Spring 2019

Smart Parking Deck

Ryne Turner
rmt68@zips.uakron.edu

Matthew McDade
mm263@zips.uakron.edu

Julie Aichinger
jha12@zips.uakron.edu

Laveréna Wienclaw
llw50@zips.uakron.edu

Smart Parking Deck

Final Honors Report

Design Team: 06

Team Leader: Julie Aichinger

Software Lead: Matthew McDade

Hardware Lead: Ryne Turner

Engineering Data Manager: Laveréna Wienclaw

Faculty Advisor: Dr. Igor Tsukerman

Date First Submitted:
26 April 2019

Table of Contents

(Written by Ryne Turner and Laveréna Wienclaw.)

Figures

Tables

Abstract

The Smart Parking Deck employs elementary circuit design elements and mobile application development. Each device module uses laser proximity sensors to check the availability of an individual parking space and a Zigbee unit to communicate with the adjacent device module. The modules are connected to a network hub that manages all of the incoming and outgoing parking data. This data is displayed on the mobile application. The system is easily manageable and energy efficient, significantly decreasing the costs associated with other smart parking systems on the market. This system is aimed at decreasing commute time for students by allowing them to find available parking spaces with ease. (Written by Julie Aichinger, Matthew McDade, Ryne Turner, and Laveréna Wienclaw.)

1. Problem Statement

Need

Decreasing the difficulty of finding parking spaces at the University of Akron is one of the primary stipulations this project seeks to fulfill. It is unusually challenging to find parking on campus throughout the day. With the new four-day week ushered in this year, the parking issue has only escalated, making the goals of this project even more necessary to implement in the near future. The current limitation is that there is no smart parking infrastructure in place, so this project design needs to be comprised of the very fundamental aspects of smart parking to the more advanced aspects such as spot occupancy metadata. (Written by Ryne Turner.)

Objective

The smart parking system will include a thorough combination of both hardware and software design. Within the parking deck, there will be sensor modules for each parking space location to detect the availability of the space. These sensor modules will communicate using a node network that will be connected to the Internet, feeding the current data from the modules to a database for the parsing and eventual analysis of the data. Applications will be designed to show drivers the presence or absence of available parking spots on a display via a mobile application for both the iOS and Android operating systems as well as displaying a simplified version of this data to a screened television-like display outside the entrance of the parking deck. (Written by Ryne Turner.)

Research Survey

The purpose of a smart parking system is to inform drivers of available parking spaces and decrease the amount of time spent on traveling. The system will be composed of a readable, all-weather user interface which can inform drivers of the locations of vacant parking spaces. A screen would display a summary of this information before the entrance to each parking deck where the system is implemented. A mobile application will also be used to provide users with more extensive versions of the collected data. The mobile application will display the number of available spaces so they are able to plan a route for finding parking before their arrival. This system not only provides greater convenience to the user, but it also provides a way to improve traffic efficiency on campus. This low-maintenance system will use laser proximity sensors positioned above ground in front of each space to detect the presence of a vehicle, and it will be connected to a network that will update the mobile applications and displays in real time. This will require an Internet-connected node per floor to send the information to a web server. In these ways the smart parking system provides a method for the University of Akron to maximize usage of its available spaces and decrease commute time. (Written by Ryne Turner and Laveréna Wienclaw.)

The advantage provided for the average Akron student is a more refined parking system that improves the on-campus traffic efficiency. Students need to have a productive time-management system to drive to class punctually and without the frustration of finding parking. It could be advantageous for the University of Akron to enact an inexpensive smart parking system that would help its students arrive in a timely and safe manner. The planned design of this system also helps collect data on traffic patterns that do not yet exist, potentially exposing other rooms for improvement in terms of which spaces are most advantageous to distribute for permanent residents versus commuting students. It could also support the new four-day week and the traffic pattern changes that have resulted, maintaining the orderliness that is necessary when many more students are present on campus for extended periods of time.

Smart parking is one form of technology that exists in some measure in the surrounding areas of Northeast Ohio, but it has yet to be actualized at the University of Akron. Older plans have existed for a brand of this system, but these plans used image processing upon entrance of the vehicle into the parking deck. This was a far less effective way of measuring which parking spaces are filled and would provide little to no discernible benefit to the student as it lacks the specificity needed on a space-to-space basis. One example of a successful demonstration of similar concepts would be the Cleveland Airport's Smart Parking Garage, CLE. This garage manages many vehicles through a combination of intuitive signage and valet parking. There are also very advanced automated parking deck systems used primarily for storage of cars for longer durations of time, but these systems exist far outside the range and scope of this project due to sensible financial limitations. (Written by Julie Aichinger and Matthew McDade.)

Some of the limitations of current smart parking designs include mobile applications with limited or unintuitive functionality, lack of a real-time update system for those entering the parking decks, and inaccurate recognition of smaller vehicles. These are all areas this design seeks to rectify. Many current designs could cost at least $70 per space, and these systems are ineffective in many cases due to the position of the sensor on the ground, bulk, and low battery life. The design of this proposal is a wireless circuit connecting sensors in a node network, which means it can be designed at lower cost, with lower maintenance, and with greater effectiveness. Little to no further maintenance should be required. There are two very different definitions of what a "smart" parking system is. The first definition is an advanced machine that automatically stores your vehicle in a mechanical compartment inaccessible from roadways. The second definition is an improved parking system based on the detection of empty spaces and the number of vehicles present in a parking deck. The second definition of a smart parking system is what this project means to substantially refine. Some of the small-scale goals of this project are designing a top-down, easy-to-read display for drivers entering the decks and a weather-resistant housing for the circuit. (Written by Ryne Turner and Laveréna Wienclaw.)

While researching specific technologies to use for implementing this idea, there are at least two relatively inexpensive types of sensors to calculate distance: infrared and ultrasonic.

Infrared sensors work by emitting an infrared light invisible to the human eye, which then reflects off any nearby surfaces. If this is applied to calculate distance, the sensor is placed near the initial infrared emitter so a timing calculation may be performed and translated into a quantified and useable voltage. Ultrasonic sensors work in a similar way, utilizing an emitter and receiver to time the difference between the sending and receiving signals but uses sound instead of light (Mohammad, 2009). After further research was performed, laser proximity sensors provided an even more affordable solution. These sensors use a light detection spectrometer to calculate distance.

Laser proximity sensors can offer high quality at short ranges. Since the current design is to mount these devices on the wall in front of the parking space, a measuring distance of two to four meters is required. Light bounces off reflective surfaces, such as a smooth metal, but have a harder time with non-reflective substances such as cloth or water. (Written by Matthew McDade and Laveréna Wienclaw.)

Two relevant patents were found to assist in the design of this project. The first one is described as "Method for managing a parking lot" (US Grant Number US7688225B1). It describes a way to manage a parking lot system and the spots within that system. It includes gathering parking data and transforming the data into highly usable metadata by using complex mathematical probability. This includes information about a moving object in a parking lot. It accomplishes this task using a video input and audio input device. It also includes an efficient method for transmitting a map of the parking lot to a mobile device. The second patent relevant to our project is titled "Computer-implemented system and method for managing motor vehicle parking reservations" (US Grant Number US8799037B2). It describes a method of managing parking locations through a server and uses sensors to detect the presence of a vehicle. This patent describes a need for the user to reserve the spot and to make a parking account. It then verifies the parking reservation against the identity of the motorist to allow use of the parking spot.

This technology anticipates the use of sensors, computer networks, and mobile applications. Both patents use similar technologies, but neither of them use all the technologies that are anticipated to be used during the design of this project. Because of this, it can be concluded that the design of this project does not risk intellectual property infringement. (Written by Julie Aichinger.)

The information of cars entering or exiting could easily be used to answer how many parking spots are left in a deck, but this number does not clue the user to the exact location in the parking lot where the open spot exists and still may lead to lost time or frustration on the part of the user. Pala and Inanc propose check-in and check-outs for parking payment using RFID to expedite the process (Pala, 2007).

A group of researchers interested in a smart parking system proposed the use of an optical sensor in a wireless sensor network (WSN) in order to detect a car coming into a deck as well as the direction the car is going (for the purpose of measuring where an open parking space could be found) (Chinrungrueng, Sunantachaikul, & Triamlumlerd, 2007). The issues found in this design lies within the detection ability of the sensors as they cannot reliably distinguish between automobiles and humans. Another issue could be the reliability of knowing where an open parking spot is located based solely on the direction of a car as it is very possible for cars to change direction at the immediate command of the driver.

The best method discovered in the efficiency of an intelligent parking system lies with the use of vehicular ad hoc networks (VANET). This type of wireless communication uses the networking capabilities between cars in order to tell a person trying to find a parking space the relevant locations using real-time updates with the added benefits of anti-theft protection and friendly parking information dissemination (Lu, Lin, Zhu, & Shen, 2009). While this method in test cases proved helpful, it is in the best interest for this design project to provide a relatively low-priced solution per parking spot in order to keep accurate data of the highest possible caliber. (Written by Julie Aichinger, Matthew McDade, and Laveréna Wienclaw.)

Marketing Requirements

1. This system should be able to accurately sense the presence of a vehicle.
2. This system should be applicable to universities in cities.
3. This system should be useful for utilization in parking decks.
4. This system should decrease travel time.
5. This system should have an intuitive user interface via a phone application.
6. This system should have displays that are easily interpretable and not distracting.
7. This system should have displays that are energy efficient.
8. This system should be able to present live data on displays and phone application.
9. This system should be able to operate under all weather conditions.
10. This system should have a long-lasting power supply.
11. This system should be easy to install, manage, and perform updates.

Objective Tree

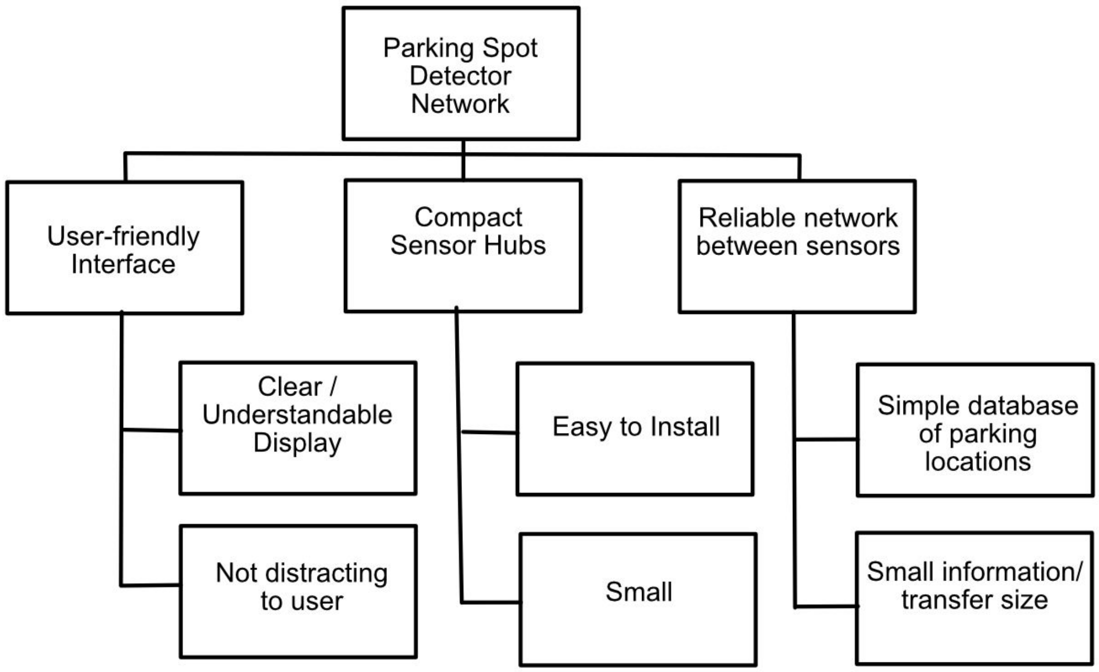

Figure 1: Objective Tree

The objective tree shows how the network connects to the sensor modules and the display interface, summarily breaking down the design of the project into individual parts. The key objectives that this design intends to cover are easily viewable from this chart, such as the modules being able to small and the display being clear. (Written by Ryne Turner.)

2. Design Requirements Specification

Marketing Requirements	Engineering Requirements	Justification
1, 3	Sensors respond to vehicles that are up to four meters away from module.	Sensors must be able to track both large and small vehicles that may be poorly centered between the lines of each parking space.
7, 10	Sensor networks update every half-minute to conserve battery power and maintain reliable data.	A good estimation of the changes in traffic movement would occur every half-minute, and with concern to the conservation of battery power, one sensor module would initiate every half-minute (at least during the daytime).
9	Module is water-resistant and survives under extreme weather conditions (-40 - 85 degrees Celsius).	Module must be able to survive changing weather patterns.
5, 6	Mobile applications must display a warning message to not use device while operating a vehicle. Project must obey traffic regulations.	Safety concerns must be acknowledged so that this application does not contribute in any way to unsafe driving practices.
3, 6	Display and mobile application should receive and update data simultaneously (within one half-minute) to prevent confusion.	Displays must be consistent with one another to prevent confusion.
8, 11	Parking spot modules should be separately addressable. The hub needs to recognize which module corresponds to which parking spot.	In order to distinguish different parking spots for displaying, the addresses of the modules must be unique.
2, 3	A battery system should last for at least four months per module.	If a battery-operated system does not last for many months, the product becomes less marketable in comparison to other designs on the market.
1, 8	A display signal to prove sensor is operating under working conditions	In order for maintenance to be simplistic, a sensor needs to be able to

	should be available to maintenance technicians.	send out a signal that validates proper functionality.
4	Wireless node network must be able to communicate across entire parking deck to central hub to accommodate users.	Nodes must transmit data sequentially across each other in the form of a node network.
2, 11	System should be easily scalable to larger parking decks of up to 65,535 parking spaces (16-bit addressing), which requires simplistic installation of new parking spot modules.	Plug-and-play functionality with minimal setup during installation is much more attractive for commercial applications, as well as a large number of spaces supported per deck.

Table 1: Engineering Design Requirements

Marketing Requirements

1. This system should be able to accurately sense the presence of a vehicle.
2. This system should be applicable to universities in cities.
3. This system should be useful for utilization in parking decks.
4. This system should decrease travel time.
5. This system should have an intuitive user interface via a phone application.
6. This system should have displays that are easily interpretable and not distracting.
7. This system should have displays that are energy efficient.
8. This system should be able to present live data on displays and phone application.
9. This system should able to operate under all weather conditions.
10. This system should have a long-lasting power supply.
11. This system should be easy to install, manage, and perform updates.

This table describes the engineering design requirements of this project. Each engineering requirement is a further explanation of the marketing requirements. The sensor module is designed in specificity in this table. (Written by Ryne Turner and Laveréna Wienclaw.)

3. Accepted Technical Design

Level 0 Block Diagram

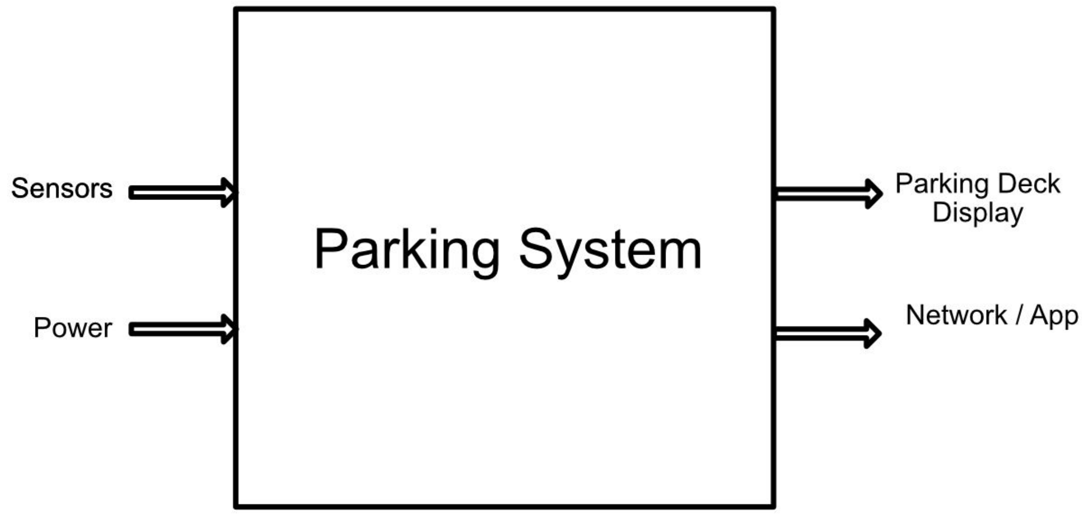

Figure 2: Level 0 Block Diagram of Hardware and Software
(Designed by Julie Aichinger.)

The Level 0 block diagram (Figure 2) shows the basic inputs and outputs of the Parking System as a whole. The entire system requires power in the form of battery packs and gets sensor data from laser proximity sensors. The system turns these inputs into the outputs of visualizing open and taken spots in a parking deck in the form of both a display outside of the parking deck displaying the number of open spots as well as a more detailed mobile app visualization. (Written by Matthew McDade.)

Module	**Parking System**
Designers	Julie Aichinger, Matthew McDade, Ryne Turner, and Laveréna Wienclaw
Inputs	- Sensors: data transmitted from the sensor modules - Power: the energy for the system
Outputs	- Network / App: the formatted sensor data to show user where there is available parking on a mobile applications - Parking deck display: the formatted sensor data to show user where there is available parking on a display within parking deck
Functionality	The system will use data collected from sensor modules and sent to WiFi accessible nodes that send data to a cloud server.

Table 2: Level 0 Block 1 Functional Requirement of Hardware
(Written by Ryne Turner.)

The Parking System describes the design of the entire project in its most basic and understandable form. The power applied to the modules and the activation of sensors are used to create the data that is then displayed both at the location of the parking facility and through the mobile applications. (Written by Ryne Turner.)

Level 1 Block Diagrams and Tables

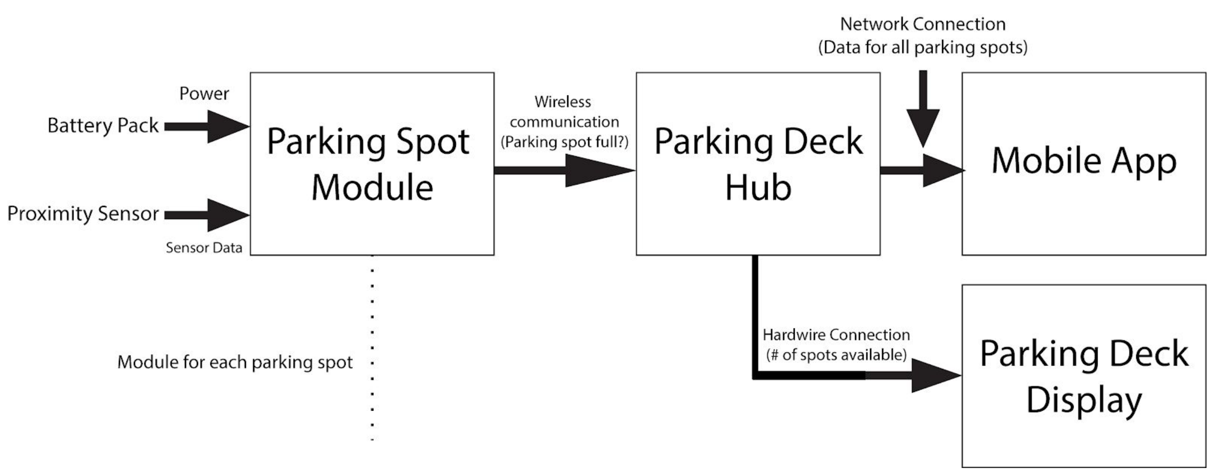

Figure 3: Level 1 Block Diagram of Hardware.
(Designed by Matthew McDade.)

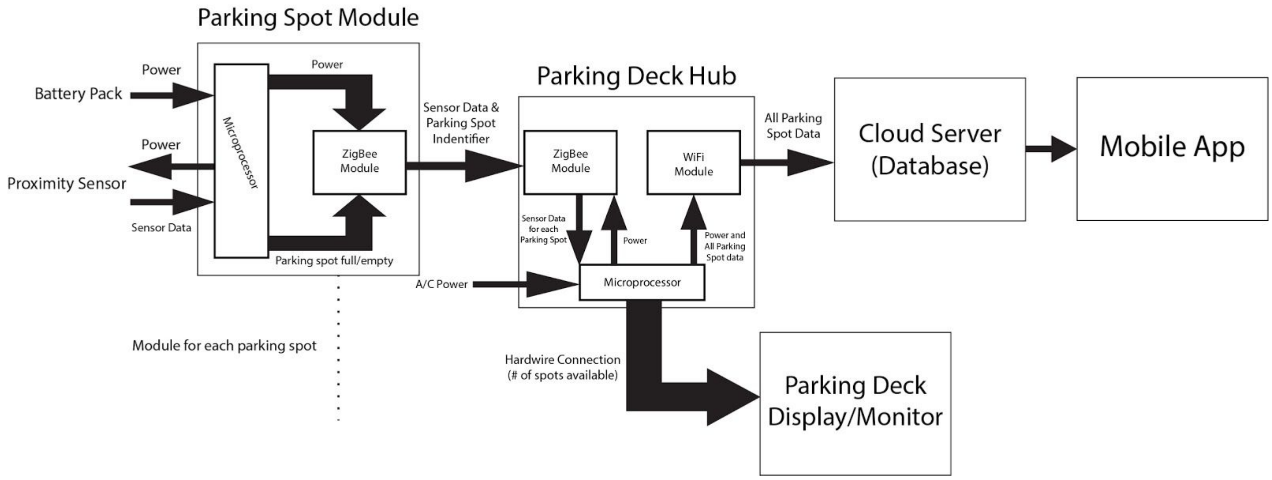

Figure 4: Level 2 Block Diagram of Hardware.
(Designed by Matthew McDade.)

This Level 1 hardware-specific block diagram (Figure 3) gives more depth on the individual modules that make up the parking system as a whole. It also highlights how each parking spot has its own parking spot "module," which all send their sensor data to the main parking deck "hub." This hub, which contains all parking spot data for an entire parking deck, can now output to a visual display outside the parking deck as well as the mobile app. In the Level 2 Block Diagram (Figure 4), a more detailed component view of the module and hubs are shown. Both contain a ZigBee wireless communication module for transmitting data from each parking spot module to the hub, and the hub contains a Wi-Fi module to send data for all parking spots to an off-site cloud server, which then serves data to the mobile app when requested. (Written by Matthew McDade.)

Module	**Parking Spot Module**
Designers	Ryne Turner and Laveréna Wienclaw
Inputs	- Proximity Sensor: Incoming network signal - Battery Pack: Power for the system
Outputs	- Wireless Communication: Ultrasonic or infrared signal and outgoing network signal for display, using Zigbee
Functionality	Activate signal once every half-minute and relay information back to parking deck hub. Data will be recorded and updated on the mobile apps and parking deck display accordingly.

Table 3: Level 1 Block Parking Spot Module Functional Requirement of Hardware (Written by Ryne Turner.)

The Parking Spot Module is the sensor hub that is able to detect whether a car is present using a proximity sensor. (Written by Laveréna Wienclaw.)

Module	**Parking Deck Hub**
Designers	Ryne Turner and Laveréna Wienclaw
Inputs	- Wireless Communication: the wireless data sent from the sensor modules in each parking space
Outputs	- Network Connection: data transmittance to a database that will hold the data for the Mobile App to then parse and display - Hardwire Connection: wired connection to the parking deck display that will be stationed at the entrance of each floor of a parking deck
Functionality	The parking hub will be the only connection to the Internet that all the hardware in the parking deck will have. This hub will transmit and receive data often for the use in the visual displayed in the parking deck and the mobile applications.

Table 4: Level 1 Block Parking Deck Hub Functional Requirement of Hardware (Written by Laveréna Wienclaw.)

The Parking Deck Hub receives sensor data from the modules placed at each parking spot by using ZigBee communication. The Parking Deck Hub is then able to send the proximity sensor data to an Internet-connected database. (Written by Laveréna Wienclaw.)

Module	**Mobile Application**
Designers	Julie Aichinger
Inputs	- Network Connection: data from database of parking hub data that details the state of the proximity sensors
Outputs	- None
Functionality	The mobile application will be the user's connection to the data output from the sensors. The visual data will help the user decide where they should travel first in order to find a parking space as quickly as possible.

Table 5: Level 1 Block Mobile App Functional Requirement of Hardware
(Written by Laveréna Wienclaw.)

Module	**Parking Deck Display**
Designers	Ryne Turner and Laveréna Wienclaw
Inputs	- Hardwire Connection: wired connection from parking hub data that transmits the data of the proximity sensors
Outputs	- None
Functionality	The mobile application will be the user's connection to the data output from the sensors. The visual data will help the user decide where they should travel first in order to find a parking space as quickly as possible.

Table 6: Level 1 Block Parking Deck Display Functional Requirement of Hardware
(Written by Laveréna Wienclaw.)

Module	**Microprocessor**
Designers	Ryne Turner and Laveréna Wienclaw
Inputs	- Power: voltage will be supplied to the microprocessor via a battery for the wireless power of the system. - Sensor Data: the data from the VL53L0CXV0DH/1, which will be stored in the memory of the microprocessor.
Outputs	- Power: voltage will be shared between the microprocessor and the rest of the circuit. - Sensor Data: from the microprocessor, the VL53L0CXV0DH/1 data will be transmitted through the ZigBee module
Functionality	The microprocessor of the sensor module will act as the brain of the operations necessary to receive intelligible proximity sensor data as well as be able to send it along to the ZigBee Module 1 in order to transmit it.

Table 7: Level 2 Block Microprocessor Functional Requirement of Hardware
(Written by Laveréna Wienclaw.)

Module	**ZigBee Module 1**
Designers	Ryne Turner and Laveréna Wienclaw
Inputs	- Power: voltage to power system - Sensor Data: the VL53L1X data about individual parking spots
Outputs	- Sensor Data & Parking Spot Identifier: the data of whether or not a car was in a space will be sent onto the Parking Deck Hub with unique identifiers for the number of spaces
Functionality	The ZigBee Module 1 serves as the communication between each spot in the parking deck, to the Hub that can connect all of this information to Wi-Fi where it can be stored for easier access for the display aspects of the design.

Table 8: Level 2 Block 2 Functional Requirement of Hardware
(Written by Laveréna Wienclaw.)

Module	**ZigBee Module 2**
Designers	Ryne Turner and Laveréna Wienclaw
Inputs	- Sensor Data & Parking Spot Identifier: the data unique to each parking space that has the information of whether a parking space if empty or not - Power: to power the system
Outputs	- Sensor data for each parking spot: this data is sent along in order to get the information to the Wi-Fi in order to store it in an easy to access place
Functionality	The ZigBee Module 2 acts as a receiver, as it will receive the individual sensor data from the different Parking Spot Modules and be able to send it along. It will tell these parking spots apart by looking at a unique 16-bit address sent by each parking spot module along with the current occupancy status of the spot.

Table 9: Level 2 Block ZigBee Module 1 Functional Requirement of Hardware
(Written by Laveréna Wienclaw.)

Module	**Wi-Fi Module**
Designers	Julie Aichinger, Matthew McDade, Ryne Turner, and Laveréna Wienclaw
Inputs	- Power: to power the system - All Parking Spot Data: the parking spot data that has come from the individual modules
Outputs	- All Parking Spot Data: the parking spot data is sent along to the cloud/database in order to make the data easier to access
Functionality	Connection to the internet, at realtime speeds, is crucial to the ability to display the parking spot data to the mobile app. The Wi-Fi module is the bridge from the hardware to the software as it sends the sensor data to a platform reachable for easy use.

Table 10: Level 2 Block Wi-Fi Module Functional Requirement of Hardware
(Written by Laveréna Wienclaw.)

Module	**Microprocessor 2**
Designers	Ryne Turner and Laveréna Wienclaw
Inputs	- A/C Power: to power the system - Sensor data for each parking spot: the data from the Modules will be controlled and sent out using the program on the microprocessor
Outputs	- Power (ZigBee Module): power connections from the microprocessor to the ZigBee module - Power (Wi-Fi Module): power connections from the microprocessor to the Wi-Fi module - All Parking Spot Data: the data passed through this block and sent to the Wi-Fi module
Functionality	The Parking Deck Hub's need a brain to control the flow of information and keep timing. The microprocessor keeps time and is able to send data between the different communicative modules.

Table 11: Level 2 Block Microprocessor 2 Functional Requirement of Hardware (Written by Laveréna Wienclaw.)

Module	**Parking Deck Display / Monitor**
Designers	Ryne Turner and Laveréna Wienclaw
Inputs	- Hardwire Connection: hardwire connection that will transmit the data from the Parking Deck Hub to the live Display within the parking deck
Outputs	- None
Functionality	As users looks for places to park, rather than pulling out their phone for the app, they can use the Parking Deck Displays within the deck for easier access of the real time data.

Table 12: Level 2 Block Parking Deck Display / Monitor Functional Requirement of Hardware (Written by Laveréna Wienclaw.)

Module	**Cloud Server (Database)**
Designers	Julie Aichinger and Matthew McDade
Inputs	- All Parking Spot Data: the data of whether the parking spots are empty or not
Outputs	- Mobile App: the data stored in the database can be accessed for the use in the mobile application likely using an HTTP request
Functionality	The Database will hold all the live data from the parking deck sensors and hold it for the use in the Mobile application.

Table 13: Level 2 Block Cloud Server (Database) Functional Requirement of Hardware (Written by Laveréna Wienclaw.)

Module	**Mobile Application**
Designers	Julie Aichinger and Matthew McDade
Inputs	- Cloud Server (Database): this live data will be requested from the app in order to display the current sensor data of the Parking Spot Modules.
Outputs	- None
Functionality	Most users will want to check where there are available spots before they enter a parking deck, so the mobile app fills this need by displaying what spots are available on mobile devices rather than just the displays within the parking decks.

Table 14: Level 2 Block Mobile App Functional Requirement of Hardware (Written by Laveréna Wienclaw.)

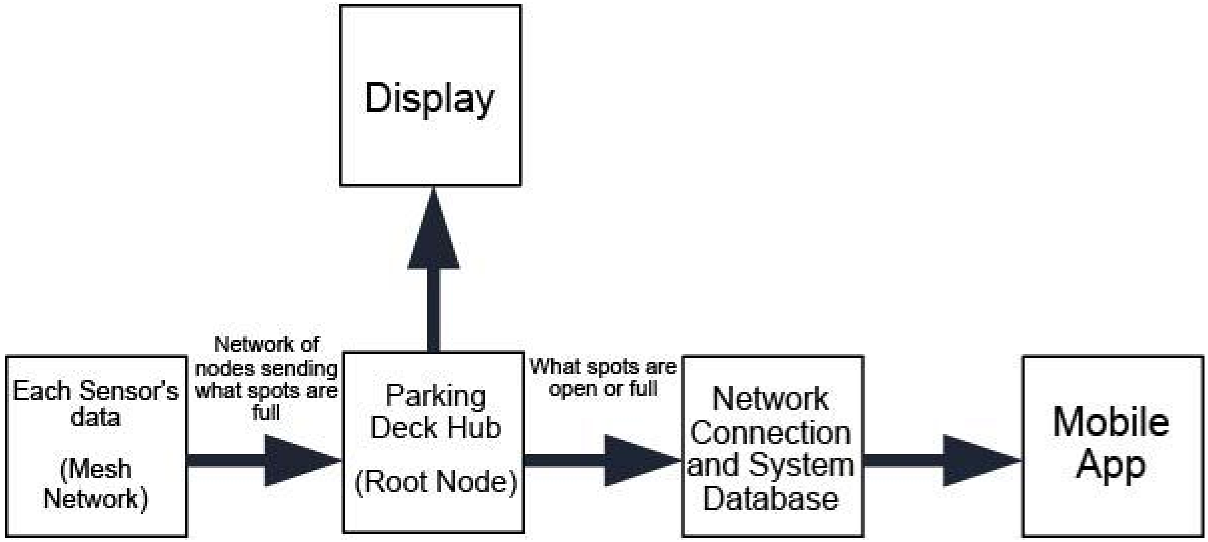

Figure 5: Level 1 Block Diagram of Software
(Designed by Julie Aichinger.)

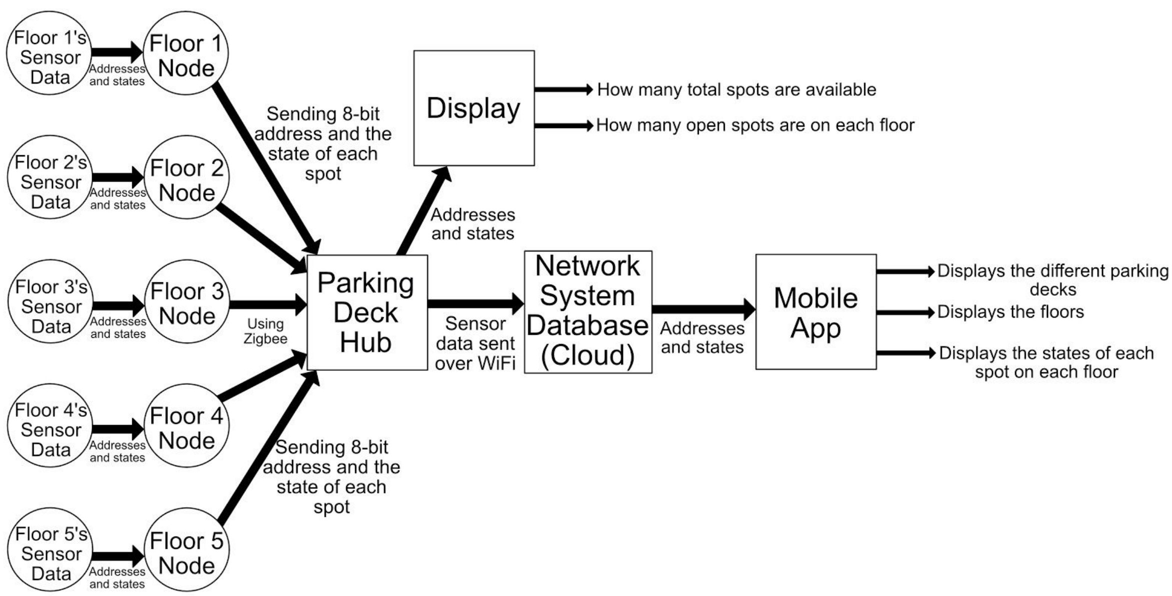

Figure 6: Level 2 Block Diagram of Software
(Designed by Julie Aichinger)

Going slightly more in depth with the software design, this Level 1 software block diagram (Figure 5) highlights the design choice of putting the parking space sensors into a mesh network. This way each transmitter only needs to transmit to the next closest parking spot module (in a chain eventually reaching the hub), rather than using more power to transmit all the way to the hub from every parking spot module. Similar to the hardware block diagram, the hub will send a video signal to a display outside the parking deck, and eventually to a mobile app. Specifically, the hub will connect with a cloud database that the phone pulls its information from. The Level 2 software block diagram (Figure 6) displays a more in-depth look at how each floors' sensors send data to the main parking deck hub. Since there may be obstacles or thick walls between floors, a node placed near the entrance/exit to another floor will capture all parking spot modules' data from a floor, then relay it to the main parking deck hub. (Written by Matthew McDade.)

Module	**Sensor Data (Mesh Network)**
Designers	Matthew McDade and Julie Aichinger
Inputs	- Mesh network - Data from sensors
Outputs	- Outgoing network signal for display - Length of time vehicle has been present
Functionality	Manage and update parking lot data to create efficient way of communicating between parking spot modules. A link between devices will be created to notify central hub in a more sensible and even pattern.

Table 15: Level 1 Block Sensor Data (Mesh Network) Functional Requirement of Software (Written by Ryne Turner.)

Module	**Parking Deck Hub (Root Node)**
Designers	Julie Aichinger and Matthew McDade
Inputs	- Network of Nodes: send data on which spots are empty or not
Outputs	- Display: displays in parking deck - Sensor data: what spots are empty or not, sent to Database
Functionality	The Deck Hub provides the intermediary service of communicating between the hardware of each space, and then sending this information to a Database that is accessible to the application.

Table 16: Level 1 Block Parking Deck Hub (Root Node) Functional Requirement of Software (Written by Laveréna Wienclaw.)

Module	**Display**
Designers	Ryne Turner and Laveréna Wienclaw
Inputs	- Sensor data: the sensor data will be sent from the Hub with a hardwire connection
Outputs	- None
Functionality	A visualization for users to see which spots are empty so that they can easily locate an open spot.

Table 17: Level 1 Block Display Functional Requirement of Software (Written by Laveréna Wienclaw.)

Module	**Network Connection and System Database**
Designers	Julie Aichinger and Matthew McDade
Inputs	- Data of Parking Spots: the individual identification of the sensors along with their information of whether their parking spot is empty or not
Outputs	- Mobile App: the connection of communication between database and the mobile application
Functionality	The Database serves as the connection between the hardware in each parking space, to the use in the mobile application.

Table 18: Level 1 Block Network Connection and System Database Functional Requirement of Software (Written by Laveréna Wienclaw.)

Module	**Mobile App**
Designers	Julie Aichinger and Matthew McDade
Inputs	- Database: the Database will hold data about the sensors that the mobile application can take and use to display a helpful graphic for users.
Outputs	- None
Functionality	Provides easy accessibility to the information about the available parking spots from the comfort of a mobile device.

Table 19: Level 1 Block Mobile App Functional Requirement of Software (Written by Laveréna Wienclaw.)

Module	**Parking Deck Hub**
Designers	Julie Aichinger and Matthew McDade
Inputs	- Floor 1-5 Node: the parking deck hubs from each floor of a parking deck
Outputs	- Addresses and States: the individualized parking space data and whether the spots are empty or not - Sensor data: the VL53L0CXV0DH/1 data transmitted over Wi-Fi
Functionality	Holds data transmitted from the Parking Spot Modules in each parking space.

Table 20: Level 2 Block Parking Deck Hub Functional Requirement of Software
(Written by Laveréna Wienclaw.)

Module	**Display**
Designers	Ryne Turner and Laveréna Wienclaw
Inputs	- Addresses and States: data sent from the sensors
Outputs	- Number of Spots Available: displays the number of spots available in the entire parking deck - Open spots on each floor: the number of parking spots available on each floor
Functionality	Displays useful parking data in the parking decks for users.

Table 21: Level 2 Block Display Functional Requirement of Software
(Written by Laveréna Wienclaw.)

Module	**Network System Database (Cloud)**
Designers	Julie Aichinger and Matthew McDade
Inputs	- Sensor Data: includes the individual ID of the sensor, and the state of whether the spot is empty or not
Outputs	- Addresses and States: the ID's of the sensors and the stat of whether the spot is empty or not
Functionality	Holds the sensor data in an easy to reach location for access from the Mobile Application.

Table 22: Level 2 Block Network System Database (Cloud) Functional Requirement of Software (Written by Laveréna Wienclaw.)

Module	**Mobile Application**
Designers	Julie Aichinger and Matthew McDade
Inputs	- Addresses and States: sensor data of individual parking modules and their state of the spot being empty or not
Outputs	- Display of different parking decks: a menu of different parking decks - Floors of each deck: a menu choice between the floors of each deck - States of parking spots on each floor: whether a spot is empty or not, a helpful graphic will be used
Functionality	The mobile application will be for the user to see the number of spaces available in a deck before having the enter the deck.

Table 23: Level 2 Block Mobile App Functional Requirement of Software (Written by Laveréna Wienclaw.)

Hardware modules designed by Ryne Turner and Laveréna Wienclaw.

Software modules designed by Matt McDade and Julie Aichinger.

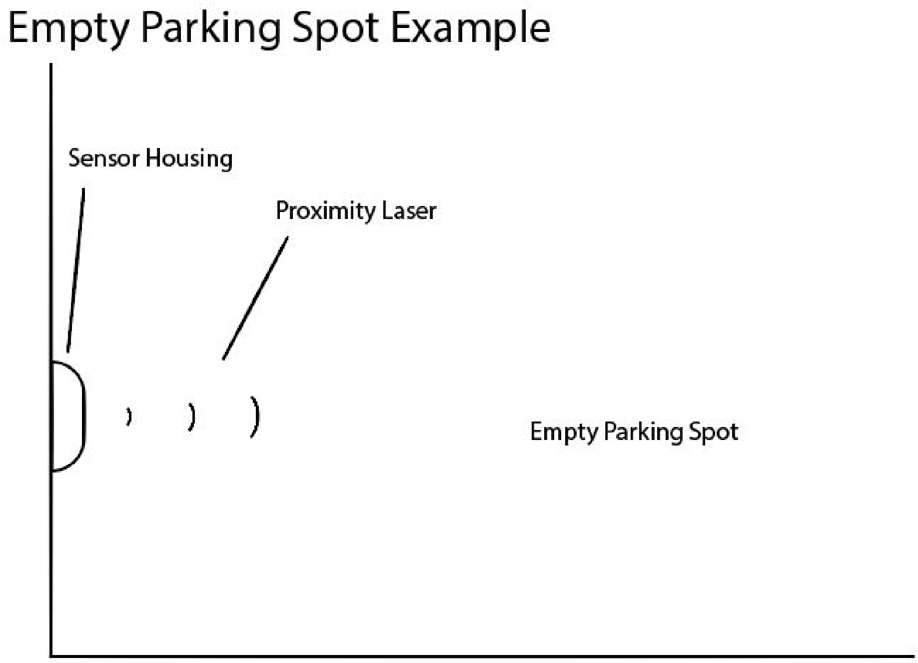

Figure 7: Mechanical Sketch of Empty Parking Spot

This figure of an empty parking spot depicts the proximity laser not bouncing off any object since there is no car parked in the parking spot. This parking spot module would then transmit an "empty" state along with its unique 16-bit address to tell the hub that a car is not present in this parking spot.

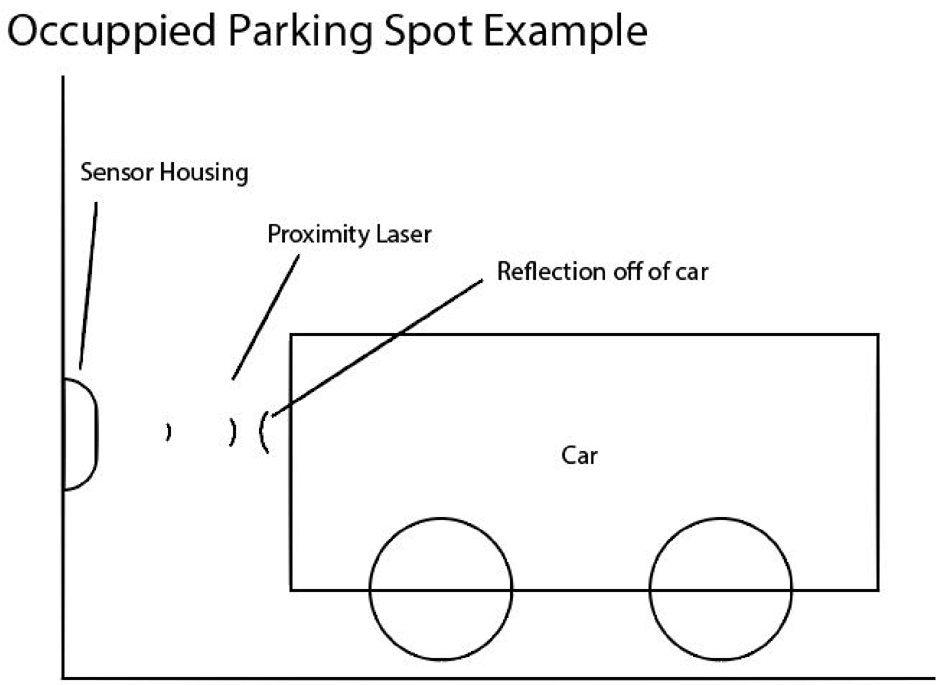

Figure 8: Mechanical Sketch of Occupied Parking Spot

Now a car is present in the parking spot, and the proximity laser reflects back into the sensor housing, letting the module know that an object, most likely a car, is parked in the parking spot. The module then transmits the taken parking spot state along with its parking spot address to the parking lot hub.

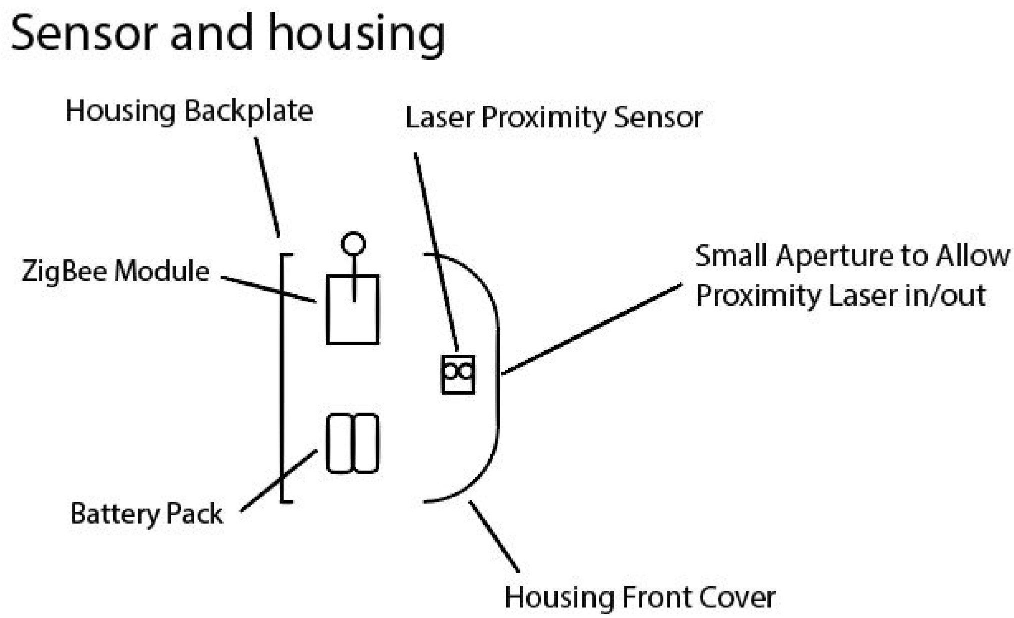

Figure 9: Mechanical Sketch of Sensor and Housing

This diagram of the parking spot module depicts all the major components housed within each module. The ZigBee module for communicating and receiving data over the mesh network, a battery pack, and the laser proximity sensor.

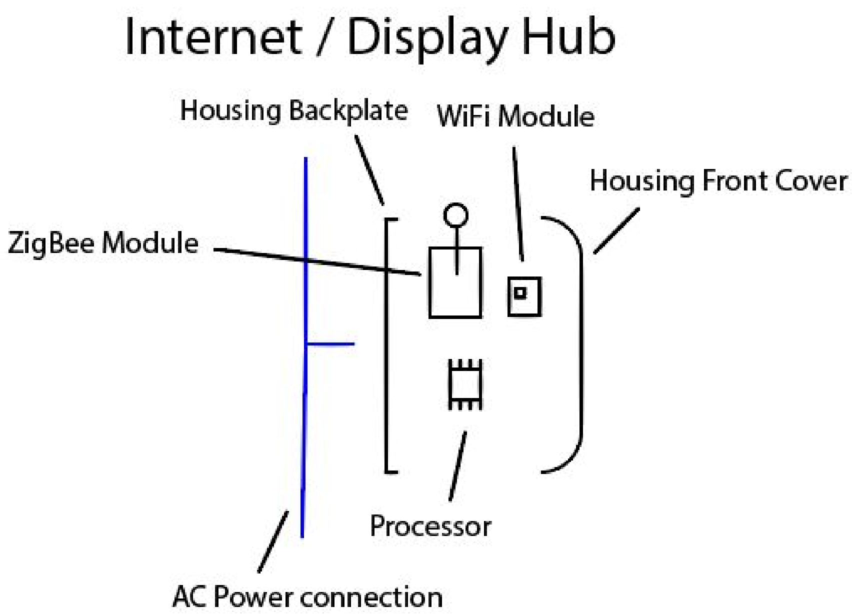

Figure 10: Mechanical Sketch of Internet/Display Hub

This sketch depicts the central hub in each parking deck, which receives all parking spot data via the ZigBee module (wireless standard used by all other modules), and communicates that data to our internet cloud database via the Wi-Fi module. A processor is also listed to aid in the calculation and displaying of information on the parking lot display, showing how many spots are open in the deck.

Design Calculations are described in detail below. These calculations reference the mechanical sketches in Figures 7-10. (Written by Matthew McDade and Ryne Turner.)

Task: Design a smart parking system that will connect individual modules together in a node network using the Zigbee communication protocol.

Given:

-Using VL53L0CXV0DH/1 Time-of-Flight Distance Sensor Carrier with Voltage Regulator

- Using ATMEGA328-AURCT-ND

- Using ATSAMR21B18-MZ210PA

Assumptions:

- Sensors will turn on once every 30 seconds during the daytime
- Each module will use 2200 mAh of electric charge (approximately two AA batteries)
- Implementation does not include any additional power sources such as a solar panel

Electrical Component Design:

In order to make the calculations more presentable, actual parts are used to obfuscate the chances of having invalid or unrealistic data. Other determinants taken into consideration are referenced in the Appendix of this report. As characterized by the mechanical sketch of an occupied parking spot in Figure 8, the vehicle must be within parked 4 meters of the sensor housing module as per the design requirements.

As an example of a potential sensor a module might use, the VL53L0CXV0DH/1 may be applied to sense the presence of a vehicle within 4 meters. The maximum allowed power of this device is 40 mA. The ATMEGA328-AURCT-ND includes an 20 MHz resonator and a voltage regulator allowing for 1.8 V to 5.5 V supplies (within the estimated voltage limit required). The Zigbee communication device transmits data using 40 mA of current and receives data using 15 mA of current.

During each module activate:

Total Estimated Percentage Time for Power Draw = 2 seconds per 30-second interval

$$\frac{2\ \text{s}}{30\ \text{s}} = 0.067 \approx 6.7\%$$

This means that power is only being used a little less than 7 percent of the time.

Total Estimated Power = 2 (40 mA) + 0.24 mA = 80.24 mA during sensor operation

Assuming a rounded-off value for two AA batteries is used (2200 mAh):

$$\frac{2200\ \text{mAh}}{80.24\ \text{mA}} = 27.418\ \text{hours with}\ 80.24\ \text{mA draw}$$

Because one half-minute is only 1/120 of an hour, this total amount of time activated is minuscule in relation to the time the modules are deactivated.

$$120\ (27.418\ \text{hours}) = 3290\ \text{hours} \approx 120\ \text{days}\ (4\ \text{months})$$

This means that two AA batteries would need to be replaced per module every four months. This meets one of the design requirements by having a long-lasting power supply. Battery life could also be extended using a small solar panel for each module, which may be integrated into the circuit design in the future.

Using the ATSAMR21B18-MZ210PA Zigbee module for communication provides this system with the advantage of working up to 1200 meters (as a network whole, with each module being able to physically transmit up to 20 meters). Each module would be able to communicate with another module throughout almost any parking deck. The digital input/output bit number is 15 (of 0-31 total), which allows for plenty of addressing possibilities across multiple decks. The current plan is to manually assign bit addresses to each parking space, though the assignment of module bit addresses might be updated in the future. The ATSAMR21B18-MZ210PA module uses the standard Zigbee transmission rate of 256 kbit/s. This can be approximated as follows (which is far less that the data being sent from node to node, meeting the requirements of the data rate transmission):

$$\left(\frac{1000\ \text{K}}{16-\text{ary symbol}}\right)\left(4\frac{\text{bits}}{\text{symbol}}\right) = 250\ \text{Kbps}$$

Pseudocode is included in the following lines. Comments are specified with two forward slashes. In the end, code on the parking spot module and parking deck hub will be running a variant of C code. The mobile application will be made using React Native libraries to compile natively to iOS and Android with one codebase. The Zigbee modules handle 16-bit addressing, which poses many advantages including meeting Design Requirement 10, which mandates the system be scalable up to 65,535 ($2^{16}-1$) individually addressable modules. Each module will be assigned a 16-bit addressing scheme as denoted in Table 9. This means that for every scan of the system that is performed (scan meaning activation for certain modules once every thirty seconds), a max of 65,535 parking spot modules will be pinged and have their data pulled. Using an approximate data rate of 250 Kbps (an IEEE 802.14.4-based specification), a maximum of 12,500 20-bit packets of data (16-bit address + 4-bit parking spot occupancy data, this is a worst-case scenario) can be sent along per second and transferred to the module, so the system may be able to handle many more than the target goal, but due to other requirements dealing with the maximum number of spots wanted to be supported, the target goal of 65,535 was decided upon (Written by Matthew McDade and Ryne Turner.)

Code for Parking Spot Module

```
// uses zigbee API mode which handles timing of
// multiple packets trying to be sent at the same time, and also sends formed
// data packets to specific addresses each time

address = unique_16_Bit_Address;
destination = destinationAddress;

// main function is called once initially, then on an interval
function loop()
	// come out of low power mode
	exitLowPowerMode()
	fullOrNot = isSpaceFull()
	sendDataPacket(fullOrNot)

	// run either every 15 minutes or every 30 seconds
	if getTime() is after midnight && getTime() is before 6am
		setInterval(loop, 15 minutes)
	else
		setInterval(loop, 0.5 minutes)

	// enter low power mode when not doing anything
	enterLowPowerMode()

// check proximity sensor data and return if space is full or not
function isSpaceFull()
	enableProxSensor()
	distance = getSensorDistance()
	disableProxSensor()
	if distance < 2 meters
		return true
	return false

// send data packet to hard-coded receiver with data fullOrNot
function sendDataPacket(fullOrNot)
	dataPacket = {destination, address, fullOrNot}
	enableZigbee()
	while (!receiverReady) {}
	zigbeeTransmit(dataPacket)
	disableZigbee()
```

Code for Parking Deck Hub

```
localData = (hash table of spots and whether they're full or not)

// run this whenever data has been received
function onPacketReceive(incomingData)
       forwardToInternetDatabase(incomingData)
       spotAddress = incomingData.address
       data = incomingData.fullOrNot
       localData[address] = data
       displaySpots()

function displaySpots()
       // creates text to output based on localData

//Code to handle collisions
//A new scan is performed when timestamp of addresses is the same
handleCollisions()
       if(a.currentTimeMillis() == b.currentTimeMillis()) {
                WakeUpModule()
                sendDataPacket(fullOrNot) }
```

According to the IEEE 802.15.4 standard, Zigbee handles data collision using carrier sense multiple access with collision avoidance (CSMA/CA). Scanning whether the network is turned on a specified device in conflict occurs, a shared resource is classified as taken, and a new transmission is made. This will require waking up the module, but a new timestamp will be assigned to the modules in conflict, thus resolving the issue. (Written by Ryne Turner.)

Code for Mobile Application

```
onAppOpen()
       getDataFromDatabase()
       showDeckListView()

onParkingDeckSelect()
       showSingleDeckView()

onRefresh()
       getDataFromDatabase()
```

Mock-Up for Mobile Application

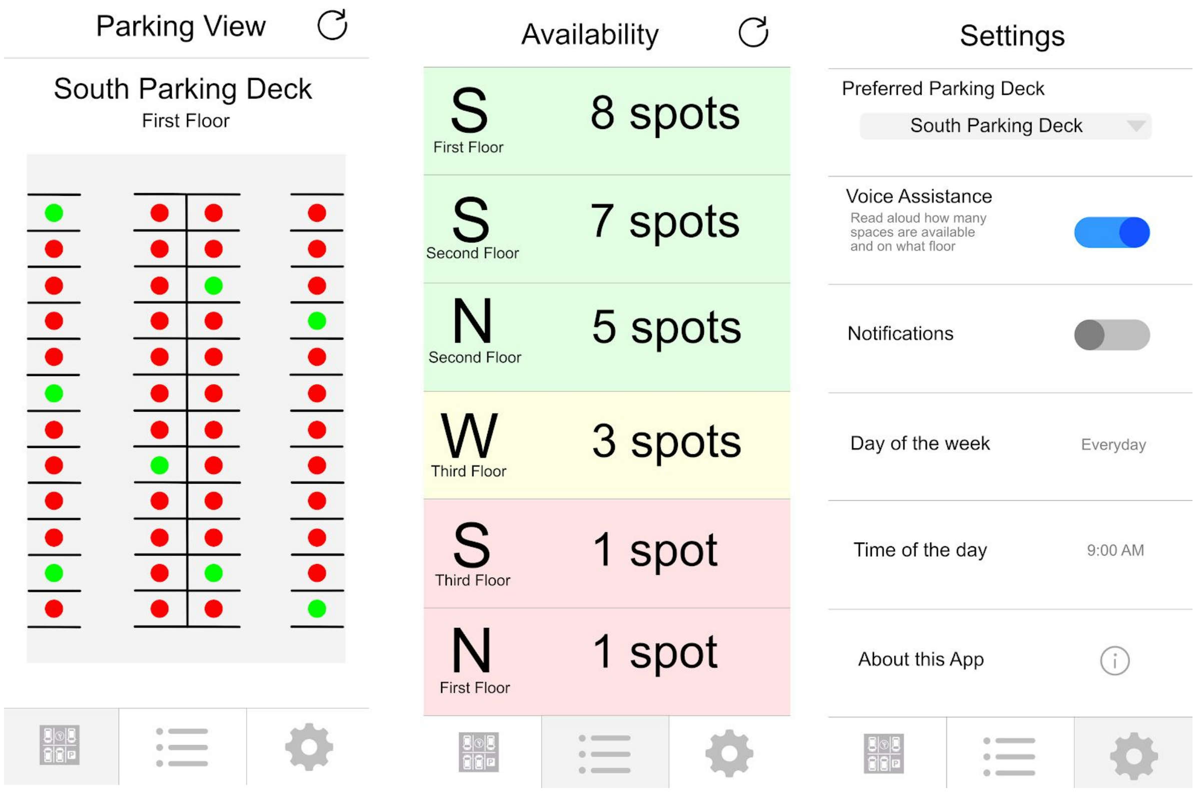

Figure 11: Mock-up Design of the Mobile Application
(Designed by Julie Aichinger.)

Eagle Schematics

The following Figures 12 and 13 are the two designed portions of the intended microprocessor board that will be used in each parking spot node in order to direct signals to the Zigbee modules (Figure 16) about the data read from the time-of-flight sensors (Figure 15). In Figure 12, the Low-Dropout Voltage Regulator shows a popular combination of capacitors, a transistor, and an LED that regulates the voltage to 3.3 V.

Figure 14 shows the other portion of the Eagle schematic that displays the main connections to the ATMEGA328 processor. Most notably, there is the crystal oscillator to keep the timing of the programs running on the microprocessor, the switch that is used to disconnect and connect power when the choice is necessary (for installation and testing). There are several pins that are left unrouted. We intend to build a board with through-hole solder points with via connections to these floating pins in order to test and program in the future. For now, they do not serve a specific or intentional purpose. Figure 17 shows an overall picture of the schematic that includes all the hardware and connections necessary in each node. (Written by Laveréna Wienclaw.)

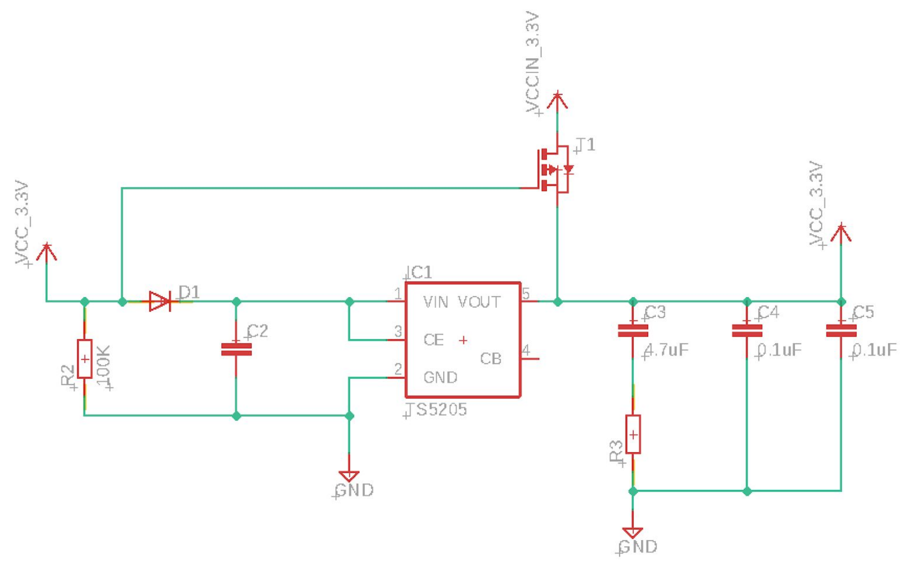

Figure 12: Low-Dropout Voltage Regulator Eagle Schematic
(Designed by Laveréna Wienclaw.)

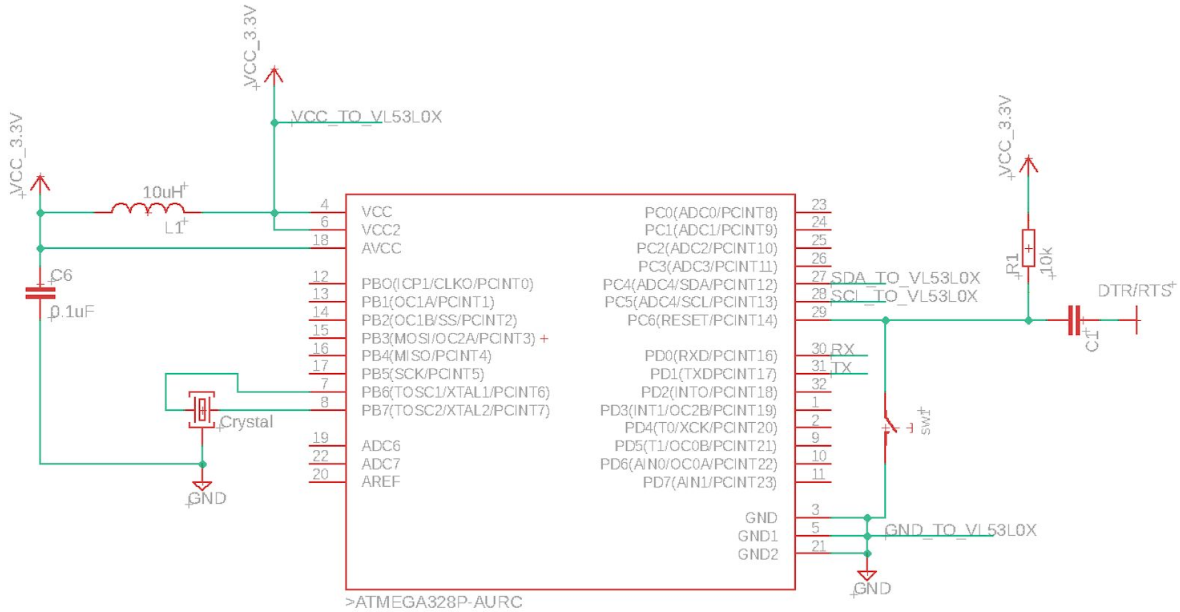

Figure 13: Microprocessor Eagle Schematic
(Designed by Laveréna Wienclaw.)

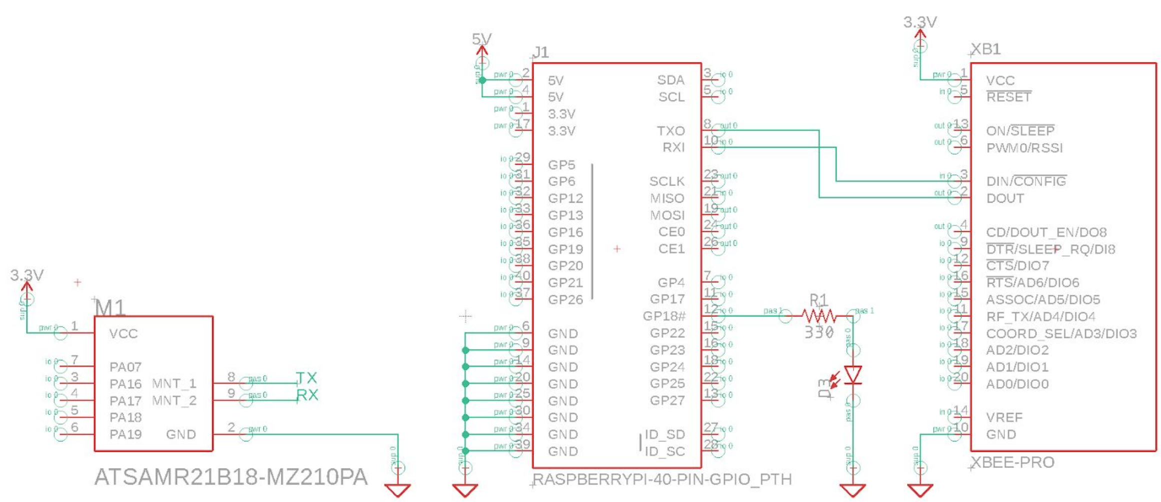

Figure 14: Hub Eagle Schematic (Including Zigbee Connections)
(Designed by Ryne Turner.)

In Figure 14 a Raspberry Pi 40-pin layout is connected to the central Zigbee Pro module that handles all of the incoming messages from each individual parking module, the Zigbee component of which is pictured on the left-hand side of the figure. Using direct serial communication, the Zigbee Pro module receives all of the incoming parking space data in connection with the corresponding addresses of each parking module, transferring that data directly to the Raspberry Pi, of which the purpose is a Wi-Fi connection routing to the mobile applications to provide the most current set of data. The Raspberry Pi will also push this data to the LED screen, which is represented in the circuit diagram by a simple LED component.

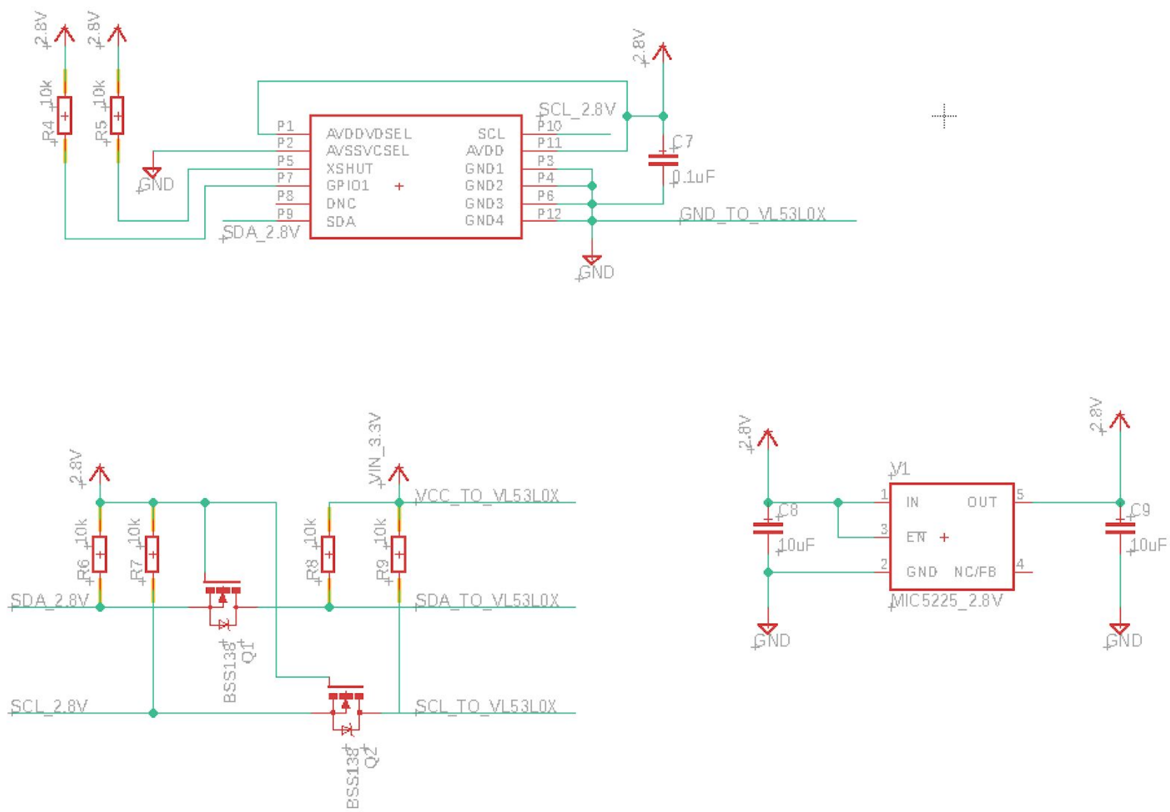

Figure 15: Time-of-Flight Sensor Eagle Schematic
(Designed by Laveréna Wienclaw.)

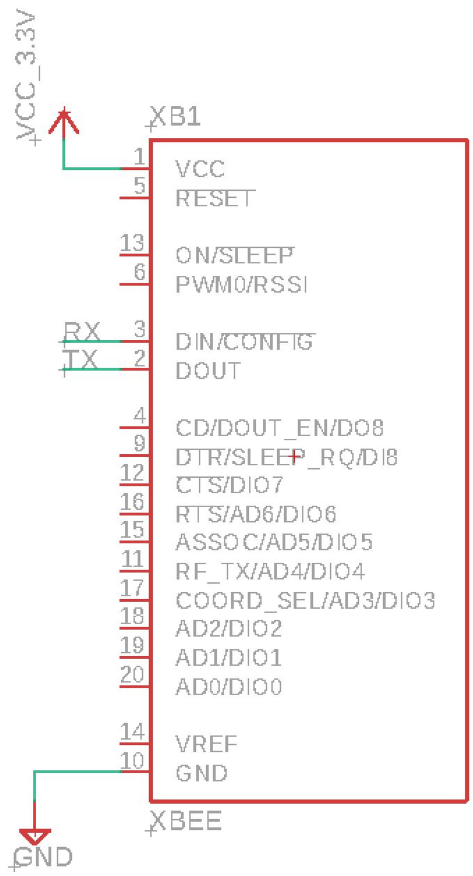

Figure 16: Module Zigbee Connection Eagle Schematic
(Designed by Laveréna Wienclaw.)

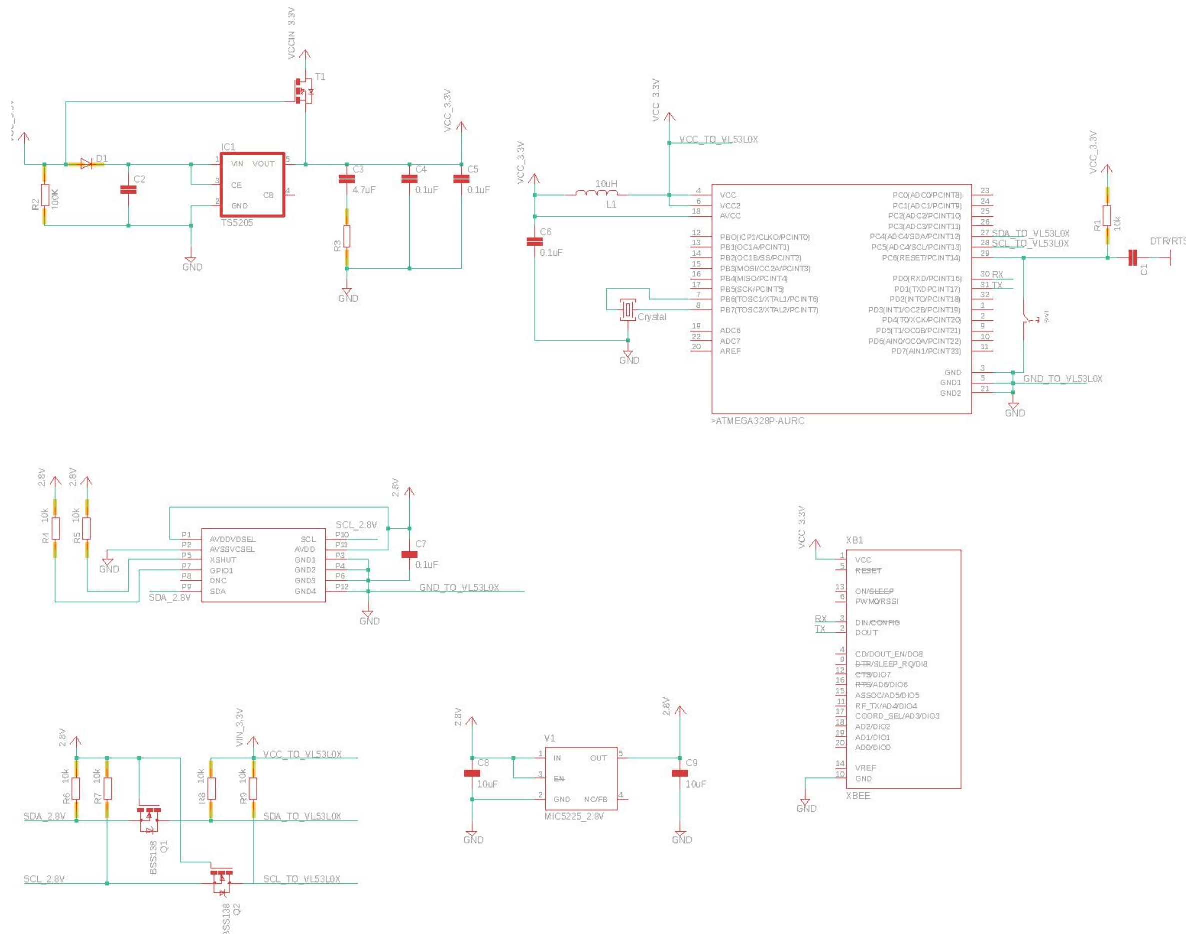

Figure 17: Overview of Module Eagle Schematic
(Designed by Laveréna Wienclaw.)

4. Parts Lists

Parking Spot Module Parts List				
Part	**Quantity**	**Reference Designators**	**Part Number**	**Description**
ATSAMR21B18-MZ210PA	10	ATSAMR21B18-MZ210PA	ATSAMR21B18-MZ210PA	Communication transceiver from module to module
ATMEGA328-AURC	10	ATMEGA328-AURC	ATMEGA328-AURCT-ND	Processes incoming and outgoing data
BATT	10	BATT	534-2463CN	Power source for module
VL53L0X	10	VL53L0X	VL53L0CXV0DH/1	Detects presence of vehicle
TS5205	10	TS5205	726-TLS810D1EJV33XUM	Regulates voltage on microprocessor to 3.3V
XBEE-PRO	1	XBEE-PRO	XBP24CZ7WIT-004	Handles central hub Zigbee communication

Table 24: Parts List Table
(Written by Matthew McDade, Ryne Turner, and Laveréna Wienclaw.)

Parking Spot Module Materials Budget List					
Part	**Quantity**	**Part Number**	**Unit Cost**	**Total Cost**	**Description**
ATSAMR21B18-MZ210PA	10	ATSAMR21B18-MZ210PA	$10.72	$107.20	1 per 10 modules
ATMEGA328-AURC	10	ATMEGA328-AURCT-ND	$1.96	$19.60	1 per 10 modules
BATT	10	534-2463CN	$2.00	$20.00	1 per 10 modules
VL53L0X	10	VL53L0CXV0DH/1	$9.15	$91.50	1 per 10 modules
XBEE-PRO	1	XBP24CZ7WIT-004	$28.50	$28.50	1 per hub

Table 25: Materials Budget List Table
(Written by Matthew McDade, Ryne Turner, and Laveréna Wienclaw.)

Tables for Bill of Materials:

Part	Value	Device	Package	Description
D3		LED3MM	LED_3MM	LED (Generic)
J1	RASPBERRYPI-40-PIN-GPIO_PTH	RASPBERRYPI-40-PIN-GPIO_PTH	2X20_SHROUDED	Raspberry Pi GPIO Header
M1	MICROCHIP_ATSAMR21B18-MZ210PAMICROCHIP_ATSAMR21B18-MZ210PA_0_0	MICROCHIP_ATSAMR21B18-MZ210PAMICROCHIP_ATSAMR21B18-MZ210PA_0_0	MICROCHIP_ATSAMR21B18-MZ210PA_0	ZigBee/802.15.4 Modules 48MHz SMT
R1	330	330OHM-0603-1/10W-1%	0603	330Î© resistor
XB1	XBEE-PRO	XBEE-PRO	XBEE-PRO	XBee (TM) /XBee-PRO(TM) OEM RF Modules

Table 26: Bill of Materials Part 1
(Written by Ryne Turner.)

Part	Value	Device	Package	Description
C1,C4, C5, C6, C7	0.1uF	Capacitor	603	Capacitor
C2	1uF	Capacitor	603	Capacitor
C8,C9	10uF	Capacitor	603	Capacitor
C3	4.7uF	Capacitor	603	Capacitor
D1	--	LED3MM	LED_3MM	LED(Generic)
L1	10uH	L0603B100MDWFT	603	Inductor
IC1	TS5205 3.3V	TS5205CX533	SOT-25	LDO Voltage Regulator
Q1,Q2	--	BSS138	SOT-23	N-Channel
R1,R4,R5,R6,R7,R8,R9	10K	Resistor	603	Resistor
R2	100K	Resistor	603	Resistor
R3	1	Resistor	603	Resistor
SW1	--	TL6330AF200Q	--	E-Switch
T1	--	621-DMP1045UQ-7	SOT-23-3	P-channel
U1	--	ATMEGA328	TQFP32-08	
V1	MIC5225_2.8V	MIC5225	SOT23-5	Low dropout regulator
SENSOR	--	VL53L0X	Optical LGA12	TOF sensor
XB1	--	XBEE	XBEE	XBEE
Y1	--	8M10AHC49T	HC49U	Crystal

Table 27: Bill of Materials Part II
(Written by Laveréna Wienclaw.)

The above tables show the different parts necessary to implement the hardware portion of the project. Most notably is Table 25 that mentions the budget portion of the project. The most expensive parts are addressed here. With the knowledge that we may be switching out different smaller components given the availability on what remains extra in the lab, we do not feel an over-detailed budget is necessary at this time.

5. Project Schedules (Gantt Chart Design)

Task Name	Duration	Start	Finish	Resource Names
SDP1 Fall 2018				
Project Design				
Preliminary report	11 days	Thu 9/6/18	Sun 9/16/18	**All**
Cover page	11 days	Thu 9/6/18	Sun 9/16/18	Ryne Turner
T of C, L of T, L of F	11 days	Thu 9/6/18	Sun 9/16/18	Ryne Turner
Need	11 days	Thu 9/6/18	Sun 9/16/18	Ryne Turner
Objective	11 days	Thu 9/6/18	Sun 9/16/18	All
Background	11 days	Thu 9/6/18	Sun 9/16/18	All
Marketing Requirements	11 days	Thu 9/6/18	Sun 9/16/18	Laveréna Wienclaw
Objective Tree	11 days	Thu 9/6/18	Sun 9/16/18	Laveréna Wienclaw
Block Diagrams Level 0, 1, ... w/ FR tables	**11 days**	Thu 9/6/18	Sun 9/16/18	
Hardware modules (identify designer)	11 days	Thu 9/6/18	Sun 9/16/18	Julie Aichinger
Software modules (identify designer)	11 days	Thu 9/6/18	Sun 9/16/18	Matthew McDade
Mechanical Sketch	11 days	Thu 9/6/18	Sun 9/16/18	Matthew McDade
Team information	11 days	Thu 9/6/18	Sun 9/16/18	Ryne Turner
References	11 days	Thu 9/6/18	Sun 9/16/18	Julie Aichinger
Preliminary Parts Request Form	11 days	Thu 9/6/18	Sun 9/16/18	Matthew McDade
Midterm Report	**35 days**	**Thu 9/6/18**	**Wed 10/10/18**	**All**
Design Requirements Specification	14 days	Mon 9/17/18	Sun 9/30/18	Ryne Turner
Midterm Design Gantt Chart	14 days	Mon 9/17/18	Sun 9/30/18	Julie Aichinger
Design Calculations	**24 days**	**Mon 9/17/18**	**Wed 10/10/18**	**Matthew McDade**
Electrical Calculations	**24 days**	**Mon 9/17/18**	**Wed 10/10/18**	**Ryne Turner**
Communication	24 days	Mon 9/17/18	Wed 10/10/18	Ryne Turner
Computing	24 days	Mon 9/17/18	Wed 10/10/18	Matthew McDade
Control Systems	24 days	Mon 9/17/18	Wed 10/10/18	Matthew McDade
Power, Voltage, Current	24 days	Mon 9/17/18	Wed 10/10/18	Matthew McDade
Mechanical Calculations	**24 days**	**Mon 9/17/18**	**Wed 10/10/18**	**Ryne Turner**
Structual Considerations	24 days	Mon 9/17/18	Wed 10/10/18	Ryne Turner
System Dynamics	24 days	Mon 9/17/18	Wed 10/10/18	Ryne Turner
Block Diagrams Level 2 w/ FR tables & ToO	**7 days**	**Mon 9/17/18**	**Sun 9/23/18**	
Hardware modules (identify designer)	7 days	Mon 9/17/18	Sun 9/23/18	Julie Aichinger
Software modules (identify designer)	7 days	Mon 9/17/18	Sun 9/23/18	Matthew McDade
Block Diagrams Level 3 w/ FR tables & ToO	**7 days**	**Mon 9/24/18**	**Sun 9/30/18**	
Block Diagrams Level N+1 w/ FR tables & ToO	**10 days**	**Mon 10/1/18**	**Wed 10/10/18**	
Midterm Design Presentations Part 1	1 day	Thu 10/11/18	Thu 10/11/18	All
Midterm Design Presentations Part 2	1 day	Thu 10/18/18	Thu 10/18/18	All
Project Poster	14 days	Mon 10/8/18	Sun 10/21/18	Matthew McDade
Secondary Parts Request Form	21 days	Mon 9/17/18	Sun 10/7/18	Ryne Turner
Final Design Report	**52 days**	**Mon 10/8/18**	**Wed 11/28/18**	**All**
Abstract	52 days	Mon 10/8/18	Wed 11/28/18	All
Software Design	**31 days**	**Mon 10/8/18**	**Wed 11/7/18**	**Matthew McDade**
Modules 1...n	**31 days**	**Mon 10/8/18**	**Wed 11/7/18**	
Psuedo Code	31 days	Mon 10/8/18	Wed 11/7/18	Matthew McDade
Hardware Design	**31 days**	**Mon 10/8/18**	**Wed 11/7/18**	
Modules 1...n	**31 days**	**Mon 10/8/18**	**Wed 11/7/18**	
Simulations	31 days	Mon 10/8/18	Wed 11/7/18	Julie Aichinger
Schematics	31 days	Mon 10/8/18	Wed 11/7/18	Laveréna Wienclaw
Parts Lists	**52 days**	**Mon 10/8/18**	**Wed 11/28/18**	**Laveréna Wienclaw**
Parts list(s) for Schematics	52 days	Mon 10/8/18	Wed 11/28/18	Ryne Turner
Materials Budget list	52 days	Mon 10/8/18	Wed 11/28/18	Laveréna Wienclaw
Proposed Implementation Gantt Chart	52 days	Mon 10/8/18	Wed 11/28/18	Julie Aichinger
Conclusions and Recommendations	52 days	Mon 10/8/18	Wed 11/28/18	Ryne Turner
Final Design Presentations Part 1	1 day	Thu 11/8/18	Thu 11/8/18	
Final Design Presentations Part 2	1 day	Thu 11/15/18	Thu 11/15/18	
Secondary Parts Request Form	14 days	Thu 10/4/18	Wed 10/17/18	Laveréna Wienclaw
Final Parts Request Form	56 days	Mon 10/8/18	Sun 12/2/18	Laveréna Wienclaw

Table 28: Midterm Design Gantt

Task Name	Duration	Start	Finish		Resource Names
SDPII Implementation 2018	**103 days**	**Mon 1/14/19**	**Fri 4/26/19**		
Revise Gantt Chart	14 days	**Mon 1/14/19**	Sun 1/27/19		Ryne Turner
Implement Project Design	**96 days**	**Mon 1/14/19**	**Fri 4/19/19**		
Hardware Implementation	**56 days**	**Mon 1/14/19**	**Sun 3/10/19**		
Breadboard Components	13 days	**Mon 1/14/19**	Sat 1/26/19		Ryne Turner
Layout and Generate PCB(s)	14 days	Sun 1/27/19	Sat 2/9/19	5	Ryne Turner
Assemble Hardware	7 days	Sun 2/10/19	Sat 2/16/19	6	Laveréna Wienclaw
Test Hardware	14 days	Sun 2/17/19	Sat 3/2/19	7	Ryne Turner
Revise Hardware	14 days	Sun 2/17/19	Sat 3/2/19	7	Ryne Turner
MIDTERM: Demonstrate Hardware	5 days	Sun 3/3/19	Thu 3/7/19	8	Ryne Turner
SDC & FA Hardware Approval	0 days	Fri 3/8/19	Fri 3/8/19	10	Ryne Turner
Software Implementation	**56 days**	**Mon 1/14/19**	**Sun 3/10/19**	**11**	
Develop Software	27 days	**Mon 1/14/19**	Sat 2/9/19		Matthew McDade
Test Software	21 days	Sun 2/10/19	Sat 3/2/19	13	Matthew McDade
Revise Software	21 days	Sun 2/10/19	Sat 3/2/19	13	Matthew McDade
MIDTERM: Demonstrate Software	5 days	Sun 3/3/19	Thu 3/7/19	15	Julie Aichinger
SDC & FA Software Approval	0 days	Fri 3/8/19	Fri 3/8/19	16	Julie Aichinger
System Integration	**42 days**	**Sat 3/9/19**	**Fri 4/19/19**		
Assemble Complete System	14 days	**Sat 3/9/19**	Fri 3/22/19		Laveréna Wienclaw
Test Complete System	21 days	Sat 3/23/19	Fri 4/12/19	19	Matthew McDade
Revise Complete System	21 days	Sat 3/23/19	Fri 4/12/19	19	Ryne Turner
Demonstration of Complete System	7 days	Sat 4/13/19	Fri 4/19/19	21	Julie Aichinger
Develop Final Report	**99 days**	**Mon 1/14/19**	**Mon 4/22/19**		
Write Final Report	99 days	Mon 1/14/19	Mon 4/22/19		All
Submit Final Report	0 days	Mon 4/22/19	Mon 4/22/19	24	Laveréna Wienclaw
Spring Recess	7 days	Mon 3/25/19	Sun 3/31/19		
Project Demonstration and Presentation	0 days	Fri 4/26/19	Fri 4/26/19		All

Table 29: Proposed Implementation Gantt
(Written by Julie Aichinger and Ryne Turner.)

6. Design Team Information

Julie Aichinger, Computer Engineering, ESI: yes.

Matthew McDade, Computer Engineering, ESI: yes.

Ryne Turner, Dual Computer & Electrical Engineering, ESI: yes.

Laveréna Wienclaw, Computer Engineering, ESI: yes.

7. Conclusions and Recommendations

In conclusion, this proposal emphasizes the need for an effective communication device that allows the students to save time and the University of Akron to maximize its current parking inventory. This smart parking deck design will be valuable in both of these areas it aims to mend. The project requires an advanced knowledge of circuit design and sensor technology, which is applicable to the logic and signal processing classes taken by the participating computer and electrical engineers. The project also requires the design of mobile applications as well as the displays for each parking deck level. This means that the proposed concept has both a hardware and a software component. The current plan of this design project is to test using ten strategically-positioned spaces of the Schrank Parking Deck. If given the opportunity to expand this project, it would be helpful to sense multiple spaces on multiple floors of multiple parking decks simultaneously, but because of the financial and time limitations of this assignment, this kind of testing will be outside of the scope of the technical design. It would be useful for future projects to improve upon this design given this set of problems to resolve. (Written by Matthew McDade and Ryne Turner.)

Honors Project Sponsor: Mr. Gregory Lewis

Readers: Drs. Kye-Shin Lee and Shiva Sastry

Honors Faculty Advisor: Dr. Igor Tsukerman

8. Final Demonstration

Final Mobile Application Design

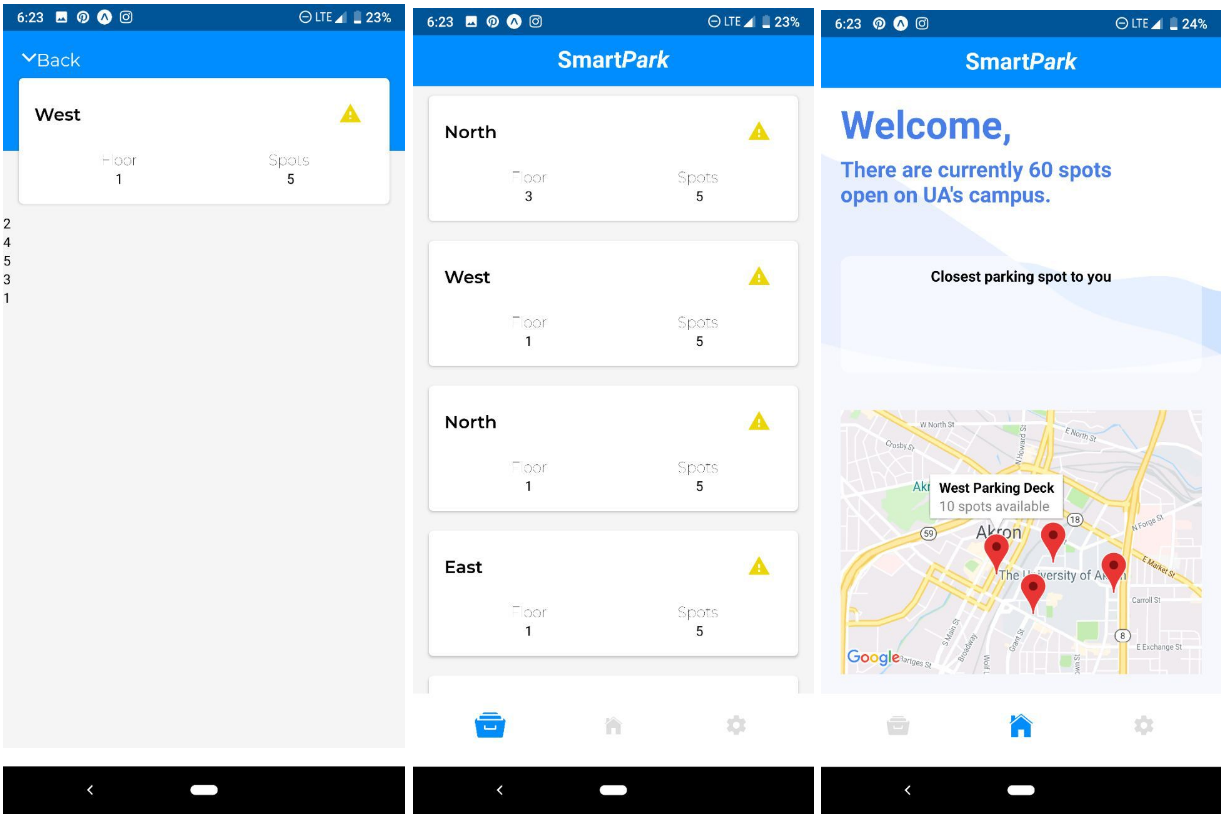

Figure 18: Final Design of the Mobile Application
(Design by Julie Aichinger.)

Final Eagle Schematics

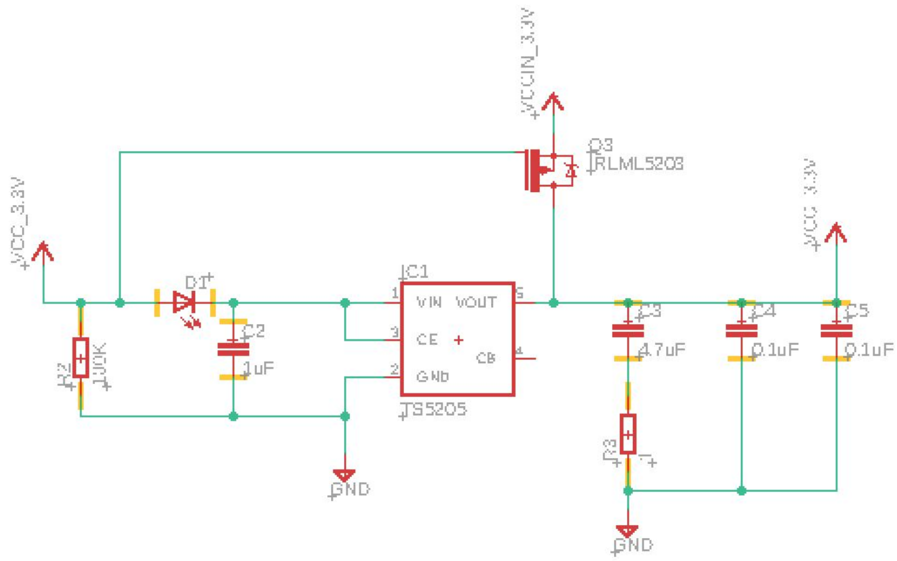

Figure 19: Low-Dropout Voltage Regulator Eagle Schematic
(Designed by Laveréna Wienclaw.)

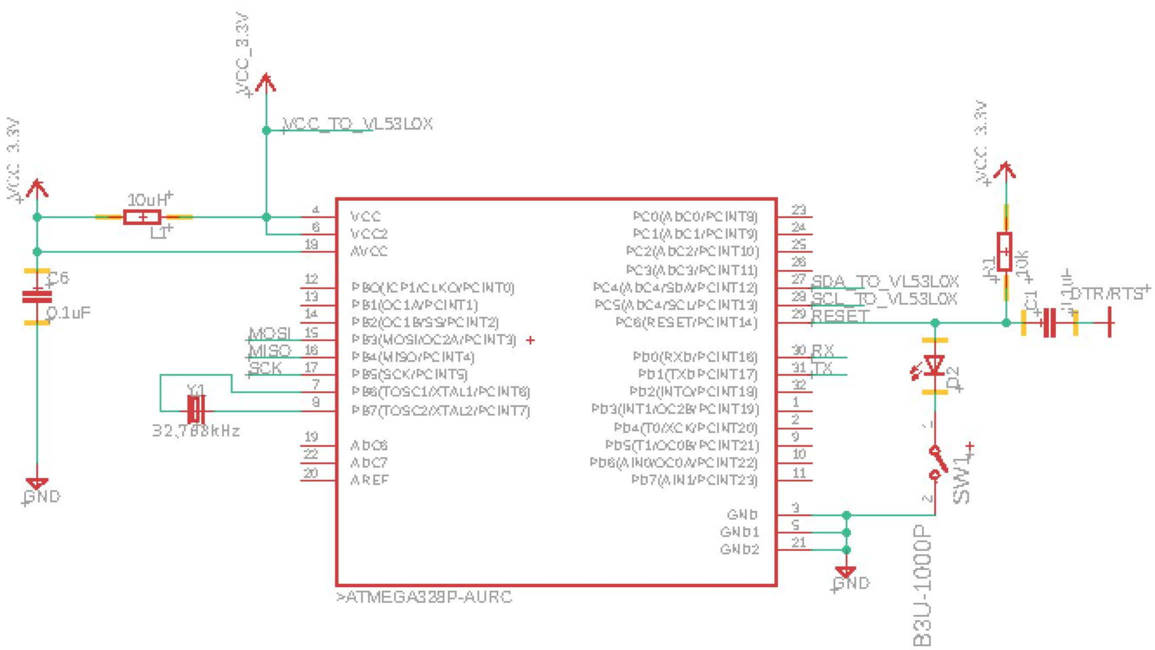

Figure 20: Microprocessor Eagle Schematic
(Designed by Laveréna Wienclaw.)

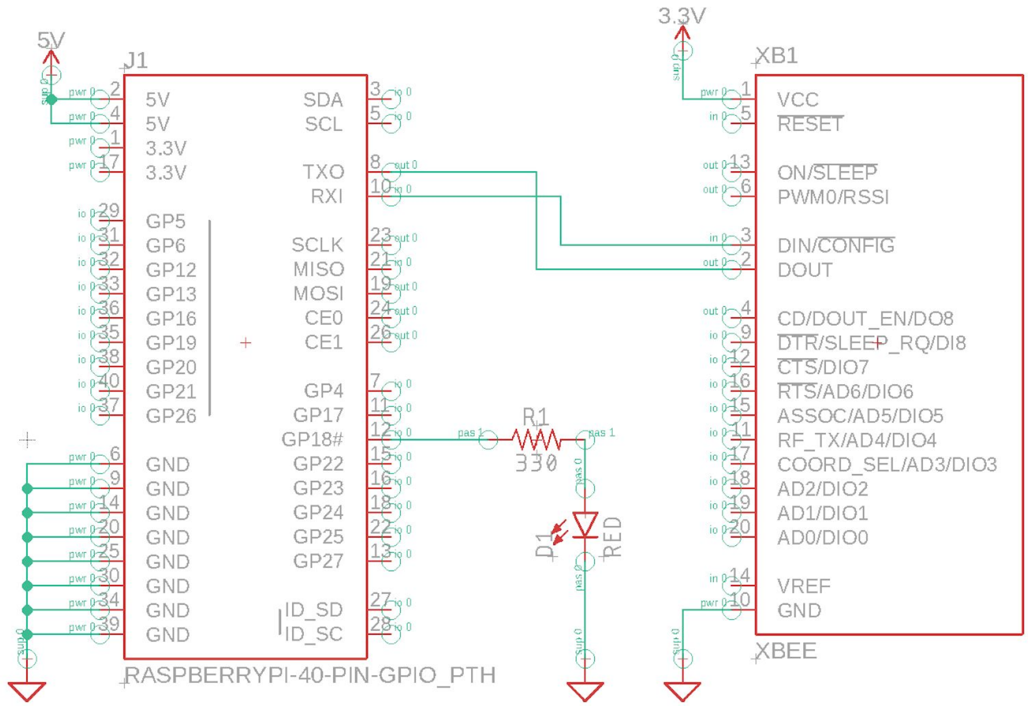

Figure 21: Hub Eagle Schematic
(Designed by Ryne Turner.)

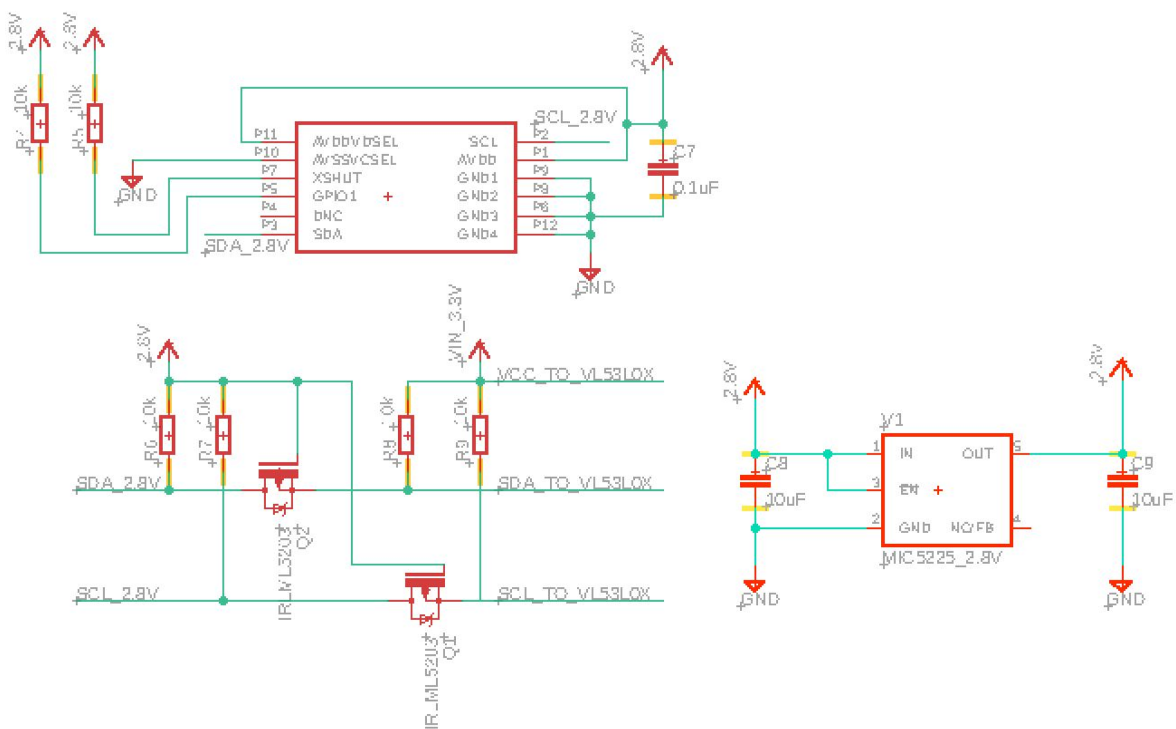

Figure 22: Time-of-Flight Sensor Eagle Schematic
(Designed by Laveréna Wienclaw.)

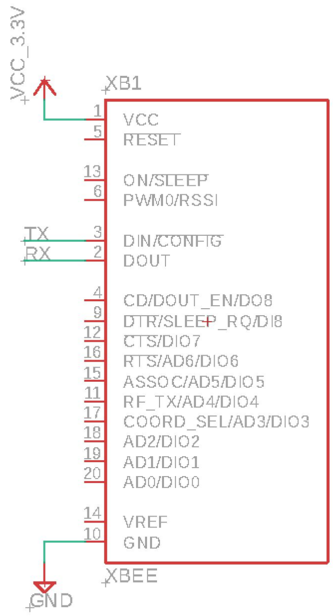

Figure 23: Module Zigbee Connection Eagle Schematic
(Designed by Laveréna Wienclaw.)

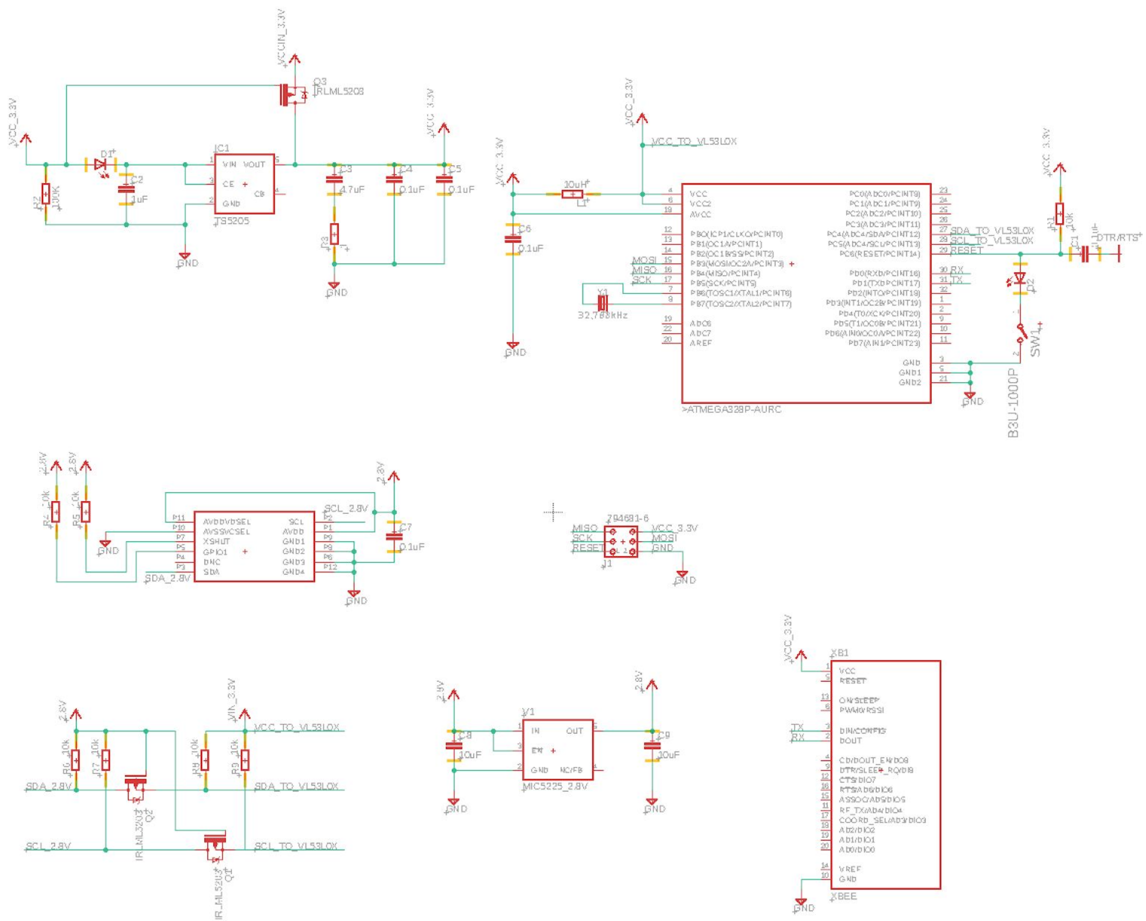

Figure 24: Overview of Module Eagle Schematic
(Designed by Laveréna Wienclaw.)

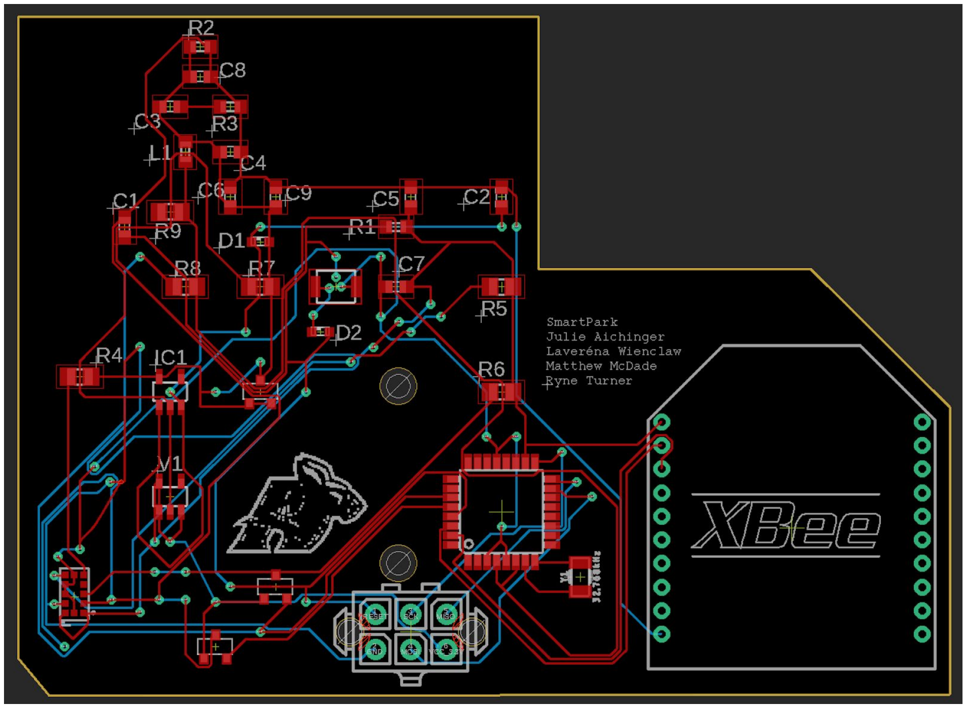

Figure 25: Module PCB Layout
(Designed by Laveréna Wienclaw and Ryne Turner.)

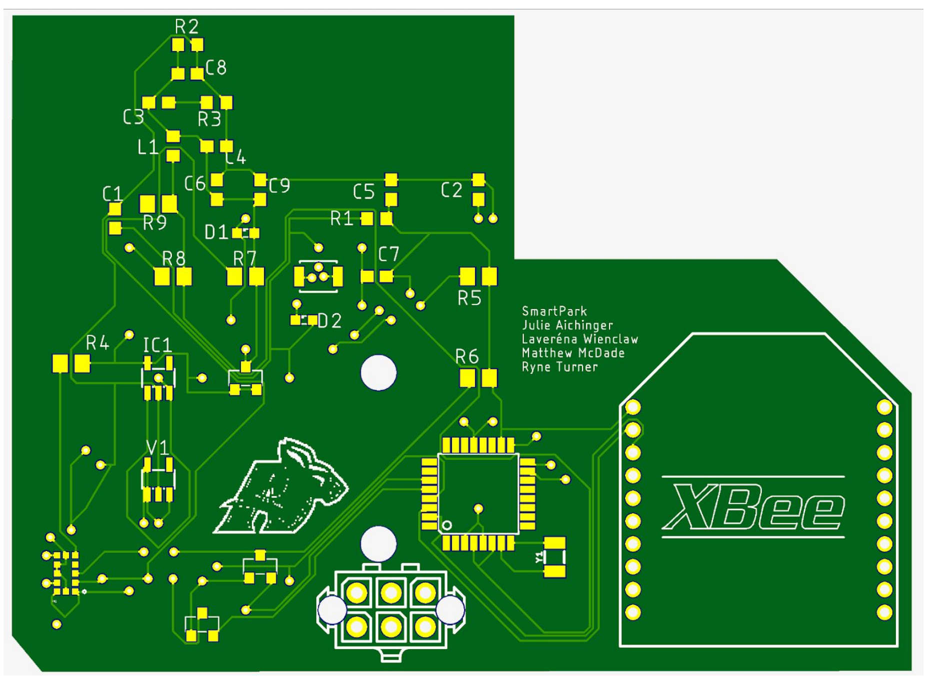

Figure 26: Module PCB
(Designed by Laveréna Wienclaw and Ryne Turner.)

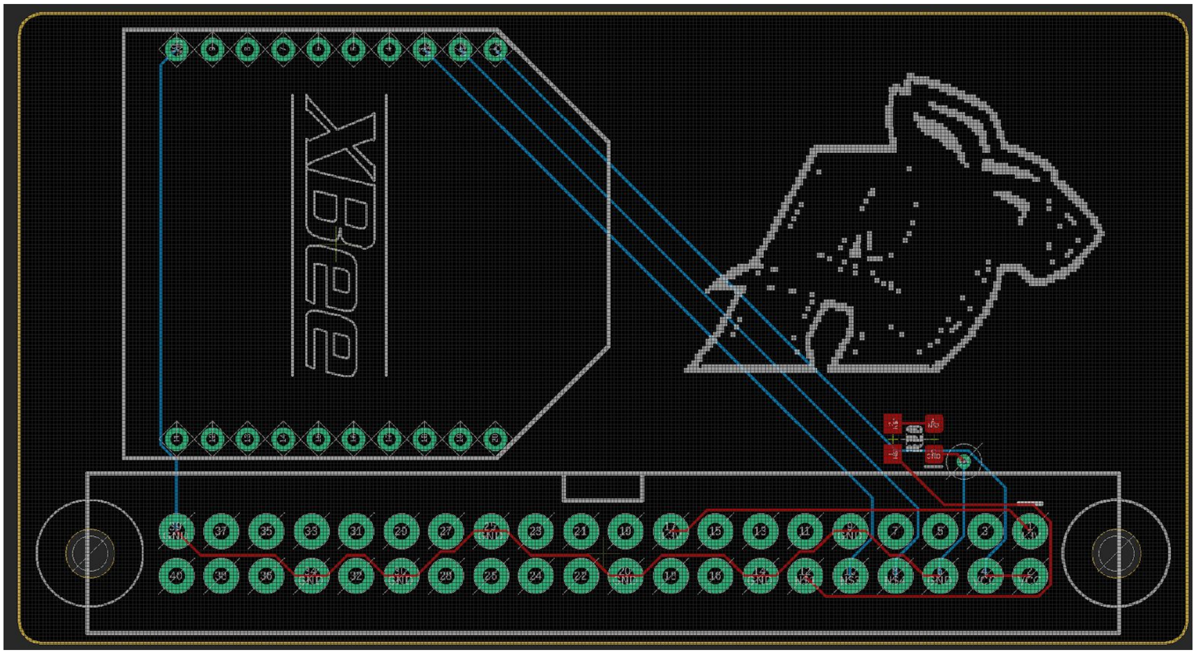

Figure 27: Overview of Raspberry Pi Hat Schematic
(Designed by Ryne Turner.)

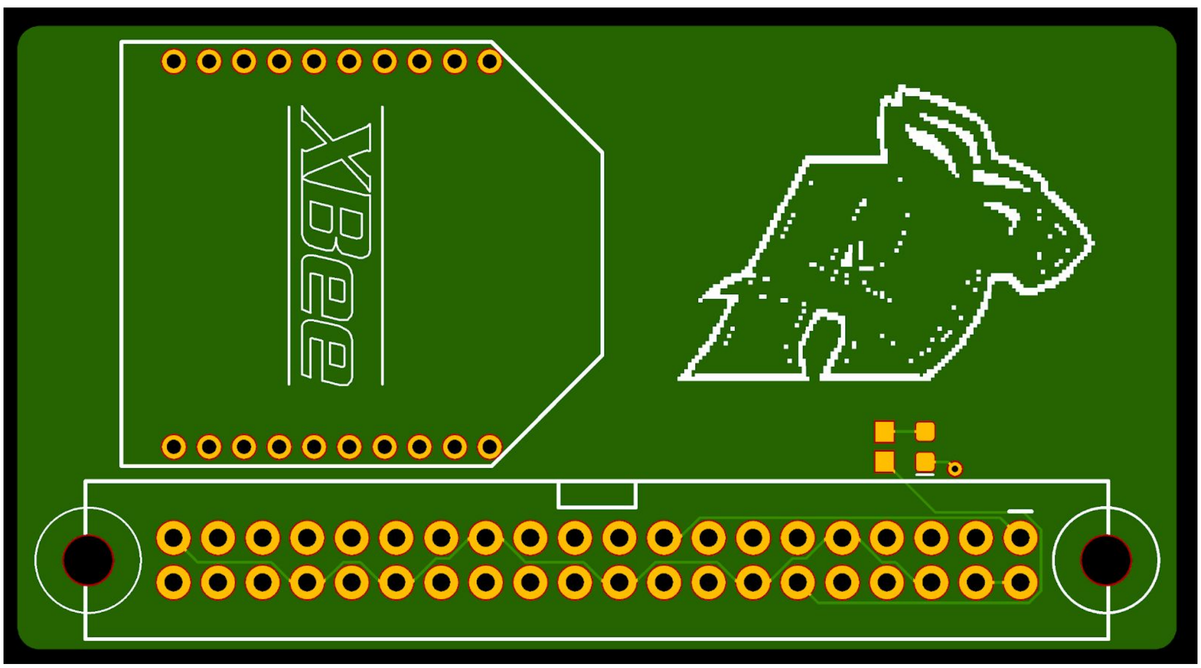

Figure 28: Raspberry Pi Hat PCB Layout
(Designed by Ryne Turner.)

9. References

Chinrungrueng, J., Sunantachaikul, U., Triamlumlerd, S. (2007). Smart Parking: an Application of optical Wireless Sensor Network. *Applications and the Internet Workshops 2007.SAINT Workshops 2007. International Symposium on.*

Lu, R., Lin, X., Zhu, H., Shen, X. (2009). SPARK: A New VANET-based Smart Parking Scheme for Large Parking Lots. *IEEE INFOCOM 2009.*

Mohammad, T. (2009). Using Ultrasonic and Infrared Sensors for Distance Measurement. *World Academy of Science, Engineering and Technology International Journal of Mechanical and Mechatronics Engineering, 3*(3), 267-272. Retrieved February 28, 2018, from http://waset.org/publications/6833

Pala, Z., Inanc, N. (2007). Smart Parking Applications Using RFID Technology. *RFID Eurasia, 2007 1st Annual.* US7688225B1 - Method for managing a parking lot. (2000, January 14). Retrieved March 01,2018, from https://patents.google.com/patent/US7688225B1

S., A. (2016). Performance comparison of Infrared and Ultrasonic sensors for obstacles of different materials in vehicle/ robot navigation applications. *Materials Science and Engineering, 149*. Retrieved February 28, 2018, from http://iopscience.iop.org/article/10.1088/1757-899X/149/1/012141/pdf

US7688225B1 - Method for managing a parking lot. (2000, January 14). Retrieved March 01, 2018, from https://patents.google.com/patent/US7688225B1

US8799037B2 - Computer-implemented system and method for managing motor vehicle parking reservations. (2010, October 14). Retrieved March 01, 2018, from https://patents.google.com/patent/US8799037B2

10. Appendix

Datasheets for Parts List:

Zigbee Transceiver (ATSAMR21B18-MZ210PA):

http://ww1.microchip.com/downloads/en/devicedoc/atmel-42486-atsamr21b18-mz210pa_datasheet.pdf

Microprocessor (ATMEGA328-AURCT-ND):

https://www.mouser.com/datasheet/2/268/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48P-1315288.pdf

Battery Pack (534-2463CN):

https://www.mouser.com/datasheet/2/215/463CN-1112770.pdf

Laser Proximity Time-of-Flight Sensor (VL53L0CXV0DH/1):

https://www.pololu.com/file/0J1187/VL53L0X.pdf

Digi Xbee-Pro Zigbee:

https://www.mouser.com/datasheet/2/111/ds_xbee_zigbee-1019686.pdf

Smart Parking Deck

Project Design Final Report

Honors Research Proposal Individual Responsibilities

Team Leader: Julie Aichinger

During the design of the Smart Parking Deck, I was the team leader. The role of team leader involved managing the team, organizing and submitting progress reports, and scheduling meetings. I was responsible for keeping the project on track and making sure deadlines were met, as well as keeping mentors and advisors in the loop on project progress. I was also part of the software team for the project. In terms of software, the project included the designing and making of a mobile application for users, as well as a network for the communications within the parking deck. The application was designed to work on both iOS and Android devices, using React Native. The communication network allows for communication between the individual sensors, the nodes and hubs for each floor and parking deck, and the system database to keep track of information and data. The system database communicates to the app.

Date Submitted:
26 April 2019

Smart Parking Deck

Project Design Final Report

Honors Research Proposal Individual Responsibilities

Software Lead: Matthew McDade

My role as the Software Lead on the Smart Parking Deck project involves managing the design and implementation that will be programmed on the various components of the project. This includes how the sensors talk to the hub, how the hub displays the data locally, how the hub connects to the internet and where/how it stores parking spot information, and how that information is then displayed in the mobile app. All of these systems need to be designed for efficiency as to not be too power hungry, yet still be run often enough to provide accurate and up-to-date data. I will manage the implementation of these different systems to make sure they're all designed as efficiently as possible.

Date Submitted:
26 April 2019

Smart Parking Deck

Project Design Final Report

Honors Research Proposal Individual Responsibilities

Hardware Lead: Ryne Turner

During the design of the Smart Parking Deck, my role as the Hardware Lead was focused on the execution of our team hardware design to sense if a vehicle is occupying a parking space. This role included elementary circuit design elements, such as designing PCBs, building devices with sensors, and communications hardware. Each individual device is able to communicate with another device as well as the network hub that manages all of the incoming and outgoing parking data. This mandated that we have a system that is both energy efficient and easily manageable. I also oversaw the financial decisions concerning the purchase of circuit components.

Date Submitted:
26 April 2019

Smart Parking Deck

Project Design Final Report

Honors Research Proposal Individual Responsibilities

Engineering Data Manager: Laveréna Wienclaw

My responsibilities for the senior design project, Smart Parking Deck, include the status of the Archivist, a hardware technician, and an associate on the design of the hardware implementation. As our project involves a lot of submission deadlines of design documents, research, and so on, it remains my duty to ensure the documents are completely filled out and maintain the designated formatting. For the intended design implementation, we must construct a sensor hub that will be able to detect whether a car is in a spot or not, and will send this data to a network node. This is the hardware portion of our project that will largely be created by Ryne Turner, whereas I will act as the associate engineer on this end and the physical technician. Ryne will design the circuit with myself as a reviewer, and I will implement the physical circuit and test the robustness of the enclosure necessary for the parking spot placement.

Date Submitted:
26 April 2019