Sensing String Displacement as a Control Modality: Sensor Design and Implementation

by

Silvia Elena Knappe

B.S. Electrical Engineering and Computer Science, Massachusetts Institute of Technology (2021)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author . . . .
Department of Electrical Engineering and Computer Science
May 11, 2022

Certified by . . . .
Ian Hattwick
Lecturer
Thesis Supervisor

Certified by . . . .
Eran Egozy
Professor of the Practice
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Sensing String Displacement as a Control Modality: Sensor Design and Implementation

by

Silvia Elena Knappe

Submitted to the Department of Electrical Engineering and Computer Science
on May 11, 2022, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

While stringed instruments are a major part of musical practice across cultures, DMIs in which strings are the primary control interface remain rare. In an effort to support the creation of new stringed DMIs we describe an approach to instrument design in which the static 2-dimensional displacement of strings is a primary control modality. We review a variety of sensing techniques for this application, including optoelectronic, electro-magnetic field, resistive, and force. After establishing a set of design criteria we describe our design process as well as implementation and analysis of a transmission mode optoelectronic sensor.

Thesis Supervisor: Ian Hattwick
Title: Lecturer

Thesis Supervisor: Eran Egozy
Title: Professor of the Practice

Acknowledgments

This thesis project would not have been possible without all of my mentors, friends, and family.

In particular, I would like to thank my thesis supervisor, Ian Hattwick, for his continued support, patience, and enthusiasm. My excitement and passion for my project grew every time we met. A big thank you also to Eran Egozy for agreeing to co-supervise this project.

I would also like to thank Adam Hartz for his guidance throughout both my undergraduate and graduate endeavours. Thank you for the opportunity to be a teaching assistant for 6.009, and most of all, thank you for always being there to chat and looking out for me.

Without the resources and wonderful staff members and co-mentors at Project Manus, this project would not have been realized. I am grateful to have had the opportunity to be part of such a welcoming community.

The encouragement and support from my close friends was a huge help throughout my time at MIT. A special thanks to Andy, my late-night study and project friend.

Lastly, but certainly not least, I would like to thank my partner, sister, and especially my parents. Without their love and support, I would truly not be where I am today. Danke Papa, Gracias Mamá.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

The design, creation, and use of digital musical instruments (or DMIs) is a research field situated at the intersection of computer music and human-computer interaction. While research on this topic is published in a variety of forums, the most visible is the International Conference on New Interfaces for Musical Expression (NIME).

Much of the focus of the NIME community is on the creation of bespoke controllers, which itself largely depends upon the knowledge and resources of the DIY electronics community. However, while general knowledge on sensor implementations is readily available, research into the design of sensors dedicated for use in musical controllers remains on-going [22].

The research in this thesis addresses one specific sensor application, focusing on the displacement of a musical string as a primary control modality for computer music performance.

Acoustic stringed instruments are ubiquitous across many musical cultures of the world, and yet the use of strings as a primary interaction element for DMIs is relatively rare within both the NIME literature and electronic music performance in general. Commercially available instruments have tended to follow traditional form factors, and either use indirect sensing [31] to generate control signals, or fall within Miranda & Wanderley's classifications of 'instrument-like' controllers [23], which tend to emphasize techniques associated with acoustic instrument performance. This emphasis continues in work such as Temprano and McPherson's use of TMR angle sensors intended to capture gestures related to guitarist's left and right handed performance techniques [5].

There is no doubt that work of this type remains highly relevant, but in addition to 'traditional' acoustic interactions with strings, like plucking, bowing, and striking, DMIs also have the ability to utilize non-acoustic interactions for control of sound synthesis. In the same way in which the current generation of keyboard-inspired controllers (such as the Haken Continuum[1], LinnStrument[2], Seaboard[3], Soundplane[4], etc.) have established a model of generating pressure or displacement on the X-, Y-, and Z-axes, the approach we take here is to conceptualize stringed instruments as arrays of physical elements capable of 2-dimensional static displacement orthogonal to string position, with a potential 3rd axis consisting of either the position of peak displacement or of pressure along the axis of the string.

This opens up a new world of possibilities for the design of stringed DMIs, using the strings to directly generate control signals instead of as a physical sound source. The decoupling of control interface from sound synthesis allows for interactions not typically associated with traditional stringed instrument performance, and also means that the sounds the instrument can produce are independent of the physical qualities of a string This makes tactile feel and string tension the primary considerations for choosing which type of string to use.

In this thesis project, we will discuss our work in designing and implementing a system for sensing 2-dimensional string displacement in the context of DMIs. There are many components related to the sensor design process and integrating the sensor into a DMI. This thesis project covers the sensor design and implementation process and signal processing, but does not delve into mapping sensor control signals into sound.

We will begin in chapter 2 with the proposal of a set of design requirements and give an overview of sensing techniques for this application. In chapter 3 we will

[1]https://www.hakenaudio.com/
[2]https://www.rogerlinndesign.com/linnstrument
[3]https://roli.com/products/seaboard/rise2
[4]https://madronalabs.com/soundplane

review examples of example instruments from literature. We will then discuss the sensor mechanical design process in chapter 4, signal processing pipeline in chapter 5, and sensor results in chapter 6. Chapter 7 will detail the use of our sensor in the context of a musical instrument and we conclude with chapter 8.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Design Considerations

The primary goal of this research is the development of a practical sensor for the creation of digital musical instruments. Several publications within the computer music and NIME literature have set forth principles for the design of DMIs, generally drawn from their author's personal experiences [27, 20, 10].

Particularly relevant is the framework for designing hardware systems for professional artistic productions presented by Ian Hattwick in his Ph.D. dissertation [15]. Within this framework Hattwick describes seven design aspects, four of which influenced the design specifications for the work carried out in this thesis: functionality, manufacturing, robustness, and reusability. Associated with these aspects are sets of design principles, itemized in Table 2.1.

This section of the thesis puts for a set of design considerations, influenced by the principles drawn from the literature as well as the the primary goal described above.

2.1 Specifications for the design of a musical string sensor.

1. The sensor design should support a variety of digital interface designs.

    (a) Should be cost-effective and easily manufacturable with commonly available digital manufacturing device such as 3D printers and lasercutters.

Principles of functionality	
Novel solutions are cool, but risky	
Re-use existing systems when possible, but be prepared to start from scratch	
It is more important that it work than how well it works	
Perform quantitative tests of the system when possible	
Principles of manufacturability	
Use appropriate manufacturing techniques	
Begin manufacturing early	
Identify opportunities to speed manufacturing	
Principles of robustness	
Repairability vs. replaceability	
Pay attention to material properties and points of failure	
Learn and use standard techniques for protecting electronics	
Principles of reusability	
Keep an eye towards future applications	
Keep documentation of the design process as well as the system	
Clarify the possibilities of system reuse from the beginning	

Table 2.1: An excerpt of relevant design principles presented in [15]

(b) Should be small enough to easily implement in a DMI with multiple strings, supporting a minimum of 1 cm string-to-string distance.

(c) Robust mechanical integration with structure of the interface, allowing for strong physical interaction between the instrument and its player without affecting the mechanical stability of the sensor mechanism.

(d) Not limited to ferromagnetic strings; should be compatible with other common string compositions such as nylon or gut.

2. Effective Sensing Characteristics

(a) The primary goal is to be able to differentiate displacement in both directions orthogonal to the length of the string.

(b) Accurate and precise measurement of displacement within expected range of motion in order to for sensor signal values to correspond to the musicians' perception of string displacement.

(c) Sensitive to very small displacements without excessive noise.

(d) Ability to generate linear values for string displacement, and avoid non-monotonic regions. Monotonicity is defined as a function that is entirely non-increasing or entirely non-decreasing. We would like to have this relationship in the function of output voltage vs string position in order to uniquely identify string position based on output voltage.

(e) Minimal cross-talk between strings, in which the movement of one string causes a shift in sensor values of an adjacent string.

3. Support for sensing oscillation

   (a) While string vibration is not intended to be used as a sound source and is not the primary goal of this research, indirect sensing of string interactions will be useful and should be supported.

   (b) An open question remains as to what frequency resolution is sufficient to allow for meaningful analysis of oscillating signals.

2.2 Sensing for Physical Interaction with Strings

Various sensor types have been used for stringed interfaces, depending on the type of desired interaction. In this section we discuss the application of different sensors sensing string displacement, which we define as a controlled movement of the string away from its resting position and orthogonal to the string's length.

2.2.1 Sensor Configurations

String displacement can be described in either cartesian or polar coordinates. In practice, most sensor implementations will provide a cartesian perspective, generally measuring the distance of the string to two identical sensors. Figure 2-1 shows one configuration in which the sensing areas of the two sensors are parallel.

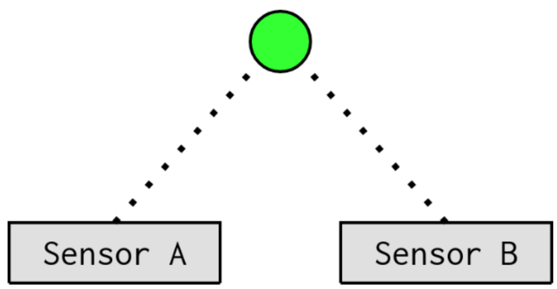


Figure 2-1: Typical sensor application measuring the distance of a string from two identical sensors with parallel sensing directions.

Primary considerations for this configuration are that the sensors can accurately measure the full displacement range of the string without becoming either non-monotonic or exiting effective sensing range, and also that the sensor output maintains sufficient resolution as the distance and angle of the string-sensor vector changes.

It should be noted that the 2-dimensional displacement in figure 2-1 above will cause the resolution of the sensors to deteriorate as the string moves further away from the sensors. Figure 2-2 shows a different configuration for sensors, each of which is focused on one direction of displacement. The configuration shown in figure 2-2 mitigates the decrease in resolution by angling the sensors to keep the string within a good sensing range.

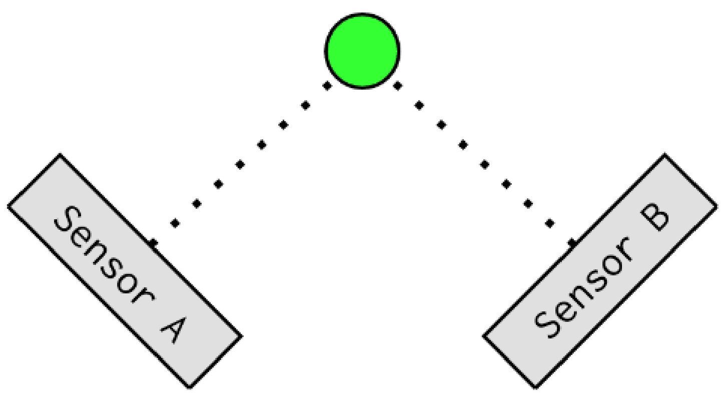


Figure 2-2: Perpendicular arrangement of identical sensors. Suitable for sensors which are primarily sensitive to string displacement in one direction.

One possible approach to sensing polar coordinates would be a modification to the STRIMIDILATOR [2], in which the displacement of the string is sensed using a linear transducer which is free to rotate. The angle can be measured using a rotary encoder.

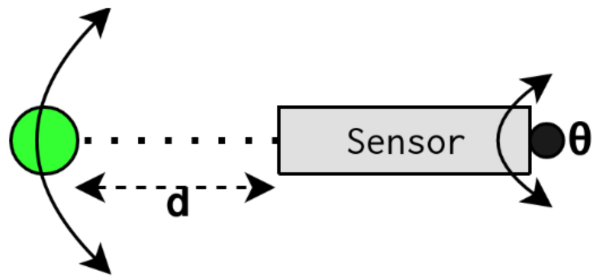


Figure 2-3: Polar position sensor, in which the displacement d is measured using a linear transducer which is able to rotate. The transducer is fixed at one end to a rotary encoder, which is able to measure angle θ.

2.2.2 Floating Sensor Elements

Some sensing techniques, such as potentiometers, require that the sensor be mechanically coupled to the string. In general, non-contact sensors like optical or magnetic sensors will be preferable as they do not interfere with the string's mechanical properties and also are less prone to failure. However, in some cases the string alone is not adequate for non-contact sensing techniques, e.g. if the string is non-ferrous for magnetic field sensing, or the string is too thin for effective optical sensing.

One alternative for these cases is to add a floating sensor element to the string. Attaching a magnet will make any string suitable for magnetic field sensing (as in the Global String [29]); a similar approach may work for optical sensors [16]. The challenge in these cases is to ensure a consistent relationship between the string, the floating sensor element, and the sensor itself.

Early experiments with floating sensor elements (such as the prototype in figure 2-4) showed great potential in being compatible with various kinds of strings, as the sensor is not sensing string motion directly. However, we found that a consistent relationship between string motion and sensor output was difficult to obtain.

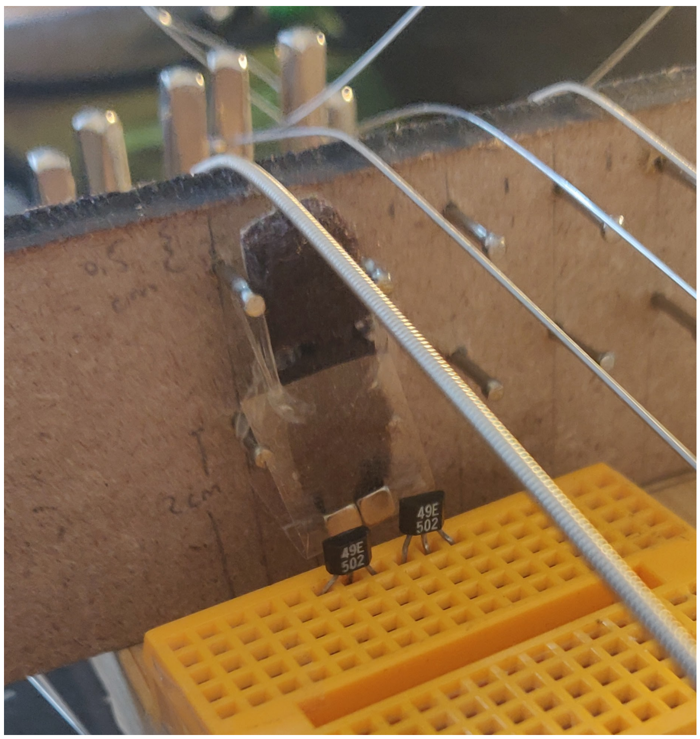


Figure 2-4: An early prototype of a floating sensor using magnets and hall effect sensors. Nails act as a support for the floating bridge which has magnets fixed to it. Hall effect sensors are used to sense the magnets on the floating bridge.

2.2.3 Electromagnetic Field (EMF) Sensing

Magnetic pickups are commonly used for electric stringed instruments, and use the oscillation of ferromagnetic strings within a magnetic field to generate an electrical current [24, 17]. While this technique does not detect static displacements, directly measuring the magnetic field is possible using hall effect sensors, a technique used in several of the instruments described in chapter 3.

Changes to the magnetic field can be created most easily by changing the relative position of a magnet and the sensor, most obviously by attaching the magnet to the string. An alternative approach is to use a biasing magnet behind the hall effect sensor, which will create a magnetic field in front of the sensor. The movement of a ferromagnetic string within this field will generate a magnetic field variation, although the amplitude of this variation is quite small. A third approach would be to run a current through a conductive string to create an electromagnetic field; however, this may require a considerable amount of current to generate a magnetic field of sufficient strength.

A similar approach is electric field sensing [26, 28], in which an AC current is applied to a transmitter, which then induces a current in a receiver with an amplitude relative to the distance between the transmitter and receiver. Similarly to magnetic field sensors, the electric field can be transmitted, received, or perturbed by a ferromagnetic string.

This is also related to capacitive sensing, useful for detecting contact with a string, proximity, and potentially also string displacement. The most common method is to periodically apply an electrical charge to a string, and then measure the time it takes to discharge. A finger touching the string will act as a capacitor and affect the discharge time [30].

Limitations for EMF sensors are that the strings need to be made out of a conductive material, compensating for interference by the performer's body may be necessary, and the time resolution for capacitive sensing is limited by the maximum discharge time.

2.2.4 Optical Sensors

Optical pickups use infrared light emitters and phototransistors to detect string oscillation [14, 19, 25]. The two main methods found for sensing with optoelectronic methods are either by placing the string between the emitter and phototransistor to interrupt the light (transmission mode), or by using the string to reflect the light from the emitter onto the phototransistor (reflection mode).

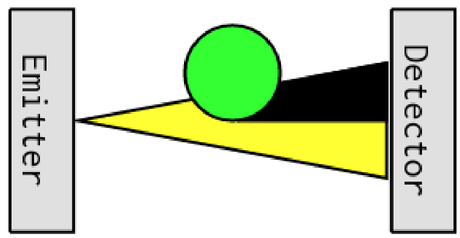


Figure 2-5: Optical sensor in transmission mode

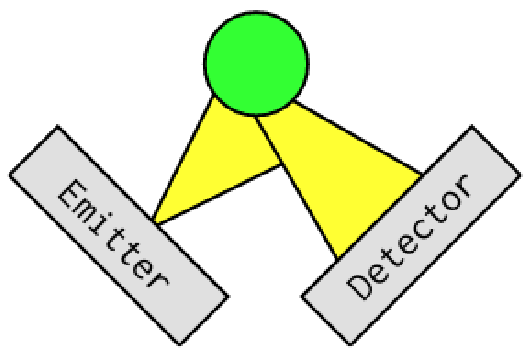


Figure 2-6: Optical sensor in reflection mode

While most optoelectronic sensing has been used for detecting single-axis oscillation, they have also been used to detect magnitude and angle of string displacement in transmission mode [8], although not in the context of a DMI.

2.2.5 Mechanical Sensors

Variable resistors [13] and force sensors [21] have been used to sense string displacement. Variable resistors will generally have a movable element whose position determines the resistance, as is the case in linear transducers or faders. Force sensitive resistors (FSRs) are made of materials that change in resistivity as strain is applied. Both FSRs and variable resistors are cost effective sensors; however, they are both physically large sensors, so it may be difficult to make a compact sensor system using these components. In the case of variable resistors, the moving parts may limit the lifespan of the sensor. For force sensors, a drawback is that force measurement doesn't necessarily translate to how much the string is displaced.

2.2.6 Piezoelectric Sensing

Piezoelectric transducers are capable of sensing string oscillation [29, 11] but are not suitable for sensing string displacement as they are only sensitive to changes in deformation. One example of 3-axis oscillation sensing was developed by Freed & Isvan [12].

2.3 Conclusion

After consideration of all the available and relevant sensing options, we decided to choose optical sensing for our string sensor because of its compatibility with non-ferromagnetic strings, ease of implementation, and capability of sensing both string displacement as well as oscillation. While we could have used a floating sensor element containing a ferromagnetic material in order to make EMF sensing work with non-ferromagnetic strings, the mechanical complexity of the floating sensor element was undesirable. Optical sensors supported the most goals from our list of specifications outlined in 2.1.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Example Instruments

In this section, we discuss existing string-based DMIs and the sensor systems that they use in order to explore different approaches for our sensor implementation.

3.1 The Web, the Finger Web, and Soundnet

The Web [18] and its counterparts the Finger Web and Soundnet [6, 7] are instruments that use multiple strings linked together resemblant of a spiderweb. When one or more strings are pulled the tension of multiple strings in the Web changes due to the strings in the Web being mechanically connected. The end of each string is attached to an assembly consisting of a magnet supported by a spring. A hall effect sensor detects the movement of the magnet when tension is applied to a string.

The Web was designed for sensing only string tension and is therefore insensitive to displacement direction. We also expect that the design would be be mostly insensitive to string oscillations.

3.2 Global String

The Global String [29] is an instrument that consists of 2 parts connected via network, so the two parts can be arbitrarily far apart. Each part consists of a single large string that takes up an entire room. When a user interacts with one part of the instrument, sensors on the string convert the string motion to a control signal used for sound synthesis as well as for inducing vibration in the string belonging to the other part.

A magnet is attached to one end to the string and low-frequency displacement data in 2 axes is sensed via Hall effect sensors in a perpendicular arrangement. Piezoelectric transducers are used to detect high-frequency oscillations.

3.3 STRIMIDILATOR

The STRIMIDILATOR [2] is a 4-stringed DMI that senses both displacement and oscillation. Displacement sensing is separated from oscillation by having the middle two strings only used for displacement, and the outer two strings for oscillation. For sensing oscillation, an electromagnetic pickup was used.

Linear transducers are used to detect overall string displacement. The tip of a linear transducer is attached to each displacement sensitive string at the string's midpoint. One end of the transducer is attached to a pivot, leaving it free to rotate. This method of sensing outputs the magnitude of the string displacement, but a sensor measuring the angle of rotation is not included (as opposed to the polar sensing configuration described in figure 2-3). In addition, since the transducer is mechanically attached to the string's midpoint, the string's natural ability to oscillate is hindered.

3.4 Manipuller I and II

The Manipullers [4, 3] are DMIs that use force-sensing resistors (FSRs) to measure tension on multiple strings. The Manipuller I features 4 parallel strings arranged in a square pattern while the Manipuller II has strings arranged around a ring in a dreamcatcher-like manner, intersecting with each other similar to the Web. Like the Web, each string is primarily sensitive to string tension; however, the Manipullers combine the sensor readings from all of the strings to create a single displacement vector (and also for other kinds of gesture sensing). Both Manipullers I and II use the FSRs not only to detect whether a string is being pulled or released, but also to detect multiple strings being pulled at once, and to detect the direction the strings are collectively being pulled in by using known string positioning to generate a coordinate system.

The Manipuller II was designed for string displacement sensing and gesture sensing. Sensing oscillation may be difficult as pressure variations of an oscillating string may be quite small (similar to the sensor in the Web).

3.5 Conclusion

While all of the sensor implementations discussed in this section work well for their intended instruments, they are not optimal for what we are trying to accomplish with our sensor. Particularly the size of the sensing elements in the Global String and Webs are too large, and some other sensing elements either only detected string displacement or string oscillation, but not both. This meant that our sensor implementation would not be able to take much inspiration from existing sensors and we would need to create a novel sensor.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Sensor Prototyping and Design

As discussed in 2.3, after evaluating different sensor techniques and existing approaches, we chose to implement a prototype using infrared (IR) optical sensors. We chose optical sensing over other sensing approaches such as hall effect sensors and linear transducers because of their ability to sense displacement without making contact with the string and their compatibility with non-ferromagnetic strings. Our ultimate goal is to create a design whose size, complexity, and price is suitable for use in instruments with relatively large numbers of strings, e.g. harp-type instruments, and which follow the design considerations in section 2.1 provided above.

4.1 Initial PCB Design

For our prototype we arranged 2 IR emitter-phototransistor pairs in a perpendicular transmission mode (figure 2-5) arrangement to sense displacement in 2 axes. Initial tests were done with through-hole emitters and phototransistors on a breadboard. While these tests showed that optical sensing was promising, more testing was needed as the position of the sensors were not firmly fixed in place, making it difficult to acquire clean data. Designing a PCB with surface mount components became the best option for acquiring predictable and stable sensor data. We designed our PCB to house only the emitters and phototransistors so that any further circuitry was still accessible and easy to manipulate.

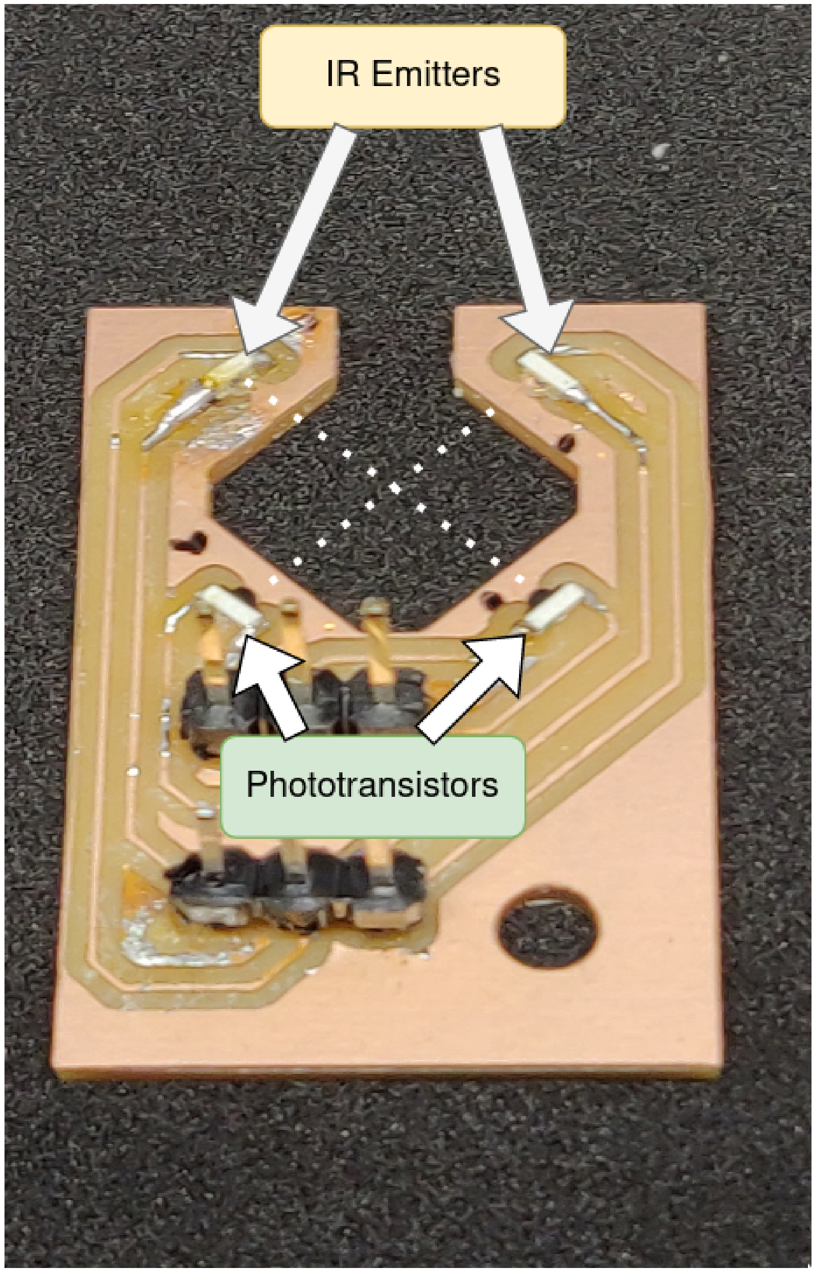


Figure 4-1: Close view of the sensor PCB showing the perpendicular layout of emitters and phototransistors

Underneath the IR emitters and phototransistors are 2 rows of 3-pin headers to power and read out from the PCB. We used a generic microcontroller to interface with the sensor as shown in the figures 4-2, 4-3, and 4-4. One row of header pins was used to provide power to the phototransistors as well as to access the output values of the phototransistors. The other row of headers was used to provide power to the

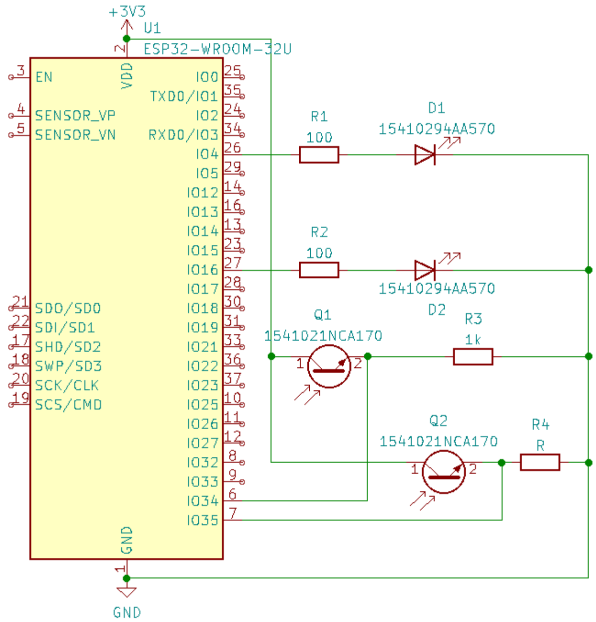


Figure 4-2: Initial schematic of the sensor showing one sensor hooked up to a microcontroller

infrared emitters as well as to control whether the emitters are turned on or off.

Since rapid turnaround time was required to test the initial PCB design, we decided on milling the PCB in-house using an Othermill [1]. The milling process on the Othermill was greatly simplified because of the single-sided PCB design, but was unable to include vias or any silkscreening/soldermask on the PCB. The initial PCB was 2cm in width and 3cm in height.

4.1.1 Initial Testing Procedure

We designed a simple one-string interface prototype to test the sensor. The string is held in place in both ends by Floyd-Rose guitar locking nuts [2]. On the right side, the end of the string is attached to a dulcimer tuning pin used to tension the string. On the left side, the sensor prototype is fixed in place as seen in Figure 4-5. The positioning of the sensor was done by hand until the sensor data seemed optimal; this proved quite a challenge and it became clear that accurate sensor positioning is

[1]Has since been discontinued, acquired by Bantam Tools: https://www.bantamtools.com/
[2]https://www.floydrose.com/products/original-locking-nut?variant=30511209490

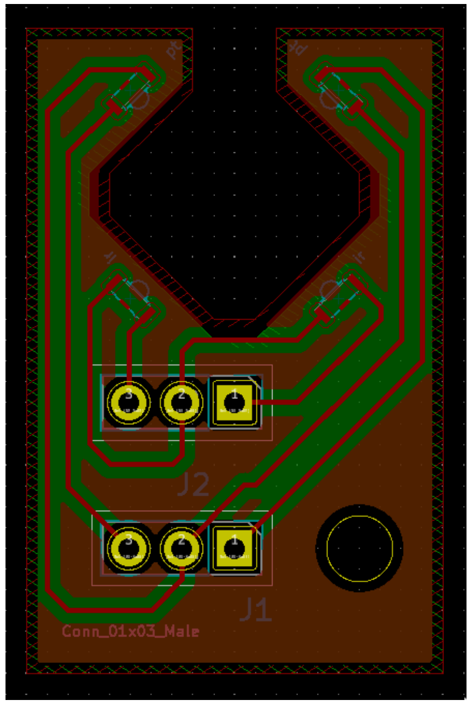


Figure 4-3: Initial PCB design for the sensor, sensors are laid out in an orthogonal configuration

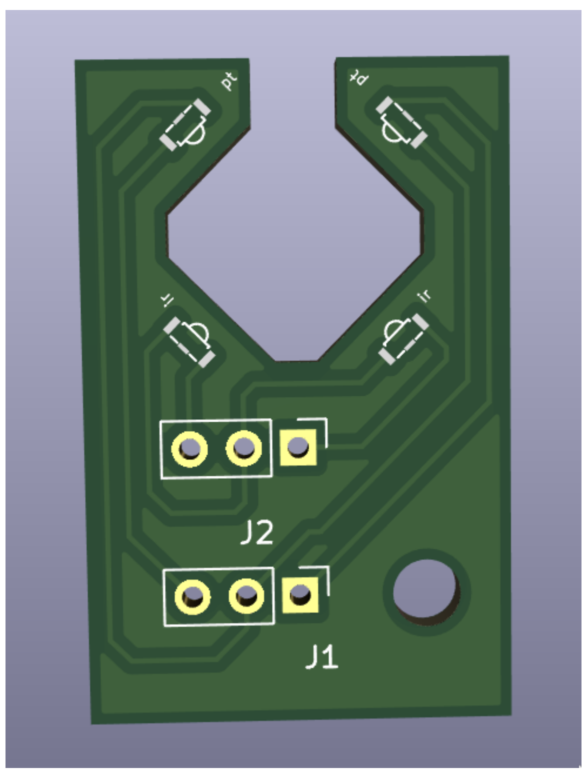


Figure 4-4: 3D model of the PCB prototype

a critical issue. For the initial measurements, the sensor was tilted 45° so that the process of calibrating the sensor by hand was simplified.

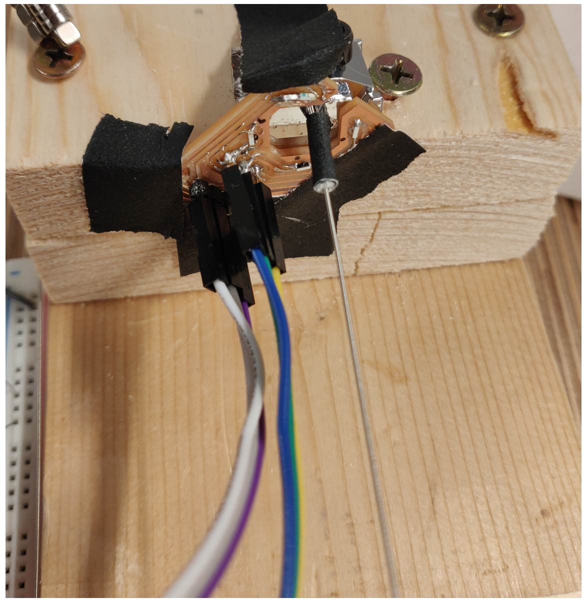


Figure 4-5: Sensor fixed in place around string. The sensor was rotated 45° so that sensor readings correspond to vertical/horizontal displacements.

To test how well the sensor captured 2-axis movement the string was displaced by hand in the vertical direction by a fixed amount and data was captured and displayed visually on a PC. The same method was then applied in the horizontal direction. We designed a device to keep the string moving a constant, fixed distance purely in the horizontal or vertical direction. This allowed us to roughly verify the sensor implementation and whether it was capable of measuring data from 2 axes independently. The device consists of a piece of wood fixed to a set of calipers, as shown in Figure 4-6. To use the device, we kept the string flush against the wood in between the caliper jaws to ensure movement along only one axis. The caliper jaws were used as a guide for how far to displace the string.

The results shown in Figures 4-7 and 4-8 were attained with the process described above. The results indicate that measuring 2-axis string displacement with the sensor is possible, but more accurate calibration is still required. This can be seen in Figure 4-7, where the string was being moved purely in the horizontal (X-axis) direction, but the sensor measuring vertical (Y-axis) string displacement was still picking up a

Figure 4-6: Caliper testing device allowing string to move 5mm in either direction

signal from some of the horizontal string movement.

4.2 First Sensor Assembly (v1)

In order to get the initial results from the previous section, the sensor had to be positioned exactly in the right place so that the optical sensors only picked up the axis of movement that they are meant to. Finding out exactly the right position by hand was a very tedious and inexact process, so designing a device that allows the user to quickly and easily adjust the vertical position of the sensor was the next important design step.

As the relative position of the string and the sensor is crucial to the sensor's functionality, we decided that the bridge which supports the string, the PCB with the sensor, and the sensor enclosure and alignment mechanism should all be part of the same assembly. This would allow us to ensure the proper alignment of the sensor and string. Having the sensor enclosure act as a bridge for the string also increases the modularity of the sensor. A person using this sensor for an instrument could choose exactly where to attach strings without needing to worry about constructing

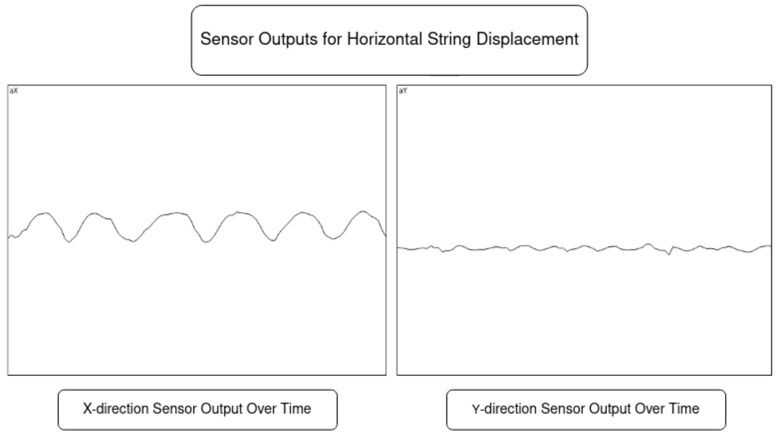


Figure 4-7: Sensor prototype measuring horizontal string displacement

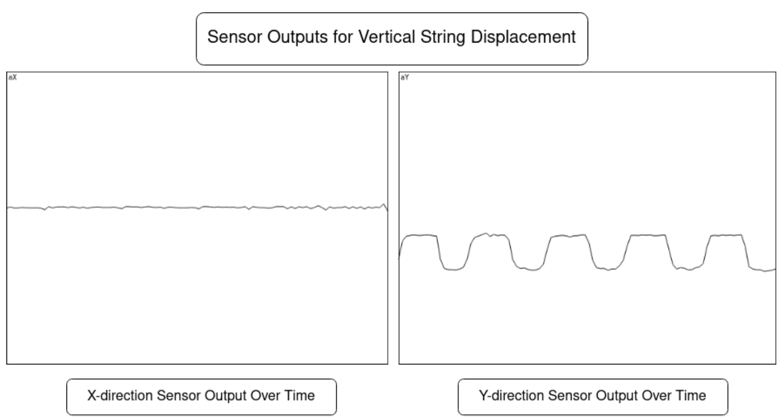


Figure 4-8: Sensor prototype measuring vertical string displacement

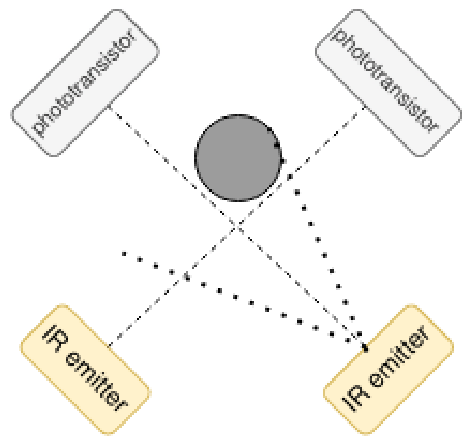


Figure 4-9: Ideal resting string position in the sensor. Light cone from the left IR emitter is not drawn for simplicity.

a bridge/pivot point for the strings, or needing to concern themselves with details regarding sensor and bridge orientation.

Our goal was to use the bridge to constrain horizontal movement of the PCB and to provide the ability to adjust the PCB easily by repositioning it purely in the vertical direction. The ideal position for the string is shown in figure 4-9. This particular positioning of the string ensures monotonicity of the sensors. For example, if the string were at the exact midpoint between the emitters and phototransistors in figure 4-9, moving the string down some amount might result in the phototransistor outputting some voltage v. However, moving the string upwards some amount might also result in the same output voltage v. This is an example of non-monotonic behavior; the same signal is being outputted even though the input to the sensor is different. We want our sensor to be able to differentiate between different directions of string movement, and therefore need to be able to position the sensor such that monotonicity is ensured. Regardless of the vertical position of the sensor relative to the string, from figure 4-9, we can see that the ideal horizontal position of the sensor is such that the string is exactly in the middle. The need for the string to be exactly in the middle is shown in the concept by having the bridge hold the sensor in place horizontally so that the sensor is always exactly in the correct horizontal position.

In order to position the sensor given the proposed sensor enclosure, there needs to be some way of reliably adjusting the vertical position of the sensor. We were

Figure 4-10: Guitar tremolo bridge saddle. Note how the screws with springs around them allow for changing the vertical position of the saddle.

inspired by the design of guitar tremolo bridge saddles as shown in figure 4-10 that use a screw with a spring around it to adjust their position.

The model for the initial sensor prototype including the enclosure is shown in figure 4-11. Here, the bridge is shown in purple, the sensor PCB is shown in green, and the mount for the PCB is shown in yellow. The PCB mount and bridge both have holes through the back to route the PCB headers and cables through. They both also have M3 screw sized holes in them to allow a screw to go through (shown in grey). The portion of the screw that is between the bridge and PCB mount has a spring around it. This allows the PCB mount, and therefore the PCB sensor, to move upwards when the screw is turned. By tightening or loosening the screw, the user would be able to easily adjust the vertical position of the sensor. The bridge encapsulates the PCB and PCB mount, providing space for both to slide into place as well as a contact point for the string to run through. The bridge also has a groove for the string to rest in.

In total, the entire sensor consists of the sensor assembly (PCB), the PCB mount, and bridge. The tuning pin or other method of affixing the string to the instrument is not part of of the sensor so as to allow the user to make the best choice for their instrument.

To assemble the completed sensor, the PCB mount first needs to be slid horizontally into the space in the bridge and into alignment with the screw hole on the

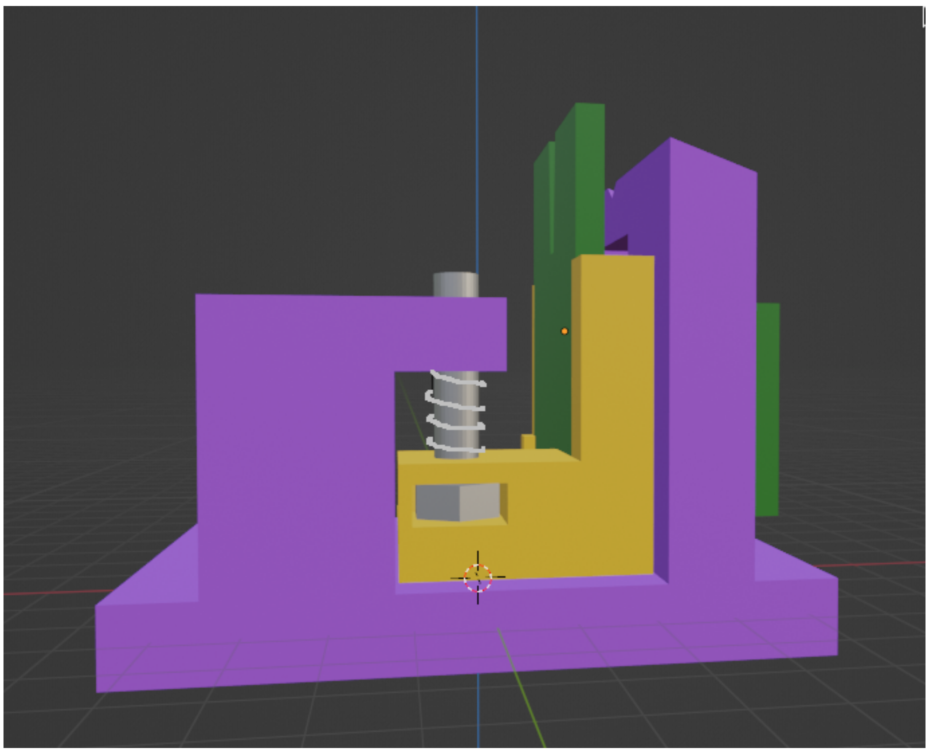


Figure 4-11: Initial sensor prototype showing bridge (purple), PCB mount (yellow), PCB (green), and screw/nut with spring (grey). The string would be affixed to the instrument off of the right side of the image and be strung up and over the right side of the bridge where it runs through the slot in the PCB.

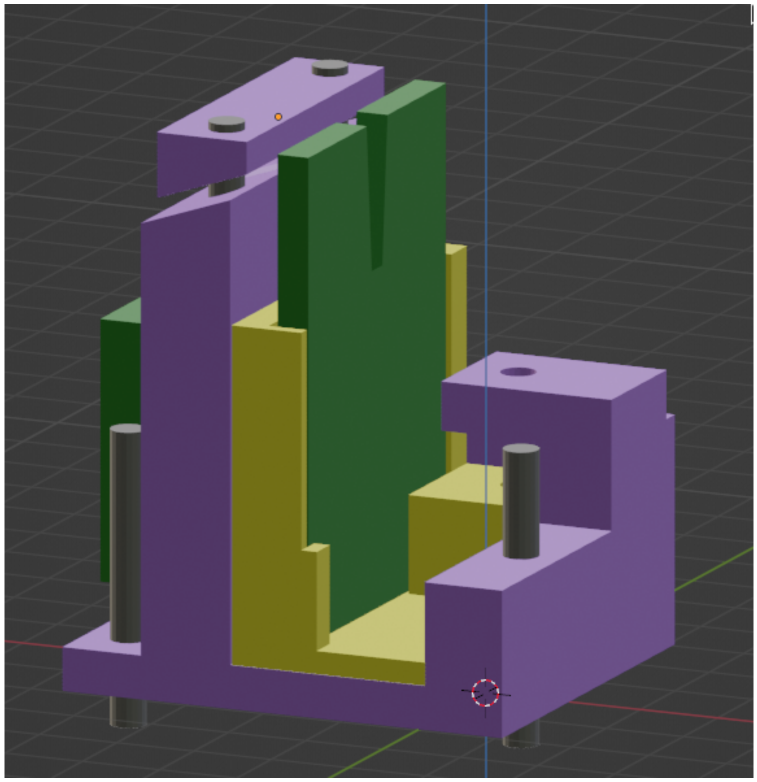


Figure 4-12: Second, more compact sensor enclosure prototype. This prototype includes a clamp on top of the bridge to secure the string in place as well as holes in the bottom of the bridge to fix the bridge in place.

bridge. The PCB can then be slid downwards to fit into the mount with the header pins fitting through the hole in the back of the bridge. Finally, the screw with the spring can then be screwed through the bridge and PCB mount.

4.2.1 Assembly analysis

The dimensions (L x W x H) of the v1 prototype without the PCB were 4.5cm x 3cm x 4.75cm. However, the width of the enclosure (3cm) is far too wide for our original goal of achieving 1cm string-to-string distance described in the list of objectives in 2.1.

We found that the process of assembling the v1 sensor demonstrated viability in the design, as the bridge helped constrain the horizontal movement of the PCB and enclosure, while still allowing the position to be adjusted in the vertical direction.

While the sensor enclosure was designed to help the process of adjusting the

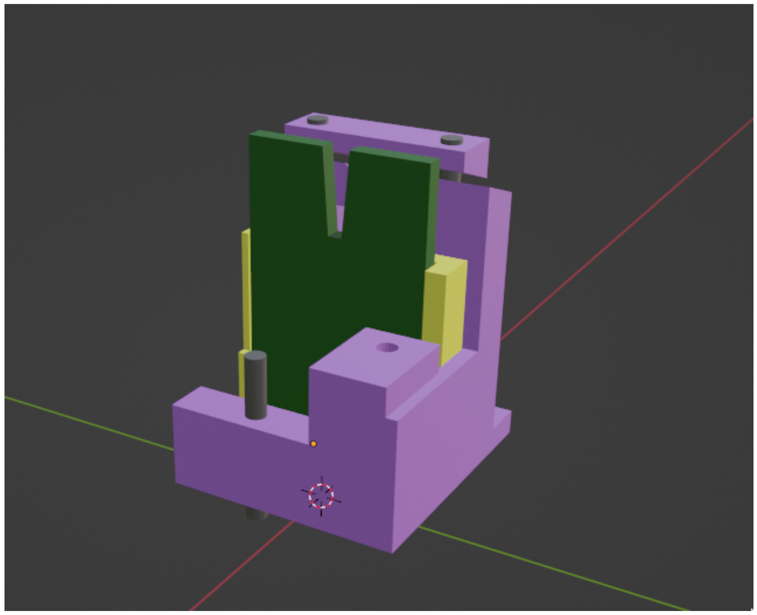


Figure 4-13: View of compact sensor enclosure with wall limiting movement of the PCB and PCB mount.

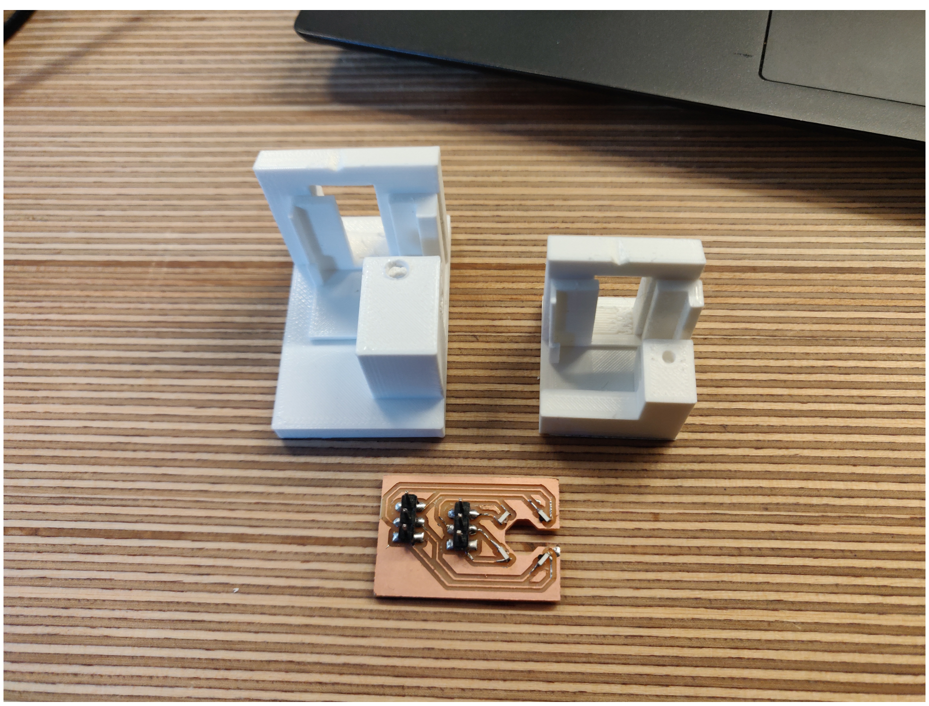


Figure 4-14: Size comparison of the initial and compact sensor enclosure prototypes respectively

vertical position of the PCB, we found that data such as those captured in figures 4-7 and 4-8 was difficult to replicate. The screw and spring were successful in adjusting the vertical position of the sensor, but the large size of the sensor and PCB mount proved too bulky to adjust gracefully. The sensor would often tip slightly to one side or the other as the screw was being rotated, breaking the alignment of the sensor and string. A smaller PCB and enclosure size as well as less tolerance for the enclosure to move horizontally would be required in future iterations.

4.3 Second sensor assembly (v2)

The design of the v1 prototype was then iterated on to optimize the size. The more compact v2 prototype (figure 4-12) included several additional elements:

1. a clamp for the string on the bridge to keep the string from slipping (shown above the bridge, also in purple in figures 4-12 and 4-13). This replaces the need for the Floyd-Rose locking nut used in previous prototypes.

2. mounting holes for the sensor. The v1 sensor relied on string tension to stay in place, which was not a robust method of mounting the sensor. The v2 sensor used 3 M2 screws (2 to the left of the bridge relative to figure 4-12, and 1 to the right, next to the adjustment screw) to keep it stably mounted to the body of the instrument, regardless of string displacement.

3. a wall (figure 4-13) to constrain horizontal movement of the PCB mount, ensuring that the horizontal orientation of the sensor and the bridge is consistent as the vertical position of the PCB mount is adjusted.

4.3.1 Assembly Analysis

Using screws to mount the v2 sensor in place increased sensor stability as desired by the v1 sensor analysis; however, M2 screws were too thin to provide the level of robustness we were looking for. The v2 sensor still tended to shift slightly in position when there were large amounts of string displacement, particularly with thicker strings. The string clamping mechanism was another factor increasing stability in the v2 sensor. Without any clamping mechanism, we noticed that displacing the string upwards off the bridge resulted in the string slipping out of the groove it is meant to rest in.

While the wall helped with sensor movement during adjustment, the sensor and its enclosure still tilted substantially just as with the v1 sensor and sensor alignment in this enclosure was not robust enough to produce meaningful data, requiring further revisions.

The v2 prototype had dimensions 2.85cm x 2.5cm x 3.1cm, which improved upon the initial prototype, but was still far off from the goal width of 1cm. A comparison of the sizes of the two enclosures can be seen in figure 4-14. At this point, the limiting factor in the width was the width of the PCB, which needed to be narrower in order to meet the desired 1cm string-to-string distance. Since we verified that the PCB design was feasible, we now made a two-sided version of the PCB which greatly reduced the size.

4.4 Final assembly design

Following our analysis of the initial two prototypes assemblies, we decided to design new sensor PCBs for professional manufacturing. This enabled us to design 2-layer PCBs, allowing for further miniaturization of the entire assembly. The final PCB design kept the schematic and components of the prototype PCB, but changed the headers and trace routes to optimize the size. In the prototype PCB design, standard 2.54mm (0.1") headers were used, which fixed the width of the PCB due to their size. In order to improve the size of the PCB, smaller headers were required. We decided on using 1.25mm (0.049") headers positioned vertically on the PCB to decrease the width as much as possible. Routing some of the traces on the back side of the PCB contributed to the size decrease as well. The modified traces can be seen in figure 4-15. Our final PCB (figures 4-16 and 4-17) measured 0.9cm in width and 2.1cm in

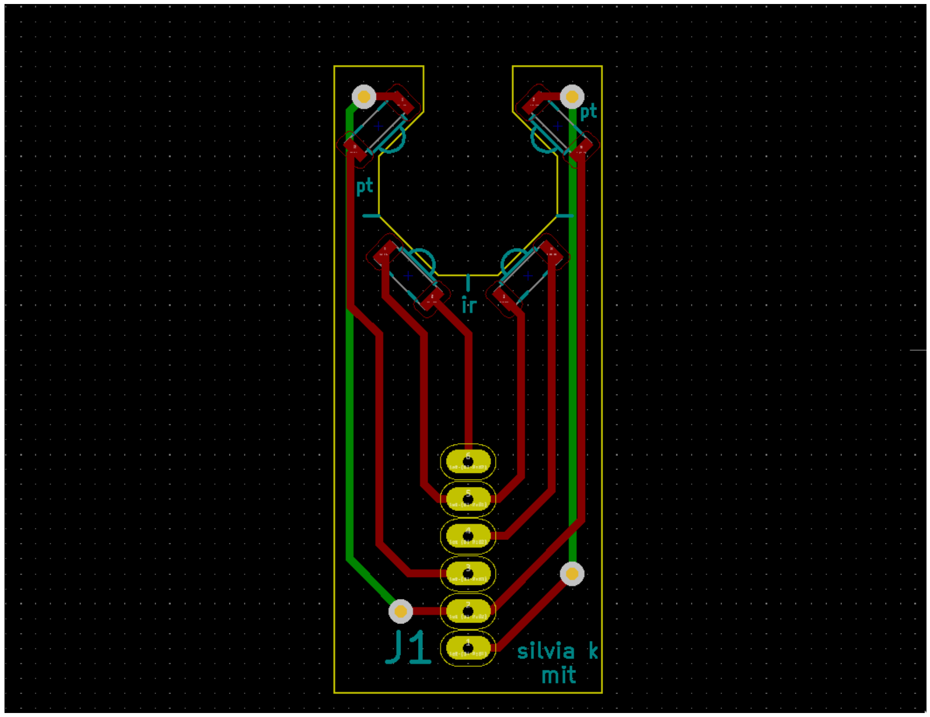


Figure 4-15: Final PCB layout. Front copper traces in red, back copper traces in green. Silkscreen marks were included around the hole for the string to go through to potentially assist with sensor calibration.

height.

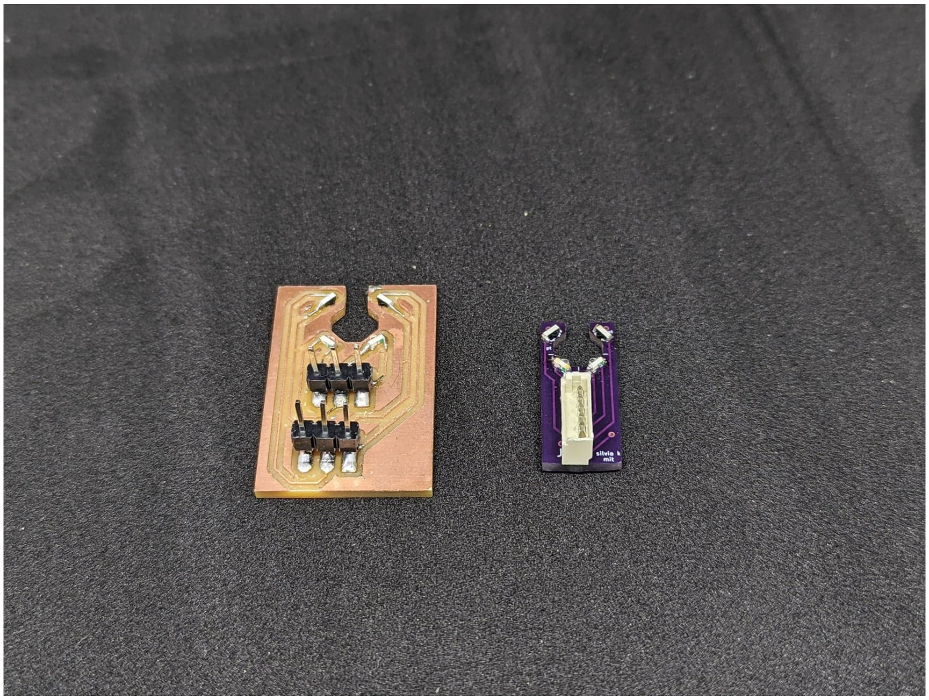


Figure 4-16: Side by side comparison of the PCB prototype and final PCB.

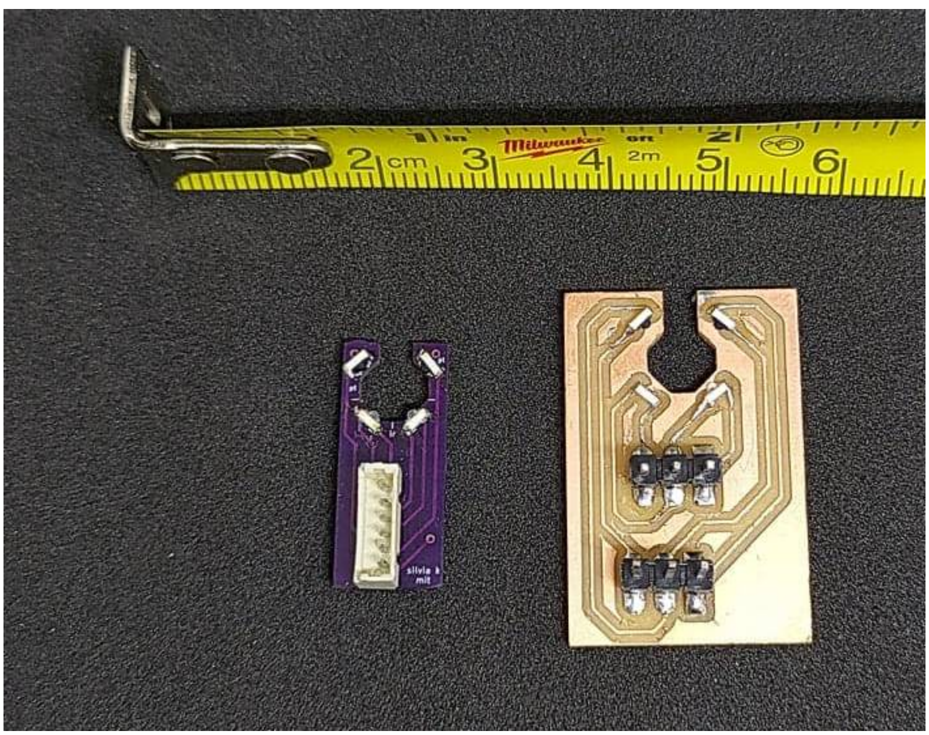


Figure 4-17: Size comparison of the final PCB compared to the prototype. The hole for the string to go through is the same size in both.

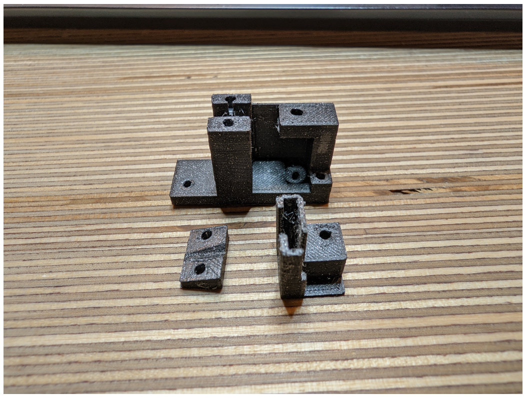
4.4.1 Final Sensor Enclosure

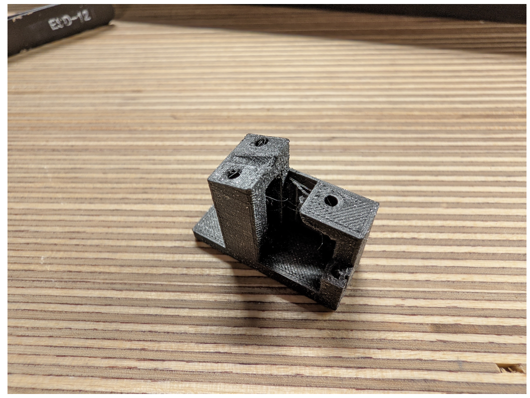
As the PCB dimensions had changed substantially, the sensor enclosure design was also altered. The width of the bridge and PCB mount was shrunk to fit around the new PCB. To limit movement of the PCB inside the PCB mount, the PCB mount now wraps around more of the PCB, holding it in place more securely. The dimensions (L x W x H) of the final sensor without the PCB inserted are 3.25cm x 1.1cm x 2.2cm. Making the PCB more compact made the entire sensor almost 3 times narrower, closely approaching our goal of 1cm string-to-string distance.

The length of the sensor enclosure was increased in the final version to allow room for a M3 mounting screw to be screwed through the back plate, providing extra stability to the sensor. The angle of the top of the bridge was also decreased in the final version so that the string wouldn't have as much of an upward angle coming off of the bridge. This would allow more range of string motion when displacing the string vertically up off of the bridge.

The sensor assembly process also changed because of the smaller sensor enclosure. Due to the constrained size, the PCB is no longer able to slip into the sensor mount. The top of the bridge blocks the PCB from entry. To mitigate this problem, the top of the bridge was sliced off before 3D printing the bridge (figure 4-18a), and screwed onto the bridge along with the string clamp after the PCB and string are in place (figures 4-19a, 4-20).

A common issue with the previous prototype sensor enclosures was that as the PCB height was adjusted, the PCB enclosure would tilt from side to side due to its own weight as well as the weight of the PCB. Due to the small size of the final sensor enclosure and the increased constraints, this was no longer a prevalent issue, and the PCB and PCB mount move smoothly up and down as the adjustment screw is tightened or loosened.

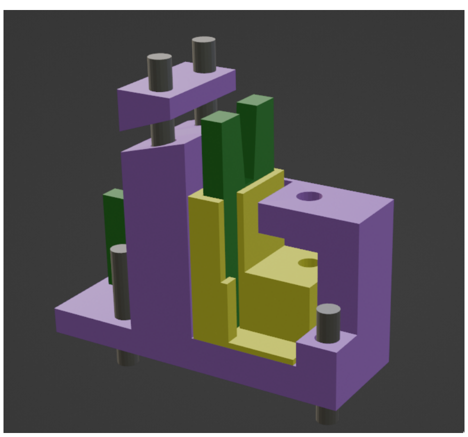




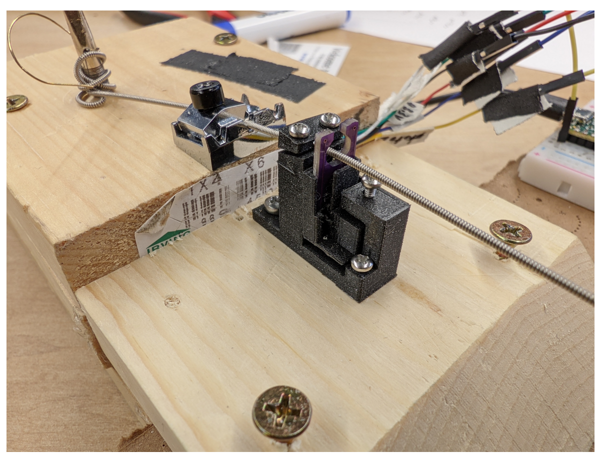
(a) 3D printed bridge with cut top and PCB enclosure (bottom right)

(b) 3D printed bridge with the cut top put into place

Figure 4-18: Disassembled 3D printed sensor enclosure parts

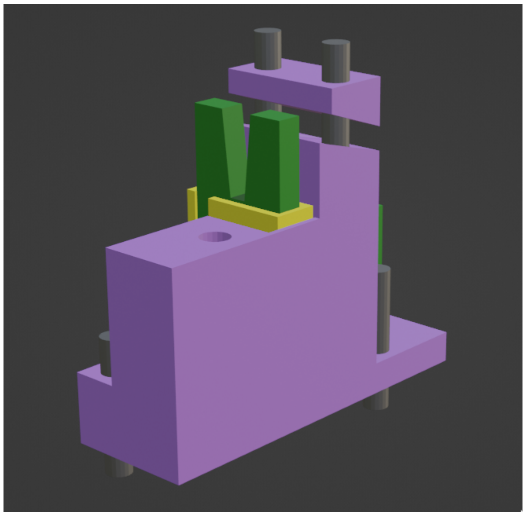


(a) Rendering of the sensor showing the open side

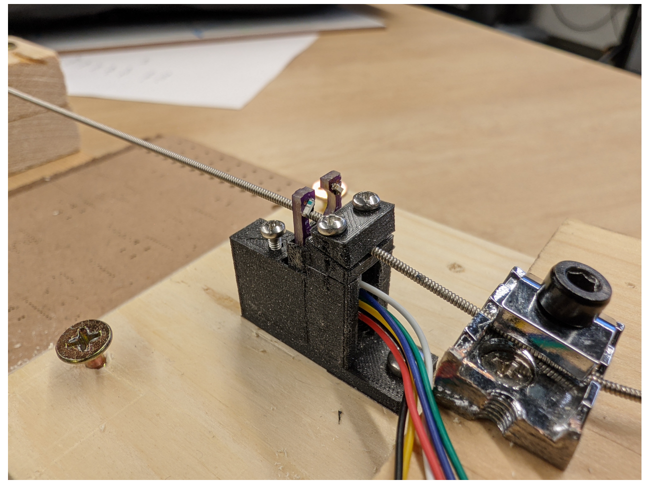


(b) Assembled sensor showing open side

Figure 4-19: Comparison of 3D sensor model compared to assembled sensor showing open side



(a) Rendering of the sensor showing the closed side



(b) Assembled sensor showing closed side

Figure 4-20: Comparison of 3D sensor model compared to assembled sensor showing closed side.

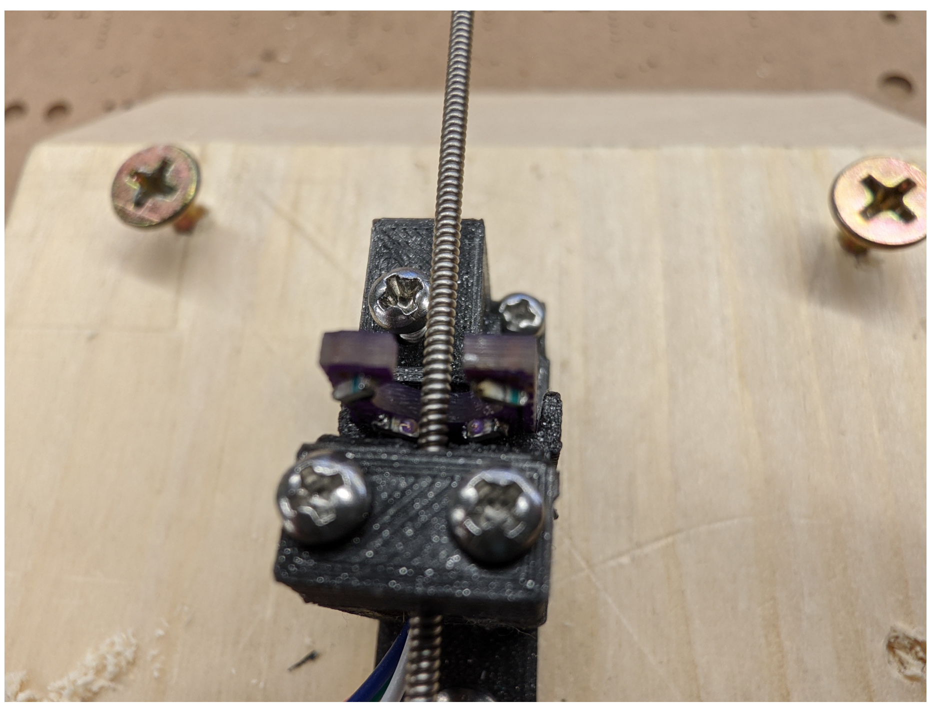


Figure 4-21: Top view of assembled sensor. The active infrared emitters glow purple in photos.

4.5 Conclusion

Our final sensor met many of the mechanical goals we set out to fulfill. Our sensor is compact, easy to manufacture, and has a reliable way of adjusting the height of the PCB mount. Although our sensor meets our mechanical design goals, there are still parts that can be improved. For example, the material that the bridge and PCB mount are currently made of is PLA, which holds up for our tests, but would not be a good long-term choice of material for the sensor due to its poor strength and durability.

Initial tests on the sensor confirmed that if in the correct position, the sensor is capable of detecting 2-axis string displacement. Further testing done in chapter 6 includes evaluating sensor resolution compared to displacement as well as how well the sensor works with different string thicknesses and materials.

Chapter 5

Signal Processing

After building the mechanical components of the sensor, we also need to acquire and process the sensor's output in order to get meaningful data. We separate this process into three parts- electrical, firmware-based, and software-based signal processing and data acquisition. A full block diagram of the entire sensor data pipeline is shown in figure 5-1.

5.1 Electrical

The sensor PCB has two infrared emitters and 2=two corresponding phototransistors[1] in an orthogonal configuration as described in detail in chapter 4. The signal from each phototransistor passes through a passive low-pass filter with a cutoff frequency of 10kHz to reduce signal noise before feeding into the microcontroller. Less signal noise increases the precision of the sensed string position. A 100nF decoupling capacitor between 3.3V and ground stabilizes the power supplied to the emitters and phototransistors. A complete schematic of the hardware involved in our sensor pipeline is shown in figure 5-2. Communication with a second microcontroller is discussed in the firmware section.

[1]We chose to use phototransistors (part no. 1541021NCA170) and infrared emitters (part no. 15410294AA570) from Würth Elektronik due to their extremely compact size and compatibility with microcontroller-level voltages and currents

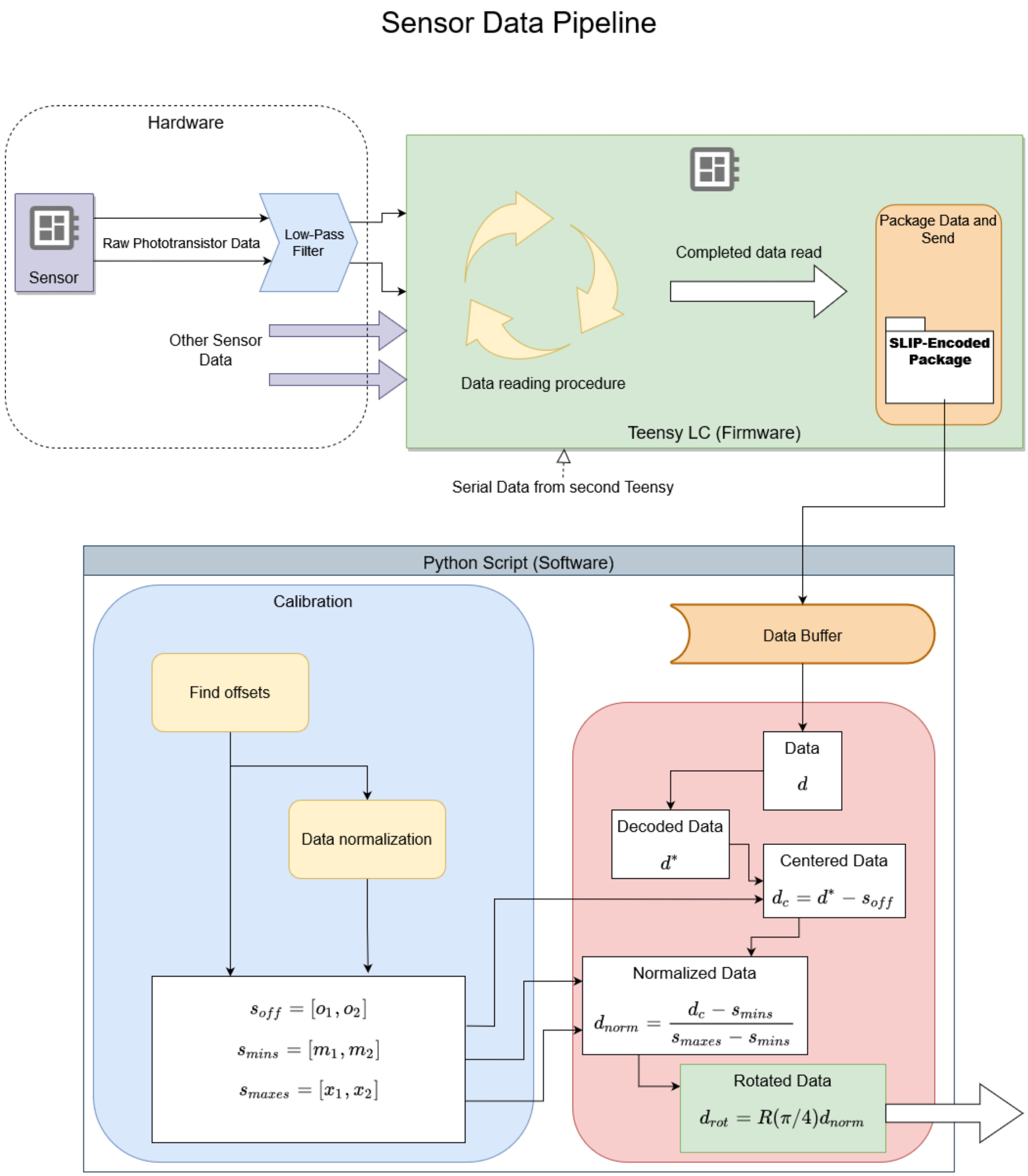


Figure 5-1: Block diagram of sensor data pipeline showing electrical, firmware, and software stages

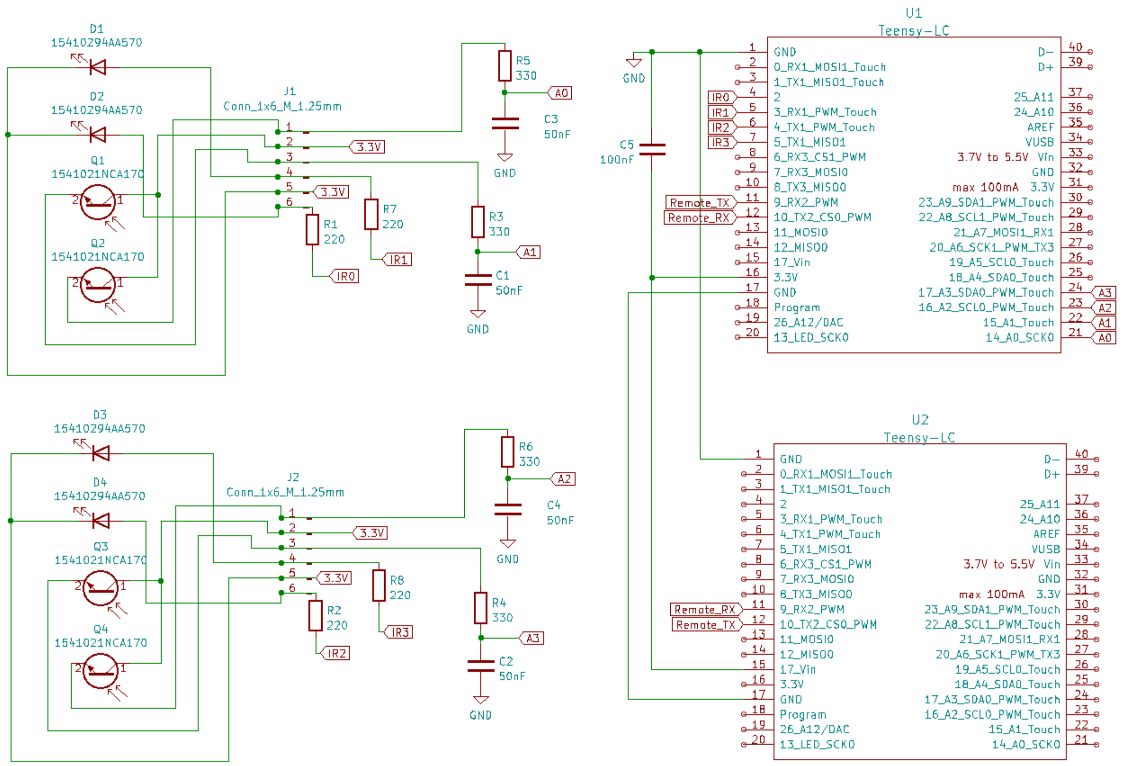


Figure 5-2: Schematic of the sensor hardware. Two sensors are attached to the top microcontroller in this schematic. For simplicity, no sensors are depicted on the bottom Teensy, but the serial connection between the two microcontrollers is shown.

5.2 Firmware

The initial PCB prototype that was tested in section 4.1 was connected to a development board based on the Espressif ESP32. For the final product, we use Teensy LC[2] microcontrollers because of their low cost and large amount of analog input pins. The large number of analog pins are useful since the data from each phototransistor on the sensor is analog; each sensor needs 2 analog input pins available on the microcontroller. The 13 analog inputs on the Teensy LC allow for up to six sensor assemblies without the need for external analog multiplexers or other electronics.

5.2.1 Optical noise reduction

One of the limitations of optical sensing is the possibility for ambient light to impact the sensor's readings. We implemented several strategies to minimize this:

1. Use of emitter/detector pairs sensitive to the same range of infrared wavelengths (most sensitive at 940nm).

2. Physical orientation of the phototransistor towards the assembly's mounting holes, with the expectation that the assembly will be mounted on a solid base which would block ambient lighting.

3. The implementation of a three-stage data reading procedure described in figure 5-3.

The data reading procedure starts with both infrared emitters turned off. Data measurements are read from both phototransistors to measure the amount of background noise, $n = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}$. One emitter is then turned on, data is read from the phototransistor across from it, d_1, and then the emitter is turned off. The other emitter then gets turned on, data is read from the other phototransistor, d_2, and the other emitter is also turned off. Once this data has been acquired, we subtract the noisy values from the data to get a more accurate measurement of what the data actually

[2]Teensy Low-Cost https://www.pjrc.com/store/teensylc.html

is, $d^* = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} - n$. The noise-removed data, d^*, is encoded and sent to the PC via serial.

5.2.2 Data encoding and rate of data acquisition

The data from the sensors is sent from the microcontroller to a PC over a USB serial connection. As the data d^* consists of readings from a pair of photosensors which are sampled by the Teensy ADC at 12-bits (and each photosensor reading is stored on the Teensy as 16-bit integers) the size of d^* is 4 bytes. Packets for each sensor are created consisting of a single byte numberic sensor ID s_{id} and the sensor data d^*, for a final packet size of five bytes.

We use a modified version of SLIP encoding[1] to transmit these packets to the PC. Each sequence of bytes is sent out as a packet with the sensor id portion preceding the data; 255 is used as the END byte. 254 is used as the ESC byte. Considering that we use only the last four bits of the high byte for each sensor value, we can avoid values of 254 and 255 for s_{id}. The minimum packet size (no packet bytes escaped) is six bytes and the maximum packet size (both low bytes for the sensor data escaped) is 8 bytes.

The current firmware uses a serial baud rate of 115200, e.g. theoretically 11,520 bytes per second[3]. Given worst-case packet sizes of 8 bytes this equates to 1,440 sensor packets per second, which would need to accommodate all of the sensors on an instrument. This data rate could easily be increased by utilizing a higher baud rate; other optimizations could include encoding both 12-bit sensor data values into 3 total bytes, or switching to an alternative serial encoding such as Consistent Overhead Byte Stuffing [9].

Sensor data may be acquired from more than one microcontroller- this is especially the case in which a long string has a sensor at either end (such as the instrument discussed in Chapter 7). Both microcontrollers acquire data via the same procedure, but one microcontroller (primary) handles packets received from the other (remote)

[3] Based on a serial configuration of 1 start bit, 8 data bits, and 1 stop bit.

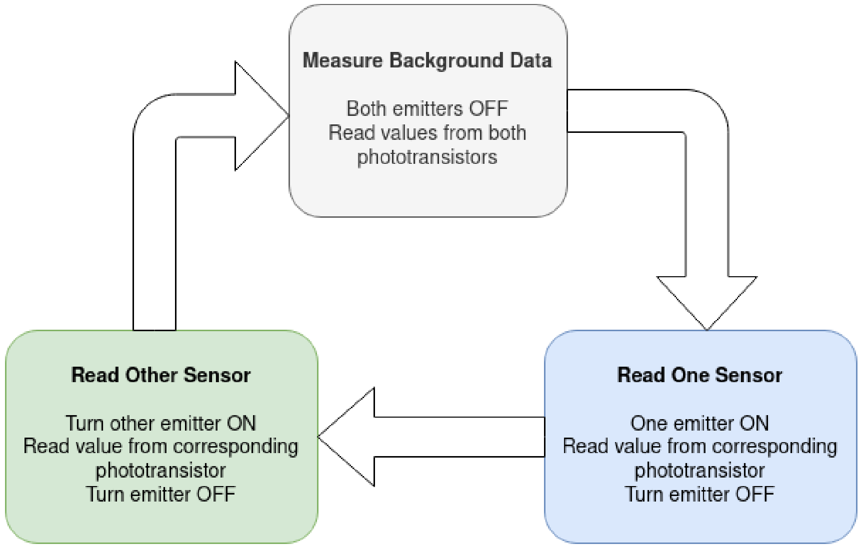


Figure 5-3: Overview of data reading procedure on microcontroller

microcontroller through an auxiliary serial port. Both microcontrollers acquire data via the same three-stage data reading process and send out data using the same encoding. The only difference between the two microcontrollers is that the primary microcontroller not only reads its own sensors' data, but also requests data from the remote microcontroller. The primary microcontroller sends out both its acquired data as well as the data received from the remote microcontroller. The remote microcontroller only reads data from its sensors and sends out the data through serial as shown in figure 5-1.

5.3 Software

Once the sensor data packet has been sent out from the Teensy, it is read in via serial and SLIP decoded by a Python script to produce the raw sensor values $s_{raw} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$. The script also handles calibration, further data processing, data visualization, and generating a control signal that is useful for sound synthesis.

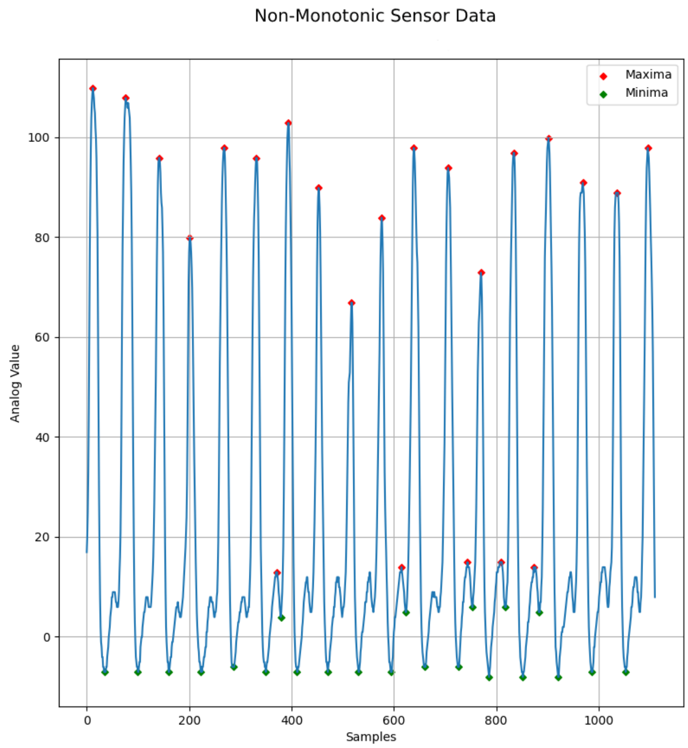
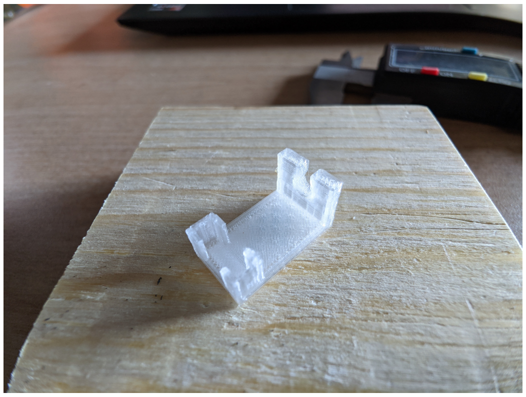


Figure 5-4: Non-monotonic behavior from one phototransistor while the string is being moved in a circle. The troughs at the bottom should not have the small peak afterwards; this indicates that the sensor is not properly aligned with the string.

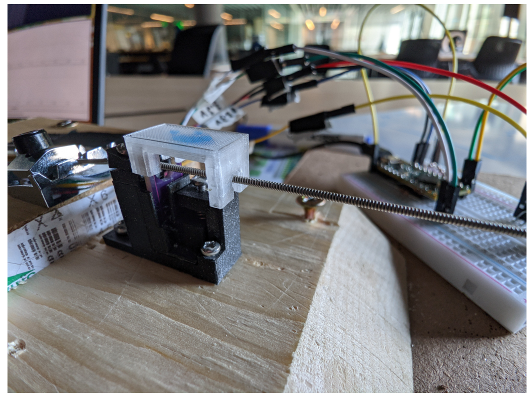
5.3.1 Calibration

Prior to and during calibration, the raw sensor data is plotted on 2 plots as shown in figure 5-6. The raw data visualization allows the user to adjust the sensor height using the screw-spring mechanism described in section 4.2. The sensor height should be adjusted until moving the string in a circle produces sinusoidal curves for each sensor, indicating monotonic behavior. An example of non-monotonic behavior can be seen in figure 5-4. Once the sensor height is adjusted appropriately, the calibration process can begin.

The calibration process consists of two parts: offset calculation and data normalization. During offset calculation, the string and sensor are left still. During the data normalization part of the calibration process, the user uses a calibration device as shown in figure 5-5 to move the string in a circular motion around the edges of a



(a) 3D printed calibration device



(b) Calibration device mounted on sensor

Figure 5-5: Calibration device to move the string in a circle of known diameter of 3mm.

circle of known diameter. The calibration device slips over the top of the PCB and around the string, providing a circular guide at a fixed position relative to the sensor PCB.

While the sensor is designed to facilitate the correct positioning of the string relative to the sensor PCB (as described in figure 4-9), the true position of the string will not be perfectly centered nor perfectly aligned; however, with correct calibration, it should still be possible to collect useful data even with an imperfect sensor alignment. We can see an example of imperfect sensor alignment in figure 5-6, where 2 sinusoidal curves are plotted as a result of the user moving the string in a circle. The curve from phototransistor 2 has a smaller amplitude than the curve from phototransistor 1 (range of 600mV for phototransistor 2 vs 800mV for phototransistor 1) even though during the calibration procedure, the string should be traveling the same distance towards and away from each phototransistor. The data from the phototransistors not only has different ranges, but is also centered around a different point- phototransistor 1 is centered around 1.2V, but phototransistor 2 is centered around 0.9V. Our aim in calibrating the sensor is to obtain a transformation that allows us to move from raw sensor data to data that gives us information about string deviation from the resting position.

In the offset portion of the calibration, the string is left still and we collect the average sensor values for the resting string position. The resting string position

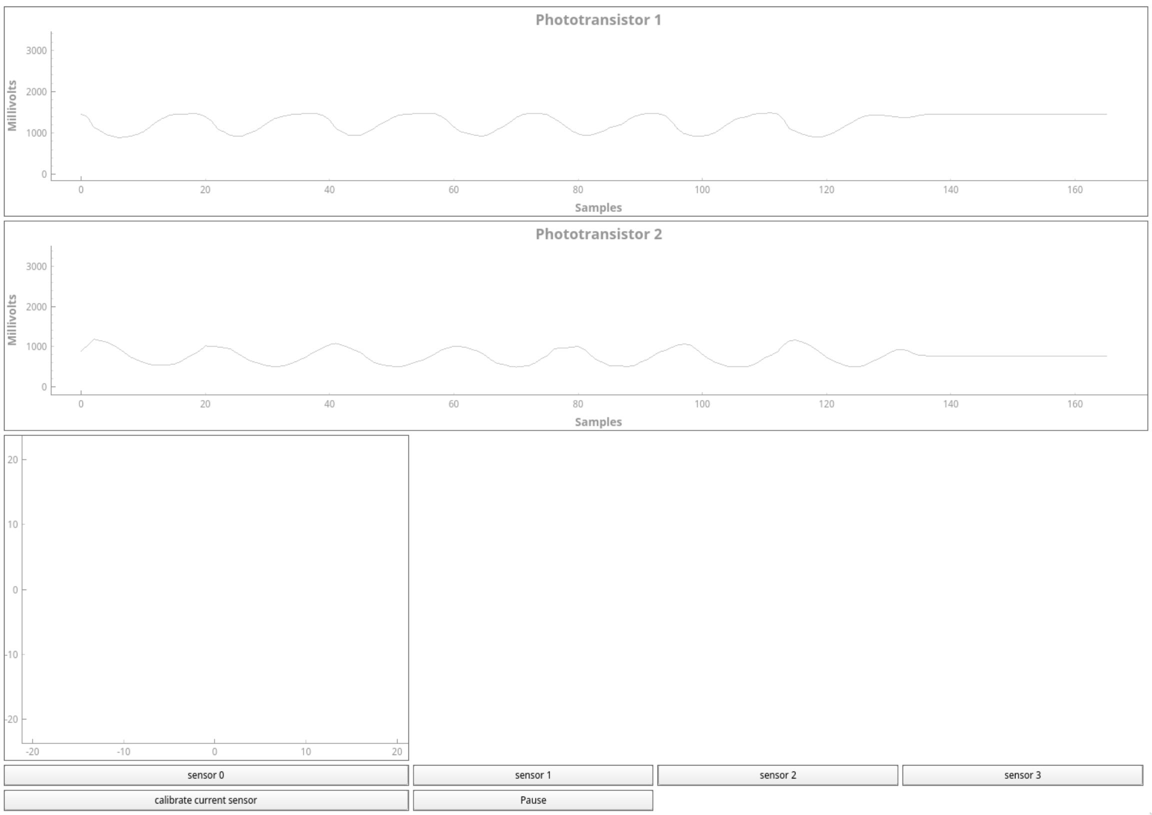


Figure 5-6: Raw phototransistor data plotted while calibrating a sensor that is not perfectly aligned with the string.

acts as the known origin for plotting and outputting calibrated string data. Using the sensor offset values $s_{off} = \begin{bmatrix} o_1 \\ o_2 \end{bmatrix}$, we can now center the sensor data around 0: $s_{centered} = s_{raw} - s_{off}$

After finding the center point for the sensor data, we can normalize the data. As mentioned above, during this phase of calibration, the user is moving the string in circles around the calibration device (figure 5-5), which has a circular cutout of diameter 3mm positioned at 15.97mm away from the phototransistors on the PCB. The phototransistors themselves are 1.42mm away from the bridge, at which point the string is fixed. While moving the string in circles, the user displaces the string by 1.5mm at a distance of 17.39mm away from the contact point of the string. This results in a displacement of 0.1334mm in the plane of the phototransistors.

The circular motion of the string around the calibration device should produce sinusoidal graphs for each phototransistor that are $\pi/2$ offset in phase, since the phototransistors are orthogonal to each other. From the sinusoidal graphs, we find the median maximum (s_{maxes}) and minimum (s_{mins}) values for each phototransistor to use min-max normalization. The new normalized data is now $s_{norm} = \frac{s_{centered} - s_{mins}}{s_{maxes} - s_{mins}}$

5.3.2 Data Visualization

We used the pyqtgraph[4] framework in Python to create a GUI to visualize our sensor data in real time (figure 5-6). Initially, in the top two rows, the GUI shows raw sensor data from each phototransistor. After performing the calibration procedure, the Y-axis scale for each phototransistor plot is centered around 0 with the same scale for each phototransistor. Also after calibration, the bottom left plot shows estimated X vs Y displacement (in mm) of the string. At the bottom of the GUI are options that let the user toggle the current sensor (the sensor from which data is being displayed), as well as the option to calibrate the current sensor and to pause the incoming stream of data. The pause functionality is particularly useful for examining our sensor data in more detail.

[4] https://www.pyqtgraph.org/

To get data from the serial input onto the graph, we keep a buffer of data values and query it for 5 samples of data. We take the average of these data values (d_{avg}) to further reduce noise. Given that we now know s_{off}, s_{maxes}, s_{mins}, and the angle between the sensor coordinate system and our desired coordinate system ($\pi/4$), we can calculate x and y values that correspond to our data.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) \\ \sin(\pi/4) & \cos(\pi/4) \end{bmatrix} \frac{(d_{avg} - s_{off}) - s_{mins}}{s_{maxes} - s_{mins}}$$

From the cartesian coordinates, we can also calculate the magnitude r and angle θ at which the string is being displaced:

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \arctan(y/x)$$

5.4 Conclusion

With the entire sensor processing pipeline in place, we are able to get meaningful data from the sensor. From the outputted sensor position data over time, there are many possibilities for creating mappings to generate sounds- for example, the velocity and acceleration of the string can be easily calculated and used for sound. Our sensing system is capable of handling and generating meaningful data even for strings that are not perfectly aligned with the sensor. While the sensor processing pipeline is able to tell us that our sensor succeeds at outputting 2-axis position data for the string, further testing is needed to quantitatively determine how well our sensor performs.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Sensor Data and Results

With both the physical sensor and signal processing pipeline complete, we collect and analyze sensor data. In this section, we discuss sensor data and evaluate the sensor. We found that the calibration procedure was a good way of quantitatively analyzing the processed sensor data because the normalization part of the procedure moves the string by a known amount. The results of the calibration procedure using different string sizes and materials can tell us about how robust the sensor is. We also wanted to verify our hypothesized range of monotonic behavior between string position and output voltage.

6.1 Calibration Measurements

Below are the results of a series of calibration tests using different string thicknesses and materials shown in table 6.1. The minimum resolution from each calibration test can be measured by finding the phototransistor graph with the smallest voltage range and subtracting the average minimum value from the average maximum value. The

String	Diameter	Material
Bass B	1.37mm	Nickel Wound
Guitar E	0.72mm	Nylon
Guitar E	0.2mm	Steel Wire

Table 6.1: Strings used in our calibration tests

minimum resolution can tell us about how precisely the sensor would respond to small changes in string position. The thick bass string had the highest minimum resolution (figure 6-1) of the 3 strings, with a value of 600.83mV. Using the same biasing resistor for the phototransistor as with the bass string, we can see that the resolution of the nylon string (figure 6-3) and thin string (figure 6-5) suffered, reaching minimum resolutions of 356.91mV and 110.78mV respectively. Lowering the biasing resistor on the phototransistor should increase the minimum resolution of the nylon and thin wire strings, but relative to the thick bass string the sensor is much less sensitive to string displacement. This raises a point of discussion for any future iterations of the sensor- should all sensors have the same values for the biasing resistor, or should we include something like a digital potentiomenter to adjust the the value of the biasing resistor? We discuss this in Chapter 8.

Due to the transparent nature of the nylon string, it was more difficult for the sensor to detect displacement, and the displacement detected was very minimal since the transparent string lets more light through to the phototransistor than a metal string.

Adjusting the sensor to the appropriate height for the thin wire string was much more difficult than adjusting the sensor height for the nylon string or the bass string. This makes sense given that the range for ideal placement (see 4-9) of the thin wire string is very narrow. With a thicker string, it is much easier to get in the vicinity of ideal string placement and still have the sensor output a signal close to the desired results. The data from the thin wire string is also much noisier than the data from the nylon string, as we can see by comparing the X-Y displacements of the nylon string (figure 6-4) and the thin wire string (figure 6-6). This is because the thin string covers up such a minimal portion of the light reaching the phototransistor that sensor noise starts playing a larger role in the output signal. Deviations from a perfectly circular calibration path are also more easily noticeable.

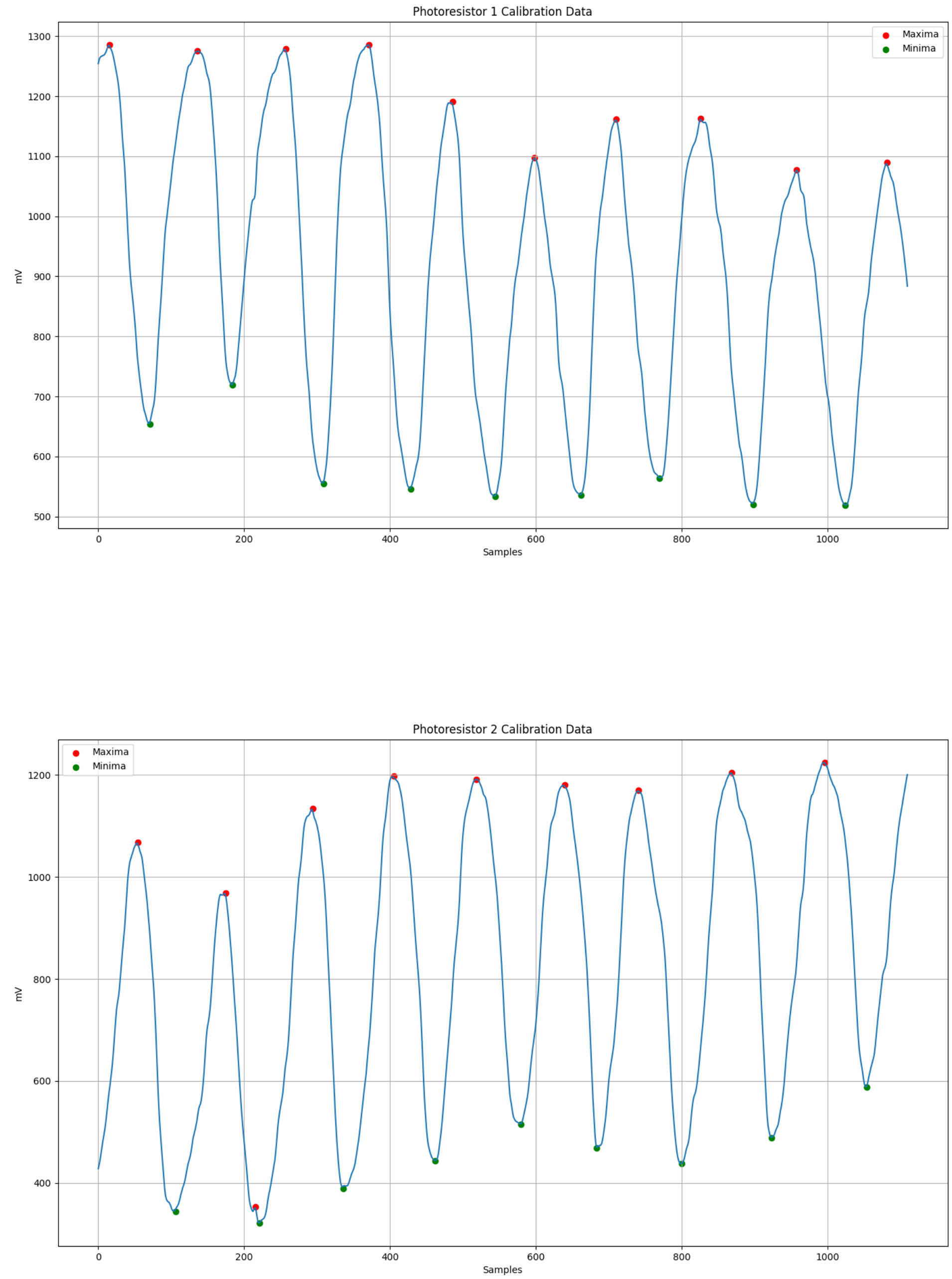


Figure 6-1: Raw calibration data for nickel wound string

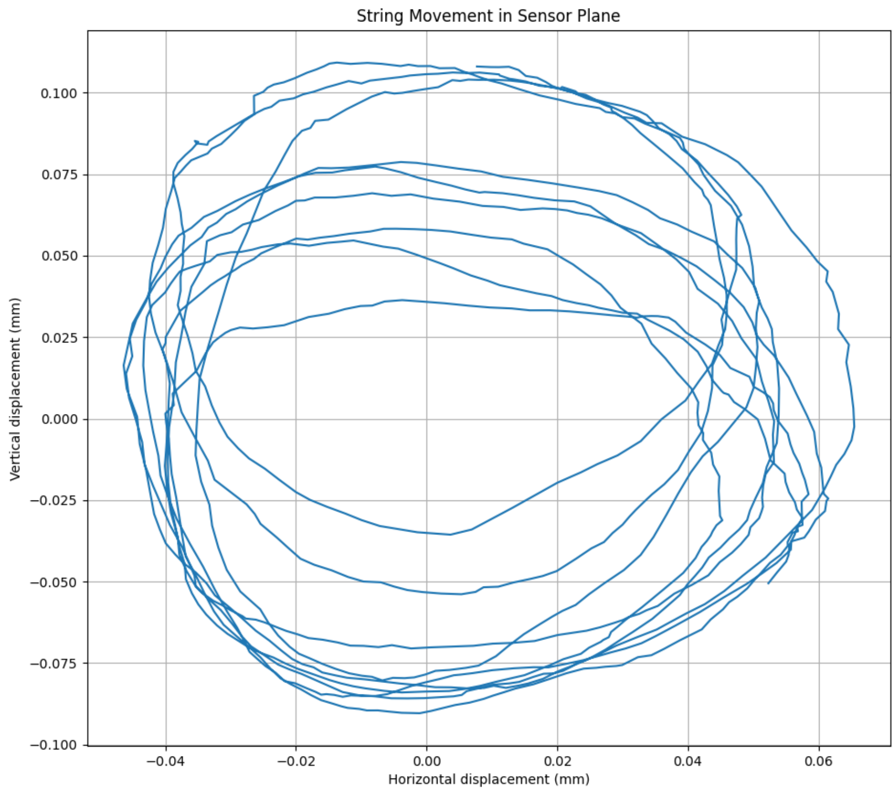
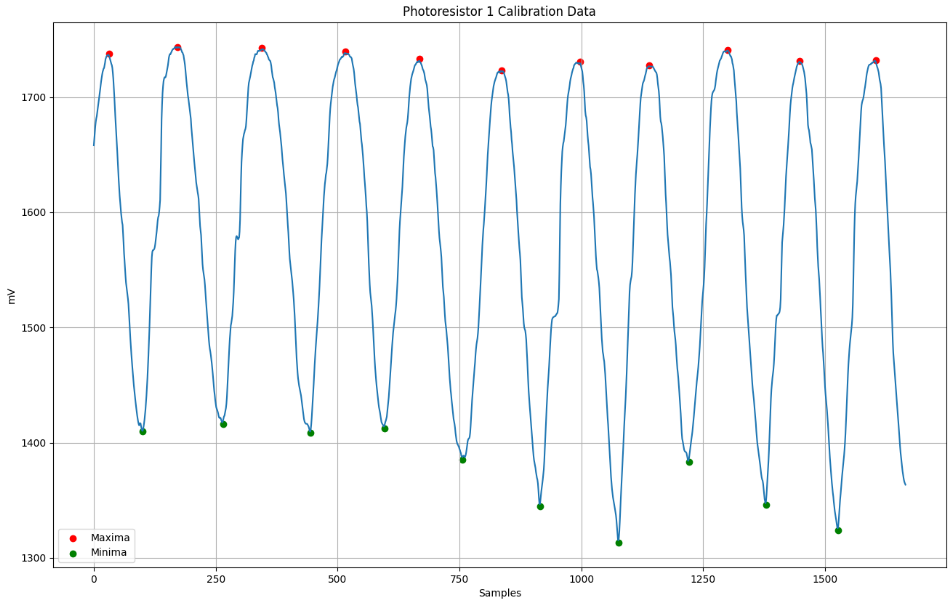


Figure 6-2: Centered, normalized, and rotated data from figure 6-1



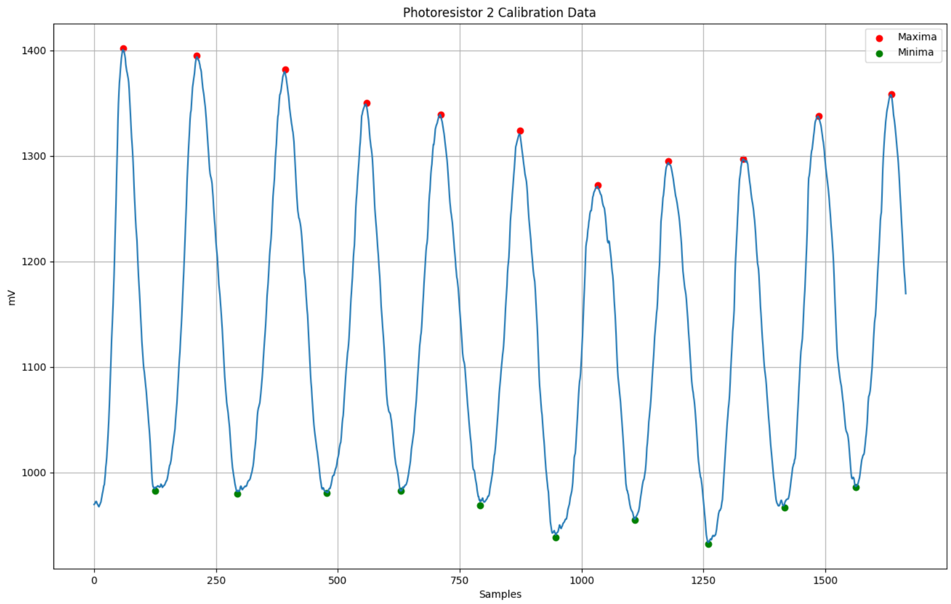


Figure 6-3: Raw calibration data for nylon string

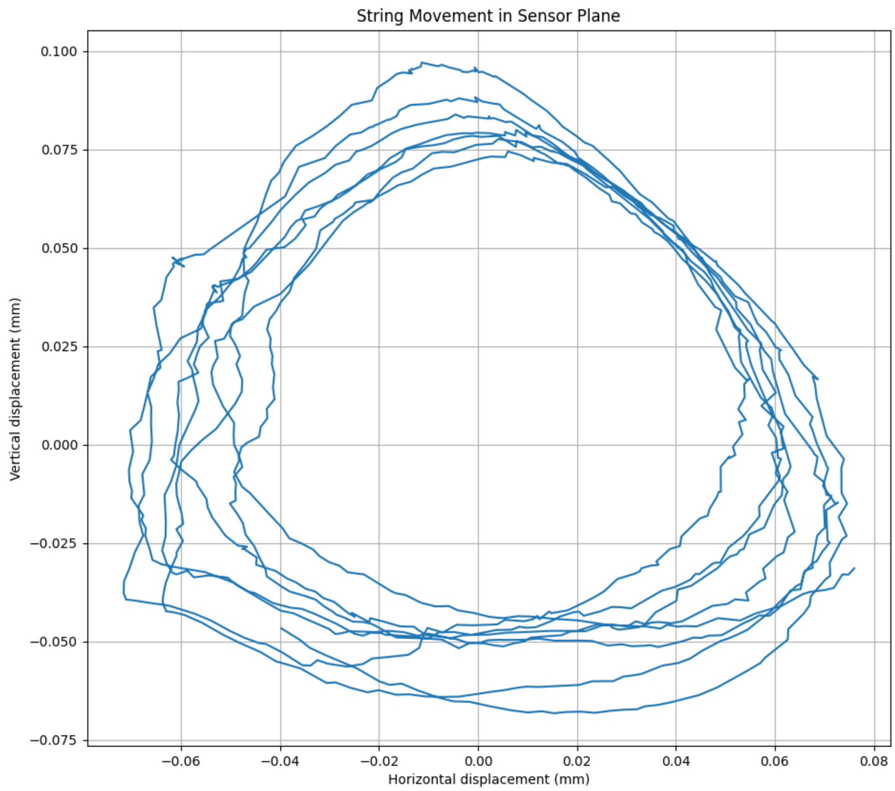
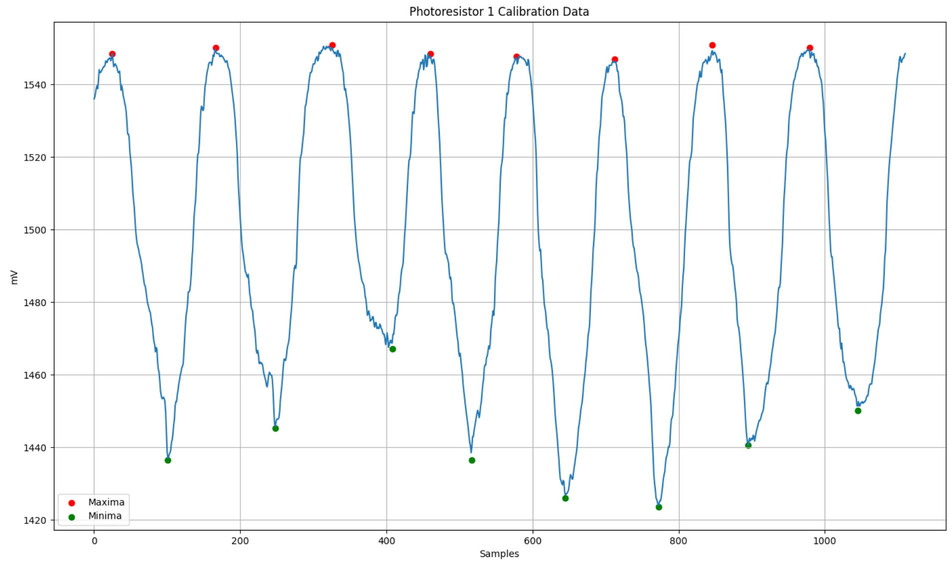


Figure 6-4: Centered, normalized, and rotated data from figure 6-3



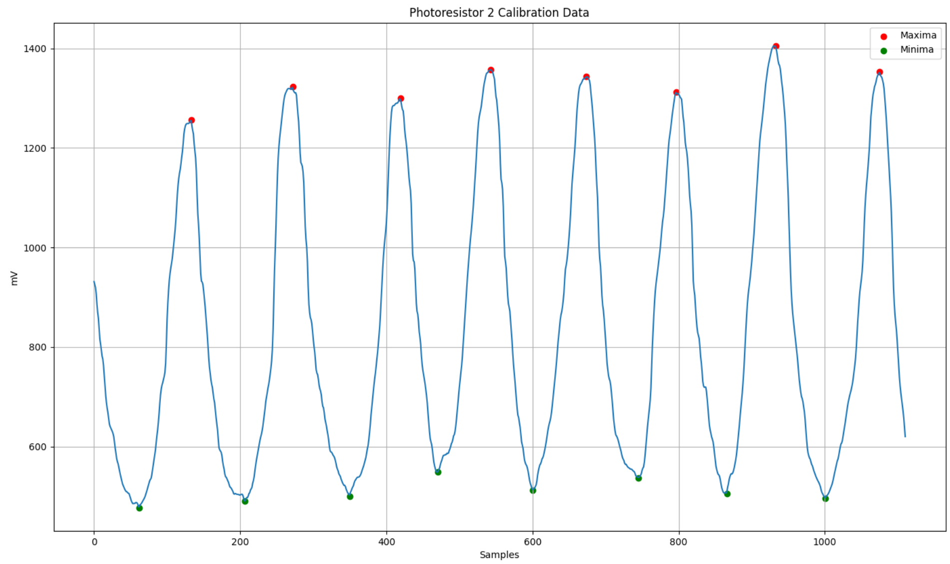


Figure 6-5: Centered calibration data for the steel wire string

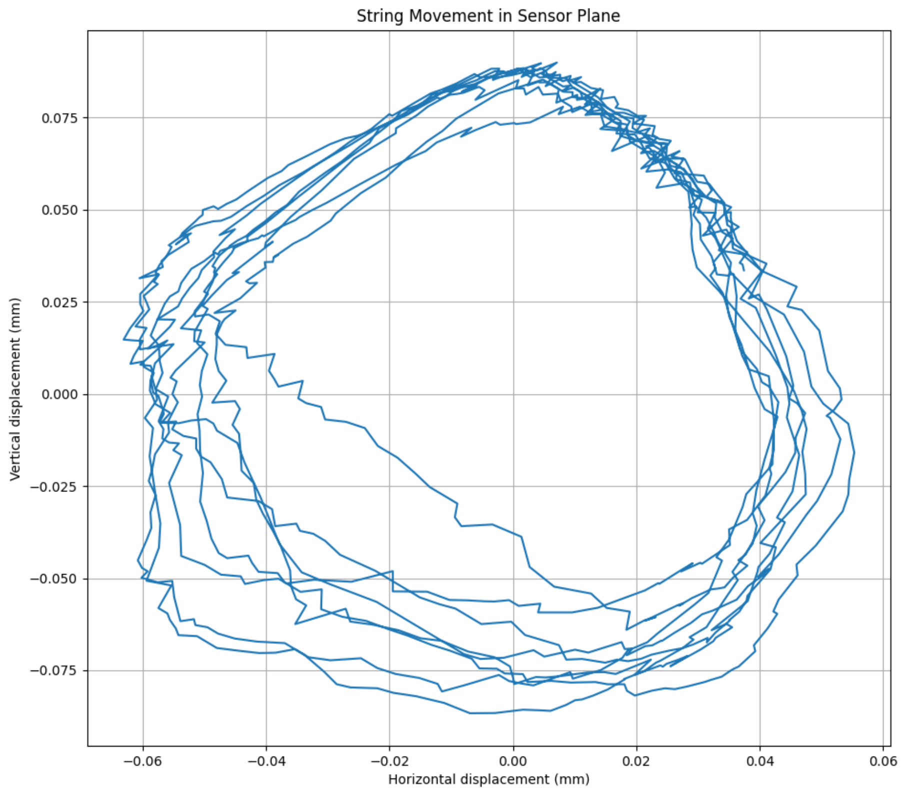


Figure 6-6: Centered, normalized, and rotated data from figure 6-5

6.2 Verifying Expected Monotonic Range

Given our definition for ideal string placement (figure 4-9), we can see that there is a point at which the relationship between output volage from a phototransistor and string position becomes non-monotonic. As soon as the string crosses past the center of a phototransistor, the output voltages from the phototransistor starts mirroring previous values. Our small sensor size, proximity of the sensing elements to the bridge, and PCB height adjustment process all exist to prevent the string from crossing into a range of non-monotonic behavior; however, we wanted to confirm whether the string crossing the center point of the sensor really was the point at which non-monotonic behavior started to manifest and whether our calculation for how much displacement necessary was correct in order to confirm our predicted sensor behavior.

What allows us to start forming a relationship between sensor output voltage and string displacement distance is the calibration procedure in which we move the string in circles around the calibration device. We used a string with 0.77mm diameter, which means that a displacement past 0.385mm in the sensor plane would cause non-monotonic behavior. The calibration device is 17.39mm away from the contact point of the string and the sensing elements on the PCB are 1.42mm away from the contact point of the string. This means that a displacement of 4.72mm displacement at a distance of 17.39mm away from the contact point of the string would cause a 0.385mm displacement in the sensor plane.

To verify, we plotted the data using the 0.77mm string. The tension in the string had to be lowered a significant amount in order to be able to come close to achieving a displacement of 4.72mm at 17.39mm from the contact point of the string. This means that it would not be very likely to enter a non-monotonic region of sensing (with this particular string). We can see in the plot in figure 6-7 that a peak occurs approximately when the sensor reaches 4.72mm of displacement. After that, even though the signal seems to indicate the string started moving in the opposite direction, in reality, the string is continuing to be pulled in the same direction to a displacement of about 5.5mm before finally being released close to sample 900.

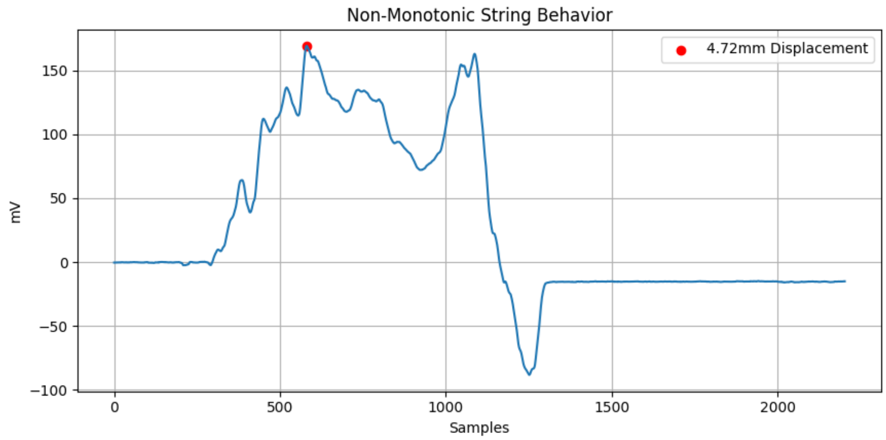


Figure 6-7: Displacement of string until non-monotonic behavior is reached. The string was moved through a range of approximately 5.5mm from its resting position. Non-linearities in string motion are a result of human error in moving the string. Due to the string being displaced so drastically, the resting position of the string was affected.

6.3 Conclusion

We have now verified that the sensor works on a variety of different string thicknesses and materials. While the sensor performs much better on thicker, opaque strings, it is still able to produce meaningful data for both transparent and thin strings. We have gathered significant quantitative data, but we still must test the sensor in the context of a DMI in order to get a sense of how well it works musically.

Chapter 7

Instrument Design

We have developed a sensor to measure string displacement for use in digital musical instruments. In order to find the artistic limitations of the sensor, such as finding out what musical techniques are not feasible given the sensor design, we designed and constructed an instrument that makes use of the sensors. Photographs documenting the instrument assembly process are in Appendix A.

7.1 Instrument Assembly

Our instrument consists of two strings mounted on a 1 x 4 piece of pine wood. The instrument has a 24" string length for both strings.

To start, the placement of the sensors, tuning pins, and circuitry was laid out. To mount the strings, tuning pins were used on one end, and ferrules on the other. Holes for the ferrules were drilled with a 7/64" drill bit, and countersunk on the back of the instrument using a 3/16" drill bit so that the instrument could lie flat on a table. Holes for the tuning pins were also drilled using a 3/16" drill bit (Associated pictures in figure A-1).

In order to assemble the sensors, all parts in Table 7.1 are needed. The PCB mount, bridge, bridge top, and string clamp were 3D printed, so the support material needs to be cleared out of those parts. All faces, especially those on the bridge and PCB mount, need to be very flat to improve sensor alignment. Since the parts are all

Part Name	Quantity
M3x8 Screw	1
M2x8 Screw	3
M2 Nut	1
M2x12 Screw	1
0.5x6x5mm Spring	1
PCB Mount	1
PCB	1
PCB cable	1
Bridge	1
Bridge Top	1
String Clamp	1

Table 7.1: List of components needed to assemble a sensor

very small, tweezers are used to clear out the support structures (Associated pictures in figure A-2).

The M2 nut must be inserted into the slot in the PCB mount. Correct alignment of the nut with the hole in the PCB mount is necessary for being able to smoothly screw in the M2x12 screw. Tweezers are used to pull the M2 nut into position and we ensure it was correctly in place by looking through the hole in the PCB mount and checking for obstructions. The PCB mount is then placed into the bridge (Associated pictures in figure A-3).

The 0.5x6x5mm spring is needed to assemble the spring mechanism. We utilize a 0.5x6x10mm spring cut in half. The spring is inserted between the bridge and PCB mount and aligned with the holes on each as best as possible. The M2x12 screw is then put in through the bridge and spring, and screwed into the PCB mount. After screwing in the M2x12 screw, the height of the PCB mount is able to be adjusted (Associated pictures in figure A-4).

Tabs left over from manufacuring need to be sanded off of the PCB and the header cable attached. The phototransistors, IR emitters, and header were already soldered onto the PCB (Associated pictures in figure A-5).

Pilot holes are drilled for the M3x8 and M2x8 screws that hold the sensor in place. The sensor enclosure is then screwed onto the instrument. The PCB is slid into place, and the instrument string fixed in place using the top of the bridge and string clamp.

2 M2x8 screws screw the string clamp and bridge top onto the bridge (Associated pictures in figure A-6).

The headers from the sensors are connected to a perf board housing the analog circuitry and Teensy so sensor data can be transferred to a PC. The GUI is used to visualize sensor data while the height of each PCB mount is adjusted. The string is unlikely to be evenly spaced between the phototransistors. Adjusting the height of the PCB mount involves making the best compromise in data resolution between the two phototransistors. Variations in the data resolution are acceptable though, and what is most important to avoid is non-monotonic behavior. The goal in adjusting the height of the PCB mount is maximising data resolution while avoiding non-monotonic regions. Once the PCB mount heights are properly adjusted, the sensors are then calibrated using the calibration device (Associated pictures in figure A-7).

The final two-stringed instrument is shown in figure 7-1.

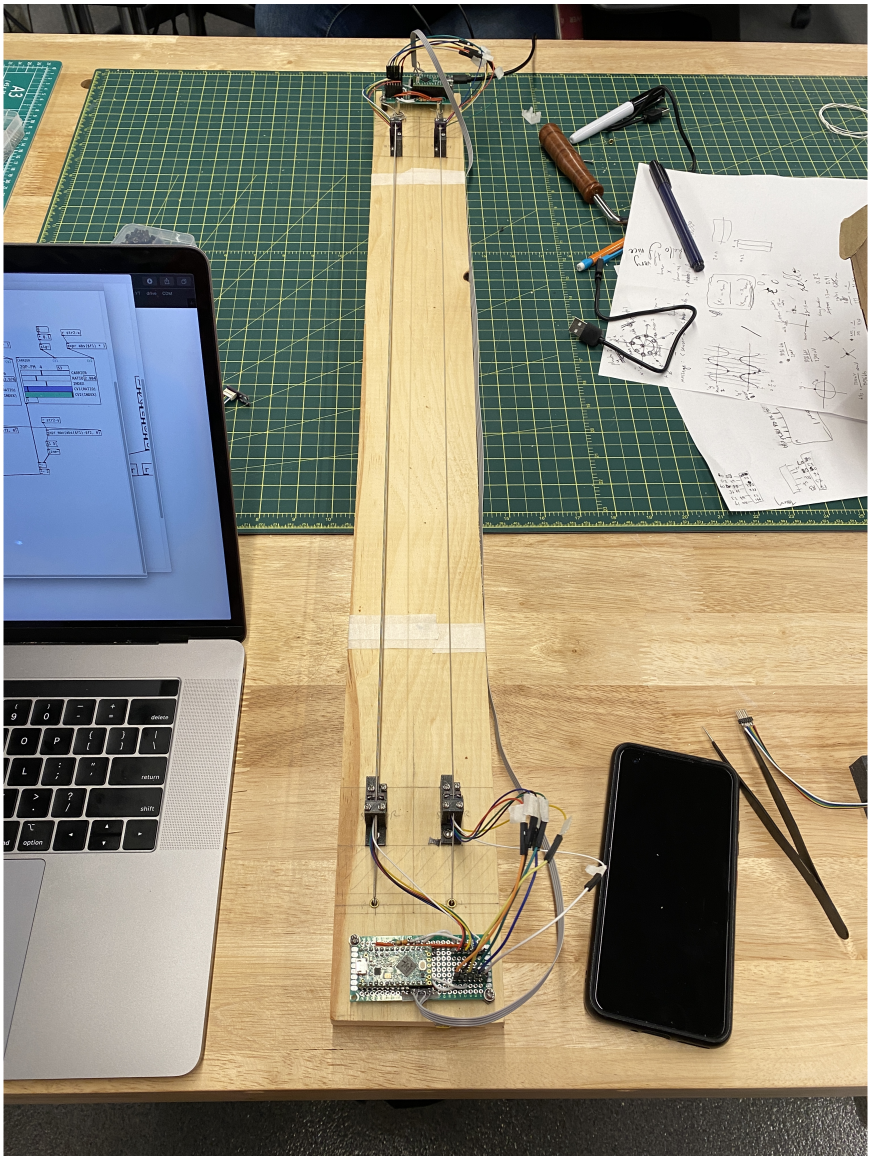


Figure 7-1: Completed two-string instrument

7.2 Performance Considerations

Once the data we got from our sensors was processed in the Python script, it is sent via OSC [1] to an audio patch in PureData[2] to generate sound. For initial testing, we created a simple FM synthesizer in PureData. We mapped the amplitude of the synthesizer to magnitude of displacement and the index of modulation to the angle of displacement. While instrument evaluation is extremely subjective, we found that this sensor encouraged slow, precise motions in which the string is displaced. The sensor also produced interesting oscillating audio effects when the string was plucked.

Although the sensor is designed to avoid a non-monotonic output, it is possible for performers using this sensor to enter a non-monotonic range. We assume that performers will perform gestures outside the range of calibration, which might in turn generate unexpected results. This is not necessarily a problem; the unexpected change in sound could be a parameter that a performer enjoys.

7.2.1 Accounting for Various Resting Positions

At a string's resting position, the sensor data sent out should be $(0, 0)$ for the x-y sensor plane. The patch is designed not to generate sound when the string is at $(0, 0)$; however, while interacting with the instrument, we found that the resting position of the string was not constant. The bridge and string clamp are not strong enough to keep the string's resting position from shifting. To compensate, the patch was altered to have a small dead zone around $(0, 0)$, but this finding strongly suggests manufacturing the next iteration of the sensor out of a more robust material.

7.2.2 Sampling Rates

While using the instrument, it became apparent that the rate at which data was sent to the patch was not fast enough to be perceptually smooth. The current software sends the data via OSC in the same process as updating the GUI. Since the GUI only

[1] https://opensoundcontrol.stanford.edu/index.html
[2] https://puredata.info/

updates approximately every millisecond, the fastest that data ever gets sent to the patch is at a rate of 1kHz[3]. As mentioned in Chapter 5, at a rate of 115200 baud, a worst-case 1440 sensor packets get sent per second, which means that for 4 sensors, each sensor sends a packet 360 times per second. Increasing the serial baud rate of the Teensys as well as separating the GUI process from the OSC message-sending process would increase the rate at which our patch receives audio data, resulting in a smoother auditory experience.

[3]https://doc.qt.io/qtforpython-5/PySide2/QtCore/QTimer.html

Chapter 8

Conclusion

This research aimed to create a robust, low-cost string displacement sensor for digital musical instruments. We chose optical sensing for our sensor design because of our goal to have the sensor work on any type of string. We created multiple prototypes and iterated on them until settling on the final version. The main goals of importance while iterating on our sensor design were to minimize size, to avoid non-monotonic regions, and to simplify manufacturing. We met many of our sensor design goals:

1. Our final sensor had support for 1.1cm string-to-string distance, which was very close to our original goal of 1.0cm.

2. The sensor height adjustment mechanism allowed us to maximize resolution and avoid non-monotonic regions.

3. The sensor enclosure can be easily 3D printed, and the sensor PCB is small and inexpensive to order in bulk.

4. We did not have any cross-talk between strings; the beam of light emitted from an IR emitter on one sensor would not be detected by a phototransistor on another sensor due to the angling of the phototransistors.

5. We were able to differentiate between displacement in both directions orthogonal to the string.

6. While our sensor has the ability to detect oscillation up to the sampling rate of our microcontroller, our focus shifted much more towards displacement sensing.

In addition meeting the sensor specific goals we laid out, our sensor also adheres to design principles presented in Hattwick 2017[15].

1. Principles of functionality: Implementing a novel sensor was risky, but after reviewing existing systems and finding that they were not suitable for our use case, we started from scratch. We validated that the sensor works using basic quantitative tests, but we did not perform extensive quantitative tests to determine precisely how well the sensor works.

2. Principles of manufacturability: We expected to iterate on the sensor design multiple times, so we chose rapid prototyping techniques such as 3D printing and milling our own PCBs to manufacture the sensor quickly. Once the PCB design was changed such that it could no longer be milled in-house, we ordered it in bulk from a manufacturer.

3. Principles of robustness: The sensor is made up of the bridge, PCB mount, PCB, and string clamp. If any of these pieces breaks, we can manufacture a replacement and use it to repair the rest of the sensor. Appropriate materials were used for designing the sensors at the current advanced prototype stage. More advanced versions of the sensor should use stronger materials, such as aluminum, especially for the bridge and string clamp due to the stress applied to them.

4. Principles of reusability: Throughout the project, the 3D models for the sensor enclosure, PCB schematics and gerbers, as well as relevant code have been stored in a Github repository.

8.1 Future Work

While this iteration of the string sensor meets many of the goals we set for it, there is a lot to work on to improve the sensor. Something we had considered but never implemented was soldering the all of the circuitry (biasing resistors for phototransistors, decoupling capacitor, etc) onto the PCB itself to increase simplicity and modularity of the sensor. Something to consider when adding these components the potential size increase of the PCB and whether it would impact the goal of having 1cm string-to-string distance between sensors.

Our sensor communicated with software via serial, but even the final version needed manual setup in the form of launching the python script and initializing the sensor at its serial port. It would be nice in the future to have more plug-and-play functionality with the sensor where very little setup is required, more closely resembling a professionally manufactured sensor. Along with making the software usage process more user-friendly, the data acquisition process in the software needs optimization. While the pyqtgraph framework is extremely useful for creating a visual aid to adjust the PCB mount position, it is not optimized for speed, so our attempts at sending data to synthesize sound from our python script were somewhat laggy. This problem could be resolved by having two separate scripts or processes- one to run for calibration, and another for actually sending data to synthesize sound.

Another detail to make the sensor more polished would be to mill the bridge and PCB enclosure out of metal instead of 3D printing them. Another tradeoff is made here in that the 3D printed version is more accessible, but a metal enclosure for the sensor would have much more structural integrity.

The calibration process for the sensor is another component that requires future work. While the current method of calibration produces meaningful data, it is very dependent on how well the user moved the string in circles. Imperfections in moving the string in circles can result in needing to recalibrate the string in order to acquire meaningful data. In future iterations of the sensor, we would like a more robust calibration procedure. We are considering using a force sensitive resistor in addition to a modified calibration device similar to the current one to measure force as well as displacement, as sensing force may be more applicable to the musical experience of interacting with strings.

Regardless of these considerations for future improvements, the sensor that we created is suitable for a variety of stringed DMIs and will support further research in human-computer interaction and musical performance.

Appendix A

Instrument Assembly Photos

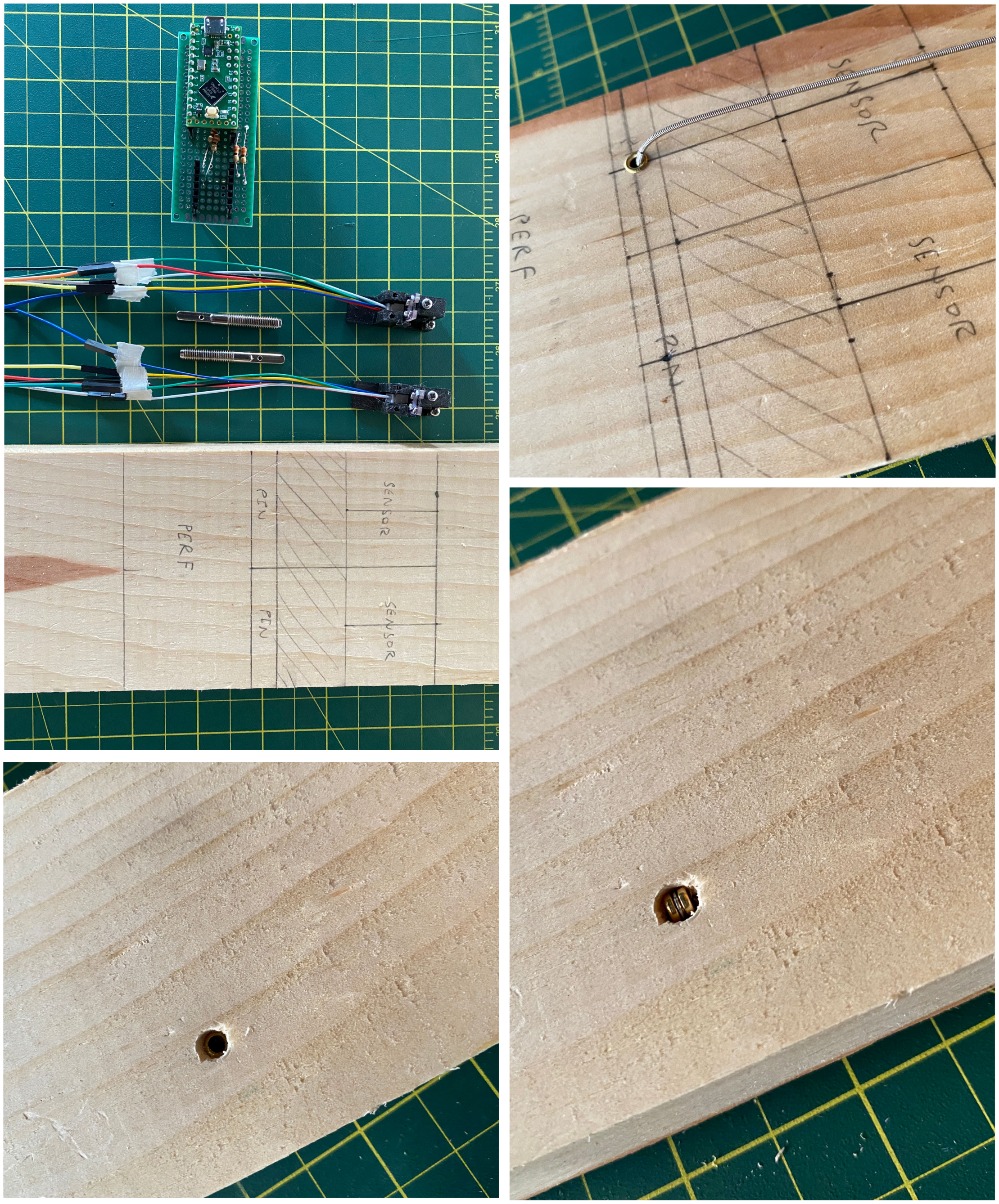


Figure A-1: Counterclockwise from top left: 1) Layout of one side of the instrument showing space for sensors, tuning pins, and perf board containing Teensy LC and other circuitry. 2) Back of instrument, hole for string showing countersunk ferrule. 3) Back of instrument, hole with string with ball end inserted. 4) Top of instrument showing second ferrule with string coming out through it.

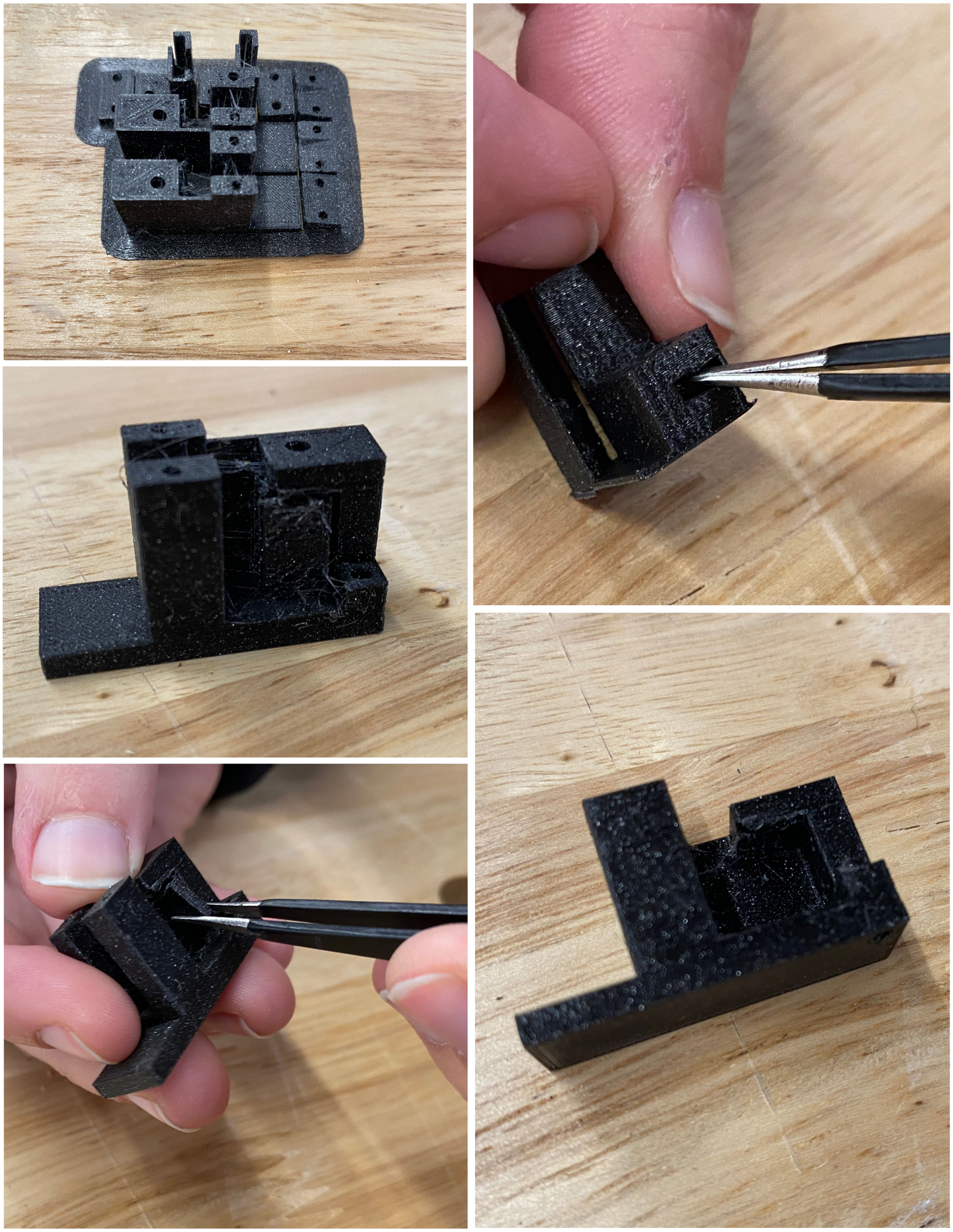


Figure A-2: Counterclockwise from top left: 1) Completed 3D print of 2 sensor enclosures. 2) Bridge with support material still attached (under right screw hole overhang). 3) Cleaning out support material from bridge. 4) Cleaned out bridge. 5) Cleaning out hex nut hole in PCB mount.

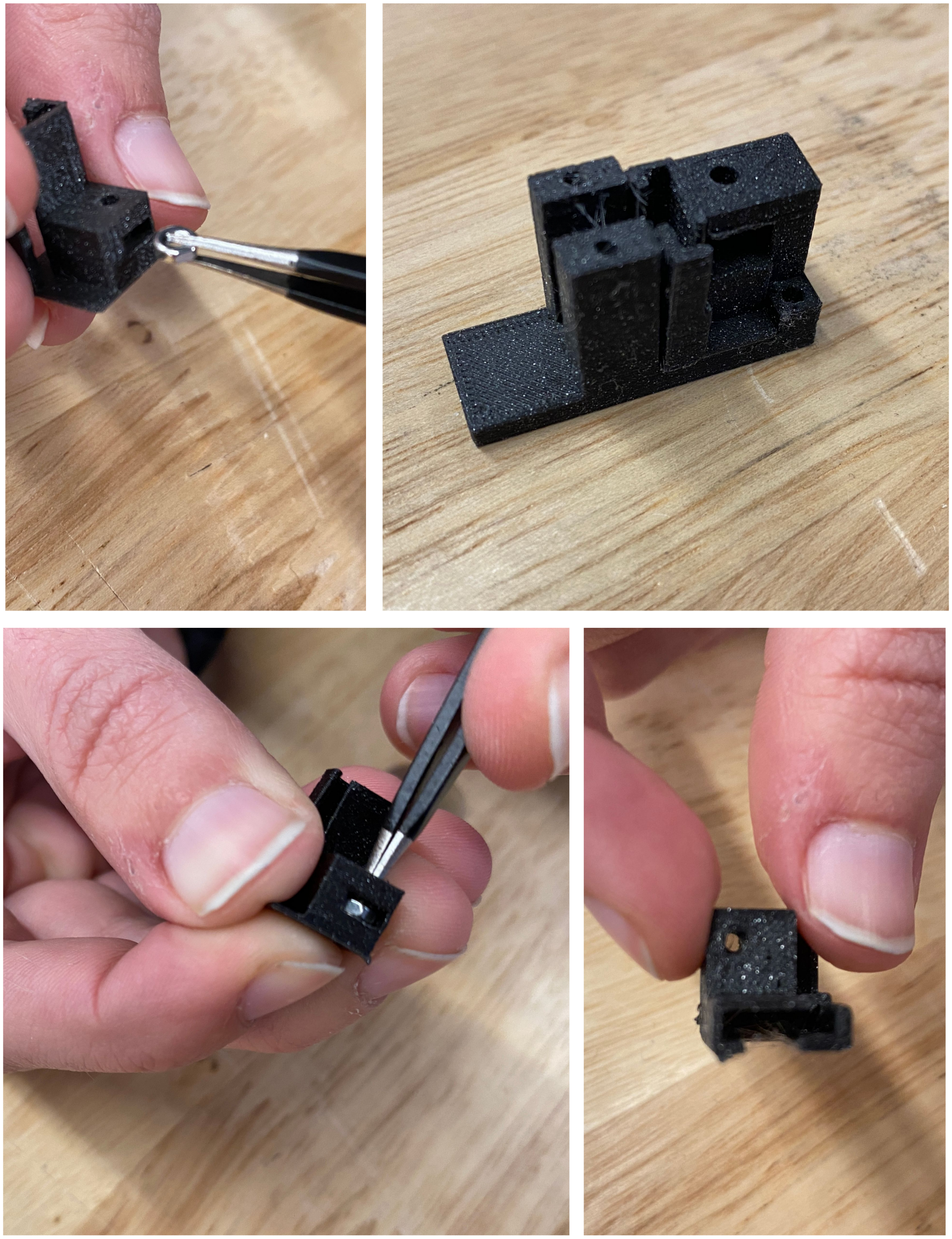


Figure A-3: Counterclockwise from top left: 1) Inserting M2 nut into PCB mount. 2) Using tweezers to pull the nut into place. 3) Correct placement of nut viewed from above- the screw hole is unobstructed. 4) Prepared PCB mount slid into bridge ready for further assembly.

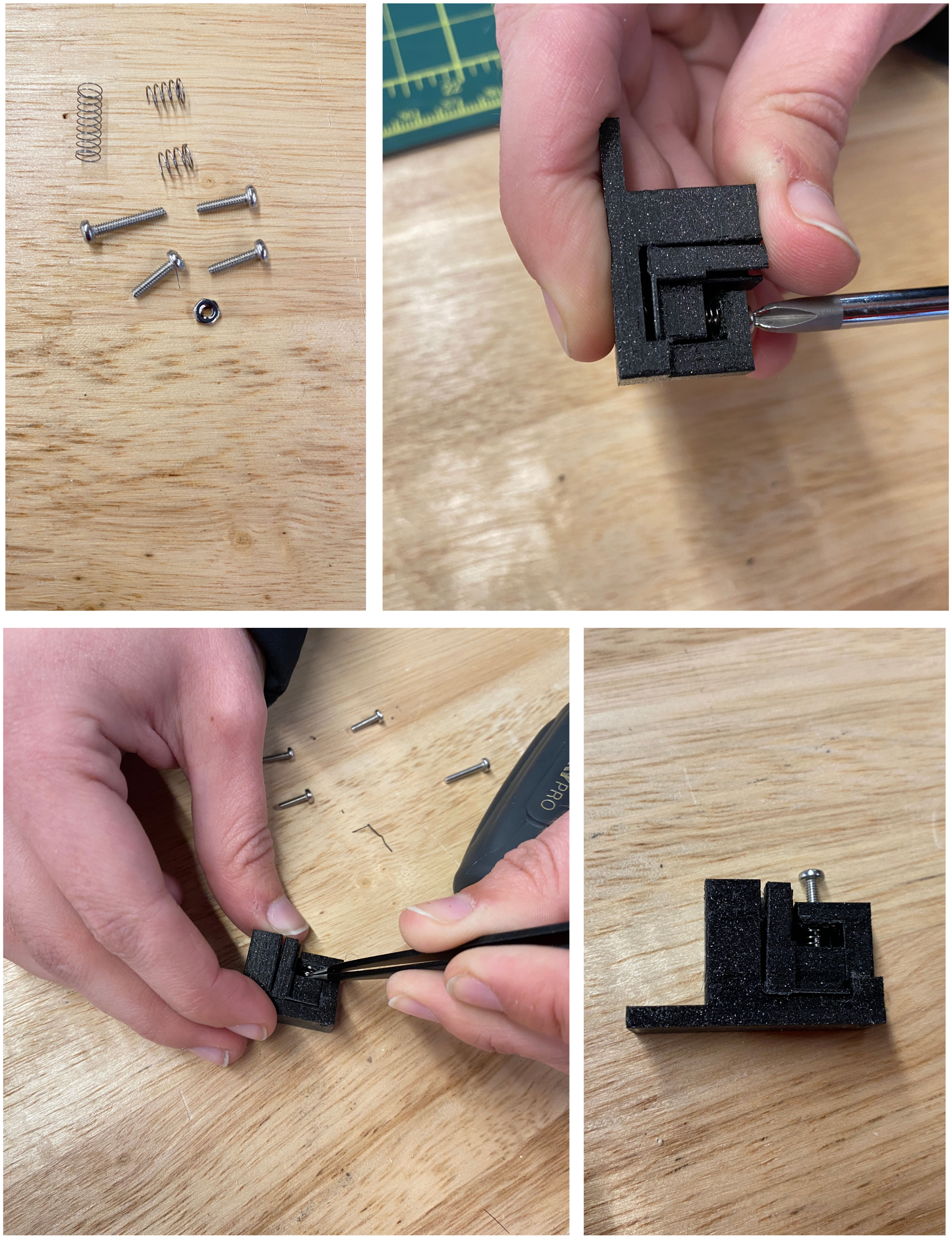


Figure A-4: Counterclockwise from top left: 1) Screws, spring, and nut needed to assemble the sensor. The spring used is made by cutting a longer (top left) spring in half. 2) Using tweezers to insert the spring between the bridge and PCB mount. 3) Correct placement of spring- M2 screw easily screws through bridge and into nut in the PCB mount. 4) PCB mount height can be changed by tightening or loosening the screw.

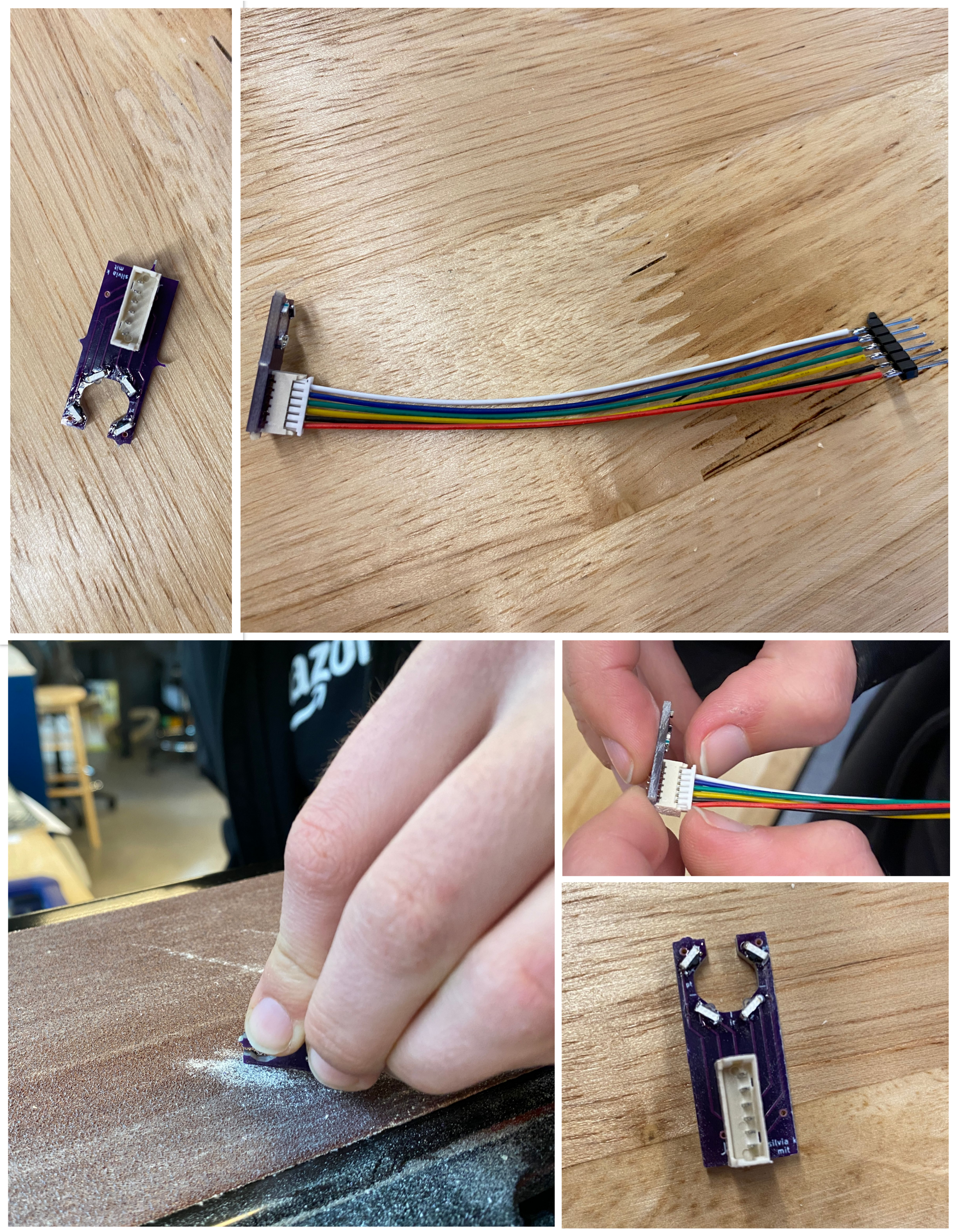


Figure A-5: Counterclockwise from top left: 1) PCB with all components soldered in place, but with tabs from manufacturing. 2) Sanding PCB to remove tabs. 3) Sanded PCB that can now fit in PCB mount. 4) Attaching wires to PCB header. 5) Completed PCB with wire from PCB headers to standard header size.

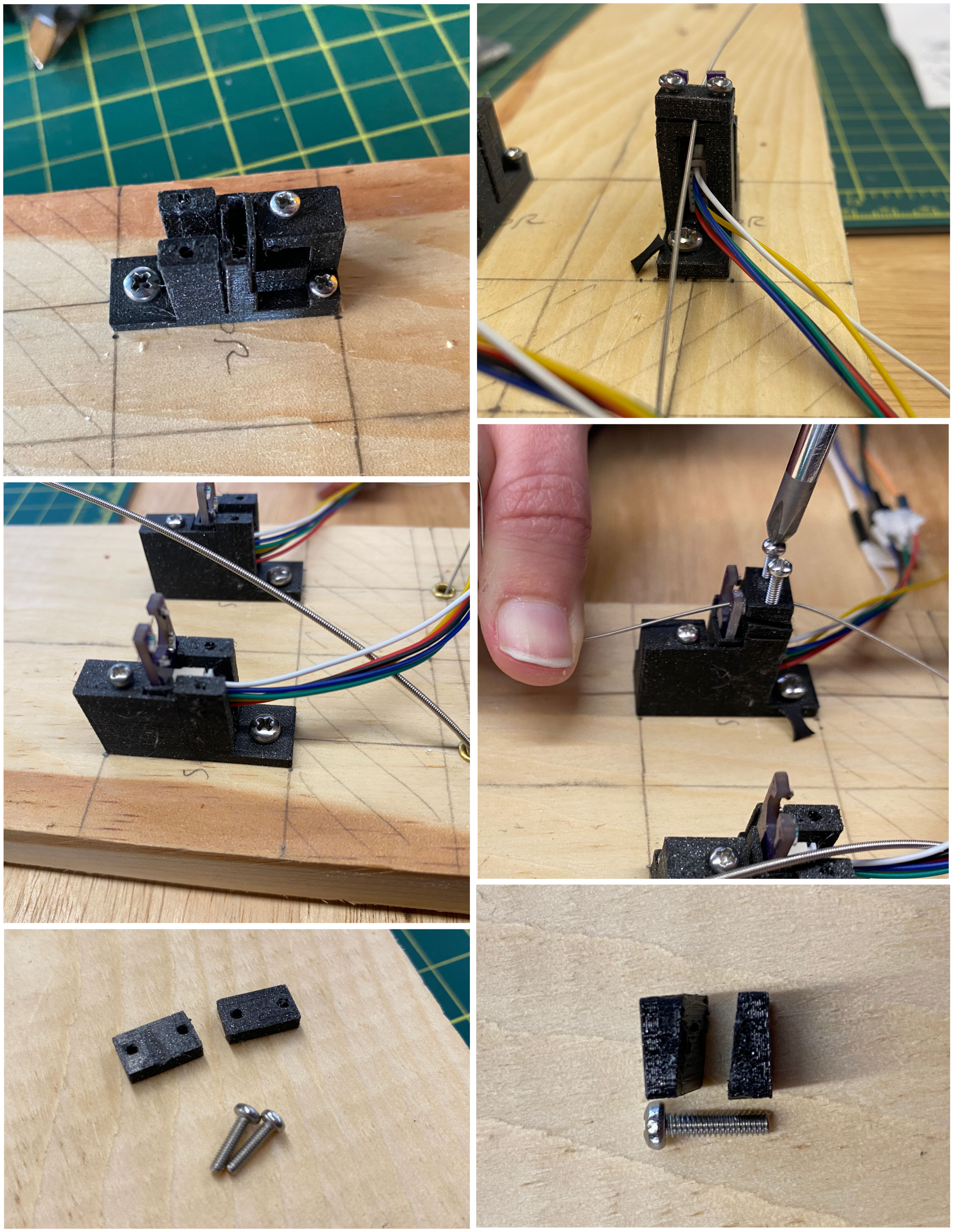


Figure A-6: Counterclockwise from top left: 1) Sensor enclosure mounted onto the top of the instrument. 2) PCB inserted into PCB mount. 3) Top of bridge (left) and string clamp with the screws used to mount them. 4) Placement of bridge top, string clamp, and screw that shows how they are aligned. 5) Screwing the bridge top and string clamp on. 6) Final assembled sensor with string.

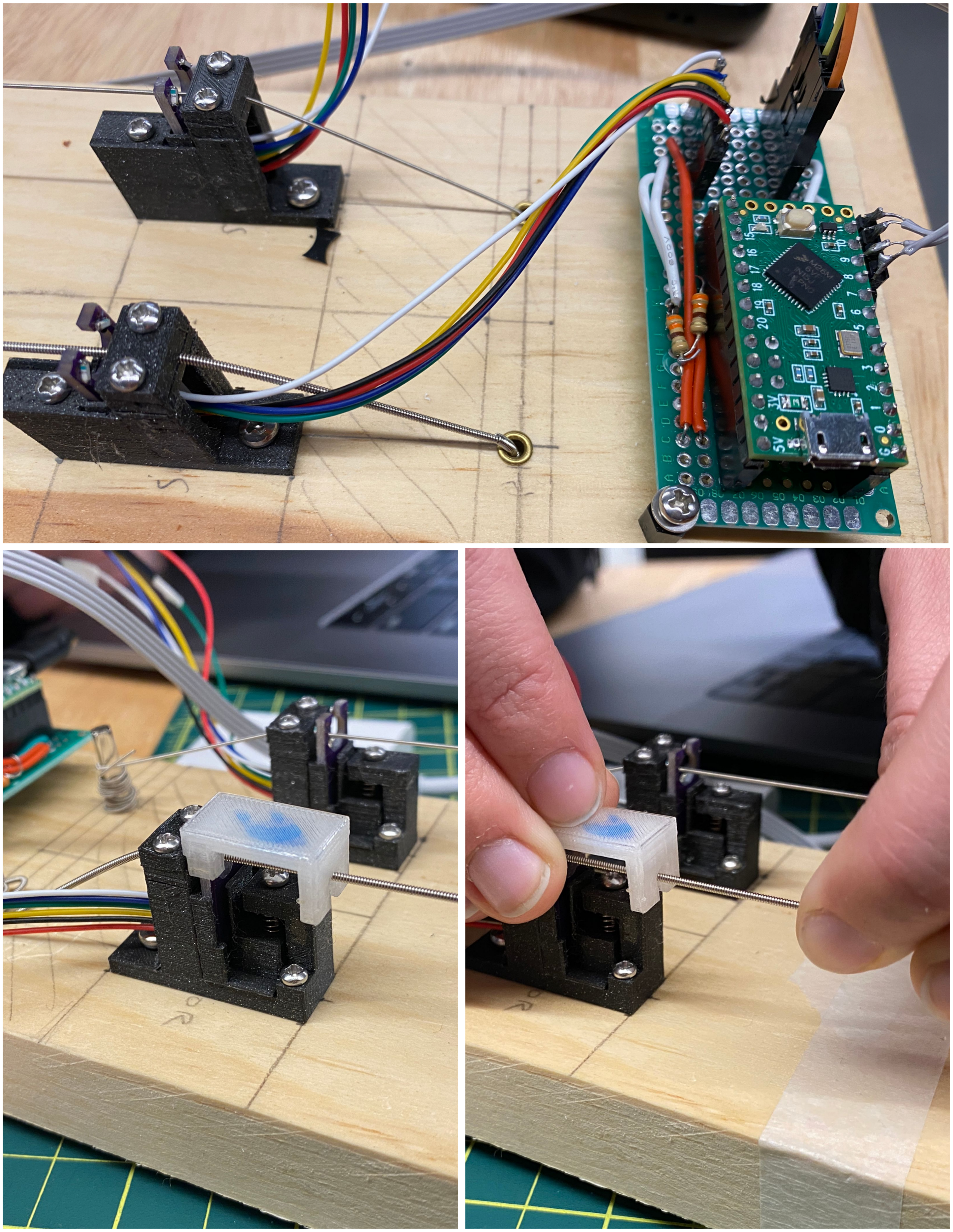


Figure A-7: Counterclockwise from top: 1) Two assembled sensors connected to the Teensy. 2) Using the calibration device to calibrate the string with the software. 3) Moving the string in circles around the calibration device.

Bibliography

[1] Nonstandard for transmission of IP datagrams over serial lines: SLIP. RFC 1055, June 1988.

[2] Marije A. Baalman. The STRIMIDILATOR: a String Controlled MIDI-Instrument. June 2003. Conference Name: International Conference on New Interfaces for Musical Expression Pages: 19-23 Publication Title: Proceedings of the International Conference on New Interfaces for Musical Expression Publisher: Zenodo.

[3] Adrián Barenca and Milos Corak. The Manipuller II: Strings within a Force Sensing Ring. pages 589–592, June 2014. Conference Name: International Conference on New Interfaces for Musical Expression Pages: 589-592 Publication Title: Proceedings of the International Conference on New Interfaces for Musical Expression Publisher: Zenodo.

[4] Adrián Barenca and Giuseppe Torre. The Manipuller: Strings Manipulation and Multi-Dimensional Force Sensing. page 4, 2011. Proceedings of the International Conference on New Interfaces for Musical Expression - 30 May - 1 June 2011.

[5] Adan L Benito Temprano and Andrew McPherson. A tmr angle sensor for gesture acquisition and disambiguation on the electric guitar. In *Audio Mostly 2021*, pages 256–263. 2021.

[6] Bert Bongers. An interview with sensorband. *Computer Music Journal*, 22(1):13–24, 1998.

[7] Bert Bongers. Electronic musical instruments: Experiences of a new luthier. *Leonardo Music Journal*, 17:9–16, 2007.

[8] J.-L. Le Carrou, D. Chadefaux, L. Seydoux, and B. Fabre. A low-cost high-precision measurement method of string motion. *Journal of Sound and Vibration*, 333(17):3881–3888, 2014.

[9] Stuart Cheshire and Mary Baker. Consistent overhead byte stuffing. *Networking, IEEE/ACM Transactions on*, 7:159 – 172, 05 1999.

[10] Perry Cook. 2001: Principles for designing computer music controllers. *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression*, pages 1–13, 2017.

[11] Marcel Ehrhardt, Max Neupert, and Clemens Wegener. Piezoelectric strings as a musical interface. June 2020. Conference Name: International Conference on New Interfaces for Musical Expression Pages: 35-36 Publication Title: Proceedings of the International Conference on New Interfaces for Musical Expression Publisher: Zenodo.

[12] Adrian Freed and Osman Isvan. Musical applications of new, multi-axis guitar string sensors. In *ICMC*, 2000.

[13] Adrian Freed, Devin McCutchen, Andy Schmeder, Anne-Marie Skriver, Dan Overholt, Winslow Burleson, Camilla Nørgaard, and Alex Mesker. MUSICAL APPLICATIONS AND DESIGN TECHNIQUES FOR THE GAMETRAK TETHERED SPATIAL POSITION CONTROLLER. page 6, 2009.

[14] Waleed Sami Haddad. Optoelectronic Pickup for Musical Instruments.

[15] Ian Hattwick. *The Creation of Hardware Systems for Professional Artistic Productions.* PhD thesis, 2017.

[16] D Hopp, Ch Pruss, W Osten, J Seybold, V Mayer, and H Kück. Optical incremental rotary encoder in low cost design. In *Sensor & Test Conference proceeding OPTO*, volume 3, 2009.

[17] Nicholas G. Horton and Thomas R. Moore. Modeling the magnetic pickup of an electric guitar. *American Journal of Physics*, 77(2):144–150, February 2009. Publisher: American Association of Physics Teachers.

[18] Volker Krefeld and Michel Waisvisz. The Hand in the Web: An Interview with Michel Waisvisz. *Computer Music Journal*, 14(2):28–33, 1990. Publisher: The MIT Press.

[19] Nicolas Leroy, Emmanuel Fléty, and Frédéric Bevilacqua. Reflective Optical Pickup For Violin. June 2006. Conference Name: International Conference on New Interfaces for Musical Expression Pages: 204-207 Publication Title: Proceedings of the International Conference on New Interfaces for Musical Expression Publisher: Zenodo.

[20] Joseph Malloch. A consort of gestural musical controllers: Design, construction, and performance. 08 2007.

[21] Mark T Marshall and Marcelo M Wanderley. Evaluation of sensors as input devices for computer music interfaces. In *International Symposium on Computer Music Modeling and Retrieval*, pages 130–139. Springer, 2005.

[22] Carolina Brum Medeiros and Marcelo M. Wanderley. A comprehensive review of sensors and instrumentation methods in devices for musical expression. *Sensors*, 14(8):13556–13591, 2014.

[23] Eduardo Reck Miranda and Marcelo M Wanderley. *New digital musical instruments: control and interaction beyond the keyboard*, volume 21. AR Editions, Inc., 2006.

[24] Sean M O'Connor. Patented Electric Guitar Pickups and the Creation of Modern Music Genres. 23:39.

[25] Dan Overholt. THE OVERTONE VIOLIN: A New Computer Music Instrument. page 4.

[26] Joseph A. Paradiso and Neil Gershenfeld. Musical Applications of Electric Field Sensing. *Computer Music Journal*, 21(2):69, 1997.

[27] Joel Ryan. Some remarks on musical instrument design at steim. *Contemporary Music Review*, 6(1):3–17, 1991.

[28] Joshua Reynolds Smith. *Electric field imaging*. PhD thesis, Massachusetts Institute of Technology, 1999.

[29] Atau Tanaka and Bert Bongers. Global String. January 2003.

[30] Hayami Tobise, Yoshinari Takegawa, Tsutomu Terada, and Masahiko Tsukamoto. Construction of a system for recognizing touch of strings for guitar. In *NIME*, pages 261–266, 2013.

[31] Caroline Traube, Philippe Depalle, and Marcelo M. Wanderley. Indirect Acquisition of Instrumental Gesture Based on Signal , Physical and Perceptual Information. June 2003. Conference Name: International Conference on New Interfaces for Musical Expression Pages: 42-47 Publication Title: Proceedings of the International Conference on New Interfaces for Musical Expression Publisher: Zenodo.