GOVERNMENT COLLEGE OF ENGINEERING, ERODE

அரசினர் பொறியியல் கல்லூரி, ஈரோடு
Government College of Engineering, Erode
(Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai)

B.E Electronics and Communication Engineering

SMART PARKING SYSTEM

Done By

TEAM LEADER: HARIKUMAR S	731121106305
TEAM MEMBER: SADAI ESWARAN M	731121106039
TEAM MEMBER: MEYYARASAN B	731121106307
TEAM MEMBER: SIVABALAN R	731121106308

Under the mentor of **Dr.. .M. .SATHYAKALA**

Department of Information Technology (IT)

Department of Electronics and Communication Engineering.

Government College of Engineering

Erode ,PO ,near Vasavi College,TamilNadu-638316, Affiliated to Anna University ,Chennai.

Introduction

In these modern days finding car parking is a big issue in congested cities. There are too many vehicles on the road but not enough parking spaces. One of the biggest problems is when we enter a parking area then we realize that there are no empty parking slots to park our cars. Another biggest problem is after entering in a big parking area we confused to find the empty parking slot to park our car. Sometimes maybe we all have been facing these two problems that wasted our important time. That's why we need efficient parking management systems in all parking areas that will provide confusion-free and easy parking.

In this "Smart Parking System Project" to overcome this problem. This project helps the car's driver to park their car with minimum wastage of time with accurate information of the availability of the space to park.

Smart Parking System Project Concept:

This smart parking system project consists of

- Arduino
- IR sensor
- One servo motor
- One LCD display

Where the Arduino is the main microcontroller that controls the whole system. Two IR sensors are used at the entry and exit gates to detect vehicle entry and exit in the parking areas. And other four IR sensors are used to detect the parking slot availability. The servo motor is placed at the entry and exit gate that is used to open and close the gates. Also, an LCD display is placed at the entrance, which is used to show the availability of parking slots in the parking areas.

When a vehicle arrives at the gate of the parking area, the display continuously shows the number of empty slots. If there have any empty slots then the system opens the entry gate by the servo motor. After entering the car into the parking area, when it will occupy a slot, then the display shows this slot is full.

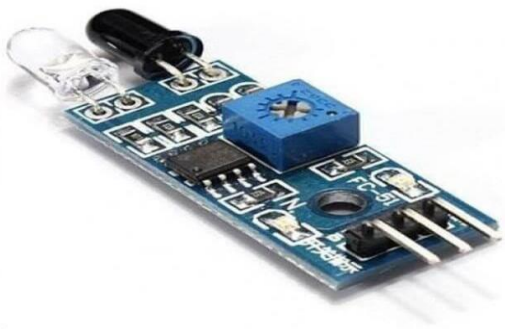
Components in this project

- Arduino Uno
- IR Sensor
- Sg90 Servo Motor
- I2C LCD (LiquidCrystal_I2C.h)

Arduino uno:

The heart of the project is an Arduino board (e.g., Arduino Uno or similar) which serves as the microcontroller to control and co-ordinate all other components.

IR Sensor:



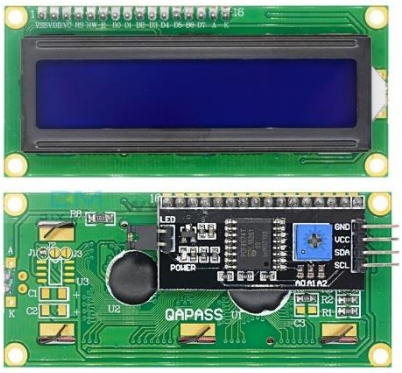
IR sensors are used for detecting the presence of cars in parking slots. In this project, there are four IR sensors (ir_car1, ir_car2, ir_car3, and ir_car4) for monitoring individual parking slots, and two additional IR sensors (IR enter and IR back) for detecting cars entering and leaving the parking areas.

Sg90 Servo Motor:

A servo motor (connected to pin D9 in this sketch) is used to simulate the gate/barrier in the parking system. It can be positioned at different angles to represent an open or closed gate.

I2C LCD:

Liquid Crystal Display (LCD) with I2C Interface A 16x2 character LCD is used to display information about the parking system's status. An I2C interface is employed to simplify the connection and control of the LCD.



PROGRAM:

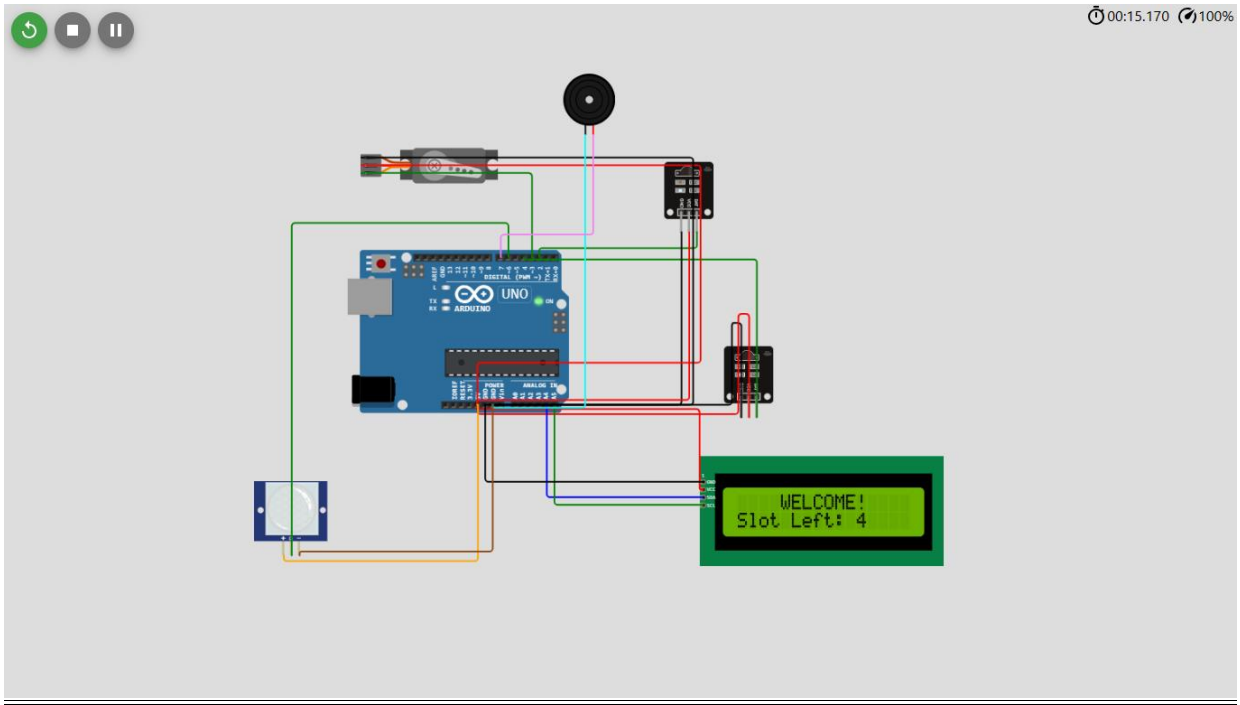
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Change the HEX address

#include <Servo.h>
Servo myservo1;

int IR1 = 2;
int IR2 = 4;
```

```
int SmokeDetectorPin = 6;  // Digital pin for the smoke detector
int BuzzerPin = 7;         // Digital pin for the buzzer

int Slot = 4;  // Enter Total number of parking Slots

bool flag1 = false;
bool flag2 = false;

unsigned long lastLcdUpdate = 0;  // Variable to track the time of the last LCD update
unsigned long lcdUpdateInterval = 1000;  // Update the LCD every 1000 milliseconds (1 second)

void setup() {
  lcd.begin(16, 2);  // Initialize LCD with 16 columns and 2 rows
  lcd.backlight();
  pinMode(IR1, INPUT);
  pinMode(IR2, INPUT);
  pinMode(SmokeDetectorPin, INPUT);
  pinMode(BuzzerPin, OUTPUT);

  myservo1.attach(3);
  myservo1.write(100);

  lcd.setCursor(0, 0);
  lcd.print("    ARDUINO    ");
  lcd.setCursor(0, 1);
  lcd.print(" PARKING SYSTEM ");
  delay(2000);
  lcd.clear();

  Serial.begin(9600);  // Start serial communication for debugging
}

void loop() {
  if (digitalRead(IR1) == LOW && !flag1) {
    if (Slot > 0) {
      flag1 = true;
      if (!flag2) {
        myservo1.write(0);
        Slot--;
      }
    } else {
```

```
      displayMessage("    SORRY :(    ", "  Parking Full  ");
    }
  }

  if (digitalRead(IR2) == LOW && !flag2) {
    flag2 = true;
    if (!flag1) {
      myservo1.write(0);
      Slot++;
    }
  }

  if (flag1 && flag2) {
    delay(1000);
    myservo1.write(100);
    Serial.println("Servo returned to initial position.");
    flag1 = false;
    flag2 = false;
  }

  // Update the LCD display with a delay
  if (millis() - lastLcdUpdate >= lcdUpdateInterval) {
    updateLcdDisplay();
    lastLcdUpdate = millis();
  }

  // ... (Rest of your code)
}

void updateLcdDisplay() {
  if (digitalRead(SmokeDetectorPin) == HIGH) {
    displayMessage("   WARNING!   ", " Smoke Detected ");
    digitalWrite(BuzzerPin, HIGH);  // Turn on the buzzer
  } else {
    displayMessage("    WELCOME!    ", "Slot Left: " + String(Slot));
    digitalWrite(BuzzerPin, LOW);   // Turn off the buzzer
  }
}

void displayMessage(const char *line1, const String &line2) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(line1);
```

```
  lcd.setCursor(0, 1);
  lcd.print(line2);
}
```

Output:



PRORAM EXPLANATION:

This Arduino program is for a parking management system with features like IR sensor-based car detection, smoke detection, and an LCD display for status updates. It monitors parking slots, moves a servo arm to block/unblock slots, and displays warnings if smoke is detected. Let's break down the key components and functions in this program:

1. *Libraries and Components Setup*:

- The program includes two libraries: Wire.h for I2C communication and LiquidCrystal_I2C.h for controlling an I2C-enabled LCD screen.
- It also includes the Servo.h library for controlling a servo motor.
- An instance of the LiquidCrystal_I2C class is created to control the LCD screen and is initialized with the I2C address, number of columns, and rows.
- Various digital pins are defined for IR sensors, a smoke detector, and a buzzer. Additionally, a servo motor is initialized.

2. *Global Variables*:

- Variables like Slot, flag1, and flag2 are defined to keep track of the available parking slots and the status of the IR sensors.
- lastLcdUpdate and lcdUpdateInterval are used for updating the LCD display at regular intervals.

3. *Setup Function*:

- The setup function initializes the LCD, sets up pins as inputs/outputs, attaches the servo motor, and displays an initial message on the LCD.
- Serial communication is initiated for debugging purposes using Serial.begin.

4. *Loop Function*:

- The loop function is the main control loop that continuously runs.
- It checks the status of IR sensors (IR1 and IR2) to detect the entry and exit of vehicles.

- If a vehicle enters (IR1 is LOW), it checks if there are available parking slots. If so, it lowers the servo arm and decreases the available slots. If no slots are available, it displays a message on the LCD.
- If a vehicle exits (IR2 is LOW), it raises the servo arm and increases the available slots.
- When both a vehicle enters and exits (both flags are set), it waits for a short delay, raises the servo arm, and resets the flags.
- The LCD display is periodically updated by calling the updateLcdDisplay function.

5. **updateLcdDisplay Function**:
- This function updates the LCD display based on the status of the smoke detector. If smoke is detected, it displays a warning message and activates the buzzer. Otherwise, it displays the number of available parking slots.

6. **displayMessage Function**:
- This function clears the LCD screen and displays a message on the specified lines of the screen.

CONCLUSION :

Overall, the program is designed to manage a parking system, controlling entry and exit of vehicles using IR sensors and providing status information on an LCD screen. It also includes a smoke detector with a warning system. The servo motor controls the physical barrier for each parking slot.