MIDI Controlled Slide Guitar

Design Review

Angad Bector, Joel Spadin, Ruichen Zhao
TA: Mustafa Mukadam

March 4, 2013

Table of Contents

1.0 Introduction

1.1 Statement of Purpose

We chose this project because, while automated musical instruments have existed for a long time, few of these instruments respond quickly enough to be used in a live performance. In addition, few automated guitar instruments exist. We hope to create an instrument that provides musicians with interesting sounds and that can be performed using existing MIDI controllers.

1.2 Objectives

1.2.1 Goals:

- Enable control from any MIDI source
- Respond to input accurately and with minimal delay
- Output audio through a standard audio jack

1.2.2 Functions:

- Guitar string with motorized parts for pitch and volume control
- MIDI input enables performing with keyboards or other MIDI controllers
- Microcontroller applies configurable digital audio effects

1.2.3 Benefits:

- New and interesting sounds for musicians
- Works with existing MIDI and audio hardware
- Enables new methods of music creation
- Novel musical instrument usable for entertainment or education

1.2.4 Features:

- Controllable in real time or with recorded data
- Takes input from any instrument that generates MIDI data
- Configurable audio effects system

2.0 Design

2.1 Block Diagrams

Figure 1 shows the general setup for using the guitar as a performance instrument. The MIDI controller and speakers are generic components and are not included as part of the project. Figure 2 gives a high-level description of the guitar itself.

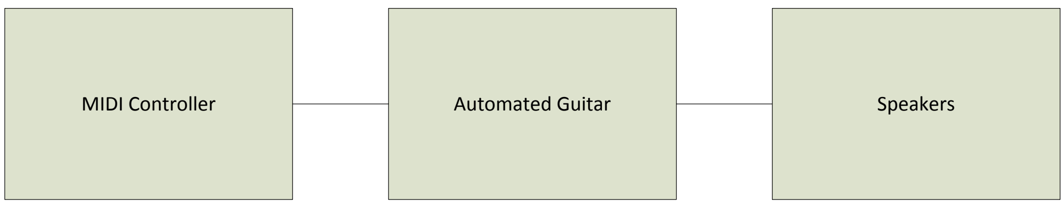

Figure 1. Performance setup

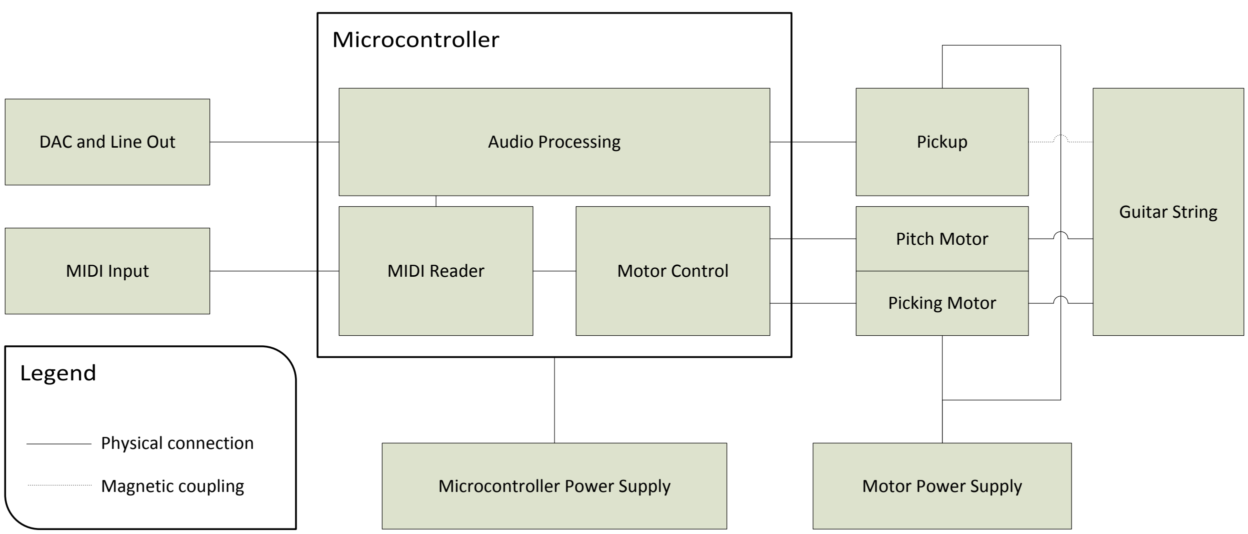

Figure 2. Top level system layout

2.2 Block Descriptions

2.2.1 MIDI Input

The MIDI input module is responsible for receiving MIDI data and transferring it to the microcontroller. It will contain a 5-pin DIN connector to connect a MIDI cable and will use an opto-isolator to condition the signal before sending it to the microcontroller's serial input. It will receive 5V power from the microcontroller.

2.2.2 Microcontroller

The microcontroller used will be a National Instruments Single-Board RIO. This processor is ideal for our application because it is fast enough to generate I^2S signals for audio output, has analog inputs for sampling the audio signal, and can be programmed using LabVIEW, which will greatly simplify software development. Two separate processes will run on the microcontroller—one responsible for translating MIDI input into motor commands, and the other for applying digital audio effects to the audio signal.

The motor controller process takes MIDI input and controls the excitation of the string. One sub-process takes the MIDI input and determines when and how hard to strike the string and where the pitch bar should be placed. A second sub-process takes these parameters and—with motor position data from encoders—runs closed-loop control of the pitch and picking motors. It should move the pitch bar to the desired position as quickly and accurately as possible in order to minimize the amount of delay the performer perceives.

The audio effect process reads the analog audio signal from the magnetic pickup using the NI board's ADC. It then applies effects such as reverberation, feedback, and distortion to the digital audio. It will also take in MIDI continuous controller data (volume, modulation, etc.) as parameters to these effects. It will output the audio using the I^2S protocol.

2.2.3 Microcontroller Power Supply

This power supply will convert power from standard AC power (120V AC, 60Hz) to 24V DC to power the microcontroller. We will use a standard 40W, +/-12V power supply supplied by the ECE shop.

2.2.4 Pitch Motor

The pitch motor module will consist of a motor, controller, and position sensor. It will move the pitch bar along the guitar string. We will use a DC motor and a belt system to move the pitch bar. The microcontroller will send a PWM signal to a Cytron Technologies MD10C motor controller to control the speed of the motor. A rotary encoder on the motor will send position data back to the microprocessor to allow it to perform closed-loop control. We will first implement this using a PID loop, and explore alternate control algorithms if we determine it to be inadequate.

2.2.5 Picking Motor

The picking motor will consist of a motor, controller, and position sensor. It will control a wheel of guitar picks that strike the string. The microcontroller will send a PWM signal to a MD10C motor controller, and position feedback will come from a rotary encoder. Control will be implemented as with the pitch motor.

2.2.6 Motor Power Supply

This power supply will convert power from standard AC power (120V AC, 60Hz) to 24V DC for the motor controllers. We separate the motors from the microprocessor to prevent voltage and current fluctuations caused by the motors from affecting or damaging the microcontroller and other low-voltage electronic components. In addition this power supply will provide 12V power to the pickup circuit. We will use a standard 40W, +/-12V power supply supplied by the ECE shop.

2.2.7 Guitar String and Mechanical Systems

The guitar string will be fixed at either end of a 0.6 meter track. A metal bar will slide along this track and presses down upon the string to change the frequency at which the string vibrates. This bar will be fixed to the pitch motor system with a belt drive. On one end, the string will be attached by a tuning key to allow tuning adjustments. The string will be fixed on the other end, and the magnetic pickup and picking motor system will be attached near this end.

2.2.8 Magnetic Pickup

The pickup converts the mechanical vibration of the guitar string to an electrical signal and amplifies it to a level that is detectable to the analogue input of the audio processing unit. It consists of a magnet and a coil with 8000–10000 turns. It will require 12V power, which will be taken from the motor power supply. We will use a humbucking pickup in order to minimize background noise from the environment and motors. This style of pickup uses two coils such that background noise in each coil is 180° out of phase from the other and therefore cancels out, while the signal from the string is in phase and adds together.

2.2.9 DAC and Line Out

The DAC module will use a Texas Instruments PCM5100a to convert I^2S data from the microcontroller to an analog audio signal. It will contain a standard audio jack for connecting the instrument to speakers or other audio hardware. It will take 5V power from the microcontroller and step it down to 3.3V to power the PCM5100a.

2.3 Circuit Diagrams

The MIDI circuit uses the standard MIDI input circuit [1]. The DAC circuit is a slight modification of the circuit given in the "Typical Application Circuits" section of the PCM5100a datasheet [2] and the application circuit from the LD1117V33 datasheet [3].

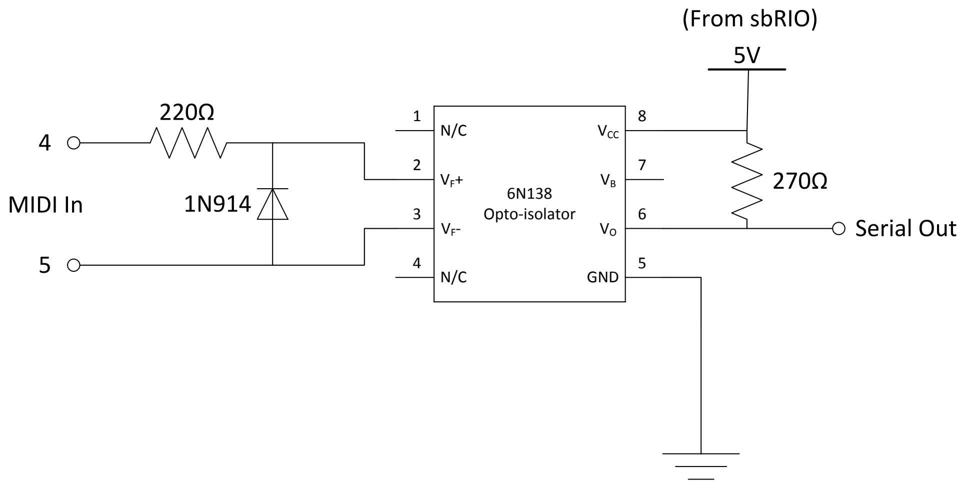

Figure 3. MIDI input circuit

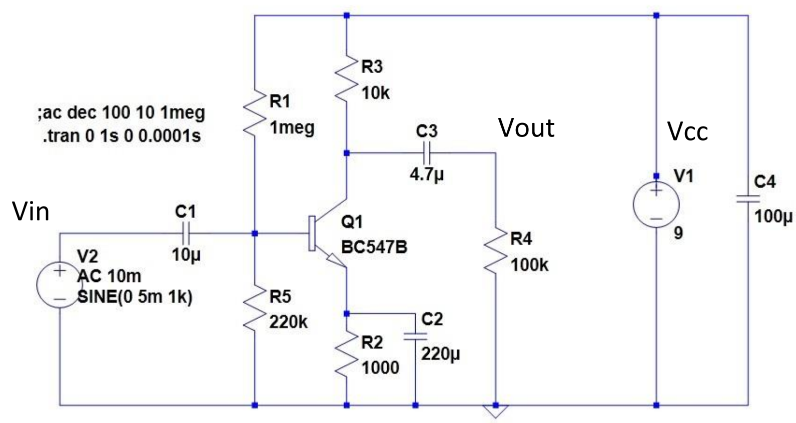

Figure 4. Pickup amplifier circuit

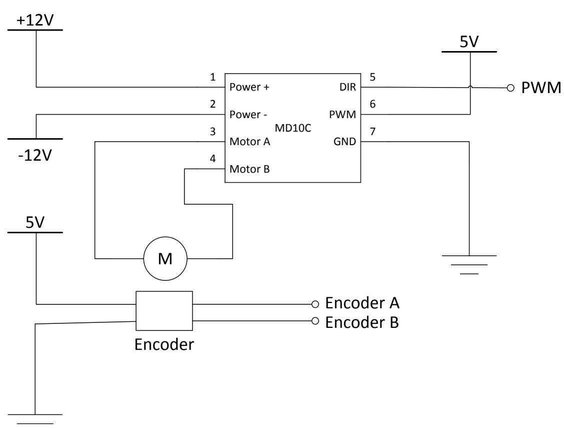

Figure 5. Motor controller and encoder

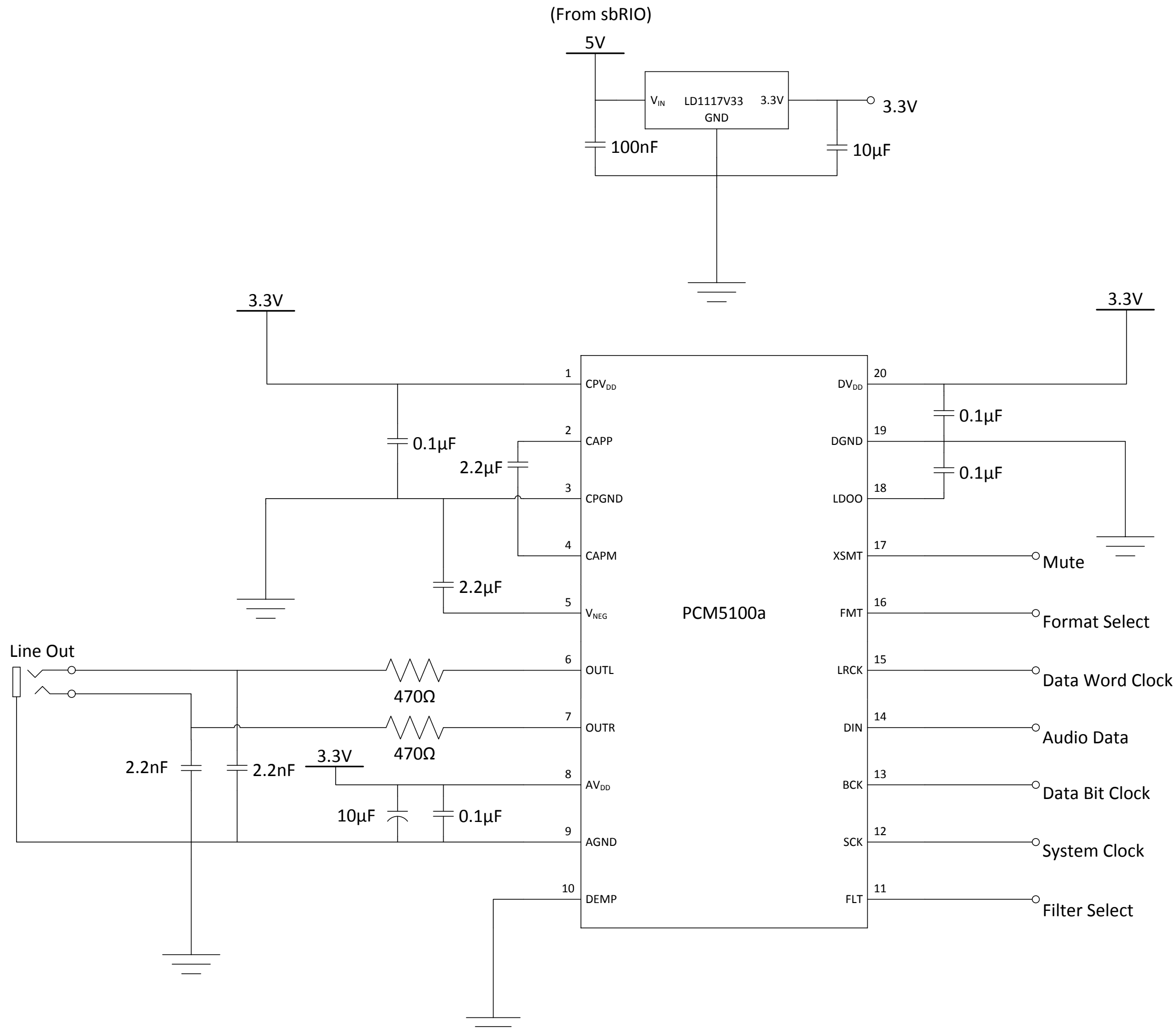

Figure 6. DAC circuit

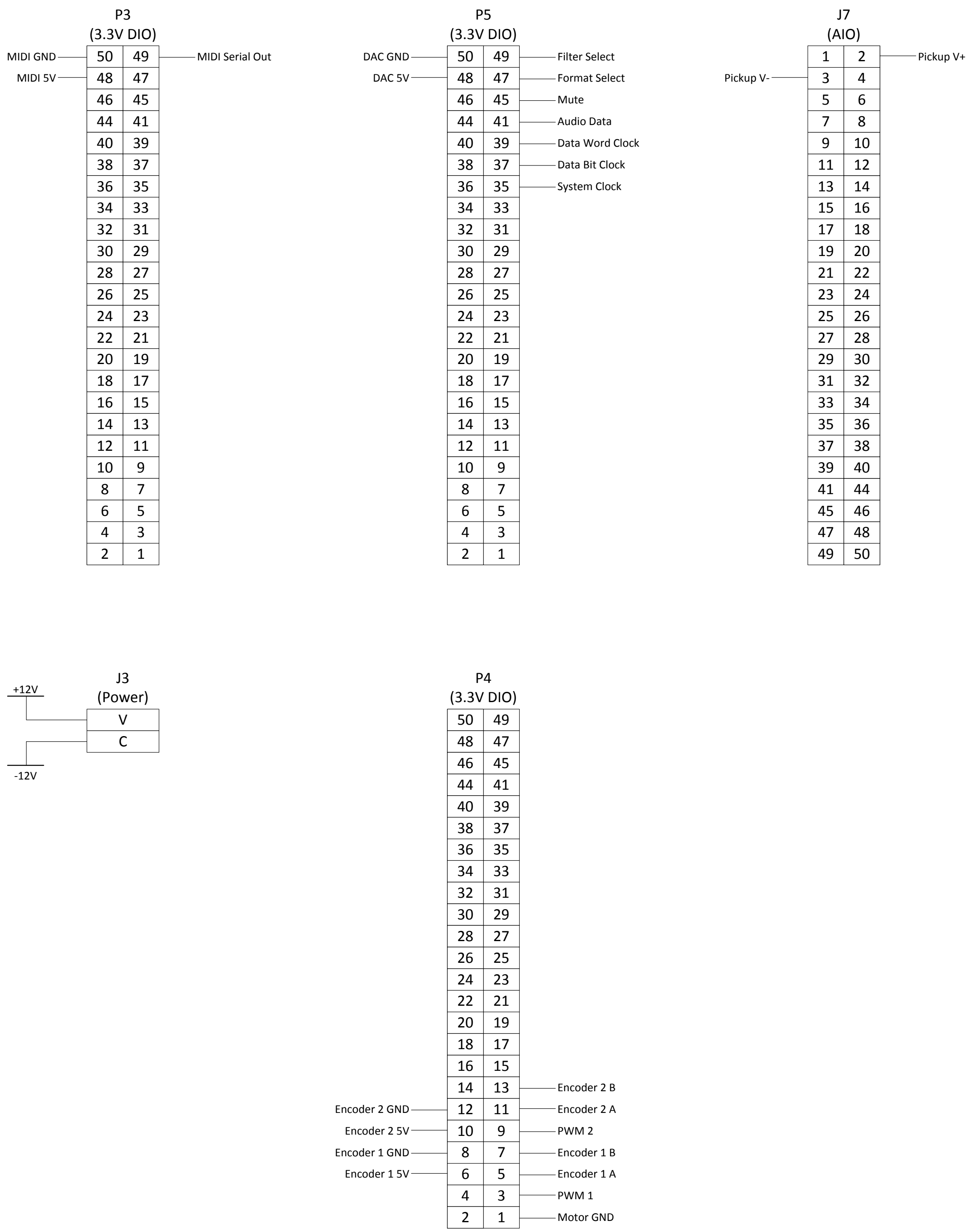

Figure 7. sbRIO microcontroller

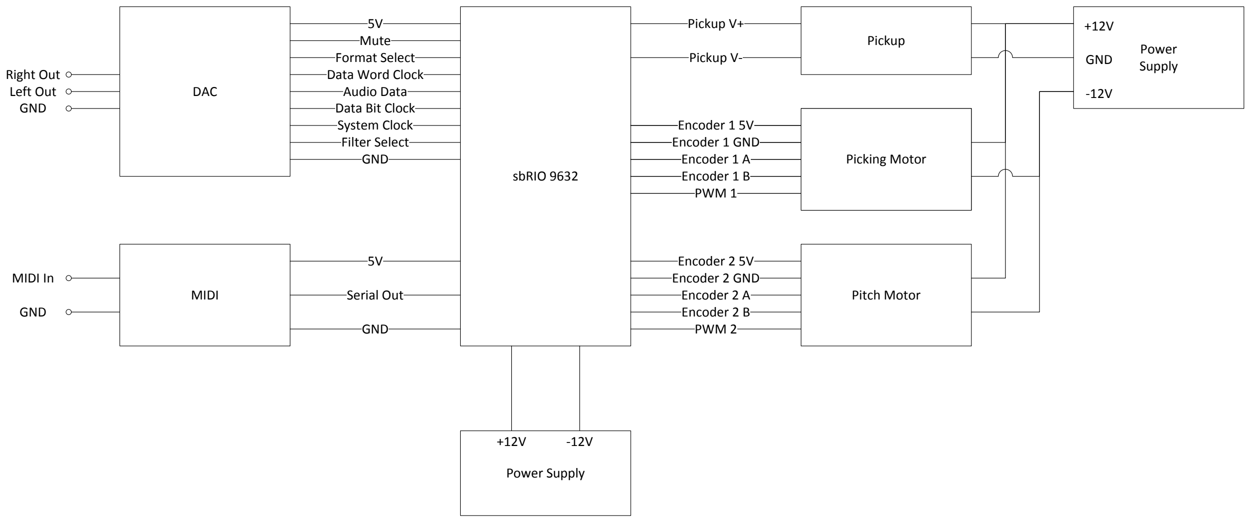

Figure 8. Component interconnections

2.4 Software Diagram

Figure 9 describes the software we will use to control the motors and process audio. Wide arrows correspond to physical inputs and outputs through the microcontroller's digital and analog IO.

The MIDI Note and CC Reader module takes the MIDI input stream and detects note events and MIDI continuous controller values. These values are then fed as parameters to the Audio Processor—which applies audio effects to the audio signal—and Note Event and Pitch Generator—which determines timing for the picking motor and the desired position of the pitch motor. The Motor Controller module is responsible for moving the motors as dictated by the Note Event and Pitch Generator.

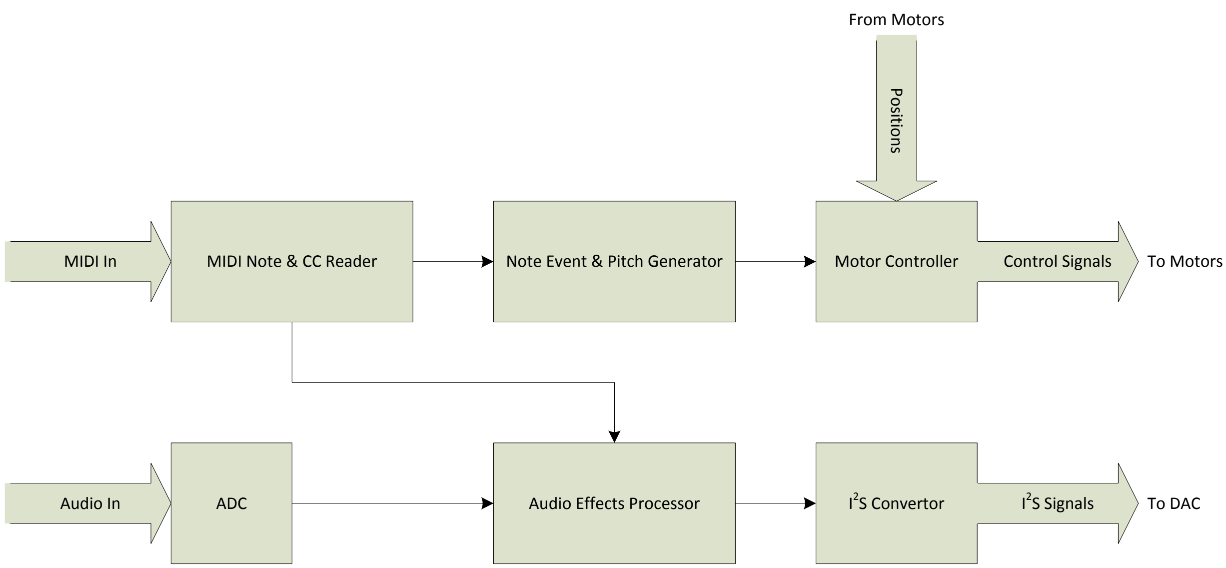

Figure 9. Software flowchart

2.5 Simulations and Calculations

2.5.1 Digital Logic Circuit Power

The MIDI input circuit contains only one current-drawing element—the 6N138 opto-isolator. It will draw a maximum of 100mW [4]. The Texas Instruments PCM5100a used in the DAC circuit, when powered at 3.3V, will draw a maximum of 148.5mW [2]. The various capacitors in the DAC circuit are used to stabilize the power source, and will draw a negligible amount of power. The power dissipated by the LD1117V33 voltage regulator can be approximated as $(V_{in} - V_{out}) \cdot I_{in}$ where $V_{in} = 5V$, and $V_{out} = 3.3V$. Assuming I_{in} is the maximum current drawn by the PCM5100a of $\frac{148.5mW}{3.3V} = 45mA$, the power dissipated by the regulator is $(5V - 3V) * 45mA = 76.5mW$. The digital logic circuits should thus draw a maximum of 325mW.

The National Instruments sbRIO-9632 can supply 2A at 5V, for 10W [5]. The power requirements for our digital logic circuits are well within the capabilities of the sbRIO.

2.5.2 Pickup Circuit

Since the voltage output from the pickup coil is very small, before we feed it into the differential input of the microcontroller, we need a pre-amplifier to boost the signal to a detectable range (0.1–2V). Some guitarists dislike the sound of amplifiers using op-amps since they introduce distortion [6], so we will use a single transistor amplifier using a BJT [7]. We have changed the value of C2 in the amplifier circuit (Figure 4) to modify the low cutoff frequency of the amplifier according to the formula $f_L = \frac{1}{2\pi RC}$.

For simulations, we model the output load as a 10kΩ resistor. As shown in Figure 10, the pre-amplifier has a high, nearly constant gain for frequencies above 100Hz. Since the range of human hearing is about 20Hz–20kHz and our guitar string is not long enough to produce very low frequencies, this is suitable for our application. Figure 11, shows the input voltage (Vn004) and output voltage (Vn003) for a 20kHz sine wave. The input voltage is 10mV peak-to-peak and the output is approximately 1.8V peak-to-peak, giving us a gain of $\frac{1.8V}{10mV} = 180\frac{V}{V}$.

2.6 Contingency Plan

Since motors powerful enough to move the pitch bar at speeds sufficient for our requirements are very expensive, if we are unable to procure a sufficiently powerful motor, we will control pitch digitally instead of physically by applying a pitch-shifting algorithm to the digital audio signal before we apply any other audio effects. In this case, the software flowchart will be slightly modified. Pitch data will be fed into the Audio Effects Processor instead of the Motor Controller, and the Audio Effects Processor will handle pitch shifting.

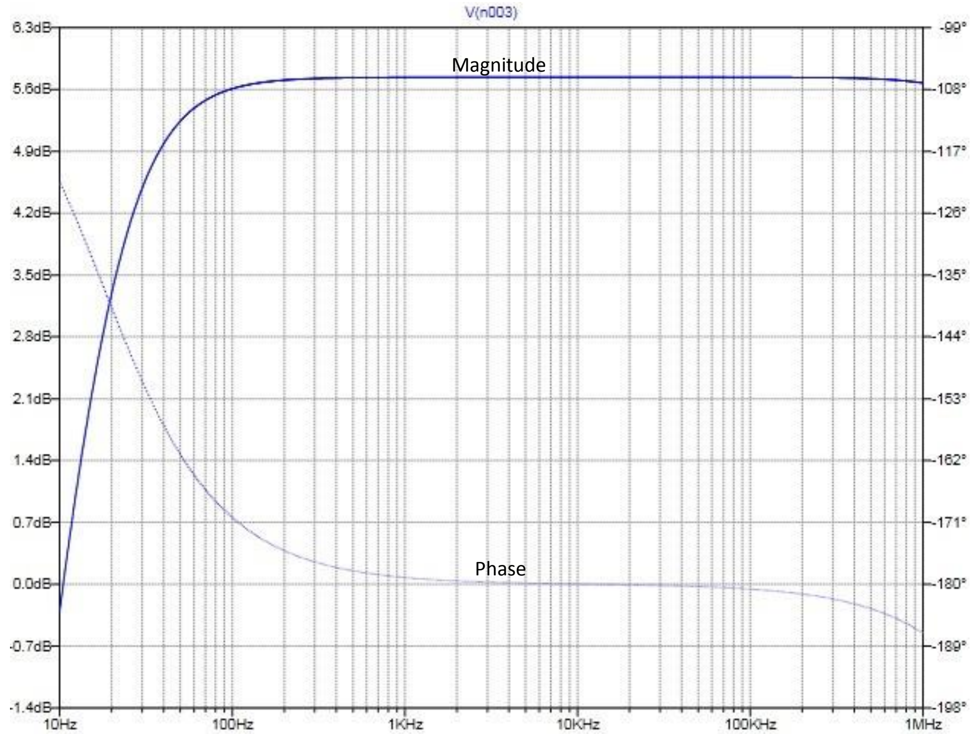

Figure 10. Amplifier frequency response

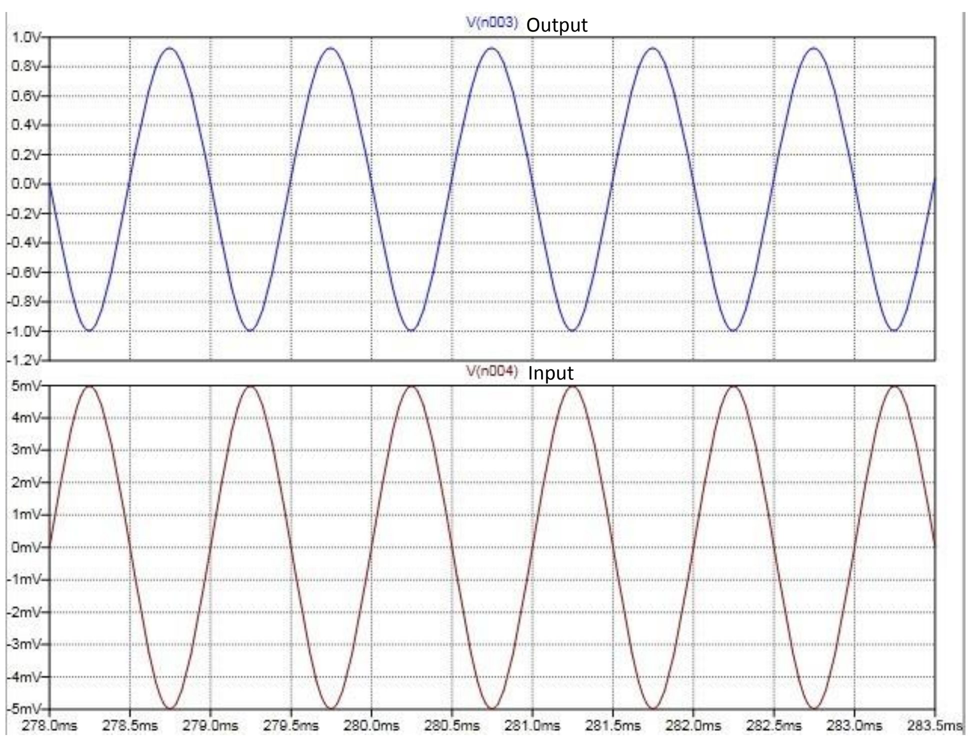

Figure 11. Amplifier voltage gain (20kHz sine wave input)

3.0 Requirements and Verification

Requirement	Verification
1) Microcontroller initializes properly	
a) Power supply should be between 17 and 30 volts	a) Verify power supply voltage with a voltmeter
b) The board should power on	b) The power (top-right) LED is lit
c) The board supplies 5 volts ±10% to digital logic circuits	c) Verify 5V pin voltages with a voltmeter
d) The board is properly configured	d) The status (bottom-right) LED is not flashing. If it flashes, check page 22 of the user guide to determine the specific error and troubleshooting steps.
e) Connection between a computer and the microcontroller can be made. LabVIEW is able to communicate with the microcontroller.	e) Connect the microcontroller to a computer with an Ethernet cable. Check that LabVIEW is able to establish a connection.
2) MIDI module and software function properly	
a) Microcontroller must initialize properly	a) See requirement 1.
b) Power terminal of MIDI input module must be at 5 volts ±10%.	b) Verify voltage with a voltmeter
c) Circuit transfers MIDI input to 5V serial output. Logic high is > 4V. Logic low is < 0.8V.	c) Connect a MIDI instrument and send arbitrary MIDI data while measuring the serial output with an oscilloscope. Freeze the oscilloscope while transmitting data. Verify voltages.
d) MIDI data received by the microcontroller is identical to that which was sent. Bit error rate is 1 in 10,000 or less.	d) Connect a computer as a MIDI source and use MIDI OX to send a stream of 10,000 bits. Record the received data and compare. Repeat to improve confidence.
e) MIDI data is read properly i) Pitches are correct ii) Velocities are correct iii) Continuous controller (CC) values are correct iv) Pitch bends are correct	e) Connect a MIDI instrument i) Play notes at different pitches and verify that received pitches match. ii) Play notes at different velocities and verify that received velocities match. iii) Send CC values of various values for various controllers. Verify that controller numbers and values match iv) Change the pitch bend and verify that received values match.
3) Motors function properly	
a) Motor controllers are supplied 24 volts ±5%.	a) Verify power supply voltage with a voltmeter
b) Motors work	b) Connect motors to a 24V power supply with variable current capability. Increase current from 0 and verify that motors rotate. Do not exceed 1A. Reverse polarity and repeat.

c) Motor controllers work	c) Connect controller outputs to a multimeter. Press the test buttons on the controller. Verify that test LEDs light and the two buttons output positive and negative currents respectively.
d) Motor controllers drive motors	d) Connect motors to controllers and press the test buttons on the motor controller. Verify that test LEDs light and motors turn in both directions.
e) Motor speeds can be controlled with PWM signal	e) Use a function generator to send a 3.3V, 10kHz square wave with 50% duty cycle to the motor controller PWM input. Verify that the motor does not rotate. Slowly sweep the duty cycle from 50% to 100% and check that the motor speed increases. Reset the duty cycle to 50%, sweep it to 0%, and verify that the motor rotates in the opposite direction and speed increases.
f) Microcontroller sends PWM signal correctly	f) Repeat verification 3.e using the microcontroller. Use an oscilloscope to verify that the high voltage is > 3V.
g) Motors are supplied enough power under load.	g) Connect the motors to their loads. Run a test program on the microcontroller that runs the motors in both directions to simulate normal operation.
i) Supply voltage does not drop	i) Verify that the input voltages to the controllers do not drop more than 5%.
ii) Supply currents do not exceed rated values for motors	ii) Verify that the current into each motor does not exceed 1A
h) Supply currents do not exceed rated values for motors when motors are stalled	h) Prevent the motors from turning. Drive the motors and verify that the current into each motor does not exceed 1A. Start with low speeds and test increasing speeds to reduce risk of damaging the motors with large currents.
4) Motor control software functions properly	
a) Microcontroller must initialize properly	a) See requirement 1.
b) Motors must function properly	b) See requirement 3.
c) Power to encoders must be 5 volts ±5%	c) Verify voltages with a voltmeter
d) Microcontroller should accurately read encoder position values	d) Disconnect power from the motors. Manually turn the motors arbitrary angles and verify that the encoder positions change properly.

Requirement	Verification
e) Control software is able to control picking motor properly	e) Instruct the control software to pick once. Verify that the picking motor rotates enough to pick the string once and stops. Repeat and send commands in different rhythms. Verify that the string is hit with the proper rhythm.
f) Control software is able to control pitch motor properly	f) Instruct the control software to move the pitch bar to a specific position. Verify that the bar reaches that position. Repeat for various positions.
g) Software properly converts pitches to pitch bar positions	g) Instruct the control software to move to a specific pitch. Wait for the pitch bar to stop, then strike the string. Use a tuner to verify the pitch.
h) Software coordinates picking and pitch motors	h) Play a sequence of notes into the control software. Verify that the correct pitches and rhythms are reproduced.
5) Pickup functions properly	
a) Pickup is supplied 12 volts ±5%	a) Verify supply voltage with a voltmeter
b) Pickup converts string vibration to electric signal. Amplitude is >1V.	b) Manually pluck the string and measure pickup output with an oscilloscope. Verify that a signal is received. Verify its amplitude.
c) Signal is can be read by microcontroller analog input	c) Connect pickup to microcontroller manually pluck the string. Verify that microcontroller receives values.
d) Signal does not saturate analog input	d) Graph analog input values while plucking the string. Verify that waveforms do not reach the maximum numerical value.
e) Resolution of analog input is high enough to capture small vibrations	e) Verify that microcontroller captures a signal when manually tapping on the string.
6) Audio software functions properly	
a) Microcontroller must initialize properly	a) See requirement 1.
b) I^2S generator converts samples to I^2S	b) Feed constant value audio samples into the generator. Use an oscilloscope on the microcontroller outputs to verify that the I^2S data matches specifications. Trigger on the LRCK channel to view a single audio frame at a time. Repeat for at least 20 different sample values.
c) Software reads string signal and outputs audio	c) Check requirements 5 and 7.b. Disable all audio effects. Connect speakers to the line out and pluck the string. Verify that it plays a sound.
d) Software applies audio effects to signal	d) Enable audio effects individually and pluck the string. Verify that each effect changes the tone of the sound properly.

e) Audio effects can respond to MIDI values	e) Check requirement 2. Configure the distortion level to respond to the modulation wheel CC. Vary the mod wheel value while playing and verify that the amount of distortion changes.
7) DAC circuit functions properly	
a) Microcontroller must initialize properly	a) See requirement 1.
b) Power terminal of DAC module must be at least 4.4 volts	b) Verify voltage with a voltmeter
a) Power after voltage regulator must be 3.0–3.6 volts	c) Verify voltage with a voltmeter
b) DAC reproduces audio from I^2S data	d) Check requirement 6.b. Generate I^2S signals from a sine wave. Sweep the waveform frequency from 100 Hz to 20 kHz while measuring the line out with an oscilloscope with FFT capability. Verify that the output frequency always matches the test signal frequency.

3.3 Tolerance Analysis

Motor Controller

The pitch control is the most critical part of this project. The motor should be able to move the pitch bar over a distance equivalent to two whole steps (from the center of the string as the distance corresponding to one whole step varies over the length of the string) within 100ms. It should reach the desired position ± 2 mm.

Since the power of our motor is limited by our budget, if the motor we are using is not capable of reacting this quickly we will use the backup plan described in section 2.6 Contingency Plan.

Test Plan

We will test this by feeding the motor controller a sequence of pitches and using the encoder position data recorded by the microprocessor to determine how quickly and accurately the motor reached each pitch. We will also test by sending a fixed pitch, waiting for the motor to reach the desired position, and physically measuring the error to make sure our encoder positions are correct.

4.0 Ethical Issues

The statements of the IEEE Code of Ethics that pertain to our project are as follows [8]:

1. “to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;”

The development of our project poses no public safety risks as we are not using any hazardous materials, but our final product will contain moving parts and dangerous electrical currents. We will shield these components to minimize risk to the end user. In addition, we will provide appropriate warning labels to inform users of the dangerous components.

3. “to be honest and realistic in stating claims or estimates based on available data;”

We will record any experimental data accurately to the best of our ability for future reference. We will not modify or falsify data for any reason.

6. “to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;”

When we encounter areas that are outside of our expertise—especially when dealing with the potentially dangerous components of our project—we will consult with qualified engineers to ensure that our design is correct and that we create a safe product.

7. “to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;”

We will check each other’s work and provide feedback to correct errors or improve the project. We will credit any work that we use—either directly or after modification—in the design and construction of our project.

5.0 Safety Issues

5.1 Electrical Safety

As our project uses a microcontroller and large motors, each of which will operate at high voltage and current, we will need to take appropriate safety precautions when working on our project. Some of our guidelines overlap with safety rules for the ECE 445 lab, but as they should be kept while working outside of the lab as well, we feel it is appropriate to list them here as well.

- While working on the project inside or outside of the lab, before powering up the microcontroller and/or motors, there must be at least two people present.
- No food, drink, or any other liquids will be allowed near the project.
- Before modifying any circuit or connecting instruments to the high voltage/current components of the project, all components must be powered down.
- Use the "one hand rule" when working around the project while it is powered on. That is, keep one hand away from electrical components—preferably by keeping it in a pocket—to reduce the risk of a shock across the heart.
- Wires should be kept organized. To reduce the risk of incorrectly connecting components, wires should be properly color coded (power = red, ground = black, data lines have other colors) and, interconnecting wires which may be frequently connected and disconnected should be labeled.

In addition to safety concerns while working on our project, we must design with the safety of the end user in mind. To that end, in our final product, all high voltage/current components must be shielded to prevent someone from coming into contact with them.

5.2 Mechanical Safety

Since our project contains parts moving at high velocities, further precautions must be taken. In addition to the electrical guidelines given above, the following precautions should be taken with regard to mechanical components.

- Keep hands away from the motors, belt drive and picking wheel during operation.
- After long periods of operation the motor will tend to heat up. Avoid touching the motors not only during operation but also until the motors have had sufficient time to cool off.
- An emergency stop switch should be added to the project to immediately cut power to the motors, preferably by physically disconnecting them. This will precaution will help to avoid physical injury or damage to components in case the motor control behaves unexpectedly. This switch should be located in an easily accessible spot, far from the motors.

Keeping the end user in mind, the final product should have its moving components reasonably shielded so that it is difficult to come into contact with moving parts or touch the hot motors.

6.0 Cost and Schedule

6.1 Cost Analysis

6.1.1 Labor

Name	Hourly Rate	Total Hours Invested	Total $(\boldsymbol{Hourly\ Rate} \times \mathbf{2.5} \times \boldsymbol{Total\ Hours\ Invested})$
Angad Bector	35.00	168	14700
Joel Spadin	35.00	168	14700
Ruichen Zhao	35.00	168	14700
Total			44100

6.1.2 Parts

Item	Quantity	Cost (USD)
SDS-50J DIN connector	1	3.75
6N138 opto-isolator	1	1.00
LD1117V33 3.3V regulator	1	1.95
PCM5100a stereo audio DAC	1	1.78
TSSOP-20 to DIP-20 SMT adapter	1	4.50
9632 NI Single-Board RIO	1	40.00
Minertia B5M-GS12 DC Motor with encoder	2	300.00
MD10C motor controller	2	29.00
Kyosan KVL40-02 40W power supply	2	50.00
PCBs	3	90.00
BC547 BJT	1	0.50
Alnico 2 polepiece magnet	2	4.96
Guitar string	1	6.00
Total		533.44

6.2 Schedule

Week		Tasks	Member assigned
1	1/14	Work on RFA	Joel & Ruichen
2	1/21	Research components	Joel & Ruichen
3	1/28	Work on Design proposal	Joel
4	2/4	Finish Design proposal	Joel
		Contact motor suppliers, place order for Motors	Joel
		Place order for DAC	Joel
		Place order for guitar pickup, guitar string	Ruichen
		Place order for NI SBRIO	Angad
5	2/11	Mock Design Review	Joel
		Prototype MIDI circuit	Joel
		Prototype DAC circuit	Ruichen
		Prototype Motor control circuits & software(part1)	Angad
		Prototype Pickup circuit(part1)	Ruichen
		Mechanical system design & submit to the machine shop	Angad

6	2/18	Design MIDI circuit PCB & software	Joel
		Design DAC circuit PCB & I2S interface	Joel
		Prototype Motor control circuits & software(part2)	Angad
		Prototype Pickup circuit(part2)	Ruichen
		Finalize Mechanical system design(if necessary)	Angad
7	2/25	Design Review	Joel
		MIDI circuit & software testing	Joel
		DAC circuit & I2S interface testing	Joel
		Motor control tuning	Angad
		Design Pickup circuit PCB	Ruichen
8	3/4	Motor control tuning	Angad
		Pickup circuit tuning	Ruichen
		Audio effect software implementation	Joel
9	3/11	Individual Progress Report Due	Joel & Angad & Ruichen
		Motor control system testing	Angad
		Pickup circuit testing	Ruichen
		Audio effect software tuning	Joel
10	3/18	Spring Break	
11	3/25	Mock-up Demo	Joel
		Verification of specifications	Ruichen
		Tolerance Analysis	Angad
12	4/1	Mock-up Presentation	Joe
		Fixing problems	Ruichen
13	4/8	Debugging MIDI circuit & audio effect software(part1)	Joel
		Debugging Pickup circuit(part1)	Ruichen
		Debugging Motor control system(part1)	Angad
14	4/15	Debugging MIDI circuit & audio effect software(part2)	Joel
		Debugging Pickup circuit(part2)	Ruichen
		Debugging Motor control system(part2)	Angad
15	4/22	Project Demo	Joel
		Prepare for presentation	Angad
		Writing Final Report	Ruichen
16	4/29	Project Presentation	Joel
		Final Report due	Ruichen
		Lab notebook due	Joel & Ruichen & Angad
17	5/9	Check in supplies	Ruichen

7.0 References

[1] T. Igoe, "MIDI Communication," [Online]. Available: http://www.tigoe.com/pcomp/code/communication/midi/. [Accessed 24 February 2013].

[2] Texas Instruments, "2VRMS DirectPath™, 112/106/100dB Audio Stereo DAC with 32-bit, 384kHz PCM Interface," September 2012. [Online]. Available: http://www.ti.com/lit/ds/slas859a/slas859a.pdf. [Accessed 24 February 2013].

[3] STMicroelectronics, "LD1117 Series Low Drop Fixed and Adjustable Positive Voltage Regulators," December 2005. [Online]. Available: http://www.sparkfun.com/datasheets/Components/LD1117V33.pdf. [Accessed 24 February 2013].

[4] Fairchild Semiconductor, "Single-Channel: 6N138, 6N139 Low Input Current High Gain Split Darlington Optocouplers," August 2008. [Online]. Available: http://www.fairchildsemi.com/ds/6N/6N138.pdf. [Accessed 24 February 2013].

[5] National Instruments, "NI sbRIO-961x/963x/964x User Guide," [Online]. Available: http://www.ni.com/pdf/manuals/375052c.pdf. [Accessed 24 February 2013].

[6] "A Discrete FET Guitar Preamp," Electronic Music Article, 6 May 2001. [Online]. Available: http://www.till.com/articles/GuitarPreamp/.

[7] Toni, "bc549 transistor pre amplifier," Xtronic, 23 Aug 2011 . [Online]. Available: http://xtronic.org/circuit/amplifier/circuit-audio-preamplifier-bc549-transistor-pcbs-volume-control/.

[8] IEEE, "IEEE Code of Ethics," [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed 25 February 2013].

[9] Cytron Technologies, "MD10C," February 2013. [Online]. Available: http://www.cytron.com.my/usr_attachment/MD10C_Rev2.0_User's_Manual.pdf. [Accessed 24 February 2013].