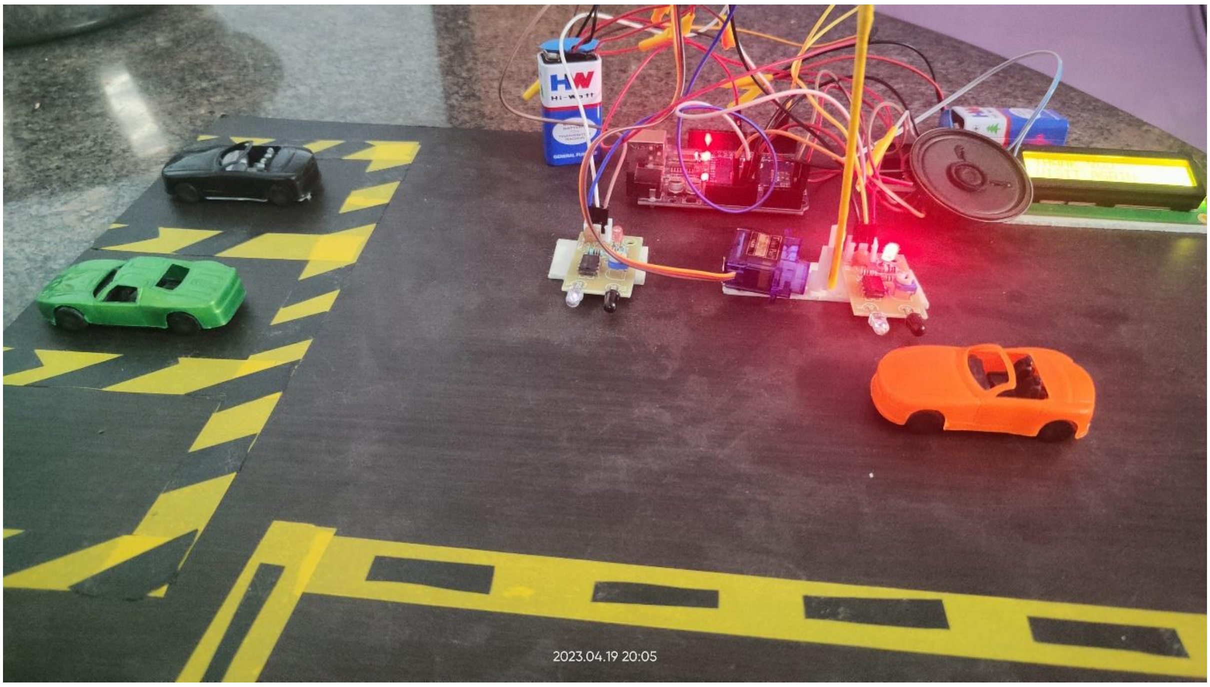
# Smart Car Parking System

## Project Overview

**Smart Car Parking System** is an embedded-system-based solution designed to provide real-time parking slot availability with both visual and voice guidance. The system reduces parking time, improves user convenience, and minimizes congestion by automatically managing entry and exit using sensors and a microcontroller.



## Problem Statement

In conventional parking systems, drivers waste time searching for available slots and often miss visual indicators in crowded or noisy environments. Existing systems mainly rely on display-only feedback, which is not always effective.

## Proposed Solution

This project introduces a **voice-assisted smart parking system** using Arduino. It detects vehicle entry and exit, updates slot availability in real time, and informs drivers through both an LCD display and audio announcements.

## System Architecture

### Main Components:

- Arduino UNO (ATmega328)
- IR Sensors (Entry & Exit detection)
- Servo Motor (Automatic gate control)
- 16×2 I2C LCD Display (Slot availability)
- PAM 8403 Audio Amplifier
- Speaker (Voice guidance)

**Working Principle:**

- IR sensor at entry detects incoming vehicles and triggers gate opening.
- Available slots are updated on the LCD.
- Voice messages announce slot availability.
- Exit IR sensor updates slot count when a vehicle leaves.

---

**Key Features**

- Real-time parking slot availability
- Automatic gate opening and closing
- Voice-based parking assistance
- Improved accessibility for drivers who miss visual cues
- Low-cost and energy-efficient embedded design

---

**Technologies Used**

- **Hardware:** Arduino UNO, IR Sensors, Servo Motor, LCD, PAM 8403 Amp
- **Programming:** Embedded C (Arduino IDE)
- **Domain:** Embedded Systems, IoT basics

---

**Results**

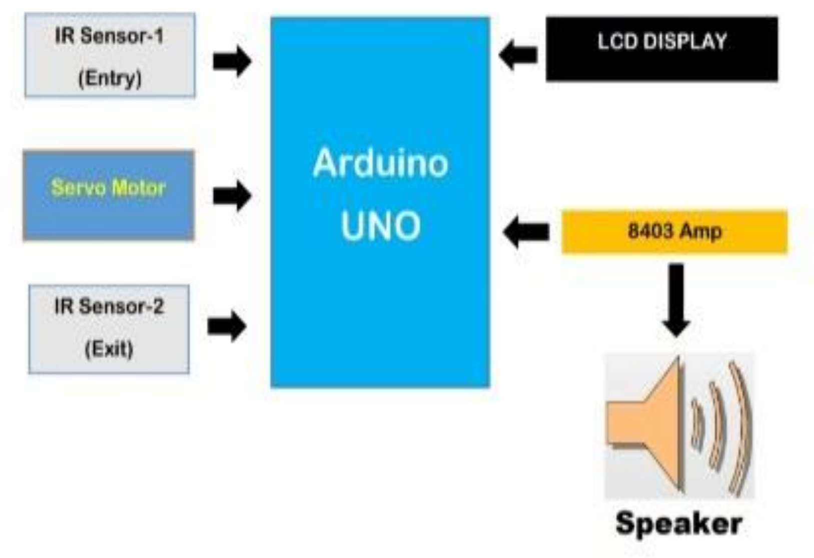
The system successfully provides accurate slot availability with audio guidance. Compared to traditional display-only systems, this approach improves usability and driver awareness, especially in busy parking areas.

---

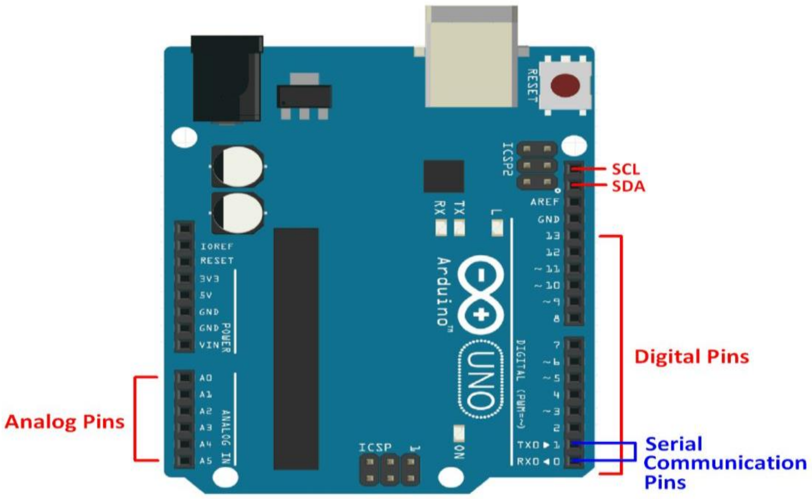
### Future Enhancements

- License plate recognition for security
- Mobile app integration for slot booking
- Solar-powered system for energy efficiency
- IoT-based cloud monitoring

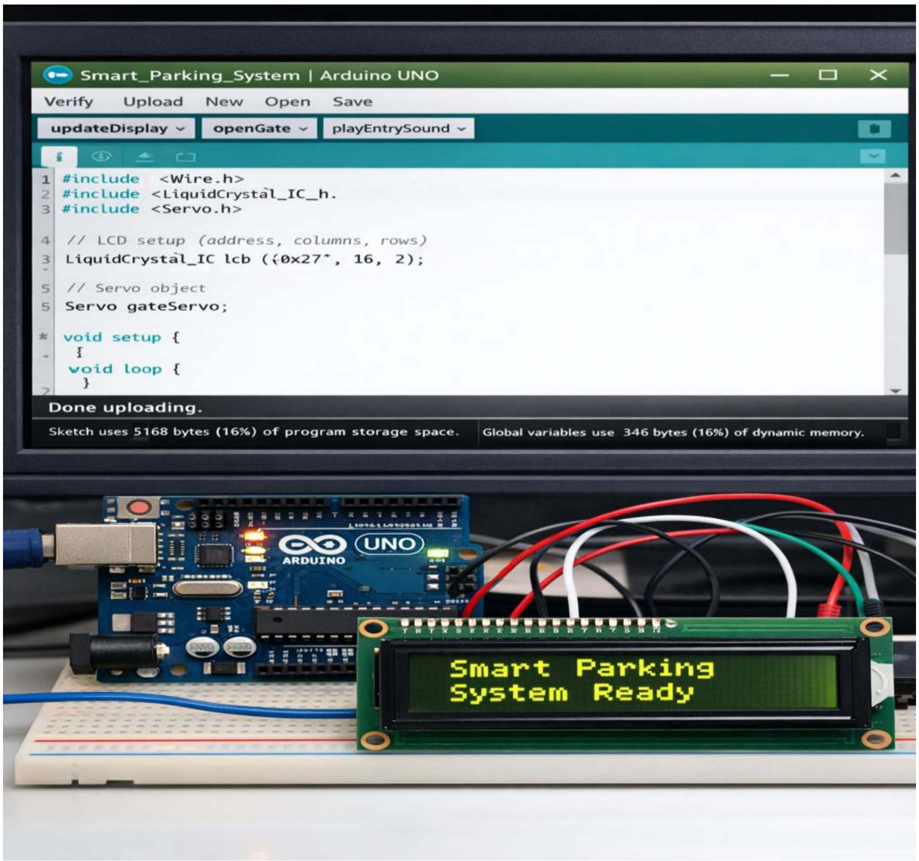
## Block Diagram



## Arduino UNO



# Embedded C for Arduino UNO Setup



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

 // LCD setup (address, columns, rows)
 LiquidCrystal_IC lcb ((0x27", 16, 2);

 // Servo object
 Servo gateServo;

void setup {
 {
 void loop {
  }
```

Done uploading.

Sketch uses 5168 bytes (16%) of program storage space. Global variables use 346 bytes (16%) of dynamic memory.

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

// LCD setup (address, columns, rows)
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Servo object
Servo gateServo;

// Pin definitions
const int entryIR = 2;
const int exitIR  = 3;
const int servoPin = 9;
const int buzzerPin = 8;   // speaker / buzzer

// Parking slots
int totalSlots = 4;
int availableSlots = 4;

// IR sensor state
int entryState;
int exitState;
```

```
void setup() {
 // Pin modes
 pinMode(entryIR, INPUT);
 pinMode(exitIR, INPUT);
 pinMode(buzzerPin, OUTPUT);

 // Servo
 gateServo.attach(servoPin);
 gateServo.write(0);   // gate closed

 // LCD
 lcd.init();
 lcd.backlight();

 // Welcome message
 lcd.setCursor(0, 0);
 lcd.print("Smart Parking");
 lcd.setCursor(0, 1);
 lcd.print("System Ready");

 delay(2000);
 lcd.clear();
 updateDisplay();
}

void loop() {

 entryState = digitalRead(entryIR);
 exitState  = digitalRead(exitIR);

 // -------- ENTRY --------
 if (entryState == LOW && availableSlots > 0) {
  availableSlots--;

  openGate();
  playEntrySound();

  updateDisplay();
  delay(2000);   // debounce delay
 }

 // -------- EXIT --------
 if (exitState == LOW && availableSlots < totalSlots) {
  availableSlots++;

  openGate();
```

```
    playExitSound();

    updateDisplay();
    delay(2000);   // debounce delay
  }

  // -------- PARKING FULL --------
  if (availableSlots == 0) {
    lcd.setCursor(0, 1);
    lcd.print("Parking Full   ");
  }
}

// ---------- FUNCTIONS ----------

void updateDisplay() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Slots Left:");
  lcd.setCursor(12, 0);
  lcd.print(availableSlots);

  lcd.setCursor(0, 1);
  if (availableSlots > 0) {
    lcd.print("Gate Open      ");
  } else {
    lcd.print("No Slot        ");
  }
}

void openGate() {
  gateServo.write(90);   // open
  delay(1500);
  gateServo.write(0);    // close
}

void playEntrySound() {
  tone(buzzerPin, 1000);
  delay(300);
  noTone(buzzerPin);
}

void playExitSound() {
  tone(buzzerPin, 1500);
  delay(300);
  noTone(buzzerPin);
}
```