control

```
private boolean wahon=true;
private String type="Flight";
private int fx=25;
private int lvl=25;
private int dry=50;
private int bal=0;
private int tshft=75;
private int mshft=75;
private int bshft=75;
private int mdtn=10;
private int pedal=100;
private int depth=5;
private int out=100;
private int rotary=10;
```

Circuit Design

The Altera Cyclone III evaluation kit mounted on the footboard will provide real-time operator feedback & parameter interface via an array of expression pedals. The Android Nexus 7 is mounted on the amp and provides the flashy consumer front-end that is mobile and connected to the internet.

Analog send and return for stereo clean & distortion, power and audio relay controls are the only conductors required to the footboard. The Bluetooth link provides communication with the PODhead through the footboard.

The FPGA Signal Path is critical to provide the highest grade audio possible a 24 bit audio codec is integrated, however a wider data width should be designed/planned for future PCB implementation. The multitude of expression pedal signals are readily input via the FPGA input/output allocation. Several digital rotary dials & inputs will be located on the footboard too.

Future Projects

Powered PA Mixing Board

Switched Mode Power Supply Design

Recording Software with Pristine Audio Coding Format