Ilkogretim Online - Elementary Education Online, 2020; Vol 19 (Issue 3): pp. 4559-4569
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2020.03.735608

Smart Parking System Using Internet Of Things

D. SATHISH KUMAR Assistant Professor Department of CSE Sri Sairam Engineering College sathish.cse@sairam.edu.in

N.L. POOJA UG Scholar Department of CSE Sri Sairam Engineering College e7cs142@sairamtap.edu.in

J. REESHITHA Software DeveloperAccenture, Chennai Reeshi1990@gmail.com

ABSTRACT- With the vast growing influx of population in the developed, industrially and technologically sound urban cities, an urgent need to make the cities are surmounted. The cities are made smart utilizing data sharing, artificial intelligence, machine learning, analytics and thousands of RFID tags and sensors. One of the significant concerns of today 's smart cities is the growing need to manage the vehicles on-road as well as to create sufficient and well - managed parking lots to prevent urban areas from traffic congestion. This leads to a call for a highly automated parking management system self-sufficient in guiding the driver to an available parking space in the nearby area. In this paper, a real - time prototype of the smart parking system (S.P system) based on Internet of Things (IoT) is discussed. The proposed smart parking system works on an electronic device that collects the parking availability status and assists drivers infinding and selecting the desired parking space among the available parking spaces that effectively reduces the traffic problems and mismanagement across the cities to a great extent.

Keywords- Ultrasonic sensor, Arduino, RFID Tag

1. INTRODUCTION

Parking of vehicles is quite possibly the trickiest territory of transport in metropolitan regions. It has been assessed that consistently, around 40% of gridlock in the urban areas around the planet is brought about by vehicles looking for the parking spot and it takes the driver a normal of 10-15 minutes to discover a parking spot. A shrewd leaving arrangement in isn't yet about vehicles that drive you home and afterward leave themselves. Notwithstanding, remote shrewd stopping arrangements that help to discover a parking space without cruising all over for 30 minutes, have effectively shown up. This is accomplished through extraordinary keen stopping sensors, in view of the Internet of Things (IoT) idea, that gather data and send it back to a focal platform. Thus, the framework gives constant stopping information permitting drivers to figure out which spots are involved and which are accessible for stopping. Such arrangements permit drivers simple admittance to stopping accessibility data directly in the application on their cell phones to discover open stopping spaces. In expansion to that, IoT assists with observing traffic and find stopping infringement all the more rapidly. At each and every parking slot, a sensor is put to distinguish the presence or nonattendance of vehicle which assembles the accessibility map for leaving direction and other services. Our exploration expects to build up a Smart stopping framework for ideal limitation of IoT-empowered remote sensor hubs to decide their situations in the stopping. The target capacities To propose an amazing failure cost sensor framework permitting ongoing stopping inhabitants checking without requiring any client/driver collaboration and furthermore its aides the client pre book a space the client can book a stopping opening of his decision preceding his appearance.

2. RELATED WORK

Length of the propagation path and signature of the channel impulse response (CIR) with respect to the vehicles, which can be obtained from UWB modules, are introduced to achieve vehicle detection. Moreover, considering that the energy consumption of UWB modules is much higher than that of magnetic sensors, the UWB-based vehicle detection algorithm is only activated when the signature of magnetic signals cannot enable MB-VDA to output an accurate decision. Finally, the proposed method was evaluated in a commercial parking lot under different conditions. The experiments show that it has a detection accuracy of 98.81% when the sampling rate of the magnetic sensor is 1 Hz.[1]

Improve the rotation of attractive spots (located nearby zones of interest) and set a usage-based parking assignment via appropriate incentives. Inherently, this is equivalent to use strategic pricing to strategically control the parking dwell time, which ensures a usage-based fair sharing of public space among users and improves the traffic conditions in the target area [2].

The quality of the sensors is as important as the fleet size (300 taxis with 10% probability of misreading provide availability information comparable to 486 taxis with 16% probability), while the use of Kalman filters did not lead to statistically significant improvements. In conclusion, the traffic management authorities should consider parking crowd-sensing via probe vehicles as a promising alternative

to the expensive deployment of the static parking sensors [3]

Architecture also makes use of Bluetooth Low Energy beacons, an Android app, a decentralized database and fog computing gateways, whose performance is evaluated in terms of response latency and processing rate. Results show that the proposed system is able to deliver information to the drivers fast, with no need for relying on remote servers. As a consequence, the presented development methodology and communications evaluation tool can be useful for future smart parking developers, which can determine the optimal locations of the wireless transceivers during the simulation stage and then deploy a system that can provide fast responses and decentralized services. [4]

Multi-objective grey wolf optimization technique for node localization withan objective to minimize a localization error. Two objective functions are considered for distance and geometric topology constraints. The proposed algorithm is compared with other node localization algorithms. Our algorithm outperforms the existing algorithms. The result shows that localization error is reduced up to 17% in comparison with the other algorithms[5].

A particle filter is implemented in order to increase the system performance and improve the credence of the results. Through extensive experimentation in both indoor and outdoor parking spaces, the system was able to correctly predict which spot the user has parked in, as well as estimate the distance of the user from the beacon [6].

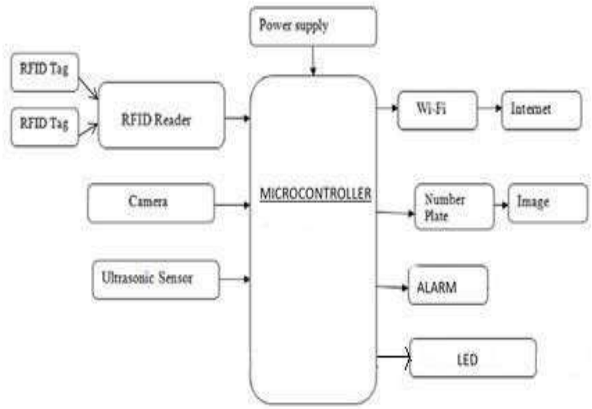
Integrate a block chain-based anonymous reputation management scheme into our system, where drivers can anonymously rate the parking service to ensure high qualityof service. [7].

3. PROPOSED SYSTEM:

The system normally consists of a microcontroller, vehicle counter, sensors, display board, RFID tags and RFID reader. The RFID reader which is installed near the entry gate of the parking lot. As soon as the RFID tag is read by the reader, the entry gate boomer opens to allow the car inside the parking area. At the same time the parking counter increments by one. Deduct the amount of parking, through the Internet we areable to find the parking zone whether theparking area is available or not. The idea behind the android application is to help the user analyze areas where parking is available and the number of slots free in that area. Additionally, prior to his expected arrival, the user can pre-book a slot in the area he desires if it is available, this will help the user to search the parking slot through application. To help avoid two users booking the same slot at the same time, the first come first serve is used and the user who fails to book will be guided to another vacant slot. After parking your vehicle in the slot, if the RFID is read again within 1 meter radius user will get a link in the application asking whether he/she is going to take the car and pay the bill, if they press yes then it will be directed to payment page else they can press no.

SYSTEM DESIGN

The block diagram representation of the proposed prototype of Parking lOT model.



According to this parking system, each parking space is installed with a parking lot model at the back end of the parking space.

Fig.1. Block diagram of system Design

The smart parking system proposed in this paper requiresthree major components:

1. Electronic meter

2. WLAN or Wi-Fi integrated local parking workstations at each parking facility with access points (APs)

3. Central parking server providing information about parking spaces availability throughout the city.

Electronic Meter (EM) is a device consisting of various hardware components for achieving different results. EM has an RFID vehicle tag sensor for detecting the presence ofa vehicle in parking space along with ultrasonic sensors. RFID tags are also used in check-ins and check-outs and for the generation of parking bills, in a microcontroller Arduino for the backend codes, in an alarm IC for buzzing in case of improper parking and mismanagement of parkingspace LED for showing the status of the parking lot, whether parking is available or not.

RFID READER

An RFID reader is the brain of the RFID system and is necessary for any system to function. Readers, also called interrogators, are devices that transmit and receive radio waves in order to communicate with RFID tags. RFID readers are typically divided into two distinct types– Fixed RFID Readers and Mobile RFID Readers. Fixed readers stay in one location and are typically mounted on walls, on desks, into portals, or other stationary locations. A common subset of fixed readers is integrated readers. An integrated RFID reader is a reader with a built-in antenna that typically includes one additional antenna port for the connection of an optional external antenna as well. Integrated readers are usually aesthetically pleasing and designed to be used for indoor applications without a high traffic of tagged items. Mobile readers are handheld devices that allow for flexibility when reading RFID tags while still being able to communicate with a host computer or smart device. There are two primary categories of Mobile RFID readers – readers with an onboard computer, called Mobile Computing Devices, and readers that use a Bluetooth or Auxiliary connection to a smart device or tablet, called Sleds. Fixed RFID Readers typically have external antenna ports that can connect anywhere from one additional antenna to up to eight different antennas. With the addition of a multiplexer, some readers can connect to up to 32 RFID antennas. The number of antennas connected to one reader depends on the area of coverage required for the RFID application. Some desktop applications, like checking files in and out, only need a small area of coverage, so one antenna works well. Other applications with a larger area of coverage, such as a finish line in a race timing application typically require multiple antennas to create the necessary coverage zone.

Fig.2 RF ID Reader

RFID TAG

An RFID tag in its most simplistic form, is comprised of two parts – an antenna for transmitting and receiving signals, and an RFID chip (or integrated circuit, IC) which stores the tag's ID and other information. RFID tags are affixed to items in order to track them using an RFID reader and antenna. RFID tags transmit data about an item through radio waves to the antenna/reader combination. RFID tags typically do not have a battery (unless specified as Active or BAP tags); instead, they receive energy from the radio waves generated by the reader. When the tag receives the transmission from the reader/antenna, the energy runs through the internal antenna to the tag's chip.

Fig.3 RF ID Tag

The energy activates the chip, which modulates the energy with the desired information, and then transmits a signal back toward the antenna/reader.

ULTRA SONIC SENSOR

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e., the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target). In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. Theformula for this calculation is D = ½ T x C (where D is the distance is the time, and C is the speed of sound ~ 343 meters/second). For example, if a scientist set up an ultrasonic sensor aimed at a box and it took 0.025 seconds for the sound to bounce back, the distance between the ultrasonic sensor. Ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self- parking technology and anti-collision safety systems. Ultrasonic sensors are also used in robotic obstacle detection systems, as well as manufacturing technology. In comparison to infrared (IR) sensors in proximity sensing applications, ultrasonic sensors are not as susceptible to interference of smoke, gas, and other airborne particles (though the physical components are still affected by variables such as heat). Ultrasonic sensors are also used as level sensors to detect, monitor, and regulate liquid levels in closed containers (such as vats inchemical factories). Most notably, ultrasonic technology has enabled the medical industry to produce images of internal organs, identify tumors, and ensure the health of babies in the womb.

Fig. 4 Ultrasonic sensor

NODE MCU (WIFI)

Node MCU is a low-cost open-source firmware for which open-source prototyping board designs are available. The name “NODEMCU” combines “node” and “MCU” (microcontroller). It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Expressive Systems, and hardware which was based on the ESP-12 module. Later, support for the ESP32 32-bit MCU was added. The term & quot; Node MCU & quot; strictly speaking refers to the firmware rather than the associated development kits. Both the firmware and prototyping board designs are open source. The firmware uses the Lua scripting language. The firmware is based on the e Lua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open-source projects, such as lua-cjson
[9] and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented. The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface- mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications.

Fig.5 Node MCU ESP 8266

3.5 ARDUINO

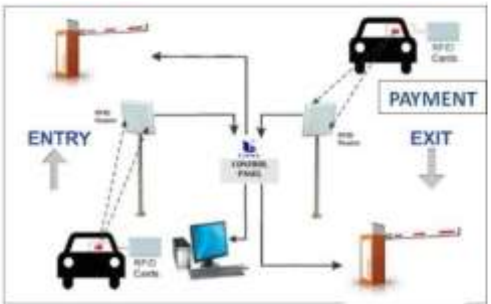
Arduino microcontrollers ae pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default boot loader of the Arduino Uno is the Opti boot bootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB- to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB- to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the ArduinoMini and the unofficial Boarduino, use a detachable USB-to- serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in- system programming (ISP) programming is used The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, sixof which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards. Many Arduino- compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education,[34] to simplify making buggies and small robots. Others are electrically equivalent, but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use differentprocessors, of varying compatibility.

Fig.6 Arduino UNO

4. IMPLEMENTATION

Parking Entry System

This module contains parking gates embedded with RFID sensors. The moment a vehicle comes near this gate, the RFID sensor senses the RFID tag on the vehicle and records all the information stored on the RFID tag installed on the vehicle. At exit time, the parking gates again read this RFID tag, update the system for billing, and generate the parking bill or charges to be paid according to the time. This charge is deducted automatically from



the linked account .

Fig.7 Parking Entry system

Parking Lot Monitoring System (PLMS)

This software module is set up on the microcontroller Arduino. Its primary functioning is to detect the availability of parking space within any parking area based on the time difference between transmitted and received signal. Here, the ultrasonic sensor sends the detecting signals for the proper confirmation of vehicle presence in that particular space. Upon successful detection of a vehicle, the occupancy of the parking space is confirmed. Then, RFID sensors come to play and reads the RFID tag already installed in the vehicle and the entry time. After all this process, this parking space status shows as being occupied.

Fig.8 Parking Lot Monitoring system

When the vehicle exits i.e., when the vehicle reaches the exit gate RFID sensor, the exit time is collected, and a billis generated.

Local Parking Management System (LPMS)

This module is set up on the server of the local parking management system with each parking lot. It monitors the overall statuses of the parking spaces, which can be free, occupied, or reserved.

Central Parking Management System (CMPS)

It is set up as the primary parking system with a global IP for communication over the internet. It maintains all the records of parking facilities and parking area statuses

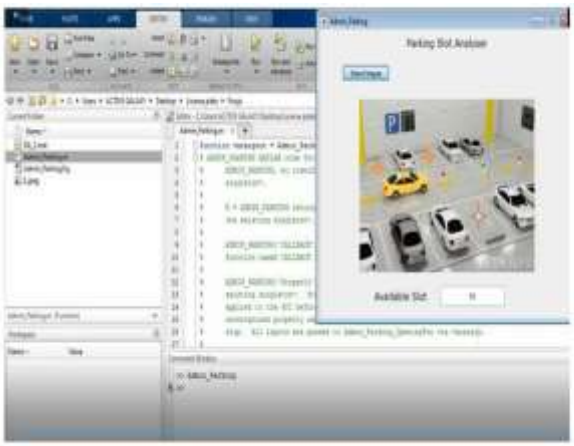


Fig. 9 Parking Slot Analyser.

5. CONCLUSION

As a result of executing this project, the goal was to develop a smart phone application that people can utilize to securely and easily find and pay for parking, while also providing management capabilities for the parking facility owners. These devices are usually embedded into parking spots or positioned next to them to detect whether parking bays are free or occupied. This happens through real-time data collection. Smart parking can reduce traffic by making it easier to locate empty parking spaces, thus lowering the risk of distracted driving. Through the use of sensors that detect whether space **is** empty or occupied and lights that indicate this, smart parking technology can help drivers locate spots.

6. FUTURE WORKS:

Efficiency of the space is achieved by storing the data in the json format in the server, which helps us to maintain server with less load and space can be unlimited. With the help of Wide Dynamic Range (WDR) algorithm incorporated the shade, glare and white balance of light canbe minimised and sharp images can be saved for the furtheranalysis

REFERENCES

[1] L. Lou, Q. Li, Z. Zhang, R. Yang and W. He, "An IoT-Driven Vehicle Detection Method Based on Multisource Data Fusion Technology for Smart Parking Management System," in IEEE Internet of Things Journal, vol. 7, no. 11, pp. 11020-11029, Nov. 2020, doi: 10.1109/JIOT.2020.2992431.

[2] Jioudi, B., Sabir, E., Moutaouakkil, F., & Medromi, H. (2019). Congestion Awareness Meets Zone-Based Pricing Policies for Efficient Urban Parking. IEEE Access, 7, 161510-161523.

[3] Bock, Fabian, Sergio Di Martino and Antonio Origlia. "Smart Parking: Using a Crowd of Taxis to Sense On- Street Parking Space Availability." IEEE Transactions on Intelligent Transportation Systems 21 (2020): 496-508.

[4] M. Celaya-Echarri et al., "Building Decentralized Fog Computing-Based Smart Parking Systems: From Deterministic Propagation Modeling to Practical Deployment," in IEEE Access, vol. 8, pp. 117666-117688, 2020, doi: 10.1109/ACCESS.2020.3004745.

[5] Ghorpade, Sheetal & Zennaro, Marco & Chaudhari, Bharat. (2020). GWO Model for Optimal Localization of IoT-Enabled Sensor Nodes in Smart Parking Systems. IEEE Transactions on Intelligent Transportation Systems. PP. 1-8. 10.1109/TITS.2020.2964604.

[6] Zhu, Liehuang & Li, Meng & Zhang, Zijian & Qin, Zhan. (2018). ASAP: An Anonymous Smart-parking and Payment Scheme in Vehicular Networks. IEEE Transactions on Dependable and Secure Computing. PP. 1- 1. 10.1109/TDSC.2018.2850780.

[7] F. H. Hung, K. Tsang, C. K. Wu, Y. Wei, Y. Liu and
W. Hao, "Cost and Time-Integrated Road-to-Park Cruising Prevention Scheme in Smart Transportation," in IEEE Access, vol. 7, pp. 54497-54507, 2019, doi: 10.1109/ACCESS.2019.2910309.