

Project Report

COM382 MICROPROCESSORS

**Car Parking for Disabled People.**

Prepared by

Rashad Karaki B2205.010018 Sec (3)

Mohanad Marzoqi B2105.020002 Sec (1)

Ali Hamandi B2205.090018 Sec (2)

Ahmad Musa B2205.020010 Sec (2)

Turki Abdallah B2105.020037 Sec(2)

Supervisor

Prof. Dr. MHD WASIM RAED

Spring 2023-24

**Presentation video using google drive link:**

<https://drive.google.com/file/d/1BZ9NbHq0VGi8tyY4YZjTwrTcVlhivH_e/view?usp=sharing>  
 **Abstract**

We designed a smart parking system that increases the accessibility of parking easier especially for disabled people, while organizing parking and reduces crowding in the parking space. This is done using Arduino microcontroller which controls all operations, an ultrasonic sensor positioned at the parking lot entrance to detect incoming cars and increases the count of the cars in the parking spot, IR sensors placed in each parking spot to check whether a car is parked, Servo Motor Controls the gate to allow cars to enter and exit the parking lot based on the ultrasonic sensor , and finally an LCD screen which displays real-time information about the number of cars in the parking lot and the available parking spaces.

**Table of Contents**

**Abstract ........................................................................................................... 1**

**Introduction .................................................................................................... 3**

**Background ..................................................................................................... 3**

**Literature Review ............................................................................................4**

**Project Equipment and Components .............................................................5**

**System Design and Architecture .....................................................................6**

**Software Implementation ................................................................................8**

**Testing and Results .........................................................................................10**

**Large-Scale Implementation ..........................................................................11**

**Conclusion .......................................................................................................11**

**References .......................................................................................................11**

***1. INTRODUCTION***

There are some significant challenges faced by disabled people in urban environment is finding and accessing a parking spot with ease which leads to several problems such as , time consuming, the process of finding a good parking spot can take a considerable amount of time which in turn becomes physically exhausting especially in crowded parking lots, secondly even if there are a designated parking spot for disabled people locating these spots can be in adequate which leads to multiple accessibility issues. The smart parking system optimizes parking accessibility for disabled people using the Arduino microcontroller and reducing the need for manual supervision which decreases cost in the long term, it can be implemented in commercial parking lots, shopping centers and public parking lots.

***2. Background***

The demand of accessible parking spaces has been increasing over the years, traditional parking systems have failed to provide the support and assistance needed for disabled people which makes it an inconvenience for disabled people which in turn leads to inefficiency.

By the help of the advancing technology in the space of microcontrollers and sensors, using them can provide a more efficient process of parking for people in general and for disabled individuals especially.

This project makes use of these technologies to provide a seamless and more convenient parking experience.

***3. Literature review***

Nandyal, S., Sultana, S., & Anjum, S. (2017).

This article focused on developing a smart parking system with the usage of Arduino microcontroller to manage parking spaces and reduce the time that the drivers need to spend to find parking spots using LCD display to provide information about available parking spots which is considered as a significant improvement in smart parking systems with the primary goal on the general population without addressing the needs for disabled people.

Our system takes advantage of the LCD Display that provides real-time information about the status of parking spots, and the LED lights to guide the disabled people to the suitable parking spots to enhance accessibility while using IR sensor for detecting the presence of a vehicle.

Another study has that ca a noticeable contribution on the field on smart parking system is Elsonbaty, A., & Shams, M. (2020). which specify and delve in the design of a whole smart parking management system which leverages technologies to ease parking process, by providing real-time information about available parking spots and using a user interface that displays and guides drivers to free spots but the main focus of the project in its focus on the general population rather than disabled individuals.

Our system utilizes the advancement made in this system along with adding other features such as LCD display which provides real-time information about the number and the where about of the available spots for disabled individuals along with a servo motor the controls gate entrance for automated gate control based on Arduino decisions.

***4. Project Equipment and Components***

1. Arduino Uno

2. Ultrasonic Sensor (HC-SR04)

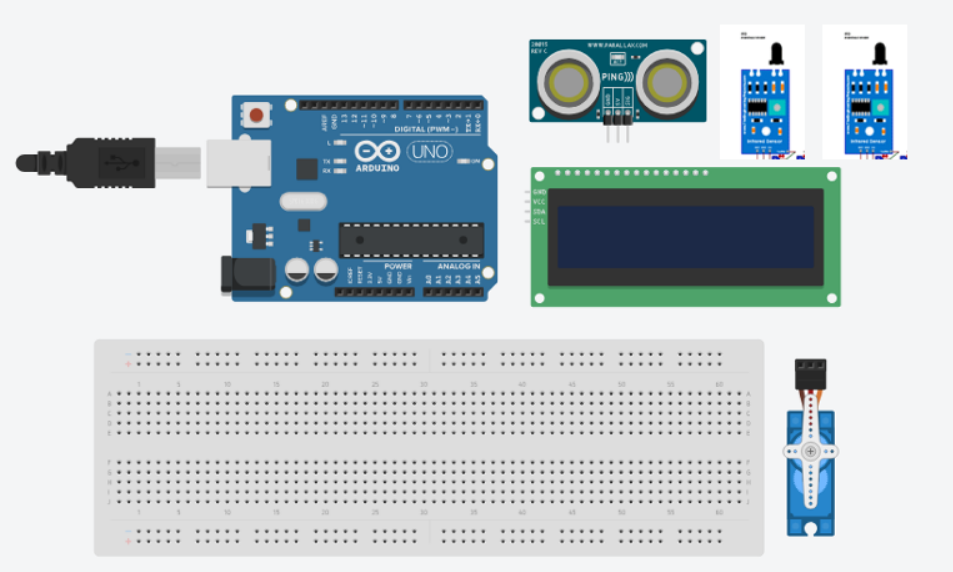
3. IR Sensors (2 units)

4. Servo Motor

5. 16x2 LCD Display with I2C Module

6. Computer Power Supply (ATX)

7. Breadboard

8. Jumper Wires  
  
  


**Fig 1 . Project Components.**

***5. System Design and Hardware Implantation***

1. *Overview*

The Smart Car Parking System is designed to detect the presence of cars approaching a parking gate, manage the opening and closing of the gate, count the number of cars, and monitor the occupancy status of designated parking spots for disabled individuals. The system provides real-time feedback via an LCD display and optional LEDs.

1. *Components and Connections*

**Arduino Uno**

The Arduino Uno serves as the central controller, interfacing with the sensors, servo motor, and LCD display. It processes input signals from the sensors and controls the output signals to the servo motor and LCD.

**Ultrasonic Sensor (HC-SR04)**

* **Trigger Pin**: Connected to digital pin 2 of the Arduino.
* **Echo Pin**: Connected to digital pin 3 of the Arduino.
* **VCC**: Connected to the 5V power supply.
* **GND**: Connected to ground.

The ultrasonic sensor detects the presence of an approaching car by measuring the distance to an object.

**IR Sensors**

* **IR Sensor 1**: Connected to digital pin 4 of the Arduino.
* **IR Sensor 2**: Connected to digital pin 5 of the Arduino.
* **VCC**: Connected to the 5V power supply.
* **GND**: Connected to ground.

The IR sensors detect whether the parking spots are occupied by a car.

**Servo Motor**

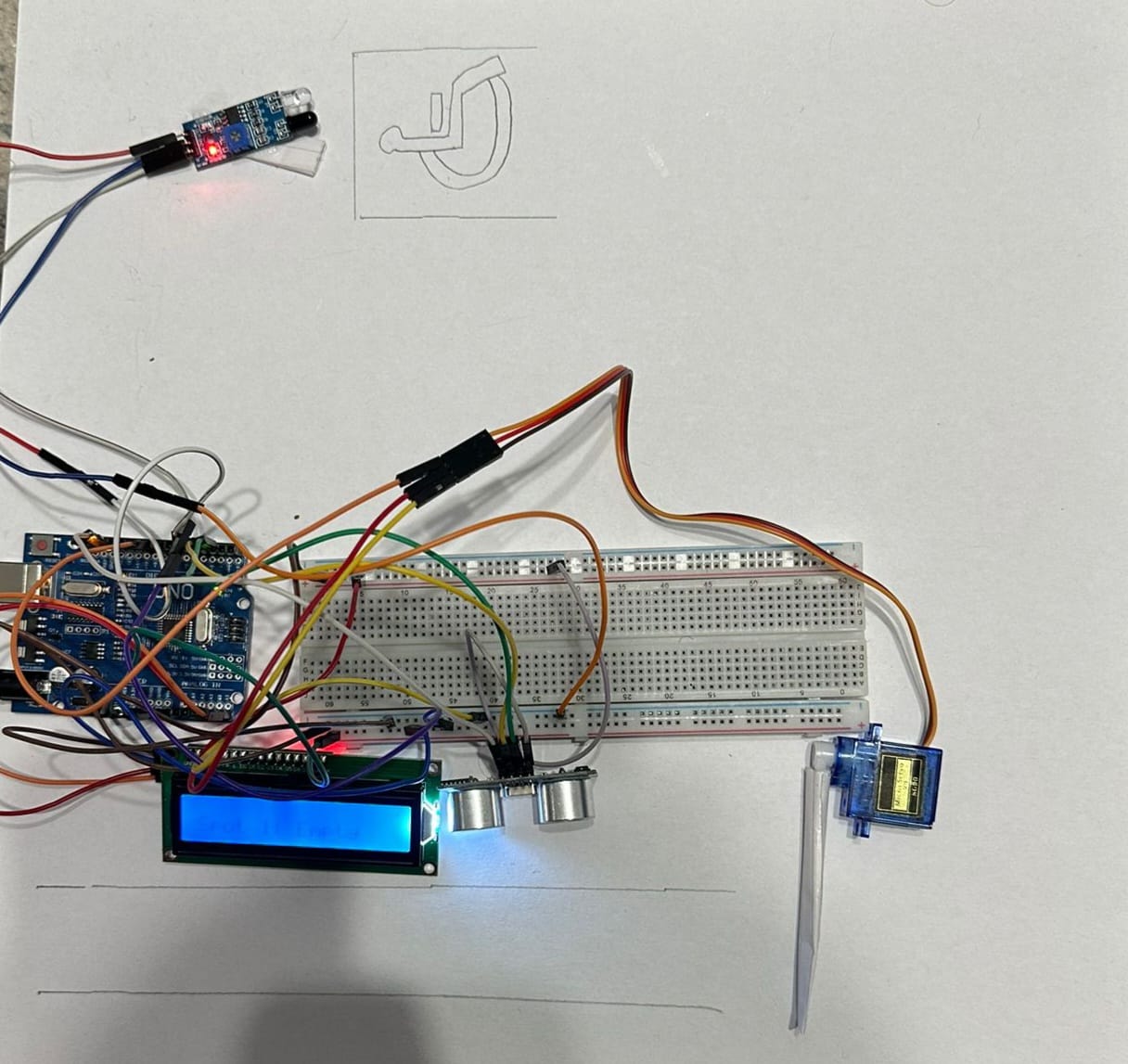
* **Control Pin**: Connected to digital pin 9 of the Arduino.
* **VCC**: Connected to the 5V power supply.
* **GND**: Connected to ground.

The servo motor controls the opening and closing of the parking gate.

**LCD Display with I2C Module**

* **SDA**: Connected to analog pin A4 of the Arduino.
* **SCL**: Connected to analog pin A5 of the Arduino.
* **VCC**: Connected to the 5V power supply.
* **GND**: Connected to ground.

The LCD display shows real-time information, including the number of cars and the status of the parking spots.



**Fig 2 . Project Hardware.**

***6. Software Implementation***

1. *Explanation*

The software for the Smart Car Parking System manages the interactions between the Arduino Uno, sensors, servo motor, and LCD display.

**1. Libraries**

The following libraries are used in the code:

* **Wire.h**: This library allows communication with devices using the I2C protocol.
* **LiquidCrystal\_I2C.h**: This library facilitates communication with the LCD display using the I2C interface.
* **Servo.h**: This library enables control of the servo motor.

**2. Setup Function**

* Initializes communication and sets pin modes.
* Attaches the servo motor and initializes the LCD display.
* Displays a startup message on the LCD.

**3. Loop Function**

* **Car Detection**: Checks if a car is approaching using the ultrasonic sensor.
* **Gate Control**: Opens and closes the gate accordingly.
* **Car Counting**: Increments the car count when a car passes through.
* **Parking Spot Monitoring**: Checks the occupancy status of parking spots using IR sensors.
* **Updates LCD**: Displays real-time information on the LCD.

**4. Delay**

* **Loop Delay**: Adds a delay of 1 second at the end of each loop iteration to control the sampling rate of the system.

**5. Functions**

* **openGate()**: Opens the gate by rotating the servo motor.
* **closeGate()**: Closes the gate after a delay.
* **updateLCD()**: Updates the LCD display with current information.

*B. Code*

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <Servo.h>

Servo gateServo;

const int trigPin = 2;

const int echoPin = 3;

const int irPin1 = 4;

int carCount = 0; // Counter for cars that have entered

bool spot1Occupied = false;

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Set the LCD I2C address

void setup() {

  Serial.begin(9600);

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  pinMode(irPin1, INPUT);

  gateServo.attach(9); // attach the servo on pin 9

  gateServo.write(0); // initial position of gate

  lcd.init(); // initialize the lcd

  lcd.backlight(); // turn on the backlight

  lcd.setCursor(0, 0);

  lcd.print("Parking System");

  Serial.println("Parking System Initialized");

  delay(2000);

  lcd.clear();

}

void loop() {

  long duration, distance;

  // Trigger the ultrasonic sensor

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  // Read the echo pin

  duration = pulseIn(echoPin, HIGH);

  distance = (duration / 2) / 29.1;

  // Check parking spot occupancy

  spot1Occupied = digitalRead(irPin1) == LOW; // assuming LOW means occupied

  updateLCD();

  if (carCount >= 5 && spot1Occupied) {

    // If car count is 5 or more and spot 1 is occupied, parking is considered full

    lcd.setCursor(0, 1);

    lcd.print("Parking is Full");

    Serial.println("Parking Full. Gate not opening.");

  } else if (distance < 20 && !spot1Occupied) { // if a car is detected within 20 cm and spot 1 is not full

    Serial.println("Car detected");

    openGate();

    delay(3000); // wait for car to pass

    closeGate();

    carCount++; // Increase car count only when a car enters and spot was previously not occupied

    updateLCD();

  }

  delay(1000);

}

void openGate() {

  if (!spot1Occupied && carCount < 5) {

    gateServo.write(90); // open the gate

    Serial.println("Gate opened");

    delay(1000);

  }

}

void closeGate() {

  gateServo.write(0); // close the gate

  Serial.println("Gate closed");

  delay(1000);

}

void updateLCD() {

  lcd.clear();

  lcd.setCursor(0, 0);

  lcd.print("Cars: ");

  lcd.print(carCount);

  lcd.setCursor(0, 1);

  lcd.print("Spot 1: ");

  lcd.print(spot1Occupied ? "Occupied" : "Empty");

  // Print to Serial Monitor

  Serial.print("Cars: ");

  Serial.println(carCount);

  Serial.print("Spot 1: ");

  Serial.println(spot1Occupied ? "Occupied" : "Empty");

}

***7. Testing and results***

We conducted thorough testing on our smart parking system to ensure it worked well for disabled individuals. First, we tested each part—like sensors and indicators—using the Arduino IDE Cloud's serial monitor. This tool helped us check if all parts were working correctly by showing real-time outputs and helping us fix any issues.

Next, we put the whole system to the test by simulating real parking conditions. We used the serial monitor again to watch how the system handled these scenarios, ensuring everything worked together smoothly. These tests confirmed that our system could reliably detect available parking spots and perform well, making parking easier and more accessible for disabled drivers. ***8. Large-Scale Implementation***

The Smart Car Parking System's large-scale deployment is ideal for buildings with multiple floors and numerous disabled parking lots. It offers a streamlined solution for disabled individuals to swiftly identify vacant parking spots upon entering the premises, thereby saving time and effort. This implementation boasts real-time monitoring of parking lot occupancy. Its scalability makes it adaptable to various building layouts, enhancing accessibility and efficiency in parking management.

***9. Conclusion***

Our project successfully developed a reliable and efficient smart parking system specifically designed for disabled individuals. The system demonstrated high accuracy and robust performance under real-world conditions, notably improving the parking experience for disabled drivers. Through this initiative, we have taken a significant step towards creating more accessible and inclusive public spaces, reflecting our commitment to enhancing urban infrastructure for individuals with disabilities.

***References***

* 1. Nandyal, S., Sultana, S., & Anjum, S. (2017). Smart car parking system using arduino uno. International Journal of Computer Applications, 169(1), 13-18.
  2. Elsonbaty, A., & Shams, M. (2020). The smart parking management system. *arXiv preprint arXiv:2009.13443*.