To detect tracking system of Parking Vehicle by using Sensors and Arduino YUN Device

Kande Archana
Asst.Professor Department of CSE MLR Institute of Technology, Hyderabad, India

Dr. N. Sandhya
Professor Department of CSE VNR VJIET, Hyd, India

Dr. A Govardhan
Principal JNTU College of Engineering, Hyderabad, India

Abstract—In large public places and shopping malls, it can often be quite irritating for visitors to drive around in search of a vacant parking spot. During busy hours or on weekends, this can cause an additional lead time in getting in and out of the parking area. In this how to build a prototype to detect tracking system of Parking Vehicle by using Sensors and Arduino YUN Device. This application can assist the driver in reserving a parking space and the system can also automatically track billing based on the driver's actions when the vehicle pulls in or out of the space. By Vehicle Parking System is to eliminate the unnecessary frustration drivers experience as they waste priceless minutes circling parking garages looking for the optimal parking spot. In addition, traffic flow within the garage will be better regulated, creating a safer atmosphere for both drivers and edestrians.

Keywords— ArduinoYUN, Ultrasonic Sensors, Master Controller, Sensor Controllers and PMS

I. INTRODUCTION

Parking is one of those universal challenges most people can relate t parking causes congestion on our city streets, contributes to pollution, and causes motorist frustration in most cities around the globe. The ways to apply technology to tackle some of the most pressing issues facing our cities today, parking is ripe for innovation.

Ultrasonic sensors embedded in the pavement of the selected parking spaces detect when vehicles occupy or leave a space and communicate that raw information to Ultrasonic repeaters and gateways, which then channel the data through a cellular gateway to Ultrasonic cloud data center. Ultrasonic's advanced inference technology then processes the sensor data in the cloud to provide the availability of parking spaces. The intelligent network platform captures the data and publishes it via Ultrasonic's free smartphone application, Parker, which displays real-time availability of these on-street spaces as well as locations of off-street parking garages and lots. The technology is designed to not only help people find spaces, but also to help reduce traffic congestion and improve parking management. Information vital to parking administrators including occupancy, turnover, and length of stay, is then made available to municipal customers through ParkSight, Ultrasonic's web-based analytics application. In addition, Ultrasonic's award winning Guided Enforcement (GE) application will now include real time images of the cars in violation, allowing for more accurate enforcement management.

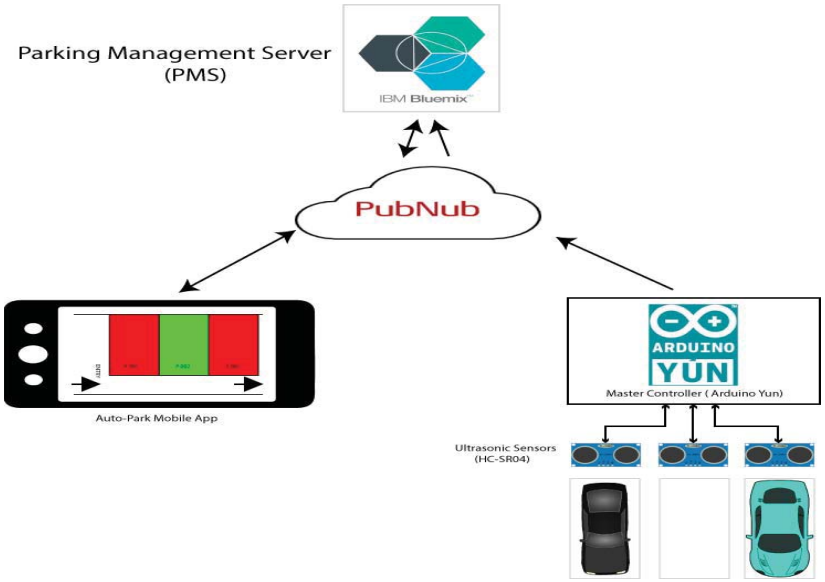
Vehicle Parking System is to reduce the unnecessary frustration drivers experience as they waste priceless minutes circling parking garages looking for the optimal parking spot. In addition, traffic flow within the garage will be better regulated, creating a safer atmosphere for both drivers and edestrians.

There are three components of this application

- *Parking Management Server (PMS)* – monitors all the parking spaces and also manages metering and billing for all users.

- *IoT hardware platform* – connects the parking spaces to PMS and also detects the presence or absence of a vehicle.
- *Mobile App* – offers an easy interface to assist the driver to find a vacant space and manage his parking usage and billing.

The hardware platform is powered by Arduino Yun and uses ultrasonic sensors to detect the presence or absence of a vehicle in a parking space. The PMS is implemented as an application server running on Python. It keeps track of all the devices, and manages billing and reservations. The Mobile App (also known as Auto Park) is a Cordova and JavaScript based Android app.



The PMS application server is hosted on the IBM Bluemix cloud platform and the entire communication between PMS to hardware and PMS to the mobile app is powered byPubNub's realtime Data Stream Network.

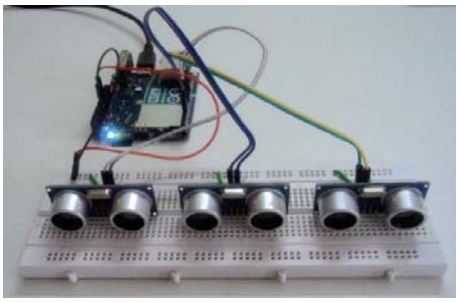
II. IMPLEMENTATION AND DESIGN

This project is a great DIY experiment for IoT enthusiasts. So if you are interested in trying it out, head over to GitHub for the complete source code of the bluemix-parking-meter project.
Refer the build instructions and readme file for detailed steps, right from configuring the hardware to hosting and running the application.
For hosting this application, you will need to create a Bluemix and PubNub account. Visit the IBM Bluemix signup page and PubNub add-on page to create your respective accounts. Both of the services offer a free tier account to play around with their offerings.

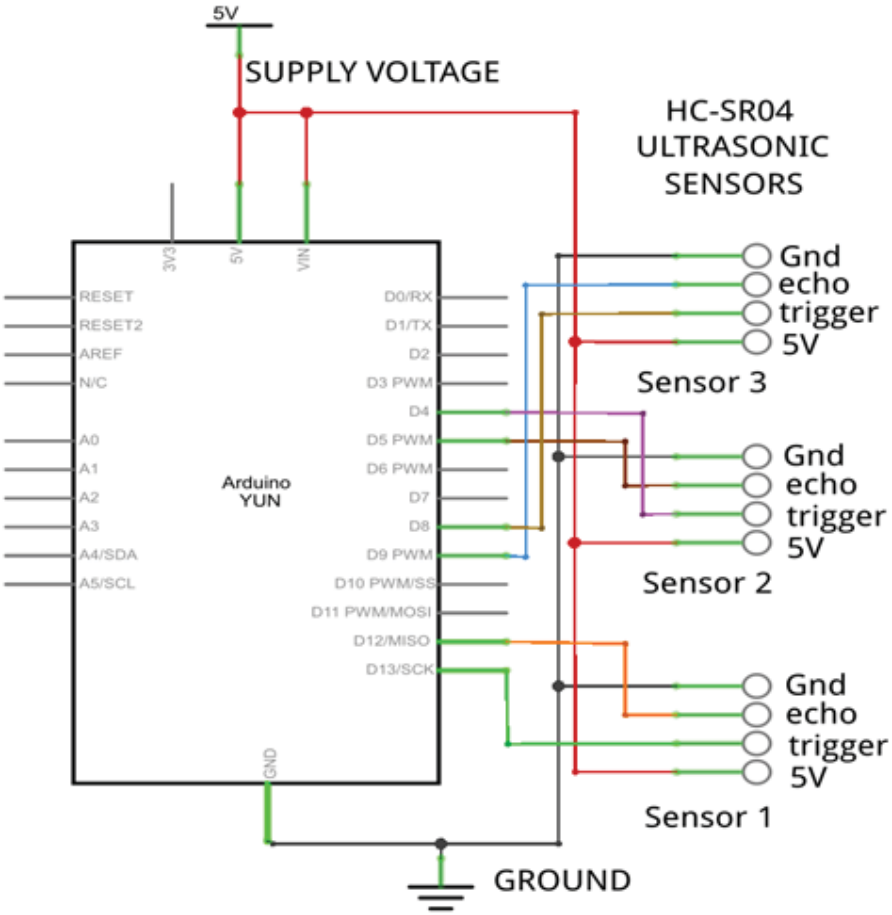
The following is the list of hardware components used for this project

- Arduino Yun
- HC-SR04 ultrasonic sensor (3 nos.)

A sample setup on a breadboard is shown below



And the associated schematic diagram for the hardware circuit is below



There are three functional parts of the hardware

- Master Controller – The Wi-Fi-enabled Arduino Yun acts as the master controller for controlling a few parking spaces. It periodically monitors and gets the status of each parking space within its jurisdiction, via ultrasonic sensors. It also interfaces with PMS via PubNub and publishes the parking status
- Sensor Controller – This is an internal component of Arduino Yun board, powered by ATMega32 chip. It directly interfaces with the sensors and runs a loop, every few seconds, to get the latest status of each sensor
- Ultrasonic Sensor – Three HC-SR04 sensors are used to simulate three parking spaces.

  The source code for hardware setup is available under yun_pubnub directory (for master controller) and device/hcsr04 directory (for sensor controller) in the GitHub repository.

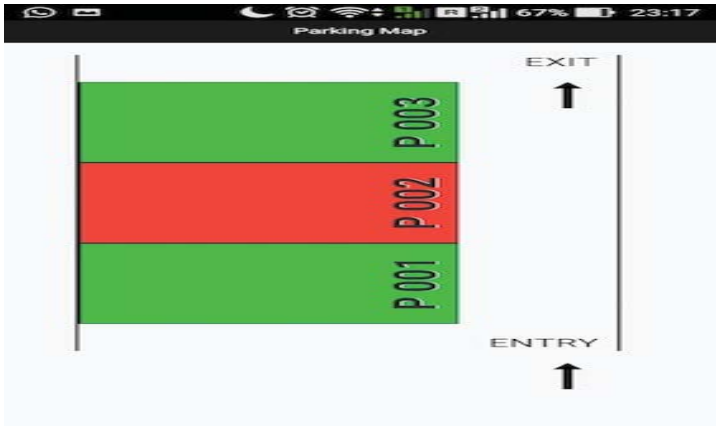
Parking Management Server (IBM Bluemix)

The PMS is written in Python and can be installed as a IBM Bluemix hosted service. IBM Bluemix provides the computing horsepower for the PMS to monitor hardware devices and manage parking metering and billing for the users. Additionally you will need to associate the PubNub add-on service with your IBM Bluemix account for the PMS to work with PubNub. Refer to the steps in README.md to understand how to setup and host a Python application under Bluemix with PubNub.

III. WORKING FUNCTIONS OF PARKING APP

Mobile App

The mobile app is a standard Cordova based Android app. It displays a map of the parking area with color coded parking spaces to assist the user in choosing a vacant space.

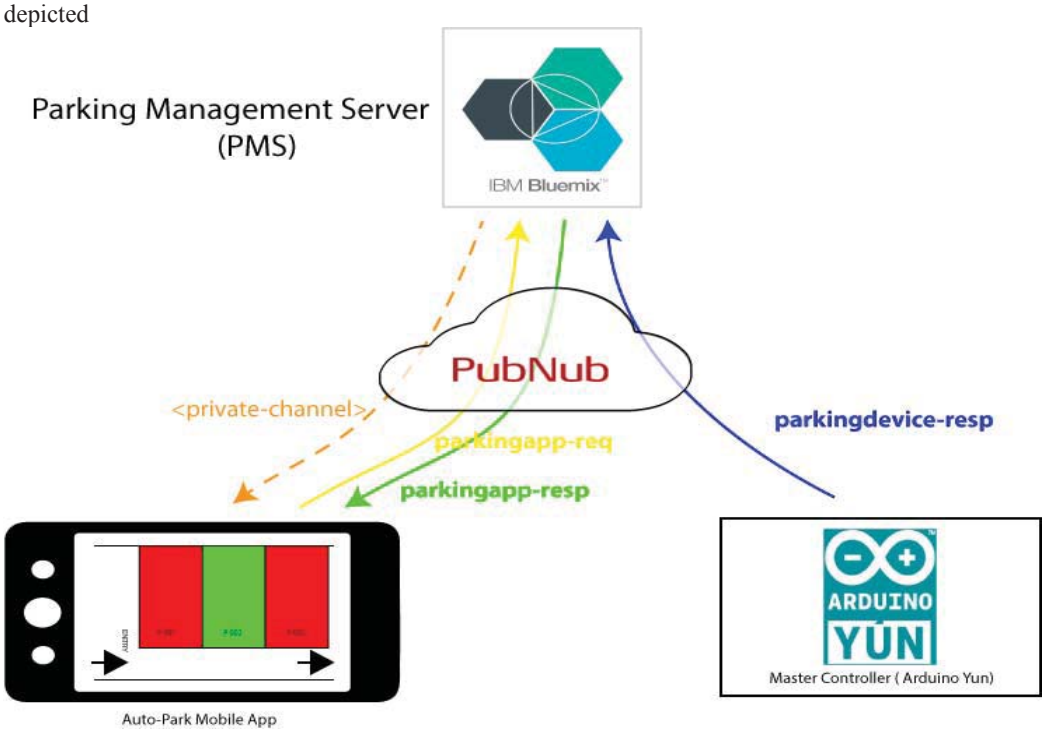


PubNub

PubNub acts as the communication middleware for the entire system. It provides a cloud-based realtime Data Stream Network which supports more than 70+ SDKs, such that it can enable any device to communicate with any other device on the Internet. This application uses three of PubNub's SDKs for all components to seamlessly communicate with each other. These are:

- Javascript SDK for the mobile app,
- Python SDK for PMS
- POSIX C SDK for the Arduino Yun

This application relies on several PubNub channels to enable communication between the components, as depicted below



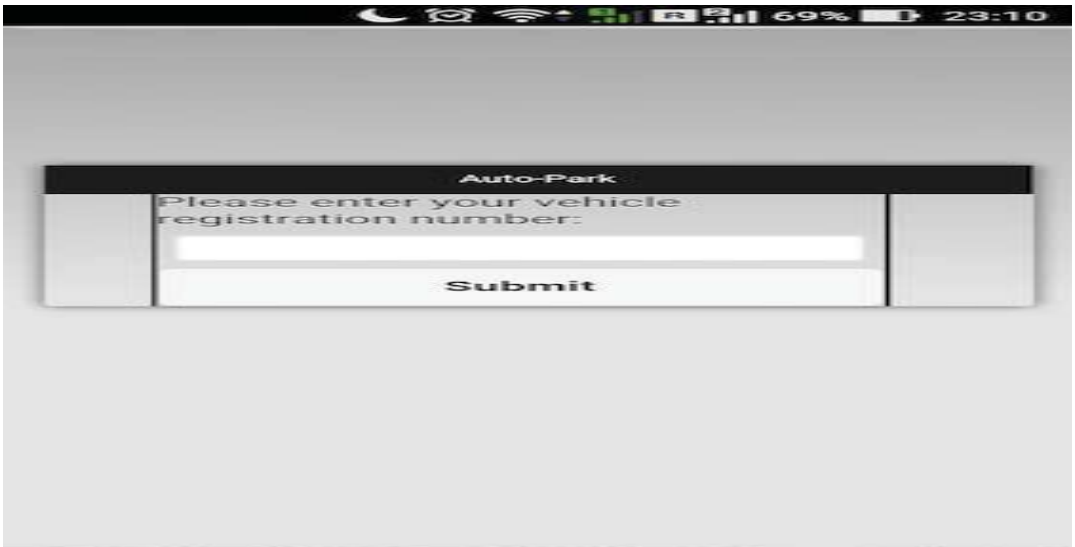
The <private-channel> in the above diagram refers to a dedicated channel between the PMS and one mobile app. All messages exchanged through the PubNub channels are in JSON format. For every mobile app requesting for parking reservation, the PMS initiates messages through this channel for that particular mobile app.

Installation of Mobile App

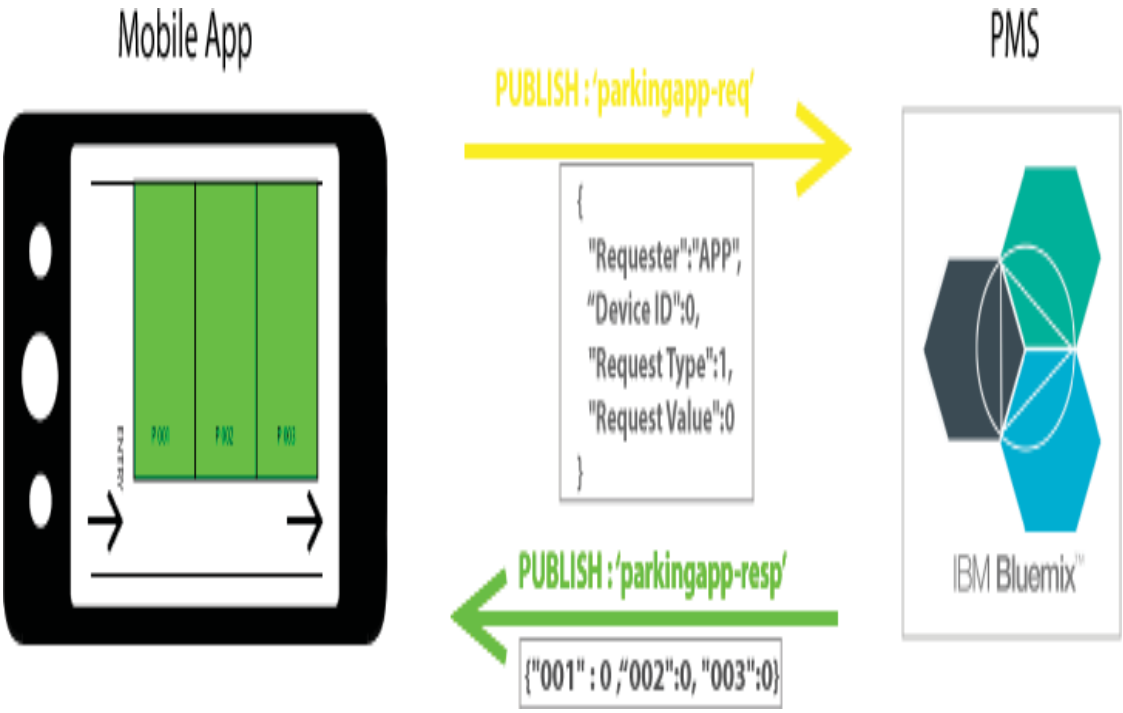
The complete operation of this system can be divided in following five scenarios.

App Initialization

When the mobile app is launched for the first time after installation, it asks for the license plate/registration number of the user's vehicle. This acts as a unique identifier for the PMS to track the app for billing purposes.

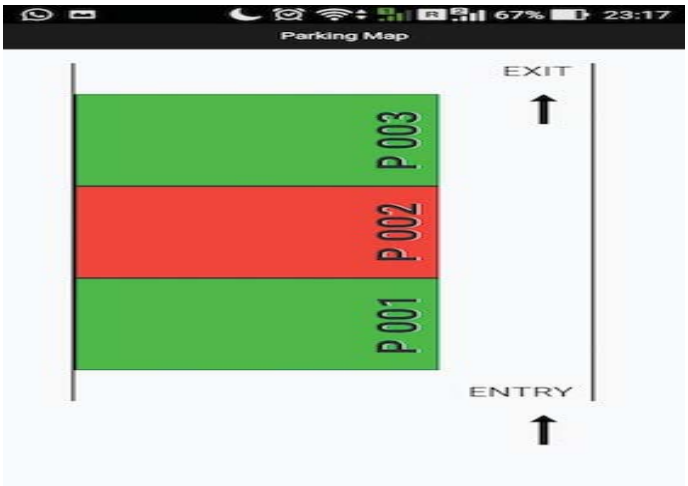


The above figure shows the app send a request to the PMS to get the status of all the parking spaces. This is used to display the map, where each space is identified with a slot number (001, 002 and 003). Here is how the message exchange takes place between the PMS and mobile app:



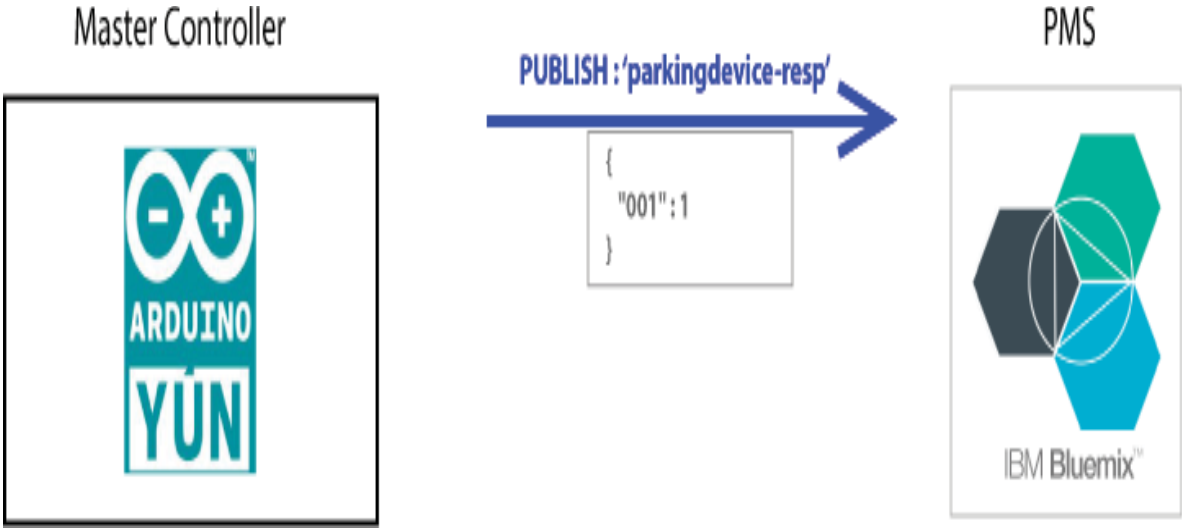
The above figure JSON request from the app contains a parameter RequestType with the value '1' to indicate a request for fetching the bulk status of all parking spaces. The JSON response from PMS contains the parking space's slot number as the parameter and its status as either '0' or '1' , to indicate that the space is either vacant or occupied.

The below given map display in mobile app, parking spaces which are currently vacant are indicated as green, whereas the ones that are occupied or reserved are indicated as red.



Device Status Update

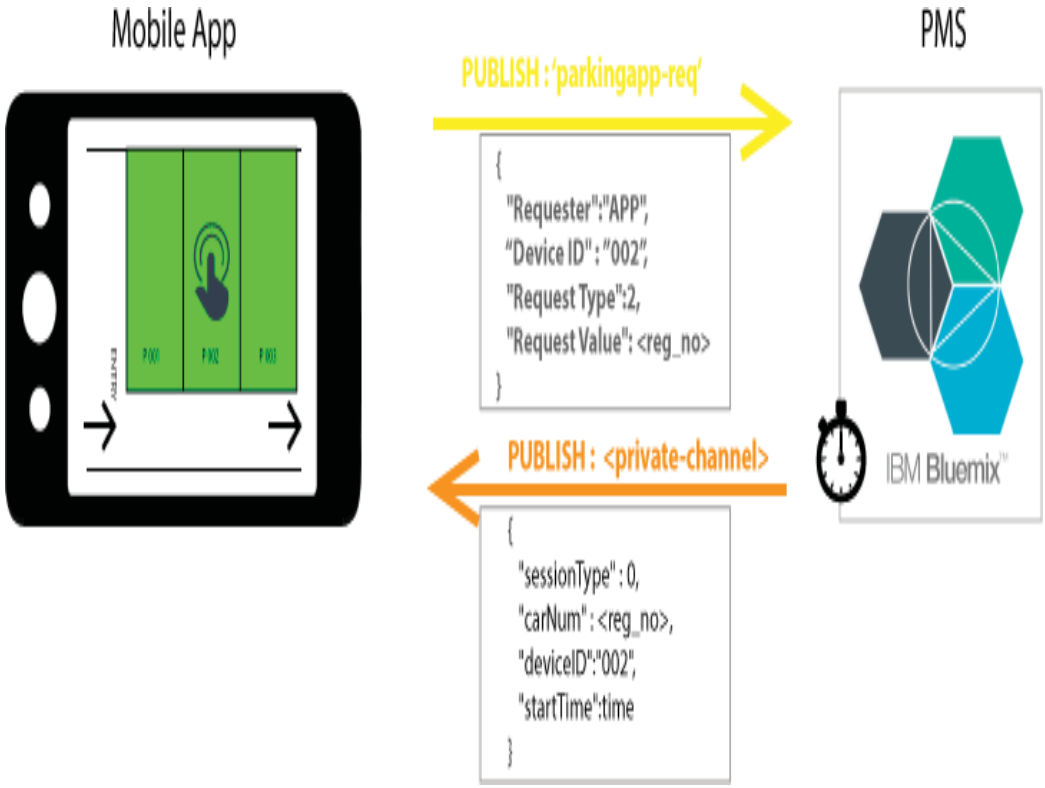
Whenever the parking space detects presence or absence of a vehicle, it immediately signals the PMS.



Here, the above diagram shows the 001 identifies the slot identification number of the space and the value '1' indicates that the parking space is occupied. Alternately, a value of '0' indicates that the space is vacant. The PMS also relays this information on a global PubNub channel parkingapp-resp so that all apps can update their parking map display.

Reservation request & Billing Start

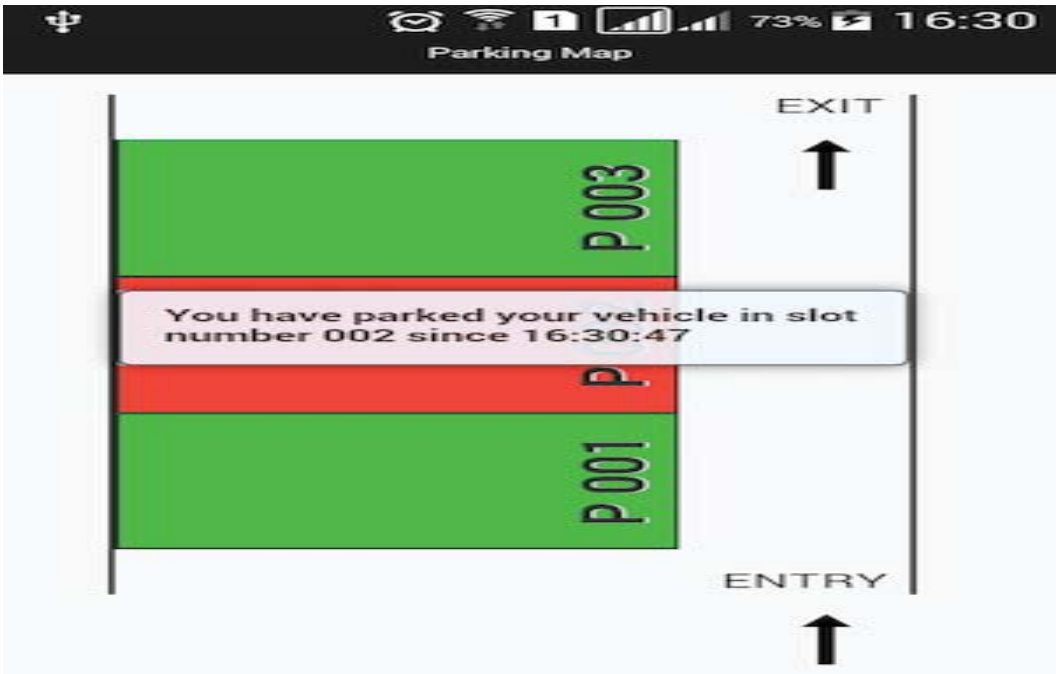
A user approaching a vacant parking space can reserve it in advance by tapping on a desired vacant parking slot on the app. This prompts the PMS to initiate a billing session for the user. PMS sends a message to the app on its private channel to initiate the start of billing. Along with this, PMS also starts a timer.



Here is a closer look at the format of JSON messages in this interaction

- The parameter RequestType with a value '2' indicates a request for reservation
- <reg_no> is the license registration number of the vehicle
- <private-channel> is identified by <reg_no> itself
- A sessionType with value '0' indicates the start of billing session for the user who has reserved the slot number identified by the value of deviceID.

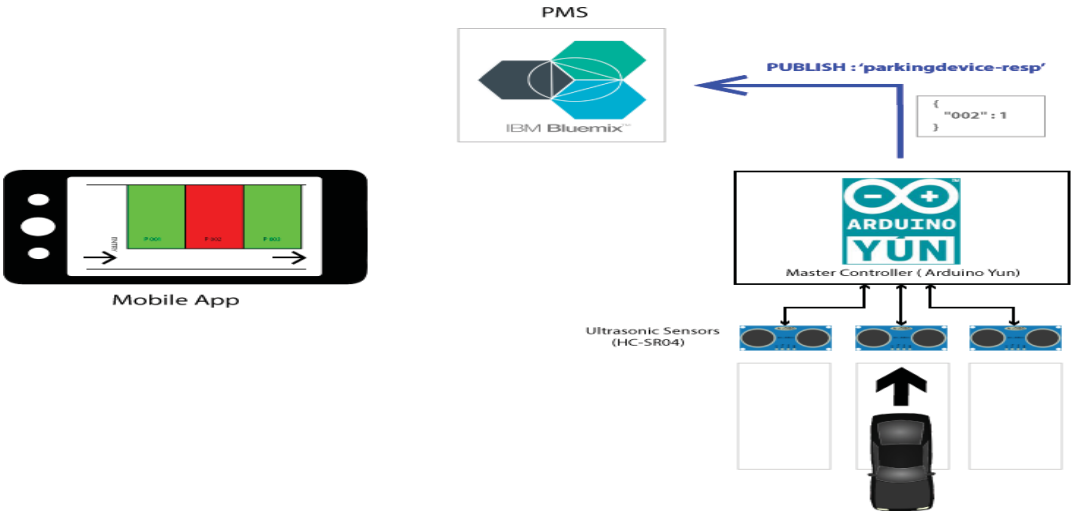
Upon receiving the message on its private channel, the mobile app displays a message to the user to confirm his parking reservation request:



Additionally, the PMS also sends an update on the channel parkingapp-resp to inform all mobile apps that the said parking space is now occupied.

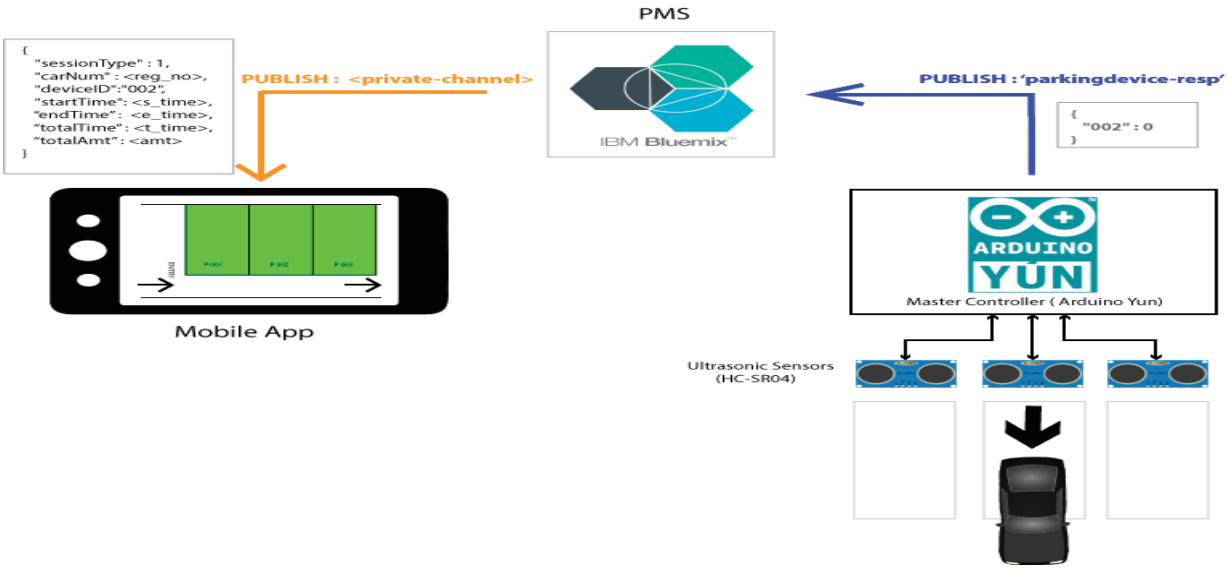
Reservation Confirmation

After reserving, when the user finally pulls in and parks his vehicle in the designated parking space, the hardware sends a status update to the PMS to indicate reservation confirmation. At this moment, the PMS stops the timer.



Billing Stop

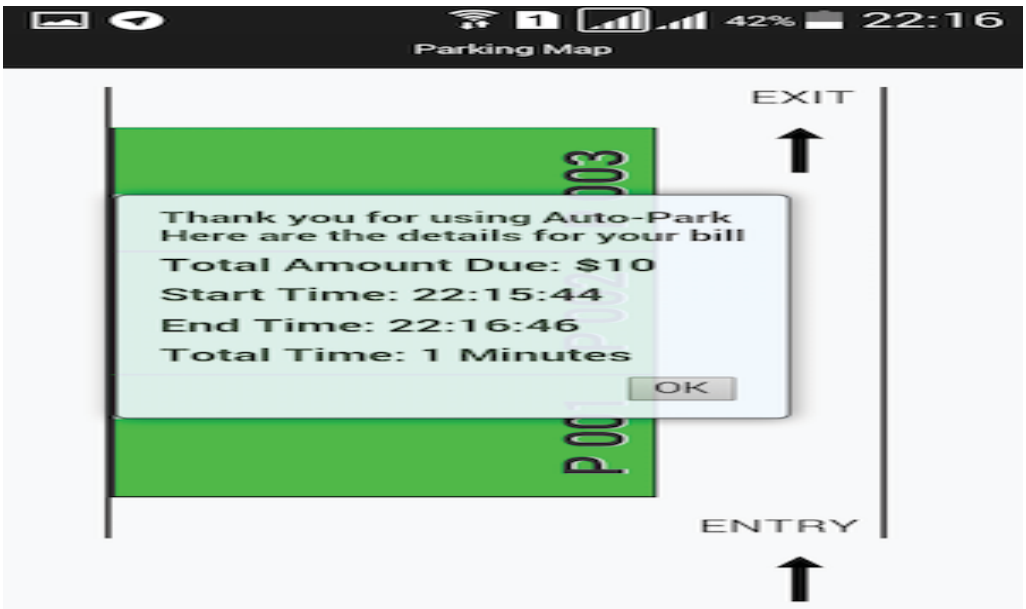
Later, when the user pulls out his vehicle from the parking space, the hardware device again senses this and sends a status update to PMS indicating that the space is now vacant. Upon receiving this update, PMS calculates the bill for the user and sends a billing session stop message along with bill details to the mobile app via its private channel.



The JSON message received by the mobile app contains the parameters for the bill details

- <s_time> is the start time of parking
- <e_time> is the end time of parking
- <t_time> is the total time of parking (in minutes)
- <amt> is the billed amount.

Finally, the mobile app displays the bill details on the screen as follows:



PMS is programmed to charge $10 for every sixty minutes of parking.

Alternate Scenarios

In case the user reserves the parking space but does not show up, the PMS timer, which is started in scenario 3, counts down from 60 seconds to 0 and then finally performs scenario 5 without the hardware trigger. In this case the the user is charged with a minimum bill $10.

Further, this application does not have any provision to authenticate the user during parking (as part of scenario 4), so as to ensure that the user who reserved the parking space is the one who actually parks his vehicle in that space. This is left as an exercise for the readers who would want to further enhance this application and make it feasible for a "closer to real life" deployment.

Wrapping Up

We had a great time building and playing with this application and testing the end-to-end scenario was quite interesting, considering that there are quite a few subsystems involved.

Using IBM Bluemix & PubNub services greatly simplifies the developmental challenges as the developer can focus on the application logic without worrying about how to make the communication work between different subsystems. Apart from this, both the services are capable of handling massive scale which can be leveraged to build similar real world applications which require 24×7 uptime and thousands of random users coming in.

IV. CONCLUSIONS

Our smart parking system project is planned to be integrated with another software application to help drivers to find the empty spot in parking lot more easily with less time. Also our project implement most of the functionalities needed in a parking lot. for example, implement an automatic way for payment.

V. ACKNOWLEDGMENT

I avail this golden opportunity to express my gratitude to all those people who have helped me to a great deal in preparing my project on ―A New Approach for To detect tracking system of Parking Vehickle by using Sensors and ArduinoYUN Device . First and foremost, I am indebted to the almighty God for providing me with ample strength and poise required to work persistently on this project.

My Proposed paper would not have been a real fulfilment without the backing and cooperation from various individuals through various means. First of all, I am highly grateful to Prof. Dr. N. SANDHYA from VNR Vignana Jyothi Institute of Engineering and Technology,CSE Department Hyderabad and Prof. Dr. A.Govardhan JNTU Campus Hyderabad.

I would like to extend my sincere thanks to Mr. A.Govardhan, my guide who gave me valuable advice and were good enough to find time for fruitful discussions.

REFERENCES

[1] Azeem Uddin, 2009. Traffic congestion in Indian cities: Challenges of a rising power.
[2] Jatuporn Chinrungrueng Udomporn Sunantachaikul Satien Triamlumlerd 2007. Smart Parking: an Application of optical Wireless Sensor Network. Proceedings of the International Symposium on Applications and the Internet Workshops
[3] Zeydin PALA* and Nihat INAN;**,April 2009.Smart Parking Applications Using RFID Technology. Jin Wen University of Science and Technology.
[4] M.Y.I. Idris, Y.Y. Leng, E.M. Tamil, N.M. Noor and Z Razak, 2009.Parking guidance system utilizing Wireless Sensor Network and ultrasonic sensor. Information Technology Journal.
[5] Hsin-Mu Tsai1, Cem Saraydar2, Timothy Talty2, Michael Ames2, Andrew Macdonald2, and Ozan K. Tonguz1,2007 .ZigBee-based Intra-car Wireless Sensor Network, ICC proceedings.