

Car Catalytic Converter Theft Prevention

Aditi Tyagi (aditit2)

Anushka Parikh (aparik28)

Shruthii Sathyanarayanan (ss57)

TA: Stasiu Chyczewski

Abstract

Catalytic converters are a part of a car's exhaust system and convert toxic gas to non-toxic gas from a car's engine. They are located in the undercarriage of the car and contain precious metals like platinum. Due to its exposed location and composition of precious metals, it has become a common target for theft during the pandemic. In fact, catalytic converter theft increased by 400% in the year 2021 alone. To prevent and capture evidence of the theft, we have proposed the following anti-theft solution. Our surveillance device consists of sensors, an alarm, and a camera. Additionally, the device can be controlled remotely by a car owner via a mobile interface. We have identified an optimal location for the placement of the device such that it detects any unusual activity and has a clear view of the catalytic converter. When human activity, like human motion or vibration/tilting of the car, is sensed by our device, the sensor subsystem will trigger the alarm and camera system. The alarm will draw attention to the suspicious activity occurring, and the camera will capture evidence of the theft. The controls and the live camera footage will be displayed on a mobile interface for the car owner. This report will discuss the design process, implementation, and verification of the product.

Table of Contents

1. Introduction	1
1.1 Problem	1
1.2 Solution	1
1.3 High-Level Requirements	1
2. Design and Verifications	2
2.1 Block Diagram	2
2.2 Subsystem Overview and Design	3
2.2.1 Power/Control Subsystem	3
2.2.2 Solid-State Relays	4
2.2.3 Sensor Subsystem	4
2.2.4 Alarm Subsystem	5
2.2.5 Camera Subsystem	5
2.3 User Interface	6
2.3.1 WiFi Module	6
2.3.2 Web Application	6
2.4 Subsystem Requirements and Verification	7
2.4.1 Power/Control Subsystem	7
2.4.2 Sensor Subsystem	8
2.4.3 Alarm Subsystem	10
2.4.4 Camera Subsystem	10
3. Cost and Schedule	11
3.1 Cost	11
3.2 Schedule	12
4. Conclusion	13
4.1 Accomplishments	13
4.2 Uncertainties	13
4.3 Ethics and Safety	14
4.4 Future Work	15
References	16
Appendix	18
Appendix A: Requirements and Verifications	18
A.1 Power/Control Subsystem	18
A.2 Sensor Subsystem	19
A.3 Alarm Subsystem	20
A.4 Camera Subsystem	21
Appendix B: First Order PCB Design	23
B.1 Circuit Design	23

B.2 PCB Design	24
Appendix C: Second Order PCB Design	25
C.1 Circuit Design	25
C.2 PCB Design	26
Appendix D: Revised PCB Design	27
D.1 Circuit Design	27

1. Introduction

1.1 Problem

A catalytic converter is a device on cars that converts toxic gas from the car's engine into non-toxic gas. This device uses precious metals like platinum, palladium, and rhodium to convert carbon monoxide, hydrocarbons, and nitrogen oxides to carbon dioxide, nitrogen, and water vapor. These metals are extremely rare and expensive, and given the increase in prices of metals in recent years, catalytic converters have become a prime target for thieves. Catalytic converters are located at the bottom of the car behind the front wheels, and can easily be removed or sawed off by someone in less than 10 minutes [1]. Additionally, catalytic converters are prime targets for thieves because they are hard to trace [2]. Once they have been stolen, catalytic converters can be sold as scrap metal for 150 to 200 dollars a piece.

In the past year alone, car catalytic converter theft has risen by 400% [1]. These thefts are a large inconvenience to many car owners. The most common targets for catalytic converter theft are SUVs, Prius, Tacoma, and the Accord based on the thefts reported in the past years [1]. In fact, the Toyota Prius has been a prime target because its hybrid nature results in a lower emission load compared to traditional gasoline vehicles. As a result, the catalytic converter in these cars does not need to work as hard and last longer, which in turn increases the value of the precious metals and scrap metal of the catalytic converter.

Driving without a catalytic converter disables the emission system in cars and is against the law in every state [3]. On average, the costs to replace catalytic converters can be as high as \$2500. According to an incident in Champaign County, the car owner paid \$1,900 for replacement and repairs [4]. Most victims end up having little evidence aside from stolen converters, and low quality video evidence.

1.2 Solution

According to experts, car owners can prevent car catalytic converter theft by adding a camera, motion sensors or flood lights to deter thieves [2]. We will implement this into our solution. Our solution is to create a surveillance device for catalytic converter theft. The device would be able to detect suspicious activity, give real time notifications to the owner when the crime is happening, and would be mounted discreetly. Another feature would be having a camera functionality to solve the low quality surveillance video issue, and setting off an alarm when the suspicious activity occurred.

To create this device, we will use a vibration or motion sensor to detect if a thief attempts to move the car when it is parked. This sensor will trigger the alarm system which will scare off the thief. Then, it will send a notification to the car owner's mobile phone or device. The sensor will also trigger the camera to try to capture the thief and create evidence for the theft.

1.3 High-Level Requirements

- 1) The device must be able to trigger the motion sensors (this includes the PIR and microwave sensors) within 5 to 15 seconds after the car has been tampered with. What

this accomplishes is that it allows the system to act quickly once the converter is being tampered with. The device must also trigger the motion sensors at least 4 out of 5 times after it has been tampered which gives the device an 80% accuracy rate.

- 2) After the motion sensors are activated, the system must trigger the alarms and send a notification to the user within 5 to 15 seconds. The camera must also start recording within 15 seconds after the sensor has been triggered.
- 3) The surveillance device must have a height between 2 to 4 inches, a maximum length of 6 inches, and maximum width of 8 inches. The height constraint was determined by the camera angle and the ideal camera position to be able to record the catalytic converter and a thief attempting to steal it. The length and width was determined by the space where the device will be placed.

2. Design and Verifications

2.1 Block Diagram

Our surveillance device works as follows: if the sensors in our device – the PIR Sensor and accelerometer – detect any human motion and/or vibration, the sensors get triggered. Once the sensors are triggered, within five to fifteen seconds, the alarm will get triggered, causing it to ring, and the camera will get triggered, and will begin recording live footage. As the camera records live footage, the user is able to see this live footage through the user interface. Our user interface shows the live feed from the camera, and it has buttons allowing the user to manually turn off the alarm and the sensors. All our subsystems are powered by our power subsystem, and all our subsystems interact with one another through our control system, which is made up of the ESP32-Wroom microcontroller. Below is a block diagram of our device and how our subsystems interact with one another and the user interface.

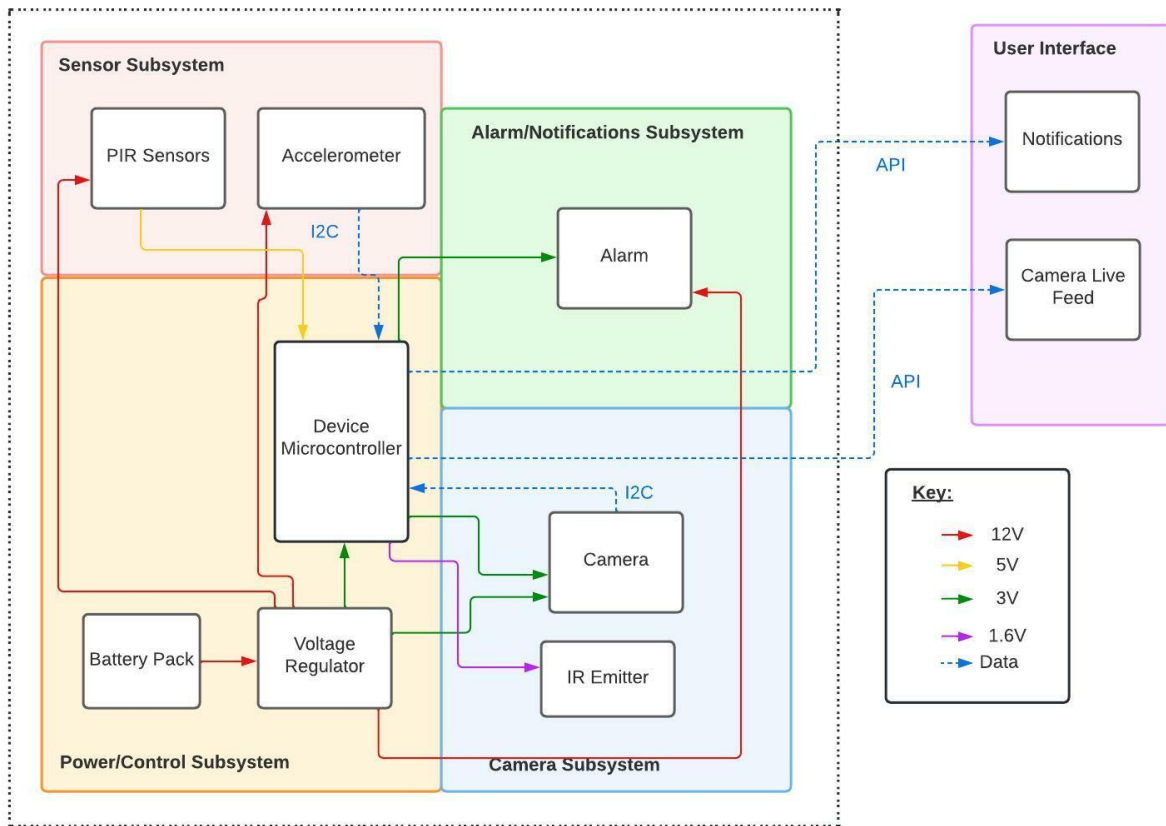


Figure 1: Block Diagram of the Solution

2.2 Subsystem Overview and Design

2.2.1 Power/Control Subsystem

The purpose of the Power/Control Subsystem is to be able to power the other adjoining subsystems and control the microcontroller. The power component comprises a 12 Volt Battery that powers the alarm subsystem, and two voltage regulators. We have a voltage regulator that converts the 12 Volt battery to 5 Volts for the PIR Sensor, and a voltage regulator that converts the 12 Volt battery to 3.3 Volts for the Camera and the Accelerometer.

The Power/Control Subsystem also includes the microcontroller and the components necessary for the microcontroller, such as two switches, one used to boot the microcontroller, and the other used to reset the microcontroller. A requirement for our device is that it must be able to send data to a user interface. For this project, we have decided to use WiFi to send the data, and by doing this, we will be able to stream videos and connect to a user interface that can show these videos.

After some research, we found the best option for the project is the ESP32-WROOM-32 [5]. We will be using this microcontroller as it has WiFi compatibility and can be used to stream video to an app [6]. This microcontroller also has modern, light, and deep sleep options which can be used when the car is in motion and to save battery.

In Figure 3, we connected the microcontroller with the necessary push buttons for the Enable and IO0. The microcontroller uses 3.3V which will be provided by the voltage divider. The microcontroller will be programmed with the CP2102 Adapter which uses a UART protocol. We plan to connect all IO pins to connectors, but this may change due to the size constraint on our device. The IO pins will be connected to the Sensor, Alarm, and Camera Subsystem, and it will be the primary communication between the subsystems.

2.2.2 Solid-State Relays

When designing the sensor, alarm, and camera subsystems, we made the design choice to incorporate solid state relays in our circuitry. Utilizing solid state relays in our subsystems allows us to control the power consumption of the subsystems and increase the longevity of our battery life. Solid state relays also establish complete electrical isolation between the input and output contacts when we attempt to remotely control/trigger the individual subsystems. This effectively blocks any surges in power and ensures subsystems aren't incorrectly or accidentally triggered. We considered mechanical relays as well as solid state relays, but decided to move forward with solid state relays due to several reasons. Solid state relays require a smaller trigger voltage and need negligible idle current. They are smaller and faster at switching than mechanical relays, which helps reduce the time taken to enable or disable our subsystems [7].

2.2.3 Sensor Subsystem

The Sensor Subsystem is made up of tilting/vibrations and motion sensors. The main purpose of this subsystem is to be able to sense any suspicious activity occurring, and the two sensors that we use to achieve this are the PIR Sensor and the accelerometer. The sensors will be triggered if a car that is in park is tilted or if any vibration is detected in the undercarriage.

The Passive Infrared (PIR) sensors are small, low power, easy to use, and inexpensive. The way it senses movement is by sensing the change in temperature between the background and a warm body [8]. In order to detect sense of movement and vibrations, we will use an accelerometer. An accelerometer is an electromechanical device used to measure acceleration forces in mainly x,y,z directions [9]. This basically means that if we were to tilt the accelerometer, it would display the x-coordinate it's been tilted to, the y-coordinate it has been tilted to, and the z coordinate it has been tilted to [10]. This goes the same for vibrations, for example if we were to place the accelerometer on the side of a box that is sitting on a table, and if the table were vibrating, we would expect to see change in coordinates in the x and y direction, while the z coordinate would remain constant due to the force of gravity.

For design purposes, the main reason why we chose the PIR Sensor to sense any movement around the device was due to the fact of its lower false positive rate compared to other sensors such as microwave sensors [8]. The main reason we chose the ADXL 343 accelerometer was because as we were researching, we were seeing that to solder an accelerometer on our PCB would be very difficult, so with the help and suggestion from our TA and professor, we found an accelerometer development board that has a connection to ground, a connection to voltage, a connection for the clock signal, and a connection for the data being created by the accelerometer [10].

Finally, we use the solid state relay in our circuit, and we designed our circuit such that the solid state relay can remotely enable/disable the circuit. This preserves the battery and reduces the false positive rate of the device.

2.2.4 Alarm Subsystem

The Alarm Subsystem has two functionalities: a physical alarm that is triggered upon detection of suspicious activity and a web application that can be used to remotely enable or disable the alarm. This subsystem utilizes a 120 dB, battery operated, voltage controlled alarm [11] and sends a notification to the car owner's phone via WiFi. The purpose of this subsystem is to set off an alarm and alert the owner if a thief attempts to steal a catalytic converter from a car. The Alarm Subsystem will be triggered by the Sensor Subsystem, i.e., when the vibration and tilting sensors are triggered. If the sensors are triggered for longer than a few seconds, the Sensor Subsystem will send a signal to the Alarm Subsystem, which will trigger the alarm to start ringing. Our subsystems are designed such that the user can determine how long the sensors should detect motion for before the alarm turns on.

The 120dB alarm will be attached to the undercarriage of the car alongside the remaining subsystems and is wear resistant, impact resistant, heat resistant, and low temperature resistant [11]. These features will help combat the various temperatures and wear and tear that the alarm will be exposed to. Additionally, the alarm will continue to function even if the car heats up or overheats. Another important feature is impact resistance, which ensures that the alarm will continue to ring (to a certain extent) even if a thief attempts to break or remove it. The alarm by default will ring for 3 to 5 minutes before shutting off. However, the owner can manually turn off the alarm sooner through their phone. The Alarm Subsystem is also designed such that the user can customize how long the alarm should ring before shutting off.

When the Alarm Subsystem is triggered, the web app will show that the alarm was enabled and will provide the user an option to disable the alarm. This function will be dependent on WiFi, as the microcontroller will use WiFi to communicate with the web application on the owner's phone, alerting them of potential theft in real time.

A solid state relay is used to turn the alarm on or off based on the input from the microcontroller. When the Sensor Subsystem is triggered, the input from the microcontroller to the Alarm Subsystem will increase, which then actuates the relay and turns on the alarm to notify the car owner of a potential catalytic converter theft. The alarm that will be used has a standard voltage of 9-12V and a standard current of 100 to 150mA.

2.2.5 Camera Subsystem

The purpose of the Camera Subsystem is to capture evidence of the theft when the sensors are triggered. This subsystem includes the camera, IR emitter, and the live footage. The camera subsystem will be triggered by the Sensor Subsystem. When the Sensor Subsystem detects motion and vibrations, it will send a signal to the Camera Subsystem which will turn on the camera and record footage.

For the control subsystem, we decided to use the ESP32 microcontroller. This microcontroller is compatible with Arduino IDE and uses WiFi. We decided to use the OV7670 camera as it is compatible with the ESP32-WROOM-32, the microcontroller.

In our original design, we built two circuit breakers to turn off the camera when the sensors are not activated as shown in Figure 3. For this circuit, we will be using a solid-state relay. When the microcontroller sends 3.3V, the relay will turn off and when the microcontroller sends 5V, the relay will turn on. We must also consider the data sent between the camera and the microcontroller. The OV7670 camera uses the I2C Interface, which also works with the ESP32-WROOM-32. The camera will send data back to the microcontroller and the microcontroller will use WiFi and API calls to send the video data to the user interface. The S10C and S10D use the I2C interface and we have connected them to circuit breakers to turn off and on the camera.

The OV7670 camera must also be connected to a clock at the XCLK input. This clock must be between 10 and 48 MHz [12]. Other outputs like D1 to D7, VSYNC and HREF will be connected directly to the microcontrollers IO connectors.

We tested a phone camera under a car to mimic our solution. During the day, there is enough light to set a camera under the car, but we will need more light for the night. The device must be discrete, so we decided to use an IR emitter. We connected the IR emitter to another circuit breaker to turn it on when the camera turns on.

2.3 User Interface

2.3.1 WiFi Module

The microcontroller was set up as a soft access point, which essentially allows any device with WiFi capability to connect to and remotely control our microcontroller. By setting up our microcontroller as an access point, we forgo the use of a router and directly create a WiFi network for the microcontroller. Since the WiFi range of our ESP32 microcontroller module is 1 kilometer, we chose to establish the network connection as a hotspot. When using this surveillance device, we assume that the owner of the car will be within this range to remotely control the device. However, this range can be significantly increased up to 10 kilometers when a high-quality antenna is incorporated into the design [13].

2.3.2 Web Application

Our web application contains buttons that can be used to enable and disable certain subsystems and live camera feed that will automatically stream footage when our device detects suspicious activity. As seen in the images below, there are buttons to enable or disable the sensor subsystem and the alarm subsystem. Additionally, the current state of the subsystems are also displayed to the user.

The web application was designed with the following use case in mind: If a user is driving and does not require the surveillance device to be on, they can simply turn the sensor detection off via the app and turn it back as and when they need. With this functionality, the user need not remove the device from the car every time they want sensor detection to be turned off. Similarly, if the user or car owner is near the car when suspicious activity is detected, they can override the duration for which the alarm rings by turning off the alarm through the app.

The images below also show how the live footage from the camera is displayed on the application once the Sensor Subsystem detects any unusual activity.

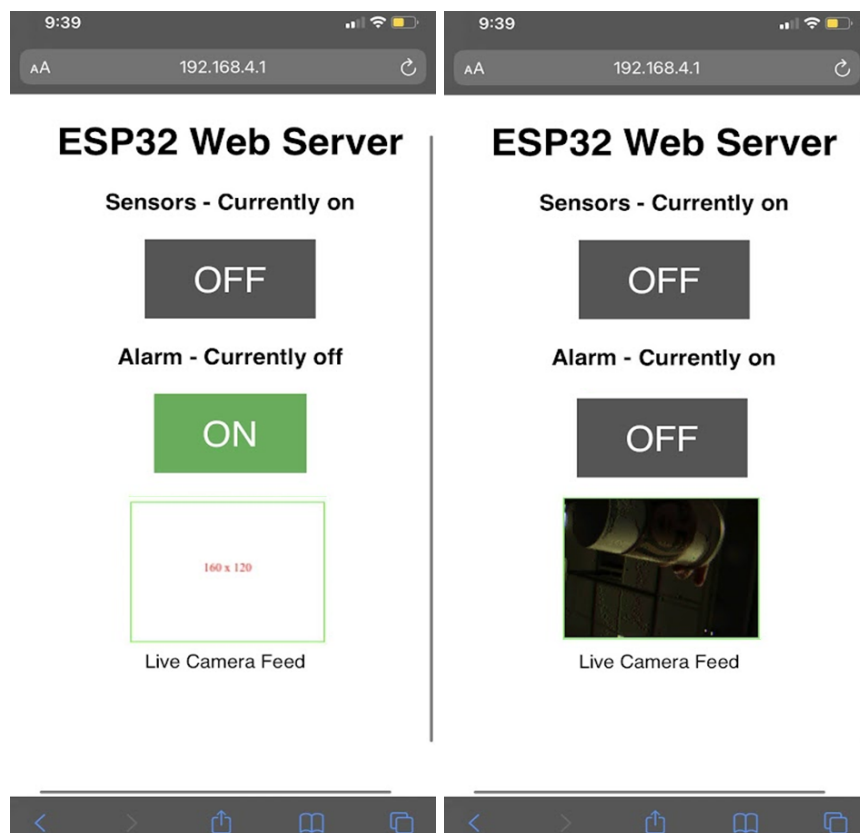


Figure 2: Screenshots of our Mobile Interface

2.4 Subsystem Requirements and Verification

2.4.1 Power/Control Subsystem

To verify the power subsystem, we had 3 main requirements. The power subsystem must supply 12V, 5V, and 3.3V as we will need these voltages to power all of the subsystems. We also had another requirement for the power subsystem to remain at a thermal temperature of 125°C, so that the system does not overheat.

First, we verified that the battery provides 12V. We used a multimeter to test the battery voltage. On average, the voltage provided 11.8V which is within our 12V (+- 5%) requirement. Next, we soldered on the voltage regulators on the PCB and connected our 12 Voltage battery. We used the multimeter to test the 5V and 3.3V regulators. We tested the 5V first, which outputted 5.05V on the multimeter. As we tested the 5V, we noticed the 3.3V voltage regulators started heating up. We quickly desoldered the voltage regulators and recreated the circuit on a breadboard.

We connected the pins of the regulator to header wires, and connected that to the breadboard. After debugging, we changed the circuits with new capacitors. When testing the 3.3 Voltage Regulator, we measured the voltage using a multimeter, and saw the voltage to range around 3.27 to 3.30 volts, which provides us a percent error ranging from -0.9% to -0 when using the formula $Percent\ Error = (V_{observed} - V_{true})/V_{true}$. We did the same thing for the 5.0 Voltage Regulator, and saw the voltage range to be around 5.05 Volts, which provides

us a percent error of 1% when using the formula

$Percent\ Error = (V_{observed} - V_{true})/V_{true}$. We also measured the voltage from the battery, to make sure that it was outputting around 12 Volts, and we saw it was outputting around 11.08-11.11 Volts, when we had frequently used the battery and hadn't recharged it yet.

For the control subsystem, the requirements were to send a 3.3V signal to turn on and 5V signal to turn off the relays for each subsystem. To achieve this, we needed to be able to program the microcontroller. According to the datasheet, the microcontroller needs to be in bootloader mode to be programmed. The switches or capacitors need to be connected to the enable and IO0. This circuit is shown on Figure 3. On the first order of the PCB, the footprint for the switches needed to be updated. Once we were able to test the microcontroller with the new PCB, we needed python to program our original microcontroller. We switched to the ESP32-WROOM-32E, so we can use the Arduino IDE.

After we were able to program the microcontroller, we tested the verifications for the microcontroller. We created code for the relays and we were able to turn on and off the relays with the microcontroller.

The power and control subsystem is able to power and control the subsystems. We were not able to get the power subsystem on the PCB, but the subsystem did work on a breadboard. We made changes to the power subsystem in the circuit design for a revised PCB as shown in Figure 7.

2.4.2 Sensor Subsystem

In order to verify that our sensors worked properly, we first tested them separately to see what their expected behavior was alone, and then we tested them together. We used an ESP32D Development board to verify our sensor activity, and then checked the activity on our PCB.

The first sensor that we verified was the PIR Sensor. The PIR Sensor that we purchased has 3 connections, a power connection, a ground connection, and a data connection. The circuit was fairly simple; connect the ground connection to the ground of the development board, the power connection to the power connection of the development board, and the data connection to any GPIO pin on the development board. We also used an LED, so that when motion was detected from the PIR Sensor, the LED would turn on. We connected the LED using a wire to a GPIO pin on the development, and then using arduino code, we were able to code if the PIR motion sensor were to detect motion, turn on the LED and vice versa [14]. Once we were able to get the PIR Sensor working alone, we added the relay to the circuit.

The solid state relay we were using has four connections, one connection being a microcontroller input, one connection connecting to the circuit of PIR Sensor, one being a no connect, and one connecting to a 430 ohm resistor that connects to ground. We connected the microcontroller input to a GPIO pin on the microcontroller, the circuit connection to the ground connection of the PIR Sensor, and the other connections respectively. In our arduino code, to see if the relay was working, we would set the relay to HIGH. When the relay is set to HIGH, that means the PIR Sensor should be able to detect motion. When the relay is set to LOW, that means the PIR Sensor shouldn't be able to detect motion.

When the relay was set to HIGH, we used a multimeter to see the voltage flowing through the relay to the PIR Sensor, and we saw the voltage to be to range between 4.8 to 4.9

volts, which provides us a percent error ranging from -2% to -5% when using the formula $Percent\ Error = (V_{observed} - V_{true})/V_{true}$. Using the relay and re-setting the microcontroller, we also wanted to see for every instance how long it would take the PIR Sensor to first get triggered, and then after it stopped sensing motion, how long would the second trigger take. Below are our results for our first ten tests.

Table 1 : Results from Testing the Triggering Time for the PIR Sensor

Test	First PIR Trigger	Second PIR Trigger
Test 1	2 seconds	4 seconds
Test 2	5 seconds	5 seconds
Test 3	7 seconds	2 seconds
Test 4	2 seconds	5 seconds
Test 5	6 seconds	6 seconds
Test 6	17 seconds	5 seconds
Test 7	9 seconds	5 seconds
Test 8	3 seconds	4 seconds
Test 9	5 seconds	5 seconds
Test 10	11 seconds	5 seconds

There was one instance where the PIR sensor initially could not detect motion within our five to fifteen second threshold, so taking that into account and calculating the success rate for our first ten tests, we had a success rate of almost 90%.

When testing out the accelerometer, the connections we had were the ground connection, the voltage connection, the SCL connection, and the SDA connection. The accelerometer requires the I²C protocol, hence why we have the SCL and SDA connections [15]. The relay connections are exactly the same, except now we will be connecting to the accelerometer circuit. The AXDL 343 has a built-in arduino library that shows the x,y,z coordinates as you tilt the sensor. Using that library, we were able to get the initial coordinates of the accelerometer, as if there was a +/- difference of 0.5 between the x,y,z coordinates, we were alerted that the accelerometer detected motion. We used the same relay logic in the sense that if the relay was HIGH, the accelerometer would detect motion, otherwise it wouldn't. We measured the voltage going through the accelerometer when the relay was high to be 3.23-3.33 Volts, which provides a percent error range of -2.12% to 0.9% when using the formula $Percent\ Error = (V_{observed} - V_{true})/V_{true}$. The time it took the accelerometer to detect vibration and tilting was immediate, as we got the initial position of the accelerometer through our arduino code, and whenever the initial position was changed, the accelerometer would immediately display that it had detected motion.

When combining the PIR Sensor with the accelerometer, the sensor's were performing the same, and we got a similar sensor result when integrating it on our PCB.

2.4.3 Alarm Subsystem

In order to test the Alarm Subsystem, we first tested the alarm individually by procuring the correct TRS audio jack connector and connecting it to 12V. We then modified the TRS connections to reflect our circuit design with solid state relays. Before testing the entire circuit, we first tested the solid state relay. We connected the solid state relay to the microcontroller and a pull down resistor, leaving the other pins disconnected. Then, using a multimeter across the disconnected pins, we verified that the relay was actuated when microcontroller input was HIGH (5V) and was not actuated when microcontroller input was LOW (3.3V). After testing the alarm and relay individually, we combined both components, verified our circuit design and tested the Alarm Subsystem in its entirety. When microcontroller input to the relay was HIGH, the relay became actuated and the alarm would turn on and when microcontroller input to the relay was LOW, the alarm circuit would be disconnected, causing the alarm to turn off.

Our next requirement for the Alarm Subsystem was to ensure that the alarm rang for a predefined amount of time when suspicious activity was detected. While the ideal amount of time is 3-5 minutes, for the purpose of testing our logic, we set our time to 5 seconds. We ran our controller code and verified this by starting a stopwatch the moment the alarm was triggered. After five trials, we were able to verify that the alarm did ring for 5 seconds before it turned off. Our stopwatch measurements varied by a few milliseconds in our trials, but that is attributed more towards human error (starting and stopping the time a few milliseconds early or late).

Additionally, we also wanted to ensure that the user would be able to override the timer logic for the alarm via the web application. In order to verify this, we repeated the same setup as the previous verification, but turned the alarm off via the app around 2 seconds after the alarm started to ring. We noticed that the alarm almost immediately stopped ringing after the button was pressed on the application, and had a delay of at most 500 milliseconds. Similarly, there was a minimal delay when attempting to turn the alarm on via the application. This delay was verified by once again using a stopwatch and conducting multiple trials to assess the average response time of the subsystem after a button on the application was pressed.

2.4.4 Camera Subsystem

The camera subsystem had four requirements. We originally used solid-state relays to control when the camera turns on and off. This means the camera should turn on and off based on the signal from the microcontroller. The camera should also upload footage with a delay of at most 30 seconds. We needed this requirement to ensure if a theft occurred, the car owner would have evidence before the thief tried to temper with the device. The final requirement was to provide 3.3V to the camera when turned on by the microcontroller.

To test the camera, we used a development board. We researched demos using the ESP32 and OV7670. We used the example from Mudassar Tamboli due to the lack of datasheets for the OV7670 camera [16]. We used an ESP32-WROOM-32D Development Board and used the pins assigned in the demo. We were able to get the camera uploading live footage on a mobile interface.

We verified that the camera can be turned on and off from the microcontroller. According to another TA, we were advised to use the Power Down Pin on the camera to turn on and off the camera instead of the solid-state relay. As a result, we needed to change our verification for the camera. The camera could still be turned on and off by a signal from the microcontroller. This change is also reflected in our revised PCB on Figure 7.

The camera also uploaded footage faster than our initial prediction. The camera started recording within 5 seconds of sensor detection and the footage had less than a second delay. The camera was also powered with 3.3V when turned on.

Due to issues with the microcontroller and pin assignments, we were not able to get the camera working on our PCB. However, the camera subsystem was fully integrated with all of our subsystems on the ESP32-WROOM-32D Development Board. We edited the pin assignments on our revised PCB and, with a new PCB order, we would be able to integrate the camera with our subsystems.

3. Cost and Schedule

3.1 Cost

Cost of Labor

Salary of Labor : \$40/hour

Average hours worked per week : 15 hours

Amount of time in Spring Semester : 16 weeks

Total Cost of Labor : $\$40/\text{hour} * 2.5 * 15(\text{hour}) * 16 = \$24,000 \times 3 = \mathbf{\$72,000}$

Component	Price	Qty	Total
ESP32-WROOM-32D	\$18.99	1	\$18.99
PIR IR motion sensor	\$9.95	1	\$9.95
120dB Electronic Alarm	\$9.99	1	\$9.99
USB to TTL Module Serial Converter Adapter	\$17.98	1	\$17.98
Solid-state relay	\$3.49	3	\$10.47
Push Buttons	\$0.17	2	\$0.34
Audio Jack	\$1.00	1	\$1.00
IR Emitter	\$0.72	1	\$0.72
Voltage Regulator (5V)	\$0.63	1	\$0.63
Voltage Regulator (3.3V)	\$0.77	1	\$0.77
Accelerometer	\$5.95	1	\$5.95

Wire for Accelerometer	\$0.95	2	\$3.10
Camera Connector (2x09)	\$3.12	1	\$3.12
Capacitors	\$0.30	4	\$1.20
Resistors	\$0.10	3	\$0.30
IR Emitter Connector (1x02)	\$0.29	1	\$0.29
Battery Connector	\$0.29	1	\$0.29
PIR Sensor Connector	\$0.28	1	\$0.28
Adapter Connectors (1x04)	\$0.53	1	\$0.53
IO Connectors (1x05)	\$0.78	1	\$0.78
IO Connectors (1x08)	\$1.25	1	\$1.25
Accelerometer Connector	\$0.55	1	\$0.55
Male-to-Male Wires	\$5.30	3	\$15.90
Female-to-Female Wires	\$2.50	1	\$2.50

3.2 Schedule

Week	Aditi Tyagi	Anushka Parikh	Shruthii Sathyanarayanan
2/20	<ul style="list-style-type: none"> Research parts for sensor subsystem Research parts for battery and power subsystem Make a draft of circuit schematic Get feedback and prepare for design review 	<ul style="list-style-type: none"> Research parts for camera subsystem Research parts for microcontroller Make a draft of circuit schematic Get feedback and prepare for design review 	<ul style="list-style-type: none"> Research parts for alarm and notification subsystem Make a draft of circuit schematic Get feedback and prepare for design review
2/27	<ul style="list-style-type: none"> Finalize circuit schematic Create PCB design Order parts for sensor subsystem Order parts for power subsystem 	<ul style="list-style-type: none"> Finalize circuit schematic Create PCB design Order parts for camera subsystem Order microcontroller 	<ul style="list-style-type: none"> Finalize circuit schematic Create PCB design Order parts for alarm and notifications subsystem
3/6	<ul style="list-style-type: none"> Research project box for machine shop 	<ul style="list-style-type: none"> Order PCB 	<ul style="list-style-type: none"> Make design for machine shop

3/13 (Spring Break)			
3/20	<ul style="list-style-type: none"> Code PCB 	<ul style="list-style-type: none"> Solder Parts on PCB Code Front-End of Android App 	<ul style="list-style-type: none"> Code Back-End of Android App
3/27	<ul style="list-style-type: none"> Test PCB 	<ul style="list-style-type: none"> Submit Parts to Machine Shop Submit Design to Machine Shop 	<ul style="list-style-type: none"> Test Android App
4/3	<ul style="list-style-type: none"> Create tests for sensor subsystem 	<ul style="list-style-type: none"> Create tests for camera subsystem 	<ul style="list-style-type: none"> Create tests for alarm and notifications subsystem
4/10	<ul style="list-style-type: none"> Test final prototype 	<ul style="list-style-type: none"> Test final prototype 	
4/17	<ul style="list-style-type: none"> Report test results Prepare for demonstrations 	<ul style="list-style-type: none"> Report test results Prepare for demonstrations 	<ul style="list-style-type: none"> Test final prototype
4/24	<ul style="list-style-type: none"> Do Demonstration Prepare for Presentation 	<ul style="list-style-type: none"> Do Demonstration Prepare for Presentation 	<ul style="list-style-type: none"> Do Demonstration Prepare for Presentation
5/1	<ul style="list-style-type: none"> Do Presentation 	<ul style="list-style-type: none"> Do Presentation 	<ul style="list-style-type: none"> Do Presentation

4. Conclusion

4.1 Accomplishments

For our final demo, we were able to show all of our subsystems integrated on a development board. Our sensors were able to detect motion and send a signal to the microcontroller. The microcontroller was able to send signals to the alarm subsystem and camera subsystem. We also developed a mobile interface to display the camera footage and turn on and off the sensors and alarm. The power subsystem was also able to power the subsystem on the breadboard.

The sensor subsystem and mobile interface were also working on our PCB. Both the PIR sensor and accelerometer were able to send signals to the microcontroller and sensor detection could be remotely enabled or disabled through the mobile interface.

We were also able to create a prototype within the size constraints from our high-level requirements. We placed the device under a SUV to demonstrate a real-life application for our device. The device was able to be placed facing the catalytic converter, so the PIR sensor could detect motion surrounding the area and the camera could capture footage.

4.2 Uncertainties

Our greatest challenge was the microcontroller for our PCB. We chose the ESP32-WROOM-32E. On our PCB, the ESP32-WROOM-32 footprint was very different from

our microcontroller. Some of the pins we needed were “no connects” on the ESP32-WROOM-32E. Due to the discrepancy, we decided to order a different version of the ESP32, ESP32-WROOM-32D. This microcontroller was similar to the original footprint and had much fewer no connect pins. We were able to reassign some of our pins and successfully implement the sensor subsystem logic on the PCB. However, the layout of certain pins like the clock were different on the footprint and the ESP32-WROOM-32D and we would have had to cut our PCB and manually reconfigure our connections to integrate the remaining subsystems with the 32D microcontroller.

We were able to fully integrate the subsystems on a breadboard. Based on the full integration with the development board, we redesigned the circuit design on Figure 7. Given the chance for a new PCB order, we would have been able to fully integrate our subsystems on the PCB.

Due to our challenges with the PCB, we were unable to fully test the device in a real world situation. There are uncertainties on the area the camera will capture. We also would like to test our device in different weather conditions and environments.

We also need to test the false positive rate of our device.

4.3 Ethics and Safety

There are several safety considerations to take into account when designing a surveillance device for catalytic converters. Catalytic converters can reach extremely high temperatures of up to 500 degrees Fahrenheit. This can potentially be a hazard depending on the placement of the device, wherein the high temperature can impact our device or even injure the car owner when they attempt to attach/detach the device when the catalytic converter is still hot. In order to ensure the safety of the user, we would determine the placement of the device such that it provides a clear view of the area from which the catalytic converter can be accessed and be placed far enough from the catalytic converter that the user will not be harmed by the high heat. In order to prevent the device from being damaged by the heat, we would consider selecting a temperature-resistant box in which the device may be placed.

A caveat with creating a detachable device is risking the chance of the device being tampered with. If a thief can easily remove the device, it defeats the purpose we aim to solve. In order to prevent tampering, we will use sensors that detect vibrations and movement, which allows us to monitor tampering of the catalytic converter as well as tampering of the device. By sealing the circuitry of the device in a sturdy box, we would be able to notify the car owner before a thief can destroy the box and the device within.

Since our device will be placed on the undercarriage of a car, it will be exposed to external environmental factors that could impact the performance of the device such as extreme temperatures, water from rain, winds, etc. To ensure our device performs well regardless of these environmental factors, we selected as many weather and temperature resistant components as possible and plan to place the device inside a weather, water, and temperature resistant casing.

Our device doesn't discriminate against who can use it. The device can be used by any car owner. The device is open to the public, and thereby complies with section 7.8 II-7 of the IEEE Code of Ethics. Our device also abides by section 7.8 I-1 , as the device is promoting

safety by trying to reduce the number of stolen catalytic devices, which increases the safety of the car itself, and the person driving the car [17].

4.4 Future Work

First, we would like to fully integrate our subsystems on the PCB. We can order a new PCB from the revised PCB circuit as shown in Figure 7.

We would also like to make our project more effective for car owners. We can change the size of the device to make it more discrete by decreasing the length and width. The height should stay the same to ensure the camera can capture evidence of the catalytic converter and its surroundings area. It would also be ideal to power the device with the car battery. Currently, our device is powered by a rechargeable battery, so the car owner must recharge it before use. The car battery can keep the device charged at all times, but we will need to consider safety and power usage.

References

1. Way, "Catalytic converter thefts: Most targeted cars listed." Way.com. [Online]. <https://www.way.com/blog/catalytic-converter-thefts-most-targetted-cars-listed/> [Accessed: 4-May-2022].
2. American Family Insurance, "How to Prevent Catalytic Converter Theft." amfam.com. [Online]. <https://www.amfam.com/resources/articles/on-the-road/prevent-catalytic-converter-theft> [Accessed: 4-May-2022].
3. CarParts, "Can You Drive Without a Catalytic Converter." carparts.com. [Online]. <https://www.carparts.com/blog/can-you-drive-without-a-catalytic-converter/#:~:text=Vehicles%20can%20be%20driven%20without,done%27t%20enforce%20emissions%20standards> [Accessed: 4-May-2022].
4. The News-Gazette, "Catalytic-converter thefts have victims' vehicles, wallets feeling the pain." news-gazette.com. [Online]. https://www.news-gazette.com/news/local/courts-police-fire/catalytic-converter-thefts-have-victims-vehicles-wallets-feeling-the-pain/article_cc7a5545-c3a5-54f3-bbda-3ae1d1c94c5c.html [Accessed: 4-May-2022].
5. ESP32S2WROOM, "ESP32S2WROOM ESP32S2WROOMI Datasheet." espressif.com. [Online]. https://www.espressif.com/sites/default/files/documentation/esp32-s2-wroom_esp32-s2-wroom-i_datasheet_en.pdf. [Accessed: 4-May-2022].
6. ESP-IDF Programming Guide, "Wifi." docs.espressif.com. [Online]. https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_wifi.html. [Accessed: 4-May-2022].
7. "Solid state relay or solid state switch," Basic Electronics Tutorials, 24-Aug-2019. [Online]. Available: <https://www.electronics-tutorials.ws/power/solid-state-relay.html>. [Accessed: 04-May-2022].
8. Arrow, "How Do Motion Sensors Work? Types & Applications." arrow.com. [Online]. <https://www.arrow.com/en/research-and-events/articles/how-motion-sensors-work#:~:text=Motion%20Sensors%20Explained&text=There%20are%20three%20types%20of.%2C%20and%20Dual%20Tech%2FHybrid> [Accessed: 4-May-2022].
9. Live Science, "Accelerometers: What They Are & How They Work." livescience.com [Online]. <https://www.livescience.com/40102-accelerometers.html> [Accessed: 4-May-2022].
10. Adafruit, "Analog Devices ADXL343 Breakout Learning Guide" learn.adafruit.com [Online]. <https://learn.adafruit.com/adxl343-breakout-learning-guide> [Accessed: 4-May-2022].
11. "Electronic Buzzer Alarm," Amazon. [Online]. Available: <https://www.amazon.com/Electronic-Buzzer-Security-Speaker-Burglar/dp/B0875>

- WCSB5/ref=sr_1_13?keywords=12+volt+alarm+siren&qid=1644354141&sr=8-13 . [Accessed: 4-May-2022].
12. Embedded Programmer, "Hacking the OV7670 camera module (SCCB cheat sheet inside)."
[http://embeddedprogrammer.blogspot.com/\[Online\].http://embeddedprogrammer.blogspot.com/2012/07/hacking-ov7670-camera-module-sccb-cheat.html](http://embeddedprogrammer.blogspot.com/[Online].http://embeddedprogrammer.blogspot.com/2012/07/hacking-ov7670-camera-module-sccb-cheat.html).
 [Accessed: 4-May-2022].
 13. Espressif, "News ESP32's Wi-Fi Range Extended to 10 km with a Directional Antenna." www.espressif.com.[Online].
<https://www.espressif.com/en/news/esp32%E2%80%99s-wi-fi-range-extended-10-km-directional-antenna> [Accessed: 4-May-2022].
 14. Random Tutorial Nerds, "ESP32 with PIR Motion using Interrupts and Timers."
<https://randomnerdtutorials.com>. [Online].
<https://randomnerdtutorials.com/esp32-pir-motion-sensor-interrupts-timers/>
 [Accessed 4-May-2022].
 15. L. Clark, K. Townsend, and I. Wellish, "Analog Devices ADXL343 Breakout Learning Guide," Adafruit Learning System, 24-Feb-2019. [Online]. Available:
<https://learn.adafruit.com/adxl343-breakout-learning-guide/arduino?fbclid=IwAR2BYtjPrmzdajlvvw3x54mIK7Am1gfUpMPsym8mPb-3TEIMcmBGR89iESA>.
 [Accessed: 04-May-2022].
 16. M. Tamboli, Medium, "ESP32+OV7670 — WebSocket Video Camera."
[medium.com.\[Online\].https://medium.com/@mudassar.tamboli/esp32-ov7670-websocket-video-camera-26c35aedcc64](https://medium.com/@mudassar.tamboli/esp32-ov7670-websocket-video-camera-26c35aedcc64) [Accessed: 4-May-2022]
 17. IEEE Board of Directors. "IEEE Code of Ethics." IEEE Code of Policies, Section 7 - Professional Activities (Part A - IEEE Policies).
<https://www.ieee.org/about/corporate/governance/p7-8.html>. [Accessed: 4-May-2022].

Appendix

Appendix A: Requirements and Verifications

A.1 Power/Control Subsystem

Table 2: Requirements and Verifications for Power and Control Subsystems

Requirements	Verifications	Verified
1. The power supply provides 12V (+- 5%).	1. Measure the output voltage using an oscilloscope, ensuring that the output voltage stays within 5% of 12V.	1.Yes
2. Voltage Regulator 1 provides fixed 5.0 V (+/-5%) from 12V source.	2. Measure the output voltage using an oscilloscope, ensuring that the output voltage stays within 5% of 6V.	2.Yes on the development board
3. Voltage Regulator 2 provides fixed 3.3 V (+/- 5%) from 12 V source.	3. Measure the output voltage using an oscilloscope, ensuring that the output voltage stays within 5% of 3.3V.	3.Yes on the development board
4. Both regulators maintain thermal stability below 125°C	4. During verification for Requirement 2 and 3, use an IR thermometer to ensure the IC stays below 125°C	4.Yes
5. The microcontroller sends a 3V to 3.3V signal to turn off the alarm and a 5V to 5.5V signal to turn on the alarm.	5. Connect sensors to microcontrollers. Measure voltage at the signal for the Alarm Subsystem using an oscilloscope. Then, activate sensors by moving the device. Measure voltage at the signal for the Alarm Subsystem using an oscilloscope.	5.Yes on development board
6. The microcontroller sends a 3V to 3.3V signal to turn off the camera and a 5V to 5.5V signal to turn on the camera.	6. Connect sensors to microcontrollers. Measure voltage at the signal for the Camera Subsystem using an oscilloscope. Then, activate sensors by moving the	6. Yes on development board

	device. Measure voltage at the signal for the Camera Subsystem using an oscilloscope.	
7. The microcontroller sends a 3V to 3.3V signal to turn off the sensors and a 5V to 5.5V signal to turn on the sensors.	7. Connect sensors to microcontrollers. Measure voltage at the signal for the Sensor Subsystem using an oscilloscope. Then, activate sensors by moving the device. Measure voltage at the signal for the Sensor Subsystem using an oscilloscope.	7. Yes on development board and PCB

A.2 Sensor Subsystem

Table 3: Requirements and Verifications for Sensor Subsystems

Requirements	Verifications	Verified
1. Once the motion sensors have been triggered, they must trigger the alarm and notification systems within 5 to 15 seconds	1. Connect circuit as shown in schematic. Record the time between the motion sensors being triggered and the alarm turning on.	1. Yes, on the development board
2. For the PIR sensor, provide 5V +/- 5% from a 5V-12V source	2. Measure the output voltage using an oscilloscope, ensuring that the output voltage stays within 5% of 5 V.	2. Yes
3. For the PIR sensor, maintain a thermal temperature that is below 70 C but is above -20 C.	3. Use an IR thermometer to ensure the IC stays below 70 C but is above -20 C	3. Yes
4. For the accelerometer, maintain a thermal temperature that is below 125 C but is above -55 C. This is for the vibration range	4. Use an IR thermometer to ensure the IC stays below 125 C but is above -55 C	4. Yes
5. The weight of the	5. Measure the weight of the	5. Yes

accelerometer should be no greater than 10% of the weight of the surveillance device.	accelerometer using a scale, and do the same for the surveillance device. Mathematically calculate the 10% difference, making sure it reaches that mark.	
6. For the accelerometer, provide a voltage of 3.3 +/- 5%.	6. Measure the output voltage using an oscilloscope, ensuring that the output voltage stays within 5% of 3.3 V.	6. Yes
7. For both accelerometer and sensor, have them be placed a foot away from the catalytic converter for optimal coverage of area and surroundings.	7. Measure the placement of the accelerometer and the sensor, making sure that it is at least 1 foot away from the catalytic converter.	7. Yes

A.3 Alarm Subsystem

Table 4: Requirements and Verifications for Alarm Subsystems

Requirements	Verifications	Verified
1. Within 5 to 15 seconds of the sensors being triggered, the alarm must start ringing and a notification should be sent to the user	1. Connect circuit as shown in schematic. Trigger sensors by tilting and moving the device. Record the time between triggering the sensors and the alarm turning on/notification being sent.	1. Yes, on the development board
2. The alarm should be turned off when the input from the microcontroller is 3V to 3.3V.	2. Connect circuit as shown in the circuit schematic. Send an input of 3V to 3.3V to the alarm subsystem. If the alarm begins to ring, increase the resistance of the resistor in the alarm circuit or decrease the input voltage.	2. Yes, on the development board
3. The alarm should be turned on when the input from the microcontroller is 9V	3. Connect circuit as shown in the circuit schematic. Send an input of 9V to 12V to the alarm subsystem. If the alarm	3. Yes, on the development board

to 12V.	does not ring, decrease the resistance of the relay coil or increase the input voltage. Input voltage should not exceed 12V.	
4. The alarm should ring for 3 to 5 minutes or until the user manually turns it off through the mobile app.	4A. Connect circuit as shown in circuit schematic. Record the duration of the alarm once it has been triggered. If triggered longer or shorter than desired time, modify components between sensors and alarm and edit code to decrease/increase time accordingly. 4B. Connect circuit as shown in circuit schematic. Once the alarm starts ringing, attempt to turn it off via the app. Record the time taken from turning off the alarm on the app to when the alarm actually turns off. If the alarm cannot be controlled from the app, verify network connections and/or reestablish connection between the MCU and phone. If the alarm takes too long to turn off, modify components between MCU and alarm and edit code to decrease time accordingly.	4. Yes, on the development board

A.4 Camera Subsystem

Table 5: Requirements and Verifications for Camera Subsystems

Requirements	Verifications	Verified
1. The camera should be turned off when the input from the microcontroller is 3V to 3.3V.	1. Connect circuit as shown in circuit schematic. Send a 3.3V signal at the input from the microcontroller. Check live footage to see if the camera is turned off. If the camera is on, add more resistance to the circuit and	1. Yes on the development board

	check the relay.	
2. The camera should be turned on when the input from the microcontroller is 5V to 5.5V.	2. Connect circuit as shown in circuit schematic. Send a 5V signal at the input from the microcontroller. Check live footage to see if the camera is turned on. If the camera is off, add less resistance to the circuit and check the relay.	2. Yes, on the development board
3. The camera will upload footage on the mobile app with at most a 30 second delay.	3. Connect circuit as shown in circuit schematic. Turn on the camera. Wave a hand in front of the camera. Record the time between waving the hand in front of the camera to the hand appearing on the mobile app. If the camera does not upload the footage within the desired time, edit the code to upload footage faster.	3. Yes, on the development board
4. When the relay is turned on, the voltage across the camera should be 3.3V.	4. Connect the circuit as shown in the circuit schematic. Send a 5V signal to the relay to turn it on. Measure (using a tool like the oscilloscope) the voltage across the camera.	4. Yes, on the development board

Appendix B: First Order PCB Design

B.1 Circuit Design

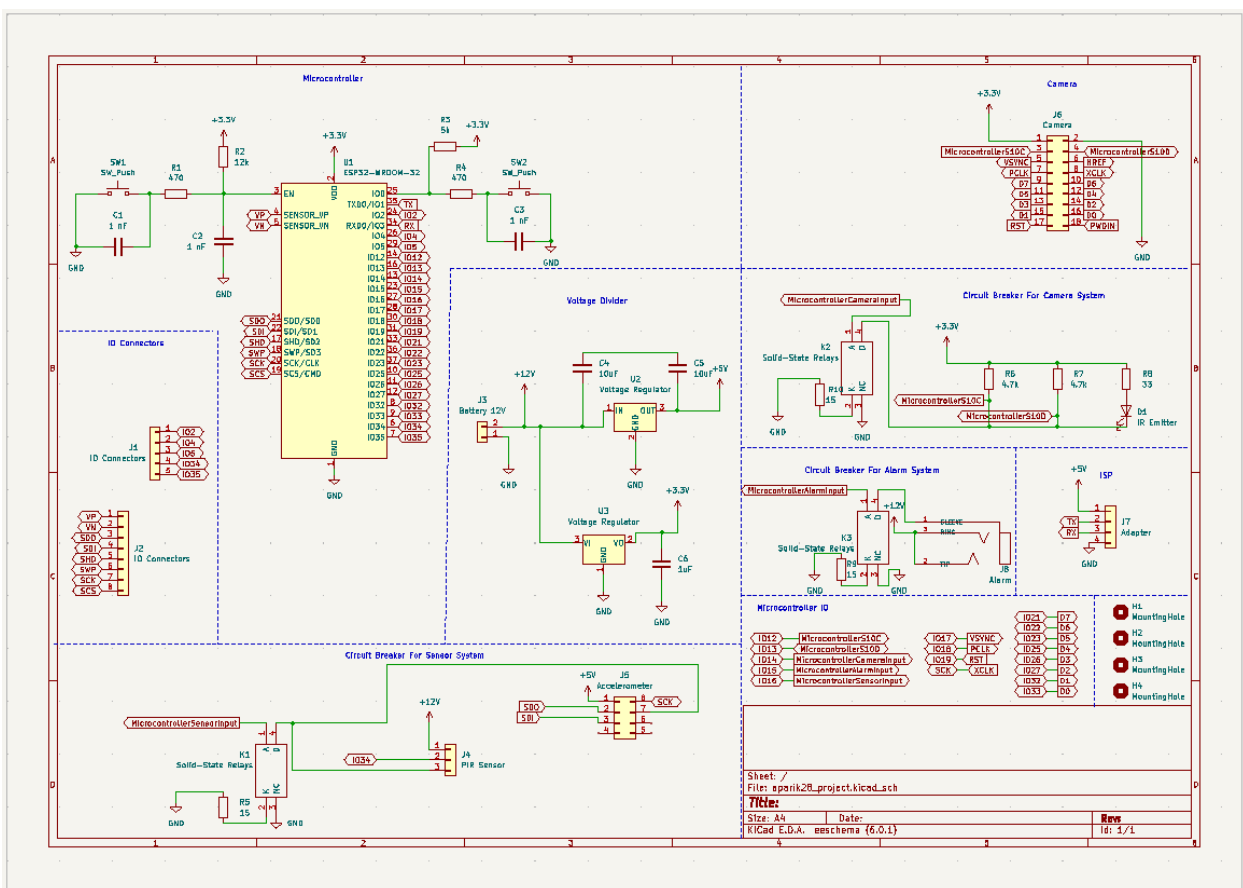


Figure 3: Circuit Design for PCB Order 1

B.2 PCB Design

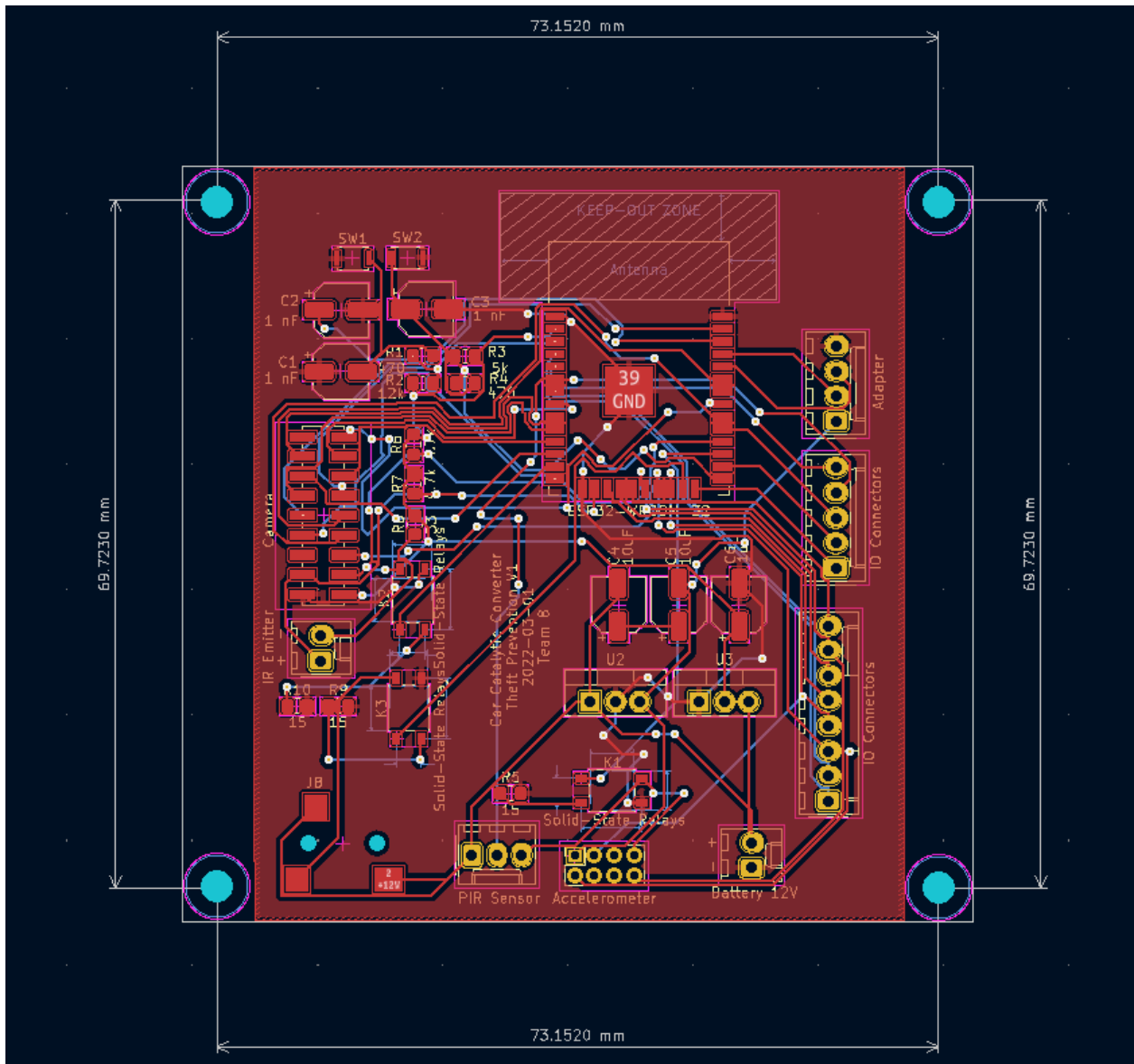


Figure 4: PCB Design for PCB Order 1

Appendix C: Second Order PCB Design

C.1 Circuit Design

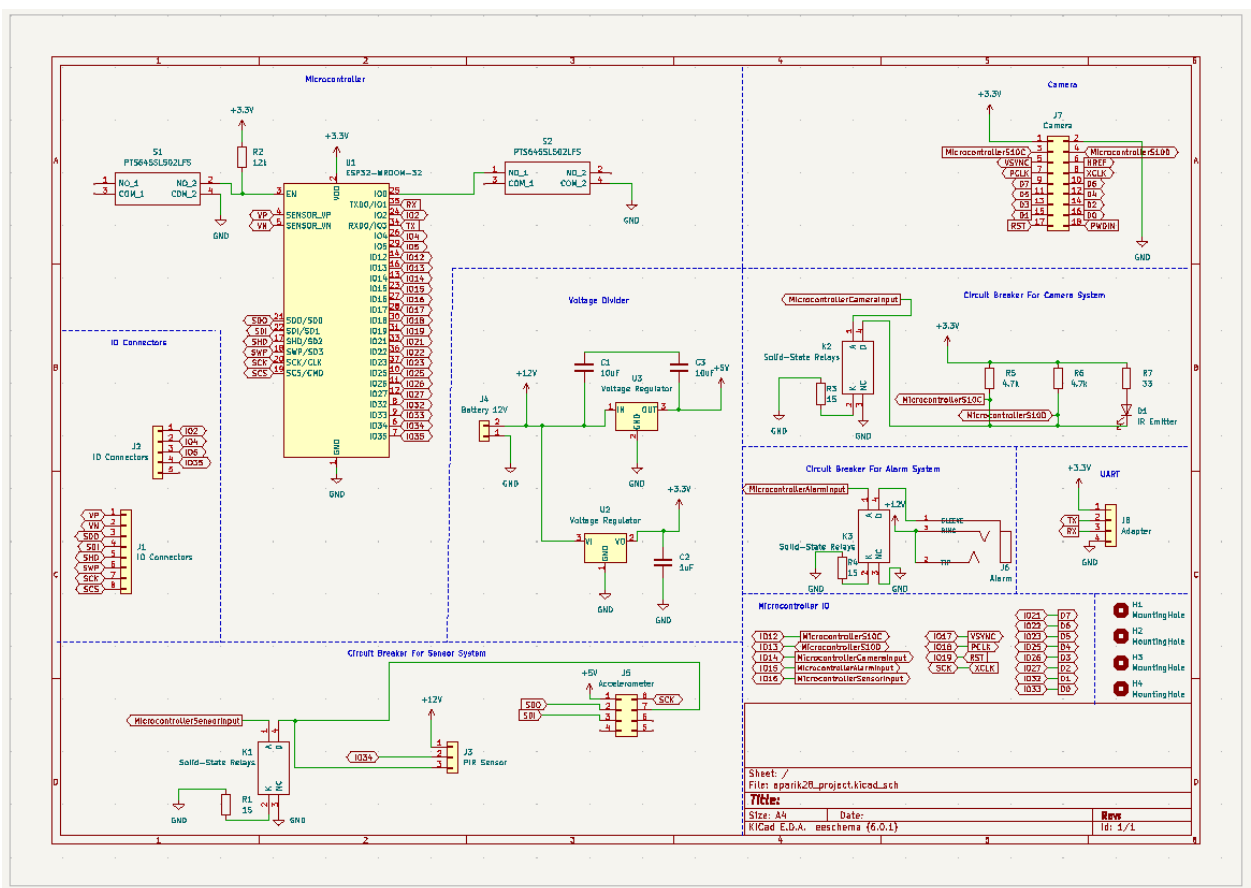


Figure 5: Circuit Design for PCB Order 2

C.2 PCB Design

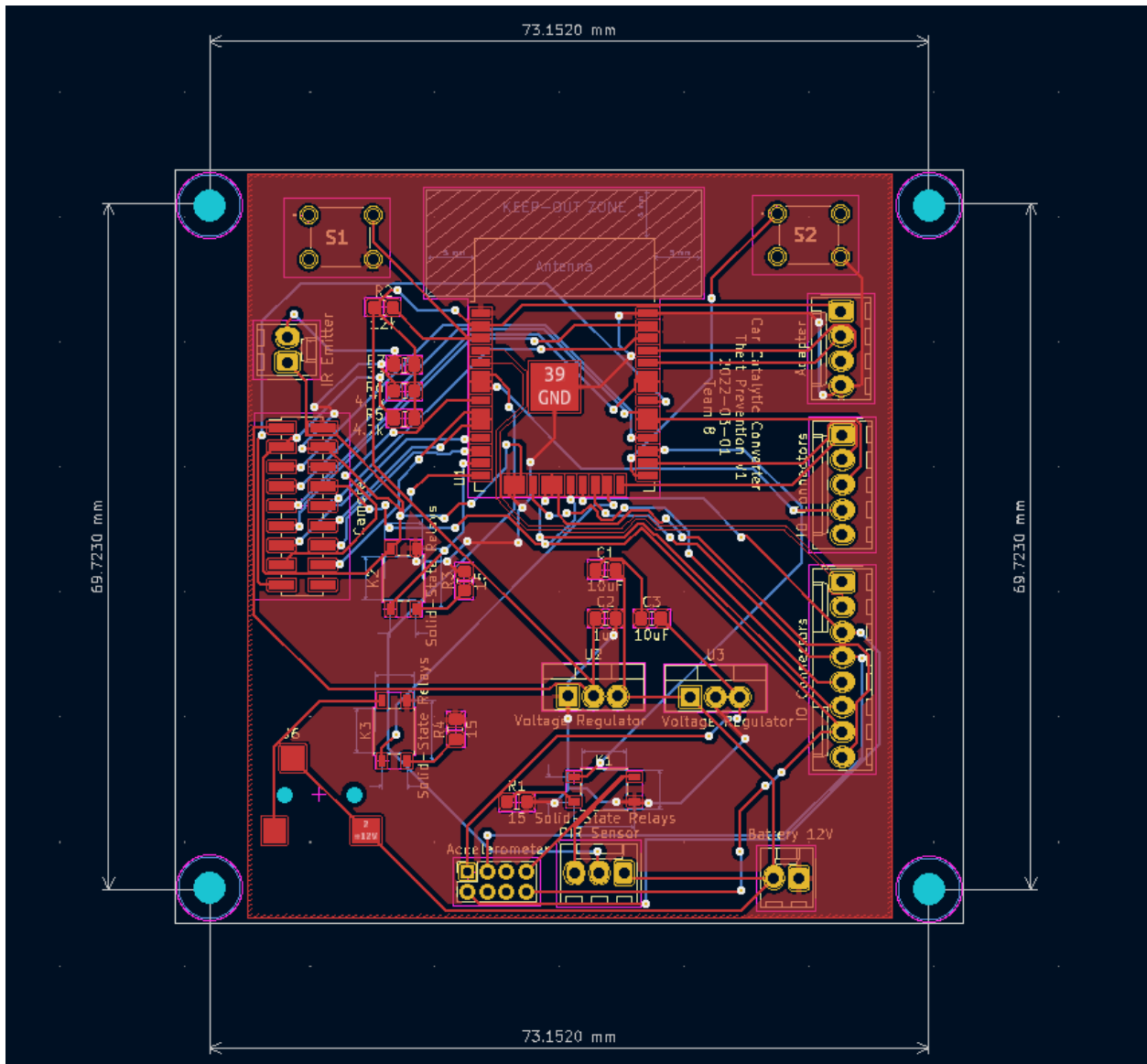


Figure 6: PCB Design for PCB Order 2

Appendix D: Revised PCB Design

D.1 Circuit Design

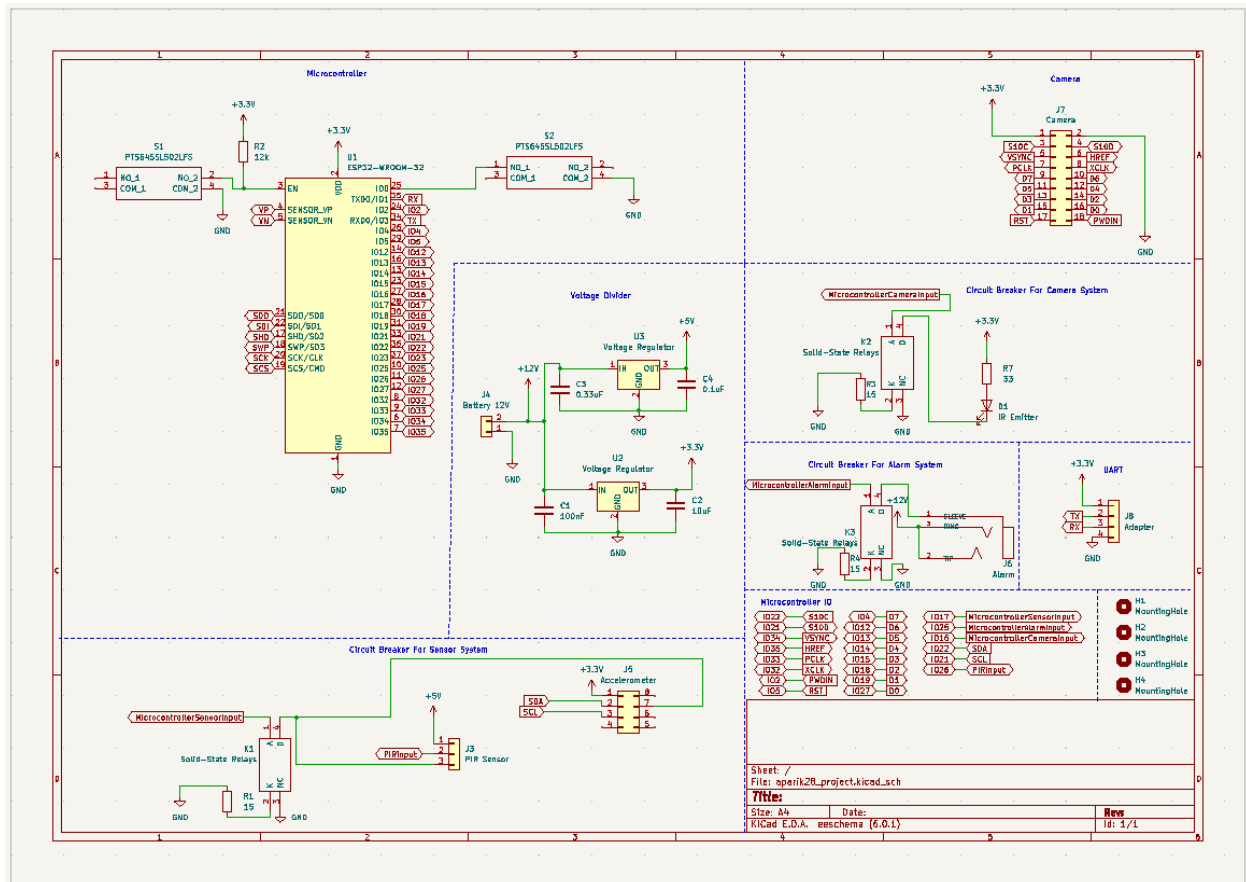


Figure 7: Circuit Design for the Revised PCB Design