ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

Team #83: Automatic Door Conversion Kits

Romeo Delgado, Love Patel, Alexander Vega

TA: Chi Zhang

Professor: Viktor Gruev

1. Introduction

1.1 Problem

With accessibility being considered more in modern infrastructure, more and more systems for accommodating people with physical disabilities are being installed every day. Most of these systems are installed in public locations and are paid for by the government. This makes it so that the general public never has to think about how the systems got there and how they were paid for. However, for those with physical limitations, installing similar accessibility systems in one's home is extremely costly and difficult. Even doors, something used many times each day by a single individual, will cost someone hundreds of dollars to make automatic. Cheaper and easier to install automatic doors meant for residential homes would alleviate this cost barrier and difficulty of installation for those that struggle to use standard, manual doors.

1.2 Solution

Our solution for the high barrier to entry for making one's home accessible is to make cost effective and quick to install automatic door conversion kits for interior, residential doors. These kits would include an actuating latch, a door opener, and a remote control. The door opener can be screwed onto both the door frame and the door, while the actuating latch would be able to replace standard door latch. The door opener would be able to be remotely activated with a Bluetooth signal from the remote control. The door opener would have a box containing the electronics that run along the top of the door frame with a motor controlling an arm that would swing the door open. To allow the door to close/open remotely without getting caught on the latch, the actuating latch that would close/open in sync with the door opener to allow the door to swing freely. Lastly, the Bluetooth remote would be able to pair to multiple door openers and have the capability to cycle between them. This would allow individuals to buy multiple kits and attach them to doors throughout their house as needed.

Since performing this demonstration on an actual door will prove to be difficult, we will scale down our design to work on a modeled door. The model version will have the same remote and door opener design. However, instead of modeling and installing a door handle, the electronics for the latch system will be placed on the door frame in another component box near where the handle would be on the door. The actuator would then simply protrude out in front of the door to represent the door being in the closed state, and then the actuator would retract the latch when the door is set to open.

1.3 Visual Aids

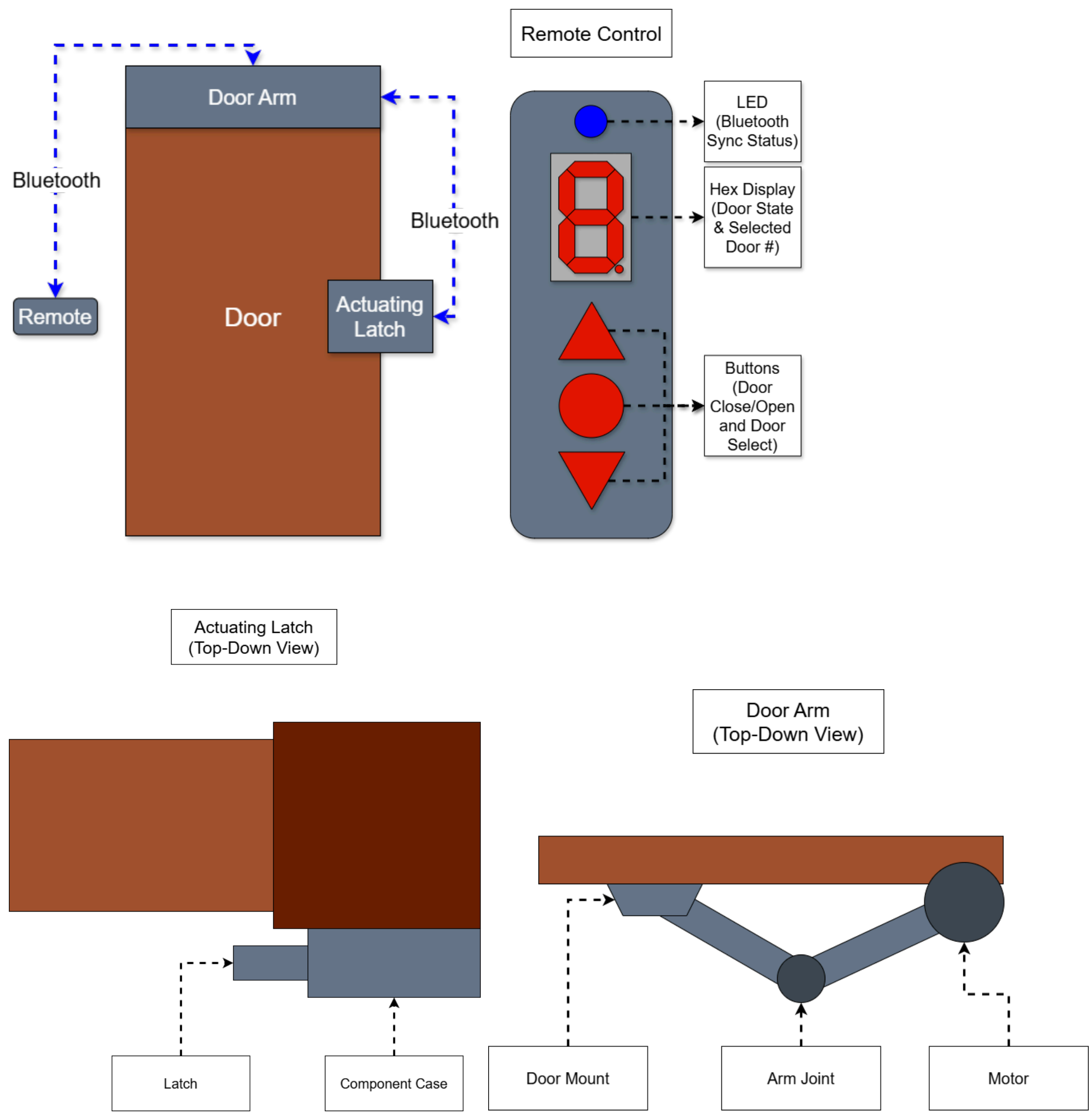

Figure 1. Automatic Door Conversion System Visual Diagram

1.4 High-level Requirements List

- For demonstrational purposes, the singular conversion kit will be placed on the door that should be able to fully swing open the 21 inch tall, 13 inch wide, and 5 lb door. The door should be able to fully swing close/open in approximately 3 seconds. A secondary kit will also be made that should still be able to fully extend or retract the arm in approximately 3 seconds. This timing will be timed from when the door first begins opening up until the door fully opens to 90 degrees. We are aiming to have less than a 5% margin of error with the timing of the door. Having a timing too far under 3 seconds can result in potential injuries from the door swinging unnecessarily fast, and a timing much larger above 3 seconds can be too slow for reasonable use.
- Each of these kits should include a remote that will be able to close/open a door after connecting to it via BlueTooth. Since the system does not need to be extremely responsive, the solenoid latch should begin to contract/extend within ½ second from when the button is pressed. When the latch system is given a bluetooth signal to open, the actuated latch must retract within 1.5 seconds to allow the door to start opening. The latch system can then extend once again after 2 seconds.
- The BlueTooth remote is able to pair with at least 2 automatic door conversion kits, as well as display and switch between which door is being targeted to close/open. The remote will be able to send close/open signals to the targeted door while also displaying each door's closed/open state. The BlueTooth connection status between the remote and door systems should be properly displayed via a blue LED. This includes blinking when searching and a solid color when a connection is found. All interfaces must be fairly responsive, which leads us to the fact that all LEDs should be properly displaying the correct information within ½ second.

2. Design

2.1 Block Diagram

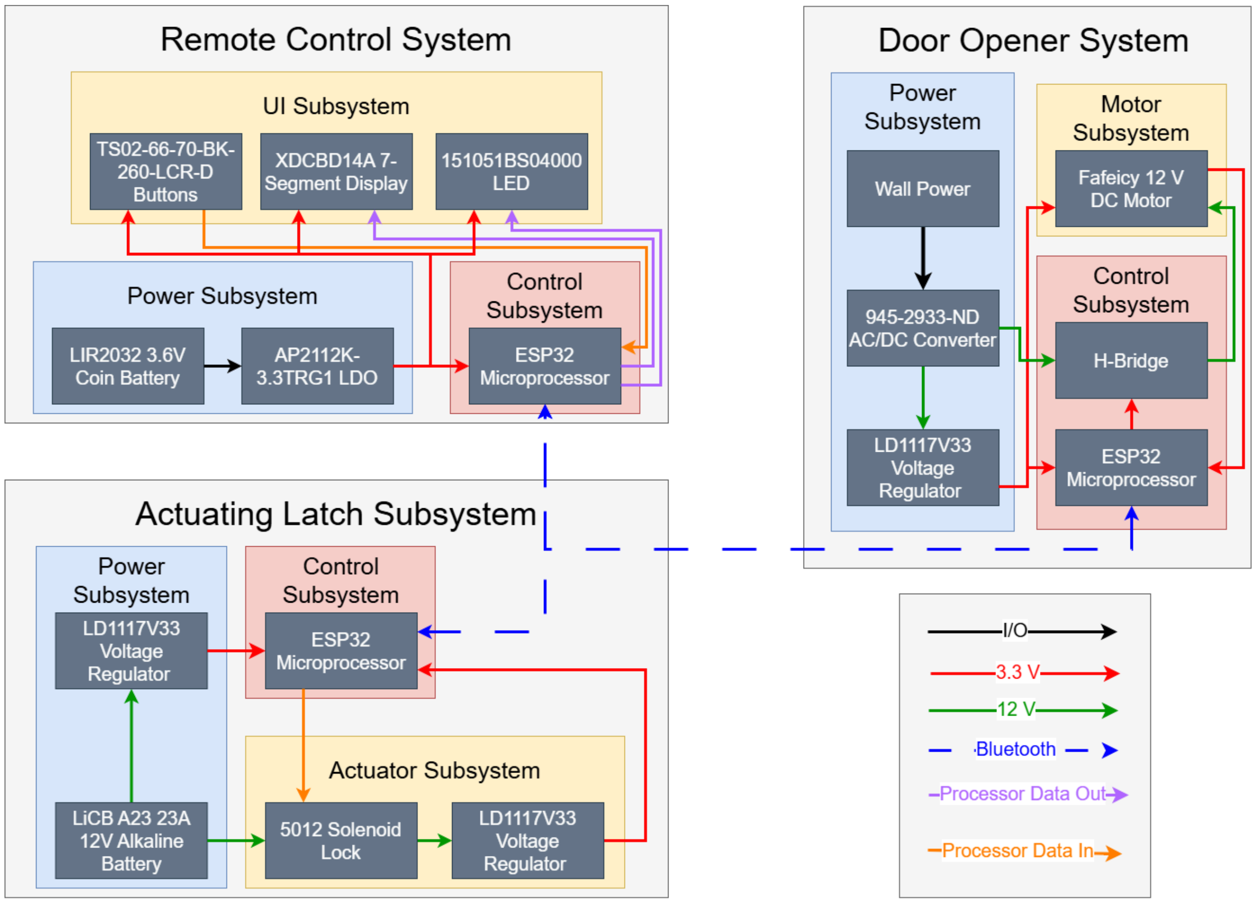

Figure 2. Automatic Door Conversion Kit System Block Diagram

2.2 Physical Design

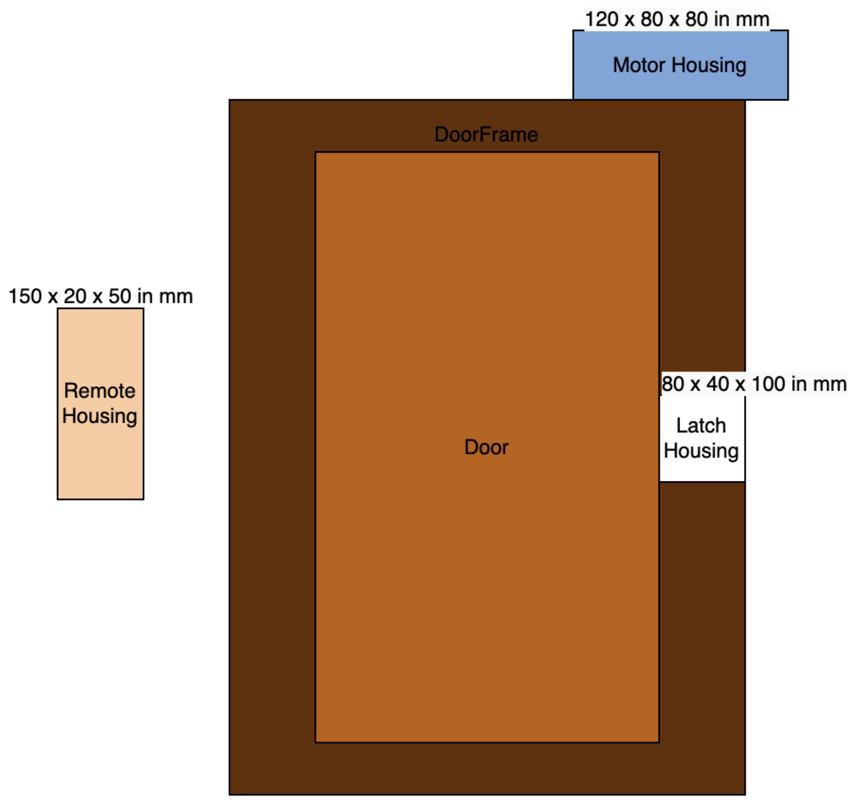

Figure 3. Physical dimensions of the housings of our systems in respect to the entire layout

The model door we will demo with is 21 inches tall, 13 inches wide, and weighs 5 pounds. The housing for the door opener will be mounted on the top of the door frame and will be 120 x 80 x 80 millimeters. The housing for the actuating latch will be mounted to the door frame level with where the door knob would be and will be 80 x 40 x 100 millimeters. Lastly, the housing for the remote will be 150 x 20 x 50 millimeters.

2.3 Actuating Latch Subsystem

The actuating latch subsystem is responsible for replacing the door's latch to allow the door to close/open upon receiving a bluetooth signal from the door opener. The actuator should remain engaged and only disengage upon receiving an open signal to allow the door to swing open with the help of the door opener without getting caught on the door's latch. The actuator's rod should be shaped as to allow the door to close without the actuator needing to disengage then reengage upon making contact with the door frame. The actuator should also send a bluetooth signal back to the door opener to confirm that it disengaged/engaged.

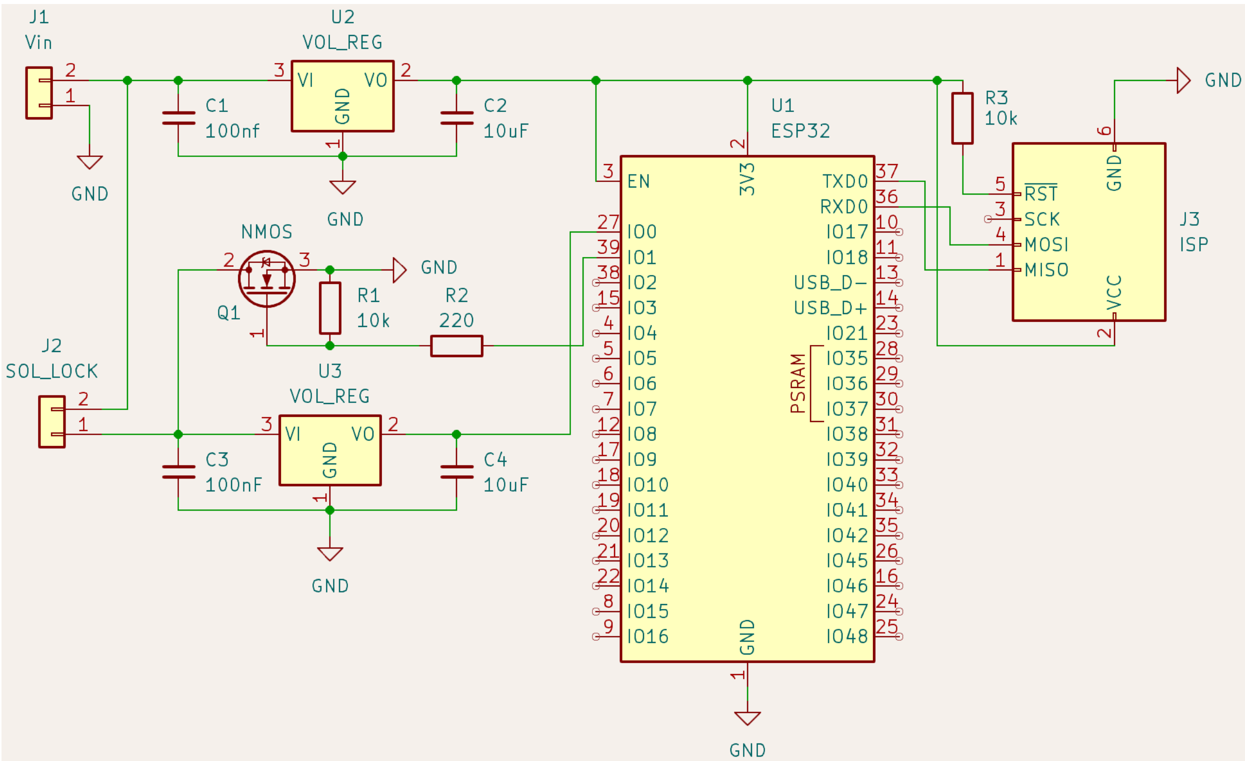

Figure 3: Actuating latch subsystem

The above schematic implements the hardware of our actuating latch subsystem. We have the Vin connector which will be the 12 V battery supply that will pass through our voltage regulator to output 3.3 V so that we have power the ESP32. The 12 V of power will also go to power the Solenoid Lock which is shown as a connector on our schematic. The output from the connector will go through a voltage regulator as well to output 3.3 V of a signal to let the ESP32 know when the latch is engaged. The ESP32 also sends out a signal from one of the GPIO pins to a NMOS gate to ground the solenoid lock which is the signal to indicate the latch to disengage. To program the ESP32, we have an ISP that will connect to its TX and RX pins. The RST of the ISP will connect to the power but through a 10k ohm resistor to make sure of no floating voltages. We also have a few resistors in the schematic to ensure there is no floating voltage. We also have capacitors at the ends of the voltage regulators which act as decoupling capacitors.

Table 1: Actuating Latch Subsystem - Requirements & Verification

Requirements	**Verification**
Transceiver receives a signal from the door subsystem and allows the latch to begin opening within ¼ second.	We will use a test script within the door latch microcontroller. This program will be able to record the milliseconds from which the signal was initially received from the door arm. The latch sends back confirmation immediately after beginning to open, so we know how long processing the signal and beginning to open the latch took.
Solenoid latch receives 12V +/- 5% across when it is activated.	We can use the oscilloscope with the negative end always on ground. We can first check, when activated, that the positive end of the battery is still outputting 12V as that's the input of the solenoid latch which is clearly exposed. Secondly, we can check the drain of the NMOS to ensure that it is grounded (0V). This will ensure that the latch is receiving the proper voltage.

2.4 Control Subsystem

The control subsystem is responsible for all of the Bluetooth signals sent between our three PCBs as well as all of the control logic for signals sent between the actuating latch, the door opener, and the remote control. The control logic will include the currently targeted door's number and closed/open state, using the RPM signal from the motor to detect jams, and the timing of the close/open signals for the actuating latch and door opener. When the remote control sends a close/open signal, the control subsystem will send the signal to the actuating latch and the door opener to open the door, and when the remote switches between which door is targeted, the control subsystem will change which latch and opener will be closed/opened with an activation signal.

Table 2: Actuating Latch Subsystem - Requirements & Verification

Requirements	**Verification**
Each microprocessor should be able to communicate with the rest of the system within 100 milliseconds of receiving and sending out signals.	Use real-time hardware testing by creating test scripts that will timestamp received and transmitted signals. Set up two of the ESP32s as communication partners to send signals between the two.
We can check that all the inputs to the ESP32 are 3.3V +/- 5%.	We can use the oscilloscope with the negative end on ground. We can then place the positive end on the outputs of each of the regulators. Ensuring that all these outputs are 3.3V will allow for safe use of the ESP32.
The correct data must be transferred to the ESP32 when programming each of the devices.	Firstly to ensure any data transmission, we can use an oscilloscope at each of the RX and TX pins of the ESP32 as we send data through UART. Secondly, proper data checks can be performed through software checks, which involve sending data through GPIO pins and then using another oscilloscope check at that said pin.
Similar to checking BlueTooth transmission signals, we must ensure that logic duration is fast and no longer than 50 milliseconds.	We can use a test script that checks the duration runtime of outputs for GPIO pins or BlueTooth signals. We can then display the runtime duration, so that we can make further optimizations if needed.

2.5 Door Opener Subsystem

The door opener subsystem is responsible for closing/opening the door upon receiving a close/open signal from the remote control. When receiving an open signal, the door opener will send a signal to the actuated latch to disengage before signaling its motor to engage, swinging open the door with a jointed arm. After fully opening with the expected RPM from the motor for the duration of the opening cycle, the door opener will consider the door as having opened without obstruction and change the door's state to open, and vice versa for closing the door.

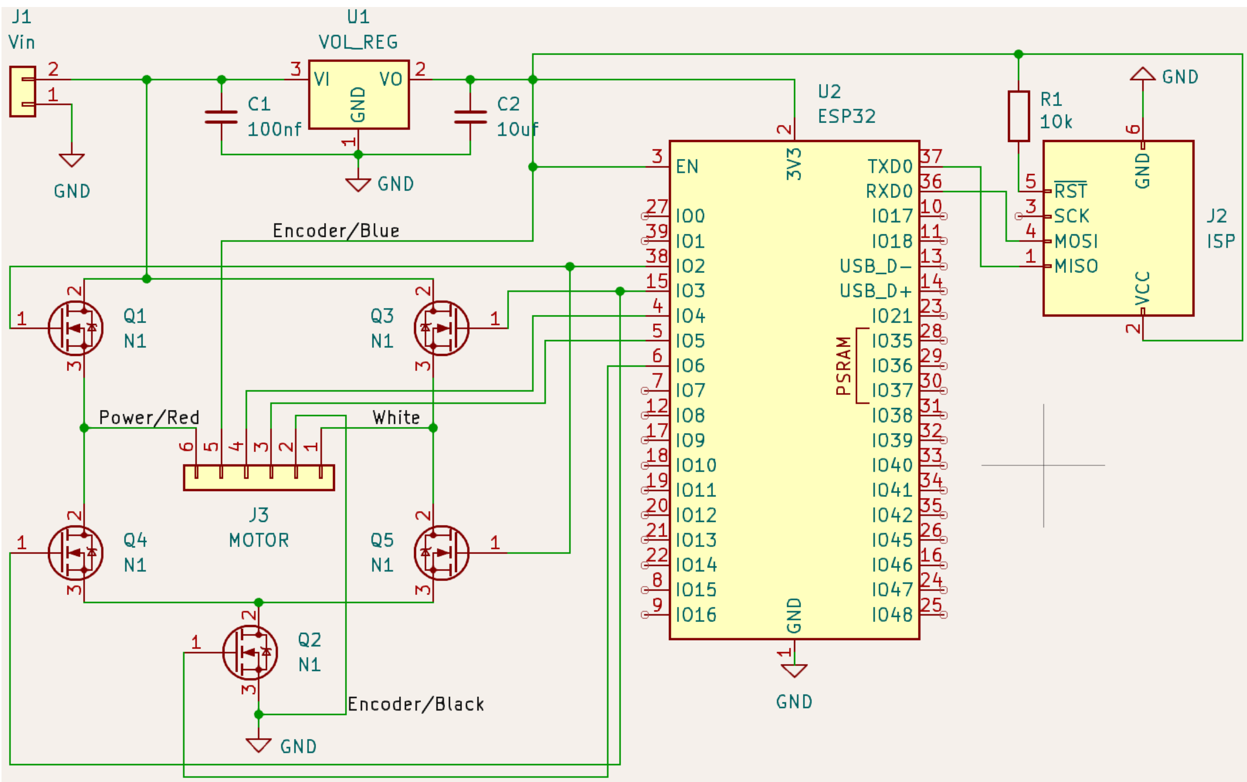

Figure 4: Door opener subsystem

The above schematic implements the door opener subsystem. The NMOSs in the schematic are used for the H-bridge motor driver and diagonal gates receive the same signal from the ESP32 in order to either drive the motor forward or in the reverse direction. The voltage regulator and vin connectors are explained in the power subsystem. The extra NMOS at the bottom of the schematic is to stop the motor and its gate also receives a signal from the ESP32 GPIO pins. To program the ESP32, we have an ISP that will connect to its TX and RX pins. The RST of the ISP will connect to the power but through a 10k ohm resistor to make sure of no floating voltages. We also have a few resistors in the schematic to ensure there is no floating voltage. We also have capacitors at the ends of the voltage regulators which act as decoupling capacitors.

Table 3: Door Opener Subsystem - Requirements & Verification

Requirements	**Verification**
Ensure that modeled door opens up to a full 90 degrees in 3 +/- 0.15 seconds	Use a protractor to measure that the door has opened a full 90 degrees by placing it at the bottom of the door, perpendicular to the door frame. Write test scripts to obtain the timestamps of when the motor started spinning to when it stopped spinning to determine the door has opened in 3 seconds.
Significant decrease in rotational speed (<9 RPM) results in door opening	This will utilize the encoder signal feedback received through the GPIO pin. The rotational speed will consistently be monitored through software, and will force the door to open under the condition that the rotational speed significantly decreases.
We must check that the voltage across the motor is 12V +/- 5%.	For the NMOS's sources entering the motor, we will place the positive end of the oscilloscope. We can place the negative end of the oscilloscope on the drains of NMOS connected to the motor. The difference must be 12V depending on the rotation of the motor. Additionally this must hold true for

	NMOS diagonally across from one another so that the motor is properly rotating clockwise or counterclockwise.
From receiving an input signal from another subsystem, the door subsystem should be able to transmit another BlueTooth signal within ¼ second.	This can be checked using test scripts, which will measure from the time receiving a signal to the time it takes to output another signal via BlueTooth.
The 12V DC gear motor should be able to produce 0.3 Nm of torque and 0.3142 W of power.	Use a multimeter to measure the voltage and current across the motor to calculate the power using the power equation, P=V*I. Use a dynamometer to measure the required torque of 0.3 Nm.

2.6 Power Subsystem

The power subsystem is responsible for powering the actuating latch, the door opener, and the remote control. For the actuating latch, power will run from a 12V battery through a voltage regulator to stabilize the 12V before running the power to the actuator and through a different voltage regulator to step down to 3.3V for the PCB. For the door opener, the power will run from the wall through an 12V AC/DC converter, then through a voltage regulator into the motor to stabilize the 12V, and also through a different voltage regulator to step down to 3.3V for the PCB. Lastly, for the remote control, 3.6V from the coin battery will run through a voltage regulator for the buttons, hex display, LED, and PCB.

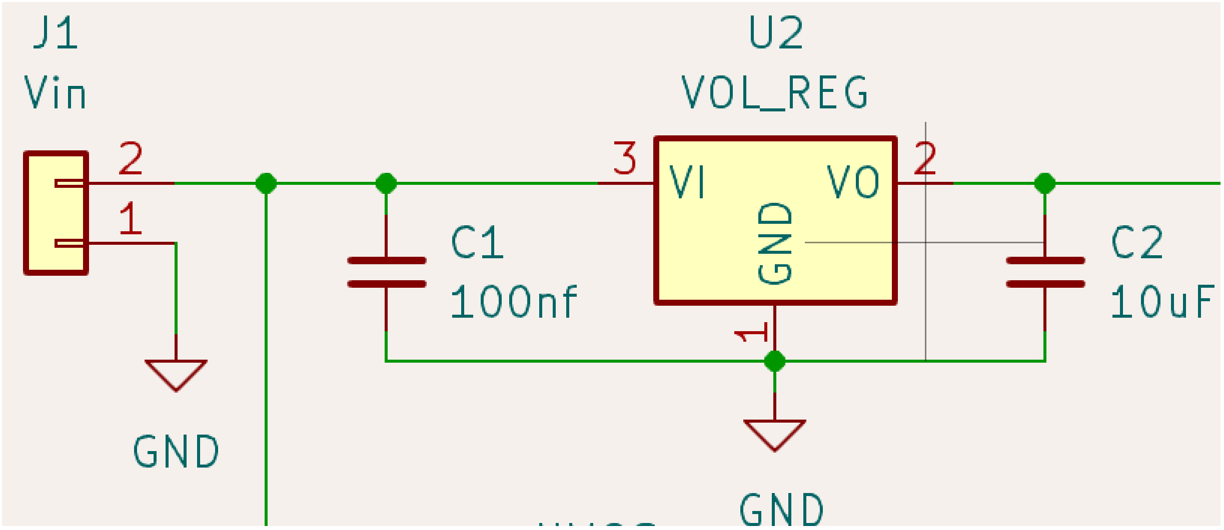

Figure 5: Actuating latch power subsystem

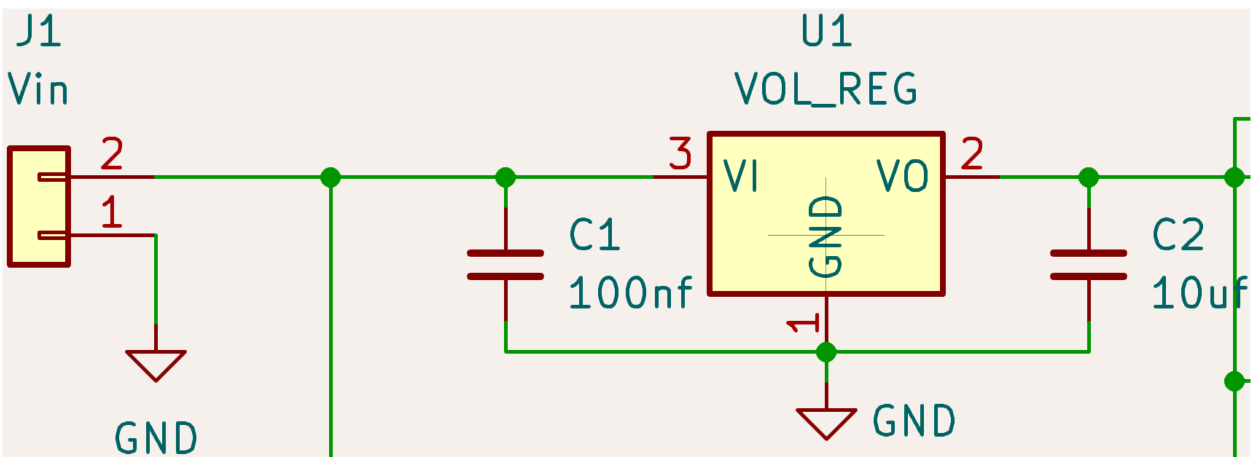

Figure 6: Door Opener power subsystem

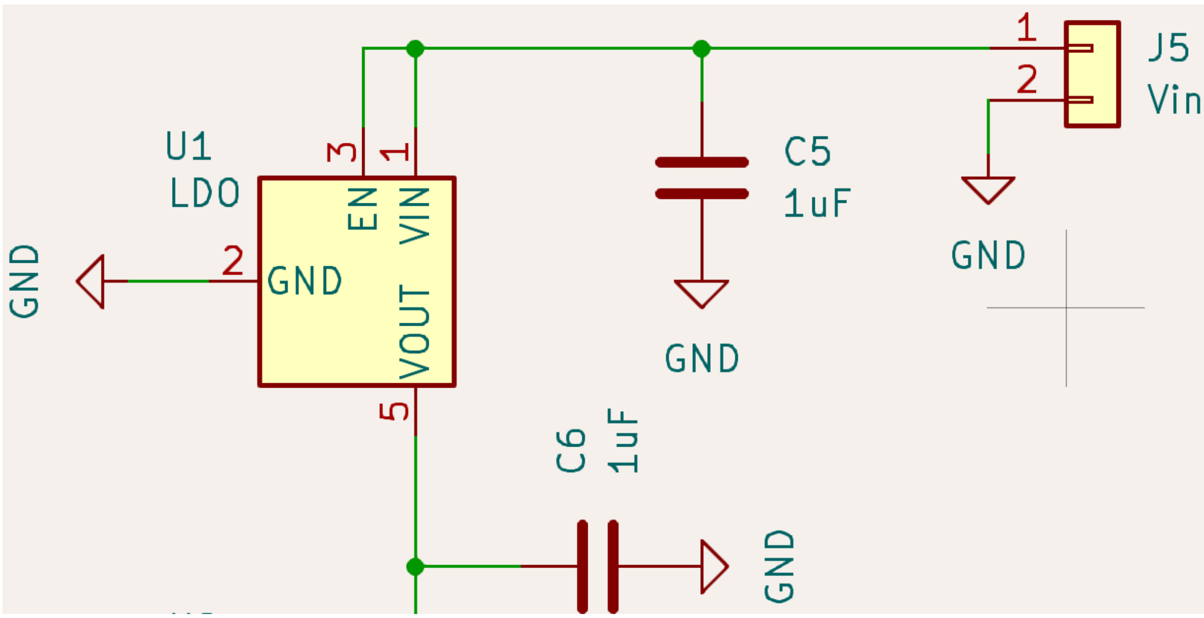

Figure 7: Remote control power subsystem

Since our design is using three different power sources we have 3 different power subsystems when it comes to the hardware aspect of the project. Figure 5 describes the actuating latch power subsystem that has a Vin connector for the 12 V power supply which goes to a voltage regulator that steps down the voltage to 3.3 V for the microcontroller. There are decoupling capacitors at the ends of the voltage regulator. Figure 6 describes the remote control power subsystem that has a connector for the 3.6 V coin battery and that goes through an LDO to drop the voltage to 3.3 V. That voltage goes to power the ESP32 and there are also decoupling resistors at the ends of the LDO. Figure 7 describes that door opener power subsystem that has a connector for the 12 V power supply that also goes to a voltage regulator to be stepped down to 3.3 V to power the ESP32. Again there are capacitors at the end of the voltage regulator to act as decoupling capacitors.

Table 4: Power Subsystem - Requirements & Verification

Requirements	**Verification**
One 12V battery should be able to produce 12	Use a multimeter to measure the output of the

+/- 0.3V in order to power the solenoid actuator.	battery. Connect the negative terminal of the multimeter to ground and the positive end of the terminal to the positive end of the battery. The difference between these values should be 12V.
Wall power should be stepped down to 12V to power the 12V DC gear motor. This should also be stepped down to 3.3V by the regulator to power the microcontroller.	Use a multimeter to measure that once the wall power has been stepped down, the voltage regulator is getting 12V inputted. Use the multimeter again to see that the voltage regulator is outputting 3.3 V for the microcontroller. The negative end of the multimeter should be connected to ground, while the positive end will be used to probe the regulator pins.
LIR2032 should be used to power the remote control by outputting 3.6 +/- 0.3V.	Use a multimeter to measure that the voltage coming out of the coin battery is reading 3.6 V. Connect the negative end of the multimeter to ground and the positive end to the positive end of the battery.
Every module in each of the subsystems should be powered by 3.3 +/- 0.3V except the motor and actuator.	Using a multimeter, go around the entire PCB to ensure that all parts receive 3.3 V across them except for the ones specified above for voltages above 3.3 V. Put the negative end of the terminal to ground and the use the positive end of the terminal to record the input voltages.

2.7 Remote Control Subsystem

The remote control subsystem is responsible for allowing the user to control the closing/opening of each automatic door conversion kit using input from the user to three buttons on the remote. Two of these buttons will control which door is being targeted, and the third will be used to sync the remote as well as send the close/open signal to the door opener. The remote control must also be able to receive a bluetooth signal from each door opener. While on and paired to a door opener, the remote should have a lit LED and should display the number of the currently targeted door as well as its closed/open state on a 7-segment display.

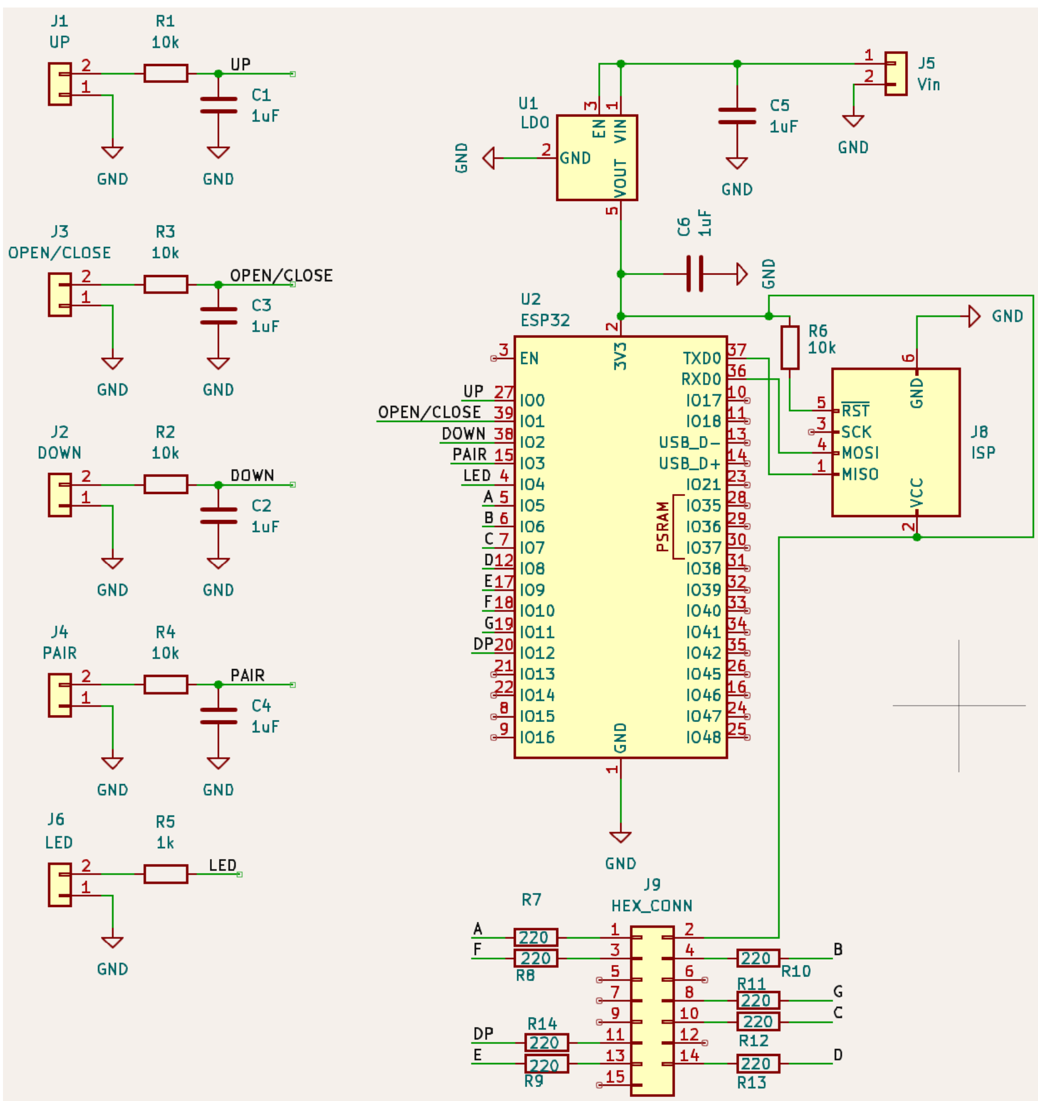

Figure 8: Remote control subsystem

The above schematic implements the hardware of the remote control subsystem. On the left hand side, we can see that there are four buttons that are labeled according to what they do. They send signals to the ESP32 to indicate the press of the button and have it perform the proper job that it needs to do. There are resistors and capacitors on them to reduce debouncing of the buttons. The voltage has been described in the power subsystem. The hex connector is used for the hex display and has been labeled according to the one we will be using in the project. Each segment has a signal being sent from the ESP32 to turn on or off. It is also powered by 3.3 V, the same as the microcontroller. The signals also go through 220 ohm resistors to ensure proper voltages are received. To program the ESP32, we have an ISP that will connect to its TX and RX pins. The RST of the ISP will connect to the power but through a 10k ohm resistor to make sure of no floating voltages. We also have a few resistors in the schematic to ensure there is no floating voltage. We also have capacitors at the ends of the voltage regulators which act as decoupling capacitors.

Table 5: Remote Control Subsystem - Requirements & Verification

Requirements	**Verification**
A led should be able to properly indicate whether a door is open or close within 100 milliseconds of receiving the Bluetooth signal from the Door Opener Subsystem.	We will be using test scripts to record the time it takes from receiving an input signal from the door to the time it takes for the GPIO to output the correct signals to turn on the hex display LEDs.
Another LED should be functioning properly to signal if the pairing of the device was successful or not also within 100 milliseconds of receiving the Bluetooth signal from the Door Opener Subsystem. Additionally, from pressing the pairing button, the LED should begin flashing within 100 milliseconds.	We will be using test scripts to record the time it takes from receiving an input signal from the door to the time it takes for the GPIO to output the correct signals to turn on the blue LED. For testing the pairing, the same script can be used with the same timing.

For high responsiveness, the display should display the correct door number within 150 milliseconds of pushing the cycle buttons.	We will be using test scripts to record the time it takes from receiving an input signal from the remote control to the time it takes for the GPIO to output the correct signals to indicate which door number is shown.

2.8 Tolerance Analysis

2.8.1 Current Analysis

Firstly, with the remote control subsystem, the Li-ion coin battery has 3500 mAh of battery capacity. The regulators can take and output 800 mA.

The energy equation is the following:

$$E = I\,x\,t$$

I = Current output

t = Duration

Therefore, we can solve for the total duration that the battery should be able to supply a proper current and voltage to a regulator.

$$t = 4.375 \text{ hours}$$

This is sufficient for our purposes as we will not continually use the remote subsystem and the testing will be done in bursts. Additionally, this is assuming maximum current draw from the regulators. This will not be the case most of the time as the ESP32 will draw nearer to 240 mA since we are solely using the BlueTooth and GPIO features on the devices. The ESP32 outputs and inputs 20 mA safely to each of the surrounding devices using the GPIO pins, which is the exact amount required by the LEDs and hex displays. We have only 9 LEDs to power, which is less than the 240 mA for safe use of the ESP32.

$$9\ x\ 20\ mA\ =\ 180\ mA \leq 240\ mA$$

For the door opener subsystem, the 6 A provided by the converter is more than enough amps for the motor and ESP32. The motor will consume a maximum of 600mA while in use and the ESP32 will be well under 240 mA. Once again, the regulator will be able to output a maximum 800 mA, which is well above the desired draw from the ESP32.

Lastly, for the door latch subsystem, we are using a 2600 mAh battery for occasional use for the latch and the ESP32 in the subsystem. The maximum expected draw is approximately 500 mAh and a consistent 240 mAh draw.

t = 5.2 hours

This has enough current to provide for the ESP32 and solenoid latch for occasional usage, since the latch will only be engaged for a few seconds at a time.

2.8.2 Motor Power & Torque Analysis

In this analysis, we will be using the following equations to determine the torque and the power that our motor needs to produce.

The torque equation is the following:

$$\tau\ =\ F\ x\ r$$

F = force need to move the door

r = distance from the hinge where the force is applied

But since this will be rotational force, we will be using moment of inertia and angular acceleration:

$$\tau = I\alpha$$

I = inertia

$$\alpha = angular\ acceleration$$

The Inertia equation is the following for a door:

$$I = \frac{1}{3}mL^2$$

m = mass of door

L = width of door

The angular acceleration equation is the following

$$\theta = w_i t + \frac{1}{2}\alpha t^2 \ -> \ \alpha = \frac{\theta*2}{t^2}$$

$w_i = initial\ angular\ velocity$, 0 since initially at rest

t = time to open door completely to 90 degrees

So,

$$I = \frac{1}{3} * 2.27 * 0.33^2 = 0.082401\, kg * m^2$$

and,

$$\alpha = \frac{\pi/2*2}{3^2} = 0.3491\, rad/s^2$$

So the torque is:

$$\tau = 0.082401 * 0.3491 = 0.0288\, Nm$$

Now to determine RPM, we will find the angular velocity in rad/s and convert it to RPM:

$$w_f = w_i + \alpha t$$

$w_i = initial\ angular\ velocity$, 0 since initially at rest

$\alpha = angular\ acceleration$

t = time to open door completely to 90 degrees

So,

$$w_f = 0 + 0.3491 * 3 = 1.0472\, rad/s$$

$$\frac{1.0472 * 60}{2\pi} = 10.00\, RPM$$

The power equation is the following:.

$$P = \tau\, x\, w$$

$\tau = torque$

w = angular velocity

So,

$$P = 0.3 * 1.047 = 0.3142\, W$$

Therefore our motor should be able to produce 0.3 Nm of torque with 0.3142 W. So the 12V DC gear motor that we are using will be able to produce 10 RPM with the required torque and power.

3. Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor Cost

All three members of our group are Computer Engineering students. Based on the ECE department website regarding average salaries, a post-graduate starting salary from the University of Illinois at Urbana-Champaign is approximately $109,176 and the average work hours per year is about 2,080 hours. This equates to approximately $52.49/hr. We approximately work for 12 hours on this project and estimate that it will take us 10 weeks to complete the entire project.

So the labor cost per person is:

$$\$52.49/hr \ * \ 2.5 * 12 * 10 \ = \ \$15,747/person$$

So the total labor cost comes out to:

$$\$15,747/person \ * \ 3 \ = \ \$47,241 \ of \ total \ labor \ cost$$

3.1.2 Parts Cost

Description	Manufacturer	Part #	Unit Cost	Quantity	Total Cost	Link
Microcontroller	Espressif Systems	ESP32-S3-WROOM-1-N16	$5.92	4	$23.68	Link
12 V DC Gear Motor with Encoder	Fafeicy	Fafeicyogdbg2c9up	$15.09	2	$30.18	Link

		-01				
12 V Solenoid lock	Adafruit Industries LLC	5012	$14.98	1	$14.98	Link
7-Segment Hex Display	Broadcom Limited	5082-7650	$2.97	1	$2.97	Link
Rechargeable LIR2032 Coin Battery	ywhome	LIR2032	$6.49	1	$6.49	Link
12V Rechargeable Battery	seasider	B0D3YW51WJ	$20.99	1	$20.99	Link
Linear voltage regulator, 12V to 3.3V	STMicroelectronics	LD1117S33TR	$0.34	4	$1.36	Link
LDO, 3.6 V to 3.3 V	Diodes Incorporated	AP2112K-3.3TRG1	$0.56	1	$0.56	Link
NMOS gates	STMicroelectronics	STP6NK90Z	$3.64	11	$40.04	Link
10 K Ohm Resistor	Stackpole Electronics Inc	RMCF0805JG10K0	$0.10	9	$0.90	Link

1 K Ohm Resistor	Stackpole Electronics Inc	RMCF0805JT1K00	$0.10	1	$0.10	Link
1 µF Capacitor	Samsung Electro-Mechanics	CL21B105KBFNNNG	$0.11	6	$0.66	Link
220 Ohm Resistor	YAGEO	RC1206JR-10220RL	$0.10	9	$0.90	Link
100 nF Capacitor	Samsung Electro-Mechanics	CL21F104ZAANNNC	$0.10	3	$0.30	Link
10 µF Capacitor	Murata Electronics	GRM21BR61H106ME43L	$0.29	3	$0.87	Link
Door Arm	Trudoor	945AR	$14.97	1	$14.97	Link
Coin battery Holder	DigiKey Standard	DKS-CR2032H	$1.78	1	$1.78	Link
12V DC Power Supply	Parts Express	120-079	$17.98	1	$17.98	Link
3D Filament			$4.97	1	$4.97	

					$166.14	

So our total parts cost came out to be $166.14.

3.1.3 Total Cost

Cost	Total
Parts Cost	$166.14
Labor Cost	$47,241
Total Cost	$47,407.14

So the total cost of our entire project comes out to be $47,407.94

3.2 Schedule

Week	Task	Member
March 2nd - March 8th	Finalize our design document	All members will work together
	Determine/order what parts we will need in order to demo our project on the breadboard	
	Start putting together the breadboard demo by the end of the week	
March 9th - March 15th	Finish breadboard and demo it to the instructor and TA	All members will work together

	Finalize our PDB designs for the second round of PCBway orders	
	Talk with the machine shop regarding the door they are providing us and finalize any design decisions by the end of the week	
March 16th - March 22nd	Have parts all ordered in so the following week we can start with soldering our parts onto the PCBs once they arrive	All members will work together
March 23rd - March 29th	Start soldering parts onto the PCBS	All members will work together
	Start working on the ESP32 firmware	
	Start testing the PCBs, ensuring that all the components are soldered on correctly and that they are receiving the correct voltages	
	Make amends for the third round of PCBway orders	
March 30th - April 5th	Send in our third round orders for PCBway orders	All members will work together

	Start programming the PCBs that seem to be correct and ensure physical components are able to receive necessary signals	
April 6th - April 12th	Send in our fourth round orders for PCBway orders if needed	All members will work together
	Complete BlueTooth configuration and make any amends to ESP32 software development	
	Start assembling the physical components such as the box that will house the remote, the box that will house the door subsystem and the box that will house the latch subsystem	
April 13th - April 19th	Prepare for the mock demo	All members will work together
	Start testing the whole system, making any adjustments as needed for the project to work	
April 20th - April 26th	Demo with TA during weekly	All members will work

	meeting to prepare for the final demo next week Make any adjustments that the TA may have suggested Start working on the final paper for the project	together
April 27th - May 3rd	Present the project to the professor and TA Do a mock presentation to prepare for final demo	All members will work together
May 4th - May 10th	Finalize the final paper, ensuring that everything has been included Do the final presentation with the professor and TA	All members will work together

4. Ethics and Safety

4.1 Safety and Reliability of the Automatic Door Mechanism

This system is designed for people with physical disabilities therefore we will make sure to follow IEEE Code of Ethics section 1.1 to ensure the automatic door does not cause injury or malfunction. We will test this to an extent that will ensure that it does not malfunction and also make sure the opening and closing speed of the door is not too fast to ensure safety entering and leaving by the person with physical disabilities. To also ensure the safety of anyone or anything being crushed by the force outputted by the motor on the door, we will stop the motion of the door if the motor detects resistance in its rotational output. This will be done reading the rotational output from the motor’s encoder.

4.2 Sustainability and Environmental Responsibility

Since we have considered using batteries that are disposable such as the 12V battery, this can lead to electronic waste. According to ACM Code of Ethics section 3.3, which states to ensure the public good, the team will consider using rechargeable ones to reduce the electronic waste and promote sustainability. In the Remote Control subsystem, the LiR2032 3.6V battery is rechargeable in an attempt to reduce waste.

4.3 Campus and Lab Policies

To make sure that there is proper handling of tools and materials, we will follow the University of Illinois at Urbana-Champaign laboratory safety guidelines. This will ensure that the team is making sure of being well organized and keeping the workspace free of potential problems. This includes keeping food and drinks out of the laboratory, and also never working alone inside the laboratory. Any equipment that we use will be returned to the state that we found them in, which includes shutting down the soldering iron.

5. References

[1] “ESP32WROOM32 Datasheet.” *Espressif Systems*, datasheet, 2023, www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.

[2] Grainger Engineering Office of Marketing and Communications. “Salary Averages.” *Electrical & Computer Engineering | Illinois*, ece.illinois.edu/admissions/why-ece/salary-averages.

[3] IEEE. *IEEE Code of Ethics*. June 2020, www.ieee.org/content/dam/ieee-org/ieee/web/org/about/corporate/ieee-code-of-ethics.pdf.

[4[Payne, Michelle. “How Many Work Hours in a Year.” *FreshBooks*, 6 Sept. 2024, www.freshbooks.com/hub/productivity/how-many-hours-do-people-work-in-a-year#:~:text=There%20are%208%2C670%20hours%20

[5] Same Sky. *TACTILE SWITCH*. 12 Sept. 2024, pp. 1–5. www.sameskydevices.com/product/resource/ts02.pdf.

[6] STMicroelectronics. *Adjustable and Fixed Low Drop Positive Voltage Regulator*. datasheet, 2025, www.st.com/content/ccc/resource/technical/document/datasheet/99/3b/7d/91/91/51/4b/be/CD00000544.pdf/files/CD00000544.pdf/jcr:content/translations/en.CD00000544.pdf.

[7] *The Code Affirms an Obligation of Computing Professionals to Use Their Skills for the Benefit of Society.* www.acm.org/code-of-ethics.