SMART PARKING SYSTEM

Project Report submitted in partial fulfillment of
The requirements for the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY
Of

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, WEST BENGAL

By

SRAYOAN RAY, 112, 10900220113
SOUVIK GHOSH, 72, 10900220073
SOUMYAJYOTI MUKHERJEE,76,10900220077
SOUMIK MONDAL, 85, 10900220086
SUVAM METIA, 136,10901320008

Under the guidance of

Prof. SUKANTA BOSE

DEPARTMENT OF INFORMATION TECHNOLOGY

NETAJI SUBHASH ENGINEERING COLLEGE
TECHNO CITY, GARIA, KOLKATA – 700 152

<< Academic year 2023-24>>

CERTIFICATE

This is to certify that this project report titled SMART PARKING SYSTEM submitted in partial fulfillment of requirements for award of the degree Bachelor of Technology (B. Tech) in INFORMATION TECHNOLOGY of MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY is a faithful record of the original work carried out by,

SRAYOAN RAY, 112, 10900220113, 201090100210002 OF 2020-21
SOUVIK GHOSH, 72, 10900220073, 201090100210042 OF 2020-21
SOUMYAJYOTI MUKHERJEE,76,10900220077, 201090100210038 OF 2020-21
SOUMIK MONDAL, 85, 10900220086, 201090100210029 OF 2020-21
SUVAM METIA, 136,10901320008, 201090100210192 OF 2020-21

under my guidance and supervision.

It is further certified that it contains no material, which to a substantial extent has been submitted for the award of any degree/diploma in any institute or has been published in any form, except the assistance drawn from other sources, for which due acknowledgement has been made.

Date:……..

_____
Guide's signature

Prof. SUKANTA BOSE

Sd/_____

Head of the Department

INFORMATION TECHNOLOGY

NETAJI SUBHASH ENGINEERING COLLEGE
TECHNO CITY, GARIA, KOLKATA – 700 152

DECLARATION

We hereby declare that this project report titled

SMART PARKING SYSTEM

is our own original work carried out as an undergraduate student in Netaji Subhash Engineering College except to the extent that assistance from other sources are duly acknowledged.

All sources used for this project report have been fully and properly cited. It contains no material which to a substantial extent has been submitted for the award of any degree/diploma in any institute or has been published in any form, except where due acknowledgement is made.

Student's names:	Signatures:	Dates:
SRAYOAN RAY	Srayoan Ray	17.05.2024
SOUVIK GHOSH	Souvik Ghosh	17.05.2024
SOUYMAJYOTI MUKHERJEE	Soumyajyoti Mukherjee	17.05.2024
SOUMIK MONDAL	Soumik Mondal.	17.05.2024
SUVAM METIA	Suvam Metia.	17.05.2024

CERTIFICATE OF APPROVAL

We hereby approve this dissertation titled

SMART PARKING SYSTEM

carried out by

SRAYOAN RAY, 112, 10900220113, 201090100210002 OF 2020-21
SOUVIK GHOSH, 72, 10900220073, 201090100210042 OF 2020-21
SOUMYAJYOTI MUKHERJEE,76,10900220077, 201090100210038 OF 2020-21
SOUMIK MONDAL, 85, 10900220086, 201090100210029 OF 2020-21
SUVAM METIA, 136,10901320008, 201090100210192 OF 2020-21

under the guidance of

Prof. SUKANTA BOSE

of Netaji Subhash Engineering College, Kolkata in partial fulfillment of requirements for award of the degree Bachelor of Technology (B. Tech) in INFORMATION TECHNOLOGY of MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

Date:……..

Examiners' signatures:

1. …….

2. …….

Acknowledgement

Firstly, we would like to express our sincere gratitude to our project guide, Prof. Sukanta Bose, for his continuous support, encouragement, and invaluable guidance throughout the development of this project.

We extend our heartfelt thanks to the faculty members of the Department of Information Technology of Netaji Subhash Engineering College for providing the necessary resources and infrastructure, as well as for their constructive feedback and encouragement.

We are also thankful to our group members who provided us with moral support, technical assistance, and constructive criticism, which greatly helped us in refining my project.

Special thanks to our families for their unwavering support and understanding, which has been a constant source of motivation during this journey.

Lastly, we would like to acknowledge all the online communities and forums where we found valuable information and solutions to various technical challenges faced during the project.

Thank you all for your contributions, support, and encouragement.

SRAYOAN RAY

SOUVIK GHOSH

SOUYMAJYOTI MUKHERJEE

SOUMIK MONDAL

SUVAM METIA

Dated:**17.05.2024**……

Abstract

This project, titled "SMART PARKING SYSTEM" aims to develop a smart parking solution that efficiently manages vehicle entry and exit while providing real-time parking availability information. The system leverages ultrasonic sensors, a servo motor, LEDs, a buzzer, and an LCD display, all controlled by an Arduino Uno microcontroller.

The automated parking system features a barrier controlled by a servo motor, which opens and closes based on the detection of vehicles using ultrasonic sensors. The traffic light system, consisting of red, yellow, and green LEDs, regulates the entry and exit of vehicles, ensuring smooth traffic flow. The LCD display provides real-time updates on the availability of parking slots, enhancing user convenience.

The project is designed to address the challenges of manual parking management by offering an automated solution that reduces human intervention, enhances efficiency, and improves the user experience. By integrating various electronic components and sensors, this system demonstrates a practical application of IoT (Internet of Things) in the field of smart parking.

The project includes detailed circuit design, component integration, and programming to achieve the desired functionality. It also incorporates safety features, such as a buzzer alert for full parking capacity, ensuring effective communication of parking status to users. This innovative approach to parking management showcases the potential of IoT technologies in creating smarter and more efficient urban infrastructure.

CONTENTS

INTRODUCTION

With the rapid urbanization and increasing number of vehicles, efficient management of parking spaces has become a critical challenge in modern cities. Traditional parking systems often suffer from issues such as space mismanagement, human error, and inefficiency, leading to traffic congestion and frustration among drivers. To address these problems, this project presents a "SMART PARKING SYSTEM."

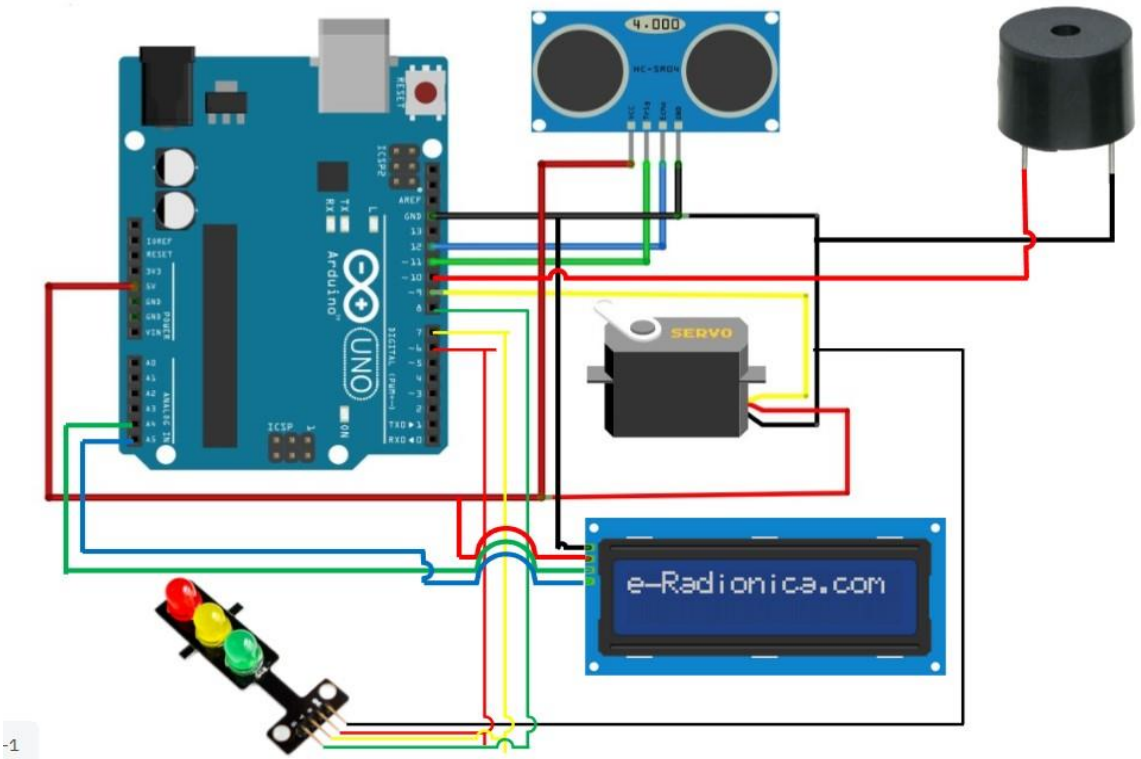
The primary objective of this project is to develop a smart parking solution that automates the process of vehicle entry and exit, manages traffic flow, and provides real-time information about parking slot availability. The system is built around the Arduino Uno microcontroller, which integrates various components, including ultrasonic sensors, a servo motor, LEDs, a buzzer, and an LCD display.

The ultrasonic sensors are used to detect the presence of vehicles at the entrance and exit points. Based on the sensor inputs, the Arduino controls a servo motor that operates a barrier, allowing or restricting vehicle access. The traffic light system, consisting of red, yellow, and green LEDs, provides visual cues to drivers, ensuring orderly movement. The LCD display serves as a user interface, showing the number of available parking slots, which updates in real-time as vehicles enter or leave the parking area.

This automated parking system not only enhances the efficiency of parking space utilization but also minimizes the need for human intervention, thereby reducing operational costs and errors. Additionally, the inclusion of a buzzer provides an audible alert when the parking lot is full, preventing unnecessary congestion at the entrance.

By implementing this project, we aim to demonstrate the practical application of IoT (Internet of Things) in developing smart city solutions. This project highlights the potential of using embedded systems and sensor technology to solve real-world problems, providing a foundation for future advancements in automated and intelligent parking systems.

CIRCUIT DIAGRAM



Components and Connections:

1. **Arduino Uno**:
    - Central microcontroller that controls all components.
2. **Ultrasonic Sensor**:
    - **VCC**: Connects to the Arduino 5V pin.
    - **GND**: Connects to the Arduino GND pin.
    - **Trig**: Connects to Arduino digital pin 2.

- **Echo**: Connects to Arduino digital pin 3.

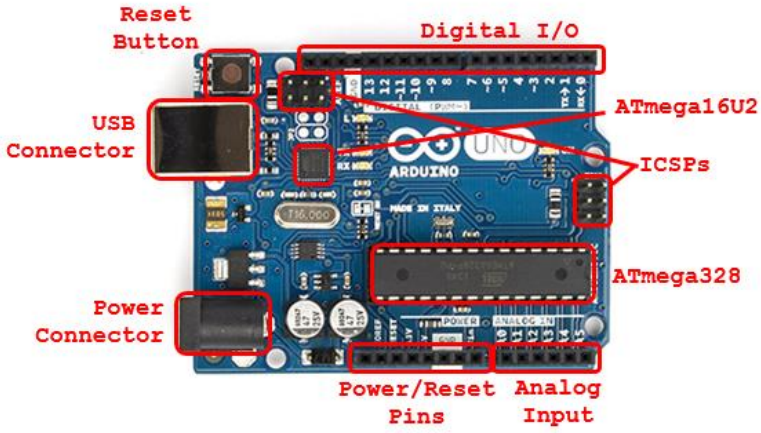
3. **Servo Motor**:
    - **Brown (GND)**: Connects to the Arduino GND pin.
    - **Red (VCC)**: Connects to the Arduino 5V pin.
    - **Orange (Signal)**: Connects to Arduino digital pin 10.
4. **Traffic Light (LEDs)**:
    - **Red LED**:
        - Anode (longer leg) connects to Arduino digital pin 6.
        - Cathode (shorter leg) connects to the Arduino GND through a resistor (typically 220Ω).
    - **Yellow LED**:
        - Anode (longer leg) connects to Arduino digital pin 7.
        - Cathode (shorter leg) connects to the Arduino GND through a resistor (typically 220Ω).
    - **Green LED**:
        - Anode (longer leg) connects to Arduino digital pin 8.
        - Cathode (shorter leg) connects to the Arduino GND through a resistor (typically 220Ω).
5. **Buzzer**:
    - **Positive (longer leg)**: Connects to Arduino digital pin 9.
    - **Negative (shorter leg)**: Connects to the Arduino GND.
6. **LCD Display with I2C Module**:
    - **VCC**: Connects to the Arduino 5V pin.
    - **GND**: Connects to the Arduino GND pin.
    - **SDA**: Connects to the Arduino SDA (A4) pin.

- **SCL**: Connects to the Arduino SCL (A5) pin.

DESCRIPTION OF THE COMPONENTS

1> ARDUINO UNO

The Arduino Uno R3 is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analogue inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.



Functions of the pins are as follows:-

➢Digital pins: Digital pins are for digital input or for digital outputs. Here we connect the led and display. There is total 14 digital pins in Uno r3.

➢ Analog pins: Analog pins are for analog out and analog input. There is total 6 analog pins. Here we connect the IR sensors.

➢ GND: There is total 3 ground pins which provide ground for complete the circuit.

➢ VCC: There are 2 voltage supply is available for use .one is 3.3 v and other is 5 v.3.3 v which is supplied for IR sensors and 5 volts are provided to stepper motor and 16*2 LCD display.

➢ Reset pin: Reset pin are for restart the system.

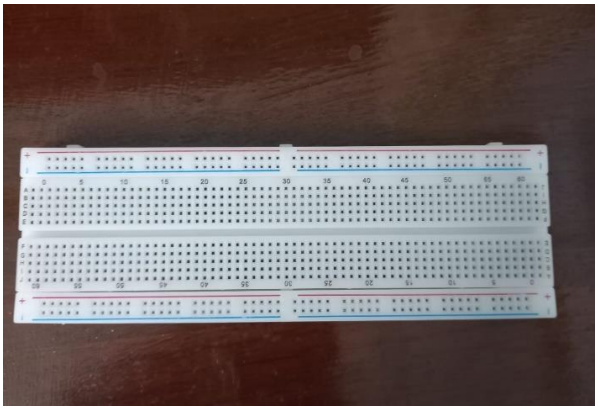
➢ Serial port: There are 2 serial ports. TX and RX.

➢ External power supply: There are an external power supply port to power the Arduino. The supply range is 9v-12v.

➢ USB PLUG: The uno cannot perform without programing. The program is uploaded with this port.

2> BREADBOARD

A breadboard (sometimes called protoboard) is essentially the foundation to construct and prototype electronics. A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure. The grid is made up of perfectly aligned spring clip holes that are 0.1" apart in both the X and Y dimension.



3> ULTRASONIC SENSOR

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. The formula for this calculation is D = ½ T x C (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second).

Ultrasonic sensors use sound waves that travel through air and measure the time needed for them to reach a target object and return back. They then convert this data into electrical signals which can be either analog or digital.

4>SERVO MOTOR

A servo motor is a rotary actuator that allows for precise control of angular position. It consists of a motor coupled to a sensor for position feedback. It also requires a servo drive to complete the system. The drive uses the feedback sensor to precisely control the rotary position of the motor.

It has three pins that helps to control the motor to rotate. They are VCC , GND and control signal wire. The VCC wire is generally red in color , GND is generally black or brown in color and the control signal wire is generally orange or yellow in color.

We give command input according to the position of shaft. If the feedback signal differs from the given input, an error signal alerts the user. We amplify this error signal and apply as the input to the motor; hence the motor rotates. And when the shaft reaches to the require position, error signal become zero, and hence the motor stays standstill holding the position. The command input is in form of electrical pulses. As the actual input to the motor is the difference between feedback signal (current position) and required signal, hence speed of the motor is proportional to the difference between the current position and required position.

The amount of power require by the motor is proportional to the distance it needs to travel. Usually, a servomotor turns 90 degrees in either direction hence maximum movement can be 180 degrees. However, a normal servo motor cannot rotate any further to a build in mechanical stop.

Two servo-motor are used in our project . One in ENTRY gate and another in EXIT gate. And the displacement of servo motor is 90 degrees. The motor is operating by Arduino Uno r3 basics of Ultrasonic sensor values.

5> 16×2 I2C MODULE LCD DISPLAY

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines.

In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

There are 16 pins in lcd display ----

1. Ground
2. VCC
3. Vee
4. Resistor selects
5. Read/write
6. Enable
7. DB0
8. DB1
9. DB2
10. DB3
11. DB4
12. DB5
13. DB6
14. DB7
15. Led+
16. Led

6>CONNECTING WIRES

These are male to male jumper wires used in connecting the male header pin of any development board to other development boards having a male connector. They are simple wires that have connector pins at each end allowing them to be used to connect two points to each other.

Male to female jumper wires are used for easy and convenient interconnection between components in electronic prototyping and testing projects, without the need for soldering. They come in groups or cables with connectors or pins at each end and can connect FRC pins, Header pins, Berg pins, and other components.

Description of the Connections:

1. **Ultrasonic Sensor:**
    - The sensor detects the presence of a vehicle. The trigger pin sends out a sound wave, and the echo pin receives the wave after it bounces off an object. The Arduino measures the time it takes for the sound wave to return and calculates the distance.
2. **Servo Motor:**
    - The servo motor controls the barrier. It rotates to open or close the barrier based on the signals received from the Arduino.
3. **Traffic Light:**
    - The LEDs (red, yellow, green) simulate a traffic light system. Each LED is controlled by the Arduino to indicate the status of the parking system (e.g., green for go, yellow for wait, red for stop).
4. **Buzzer:**
    - The buzzer provides audible alerts. It sounds when the parking lot is full to inform drivers.
5. **LCD Display with I2C Module:**
    - The LCD displays messages such as "Welcome! Slots left: X" or "Sorry! No slots are left". The I2C module simplifies the connection by using only two pins (SDA and SCL) for communication.

Circuit Diagram Explanation:

1. **Power and Ground**:
    - All components that require power (5V) are connected to the 5V pin of the Arduino.
    - All ground (GND) connections are tied to the GND pin of the Arduino.
2. **Digital Pins**:
    - The ultrasonic sensor's trigger and echo pins are connected to digital pins 2 and 3, respectively.
    - The LEDs (traffic light) are connected to digital pins 6, 7, and 8 for red, yellow, and green LEDs, respectively.
    - The buzzer is connected to digital pin 9.
    - The servo motor signal wire is connected to digital pin 10.
    - The I2C module for the LCD uses analog pins A4 (SDA) and A5 (SCL).

Functionality:

1. **Initialization**:
    - When the system initializes, the LCD displays a welcome message indicating the number of available slots.
2. **Vehicle Detection**:
    - When a vehicle is detected by the first ultrasonic sensor, the servo motor rotates to open the barrier, the yellow light turns off, and the green light turns on.
    - Once the vehicle passes the second ultrasonic sensor, the barrier closes, the green light turns off, and the yellow light turns back on.
    - The LCD updates to show the remaining number of slots.

3. **Full capacity**:

   - When all slots are filled, red light turns on and the buzzer sounds for a brief period and the LCD displays a “No slots left” message.

   This setup ensures an efficient and automated parking management system that ensures real-time feedback to users and manages traffic flow effectively.

DETAILED DESCRIPTION OF THE PROJECT ALONG WITH THE CONNECTIONS

1. *Arduino Uno: The central microcontroller that controls the entire system.*
2. *Ultrasonic Sensors (HC-SR04): Used to detect the presence of vehicles.*
3. *Servo Motor: Controls the barrier that allows or restricts vehicle access.*
4. *LED Traffic Lights (Red, Yellow, Green): Indicates the status of the system to the drivers.*
5. *Buzzer: Provides audible alerts when the parking lot is full.*
6. *16x2 LCD Display with I2C Module: Displays real-time information about the availability of parking slots.*
7. *Breadboard and Jumper Wires: Used to make connections between the components and the Arduino.*
8. *Power Supply (9V battery): Provides power to the Arduino and other components.*

Here's a detailed explanation of how each component is connected to the Arduino:

1. **Ultrasonic Sensors**:
    - **VCC**: Connects to the Arduino 5V pin.
    - **GND**: Connects to the Arduino GND pin.
    - **Trig (Trigger Pin 1)**: Connects to Arduino digital pin 2.
    - **Echo (Echo Pin 1)**: Connects to Arduino digital pin 3.
    - **Trig (Trigger Pin 2)**: Connects to Arduino digital pin 4.
    - **Echo (Echo Pin 2)**: Connects to Arduino digital pin 5.
2. **Servo Motor**:
    - **Brown (GND)**: Connects to the Arduino GND pin.
    - **Red (VCC)**: Connects to the Arduino 5V pin.
    - **Orange (Signal)**: Connects to Arduino digital pin 10.
3. **LED Traffic Lights**:
    - **Red LED**:
        - Anode (longer leg) connects to Arduino digital pin 6.
        - Cathode (shorter leg) connects to the Arduino GND through a resistor (typically 220Ω).
    - **Yellow LED**:
        - Anode (longer leg) connects to Arduino digital pin 7.
        - Cathode (shorter leg) connects to the Arduino GND through a resistor (typically 220Ω).
    - **Green LED**:
        - Anode (longer leg) connects to Arduino digital pin 8.

- Cathode (shorter leg) connects to the Arduino GND through a resistor (typically 220Ω).

4.**Buzzer**:

- **Positive (longer leg)**: Connects to Arduino digital pin 9.
- **Negative (shorter leg)**: Connects to the Arduino GND.

5.**Display with I2C Module**:

- **VCC**: Connects to the Arduino 5V pin.
- **GND**: Connects to the Arduino GND pin.
- **SDA**: Connects to the Arduino SDA (A4) pin.
- **SCL**: Connects to the Arduino SCL (A5) pin.

WORKING PRINCIPLE

1. **Initialization**:

- When the system starts, the LCD displays a welcome message indicating the number of available slots (e.g., "Welcome! Slots left: 4").

2. **Vehicle Detection and Barrier Operation**:

- The ultrasonic sensors detect the presence of a vehicle at the entrance.
- When a vehicle is detected by the first sensor, the Arduino triggers the servo motor to open the barrier, turning on the green light and turning off the yellow light.
- As the vehicle passes the second sensor, the barrier closes, the green light turns off, and the yellow light turns back on.
- The LCD display updates the number of available slots.

3. **Full Capacity Management**:

- When all slots are filled, the system activates the red light, turns off the yellow and green lights, and sounds the buzzer for a brief period.
- The LCD displays a message indicating that no slots are available.

SOFTWARE CODES

LCD DISPLAY CODE IS AS FOLLOWS :-

```
#include <Wire.h>  // Include the Wire library for I2C communication
#include <LiquidCrystal_I2C.h>  // Include the LiquidCrystal_I2C library

#define LCD_ADDRESS 0x27  // Set the I2C address of your LCD display
#define LCD_COLUMNS 16    // Number of columns in your LCD display
#define LCD_ROWS 2        // Number of rows in your LCD display

// Initialize the LiquidCrystal_I2C library with the I2C address
LiquidCrystal_I2C lcd(LCD_ADDRESS, LCD_COLUMNS, LCD_ROWS);

void setup() {
  // Initialize the LCD display
  lcd.init();

  // Turn on the backlight (if your LCD display has one)
```

```
 lcd.backlight();

  // Clear the LCD display and print a message on the first line
  lcd.clear();
  lcd.setCursor(0, 0); // Set cursor to the first column of the first row
  lcd.print("Hello, World!");
}

void loop() {
  // Your main code here
}
```

Every I2C lcd display module has a particular address. This code helps to generate the address of the the I2C module LCD display.

SMART PARKING SYSTEM code installed in Arduino is a s follows:-

```
#include <Wire.h>
#include <Servo.h>
#include <LiquidCrystal_I2C.h>

#define TRIGGER_PIN1 2
#define ECHO_PIN1 3
#define TRIGGER_PIN2 4
#define ECHO_PIN2 5
#define RED_LED_PIN 6
#define YELLOW_LED_PIN 7
```

```
#define GREEN_LED_PIN 8
#define BUZZER_PIN 9
#define SERVO_PIN 10

#define LCD_ADDRESS 0x27
#define LCD_COLUMNS 16
#define LCD_ROWS 2

Servo barrierServo;
LiquidCrystal_I2C lcd(LCD_ADDRESS, LCD_COLUMNS, LCD_ROWS);

void setup() {
  pinMode(TRIGGER_PIN1, OUTPUT);
  pinMode(ECHO_PIN1, INPUT);
  pinMode(TRIGGER_PIN2, OUTPUT);
  pinMode(ECHO_PIN2, INPUT);
  pinMode(RED_LED_PIN, OUTPUT);
  pinMode(YELLOW_LED_PIN, OUTPUT);
  pinMode(GREEN_LED_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);

  barrierServo.attach(SERVO_PIN);
  barrierServo.write(90); // Initially, set the servo to the closed position

  // Initialize the LCD display
  lcd.init();
  lcd.backlight();
```

```
lcd.clear();
}

void loop() {
  int slotsLeft = 4;

  // Display initial welcome message with slots left
  lcd.setCursor(0, 0);
  lcd.print("Welcome   Buddy! ");
  lcd.setCursor(0, 1);
  lcd.print("Slots left: ");
  lcd.print(slotsLeft);

  while (slotsLeft > 0) {
    digitalWrite(YELLOW_LED_PIN, HIGH); // Yellow light on

    // Measure distance from ultrasonic sensor 1
    long duration1, distance1;
    digitalWrite(TRIGGER_PIN1, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIGGER_PIN1, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGGER_PIN1, LOW);
    duration1 = pulseIn(ECHO_PIN1, HIGH);
    distance1 = duration1 * 0.034 / 2;

    // Check if car is detected by sensor 1
    if (distance1 < 10) {
```

```
// Rotate servo motor to open barrier
    barrierServo.write(0);
    digitalWrite(GREEN_LED_PIN, HIGH); // Green light on
    digitalWrite(YELLOW_LED_PIN, LOW); // Yellow light off
    delay(1000); // Wait for 1 second

    // Measure distance from ultrasonic sensor 2
    long duration2, distance2;
    digitalWrite(TRIGGER_PIN2, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIGGER_PIN2, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGGER_PIN2, LOW);
    duration2 = pulseIn(ECHO_PIN2, HIGH);
    distance2 = duration2 * 0.034 / 2;

    // Check if car is detected by sensor 2
    if (distance2 < 10) {
      // Rotate servo motor to close barrier
      barrierServo.write(90);
      digitalWrite(YELLOW_LED_PIN, HIGH); // Yellow light on
      digitalWrite(GREEN_LED_PIN, LOW); // Green light off
      delay(1000); // Wait for 1 second

      // Update slots left
      slotsLeft--;
      lcd.clear();
      lcd.setCursor(0, 0);
```

```
 lcd.print("Slots left: ");
        lcd.print(slotsLeft);
      }
    }
  }

  // When all slots are filled
  digitalWrite(RED_LED_PIN, HIGH); // Red light on
  digitalWrite(YELLOW_LED_PIN, LOW); // Yellow light off
  digitalWrite(GREEN_LED_PIN, LOW); // Green light off
  digitalWrite(BUZZER_PIN, HIGH); // Buzzer sound on
  delay(1000); // Wait for 1 second
  digitalWrite(RED_LED_PIN, LOW); // Red light off
  digitalWrite(BUZZER_PIN, LOW); // Buzzer sound off

  // Display message for no slots left
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Sorry! No slots");
  lcd.setCursor(0, 1);
  lcd.print("are left");

  // Wait for 5 seconds
  delay(5000);
}
```

Explanation of the code and the functionality is as follows :-

- **Slot Counter**: Initialize the slot counter (`slotsLeft`) to 4.
- **Initial Display**: Clear the LCD and display the initial message showing the number of available slots.

Ultrasonic Sensor 1 (Car Detection at Entry)

- **Yellow Light On**: Turn on the yellow light to indicate the system is ready.
- **Distance Measurement**: Measure the distance using the first ultrasonic sensor.
    - Trigger the sensor: Send a pulse for 10 microseconds.
    - Read the pulse duration from the echo pin.
    - Calculate the distance using the formula `distance = duration * 0.034 / 2`.

Servo Motor and Green Light (Barrier Open)

- **Check Car Detection**: If the distance is less than 10 cm, it means a car is detected.
    - Rotate the servo motor to `0 degrees` to open the barrier.
    - Turn on the green light and turn off the yellow light.

Ultrasonic Sensor 2 (Car Detection at Exit)

- **Distance Measurement**: Measure the distance using the second ultrasonic sensor.
    - Same process as sensor 1.
- **Check Car Detection**: If the distance is less than 10 cm, it means the car has passed the barrier.
    - Rotate the servo motor back to `90 degrees` to close the barrier.
    - Turn on the yellow light and turn off the green light.
    - Wait for 1 second to ensure the barrier is fully closed.

Update Slots and Display

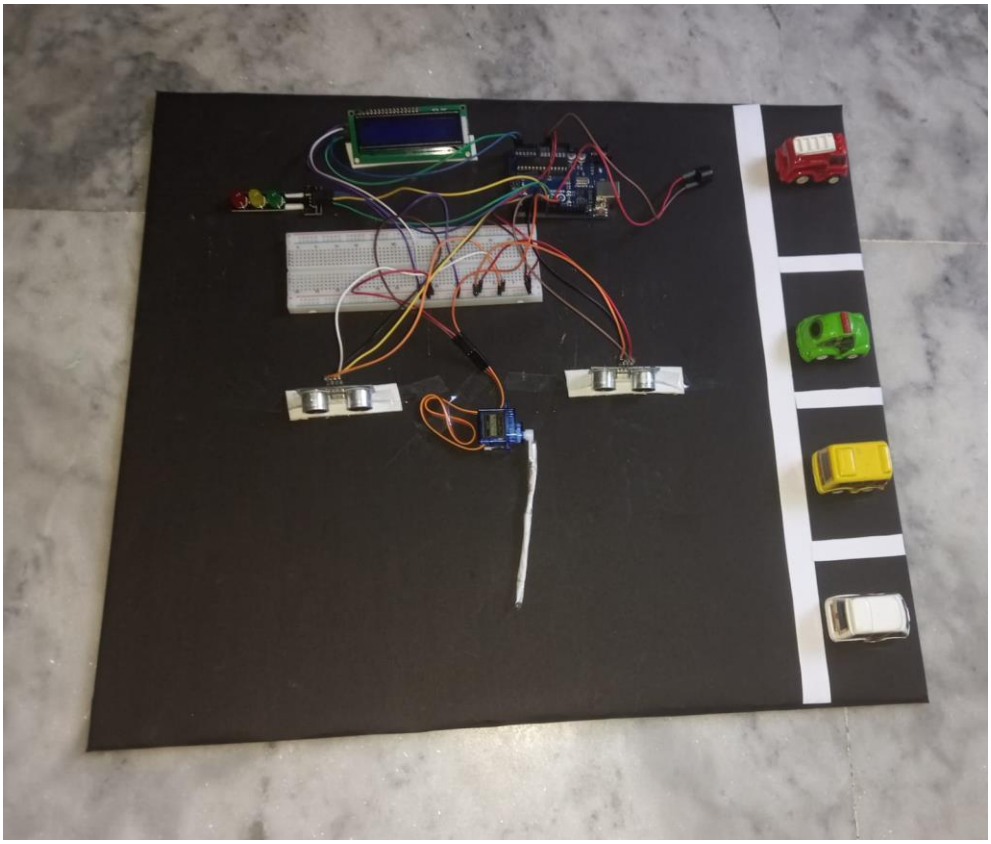
- **Update Slots**: Decrease the slot counter (`slotsLeft`).
- **Update Display**: Clear the LCD and update the number of available slots.

No Slots Available

- **Red Light and Buzzer**: When all slots are filled (`slotsLeft` equals 0):
    - Turn on the red light.
    - Turn off the yellow and green lights.
    - Turn on the buzzer for 1 second.
- **No Slots Message**: Clear the LCD and display the message "Sorry! No slots are left".

- **Wait**: Wait for 5 seconds before checking again.

PROJECT DEMO LINK WITH IMAGE :-



Video Demonstration Link :-

https://youtu.be/9XBMm6LkfTw?si=Z1fdBKgrj8w9Quqp

FUTURE SCOPE :-

1. **Remote Monitoring and Control**:
    - Integrate the system with IoT technology to enable remote monitoring and control of the parking slots via a web interface or mobile app. This would allow users to check slot availability in real-time and reserve slots in advance.
2. **Enhanced Security Features**:
    - Incorporate additional security features such as CCTV integration and automatic number plate recognition (ANPR) to monitor and log vehicle entries and exits. This can enhance the security and management of the parking area.
3. **Energy Efficiency Improvements**:
    - Utilize solar panels to power the system and implement energy-efficient components to reduce the overall power consumption. This can make the system more sustainable and cost-effective in the long run.
4. **Scalability for Larger Facilities**:
    - Expand the system's capability to manage larger parking facilities by incorporating multiple entry and exit points, and using a centralized database to track slot availability across various locations within a facility.
5. **Advanced User Interaction**:
    - Develop advanced user interaction features such as voice control, automated payment systems, and integration with smart city infrastructure. This can enhance user convenience and improve the overall efficiency of urban parking management.

CONCLUSION :-

This project successfully demonstrates an efficient and automated car parking system using an Arduino-based platform. By integrating ultrasonic sensors, a servo motor, an LED traffic light, a buzzer, and an LCD display, we created a system that manages parking slots dynamically and provides clear visual and audio indicators to users.

The system begins by welcoming the user and displaying the number of available slots. As a car approaches, the ultrasonic sensors detect its presence and control the barrier accordingly, ensuring smooth entry and exit. The real-time updates on the LCD display enhance user experience by providing immediate feedback on slot availability.

Throughout the project, various components were interfaced with the Arduino Uno, showcasing the potential of microcontroller-based automation in real-world applications. The system's ability to handle multiple cars efficiently while ensuring proper signaling demonstrates its robustness and reliability.

Overall, this project not only meets the intended objectives but also lays a foundation for future enhancements such as remote monitoring, energy efficiency improvements, and integration with larger parking management systems. It serves as a practical example of how technology can be leveraged to solve everyday challenges, making parking management more effective and user-friendly.

REFERENCES :-

- Arduino. (n.d.). "Arduino - Home." Retrieved from https://www.arduino.cc/.
- Google resources
- Youtube videos
- Chatgpt for some queries which google and youtube failed to answer
- Informations from our guide
- Informations gathered by all of the group members.

THANK YOU