A Technical Report Submitted to the

Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements of the Degree
Bachelor of Science, School of Engineering

,

Technical Project Team Members
Nawar Wali
Sean Reihani
Gunther Abott
Cameron Davis

Nawar Wali
Spring, 2021

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature ____ Date __________
Nawar Wali

Approved ____ Date __________ ,
Harry Powell, Department of Electrical and Computer Engineering

Where2Park

Where2Park is an IoT system that tracks the status of parking spots in real time, and conveys this information to a central application that in turn conveys this information to end users. The system employs a fleet of vehicle-detecting sensor nodes, one per parking spot, which form a mesh network that relays information about each spot (i.e., the presence/absence of a car, node battery level) back to a central computer system. This central computer provides a front-end user interface that, combined with pre-set metadata such as location, allows the user to see exactly which parking spots are vacant in real time via a graphical user interface.

Nodes communicate with one another using Zigbee technology, an 802.15.4 based specification for low data rate, wireless, personal area networks. The sensor nodes will form a mesh network using Zigbee, which allows for greater practical range than a star network topology.

Any node can act as the Zigbee coordinator for the network, which will be the node that creates the network and is connected to a host computer through a USB connection. This node will interface between the mesh network and the desktop application running on the host computer.

To detect the car in the respective spot, a specific type of detector must be used. If an ultrasonic sensor is used, soiling could result in the misidentification of a parked car. To prevent common deviations, a metal detector is the best option. A series of a capacitor and inductor forms a tank circuit. A tank circuit is where energy is transferred continuously between an inductor and capacitor, which results in oscillations. The current flow provides a change in polarity that will allow magnetic fields to form and collapse. Ideally, the inductor would be the metal detecting coil which would oscillate near metals without a power source. To offset this, a BJT will provide continuous gain. The Colpitts oscillator can be described as a tank circuit with a BJT inverting amplifier. The oscillator can provide frequencies up to the 100kHz range which can support detecting cars. In order to account for external features such as Zigbee, a metal detector can certainly be accomplished with a microcontroller.

The metal detector should not inhibit the overall project progress. However, if the metal detector fabrication takes significant time, it would be wise to purchase a premade sensor. This would only be considered in an effort to save time.

The metal detector has a power supply of 4.5V in the form of 3 AA batteries with bypass capacitors. To make sure the current flow to the microcontroller stays at approximately 40mA, a sensitivity potentiometer is set up as a voltage divider and to read the division, an analog pin will be used.

The purpose of the sensor node is to interpret the vacancy of its parking spot. It consists of the metal detecting coil, battery, Zigbee antenna, and microcontroller that manages the node's power supply and interfaces with the mesh network.

The microcontroller used on the sensor node is a TI SimpleLink Cortex-M device with built-in Zigbee functionality and multiple analog to digital converters. This part was chosen due to its integration with Zigbee and low-power architecture which are characteristics well suited for battery-powered wireless sensor nodes. Two analog to digital converters will be used on the microcontroller, one for monitoring the voltage of the battery and another for reading from the metal.

Each node will be identical from a hardware standpoint, i.e. any node can serve as a router, coordinator, or end-device. This serves two purposes: scalability and reliability. In regards to the former, if new parking spaces are added to a lot, then installing new sensor nodes is a trivial matter. In regards to the latter, if one device fails, then another can take over. This means that if a coordinator/end-device fail, then a sensor node in router configuration can take its place, and if a router fails, traffic can be rerouted through other nodes.

The desktop application will serve as the interface to visualize the data. The application will connect to a Zigbee node fashioned as a router. The other Zigbee nodes, fashioned as coordinators, will send signals so that the system can manage and interact with the number of available parking spots. The application will be designed to show a of the structure where the system is implemented and the available parking spots will be displayed. Additionally, the system will keep a running count of available spaces out of the total available spaces. Our mesh network will connect to our computer using a USB and functioning on UART communication protocol. The application will be designed using the Nana C++ library, a cross platform library for GUI programming. [11] This platform will allow us to seamlessly integrate with our chosen microcontroller and produce an interface available to the user.

There are a series of problems anticipated in the development of the parking system. First, it is anticipated that Zigbee will take some trial-and-error to get to a working state. Zigbee, while considered simpler than Bluetooth or WiFi, is still a complex standard with many facets, including Zigbee coordinators, Zigbee routers, and end devices.

The metal detector will also pose several problems, and it is anticipated that time will be needed after construction to tune and redesign it if necessary. For instance, its range or power consumption may be greater or less than initially designed for, and could need adjustment. In addition to the metal detector range and power, the battery consumption and replacement will be considered. For the purposes of the Capstone, a regular alkaline battery will be used. However, if time were not a limiting factor, a solar cell might have been used to power the system instead of batteries. If a complete board is designed rather than a Launchpad header, the project would be faced with the challenge of designing an antenna to broadcast and receive Zigbee signals. TI provides design documentation for this, but it would pose a challenge nonetheless due to the general lack of RF experience amongst the members of our group. Another potential issue is the interfacing between the sensor node network and the desktop application. There are few options to connect the mesh network and the desktop application. However, the USB option is the most feasible option.

The expected outcomes of the project include a successful interface between sensor nodes and host machine as well as proper calibration of ADC and successful detection of a vehicle. To build a successful project, PCB software such as KiCad will be needed[7]. To build the desktop application, a suitable GUI framework is needed, such as Juce. Another component which is integral to the project is Github and GitKraken. Both of which allow group members to check on the progress of the project and provide housekeeping details. To successfully test for the project outcomes, a vehicle will be needed for testing purposes.