VACANT PARKING DETECTOR 2.0

By

Jiahe Liu

Qingtao Hu

Zeyu Zhang

Final Report for ECE 445, Senior Design, Spring 2018

TA: Zhen Qin

02 May 2018

Project No.21

Abstract

This project is a vacant parking detection system 2.0 version. It has ultrasonic ranging sensors monitoring the parking space and spotting any parked vehicle. Then all the occupancy status information is centralized in control hub through wireless communication. At last, the control hub shares such information on a local web server, and drivers can use their smartphones to visit the server and check parking availability before heading to the street. In practice, the whole system operates steadily, and the project webpage displays accurate and updated parking space occupancy information that reflects the real-world situation. In short, this project is a successful proof of concepts for detecting objects using echolocation and sharing data wirelessly. It also has some practical values that yet to be discussed.

Contents

1. Introduction
This section serves as a brief introduction to our project objectives as well as improvements comparing to a previous design by other students.

1.1 Project Objectives
Finding a parking space is a time-consuming process during rush hours. A research from USAToday states that “motorists spend an average of 17 hours a year searching for spots on streets, in lots, or in garages." [1] INRIX Research, a renowned company specializes in transportation, combined the world's largest parking database with a survey, declaring that searching a parking space costs $345 per driver in wasted time, fuel and emissions. [2] With the development of societies, the number of personal automobiles keeps increasing. Thus, cities become more and more crowded and people need even more time to find a parking lot. This headache problem not only affects mood negatively since time is wasted, but also exacerbates the growing air pollution problem in big cities.

To save time and energy wasted, we plan to design and implement a parking space monitor, which detects parking spaces available, locates the available parking space for users, and helps users to find a parking space efficiently. Using an ultrasonic sensor, we can determine if there is a car in the parking space. The real-time information could be updated to the microcontroller with WLAN, which processes data and uploads them to the server. Then users could access the real-time parking space data from a server via the mobile device and be fully prepared for the trip.

1.2 Project Considerations
Team39, Fall 2017 has designed a similar parking detector system. [3] Even though the objectives of two projects are same, we have major differences that make our project more practical.

1) For the sensor to detect the availability of parking space, team39 uses infrared sensor instead of the ultrasonic sensor.
    a) The accuracy of the infrared sensor is vulnerable to temperature and daylight change. Ultrasonic sensor, however, is reliable in all weather conditions.
    b) Infrared sensor costs 13.95 each on SparkFun website. But ultrasonic sensor only costs $3.95.
    c) The detection limit of the infrared sensor is 10cm to 80cm. [4] But ultrasonic sensor can detect the object with distance ranging from 2cm to 400cm. [5]
2) Team 39 uses radio frequency transceiver for communication. Our project uses ESP8266 Wi-Fi for communication.
    a) A transceiver cannot connect to another Wi-Fi hotspot. But ESP8266 is able to connect hotspot and create a web page.
    b) Wi-Fi communication has a standard communication protocol.
3) Team 39 has the central hub display and LED to show the status of parking spaces. In addition to LED, we designed a web page, so that every user can access the status information conveniently.

2. Design Procedures and Details

The detection system can be separated into five different modules: Power Module, Sensor Module, Control Module, Notification Module, and Mobile Device. The Power Module keeps Control and Notification Modules running; while Sensor Module is powered by its own solar battery pack. The Sensor Module detects vehicles within its duty range and reports the space occupancy status back to central control via WLAN. The Control Module then gathers status information from multiple Sensor Modules, and upload the organized data to a local web server. The Notification Module is the program that creates and operates the local web server. Last but not the least, drivers can use their Mobile Devices to access the local web server for real-time parking status information.

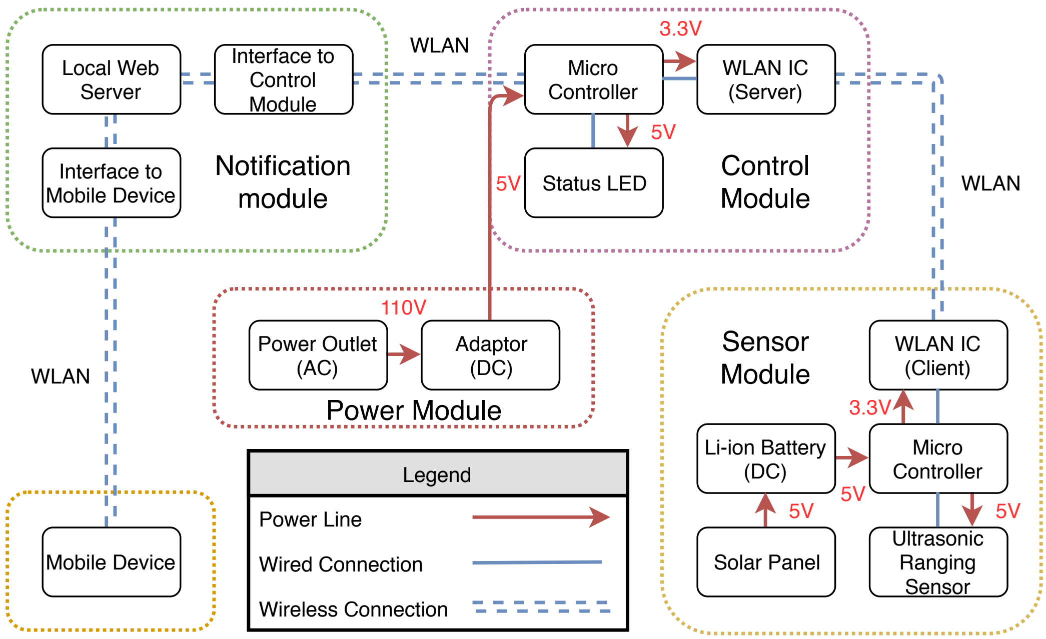

Figure 1: Project Block Diagram

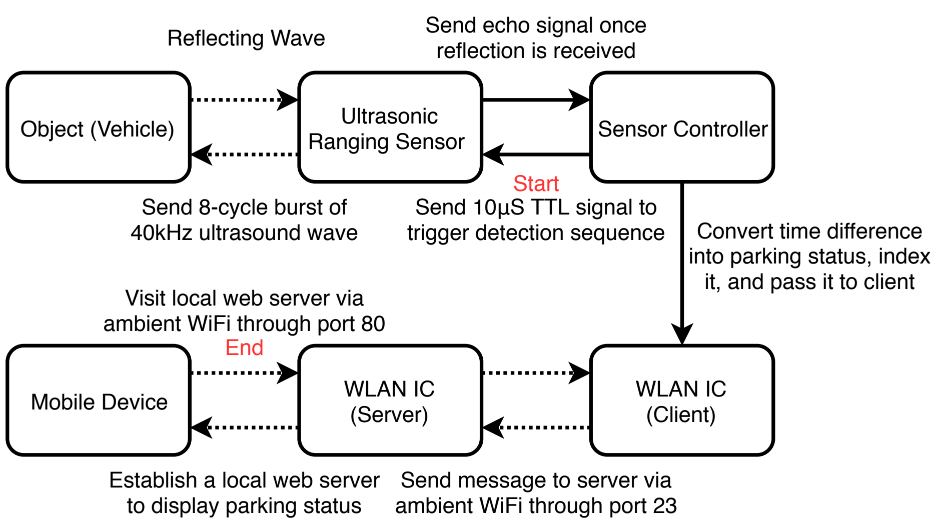

Figure 2: Project Data Flowchart

2.1 Sensor Module

This module detects vehicles within its duty range and reports parking space status to control module wirelessly. It contains four major components: an HC-SR04 ultrasonic ranging sensor (URS) for object detection, an ATMEGA-328P micro controlling unit (MCU) for modular coordination, an ESP-01S client WLAN IC for wireless data transmission, and a 5V solar battery back as the power supply. We have deployed two sets of the sensor module for the proof of concept, with each of them monitoring a single parking space. The circuit schematic is derived from an online example of Arduino clone board with modifications to fit our project needs. [6]

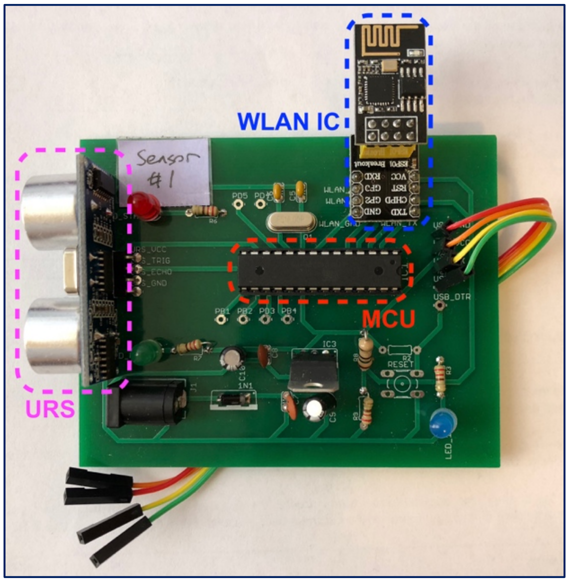

Figure 3: Sensor Module Physical Design

2.1.1 Ultrasonic Ranging Sensor (URS)

Each sensor has a transmitter and receiver. The detection sequence is initiated by a trigger signal from sensor module MCU and the transmitter then sends out a burst of ultrasound waves. When the emitted waves hit an object nearby, the reflected waves are captured by the receiver after a small time increment and an echo signal is sent back to sensor module MCU. Time difference between trigger and echo is used to calculate object distance, namely the process of echolocation. The longer time it takes, the farther the object is. The sensor can detect objects that present between 4cm to 4m away from it, and we only use it for up to 2.5m in our project. The conversion from time difference to distance is illustrated below in Figure 4.

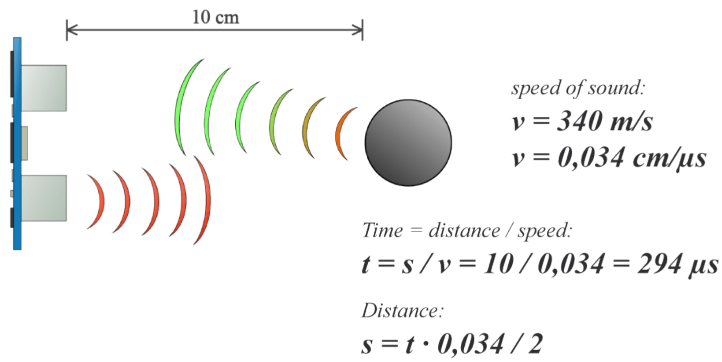

Figure 4: URS Detection Schematic [7]

2.1.2 Micro Controlling Unit (MCU)

The microcontroller coordinates all operations inside the sensor module, from initiating detection sequence and analyzing URS raw signal, to forwarding detection result to client WLAN IC. The formulas used in data conversion is listed below in Formula (1) and Formula (2). Basically, it takes in the time difference t from URS and calculates the object distance d. Then by applying a proper threshold to d, the MCU can determine whether the parking space is occupied. (Note: t is time for a round trip between sensor and object, so there is a 0.5 factor for distance calculation, v is the speed of sound, b is the Boolean value representing parking space status, and $\mathrm{U(x)}$ is the unit step function. We use 2 meters as the distance threshold)

$$\mathrm{d} = 0.5 * \mathrm{v} * \mathrm{t} \tag{1}$$

$$\mathrm{b} = \mathrm{U}(2 - \mathrm{d}) \tag{2}$$

2.1.3 Client WLAN IC

The client WLAN IC is paired with the server in control module to transmit local parking space status. It acquires such data from sensor module MCU via onboard serial communication, and send it to the server through environment Wi-Fi network. This process is repeated every 2 seconds to ensure control module get the real-time status. (Note: WLAN IC required 3.3V input, so there is a voltage divider built for it to supply satisfying power)

2.1.4 Solar Battery Pack

It is one of the off-the-shelf components that supplies 5V-1A DC to power sensor module. We only deploy one pack as a proof of concept that this module can use sustainable energy. It would be more practical to power our circuit with solar energy once the price for solar panels is sufficiently low.

2.2 Control Module

organized data to a local web server created by the notification module. It contains three major components: an ATMEGA-328P micro controlling unit (MCU) for modular coordination, an ESP-01S server WLAN IC for wireless data reception, and some LEDs for parking space status indication. The module is powered by a 9V alkaline battery pack with a voltage regulator circuit restricting a 5V output to the primary circuit. In practice, the MCU is dropped and all computations are moved to server WLAN

IC, who still operates smoothly when interacting with both multiple sensor module clients and a mobile device at the same time.

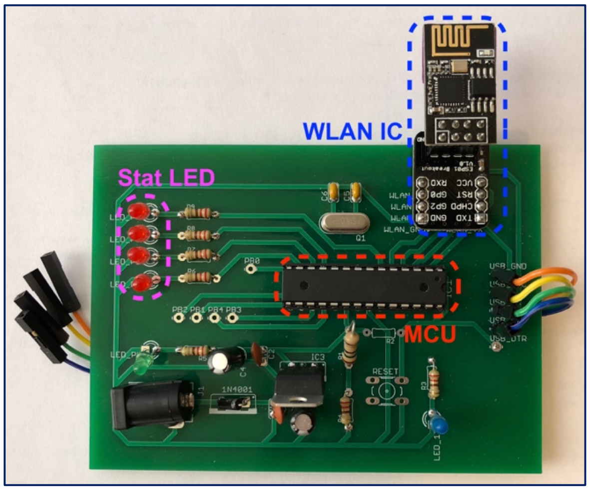

Figure 5: Control Module Physical Design

2.2.1 Micro Controlling Unit (MCU)

It was originally used as client message decoder by identifying client index as well as corresponding parking space status. Later when the design shifted to display data on a local web server, the MCU got abandon, since the server WLAN IC handled all the work smoothly.

2.2.2 Server WLAN IC

The server WLAN IC is paired with multiple clients in sensor modules to receive messages of local parking space status through communication port 23. This process is repeated until there is no more new client message. The IC also creates a local web server to share received parking status information online through communication port 80. Drivers can use their mobile devices to check such information by visiting the static IP address of the server.

2.3 Power Module

This module contains an off-the-shelf 5V-1A DC adaptor that supplies reliable power to whichever module needs. This setup limits the mobility of our project, hence in practice, all hardware modules are powered by either solar battery or alkaline battery. However, batteries are never a long-term power source, so the adaptor is useful for longstanding project deployment.

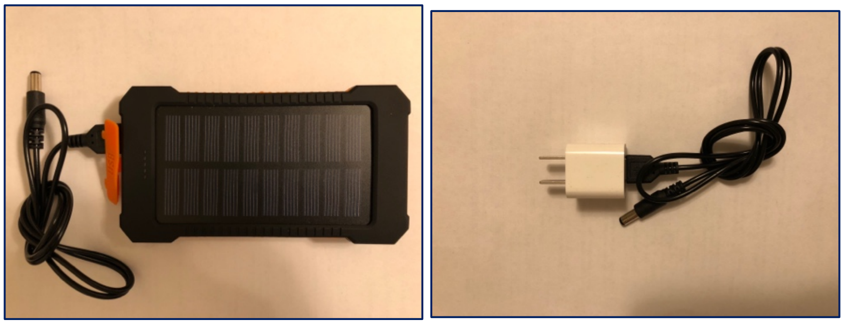

Figure 6: Solar Battery Pack (Left) & Power Adaptor (Right)

2.4 Notification System

This system is one of the two software modules in our design. It is basically a website that shares organized parking information from control module online within the environment network boundary. Its IP address is programmed to be static for easier access on mobile devices. Raw data shared online are converted into simple text and graphics for the sake of readability.

Figure 7: Notification System Website

2.5 Mobile Device

This is the other software module in our design. It is the terminal of data sharing, because the ultimate objective of the whole system is to inform drivers about real-time parking information. Drivers can access such information on the website using any device that connects to the same network as the local web server. Once again, it is a proof of concept, and the data sharing mechanism is preliminary. In the future, the web browsing process could be replaced by a mobile application that helps drivers find parking space. And the local parking information could be shared globally such that users can check this information everywhere.

3. Design Verification

We have conducted several unit tests on hardware module components, specifically on the ultrasonic ranging sensor, the micro controlling unit, and the WLAN ICs. Passing these tests lays the foundation for success in modular and system integrations.

3.1 Sensor Module

We focus on testing the URS, MCU, and client WLAN IC.

3.1.1 Ultrasonic Ranging Sensor (URS)

To verify that the URS can precisely detect objects within the selected distance threshold. We compared the reading from the URS with the physical measurement ranging between 0.5m to 2.5m. The result is tabulated below in Table 1. As shown in the table, URS readings accurately represent the physical world distance, with some tolerable measurement differences caused by human errors.

Table 1: URS Distance Reading vs. Physical Measurement

Test Point (#)	1	2	3	4	5	6	7	8	9	10
URS Reading (cm)	45	67	89	113	135	158	181	203	226	249
Physical Measurement (cm)	45.6	68.4	91.2	114	136.8	159.6	182.4	205.2	228	250.8
Differences (cm)	0.6	1.4	2.2	1	1.8	1.6	1.4	2.2	2	1.8

We also tested the impact of horizontal and vertical deviations on distance measurement. Since the URS only has a 15-degree valid detecting angle, [5] we want to verify the detection mechanism works when the emitted wave from the URS does not incident the object in the normal direction. The result is tabulated below in Table 2. As shown in the table, the URS works better in near field, and its actual detecting angle drops linearly as object distance increases. This phenomenon helps us to determine the mounting angle (45 degrees toward the parking space) and height (0.5m above the ground) of our sensor module in practice. Such setup renders incident wave hit vehicle's bumpers in most cases, and the reflection wave is in a wide range of angles, including the one pointing to the receiver of the URS.

Table 2: URS Detection Deviation Tolerance

Test Point (#)	1	2	3
Distance (cm)	0.5	0.75	1
Horizontal Deviation (degree)	+/-24	+/-16	+/-10
Vertical Deviation (degree)	+/-20	+/-13	+/-8

3.1.2 Micro Controlling Unit (MCU)

To verify that the MCU can convert time difference into parking space status, we programmed the MCU with detection and data conversion functions, then monitored its serial outputs. "Client" represents the identity of the serial message; "Status" indicates the parking space occupancy; "Distance" is the intermediate result in cm. The distance threshold is selected to be 200cm. When the measured distance is less than the threshold, status shows 1 as space occupied, otherwise status shows 0 as space vacant.

```
Client{1} Status{1} Distance{159}
Client{1} Status{1} Distance{157}
Client{1} Status{1} Distance{156}
Client{1} Status{1} Distance{162}
Client{1} Status{0} Distance{316}
Client{1} Status{0} Distance{316}
Client{1} Status{0} Distance{317}
Client{1} Status{0} Distance{316}
```

Figure 8: MCU Data Conversion Result

3.1.3 Client WLAN IC

To verify that the client can send local parking status to the server, we programmed the client with message forwarding function, then monitored messages received on the server. The result is in the exact same format as in above figure, since the client only reads in serial messages and send it to the server without modification.

```
// Connect to server via port 23
if (client.connect(server, 23)) {
  digitalWrite(ledPin, HIGH);
  // Fetch message from serial
  String stat = Serial.readString();
  // Send message to server
  client.println(stat+'\r');
  client.flush();
  digitalWrite(ledPin, LOW);
  // Set refreshing rate to 0.5Hz
  delay(2000);
}
```

Figure 9: Client Message Forwarding Function [8]

3.2 Control Module

Since we have abandoned the control module MCU in practice, all verifications are related to server WLAN IC about handling messages from multiple clients in a reasonable time interval and interacting with mobile devices to display parking information.

First, to verify that the server can receive multiple client messages, we used the same technique as defined in section 3.1.3, except that we spanned the number of clients to two.

```
Client{1} Status{1} Distance{182}
Client{2} Status{1} Distance{190}
Client{1} Status{0} Distance{288}
Client{2} Status{1} Distance{191}
Client{1} Status{0} Distance{289}
Client{2} Status{1} Distance{191}
```

Figure 10: Server Receiving Multiple Client Messages

During the unit test, we found that the message processing time for the server is not always a constant. In a very rare situation, the server processes message faster than the 0.5Hz refreshing rate, hence it will read in an empty line from the client. To ensure the reliability of the server, we selected three batches of serial outputs on the server, each containing 50 lines of client messages. Then we manually counted the number of empty lines appeared in selected batches. The result is tabulated below in Table 3.

Table 3: Empty Line Rate on Server

Batch number (#)	1	2	3	Total
Number of empty lines in a batch	1	1	0	2
Empty line rate (out of 50)	2%	2%	0%	1.33%

As shown in the table, the client messages are successfully transmitted to the server with a rate of 98.67%. This rate is sufficient to provide accurate and timely parking information to the users. Even if there is an empty line read by the server, the supplement message containing useful information would arrive within 2 seconds.

Second, to verify that the server can respond to mobile device's request for parking information. We programmed the server to display web client credential and compared it to the identity of the smartphone.

```
New Web Client.
GET / HTTP/1.1
Host: 172.20.10.10
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,
User-Agent: Mozilla/5.0 (iPhone; CPU iPh
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive

Client disconnected.
```

Figure 11: Server Handling HTTP Request

3.3 Power Module

To verify that both solar battery pack and power adaptor provide 5V DC output, we wired each of their outputs to a 10kΩ resistor and measured the voltage across it. The result is tabulated below in Table 4. It is obvious that they provide a sufficient and safe output voltage for other hardware modules. But since the voltage regulator in both sensor module and control module has a cutoff voltage of 5.8V, it must be bypassed before pairing power module to other modules. This is one of the minor defects in our circuit design, but it does not impair any system functionality.

Table 4: Power Module Output Measurement

Power Source	Output Voltage
Solar Battery Pack (V)	5.11
Power Adaptor connecting to 110V outlet (V)	5.11~5.12

4. Costs

This section is a brief estimation of out project costs, evaluated based on both parts and labors.

4.1 Parts

According to the parts price, each sensor module costs \$28.25 (without solar battery), \$38.61 (with solar battery). Each control module costs \$20.35(without solar battery), \$30.71 (with solar battery).

Table 5: Project Parts Costs Estimation

Description	Part No	Quantity	Cost/unit	Total Cost
Ultrasonic Sensor	HC-SR04	2	\$3.95	\$7.9
Wifi Module	ESP8266	3	\$6.95	\$20.85
Microcontroller	Atmega328	3	\$4.12	\$12.36
5V 5W solar battery	N/A	1	\$15.64	\$15.64
AA battery	N/A	12	\$0.88	\$10.56
PCBs	N/A	3	\$1	\$3
Resistors, Regulators, LEDs	N/A	N/A	N/A	\$10
Total				\$80.31

4.2 Labor

Our fixed development costs are estimated to be 30/hour, 8 hours/week for three people. We consider we need approximately 10 weeks to finish the design:

$$\frac{3\times 50}{hr}\times\frac{8hr}{wk}\times 10wks\times 2.5 = 30{,}000 \tag{3}$$

5. Conclusion

This section concludes our project by addressing our accomplishments, design uncertainties, project ethical considerations, and possible future work.

5.1 Accomplishments

The vacant parking detector project is successful not only in the lab test, but also in the real-world parking lot test. The detector detects vehicle's occupancy status by emitting and receiving signals via ultrasonic sensors, transmits data to one server from multiple client Wi-Fi IC units, and then wirelessly sends the data to the user. During the test, all LED bulbs operate correctly, the wireless connection is stable, and users can get latest parking status on their phone. All modules play their roles correctly and the integration of all modules works as expected.

By using this product, parking lot owners can remotely monitor the parking status, and parking enforcement officers can have their jobs done without driving around the city. On the other side, drivers can get more informed about parking status and make better decisions.

5.2 Uncertainties

Although the vacant parking detector has stable performance in almost all the situations if it is set up correctly and used properly. There exist uncertainties in some rare cases because of multiple types of interferences.

First, the detector could suffer from interference during the ultrasonic sensor's process of occupancy detection because of other electronic devices emitting waves of the similar frequency. To avoid this insidious bug, engineers should check the interference in the parking lot carefully before the implementation of the system. To solve this problem, engineers can find the sources of interference and do appropriate steps, or they can simply change the frequency of the detector until there is no interference.

Moreover, certain materials could absorb the signals emitted by the detector. Because of the nature of the sound wave, sometimes the emitted sound wave is absorbed so that it is never reflected back to the detector. This will cause the detector to consider the area unoccupied. For example, if a person, who has trousers made of cotton, stands in front of the detector and waits for it to change status to "occupied", the detector will not change because the signals are absorbed by the cotton. Nevertheless, this uncertainty is not a big problem because people don't normally stand in the parking area for a very long time. If the parking lot owners are concerned, they can always set up more detectors or set up them in a specific location where people don't go.

Finally, Wi-Fi IC unit which transmits data between the sensor and control modules could suffer from other Wi-Fi interference. People have done lots of research on the subject and this problem is still unsolved, because this is the nature of Wi-Fi data transmission. This is also not a severe problem since this doesn't occur very often. When it occurs, it will also not influence user's experience if one data update is completed during a period. If people want to improve the system, they can add wires to ensure a stable connection between the server and the clients, but that can be a lot more expensive.

5.3 Ethical Considerations

When users use our vacant parking detector system, certain ethical issues could arise. Because of the limited number of parking slots, more than one drivers may look for places to park. At this point, the information about which slot is empty is very important for a driver to quickly occupy the place. According to IEEE Code of Ethics, #2: "To circumvent conflicts of interest among people related to the engineering design and tell them the whole story related to them when the conflicts exist," [9] our users should be able to get the updated information at the same time to realize the limitation of parking slots and to take appropriate actions regarding the situation. In our implementation, we use communication network devices that provide real-time data transmission to display the parking slots occupancy information to our users. Therefore, they can easily take the control of the situation and make more informed decisions.

To comply with IEEE Code of Ethics, #8: "To treat everyone in the same way regardless of their gender, sexual orientation, religion, disability, age, nationality, or other identity," [9] and avoid any discrimination in parking lot, it's important to design and create a fair environment for parking. Because of the choices of our hardware devices, there is no way to get information about our drivers' personal information such as gender and nationality. This makes the discrimination of any kind impossible.

By developing the vacant parking detector system, we believe this design of the system would benefit all the drivers by improving the efficiency of looking for a parking slot and letting them make more informed choices regarding the correct places to park. This is also a step following IEEE Code of Ethics, #1: “To regard public safety and wellbeing as the most important thing, to keep developing ethical practices, and to show all the possible hazards behind the implementation.” [9]

5.4 Future Work

There are many additional features that can help our vacant parking detector to become more powerful. By using a box to cover the hardware, it can be waterproof so that clients who set up the detector will not need to worry about its conditions on a rainy day. This feature is particularly useful when the parking slots are outside or in a damp underground parking lot.

Plate recognition algorithm will enable the detector to record plate number of the car and let database store more information. Nowadays many parking lots, especially those in big shopping malls, have very large scales. This makes it difficult for the drivers to remember where they park, and time-costing for them to walk around the parking lot to find their cars. If the parking system also stores information about vehicle's plate number, drivers can easily search on a computer or a mobile phone to retrieve information about their parking locations.

By implementing additional sensors and writing more algorithms, the detector can also be used to detect vehicle’s type. This way, parking lot owners can charge parking fee according to resources that different vehicles use. For example, since big trunks can occupy a larger area, their drivers may pay more.

As the system grows, communication between systems from different network becomes very slow and expensive. Moreover, ideally people may want one big system to control all the parking lots, but dangerous data collision or information loss can happen, when there is a failure in any member of the whole system. distributed system will avoid single-point failure and make the system cheaper and more efficient.

References

[1] usatoday.com, "Drivers spend an average of 17 hours a year searching for parking spots", 2017. [Online]. Available: https://www.usatoday.com/story/money/2017/07/12/parking-pain-causes-financial-and-personal-strain/467637001/. [Accessed: 08-Feb-2018].

[2] inrix.com, "Searching for Parking Costs Americans $73 Billion a Year", 2017. [Online]. Available: http://inrix.com/press-releases/parking-pain-us/. [Accessed: 08-Feb-2018].

[3] courses.engr.illinois.edu, "Parking Space Monitoring System", 2017. [Online]. Available: https://courses.engr.illinois.edu/ece445/getfile.asp?id=11666. [Accessed: 19-Feb-2018].

[4] www.sparkfun.com, "Infrared Proximity Sensor - Sharp GP2Y0A21YK", 2015. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf. [Accessed: 02-May-2018].

[5] elecfreaks.com, "Ultrasonic Ranging Module HC - SR04", 2013. [Online] Available: http://www.micropik.com/PDF/HCSR04.pdf. [Accessed: 08-Feb-2018].

[6] maker.pro, "How to Make an Arduino Clone Board". [Online] Available: https://maker.pro/education/how-to-make-an-arduino-clone-board. [Accessed: 19-Feb-2018].

[7] howtomechatronics.com, "Ultrasonic Sensor HC-SR04 and Arduino Tutorial", 2015. [Online] Available: http://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/. [Accessed: 19-Feb-2018].

[8] randomnerdtutorials.com, "Build an ESP8266 Web Server – Code and Schematics", 2015. [Online] Available: https://randomnerdtutorials.com/esp8266-web-server/. [Accessed: 02-May-2018].

[9] ieee.org, "IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 08-Feb-2018].

Appendix A Requirement and Verification Table

Table 6: Sensor Module Overall R&V

Requirement	Verification	Verification status (Y or N)
Must be protected from weather, i.e. have some covers for electronics and IC.	1. Seal the module with water-sensitive papers placed near wire and sensor outlets inside a plastic case. 2. Spray the case with water to simulate rain, unbox the module, and check the paper to see the effectiveness of protection.	N (It should not be hard, but we do not have enough time to implement it)

Table 7: Ultrasonic Ranging Sensor (URS) R&V

Requirement	Verification	Verification status (Y or N)
Must be able to detect the vehicle within 2m range using echolocation.	1. Mount the URS such that transmitter and receiver face the horizon. 2. Use a plate larger than $0.5m^2$ to simulate vehicle, and move it from 0.5m to 2.5m away from the URS. 3. Face the plate toward the URS with 0-degree, 15-degree, and 30-degree horizontal or vertical deviations. Check for tolerance in different orientations.	Y

Table 8: Sensor Module MCU R&V

Requirement	Verification	Verification status (Y or N)
Must be able to convert time difference into space status with a ~0.5Hz refreshing rate	1. Access both signals from the URS and the MCU. 2. Perform manual conversions and compare results to MCU output 3. Inject a 0.5Hz time-varying signal to the MCU and see if output updates timely	Y

Table 9: Client WLAN IC R&V

Requirement	Verification	Verification status (Y or N)
Must be able to send messages to Server WLAN IC with ~0.5Hz refresh rate	1. Inject a 0.5Hz time-varying signal to the Client WLAN IC 2. Check client message in Server WLAN IC to see if it updates timely	Y

Table 10: Control Module Overall R&V

Requirement	Verification	Verification status (Y or N)
Must be able to connect to all Sensor Module clients within range <= 25m via WLAN	1. After configuration, place two WLAN ICs 25m apart. 2. Confirm two-way wireless transmissions using predefined data packet.	Y

Table 11: Server WLAN IC R&V

Requirement	Verification	Verification status (Y or N)
Must be able to process data from different client WLAN ICs with ~0.5Hz refresh rate	1. Send a 0.5Hz time-varying signal from another WLAN IC to it 2. Monitor its serial ports to see if output updates timely	Y

Table 12: Peripherals R&V

Requirement	Verification	Verification status (Y or N)
Must be able to supply 5V±0.1V and <1A to support module normal operation	1. Connect the power supply to a 10kΩ resistor 2. Measure resistor voltage and current	Y

Appendix B Module Schematics

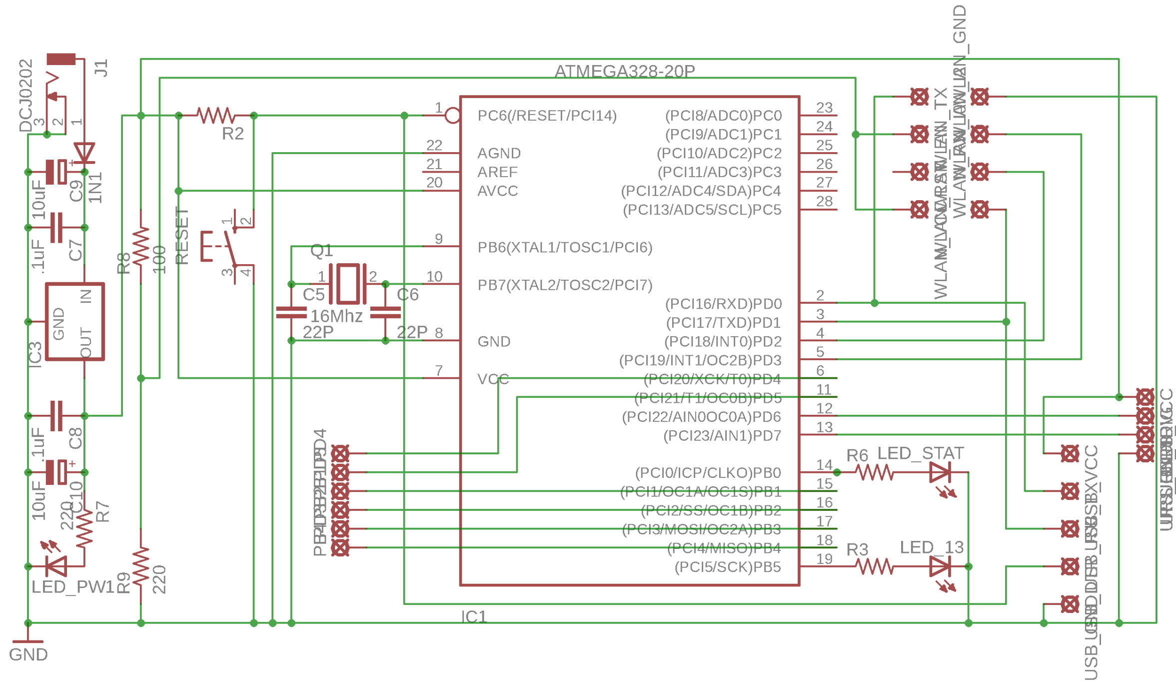

Figure 12: Sensor Module Schematic

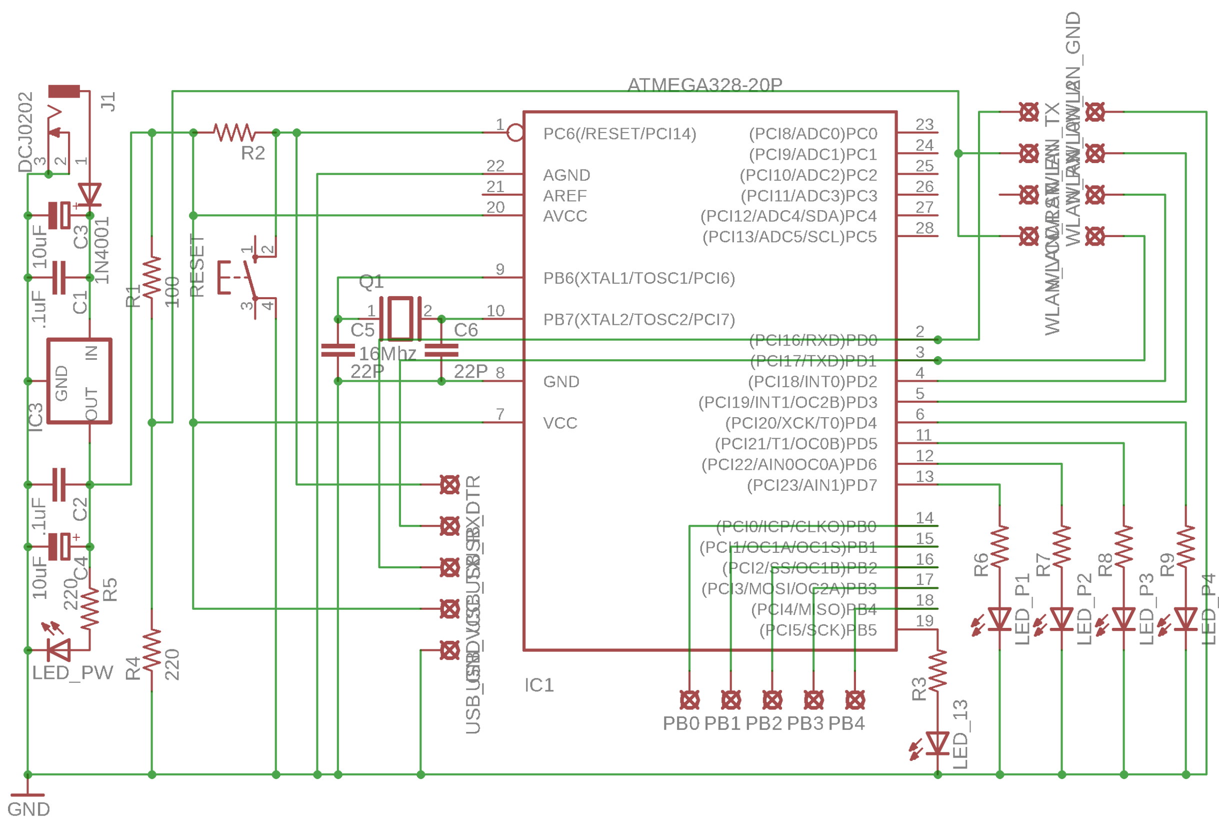

Figure 13: Control Module Schematic