MIDI Controlled Slide Guitar

Project Proposal

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

February 6, 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.

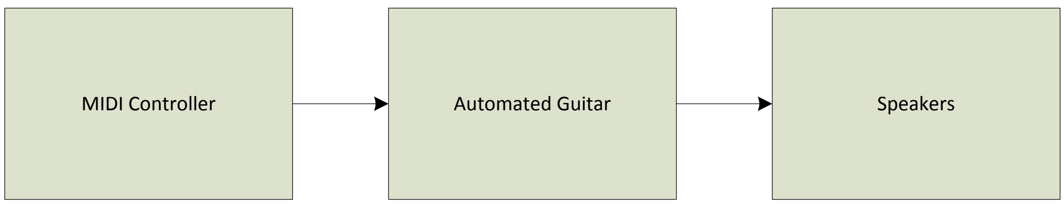

Figure 1. Performance setup

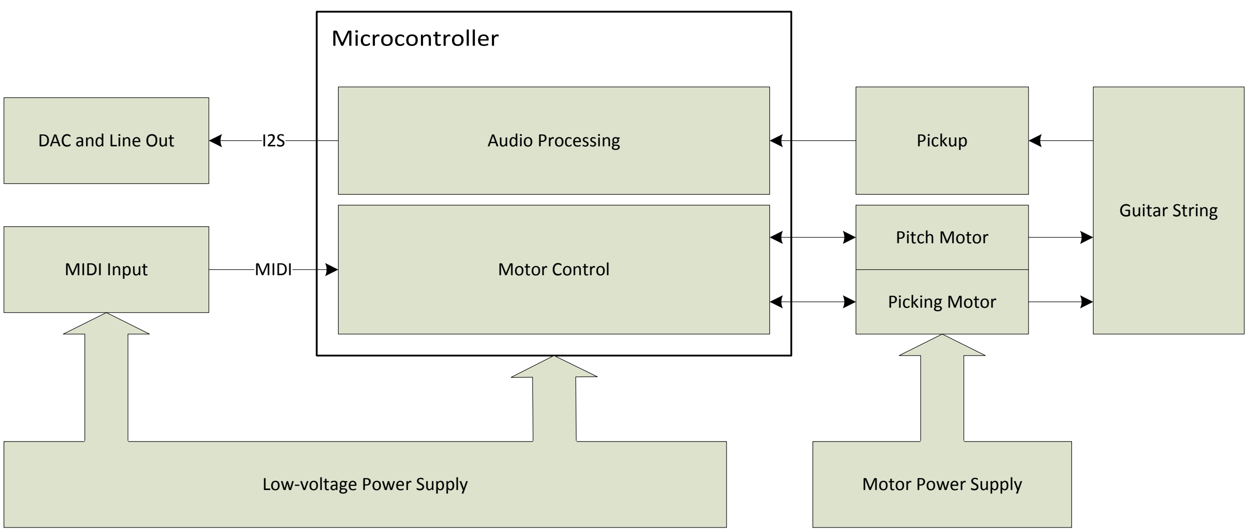

Figure 2. Top level system layout

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.

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 Low-voltage Power Supply

This power supply will provide power from standard AC power (120V AC, 60Hz) to the microcontroller, MIDI input circuit, and DAC circuit. The DAC requires 3.3V and the MIDI circuit requires 5V, so we will use a standard 5V AC adaptor and a 3.3V voltage regulator to provide both 5V and 3.3V power.

2.2.4 Pitch Motor

The pitch motor module will consist of a motor, amplifier, and position sensor. It will move the pitch bar along the guitar string. We would like to use a linear motor, as that would provide the best response times and accuracy, but if we cannot acquire one we will use a powerful servo motor and a belt system to move the pitch bar. The type of input signal will depend on the particular motor and amplifier we use, but it will likely be a PWM signal. The position sensor on the motor (a rotary encoder in the case of a rotational motor, or an equivalent for linear motors) will send data back to the microprocessor to allow it to perform closed-loop control.

2.2.5 Picking Motor

The picking motor will consist of a servo motor, amplifier, and position sensor. It will control a pick that strikes the string. Again, the input signal will depend on the motor and amplifier, but will likely be a PWM signal. A rotary encoder on the motor will position data back to the microprocessor.

2.2.6 Motor Power Supply

This power supply will provide power from standard AC power (120V AC, 60Hz) to the motor amplifiers. We separate the motors from the microprocessor power supply because the motors have much higher voltage and current requirements. In addition, this will prevent voltage and current fluctuations caused by the motors from affecting or damaging the microcontroller and other low-voltage electronic components.

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 slides along this track and presses down upon the string to change the frequency at which the string vibrates. This bar is fixed to the pitch motor system. 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.

2.2.9 DAC and Line Out

The DAC module will use a Texas Instruments PCM5100 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.

3.0 Requirements and Verification

3.1 Requirements

1. Microprocessor should read MIDI input and determine desired pitch and picking times/velocities.
2. Microprocessor should read the analog audio input and generate I^2S output.
3. Pitch motor should be able to make changes of two whole-steps within 100ms (from the center of the string).
4. Picking motor can strike the string from stationary position within 50ms and strike again within 150ms.
5. Low voltage power supply should supply 3.3V and 5V power with less than 1% voltage ripple.
6. Motor power supply should be able to drive both motors at the same time with less than 5% voltage ripple.
7. Magnetic pickup should convert string's vibration into an electric signal large enough to be detected by the NI board's ADC.

3.2 Verification

3.2.1 Microcontroller

1. Test the low-voltage power supply (as explained below).
2. Connect the microcontroller to a computer and check that it establishes a connection.

3.2.2 MIDI Input

1. The MIDI source should be tested by connecting it to a computer or synthesizer and verifying that it properly generates MIDI notes.
2. Verify that the MIDI input module generates a 5V serial data signal by connecting it to an oscilloscope and taking snapshots of the output while the MIDI source is sending data.
3. Connect the MIDI input circuit to the microcontroller and send MIDI data in. Check that the microcontroller receives the same notes and continuous controller data that is sent by the MIDI source.

3.2.3 Motors

1. Test the motor power supply (as explained below).
2. Disconnect the motors from the power source. Set the microcontroller to measure the encoder positions. Manually move the motors and verify that the encoder positions change appropriately.
3. Reconnect the motors. Run a test program on the microcontroller that provides a series of notes and check that the motors react appropriately. The picking motor should rotate 90 degrees per note and the pitch motor should move the pitch bar to the appropriate position to set the pitch of each note.

3.2.4 Power Supplies

1. Test the low-voltage power supply by connecting the microcontroller, MIDI input, and DAC and measuring the 3.3V and 5V outputs. Check that the voltages are at the proper levels and that the ripples are within the required tolerances.
2. Test the motor power supply by connecting both motors and running a test program that continually moves the motors to simulate normal operation. Measure the output voltage and check that it is at the proper level and that ripple is within the required tolerance.

3.2.5 Magnetic Pickup

Pluck the string and measure the output voltage. Check that its amplitude is large enough to be easily detected by the NI board's ADC.

3.2.6 DAC and Line Out

1. Use a function generator or the microprocessor to generate a test I^2S signal such as a 440Hz sine wave. Verify the signal using an oscilloscope.
2. Feed the signal into the DAC. Measure the analog audio signal using an oscilloscope with FFT capability. Verify that the spectrum of the output signal matches that of the test input.
3. Connect the DAC to standard speaker hardware. Listen and verify that the generated audio matches the test input.

3.3 Tolerance Analysis

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. 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 Cost and Schedule

4.1 Cost Analysis

4.1.1 Labor

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

4.1.2 Parts

Item	Quantity	Cost
DIN connector	1	3.75
6N138 opto-isolator	1	1.00
AS5145 rotary position sensor	2	12.70
PCM5100 stereo audio DAC	1	1.78
NI Single-Board RIO	1	40.00
Servo Motor (pitch motor)	1	300.00–1000.00
Servo Motor (picking motor)	1	100.00
Motor amplifiers	2	800.00
Power supply	2	30.00
PCBs	2	60.00
Pickup	1	10.00
Guitar string	1	5.00
Total		1364.23–2064.23

4.2 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	Angad
		Prototype Pickup circuit	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	Angad

		Prototype Pickup circuit	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	Ruichenl
13	4/8	Fixing problems	Joel
14	4/15	Fixing problems	Joel
		Prepare for demo	Ruichen
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