Project Progress Report (Phase-1) on

SMART CAR PARKING SYSTEM USING ARDUINO UNO

submitted in partial fulfillment of the
Degree of B. Tech in Electronics and Communication Engineering
under Bihar Engineering University, Patna

Submitted By:

Raju Kumar – 21104130903

Shubham Kr. Jha –20104130016

Adarsh kumar – 20104130038

Under supervision of

Mr. Ravi Ranjan

Assistant Professor & HoD

Electronics and Communication Engineering
Supaul College of Engineering, Supaul
Bihar Engineering University, Patna

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
SUPAUL COLLEGE OF ENGINEERING, SUPAUL - 852131.
UNDER DEPT. OF SCIENCE & TECHNOLOGY, PATNA, GOVT. OF BIHAR

CANDIDATE'S DECLARATION

We hereby declare that the work which is being presented in this project progress report entitled, "**Smart Car Parking System Using Arduino UNO"** in partial fulfillment of requirement for the 06 credit course Project-I (Course code-100709) of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering**, submitted in the department of ELECTRONICS AND COMMUNICATION ENGINEERING, SCE Supaul, is an authentic record of our own work carried out under the guidance of **Mr. Ravi Ranjan,** Assistant Professor and Head of Department of ELECTRONICS AND COMMUNICATION ENGINEERING, SCE Supaul.

The matter embodied in this report has not been submitted anywhere by us for the award of any other degree or diploma.

Sl. No.	**Name of Student**	**Registration No.**	**Signature**
1.	**SHUBHAM KR. JHA**	**20104130016**	
2.	**RAJU KUMAR**	**21104130903**	
3.	**ADARSH KUMAR**	**20104130038**	

Date:

Palace: SCE, Supaul

CERTIFICATE

This is to certify that project report entitled "**Smart Car Parking System Using Arduino UNO**" which is submitted by "**Shubham kr. jha (20104130016)**, **Raju Kumar (21104130903)**, **Adarsh kumar (20104130038)**" in partial fulfillment of the requirement, for the 06-credit course Project-I (Course code- 100709), of the degree of Bachelor of Technology in department of Electronics and Communication Engineering of Supaul College of Engineering, Supaul is a record of the candidate own work carried out by them under my/our supervision. The matter embodied in this project report is original and has not been submitted anywhere for the award of any other degree or diploma.

Project Supervisor	Head of Dept. ECE
(Mr. Ravi Ranjan)	(Mr. Ravi Ranjan)
SCE, Supaul	SCE, Supaul

(Internal Examiner)	(External Examiner)

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people. We are extremely fortunate to have this all along the completion of our project. First and foremost, we would like to thank our supervisor Mr. Ravi Ranjan, Assistant Professor, Head of Dept. Electronics and Communication Engineeringof this project for the guidance and advice. He inspired us greatly to work on this project. His willingness to motivate us contributed tremendously to our Project.

Finally, an honorable mention goes to our Principal Dr. A.N. Mishra and other faculty members for their support and suggestions. Also, we would like to extend our sincere regards to all non-teaching staff of our college for their timely support and cooperation.

PREFACE

We take great opportunity to present this Mini Project report on "**Smart Car Parking System Using Arduino UNO"** and put before readers some useful information regarding our project.

We have made sincere attempts and taken every care to present this matter in precise and compact form, the language being as simple as possible. We are sure that the information contained in this volume certainly proves useful for better insight in the scope and dimension of this project in its true perspective.

The task of the completion of the project though being difficult was made quite simple, interesting and successful due to deep involvement and complete dedication of our group members.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The Smart Car Parking System is a remarkable advancement poised to transform the parking landscape through the integration of cutting-edge technology. Incorporating Arduino Uno, IR sensors, and servo motors, this innovative solution promises to revolutionize the traditional parking experience, addressing key challenges such as congestion, space utilization inefficiencies, and manual ticketing processes. In this era of rapid urbanization and technological evolution, the need for smarter, more efficient parking solutions has never been more pronounced. The Smart Car Parking System emerges as a beacon of progress, offering a holistic approach to parking management that enhances efficiency, improves user experience, and fosters sustainable urban mobility.

At the heart of the Smart Car Parking System lies Arduino Uno, a versatile microcontroller renowned for its flexibility and functionality. Serving as the central brain of the system, Arduino Uno orchestrates the various components and processes, ensuring seamless operation from vehicle detection to barrier control. Its ability to process inputs from IR sensors, communicate with the user interface, and control servo motors makes it an indispensable element of the parking system's architecture.

Complementing Arduino Uno are IR sensors, which play a pivotal role in detecting the presence of vehicles within parking spaces. Strategically placed throughout the parking facility, these sensors continuously monitor occupancy status, providing real-time data to Arduino Uno. This information forms the foundation of the system's intelligence, enabling it to accurately assess space availability and facilitate efficient parking allocation.

Furthermore, servo motors serve as the mechanical arms of the Smart Car Parking System, responsible for opening and closing barriers or gates to regulate vehicle entry and exit. Controlled by Arduino Uno based on input from IR sensors, servo motors ensure smooth and precise operation, contributing to the overall efficiency and reliability of the system.

The operation of the Smart Car Parking System unfolds in a series of coordinated steps, each designed to optimize the parking experience for drivers and operators alike. As a vehicle approaches the entrance of the parking facility, IR sensors detect its presence and relay this information to Arduino Uno. Utilizing this data, Arduino Uno determines the availability of parking spaces and communicates the information to the user interface, allowing drivers to make informed decisions regarding parking allocation.

Upon selecting an available parking space, the driver proceeds to the designated location. Arduino Uno signals the corresponding servo motor to open the barrier, granting access to the parking area. Throughout this process, the user interface provides real-time guidance and updates, ensuring a seamless and convenient experience for the driver.

Once parked, the driver can rest assured that the Smart Car Parking System continues to monitor the parking space, ready to assist with the exit process when needed. When the driver is ready to leave, Arduino Uno triggers the servo motor to open the exit barrier, allowing the vehicle to exit the parking area smoothly. This automated process minimizes delays and congestion, contributing to a more efficient flow of traffic within the parking facility.

In addition to its core functionalities, the Smart Car Parking System offers several value-added features designed to enhance user experience and streamline parking operations. These may include parking guidance systems, which provide visual cues to assist drivers in maneuvering into their designated spaces, as well as automated payment options, allowing for seamless transactions without the need for manual intervention.

Furthermore, the system may incorporate feedback mechanisms through the user interface, enabling drivers to provide input on their parking experience. This valuable feedback can be used to identify areas for improvement and optimize the system's performance over time, ensuring continued satisfaction for users and operators alike.

The benefits of the Smart Car Parking System extend beyond mere convenience, encompassing broader implications for urban mobility and sustainability. By optimizing space utilization and minimizing congestion, the system contributes to a reduction in carbon emissions and environmental impact associated with vehicle emissions and traffic congestion. Moreover, the efficiency gains achieved through automation and smart management translate into cost savings for parking operators, further incentivizing the adoption of this transformative technology.

As cities continue to grow and evolve, the demand for innovative parking solutions will only intensify. The Smart Car Parking System represents a significant step forward in meeting this demand, offering a scalable and adaptable solution that can be tailored to the unique needs of any parking facility. Whether in urban centers, commercial complexes, or residential developments, the integration of smart parking technology promises to enhance the quality of life for residents and visitors alike.

In conclusion, the Smart Car Parking System stands as a testament to the power of technology to address complex urban challenges and improve the way we live, work, and move within our cities. By harnessing the capabilities of Arduino Uno, IR sensors, and servo motors, this innovative solution offers a glimpse into the future of parking management—one characterized by efficiency, convenience, and sustainability. As we embrace this era of digital transformation, the Smart Car Parking System serves as a beacon of progress, guiding us towards a more connected, intelligent, and inclusive urban environment.

CHAPTER 2

REVIEW OF LITERATURE

2.1 Smart Car Parking System using Arduino[1]

The paper discusses a Smart Parking system leveraging IoT technology, aiming to streamline parking by monitoring space availability. It introduces an IoT-based framework for efficient parking, detailing the deployment of IoT modules to monitor and signal individual parking spot availability. Additionally, it outlines the system architecture and presents a high-level view. The implementation involves Ultrasonic Range Detection Sensors paired with Arduino to detect empty slots. By measuring distances, these sensors help drivers easily locate vacant spots, reducing search time. The system's efficacy is validated through a use case demonstration, highlighting its functionality.[1]

2.2 Advance Car Parking System using Arduino[2]

This paper explains the architecture and design of Arduino based car parking system. Authorization of driver or user is the basic rule used to park a vehicle in a parking place. Authorization card will be given to each user, which carries the vehicle number or other details. If the user is authorized and space is available in the parking, then the parking gate will open and the user is allowed to park the vehicle in parking place else the user is not allowed even the user is authorized person. If car is allowed to park, then mobile notification will be send to user about parking. It solves the parking issue in urban areas, also provides security to a vehicle and an unauthorized user is not allowed to enter into a parking place. It helps to park vehicle in multifloored parking also as it will display which floor has free space.[2]

2.3 Automatic Car Parking Assistance Using Arduino[3]

In the context of today's crowded urban environments, the growing issue of insufficient parking availability due to the rising number of vehicles and inadequate infrastructure poses significant challenges. This paper tackles the problems of arriving at parking facilities only to find no spaces and struggling to locate open parking spots within extensive lots, resulting in time wastage and driver frustration. To alleviate these difficulties, the implementation of an efficient Smart Parking Management System is required for various parking facilities. This system aims to streamline and simplify the parking experience, minimizing time wastage, and providing real-time information on available parking spaces.

The Smart Parking project leverages an array of technologies, including Arduino micro-controllers, a user-friendly display system, and servo motors, to automate the identification and management of open parking slots. Furthermore, the system offers real-time parking space information through a 16x2 LCD display located outside the parking facility, enhancing user convenience by eliminating the need to enter the facility in search of parking. Its adaptability and potential for further enhancements make it a promising solution for addressing contemporary urban parking challenges.[3]

2.4 Smart Parking System Project using Arduino and IR Sensor[4]

In these modern days finding car parking is a big issue in congested cities. There are too many vehicles on the road but not enough parking spaces. One of the biggest problems is when we enter a parking area then we realize that there are no empty parking slots to park our cars. Important time. Another biggest problem is after entering in a big parking area we confused to find the empty parking slot to park our car. Sometimes maybe we all have been facing these two problems that wasted our important time. That’s why we need efficient parking management systems in all parking areas that will provide confusion-free and easy parking.

In this tutorial, we will design a “Smart Parking System Project” to overcome this problem. This project helps the car’s driver to park their car with minimum wastage of time with accurate information of the availability of the space to park.[4]

2.5 Parking System[5]

Due to the increasing number of cars in cities, finding adequate parking solutions has become essential. Traditional methods of parking are no longer feasible due to the substantial space they require, which is increasingly scarce and expensive in urban areas. Consequently, there's a pressing need for parking solutions that occupy minimal space while accommodating the maximum number of vehicles. The escalating cost of land in cities further emphasizes the necessity for space-efficient parking solutions. Additionally, the average person spends a significant portion of their travel time searching for parking spots in metropolitan areas, highlighting the urgency for effective systems. A functional mechanism is required to operate the parking system, along with a detection system to inform users about the availability of free parking spots. It's crucial to consider the safety of both cars and pedestrians when designing such systems. This paper delves into various types of parking systems and the diverse array of sensors utilized to enhance safety and efficiency in parking management.[5]

CHAPTER 3

THEORETICAL BACKGROUND

3.1 Arduino UNO

Arduino is an open-source hardware board with many open-source libraries to interface its onboard microcontroller with many other external components like LEDs, motors, LCDs, keypads, Bluetooth modules, GSM modules, and many other things one wantsto interface with Arduino board. Arduino is basically made from a microcontroller, but Arduino has all external pinouts to connect with other devices, and it also has a built-in programmer which is used to program Arduino from a computer. So Arduino is a complete board which includes all things to connect with external peripherals and is easy to program through a computer. [7]

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. [8]

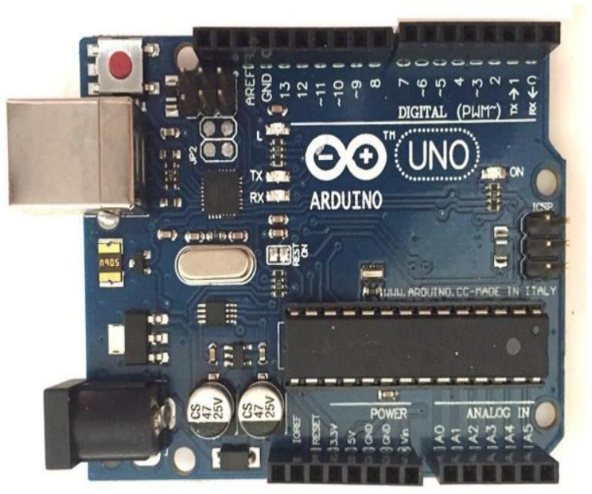


Fig 3.1: Arduino UNO [9]

3.1.1 Pinout Configuration [9]

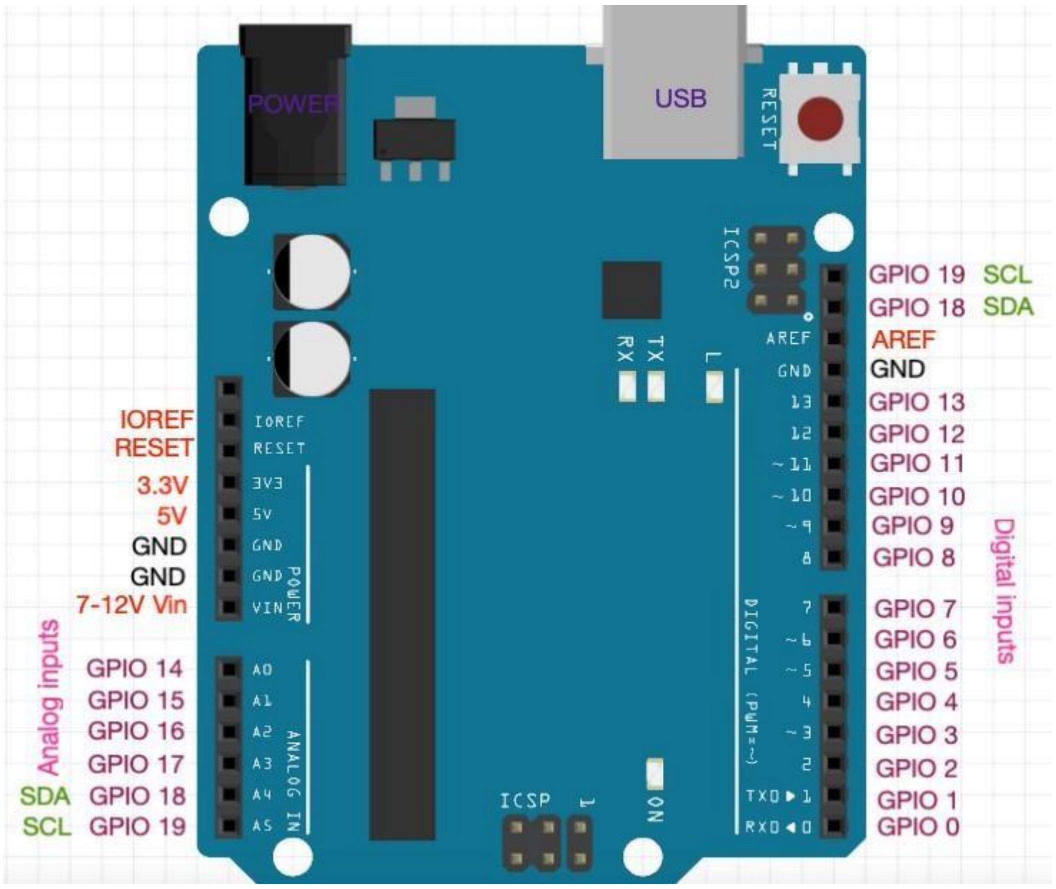


Fig 3.1.1.1: Arduino Board description[10]

Pin No.	Name and Use

Pin 1. **Power USB**

An Arduino board can be powered by using the USB cable from yourcomputer. All you need to do is connect the USB cable to the USB connection (1).

Pin 2. **Power (Barrel Jack)**

Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

Pin 3. **Voltage Regulator**

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

Pin 4. **Crystal Oscillator**

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 Mhz.

Pin 5 **Arduino Reset**

You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board.

Pin 6,7,8,9 **Pins (3.3, 5, GND, Vin)**

- 3.3V (6) − Supply 3.3 output volt
- 5V (7) − Supply 5 output volt
- Most of the components used with Arduino boards work fine with 3.3 volt and 5 Volt.
- GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

Pin 10. **Analog pins**

The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

Pin 11. **Main microcontroller (ATMega328P)**

Each Arduino board has its own microcontroller. You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company.

Pin 12. **ICSP pin**

Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND.

Pin 13. Power LED indicator

This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

Pin 14. TX and RX LEDs

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speeds while sending the serial data. The speed of flashing depends on the baud rate used by the board.

Pin 15. Digital I/O

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled "~" can be used to generate PWM.

Pin 16. AREF

AREF stands for Analog Reference. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Microcontroller used in Arduino - Atmega328P [11]

ATmega328P is one of the high Performance AVR technology microcontrollers with a large number of pins and features. It is designed by 8-bit CMOS technology and RSIC CPU which enhance its performance and its power efficiency get improved by auto sleeps and internal temperature sensor. This ATmega328P IC comes with internal protections and multiple programming methods which helps the engineers to prioritize this controller for different situations. The IC allows multiple modern era communications methods for other modules and microcontrollers itself, which is why the microcontroller ATmega328P usage has been increasing every day.

The high-performance Microchip picoPower® 8-bit AVR® RISC-based microcontroller combines 32 KB ISP Flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented Two-Wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter.

Fig 3.1.1.2: ATMEGA328P Microcontroller [11]

3.1.2 Technical Specification

MCU	ATmega328P
Architecture	AVR
Operating Voltage	5V
Input Voltage	6V – 20V (limit) 7V – 12V (recommended)
Clock Speed	16 MHz
Flash Memory	32 KB (2 KB of this used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Digital IO Pins	24 (of which 6 can produce PWM)

3.2 IR Sensor

An IR (Infrared) sensor is a device that detects infrared radiation, which is invisible to the human eye. It works based on the principle of detecting changes in the intensity of infrared radiation emitted by objects

IR sensors are widely used in various applications such as motion detection, proximity sensing, object detection, temperature measurement, and more. They are commonly found in consumer electronics, security systems, industrial automation, and automotive applications. . Here's how they work.

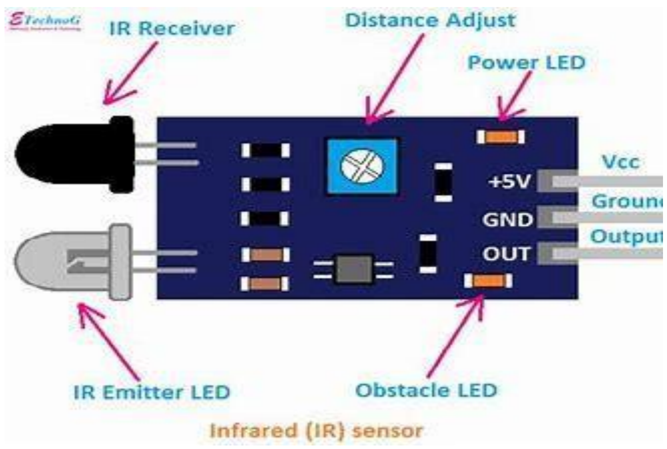


Fig 3.2. IR Sensor[12]

3.1.1 How Does a IR Sensor Work? [12]

The working principle of an IR sensor is based on the interaction between infrared radiation and objects. When an object emits or reflects infrared radiation, it generates heat. IR sensors detect this heat radiation and convert it into an electrical signal, which can be processed to determine the presence, distance, or temperature of the object.

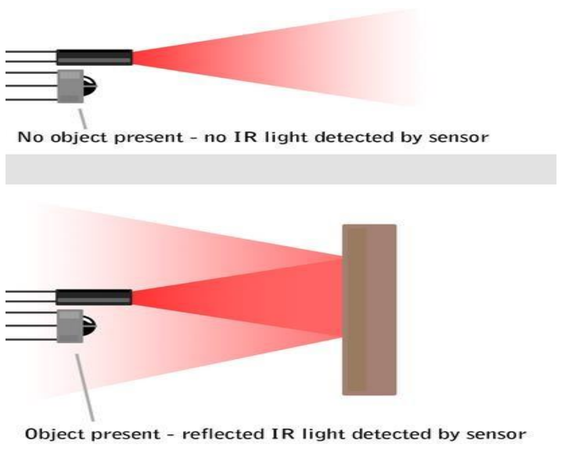

No object present - no IR light detected by sensor
Object present - reflected IR light detected by sensor

Fig 3.2.1: Working of IR Sensor [13]

Infrared (IR) sensors are indispensable components in various technological applications, leveraging the properties of infrared radiation to detect objects, monitor temperature, and facilitate communication. This comprehensive guide will explore the working principles, types, applications, and advancements of IR sensors.

Working Principle:

The fundamental principle behind IR sensors lies in the detection and interpretation of infrared radiation. Infrared radiation is a form of electromagnetic radiation with wavelengths longer than those of visible light but shorter than microwaves. Objects emit, absorb, and reflect infrared radiation based on their temperature and composition.

An IR sensor typically consists of an emitter and a receiver. The emitter emits infrared radiation, while the receiver detects the radiation that is reflected or emitted by objects within its field of view. When an object enters the sensor's range, it interacts with the emitted infrared radiation, causing changes in intensity, wavelength, or frequency, which the receiver detects.

Types of IR Sensors:

1. **Passive Infrared (PIR) Sensors**: PIR sensors detect changes in infrared radiation emitted by or reflected from objects in their field of view. They are commonly used in motion detection systems, security alarms, and automatic lighting systems.
2. **Active Infrared (AIR) Sensors**: AIR sensors emit infrared radiation and measure the reflection or absorption of the emitted light to detect the presence or distance of objects. They are used in proximity sensors, object detection systems, and distance measurement devices.
3. **Infrared Temperature Sensors**: These sensors measure the temperature of objects by detecting the infrared radiation they emit. They find applications in industrial processes, HVAC systems, medical devices, and thermal imaging cameras.
4. **Infrared Communication Sensors**: IR communication sensors transmit and receive data encoded in infrared light. They are used in remote controls, infrared data transmission systems, and proximity sensing applications.

Applications:

1. Security Systems:

- PIR sensors are widely used in security systems to detect intruders or unauthorized movement in homes, offices, and outdoor areas.
- Infrared cameras and thermal imaging systems provide surveillance and monitoring capabilities for perimeter security, border control, and law enforcement.

2. Consumer Electronics:

- IR remote controls use infrared communication to send commands to televisions, DVD players, air conditioners, and other electronic devices.
- Proximity sensors in smartphones and tablets use IR technology for touchless gesture recognition and automatic screen dimming.

3. Automotive:

- IR sensors assist in driver assistance systems such as parking assistance, blind-spot detection, and collision avoidance.
- Night vision systems in vehicles use IR cameras to improve visibility in low-light conditions and detect pedestrians or obstacles on the road.

4. Industrial Automation:

- Infrared temperature sensors monitor the temperature of machinery, equipment, and industrial processes to ensure optimal performance and safety.
- Proximity sensors and object detection systems in manufacturing plants facilitate

automated material handling, assembly line operations, and quality control.

5. Environmental Monitoring:

- IR sensors are used in weather stations and environmental monitoring systems to measure atmospheric temperature, humidity, and radiation levels.
- Infrared spectroscopy techniques analyze the composition of gases, liquids, and solids in environmental samples for pollution detection and analysis.

Recent Advancements:

1. **Miniaturization and Integration**: Advances in microelectronics and MEMS (Microelectromechanical Systems) technology have enabled the development of smaller, more compact IR sensors suitable for portable devices and wearable gadgets.
2. **Multispectral Imaging**: Multispectral IR cameras combine infrared imaging with other spectral bands (visible, ultraviolet, thermal) to provide enhanced imaging capabilities for medical diagnosis, agricultural monitoring, and scientific research.

3. **Machine Learning and AI Integration**: IR sensor data combined with machine learning algorithms enable predictive maintenance, anomaly detection, and pattern recognition in industrial automation, predictive healthcare, and smart infrastructure systems.

3. **Wireless Connectivity**: IR sensors with wireless communication capabilities, such as Bluetooth Low Energy (BLE) or Wi-Fi, enable remote monitoring, data logging, and real-time analytics in IoT (Internet of Things) applications.

Conclusion:

Infrared sensors play a pivotal role in modern technology, offering versatile solutions for a wide range of applications including security, consumer electronics, automotive, industrial automation, and environmental monitoring. With ongoing advancements in sensor technology, miniaturization, integration, and data analytics, IR sensors continue to evolve, providing enhanced performance, efficiency, and functionality in diverse fields.

3.2 Servo Motor

A servo motor is a rotary actuator that allows for precise control of angular position, velocity, and acceleration. It consists of a motor coupled with a sensor for position feedback, enabling accurate and controlled movement to a desired position..

Servo motors are widely used in various applications that require precise and controlled motion, such as robotics, automation, RC vehicles, 3D printers, CNC machines, and more. They offer high precision, reliability, and flexibility in positioning tasks. Here's how they work.

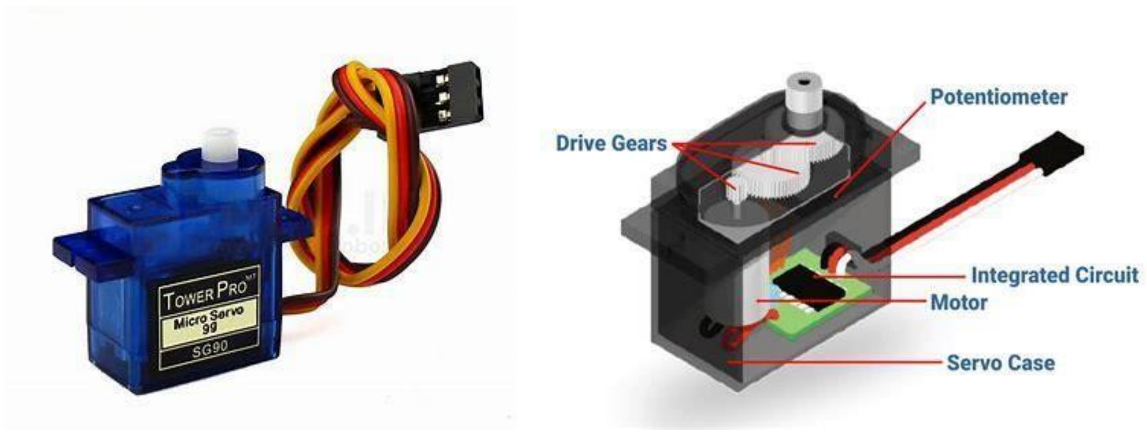


Fig 3.3: Servo Motor [14]

3.2.1 How Does a Servo Motor Work?

The working principle of a servo motor involves a closed-loop control system comprising three main components: the motor, feedback device, and control circuit.

1. **Motor:** The motor in a servo motor system is typically a DC motor or a brushless DC (BLDC) motor. It converts electrical energy into mechanical rotation.
2. **Feedback Device:** A feedback device, usually a potentiometer or an encoder, is attached to the output shaft of the motor. This device provides positional feedback to the control circuit by measuring the actual position of the shaft.
3. **Control Circuit:** The control circuit receives the desired position (setpoint) and compares it with the feedback signal from the sensor. It then adjusts the motor's speed and direction to minimize the error between the setpoint and the actual position. This continuous feedback loop ensures precise positioning of the motor shaft.

Components of a Servo Motor:

- **Motor:** Converts electrical energy into mechanical rotation.
- **Feedback Device:** Provides positional feedback to the control circuit.
- **Control Circuit:** Processes the feedback signal and generates control signals to drive the motor.
- **Gear Train:** Increases torque and reduces the speed of the motor's output shaft.
- **Output Shaft:** Transmits rotational motion to the external mechanism or load.

Operation:

1. The control circuit receives a command signal (setpoint) indicating the desired position or angle.
2. The feedback device continuously monitors the actual position of the motor shaft.
3. The control circuit compares the setpoint with the feedback signal to determine the error.
4. Based on the error, the control circuit adjusts the motor's speed and direction to minimize the error and bring the shaft to the desired position.
5. The process continues in a closed-loop fashion, ensuring accurate and stable positioning of the motor shaft.

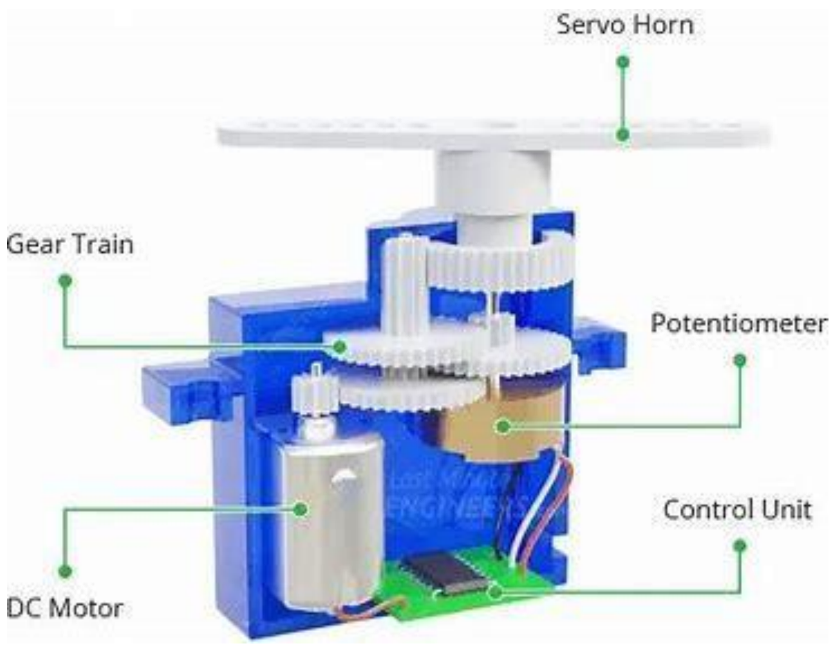


Fig 3.3.1: Servo Motor [15]

1. **Connections**:
   - Connect the power (usually 5V) and ground pins of the servo motor to the respective 5V and GND pins on the Arduino Uno.
   - Connect the signal pin of the servo motor to one of the digital pins on the Arduino Uno.
2. **Code**:
   - Include the Servo library in your Arduino sketch. This library provides functions to control servo motors.
   - Create a Servo object and assign it to the digital pin you've connected the servo to.
   - Use the **attach()** function to attach the servo object to the pin.
   - Use the **write()** function to set the angle of the servo motor. The angle ranges typically from 0 to 180 degrees.

3. **Power Supply**:
   Make sure your power supply (such as the USB connection to your computer or an external power supply) can provide enough current to drive the servo motor

4. **Calibration**:
   Servo motors may vary slightly, so you might need to adjust the angles in the code to match the actual movement of your servo motor.

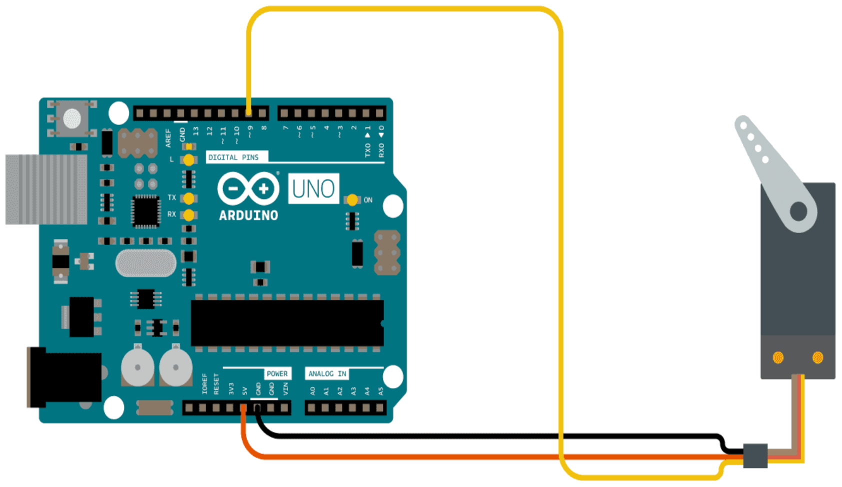


Fig :3.3.2: Circuit Connection Of Servo motor [16]

3.3 Liquid Crystal Display (LCD) display

LCD stands for Liquid Crystal Display. A Liquid Crystal Display (LCD) is a flat- panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers. Unlike traditional displays, LCDs do not emit light directly. Instead, they use a backlight or reflector to produce images in color or monochrome.[25]

Each pixel of an LCD typically consists of a layer of molecules aligned between two transparent electrodes, often made of indium tin oxide (ITO), and two polarizing filters. The axes of transmission of these filters are usually perpendicular to each other. An electric field, induced by a small electric voltage, can change the orientation of molecules in a layer of liquid crystal and thus affect its optical properties2. This process is termed an electro-optical effect, and it forms the basis for LCDs. [25]

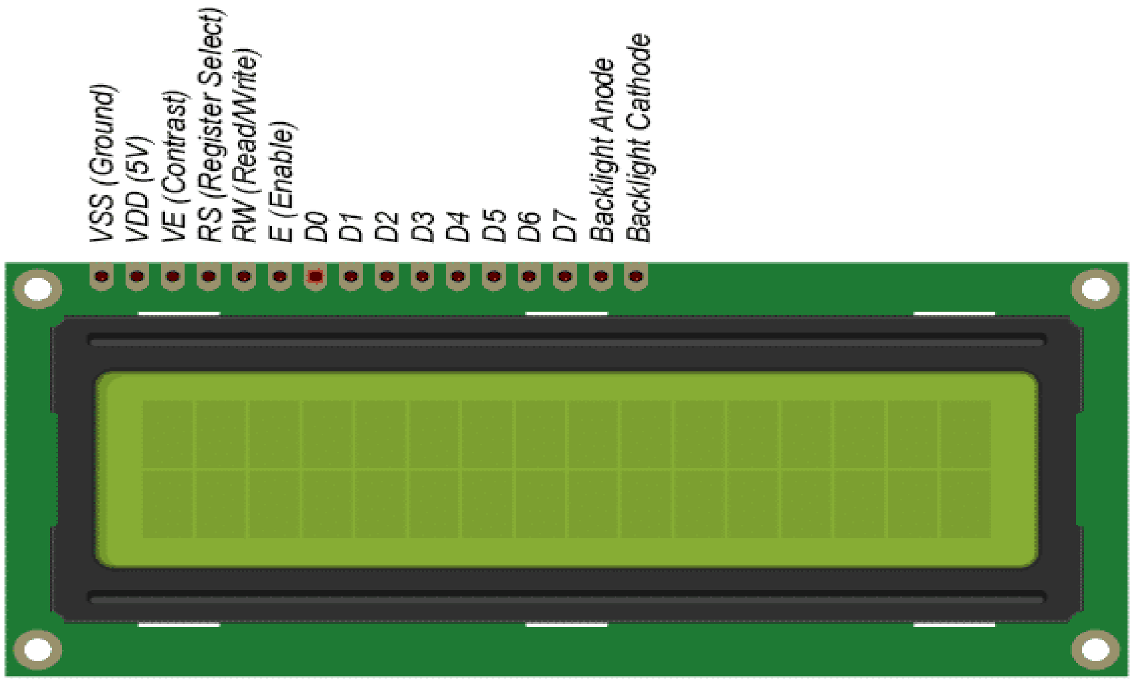


Fig 3.4 : 16*2 LCD [17]

Working

An LCD display works by receiving data and commands from the Arduino board through digital pins. It uses these instructions to control the liquid crystal elements on the display, creating characters or images based on the received input. 16*2 displays mostly depend on multi-segment LEDs. There are different types of displays available in the market with different combinations such as 8*2, 8*1, 16*1, and 10*2, however, the LCD 16*2 is broadly used in devices, DIY circuits.

PINOUT

The LCDs have a parallel interface, meaning that the microcontroller has to manipulate several interface pins at once to control the display. The interface consists of the following pins .

A **register select (RS) pin** that controls where in the LCD's memory you're writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD's controller looks for instructions on what to do next.

A **Read/Write (R/W) pin** that selects reading mode or writing mode.

An **Enable pin** that enables writing to the registers.

8 **data pins (D0 -D7)**. The states of these pins (high or low) are the bits that you're writing. **Contrast, Power Supply, and LED Backlight Pins**: These pins are used to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.

3.6.1 I2C Module

I2C combines the best features of SPI and UARTs. With I2C, you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single, or multiple slaves. This is really useful when you want to have more than one microcontroller logging data to a single memory card or displaying text to a single LCD.

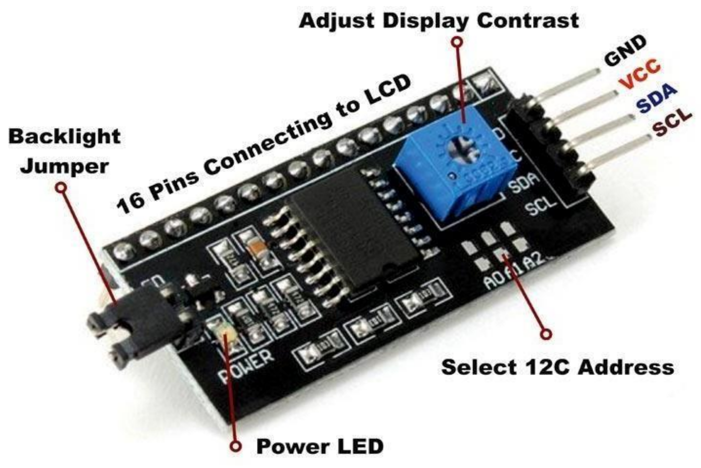


Fig 3.4.1: I2C Serial Interface Adapter Module Pinout [18]

Why use the I2C Module?

Due to limited pin resources in a microcontroller/microprocessor, controlling an LCD panel could be tedious. Serial to Parallel adapters such as the I2C serial interface adapter module with PCF8574 chip makes the work easy with just two pins. The serial interface adapter can be connected to a 16x2 LCD and provides two signal output pins (SDA and SCL)which can be used to communicate with an MCU/MPU. [27]

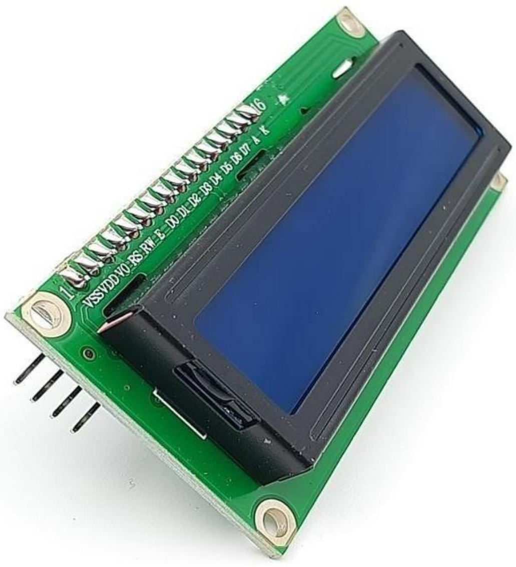


Fig 3.4.2: I2C Module interfaced with 16*2 LCD [19]

3.4 Breadboard

A Breadboard is simply a board for prototyping or building circuits on. It allows you to place components and connections on the board to make circuits without soldering. The holes in the breadboard take care of your connections by physically holding onto parts or wires where you put them and electrically connecting them inside the board. The ease of use and speed are great for learning and quick prototyping of simple circuits. More complex circuits and high frequency circuits are less suited to breadboarding. Breadboard circuits are also not ideal for long term use like circuits built on perfboard (protoboard) or PCB (printed circuit board), but they also don't have the soldering (protoboard), or design and manufacturing costs (PCBs). [29]

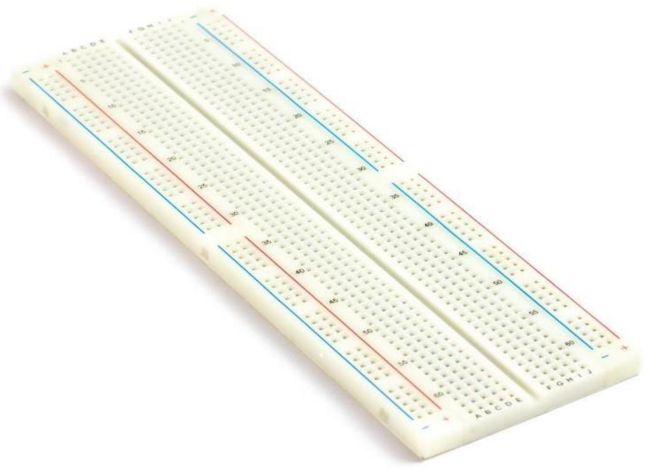


Fig 3.5 : Bread-board [20]

3.5 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. [31]

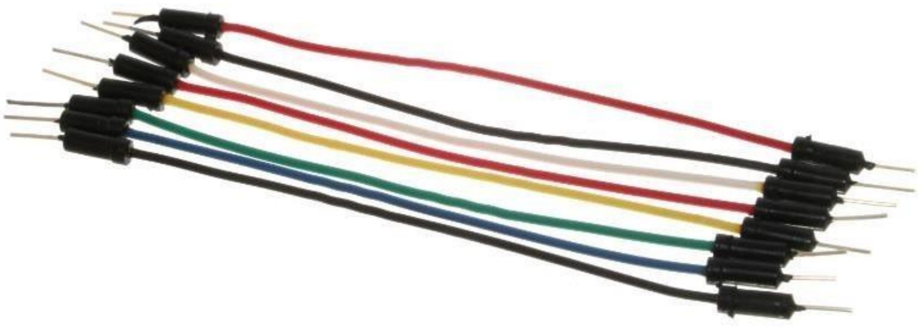
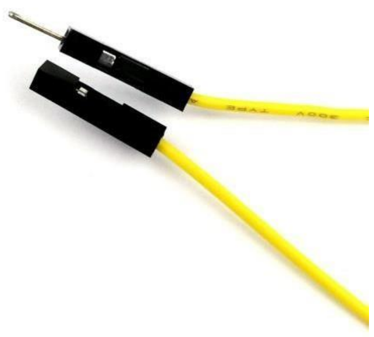


Fig 3.6 : Male-to-male Jumper Wire [21]

Types of jumper wire

- Male-to-male jumper
- Male-to-female jumper
- Female-to-female jumper



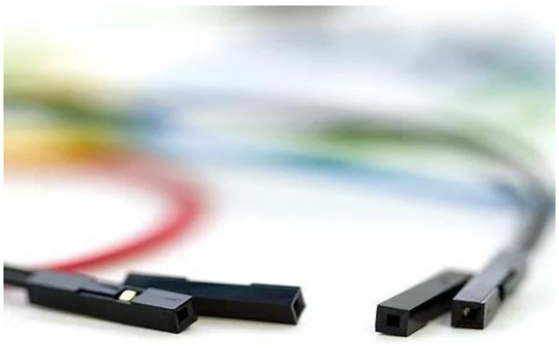
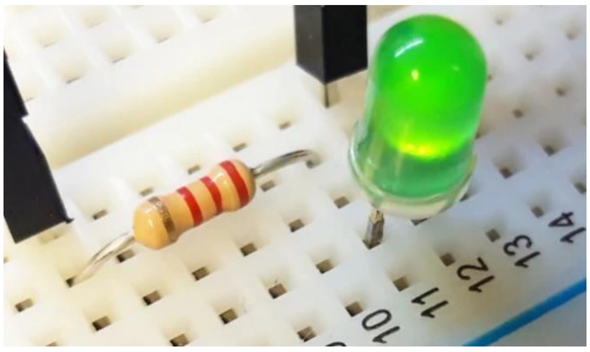


Fig 3.6.1: Male to Female Jumper wire **Fig 3.6.2: Female to Female Jumper wire[22]**

3.7 Switch

[23]

3.8 Resisters & LEDs



[24]

3.9 Battery

[25]

CHAPTER 4

WORKING

4.0 Interfacing :

Procedure to Connect the Entire System:

Gather Components:

- Arduino Uno
- IR sensor module
- Servo motor
- Jumper wires
- Breadboard (optional, but recommended for ease of connection)

1. **Connect IR Sensor**:
    - Connect the VCC pin of the IR sensor to the 5V pin on the Arduino.
    - Connect the GND pin of the IR sensor to the GND pin on the Arduino.
    - Connect the OUT pin of the IR sensor to a digital pin on the Arduino (e.g., pin 2).
2. **Connect Servo Motor**:
    - Connect the red wire (power) of the servo motor to the 5V pin on the Arduino.
    - Connect the brown wire (ground) of the servo motor to the GND pin on the Arduino.
    - Connect the orange or yellow wire (signal) of the servo motor to a PWM pin on the Arduino (e.g., pin 9).
3. **Connect LEDs to Arduino:**

- Connect the longer leg (anode) of each LED to a digital pin on the Arduino (e.g., pin 3, pin 4, etc.).
- Connect the shorter leg (cathode) of each LED to a current limiting resistor (e.g., 220 ohms).

4. **Code Implementation**:
    - Write a code in the Arduino IDE to control the servo motor based on the input from the IR sensor. The code should include:
        - Initialization of the IR sensor pin as INPUT.
        - Initialization of the servo motor pin.
        - Reading the state of the IR sensor.
        - Controlling the servo motor to open/close the parking gate based on the state of the IR sensor.
5. **Upload Code to Arduino Uno**:

- Connect your Arduino Uno to your computer using a USB cable.
- Open the Arduino IDE and write/upload the code to your Arduino Uno.

6. **Testing**:
   - Once the code is uploaded, power up your Arduino Uno.
   - Test the system by placing an object (like a car) in front of the IR sensor. The servo motor should respond accordingly by opening or closing the gate.
7. **Adjustments**:
   - Fine-tune the code and the physical setup as needed to ensure smooth operation of the smart parking system.
8. **Final Setup**:
   - Once everything is working as expected, finalize the setup by securing the components in place and making any necessary adjustments for stability and reliability.

4.1 WORKING PROCESS :

1. **IR Sensor:**
   - IR sensors are used to detect the presence of vehicles in parking spaces. They emit infrared light and measure the reflection to determine if a vehicle is present or not.
   - When a vehicle enters a parking space, it obstructs the infrared beam emitted by the sensor, causing a change in the sensor's output.
   - The Arduino Uno reads the digital output from the IR sensor and processes it to determine the occupancy status of the parking space.
2. **Arduino Uno:**
   - The Arduino Uno acts as the central processing unit of the smart parking system.
   - It receives input from the IR sensors regarding the occupancy status of each parking space.
   - Based on the sensor readings, the Arduino Uno controls the LEDs or other indicators to display the parking status (occupied or available) to users.
3. **Servo Motor:**
   - A servo motor is typically used to control a physical barrier such as a gate or arm in the parking lot.
   - In the context of a smart car parking system, the servo motor can be used to raise or lower a barrier to allow or restrict vehicle access to a parking space.
   - Arduino Uno controls the movement of the servo motor based on the parking status detected by the IR sensor.

- For example, when a parking space is vacant, the servo motor lowers the barrier to allow vehicles to enter. When the space is occupied, the servo motor raises the barrier to prevent access.

4. **LEDs (or Indicators):**
   - LEDs are used as visual indicators to display the parking status to users.
   - When a parking space is occupied, the corresponding LED is turned off or illuminated in red to indicate that the space is unavailable.
   - When a parking space is available, the corresponding LED is turned on or illuminated in green to indicate that the space is vacant.
   - The Arduino Uno controls the state of the LEDs based on the sensor readings received from the IR sensors.

5. **Power Source:**
   - The power source provides electrical power to the Arduino Uno and the components connected to it, such as IR sensors and LEDs.
   - It can be a USB cable connected to a computer or a power adapter plugged into a power outlet.
6. **Breadboard and Jumper Wires:**
   - Breadboards and jumper wires are used to create a prototyping platform for connecting the components together.
   - They facilitate easy and temporary connections between the Arduino Uno, IR sensors, LEDs, and other electronic components.

4.2 Arduino Programming :

CAR PARKING SYSTWM CODE：

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F,16,2);  //Change the HEX address

#include <Servo.h>
```

```
Servo myservo1;

int IR1 = 2;
int IR2 = 4;

int Slot = 4;           //Enter Total number of parking Slots

int flag1 = 0;
int flag2 = 0;

void setup() {
lcd.begin();
lcd.backlight();
pinMode(IR1, INPUT);
pinMode(IR2, INPUT);

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();
}

void loop(){

if(digitalRead (IR1) == LOW && flag1==0){
```

```
if(Slot>0){flag1=1;
if(flag2==0){myservo1.write(0); Slot = Slot-1;}
}else{
lcd.setCursor (0,0);
lcd.print("    SORRY :(    ");
lcd.setCursor (0,1);
lcd.print("  Parking Full  ");
delay (3000);
lcd.clear();
}
}

if(digitalRead (IR2) == LOW && flag2==0){flag2=1;
if(flag1==0){myservo1.write(0); Slot = Slot+1;}
}

if(flag1==1 && flag2==1){
delay (1000);
myservo1.write(100);
flag1=0, flag2=0;
}

lcd.setCursor (0,0);
lcd.print("    WELCOME!    ");
lcd.setCursor (0,1);
lcd.print("Slot Left: ");
lcd.print(Slot);
        }
```

LCD DISPLAY CODE :

```
#include <Wire.h>

void setup()

{

Wire.begin();

Serial.begin(9600);

Serial.println("\nI2C Scanner");

}

void loop()

{

byte error, address;

int Devices;

Serial.println("Scanning...");

Devices = 0;

for(address = 1; address < 127; address++ )

{


Wire.beginTransmission(address);

error = Wire.endTransmission();

if (error == 0)

{

Serial.print("I2C device found at address 0x");

if (address<16)

Serial.print("0");

Serial.print(address,HEX);

Serial.println("  !");

Devices++;

}

else if (error==4)

{

Serial.print("Unknown error at address 0x");
```

```
        if (address<16)

        Serial.print("0");

        Serial.println(address,HEX);

        }

        }

        if (Devices == 0)

        Serial.println("No I2C devices found\n");

        else

        Serial.println("done\n");

        delay(5000);

}
```

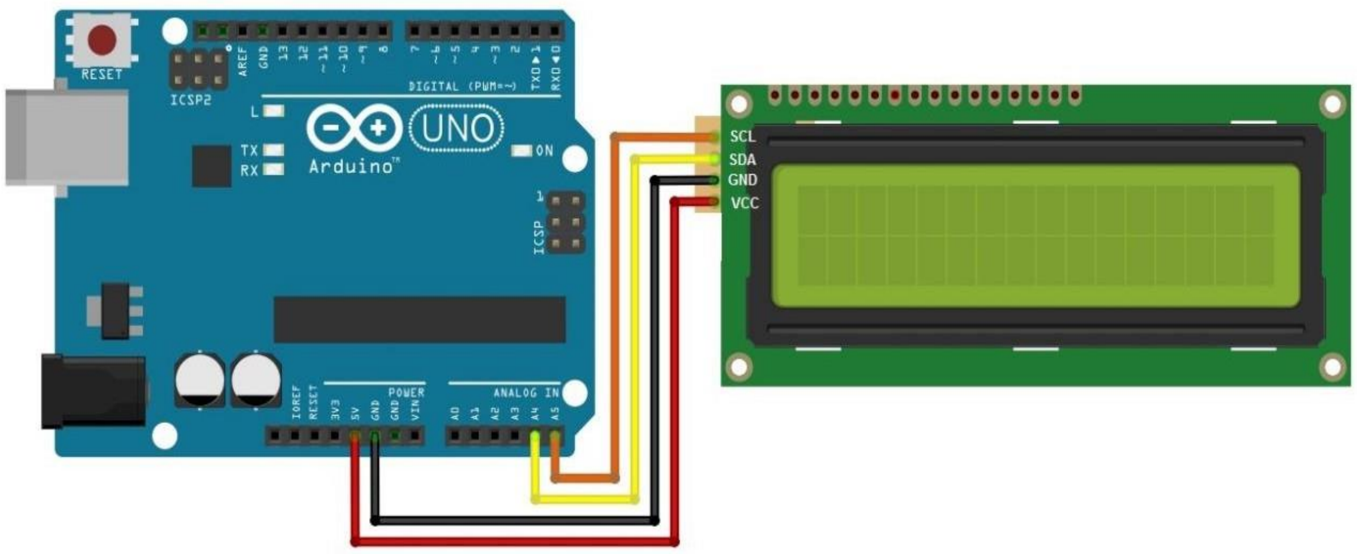


Fig 4.2.1:- Connection Of Lcd Display With Arduino Uno [26]

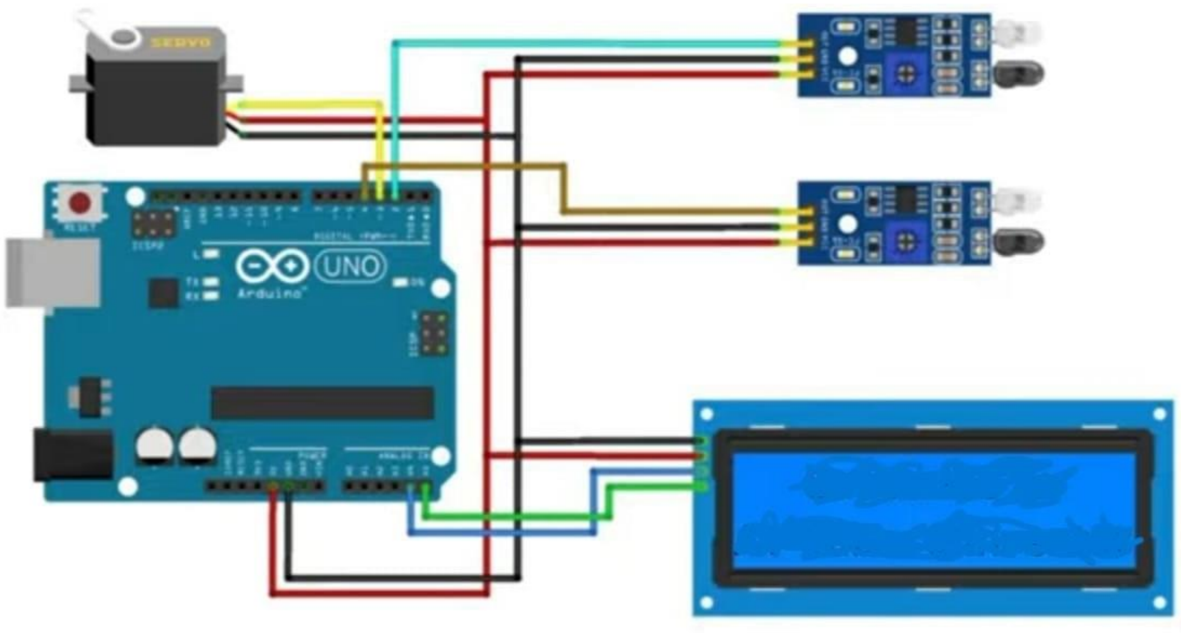


Fig 4.2.2 :- Circuit Diagram [27]

CHAPTER 5

FUTURE WORKS

.

1. **Define Requirements**: Clearly outline what features you want your smart parking system to have. For example, do you want it to detect the presence of a car, indicate available parking spots, and provide a way to reserve a spot.

2. **Hardware Setup**:
   - Arduino Uno: The brain of your system.
   - Ultrasonic Sensors: To detect the presence of cars in parking spots.
   - LED indicators: To show the status of each parking spot (occupied or available).
   - LCD display: To provide information such as available spots or reservation status.
   - Servo motors (optional): To control barriers or gates for reserved spots.

3. **Sensor Integration**:
   - Connect the ultrasonic sensors to the Arduino Uno. These sensors will be placed in each parking spot to detect the presence of cars.

4. **Data Processing**:
   - Write code to process sensor data. When a car is detected, update the status of the corresponding parking spot (occupied or available).

5. **User Interface**:
   - Design a simple user interface using the LCD display and LED indicators. This could show the number of available spots and guide drivers to empty spaces.

6. **Optional Features**:
   - Reservation System: Allow users to reserve parking spots in advance. This might involve integrating additional components like RFID or keypad systems for authentication.

References

[1] International Journal of Computer Applications (0975 – 8887) Volume 169 – No.1, July 2017

[2] Hemant Chaudhary; Prateek Bansal; B. Valarmathi IEEE Xplore

[3] ResearchGate Recent Trends in Information Technology and its Application
Volume 7 Issue 1

[4] Electroduino 10 November 2021

Circuit Diagram

[1] ESP8266 - Wikipedia

[2] ESP8266 Pinout, Pin Configuration, Features, Example Circuit & Datasheet

[3] ESP8266 Setup Tutorial using Arduino (deviceplus.com) (DOI : 10/02/2024)

[4] Quick introduction to the ESP8266 wifi module | ashishware.com (DOI : 10/02/2024)

[5] Liquid-crystal display - Wikipedia

[6] DIY Arduino Uno Mini Limited Edition Weather Station (jameco.com) (DOI :18/02/2024)

[7] LCD 16×2 con módulo I2C incluido | Tettsa - Tienda (DOI : 18/02/2024)

[8] What is a Breadboard? | CircuitBread

[9] Breadboard (830 point) (pimoroni.com) (DOI : 16/03/2024)

[10] What is a Jumper Wire? (sparkfuneducation.com) (DOI : 16/03/2024)

[11] What is a Jumper Wire? (sparkfuneducation.com) (DOI : 16/03/2024)

[12] 7.4Wh 3.7V 2000mAh 18650 Lithium Ion Battery Pack (18650batt.com) (DOI :18/03/2024)

[13]2x SPST On/Off Black Square I/O Rocker Switch Mini Small 12V Automotive/Car/Boat -

[14]Which Resistor Should I Use with my LED? – Kitronik Ltd (DOI : 18/03/2024)

[15]7.4Wh 3.7V 2000mAh 18650 Lithium Ion Battery Pack (18650batt.com) (DOI :18/03/2024)