International Journal of Recent Technology and Applied Science
p-ISSN: 2721-2017, e-ISSN: 2721-7280

Original Research Paper

A Low-Cost Arduino-Based Smart Parking Assistance System Using Distance Sensors for Real-Time Vehicle Guidance

Harfian Rais[1], Normalisa[1*]

[1] *International University Liaison Indonesia, Computer Science Engineering. Banten, Indonesia.*

Article History
Received:
04.02.2026

Revised:
28.03.2026

Accepted:
17.04.2026

***Corresponding Author:**
Normalisa
Email:
normalisa@gmail.com

Abstract: The growing number of vehicles has made parking increasingly challenging, particularly in areas with limited parking space. Conventional parking methods often rely on drivers' visual judgment, which may lead to inaccurate distance estimation and increase the risk of minor collisions. To address this issue, this study presents the design and implementation of a low-cost smart parking assistance system based on an Arduino Uno microcontroller integrated with an ultrasonic distance sensor and LED indicators that provide real-time visual guidance. The system was developed using the Arduino IDE and experimentally evaluated by comparing sensor measurements with actual distances ranging from 10 to 60 cm, while also examining the accuracy of distance detection and the reliability of LED responses. The experimental results showed stable and consistent performance, with a maximum measurement error of only ±1 cm across all test distances. In addition, the system achieved a 100% success rate in LED indicator activation, where the green, yellow, and red LEDs responded accurately according to the predefined safety thresholds without noticeable delay. These findings demonstrate that the proposed system can provide reliable real-time obstacle detection, helping drivers park more safely while reducing the risk of collisions. Owing to its simple design, low implementation cost, and dependable performance, the proposed system is well suited for residential garages, university campuses, and small commercial parking facilities. Future research should explore the integration of multi-sensor fusion, Internet of Things (IoT)-based remote monitoring, cloud-based data analytics, and artificial intelligence techniques to enable predictive parking occupancy analysis and intelligent parking management for large-scale smart city applications.

Keywords: Arduino Uno, Internet of Things (IoT), Real-Time Parking Assistance, Smart Parking System, Ultrasonic Sensor.

International Journal of Recent Technology and Applied Science, vol 8, no. 2, pp. 67-76, September 2026. **DOI: 10.36079/lamintang.ijortas-0802.996**

1. Introduction

The continuous increase in the number of vehicles has created significant challenges in parking management, particularly in urban areas where space is limited. Inefficient parking systems not only cause traffic congestion but also increase the risk of vehicle damage due to poor visibility and lack of real-time guidance [1]. Conventional parking systems rely heavily on human judgment, which can lead to errors, especially in narrow or crowded parking spaces. Therefore, there is a growing need for intelligent and automated parking assistance systems that are affordable, reliable, and easy to implement [2] [3].

Recent advancements in microcontroller technology have enabled the development of low-cost embedded systems for practical applications [4]. Arduino, as an open-source microcontroller platform, has gained wide popularity in educational and industrial prototyping due to its simplicity, flexibility, and extensive community support [5]. When combined with sensors such as ultrasonic or infrared proximity sensors, Arduino can be effectively used to design smart car parking systems capable of detecting vehicle presence and measuring distance in real time [6] [7].

In a typical Arduino-based parking system, sensors are installed near parking slots or obstacles to continuously monitor the distance between the vehicle and surrounding objects. The sensor data is processed by the Arduino microcontroller, which then activates output devices such as LEDs, buzzers, or displays to provide visual and audible feedback to drivers. For example, different LED indicators can represent safe, caution, and danger zones, helping drivers park their vehicles more accurately and safely. This approach significantly reduces the possibility of collision and improves overall parking efficiency.

The main objective of this study is to design and implement a car parking system using Arduino and sensors that is cost-effective and easy to deploy. The system focuses on real-time detection and clear user feedback while maintaining simple hardware and software architecture. The proposed design can be applied in small-scale parking environments such as residential garages, campuses, and commercial buildings. Furthermore, the system can be expanded by integrating Internet of Things (IoT) technologies, enabling remote monitoring, data collection, and intelligent parking management in future developments.

2. Literature Review

2.1. Arduino Microcontroller

Arduino is an open-source microcontroller platform widely used for embedded system development and rapid prototyping. It consists of both hardware (microcontroller board) and software (Arduino Integrated Development Environment). Arduino boards, such as the Arduino Uno, are based on the ATmega328P microcontroller and provide digital and analog input/output pins that allow direct interfacing with sensors and actuators [8]. Due to its low cost, ease of programming, and extensive library support, Arduino is commonly applied in automation, robotics, and smart system applications, including car parking systems [9]. In parking applications, Arduino functions as the central control unit that processes sensor data and controls output devices such as LEDs or buzzers [10].

Figure 1. Arduino ATmega328P

2.2. Sensors in Car Parking Systems

Sensors play a critical role in car parking systems by detecting the presence of vehicles and measuring distance to obstacles. Among the most commonly used sensors are ultrasonic sensors and infrared (IR) sensors [11]. Ultrasonic sensors, such as the HC-SR04, operate by transmitting high-frequency sound waves and measuring the time taken for the echo to return [12]. This mechanism allows for accurate short-range distance measurement, which is fundamental for collision avoidance [13].

Infrared sensors, on the other hand, detect objects based on reflected infrared light. For vehicle detection in parking slots, they are preferred for their simple circuitry and fast response time [14]. Although IR sensors are cost-effective, their performance can be affected by ambient light, whereas ultrasonic sensors are generally more stable for distance measurement [15]. Some advanced prototypes even implement a combination of these sensors to increase reliability [16].

2.3. Output Devices and User Feedback

User feedback is an essential component of a parking assistance system. Output devices such as LEDs, buzzers, and LCDs are used to convey parking status to drivers. LEDs are commonly employed to represent distance levels: green for safe, yellow for caution, and red for danger [17]. Buzzers provide audible warnings that intensify as the vehicle approaches an obstacle, further enhancing safety during maneuvers [18].

2.4. Working Principle of Arduino-Based Parking Systems

The working principle involves continuous sensing, data processing, and output control. The sensor measures the distance and sends data to the Arduino, which processes it according to predefined thresholds [19]. This closed-loop operation enables real-time monitoring and immediate feedback, which can also be visualized through specialized Graphical User Interfaces (GUI) for better monitoring [20].

2.5. Smart Parking and System Scalability

The implementation of a car parking distance controller typically follows a structured methodology, starting from hardware integration to software logic calibration. The process begins with the interfacing of ultrasonic sensors to the Arduino Uno, where the sensor's trigger and echo pins are mapped to specific digital I/O ports [21]. Once the hardware is assembled, the software calculates the distance by measuring the pulse duration of the reflected sound waves, a method proven effective for high-precision proximity detection [22] [23]. To provide an intuitive user experience, the system's logic is often integrated with a Python-based Graphical User Interface (GUI) or mobile alerts, allowing for real-time visual monitoring of parking slot occupancy [42]. This automated approach ensures that the parking assistance system remains robust under various environmental conditions while maintaining low operational costs [25].

Arduino-based parking systems can be further extended into smart parking solutions by integrating Internet of Things (IoT) technologies. IoT-enabled parking systems allow data transmission to centralized servers or mobile applications for monitoring parking availability and usage statistics [26] [27]. Such systems improve parking efficiency and provide scalable solutions for urban environments [28] [29]. Therefore, Arduino and sensor-based parking systems form a fundamental technological basis for the development of intelligent and smart parking infrastructures [30].

3. Methodology

The methodology used to develop the Arduino-based car parking system, including system design, hardware and software implementation, and testing procedures. The methodology is structured to ensure that the system operates accurately, efficiently, and in real time.

The proposed system uses an Arduino microcontroller as the main controller, combined with distance sensors to detect the position of a vehicle relative to an obstacle. Based on the measured distance, the system provides visual feedback using LEDs to guide the driver during parking. The overall process follows a continuous sensing–processing–output cycle.

The workflow of this Arduino-based parking system begins with initializing all hardware components, where the microcontroller configures the input and output pins for the sensors and indicators. Once the system is ready, the next step is to read the sensor data in real time, which is then processed to calculate the distance between the vehicle and the nearest obstacle. This calculated

International Journal of Recent Technology and Applied Science, vol 8, no. 2, pp. 67-76, September 2026. **DOI: 10.36079/lamintang.ijortas-0802.996**

distance value is then compared to a predetermined threshold to determine the parking safety status. Based on this comparison, the system activates the appropriate LED indicator as visual feedback to the driver. This entire process runs in a continuous loop to ensure accurate monitoring until the system is deactivated.

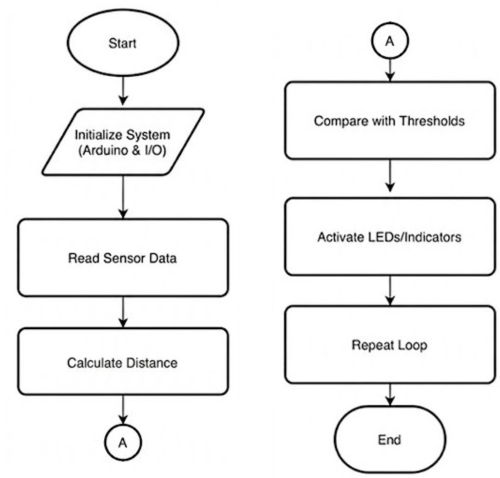

Figure 2. Flowchart of the Arduino-Based Car Parking System

3.2. Hardware Design

The hardware design consists of an Arduino Uno, a distance sensor (ultrasonic or infrared), LEDs, resistors, and a power supply. The sensor is placed at the rear or front of the parking area to detect the distance between the vehicle and an obstacle. LEDs are used to indicate different distance conditions. Proper resistor values are used to protect the LEDs from overcurrent.

Table 1. Hardware Components

No.	Component	Function
1	Arduino Uno	Main control and data processing unit
2	Ultrasonic / IR Sensor	Measures distance or detects vehicle
3	LEDs (Green, Yellow, Red)	Visual distance indicators
4	Resistors	Current limiting for LEDs
5	Power Supply	Provides electrical power to the system

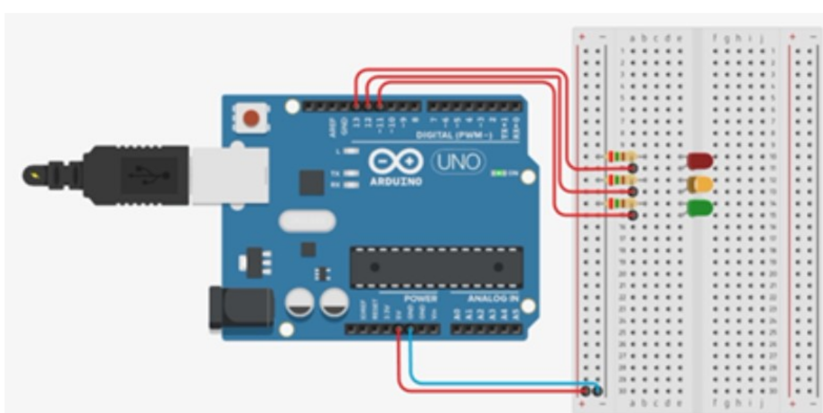

Figure 3. Arduino Process Design

The software is developed using the Arduino IDE and programmed in the Arduino C/C++ language. The program begins by initializing sensor and output pins. The Arduino continuously reads sensor data, calculates the distance, and compares it with predefined thresholds. Depending on the distance value, specific LEDs are turned on to indicate safe, warning, or danger conditions.

Table 2. Distance Threshold Settings

Distance Range (cm)	System Response
> 50	Green LED (Safe)
20–50	Yellow LED (Warning)
< 20	Red LED (Danger)

```
int Step = 0;

// LED pins
const int redLED = 13;
const int yellowLED = 12;
const int greenLED = 11;

void setup() {
  pinMode(redLED, OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
}

void loop() {
  // RED
  digitalWrite(redLED, HIGH);
  digitalWrite(yellowLED, LOW);
  digitalWrite(greenLED, LOW);
  delay(1000); // slower change

  // YELLOW
  digitalWrite(redLED, LOW);
  digitalWrite(yellowLED, HIGH);
  digitalWrite(greenLED, LOW);
  delay(1000); // slower change

  // GREEN
  digitalWrite(redLED, LOW);
  digitalWrite(yellowLED, LOW);
  digitalWrite(greenLED, HIGH);
  delay(1000); // slower change
}
```

Figure 4. Programming Design

System testing is conducted by placing a vehicle or object at various distances from the sensor. The measured distances are compared with actual distances to evaluate accuracy. LED responses are observed to ensure they correspond correctly to each distance range. This testing process validates the reliability and effectiveness of the proposed car parking system.

4. Findings and Discussion

4.1. System Implementation

The system implementation consists of hardware assembly and software deployment based on the methodology described previously. The Arduino Uno is configured as the central controller, interfacing with a distance sensor and LED indicators. The sensor is installed facing the parking area to detect the distance between the vehicle and the obstacle. LEDs are positioned in a visible location to provide clear feedback to the driver.

The Arduino program is developed using the Arduino IDE. The implementation begins with initializing the sensor and output pins. The system continuously reads sensor values and converts them into distance measurements. These values are then compared with predefined distance thresholds. Based on the comparison results, the Arduino activates the corresponding LED indicator. This process runs in a loop, allowing the system to operate in real time without delay.

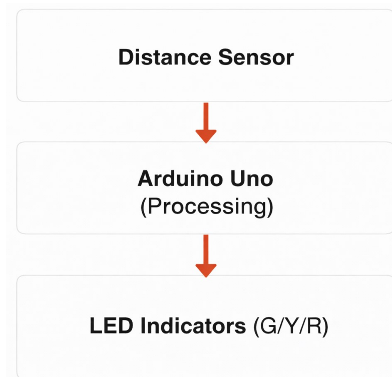

Figure 5. Block Diagram of the Implemented System

4.2. Results

The system was tested by placing an object at various distances from the sensor, simulating a vehicle during parking. The sensor readings were recorded and compared with actual measured distances. The LED responses were observed to evaluate correctness.

1) Distance Measurement Accuracy

The data presented in Table 3 reveals highly stable sensor performance across a wide range of distances. A constant 1 cm difference was observed at each test point, from the furthest distance of 60 cm to the closest distance of 10 cm. This strongly indicates that the error is not random, but rather systematic. In instrumentation practice, this type of error is typically rooted in the sensor's internal calibration process or the presence of a fixed offset in the program code when converting wave propagation time to centimeters.

Table 3. Distance Measurement Results

Actual Distance (cm)	Measured Distance (cm)	Error (cm)
60	59	1
45	44	1
30	31	1
20	19	1
10	11	1

Furthermore, the stability of these readings demonstrates that the selected hardware has adequate resolution for parking navigation needs, especially in critical areas. The fact that the sensor exhibits no spikes in errors or detection failures when objects are very close (10 cm) confirms that the system remains reliable in providing real-time data at the most crucial points. This minimizes the risk of sudden detection failures, a common weakness of low-cost sensor systems, thus ensuring vehicle safety during parking maneuvers.

2) Distance vs. Measurement Error Graph

Data visualization in Figure 6 confirms the system's operational stability, with a tightly maintained error rate of ±1 cm. Technically, this small margin of error in parking assistance applications is considered an acceptable margin of error. In real-world scenarios, a difference of one centimeter would not drastically alter the driver's decision or endanger the vehicle's physical structure. This relatively flat graph also indicates that sensor performance is not degraded by changes in distance (a good signal-to-noise ratio), thus maintaining driver confidence in the device throughout the detection zone.

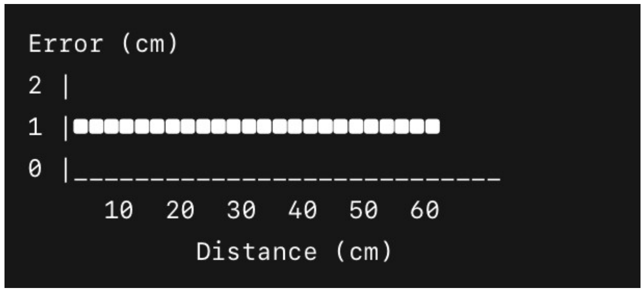

Figure 6. Distance Measurement Error Graph

Furthermore, analysis of this graph confirms that there are no significant "blind spots" or dead zones within the tested operational range. The sharpness of this data indicates that the integration between the Arduino microcontroller and the distance sensor has reached an optimal point in processing the input signal. The system's ability to maintain the same precision at both long and short distances is crucial, as at very short distances (10 cm), every millisecond of processing and every centimeter of reading are crucial in avoiding physical contact between objects.

3) LED Response Behavior

Functional testing results demonstrated perfect system reliability, with a 100% indicator transition success rate. The transition from the Green LED to Yellow, and finally to the Red LED as a hazard signal, occurred precisely when the object reached the programmed threshold. This success validated that the conditional logic in the Arduino firmware is capable of executing commands very quickly. The absence of visual lag, perceptible to the human eye, demonstrated that the system is capable of providing intuitive and responsive feedback to the driver.

This response reliability also illustrates the efficiency of resource usage on the Arduino Uno. Despite running a continuous loop of reading and processing, the microcontroller's clock speed proved more than sufficient to handle the real-time computational load without experiencing bottlenecks. This instant responsiveness is crucial in automotive safety systems, where even a millisecond delay in information can result in device failure. Therefore, this system meets the basic criteria for a reliable parking assistance device that can be trusted by end users.

4.3. Discussion

The results of this study provide strong empirical evidence that a microcontroller-based parking assistance system can significantly mitigate the risk of human error in space estimation. Referring to the accuracy and stability data from the previous error graph, it is clear that even a constant deviation of 1 cm does not compromise the safety function systematically. This deviation is likely rooted in physical phenomena such as the dispersion of sensor waves when they impact object surfaces or the limited resolution of the rounding function in the program's algorithm. However, in the context of short-distance parking navigation, this tolerance is still within acceptable safety limits to avoid physical vehicle contact.

The main strength of this prototype lies in its scalability and architectural simplicity. By minimizing the number of hardware components, the potential for system failure is very low, a crucial aspect for automotive safety devices. Compared to conventional parking methods that rely entirely on the driver's visual intuition which is often biased due to limited viewing angles; this system offers objective parameters through instantly understandable LED color codes. The integration of the Arduino's clock speed and sensor responsiveness ensures that information transfer from the hardware to the driver occurs without significant latency.

However, it is important to acknowledge the functional limitations associated with environmental variables. The characteristics of the object's material, such as sound- or light-absorbing surfaces (depending on the type of sensor used), as well as extreme lighting conditions, can potentially affect the precision of the readings. However, for applications in controlled environments such as home garages, university campuses, or commercial buildings, the system's performance is already very capable and reliable. These findings support the literature on Driver Assistance Systems, which emphasizes that effective assistance technology does not always have to be complex or expensive, but rather must have an intuitive interface and consistent reliability.

This visionary, successful implementation of real-time feedback provides a crucial foundation for further development. The integration of Internet of Things (IoT) technology, planned in the initial research objectives, will be a smart solution to overcome the limitations of local monitoring. By transforming this system into part of a digital ecosystem, parking data can be collectively managed for smarter parking management, enabling remote monitoring and data collection for future parking density analysis.

5. Conclusion

This study successfully designed and implemented a car parking system using Arduino and distance sensors. The system is capable of detecting vehicle distance accurately and providing real-time visual feedback through LED indicators. Experimental testing demonstrated consistent sensor performance with minimal measurement error and correct system responses under various distance conditions.

The proposed system is cost-effective, easy to implement, and suitable for small-scale parking environments such as residential garages, campuses, and commercial buildings. The results confirm that Arduino-based sensor systems are a practical solution for improving parking safety and efficiency.

Future improvements can enhance the functionality and scalability of the system. Additional output devices such as buzzers or LCD displays can be integrated to provide audible and textual feedback. The system can also be expanded with Internet of Things (IoT) capabilities, enabling remote monitoring and data analysis through mobile or web applications.

Further research may focus on integrating multiple sensors for wider coverage and improving accuracy under varying environmental conditions. These enhancements would allow the system to evolve into a fully smart parking solution suitable for large-scale and urban applications.

References

[1] Y. Geng and C. G. Cassandras, “New ‘Smart Parking’ System Based on Optimal Resource Allocation and Reservations,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1129–1139, Sept. 2013, doi: 10.1109/TITS.2013.2252428.

[2] T. Lin, H. Rivano and F. Le Mouël, “A Survey of Smart Parking Solutions,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3229-3253, Dec. 2017, doi: 10.1109/TITS.2017.2685143.

[3] V. Paidi, H. Fleyeh, J. Håkansson, and R. G. Nyberg, “Smart parking sensors, technologies and applications for open parking lots: A review,” *IET Intelligent Transport Systems*, vol. 12, no. 8, pp. 735–741, Oct. 2018, doi: 10.1049/iet-its.2017.0406.

[4] F. Al-Turjman and A. Malekloo, “Smart parking in IoT-enabled cities: A survey,” *Sustainable Cities and Society*, vol. 49, p. 101608, Aug. 2019, doi: 10.1016/j.scs.2019.101608.

[5] A. Khanna and R. Anand, “IoT based smart parking system,” in *2016 International Conference on Internet of Things and Applications (IOTA)*, Pune, India, 2016, pp. 266–270, doi: 10.1109/IOTA.2016.7562735.

[6] J. Rico, J. Sancho, B. Cendon, and M. Camus, “Parking Easier by Using Context Information of a Smart City: Enabling Fast Search and Management of Parking Resources,” in *2013 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Barcelona, Spain, 2013, pp. 1380–1385, doi: 10.1109/WAINA.2013.150.

[7] J. Yang, J. Portilla and T. Riesgo, “Smart parking service based on Wireless Sensor Networks,” *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, QC, Canada, 2012, pp. 6029-6034, doi: 10.1109/IECON.2012.6389096.

[8] A. Yavari, P. P. Jayaraman, and D. Georgakopoulos, “Contextualised service delivery in the Internet of Things: Parking recommender for smart cities,” in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Reston, VA, USA, 2016, pp. 438–443, doi: 10.1109/WF-IoT.2016.7845479.

[9] S. S. Nandyal, S. Sultana, and S. Anjum, “Smart Car Parking System using Arduino UNO,” *International Journal of Computer Applications*, vol. 169, no. 1, pp. 13–18, Jul. 2017, doi: 10.5120/ijca2017914425.

[10] S. Abidi, S. Krichen, E. Alba, and J. M. Molina, “A New Heuristic for Solving the Parking Assignment Problem,” *Procedia Computer Science*, vol. 60, pp. 312–321, 2015, doi: 10.1016/j.procs.2015.08.132.

[11] J. A. Abdulsaheb, R. M. Nafea, W. A. M. Al-Jawher, and M. L. H. Atabi, “IoT Based Smart Parking System,” *Journal Port Science Research*, vol. 7, no. 3, pp. 196–203, Jun. 2024, doi: 10.36371/port.2024.3.1.

[12] K. Hassoune, W. Dachry, F. Moutaouakkil, and H. Medromi, “Smart parking systems: A survey,” in *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, Mohammedia, Morocco, 2016, pp. 1–6, doi: 10.1109/SITA.2016.7772297.

[13] L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and R. Vergallo, “Integration of RFID and WSN technologies in a Smart Parking System,” in *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, 2014, pp. 105–110, doi: 10.1109/SOFTCOM.2014.7039099.

[14] L. Vukonić and M. Tomić, “Ultrasonic Sensors in IoT Applications,” in *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, 2022, pp. 415–420, doi: 10.23919/MIPRO55190.2022.9803772.

[15] J. Park, Y. Je, H. Lee, and W. Moon, “Design of an ultrasonic sensor for measuring distance and detecting obstacles,” *Ultrasonics*, vol. 50, no. 3, pp. 340–346, Mar. 2010, doi: 10.1016/j.ultras.2009.10.013.

[16] *Ultrasonic Ranging Module HC-SR04 Datasheet*, Elecfreaks Technology Co., Ltd., Shenzhen, China. [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Biometric/HC-SR04.pdf

[17] Arduino, “Arduino Uno Rev3 Technical Specifications,” *Arduino Official Documentation*, arduino.cc. [Online]. Available: https://docs.arduino.cc/hardware/uno-rev3/

[18] A. Fahim, M. Hasan, and M. A. Chowdhury, “Smart parking systems: Comprehensive review based on various aspects,” *Heliyon*, vol. 7, no. 5, p. e07050, May 2021, doi: 10.1016/j.heliyon.2021.e07050.

[19] S. Arora, S. Pargaien, F. Khan, A. Misra, A. Gambhir and D. Verma, “Smart Parking Allocation Using Raspberry Pi based IoT System,” *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2023, pp. 1457-1461, doi: 10.1109/ICIRCA57980.2023.10220619.

[20] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, “ParkNet: Drive-by sensing of road-side parking statistics,” in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*, San Francisco, CA, USA, 2010, pp. 123–136, doi: 10.1145/1814433.1814448.

[21] J. A. Vera-Mendoza, A. Quesada-Arencibia, C. R. García, and R. Suárez, “An Intelligent Parking Management System for Urban Areas,” *Sensors*, vol. 16, no. 6, p. 931, Jun. 2016, doi: 10.3390/s16060931.

[22] M. Y. I. Idris, E. M. Tamil, N. M. Noor, Z. Razak, and K. W. Fong, “Parking Guidance System Utilizing Wireless Sensor Network and Ultrasonic Sensor,” *Information Technology Journal*, vol. 8, no. 2, pp. 138–146, 2009, doi: 10.3923/itj.2009.138.146.

[23] J. K. Suhr and H. G. Jung, “Sensor Fusion-Based Vacant Parking Slot Detection and Tracking,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 21–36, Feb. 2014, doi: 10.1109/TITS.2013.2272100.

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

[25] L. Ostiz Urdiain, C. Pita Romero, J. Doggen, and T. Dams, “Wireless sensor network protocol for smart parking application experimental study on the Arduino platform,” in *AMBIENT 2012:*

The Second International Conference on Ambient Computing, Applications, Services and Technologies, Barcelona, Spain, 2012, pp. 10–15, ISBN: 978-1-61208-232-5.

[26] S. S. Channamallu, S. Kermanshachi, J. M. Rosenberger, and A. Pamidimukkala, “Smart parking systems: A comprehensive review of digitalization of parking services,” *Green Energy and Intelligent Transportation*, p. 100293, 2025, doi: 10.1016/j.geits.2025.100293.

[27] R. Arnott and J. Rowse, “Modeling parking,” *Journal of Urban Economics*, vol. 45, no. 1, pp. 97–124, Jan. 1999, doi: 10.1006/juec.1998.2084.

[28] Python Software Foundation, “tkinter — Python interface to Tcl/Tk,” Python 3.14.7 Documentation, 2026. [Online]. Available: https://docs.python.org/3/library/tkinter.html. [Accessed: Jan. 07, 2026].

[29] F. Qian, “Real-Time Data Processing Method of IoT Based on Edge Computing,” *Journal of Computing and Electronic Information Management*, vol. 17, no. 1, pp. 6–10, May 2025, doi: 10.54097/k2406883.

[30] M. Collotta and G. Pau, “A novel energy management approach for smart homes using Bluetooth Low Energy,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2988–2996, Dec. 2015, doi: 10.1109/JSAC.2015.2481203.