AUTONOMOUS
HOT CAR AND CO POISONING MITIGATOR

By

Cathy Boman

Parvati Menon

Emily Xu

Final Report for ECE 445, Senior Design, Spring 2025

TA: Rui Gong

07 May 2025

Team No. 35

Abstract

This project is an autonomous solution to reduce the number of heatstroke deaths of children in hot, locked cars and the number of deaths from CO poisoning. Current products on the market do not address these issues urgently. Our product solves the aforementioned problems by creating ventilation for any passengers in a hot, locked vehicle. In addition, an alarm will sound and users will receive push notifications if the temperature or CO has reached the threshold. A camera will relay the vehicle's interior in real-time to the app so the user can monitor what is happening inside the vehicle.

Contents

1. Introduction

In 2024 alone, 39 kids died from heatstroke from being in a hot car [1]. Parents often forget or knowingly leave their children and pets behind in a hot, locked car. The temperature in a car can rise by 20 degrees in ten minutes. This increases the risk of heatstroke [2]. Despite laws and modern car technology, this issue is still prevalent today. Thus, it is critical to add protection and safety measures to vehicles to prevent further deaths.

Currently, there exist devices on the market that remind users to open the back door or check the backseat [3]. However, there are no autonomous solutions that work to mitigate the situation when the car's interior temperature is unsafe. Additionally, a vehicle's exhaust system produces carbon monoxide. There are CO detectors on the market today, but these devices simply sound an alarm when a certain threshold is reached. Since cars do not typically come equipped with CO detectors, this means that people might get notified of these dangerous emission levels too late.

Our device creates ventilation for passengers to prevent deaths in a hot car and alert users of a defective exhaust system. The device has a temperature and carbon monoxide sensor, which is attached near the driver's window (without obstructing the driver's view), that will lower all four windows using a signal. If the carbon monoxide levels are too high, the system will alert the user and recommend that they get their exhaust checked. When the temperature or the CO levels pass the threshold, the car's owner will be alerted through an app that the windows have been lowered. Furthermore, the vehicle has an intermittent alarm that sounds until the temperature levels are safe. A camera that is attachable to the rearview mirror streams footage to the app. Through the app, the user can monitor the inside of their car.

Our physical design can be found in Appendix E.

1.1 High-Level Requirements

1. Once the temperature sensor surpasses the threshold temperature of 85 ℉, the windows lower to the set position within two minutes.
    a. This is to ensure that the windows are being lowered quickly so that the car does not increase in temperature and begins cooling immediately.
2. The notification for either the CO sensor or the temperature sensor should be sent to the phone application within two minutes.
    a. This ensures that our system notifies the user as soon as possible of any issues in the vehicle.
3. The speaker will alert the user once CO levels within the car have reached nine or more ppm.
    a. Once the CO level reaches nine ppm, there is an increased risk of CO poisoning with minor side effects. This is a safe level to be exposed to CO for eight hours [4]. This requirement is here to ensure that our system will act as a CO detector to prevent cases of CO poisoning.

1.2 Subsystem Overview

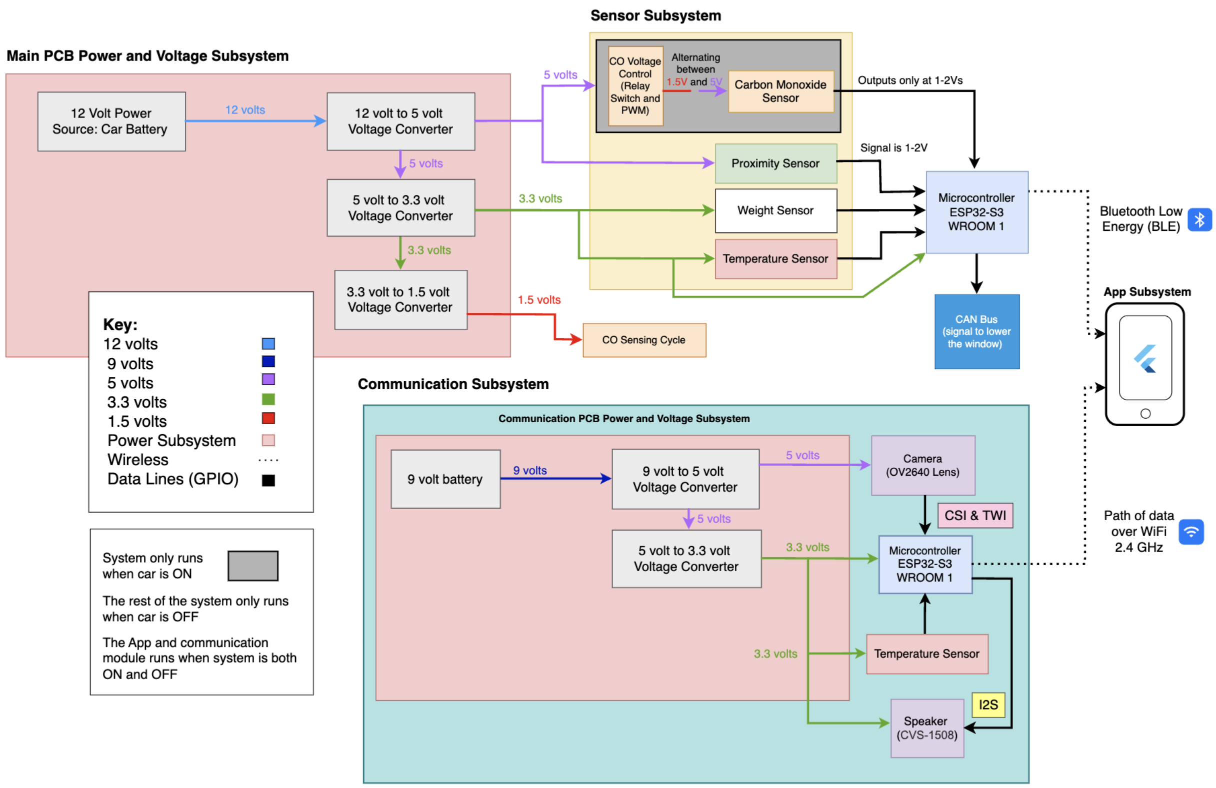


Figure 1: Block Diagram

Figure 1 depicts the final block diagram for our project. The critical subsystems of our project are the Power and Voltage Subsystem, Sensor Subsystem, App Subsystem, and Communication Subsystem. The Sensor Subsystem consists of all the sensors, which communicate with the microcontroller to execute the project's main functions. The app subsystem allows the system to notify the user by alerting them to the conditions of their car based on the outputs from the sensor subsystem. The user can monitor their car using the app. The Power and Voltage Subsystem is in charge of providing the correct voltage to each component. Lastly, the Communication Subsystem allows the user to monitor the interior of their car while they are away.

Through testing, we made three main changes to the block diagram: the removal of the speaker from the Main PCB, the elimination of level shifting for the speaker in the Communication PCB, and changing the transfer of data to be through Bluetooth.

2. Design

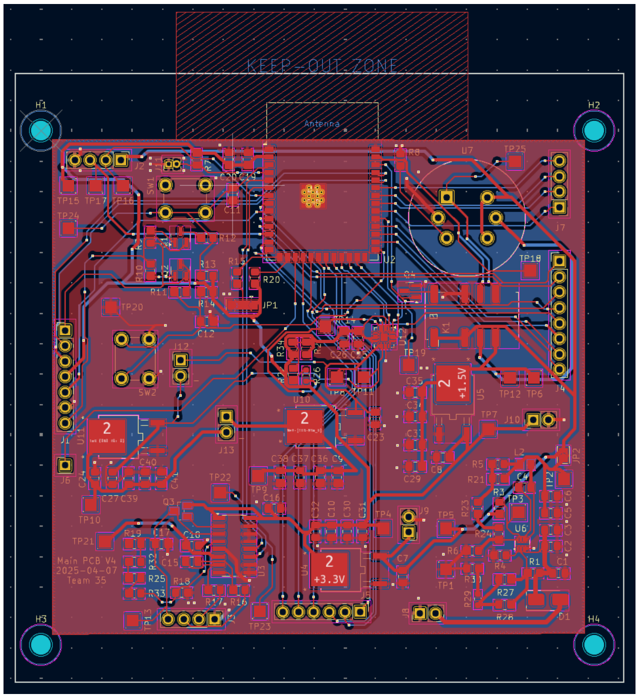


Figure 2: Main PCB

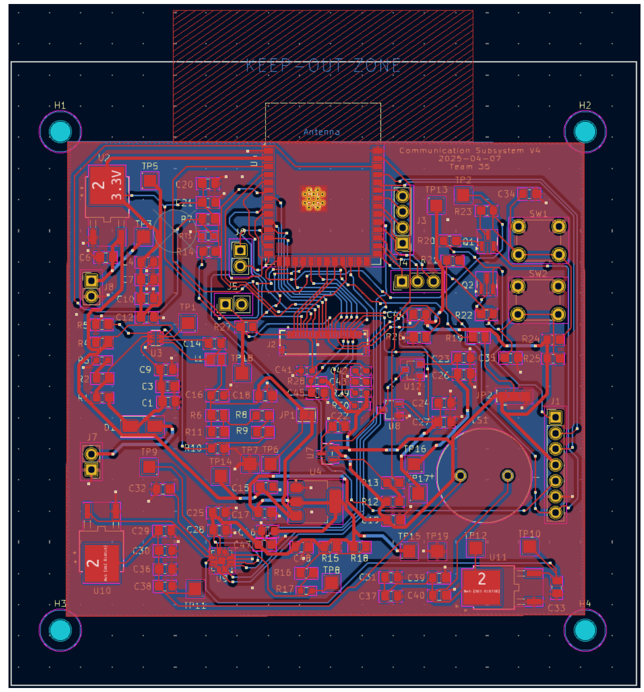


Figure 3: Communication PCB

In total, our design has two PCBs: the Main PCB, which controls all of the sensors, and the Communication PCB, which manages the camera and speaker systems. Our design is modular since the PCBs are composed of four separate subsystems. The two PCBs can act independently of each other.

2.1 Power and Voltage Control Subsystem

The Power and Voltage Control subsystem manages the system's power needs. This subsystem consists of multiple step-down voltage converters that convert the voltage from the primary power source to the required lower voltage for each component. Across both PCBs, this subsystem has voltage regulators for converting to five volts, 3.3 volts, and 1.5 volts.

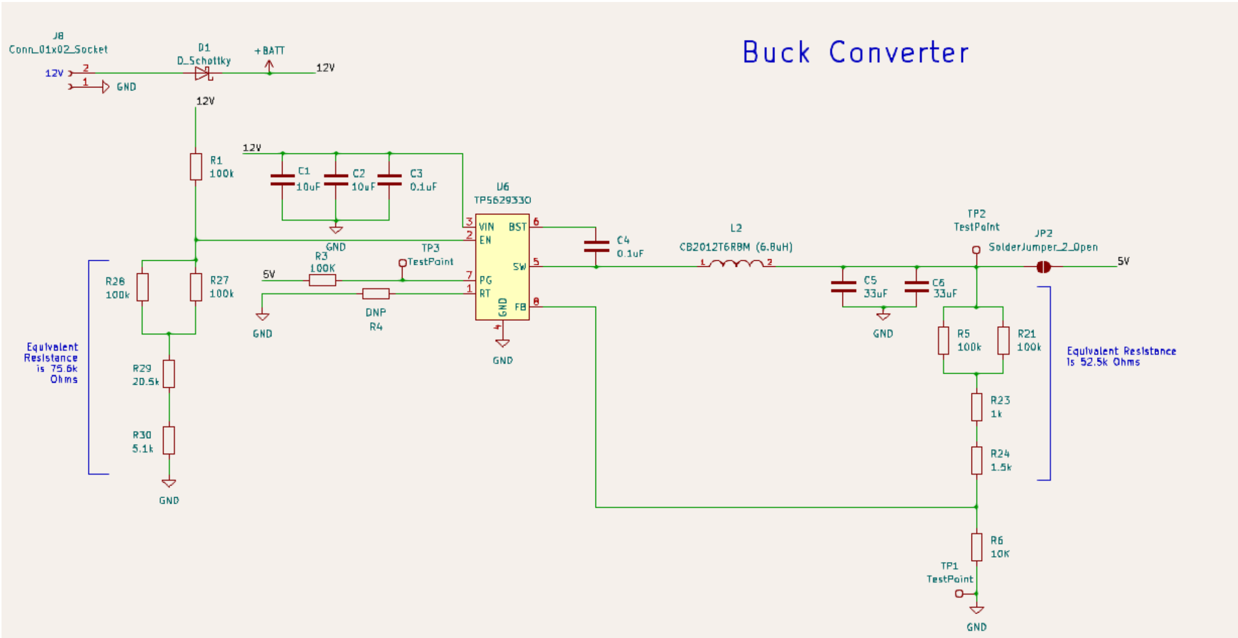


Figure 4: Buck Converter for 12 to 5 volts

For our main PCB, the input voltage is 12 volts. This voltage comes from the car battery and is considered the car's resting voltage. We need to convert down to five volts to power the CO sensor and the proximity sensor in our sensor subsystem. For demonstration purposes, a power adapter is used to supply 12 volts through the sockets. We chose to use a buck converter rather than a linear regulator because the change in voltage is significant. This results in considerable heat dissipation, requiring more complex circuitry to dissipate the heat efficiently. The base schematic for the buck converter is obtained from the Illinois Wiki [5].

A couple of values are adjusted for the resistors to match the voltage shift. A voltage divider circuit is made to supply the appropriate voltage level to the input of the buck converter's enable pin. The TPS629330 chip needs a maximum of six volts for this pin [6]. With a starting voltage of 12 V, we can achieve a voltage of 5.166 V with resistances of 100 kΩ and 75.6 kΩ to power the enable pin.

$$V_{EN} = 12 * \frac{75{,}600}{100{,}000 + 75{,}600} = 5.166\,V \quad (1)$$

To output five volts using this buck converter, there is a feedback pin composed of a 10 kΩ resistor and a 52.5 kΩ resistor. The converter maintains an internal voltage of 0.8 volts to ensure the output voltage is always constant [6]. This pin is made of a resistor divider circuit. The 10 kΩ resistor is recommended by TI, and we found the resistance for the 52.5 kΩ using the resistor divider formula (see equation 2 below).

$$R_{FBT} = \frac{V_{OUT} - V_{REF}}{V_{REF}} * R_{FBB} = \frac{5\,V - 0.8\,V}{5V} * 10k = 52.5\ k\Omega \quad (2)$$

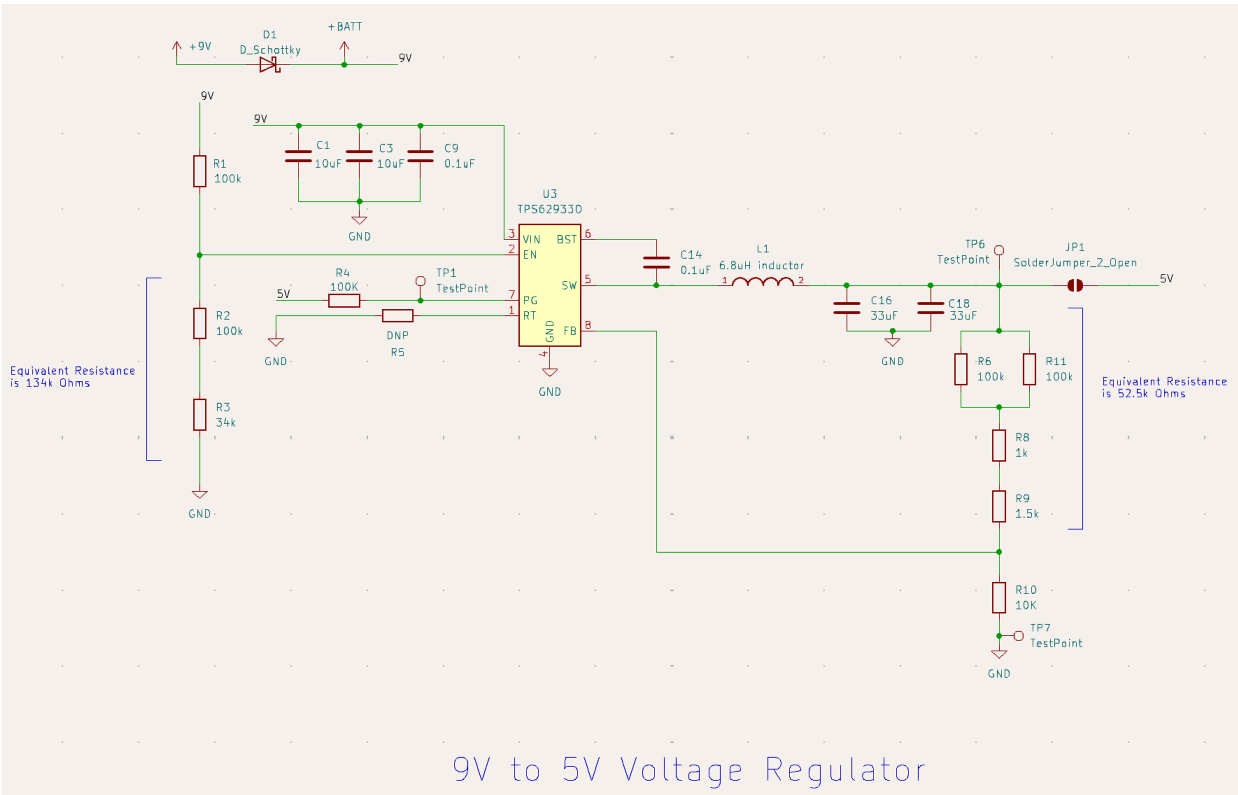


Figure 5: Buck Converter for 9 to 5 volts

The communication PCB will be powered by a nine-volt lithium battery. The 9-to-5 voltage converter powers the camera. Thus, the same circuit is used for the base buck converter module, and only some resistor values to the enable pin are modified. The input to the enable pin still needs to be at a maximum of six volts. Values of 100 kΩ and 134 kΩ give the enable pin roughly 5.154 V.

$$V_{EN} = 9 * \frac{134{,}000}{100{,}000 + 134{,}000} = 5.154\,V \quad (3)$$

Similar to the 12-to-5 buck converter, to output five volts using this buck converter, there is a feedback pin composed of a 10 kΩ resistor and a 52.5 kΩ resistor. The converter maintains an internal voltage of 0.8 volts to ensure the output voltage is always constant [6]. This pin is made of a resistor divider circuit. The 10 kΩ resistor is recommended by TI, and we found the resistance for the 52.5 kΩ using the resistor divider formula (reference equation 2, since V_{OUT} is the same for both converters).

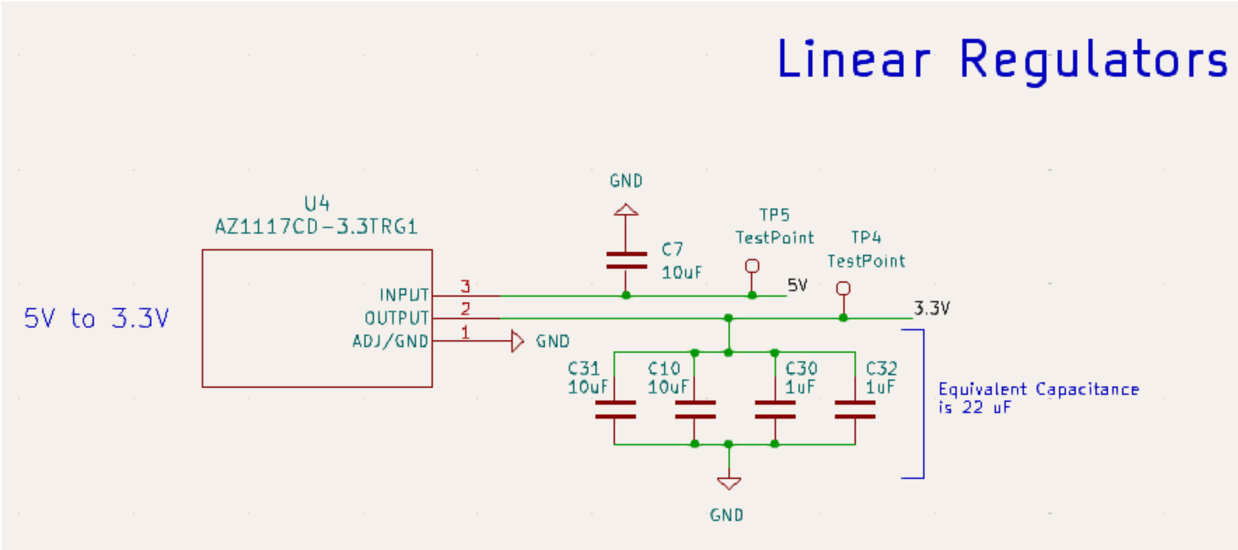


Figure 6: Linear Regulator for 5 to 3.3 volts

The temperature sensor, weight sensor, and both the ESP32-S3-WROOMs require 3.3 volts. A linear regulator, AZ1117-3.3, converts the five volt power source to 3.3 volts. The circuit for this linear regulator is based on the datasheet, and it includes a low-pass filter, referred to as “Ripple Rejection,” to ensure no unnecessary interferences are transmitted through the output [7].

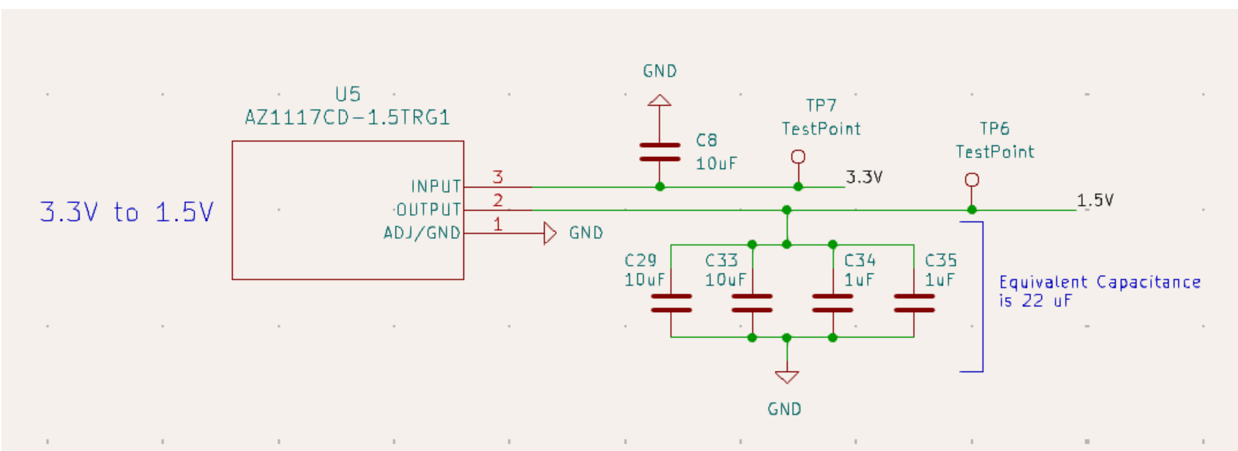


Figure 7: Linear Regulator for 3.3 to 1.5 volts

Since the CO sensor needs 1.5V, our system will also need a 3.3-to-1.5 linear regulator. The schematic for this converter is the same as the one for our 5-to-3.3 voltage converter since the electrical characteristics for the AZ1117C-3.3 are almost identical to AZ1117C-1.5, with the only changes being output voltage levels [7]. This converter was used to level shift the signal for the speaker, however, we found that there is no need to level shift the signal since the audio amplifier chip that we are using already outputs a voltage that the speaker can handle. This regulator now only exists on the Main PCB for the CO sensor.

The speaker on the Main PCB has been removed and is only on the Communication PCB since the speaker's only purpose is to alert the user through the alarm.

2.2 Sensor Subsystem

The sensor subsystem holds all the detection tools to monitor the car's state. All four of these sensors are powered by the Power and Voltage Control Subsystem.

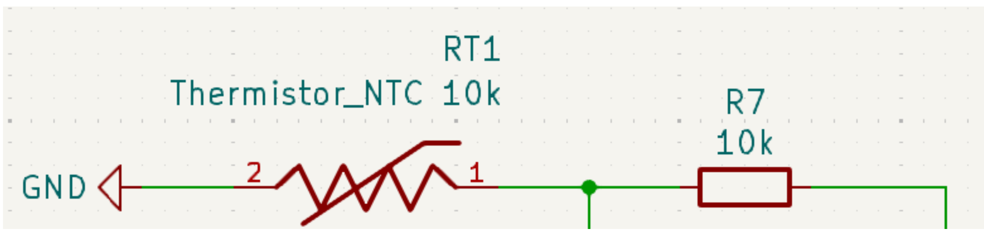


Figure 8: Temperature Sensor

The temperature sensor measures the car's internal temperature and relays it to the microcontroller for further processing. The values provided by this circuit dictate whether or not subsequent signals will be sent to lower the car windows. Since the temperature sensor can be easily made using a thermistor and a resistor [8], we are creating this sensor from scratch rather than buying a pre-built one. Additionally, since this circuit is powered by the same power source as the microcontroller at around 3.3V, there is no need to worry about converting voltages between components.

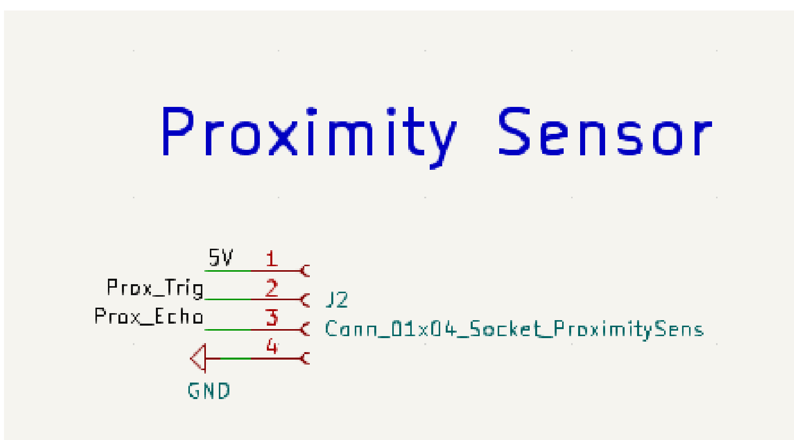


Figure 9: Proximity Sensor Connection to PCB

The proximity sensor detects whether or not the window is lowered enough. This is done by setting a flag and having the proximity sensor constantly check the distance of the object in front of it. When the temperature surpasses the threshold, a flag to lower the window is set. If a passenger is detected in the car and the temperature is past the threshold, then the microcontroller checks if the flag is set and if a window is present in front of the sensor. Once these conditions are met, the microcontroller sends a signal to the CAN bus (Controller Area Network Bus) to begin lowering the window. This bus is a network that is frequently used in vehicles to allow for communication between various components of the car, such as the brakes and engine [9]. The window continues lowering until the sensor does not detect a window in front of it. The flag is reset to indicate the window has finished lowering. An additional flag is set to confirm that the window has lowered and the motors are functional.

We used the SEN-24049 proximity sensor, which has a simple interface with the microcontroller. This is desirable since our proximity sensor is extended from our main PCB to detect the windows correctly.

The proximity sensor works by the trigger pin sending an ultrasonic signal first. From here, the echo pin times how long the signal takes to travel to the object and back. This time is then multiplied by the speed of sound to calculate the distance from the sensor to the object. The distance is checked against our threshold to ensure the window is lowered past the set level. The proximity sensor requires five volts to be powered. However, the output voltages of the trigger and echo pins are below 3.3V. This means that no level shifters are required for these when they send data to the microcontroller.

The code to make the proximity sensor work is based on here [10].

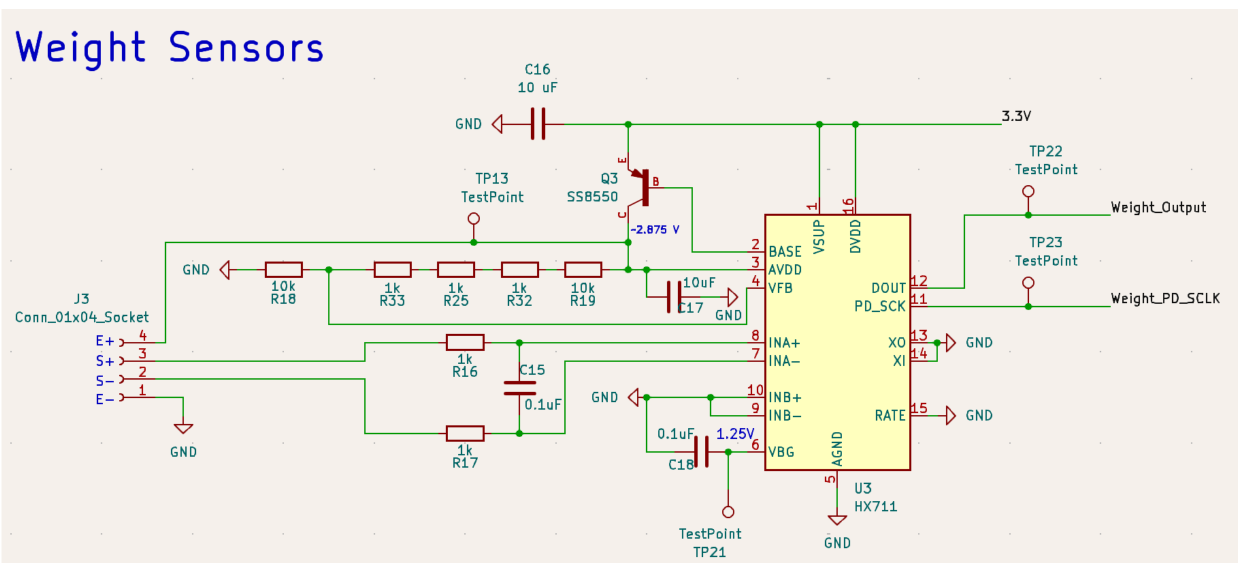


Figure 10: Weight Sensor

The weight sensor alerts the microcontroller if it measures anything greater than or equal to 40 pounds. The purpose of this sensor is to detect the presence of passengers; if sufficient weight is detected, a flag is set to enable the window-lowering system. Otherwise, even if the temperature in the car exceeds the temperature threshold, it will not open the windows. Our weight sensor consists of two parts: an HX711 chip and a load cell. A singular load cell can measure up to 20 kg (44 pounds) and interfaces with the HX711 chip through four wires. Thus, a 1x4 connector is located on our PCB to directly connect these wires to our design. Initially, we used the four half-bridge load cells; however, we found that it was difficult to get the correct reading from these load cells. For example, they were unstable even after we placed a plate over them, and the calibration values were highly inconsistent. After switching to a rectangular load cell, the readings were correct and within our margin of error.

The HX711 chip is an analog-to-digital converter that translates the output from the load cell to the correct weight values [11]. The circuit for this sensor follows the one provided by the official datasheet [12, Fig. 5]; the original schematic ensures that all the analog signal pins have low-pass filters, which makes it so that noise does not negatively affect the digital signals. However, we did change the resistance values to adapt to our system requirements. The resistance values for R1 and R2 in the datasheet are derived from the equation below [11].

$$V_{AVDD} = (R1 + R2)/R2 * V_{BG} \quad (4)$$

For an input voltage of 3.3 volts (the voltage of the microcontroller), the appropriate value of R1 lies between 12 kΩ - 15 kΩ [13]. The R2 resistor is 10 kΩ to keep V_{AVDD} in the needed range of 2.6 - 3.2 volts [13]. To ensure that V_{AVDD} isn't too close to the chip's upper and lower voltage boundaries, a value of 13 kΩ is chosen for R1, resulting in V_{AVDD} = 2.875 volts.

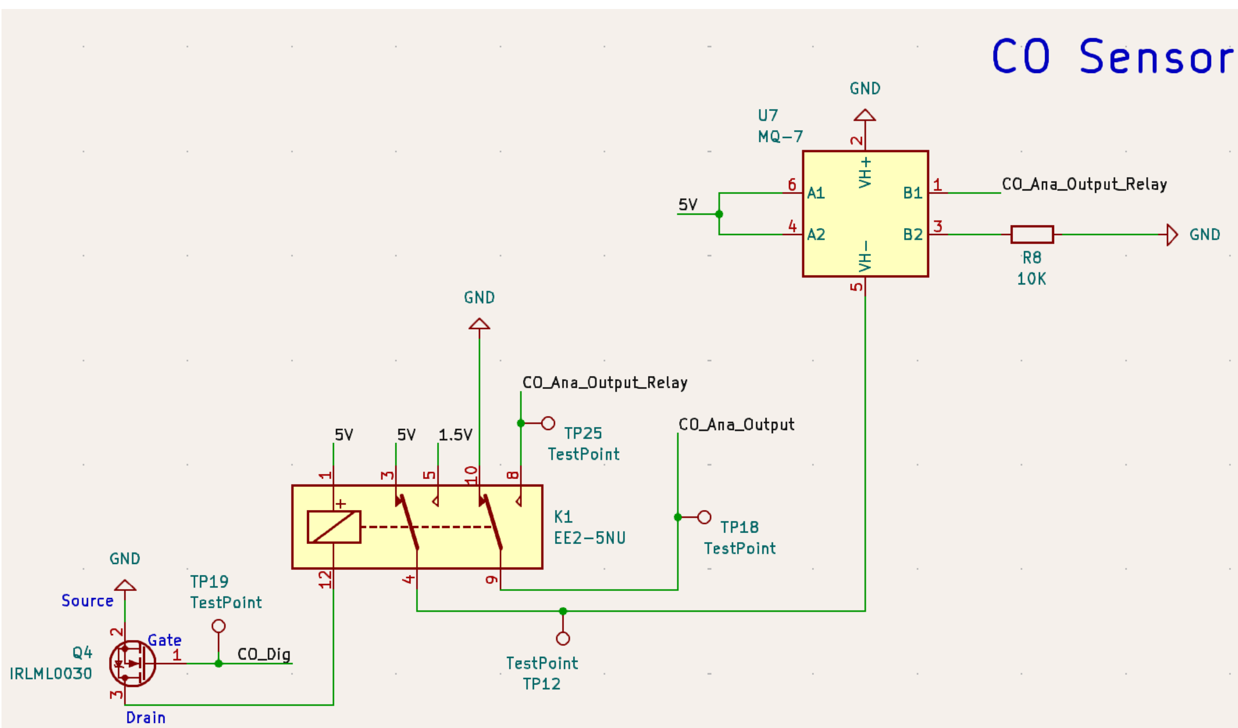


Figure 11: CO Sensor

The CO sensor periodically measures the CO levels in the vehicle. It communicates with the microcontroller if the levels are at or exceeding nine ppm.

Additional processing is done for the CO sensor to accommodate its unique voltage requirements. The MQ-7 sensor needs to cycle between five volts and 1.5 volts [14], so a five-volt powered relay switch [15] is used to change between the two values. The microcontroller controls the signal "CO_Dig" (the digital output) and activates the MOSFET at certain intervals to emulate a PWM. The intervals are 60 seconds for the heating cycle (powered by five volts) and 90 seconds for the sensing cycle (powered by 1.5 volts). Refer to Appendix D.2 Graph 1 for the pulse width modulation graph.

We use an always-on, non-latching relay, meaning the MQ-7 sensor receives five volts at default until CO_Dig is set to high. Once CO_Dig is high, the GND pin of the relay switch is connected. This change in voltage causes the relay to switch to 1.5 volts [16]. The MQ-7's output is routed to the relay switch to decide when to send the CO levels to the microcontroller. For example, when the CO sensor is receiving five volts, the input to the GPIO pin of the ESP32-S3 is zero volts. When the sensor receives 1.5 volts, the input to the GPIO pin is connected to the analog pin of the sensor, and the microcontroller receives the CO levels. This is to prevent sending a five-volt signal to the ESP32-S3 since the microcontroller can only handle signals at a maximum of 3.3V.

2.3 App Subsystem

The app allows the user to remotely monitor their car's interior through the camera in real-time. When the temperature inside the car has reached the threshold, the app alerts the user that the car windows have been lowered. If the CO levels reach nine or more ppm, the app notifies the user and recommends that the car exhaust be changed. This is done through wireless communication: the microcontroller on the Communication PCB sends the camera data over WiFi, and the microcontroller on the Main PCB sends the data over Bluetooth.

Our Communication PCB microcontroller is also able to host its own WiFi server. Users can connect to the wifi and access the live camera footage. They also have the option to save a screenshot of the footage to their device.

The app also allows the user to scan for any available Bluetooth devices. In our case, we connected the application to the Main PCB microcontroller's personal Bluetooth. Originally, the sensor data would be transmitted to our application through WiFi. However, due to insufficient documentation, we made the switch to Bluetooth for stable communication.

2.4 Monitoring and Communication Subsystem

The monitoring and communication subsystem allows the user to monitor their car remotely from the app. Since the communication subsystem is located above the rearview mirror, a second PCB is used for this system. The data from the camera and speaker is stored in the second ESP32-S3 WROOM microcontroller located on the Communication PCB. The system consists of a camera that relays the vehicle's interior in real-time through WiFi and a speaker to produce the alarm sound. The OV2640 lens is integrated with the PCB via an FFC-24 connector.

Originally, a second temperature sensor would be used to regulate the battery's temperature since the system is located in a heated setting, simulating the environment of a hot car. We changed this design in our final version, and the sensor now functions to sound the alarm once the vehicle's internal temperature has reached the threshold of 85°F.

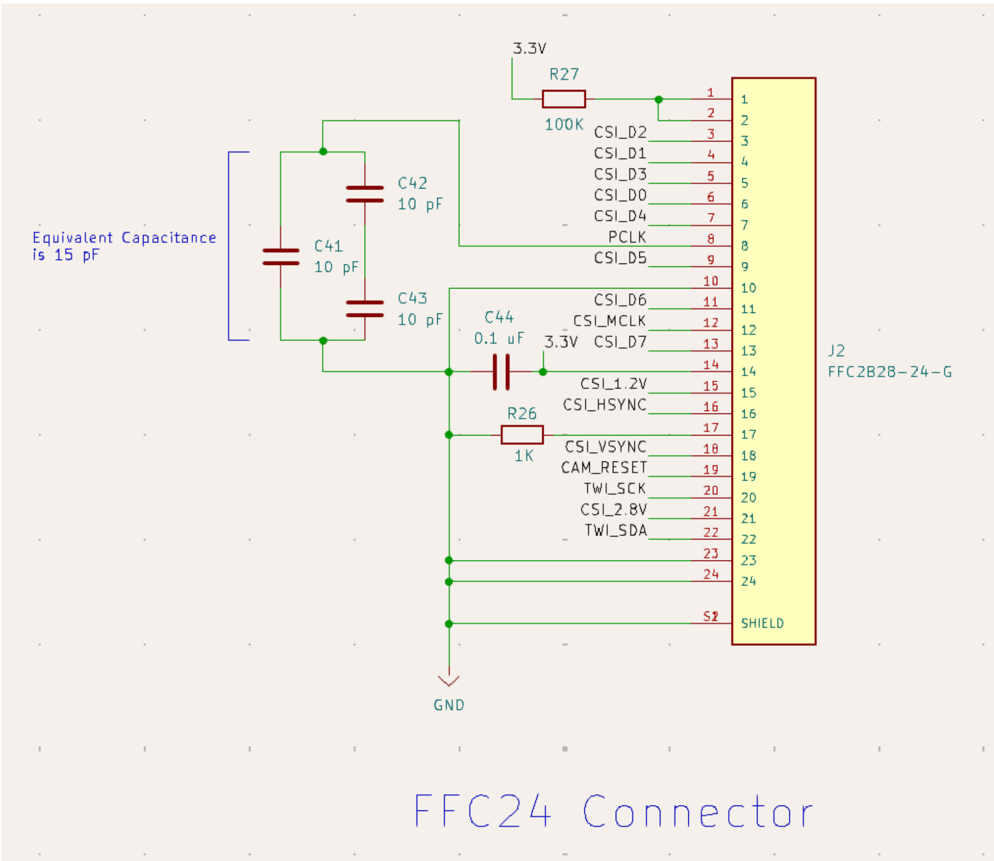


Figure 12: FFC-24 Connector

The pins of the FFC24 Connector connect to the microcontroller via the CSI and TWI protocols. It will attach to the OV2640 lens for recording the car's interior.

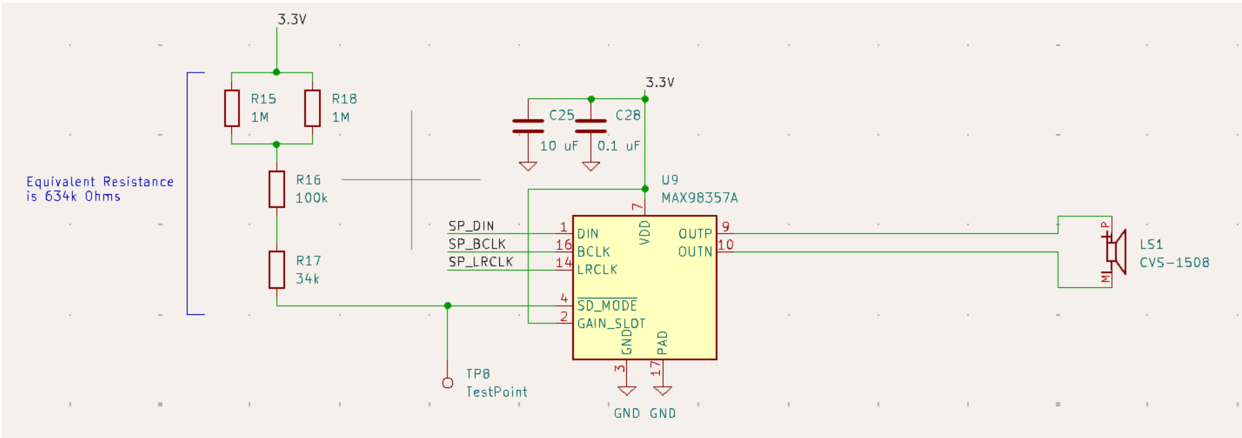


Figure 13: Audio Amplifier and Speaker

The Audio Amplifier accepts I2S data protocol from the microcontroller via the SP_DIN, SP_BCLK, and SP_LRCLK pins. Serial data is input on the rising edge of the SP_BCLK (bit clock input), and SP_LRCLK (left/right frame clock) specifies if the left channel is selected. Pin 4, SD_MODE, is used to select the data channel that is output by the amplifier. It uses a pull-up resistor value of 634 kΩ, which provides a high enough value to select both the left and right channels of the stereo input data [17].

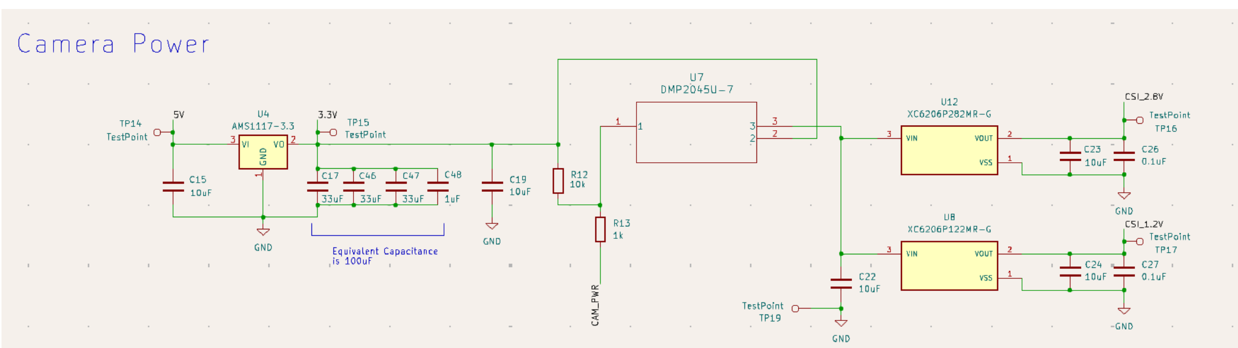


Figure 14: Camera Power

The Camera Power circuit utilizes a 3.3-to-1.5 voltage regulator, P-channel MOSFET, 2.8V linear regulator, and 1.2V linear regulator. These regulators are used to power the CSI_2.8V and CSI_1.2V pins. These two pins require different voltages; thus, two linear regulators are used to provide the correct voltage to each pin. The 3.3V source is from the ESP32-S3 WROOM. CSI_2.8V and CSI_1.2V connect to pins 21 and 15 on the FFC-24 connector, respectively.

The code to get the camera working is referenced here [18], while the code to activate the microcontroller's WiFi is referenced here [19].

3. Verification

3.1 Power Subsystem

The power subsystem has a total of five step-down converters. We needed to ensure that all of our converters were correctly outputting the right voltage levels to properly power our components. The full verification table for our power subsystem is found in Appendix A Table 1 and the verification figures are in Appendix D.1 Images 1-10. We were able to verify the success of this subsystem using a multimeter. To measure all voltage levels, we placed the ground probe of the measuring tool to GND and the power probe to the respective input or output of the converter chip.

On the Main PCB, there are a total of three voltage converters. For our 12-to-5V buck converter, we were able to step down a 12.18V input to a 5.01V output. For our 5-to-3.3V linear regulator, we successfully stepped down an input of 5.01V to 3.3V. The 3.3-to-1.5V linear regulator was able to step down the 3.3V input to a 1.49V output.

On the Communication PCB, there are two voltage converters in the system. The 9-to-5V buck converter converted an input of 9.01V to a 5.04V output. Lastly, the 5.04V was stepped down to 3.29V in our 5-to-3.3V linear regulator.

Since all of our measured voltages are within 0.5V of the target voltage, we were able to confirm the success of this subsystem.

3.2 Sensor Subsystem

3.2.1 Temperature

For the full verification table of the temperature sensor, please refer to Appendix A Table 2 and the verification figures are in Appendix D.2 Images 11-12. To ensure that we are sufficiently powering the temperature sensor, we used a multimeter to verify its input voltage. After placing the ground probe to GND and the power probe to the input of our temperature sensor, we discovered that 3.29V was going into this sensor, satisfying our voltage requirements.

To test the reliability of our temperature sensor, we compared its readings to those produced by an external thermometer. Our temperature sensor reads within a 1.5-degree range of the thermometer, signifying that our sensor is accurate.

3.2.2 Proximity

The verification for the proximity sensor can be found in Appendix A Table 3 and the verification figures are in Appendix D.2 Images 13-15. To validate that the proximity sensor is properly powered, we used a multimeter and measured the voltage by placing the power probe on the VCC pin of the proximity sensor and the ground probe to GND on the PCB. The resulting voltage was 5.01 volts, which is within our margin of error of 0.5 volts. This satisfies our first requirement for the proximity sensor. The second requirement to validate is ensuring that the proximity sensor reads if a window is 1.5 to 2.5 inches in front of it. We placed an object about 2 inches in front of the proximity sensor, and the sensor detected the distance as 2.23 inches exactly. This distance was confirmed accurate with a tape measure.

3.2.3 Carbon Monoxide

The verification for the CO sensor can be found in Appendix A Table 4, and the verification figures are in Appendix D.2 Images 16-17 and the PWM graph is in Appendix D.2 Graph 1. To ensure that our relay system for our PWM is working correctly, we measured the input voltage to our CO sensor with a multimeter by placing the power probe on the MQ-7's power pins and the ground probe on GND. For the first 60 seconds, we verified that 5.01V was being sent to the sensor during the heating cycle. In the next 90 seconds, we also confirmed that 1.49V was input to the MQ-7 component during the sensing cycle.

To test for CO levels, we lit a Post-it note, blew out the flames so that only smoke remained, and placed it within a plastic container with our CO sensor. This is because a blown-out Post-it releases CO. By containing the gases in a jar, we can measure the CO levels since burning objects without enough oxygen can emit CO rather than CO_2 [20].

Unfortunately, we were not able to validate the accuracy of our CO sensor itself due to issues with its wiring and insufficient documentation. No reasonable values were being read (i.e., 0 ppm values were constantly being recorded despite being in the presence of CO, or dangerously high ppm values were recorded in fresh air). For further explanations on potential reasons why this occurred, please refer to section 5.2: Uncertainties in our document.

3.2.4 Weight

The verification for the weight sensor can be found in Appendix A Table 5 and the verification figures are in Appendix D.2 Images 18-20. We verified that the voltage reading across the sensor was correct using a multimeter. We placed the ground probe of the multimeter to GND on the PCB and the power probe of the multimeter to the power pin of the HX711 chip. The result was 3.23 volts, which validates that the sensor is correctly powered since this is within our margin of error of 0.5 volts.

To confirm that the weight sensor is reading the correct weight within our margin of five pounds, we measured an object on a normal scale. We then used that same object on our weight sensor and found the readings to be the same. We weighed a MacBook, and the normal scale stated that its weight was 3.8 pounds. Our weight sensor reported the weight of the same laptop as 3.80 pounds.

3.3 App Subsystem

The verification table for the app subsystem is found in Appendix A Table 6 and the verification figures are in Appendix D.4 Images 25-27. To verify that the application can receive microcontroller data from 40 feet away, someone sat 40 feet away and checked that temperature data and push notifications were still being sent at that distance. We also timed how long it took to connect to Bluetooth and receive push notifications, and verified that it took less than two minutes.

3.4 Communication Subsystem

Refer to Appendix A Table 7 for the full verification table and the verification figures are in Appendix D.3 Images 21-24 for the communication subsystem.

3.4.1 Camera

The camera should receive five volts, and this was verified using a multimeter. This was done by placing the ground probe on GND and the power probe on the input to the camera power. To ensure good camera visibility, we placed an object 7+ feet away. The livestream footage displays the object clearly. Lastly, by waving in front of the camera, we were able to check that the video stream was smooth and had a minimum delay.

3.4.2 Speaker

For the speaker, we tested that the voltage reading across each leg was 1.5 volts and within a 0.5-volt margin of error. For the two legs of the speaker, we placed the ground probe of the multimeter on GND and the power probe on each leg to measure its output. We used a sound level meter to measure that the audio output level was 72.2 dBA, which is within our requirements [21].

4. Cost & Schedule

4.1 Parts

The total cost of all components and materials used to develop this project was $344.35. For a more detailed breakdown, please refer to Appendix B to see the individual components and costs.

4.2 Labor

For labor costs, we can expect a salary of $52/hr for each team member. These values come from the average salary of computer engineering obtained from the UIUC Grainger website [22]. This project involved a lot of research, discussion, design sessions, and testing. On average, our group estimates that we work 40 hours every week. We spent roughly 12 weeks on the project. Using this equation to calculate one partner's labor costs:

$$(\$/hour) \ * \ 2.5 \ * hours\ to\ complete \ = \ TOTAL \quad (5)$$

We get that one team member's labor cost is:

$$(\$52/hour) \ * 2.5 \ * 40 \ * \ 12 \ = \ \$62,400 \quad (6)$$

Thus, the cost for all three team members will be:

$$\$62,400 \ * \ 3 \ = \ \$187,200 \quad (7)$$

For the Machine Shop, we are assuming two staff members will spend 40 hours on our project.

$$(\$52/hour) \ * 2.5 \ * 40 \ = \ \$5,200 \ * 2 \ = \$10,400 \quad (8)$$

The grand total will be for this project would be:

$$\$344.35 \ + \ \$187,200 \ + \$10,400 \ = \$197,944.35 \quad (9)$$

4.3 Schedule

The first two weeks of our project development were devoted to setup. This included communicating with the Machine Shop, creating the initial block diagrams, and writing the project proposal. From February 24th to March 10th, our team worked on PCB designs and breadboard demonstrations. From March 10th to April 28th, we continuously built, tested, and revised our PCB design for our final demonstration on April 29th. Lastly, our team prepared for final presentations and papers until May 8th.

For a more detailed breakdown of our schedule, please refer to Appendix C.

5. Conclusion

5.1 Accomplishments

We had numerous accomplishments with our project. All our voltage converters worked independently as well as in series. We were successful in integrating our sensors to send a signal to the CAN Bus and for the speaker to sound when the temperature threshold was reached. The camera displayed its surroundings with good visibility and minimum delay.

On the software side, we successfully set up wireless communication between each PCB and our custom application. WiFi was used for the Communication PCB and Bluetooth for the Main PCB. We also ensured that the connection was stable and push notifications were still being sent at the range we specified. Our final product was able to access the camera live stream and receive real-time data updates with push notifications alerting the user when levels were dangerous.

5.2 Uncertainties

There were two uncertainties in our projects. We initially planned to have a speaker on our Main PCB in addition to the one on the communication PCB. However, the speaker on the main PCB had a lower decibel sound and inconsistent output. An example of the inconsistent output is how the speaker would play for a short duration of time and then stop while the program was still running. This problem is most likely a result of PCB design, specifically in designing the traces. The speaker circuitry for both PCBs was exactly the same. We also unit-tested this circuitry on a breadboard and found the output to be correct. Our solution to this problem was to remove the speaker from the Main PCB and only have it on the Communication PCB since the purpose of the speaker was to simply alert the user of high temperatures and dangerous CO levels.

The second uncertainty was with our carbon monoxide sensor. While we were able to get the CO sensor to cycle between its two voltages, the sensor provided incorrect readings. There are two possible reasons why we believe the CO sensor failed. The first reason is that when we tested the sensor, we saw incorrect ppm values. For example, the value would read 166.6 ppm in fresh air with no CO introduced. When we saw this, we thought we had wired the sensor wrong since the datasheet did not label the pins or the orientation of the sensor. Furthermore, the sensor is a circle with three pins on each side; thus, discerning the orientation is very difficult. The significance of this is that two pins (1,3) should never have voltage directly applied to them. When we switched the wiring, we may have applied voltage directly to these pins, damaging the sensor. The second reason is that the sensor itself could have been faulty. After we purchased the sensors, the type of CO sensors that we purchased was discontinued.

5.3 Ethical Considerations

Our project follows the IEEE Code of Ethics [23].

Protection of Car Components: We use the CAN Bus signal to communicate with the buttons to lower the windows. A problem that can arise is the window getting stuck instead of going down. This would lead to the motors continuously running, which ruins the user's car window [24]. To prevent this, we will ensure the signal is only being sent for two minutes. Our proximity sensor will be an additional layer of prevention by sending a signal to stop the window from lowering past a defined threshold. This preventative measure is meant "to avoid injuring … their property … by false or malicious actions," as defined by the IEEE Code of Ethics 7.8.II.9 [23].

Sensor Safety: We will use a multimeter to test the current reading across the CO sensor to ensure we do not consume too much power by exceeding the maximum it allows. This is also to ensure that we "hold paramount the safety, health, and welfare" as defined by the IEEE Code of Ethics 7.8.I.1 [23].

CO Testing/Demo Safety: Our project uses a CO sensor to detect whether or not the car is emitting carbon monoxide. To test and demo this sensor safely, we will test the CO sensor outside in a well-ventilated and isolated area. This is to ensure that we do not expose ourselves or others to carbon monoxide poisoning.

This complies with OSHA 1917.24(a), which states that "the carbon monoxide content of the atmosphere in a room, building, vehicle, railcar, or any enclosed space shall be maintained at not more than 50 parts per million (ppm) (0.005%) as an eight1. hour average area level" [25].

5.4 Future Work

We would like to improve the user experience of the application. Currently, our custom application includes a link to the camera livestream. If the camera footage were directly embedded into the Home Page, it would make for a more seamless design. Furthermore, we can advance our Bluetooth scanner function by having it become an automated process so that users do not have to manually scan and connect to the Bluetooth. Lastly, we would like to research more into the long-range antenna capabilities of our ESP32-S3 microcontroller. This will allow users to monitor their vehicle from a further distance, increasing the usability of our product.

References

[1] "Hot Car Deaths." *NSC Injury Facts,* Accessed: Feb 13, 2025 [Online]. Available: https://injuryfacts.nsc.org/motor-vehicle/motor-vehicle-safety-issues/hotcars/

[2] "Safety Resources: The Dangers of Leaving Your Child Alone in a Car." Seattle Children's Hospital. Accessed: Feb. 13, 2025. [Online]. Available: https://www.seattlechildrens.org/health-safety/injury-prevention/dangers-child-alone-car/

[3] P. Valdes-Dapena. "Hot car deaths can be prevented with this new technology," CNN.com. Jun. 29, 2023. Accessed: Feb 13, 2025. [Online]. Available: https://www.cnn.com/2023/06/29/business/hot-car-interior-radar/index.html

[4] "Carbon Monoxide Levels Chart." co2meter.com. Accessed: Feb 13, 2025 [Online]. Available: https://www.co2meter.com/blogs/news/carbon-monoxide-levels-chart

[5] "ECE 445 - Senior Design Laboratory." Illinois.edu, 2025. Accessed: Accessed: Mar. 06, 2025. [Online]. Available: https://courses.grainger.illinois.edu/ece445/wiki/#/esp32_example/index

[6] Texas Instruments. "TPS6293x 3.8-V to 30-V, 2-A, 3-A Synchronous Buck Converters in a SOT583 Package. " 2022. Accessed: Mar. 06, 2025. [Online]. Available: https://www.ti.com/lit/ds/symlink/tps62933.pdf

[7] Diodes Incorporated. "AZ1117C. " Sept. 2022. Accessed: Mar. 06, 2025. [Online]. Available: https://www.diodes.com/assets/Datasheets/AZ1117C.pdf

[8] Sunfounder, "5.10 Temperature Sensing — SunFounder ESP32 Starter Kit documentation." Sufounder.com. 2025. Accessed: Mar. 03, 2025. [Online]. Available: https://docs.sunfounder.com/projects/esp32-starter-kit/en/latest/micropython/basic_projects/py_thermistor.html

[9] M. Falch. "CAN Bus Explained - A Simple Intro [2025]." CSS Electronics, Jan. 2025. Accessed: May 06, 2025. [Online]. Available: https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial

[10] "ESP32 with HC-SR04 Ultrasonic Sensor with Arduino IDE." randomnerdtutorials.com. Jul. 20, 2021. Accessed: Mar. 05, 2025. [Online]. Available: https://randomnerdtutorials.com/esp32-hc-sr04-ultrasonic-arduino/

[11] Avia Semiconductor, "24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales." Accessed: Mar. 05, 2025. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf

[12] Avia Semiconductor, "HX711 Datasheet." 2025. Accessed: Mar. 05, 2025. [Online]. Available: https://www.lcsc.com/datasheet/lcsc_datasheet_2011051703_Avia-Semicon--Xiamen-HX711_C43656.pdf

[13] D. R. Jayakody. "Adapting HX711 modules for 3.3V Operation," Medium.com. Nov. 27, 2024. Accessed: Mar. 06, 2025. [Online]. Available: https://jayakody2000lk.medium.com/adapting-hx711-modules-for-3-3v-operation-38b04f309ad

[14] Zhengzhou Winsen Electronics Technology Co., LTD. "Toxic Gas Sensor (Model: MQ-7) Manual," 2014. Accessed: Feb 13, 2025. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-7%20Ver1.3%20-%20Manual.pdf

[15] "Hazardous Gas Monitor - SparkFun Learn." Sparkfun.com, 2025. Accessed: Feb 13, 2025. [Online]. Available: https://learn.sparkfun.com/tutorials/hazardous-gas-monitor

[16] A. Danner, "How to use a relay as a switch (#23 AC and Switching Circuits)." YouTube.com. Feb. 20, 2022. Accessed: Mar. 06, 2025. [Online]. Available: https://www.youtube.com/watch?v=T12PQRW2V10

[17] Maxim Integrated Products, Inc. "MAX98357A/ MAX98357B." 2018. Accessed: Mar. 06, 2025. [Online]. Available: https://cdn.sparkfun.com/assets/3/5/4/d/5/MAX98357A-MAX98357B.pdf

[18] SunFounder. "ESP32 CAM" docs.sunfounder.com. Accessed: Mar. 06, 2025. [Online]. Available: https://docs.sunfounder.com/projects/galaxy-rvr/en/latest/hardware/cpn_esp_32_cam.html

[19] "How to Set an ESP32 Access Point (AP) for Web Server." randomnerdtutorials.com. Accessed: Mar. 06, 2025. [Online]. Available: https://randomnerdtutorials.com/esp32-access-point-ap-web-server/

[20] T. King. "Case study: Carbon monoxide in candles." ADSL, Mar. 16, 2023. Accessed: Mar. 06, 2025. [Online]. Available: https://adslaboratories.com/case-study-carbon-monoxide-in-candles/

[21] Same Sky, "Model: CVS-1508," sameskydevices.com, 2020. Accessed: Feb. 13, 2025. [Online]. Available: https://www.sameskydevices.com/product/resource/cvs-1508.pdf

[22] University Of Illinois Urbana-Champaign. "Salary Averages." ece.illinois.edu. Accessed: Mar. 06, 2025. [Online]. Available: https://ece.illinois.edu/admissions/why-ece/salary-averages

[23] IEEE, "IEEE Code of Ethics," ieee.org, Jun. 2020. Accessed: Feb 13, 2025. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[24] Web Developer, "Power Window Safety Concerns: All You Need to Know," GlassFixit, Oct. 25, 2023. Accessed: Feb 13, 2025. [Online]. Available: https://www.glassfixitauto.com/blog/power-window-safety-concerns/

[25] "1917.24 - Carbon monoxide. | Occupational Safety and Health Administration," www.osha.gov.Accessed: Mar. 6, 2025. [Online]. Available: https://www.osha.gov/laws-regs/regulations/standardnumber/1917/1917.24

Appendix A Requirements and Verification Table

Requirement	Verification	Verification Status (Y or N)
12-to-5 Voltage Converter: ● Converts the 12 $\pm$ 0.5 V power source to 5 $\pm$ 0.5 V.	● The voltages are measured by a multimeter. ○ Place the ground pin of the multimeter to GND. ○ Place the power pin of the multimeter to the respective power pin of each component. ○ The input to the 12-to-5 V buck converter will be connected to the 12 V wall adapter. The input should read 12 $\pm$ 0.5 V. ○ The output of the 12-to-5 V converter should measure 5 $\pm$0.5 V. ○ The input of the proximity sensor should be connected to the output of the 12-to-5 V buck. This should read 5 $\pm$0.5 V. ○ The input to the power pin of the relay switch sensor should be connected to the output of the 12-to-5 V buck. This should read 5 $\pm$ 0.5 V.	Y
5-to-3.3 Voltage Converter: ● Converts the 5 $\pm$ 0.5 V power source to 3.3 $\pm$ 0.5 V volts.	● A multimeter is used to check the voltage across the voltage converter. The input should	Y

	read 5 $\pm$ 0.5 V and the output pins should read 3.3 $\pm$ 0.5 V. ○ Place the ground pin of the multimeter to GND. ○ Place the power pin of the multimeter to the respective power pin of each component. ○ The voltage reading across both microcontrollers measures 3.3 $\pm$ 0.5 V volts. ○ The voltage input to the temperature sensors measures 3.3 $\pm$ 0.5 V volts. ○ The voltage input to the weight sensors reads 3.3 $\pm$ 0.5 V. ○ The proximity sensor signal output reads 3.3 $\pm$ 0.5 V.	
3.3-to-1.5 Voltage Converter: ● Converts 3.3 $\pm$ 0.5V volts power source to 1.5 $\pm$ 0.5V volts.	● The voltages are all measured by a multimeter. ○ Place the ground pin of the multimeter to GND. ○ Place the power pin of the multimeter to the respective power pin of each component. ○ The input to the linear regulator is 3.3 $\pm$ 0.5 V. ○ The output of the linear regulator measures 1.5 $\pm$ 0.5 V. ○ The input to the speaker components	Y

	measures 1.5 $\pm$ 0.5 V.	
9-to-5 Voltage Converter: ● Converts the 9 $\pm$ 0.5 V volts from the battery to 5 $\pm$ 0.5 V volts. ● The voltage reading across the camera measures around 5 $\pm$ 0.5 V.	● The voltages are all measured by a multimeter. The input voltage reads 9 $\pm$ 0.5 V and the output reads 5 $\pm$ 0.5 V. ○ The voltage input to the ESP32-CAM measures 5 $\pm$ 0.5 V.	Y

Table 1: Power and Voltage Control Subsystem Requirements and Verifications

Requirement	Verification	Verification Status (Y or N)
The voltage reading into the temperature sensor is 3.3 $\pm$ 0.5 volts.	● Ensure that the temperature sensor is powered by pin 2 from the microcontroller. ● The voltage going into the temperature sensor circuitry is evaluated with a multimeter. ○ Place the ground probe of the multimeter in GND. ○ Place the power probe of the multimeter in pin 2 of the microcontroller. ○ Check that the input voltage is between 2.8 and 3.8 volts.	Y
The temperature sensor correctly reads the temperature within $\pm$ 5 degrees.	● In our enclosed testing environment, there is a heater, a temperature-checking meter, and the PCB with the temperature sensor together. ● Have the heater warm up the testing environment. ● Check the temperature readings from our sensor against the other temperature meter.	Y

	○ Ensure the sensor reading is within $\pm$ 5 degrees of the temperature meter.	

Table 2: Temperature Sensor Requirements and Verifications

Requirement	Verification	Verification Status (Y or N)
The voltage input to the proximity sensor is 5 $\pm$ 0.5 volts.	● The voltage across the proximity sensor will be confirmed using a multimeter. ○ The GND pin of the multimeter will be placed in GND. ○ The power pin of the multimeter will be placed in the VCC pin. ○ The proximity sensor is connected to the output of our 12-to-5V buck converter. ○ Check that the multimeter reads a voltage of 4.5 - 5.5 volts.	Y
The proximity sensor reads if a window is 2 $\pm$ 0.5 inches in front of it.	● Place an object 1.5 - 2.5 inches from the front of the proximity sensor. The distance will be measured by a measuring tape. ○ Check that the distance reported by the proximity sensor is accurate within 0.5 inches of the measured distance. ● Check that when the sensor reads that there is an object 1.5 - 2.5 inches in front of it, a signal is sent to stop lowering the window.	Y

	○ This can be checked by observing that the window has lowered.	

Table 3: Proximity Sensor Requirements and Verifications

Requirement	Verification	Verification Status (Y or N)
The voltage input to the CO sensor is 5 ± 0.5 volts when heating and 1.5 ± 0.5 volts when sensing.	● A multimeter is used to measure the voltage across the CO sensor. ○ Place the ground probe of the multimeter into GND. ○ Place the power probe of the power probe of the multimeter into pin 5. ● When the CO sensor begins the heating cycle, the multimeter measures 5 volts for 60 seconds ● After 60 seconds, the multimeter measures 1.5 volts for 90 seconds from the CO sensor.	Y
The CO sensor reads the CO levels (ppm) in the car within $\pm$ 1 ppm when the vehicle is on.	● A jar and a Post-it will be used to measure the CO level to set up our verification. On the software side, a flag is used to mimic the car being on. ● Light a Post-it note and place a jar over it to put it out. ● Place the CO sensor and the pocket carbon monoxide alarm in the container for comparison.	N

	• Confirm that the CO sensor and pocket carbon monoxide alarm readings match within 1 ppm.	

Table 4: Carbon Monoxide Sensor Subsystem Requirements and Verifications

Requirement	Verification	Verification Status (Y or N)
The voltage reading across the weight sensor is 3.3 $\pm$ 0.5 volts.	• The input to the VSUPP and DVDD pins of the HX711 chip should be connected to the output of the 5-to-3.3V linear regulator. These should be measured to 3.3 $\pm$ 0.5 volts with a multimeter. ○ Place the ground probe of the multimeter into GND. ○ Place the power probe of the power probe of the multimeter into VSUPP and DVDD. ○ Check that both pins output 3.3 $\pm$ 0.5 volts.	Y
The sensor reads the weight within $\pm$ 5 lbs.	• Weigh an object using a scale. • Weigh the same object using the weight sensor. Check that the weight from our sensor is within $\pm$ 5 lbs of the comparison weight.	Y

Table 5: Weight Sensor Subsystem Requirements and Verifications

Requirement	Verification	Verification Status (Y or N)

The app is able to receive the microcontroller data at least 40 feet.	● One person walks at least 40 feet away from the system with the app. ● Check that the microcontroller data is sent to the phone.	Y
The app notifies the user within two minutes of any changes detected in the car.	● Time how long it takes for the notification to be sent to the user via the app with a timer. ○ The time should be within two minutes. ● Test multiple times to ensure consistency.	Y

Table 6: App Subsystem Requirements and Verifications

Requirement	Verification	Verification Status (Y or N)
The voltage reading across the camera should be 5 ± 0.5 V.	● Place the positive and negative probes of the multimeter on the 5V and GND pins that connect to the camera. ● Test that the reading is 5 ± 0.5 V volts.	Y
The camera has a visibility of 6 - 8 feet.	● Place an object 6 - 8 feet away. ● Check that the object is within the video frame.	Y
The camera sends photos to the application within 35 to 45 feet.	● Check if a photo is sent to the user in the same room. ● Check if a photo is sent to the user within 35 - 40 feet.	Y
The camera video delay is at a maximum of 5 seconds.	● Wave an object in front of the camera. ● Time the delay of the object waving in the camera.	Y

The voltage reading across the speaker will be 1.5 $\pm$ 0.5 V volts.	• Place the positive and negative probes of the multimeter on the 1.5 V and GND pins that connect to the speaker. • Test that the reading is 1.5 $\pm$ 0.5 V volts.	Y
The audio output of the speaker will measure 73 dBA $\pm$ 3 dBA.	• A sound is played, and a sound level meter is used to confirm that the audio output level is 70 - 76 dBA.	Y

Table 7: Communication Subsystem Requirements and Verifications

Appendix B Components Costs

Description	Manufacturer	Part Number	Quantity	Cost($)
HX711 with 4pcs 50kg Load Cell Half Bridge Strain Gauge	Nextion	SC902 (Load Cell) HX711 (ADC chip)	1	8.99
CO Detector (MQ-7)	Shenzhen Weijia Security Technology Co.	WJ-CO997	1	9.99
9V Lithuim Battery	Voniko	CR-V9	2	13.99
12 Volt Adapter 3A	Alitove	N/A	1	9.99
12V to 5V Buck Breakout Board	ACEIRMC	16528	2	8.99
5V to 3.3V Voltage Regulator	Shutao	13397-1	15	7.99
3.3V to 1.5V Voltage Regulator	MECCANIXITY	mea220429ee1304	5	6.69
ESP32-CAM	HiLetgo	ESP32-CAM, OV2640	2	18.49
micro-USB cable	Monoprice	104867	2	3.98
USB-C to USB	JXMOX	4334964235	2	5.99
ESP32 S3	AYWHP	WROOM-1-N16R8	4	31.98
Thermistors 10k NTC	Cantherm	MF52A2103J3470	5	1.37
Buck Converter	Texas Instruments	TPS62933DRLR	13	10.59
3.3V to 1.5V Chip	Diodes Incorporated	AZ1117CD-1.5TRG1	24	14.6
Camera Connector	GCT	FFC2A32-24-T	1	0.64
PCB Speaker	Same Sky	CVS-1508	5	12.00
P-channel MOSFET (PCB)	UMW	S8550	8	1.44
Audio Amplifier	SparkFun	MAX98357A	1	6.50
Carbon Monoxide Sensor	Winsen	MQ-7	2	11.00
HX711	WWZMDiB	HX711	4	6.99
Relay Switch	KEMET	EE2-5NU-L	2	4.16

MQ-7 Development Board	ACEIRMC	MQ7	5	11.99
PCB Thermistor	TDX Corporation	NTCG203NH103JT1	5	0.95
NTC 10k Breadboard	PATIKIL	MF52103	20	6.29
XC6206-1.2V	Torex Semiconductor Ltd	XC6206P122MR-G	6	3.72
XC6206-2.8V	Torex Semiconductor Ltd	XC6206P282MR-G	6	3.48
AMS1117-3.3 Linear Regulator	UMW	AMS1117-3.3	2	1.24
Mini Hygrometer & Thermometer	Shenzhen Yongsheng Innovation Technology Co., Ltd	A01-2 Pack	2	7.99
Thermal Insulation Pads	Outus	N/A	1	7.99
Audio Amplifier Module	AITRIP	MAX98357A	5	9.99
Boost Switching Regulator 5V	Texas Instruments	TPS61222DCKT	5	6.35
Pin Headers and Connectors	Ruibapa	N/A	40	6.99
4.7 µH inductor	Taiyo Yuden	NR3015T4R7M	6	1.50
FTDI Serial Adapter	WWZMDiB	N/A	2	9.99
Extension Connector Board	MECCANIXITY	N/A	1	7.99
Solder Paste	Wonderway	Sn42/Bi58 T5	1	9.99
Tweezers	BoltHub	N/A	3	3.99
6.8 µH inductor	Taiyo Yuden	CB2012T6R8M	10	1.22
24 Pin FFC Connector	GCT	FFC2B28-24-G	10	5.03
Load Cell	ShangHJ	N/A	2	9.99
Main PCB 5th Pass	JLCPCB	N/A	10	18.15

Table 1: Cost of Materials

Appendix C Schedule

Week	Task
2/3	● Work on Project Proposal (All team members) ● Talk with Machine Shop (All team members)
2/10	● Work on Block Diagram ○ Sensor Subsystem (Emily) ○ Power Subsystem (Parvati) ○ Communication Subsystem (Cathy) ● Work on Project Proposal (All team members for most parts) ○ Sensor Subsystem (Emily) ○ Power Subsystem (Parvati) ○ Communication Subsystem (Cathy) ● Team Contract (All team members)
2/17	● Proposal Review Preparation (All team members) ● Proposal Review (All team members) ● Work on Subsystems of KiCAD Schematics ○ Sensor Subsystem (Emily) ○ Power Subsystem and CO sensor (Parvati) ○ Communication Subsystem (Cathy)
2/24	● Compile Individual KiCAD Schematics (All team members) ● PCB Design (All team members) ● PCB Review (All team members) ● Order Parts (All team members)
3/3	● PCB Revisions for PCBWay Orders 1 on **3/3** ● Design Document (All team members) ● Breadboard Layout and Testing (All team members)
3/10	● Breadboard Testing (All team members) ● Breadboard Demo (All team members) ● Follow-up with Machine Shop with Components (All team members) ● Solder components on Main PCB and Communication PCB ● PCB Revisions for PCBWay Orders 2 on **3/13**
3/17	● PCB Revisions (All team members)
3/24	● PCB Revisions for PCBWay Orders 3 on **3/31** ● Work on PCB and Testing (All team members)
3/31	● Work on PCB and Testing (All team members)
4/7	● Work on PCB and Testing (All team members) ● Demo Preparation (All team members) ● PCB Revisions for PCBWay Orders 4 on **4/7**

4/14	• Finalize PCB and Testing (All team members) • PCB 1 Revisions for JLCPCB Orders 5 on **4/17** • Team Contract Assessment (All team members) • Flutter Application Development (All team members) • Demo Preparation (All team members)
4/21	• Finalize PCB and Testing (All team members) • Flutter Application Development • Demo Preparation (All team members) • Mock Demo (All team members)
4/28	• Flutter Application Development • Demo and Presentation Preparation (All team members) • Final Demo (All team members) • Mock Presentation (All team members)
5/5	• Final Presentation (All team members) • Final Paper (All team members)

Table 1: Semester Schedule

Appendix D Verification Images and Graphs

D.1: Power and Voltage Subsystem Requirements

Image 1: 12 Volt Input (12-to-5V)

Image 2: 5 Volt Output (12-to-5V)

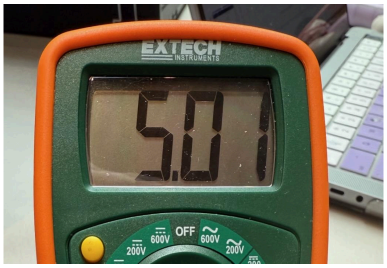


Image 3: 5 Volt Input (5-to-3.3V)

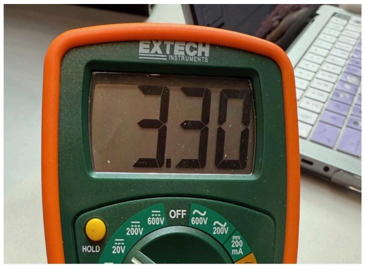


Image 4: 3.3 Volt Output (5-to-3.3V)

Image 5: 3.3 Volt Input (3.3-to-1.5V)

Image 6: 1.5 Volt Output (3.3-to-1.5V)

Image 7: 9 Volt Input (9-to-5V)

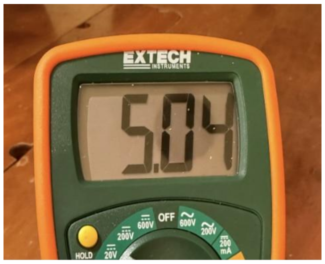


Image 8: 5 Volt Output (9-to-5V)

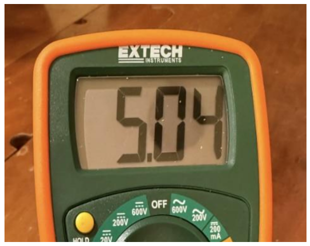


Image 9: 5 Volt Input (5-to-3.3V) ***Image 10: 3.3 Volt Output (5-to-3.3V)***

D.2 Sensor Verifications

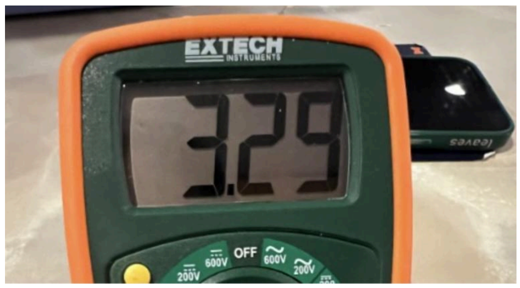


Image 11: Temperature Sensor Input Voltage

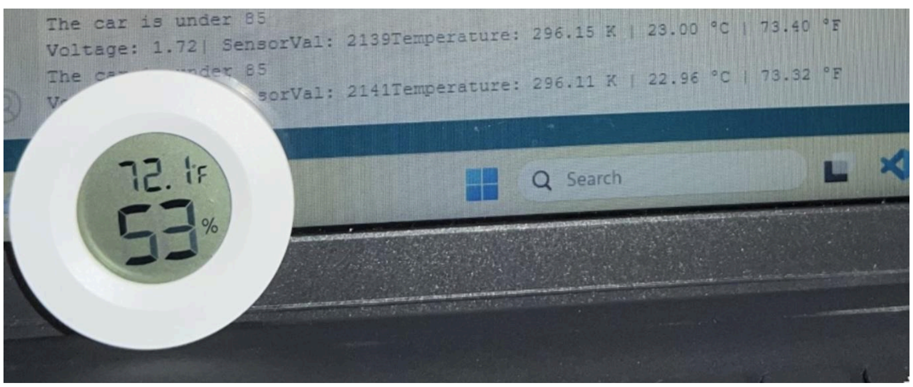


Image 12: Temperature Sensor Readings

Image 13: Proximity Sensor Input Voltage

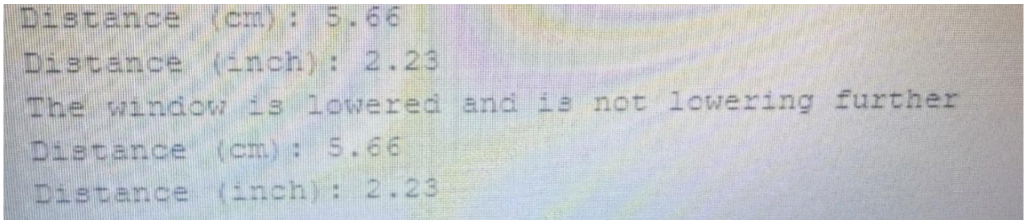


Image 14: Proximity Sensor Readings (Reads object is 2.23 inches in front)

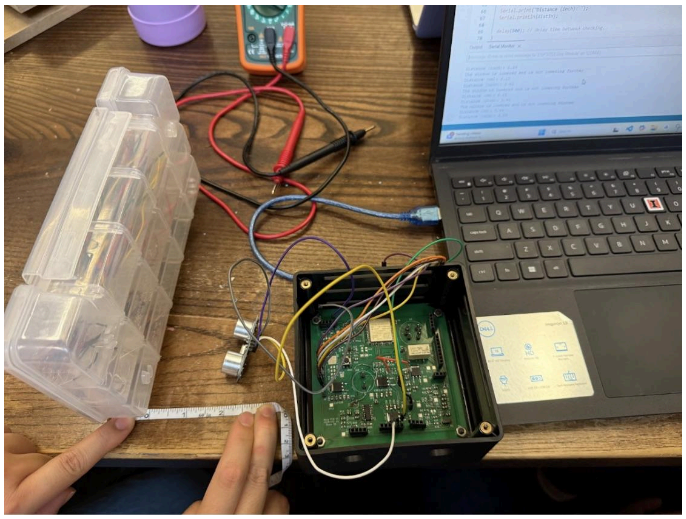
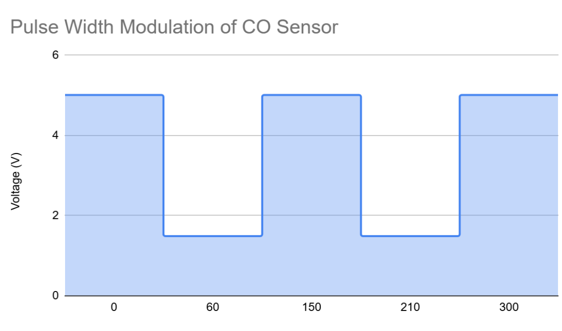


Image 15: Proximity Sensor Setup

Image 16: CO Sensor Sensing Voltage

Image 17: CO Sensor Heating Voltage



Graph 1: Pulse Width Modulation for Carbon Monoxide Sensor

Image 18: Weight Sensor Input Voltage

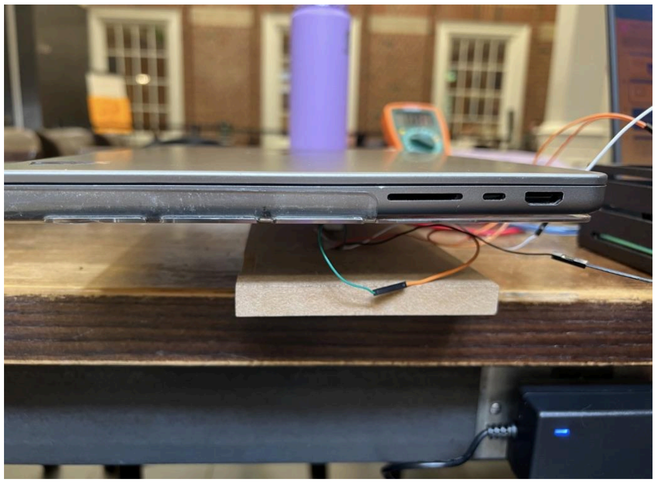


Image 19: Weight Sensor Setup

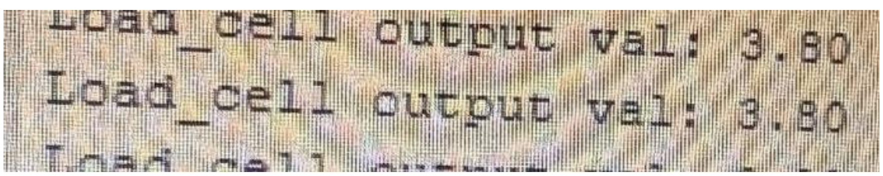


Image 20: Weight Sensor Readings

D.3 Communication PCB Verifications

Image 21: Camera Input Voltage

Image 22: PCB Camera Output

Image 23: Speaker Input Voltage

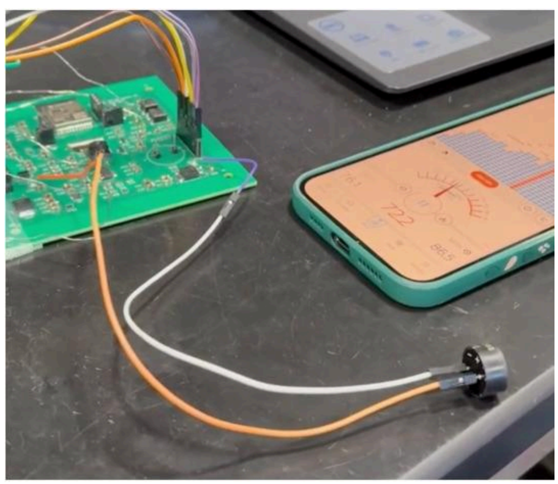


Image 24: Speaker dBA Readings

D.4 App Verifications

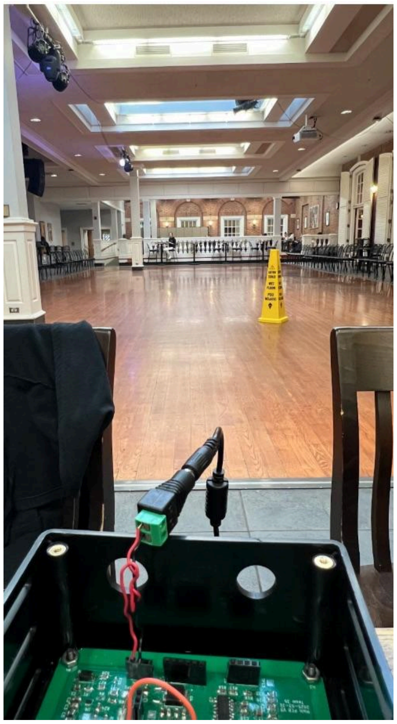


Image 25: Receiving Data from 40ft Away

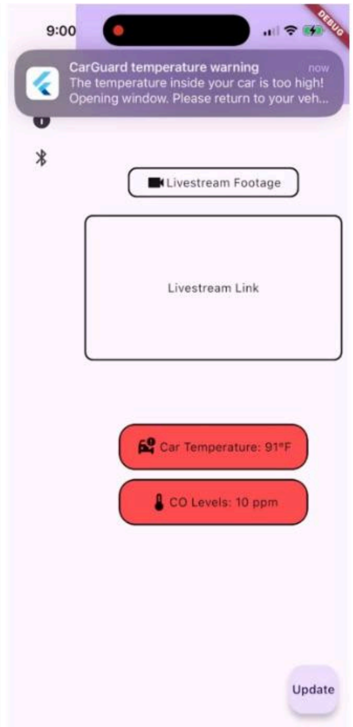


Image 26: Application (Home Page)

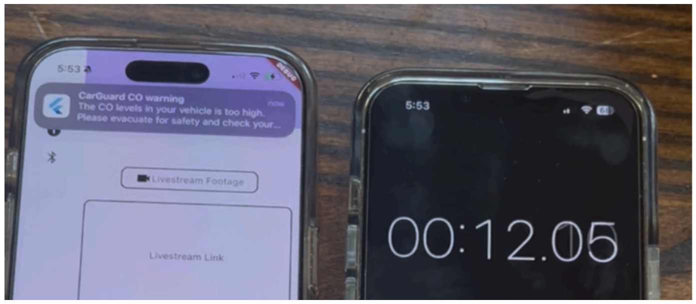


Image 27: Time it Takes to Receive Push Notifications

Appendix E Physical Design

Image 1: "Car" Setup with PCBs