**SMART CAR PARKING**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <Servo.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Set the LCD address and dimensions

Servo gateServo;

const int servoPin = 9; // Define the pin for controlling the servo motor

const int buzzerPin = 8; // Define the pin for the buzzer (optional)

const int irSensor1 = 2; // Define the IR sensor pins for slot 1

const int irSensor2 = 3; // IR sensor pins for slot 2

int availableSlots = 2; // Number of available parking slots

bool parkingFull = false;

void setup() {

gateServo.attach(servoPin);

pinMode(buzzerPin, OUTPUT);

pinMode(irSensor1, INPUT);

pinMode(irSensor2, INPUT);

lcd.init();

lcd.backlight();

lcd.print("Available: 2"); // Initial display

}

void loop() {

if (!parkingFull) {

// Check IR sensors for cars entering parking slots only if parking is not full

if (digitalRead(irSensor1) == LOW) {

// Car entered slot 1

availableSlots--;

openGate();

} else if (digitalRead(irSensor2) == LOW) {

// Car entered slot 2

availableSlots--;

openGate();

}

}

// Check if the parking is full

13

if (availableSlots == 0) {

parkingFull = true;

digitalWrite(buzzerPin, HIGH); // Sound the buzzer (optional)

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Parking Full");

closeGate();

} else {

parkingFull = false;

digitalWrite(buzzerPin, LOW); // Turn off the buzzer

// Update the LCD display with the available slots count

lcd.clear();

lcd.setCursor(0, 1);

lcd.print("Available: " + String(availableSlots));

}

// Check IR sensors for cars exiting parking slots

if (digitalRead(irSensor1) == HIGH) {

// Car exited slot 1

availableSlots++;

}

if (digitalRead(irSensor2) == HIGH) {

// Car exited slot 2

availableSlots++;

}

}

void openGate() {

// Control the servo motor to open the gate

gateServo.write(90); // Adjust the angle according to your servo

delay(2000); // Adjust the delay according to your gate's speed

}

void closeGate() {

// Control the servo motor to close the gate

gateServo.write(0); // Adjust the angle to close the gate

delay(2000); // Adjust the delay according to your gate's speed

}