Journal of Engineering Technology Vol. 10(1): 136-142,
2022ISSN 2231-8798© 2013 UniKL BMI

ONLINE PARKING SYSTEM USING ESP32

Mohd Alif Mohd Sofian, Lili Azwani Tiron
Bachelor of Telecommunication Engineering Technology With Honours, Universiti Kuala Lumpur, British Malaysian Institute, Gombak
E-Mail: alif.sofian@s.unikl.edu.my, liliazwani@unikl.edu.my,

ABSTRACT

Our world has been developing into a digital era since the introduction of technologies which is responsible for the evolution and changes in the lives of many people. However, there are still no development on the issue of parking system that is usually full especially in crowded areas to notify people on the absence of parking spots that ends up wasting their time. The anxiousness and worries of people results in their daily schedule change to fit in the time taken to look for a parking spot. The project that is made to be implemented focuses on a web-application to notify users on the availability of parking and the location of the parking. This project allows users to locate and identify available parking space with the view of just a web application. Light-dependant resistor(LDR) sensor is used to detect the presence and absence of car.The data transferred from the sensor to the web application is sent by using ESP32. The feature available in the web application is the indicator of an occupied parking space and a free parking. To enhance the effectiveness of the web application, we made it available to be used on website to ensure users are able to apply and use the system in real time.

Keywords: web application, ESP32, LDR sensor

1. INTRODUCTION

The idea came to light after an observation was made in daily life when going to crowded areas. Status quo shows that car parking areas do not have an indicator to indicate vacant parking space to individuals that are looking for a parking space even during this modern era technology-based society. Parking space has been a very crucial on current day and age since most individual owns a car, therefore there are an increase number of parking spots occupied, which makes it harder to find one especially when someone is in a rush. This is a global issue that is not addressed due to the perception of the issue considered as a non-primary objective to be focused on. The anxiety and worries of no available parking spot has a huge impact in people's daily lives to an extend of people rescheduling their daily time table to be able to secure parking to be on time for work [5]. With this project, we are able to help users by having an indicator system through a web-application so that people will be able to find parking spots at a faster rate especially in crowded areas.

Figure-1. Crowded Parking Areas which makes it hard to identify parking spaces

This project uses LDR sensor to identify and indicate free parking, sending the vacant or occupied signal to the microcontroller. Once the microcontroller receives the input, it will then light up an LED at the parking spot so that individual is able to identify free parking from afar, as well as sending the signal on webapp that is accessible by laptops or mobile phones [3]. The webapp will display the layout of the parking area, as well as the position of free parking to the users.

In addition, the webapp will also display the total amount of parking available, the number of occupied parking as well as the number of vacant parking [11]. This is to easily navigate users on the amount of vacant parking left since the parking area is huge and could easily miss out on any vacant parking.

The name of this project is Development Of Internet Of Things(IOT) Car Parking System Using ESP32 because we are using ESP32 as our microcontroller that works the backend of the project.

The problem arises for the project is that when we are entering a parking area, we do not know if the parking area is full or has a few vacant parking. It takes up a long time to circle the area looking for a parking spot that ends up at full parking.

Next, when a person exits the parking area, the other person who is waiting for a parking in the area is not notified and miss their parking. This has been an existing issue because people tend to demand for a bigger parking area because they could not find any parking that ends up taking more space rather than easily getting an update of available parking through their phones and plan their journey.

Lastly, the vacant parking position is not available for people who wants to park their cars in the area. Due to the uninformed parking position, people will waste their time circling around to find any good parking

which will end up causing a traffic jam assuming there are a lot of people looking for parking in the area.

2. 2.0 LITERATURE REVIEW

An occupied parking detection system is a system that is developed to identify a series of parking space and sends data back to alert whether the parking space is occupied or vacant. This system has been developed previously with different ways of implementations on the project. The difference between the projects existed using this system is the sensors used to detect the presence of vehicles and the ways each project utilize the data gathered from the system.

2.1 Home Automation Using ESP32, Arduino and XBee

In this research paper, it focuses on the connection of home automation to the internet. It is a simple and basic design which has a humongous functionality of monitoring and wireless control by only using the Internet. The basic approach on developing the project is to ensure all automation work can be controlled and status monitoring is available through the internet on mobile phones or computers [1]. Home automation has been booming and developing since the past decade and it has been hard to keep track and control the existing automation. Creating an online system to monitor and control will save numerous times to notice system failure and status [2]. In this project, all electronic equipment that is compatible with the system will be able to connect and display its status through the home automation device and is able to be control remotely.

Figure 2 shows the home automation system with wireless control

From the project above, there are a few downside that can be improved. Firstly, the project uses two microcontroller which is not ideal. It complicates the project into linking the ports and data from arduino to the ESP32 to be sent wirelessly and enable wireless control. One of the changes can be made is by using only one microcontroller to unify the system and reduce the total amount of device used which can reduce the cost of the project as well. Secondly, the devices data sent to the microcontroller is wireless. This is a bit problematic for houses that has a bigger range and needs a few Wi-Fi to enable the other devices to connect. One improvement that can be made is by sending data to the microcontroller without wireless signal. This would reduce the needs of using Wi-Fi and only connects the Wi-Fi with the microcontroller and enables a more stable connection and control from the remote to the other devices.

2.2 Application of Wireless Internet Networking using NodeMCU and Blynk Application

Well known arduino projects that connects via wireless network usually uses the ready-to-use Blynk Application. Blynk is an application which enables user to control projects related to Arduino, and Raspberry Pi to connect to the Internet and having control over them. Blynk service is offered on iOS and Android platform which is available for both android and iphone users. It is an application that allows users to build a digital dashboard and freely control the graphic interface by using the infamous drag and drop widgets. Moreover, Blynk application is very flexible as it is not tied to any type of hardware devices. Whereas, NodeMCU is known as a controller unit which is known as a very cheap open source IoT platform.It is able to run any wireless hardware related especially if it is based on the ESP module.

This project studies the connection between NodeMCU and Blynk application on how responsive, good, and limitations. NodeMCU offers Wi-Fi network in order for hardware to connect to Blynk application. Projects made and analyzed in the research includes Tiny Internet Weather Station and Wireless Light Control.

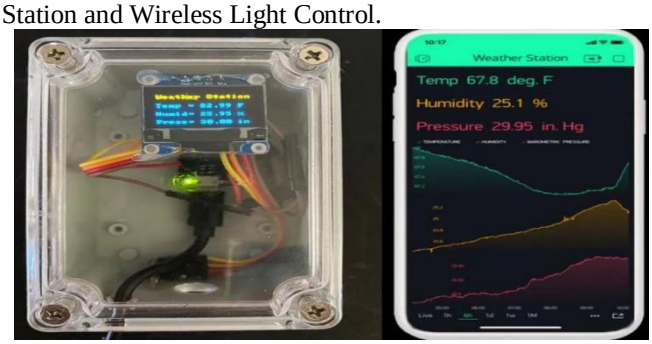


Figure 3 shows the Tiny Weather Station Project

From the project above, the major problem is the usage of blynk application. Blynk is a powerful tool to make a prototype, but it is a web server that depends on the status of the host. If the host of Blynk application breaks down, we cannot access the data wirelessly even though the project is successfully made.

2.3 Performance Analysis of Proximity and Light Sensors for Smart Parking

This project is the evaluation of the effectiveness between proximity and light sensor. Proximity sensor is a sensor that is able to detect objects around the area without the need of physical contact. The sensor emits an electromagnetic field or an infrared radiation around the sensor. When the field

observes a change by having interruption on the radiated field, it will send an indication to indicate there is an interference, whether an object or a different magnetic field distorting the emitted signal. Whereas light sensor is a device that detects light energy which is photons or infra-red light and emits an electrical signal if light is present.The sensor sends an output signal to indicate the intensity of light received by measuring the radiant energy exist in light.

Figure 4 shows the car parking project using proximity sensor

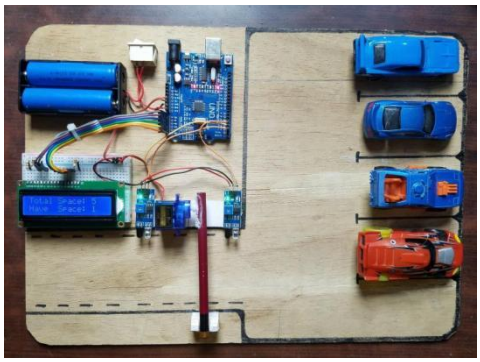


Figure 5 shows the car parking project using ldr sensor

From the research paper above, I have tested and compare the effectiveness of sensors that is available to be used for my project. PIR sensor is not suitable since it sends and receive analog data, which needs extra coding to convert the input and data to digital. However, proximity sensor is the best sensor to be used but it is too expensive to be made for a project since this project is made with a prototype and not suitable to use an industrial sensor. Therefore, it comes down to two sensors, which are LDR sensor and ultrasonic sensor. The reason I dismiss ultrasonic sensor is due to the fact that it is more prone to noise and external environment. If a trolley is parked at the parking, it will also detect as occupied which is technically is not a vehicle and can be moved.

3.0 Methodology

The development of IoT Car Parking System is made with an intention to provide a layout of parking space in the area to save time of users from spending too much time on searching for a parking space. This is a daily routine especially for individuals who goes to work everyday and need to find an empty parking space every single time.The approach of this project is fully automated, users only need to access the web application to view the available and occupied parking space and decide which parking space can be entered. The components of this project are detailed in this chapter including a detailed explanation of planning and testing to ensure system's efficiency and effectiveness.

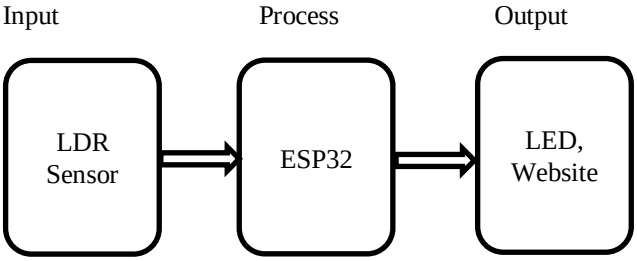
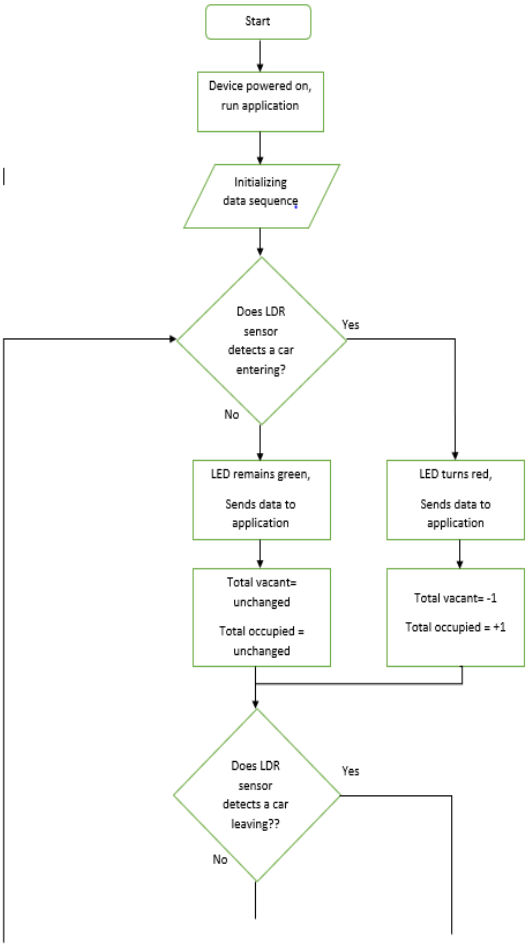


Figure 6: Block Diagram for the parking system

The block diagram above as shown in figure 7, consisting of input, process and output of the project. The LDR sensor will detect the availability of cars in the parking space. It is considered as an input. From the sensor, it will generates an electrical signal and sends it to the ESP. If there is a car present at the parking space, ESP32 will switch on the LED light to identify if the parking is occupied and sends a signal to the website. ESP32 will then send the data received from the LDR sensor and display the data from the LDR sensor to the web application.



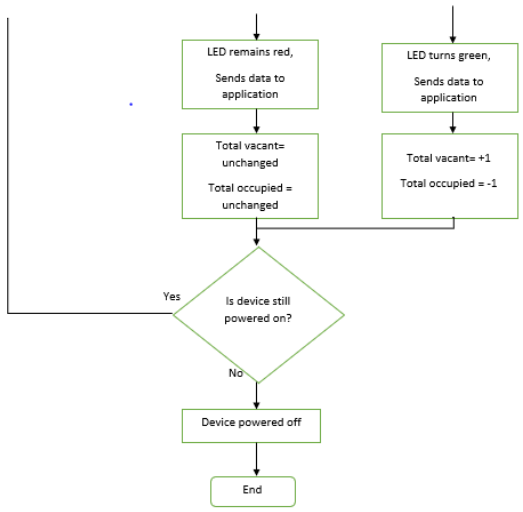


Figure 7 shows the process flowchart of the project

From the flowchart on figure 7, the system starts when it is powered on. When there is no vehicle at the parking area, it will lit up the green LED indicating a vacant parking area. When the parking space is occupied by a vehicle, the LED will change its colour to red. This is because LDR sensor sends a signal to the ESP32 indicating there are no lights detected at the designated space. ThenESP32 will send the data to the website and indicate the LDR does not sense any light to the web application. Whereas when a car leaves the designated space, LDR will detect light source an sends the data to the ESP32. ESP32 will then switch off the red LED and switch on the green LED to indicate the parking area is vacant. It then sends the data to the web application to indicate there is a light source on the LDR sensor.

3.1 Hardware Development

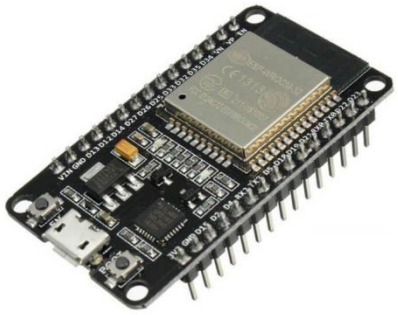


Figure 8: ESP32

ESP32 is a microcontroller with an integrated Wi-Fi and Bluetooth connection that is able to be used as a wide range communication between devices. It has a small design which can withstand temperature from -40 Celsius to 125 Celsius. It also consumes very low power with a requirement of 5V.

Figure 9: LDR sensor

LDR sensor based on figure 10 is a very important component in this project. LDR sensor which is also known as light dependent resistor sensor is a device that is sensitive to light which always used to indicate the presence or absence of light. It is also capable to measure the intensity of light to determine the brightness and dimness of an area. It has a high resistance component up to 1M ohm, which will drop if the sensor is exposed to a light source. The resistance drops depending on the intensity of light exposed to the sensor.

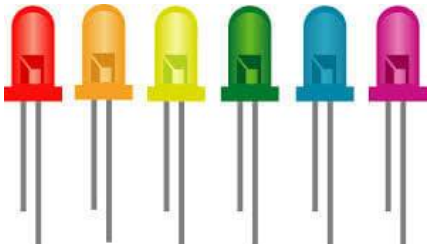


Figure 10: LED

Led is a light source that is usually used as an indicator in projects.LED produces light up to 90% efficiency which is very efficient compared to an incandescent light bulbs. It lights up when an electrical signal passes through a microchip inside the LED which produces a tiny light source resulting to a visible light.

4 Results

4.1 LDR Sensor Sensitivity

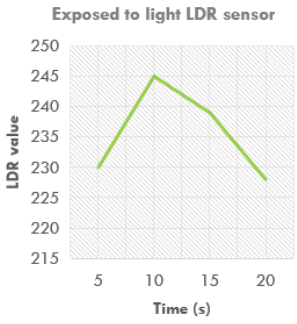


Figure 11 shows the LDR reading when exposed to room lighting

From figure 11, the LDR sensor has a reading range of 200 and 250 when the room is lighten up with a light source. The reading does not drop below 200 and does not go

above 250 due to the constant lighting.

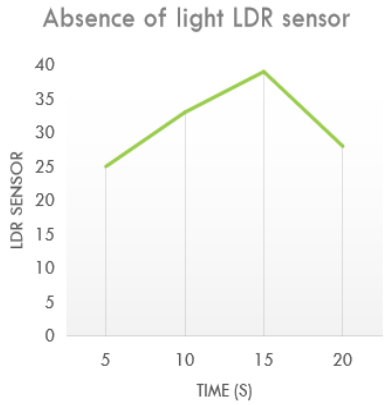


Figure 12: LDR sensor reading when light is blocked

When LDR sensor is blocked from any light source, the value will go down until 0. Since the project blocks light source from an actual car, the value ranges from 20 to 50 in a controlled environment when testing the project.

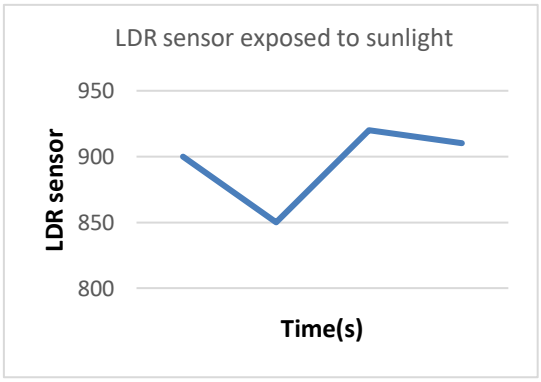


Figure 13 shows the LDR sens reading when exposed to sunlight

When the LDR sensor exposed to sunlight, the value spikes up in range of 800 up to 1000. this is due to the brightness from the sunlight is greater compared to light source that we normally use inside a closed environment.

Figure 14 shows the homepage of the webserver

From the figure above, it is the homepage I designed using html, css and javascript in order to pick a location to access the web page that is connected to the project. The homepage is designed for future projects to add on several other parking spaces and it is easier to choose the location by just selecting the drop-down button.

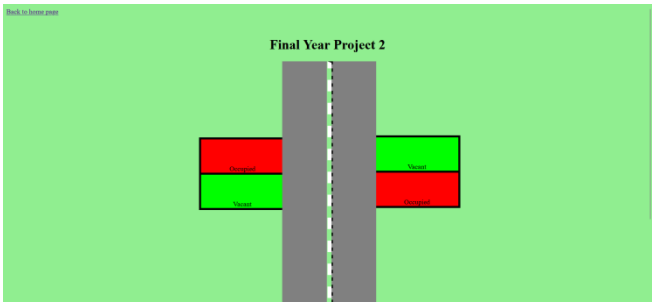


Figure 15: Parking Layout

The above figure shows the layout of the parking I designed by using html, css and javascript. The layout resembles the actual project including having the similar position

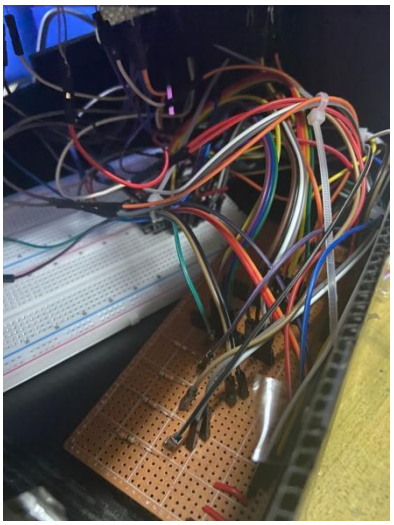


Figure 16 shows the circuit on the stripboard

The circuit above is soldered and designed based on the layout of the parking on the website. ESP32 is used on the breadboard is because a few of the pin-header that is soldered is not working as expected. There are a lot of wires because I use 8 LEDs and 4 LDR sensors, which requires 2 pin each to connect from the I/O pin to the ground, in total of 24 pins.

Figure 17 shows the project model of the parking system

The project model I created is a square black box, with each green LED, red LED, and an LDR sensor is group together in 4 groups to resembles a parking spot each. It is arranged based on the layout I designed on the web server.

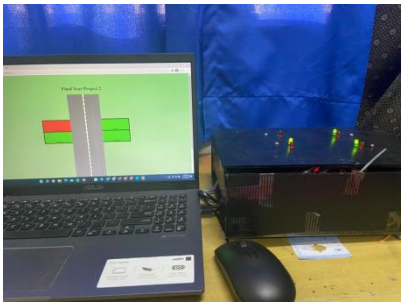


Figure 18 shows the functionality of the project

When the project is turned on, it will be linked simultaneously with the web server. From the figure above, the green LED lights up according to the layout of the web page. The top-left parking has a non-functioning LDR sensor due to the circuitry problem happened during the process.

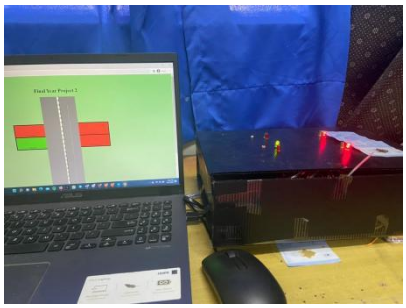


Figure 19 shows the functionality of the project(2)

Figure above shows what happens when something blocks the LDR sensor from detecting the presence of light. From the result above, I cover the LDR sensor with a ringgit note to block it from reaching any light source. The green LED switches off while the red LED switches on, and the colour of the parking on the web page changes to red due to the blocking of light source at the LDR sensor. This shows that the project is fully functioning and the web page changes colour in accordance of the physical prototype I made.

5. Discussion

From the project above, we are able to achieve the results we wanted based on our objectives. Although there are a few miscalculations and unexpected situation, I am able to make the project and the prototype as close as possible.

Firstly I managed to create a hardware system using ESP32, LEDs, and LDR sensor. ESP32 is used for the I/O input/output pin which is code by using Arduino IDE. LDR is able to detect light source while green LED and red LED reacts accordingly to the LDR sensor. One of the problem is the circuit since it requires too many wires, it ends up costing a parking space to be malfunctioned and needed repair.

For the web server and web page, I managed to create a web server that has a home page and a layout page. The homepage will redirect towards the parking layout according to the location chosen on the homepage.

Lastly, from the results above, I manage to successfully make a project that answers to all of the objectives which is to use LDR sensor for my parking detection system, LED to indicate the availability of parking in physical form, and a parking layout that displays the parking in accordance to the input of the LDR sensor.

6. Conclusion

In this project, we aim to create an online parking system where we are able to detect the availability of a parking, and display the data received on a web page in accordance to the layout designed. LDR sensor is used to detect the availability of a parking by sensing the presence of light, while green and red LED is used to indicate the parking physically.

When the LDR sensor detects an absence of light, it will turn on the red LED while switching off the green LED. Whereas if the LDR sensor detects the presence of light, greed LED will turn on instead and the red LED will be turned off. While this is happening, the web page will display the data received from the LDR sensor via the webserver of ESP32.

ESP32 will send the data to the web page, and the web page will change the colour of the parking based on the layout and the designed set. If LDR sensor sense the presence of light, the parking layout in the web page will display the parking spot as green as an indicator for available parking online, whereas if LDR sensor detects an absence of light, it will change the colour of the parking to red to indicate that the parking is occupied.

For further improvement of this project, one of the things that can be added on is the detection of disabled parking and women only parking by using RFID. This will make things easier to detect if cars parked at the disabled spot and women only spot has registered for the parking.

Moreover, this project can be improved more by simplifying the circuit to use less wires, by using rgb light to change the colour of the light instead of using 2 LEDs, and by using a relay to reduce the amount of input and output of the project.

7. Reference

[1] Suresh S., J. Bhavya, S. Sakshi, K. Varun and G. Debarshi, "Home Monitoring and Security system," 2016 International Conference on ICT in Business Industry & Government (ICTBIG), 2016, pp. 1-5, doi: 10.1109/ICTBIG.2016.7892665

[2] C. Visvesvaran, S. Kamalakannan, K. N. Kumar, K. M. Sundaram, S. M. S. S. Vasan and S. Jafrrin, "Smart Greenhouse Monitoring System using Wireless Sensor Networks," *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, 2021, pp. 96-101, doi: 10.1109/ICOSEC51865.2021.9591680.

[3] S. P. Parikh, D. R. Shah and P. R. Shah, "Park Indicator: Book Your Parking Spot," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-3, doi: 10.1109/ICCUBEA.2018.8697378.

[4] A. Anand, A. Kumar, A. N. M. Rao, A. Ankesh and A. Raj, "Smart Parking System (S-Park) – A Novel Application to Provide Real-Time Parking Solution," 2020 Third International Conference on Multimedia Processing, Communication & Information Technology (MPCIT), 2020, pp. 93-96, doi: 10.1109/MPCIT51588.2020.9350429.

[5] C. Seo, J. Kim, Y. Lee and W. -Y. Kim, "Vision-based Approach in Finding Multitype Parking Stall Entrance," 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC), 2018, pp. 352-357, doi: 10.1109/ICNIDC.2018.8525525.

[6] S. Shih and W. Tsai, "A Convenient Vision-Based System for Automatic Detection of Parking Spaces in Indoor Parking Lots Using Wide-Angle Cameras," in IEEE Transactions on Vehicular Technology, vol. 63, no. 6, pp. 2521-2532, July 2014, doi: 10.1109/TVT.2013.2297331

[7] S. -H. Kim, J. -S. Kim and W. -Y. Kim, "A method of detecting parking slot in hough space and pose estimation using rear view image for autonomous parking system," 2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC), 2016, pp. 452-454, doi: 10.1109/ICNIDC.2016.7974615.

[8] P. Sadhukhan, "An IoT-based E-parking system for smart cities," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 1062-1066, doi: 10.1109/ICACCI.2017.8125982.

[9] C. O. Martinez-Ojeda and J. C. Dela Cruz, "Photoplethysmographic Detection of LDR And Photodiode In Varying Distance of Light Source To Area," 2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), 2020, pp. 189-194, doi: 10.1109/I2CACIS49202.2020.9140179.

[10] I. Allafi and T. Iqbal, "Design and implementation of a low cost web server using ESP32 for real-time photovoltaic system monitoring," 2017 IEEE Electrical Power and Energy Conference (EPEC), 2017, pp. 1-5, doi: 10.1109/EPEC.2017.8286184.

[11] P. Macheso, S. Chisale, C. Daka, N. Dzupire, J. Mlatho and D. Mukanyirigira, "Design of Standalone Asynchronous ESP32 Web-Server for Temperature and Humidity Monitoring," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 635-638, doi: 10.1109/ICACCS51430.2021.9441845.