Self-Playing Clarinet

ECE 395/397 | SP-FA 2016

UNIVERSITY OF ILLINOIS

Joan Brown, Adam Long, Jordi Pakey-Rodriguez

jbrown76, arlong2, pakeyro2

ABSTRACT

With common ground and passion for music, the group's goal is to create an electronically controlled 'self-playing' clarinet. Parts and inspiration can be observed from previous projects in ECE 395, such as a self-playing guitar and slide whistle. While not intended to replace musicians, the project itself is a display of the complexity of producing sound, controlling pitches, and playing a composition of music in a reed instrument.

INTRODUCTION

The production of particular pitches of sound through a reed instrument begins with precise air speed and pressure and ends with specific fingerings of the instrument. Along the way, there are various factors that can affect the volume, tone, frequency, and overall quality of the sound produced. These factors, discussed in their corresponding sections of the Description, pose as the most difficult obstacles to overcome in creating the self-playing clarinet. Push-pull solenoids are used in place of human fingers to push keys down and release them. Solenoids are preferred to servos for their lessened time delays, resulting in superior ability to push keys quickly.

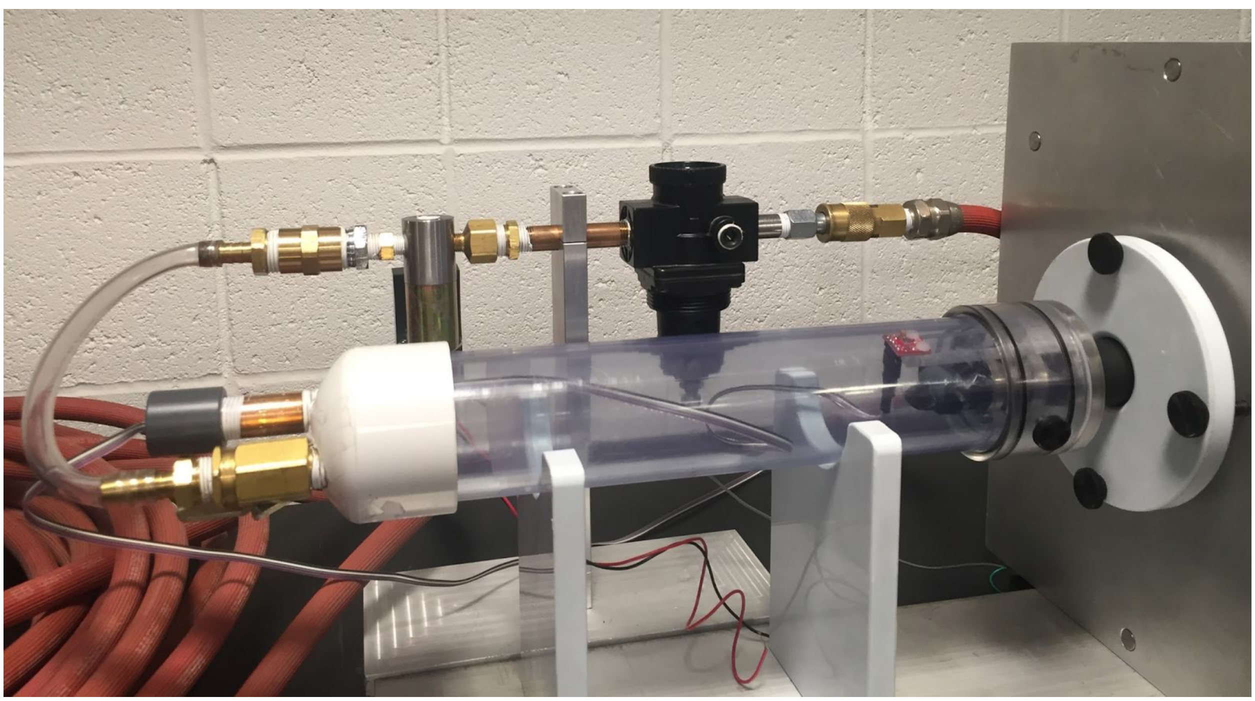
At the end of the spring semester, the project was a lip-syncing clarinet as opposed to a self-playing clarinet due to time limitations. Precise control over the air flow into the chamber and air pressure within the chamber are necessary in order for the clarinet to produce sound. While the equipment utilized provides sufficient control to play a small range of notes, it does not provide fine enough control in order to ensure the full playable range of the clarinet be displayed. By the time the course reached an end, there was no time to perfect the acoustic aspects of the project. Continued work on the project in the summer and fall nearly finalized the project as a self-playing clarinet. See Continuation for future recommendations and work completed the project post- spring semester.

DESCRIPTION

Air Chamber

The air chamber is a clear one-inch piece of PVC with a cap on each end. The first cap is permanently attached to the PVC and has two threaded mounts for the air control. The other cap provides the interface between the chamber and the clarinet. The end of this cap sits between the mouthpiece and barrel of the clarinet creating a pressure seal for the end of the chamber. The second cap is attached with two screws and o-rings so that it can be removed.

The air control system is responsible for setting and maintaining the pressure inside the chamber. The system takes as an input the pressurized air from the taps in the lab. This air then passes through a regulator and flows into the the chamber via the first threaded mount. A proportional valve is attached to the second threaded mount of the chamber to leak excess air. The system works by adjusting the regulator and the leak valve such that the chamber is at the desired pressure for the given note. Unfortunately, the regulator used in the first iteration of this project was not fine enough and the chamber would become too pressurized for the leak to handle. This prevented the clarinet from playing its full range.



Air Pressure Control System

Mouthpiece and Reed

A strength three synthetic reed was attached to the mouthpiece with a standard clarinet ligature in the spring. The reed was then pushed down with an elastic band and a piece of foam sat between the band and the reed. This was done to simulate the human embouchure pushing down on the reed. This system then sat inside of the pressurized chamber and the air was forced through the mouthpiece, vibrating the reed and playing the clarinet.

The finalized design used a strength two reed so that less air pressure was needed. During testing, we found that the sound quality improved as well in terms of a softer volume. The design used for liip force involved a black elastic wrapped around the reed and mouthpiece, hot glued on the curved side of the mouthpiece and secured by a nail. The end of the nail was shaved off in order to fit in the air chamber. By using a wound elastic, more surface area of the reed was covered than in the previous design. This improved the sound quality produced as well.

Mount

The project mount consists of a long metal base with four upright plates. Two of these plates are plastic, and serve as holders for the chamber. The remaining two plates are metal, and hold the clarinet in place. Each metal plate has a hole in its center; one hole encompasses the barrel of the clarinet, and the other securely holds the end of the bell. The barrel is securely held by a plastic ring beyond the metal plate. All holes and holders are horizontally aligned so as to not put unwanted pressure on the clarinet or cause a joint to leak air.

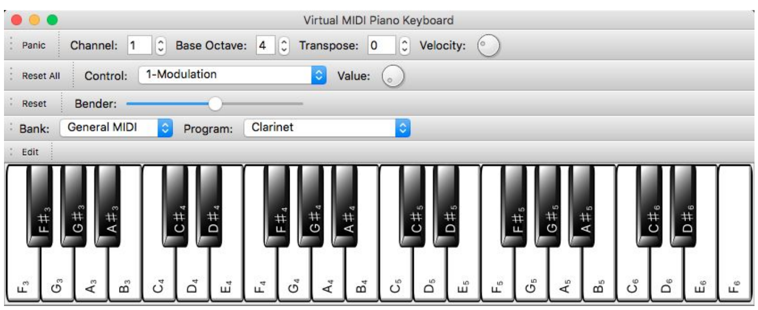
Four rods extend parallel to the clarinet from metal plate to metal plate in order to mount the solenoids. Three are above the clarinet with one broadside and two offset, and one rod is directly underneath the clarinet. The design is implemented for proper solenoid placement relative to the location of the keys that must be able to be pressed to play the full range of the clarinet. If keys are too close to one another for the solenoids to be placed adjacently, the solenoids can be mounted in a staggered pattern with adjusted throttles, or the solenoid can be placed elsewhere using an alternate fingering.

Fingerings

In order to create a seal between a pushed solenoid and key rung/open hole of the clarinet, soft rubber is cut into cylindrical sections. Once secured to the plunger of the solenoid, their flexibility allows a seal to be created in order to fully cover the hole and the key ring when applicable. If not completely covered, a poor seal is created and the following can happen: the sound produced is muffled, the pitch is incorrect, no sound is produced, or a squeak is produced. If two keys are too close to one another for two solenoids to be mounted adjacently, an alternate fingering may be used.

Note that not all of the soft rubber pads completely cover the holes; this is the case for some keys that use a metal and plastic rung, and for holes completely exposed such that pad used to cover the hole must be curved. Testing involving different flexibilities if rubbers was used, and the type chosen was the most flexible available. This still did not solve the above problem, so a design of using a second smaller ‘nipple’ of rubber was testing in order to ensure that the plastic ring was completely covered while the metal ring created a complete seal, however, this was unsuccessful. See Future Work for more.

All of the keys on the clarinet are controlled via MIDI, meaning that the control interface is quite general and can be easily connected to various input sources. In our demonstration, we had a laptop running both a virtual MIDI keyboard for live input as well as an instance of Ardour with several well-known pre-written songs. The nature of the Ardour software is such that one can download any MIDI file from the internet and, after transposition to the clarinet’s playable range, play it on the clarinet. We demoed this functionality with The Stars and Stripes Forever and Flight of the Bumblebee.



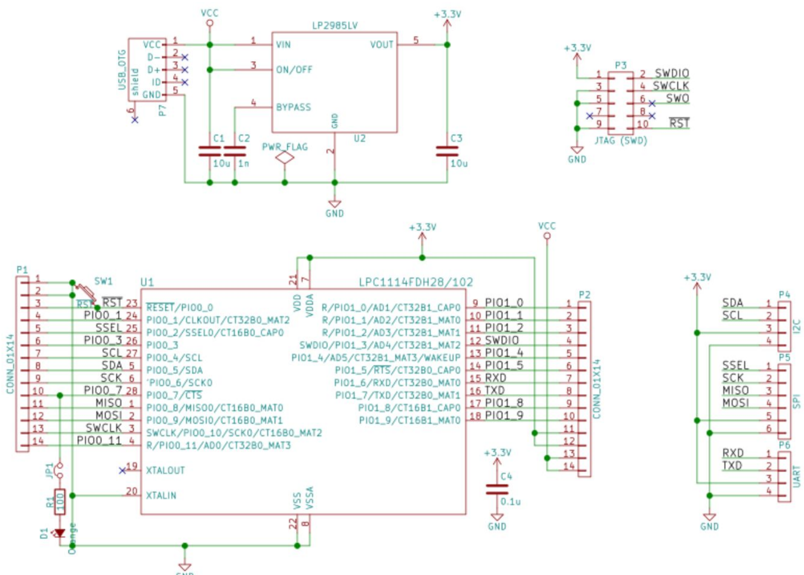
Virtual MIDI Keyboard

Ardour MIDI Sequencer

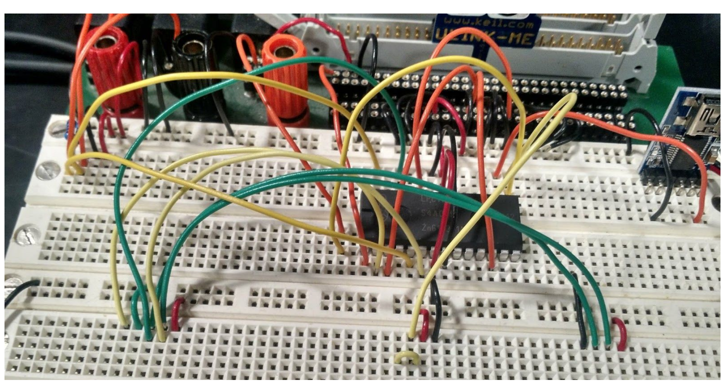
Hardware

Processor

The component responsible for controlling every part of the clarinet is a 32-bit ARM Cortex-M0 processor. The processor is connected to various peripherals and ultimately serves as the intermediary between the control inputs and the actuator outputs. In the future we will layout a dedicated PCB for the processor, voltage regulator, and all the required connectors. The hope is that this board will also be general purpose so that it can be used in future projects as well.



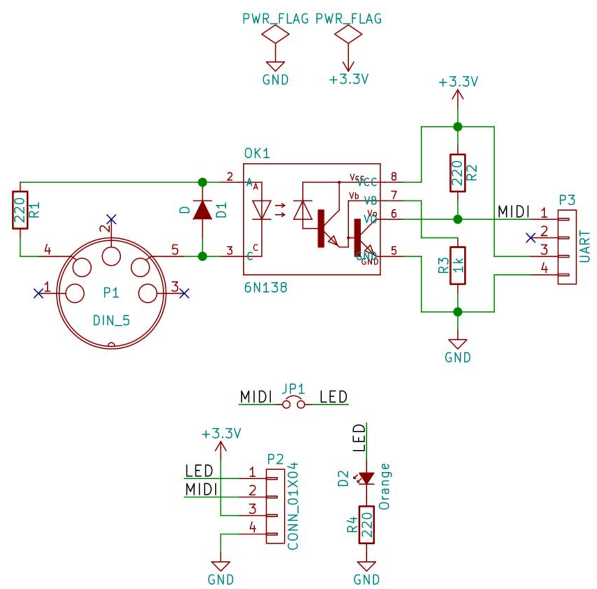
32-bit ARM Cortex-M0 Processor Schematic



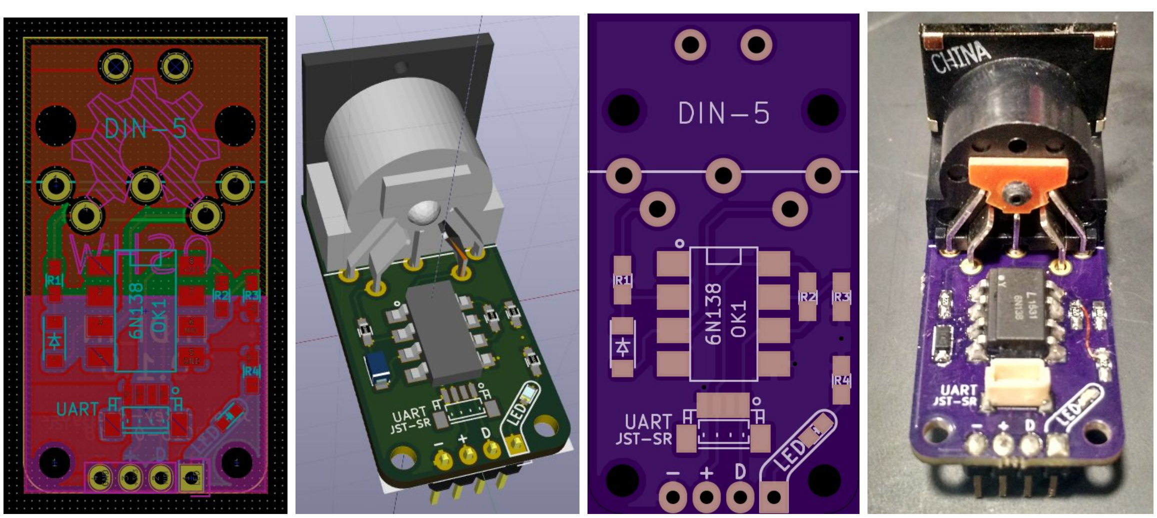
Breadboard Circuit

MIDI Interface

The clarinet is controlled via a MIDI interface, thereby allowing it to be connected to and played by any standard MIDI device. We created the MIDI PCB in KiCad and had it fabricated by the OSHPark board house. We soldered the components and tested the completed board in-house. Testing revealed that the board had been mistakenly designed such that the activity LED was normally ON, but a “green-wire” fix with a length of magnet wire was sufficient to fix the issue.



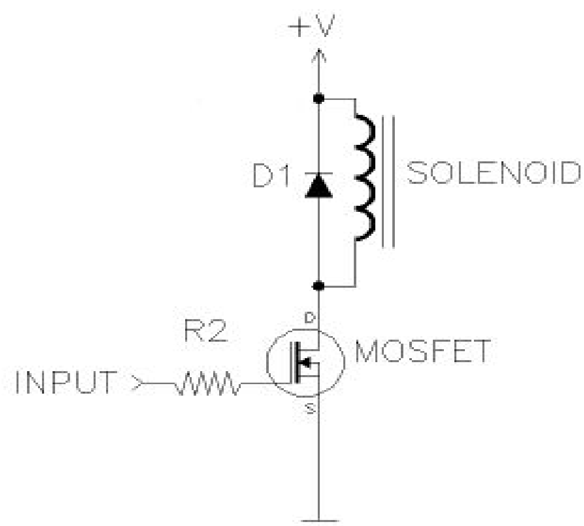
MIDI Interface Schematic



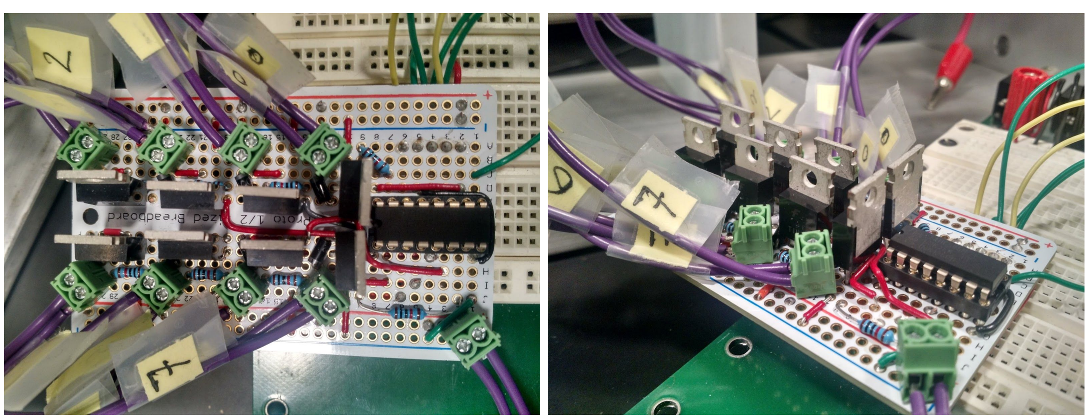
MIDI Interface Board

Solenoid Drivers

The solenoids, responsible for pushing all of the keys on the clarinet, are controlled by two controller boards. The boards are identical and each controls 8 of the solenoids. They have the simple transistor circuit shown below duplicated for every solenoid as well as a shift register with the parallel outputs connected to the input lines of the transistor circuits. The shift registers are then controlled by the ARM processor via SPI to essentially increase the number of outputs on the processor.



Transistor Drive Circuit

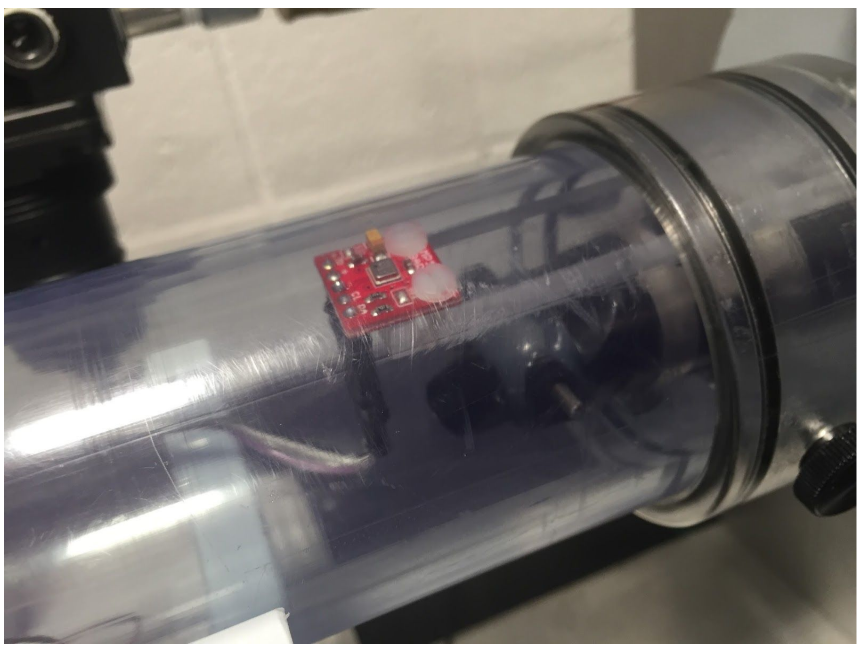


Solenoid Control Board

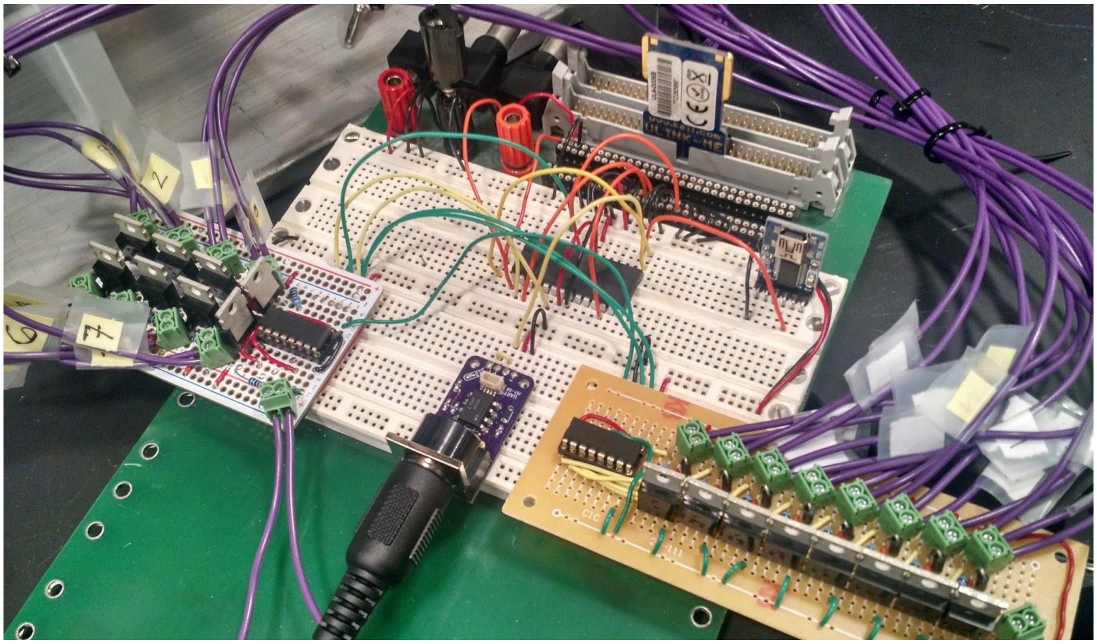
Pressure Sensor

The air chamber is ultimately responsible for the clarinet actually producing sound. A PV101-MA proportional valve was used, whose control signal would come from the PID controller output such that a current between 4-20mA is given to the valve in order to open or close it to create a desired air pressure in the chamber. This was implemented because each note requires a different air pressure in order to be produced, and the system must be relatively sophisticated and respond quickly. On the demo day of the fall semester of 2016, we believe there was a connection issue with power for the proportional valve, as determined by unsuccessful testing until one of the power lines was accidentally tugged, and the valve momentarily functioned properly. See Future Work for more.

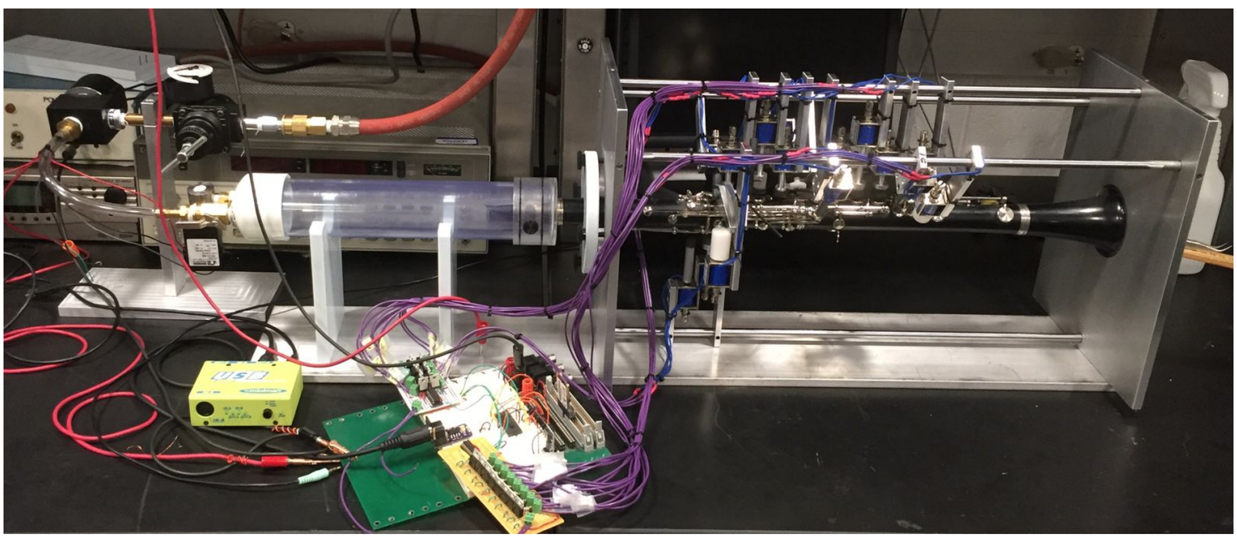
The pressure sensor is mounted in the air chamber via a plastic rectangular addition located vertically at the bottom of the chamber near the mouthpiece. Its orientation is to ensure as little air is disturbed as possible when being pushed through to the mouthpiece, but so that the air pressure measurement is as accurate as possible. See below for an image of the setup of the pressure sensor and images from the fall 2016 semester (no pressure sensor mounted/utilized).



Pressure Sensor in Chamber



Overall Control Circuit, FA2016



Full System, FA2016

Firmware

When writing the firmware we made an effort to modularize all of the interface drivers. This means that each one is very general and can be used in future code on the same microcontroller. Each firmware module is described below, and all code and hardware files have been uploaded to our project's git repository at https://github.com/0xdec/ECE395-Clarinet and will be continually updated.

Main

The main program code calls several initialization functions and then loops forever receiving and parsing MIDI messages and controlling both air pressure and finger solenoids. The iocon_register array is a static reference to the correct LPC_IOCON struct for each pin.

main.c

```
int main() {
  // Initialize all required peripherals
  system_init();
  // Configure MIDI to listen on channel 1
  MIDI_init(1);

  while (1) {
    // Call the receive function to handle incoming MIDI messages
    MIDI_receive();
  }

  return 0;
}
```

header.h

```
// Pin Registers
static __IO uint32_t *const iocon_register[2][12] = {
  {&LPC_IOCON->RESET_PIO0_0, &LPC_IOCON->PIO0_1, &LPC_IOCON->PIO0_2, &LPC_IOCON->PIO0_3,
&LPC_IOCON->PIO0_4, &LPC_IOCON->PIO0_5, &LPC_IOCON->PIO0_6, &LPC_IOCON->PIO0_7,
&LPC_IOCON->PIO0_8, &LPC_IOCON->PIO0_9, &LPC_IOCON->SWCLK_PIO0_10,
&LPC_IOCON->R_PIO0_11},
  {&LPC_IOCON->R_PIO1_0, &LPC_IOCON->R_PIO1_1, &LPC_IOCON->R_PIO1_2,
&LPC_IOCON->SWDIO_PIO1_3, &LPC_IOCON->PIO1_4, &LPC_IOCON->PIO1_5, &LPC_IOCON->PIO1_6,
&LPC_IOCON->PIO1_7, &LPC_IOCON->PIO1_8, &LPC_IOCON->PIO1_9, NULL, NULL}
};
```

System

The system module provides a simple initialization function that sets up the peripheral hardware clock. It also provides systick timer functions that utilize the systick interrupt.

```
uint32_t system_init(void);
uint32_t system_micros(void);
uint32_t system_millis(void);
```

GPIO

The GPIO module provides access to any microcontroller pin that is configured as GPIO. Pins can act as either inputs or outputs, and support various types of pull resistors.

```
void GPIO_init(void);
void GPIO_direction(uint8_t port, uint8_t pin, bool direction);
void GPIO_write(uint8_t port, uint8_t pin, bool state);
bool GPIO_read(uint8_t port, uint8_t pin);
```

I2C

The I2C module is a general purpose library for communication via I2C. It was written to spec based upon documents from NXP and uses state machines to decipher I2C statuses and take the appropriate actions.

```
void I2C_init(void);
uint8_t I2C_transmit(uint8_t address, uint8_t length, uint8_t* data);
uint8_t I2C_request(uint8_t address, uint8_t length);
uint8_t I2C_available(void);
uint8_t I2C_read(uint8_t index);
```

PWM

The PWM module provides functions for initialization, enabling, and disabling of the PWM timer, as well as a function for setting the desired pulse width.

```
void PWM_init(uint16_t period);
void PWM_enable(void);
void PWM_disable(void);
void PWM_width(uint16_t width);
```

SPI

The SPI module provides functions for initializing the SPI interface and sending data. It supports variable data sizes, from 4 to 16 bits. Support for receiving data via SPI is planned for the future.

```
void SPI_init(uint8_t size);
void SPI_transmit(uint16_t data);
```

UART

The UART module, often known simply as “serial”, provides functions for initializing the UART port and communicating bidirectionally. It is capable of various baud rates and has utilities for determining when data has been received.

```
void UART_init(uint32_t baudrate);
void UART_transmit(uint8_t data);
uint8_t UART_available(void);
uint8_t UART_receive(void);
```

BMP180

The BMP180 module utilizes I2C to communicate with the Bosch BMP180 pressure sensor. It was ported from Sparkfun's BMP180 library for Arduino.

```
bool BMP180_init(void);
uint8_t BMP180_start_temperature(void);
bool BMP180_temperature(double* T);
uint8_t BMP180_start_pressure(uint8_t oversampling);
bool BMP180_pressure(double* P, double* T);
double BMP180_sealevel(double P, double A);
double BMP180_altitude(double P, double P0);
```

Servo

The servo module is a wrapper for the PWM module that handles angle calculations for servo motors. It was formerly used to control a servo that was connected to the control knob of an air pressure regulator.

```
void servo_init(int8_t max);
void servo_pos(int8_t deg);
```

PID

The PID module implements a general purpose PID (Proportional, Integral, Derivative) controller that we use to accurately control pressure in the mouthpiece chamber. It was ported and modified from the Arduino PID library.

```
extern double PID_process, PID_manipulated, PID_setpoint;

void PID_init(double kp, double ki, double kd, bool direction);
bool PID_compute(void);
void PID_params(double kp, double ki, double kd);
void PID_limits(double min, double max);
void PID_period(uint32_t dt);
void PID_direction(bool direction);
void PID_mode(bool mode);
```

Pressure

The pressure module integrates the BMP180 pressure sensor with a PID controller in order to accurately regulate the air pressure in the mouthpiece chamber. It outputs data via an external DAC that in turn controls an op amp in order to create a 4-20mA current loop that controls a proportional valve.

```
bool pressure_init(void);
void pressure_set(double pressure);
void pressure_update(void);
```

Note

The note module is the core bit of code that handles decoding of MIDI notes into clarinet fingerings. It checks the requested note value to determine whether it falls within the clarinet's playable range, and then consults a lookup table to determine the corresponding fingering. Conveniently, only 16 solenoids are needed to play all of the notes in the clarinet's range. In order to convert each of the 37 note fingerings into a valid C array, we created a simple [web app]() that allowed us to create each fingering graphically, saving us a significant amount of time.

After consulting the lookup table the fingering is sent via the SPI interface as 16 bits of serial data. The SPI interface is connected to two daisy-chained 8-bit shift registers (74HC595) where the serial data is converted to parallel outputs to the solenoids. The note module also determines the volume of the requested note and encodes it to a pressure level relative to that note's base pressure.

```
void note_init(void);
void note_on(int8_t note, int8_t velocity);
void note_off(int8_t note);
void note_volume(int8_t note, int8_t volume);
void note_transpose(int8_t interval);
int8_t note_get(void);
```

MIDI

The MIDI module is a high-level interface that handles the actual receiving and parsing of MIDI messages. It is responsible for determining the type and length of any given message, and in turn calls the required note.c function when a message is completed.

```
void MIDI_init(uint8_t chan);
void MIDI_receive(void);
```

CONTINUATION

Spring 2016

All functionality besides the air chamber was successful. In experimentation, only the midrange of the clarinet was achieved, although loud and with poor tone. A threshold of air pressure exists such that when reached, sound can be produced for open G, and once played, pressure can be reduced while still producing the proper pitch. A nonlinear relationship between air pressure and pitch exists. For example, E (below open G) requires less air than an open G, but a D (below open G) requires more air than E. This can be tested by manually controlling the airflow into the chamber. The playable range is reached just beyond the squeaking ranges.

Being able to finely control the airflow and pressure in the chamber utilizing a regulator and a proportional valve or a similar mechanism is imperative for the lip-syncing clarinet to become a self-playing clarinet. The ability to electronically control this process with speed and accuracy pose as goals for future work on the project. Note that the actual playing range for a given reed strength is very small relative to pressure, and a pressure gauge would be useful for analysis and monitoring.[1]

Summer/Fall 2016

In order to gain further insight for the air flow set-up, contact was made with Joe Wolfe from the School of Physics of the University of New South Wales. He is part of a team from UNSW had successfully built a clarinet robot which won the Artemis Orchestra Competition in 2008. Given further documentation and setup tips, modifications such as inclusion of a dampening material to mitigate the nonlinear effects were tested.[2]

As expected, mimicking human behaviors while playing a clarinet such as varying embouchure and air flow proved to be the largest difficulties in the project. Surprisingly so, the effect of a musician’s vocal tract has minimal influence on good tone production.[3] Thus, when focusing on achieving a decent quality of sound, the team needed only consider the lip force and reed strength.

Further advancements in this project can include adding the ability to articulate, implementing more precise valves, and designing a better fingering system to showcase the clarinet and provide a more aesthetically pleasing product. Advanced additions to the project could be implementing some type of pitch bending, investigating further the quality of sound production by controlling the air temperature flowing into the clarinet, and introducing the ability to implement different stylistic aspects of music such as accents and marcatos.

Future Work

Some solenoids do not cover the holes properly, and testing of different materials and designs proved to be unsuccessful. We recommend an entirely different material be tested, possibly silicon-based or a similar material. It must be flexible enough to cover both a plastic and metal ring and fit to a curved surface, yet firm enough to create a proper and complete seal in a time efficient manner. Our group settled with the softest rubber material, currently on all of the solenoids requiring such, because it was the best method of all those tested.

A new proportional valve must be purchased; previously, a PV101-MA was used. We recommend looking into a more precise valve whose power and signal ground lines are separated (the PV101-MA connects the power and signal ground together). This would allow for simpler implementation of different systems for the 4-20mA signal. However, note that a new proportional valve would likely fall around $500. If a budget allows, a second proportional valve could be used to tune the system even more so that the air flowing in or out could be modified. We suspect this would help with the air pressure build-up upon starting. The leak used this semester was kept constant, more specifically, the leak valve was kept at a constant position (this method was used because it was financially unreasonable to purchase another proportional valve).

Misc. tests: We do not recommend adding in a dampening material to the air chamber for a more ‘uniform’ air pressure- loose cotton was tested with poor results. If a design modification to the lip force (reed and mouthpiece) system is planned, remember to keep a moderately ‘large’ surface area covered on the reed. This is modeled through musicians, as the lip is physically compressed and flattened against the reed. For a more specific measurement, lines can be drawn from a musician holding the proper embouchure around the mouthpiece and reed. This was the process followed for the project, and worked well for determining how wide the windings of the black elastic should be. Note that the system used for creating the lip force must also be kept in place well; hot glue was used in order to hold the elastic to the nail to the mouthpiece so that it does not slide up and cause the lip force to be applied too far up on the reed. The mouthpiece is curved and smooth, and we do not recommend shaving down the mouthpiece in any way.

Future work can also include advanced features listed previously, such as articulation and pitch bending. Pitch bending would involve very precise control of the fingerings themselves, adding a mechanical aspect (the pads covering the holes would need to have the ability to be ‘slid’ off with a controlled speed instead of simply pressed up or down). Pitch bending typically involves very precise lip force manipulations, but this would extensively complicate the air chamber design and mechanisms.

REFERENCES

[1] D. George *et al.*, “The clarinet: How blowing pressure, lip force, lip position and reed “hardness” affect pitch, sound level, and spectrum,” *PACS*, 43.75.Pq, 43.75.Yy, pp. 2247-2255, June 2013

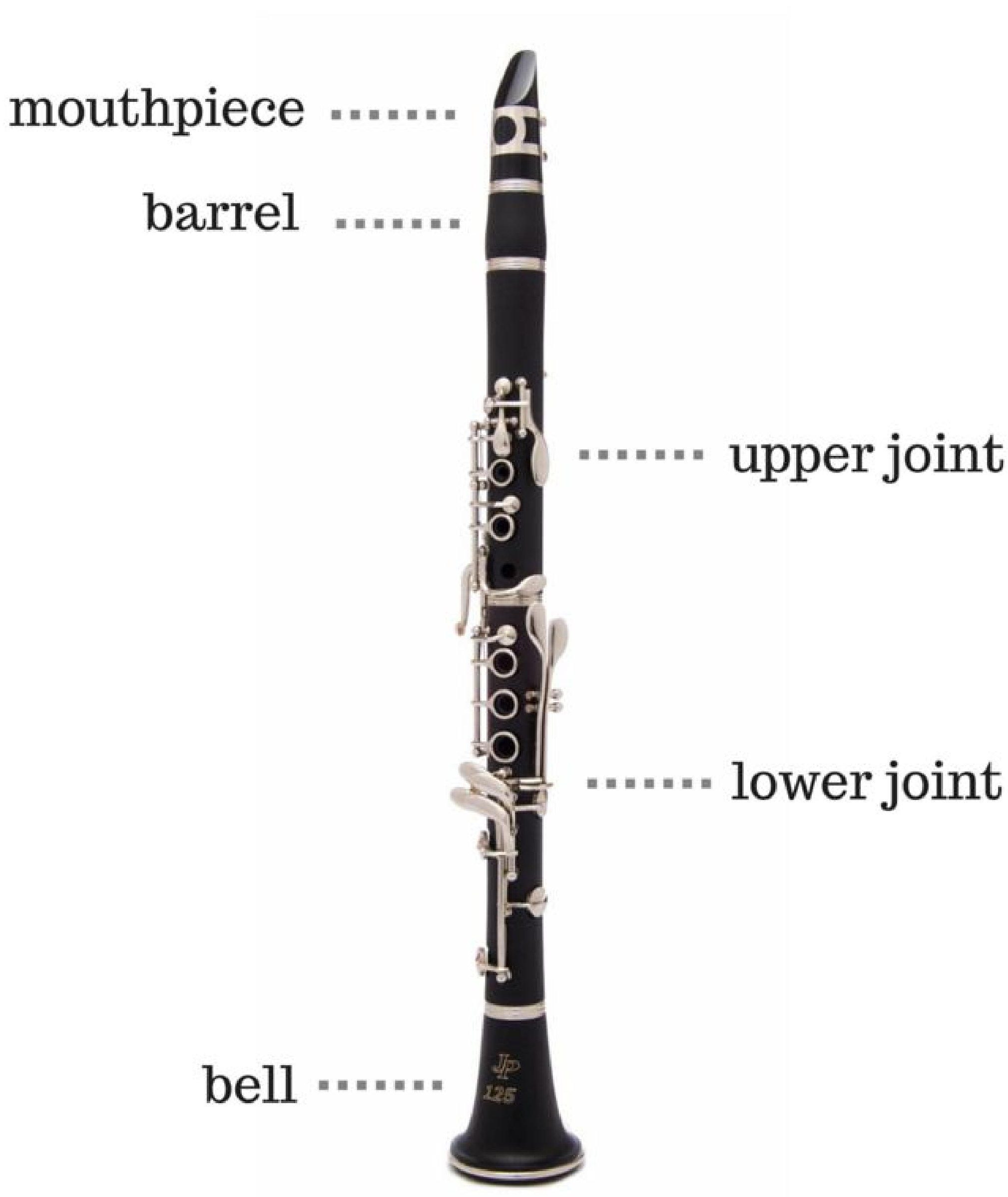
[2] J. Chen *et al.*, “Pitch bending and glissandi on the clarinet: Roles of the vocal tract and partial tone hole closure,” *PACS*, 43.75.Pq, 43.75.St, pp.1511-1520, June 2009

[3] J. Backus, “The effect of the player’s vocal tract on woodwind instrument tone,” *J. Acoust. Soc. Am.* 78, pp.17–20. July 1985.

APPENDIX

Clarinet Parts

Different portions of the clarinet will be referenced. An image of a clarinet and its corresponding parts are labeled below.



Musical Terms

Various acoustic and musical terms are also referenced; the table below gives their descriptions and relation to the project.

Embouchure	The combination of facial muscles and lip forming in order to create a proper seal on the mouthpiece of the clarinet. This is particularly important, as the lower lip exerts a force on the reed.
Fingering	Fingerings refer to the mapping of what keys are pressed down in order to play a certain note. The clarinet fingerings are utilized in MIDI and mount design/solenoid placement. Alternate fingerings exist, meaning that a single note can be played by multiple fingerings. Alternate fingerings are utilized due to physical space parameters of the solenoids. See Fingerings of Description for more.
Ligature	A ligature holds the reed to the mouthpiece of the instrument. The metal ligature used in this project wraps around the full mouthpiece and is secured with two screws, providing four total pressure points on the reed.
Pad/Seal	There are physical pads on many of the keys that are moved when a particular fingering is executed. They provide a necessary seal to cover holes in the clarinet. Human fingers act as pads to cover open holes of the clarinet, creating a seal when pressed against the key rung and/or hole. See Fingerings of Description for more.
Pitch	The percepted sound created by the instrument; it can be quantified in terms of frequency.
Reed	The reed lies between the mouthpiece and ligature of the clarinet and vibrates when disturbed by pressured air flow. This is fundamental for sound production in a reed woodwind instrument. A plastic reed is used in the project to ensure it does not dry out, as classic reeds are made of cane which must be moist in order to create the desired sound.
Squeak	A squeak is a an unpleasant high-pitched noise created that indicates an improper embouchure, air pressure, or seal exists.

Thank you to ECE 395 course director Lippold Haken and Teaching Assistants Chad Li and Aaron Smith. Thank you to ECE Electronic Services Shop and Machine Shop for their time and investment in the project. A special thanks to Scott A. McDonald for project advising, Glen W. Hedin for creating the chamber, and Greg Len Bennett for creating the project mount,solenoid throttles, seals, and chamber additions. Without the work and aide of these men, the project would not have reached such success.