CPE 4850

Final Project Report

DECEPTICARS

a better smart parking solution

Monae Edmead, Ben Elle, Johnny Lozano and Michael Lozen

Professor Scott Tippens

May 2, 2023

Table of Contents

Abstract

The Decepticars smart parking solution offers a transformative approach to traditional parking structures, providing a comprehensive data hub that offers increased user experience, efficient and effective parking enforcement, and key user data to improve the structure further. The system utilizes affordable sensor nodes that are placed at every entry and exit point, providing vital data to everyday drivers such as current and projected capacity.

In addition to user data, critical information is provided to the parking enforcement team, including images and identifying information of every vehicle that enters or exits, as well as whether or not a vehicle is authorized to park in the structure. All of this information is delivered through a robust yet intuitive user interface that creates a connected and informed environment.

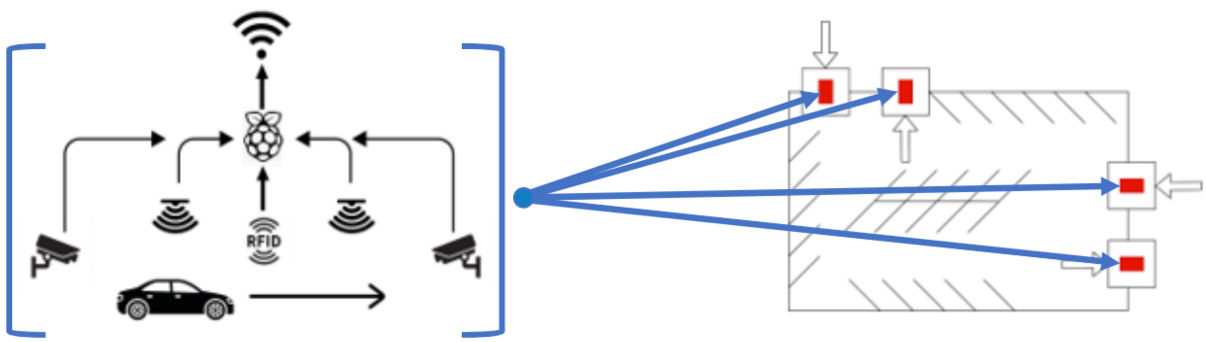


Figure 1. System Description

One of the key benefits of the Decepticars system is its scalability, meaning it can be customized to fit the specific needs of a parking structure regardless of size. Its modular design, both inside and out, allows for simple and affordable maintenance, with individual sensor nodes being easily added or removed without affecting the functionality of others.

The system also employs a predictive occupancy model that provides incremental value to the system over time. This model is adaptive, meaning that the longer the system collects data, the more accurate it becomes.

This report documents the development of the Decepticars smart parking solution from the ground up, detailing the design, implementation, and testing of the system to provide a complete picture of its capabilities and potential impact on the smart parking market.

Introduction

Parking garages are a necessary component on college campuses. Over the years due to increasing enrollment reported by colleges, there have been noticeable parking issues. Students all over the country have made their frustration known about not being able to use the parking garages that they paid to access. One example can be seen by students who attend Truman State University with more than 80% of participants in a survey[1] claiming that there is a serious parking issue that needs to be addressed and more than 60% have trouble finding spots on campus. Similar concerns have been expressed by Kennesaw State University students[2]. They have shown concern due to the campuses not having enough parking spaces to accommodate everyone.

There are currently other smart parking solutions on the market, however they have their flaws. For starters, they require large financial and time investments. According to the US Department of Transportation[3], the cost of implementing a smart parking system into a parking garage starts at $50,000 and can skyrocket depending on the intricacies of the garage. Existing systems typically have to be designed specifically for the garage, decommissioning it for some time. Systems that don't use gates and kiosks, which would significantly slow down the entryways and exits, causing bottlenecking.

This project intends to deliver an improved smart parking experience. The system should cut down on unauthorized traffic, be informative to all users, and provide value to the parking structure as a whole. To beat the current market, this product will be delivered at a low cost, in a modular, easy to install and maintain system that can conform to any parking structure.

[1] Source: https://tmn.truman.edu/blog/news/campus/campus-parking-causes-student-frustration/
[2] Source: https://tinyurl.com/2m3bksnc
[3] Source: https://www.itskrs.its.dot.gov/its/benecost.nsf/ID/e4717c6f075baaa38525789b00610ecc

Design Requirements

Prior to the design process, a list of requirements must be established to ensure the versatility and comprehensiveness of the final design. Factors such as operational, environmental, user, shipping, and lifecycle requirements should be taken into consideration in order to create a durable and well-rounded product.

Operational Requirements

The system can be broken down into four critical processes:

- The system shall identify all vehicles that are entering or exiting the parking garage.
- The system shall count the amount of vehicles currently inside of the parking garage.
- The amount of spots available within the parking garage shall be displayed to the user, and the display shall be at the parking garage entrance and a web application accessible to users.
- A predicted number of spots available shall be made available to all users via a web application.
- Time of identification, pass validity, license plates, car make, model, color, and images of the front and back of the vehicle upon entry and exit shall be given to and accessible by parking garage owners. The use of this information to enforce parking fees and legal action is of the owner's discretion.

Environmental Requirements

As most parking garages are outdoors,the system will have to endure more extreme environmental conditions. Because the system will be covered by the parking garage, precipitation will not be a major factor of concern directly. However, temperature, humidity, lightning, and vibrations are all factors worth considering in design.

National temperatures range from 38°F to 87°F[4], and can be as extreme as -10°F to 115°F. Humidity can range from 25% to 91%. Lightning strikes[5] could knock out power temporarily or create power surges. Finally, vehicle traffic can cause 10 Hz vibrations[6] from within the parking garage.

[4] Source: https://www.currentresults.com/Weather/US/average-annual-temperatures-large-cities.php
[5] Source: https://tinyurl.com/vfxpbb4x
[6] Source: https://tinyurl.com/2p83dkn5

User Requirements

Since drivers will make up a large majority of the users, the transition process should be made the most smooth for them. The system will be designed so that they need not change any driving habits. Instead, all the drivers will need to do beforehand is place the RFID tag in the corner of their windshield that they receive from registering their vehicle. From then users will be able to read the current or predicted availability of their preferred parking decks from either the web application or the physical display.

The parking garage owners are already responsible for so much when it comes to budgeting, staffing, and other duties, so their transition should also be as seamless as possible. However this process requires some amount of involvement to set up. The parking garage owner will need to be responsible for the installation and initialization of the system which will be as easy as mounting, powering, connecting a display (optional), and connecting to the internet either wirelessly or via ethernet. Similarly, if there is a need for maintenance, they will have to contact the company to order a replacement part or unit. From then, the parking garage owner will have access to crucial data in making future decisions on the parking garage as well as have a satisfied clientele.

Parking enforcement is perhaps the most vital aspect in making sure this system is effective. Because this design will not use a gate in an effort to keep traffic moving, they are responsible for enforcing when someone is parking in the garage without a valid pass. Parking enforcement should be able to read a special designed admin dashboard that will highlight when someone has entered the garage without a pass. From there they should be able to read the provided car information that includes time of entry, images of the front and back of the vehicle, plate number, vehicle make, model, year, and color to quickly and efficiently enforce any parking violations. In the event that law enforcement becomes involved due to a nearby crime, they should also be able to provide any necessary data from the system to quickly identify any potential suspects.

Shipping and Installation Requirements

The system shall be designed to work with existing parking garage infrastructure. The system shall be preassembled and stored inside of a heavy duty shipping box. Mounting brackets will also be included to mount all external components, such as displays, sensors, etc., in locations to allow the system to function, however, installation will not be provided. The environmental requirements previously discussed will be rigid enough to survive within any shipping vehicle.

Lifecycle Requirements

The system will provide ports for external components to allow for easy replaceability and recyclability. Most internal components will support replace/recycle with servicing from a technician. The system will support upgradeable components for optimized results provided that it integrates with the system software. The system software shall allow support across various hardware components provided that the hardware components do not require specialized software modification.

Functional Decomposition

Based on the requirements analysis, the system can be further broken down into system functions. These functions are the necessary actions that must be performed in order for the system to perform and maintain operation in the anticipated environment. The following functions were identified as necessary to satisfy the requirements set:

- Detects vehicles upon entry and exit.
- Determine and provide key vehicle information such as permit validity, vehicle make, model, year, color upon entry and exit.
- Determine and provide the time of entry and exit for each vehicle.
- Take pictures of the front and back of vehicles upon entry and exit.
- Count the amount of vehicles within the garage.
- Display the number of parking spots available within the garage on a physical display as well as on a web application.
- Display the predicted number of spots available throughout the day for any given day.
- Provide all gathered data regarding vehicles (time of entry/exit, pictures of front/back, permit validity, vehicle make, model, year, color) to the web application and make it accessible to the admin UI.
- Include sufficient mounting to withstand vibrations at 10 Hz from vehicle traffic within the parking structure.
- Detects the internal temperature level.
- Control temperature level to be within operable range.
- Detect humidity level.
- Control humidity to be within operable range.
- Provide adequate power protection such that the system will not be damaged from surges.
- Provide product in assembled, modular form.
- Make the UI easy to use for the end-user.
- Provide modules at a fair price point with a trade-in system to allow repair and subsequent resale with replaceable components.
- Write software not specific to one device to allow for future upgrades to components in future generations.

Subsystem Decomposition

Based upon the functions set out, subsystems were created so the system could achieve all functions. See Figure 2 below for the subsystem diagram.

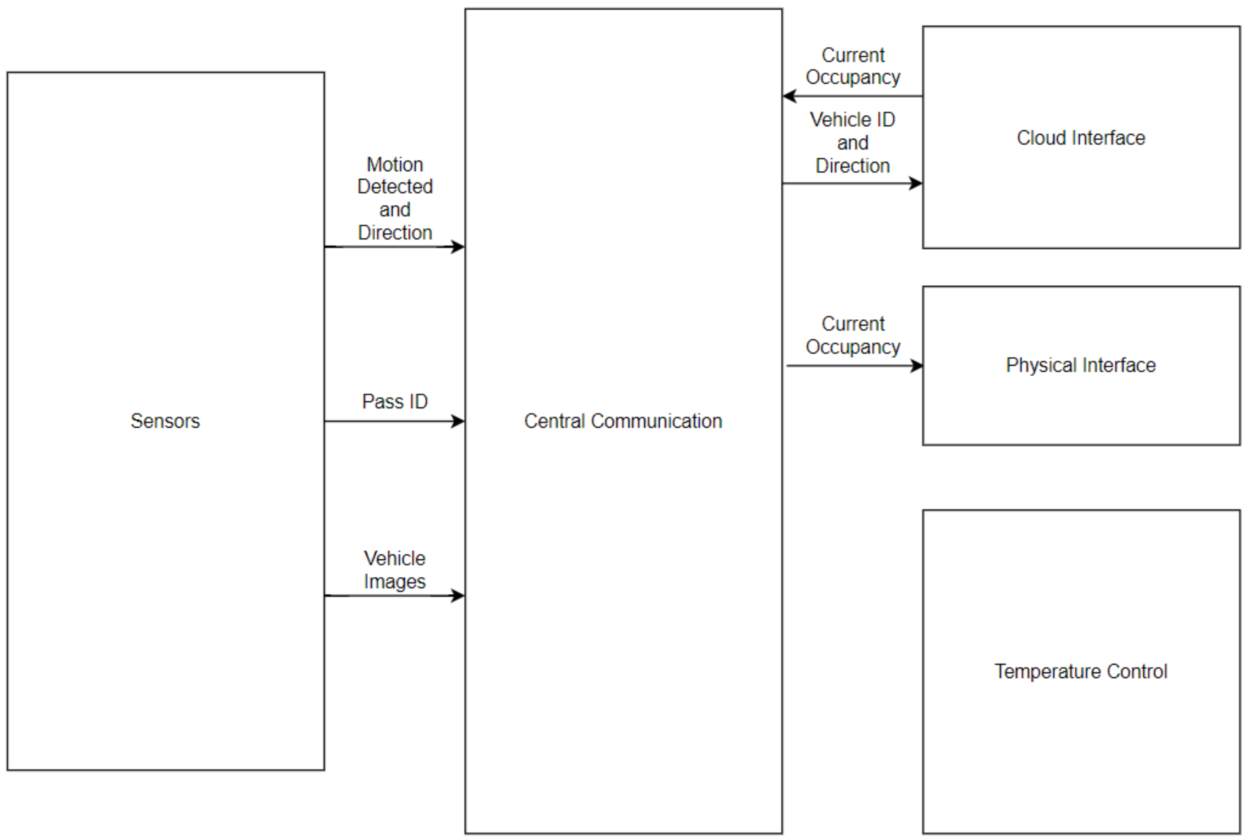


Figure 2. Subsystem Diagram

Sensors

This subsystem is responsible for collecting vehicle information upon vehicle entry or exit to the parking structure. If motion is detected, it will send data to the Central Communication including direction, detected pass ID, and images of the front and back of the vehicle.

Central Communication

This subsystem is responsible for delivering collected data from one node to the cloud so that the system can remain informed. When data is received from the sensors, it will be forwarded to the Cloud Interface subsystem. Additionally it will receive and forward the current occupancy information from the Cloud Interface to the Physical Interface.

Cloud Interface

This subsystem is responsible for handling data from multiple sensor nodes and organizing it in a presentable fashion to the end users. Upon receiving information from the Central Communication subsystem, it will populate the data in a table. The data will be delivered in a variety of ways to a variety of end users. Additionally, current occupancy will be derived from this table and sent to each sensor node.

Physical Interface

This subsystem is responsible for delivering real time occupancy to drivers near the garage. It will receive the current occupancy from the Central Communication and display that number on an informative GUI.

Temperature Control

This subsystem does not communicate with the other subsystems, but it still carries an integral role in the system. It is responsible for maintaining healthy operating temperature within each sensor node so that none of the other subsystems are affected by deprecated performance in the Central Communication subsystem.

System Interface

The system interface is a critical element in this design as communication between and within each subsystem can be well documented for future maintenance and improvements. It also helps show the functionality of each component within a given sensor node as they pertain to their subsystems.

Interface Diagram

Figure 3 below details the system interface between all subsystems and components.

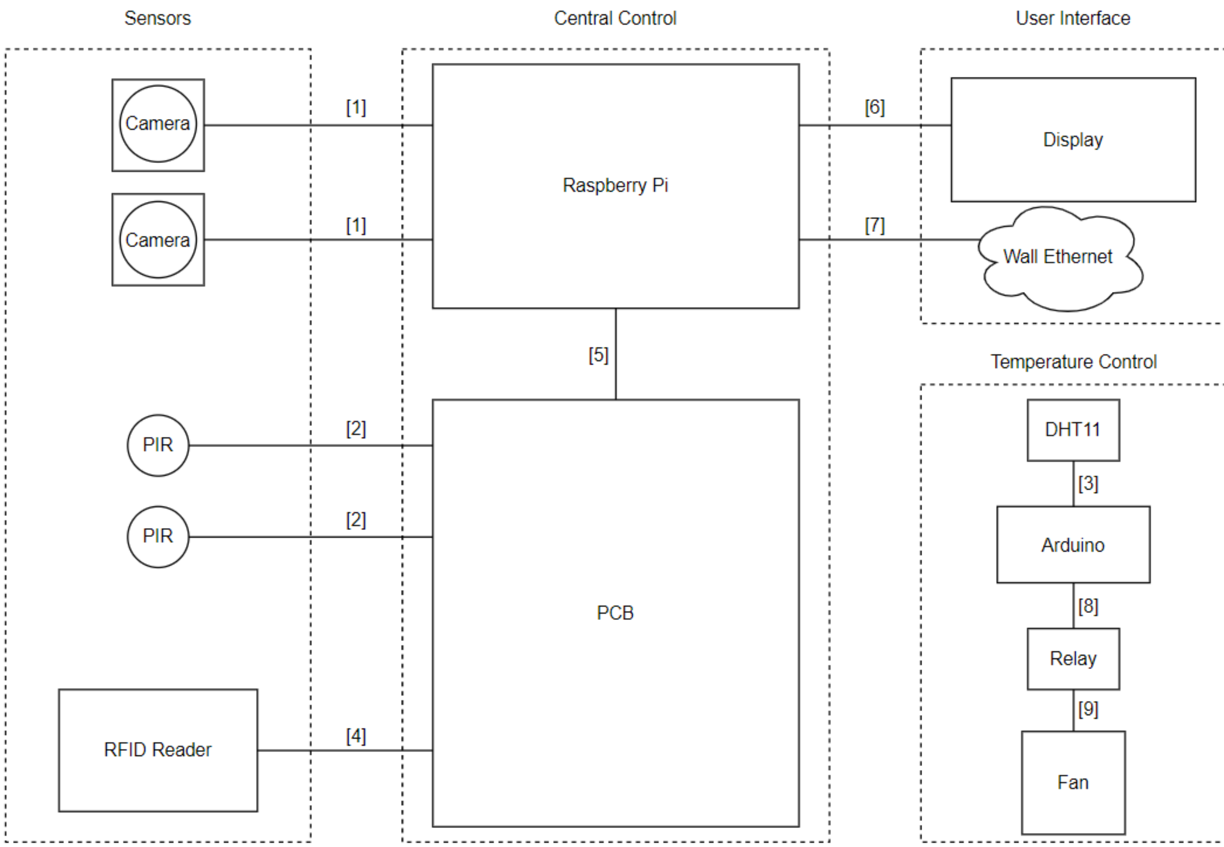


Figure 3. System Interface Diagram

Interface Connections

The connections of the system interface diagram are thoroughly detailed with the below tables to make sure there is no ambiguity with any system interfaces.

Table 1, System Interface Connections [1] through [4]

<table>
<tr><th>#</th><th>Description</th><th>#</th><th>Description</th></tr>
<tr><td>[1]</td><td>Camera to RPi
USB 2.0 Cable to USB 2.0 Port on Raspberry Pi
Data format: MJPG
Picture Transfer Protocol (PTP)
(See Attached Datasheet)</td><td rowspan="3">[4]</td><td rowspan="3">RFID Reader to PCB
7-wires
Ribbon Cable
Ribbon Connector
2.5V to 3.3V Power Supply
Hard reset with low power function
GND
MISO
MOSI
SCK
SS/SDA/Rx

Data format: Authentication command code (60h, 61h) + Block address + Sector key byte 0 + Sector key byte 1 + Sector key byte 2 + Sector key byte 3 + Sector key byte 4 + Sector key byte 5 + Card serial number byte 0 + Card serial number byte 1 + Card serial number byte 2 + Card serial number byte 3

Serial Peripheral Interface up to 10 Mbit/s

(See Attached Datasheet)</td></tr>
<tr><td>[2]</td><td>PIR Sensor to PCB
3-wire
Jumper Cable
Jumper Connection
5VCC - Red
OUT - 3.3V Pulse - Yellow
   HIGH (3.3V) motion detected
   LOW (0V) no motion detected
GND - Black
(See Attached Datasheet)</td></tr>
<tr><td>[3]</td><td>DHT11 to Arduino
3-wire
Jumper Cable
Jumper Connection
VCC - Red - 5V Pin
DATA - White - Pin 5
GND - Black - GND Pin
Serial Interface
Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T + 8bit checksum
(See Attached Datasheet)</td></tr>
</table>

Table 2, System Interface Connections [5] through [9]

<table>
<tr><th>#</th><th>Description</th><th>#</th><th>Description</th></tr>
<tr><td rowspan="4">[5]</td><td rowspan="4">PCB to RPi
RPi Terminal Bracket Connector
40 pins
PIR Route 1
Pin 1 - 3.3V
Pin 6 - GND
Pin 7 - GPIO 4 to OUT
PIR Route 2
Pin 1 - 3.3V
Pin 6 - GND
Pin 13 - GPIO 27 - OUT
RFID Route
Pin 17 - 3.3V
Pin 22 - RST
Pin 20 - GND
Pin 21 - MISO
Pin 19 - MOSI
Pin 23 - SCK
Pin 24 - SDA

*Communication protocols for PIR, and RFID are same as [2] and [4] respectively</td><td>[6]</td><td>RPi to Display
HDMI Cable
HDMI Port

Transition-minimized differential signaling (TMDS)

Data format: The 10-bit TMDS symbol can represent either an 8-bit data value during normal data transmission, or 2 bits of control signals during screen blanking.</td></tr>
<tr><td>[7]</td><td>Ethernet to RPi
Ethernet CAT-6 connecting Raspberry Pi ethernet port to external router
Open Systems Interconnection (OSI)</td></tr>
<tr><td>[8]</td><td>Arduino to Relay
3-wire
Jumper Cable
Jumper Connection
5V - 5V
Pin 10 - Signal
5V Standard Digital Logic
HIGH = Open
LOW = Close
GND - GND
(See Attached Datasheet)</td></tr>
<tr><td>[9]</td><td>Power to Fan
2-wire
12VDC - pass through relay
GND

12VDC power supply from surge protector</td></tr>
</table>

Physical Modeling

After the completion of the system interface, the first prototype - the physical model - was developed. Although this model does not have any signals passing through it, it served the purpose of providing an initial look at how all the components come together. This allowed for design changes to be made based on appearance and interaction.

Modeling Process

The materials for the model were decided at the beginning of the project, taking into account the size of the system and sensors. A shoebox was selected as the main housing for the system due to its suitability for housing the system. Craft foam was used for insulation and cut into shapes for other sensors. Jumper cables were used for most of the wires, while nylon string was used for longer wires to keep costs low. Cardboard and cardstock were used to construct other components that were more difficult to model.

All components were researched and modeled to scale, ensuring that the ports of each component were in the correct location and all wires were properly connected. After all individual components were accurately built, the unit was assembled using hot glue to attach everything together.

Design Challenges

The physical modeling process allowed for better assessment of several components of the design as problems could be identified physically. The LED lights were initially included to provide better photos for the cameras, but were later removed due to their large connectors crowding the container and their unnecessary nature. The first iteration of the physical model had a disorganized mass of wires, which was resolved by changing the connection from loose jumpers to a terminal bracket connector and adding through ports for all components outside of the container. This made for easier connections during assembly and maintenance, as well as a more refined interior. The placement and rigidity of the wires was also considered to ensure that ports such as USB and HDMI did not experience excessive stress.

Physical Model

The final physical model was created after several iterations of trial and error. The model shown below accurately represents all the components of the system. The sensors on the left are arranged in the same order they would be mounted and are clearly distinguishable. The display is a real-size model and there are also representations of a wall and an Ethernet port. The model is designed to be versatile and adaptable to fit the needs of different parking garages.

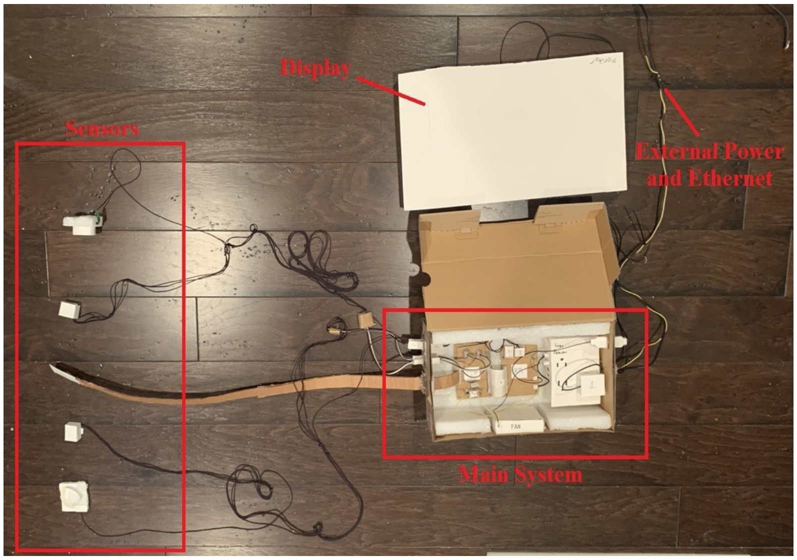


Figure 4. Physical Model (Full)

The inside of the physical model reveals that all interior components have been accurately modeled to scale, and the location of all ports corresponds with their real-life counterparts. The impact of using different types of connectors is also apparent, as it is easy to see where each connection should be made, and they can be plugged and unplugged with ease when necessary.

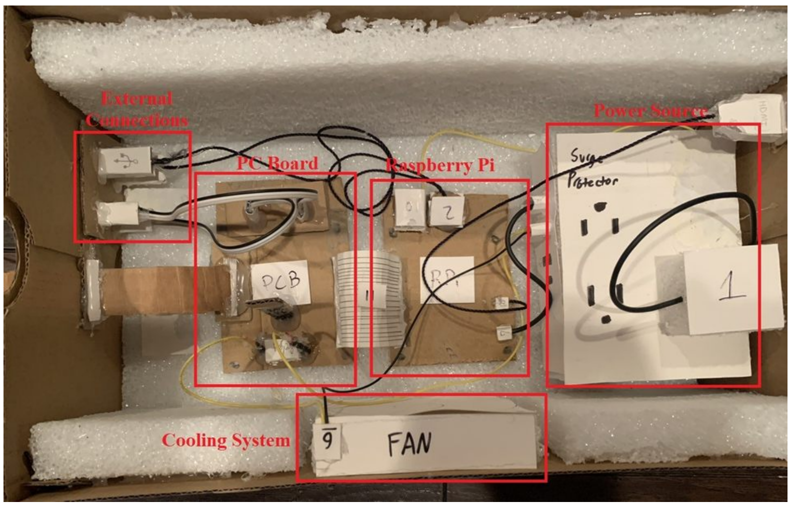


Figure 5. Physical Model (Inside)

Feasibility Analysis

As a smart parking garage, sensor integration is one of the most valuable components of the project. Two of the more complex sensors selected for feasibility analysis are the RFID reader and the PIR sensor, which are vital to the system's functionality for vehicle detection and authentication. Going into the feasibility analysis, questions regarding each of these sensors needed to be addressed aside from determining basic functionality.

For the RFID reader, the range or clearance necessary to read from a specified tag needed to be determined. Multiple RFID tags were tested for range by measuring whether or not the tag could be read from different distances. The "large white" RFID tag provided the greatest range for the project with up to 40 mm of detection.

Concerning the PIR sensor, there was concern about the sensor picking up too much motion. A housing was designed and manufactured for it to control the area it detects motion. Functionality was verified by reading output from the sensor with and without the housing at measured distances up to 200 mm or well beyond the range of the RFID reader. A full measurement for the PIR's range of detection with the housing attached was also obtained by bringing an object closer and closer to the PIR sensor until motion was detected. Each spot was marked down and repeated at different locations until an accurate approximation could be made, and the range of detection was determined to have an angle of approximately 12.81°.

After performing these feasibility tests, confidence was gained in the implementation of the RFID and PIR sensors into the design and interfacing with them to receive the desired data from vehicle entering or exiting.

RFID Reader

In the system, the RFID reader allows for vehicle information regarding plate number and pass validity to be passively gathered from vehicles entering or exiting the garage. As the design is constrained by price, an ultra high frequency RFID reader is not available which would provide an ample range of detection. Instead, the much more affordable RC522 reader will be used. This sensor has a much lower range of detection, so a measurable range of detection was desired. The range can change based on the tag it is reading, so a variety of tags were tested to optimize the range of the RFID readings to prevent any data from being lost. To begin, two different tags were identified, pictured below, to test on.

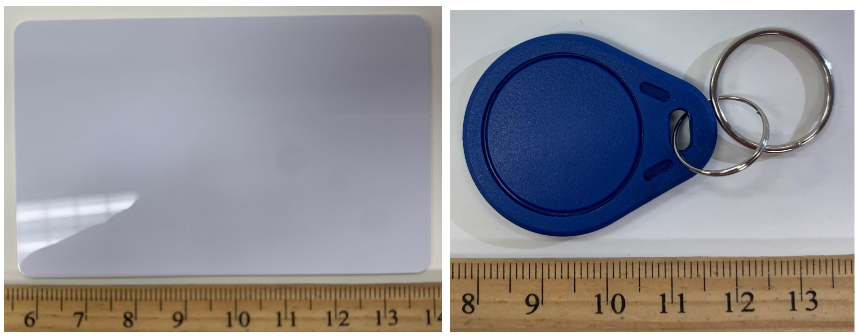


Figure 6. “Large White” and “Small Blue” RFID Tags

To ensure the tags could be easily distinguished during testing, the names we assigned to them were written to each specific tag. As can be seen in the below screen captures, the "Large White" and "Small Blue" tags were successfully written to.

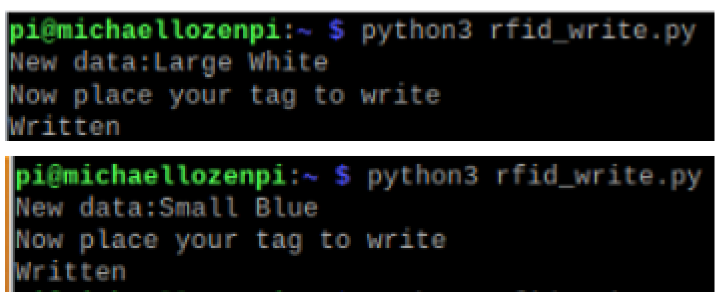


Figure 7. Writing Message to Each RFID Tag

For the two tags, the read functionality was verified, as can be seen below. Aside from the message written to the tags, there was also a unique tag ID present which could be used in the future for a more robust system. The status of both this ID and the message were monitored during testing.

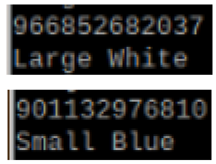


Figure 8. Reading Message from Each RFID Tag

Now that basic functionality of the RFID tags had been confirmed, range testing could begin. The tags were raised in increments of 5 mm (starting at 0 mm) and read from, with the results recorded until no information could be gathered. To measure the distance, a ruler was placed perpendicular to the RFID sensor. The testing results are shown in the below table and figures provide a visual of the testing setup.

Table 3, RFID Range Testing Results

Distance (mm)	Large White Tag	Small Blue Tag
0	Functional	Functional
5	Functional	Functional
10	Functional	Functional
15	Functional	Functional
20	Functional	Functional
25	Functional	Tag ID, No Message
30	Functional	No Detection
35	Functional	No Detection
40	Tag ID, No Message	No Detection
45	No Detection	No Detection

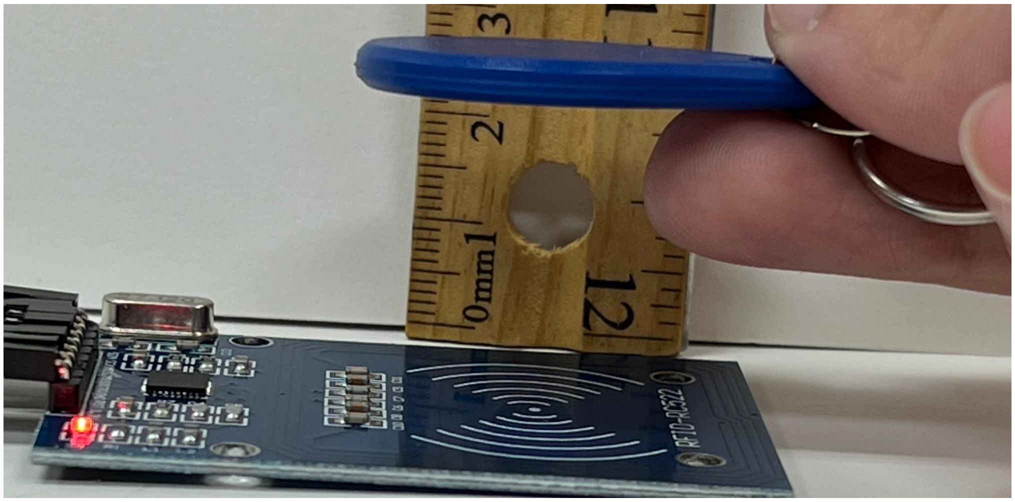


Figure 9. RFID Range Testing Setup with “Small Blue” Tag at Edge Range (25 mm)

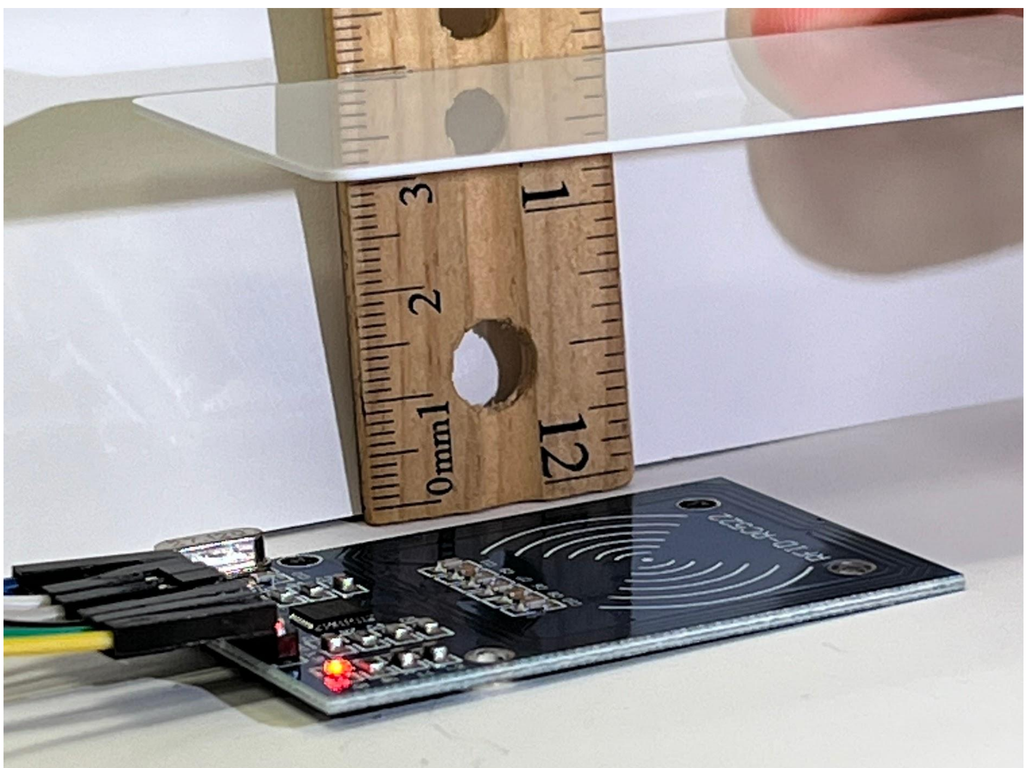


Figure 10. RFID Range Testing Setup with “Large White” Tag at Edge Range (40 mm)

Based on these findings, it is clear that the “Large White'' tag provides the most amount of range. The design needs to be constrained to have passing tags within its operating range (<40mm) to acquire the full amount of information for each passing vehicle, which includes plate number and pass validity.

PIR Sensor

The PIR Sensor in the system detects vehicle movement in and out of the parking garage with high accuracy, which is essential for maintaining an active count of the number of vehicles in the garage and triggering the RFID reader to collect further information. The HC-SR501 PIR sensor was chosen for our application.

When testing the PIR sensor, it was observed that the cone of detection was larger than desired, detecting motion outside of the intended area. To address this, a housing was designed and manufactured to restrict the cone of detection and improve accuracy.

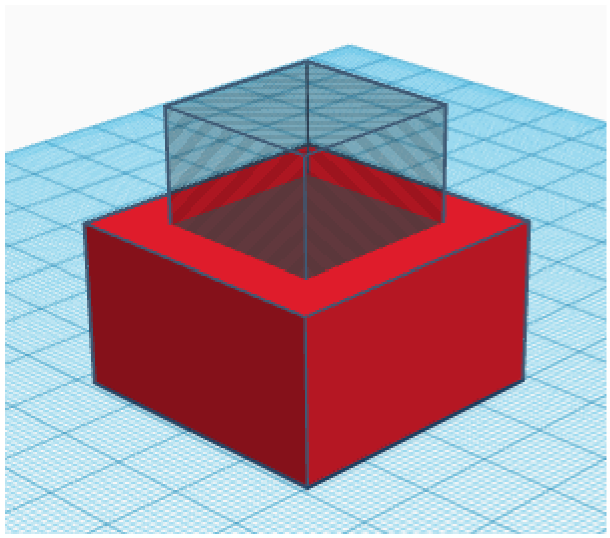


Figure 11, PIR Housing 3D Model

The designed housing was utilized to focus the range of detection of the PIR sensor, aiming to ensure that (1) motion could still be consistently detected from a far enough range, and (2) the cone of detection was of an appropriate size.

The first test ensured that the PIR sensor can operate from an appropriate distance with the housing. The distance was incremented by 50 mm up to 200 mm (measured by ruler) to ensure motion can be detected well beyond the requirements of the RFID sensor's range. See the below table for results and figures for setup.

Table 4, PIR Range Testing Results

Distance (mm)	No Housing	Housing
0	Motion Detected	Motion Detected
50	Motion Detected	Motion Detected
100	Motion Detected	Motion Detected
150	Motion Detected	Motion Detected
200	Motion Detected	Motion Detected

Figure 12, PIR Range Testing Setup without Housing

Figure 13, PIR Range Testing Setup with Housing

Now that basic functionality of the PIR with the housing was verified, the cone of detection of the sensor with the housing was tested. This was crucial in the design to ensure an optimally sized cone of detection to consistently detect motion where required and not detect far away. Ultimately, this would help determine how far to separate the motion detecting sensors to properly detect motion and direction.

To conduct the test, an object (ruler) was brought slowly towards the sensor from the side to mark the edge of the range. The ruler was then brought further away, repeating the process until accurate markings could be made to predict where motion would and wouldn't be detected. This was repeated across both sides so that a 2D slice of the cone of detection could be created. From there, the angle was measured by taking the tangent of the horizontal and vertical lengths for both sides and averaging. The table below shows the results and the setup figure.

Table 5, PIR with Housing Angle of Detection Measurements

Side	Horizontal (cm)	Vertical (cm)	Angle of Detection
Left	5.8	25.4	12.81°
Right	5.8	25.4	12.81°
Average	5.8	25.4	12.81°

Figure 14, PIR with Housing Angle of Detection Measurement Setup

The rendering of the range of detection in the figure below shows a cone-shaped area directly below the PIR sensor. It indicates the area where the sensor can detect motion. Based on the measurements taken, an angle of 12.81° was determined to be the ideal cone of detection for this application. This angle allows for accurate detection of a large area below the sensor while intentionally creating blind spots for any stray movements outside the intended area.

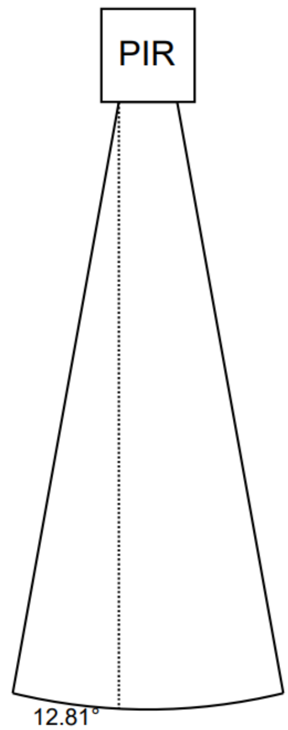


Figure 15, PIR with Housing Range of Detection

Based on the findings, the design will need to be constrained to provide enough space between the two PIR sensors in-line to ensure that their reading ranges are not overlapped. However, they should be close enough together to accurately detect motion from either side and determine direction as an array.

System Development

The development process for this system involved the areas of the sensor housing, main housing, PCB design, temperature control system, data collection system, publisher-subscriber connection, predictive model, and the user interface. The below subsections detail the design and functionality of these subsystems as they pertain to the system as a whole.

Sensor Housing

Sensor Spacing

In order to properly simulate the functionality of the sensor node design at a smaller scale, a sensor housing that is capable of accounting for the detection ranges of the sensors needed to be constructed. Using the sensor ranges of the PIR and RFID sensors found in the feasibility analysis, the sensor spacing was determined as shown in Figure 16 below. The spacing allowed for a model car with a height of approximately 35 mm and an RFID tag on top to pass through the housing and be well within the range of each sensor throughout its passage through the sensor housing. This was an essential part of the system functionality, as every vehicle entering had to be properly detected and verified to keep an accurate count and inventory of the parking structure from the sensor nodes.

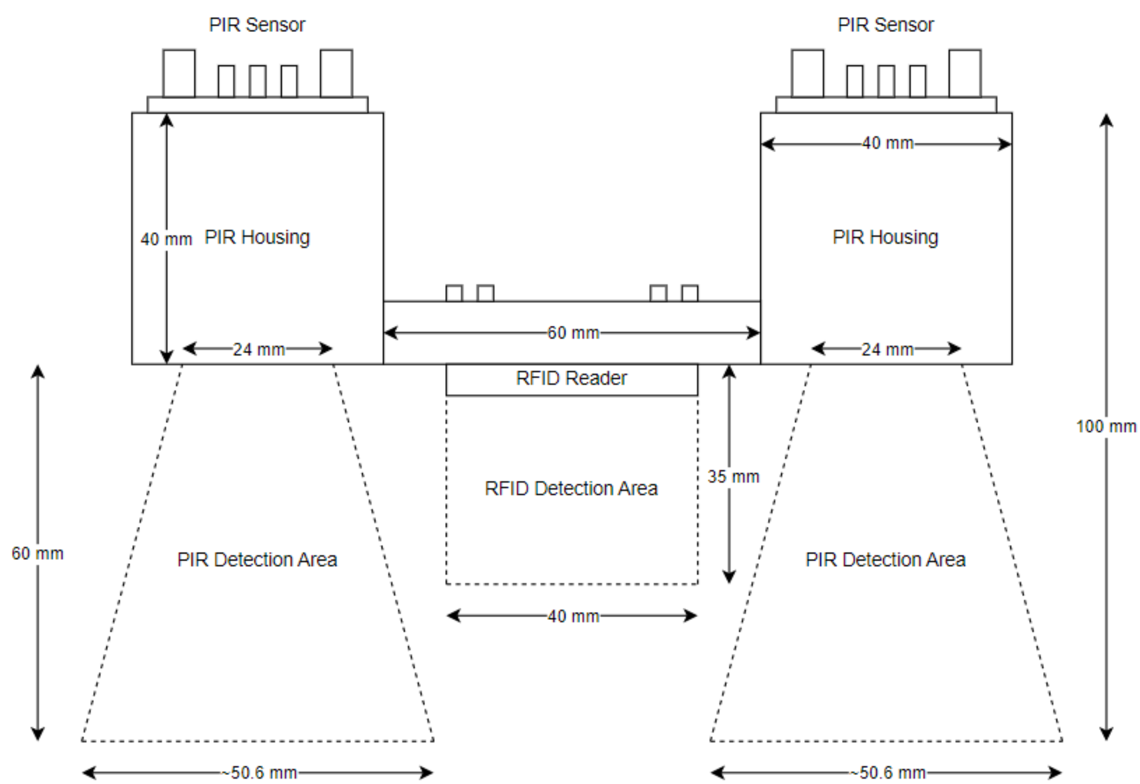


Figure 16. Sensor Spacing Diagram

Sensor Housing Design

After the proper sensor spacing was determined, the sensor housing was ready to begin development. The first step in this process was to properly design a structure that would allow the aforementioned sensor spacings to be realized after manufacturing. The sensor housing was developed in four parts in Solidworks with a main housing, large side wall, and two smaller side walls. These parts would form one assembly with tight fitting notches that would allow the sensor housing to act as a “tunnel” for the vehicle to pass through and be properly detected every time. The width of this “tunnel” was also considered for this design so that the vehicle would not be able to avoid any of the sensors when passing through. Figures 17 through 19 below show each of these parts’ designs in Solidworks.

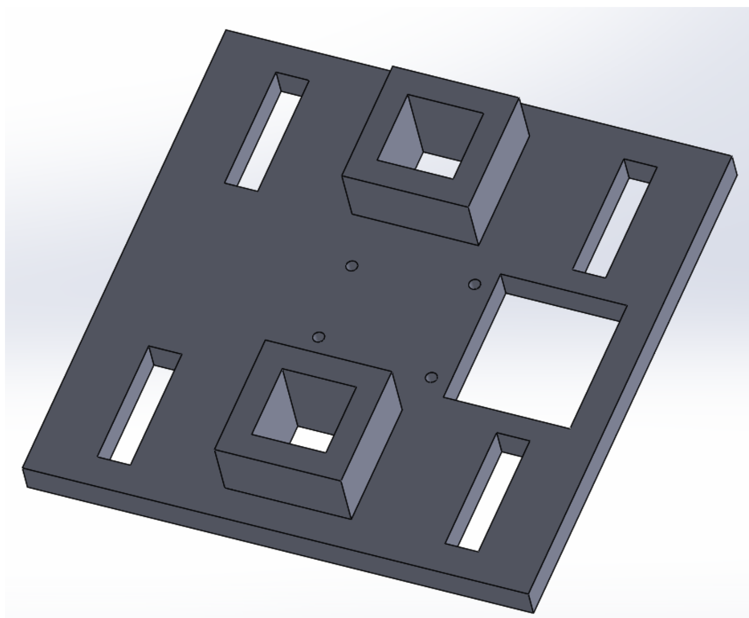


Figure 17. Main Sensor Housing

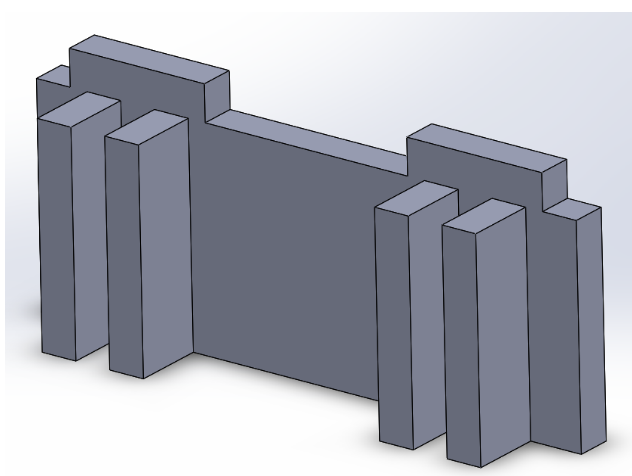


Figure 18. Large Side Wall

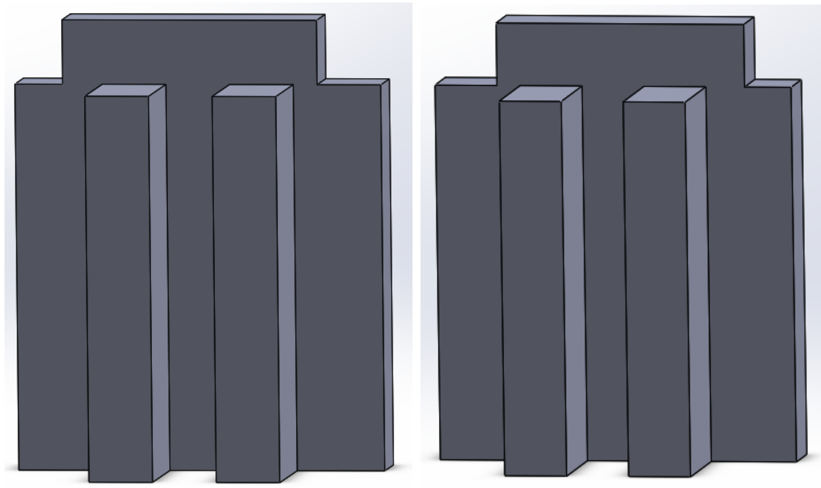


Figure 19. Short Wall 1 and 2

Sensor Housing Manufacturing

After the 3D modeling process was complete, the parts were ready for manufacturing. The parts were 3D printed as this process provided high accuracy development at a low material cost and in a relatively fast manner. To do so, the Solidworks part files were sent to a slicer which generated the GCODE file type to be interpreted by a 3D printer. The parts were then printed on a Creality Ender 3 Pro using PLA+ filament. The printing process took just under a week altogether, as the high precision option was used and a few prints had to be restarted due to warping. Figure 20 below shows the final sensor housing with the sensors attached.

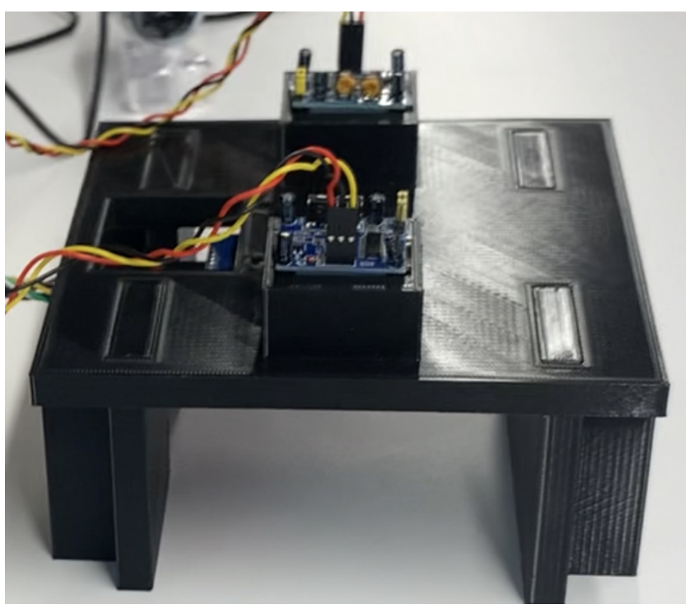


Figure 20. Final Sensor Housing

Main Housing

The design of the housing for the sensor nodes is a critical component of the overall system. It not only needs to house all of the interior components of the sensor nodes in an efficient and safe manner, but also needs to be durable enough to withstand the harsh conditions commonly found in a parking structure.

To ensure that the housing design met these requirements, significant inspiration was taken from the physical modeling performed in both functionality and layout. This allowed the team to identify the key locations and spacings needed for the embedded system (Raspberry Pi), PCB, surge protector, temperature control system, and power supplies prior to development. This ensured that the housing design would allow for simple installation and maintenance of these components.

The housing also needed to be closed but accessible to allow for occasional maintenance. Therefore, the design included ports for exterior components and sensors to add additional ease to the installation and maintenance processes. This not only made it easier to install and maintain the sensor nodes, but also helped to reduce downtime and increase the overall efficiency of the system.

Finally, the housing needed to be rigid enough to stay durable to the shaking conditions commonly found in a parking structure, but light enough to be mounted from the ceiling by brackets. This required careful consideration of the materials used in the housing design and the structural integrity of the overall design. A balance needed to be struck between durability and weight to ensure that the housing would be able to withstand the harsh conditions of a parking structure while also being easy to install and maintain.

The design for the housing of the sensor nodes incorporated the use of aluminum panels and 3D printed corner and edge fasteners. Aluminum was chosen as the primary material for the housing due to its light yet rigid properties, making it an ideal material for a housing that needed to be mounted from the ceiling by brackets. Its lightweight nature would allow for easy installation and maintenance, while its rigidity would ensure that the housing could withstand the shaking conditions commonly found in a parking structure.

In addition to the aluminum panels, 3D printed corner and edge fasteners were utilized to securely fasten the six faces of the housing together. This provided a low-cost solution that allowed for the connectors to be a tight fit. The use of 3D printing also allowed for easy customization of the connectors to ensure that they fit the specific dimensions of the aluminum panels. This ensured a tight fit between the panels, further enhancing the overall rigidity of the housing design.

Panel Design

Before cutting the aluminum to shape, the dimensions of each face along with individual screw holes, through ports, or opening were to be determined. These dimensions and locations for each face were designed on Solidworks as part files. Since the panels were to be cut from an aluminum sheet of 2 ft x 3 ft (610 mm x 715 mm), the dimensions of each panel were to be restricted to all be cut from the same sheet. This still allowed ample spacing for all components inside the housing. Figure 21 below shows the dimensions of each panel to be cut from the aluminum sheet. Figures 22 through 27 show the individual screw holes, through ports and openings for each panel as designed.

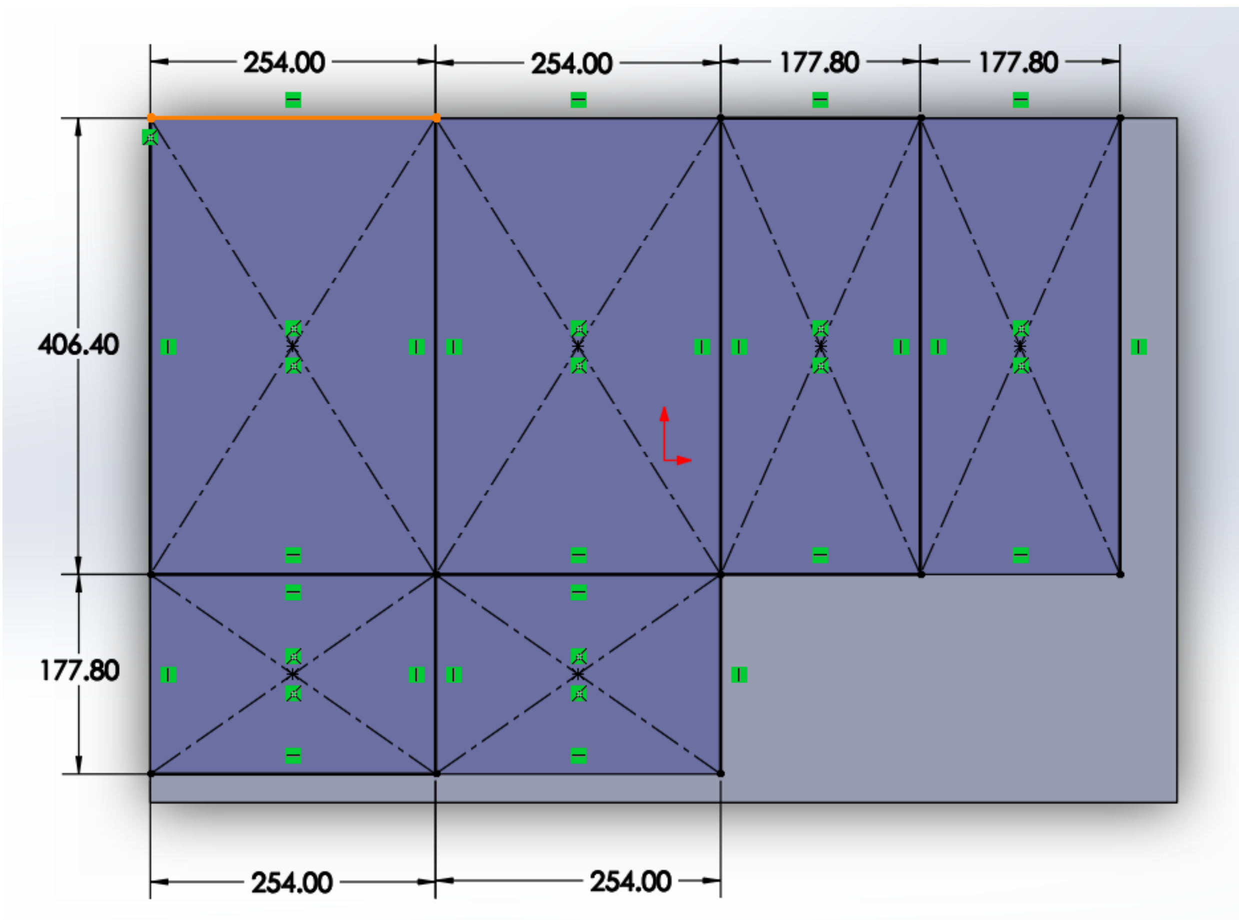


Figure 21. Panel Dimensions

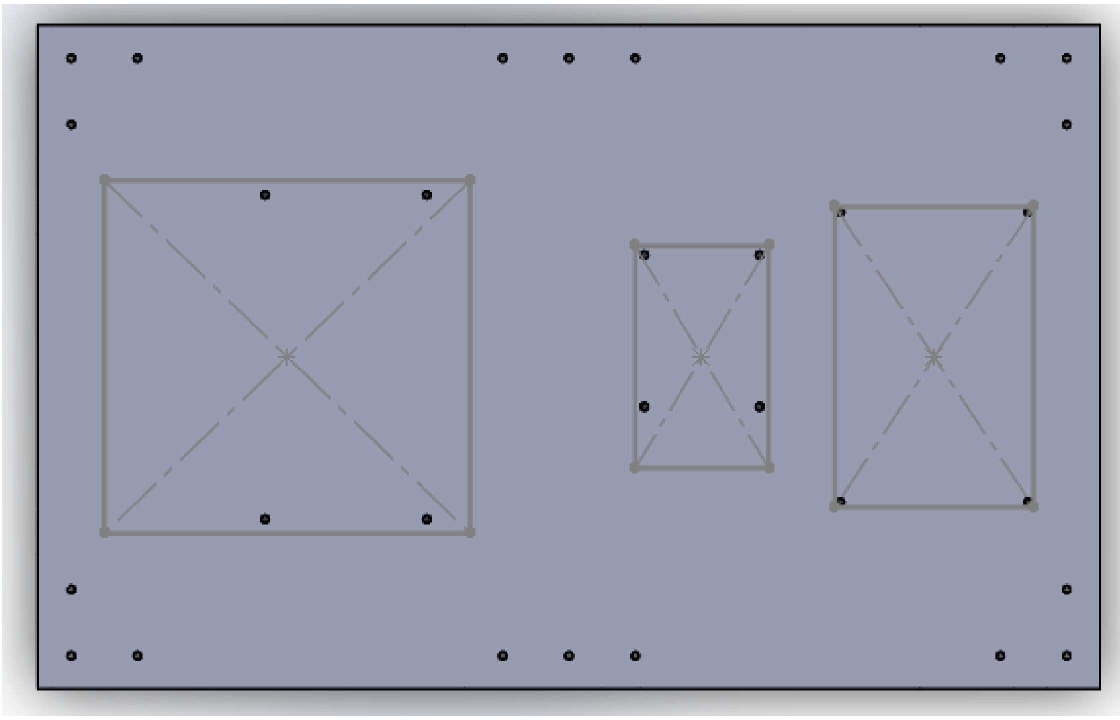


Figure 22. Bottom Panel

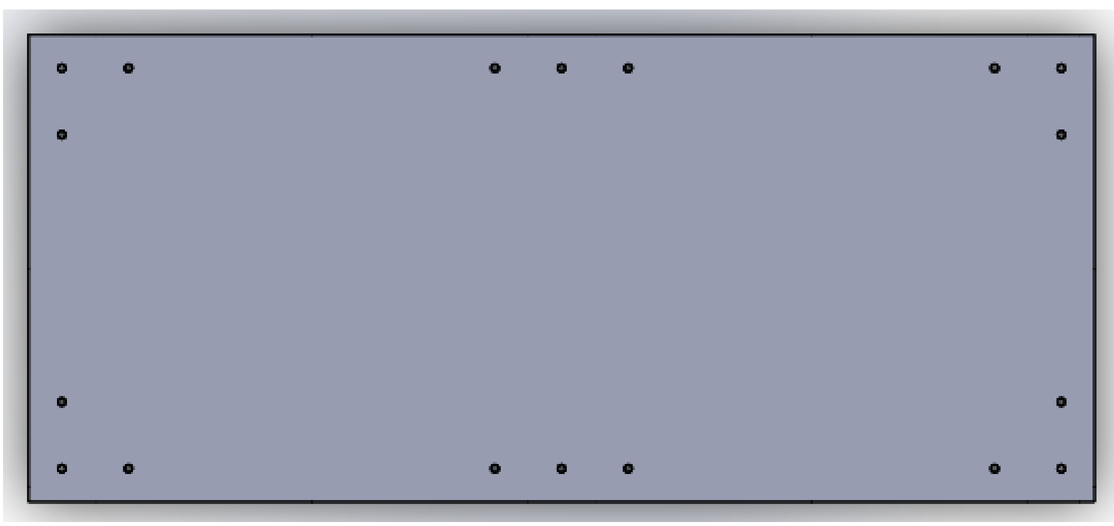


Figure 23. Side Panel

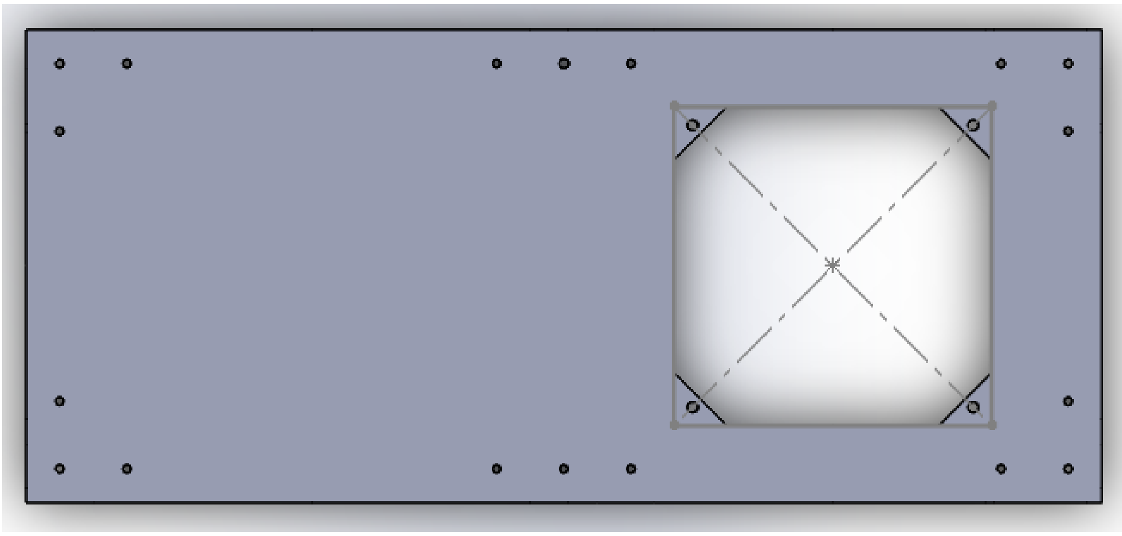


Figure 24. Side Panel with Fan Opening

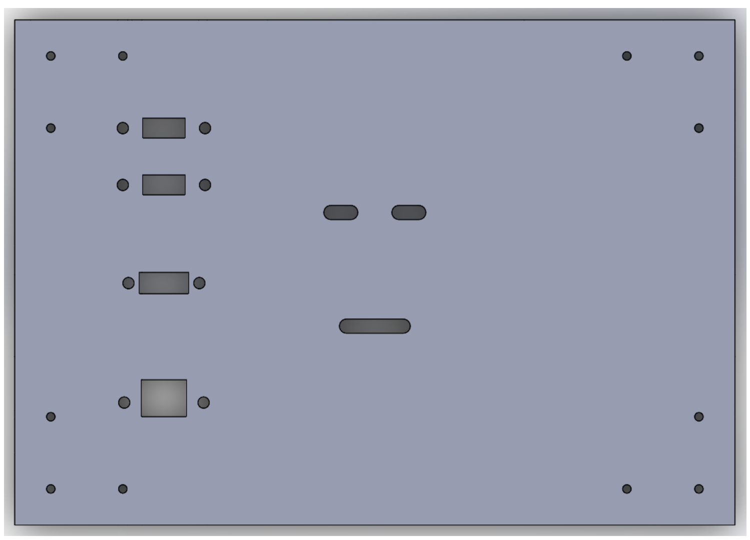


Figure 25. Front Interfacing Panel

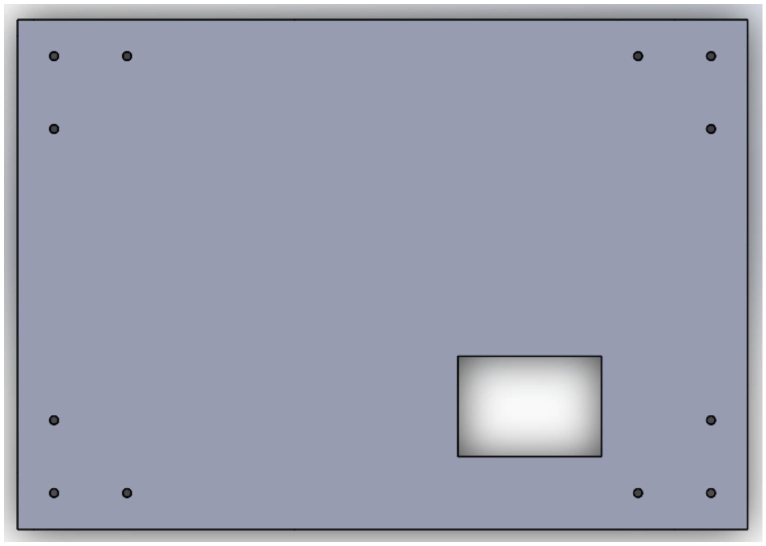


Figure 26. Rear Power Access Panel

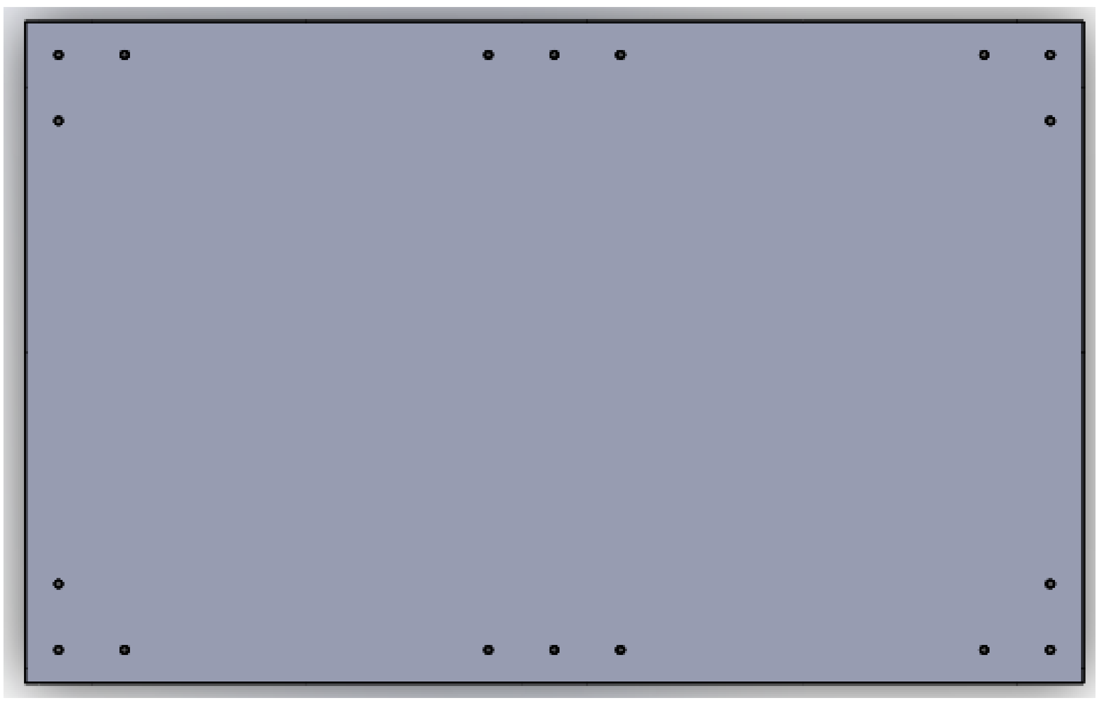


Figure 27. Top Panel

Panel Manufacturing

After the design of the aluminum panels for the sensor node housing was completed in Solidworks, the next step was to prepare them for manufacturing. The part files for the panels were exported as DXF files for compatibility with the waterjet software. These files were uploaded to the waterjet software and a path was generated for cutting the panels from the sheet. By utilizing waterjet cutting, the aluminum panels were cut with precision and accuracy, ensuring that they were suitable for use in the sensor node housing. This allowed for a seamless integration of the panels into the final product, enhancing the overall functionality and performance of the sensor nodes in a parking structure environment. Figure 28 below shows the waterjet cutting process of the panels from the sheet. Figure 29 shows one of the panels cut from the sheet immediately after cutting.

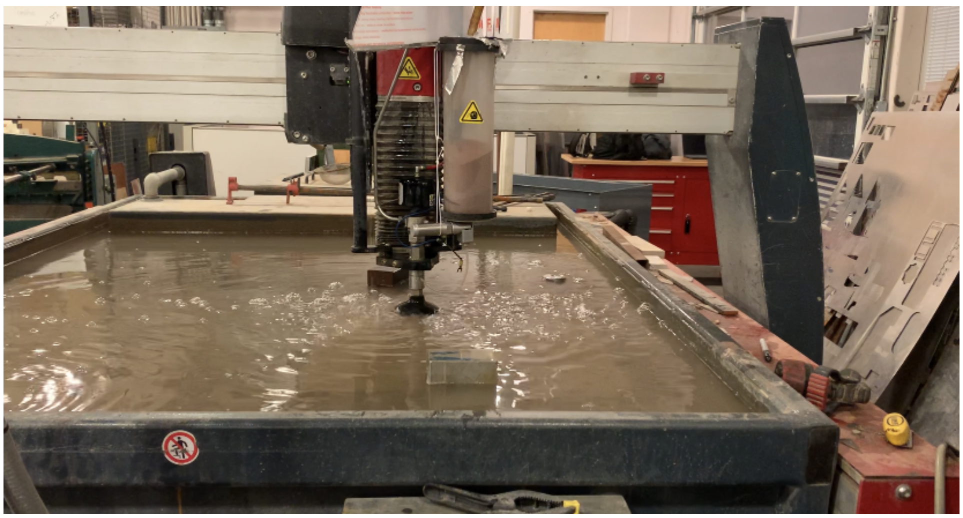


Figure 28. Waterjet Cutting Panels from Sheet

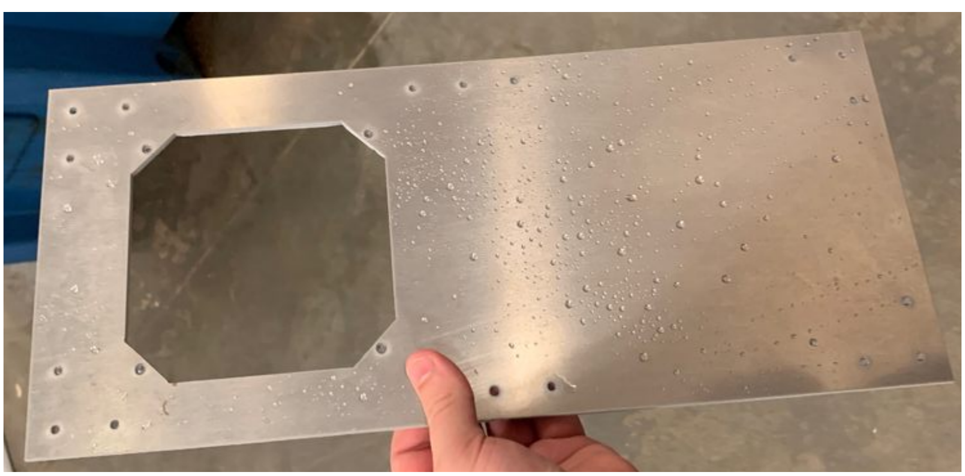


Figure 29. Waterjet Cut Panel

Panel Fastener Design

To connect the panels to perform a housing, panel fasteners were to be designed. As stated earlier, these would be 3D printed to allow for precision and customizability. To properly fasten the panels into a box that could be opened from the top, four different types of fasteners were necessary: corner, left corner passthrough, right corner passthrough, and edge. These were developed on Solidworks and can be seen below in Figures 30 through 33.

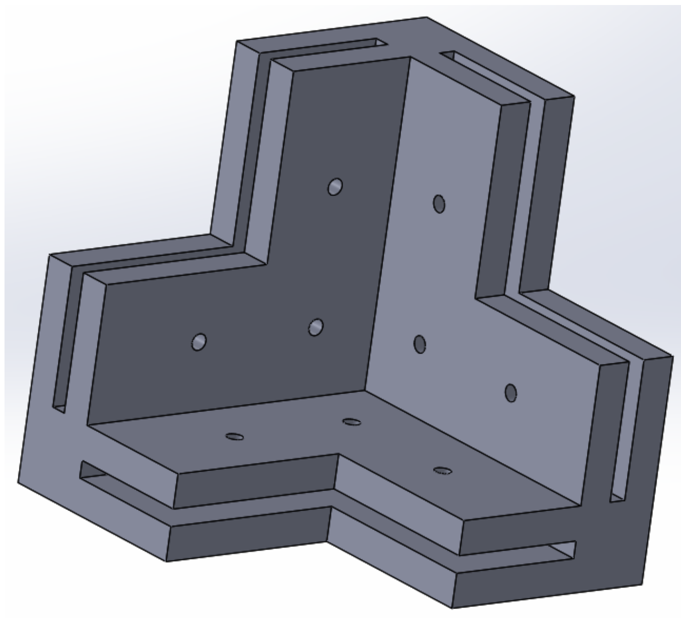


Figure 30. Corner Fastener

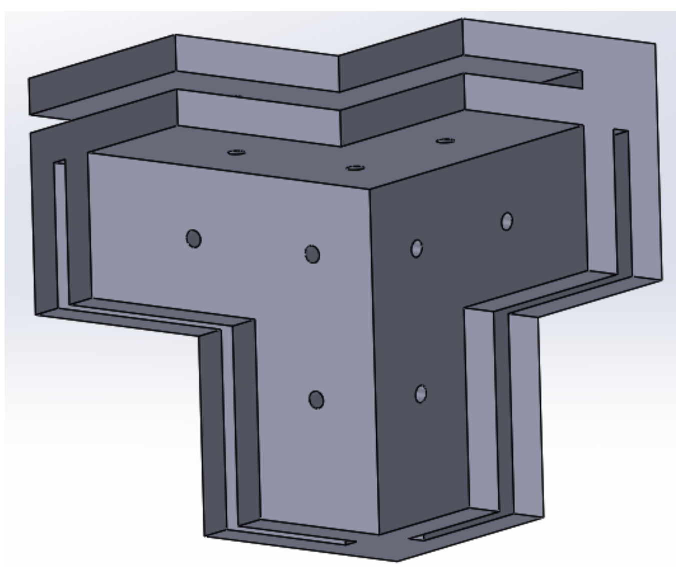


Figure 31. Right Corner Passthrough

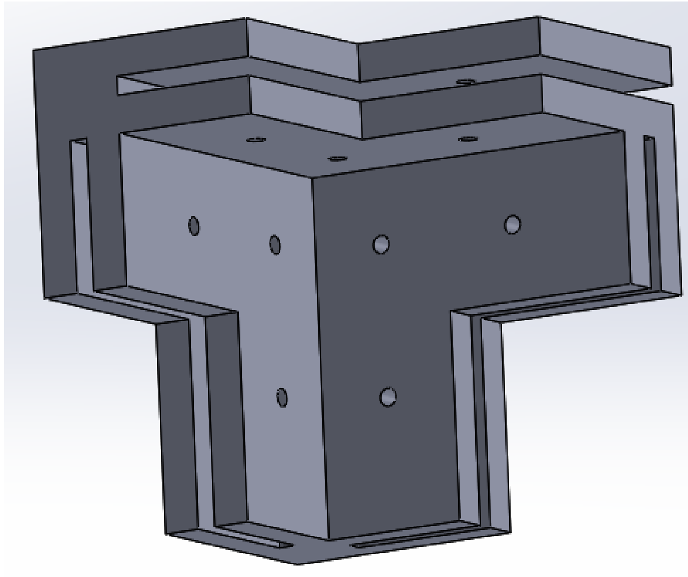


Figure 32. Left Corner Passthrough

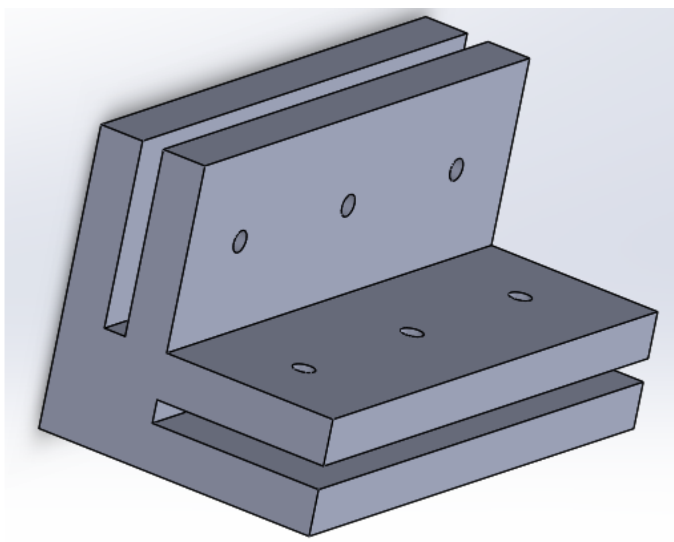


Figure 33. Edge Fastener

Panel Fastener Development

The housing required 4 corner fasteners. 2 left corner passthrough fasteners, 2 right corner passthrough fasteners, and 4 edge fasteners. The Solidworks part files for each of these were sent to a slicer which generated the GCODE file type to be interpreted by a 3D printer. The parts were then printed on a Creality Ender 3 Pro using PLA+ filament. The printing process for each of these fasteners took about a week and a half as some parts were printed together.

Housing Assembly and Parts Integration

With all components for the housing developed, it was then assembled. Figure 34 below shows the assembled version with only the fasteners (not every fastener as some were still being printed at the time) and panels.

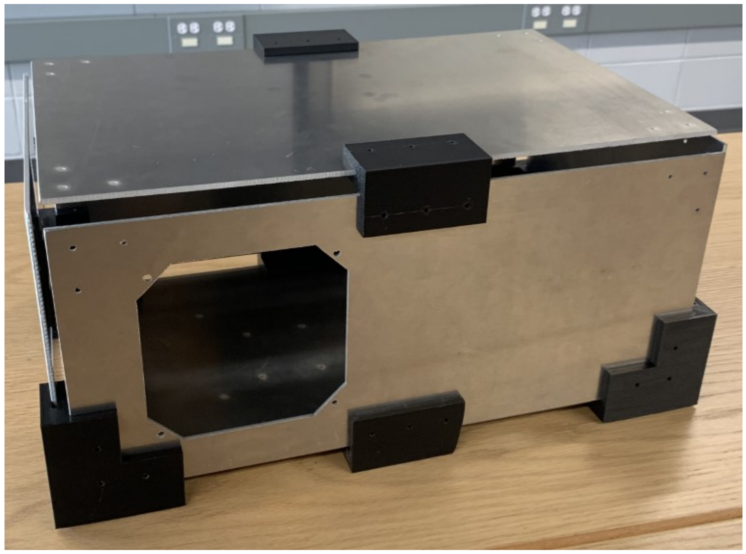


Figure 34. Assembled Housing (Only Panels and Fasteners)

To help maintain safe temperature levels inside the housing for the electronics, insulating foam and a fan were installed onto the housing. Figure 35 below shows the inside of the housing after this installation.

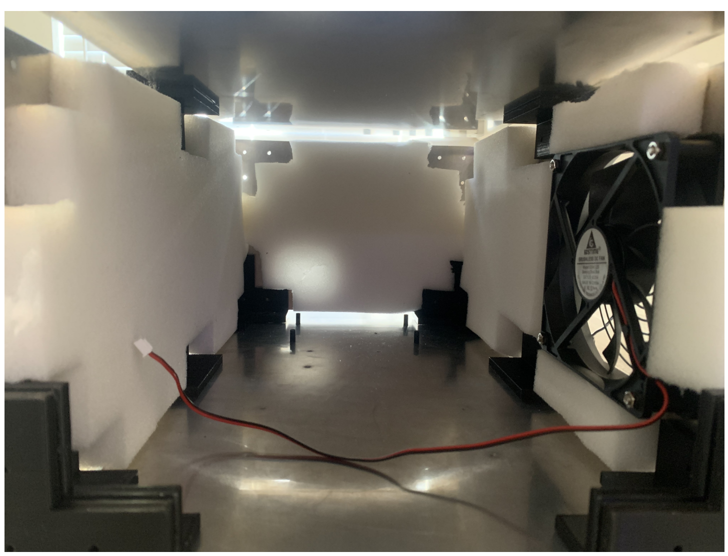


Figure 35. Interior of Housing with Insulation and Fan Added

Finally, all components were added to the housing. Figures 36 through 39 show the final sensor node housing.

Figure 36. Interior of Sensor Node Housing

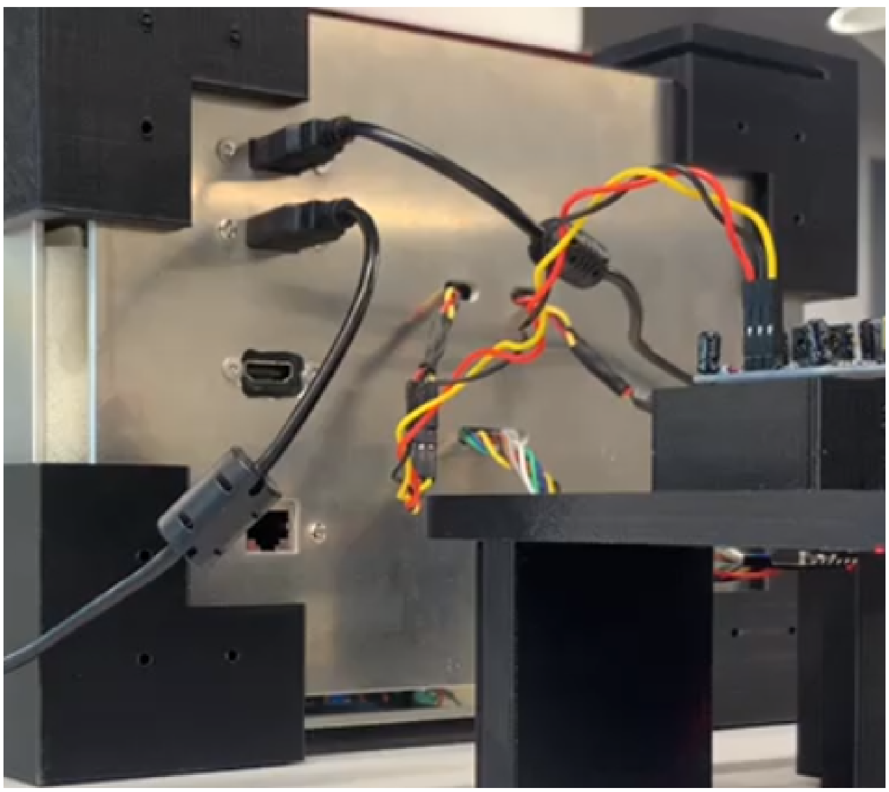


Figure 37. Exterior of Sensor Node Housing (Interfacing Panel)

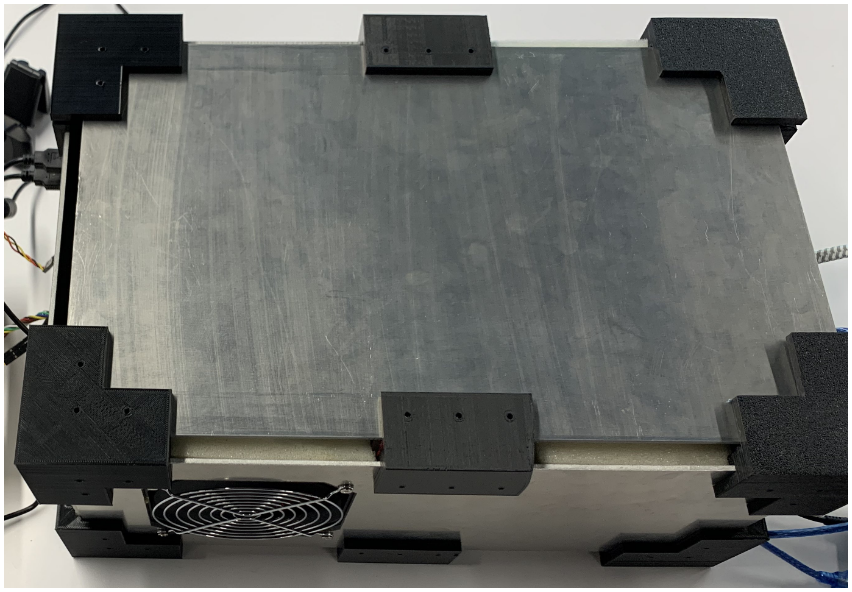


Figure 38. Exterior of Sensor Node Housing (Top)

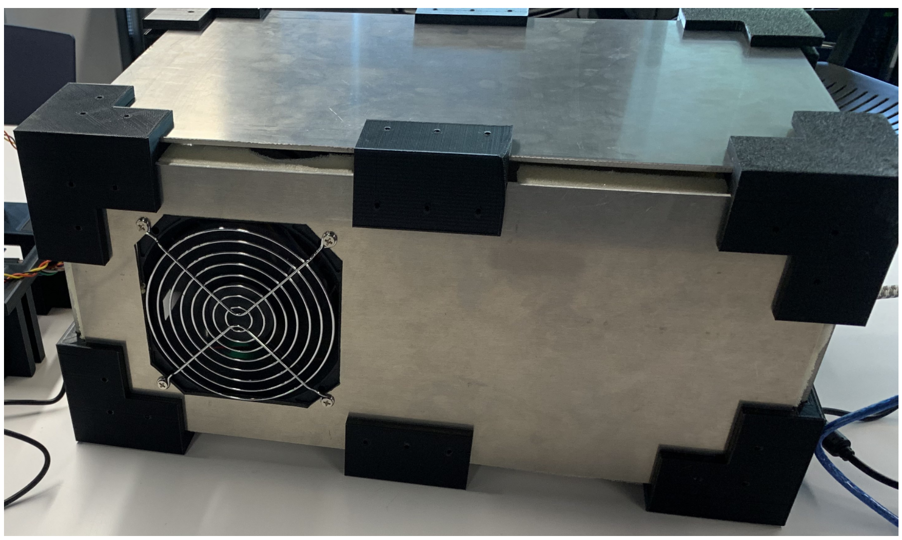


Figure 39. Exterior of Sensor Node Housing (Side)

Printed Circuit Board (PCB) Design

Circuit Schematic

Optimizing the system to be easily maintainable and repairable was an important aspect of the design. A printed circuit board (PCB) was designed to organize the wiring based on each sensor's function. The design process utilized the EasyEDA software tool, which allowed for the creation of a comprehensive circuit schematic that connected the Raspberry Pi's pinouts to each sensor. Figure 40. below shows the resulting schematic.

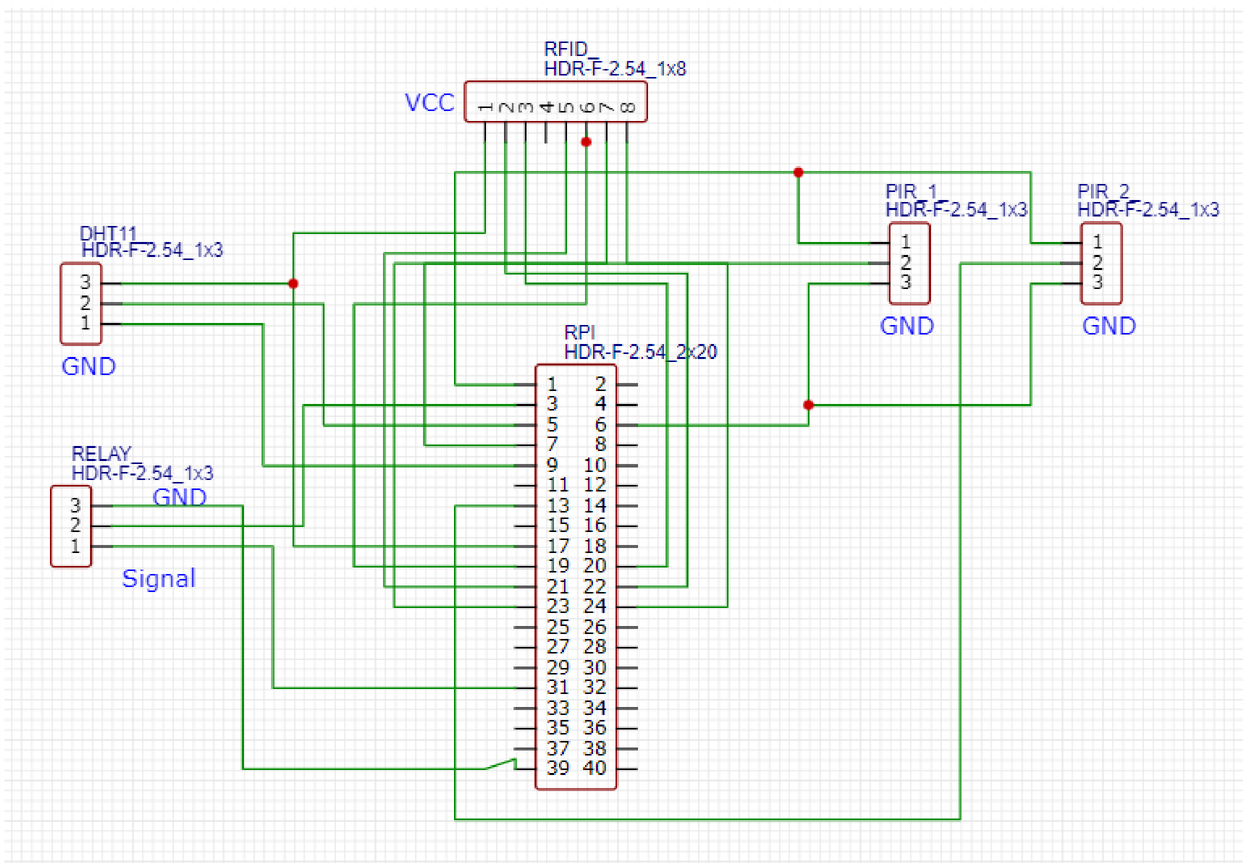


Figure 40. Circuit Schematic of PCB

PCB Layout

Following the completion of the circuit schematic, the next step was to create a PCB layout that organized the wiring based on each sensor's function. The PCB layout was designed by tracing the connection paths for each pinout, a task that presented some challenges such as avoiding overlaps. To overcome these challenges, the dimensions of the PCB were increased to allow for additional space for the traces. This increased the overall size of the board, which also required careful consideration of the available space within the system housing.

In addition to providing more space for the traces, mounting holes were incorporated into the PCB to facilitate the installation of the relay and the attachment of the board to the system housing. The resulting PCB effectively organized the wiring based on each sensor's function, allowing for easier identification and troubleshooting of any issues that may arise. Furthermore, the design contributed to the longevity and sustainability of the system, making it easier and more cost-effective to maintain and repair over time. Figures 41 and 42 below show the final design of the PCB.

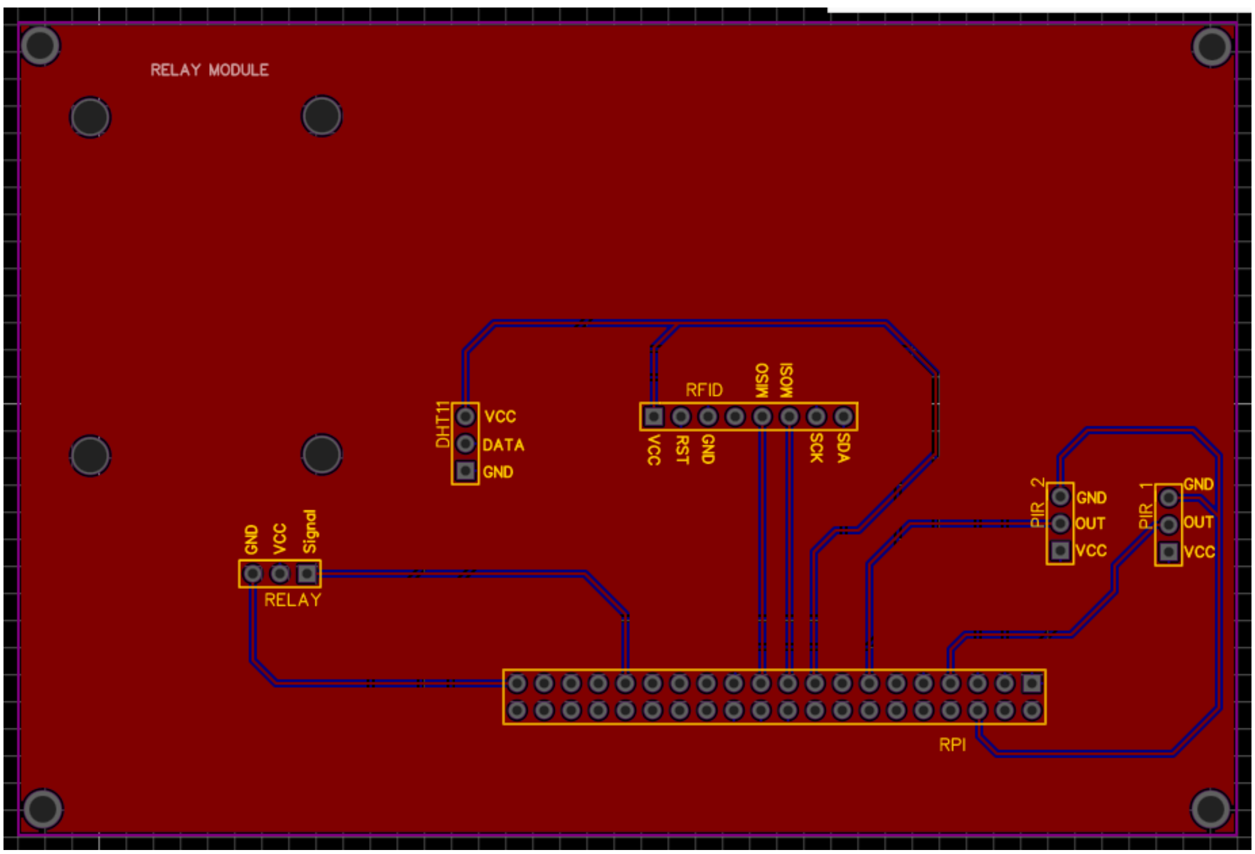


Figure 41. Top of PCB Design

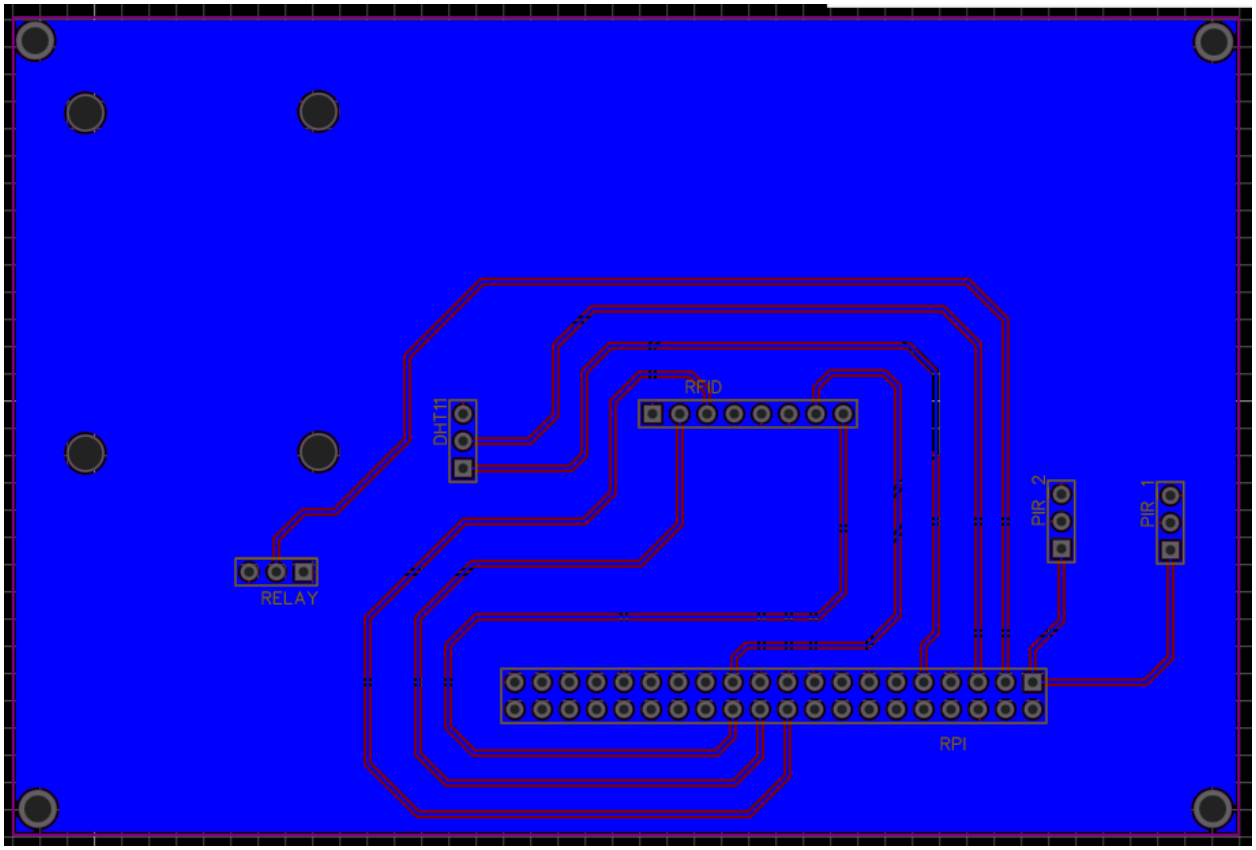


Figure 42. Bottom of PCB Design

Temperature Control System

Temperature control is a critical aspect of any electronic system, especially one that is outdoors, and the sensor nodes in a parking structure are no exception. In order to ensure that the electronics within the sensor nodes function properly, a temperature control system was designed and constructed. The system uses a DHT11 sensor to detect the temperature in real-time and an Arduino to process the data and control a fan based on the temperature readings.

If the temperature is detected above 80 degrees Fahrenheit, the Arduino sends a signal to a relay to close, turning on a fan to cool the housing. The fan helps to dissipate heat from the sensor node housing, which prevents overheating and potential damage to the electronics. To prevent hysteresis, the relay will remain closed until the temperature reaches below 76 degrees Fahrenheit.

Overall, the temperature control system provides an effective solution for maintaining the functionality of the sensor nodes in a parking structure environment, where the temperatures can reach high levels. This system ensures that the electronics within the sensor nodes are protected from overheating, leading to improved performance and longevity.

Figure 43. Below shows the electronics that drive the temperature control system from within the sensor node housing.

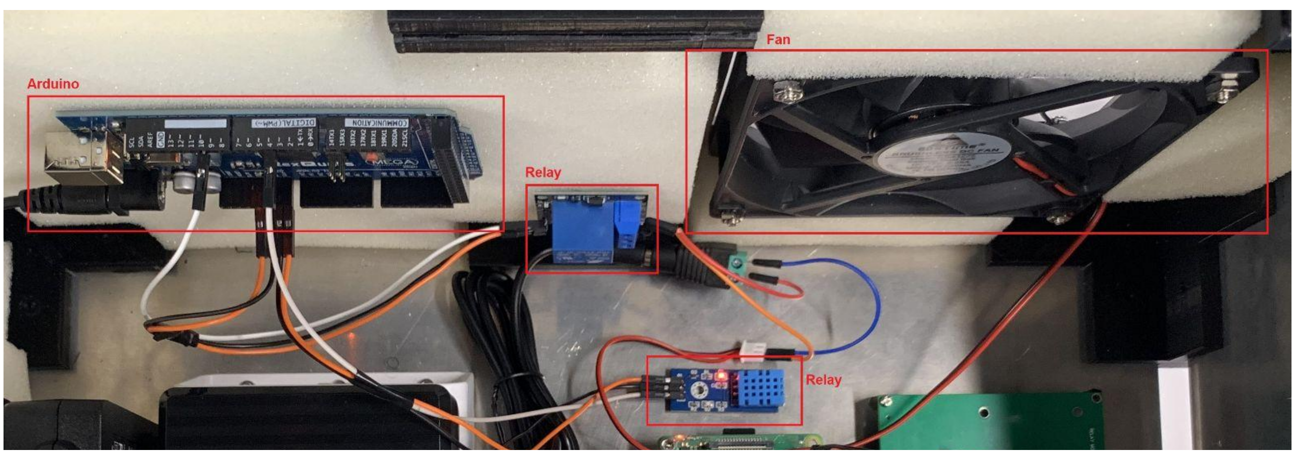


Figure 43. Temperature Control System Components

Publisher Subscriber Connection

The objective of this part of the project was to establish a cloud-based connection between the RaspberryPi4 (RPI4) and Amazon Web Services (which includes the MQTT and DynamoDB) to inform the UI of actions occurring at any of the sensor nodes. The connection path that needs to exist includes the RPI4 going to the MQTT connection within AWS and then a connection from the MQTT to DynamoDB. This path is shown in Figure 44 below.

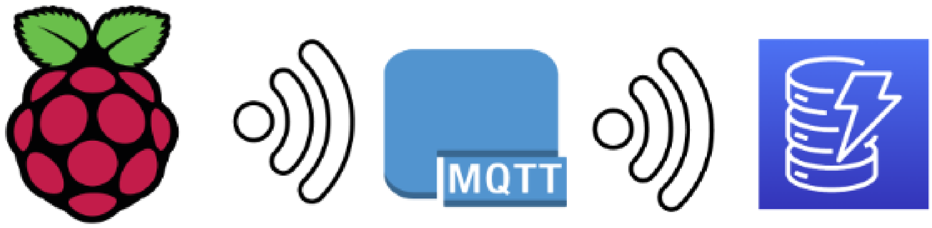


Figure 44

To achieve this end goal, an account was first created for Amazon Web Services (AWS). Once this was done, certificates and keys were downloaded from AWS and uploaded to the RPi4 for authentication purposes. Once this was completed, the API end-point on AWS was taken and added to the code used for the RPi4. Now the RPi4 is connected to AWS and testing could begin.

The next step was to make sure that a stable connection was made and that all messages sent out from lines of code in the RPi4 could be read clearly in the MQTT window. Now that a stable connection had been made between MQTT and the RPi4, a database table, rule, and role needed to be made to format all messages coming in and push it into the database. A table was made in AWS DynamoDB along with the rule to push all MQTT messages into DynamoDB. The rule would make sure that the incoming data from the project would be put into separate columns. The role would make sure that certain pre-existing measures for DynamoDB were accessible for the rule to use when pushing data into the table.

A diagram of the general construction of the physical model integrated with the cloud is shown below in Figure 45.

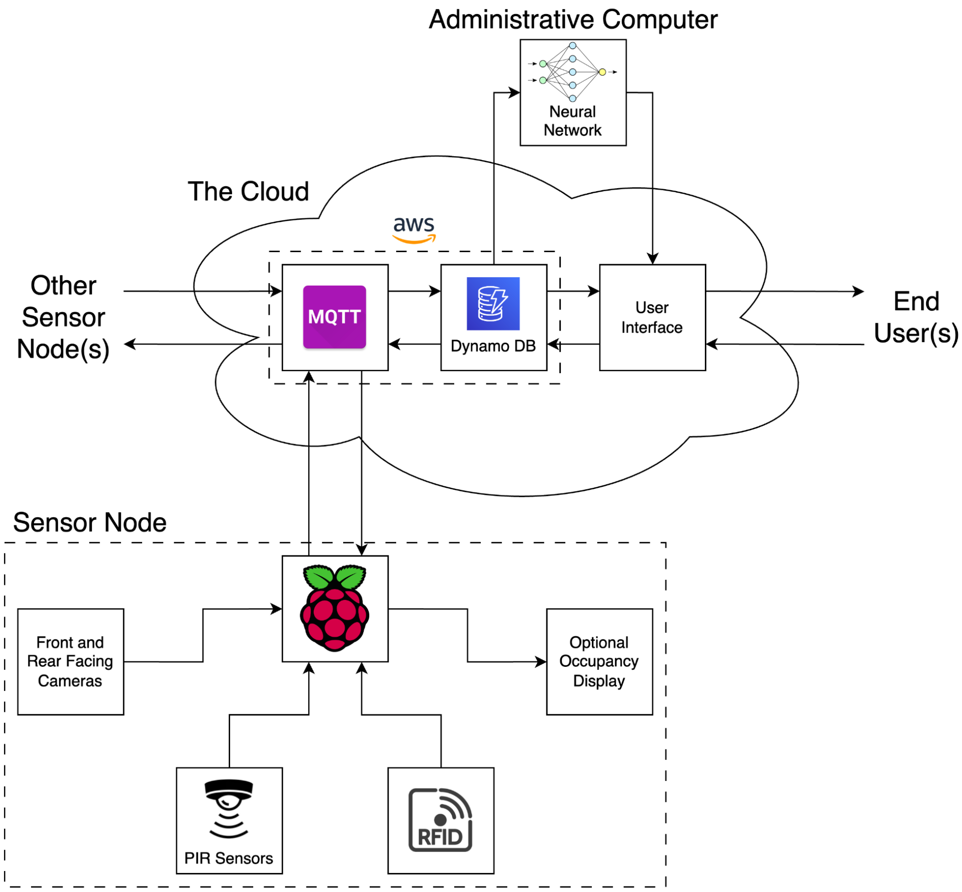


Figure 45

This sub-section of the project had a few areas of challenges. When setting up DynamoDB and making sure it was going to have the correct data entered into it, there was confusion with the primary key and indexes during the creation of the table. To overcome this, when a new table was created, the primary key was made to be the first value in the string of data being sent from the RPI4. Another challenge was that when data was being pushed into the database, there were overlapping values that would lead to incorrect and invalid data being read in the table. To overcome this challenge, an index variable was added to the line of data being pushed into the table to guarantee that each row of data was unique and would allow for multiple entries as intended.

Data Collection System

For a part of this submodule, a camera module and USB cameras needed to be added to the project design to collect images of both the driver and the make and model/license plates of the vehicles that would pass underneath the cameras. The process for the camera module was difficult. A camera module needed to be installed on the Raspberry Pi 4 (RPI4) drive to access any USB cameras plugged into the USB slots on the RPI4. Code was then written to access the cameras and take a picture and download them to the hard drive of the RPI4.

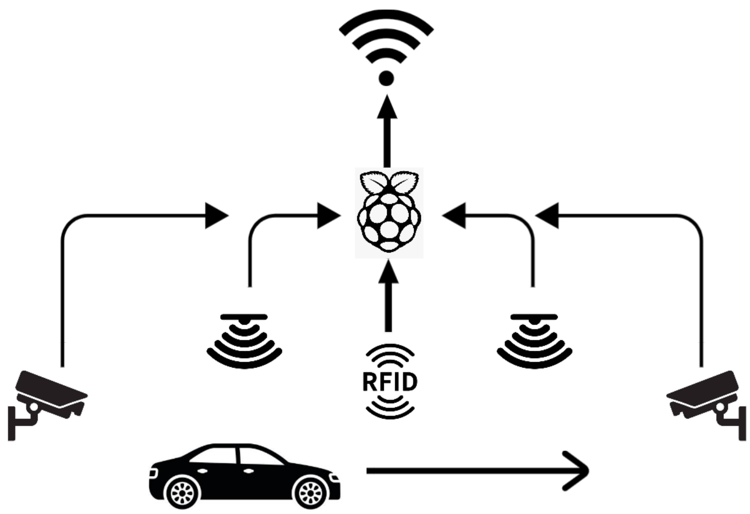


Figure 46. Data Collection Diagram

Figure 46 above describes where each sensor is located. Once one of the PIR sensors are triggered both cameras take pictures of the front and back of the vehicle, while the RFID reader reads for a tag. There are two PIR sensors integrated into the system, one on the front of the device and one in the back. The placement of the PIR sensors are shown below in Figure 47. If the PIR sensor on the front of the device is triggered it represents that a car entered the garage. If the other PIR sensor in the back is triggered, then it represents that a car exited the garage. All of the information gathered from the system is then sent to the server, where the information is inputted into a DynamoDB table. Also the pictures that were taken would be sent to an email address for administrators reference. The heading of these emails sent include the time the pictures were sent and if they entered or exited the garage.

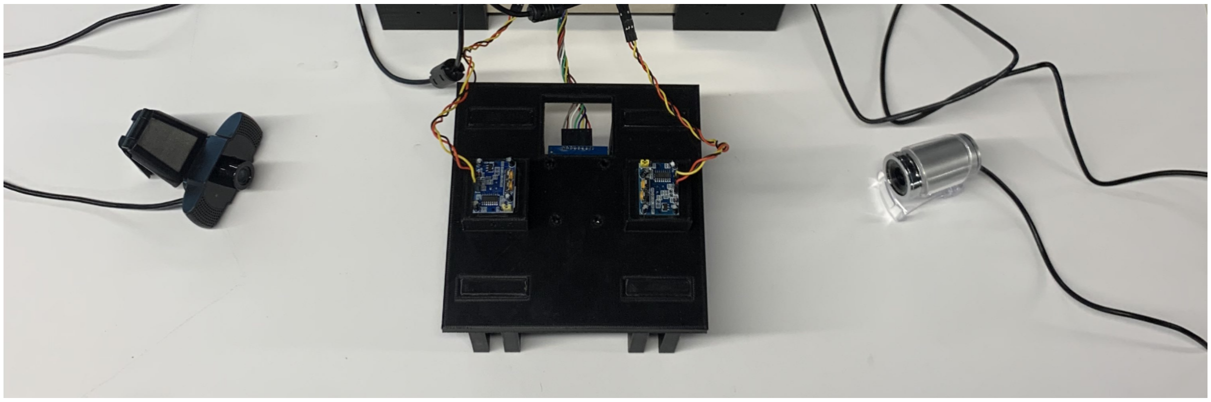


Figure 47. Data Collection

When a car is detected, the RFID becomes active and attempts to read from a pass. The pass will include information such as the name of the driver, the pass id number, the classification of the driver(student, faculty,…), and information regarding their vehicle. The vehicle information includes the make, the model, and color of the car. When there is no pass detected when a car is detected by the system, the values that would be read from the pass, would automatically change to default values. For example, the pass id number will be 0. This will make it easier for the administrators to detect who are not authorized to be in the garage.

When the pictures are taken from the cameras, the pictures are stored locally and then sent to the administrator email for future reference. Each time a picture is taken the previous picture is overwritten locally on the Raspberry Pi. This is to ensure that the Raspberry Pi does not lose memory. Admins will be able to cross reference with the table that includes all of the cars in the garage if needed.

For another part of this submodule, an email script needed to be made to overcome a challenge. For this project, each image captured needed to be accessible to the owner/security personnel. Trying to push an image with MQTT was not an option for the project as there were size limitations put in place by AWS that withheld the project from using MQTT to chop up an image in encoded Base64 code and decoding it after it had passed through the cloud. The most obvious solution to this issue was to send each image taken over email. To accomplish this, a Gmail account was created with third-party accessibility. Code was then created to send an email to the garage/security personnel email with a time stamp and both images of the front of the car and the back of the car. A challenge for this part of the project was that there was an issue with the gmailer module that was installed on the RPi4. It failed to format the pictures properly and could not send anything over to the destination email. The solution to this was to find an older type of python script email and format the code to include an image. The image processing worked in the script and would successfully grant the project access to images taken by the RPI4.

Predictive Model

The system used a sequential multi-class neural network to make predictions on the occupancy of the parking garage based on the inputs of day and time. In the system, the model is periodically retrained in order to provide more accurate predictions for the garage. On the Decepticars' website, users are able to see the predicted occupancies of the garage between 8am and 5pm on the day that is selected by the user. An example is shown below in Figure 48.

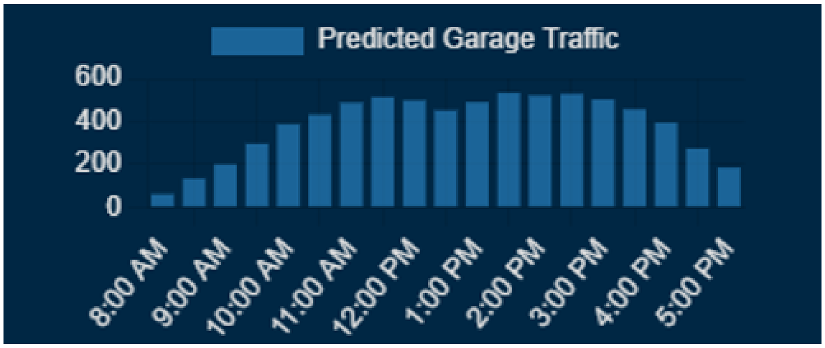


Figure 48. Predicted Model on Website

A neural network is a deep learning technique that mimics the human brain. The neural network consists of several layers, each layer consisting of several neurons. Each neuron consists of two values known as weights, and thresholds. The output of the neuron is compared to the threshold value. If the output of the neuron is greater than the threshold value, the neuron is now active and sends the data to the next layer of neurons.

To train the model, a dataset of an existing parking garage was used. From that dataset, the occupancy of the garage at certain times, as well as the day and time that corresponded to each recorded occupancy was used to build the model. The model was created using Keras, a Python library used for deep learning models built on top of another library known as Tensorflow. The resulting model consisted of eight layers and 8,641 neurons. After the model was trained, the accuracy of the predictions was over 80%. In future applications, adding more layers and neurons to the model can increase the accuracy of the predictions.

Having these predictions available benefits the drivers and parking management. Drivers will be able to see the predictions online on the Decepticars' website. Drivers will be able to make parking decisions in advance based on when they expect to park in the garage. Also parking management can use the predicted values to decide how many parking enforcement workers are needed on certain days and on certain parts of the day.

User Interface

Interface Design

Providing real-time, relevant information to the users is a critical aspect of the system. To achieve this, the team identified the requirements of the system and user interaction. Figure 49. illustrates some basic designs that were created using Figma. These designs served as the starting point for a series of iterative refinements based on feedback from users. By incorporating feedback, the team was able to optimize the design of the user interface, resulting in an intuitive and user-friendly way to provide real-time data.

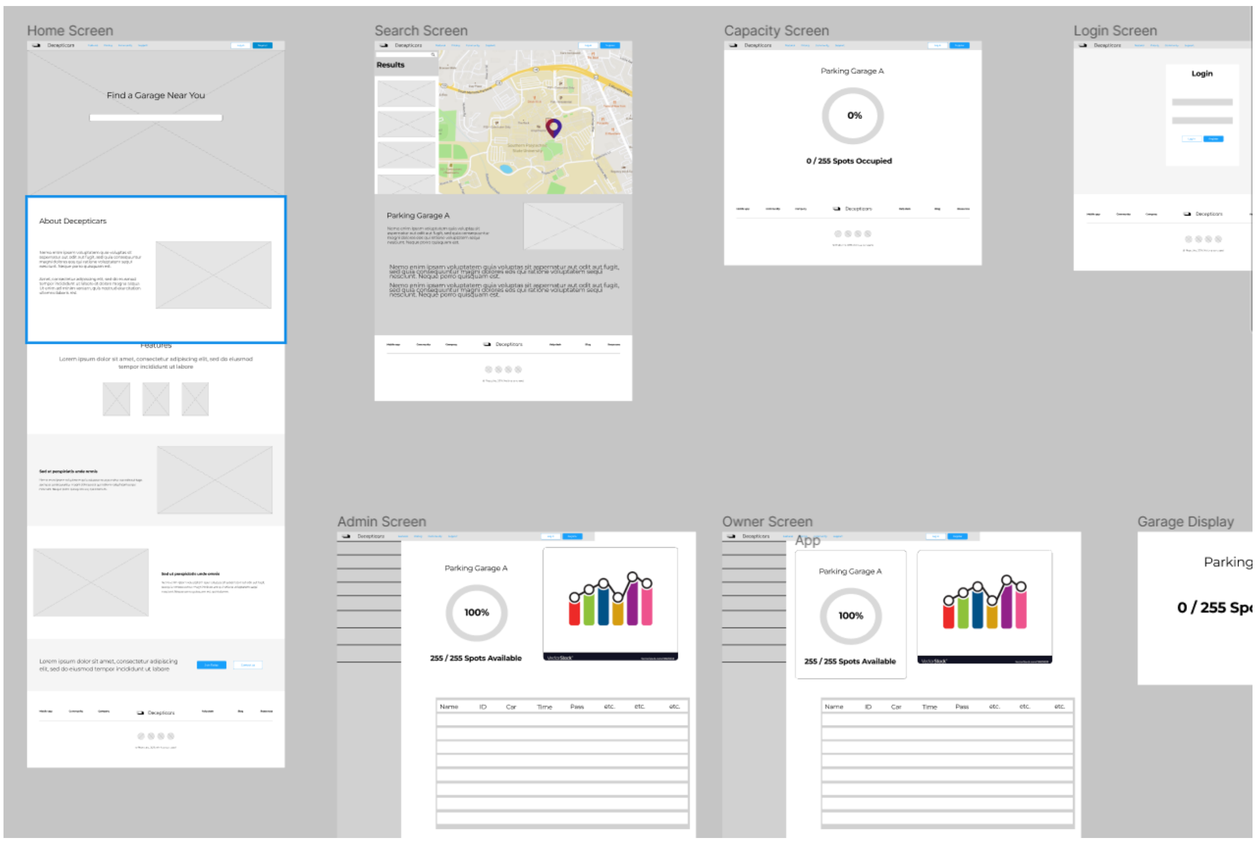


Figure 49. Mockups of Website

Platform Implementation

To ensure the effective display of available parking spaces across multiple platforms, a website was chosen as the user interface. The selection of a website was motivated by the need for cross-platform compatibility. By utilizing a web-based interface, the system was able to ensure that users could access real-time data from any device with internet access, regardless of its operating system or hardware. This approach increased flexibility of the user interface, making it easier to maintain and manage while also reducing the need for platform-specific development.

Figure 50. Below shows the final design that was implemented into the website.

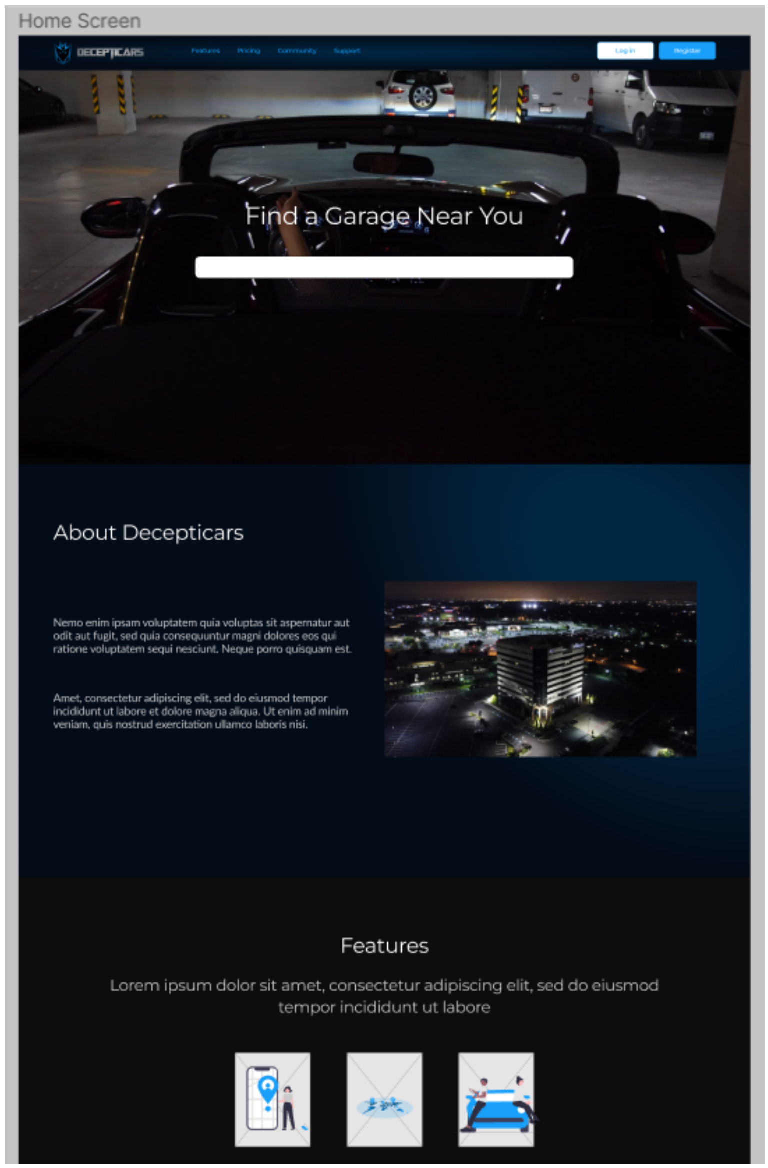


Figure 50. Final Design of Website

Communication Infrastructure

For managing the system's back-end logic and architecture, AWS Amplify was chosen as the ideal solution, providing a flexible API and data-handling infrastructure. This cloud-based service offered a scalable and easily maintainable solution, which included authentication, API management, and database management, all accessible through the AWS console. By utilizing AWS Amplify, the system was able to benefit from a robust and reliable infrastructure that could accommodate the system's growing demands over time.

Figure 51 below depicts the communication flow between the website and AWS, showcasing the interaction between the two systems.

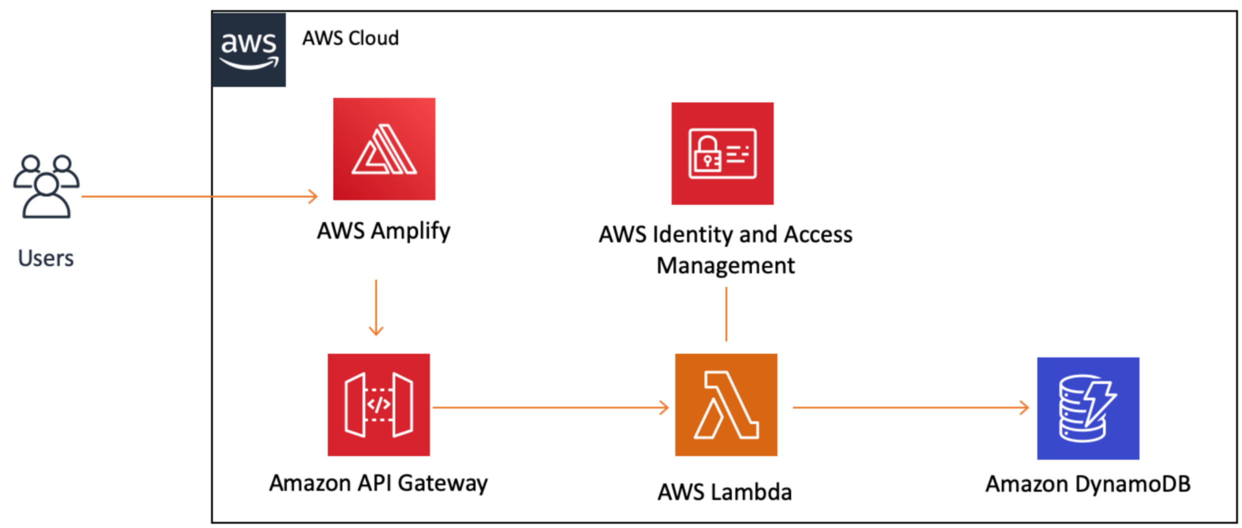


Figure 51. Flow of API Diagram

Market Analysis

The implementation cost of a smart parking system can vary significantly, ranging from a base cost of $50,000 to potentially much higher, depending on the size and complexity of the garage. Factors that can impact the implementation cost include the number and type of sensors required, the level of automation needed, and any customization or integration with existing systems. For example, custom software to operate the smart parking garage alone can cost $125,000 in development. Given the limitations and exuberant costs in the current market, there is a significant need for an innovative solution.

Target Market

Since our motivation for the creation of this product came from frustration with campus parking, and on-campus parking is an issue across many schools, we want to focus (but not limit ourselves) on that market. The number of commuter students make up a large portion of students with above 80%[7] living off-campus and driving to school. There are over 4000[8] colleges across the United States, each with numerous parking decks and lots.

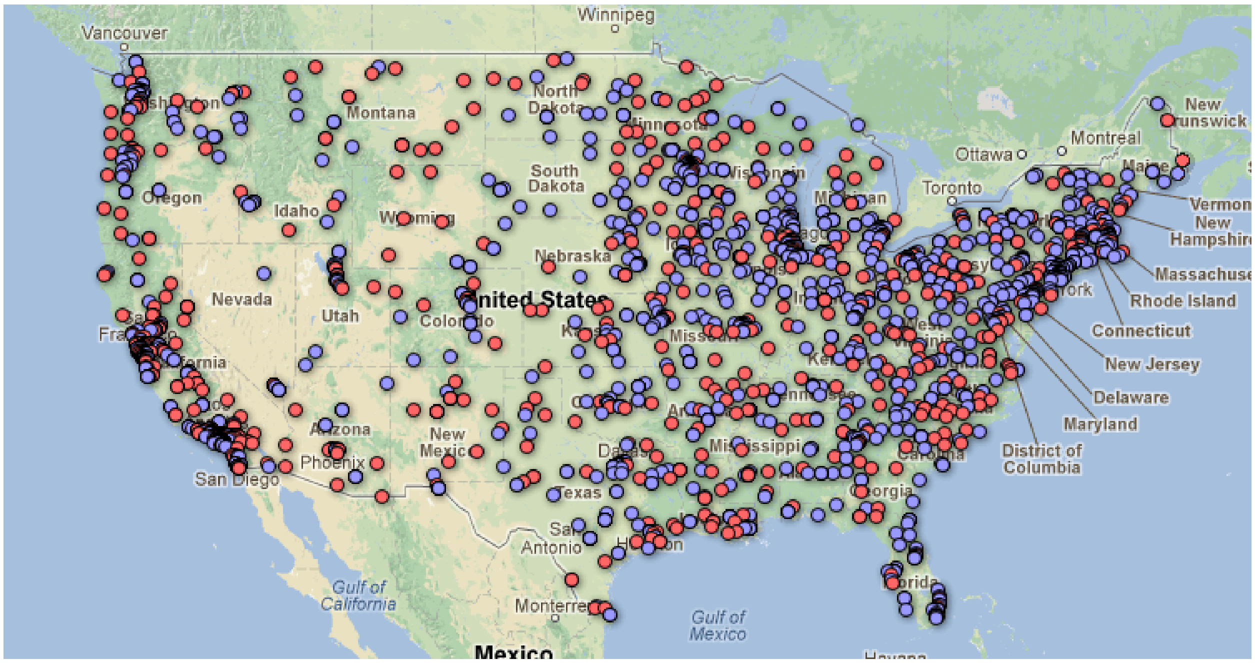


Figure 52. Map of Major Universities in United States

This market has yet to find a definitive solution for parking enforcement nor collect proper data on how many parking passes to sell. Therefore, there is ample need and demand for a low-cost solution that can simultaneously increase happiness in both parking enforcement as well as for the commuters by providing a more effective and streamlined solution for parking in the market.

---

[7] Source: https://education.stateuniversity.com/pages/1875/Commuter-Students.html
[8] Source: https://research.com/universities-colleges/college-statistics#:~:text=Of%20the%204%2C360%20higher%20education,community%20colleges%20in%20the%20U.S

Market Analysis

In 2022, the size of the Parking Lot & Garage market was estimated to be around $8[9] billion. The industry has fallen from its high of around $11 billion in 2017, but a large portion of that decline was a result of the COVID pandemic in 2020, so it might be realistic to expect some recovery in the coming years. This product is specifically geared towards parking garages, but could also be applied to parking lots with limited entry and exit points, so while this market estimation might be higher than applicable, it is still relevant.

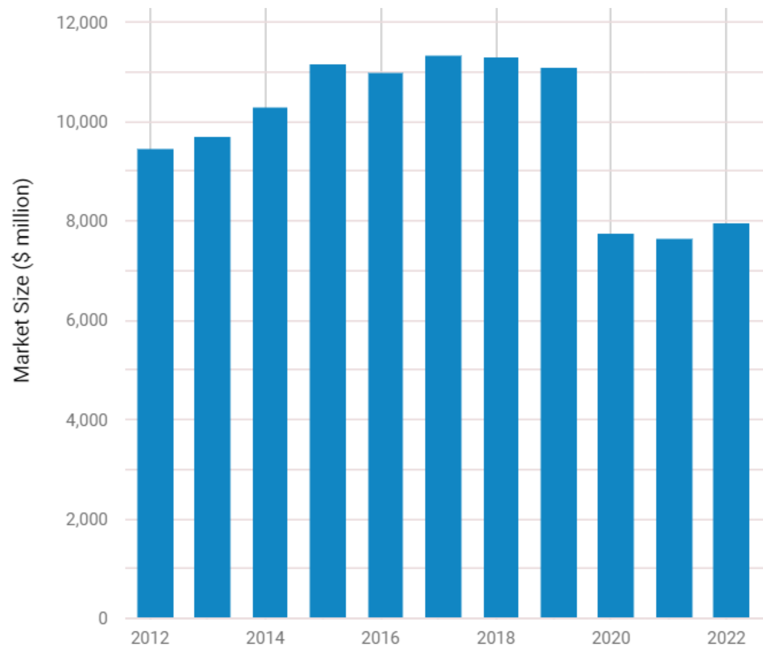


Figure 53. Parking Lot & Garage Market Size

Even at the $8 billion market size the industry is at today, just a 0.5% stake in the market could mean as much as $40 million. This demonstrates that a low-cost and easy to install solution has the potential to capture a significant market share and establish itself as the new industry standard.

[9] Source: https://www.ibisworld.com/industry-statistics/market-size/parking-lots-garages-united-states/

Prototype Development Pricing

The team developed a prototype model to demonstrate the feasibility of the proposed design. As a proof of concept, the prototype does not require the use of high-priced sensors. Instead, a lower-cost RFID reader with a shorter detection range will be used. However, due to the smaller scale of the project, the cost of other components may be higher, as bulk purchasing discounts are not available. A detailed breakdown of the prototype development costs can be found in the table below.

Table 6 Prototype Development Pricing Breakdown

Name	Price per Unit	Quantity	Total
Filament	$17.09	2	$34.18
Surge Protector w/ Backup Battery	$15.97	1	$15.97
Raspberry Pi	$125.00	1	$125.00
Outside Display (TV)	$80.00	1	$80.00
Fan	$7.99	1	$7.99
Ultrasonic Sensors	$4.50	2	$9.00
Camera	$4.70	2	$9.40
RFID Reader	$7.99	1	$7.99
			$289.53

Production Pricing

Upon completion of the prototyping phase, it is estimated that the cost of materials for production of the model may be approximately $650, taking into account bulk pricing and the use of higher quality sensors such as an upgraded RFID reader with a longer detection range. This estimate, however, does not include labor costs or shipping expenses. A detailed breakdown of the expected production costs is provided in the table below.

Table 7 Production Pricing Breakdown

Name	Price per Unit	Quantity	Total
Filament	$17.09	2	$34.18
Surge Protector w/ Backup Battery	$15.97	1	$15.97
Raspberry Pi	$34.00	1	$34.00
Outside Display (TV)	$80.00	1	$80.00
Fan	$7.99	1	$7.99
Ultrasonic Sensors	$4.50	2	$9.00
Camera	$4.70	2	$9.40
UHF RFID Reader	$249.00	1	$249.00
PCB	$35.00	1	$35.00
Aluminum Sheet	$137.69	1	$137.69
Main Housing Supplies	$38.23	1	$38.23
			$650.46

Future Applications

If this project were to be taken to market, there would be a few areas to develop in future iterations to provide more value to the product and more information to the users.

Computer Vision

Using computer vision to detect license plate numbers would bring significant benefits to the Decepticars smart parking solution. While the current system uses RFID technology to identify vehicles, computer vision could add another layer of validation through the detection of license plate numbers.

Another potential application of computer vision in the Decepticars smart parking solution is the ability to detect not only license plate numbers but also the make, model, and color of each vehicle. This would provide additional data to the system, allowing for more accurate tracking of vehicles within the parking structure. By collecting this information, the system could provide more detailed analytics to parking operators, allowing them to make better decisions about parking lot usage and design. Additionally, by tracking the color of each vehicle, operators could detect any unusual or unauthorized vehicles within the parking structure, improving security and safety for all users.

Overall, integrating computer vision into the Decepticars smart parking solution could unlock a variety of potential applications and benefits, making the system more robust, efficient, and appealing to the commercial market.

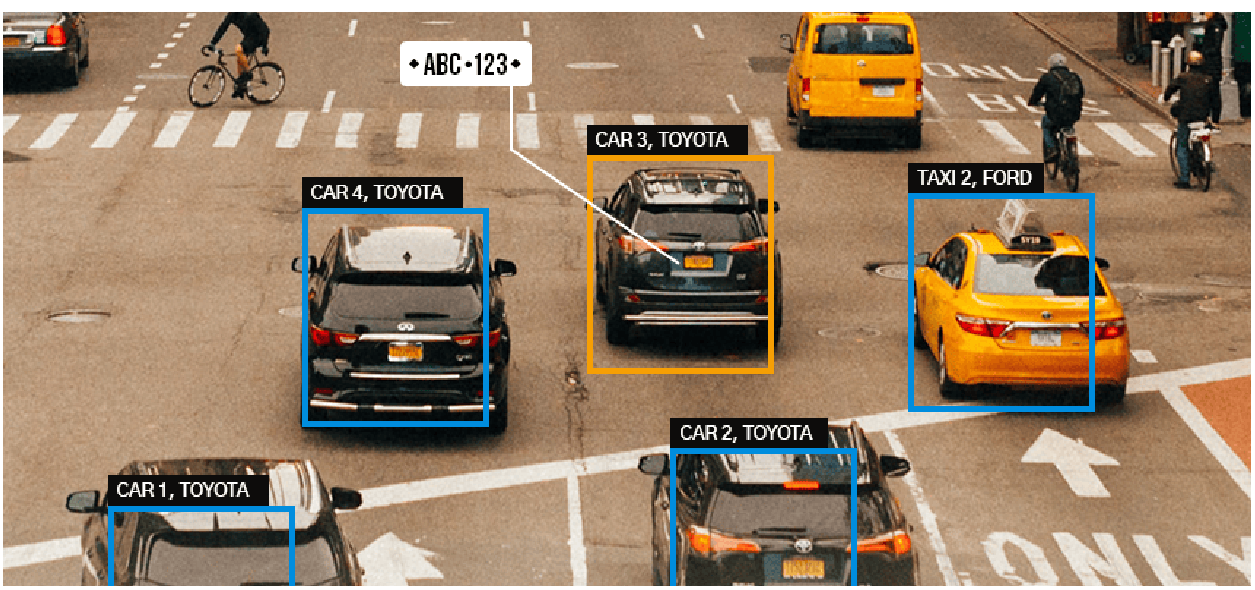


Figure 54. Computer Vision Application with Vehicle Identification[10]

[10] Image Attribution: https://tinyurl.com/2xmjpx5k

Navigation Systems

Another potential application for a future development idea for the Decepticars smart parking solution is the integration of parking occupancy data into navigation systems. By providing real-time parking occupancy information, drivers could be more informed about the availability of parking spots and the estimated time it would take to park their vehicle.

This integration would not only provide a more convenient experience for drivers but could also reduce traffic congestion and greenhouse gas emissions. By providing drivers with accurate parking occupancy data, they would be able to make more informed decisions about their routes, reducing the likelihood of circling around looking for parking spots and ultimately reducing traffic congestion.

Additionally, the system could be integrated with ride-sharing applications to optimize pick-up and drop-off locations based on parking availability. Drivers for ride-sharing services could be directed to areas with a high concentration of available parking spots, reducing the amount of time spent searching for parking and ultimately increasing the efficiency of the ride-sharing service.

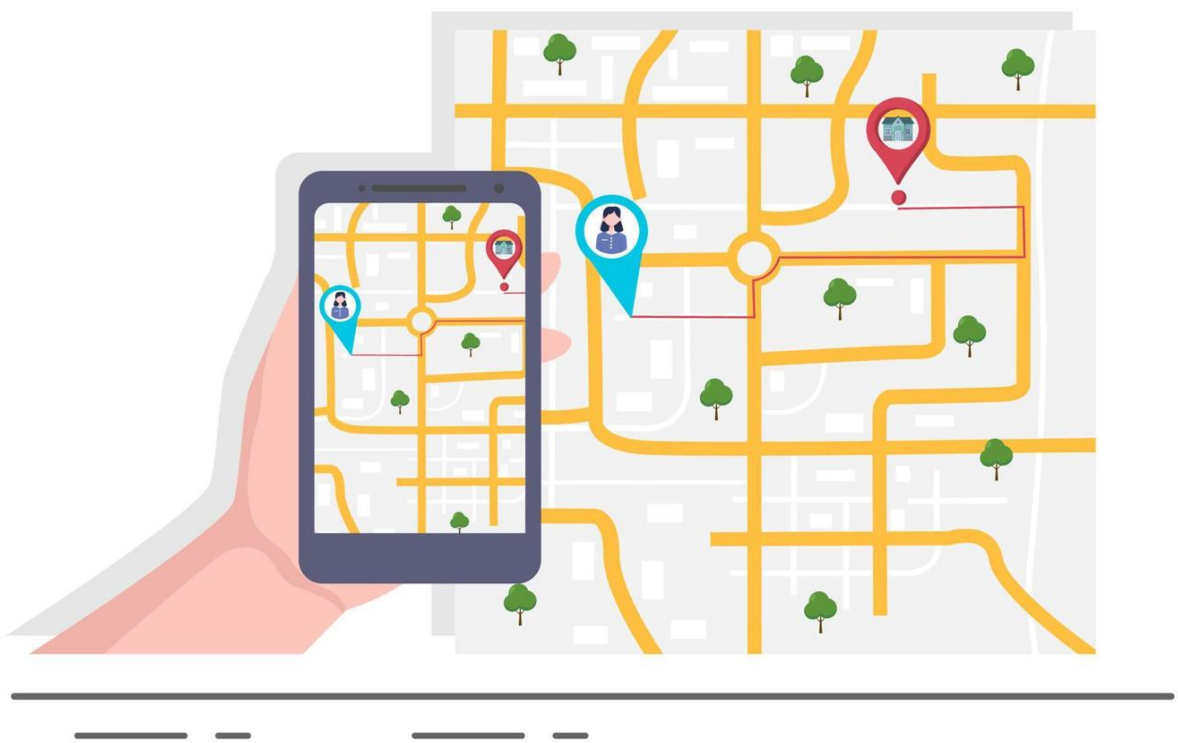


Figure 55. Routing Application[11]

[11] Image Attribution: https://tinyurl.com/5n72jpwy

Billing Systems

Integrating billing systems into the Decepticars smart parking solution could make it more appealing to the commercial market. With the ability to track and identify every vehicle entering and exiting the parking structure, billing systems could be implemented to automatically calculate and charge for parking based on various factors such as the length of stay, time of day, and even location within the structure. This could eliminate the need for traditional ticketing systems or manual payment processing, providing a seamless and convenient experience for both the parking structure and its users.

Furthermore, data collected from the billing systems could be used to further improve the overall efficiency of the parking structure. For example, it could identify peak usage times and pricing models, allowing the parking structure to adjust pricing to maximize profits while still providing fair and competitive pricing to its users. This information could also be used to optimize staffing levels, ensuring that parking enforcement and customer service personnel are available when and where they are needed most.

Integrating billing systems into the Decepticars smart parking solution would be a significant step towards making the system more comprehensive and appealing to commercial markets such as shopping centers, airports, and other high-traffic areas. By providing a one-stop-shop for parking management, from occupancy tracking to payment processing, the Decepticars system would stand out as a comprehensive and user-friendly solution

Appendix

DHT11 Sensor Datasheet

DHT11 Specifications

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±1°C and ±1%

Parameters	Condions	Minimum	Typical	Maximum
Humidity				
Resoluon		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resoluon		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

MegaPixels USB Camera Datasheet

SPECIFICATION

- Photosensitive Element: CMOS
- AGC/AEC/AWB: auto
- Infrared Filter: 650±10nm
- Supports MJPG and YUV format
- FPN: < 0.03% of VPEAK-TO-PEAK
- Connector: USB 2.0
- Pixel: 30W
- Resolution: 640*480
- Operating Voltage: 5V
- Operating Temperature: -20℃-70℃
- FCC & CE Certification
- Dimension: 30*25*21.4mm/1.18*0.98*0.84"
- Supports Windows, iOS, Android, Linux

Table 5.2-1 Format of Setup Data for the Cancel Request

Offset	Field	Size	Value	Description
0	bmRequestType	1	bitmap	00100001 Host-to-Device, Class-Specific, Recipient-Interface
1	bRequest	1	code	Cancel_Request (0x64, for this request)
2	wValue	2	value	value equal to zero.
4	wIndex	2	value	value equal to Interface Number.
6	wLength	2	count	Value = 0x0006

Table 5.2-3 Format of Setup Data to retrieve the Extended Event Data

Offset	Field	Size	Value	Description
0	bmRequestType	1	bitmap	10100001 Device-to-Host, Class-Specific, Recipient-Interface
1	bRequest	1	code	Get_Extended_Event_Data (0x65, code for this request)
2	wValue	2	value	value equal to zero.
4	wIndex	2	value	value equal to Interface Number.
6	wLength	2	count	Size of the host buffer allocated for the Extended Event Data

Table 5.2-5 Format of Setup Data for the Device Reset Request

Offset	Field	Size	Value	Description
0	bmRequestType	1	bitmap	00100001 Host-to-Device, Class-Specific, Recipient-Interface
1	bRequest	1	code	Device_Reset_Request (0x66, for this request)
2	wValue	2	value	value equal to zero.
4	wIndex	2	value	value equal to Interface Number.
6	wLength	2	count	Value of 0x0000, there is no data associated with this request

PIR Sensor Datasheet

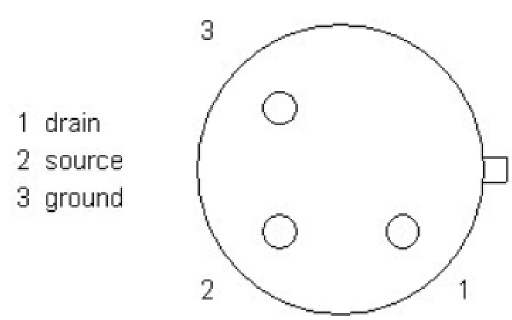
Pin Definitions and Ratings

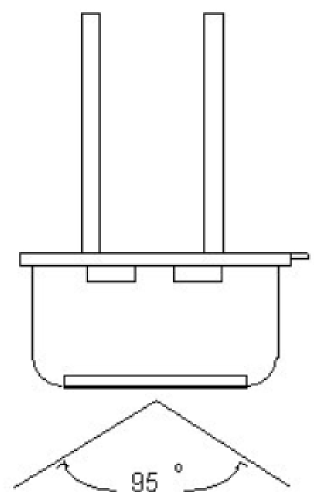
Pin	Name	Function
-	GND	Connects to Ground or Vss
+	V+	Connects to Vdd (3.3V to 5V) @ ~100uA
OUT	Output	Connects to an I/O pin set to INPUT mode (or transistor/MOSFET)

Jumper Setting

Position	Mode	Description
H	Retrigger	Output remains HIGH when sensor is retriggered repeatedly. Output is LOW when idle (not triggered).
L	Normal	Output goes HIGH then LOW when triggered. Continuous motion results in repeated HIGH/LOW pulses. Output is LOW when idle.

BOTTOM VIEW





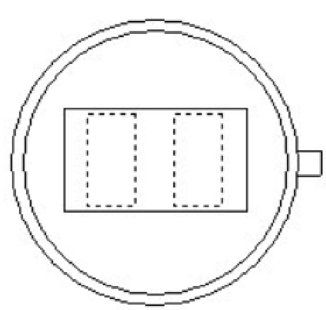
View Angle

PIR325

SENSITIVE AREA 2 ELEMENTS
SPECTRAL RESPONSE 5 - 14 um
OUTPUT VOLTAGE mv pp 20
NOISE uVpp 20
OFFSET VOLTAGE volts 1.0
SUPPLY VOLTAGE volts 2.5 - 15
OPERATING TEMP c 30 - 70

Test Conditions for output voltage:
Supply voltage = 5 volts
100K load resistor from pin 2 to 3
IR source = Hand moving 6" from sensor

TOP VIEW



RFID Sensor Datasheet

7.1 Pin description

Table 3. Pin description

Pin	Symbol	Type[1]	Description
1	I2C	I	I^2C-bus enable input[2]
2	PVDD	P	pin power supply
3	DVDD	P	digital power supply
4	DVSS	G	digital ground[3]
5	PVSS	G	pin power supply ground
6	NRSTPD	I	reset and power-down input: power-down: enabled when LOW; internal current sinks are switched off, the oscillator is inhibited and the input pins are disconnected from the outside world reset: enabled by a positive edge
7	MFIN	I	MIFARE signal input
8	MFOUT	O	MIFARE signal output
9	SVDD	P	MFIN and MFOUT pin power supply
10	TVSS	G	transmitter output stage 1 ground
11	TX1	O	transmitter 1 modulated 13.56 MHz energy carrier output
12	TVDD	P	transmitter power supply: supplies the output stage of transmitters 1 and 2
13	TX2	O	transmitter 2 modulated 13.56 MHz energy carrier output
14	TVSS	G	transmitter output stage 2 ground
15	AVDD	P	analog power supply

NXP Semiconductors **MFRC522**

Standard performance MIFARE and NTAG frontend

Table 1. Quick reference data *...continued*

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
I_{pd}	power-down current	$V_{DDA} = V_{DDD} = V_{DD(TVDD)} = V_{DD(PVDD)} = 3$ V					
		hard power-down; pin NRSTPD set LOW	[4]	-	-	5	μA
		soft power-down; RF level detector on	[4]	-	-	10	μA
I_{DDD}	digital supply current	pin DVDD; $V_{DDD} = 3$ V		-	6.5	9	mA
I_{DDA}	analog supply current	pin AVDD; $V_{DDA} = 3$ V, CommandReg register's RcvOff bit = 0		-	7	10	mA
		pin AVDD; receiver switched off; $V_{DDA} = 3$ V, CommandReg register's RcvOff bit = 1		-	3	5	mA
$I_{DD(PVDD)}$	PVDD supply current	pin PVDD	[5]	-	-	40	mA
$I_{DD(TVDD)}$	TVDD supply current	pin TVDD; continuous wave	[6][7][8]	-	60	100	mA
T_{amb}	ambient temperature	HVQFN32		−25	-	+85	°C

[1] Supply voltages below 3 V reduce the performance in, for example, the achievable operating distance.

[2] V_{DDA}, V_{DDD} and $V_{DD(TVDD)}$ must always be the same voltage.

[3] $V_{DD(PVDD)}$ must always be the same or lower voltage than V_{DDD}.

[4] I_{pd} is the total current for all supplies.

[5] $I_{DD(PVDD)}$ depends on the overall load at the digital pins.

[6] $I_{DD(TVDD)}$ depends on $V_{DD(TVDD)}$ and the external circuit connected to pins TX1 and TX2.

[7] During typical circuit operation, the overall current is below 100 mA.

[8] Typical value using a complementary driver configuration and an antenna matched to 40 Ω between pins TX1 and TX2 at 13.56 MHz.

DECEPTICARS
a better smart parking solution

T81 Relay Datasheet

Approvals

UL E29244, CSA LR48471

Technical data of approved types on request

Contact Data

Contact arrangement	1 form C (CO)
Rated voltage	24VDC, 120VAC
Max. switching voltage	60VDC, 125VAC
Rated current	1A
Limiting continuous current	2A
Switching power	120VA, 30W
Contact material	Au overlay AgPd alloy
Contact style	single contact
Min. recommended contact load	1mA at 1VDC
Initial contact resistance	50mΩ at 100mA, 6VDC
Frequency of operation	72000 ops/h
Operate/release time max.	
standard coil	5/7ms
sensitive coil	10/7ms
Electrical endurance	
1A, 120VAC, resistive	$100x10^3$ ops.
1A, 24VDC, resistive	$100x10^3$ ops.
Contact ratings	1A, 120VAC/24VDC, resistive 2A, 125VAC/30VDC, resistive (NO)
Mechanical endurance	$10x10^6$ ops.

Coil Data

Magnetic system	neutral
Coil voltage range	5 to 24VDC
Max. coil temperature	105°C

Coil Data (continued)

Coil versions, DC coil

Coil code	Rated voltage VDC	Operate voltage VDC	Release voltage VDC	Coil resistance Ω±10%	Rated coil power mW
Standard coil, 450mW					
05	5	3.5	0.25	55	450
06	6	4.2	0.3	80	450
09	9	6.3	0.45	180	450
12	12	8.4	0.6	320	450
24	24	16.8	1.2	1280	450
Sensitive coil, 200mW					
05	5	3.75	0.5	125	200
06	6	4.5	0.6	180	200
09	9	6.75	0.9	400	200
12	12	9.0	1.2	700	200
24	24	18.0	2.4	2800	200

All figures are given for coil without preenergization, at ambient temperature +23°C.

Insulation Data

Initial dielectric strength	
between open contacts	$500V_{rms}$
between contact and coil	$1000V_{rms}$
Initial surge withstand voltage	1500Vrms (10/160µs)
Initial insulation resistance	
between insulated elements	$10^9Ω$
Clearance/creepage	
between contact and coil	1.5/1.76mm