

# Advanced MIDI Guitar Effects System

Ronan O'Malley

## Presentation Overview

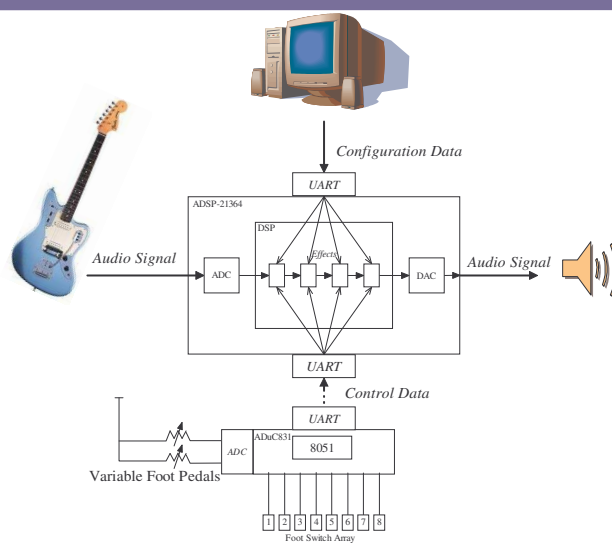
- Background
- Project Overview
- Initial Research
- Pedal-board System
- Embedded DSP System
- Java MIDI controller
- Inter-System Communication
- Conclusion



## Background

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

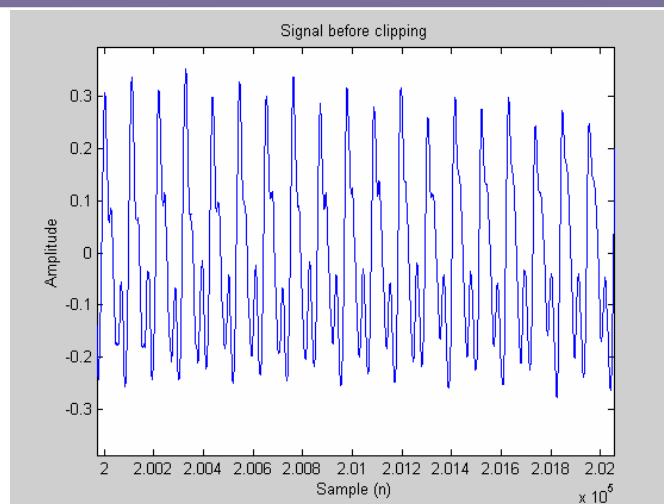
## System Block Diagram



## Initial Research

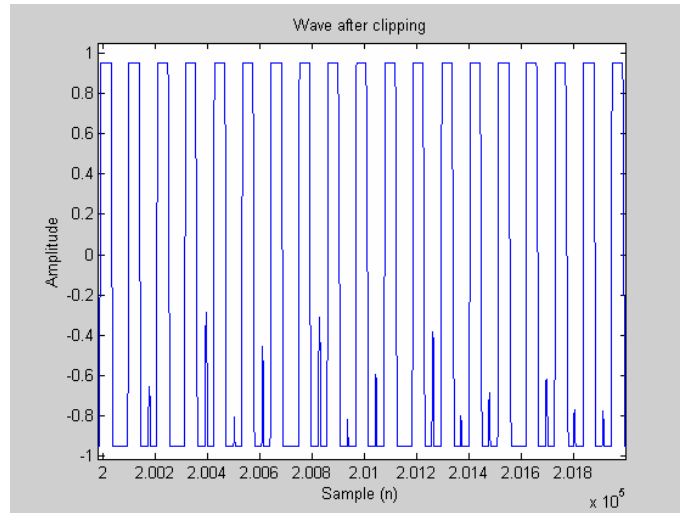
- Wav guitar samples
- Algorithm research
- Established parameters
- Find Boundaries and Limits
- Effects
  - Fuzz, Tremolo, Delay, Wah-wah, Flanger

## Fuzz Distortion

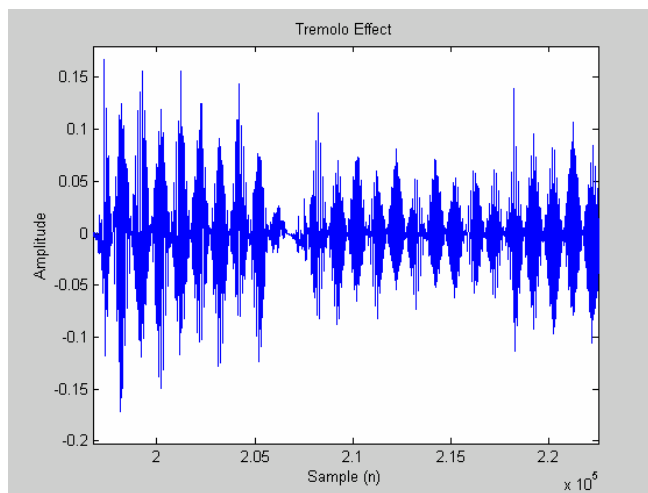


- All inputs above or below a threshold are limited

## Fuzz Distortion



## Tremolo



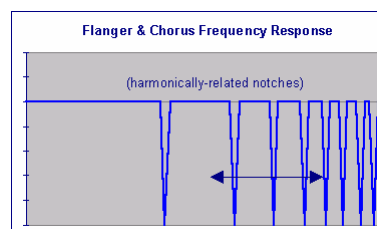
- Signal is masked by a low frequency oscillator

## 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

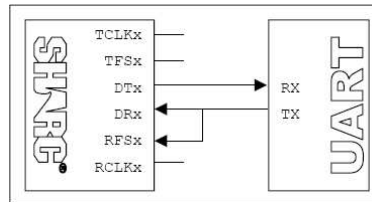


Fig 1

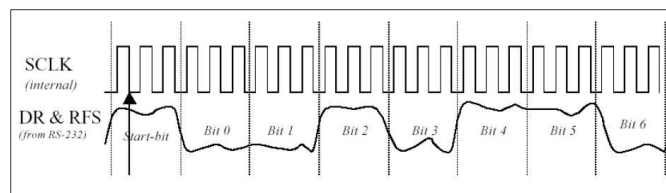


Fig 2

## 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

The screenshot shows the 'Advanced Effects Configuration' window with several effects modules. Annotations point to various controls:

- Control of switches and pedals:** Points to 'Switch Control' and 'Pedal Control' tabs at the top.
- Configure GUI:** Points to the overall layout of the effect modules.
- Set effect parameters:** Points to sliders for 'Clipping Threshold', 'Level Boost', 'Style', 'Rate', 'Depth', and 'LFO'.
- Set effect status:** Points to 'On / Off' radio buttons for each effect.
- Display current device:** Points to the 'Current MIDI Output Device:' section at the bottom right.

A 'Choose MIDI output device' dialog box is open, showing a list of MIDI devices including 'Microsoft MIDI Mapper, Windows MIDI\_MAPPER' and 'Java Sound Synthesizer, Software wearable synthesizer and receiver'.

# 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](http://ohm.nuigalway.ie/02omalley/fyp_0506)