Development of An Integrated Web-based Indoor Parking System with Sensors

Mosaab M. Hamed[1], L. A. Latiff[2*], Izzaty A. Kamil[3], Rudzidatul Akmam Dziyauddin[4], Hazilah Mad Kaidi[5]

[2,4,5] Ubiquitous Broadband Access Network (U-BAN) Research Group,
Universiti Teknologi Malaysia, Kuala Lumpur, 54100, MALAYSIA

[1,3] Razak Faculty of Technology and Informatics,
 Universiti Teknologi Malaysia, Kuala Lumpur, 54100, MALAYSIA

*Corresponding Author

Abstract: Growing number of vehicles in recent years has an impact to road capacity and parking spaces. Even with support of an efficient public transport, some users prefer to use their personal vehicles. Hence, it resulted in lack of parking space in public areas. Looking for parking spaces is time, money and fuel consuming and causes unnecessary stress to drivers. This work will design and develop the sensing infrastructure which will be used as a part of an integrated smart parking system that integrates the sensing capability with wireless transmission to the central web portal. Then, the central web portal will provide a platform for vehicle parking reservation system made online through a computer. This system also permits drivers to make a reservation on available parking slots utilizing ID code keyed in for authentication and several indicators to denote availability, non-availability and reservation status at the parking slot. The communication technologies used has to upload and download information to and from the web. This work has shown that the development was successful for a single floor parking system and can be upgraded to multi-storey parking complex.

Keywords: smart parking, Arduino, Internet of Things, wired sensors, ESP32.

1. Introduction

The Internet of things (IoT) is defined as a network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to connect and exchange data. (Brown, Eric et al., 2016).

This huge progress in technology affected every aspect of peoples' life where their lifestyle has to adapt to the advancement and the trend is towards a fast, faster and extreme service. This will socially impact societies to require less time to do a certain task and the fast routine forces society to look for the opportunities and alternatives which will save their time, money, fuel, etc.

From other prospective, societies also do not like restrictions. Even with technology, though it provides them with fast services, it must have certain requirements and specifications to fulfill them. Portability is the main criteria in choice, where the technology travels together without restrictions to a certain location, which leads to the term mobility and portable small size device (the smaller the better) or even hidden (Uckelmann et al., 2011). All of those requirements by human nature have guided the technological advancement to the level of ubiquitous computing.

Exceptional growth in the number of cars, buses, trucks, and other modes of transportation over the past few decades give arise to new problems (Alkheder et al., 2016) and a challenge for road capacities and health measurements. It is necessary to assure smoother traffic movement and to save driver's time in looking for available parking slots.

Concerns for parking are becoming important to best support the urban core (Atif et al., 2016) which is where the flow process of this project started from a research down to implementation. After selecting the required electronics component and ensuring the feasibility, suitability and compatibility of the selected, the work will be simulated to verify the safety of the components, then the wiring and installation for the system in physical field (Bachani et al., 2016).

With the above scenario, that is to provide a fast and non-time-consuming search for a parking slot, a smart parking system is proposed. The project is divided into three stages: detection, indication and reservation. The first two stages are included in the designing and implementing of hardware infrastructure, which can be used for indoor and outdoor sensing with minor changes at the sensing capability. Reservation is two-way communication since the user can place reservation order through the website and then the indicators would change as a response.

In general, it could be considered as a system where every component will be integrated to acquire certain specifications such as suitability, and compatibility and environmental friendly. The technical details of power systems and electronic devices with programming skills are all considered also. Hence, the system will provide a service that is portable, inconspicuous with minimal level of required interaction.

The proposed system when implemented is expected to reduce roaming time by easily indicating the status of each parking slot. GREEN light means availability, RED indicator means this slot is occupied by another car, YELLOW indicator means the slot had been reserved by using mobile application (Yadavalli, 2016) and/or website gateway (Zhou, Honbo et al., 2013). A website shows the parking bays status in real time had been developed which is also gives users ability to place reservation orders for the available spots.

This paper is organized as follows where the related work will be discussed in Section 2, followed by methodology, which consists of simulation and hardware design and development in Section 3. Section 4 will present the initial results followed by Section 5 which will discuss the recommendations and future work to be carried to implement an integrated smart parking system. Finally, Section 6 will conclude the paper.

2. Related work

In (Geng and Cassandras, 2012), a new "smart parking" system has been proposed for an urban environment where the system assigns and reserves an optimal parking space for a driver based on user's requirement and proximity destination and parking cost. It ensures the overall parking capacity will be efficiently utilized. This system uses the Mixed Integer Linear Program (MILP) approach to obtain optimal allocation based on current state information and subject to random events such as new user requests or parking spaces becoming available. However, their proposed system used complex system processes for detection, which requires very expensive devices.

(Vera-Gómez et al., 2016) described a low-cost, minimally intrusive system for the efficient management of parking spaces on both public roads and controlled zones. This system is based on wireless networks of photoelectric sensors that are deployed on the access roads into and out of these areas. These sensors will assist drivers and facilitate their search for parking space. However, this method incorporates additional equipment to be installed in public area such as solar power system, the processing and communication node together with the sensors in the speed bump, so as to assist and further facilitate adoption of this monitoring technology.

To mitigate problems such as illegal parking (Yang et al., 2017) proposed iParking, a real-time parking space monitoring and guiding system. Image analysis from event recorders embedded in cars on the road is used to recognize the availability of parking spaces. Drivers are navigated to the nearest available parking spaces upon request. Problems arise when background drivers (i.e. drivers who do not use the system) might accidently occupy a parking spot, therefore information on the availability of more than one parking space should be considered.

3. Methodology

Since the objective of the system is to detect, indicate and reserve parking slots, therefore utilizing several indicators at the parking space is considered, Then, with these indicators uploaded to the central web portal that will update the available parking space, users will be able to make reservations through the web using mobile apps or computer. This can be done even before the user starts their journey. Hence, this project requires electronic devices and algorithm that are simple, interoperable and fast.

Figure 1 shows the layout of the sensing node, which is used throughout the project, a battery that supplies the power to ESP32 board via Vin and ground pins. Four pins on the ultrasonic sensor board will be connected to ESP32 with two pins to supply power, one for ping and one for echo (to receive the pong). The microcontroller sends the signal and waits in the listening process. The time difference between the two signals will be extracted and used to calculate the distance.
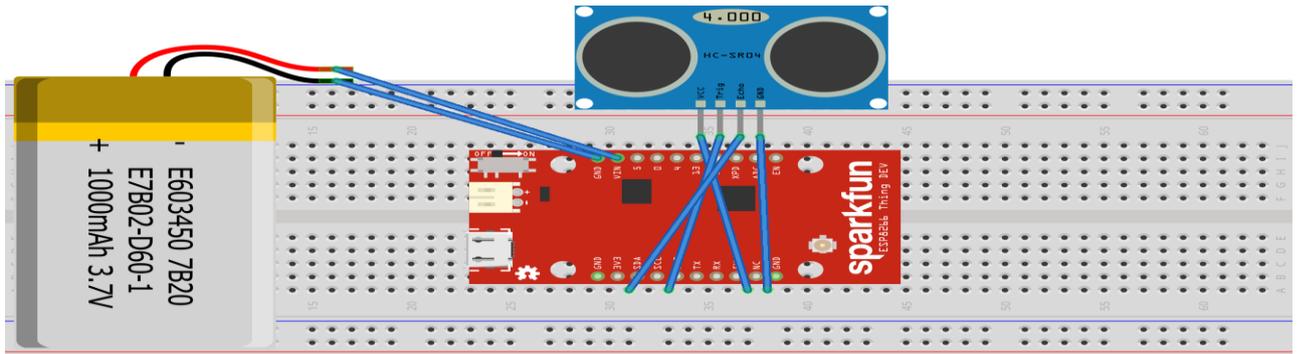
Fig. 1 - Schematic diagram of sensor node

Figure 2 shows the algorithm in ESP32 controller to detect and indicate the presence of the car. The distance determined must be validated that the car is in the designated parking lot. Hence, the sensor will be placed at the back of the parking lot and the maximum distance is 2 meters. A Red light indicator will be activated if the system correctly detects the presence of the car. Otherwise, the green light is maintained.

Updates on the availability of each parking space will be updated in the website to enable the reservation of the web-based indoor parking system.
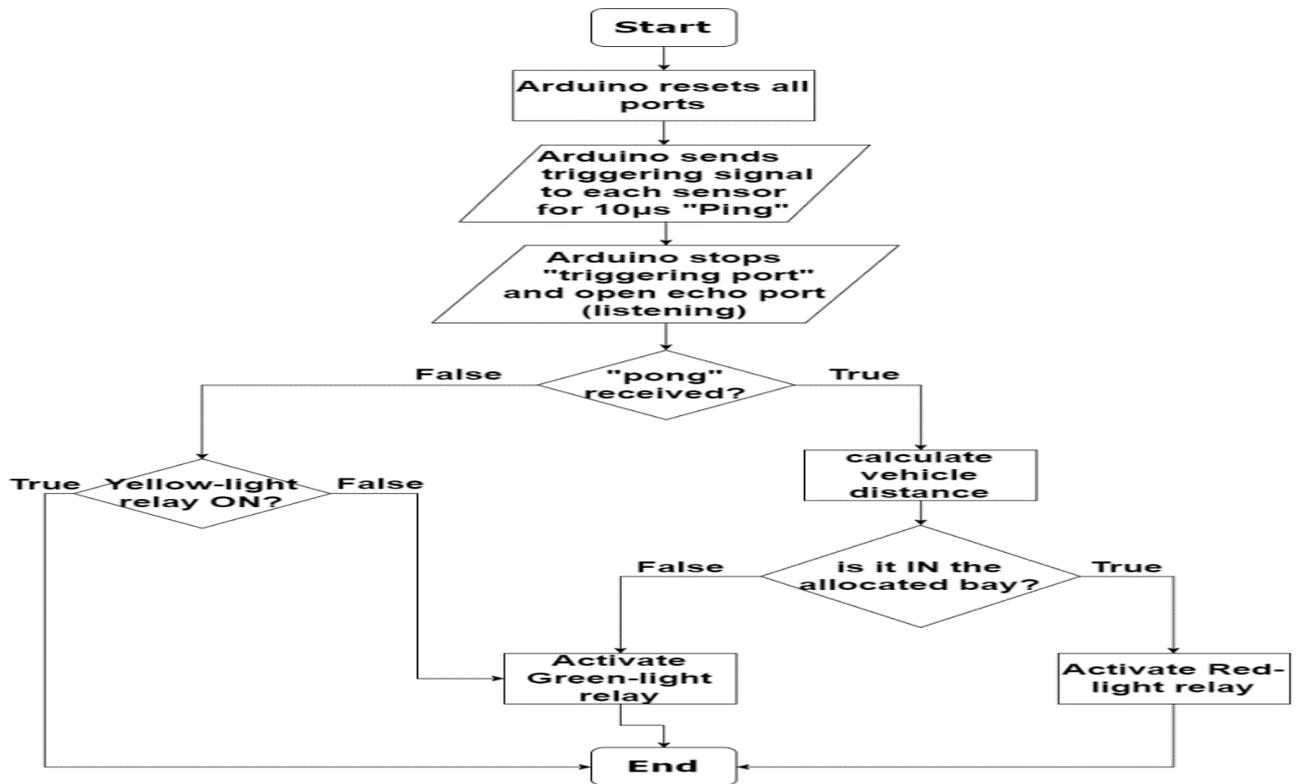

Fig. 2 - Sensing and indication mechanism

A. Selection of Appropriate Components

Every component comes with a specifications data sheet, sensors for instance has many manufacturing brands and specifications. The hardware consists of sensors, microcontroller and relays, and lastly indicators and its appropriate power supply. Since this is an indoor environment project, ultrasonic sensors and using proximity as the sensing criteria is selected. Ultrasonic sensors use 5V supply (Allan, Alasdair et al., 2011) and is able to work in 500mA current which is appropriate for indoor environment.

Besides this, the waterproof and humidity resistance characteristics must also be considered as a determining factor in the selection. Error tolerances in resistors vary from manufacturer to another and some resistors have 5% tolerance while others have 1% admittance. This must be taken into account to avoid mismatches when integrating all the sensing hardware later.

For the indicator lights stage, there are two types of voltages required: 220V AC for the LEDs and 5V for the control lights from the microcontroller. So, the focus here is how to control a device, which works with medium AC voltage and a very low DC voltage (S. Ananthi et al., 2006), (Avago Technologies et al., 2010). For this purpose, a special relay module called Arduino relay module had been used because it has a component that transfers electrical signals between two isolated circuits by using light called Opto-isolators. It prevents high voltages from affecting the receiving system. The light lamps chosen should fulfill the indoor ambiance and consists of an array of LEDs. For outdoor indicator lights, it could be a surface-mount device (SMD) LED module in order to have a powerful indicator under strong luminance.

The brain of the system is the Arduino which is an open source computer hardware and software developer, supporting projects and user community and designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world (David Kushner et al., 2011). There are many Arduino boards on the market but for this project ESP32 has been selected. The Arduino Mega has 16 analogue input pins and is a suitable choice for small to medium sized parking. It has ATmega1280 chip on board with 16MHZ oscillator, which means enough processing operations to handle (Jonathan Oxer et al., 2013). To control the reservation lights from central online service, the Arduino must be able to access the internet which requires a mini computer such as RaspberryPi. The use of both Arduino Mega and RaspberryPi have escalated the total price of the hardware. To reduce cost by 50%, the ESP32 is used which is equipped with a built-in Wi-Fi chip (Karvinen, Tero; Karvinen, Kimmo et al., 2011) to access the internet and thus eliminating the use of RaspberryPi.

Once a customer place a reservation through the website or mobile apps, the chosen parking slot will indicate the Yellow light, to show that the parking space has been reserved. To avoid manipulation or unserious reservations, ESP32 will activate an internal timer for 15 minutes (this can be changed with thorough process evaluation), which must be ample for the car to arrive at the allocated slot. Another level of authentication will be provided to deter unwanted or walk-in users. For this, a user has to key-in an authentication code so that the lights status will be changed to Red. Otherwise, the sensors will detect unauthorized parking, as the systems will counter check and detect a no authentication approval on the allocated slot. Failing to authenticate within the allocated time will force the reservation to be cancelled and a timeout error will be sent to the customer (Monk, Simon et al., 2011).

B. Programming

The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing) (Timmis, Harold et al., 2011) and the process flow is as shown in Figure 2.

Arduino ESP32 uses the Arduino C language and the platform utilized a software called Arduino Integrated Development Environment (IDE) which is a programming environment with debugging ability to debug programming errors (Noble, Joshua et al., 2012).

3.1 Circuit Design

To proceed with the project, and to design the circuitry based on the methodology described, the design will undergo simulation and implementation stage. Figure 1 illustrates the sensing node for an integrated smart parking system. A microcontroller inside the ESP32 handles the signals to/from the ultrasonic sensors and controls the relays board based on the occupation of each parking bay.

A. Online Circuit Simulation

The circuitry was designed using an online simulation platform in www.tinkercad.com to build-up a virtual circuitry but real implementation with wiring between sensors and breadboard with wide range of hardware libraries. Then the circuit was simulated to using the Arduino code written on the same platform too. The debugging feature was also done online.

Detection and indication stages can be easily seen in the simulation since they rely on hardware infrastructure while reservation and online platforms can be tested on a different way from the cloud-computing service provider. The concept overview will be presented in Figure 3.

B. Circuit Development

Next stage is hardware development, which consists of physical wiring and sequential testing. This stage is divided into several phases to ensure correct implementation and to avoid errors. The first phase is installing the Ultrasonic sensors in the allocated parking slots, followed by installing indicators lights in each slot. This is followed with the wiring process, which starts, from the indicators up to the relay module (16 channels). Each parking slot would require three relays to control the red, green and yellow lights and two pins from the microcontroller board.

Direct current power supply is used to provide the system (Bhargava, Kulshrishtha et al., 1984) with two voltages; 5V to power the Arduino and the small electronic components attached to the breadboard; and 12V to supply the 16-channel relay module.

An online reservation through webpage, send signals to ESP32. ESP32 is able to collect data and will change the light indicator from Green (available) to Yellow (reserved).

To guarantee a safe and efficient working environment for the controller and relay module, it is highly recommended to put the Arduino ESP32 board, relay module, dc power supply and all sensitive electronic element in an air-conditioned room or a good ventilated environment. Wires are selected to consider the distance from power supply to the indication lamps, where voltage drop and wire medium resistance are factors to be considered.

Sensing infrastructure is part of the complete system of an indoor smart parking system. This system will be an advance and efficient system compared to the existing smart parking system. Most existing smart parking system used two lights indicator, Green for the availability of parking slots, and Red light indicate the parking slot is occupied. This project introduces yellow light indicator for reservation. Users are able to make a reservation using a mobile application anywhere and anytime. It will give user some time to arrive at the destination and park their car without hassle. This method is able to reduce roaming time looking for available parking slot. Once the reservation is being made through the webpage or mobile application, a QR code is generated. The QR code is being scanned at the boom gate for authentication and substitute for parking tickets, which will minimize the usage of paper. This system allows users to pay parking fees online using the payment gateway integrated to the system; without using cash and eliminate the problem of queuing and finding a machine to pay for parking fees. In addition, this system will also accommodate for walk-in users, users who did not make any reservation for parking slots. Walk-in users will need to take a parking ticket at boom gate and park at available parking slot where the green light indicator is on. This system optimizes the usage of parking space and also improves on comfort and peace of mind.

Parking   Ultrasonic Sensor - HC-SR04   ESP32 Development Board

Parking web-page

parking mobile app.

Cloud server (hosting website& SQL database)

Fig. 3 - Concept Overview

3.2 Website Development

The website must be able to extract the information from the sensors and display it for users. The status of the parking slots should indicate 'available', 'reserved' and 'occupied' in the webpage which will be auto refresh every 3 seconds. The light will be GREEN, YELLOW and RED lights respectively at the parking bay. The process at he webpage is indicated in Figure 4.

Application programming interface (API) has been used in this project for the purpose of read/write data to and from the database to access parking status or retrieve the reservation records. All API were written in Hypertext Preprocessor PHP programming language with seven API's included in each parking bay folder in the web-hosting page and given as follows:

- db_connect.php is responsible of establishing the connection to the database.
- dbconfig.php contains the credentials to access the database like database user name, database password, database name and database server which is the local host in many cases. If the credentials are correct then the db_connect.php will return with access key and the connection is considered established. Otherwise the connection to the database closes and mysql_error() function will be called as a connection failure.
- insert.php is used to insert values to the tables in the SQL database.
- update.php will modify pre-existing value in the table.
- delete.php will delete a certain value from the table or even certain row or column.

- read_all.php consists of 52 coding lines with the function to read all the values of a specific table, show those values on the project webpage so that the end user can easily understand.
- read_specific.php contains 72 php coding lines to retrieve and get a specific value stored in a specific field of the table, most of the HTML code of this project mainly depended on this API to retrieve the "Reservation" table of each parking bay to get the status of it.
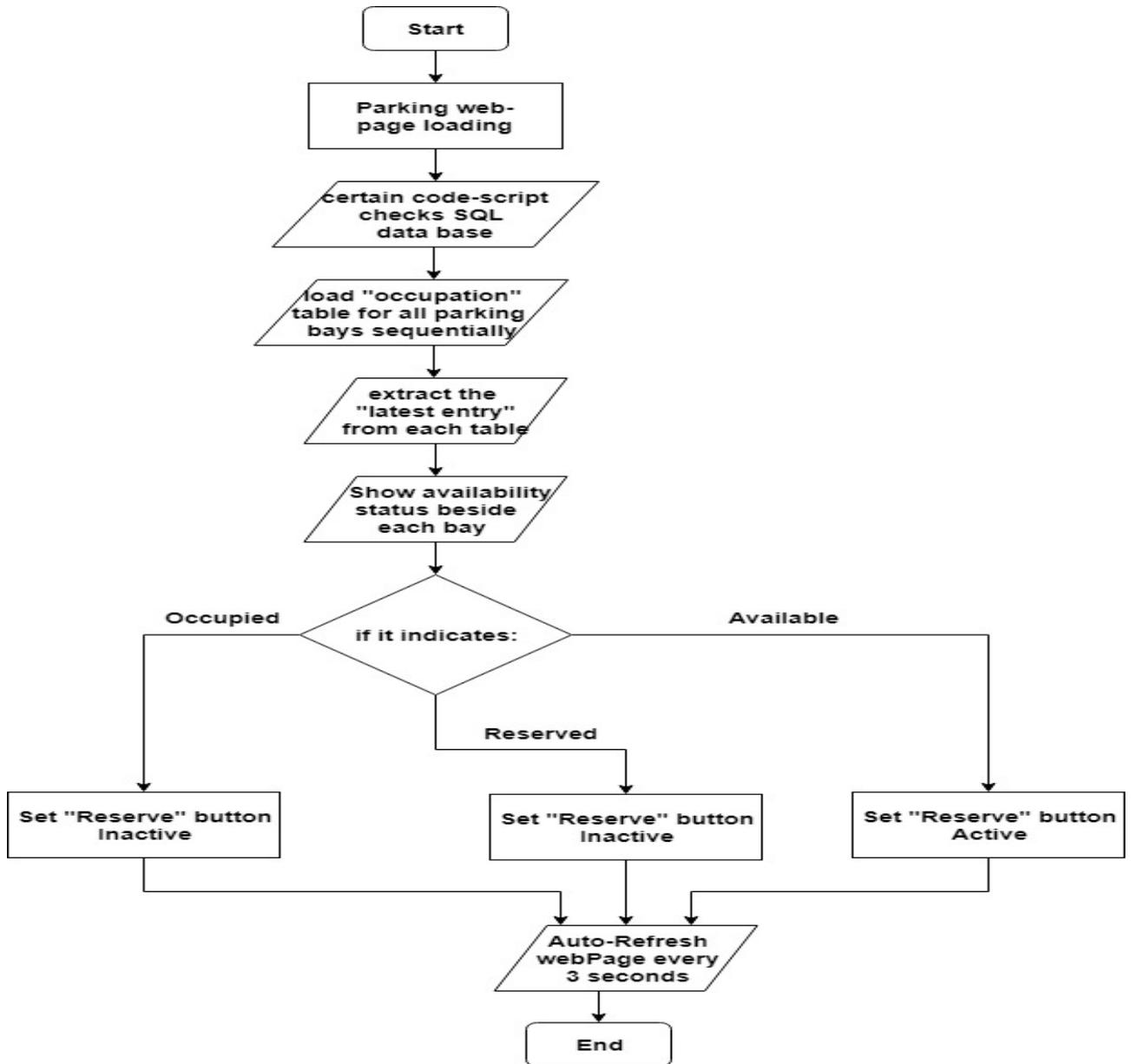
Fig. 4 - Webpage process flow

All seven API will be called by the controllers to send sensors data and record it to the corresponding table fields, also to retrieve the reservation status from the tables to check whether the parking bay has been reserved or not, in case of positive acknowledgment (reservation had been requested) the controller processor will jump to the coding part which will set the YELLOW light on. Figure 5 shows the process flow for the YELLOW light.
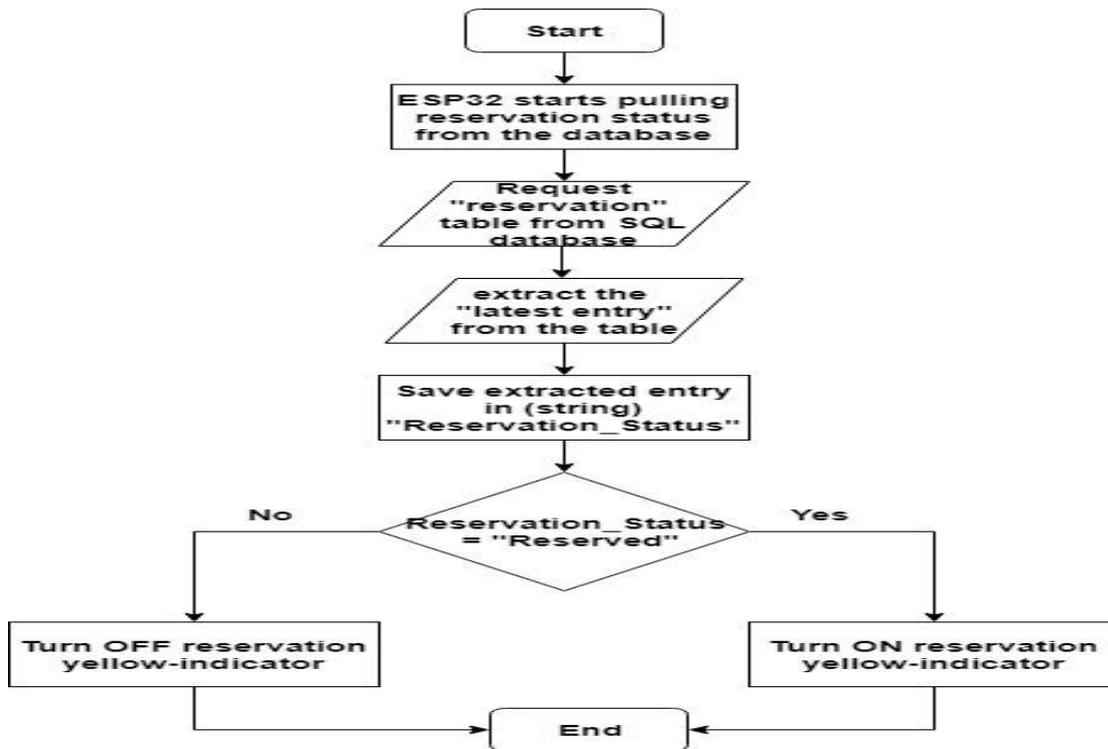
Fig. 5 - Process flow in reservation phase

The website also has a graphical user interface (GUI) for end user to quickly view the status of the parking slots. If the bay shows "Occupied" status, the reservation option for it is inactive and at the same time the Red light at the parking bay field is turned ON. If the parking bay is available, the "Reserve" button should be active so that the user can make reservation and at the same time the Green light is ON to indicate bay availability. Once Reserve button is clicked, the Yellow light is turned ON to indicate the reservation status and Reserve button become inactive for this particular field.

Figure 6 shows the simple GUI developed displayed at the webpage to show the status of the parking bays (bay1, bay2 and bay3) which are available and there is no reservation made via the online parking website.
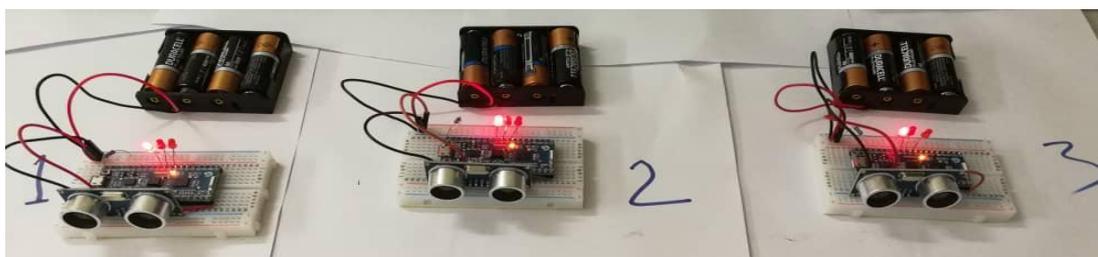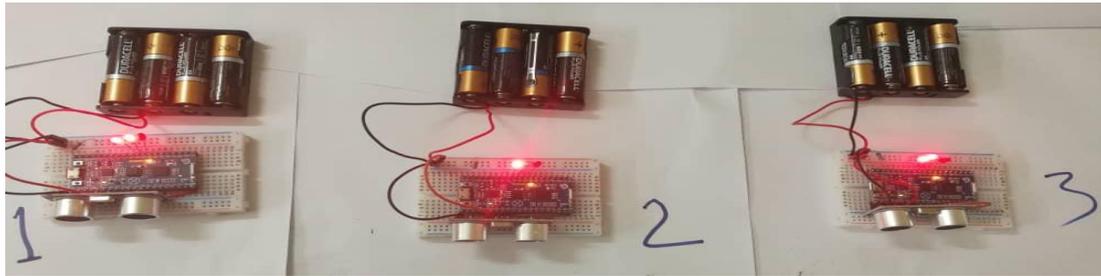

(a)

(b)

Fig. 6 - (a) All bays are available; (b) GUI at website shows availability and reservation options.

Figure 7 presents a scenario where parking bay1 and parking bay3 are reserved indicated by Yellow while parking bay2 is available.

(a)

(b)

Fig. 7 - (a) two bays reserved; (b) webpage shows parking status in real time

Each parking bay should have two tables in the SQL database, which also requires the developer to wisely choose a web server that could provide sufficient number of databases to accommodate tables required for the whole parking facility. The current project can handle a single floor parking system.

4. Recommendation and future work

The project can be further improved to accommodate multiple floors of parking space and users are aided once they are in the parking facility. Users will have information on the parking status before they even arrive, easily browse all the bays arranged by floor numbers and bay numbers. It must be designed to show users and are indicated a red car under the bay number to indicate occupied, green car to indicate the availability of the bay while the yellow car indicates the reserved status. It is up to the user if he/she wants to proceed with the reservation or simply do a (drive-in), either cases it drastically reduces the time wasted by the drivers roaming to find a place to park. Once the driver is about to reach the parking he would easily check the webpage to find the nearest green-indicated parking bay and then go straight to it. However, drive-in users cannot guarantee the parking is still available until the driver arrives at the parking bay, since it is on first in first served bases. Reservation on the other hand could guarantee the booking of the allocated bay for the preserver.

Further on, the reservation process may include online payment gateway where the user would get a randomly generated password after the payment has been accepted. This password will be used to identify the ownership of the reserved position.

The webpage is coded using Hypertext Markup Language HTML and JavaScript. The general page look and design are shaped by HTML codes while the interactive button functions, the dynamic parking bays status and reservation requests and status are all made by JavaScript codes involved inside the main code of the page itself. Each parking bay has its own JavaScript code which recalls the read_specific.php API to retrieve the parking status from the table, a process to extract the latest value recorded in the table (because the recording process might be using insert or update API's as mentioned previously) the latest value from the table is always what needed no matter what recording procedure was used. Depends on the latest value extracted, the Java code will pass the value to the HTML code which is responsible of the overall appearance to show the appropriate parking bay status on the screen to the end users.

The same process is used in reservation where the JavaScript code send retrieve command to the SQL database to access the reservation table. Once the data received, the latest value stored in the reservation-status field is extracted and then passed to the HTML code to show the status. If the bay is reserved, the yellow car appears and the reservation button became inactive while if the bay is occupied the reservation button would also be inactive.

From the in-the-field controller perspective as ESP32 is coded using ArduinoC programming language, it also include API publish and subscribe concept, the same concept as in the webpage. The controller uses the API commands to record sensors data into the SQL database, and also retrieve the reservation commands from the cloud and acts accordingly. If a reserve command received by the processor, it instantly indicates reserved status. Otherwise, it will keep tracking the data being received from the sensors and update the LED indicators as it should be. The processor goes line by line

through the code and repeat this process periodically, each processing cycle includes sensors Ping and Pong, indicators update, reservation database check and parking bays status database update and website data refresh.

Arduino boards are being developed continuously, with a new board introduced almost every year to the market with better processing capabilities, faster operations and bigger internal memory size. Hence, it is extremely important to obtain the latest product available to ensure that the system is modern, fast and secure enough. This can be done easily as the code can be transferred from the old board to a newer one without any need for modification or editing.

Addition of new parking slots will only require adding some lines into the code thus the labeling of the older hardware structure may confuse the programmer. Therefore, it is recommended to choose variables and names, which will not confuse the processor. The above check list test should be done after each code modification or extension of the system.

The system should be equipped with security system to prevent an unauthorized access to reserved parking spaces. Once the reservation is being made, the user application sends a random pin code to the user for authentication. Matched pin code is being keyed-in allow the yellow light indicator to change to red light indicator, to show the parking space is occupied. Unmatched pin code will trigger the alarm and send a signal to the nearest security guard to take action.

Another suggestion is to link reservation process with a payment gate, thus the system should be safe from intrusive bookings and manipulation. This security feature needs to be embedded in the original code by setting internal timer in the processor on after reservation, after fifteen-minute time outs and an error message sent to the designated user. Yet adding payment feature could further ensure security for the system and the money collected is used for maintenance and development intentions.

This newly developed hardware sensing capability is an integration of many aspects of hardware and software at the sensors. The smart parking system will integrate the online reservation portal, controlling the reservation status, authentication of registered users, detection of unregistered users, and payment facility developed in the second stage which will be developed in the second stage. Further enhancement to the online reservation portal using a mobile application is highly recommended as this will simplify user reservations because majority of users uses their hand phone for internet more than personal computers and laptops (MERLE, Patrick et al., 2017), (Hern, 2015).

5. Conclusion

In a fast-moving world, IoT solutions helped a lot in changing life style of the people, finding an available parking slot is a time-consuming process but with smart parking system, the driver can easily spot how many slots are available from a distance.

Advancement in programming languages male applications portable (Chung et al., 2003) which means the software can be easily transferred from a platform to another (Medvidovic et al., 2000). Roughly every smart phone has internet access so the mobile application of the smart parking would allow the driver to check the status of the parking even before heading to the destination, booking or reservation can be made from the website or the app with ease.

This paper begins with the introduction which briefly mentioned the importance of IoT, parking issues, impact of the issue on users' daily life and proposed solution. Section two goes through related work to define what is done and is to be included to get the most of a smart parking system. Then the methodology elaborated in details the phases (hardware, programming, simulation, test and quality assurance) to achieve the final, integrated and working system. Methodology covered also the website algorithm to read and write data from and to the SQL database stored in a cloud service which works as a broker between the user's online platform interface and on-ground microcontroller to control the reservation procedures and publish real-time parking bays status. Prototype of the project developed and all scenarios tested results in a fully functioning smart parking system, snapshots provided under results and discussion section followed by recommendations and future work. Success of the prototype is a step towards real implementation in a scalable form, all the requirements are cheap and affordable which could consider as an economic smart parking system.

Acknowledgement

References

[1]   Van der Geer, J., Hanraads, J. A. J., & Lupton, R. A. (2000). The art of writing a scientific article. Journal of Science Communication, 163, 51–59.
[2]   Strunk, W., Jr., & White, E. B. (1979). The elements of style (3rd ed.). New York: MacMillan.
[3]   Alkheder, S. A., Al Rajab, M. M., and Alzoubi, K. (2016). Parking problems in Abu Dhabi, UAE toward an intelligent parking management system "ADIP: Abu Dhabi Intelligent Parking". Alexandria Engineering Journal, 55(3), 2679-2687.
[4]   Adki, P. R., and Agarkhed, J. (2016). Cloud assisted time-efficient vehicle parking services. Paper presented at the Inventive Computation Technologies (ICICT), International Conference on, 1-7.

[5] Atif, Y., Ding, J., and Jeusfeld, M. A. (2016). Internet of Things Approach to Cloud-based Smart Car Parking. Procedia Computer Science, 98(Supplement C), 193-198.

[6] Ahmad, F., Basit, A., Ahmad, H., Mahmud, S. A., Khan, G. M., and Yousaf, F. Z. (2013). Feasibility of deploying wireless sensor based road side solutions for intelligent transportation systems. Paper presented at the Connected Vehicles and Expo (ICCVE), 2013 International Conference on, 320-326.

[7] Allan, Alasdair (September 22, 2011). iOS Sensor Apps with Arduino: Wiring the iPhone and iPad into the Internet of Things (1st ed.). O'Reilly Media. p. 126. ISBN 978-1-4493-0848-3.

[8] Avago Technologies (2010). Safety Considerations When Using Optocouplers and Alternative Isolators for Providing Protection Against Electrical Hazards. January 2010. Retrieved November 5, 2010.

[9] Bachani, M., Qureshi, U. M., and Shaikh, F. K. (2016). Performance Analysis of Proximity and Light Sensors for Smart Parking. Procedia Computer Science, 83(Supplement C), 385-392.

[10] Benson, J. P., Donovan, T. O., Sullivan, P. O., Roedig, U., Sreenan, C., Barton, J., et al. (2006, 14-16 Nov. 2006). Car-Park Management using Wireless Sensor Networks. Paper presented at the Proceedings. 2006 31st IEEE Conference on Local Computer Networks, 588-595.

[11] "Chip Hall of Fame: Atmel ATmega8". IEEE Spectrum: Technology, Engineering, and Science News. Retrieved 2017-10-10.

[12] Caballero-Gil, C., Caballero-Gil, P., and Molina-Gil, J. (2016). Cellular Automata-Based Application for Driver Assistance in Indoor Parking Areas. Sensors, 16(11), 1921.

[13] Chabanne, Herve, Pascal Urien, and Jean-Ferdinand Susini. RFID and the Internet of Things. London: ISTE, 2011.

[14] Chung, J.-Y., Lin, K.-J., and Mathieu, R. G. (2003). Web services computing: advancing software interoperability. Computer, 36(10), 35-37.

[15] David Kushner (2011-10-26). "The Making of Arduino". IEEE Spectrum.

[16] Faheem, Mahmud, S. A., Khan, G. M., Rahman, M., and Zafar, H. (2013). A Survey of Intelligent Car Parking System. Journal of Applied Research and Technology, 11(5), 714-726.

[17] Geng, Y., and Cassandras, C. G. (2012). A new "smart parking" system infrastructure and implementation. Procedia-Social and Behavioral Sciences, 54, 1278-1287.

[18] Gupta, A., Sharma, V., Ruparam, N. K., Jain, S., Alhammad, A., and Ripon, M. A. K. (2014, 24-27 Sept. 2014). Context-awareness based intelligent driver behavior detection: Integrating Wireless Sensor networks and Vehicle ad hoc networks. Paper presented at the 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2155-2162.

[19] Hern, A. (2015). Smartphone now most popular way to browse internet– Ofcom report. The Guardian. Retrieved from http://www. theguardian. com/technology/2015/aug/06/smartphones-most-popularway-tobrowse-internet-ofcom.

[20] Karvinen, Tero; Karvinen, Kimmo (April 6, 2011). Make: Arduino Bots and Gadgets: Six Embedded Projects with Open Source Hardware and Software (1st ed.). O'Reilly Media/Make. p. 296. ISBN 978-1-4493-8971-0.

[21] Kaur, R., and Singh, B. (2013). Design and implementation of car parking system on fpga. arXiv preprint arXiv:1307.3051.

[22] Latiff, L. A., Ali, A., Chia-Ching, O., and Fisal, N. (2005, 17-20 July 2005). Location-based geocasting and forwarding (LGF) routing protocol in mobile ad hoc network. Paper presented at the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE'05), 536-541.

[23] Monk, Simon (November 15, 2011). Arduino + Android Projects for the Evil Genius: Control Arduino with Your Smartphone or Tablet (1st ed.). McGraw-Hill. p. 224. ISBN 978-0-07-177596-0.

[24] MERLE, Patrick et al. Computers, Tablets, and Smart Phones: The Truth About Web-based Surveys. Survey Practice, [S.l.], v. 8, n. 6, dec. 2015.   ISSN   2168-0094.   Available   at: <http://www.surveypractice.org/index.php/SurveyPractice/article/view/254>. Date accessed: 09 Nov. 2017.

[25] Medvidovic, N., Gamble, R., and Rosenblum, D. (2000). Towards Software Multioperability: Bridging Heterogeneous Software Interoperability Platforms. Paper presented at the Proceedings, Fourth International Software Architecture Workshop

[26] Noble, Joshua (January 30, 2012). Programming Interactivity (2nd ed.). O'Reilly Media. p. 726. ISBN 978-1-4493-1144-5.

[27] N. N. Bhargava and D. C. Kulshrishtha (1984). Basic Electronics & Linear Circuits. Tata McGraw-Hill Education. p. 90. ISBN 978-0-07-451965-3.

[28] Olasupo, T. O., Otero, C. E., Otero, L. D., Olasupo, K. O., and Kostanic, I. (2017). Path Loss Models for Low-Power, Low-Data Rate Sensor Nodes for Smart Car Parking Systems. IEEE Transactions on Intelligent Transportation Systems, PP(99), 1-10.

[29] S. Ananthi (2006). A text book of medical instruments. New Age International. ISBN 81-224-1572-5

[30] Timmis, Harold (November 9, 2011). Practical Arduino Engineering (1st ed.). Apress. p. 328. ISBN 978-1-4302-3885-0

[31] Liu, J., Mohandes, M., and Deriche, M. (2013). A multi-classifier image based vacant parking detection system.

[32] Uckelmann, Dieter, Mark Harrison and Florian Michahelles. Architecting the Internet of Things. Berlin: Springer, 2011.

[33] Vera-Gómez, J., Quesada-Arencibia, A., García, C., Suárez Moreno, R., and Guerra Hernández, F. (2016). An Intelligent Parking Management System for Urban Areas. Sensors, 16(6), 931.

[34] Wang, H., and He, W. (2011). A reservation-based smart parking system. Paper presented at the Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on, 690-695.

[35] Yadavalli, S. C. (2016). Smart Parking System: Kansas State University.

[36] Yang, C.-F., Ju, Y.-H., Hsieh, C.-Y., Lin, C.-Y., Tsai, M.-H., and Chang, H.-L. (2017). iParking – a real-time parking space monitoring and guiding system. Vehicular Communications, 9(Supplement C), 301-305.