Easy Park

Muhammad Khan, Jayson Asplin,
Peter Nguyen, and Lorenzo Casimir.

Dept. of Electrical and Computer Engineering
University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — With the increasing number of vehicles and drivers on the road, all around the world, it has gotten somewhat difficult to find parking spaces around populated areas. Here at the University of Central Florida, the probability is well under 10% for finding an empty parking spot in most garages. This project would be looking at ways to minimize or help students, faculty, and UCF visitors to find parking spaces, without wasting too much of their time.**

***Index Terms* — Ultrasonic transducers, maximum power point trackers, wireless mesh networks, application software, microcontrollers, solar energy, database.**

I. Introduction

Easy Park is a project that is meant to help everyone find predetermined parking spaces, inside parking garages. Through the use of a simple app, webpage, and green LEDs; finding a parking space would be feasible. While the idea of finding available parking spaces is hackneyed, this project is one of the first to develop and implement a rudimentary Bluetooth low-energy (BLE) mesh network. Other previous projects have had mesh networks based on WiFi components, e.g. Xbee 2C. The main comparative difference between BLE and WiFi mesh would be the power consumption: BLE consume 10x lower current to do the same mesh operation.

Each parking space in the mesh network has its own module consisting of an ultrasonic sensor (used for detecting vehicles in the space) and a BLE module (used for communicating between other spaces). One of the modules in the mesh network also has a built in WiFi chip. This chip is used to connect to a database via web server. All of the modules would be battery operated. Each battery will be charged through a maximum power point tracking (MPPT) device via solar panels as the input.

Since the main objective of this project was power efficiency, only ultrasonic sensors are used to detect parking spots. However, with similar and previous projects, Hall effect and PIR sensors were also taken into account, but were not implemented.

The mobile application in this project was based in an android environment because android exists in ~85% of global operating system market. The integrated development environment (IDE), used for application development, was android studio (Reason: mostly due to good documentation and support available online).

II. Component Details and Specifications

In this section, we will briefly discuss the component details and specifications. The components used on the boards are listed in the following table:

Component name	Model/Brand
Bluetooth (BLE) module	HM-10
WiFi module	ESP-12F
Microcontroller (MCU)	ATMega328P
Ultrasonic Sensor	HC-SR04
LED	5050-G3500 SMD
9V Li-ion Battery	EBL
1W Solar Panel	Seeed Studio

Table 1. Components List

To expand on the details of each component, each of the following table goes over the specifications. Starting with the bluetooth module (HM-10):

Features	Specifications
Range	100m (open space)
Working temperature	-5 ~ 65 C
Current (Active mode)	8.5mA
Input Voltage/Current	+3.3VDC/50mA
Current (Sleep Mode)	400uA ~ 1.5 mA
Chipset	TI CC2541

Table 2. HM-10 specifications

Over here, it can be seen that the BLE module uses a nominal amount of only 8.5mA (during active power mode). This was one of the key factors to having a low power communicating device, relating to an overall low power module.

The details of the WiFi module can be seen in the following table:

Features	Specifications
WiFi protocol	802.11 b/g/n
Working temperature	-40 ~ 125 C
Current (Active mode)	~80mA
Operating Voltage	3.0 ~ 3.6 V
Frequency Range	2.4GHz-2.5GHz (2400M-2483.5M)

Table 3. ESP-12F WiFi module specifications

Previous projects have used WiFi modules to form a mesh network. It can be seen that there is ~ 8x current draw difference between the WiFi and BLE modules in normal operating mode. Since the WiFi module also has a built-in MCU, we used that MCU to communicate it with the BLE and ultrasonic sensor.

Lastly is the Specs of the ultrasonic sensor (HC-SR04):

Features	Specifications
Working Voltage	5 V
Working Current	15mA
Working Frequency	40Hz
Operating Range	2cm ~ 4m

Table 4. HC-SR04 specifications

The HC-SR04 module also meets our small current consumption specification (each module only requires about 15 mA with a sensing range of 2 cm ~ 4m). Increasing the amount of sensors per module will provide greater accuracy, however, the placement of the Easy Park modules will minimize the need of excessive sensing. Furthermore, power consumption will also increase, thus minimizing the modules' battery life..

A single lithium ion battery is responsible for supplying power to the hardware components of Easy Park. Each battery(manufactured by EBL Mall) is rated at 9V, has a capacity of 600 mAh, and has a charge rate up to 1200 times. These batteries are readily available at local retailer locations. Therefore, replacing the battery does not cause an inconvenience. The battery is able to supply more than enough power to the mesh network system with/without the WiFi module: The WiFi requires a maximum current draw of about 350 mA.

The regulators that are used to power the Wifi and BLE will be discussed in the hardware section. For the modules that did not have the WiFi module onboard, we used the ATMega328p to communicate with the BLE and sensor. The 5050-G3500 surface mount LED is on all of the parking modules. The purpose of the LED is to illuminate a green light in front of the parking space when the space is empty and toggle the LED off when the space is occupied. We have limited its operating current to 8mA using a 300-ohm resistor. In doing so, its brightness will not distract drivers.

III. Overview of the System

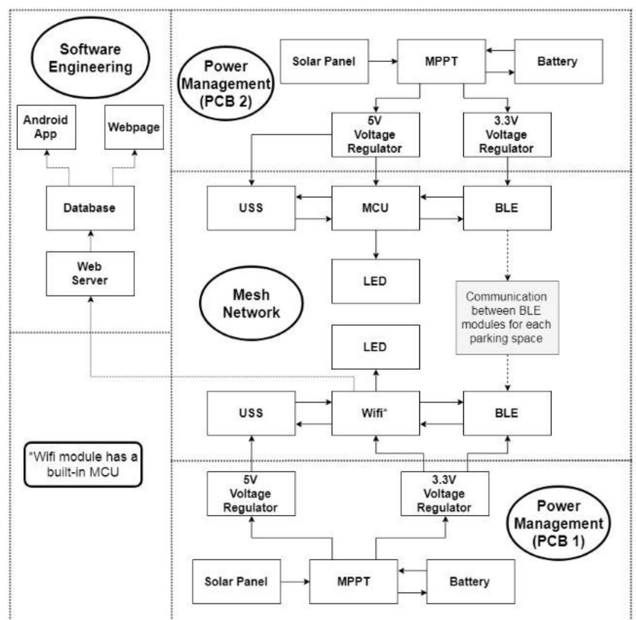


Fig. 1. Overall block diagram for Easy Park

A general block diagram of Easy Park system is shown in the above Figure 1. It consists of three subsections: Power management, mesh network, and software engineering.

A. Power Management

The power management consists of the solar panel, maximum power point tracker (MPPT), li-ion battery, and voltage regulators. First, the solar panel harvests the energy from the sun and converts it into electricity. Next, electricity would supply the MPPT the current to charge the battery. Lastly, the battery would provide enough current through the individual voltage regulators to power the whole system.

B. Mesh Network

The mesh network consists of two different types of Easy Park devices. One of which has the Wi-Fi module and another which has the ATmega328p microprocessor. The rest of the components are on both: the BLE module, HC-SR04 ultrasonic sensor, and the LED. The BLE is utilized as the main source of communication to relay data in the mesh network.

C. Software Engineering

In the software engineering section, the data is gathered and taken from the mesh network for connecting to the web server from the internet. At the same time, data is also relayed across the WiFi module. This web server acts as a gateway for connecting to the Amazon web service database (written in PHP). this is for inputting data from the parking sensors. Another PHP page takes this data from the database, encodes it in JSON, and displays onto the android app. There is also a webpage that shows the data.

IV. Hardware Detail

In this section, we will discuss the hardware aspect of the project in great detail. This includes the voltage regulator selections for the battery, schematic design for each parking spot, and the solar charger.

A. Voltage Regulators

The selection of the voltage regulators was aided with the help of the Webench designer. This software, developed by TI, allows for multiple regulator selections. Based off of the necessary voltage and current specifications, the PCB components is properly powered by the lithium-ion battery. The software aids the selections with number of parts, chip sizes, price, and power efficiency. Three, regulator centered, circuit designs were chosen, offering over 90 percent power efficiency.

TVL6256(8,9)DBVR: TI's TLV is a high efficiency step-down buck converter. This device takes an input voltage range of 2.5V to 5.5V and outputs from 0.6V to input. The device provides soft start up, over current protection, and thermal shutdown protection. This regulator also provides up to 95 percent efficiency. The TLV regulator is needed for communication between the wifi module and the CPE team. As of today, the device is in proper working order, capable of providing 5 to 3.3 voltage regulation and allows over 150mA of current to power and program the wifi module via USB.

TPS560200DBVR: Unlike the previous regulator, this TPS chip will be used more frequent. The TPS is a synchronous monolithic buck converter that offers 0.5 A of continuous output current. This chip has similar features with the previous regulator. However, the TPS is meant to take an input of 4.5 to 17 volts and output 0.8 to 6.5 volts. The importance of the 0.5 A of continuous current is that the chip will be used to power both the wifi and bluetooth PCB components. These devices may require more current than specified, mainly due to the wifi/ bluetooth module communication range. This regulator is meant to step down a 9 volt battery input and output 3.3 volts. The input voltage so far is too high for our current regulator.

TPS56(2,3)200DDCR&TPS56220(1,8)DDCR: Just like the previous chips. Each of the different TPS chip versions offers similar features. The TPS56X is a better version of the TPS56220X chips. Each chip offers a 4.5 to 17 volt input step down to an output of 0.76 to 7 volts regulation. The chips are very similar substitutions in terms of TI parts availability. The purpose of the regulator is to step down 9V battery input and regulate to an output of 5V for the ultrasonic sensor and MCU or 3.3 volts for the bluetooth. The regulators alone can replace the previous ones, but does not offer as much efficiency and parts availability. The chips work properly as of today.

B. Solar Charger

Our selection is the BQ24650 device, a highly-integrated switch-mode battery charge controller from Texas Instruments (TI). It charges the lithium-ion battery in different phases. A precondition phase will charge the battery at ten percent of the charging rate if the battery is completely depleted, the battery becomes prepared to accept more charge after 30 minutes.. Next, a constant-current fast charging phase charges at a hundred percent of the charging rate until the voltage reaches a certain voltage. A constant-voltage phase comes into effect that supplies enough current to maintain the max voltage constant. A decrease in charging current will happen as a result of the voltage maintenance. The charging will terminate once the charging current drops below ten percent of the charging rate.

The regulation of the battery voltage is programmed using a resistor divider at the output of the solar charger. We set the battery regulation voltage to 8.4V by using equation (1) and setting V_{BAT} equal to 8.4V and R_1 equal to 100 kΩ. R_2 was calculated to be 300 kΩ.

$$V_{BAT} = 2.1V \times [1+\frac{R_2}{R_1}] \quad (1)$$

The charging current is determined by a current sense resistor R_{SR}. It was the group's decision to set the charging current to 200 mA by using a 200 mΩ resistor. The equation used to calculate the value of the sense resistor is shown in equation (2) by setting I_{CHARGE} equal to 200 mA.

$$I_{CHARGE} = \frac{40mV}{R_{SR}} \quad (2)$$

The other parameters are determined based on the charging rate. The inductor and output capacitor selection depends on it. The pre-charging rate is ten percent of the charging rate, which is 20mA in this case. The charging termination will occur when current supplied to the battery is less than ten percent of the charging rate.

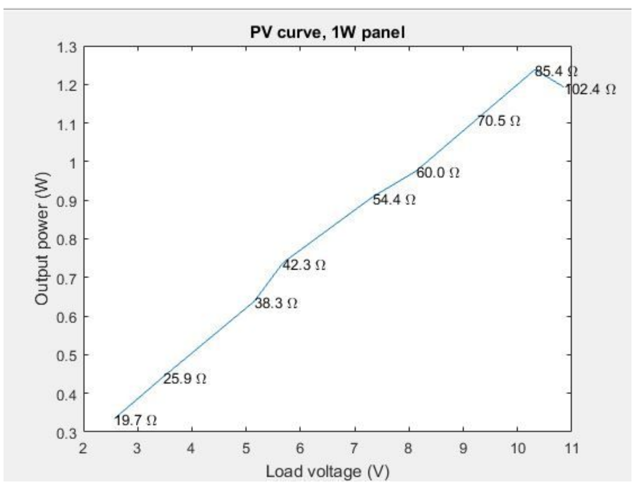


Fig. 2. PV curve for 1W panels in series

Maximum power point tracking (MPPT) is an electronic device that outputs the maximum amount of power from the solar panel. To harvest the most power from the sun, the maximum power point has to be determined. We decided to connect two panels in series, then tested the panels with multiple resistor loads under 100 ohms and one a bit over 100 ohms in parallel to the panels for currents and voltages on a bright, sunny day. The 85.4 Ω resistor yielded the most power of 1.2384 W with a voltage of 10.32V and current of about 120 mA. This voltage will be discussed further on implementation with the TI chip. The PV curve for the panels is illustrated by the MATLAB plot in Figure 2.

The TI chip implements maximum power point tracking for the solar panels in series by regulating the input voltage coming from the panels. This allows us to extract the panel's maximum output power. A constant voltage algorithm is used, the simplest MPPT method, and is preset with a constant voltage. From testing the panels on a typical sunny day, we have found the voltage at which the solar panel is at its maximum power is at 10.32V. The voltage at the MPPSET pin is regulated to 1.2V. This gives the regulation voltage shown in equation (3), where V_{MPPSET} is set to 10.32V and the resistor values R3 and R4 were determined based on the availability on Mouser Electronics. We selected a 357 kΩ resistor for R3 and a 47 kΩ resistor for R4, which yields V_{MPPSET} value of 10.31V.

$$V_{MPPSET} = 1.2V \times [1+\frac{R_3}{R_4}] \quad (3)$$

V. Software Details

In this section, we will discuss the software aspect of the project in great detail. This section will discuss the mobile app, mesh network, database, and web server.

A. Mobile App

The mobile application was one of a few ways we chose to display the data to the users. It was designed to let users know the garages' occupancy. The idea in mind was to have users using this mobile app while driving. The mobile app should be simple, an easy to read, and easy to use. This is to keep the users eyes off their phone as much as possible and on the road. A glance should be all it takes to see all the data they need in order to make the decision of where to park.

The Easy Park App utilizes a progressbar to show how full the garages are. The progressbar are colored coded to easily see garages occupancy. When less than 50%, full the bar is green; between 50% to 70%, the bar is yellow; between 70% and 90%, the bar is orange; and greater than 90% is red. Using colors to display how full the garages are will give the users a better representation of the garages occupancy compared to using numbers as the primary way to display the data. Furthermore, we included the number of vacant spots and the actual percentage to give the user more detailed information. This is demonstrated in Figure 3. A button is also place on top of the progressbar to easily access the next page or data for a garage. Theses pages have the same layout as the homepage but instead shows the floor occupancy.

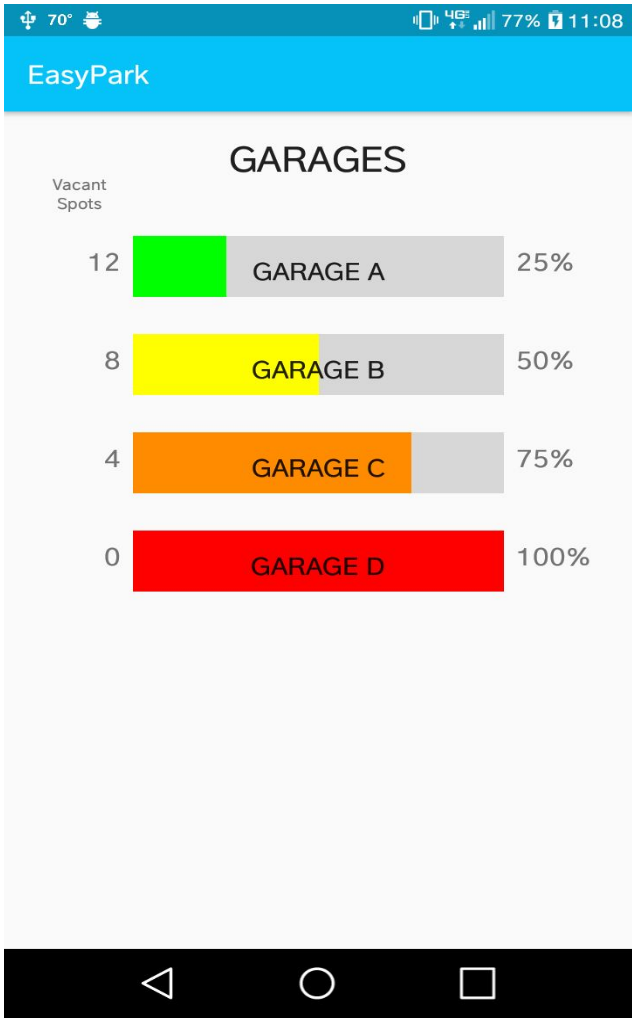


Fig. 3. Easy Park Homepage.

The IDE used to create the mobile app was Android Studio, Android's official IDE. Android Studio is a power tool, used for making android applications. Android studio allows us to keep the application simple and clean. Placing a button on top of a progress provides the mobile app a simple and clean look. Doing this will also help the users intuitively use the mobile application. If the users are familiar with using applications on smartphones, they could easily use the Easy Park app. There user will not necessarily see a button to go to garage A since there is no other indications of how to get there. Furthermore, there is only one instance of Garage A on the homepage. The user can most likely guess that pressing on the Garage A progressbar will lead them further into the app to show the floor occupancy of that garage. Android studio is also good with managing different layouts to work with different screen sizes. It's already implemented in the IDE so that the developer won't, for the most part, create different dimension for how the app looks, depending on the screen size and resolution.

B. Mesh network

Easy Park uses a mesh network to connect all the nodes together. A mesh network is best suited for Easy Park compared to the different network topologies. It is more reliable and the data will have different routes to get to the end node, as a contingency to node failure. We used Bluetooth Low-Energy as our source of communication.

The mesh network for Easy Park was implemented using the rudimentary method. Each node would gets its own specific code. This was due to having each BLE module connecting to the next BLE module MAC address. A MAC address is specific to each BLE module. Since the BLE needs to connect to the next BLE, the MAC prevents the data from going in the wrong direction. The mesh network was also designed to have the data always reach the end node. If the data was not sent properly to the next node, it would keep trying to connect the available BLE modules in the general correct paths, until the data has been sent successfully. A reason the data could not be sent properly could be due to a node not working properly or a busy node, the BLE is currently connected to another BLE. Using this method for the mesh network is the most optimal form of reliability.

Figure 4 shows an overview of the mesh network. The circles represent Easy Park PCB2 modules and the square represent Easy Park PCB1. The circle nodes end goal is to reach the square node to send the data to the server. The directional arrows show the flow of where the data travels.

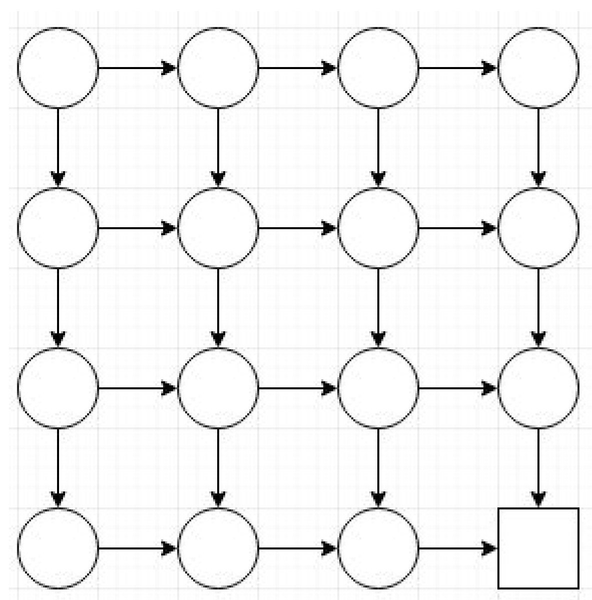


Fig. 4. Mesh Network

The IDE used to program the mesh network was Arduino IDE. The Arduino IDE uses a set of C/C++ functions. This vastly improves the development of the mesh network since our group is familiar with the syntax. The Arduino IDE also provides sample codes to test some of our components. This also include the Arduino ISP which was used to burn the bootloader on our ATmega328p chip. Arduino IDE was also used to upload the code onto the ATmega328p chip.

C. Database and Web Server

For selecting the database, we ended up using amazon web services. It has up to 750 hrs a month, free for the first twelve months, and storage capacity of 20GB. Other options were GoDaddy and DigitalOcean. For our purpose, this worked out quite well. As for the web server, instead of using amazon's EC2 service, we went with 000webhost.com. The reason being was that at signup, we immediately get our own domain right off the bat. This initially seemed to be convenient for testing purposes, and eventually, we ended up keeping it for our project as well.

The following image (Fig. 5) is a rough ER diagram for this project.

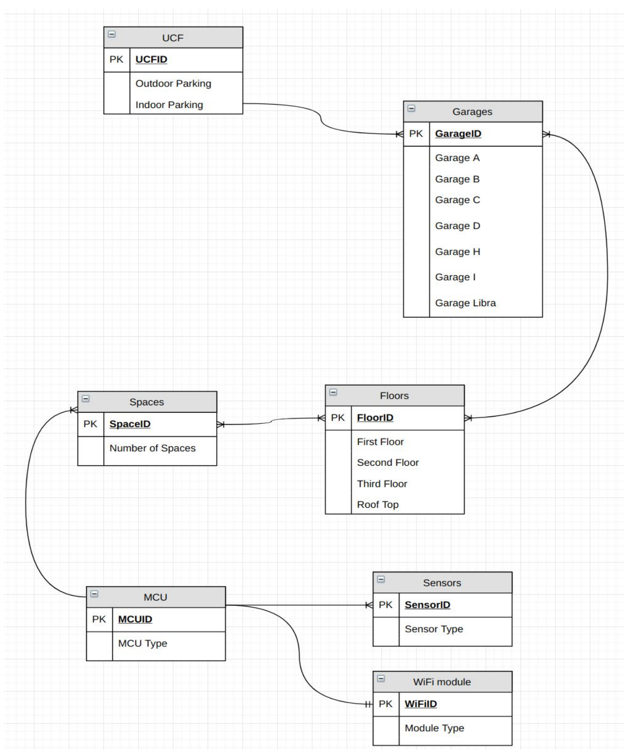


Fig. 5 Entity relationship (ER) diagram for Easy Park

UCF has multiple parking garages, multiple floors, and many parking spaces to each floor. Depending upon the layout of these spaces, the end node will be the WiFi module

The function of the web server is to act as a gateway to inserting and retrieving data from the database. This was done through a php script. We have one PHP page for inputting data to the database. The image below, lets us know if the operation was successful by toggling between vacant and occupied (this was mostly used for our testing). We have another php page that connects to the database and encodes the data in JSON format (for the purpose of using it with the android app).

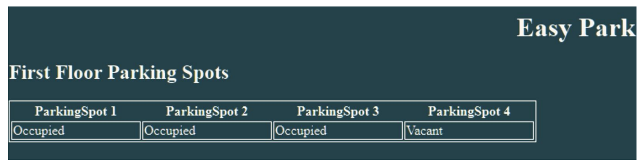


Fig. 6. Easy Park Web Page

VI. Printed Circuit Board Design

The custom printed circuit boards (PCBs) were designed by the EE members of Group 26 and were fabricated by OSH Park. Three PCB designs are utilized in the final prototype of the entire wireless mesh network of Easy Park.

A. 'PCB 1' Board

This PCB is the brain of Easy Park. It contains the ESP-12F WiFi module in the network. The board houses a HM-10 BLE module and a 6 pin male header for the WiFi module. The 6 pin is used for programming the module itself. A HC-SR04 ultrasonic sensor and 5050-G3500 SMD green LED are mounted, facing upright, for vehicle detection. A total of three regulator circuits are used on this board. A single regulator circuit will manage both the WiFi and BLE modules. These modules requires 3.3V, while the sensor requires 5V, using a different regulator circuit. Lastly, there is an internal WiFi regulator connected to one of the header pins. The pins provide the required 3.3V via USB. Implemented using a two-layer design, the components are almost exclusively mounted on the surface of the board. An image of the layout of PCB 1 is shown in Fig. 7.

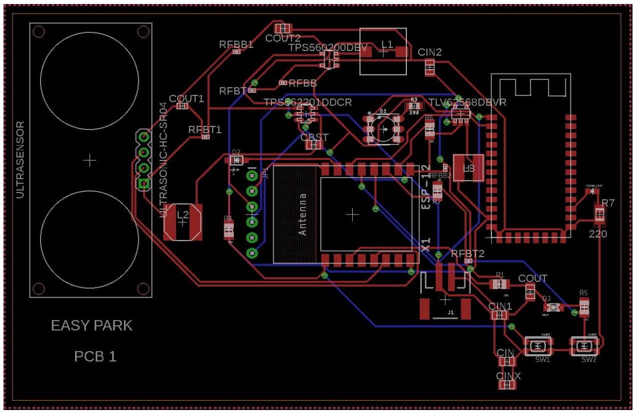


Fig. 7. PCB 1 layout

B. 'PCB 2' Board

The next PCBs are extensions of PCB 1. No WiFi module is onboard here as the ATMega328P microchip replaces it. There is also a 16MHz crystal. Similar to PCB 1, this board houses the BLE, ultrasonic sensor, and the SMD green LED. A 2x3 ICSP header and 1x3 header was included on the board. Whether we want to install a bootloader or upload sketches for the MCU, it will serve as a programming contingency. The board was implemented using a two-layer design, with a ground plane on the top and bottom layer. most the components are completely surface-mounted. An image of the layout of PCB 2 is shown in Fig. 8, below.

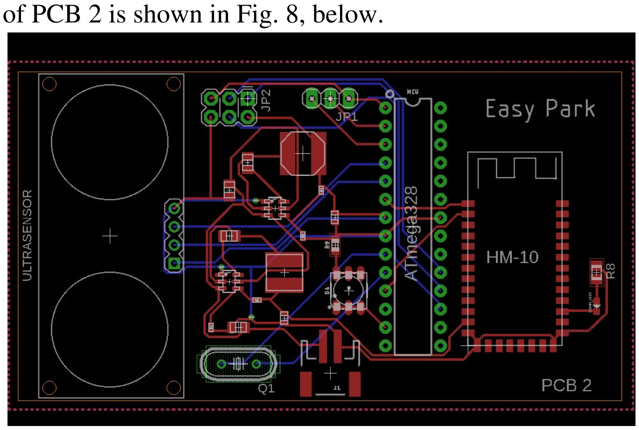


Fig. 8. PCB 2 layout

C. Solar Charger Board

The BQ24650 MPPT integrated circuit was originally designed by Texas Instruments, using a four-layer board, as seen from BQ24650EVM Evaluation Module user's guide. While researching relevant information for the layout guidelines for the circuit, we have examined a support forum post about redesigning the board using two layers instead of four, from a previous UCF senior design group. All the components of this board are all surface-mounted. There is a ground plane on the top and bottom layers like the previous boards. An image of the solar charger layout is shown in Fig. 9.

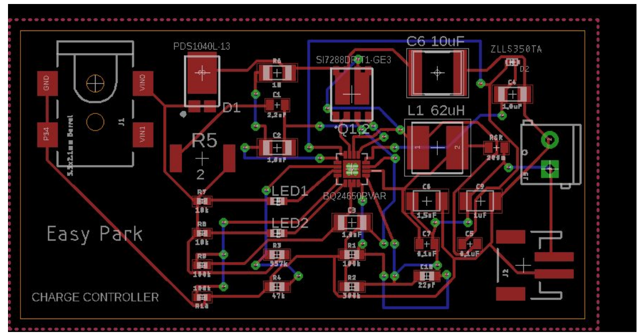


Fig. 9. Solar charger board layout

VIII. Conclusion

In this project, we have looked at different components of electrical and computer engineering, bringing the mesh network to life. In Senior Design 1, we started the project with the main goal of attaining a low-power network, where parking spots talk to each other. WiFi had been one option, but since it had been done before, we decided to use Bluetooth low-energy. We ended up creating a rudimentary mesh network through the use of specified AT commands that came preloaded with each bluetooth module. With the aid of smart programming practices and delays in-code, we are able to communicate a series of bluetooth modules by relating parking spot information. For these modules to be fully mobile and wireless, apart from being powered by a battery, we also used solar panels to charge the batteries through a MPPT. The MPPT was a custom design that was modeled around a BQ chip, with the application provided in the TI data sheet. The information for parking spots are being displayed on an android app as well as on the PCBs, via LED indication.

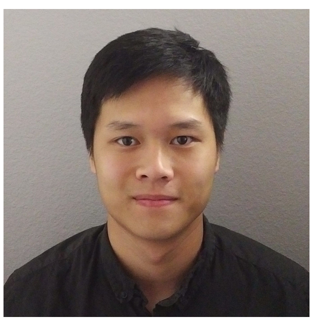
Acknowledgement

The authors wish to acknowledge Dr. Samuel Richie for his advice and guidance with this project. The authors would like to thank Quality Manufacturing Services (QMS) at Lake Mary, FL for their assistance in assembling the components for each board.

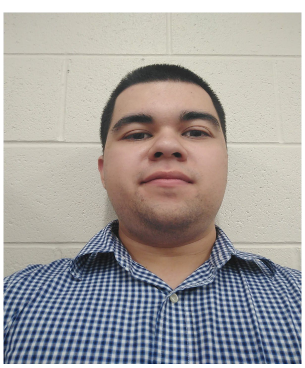
Authors

Muhammad Khan is a senior Computer Engineering student that further pursued his studies with the MSC program at UCF. Particular fields of interest include deep learning applications and frameworks, spintronics and TFET's.

Lorenzo Casimir is an Electrical Engineering from the University of Central Florida. His plans involves joining the Military when he graduates.



Peter Nguyen is a Computer Engineering student. After he graduates he plans on finding a job to start his career as a Computer Engineer. His interests are in computer architecture,software engineering, and entrepreneurship.



Jayson Asplin is a senior Electrical Engineering student. His interests are in electronics design and communications. After he graduates with his degree in May 2018, he plans on finding a job to start his career as an Electrical Engineer and pursue a master's degree in the future.