Log Book: Bobby Stephens

Newton College and Career Academy, NCSS

Table of Contents

Project Development

Sep 28, 2023

- Completed table of items needed to fabricate design
- Developed python program to create QR codes
- Developed python program to read QR codes using OpenCV
- #TODO: Create a program to read multiple QR codes in one frame

Oct 1, 2023

- Finished python program outlining and reading multiple QR codes using OpenCV
- #TODO: Migrate program to use ZBar Library

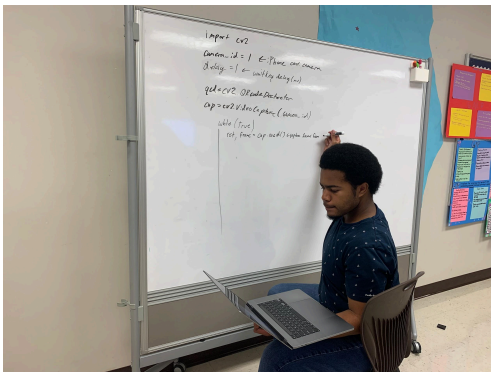
Oct 5, 2023

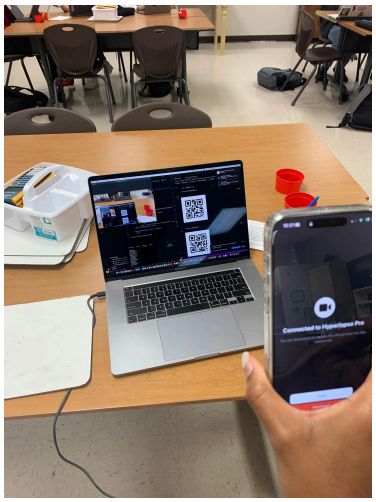
Today in class we finished up our research plan by adjusting different factors to make it more efficient. I also observed our QR system and explored possible solutions as to how the objects will associate with the locations. The QR codes were exported to a .docx format for printing.
the locations that we defined are :

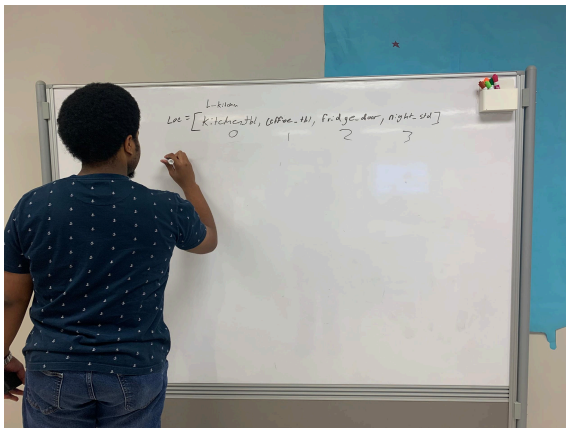
- kitchen table
- bedroom closet
- bedroom night stand
- bookshelf
- dining room table

the objects that we defined are:

- water bottle
- phone
- headphones
- laptop
- tablet







Oct 9, 2023 10:00 PM

From Home, I continued working on the Python Script with moderate success. Some improvements include the implementation of code used to store the location information of a QR code that has been detected and decoded from a video stream. If the location information is already in the loc_list, it is stored in the temp_location variable. If the location information is not in the loc_list, it is added to the obj_list and the temp_location variable (if it exists).

```
l_kitchenTable = []
l_bookShelf = []
l_bedRoomNightStand = []
l_bedRoomCloset = []
l_diningRoomTable = []

loc_list = ["l_kitchenTable",
            "l_bookShelf",
            "l_bedRoomNightStand",
            "l_bedRoomCloset",
            "l_diningRoomTable"]
obj_list = []
```

```
if s in loc_list:
                        temp_location = eval(s)
                        print("Temporary Location Stored: " + s)
                    elif (s not in loc_list):
                        if s not in obj_list:
                            obj_list.append(s)
                        try:
                            if s not in temp_location:
                                temp_location.append(s)
                                print("Location: " + str(temp_location) + " Item: " + s)
                        except NameError:
                            print("Location not stored")
```

Oct 17, 2023 9:30 AM

In today's class, the team was able to finalize the logic in the location-object recognition and storing system. The system is complete, and features error handling for long-term use. Additionally, code has been added to remove an object's name from its previous location when added to a new location. The next step is to begin working on the audio-feedback system and hardware integration.

Oct 19, 2023 9:30 AM

In today's class, I began setting up multithreaded processing for the QR reader. This will allow the program to process the location tracking system while being able to relay the acquired information to the text-to-speech system. Our next step as a team is to begin working on our physical device in parallel with the program

#end of first logbook check#

Oct 24, 2023 9:30 AM

Location: Mrs. Consuegra's Room
Goals: Begin working on Text-To-Speech System

Work Summary: In today's class, I began working on the implementation of the text-to-speech system for the program. I encountered many difficulties, including errors in the speech synthesizer and multithreading. For the next work session, I will continue to work on the text-to-speech system and continue working on the hardware.

Oct 26, 2023 9:30 AM

Location: Mrs. Consuegra's Room
Goals: Research and order required parts

Work Summary: In this shortened work session, I was able to order some of the hardware to allow the program to take advantage of a physical device. The parts ordered are as follows.

- **Raspberry Pi Zero FPC Camera Cable - 30cm long (Converts the RPi camera connector to a RPi Zero camera connector.**
- **Break-away 0.1" 2x20-pin Strip Dual Male Header (allows for GPIO Cables to be used in prototyping)**
- **Raspberry Pi Camera Module 3 NoIR - 12MP 75 Degree Infrared Lens (Camera used with the Raspberry Pi Zero for the program. NoIR Means that the camera does not include a built-in infrared blocker making night vision possible)**

Oct 30, 2023 12:00 AM

Location: Bobby's Bedroom
Goals: Continue making progress on Text-to-speech system, research and source parts

Work Summary: In my work session at home, I finalized the program with a standard and fallback version of the text-to-speech system. When connected to the internet, the program uses Google's Text-To-Speech API. When offline, the program uses pyttsx3 and eSpeak for vocal synthesis. I tested the program on an x86 version of Raspberry Pi OS. I also ordered the Raspberry Pi Zero 2W. Our goals for tomorrow are to finalize our chassis for our prototype.

```
81  def speak(txt, lang='en'):
82      try:
83          gTTS(text=txt, lang=lang).write_to_fp(voice := NamedTemporaryFile())
84          playsound(voice.name)
85          voice.close()
86      except Exception as e:
87          print(e)
88          engine = tts.init(driverName="espeak")
89          engine.say(txt)
90          engine.runAndWait()
91          engine.stop()
```

Oct 31, 2023 9:30 AM

Location: Mrs. Consuegra's Room

Goals: Finalize chassis for Raspberry Pi

Actions taken to complete goals: Explored CAD Sites for models of Raspberry Pi Cases, Found a pair of 3D printable frames to test the prototype on.

Work summary: In today's class, the team struggled to find CAD models of Raspberry Pi Cases to work on. We experienced many issues with the formatting of the CAD models found online, which featured files that were incompatible with the software the team had access to. Someone in another team suggested using 3D slicing software to load and manipulate the 3D printing file, which we plan to pursue in the next work session.

Nov 1, 2023 1:00 AM

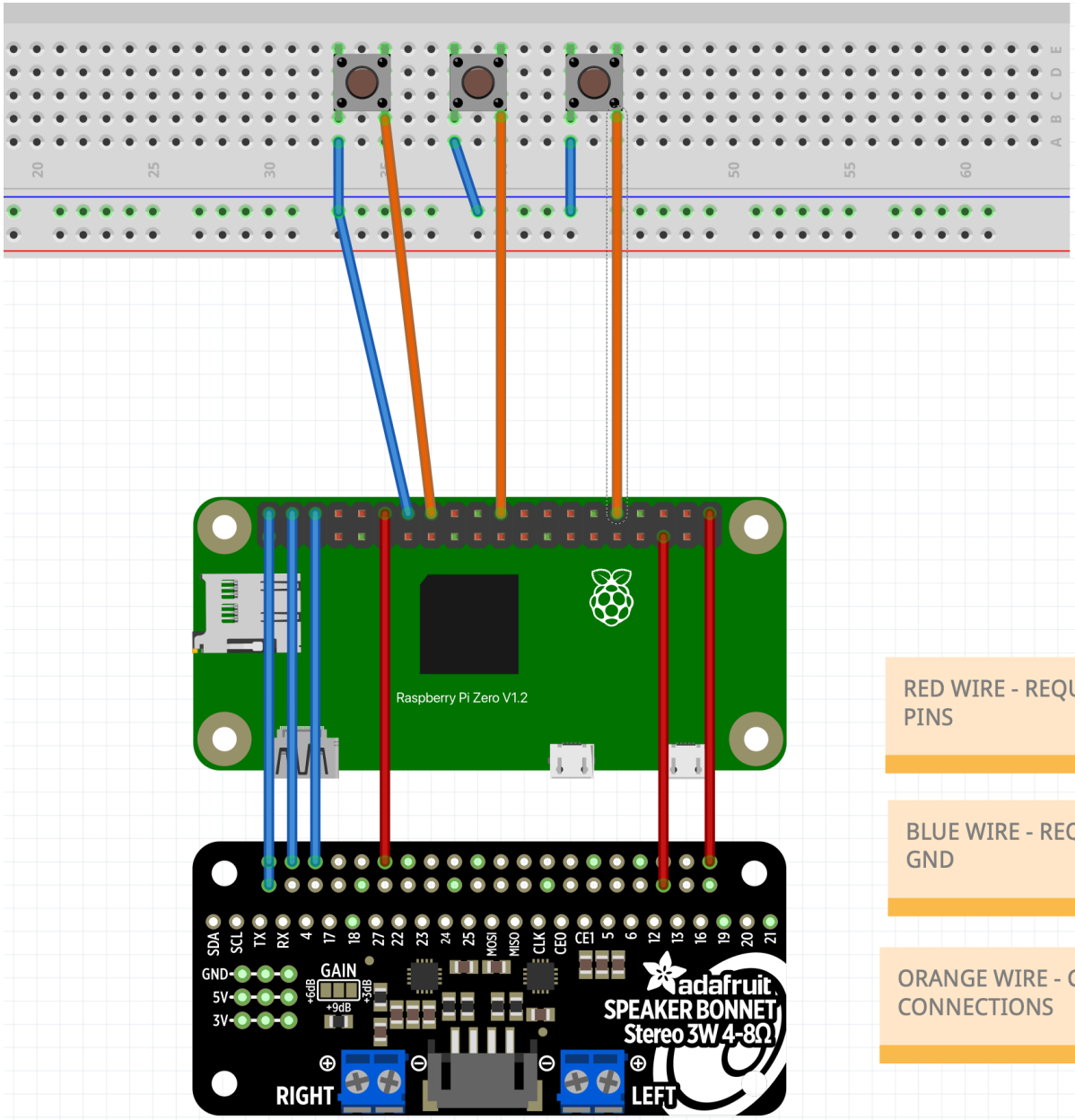
Location: Bobby's Bedroom
Goals: Continue Making progress on the Python program, Establish a wiring diagram for use in creating a physical prototype

Work summary: In my nightly work session from my home, I was able to add integration with Google Firebase into the program. This integration allows the location data to be uploaded to an online database in real time. This is convenient as it allows for a seamless app integration in the future. Below is the included function and its resulting entry in the real time database.

```python
def databaseSend():
    # Fetch the service account key JSON file contents
    cred = credentials.Certificate(
        'PythonQrReader/QR Implemetation/projectsmartglass-aee0e-firebase-adminsdk-vmdds-842cf32920.json')

    # Initialize the app with a service account, granting admin privileges
    firebase_admin.initialize_app(cred, {
        'databaseURL': 'https://projectsmartglass-aee0e-default-rtdb.firebaseio.com/'
    })

    # As an admin, the app has access to read and write all data, regradless of Security Rules
    ref = db.reference('/')
    while True:
        # for every object in the list, update the database wtih the object and the location
        if obj_list:
            for i in obj_list:
                # find the location of the object
                for location in loc_list:
                    if i in eval(location):
                        ref.update({
                            i[2:]: location[2:]
                        })

                        break
        else:
            print("HOLDING ON DATA")

```

The database is created in the JavaScript object notation (JSON) format. I can use this database in conjunction with JavaScript to create an app that relays this information from any location.

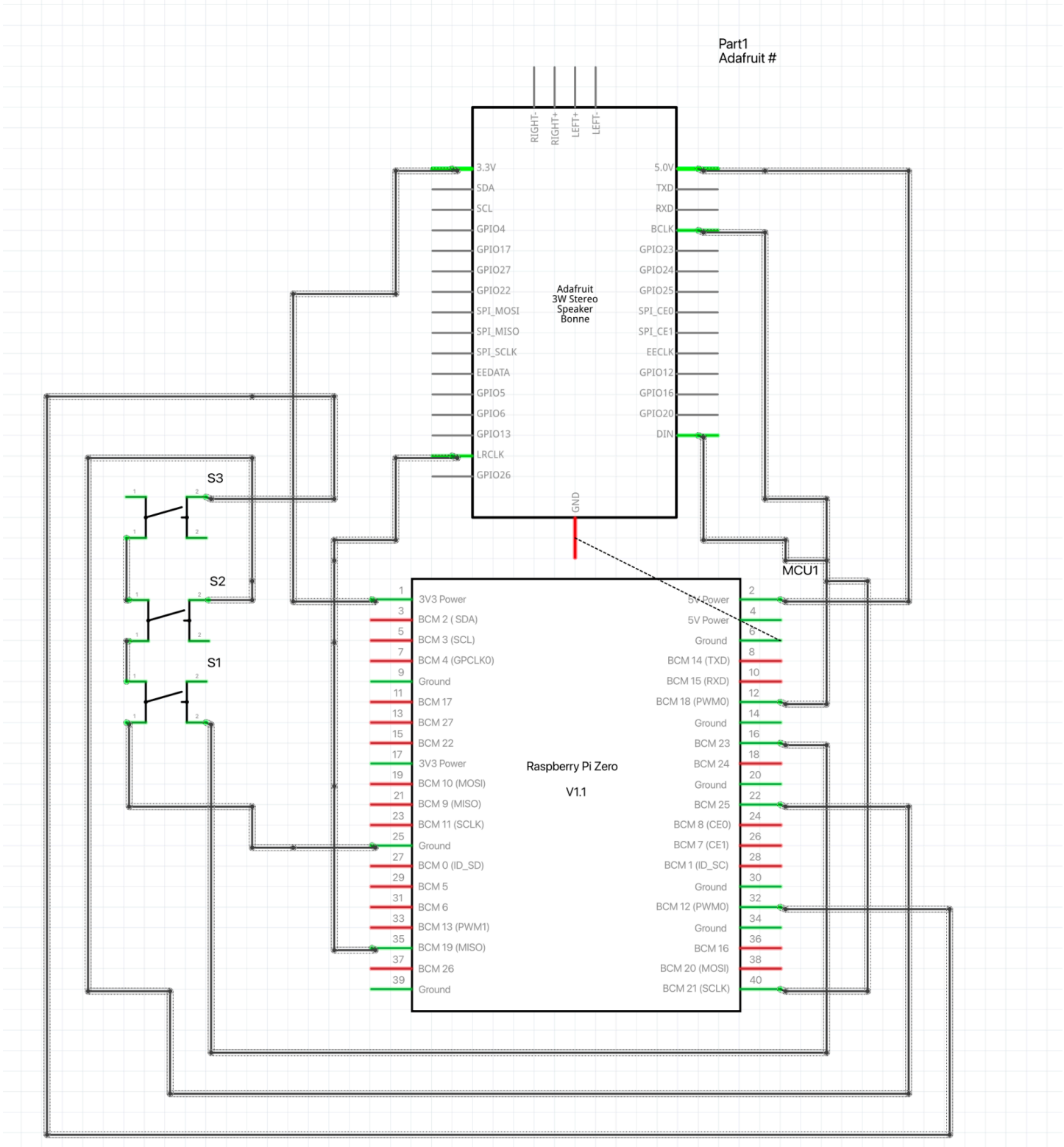
After I completed the Firebase integration, I began working on the wiring diagram for the V1 prototype. After many attempts to obtain the software illegally, I eventually decided to purchase Fritzing, a software made especially for connecting electronic devices. Below is the breadboard sketch and the device-to-device schematic.



RED WIRE - REQUIRED I2C PINS

BLUE WIRE - REQUIRED PWR + GND

ORANGE WIRE - GPIO CONNECTIONS



The sketch features the Raspberry Pi Zero, Adafruit 3 Watt Stereo Speaker Bonnet, and three buttons. Each external device will be soldered at the pins designated by the photos.

Nov 2, 2023 1:00 AM

Location: Bobby’s Bedroom
Goals: Setup Raspberry Pi Zero

Work summary: In my nightly work session, I was able to successfully install a headless version of Raspbian to the Raspberry Pi and install packages such as pyenv and python 3.11.6. During this time, I encountered an issue with my installation of OpenCV, the library used for QR code detection and decoding. After ~1 hour of troubleshooting, I decided to quit for the night and continue working on it during class.

Nov 2, 2023 9:30 AM

Location: Mrs. Consuegra’s Room
Goals: Complete Raspberry Pi Zero Setup, Continue working on CAD Model

Work Summary: During class, I had intended to continue the installation of the packages needed to run the program. I was not able to install the packages because the Raspberry Pi was not able to connect to the internet. This issue caused the team to switch our goals for today to continue working on the cad design for the glasses. After some consideration, it is best to postpone the CAD design to a later date.

Nov 3, 2023 9:00 AM

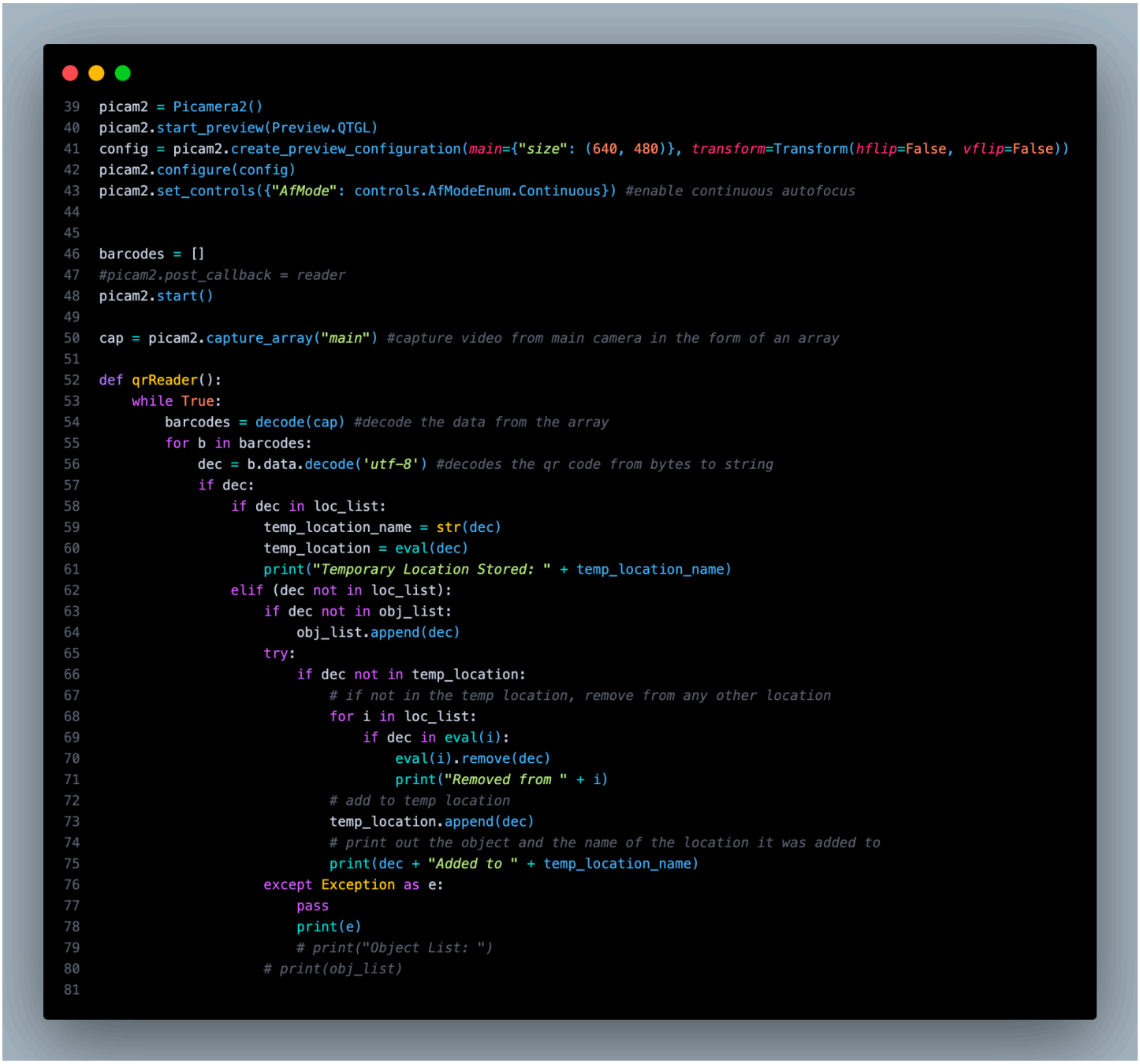
Location: VEX Robotics Room
Goals: Continue Raspberry Pi Setup

Work Summary: Throughout the day I continued the package installation process on the Raspberry Pi. After reading the error logs, it appeared that the Raspberry Pi was missing many dependencies that were required to the installation of OpenCV. By the end of the school day, All the necessary dependencies were installed, leaving only the installation of OpenCV, which will be installed tonight.

Nov 6, 2023 12:00 AM

Location: Bobby’s Bedroom
Goals: Program Raspberry Pi Zero

Work summary: To catch up on an entire weekend of troubleshooting, I was able to port the program over to the Raspberry Pi successfully. Over the weekend, I faced many challenges, including improper installations of Python environments and the discovery that the camera module purchased wasn’t compatible with the video decoder module. As a result, I spent the entire day yesterday searching for an alternative to OpenCV. In my preliminary research behind my program, I stumbled upon two libraries that could handle QR codes, one being OpenCV and the other being Zbar. Remembering this, I began rewriting the program to use the Raspberry Pi Camera with the Zbar library. Below is an updated version of the program similar to the original one written for MacOS 14.



```
picam2 = Picamera2()
picam2.start_preview(Preview.QTGL)
config = picam2.create_preview_configuration(main={"size": (640, 480)}, transform=Transform(hflip=False, vflip=False))
picam2.configure(config)
picam2.set_controls({"AfMode": controls.AfModeEnum.Continuous}) #enable continuous autofocus


barcodes = []
#picam2.post_callback = reader
picam2.start()

cap = picam2.capture_array("main") #capture video from main camera in the form of an array

def qrReader():
    while True:
        barcodes = decode(cap) #decode the data from the array
        for b in barcodes:
            dec = b.data.decode('utf-8') #decodes the qr code from bytes to string
            if dec:
                if dec in loc_list:
                    temp_location_name = str(dec)
                    temp_location = eval(dec)
                    print("Temporary Location Stored: " + temp_location_name)
                elif (dec not in loc_list):
                    if dec not in obj_list:
                        obj_list.append(dec)
                    try:
                        if dec not in temp_location:
                            # if not in the temp location, remove from any other location
                            for i in loc_list:
                                if dec in eval(i):
                                    eval(i).remove(dec)
                                    print("Removed from " + i)
                            # add to temp location
                            temp_location.append(dec)
                            # print out the object and the name of the location it was added to
                            print(dec + "Added to " + temp_location_name)
                    except Exception as e:
                        pass
                        print(e)
                        # print("Object List: ")
                    # print(obj_list)

```

Lastly, I purchased the audio hat + the flux needed for soldering.The team's next step in completing the project will be to source helping hands and begin soldering.

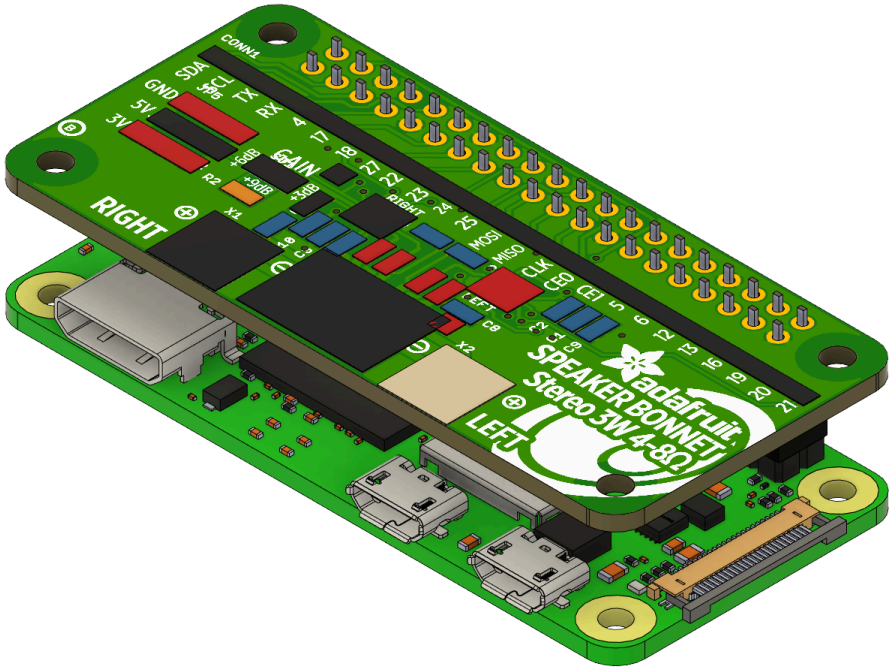
Nov 7, 2023 2:00 AM

Location: Bobby's Bedroom
Goals: Find models for the Raspberry Pi audio hat
Work log:

- ☑ Search the internet for additional CAD files relating to the project
- ☑ Port the CAD files into Autodesk Fusion 360 and EAGLECAD
- ☑ Attempt to create casing for the fully assembled prototype

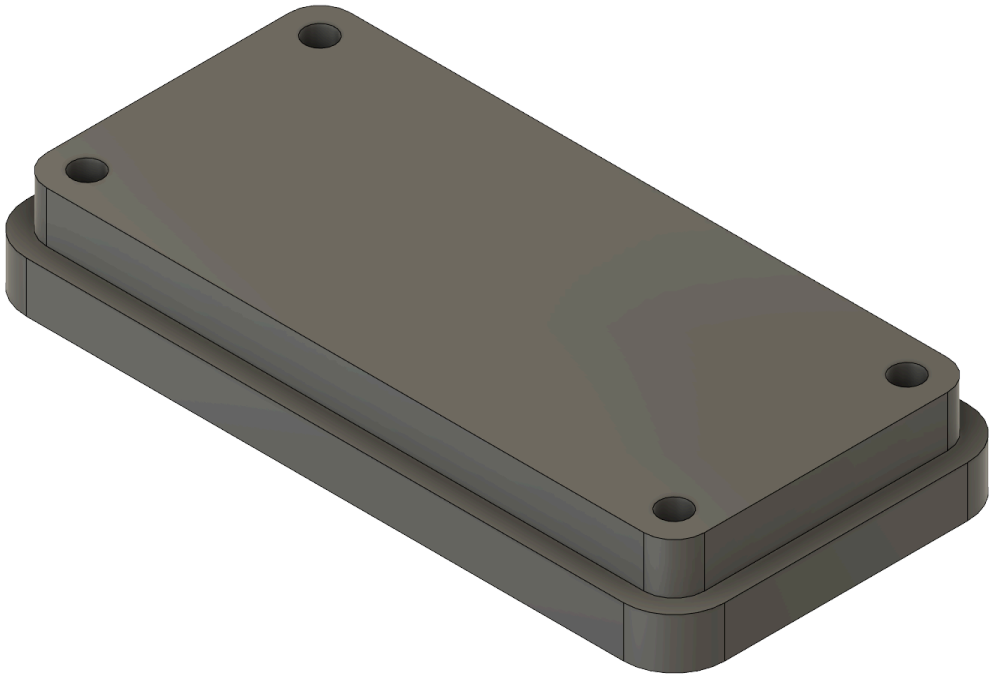
- [x] Export to slicer for print estimates

Work Summary: In this nightly work session, I wanted to make great leaps in progress since I will not be meeting with the team on 11/07/23 and 11/09/23 due to Election Day and a school field trip. In summary, this lengthy work session involved me making a strenuous effort to learn Fusion 360 in order to make the case for the prototype. First I created a rough assembly of what the unit will look like, as seen below:

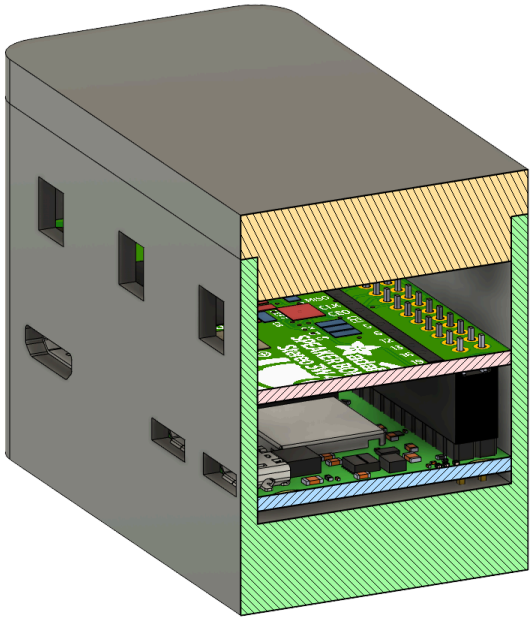


After this was completed, it was time to form a case around the prototype. I began by taking the measurements of the Raspberry PI board schematic and replicating them in a CAD drawing. From there, I continued by offsetting the perimeter of the rounded rectangle to build walls. After a couple of hours, The case was completed. It included ports for mini HDMI, power, USB data, and the camera connector. Additionally, there are small, evenly spaced squared above the ports reserved for the buttons used to control the program.

Lastly, I made a lid for the case. I did it by taking the original drawing of the Raspberry Pi with the 4mm offset by extruding the whole profile, including the offset, and extruding the inside by 1.5x.



Shown below is a cut model of full prototype, excluding the camera.



To conclude for the night and to further validate my work, I exported both of the case files (case body and top) as .step files and imported them into slicer software. Slicer software is an application that converts 3D CAD models to 3D printer machine code. Another feature of the software is print time estimates. Since the models are quite small, it shouldn't take more than three hours to fully print, as reported by the software.

Data From GrabCad:

Print Time	F123 ABS (in³)	F123 QSR support (in³)
2h 37m	3.178	0.845

Next Steps: The next steps the team needs to take to have a complete project by the deadline are to 3D print the part, retrieve the audio module, begin soldering, and begin working on the triboard.

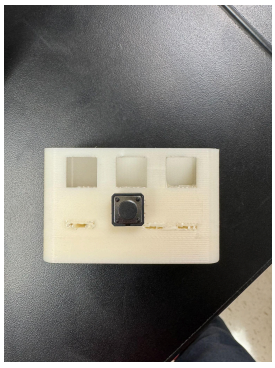
Nov 8, 2023 3:00 PM

Location: Allred's Engineering Design Classroom
Goals: 3D Print and inspect case

Work log:

- ☑ Begin the 3D print
- ☑ Wait 3 hrs
- ☑ Retrieve the plastic part and remove all support
- ☑ Conclude and plan adjustments

Work Summary:
In my day-long work session, I was able to begin the fabrication of the case that the raspberry pi and audio hat fit in. I began the print around 11 AM, and I returned to the design lab to retrieve the part at 3 PM. When I returned, I noticed that there was an error in the print with only 13 layers remaining. Later it was discovered that the error was caused by a piece of plastic filament lodged in the nozzle. To a certain extent, the print was still usable, so I decided to keep it. I had difficulty removing the support material from the print because I was unaware of its strength and that it requires submersion in a caustic water solution. Removing what I can, I was able to check the fitment of the electronics within the case. The dimensions of the pi + hat fit inside, and the top fits perfectly. The only need for adjustment in the case is found in the size of the buttons The opening for the button is too small to snugly fit the button. In the future, I plan to fix this by creating a larger opening for the button and printing on a printer with easy to remove supports.



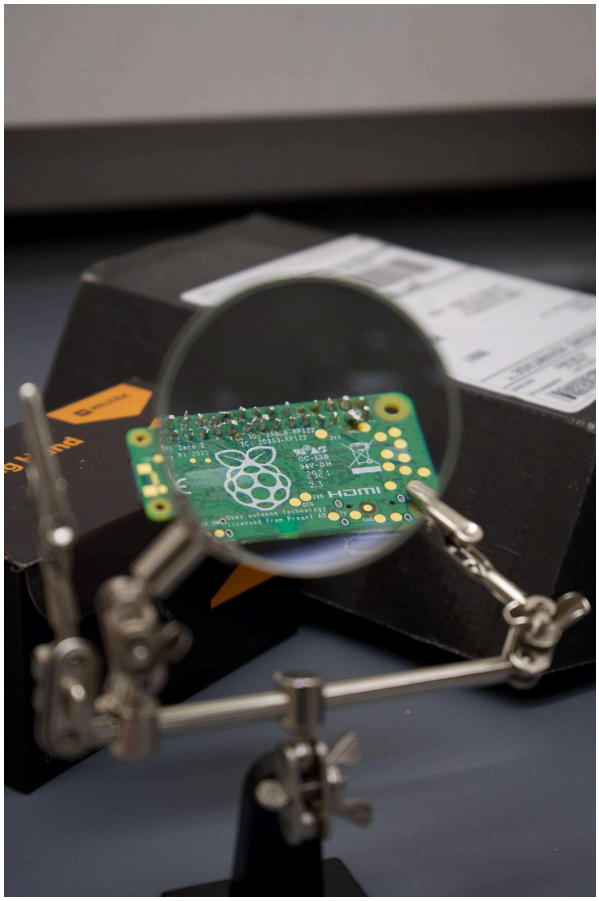
Nov 13, 2023 12:00 PM

Location: Consuegra's Class + Allred's Engineering Lab
Goals: begin soldering 2x20 pins to Raspberry Pi, Begin reprinting part

Work log:

- [x] ~~Export Updated case from Fusion 360 to Snapmaker Luban Or Markforged Slicer~~
- [x] ~~Begin printing the case~~
- [x] ~~Prepare Solder iron in a safe environment~~
- [x] ~~Begin soldering~~
- [x] ~~Clean up~~

Work Summary:
In today's work session, I was able to update the case design, modifying the sides of each button slot by +0.100 millimeters. After I modified the dimensions, I exported the files and went to the engineering lab. Once I entered the engineering lab, I was told that the Printer that had jammed previously was fully operational. Being encouraged to use it, I sent my files to the printer to be fabricated again. For this print, I will make sure to place the model into the caustic water solution to remove all support structure and to have a clean print.
I returned to Consuegra's classroom to begin soldering. The reason I chose the biotechnology lab as a place for an engineering job is because of the installation of a fume hood in most science classes. The fume hood allows for me to solder without inhaling the toxic fumes associated with it. Soldering was a tricky process that resulted in the damage of a few pins, but they still should be usable for the sake of this project. Below is a snippet of the soldering setup.



The next steps that I need to take to continue working on this project in an efficient manner are to Collaborate with my other team members on the display poster and continue working on the Raspberry Pi (NEEDS TO BE DONE BY THE END OF THE WEEK)

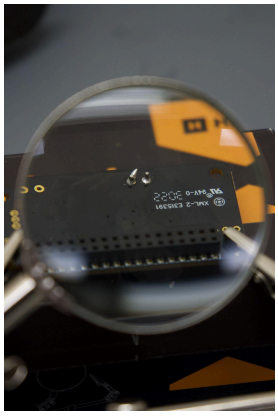
Nov 14, 2023 9:00 AM

Location: Consuegra's Biotechnology Lab
Goals: Update team on Engineering Work, Finish soldering to test the audio

Work Log:

- ☑ Update team on engineering work
- ☑ Begin soldering audio board
- ☑ Begin testing
- ☑ Pray I don't short anything

Work Summary: In today's class, I was able to update the team on all the work that has been accomplished on the project. Following, I began soldering the audio terminals to the raspberry pi hat and connecting the wires from the bone conductor transducer to the newly soldered terminals. This attempt to connect was unsuccessful, as the wires connected to the bone conductor were too small. To resolve this problem, I removed the terminals from the board by reheating the solder, and letting the pins that connected the terminals to the board slip out. Then, I soldered the wires from the bone conductor transducer to the audio board, completing the connection



After connecting the bone conductor transducer to the audio board, it was time to begin setting the device up with the Raspberry Pi. I installed the shell script from Adafruit and restarted the Raspberry Pi Zero twice. After doing so, I began testing the audio using pink noise. On my first attempt, no audio was being played. My initial reaction was that my pins were not soldered correctly, so I took a multimeter and began investigating the power being sent to the board. I took the power(red) probe and touched it against a vacant 5v pcb hole, and the ground(black) probe against a vacant GND pcb hole. The multimeter read slightly over 5v, which means that power is correctly being run to the board. After ten minutes of troubleshooting, The bone conductor began to play white noise.

Nov 16, 2023 9:30 AM

Location: Consuegra's Classroom
Goals: Assist in the poster documentation, begin soldering buttons

Work Log:

- ☑ Wake up
- ☑ Begin working on poster document
- ☑ Complete poster document

- ☑ Check case dimensions
- ☒ Begin soldering buttons

Work Summary: In today's class, I was first tasked with including the engineering background behind the project. This description of the project is to be displayed on the team's poster board during our debut presentation at the Newton County Science Fair on Dec 16, 2023. It is written as follows:

"To solve this problem, The team has taken a novel approach to improve the lives of people who suffer from memory loss by designing and fabricating an assistive device to help individuals recall key household locations. This is done through Computer Vision and QR codes. When the device is worn, a camera actively scans the environment, seeking QR codes. The QR codes contain unique identifiers that specify whether it's connected to an object or a location. In an environment where a location and object code are detected, they become associated with each other and can be later recalled using tactile buttons on the unit. In creating the device, the team has taken advantage of low-cost computing and additive manufacturing to address the global need for affordable assistive technology. The price to fully manufacture the device is around 78 United States Dollars."

After completing the engineering background, I returned to the best part of the project. Today, I was checking case dimensions to see if they would work well. After taking precision measurements using calipers and physically testing the fitment of various parts, I determined that the diameter of the guides were too large in diameter to accommodate the holes in the Raspberry Pi. This will later be reflected in the fusion model. During this session, I was not able to solder any buttons to either the raspberry pi or the audio Pi Hat.

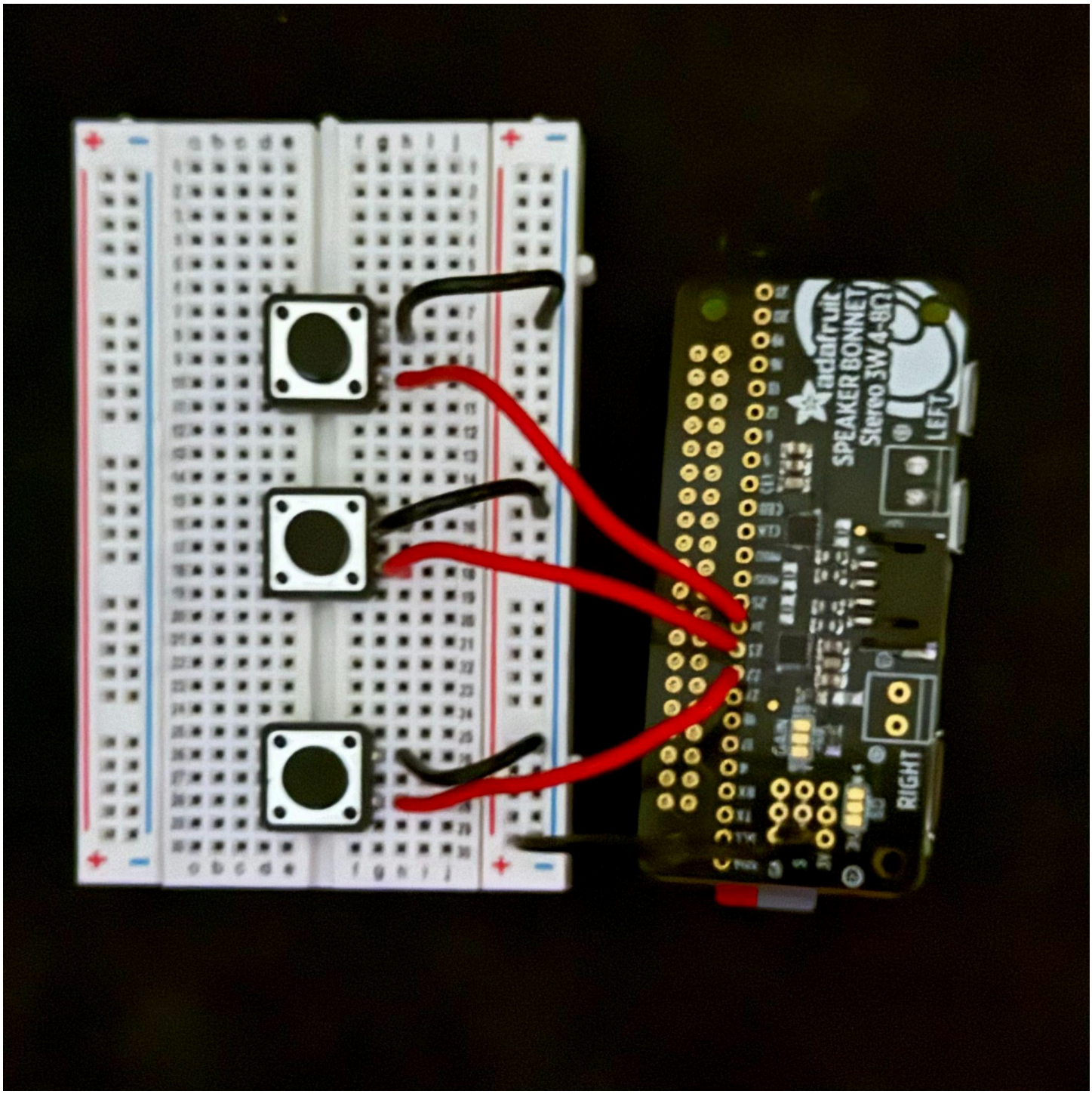
Nov 22, 2023 12:00 AM

Location: Bobby's Bedroom
Goals: Make Progress, Begin using buttons in program

Work Log:

- ☑ Try not to blow up anything
- ☑ Begin prototyping with breadboard
- ☑ Create code to test design
- ☑ Integrate buttons into qrReader.py
- ☑ Blow something up anyway
- ☑ Share new way of modifying code on the RPi using visual studio code

Work Summary:
In another restless work session from home, I began working on the button navigation system for the program. Before summarizing tonight's work, I believe that it is important to describe how the Raspberry Pi receives signals from the buttons. The Raspberry Pi sends a constant ground signal to one pole of the button, while the other pole holds no signal, but is connected to an available GPIO port. When the program is set to receive a power signal (digital input) from the button and the button is pressed, the circuit is complete and a boolean value will be returned. This can be used as a way to create an input to the program using physical buttons. For visual reference, here is a display of the system using a breadboard.



After doing some light research, I discovered that two python libraries could help me accomplish this task. The first
Library was RPi.GPIO, a library developed to support GPIO Devices with the Raspberry Pi family of devices in mind. The problem that lies with this library is that it fosters older programming methods from the likes of arduino, which is based on the C++ programming language. It lacks functionality and doesn't exactly follow the efficiency of Python. As a superior alternative, gpiozero is an expansive library that takes advantage of many of Python's most efficient and convenient features. It also features callback functions, which may be useful in the future. To test out the functionality of the gpiozero library, I wrote the following program below:

```python
from time import sleep
import gpiozero
from gpiozero import Button

buttonA = Button(22)
buttonB = Button(23)
buttonC = Button(24)

def buttonAPressed():
    print('Button A Pressed')
    sleep(0.2)
def buttonBPressed():
    print('Button B Pressed')
    sleep(0.2)
def buttonCPressed():
    print('Button C Pressed')
    sleep(0.2)

while True:
    buttonA.when_pressed = buttonAPressed
    buttonB.when_pressed = buttonBPressed
    buttonC.when_pressed = buttonCPressed

```

Setting up the program to read the buttons was simple. I first created 3 instances of buttons through GPIO, uniquely titled buttonA, buttonB, and buttonC. Each instance is set equal to the GPIO Port that senses the change in power. In this case it will be ports 22, 23, and 24. Moving forward, I created functions that will print a string to the terminal once they have been called. In the while loop at the end, the callback functions have been established, creating an event-driven program. This makes it so that if any button is pressed, its corresponding function will be called and executed.

After my tests with the library and the program were successful, I began to integrate it with the main qrReader.py file and the selection function/thread. The successful integration with the file only required a small bit of tweaking. The comparison between the original version made for the Mac and the updated version version are below:

Mac:

```
def selection():
    selected_index = 0

    while True:
        # get the selected string
        if obj_list:
            selected_string = obj_list[selected_index]
            print(f"Selected: {selected_string[2:]}")

            # get the user input
            user_input = input(
                "Enter 'a' to move left, 'd' to move right, or 'enter' to select: ")

            # if the user enters 'a', move the selection to the left
            if user_input == 'a':
                selected_index = (selected_index - 1) % len(obj_list)

            # if the user enters 'd', move the selection to the right
            elif user_input == 'd':
                selected_index = (selected_index + 1) % len(obj_list)

            # if the user presses enter, find the location of the selected item
            elif user_input == '':
                # find the list the selected string is in
                for location in loc_list:
                    if selected_string in eval(location):
                        print(f"{selected_string[2:]} is in {location[2:]}")
                        speak(f"{selected_string[2:]} is in {location[2:]}")
                        break
```

Raspberry Pi:

```python
def selection():
    selected_index = 0

    while True:
        # check to see if obj_list is available
        if obj_list:


            # sleep
            sleep(.15)

            # if the user presses buttonA, move the selection to the left
            if buttonA.value == 1:
                selected_index = (selected_index - 1) % len(obj_list)
                selected_string = obj_list[selected_index]
                print(f"Selected: {selected_string[2:]}")
                speak(f"Selected: {selected_string[2:]}")


            # if the user presses buttonC, move the selection to the right
            if buttonC.value == 1:
                selected_index = (selected_index + 1) % len(obj_list)
                selected_string = obj_list[selected_index]
                print(f"Selected: {selected_string[2:]}")
                speak(f"Selected: {selected_string[2:]}")

            # if the user presses buttonB, find the location of the selected item
            if buttonB.value == 1:
                # find the list the selected string is in
                for location in loc_list:
                    if selected_string in eval(location):
                        print(f"{selected_string[2:]} is at the {location[2:]}")
                        speak(f"{selected_string[2:]} is at the {location[2:]}")
                        break

```

The next steps would be to complete the physical build.

Nov 28, 2023 9:30 AM

Location: Consuegra's Classroom
Goals: Touch Base with group, Order improved wire

Work Log:

- [x] Bring Materials back to school
- [x] Review feedback from Poster Check 1
- [x] Order new wires
- [x] Commit last week's changes to GitHub

Work Summary:
In today's work session, I was a little relaxed with my work order. Following the fire drill, I began to reflect on the feedback, received from the first poster check. Moving forward, I will be sure to add definitions to terms that many people outside of engineering may not understand. After my reflection, I began to refocus on my project, asking myself what is done and what needs to be done. When looking at all of the parts in my bin, I remembered that I needed to purchase new wires to connect the speaker. After doing some research online, I decided that moving to the 22 AWG(American Wire Gauge) would be the best replacement to my 18 AWG wire. The 22 gauge wire has a better balance of strength and flexibility when compared to the 18 AWG. In usage, the 18 gauge wire was extremely hard to bend, and sometimes put stress on the solder joints. In multiple cases, combined with the weakened glue strength on the solder pads, the connection to the speaker would be permanently broken. I believe that 22 gauge wire, in addition to an improved connection strategy(low heat solder??) will solve this problem.

Towards the end of the period, I realized that in the event of catastrophic failure, my python program would not be saved. To protect my data, I connected the raspberry pi to my computer, created an SSH connection (remote connection) from my computer to the raspberry pi using VS Code, and pushed the changes to my GitHub repository. The changelog is linked below, officially merging my changes from November 22 with the November 9th edition.

Changes

My next step in completing the build is to assemble all the pieces inside the case and establish a method for mounting the unit to the glasses.

Nov 30, 2023 9:30 AM

Location: Consuegra's Classroom
Goals: await necessary parts, continue work on visual poster, reveal case v3 to the team

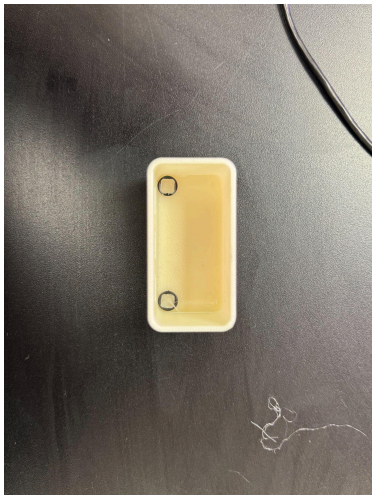
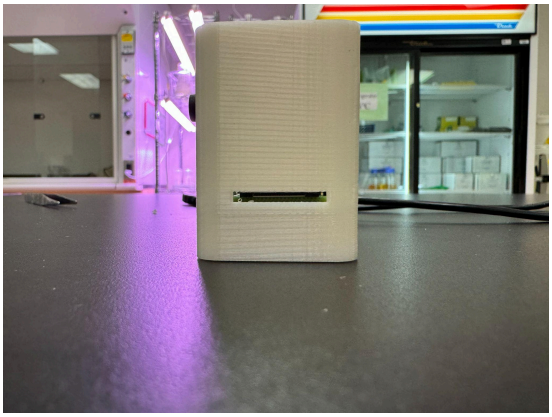
Work Log:

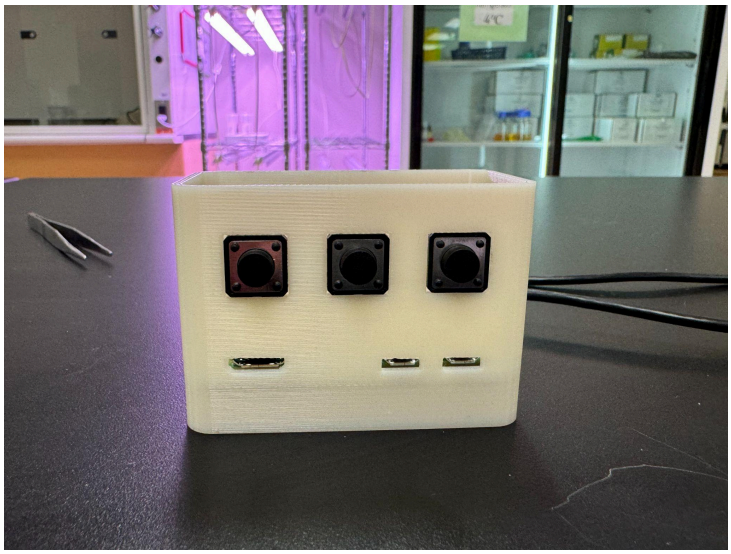
- [ ] Eat biscuit
- [ ] Drink unhealthy energy drink
- [ ] Add improved engineering background
- [ ] Take an updated screenshot of Fritzing Prototype

Work Summary:

After receiving word that the team's science fair interview would be moved to better accommodate my schedule, I began to update and refine the engineering background for the project. I made sure to include many engineering-centric terms that were key components of the project and a relatively thorough explanation of the concept. The concepts listed inside of the engineering background include GPIO(general purpose input/output), I^2C(Inter-Integrated Circuit), QR(quick response) codes, and Computer Vision.

Following, I began to work on the v3 case for the device. This version of the case features a couple of improvements from the previous version, featuring larger side ports for the SD card and camera flex cable, as well as shrinking the diameter of the guides, originally referenced November 16th. When installing the raspberry pi into the case, the guides posed an issue once again. This time, the issue lied with the guides' placement within the floor of the case. Frustrated with the issue, I used a dremel to grind away the guides, and hot glued 2 eighth-inch washers to the bottom of the case. After the glue cooled, I placed the Raspberry Pi into the case, and to my surprise, the fitment was perfect. Additionally, I pressed the buttons into their slots, as well as inserted the camera flex cable and SD card through their respective ports on the case.




Here, in this stage, the build is very close to the version 1 completion goals. All that needs to be done is permanently fitting the buttons, bone conductor placement, and wiring.

Dec 5, 2023 ALL DAY

Location: Consuegra's classroom
Goals: FINISH THE PROJECT

Work Log:

- ☑ Wake up mad
- ☑ Eat Breakfast
- ☑ Skip Class
- ☑ Break Things
- ☑ Print camera case
- ☑ ITS DONE

Work Summary:

Throughout the day, I was determined to finish the build of the wearable. Earlier that morning, I began to print the casing for the camera module. While It was printing, I began assembling the device for long-term usage. Starting with the Audio board, I had to solder two more black wires to the GND port and solder a brand new bone conductor. During the process, I broke a solder pad on the right channel and had to use the left as a backup. The Audio Hat is now prepared for installation. The next part that needs to be prepared is the Raspberry Pi. When installing the camera connector to the board, the plastic piece that locks the flex cable into the port broke off and flew across my workspace. At first, I thought the damages to the board were catastrophic, and that the camera connector was useless. But to my surprise, Just reinserting the plastic piece alongside the flex cable allowed for the port to continue working with its intended function. After the flex cable was successfully installed, I placed the audio board on top of the Raspberry Pi and placed it in the case. This process was strategic, and starts by threading the flex cable through its appropriate slot on the case. Once threaded, it is safe to place the boards inside the case. After successful placement, I began installing the buttons. The buttons fit nicely into the square holes, and are secured using hot glue. Once the glue had set, I cut and adjusted the wires to reach each button, and connected the buttons to the audio board using low-temp solder.

Now that the internals are complete, I retrieved the newly-printed camera case from the engineering lab and tested the fitment with the Raspberry Pi Camera. The camera module did not fit, So I had to employ a dremel to grind the pegs off. With the pegs removed, The Raspberry Pi camera fit… for the most part. With the camera in the case, I was able to adhere it to the larger case. The only thing that remains in the assembly is routing the camera connector to the camera in the best way possible. Once complete, I tested the system with everything in place, and everything worked successfully.

V1 COMPLETE!!

Jan 3, 2024 3:00 AM

Location: Bobby’s Bedroom
Goals: Do some preliminary research on facial recognition systems on the Raspberry Pi

Work log:

- ☑ Procrastinate the whole break away
- ☑ Power the wearable computer and base
- ☑ Research Github
- ☑ Discover possible solution #1

Work Summary: Tonight I did some research on how I can configure facial recognition using arrays rather than video capture on OpenCV. Solution: use the facial recognition library from https://github.com/ageitgey/face_recognition

Jan 11, 2024 9:30 AM

Location: Consuegra’s Classroom
Goals: Record Data

Work log:

- ☑ Gather Distance measuring tools
- ☑ Gather QR Codes for the device to read
- ☑ Prepare Excel Spreadsheet
- ☑ Complete Testing Table w/ formulas

Work summary: During today’s work session, and as a response to the feedback from the local science fair, the team has decided to gather data on the accuracy on the SmartGlass device. The test was conducted by placing the QR codes at a fixed location and measuring the distance from the device to the QR code. The testing procedure was conducted in the classroom, with overhead lighting similar to the lighting in many homes worldwide. Once the testing environment was set up, we recorded the data. With each distance tested, we started the QR tracking program on the device and recorded if the device could read the QR code. This process was repeated ten times for each distance. The data was then recorded in an Excel spreadsheet, and percentages were calculated to determine the device’s accuracy at each distance. The team will later use and reorganize the data to place on the team's poster board for the regional science fair.

Jan 16, 2024 10:00 AM

Location: Consuegra’s Classroom
Goals: Facial Recognition Progress, Organize Data

Today, the team has worked on preparing the data for the poster and made progress on facial recognition.

Poster:
After completing the testing of the SmartGlass device, the team has verified the data and organized it into a table fitting the overall theme of the poster.

Facial Recognition:
After a week of research and testing, the team has encountered a fatal roadblock in implementing facial recognition. During today’s work session, the facial recognition program was tested on the device, and the team discovered that the device wasn’t powerful enough to recognize faces promptly. The team has decided to step back and continue preparing for the regional science fair.

Next Steps:
Going forward, I will consider and test alternative methods of processing facial recognition, such as using another Raspberry Pi or a cloud-based service.