INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 8.854 Volume 13, issue 04 (2026)

# MICROCONTROLLER-BASED AUTOMATIC PARKING MONITORING SYSTEM.

**G'ulomova Ko'hinur Murod qizi**

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

**Ko'chimova Oyshabonu O'tkirjon qizi**

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

**Usmonov Faxriddin Sharofiddin o'g'li**

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

**Muhammatjonov Muxammadqodir Rashidbek o'g'li**

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

**Annotation**

This article analyzes the problems of efficient management of parking systems arising from the increasing number of vehicles in modern cities. The shortcomings of traditional manually controlled systems are highlighted, in particular errors related to the human factor, increased time consumption, and decreased monitoring efficiency. To address this problem, the development of an automated parking monitoring system based on a microcontroller is proposed. The system is capable of detecting vehicle entry and exit, counting available parking spaces, and automatically controlling the entrance gate, operating in real time. In addition, within the scope of the project, an appropriate microcontroller platform, sensors, and actuators are selected, and the system block diagram and control program are developed. The system is tested in a simulation environment, and the obtained results are analyzed.

As a result, the proposed solution is justified as a means of reducing human labor in parking management, simplifying the monitoring process, and increasing system efficiency.

**Keywords**

Arduino Uno, LED, LCD, display, ESP32, PWM, Wi-Fi, sensors, GND, VCC, TRIG, ECHO.

## INTRODUCTION

Nowadays, the increasing number of vehicles in cities is making the efficient organization of parking systems an important issue. Due to the growing flow of cars near large shopping centers, higher education institutions, hospitals, hotels, airports, and various service facilities, the need to manage parking spaces, identify available spots, and control entry and exit processes is increasing. In traditional manually controlled systems, errors increase due to the human factor, time consumption rises, and monitoring quality decreases.

In solving this problem, automated solutions based on embedded systems and microcontrollers play an important role. Such systems enable real-time data collection, processing, rapid information delivery to users, and control of various actuators. Therefore, creating an automatic parking monitoring system based on a microcontroller can serve as a clear example of the practical application of modern embedded systems.

The main goal of this project is to develop a monitoring system based on a microcontroller that detects vehicles entering and exiting the parking area, counts the number of available parking spaces, and automatically controls the entrance gate. The system should display the current state of the parking area, reduce human labor, and simplify the monitoring process.
The main tasks of the project are as follows:
1. to study the operating principle of an automatic parking monitoring system;
2. to select an appropriate microcontroller platform;
3. to determine the sensors and actuators used in the system;
4. to develop the block diagram and electrical circuit of the system;
5. to create a control program for the microcontroller;
6. to test the system in a simulation environment;
7. to analyze the obtained results and identify ways for improvement.

The application scope of the project is wide, and it can be used in open and closed parking areas, business centers, shopping complexes, university campuses, and residential parking areas. This project demonstrates the practical integration of embedded systems, automation, and digital monitoring technologies.

**General concept of embedded systems**

Embedded systems are an integrated set of hardware and software designed to perform a specific task. Unlike general-purpose computers, they are usually specialized for a narrow range of functions. Embedded systems are widely used in household appliances, automotive engineering, industrial automation, medical devices, security systems, and smart infrastructure elements.
The central part of such systems is often a microcontroller, which receives data from the external environment through sensors, processes it programmatically, and controls the appropriate output devices based on the result. The advantages of embedded systems include compactness, energy efficiency, low cost, real-time operation capability, and high reliability.
An automatic parking monitoring system also belongs to the class of embedded systems, where the presence of vehicles is detected using sensors, calculations are performed using a microcontroller, and components such as a servo motor, display, and signaling elements are controlled.

**Relevance of the automatic parking monitoring system**

As urbanization accelerates and the number of vehicles increases, the shortage of parking spaces is becoming one of the most pressing problems in transport infrastructure. Many drivers spend extra time searching for available parking spaces. This leads to increased fuel consumption, heavier traffic congestion, and negative environmental impacts.
An automatic monitoring system helps to solve the following problems:
1. accurate counting of free and occupied spaces;
2. fast control of entry and exit;
3. automation of gate control;
4. reduction of dependence on operators;
5. provision of convenient information to users;
6. improvement of safety and control levels.

In addition, such systems can become part of the smart city concept in the future, enabling remote control, integration with mobile applications, and connection to cloud-based monitoring platforms.

**Microcontroller selection and its capabilities**

INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 8.854 Volume 13, issue 04 (2026)

In this project, the Arduino Uno platform was selected as the microcontroller. Arduino Uno is based on the ATmega328P microcontroller and is widely used in educational and practical projects. It has the following advantages:

1. simplicity of programming;
2. availability of many libraries;
3. easy integration with sensors and modules;
4. convenient uploading via USB;
5. wide support in simulation environments.

Arduino Uno has 14 digital input-output pins, some of which can operate in PWM mode. It also has 6 analog pins. The capabilities of Arduino Uno are sufficient for this project, as the system uses several sensors, one servo motor, LED indicators, a buzzer, and an LCD display.

If in the future it is necessary to connect the system to the internet, perform remote monitoring, or integrate with a mobile application, it would be appropriate to use microcontrollers such as ESP32 that have Wi-Fi and Bluetooth modules. A microcontroller-based automatic parking monitoring system is one of the modern automated control systems. The main function of such a system is to control the entry and exit of vehicles in the parking area, determine the number of free and occupied spaces, display the result to the user, and automatically control the entrance gate when necessary. The central part of this system is the microcontroller. A microcontroller is an electronic control device that integrates a processor, RAM, ROM, and input-output ports into a single integrated circuit. In simple terms, the microcontroller is the “brain” of the system; it receives signals from sensors, analyzes them according to a program, and ultimately produces a specific command. In the automatic parking monitoring system, the microcontroller manages all processes, detects vehicle arrival, counts available spaces, opens or closes the barrier, and displays information on the screen.

For this project, Arduino Uno is most commonly chosen as the microcontroller. Arduino Uno is based on the ATmega328P microcontroller and is widely used in educational and practical projects. There are several important reasons for its selection. First, working with Arduino Uno is very simple and convenient, and its programming environment is easy to understand. Second, it has a sufficient number of input-output pins to connect sensors, a servo motor, an LCD display, and other external devices. Third, this board is well supported in simulators such as Tinkercad and Wokwi, meaning the system can be tested without physically assembling it. Fourth, programming becomes easier due to the availability of ready-made libraries for Arduino. In addition, it is relatively inexpensive, consumes low power, and is very suitable for educational projects. Therefore, Arduino Uno is one of the most optimal solutions for projects such as an automatic parking monitoring system.

In such a system, the main sensing element is usually an ultrasonic sensor. One of the most common sensors is the HC-SR04. The function of this sensor is to determine the distance to the object in front of it. In a parking system, this sensor is used to detect whether a car is present or not. For example, a sensor installed at the entrance detects an approaching car, while a sensor at the exit determines whether a car is leaving the area. If multiple parking spaces need to be monitored, a separate sensor can be installed for each space. The general working principle of the sensor is that it sends an ultrasonic wave and measures the time it takes for the wave to reflect back from an object. Based on this time, the distance to the object is calculated. If the distance is less than a predefined value, it is concluded that a car is present in front of the sensor.

**Working principle of sensors and actuators**

In such a system, the main sensing element is usually an ultrasonic sensor. One of the most common sensors is the HC-SR04. The function of this sensor is to determine the distance to the object in front of it. In a parking system, this sensor is used to detect whether a car is present or

not. For example, a sensor installed at the entrance detects an approaching car, while a sensor at the exit determines whether a car is leaving the area. If multiple parking spaces need to be monitored, a separate sensor can be installed for each space. The general working principle of the sensor is that it sends an ultrasonic wave and measures the time it takes for the wave to reflect back from an object. Based on this time, the distance to the object is calculated. If the distance is less than a predefined value, it is concluded that a car is present in front of the sensor.

The physical working principle of the ultrasonic sensor is based on the propagation and reflection of waves. Sound is a mechanical wave that propagates through a medium. The ultrasonic sensor uses sound waves with frequencies above 20 kHz, which cannot be heard by the human ear. The HC-SR04 sensor typically emits a signal at a frequency of 40 kHz. Inside the sensor, there is a piezoelectric element that generates vibrations under the influence of an electrical impulse, producing ultrasonic waves. These waves propagate through the air, hit an object, and reflect back. The receiving part of the sensor captures the reflected wave and converts it back into an electrical signal. Thus, physical processes such as the piezoelectric effect, sound wave propagation, and reflection occur. The distance is determined based on the travel time of the signal. The formula used is: distance equals half of the product of sound speed and time. The division by two is because the signal travels to the object and returns back.

The signal generated by the sensor is processed by the microcontroller. For this, the microcontroller works with the TRIG and ECHO pins of the sensor. The program first sends a short electrical pulse to the TRIG pin. Then the sensor emits an ultrasonic wave. When the wave returns, a signal of certain duration appears on the ECHO pin. The microcontroller measures how long this signal lasts. This time value is analyzed by the program, and the distance is calculated using a specific formula. Then this distance is compared with a predefined threshold. For example, if the distance is less than 10 centimeters, the system considers that a car is present in front of the sensor. After that, the microcontroller makes the next decision according to the algorithm. If a car is detected at the entrance and there is a free space, the servo motor is activated to open the barrier. At the same time, the number of available spaces is decreased by one. Conversely, if a car is detected at the exit, the system opens the barrier and increases the number of available spaces by one. The results are displayed on the LCD screen. If the parking is full, a message such as “Parking Full” appears on the display.

The signal processing process of the microcontroller consists of several stages. First, the sensor receives a physical quantity (distance) from the environment. Then this physical quantity is converted into an electrical signal. After that, the microcontroller receives this signal in digital form and measures it over time. Based on the program algorithm, this data is analyzed, compared, and a decision is made. According to the decision, output devices are controlled. Output devices include a servo motor, LED indicators, a buzzer, and an LCD display. The servo motor is used to open and close the gate or barrier. The LCD display shows the system status, number of available spaces, or warning messages. LED indicators can represent states such as green and red. The buzzer is used to provide an audio signal when necessary.

The working algorithm of this system is usually a continuously repeating process. Once the system starts, the microcontroller sequentially reads data from all sensors. If the entrance sensor detects a car, the program first checks whether there are free spaces in the parking area. If there is a free space, entry is allowed, the barrier is opened using the servo motor, and after the car passes, it is closed again. At this moment, the number of available spaces decreases by one. If there are no free spaces, the system does not allow entry and displays a message indicating that there are no available spots. Similarly, if the exit sensor detects a departing car, the barrier opens, and after the car leaves, the number of available spaces increases by one. Through this algorithm, the number of cars in the parking area is automatically controlled.

The selection of the microcontroller for this project is based on technical and practical reasons. First, the project does not require complex calculations but requires fast and reliable control. Arduino Uno is sufficient for such tasks. Second, it has the necessary interfaces to receive data from sensors and control devices such as a servo and LCD. Third, from a software perspective, Arduino is programmed in a simple language based on C/C++, which simplifies development. Fourth, this microcontroller is one of the most widely used platforms in educational institutions, laboratory work, and course projects, making it easy to learn and explain. Therefore, the basis for this selection includes convenience, compatibility, reliability, affordability, and the availability of simulation capabilities.

In conclusion, a microcontroller-based automatic parking monitoring system detects the state of vehicles using sensors, processes this data through the microcontroller, and controls various devices accordingly. In this system, the microcontroller serves as the central control unit, while the ultrasonic sensor acts as the main sensing device. The physical operation of sensors is based on the propagation and reflection of sound waves, while signal processing relies on measuring the duration of electrical impulses and software analysis. The choice of a microcontroller such as Arduino Uno is explained by its simplicity, functionality, and suitability for practical projects. Therefore, such a system allows efficient implementation of automation, monitoring, and accounting processes.

Ultrasonic sensor

Ultrasonic sensors can be used to detect vehicles. Such a sensor sends sound waves and calculates the distance based on the reflected signal from an object. If the measured distance is less than a predefined threshold, the system considers that an object is approaching. In a parking project, an ultrasonic sensor is suitable for detecting vehicles at entry and exit points.

IR sensor

An infrared sensor can also be used to detect the presence of a vehicle. It operates based on the reflection or interruption of radiation. Although such sensors are inexpensive and simple, they may be sensitive to lighting conditions. IR sensors are also a convenient option for simulation.

Servo motor

A servo motor is used to automatically open and close the entrance gate. It can rotate to a specific angle. For example, the 0° position may correspond to a closed gate, while the 90° position corresponds to an open state. The microcontroller controls the servo motor using a PWM signal.

LCD display

The LCD 16x2 display is used to show the current system status to the user. It displays information such as the number of available spaces, system status, and messages like “Space available”, “Parking full”, “Entry allowed”, and “Exit completed”. This creates a convenient interface for operators and users.

LED indicator and buzzer

The LED indicator visually shows the system status. For example, a green LED indicates that space is available, while a red LED indicates that the parking is full. The buzzer provides an audio signal during entry or exit, indicating the system’s response.

**System block diagram**

The simplified block diagram of the system is presented below:

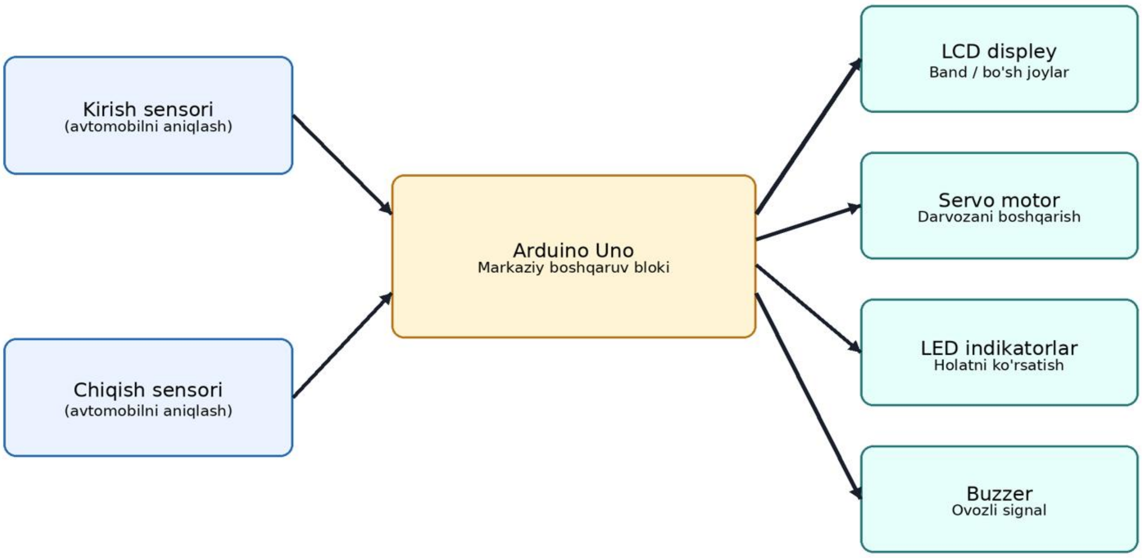



**Figure 1. System block diagram**

As can be seen from this block diagram, the central element of the system is the Arduino Uno microcontroller. It receives data from two sensors, processes it based on an internal algorithm, and sends control signals to the output devices.

**Signal flow and system architecture**

In the system, the signal flow is carried out in the following sequence:

1. The entrance sensor detects a vehicle approaching the parking area.
2. The microcontroller checks whether there is available space or not, taking into account the total capacity and the number of occupied spaces.
3. If there is available space, the gate is opened using a servo motor.
4. After the vehicle enters, the number of occupied spaces increases by one, and the number of available spaces decreases by one.
5. The LCD display and LED indicators are updated.
6. When the exit sensor detects a vehicle leaving, the number of occupied spaces decreases by one, and the number of available spaces increases.
7. The system returns to standby mode.

This architecture is based on local control, where all decisions are made within the microcontroller. Therefore, the system operates quickly, does not depend on the internet, and is suitable for simple projects.

**PRACTICAL PART**

**List of devices used in the project**

To implement this project, the following devices were selected:

1. Arduino Uno microcontroller board
2. LCD 16x2 I2C display
3. 4 HC-SR04 ultrasonic sensors
4. 2 servo motors
5. Breadboard
6. 2 LEDs
7. 2 electrolytic capacitors
8. 1 resistor

9. Jumper wires
10. USB cable / power connection

Each element performs a separate functional part of the system. Sensors are used to detect vehicles, the servo motor is used for mechanical control, the LCD display is used for information display, and LEDs and the buzzer are used for signaling.

**3.2. Description of the electrical circuit**

In the electrical circuit, all devices are connected to the microcontroller accordingly. The VCC pins of the sensors are connected to 5V, and the GND pins are connected to the common ground. The TRIG and ECHO pins of the ultrasonic sensors are connected to the digital pins of the Arduino. The signal pin of the servo motor is connected to one of the PWM-supported pins. The LCD display is connected via a parallel or I2C interface. For simplification, it is recommended to use an LCD with an I2C adapter in practice.

LED indicators are controlled through digital output pins. Each LED is connected in series with a resistor. The buzzer is also activated through a digital pin.

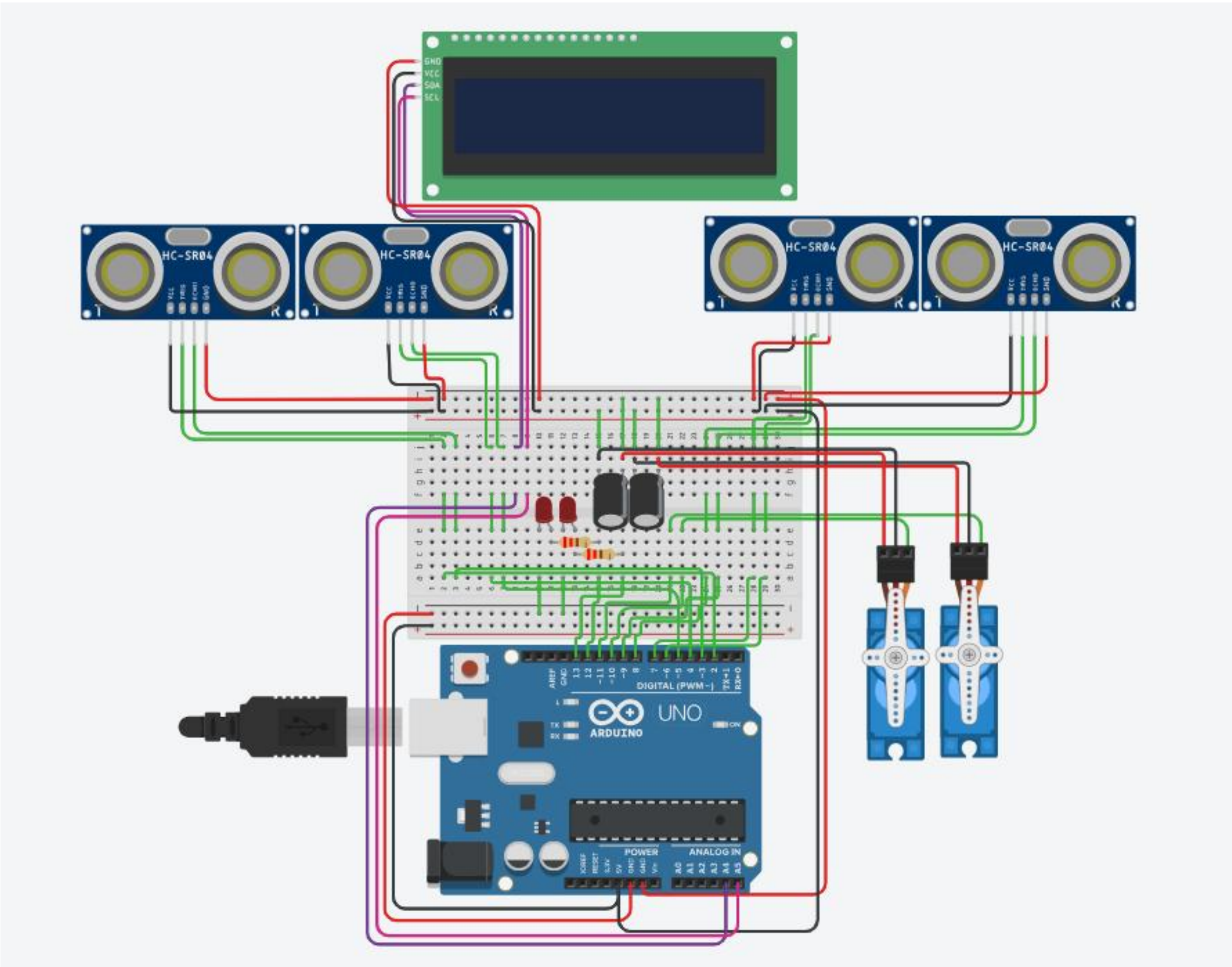



Figure 2. Description of the connection diagram

**Connection diagram:**

**1. LCD 16x2 I2C display**

1. GND → Arduino GND
2. VCC → Arduino 5V
3. SDA → Arduino A4

4. SCL → Arduino A5

**2. 1st ultrasonic sensor (left outer)**

1. VCC → 5V
2. GND → GND
3. TRIG → Arduino digital pin
4. ECHO → Arduino digital pin

**3. 2nd ultrasonic sensor (left inner)**

1. VCC → 5V
2. GND → GND
3. TRIG → Arduino digital pin
4. ECHO → Arduino digital pin

**4. 3rd ultrasonic sensor (right inner)**

1. VCC → 5V
2. GND → GND
3. TRIG → Arduino digital pin
4. ECHO → Arduino digital pin

**5. 4th ultrasonic sensor (right outer)**

1. VCC → 5V
2. GND → GND
3. TRIG → Arduino digital pin
4. ECHO → Arduino digital pin

**6. 1st servo motor**

1. VCC → 5V
2. GND → GND
3. SIG → Arduino digital pin

**7. 2nd servo motor**

1. VCC → 5V
2. GND → GND
3. SIG → Arduino digital pin

**8. 1st LED**

1. Anode (+) → Arduino digital pin through resistor
2. Cathode (-) → GND

**9. 2nd LED**

1. Anode (+) → Arduino digital pin through resistor
2. Cathode (-) → GND

**10. Capacitors**

1. Positive and negative terminals are connected to the power lines
2. Usually connected between 5V and GND

**11. Breadboard**

1. All 5V lines are connected to a common power rail
2. All GND lines are distributed to a common ground rail

### 3.3. Pin connection table

Pin connection table of the automatic parking monitoring system

**Table 3.3.1.**

| T/r | Qurilma nomi | Qurilma pini | Arduino Uno pini | Vazifasi |
|---|---|---|---|---|
| 1 | LCD 16x2 I2C displey | GND | GND | Umumiy yer |

| | | | | |
|---|---|---|---|---|
| | | | | ulanishi |
| 2 | LCD 16x2 I2C displey | VCC | 5V | Displeyni quvvatlantirish |
| 3 | LCD 16x2 I2C displey | SDA | A4 | Ma'lumot uzatish liniyasi |
| 4 | LCD 16x2 I2C displey | SCL | A5 | Sinxronizatsiya liniyasi |
| 5 | 1-ultrasonik sensor (chap tashqi) | VCC | 5V | Sensorni quvvatlantirish |
| 6 | 1-ultrasonik sensor (chap tashqi) | TRIG | D2 | Signal yuborish |
| 7 | 1-ultrasonik sensor (chap tashqi) | ECHO | D3 | Qaytgan signalni qabul qilish |
| 8 | 1-ultrasonik sensor (chap tashqi) | GND | GND | Umumiy yer ulanishi |
| 9 | 2-ultrasonik sensor (chap ichki) | VCC | 5V | Sensorni quvvatlantirish |
| 10 | 2-ultrasonik sensor (chap ichki) | TRIG | D4 | Signal yuborish |
| 11 | 2-ultrasonik sensor (chap ichki) | ECHO | D5 | Qaytgan signalni qabul qilish |
| 12 | 2-ultrasonik sensor (chap ichki) | GND | GND | Umumiy yer ulanishi |
| 13 | 3-ultrasonik sensor (o'ng ichki) | VCC | 5V | Sensorni quvvatlantirish |
| 14 | 3-ultrasonik sensor (o'ng ichki) | TRIG | D6 | Signal yuborish |
| 15 | 3-ultrasonik sensor (o'ng ichki) | ECHO | D7 | Qaytgan signalni qabul qilish |
| 16 | 3-ultrasonik sensor (o'ng ichki) | GND | GND | Umumiy yer ulanishi |
| 17 | 4-ultrasonik sensor (o'ng tashqi) | VCC | 5V | Sensorni quvvatlantirish |
| 18 | 4-ultrasonik sensor (o'ng tashqi) | TRIG | D8 | Signal yuborish |
| 19 | 4-ultrasonik sensor (o'ng tashqi) | ECHO | D9 | Qaytgan signalni qabul qilish |
| 20 | 4-ultrasonik sensor (o'ng tashqi) | GND | GND | Umumiy yer ulanishi |
| 21 | 1-servo motor | VCC | 5V | Servo motorni quvvatlantirish |
| 22 | 1-servo motor | SIG | D10 | Boshqaruv signali |
| 23 | 1-servo motor | GND | GND | Umumiy yer ulanishi |
| 24 | 2-servo motor | VCC | 5V | Servo motorni quvvatlantirish |
| 25 | 2-servo motor | SIG | D11 | Boshqaruv signali |
| 26 | 2-servo motor | GND | GND | Umumiy yer ulanishi |
| 27 | 1-LED | Anod (+) | D12 | Holat indikatori |
| 28 | 1-LED | Katod (-) | GND | Umumiy yer |

| | | | | |
|---|---|---|---|---|
| | | | | ulanishi |
| 29 | 2-LED | Anod (+) | D13 | Holat indikatori |
| 30 | 2-LED | Katod (-) | GND | Umumiy yer ulanishi |
| 31 | Kondensator 1 | + / - | 5V / GND | Quvvatni barqarorlashtirish |
| 32 | Kondensator 2 | + / - | 5V / GND | Quvvatni barqarorlashtirish |

**Note:** The table shows the connection pins of all main devices used in the automatic parking monitoring system to the Arduino Uno microcontroller.

**4.1. Algorithm description**

The operating algorithm of the automatic parking monitoring system consists of sequential operations controlled by a microcontroller, and its main task is to determine free and occupied spaces in the parking area, control entry and exit processes, and provide the user with information about the current status. When the system starts, all devices are initially set to their default state. The microcontroller establishes communication with sensors, servo motors, LED indicators, and the LCD display. At this stage, the total capacity of the parking area is predefined. For example, the system may assume that there are 10 parking spaces. Initially, the number of occupied spaces is set to 0, and the number of available spaces is set to 10. The LCD display shows that the system is ready and displays the number of available spaces.

After that, the program enters continuous monitoring mode. In this mode, data from sensors installed at the entrance, exit, and parking spaces are continuously read. The values received from the sensors are analyzed to determine whether a vehicle is present or not. If the entrance sensor detects a vehicle, the system first checks whether there is available space in the parking area. This stage is the most important part of the algorithm, as permission for entry depends on the number of available spaces. If at least one space is available, the system sends a command to open the entrance gate. The barrier or gate is opened using a servo motor, allowing the vehicle to enter.

After the vehicle passes through the entrance area, the system records this event and increases the number of occupied spaces by one. Accordingly, the number of available spaces decreases by one. The updated information is immediately displayed on the LCD screen. If LED indicators are used to show the parking status, they are also updated. For example, a green LED may indicate that space is available, while a red LED may indicate a fully occupied state. After the vehicle enters, the servo motor returns the gate to the closed position. Thus, the entry process is completed, and the system returns to monitoring mode.

If a vehicle is detected at the entrance while there are no available spaces in the parking area, the system does not allow entry. In this case, the gate remains closed, the red LED turns on, and a warning message such as “No space” or “Parking Full” is displayed on the LCD screen. This informs the user that the parking area is fully occupied. As a result, the system prevents unauthorized entry and avoids exceeding the parking capacity.

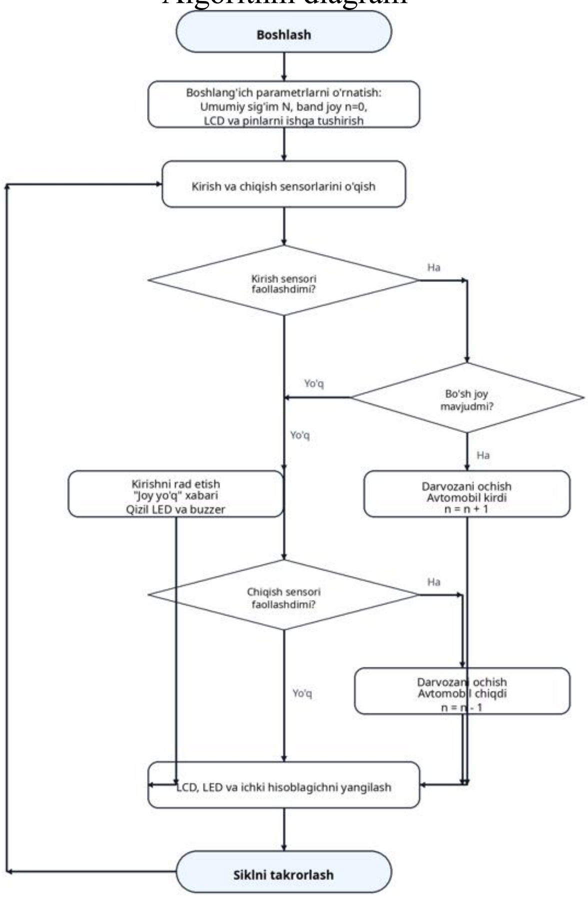
The algorithm for the exit process operates in a similar sequence. If a vehicle is detected by the exit sensor, the system opens the gate and controls the vehicle’s departure. After the vehicle leaves, this event is recorded, and the number of occupied spaces decreases by one. At the same time, the number of available spaces increases by one. The information on the LCD display is updated, and the user is shown the new status. If LED indicators are used, they also change according to the updated condition. After the exit process is completed, the gate is closed again, and the system returns to standby mode.

When separate sensors are installed for each parking space, the algorithm operates more accurately. In such cases, the system monitors whether each individual space is occupied or free. If a vehicle is present in a specific space, that space is marked as occupied, and the corresponding indicator is activated. If the space is free, the indicator is turned off or switched to another state. This method allows the system to determine available spaces not only based on entry and exit counts but also on actual occupancy. As a result, the system operates more reliably and accurately.

One of the important features of the algorithm is that it operates based on a continuous repetitive cycle. Due to the loop() function in the Arduino program, the system continuously reads data from sensors and updates the status. This ensures real-time operation of the system. Every new change is immediately recorded and transmitted to output devices. In this way, monitoring, calculation, control, and information display processes are carried out continuously as a whole.

The advantage of this algorithm is its simplicity, clarity, and practical convenience. It reduces human involvement, automates the detection of available spaces, speeds up entry and exit processes, and organizes parking management more efficiently. At the same time, this algorithm can serve as a foundation for future integration with RFID, cameras, mobile applications, or databases. Thus, the developed algorithm serves not only as an educational model but also as an important logical basis for future smart parking systems.

Algorithm diagram



INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 8.854 Volume 13, issue 04 (2026)

**Figure 3. Sequence of algorithm operation**

**Program code and its explanation**

Below is a sample Arduino code of the project. The code is simplified and performs sufficient functions for an educational project.

```
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
// ===== LCD DISPLEY =====
LiquidCrystal_I2C lcd(0x27, 16, 2);
// ===== SERVO MOTORLAR =====
Servo kirishServo;
Servo chiqishServo;
// ===== AVTOTURARGOH JOYLARI UCHUN SENSORLAR =====
const int joy1Trig = 2;
const int joy1Echo = 3;
const int joy2Trig = 4;
const int joy2Echo = 5;
// ===== JOYLAR HOLATINI KO’RSATUVCHI LEDLAR =====
const int ledJoy1 = 12;
const int ledJoy2 = 13;
// ===== KIRISH VA CHIQISH PALANGLARI UCHUN SENSORLAR =====
const int kirishTrig = 7;
const int kirishEcho = 6;
const int chiqishTrig = 8;
const int chiqishEcho = 9;
// ===== SERVO PINLARI =====
const int kirishServoPin = 10;
const int chiqishServoPin = 11;
// ===== O’ZGARUVCHILAR =====
int masofaJoy1, masofaJoy2;
int masofaKirish, masofaChiqish;
int boshJoylarSoni = 2;
// ===== ULTRATOVUSH SENSORIDAN MASOFA O’LCHASH FUNKSIYASI =====
int masofaniOlchash(int trigPin, int echoPin) {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long signalDavomiyligi = pulseIn(echoPin, HIGH);
  int masofa = signalDavomiyligi * 0.034 / 2;
  return masofa;}
void setup() {
  Serial.begin(9600);
  // LCD ishga tushirish
  lcd.init();
```

```
 lcd.backlight();
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Avtoturargoh");
 lcd.setCursor(0, 1);
 lcd.print("Tizimi ishga tayyor");
 delay(1500);
 lcd.clear();
 // Servo motorlarni ulash
 kirishServo.attach(kirishServoPin);
 chiqishServo.attach(chiqishServoPin);
 // Dastlab palanglar yopiq holatda turadi
 kirishServo.write(0);
 chiqishServo.write(0);
 // Slot sensorlari
 pinMode(joy1Trig, OUTPUT);
 pinMode(joy1Echo, INPUT);
 pinMode(joy2Trig, OUTPUT);
 pinMode(joy2Echo, INPUT);
 // Kirish-chiqish sensorlari
 pinMode(kirishTrig, OUTPUT);
 pinMode(kirishEcho, INPUT);
 pinMode(chiqishTrig, OUTPUT);
 pinMode(chiqishEcho, INPUT);
 // LEDlar
 pinMode(ledJoy1, OUTPUT);
 pinMode(ledJoy2, OUTPUT);}
void loop() {
 // ===== JOYLAR UCHUN SENSORLARDAN MASOFA O'QISH =====
 masofaJoy1 = masofaniOlchash(joy1Trig, joy1Echo);
 masofaJoy2 = masofaniOlchash(joy2Trig, joy2Echo);
 // ===== KIRISH VA CHIQISH PALANG SENSORLARIDAN MASOFA O'QISH =====
 masofaKirish = masofaniOlchash(kirishTrig, kirishEcho);
 masofaChiqish = masofaniOlchash(chiqishTrig, chiqishEcho);
 // ===== AVTOTURARGOH JOYLARI HOLATINI ANIQLASH =====
 bool joy1Band = masofaJoy1 < 200;
 bool joy2Band = masofaJoy2 < 200;
 // LEDlar orqali holatni ko'rsatish
 digitalWrite(ledJoy1, joy1Band ? HIGH : LOW);
 digitalWrite(ledJoy2, joy2Band ? HIGH : LOW);
 // Bo'sh joylar sonini hisoblash
 boshJoylarSoni = 2 - (joy1Band + joy2Band);
 // ===== KIRISH PALANGI BOSHQARUVI =====
 if (masofaKirish < 200) {
  kirishServo.write(90);   // palang ochiladi
 } else {
  kirishServo.write(0);    // palang yopiladi
 }
```

```
// ===== CHIQISH PALANGI BOSHQARUVI =====
if (masofaChiqish < 200) {
  chiqishServo.write(90);  // palang ochiladi
} else {
  chiqishServo.write(0);   // palang yopiladi
}
// ===== LCD DISPLEYGA MA'LUMOT CHIQARISH =====
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Bosh joylar soni:");
lcd.setCursor(0, 1);
lcd.print(boshJoylarSoni);
// ===== SERIAL MONITORGA MA'LUMOT CHIQARISH =====
Serial.print("1-joy masofa: ");
Serial.print(masofaJoy1);
Serial.print(" sm | 2-joy masofa: ");
Serial.print(masofaJoy2);
Serial.print(" sm | Bosh joy: ");
Serial.println(boshJoylarSoni);
delay(300);}
```

**Code explanation**

This program is developed to control an automatic parking monitoring system based on a microcontroller, and it operates on the Arduino Uno platform. The program uses ultrasonic sensors, servo motors, LED indicators, and an LCD display with an I2C interface. The main task of the system is to determine free and occupied parking spaces, automatically control entry and exit barriers, and provide the user with real-time information about the current state. At the beginning of the code, the Servo.h and LiquidCrystal_I2C.h libraries are included; the first is used to control servo motors, and the second is used to work with the LCD display. The line LiquidCrystal_I2C lcd(0x27, 16, 2); declares an LCD display with I2C address 0x27, 16 columns, and 2 rows. Additionally, two servo motor objects are created using the lines Servo kirishServo; and Servo chiqishServo;. One of them controls the entrance barrier, and the other controls the exit barrier.

In the program, two ultrasonic sensors are used for parking spaces, connected through the pins joy1Trig, joy1Echo, joy2Trig, and joy2Echo. These sensors are used to detect whether a vehicle is present in each parking space. In addition, LED indicators connected to the pins ledJoy1 and ledJoy2 show the status of each space. If a space is occupied, the corresponding LED turns on; if it is free, the LED turns off. Separate ultrasonic sensors are also used to control the entrance and exit barriers, connected to the pins kirishTrig, kirishEcho, chiqishTrig, and chiqishEcho. These sensors detect when a vehicle approaches the gate. The signal pins of the servo motors are assigned through kirishServoPin and chiqishServoPin. In the program, variables such as masofaJoy1, masofaJoy2, masofaKirish, and masofaChiqish are used to store distances obtained from the sensors, while boshJoylarSoni represents the number of available parking spaces. Initially, this value is set to 2, since the model includes two parking spaces.

One of the most important functions in the code is the masofaniOlchash() function. This function is used to measure distance using the ultrasonic sensor. During its operation, a very short pulse is first sent to the Trig pin. With this pulse, the sensor emits an ultrasonic signal into the environment. After the signal hits an object and returns, its duration is measured at the Echo pin using the pulseIn() function. Then the obtained time value is converted into centimeters based on the speed of sound formula. As a result, the approximate distance between the sensor and the object is determined. This method allows the parking system to detect whether a space is occupied and whether a vehicle is approaching the entry or exit gate.

The setup() function is executed once when the system starts and prepares all devices for initial operation. First, Serial.begin(9600); establishes serial communication with the computer, allowing data to be monitored through the Serial Monitor during testing. Then the LCD display is initialized, its backlight is turned on, and an initial message "Parking system is ready" is displayed. This informs the user that the system has started. After that, the servo motors are attached to Arduino pins and set to the closed position initially. The Trig pins of the sensors are configured as outputs, the Echo pins as inputs, and the LED indicators are set as outputs. In this way, all hardware components are prepared for normal operation.

The main control process of the program is carried out in the loop() function. This function repeats continuously while the Arduino is running. In each iteration, the program first reads distance values from the two sensors installed in the parking spaces. Then it reads distances from the sensors installed at the entrance and exit barriers. Based on the obtained results, it determines whether each space is occupied or free. In the code, this is implemented through the lines bool joy1Band = masofaJoy1 < 200; and bool joy2Band = masofaJoy2 < 200;. If the measured distance is less than the predefined threshold, it is assumed that an object (a car) is present in that space. Accordingly, the LED indicators are turned on or off. Then, the number of available spaces is calculated using the formula boshJoylarSoni = 2 - (joy1Band + joy2Band);. If both spaces are free, the result is 2; if one is occupied, it becomes 1; and if both are occupied, it becomes 0.

After that, the program proceeds to control the entrance and exit barriers. If the entrance sensor detects a vehicle within a predefined distance, the entrance servo rotates to 90 degrees, and the barrier opens. If no vehicle is detected, the servo returns to 0 degrees, and the barrier closes. The same logic applies to the exit barrier. If the exit sensor detects a vehicle nearby, the exit servo opens the barrier; otherwise, it remains closed. In this way, the system automatically manages vehicle entry and exit. This approach reduces operator involvement and ensures fast system operation.

The LCD display is actively used in the program to provide information to the user. In each iteration, the display is cleared, and the first row shows the message "Available spaces:", while the second row displays the current number of available spaces. As a result, the user or operator can monitor the parking status in real time. In addition, data is also sent to the Serial Monitor window. It displays the distance values for space 1 and space 2, as well as the number of available spaces. This is very convenient for testing and diagnostics, allowing analysis of the system's internal operation. The delay(300); line slightly slows down the loop cycle and stabilizes data acquisition from the sensors.

In general, this program performs the main functions of an automatic parking monitoring system. It monitors the status of parking spaces, calculates the number of available spaces, shows occupancy using LED indicators, controls entry and exit barriers using servo motors, and provides information to the user through the LCD display. As a result, the system operates in real time as a simple and practical automated monitoring solution. However, the threshold value of 200 cm used in the code may be relatively large in practice. For more accurate operation, it is recommended to set smaller values, for example, 10–15 cm for parking space sensors and 8–12 cm for barrier sensors. This improves the system's sensitivity and accuracy and reduces the likelihood of incorrect detection.

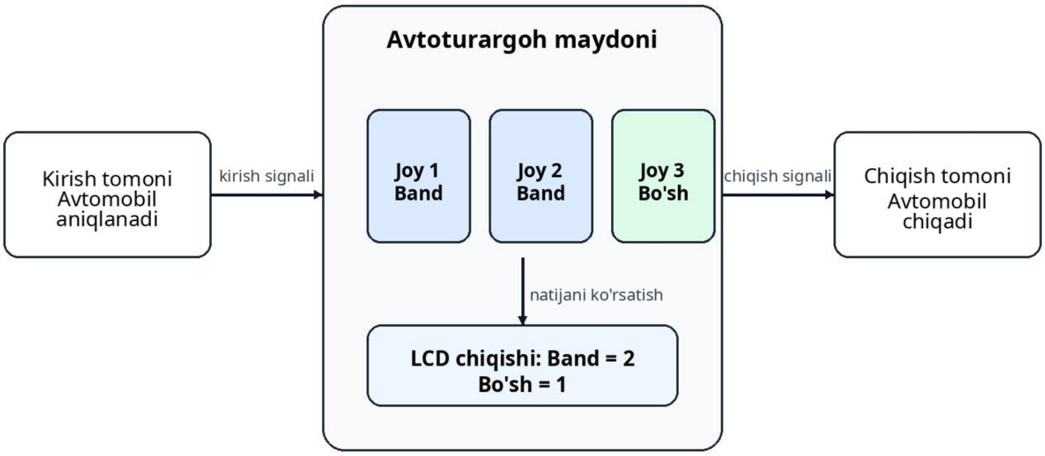



Figure 4. Simplified model of parking state and flow.

This code is convenient for an educational project, and in the future it can be expanded with capabilities such as RFID, Wi-Fi modules, mobile applications, or database integration.

**RESULTS AND TESTING**

**Testing conditions and system performance results**

The project was tested in a simulation environment. To verify the correct operation of the system, several scenarios were applied:

- 2 parking spaces are monitored
- 2 slot sensors detect free/occupied spaces
- 2 gate sensors control the entry and exit gates
- 2 servo motors open the entry and exit barriers
- The LCD displays the number of available spaces
- 2 LEDs indicate space occupancy

During the testing process, sensor values were varied, and the opening-closing state of the servo motors, the information on the display, the LED indicators, and the buzzer response were observed.

INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 8.854 Volume 13, issue 04 (2026)

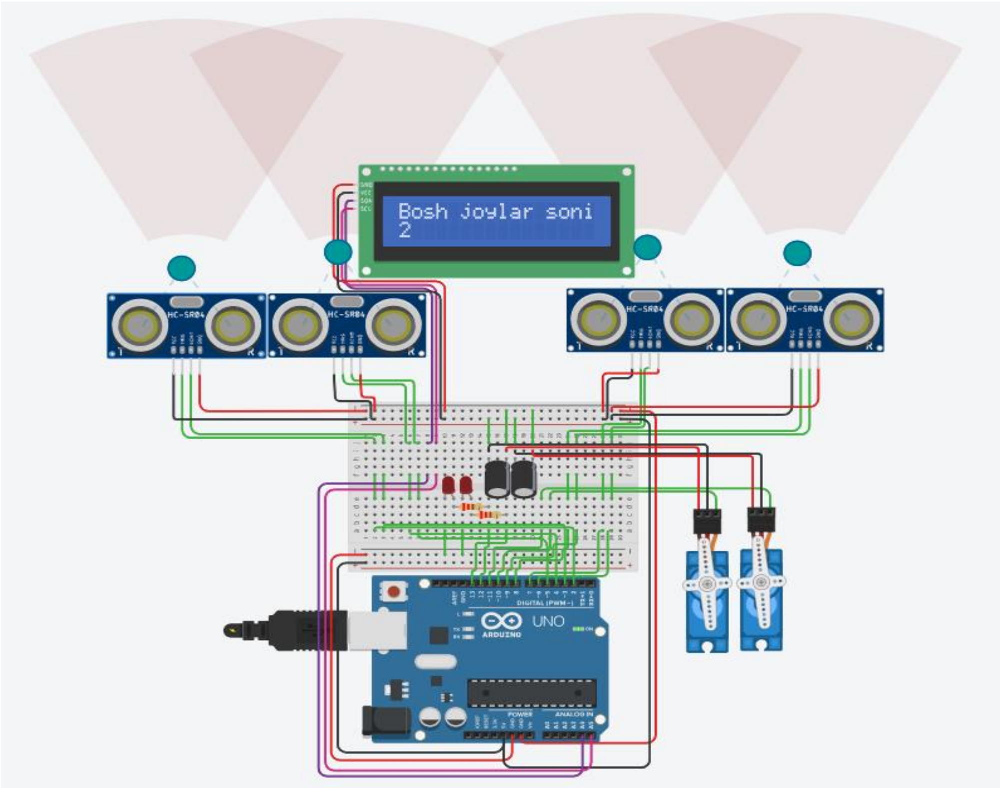



Figure 5. Parking state when it is empty.

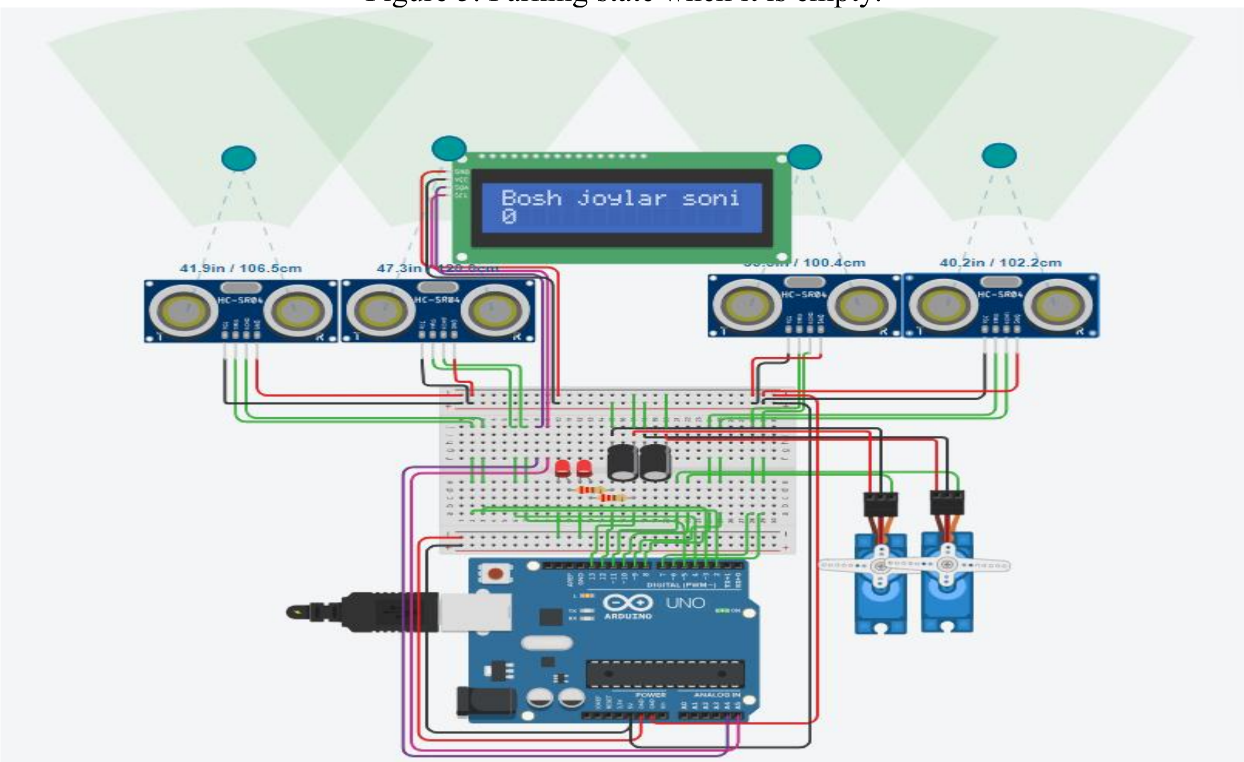



Figure 6. Parking state when it is full.

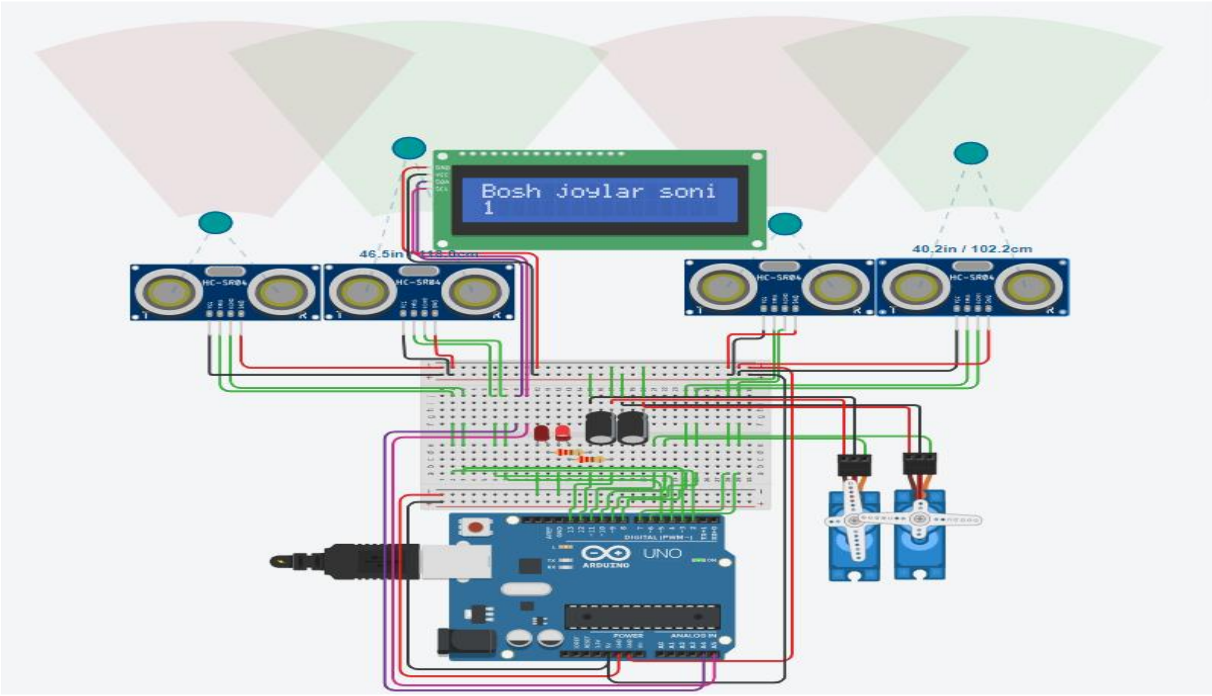


Figure 7. Parking state when one space is available.

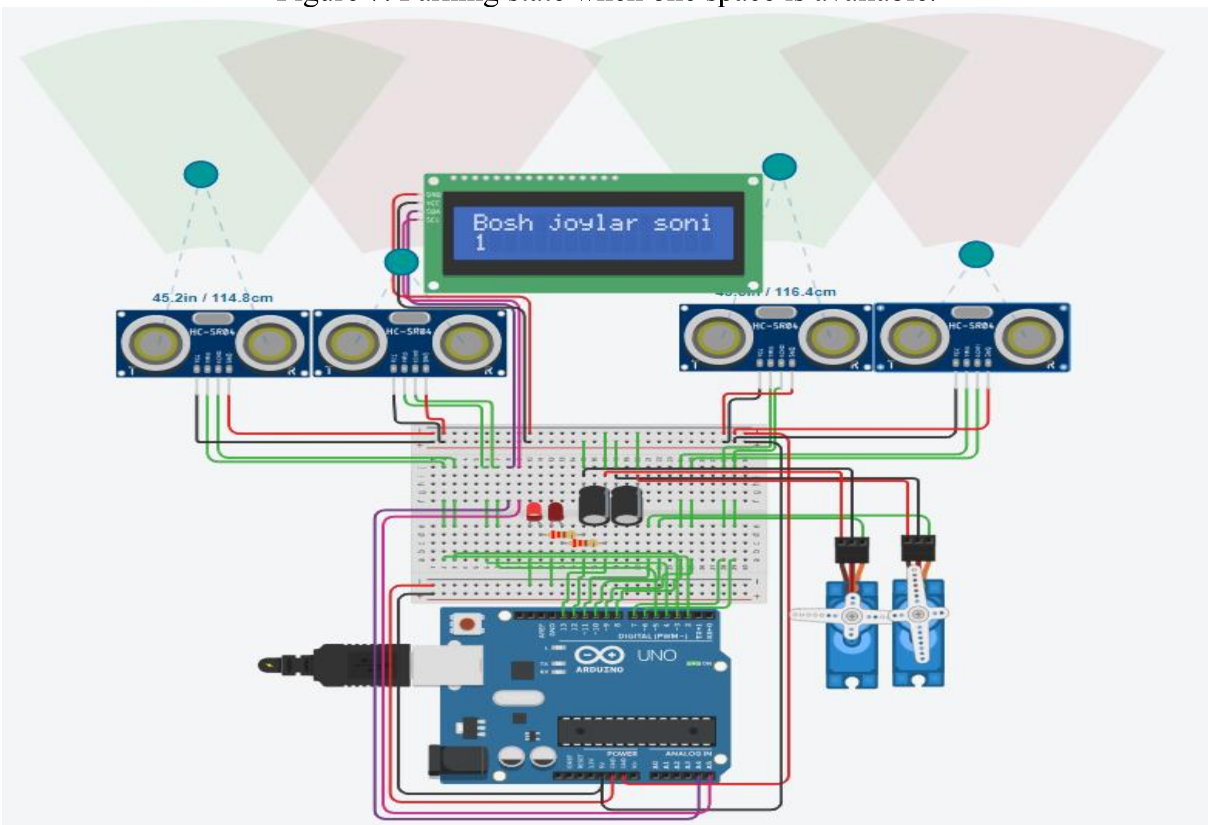


Figure 8. Parking state when one space is available.

**Project link:** https://www.tinkercad.com/things/boOEQ43F7dT-mustaqil-ish/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard&sharecode=3HcYOHQJl3TK7_ibf146vKRLys1_Q4v_ZJZdt8qsFHo

According to the test results, the system successfully performed the following tasks:

First, when a vehicle was detected at the entrance and a free space was available, the servo motor

opened the gate, the buzzer produced a signal, and the number of available spaces on the LCD display decreased. This demonstrated that the system automatically controls the entry process. Second, when all parking spaces were occupied, the system did not allow entry. The message “No space” was displayed on the screen, the red LED turned on, and the gate remained closed. This result confirmed that the system’s protection function operates correctly. Third, when a vehicle was detected at the exit, the system opened the gate, and after the exit was completed, it recalculated and updated the number of available spaces. This confirmed that the system supports bidirectional monitoring. Fourth, the display served as a simple and understandable source of information for the user, showing the number of occupied and available spaces in real time.
Overall, the functional model of the system operated in accordance with its intended purpose and produced positive results as an educational project.

**Observed errors and their solutions**
During testing, some technical issues and limitations were also observed.

1. **Incorrect repeated sensor readings**

**In some cases, the sensor may generate multiple signals for a single vehicle. This leads to incorrect changes in the number of occupied spaces. To solve this problem, delay() and additional condition checks were added to the program.**

2. **Instability of ultrasonic sensor readings**

**The sensor may measure the returned signal inaccurately at different angles and distances. To reduce this, it is recommended to take the average value of several readings.**

3. **Abrupt movement of the servo motor**

**In some simulations, the servo rotates abruptly. To smooth this, a function that changes the servo position step by step can be used.**

4. **Power supply issue**

**In a real device, if the servo motor and other components are powered incorrectly from a single source, the system may operate unstably. Therefore, in a real project, a power source capable of providing sufficient current should be selected.**

5. **Lack of extended monitoring**

**This project is based on local monitoring. If remote monitoring is required, it is necessary to add a Wi-Fi module and communication with a server to the system.**APPLICATION POSSIBILITIES

This automated parking monitoring system has significant practical importance and can be applied in various fields and facilities. The main advantage of the system is that it automatically detects free and occupied parking spaces, controls entry and exit processes, and provides users with fast and accurate information. Therefore, such a system can serve not only as an educational project but also as an effective real-world solution.

First of all, this system is highly suitable for large shopping centers, supermarkets, and entertainment complexes. In such places, the flow of vehicles is very high throughout the day.

As a result, drivers often face difficulties in finding free parking spaces, leading to congestion within parking areas. With the help of an automatic monitoring system, the number of available spaces can be determined in real time and displayed at the entrance. This helps regulate vehicle flow, save time, and provide additional convenience to users.

The system can also be effectively used in parking areas of higher education institutions, colleges, and schools. In cases where the number of vehicles belonging to students, faculty, and staff is high, such a system simplifies the management of available spaces, detection of free spots, and control of vehicle entry and exit. This contributes to the efficient organization of institutional infrastructure.

Furthermore, the system is highly important for hospitals, clinics, and other medical institutions. In such environments, efficient parking management is crucial for ambulances, doctors, and patients. The monitoring system helps quickly identify available spaces and prevents time loss, which is especially critical in emergency situations.

The system is also effective in locations with continuous traffic flow, such as hotels, airports, railway stations, and bus terminals. In such facilities, manual control of parking is inefficient due to constant vehicle movement. An automated system reduces human involvement and ensures fast, accurate, and reliable management, while also improving service quality for users.

In residential complexes, apartment buildings, and private territories, this system can also be applied. In such places, it is important to control vehicle entry based on authorization, restrict unauthorized access, and enhance security. The automated monitoring system can not only count available spaces but also control gates, register incoming vehicles, and monitor activity, thereby increasing security and improving area management.

Additionally, the system can be implemented in government institutions, office buildings, industrial enterprises, and production facilities. In such environments, it helps regulate vehicle movement, control access to restricted areas, and automate traffic flow. Especially in large organizations, this significantly improves management efficiency and convenience.

Another important aspect of the project is its scalability and potential for further improvement. Although the current model is designed mainly for educational and initial practical purposes, it can be significantly enhanced by integrating additional modules and information technologies.

Firstly, by adding RFID cards, QR codes, or NFC technology, automatic identification of vehicles or users can be achieved. This allows quick recognition of authorized vehicles, creation of a subscriber database, and more organized parking management.

Secondly, by integrating GSM, Wi-Fi, or Bluetooth modules, the system can be monitored and controlled remotely. As a result, users or operators can observe the parking status via smartphones, tablets, or computers in real time.

Thirdly, integration with a mobile application can provide users with information about available spaces, access permissions, payment status, or reservation options. This approach is essential for modern smart infrastructure, as it enables direct interaction between users and the system.

Fourthly, by connecting the system to a cloud server or database, collected data can be stored, analyzed, and used to generate reports. For example, daily, weekly, or monthly traffic flow, occupancy rates, and peak usage times can be analyzed, which is valuable for decision-making.

Fifthly, by installing a separate sensor for each parking space, a complete occupancy map can be created. In this case, the system can show not only the total number of free spaces but also exactly which spots are free or occupied, providing precise guidance to users.

Sixthly, the system can be integrated with payment systems, allowing automatic payment processing, time-based billing, and electronic receipt generation. This is particularly useful for commercial parking facilities.

Seventhly, by adding surveillance cameras, license plate recognition, and artificial intelligence elements, the system's security level can be significantly enhanced. It would enable automatic vehicle identification, storage of entry records, and prevention of unauthorized access.

Moreover, this project can also be applied within the concept of smart cities. In modern cities, optimizing traffic flow, reducing environmental impact, and creating convenient digital services for citizens are key priorities. An automated parking monitoring system represents one of such solutions and can become an essential component of urban infrastructure.

In general, this automated parking monitoring system is a promising project with wide application potential. Its practical use covers shopping centers, educational institutions, medical facilities, residential areas, transport hubs, and large organizations. With further technological improvements, the project can evolve from a simple educational model into a modern, intelligent, and reliable parking management system suitable for real-world applications.

## CONCLUSION

In this independent project, the tasks of developing a microcontroller-based automatic parking monitoring system, studying its operating principle, and creating its practical model were successfully accomplished. During the project, the main problems encountered in modern parking systems—such as detecting free and occupied spaces, controlling vehicle entry and exit, and providing fast and convenient information to users—were analyzed. Based on this, an automated control solution relying on embedded systems technology was proposed.

During the implementation process, both hardware and software components of the system were developed in an integrated manner. The Arduino Uno microcontroller was selected as the main control unit, and a functional model consisting of ultrasonic sensors, servo motors, an LCD display, LED indicators, and other auxiliary components was created. With the help of ultrasonic sensors, the presence and distance of vehicles were detected, and the collected data were processed by the microcontroller. As a result, the system became capable of automatically calculating available parking spaces, controlling gates via servo motors, and displaying the current status on an LCD screen.

Throughout the project, the system's block diagram, electrical circuit diagram, pin configuration table, algorithm diagram, and software were developed. This not only helped in understanding the structural and technical design of the device but also provided a clear picture of the system's operational sequence. In the software part, key functions such as receiving signals from sensors, processing data, calculating available spaces, controlling entry and exit barriers, and displaying results were implemented. In this regard, the project served as an effective platform for applying theoretical knowledge in practice.

Simulation and testing processes were among the most important stages of the project. Practical tests showed that the system is capable of effectively performing parking monitoring tasks. In particular, when a vehicle approached the entrance, it was detected by the sensors; if a free space was available, the entrance gate opened, and if no space was available, access was denied. Similarly, when a vehicle was detected at the exit, the system updated the number of available spaces accordingly. These results confirmed that the developed model is functionally correct and logically consistent.

This project contributed to a deeper understanding of the practical importance of embedded systems. During the work, important skills were developed, such as programming microcontrollers, using sensors, connecting hardware components, designing control algorithms, and automating real-world problems. In particular, the experience gained in integrating hardware and software, testing the system, and analyzing errors will be highly valuable in future engineering and automation-related work.

At the same time, the project has great potential for further expansion and improvement. For example, by adding RFID technology, vehicle identification can be implemented; by using IoT technologies, remote monitoring can be achieved; through mobile applications, users can be informed; with databases, statistical reports can be generated; and integration with cameras or license plate recognition systems can be introduced. With such enhancements, the project can evolve from a simple educational model into a fully functional smart parking system applicable in real-world environments.

In general, all the objectives and tasks of this independent project were fully achieved. The working principle of the automatic parking monitoring system was studied, its hardware and software models were developed, its functional capabilities were tested, and an effective solution was proposed. As a result, it was demonstrated that microcontroller-based automated monitoring systems are highly efficient not only theoretically but also practically. This project serves as an important experience in developing modern automation systems and provides a solid foundation for creating more advanced and intelligent control systems in the future.

**REFERENCES**

1. M. A. Mazidi, J. G. Mazidi, and R. D. McKinlay, *The 8051 Microcontroller and Embedded Systems Using Assembly and C*, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2008.
2. M. Margolis, *Arduino Cookbook*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2011.
3. S. Monk, *Programming Arduino: Getting Started with Sketches*, 2nd ed. New York, NY, USA: McGraw-Hill Education, 2016.
4. R. Kamal, *Embedded Systems: Architecture, Programming and Design*, 2nd ed. New Delhi, India: McGraw-Hill Education, 2009.
5. D. Ibrahim, *Mastering the Arduino Uno R4: Programming and Projects for the Minima and WiFi*. Elektor, 2023.

6. Arduino, “Official documentation and library reference.” [Online]. Available: https://www.arduino.cc
7. Wokwi, “Simulation environment practical materials.” [Online]. Available: https://wokwi.com
8. Tinkercad Circuits, “Educational materials and platform guides.” [Online]. Available: https://www.tinkercad.com