**Sindh Madressatul Islam**

**University Karachi**

**COURSE: Artificial Intelligence TERM: SPRING 2024, CLASS: 6D**

**PROJECT NAME: AUTOMATIC CAR PARKING COUNTER**

**Course Instructor: Mr. Sajid Ali**

**Group Members**

|  |  |
| --- | --- |
| **Names** | **Student IDs** |
| Zulqarnain Ahmed | CSC-21F-051 |
| Samad | CSC-21F-057 |
| Munam Aftab | CSC-21F-110 |
| Sahil | CSC-21F-111 |
| Suneel Bharthi | CSC-21F-121 |

## ABSTRACT:

This project report illustrates the configuration of a parking sensor that help the driver while enter the car in parking. Sometimes it can be difficult to judge and search for the place to park when you enter the parking. You will get the info while entering in parking that the pace for parking has remain or not. The sensor which is at parking gate increment the number when any car enters in parking and show that on screen how many cars are in parking. When any car is left from parking this screen shows the decrement value from the existing value of the number of cars.

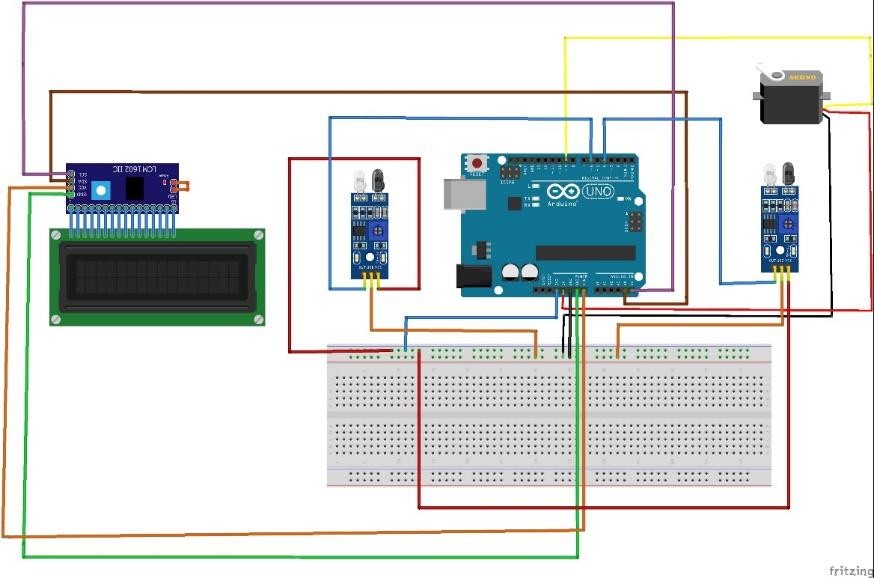
### INTRODUCTION & PROBLEM:

The Car Parking Management System aims to increase efficiency and save time in managing parking spaces. It provides real-time information on parking availability, reducing traffic issues caused by illegal parking. The system utilizes microcontrollers, infrared technology, and displays to simplify the parking process. Implementing this system helps reduce congestion in cinemas, multiplexes, industries, and commercial areas.

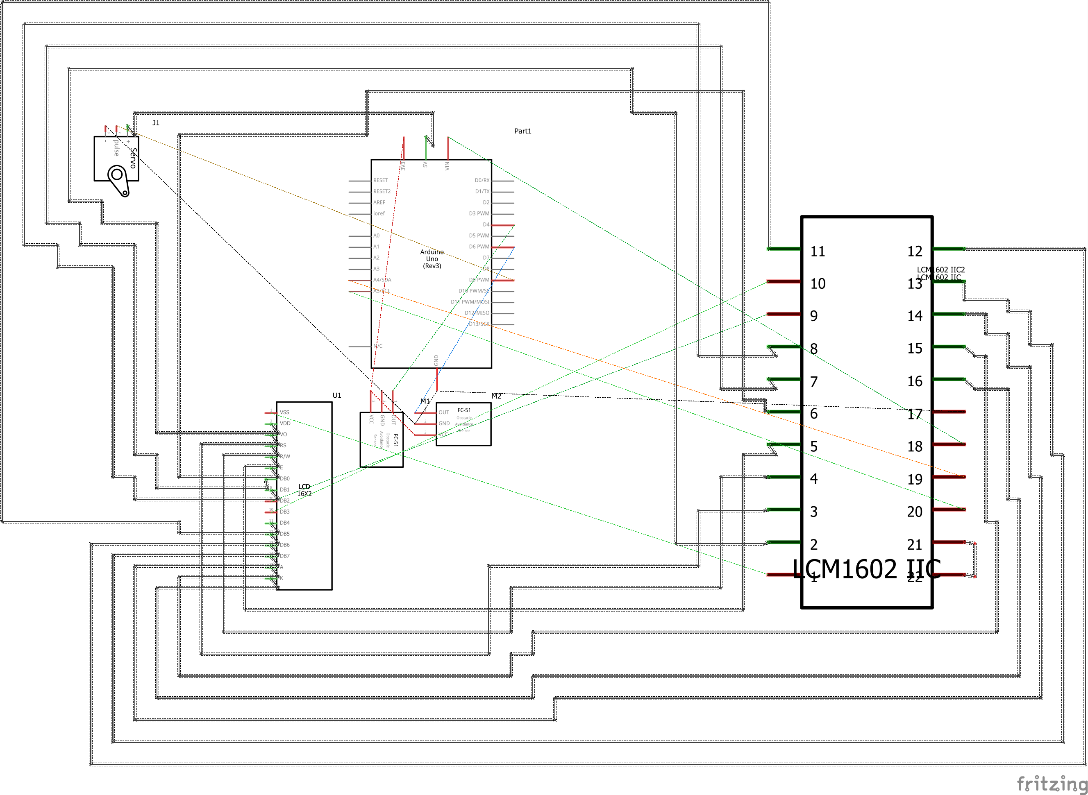
### COMPONENTS DETAILS AND COST ESTIMATION:

|  |  |
| --- | --- |
| **Name of Components & Quantity** | **Cost** |
| * Solderless Breadboard (1Qty) | Rs 300 |
| * Arduino Uno (1Qty) | Rs 2300 |
| * IR Sensor (2Qty) | Rs 300 |
| * I2c module 16×2 LCD Display (1Qty) | Rs 600 |
| * Jumper Wires (1Qty) | Rs 300 |
| * Servo SG-90 (1Qty) | Rs 300 |
| * Stationary (Glue,Cardboard,Tape etc ) (1Qty | Rs 900 |
| **Total Cost** | **Rs 5000** |

1. **CIRCUIT DIAGRA**



## BLOCK DIAGRAM:



1. **Hardware Specifications:**
   * **ARDUINO UNO:**

The Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (6 of which can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, power jack, ICSP header, and a reset button. It provides everything needed to support the microcontroller, making it easy to connect to a computer or power source. The Uno board was named to commemorate the release of Arduino Software (IDE) 1.0 and is the reference model for the Arduino platform.



*Figure 1: ARDUINO UNO*

# LCD Display:

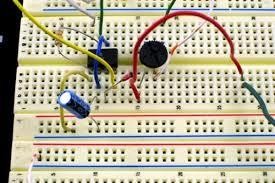
LCD (Liquid Cystal Display) is a thin, energy-efficient flat panel display that uses liquid crystals to block or allow light, while LED 7-segment displays are either common cathode (CC) or common anode (CA), with CC having connected cathodes and CA having connected anodes.

*Figure 2: LCD Display*

# Bread board:

**The specifications & features of a breadboard include the following.**

* Distribution Strips are two.
* Wire Size is 21 to 26 AWG wire.
* Tie Points are two hundred.
* Withstanding Voltage is 1,000V AC.
* Tie points within IC are 630.

The purpose of a breadboard is to facilitate quick electrical connections between components for testing circuits before permanent soldering. It has interconnected sockets for easy component placement.

*Figure 3: bred boa*rd

# IR Sensor

IR sensors are optoelectronic components that detect infrared radiation within a specific wavelength range. They are commonly used in motion detectors and alarm systems. Line sensors, such as the TCRT5000, utilize infrared light emission and detection to detect the presence of a black line.

*Figure 4: IR Sensor*

### Maximum distance of IR sensor:

Most IR thermometers have a maximum measuring distance of approximately **100 feet (30 meters)** in real life, depending on environmental conditions.

# Jumper Wires:

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used

with [breadboards](https://blog.sparkfuneducation.com/what-is-a-breadboard) and other prototyping tools in order to make it easy to change a circuit as

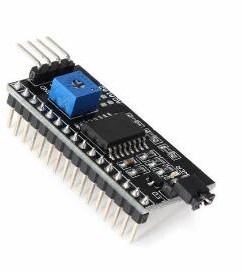
needed. Fairly simple. In fact, it doesn’t get much more basic than jumper wire.



# I2c Module:

*Figure 5: Jumper Wires*

I2C is a communication protocol that allows devices to exchange data using only two wires: a data line and a clock line. It simplifies wiring and is commonly used for connecting components in embedded systems.



*Figure 6: I2c Module*

# Connection of i2c with LCD Display



To connect an I2C LCD to an Arduino, you only need to connect four pins. Connect VCC to 5V,

GND to ground, of bread board SDA (data line) to A4, and SCL (clock line) to A5.

*Figure 6: I2c Module Connection with LCD display*.

## WORKING STEPS:

As Car Enters from gate in parking the IR-Sensors detects it and opens the gate by servo motor and add a count and display it on LCD Display and when the car went through next IR sensor the gate will be closed automatically and add count about the available spaces and total spaces in parking area. And when a car exits the parking lot its detected by IR-Sensor and it subtracts the count and displays it on LCD Display.

1. **CODE FOR ARDUINO:**

// Include the necessary libraries for I2C communication, LCD display, and servo motor control

#include <Wire.h>

#include <LiquidCrystal\_I2C.h> #include <Servo.h>

// Initialize the LCD object with the I2C address and dimensions LiquidCrystal\_I2C lcd(0x27, 16, 2);

// Initialize the servo motor object Servo myservo1;

// Define the IR sensor pin numbers int IR1 = 4;

int IR2 = 6;

// Define the number of available parking slots int Slot = 4;

// Initialize flags for tracking the status of the IR sensors int flag1 = 0;

int flag2 = 0;

// The setup function runs once when the Arduino board is powered on or reset void setup() {

// Initialize the LCD display lcd.begin(16, 2); lcd.backlight();

// Set the IR sensor pins as inputs pinMode(IR1, INPUT);

pinMode(IR2, INPUT);

// Attach the servo motor to pin 9 and set its initial position to 145 degrees myservo1.attach(9);

myservo1.write(145);

// Display a welcome message on the LCD for 3 seconds lcd.setCursor(0, 0);

lcd.print(" Welcome To"); lcd.setCursor(0, 1); lcd.print(" SMIU Parking "); delay(3000);

lcd.clear();

}

// The loop function runs repeatedly after the setup function has completed void loop() {

// Check if IR sensor 1 is triggered and flag1 is not set if (digitalRead(IR1) == LOW && flag1 == 0) {

// Check if there are available parking slots if (Slot > 0) {

flag1 = 1;

// Check if flag2 is not set if (flag2 == 0) {

// Open the servo motor (write 0 degrees) and decrease the available slot count

myservo1.write(0); Slot = Slot - 1;

}

}

else {

// Display a message on the LCD indicating that the parking is full lcd.setCursor(0, 0);

lcd.print(" Sorry"); lcd.setCursor(0, 1); lcd.print(" Parking Full"); delay(2000);

lcd.clear();

}

}

// Check if IR sensor 2 is triggered and flag2 is not set if (digitalRead(IR2) == LOW && flag2 == 0) {

flag2 = 1;

// Check if flag1 is not set if (flag1 == 0) {

// Open the servo motor (write 0 degrees) and increase the available slot

count

myservo1.write(0); Slot = Slot + 1;

}

}

// Check if both flags are set if (flag1 == 1 && flag2 == 1) {

delay(500);

// Close the servo motor (write 145 degrees) and reset the flags myservo1.write(145);

flag1 = 0;

flag2 = 0;

}

// Display a welcome message and the number of available slots on the LCD lcd.setCursor(0, 0);

lcd.print(" Welcome!"); lcd.setCursor(0, 1); lcd.print(" Slot Left: "); lcd.print(Slot);

}

## CONCLUSION:

This project utilizes an Arduino Uno, servo motors, LCD display, and ultrasonic sensors (HC-05) to create an efficient car parking system. It automates vehicle entry and exit, providing real-time information on available parking spaces. The objective is to design a digital controller circuit with logic circuits for 99 parking slots. Sensors at the entrance and exit communicate digital signals to display the number of empty slots and indicate when the parking is full.