

```

#include <Servo.h>           // Library to control the servo motor (for the barrier)
#include <Wire.h>            // Library for I2C communication
#include <LiquidCrystal_I2C.h> // Library for LCD display

LiquidCrystal_I2C lcd(0x27,16,2); // Set the LCD address to 0x27 for a 16x2 display

#define irIn 3              // Define pin for entry IR sensor
#define irOut 4             // Define pin for exit IR sensor
#define slot1 8             // Define pin for parking slot 1 IR sensor
#define slot2 7             // Define pin for parking slot 2 IR sensor
#define slot3 6             // Define pin for parking slot 3 IR sensor
#define slot4 5             // Define pin for parking slot 4 IR sensor

Servo s1;                  // Create a Servo object to control the barrier

int outFlag = 0;           // Flag to monitor if a car is leaving (not used here but reserved for
                             future logic)

void setup() {
  s1.attach(9);             // Attach servo to pin 9
  pinMode(irIn, INPUT);     // Set entry IR sensor pin as input
  pinMode(irOut, INPUT);    // Set exit IR sensor pin as input
  pinMode(slot1, INPUT);    // Set each slot sensor pin as input
  pinMode(slot2, INPUT);
  pinMode(slot3, INPUT);
  pinMode(slot4, INPUT);

  s1.write(0);              // Initialize the barrier in the closed position (0 degrees)

  lcd.init();               // Initialize the LCD
  lcd.backlight();          // Turn on the LCD backlight
  lcd.setCursor(3, 0);
  lcd.print("Automated");   // Display initial message
  lcd.setCursor(4, 1);
  lcd.print("Parking");
  delay(2000);              // Delay to let users read the message
  lcd.clear();
  lcd.setCursor(3, 0);
  lcd.print("Ready...");
  delay(1000);              // Indicate system is ready
  lcd.clear();
}

void loop() {
  int inVal = digitalRead(irIn); // Read the entry IR sensor
  int outVal = digitalRead(irOut); // Read the exit IR sensor

  // If a car is detected at the entry and slots are available

```

```

if (inVal == 0 && slot != 0) {
    s1.write(50);          // Open the barrier to 50 degrees
    slot = slot - 1;        // Decrease available slots count by 1
    delay(3000);           // Wait for the car to pass
    s1.write(0);           // Close the barrier
}
// If a car is detected at the exit
else if (outVal == 0) {
    s1.write(50);          // Open the barrier to 50 degrees
    slot = slot + 1;        // Increase available slots count by 1
    delay(3000);           // Wait for the car to pass
    s1.write(0);           // Close the barrier
}

lcd.clear();              // Clear the LCD for fresh display

// Display available slots
lcd.setCursor(1, 0);
lcd.print("Slots Avl");
lcd.setCursor(1, 1);

// Check each slot sensor and display the slot number if it's available
if (digitalRead(slot1) == HIGH) {
    lcd.print("S1,");
}
if (digitalRead(slot2) == HIGH) {
    lcd.print(" S2,");
}
if (digitalRead(slot3) == HIGH) {
    lcd.print(" S3,");
}
if (digitalRead(slot4) == HIGH) {
    lcd.print(" S4");
}

// If all slot sensors detect cars (LOW means occupied), display "All Slots Full"
if (digitalRead(slot1) == LOW && digitalRead(slot2) == LOW && digitalRead(slot3) == LOW &&
digitalRead(slot4) == LOW) {
    lcd.clear();
    lcd.print("All Slots Full");
}

delay(500);              // Delay to update the LCD every half a second
}

```