LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title
CS5053NI/CC5068NI– Cloud Computing & IoT

"Automated Car Parking System"

Assessment Type

System Development Report

Semester

2023 Spring/Autumn

Group members

London Met ID	Student Name
22085816	Bibek Kumar Thakur
22085498	Apoorva K.C.
22085643	Susmita Thakur
22085504	Aryan Pudasaini

Assignment Due Date: Friday, January 12, 2024

Assignment Submission Date: Friday, January 12, 2024

Submitted To: Mr. Sugat Man Shakya

Word Count: 2907

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Acknowledgment

We deeply appreciate everyone's individual and group contributions that went into making our project, "Automated Car Parking System," a reality. This initiative aims to ease Nepal's urban parking problems.

We sincerely appreciate our lecturer **Mr. Saugat Man Shakya** and our tutor **Mr. Suryansh Mathema** for their invaluable guidance and expertise. Their help greatly influenced our project. We are also thankful our **Islington college** for providing us the required materials.

We also wish to extend our heartfelt gratitude to our peers, mentors, and fellow team members for their steadfast support and encouragement, which significantly propelled the advancement of our project. This accomplishment stands as evidence of the collaboration, commitment, and diligent efforts of everyone involved.

Abstract

Using the potential of Infrared Sensor technology, our project "**Automated Car Parking System**," aims to transform the way urban parking difficulties are managed. Our main objective is to help parking facilities make the most of their available space while also improving the parking experience for drivers. By combining inexpensive infrared sensors with a centralized processing unit, our solution can increase parking facilities' operating efficiency, decrease the need for manpower, and save infrastructure costs.

This invention offers long-term advantages for parking management companies as well as drivers, providing a sustainable solution to parking problems. In addition to providing instantaneous enhancements to parking experiences, our solution lays the foundation for improved operational methods that may boost productivity and economy in the parking sector. This program aims to provide a sustainable foundation for parking management techniques in the future, in addition to addressing current issues.

Table of Contents

Table of figures

Table of Tables

1. Introduction:

The Internet of Things (IoT) is a network where physical objects that are equipped with various sensors and software communicate over the internet alongside exchanging real-time data for a more enhanced process. With the help of IoT, machines can complete laborious jobs without assistance from human (Gillis, 2024).

The Internet of Things (IoT) serves as a network of innovative technology that allows real-time data exchange and communication between devices. Our concept, the "Automated Car Parking System," which integrates IR sensors to transform parking infrastructure, is proof of the promise of the Internet of Things. In addition to automation, this project highlights the significant influence of IoT by improving driver comfort and optimizing space use in cities. Implementing the system is also expected to result in significant staffing cost savings for parking companies, ushering in a new era of effective and user-friendly parking management solutions. This project aims to reinvent the parking experience by offering a cutting-edge strategy that streamlines operations and guarantees a smooth, profitable system for both consumers and companies.

1.1 Current Scenario:

Parking problems has been a huge issue in Nepal since a very long time. The current situation is the result of poor planning, managing and lack of foresight. According to research by The Rising Nepal, the lack of defined parking spaces forces cars to park in crowded roads and alleyways, which annoys pedestrians and exacerbates traffic congestion. In addition to being inconvenient, this condition raises the possibility of accidents, thus parking management policies need to be immediately re-evaluated in order to reduce traffic, improve safety precautions, and maximize urban mobility for the benefit of the community (The Rising Nepal, 2022).

1.2 Problem Statement and Project as a solution:

Since parking issue is a problem that affects everyone, parking stations must be properly available in various locations around the nation's major cities. In addition to the necessary modifications, parking station automation is a duty that is equally vital. This would assist in resolving the parking issue.

A partial solution to the issue of uncontrolled parking would be the thoughtful installation of this “Automated Car Parking System” inside various parking stations. Additionally, it provides a more sensible, secure, and approachable way to deal with parking management issues.

1.3 Aims:

The main aim of this project is to create an IoT system that enhances parking efficiency by automating traffic flow of the parking area.

1.4 Objectives:

- The system is expected to have an automatic opening and closing gate, with the intention of IR sensors and Servomotor.
- The details of the cars shall be displayed on LCD module which acts as an actuator.
- A microcontroller shall be used as controlling unit for the IR sensors, Servomotors, and LCD.
- There shall be proper integration of hardware components using different jumper wires, and the placement of components shall be done in accordance with the real scenario of a parking station.

2. Background:

2.1 System Overview:

This, “Automated Car parking System” System is a working prototype matching with the real-world parking station, and it has the potential to solve the carparking issues in crowded cities if it is implemented in large scale. This system is for automating the traditional parking styles with the implementation of various IoT components together with a microcontroller.

Its main purpose is to use these sensors to identify the presence of vehicles and automate gate or barrier actions upon entry and leave in order to automate parking procedures. Driver may park more easily and efficiently thanks to this automated method, which drastically lowers the need for manual activities in the parking infrastructure. This display, which is integrated with IR sensors and a central controlling unit, shows the availability of parking spaces as of right now. This visual depiction helps drivers to find

available places quickly, cutting down on search time, and improving their overall parking experience. By combining these elements, the system seeks to maximize parking space use and lessen the annoyance that comes with conventional parking systems, offering a smooth and convenient solution for parking managers and drivers alike.

The system's effectiveness depends on all its components working seamlessly. IR sensors are essential since they start automation by managing gates and showing the current parking status. Parking is made easier with this integration, which maximizes space usage. The system's ultimate goal is to solve urban parking issues and automate traditional parking approach and also giving user convenience and operational effectiveness first priority.

2.2 Design Diagrams:

2.2.1 Hardware Architecture:

The hardware components along with their major connections and data flow is shown in architecture diagram. There is car detection by IR sensors and Input data is sent to the Controller. LCD is used as actuator to display the information given by the microcontroller about spaces in parking station. Servomotor is an actuator used as gate to open and close while entering and exiting the cars. Following table shows the overview of the system.

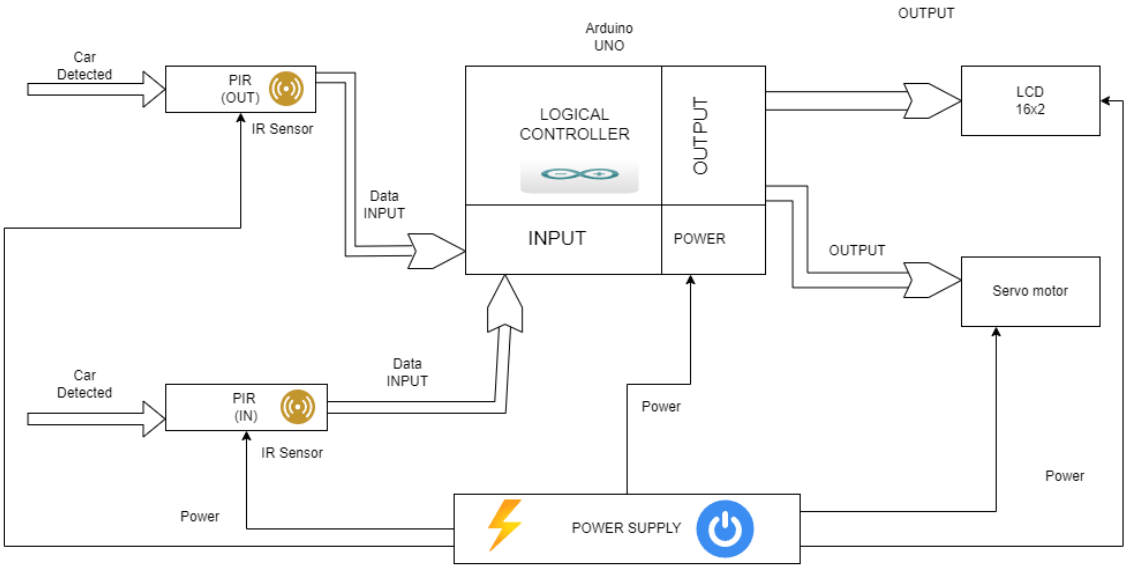


Figure 1: System/Hardware Architecture

2.2.2 Flowchart

The whole system flowchart is given below. Flowchart is the graphical representation of overall system and its functionality. It has all the conditions of in gate and out gate. The source code was written based on the following flowchart in the Arduino IDE.

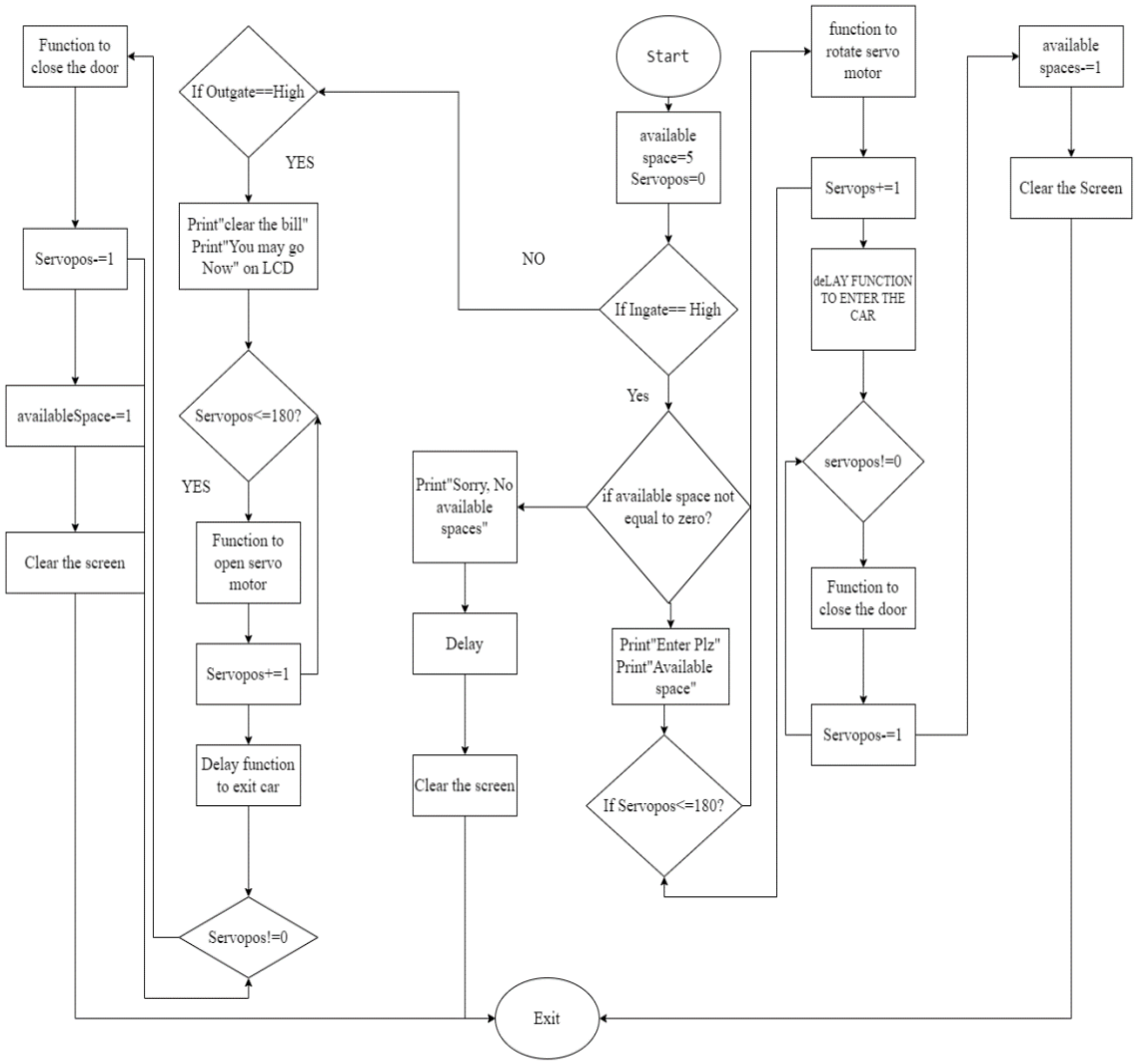


Figure 2: Flowchart

2.2.3 Circuit Diagram

A circuit diagram is a graphical depiction that makes an electrical circuit simpler. It is sometimes referred to as an electrical diagram, elementary diagram, or electronic schematic. It functions as a visual tool for electrical and electronic equipment design, creation, and maintenance (BYJU'S, 2023).

The figure below shows the depiction of the connection among various components of the system. The circuit diagram was made in “Tinker cad” application. The virtual working of the connected system was checked as simulation in that web application.

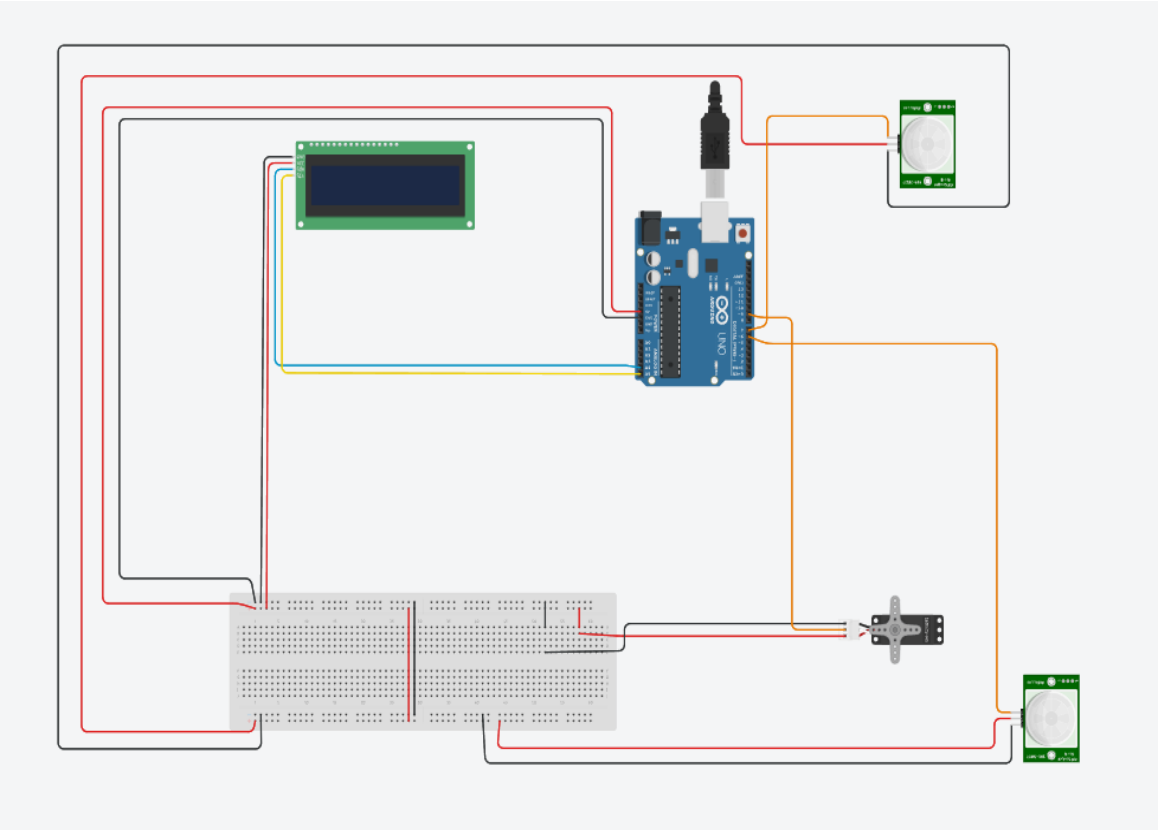


Figure 3: Circuit Diagram

2.3 Requirement Analysis:

2.3.1 Hardware Components:

i. Arduino UNO:

The Arduino Uno is a microcontroller board that operates using the ATmega328P processor. It includes 14 digital I/O pins, 6 analog inputs, a USB connection, a power jack, an ICSP header, and a reset button. It was used for coordination and control of various hardware components within the system (Flyrobo, 2023).

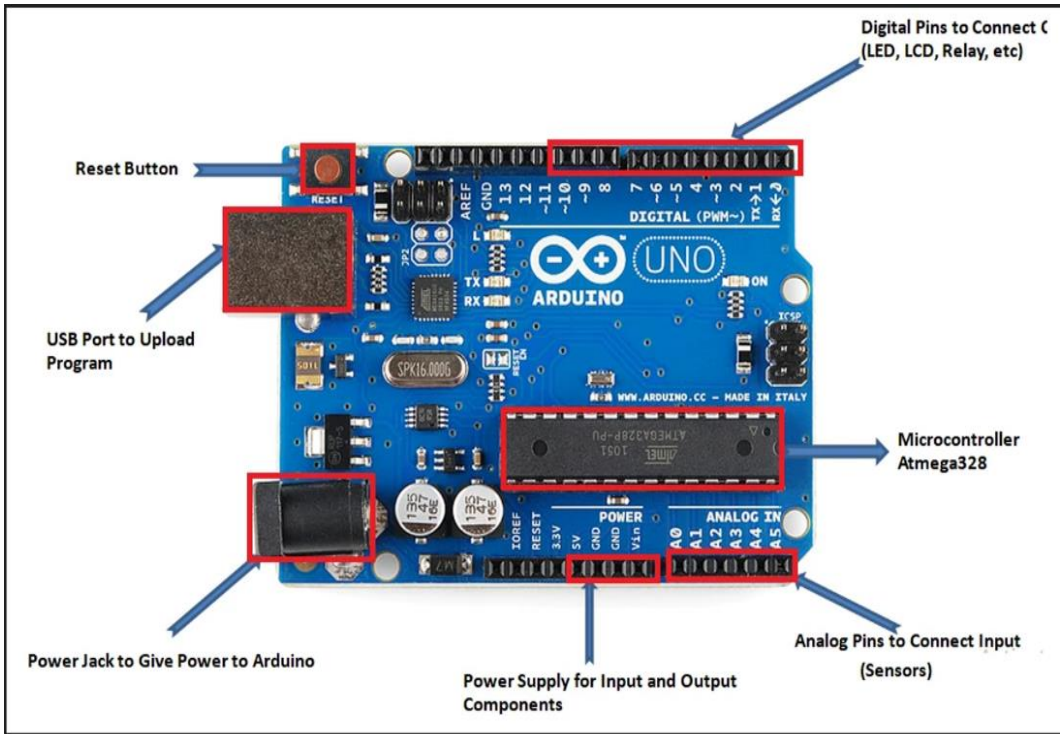


Figure 4: Arduino UNO

ii. Breadboard:

Bread Board is a plastic block holding a matrix of electrical sockets of a size suitable for gripping thin connecting wire, component wires or the pins of transistors and integrated circuits. A breadboard was used to facilitate the easy connection and testing of various electronic components, enabling rapid assembly and modification of the system's circuit (Probots, 2024).

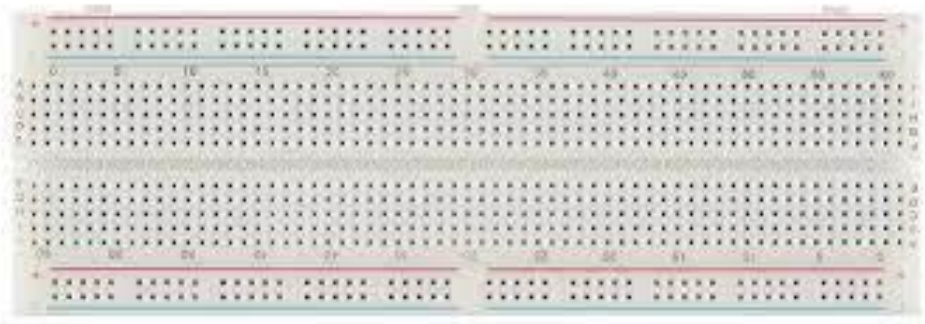
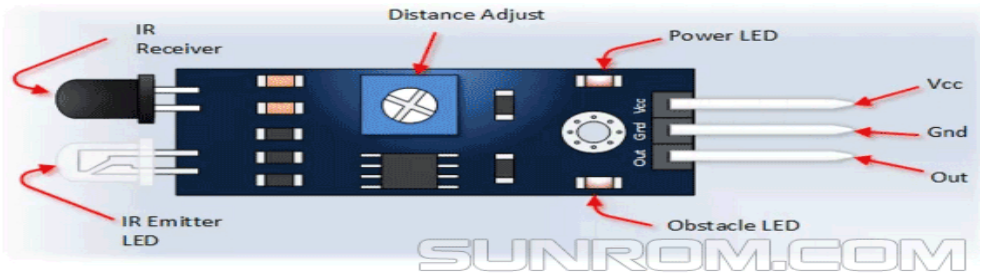


Figure 5: Breadboard

iii. Infrared Proximity Sensor (PIR):

Proximity IR sensor is used to detect the objects or obstacles within the small range. The sensor continuously emits infrared light, and it detects objects approaching it by keeping an eye on the reflected light from the object. It can be used in robots for obstacle avoidance, for automatic doors, for parking aid devices or for security alarm systems (Sunrom, 2024).



Pin, Control Indicator	Description
Vcc	3.3 to 5 Vdc Supply Input
Gnd	Ground Input
Out	Output that goes low when obstacle is in range
Power LED	Illuminates when power is applied
Obstacle LED	Illuminates when obstacle is detected
Distance Adjust	Adjust detection distance. CCW decreases distance. CW increases distance.
IR Emitter	Infrared emitter LED
IR Receiver	Infrared receiver that receives signal transmitted by Infrared emitter.

Figure 6: Proximity IR sensor

iv. Liquid Crystal Display (LCD 16 x 2) with I2C module:

The following image shows the LCD with I2C module integrated. It can display 16x2 characters on 2 lines, white characters on blue background.

This I2C 16x2 Arduino LCD Screen is using an I2C communication interface. It means it only needs 4 pins for the LCD display: VCC, GND, SDA, SCL. It will save at least 4 digital/analog pins on Arduino (DFROBOT, 2024).

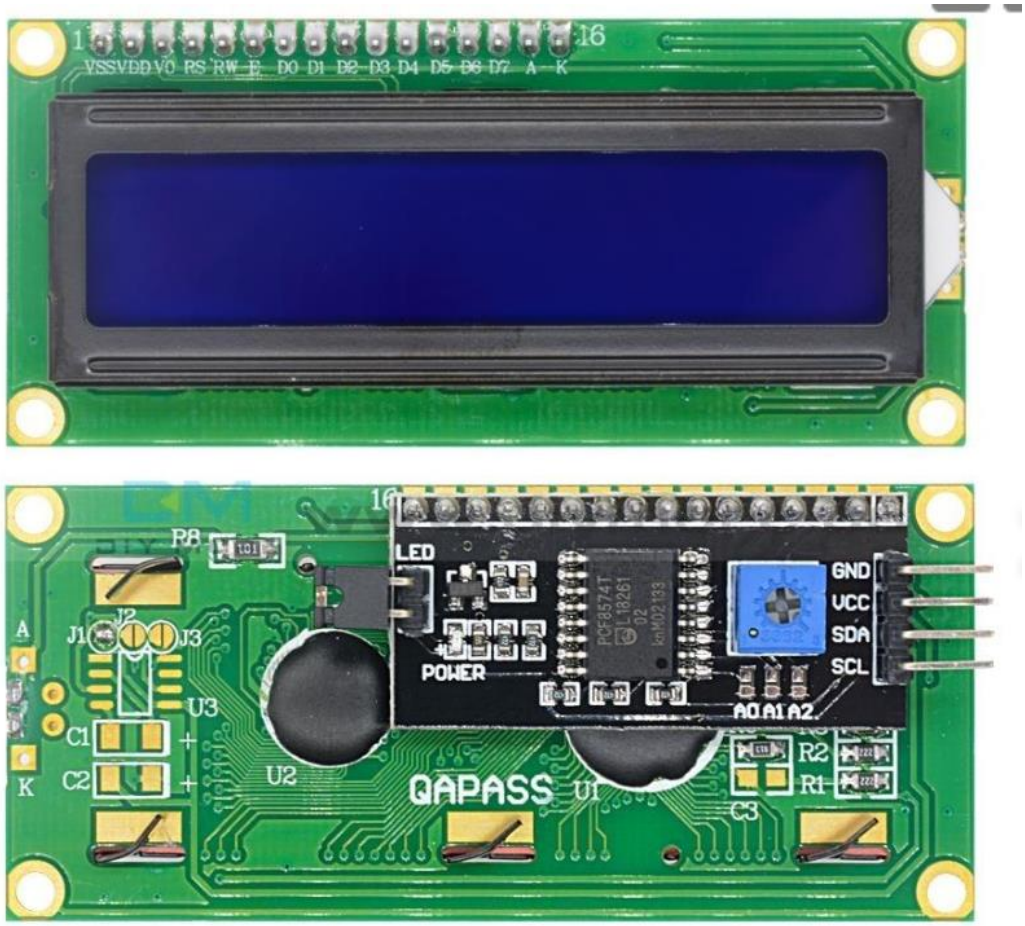


Figure 7: I2C 16x2 Arduino LCD

v. Mini Servo Motor:

A mini servo motor is a small electromechanical device used in robotics and electronics to control and adjust the position of objects (Rozum, 2024).

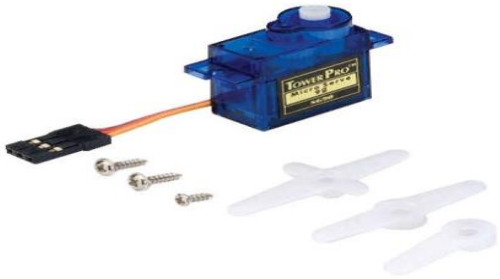


Figure 8: Servo motor

vi. Jumper Wires:

Jumpers are tiny connectors used to close or open a circuit part. They have two or more connection points, which regulate an electrical circuit board. They are used to connect the components with breadboard and Arduino.

These jumper wires also help to prevent the hectic connections and soldering problems (Wiltronics Research Pty Ltd , 2022).

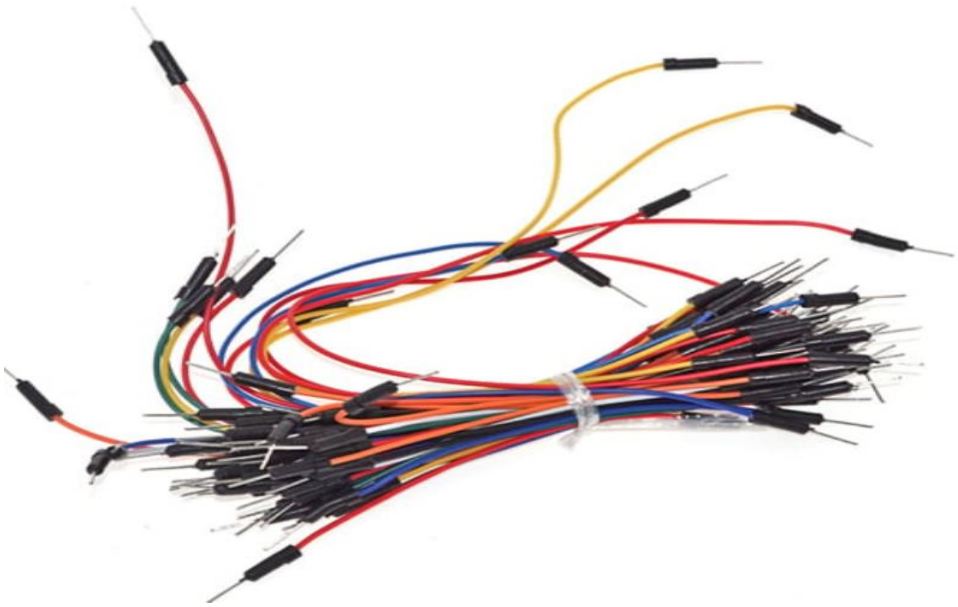


Figure 9: Jumper wires

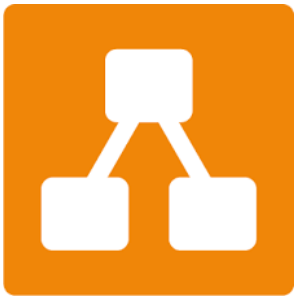
2.3.2 Software Used:

- **Arduino IDE**

  An essential software platform for programming Arduino microcontroller boards, such as the Uno, is the Arduino IDE. It makes creating, compiling, and uploading code easier allows you to manage different hardware parts of the Automated Car Parking System. Its user-friendly interface was used in this project to enable effective code generation and maintenance, which made it easier to coordinate sensors, motors, and displays for automatic parking procedures and real-time updates on available space (Arduino, 2023).

- **Draw.io**

  Draw.io is a popular diagramming tool well-known for its adaptability and versatility. It's user-friendly interface and large library made simplifying diagram creation, visualizing, and planning a lot more convenient. Using this software; flowcharts, System Architecture diagram, Block Diagram were made (Draw.io, 2023).



- **Tinker cad:**

  Tinkercad is a free web app for 3D design, electronics, and coding. The virtual simulation of the system was done on this web app. Circuit diagram and Schematics diagram were also created from this web application (Autodesk, Inc, 2024).

- **Ms- Word**

  Microsoft word is a word processor which is widely used for various documentation purposes. It offers a full range of tools for documentation, editing, formatting, and sharing. This word processor was used for documentation of the proposal and system development report for the “Automated Car Parking System” project (Geeksforgeeks, 2024).

3. Development

This development part includes following steps of development of the IOT project. The overall project from its planning to the actual working with components and development of the system is as follows.

3.1 Design & Planning

First of all, the research was conducted on the selection of the project by all the members. After an hour of looking at the various IoT based projects online and a lot of discussion, we came to the conclusion of choosing a system that would automate the traditional parking style of the car stations. The name, “Automated Car parking System” was given to the project.

The rough sketch of the system architecture of our prototype was created. After some research, about the IoT components we figured out the required hardware for our project. The major components were Arduino UNO, IR sensors, Mini Servo Motor, and LCD screen for displaying output. We learnt about building the circuits on Tinkercad and using the connected system as virtual simulation. Then we planned to create the circuit diagram of whole system using that web application. The simulation was done after connecting the components virtually and coding inside the tinkercad. We approached our tutor, Mr. Suryansh Mathema for his checking and corrections. After simulation we went to the further steps.

3.2 Resource Collection

We requested to the Resource Department, Islington College by writing the application for our required hardware components. From resource department, we gathered the following components:

- Arduino UNO
- Jumper wires
- Bread Board
- Mini servo motor
- I2C 16x2 LCD Module etc.

We brought the Proximity IR sensors, sample cars and others from market. All the components were collected and placed in a small box. Then, we proceeded to next steps of developing the system.

3.3 System Development

Step 1: Arduino IDE installation and Arduino Checking:

In this step, Arduino IDE software was installed on PC, and we make sure that its latest version and it has no bugs. After the successful installation, we connected the Arduino UNO with PC to check whether it will work or not. The IDE detected Arduino and lights were emitting from the indicator of Arduino which confirms the working condition of the component successfully.

Figure 10: Arduino with PC

Step 2: Proximity IR Sensor checking.

We checked the proximity IR sensor which is one of the essential components of the system. We simply used it to operate a LED bulb to check the working of the IR sensor. At first the relevant code to operate LED with proximity sensor was written in the IDE and it operated successfully as shown in the figure below:

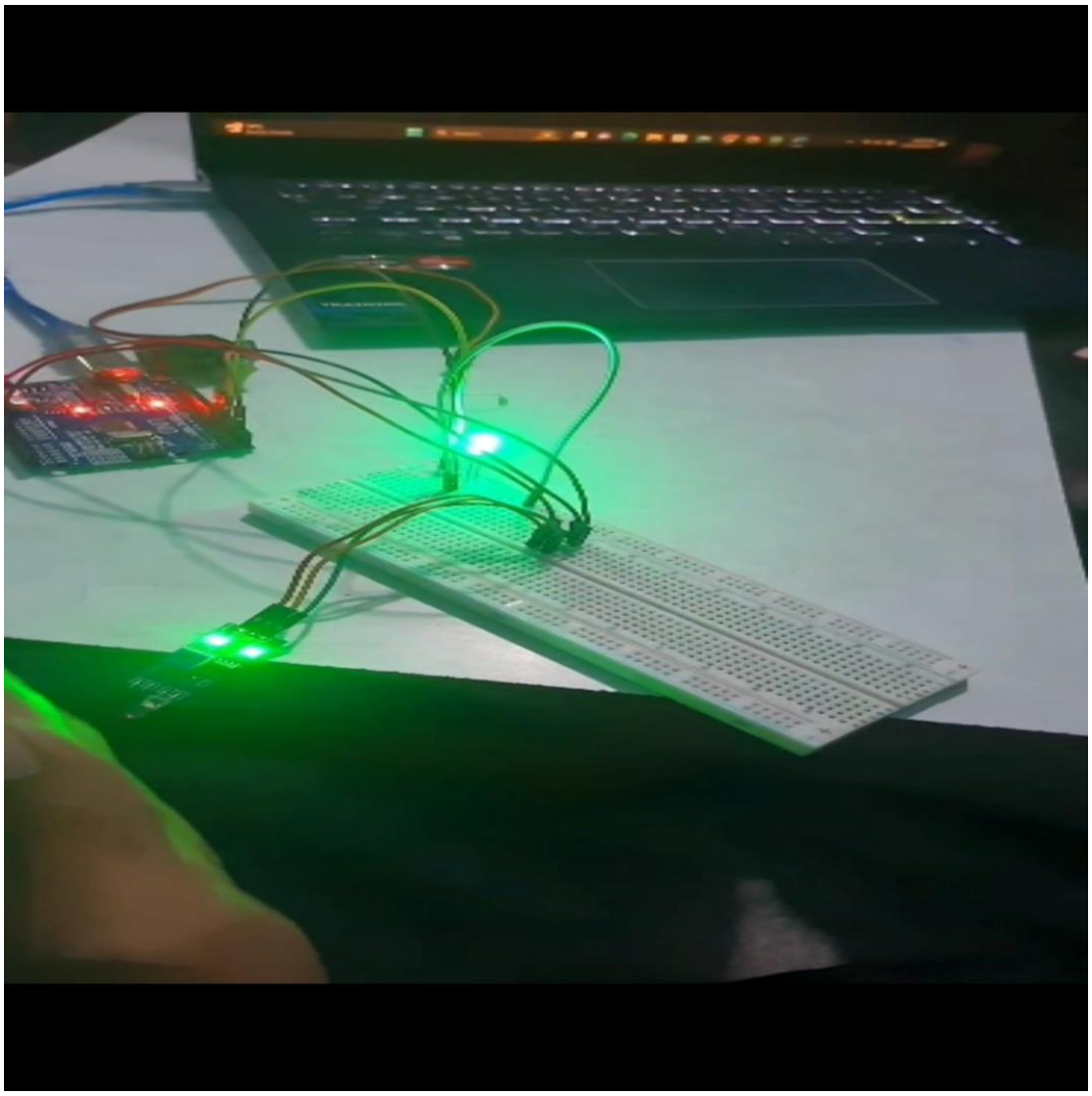


Figure 11: Checking the IR sensor.

Step 3: Checking the I2C LCD module.

LCD 16x2 with I2C module integrated was first checked before the actual integration in the system of the "**Automated Car Parking System**". The code for LCD was done in IDE. LCD was attached to Arduino UNO along with the relevant connections as shown in the following figure.

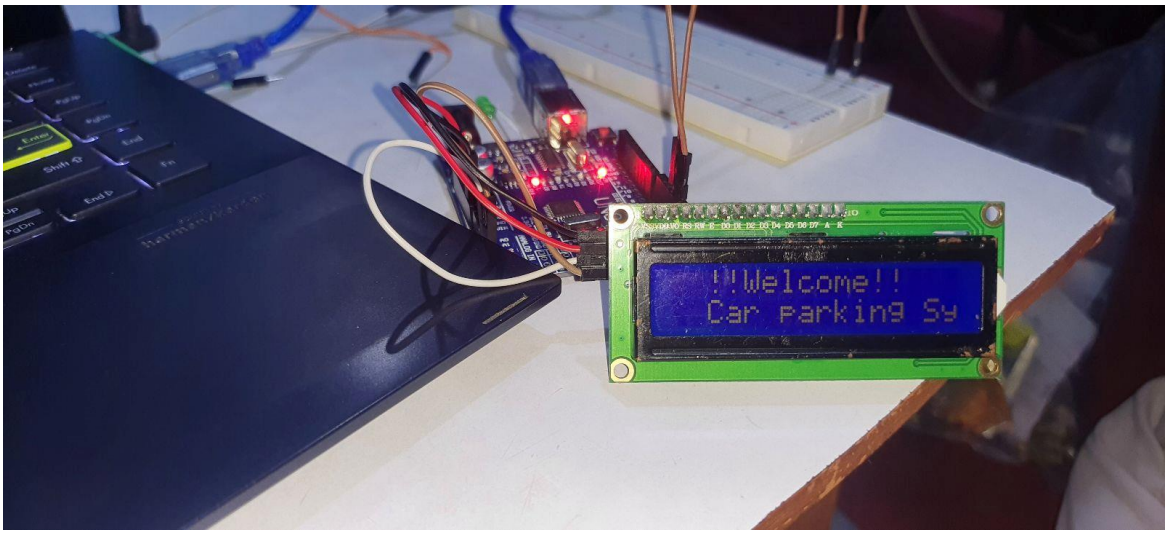


Figure 12: LCD checking

Step 4: Connecting all the components to develop the system.

The hardware components were connected, and code was done in accordance. The system was working as shown below.

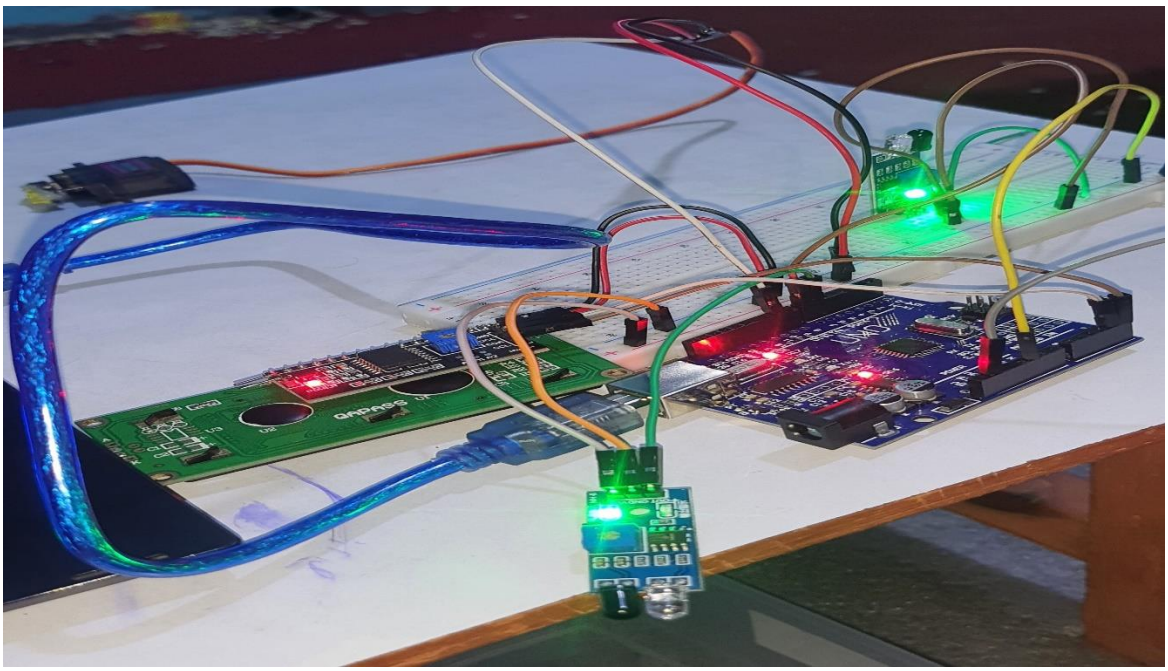


Figure 13: Entire System Connection

3.4 Pins Connection to other Components

Table 1: Pins connection

Components	Components Pins	Arduino Pins
Proximity IR sensor 1 (In gate)	VCC GND OUTPUT	+5V 0 Pin 6
Proximity IR sensor 2 (out gate)	VCC GND OUTPUT	+5V 0 Pin 7
I2C LCD 16x2 Screen	VCC GND SDA SCL	+5V 0 A4 A5
Mini Servo Motor	VCC GND Signal Or PWM	+5V 0 Pin 9

3.5 Writing Program to run the project.

Code of the system.

```
#include<Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

LiquidCrystal_I2C lcd(32, 16, 2);//set screen to the correct value
Servo myServo; // object creation
#define inGate 6 // Ir1
#define outGate 7 // Ir2
#define servoPin 9
int availableSpaces=5;//avalaible space for car parking

int servoPos = 0;//initialiase servo position

void setup() {
  lcd.init();
  lcd.clear();
  lcd.backlight();
  lcd.display();
  lcd.setCursor(0,0);
  pinMode(inGate, INPUT);
  pinMode(outGate, INPUT);
  myServo.attach(servoPin);
```

```
  Serial.begin(9600);


}

void loop() {

  if (digitalRead(inGate) == HIGH) {
    Serial.print(digitalRead(inGate));
    delay(25);
    Serial.flush();
    lcd.setCursor(0, 0);
    if (availableSpaces != 0) {
      lcd.print("Enter Plz!");
      lcd.setCursor(0, 1);
      lcd.print("Avai_spaces: ");
      lcd.print(availableSpaces);
      for (servoPos = 0; servoPos <= 180; servoPos += 1) {
        myServo.write(constrain(servoPos, 0, 180));
        delay(10);
      }
      delay(1000); // Adjust delay time as needed
      for (servoPos = 180; servoPos >= 0; servoPos -= 1) {
        myServo.write(constrain(servoPos, 0, 180));
        delay(15);
      }
    }

    availableSpaces -= 1;
    delay(1000);// Adjust delay time as needed
    lcd.clear();
  } else {
    if (digitalRead(outGate) == HIGH) {
        Serial.print(digitalRead(outGate));

    delay(25);
    Serial.flush();
      lcd.setCursor(0, 0);
      lcd.print("Clear Bills plz!");
      lcd.setCursor(0, 1);
      lcd.print("You May Go now!");
      for (servoPos = 0; servoPos <= 180; servoPos += 1) {
        myServo.write(constrain(servoPos, 0, 180));
        delay(15);
      }
      delay(1000); // Adjust delay time as needed
```

```
    for (servoPos = 180; servoPos >= 0; servoPos -= 1) {
      myServo.write(constrain(servoPos, 0, 180));
      delay(15);
    }
    availableSpaces += 1;
    delay(1000); // Adjust delay time as needed
    lcd.clear();
  }}


}
```

4. Results and Findings

After the development phase of the system, the system was ready, and various tests were conducted to check the results and findings from the system.

Various Tests Performed:

Test-1: Test if the IR sensor placed at (In) entry gate detects objects or not.

Table 2: Test-1

Objective:	To test the working of IR sensor of (in) gate of parking.
Activity:	Arduino connected to the laptop and running the code.
Expected Result:	IR sensor should detect the objects and display the result on LCD.
Actual Result:	The LCD screen shows the working of IR sensor placed at entry gate.
Conclusion:	The test is successful.

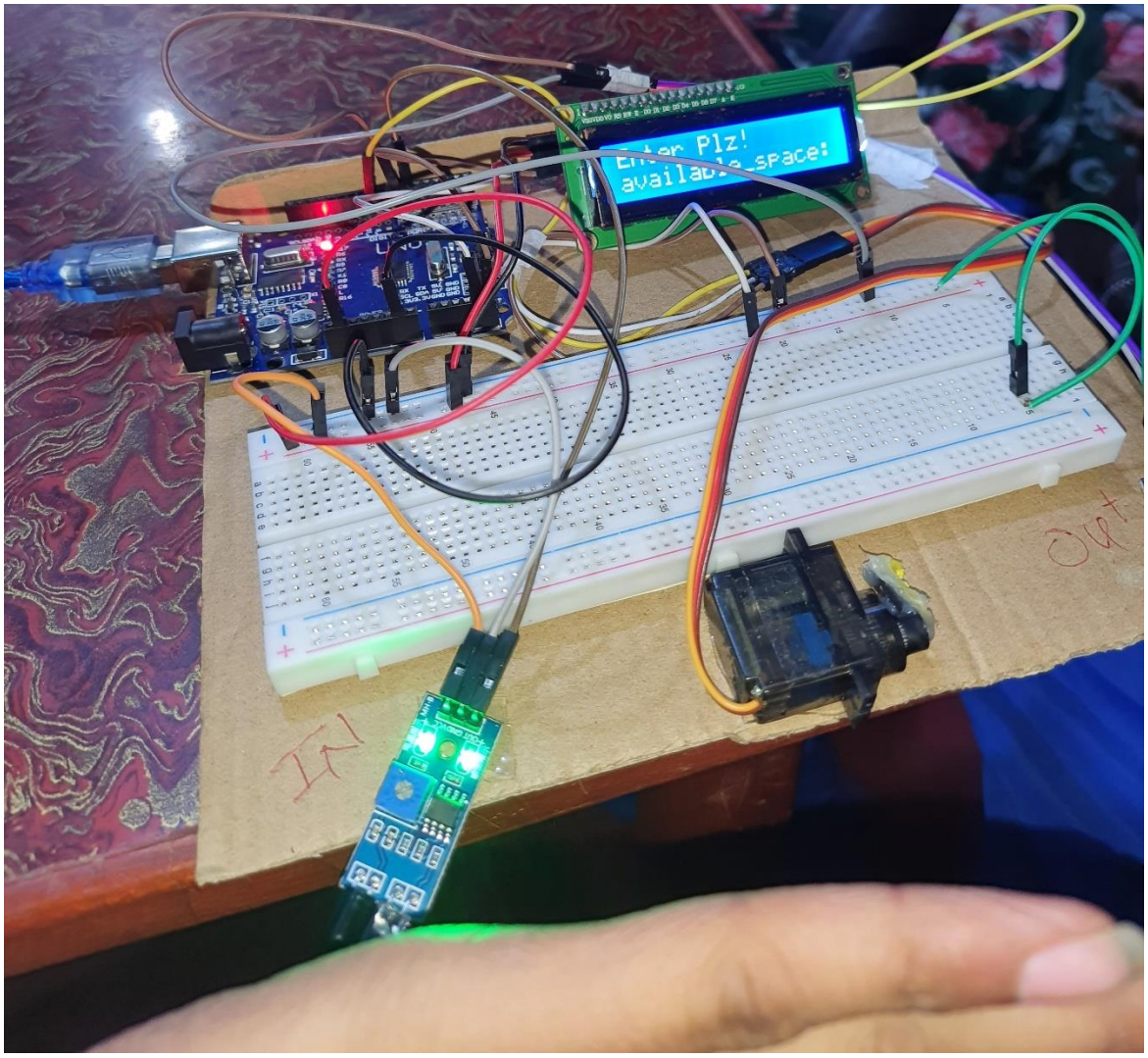


Figure 14: Entry Gate IR sensor working.

Test-2: Test if the IR sensor placed at (out) exit gate detects objects or not.

Table 3: Test-2

Objective:	To test the working of IR sensor of (out) gate of parking.
Activity:	Arduino connected to the laptop and running the code.
Expected Result:	IR sensor should detect the objects and display the result on LCD.
Actual Result:	The LCD screen shows the working of IR sensor placed at entry gate.
Conclusion:	The test is successful.

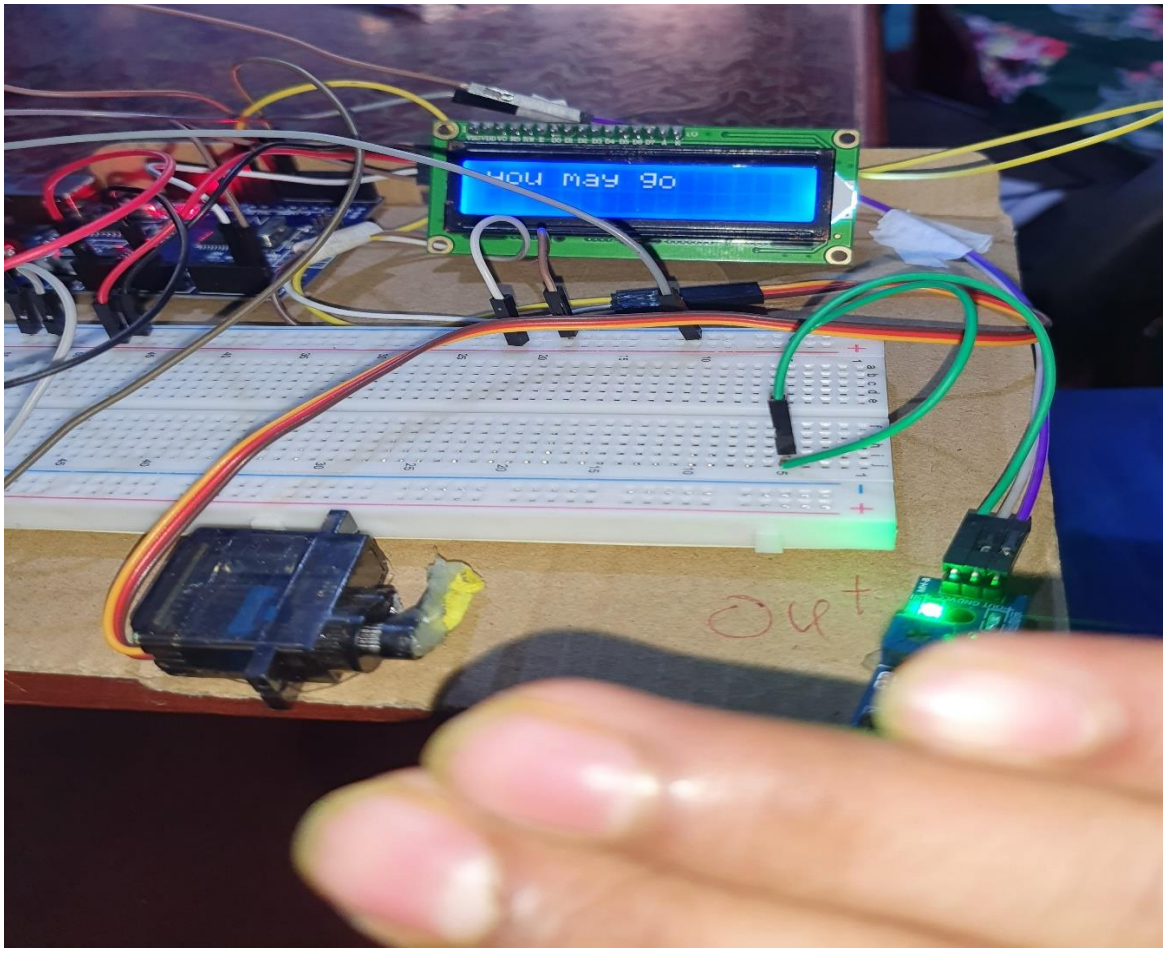


Figure 15: Exit Gate IR sensor working.

Test-3: Test if the Servo motor works or not.

Table 4: Test-3

Objective:	To test the working of Servo Motors.
Activity:	Arduino connected to the laptop and running the code.
Expected Result:	Servo motor should work to open and close the gate.
Actual Result:	The servo motor worked successfully to act as an actuator for opening and closing the gate.
Conclusion:	The test is successful.

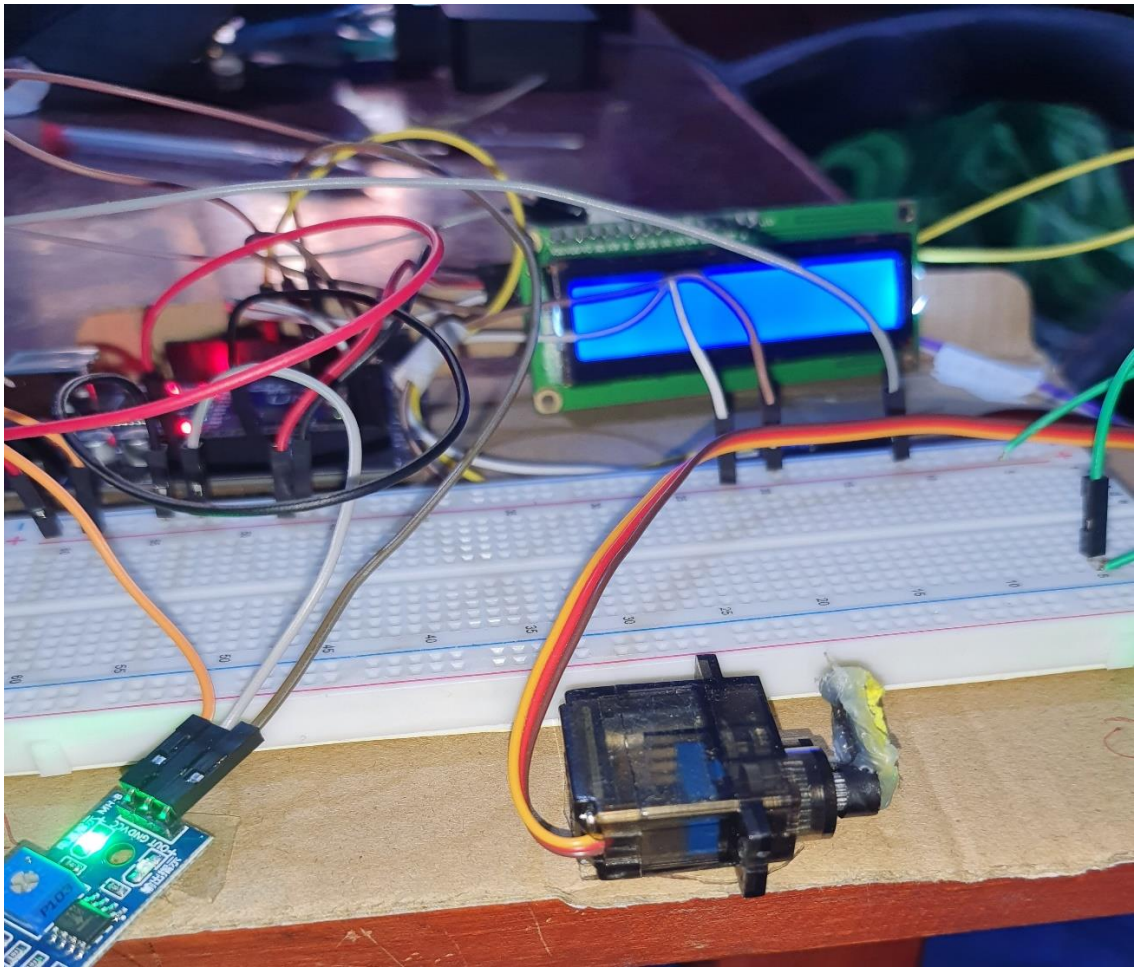


Figure 16: Servo motor Working

5. Future Works:

- Integration of the website application to the project for users' convenience.
- Implementation of AWS services to host website, upload backups data and many more.
- Integrating Camera system to the project for more safety.
- Using the various components like LED lights and Buzzers to make the alarm system while entering or exiting the car in the parking lot.
- Enhance the system with more security systems like integration of the laser light alarm system around the parking lot.
- Using ultrasonic sensors attached with buzzer to reduce the collision of the cars against the wall while parking.
- Using a RFID cards for the authorized users only while parking inside the private parking lot.

6. Conclusion:

This working prototype project is based on IoT components. From this project, we learnt about various IoT components, Circuit Connections, Arduino IDE, Tinkercad and many more.

The “Automated Car Parking System” is a revolutionary approach to parking management since it integrates hardware components such as IR sensors, mini servo motor and an LCD module with the Arduino Uno microcontroller. These components work together to automate parking procedures, allowing for effective parking area usage and real-time monitoring of available spots. The project was able to accomplish smooth integration and through planning by using the Arduino IDE for programming and Tinkercad as simulation. This project’s unique technology and cooperative effort sets the stage for future developments in smart parking solutions, providing a viable answer to the problem of urban parking.

Finally, the successful creation of the “Automated Car Parking System” stands as a testament to our group’s diligence and commitment. We extend our sincere appreciation to all individuals and involved, whose invaluable contribution led to successful completion of our system.

7. References

Arduino, 2023. *Arduino IDE.* [Online]
Available at: https://www.arduino.cc/en/software
[Accessed 11 01 2024].
Arduino, n.d. *arduino.com.* [Online]
Available at: https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics
[Accessed 2024].
Autodesk, Inc, 2024. *Tinker Cad introduction.* [Online]
Available at: https://www.tinkercad.com/learn
[Accessed 11 01 2024].
BYJU'S, 2023. *Circuit Diagram.* [Online]
Available at: https://byjus.com/physics/circuit-diagram/#:~:text=A%20circuit%20diagram%2C%20also%20known,of%20electrical%20and%20electronic%20equipment
[Accessed 11 01 2024].
DFROBOT, 2024. *I2C LCD Module.* [Online]
Available at: https://www.dfrobot.com/product-135.html#:~:text=This%20is%20a%2016x2%20LCD,the%20wire%20soldering%20and%20connection.
[Accessed 10 01 2024].
Draw.io, 2023. *Introduction.* [Online]
Available at: https://www.drawio.com/
[Accessed 11 1 2024].
Flyrobo, 2023. *Arduino UNO.* [Online]
Available at: https://www.flyrobo.in/blog/what-is-arduino-uno
[Accessed 11 01 2024].
Geeksforgeeks, 2024. *Ms word.* [Online]
Available at: https://www.geeksforgeeks.org/introduction-to-microsoft-word/
[Accessed 2024].
Gillis, A. S., 2024. *Techtarget-Internet of Things(IoT).* [Online]
Available at: https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT
[Accessed 11 01 2024].
Probots, 2024. *Breadboard.* [Online]
Available at: https://probots.co.in/830-points-solder-less-prototype-pcb-breadboard-premium-quality.html
[Accessed 11 01 2024].
Rozum, 2024. *Servo motor.* [Online]
Available at: https://rozum.com/mini-servo-motors/#:~:text=Basic%20working%20principles%20of%20mini,hold%20and%20re%2Destablish%20position.
[Accessed 11 01 2024].
Sunrom, 2024. *IR proximity sensor.* [Online]
Available at: https://www.sunrom.com/p/infrared-sensor-board#:~:text=%2F%2D%20Buy%20Now!-,IR%20Proximity%20Sensor,reflected%20light%20from%20the%20object.
[Accessed 11 01 2024].

The Rising Nepal, 2022. *Tackle Parking Problems.* [Online]
Available at:
https://risingnepaldaily.com/news/15862#:~:text=As%20the%20Kathmandu%20Metropolitan%20City,in%20the%20core%20city%20areas.
[Accessed 11 01 2024].
Wiltronics Research Pty Ltd , 2022. *Jumper wires.* [Online]
Available at: https://www.wiltronics.com.au/wiltronics-knowledge-base/what-are-jumper-wires/
[Accessed 11 01 2024].

8. Appendix:

8.1 Appendix A: Source Code

The following code represents the source code for overall system. The coding was done in Arduino IDE.

```
#include<Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

LiquidCrystal_I2C lcd(32, 16, 2);//set screen to the correct value
Servo myServo; // object creation
#define inGate 6 // Ir1
#define outGate 7 // Ir2
#define servoPin 9
int availableSpaces=5;//avalaible space for car parking

int servoPos = 0;//initialiase servo position

void setup() {
  lcd.init();
  lcd.clear();
  lcd.backlight();
  lcd.display();
  lcd.setCursor(0,0);
  pinMode(inGate, INPUT);
  pinMode(outGate, INPUT);
  myServo.attach(servoPin);
  Serial.begin(9600);


}

void loop() {

  if (digitalRead(inGate) == HIGH) {
    Serial.print(digitalRead(inGate));
    delay(25);
```

```
    Serial.flush();
    lcd.setCursor(0, 0);
    if (availableSpaces != 0) {
      lcd.print("Enter Plz!");
      lcd.setCursor(0, 1);
      lcd.print("Avai_spaces: ");
      lcd.print(availableSpaces);
      for (servoPos = 0; servoPos <= 180; servoPos += 1) {
        myServo.write(constrain(servoPos, 0, 180));
        delay(10);
      }
      delay(1000); // Adjust delay time as needed
      for (servoPos = 180; servoPos >= 0; servoPos -= 1) {
        myServo.write(constrain(servoPos, 0, 180));
        delay(15);
      }
    }

    availableSpaces -= 1;
    delay(1000);// Adjust delay time as needed
    lcd.clear();
  } else {
    if (digitalRead(outGate) == HIGH) {
        Serial.print(digitalRead(outGate));

    delay(25);
    Serial.flush();
      lcd.setCursor(0, 0);
      lcd.print("Clear Bills plz!");
      lcd.setCursor(0, 1);
      lcd.print("You May Go now!");
      for (servoPos = 0; servoPos <= 180; servoPos += 1) {
        myServo.write(constrain(servoPos, 0, 180));
        delay(15);
      }
      delay(1000); // Adjust delay time as needed
      for (servoPos = 180; servoPos >= 0; servoPos -= 1) {
        myServo.write(constrain(servoPos, 0, 180));
        delay(15);
      }
      availableSpaces += 1;
      delay(1000); // Adjust delay time as needed
      lcd.clear();
    }}


}
```

8.2 Appendix B: Design Diagrams

8.2.1 Block Diagrams

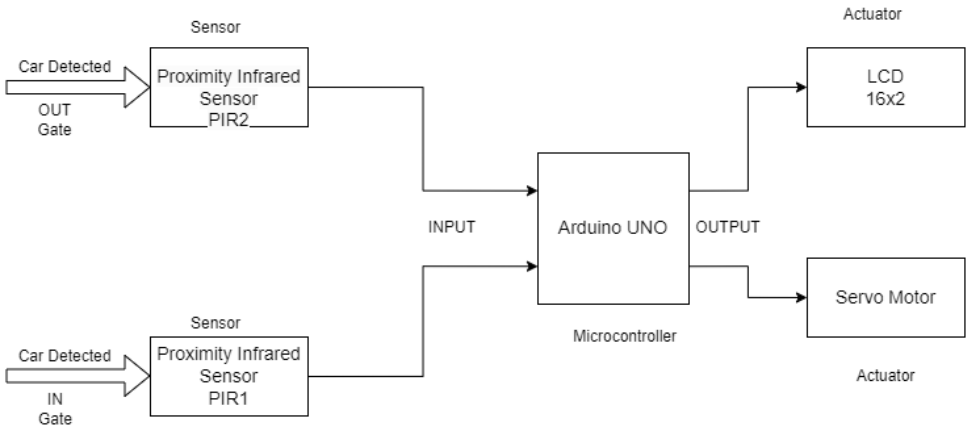


Figure 17: Block Diagram

8.2.2 Schematic View diagram

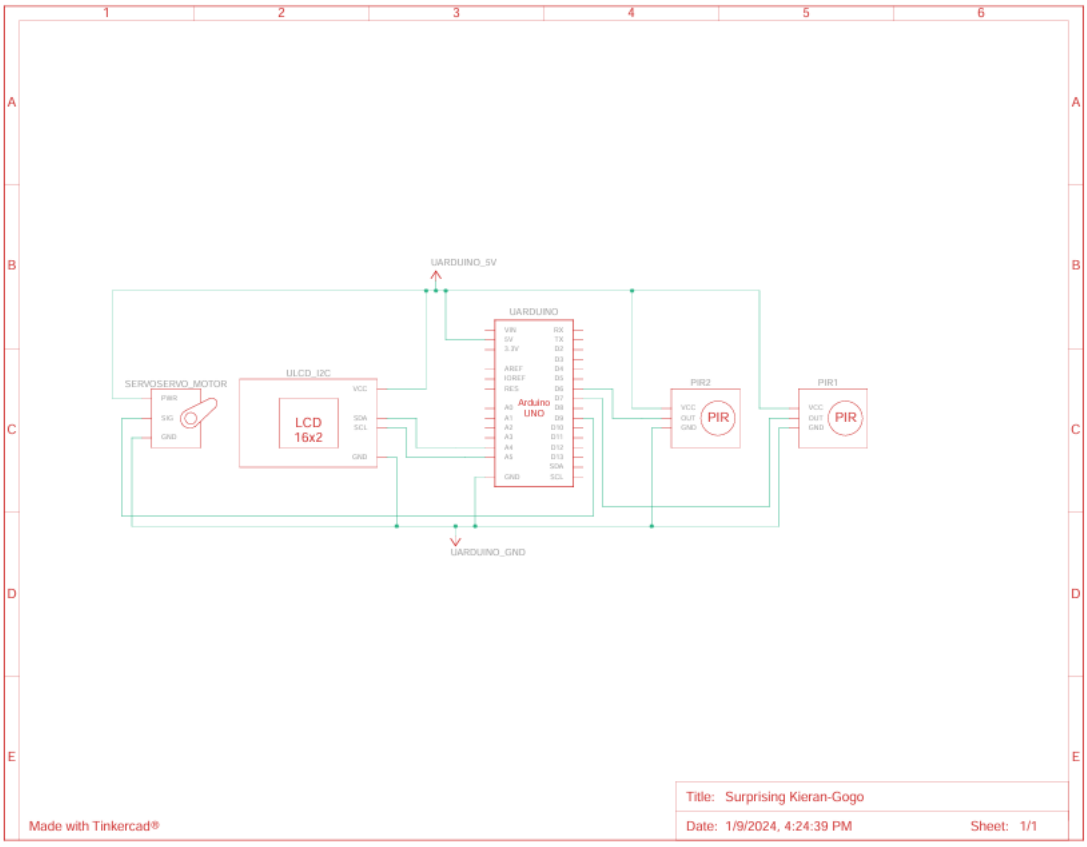


Figure 18: schematic diagram

8.3 Appendix C: Individual Contribution

Table 5: Individual contribution

Student Name	Assigned Task	Contribution
Bibek Kumar Thakur (Leader)	Task: Project Idea, proposal, System Architecture, Coding on IDE, designing prototype, device connections, Report Writing and Arranging, Simulations, Flowchart	30%
Apoorva K.C.	Task: Code debugging**,** documentation writing, Block diagram, Components descriptions, managing devices and inspecting connections among each device.	25%
Susmita Thakur	Task: Building a setup for demo of project, Information gathering for reports, storing hardware safely, System Architecture diagram, helping in connection procedure, Searching for online materials, etc	25%
Aryan Pudasaini	Task: Collecting hardware components, Connections in Tinkercad, Future works, Guiding connections, and setup, gathering additional resources required in project.	20%

8.4 Appendix D: Screenshots of the System.

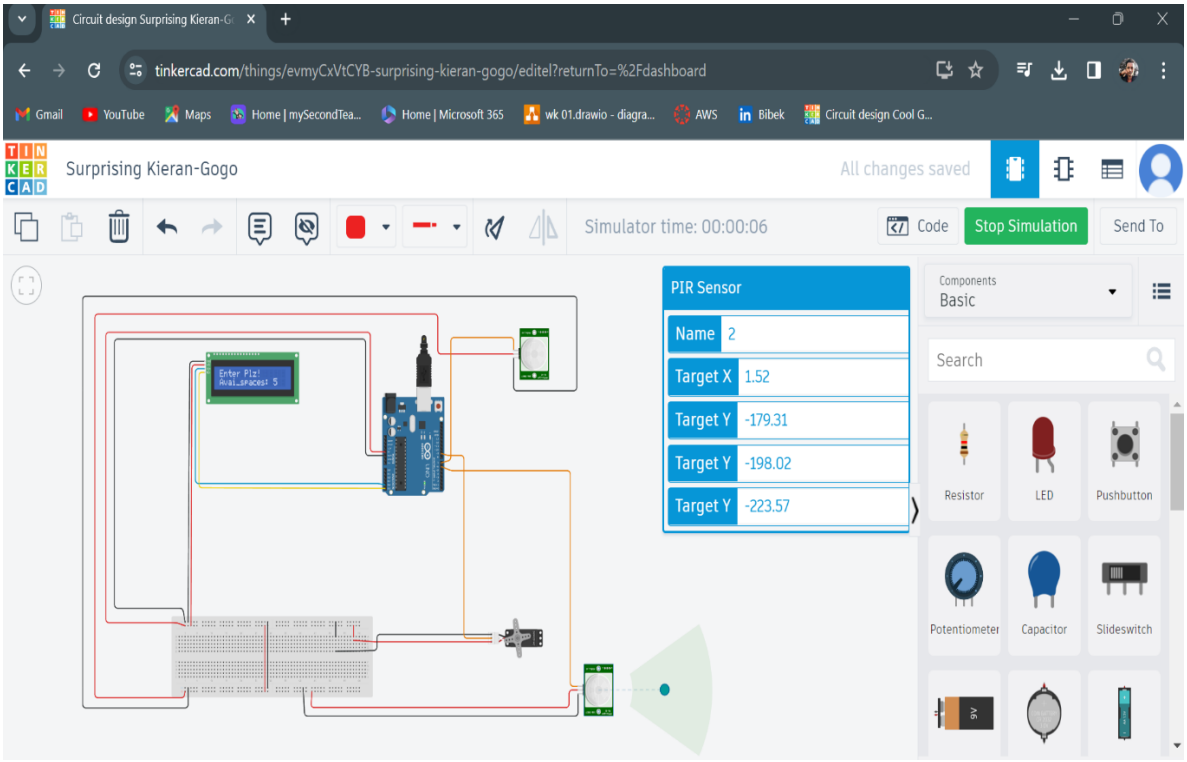


Figure 19: Screenshot 1 (IR1) and LCD

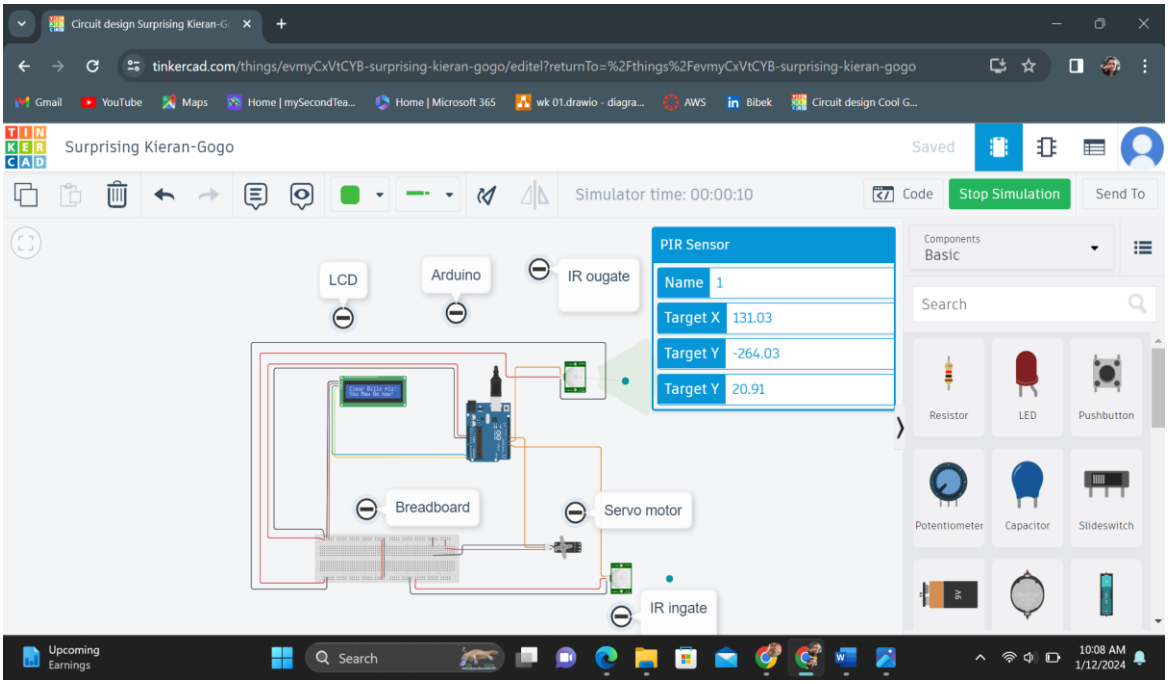


Figure 20: Screenshot 2 (IR2) and servo motor

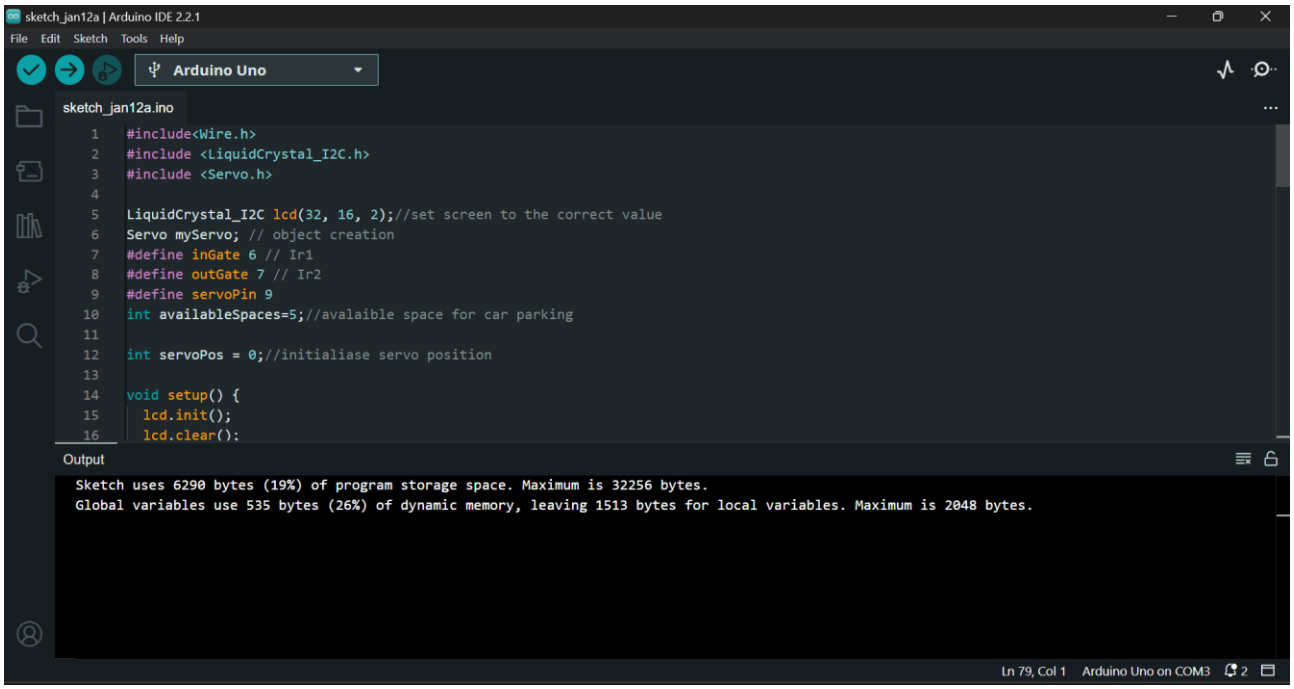


Figure 21: Coding in IDE