Automated Parking Assistant

By
Daniel Ahn (dka3)
Mehul Dugar (mdugar2)
Freddy Zhang (fz8)

Final Report for ECE 445, Senior Design, Fall 2021
TA: Dean Biskup

08 December 2021
Project No. 08

Abstract

The Parking Assistant is a small, modular product designed to alleviate many of the annoyances associated with parking vehicles in larger parking garages. Consisting of an RF communication protocol, an IR sensor, a camera, and the computing power to run, the Parking Assistant is designed to seamlessly sense when a car arrives in a parking space, take a picture of its license plate, and log how long the car was stationary for. This information is communicated to a Central Logger at the payment station at the exit. This report will delve into the design of the product, how that design was verified, and the cost/effort put into implementing that design. Future improvements that can be made will also be outlined in the conclusion of this report.

1. Introduction

1.1 Problem & Solution Overview

It is difficult to argue against the idea that traffic congestion is one of the more annoying aspects of driving. Congestion causes more fuel burn, more delays, and can even cause health problems due to higher concentrations of air pollution, according to a survey by the University of Surrey [1]. The US Department of Transportation's Federal Highway Administration notes 7 causes of congestion: traffic incidents, work zones, weather, fluctuations in normal traffic, special events, traffic control devices, and physical bottlenecks [2]. Of these 7 causes, 3 (fluctuations in normal traffic, special events, and physical bottlenecks) have to do with traffic volume reaching road capacity. These capacity issues, while possible to correct on traditional roads, are extremely difficult to correct in parking lots. In order to remain economically viable, parking lots are required to maintain as many parking spaces as possible, reducing traffic capacity available for entrances, passages, and exits. The need to collect tickets at the entrance further reduces traffic volume capacity as each vehicle needs to stop for a few seconds in an already bottlenecked part of the parking lot. Even worse is that cars, once they collect their ticket, are forced to pay for time they spent roaming around the parking lot in search of a space. All these aspects of parking lots cause customer dissatisfaction.

This parking lot entrance bottleneck issue is what we are solving with this project. Further need for our solution can be found in a report released by the International Parking & Mobility Institute in 2018 underscoring the need for better parking lot entrance systems. Of the professionals in parking, transportation, and mobility surveyed, 46% said that there is demand for technology to improve access control systems. Our problem is that the parking lot customer experience is too poor due to congestion at the entrance and having to pay fees while looking for a parking space.

Our solution is to create a parking lot system that can remotely scan license plates and sense car movement to design a car payment system that is able to communicate with a central logger to handle payment systems without needless effort and materials. Our solution will combat the problem by automating all previously man-made bottlenecks at the entrance of the parking lot and delegating those responsibilities to individual modules located at each parking space. With our solution, we will eliminate the physical ticket, the need for cars to queue at the entrance, and needlessly wasted money spent by drivers roaming the parking lot in search of a spot.

1.2 Visual Aid

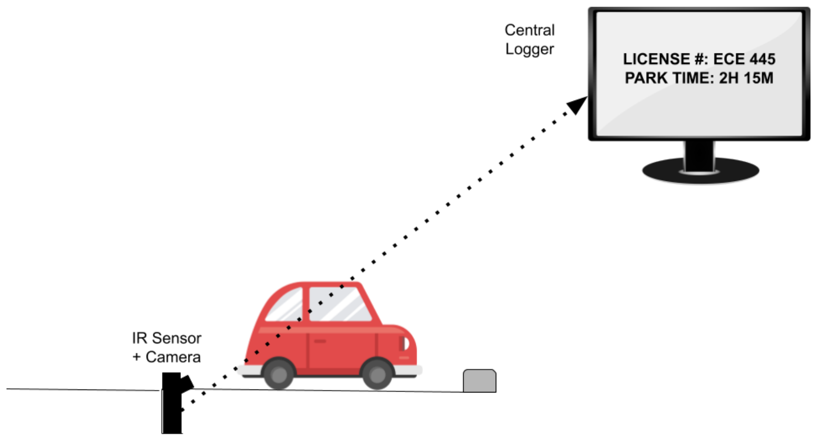


Figure 1. Visual Aid

1.3 High-level requirements list

- The automated parking assistant must be able to read a license plate to 85% (368/433 cars) accuracy when conditions are 'reasonable' (clear conditions, i.e - no obstruction in front of the camera)
- The automated parking assistant must be able to determine how long a car was parked and send the data via RF to the central logger system with a total latency not exceeding 5 minutes.
- The automated parking assistant must be able to withstand harsh weather conditions (-40°C to 40°C), waterproof (IPX4), and shock resistant (functional after a 10 ft. drop)

1.4 Block Diagram and Descriptions

The parking assistant consisted of two parts, the PCB and the Central Logger System. The PCB was divided into five separate systems: the RF System, the Camera System, the IR Sensor System, the Control System, and the Power System.

The PCB facilitated communication between itself and the Central Logger System using a LoRa RF transceiver module. Using a 915 MHz signal, the signal was strong enough to be detected at long distances and through concrete structures found in many parking garages. It transmitted the length of time the car was parked for and the license plate number to the Central Logger System, which will facilitate the payment process. This data was encrypted to prevent unnecessary sharing of potentially personal data.

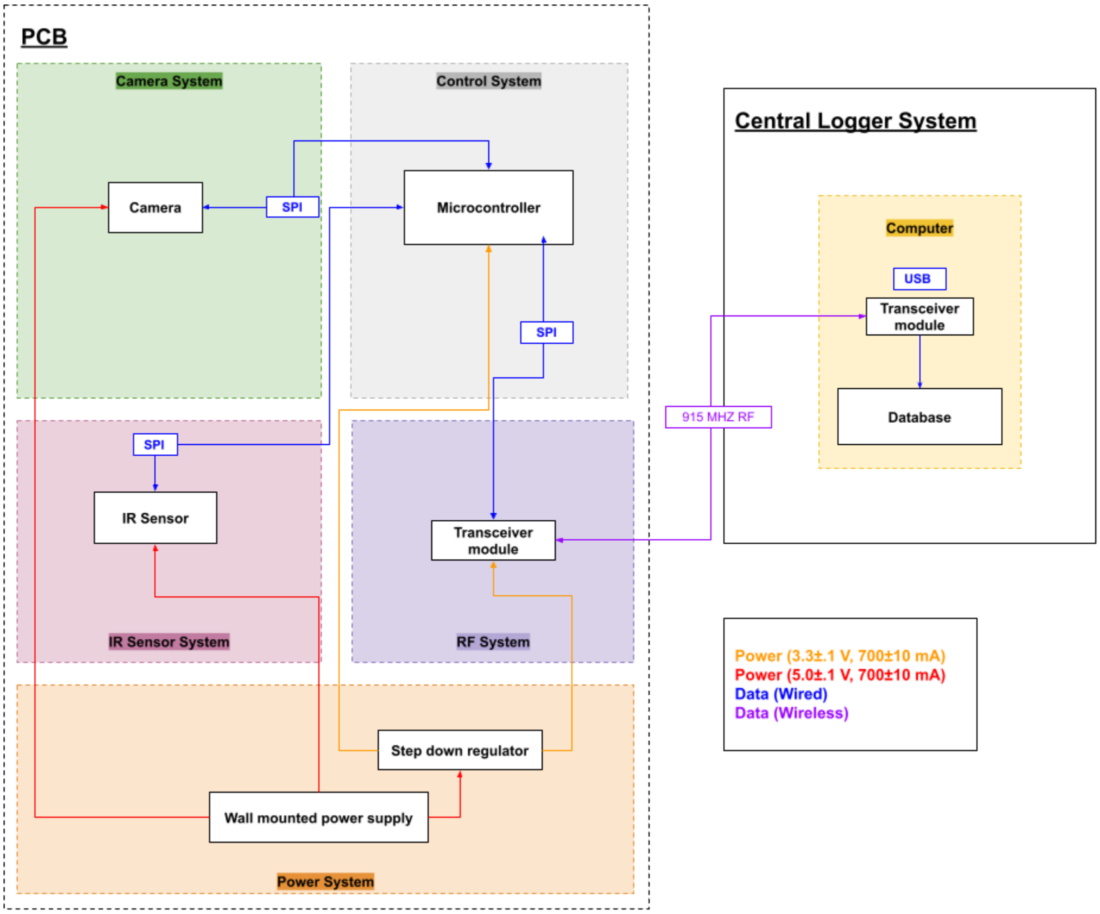


Figure 2. Block Diagram

The data transmitted to the PCB was obtained from the Camera System and the IR Sensor System. The IR Sensor System consisted of an IR sensor that was triggered when it detected that a car was entering or exiting the parking spot it was placed in front of. Using timestamps when this sensor was triggered, the length of time the car was parked for in the parking space could be calculated. The Camera System took an image of each car's license plate. Once installed, the camera could be focused to guarantee that the image was taken under as favorable conditions as possible. For our purposes, we were assuming favorable lighting conditions without extreme weather occurrences.

Using this image, the Control System (consisting of a microcontroller) was responsible for using the characters of the license plate to identify each vehicle separately. The Control System was also responsible for calculating the amount of time each vehicle was parked for using the timestamps from the IR Sensor System.

Powering all this was the Power System. The Power System needed to accept wall power (120V-240V AC current) and output both 5V and 3.3V to power all the other systems on the PCB. Given that each parking assistant module would be implemented permanently, we decided to go with a battery-less setup as portability was far less important than reliability in our case.

The Central Logger System was the only distinct system separate from the PCB. The Central Logger System would be present on a client's computer that could communicate with each of the Parking Assistants via the aforementioned RF. The Central Logger System stored a log of when vehicles parked and left along with their detected license plate numbers such that the billing process can easily be completed.

2. Design

2.1 Design Procedure

The initial procedure for the design of the Parking Assistant involved deciding which parts were necessary for the operation of the entire system. Initial designs for the Parking Assistant did not include entire systems present on the final design. In this section, we will discuss the rationale behind each of the 6 systems on the design.

The usage of a LoRa RF communication protocol was chosen based on our needs for the signal to be able to travel long distances reliably through entire parking lots. Compatibility with SPI was also a plus, as we were planning to use SPI with all of our other systems. Alternatives to LoRa in the form of Wi-Fi and Bluetooth were abandoned due to shorter ranges and in the case of Wi-Fi, the need for a separate router.

Given that one of our core objectives for the Parking Assistant was to accurately read license plates, the presence of a camera was never questioned. The Camera System used an Arducam because of its 'jack of all trades' nature. Widespread compatibility, high resolution, and thorough documentation meant that the Arducam had no competition.

The IR Sensor System's inclusion in the final design was highly dependent on the availability of a longer-range IR sensor. A longer-range IR sensor was placed because we decided that the Camera System should be placed such that any picture taken from the same location would reliably take a coherent image regardless of the position of the car. This required the camera to be placed further away from where a car would be parked to account for possible shifts in position of the car. As all PCB systems are directly connected to each other and placed in a small housing, placing the camera further away also meant placing the IR sensor further away.

For the Control System, we chose the STM32H743BIT6 for its powerful high-performance processor, the ARM Cortex-M7. Since we were planning on running an OCR program on the microcontroller, we needed one that was powerful enough to be able to process the image in a timely manner. We were originally going with the STM32H743VIH6 since they were cheaper, but they were all sold out by the time we were planning on ordering our components for the project.

The Power System used a USB-C input head to supply power to the board. We decided on a USB-C input because they are very easy to come by which made them more accessible which helps with scalability. The Power System also incorporated the use of an LD1117 voltage regulator that could convert the incoming 5V current from the USB port into 3.3V 800mA current since some of our components that we were using require 3.3V to be supplied to the input.

The Central Logger System was built around compatibility with a wide range of computers. Since this product needed to be able to be implemented across a wide range of client computers with as little hassle as possible, we decided to connect an RF module (the same as the modules in the RF System) to the computer via USB and run a simple program that can log when a car entered a parking space and left. The details would be stored in a lightweight .csv file on the client's computer, ensuring widespread compatibility.

2.2 Design Details

In this section, the overall schematic will be parsed. As a reference, Figure 3 shows the schematic in its entirety.

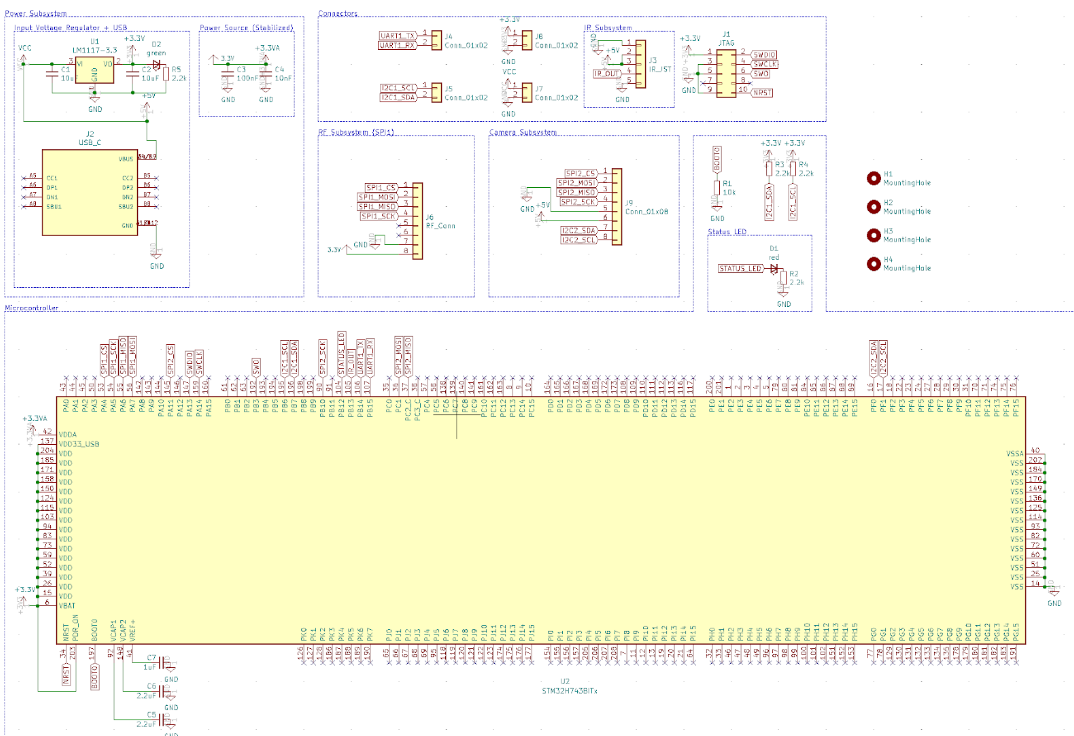


Figure 3. Overall Schematic

2.2.1 Power System

Based on the datasheets, all of the components accepted either 3.3V or 5V [3] so we needed to be able to supply those two voltages in the PCB. Common wall adapters that use USB-C inputs supply 5V so we only needed to be able to supply 3.3V with the 5V. There were capacitors that were installed to stabilize the voltages that can be seen in Figure 4. 10 µF capacitors were used to stabilize the 5V output supplied by the USB-C connection to the 3.3V regulator and the components that required 5V, the IR sensor and camera. 100nF and 10 nF capacitors were used to stabilize the 3.3V output from the regulator to ensure a steady output of 3.3V to the other components like the RF transceiver and the microcontroller.

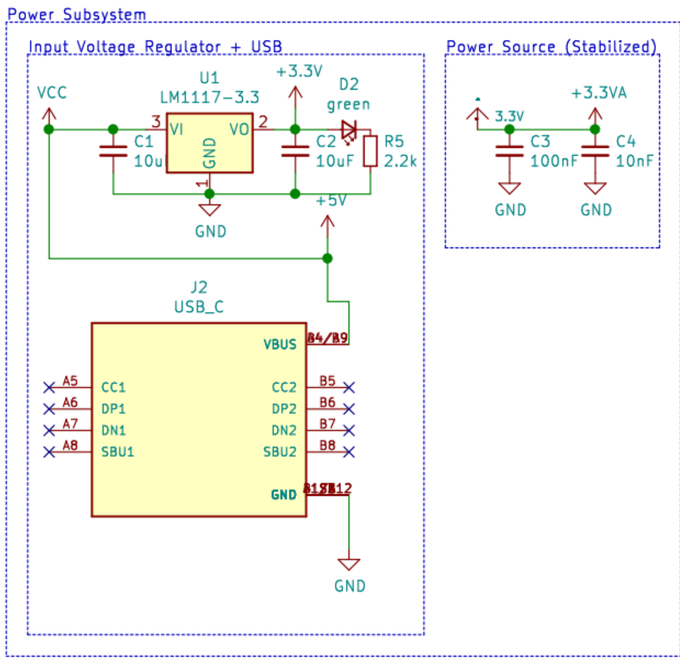


Figure 4. Power System Schematic

2.2.2 IR System

Figure 5 was provided in the datasheet for the IR sensor that displayed the relationship between the output voltage and the inverse of the distance [4]. Since it is an approximately linear increase, the slope can be calculated with $m = (y_2 - y_1) / (x_2 - x_1) = (2.5 - 2)/(0.01 - 0.006) = 125$. With the slope, the equation for the line can be found using the point-slope formula (Eq 2.2.2.1) to get the equation: $y = 125x + 1.25$ where y is the output voltage and x is the inverse of the distance away from the IR sensor.

$$y - y_1 = m(x - x_1)$$
$$y - 2.5 = 125(x - 0.01)$$
$$y = 125x + 1.25$$

(Eq. 2.2.2.1)

In order to calculate the distance away an object is from the IR sensor, the linear equation was manipulated to obtain the distance as a function of the output voltage: $d = 125/(y - 1.25)$ where d is the distance from an object to the IR sensor and y is the output voltage from the IR sensor. With this equation, it was possible to calculate the distance the IR sensor detected simply based on the output voltage that was supplied to the microcontroller.

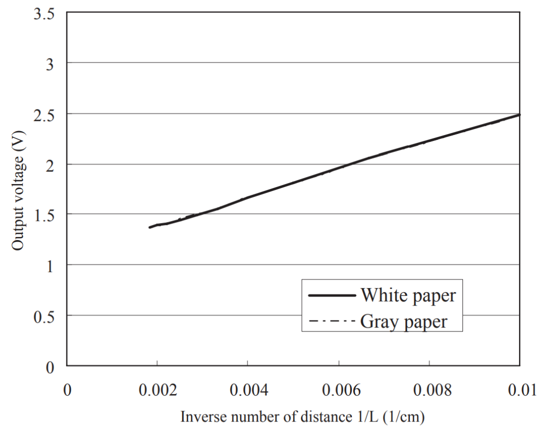


Figure 5. IR Sensor Distance Characteristics

2.2.3 Camera System

An interface was developed by Arducam to test the features of the camera with an Arduino. The GUI can be seen in Figure 6. There are many different fields that can be modified such as the pixel density, resolution, and camera mode. If the resolution and pixel density was set too high, the picture would be corrupted due to the lack of computing power on the Arduino to successfully take a clear picture. Based on some trial and error, we decided that having a pixel rate of 5640 and a resolution of 800x480 was ideal for taking pictures of the license plate since it was clear enough to be read as well as small enough for the Arduino to render successfully with consistency. Rather than having a continuous video feed, we set the mode to 'Single' to take singular images to conserve computing power and storage space.

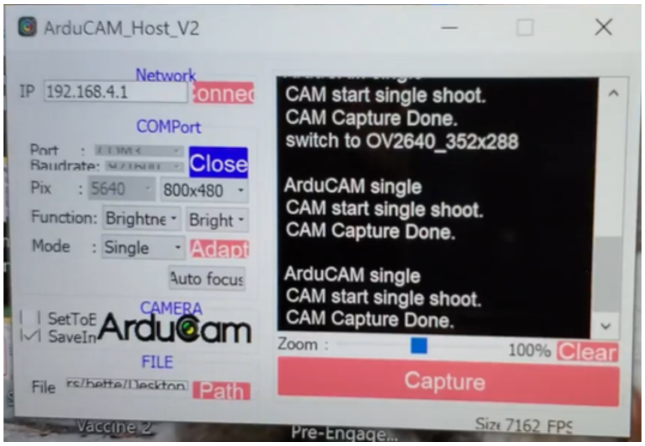


Figure 6. Arducam GUI

2.2.4 Control System

Our choice for the microcontroller for this project was largely guided by the supported MCU's listed under *TensorFlow Lite*'s website that outlines all the supported MCU types [5]. They specifically mention

STM32746 Discovery kit that runs on a Cortex M7 Microcontroller, due to lack of availability for the exact M7 model used, we chose to go with the STM32H743BIT6.

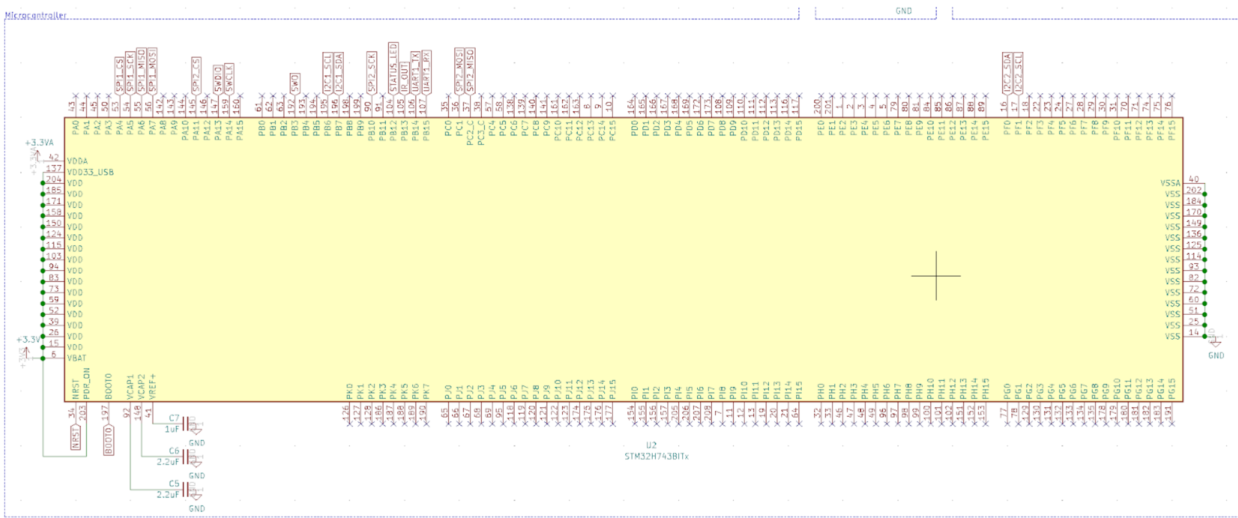


Figure 7. Microcontroller Schematic

2.2.5 RF System

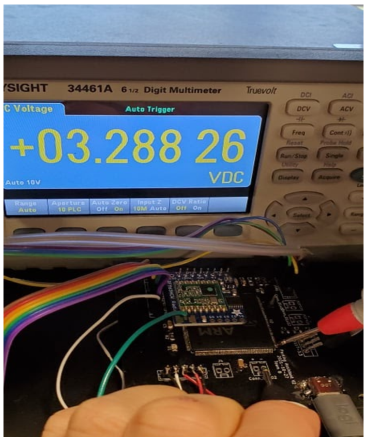
The use of low frequency to transmit the packets was intended to be able to transmit the packets at long distances. With the RF transceiver that was chosen, the datasheet stated that the simple antenna can transmit 200-500 meters away [6] with line-of-sight which was much higher than our requirements since we required the transceiver to be able to transmit 150 ft away, which is about 45 m. Since the simple antenna was sufficient, there was no need to install a more complex antenna that would cost more.

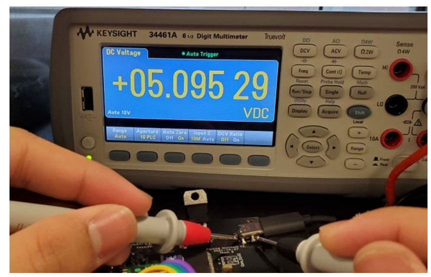
2.2.6 Central Logger System

The central logger system was intended to be easy to read with all of the important details visible, the license plate number and timestamp. A program was developed to receive the transmitted packet from the transceiver in the main module to the central logger system and log in a CSV file. The accuracy of the timestamp should only need to be precise to a few seconds since most parking lots only have increments to the minute.

3. Verification

3.1 Power System





Figures 8 and 9. Power System Verification

The verification for the Power System was performed using probes connected to a multimeter. In order for this to be possible, we needed both the USB-C connector and the voltage regulator soldered onto the board. Due to our workflow process, we had everything else soldered as well. Once everything was soldered in place to the PCB, we probed the legs of the voltage regulator and obtained the necessary values as specified in our RV table (Appendix B). As Figure 8 and Figure 9 show, voltage values of 5.0 ± 0.2V and 3.3 ± 0.2V were obtained as expected.

3.2 IR System

Two tests were performed to verify the IR System, one before the soldering took place and one after. The first test involved connecting the IR sensor directly to an Arduino and using it to determine the distance between itself and an object a set distance away. This test verified the IR sensor's accuracy for objects at the 1-1.5 meter range as needed by our design. This gave us the confidence to use this module for our final design.

The second test involved the IR sensor after it was soldered to the PCB. Once the connections between the IR sensor and the Control System were made, a status LED separately installed on the PCB would be set-up to flash when the output voltage of the IR sensor reached a certain value such that it would signal when an object is at a set distance away. This was also verified by moving an object closer and further from the IR sensor and seeing if the status LED would flash as expected.

3.3 Camera System

Some camera systems have difficulty capturing images in various temperatures due to the delicate nature of the sensors and lenses. In order to verify whether the camera we were using was suitable for the task in any weather, our first test involved placing the camera in environments of various temperatures and testing the functionality. We tested the warm weather functionality of the camera by placing it in a hot water bath for 5 minutes, then testing it. We tested the cold weather functionality of the camera by placing the system in a freezer for 5 minutes, then testing it again. In both cases the Camera System performed as expected.

Once the operating conditions of the Camera System were established, we moved onto testing the camera's functionality. The camera would be set up with the Arduino Uno we were using to verify individual components. When we ran the program for the camera, we manually sent a signal to the camera using the SPI communication protocol to take an image. Below is an example of one of those images.

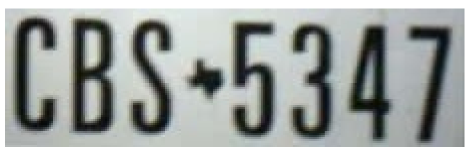


Figure 10: An example image taken with the Camera System during testing

Something to note is that the camera's performance was hampered by the testing process as we were physically holding the camera and taking images at various distances and angles without focusing the camera before each shot or adjusting for glare. When implemented, the Camera System setup will be fine-tuned to ensure a consistent shot every time.

3.4 Control System

Since the Control System was at the heart of the entire product and connected to all the systems on the PCB, the first step in the verification process was to make sure that the microcontroller unit (MCU) could be recognized when connected to a computer. This was how we verified that one of the MCUs failed as a result of an orientation mistake during the soldering process. The other chip was recognized as normal.

The remaining tests involved seeing if the chip was able to be connected to the other Systems on the PCB. Connection to the Power System was made and verified by testing to see if the chip was receiving any power at all by flashing to the chip and verifying the flash. Connection to the IR System was established as expected which was verified using the status LED mentioned in Section 3.2 by seeing if it would flash when the IR sensor encountered an object at a set distance.

Unfortunately, connections could not be made between the MCU and the RF System and the Camera System. While we have no doubt that this would be possible given enough time, the lack of available libraries for us meant that we were unable to implement and verify this section of the Control System requirements.

Another important feature of the microcontroller was the ability to identify the license plate number from the picture taken by the camera. Our OCR program was able to correctly identify license plates obtained from the Internet with an accuracy of about 85%. Furthermore, our code was able to identify 80% of the characters on license plates we took with our camera we used for the main module correctly. The accuracy could potentially be increased since the pictures that were taken have a lot of variance due to human error as mentioned in the previous verification section. Once a consistent picture was able to be taken, accuracy could potentially increase as well.

3.5 RF System

The RF System's requirements involved the reliable transmission of data through any expected parking structure so that the product could easily be implemented in a variety of parking lot/structure designs. The first test was to verify that the RF System could transmit data over longer distances as not all parking spaces are located near a singular point (where the Central Logger System would be located). We decided on a distance of 150 ± 5 ft despite LoRa's capability for longer distances to ensure an accurate signal would be received every time. This was achieved by connecting two RF transceivers to each other and sending packets back and forth between them over a distance of 150 ft.

The second test ensured that the RF signal would be able to be transmitted through any structures that might exist in parking lots. Since multi-story parking garages tend to be concrete structures, we also tested the RF signal could be transmitted through concrete pillars by placing each two RF transceivers connected to each other behind separate concrete pillars. This was verified in an actual parking garage.

Lastly, we recognized that the RF signal would need to be encrypted due to the possibly personal nature of the data being transmitted. By implementing an encryption key, we were able to reduce the likelihood of data compromise. This was verified by attempting to receive encrypted data without an encryption key, in which case we would receive something utterly unrecognizable.

3.6 Central Logger System

The Central Logger System was responsible for the intake of data from all the Parking Assistants located throughout the parking lot and using that information to determine when a car entered and left a parking spot. The Central Logger System must therefore reliably receive incoming data from an RF transceiver and be able to store it in an easy-to-read format.

To verify that an incoming signal from an RF transceiver was read correctly, a .csv file was set up that would be able to receive any data manually inputted into the RF transmitter connected to the Central Logger System. Below is an example of what the CSV file might look like as license plates are recognized and sent to the Central Logger System.

cars

02/12/2021 2:45:27 PM	EROS
02/12/2021 2:46:32 PM	BP76108
02/12/2021 2:47:55 PM	CUK6700
02/12/2021 2:48:03 PM	B933
2/12/2021 2:49:21 PM	8ROM674
2/12/2021 2:50:81 PM	ABC92P

Figure 11. Sample CSV file at the Central Logger

Given that CSV files in our case are lightweight in terms of their storage capacity, we determined that 10MB of storage would be sufficient to require for a large parking setup with many Parking Assistants. This requirement was verified by checking the remaining storage in the computer used as the Central Logger System as this can depend on which computer the Central Logger System is set up on.

4. Costs

4.1 Parts

Table 1. Part Costs

Item	Manufacturer	Part #	Retail Cost	Bulk Cost
Camera	Arducam	OV2640	$25.99	$25.99
IR Sensor	Sharp	R316-GP2Y0A710YK	$14.00	$14.00
Microcontroller	STMicroelectronics	STM32H743BIT6	$21.10	$16.74
Transceiver	Adafruit Industries LLC	3070	$9.95	$9.95
Power supply	iMBAPrice	iMBA-5V500MA-1PK	$6.99	$6.99
Regulator	STMicroelectronics	LD1117 3.3V	$0.76	$0.33
Case	Polycase	XR-44PMBR	$6.41	$3.88
USB C Input	GCT	USB4110-GF-A	$0.47	$0.29
PCB	PCBWay		$5.00	$1.00
Module Cost			**$90.67**	**$79.17**
Transceiver	Adafruit Industries LLC	3070	$9.95	$9.95
Computer	Arduino	Arduino Nano	$13.00	$13.00
Total Cost			**$113.62**	**$102.12**

4.2 Labor

Based on UIUC’s ECE website, the average starting salary of a computer engineering student was $96,992 and the average starting salary of an electrical engineering student was $76,079 [7]. Since our team contains two computer engineering students and one electrical engineering student, the average salary for the three of us based on the website is $91,232. We estimated that we each worked around 10 hours per week on developing our prototype for the past eight weeks.

Furthermore, we needed help from the machine shop which is estimated to cost $70/hr [8]. The work that needed to be done on the case was relatively simple and Greg Bennett, the research lab shop supervisor, estimated it would take around four hours to finish it.

Student labor: Average Salary = $91,232 ≈ $45/hr,
$45/hour x 10 hours/week x 8 weeks x 3 people x 2.5= **<u>$27,000</u>**

Machine shop labor: $70/hour x 4 hours = **<u>$280</u>**

5. Conclusions

5.1 Accomplishments

At the end of the semester, we were able to create a viable PCB design that supported a power input that was both easy and cheap to use and scale. We were also able to identify and procure components that would work best for our device. For example,choosing the right IR sensor for our use case was key. We wanted to be able to detect a car entering the lot within a range of distance and we successfully managed to get the IR sensor to both detect as well as interact with the MCU.

Some other accomplishments during the duration of this project included being able to build and verify each individual subsystem successfully. We also would like to acknowledge the knowledge gain we all had having gone through this project. We strongly believe that we all learned something new having worked on all the components of this project collaboratively and equally.

5.2 Ethical Considerations

A potential ethical issue related to image data could arise if the images taken by our solution contain sensitive data which may violate a person's privacy and in turn violate the 1st Code of Ethic [9]. Our solution thus aimed to avoid any sort of image data being transmitted by performing on-board image segmentation and transmitting only the acquired license plate data (which is public knowledge) through RF. This avoided any privacy breaches by disallowing any sensitive information to be vulnerable during transmission. To further avoid any identifiable information such as License Plate numbers from being used to track a person's location at a given time, we also included an encryption algorithm that allows the transmission of the plate numbers to be secure.

Another source of concern could be encountered while handling the payment information, specifically associating license plates with Credit Cards(or other forms of payment). A simple way to deal with this would be to make use of industry standard encryption protocols and make use of secure services to facilitate payments.

5.3 Future Work

Due to the lack of software support for a number of our components, specifically for our MCU choice, we were unable to complete the system integration. To overcome this challenge, one of our primary goals is to create support for these components and also share them as Open Source software so that others may benefit from our work. This will lead to a complete integration with every component.

Another area of the project that could benefit from work is the payment system. We hope to add in an OAuth based payment system that would utilize the information that has been collected from the device and also generate a secure platform to conduct transactions. This platform would be customizable for the client to tune the pricing as well as the UI.

Lastly, we would love to improve the scalability of this device, considering that all of our components were bought in small quantities, our costs can only get lower if we were to create bulk orders. Another way to cut costs would be to make use of individual components rather than their development board versions to avoid paying for pinouts that we don’t necessarily utilize.

Appendix A: PCB Design

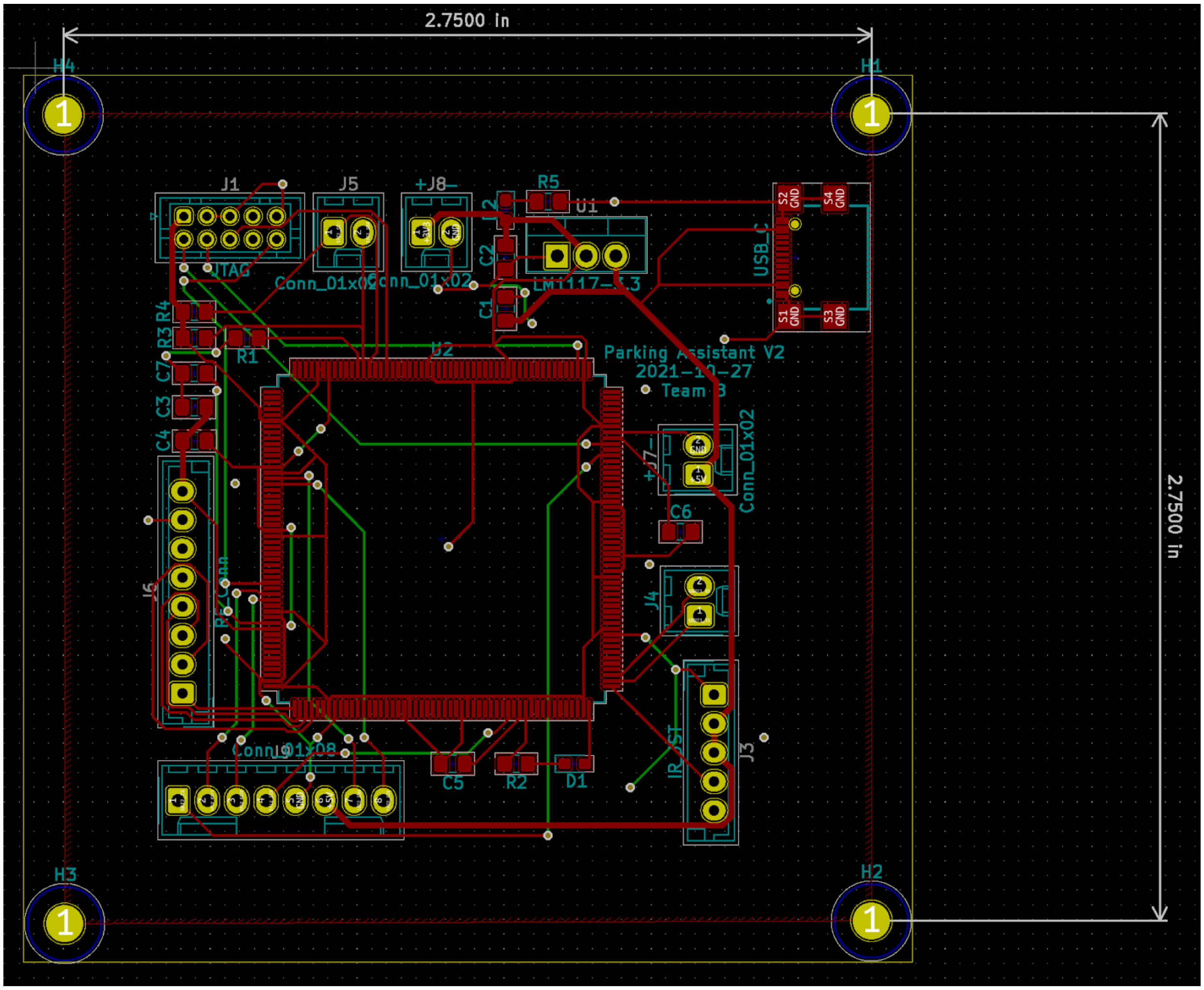


Figure 12. PCB KiCAD Design

Appendix B: Requirement and Verification Table

Table 2 Power System Requirements & Verifications Table

#	Requirements	Verifications	Verified? (refer to images)
1	Accept 120-240VAC as input and output 5±.2 V.	Use a multimeter to check the voltage going to the PCB at the USB-C input.	Yes
2	Outputs 5±.2 V to the camera and the IR sensor.	Use a multimeter to check the voltage going to the camera and IR sensor and see that it is showing 5±.2 V	Yes
3	Outputs 3.3±.2 V to the microcontroller and the transceiver	Use a multimeter to check the voltage going to the microcontroller and transceiver and see that it is showing 3.3±.2 V	Yes

Table 3 IR System Requirements & Verifications Table

#	Requirements	Verifications	Verified?
1	The sensor must be able to detect the car's distance correctly within 10 cm.	Point the IR sensor at an object at a known distance less than 2 meters away and check the distance measured.	Yes (video)
2	The sensor must be able to send a signal to the microcontroller if an object is identified to be within 120 ± 10 cm of the sensor.	Point the IR sensor at an object 120 cm away and check the status LED on the PCB to see that it lights up.	Yes (live demo)

Table 4 Camera System Requirements & Verifications Table

#	Requirements	Verifications	Verified?
1	The Camera System must be functional between ambient temperatures of -10°C to 40°C.	<u>-10°C</u> Put the camera module in a freezer for 5 minutes and verify its functionality. <u>40°C</u> Put the camera module in a warm water bath for 5 minutes and verify its functionality.	Yes
2	The Camera System must be able to receive a signal from the microcontroller that allows it to take a picture and send the picture data out.	Send a signal through the SPI that should take a picture and verify whether or not the output data stream contains the picture that was taken.	Yes (video)

Table 5 Control System Requirements & Verifications Table

#	Requirements	Verifications	Verified?
1	MCU is programmable.	Flash a program using *STM32CUBEIDE* and verifying the flash.	Yes (live demo)
2	MCU connects to the IR subsystem	Use the *STATUS_LED* to verify that the object has been detected within a given distance.	Yes (live demo)
3	MCU connects to the RF subsystem	Use the RF receiver connected to the central logger to verify receiving a signal	No
4	MCU connects to the Camera Subsystem	Verify that an image has been successfully captured using the *STM32CUBEIDE*	No
5	MCU connects to the Power Subsystem	Verify that the MCU is successfully powered on.	Yes
6	The OCR program must be able to correctly identify the license plate number from a picture with an accuracy of 85%	The program will read 433 images of license plates in the Kaggle License Plate Dataset, and it must correctly identify at least 368 of the images.	Yes
7	The program must be able to correctly identify the license plate number from the pictures taken from the Camera System with an accuracy of 80%.	20 pictures of license plates around UIUC campus will be taken using the Camera System, and it must correctly identify 17 of the license plates.	Yes (Live Demo)

Table 6 RF System Requirements & Verifications Table

#	Requirements	Verifications	Verified?
1	The transceiver must be able to successfully transmit the correct data from a distance of 150 ± 5 ft to ensure a successful communication over a long distance.	Have a LoRa receiver successfully receive a packet from the transceiver that is 150 ± 5 ft away.	Yes (video)
2	The transceiver must be able to successfully transmit the correct data through concrete structures to maintain a good signal in a car parking lot.	Have a LoRa receiver successfully receive a packet from the transceiver that is behind a concrete structure.	Yes (video)
3	The transceiver must be able to encrypt and decrypt messages.	Have a pair of LoRa transceivers communicate with each other with a set encryption key and verify the message transmitted is successfully decrypted.	Yes (live demo)

Table 7 Central Logger System Requirements & Verifications Table

#	Requirements	Verifications	Verified?
1	The computer program must be able to read the incoming RF signals from the PCB using the USB transceiver.	Send data packets using 915 MHz and see if the computer receives the packets through the transceiver.	Yes (live demo)
2	The computer must have a database of 10 MB to store the parking lot occupancy data.	Check the computer to see how much space it has allocated for the program and the database.	Yes (live demo)

References

[1] Kumar, P., & Goel, A. (2016). *Concentration Dynamics of Coarse and Fine Particulate Matter at and Around Signalised Traffic Intersections. Environmental Science: Processes & Impacts,* [Online] *18(9)*, 1220–1235. https://doi.org/10.1039/c6em00215c.

[2] Federal Highway Administration. Traffic Congestion and Reliability: Trends and Advanced Strategies for Congestion Mitigation. [Online]. Available: https://ops.fhwa.dot.gov/congestion_report/

[3] *LD1117 Adjustable and Fixed Low Dorp Positive Voltage Regulator Datasheet*, STMicroelectronics International N.V., Geneva, Switzerland, 2020

[4] *GP2Y0A710K0F Distance Measuring Sensor Unit Measuring Distance: 100 to 550 cm Analog Output Type Datasheet*, Sharp Corporation, Osaka, Japan, 2006

[5] Tensorflow, TensorFlow Lite for Microcontrollers. [Online]. Available: https://www.tensorflow.org/lite/microcontrollers

[6] *RFM69HCW ISM TRANSCEIVER MODULE V1.1 DATASHEET*, Shenzhen HOPE Microelectronics Co., Ltd., Shenzhen, China, 2014

[7] Grainger Engineering office of Marketing and Communications. Salary Averages. [Online] Available: https://ece.illinois.edu/admissions/why-ece/salary-averages

[8] Patriot Manufacturing, LLC. CNC Pricing - Are you Getting the Best Possible Rates? - Patriot Manufacturing. [Online]. Available: http://patriotmfg.net/cnc-pricing-best-possible-rates/

[9] IEEE, IEEE Code of Ethics. [Online] Available: https://www.ieee.org/about/corporate/governance/p7-8.html