**SMART SYSTEM PARKING**

**PYTHON SCRIPT:**

#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);

}