

OPERATING SYSTEM

Lab
(2+1 Credit
Hours)

CSL-320



OS Project Report

Project Title: Car Parking System

Submitted by:

Madeeha Talib 02-235221-010

Irma Kaleem 02-235221-018

Sara Tariq 02-235221-032

Submitted To:

Miss Fatima

Miss Faiza Khan



DEPARTMENT OF COMPUTER SCIENCE

BAHRIA UNIVERSITY,
KARACHI, PAKISTAN

ABSTRACT

This project presents a car parking system using an Arduino microcontroller to automate parking space management. The system utilizes infrared (IR) sensors to detect vehicle presence at entry and exit points, while a servo motor controls the gate barrier based on real-time parking availability. A 20x4 LCD display provides user-friendly information about free slots and entrance/exit statuses. This Arduino-based solution offers cost-effective and efficient parking management, eliminating manual oversight and streamlining vehicle flow.

Keywords: Arduino, Car Parking System, IR sensors, Servo Motor, LCD Display, Real-time information, Automation.

Table of Contents

Car parking System:

| | |
|--------------------------------|----|
| ABSTRACT..... | 1 |
| 1. Introduction: | 3 |
| 2. Key Features: | 3 |
| 3. Key Components: | 3 |
| 4. System Functionality: | 4 |
| 5. Circuit Diagram[5]: | 5 |
| 6. Code: | 5 |
| 7. Output: | 10 |
| 8. Conclusion: | 20 |
| 9. References: | 22 |

1. Introduction:

The Car Parking System is designed to automate the process of managing parking spaces. The Arduino-based system incorporates IR sensors to detect vehicle presence at entrance and exit points. A servo motor controls the gate barrier, allowing or denying access based on available parking slots. The 20x4 LCD Display provides real-time information about available slots and entrance/exit statuses [1].

2. Key Features:

- The Automated parking slot management.
- Real-time display of available parking spaces.
- Efficient gate control using IR sensors and a servo motor

3. Key Components:

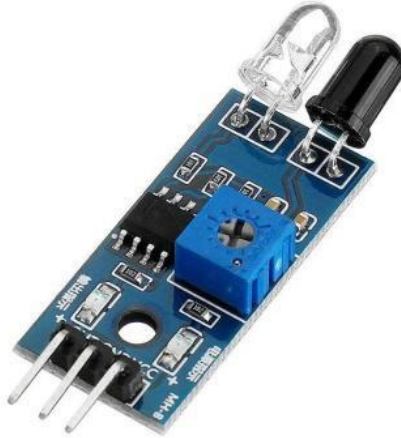
Arduino Uno: Central control unit for system operations[2].



20x4 LCD Display: Presents parking slot availability and system messages.



IR Sensors: Detects vehicle presence at entrance and exit points.



MG995 Servo Motor: Controls the gate barrier to allow or deny access.



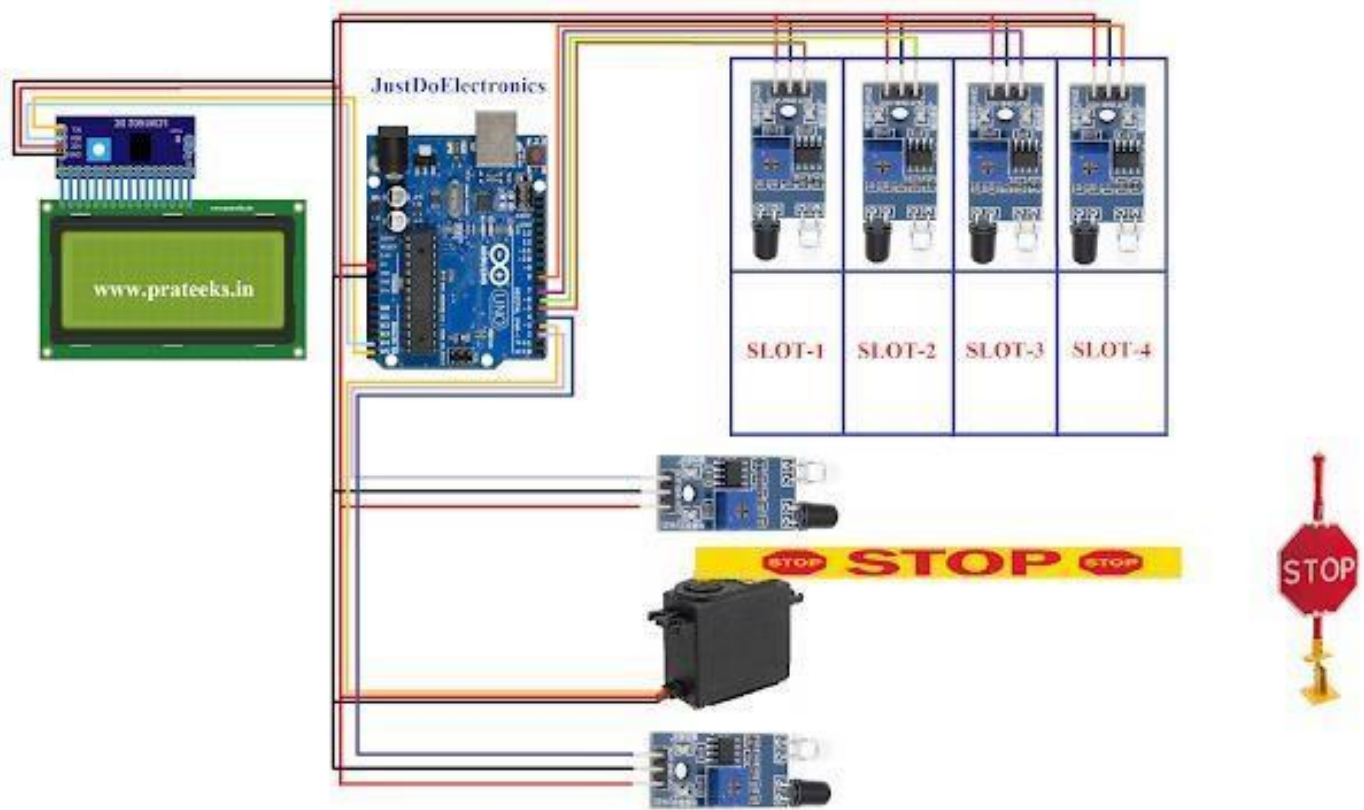
4. System Functionality:

Entrance Monitoring: IR sensors detect incoming vehicles. If space is available, the gate barrier opens, and the vehicle is allowed to enter. Otherwise, entry is denied.

Exit Monitoring: IR sensors detect exiting vehicles. The gate barrier opens, allowing vehicles to exit while updating the available parking slots [3].

LCD Display: Provides real-time updates on total slots, available spaces, and status of individual parking slots [4].

5. Circuit Diagram [5]:



6. Code:

```
#include <TaskScheduler.h>
#include <Wire.h>#include <LiquidCrystal_I2C.h>
#include <Servo.h>
Scheduler scheduler;
LiquidCrystal_I2C lcd(0x27, 20, 4);
Servo myservo1;
#define ir_s1 2
// Entrance sensor
#define ir_s2 4
// Exit sensor
#define Total 4
// Limited parking slots
#define ir_car1 5 // Car sensor for slot 1
```

```

#define ir_car2 6 // Car sensor for slot 2
#define ir_car3 7 // Car sensor for slot 3
#define ir_car4 8 // Car sensor for slot 4
int Space;
int flag1 = 0;
int flag2 = 0;
// Sensor variables
int S1 = 0, S2 = 0, S3 = 0, S4 = 0;
// Declare Read_Sensor function
void Read_Sensor();
// Define callback functions for tasks
void task1(); void task2(); void
task3(); // Define Task objects
Task t1(1000, TASK_FOREVER, &task1);
Task t2(2000, TASK_FOREVER, &task2);
Task t3(100, TASK_FOREVER, &task3);
void setup() {
  pinMode(ir_s1, INPUT);
  pinMode(ir_s2, INPUT);
  pinMode(ir_car1, INPUT);
  pinMode(ir_car2, INPUT);
  pinMode(ir_car3, INPUT);
  pinMode(ir_car4, INPUT);
  myservo1.attach(3);
  myservo1.write(100);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 1);
  lcd.print(" Hi Welcome To ");
  lcd.setCursor(0, 2); lcd.print("
Sara Tariq"); delay(5000);

```

```

lcd.clear(); lcd.setCursor(0, 0);
lcd.print(" Today's Project ");
lcd.setCursor(0, 1);
lcd.print(" Car Parking ");
lcd.setCursor(0, 2);
lcd.print("
System ");delay(5000);
lcd.clear();
Space = Total;
Total= S1+S2+S3+S4;
Serial.begin(9600); // Initialize serial communication
scheduler.init();
scheduler.addTask(t1);
scheduler.addTask(t2);
scheduler.addTask(t3);
scheduler.enableAll(); // Start all tasks
}
void loop() {
scheduler.execute();
// Your existing loop logic for parking system
if (digitalRead(ir_s1) == LOW && flag1 == 0) {
// Existing logic for car entrance // Enqueue the car ID or any
identifier if (Space > 0) { flag1 = 1; if (flag2 == 0) {
myservo1.write(0); Space = Space - 1; carQueue.enqueue(1); //
Enqueue the car ID or any identifier
}
} else {lcd.setCursor(0, 0); lcd.print("
Parking Full "); lcd.setCursor(0,
1); lcd.print(" No Entry
Allowed "); delay(1000);
lcd.clear();

```



```

}
}
if (flag1 == 1 && flag2 == 1) {
// Existing logic when both flags are set
delay(1000); myservo1.write(100);
flag1 = 0, flag2 = 0;
}
lcd.setCursor(0, 0);
lcd.print("Total Slots: ");
lcd.print(Total); lcd.setCursor(0,
1); lcd.print("Available: ");
lcd.print(Space);
}
void task1() {
// Car entrance logic
Serial.println("Car entered the parking lot");
lcd.clear(); lcd.setCursor(0, 0);
lcd.print("Car Entered: Slot ");
lcd.print(Total - Space);
lcd.setCursor(0, 1); lcd.print("Available: ");
lcd.print(Space);
}
void task2() {
// Car exit logic
Serial.println("Car exited the parking lot");
lcd.clear(); lcd.setCursor(0,
0); lcd.print("Car Exited:
Slot "); lcd.print(Total -
Space + 1); lcd.setCursor(0,
1); lcd.print("Available: ");
lcd.print(Space);

```

```

}

void task3() {
// Read sensors and update display
Read_Sensor();
lcd.setCursor(0, 0);
lcd.print(" Available Slot: ");
lcd.print(Space); lcd.print("
");
lcd.setCursor(0, 1);
lcd.print("S1:");
lcd.print(S1 == 1 ? "Full " : "Empty");
lcd.setCursor(11, 1);
lcd.print("S2:");lcd.print(S2 == 1 ? "Full " : "Empty");
lcd.setCursor(0, 2);
lcd.print("S3:");
lcd.print(S3 == 1 ? "Full " : "Empty");
lcd.setCursor(11, 2);
lcd.print("S4:");
lcd.print(S4 == 1 ? "Full " : "Empty");
}

// Function to read sensor values
void Read_Sensor() {
S1 = digitalRead(ir_car1);
S2 = digitalRead(ir_car2);
S3 = digitalRead(ir_car3);
S4 = digitalRead(ir_car4);
}

```

7. Output:

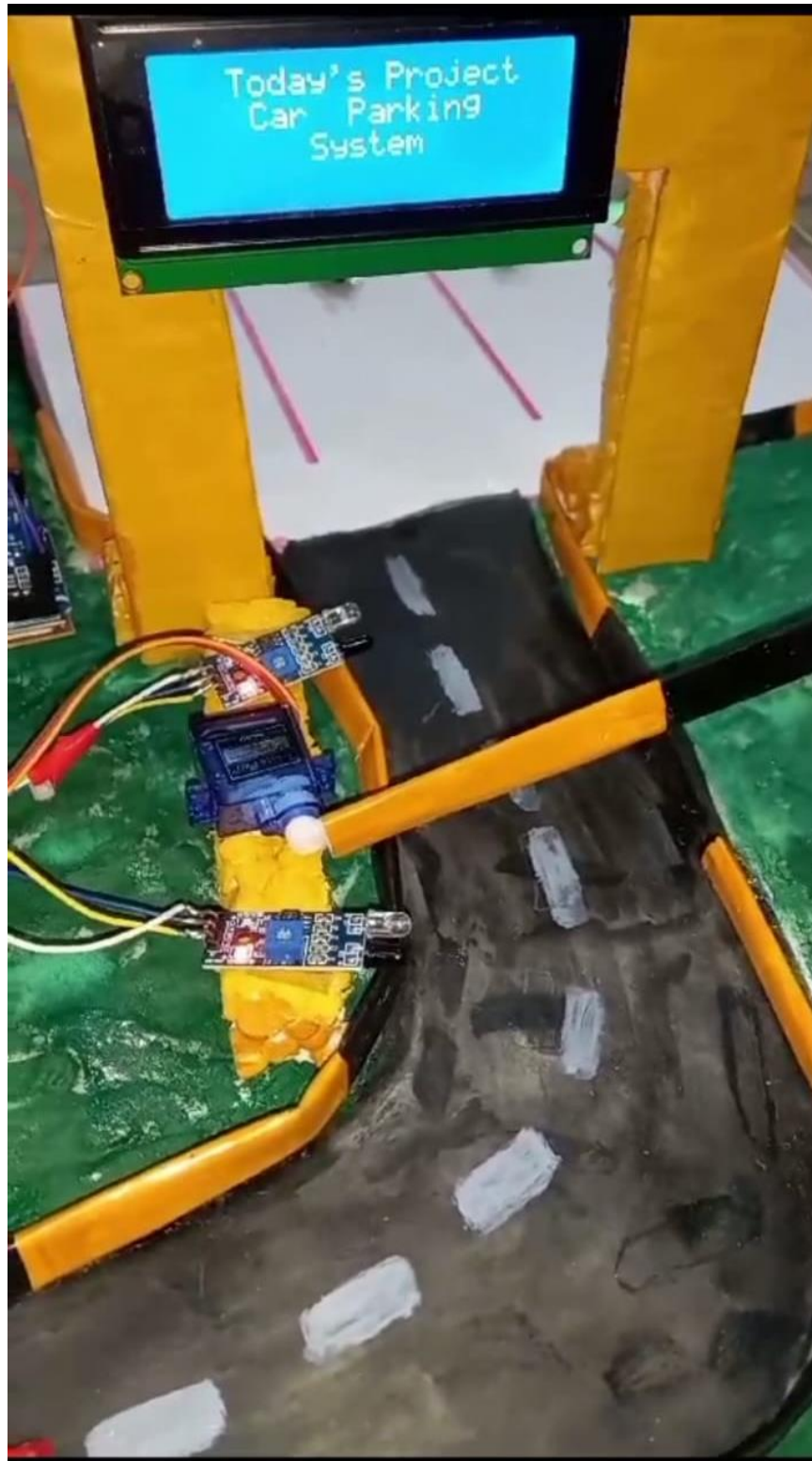


Figure: 1 (Starting of project)



Figure: 02(Available Slots)

In figure 2, we show you how many slots are available for parking. Right now there is no car is parking so all slots are available.



Figure: 03 (1 car parked)

In figure 3, we show that one car is parked now.



Figure: 04 (S4 is full)

In figure 4, S4 is showing full because there is one car parked on S4 spot.



Figure: 05 (2 cars parked)

In figure 5, we show that two cars are parked now.



Figure: 06 (S3 is full)

In figure 6, S3 is showing full because there are two cars parked on S4 and S3 spot.



Figure: 07 (3 cars parked)

In figure 7, we show that three cars are parked now.



Figure: 08 (S2 is full)

In figure 8, S2 is showing full because there are three cars parked on S4, S3 and S2 spot.



Figure: 09 (4 cars parked)

In figure 9, we show that four cars are parked now.



Figure: 10 (S1 is full)

In figure 10, S1 is showing full because there are four cars parked on S4, S3, S2 and S1 spot.



Figure: 10 (Final work)

8. Conclusion:

Car Parking System successfully demonstrates an automated solution for managing parking spaces using Arduino Uno, IR Sensors, Servo Motor, and an LCD Display. The system efficiently controls vehicle entry and exit, effectively managing available parking slots and providing users with real-time parking status updates.

9. References:

[1] Author name: Prateek

Date/year: 16/06/2023

Link: <https://justdoelectronics.com/car-parking-system-using-arduino/>

[2] Author name: arduino.cc

Date/year: 2022-10-27

Link: https://en.wikipedia.org/wiki/Arduino#Further_reading

[3] Author name: Electrobes

Date/year: 24-08-2022

Link: <https://electrobes.com/product/ir-infrared-obstacle-avoidance-sensor-module-for-arduino/>

[4] Author name: Majju PK

Date/year: 10-06-2012

Link: <https://www.majju.pk/product/lcd-16x2-1602-alphanumeric-display-module-blue-backlight-16-character-2-line/>

[5] Author name: Prateek

Date/year: 16/06/2023

Link: <https://justdoelectronics.com/car-parking-system-using-arduino/>