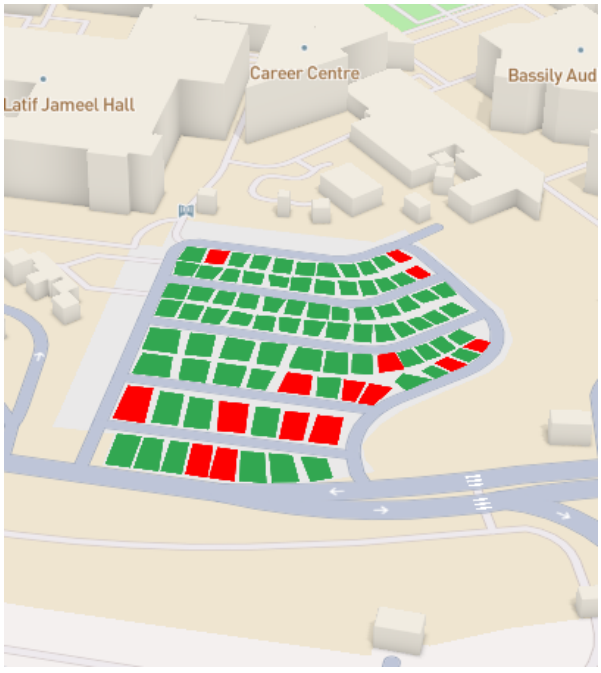
Efficient Parking using Maps and Sensor data

Aly Youssef



Outline

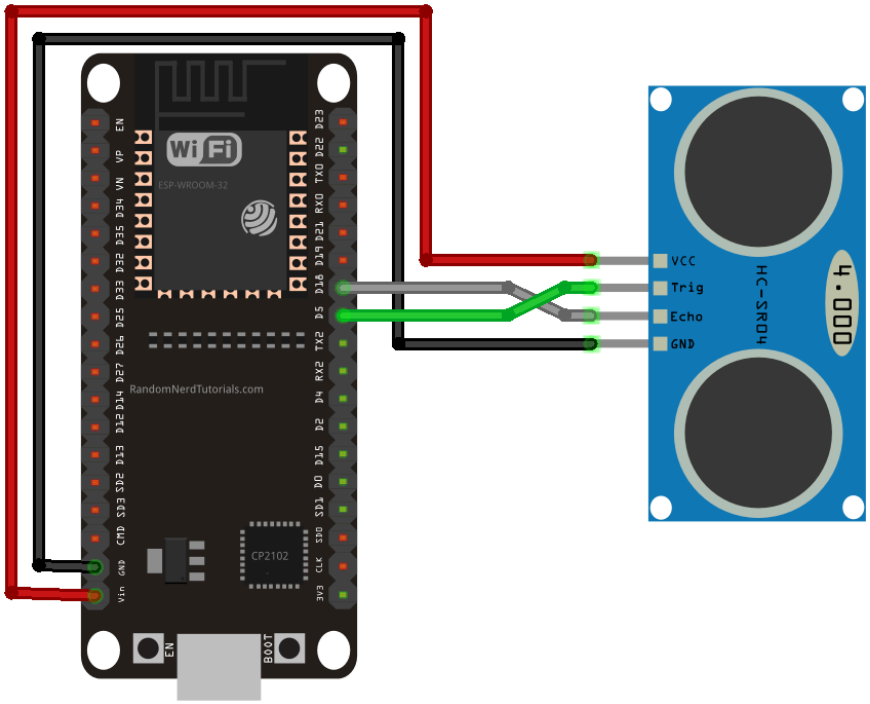
Finding parking spots on campus is inefficient due to imperfect information i.e people don't know where the free parking spots are so they go around looking for free parking spots which takes a lot of time and congests the parking as people are looking for spots. My approach eliminates this problem by using sensors on each parking spot connected to wifi which is connected to a database which updates an interactive map to show the free spots.

Hardware

For hardware and sensors we will use the esp32 wifi microcontroller and the HC-SR04 ultrasonic sensor. The esp32 has at least 18 digital pins so over nine sensors can be connected to one microcontroller. Assuming 3,000 parking spots we would need 334 esp32's (3,000 / 9 = 334) and 3,000 ultrasonic sensors. Using prices from alibaba for these specific quantities we'd get each esp32 for around 145 EGP and each ultrasonic sensor for around 30 EGP respectively. Adding up the totals 145 * 334 = 48430 and 30 * 3000 = 90000 resulting in 138,430 EGP. We will count in an additional 10,000 EGP for power cables to power these microcontrollers resulting in 148,430 EGP.

https://www.alibaba.com/product-detail/ESP32-Development-Board-CP2102-CH340C-NodeMCU_1600867844607.html

https://www.alibaba.com/product-detail/Ultrasonic-Transmitter-Receiver-Sensor-HC-SR04_1601119495192.html



Credit: randomnerdtutorials.com

In this schematic only one sensor is connected but you can see that there's space for a lot more sensors. As for power consumption the esp32 averages at 0.5W and each sensor uses 0.075W so (0.075 * 9) + 0.5 = 1.2W. So for 334 of these boards it will be 400W which is tiny considering that it's powering a parking system for 3,000 people.

Software

To make the map as intuitive and easy to use as possible I used mapbox (https://www.mapbox.com/) as the base layer for the map. It offers control identical to google

maps which users are used to but offers additional features like 3d maps that look cool and make the user more likely to use it. To get the data of the parking spots YOLO, an object detection system can be used. We will give it data from google maps satellite images of parking areas around campus and in return we'll get the latitude and longitude of each parking spot which we can convert into GeoJSON data (the type of data used to add layers to mapbox) and add the occupied/free layer (green for free and red for occupied) on the map.

Labeled Parking Spots on Campus to be used for object detection