TAGALONG TRAIL

FINAL DESIGN REVIEW

PREPARED BY

Jason Davis

Ryan Johnson

Quinlan Stephens

Vinh Vo

Sean Wahl

cptagalongtrail@gmail.com

Mechanical Engineering Department

California Polytechnic State University

San Luis Obispo

December 8, 2023

PREPARED FOR

Summer Helmuth

Girl Scouts of California’s Central Coast

AND

Matthew Meadows

Girl Scouts of California’s Central Coast

AND

Joseph Metcalfe

Walk Of Wonders

ABSTRACT

The Cal Poly mechanical engineering team working on the Path of Lights and Sounds for the Girl Scouts of California's Central Coast has now completed their project. In the last quarter of their project, they were able to finalize their tile's structural design while iterating through several electronics board designs. Through this iteration, they were able to create a final electronics board and designed a water-resistant enclosure to house the board. A new team member was added, allowing them to manufacture 28 tile structures and 30 electronics assemblies over the course of two and a half weeks. This proved to be an ill-considered decision, as the manufacturing and assembly of all tiles completely consumed their time, even with a Girl Scout event held to help with assembly. Despite this, they were still able to follow through on most testing planned to determine the quality of their design. Though lacking in some specifications, the team evaluated and tested their final design and all 28 tiles to the Girl Scouts with newly guided recommendations for use (based on their testing results). After delivery, the tiles sustained minor damage from an unanticipated electrical condition, but the team was able to diagnose possible causes and provide a solution to the problem as well as updated usage recommendations to avoid future issues. The tiles now reside solely with the Girl Scouts and the team has provided them with guidance for future use and upkeep as well as how to completely reproduce more tiles if desired for future events.

Table of Contents

STATEMENT OF DISCLAIMER

Since this project is the result of a class assignment, it has been graded and accepted as fulfillment of the course requirements. Acceptance does not imply technical accuracy or reliability. Any use of information in this report is made at the risk of the user. The authors of this document cannot be held liable for any use or misuse of the project, or any infringement of property.

1 INTRODUCTION

The Girl Scouts of California's Central Coast (GSCCC) and the Walk of Wonders (WoW) organized a themed Christmas event, and the project team has created a light-up, musical, step-activated path for the attendees. The path's tiles light up, play sound, intercommunicate, and can play "Jingle Bells" when stepped on sequentially. The primary sponsor of this project has transitioned from Joseph Metcalfe of WoW to Summer Helmuth of GSCCC.

Since the Critical Design Review (CDR), the team has made improvements to the design. The primary modifications include a reconfiguration of the support structure and an optimized placement of the springs. Additionally, the team has had great success in creating an electronics board capable of interfacing with light emitting diodes (LEDs) and speakers. Code was written for the tiles to allow them to function properly. A graphical user interface (GUI) was also created for reprogramming the tiles for other songs and color schemes.

This detailed report is structured into four principal sections: Design Overview, Implementation, Design Verification, and Discussion & Recommendations. Together, they provide a thorough outline of the team's final design, procurement of materials, manufacturing and assembly processes, and testing. Recommendations regarding the future use of the tiles and possible senior projects are listed as well.

The Design Overview is to provide a summary of the structural edits the team made to the tile since the CDR. It also covers the full electronics design of the tiles, as this was fully developed after the CDR had occurred. This not only includes the printed circuit boards (PCBs) that the team designed, but the full enclosure that houses the PCBs and a wiring diagram for each electronics assembly.

The Implementation section details the sourcing of materials, manufacturing processes for each custom component, assembling the full tile, and references the User Manual which contains the full build instructions. Implementation includes both the tile's structure and electronics. Along with the electronics themselves, the software that implements the tile's functionality is explained. This section also outlines the team's challenges, including delays in shipping, manufacturing issues, and unexpected technical difficulties.

In the Design Verification section, compliance with the specifications that were developed in the Scope of Work (SoW) is evaluated. Specifications met by observations and met by testing are both covered in this section. The Design Verification Plan and Report (DVP&R) is referenced for further testing information. This section covers various tests performed, the numerical data collected with uncertainty analysis, and includes recommendations derived from how the tiles performed in testing.

The Discussion & Recommendations section reviews what was learned during the project, identifies potential improvements, and suggests how the project should be handled moving forward. It discusses how the tiles are faring so far and what has been observed in their short-term use along with updated guidelines of use based on this information. The sponsor is greatly urged to reference this section when deciding to move forward with the future development of the tiles.

The team has prepared this document to provide the sponsors with essential information about the project. The sponsors are encouraged to refer to the complete report. While all sections are important, particular emphasis should be taken in reading the Discussion & Recommendations section as well as the User Manual. All material in this document is subject to the sponsor's feedback, though the team is no longer required to partake in the project after delivery of this document. If needed, the sponsor is urged to hire outside help to receive technical support and maintenance for the long-term life of the project.

2 DESIGN OVERVIEW

The tile design is comprised of two primary systems: Structures and Electronics. Electronics includes software as well, but it shall be covered in Section 3, Implementation. Each system is mentioned below along with its breakdown and where it will be covered:

1. **Structures:** This section captures all structural components of the tile, comprised of two primary subassemblies (P/N 110 & 120 as referenced in **Appendix A – Bill of Materials**). It covers the tile's frame, including everything except the electronic elements, such as LEDs, limit switches, electronics enclosure, and speakers. This part of the design was focused on in quarter 2 of the project and is extensively covered in the Critical Design Review (CDR). Modifications to the hardware will be covered in Section 2.1 below.
2. **Electronics:** This section refers to the electronic components that are excluded in the Hardware section. These parts require careful assembly and connection as per the guidelines provided in Section 3 of the User Manual. The precise integration of these electronic elements is crucial for the tile's functionality. A general overview of the electronics' capabilities are in Section 2.2 below.

While this section will give a general overview of each part, sponsors looking for in-depth guidance should refer to **Appendix B – User Manual,** hereby referred to as the User Manual. The User Manual contains information regarding all the manufacturing, assembly, coding, programming, and setup of the tiles. The safe use of the tiles is also of high priority, and the risks associated with the project can be found in **Appendix C – Risk Assessment**. From the risk assessment, the tiles were found to be perfectly safe when the proper protections are in place.

2.1 DESIGN MODIFICATIONS

Most modifications made to the design stemmed from iterating the Structural Prototype until it became the Verification Prototype. While these changes are discussed in this section, the Verification Prototype in its entirety will be listed in Section 3.1.

Following the Critical Design Review (CDR), the team revised the previous support structure, transitioning from four 90-degree aluminum bars to eight Ultra-High Molecular Weight (UHMW) polyethylene rods. This enhancement not only advanced the aesthetic appeal of the design through superior light diffusion but also offered a more efficient load distribution. The increased number of support rods enables the structure to accommodate a greater weight capacity, thereby enhancing user safety and the overall integrity of the design. Figure 1 illustrates the new support's design.

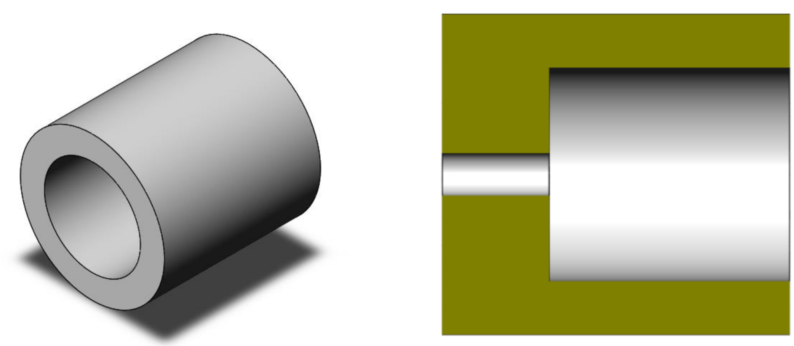


Figure 1 – Updated support design.

Originally, the design included four aluminum bars, each positioned at a 90° angle, forming a cross in the middle of the tile. This configuration, while structurally effective, had two significant downsides. It limited the passage of light inside of the tile, leading to uneven light distribution across the tile's surface. It also created four large unsupported areas that were weaker and more prone to breaking.

The updated support system has eight individual supporting rods as visualized in Figure 2, and represents a considerable advancement from the previous model. The new structure is designed to let more light through the tile, avoiding the shadowy effects caused by the old cross-like design. The figure shows how the supports are now arranged to allow maximum light penetration, ensuring a brighter and more evenly lit area.

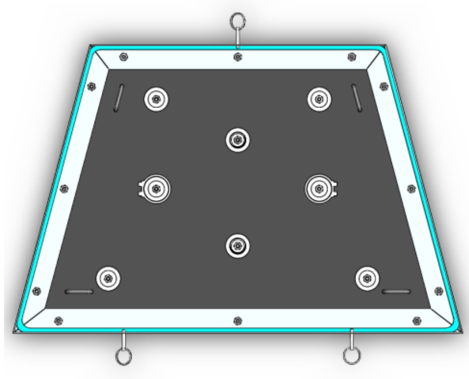


Figure 2 – Updated support location.

The design also includes a subtle change, such as the placement and size of the springs. The team opted for two larger springs capable of handling 4.25 pounds for a 0.25 inch displacement of the top surface. These springs are nested within the support rods, improving the 'bounce-back' translation, and enhancing durability. Figure 3 provides a detailed illustration of the new location of the springs.

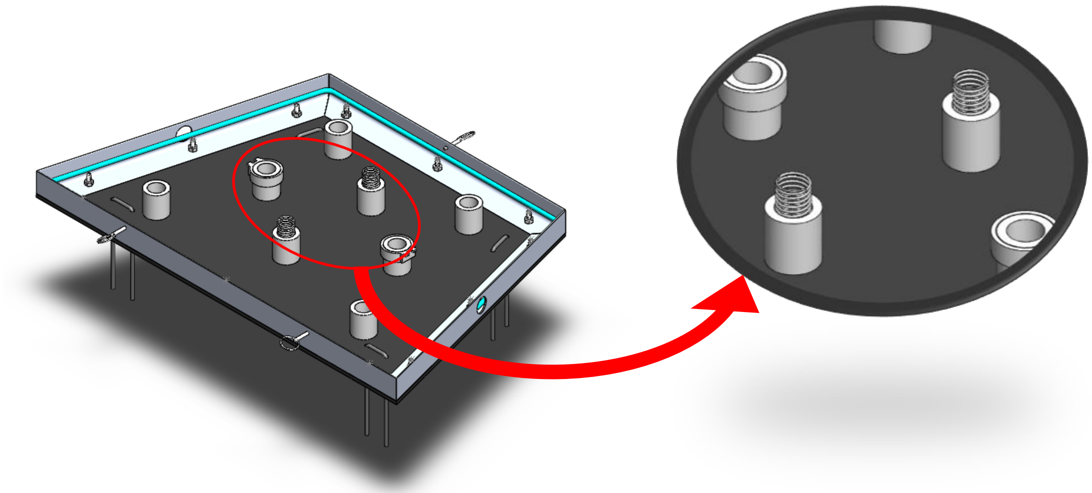


Figure 3 – Updated spring location.

The final update to the design includes a limit switch bracket. This bracket is meant to keep the limit switch in place just above the support, as seen in Figure 4. It attaches to the support with a press fit and one set screw on the side, ensuring a tight and secure connection. This is essential because the materials of the bracket (PLA, used in 3D printing) and the support rod (UHMW) can expand or contract differently in extreme temperatures. Using both a press fit and set screws is a reliable way to keep the bracket firmly in place, even if the materials expand and the fit becomes looser due to heat.

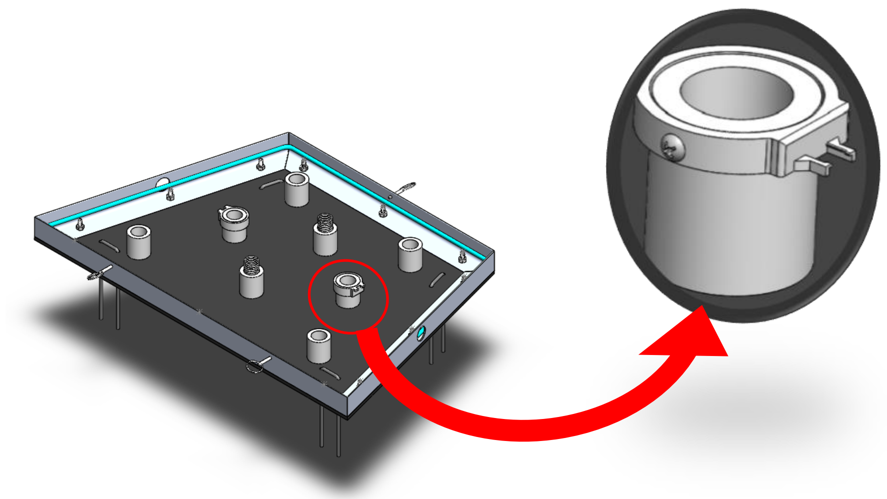


Figure 4 – Limit switch bracket.

The bracket has been designed to simplify the process of inserting and removing the switch. This is achieved through a combination of rails and a snapping mechanism. The rails ensure the switch's slot moves in and out of the bracket smoothly, while the material's flexibility securely snaps it into place. This design makes the entire process effortless. Figure 5 provides a visual demonstration of how the switch is installed to the brackets.

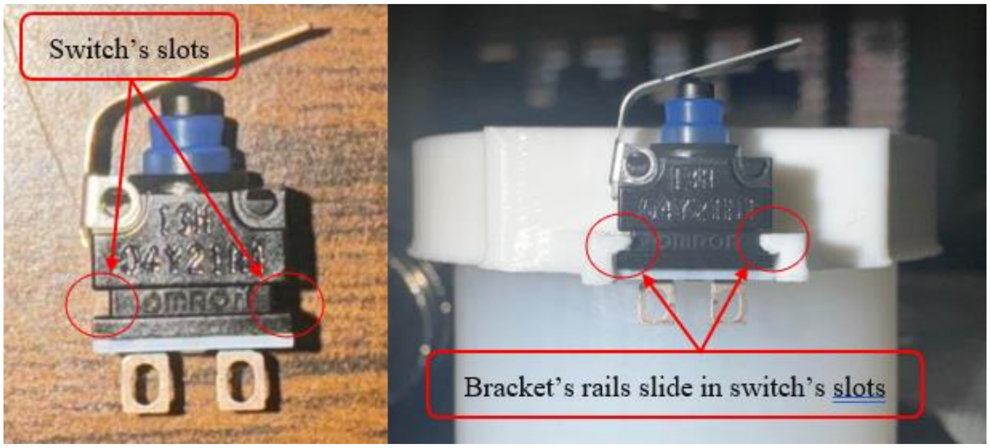


Figure 5 – Limit switch with brackets.

2.2 ELECTRONICS DESIGN

The electronics were designed almost completely over the summer and in the third quarter of Senior Project. Prototyping had to progress quickly, so the team donated some of their own funds to speed this process up. The tile's electronics needed several basic functions or features to be considered a success:

- Microcontroller to process code.
- Ability to generate sound waves and interface with a speaker.
- Ability to modulate power to control and interface with red, green, and blue LEDs.
- Hardware to take in an input voltage and regulate it to the required 12V and 5V for the LEDs and speakers respectively.
- Connections and communication protocol to interface with adjacent tiles.
- Limit switches to detect when the top surface is depressed by a step.

To make this process slightly easier, the team decided to design their electronics board off of a Raspberry Pi Pico microcontroller, eliminating the bulk of the first bullet point above. The board would allow the Pico to be inserted into the board's headers, making it easily removable if the Pico itself breaks. On top of this, the Pico is only $5 per board. While it would be cheaper to incorporate the microcontroller directly on the team's board, $5 is a price the team was willing to pay to use premade, working boards, avoiding any complications with getting the RP2040 (the Pico's processor) integrated with their PCB. Reference Figure 6 to see what the Pico looks like.

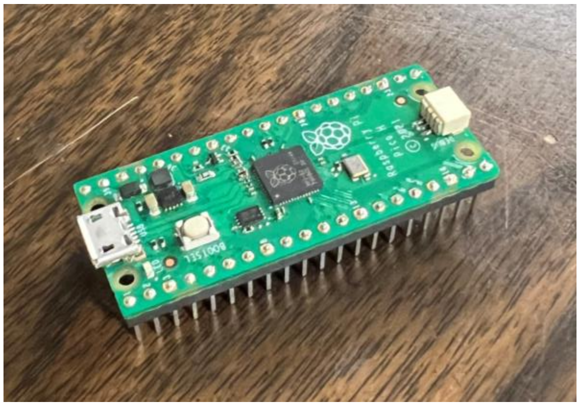
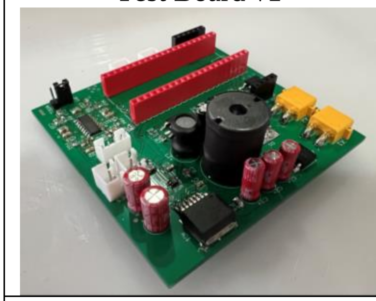
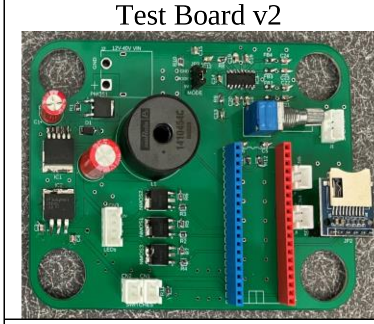


Figure 6 – Raspberry Pi Pico microcontroller board.

Two versions of test boards and a final board were made throughout the electronics design process. The test boards will not be covered in depth, though notable features of each board along with suggestions for improvement are covered in Table 1 below. Additionally, the table shows what each board looked like.

Fusion 360's PCB (printed circuit board) software was used to design all of the boards. The Final PCB file is attached in the Software Package delivered to the sponsor. In that file, previous landmarks can be reverted to in order to see Test Board v1 and Test Board v2. However, moving forward should only entail building off of the Final Board. How to reproduce the Final Board is covered in Section 3 of the User Manual.

Table 1 – Electronics board revision chart.

Board Version	Notable Features	Comments
Test Board v1 	• Runs off of a Raspberry Pi Pico (RP2040 processor) • Capability to run LEDs off of MOSFETs and GPIO Pins • Capability to produce sound waves using PWM and an audio driver • Two power ports	• Board layout needed to be considered more • Add mounting holes • Add audio filtering and volume control to improve sound quality • Need to remove one power port
Test Board v2 	• Audio filtering and volume control • Considered connector layout • Added mounting holes • Switched 5V regulator from switching to linear	• Volume control not necessary, need to remove • Connector layout should align with enclosure glands • Linear regulator too inefficient • Power port at a bad angle • Entire board slightly too big
Final Board 	• Back to 5V switching regulator • Connectors placed in front of gland locations • Test points added • Clip-on points added • Gave Pico space to plug-in cable while connected to board • Fits in enclosure	• Soft-start functionality would be a luxury to have • A better enclosure is desired • More space for the JST connectors is desired • A waterproof coating could be sprayed on to ensure waterproofing

To be explicitly clear, the Final Board takes power in from 12-40V (desired 36V) and regulates it down to 12V and 5V. The 12V signal is used to power the LEDs and feeds the 5V regulator. The 5V line powers both the Pico and the speakers. The Pico has an on-board regulator down to 3.3V. The Pico drives the LEDs using pulse width modulation (PWM) and MOSFETs connected to the 12V line. The Pico also drives the speakers using PWM, though it feeds it to an audio driver to create an approximated analog signal. For communications with other tiles, universal asynchronous receiving/transmitting (UART) was used. When tested, the Final Board drew a maximum of around 40W while playing sounds and setting the LEDs white.

For a virtual picture of the Final Board, see Figure 7. In this photo, all connectors and components are visible. Notably, there are the following connectors:

- 6 JST connectors
  - 2 JST B2B-XH for limit switch connections
  - 2 JST B2B-XH for UART communications forward and backward
  - 1 JST B4B-XH for LED connector
  - 1 JST B4B-PH for speaker
- 1 XT-60 power connector
- 1 Six-pin header for an SD card breakout board
- 2 Twenty-pin headers for the Pico

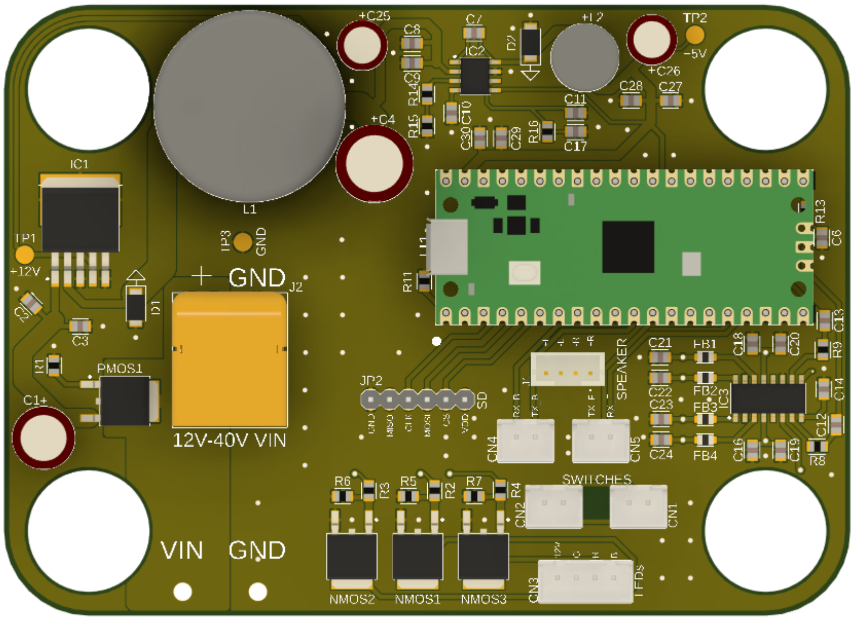


Figure 7 – Fusion 360 PCB model of the Final Board.

Special care was taken to ensure the PCB maintained a necessary degree of water resistance. The team therefore employed inexpensive plastic containers to house the PCBs. While more highly rated enclosures exist and remain an option for the Girl Scouts in the future, this was determined to be an acceptable compromise given budgetary constraints. They placed sealant beads along the rim of the enclosures, as seen in Figure 8. They also used waterproof IP68 PG glands to run wires in and out of the box. Since these glands were meant for single wires, they were also sealed shut with sealant. One was used for the power cable and the other was used for all other connectors' cables. This can be seen in Figure 9, which also shows the board installed with speakers mounted to the side of the enclosure.


Figure 8 – Electronics enclosure box with sealant bead around rim. The sealant is white.

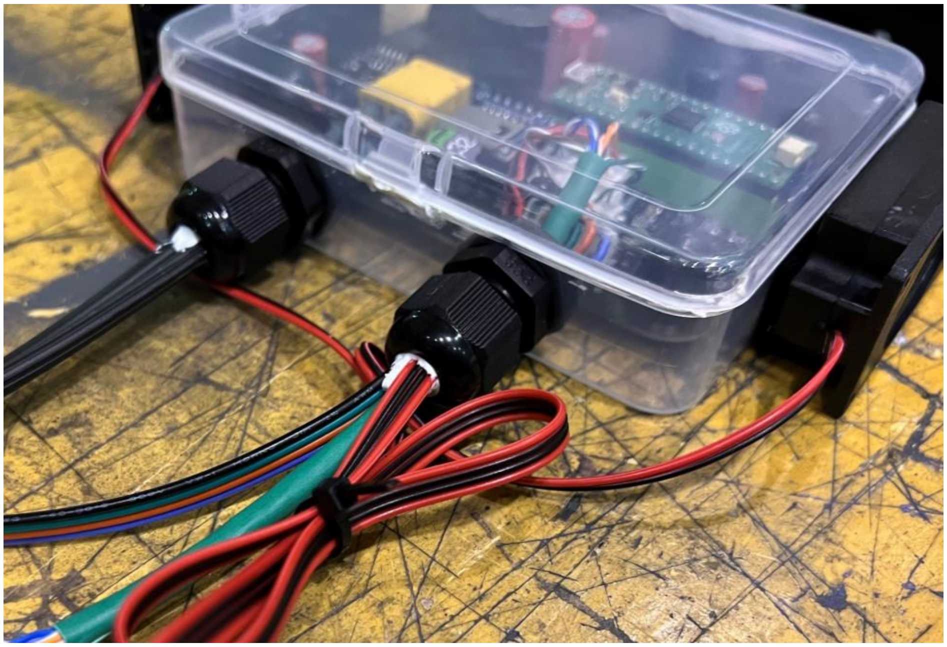


Figure 9 – Electronics enclosure with waterproof glands installed and sealed. The final board is also installed and the speakers are mounted to the side using Velcro.

For a full guide on the Final Board along with the complete electronics enclosure, please reference Section 3 of the User Manual. Not only will the User Manual detail the design, it guides the user through making a full electronics enclosure for future tile use. Many pictures are included as no formal drawings for producing the electronics were created. Important notes covered in the User Manual include where to access the PCB files, how to code a Pico, and how to configure the wiring. Just to be clear, the connectors used outside of the enclosure are all waterproof connectors.

2.3 POWER SUPPLY

Because each enclosure can consume a maximum of around 40W, a sufficient power supply needed to be sourced to properly provide this power. Given that “Jingle Bells” contains 25 notes, an absolute maximum consumption would lie around 1000W, though this is unrealistic. Realistically, if two users are on the tiles at once, 2 tiles would be fully on while several may only have LEDs active. Assuming that tiles stay on for 1 second after being stepped on and that “Jingle Bells” covers around 2.5 notes a second, we can guess that perhaps 6 tiles would be on, but only consuming about 20W. This puts the tiles at around 200W of consumption.

The previous paragraph fails to acknowledge misuse of the tiles. Children are unpredictable and may not follow the tempo of the song. In the worst case, they may run straight through all of the tiles. This would have several tiles fully on and many more with only LEDs on. Several actions were taken to safeguard against this scenario:

1. The “After Color” that lights up after the tile is stepped on is set to 25% brightness to reduce power consumption.
2. The tiles are set to only stay lit up for 1 second after the sound stops playing.
3. There is a 7 second delay at the start of the path to reduce the number of people using the attraction at one time.

With all this in mind, the team safely decided to source a power supply from DigiKey that could provide 600W at the desired 36V for the Final Boards to run off of. See Figure 10 for the power supply choice.

Figure 10 – The LRS-600-36 power supply from MEAN WELL, sourced off of DigiKey. Capable of providing 600W at 36V. Takes an input of 110VAC or 220VAC if configured properly.

This power supply alone does not have any protection measures implemented. The team decided to put a switch and a ground-fault circuit interrupter (GFCI) outlet to ensure safe use of the power supply. On top of this, the team put the GFCI and LRS-600-36 inside of a waterproof box from Home Depot. See Figure 11 for the GFCI outlet and power supply inside of the box. See Figure 12 for the waterproof On/Off switch. How to construct this will not be covered in the User Manual as it is a one-time build.

Figure 11 – GFCI and LRS-600-36 in waterproof box (lid off).

Figure 12 – On/Off power switch leading to the power supply.

Once fully assembled, the power supply had a 25 foot extension cable leading off from it, able to be plugged into any wall outlet providing 110VAC. On the other end, it had 4 feet of a 12-gauge, 2-wire cable for plugging into the first tile. See Figure 13 for the full power supply setup.

Figure 13 – Full power supply box, fitted with extension cord and the tiles' power cable.

If more tiles are produced in the future, it is recommended that a new power supply be designed and built by a qualified electrical engineer. Other workarounds, such as altering the code to make the LEDs dimmer or increasing delay times, would also work. However, these are not recommended as long-term, sustainable solutions to power issues.

3 IMPLEMENTATION

For their implementation, the team made a Verification Prototype to test their design and make improvements. However, the team went even further and completed the full manufacturing run of 28 tiles requested by the GSCCC for their event. This is atypical for Senior Projects at Cal Poly and put the team on a much tighter schedule with far less room for error.

3.1 VERIFICATION PROTOTYPE

After fully completing their critical design milestone, the team ordered, constructed, modified, and tested their design to be made into a final product. Several changes had to be made to the Structural Prototype to turn it into the Verification Prototype. This Verification Prototype is essential to ensure a functional end product. Most of the changes implemented because of the Verification Prototype have already been listed in Section 2.1, though the rest that stemmed from the Verification Prototype will be covered here.

The Verification Prototype built off the Structural Prototype that the team made from their CDR. They swapped out the white high-density polyethylene (HDPE) base for a recycled black HDPE sheet to save on cost. The team implemented the new support structure of eight cylindrical rods, inserted a Test Board v2, added LEDs, and wrapped the inside of the base in tin foil. With this, the bones of the Verification Prototype were complete. See Figure 14 for a visual. Reference Figure 15 to see how the springs sit in the supports and see Figure 16 to see a close up of Test Board v2 as implemented in the tile.

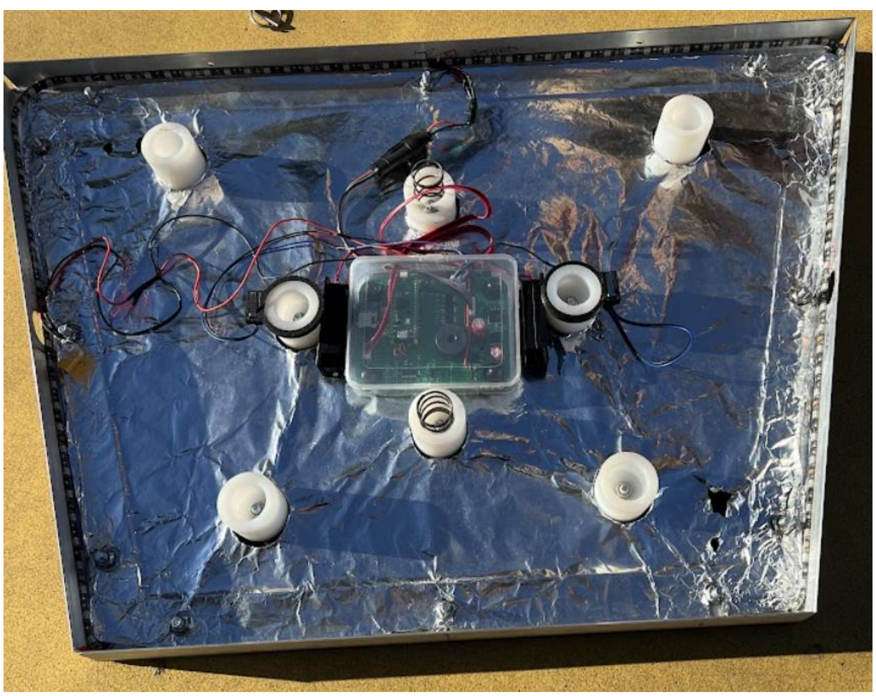


Figure 14 – Inside of the Verification Prototype's base.

Figure 15 – Spring seated in internal support. This method proved to be quite easy and much simpler than the previous method of using bolts to do so.

Figure 16 – Closeup of Test Board v2 as seated in the Verification Prototype. Note how the limit switches and their mounts differ from the version presented in Section 2.1.

Seen in Figure 16, the switches and switch mounts used are different than in the final design. This is because while the switches in the Verification Prototype worked well, they went out of stock before the final production order could be placed. This drove the change to the smaller limit switches with the rail mounts.

As is seen in the User Manual, the tin foil became optional. While improving the brightness of the LEDs inside of the tile, the tin foil was more effort in an already lengthy process. For this reason, it did not survive past the Verification Prototype. In the future, it may be desired to put tin foil back into the build process. Consult Figure 17 to see the difference that the tin foil makes in terms of brightness.

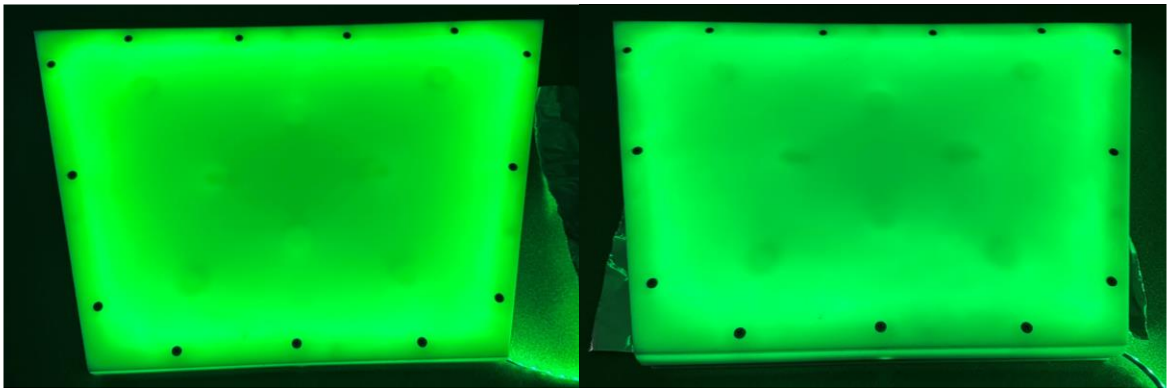


Figure 17 – Tile with no tin foil (left) versus a tile with tin foil (right). The left tile bright at the edges and dark in the center, while the right tile diffuses light much more evenly towards the center.

Test Board v2 was held into the tile by Velcro. The Velcro patch on the base of the tile is seen in Figure 18. This proved to be an effective fixation method, as it was desired to be able to take out the electronics if the tiles started gathering excess water. The speakers also ended up being Velcroed onto the electronics enclosure because of its ease.

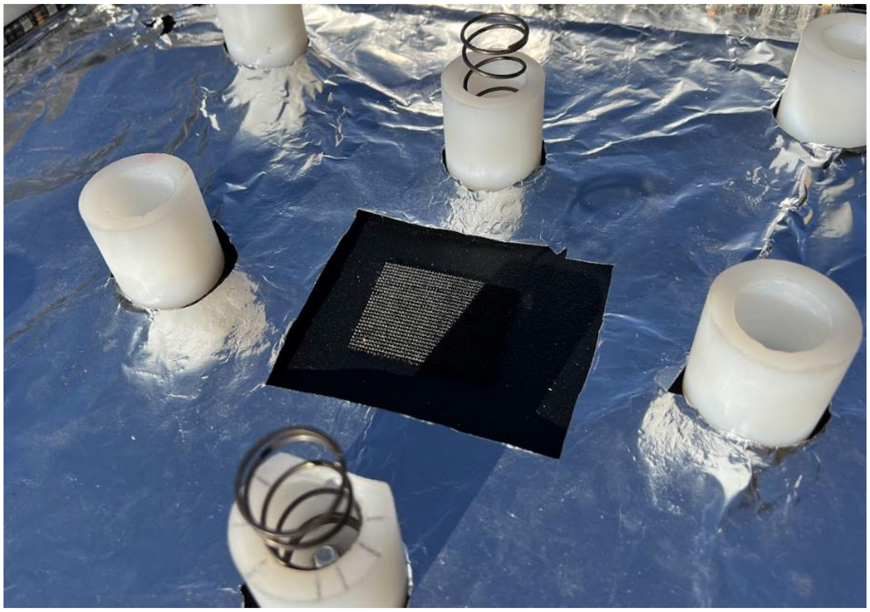


Figure 18 – Verification Prototype with Test Board v2 removed. As seen in the center, a small Velcro patch is adhered to the base of the tile.

Fully assembled, the Verification Prototype looked like Figure 19. Notice that the top sides are 1.5 inches tall. In the final design, these are changed to 1 inch to try and reduce cost, but sacrificed some of the overall aesthetic. Also notice that the grip tape in the image is applied in strips. In the final design, the grip tape is applied in one large sheet spanning the width of the tile.

With the Verification Prototype assembled, it is easier to see the pinholes and slots in the side of the tile, as in Figure 20. In the final design, these slots were changed to 0.5 inch holes for easier manufacturing. You can also see the wire holes and cutouts in the angled sides of the tiles in Figure 21. The wire cutout changed from rectangular to semi-circular in the final design because assembly line processes were not being used, and a hole saw is much quicker than a jigsaw or other setup that could have been used.

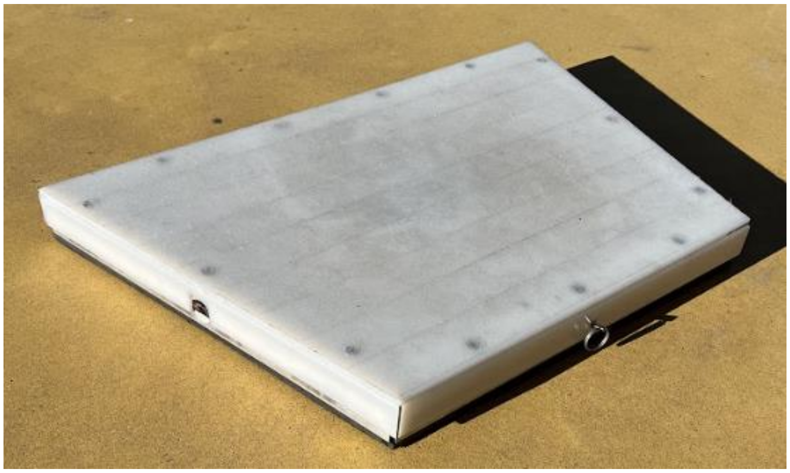


Figure 19 – Fully assembled Verification Prototype. Take note of the large sides and the grip tape strips.

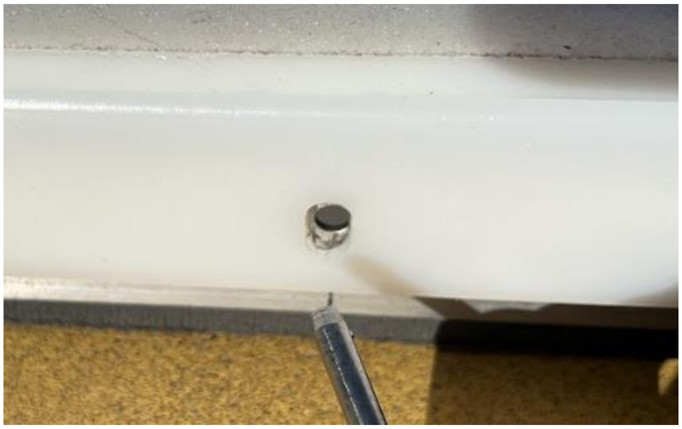


Figure 20 – Pinhole in bottom siding viewed through slot in top siding. In the final design, this slot was changed to a 0.5 inch hole for ease of manufacturing.

Figure 21 – Wire hole and cutout on the angled side of the Verification Prototype. In the final design, the rectangular cutout was updated to a semi-circular hole for ease of manufacturing.

The last service that the Verification Prototype provided the team was a measure of its structural integrity. The team heavily abused the Verification Prototype, doing all kinds of jumps on the tile's surface to try and break it. No one on the team successfully administered any sort of damage, with the heaviest person weighing around 220 pounds. This makes sense, as the tile was designed to support a 300-pound person fully jumping onto the tile. See Figure 22 for the tile being jumped on.

Figure 22 – Tile being administered a hefty blow from a 180 lbm participant jumping.

Note that in all observed use of the tiles, older and heavier participants usually chose to walk slowly over the tiles instead of jumping. Kids were the ones that truly tried to test the structural integrity of the tiles but, due to their smaller mass, couldn't cause any issues. With all this in mind, the team was ready to begin the manufacturing of the full run of event tiles.

3.2 PROCUREMENT

The team's approach to procurement was to pick reputable vendors or websites that offered quality materials with little shipping delay. This guarantees little hassle when another design or manufacturing cycle is warranted, providing the consumer with few headaches due to part procurement. Only 6 vendors were used. All final procurement resources can be found in the final **Bill of Materials in Appendix A.**

Most structural materials are from McMaster-Carr, a huge material vendor with several locations around the United States, offering quick turnaround times on orders. McMaster-Carr is well known and offers pick-up for materials if desired, which is advantageous to save on shipping cost in some cases. For the final order of 28 tiles, all except one item from McMaster arrived in three days, with the last item arriving within a week. A few miscellaneous structural materials were ordered from Amazon or Home Depot, such as stakes or rubber cement.

For the circuit boards inside of the tiles, all the electronic components were ordered from DigiKey, while the circuit boards themselves were from JLCPCB. DigiKey is one of the most well-known electronic component distributors in the US. They have a very wide selection of products necessary for circuit board development and claim to have decent shipping times. However, in the team's experience, these shipping times could often be delayed by up to twice the expected time, though this may be due to San Luis Obispo's hard to reach location for long-distance ground shipping. JLCPCB is a common printed circuit board (PCB) manufacturer and offers cheap and quick PCBs.

Adafruit was used in sourcing the speakers, and Amazon and Home Depot were used for the LEDs, wiring, and miscellaneous project items such as the enclosure that houses the PCB, glands, tin foil, etc. All in all, the team believes that though they wish it was possible to just order from 2 or 3 sources, they have done the best in their ability to choose stress and hassle-free procurement options for the project.

It is of note that the team exceeded the sponsor's original budget of $5000, spending around $6500 of their money on final materials and equipment. Please reference the Bill of Materials for component costs and reference **Appendix D – Final Project Budget** for where the money we spent is specifically tabulated.

3.3 STRUCTURAL MANUFACTURING

The team tried to make the manufacturing processes as easy as possible (within the capabilities of the members). With this, they have created a manufacturing guide to be used in the next manufacturing run of the project. The next person/persons to take on this project should use the experience gained by the group to build off and improve the manufacturing process. Please reference the attached Manufacturing Guide within Section 1 of the User Manual. Because there is a separate Manufacturing Guide, only important details and process improvements will be covered here. Figure 23 helps showcase just how many parts and steps were required to produce 28 tiles.

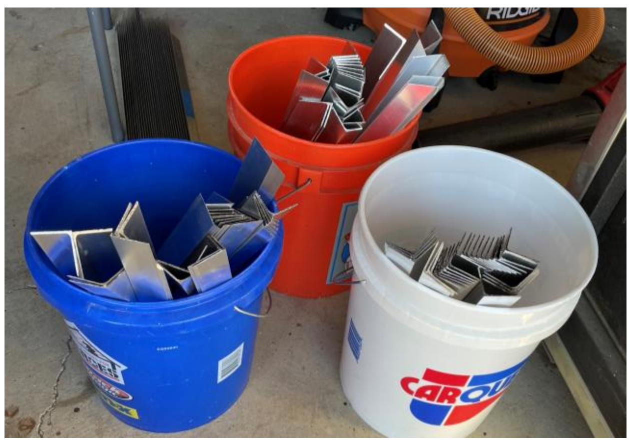


Figure 23 - Cut and categorized aluminum 90 degree angles waiting to have holes marked and drilled.

In manufacturing the tiles, the team recommends outsourcing the water-jetting of the top and bottom surfaces of the tile. This is the preferred method of manufacturing for mass producing tiles out of large sheets of plastic. The water jet used by the team can be seen in Figure 24.

Figure 24 - Water jet cutting tiles.

All other processes are performed using common manufacturing methods typically available in any workshop. As the Girl Scouts of California's Central Coast have their own land manager (Matthew Meadows) with a workshop, the team is assuming that Matthew and his team will be able to perform the rest of these processes.

After the CDR, the team implemented marking jigs to help in constructing the top and bottom angle siding. This was a massive improvement from manually measuring lengths and angles in terms of effort required and helped with repeatability, which is essential when making large quantities of parts. See Figure 25 through Figure 28 for images on using the marking jigs as templates.

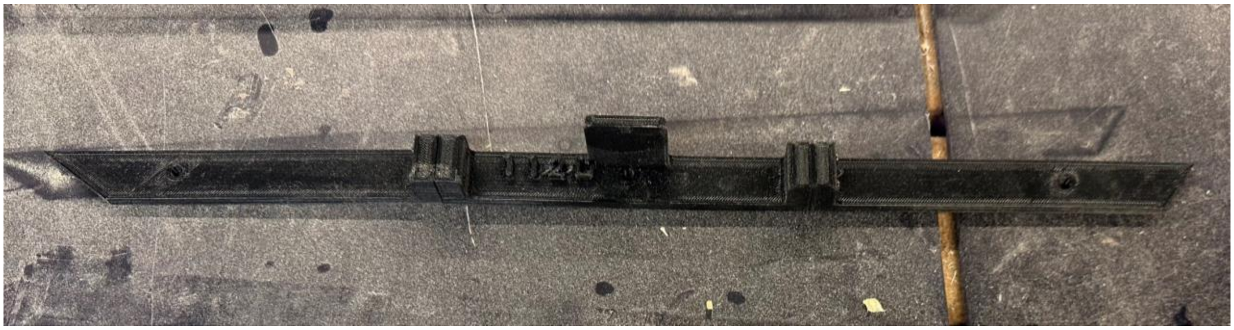


Figure 25 – Example marking jig, used for marking, punching, and scribing lengths, angles, holes and slots in the 90 degree angles.

Figure 26 – Using a marking jig to mark the length and angle of one end of the plastic angle siding.

Figure 27 – Using a marking jig to center punch a hole.

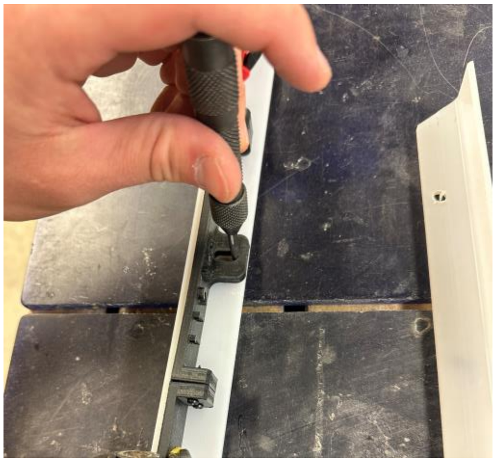


Figure 28 – Using a marking jig to centerpunch the edges of a slot.

3.4 STRUCTURAL ASSEMBLY

Once all custom components have been manufactured, assembly can begin. Again, the User Manual steps through all assembly processes in detail. Only important improvements that could further help the assembly process will be noted here.

One of the hardest parts of assembly was the fact that the locknuts required significant torque to apply. In the future, the team is recommending that the locknuts be swapped for regular nuts and that lock washers are used instead. Decreasing the length of the screw used in securing the top and bottom surfaces to the angles would also make this easier.

The wire slots used in the angled sides of the tiles should also be changed. The bottom's cutout should be rectangular and be open at the top of the tile. The top's cutout should be a larger semi-circle. Both of these processes should be easier than the methods the team used in drilling both the top and bottom cutouts.

Lastly, as previously mentioned, the team excluded tin foil from the assembly process. While it does improve the tile's brightness, it was too much work to be done at the last minute in the project. In the future, it is recommended that tin foil is added back to the base of the tiles.

To aid in assembly and help raise funds for the project, a Girl Scout assembly day was held. See Figure 29 and Figure 30 for pictures from the event. The event was fun and gave the team valuable feedback on the structural assembly of their tiles. This informed them that the lock nuts and bolts were too hard to install for children, the locating of the side supports was difficult, and that an in-depth instruction manual was needed to put together the tiles.

The team assisted with tile construction during the event (seen in Figure 31) and enjoyed working with the Girl Scouts and teaching them about engineering during the build process. It was a successful day and valuable lessons were learned about how to make the event even more accessible for children in the future.

Figure 29 – The Girl Scouts hard at work during the assembly day.


Figure 30 – Project advisor Professor Sarah Harding helping Girl Scouts assemble a tile frame.


Figure 31 – The tiles constructed from the Girl Scout assembly day

3.5 SOFTWARE & ELECTRONICS

It is highly recommended to have the PCBs professionally assembled. The presence of toxic and carcinogenic chemicals required during the fabrication process makes this an unsuitable task for children under the age of 12. It is also a very time-consuming process. If creating small quantities, self-making the PCBs is possible, but in large manufacturing runs it becomes very difficult and time-consuming.

Assembling the electronics enclosures takes less experience, though still takes some practice to get right, not to mention the entry cost for buying all the tools, such as the soldering iron, hot plate, heat gun, and more. If possible, recruit someone with experience or outsource this work to someone who knows what they are doing. Section 3 of the User Manual has the information required for a technical person to complete manufacturing of the PCBs and enclosures. See Figure 32 for a finished electronics enclosure.

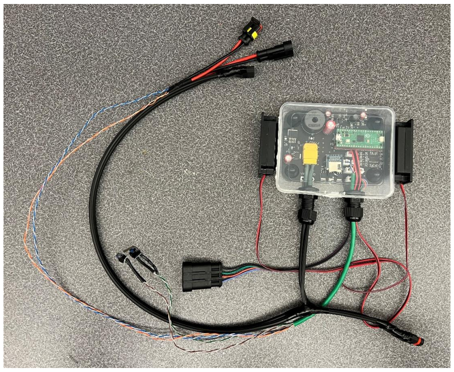


Figure 32 – Finished electronics enclosure with final board and wiring completed.

The sections listed below are more about the software of the tiles instead of the construction of the electronics as the User Manual covers the manufacturing and assembly steps. As so, Table 2 below contains important pin connections to the Raspberry Pi Pico for use in coding.

Table 2 – Pin functions of the Raspberry Pi Pico connected to the Final Board.

Category	Specific Use	Pin Label
Speaker Driving	Left Channel	GP16
	Right Channel	GP17
LED Driving	Red	GP27
	Green	GP26
	Blue	GP28
UART Backwards	Transmitting	GP8
	Receiving	GP9
UART Forwards	Transmitting	GP12
	Receiving	GP13
SD Card SPI (Serial Peripheral Interface)	Clock	GP2
	MOSI	GP3
	MISO	GP4
	CS	GP5
Limit Switch	Reading	GP0

3.5.1 STATE MACHINE

The tiles use a finite state machine (FSM) coding structure. Every tile is coded identically, but it enters different sections of code based on the information available to it. That is, if the tile has a "controller.txt" file on its SD card, it will be the head tile. Otherwise, the tile is a follower and does not start the song. See Figure 33 for a visual depiction of the FSM. After the tiles decide whether they are a controller file or not, they will follow the standard progression below:

Controller Tile

1. If stepped on, send out the song information.
2. Wait for followers' response.
3. Start glowing, indicating it can be stepped on.
4. Once stepped on, play a sound and send info to the next tile.
5. Fade out the LEDs after sound is done.
6. Delay 7 seconds to offset users.
7. Repeat at step 4.

Follower Tile

1. If a note.txt file is present, set tile notes.
2. Wait for a message.
3. If the message is encoding song info, take tile info and pass on the message.
4. If the message is random 0's, do nothing.
5. If the message is saying the previous tile was stepped on, start expecting a step.
6. When stepped on, play a sound and send info to the next tile.
7. Fade out the LEDs.
8. Repeat at step 2.

The tiles need 5 basic Circuit Python files in order to function correctly. These files are listed below and can be found in **Appendix E – Code**, or in the Software Package provided to the Sponsor under the "Pico Coding" folder.

- **ledstrip.py –** contains functions that help control the LEDs.
- **speaker.py –** contains functions that help operate the speakers.
- **dataFunctions.py –** contains functions that help transmit data back and forth between tiles as well as initially read the 'controller.txt' file.
- **Lookup.py –** contains information to transmit readable text into encoded numbers.
- **main.py –** uses all other files to create a state machine and fully run the tiles.

It is important to note that in addition to the tile's regular Circuit Python files stored on the on-board memory, there are sounds stored on the SD card. In addition to these sounds, the tiles can also have a "note.txt" or "controller.txt" file on the SD card. The "note.txt" may be present on any tile and does not need to be altered by the user. This is created or overwritten every time the tile receives new song information through the UART communications. This makes it so the tiles do not need to have the song information sent out every time they get powered on.

The "controller.txt" file is more complicated and, besides the first line, is not meant to be read by humans. This text file contains all song information for the desired song and must only be placed on the SD card of the first tile. Multiple "controller.txt" files in the path will cause issues.

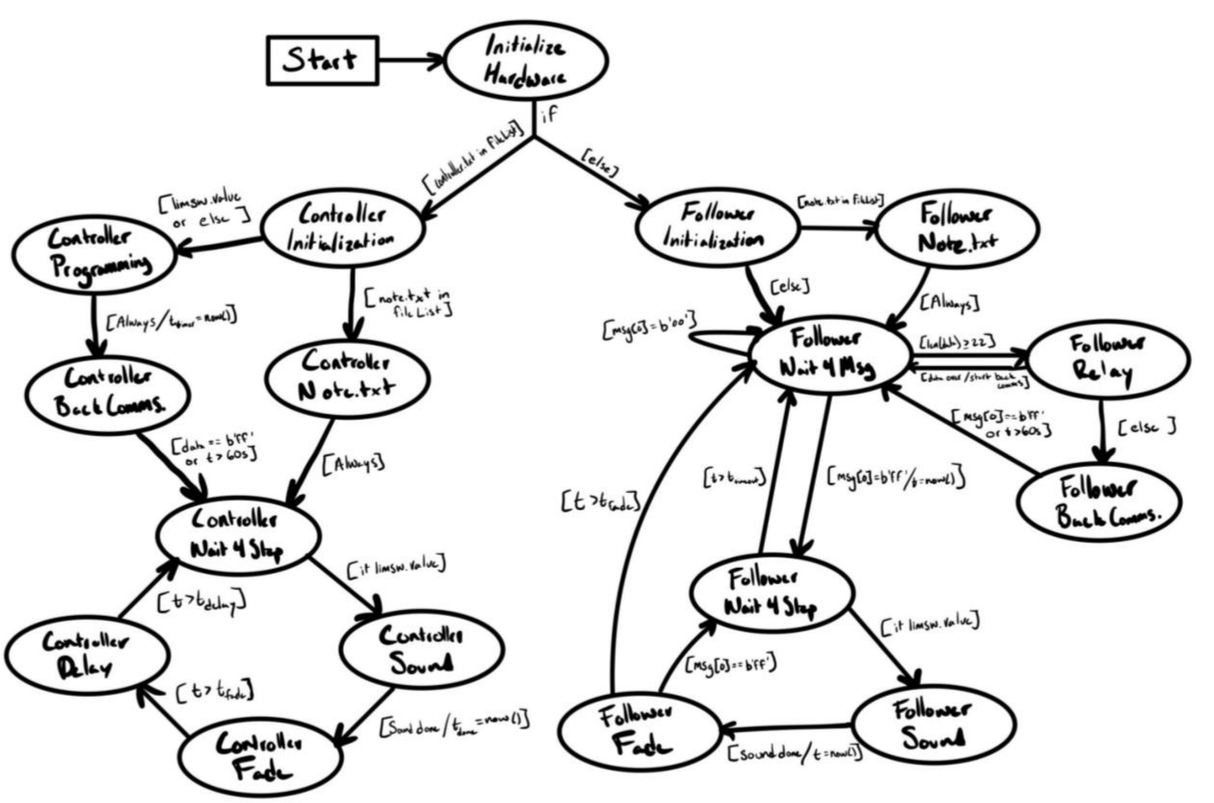


Figure 33 – Finite State Machine of the tile's coding logic. All tile code runs off of the same Circuit Python files.

An example "controller.txt" file's structure is broken down in Figure 34. A "controller.txt" file is generated using the graphical user interface (GUI) covered in the next section. Bytes are compact packages of information that microcontrollers can easily read and use for encoded information. The "Lookup.py" file is the file that tells these bytes what to translate to/from in English. See the User Manual for how the "Lookup.py" file and "sounds" folder can be modified to increase tile capability.

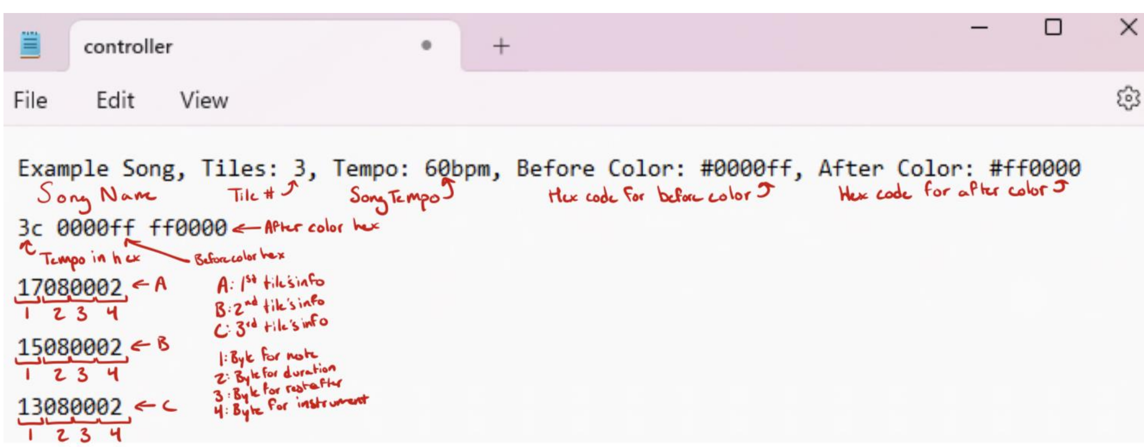


Figure 34 – Breakdown of a "controller.txt" file. The first line is readable information for the user to understand the file's general information. The second line allows every tile to get the song tempo and the colors that they are supposed to light up. The following lines correspond to individual tiles and encode the instrument and duration of every note in the song sequentially. A corresponds to the controller tile, B to the 2nd, C to the 3rd, and so on.

The reason for the structure of the "controller.txt" file is due to the means of communication of the tiles. UART is able to pass information between adjacent tiles but cannot travel all the way to the last tile instantly. With this file structure, each tile can pick off its information and send the remainder of the file to the next tile down the line until there is no information left. While better methods exist such as I2C or CAN, this method allows for easy and quick communication between the tiles and can be infinitely modular without concerns of noise pickup because of the short wire span between tiles.

3.5.2 GRAPHICAL USER INTERFACE

As mentioned previously, the graphical user interface (GUI) is how "controller.txt" files are created. The GUI, seen in Figure 35**,** takes inputs such as a name, the number of tiles, colors, and each tile's musical characteristics. It is very useful to take this information in text form and translate it into numbers, deciphered with a key file called 'Lookup.py', for compact storage and each transfer between tiles.

The GUI is by no means a professionally made construct, as the team tried their best to learn how to make and use one specifically for this project. It functions and serves the purpose it was made for, though lacks some features that would make the user's quality of life much better. These features include dynamic updating of GUI fields, being able to load a previously made "controller.txt" file, and better overall aesthetics. It is recommended that the GUI be redone as part of a future project.

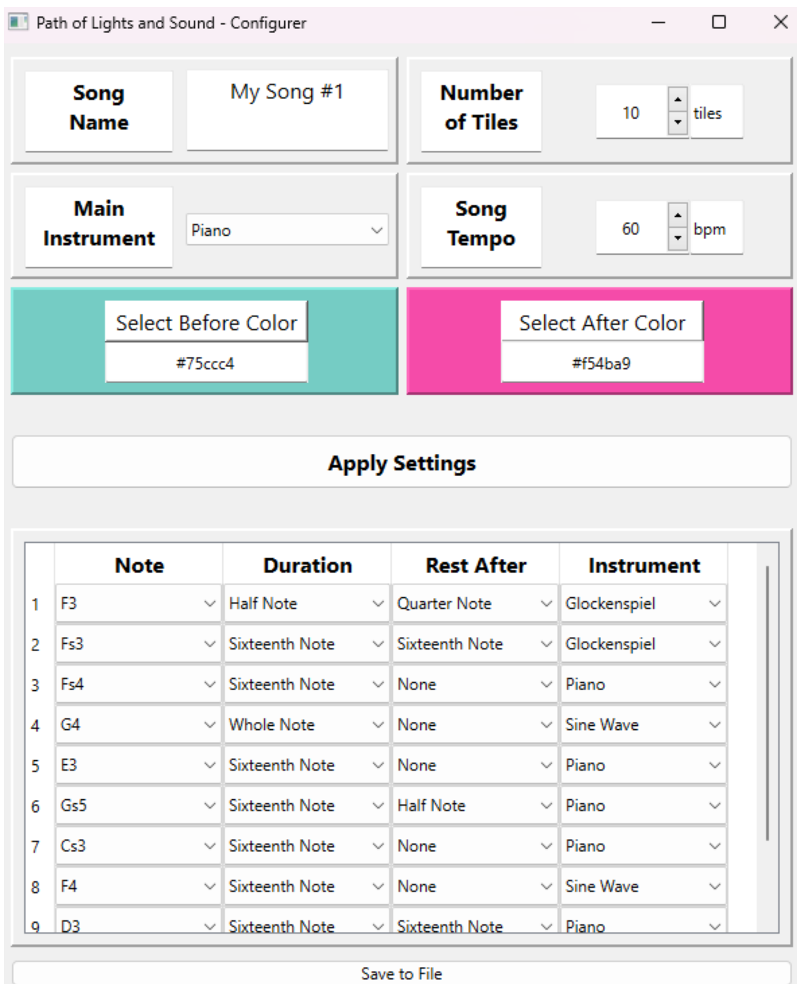


Figure 35 – Graphical user interface used for programming the tiles to play a new song.

A guide on using the GUI is in Section 4 of the User Manual and details every step of the process of reprogramming the tiles. For redundancy, the tile's startup process for initiating reprogramming looks as such after the "controller.txt" is put on the first tile's SD card and the tiles are fully set up:

1. If the first tile is being stepped on as power is turned on, it sends out the song information to all other tiles. The information is saved so with the same path configuration, the tiles will not need to send this information again. If the tile configuration does change, just step on the tile during startup again. You will know that reprogramming starts if the color green starts progressing down the path.
2. After programming, the first tile starts glowing one color, green for example. Once stepped on, the tile turns a different color, such as red, plays a note, and sends information to the next tile. The first tile has a 7 second cooldown period where it can't be stepped on to stagger users.
3. With the sent information, the next tile grows in brightness until the correct timing for it to be stepped on. Regular tiles don't have a cooldown period but must be stepped on in order. If not stepped on within 5 seconds, the tile stops glowing and can't be stepped on until the previous tile comes around again.
4. This chain continues until the path is complete, or the user has decided to step off the tiles and the tile times out with its 5 second buffer.

Circuit Python is a great introductory coding language and if the files 'main.py' or any of its associated files are inspected, a reasonable amount of understanding can be gained. Google searches are very valuable in coding and can provide much information on the functions being used. It is not expected that the end user will read this code, though if it needs to be changed for whatever reason, used as a teaching lesson, or updated in a future project it should be straightforward to do so.

3.6 CHALLENGES

After completing the tile's manufacturing, assembly, and coding, the team used them to try and fix any bugs present. The tiles worked fairly well except for one bug that kept showing up, happening seemingly at random. The bug was that some tiles would change color once stepped on, not play a sound, and be stuck in this lit-up state not being able to be used. The bug is fixed by turning the power supply off for 5 to 10 seconds and then turning it back on.

The team tried their best to put stops in the code to avoid this error, but none of their attempts were successful. It specifically happens in transitioning from waiting for a step to playing a sound. They believe it may be an issue either providing power to the speakers or in loading the sound file from the SD card. However, if either of these were the case then the team is unsure how resetting the tiles would have fixed the issues.

If the project is reconsidered in the future, this is the largest programming issue that the team urges to be fixed. By the time they ran into the bug, they were out of time and could not properly diagnose and address the issue. Though, since toggling the power is a solution, they do not consider it to be a fatal design flaw.

4 DESIGN VERIFICATION

With the design completed and full prototypes made, the team needed to verify that the design met everything it set out to do. Design objectives were cataloged into specifications in the previous design reviews. Determination of whether the design met these specifications was based on tests that the team designed to address every aspect of importance or simply based on inspection and observation.

4.1 SPECIFICATIONS

Some specifications were simply met by observation or the construct of the design. These tests are cataloged with their results in Table 3. Other specifications required testing to determine if they were met or not. These specifications are listed in Table 4 along with their corresponding test. These tests were specifically designed to determine the design's adherence to the specification. The testing results are presented in the next section.

Table 3 – Specifications met by observations, not testing.

Specification	Criteria	Result	Pass/Fail
Cost	< $300	~$235	Pass
NFPA 70	Standard Met	Undetermined*	Fail
IP Rating	IPX4	IPX2**	Fail
Material Standards	Frame made of certified materials	Aluminum - Yes Plastic - No	Fail
Weight	< 25 lbf	11.95 lbf	Pass
Size (Volume)	25 tiles in < 35 ft^3	25 tiles in 14.58 ft^3	Pass

* While not formally analyzed, the team has included a GFCI protected power source as well as a easy-to-see switch on the power supply. This should keep the event safe from unforeseen issues that could cause dangerous situations.

** IP68 glands, waterproof connectors, and waterproof LEDs were used, though the enclosure itself only had a sealant rim. Due to enclosure warpage creating gaps, this is likely only safe from water droplets, not splashing or pooling water.

Though not all these specifications were met, it is important to consider that the team was working with limited budget and time. Some specifications could not be properly considered within these limitations. In the future, it would be ideal to revisit these and try to address each unmet specification. The team does consider the unmet specifications to be far from catastrophic.

Table 4 – Planned testing and the specifications addressed.

Specification	Planned Test	Date Completed
Coefficient of Friction	Angled Slip Test	11/6
Impact Protection	Drop and Jump Test	11/6
Force Required to Move	Kicking/Budge Test	11/6
Manufacturing Time	Manufacturing and Assembly Test	11/17
Decibel Reading	Sound/Decibels Test	11/15
Maintenance Time	Assembly and Disassembly Test	11/14
Consumer Focus Group	Consumer Focus Group	11/14
User Test Group	User Test Group*	Not Completed
Portability Storability Rating	User Test Group*	Not Completed
Programming Rating	User Test Group*	Not Completed

*Time constraints stemming from the extensive manufacturing that the team completed resulted in a lack of time to complete the User Test Group. These specifications are assumed to be unmet because of this. The team is confident that if the test were completed, the goal would be met.

For clarity, each test has a brief description and its results summarized in **Appendix F – Design Verification Plan & Report.** The full procedures and results are in **Appendix G – Test Procedures**. Appendix E is meant to generally describe the test and catalog the progress the team has made in testing while Appendix F is meant more specifically to control each individual test, its procedures, and its pass/fail criteria.

4.2 TESTING AND RESULTS

Unfortunately, due to time constraints from the manufacturing of all 28 tiles, the team was not able to conduct the User Test Group. However, the other 7 tests were completed to the best of the team's ability, and a summary can be found in Table 5 or in **Appendix F**. Mainly significant testing outcomes will be discussed here. For full test information, consult **Appendix G**.

Table 5 – Testing results.

Test Name	Purpose	Criteria	Result	Pass/Fail
Friction Test	User traction	$\mu \geq 0.5$	1.20	Pass
Impact Protection Test	Early detection of long-term damage	No detrimental damage	Light scratching due to internal supports	Pass
Kick Test	How well the tile is fixated to the ground	25 "kicks" move tile < 1in	Staked down: 3/8" (25) No stakes: 5/4" (1)	Staked down: Pass No stakes: Fail
Manufacturing Time	Manufacturability for future reproduction	1 tile made in ≤ 8 hours	w/ electronics: 10.58 hrs w/o electronics: 4.25 hrs	w/ elec.: Fail w/o elec.: Pass
Decibel Test	Loudness capability of the tile	Sound produced ≥ 80dB	83.22 ± 4.71 dB	Pass
Maintenance Time	Serviceability of the tile for upkeep	Dis/Re-assembly time ≤ 10 min	33 minutes and 42 seconds	Fail
Consumer Feedback	Tile's appeal in terms of public opinion	Rating ≥ 4 stars	4.55	Pass

First and foremost, the tile passed the kick test when staked down but not when only on its rubber feet. The kick test was set up to gauge how difficult the tile was to move when set up and impacted with a repeatable input. See Figure 36 for a visual on how the kick test was set up. When staked into grass, the tile did an excellent job of staying put. However, on concrete the tile moved visibly after only 1 hit, as measured in Figure 37. Because of this, the team recommends always staking down the tiles. If necessary, please only use the tiles on hard surfaces when the event is intended for mature audiences. Otherwise, the rubber pads may need to be improved for sustained use on hard floors.

Figure 36 – Kick test setup. A weighted bucket was repeatedly swung into the side of the tile.

Figure 37 – Kick test result from using the tile on concrete with rubber feet. The tile moved 5/4" after 1 kick.

It is also of note that the manufacturing and maintenance time tests were both failed in some capacity. The team was far from meeting the criteria they set as passing. While it is important to remember that the criteria were semi-arbitrarily set, they still hold significance in how hard or easy something may be to complete. The manufacturing of the tiles can still be completed in under a day if neglecting the electronics, but a tile without electronics is not inherently useful. Suggestions stemming from this failure are listed in Section 5, Discussion and Recommendations.

Lastly, the decibel test had uncertainty analysis done on it to reach the 83.22 ± 4.71 dB reading. In fact, this reading is solely for the "Sine Wave" instrument that comes with the tile's sound library. The "Piano" library resulted in a reading of 71.12 ± 3.65 dB. The difference in uncertainty in the two values stems from statistical uncertainty, coming from changes in individual readings. The other two types of uncertainty are resolution (how fine the measurement device can read) and calibration. Calibration uncertainty could not be officially done on this test, but it was estimated from a study by Murphy and King.[1]

This study conducted tests on the accuracy of phone apps in measuring decibels. The iPhone used in the team's test was then assumed to be an average iPhone as Murphy and King's study suggested. The Decibel X iPhone app was used in measuring the sound readings. A view of the app's interface is in Figure 38. In the end, this test was passed as the speakers were capable of generating greater than 80 dB. If the "Piano" library is desired to produce a similar volume as the "Sine Wave" library, the gain on the sound files can be adjusted digitally using sound processing software such as Audacity.

Figure 38 – Decibel X app's interface. Note that the "Max" reading was used and reset for every trial of the test.

4.3 COMMENTS

The team is happy that 7 out of 16 specifications were met, with an additional 2 being conditionally satisfied. While at first glance this seems like an abysmal outcome, when considering the increase in project scope, adding much more work than originally intended, one must understand the sacrifices made to complete the project.

This goes hand in hand with the medium-scale production of the tiles. When doing small productions, time and care must be taken to make a perfect product. When doing large productions, methods are used to ensure the quality of every product. Medium-scale productions can neither take the time nor use the methods that come with small and large scales, especially when packed into one quarter of school, while also handling full workloads in other classes.

If done again, the team recommends addressing the IP rating of the electronics, the manufacturing/maintenance time, and the force required to move specifications. Improving these will greatly improve the future of the design.

5 DISCUSSION & RECOMMENDATIONS

5.1 SENIOR PROJECT EXPO

Senior Project Exposition was a great success for the team. Many people showed a great interest in the project and enjoyed using the three tiles that were at the exhibit. Professors, students, recruiters, and children alike all seemed to be impressed with the feat that the team accomplished. See Figure 39 and Figure 40 for photos from the expo.

Figure 39 - A very good boy enjoying the tiles.

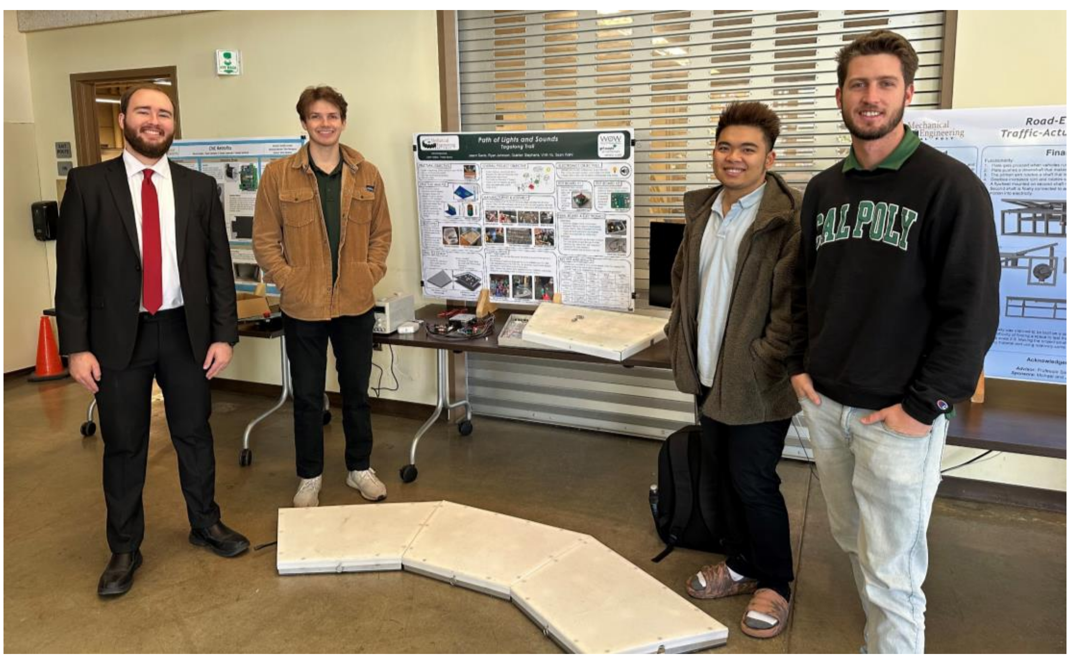


Figure 40 - The team gathered around their booth at expo.

What was especially interesting for the team was to see the users' habits in using the tiles. At first, many people were skeptical and tried to activate the tiles with only one foot, not supporting their weight on the tile. After this initial investigation, users were comfortable in walking on the tiles. Adult users seemed to not want to abuse the tiles, comfortably walking across the tiles. However, kids seemed to want to run and jump and do everything with the tiles. They were eager to go simultaneously and impatiently waited for the tile delay to reset.

The team was happy to see this feedback and felt the design considerations they made were well chosen. The team was especially happy that they put in a delay on the first tile. While it would be fun for kids to rapidly run across the tiles, the 7 second delay on the first tile preserves the integrity of the song. It also limits the power usage of the tiles at any given time.

5.2 EVENT IMPLEMENTATION

The team set up and tested the tiles several times after fully constructing all 28. Though they had a bug (covered in Section 3.6), they were more than suitable and ready for the event. On the first day of the event, the tiles were moved from one location on the Girl Scout's camp to another, requiring them to be re-setup.

After being moved, the tiles were not behaving correctly. The first two tiles lit up as desired but after that, none of the tiles were able to function correctly. Some tiles even lit up all white despite not being told to do anything. See Figure 41 for reference. Upon initial investigation, the team could not help the Girl Scouts get the tiles working correctly for the first event weekend.

Figure 41 – Tiles experiencing the issue at the event. The start of the path is at the top of the image. Note the fully white tile.

The tiles were brought back up to the Cal Poly Senior Project team and they were able to do a real investigation of the tile's issues. Of all 29 tiles in their possession, 19 were found to be fully functional. However, the other 10 had broken microcontrollers. The team tested replacing the Picos and found that this fixed the tiles. Three running hypotheses were created for as to what happened:

1. Power surge from the generator
2. Water damage from humidity in the area
3. Plugging in the tiles while the power supply was on.

While all perfectly plausible, the team could not definitively say what caused the issue. However, they lean towards the third cause. In the first case, more tiles would seem to be damaged, especially near the beginning of the path. For the second case, more components would also likely have water damage besides the Picos. To safeguard the use of the tiles and avoid all causes, they recommended that the tiles only be used indoors for the remainder of the event. They also recommended that the set-up instructions be rigorously followed to avoid power spikes, as well as only hooking up the tiles to wall outlets.

With these new usage guidelines and 10 more Picos ordered, they hope that the rest of the event goes without a hitch. While unfortunate that this problem arose in the first place, they believe that the tiles will work properly when taken care of. To expand usage in the future, the tiles may need extensive testing outdoors to open the possibility of safe, long-term outdoor use.

5.3 DISCUSSION

The main takeaway that the team was left with is that it was significantly more difficult to manufacture this project than initially planned for. If the team were to continue this project, they would investigate ways to make manufacturing and assembly easier. Improving the water-resistance of the electronics and making the set-up process simpler and quicker would also be a priority.

One major change that the team would make if they were to repeat this project is that they would split the team earlier. Previously, the team split into mechanical and electrical sub-teams after the second quarter of the project. The team feels that this caused the electronics design to be rushed. While the electronics still ended up working, more time to assemble, test and debug the electronics and software would have been ideal.

Another change the team would make is to follow a different schedule to the one prescribed by the class. The main challenge of this project was testing and manufacturing, so this should have started significantly earlier than it did. Coming together to manufacture parts, instead of sticking to the split groups would have also helped relieve some monotony of manufacturing. This is all to say that the team agreed to do way too much work in comparison to what they were asked to do at the outset of the project.

5.4 RECOMMENDATIONS AND NEXT STEPS

Though the project was successful, there are a handful of items that must be reconsidered if these are ever to be manufactured again. This section covers all future design edits that should be made in such a case.

If the electronics go through another redesign, the team recommends using more surface mount components instead of through hole components. This would improve electronics assembly time and ease of assembly. The team would also recommend searching for a single connector that can handle all necessary connections between tiles. This would improve the setup process and reduce wasted time.

The enclosure also needs to be addressed in the electronics redesign as well. The team failed to make their enclosure fully waterproof, and doing so could guarantee outdoor use of the tiles. Due to budget and time constraints, the team used cheap, malformed, plastic enclosures. What really should have been used are slightly more expensive but fully sealable containers that can guarantee waterproofing. A conformal coating applied directly to the PCB would also work.

In the event that more tiles are manufactured, the team recommends automating as much as possible, including the machining of the internal supports, PCB assembly, and connector assembly, as this is where the majority of time was spent on this project.

If more tiles are added to a single string, the team recommends either upgrading the power supply or feeding power from both ends of the string. The current power supply is safe up to 30 tiles, but after that, a second or a larger power supply will be necessary.

The team also strongly recommends introducing an uninterruptable power supply (UPS) before the power supply box to prevent power conditioning issues or power surges. These are relatively inexpensive and are a good method to protect the electronics. One example would be the CyberPower CST1500SUC, but any unit with similar capabilities should work.

To improve the re-programming experience as a user, the team advises that an improved GUI for programming the tiles may assist in ease of use. Another potential step to improve the programming process would be to improve the inter-tile communication to include the necessary sound files.

The group was happy with the capabilities of the tile's structure and felt that the only major issues were that the clearance between the top assembly and the bottom assembly was too tight, and the screw at the center of the short side of the tile interfered with the pin in the side of the tile. In any future design of the tile, this clearance should be doubled to ensure free movement of the top surface when stepped on, and easy return to its original position when stepped off. The hole location for the interfering screw should be removed, and the two other screw locations should simply be moved toward the center to compensate. On a related note, each of the sides can have one of the screw hole locations removed and the other screw hole locations shifted to improve manufacturing and assembly times.

If more tiles are to be made, the group recommends looking into making the top and bottom assemblies into 1 part each with a process like injection molding or casting. The production cost would increase greatly, but the labor cost for things like assembly and manufacturing time would decrease and offset this increase in cost.

Another change that should be implemented to the structure that would help with set-up and tear-down time would be to make the holes in the side supports of the tiles, shown in Figure 42, meant for wires to pass through, into open square slots. This would make it so that the connectors would not need to be passed through holes, but the wires could instead be simply set into place before putting the lid back on the tile.

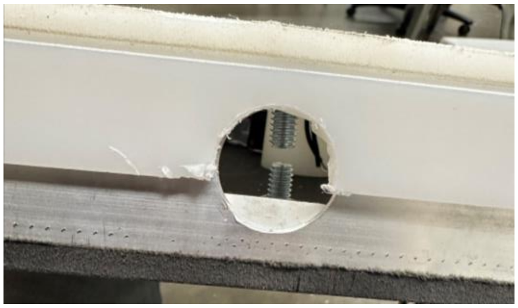


Figure 42 - Shows both the oversized bolts and the hole in the side supports for the wires to pass through.

A perceived issue that the team could not solve was that the supports created shadows on the surface of the tile when it was compressed, seen in Figure 43. Utilizing decals of some kind to control the shape of the shadows is recommended for future iterations of this project. Similarly, to limit the noticeability of the shadow created by the electronics enclosure, the group recommends covering this in a reflective material, like aluminum foil. Reflective material is also recommended to be placed on the base of the tile, to improve even light distribution.

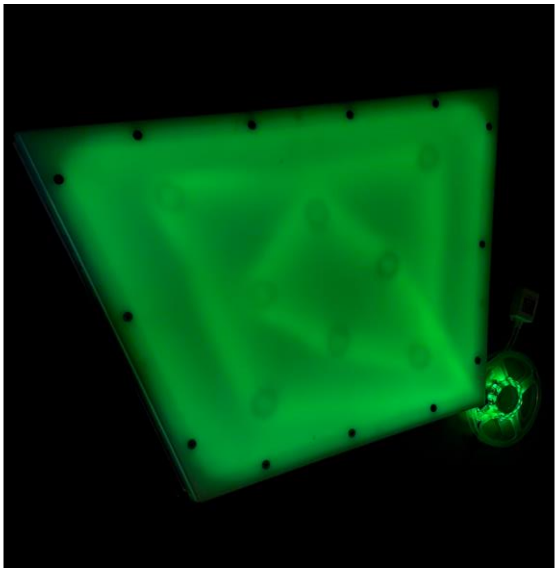


Figure 43 - Tile with shadows created by internal supports.

The length of bolts chosen turned out to be much longer than what was needed, which can be seen in Figure 42. The group recommends switching from 1” to 5/8” long bolts. The nylon locking nuts also proved to be much more difficult to install than expected. The group recommends switching to locking washers with a non-locking nut as well.

6 CONCLUSION

The Path of Lights and Sounds project, completed by The Tagalong Trail Team for their Senior Project at Cal Poly San Luis Obispo, resulted in 28 portable, intercommunicating, walkable tiles capable of producing lights and playing notes that, when programmed and lined up, produced a song as the user walked down the path. The team combined each of their skillsets, all honed by the Cal Poly Engineering Program, in order to ideate, design, manufacture, test, and report their findings for this project. For a fun video of the path working, check out this video.[2]

Senior Design taught the team how to properly research, perform ideation techniques, and iterate through early design ideas. The team utilized skills in finite element analysis (FEA), computer aided design (CAD), and strength and stiffness of materials to analyze and evaluate the validity of their early designs. The team iterated through ideas and concepts, utilizing feedback from their instructor and peers, and constructed a design prototype to validate their findings. After several alterations to this prototype, the team felt ready to mass manufacture the design. Utilizing manufacturing and metrology skills learned in their industrial manufacturing engineering (IME), and engineering design courses, the team set to work producing the structures. Simultaneously, team members utilized their knowledge of electronics, board design, and mechatronics learned from classes taken that fulfilled their mechatronics concentration path to design, iterate, and assemble the electronics design for the tiles. After several hundred hours of manufacturing work on both the structure and the electronics, the team produced 28 tiles and began the report.

The process taken by the group to complete this project taught them valuable skills that will be carried on into their careers and all future endeavors. The group learned the value of teamwork, communication, peer feedback, and trusting the process.

The team fell short in their testing. Although the majority of the tests that the team designed were completed, it did not seem as though enough testing was done to confidently approve the tiles. The manufacturing time engulfed most of the time the team had planned to test. If more tests were able to be completed, the team suggests performing the user survey, as well as a waterproofing test to the electronics enclosures.

The Tagalong Trail team of seniors set out to create a path of portable light up tiles, capable of being lined up and playing a song as the path was traversed. This team accomplished that goal, creating and delivering 28 functional tiles to their sponsor several days before the event. A lot was learned throughout this process, and a lot can be improved upon, but the team is proud of what they accomplished. This was a successful senior project.

REFERENCES

[1] Murphy, Enda, and Eoin A. King. "Testing the accuracy of smartphones and sound level meter applications for measuring environmental noise." *Applied Acoustics*, vol. 106, 5 Jan. 2016, pp. 16–22, https://doi.org/10.1016/j.apacoust.2015.12.012.

[2] Wahl, Sean, director. *Working Path of Lights and Sounds*. *YouTube*, YouTube, 8 Dec. 2023, https://www.youtube.com/shorts/u1-BN-DRcQI. Accessed 8 Dec. 2023.

APPENDIX A – BILL OF MATERIALS

* See Bill of Materials excel file for working links.

Table A1 – Structural Bill of Materials.

Assy Level	Part Number	Descriptive Part Name			Qty	Unit Cost	Total Cost	Source	More Info
		Lvl0	*Lvl1*	*Lvl2*					
0	100	Final Tile Assembly			1				
1	110		Top Assembly		1				
2	111			Top Surface	1			custom	waterjet
2	112A			TLong Side	1			custom	miter saw, drill
2	112B			TShort Side	1			custom	miter saw, drill
2	112C			TSide L1	1			custom	miter saw, drill
2	112D			TSide R2	1			custom	miter saw, drill
2	114			Clear Grip Tape*	1			custom	exacto knife
2	115			1/4-20 Zinc Plated Bolts	13	$ 0.13	$ 1.72	McMaster-Carr	90273A542, Pack of 100
2	116			1/4-20 Stainless Steel Locking Nuts	13	$ 0.13	$ 1.67	McMaster-Carr	95462A029, Pack of 100
1	120		Base Assembly		1				
2	121			Bottom Surface	1			custom	waterjet
2	122A			BLong Side	1			custom	miter saw, grinding wheel, drill
2	122B			BShort Side	1			custom	miter saw, grinding wheel, drill
2	122C			BSide L1	1			custom	miter saw, grinding wheel, drill
2	122D			BSide R2	1			custom	miter saw, grinding wheel, drill
2	122E			Internal Supports	8			custom	miter saw, drill
2	122F			Switch Holster	2			custom	3D Printer
2	122G			Switch Holster Set Screw	2	$ 0.09	$ 8.98	Home Depot	Pack of 100
2	123			Rubber Pads*	5			custom	Exacto knife
2	123B			Weldwood Contact Cement	1	$ 0.50	$ 22.98	Home Depot	
2	115			1/4-20 Zinc Plated Bolts	20	$ 0.13	$ 2.65	McMaster-Carr	90273A542, Pack of 100
2	116			1/4-20 Stainless Steel Locking Nuts	20	$ 0.13	$ 2.56	McMaster-Carr	95462A029, Pack of 100
1	130		Springs		2	$ 0.76	$ 1.51	McMaster-Carr	1986K28, Pack of 6
1	140		Ring-Grip Quick-Release Pin		3	$ 2.20	$ 6.60	McMaster-Carr	98404A009
1	150		Stakes		4	$ 0.17	$ 0.68	Amazon	Pack of 100
				Total Parts:	**107**				

Table A2 – Electronics per board Bill of Materials.

Per Board Purchases						
Category	**Component**	**QTY**	**Price per Source**	**Price per Tile**	**Source**	**Important Values**
General Electronics	Digikey Order	0.033	$882.83	$29.13	Digikey	for 25 boards (update as needed)
	JLC PCB order	0.033	$100.34	$3.31	JLCPCB	based on order for 30 boards+stencil
	Stereo Speakers	1	$7.50	$7.50	Adafruit	3W 4ohm 2 stereo set, jst-ph
	LEDs	0.5	$14.99	$7.50	Amazon	12V, 5m, IP65
	microSD Breakout Board	0.1	$8.89	$0.89	Amazon	6pin, SPI, microSD
	microSD Card	0.2	$19.99	$4.00	Amazon	32GB, 5pack, microSD
Misc. Electronic Stuff	Xt60-pw-m board connector	1	$0.88	$0.88	TME	Solder to Board
	Xt60 female to board connector	1	$0.45	$0.45	TME	Connect to Board
	Waterproof xt60 connectors	0.1	$9.99	$1.00	Amazon	1 pair needed
	Waterproof connectors for uart	0.1	$10.52	$1.05	Amazon	1 pair needed
	Waterproof connectors for Led	0.1	$13.99	$1.40	Amazon	1 pair needed
	b4b-ph connector, speaker	1	$0.12	$0.12	See here	jst b4b-ph, straight, 2mm
Enclosure	Boxes	0.083	$14.99	$1.24	Amazon	5"x4"x1.3"
	Internal Standoffs	0.16	$12.99	$2.08	Amazon	0.75"x0.75" adhesive cones
	Velcro	0.125	$7.92	$0.99	Amazon	4"x2" strips
	Cable glands	0.1	$9.99	$1.00	Amazon	PG9

Table A3 – Electronics single purchase Bill of Materials.

Single Time Purchases						
Category	**Component**	**QTY**	**Price per Source**	**Total Cost**	**Source**	**Important Values**
PCB	Solder Paste	1	$9.99	$9.99	Amazon	30g, eutectic
	Solder Wire	2	$11.99	$23.98	Amazon	100g, eutectic
Power Supply	Power Converter	1	$74.15	$74.15	Digikey	36V DC Out, 350W
	Power Supply Enclosure	1	$58.38	$58.38	Home Depot	12"x12"x6" junction box
Wiring	Power Wire	2	$21.98	$43.96	Home Depot	50ft 12/2 LVLLW
	Uart Wire	2	$5.25	$10.50	Home Depot	25ft 16/2 LVLLW
	LED Wire	1	$16.49	$16.49	Amazon	40ft, 4 strand 20 awg
	Heat Shrink	3	$2.50	$7.50	Harbor Freight	5/16", 8ft
	Sealant	5	$7.98	$39.90	Home Depot	5.5oz, white
	Uart/Switches Wire	1	$23.99	$23.99	Amazon	24awg, cat5 cable, 100ft
	Extension Cord	1	$28.99	$28.99	Amazon	25ft outdoor cord
	Additional heat shrink	2	$5.49	$10.98	Amazon	240pcs of 3mmx30mm sections

Table A4 – Electronics DigiKey Bill of Materials.

System	Component	QTY/board	QTY	Manu. Part #	Digi. Part #	Source	Important Values
General	Reverse polarity PMOS1	1	30	DMP6180SK3-13	DMP6180SK3-13DICT-ND	Digikey	60V, 14A, TO-252-3
	LED NMOS1, 2, 3	3	90	DMN3016LK3-13	DMN3016LK3-13DICT-ND	Digikey	30V, 12.4A, TO252
	12V Switching Reg, IC1	1	30	LM2576S-12/NOPB	LM2576S-12/NOPB-ND	Digikey	12V, 3A, Switching
	5V Switching Reg, IC2	1	30	TPS5405DR	296-30579-1-ND	Digikey	5V, 2A, Switching
	PAM 8406 Audio Driver IC3	1	30	PAM8406DR	PAM8406DRDICT-ND	Digikey	5W, 16SOP
	Limit Switches	2	60	D2GW-A13H	39-D2GW-A13H-ND	Digikey	Waterproof, lever arm
	Pico	1	30	SC0917	2648-SC0917-ND	Digikey	raspberry pi pico microcontroller
Resistors	R2, R3, R4	3	90	RNCP0805FTD1K00	RNCP0805FTD1K00CT-ND	Digikey	1k, 0805 thin film, 1/4W
	R5, R6, R7	3	90	RMCF0805FT100K	RMCF0805FT100KCT-ND	Digikey	100k, 0805 thick film, 1/8W
	R1, R8, R9, R11, R14, R15	6	180	RNCP0805FTD10K0	RNCP0805FTD10K0CT-ND	Digikey	10k, 0805 thin film, 1/4W
	R16	1	30	RMCF0805JT47K0	RMCF0805JT47K0CT-ND	Digikey	47k, 0805 thick film, 1/8W
	R10, R12, R13	3	90	RNCP0805FTD10R0	RNCP0805FTD10R0CT-ND	Digikey	10, 0805 thin film, 1/4W
Capacitors	C1	1	30	8.60011E+11	732-8666-1-ND	Digikey	100uF, 50V, rad
	C2, C9, C30	3	90	CL21B105KBFNNNG	1276-6470-1-ND	Digikey	1uF, 50V, cer 0805
	C3, C7, C8, C29	4	120	CL21B104KBCNNNC	1276-1003-1-ND	Digikey	0.1uF, 50V, cer 0805
	C4	1	30	8.6002E+11	732-8835-1-ND	Digikey	1000uF, 25V, rad
	C5, C6, C12, C13, C14, C15, C16, C18	8	240	CL21B105KPFNNNE	1276-1275-1-ND	Digikey	1uF, 10V, cer 0805
	C19, C20, C27, C28	4	120	CL21A106KPFNNNG	1276-6456-1-ND	Digikey	10uF, 10V, cer 0805
	C25, C26	2	60	8.6001E+11	732-8634-1-ND	Digikey	220uF, 25V, rad
	C10	1	30	CL21B103KAANNNC	1276-2434-1-ND	Digikey	10nF, 25V, cer 0805
	C11	1	30	8.85012E+11	732-7820-1-ND	Digikey	100pF, 16V, cer 0805
	C17	1	30	8.85012E+11	732-8057-1-ND	Digikey	2.2nF, 25V, cer 0805
	C21, C22, C23, C24	4	120	8.85012E+11	732-8015-1-ND	Digikey	220pF, 10V, cer 0805
Inductors	L1	1	30	1410454C	811-1345-ND	Digikey	100uH, 5.4A, rad
	L2	1	30	RLB0914-120KL	RLB0914-120KL-ND	Digikey	12uH, 2.5A, rad
	FB1, FB2, FB3, FB4	4	120	MH2029-121Y	MH2029-121YCT-ND	Digikey	120, 2A, 0805
Diodes	D1, D2	2	60	B340AXS-13	31-B340AXS-13CT-ND	Digikey	40V, 3A, Schottky
Misc	Jst b4b-xh for LEDs	1	30	B4B-XH-A	455-B4B-XH-A-ND	Digikey	4 pin jst xh connector
	Jst b4b-xh male for LEDs	1	30	XHP-4	455-2267-ND	Digikey	4 pin recetacle for jst xh
	Jst b2b-xh for uart and switches	4	120	B2B-XH-A	455-B2B-XH-A-ND	Digikey	2 pin jst xh connector
	Jst b2b-xh male for uart and switches	4	120	XHP-2	455-2266-ND	Digikey	2 pin recetacle for jst xh
	jst xh wire crimps	12	360	SXH-001T-P0.6	455-1135-1-ND	Digikey	may need extra
	20pin header for pico	0.4	12	4160	1528-2932-ND	Digikey	5pack, 20pin female, 2.54mm
	6pin female, sd card	1	30	RS1-06-G	2057-RS1-06-G-ND	Digikey	6pin female, 2.54mm

APPENDIX B – USER MANUAL

Path of Lights and Sounds
User Manual

Produced by:
Cal Poly Senior Project Team
Tagalong Trail

Produced for:
Girl Scouts of California's Central Coast
in conjunction with
Walk of Wonders

girl scouts of california's central coast

WOW Walk Of Wonders Events

Contents

Foreword

This manual has been created with the intent of aiding in all processes relevant to the light-up, musical tiles designed by Tagalong Trail, a group of 5 Cal Poly students. Care has been taken to detail every step of the process needed to remake, assemble, program, and set up the tiles for future events. This is with the caveat of exceptionally difficult parts of the design, as these should likely be outsourced or redone in the future to eventually be simpler and more efficient to make. If something is left out or glossed over, the team is likely trying to save you from a bad time.

1 Manufacturing

All parts and drawings referenced in this User Manual are available as part of the Hardware file Package as provided to the sponsor. Along with this comes the Bill of Materials (BOM). Both the drawings and the BOM are available bundled in the Drawings and Specifications package as well. These files are essential to understanding the sections of this manual and completing the manufacturing of the tiles.

1.1 Top and Bottom HDPE Sheets

Drawings for the top and bottom of the sheets of the tiles, Figure 44 and Figure 45 respectively, can also be referenced in the Drawing and Specifications package. Both sheets are made from a black, recycled High Density Polyethylene (HDPE) sheet.

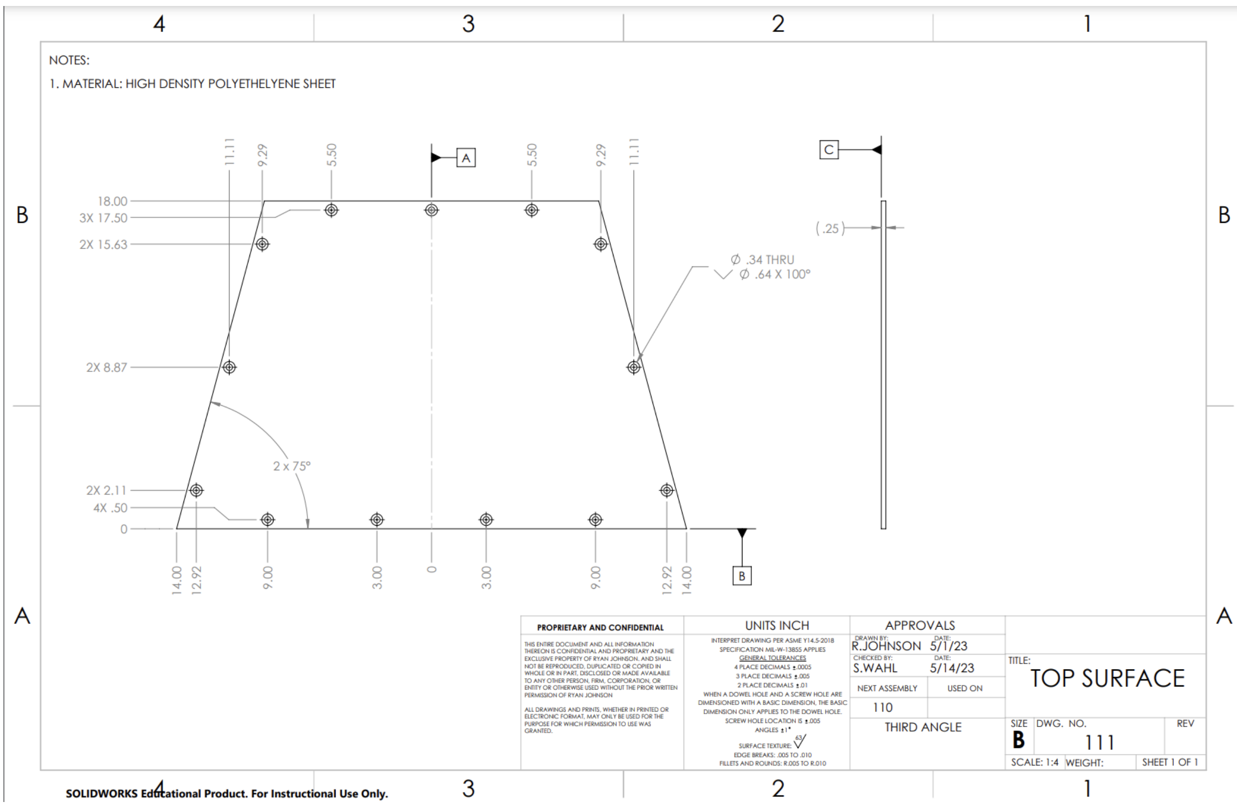


Figure 44 – Dimensioned drawing of the Top Surface (Part #111).

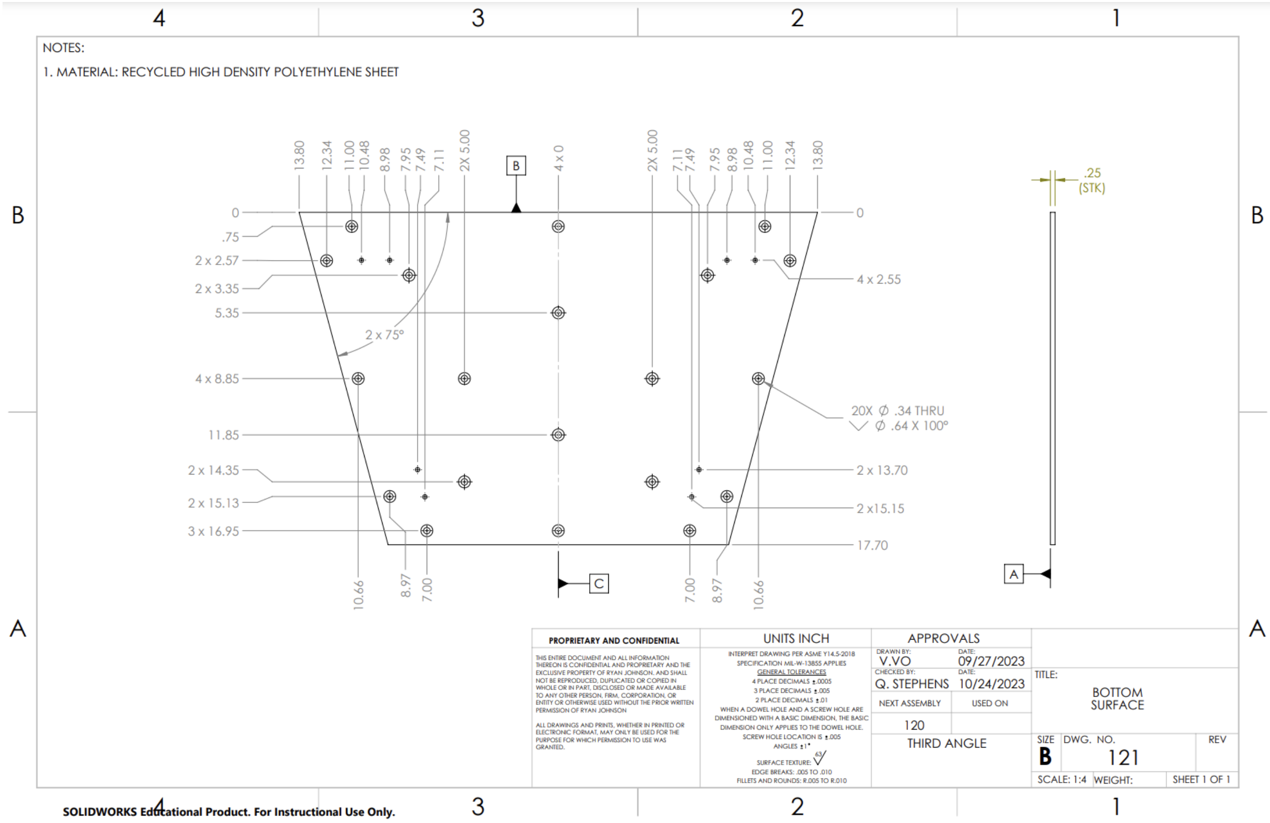


Figure 45 – Dimensioned drawing of the Bottom Surface (Part #121).

The manufacturing of the two surfaces was outsourced to the water jet team in the Cal Poly shops. With the use of the drawings shown in Figure 44 and Figure 45, and DXF (Drawing Interchange Format) versions of these same files, the water jet team was able to cut 7 of each surface from 48" x 96" stock sheets. Some unforeseen problems did occur with this process. The first of which was that the material sat on a vertical shelf for several days and warped under its own weight. This led to the water jet team needing to clamp or weight down the stock sheet. These clamps interfered with the path of the nozzle and made it more difficult to fit 7 tiles onto the stock pieces. The final problem, which delayed the manufacturing of the tiles significantly, was that the plugs created by the holes cut out by the water jet were the perfect size to plug the tank's drain.

Figure 46 – Water Jet cutting top sheets from stock HDPE sheet.

The post-processing required for the sheets was to countersink the holes on the bottom of the bottom sheets, and the top of the top sheets. This was done with a countersink bit, and a cordless drill. The depth of the countersink was eyeballed and checked by inserting a screw into the hole and checking for a flush fit. Additionally, the tabs created by the waterjet needed to be trimmed flush and the edges of the sheet needed to be deburred.

Another method that the team considered was to use a table saw and a specialized sled with jigs attached to cut out the sheets. This would have also required the holes to have been located through measurement and drilled out using a centerpunch and drill press with a 5/16” drill bit. Ultimately, the water jet team was able to finish the sheets in time and this idea was not required, but it might be worth pursuing if these sheets are manufactured again.

1.2 Aluminum Bottom 90 Degree Angles

Twenty-eight, 8 foot long, 1 ½" x 0.0625" sections of aluminum were used to manufacture the base frame for the tiles. These sections were processed systematically in a sequence of multiple steps to achieve uniformity and shorten manufacturing time.

Figure 47 – Layout of each aluminum section for initial bevel cut.

First, each section was clamped into a chop saw as shown in Figure 47 at a 37.5 degree angle so that the end of each section could be cut at this angle.

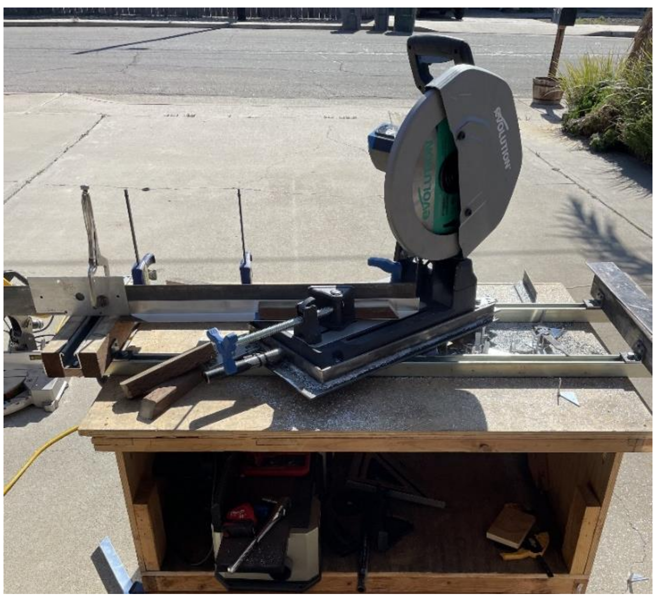


Figure 48 – Each section was flipped and cut at 37.5 degrees using a limit stop.

Next, each section was flipped and clamped as shown in Figure 48 at a 37.5 degree angle with a limit stop clamped against the fence of the saw. This technique ensures faster production time and greater precision over measuring each piece. At the completion of this step, part 122A was cut and ready for all tiles.

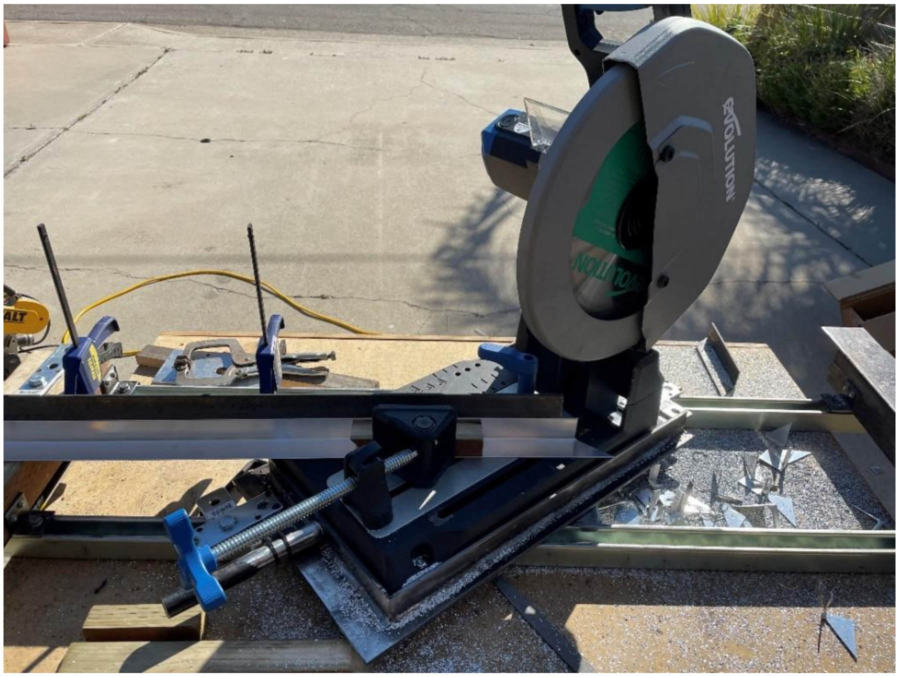


Figure 49 – Each section was flipped and cut again to re-establish 37.5 degree bevel.

Next, each section was flipped and clamped as shown in Figure 49 at a 37.5 degree angle to restore the appropriate bevel on the cut end of each piece. Each section was flipped and cut once more to finish part 122D.

Figure 50 – Saw was repositioned to a 52.5 degree angle to complete parts 122C and 122B.

To complete parts 122C and 122B, the saw was adjusted to a 52.5 degree angle. The remaining sections were cut at this angle using a limit stop clamped to the fence as shown in Figure 50. At the completion of this step, sections parts 122C and 122B were cut and ready for drilling.

Figure 51 – Drill press station assembled with fence to maintain consistent hole locations.

Mounting holes were drilled in each piece using the workstation shown in Figure 51. The drill press was used to ensure smooth and consistent holes. Parts were clamped against the fence to maintain consistency in mounting hole locations.

Figure 52 – First article test fit.

Figure 9 shows the completed first article. Dimensions and hole locations were within specified tolerances. All remaining parts were deburred, reamed, and separated into buckets for assembly as seen in Figure 53.

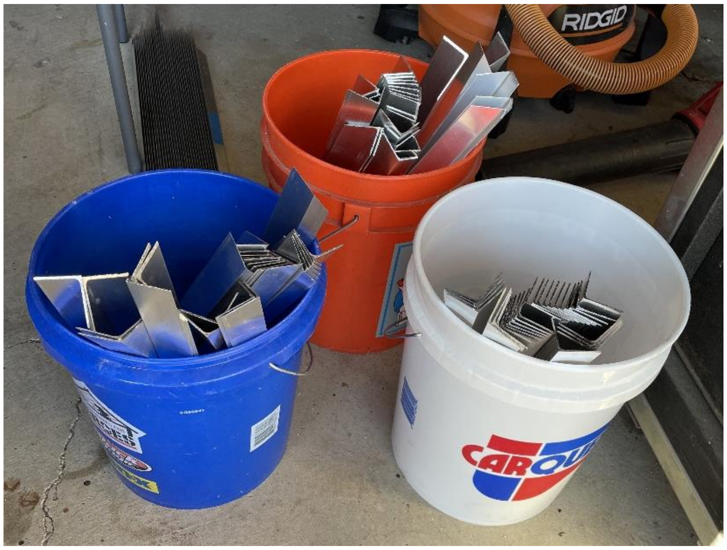


Figure 53 – Completed parts organized for final assembly.

1.3 UHMW Top 90 Degree Angles

The drawings for the UHMW (ultra-high-molecular-weight) top 90-degree angles drawings can be found in Drawing Package, or separately as their own files. Ultimately, the team designed the tiles based on the angles of the trapezoids, and this led to difficult to measure lengths of the 90-degree angles. Therefore, 3D-printed jigs were created to speed up and allow for accurate marking of the stock materials needed to be cut. The drawings for these Jigs can also be found in the Drawing Package, though the team has handed over the jigs they used to the sponsor. Figure 54, Figure 55, and Figure 56 all show the processes the Jigs allowed the team to accomplish.

First, the Jigs were clamped into place on the 5' sections of UHMW stock. Jigs for 112B, 112C and 112D all fit on one 5' length, while the Jig for 112A could fit twice on one 5' length. Every tile requires one of each of these four side supports. The edges of the jigs were marked using a fine tipped sharpie, then a centerpunch was utilized to mark the location of the holes that needed to be drilled as well as the locations for the "slots." The team advises that a mark is made with the sharpie in the locations that get centerpunched to make the centerpunch easier to locate when the holes are drilled.

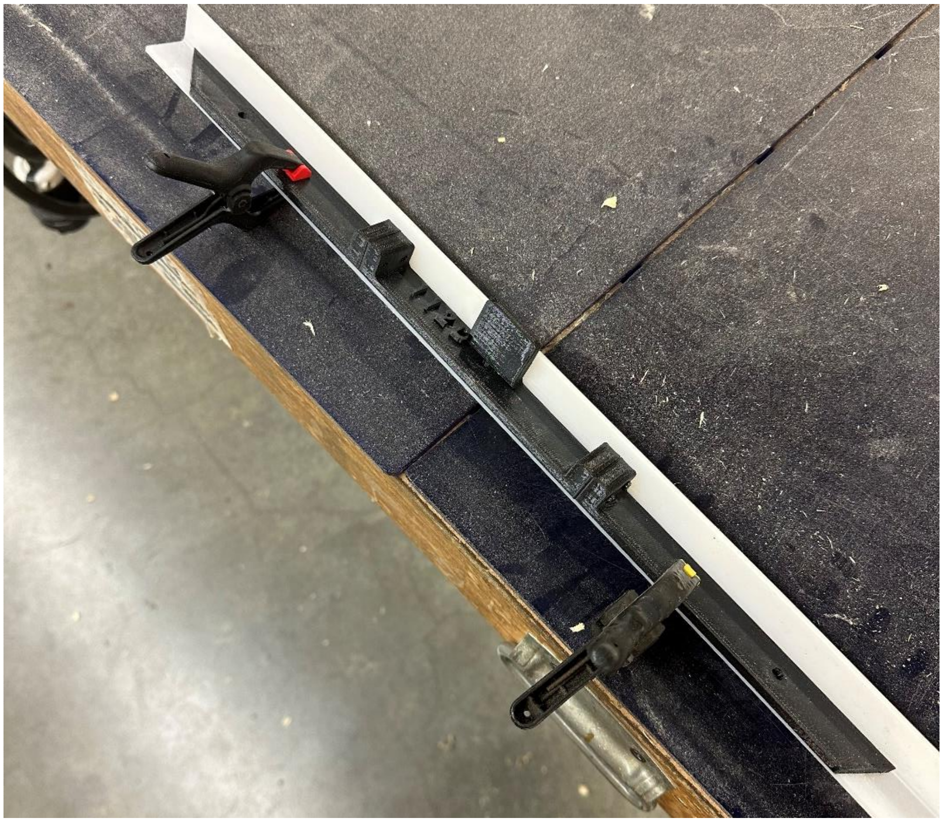


Figure 54 – Jig for part 112D clamped to stock UHMW.


Figure 55 – Marking of UHMW stock to be cut.

Figure 56 – Using a centerpunch to mark the hole locations in the stock UHMW.

After the stock was marked using the Jigs as guides, a miter saw with angle adjustability was used to make the cuts. The fence was pushed together as close to the blade as possible, to prevent the small end pieces from being pushed through and launched across the room. A piece of scrap wood, about 1" x 2" x 14", was utilized to help hold the stock against the fence while keeping the users' hands away from the blade.

Figure 57 – UHMW 90-degree angle stock being cut at a 37.5-degree angle on the miter saw.

Figure 58 – 90-degree UHMW stock being cut to size using a miter saw set to a 52.5-degree angle, and scrap wood used to help hold stock against the fence.

Once the stock was cut to length at the correct angles, the holes were drilled out using a 5/16" drill bit and a cordless drill. The team recommends clamping the pieces down to a scarp piece of wood and having both pieces clamped down onto a tabletop. The "slots" turned out to be much too difficult to accurately, and cleanly cut, so the group switched to utilizing a ½" drill bit to create a hole that functions as a slot.

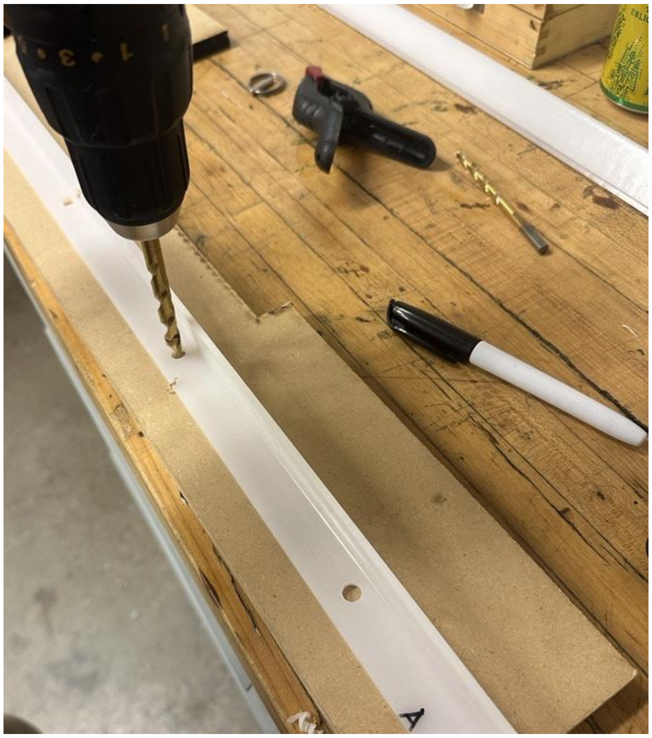


Figure 59 – Drilling the 5/16" holes in the cut UHMW at the locations marked with the center punch.

The team finished with the side slots that allowed for clearance between the side supports and the wiring that allows connection from one tile to another. They switched to using a 1” hole saw for this process as making a semi-circular cutout was quicker and easier than making a rectangular cutout.

Figure 60 – Cutting the side-slot in the top angled sides using a 1” Forstner bit.

1.4 Internal Supports

The drawings for the internal supports can be found in the Drawing Package. These supports work well, but the team found that manufacturing over 220 of them was much more work than they meant to sign up for and would suggest redesigning the supports to any group who might take on this project.

The first step in making the supports was to cut the 1.5” x 5’ UHMW rods down to 1 5/8” long pieces. To accomplish this, the group clamped a piece of wood down on the miter saw so that the face was 1 5/8” away from the face of the blade. This allowed for the group to quickly cut each piece precisely and quickly by simply butting the rod up against the wood before each cut. Vibrations from the saw cutting did result in the system sliding away from the blade slowly over time, so the group suggests adding a small piece of rubber or soft plastic under the wood to prevent this sliding. To remain as consistent as possible, the group used a pair of calipers to check the length after every 10 cuts.

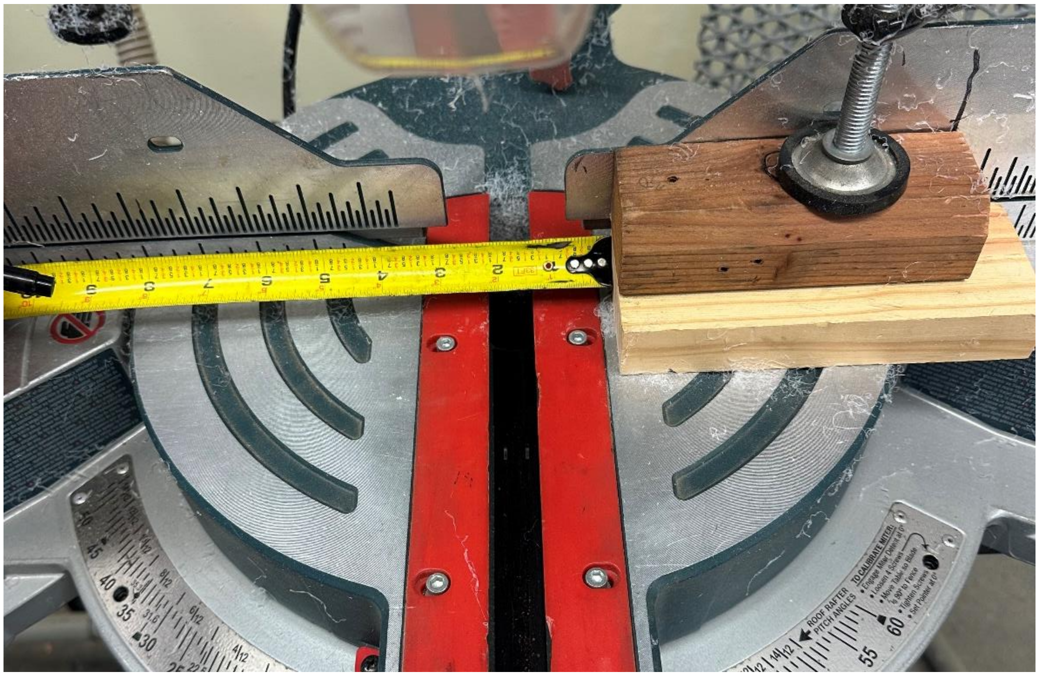


Figure 61 – Wood stop clamped down at 1 5/8" from face of the blade.

Figure 62 – Stock UHMW rod butted against stop and held in place to cut to length.

Figure 63 – Cut UHMW rods grouped into 8 pieces.

After the rods were cut to length and checked, they were moved to the lathe, where they were first drilled with a 5/16" through hole, then a 1" counterbore approximately 1 1/4" deep.

Figure 64 – Larger diameter counterbore being turned on the lathe.

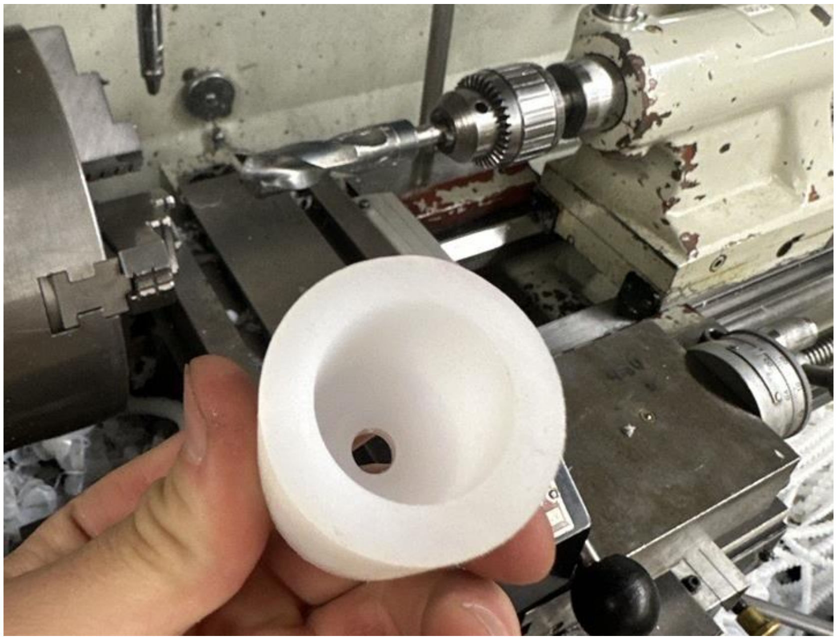


Figure 65 – Finished support.

Figure 66 – Ecstatic group member happy to be working on the very last support (number 220) needed for the project.

1.5 Miscellaneous

It is simple to make the rubber pads that get glued to the bottom of the tile. The group used the 6" x 6" 3D printed jigs (one square, and one with an angle of 75 degrees) and slapped them down on the 36" x 12' piece of 1/16" thick rubber stock. An Xacto knife was used to cut the pieces. Four pieces are needed with the angle cut into them, and one square piece is needed for every tile. The group then used rubber cement to glue the cut pieces into place. Finally, a wood burning pen (or soldering iron) was used to melt the holes for the stakes to go through into the rubber pads. However, the stakes are pointy enough to not need the holes burned through beforehand.

The limit switch supports were 3D-printed and the model can be seen in Figure 67. The CAD file can be found in the Hardware Package and should be saved as an ".stl" file and then sliced with the proper software before 3D-printing. Two limit switch supports are needed per tile and, once printed, fit snugly over the internal supports. They sit flush with the tops of the internal supports so that they are not load-bearing and so that the limit switch is at the perfect height to be activated. They also have a set-screw to be added so the support does not move around once installed.

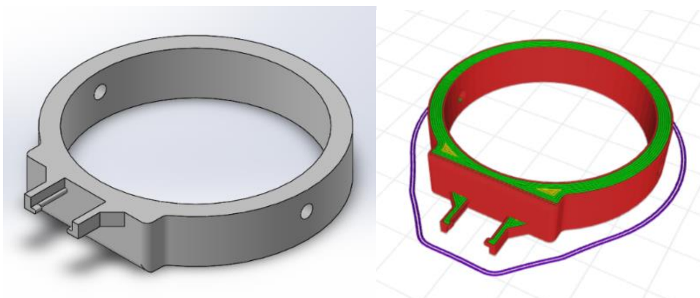


Figure 67 – CAD model of the Switch Bracket.

The provided CAD model is specifically designed to accommodate the limit switch detailed in the Bill of Materials (BOM). Modifications to the model are necessary if an alternative switch is to be used. This model incorporates a switch bracket, which is equipped with two slender slot slides, each measuring approximately 1mm in width, as depicted in the figure provided.

To achieve the best possible fit, certain 3D printing parameters and file orientation guidelines are recommended. It is essential to adhere to these guidelines during the slicing process to ensure compatibility and optimal results. It is flexible to choose any 3D printer since this is an easy bracket. However, it is recommended to ensure that the printing parameters closely follow the specifications outlined in Table 6 below. This approach ensures that the final printed component aligns accurately with the intended design with the specified limit switch.

Table 6 – Slicing parameters for limit switch supports.

Parameter	Value	Unit
Layer Height	<0.16	mm
Speed	<100	mm/s
Support	None	N/A
Adhesion Type	Skirt	N/A
Infill Density	>75%	N/A

2 Structural Assembly

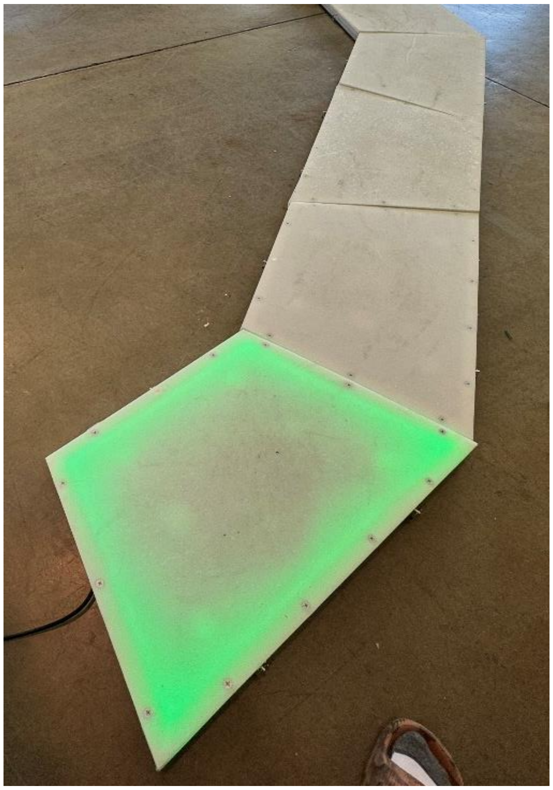


Figure 68 – Fully assembled and activated tiles ready to be enjoyed.

2.1 Top Assembly

The assembly of the top of the tile involves the use of a Phillips head screwdriver, a socket wrench (ideally a drill with a socket attachment) with a 7/16” socket, a foam roller at least 18” long, and blue painter’s tape. The group suggests starting the assembly by gathering the 4 separate sides, the top surface, and the 13 nuts and bolts.

Figure 69 – Group member hand tightening nuts for top surface assembly.

Figure 69 shows the group member hand tightening the nuts and bolts that secure the top surface to the side supports. Hand tightening the bolts will allow for the assembler to line up the holes in the side supports, with the holes in the top surface, while still allowing for adjustment. Next, the group used the blue painter's tape to tape the corners tightly together, and to tape the side supports to the top surface. This is shown in Figure 70.

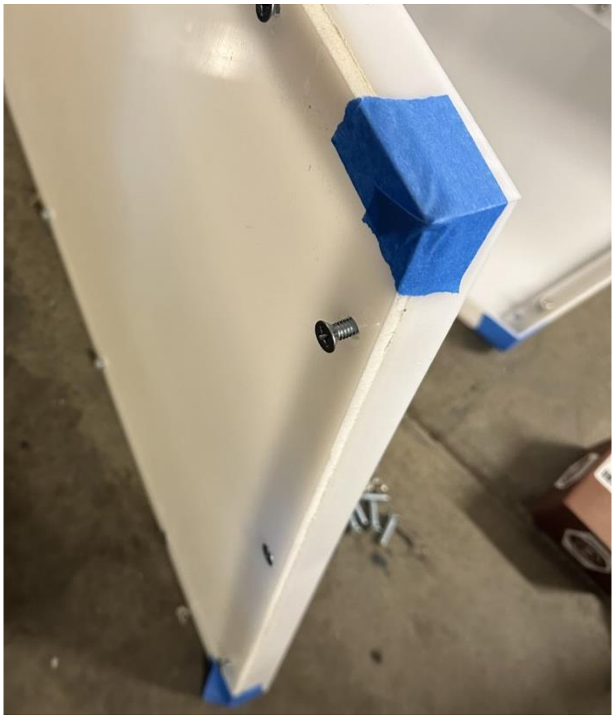


Figure 70 – Taped corners for top assembly.

For the next step in the process, the group heavily suggests using a drill with a socket attachment and a Phillips head screwdriver to complete. Holding the screwdriver in one hand, and the drill in the other, use the screwdriver to hold the screw in place, and then use the 7/16" socket to turn the nut down until it is tight. The tape should hold the side supports in place, but the group suggests using a partner to help line up the side supports so they are flush with the top surface. Figure 73 in Section 2.2 shows this action being completed for the bottom assembly, which is similar to this top assembly step.

Once the tile has its side supports in place, clear grip tape can be laid down on the tops of the tiles. The group found it was best to clean the top surface first with 70% Isopropyl Alcohol and a rag, then lay the grip tape straight sown while holding at all four corners to ensure it lines up with the tile. This can be seen in Figure 71.

Figure 71 – Team laying down clear grip tape.

Once the grip tape is in place, a foam roller was used to evenly pressure roll the tape onto the top surface of the tile. After the tape was pressed into place, the edges were trimmed using a razor blade. The screw locations will be small air pockets, so to fix this, simply poke hole in the grip tape using a Philips screwdriver. This can be seen in Figure 72. This completes the top assembly.

Figure 72 – Using a Philips head screwdriver to poke a hole in the grip tape.

2.2 Bottom Assembly

The bottom assembly matches a lot of steps with the top assembly, but differs in a few key ways. The bottom sheet is made from recycled HDPE, and the side supports are 1.5” aluminum 90-degree angles. The bottom assembly is also where the internal supports and electronics are secured. No grip tape is needed, but instead rubber pads are attached to the base of the bottom sheet.

Like in Section 2.1, the group suggests using a drill with a socket attachment to secure all of the nuts and bolts in this assembly. The group also suggests using tape on the aluminum side supports similarly to how the top side supports were taped, shown in Figure 70. Securing the side supports to the base, making sure that the bottom sheet has the countersunk side face down, requires 12 bolts and nuts. Assembly can be seen in Figure 73.

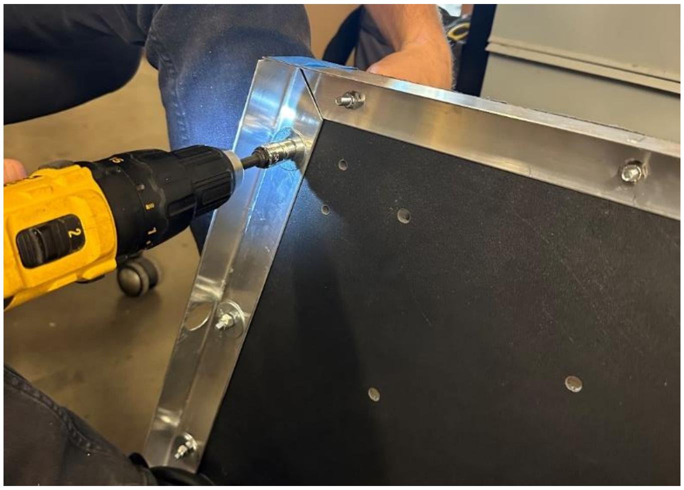


Figure 73 – Group member using drill with socket attachment and Phillips head screwdriver to tighten bolts into place.

After this step is complete, the bottom assembly stops mimicking the top assembly. The next step is to secure the internal supports. This is when the drill extension seen in Figure 73, or a deep socket is required. Place the bolt through the holes in the internal section of the tile and hold that with one hand. Then place the nut in the internal support and use a finger to hold that securely in place while lining up the through hole in the support with the extending portion of the bolt. Make sure that the nut is securely lined up with the through hole and placed in the larger bored hole. Continue holding pressure on the nut and hand tighten the bolt onto the nut. Use the drill and screwdriver to secure the support. Each tile requires 8 internal supports.

The final step in the bottom assembly is to glue the rubber pads to the bottom of the bottom sheet. This step requires five 6" by 6" rubber pieces, and Wellwood contact cement. The process for this can be seen in Figure 74. This process involves applying the glue to both the rubber pads and the base of the tile, then waiting 15 minutes for it to cure before applying and pressing. The pads are placed in the four corners of the tile, as well as the center. Any overhang can be trimmed with a razor once dried.

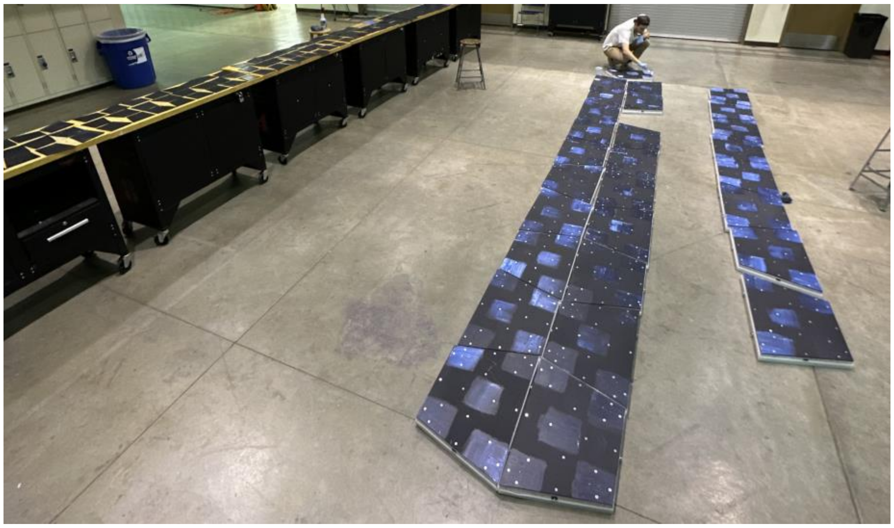


Figure 74 – Group member applying contact cement to the base of the tiles, while the contact cement applied to the rubber pieces (in the left side of the image) cures.

2.3 Full Assembly

To fully assemble the tiles, you will need to have the electronics enclosure complete as well. Please reference Section 3 to get this done. Once the electronics enclosures are made, come back to this section.

The first step in finishing the tiles is to apply the LED strip along the inside edge of the base of the tile. Take an LED strip (should have been cut in half and connectorized in Section 3) and lay it along the inside edge as seen in Figure 75. Have the connector lay underneath the pinhole as seen in the figure. After laying it out, begin to apply the strip as seen in Figure 76 by peeling back the blue tape-cover and pressing it against the tile.

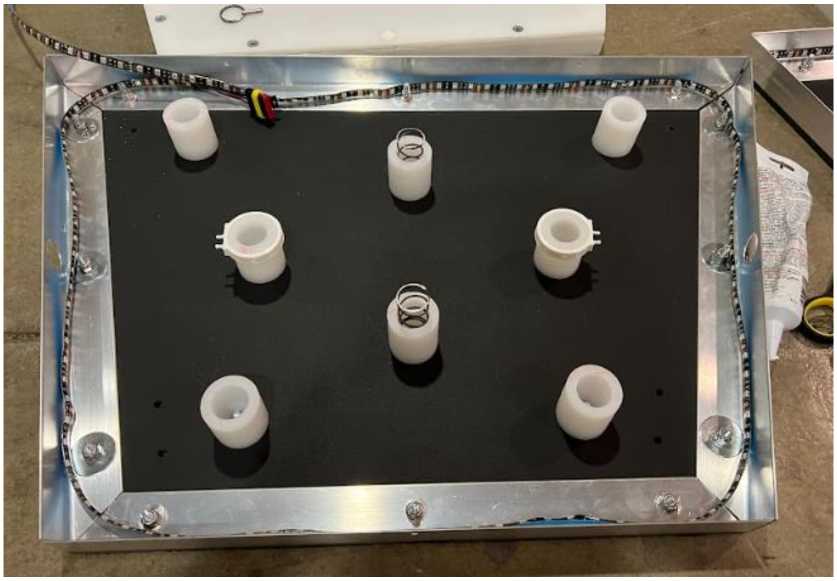


Figure 75 – LED strip laid out in the base of the tile, unapplied. Note that the connector is underneath the long-side's left pinhole and the strip goes around counter-clockwise from there.

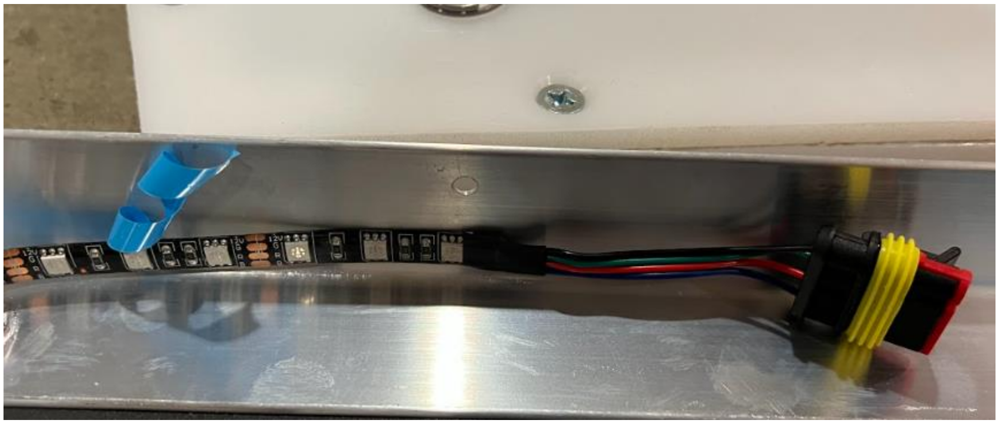


Figure 76 – The beginning of LED strip application. Make sure that the strip is being placed along the bottom edge of the vertical side.

Next, continue applying the LED strip all the way around the perimeter of the tile. Be sure to barely obstruct the wire holes as in Figure 77. Also be sure to not pinch the cable in the corners, leaving them looking something like Figure 78. Once fully around, it is noted that there is excess length on the LED strip. Trim the strip to the nearest silver or bronze line, as seen in Figure 79.

Figure 77 – Ensure the LEDs do not significantly cover the wire cutout.

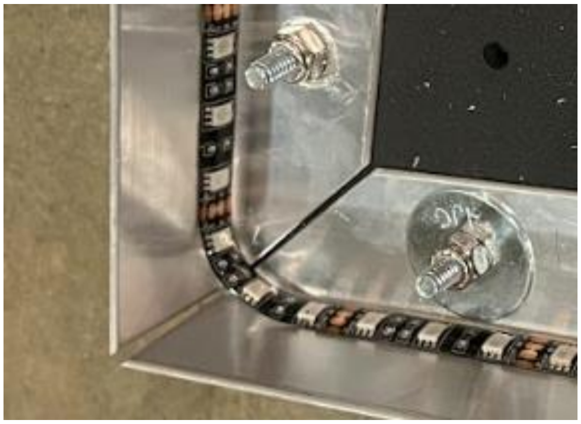


Figure 78 – Do not sharply bend the LEDs when transitioning around corners. Leave an adequate bend radius as seen above.

Figure 79 – Trimming the LEDs with scissors at a silver line. If needed, the LEDs can be trimmed at the bronze line seen near the right of the image.

After trimming the LED strip, apply sealant to the end of the strip as seen in Figure 80. After this, the LED strip can finish being adhered to the perimeter of the tile as in Figure 81.

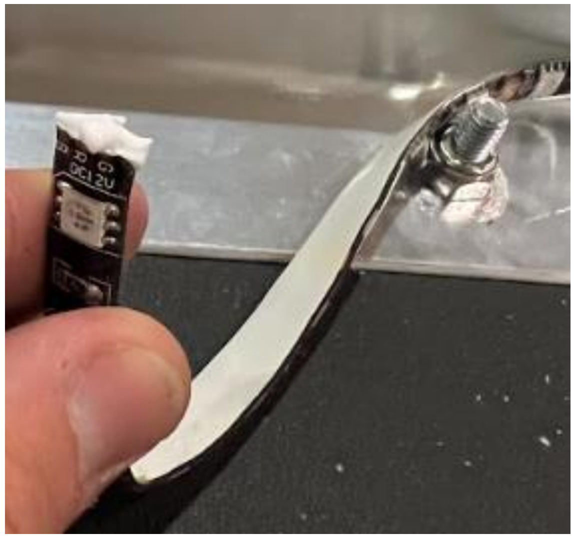


Figure 80 – LED strip with freshly sealed end.

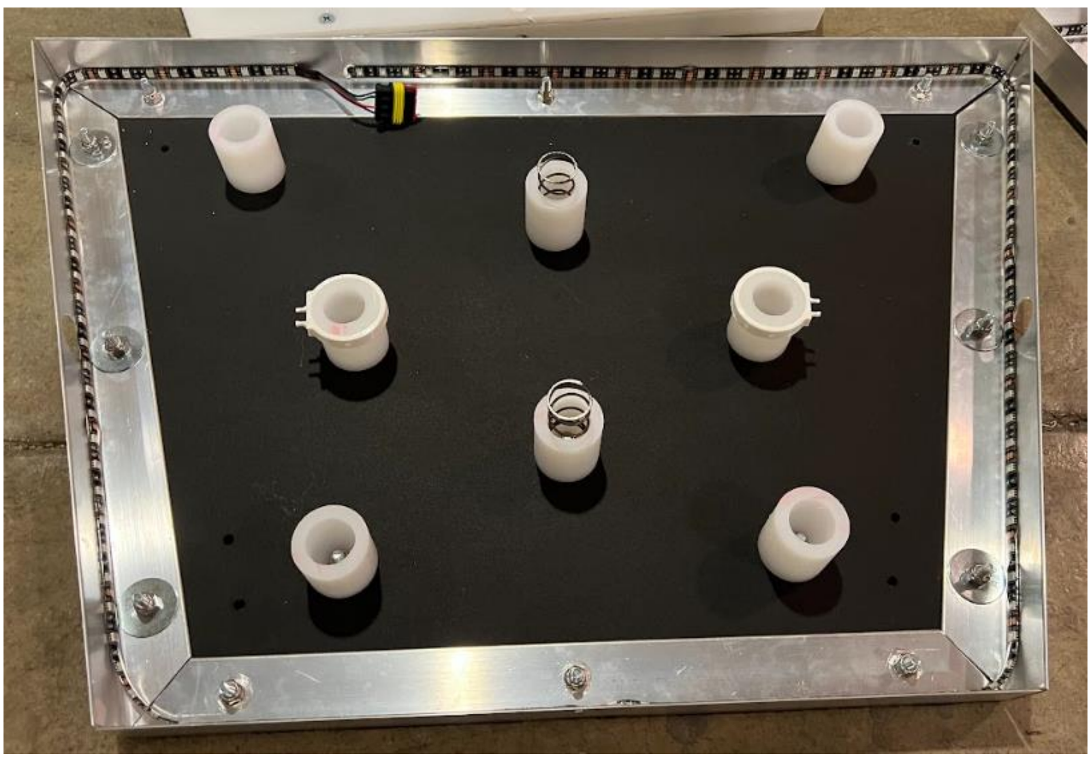


Figure 81 – Fully adhered LED strip.

Now, with 1”x1” Velcro squares, apply the hooked-side of the Velcro to the center of the base’s internal side. Apply the looped-side to the center of the electronics enclosure’s underside as in Figure 82. Afterwards, place the enclosure inside of the tile, oriented as in Figure 83. Connect the LED connector.

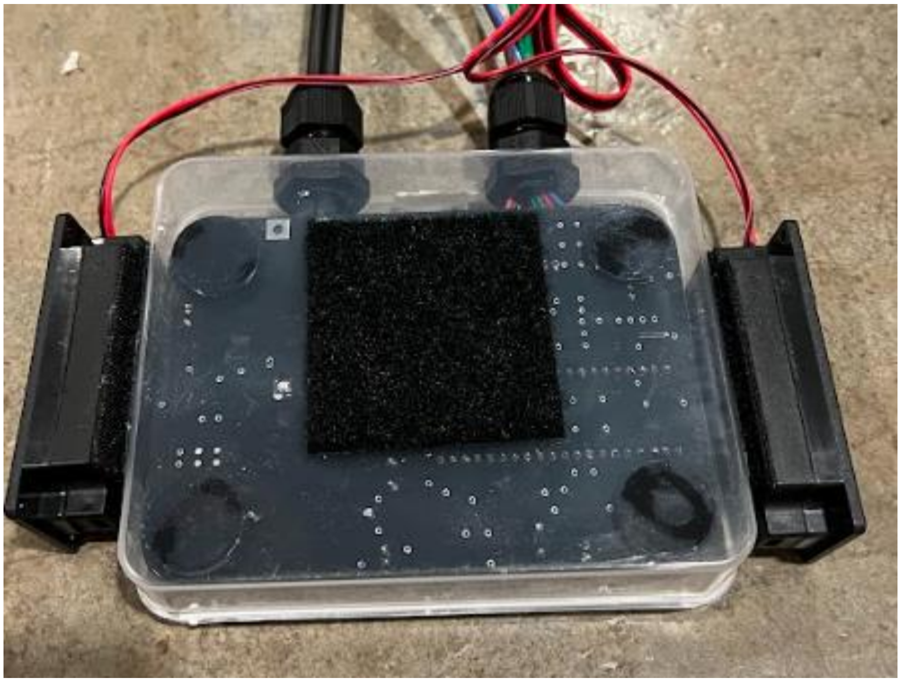


Figure 82 – Enclosure with Velcro placed on underside.

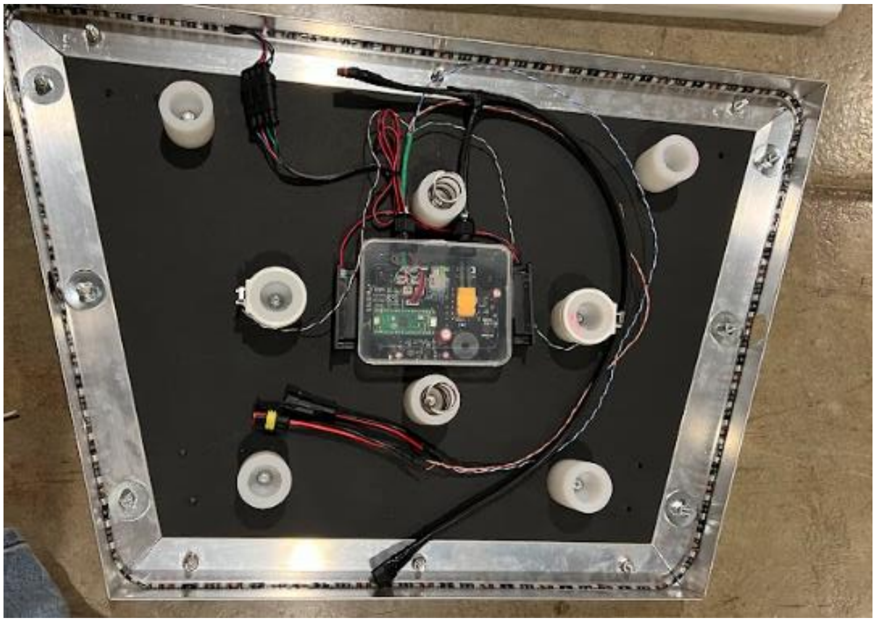


Figure 83 – Enclosure stuck into the base of the tile. Ensure wires are facing the long edge.

Two of the internal supports house the tiles springs, and two require the limit switch holsters. The switch holsters can be press fit onto the supports and a set screw can be drilled into place as in Figure 84. The springs will simply fit into place in the supports. For an image showing where these springs and holsters are located, see Figure 83 above. Insert the limit switches into their holsters as in Figure 85.

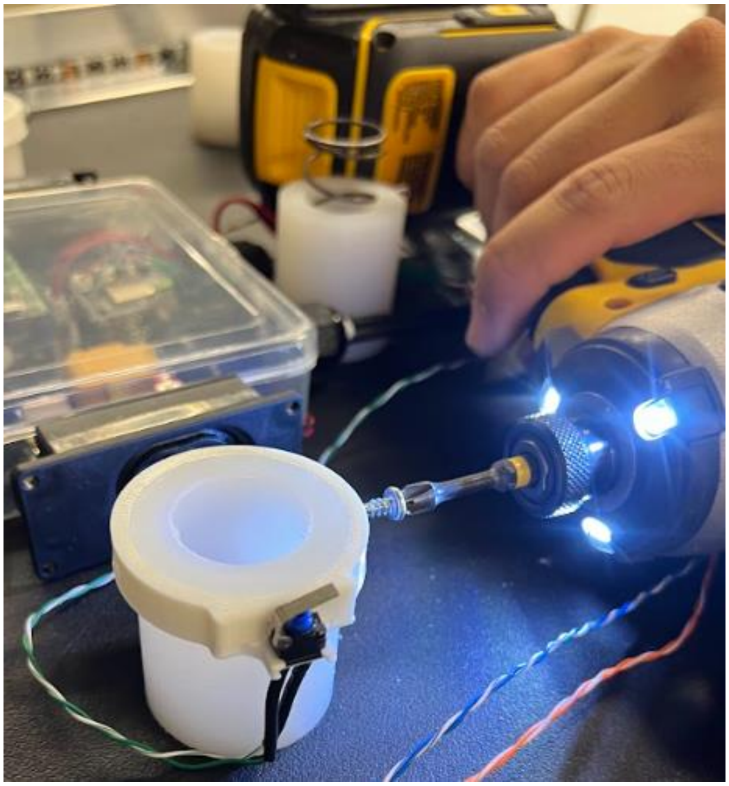


Figure 84 – Drilling set-screw into limit switch support's hole.

Figure 85 – Limit switch seated in its holster.

With this, the tile can now be shut until it is time to be set up. Once fully assembled, the top assembly will fit over the top of and slide down onto the bottom assembly. This clearance fit is very tight and can be tricky. The group recommends first fitting the short ends together with the top assembly at an angle and rotating it down onto the bottom assembly. The long side may require some slight pressure to be applied to both the bottom and top side supports in order to get the tile to seat down correctly. Once in place, the three ringed pins can be pushed into place in the short and long side supports. The pins can be seen in Figure 86 and Figure 87. They should go through the larger top side support hole, and into the smaller hole in the aluminum bottom side supports. This completes the assembly of the tiles.

Figure 86 – Three quick-release pins should be placed in the holes under the three locations above. Note that there should be a quick-release pin in the red circle.

Figure 87 – Insert all quick-release pins into their holes, going through the lid first.

3 Electronics Assembly

The goal of this section will be to create one full electronics assembly as pictured below in Figure 88, complete with a circuit board, speakers, an LED attachment cable, 2 UART cables, 2 limit switches, and a power cord.

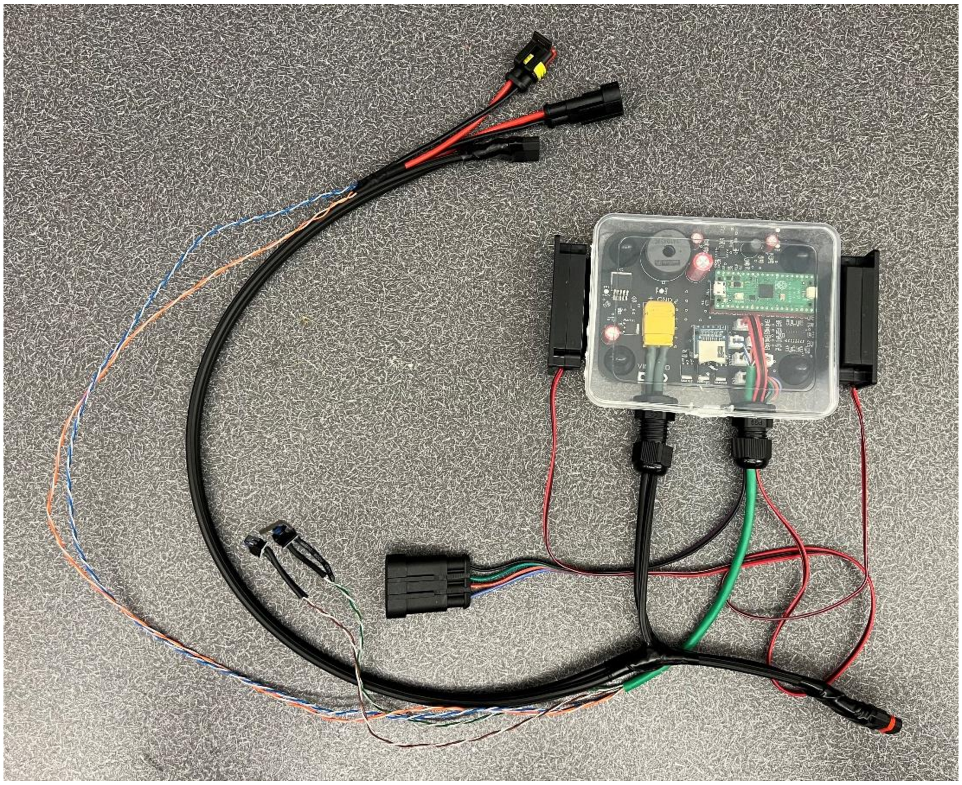


Figure 88 – One completed electronics assembly.

Though entirely possible to make, the team has found this part of the design to be completely cumbersome to manufacture and assemble without previous experience and excessive mental fortitude. The team spent an excess of about 120 man-hours to make 30 of these enclosures. In small quantities, these can be made by hand by someone with a background in electronics, PCB (printed circuit board) design, or robotics. Otherwise, the team would recommend one of three options:

1. Outsource the work to a professional. They would need the wiring diagram, the PCB files, and the bill of materials to do this properly.
2. Learn the necessary skills to be able to make this assembly. This could also entail just hiring an employee with these skills to do this for you.
3. Redesign the electronics enclosure in a future project specifically with the goal of refining the manufacturing and assembling processes.

The rest of this section is going to be assuming that you chose option two and are putting yourself through the pain of making some amount of electronics enclosures, though the recommended option is choice 3.

3.1 Printed Circuit Board

The final printed circuit board is featured in Figure 89 fully assembled. Though some improvements could be made to this board, such as changing the power connector and relocating the connectors, this board is completely functional and was used in the final build provided by the team.

Figure 89 – Fully assembled PCB in an unfinished enclosure.

To get a finished circuit board, there are two methods take:

1. Assemble the PCB yourself. Only choose this option if you are comfortable with soldering and know how to use stencils, solder paste, and a hot plate.
2. Have a third party build and assemble the board for you. To do this, some edits to the board may need to be made as well as making sure the assembling company has access to all the necessary parts in reeled supply. This is going to be more expensive.
3.

The team believes that both options are possible through JLCPCB, a PCB manufacturing company. However, because of the team's experience, they chose option 1 of assembling the boards themselves once received. If option 2 is chosen, there is likely going to be more money spent as well as time committed to ensuring the design can be assembled by JLCPCB with components that they have in house, or researching if the current components can be sent to them for assembly. Since the team has not explored option 2, only option 1 will be covered, though this does not negate the validity of using option 2.

Pursuing option 1, one should first navigate to JLCPCB's website. Once there, take the "W15_Final_Board_JLCPCB.zip" file from the Hardware Package provided and drop it in the "Add gerber file" selection box (seen in Figure 90).

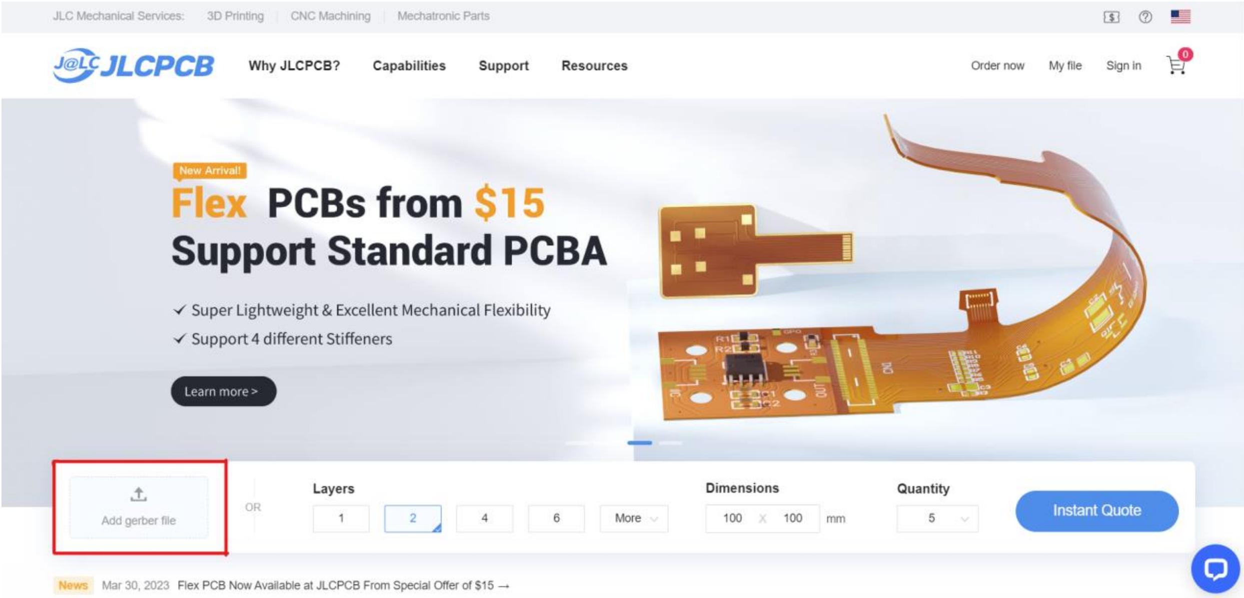


Figure 90 – JLCPCB's homepage with "Add gerber file" selection box outlined in red.

After this, a new page is brought up that looks something like Figure 91. A preview of the board is shown, followed by many options below. Don't be intimidated, most of these options can be left alone.

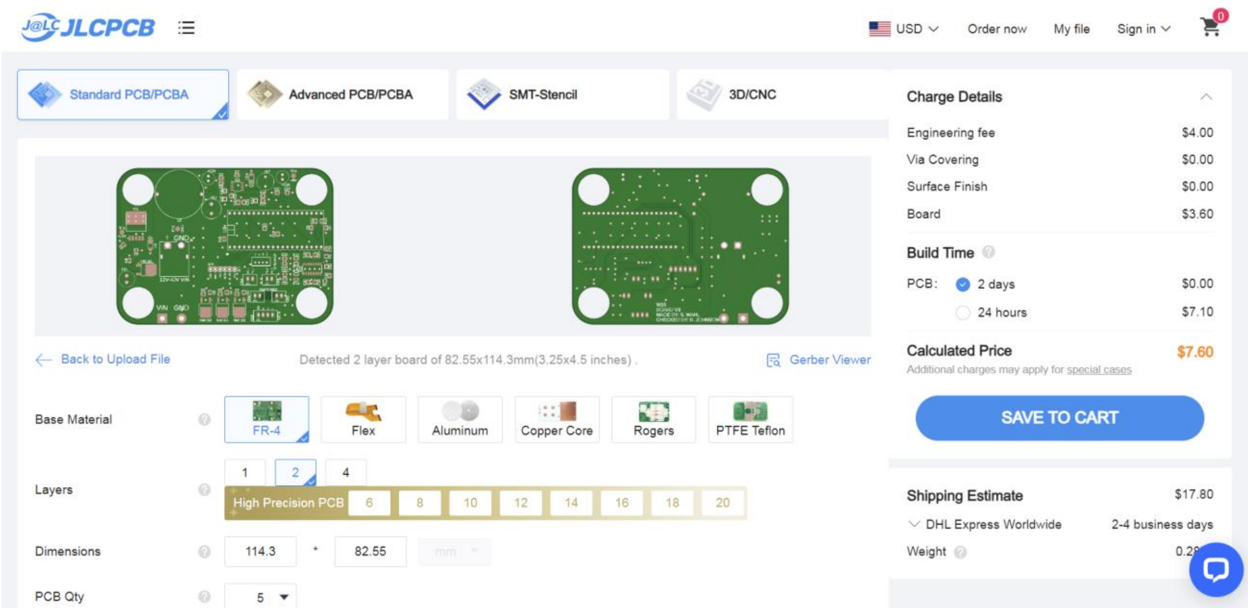


Figure 91 – Resulting JLCPCB page after dropping in a gerber ".zip" file collection.

Of the options presented, leave all of them at their default setting except for the following:

- **PCB Qty:** Change to however many PCBs you are trying to make.
- **PCB Color:** Change to whatever color you want. Some colors are more expensive and come with longer lead times, but the team made their PCBs black.
- **Remove Order Number:** The team specified 'Yes,' though this is not a required option. If 'No' is selected, JLCPCB's internal order number will be printed somewhere random on the board.

No other options of note need to be changed. DO NOT order a stencil. The team ordered a stencil for their production and has provided it to the sponsor for future use. When the PCBs are received, they should come in a blue box and be packaged similarly to Figure 92 inside.

Figure 92 – Packaged PCBs once received and opened.

Next, to order the components for the PCB, go to the 'Bill of Materials – W15.xlsx" file and open the "DigiKey BOM" sheet. Update the number in the top left to the number of boards you are planning on ordering. Save the sheet as a ".csv" file and go to DigiKey's website. Open your cart in DigiKey and drag the ".csv file into the highlighted section in Figure 93.

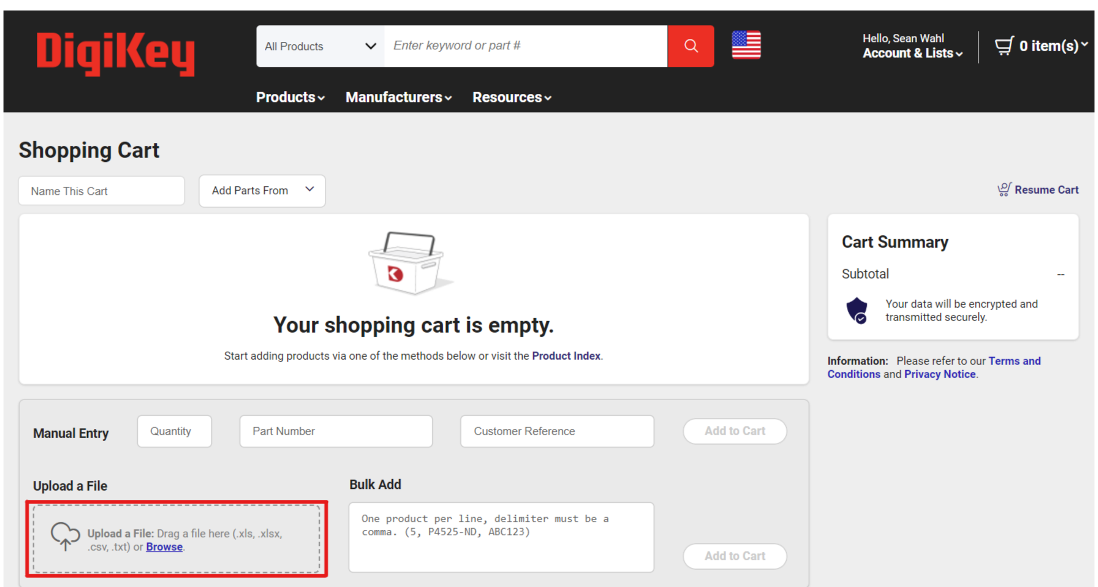


Figure 93 – DigiKey's cart page, with the ".csv" upload box outlined in red.

Ensure that DigiKey has correctly identified the "Quantity," "Manufacturer Part #," and "DigiKey Part #" columns. Select the "Component" column from the ".csv" file and make it the "Customer Reference" column. Once added to the cart, go through DigiKey's suggested changes to finalize the order. Most of DigiKey's suggested changes will save you money, but the choice is at your discretion. It is recommended to add a couple of spares for every component in case one is nonfunctional or gets lost during assembly.

There are some components that need to be ordered separately, such as the XT-60 connectors and the JST PH connectors. These would be found on the "Electronics BOM" sheet of the excel file previously mentioned.

Once the DigiKey order and other components are received, the soldering is up to you. The team used the PCB stencil to apply solder paste, placed all surface mount components, and then used the hot plate to adhere them to the board. After this, all through hole components were soldered by hand to the board. The customer references should be clear enough to match with the board designators.

Once the boards are complete, it is recommended. to hook them up to a 35V power supply with a low current limit to check that the regulators are working and provide the specified voltages. There are banana clip locations at the front of the board in line with the spot for the XT-60 for easily supplying power to the board. There are three test points on the board for easy multimeter probing. These are TP1, TP2, and TP2, respectively corresponding to +12V, +5V, and GND.
This completes the PCB manufacturing section.

3.2 Enclosure

Making the enclosure is relatively straightforward compared to the other processes in this section. At the end of this section, you should have an item that looks like Figure 94.

Figure 94 – Finished enclosure at the end of the processes in this section.

Though no formal engineering drawing was created for this part due to its rudimentary nature, a sketch of important dimensions for drilling the holes in the front of the enclosure is present in Figure 95.

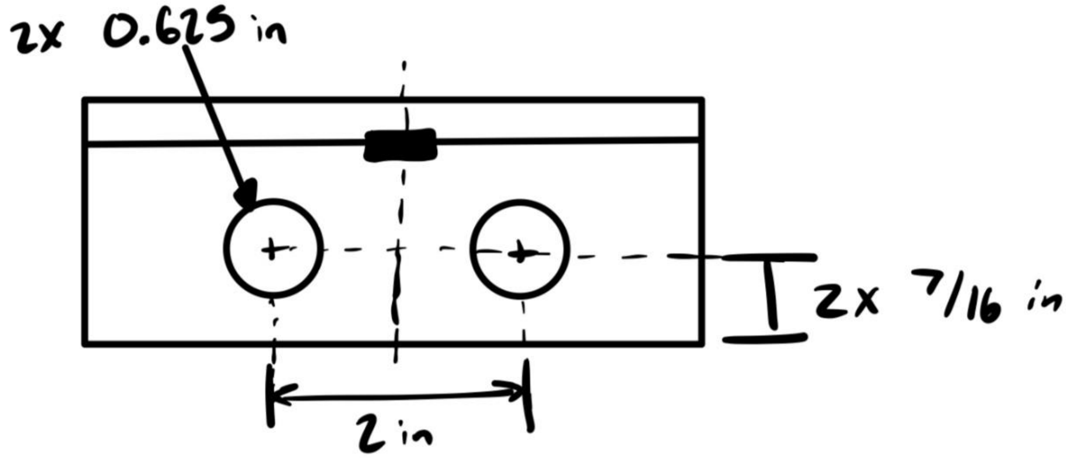


Figure 95 – Enclosure with hole dimensions located.

The first step in making this enclosure is to drill the holes above. Since the enclosure is very fragile and easy to break, several steps need to be taken in doing this. Below is a detailed list to create these holes.

1. Use calipers to mark the 7/16" horizontal line across the front of the enclosure. The edge of the calipers should be sharp enough to scribe a thin line in the plastic.
2. Have the calipers set at a distance of 2". Eyeball the middle of the container and mark vertically down across the front of the enclosure. This location is not critical but should be close enough to centered around the midplane of the enclosure.
3. Center drill at the two intersections created by the scribes that were just made (the locations of the two holes). Ensure that the plastic is supported on the other side of the thin wall by something like a piece of wood. A small enough drill should be used to accurately maintain the location of the holes. Personally, the team used a center drill from a mill for this step.
4. Slowly size up drills, supporting the plastic and increasing the hole size. The team used 3 separate drills after the center drill to get up to the 5/8" hole size. The two intermediate drills that the team used were ¼" and 7/16". Care should be taken when drilling to ensure the hole location is not moving and that the plastic is not bending or fracturing.
5. The holes should now be complete. Inspect the holes and ensure no cracks have propagated elsewhere through the enclosure.

After the holes are drilled, all that needs to be done is to lay a bead of sealant around the thin rim of the enclosure. This will not completely waterproof the enclosure but it will improve its water resistance. Areas where gaps are present in the enclosure should also be covered in sealant. Let the sealant dry for around 24 hours before closing the enclosures again.

3.3 Wiring

Getting the wiring setup properly is more time intensive than it initially appears. A wiring diagram of the system can be seen in Figure 96, followed by detailed harness images in Figure 100 to Figure 103.

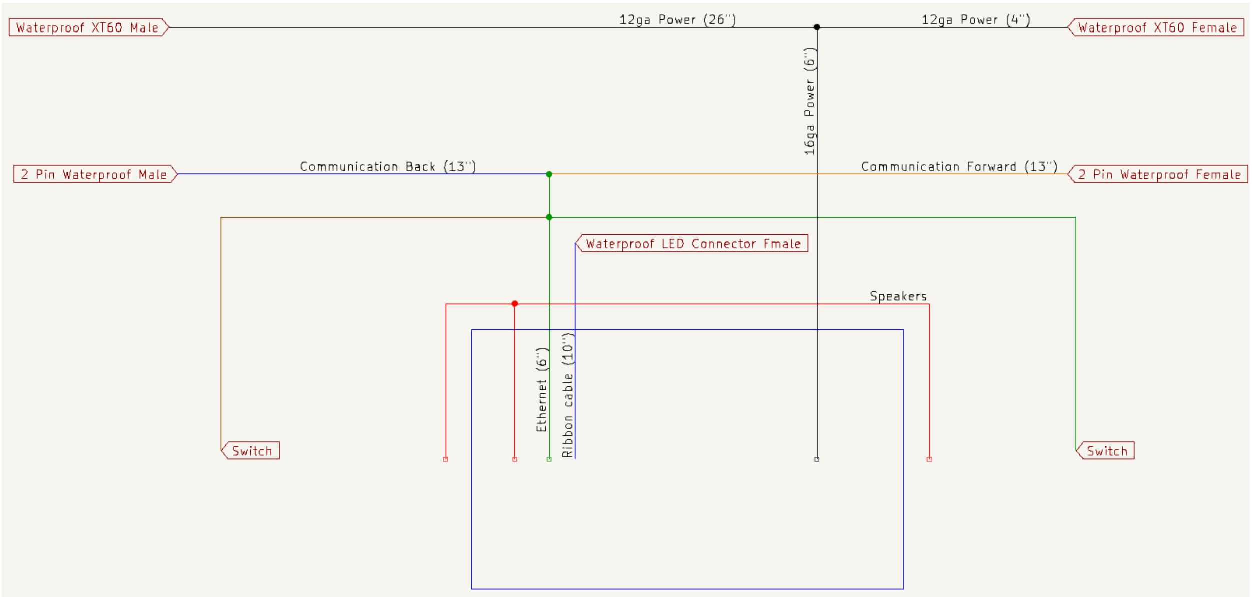


Figure 96 – Wiring diagram with wire lengths.

Following the wiring diagram, cut a set of the wires detailed for each of the tiles you are planning on creating. The next step is to crimp each of the individual wires with the correct crimp for their housing. Explicitly, the wires the need to be crimped are listed below:

- LED wire, board side, JST XH (4 total crimps)
- LED wire, junction side, waterproof AD2 (4 total crimps)
- UART wires, board side, JST XH (4 total crimps)
- Switch wires, board side, JST XH (4 total crimps)
- LEDs (5m strip cut in half), junction side, waterproof AD2 (4 total crimps)

JST connectors can be put on the Ethernet cable wires connecting to the board (for UART and the switches) and on the LED wire board-side. The waterproof AD2 connector can be put on the LEDs themselves, but not on the LED wire connecting to the board. Be sure to follow the connection convention from the images above when putting on the housings.

3.4 Assembly of Enclosure

To begin the assembly of the enclosure, the PCB needs to be added. To do so, take four of the internal standoffs and push them through the bottom of the PCB's four mounting holes. They should sit in place, as seen in Figure 97.

Figure 97 – PCB with internal rubber standoffs pushed through the mounting holes.

Once secure in the mounting holes, the sticker protecting the adhesive on the standoffs can be removed. Next, stick the PCB securely down in the back-center of the enclosure. If needed, hold the PCB by the large, black inductor on the back-left of the board. Once in the enclosure, press down on each of the four standoffs to adhere them to the enclosure. See Figure 98 for a properly seated board.

Figure 98 – Enclosure with properly seated PCB.

Once the PCB is located inside the enclosure, the left PG9 gland can be inserted into the enclosure, as seen in Figure 99. Note that the gasket should be on the outside with only the nut inside.


Figure 99 – Enclosure with PCB and front left gland installed.

Installing the right gland is not as simple as the left gland, as the right gland has many sets of wires passing through. The connectors do not easily pass through the gland, so they must be fed through before installation. These steps should be followed in doing so:

1. Separate the gland's large outside cover, its gasket, and its main body (with nut preferably unattached, though this does not matter much until the next process).
2. Pass through the speaker's small JST connector first through the large outside cover, then the gasket, and then the main body. It will not exactly fit easily through any of these items. A good strategy is to use tweezers and hold the JST connector sideways as it passes through each, with its wires bending perpendicularly behind it.
3. Pass the LED wire's crimped ends through the opposite way, first going through the main body, then the gasket, and then the outside cover. Do not pass through all 4 pins at the same time. Pass two through at a time, and feel free to peel some of the wire from itself.
4. Pass the green ethernet wire through (uncrimped end first) the main body, then the gasket, and then the large outside covering. This should be the easiest to do.

At this point, the gland should look something like Figure 100. A close-up version of the gland is in Figure 101.

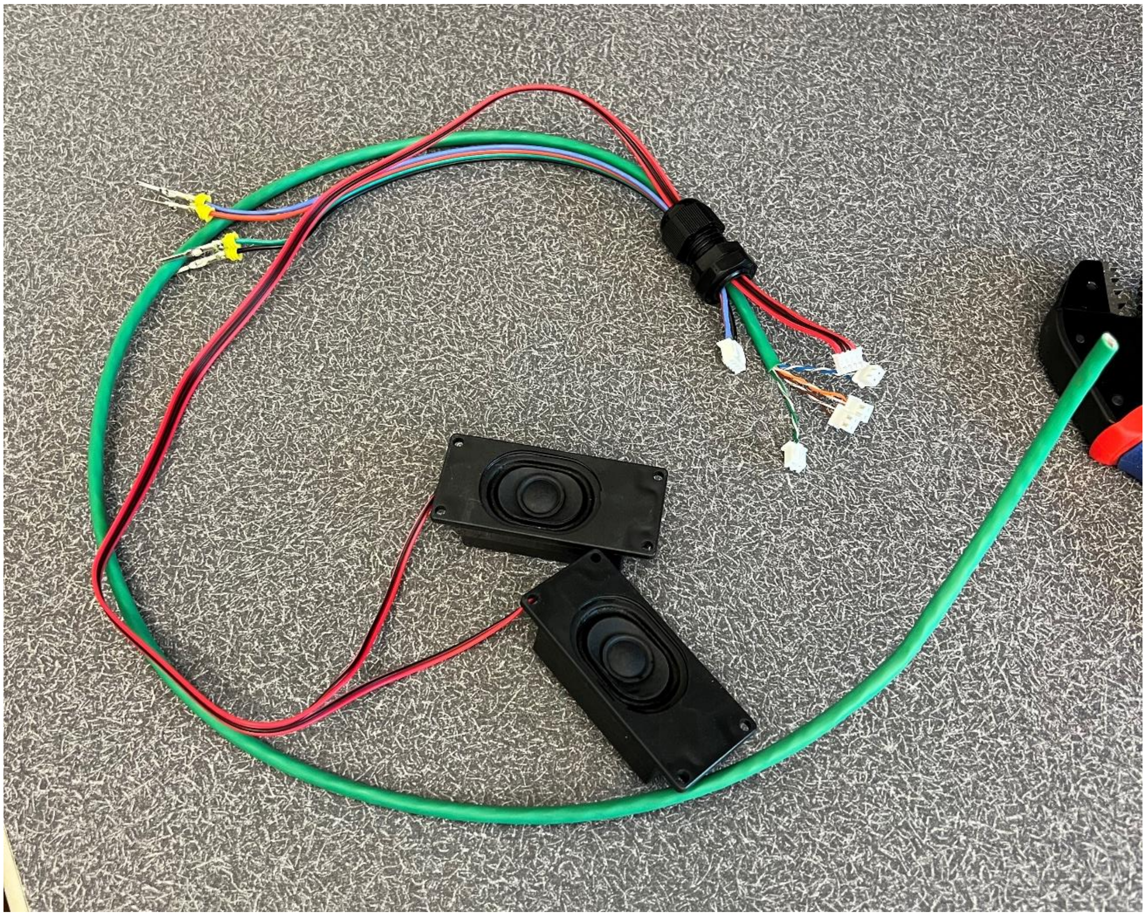


Figure 100 – Right gland with wires fed through.

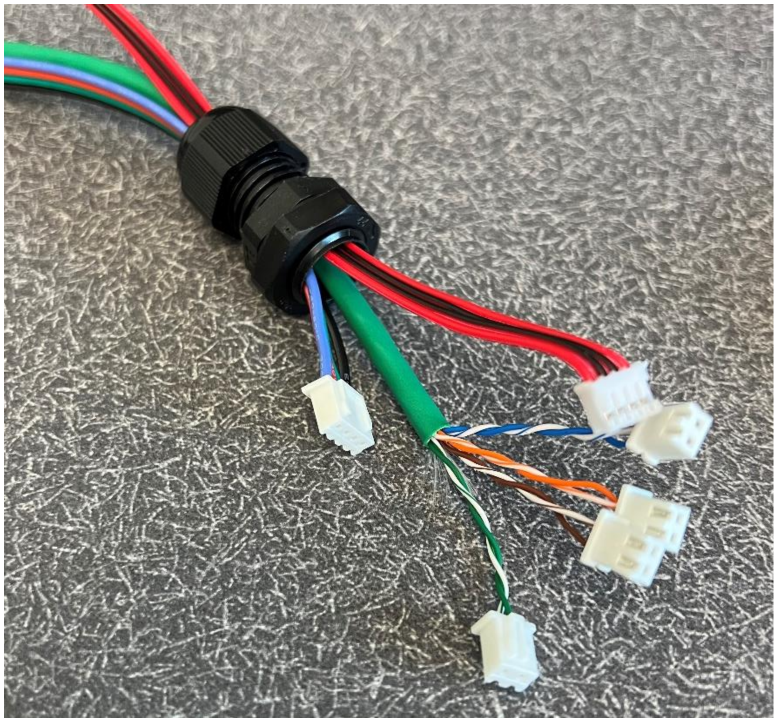


Figure 101 – Close-up of right gland, pre-installation.

Next, the wires need to be inserted into the box and the gland needs to be installed. To do so, take the nut off the gland. If you did the last steps with the nut already off, you can skip this step. The wires should look like they do in Figure 102.

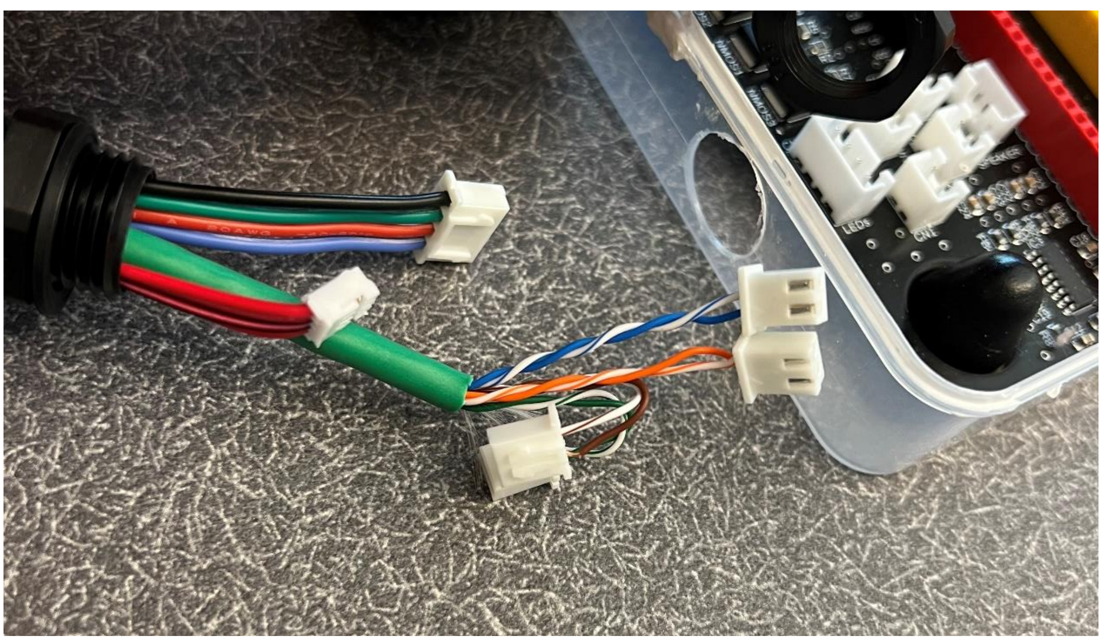


Figure 102 – Wires with nut removed pre-insertion.

Then, feed all four wires through the enclosure’s right hole. Feed the wires back through the nut. Then, starting with the LEDs, begin plugging in JST connectors. After the LEDs, plug in the switch connectors followed by the UART connectors, being careful not to break the connections. The ethernet cable is fragile and can easily break, requiring a quick fix to be done by removing the crimp and redoing it. Lastly, plug in the speaker connector. Figure 103 shows the final installation configuration, with all connectors attached. Please also consult this figure if lost on how the color convention should work at this step.

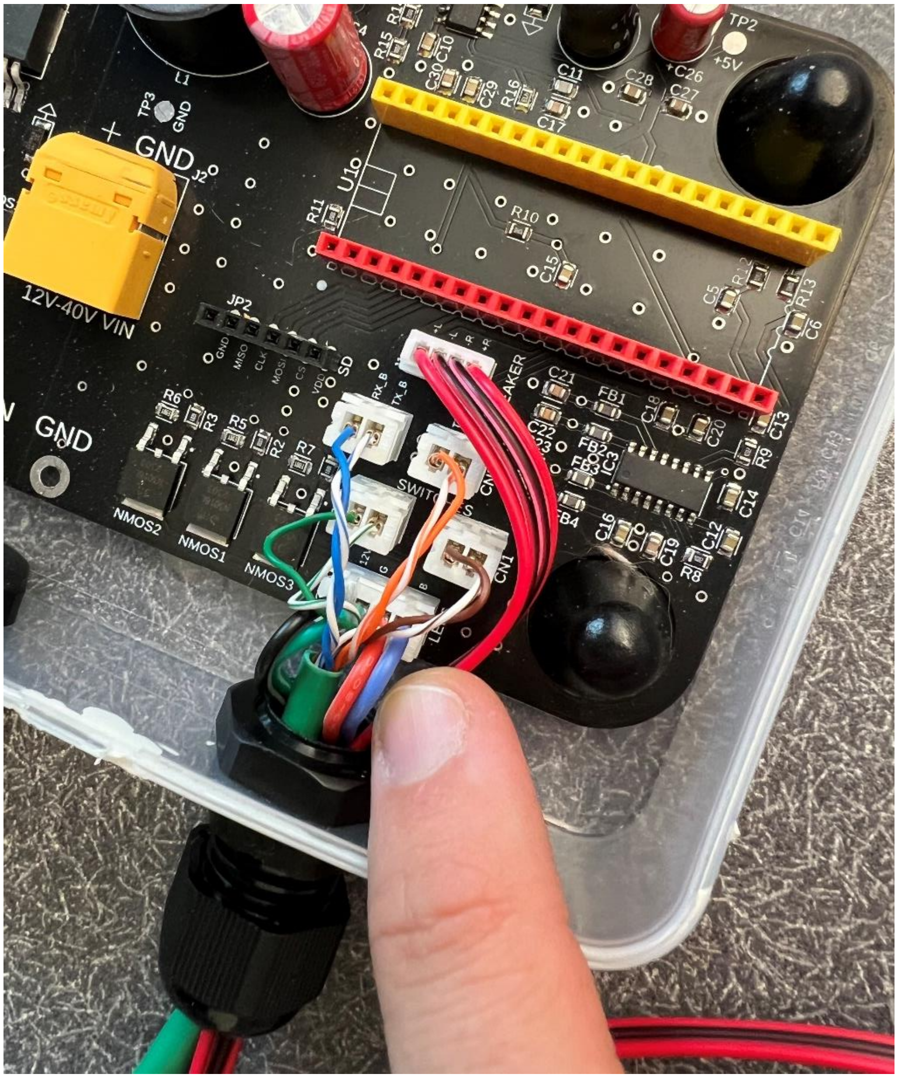


Figure 103 – Right gland fully installed.

Begin by inserting the 6” tail of the power wire through the left gland and feeding it into the box. Once this is done, strip a short section of the wire, place heat shrink casing onto the wire, and solder it to a female XT-60 connector. Once the XT-60 connector is fully soldered, move the heat shrink casing to cover the joints and use a hot air gun to shrink it. Since this is inside the enclosure, this heat shrink casing does not need to be waterproof. Plug the XT-60 into the board connector. Seal the glands with sealant and tighten them. The final enclosure should look like Figure 104.

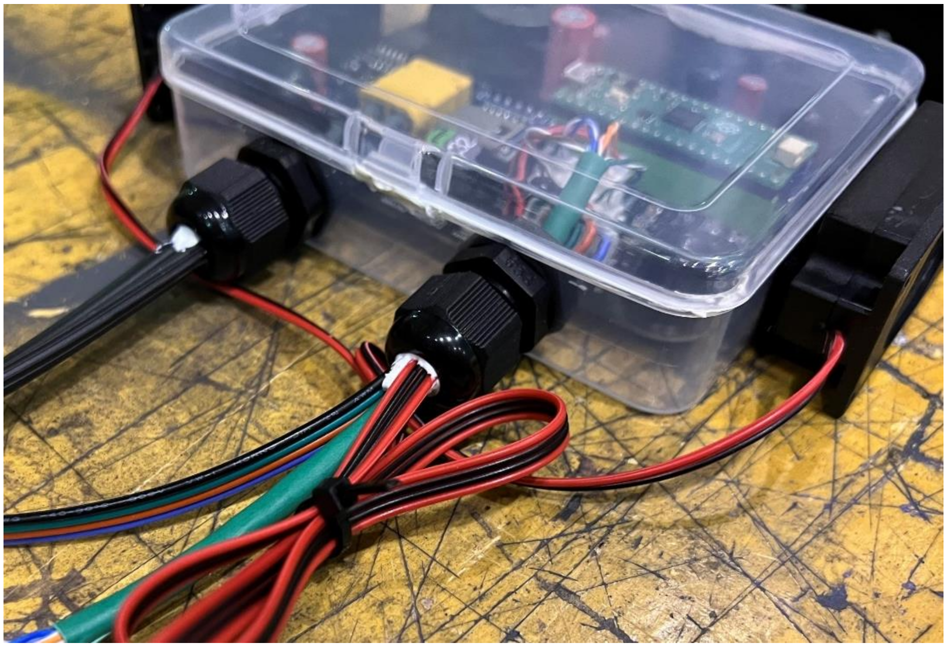


Figure 104 – Finished electronics enclosure with sealed glands.

3.5 Final Touches

For information about programming the Pico, consult Section 4.1. With two .75”x2” strips of Velcro, attach the speakers to the sides of the enclosure.

The SD card’s breakout board should have the headers it comes with soldered to itself and then inserted into the Final Board’s female receptacle for the SD card. Then, insert the SD card into the SD card breakout board. This should already have all of the “sounds” folder on it. For a board with a correctly installed Pico, speakers, and SD card board, see Figure 105.

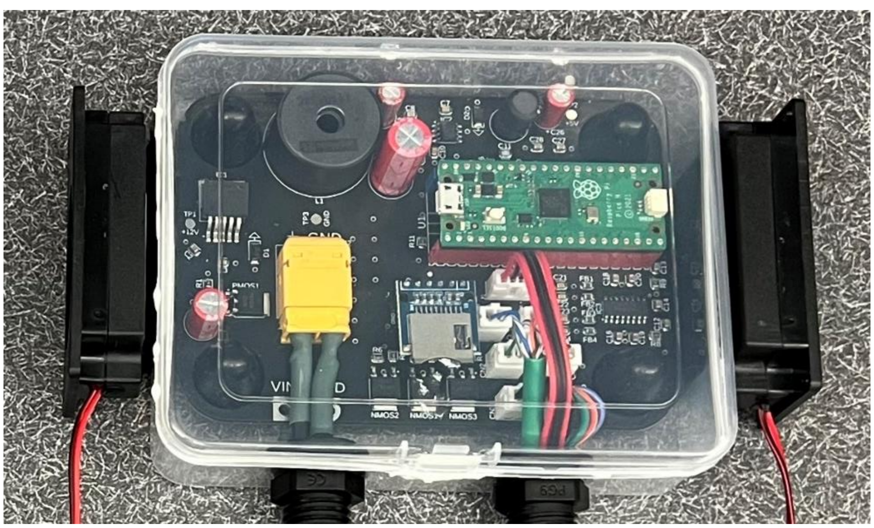


Figure 105 – Enclosure with correctly installed Pico and SD card holder.

Strip away a section of the jacket of the ethernet cable to the length shown in the wiring diagram Figure 96. Strip a small section and solder the blue and orange pairs to switches with heat shrink casing to cover the exposed wire. The switches should have the solid color wire soldered to one terminal and the striped wire of the same color soldered to the other terminal. Repeat this for both switches and heat up the heat shrink casing to cover the terminals.

Repeat the same steps for the communication connectors. The male connector should be soldered to the blue pair of wires and the female (has the yellow gasket) to the orange pair of wires. With both of these connectors, make sure that the solid color wire connects to the black wire of the connector.

Figure 106 has a picture of the completed switches and communication connectors. This completes the electronics enclosure.

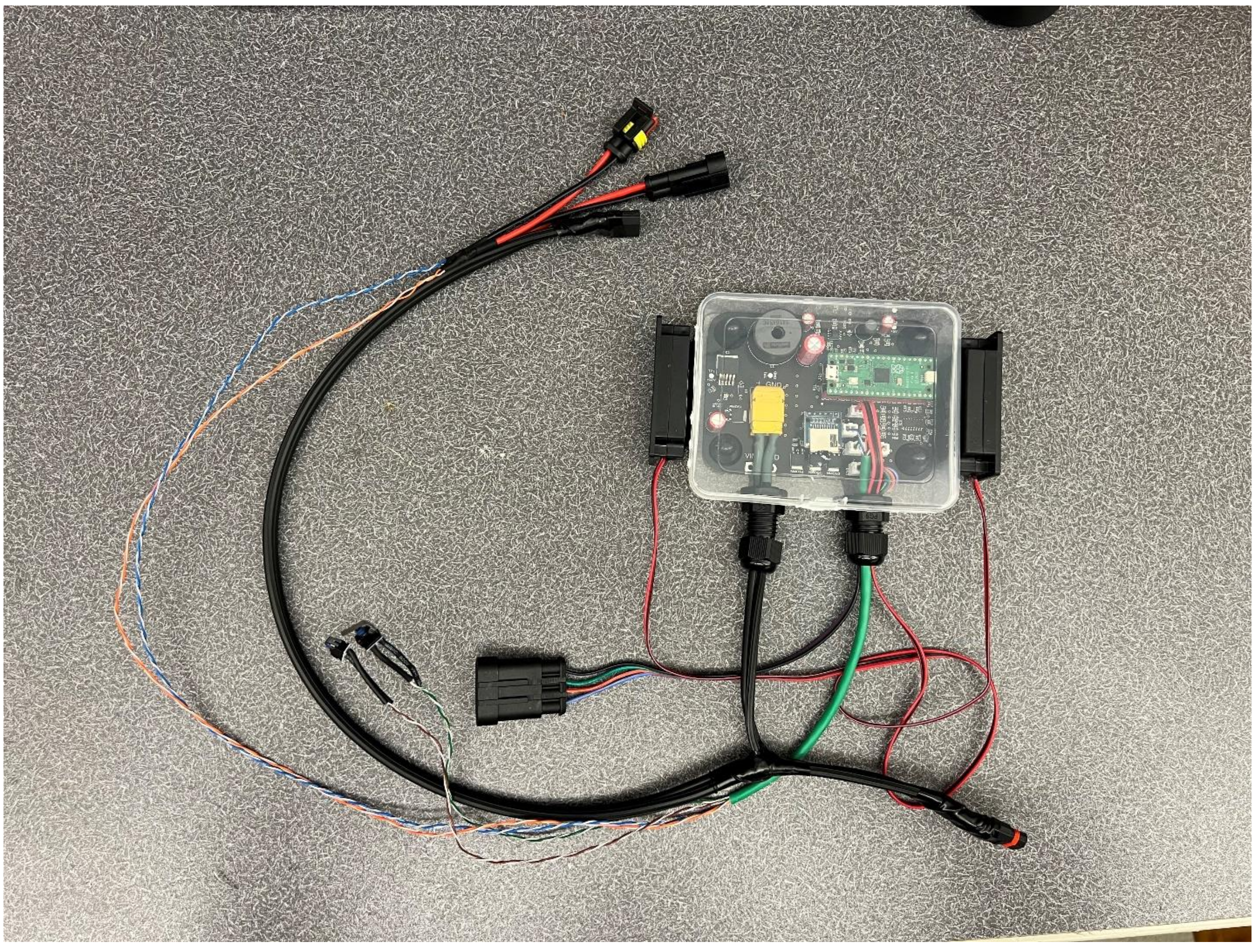


Figure 106 – Enclosure with finished connectors.

4 Programming the Electronics

This section's main goal is to give to common user enough information to be able to properly understand the tile's coding structure, reprogram the tile's song using the user interface, update the tiles with the new song, and to increase the tile's note and song capabilities in the future.

4.1 Tile Code Basics

At the simplest level, only 5 python files (ending in '.py') are required to be on the tiles for them to function properly. The files with brief descriptions are as follows:

- **ledstrip.py –** contains functions that help control the LEDs.
- **speaker.py –** contains functions that help operate the speakers.
- **dataFunctions.py –** contains functions that help transmit data back and forth between tiles as well as initially read the 'controller.txt' file (to be covered later).
- **Lookup.py –** contains information to transmit readable text into encoded numbers.
- **main.py –** uses all other files to create a state machine and fully run the tiles.

If a new Pico needs to be programmed, here are the steps to follow:

1. Go to the software '.zip' package that the team has provided you. Unzip the file to get all of it's contents. This folder will now be referred to as the software package.
2. Plug in the Raspberry Pi Pico to your computer with a micro-USB cable capable of transmitting data and providing power. The team has provided one such cable that does both of these functions, as seen below in Figure 107. The Pico device should show up as a flash-drive type device called "RPI-RP2 (D:)".

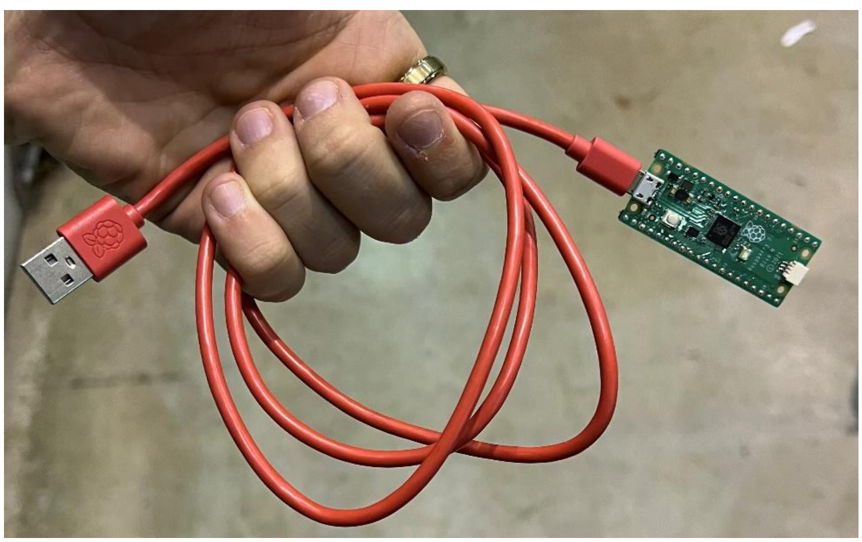


Figure 107 – Pico plugged into provided micro-USB cable.

3. Drag and drop the "adafruit-circuitpython-raspberry_pi_pico-en_US-8.2.6.uf2" file from the "Pico Coding" folder in software package onto this device. This "flashes" the device with the Circuit Python infrastructure and libraries to run the team's code. The device should disappear and show back up as a flash device now called "CIRCUITPY (D:)".

4. Drag and drop all the files inside of the “Pico Coding>State Machine” folder to the new CIRCUITPY device. These should be the 5 files that were discussed in this section. Your device should now look like Figure 108.

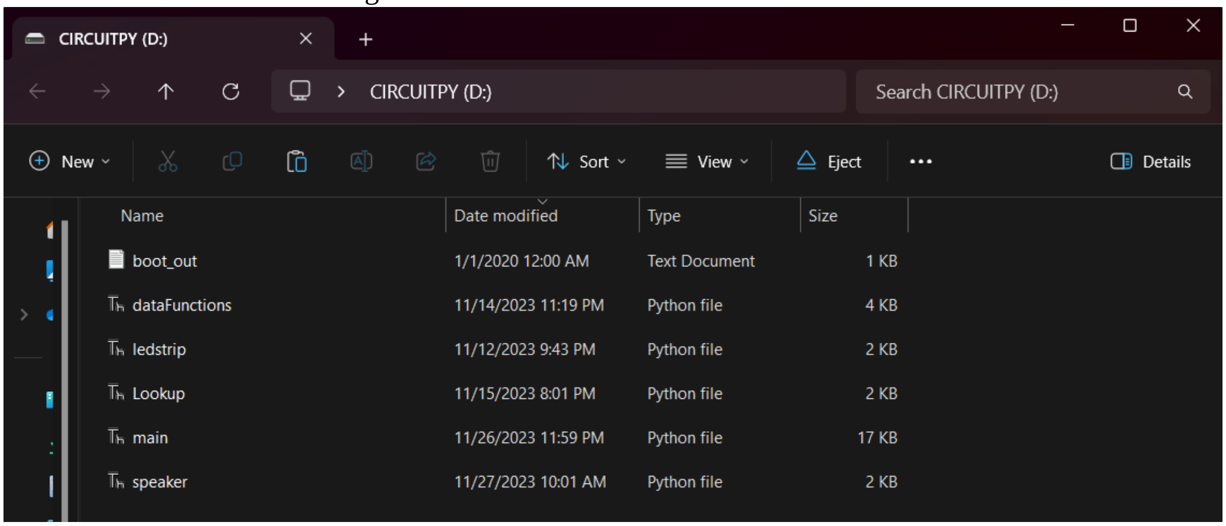


Figure 108 – CIRCUITPY device file folder once properly set up.

5. This is enough code to run any tile. Place it into the 20 pin colorful headers on the electronics board as seen done in Figure 109 to Figure 111. You may skip this step if you were only updating code on a Pico that was already seated in an enclosure.

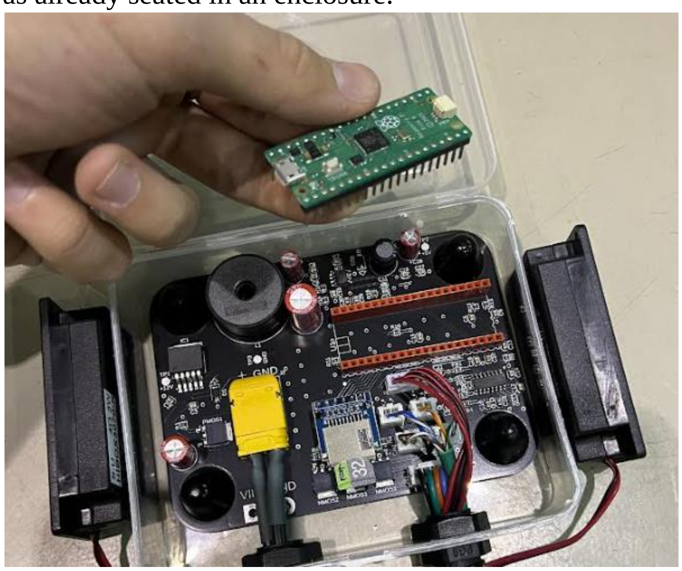


Figure 109 – Locate the Pico and open electronics enclosure. Be sure to have the Pico's micro-USB port pointing towards the left when oriented as so.

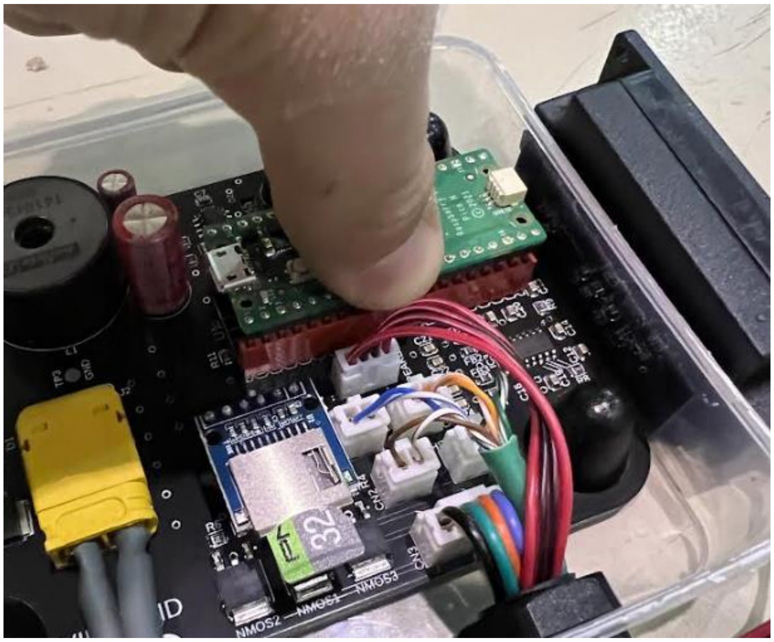


Figure 110 – Insert the Pico's pins into the headers below. Once located, push down firmly on the Pico's sides as seen above.

Figure 111 – Fully seated Pico. The enclosure is now ready for insertion into a tile.

6. Ensure that the SD card in the tile has the full "sounds" folder from the software package. If not, drag and drop the whole "sounds" folder onto the SD card. This is the only folder that should be present in the SD card. It is okay if a "note.txt" is present, or even a "controller.txt" for the controller tile (covered later).

This is enough code for any tile to participate in the line of tiles. For how to set up a tile to be a Controller tile, please read the following two sections. For more information on how the code works and operates, please reference the main software/coding sections of the FDR along with the associated coding appendix.

4.2 Changing the Song Details

This section will walk through changing the song that the tiles play along with their color scheme. Please follow the instructions below:

1. As in the last section, have the software package unzipped.
2. In the "User Interface" folder, launch the "Interface.exe" file.
3. The first prompt that shows up is for selecting the directory in which "Lookup.py" is located. From the previous section, it is known that this is the "Pico Coding>State Machine" folder, so navigate there using the "Select File Directory" button. Once done, continue by pressing "OK".
4. At this point, the window in Figure 112 should have popped up. Take some time to digest the available fields.

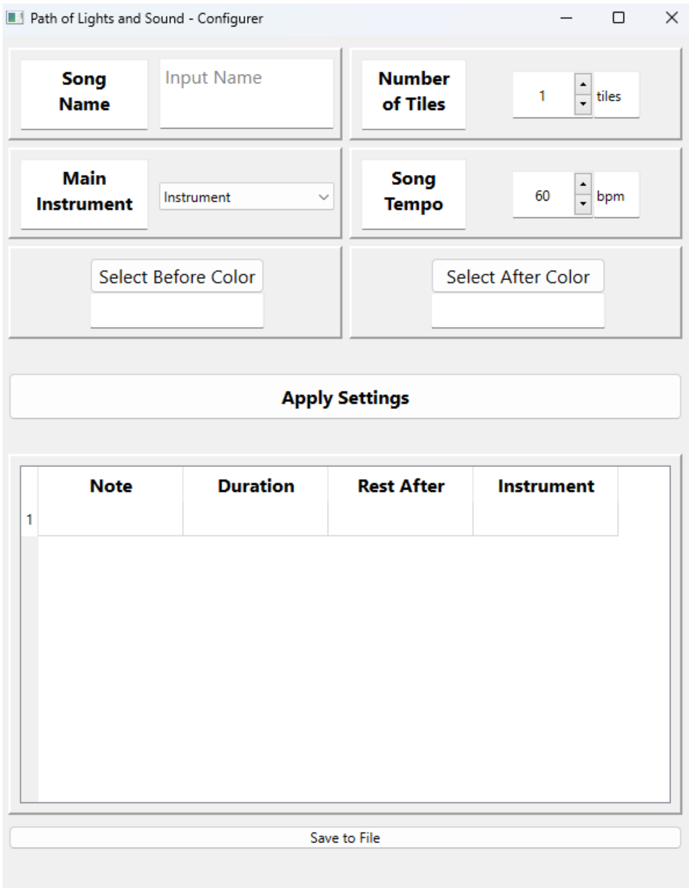
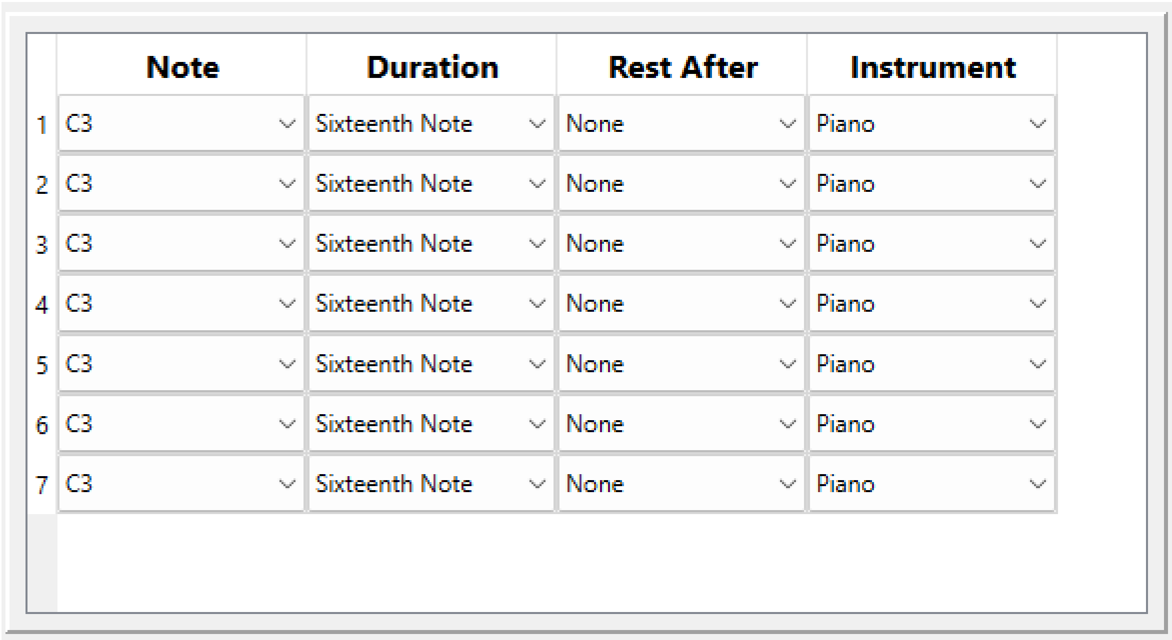


Figure 112 – User Interface pop-up window.

5. Go through the top fields (above the "Apply Settings" button) and select your song's inputs.
   - **Song Name –** Can be any designator for the song you're creating.
   - **Number of Tiles** – How many tiles you will be using for the song, or how many notes the song has.
   - **Main Instrument** – The most common instrument in the song. Individual note's instruments can be changed later.
   - **Song Tempo –** How fast the song should be played in beats per minute (60 to 200).
   - **Before Color –** The color the tile is while awaiting a step.
   - **After Color –** The color the tile is after being stepped on.

6. Once the fields from the last step are filled out, click “Apply Settings”. You should now see the bottom section populated with something like Figure 113. If you change anything in the top-half again, you need to apply the settings again and it will reset your progress in the bottom section.



	Note	Duration	Rest After	Instrument
1	C3	Sixteenth Note	None	Piano
2	C3	Sixteenth Note	None	Piano
3	C3	Sixteenth Note	None	Piano
4	C3	Sixteenth Note	None	Piano
5	C3	Sixteenth Note	None	Piano
6	C3	Sixteenth Note	None	Piano
7	C3	Sixteenth Note	None	Piano

Figure 113 – Bottom section of the User Interface once settings are applied.

7. Go through and change every note to have the correct parameters. WARNING: While “Sine Wave” and “Piano” instruments have the full 3 octaves of notes present, “Glockenspiel” only has the notes necessary to play Jingle Bells. It is wise to avoid this instrument before providing more sound ‘.wav’ files.
    - **Note –** The note to be played, from C3 to C6. Only sharps are labeled, so adjust flats to their corresponding sharp.
    - **Duration –** How long the note is to play. This is assuming that one beat of the song is a quarter note.
    - **Rest After** – Likely none, but this is how long of a gap there is until the next note.
    - **Instrument** – The instrument to play the note. This is auto-populated with the default instrument.
8. Be sure that the song information is correct as there is no way to load a file in the user interface, so if you made a mistake you will have to restart. Once content with the song settings, click “Save to File” at the bottom of the User Interface. This automatically generates and opens a “controller.txt” file. Only the top line of this file is readable. Feel free to rename this file to something more descriptive, but when it goes on the controller tile’s SD card, it NEEDS to be named “controller.txt”. It is automatically saved in the same folder as the “Interface.exe” file.
9. Place the “controller.txt” file on the controller tile’s SD card. Make sure there is only controller file on the SD card and only one controller tile in the path.

Again, ensure that only one tile in the path has a “controller.txt” file on its SD card. If two tiles both have a “controller.txt” file, the tiles will not work as intended. The best way to ensure that this is always the case is to only ever replace the controller tile’s “controller.txt” file, as by default this is the only tile that has the text file.

4.3 Initiating Reprogramming

Now that the controller tile has the "controller.txt" file, the rest of the tiles in the path can have their note information updated. This is done by stepping on the controller tile while switching the system on. If done correctly, a green light should progress down the path before flashing back to the start when completed successfully. If the green light gets stuck somewhere, there's potentially an error with the wiring or with one of the electronics boards itself.

After you do this reprogramming once, you will not have to do it for the same configuration of tiles/electronics enclosures again. If anything moves around, you will notice that the song is not playing correctly and reprogramming will need to occur once more. This is because each tile saves a "note.txt" file to their SD card once they receive a program. When the system restarts, this tile is still on that specific tile. If no "note.txt" is present on the SD card, the tile has a default set of settings that will be played, though the path should be reprogrammed at this point.

The reason for this reprogramming method is so that the user only has to update one tile to program in a new song. It would be especially annoying to have to open up every single tile, take out their SD cards, put a specific note on it, and then replace everything every single time that a new song is desired.

This method also highlights why there can be only one controller tile, as having two will initiate reprogramming somewhere in the middle of the path, at which point those tiles will not be receiving the correct song information from the head controller.

4.4 Increasing Tile Capability

If a day comes in which you would like to add more sound types, notes, or even chords, this section covers how to do so. But first, a word on how the tile's encoding works with "Lookup.py". "Lookup.py" contains coding dictionaries to translate text strings into numbers. These numbers are easily communicated between tiles in single bytes, whereas text strings are more unpredictable in length and take up more space.

Through "Lookup.py", both the User Interface and tiles themselves are able to convert between text and number for notes, durations, and instruments. The basics to adding more capability is adding more lookup pairs to "Lookup.py". After that, if new notes or instruments are added then sound files need to be made and added to the "sounds" folder. This means "Lookup.py" needs to be updated on every Pico and the "sounds" folder needs to be updated on every SD card. This is quite time intensive, though if you want more capabilities then you must do so.

Here are the steps to follow:

1. Open "Lookup.py". Reminder that it is located under the "Pico Coding>State Machine" folder in the software package.
2. TO ADD AN INSTRUMENT, add a comma after the last instrument in the list, create a new line indexed 1 after the previous number, add a colon, and put in text for your instrument name. For an example, see Figure 114. The instrument name must contain no spaces. A maximum of 256 instruments can be present.

```
instrument = { 1: "Piano",
               2: "Glockenspiel",
               3: "SineWave",
               4: "NewInstrument"}
```

Figure 114 – Example new instrument being added to "Lookup.py".

3. TO ADD A NOTE, add a comma after the last note in the list, create a new line and have its number be 1 more than the previous number, add a colon, and put in text for your note name. This could also be a chord. No spaces can be used. See Figure 115 for an example. The order of the names does not matter, though there must not be any duplicate names or numbers. A maximum of 256 notes/sound names can be present.

```
32: "Gs5",
33: "A5",
34: "As5",
35: "B5",
36: "C6",
37: "D6",
38: "ACsE"}
```

Figure 115 – Adding new notes to "Lookup.py".

4. Once the desired notes or instruments have been added, 5 second long sound files must be created, specifically ".wav" files. They can be longer than 5 seconds but must not be shorter. Once sound files have been obtained, they must be saved as mono tracks, 16-bit PCM, and sampled at 16-44kHz. Note that the tiles themselves do not have volume control, so make sure the sound files are loud enough but don't clip. See the Piano files in the "sounds" folder for reference.
5. Name the '.wav' file using the following convention: **instrument + sound name.** For example, a Piano playing a C4 would be named "PianoC4.wav". Once this is done, put the file in the sounds folder in a new folder that is solely the instrument name. That means the above file would be saved under "Piano>PianoC4.wav". Do this for all files.
6. Once completed, the Pico's must be updated with the new "Lookup.py" file through micro-USB and the SD cards must receive the new "sounds" folder. After this is done, tile capability has been increased and can be used in the User Interface.

Hopefully this is enough to understand how to add more sounds and instruments to the tiles. It is of best interest to use good quality sound files as any ambient noise or clicking will be reproduced by the speakers and cause popping near the end of files.

If any questions arise, specifically reach out to Sean Wahl at seanwahl19@gmail.com. He is aware of issues that may come up in producing sound files or using the User Interface and can help get around them if necessary. However, you are urged to please try to consult other resources before reaching out to Sean.

5 Setup

This section will help the user through different stages of the tiles' setup. This includes initially setting the tiles up, tearing them down, and taking out or inserting electronics in case of rain.

5.1 Initial Setup

Step 1: Gather the required materials and tools.
Tools needed:

- Hammer or handheld rubber mallet
- Screwdriver
- Leather or gardening gloves (for handling grip tape frequently)

Materials needed:

- A matching number of tiles and electronics (may be inside the tiles)
    - Number is based off of the notes in the desired song
    - 1 tile must be a "controller" tile. See previous section for setting up a controller tile
- Power supply
- Stakes (may be inside of tiles)

An example pile of materials is presented in Figure 116. Please take these materials to a suitable location. A suitable location is defined as a dirt or grass area that is flat and won't have water easily pooling under the tiles. This location also needs access to a standard wall outlet (110V).

Figure 116 – Example materials for tile setup. This example will be setting up 6 tiles. This is in the suitable location of a flat, grass field (with outlet nearby).

Step 2: Place power supply near projected path start.
Plug the power supply into a wall outlet and stake down if necessary. Ensure that the green switch is flipped to the "O" position, meaning off. Consult Figure 117 - Figure 119 for clarification.

Figure 117 – Power supply near projected start of path. Plug in the end of the yellow cable to a wall outlet.

Figure 118 – Holes for staking power supply to the ground.

Figure 119 – Power supply's green switch turned to the "O" position. The power supply should always be off when setting up or tearing down tiles.

Step 3: Locate and place the "controller" tile.
This tile can program all of the other tiles' notes, but it must be the first tile in the path to do so. The "controller" tile should be clearly labeled with some sort of marking. If not, it will be clearly marked internally, and every tile will need to be opened to determine this. See Figure 120 for an example of the controller tile. Place it in range of the power supply's black cable. See Figure 121 for reference.

Figure 120 – Labeled "controller" tile.

Figure 121 – "Controller" tile set up in range of the power supply's black cable. After placing the "controller" tile here, the external label may be removed, though if done, needs to be replaced upon teardown.

Step 4: Layout the rest of the path's shape.
Following the controller tile, place down the rest of the tiles following. This order does not matter. Alternate tile direction to go straight or maintain tile direction to turn. See Figure 122 for reference. Maintain around a 1/4-inch gap between tile edges. See Figure 123 for example tile spacing.

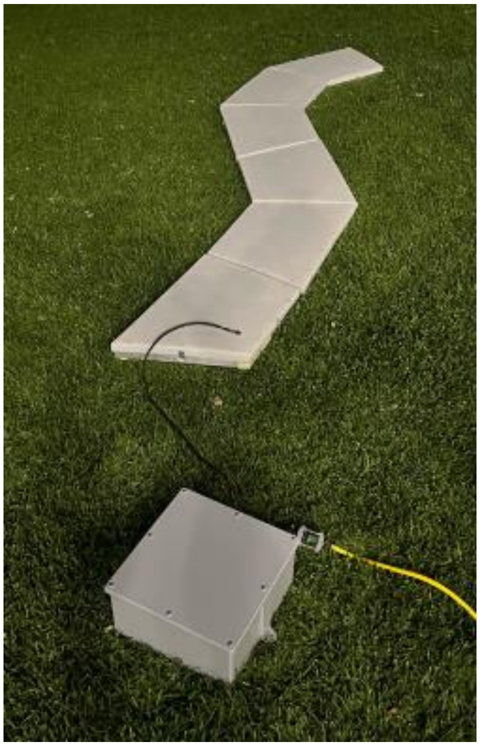


Figure 122 – Example path layout. Note how the path is able to turn or go straight.

Figure 123 – Example tile spacing. Please keep tile spacing to ~1/4-inch. If much larger, power cables may not be able to reach between tiles.

Step 5: Remove all quick-release pins from the sides.

Each tile has 3 quick-release pins in their sides. Remove all 3, but do not lose track of them. It is recommended to place the removed pins in or on the tile. Take the lid off every tile and place it next to the tile. Do not put grip tape face down on dirty surfaces. Storing the quick-release pins inside of the tiles' bases can help to keep track of them. See Figure 124 - Figure 126 for help.

Figure 124 – Quick-release pin. Pull on the ring to remove the pin.

Figure 125 – Path with all lids removed. Place lids face up if the ground is dirty to avoid getting the grip tape unnecessarily dirty. Leave space next to every tile for working room.

Figure 126 – Quick-release pins stored in tile's base. DO NOT LOSE THESE.

Step 6: Stake down the tiles.

Take the box of stakes seen in Figure 127. Place two stakes through the holes in at least 2 opposite corners of the tile, using Figure 128 as reference. It is possible to use stakes in all 4 corners if desired. Again, make sure that the tiles' alignment in relation to one another stays the same through the staking process. Use a hammer or rubber mallet to drive stakes into the ground. Ensure the stakes are at least halfway below the tile's height (as seen in Figure 129).

Figure 127 – Box of stakes.

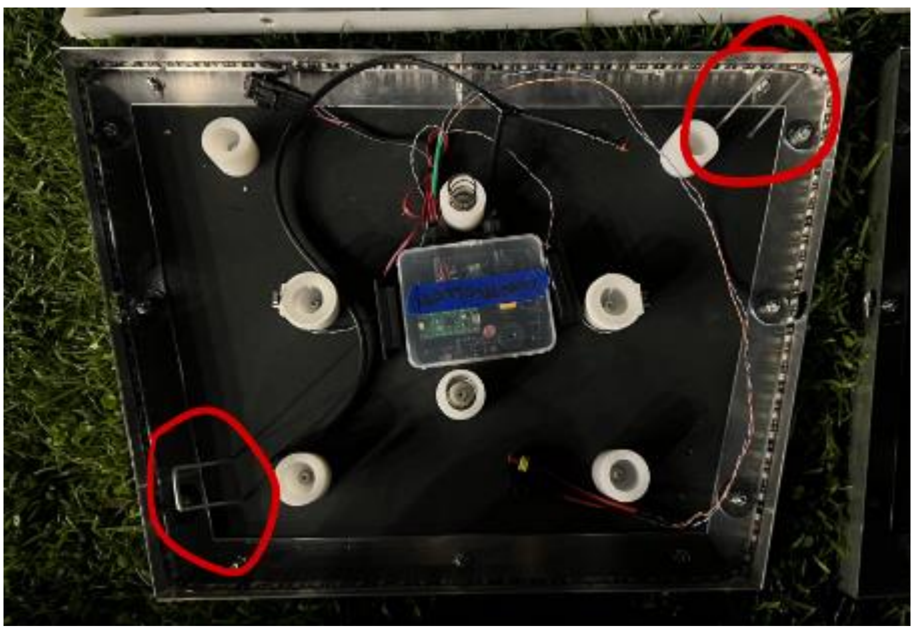


Figure 128 – Minimum tile stake-down locations. At least 2 stakes need to be used in the red circles for proper tile fixation.

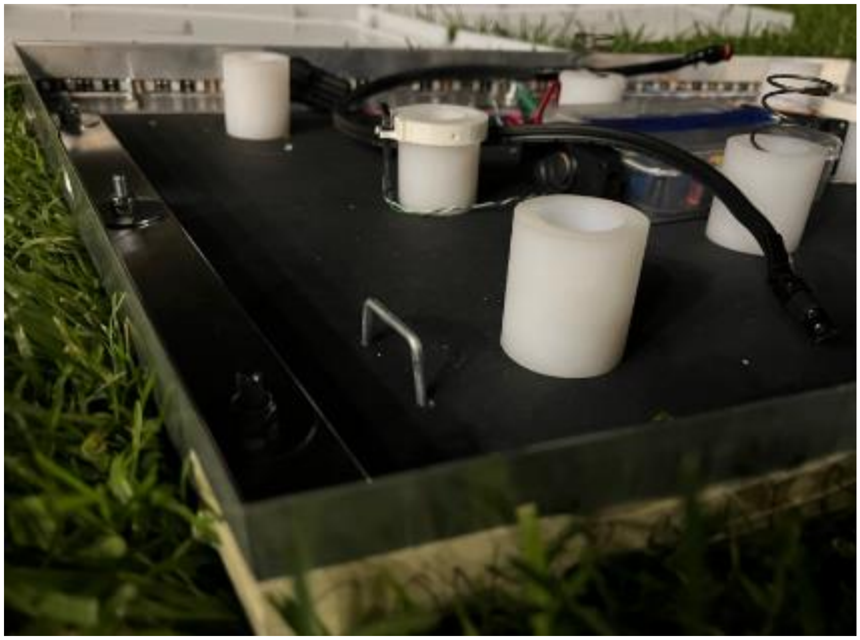


Figure 129 – Ensure stakes are more than halfway below the tile's height.

Step 7: Plug in power to the "controller" tile.
Take the power supply's black cord and feed it through the first tile's side wire hole. Plug it into the first tile's long power cable (black with no visible red on the connector). See Figure 130 for an example. Also, it is of note that the "controller" tile's blue cable does not get plugged in anywhere.

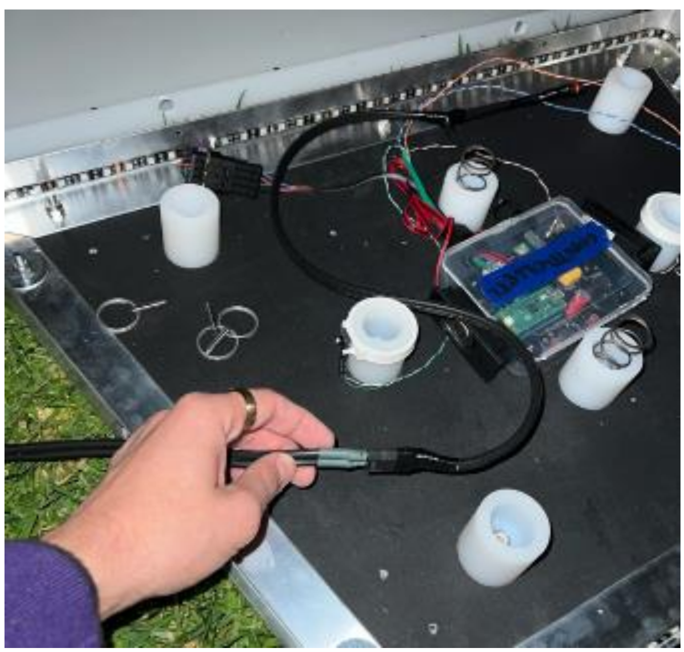


Figure 130 – Power supply's black cable plugged into the "controller" tile's longer black cable. Tuck this connection inside of the first tile as seen above.

Step 8: Plug in the remainder of the tiles' connections.
The best way to go about this is in groups of 2, all the way down the path. See Figure 131 for the connections that need to be made between the pair of tiles. See the following pictures in this step for further guidance on making these connections.

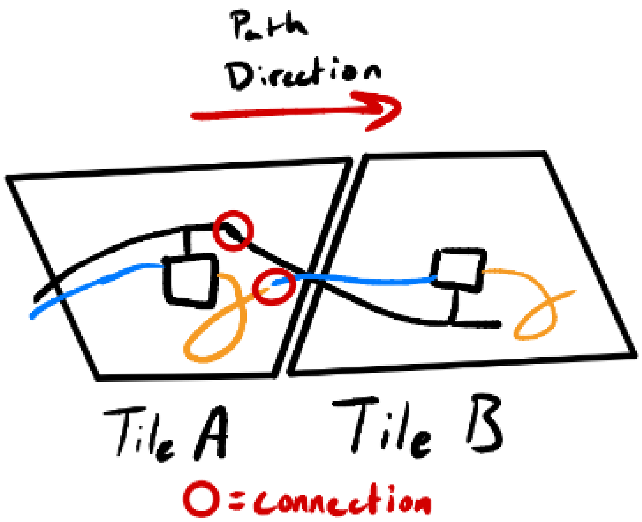


Figure 131 – Connections that need to be made between tile pairs. A tile pair is simply defined as two adjacent tiles. The tile closer to the start of the path is "Tile A" and the one further is "Tile B." Tile A's orange cable needs to the connect to Tile B's blue cable. Tile A's short, black power cable needs to be connected to Tile B's long, black power cable. Wires should be fed through the holes on the side of the tiles and connections should remain inside tiles, not leaving connectors in the gap between tiles.

Figure 132 – First, feed through Tile B's (on the right) blue and long, black cable through both wire holes, meeting on their adjacent side. The blue connector should pass through first due to its size.

Figure 133 – Plug in Tile A's orange wire/connector to Tile B's blue wire/connector. This should occur within Tile A.

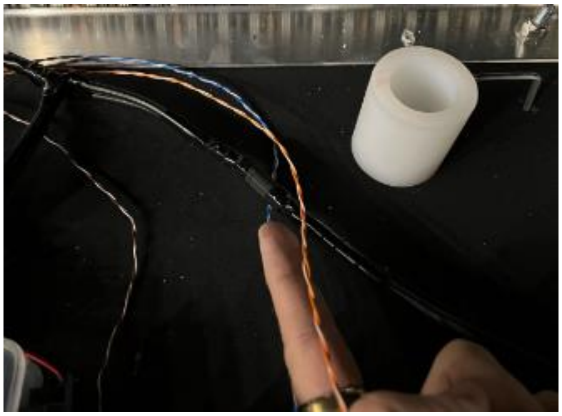


Figure 134 – Plug in Tile A's short, black power cable to Tile B's long, black power cable. This should occur within Tile A.

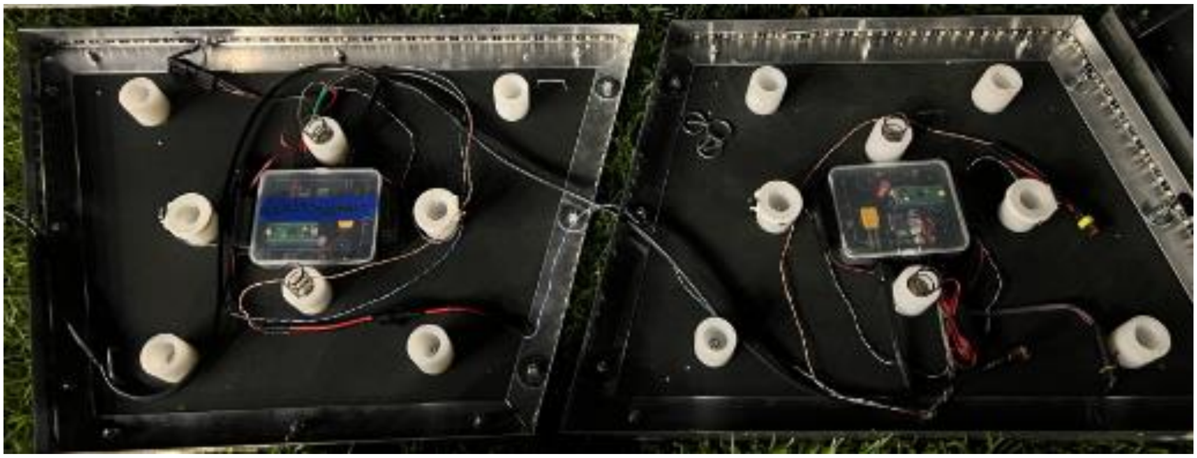


Figure 135 – Tile A and B with required connections made.

Once one pair of adjacent tiles has been connected, shift down one tile, and repeat this process all the way down the path of tiles. The final tile will have an orange cable that does not need to connect to anything.

Step 9: Put the tile lids back on.
Before doing this step, **move all wires out of the way of supports**. This means laying out wires so that they sit on the bottom of the tile. If difficult, masking tape can be used to aid in this effort. Once the wires are out of the way of the internal supports, grab the quick-release pins from out of the tile. The lid can be placed back on, and the pins inserted. On some tiles the lid can be difficult to place over the bottom assembly. The following are a couple of tips to help with this task.

- Watch out for pinching. Don't place your finger up against the interface between the lid and the base while applying pressure to the lid.
- Use two people. This allows for all four corners to be properly lined up and held flush before being pushed down over the base.
- If certain edges don't seem to fit, pull outwards on the top side of the tile while pushing inwards on the aluminum siding beneath with your thumb or forefingers, as in Figure 136.

All lids are known to be able to fit on their respective tiles, so don't give up hope if a lid doesn't instantly fit! Once the lids are back on, the quick-release pins can be replaced, shown in Figure 137.

Figure 136 – Lid placement technique. Usually performed with two hands, the thumb is pushing inwards on the aluminum while the forefinger is pushing downwards. When performed with two hands, this move is especially effective on corners.

Figure 137 – Reinserting the quick-release pins. Make sure the pin goes through both the white top panel and metal hole. Ensure all 3 are replaced on every tile.

Step 10: Powering up.
Now that the path has been laid out, fixed down, and connected, the tiles are ready to use. Flip the green power switch on the power supply to the "I" position, or on. The switch should light up green as in Figure 138. If the switch does not turn green or the tiles seemingly don't have power, it is possible that the GFCI outlet inside of the large gray power supply box has tripped. To reset it, open the box by removing the 8 screws and press the reset button on the outlet. If the problem persists, it is possible that either other connections are undone or that there is a larger problem at hand.

Figure 138 – Power supply's green switch flipped to the on position.

IF THIS IS YOUR FIRST TIME USING THESE TILES IN THIS CONFIGURATION, ACTIVATE THE FIRST TILE BY STANDING ON IT WHILE SIMULTANIOUSLY FLIPPING THE SWITCH. This allows the tiles to enter programming mode and will distribute the song (contained on the first tile) to all the other tiles. You will know you have entered programming mode if the tiles progressively light up green down the path, then return in a flash to the first tile. You may stop stepping on the first tile once this process starts. The tiles are ready to use when the first tile starts slowly brightening and darkening the desired starting color. If this path configuration with these tiles in a specific order remains the same for the duration of the event, this mode will never need to be entered again. If electronics or tiles are moved around, then the tiles will need to be reprogrammed once again.

5.2 Electronics Removal

In the case that rain is coming or that the tiles are around pooling water, the electronics should be removed. To do so, follow the procedure below. Alternatively, the entire path can be taken down as done so in Section 5.4.

Step 1: Turn off tile power.
To do this, flip the green switch on the power supply box to the "O," or off, position. The green light should turn off. See Figure 139 for a visual.

Figure 139 – Power supply switch in the off position.

Step 2: Remove the quick-release pins and take off the tile lids.
As in the setup procedure, take out all the quick-release pins and remove the tile tops. Examples of each are shown in Figure 140 and Figure 141. Please keep track of the quick-release pins by placing them in the tile's base once removed.

Figure 140 – Removing the quick-release pins.

Figure 141 – Tiles with lids removed and placed aside.

Step 3: Disconnect all connections.
The next step is to disconnect the LED connectors, shown in Figure 142, the inter-tile communication connection, shown in Figure 143, and the power supply connection, shown in Figure 144.

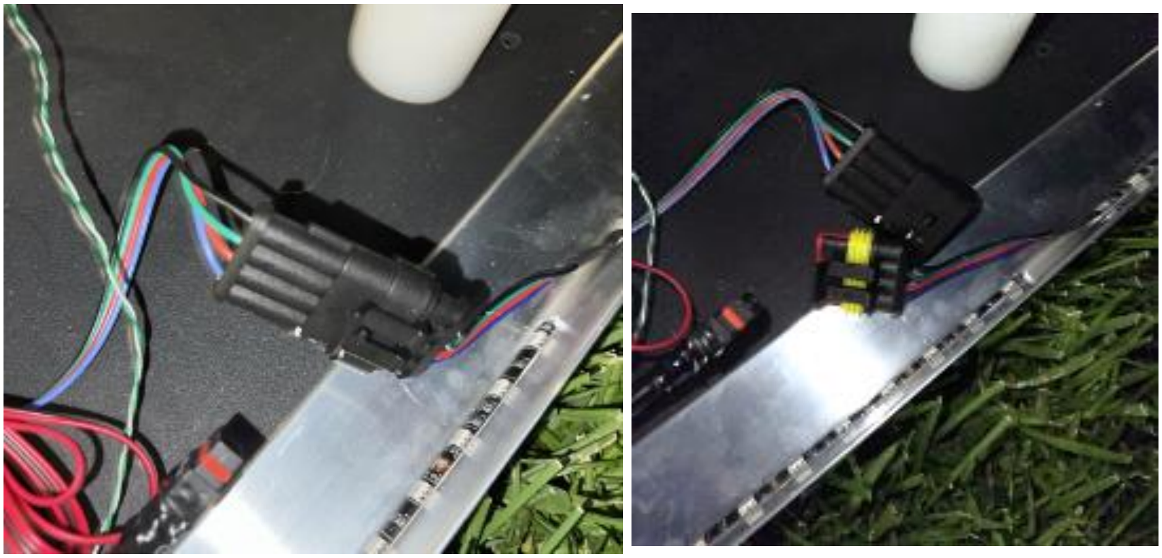


Figure 142 – Unplug LEDs.

To remove the LED connector, the latch must be pushed up, while simultaneously pulling apart the male and female connectors. Using one hand to grip the female end, while pushing the latch and male end away with your thumb and forefingers was found to be a very effective way to separate these components. Do NOT pull on the wires. Only pull or push on the black plastic.

Figure 143 – Unplug orange/blue connection and feed the blue wire back through the side wire hole into its original tile.

This step is very similar to the previous one. To disconnect the inter-tile communication, the orange and blue wires, a latch must simultaneously be pushed up while pulling/pushing the male and female parts of the connector away from one another. Again, using the one hand with the thumb and forefingers to push up the latch and push the male end away from the female end was found to be most effective. Once disconnected, make sure to route the blue wire back through the hole in the aluminum angle and back into its original tile. Do NOT pull on the wires.

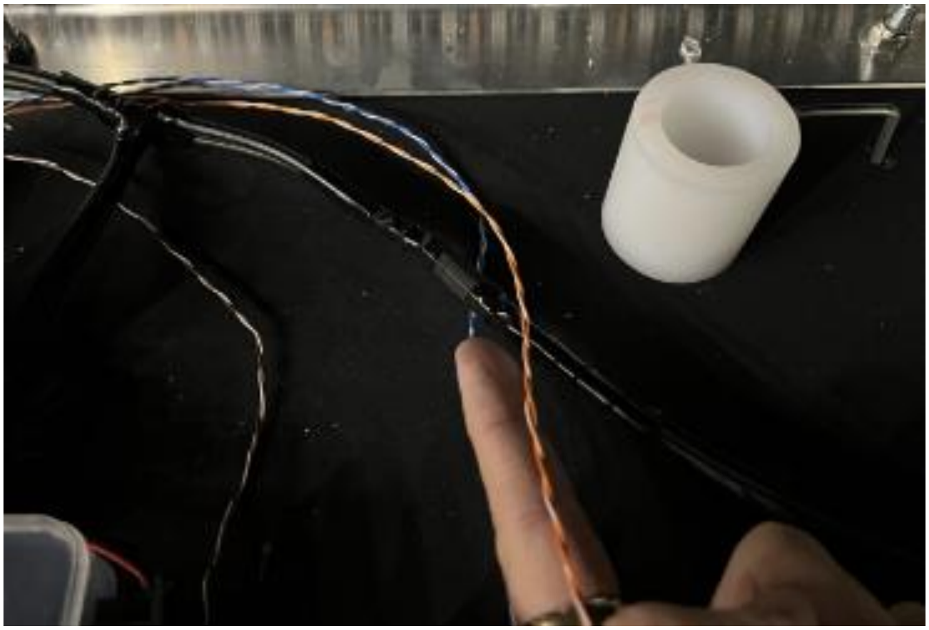


Figure 144- Unplug the power connector and feed it back through the side wire hole into its original tile. Don't forget to unplug the power supply's cord from the first tile as well.

The final pair that needs to be disconnected is the power connector. This can once again be accomplished with one hand by gripping the connector and using the thumb and pointer finger to push the connector apart. Another method involves gripping the connector from both sides with both hands and pulling apart the connector. It is OKAY to pull on the wires themselves in this step, but best to pull from as close to the joint as possible.

Step 4: Remove the limit switches.
Removing the limit switches can be a bit tricky but should not be too difficult if done correctly. See Figure 145 and Figure 146 for reference.

Figure 145 – Limit switch in place on limit switch mount, resting on internal support.

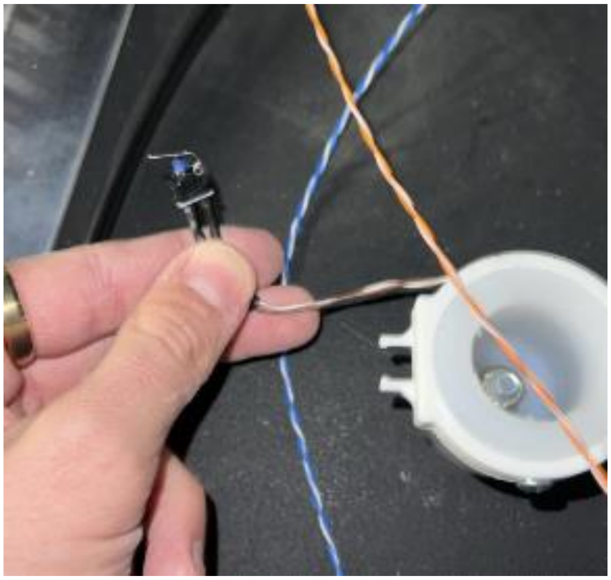


Figure 146 – Limit switch removed. Grab it by the large sections of heat shrink and pull outward away from the support.

Make sure to use the thumb and forefingers to firmly grasp the limit switch from the heat shrink just below the plastic body of the switch. The closer to the mount, the better. Pull away from the support and the switch will pop out of the mount. Do NOT pull on the wires themselves, or the metal arm of the switch.

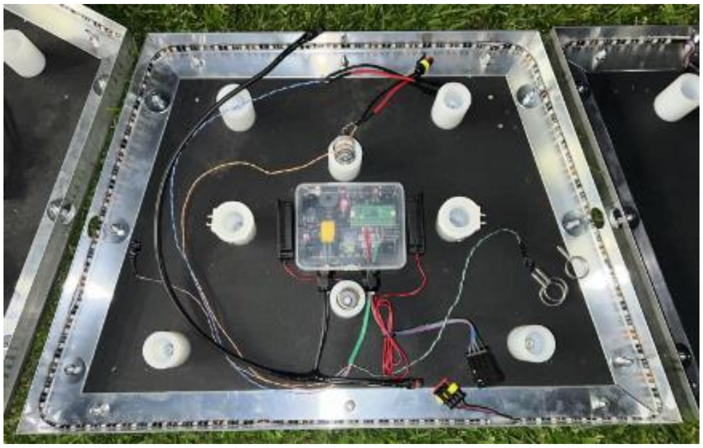


Figure 147 – Tile with all internal connections undone.

If done correctly, the only thing that should be connecting the electronics to the tile is the Velcro underneath the box.

Step 5: Remove the electronics enclosure.

The next step is to CAREFULLY remove the electronics enclosure. Once this is removed, all of the electronics, besides the LEDs themselves, will be able to be pulled from the tile.

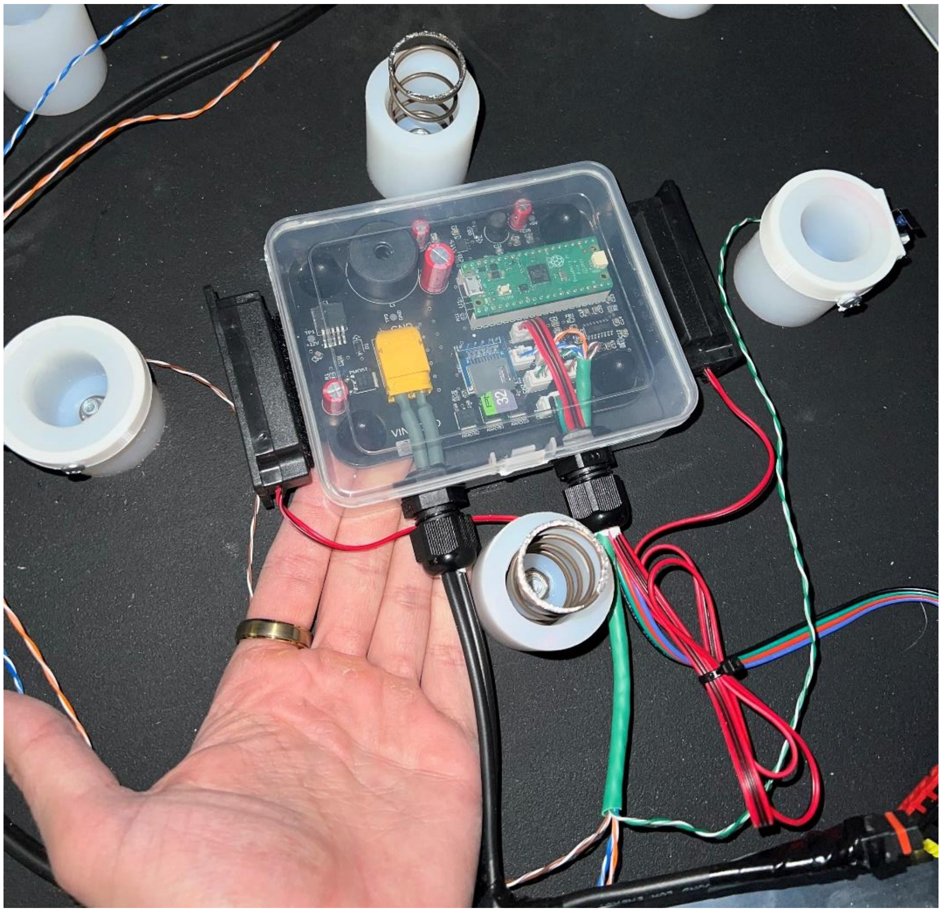


Figure 148 – Wedge hand underneath electronics gently.

Make sure not to be too aggressive in this step. As shown in Figure 148, use the hand to wedge under the tile, starting from one side, and slowly pry the enclosure from the Velcro. If done too aggressively, the enclose may bend and crack, or even snap.

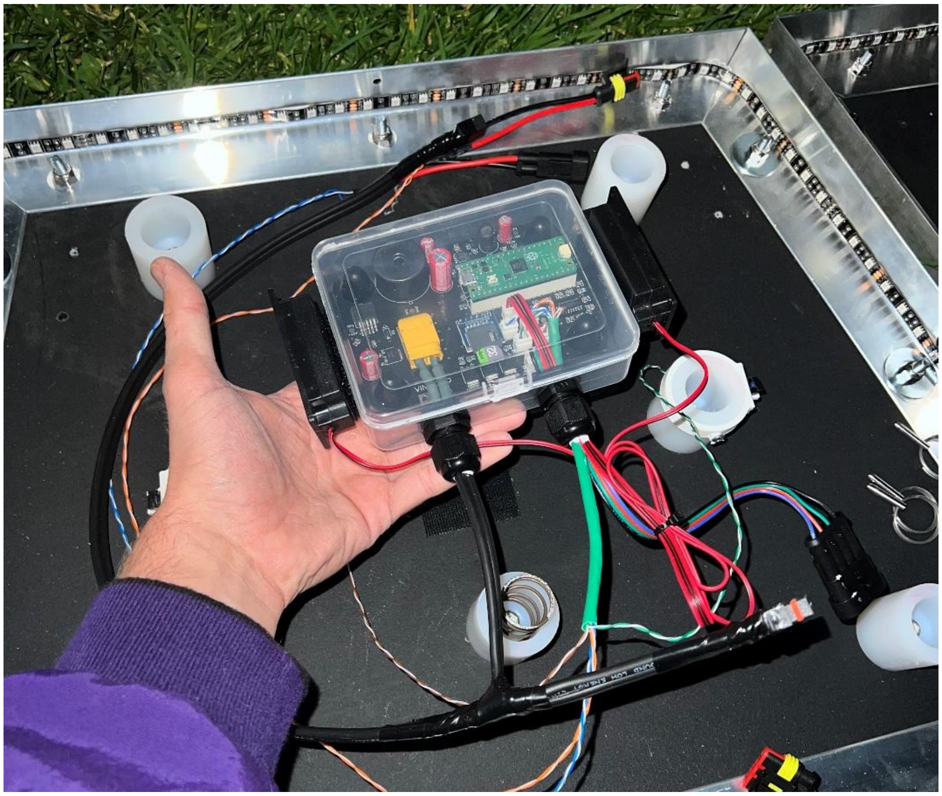


Figure 149 – Slowly get whole hand underneath and massage enclosure away from Velcro.

Once the enclosure has been peeled off the Velcro, the entire system can be pulled from the tile, as shown in Figure 149. It is best to hold the enclosure by the box itself, but the large black wires are sturdy enough to use as well as shown in Figure 150.

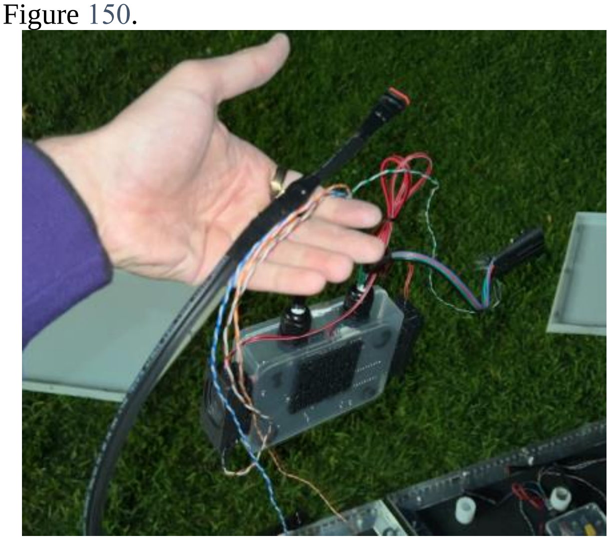


Figure 150 –Hold the electronics enclosures by the large black wire if needed.

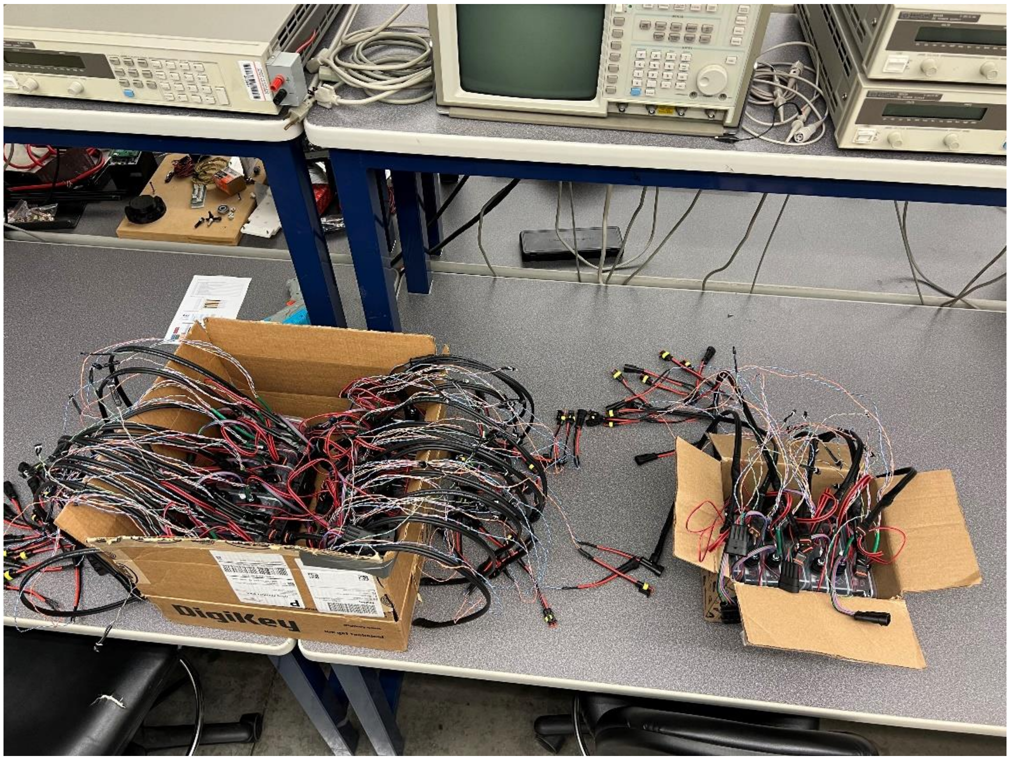


Figure 151 – Example of how electronics can be stored upon a rain delay. Putting all the wires in the box or closing the box would cause the wires to become very tangled.

The storage of the electronics components in cardboard boxes, shown in Figure 151, is just an example of how the enclosures might be stored. It is advised to store them in sealable larger containers with a lid if they are to be stored for a long period of time.

Step 6: Replace the lid.

Replace the lid back on the tile now that the electronics are removed. Do not forget to put back the pins as well. Figure 152 and Figure 153 are duplicated here purposefully for convenience.

Figure 152 – Lid placement technique. Usually performed with two hands, the thumb is pushing inwards on the aluminum while the forefinger is pushing downwards. When performed with two hands, this move is especially effective on corners and at the center of the sides.

Figure 153 – Reinserting the quick-release pins. Make sure the pin goes through both the white top panel and metal hole. Ensure all 3 are replaced on every tile.

Step 7: Take electronics enclosures and power supply to dry area (indoors).

Ideally store these electronics and power supply indoors until the rainy spell is over. If possible, place them in a sealable container with a lid. Once the rainy spell is over, you can proceed to the Electronics Installation section.

5.3 Electronics Installation

Step 1: Open tile lids.
Follow instruction and description in Figure 124 and Figure 125 to remove the lid if needed, but hopefully you are well versed with taking off the lids at this point. Remember to put the pins inside the tile to avoid losing them.

Step 2: Push enclosure gently down onto Velcro.
Ensure wires are pointing out of the enclosure towards the longest side of the tile. Wiggle the enclosure to make sure the Velcro grabs. When pushing down on the enclosure, try to only push near the edges of the box as the center can be quite weak. ENSURE THE CONTROLLER ELECTRONICS RETURN TO THE FIRST TILE.

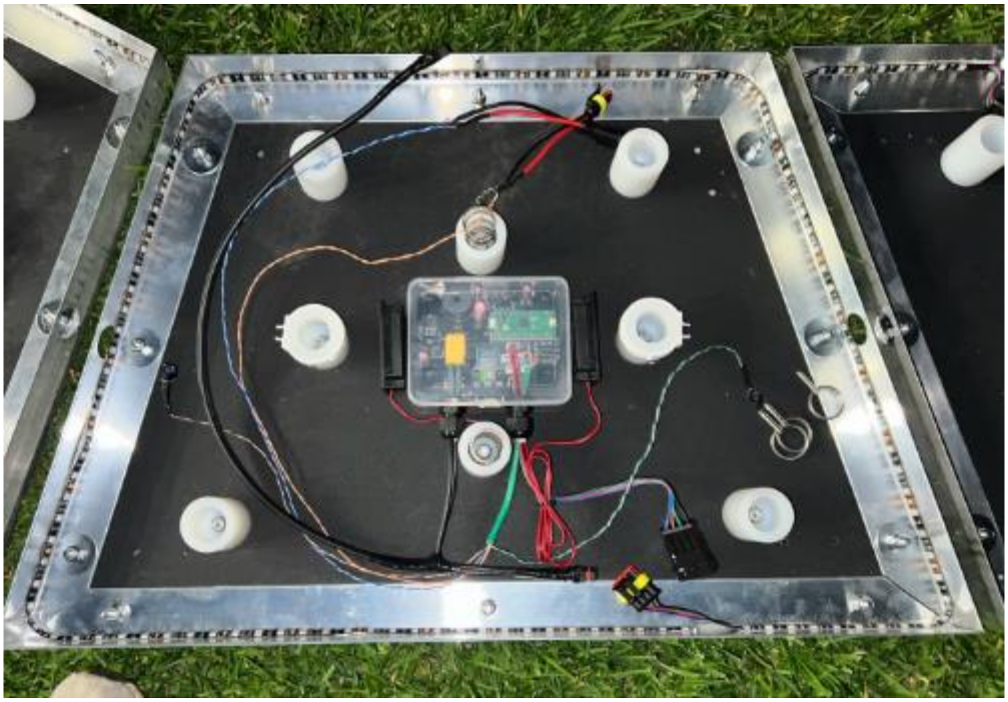


Figure 154 – Tile should be seated as so before connecting Velcro.

Step 3: Reconnect limit switches to supports.

Ensure that you are lining up the limit switches' grooves with the supports' rails (See Figure 155). Once lined up, push on the switch's body, towards the supports center, to insert it (See Figure 156). Make sure that the switch's metal lever points up and towards the longest side of the tile when inserted.

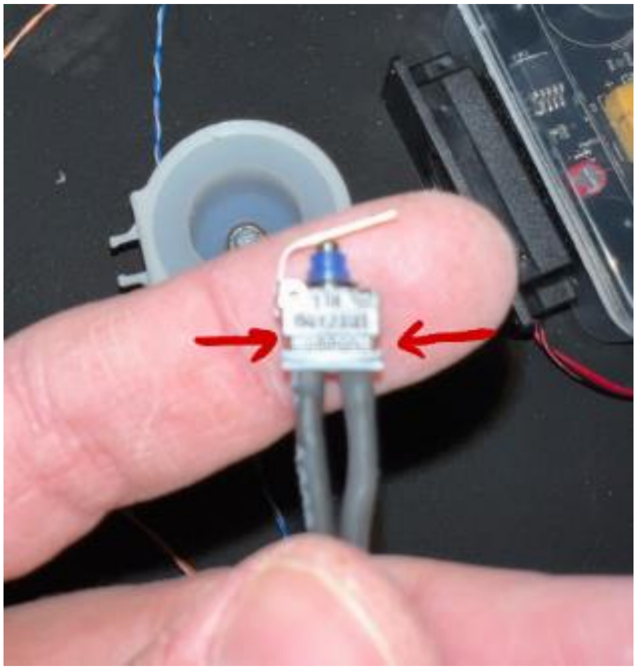


Figure 155 – Switch's grooves that line up with the support mount's rails.

Figure 156 – Pressing on the switch's body to insert it onto the rails. Ensure that the switch's metal arm points up towards the long side of the tile when inserted.

Step 4: Reconnect other connectors.
When reconnecting the components, refer to the Initial Setup section for guidance. Begin by attaching the LED connectors, ensuring they are properly aligned. Next, connect the Blue and Orange inter-tile connectors, followed by the Black power connection. For a visual reference, consult Figure 130 to Figure 135. It's important to remember to connect the "controller" tile as the FIRST in the sequence, as this ensures proper system coordination.

Step 5: Ensure no wires are in the way and replace the lid.
Before replacing the lid, check that no wires or connections are protruding or interfering with the side slots, or might be pinched by the top of the tile and the supports. Proper wiring through the slots is shown in Figure 132. Figure 145 is a good example of a wire being in a bad position, resting on top of one of the supports. If any wires that might obstruct the lid, and refuse to stay out of the way, use electrical tape to secure them neatly to the base. This step is crucial to ensure that the lid fits correctly, and the internal components are protected.

Step 6: Power on sequence (see section 5.1, step 10)
After the electronic components have been reconnected, proceed with the power-on sequence as outlined in section 5.1, Step 10. It's important to note that reprogramming of the system is required at this stage as all the electronics have likely moved around in the sequence. This reprogramming ensures that all newly connected components are functioning correctly.

5.4 Teardown

Step 1: Power down tiles.
Begin by turning off each tile. Do this by toggling the power switch to the "O" position, as shown in Figure 119. After switching it off, wait approximately 5 seconds to ensure all tiles are completely powered down. All of the tiles going dark is a good indicator that they have all shut completely off. This pause is for safely handling the tiles in the subsequent steps.

Step 2: Take off all lids.
Next, remove the lids from all tiles. Start by pulling out three pins per tile, as illustrated in Figure 140 and Figure 141. Handle the lids and pins with care and remember to keep them together with their corresponding tile for easy reassembly.

Step 3: Undo all cable connections (besides LEDs).
Refer to Section 5.2, Step 3, for guidance on disconnecting cable connections, excluding the LEDs. The LEDs can be left connected since the electronics enclosure can be left inside the tile.

Step 4: Pull up on tile to pull out stakes. Remove stakes.

To remove stakes, first position your hand beneath the tile as shown in Figure 157. Gently lift the tile upwards; this action will disengage the stake from the ground, as depicted in Figure 158. Once the stake is loose, lower the tile back to the ground and then remove the stake completely from the tile, as indicated in Figure 159. This method ensures smooth and efficient removal of the stakes without exerting unnecessary force. If this proves to be difficult, a hammer or prybar can be utilized to pull the tiles up.

Figure 157 – Place hand underneath tile near stake.

Figure 158 – Lift hand upwards and notice stake come out of the ground.

Figure 159 – Let tile return to the ground and pull out the stakes by hand now. Store in a safe location.

Step 5: Replace lids.
Carefully place the lids back onto each tile, ensuring they fit snugly and securely. Once the lids are in place, stack the tiles in an organized manner. Align them so that the grip taped tops are facing each other, and similarly, the rubber bottom are also facing each other. This stacking method will prevent the top grip tape surface from getting dirty from the rubber pads. See Figure 160 for reference.

Figure 160 – Tiles stacked in alternating method so grip tape stacks against grip tape and rubber against rubber.

Step 6: Store in indoor location (Power supply included).
Once all the tiles are securely closed and stacked, store them in an indoor location. Along with the tiles, remember to include the power supply in this storage. Keeping the power supply with the tiles ensures that everything necessary for operation is in one place, reducing the risk of misplacing or damaging any components.

APPENDIX C – RISK ASSESSMENT

designsafe Report

Application:	Path of Lights and Sounds	Analyst Name(s):	Sean Wahl, Ryan Johnson, Quinn Stephens, Vinh Vo
Description:		Company:	Tagalong Trail
Product Identifier:		Facility Location:	Cal Poly
Assessment Type:	Detailed		
Limits:			
Sources:			
Risk Scoring System:	ANSI B11.0 Two Factor		

Guide sentence: When doing [task], the [user] could be injured by the [hazard] due to the [failure mode].

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
1-1-1	tile user Common Tasks	noise / vibration : noise / sound levels > 80 dBA intermittent, inherent to use	Minor Remote	Negligible		Minor		
1-2-1	tile user jump/walk	slips / trips / falls : slip misuse only	Minor Remote	Negligible		Minor		
1-2-2	tile user jump/walk	slips / trips / falls : trip misuse only	Minor Remote	Negligible		Minor		
1-3-1	tile user misuse	mechanical : pinch point reassembling tile creates pinch point	Minor Remote	Negligible		Minor		
1-3-2	tile user misuse	mechanical : break up during operation extreme misuse may break surface	Minor Remote	Negligible		Minor		
1-3-3	tile user misuse	slips / trips / falls : slip misuse only	Minor Remote	Negligible		Minor		
1-3-4	tile user misuse	slips / trips / falls : trip misuse only	Minor Remote	Negligible		Minor		

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
2-1-1	operator clean up	mechanical : pinch point reassembling tile creates pinch point	Minor Remote	Negligible		Minor		
2-1-2	operator clean up	electrical / electronic : energized equipment / live parts programming requires power	Minor Remote	Negligible		Minor		
2-1-3	operator clean up	electrical / electronic : water / wet locations tile will be outdoors	Minor Remote	Negligible		Minor		
2-2-1	operator basic trouble shooting / problem solving	electrical / electronic : energized equipment / live parts programming requires power	Minor Remote	Negligible		Minor		
2-3-1	operator load / unload materials	mechanical : pinch point reassembling tiles creates pinch point	Minor Remote	Negligible		Minor		
2-4-1	operator set up	mechanical : pinch point reassembling tiles creates pinch point	Minor Remote	Negligible		Minor		
2-4-2	operator set up	electrical / electronic : energized equipment / live parts Tiles will be powered	Minor Remote	Negligible		Minor		
2-5	operator adjust controls/ settings / alignment	<None>						
3-1-1	maintenance technician parts replacement	mechanical : pinch point reassembling tiles createes pinch point	Minor Remote	Negligible		Minor		

Item Id	User / Task	Hazard / Failure Mode	Initial Assessment Severity Probability	Risk Level	Risk Reduction Methods /Control System	Final Assessment Severity Probability	Risk Level	Status / Responsible /Comments /Reference
3-1-2	maintenance technician parts replacement	electrical / electronic : energized equipment / live parts tiles will be powered	Minor Remote	Negligible		Minor		
3-2-1	maintenance technician periodic maintenance	mechanical : pinch point reassembling tiles creates pinch point	Minor Remote	Negligible		Minor		
3-2-2	maintenance technician periodic maintenance	electrical / electronic : energized equipment / live parts tiles will be powered	Minor Remote	Negligible		Minor		
3-3-1	maintenance technician trouble-shooting / problem solving	mechanical : pinch point reassembling tiles creates pinch point	Minor Remote	Negligible		Minor		
4-1-1	passer by / non-user walk near product	noise / vibration : noise / sound levels > 80 dBA tiles create noise	Minor Remote	Negligible		Minor		

APPENDIX D – FINAL PROJECT BUDGET

Table D1 – Purchases Made.

Description	Vendor	Date Purchased	Cost	Funder
Structural Prototype	McMaster-Carr	4/21/2023	$306.43	Cal Poly
Structural Prototype Add-ons I	McMaster-Carr	5/04/2023	$141.58	Team
Structural Prototype Add-ons II	McMaster-Carr	6/01/2023	$143.65	Cal Poly
Extra Speakers for Testing	Adafruit	9/06/2023	$35.06	Team
Test Board 1 Electronic Components	DigiKey	9/06/2023	$92.91	Team
Test Board 1 PCBs	JLCPCB	9/07/2023	$16.75	Team
Wire and Sealant	Home Depot	10/03/2023	$67.90	Team
Test Board 2 Electronic Components	DigiKey	10/08/2023	$77.05	Team
Test Board 2 PCBs	JLCPCB	10/08/2023	$27.65	Team
Miscellaneous Amazon Orders	Amazon	Spring - Fall	$301.86	Team
Final Structural Order	McMaster-Carr	10/20/2023	$4437.28	GSCCC
Final PCB Order	JLCPCB	10/25/2023	$100.94	GSCCC
Final Amazon Order	Amazon	10/25/2023	$694.33	GSCCC
Final Speaker Order	Adafruit	10/26/2023	$217.18	GSCCC
Final Electronics Components Order	DigiKey	10/26/2023	$963.23	GSCCC
Power Transformer Enclosure	Home Depot	10/27/2023	$58.72	GSCCC

Table D2 – Budget Available vs. Spent.

Entity	Budget Available	Budget Spent
GSCCC	$6750*	$6471.68
Cal Poly	$500	$450.08
Team**	$619.18	$760.76
	Total Spent	$7682.52

* The original budget from the Girl Scouts was $5000, increased to a suggested $6750 with an absolute maximum of $7000.

** The team donated money to the project during the prototyping phase of the electronics as a generous gift to the Girl Scouts. This also allowed the prototyping phase to move much faster and make significant progress more quickly.

APPENDIX E – CODE

All code is present here as well as provided in the Software Package provided to the sponsor.

main.py

```
# Importing standard modules
import board, digitalio, time, math, os, binascii, struct, busio, sdcardio,
storage
from time import monotonic as now

# Importing custom modules
import ledstrip, speaker, Lookup
from dataFunctions import getControllerInfo, getFollowerInfo, readNoteTxt,
writeNoteTxt

## ---------------------------------------------------------------------- ##
# Initialize all pins and hardware #
## ---------------------------------------------------------------------- ##

# Set up UART Pins
uart_backward = busio.UART(board.GP8, board.GP9, baudrate=9600, timeout =
.01)
uart_forward  = busio.UART(board.GP12, board.GP13, baudrate=9600, timeout =
.01)

# Set up limit switch reading pin
limsw = digitalio.DigitalInOut(board.GP0)
limsw.direction = digitalio.Direction.INPUT

# Set up LED pins (see ledstrip.py for more details)
leds = ledstrip.LedStrip(board.GP27, board.GP26, board.GP28)

# Set up SD Card's pins for accessing SD Card
spi = busio.SPI(board.GP2, MOSI=board.GP3, MISO=board.GP4)
cs = board.GP5
sdcard = sdcardio.SDCard(spi, cs)

# Mount the SD Card's file system under the virtual path "/sd"
vfs = storage.VfsFat(sdcard)
sdpath = "/sd"
storage.mount(vfs, sdpath)

# Define default path to the SD Card's sound .wav files
path = sdpath + "/sounds/"

# Set up the speakers input pins
speaker = speaker.Speaker(path, board.GP16, board.GP17)

## ---------------------------------------------------------------------- ##
# Initialize default variables, states, etc. #
## ---------------------------------------------------------------------- ##

# Initialize state number placeholder variables
CONTROLLER_INIT        = 0
```

```python
CONTROLLER_PROGRAMMING = 1
CONTROLLER_NOTETXT     = 2
CONTROLLER_BACKCOMMS   = 3
CONTROLLER_WAITFORSTEP = 4
CONTROLLER_SOUND       = 5
CONTROLLER_FADE        = 6
CONTROLLER_DELAY       = 7

FOLLOWER_INIT          = 10
FOLLOWER_NOTETXT       = 11
FOLLOWER_WAITFORMSG    = 12
FOLLOWER_RELAY         = 13
FOLLOWER_BACKCOMMS     = 14
FOLLOWER_WAITFORSTEP   = 15
FOLLOWER_SOUND         = 16
FOLLOWER_FADE          = 17

# Intialize tile values that will always be the same
t_fade    = 1 # The time it takes the light to fade after being stepped on
[s]
t_delay   = 7 # The time that the first tile waits before accepting a new
user [s]
t_timeout = 5 # The time that the tiles will wait for a step before resetting
their state [s]

# Initialize default tile values relating to light and sound
beforeColor = ledstrip.ColorStruct(255, 255, 0) # Tile color before being
stepped on (RGB values)
afterColor  = ledstrip.ColorStruct(0, 255, 255) # Tile color after being
stepped on (RGB values)
note        = 'C4'    # Note to be played
duration    = 1       # Duration of the note [s]
rest        = 0       # Rest after this note before the next note [s]
instrument  = 'Piano' # Instrument to be played

# Init state variable, set a timestamp
directoryFiles = os.listdir(sdpath)
head = True if 'controller.txt' in directoryFiles else False
state = CONTROLLER_INIT if head else FOLLOWER_INIT


while True:
## ---------------------------------------------------------------------- ##
# Controller State Machine #
## ---------------------------------------------------------------------- ##
    if state == CONTROLLER_INIT:
        # If being stepped on on startup, send out tile programming
        if limsw.value:
            state = CONTROLLER_PROGRAMMING

        # If a note.txt file is not present, send out tile programming
        elif 'note.txt' in directoryFiles:
            state = CONTROLLER_NOTETXT

        # Else, go read the note.txt file to get tile information
        else:
            state = CONTROLLER_PROGRAMMING
```

```
## ---------------------------------------------------------------------- ##
    elif state == CONTROLLER_PROGRAMMING:
        # Let all other tiles power on definitively
        time.sleep(2)

        # Convert the file's info into the controller tile's information and
sendable bytes to the next tiles
        myInfo, dataBytes = getControllerInfo('/sd/controller.txt')

        # Send the song's information bytes onwards
        uart_forward.write(dataBytes)

        # Take the controller tile's information and assign it to the right
variables
        beforeColor = ledstrip.ColorStruct(myInfo[0], myInfo[1], myInfo[2])
        afterColor  = ledstrip.ColorStruct(myInfo[3], myInfo[4], myInfo[5])
        note        = myInfo[6]
        duration    = myInfo[7]
        rest        = myInfo[8]
        instrument  = myInfo[9]

        # Write current settings to note.txt
        writeNoteTxt('/sd/note.txt', myInfo)

        # Go to waiting for back communication and light up to indicate that
this tile has received its values
        leds.color(0, 0x40, 0)
        state = CONTROLLER_BACKCOMMS
        t_comms = now()

## ---------------------------------------------------------------------- ##
    elif state == CONTROLLER_NOTETXT:
        # Receive program from note.txt
        myInfo = readNoteTxt('/sd/note.txt')

        # Assign the info from note.txt to the proper variables
        beforeColor = ledstrip.ColorStruct(myInfo[0], myInfo[1], myInfo[2])
        afterColor  = ledstrip.ColorStruct(myInfo[3], myInfo[4], myInfo[5])
        note        = myInfo[6]
        duration    = myInfo[7]
        rest        = myInfo[8]
        instrument  = myInfo[9]

        # Go to the waiting for step state
        state = CONTROLLER_WAITFORSTEP
        t_waiting = now()

## ---------------------------------------------------------------------- ##
    elif state == CONTROLLER_BACKCOMMS:
        # Wait for a message to come in
        data = uart_forward.read()

        if data is not None:
            data = binascii.hexlify(data)
            # If the correct back communications are received, enter here
            if data[:2] == b'ff':
```

```
                # Turn off the leds and delay for 1 second
                leds.off()
                time.sleep(1)

                # Go to the wait for step state
                state = CONTROLLER_WAITFORSTEP
                t_waiting = now()

        # If it has been a minute and no communications have been received,
stop waiting
        elif (now() - t_comms) > 60:
            leds.off()
            state = CONTROLLER_WAITFORSTEP
            t_waiting = now()

## ----------------------------------------------------------------------- ##
    elif state == CONTROLLER_WAITFORSTEP:
        # Breathe leds while waiting for a step
        dt = now() - t_waiting
        scale = 0.5 * math.sin(math.pi/2 * (dt - 1)) + 0.5
        leds.colorS(beforeColor, scale)

        # Check if the tile has been stepped on
        if limsw.value:
            # Go to the sound state
            state = CONTROLLER_SOUND
            t_step = now()

## ----------------------------------------------------------------------- ##
    elif state == CONTROLLER_SOUND:
        # Change to after color (at half brightness)
        leds.colorS(afterColor, 0.5)

        # Tell the next tile that this tile to expect a step and give it a
wait time
        try:
            nextWait = struct.pack('f', duration+rest)
            msg = b'\xff' + nextWait
            uart_forward.write(msg)
        except:
            nextWait = struct.pack('f', 1)
            msg = b'\xff' + nextWait
            uart_forward.write(msg)


        # Play tile's sound
        try:
            soundFile = instrument+'/'+instrument+note+'.wav'
            speaker.play(soundFile, duration)
        except:
            pass

        # After playing the sounds, go to the fade state
        state = CONTROLLER_FADE
        t_soundDone = now()

## ----------------------------------------------------------------------- ##
```

```
    elif state == CONTROLLER_FADE:
        # See how much time has passed since the sound finished
        t_diff = now() - t_soundDone

        # Fade out the color linearly based on t_fade
        scale = 0.5*(1 - t_diff/t_fade)
        leds.colorS(afterColor, scale)

        # If it has been longer than t_fade, go to the delay state
        if t_diff >= t_fade:
            state = CONTROLLER_DELAY
            leds.off()

## ------------------------------------------------------------------------ ##
    elif state == CONTROLLER_DELAY:
        # Wait for the delay to be over, go back to waiting for a step
        if (now() - t_soundDone) >= (t_fade + t_delay):
            state = CONTROLLER_WAITFORSTEP
            t_waiting = now()


## ------------------------------------------------------------------------ ##
# Follower State Machine #
## ------------------------------------------------------------------------ ##
    elif state == FOLLOWER_INIT:
        # If a note.txt is present, go to get its information
        if 'note.txt' in directoryFiles:
            state = FOLLOWER_NOTETXT

        # Else, set default tile parameters and go to waiting for message
        else:
            beforeColor = ledstrip.ColorStruct(255, 255, 0) # Tile color
before being stepped on (RGB values)
            afterColor  = ledstrip.ColorStruct(0, 255, 255) # Tile color
after being stepped on (RGB values)
            note        = 'C4'    # Note to be played
            duration    = 1       # Duration of the note
            rest        = 0       # Rest after this note before the next note
            instrument  = 'Piano' # Instrument to be played

            state = FOLLOWER_WAITFORMSG

## ------------------------------------------------------------------------ ##
    elif state == FOLLOWER_NOTETXT:
        # Receive program from note.txt
        myInfo = readNoteTxt('/sd/note.txt')

        # Assign the info from note.txt to the proper variables
        beforeColor = ledstrip.ColorStruct(myInfo[0], myInfo[1], myInfo[2])
        afterColor  = ledstrip.ColorStruct(myInfo[3], myInfo[4], myInfo[5])
        note        = myInfo[6]
        duration    = myInfo[7]
        rest        = myInfo[8]
        instrument  = myInfo[9]

        # Go to the waiting for step state
        state = FOLLOWER_WAITFORMSG
```

```
## ------------------------------------------------------------------------ ##
    elif state == FOLLOWER_WAITFORMSG:
        # Wait for a message to come in
        dataInRaw = uart_backward.read()

        # See what kind of message has come in
        # NOTE: There are three types of messages:
        # 1. programming messages (aka containing song information)
        #   -these are long and contain at least 11 bytes of information
        #   -WILL NEVER start with 0x00 or 0xFF (first byte is tempo, 60-190
[bpm])
        # 2. "step incoming" messages
        #   -ALWAYS starts with 0xFF, followed by 4 bytes encoding a floating
point
        #    number indicating how long to expect the step in [s]
        # 3. Random noise messages
        #   -mostly always a lot of zeroes
        #   -usually only happens on start up
        if dataInRaw is not None:
            # Convert the data to a readable format
            dataIn = binascii.hexlify(dataInRaw)
            # Pick off the first byte of data and convert it
            check = dataIn[:2]

            # If the first byte is 0, ignore the message
            if check == b'00':
                pass

            # If the first byte is 255, start expecting a step
            elif check == b'ff':
                wait, = struct.unpack('f', dataInRaw[1:5])
                state = FOLLOWER_WAITFORSTEP
                t_waiting = now()

            # Otherwise the data is encoding a song if theres at least 11
bytes (2 characters each)
            elif len(dataIn) >= 22:
                myInfo, dataForward = getFollowerInfo(dataInRaw)

                # Go to the relay state
                state = FOLLOWER_RELAY


## ------------------------------------------------------------------------ ##
    elif state == FOLLOWER_RELAY:
        print('hi')
        # Turn the LEDs on to indicate that its data has been received
        leds.color(0, 0x40, 0)

        # Assign the info from note.txt to the proper variables
        beforeColor = ledstrip.ColorStruct(myInfo[0], myInfo[1], myInfo[2])
        afterColor  = ledstrip.ColorStruct(myInfo[3], myInfo[4], myInfo[5])
        note        = myInfo[6]
        duration    = myInfo[7]
        rest        = myInfo[8]
        instrument  = myInfo[9]
```

```
        # Write this info to note.txt file
        writeNoteTxt('/sd/note.txt', myInfo)

        # Sleep for a short time to make progression semi-visible
        time.sleep(.1)

        # See if there's information to relay
        if dataForward is not None:
            # Send the data forward
            uart_forward.write(dataForward)

            # Go to the back communications state
            state = FOLLOWER_BACKCOMMS
            t_comms = now()

        # If there's no more information, being the back communication chaing
        else:
            print('going back')
            # Wait 1 second to let this light be visible
            time.sleep(1)

            # Turn LEDs off, begin back communication, and begin waiting for
a message again
            leds.off()
            uart_backward.write(bytes([0xFF]))
            state = FOLLOWER_WAITFORMSG

## ------------------------------------------------------------------------ ##
    elif state == FOLLOWER_BACKCOMMS:
        # Wait for a message to come in
        data = uart_forward.read()

        if data is not None:
            data = binascii.hexlify(data)
            # If the correct back communications are received, enter here
            if data[:2] == b'ff':
                # Continue relaying the message backwards
                uart_backward.write(bytes([0xFF]))

                # Turn off the leds
                leds.off()

                # Go to the wait for message state
                state = FOLLOWER_WAITFORMSG

        # If it has been a minute and no communications have been received,
stop waiting
        elif (now() - t_comms) > 60:
            leds.off()
            state = FOLLOWER_WAITFORMSG

## ------------------------------------------------------------------------ ##
    elif state == FOLLOWER_WAITFORSTEP:
        # See how much time has passed since the message that a step is
coming has been received
        t_passed = now() - t_waiting
```

```
        # Calculate how bright the tile should be at this time step (up to a
maximum of 100%)
        scale = math.exp(2.5 * (t_passed - wait))
        leds.colorS(beforeColor, scale)

        # See if the tile has been stepped on
        if limsw.value:
            state = FOLLOWER_SOUND
            t_step = now()

        # Time out if too long has passed
        elif (t_passed - wait) > t_timeout:
            leds.off()
            state = FOLLOWER_WAITFORMSG

## ----------------------------------------------------------------------- ##
    elif state == FOLLOWER_SOUND:
        # Change to after color (at half brightness)
        leds.colorS(afterColor, 0.5)

        # Tell the next tile that this tile to expect a step and give it a
wait time
        try:
            nextWait = struct.pack('f', duration+rest)
            msg = b'\xff' + nextWait
            uart_forward.write(msg)
        except:
            nextWait = struct.pack('f', 1)
            msg = b'\xff' + nextWait
            uart_forward.write(msg)

        # Play tile's sound
        try:
            soundFile = instrument+'/'+instrument+note+'.wav'
            speaker.play(soundFile, duration)
        except:
            pass

        # After playing the sounds, go to the fade state
        state = FOLLOWER_FADE
        t_soundDone = now()

## ----------------------------------------------------------------------- ##
    elif state == FOLLOWER_FADE:
        # See how much time has passed since the sound finished
        t_diff = now() - t_soundDone

        # Fade out the color linearly based on t_fade
        scale = 0.5*(1 - t_diff/t_fade)
        leds.colorS(afterColor, scale)

        # If it has been longer than t_fade, go to the delay state
        if t_diff >= t_fade:
            state = FOLLOWER_WAITFORMSG
            leds.off()
```

```python
# See if the previous tile was pressed
dataRaw = uart_backward.read()
if dataRaw is not None:
    data = binascii.hexlify(dataRaw)

    # If the first byte is 255, start expecting a step
    if data[:2] == b'ff':
        leds.off()
        wait, = struct.unpack('f', data[1:5])
        state = FOLLOWER_WAITFORSTEP
        t_waiting = now()
```

ledstrip.py

```python
import board
from pwmio import PWMOut

class LedStrip:

    def __init__(self, Rpin, Gpin, Bpin):
        self.Rpin = PWMOut(Rpin, frequency=5000, duty_cycle=0)
        self.Gpin = PWMOut(Gpin, frequency=5000, duty_cycle=0)
        self.Bpin = PWMOut(Bpin, frequency=5000, duty_cycle=0)
        self._max = 65535

    def off(self):
        self.Rpin.duty_cycle = 0
        self.Gpin.duty_cycle = 0
        self.Bpin.duty_cycle = 0

    def white(self):
        self.Rpin.duty_cycle = self._max
        self.Gpin.duty_cycle = self._max
        self.Bpin.duty_cycle = self._max

    def color(self, R, G, B):
        self.Rpin.duty_cycle = round(R * self._max / 255)
        self.Gpin.duty_cycle = round(G * self._max / 255)
        self.Bpin.duty_cycle = round(B * self._max / 255)

    def colorS(self, RGBStruct, scale = 1):
        if scale > 1:
            scale = 1

        if scale < 0:
            scale = 0

        self.Rpin.duty_cycle = round(RGBStruct.R * scale * self._max / 255)
        self.Gpin.duty_cycle = round(RGBStruct.G * scale * self._max / 255)
        self.Bpin.duty_cycle = round(RGBStruct.B * scale * self._max / 255)


class ColorStruct:

    def __init__(self, R, G, B):
        self.R = R
        self.G = G
        self.B = B

if __name__ == "__main__":
    leds = LedStrip(board.GP1, board.GP2, board.GP3)
    leds.off()
    leds.white()
    leds.color(0xFF, 0x00, 0x10)
```

speaker.py

```
import board
from time import monotonic as now
from audiocore import WaveFile
from audiopwmio import PWMAudioOut as AudioOut
from busio import UART
import microcontroller


class Speaker:

    def __init__(self, path, Lchan, Rchan = False):
        self.Lchan = Lchan
        self.Rchan = Rchan
        self.path = path

        if self.Rchan:
            self.audio = AudioOut(left_channel = self.Lchan, right_channel =
self.Rchan)
        else:
            self.audio = AudioOut(self.Lchan)

    def play(self, filename, duration = float('inf')):
        try:
            with open(self.path+filename, "rb") as wave_file:
                wave = WaveFile(wave_file)
                t1 = now()
                self.audio.play(wave)
                while self.audio.playing:
                    if (now() - t1) >= duration:
                        self.audio.stop()
        except:
            microcontroller.reset()

    def deinit(self):
        audio.deinit()

    def reinit(self):
        if self.Rchan:
            self.audio = AudioOut(left_channel = self.Lchan, right_channel =
self.Rchan)
        else:
            self.audio = AudioOut(self.Lchan)
```

dataFunctions.py

```
import Lookup, binascii

def getControllerInfo(filename):
    # Output format
    # 0 - R beforeColor (0 to 255)
    # 1 - G beforeColor (0 to 255)
    # 2 - B beforeColor (0 to 255)
    # 3 - R afterColor  (0 to 255)
    # 4 - G afterColor  (0 to 255)
    # 5 - B afterColor  (0 to 255)
    # 6 - note (string)
    # 7 - duration (seconds)
    # 8 - rest (seconds)
    # 9 - instrument ( string)

    with open(filename) as f:
        fileData = f.readlines()
    del fileData[0]
    for index in range(len(fileData)):
        fileData[index] = fileData[index].replace('\r','')
        fileData[index] = fileData[index].replace('\n','')

    out = ['']*10
    # Split first line in tempo, before color, and after color
    tempo, beforeColor, afterColor = fileData[0].split()
    tempo = int(tempo, 16) # tempo in bpm

    # out[0:2] = R,G,B before color
    out[0] = int(beforeColor[:2], 16)
    out[1] = int(beforeColor[2:4], 16)
    out[2] = int(beforeColor[4:6], 16)

    #out[3:5] = R,G,B, after color
    out[3] = int(afterColor[:2], 16)
    out[4] = int(afterColor[2:4], 16)
    out[5] = int(afterColor[4:6], 16)

    #out[6:9] = note, duration, rest, instrument
    out[6] = Lookup.note[int(fileData[1][:2], 16)]
    out[7] = int(fileData[1][2:4], 16)*(60/(4*tempo))
    out[8] = int(fileData[1][4:6], 16)*(60/(4*tempo))
    out[9] = Lookup.instrument[int(fileData[1][6:8], 16)]

    outList = [tempo, out[0], out[1], out[2], out[3], out[4], out[5]]
    del fileData[0:2]
    for line in fileData:
        outList.append(int(line[:2],16))
        outList.append(int(line[2:4],16))
        outList.append(int(line[4:6],16))
        outList.append(int(line[6:8],16))

    dataBytes = bytes(outList)
    return out, dataBytes
```

```
def getFollowerInfo(dataUART):
    # Output format
    # 0 - R beforeColor (0 to 255)
    # 1 - G beforeColor (0 to 255)
    # 2 - B beforeColor (0 to 255)
    # 3 - R afterColor  (0 to 255)
    # 4 - G afterColor  (0 to 255)
    # 5 - B afterColor  (0 to 255)
    # 6 - note (string)
    # 7 - duration (seconds)
    # 8 - rest (seconds)
    # 9 - instrument ( string)

    dataHex = binascii.hexlify(dataUART)
    tempo = int(dataHex[:2], 16)

    out = ['']*10
    out[0] = int(dataHex[2:4],16)
    out[1] = int(dataHex[4:6],16)
    out[2] = int(dataHex[6:8],16)
    out[3] = int(dataHex[8:10],16)
    out[4] = int(dataHex[10:12],16)
    out[5] = int(dataHex[12:14],16)
    out[6] = Lookup.note[int(dataHex[14:16],16)]
    out[7] = int(dataHex[16:18],16)*(60/(4*tempo))
    out[8] = int(dataHex[18:20],16)*(60/(4*tempo))
    out[9] = Lookup.instrument[int(dataHex[20:22],16)]

    dataTrimmed = dataHex[:14] + dataHex[22:]
    if len(dataTrimmed) < 22:
        dataUART = None
    else:
        dataUART = binascii.unhexlify(dataTrimmed)

    return out, dataUART

def writeNoteTxt(filename, tileInfo):
    with open(filename, "w") as f:
        for item in tileInfo:
            f.write(str(item)+'\n')

def readNoteTxt(filename):
    with open(filename, "r") as f:
        fileData = f.readlines()

    for index in range(len(fileData)):
        fileData[index] = fileData[index].replace('\r','')
        fileData[index] = fileData[index].replace('\n','')

    out = ['']*10
    out[0] = int(fileData[0]) # bR
    out[1] = int(fileData[1]) # bG
    out[2] = int(fileData[2]) # bB
    out[3] = int(fileData[3]) # aR
    out[4] = int(fileData[4]) # aG
    out[5] = int(fileData[5]) # aB
    out[6] = fileData[6] # note string
```

```
out[7] = float(fileData[7]) # duration
out[8] = float(fileData[8]) # rest
out[9] = fileData[9] # instrument string

return out
```

Lookup.py

```python
instrument = { 0x01: "Piano",
               0x02: "Glockenspiel",
               0x03: "SineWave"}

instrumentRev = {}
instrumentCopy = instrument.copy()
while instrumentCopy:
    key, value = instrumentCopy.popitem()
    instrumentRev[value] = key

note = { 0x00: "C3",
         0x01: "Cs3",
         0x02: "D3",
         0x03: "Ds3",
         0x04: "E3",
         0x05: "F3",
         0x06: "Fs3",
         0x07: "G3",
         0x08: "Gs3",
         0x09: "A3",
         0x0A: "As3",
         0x0B: "B3",
         0x0C: "C4",
         0x0D: "Cs4",
         0x0E: "D4",
         0x0F: "Ds4",
         0x10: "E4",
         0x11: "F4",
         0x12: "Fs4",
         0x13: "G4",
         0x14: "Gs4",
         0x15: "A4",
         0x16: "As4",
         0x17: "B4",
         0x18: "C5",
         0x19: "Cs5",
         0x1A: "D5",
         0x1B: "Ds5",
         0x1C: "E5",
         0x1D: "F5",
         0x1E: "Fs5",
         0x1F: "G5",
         0x20: "Gs5",
         0x21: "A5",
         0x22: "As5",
         0x23: "B5",
         0x24: "C6" }

noteRev = {}
noteCopy = note.copy()
while noteCopy:
    key, value = noteCopy.popitem()
    noteRev[value] = key
```

```
duration = {0x10: 'Whole Note',
            0x08: 'Half Note',
            0x04: 'Quarter Note',
            0x02: 'Eighth Note',
            0x01: 'Sixteenth Note'}

durationRev = {}
durationCopy = duration.copy()
while durationCopy:
    key, value = durationCopy.popitem()
    durationRev[value] = key

rest = {0x10: 'Whole Note',
        0x08: 'Half Note',
        0x04: 'Quarter Note',
        0x02: 'Eighth Note',
        0x01: 'Sixteenth Note',
        0x00: 'None'}

restRev = {}
restCopy = rest.copy()
while restCopy:
    key, value = restCopy.popitem()
    restRev[value] = key
```

APPENDIX F – DESIGN VERIFICATION PLAN & REPORT (DVPR)

DVP&R - Design Verification Plan (& Report)

Project: Path of Lights and Sounds | Sponsor: Walk of Wonders and Girl Scouts of CCC | Edit Date: 12/7/2023

TEST PLAN										TEST RESULTS	
Test #	Specification	Test Description	Measurements	Acceptance Criteria	Required Facilities/Equipment	Parts Needed	Responsibility	TIMING Start date	TIMING Finish date	Numerical Results	Notes on Testing
1	Coefficient of Friction	Take the tile's top surface (with grip tape applied) and put a shoe on top of it. Tilt the surface until the shoe starts to slip. Measure the angle at which slippage occurs. Repeat for different shoe types.	Angle of tipping	u >= 0.5	Angle measurement device or video camera (to visually measure)	Tile top surface	Ryan	10/20/2023	11/6/2023	Average coefficient of friction between shoe and tile of 1.20 across three different shoe types	**PASS**
2	Impact Protection	Fixate tile to ground using stakes and rubber pads separately. Drop objects of known weights onto the tile. Jump on the tile several times.	Visual inspection for surface/frame damage or	No critical damage	Objects of known weight, grass or dirt location	Full tile	Vinh	10/20/2023	11/6/2023	No numerical results	Only small scratches were observed on the underside of the top surface at the point of contact with the interior supports. Deemed insignificant: **PASS**
3	Number of Kicks to Move	Instead of kicks, an object of known weight will be swung from a known angle for a consistent impact. Once the tile is fixed down, succesive impacts will be applied. The distance moved after 25 impacts will be recorded. Repeat for both fixation methods (stakes and pads).	Number of consecutive impacts	25 impacts moves the tile < 1in	Outdoors in grass or dirt location	Full tile	Sean	10/20/2023	11/6/2023	Stakes on grass moved 3/8" with 25 kicks. Rubber pads on concrete moved 1 1/4" with 1 kick.	Staked down: **PASS** Rubber pads: **FAIL** Due to these results, the tiles should only ever be used with stakes. Using on hard ground with pads presents unnecessary slipping danger.
4	Manufacturing Time	Once the manufacturing plan has been finalized, the team will time how long it takes one person to make one tile.	Time to construct tile	t < 1 day (~6-8 hours)	Machine shops	Full inventory from Bill of Materials	Quinn	10/25/2023	11/17/2023	Structural: 4.25 hours per tile. Electronics: 6.33 hours per tile.	Structrual Only: **PASS** Structural and Electronics together: **FAIL** If the electronics work is outsourced, it is possible to make 1 tile per day.
5	Decibel Reading	With the full tile set-up, play sound using the tile's speaker system. Use a decibel reader to measure the loudness of the speaker from ~5 feet above the tile. The speaker should be at maximum volume.	Decibel reading	> 80 dB at full volume	Outdoor area	Full tile	Ryan	10/12/2023	11/15/2023	Sine Wave: 83.22 ± 4.71dB. Piano: 71.12 ± 3.65dB	**PASS** This test was measuring the capability of the speakers to produce at least 80dB. If needed, the piano sound files can be altered to also reach over 80dB.

DVP&R - Design Verification Plan (& Report)

Project:	Path of Lights and Sounds	Sponsor:	Walk of Wonders and Girl Scouts of CCC		Edit Date: 12/7/2023

TEST PLAN										TEST RESULTS	
Test #	Specification	Test Description	Measurements	Acceptance Criteria	Required Facilities/Equipment	Parts Needed	Responsibility	TIMING		Numerical Results	Notes on Testing
								Start date	Finish date		
6	Maintenance Time	After being fully constructed, the tile's fastened components must be deconstructed and then put back together. This gives overhead on the time to replace any component.	Time to deconstruct and reconstruct fastened components	t < 10 minutes	Machine shop (likely), screw driver, wrench	Full tile	Quinn	10/25/2023	11/14/2023	33 minutes and 42 seconds to fully disassemble and reassemble the tile (without LEDs or grip tape)	**FAIL** LEDs would only slow down the process
7	Consumer Focus Group	User survey rating different aspects of tile design such as aesthetics, functionality, etc.	Consumer rating	rating >= 4/5	UU/ ME Labs/ Public campus location	Full tile	Sean	10/12/2023	11/14/2023	4.55 average rating with 19 participants.	**PASS** Though more participants would have been ideal, the team is happy with this result and believe more participants would not have brought the rating below 4.
8	User Test Group	User experience and follow up survey rating use of tile	User rating	rating >= 4/5	UU/ ME Labs/ Public campus location	Full tile	Sean	10/19/2023	N/A	N/A	Not completed
9	Portability Storability Rating	User test group for carrying/storing a tile	User rating	rating >= 4/5	UU/ ME Labs/ Public campus location	Full tile	Vinh	10/19/2023	N/A	N/A	Not completed
10	Programming Rating	User test group for programming the tile to play a specific song	User rating	rating >= 4/5	UU/ ME Labs/ Public campus location	Full tile	Sean	10/19/2023	N/A	N/A	Not completed

APPENDIX G – TEST PROCEDURES

Test Procedure – Friction Coefficient Test W15

Purpose: To ensure that the top surface of the tile provides traction for the user.

Scope: This test is meant to get data about the top surface's ability to give the user traction. It's a relatively rudimentary test as the only relevant coefficient of friction to the group is that of shoes on the tile itself, so the coefficient of friction will be measured for several shoes to ensure safety.

Equipment:
- Different types of shoes
- The top surface of the tile with grip tape applied
- Angle measurer

Hazards:
- The tile may slam down, hitting toes or fingers.

PPE Requirements: N/A

Facility: Any place with a flat surface large enough to place the tile on.

Procedure:
1. Increase the angle of the tile until the shoe starts slipping.
2. Record the angle at which the shoe slips with protractor for three trials.
3. Calculate the coefficient of friction for the average trial.
4. Repeat for 2 additional shoe types.

Pass/Fail: A pass is if the coefficient of friction is greater than or equal to 0.5.

Test Date(s): 11/6/2023

Test Results: As conducted, the testing provided the following results.

Table 1 – Test Results

Shoe Type	Angle of Slip (degrees)				Coefficient of Friction*
	Trial 1	Trial 2	Trial 3	Average	
Comfort	45.7	53.1	46.4	48.4	1.12
Sneaker	51.7	51.2	52.8	51.9	1.28
Boot	49.5	49.2	51.8	50.2	1.20

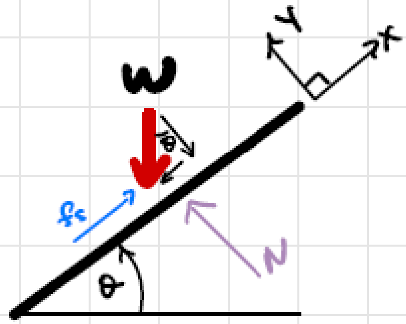
*Hand calculation for the formula on the next page.

As the test results indicate, the effective coefficient of friction passes the test for each shoe type, with an average coefficient of friction of 1.20.

Performed By: Sean Wahl

Hand Calculation:

FBD & HAND CALCULATION



Y - AXIS → $-W\cos\theta + N = 0 \rightarrow N = W\cos\theta$

X - AXIS → $-W\sin\theta + f_s = 0$

→ $f_s = \mu_s N$

→ $-W\sin\theta + \mu_s W\cos\theta = 0$

→ $\mu_s = \dfrac{W\sin\theta}{W\cos\theta} = \tan\theta$

→ $\boxed{\mu_s = \tan\theta}$

Test Procedure – Impact Test W15

Purpose: The purpose of the test is to observe the tile for damage after dropping various padded weights and jumping on the tile several times. If visible damage is observed in this preliminary test, there will most definitely be problems later on with the tile's design.

Scope: This test is to make sure there aren't any weak points on the tile or existing fatigue issues with the grip taped HDPE surface, the supports, or sides.

Equipment:

- 5, 10, 25 lbf weights
- Pads for the weights (cloth, foam, etc.)
- Human capable of jumping
- Fully constructed tile (minus electronics)

Hazards:

- Splintering or breaking of top surface.
- Weights bouncing off tile and falling towards people.

PPE Requirements: Safety glasses and shoes must be worn.

Facility: Testing should occur outside on flat ground away from people.

Procedure:

1. With weight padded, measure the mass of the weight.
2. Drop weight from approximately 3 feet off the group onto various spots of the tile. Repeat 5 times at this weight.
3. Repeat for the other two weights.
4. Measure the mass of the human.
5. Jump repeatedly on the tile, jumping as hard as possible. Vary landing on one foot versus two feet. Repeat for at least 25 jumps.
6. Inspect the tile and grip tape for visible damage, internally and externally. Make note and take pictures of any suspected damage.

Pass/Fail: Passing is having minimal to no observable damage on the tile. Greater than minimal damage is defined as damage that would effect the event or its longevity.

Test Date(s): 11/6/2023

Test Results: The test went smoothly and the tile endured heavy impacts. The human subject jumping on the tile was 186 lbf. There was only one area where so-called damage could be observed, and this was on the underside of the tile's top surface. As seen in Figure 1, the round interior supports slightly scratch the top surface upon impact.

Figure 1 – The underside of the tile's top surface after testing. The scratches are hard to see due to how minor they are.

Projected solutions to this would be to add padding or a softer buffer material at the contact point, though the scratches are so minor that they can likely be ignored. Scratches like these can also happen in the manufacturing of the tile from minor contact with hard tools.

Performed By: Sean Wahl

Purpose: Quantify ground fixation methods' effectiveness.
Scope: This test is going to provide a repeatable impact on the edges of the tile in order to quantify how securely it is fastened to the ground. This provides a way to gauge if the tiles will experience significant movement during the event. If the results are unsatisfactory, additional fixation may be required.
Equipment:

- 5 lbf weight
- Rope (at least 1 yard long)
- Yard stick
- Full tile (minus electronics)
- Stakes
- Rubber pads

Hazards:

- The tile's edge can break and potentially create shrapnel or a sharp, exposed surface.
- Swinging weights can hit nearby participants.

PPE Requirements: Safety glasses must be worn at all times.
Facility: Wherever intended fixation method is applicable (grass or dirt area for stakes, concrete for rubber pads, etc.).
Procedure:

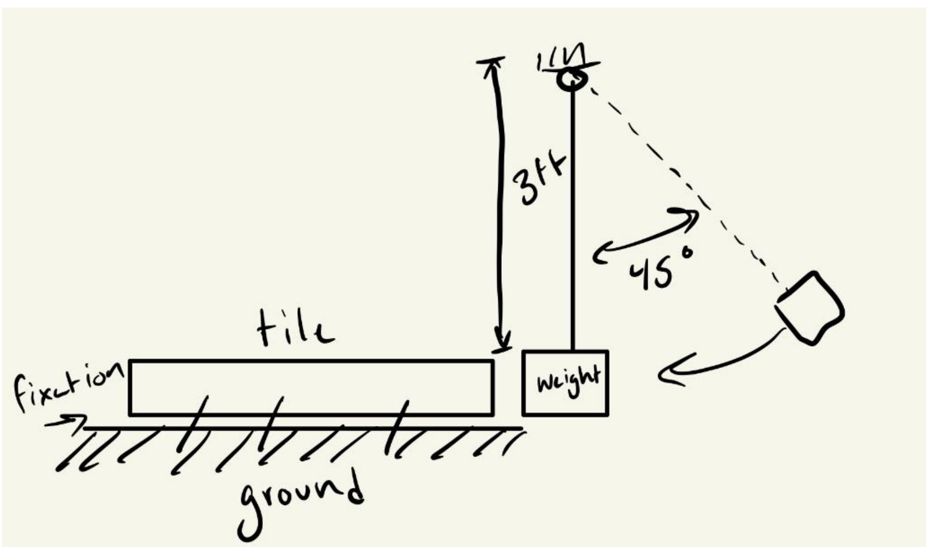


Figure 1 – Test Setup

1. Procure necessary materials and arrive at the testing location.
2. Verify all participants are wearing safety glasses.
3. Attach weight to rope.
4. Mark the tiles' initial location, parallel to the edge the weight will be contacting.
5. Have one participant hold the rope approximately 1 yard above the ground such that the weight is not touching the ground. This should be directly over the edge of the tile.

6. Bring the weight backwards so that the rope is taut and forms an approximate 45° angle with the ground.
7. Release the weight and let it swing down and hit the edge of the tile.
8. Repeat steps 6 and 7 25 times.
9. Measure the distance that the tile has moved during the test. If 25 repetitions were not reached, record the distance that the tile moved and note how many swings it took.

Pass/Fail: A pass is if the tile sustains 25 impacts and moves less than an inch for the given fixation method.
Test Date(s): 11/6/2023
Test Results: As conducted, the testing provided the following results.

Table 1 – Test Results

Fixation Method	Ground Material	Number of Impacts	Distance Moved (in)
Stakes	Grass	25	3/8
Rubber Pads	Concrete	1	5/4

As the test results indicate, the stakes holding the tile on grass successfully prevented major slippage and the rubber pads on concrete failed. With these results in mind, the team advises that the tiles only be used with the rubber pads only with mature audiences who are less likely to kick and mess with the tiles. Staked down tiles are still regarded as safe for all audiences.

Performed By: Sean Wahl

Purpose: The purpose of this test is to ensure that the time to manufacture and assemble a single tile is reasonably doable in a day.

Scope: This test is to see how long it would take to make every component that goes into a finished tile and then assemble it. Because the team is recommending the electronics to be outsourced, this will have times with and without the electronics' manufacturing time.

Equipment:

- Access to machine shops (files, drills, chop saw, miter saw, mill, measuring equipment, stencils)
- Material for one full tile
- Assembly tools (screwdriver, socket wrench, box cutter)
- Safety equipment (glasses, long pants, closed-toe shoes)
- Timer

Hazards: General machine shop hazards, please adhere to machine shop rules.

PPE Requirements: Safety glasses, long pants, and shoes must be worn.

Facility: In Mustang 60 or the Hangar machine shop.

Procedure:

1. Lay out all the parts and ensure that everything to build one tile is on hand.
2. Start the timer, begin following the manufacturing plan.
3. If machines are unavailable or there is down time due to machine stop issues, stop the timer. This prevents unworkable time from being accounted for in the manufacturing time. Restart the timer when resuming work.
4. Once all custom parts are manufactured for one tile, stop the timer. Record this as the manufacturing time.
5. Again, ensure that all custom and bought parts are on hand for assembly.
6. Start the timer from scratch, assemble one full tile following the assembly plan.
7. Once complete, stop the timer and record this as the assembly time.

Pass/Fail: Passing is if 1 tile takes less than 1 day (8 working hours) to fully manufacture and assemble.

Test Date(s): Month of November

Test Results: As conducted, the manufacturing and assembly of the tiles took much, much longer than originally expected. For this reason, the method of using a timer during this process was not viable. Instead, each team member kept mental track of how much total time they were putting into the project. Those results are tallied in Table 1.

Table 1 – Test Results

Team Member	**Structural** Manufacturing and Assembly [hr]	**Electronics** Manufacturing and Assembly [hr]
Quinn	45	0
Vinh	24	0
Jason	40	6
Ryan	0	100
Sean	20	87
Total	**129**	**193**

Taking into account the fact that 28 tile structures and 30 electronics assemblies were made, a tile's structure takes around 4.61 hours to make and an electronics assembly takes 6.43 hours. Together, this sum fails the test, at 11.04 hours to make a single tile. However, the team is highly recommending the outsourcing of the electronics work as it is too difficult and tedious for the average person with no background in electronics, and even for those with a background it is monotonous and laborious. This would bring the total time to make one tile back down to around 4.61 hours, which passes the test.

It is believed by the team that if the next person to manufacture these tiles uses the same or better equipment as the team did (outlined in the User Manual's manufacturing sections), it is possible to make at least 1 tile per working day. Please keep in mind an assembly line mindset in this process though, as 1 process should be done on all the tiles at one time, not completing 1 tile fully and then restarting for the next.

It is also essential to keep in mind that the 8 hours to make 1 tile benchmark for pass/fail was completely arbitrary. The team chose this because under the original circumstances of the project, the month of November would be used by the Girl Scouts manufacturer to create at least 25 tiles, giving him 1 day per tile (more or less).

Performed By: Entire Team

Test Procedure – Decibel Test W15

Purpose: The purpose of this test is to try and gauge how loud the speaker can be during operation.
Scope: This test covers two of the instruments in the sound files that the team is providing for normal operation of the tiles. To get a wide range of sound, the lowest note and highest note provided will be played.

Equipment:

- 1 tile capable of playing sound
- 1 phone with a decibel meter app (iPhone)

Hazards: Loud noises may cause hearing damage, don't put speakers up to ears.

PPE Requirements: None.

Facility: Classroom 192-118, the Mechatronics lab.

Procedure:

1. Use a study to estimate the phone's calibration uncertainty
2. Hold phone 4±.25 ft above tile with decibel meter app open.
3. Activate tile speakers to play C3 followed by C6.
4. Have decibel meter app record the max dB reading, write this number down.
5. Repeat for a total of 5 tests, with two separate instrument's sound files.

Pass/Fail: Passing is if the average decibel reading for the instrument is above 80dB (not including uncertainty bounds).

Test Date(s): 11/15/2023

Test Results: As no official decibel meter could be acquired, a study[1] by Enda Murphy and Eoin A. King, respectively from University College Dublin and the University of Hartford, will be used to estimate the iPhone's calibration uncertainty. From equation 3.14b from the ME 236 textbook[2], the population mean of iPhone apps accuracy in reading decibel levels as gathered in the study can be found as follows:

$$\mu = \bar{x} \pm \frac{ts}{\sqrt{n}} \quad \text{w/} \quad \bar{x} = 2.92\text{dB},\; s = 6.80\text{dB},\; n = 1052$$

$\rightarrow t = 1.96$ @ 95% confidence

$\rightarrow 2.92 \pm 0.41\text{dB}$ ← dB reading error from study

Assume at worst

$$u_{cal} = \pm 3.33\text{dB}$$

At worst, we will assume that the calibration uncertainty is the outside of this range mirrored, meaning **±3.33dB.** This is conservative but also assumes a normal distribution of iPhone decibel reading errors. This is not a true calibration but is sufficient for the purposes of this test.

The second source of uncertainty comes from the resolution of the decibel app used. The “Decibel X” iPhone app reads up to the first decimal place, meaning that its resolution uncertainty is half of the lowest value. This turns out to be **±0.05dB.**

Lastly, statistical uncertainty comes from the actual data collected, as seen in Table 1.

Table 1 – Test Results

		Maximum Decibel Reading (dB)				
Trial #		1	2	3	4	5
Instrument	Sine Wave	83.9	87.3	83.3	80.8	80.8
	Piano	71.6	72.3	72.0	70.1	69.6

The statistical and then total uncertainty for each instrument can be calculated as follows, using equation 3.15b again.

Sine Wave

$\bar{x} = 83.22dB$

$s = 2.68dB$

$n = 5$

$t = 2.776 @ 95\%$

$u_{tot} = \pm\sqrt{u_{cal}^2 + u_{stat}^2 + u_{res}^2}$

$u_{stat} = \pm \frac{ts}{\sqrt{n}} = \pm 3.33dB$

$u_{tot} = \pm\sqrt{3.33^2 + 3.33^2 + .05^2}$

$u_{tot} = \pm 4.71dB$

Piano

$\bar{x} = 71.12dB$

$s = 1.20dB$

$n = 5$

$t = 2.776 @ 95\%$

$u_{stat} = \pm \frac{ts}{\sqrt{n}} = \pm 1.49dB$

$u_{tot} = \pm\sqrt{3.33^2 + 1.49^2 + .05^2}$

$u_{tot} = \pm 3.65dB$

The Sine Wave’s decibel reading is therefore **83.22 ± 4.71dB** and the Piano’s is **71.12 ± 3.65dB**. As this test was to measure the speaker’s capability to produce over 80dB, the team can be reasonably confident that the Sine Wave sound files did just that. If the Piano wishes to be higher, the sound files should be alter to dull the peaks and raise the lower parts of the sound file. By all intents and purposes, the electronics **PASSED** this test.

Performed By: Sean Wahl

References:

[1] Murphy, Enda, and Eoin A. King. “Testing the accuracy of smartphones and sound level meter applications for measuring environmental noise.” *Applied Acoustics*, vol. 106, 5 Jan. 2016, pp. 16–22, https://doi.org/10.1016/j.apacoust.2015.12.012.

[2] Thorncroft, Glen E. “Engineering and Measurement Data Analysis” from the ME 236 Coursepack, Spring 2020 Revision.

Test Procedure – Maintenance Time W15

Purpose: The purpose of this test is to accurately describe the amount of time required to performance maintenance on the tiles.

Scope: This test will result in the average time required to complete disassembly and reassembly of a tile, which will be used as an estimate for the maximum time required to perform any maintenance required of the tile. Electronics will be treated as 1 unit in this test.

Equipment:

- 1 tile (minus LEDs at this stage)
- 7/16" deep socket with bit attachment (to attach to a drill)
- An adjustable speed drill
- Phillips-head screwdriver
- Timer

Hazards: Use caution with power tools.

- Drill could poke or lacerate the experimenter.
- Screwdriver could slip and bruise or pinch the experimenter.
- Aluminum and plastic supports could cut the experimenter if handled hastily.

PPE Requirements: Safety glasses, long pants, and shoes must be worn.

Facility: In Mustang 60, the Hangar, or any on campus workspace.

Procedure:

1. Completely assemble the tile and set it up as though it was ready for use.
2. Set out all tools.
3. Start timer.
4. Begin disassembly at an even, unrushed pace.
5. Once tile is completely disassembled, begin reassembly at the same, unrushed pace.
6. Stop the timer once the tile is completely assembled again. Record this time.

Pass/Fail: Passing is if the disassembly and reassembly of one tile takes less than or equal to 10 minutes.

Test Date(s): 11/14/2023

Test Results: Note: the tile used for this test did not have LEDs or grip tape at this point, though it did have an electronics assembly attached with Velcro as designated. As conducted, the tile took an unprecedented **33 minutes and 42 seconds** to completely disassemble and reassemble. For this reason, the test has been failed.

Part of the reason that this took so long is the use of locknuts in the design. The team is putting in the report that these should be changed to regular nuts with lock washers, as they will be much, much easier to install at that point. If done by hand, this task would've taken much, much longer, though the power drill speeds it up considerably.

It is also of note that if the base is disassembled after the LEDs have been applied, additional double-sided tape will be required. This is to reapply the LED tape after reinstallation of the sides. Grip tape being applied on the top surface would only marginally slow down the process, as the screws will still be able to come out of their holes without full removal of the grip tape. However, handling the top surface with grip tape must be done so with care, as it can very quickly dry out and damage your hands with its grit.

Performed By: Quinn Stephens

Test Procedure – Consumer Focus Group W15

Purpose: The purpose of this test is to rate the tiles appeal to general consumers. This includes visual aesthetics, sound, and the general experience that comes from using the tile**.**

Scope: This test is to get consumer feedback to edit the tile as needed to be more appealing.

Equipment:

- 1 fully working tile
- Willing participants
- Survey (available at QR code →)

Hazards: None.

PPE Requirements: None.

Facility: In a room able to be darkened, preferably near the UU or Engineering IV.

Procedure:

1. Gather willing participants. Brief them on their role in rating the tile. Have them scan the QR code at this point so they can see the survey's fields.
2. Demonstrate the tile's sound, general look, and brightness. Be willing to answer any questions and demonstrate further upon participants' requests.
3. Allow participants to use the tile themselves.
4. Ensure the participants have completed the survey.
5. Tally up results, calculate average rating.

Pass/Fail: Passing is a 4/5 or better overall rating. Around 25 people should be surveyed.

Test Date(s): 11/14/2023

Test Results: From 19 participants, consult the data in Table 1.

Table 1 – Test Results

Category	Brightness	Volume	Looks	Quality	Overall
Average Rating	4.68	4.63	4.15	4.74	4.55

Based on these results, we are considering this test a success. Though only 19 participants were able to be gathered due to time constraints, the tile fared well in the ratings. The team is more than happy with a 4.55 overall tile rating.

Performed By: Sean Wahl

Test Procedure – User Test Group W15

Purpose: This test is to rate the tile based on performance with users. Users here shall be defined as both operators and tile-hoppers.

Scope: Specifically, this test will be ranking tile usability, portability/storability of the tiles, and programmability of the tiles.

Equipment:

- 2 fully working tiles
- 2 spare tiles
- Willing participants

Hazards: None.

PPE Requirements: None.

Facility: In a room able to be darkened, preferably near the UU or Engineering IV.

Procedure:

1. Gather willing participants.
2. Ask participants to use the working tiles and rate the usability of the tiles' intended use.
3. Ask participants to heft the spare tiles, move around with them, and put them down; then have them rate the portability/storability of the tiles.
4. Ask the participants to use the tile coding software and have them rate the programmability of the tiles.
5. Tally up all results and calculate the average for each of the three categories.

Pass/Fail: The three categories (usability, portability/storability, and programmability) are considered separately. Passing in each category is achieving a rating of 4/5 stars or better, with at least 25 users having participated in the group.

Test Date(s): N/A

Test Results: Not completed due to time constraints.

Table 1 – Test Results

Category	Usability (UTG)	Portability/Storability	Programmability
Average Rating			

Performed By: N/A