SMART CAR PARKING SYSTEM

A MINI PROJECT – II REPORT

Submitted by

JEEVANANTHAM CP	**717821L218**
SATHYANARAYANAN M	**717821L248**
THIRUPATHY S	**717821L259**
VIGNESH S	**717822L262**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

ANNA UNIVERSITY, CHENNAI

NOV 2023

KARPAGAM
COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

SMART CAR PARKING SYSTEM

Bonafide record of work done by

JEEVANANTHAM CP	**717821L218**
SATHYANARAYANAN M	**717821L248**
THIRUPATHY S	**717821L259**
VIGNESH S	**717822L262**

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF

ENGINEERING IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Anna University, Chennai

NOV 2023

...……
……..……
…….

Dr.S.Sobana M.E.,Ph.D
Faculty guide

Dr.R.Sarankumar M.E., Ph.D
Head of the Department

Certified that the candidate was examined in the viva-voce examination held on

……

……..

(Internal Examiner)

ACKNOWLEDGEMENT

We would like to show our gratitude to the management of Karpagam College of Engineering **Dr.R.Vasanthakumar, B.E., (Hons), D.Sc.,** Chairman and Managing Trustee, Karpagam Educational Institutions for providing us with all sorts of supports in completion of this project.

We express our sincere and profound gratitude to our Principal **Dr.V. Kumar Chinnaiyan M.E., Ph.D** for his guidance and sustained encouragement for the successful completion of this project.

We feel immense pleasure in expressing our humble note of gratitude to our Head of the Department **Dr.R.Sarankumar M.E., Ph.D** for his remarkable guidance and besides his positive approach he has offered incessant help in all possible way from the beginning.

We are grateful to our Project Coordinator **Prof . G . Rajarathinam M.E.,**Associate Professor, Department of Electronics and Communication Engineering for his valuable suggestions and guidance throughout the course of this project

We are thankful to our project guide **Dr.S.Sobana M.E.,Ph.D** Associate Professor, Department of Electronics and Communication Engineering for his valuable suggestions and guidance throughout the arise in the course of the project.

We also extend our thanks to other faculty members, parents and friends for\ providing their moral support in successfully completion of this project.

INDEX

ABSTRACT

The Smart Car Parking System mini project introduces an innovative and cost-effective solution to address parking management challenges in urban environments. This system combines Arduino Uno microcontroller, IR (Infrared) sensors, and an LCD display to create an intelligent parking system that enhances the parking experience for drivers and improves parking facility management.The system utilizes IR sensors strategically placed within parking spaces to detect the presence of vehicles. These sensors transmit data to the Arduino Uno microcontroller, which processes the information in real-time. The results are then displayed on an LCD screen, providing drivers with real-time parking availability information.The LCD display not only shows the availability of parking spaces but also offers additional features such as displaying parking rates, guidance to available spots, and emergency notifications. The system can be integrated with a mobile application to enable drivers to access parking information remotely.This mini project showcases the integration of hardware components and microcontroller programming to create a cost-effective and efficient smart parking solution. By providing drivers with real-time information on parking availability, the system helps reduce congestion and saves time, ultimately contributing to more efficient and convenient urban parking management.

LIST OF FIGURES

INTRODUCTION

CHAPTER1

1.1 Introduction to the Project

The "Smart Car Parking System" mini project is a creative fusion of hardware and software technologies aimed at revolutionizing the conventional parking experience. In an era marked by rapid urbanization and increasing vehicle ownership, finding an available parking spot has become a frustrating challenge for many. This project introduces a novel solution that leverages the power of Arduino Uno microcontroller, IR (Infrared) sensors, and an LCD display to simplify the parking process, making it more efficient and user-friendly.The conventional parking experience is often marred by the time-consuming search for an open parking space, leading to traffic congestion and driver frustration. The Smart Car Parking System offers a streamlined approach to tackle these issues. By deploying IR sensors in each parking space, the system can detect the presence or absence of a vehicle in real-time.The LCD display not only indicates the availability of parking spaces but can also provide additional helpful details such as current parking rates, directions to vacant spots, and emergency notifications, enhancing the overall parking experience.This mini project demonstrates the powerful combination of hardware components and microcontroller programming, offering a cost-effective and user-centric solution to urban parking challenges. By offering real-time parking availability information, the Smart Car Parking System contributes to reducing traffic congestion, saving time for drivers, and improving overall urban mobility. In the following sections, we will delve into the technical aspects and the implementation of this innovative smart parking solution.

1.2 Arduino UNO:

Arduino Uno is the most standard board available and probably the best choicefor a beginner. We can directly connect the board to the computer via a USB Cablewhich performs the function of supplying the power as well as acting as a serial port.

The pin diagram of Arduino Uno (Fig 1.1)shown below:

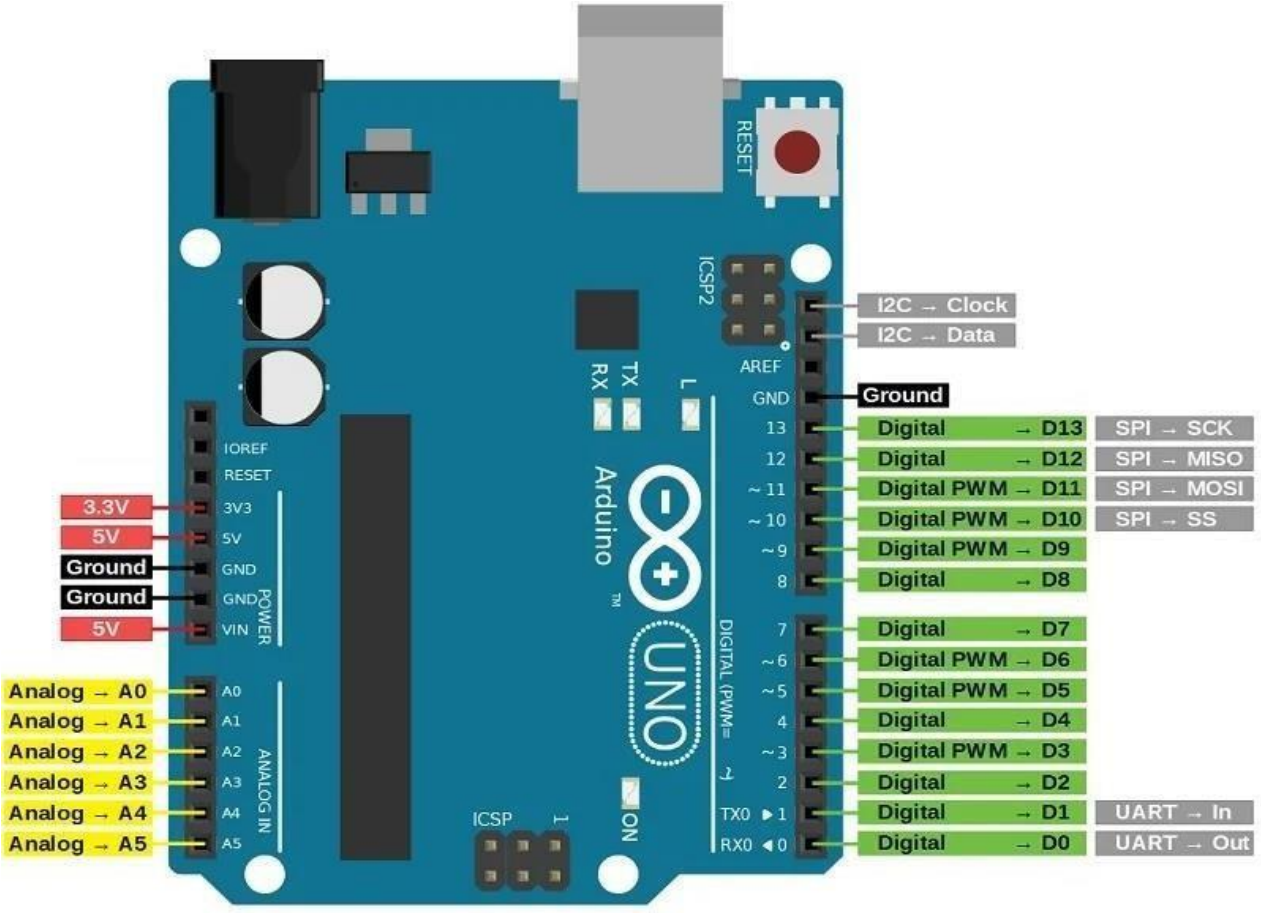

Fig 1.1 Pin diagram of Arduino Uno

1.2.1 Pin Description

Vin: This is the input voltage pin of the Arduino board used to provide input supply from an external power source.

5V: This pin of the Arduino board is used as a regulated power supply voltage and it is used togive supply to the board as well as onboard components.

3.3 V: This pin of the board is used to provide a supply of 3.3V which is generated from a voltageregulator on the board.

GND: This pin of the board is used to ground the Arduino board.

Reset: This pin of the board is used to reset the microcontroller. It is used to Resets the microcontroller.

Analog Pins: The pins A0 to A5 are used as an analog input and it is in the range of 0-5V.

Digital Pins: The pins 0 to 13 are used as a digital input or output for the Arduino board.

Serial Pins: These pins are also known as a UART pin. It is used for communication between the Arduino board and a computer or other devices. The transmitter pin number 1 and receiverpin number 0 is used to transmit and receive the data resp.

External Interrupt Pins: This pin of the Arduino board is used to produce the External interruptand it is done by pin numbers 2 and 3.

PWM Pins: This pins of the board is used to convert the digital signal into an analog by varyingthe width of the Pulse. The pin numbers 3,5,6,9,10 and 11 are used as a PWM pin.

SPI Pins: This is the Serial Peripheral Interface pin, it is used to maintain SPIcommunication with the help of the SPI library. SPI pins include:

1. SS: Pin number 10 is used as a Slave Select
2. MOSI: Pin number 11 is used as a Master Out Slave In
3. MISO: Pin number 12 is used as a Master In Slave Out
4. SCK: Pin number 13 is used as a Serial Clock

LED Pin: The board has an inbuilt LED using digital pin-13. The LED glows only when the digital pin becomes high.

AREF Pin: This is an analog reference pin of the Arduino board. It is used to provide a referencevoltage from an external power supply.

The Arduino Uno board is a microcontroller based on the ATmega328. It has 14 digital input/output pins in which 6 can be used as PWM outputs, a 16 MHz ceramic resonator, an ICSP header, a USB connection, 6 analog inputs, a power jack and a reset button. This containsall the required support needed for microcontroller. In order to get started, they are simply connected to a computer with a USB cable or with a AC-to-DC adapter or battery.It is featured by the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

In order to facilitate the access and use of electronic and programming Several students from the Institute of Interactive Design of Ivrea, Italy created it. They did it so that electronics students would have a cheaper alternative to the popular BASIC Stamp , boards which costs more than a hundred dollars in those days, So, not everyone could afford them.There hard work resulted an Arduino, a board with all the necessary elements to **connect peripherals to the inputs and outputs of a micro controller**, and which can be programmed in Windows as wellas macOS and GNU / Linux.

1.2.2 FEATURES OF THE ARDUINO UNO BOARD:

- It is an easy USB interface. This allows interface with USB as this is like a serial device.
- The chip on the board plugs straight into your USB port and supports on your computer as a virtual serial port. The benefit of this setup is that serial communication is an extremely easy protocol which is time-tested and USB makes connection with modern computers and makes it comfortable.
- It is an open source design and there is an advantage of being open source is that it has alarge community of people using and troubleshooting it. This makes it easy to help in debugging projects.
- It is a 16 MHz clock which is fast enough for most applications and does not speeds up the microcontroller.
- It is very convenient to manage power inside it and it had a feature of built-in voltage regulation. This can also be powered directly off a USB port without any external power. You can connect an external power source of upto 12v and this regulates it to both 5v and 3.3v.
- 13 digital pins and 6 analog pins. This sort of pins allows you to connect hardware to your Arduino Uno board externally. These pins are used as a key for extending the computing capability of the Arduino Uno into the real world. Simply plug your electronicdevices and sensors into the sockets that correspond to each of these pins and you are good to go.
- It has a 32 KB of flash memory for storing your code.
- An on-board LED is attached to digital pin 13 to make fast the debugging of code and tomake the debug process easy.
- Finally, it has a button to reset the program on the chip.

1.2.3 USES OF THE ARDUINO UNO:

The Arduino UNO board is primarily used over other Arduino products because of the following reasons.

- As the board can be easily connected to the other computer system via USB port. The USB port fixed in the board serves two purposes. It can be used to supply the power supply to the board and can act as a serial device to connect the board to a computer system.

- The board is capable to get the power supply from DC adaptor having a voltage of 12V. Theboard can be charged from this external power supply.

- The microcontroller used in the board I.e. ATmega328 has the flexibility provided to the board. It means the controller chip can be replaced, removed from the board in case of damageor improper functioning of the chip. This flexibility functionality is not provided in other Arduino boards.

- The benefit of this setup is that serial communication is an extremely easy protocol which istime-tested and USB makes connection with modern computers and makes it comfortable.

- The board pins are capable of functioning for constant power supply of 5 v. The digital andanalog pins are used to adjust the voltage supply in the board.

- The Arduino UNO board has a list of several hardware components and has the capability tointeract with those devices. The device includes Bluetooth, internet, motor control, and many more.

- The main use of the Arduino UNO board over other Arduino board is the price factor. The price of this board is lowest compared to other Arduino products. This is the main reason beginners prefer this board over other boards.

1.2.4 ADVANTAGES OF ARDUINO UNO:

1. Open-Source Platform: Arduino is an open-source platform, which means that the hardware and software designs are freely available for anyone to use, modify, and distribute. This fosters a large and active community of users and developers.

2. User-Friendly: Arduino is known for its user-friendly environment. The Arduino IDE (Integrated Development Environment) simplifies programming and is suitable for both beginners and experienced developers.

3. Wide Hardware Compatibility: Arduino boards are compatible with a wide variety of sensors, actuators, shields, and modules, making it easy to interface with a multitude of components.

4. Low-Cost Hardware: Arduino boards are relatively affordable, making them accessibleto hobbyists and students. There are even cheaper Arduino-compatible clones available.

5. Cross-Platform Compatibility: The Arduino IDE is available for multiple operating systems, including Windows, macOS, and Linux, allowing users to work on their platform of choice.

6. Educational Tool: Arduino is frequently used in educational settings to teach electronicsand programming. It provides a hands-on approach to learning.

7. Prototyping and Rapid Development: Arduino is excellent for prototyping, allowing you to quickly build and test your ideas before developing a final product.

1.2.5 DISADVANTAGES OF ARDUINO UNO:

1. Limited Processing Power: Arduino boards are equipped with microcontrollers, which have limited processing power compared to full-fledged microprocessors. This can be a limitation for complex tasks and real-time applications.

2. Limited Memory: Arduino boards typically have limited RAM and program memory, which can restrict the size and complexity of programs that can be run on them.

3. Not Suitable for High-End Graphics: Due to their limited processing power and memory, Arduino boards are not well-suited for applications that require high-end graphics or video processing.

4. Lack of Multithreading: Arduino programming is primarily single-threaded, which can make it challenging to handle multiple tasks simultaneously. Real-time operating systems (RTOS) can be used, but they add complexity.

5. Limited Connectivity: While Arduino boards offer basic I/O capabilities, they may lack built-in connectivity options for Wi-Fi, Bluetooth, or cellular communication. Additional shields or modules are required for these features.

6. Power Consumption: Arduino boards are not optimized for low power consumption. They are often designed to be powered continuously, which may not be suitable for battery-powered applications without careful power management.

7. No Operating System: Arduino lacks a full operating system, which limits the capabilities for running concurrent processes or more advanced software features.

8. Compatibility Issues: Newer Arduino boards may not be fully compatible with older libraries and sketches, requiring updates and adjustments when upgrading hardware.

CHAPTER 2

PROJECT MODULES

2.1 CIRCUIT DIAGRAM

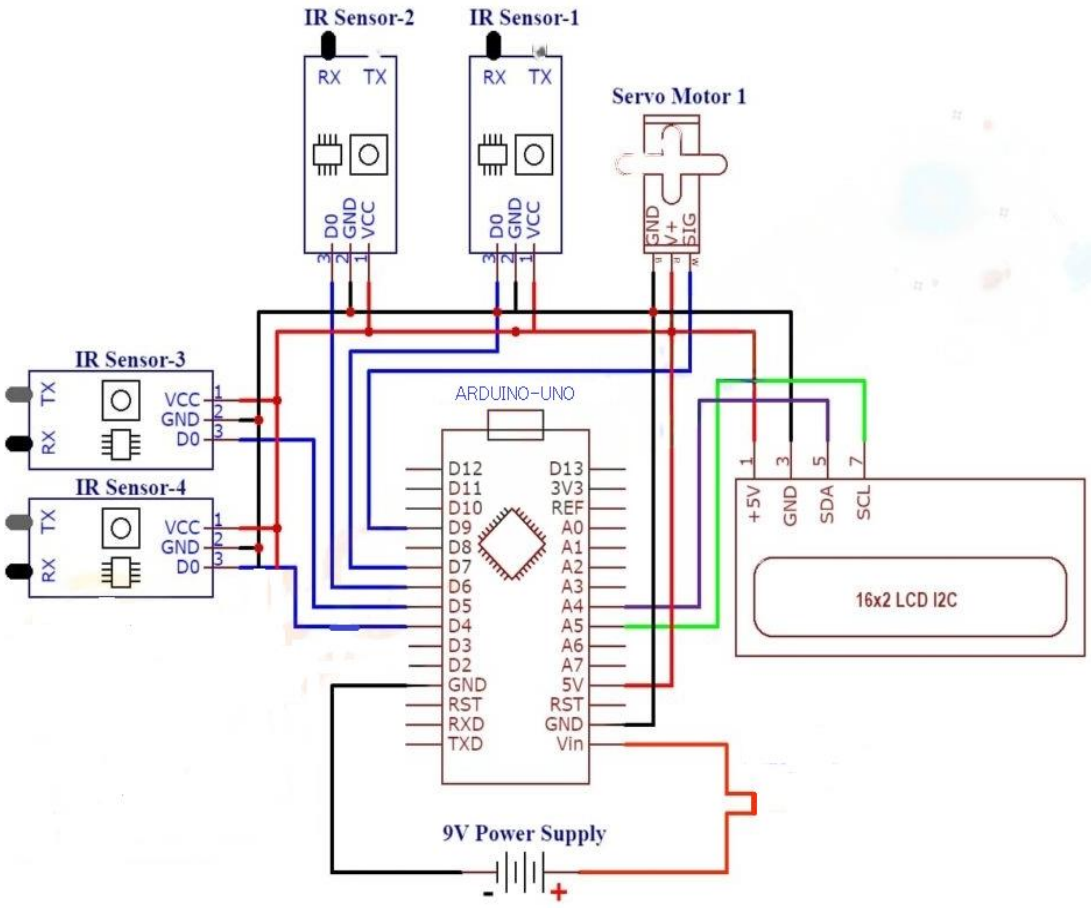

Fig2.1 Smart Car Parking System.

Fig 2.1 includes **Reset Circuit Design:** The reset resistor is selected such that the voltageatthe reset pin, across this resistor is at minimum of 1.2V and the width of thepulse appliedto this pin is greater than 100 ms..

Arduino Uno Interfacing Design: The set of 4 IR sensor such as slots, enter and exit is interfaced to pins 4,5,6,7 of the Arduino Uno. Servo motor

is interfaced to pin 9. Such that I2C is connected to 5v,Gnd and with analog input pins(A5,A4).

2.2 SYSTEM COMPONENTS:

- Arduino uno
- I2C LCD(16x2)Display
- Servo motor
- Jumper Wires
- 9V Rechargeable Battery

2.2.1 ARDUINO UNO:

The Arduino uno is a standard board of Arduino. Here UNO means 'one' in Italian. It wasnamed as UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is considered as the powerful board used in various projects. Arduino.cc developed the Arduino uno board. Arduino UNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits.The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms. The IDE is common to all available boards of Arduino.

The Arduino uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller (MCU) and developed by

Arduino. cc and initially released in 2010. The microcontroller board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) andother circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, andis programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by a USB cable or a barrel connector that accepts voltages between 7 and 20 volts, such as a rectangular 9-volt battery. It has the same microcontroller asthe Arduino Nano board, and the same headers as the Leonardo board.The hardware reference design is distributed under a Creative Commons Attribution Share-Alike
2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

2.2.2 I2C LCD(16X2)Display:

As the name suggests, these LCDs are ideal for displaying only characters. A 16×2 character LCD, for example, can display 32 ASCII characters across two rows.At the heart of the adapter is an 8-bit I/O expander chip – PCF8574. This chip convertsthe I2C data from an Arduino into the parallel data required for an LCD display.

The board also includes a tiny trim pot for making precise adjustments to the display's contrast. There is a jumper on the board that provides power to the backlight. To control the intensity of the backlight, you can remove the jumper and apply external voltage to the header pin labelled 'LED'.

Fig 2.2 I2C LCD Display

2.2.3 SERVO MOTOR:

A servo motor is a type of electric motor that can rotate or move to a specific position, speed, or torque based on an input signal from a controller. However, modern servo motors are capable of providing high performance and precision as main drives in various applications. servo motors are electric motors that allow for precise control of angular or linear position, speed, and torque. They consist of a motor, a sensor, and a controller that form a closed-loop feedback system.

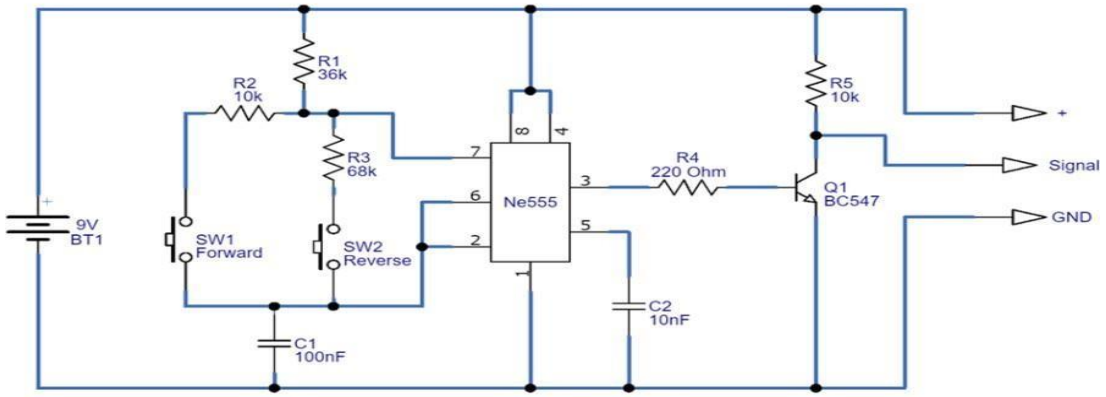

Fig 2.3 Servo Motor

2.2.4 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 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 wires. Though jumper wires come in a variety of colors, the colors don't actually mean anything. This means that a red jumper wire is technically the same as a black one. But the colors can be used to your advantage in order to differentiate between types of connections, such as ground or power as shown in Fig 2.5.

Fig 2.4 Jumper Wires

2.2.5 9V Rechargeable Battery:

The nine-volt battery, or 9-volt battery, is an electric battery that supplies a nominal voltage of 9 volts. Actual voltage measures 7.2 to 9.6 volts, depending on battery chemistry.Batteries of various sizes and capacities are manufactured a very common size is known asPP3, introduced for early transistor radio. Most

battery voltage testers and chargers that canalso test nine-volt need anothersnap clip to hold the battery, while cylindrical batteries often share a holder that may be adjustable in size.

2.3 CIRCUIT CONNECTION

Connect the VCC pin of Arduino, IR sensor, I2C to 5V power supply. Connect the GND pin of Arduino, IR sensor, I2C to GND. Connect the output digital pin of the servo motor to Arduino 9 Pin. Similarly, connect the output of IR sensor to Arduino digital Pin 1,2,4 and 6. A LCD display, which displays the information of the reading of IR sensor, is interfaced with the Arduino board.Lcd is conneted to the I2C SCA and SCL is conneted to the A4 and A5. The Arduino is programmed in such a way that if the sensor senses then the corresponding slots and output is displayed on the LCD Display.

2.4 OPERATION

- After assembling all components according to the circuit diagram and uploading the code to the Arduino board. Now place the sensors and servo motor at accurate positions.
- There are two parking slots in this project, IR sensor 4 and 6 are placed at slot 1and 2 respectively. IR sensor-1 and 2 are placed at the entry and exit gate respectively and a servo motor is used to operate the common single entry and exit gate. The LCD display is placed near the entry gate.
- The system used IR sensor-3 and 6 to detect whether the parking slot is empty or not and IR sensor-1, and 2 for detecting vehicles arriving or not at the gate.

- In the beginning, when all parking slots are empty, then the LCD display shows all slots are empty.
- When a vehicle arrives at the gate of the parking area then the IR sensor-1 detects the vehicle and the system allowed to enter that vehicle by opening the servo barrier. After entering into the parking area when that vehicle occupies a slot then the LED display shows that the slot is full. In this way, this system automatically allows 2 vehicles.
- In case the parking is full, the system blocked the entrance gate by closing the servo barrier. And the LED display shows that slot-1 and 2 all are full.
- When a vehicle leaves a slot and arrives at the gate of the parking area then the IR sensor-2 detects that vehicle and the system open the servo barrier. Then the LED display shows that the slot is empty. Again the system will allow entering a new vehicle.

2.5 OUTPUT

Fig 2.5 Shown that the entire circuit connection of smart car parking system.

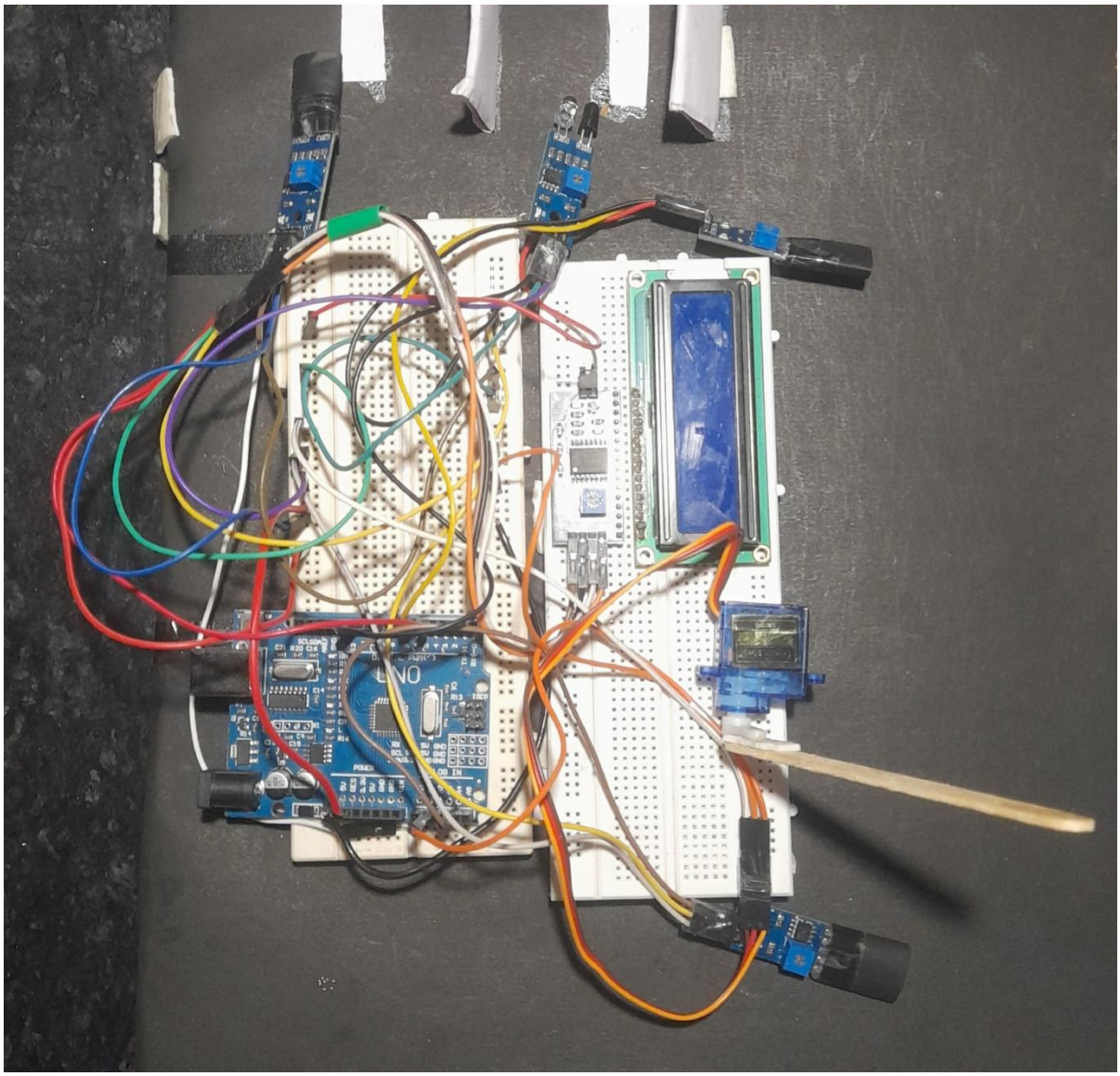

Fig2.5 circuit connection

Fig 2.6 Shown that the car parking is in initial state before entering an car.

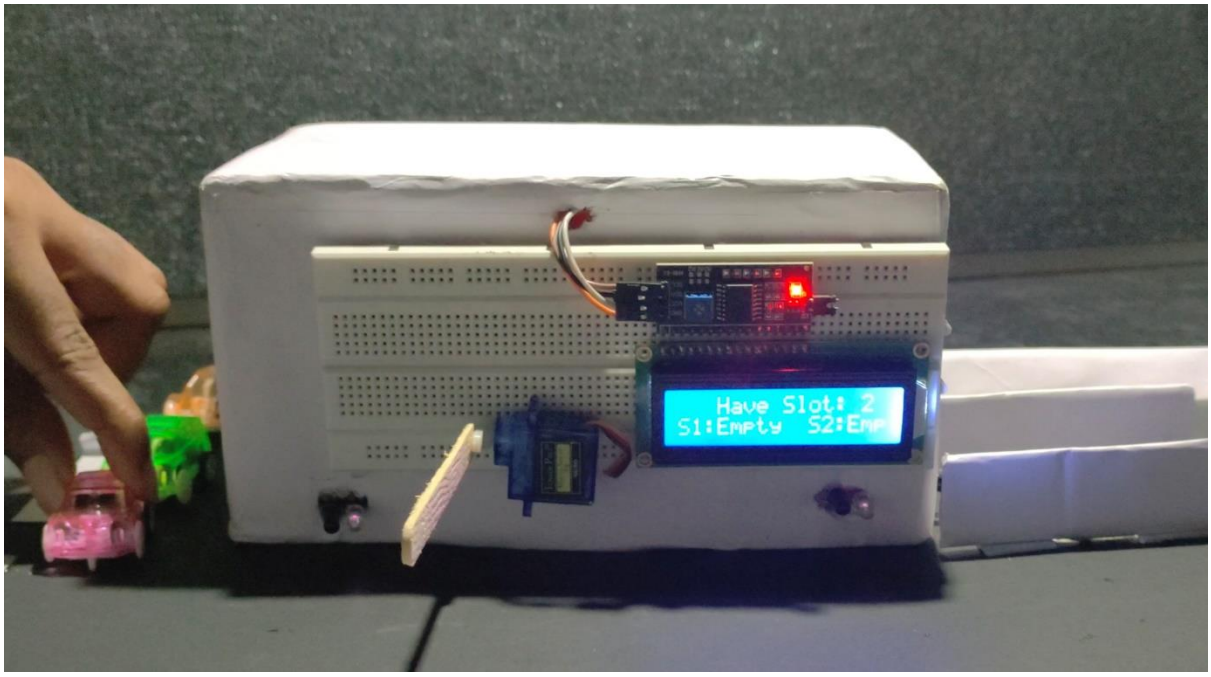

Fig 2.6 Initial State

Fig 2.7 Shown that the car is entering in the car parking system.

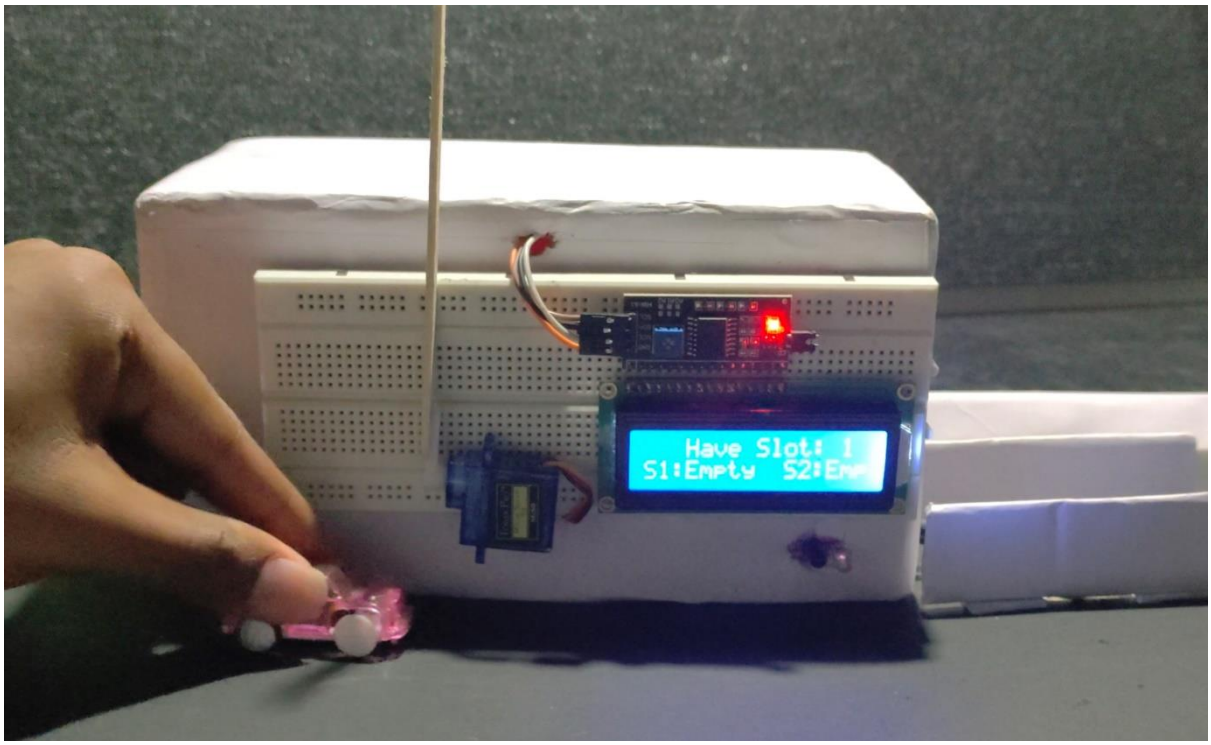

Fig 2.7 Enter

Fig 2.8 Shown that the parking is full while car entering in the car parking.

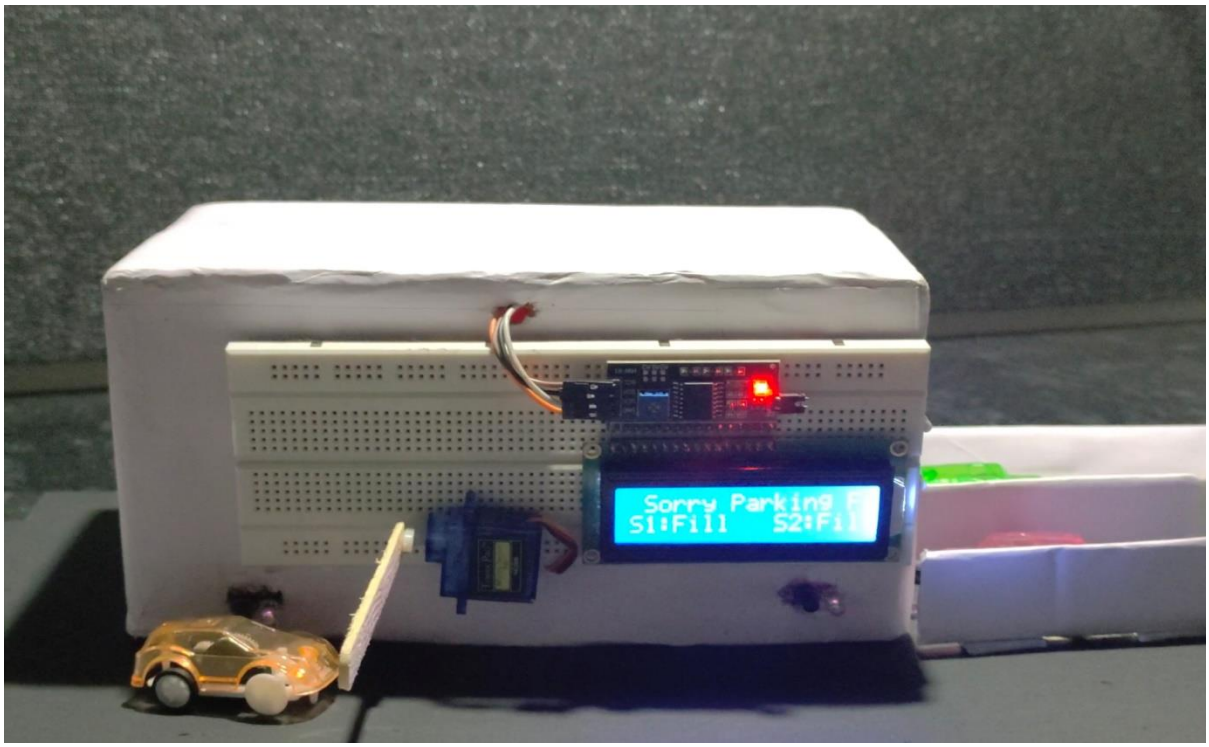

Fig 2.8 parking full

Fig 2.9 shown that the car is exiting from the car parking.

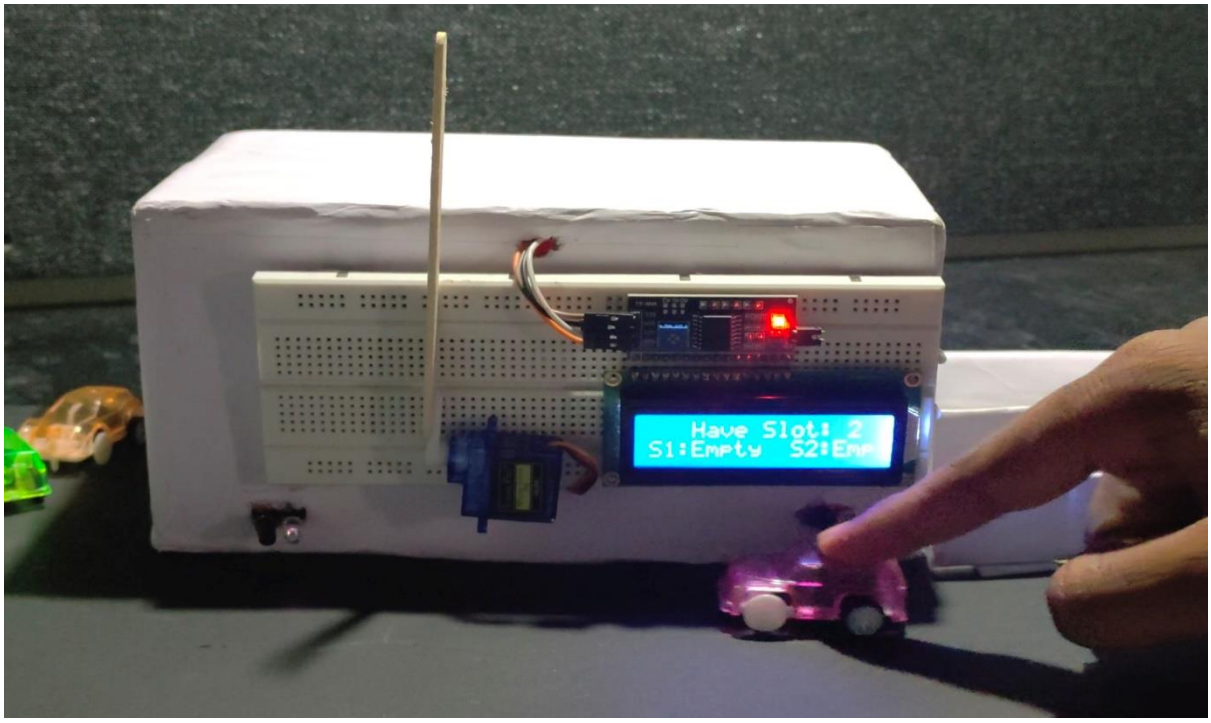

Fig 2.9 exit

CHAPTER 3

CONCLUSION

A smart car parking system using an Arduino Uno, LCD display, and IR sensor can be a useful mini project. Here's a conclusion for such a project:

In conclusion, our smart car parking system successfully combines hardware components like the Arduino Uno microcontroller, IR sensors, and an LCD display to create an efficient parking management solution. Throughout the project, we achieved the following key objectives:

1. Detection and Display: The IR sensors accurately detect the presence or absence of vehicles in parking spaces. This information is displayed in real-time on the LCD screen, providing users with clear feedback.

2. User-Friendly Interface: The LCD display simplifies the user experience by showing the availability of parking spaces and guiding drivers to empty slots.

3. Efficient Space Management: By continuously monitoring parking space occupancy, the system helps maximize parking efficiency, reducing the time and frustration of finding a spot.

4. Cost-Effective: The project uses affordable components, making it a cost-effective solution for small-scale parking management.

5. Learning Experience: This project serves as an excellent learning opportunity for understanding microcontroller programming, sensor integration, and user interface design.

REFERENCE

- https://techatronic.com
- https://arduinogetstarted.com/tutorials/arduino-lcd-i2c
- https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/
- https://www.hackster.io
- https://www.electroduino.com
- https://www.instructables.com

APPENDIX

Source code:

```
#include <Servo.h> //includes the servo library

#include <Wire.h>

#include <LiquidCrystal_I2C.h> //includes LiquidCrystal_I2C library

LiquidCrystal_I2C lcd(0x27, 20, 4);

Servo myservo;

#define ir_enter 2

#define ir_back  4

#define ir_car1 5

#define ir_car2 6

int S1=0, S2=0;

int flag1=0, flag2=0;

int slot = 2;

void setup(){

Serial.begin(9600);

// initialize digital pins as input.

pinMode(ir_car1, INPUT);

pinMode(ir_car2, INPUT);

pinMode(ir_enter, INPUT);

pinMode(ir_back, INPUT);

myservo.attach(9); // Servo motor pin connected to D9

myservo.write(90); // sets the servo at 0 degree position

// Print text on display
```

```
lcd.begin(20, 4);
lcd.setCursor (0,1);
lcd.print("    Smart Car   ");
lcd.setCursor (0,2);
lcd.print(" Parking System ");
delay (2000);
lcd.clear();
Read_Sensor();
int total = S1+S2;
slot = slot-total;
}
void loop()
{
 Read_Sensor();
 lcd.setCursor (0,0);
 lcd.print("   Have Slot: ");
 lcd.print(slot);
 lcd.print("    ");
 lcd.setCursor (0,1);
  if(S1==1)
   {
    lcd.print("S1:Fill ");
   }
  Else
```

```
   {
    lcd.print("S1:Empty");
   }
 lcd.setCursor (10,1);
  if(S2==1)
   {
    lcd.print("S2:Fill ");
    }
  else
   {
    lcd.print("S2:Empty");
    }
/* Servo Motor Control
**********************/
  if(digitalRead (ir_enter) == 0 && flag1==0) // read degital data from
IR sensor1
   {
    if(slot>0)
     {
      flag1=1;
      if(flag2==0)
       {
        myservo.write(180);
        slot = slot-1;
```



```
        }
```

```
      }
     else
      {
       lcd.setCursor (0,0);
       lcd.print(" Sorry Parking Full ");
       delay(1500);
       }
    }
  if(digitalRead (ir_back) == 0 && flag2==0) // read degital data from
IR sensor2
    {
     flag2=1;
     if(flag1==0)
      {
       myservo.write(180); // sets the servo at 180 degree position
       slot = slot+1;
       }
    }
  if(flag1==1 && flag2==1)
    {
     delay (1000);
     myservo.write(90); // sets the servo at 90 degree position
     flag1=0, flag2=0;
```



```
     }
     delay(1);}
```

```
void Read_Sensor()

{

 S1=0, S2=0;

 if(digitalRead(ir_car1) == 0){S1=1;} // read degital data from IR
sensor3

 if(digitalRead(ir_car2) == 0){S2=1;} // read degital data from IR
sensor4}
```