IoT-Based Smart Parking Management System Using ESP32 Microcontroller

Joni Welman Simatupang, Aida Mahdalena Lubis
Study Program of Electrical Engineering
President University
Cikarang 17530, Indonesia
joniwsmtp@president.ac.id, aidamahda579@gmail.com

Vincent
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung 40132, Indonesia
23221049@std.stei.itb.ac.id

***Abstract* — In recent years, the number of vehicles has increased significantly. As a result, drivers waste a lot of time looking for an open parking spot or queuing to enter one. It leads to fuel waste and raises air pollution at the same time. One solution is to implement an innovative system that provides a comprehensive parking solution. This work focuses on designing a smart parking management system that allows drivers to see the parking slots, then reserve a certain slot before arriving to the parking area. It is a web-based system where the drivers do not have to download any mobile application. Practically, the system implemented ultrasonic sensors to check slot availability. In addition, an indicator is installed at each slot to differentiate the one that is vacant, reserved, and occupied. They are controlled by ESP32 modules, and the data was sent wirelessly by using MQTT protocol. Quantitatively, the ultrasonic sensors are set to have 100 cm of distance as the threshold to differentiate vacant and occupied slots. After experiments, it was found that the sensor has an accuracy of 94.72%. However, there is an approximately 10 seconds of delay since the reservation is made until it can be successfully confirmed by the system. Finally, the system also informs the drivers of how long they have been parking, as well as the parking fee based on the duration of time.**

Keywords—Microcontroller, ESP32, IoT, MQTT, Smart Parking

I. Introduction

In recent years, the number of vehicles has increased significantly. Based on the data, the number kept increasing annually. In the beginning of 2000, there are approximately 5 million registered vehicles. Within two decades, there are approximately 4 times of this value, reaching 21 million and is still increasing [1].

One direct implication of this huge number is the parking area, which must be able to serve at least the same amount of the vehicle. However, it does not expand easily for a certain venue (e.g., shopping malls). In this case, the parking area might be filled quickly, leaving no vacant spot for a lot of drivers. At the same time, they do not have prior knowledge about it, thus they waste a lot of time seeking for an empty slot, or simply just simply queuing. In fact, 86% drivers found it hard to locate a vacant parking slot, especially in a place with multilevel [2]. It became even worse when it came to weekends or public holidays, where 66% of drivers spent additional ten minutes simply to find a place to park the vehicle [3]. In short, insufficient parking slots result in more time wasted to locate a vacant parking slot. Additionally, traffics became congested, drivers were frustrated, gasoline consumption increased, and gas pollution worsened at the same time [4], [5].

The Internet-of-Things (IoT) has been able to solve problems in many areas [6]–[9]. Its ability to have the information transmitted over the Internet has been very helpful in information sharing, as well as remote controlling. For instance, the information from a sensor in one location can be sent to remote users relative instantly. At the same time, the users can provide feedback in a form of a signal to control the actuator of the system.

Therefore, one solution for the problem comes in the form of parking management system. It helps the visitors by providing information of vacant spot to park the vehicle in a certain place (e.g., stadium, shopping malls). This way, the time spent to get to the parking spot is minimized. As the effect, the gasoline consumption gets more efficient.

This work focuses in designing a low-cost smart parking management system. The key point is to inform the vacant slots to the drivers and allow them to reserve a certain slot to ensure its availability. In addition, the parking space is equipped with indicator to let drivers know whether the slot is already filled, is already reserved by other drivers, or if it is still vacant and not reserved yet. It is combined with the IoT to simplify and extend the range of usage, where the drivers are able to access the system via their smartphone simply with the Internet connection. Finally, the system is designed to have low-power consumption components, while maintaining the performance at the same time.

The rest of this paper is organized as follow. Section 2 provides the survey of previous works done by other researchers within similar topic of interest. Next, section 3 elaborates the methodology of this work. It covers the system design, as well as the flow of the system. The result and its discussion are then explained in section 4. Finally, the conclusion is drawn in section 5.

II. Related Works

This section focuses on the survey of related works done by other researchers within similar topic of smart parking system. There are at least five other works, and all of them are discussed in this section.

Author in [10] focused in the implementation of Bluetooth in the system. It utilized Arduino, and ultrasonic sensors to detect the availability of a space, then sent the notification via Bluetooth connection. In addition, an LED indicator was installed to inform drivers of any vacant slot. The drawback in this work comes in term of range, where Bluetooth connection is strictly limited to a few meters only. Author in [11] improved the usage range by using GSM with SMS service. Besides, the system installed infrared sensors as vacant slot detector. The sensors were controlled by Arduino. Even though this sensor is able to detect rough surface better than the ultrasonics, it consumes more power. The system also did not have any indicator for the slot. Next, author in [12] worked on IoT implementation on the system. In this case, the Arduino had NodeMCU ESP8266 connecting the system to the Internet. Besides detecting vacant slot using ultrasonic sensors, the system was extended to be able to control the parking space gate as well. The application must be downloaded before the driver can be served to the system. After that, author in [13] also implemented IoT in the system. In this case, the Arduino controlled infrared sensors as slot detector, and the drivers had to get into the mobile application before they can be served by the system. In addition, there were no indicator for the vacant slot. Finally, author in [14] combined the functionality of Bluetooth and IoT for the system. It was able to send low power signal to the driver, so they were well notified of vacant slot. Then the system automatically sent the information of filled slot via Internet, so then the other drivers knew which slots were and were not available.

There are four main parameters taken as comparison to previous works. The first one is whether the system utilized IoT or not. Then the cost of the system is also considered. Specifically saying, it considers the power consumption. It was highlighted that infrared sensors consume more power than the ultrasonics. Besides, having two microcontrollers (i.e., Arduino and NodeMCU ESP8266) is considered to be inefficient since the Internet connection can be established by simply having the ESP8266 Wi-Fi module instead of using the NodeMCU ESP8266. In this parameter, a dash represents system that is low in cost, single tick represents moderate cost, while double ticks represent high cost. Next, the other parameter is whether the system utilized a website for the system or not. All previous works did not have this, so they have a dash for it. Finally, only the first reference work had indicator for the slot. Table I summarizes the comparison of this work with the five previous works.

TABLE I. COMPARISON STUDIES ON SMART PARKING LOT

Parameter	**References**					**Present Paper**
	[6]	[7]	[8]	[9]	[10]	
IoT	–	–	✓	✓	✓	✓
Cost	✓	✓✓	✓	✓✓	–	–
Website	–	–	–	–	–	✓
Indicator	✓	–	–	–	–	✓

Compared to the previous works, the Author implemented IoT to have widest coverage of usage, so the driver can access the system as long as they have an Internet connection. The system implemented components with low power consumption, i.e., ultrasonic sensors and ESP32, which is sufficient in controlling the system while being able to establish a connection to the Internet at the same time. The system implemented RGB LED to indicate three different statuses of slot. Specific discussion of the system will be done in the next section. Finally, website-based system was chosen instead of mobile application, because it would deliver a more convenient experience to the driver. In this case, the driver does not have to download an application before they can access the system.

III. SYSTEM DESIGN

This section elaborates the system design, including the brief overview of the system, the hardware implementation, settings for the sensors, as well as the flowchart of the system.

In this work, the system was made as a prototype compromising three ultrasonic sensors and three RGB LEDs. Each sensor and LED are installed for one slot, thus there are three slots in this prototype. One ESP32 was programmed to control all ultrasonic sensors, while another ESP32 was programmed to control the RGB LEDs. Both ESP32s are connected to the MQTT broker, where all of the data were exchanged there. The drivers as the users were able to connect to the system via a webpage, which can be accessed with this link "http://cpms-apps.herokuapp.com/".

For the system implementation itself, Table II shows the pin configuration of the two ESP32. Since both ESP32 were communicating with each other via the data in MQTT broker, an LCD was installed along in the second ESP32 to show the update of the system. After the implementation of the wiring diagram, the prototype is shown in Fig. 1.

TABLE II. PIN CONFIGURATION

ESP32	**GPIO Pin**	**Function**
#1	13	Trigger pin Ultrasonic #1
	12	Echo pin Ultrasonic #1
	27	Trigger pin Ultrasonic #2
	26	Echo pin Ultrasonic #2
	33	Trigger pin Ultrasonic #3
	32	Echo pin Ultrasonic #3
#2	13	LED #1 (red pin)
	12	LED #1 (green pin)
	14	LED #1 (blue pin)
	27	LED #2 (red pin)
	26	LED #2 (green pin)
	25	LED #2 (blue pin)
	33	LED #3 (red pin)
	32	LED #3 (green pin)
	35	LED #3 (blue pin)

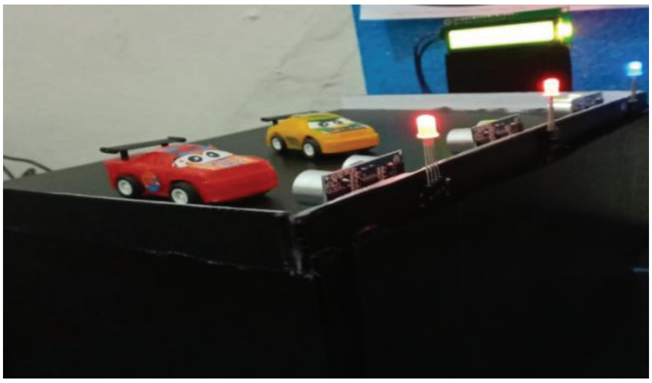

Fig. 1. Prototype of the System

The first ESP32 is set to have 100 cm as the threshold to distinguish a vacant slot. If an ultrasonic detects a vehicle with a distance of lower than 100 cm, then the slot is considered to be occupied. Otherwise, the slot is vacant. The closest distance the sensor can read is set to be 2 cm. In this case, a vehicle with 2 cm distance will be very close to the sensor, and 100 cm will be sufficiently far to serve as the threshold. The second ESP32 has multiple RGB LEDs. Each LED is set to emit green light by default, indicating a vacant slot. If then the slot is reserved, then the ESP32 sets the slot's LED to blue. Finally, if a vehicle parks in a slot –either it was reserved or not–, the slot's LED will set to turn into red, indicating occupied slot. Table III shows the look up table of the three conditions for each slot.

TABLE III. SYSTEM LOOK UP TABLE

Slot	Range (cm)	Availability	LED
1	> 100	Available	Green
	2 – 100	Reserved	Blue
	2 – 100	Occupied	Red
2	> 100	Available	Green
	2 – 100	Reserved	Blue
	2 – 100	Occupied	Red
3	> 100	Available	Green
	2 – 100	Reserved	Blue
	2 – 100	Occupied	Red

Next, the discussion for the flowchart has three sections. The first one focuses to the reservation scenario, where the driver can access the data of the parking slot and reserve a certain slot. The next one focuses to the vacant slot detector. The last one focuses to the indicator of the slot.

Fig. 2 shows the flowchart for the reservation scenario, which come in the form of web page. It is also accessible by the driver, which can be either customer or administrator of the parking space. First of all, the user must have an account. Otherwise, they have to register for one. After that, they can log into the system. If the account is classified as an administrator, then they can access the data in the system, and do the analysis if it is necessary. Otherwise, it must be a customer, so the system will direct the user to the page showing the parking slots. This page is linked to the system, so if there is any update, the data will be updated automatically, and the users have real-time data for the service. Then, they can reserve a vacant slot here. If an error occurred and making the reservation unsuccessful, the user will be redirected back to the parking slot page. If the process is successful, then the system will update the status of the specific slow, and the user can come to the place directly. In this case, the user cannot reserve two slots, so after a reservation is successful, they can only see the webpage without being able to reserve another slot.

Next, Fig. 3 shows the flowchart for the vacant slot detector. This part of the system has the first ESP32 as the controller of the ultrasonic sensors, and it also establishes the Internet connection to transfer the readings of the sensors to the MQTT broker. If any of the ultrasonic detects a car in the slot, the ESP32 will update the information in the broker for the specific slot.

Finally, Fig. 4 shows the flowchart for the indicators. This part of the system has the second ESP32 as the controller of all indicators. In addition, it also establishes the Internet connection, so it can receive updates for each slot. By default, all indicators are set to green, indicating vacant slots. In the case of a reserved slot, the updated data on MQTT broker will be accessed by the second ESP32, so that the slot's indicator is set to be blue. Later, if a slot –either it has been reserved or not– is occupied, the first ESP32 will detect the vehicle and send the update to the system, so the second ESP32 will be well-notified of the update, and set the indicator to red.

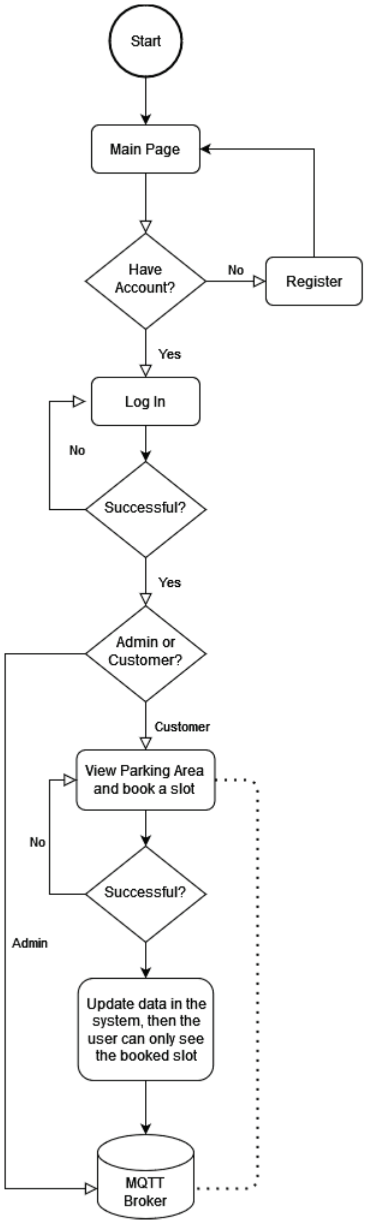

Fig. 2. Flowchart for the reservation scenario

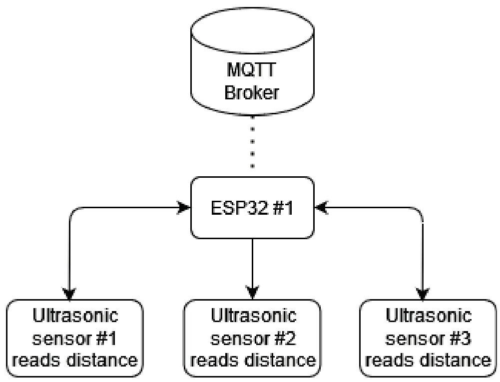

Fig. 3. Flowchart for vacant slot detector

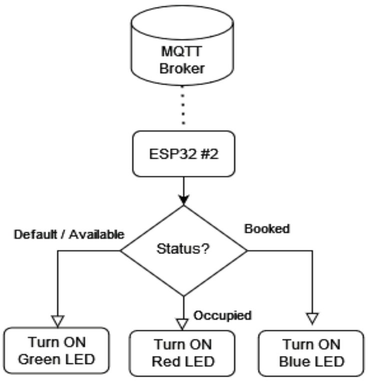

Fig. 4. **F**lowchart for the indicator of the slot

IV. Result and Discussion

Table IV shows the trial result of each ultrasonic sensors. The trial was conducted for five different distances, viz. 2 cm, 25 cm, 50 cm, 75 cm, and 100 cm. Three trials were done for each distance. After the three trials, the average was drawn, and then compared to the reference. Finally, the accuracy is calculated for each case.

Fig. 5 shows the statistical summary of the trial result. For the distance of 2 cm, the average success rate is shown to be 77.78%. For distance of 25 cm, the accuracy is 97.33%, significantly higher than 2 cm. Then for the distance of 50 cm, the accuracy is 99.56%. As for the distance of 75 cm, the accuracy is 99.70%. Then, for the distance of 100 cm, the accuracy is 99.22%. Besides, seeing from the perspective of each ultrasonic sensor, the first sensor is noted to have accuracy 96.27%, while the second and the third ultrasonic sensor have the accuracy of 95.84% and 92.04%, respectively. Overall, the success rate is noted to be 94.72%.

It is noticeable that each sensor has approximately similar accuracy. However, when the evaluation was run for low distance, especially lower than 25 cm, the accuracy is noticeable low. Even though the accuracy did not differ much, it was at peak when the distance is 75 cm instead of 100 cm.

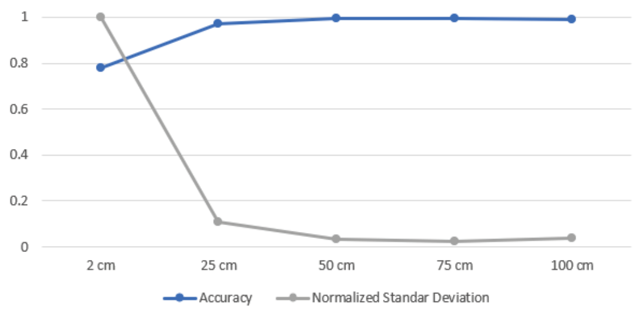

Fig. 5. **S**tatistical summary of ultrasonic sensors test

Besides, the standard deviation of each distance was also calculated, and is depicted in Fig. 5. As a note, the graph of standard deviation is normalized to 1 at the distance of 2 cm. In this case, the standard deviation at other distance is relative to that value. Without it, the graph would be unreadable due to its tiny value compared to the accuracy. It was found that the measurement of 2 cm has a standard deviation of 0.2635. For the distance of 25 cm, 50 cm, 75 cm, and 100 cm, the standard deviation was 0.0283, 0.0088, 0.0059, and 0.0097, respectively.

Even though the distance of 75 cm provides the best accuracy and lowest standard deviation, the sensor has 99.22% of accuracy and 0.0097 is standard deviation when it was measuring 100 cm. It is considerably ample, especially because the distance between the ultrasonic sensor and the parked vehicle is considerably close. In fact, a parking spot is normally designed to be as small as possible to have maximum allocation. Additionally, the ultrasonic sensors were installed by the end of each slot. Thus, having the threshold at 100 cm leads to a faster vehicle detection, allowing a faster information update to the system. Regardless of the accuracy of the sensors, the measurement readings are apparently sufficient for the implementation, considering the low variation of them all. In addition, the misreading between slots is considered negligible due to the separation distance between sensors (which can go up to a few meters) as well as the low beamwidth of each sensor (i.e., 30^{o}).

TABLE IV. Ultrasonic Trial Results

Ultrasonic	Exact Value (cm)	Measured Distance (cm) Trial 1	Trial 2	Trial 3	Average (cm)	Accuracy
A	2	2	3	2	2.33	83.33%
B	2	2	2	3	2.33	83.33%
C	2	3	2	3	2.67	66.67%
A	25	25	26	25	25.33	98.67%
B	25	26	25	26	25.67	97.33%
C	25	27	26	25	26.00	96.00%
A	50	50	50	50	50.00	100.00%
B	50	50	51	50	50.33	99.33%
C	50	51	50	50	50.33	99.33%
A	75	75	75	75	75.00	100.00%
B	75	75	75	76	75.33	99.56%
C	75	75	75	76	75.33	99.56%
A	100	100	99	99	99.33	99.33%
B	100	99	100	100	99.67	99.67%
C	100	97	99	100	98.67	98.67%

TABLE V. RESERVATION SIMULATION RESULTS

Slot	Range (cm)	Status	Indicator	Info	Result
1	> 100	Vacant	Green	Default Condition	Correct
2	> 100	Vacant	Green	Default Condition	Correct
3	> 100	Vacant	Green	Default Condition	Correct
1	> 100	Reserved	Blue	Add Command to System	Correct
1	2 – 100	Occupied	Red	Vehicle Detected	Correct
2	> 100	Reserved	Blue	Add Command to System	Correct
2	2 – 100	Occupied	Red	Vehicle Detected	Correct
3	> 100	Reserved	Blue	Add Command to System	Correct
3	2 – 100	Occupied	Red	Vehicle Detected	Correct

Fig. 6. Webpage to reserve a slot

Next, the reservation was evaluated by doing a simulation. The result of it is shown in Table V. The simulation was conducted by first starting the system without any vehicle in the parking area. In this case, all three slots are shown to have green LED lighted up, indicating available slots. When the first slot was booked, the data in the system was updated, so the indicator was changed to blue LED. The display of the webpage when the driver was booking the first slot is shown in Fig. 6. Then, when the vehicle was detected at the slot, the ultrasonic received the signal, update the information in the system, and the indicator was set to red. The simulation was also repeated for the second and the third slot, and the similar result was found. Finally, when the driver decided to check out, the duration of parking could be seen at the system, and the parking fee was calculated automatically.

However, there was a delay of approximately 10 seconds when a reservation was made by a driver. It implies a 10 seconds gap before the indicator's color is changed. In this case, the driver who did not make any reservation will not be aware of the reserved slot during this gap.

V. CONCLUSION

As the conclusion, a smart parking management system is able to deliver a service by allowing drivers to reserve a specific slot before going to the parking area. Drivers do not have to download a specific mobile application. Instead, they could access the website to make the reservation. The system also has an indicator for each slot, letting the drivers know whether the slot is vacant, reserved, or occupied. The ultrasonic sensors are set to have 100 cm of threshold distance to differentiate vacant and occupied slots. After prototype testing measurements, it was found that the sensor has an accuracy of 94.72%. Also, there is an approximately 10 seconds of delay since the reservation is made until it can be successfully confirmed by the system. Finally, the system also informs the drivers of how long they have been parking, as well as the parking fee based on time duration.

As for the future work, the system should have a protocol to solve the case of having unauthorized vehicle attempting to park in the reserved slot. It may be done by allocating a maximum duration between slot reservation and the parking. If the driver takes too long, then the reservation will be cancelled, and the slot will be fully available. Another way is to apply a penalty fee for those who trespasses.

REFERENCES

[1] CEIC, "Number of Registered Vehicle in Indonesia," 2021. https://www.ceicdata.com/en/indicator/indonesia/number-of-registered-vehicles (accessed on Oct. 02, 2021).

[2] A. Kianpisheh, N. Mustaffa, J. M. Y. See, and P. Keikhosrokiani, "User Behavioral Intention Toward Using Smart Parking System," in *Communications in Computer and Information Science (CCIS)*, 2011, pp. 732–747, doi: https://dx.doi.org/10.1007/978-3-642-25453-6_61.

[3] D. B. L. Bong, K. C. Ting, and K. C. Lai, "Integrated Approach in the Design of Car Park Occupancy Information System (COINS)," *IAENG Int. J. Comput. Sci.*, vol. 35, no. 1, 2008.

[4] M. Y. I. Idris, E. M. Tamil, N. M. Noor, and K. W. Fong, "Parking Guidance System Utilizing Wireless Sensor Network and Ultrasonic Sensor," *Inf. Technol. J.*, vol. 8, no. 2, pp. 138–146, 2009, doi: https://dx.doi.org/10.3923/itj.2009.138.146.

[5] A. Kianpisheh, N. Mustaffa, P. Limtrairut, and P. Keikhosrokiani, "Smart Parking System (SPS) Architecture Using Ultrasonic Detector," *Int. J. Softw. Eng. Its Appl.*, vol. 6, no. 3, pp. 51–58, 2012.

[6] M. Galina, M. W. Ramadhani, and J. W. Simatupang, "Prototype of Postpaid Electricity and Water Usage Monitoring System," in *ICSECC 2019 - International Conference on Sustainable Engineering and Creative Computing: New Idea, New Innovation, Proceedings*, 2019, pp. 304–308, doi: 10.1109/ICSECC.2019.8907095.

[7] B. R. Prasetya, G. Suryo Gumilang, Y. Saputra, and J. W. Simatupang, "Light Detection for White Asparagus Farm Using Arduino," *J. Electr. Electron. Eng.*, vol. 2, no. 2, pp. 60–64, 2018.

[8] J. B. Awotunde, C. Chakraborty, and A. E. Adeniyi, "Intrusion Detection in Industrial Internet of Things Network-Based on Deep Learning Model with Rule-Based Feature Selection," *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021, doi: 10.1155/2021/7154587.

[9] V. Vincent, J. V. Harryanto, A. M. Lubis, and J. W. Simatupang, "Kotak Kendali Perangkat Elektronik Nirkabel untuk Aplikasi Smart Home," *J. Telekomun. dan Komput.*, vol. 10, no. 2, pp. 67–76, 2020, doi: 10.22441/incomtech.v10i2.8264.

[10] Y. P. A. Nugraha, "Parking Lot Management Information System Based on Arduino," Bachelor Thesis, President University, 2013.

[11] M. Chandran *et al.*, "An IoT Based Smart Parking System," *Journal of Physics: Conference Series*, 2019, vol. 1339, no. 1, doi: 10.1088/1742-6596/1339/1/012044.

[12] M. Yuvaraju and M. Monika, "IoT Based Vehicle Parking Place Detection Using Arduino," *Int. J. Eng. Sci. Res. Technol.*, vol. 6, no. 6, pp. 148–154, 2017, doi: 10.5281/zenodo.375816.

[13] A. Mishra, A. Karmakar, A. Ghatak, S. Ghosh, A. Ojha, and K. Patra, "Low Cost Parking System for Smart Cities: A Vehicle Occupancy Sensing and Resource Optimization Technique Using IoT and Cloud Pass," *Int. J. Sci. Technol. Res.*, vol. 8, no. 9, pp. 115–122, Sep. 2019.

[14] A. Mackey, P. Spachos, and K. N. Plataniotis, "Smart Parking System Based on Bluetooth Low Energy Beacons with Particle Filtering," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3371–3382, Sep. 2020, doi: 10.1109/JSYST.2020.2968883.