Smart Parking Lot

Milestone 2

Andrew Lu, Finn Linderman, Luyao Han, Seungjun Cho

Specification

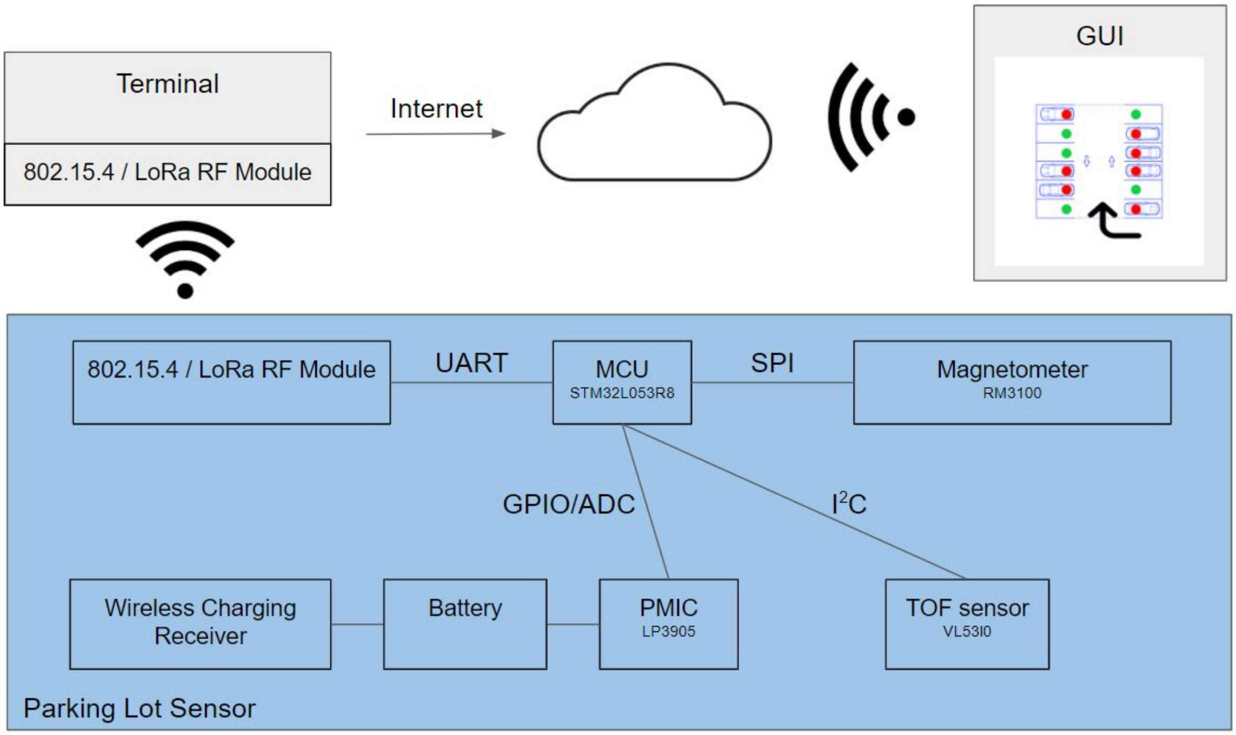
Our project is to prototype an autonomous platform used to navigate drivers to parking spot vacancies.

A. The driver will access available parking space information through a graphical user interface, ideally programmed to an app, or alternatively, a web page.
    a. When the driver arrives at the entrance, they will push the "Search for Space" button. The driver may be required to enter the floor they are interested in.
    b. The GUI will then present occupancy information of the entire floor using a sketch with all the parking spaces on that floor.
    c. The occupied spots are marked red, and the available spots are marked green.
    d. The driver will drive to the available spot and park the vehicle. No more interaction with the GUI is needed.
    e. (tentative) if the entire floor is rather huge and involves a lot of parking spaces, the GUI will actively direct the driver to the parking space by providing the shortest path with waypoints on the floor sketch.

B. The graphical user interface is actively updated, representing parking occupancy information received from a backend host.
    a. Parking Lot staff can utilize the GUI to overwrite occupancy information, allowing parking spots to be temporarily marked as "occupied" for maintenance.

C. The host communicates with the entire mesh network of sensors. Each sensor records either a vehicle's arrival or a vehicle's departure to the single parking space the sensor is located at. Upon each successful detection, a change of state will be transmitted to the host.
    a. Sensors will be installed in the ground.
    b. The host receives updates from each sensor in passive mode in daily usage.
    c. Parking Lot staff may request diagnostics from sensors, such as battery usage, daily count, functionality health check, and guidance light.

Block Diagram



Member Responsibilities

These will be the long-term responsibilities for each member:

- Andrew Lu - Parking Lot Terminal / Gateway
- Finn Linderman - Wireless Communication Firmware
- Luyao Han - Sensors / Sensor Firmware / PCB Design / Power Management
- Seungjun Cho - GUI / Path Finding Algorithms

Since we are still in the early stages of development, and since the group is split in terms of location, these are the short-term responsibilities:

- Finn Linderman and Luyao Han
  - Vehicle detection sensor research (i.e. Magnetometer)
  - PCB creation
- Andrew Lu and Seungjun Cho
  - Communication module research (i.e. LoRa vs. Zigbee)

Software Flow

- Parking Lot Sensor
  - Every 30 seconds, the parking lot sensor will use the magnetometer and time-of-flight sensor to detect the presence of a vehicle.
  - The parking lot sensor will report this to the floor's gateway using ZigBee or LoRa
  - When sending messages to the gateway / terminal application, the device will send its unique identifier, space status, and device health information.
  - This will be implemented using low-level C
- Gateway / Terminal Application
  - This device has a Zigbee or LoRa receiver that will accept all the incoming packets for devices on that floor.
  - When a packet is received, the module will communicate that to a small computer (like a Raspberry Pi) over a UART signal. The device will then propagate that information to an online database.
  - This will likely be implemented using a higher level programming language such as Python or JavaScript
- User Application
  - The user application reads the online database and converts the information from the devices into a user-interactable graphical interface.
  - This will take the form of either an Android app (Java) or a webpage (JavaScript / HTML).

List of Finalized Parts

Click on a part name to view its datasheet

Microcontroller - STM32L053R8

- This MCU was selected for it's ultra-low power consumption, using only 88 μA of power in active mode per MHz and 0.27 μA in standby mode.

Magnetometer - RM3100

- Magnetic field measurement range: ±800uT
- Low Power Consumption
- Small noise: 15 nT @ 200 cycle count
- High-resolution: 13 nT@200

PMIC - LP3905

- supplies 0- 600mA
- Two buck output, two LDO output supplying the two sensors, wireless module

TOF - VL53L1X

- Works with 3.3V voltage range
- Up to 80 cm of detection range

Wireless Charging Receiver - BQ51013B

- QFN package
- Compatible to WPC1.2 standard
- 5W

List of Tentative Parts

Click on a part name to view its datasheet

LoRa Radio - Adafruit RFM95W

- Works with 3.3V and 5V voltage range
- Approximately 2 km transmission radius.
- Operates on the 915 MHz band in compliance with American license free ISM bands
- +5 to +20 dBm transmission power
- Average peak 100 mA power consumption during transmission at +20 dBm
- Average ~30 mA power consumption during active listening.
- $25 including antenna and external antenna connector.

802.15.4 (Zigbee) RF Module - XBee

- Will be used in the parking lot sensor if choosing the Zigbee stack
- RF communication module capable of transmitting on the 2.4 GHz band according to the 802.15.4 network stack using the Zigbee protocol
- Communicates with the MCU with either UART, SPI, I^2C
- +8 dBm transmitting power
- 40 mA power consumption when active
- 2 μA power consumption when inactive
- $25 including antenna

MB1293C Development Board (STM32WB55)

- Will be used in the terminal / gateway if choosing the Zigbee stack
- USB Dongle with STM32WB55CGU6 MCU
    - Includes wireless coprocessor and onboard PCB antenna
- Full-size USB-A Male port that is used for programming and UART communication with another host device (likely to be a small computer)

List of Accomplishments for Fall Quarter

- Hardware
    - Microcontroller
        - Firmware and I/O assignment specific for peripherals
    - Established the method of vehicle detection
        - We originally compared magnetometer, hall effect, ultrasonic, infrared, and time-of-flight sensors.
        - Field test magnetometer with a vehicle
        - Magnetometer + Time of Flight
    - Established power management features
        - Due to size and power constraints, we may not be able to achieve our initial planned 5-10 years of battery life.
        - However, to mitigate this concern, we are looking into the use of wireless charging
        - We also have a dedicated PMIC
    - Research on Communication Modules
        - We initially researched all suitable methods of communication that could fit our needs (including WiFi, Zigbee, LoRa, and BLE), but settled on LoRa and Zigbee for low power consumption.
        - LoRa
            - Extremely long range while consuming low power, thanks to using a lower bandwidth.
            - The shorter bandwidth makes transmission of larger packets less efficient.
            - However, this long range has the potential to cause problems with packet loss.
        - Zigbee
            - Much shorter range and more susceptible to interference, given that the 2.4 GHz band is commonly used.
            - However, protocol allows devices to be meshed together, reducing the need for long transmission ranges and therefore reducing interference between parking lot sensors.
- Software
    - Firmware written for the microcontroller
    - Since we are prioritizing hardware for now, we have not made progress on a user GUI

Plan for Winter Quarter

Weeks 1-2: PCB Verification / Final Decision on Transmission and Communication Hardware

The PCB will be manufactured during the winter break. The first step after winter break is to do PCB verification. Depending on verification, the schedule for the quarter will be dictated. The transmitter, PMIC and sensor will all be tested, and checked to see if they are able to communicate amongst each other. Lastly, the research for Zigbee meshing will be completed over the break, so we can move forward with the design.

Weeks 2-5: Rework Firmware, Define Message Protocol

After the PCB verification is finished, our firmware will need to be reworked. We also need to standardize our transmission protocols in order to make sure that there is no interference, and the information that is being transmitted is accurate. On the software side, we will start to define the message structure for transmissions between the parking lot devices and gateway, as well as the database structure for storing sensor data.

Week 6+: Start Software

After the PCB and firmware have been finished, we can actually begin to write the software for the parking algorithm and GUI. We will first start by creating a GUI for displaying the available parking spaces and testing to ensure robustness once it is scaled up. Once this is complete, we can start building the path finding algorithm for navigating drivers to empty parking spaces using the app.

Week 8+: Extensive Testing in Parking Lot

Once all of the above is complete, we will do extensive testing in an actual parking structure, likely UCSB’s Parking Lot 10.