Audio Effect Pedal

Milestone: *Critical Design Review*

Douglas Brock, Jonathon Miller, Stephen Piechowicz

Methods and Technical Approach

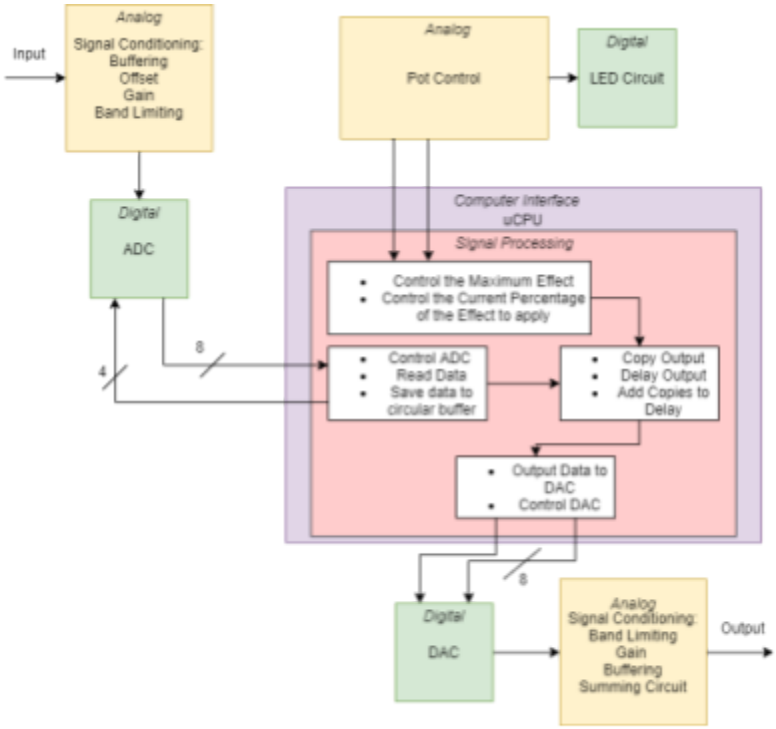

Figure 1: Block Diagram of the Audio Effect Pedal

Figure 1 shows a block diagram of the Audio Effect Pedal. The different shaded backgrounds are indicative of which of the four essential components to the project. Analog, Digital, Interface to a microcontroller, and data collection with signal processing. An overview of each of these components is described below.

Digital Design: The digital components of this project involve making a potentiometer control for the microprocessor, and an LED display for the data of the signal intensity. The LED display will be connected to ladder circuits that will control which LEDs are lit up, allowing the user to see the level at which they have their effect set to. The separate LED display will show the maximum intensity of the effect, and the percentage of the maximum that is currently being utilized. The ADC and DAC will be controlled by code. A GPIO pin will send a pulse to the ADC and DAC signaling to collect data and to process data. After the ADC has a sample ready, the uCPU will collect the data. With 1.5kHz as the upper bound of frequency, the ADC will need to sample the input at 3.5kHz. Both the ADC and DAC will be 8 bit. The DAC will output a signal on a pulse from the uCPU.

Analog Design: For the analog design, the input stage impedance will need to be matched to the guitar's impedance. The signal will need to be offset to

accommodate the ADC taking positive values only. The signal may need to be boosted as some guitars output 100mV - 400mV. Though the normal guitar output frequencies are 80hz to 1.2Hhz, it would be better to provide a bit of a wider range to preserve the original signal so we will pass the frequency 60Hz to 1.5kHz. The DAC's output will have to be filtered as well. A slight gain is also applied to the output signal. After the delayed signal goes through DAC, the delayed signal will go into a summing circuit and combine with the original signal.

Computer Interface: The microcontroller (μCPU) that we have settled on is the STM32 Nucleo F446RE μCPU. This μCPU is similar to an arduino; however, it has better specifications for real-time audio processing with minimal time delay which is what this project focuses on. The μCPU interfaces with three main components within the build; the ADC, the DAC, and the potentiometer controls for the signal. The interface to the ADC is a four pin control mechanism alongside an 8-bit stream which contains the data. For the DAC, the interface is a one pin control signal, alongside an 8-bit stream containing the output data. Finally, the interfaces to the potentiometers are the pins connected to the internal ADCs of the μCPU. One of these potentiometers will contain the signal for the maximum time delay the μCPU applies to the signal, and the other contains a signal telling the μCPU how much of that maximum delay to use.

Data Collection with Signal Processing Elements: A circular buffer is used to read in and store up to the 5000 most recent 8-bit data points from the ADC as uint_8 objects, sampled 3500 times a second, which is more than twice the maximum frequency going into the ADC. The buffer then sends the data through the main signal processing algorithm. This algorithm will delay the signal by time t, copy the values into a second circular buffer at a decreased intensity, then combine the values taken from the buffer with values from t seconds later to run through the algorithm again. The two internal ADCs for the pot controls will sample at the same rate as the external ADC. They will have that data put into a known location, and immediately used to affect the delay.

Detailed Design:

The following section provides schematics of the digital, analog, and interface to the microcontroller components of the audio effect pedal. This will include the flowchart explaining the data collection/signal processing elements.

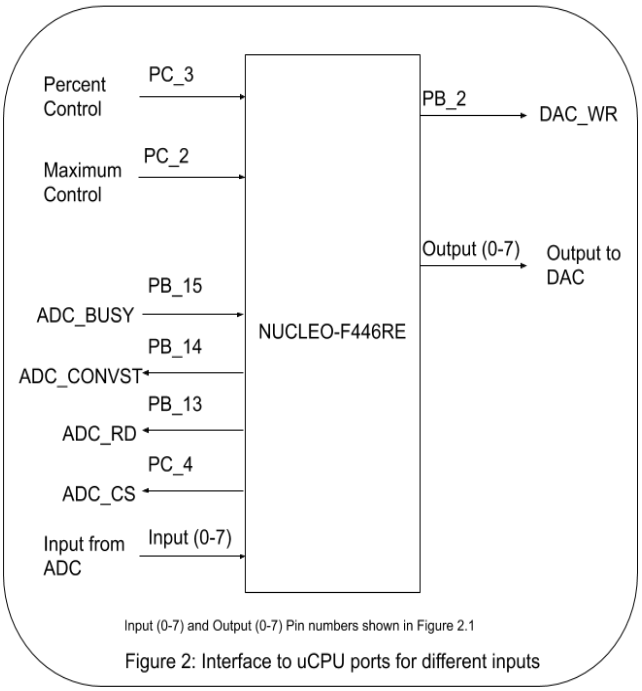

Figure 2: Interface to uCPU ports for different inputs

Computer Interface:

Figures 2 contains a schematic of the NUCLEO-F446RE microcontroller, including pin locations, and signal definitions for devices connecting to the µCPU: an 8-bit parallel to the ADC, inputs to the built-in ADCs from pot controls, shown by percent control and maximum control, signal to the ADC alongside the 8-bit parallel port, and utilization of the GPIO ports. Figure 2.1 more accurately outlines which pins are represented by the inputs from the ADC and outputs to the DAC

Input	Pin# (Input)	Output	Pin# (Output)
0	PA_3	0	PA_13
1	PA_2	1	PA_14
2	PA_10	2	PA_15
3	PB_3	3	PB_7
4	PB_5	4	PC_13

5	PB_4	5	PC_14
6	PB_10	6	PC_15
7	PA_8	7	PH_0

Figure 2.1: Pin number configurations for Input(0-7) and Output(0-7)

Data Collection/Signal Processing

Figure 3 contains a flow chart identifying key steps in the data collection/signal processing program on the μCPU: reading and storing the gathered 8-bit samples from the GPIO ports into the circular buffer, input of the circular buffer into time delay functions, and a control algorithm (modeled in Figure 4) for the time delay functions which is modified by inputs from the built in ADCs. There is a central process which is where most of the signal processing takes place. This process is boxed off in the flowchart as its own section, while most of what is outside of this process is data collection, and outputs. This central process takes the collected

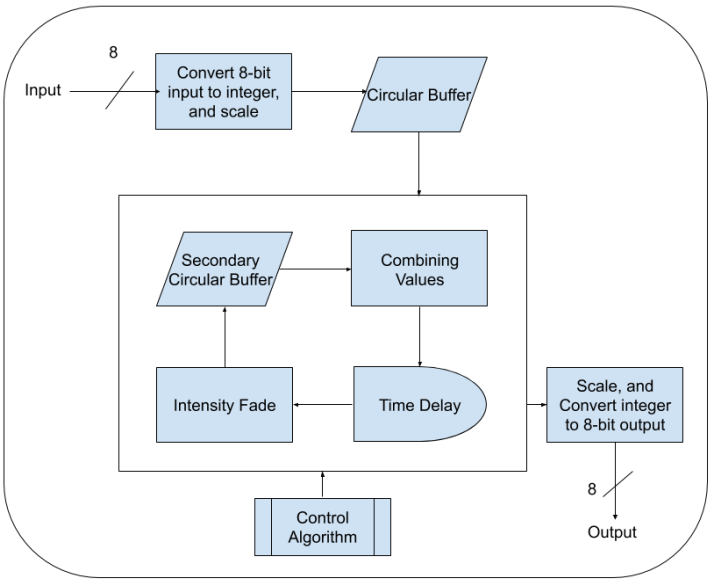

Figure 3: Main algorithm for Data Collection/ Signal Processing

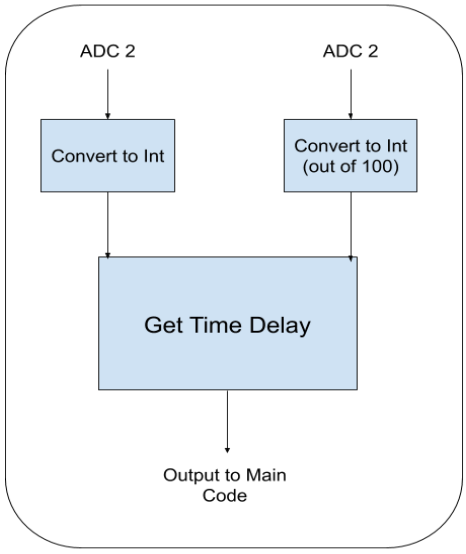

Figure 4: Control Algorithm

data, delays it, fades the intensity, and puts the values into another circular buffer. The values in the circular buffer will be combined with later values, which will go through the same process.

Analog Design

Figure 5 shows the input circuit. The initial input is passed through a buffering op amp and provides some of the gain needed to get the signal to fill the ADC's operating range. The band pass filter consists of two filters, a high pass filter and a low pass filter. Each filter is separated by an op amp. The op amp separating each stage provides a gain for the signal. The gain of each op amp was designed so that the highest input from the guitar gets a final peak to peak value of 4.8 V_{pp}. After the filter, the filtered signal goes through a summing circuit where it is combined with a 2.5V DC offset. The point of the off set is to put the signal in the center of the ADC operating voltage.

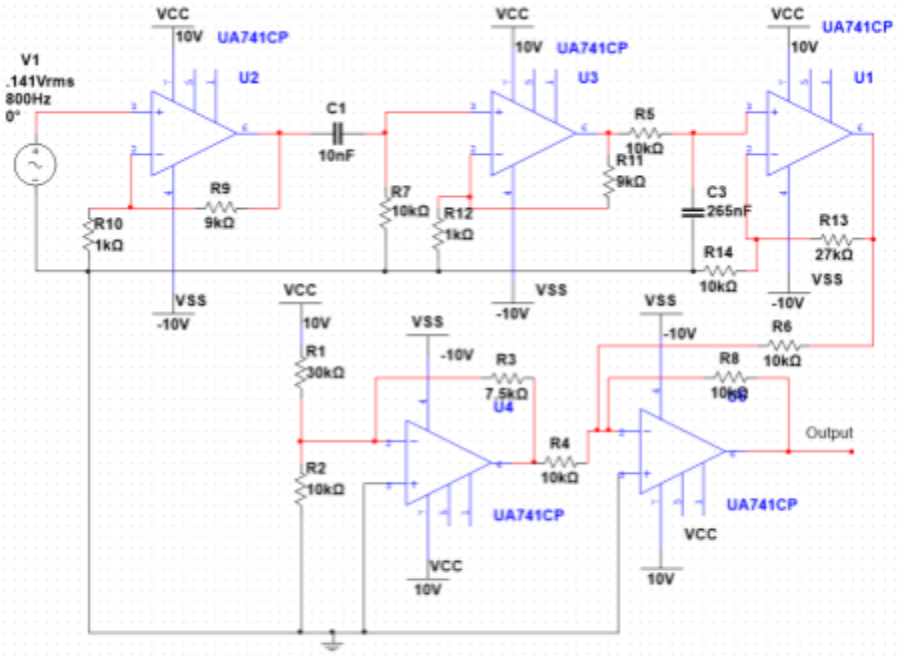

Figure 5: Input Circuit

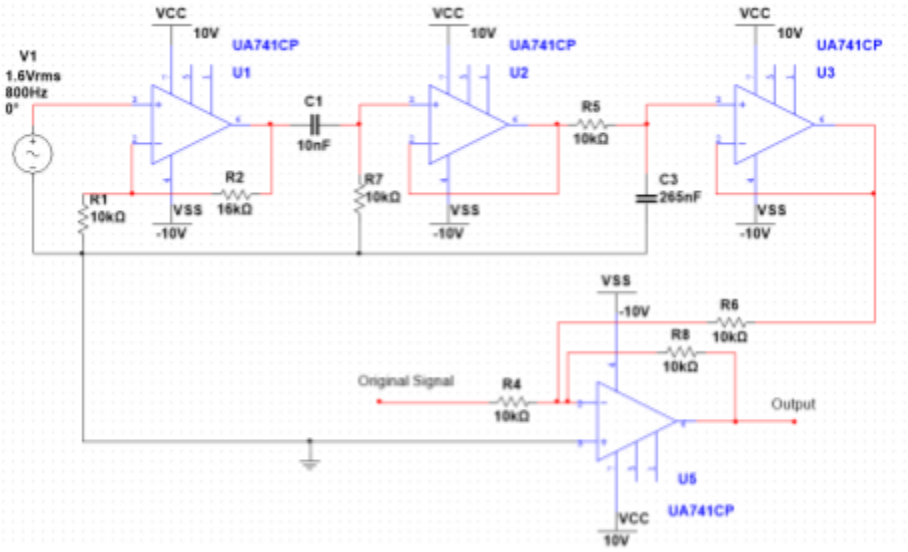

Figure 6: Output Circuit

The output circuit has a very similar design to the input circuit. The output circuit is shown in Figure 6. The filter has the same stages as the input. The gain of the output circuit is such that a signal of $4.8V_{pp}$ ends up being $0.399V_{pp}$ after passing through the filter. With the voltage going back $0.399V_{pp}$, the delayed signal can then be combined with the original signal through the summing circuit shown in the bottom of Figure 6.

The only conditioning for the potentiometer that needs to be considered is its output voltage. The three main reasons to condition a signal before sending the signal through an ADC can be avoided by matching the max the potentiometer can output to the maximum input voltage of the ADC. The first reason to condition a signal is to offset the signal so it is always positive; the potentiometer never has a negative voltage. The second reason to condition the signal is to provide a gain to the signal to get the signal to use the full range of the ADC; by setting the maximum output of the potentiometer to the maximum input of the ADC this conditioning is not needed. The final reason to condition the signal is to filter out high frequency components; the potentiometer is DC.

Digital Design

The next part of the analog design is the LED indicator circuit. The comparator circuit is shown below in Figure 7. The LED indicator circuit is a comparator circuit where only after a certain threshold voltage does an LED turn on. The circuit takes a reference voltage from the input (the magnitude of the effect), and compares that to the intensity of the effect. Through the comparator circuit, this results in a percentage output,

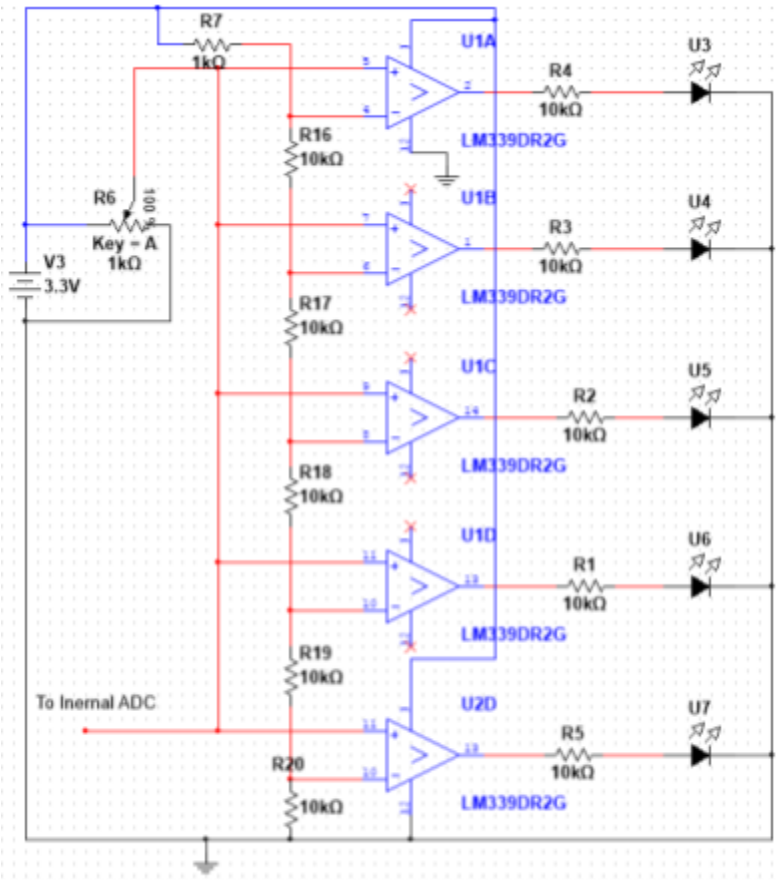

Figure 7: LED Circuit

which will light one to five LED lights corresponding to 20% increments.

The digital design revolves around the ADC and DAC. The control for the ADC is shown in Figure 8. The ADC's pins to start taking a sample are turned on by the uCPU. After the sample is complete, the ADC's sample ready pin will activate and the sample will be fed to the uCPU where it will be stored. The DAC goes through a similar process. The DAC's pin to start processing data gets activated. When the sample is processed, an analog signal is the output. The digital design for the potentiometer is pretty simple. The DAC's flowgraph is shown in Figure 9. The potentiometer will use an internal ADC.

Figure 8: Flow graph of ADC

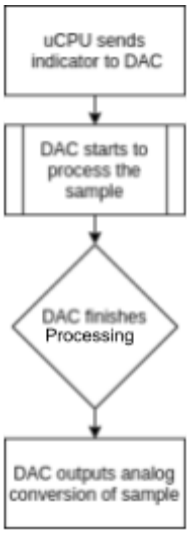

Figure 9: DAC flowgraph

The ADC being used is an Analog Devices Inc. part, AD7819. The DAC is the MAX5102.

Results and Discussion:

Currently there are no results available for the different project aspects for the Audio Effect Pedal.

Gantt:

Figure 10 shows the Gantt chart which is slightly behind in the digital simulation and digital testing as the next part of simulation is adding different components together. For the digital simulation, adding the ADC to the input circuit to the DAC to the output circuit. For the processing testing, we do not know how to test the code.

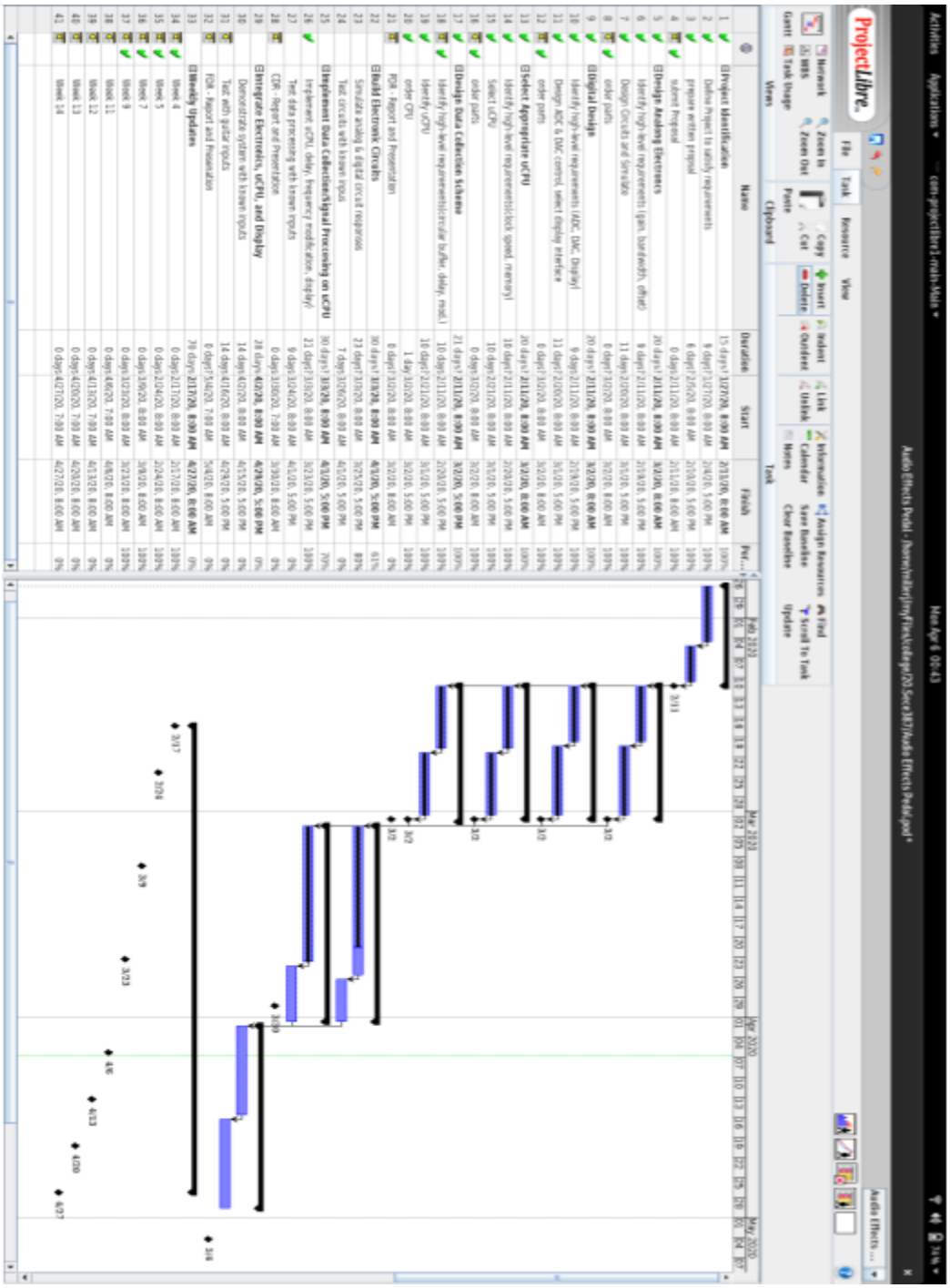

Figure 10:
Gantt Chart