Electronic Replacement for COVID-19 Building Monitors at UIUC

Team 13 - Patrick McBrayer, Zewen Rao, and Yijie Zhang
ECE 445 Design Document - Spring 2021
TA: Yichi Zhang

1 Introduction

1.1 Objective

Human building monitors around the UIUC campus pose an unnecessary risk to students and staff, when a technological solution could be implemented. These monitors are in place to make sure that the right people are being let into the right places, using the Safer Illinois App developed by the university. This app provides access to testing results for the students, and faculty, and creates a randomized, unique building access pass, so it cannot be replicated. Even with this amazing software, we still put these human monitors at risk, and those who have to interact with them, at risk.

Our goal is to create an automated replacement system for these building monitors that still adequately protects the residents of the building. We will do this by creating a physical two-factor authentication network, that uses a central microcontroller to monitor the door for suspicious activity. As we do not have access to the backend of the Safer Illinois application, or the ability to use campus buildings as a workspace for our project, we will be designing a proof of concept 2FA system for UIUC building access. Our solution would be composed of two main subsystems, one that allows initial entry into the "airlock" portion of the building using a scannable QR code, and the other that detects the number of people that entered the space, or a form of people counter, to determine whether or not the user will be granted access to the interior of the building.

1.2 Background

COVID-19 has put everyone across the world in dire straits. Every system or service that is a part of life has been put through one of the most extreme stress tests we have seen in over a hundred years. However, places like the University of Illinois have been trend setters for solutions to the many problems that COVID has placed upon these systems. They have developed new forms of testing, state of the art tracking applications, and many other procedures and systems. However, even with these huge strides in innovation, some approaches are archaic in comparison.

Studies show that appropriate social distancing and mask wearing results in an approximately 3% transmission rate of SARS-CoV-2 [1]. While this seems positive in the grand scheme of things, these monitors are still at risk, and taking away the human element could potentially save these people from infection. On average, humans make about 4-6 errors per hour [2]. Let's say in the case of a building monitor, this error includes misuse of masks or improper social distancing, this creates great risk for all involved, and is not isolated to the monitor either. Since the students/staff being checked are interacting with those monitors, they make errors as well, further complicating the issue. People are not perfect, and human error will always find a way to create problems. With a fully automated system, we can remove that human element, and create a system that can monitor all buildings on campus around the clock.

1.3 Physical Design

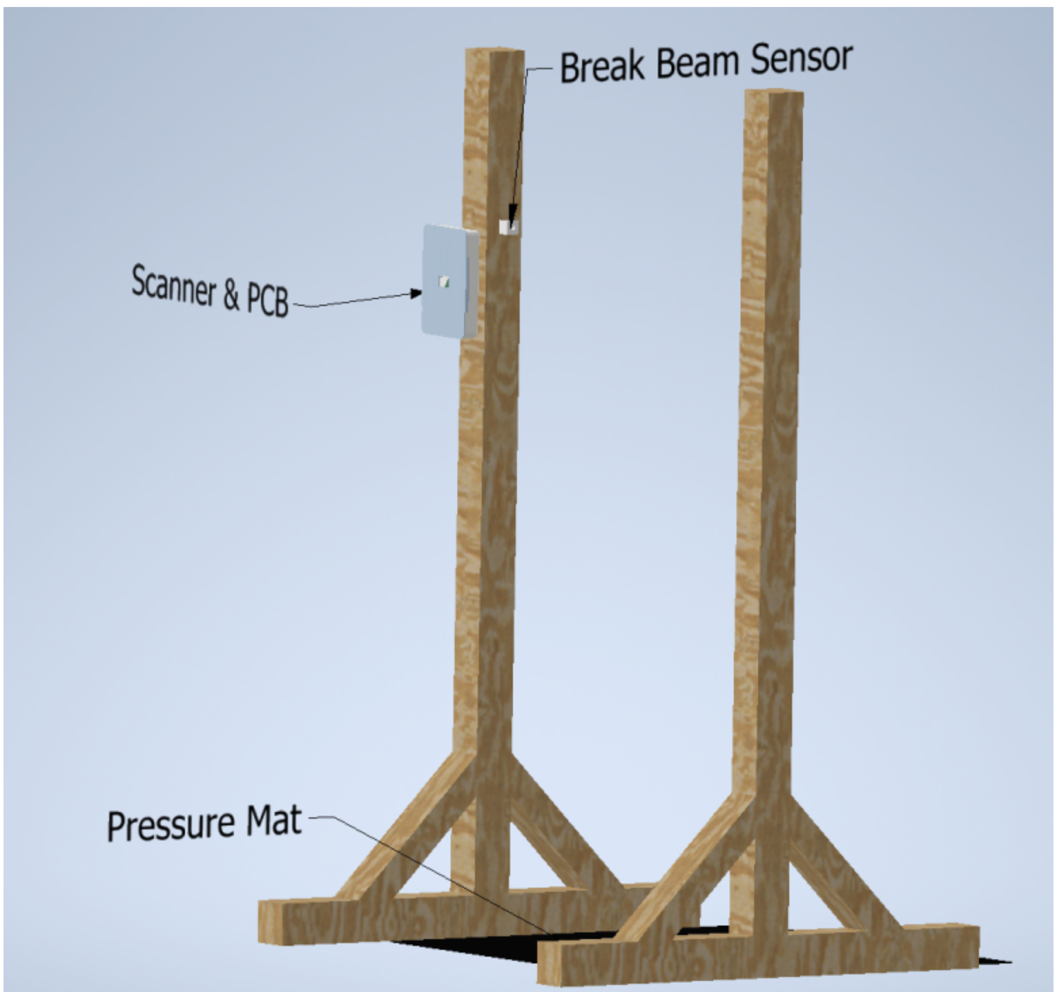

Figure 1: High Level Pictorial Diagram of Design

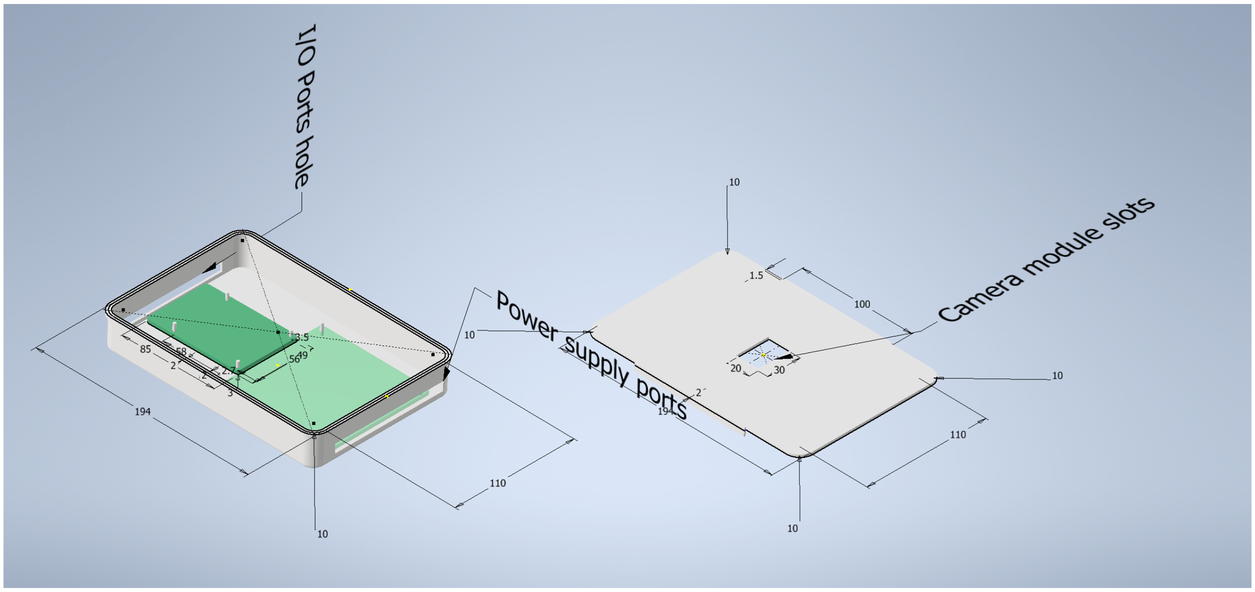

Figure 2: Low Level Pictorial Diagram of Design (Control Unit + Beam Placement)

1.3 High-Level Requirements

- QR Code Scanner should be able to determine if the user is allowed primary entry using the encoded information from the generated QR code, and refer back to that user when the next scan is performed to prevent repeat access being granted.
- Break-beam sensor and pressure mat within the airlock must work synchronously to determine the presence of a single student/faculty member, as well as accurately keep track of entries with a <5% error rate.
- Must be able to process each person within a 30 second time frame, to prevent congestion at the entryways of buildings across campus. The 30 seconds will begin with the QR code scan, and end when the second door is unlocked.

2 Design

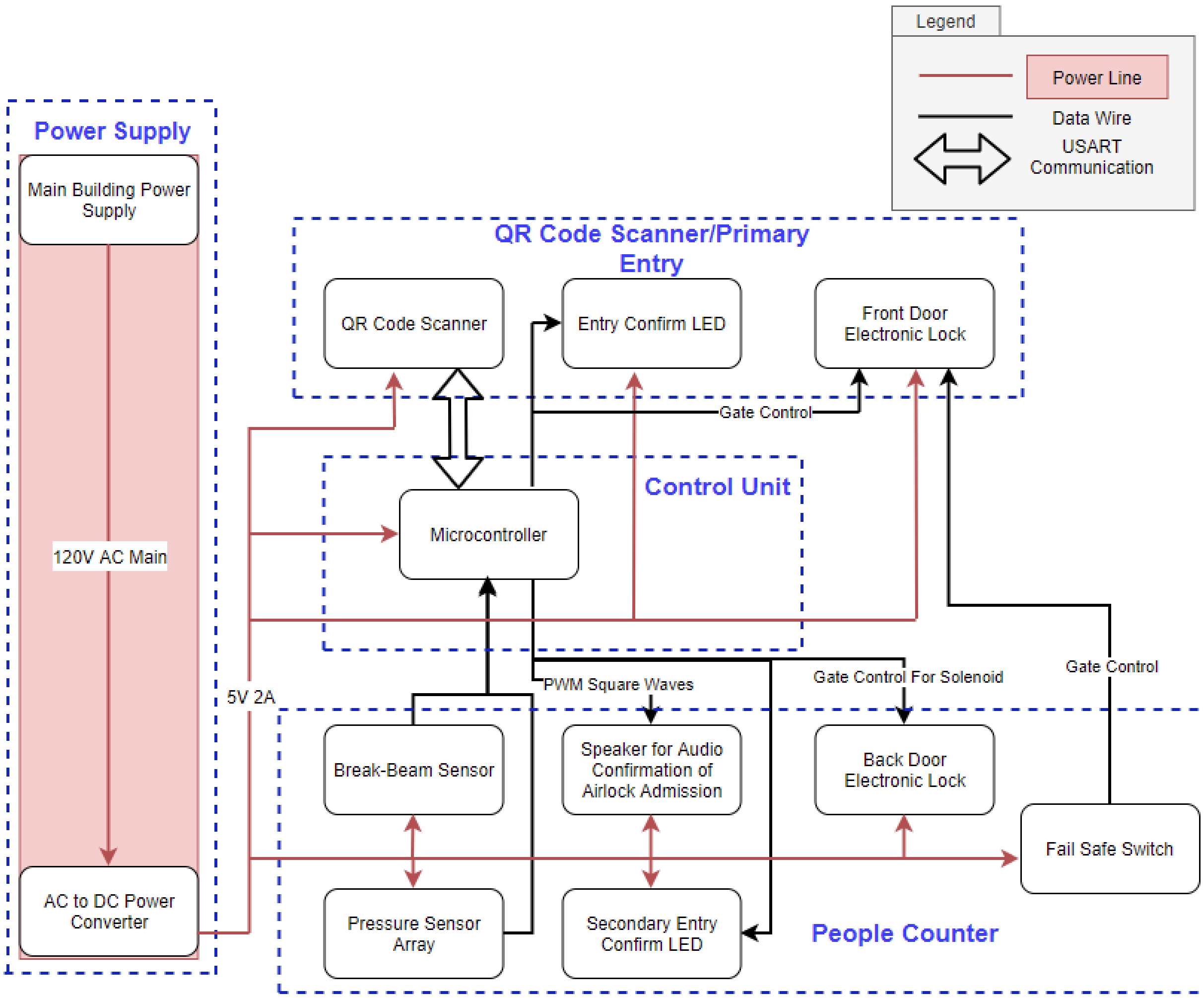

Figure 3: Block Diagram

2.1 Power Supply

A power supply is required to keep the communication network up continually. Power from the main building supply will charge a battery, which is then regulated to 5V for the rest of the system.

2.1.1 Main Power Supply

This module has two parts to work properly, the external power source and the AC to DC Power Converter. For the external power source, since the system was installed at the door, it can get power directly from the building. Since we need to provide power to the PCB board and the solenoids, we need to convert the AC power from the supply to the DC power that can be used in the system. With the converter, we can convert 120V AC to 5V DC, with a maximum regulated current of 2A.

Requirements	Verification
1. The external power source should provide stable power of 100V-240V, which can fulfill the input requirement of the AC to DC converter. In the United States, since the supply voltage is 120V, we may assume the external power source will be 120V (+/- 15%). 2. The AC to DC converter must be able to convert 100V-240V AC to 5V (+/- 5%) 2A(+/- 5%) DC, which is needed to power the system. Since the microcontroller can handle 5.5V at most, setting the voltage up to 5*1.05 = 5.25V is safe for the device.	1A. To verify an AC power, set the multimeter to AC mode. Use one hand to hold the red and black probes to ensure safety. Do not let the metal parts of the probes touch each other to avoid short circuit. Insert the red probe into the smaller slot and the black probe into the larger slot. If it gives a reading of 110v-120v, the outlet is qualified. 2A. To verify the DC power converted by the converter, a multimeter will be used to check if the output voltage of the converter is equal to 5V. If it gives a reading of 4.75 - 5.25V, the outlet is qualified. 2B. To make sure the power supply is enough for both the components on the PCB board and the solenoid, we need to verify the converter can provide constant 2A current. To verify it, we can still use the multimeter under the current mode. If it gives a reading of 1.9A - 2.1A, the outlet is qualified.

2.1.2 Breakout Connector

Requirements	Verification
1. This connector should break out the micro USB's pins to VCC, GND, ID, D+, and D-. With the power output remaining consistent to the 5V 2A output of the Converter.	1A. Use a wall outlet adapter to connect to a micro USB to USB A cable. Plug the micro USB part into the Breakout Connector. Use the multimeter to measure the voltage. If the measurement is correct, then the Breakout connector is qualified.

2.2 QR Code Scanner/Primary Entry

This part of the system will be installed on the first/outside door. It is used to make sure everyone entering the area between the outside and the inside door is safe and authorized.

2.2.1 QR Code Scanner

We are planning to use Raspberry Pi 4 and a camera module to build a QR Code Scanner by ourselves. With the help of OpenCV library, we can implement a program continuously reading the QR code and sending the information to the microcontroller. The QR Code Scanner is able to scan the QR code on every student's phone. Only when the QR code is scanned, the information will be sent to the Control Unit. The scanner needs the reliable power of 5V to work properly, and uses UART RS232 to connect to the microcontroller.

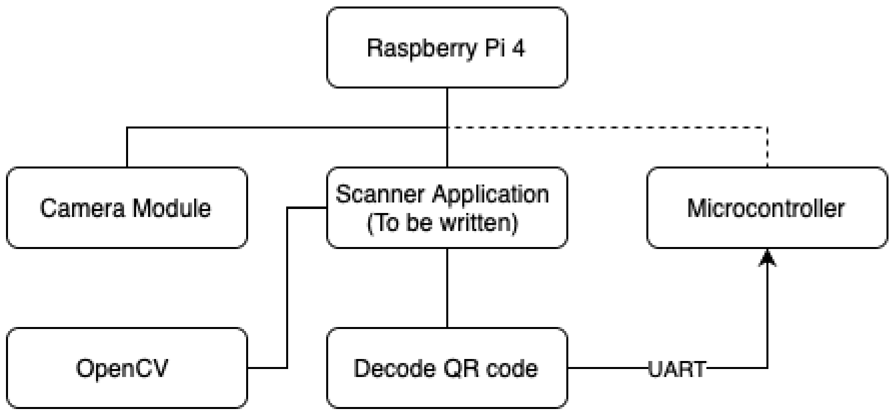

Figure 4: The workflow of Raspberry Pi based Scanner

Requirements	Verification
1. Since the system must be able to process each person within a 30 second time frame, it's important to have the scanner have a high accuracy (> 95%) to detect the QR code. 2. The scanner will be used by many people with different models of phone, so it must be able to handle them. As the scanner will be installed outdoors, it must be able to work properly under the sunlight. 3. The scanner will send the information retrieved from the QR code to the microcontroller within 3 seconds.	1A. Connect the scanner to a monitor directly. Instead of sending information to the microcontroller, we may display the information on the monitor. Make sure the QR code scanner can output the pre-set string correctly. 2A. Use different models of phones to display the QR code, and use the scanner to scan them. The expected strings included in the QR codes should be displayed on the monitor. 2B. Repeat the previous step in different environments, and make sure it works well under the sunlights. 3A. Connect the QR code scanner with an Arduino using UART for the test purpose.

	Connect the Arduino to the computer, and open the Serial Monitor to check whether the Arduino can receive the information decoded by the scanner.

2.2.2 Entry Confirm LED

We are using a two-color LED, LTL1BEKVJNN, as the entry-confirm LED. When the QR Code Scanner scans the correct QR code, it will send the information to the Control Unit. After that, the Control Unit will send a control signal to the LED driver, which is powered by an inverter, and will swap the color of the LED from red to green, informing the user that they are allowed entry.

Requirements	Verification
1. The LED runs well under 5V (+/- 5%) and 30mA (+/- 5%). 2. The LED must swap colors when a control signal is sent to the inverter. 3. The Entry Confirm LED must be clearly visible from 0.5 meters away.	1A. Build a simple circuit containing only LED, power supply and resistor. The power supply will be 5V, and the resistor will be 100Ω. 2A. Connect the functioning LED with the red input on the output of the inverter and the green input connected to the input to the inverter. 2B. Send a control signal, and check that the LED switches colors 3A. Keep the previous test, check the LED from 0.5 meters away.

2.2.3 Front Door Electronic Lock

At the same time of the LED on, the Control Unit will also unlock the outside door lock to let the person enter the People Counter area. We will be simulating an electronic locking mechanism through a Mini Push-Pull Solenoid actuator, which signifies the lock/unlocking of the door.

Requirements	Verification
1. The lock will work under 5V, and when receiving the signal from the microcontroller, it must unlock/lock the	1A. Use the Arduino as the power source to test the solenoid.

door and keep that state until the power status changes.	1B. Build a small circuit on a breadboard, connect the Arduino's 5V output to a button, as well as connect the button to the positive side of the solenoid. 1C. Connect the negative side of the solenoid directly to the GND pin on the Arduino. 1D. Verify when the button is pushed, which means the 5V voltage power supply is turned on, the lock is immediately unlocked. 1E. Verify when the power supply is cut, the lock is locked.

2.3 Control Unit

The Control Unit is able to control the whole system as well as the computation of the logic. It will control and decide the status of the inside door based on the data collected from the sensor, as well as use the information scanned from the QR code to determine initial access into the people counting subsystem.

2.3.1 Microcontroller

The Microcontroller is the core of the Control Unit. It will be in charge of the control of the whole system and make the decision of the lock status of the inside door. We will be using the recommended ATMEGA32-16PU to run our system.

Requirements	Verification
1. One easy way to program the ATMEL microcontroller is using the Arduino as an ISP. 2. The microcontroller needs a bootloader to write the program into it successfully using the Arduino IDE. 3. The microcontroller can control the digital output pins correctly 4. The microcontroller can transfer data over Serial UART at the baud rate of 9600 and 115200.	1A. Set the Arduino to the ISP mode. Write the ArduinoISP sketch to the Arduino. 1B. Make sure the IDE doesn't show any error logs. 2A. Burn the bootloader into the microcontroller using Arduino IDE. 2B. Make sure the IDE doesn't show any error logs. 3A. Write a short LED blink code, and write it into the microcontroller using the Arduino

	as the ISP. 3B. Connect a LED bulb to one of the digital pins, and check whether it can flash. 4A. On the basis of the previous code, write a new part of the code that every time the LED blinks, the microcontroller will send the Serial message. Connect the RX/TX pins on the ATMEGA32 to the Arduino. 4B. Push the reset button on the Arduino. Open a serial console. Make sure every time the LED blinks, there will be a message sent by the microcontroller displayed in the monitor.

2.4 People Counter

This module can count the people entering the first/outside door. When more than one person enters the area between two doors (which means unauthorized people enter), the second door will not open and people will be notified to leave the area.

2.4.1 Break-Beam Sensor

The sensor will count the number of times that the line connecting them is broken by some external object. This information will then be sent to the microcontroller to be processed, and determine the number of people who crossed the passageway.

Requirements	**Verification**
1. The Break-Beam Sensor can work under 5V DC and transfer the information to the microcontroller immediately. 2. The sensor must have the longest detection distance >= 50cm, and provide enough accuracy to avoid the false alarm.	1A. Use the Arduino as a prototype test. Connect the sensor to the 5V and GND pin on the Arduino. To read the feedback of the sensor, connect the data line to a digital pin of the Arduino. 1B. Connect a LED bulb to another digital pin on the Arduino. Write a short code, let the Arduino send a high signal to the LED pin when it receives the high signal from the sensor pin. 2A. Put the sensor in the distance of 50cm. When an obstacle appears between the receiver and the sender, the LED will be

	turned on. Perform multiple tests and calculate the accuracy rate to see if it is within the range we need.

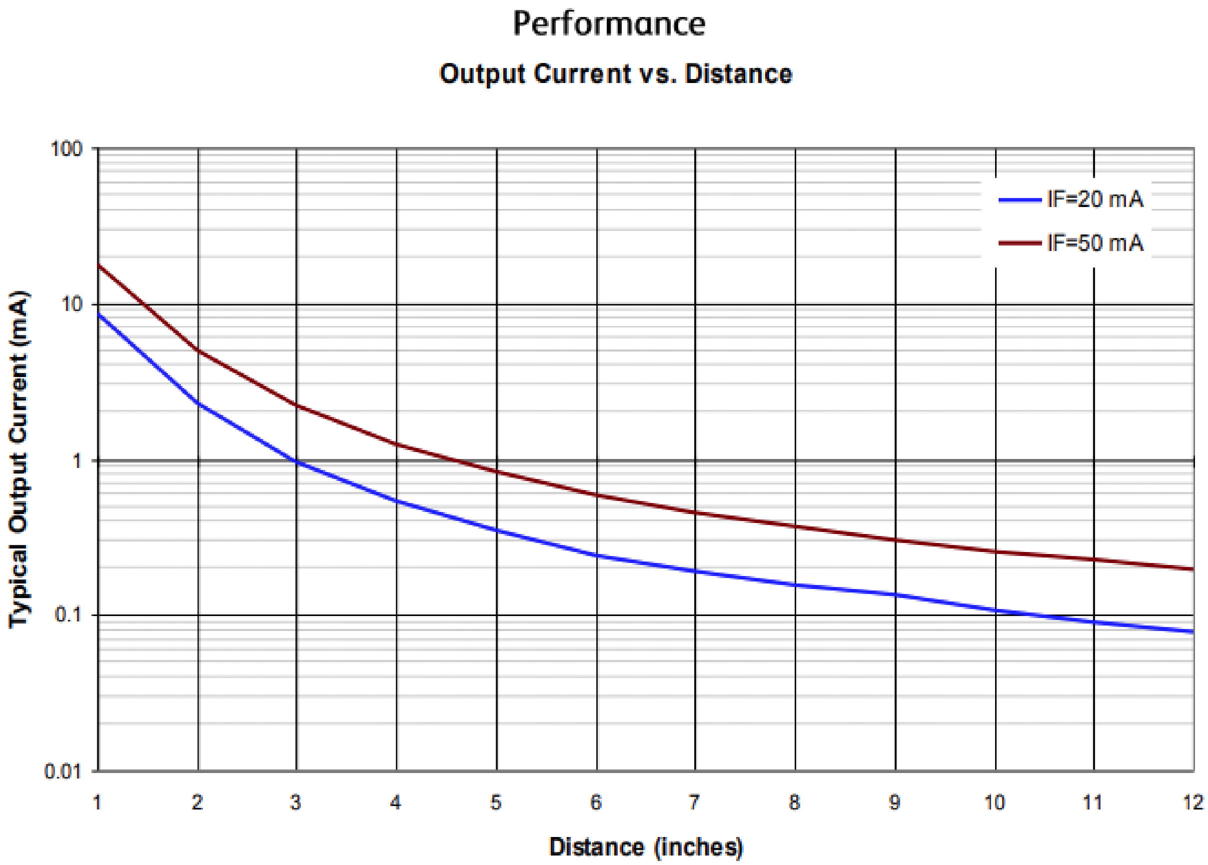

Figure 5: Current output of our Break-Beam Sensor vs. Distance from Data Sheet [5]

2.4.2 Pressure Detection Sensor

This sensor will detect the number of unique pressure points present across the mat, and send that information to the microcontroller, to add more information required to determine the number of people present. This will be done with an array of small pressure sensors, underneath a standard mat. We will then process the information sent from that array, to better understand the placement of feet on the mat.

Requirements	Verification
1. The Pressure Detection Sensor needs to be installed at a place where people who enter the building must pass.	1A. Try to enter the building multiple times with the Pressure Detection Sensor installed to see if it can detect every time

2. Across a wide range of unique pressure profiles, the MCU must be able to make a rough estimate as to how many feet are preset on the mat, as well as consider outliers such as crutches or wheelchairs	a person enters. 1B. We will construct 6 unique tests, to determine the accuracy of our system: - 1 person walking in normally as a baseline - 1 person walking in sideways - 2 people walking in back to back - 2 people walking in sideways - Wheelchair entry - Person walking in on crutches 2A. Use the Arduino to test the pressure sensor. Connect the sensor to one of the digital pins of the Arduino. 2B. Write a code to monitor the pin. Open the serial monitor to make sure the sensor sends back the right data.

2.4.3 Speaker & Amplifier

The speaker will be used to tell the person whether he is authorized to enter the next door, or there's more than one person in the area so they are required to leave and enter the area one by one. It will be driven by the MONO 2.5W CLASS D AUDIO AMPLIFIER, which is working under 5V and can drive the speaker of 2.5W at 4Ω.

Requirements	Verification
1. The speaker must be clearly heard when people stand in the counter area. 2. The system must be able to communicate with the I2S interface on the MCU. 3. The Amplifier can drive the speaker of 2.5W at 4Ω.	A. Attach the amplifier to a speaker with 4Ω and 2.5W. Turn on the amplifier and the speaker to see if the amplifier works properly. B. Repeat this step in the counter area to see if they can be heard clearly.

2.4.4 Secondary Entry Confirm LED

We are using a two-color LED, LTL1BEKVJNN, as the entry-confirm LED. The LED will be switched on when the person is allowed to enter the second door.

Requirements	Verification
1. The LED runs well under 2.1V (+/- 5%) and 30mA (+/- 5%). 2. The Entry Confirm LED must be clearly visible from 0.5 meters away.	1A. Build a simple circuit containing only LED, power supply and resistor. The power supply will be 5V, and the resistor will be 100Ω. 2A. Keep the previous test, check the LED from 0.5 meters away.

2.4.5 Back Door Electronic Lock (Mini Push-Pull Solenoid)

The control unit will decide whether the person is allowed to enter the second door, and send the signal to the Electronic Lock. Using a Mini Push-Pull Solenoid, we can easily control the process of lock/unlock by controlling the voltage.

Requirements	Verification
1. The lock will work under 5V, and when receiving the signal from the microcontroller, it must unlock/lock the door and keep that state until the power status changes.	1A. Use the Arduino as the power source to test the solenoid. 1B. Build a small circuit on a breadboard, connect the Arduino's 5V output to a button, as well as connect the button to the positive side of the solenoid. 1C. Connect the negative side of the solenoid directly to the GND pin on the Arduino. 1D. Verify when the button is pushed, which means the 5V voltage power supply is turned on, the lock is immediately unlocked. 1E. Verify when the power supply is cut, the lock is locked.

2.4.6 Emergency Button

When unexpected system failure occurs, like both the inside and outside doors are locked, people in the People Counter area can use this button to manually unlock the outside door to leave.

Requirements	Verification
1. The emergency button must be easily-pressible. 2. The emergency button must be large enough for people to find.	1A. Press the button and make sure it can be pressed without problem. 2A. Put the button at 1 meter distance and make sure it can be found easily.

2.5 Case

Requirements	Verification
1. The case should be IP55 environmental protection.	1A. Water projected by a nozzle (6.3 mm (0.25 in)) against enclosure from any direction. Check the inside of the case, make sure it's dry. 1B. Put the case in a dusty environment for over 24 hours. Open the case and make sure there's no dust in it.

2.6 Server & Software

The server will be running on a Raspberry Pi 4 (the same as the one used as QR code scanner). On the server, we need an OS, a server application (Apache), a database (MySQL). The server will be used to store user information and generate the QR code. The PHP code will be running on the server to retrieve user information from the SQL database and generate the corresponding QR code.

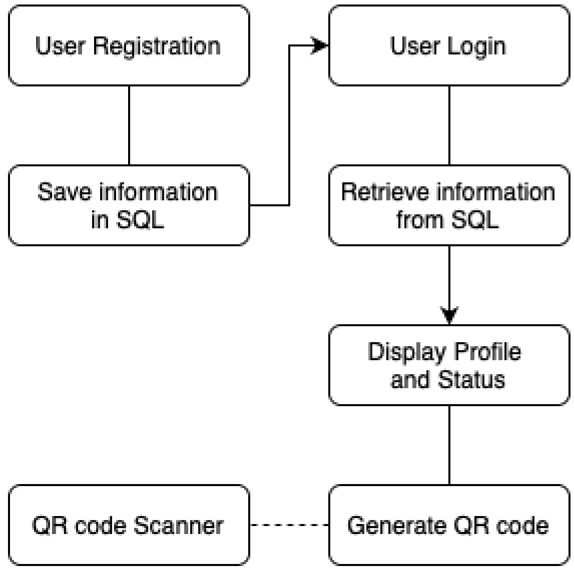

Figure 6: The Workflow of the server

Requirements	Verification
1. Can be accessible from different mobile devices by different users, and everyone can only retrieve their own information after they log into the system. 2. The website can generate a QR code immediately when the user wants to scan the QR code to enter the building. 3. The created QR code can be scanned by the QR code scanner system we built correctly and immediately.	1A. Set up a test database with different usernames and covid-19 test results. There should be multiple users with several different status to make sure every scenario can be considered. 1B. Log into the website with different usernames from different phones, and make sure the displayed user profiles are correct, including the name and the preset test results. 2A. There will be a QR code on every user's front page. 2B. Use a QR code scanner software provided by the phone to scan it, compare the decode information with the actual status stored in the database, make sure they are identical. 3A. Using the QR code scanner we set up before to scan the generated QR code, and make sure the information retrieved from the QR code is correct and the same as the user's status.

2.7 Circuit Schematics

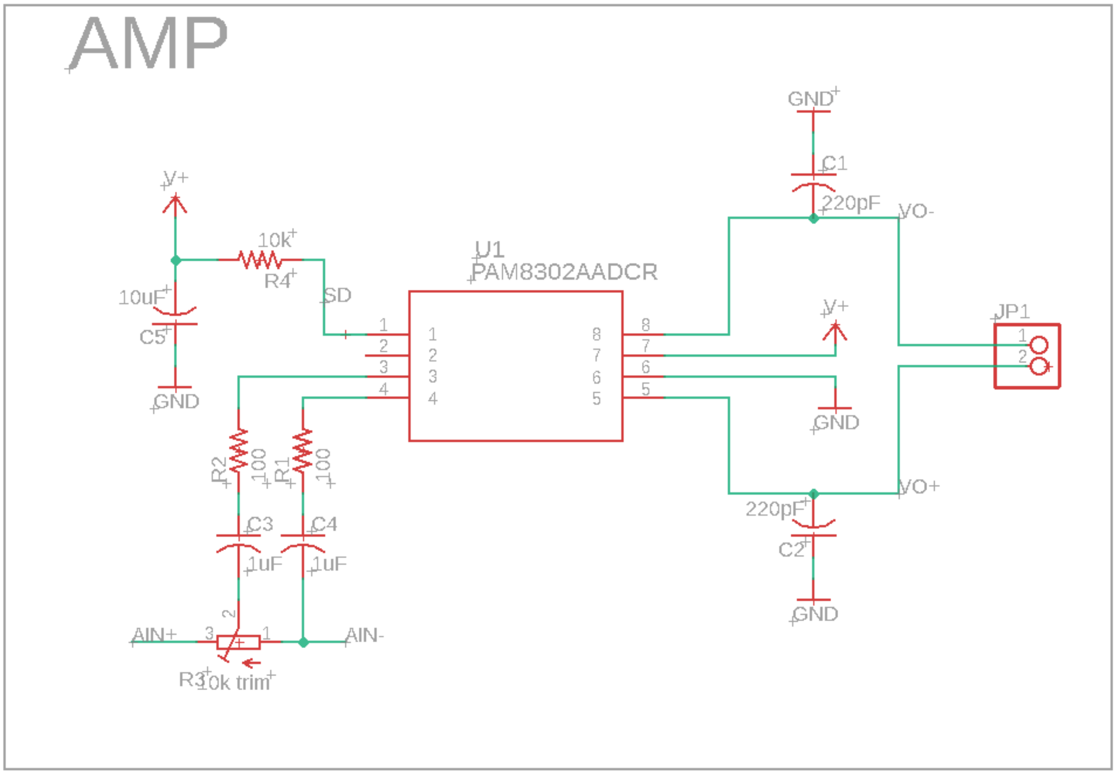

Fig. 7: Amplifier Schematic

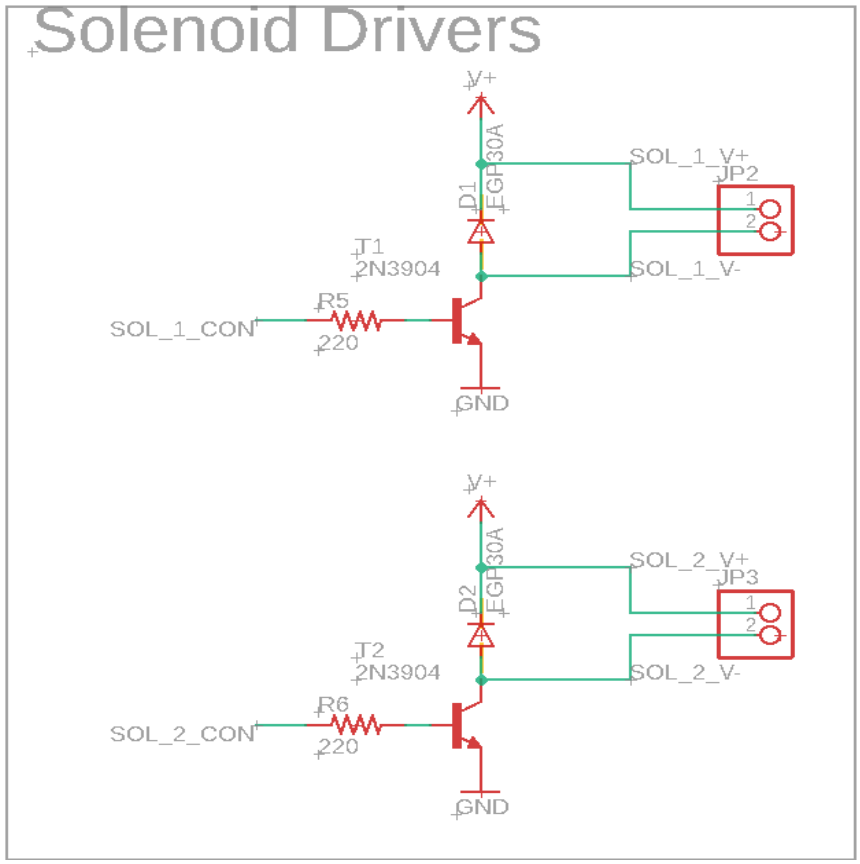

Figure. 8:. Solenoid Driver Schematic

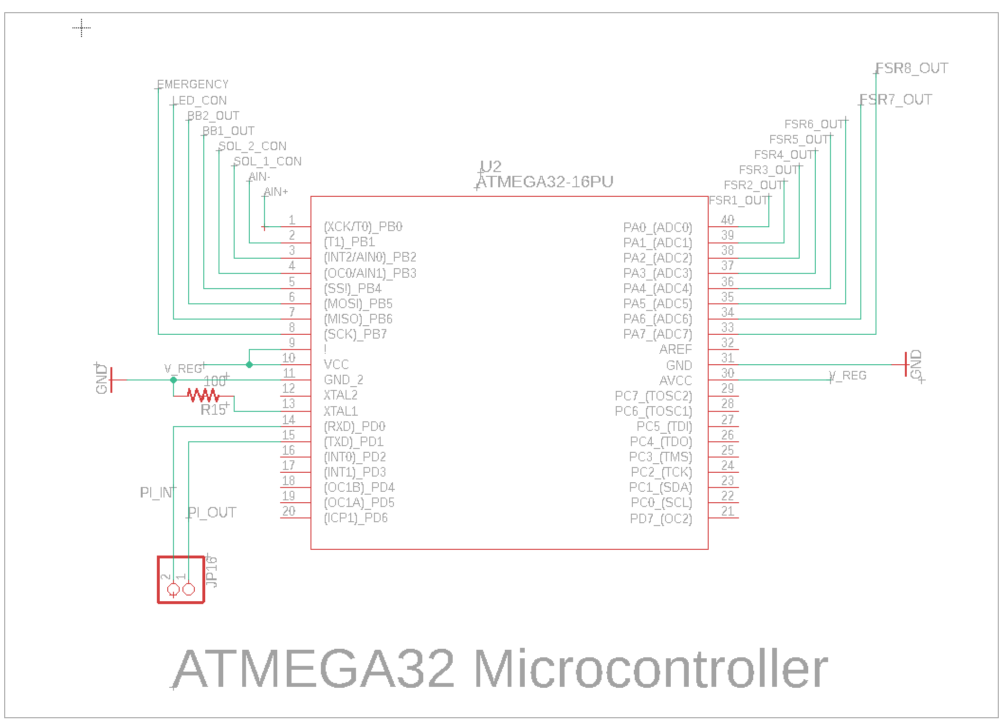

Figure 9: Control Unit Schematic

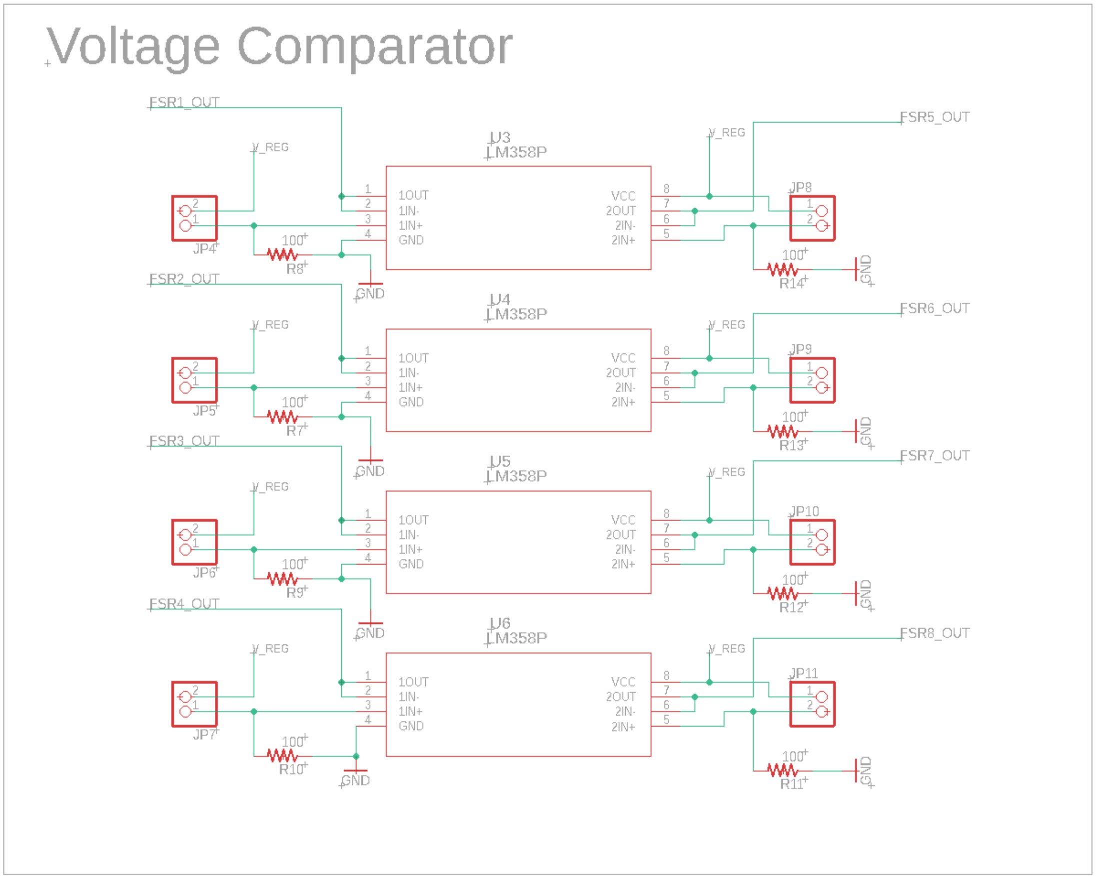

Figure 10: Pressure Sensor Array Voltage Comparator

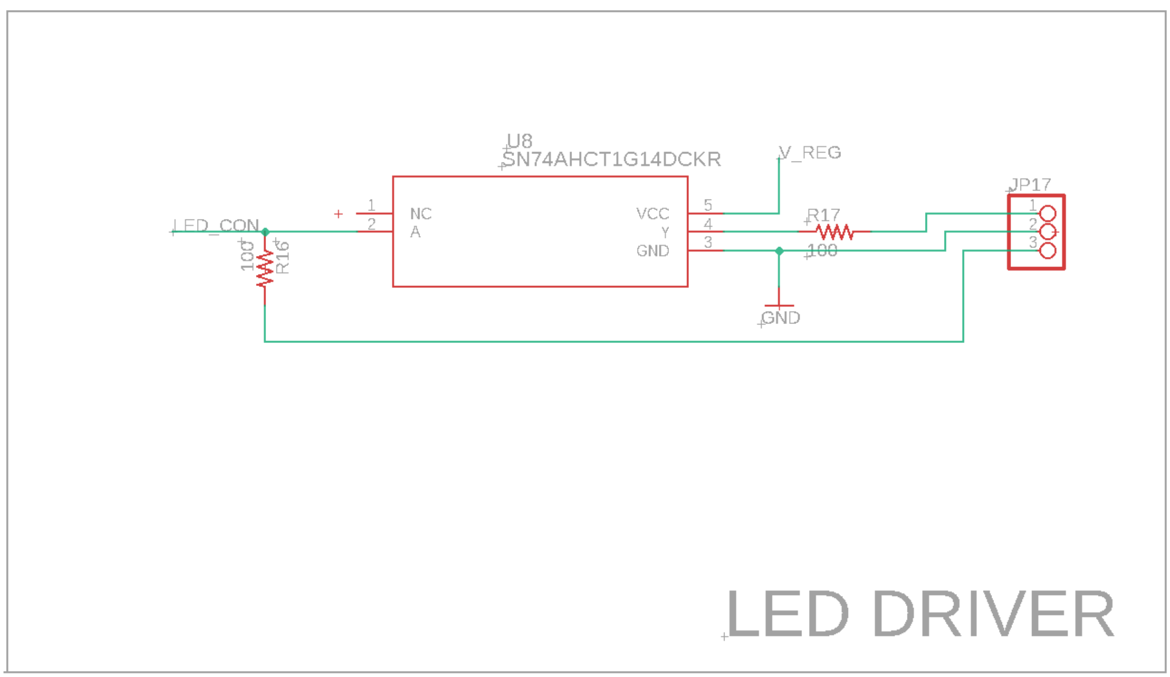

Figure 11: LED Driver/Inverter Schematic

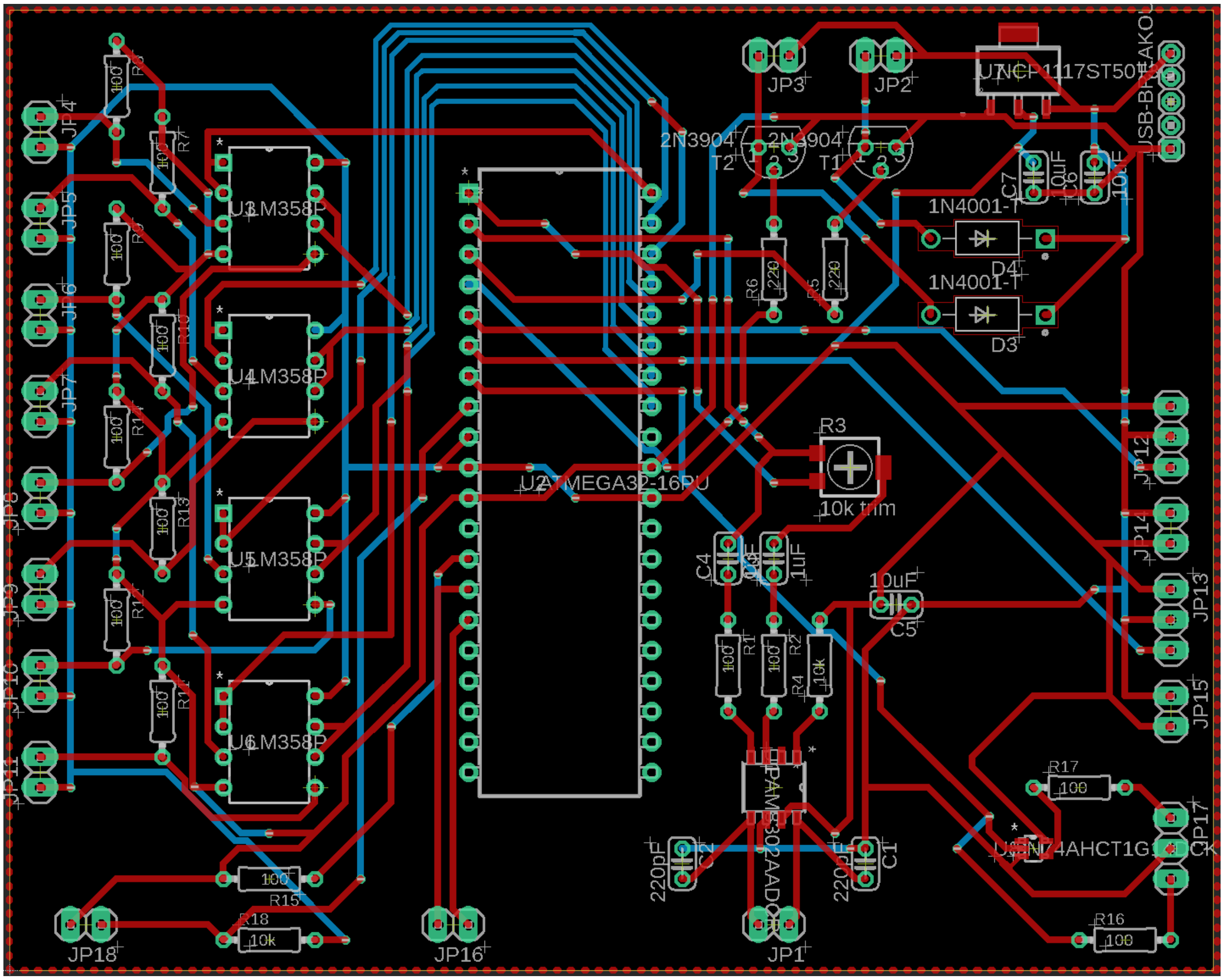

Figure 12: PCB Layout

2.8 Tolerance & Risk Analysis

One important tolerance that we must maintain is the current draw of our system. Since we have solenoids to act as our electronic locking system, and they don't have a separate power supply, we must be keenly aware of the current draw of our system. Each solenoid has a current draw of 1A, while the MCU has a rated current of 200mA. This means that if both solenoids were to be in the active state at the same time, then power would be lost to the MCU, causing the entire system to reset. This doesn't take into account the other power hungry devices of our circuit either. As each sensor, and the amplifier, will require some amount of additional power/current to run properly. To counteract this, we will fire the solenoids with the intention of all other components of the circuit resetting. This means that there must be a sequence of events, where this reset of sensors and other ICU's will not affect the overall integrity of the system. As the firing of the solenoid is the final step at the end of each of our sub-systems, this should be relatively simple. However we must make sure that at all costs the MCU is never reset, as this could cause issues with people being left locking within the “airlock” of our system, requiring usage of our fail-safe switch to safely exit.

We will be using a wallwart 5V 2A AC/DC Converter to power our system, meaning the maximum power that can be delivered is 10W of power. This can be seen using the power equation:

$$P = V x I$$

With this we can determine whether the current system can be supported with this maximum power supply. The QR Code Scanner is being powered separately using a raspberry pi, and will not be included in this calculation.

Component	Max Rated Voltage	Max Rated Current	Rated Power
ATMEGA32 MCU	4.5-5.5V	20mA	0.11W
Break-beam Sensor	5V	20mA	0.11W
Solenoid	5V	1A	5W
LED	5V	20mA	0.11W
Op-Amp x8	5V	40mA (5mA each)	0.2W
Inverter	5V	10uA	.00005W
Total Power		5.53W	

This shows that the device is easily powered by the given power supply, given that only one solenoid is drawing current at a time. This will be dealt with on the software side as specified above.

Another Tolerance that we will have to analyze closely is the bandwidth of our system. Currently, we plan on having a 30 second processing time, per person entering the system. This means that from the time of scanning the QR code, the user should be able to enter the “airlock”, the system should process the number of people within the space, and a decision should be made whether to let that person in or not within that 30 second window. To make this happen we plan on going forward with a pipeline style approach. Once a user has scanned, and been let into the inner space, the system will then prime the scanner to scan the next potential user, and have them queued up for entry into the “airlock” following the processing of the initial user. This should allow for a quicker processing time, as the next user will already be ready to enter the space, while the initial user is being processed. To further expand upon this 30 second requirement, we should take a loot at the sensors involved, and their processing speed. As the two main sensors within the device are a break-beam sensor and a pressure pad sensor, the speed at which these devices can update the control unit with information is very important to meeting this requirement.

Component	Response Time
Break Beam Sensor	2ms
QR Code Scanner	9600 Baud
MCU Internal Clock:	8MHz
MCU Optional External Crystal:	16MHz
Pressure Plate FSR Array (Op-Amps)	1 us
Estimate of Entry Time	~6 Seconds

Our QR code sensor built out of a raspberry pi will operate at a baud rate of 9600. Meaning that the device can deliver 9600 bits to the MCU per second.

Since a majority of the time spent will be the user moving into the space, and standing in the correct position on the pressure plate. The response time of the system needs to be relatively quick. However with the given data, the entire detection system should be able to process the user within a second, meaning that the 30 second time frame that we have set for ourselves should be achievable. Even without a pipeline style scanning system.

The Break-Beam sensor needs to detect the numbers of people coming across it, so it needs to be set at a proper threshold. We need to use our devices to test enough times to get the expected time of one person walking across the sensor. For example, if the threshold is set to a very small number, when one person walks across the sensor, the system may detect both of the legs as two people then send out a false alarm and slow down the whole system.

The other risk is how to successfully and correctly detect the people entering the first door by the pressure detection sensor. When a person with luggage enters the building, we have to detect "one person with a luggage" and "two people", or it will be a false alarm. Also, the miss is a more critical risk. If two people enter the building together and the system fails to detect that, it will increase the risk of spreading the COVID-19.

3 Safety and Ethics

According to ACM's ethical guidelines [3], we should use personal data legally and collect only the necessary personal information. Given that this project cannot use data from safer Illinois, this project will simulate a two-factor authentication system. Therefore, this project will not collect data from users and will not use biometric information. If this project does go into realization, the system will not collect biometric information from students and will only use publicly available identifying information, such as students' name, and their NetId. And they will not be made public by this system.

To avoid the system adding any other new safety problems and concerns to the campus society, and according to IEEE Code of Ethics [9], the system is hand-free and adds no steps to the original people-checking process. The system just uses a QR code scanner to avoid the possibility of human processing errors as well as not decreasing the efficiency of entering the building. Also, since there can only be one person in the people detecting area, it will not violate the Social Distancing regulations in Illinois and UIUC campus [8].

Potential security issues for this project may include the use of electricity. This project may use 120v and convert it to low voltage for use. Therefore, the risk of electrical shock to humans is extremely low.

This system will use lithium batteries as a backup power source. According to Occupational Safety and Health Administration [4], lithium batteries can be damaged by physical impacts including shocks, punctures and crushes, posing a risk of explosion. In addition, high environmental temperatures can shorten battery life. Therefore, this project will protect the lithium battery from physical shocks and ensure that the lithium battery works in a suitable environment. The batteries will be checked regularly and replaced with new batteries when needed. There will be routinely checks to ensure that there are no short circuits in the system and that circuits will only be operated and modified when they are not charged.

Since the system involves a closed space that we are nicknaming the "airlock", there is a chance of a malfunction that could result in a user being locked within this environment. The main reason this could occur is that both Solenoids fired at the same time, resulting in a system reset, leaving the user within. The workaround for this issue will come in the form of a fail switch within the "airlock" that will be hardwired into the control circuit of the entrance solenoid, allowing the door to be unlocked, and letting the user leave the space.

Some of the devices in this project will work outdoors, such as the QR code scanner. Given that it is difficult to guarantee a dry environment outdoors all the time, all devices outdoors must meet the IP55 [7] standard to guarantee safety.

Another safety problem is “hijacking” the system. Since People who haven’t got a negative COVID-19 test result recently cannot generate a QR code with permission of entering the building, they may force or “borrow” the QR code generated by another person with the negative test result. This problem can’t be solved since we are lacking a biometrics ID system, but adding a webpage for “reporting” may partially solve the problem. The University or the other administrations may use the information reported to start an investigation, and it can reduce the possibilities of people “hijacking” the system.

4 Costs

Our fixed development costs are estimated to be $35/hour, 10 hours/week based on the average salary of entry level ECE graduates [6]. Operating on a 16 week semester results in a total cost of:

$$2.5 * \frac{\$35}{hr} * \frac{10hr}{wk} * 16wks \ = \$14,000 \text{ per person}$$

A breakdown of our components can be seen below, and the total cost of our prototype comes out to be roughly $128.48. This is with an estimated cost of smaller components such as resistors/capacitors and small IC’s.

Part	Cost
1 Channel Amp (PAM8302AADCR)	$0.65
2x Break-Beam (1528-2526-ND)	$13.00
MCU (ATMEGA32-16PU)	$6.61
Wall Adapter (5VDC, 2A)	$5.95
MicroB USB Breakout	$2.75
BiColor LED x2 (LTL1BEKVJNN)	$0.74
Raspberry Pi + Camera Module	$72.21
Assorted Resistors, Capacitors, ICs, Sockets (Digikey; est.)	$7.00
Floor Mat Switch (KIT-AC-700-30-926B)	$19.95
Solenoid Actuators x2 (adafruit 2776)	$9.90

Total	$138.76

Table 1: Cost Analysis

We are also considering the construction of our own pressure sensors using aluminum foil and cardboard, as mass market options are quite expensive. This would also allow us to have an array of them, so the detection of multiple pressure points would be an attainable goal. This would reduce the cost by about $15.00.

5 Schedule

Week	Patrick	Zewen	Yijie
3/1	Hardware search, and schematic creation. Cost Analysis and Schedule Creation. MCU Research	RV Table Creation and QR Code Sensor Development.	Hardware Search.Server Database research. QR Code Sensor Development
3/8	Prepare for Design Review/Finalize Subsystem #1 Schematic/Begin Subsystem #2 Schematic	Create QR code scanner from raspberry PI	Research into UART data transmission for encoded QR code information
3/15	Finalize Subsystem #2 Schematic/Begin PCB Design. Construction of Pressure Plates.	Start research into MCU programming, and I/O port usage.	Create backend for subsystem #1 though Server creation on PI.
3/22	If first round PCB orders were not met, revise issues and make sure round 2 is met. Begin Testing subsystem #1 for edge cases/ Meeting Verifications	Continue research on MCU programming, and implementing the final code base onto the chip.	Extensive range testing on Scanner and Break-beam sensors, evaluating the accuracy of our sensors.
3/29	Sanity Check, are we meeting deadlines, and if we are behind, where do we need to catch up?	Sanity Check, are we meeting deadlines, and if we are behind, where do we need to catch up?	Sanity Check, are we meeting deadlines, and if we are behind, where do we need to catch up?
4/5	Begin construction of our proof of concept work space (Small doors,	Begin working on 3D printed Case	Begin Testing of Subsystem #2 for edge cases of people counting/Meeting

	detection field.)		Verifications
4/12	Conduct environmental testing on Subsystem #2	Conduct environmental testing on the QR Code Scanner + Case	Conduct environmental testing on QR Code Scanner + Case
4/19	Mock Demo Preparation + Final Checklist of R/V Tables	Mock Demo Preparation + Final Checklist of R/V Tables	Mock Demo Preparation + Final Checklist of R/V Tables
4/26	Prepare Final Presentation/Paper	Prepare Final Presentation/Paper	Prepare Final Presentation/Paper

Table 2: Team Scheduling

Orange = Sub-System 1 (QR Code Scanner/Initial Detection)

Green = Sub-System 2 (People Counting Solution)

References

[1] The Lancet, ‘Physical distancing, face masks, and eye protection to prevent person-to-person transmission of SARS-CoV-2 and COVID-19: a systematic review and meta-analysis’, 2020. [Online]. Available: https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(20)31142-9/fulltext. [Accessed: 15-Feb-2021]

[2] The Pulse, ‘Human Factors, Human Error and the role of bad luck in Incident Investigations’, 2016. [Online]. Available: https://www.linkedin.com/pulse/human-factors-error-role-bad-luck-incident-graham-edkins/ [Accessed: 18-Feb-2021]

[3] Association for Computing Machinery, “ACM Code of Ethics and Professional Conduct,” Association for Computing Machinery, 2018[Online]. Available: https://www.acm.org/code-of-ethics [Accessed: Fed. 16, 2021]

[4] United States Department of Labor, “Preventing Fire and/or Explosion Injury from Small and Wearable Lithium Battery Powered Devices” Occupational Safety and Health Administration, 2019[Online]. https://www.osha.gov/dts/shib/shib011819.html [Accessed: Feb. 17, 2021]

[5] TT Electronics ‘Optical Emitter and Sensor Pair OPB100Z’, 2016. [Online]. Available: https://www.ttelectronics.com/TTElectronics/media/ProductFiles/Optoelectronics/Datasheets/OPB100.pdf
[Accessed: 27-Feb-2021]

[6] Salary.com ‘Hourly Wage for Electrical Engineer I in the United States’, 2021. [Online] Available: https://www.salary.com/research/salary/benchmark/electrical-engineer-i-hourly-wages
[Accessed: 1-March-2021]

[7] sourceiex.com ‘Degrees of Protection’, 2021. [Online] Available: https://www.sourceiex.com/Catalogs/IP%20Degress%20Testing%20Details.pdf
[Accessed: 1-April-2021]

[8] University of Illinois Covid-19 ‘Basic Person Actions’, 2021. [Online] Available: https://covid19.illinois.edu/health-and-support/basic-personal-actions
[Accessed: 1-April-2021]

[9] IEEE ‘IEEE Code of Ethics’, 2020. [Online] Available: https://www.ieee.org/about/corporate/governance/p7-8.html
[Accessed: 1-April-2021]