Lab Notebook - Calvin Lee (calvinl4)

Date: 8/30

Objective: Brainstorm ideas for what we could do for our project

Several ideas were presented, and we all posted our ideas onto the website where necessary. My idea that I wanted to push forward with is a computer vision based lock system that unlocks a door based on facial recognition. Another idea I had was to track a basketball shot and measure certain statistics about it to help the shooter have a better way to analyze their shot.

Date: 9/5

Objective: Finalize a plan for which idea we want to go through with

Of all of our project proposals, two got approved. We had one idea that was to correct the form of gym goers, and another one to measure the availability at the arc. The one we decided to go with is the availability idea, and to use a motion sensor to recognize whether or not someone is actually at the arc, and to transmit this data through a raspberry pi to a webserver.

Date: 9/7

Objective: Make changes to our idea to reflect what Professor Arne said.

He mentioned how it wouldn’t make much sense to have a whole sensor system just to measure availability as it would be too complex for no reason. We ended up deciding on using a button for the initial signal, and to use a PIR sensor as a failsafe just in case the user forgets to push the button.

Date: 9/12

Objective: Start writing project proposal

We spent most of this time writing up our project proposal and finalizing some ideas about how we wanted to carry out our project. This is where we decided on our subsystems:

- machine sensor pcb to IoT device
- IoT device to AWS Server
- AWS Server to website
- load sensor to alarm

We also decided on using specific parts.

Gikfun button:

https://www.amazon.com/Gikfun-Waterproof-Button-Momentary-Arduino/dp/B07F8GBWGG/ref=asc_df_B07F8GBWGG/?tag=hyprod-20&linkCode=df0&hvadid=343224601369&hvpos=&hvnetw=g&hvrand=2325432847971158710&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9022196&hvtargid=pla-757057055269&psc=1&tag=&ref=&adgrpid=70737352522&hvpone=&hvptwo=&hvadid=343224601369&hvpos=&hvnetw=g&hvrand=2325432847971158710&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9022196&hvtargid=pla-757057055269

Alarm to alert the user if they dont press the button:

https://www.digikey.com/en/products/detail/tdk-corporation/PS1440P02BT/2236832?utm_adgroup=Alarms%2C%20Buzzers%2C%20and%20Sirens&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Product_Audio%20Products_NEW&utm_term=&utm_content=Alarms%2C%20Buzzers%2C%20and%20Sirens&gclid=Cj0KCQjw39uYBhCLARIsAD_SzMRocrg56djQZdtSr1banc2WuquRuRWNwZ3Xb1x-w5BqNdJqJw-9-HQaAjB5EALw_wcB

A network card to communicate with the raspberry pi:

https://www.espressif.com/en/products/socs

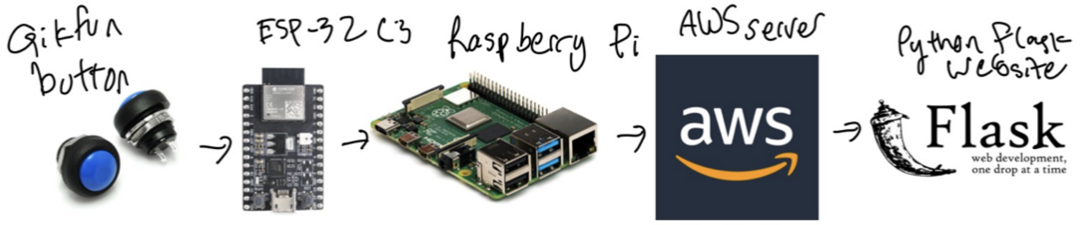
AWS to manage databases and cloud connections:

https://docs.aws.amazon.com/iot/latest/developerguide/connecting-to-existing-device.html#gs-device-view-msg

Date: 9/14

Objective: Finalize Project Proposal

Finalized proposal and added diagrams and pictures to support the presentation.



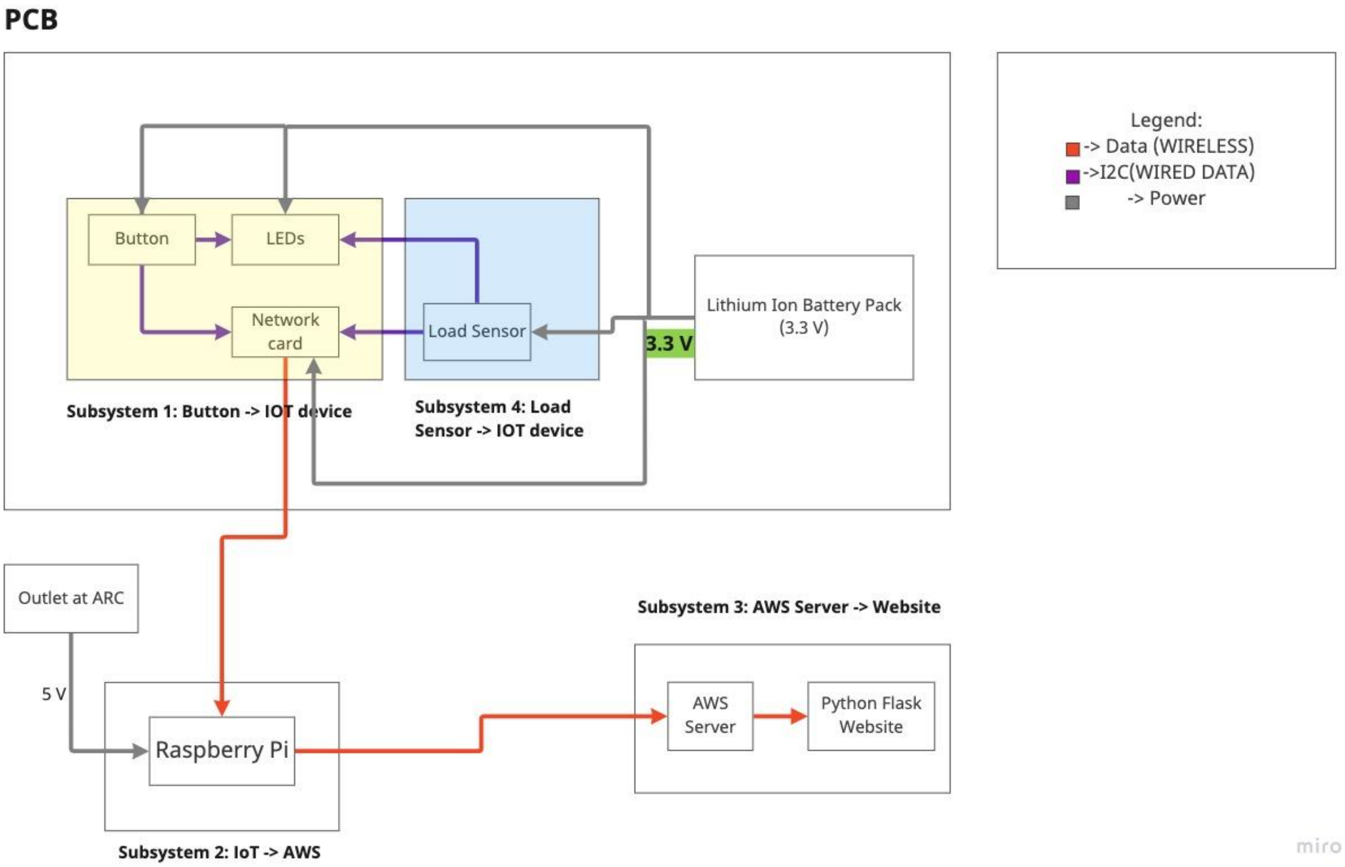
Decided on these high level requirements:

Battery Life: PCB should be able to last for multiple hours without a direct power source.

Multiple Machines Serviced: Our system will be able to support up to 5 machines being used.

Website Access: Multiple people will be able to access the website at one time

Designed our first draft block diagram with connections and subsystems labeled:



We decided on removing the alarm from the equation as we realized that it is not that practical of a feature.

References:

MQTT document:

https://diyi0t.com/microcontroller-to-raspberry-pi-wifi-mqtt-communication/

External battery for PCB:

https://www.google.com/shopping/product/5898283066540390660?q=external%2Bcircular%2Bbattery%2Bfor%2Bpcb&client=safari&rls=en&sxsrf=ALiCzsZaA5BHehX59EQpajrvZOfjehFTwg%3A1663275125169&biw=1440&bih=735&dpr=2&prds=eto%3A2725741664819496371_0%2Clocal%3A1%2Cpid%3A1185675488743134041%2Cprmr%3A2%2Crsk%3APC_10260653902913479911&sa=X&ved=0ahUKEwiX2P6R15f6AhXpjIkEHVlACSgQ8wIIuBE

Battery holders:

https://www.te.com/usa-en/product-2-1775485-1.html?te_bu=Cor&te_type=srch&te_campaign=ggl_usa_cor-ggl-usa-srch-smbmktg-fy22-googlefeed_sma_sma-2210_2&elqCampaignId=115724&mkwid=VDybDFVR%7Cpcrid%7C386964346943%7Cpkw%7C%7Cpmt%7C%7Cpdv%7Cc%7Cslid%7C%7Cproductid%7C2-1775485-1%7Cpgrid%7C78782457763%7Cptaid%7Cpla-298884436745%7C&utm_content=VDybDFVR%7Cpcrid%7C386964346943%7Cpkw%7C%7Cpmt%7C%7Cpdv%7Cc%7Cslid%7C%7Cproductid%7C2-1775485-1%7Cpgrid%7C78782457763%7Cptaid%7Cpla-298884436745&gclid=Cj0KCQjwmouZBhDSARIsALYcourXPqfM1fucoo39DQsdyYPBcSBieWtgXxXjzO0ox1Y1Peppnti2pE0aAkw-EALw_wcB

Date: 9/19

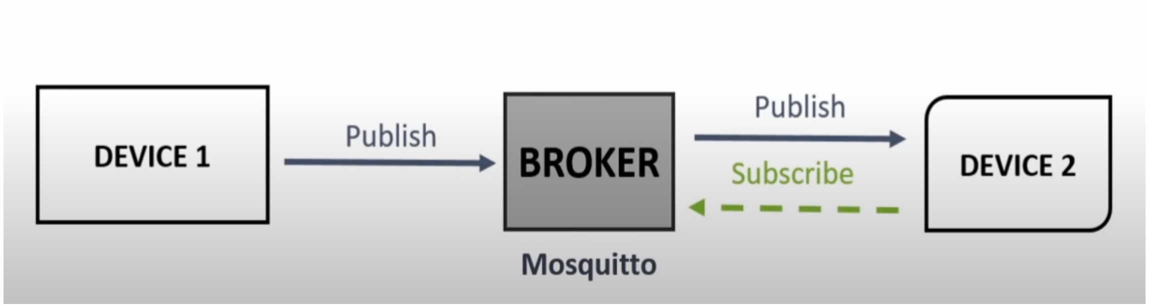
Objective: Begin first draft of Design document

Started writing the design document. One of the main differences from the project proposals is its inclusion of the requirements and verifications tables, or R/V tables for short. Here is an example:

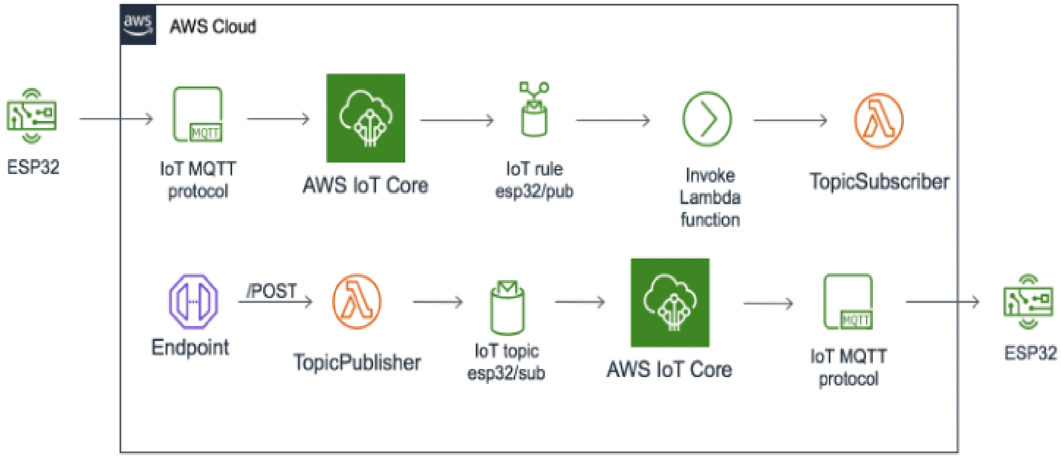
Button Requirements	Verification
Requirement 1: A button press should be able to send a MQTT packet to our IoT device.	Verification 1: On our IoT device, which will act as a MQTT server, we can verify what devices are requesting information on it, by a simple command on the IoT device.
Requirement 2: When a user presses the button to use the ARC machine, the red LED should light up to signify the machine is in use.	Verification 2: This can be easily verified by pressing the button on our system and visually checking the activation of the red led.
Requirement 3: When a user presses the button to use the ARC machine, the green LED should light up to signify the machine is not in use.	Verification 3: This can be easily verified by pressing the button on our system and visually checking the activation of the green led.
Requirement 4: Multiple machines must be able to send messages simultaneously	Verification 4: on the IoT device command line can be used to check what devices are trying to send messages to the device

Requirement 5: Control subsystem should last 4 days without being recharged.	Verification 5: We will conduct tests to make sure the Control subsystem can last 4 days.

This is a model of the MQTT publish/subscribe model that we will be using to communicate from the ESP-32 to the Raspberry PI



This is a model of how the AWS system will connect with our ESP-32 to update our database whenever a new availability signal is sent from the user.



We decided on using lithium ion batteries as they would be able to be recharged, causing less effort necessary on the ARC workers' end. In this way, they will only have to recharge with a USB-C cable, rather than needing to replace the batteries once the device dies. We did some preliminary battery life testing and came up with the result that our device should last about 5 days before it needs to recharge. This is the power calculations table:

Component	**Power Consumption**	**Power Consumption (5 days)**
Button	-	-
ESP32	55 mA * 17 hrs * 5 days	4675 mAh
Battery Life Indicator	10 mA * 17 hrs * 5 days	850 mAh
LEDs	12mA * 17 hrs * 5 days	1020 mAh
PIR Sensor	0.1 mA * 17 hrs * 5 days	8.5 mAh
		Total: 6553.5 mAh

During this meeting, we also started to finalize the initial list of parts that we wanted to buy for our PCB. This includes a FireBeetle ESP-32 (which was chosen for its deep sleep capabilities), sensors, and all of our power subsystem components. The following is the table of all the items and how much they cost.

Description	Manufacturer	Quantity	Extended Price	Link
FireBeetle 2 ESP32-E MCU	FireBeetle	2	$8.90	link
Raspberry Pi 3 - Model B	Adafruit	1	$35.00	link
12mm Button	Gikfun	1	$8.78	link
5x Stemedu HC-SR501 PIR Sensor	Stemedu	1	$9.99	link
RGB LEDs	Adafruit	1	$3.95	link
5x AM312 PIR Sensor	Aideepen	1	$9.59	Link updated
Rechargeable 18650 Battery	Panasonic	4	$4.99	link
Raspberry Pi 3 Power Adapter	Canakit	1	$9.95	link
2x18650 Battery Holder	E-outstanding	1	$8.99	link
BQ2057CTS Charge Controller	Texas Instruments	4	$1.56	link

TP4056 USB C Li-Ion Charger Module	diymore	1	$9.59	link
8GB Micro SD Card for Raspberry Pi	Verbatim	1	$6.19	link
MAX1797EUA+				

Date: 10/5

Objective: Begin initial PCB design

Everyone in our group are computer engineers so none of us have ever made a PCB before. The PCB design homework was helpful, but did not give much intuition behind why we were doing the things that we were doing. This process was very time consuming, and we still have not really gotten too much done today. Mostly worked on trying to find and import footprints from different websites for all of the parts that we wanted in our PCB. We also found out that there were many parts that did not already have a footprint available on the web, necessarily forcing us to make these footprints on our own. These footprints are very crucial to the finished product, and any small mistake can make the parts impossible to solder onto the PCB.

Date: 10/6

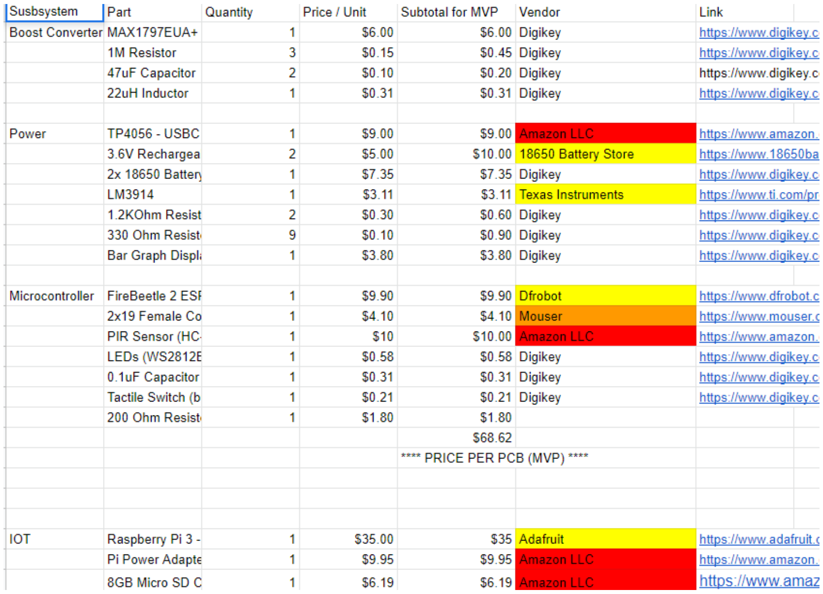
Objective: Finish custom footprints and finish schematic

We were able to import all of our new custom footprints as well as all of the other footprints that are required. Today we started researching heavily into the peripheral components required around our main parts. We learned how to utilize datasheets extensively for real applications, such as looking at the typical applications portion of the datasheet to see what other parts (resistors, capacitors, inductors) are necessary for the part to function in the application that we wanted. We started compiling a full parts list including these additional parts.

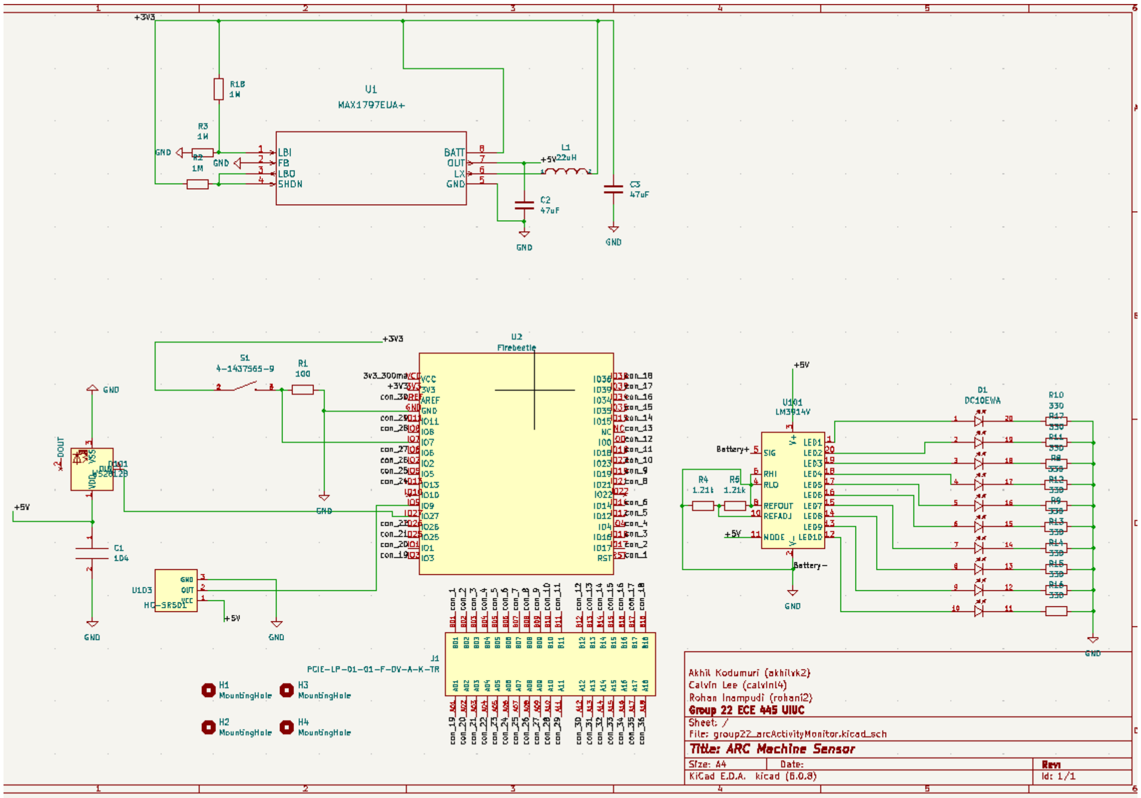
Date: 10/10

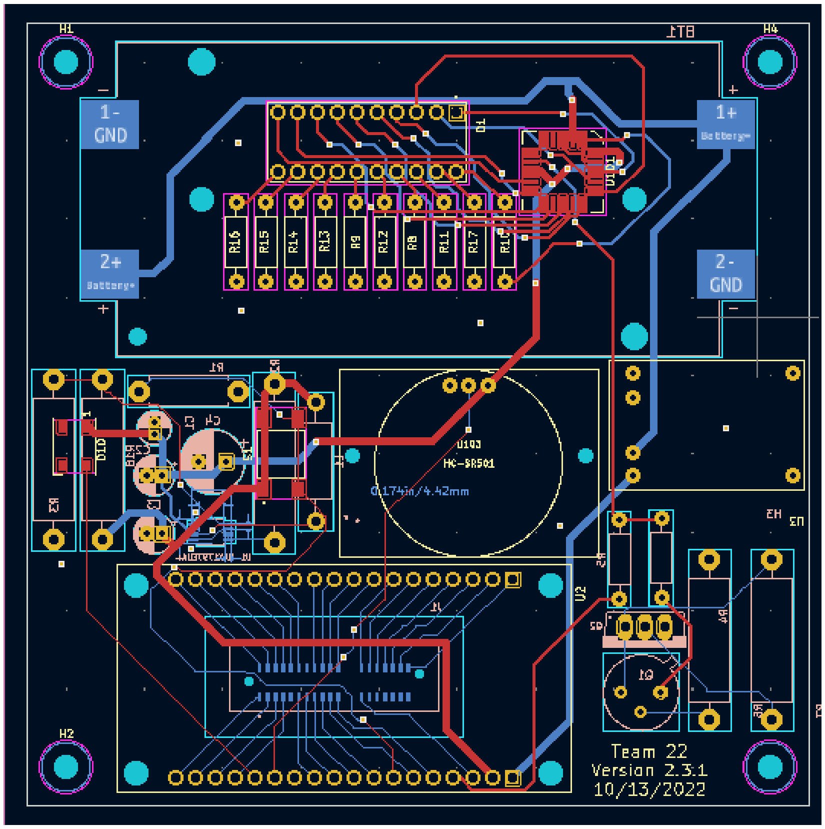
Objective: Finish PCB side and send it into PCB Way for first order

We had the entire schematic finished at the end of the last session. In this session we were able to finish the ERC as well as the DRC. We were able to connect all of the pieces on the footprint and got everything to pass the PCB review on PCB way. We also finished a finalized parts list with the links to buy as well as how much each part is. Attached is an image of PCB schematic and PCB as well as our parts list.



Susbsystem	Part	Quantity	Price / Unit	Subtotal for MVP	Vendor	Link
Boost Converter	MAX1797EUA+	1	$6.00	$6.00	Digikey	https://www.digikey.c
	1M Resistor	3	$0.15	$0.45	Digikey	https://www.digikey.c
	47uF Capacitor	2	$0.10	$0.20	Digikey	https://www.digikey.c
	22uH Inductor	1	$0.31	$0.31	Digikey	https://www.digikey.c
Power	TP4056 - USBC	1	$9.00	$9.00	Amazon LLC	https://www.amazon.
	3.6V Rechargea	2	$5.00	$10.00	18650 Battery Store	https://www.18650ba
	2x 18650 Battery	1	$7.35	$7.35	Digikey	https://www.digikey.c
	LM3914	1	$3.11	$3.11	Texas Instruments	https://www.ti.com/pr
	1.2KOhm Resist	2	$0.30	$0.60	Digikey	https://www.digikey.c
	330 Ohm Resist	9	$0.10	$0.90	Digikey	https://www.digikey.c
	Bar Graph Displ	1	$3.80	$3.80	Digikey	https://www.digikey.c
Microcontroller	FireBeetle 2 ESP	1	$9.90	$9.90	Dfrobot	https://www.dfrobot.c
	2x19 Female Co	1	$4.10	$4.10	Mouser	https://www.mouser.c
	PIR Sensor (HC-	1	$10	$10.00	Amazon LLC	https://www.amazon.
	LEDs (WS2812E	1	$0.58	$0.58	Digikey	https://www.digikey.c
	0.1uF Capacitor	1	$0.31	$0.31	Digikey	https://www.digikey.c
	Tactile Switch (b	1	$0.21	$0.21	Digikey	https://www.digikey.c
	200 Ohm Resist	1	$1.80	$1.80		
				$68.62		
				**** PRICE PER PCB (MVP) ****		
IOT	Raspberry Pi 3 -	1	$35.00	$35	Adafruit	https://www.adafruit.c
	Pi Power Adapte	1	$9.95	$9.95	Amazon LLC	https://www.amazon.
	8GB Micro SD C	1	$6.19	$6.19	Amazon LLC	https://www.amaz

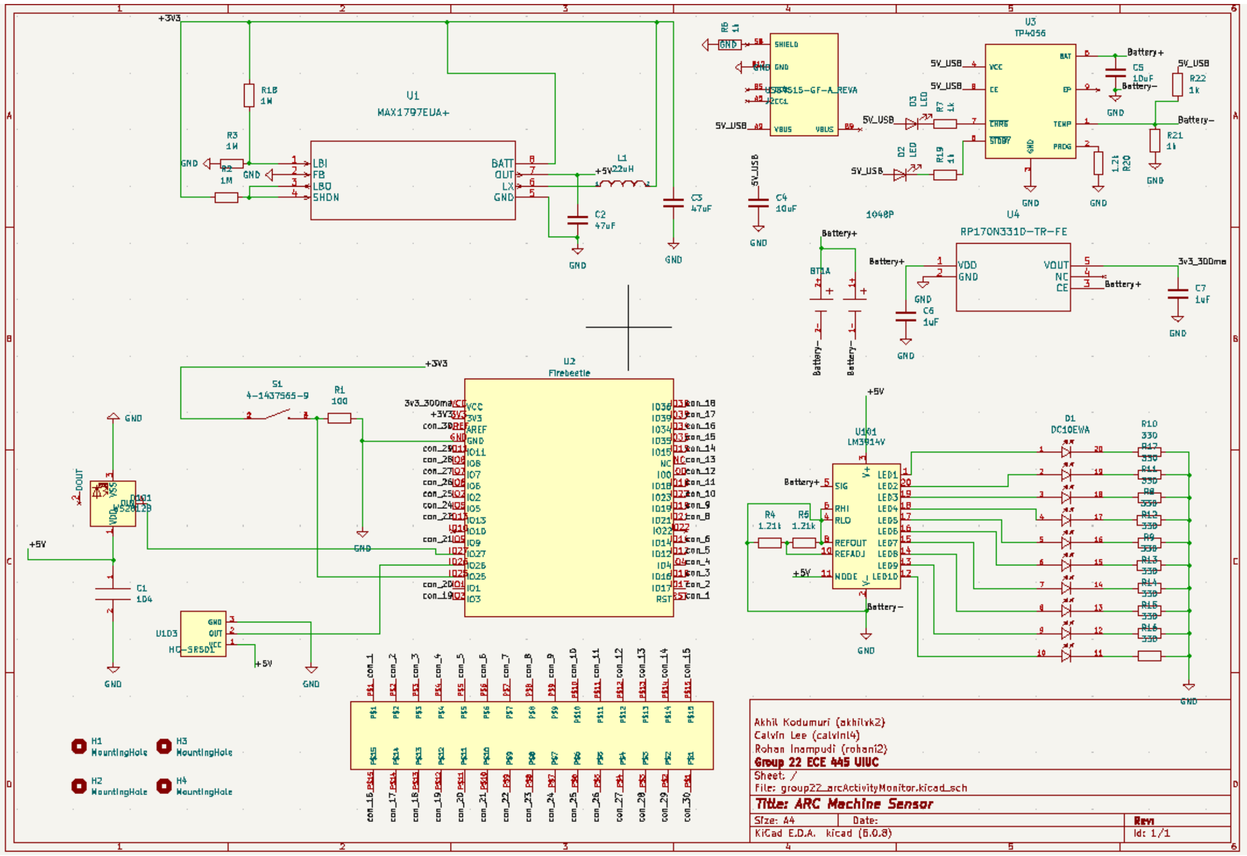

+3V3
R18
1M
U1
MAX1797EUA+
R3
1M
R2
1M
GND
LBI
FB
LBO
SHDN
BATT
OUT
LX
GND
L1
22uH
+5V
C2
47uF
C3
47uF
GND
GND
S1
4-1437565-9
R1
100
+3V3
U2
Firebeetle
3V3_300mA
VCC
3V3
AREF
GND
IO36
IO39
IO34
IO35
IO18
NC
IO0
IO16
IO23
IO19
IO21
IO22
IO14
IO12
IO4
IO16
IO17
RST
GND
+5V
DOUT
VDD
VSS
D101
WS2812B
C1
104
U103
HC-SR501
GND
OUT
VCC
+5V
GND
GND
H1
MountingHole
H2
MountingHole
H3
MountingHole
H4
MountingHole
J1
PCIE-LP-01-01-F-DV-A-K-TR
+5V
U101
LM3914V
Battery+
SIG
RHI
RLO
REFOUT
REFADJ
MODE
LED1
LED2
LED3
LED4
LED5
LED6
LED7
LED8
LED9
LED10
R4
1.21k
R5
1.21k
Battery-
GND
D1
DC10EWA
R10
330
R17
330
R11
330
R8
330
R12
330
R9
330
R13
330
R14
330
R15
330
R16
330
GND
Akhil Kodumuri (akhilvk2)
Calvin Lee (calvinl4)
Rohan Inampudi (rohani2)
Group 22 ECE 445 UIUC
Sheet: /
File: group22_arcActivityMonitor.kicad_sch
Title: ARC Machine Sensor
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (6.0.8)
Id: 1/1

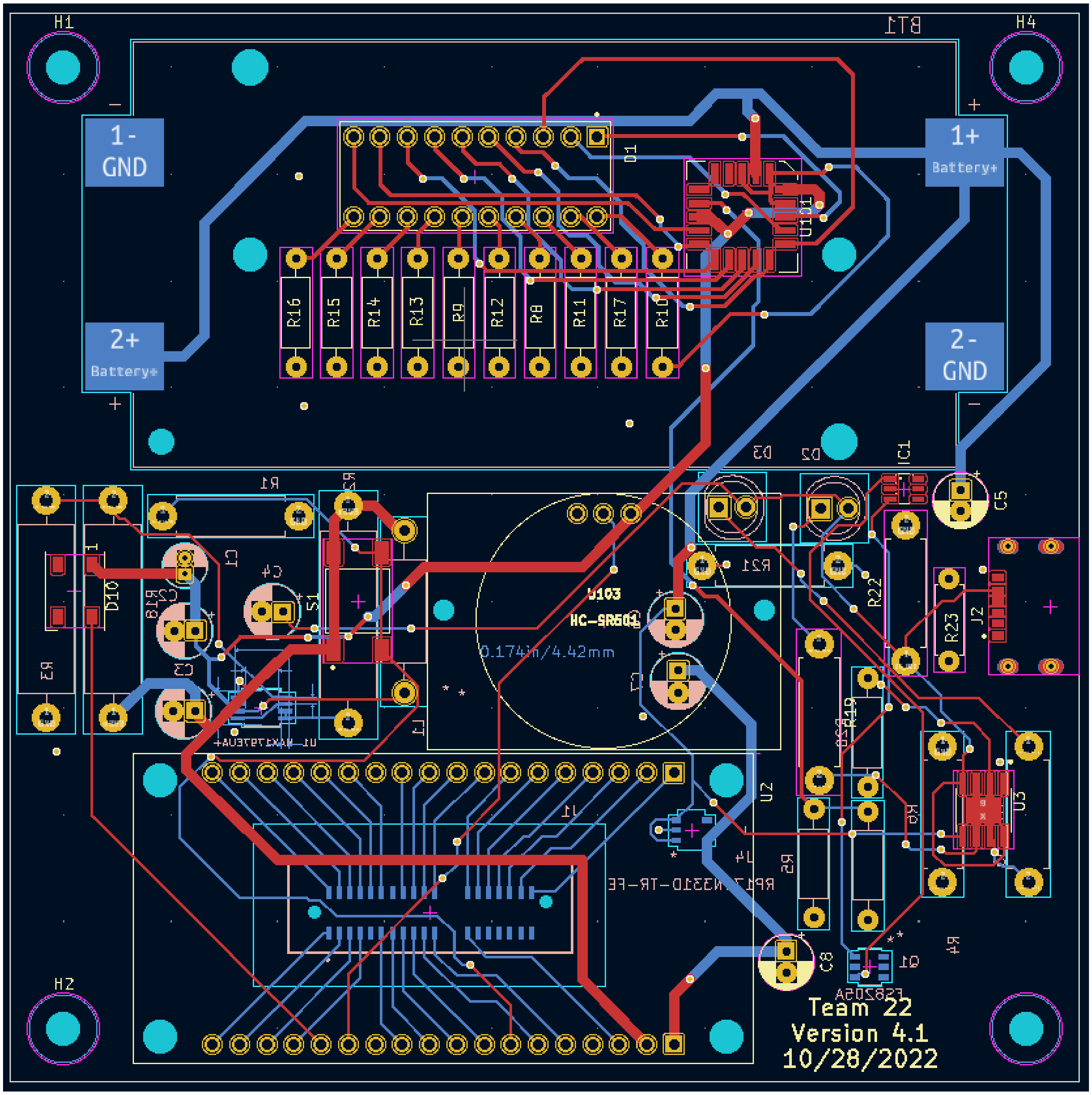

H1
H4
BT1
1-
GND
1+
Battery-
2+
Battery+
2-
GND
D1
U101
R16
R15
R14
R13
R9
R12
R8
R11
R17
R10
R1
R3
D101
C1
U103
HC-SR501
0.174in/4.42mm
U3
H3
U2
J1
R5
R4
R2
S1
H2
Team 22
Version 2.3.1
10/13/2022

Date: 10/17

Objective: Fix our PCB due to Gruev's comments about needing to implement power circuit

We met up and decided to meet up and change our design to work on our power circuit. We implemented the design of the USB-C module ourselves. Here are the updated schematic and PCB:





Date: 10/24

Objective: Research into the software side of things

Because our updated PCB that Gruev wanted us to make did not deliver with the first PCB order, our team focused on figuring out the exact logistics of how we wanted to implement the software side of things. We decided that AWS was not fit for our use case since AWS generally is expecting several thousands requests at once, and in our project, we would not be expecting more than fifty requests realistically. We therefore decided to pivot to a locally host MySQL workbench database, as well as a locally hosted python flask website. We also found some resources online on how to use the MQTT protocol.

Date: 10/31

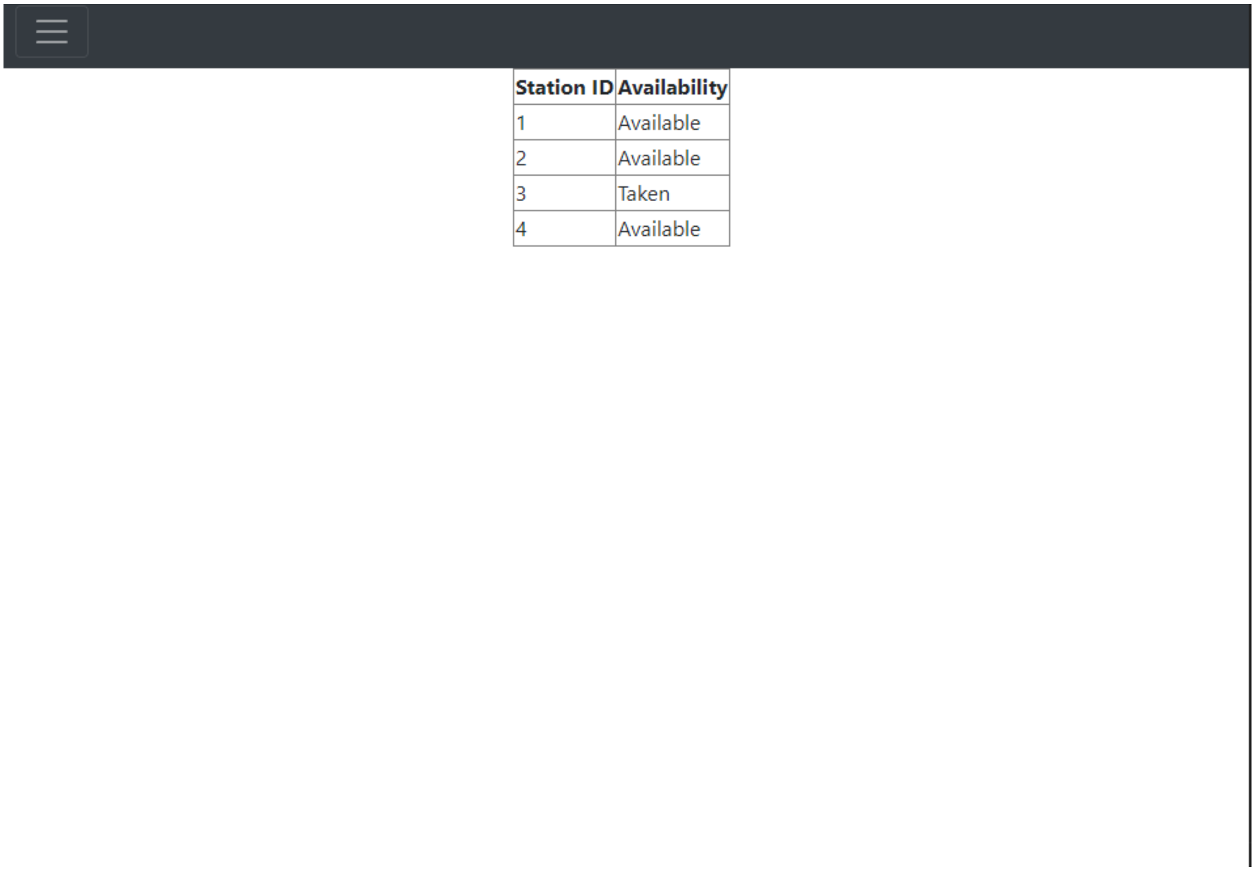
Objective: Start testing and working on building out our project

Our PCB was finally delivered and all of our parts came in as well. We immediately realized that a couple of our custom footprints were incorrect, and that we had wired a couple of the components incorrectly as well. We knew we had to reorder new components, but we first tried to breadboard out whatever components we could so that we could make sure there were no other revisions that were necessary.

Date: 11/10

Objective: Get prepared for the mock demo next week

We weren’t able to meet up last week because most of our team members were busy with exams. Mock demo is next week and we wanted to make sure that we have something ready to demonstrate. Unfortunately, our group member Akhil got sick at that point and was unable to help out, so me and Rohan decided to focus on software. We started working on building out the basics of the website and making sure that requests were able to go through. Attached is an image of the table on our website that is updated through python scripts:

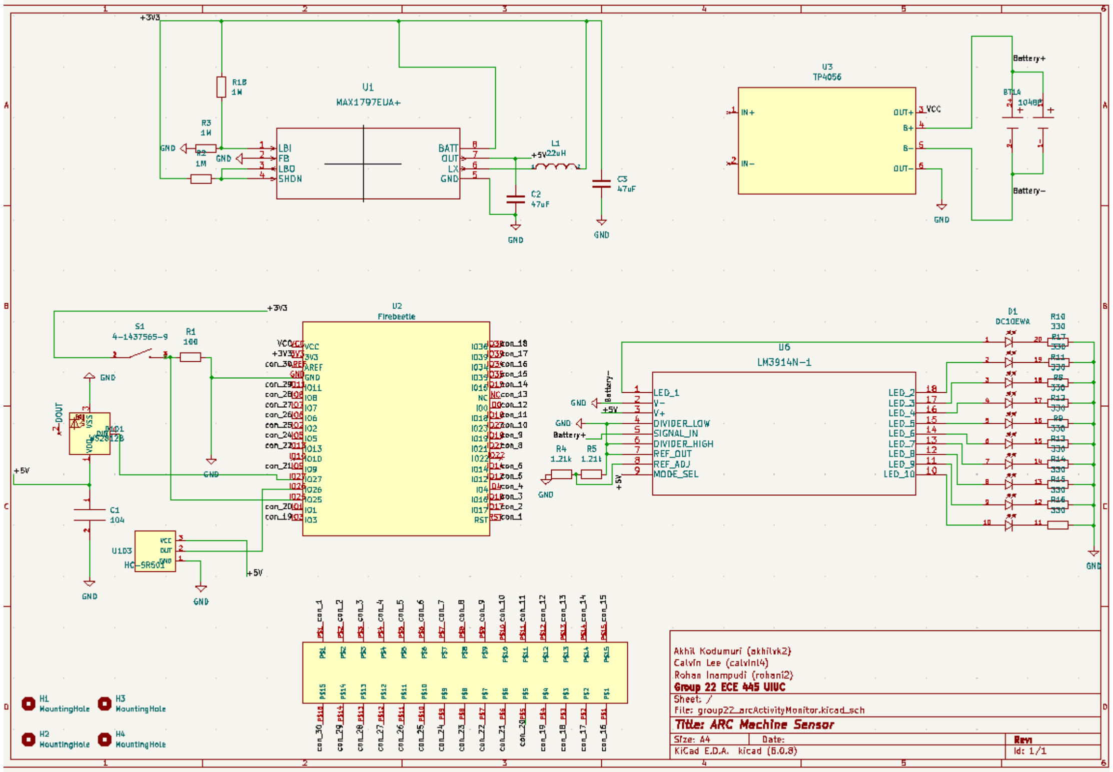


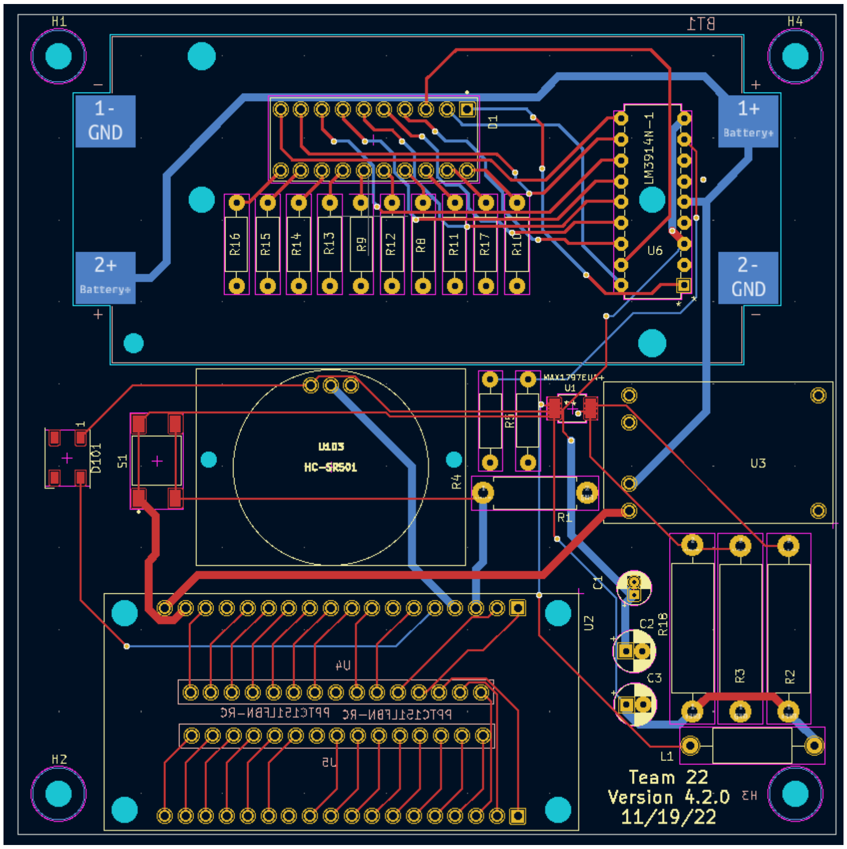
Station ID	Availability
1	Available
2	Available
3	Taken
4	Available

Date: 11/19

Objective: Remake PCB one final time

When our new pcb came in we realized that there were still a decent amount of errors, and that we needed to create a new final version with the original USB-C module so that we can cut back on the amount of places things would go wrong. We spent time creating a new PCB, but we were also able to get the entire website working, as well as the flow from the ESP-32 to the website. The script for the button as well as the PIR motion sensor failsafe was also implemented at this time. We sent in our order for a new PCB and had it shipped to my house over break to solder.


+3V3
R18
1M
U1
MAX1797EUA+
R3
1M
R2
1M
GND
LBI
FB
LBO
SHDN
BATT
OUT
LX
GND
L1
22uH
+5V
C2
47uF
C3
47uF
GND
U3
TP4056
IN+
IN-
OUT+
B+
B-
OUT-
VCC
Battery+
Battery-
BT1A
18650
GND
S1
4-1437565-9
R1
100
D101
WS2812B
DOUT
VDD
VSS
DIN
C1
104
+5V
U103
HC-SR501
VCC
OUT
GND
U2
Firebeetle
D1
DC10EWA
R10
330
U6
LM3914N-1
LED_1
V-
V+
DIVIDER_LOW
SIGNAL_IN
DIVIDER_HIGH
REF_OUT
REF_ADJ
MODE_SEL
LED_2
LED_3
LED_4
LED_5
LED_6
LED_7
LED_8
LED_9
LED_10
R4
1.21k
R5
1.21k
Battery-
Battery+
H1
MountingHole
H2
MountingHole
H3
MountingHole
H4
MountingHole
Akhil Kodumuri (akhilvk2)
Calvin Lee (calvinl4)
Rohan Inampudi (rohani2)
Group 22 ECE 445 UIUC
Sheet: /
File: group22_arcActivityMonitor.kicad_sch
Title: ARC Machine Sensor
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (6.0.8)
Id: 1/1


H1
BT1
H4
1-
GND
1+
Battery+
2+
Battery+
2-
GND
D1
LM3914N-1
U6
R16
R15
R14
R13
R9
R12
R8
R11
R17
R10
MAX1797EUA+
U1
D101
S1
U103
HC-SR501
R4
R1
U3
C1
C2
C3
R18
R3
R2
U2
U4
U5
PPTC151LFBN-RC
PPTC151LFBN-RC
L1
H2
H3
Team 22
Version 4.2.0
11/19/22

Date: 11/26

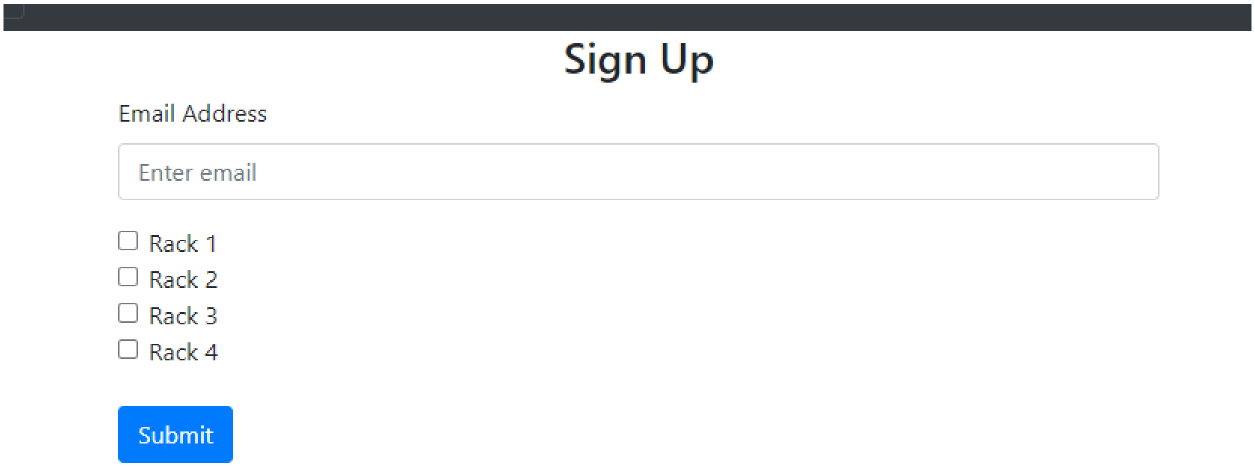
Objective: Solder new PCB

Our new PCB came in and I spent time in my dad's lab to solder our entire PCB together. I was able to, but once I did we realized that the ESP-32 was not getting enough power and was unable to supply 3.3V throughout our system. This was a fatal flaw, and we knew that we would not have a working PCB available to demonstrate for the final demo. We decided that a fully breadboarded demo is our only option, and we moved forward with that in mind.

Date: 11/30

Objective: Finish up final demo

This was our final working day. We were able to get our entire system breadboarded out, and were able to connect all of our subsystems together so that they would work in unison. We also implemented an email notification system so that people can sign up for certain gym spots and our system will let them know if it becomes available. It is unfortunate that we were unable to get the PCB to work, but we definitely learned a lot of things along the way that will help if I was to do anything similar in the future. We're ready for demo!



Sign Up

Email Address

Enter email

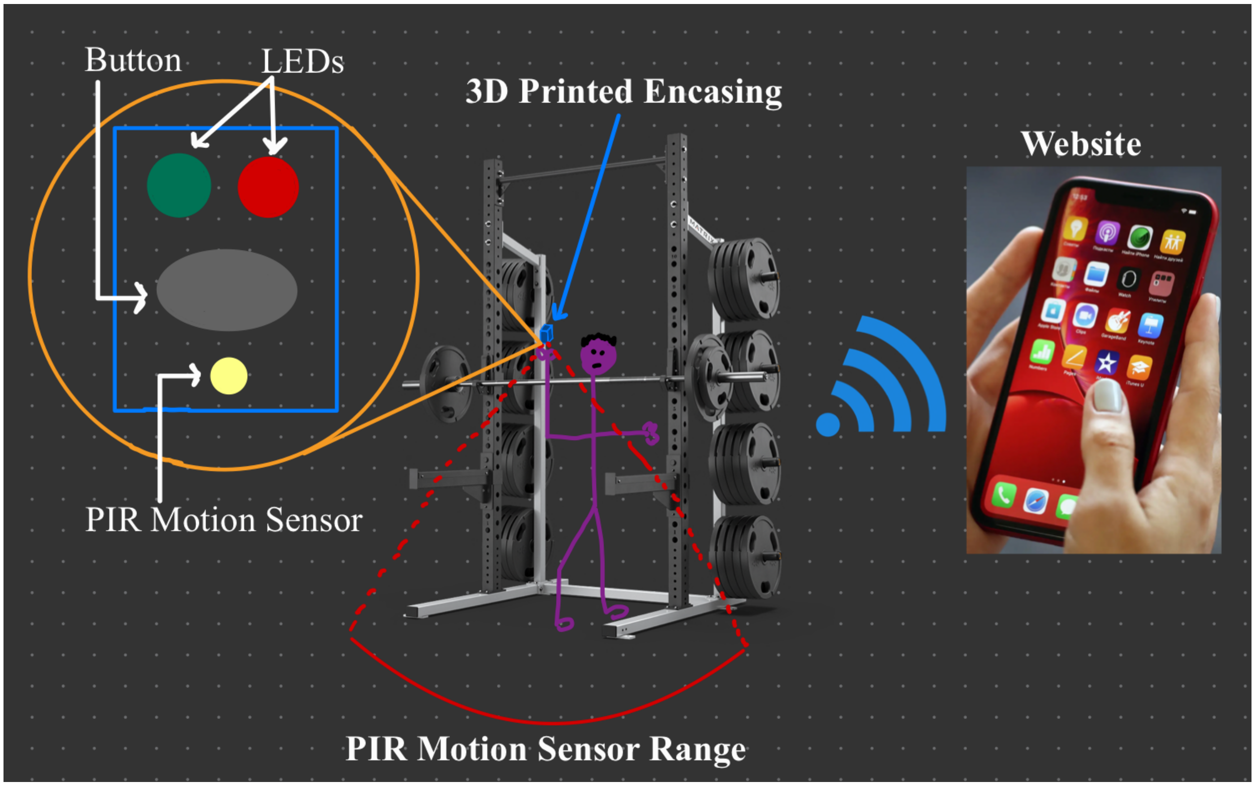
□ Rack 1
□ Rack 2
□ Rack 3
□ Rack 4

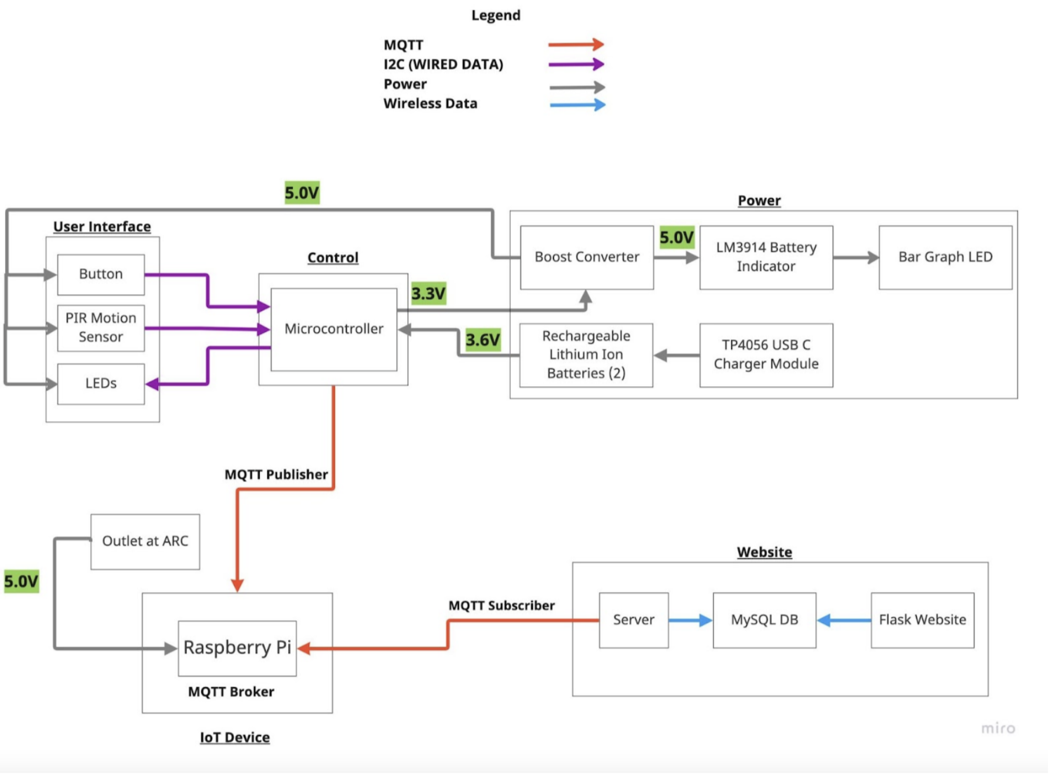
Submit

Date: 12/5

Objective: Create slides for final demo and work on final paper

We created all the slides for the final demo and created a new block diagram to better represent how our final system unfolded. We also put together all of our results and combined them into our final paper, which we also finished in this session. Finally done with the class! Attached are some of the new diagrams that we created for the final paper.


Button
LEDs
3D Printed Encasing
Website
PIR Motion Sensor
PIR Motion Sensor Range


Legend
MQTT
I2C (WIRED DATA)
Power
Wireless Data
5.0V
User Interface
Button
PIR Motion Sensor
LEDs
Control
Microcontroller
3.3V
3.6V
Power
Boost Converter
5.0V
LM3914 Battery Indicator
Bar Graph LED
Rechargeable Lithium Ion Batteries (2)
TP4056 USB C Charger Module
MQTT Publisher
Outlet at ARC
5.0V
Raspberry Pi
MQTT Broker
IoT Device
MQTT Subscriber
Website
Server
MySQL DB
Flask Website
miro