Open Science and Technology
Volume 5, Issue 1, 18-26.
E-ISSN :2776-1681
https://opscitech.com/journal
DOI: DOI: 10.33292/ost.vol5no1.2025.147

Design of Automatic Car Parking Sensor System Using Arduino Uno Program and HC-SR04 Ultrasonic Sensor

I Wayan Suriana[1]*, I Wayan Dikse Pancane[1], I Wayan Sugarayasa[1], I Nyoman Gede Adrama[1]

[1]Electrical Engineering Study Program, Faculty of Engineering and Informatics, Universitas Pendidikan Nasional, Indonesia

*Corresponding author's email: wayansuriana@undiknas.ac.id

Article Info

Article history:

Received: 25 February 2025
Revised: 07 June 2025
Accepted: 11 June 2025
Published: 14 June 2025

Keywords:

Arduino Uno,
Information,
Parking,
Sensor,
Ultrasonic.

Abstract

Aims and Methods: One of the challenges that drivers often face when parking a vehicle is limited parking slots. Many drivers face difficulty finding parking slots due to lack of clear information about the slot availability so that they are often forced to make a U-turn after entering a parking lot that turns out to be full. In addition, the limitations of automatic parking systems in conventional vehicles also add to the difficulty in improving parking safety and efficiency, especially in narrow places. This study aims to design and build an automatic car parking sensor system by integrating Arduino-Uno program and HC-SR04 ultrasonic sensor to help drivers by providing accurate and real-time information about the availability of parking slots. The method comprised several stages: hardware designing, consisting of an Arduino Uno microcontroller, HC-SR04 ultrasonic sensor, buzzer, and LED as a warning indicator, then software programming to organize the system logic. The developed system was tested in various scenarios to evaluate both the accuracy of parking slot presence detection and the response speed.
Result: The testing results show that the system is able to provide precise warnings according to the detected object in the slot, thus improving safety during the parking process.
Conclusion: The Arduino-based car parking sensor system with HC-SR04 ultrasonic sensor could serve as an effective tool in detecting the presence of vehicles in the parking lot.

To cite this article: Suriana, I. W., Pancane, I. W. D., Sugarayasa, I. W., Adrama, I. N. G. (2025). Design of automatic car parking sensor system using Arduino Uno program and HC-SR04 ultrasonic sensor. *Open Science and Technology, 5*(1), 18–26.

1. Introduction

According to data released by Bali Province's Central Bureau of Statistics, Denpasar City is listed as the area with the largest number of vehicles. The number of vehicles on the road has an impact on increasing visits to shopping centers, recreation areas, and entertainment locations. This often leads to full parking lots. In 2024, the number of vehicles in Bali Province reached 5,016,351 units, where in Denpasar City alone, the number reached 1,540,337 units.

The number of vehicles parked in an area can trigger long congestion, especially if the parking system does not run properly. Therefore, the implementation of a parking system equipped with sensors to detect parked vehicles is very important. This system can display information on the main gate screen regarding parking slot availability so that the drivers can find out before entering the area whether parking slots are still available. If there are no slots available, they can decide to leave and return at another time. In this way, an effective parking system can help reduce congestion and traffic jams due to insufficient parking slots.

A parking system is a mechanism that regulates the movement of vehicles as they enter and exit a parking lot. Despite this, the application of computer-based parking systems in various institutions and businesses, such as shopping centers, squares, hospitals, hotels, railway stations, and terminals, is still fairly limited.

Parking information systems mainly function to display the availability of parking slots, and have been commonly implemented in urban areas. These systems provide notification of parking slot availability, either through changes in traffic signs or navigation systems in modern vehicles. In more advanced navigation systems, drivers can immediately obtain information about vacant parking slots. By utilizing these information systems, the time wasted looking for a vacant parking slot can be significantly reduced (Siregar, 2021).

Parking is a condition in which a motorized vehicle is not driven for a certain period of time, depending on the situation and needs of the area that regulates it, either by the government or other parties, such as individuals or commercial organizations. Therefore, vehicles that are not properly organized in the parking lot can hinder the smooth flow of traffic. Through proper policy implementation, parking can serve as a traffic management tool (Auliani *et al.,* 2024).

A parking lot is a location provided for placing motorized vehicles within a certain period of time and supervised by a responsible party. Usually, parking facilities are available in various places, such as shops, supermarkets, hotels, minimarkets, tourist attractions, and others. In each location, there is a party in charge of guarding the vehicle in the parking lot. Currently, in addition to maintaining vehicle security, parking attendants also assist drivers in parking their vehicles.

Parking information system is a system commonly used in city centers that displays the availability of parking slots. The transportation community receives information through changes in signage or modern car navigation systems (Auliani *et al.,* 2024). A microcontroller is a small computer represented as an integrated circuit (IC) designed to perform a specific task. The microcontroller circuit basically consists of one or more CPUs, memory (RAM and ROM), and programmable inputs and outputs (Wisaksono *et al.,* 2022). One type of microcontroller widely used in this case is Arduino, in which one of the most popular models is Arduino Uno. It is also the first product belonging to Arduino. The Italian word "Uno" means "one" or "first". The features of Arduino are generally the same as those of other microcontroller types. Utilizing several programs, such as DC motors, relays, servos, modules, and sensors, Arduino Uno allows users to control electronic components more easily (Sari *et al.,* 2022).

Ultrasonic waves are often used to detect the presence of an object by estimating the distance between the sensor and the object (Priyulida *et al.,* 2024). Therefore, human's ear cannot hear ultrasonic waves that have frequencies above 20,000 Hz.

I2C LCD is an LCD module that is controlled serially using the I2C/IIC (Inter Integrated Circuit) or TWI (Two Wire Interface) protocol. Generally, LCD modules are operated in parallel for their data and control lines. However, the use of parallel lines requires multiple pins on the controller, such as Arduino, Android, or computer (Mardhalena & Nathasia, 2022).

A servo motor is an actuator that functions as a driving device equipped with a closed-loop feedback control system so that it can precisely set and ensure the position of the motor output shaft. AC servo motors are commonly used in various industrial machines, while DC servo motors are more suitable for small-scale applications. A control cable is used to navigate the servo motor through a technique known as pulse width modula (PWM). In order for the servo motor to remains at the specified position, this pulse signal must be repeated every 20 milliseconds. Thus, the position of the servo motor cannot remain stationary forever (Anugrah *et al.,* 2020).

A jumper cable is a type of electrical cable equipped with pins at each end, allowing two Arduino-related components to be connected without the need for a soldering process. Generally, jumper cables function as conductors to connect various electrical circuits into a single unit (Hilman *et al.,* 2021). Meanwhile, Arduino IDE is a software that makes it possible to sketch programming. In other words, the Arduino IDE makes it easy to program the intended Arduino board (Abrianto *et al.,* 2021). The process of editing, creating, uploading, and coding various programs can be conducted very easily through the Arduino IDE. Each sketch in the Arduino IDE is saved with the ".ino" file extension. Several basic structures need to be considered when writing programs using Arduino. The "Check" button functions to compile the sketch code and check for errors. If errors are detected in the code being written, user will usually receive a message indicating a problem. In other words, this verification feature checks whether the program created could be executed properly (Darmawan *et al.,* 2023).

Sometimes when looking for a parking slot, a driver encounters a situation where the parking lot is full. After entering the area, the driver will usually try to find a vacant slot to park his/her vehicle. However, since there is no information indicating that the parking lot is full, the driver is forced to make a U-turn and exit the area. Meanwhile, other vehicles continue to arrive in the hope that the parking slots are still available. The absence of information regarding parking slot availability often leads to frustration, as time is wasted searching for vacant parking slot, as well as congestion that occurs due to vehicles exiting after not finding vacant slots.

Therefore, based on this background, a study entitled "Design of Automatic Car Parking Sensor System Using Arduino Program and HC-SR04 Ultrasonic Sensor" is conducted. Ultrasonic sensors were chosen owing to their advantage of being unaffected by changes in light, both during the day and night, as well as by varying weather conditions. This is in contrast to infrared (IR) sensors, which are very sensitive to changes in light. Ultrasonic sensors focus on the presence of objects, such as vehicles in this case.

By implementing ultrasonic sensors in each parking slot, this system is able to detect whether a parking slot is vacant or occupied (Pulungan *et al.,* 2022). Each parking slot has two states, namely 'available' or 'unavailable'. Information regarding parking slot availability, including the number of vacant slots, will be displayed on an LCD mounted at the entrance of the parking lot. In this way, the drivers can see from a distance whether the slots are still available. If the information reveals that all slots are occupied, they can immediately look for other alternatives without having to enter the parking lot, thus reducing congestion at the entrance and exit location.

2. Materials and Methods

In designing this car parking sensor system, it was necessary to make a modeling of the entire process of manufacturing and testing the hardware and programs. The design process employed the Waterfall method. This method is one of the oldest software development methods due to its structured nature. The process in this method is carried out sequentially, starting from system planning, analysis, design, to implementation.

Thus, this study's method comprised a systematic approach, starting from the system requirements identification stage, proceeding to the analysis, design, coding, testing/verification, and maintenance stages. Each stage was conducted sequentially, thus resembling a waterfall's flow. The Waterfall method stages in this study are shown in Figure 1.

Figure 1. Research design based on the Waterfall method

The Waterfall method in this study comprised several stages as follows:

a. *Requirements Analysis* aims to identify and document users' needs and determine the type of system to be built.
b. *Design* aims to create the system's technical design that is in accordance with the users' needs that had been documented.
c. *Development* aims to build the system based on the established design.
d. *Testing* is implemented to ensure that the system functions as needed and is free of errors (bugs).

e. *Maintenance* aims to ensure that the system continues to function properly after use.

In designing this system, a model was made as an initial description of the system workflow. Figure 2 presents the main components of the designed car parking sensor system using Arduino Uno program and HC-SR04 ultrasonic sensor. Meanwhile, Figure 3 displays the circuit image of this system.

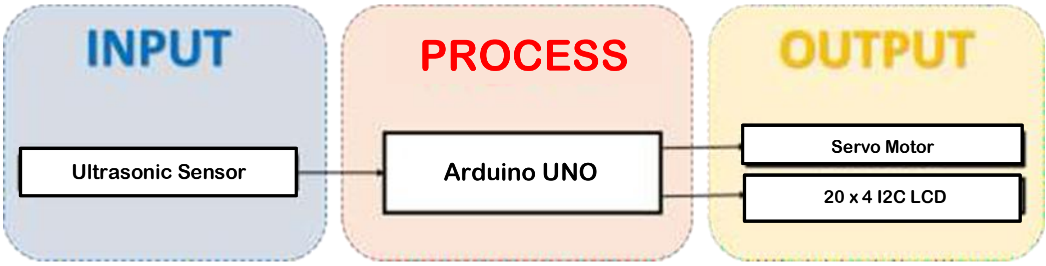

Figure 2. Main components of the designed car parking sensor system

Figure 2 shows a block diagram of a microcontroller (Arduino Uno)-based system, consisting of three main components: Input, Process, and Output. Each component is described as follows:

a. INPUT
- Ultrasonic Sensor
  This sensor, used to measure the distance between the sensor and the object in front of it, works by sending ultrasonic waves and measuring the reflection time of the waves back to the sensor. This distance data will be sent to the Arduino as an input.

b. PROCESS
- Arduino UNO
  Arduino serves as the brain of this system. This microprocessor will
  1. Receive data from the ultrasonic sensor.
  2. Process the data (e.g. calculate the distance).
  3. Give commands to output devices based on the processing results (e.g., if the distance is smaller than the threshold, activate the servo motor).

c. OUTPUT
- Servo Motor
  This servo may be used to open or close the automatic doors, drive the robot's arm, or other mechanical components depending on the program logic.
- 20 x 4 I2C LCD
  This screen is used to display information to the user, such as sensor readings (distance), system status, or instructions.

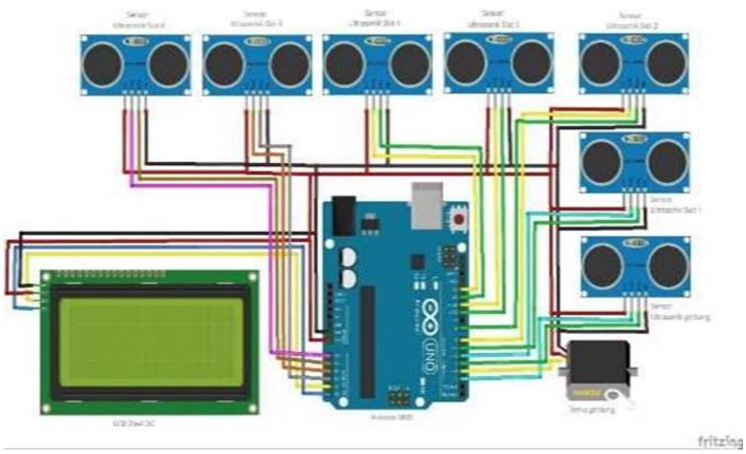

Figure 3. Circuit image of the designed car parking sensor system

This system functions by detecting incoming vehicles through ultrasonic sensors mounted on the gate. If the parking slots are still available, the servo will open the gate to give access to the vehicle. However, if all parking slots are occupied, the servo will not open the gate and the buzzer will sound as a warning sign. In addition, the LCD will display information that the parking lot is full so that the drivers can decide to turn around and return at a later time.

Figure 4 depicts the working process of the car parking sensor system. It starts with the initialization of Arduino, followed by reading data from the sensor, then executing the program logic to open and close the gate, and finally displaying information on the LCD to inform the user whether the parking slots are "occupied" or "vacant".

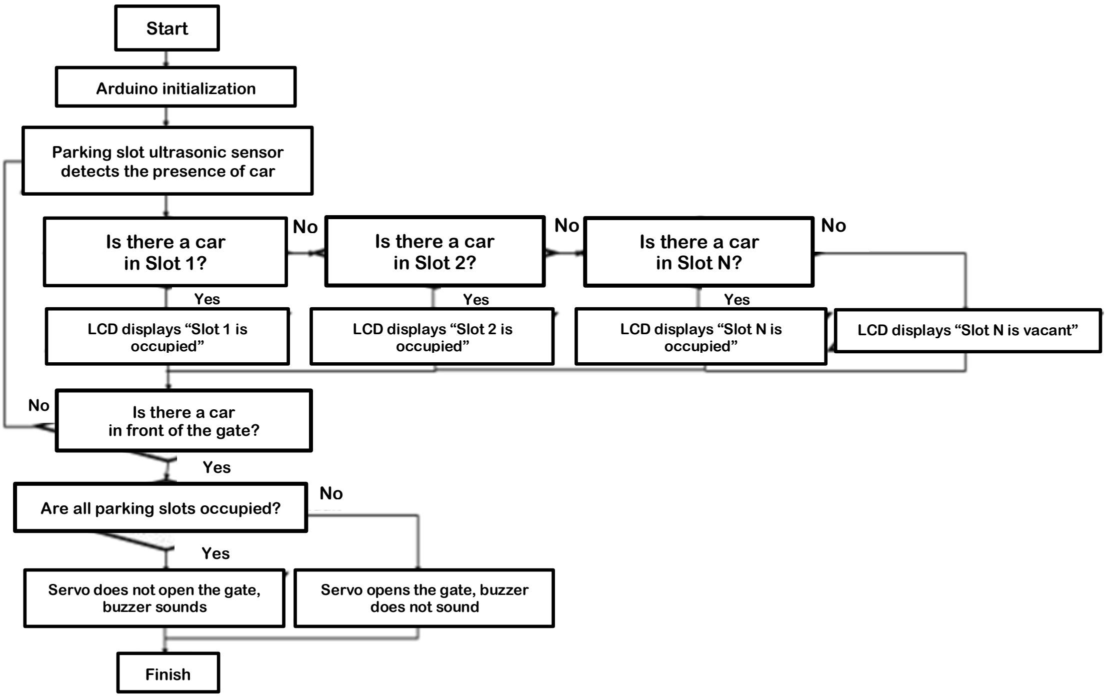

Figure 4. Working process of the designed car parking sensor system

3. Results and discussion

This chapter describes the results of the designed car parking sensor system, with an explanation regarding the Arduino program run through the Arduino IDE. In addition, various tests and measurements on each component were also conducted to ensure optimal system performance.

3.1 System implementation

Implementation is the process of applying features and tool designs to build a prototype system in the form of a miniature parking lot. The prototype consists of four parking slots, each equipped with ultrasonic sensors to detect the presence of four vehicles. On the left, there are ultrasonic sensors and servo motors that function as an automation system at the entrance gate. Meanwhile, in the center, there is a 20 x 4 I2C LCD that is used to display information regarding the availability of parking slots. The prototype of the car parking sensor system using ultrasonic sensor is shown in Figure 5.

Figure 5. Prototype of system implementation

It is clarly seen in Figure 5 that each parking slot is equipped with an ultrasonic sensor mounted on the back. If a vehicle is parked in front of the sensor, the system will detect its presence and automatically change the information on the LCD from "vacant" to "occupied".

Meanwhile, at the entrance gate, there is an ultrasonic sensor that functions to detect vehicles in front of the gate. If the parking slots are still available, the servo motor will open the gate to allow vehicles to enter. Meanwhile, if all parking slots are occupied, a buzzer will sound to indicate that parking slot is not available, and the servo motor will not open the gate. Information regarding the availability of parking slots can be directly monitored through the LCD display.

3.2 Program implementation

Program implementation was carried out through several stages. First, the Arduino code used #include to call the required library. Furthermore, #define was used to give a name to each Arduino pin as needed. It is recommended to name the pin based on the component to make it easier to remember.

The libraries used in this program included LCD and servo. Meanwhile, pin naming included "TRIG" and "ECHO" for the ultrasonic sensor, as well as the pin for the buzzer. The implementation appearances of the Arduino program and ultrasonic sensor are shown in Figure 6 and 7.

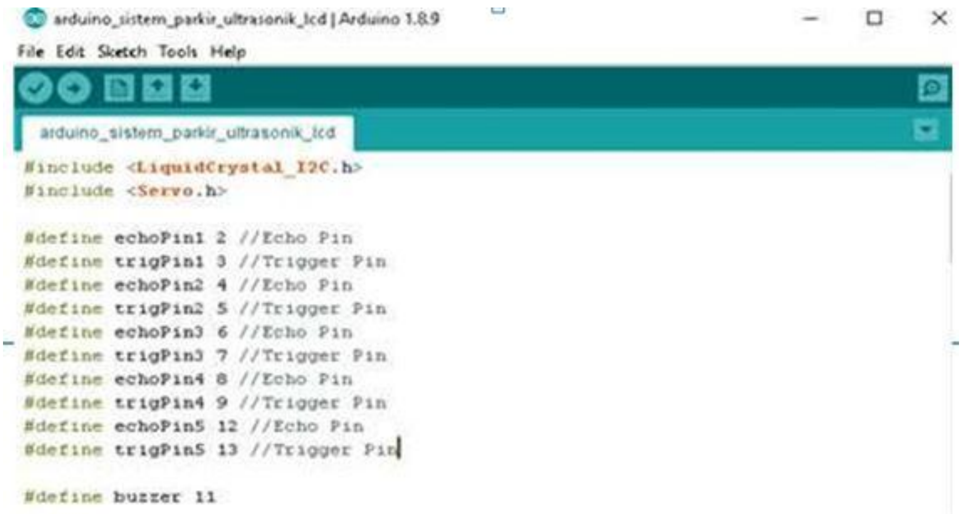

Figure 6. Arduino program implementation

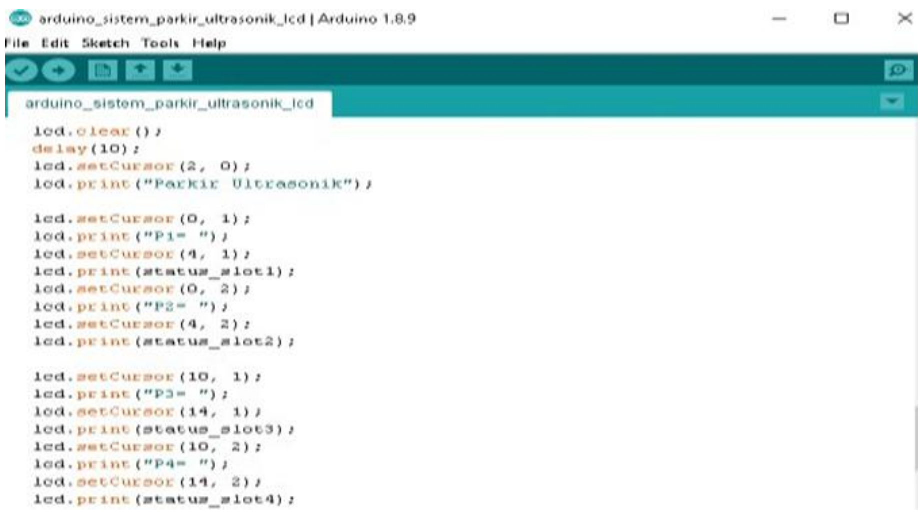

Figure 7. Ultrasonic sensor implementation

3.3 Measurement and testing results

This measurement was carried out on ultrasonic sensors in slots 1, 2, 3, and 4 of the parking lot. This sensor has several pins, namely VCC, GND, TRIG, and ECHO. Testing was conducted by measuring the voltage on the VCC and GND pins. In order to function properly, the ultrasonic sensor in slot 1 must have a voltage of around 5V, with a tolerance of ± 1V. The measurement results of slot 1 show a voltage of 4.25V, which means the sensor operates normally. The measurement result details are shown in Figures 8 to 11, while the complete measurement results are shown in Table 1.

Figure 8. Ultrasonic voltage measurement of slot 1

Figure 9. Ultrasonic voltage measurement of slot 2

Figure 10. Ultrasonic voltage measurement of slot 3

Figure 11. Ultrasonic voltage measurement of slot 4

Table 1. Ultrasonic sensor voltage measurement results of each slot.

Slot	Voltage (Volt)
1	4.25
2	4.23
3	4.25
4	4.25

The testing results of the Arduino Uno program and ultrasonic sensors in each parking slot are shown in Tables 2 to 8.

Table 2. Arduino program testing results.

Data	Result	Status
Electric voltage is applied to VIN and GND	LED indicator light is on	Success
Pin is connected to laptop via communication cable	Port is detected	Success
Arduino UNO reset button is pressed	Arduino restarts	Success

Table 3. Ultrasonic sensor testing results of slot 1.

Object Positioning	Serial Monitor	Status
3 cm from the sensor	3 cm	Success
4 cm from the sensor	4 cm	Success
5 cm from the sensor	5 cm	Success
6 cm from the sensor	6 cm	Success
7 cm from the sensor	7 cm	Success

Table 4. Ultrasonic sensor testing results of slot 2.

Object Positioning	Serial Monitor	Status
3 cm from the sensor	3 cm	Success
4 cm from the sensor	4 cm	Success
5 cm from the sensor	5 cm	Success
6 cm from the sensor	6 cm	Success
7 cm from the sensor	7 cm	Success

Table 5. Ultrasonic sensor testing results of slot 3.

Object Positioning	Serial Monitor	Status
3 cm from the sensor	3 cm	Success
4 cm from the sensor	4 cm	Success
5 cm from the sensor	5 cm	Success
6 cm from the sensor	6 cm	Success
7 cm from the sensor	7 cm	Success

Table 6. Ultrasonic sensor testing results of slot 4.

Object Positioning	Serial Monitor	Status
3 cm from the sensor	3 cm	Success
4 cm from the sensor	4 cm	Success
5 cm from the sensor	5 cm	Success
6 cm from the sensor	6 cm	Success
7 cm from the sensor	7 cm	Success

These testing results show a success rate of 100%, suggesting that all examinations on the hardware run smoothly without constraints. The designed car parking sensor system is able to respond and produce outputs in accordance with the commands and program logic that had been set. The total number of testing conducted are 37, all of which are declared successful. The calculation result of the absolute error is displayed as follows: measurement result - expected result = 37 - 37 = 0. Thus, the resulting deviation value or error in measurement and testing is 0, indicating that the designed system functions properly as planned.

5. Conclusion

Based on the testing results, it is concluded that this study has successfully designed and developed an automatic car parking sensor system that utilizes Arduino Uno program and HC-SR04 ultrasonic sensor.

Testing of each hardware was conducted 37 times, with a success rate of 100%, meaning it was successful 37 times. This parking sensor system in the form of prototype tool can simulate the condition of a parking lot that has been integrated with the system. Information about the parking slot availability can be monitored in real-time through the LCD display, making it very easy for vehicle drivers to obtain the status of parking slot availability. Based on this study's findings, it is recommended for every motorized vehicle to install a parking sensor tool that utilizes Android/iOS application or web dashboard, since it allows drivers to remotely obtain information about the parking slot availability. In addition, a combination of ultrasonic sensors and other sensors, such as infrared (IR) or RFID sensors, could also be utilized to identify parked vehicles in a parking lot.

6. References

Abrianto, H. H., Sari, K., & Irmayani, I. (2021). Remote navigation room temperature monitoring and control system using WEMOS D1 mini. *JNKTI: National Journal of Computing and Information Technology*, *4*(1), 38–49. https://doi.org/10.32672/jnkti.v4i1.2687

Anugrah, Z., Sutisna, S. P., & Sutoyo, E. (2020). Design of an Arduino-based automatic waste sorting tool for organic, inorganic, and metal. *ALMIKANIKA: Scientific Journal of Mechanical Engineering*, *5*(1), 1–7. https: doi.org//10.32832/almikanika.v5i1.8001

Auliani, Q., Aditya, R., Saputra, M. D., Aldi, M., Pahlevi, B. R., & Aribowo, D. (2024). Parking sensor simulation based on Arduino Uno microcontroller with HC-SR04 sensor using Wokwi Website. *JTMEI: Scientific Journal of Mechanical, Electrical, and Industrial Engineering*, *3*(4). https://doi.org/10.55606/jtmei.v3i4.4557

Darmawan, E., Sujana, N., & Faqih, M. (2023). Design of a door automation system using ultrasonic sensors based on Atmega328P microcontroller at MedikaCom Vocational High School, Bandung. *Journal of Technology and Information*, *5*(1), 32–43.

Hilman, A., Wijaya, D. P., & Saidi, B. (2022). MONTABU: Monitoring soil moisture in sugarcane plants based on the internet of things (IoT). *Asian Journal of Innovation and Enterpreneurship*, *6*(1). https://doi.org/10.20885/ajie.vol6.iss1.art1

Mardhalena, M. M., & Nathasia, N. D. (2022). Parking sensor system to detect safe vehicle distance using Arduino Uno Atmega328-Based ultrasonic sensor. *JIPI: Scientific Journal of Research and Learning Informatics,* *7*(4), 1391–1400. https://doi.org/10.29100/jipi.v7i4.3888

Priyulida, F., Putra, R. A., & Situmorang, H. (2024). Arduino Uno-based automatic parking crossbar. *Go Infotech: Scientific Journal of STMIK AUB*, *30*(1), 87–95. https://doi.org/10.36309/goi.v30i1.263

Pulungan, A. I., Sumarno, S., Gunawan, I., Tambunan, H. S., & Damanik, A. R. (2022). Parking system design and automatic parking slot availability using Arduino. *Journal of Computer and Information Science*, *2*(2), 127–136. https://doi.org/10.54082/jiki.33

Sari, R., Herlawati, H., Khasanah, F. N., & Atika, P. D. (2022). Prototype of automatic parking sensor in vehicle blind-spot area using microcontroller. *JOSH: Journal of Information System Research*, *3*(2), 76–84. https://doi.org/10.47065/josh.v3i2.1245

Siregar, S. (2021). *Vehicle parking system at Jember train station parking area*. Undergraduate Thesis. Jember Muhammadiyah University, pp. 1–31.

Wisaksono, A., Syahrorini, S., & Anshor, I. (2022). *Introduction to interfacing techniques*. UMSIDA Press.