Smart Parking System

Mounika Manchala, Puneet Bhushan, Sai Grandhi, Ruchi Patel

Advisor

Behnam Dezfouli

June 2018

Introduction

Problem Statement

"Parking and traffic congestion are constant sources of frustration for drivers, merchants, employers and public officials in most cities around the world. It is no surprise that smart parking services are top of mind with public officials, city information technology (IT) and innovation executives when planning smart cities."

IoT based Smart Parking System - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/Smart-Parking-System-Talking-of-the-above-mentioned-figure-it-depicts-a-parking-area_fig1_303842610 [accessed 13 Jun, 2018]

Motivation

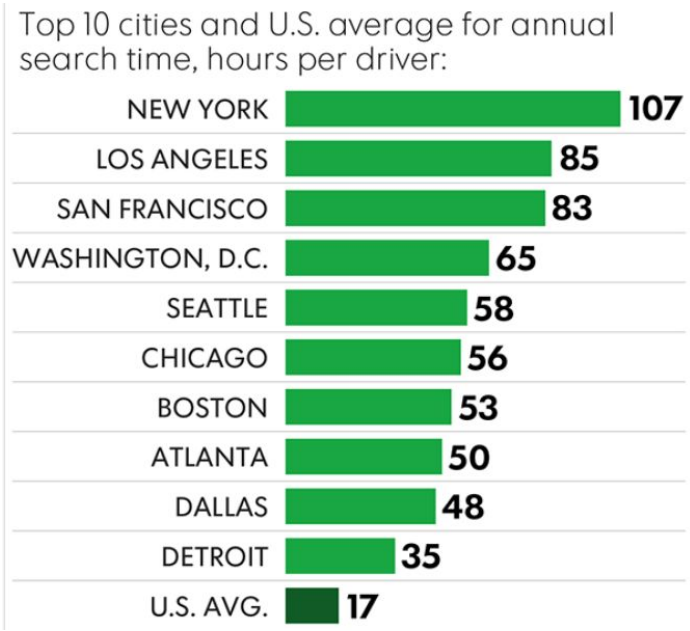
The current system followed for car-parking is:

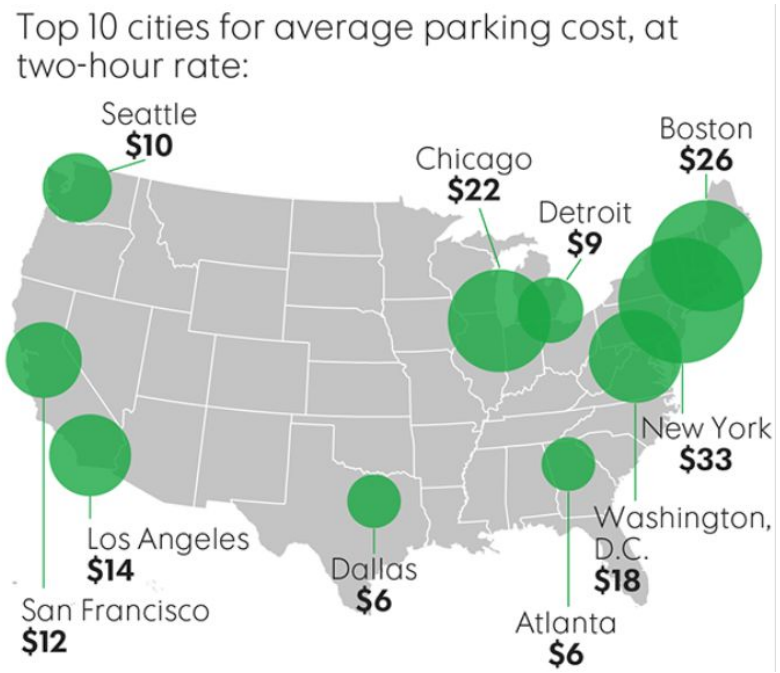
1. A car enters a car parking area.
2. The driver checks for an available slot either through help of a parking assistant or himself maneuvering over all the parking area.
3. If there is no parking space available, the car has to go back and search for a parking slot in some other parking area.
4. This routine drill, faced on daily basis, wastes a lot of time, energy and money on the part of the driver as with the increase in car parking availability, costs go up.

Parking is Painful

Parking spaces is a big issue in urban cities.

Motorists spend an average time of 17 hours and about $97 searching for places to park.





Existing Solutions

- Companies providing Smart Parking Solutions:
  - Cisco Kinetic for Cities Parking
  - Streetline:

- Apps/Platforms
  - ParkMe
  - ParkNow

SmartPark

- The proposed smart car parking system will overcome the challenges and difficulties faced in the current conventional parking system.
- It saves the time and effort that is put into this task and also saves fuel wastage.
- Smart parking system will be a revolutionary change in the parking system which is a dire requirement in urban societies with increasing number of automobiles.
- It could help bring order in current car parking system make it really simple and easy.

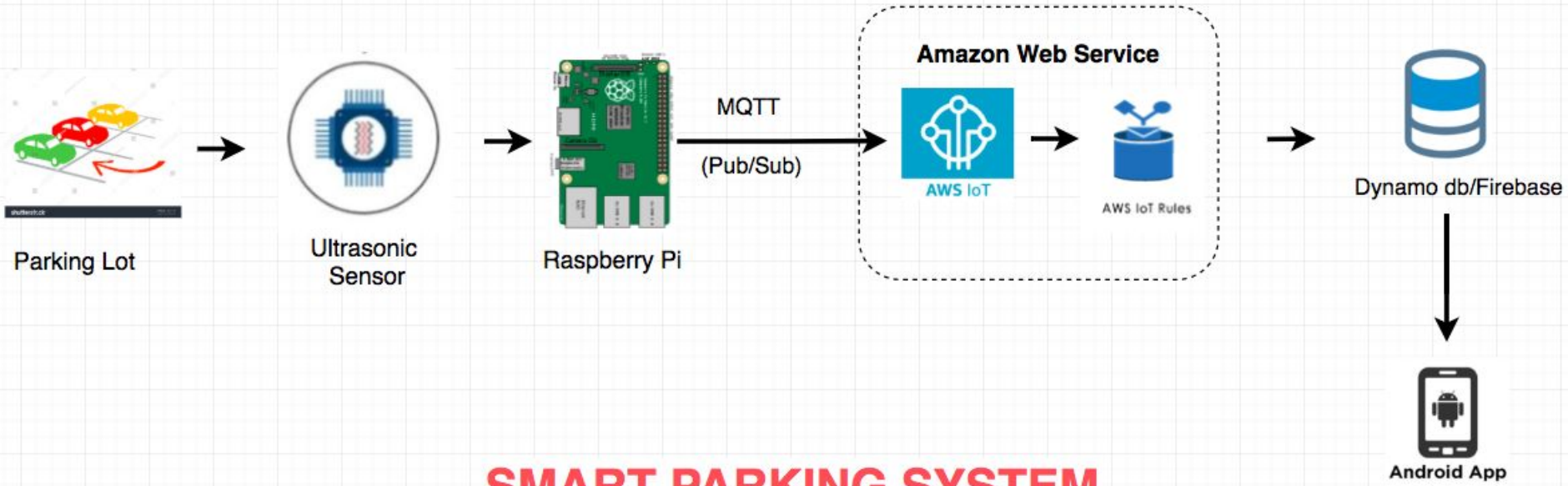
SmartPark

- **SmartPark** is a smart parking solution that uses several components of IoT
- Our system allows users to view available parking slots in real time and also you can reserve a slot.
- Technologies used to make this possible:
  - Hardware: HC-SR04 ultrasonic sensor, Raspberry Pi 3, Android Phone
  - Software:
    - PiGPIO
    - AWS IoT : Acts as Virtual Device in cloud for interaction
    - AWS Lambda : Communicating from AWS IoT to FireBase
    - AWS Cognito : Used for Sign In and Sign Up
    - AWS DynamoDB : Storing Sensor Data
    - Google Firebase : Realtime communication for Mobile App
  - Communication: MQTT Protocol in AWS IoT

Design and Development

- Connecting the ultrasonic sensor with the RaspberryPi 3
- C code using PiGPIO library to communicate with ultrasonic sensor and collect data.
- Send data to AWS IoT broker using a MQTT client on Raspberry Pi written in NodeJS using AWSIoTNodeSDK
- An AWSIoTLambda pushes retrieved data to DynamoDB and Firebase.
- Firebase is connected to an Android application that the user can use to see if parking is available and also he can reserve a parking lot.

SYSTEM DIAGRAM



SMART PARKING SYSTEM

Why Ultrasonic Sensor?

1. Inaudible to humans and therefore undetectable by users.
2. Ultrasound waves can be produced with high directivity.
3. They have lower propagation speed than light or radio waves, providing much accuracy while measuring distances.

HR-SC04

The ultrasonic sensor uses SONAR to determine the distance to an object.

- Used to measure a wide range of 2-400cm.
- Effectual measuring angle is 30 degrees.
- 4 Pin Module:

  Vcc : +5V

  TRIG : Trigger (Input)

  ECHO : Output (Output)

  GND : Ground

- Trigger input pulse width is 10 microseconds

which sends a 8 cycle sonic burst travelling at a speed ot sound.

HR-SC04 Working

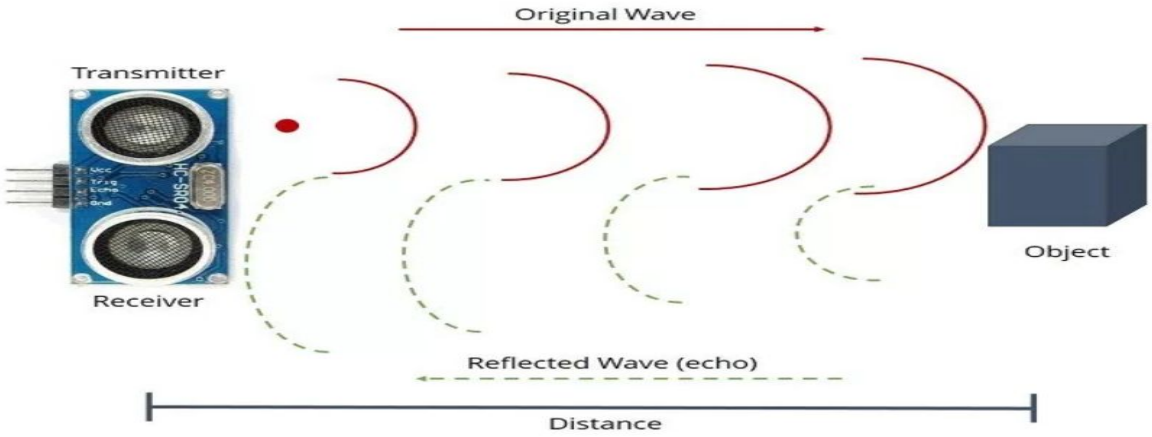
The sensor has a transmitter(trig) and a receiver(echo).

The transmitter transmits a ultrasonic sound wave of frequency 40KHz that has high pitch that humans cannot hear.

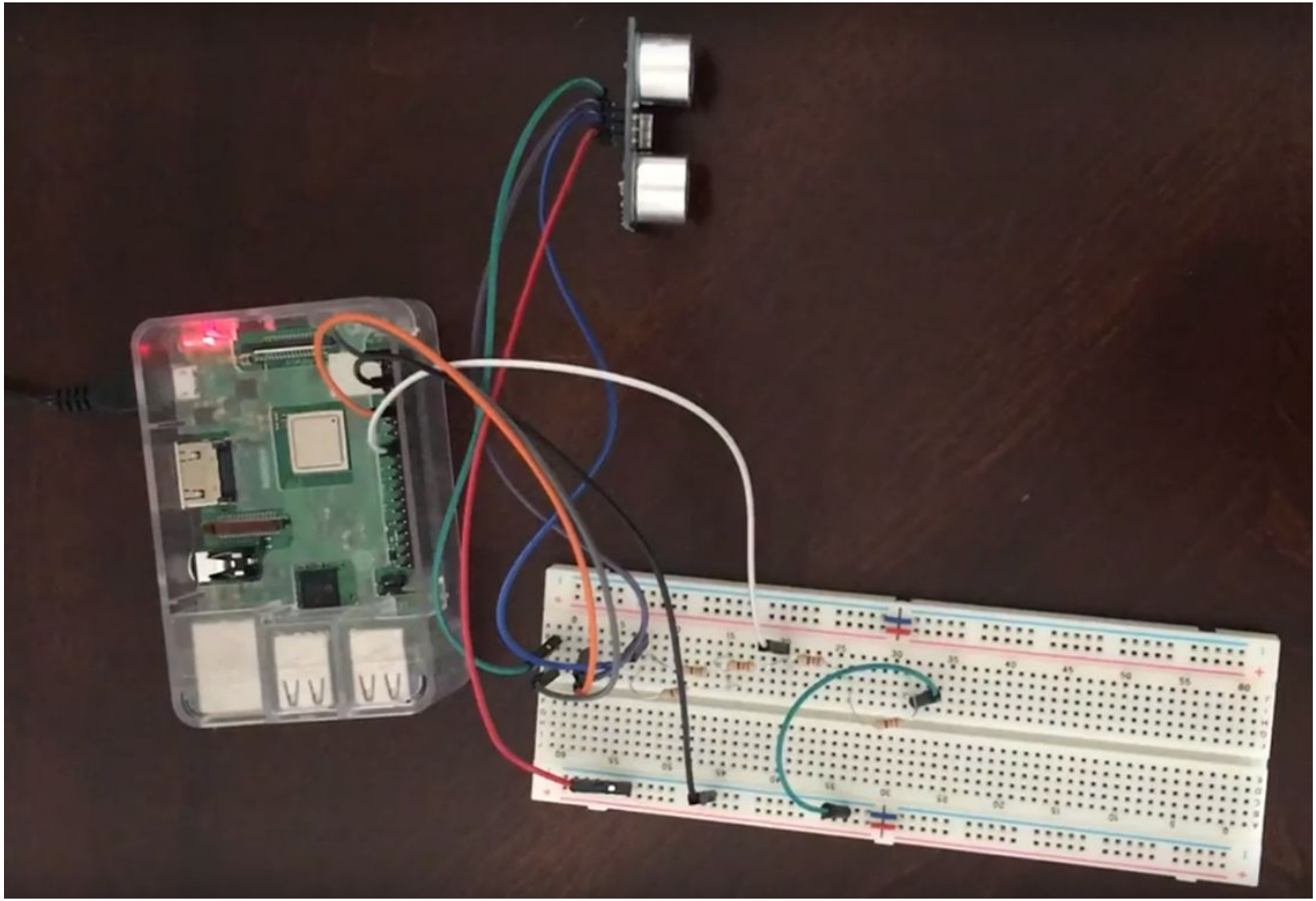
The wave gets reflected when it founds an object.

The receiver receives it.

Considering the travel time and speed of sound, we calculate distance.



Circuit Diagram



Evaluation

Testing

- Distance sensor: By placing the objects in the proximity of the sensor.
- For App: Placing and removing car near the sensor's proximity, this will change the availability of the parking spots.
- Reserve Feature: In the view parking section, we can see the list of reserved slots and available slots.

Project Demo Link

“ https://youtu.be/AeUGq5LNJO4”

Comparison

Some system uses only IR sensors for the smart car parking which only tells about the obstacle and are not very good in measuring distances.

Some system have used RPi camera just to check and verify the parked vehicle which may be an overhead since it involves lot of cost and need huge data storing capability.

A graph showing distance(mm) vs time(us)

Challenges Faced

- Coordinating the sensors and other hardware devices was a tedious task.
- Connecting the AWS Core IoT to Dynamodb took a lot of time because of tedious documentation. The reason was because AWS has IAM roles, which needed a service-role level access to connect to the Database.
- Integration of data from sensor was tough because of communicating between two different languages.
- Changes were not recognized on the Mobile App when Car gets parked/removed. So, we have to try using the Realtime database which was a big learning from the challenge we faced.
- We used many technologies ranging from C, Java, NodeJS, Javascript and Android SDK.

Future Prospects

- The parking slots can be distributed into different categories such as that for handicapped, electric vehicles and general parking to give a better estimate of the available number of slots depending on personal preference.

- Vehicle verification in case of remote booking can be introduced using number plate verification by installing a camera or sharing an OTP that the user can enter in the kiosk in a parking area.

References

1. Computer Networking: A Top Down Approach, 7th edition Jim Kurose, Keith Ross, Pearson/Addison Wesley, April 2016
2. Dierks, Tim. "The transport layer security (TLS) protocol version 1.2." (2008).
3. Rescorla, Eric. *SSL and TLS: designing and building secure systems*. Vol. 1. Reading: Addison-Wesley, 2001.
4. Stallings, William, and Mohit P. Tahiliani. *Cryptography and network security: principles and practice*. Vol. 6. London: Pearson, 2014.
5. Blundell-http://blog.blundellapps.co.uk
6. Aws Documentation - https://aws.amazon.com/documentation/