AUTONOMOUS HOT CAR AND CO POISONING MITIGATOR

ECE 445 Spring 2025

Design Document

By

Team 35

Parvati Menon

Cathy Boman

Emily Xu

TA: Rui Gong

Professor: Viktor Gruev

March 6th 2025

Table of Contents

1 Introduction

1.1 Problem Statement

Every year, many children and pets die from heatstrokes in hot cars or carbon monoxide poisoning when they are left in a locked vehicle. Parents often forget or knowingly leave their children and pets behind in a hot, locked car. Even if the parent leaves for a quick 10 minute errand, there are still concerns about heatstrokes since temperatures inside can rise as much as 20 degrees in that short duration of time [1]. In 2024 alone, 39 kids died from heatstroke from being in a hot car [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, the volume of these alarms can be reduced, and as a result, parents can forget they are going off. There are no autonomous solutions that work to mitigate the situation when the car's interior temperature is unsafe. Additionally, there are carbon monoxide detectors on the market today, but these devices simply sound an alarm when a certain threshold is reached. However, if the user is not in the vicinity, they might get notified of the incident too late.

1.2 Solution

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 vehicle's temperature is too high and passengers are detected inside, then the device will lower the car windows. If the carbon monoxide levels are too high, the system will alert the user and recommend 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. The alarm shuts off automatically once the levels are safe or once the car is turned on. 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.

1.3 Visual Aid

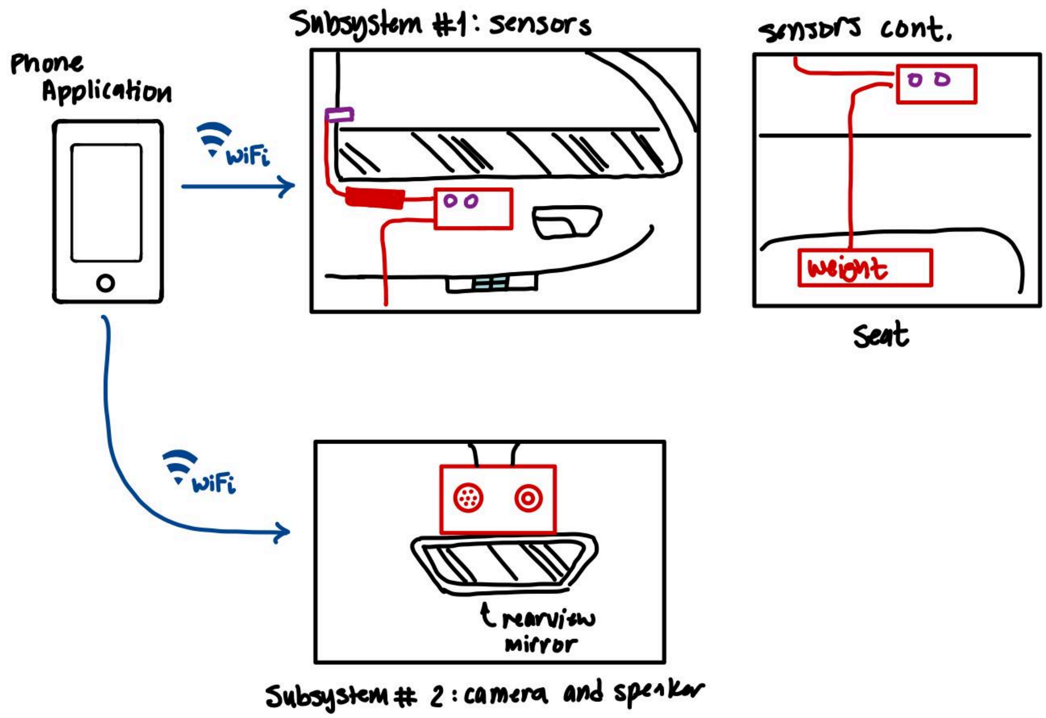

Figure 1: Car System Layout

1.4 High-Level Requirements

1. Once the temperature sensor surpasses the threshold temperature of 85 ℉, the windows lower to the set position within two minutes.
2. The notification for either the CO sensor or the temperature sensor should be sent to the phone application within two minutes.
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].

2 Design

2.1 Block Diagram

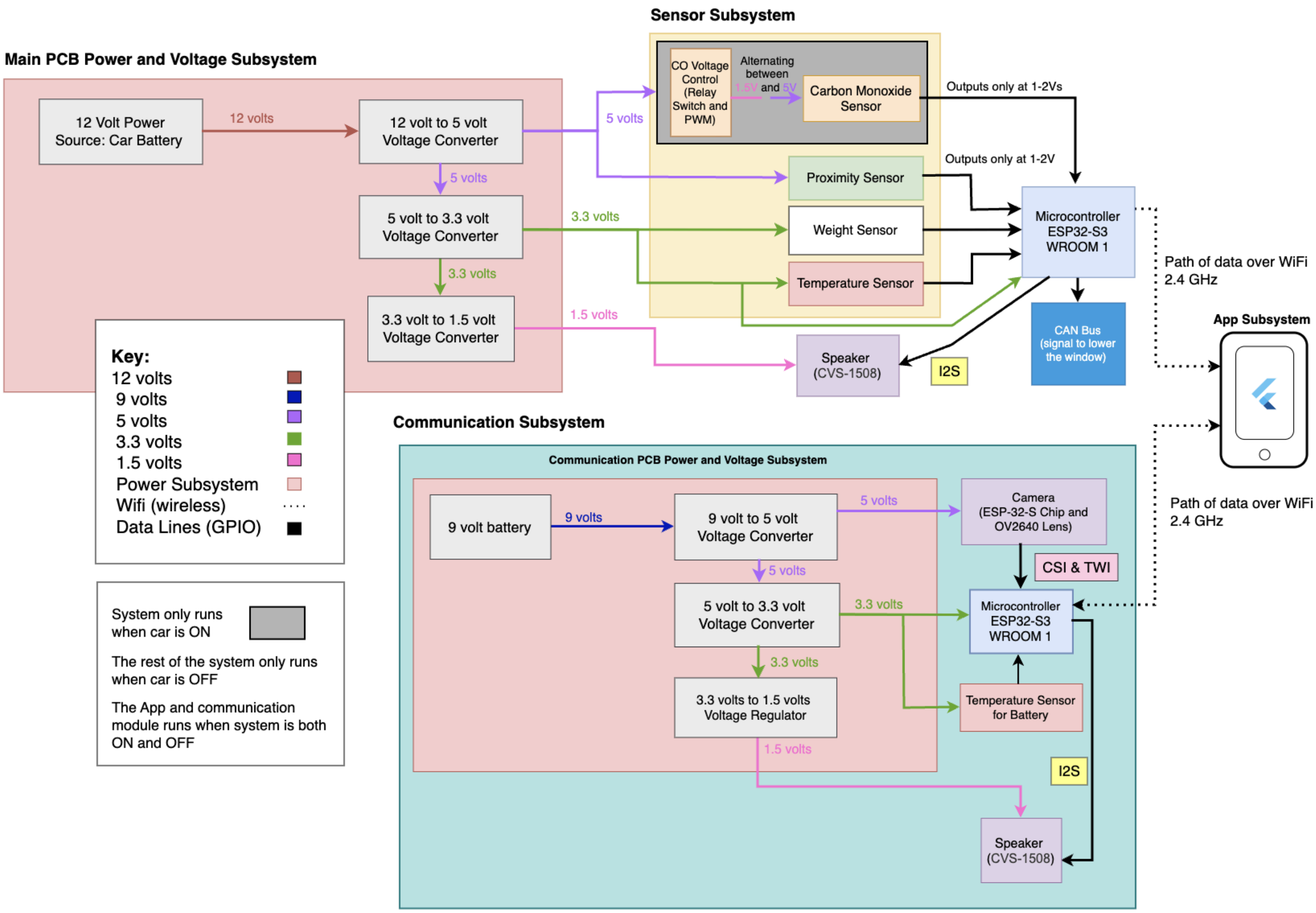

Figure 2: Car System Block Diagram

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

2.2 Physical Design

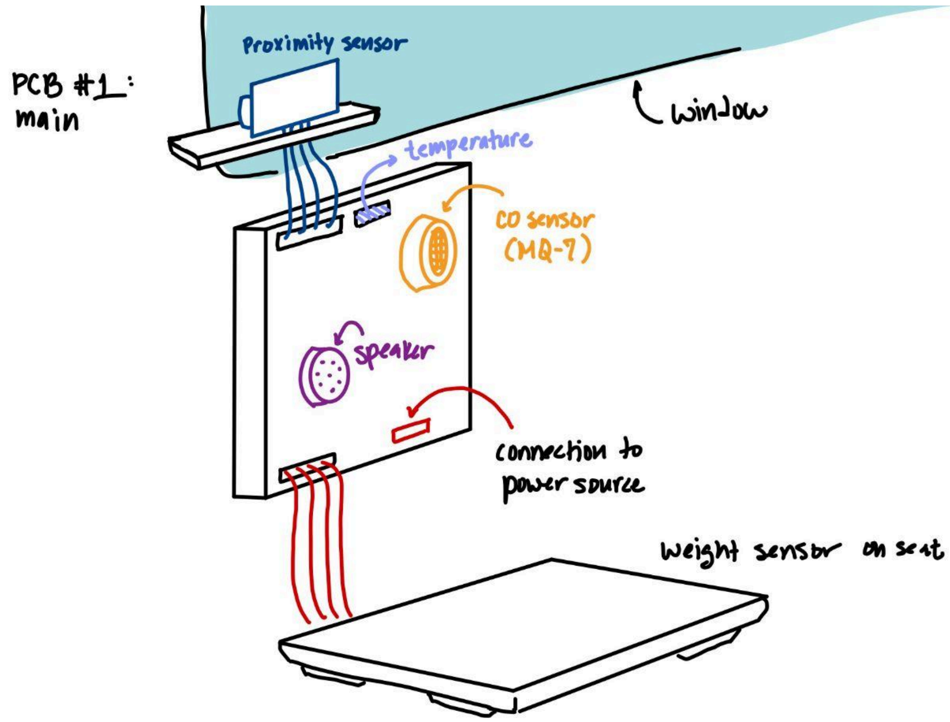

Figure 3: Physical Design of Main PCB (Sensors)

The PCB is mounted and encased in a box. This box is then attached to the car door. Holes are cut out for the connections to the weight and the proximity sensor. Additionally, holes are cut out so that the speaker, temperature sensor, and CO sensor can protrude from the box. This way, most PCB components are hidden and protected while allowing the sensors outside the box to detect the car's environment properly.

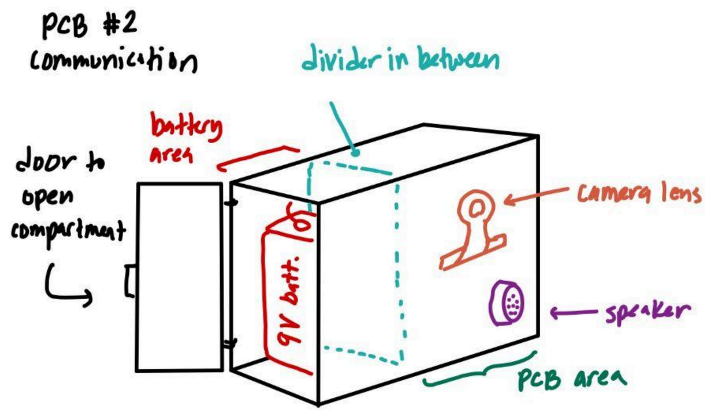

Figure 4: Physical Design of Communication PCB

The communication submodule is also enclosed with a box, but there is a divider inside the compartment. This way, if issues occur with the battery, it will not affect the PCB. There are places cut out of the box to allow the camera and the speaker to be visible and heard. A door is placed on the side of the box to allow for easy access to the battery. That way, it can be easily removed when it needs to be replaced.

2.3 Subsystem Overview

2.3.1 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: the CO and proximity sensors are connected to the 12-to-5 voltage converter, while the temperature sensor and the weight sensor connect to the 5-to-3.3 voltage converter. The CO sensor is also connected to the 3.3-to-1.5 voltage converter during its sensing cycle. The temperature, proximity, and weight sensors activate when the vehicle is off. The CO sensor is activated while the car is on.

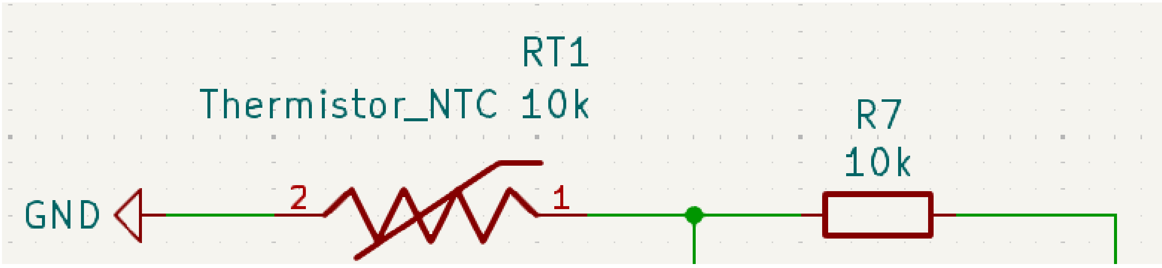

Figure 5: Temperature Sensor

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 [5], 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.

Requirements	**Verification**
The voltage reading into the temperature sensor is 3.3 ± 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.

The temperature sensor correctly reads the temperature within ± 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. ○ Ensure the sensor reading is within ± 5 degrees of the temperature meter.

Table 1: R&V Table for Temperature Sensor

Image 1: Image From SEN-24049 Digikey [6]

The proximity sensor detects whether or not the window is lowered enough. Once the proximity sensor no longer senses the window, it alerts the microcontroller to stop lowering it. We will use 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 straightforward four-pin connections allow us more flexibility in placing our proximity sensor [7]. Thus, all we need on our PCB is to ensure that the necessary connections route correctly to our microcontroller.

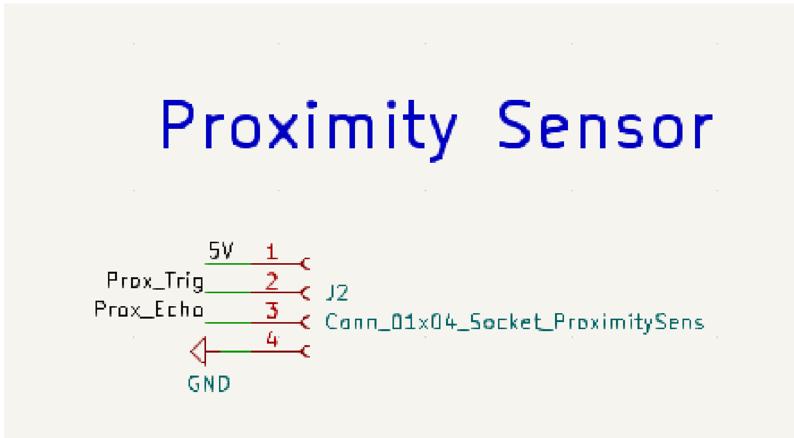

Figure 6: Proximity Sensor Connection to PCB

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. Thus, it is powered by the five volts from the 12-to-5 voltage converter. 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 [7].

Requirements	**Verification**
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.
The proximity sensor reads if a window is 2 ± 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. ○ This can be checked by observing that the window has lowered.

Table 2: R&V Table for Proximity Sensor

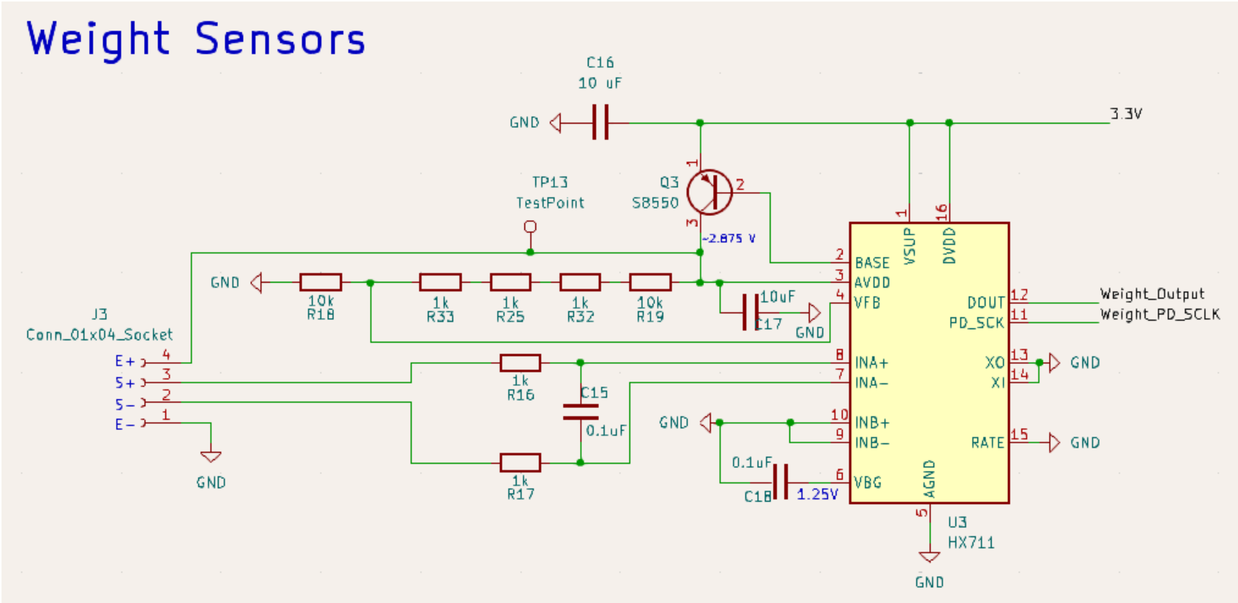

Figure 7: Weight Sensor

The weight sensors alert the microcontroller if it measures anything greater than 40 pounds. The purpose of the weight sensor is to detect the presence of passengers; if sufficient weight is detected, then the window-lowering system is enabled. 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 four half-bridge microload cells (SC902) [8].

The HX711 chip is an analog-to-digital converter that translates the output from the load cells to the correct weight values [9]. The circuit for this sensor follows the one provided by the official datasheet [10, Fig. 5]; the original schematic ensures that all the pins that work with the analog signals (pins 2, 3, 6, 7, and 8) have low-pass filters; this makes it so that noise does not negatively affect the digital signals. However, a couple of values are modified. The datasheet does not define the resistance of the voltage divider resistors connected to the AVDD pin, but they are derived from the equation $V_{AVDD} = \frac{(R1+R2)}{R2} * V_{BG}$ [9]. For an input voltage of 3.3 volts (the voltage of the microcontroller), the appropriate value of R1 lies between 12 kΩ - 15 kΩ [11]. The R2 resistor is 10 kΩ to keep V_{AVDD} in the needed range of 2.6 - 3.2 volts [11]. To ensure that the value of 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.

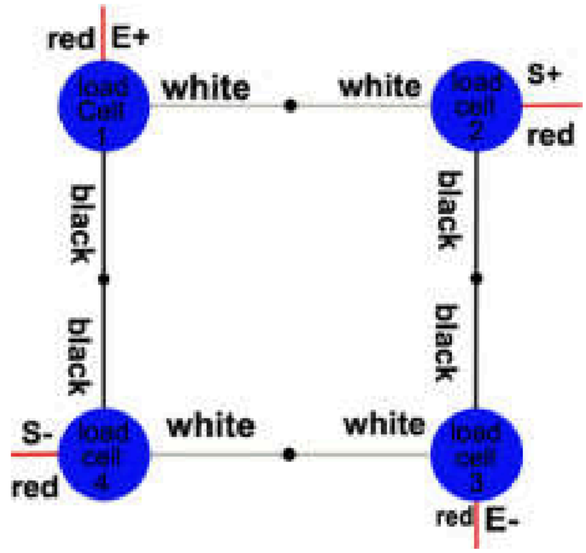

Figure 8: Weight Sensor Connections [8, Fig.3][12]

Each of the four load cells can measure up to 50 kg (110.23 lbs), allowing our system to detect up to 200 kg (440.925 lbs) [8]. These cells are connected to the HX711 module and each other [12][13]. Thus, a 1x4 connector is located on our PCB to allow the four sensors to communicate with the ADC chip.

Requirements	**Verification**
The voltage reading across the weight sensor is 3.3 ± 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 ± 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 ± 0.5 volts.
The sensor reads the weight within ± 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 ± 5 lbs of the comparison weight.

Table 3: R&V Table for Weight Sensor

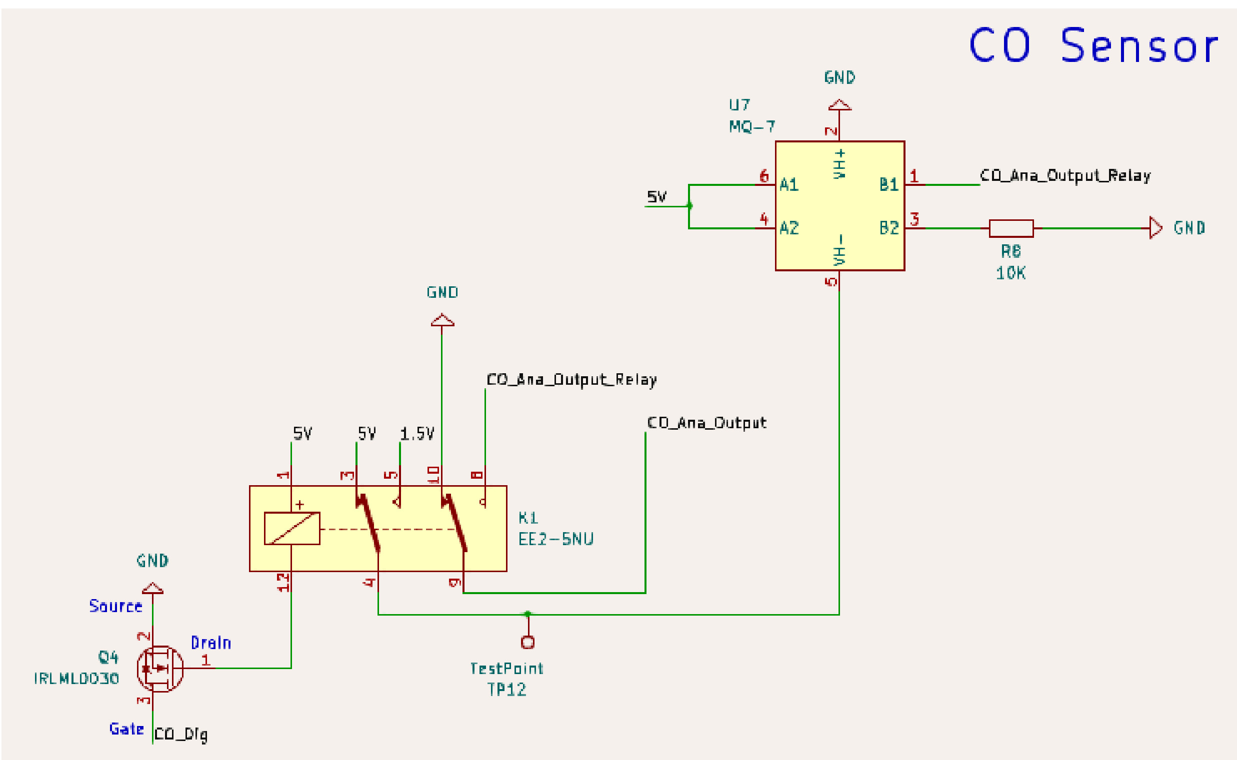

Figure 9: CO Sensor

The CO sensor periodically measures the CO levels in the vehicle when it is on. 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 needs to cycle between five volts and 1.5V [14], so a five volt powered relay switch [15] is used to change between the two values respectively. The microcontroller controls the signal "CO_Dig" (the digital output) and activates the MOSFET at certain intervals to emulate a PWM. We use an always-on, non-latching relay, meaning the MQ-7 receives five volts at default until CO_Dig sends the appropriate signal. Once CO_Dig receives the appropriate signal, the GND pin of the relay switch is connected. This change in voltage causes the relay to switch to 1.5V [16]. The MQ-7's output is routed to the relay switch as well since the sensor readis the CO levels at 1.5V. Thus, the input to the GPIO pin to the ESP32-S3 is zero volts when the CO sensor receives five volts. The input to the GPIO pin is 1.5V when the sensor receives 1.5V. 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.

Requirements	**Verification**
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.

The CO sensor reads the CO levels (ppm) in the car within ± 1 ppm when the vehicle is on.	● A jar and a candle 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. ○ A blown-out candle releases CO. However, candles are usually lit in well-ventilated areas, so the levels are negligible. By containing the gases in a jar, we can measure the CO levels. This is because candles without enough oxygen can emit CO rather than CO_2 [17]. ● Light a candle and place a jar over it to put it out. ● Place the CO sensor and the pocket carbon monoxide alarm in the jar for comparison. ● Confirm that the CO sensor and pocket carbon monoxide alarm readings match within 1 ppm.

Table 4: R&V Table for CO Sensor

2.3.2 App Subsystem

The app allows the user to remotely monitor their car's interior through the camera in real-time. This is done through wireless communication (WiFi) between the microcontrollers and the app. 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 the car exhaust be changed.

Requirements	Verification
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.
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.

Table 5: R&V Table for App Subsystem

After further testing, we discovered that our proposed 1 km range [18] might not be feasible. Thus, we have adjusted the value to reflect the limitations of the ESP32 WiFi more accurately. If this project is commercially produced, it is possible to extend the range given its own data plan.

2.3.3 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.

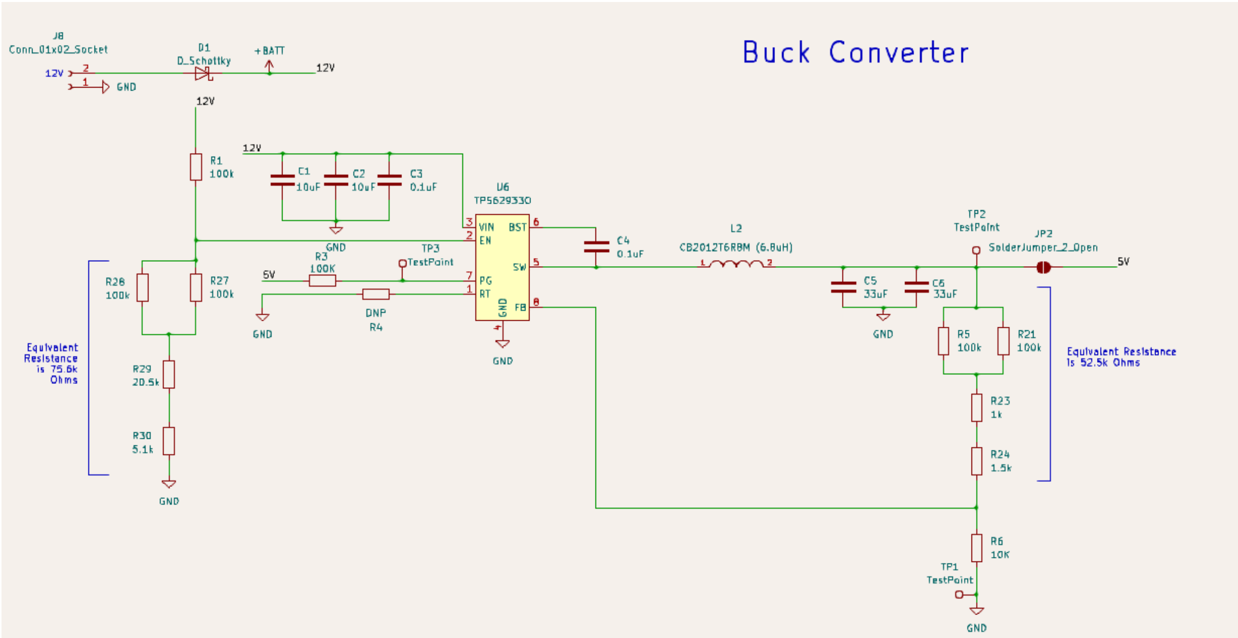

Figure 10: Buck Converter for 12 to 5 volts

For our main PCB, which sits on the car door, the input voltage is 12 volts. This voltage comes from the car battery and is considered the car's resting voltage. For demonstration purposes, a power adapter is used to supply the 12 volts through the sockets. The carbon monoxide and proximity sensors require five volts; thus, a voltage converter converts the 12 volts to five volts. A buck converter is chosen rather than a linear regulator because the change in voltage is significant. This results in considerable heat dissipation, requiring more complex circuitry. The base schematic for the buck converter is obtained from the Illinois Wiki [19].

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 [20].

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

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.

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 [20]. 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).

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

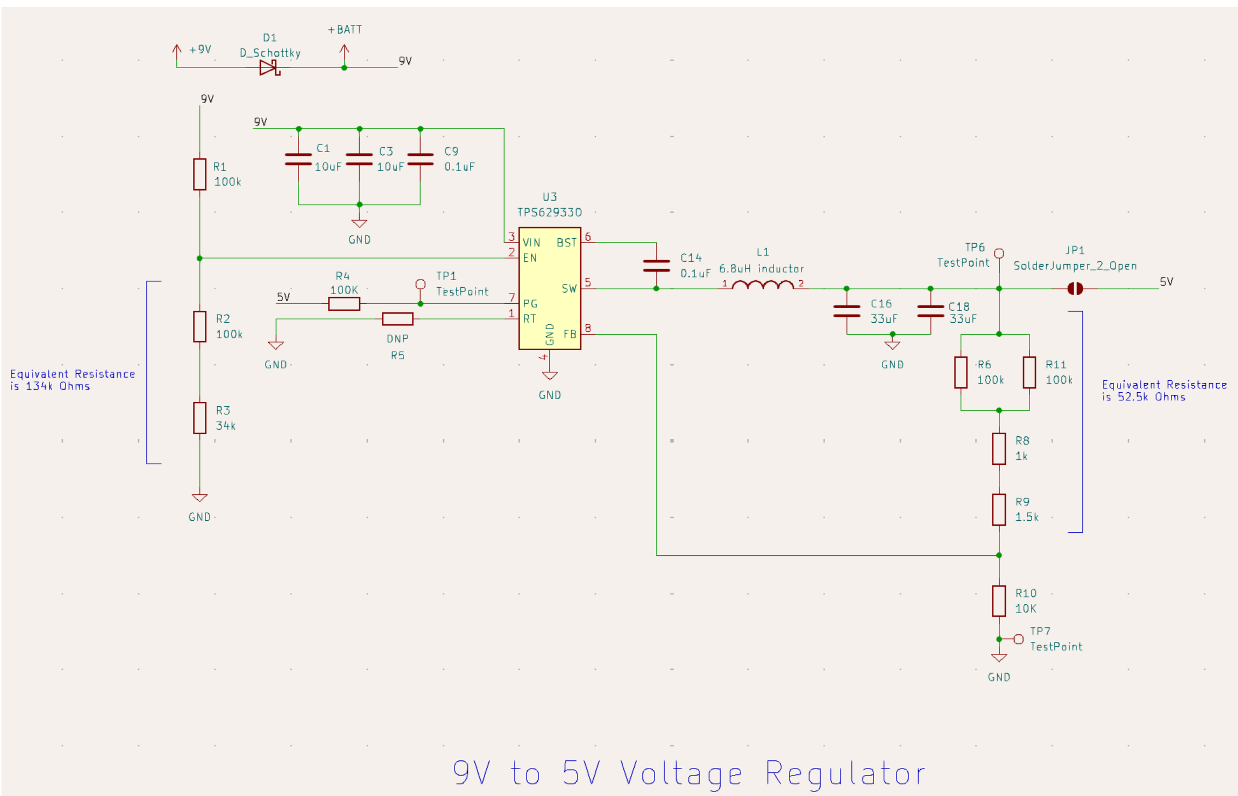

Figure 11: Buck Converter for 9 to 5 volts

The communication module PCB (second PCB) will be powered by a nine-volt lithium battery. The camera requires five volts to be powered. 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.

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

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.

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 V to ensure the output voltage is always constant [20]. 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 4 below).

$$R_{FBT} = \frac{V_{OUT} - V_{REF}}{V_{REF}} * R_{FBB} = \frac{5\,V - 0.8\,V}{5V} * 10k = 52.5\text{ k}\Omega \quad (4)$$

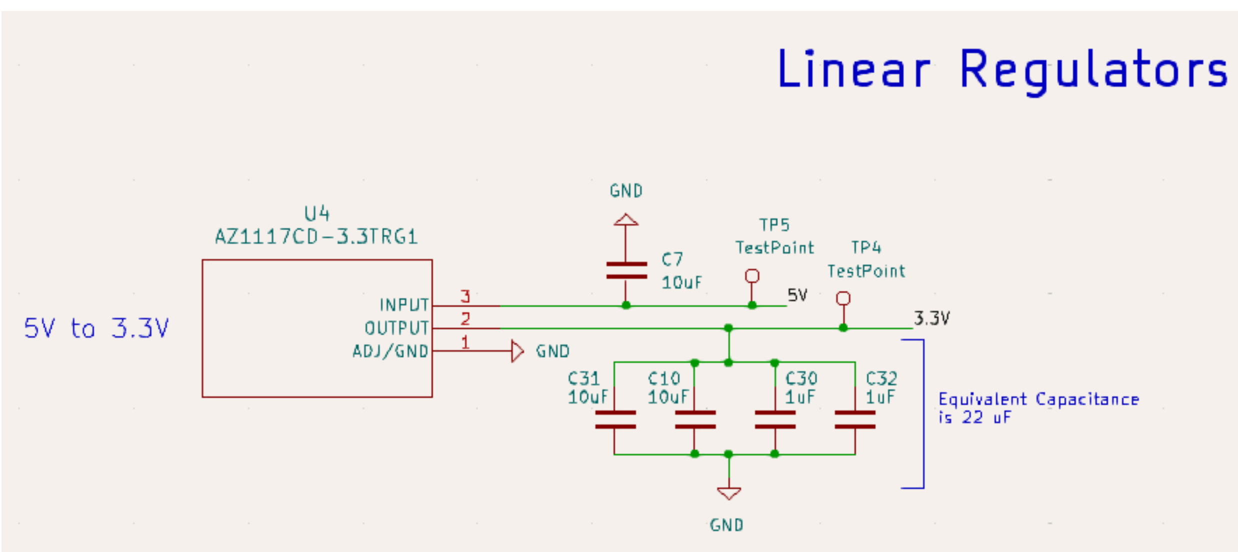

Figure 12: Linear Regulator for 5 to 3.3 volts

The temperature sensors 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 [21].

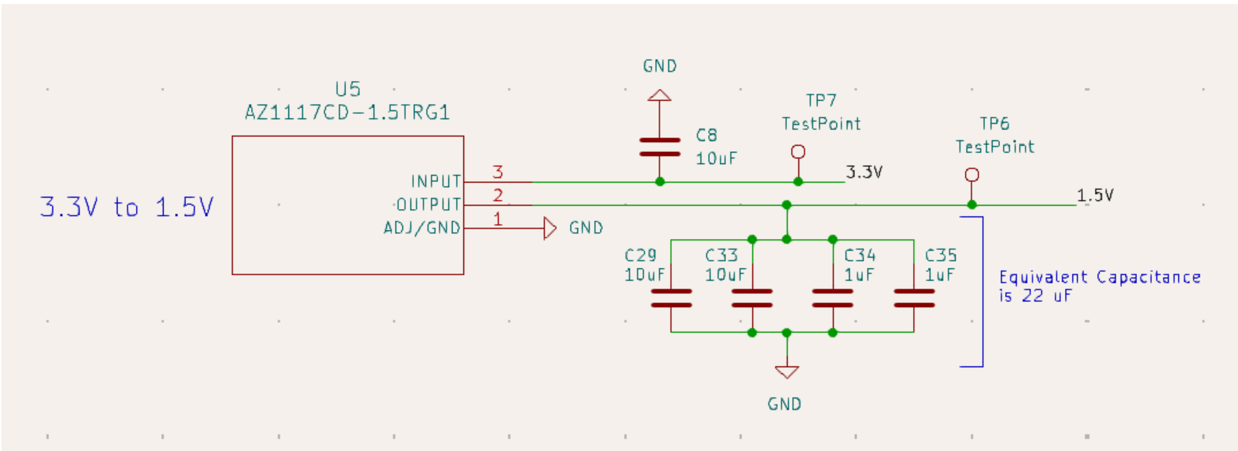

Figure 13: Linear Regulator for 3.3 to 1.5 volts

Since the speaker component 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 [21].

Requirements	**Verification**
12-to-5 Voltage Converter: ● Converts the 12 ± 0.5 V power source to 5 ± 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 ± 0.5 V. ○ The output of the 12-to-5 V converter should measure 5 ±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 ±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 ± 0.5 V.
5-to-3.3 Voltage Converter: ● Converts the 5 ± 0.5 V power source to 3.3 ± 0.5 V volts.	● A multimeter is used to check the voltage across the voltage converter. The input should read 5 ± 0.5 V and the output pins should read 3.3 ± 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 ± 0.5 V volts. ○ The voltage input to the temperature sensors measures 3.3 ± 0.5 V volts. ○ The voltage input to the weight sensors reads 3.3 ± 0.5 V. ○ The proximity sensor signal output reads 3.3 ± 0.5 V.
3.3-to-1.5 Voltage Converter: ● Converts 3.3 ± 0.5V volts power source to 1.5 ± 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 ± 0.5 V. ○ The output of the linear regulator measures 1.5 ± 0.5 V. ○ The input to the speaker components measures 1.5 ± 0.5 V.

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

Table 6: R&V Table for Temperature Sensor

2.3.4 Microcontroller

There will be two ESP32-S3 WROOM microcontrollers within our system.

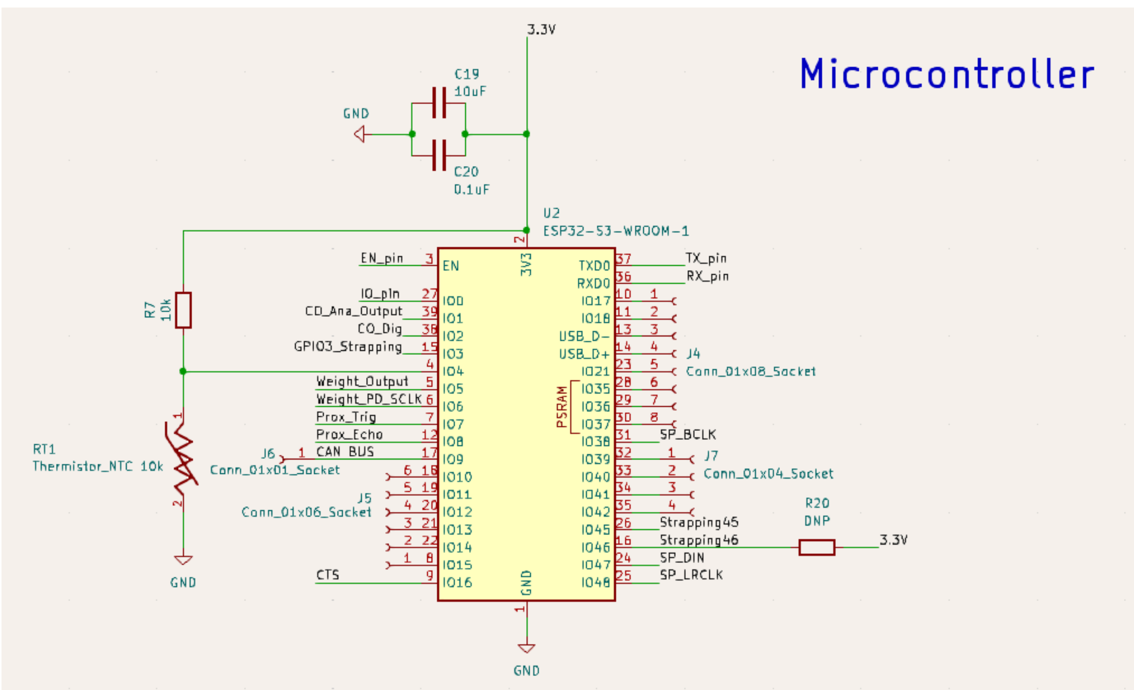

Figure 14: ESP32-S3-WROOM Microcontroller for Main PCB

The first microcontroller (ESP32-S3-WROOM) manages the values from the sensors and computes the necessary actions. The microcontroller is powered by 3.3 volts from the 5-to-3.3 voltage converter.

The microcontroller reads the temperature sensor through GPIO pin 4. The microcontroller is programmed to convert the signals from the thermistor into comprehensible temperature readings. From these values, the microcontroller checks if the measured temperature is greater than or equal to the threshold. If so, the microcontroller sends a signal to the CAN Bus to lower the windows. The microcontroller also sends a notification via the app to notify the user that the windows are lowered and the temperature in the car is too high.

For the weight sensor, when the value is above the threshold, the microcontroller records that passengers are present in the car. This allows the system to check if the windows are lowered when the temperature exceeds the set threshold. The system does not check the temperature if no one is in the car since we do not want the windows to lower in an empty vehicle.

The carbon monoxide sensor sends the level of CO in the vehicle in ppm to the microcontroller through GPIO 1 (CO_Ana_Output) and GPIO 2 (CO_Dig). CO_Ana_Output comes from the CO sensor. The microcontroller compares this value to the threshold. If the value is higher than the threshold, then the microcontroller will have the speaker announce that the CO levels are high. The microcontroller also sends a notification to the app to notify the user that their car's CO levels are too high and they need to get their exhaust checked.

Lastly, the proximity sensor signals the microcontroller through GPIO pins 7 (Prox_Trig) and 12 (Prox_Echo). The microcontroller uses this signal to confirm that the windows are lowered. It also stops the windows from continuing to lower and notifies the app. If this fails, the microcontroller will alert the user that the temperature is high, but the windows cannot be lowered. This will be done by keeping track of a flag: when the CAN Bus signals the windows to open, a flag will be set to one to denote that it is lowering. When the flag equals one and the proximity sensor no longer detects the window in front of it, the flag will be set to zero, confirming that the window is lowered.

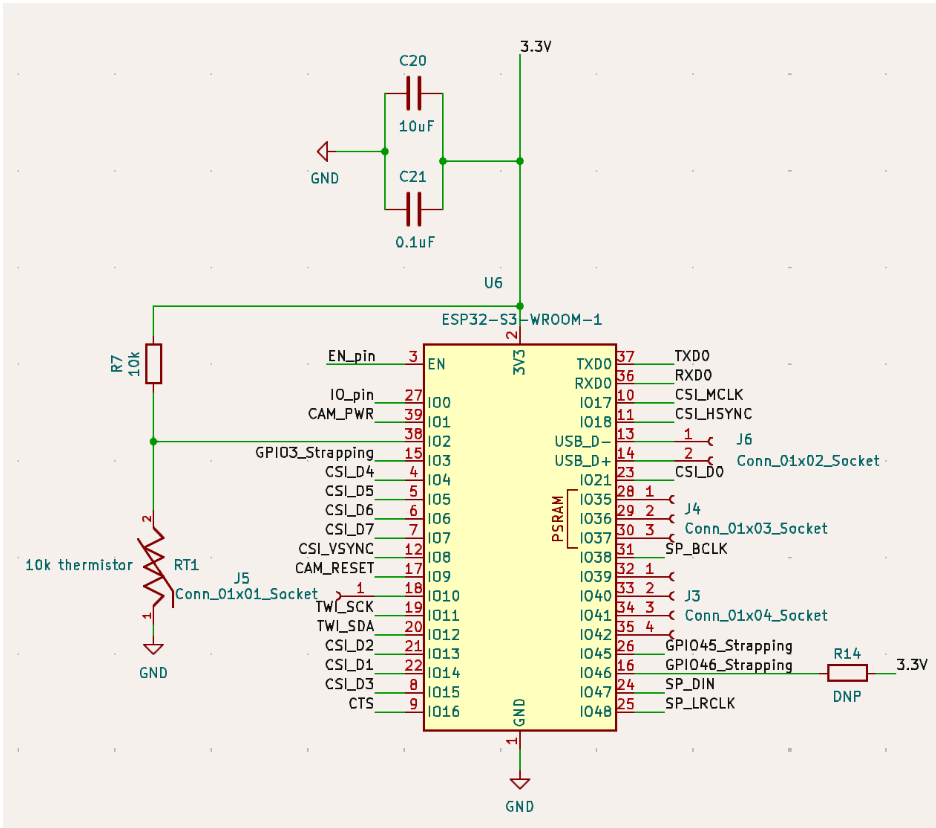

Figure 15: ESP32-S3-WROOM Microcontroller for Communication PCB

The secondary microcontroller manages the data from the camera and the speaker module and is powered by a 5-to-3.3 voltage converter. It takes values from the OV2640 lens and streams the camera footage to the app over WiFi. Furthermore, this microcontroller transmits audio data to an external speaker (CVS-1508) within the car. Lastly, the microcontroller takes in readings from a second temperature sensor, which monitors the internal temperature of the communication apparatus. This ensures that none of the components (especially the battery) overheat.

GPIO pin 2 contains the temperature sensor, which maintains a safe temperature within the communication subsystem. The sensor is programmed to notify the user once the battery temperature exceeds 100℉. The notification sent to the app tells the user that the battery temperature in the car is too high and should be replaced.

GPIO pins 4, 5, 6, 7, 8, 9, 10, 11, 12, 17, 19, 20, 21, 22, and 23 are the data protocol lines that connect to the FFC Connector to communicate with the camera. GPIO pins 24, 25, and 31 communicate with audio amplifiers via the DIN, BCLK, LRCLK pins. GPIO pin 39 hooks up to the camera power.

Requirements	**Verification**
The voltage reading across the microcontrollers is 3.3 $\pm$ 0.5 volts.	● A multimeter is used to confirm that the input voltage is around 2.8 - 3.8V. ○ Place the ground pin of the multimeter to GND. ○ Place the power pin of the multimeter on the 3.3 volt pin.
The ESP 32-S3 WROOM microcontroller waits a maximum of two minutes for the proximity sensor to confirm the windows have been lowered.	● Once the signal to lower the windows has been sent, a timer is used to confirm that the microcontroller waits a maximum of two minutes for the signal from the proximity sensor.
When the ESP32-S3 WROOM receives a value of nine or more ppm, it will warn the app of abnormal CO levels.	● When the CO sensor reads nine or more ppm, a notification is sent to the app, conveying that the levels are too high.
When the ESP32-S3 WROOM receives a temperature of 85 ℉ or greater, the windows will lower.	● Check the temperature sensors read 85℉ or greater. ○ The window should begin lowering.
When the ESP32-S3 WROOM receives that there are no objects detected in front of it 2 $\pm$	● The proximity sensor sends the microcontroller a signal confirming that the windows were lowered.

0.5 inches away, the windows will stop lowering.	● The windows should be lowered to at least the threshold level. This is confirmed with a measuring tape.
When the ESP32-S3 WROOM receives that there are 40+ lbs, it allows the window-lowering system to activate.	● The weight sensor sends the microcontroller a signal stating that the weight threshold has been met - indicating a person is in the car. The flag that indicates that a person is in the car is set. ● The microcontroller then checks the temperature in the car and deploys any necessary systems.
The ESP32-S3 WROOM will send a signal to the CAN Bus to lower the windows.	● Check the temperature sensors read 85℉ or greater. ● Check that the flag to lower the windows has been set. ● The window should begin lowering.
The video feed of the ESP32-S3 WROOM (communication PCB) microcontroller sent to the application has a maximum of a five second delay.	● Use a timer to check the video feed sent from the camera to the app has a maximum of a five second delay.

Table 7: R&V Table for Microcontrollers

2.3.5 Monitoring and Communication Subsystem

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

This system is powered by the Power Subsystem. This camera is powered by a nine volt battery that goes into a 9-to-5 voltage regulator since the voltage reading across the camera will be five volts. The speaker is powered by a 5-to-3.3 voltage converter chained with a 3.3-to-1.5 voltage converter since the voltage reading across the speaker is around 1.5 volts. A second temperature sensor is used to regulate the battery's temperature since the system is located in a heated setting simulating the environment of a hot car.

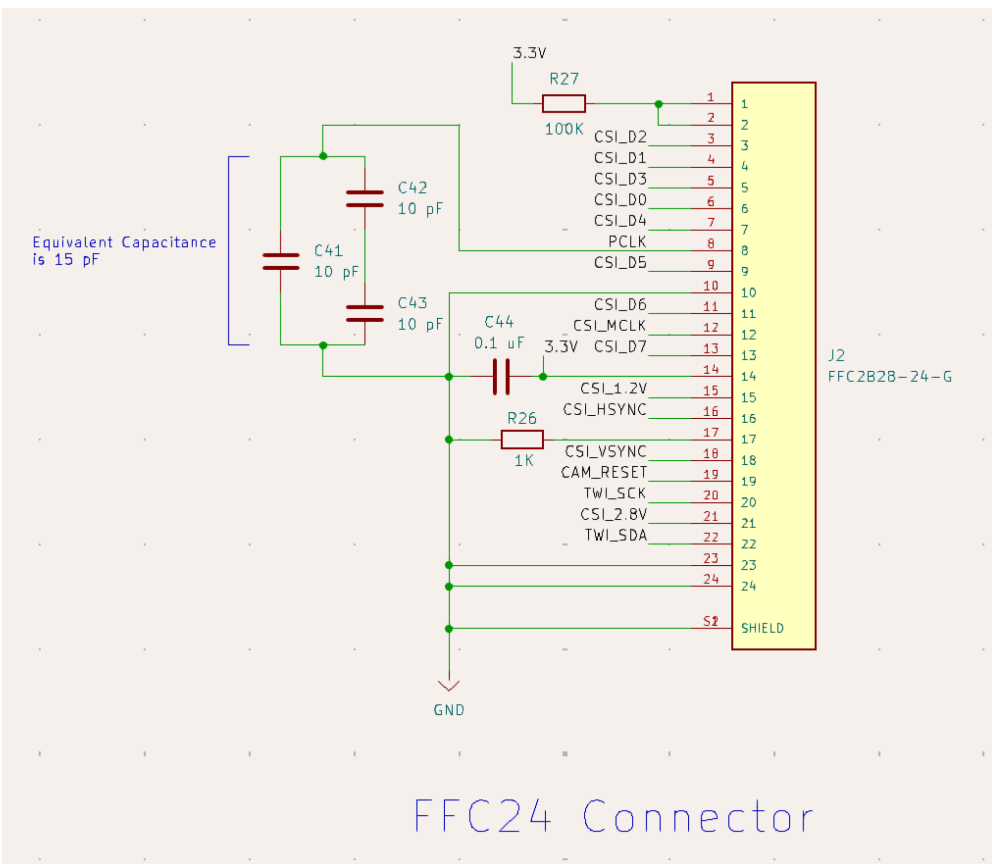

Figure 16: FFC-24 Connector

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

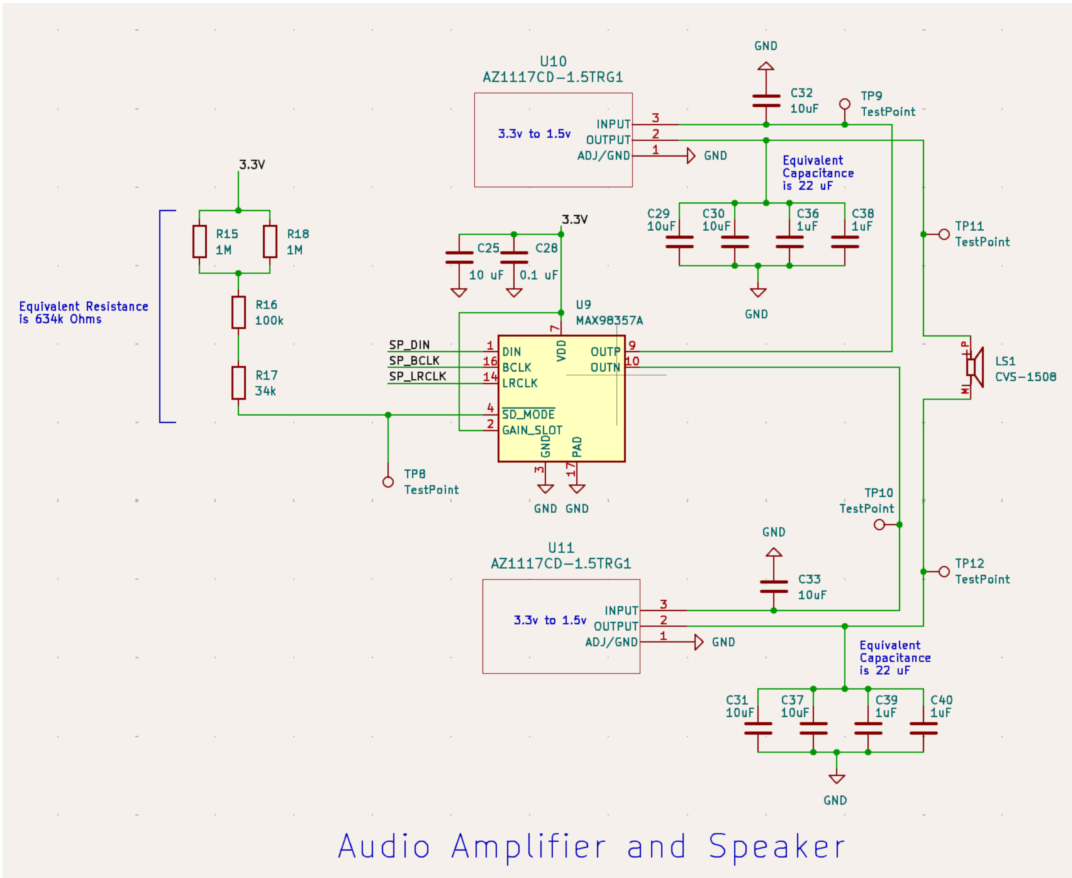

Figure 17: Audio Amplifier and Speaker

The Audio Amplifier accepts I2S data protocol from the microcontroller via the SP_DIN, SP_BCLK, and SP_LRCLK pins. Serial data is inputted 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 [22]. Pins 9 and 10 pass through a 3.3-to-1.5 voltage regulator to connect to the speaker.

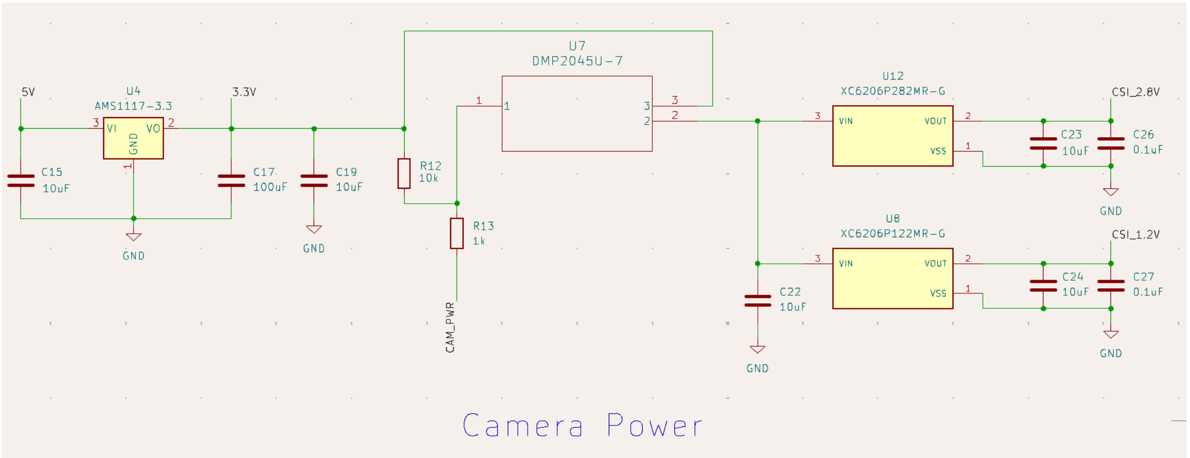

Figure 18: Camera Power

The Camera Power utilizes a 3.3-to-1.5 voltage regulator, P-channel MOSFET, 2.8V linear regulator, and 1.2 V 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 [23], while the code to activate ESP32-S3's WiFi is referenced here [24].

Requirements	**Verification**
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.
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.

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.
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.
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.
The audio output of the speaker will measure 73 dBA $\pm$ 3 dBA [25].	● A sound is played, and a sound level meter is used to confirm the audio output level is 70 - 76 dBA.

Table 8: R&V Table for Communication Subsystem

2.4 Tolerance Analysis

One of the most important aspects of our project is the Carbon Monoxide Sensor. This sensor measures the level of carbon monoxide that is in the car when it is running. Since CO is an odorless and colorless gas, it checks that the user does not have a faulty engine and that the car is safe to drive in.

There are two requirements to ensure that the CO sensor is correctly functioning: the sensor is preheated ahead of time, and the voltage cycles between 1.5 and 5 volts [5, Table.1], [14]. The sensor measures the level of carbon monoxide in the environment during the 1.5 volt cycle. These cycles can be done with a Pulse Width Modulation (PWM) [14], [26].

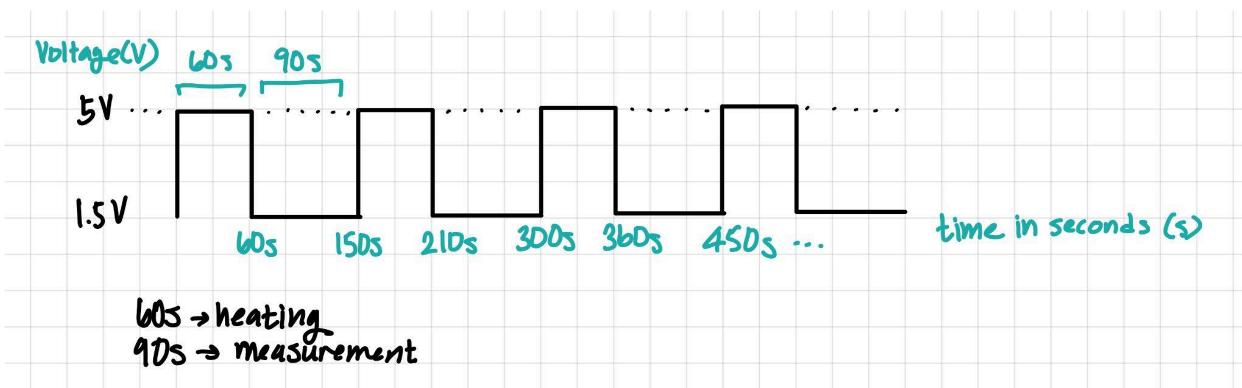

Figure 19: Graph of the Pulse Wave Modulation for How the Voltage of CO Cycles

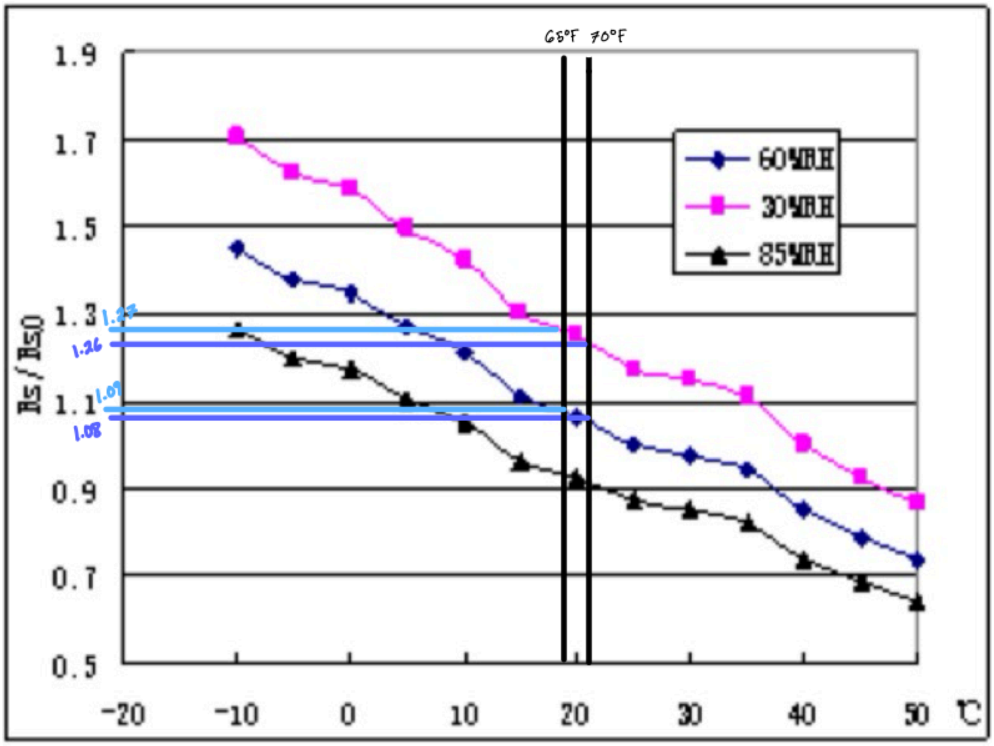

Figure 20: Graph of the Ratio of Resistance of the Sensor Based on Temperature and Humidity. [14, Fig.4] and annotations done by Parvati Menon

The MQ-7 datasheet [14, Fig.4] provides the range of resistances the sensor will have under different temperature and humidity conditions. This is displayed as a ratio (Rs/Rso), with Rs being the resistance of the MQ-7 sensor under adjusted conditions, while Rso is the resistance under the manufacturer's standard testing setting. Thus, the approximate resistance of the sensor in our system can be calculated.

Since carbon monoxide is a potential threat that can only happen when the car is on with the exhaust running, the average condition of the MQ-7 will be in AC. This means that the temperature will most likely be around 65 - 70℉ (18.33 - 21.11℃), a comfortable temperature for people to be in. Additionally, the average humidity level inside a running car is around 27.5% - 49.3% [27]. Thus, humidity levels between 30% and 60% are analyzed.

Temperature	Humidity	Rs/Rso (resistance ratio)	Resistance
65℉ (18.33 ℃)	30%	1.27	$36.83 \pm 3.81\ \Omega$
70 ℉ (21.11 ℃)	30%	1.26	$36.54 \pm 3.78\ \Omega$
65 ℉	60%	1.09	$31.61 \pm 3.27\ \Omega$
70 ℉	60%	1.08	$31.32 \pm 3.24\ \Omega$

Table 9: Resistance across CO sensor based on the Temperature and Humidity

Given the Rso value of $29 \pm 3\ \Omega$ [14, Table.1], we calculated the resistance across the CO sensor for the upper and lower bounds of the temperature and humidity.

For 65℉ and 30% humidity:

$$\frac{Rs}{Rso} = \frac{Rs}{29\ \Omega} = 1.27 \rightarrow Rs = 36.83 \pm 3.81\ \Omega \tag{5}$$

For 70℉ and 30% humidity:

$$\frac{Rs}{Rso} = \frac{Rs}{29\ \Omega} = 1.26 \rightarrow Rs = 36.54 \pm 3.78\ \Omega \tag{6}$$

For 65℉ and 60% humidity:

$$\frac{Rs}{Rso} = \frac{Rs}{29\ \Omega} = 1.09 \rightarrow Rs = 31.61 \pm 3.27\ \Omega \tag{7}$$

For 70℉ and 60% humidity:

$$\frac{Rs}{Rso} = \frac{Rs}{29\ \Omega} = 1.08 \rightarrow Rs = 31.32 \pm 3.24\ \Omega \tag{8}$$

At 1.5V, the CO sensor is sensing CO levels.

Equations 5-7 are calculated for 65℉ and 30% humidity.

$$P = \frac{V^2}{R} = \frac{1.5^2}{33.02\ \Omega} = 0.068\ W \tag{9}$$

$$P = \frac{V^2}{R} = \frac{1.5^2}{36.83\ \Omega} = 0.061\ W \tag{10}$$

$$P = \frac{V^2}{R} = \frac{1.5^2}{40.64\ \Omega} = 0.055\ W \tag{11}$$

Equations 8-10 are calculated for 70℉ and 30% humidity.

$$P = \frac{V^2}{R} = \frac{1.5^2}{32.76\ \Omega} = 0.069\ W \tag{12}$$

$$P = \frac{V^2}{R} = \frac{1.5^2}{36.54\ \Omega} = 0.062\ W \tag{13}$$

$$P = \frac{V^2}{R} = \frac{1.5^2}{40.32\ \Omega} = 0.056\ W \tag{14}$$

Equations 11-13 are calculated for 65℉ and 60% humidity.

$$P = \frac{V^2}{R} = \frac{1.5^2}{28.34\ \Omega} = 0.079\ W \tag{15}$$

$$P = \frac{V^2}{R} = \frac{1.5^2}{31.61\ \Omega} = 0.071\ W \tag{16}$$

$$P = \frac{V^2}{R} = \frac{1.5^2}{34.88\ \Omega} = 0.065\ W \tag{17}$$

Equations 14-16 are calculated for 70℉ and 60% humidity.

$$P = \frac{V^2}{R} = \frac{1.5^2}{28.08\ \Omega} = 0.080\ W \tag{18}$$

$$P = \frac{V^2}{R} = \frac{1.5^2}{31.32\ \Omega} = 0.072\ W \tag{19}$$

$$P = \frac{V^2}{R} = \frac{1.5^2}{34.56\ \Omega} = 0.070\ W \tag{20}$$

At 5V, the CO sensor stops sensing CO levels.

Equations 17-19 are calculated for 65℉ and 30% humidity.

$$P = \frac{V^2}{R} = \frac{5^2}{33.02\ \Omega} = 0.757\ W \tag{21}$$

$$P = \frac{V^2}{R} = \frac{5^2}{36.83\ \Omega} = 0.679\ W \tag{22}$$

$$P = \frac{V^2}{R} = \frac{5^2}{40.64\ \Omega} = 0.615\ W \tag{23}$$

Equations 20-22 are calculated for 70℉ and 30% humidity.

$$P = \frac{V^2}{R} = \frac{5^2}{32.76\ \Omega} = 0.763\ W \tag{24}$$

$$P = \frac{V^2}{R} = \frac{5^2}{36.54\ \Omega} = 0.684\ W \tag{25}$$

$$P = \frac{V^2}{R} = \frac{5^2}{40.32\ \Omega} = 0.620\ W \tag{26}$$

Equations 23-25 are calculated for 65℉ and 60% humidity.

$$P = \frac{V^2}{R} = \frac{5^2}{28.34\ \Omega} = 0.882\ W \tag{27}$$

$$P = \frac{V^2}{R} = \frac{5^2}{31.61\ \Omega} = 0.791\ W \tag{28}$$

$$P = \frac{V^2}{R} = \frac{5^2}{34.88\ \Omega} = 0.717\ W \tag{29}$$

Equations 26-28 are calculated for 70℉ and 60% humidity.

$$P = \frac{V^2}{R} = \frac{5^2}{28.08\ \Omega} = 0.890\ W \tag{30}$$

$$P = \frac{V^2}{R} = \frac{5^2}{31.32\ \Omega} = 0.798\ W \tag{31}$$

$$P = \frac{V^2}{R} = \frac{5^2}{34.56\ \Omega} = 0.723\ W \tag{32}$$

The average car battery capacity is 48 amp hours [28], and the maximum number of hours it is safe for one to drive continuously without a break is 8.5 hours [29]. We are considering the worst-case scenario. To support this drive, the car battery supplies 5.647 amps.

$$I = \frac{48\ amps*hrs}{8.5\ hrs} = 5.647\ amps \tag{33}$$

When our sensor operates at 1.5 volts, the minimum power consumption is 0.055 watts, and the maximum power consumption is 0.080 watts. The resulting current is 0.037 amps and 0.053 amps, respectively. These two values are less than 5.647 amps, thus, while the sensor is checking the CO levels in the car, it is not consuming more current than the car battery supplies.

At 1.5 volts, using minimum power

$$I = \frac{P}{V} = \frac{0.055\ W}{1.5\ V} = 0.037\ amps \tag{34}$$

At 1.5 volts, using maximum power

$$I = \frac{P}{V} = \frac{0.080\ W}{1.5\ V} = 0.053\ amps \tag{35}$$

When our sensor operates at 5 volts, the minimum power consumption is 0.615 watts and the maximum power consumption is 0.890 watts. The resulting current is 0.123 amps and 0.890 amps, respectively. These two values are less than 5.647 amps. Thus, while the sensor is in the heating cycle, it is not consuming more current than the car battery supplies.

At 5 volts, using minimum power:

$$I = \frac{P}{V} = \frac{0.615\ W}{5\ V} = 0.123\ amps \tag{36}$$

At 5 volts, using maximum power:

$$I = \frac{P}{V} = \frac{0.890\ W}{5\ V} = 0.178\ amps \tag{37}$$

Considering the worst-case scenario time that the car is on, the carbon monoxide sensor draws a very low current from the car battery. This sensor is safe to use within our system since it draws a small amount of current to operate and does not cause the car battery to excessively lose charge.

3 Cost Analysis

3.1 Cost Analysis

The total cost of purchasing the components for prototyping and constructing the final product is $193.76. Including shipping, the cost is estimated to be roughly $209.21.

Description	Manufacturer	Part Number	Quantity	Cost	Link
HX711 with 4pcs 50kg Load Cell Half Bridge Strain Gauge	Nextion	SC902 (Load Cell Half Bridge Strain Gauge) HX711 (ADC chip)	1	8.99	Link
CO Detector	Shenzhen Weijia Security Technology Co.	WJ-CO997	1	9.99	Link
9V Lithuim Battery	Voniko	CR-V9	2	13.99	Link
12 Volt Adapter 3A	Alitove	N/A	1	9.99	Link
12V to 5V Buck Breakout Board	ACEIRMC	16528	2	8.99	Link
5V to 3.3V Voltage Regulator	Shutao	13397-1	15	7.99	Link
3.3V to 1.5V Voltage Regulator	MECCANIX ITY	mea220429ee1304	5	6.69	Link
ESP32-CAM	HiLetgo	ESP32-CAM, OV2640	2	18.49	Link
micro-USB cable	Monoprice	104867	2	3.98	Link
USB-C to USB	JXMOX	4334964235	2	5.99	Link

ESP32 S3	AYWHP	WROOM-1-N16R8	1	10.99	Link
Thermistors 10k NTC	Cantherm	MF52A2103J3470	5	1.37	Link
12V to 5V Chip	Texas Instruments	TPS62933DRLR	3	3.12	Link
3.3V to 1.5V Chip	Diodes Incorporated	AZ1117CD-1.5TRG1	4	5.84	Link
Camera Connector	GCT	FFC2A32-24-T	1	0.64	Link
PCB Speaker	Same Sky	CVS-1508	4	9.00	Link
P channel MOSFET (PCB)	UMW	S8550	2	0.36	Link
Audio Amplifier	SparkFun	MAX98357A	1	6.50	Link
Carbon Monoxide Sensor	Winsen	MQ-7	2	11.00	Link
HX711	WWZMDiB	HX711	4	$6.99	Link
Relay Switch	KEMET	EE2-5NU-L	2	$4.16	Link
MQ-7 Development Board (For breadboard)	ACEIRMC	MQ7	5	$11.99	Link
PCB Thermistor	TDX Corporation	NTCG203NH103JT1	5	$0.95	Link
NTC 10k Breadboard	PATIKIL	MF52103	20	$6.29	Link
XC6206-1.2 V	Torex Semiconductor Ltd	XC6206P122MR-G	2	$1.24	Link

XC6206-2.8 V	Torex Semiconductor Ltd	XC6206P282MR-G	2	$2.26	Link
AMS1117-3.3 Linear Regulator	UMW	AMS1117-3.3	2	$1.24	Link
Mini Hygrometer & Thermometer	Shenzhen Yongsheng Innovation Technology Co., Ltd	A01-2 Pack	2	$7.99	Link
Thermal Insulation Pads	Outus	N/A	1	$7.99	Link

Table 10: Cost of Materials

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 [30]. 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 will spend roughly 12 weeks on the project. Using this equation to calculate one partner's labor costs:

$$(\$/hour) * 2.5 * hours\ to\ complete = TOTAL \tag{38}$$

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

$$(\$52/hour) * 2.5 * 40 * 12 = \$62,400 \tag{39}$$

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

$$\$62,400 * 3 = \$187,200 \quad (40)$$

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

$$(\$52/hour) * 2.5 * 40 = \$5,200 \quad (41)$$

Thus, the cost for two staff members will be:

$$\$5,200 * 2 = \$10,400 \quad (42)$$

The grand total will be for this project would be:

$$\$209.21 + \$187,200 + \$10,400 = \$197,809.21 \quad (43)$$

3.2 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) • Team Contract Assessment (All team members) • Demo Preparation (All team members)
4/21	• Finalize PCB and Testing (All team members) • Demo Preparation (All team members) • Mock Demo (All team members) • Work on Final Paper (All team members)
4/28	• Demo and Presentation Preparation (All team members) • Final Demo (All team members) • Mock Presentation (All team members) • Work on Final Paper (All team members)
5/5	• Final Presentation (All team members) • Final Paper (All team members)

Table 11: Semester Schedule

4 Ethics and Safety

Our project follows the IEEE Code of Ethics [31].

Ethical Concerns: Leaving your child or pet in a locked, hot car is illegal and dangerous. This project is in no way promoting or condoning this behavior. Instead, it is a safety measure to mitigate any potential heat strokes or deaths in the case that a child/pet is left in the hot car. Our project strives to "comply with ethical design and sustainable development practices" as defined by the IEEE Code of Ethics 7.8.I.1 [31].

Concerning electrical safety:
Battery Safety: We are using a nine volt lithium battery due to its durability in extreme temperatures and stability for power. We have thermal insulation pads to ensure the battery does not overheat. We also have a divider to prevent contact with the other components. In addition, we will "hold paramount the safety, health, and welfare" as defined by the IEEE Code of Ethics 7.8.I.1 [31].

Data Privacy: Our device asks the user for consent to record them. This data is not used anywhere else. This is "to protect the privacy of others and to disclose promptly factors that might endanger the public or the environment," as defined in the IEEE Code of Ethics 7.8.I.1 [31].

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 [32]. 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 [31].

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 [31].

Concerning heat safety:

Component Heat Protection: Our system is built to last in high heat conditions since it is meant to sit in a hot car. We will place thermal insulation pads to protect the heat-sensitive components. This ensures the components are not damaged by excessive heat, which will "hold paramount the safety, health, and welfare," as defined by the IEEE Code of Ethics 7.8.I.1 [31].

All group members have completed the required lab safety training before beginning the project. Additionally, we have read the "General Battery Safety" document [33], which outlines the safety measures for using batteries to power our system.

Concerning CO safety:

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 eight hour average area level" [34].

5 Code Repository

The code used in this project can be found in this repository.

References

[1] "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/

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

[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] 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

[6] Digikey, "SEN-24049." 2025. Accessed: Mar. 05, 2025. [Online]. Available: https://www.digikey.com/en/products/detail/sparkfun-electronics/SEN-24049/22258319?=&utm_adgroup=&utm_source=google&utm_medium=cpc&utm_campaign=PMax%20Shopping_Product_Low%20ROAS%20Categories&utm_term=&utm_content=&utm_id=go_cmp-20243063506_adg-_ad-__dev-c_ext-_prd-22258319_sig-Cj0KCQiAwOe8BhCCARIsAGKeD55DTjv9gZ26rOjM1WIFrbexaIXRYJ1tzdX1c1VNdn7GUpu1ihEBv28aAkuoEALw_wcB&gad_source=1

[7] "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/

[8] "SC902 Half Bridge Micro Load Cell Size." Accessed: Mar. 06, 2025. [Online]. Available: https://akizukidenshi.com/goodsaffix/SC902-50kg.pdf

[9] 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

[10] 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

[11] 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-38b04f309a1d

[12] N. Seidle. “SparkFun Load Sensor Combinator v11.” cdn.sparkfun.com. July 21, 2016. Accessed: Mar. 06, 2025. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/SparkFun%20Load%20Sensor%20Combinator%20v11.pdf

[13] S. Al-Mutlaq and Alex The Giant. “Load Cell Amplifier HX711 Breakout Hookup Guide.” learn.sparkfun.com. Accessed: Mar. 06, 2025. [Online]. Available: https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide

[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] 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/

[18] Espressif, “Wi-Fi Driver - ESP32-S3.” Espressif.com, 2020. Accessed: Mar. 06, 2025. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32s3/api-guides/wifi.html

[19] “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

[20] 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://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-7%20Ver1.3%20-%20Manual.pdf

[21] Diodes Incorporated. “AZ1117C. ” Sept. 2022. Accessed: Mar. 06, 2025. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-7%20Ver1.3%20-%20Manual.pdf

https://socialwifi.com/knowledge-base/wifi-technology/difference-between-24-and-5-ghz-wifi/

[22] 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

[23] 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

[24] “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/

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

[26] the_3d6, “Arduino CO Monitor Using MQ-7 Sensor.” Instructables. Accessed: Feb 13, 2025. [Online]. Available: https://www.instructables.com/Arduino-CO-Monitor-Using-MQ-7-Sensor/

[27] K. Gładyszewska-Fiedoruk and T. J. Teleszewski. “Experimental research on the humidity in a passenger car cabin equipped with an air cooling system - development of a simplified model,” in *Applied Thermal Engineering, vol. 220, p. 119783*, Feb. 2023, doi: https://doi.org/10.1016/j.applthermaleng.2022.119783.

[28] D. Hawley. “How Many Amps Is A Car Battery?” Jdpower.com. 2022. Accessed: Feb 13, 2025. [Online]. Available: https://www.jdpower.com/cars/shopping-guides/how-many-amps-is-a-car-battery

[29] S. Abramovich, “If You Drive More Than This Many Hours In One Day, You’re Putting All Road Trip Travelers At Risk.” Explore.com. Aug. 02, 2023. Accessed: Feb 13, 2025. [Online]. Available: https://www.explore.com/1352480/road-trip-drive-hours-per-day-safety-risk-travelers/

[30] 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

[31] 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

[32] 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/

[33] “Safe Practice for Lead Acid and Lithium Batteries.” April 13, 2016. Accessed: Feb 13, 2025.

[Online]. Available:
https://courses.grainger.illinois.edu/ece445/documents/GeneralBatterySafety.pdf

[34] "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