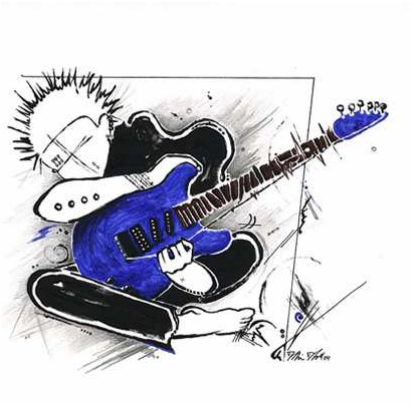
Advanced MIDI Guitar Effects System

Ronan O'Malley

Presentation Overview

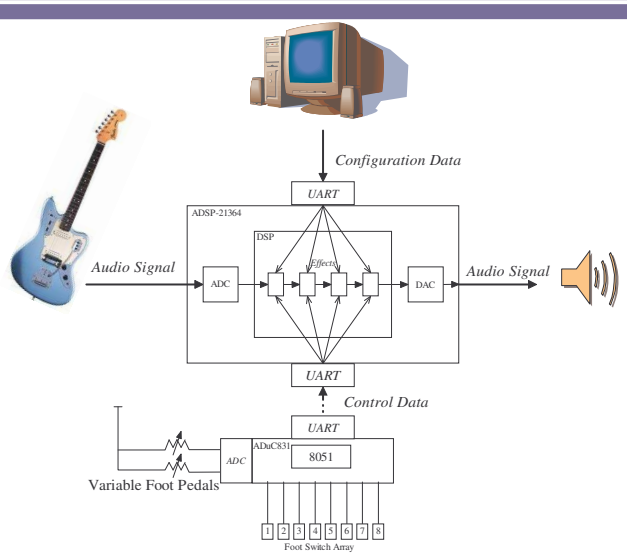
- Background
- Project Overview
- Initial Research
- Pedal-board System
- Embedded DSP System
- Java MIDI controller
- Inter-System Communication
- Conclusion



Background

- Analogue vs Digital Effects
- User Friendly Configuration
- MIDI compatibility
- Wireless Implementation

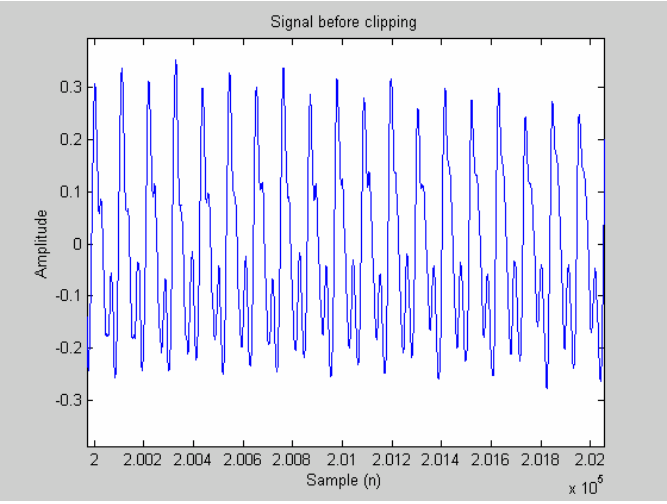
System Block Diagram



Initial Research

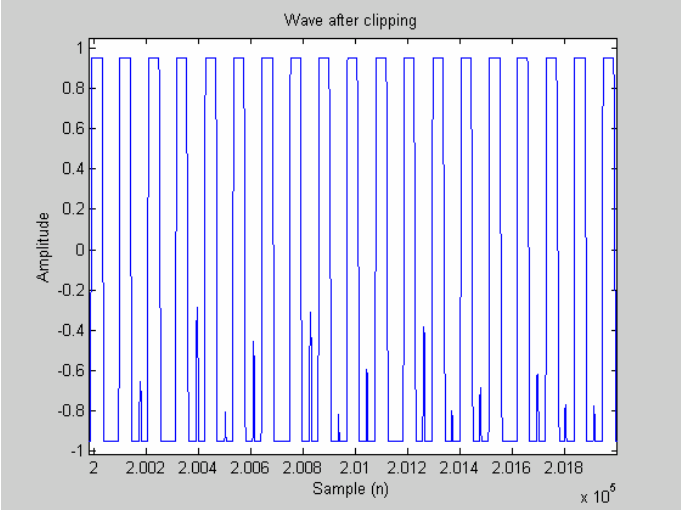
- Wav guitar samples
- Algorithm research
- Established parameters
- Find Boundaries and Limits
- Effects
  - Fuzz, Tremolo, Delay, Wah-wah, Flanger

Fuzz Distortion

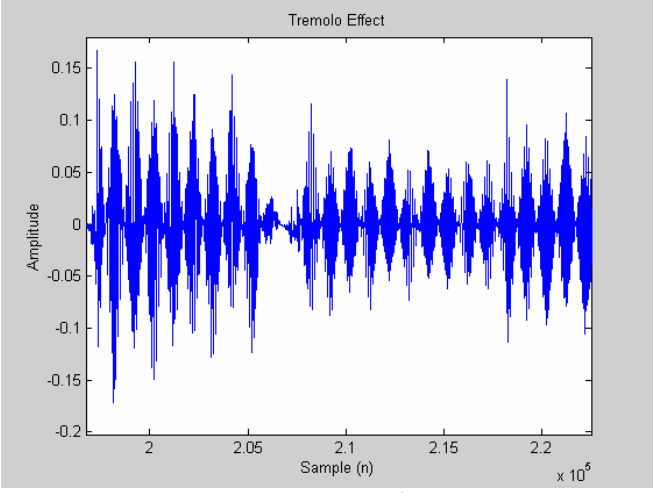


- All inputs above or below a threshold are limited

Fuzz Distortion



Tremolo



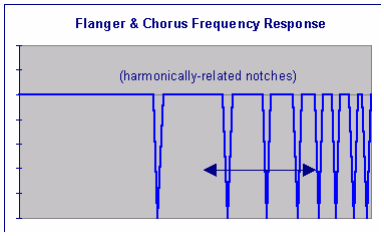
- Signal is masked by a low frequency oscillator

Delay

- Simple FIR filter.
- Add reduced delayed copies of the input sample every N samples.
- $N = T*F_s$
- Different parameters produce different effects.

Flanger

- Single delay
- Oscillated within a short range at a low frequency
- Sine wave use as oscillatory reference
- Creates notches in frequency response



Wah-wah

- Oscillate ‘peak’ filter up and down spectrum
- Filter centre frequency controlled by user pedal input
- Damping co-efficient

Pedal-Board System

- 8051 assembly code system
- Switch array read through port
- Variable resistor pedals connected to ADCs
- Simple polling style operating system
- Readings converted to MIDI and transmitted on UART
- 12 MHz clock, MIDI baud rate

Embedded DSP

- Newest generation of Analog Devices SHARC DSP
- 'Talkthrough' system constructed in C
- Effects Embedded
  - Delay, external memory
  - Tremolo triangle wave lookup table
  - Flanger interpolation

Embedded DSP System

Software UART created using synchronous serial ports

- Frame Sync Connection
- Oversampling
- DAI Interference
- 5V - 3.3V Logic level conversion

Embedded DSP System

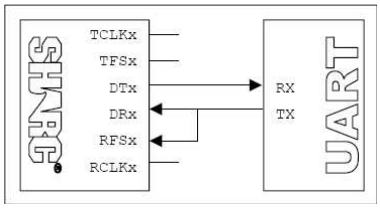


Fig 1

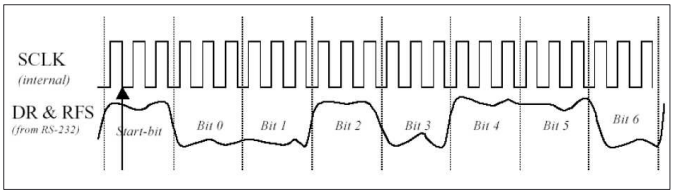
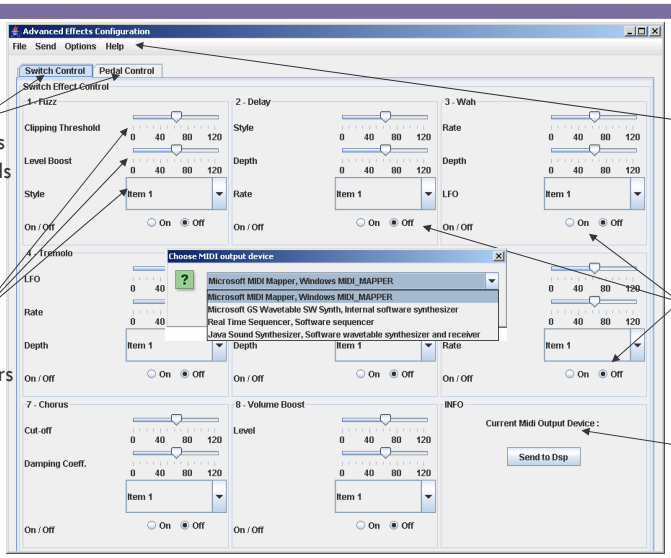


Fig 2

Java MIDI Controller Programme

- Complex GUI using *Javax.swing* API
- MIDI device selection and interaction
- *Javax.sound.midi* API
- Configuration of static ON/OFF data and multiple effect parameters
- USB to MIDI converter or sound card MIDI port can be used

Java MIDI Controller Programme



Control of switches and pedals

Set effect parameters

Configure GUI

Set effect status

Display current device

Wireless

- Not fully implemented, time constraints
- Infrastructure present

Inter-System Communication

- 5V to 3.3V conversion
  - 3.3V latch with 5V tolerant inputs
- On board interference
- MIDI opto-isolation circuit
- Baud rate configuration

Conclusion

- Research proven effect theory
- Conforming to MIDI protocol
- Different environments
- Communication between systems

Further info…

http://ohm.nuigalway.ie/02omalley/fyp_0506