60-223 Intro to Physical Computing

The Catheter Alarm By The Waterlilies: Final Project

Introduction

Evelyn **Justin** **Evan**

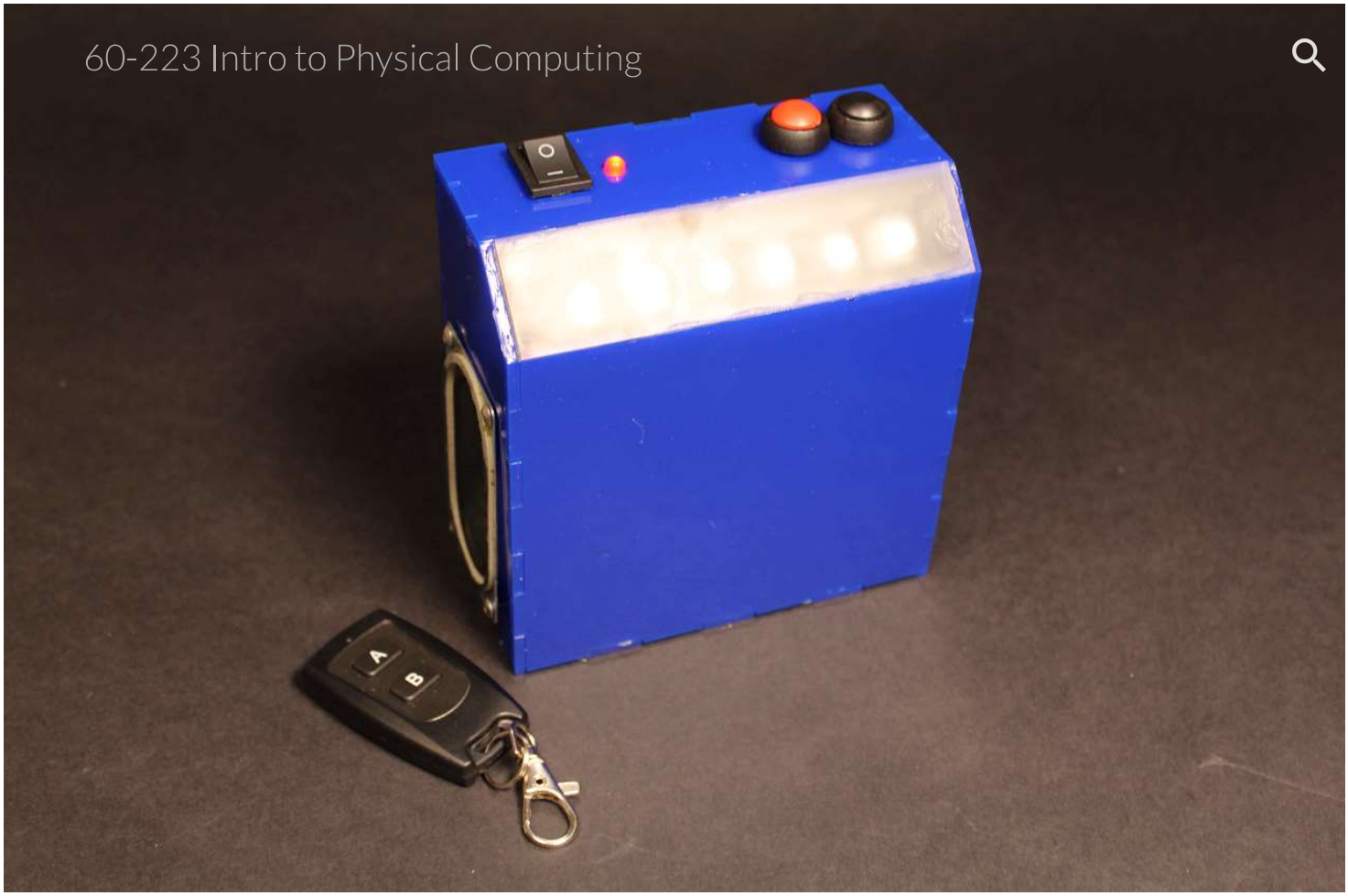
This is a photo of each of the group members that worked on this project. Evelyn is on the left smiling in a chair, Justin leaning down with his hands on his knees smiling at the camera, and Evan is on the right holding the wireless key fob leaning into his left. The C-alarm is in the center of the 3 on top of a table just below Justin.

What We Built

The Catheter Alarm is an assisted device for client Monica who needs to catheterize every 3 hours. She is a nurse and wants the alarm not to make a beeping sound because she is desensitized by the sounds. She enjoys music in 80's rock music such as Journey and Bon Jo. She is also a Trek-y but she only enjoy the original Star Trek. The alarm begins when the power button is switched on. There are Velcro straps on the back so it may be mounted on the back of her wheelchair or taken off and set on a table.

Front view of the Catheter Alarm and remote. The Catheter Alarm is a virtual box, made out of a blue acrylic on all sides. There is a 45 degree angle cut on the top of the device that is made out of frosted acrylic. There are two buttons seen on top to the right. One button is red and one is blue. On the left top there is a black power switch and a red power indicator LED to the right of the switch. A Key fob is on it's side balance on the front face of the C-alarm. It is on it's side with the buttons facing the camera.

Final Product

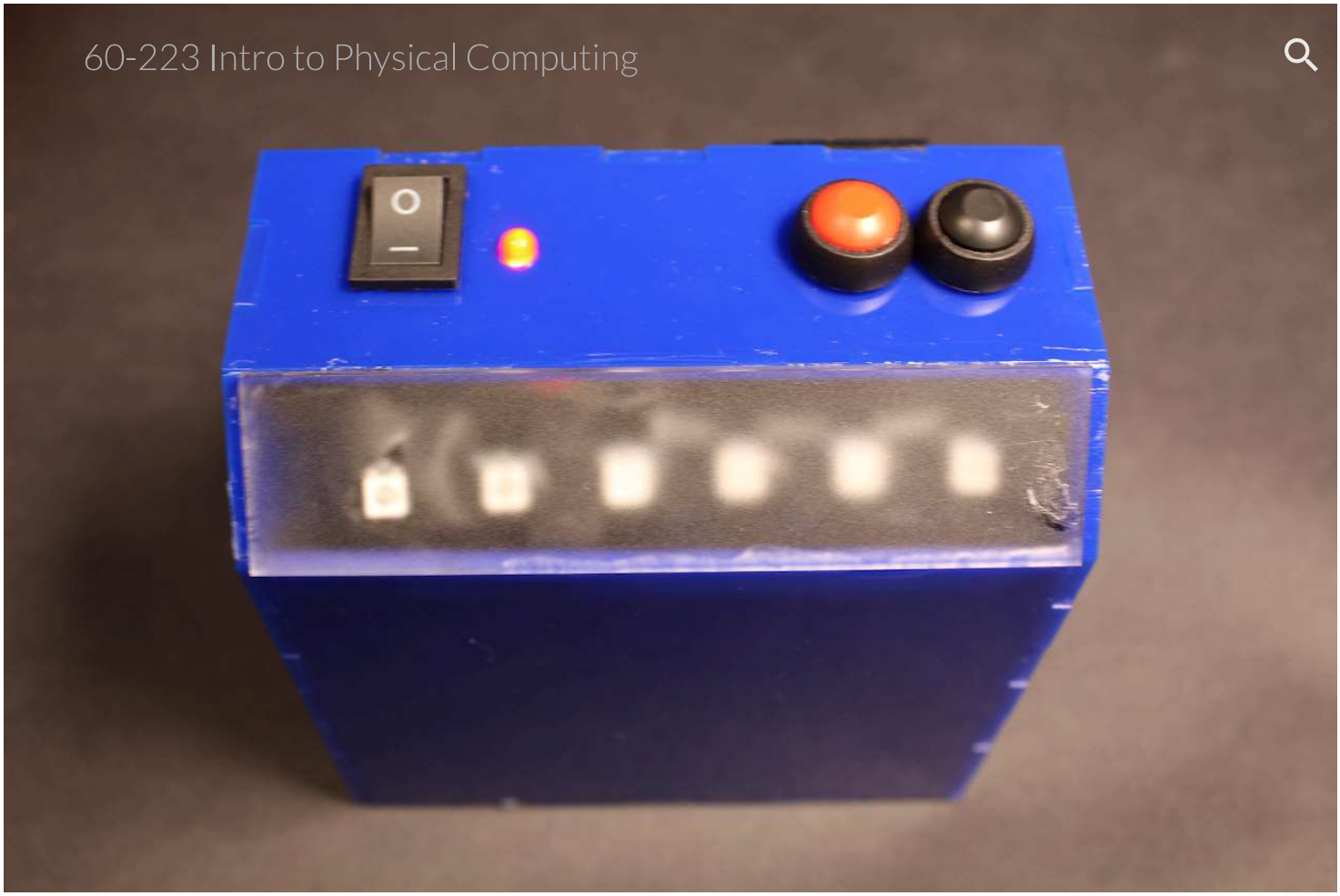


Final photo of the Catheter Alarm

Left view of the Catheter alarm

Right-side view of the Catheter alarm

Top-side view of the Catheter alarm



Top-side view of the Catheter alarm

Front 3/4 view of the Catheter alarm

Back 3/4 view of the Catheter alarm

Narrative

When Monica wakes up, she rolls out of bed and turns on her C-alarm. She doesn't have a consistent schedule, so when the alarm turns on, that's when the timer starts. She goes through out her morning, getting ready for the day. The alarm goes off in the middle of her cooking brunch. She's a distance away so she pulls a wireless key fob out to quiet the alarm from afar.

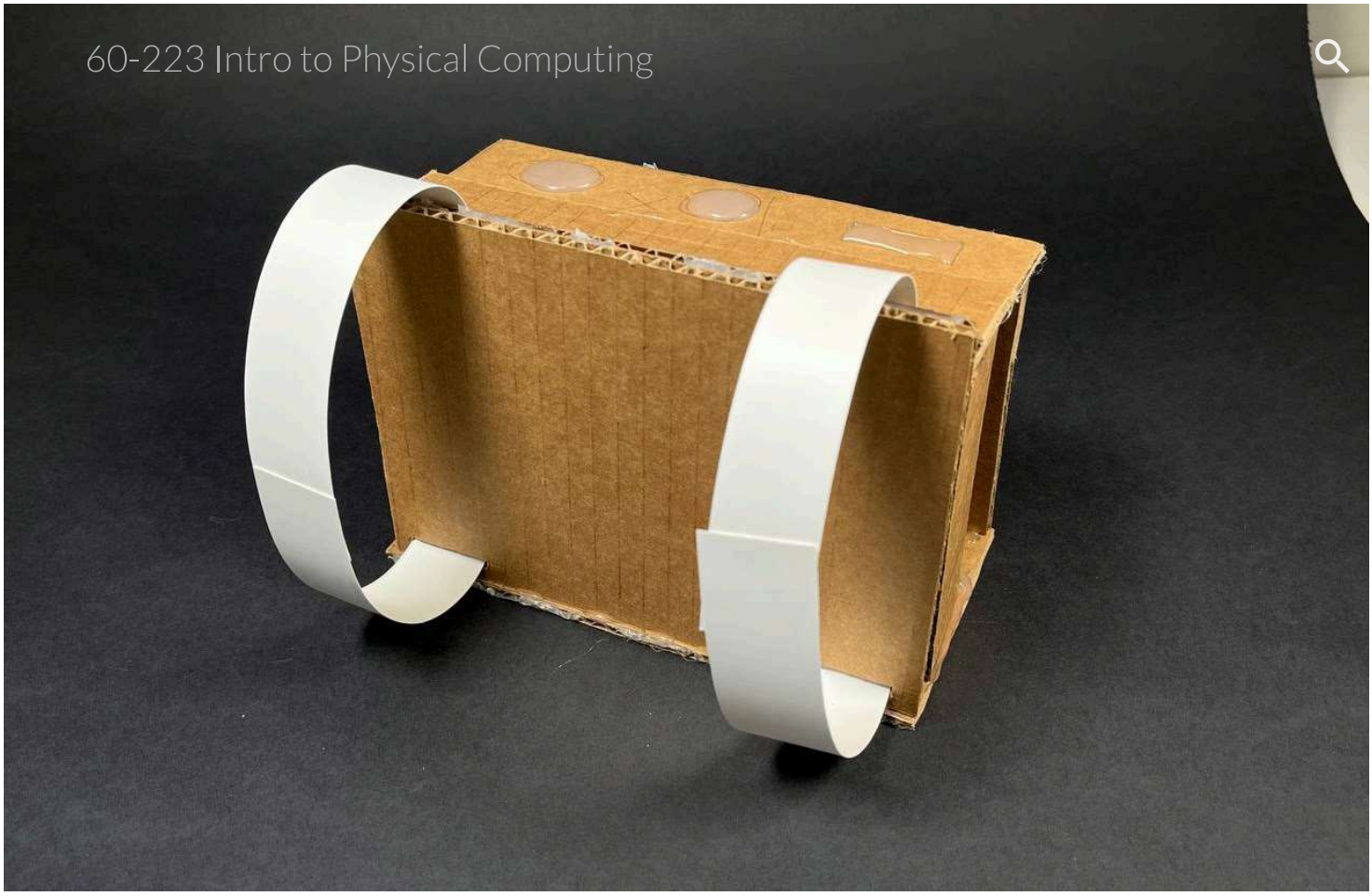
As she packs up to head out to the farmers market, she takes her alarm with her. Once she arrives, she sets up her wheelchair and straps the C-alarm onto the back of the wheelchair. As she finished purchasing her goods, she hears the alarm start. Monica turns around and presses the red button on the top of the alarm to put it asleep, restarting the timer.

Whenever she arrives at home, she is tired and is going to head to bed. That is when she turns off her alarm with the power switch in order to start the same process again the next day!

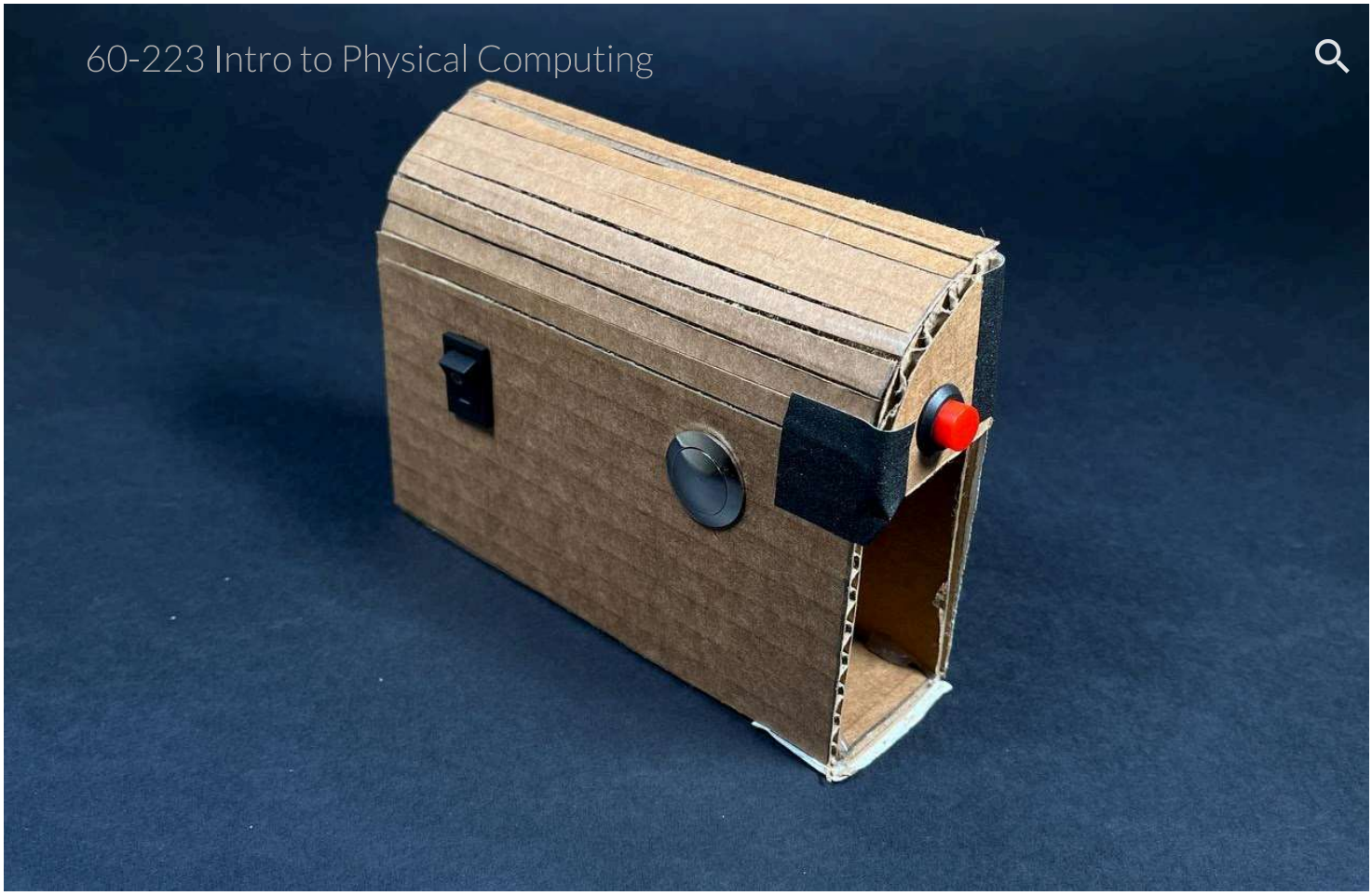
Prototype

60-223 Intro to Physical Computing

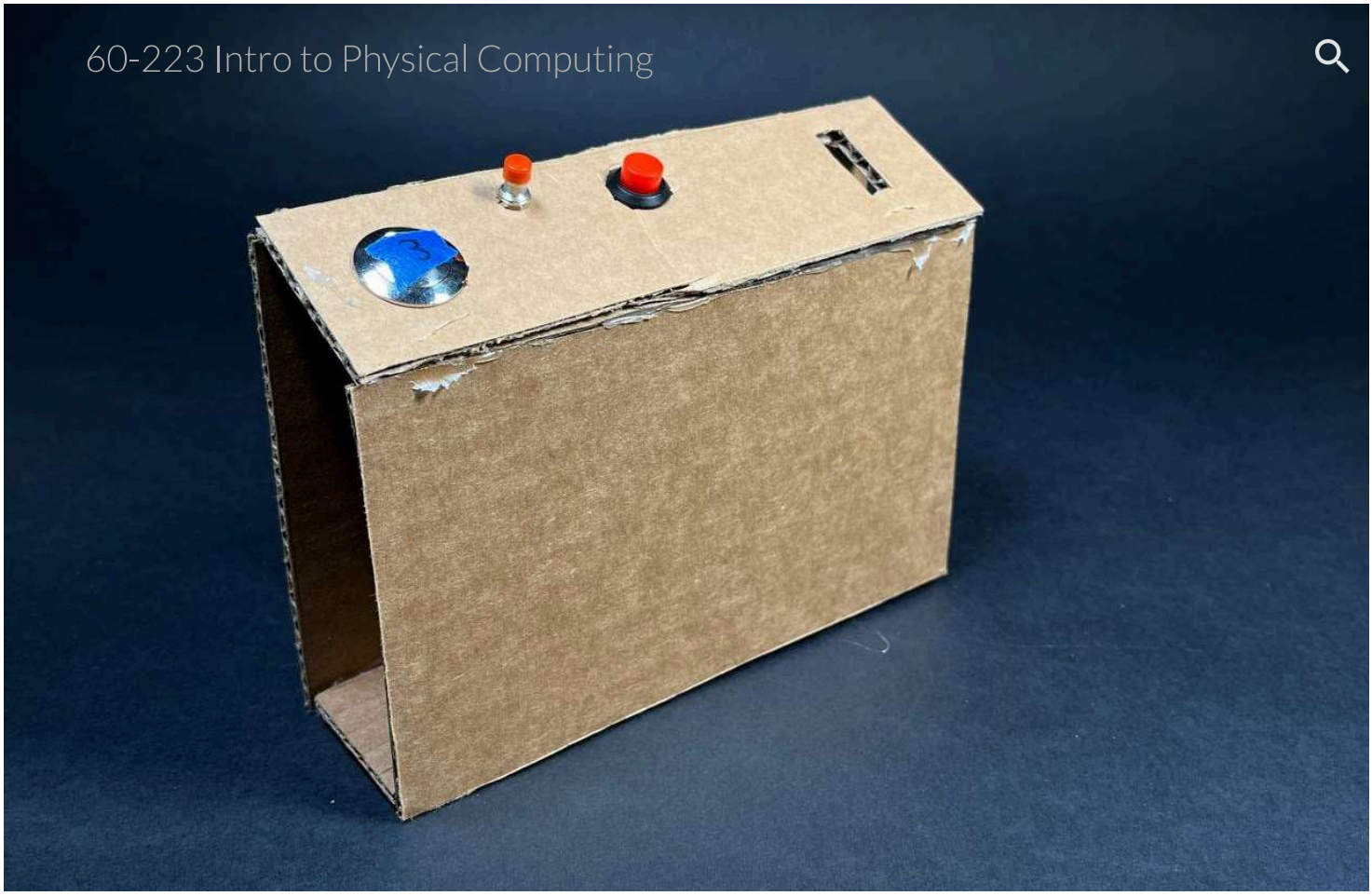
The main question we were able to prototype for was: how is this going to mount on Monica's wheel chair and how do we keep this device ergonomic and out of the way, while also having enough space for components? We built a total of 4 prototypes, all with similar aspects and functionality.



Prototype #1 with straps and buttons on top



Prototype #2 with a button on the front and side and a switch on the front. A curved hand placement.



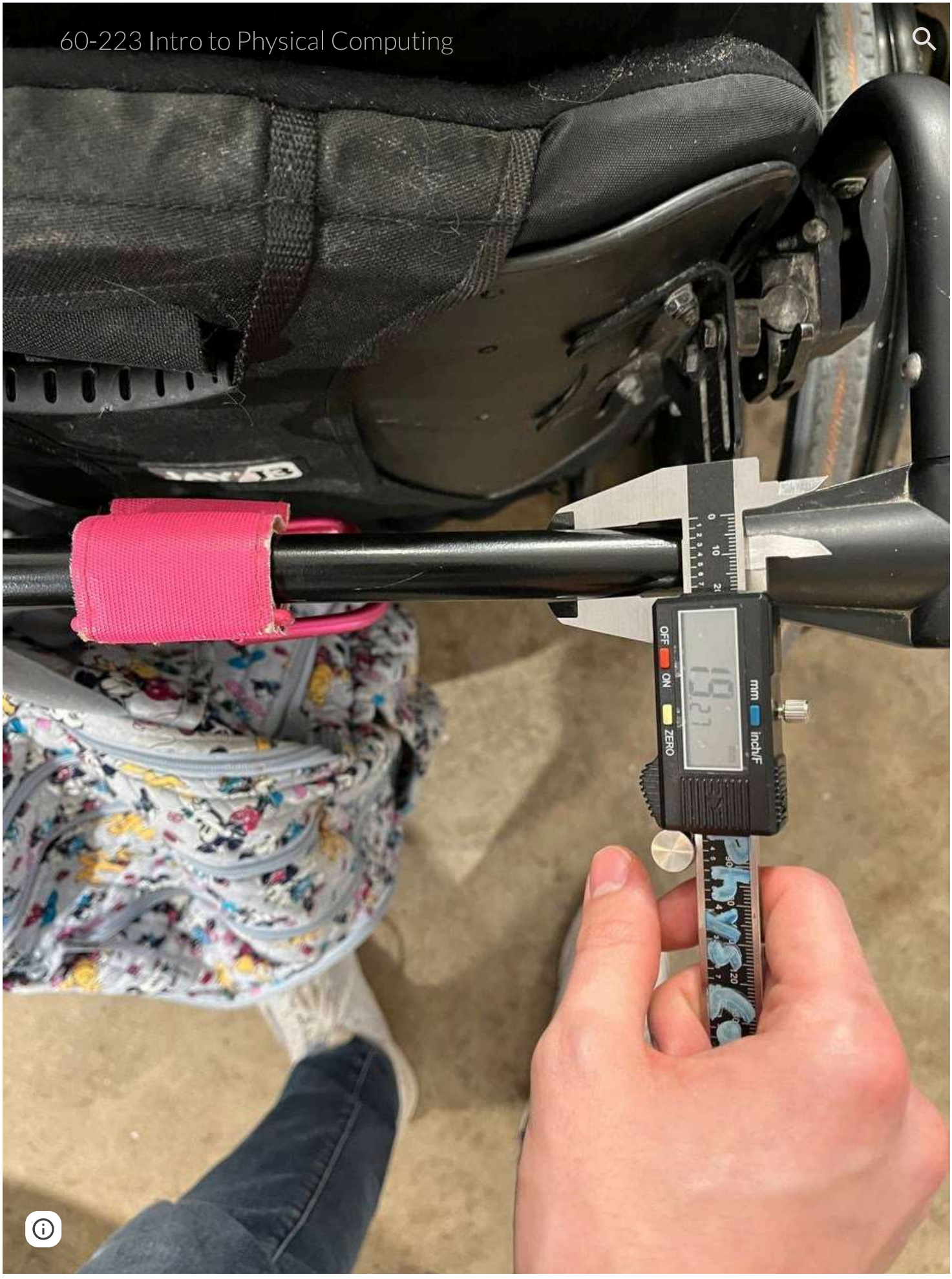
Prototype # 3 with slanted top with buttons and functionally for schedule organization of the alarm set.

Prototype #4 with 3 buttons on top and there is a curved transition edge from the top side to the front side of the device.

3/4 view: Trying to fit prototype #4 between the back pole of the wheelchair and the backrest of the wheelchair


60-223 Intro to Physical Computing

Top view: Trying to fit prototype #4 between the back pole and the backrest of the wheelchair


60-223 Intro to Physical Computing
OFF
ON
ZERO
mm
inch/F

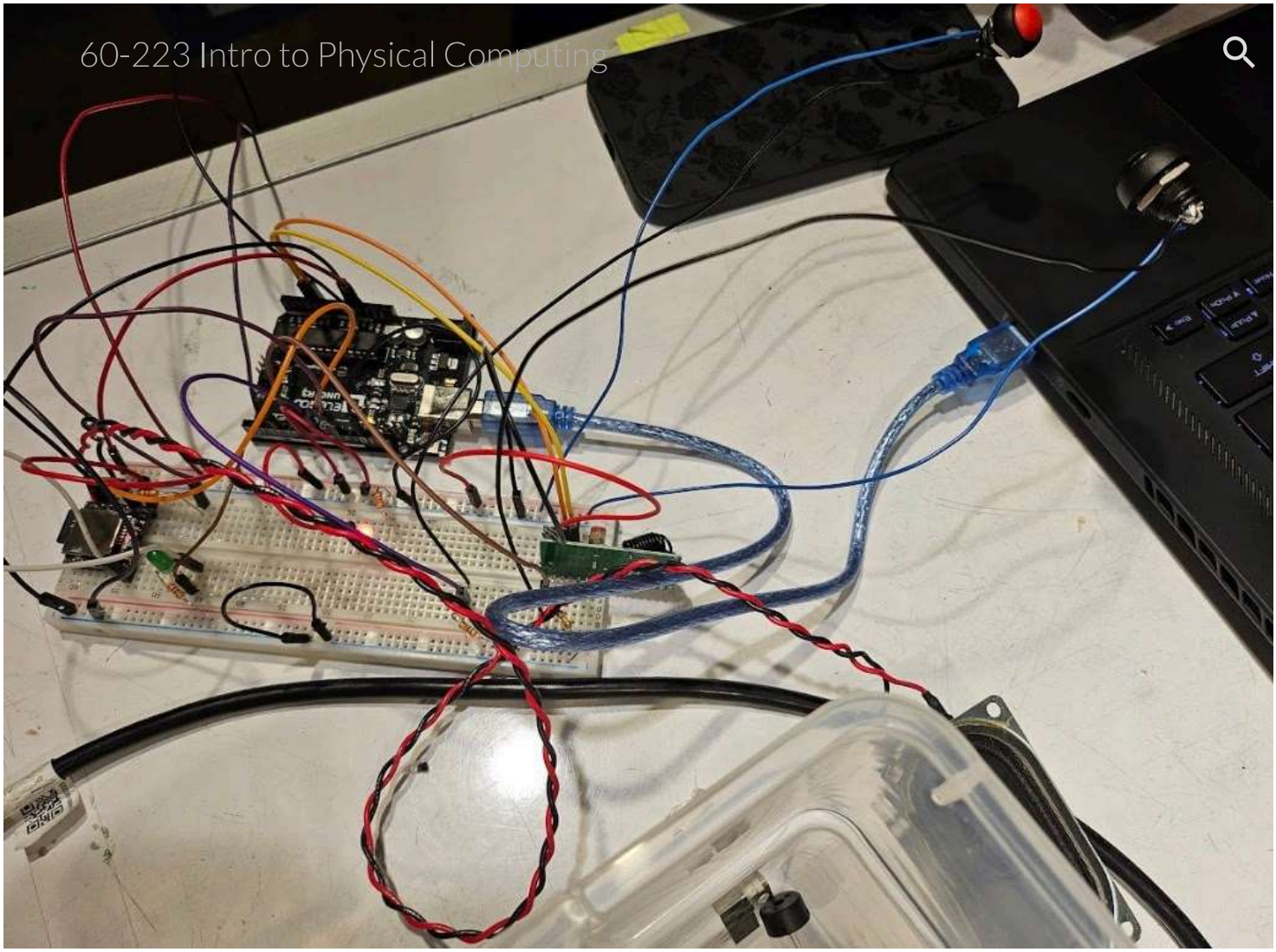
Measuring the diameter of the pole of Monica's back handle of her wheel chair

Most of the prototypes we had were cardboard with no electronics. The goal for our prototype was ergonomics first. You will see above several variations of the same design each one considering how Monica was supposed to silence the alarm and be able to see and hear when the alarm triggered.

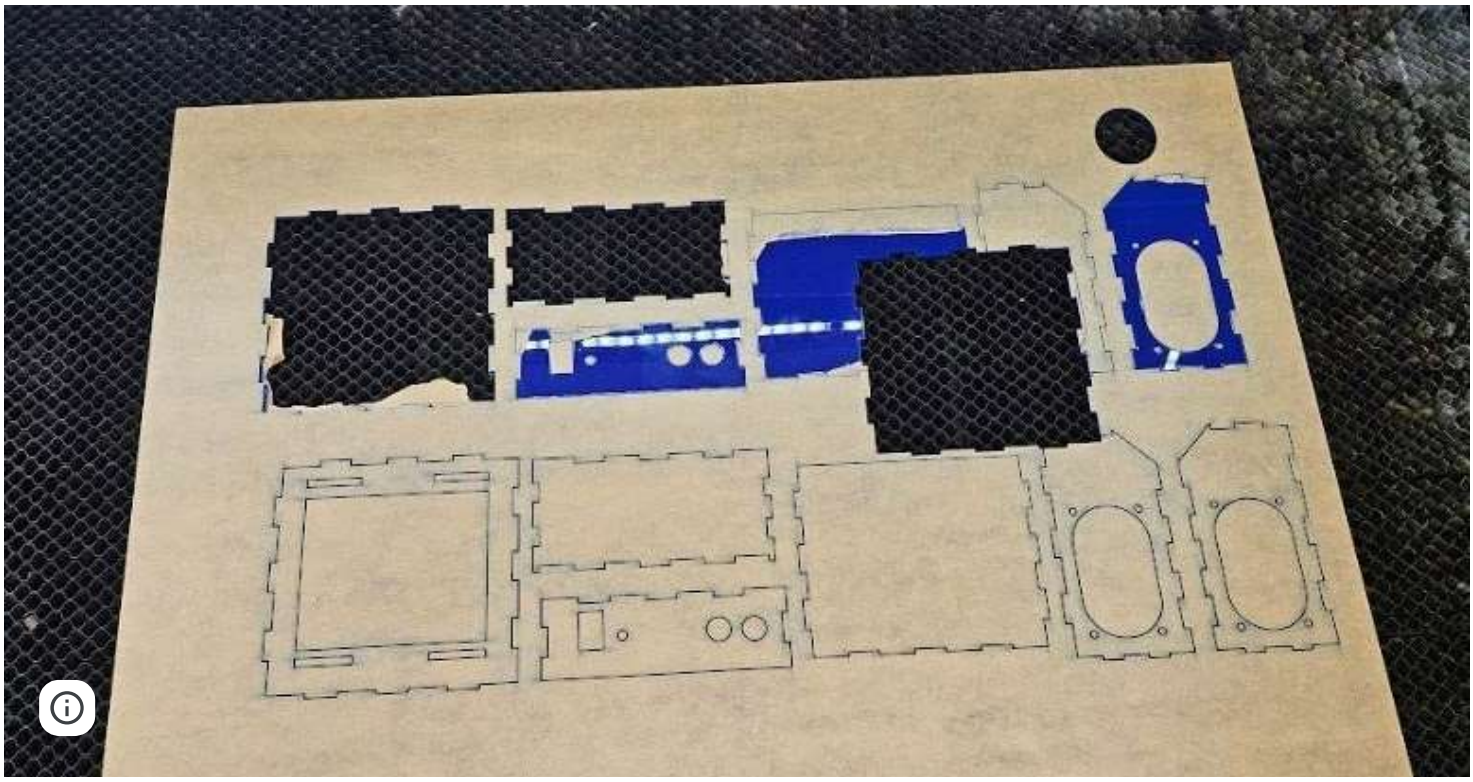
We initially were unable to visit Monica in person and had to make some assumptions for our prototype. One, was the assumptions that her wheel chair had arms. Each prototype had this in mind. When we got to see her in person for prototype feedback we swiftly moved the design to the bar on the back of her wheel chair. We made sure it was accessible to her as she turned around. The initial concept of having straps easily moved to this version of the prototype.

We found out many of Monica's preferences, like how she wanted the alarm on the back of the wheelchair. She also told us that having the alarm dynamically be able to start at any time made it much easier to use. With these ideas, we were able to very firmly determine the capabilities of the device itself.

Process



Wires during coding process



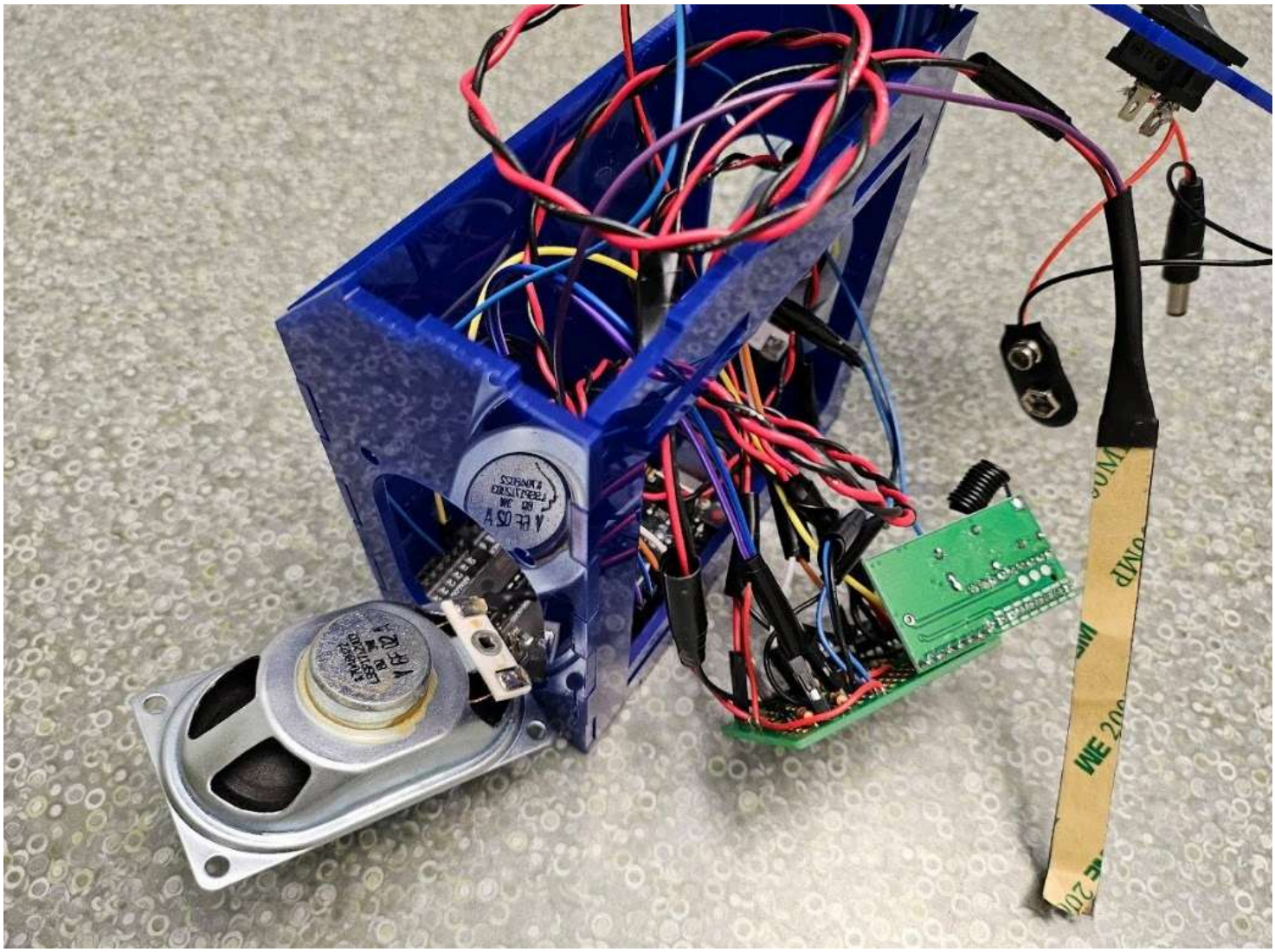
Laser cut pieces

Glued pieces of laser cut acrylic

60-223 Intro to Physical Computing

Testing speaker size and measurements

Gluing the acrylic pieces together



Box with the wiring and pieces inside

We had fabricated a 3D print, but it was too thin and could not be redone due to an overly long waiting line. Instead, we laser cut the box and made sure everything fit inside. Our schedule was pushed back a lot due to not being able to meet our first client and being transferred to another. We also ordered some parts in which one was not available. So, we had to compromise and not have a battery pack available for the product. Otherwise, we successfully created a clear acrylic piece that allowed the LED strip light to shine through without exposure. We managed to just push back a few days and keep on schedule otherwise.

Discussion

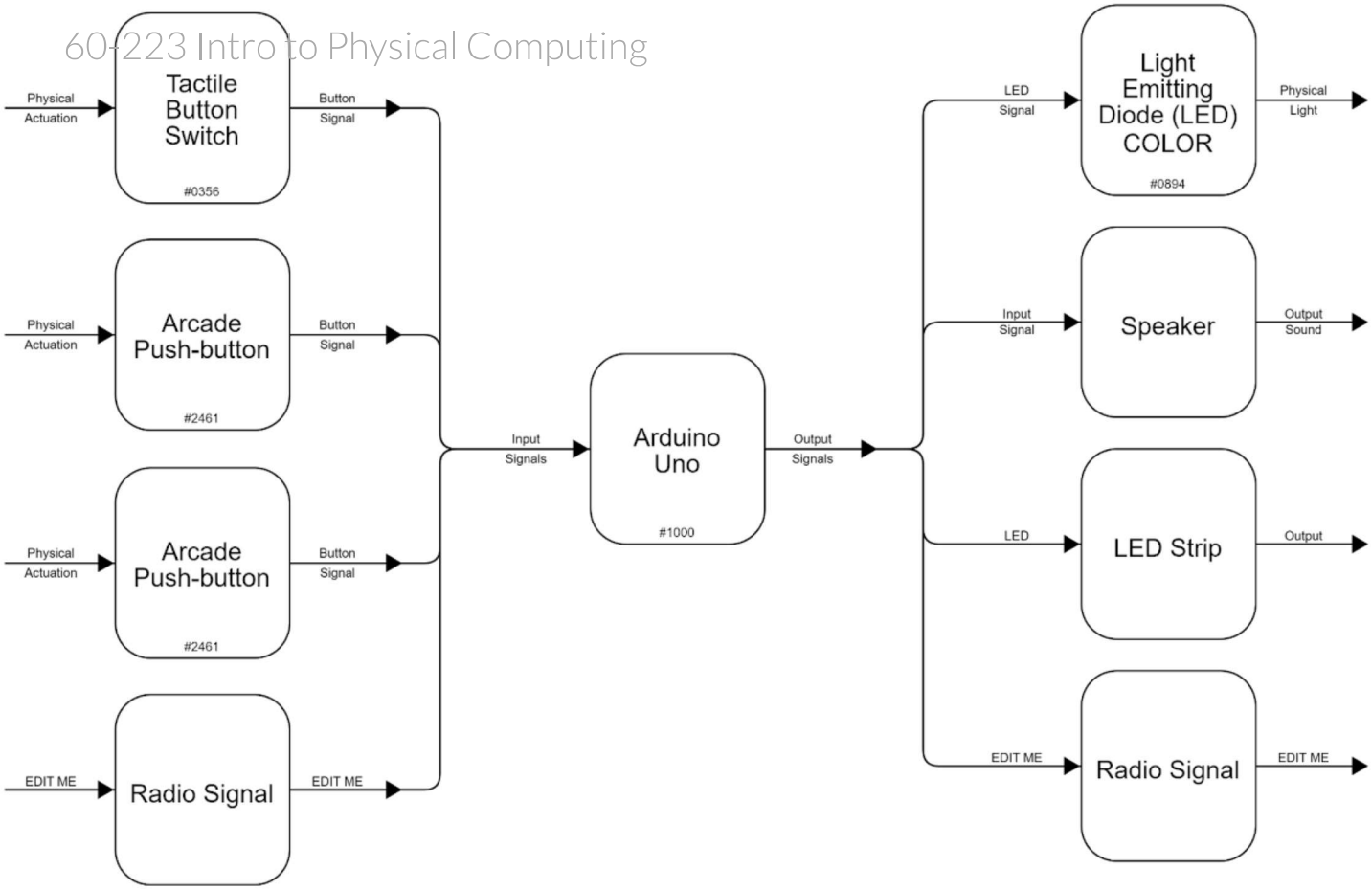
We are very happy with how the project turned out, other than the battery pack not being available. We had a lot of fun working with Monica. Trying to help her out with a very real problem she has made it challenging as well as interesting. We were able to see a snip bit into her life as a nurse and a tech nerd. Without this project, we could not have been able to see how to approach user-centered projects from a brand new perspective, in which we are very thankful for.

During the first critique we were able to finalize our idea and start the building there after. We chose what components would be the most useful in the most simple way. After the final critique, we found that there were many insightful comments such as the blinking of the power LED to lower the cost of resources as well as placing the LED strips on the other side of the box for specific user preferences. We also came across a problem of not being able to find a battery pack that was thought to be available. This made it impossible to change the battery, rendering the device to function temporarily. Another point was to make sure the speaker was behind some form of hard form so that it isn't exposed. We got many positive comments about how finely tuned the design of the box was as well as the sizing and compactness of it.

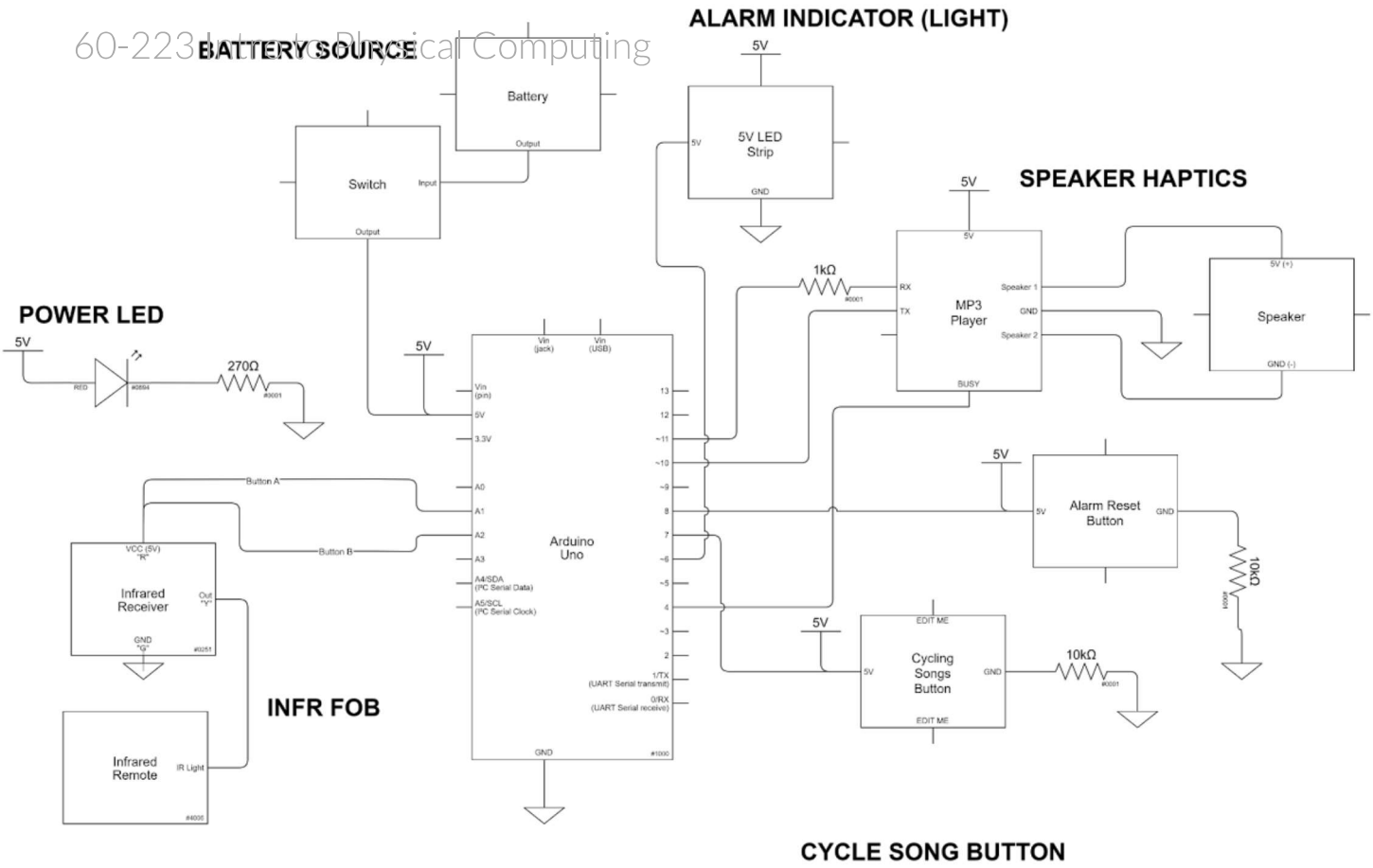
What we have learned during the process of building the Catheter Timer is that it was difficult but in a fun way. We had some challenges starting up with the MP3 player as well as connecting it to a speaker, but we were able to smooth it out in the end. Another point to make was that we were very excited to make different prototypes to test out our ideas in order to make sure Monica got what she wanted. This could mean addressing ideas that we might've thought as silly but in turn becoming a feature to the product. Some trouble we did have was that the soldering for the wires kept coming off, which might've been due to the constant handling of the wires that were constantly bent and straightened out.

If we could build another iteration of this project, we would definitely make sure there was a battery pack on the back of the project with the logo in the front. We would also make the form factor more smooth around the edges in order to combat any sharp edges that could be dangerous for the user.

Technical Details



Box Schematic diagram



Schematic diagram

Code

```
/*  60-223 Intro to Physical Computing
  Author: Evelyn Bang, Evan Alexander, Justin Song
  Title: Catheterize Timer
  Description: Timer goes off every 3 hours and plays a song in a playlist while LED
               strip lights until a button is pressed to turn it off. A red LED light
               is used as an indicator to show if the timer is on or off.
  PIN MAPPING:
  PIN |    MODE    |    COMPONENT    | DESCRIPTION
  ----------------------------------_-----------------------------------------------
  A1  |    INPUT   |REMOTE RECEIVER A| Receives signal from a two button keyfob to turn off the
alarm + reset the timer
  A2  |    INPUT   |REMOTE RECEIVER B| Receives signal from a two button keyfob for shuffling
through playlist of songs
  08  |    INPUT   |    RED BUTTON   | Turns off alarm and restarts the 3 hour timer
  07  |    INPUT   |   BLACK BUTTON  | Plays next song on the list
  NA  |   OUTPUT   |     SPEAKER     | Plays alarm sound (connects to MP3 player)
  11  |    INPUT   |  MP3 PLAYER RX  | Connecting MP3 player to Arduino
  10  |    INPUT   |  MP3 PLAYER TX  | Connecting MP3 player to Arduino
  02  |   OUTPUT   |    LED STRIP    | Turns on when alarm goes off (PololuLED Strip)
  NA  |   OUTPUT   |    LED SINGLE   | Turns on when device is on (connects directly to power)
  LINKS:
    Forum link to MP3 Player         : https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299
    Youtube Link to Song Playlist    : https://www.youtube.com/playlist?
list=PLri2B0JyFfoYqYZyf_NyMQUEaabbSORFC&jct=w2Nn5378-KGVo57GNyAiHdORdLDtEw
    Library File: https://drive.google.com/drive/folders/13ZcTia8K_bbBIkyIEfAHw9lvFljyCtIP?
usp=sharing
*/
#include "mp3tf16p.h"
#include <PololuLedStrip.h>

/*
  GLOBAL VARIABLES
*/
// Timer
unsigned int timer = 0;
unsigned const int threeHours = 10800000; //Use 10000 for testing
bool alarm = false;
```

```
boot alarm = false;
// Remote
const int remoteA = A0;
const int remoteB = A1;
// Buttons
const int alarmPin = 8;
int alarmButtPressed = 0;
const int songChangePin = 7;
int songChangedPressed = 0;
// MP3 & Speaker
const int mp3TX = 10;
const int mp3RX = 11;
MP3Player mp3(mp3TX, mp3RX);
int currTrack;
const int numTracks = 6;
int vol = 30;
// LED Strip of Light (Neo)
const int ledPin = 2;
const int numLED = 7;
PololuLedStrip<ledPin> ledStrip;
rgb_color colors[numLED];
rgb_color magenta = rgb_color(255, 0, 255);
rgb_color black = rgb_color(0, 0, 0);

/*
  SETTING UP THE VARIABLES AND PINS
*/
void setup() {
  // Remote
  pinMode(remoteA, INPUT);
  pinMode(remoteB, INPUT);
  // Buttons
  pinMode(alarmPin, INPUT);
  pinMode(songChangePin, INPUT);
  // MP3
  mp3.initialize();
  currTrack = random(1, numTracks + 1);
  // Misc
```

```
  Serial.begin(9600);
}


/*
  LOOPING
*/
void loop() {
  // Alarm every three hours
  if ((millis() - timer) > threeHours) {
    // Time
    alarm = true;
    timer = millis();
    // Turn on Light
    for(int i = 0; i < numLED; i++) {
      colors[i] = magenta;
    }
    ledStrip.write(colors, numLED);
    // Starts playing the current song on the list
    mp3.playTrackNumber(currTrack, vol);
  }
  // Alarm turns on
  if (alarm) {
    // Turn off alarm
    // Button press check if previously not pressed and is currently being pressed
    bool buttonA = false;
    if(analogRead(remoteA) > 1000) {
      buttonA = true;
    }
    alarmButtPressed = (digitalRead(alarmPin) || buttonA);
    if (alarmButtPressed == 1) {
      // Turn off lights
      for(int i = 0; i < numLED; i++) {
        colors[i] = black;
      }
      ledStrip.write(colors, numLED);
      // Turn off and reset song/sound
      mp3.pauseIt();
```

```
      alarmButtPressed = false;
      alarm = false;
    }
    // Change song
    // Button press check if previously not pressed and is currently being pressed
    bool buttonB = false;
    if(analogRead(remoteB) > 1000) {
      buttonB = true;
    }
    songChangedPressed = (digitalRead(songChangePin) || buttonB);
    if (songChangedPressed) {
      int lastTrack = currTrack;
      while (lastTrack == currTrack) {
        currTrack = random(1, numTracks + 1);
      }
      mp3.pauseIt();
      delay(100);
      mp3.playTrackNumber(currTrack, vol);
      songChangedPressed = false;
    }
  }
}
```

60-223 Intro to Physical Computing