Progress Report

Advanced Digital Guitar Effects System

Ronan O'Malley

02007894

Final Year Electronic and Computer Engineering

Supervisor: Dr Martin Glavin

Contents

Table of Figures

Abstract

This project describes a unique wireless digital effects system for guitarists, which is suitable for use in live performances. The system will allow a guitar player to easily configure effects in advance from a PC, a feature that sets this system aside from conventional effects systems that use dials and buttons. The wireless pedal board MIDI device provides real-time adjustment and selection of effects from an extensive effect database.

1. Introduction

The aim of this project is to research and develop a digital effects system with wireless MIDI control for the electric guitar.

The first stage of the project is to confirm existing effects theory with Matlab, and develop unique algorithms that can be ported to a DSP board without extended alteration. A set of variable parameters with boundaries for each effect has been established to ensure easy manipulation for users who have no underlying knowledge of the algorithm.

The second phase of this project consists of a pedal board for real-time control. The 8051 based control system utilises existing MIDI (Music Instrument Data Interface) protocol. The DSP system is controlled in two ways using this technology:

- Static on/off data sent from foot-switches, indicating when an effect should be enabled or disabled in real-time.
- Dynamic positional data sent from a variable pedal, this can alter an effect parameter in real-time and makes for a versatile performing environment.

A link will be created, using serial or I^2C protocol, to facilitate communication from the pedal-board to the DSP.

Upon receiving the necessary equipment, effects developed in Matlab and other effects will be re-written in C and embedded on an Analog Devices ADSP-21364 Digital Signal Processor. A further operating system and Midi-message decoder will be implemented on the processor.

The final element of the project is to create a serial link from the DSP to a PC, where a GUI will be constructed to enable fast and flexible reconfiguration of effects, and storage of set “scenes” which can be recalled at the push of a footswitch.

Time permitting the links between system elements will be converted to wireless communication using RF transceivers.

It is intended to conduct further research in the area of advanced effect control using some kind of on-guitar lever, joystick or sensor, in the view to apply for a patent and to document the entire project in the form of a formal technical paper, suitable for submission to a conference.

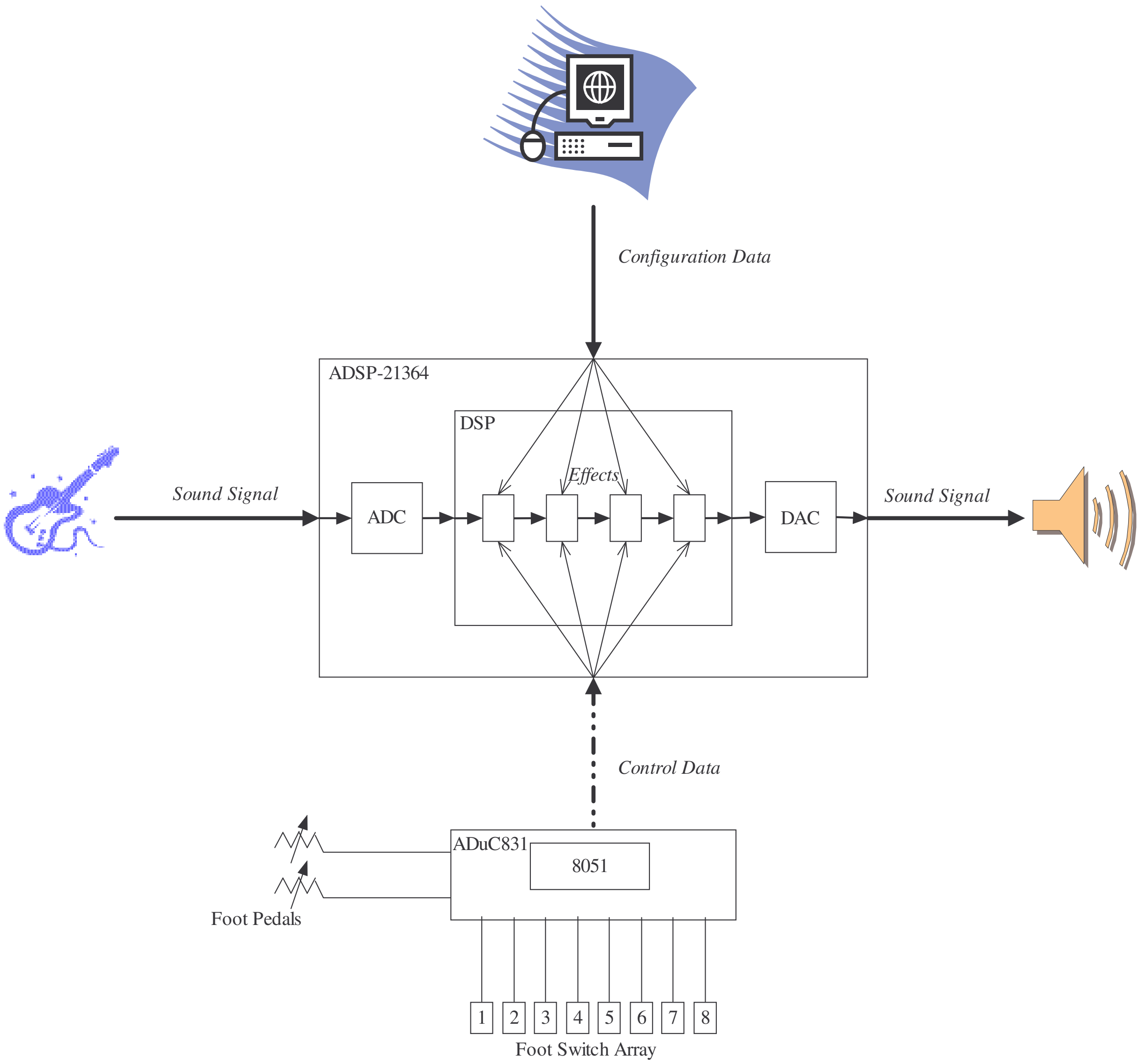

Figure 1. Full System Block Diagram

2. Project goals achieved

2.1 Matlab effect theory

A range of effects have been coded using Matlab. Samples for effect experimentation were recorded using a Peavey Raptor electric guitar and Audacity open source recording software. The Matlab programs were then executed on the 48 kHz 16 bit wav files, and results observed. Efforts have been made to standardise a set number of variable parameters for each effect, and establish clear boundaries. This has been achieved to a certain degree; however, all final effect modification will take place when the effects are embedded.

2.1.1 Fuzz Distortion

This effect is also known as symmetrical clipping or the 'Hendrix Effect'. The effect is a harsh distortion effect, any sample above or below a threshold is limited to that amplitude. The signal is then amplified to compensate for the loss of signal magnitude.

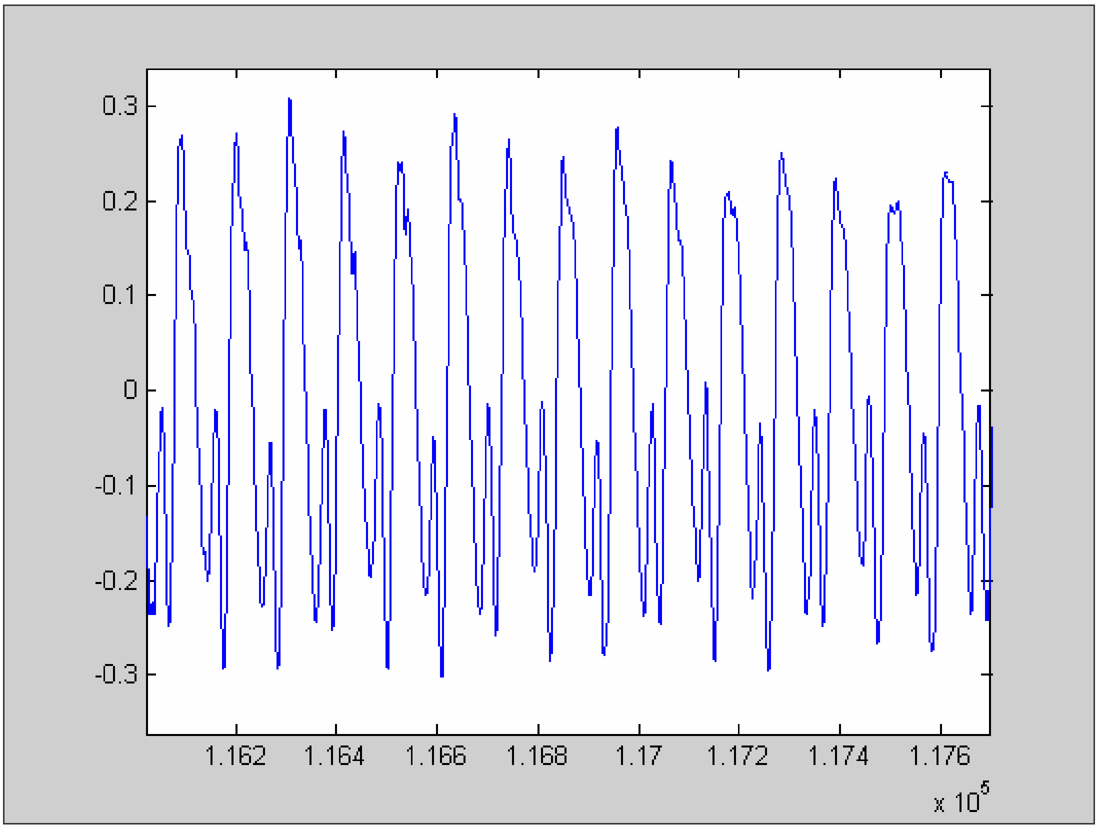

Figure 2. Sound wave before clipping is applied

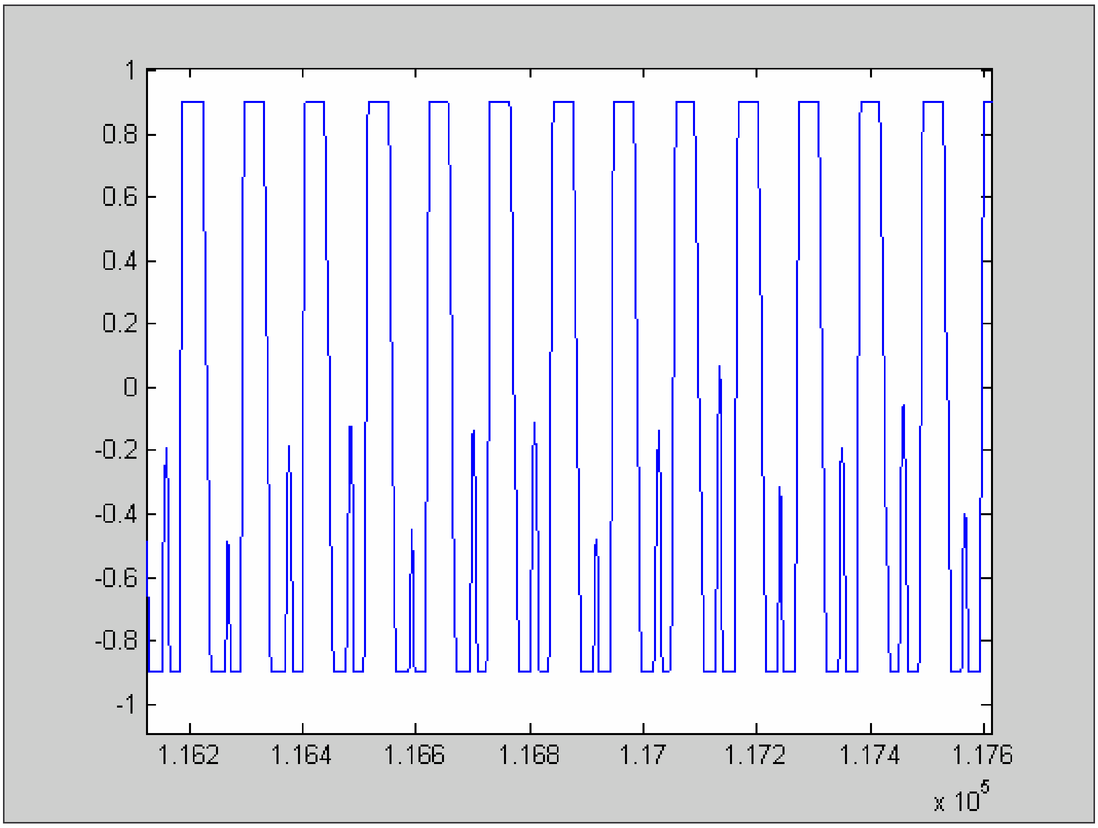

Figure 3. Sound wave after clipping is applied

The adjustable parameters for this effect are:

- The clipping threshold
- The amplification co-efficient

If different thresholds are used for positive and negative clipping then the effect is asymmetrical clipping. In experiments to date no major audible differences have been noted between the two techniques.

2.1.2 Tremolo

A tremolo effect is produced by applying a low frequency oscillating mask to the incoming signal. Triangle-wave oscillator and sine-wave oscillator variants were coded, but only the triangle variant will be embedded in the DSP board due to the superior results produced. To extend this effect a square wave version may be introduced.

The following parameters and boundaries were established through experimentation with Matlab:

- Frequency of oscillator – 2.4Hz to 24Hz
- Wave shape of oscillator – Triangle is best

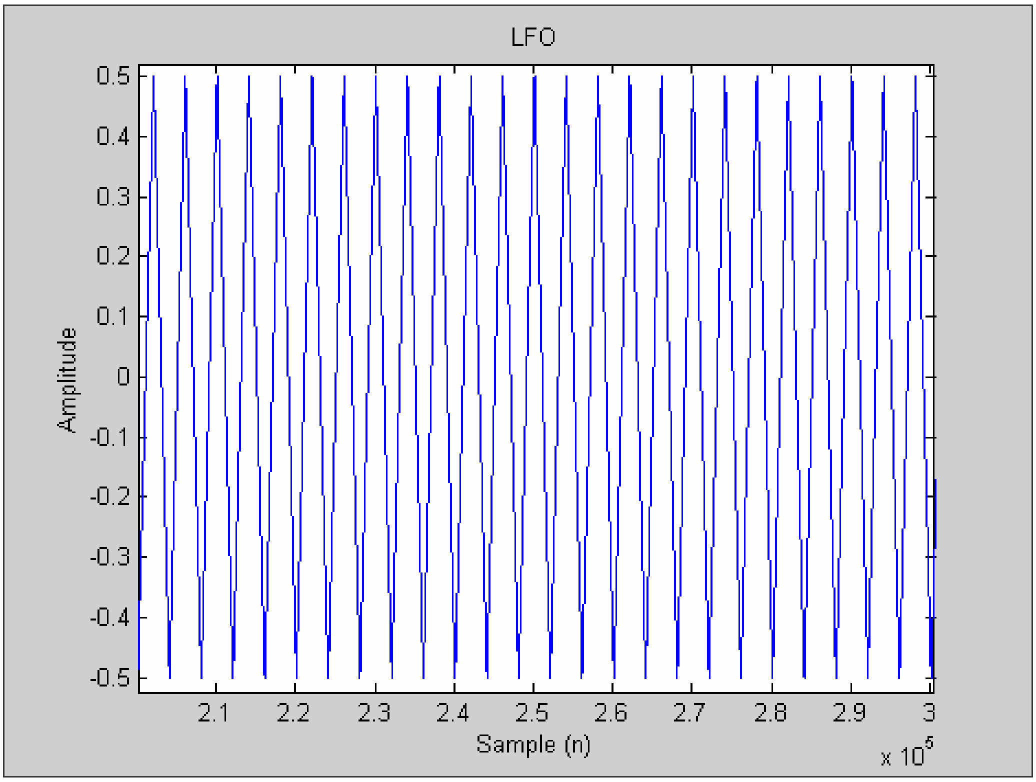

Figure 4. Low frequency oscillator used for tremolo

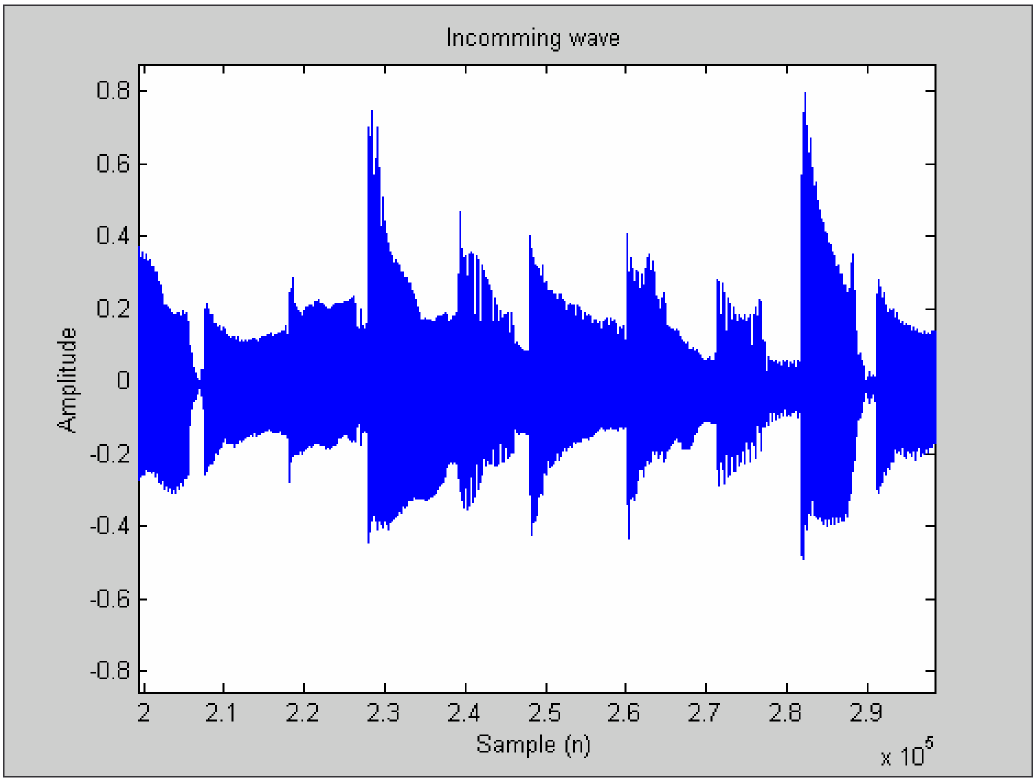

Figure 5. Incoming guitar signal

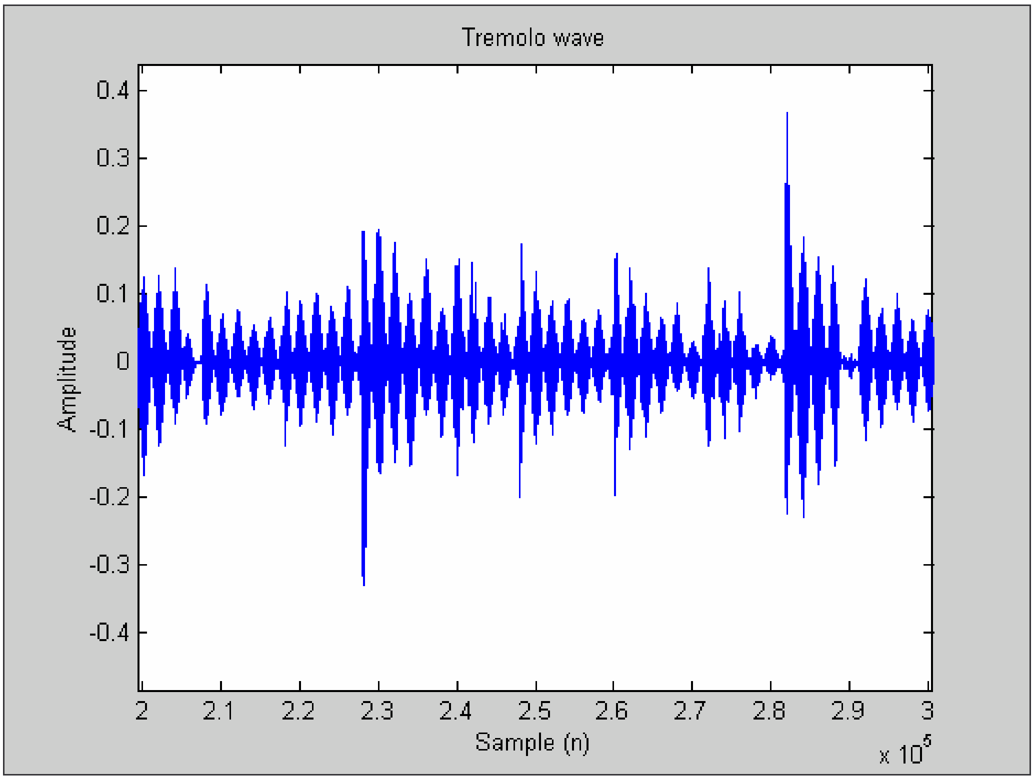

Figure 6. Guitar signal after Tremolo effect applied

2.1.3 Delay

This effect is created by adding delayed diminished samples to the incoming signal. The following parameters were established to vary the effect:

- Time between delays.
- Diminishing rate - this determines the amplitude of the next delay as a function of the current sample.
- No of delays – this is an optional constraint. Generally this would be determined by the diminishing rate reducing the delay amplitude to zero.

Varying these parameters can produce a number of different effects. For example, one set of parameter boundaries can produce a traditional ‘delay’ effect; another will produce an ‘echo’ effect.

Delay range (ms)	Modulation	Effect Name
0 – 20	-	Resonator
0 – 15	Sinusoidal	Flanging
10 – 25	Random	Chorus
25 - 50	-	Slapback
>50	-	Echo

Figure 7 Typical Delay Based Effects, DAFX [1], Section 3.3.2

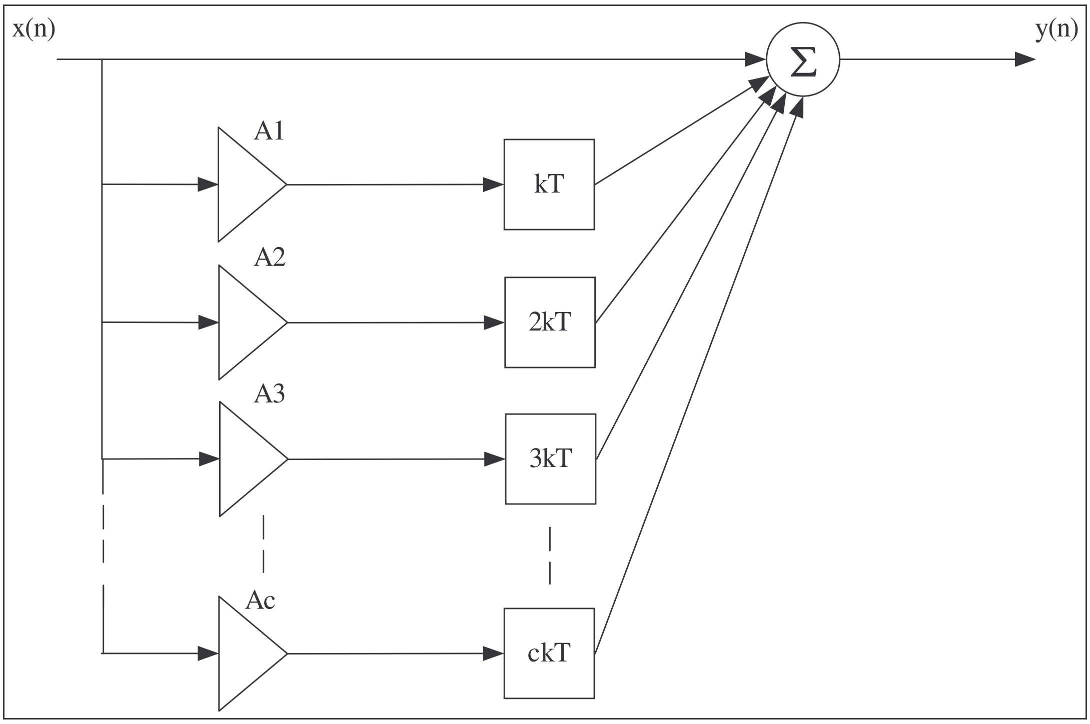

Figure 8 Block Diagram of Delay Effect

$y(n) = x(n) + A_1x(n-k) + A_2x(n-2k) + A_3x(n-3k) + \ldots + A_cx(n-ck)$

c is the number of delays.

k is the number of samples between delays. This is a function of the time between delays

t (usually in ms)

f sampling frequency

Eg. For a delay of 20ms and a sampling frequency of 48 kHz

$t = 20*10^{-3}$

$f = 48000$

$k = t*f = (20*10^{-3})*48000 = 960$ *samples between each delay*

If *k* were not a whole number it would have to be rounded to the nearest sample.

2.1.4 Wah-wah

A wah-wah effect was reproduced in Matlab by constructing a peak filter, and oscillating the filter's centre frequency through a set frequency range. A peak filter is a band-pass filter with a narrow pass band. Through experimentation, aided by some knowledge of the frequency content of a guitar sound, a parameter range of 500 Hz – 5 kHz was established for the filter's centre frequency.

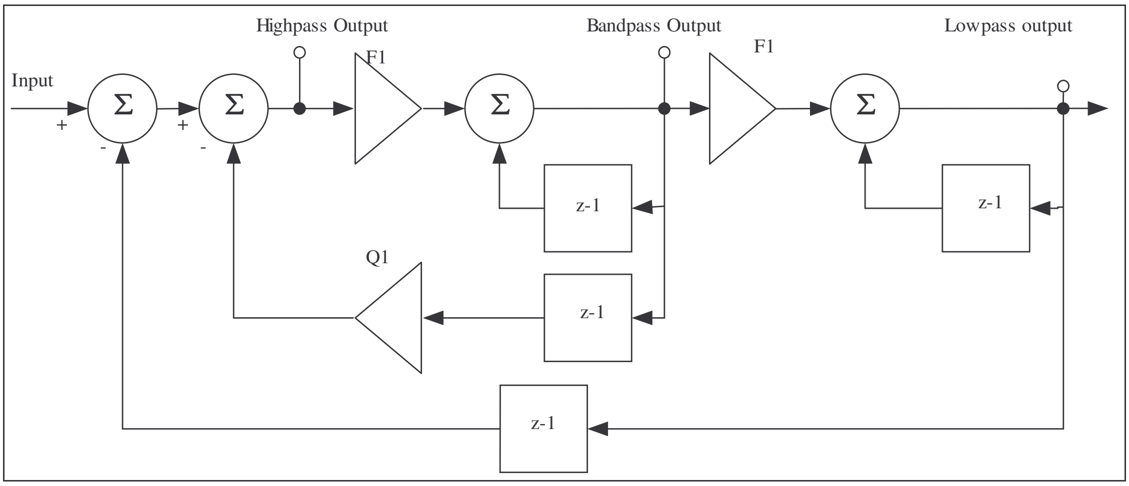

Figure 9. State variable digital filter from DAFX [1], chapter 2

$y_{lp}(n) = F_1 y_{bp}(n) + y_{hp}(n-1)$

$y_{bp}(n) = F_1 y_{hp}(n) + y_{hp}(n-1)$

$y_{hp}(n) = x(n) - y_{lp}(n-1) - Q_1 y_{bp}(n-1)$

$F_1 = 2\sin(\pi f_c/f_s)$

$Q_1 = 2\zeta$

The variables associated with this effect will be

- the filter's damping ratio (ζ)
- filter's centre frequency (this is dynamically controlled by a variable foot pedal, creating the "wah" sound that gives the effect its name.)

In a later stage of this project a version of this effect will be created that uses a notch filter instead of a peak filter. This will produce a "phaser" effect.

2.1.5 Flanger

A flanger effect is constructed of a single delay, where the delay time oscillates within a short range at a low frequency. The single delay oscillates slowly between a maximum and minimum delay time. Parameter boundaries for this effect, were taken from DAFX [1], Section 3.3.2.

Effect variables:

- Delay range (0 to 3ms)
- Oscillator (Sin wave, triangle wave)
- Oscillating frequency (1 Hz)
- Amplitude of the delay (70% of the input sample)

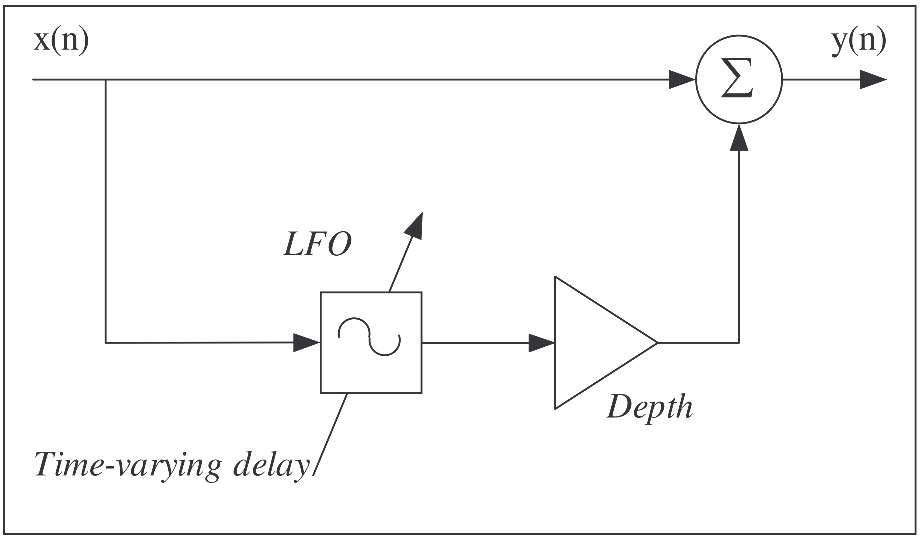

Figure 10. Flanger system block diagram

2.2 Microcontroller Operating System

An operating system was created to obtain data from the pedal board input equipment, convert the data into MIDI message format and transmit to the DSP unit.

The 8-bit assembly code system was embedded in an 8051 based microcontroller, within an Analog Devices ADuC831 development board. The operating system firstly polls two variable foot pedals, these are intended to control volume and the centre frequency of the wah-wah filter. The last transmitted value of each pedal is stored in a buffer. If the pedal's value has changed since the previous transmission, then the new value is converted into MIDI format, and three MIDI messages are sent. These messages indicate there is a control change, the controller number of the pedal that has changed and that pedals new value. The 831's on board ADCs produce a twelve bit result. To conform to MIDI protocol this result is concatenated leaving the seven most significant bits to determine the pedals position, which provides a sufficiently accurate resolution.

A similar setup is employed for the switches that turn the effects on and off. Buffers hold values of the most recent switch value transmissions. If the value of a switch has changed, three MIDI messages are transmitted corresponding to the variable pedal messages. The DSP board will receive these messages, decode them and apply the corresponding changes to the embedded effects.

It is intended to extend this system by embedding a 'scene selection' system. This will allow the guitarist to store groups of effect configurations ('patches' or 'scenes'), and recall them at the push of a switch. This functionality will use the 'program change' midi message format and allow advanced effect changes, suitable for live performance.

A circuit has been constructed to ensure a steady reference voltage from the board. This is linked to two potentiometers which act as variable foot pedals. The steady reference voltage is essential so that when the pedal is still, the data read on the ADC is the same for each consecutive poll. If this were not the case the operating system would interpret the fluctuating readings as a change in pedal position and transmit MIDI messages.

Function	Midi Message	Definition
Control Change	1011 nnnn	Midi Channel (1 – 16)
	0ccc cccc	Controller Number (0 – 127)
	0vvv vvvv	Control Value (0 – 127)
Program Change	1100 nnnn	Midi Channel (1 - 16)
	0ppp pppp	Program Number (1 – 16)

Figure 11. Midi messages used in pedal-board operating system [2]

MIDI channel	Binary Value	Assignment
1	0000	Variable pedals
		Controller 0 (0000) – pedal 1
		Controller 1 (0001) – pedal 2
16	1111	Foot Switches
		Controller 0 (0000) – switch 1
		Controller 1 (0001) – switch 2
		Controller 2 (0010) – switch 3
		Controller 3 (0011) – switch 4
		Controller 4 (0100) – switch 5
		Controller 5 (0101) – switch 6
`		Controller 6 (0110) – switch 7
		Controller 7 (0111) – switch 8

Figure 12. Table of MIDI channel assignments for pedal board system

Examples of MIDI messages transmitted by the system:

Pedal 1 has a new value of 0xC35 = 1100 0011 010, 7 MSB taken => 110 0001

MIDI message constructed

1011 **0000 –** Control change MIDI channel one (variable pedals).

0**000 0000 –** Controller 0 (pedal 1).

0**110 0001** – New value.

2.3 Project Website

An extensive project website is available for viewing at http://ohm.nuigalway.ie/02omalley/fyp_0506/

This website was completed on 28 October 2005, and consists of over 10 pages of project information, latest news, updates and links to all related documents and code.

Figure 13. Project website screenshot

3. Remaining project goals

3.1 Embedding of Guitar Effects

All guitar effects outlined in section 2.1 will be re-written in C and embedded in an Analog Devices SHARC processor. It is intended to build on the array of effects already created and have an extensive library of popular effects.

The ADSP-21364 is aimed specifically at professional audio processing and so is perfect for this task. The development board is well equipped with audio-in and audio-out RCA jacks enabling direct connection of guitar and amplifier using standard leads. It also boasts a USB JTAG interface allowing easy device programming and onboard flash to provide non-volatile storage of effect configurations.

"The ADSP-21364 is a high performance 32-bit/40-bit floating-point processor optimized for professional audio processing. At 333 MHz/2 GFLOPs, with unique audio centric peripherals such as the digital audio interface that includes a high-precision 8-channel asynchronous sample rate converter among others, the ADSP-21364 SHARC processor is ideal for applications that require industry leading equalization, reverberation and other effects processing." [3]

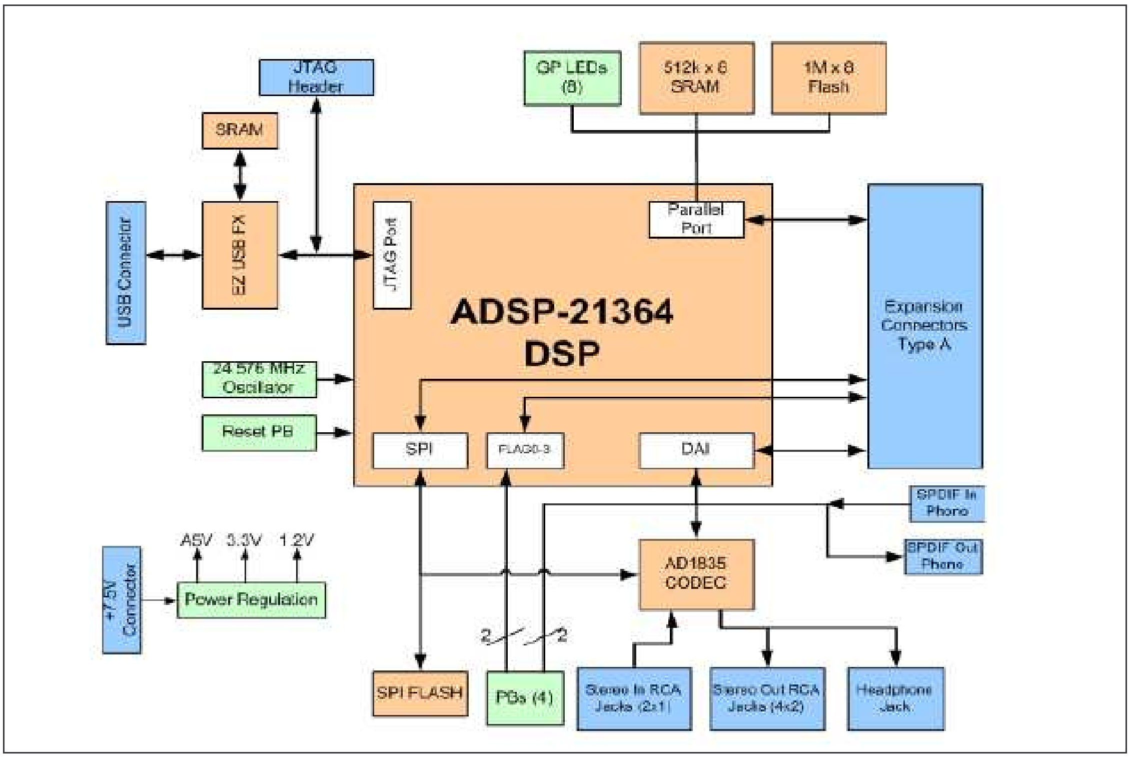

Figure 14. Block diagram of ADSP-21364 Architecture [4]

The following effects will also be coded in embedded C and added to the effect library time permitting.

Chorus

An extension of the flanger effect, with multiple oscillating delays and a longer delay range.

Non-linear distortion (Tube amp emulator)

This is achieved by applying a non-linear transfer function to emulate the sound produced by a tube-valve amplifier. This produces a sweeter, more melodic distortion, less harsh than the fuzz effect.

Compression

A compressor is a device where the gain given to the output depends on the level of the input. It essentially makes quiet sounds louder and louder sounds quieter.

Octave

This effect shifts the incoming signal up or down one or two octaves on the musical scale and mixes the result with the original signal. The result sounds like two guitars playing a similar tune in unison.

Sampler

It is intended to take advantage of the ADSP-21364's on board SRAM to allow the guitarist to record samples and set them on loops.

Reverb

This is a tricky effect that emulates playing in a room. The time it takes the original signal to bounce off each wall must be considered, as well as the amount of sound that the wall absorbs.

Phaser

As described in section 2.1.4, this effect formed by replacing the peak filter used in the wah-wah effect with a notch filter.

An operating system will be embedded around these effects, to decode and interpret MIDI messages received from the foot-pedal and serial data from the PC.

3.2 Pedal-board to DSP link

The 8051 pedal-board will transmit its MIDI messages over a serial link to the DSP effect processor. To conform to MIDI protocol the link must have a baud rate of 31.25 kbps. This poses several challenges. The ADuC831 board's serial port interface set's the baud rate by dividing the board's core clock. As the board is equipped with an 11.0592 MHz crystal as standard, it is not possible, through any combination of divider coefficients, to achieve the desired baud rate. Therefore the board's crystal must be replaced with a 16 MHz oscillator.

Using the following equations, outlined in the ADuC831 datasheet [5], the divider coefficient DIV, and the value of special function register T3FD can be calculated.

$$DIV = \frac{\log(\frac{f_{CORE}}{32 \times BaudRate})}{\log(2)} \qquad (1)$$

$$ActualBaud = \frac{2 \times f_{CORE}}{2^{DIV} \times (T3FD \times 64)} \qquad (2)$$

$$T3FD = \frac{2 \times f_{CORE}}{2^{DIV} \times BaudRate} \qquad (3)$$

$f_{core} = 16 * 10^6$

baud = 31250 bps

⇨ DIV = 4

⇨ T3FD = 0x40

⇨ Actual baud = 15625 = 31350/2

Equations (1), (2) and (3) are sourced from the ADuC831 data sheet, [5].

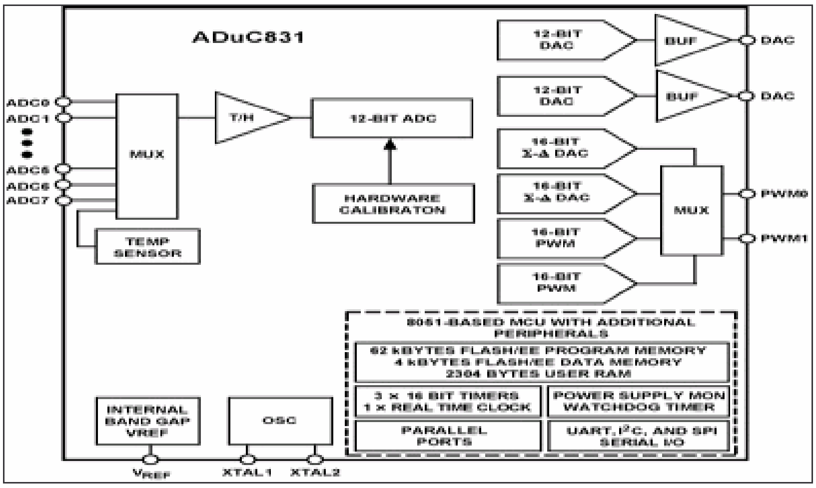

Figure 15. ADuC831 Functional Block Diagram [5]

3.3 PC to DSP Serial Connection

To enable superior effect and patch configuration, a PC program will send configuration data to the DSP board. This will be done via a serial link. This link is not intended to conform to midi protocol, and will not be designed to provide real-time enhancement to guitar playing. Its sole purpose is to configure the device before use. The serial link can therefore use one of the ADSP-21364's serial ports, and connect directly to the PC's serial port.

3.4 PC GUI

A program with a graphical user interface will be coded in Java. Users will make their alterations to the effect setup, and these changes will be sent over the serial link to the DSP. This program will utilise the 'javax.comm' package of the Java API, which provides the methods and classes for dealing with serial port communications. The java code will define how the setup configuration data is serialised when transmitted. This will allow an instruction decoder to be embedded into the DSP operating system.

3.5 Pedal-board construction

While two circuits have been built to simulate two foot pedals and eight foot switches, it is desirable, at some point, to be able to provide a live project demonstration. This may only be fully achieved through the use of foot controlled devices. The present configuration uses potentiometers and switches that have to be switched by hand. The potentiometers will be linked to foot pedals through a 'rack and pinion' cog wheel configuration, and the hand switches will be replaced by foot switches.

3.6 Wireless Link

It is desirable to have a wireless link between the pedal board and the DSP processor. This would allow a certain amount of freedom when playing, and positioning equipment. Once a wired serial link has been successfully tested, an RF link will be put in its place, keeping to the same baud rate so that the adherence to MIDI protocol is upheld and buffering is not necessary at either side.

As the project stands, it is felt that this project objective is of low priority and the project as a whole may benefit more by applying the remaining time to further research in the area of advanced hand control of effects (see Section 3.7).

3.7 Further research – advanced hand control of effects

It is intended to devote some of the remaining project time to the research in the area of advanced control of guitar effects using some kind of hand device. This is an area that has much potential, as convention clearly favours foot controlled equipment. However this approach is severely limited, more advanced control could theoretically lead to whole new ways of playing the electric guitar.

Initial brainstorming has produced the idea of some format of touch sensor mounted on the face of the guitar. This would be controlled by the strumming or picking hand (right hand for right handed players). One way to approach this idea would be to manufacture the guitar with this sensor built in. This would ease the problems of data transfer and power connections, however it is fundamentally flawed. An alternative approach could appeal to a much broader market. If the touch-sensor was a stand-alone device, with no power supply and no wires for data transfer, any guitar owner could install it on the face of their guitar with zero overhead.

For this approach to be by any means feasible, it is intended to study that area of RFID chips. RFID chips contain no power supply, a reading devices bounces a signal off it a received its embedded data back.

If this idea could be altered, the pedal board could act as the reading device and poll the touch sensor for its current position. This would mean no replacing of batteries, routing of wires or bulky on board transceiver. It would also consist of a very simple installation procedure.

If, after extensive research, this approach is deemed viable, it could be applied to many other areas of remote control.

4. Possible Challenges and Solutions.

Time Budget

Due to emphasis on further research (Section 3.7), the plan to incorporate a wireless link may have to be revised to meet the project deadline.

Midi baud-rate

The baud rate of the MIDI standard is proving to be very difficult. Replacing the oscillator on the ADuC831 board may cause complications in the serial link. I^2C may be a viable work-around.

5. Project Objective Timeline

Goal	Target Date	Length of Time
Embed all effects in DSP	1st February 2006	23 days
Serial 8051 to DSP link	13th February 2006	12 days
PC GUI	16th February 2006	3 days
PC to DSP Link	1st March 2006	15 days
Wireless Link / Further Research	12th March 2006	11 days
Final Report	22nd March 2006	10 days

Figure 16. Project Objective Timeline

Appendix I – Table of References

[1] Udo Zölzer, *DAFX Digital Audio Effects*, Chichester, England, Wiley, 2002.

[2] Summary of MIDI messages, http://www.midi.org/about-midi/table1.shtml

[3] ADSP-21364 SHARC® Processor Preliminary Data Sheet, Analog Devices 110075906ADSP_21364_prd.pdf
http://www.analog.com/UploadedFiles/Data_Sheets/71670759ADSP_21364_0.pdf

[4] ADSP-21364 EZ-KIT Lite Evaluation System Manual, Analog Devices
http://www.analog.com/UploadedFiles/
Associated_Docs/50400613ADSP_21364_EZ_KIT_Lite_Manual_Rev_2.0.pdf

[5] ADuC831 data sheet, Analog Devices
http://www.analog.com/UploadedFiles/Data_Sheets/90530111ADuC831_0.pdf

Appendix II - Table of Acronyms

RTOS	Real-Time Operating System
DAFX	Digital Audio Effects [1]
DSP	Digital Signal Processor/Processing
MIDI	Musical Instruments Digital Interface [2]
ADuC	Analog Devices Micro-Controller
ADC	Analog to Digital Converter
MSB	Most Significant Bits
RCA	Radio Corporation of America
USB	Universal Serial Bus
JTAG	Joint Test Action Group
kbps	Kilo-Bits per second
bps	Bits per second
PC	Personal Computer
GUI	Graphical User Interface
API	Application Programming Interface
RFID	Radio Frequency IDentification
LFO	Low Frequency Oscillator
SHARC	Super Harvard ARchitecture Computer
ADSP	Analog Devices Digital Signal Processor
GFLOP	Giga-FLoating-point OPerations