SCHOOL OF COMPUTER SCIENCE & ENGINEERING
Sacred Heart University

Easy Park

Nelson Surico, Bryan Powell, Iburahim Fisiru, John Lorusso

D. Cenk Erdil, PhD; Tolga Kaya, PhD
Sacred Heart University

SCHOOL OF COMPUTER SCIENCE & ENGINEERING
Sacred Heart University

Abstract

At one point or another, we all had the unfavorable experience of driving around a parking lot searching for an empty parking spot. God forbid you are in a rush. This experience is not only frustrating but also very stressful. This experience is slightly better for indoor parking garages that tell you the number of spots available but those type of parking garages aren't located everywhere. Let's say that every parking lot garage had some sort of indication off the number of spots available, we will still have the same dilemma, there is no direction being given to an empty parking spot. We present to you Easy Park. This app will be our solution to the indoor parking lot dilemma. This app will direct users to empty parking spots within the indoor parking garage and this will be achieved through the use of ultrasonic sensors.

Introduction

One time or another you're going to run into a problem in a parking lot or garage where you cannot seem to find a parking space. Especially for many universities, the process of getting to the lot and finding a parking space within a reasonable amount of time can be a daunting task. Many times you will find yourself driving around in circles throughout the parking garage looking for that seemingly nonexistent free spot. There needs to be an easier way to let people know what spots are available to them without them driving all over the lot in search of one.

The main problem that we are looking at is how can we minimize the amount of time people spend looking for an empty spot and the amount of congestion within the lots or garages? Having some sort of application that allows a user to see what spots are currently available to them can decrease so much confusion when finding parking and ultimately lead to much greater satisfaction within the community. For our design, we looked to create a set up involving sending ultrasonic sensor data from an Arduino micro controller to our own application by way of bluetooth connection.

Implementation - 1

We originally had the idea of using this application for an entire parking garage. Considering this approach we soon realized that this was not feasible for our senior project. It would cost us way too much and probably take way too long for us to implement. Due to this, we decided to scrap our initial idea and took a different approach to the design. We thought it would be better to break it down to a smaller scale. We agreed upon the idea to use 4 parking spots and 4 ultrasonic sensors to better display our concept.

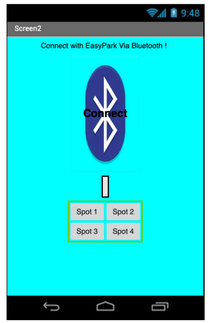
For the application, we picked a basic design for our start up screen. We were not that familiar with coding applications, so we chose to go with MIT App Inventor 2. This provided an easy to use structure and platform for us to begin coding our application. Upon opening the app, you are given the choice of logging in, resetting your password, or creating a new account. If you want to create a new account you will be taken to a new page where you can enter all relevant information. Choosing to reset your password will take you to a separate screen for you to create a new password for your account.

All of these pages also provide extensive validation to ensure no mistakes happen and everything runs smoothly. After logging into your account you will be asked to pair your device to the bluetooth module connected to our parking sensors. Upon successful connection, you will be shown a grid that acts as an overview of the current parking spots. If one of the squares is colored green, this means the parking spot is available and if the square is red, that means the spot is currently occupied.

Login Screen

Registration Screen

Password Reset Screen



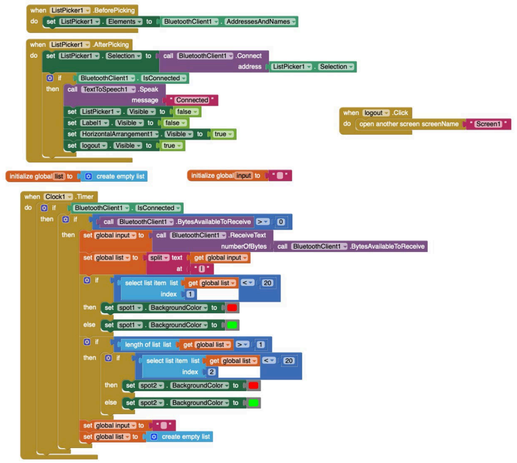
Bluetooth & Grid Screen

Implementation - 2

When it came to the design of the sensors, we first decided on designing it with just one sensor in use. We set the ultrasonic sensor up on our breadboard and then connected all the wires to their proper places on our Arduino microcontroller. Once we were able to determine the sensor was working correctly with our micro controller and code, we could add the bluetooth module.

After we determined the size of the square grid on our application, we looked at how we wanted to design the final look of the sensors and microcontroller. We got the idea to use a rather long breadboard, with two sensors facing out on the ends of the breadboard. We hooked up the bluetooth module in the middle of our breadboard, with our microcontroller sitting close by.

To test our sensors along with the new application, we set up a small test demo that allowed us to simulate a real life scenario. For our demo we used pieces of tape on the ground to symbolize the lines of the parking spots. We then set up the sensors in the middle of all four spots so that the sensors lined up with their corresponding parking spots. To take the place of real life cars, we used two shoe boxes that were big enough to give us a proper reading from the sensors. To test out our application, we parked one of the shoe boxes in each spot, showing that the spots appear as unavailable upon entering the app.



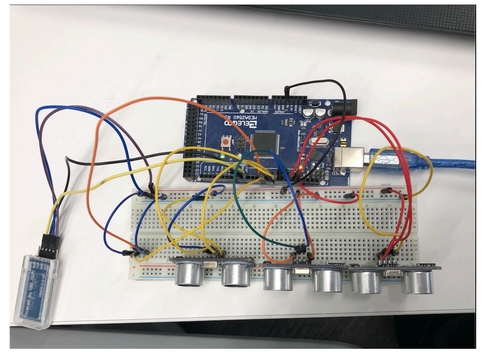
MIT App Inventor Block Code

```
void loop() {
   a=sr04.Distance();
// Serial.print("A :");
   Serial.print (round(a/2.54));
   Serial.print("I");
 //  Serial.println();
 //delay(1000);

//  if (a/100 < 3.048){   // 3.9624 is 13 feet
(ceiling height) coverted to meters
//   Serial.print("Space 1 is Occupied, ");
//   }
//   else {
//   Serial.print("Space 1 is Empty, ");
//   }


   b=sr05.Distance();
 //  Serial.print("B :");
   Serial.print(round(b/2.54));
   Serial.print("I");
 //   Serial.println();
 delay(1500);
```

Arduino Code for Sending Data to App



Future Work

Looking towards the future, we would like to look at the possibility of implementing this idea on a larger scale than we currently have it. This was our original intention for the project but due to limitations and cost issues we were unable to do so. We would really like to get our idea working with an entire parking lot. Having the ability to see the availability of all the spots in a parking lot would be so beneficial to people and could cause a lot of parking related problems. If we looked back into that problem, we would also try implementing our idea for directing drivers. This seemed to be a good idea as the apps layout could be seen as confusing and this would hopefully decrease any problems users have in the lot. Essentially, we would like to fix any small, unresolved problems within our project, as well as deliver our idea on a much bigger scale. The last thing we plan to do is get our project out to the whole world. After putting our application on the Android store, we want to make all of our code and schematics available to anyone that might want to replicate our idea. It is a great idea to allow others access to our designs. They could potentially get help for whatever they are trying to implement or even possibly improve our own project. In the end we want to continue supporting our project, improving it anywhere we can, while also getting our idea out there to get people thinking about the possibilities.

References

ai2.Appinventor.mit.edu

"Arduino Bluetooth Basic Tutorial." *Arduino Project Hub*, create.arduino.cc/projecthub/mayooghgirish/arduino-bluetooth-basic-tutorial-d8b737.

"Home." *Arduino*, www.arduino.cc/.

"How To Receive Multiple Sensor Readings From Your ARDUINO Into Your MIT APP INVENTOR 2 APP." *Z*, 29 Mar. 2019, zhutarduino.wordpress.com/arduino-projects/how-to-receive-multiple-sensor-readings-from-your-arduino-into-your-mit-app-inventor-2-app/.

Pendergast, Robert L. "Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino." *Random Nerd Tutorials*, 2 Apr. 2019, randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/.

1.

EASYPARK