JCRINN Journal of Computing Research and Innovation
e-ISSN : 2600-8793

Available online at
https://jcrinn.com/
https://crinn.conferencehunter.com/

Journal of Computing Research and Innovation 10(2) 2025

Journal of Computing Research and Innovation

Cyberpark: An-IoT based Automated Parking Management and Monitoring System using RFID and ESP32

Nur Amalina Muhamad[1*], Norhalida Othman[2], Nor Diyana Md Sin[3], Noor Haliza Abdul Rahman[4], Muhammad Iskandar Mohd Rodzi[5]

[1,2,3,4,5]*Faculty of Electrical Engineering, Universiti Teknologi MARA Johor Branch, Pasir Gudang Campus, 81750 Masai, Johor, Malaysia.*

ARTICLE INFO

Article history:
Received 24 June 2025
Revised 5 August 2025
Accepted 6 August 2025
Online first
Published 1 September 2025

Keywords:
Smart Parking System
Internet of Things (IoT)
RFID
Real-Time Monitoring
ESP32 and Arduino Integration

DOI:
10.24191/jcrinn.v10i2.547

ABSTRACT

The growing number of vehicles in urban and institutional areas has created significant challenges in parking space management. Traditional parking systems, which rely on manual processes, are often inefficient, time-consuming, and lack real-time information, leading to increased traffic congestion, fuel consumption, and user dissatisfaction. To address these issues, this paper presents Cyberpark, an IoT-based Automated Parking Management and Monitoring System that integrates smart technologies to improve parking operations, enhance user convenience, and increase infrastructure safety. Cyberpark utilizes RFID technology for secure and automated vehicle access, infrared (IR) sensors to detect parking slot occupancy, servo motors to control entry and exit gates, and a water sensor to detect flooding within the parking area. The system incorporates two microcontrollers: Arduino UNO for local control and ESP32 for wireless communication with the Adafruit IO cloud platform. Real-time data from all sensors is uploaded to the cloud, allowing for remote monitoring and status updates via an online dashboard. The development process involved circuit simulation using Proteus 8 Professional and functional testing through the Wokwi simulator. Hardware components were assembled on a custom PCB, and the software was developed using the Arduino IDE. Results showed effective system performance in automating gate operations, accurately monitoring parking slot status, and issuing timely alerts during flood conditions. The real-time dashboard allowed for seamless user interaction and remote oversight. Cyberpark demonstrates the feasibility of integrating low-cost, open-source hardware with cloud-based IoT platforms to create a scalable, user-friendly, and environmentally responsive smart parking solution. This system is suitable for deployment in campuses, shopping complexes, or residential areas. The project contributes to the development of smart city infrastructure by offering a reliable, efficient, and safe approach to modern parking management.

1. INTRODUCTION

The increasing number of vehicles in urban areas has escalated the demand for efficient and intelligent parking solutions. Traditional parking systems, which depend on manual management or basic mechanical

[1*] Corresponding author. *E-mail address:* amalina0942@uitm.edu.my

infrastructure, often struggle with inefficiencies such as poor space utilization, long vehicle queues, and driver frustration. These inefficiencies result in increased fuel consumption and carbon emissions as drivers circulate in search of available parking spaces, contributing to environmental degradation and worsening urban traffic congestion (El Bilali et al., 2022; Suhartono et al., 2023).

Smart parking systems have emerged as a transformative solution, leveraging the capabilities of the Internet of Things (IoT), embedded sensors, microcontrollers, and wireless communication to address modern urban challenges. These systems aim to provide real-time monitoring of parking slot occupancy, automate gate control, enhance user experience, and improve operational efficiency. When integrated with cloud platforms, smart parking systems can also enable remote monitoring, data analytics, and predictive maintenance, aligning with broader smart city initiatives (Alymani et al., 2025).

The deployment of microcontrollers such as ESP32 and Arduino UNO in smart parking solutions has gained popularity due to their cost-effectiveness, programmability, and Wi-Fi connectivity. Coupled with RFID (Radio Frequency Identification) for access control, infrared (IR) or ultrasonic sensors for presence detection, and cloud services like Adafruit IO or Firebase for data visualization, these systems offer real-time visibility and automation (Ahmed et al., 2018; Gowda et al., 2021; Vantagudi et al., 2024). Furthermore, incorporating environmental sensors—like flood detectors—addresses safety concerns in flood-prone regions, offering added value to smart parking infrastructures (Abdulsaheb et al., 2020; Alsafar et al., 2025).

Numerous studies have demonstrated the potential of emerging technologies to enhance the efficiency and functionality of parking management systems. For example, Azmi and Ismail (2021) utilized ultrasonic sensors in conjunction with a Raspberry Pi to improve the speed and accuracy of parking slot detection, thereby reducing vehicle idle time. In a similar vein, Mani and Bhat (2020) developed an ESP8266-based system incorporating infrared sensors and LCD interfaces, which contributed to alleviating traffic congestion within parking facilities. Expanding on these foundational models, Gowda et al. (2021) implemented RFID-enabled slot allocation using the ESP32 microcontroller, streamlining the entry and exit processes through automated access control. Additionally, Elakya et al. (2019) proposed a hybrid framework that integrated ESP32, RFID, and IR sensors, thereby enhancing the responsiveness and adaptability of parking operations. To address environmental concerns, Abdulsaheb et al. (2020) incorporated flood detection capabilities into their Adafruit IO–based system, offering real-time alerts in adverse weather conditions and strengthening the system's safety features. Building upon these innovations, Alymani et al. (2025) introduced machine learning algorithms for occupancy prediction in IoT-based smart parking environments, thereby increasing the system's predictive accuracy and operational efficiency. Similarly, Alsafar et al. (2025) designed an advanced IoT-integrated solution focused on mitigating traffic congestion and reducing vehicular emissions, aligning with broader sustainability goals. Despite these advancements, existing smart parking systems continue to exhibit limitations, including the lack of comprehensive environmental integration, challenges in scalability, insufficient user feedback mechanisms, and reliance on centralized infrastructure. These gaps underscore the necessity for a more cohesive, adaptable, and user-oriented smart parking architecture.

Therefore, this paper introduces Cyberpark: Automated Parking Management and Monitoring System, a cost-effective, modular, and user-friendly solution that seamlessly integrates RFID-based access control, servo motor–driven gate operation, real-time IR occupancy detection, flood monitoring, and cloud-based IoT connectivity into a unified architecture. By synthesizing these features into a single scalable platform, Cyberpark contributes to the field of smart infrastructure, enhancing urban mobility, improving safety, and promoting sustainability.

2. METHODOLOGY

The development process for the **Cyberpark: Automated Parking Management and Monitoring System** was carried out in three main phases: simulation, hardware construction, and cloud integration. The

https://doi.org/10.24191/jcrinn.v10i2.547

initial stage emphasized simulation and design validation using tools such as **Proteus 8 Professional** and **Wokwi.** These simulations enabled the design and testing of circuit schematics, system logic, and component interactions within a virtual environment. Concurrently, program codes for both **Arduino UNO** and **ESP32** microcontrollers were developed and tested using the **Arduino IDE**, ensuring functional integrity before physical implementation. After completing the simulation phase, the project advanced to hardware prototyping. This phase involved assembling electronic components, wiring connections on a breadboard, and finally transferring the validated layout onto a custom-made **printed circuit board (PCB).** The core of the system includes **Arduino UNO** for hardware control (RFID, servo motors, water sensor, and buzzer) and **ESP32** for IoT communication. Key modules used are **MFRC522** RFID readers, **infrared sensors** for parking slot detection, **servo motors** for gate automation, **water level sensor, LED indicators**, and an **LCD display** for user feedback. The choice of Arduino UNO was due to its suitability for handling input/output control, while ESP32 was selected for its built-in Wi-Fi functionality and compatibility with real-time cloud systems. In the final stage, the ESP32 was configured to interface with the **Adafruit IO** cloud platform using the **MQTT protocol**. This connection allows real-time data transmission for remote monitoring of parking occupancy and flood alerts. The system sends continuous updates to the cloud, enabling users and operators to access live data via an intuitive dashboard, either on a web browser or mobile device.

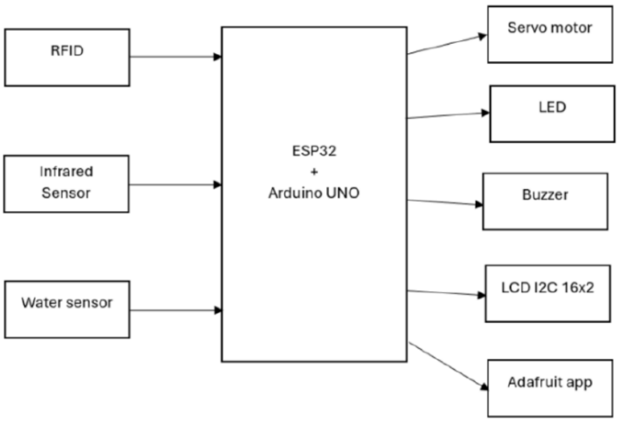

Fig. 1. Block diagram of Cyberpark: Automated parking management and monitoring system

Fig. 1. illustrates the block diagram of the Cyberpark: Automated Parking Management and Monitoring System, showcasing the interaction between input devices, processing units, and output components. The system's core is built on the integration of ESP32 and Arduino UNO, which serve as the main processing units to control operations and handle communication. The input section consists of three key sensors: the RFID module, infrared (IR) sensor, and water level sensor. The RFID module authenticates vehicles at the entry and exit points. The IR sensors detect vehicle presence in parking slots, allowing the system to determine whether a slot is occupied or vacant. The water sensor monitors flooding conditions within the parking area. Based on these inputs, the system activates various output components. The servo motors open or close the gates when valid RFID cards are scanned. LED indicators display the status of each parking space (red for occupied, green for available). The buzzer alerts users during flood detection, while the LCD 16x2 displays real-time instructions and status. Additionally, sensor data is sent via ESP32 to the Adafruit cloud dashboard, enabling remote monitoring and data logging. This integration enables a responsive, safe, and efficient smart parking environment.

Fig. 2. illustrates the operational flow of the Cyberpark: Automated Parking Management and Monitoring System, outlining the sequence of actions involved in managing vehicle entry, parking

https://doi.org/10.24191/jcrinn.v10i2.547

occupancy, environmental monitoring, and vehicle exit. The process begins when a car arrives at the entrance. The driver scans an RFID card, which is verified by the system. If the card is valid, the servo motor opens the gate, allowing access. If invalid, entry is denied. Upon successful entry, the parking slot is marked as occupied, and the red LED is activated to indicate the slot is taken. Simultaneously, the system continuously monitors the environment using a water level sensor. If the water level exceeds the threshold (e.g., level > 10), a buzzer is triggered to alert users to potential flooding. If no risk is detected, the system proceeds with regular monitoring. As the car parks, infrared sensors detect its presence and update the parking status. When a space is occupied, the red LED turns on; if vacant, the green LED is illuminated. The status of each slot is sent in real-time to the Adafruit IO cloud platform via ESP32, enabling remote monitoring through a dashboard. When the driver is ready to exit, the RFID card is scanned again at the exit gate. Upon successful verification, the servo motor opens the gate, and the slot is marked as available. The system updates the LED status and sends new data to the cloud. This flowchart summarizes the integration of hardware and software to deliver a smart, automated parking solution that improves efficiency, enhances safety, and provides real-time data to users and operators.

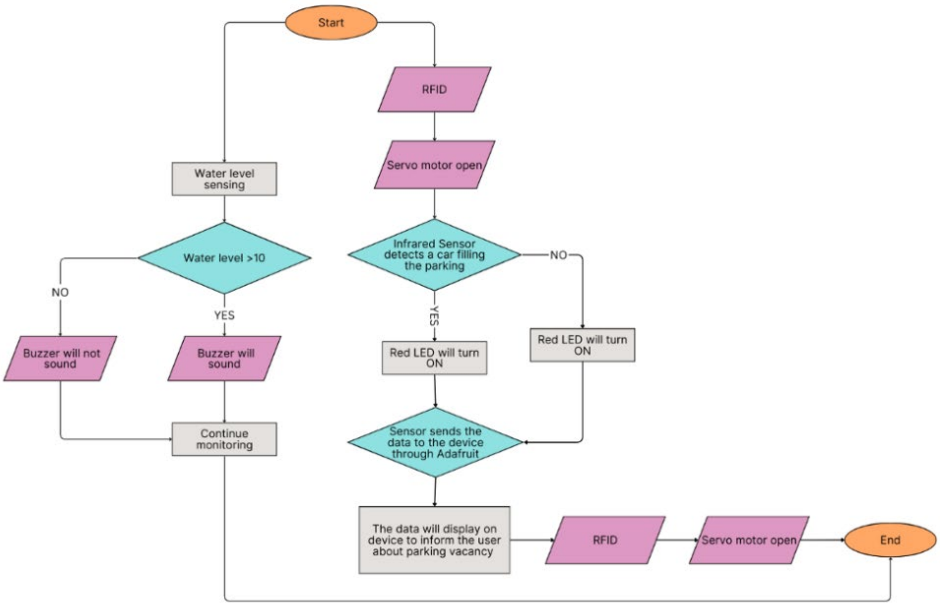

Fig. 2. Flow chart of the Cyberpark

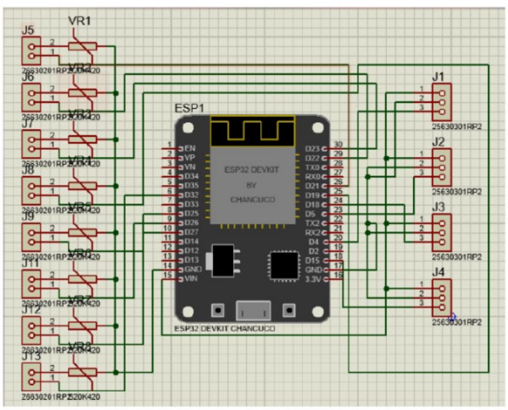

Fig. 3. The schematic diagram of the Cyberpark

https://doi.org/10.24191/jcrinn.v10i2.547

Fig. 3. displays the schematic wiring diagram of the Cyberpark system, built around the ESP32 DevKit microcontroller. This diagram illustrates the connections between the ESP32 and various input/output components through multiple digital and analog GPIO pins. The diagram shows how different devices such as infrared sensors, RFID readers, water sensors, LEDs, buzzers, and LCD displays are interfaced with the microcontroller for system functionality. On the left side, a series of components such as IR sensors and input devices are connected to digital pins D2 through D15. These pins handle signals for parking space detection, RFID card scanning, and water level monitoring. On the right side, the diagram shows output devices connected to pins D16 through D30, which include the LCD, servo motor, LED indicators, buzzer, and cloud communication interface.

This schematic serves as the electronic foundation of the Cyberpark system, ensuring all peripherals are appropriately powered and assigned to correct pins for accurate operation. Table 1 presents the detailed connection mapping between the ESP32 microcontroller and various components used in the Cyberpark system, including sensors, actuators, display units, and communication interfaces.

Table 1. Connection of component with ESP32 microcontroller

Component	Pin Description	ESP32 Pin Connection	Remarks
RFID Module (MFRC522)	SDA	D21	SPI/I2C Communication
	SCK	D18	Clock signal
	MOSI	D23	Master Out Slave In
	MISO	D19	Master In Slave Out
	RST	D22	Reset
Infrared Sensor 1	Signal	D2	Parking Slot Detection
Infrared Sensor 2	Signal	D4	Parking Slot Detection
Water Sensor	Analog Out	VP (A0)	Flood Detection
Servo Motor	PWM Control	D5	Gate Control
Red LED	Positive	D12	Slot Occupied Indicator
Green LED	Positive	D13	Slot Vacant Indicator
Buzzer	Positive	D14	Flood Alert Sound
LCD 16x2 (I2C)	SDA	D21	Shared with RFID (I2C Bus)
	SCL	D22	Shared with RFID (I2C Bus)
Adafruit IO (via WiFi)	-	Built-in WiFi (ESP32)	Cloud Communication

Fig.4. The completed prototype of the Cyberpark

Fig. 4 displays the completed prototype of the Cyberpark: Automated Parking Management and Monitoring System, which visually represents a scaled-down parking facility integrated with smart automation features. The setup includes miniature lanes, servo-controlled entry and exit gates, and parking

https://doi.org/10.24191/jcrinn.v10i2.547

bays equipped with infrared (IR) sensors and LED indicators. Two RFID modules are mounted at the gate areas to authenticate vehicles using RFID cards. Green and red LEDs positioned above each parking bay indicate available and occupied slots, respectively, based on real-time sensor input. A buzzer is included to alert users during flood simulation, while an LCD display at the front panel guides users with system messages such as "WELCOME" or "SCAN CARD." Internally, the ESP32 microcontroller manages cloud communication with the Adafruit IO platform, while the Arduino UNO handles gate control and hardware interfacing. The prototype demonstrates smooth coordination between access control, slot monitoring, and remote data updating, effectively simulating the real-world operation of a smart parking system.

3. RESULTS AND DISCUSSION

3.1 Simulation flow of the Cyberpark system

The simulation of the Cyberpark system was developed using the Wokwi platform to validate the functional flow of hardware components before physical implementation. Each stage of the simulation represents a core part of the parking management operation, including entrance access, vehicle detection, gate automation, and cloud monitoring via IoT. Fig. 5. illustrates the initial state of the entrance system, where users are prompted to input a code through a 4x4 keypad to simulate RFID card authentication. Upon activation, the LCD screen displays "Enter Password!" to prompt the user. The Arduino microcontroller processes the input and prepares to respond based on authentication results.

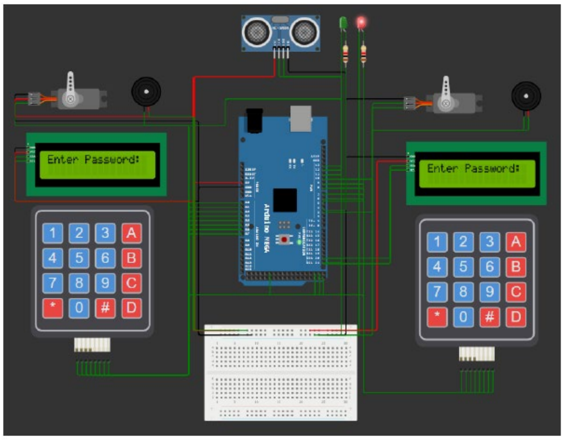

Fig. 5. Simulation diagram of the Cyberpark

Once the correct code is entered, as shown in Fig. 6. (left) the servo motor simulates gate operation by rotating to allow the vehicle to enter. Simultaneously, the LCD screen updates to display "Access Granted," confirming successful authorization. This mechanism effectively mimics the RFID-based entry control in a real parking system and ensures secure access control. Fig. 6. (right) focuses on the vehicle detection system using the HC-SR04 ultrasonic sensor, which simulates the presence of a vehicle within a parking slot. When a vehicle is detected within a specified range, the red LED turns on, indicating that the slot is occupied. If no vehicle is detected, the green LED lights up, showing the space is vacant. This sensor logic ensures accurate, automated status monitoring of each parking bay.

https://doi.org/10.24191/jcrinn.v10i2.547

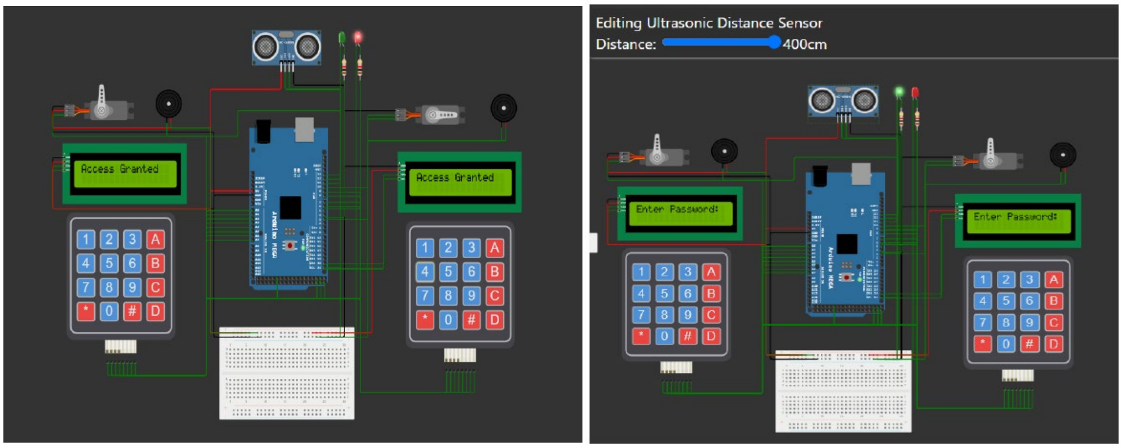

Fig. 6. (left) Simulation of entrance system and Fig. 6. (right) Simulation of sensor detection

The simulation for the IR sensor and LED functionality is represented using the HC-SR04 ultrasonic sensor in Wokwi. This setup detects the presence of a vehicle in the parking space. When a car is detected, the red LED activates, signalling the space is occupied. If no vehicle is detected, the green LED lights up.

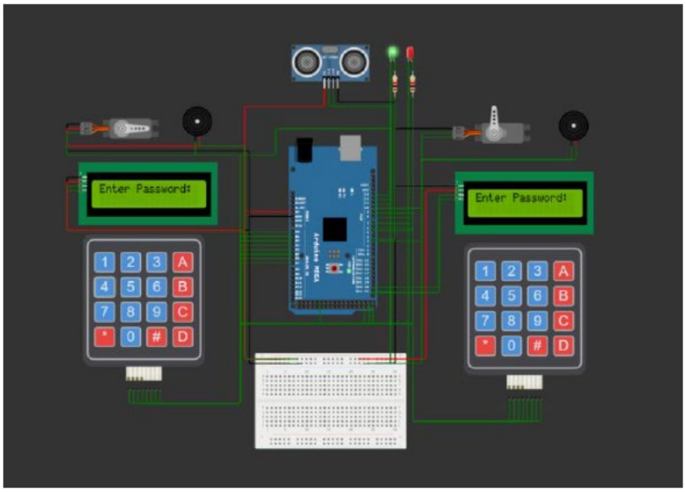

Fig. 7. Simulation of exit process

The exit process is demonstrated in Fig. 7, where a similar keypad is used to simulate RFID verification for vehicle departure. Upon entering the correct exit code, the servo motor reactivates to open the gate, allowing the vehicle to leave. A delay mechanism ensures the gate closes automatically after a short period, simulating safe and controlled exit behaviour.

https://doi.org/10.24191/jcrinn.v10i2.547

Fig. 8. IoT dashboard of Cyberpark

Finally, Fig. 8. presents the IoT dashboard interface via Adafruit IO, which displays real-time slot status. The dashboard shows circular indicators representing each parking bay, turning green for available and red for occupied. This cloud-based monitoring system allows users or administrators to remotely observe the availability of parking slots, ensuring timely decision-making and improved user experience. The dashboard connectivity highlights the effectiveness of IoT integration in modern smart parking systems. Together, this simulation sequence validates the full operational cycle of the Cyberpark system from secure entry, parking detection, and exit, to live cloud monitoring demonstrating the functionality and reliability of each integrated component.

3.2 Final hardware implementation and testing

Fig. 9. illustrates the vehicle entrance operation in the Cyberpark prototype. The system uses RFID technology for secure access control. As a vehicle approaches the gate, the driver presents an RFID card. The Arduino Uno reads and verifies the card's authenticity. Upon successful validation, a servo motor is triggered to lift the gate, allowing the vehicle to enter. To guide the user, an LCD screen mounted on the prototype displays status messages such as "WELCOME" or "ACCESS GRANTED," enhancing user interaction and ensuring a smooth entry process. This component reflects the real-world scenario of automated entry gates in smart parking facilities.

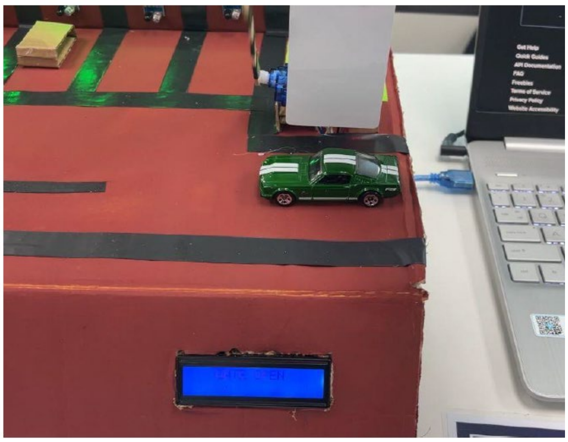

Fig. 9. Vehicle entrance

https://doi.org/10.24191/jcrinn.v10i2.547

Fig. 10. demonstrates how the system uses infrared (IR) sensors to detect the presence of a vehicle once parked inside a bay. Each sensor is connected to the ESP32 microcontroller, which processes occupancy data in real-time. If a vehicle is detected, the corresponding red LED turns on, indicating that the slot is taken. This information is then pushed to the Adafruit IO cloud platform, enabling remote monitoring of parking space availability. When a vehicle exits, the sensor detects the absence and switches the LED to green, signalling that the space is available again. This setup ensures accurate and efficient management of parking slots with minimal human intervention.

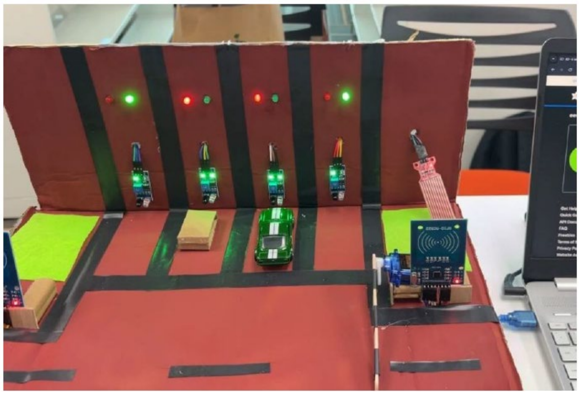

Fig. 10. Sensor detects the vehicle

Fig. 11. shows the water detection system in action. A water level sensor is submerged in a container to simulate flooding. When the sensor detects that the water level has exceeded a critical threshold (analog reading > 500), a buzzer is activated to alert users and system operators of the flood condition. This feature adds an important safety layer, ensuring that environmental hazards such as flooding are actively monitored. This helps prevent vehicle damage during heavy rain and supports the overall resilience of the parking infrastructure. Together, these hardware modules validate the complete flow of the Cyberpark system from access authorization and parking status monitoring to environmental safety demonstrating its readiness for real-world smart parking deployment.

Fig. 11. The water detection as a flood system alert

https://doi.org/10.24191/jcrinn.v10i2.547

4. CONCLUSION

This project successfully developed and demonstrated Cyberpark, an IoT-based automated parking management and monitoring system that addresses common challenges in urban and institutional parking facilities. By integrating RFID-based access control, infrared sensors for real-time occupancy detection, servo motor-driven gates, environmental monitoring via water sensors, and cloud connectivity using Adafruit IO, the system offers a comprehensive solution for smart parking operations. The project began with simulations using Proteus and Wokwi, validating the logic and component interactions before hardware implementation. The prototype was built using Arduino UNO and ESP32, combining local hardware control with wireless cloud communication. Results from the hardware testing showed that the system effectively automated vehicle entry and exit, accurately monitored parking slot availability, and reliably detected flood conditions. Real-time feedback was provided through LEDs, LCDs, buzzers, and a cloud dashboard, ensuring both user interaction and remote monitoring capabilities. The prototype proved to be cost-effective, scalable, and user-friendly, making it suitable for deployment in environments such as campuses, residential areas, or commercial buildings. In addition, the inclusion of environmental safety features such as flood alerts adds value beyond traditional parking systems. In conclusion, Cyberpark demonstrates the potential of combining embedded systems and IoT technologies to build intelligent, responsive infrastructure. Future work may include expanding the number of parking bays, integrating license plate recognition, or enabling mobile app access to enhance the overall system performance and user experience.

5. ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of Universiti Teknologi MARA (UiTM, Johor Branch, Pasir Gudang Campus and Faculty of Electrical Engineering for supporting this study.

6. CONFLICT OF INTEREST STATEMENT

The authors agree that this research was conducted in the absence of any self-benefits, commercial or financial conflicts and declare the absence of conflicting interests with the funders.

7. AUTHOR'S CONTRIBUTION

Nur Amalina Muhamad: Project coordination, manuscript writing, and overall supervision; **Norhalida Othman**: Data analysis, validation of system functionality, and manuscript editing; **Nor Diyana Md Sin**: Literature review, system testing, and result interpretation; **Noor Hasliza Abdul Rahman**: Technical documentation, graphical representation, and formatting; **Muhammad Iskandar Mohd Rodzi**: System design, hardware integration, and prototype development.

8. REFERENCES

Abdulsaheb, J. A., Hussien, A. A., & Al-Naji, L. A. (2020). IoT based smart parking system. *International Journal of Scientific & Engineering Research*, *11*(3), 50–55.

Ahmed, M. M., Chowdhury, S. A., & Reza, M. M. (2018). IoT-based smart parking system for urban cities. *International Journal of Computer Applications*, *180*(20), 7–10.

https://doi.org/10.24191/jcrinn.v10i2.547

Alymani, M., Alsharif, M. H., & Alfarraj, O. (2025). Enabling smart parking for smart cities using Internet of Things (IoT) and machine learning. *PeerJ Computer Science*, *11*, e2544. https://doi.org/10.7717/peerj-cs.2544

Alsafar, A., Bader, A., & Hamad, S. (2025). Advanced IoT-integrated parking systems with automated license plate recognition and flood detection. *Scientific Reports*, *15*(1), 1123. https://doi.org/10.1038/s41598-025-1123-x

Azmi, N. M. F. A. B., & Ismail, M. B. (2021). Smart parking system using IoT with ultrasonic sensor. *National Seminar on Embedded Systems*, 1–5.

El Bilali, H., Arrar, A., & Cherti, L. (2022). Intelligent parking systems and their economic and environmental impact: A review. *Energy Reports*, *8*, 240–251. https://doi.org/10.1016/j.egyr.2022.09.012

Elakya, R., Lakshmi, P., & Sharmila, D. (2019). Smart parking system using IoT. *International Research Journal of Engineering and Technology (IRJET)*, *6*(4), 275–279.

Gowda, Y., Ashwini, R., & Kiran, M. (2021). Smart parking system using IoT. *International Journal of Advanced Research in Computer & Communication Engineering*, *10*(6), 200–204.

Mani, R. M., & Bhat, V. V. (2020). Smart parking system. *International Journal of Engineering Research & Technology (IJERT)*, *9*(3), 122–125. https://doi.org/10.17577/IJERTV9IS090305

Suhartono, P., Nugraha, H. R., & Pratama, M. D. (2023). Development of smart parking systems using RFID sensors for online verification in different areas. *IoT and Applications (IOTA)*, *3*(2), 45–50.

Vantagudi, X., Sharma, R., & Raj, K. (2024). IoT-enabled smart car parking system through integrated sensors and mobile applications. *arXiv preprint*. https://arxiv.org/abs/2403.10123

https://doi.org/10.24191/jcrinn.v10i2.547