CS362 Project

SpotterSense

Parking Lot Availability Monitoring

Jason Liang - jlian21
Javid Uddin - muddi7

SpotterSense

Project Abstract

SpotterSense is an intelligent parking lot system that monitors and displays real-time parking spot availability. By using Arduinos with ultrasonic sensors and Bluetooth communication, SpotterSense provides accurate and updated parking spot counts for users. One Arduino acts as a central hub that communicates with (an)other Arduino(s) equipped with sensors to detect vehicle presence. This setup offers a seamless and feasible solution by providing real-time availability data and significantly improving the user parking experience.

Project Idea

The overall idea of our project is to create a system to **track the availability of individual parking spots within a parking lot**. Parking availability within any parking lot has always been very ambiguous and scattered, leaving drivers to spend a significant amount of their time searching for open spots, perhaps even on multiple floors.

SpotterSense will address this issue by displaying real-time availability to alert drivers directly to available spots, saving time and reducing frustration.

SpotterSense

AVAILABLE
SENSOR
SENSOR

Smart Parking Monitoring System

I/O Devices

Sensor Arduino (Client)

- **Ultrasonic sensors** - for sensing distance
- **RGB LED** - to help drivers gauge distance
- **Buzzer** - to help drivers gauge distance
- **HC-05 Bluetooth module** - to facilitate communication

Central Arduino (Server)

- **LCD** - to relay availability information to drivers
- **Buttons** - for drivers to switch the shown parking spot on the LCD
- **HC-05 Bluetooth module** - to facilitate communication

Process and Roles

We first began working on the project by coming up with the circuitry of the sensor and central Arduinos, along with the control flow of each side.

Then we each assigned ourselves either the sensor or central Arduino to work on

- **Central → Javid U.**
- **Sensor → Jason L.**

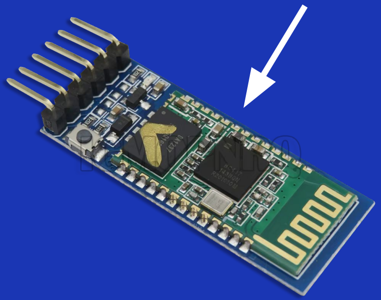
Once both halves were done, we began on the communication component. We decided how data needs to be sent and at what time interval it should be taking place.

Communication

Communication between sensor Arduinos and the central Arduino is **performed through Bluetooth**. Using two external HC-05 modules, we were able to successfully send status updates from the sensor Arduino to the central Arduino.

Each sensor Arduino will send a simple string containing **a spot number and a status** to the central Arduino:

- "1:0" would translate to "Spot #1 is **available**."
- Likewise, "2:1" would translate to "Spot #2 is **unavailable**."



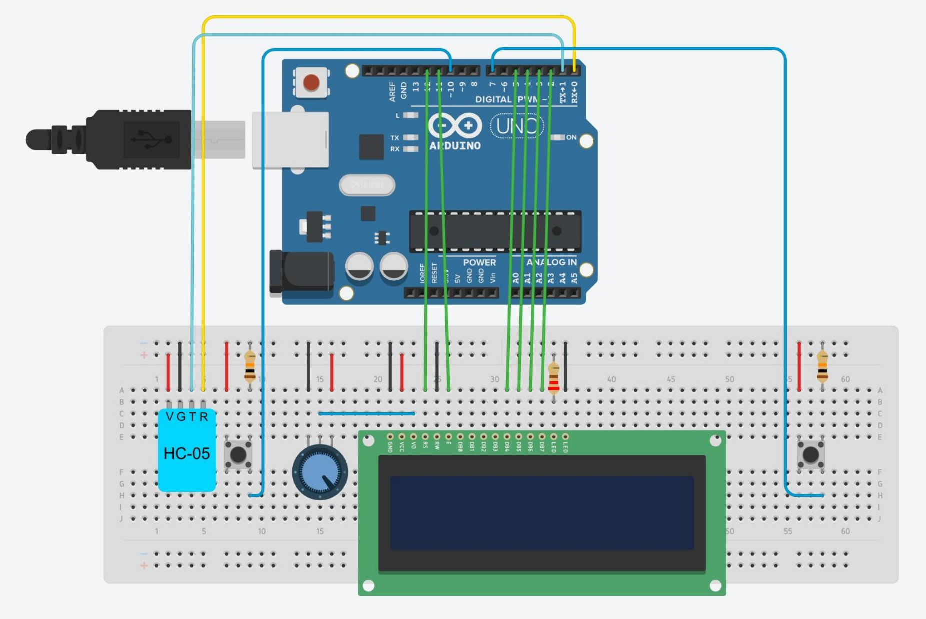
HC-05 BT Module

Original Work

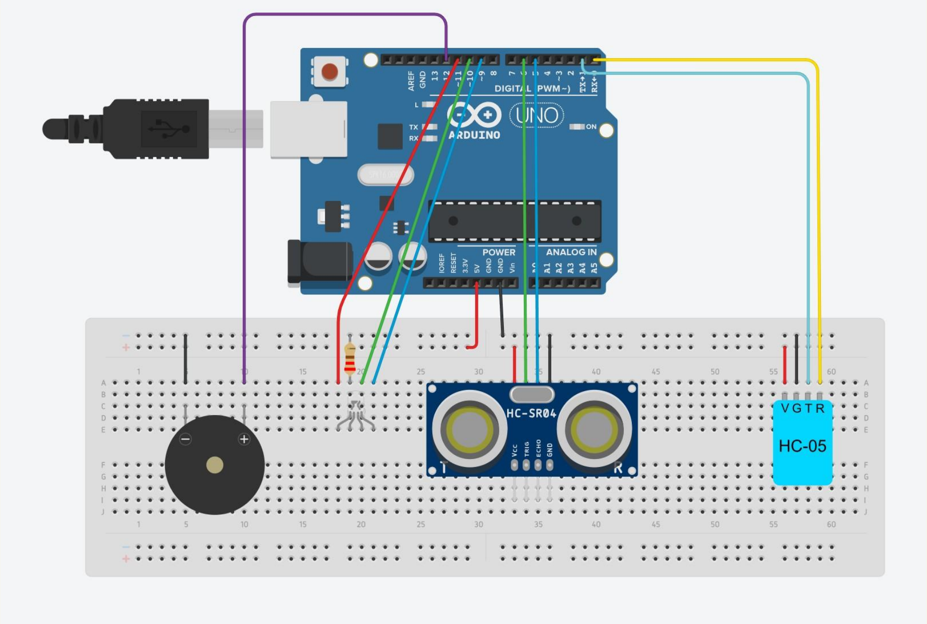
The original work we’re attempting to implement for SpotterSense is a **parking guide** to help drivers park their cars.

- **Using the ultrasonic sensors, we can help drivers gauge the distance their car is away from the curb.** By measuring the distance using the sensors, we can provide feedback back to the driver using a **combination of colors on the RGB LED and a buzzer.**
- When the car is within specific ranges, the **LED will light up a different color** and the **buzzer will have a different interval (intensity)**.

Central Arduino



Sensor Arduino



Successes

The initial plan of using a ultrasonic sensor to detect the distance of objects has worked pretty well:

1. We're able to tell when an object is within 5 to 6 feet of the sensor. In our testing, through the LED and buzzer, we're able to get the **sensor to detect changes in distance within 1 second of moving**.
2. Once we were able to get the bluetooth communication working, we were also able to successfully **parse the data sent** from the sensor Arduino and **display that information to the LCD** quite easily. We were also able to implement user control of the LCD as well.

Problems Encountered

Through to a lack of knowledge and experience with bluetooth, we encountered significant trouble trying to use the HC-05 bluetooth modules.

1. **First there was an issue of pairing the modules.** Then we were unable to get the modules to communicate even after they have successfully paired as a result of incorrect code.
2. We also had troubles with the **accuracy or reliability of the ultrasonic sensors**. In our testing, once past the 6-7 feet range, the sensor became unreliable in its detection. We decided to limit the effective range of our project to within 6 feet as a result.