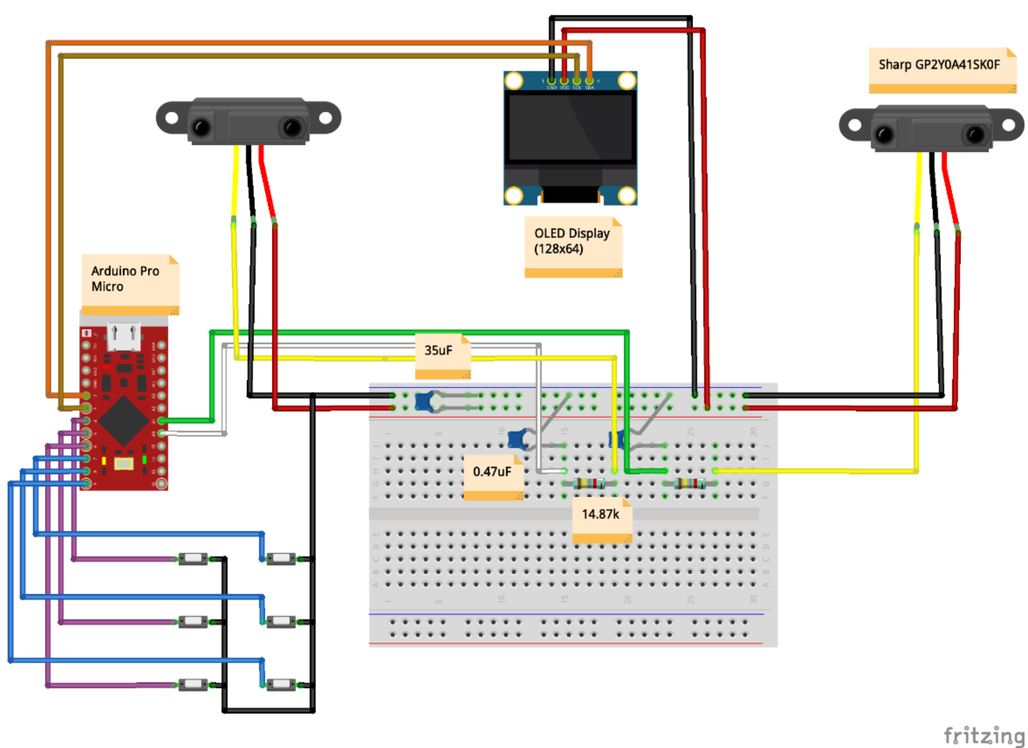
Theremidi

Puru Samal
AUDI 314
10th May 2021

Introduction.

The goal of my final project was to create a MIDI controller that simulates the Theremin while being capable of converting the gestures to MIDI data. Throughout the years, I have found myself spending hundreds of dollars on MIDI controllers and I never questioned it. But recently, I was introduced to the idea of creating one's own MIDI controllers, and how one can implement any functionality at a fraction of the price of commercial MIDI controllers. I took this project to create a controller with a level of expression that is not cheaply acquired and to further educate and explore this concept of making custom MIDI controllers. I was inspired by the Moog Theremini which costs around $400 and uses capacitive antennas to shape the sound. The Moog Theremini does have additional features such as a synth sound engine with multiple oscillators and onboard effects. But requiring only $40 to construct, The Theremidi is only one-tenth of the price and incorporates features such as MIDI class compliancy, pitch correction, and scale quantization. In the future, I would like to further tweak the sensors to get better hardware readings and better note-mapping, incorporate the ability to input user-defined scales, add a transpose feature and generate CV data to control external sources.

Schematic.



Hardware Components Used.

1x Arduino Pro Micro with ATmega32U4 chip: The brain of the entire device.

2x Infrared Proximity Sensors: Sensors that were used to convert and gestures to MIDI CC.

1x SSH1306 OLED display: Displays device state such as scale used, midi channel and octave.

6x Push buttons: Mapped to octave control, MIDI channel toggle and scales.

2x 10kΩ Resistors: Used to get a better hardware reading from the IR sensors.

2x 5kΩ Resistors: Used to get a better hardware reading from the IR sensors.

1x 0.1µF Capacitor: Used to get a better hardware reading from the IR sensors.

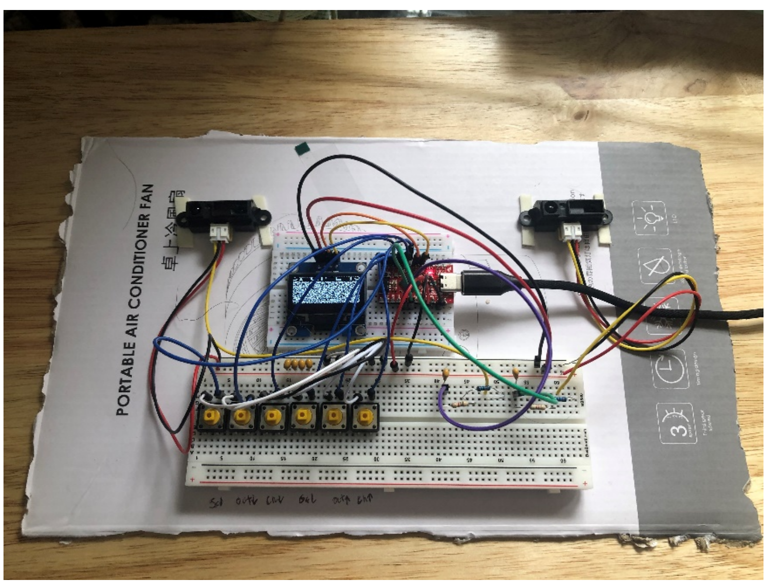
1x 0.47µF Capacitor: Used to get a better hardware reading from the IR sensors.

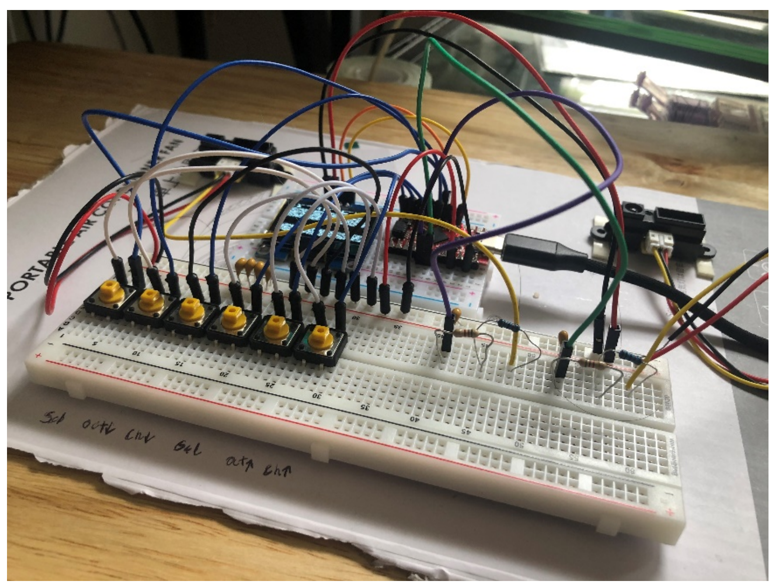
1x 4.7µF Capacitor: Used to get a better hardware reading from the IR sensors.

3x 10µF Capacitor: Used to get a better hardware reading from the IR sensors.

Clip Leads: Used to make all the connections.

2x Breadboard: Holds all the components and makes the connections.





Software Components Used.

Arduino IDE: To write, compile, debug and execute the code into the Arduino Board.

Hairless MIDI: To convert serial protocol to MIDI notes and control changes.

LoopBe: To create a virtual MIDI output that received MIDI information from Hairless MIDI.

Ableton Live: The map the interactions with the Theremidi to sounds using MIDI data.

The code.

The Arduino code consisted of a Master code which was the DIY MIDI Controller code made by Gustavo Silveira. The potentiometer portion of the code was amended and restructured to accurately read the IR sensor and appropriately map these hardware readings. This portion of the code was inspired by Moritz Simon Geist's MIDI Theremin. That sketch used an Ultrasonic Sensor to produce a MIDI out signal was worked properly in conjunction with my IR sensors after tweaking the variables.

I also constructed debugging codes for the buttons and the IR sensor from scratch that tested the individual workings of the components and would print the data in the serial monitor.

Mapping.

The left IR Sensor (IR1) outputs MIDI CC 7 and can be mapped to any parameter that takes MIDI input.

The right IR sensor (IR0) controls MIDI note number and is controls the pitch.
Hand Down: Higher Pitch
Hand Up: Lower Pitch

There are two modes:

Scale Mode: The output of IR0 has been quantized to 4 scales: Ionian, Minor Harmonic, Minor Pentatonic, Whole Tone.

Pitch bend mode: Emulates the traditional glissando-like effect of a Theremin.

Sensor	Input Min/Max	Output Min/Max	CC
IR0	90, 530	0, 127	--
IR1	90, 530	0,13	7

Use Case.

The following use cases would be feasible in my opinion:

Performance: Because of its ability to convert gestures to MIDI data and the in-built pitch correction, The MIDI Theremin would be a welcome addition as a live performance tool as controller for virtual instruments as well as physical instruments capable of receiving MIDI IN.

Studio: Because of its ability to convert gestures to MIDI data and the in-built pitch correction, The MIDI Theremin can be mapped to almost any parameter in a DAW environment. The ability to seamlessly glide between notes to generate some interesting melodic ideas would be inspiring as a compositional tool. Also, the ability to map gestures to any parameters would open up more organic sound design possibilities.

Educational: Theremins are historically significant as being one of the first synthesizers ever created. But they are very expensive, and this $40 substitute would serve as a nice alternative in a classroom environment to showcase the gestures associated with a Theremin. It can also be used as a cool little sidepiece to demonstrate automation, sound design and synthesis parameters.

Problems	**Solutions**
COM Port confusion while using Windows. Sketch would compile but not upload.	Resolved by uploading the same code through Mac OS. Cause still unknown.
Unstable hardware reading.	Resolved through experimentation with different resistor and capacitor values.
Serial to MIDI communication error.	Resolved by changing Baud rate, ‘locking’ COM port in Windows device manager and restarting Serial-to-MIDI bridge.
OLED display did not configure properly.	Still unresolved but cause identified. Libraries used in code for SSH1306 type of OLED display. OLED display used is SSH1106. Need to rewrite code according to U8g2 library or get an SSH1306 display.
Could not input more than 4 scales.	Still unresolved but cause identified. The piece of code required to incorporate scales took up too much memory. Need to use a Arduino board with more onboard memory to incorporate more scales.
Sketch stuck during uploading.	Fixed by installing 3rd party board drivers from Sparkfun’s official website.

Conclusion.

The Theremidi is a MIDI class compliant Theremin-like device that captures and converts hand gestures into MIDI notes and control changes. The Theremidi uses two infrared distance sensors, instead of the capacitive antennas, and can track the position of your hands in the air. This position is then converted into MIDI notes, or pitch bend - right hand, or MIDI CC - left hand. There is also an OLED display and some buttons that allow you to change octave, transpose, MIDI channel, and switch between scales and pitch bend! While being involved with this project, I learnt more about Arduinos, hardware implementation and serial protocol. I also leant more about coding in the IDE, and capturing, converting and mapping analog signals to digital data. I was able to get the Theremidi to work as intended but during the process, was also able to think of several important features that would be useful to implement down the road such as tweaking the sensors and code to get better hardware readings and better note-mapping, incorporating the ability to input user-defined scales, add a transpose feature and generate CV data to control external sources.