PROJECT REPORT

On

SMART CAR PARKING SYSTEM USING INTERNET OF THINGS

Submitted for partial fulfilment of award of the degree of

Bachelor of Technology

In

Electronic and Communication Engineering

Submitted by

Rohan Pathriya – 00618002820

Hardik Kapoor – 00418002820

Aditya Verma – 00118007321

Shaheen Parveen– 01018002820

Piyush-00518002820

Under the Guidance of

Dr. Deepti Agarwal
Associate Professor

Affiliated to GGSIP University, New Delhi
Approved by AICTE & Council of Architecture

Department of Electronics and Communication Engineering

DELHI TECHNICAL CAMPUS, GREATER NOIDA

(Affiliated Guru Gobind Singh Indraprastha University, New Delhi)

Session: 2023-24(Even Sem)

DECLARATION BY THE STUDENT

The work contained in this Project Report is original and has been done by us under the guidance of Dr. Deepti Agarwal. The work has not been submitted to any other University or Institute for the award of any other degree or diploma. We have followed the guidelines provided by the university in the preparing the Report. We have confirmed to the norms and guidelines in the ethical code of conduct of the University. Whenever we used materials (data, theoretical analysis, figure and texts) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the reference. Further, we have taken permission from the copywrite owners of the sources, whenever necessary. The plagiarism of the report is 15% i.e. below 20 percent.

Place: Greater Noida	Hardik Kapoor (00418002820)
	Rohan Pathriya (00618002820)
Date:22/05/2024	Aditya Verma (00118007321)
	Shaheen Parveen (01018002820)
	Piyush (00518002820)

CERTIFICATE OF ORIGINALITY

On the basis of the declaration submitted by Hardik Kapoor, Rohan Pathriya, Aditya Verma, Shaheen Parveen, Piyush of B. Tech ECE. We hereby certify that the project titled "***Smart Car Parking System using Internet of Things***" which is submitted to, DELHI TECHNICAL CAMPUS, Greater Noida, in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in ECE, is an original contribution with existing knowledge and faithful record of work carried out by him/them under my guidance and supervision.

To the best of our knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Date: 22/05/2024

Dr. Deepti Agarwal
Associate Professor
Department of ECE
DELHI TECHNICAL CAMPUS
Greater Noida

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all the efforts with success. They have been a guiding light and source of inspiration towards the completion of the project.

With a sage sense of gratitude, we acknowledge various people who directly or indirectly contributed to the development of the project and influenced our thinking, behavior, and acts during the process. We would like to express our sincere gratitude to our project guide "**Dr. Deepti Agarwal"** for giving us the opportunity to work on this topic. It would never be possible for us to take this project to this level without her innovative ideas, relentless support and encouragement.

Finally, we would like to take this opportunity to thank our families for their support all through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

Hardik Kapoor (00418002820)
Rohan Pathriya (00618002820)
Aditya Verma (00118007321)
Shaheen Parveen (01018002820)
Piyush (00518002820)

CONSENT FORM

This is to certify that we, Hardik Kapoor, Rohan Pathriya, Aditya Verma and Shaheen Parveen, Piyush student of B. Tech (ECE) of batch 2020-2024 presently in the VIII Semester at DELHI TECHNICAL CAMPUS, Greater Noida give my consent to include all my personal details Hardik Kapoor (00418002820) , Rohan Pathriya (00618002820) , Aditya Verma (00118007321), Shaheen Parveen (01018002820) and Piyush (00518002820) for all accreditation purposes.

Place: Greater Noida	Hardik Kapoor (00418002820)
	Rohan Pathriya (00618002820)
Date:22/05/2024	Aditya Verma (00118007321)
	Shaheen Parveen (01018002820)
	Piyush (00518002820)

DECLARATION FORM (Health, Safety & Plagiarism)

We Hardik Kapoor, Rohan Pathriya, Aditya Verma and Shaheen Parveen and Piyush student of B. Tech 4th Year Electronics and Communication Engineering, Enrollment No. 00418002820, 00618002820, 00118007321, 01018002820 and 00518002820 batch 2020-2024, Department of Electronics and Communication Engineering, Delhi Technical Campus, GGSIP University, New Delhi, hereby declare that I havegone through project guidelines including policy on health and safety, policy on plagiarism etc.

Date: 22/05/2024 Student's Signature

Place: Greater Noida

LIST OF FIGURES

TABLE OF CONTENTS

ABSTRACT

In current times, the concept of smart cities has gained much popularity. Consistent efforts are being made to maximize the productivity and reliability of urban infrastructures. Parking is costly and limited in almost every major city of India. Innovative parking systems for meeting near term parking demand are needed. Due to rapid increase in vehicle density especially during the peak hours of the day, it is a tedious task for the public to find a secured parking space for their vehicles. This proposes a novel, secured, efficient and intelligent parking system based on Arduino and sensors. The proposed smart parking system includes of an onsite slot module development that will monitor the availability of each parking slot. Due to number of vehicles on road, causes traffic problems our smart parking system helps us to find parking spaces in parking lot with the implementation of smart parking system patrons can easily locate vacant parking spaces subsequently , the various sensor systems used in developing the systems and our car parking system reducing congestion in cities and our proposed models such that no man is needed to park the cars and no receipt system problem in our project has the automatic receipt system and this project helps in reducing the average time of users for that they wait for the parking of car all such systems will coordinate they do not require to wait we have automatic door opening and closing system using Servo motor by using push buttons such that car enter and exit though entry and exit parking system we have created this project to 7 cars we can increase the parking spaces such that many cars are parked in our parking lot and we have used IR sensors for visualization . Further modifications include a mobile application that allows the user to track the availability of parking space in the required locality. This smart parking can increase the economy by lowering the labour cost, reducing the fuel consumption.

CHAPTER I
INTRODUCTION

1.1 Background

The proliferation of urbanization, coupled with the exponential growth in the number of vehicles worldwide, has intensified the challenges associated with urban mobility. As cities grapple with issues of traffic congestion, pollution, and inefficient parking management, there is an increasing recognition of the need for innovative solutions to address these pressing concerns. The Internet of Things (IoT) has emerged as a transformative technology with the potential to revolutionize various aspects of urban living, including parking systems. In India, rapid urbanization has led to a surge in the number of vehicles, exacerbating the already strained urban infrastructure. Major cities such as Delhi, Mumbai, and Bangalore face severe traffic congestion, and finding parking spaces has become a daunting task for motorists. Recognizing the need for smart urban solutions, India has been exploring the integration of Iota technologies into its urban infrastructure, with a specific focus on parking management. On a global scale, several developed countries have already made significant strides in implementing Iota based car parking systems. Cities like Singapore, Barcelona, and San Francisco have successfully deployed smart parking solutions that leverage Iota devices to monitor parking space availability in real time. These systems utilize sensors embedded in parking spaces, connected to a centralized platform that communicates with drivers through mobile applications, guiding them to available parking spots and optimizing overall parking space utilization.

1.2 Problem Statement

The problem at hand revolves around the inadequacies of traditional parking management systems in the face of escalating number of vehicles on the road. Existing systems often lead to traffic congestion, increased carbon emissions, and frustrated drivers spending significant time searching for parking spaces. In India and across the globe, there is a critical need to address these challenges and develop innovative solutions that can alleviate the strain on cities infrastructure. The current lack of efficient, real-time parking information exacerbates these issues.

1.3 Proposed Solution

For the above problem statement, the solution we propose here is to design and implementan Iota- based car parking system that leverages sensor technology provide accurate, up-to-the- minute parking slot information, so that driver can see if there is any empty parkingspot available or not which contributing to a more sustainable, accessible, and efficient urban mobility landscape.

1.4 Objective

The primary objective of the "Smart Car Parking System" project is to revolutionize conventional parking infrastructure through the implementation of advanced technologies, with a focus on Internet of Things (Iota) integration. This enables real- timemonitoring of parking spaces, reducing congestion and improving traffic flow. Provide administrators with insights into parking patterns for better resource management. Leverage Iota technology to create a cost-effective and scalable parking system. Reduce the need for extensive physical infrastructure while maximizing operational efficiency. Mitigate traffic congestion and reduce unnecessary fuel consumption by guiding users toavailable parking spots directly. Contribute to a reduction in carbon dioxide emissions associated with circling for parking.

1.5 Block Diagram

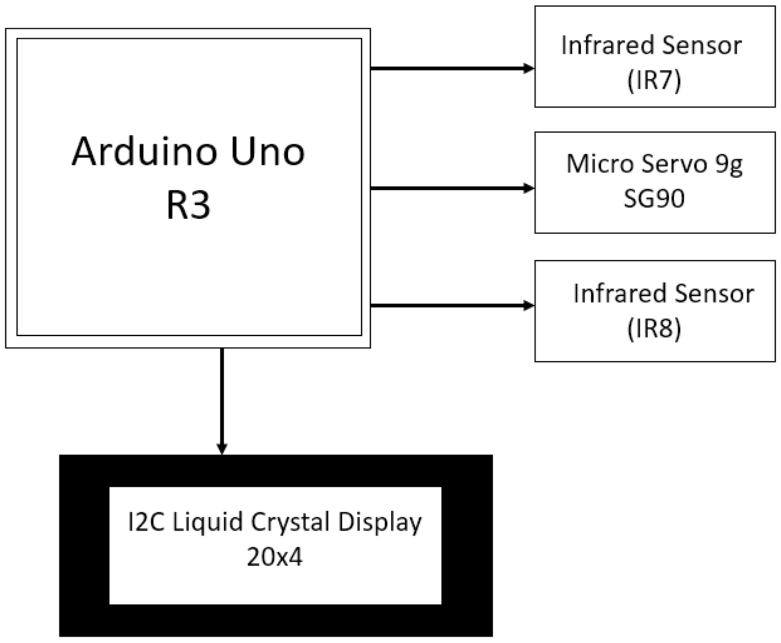


Fig 1.1 Block Diagram of Arduino Uno System

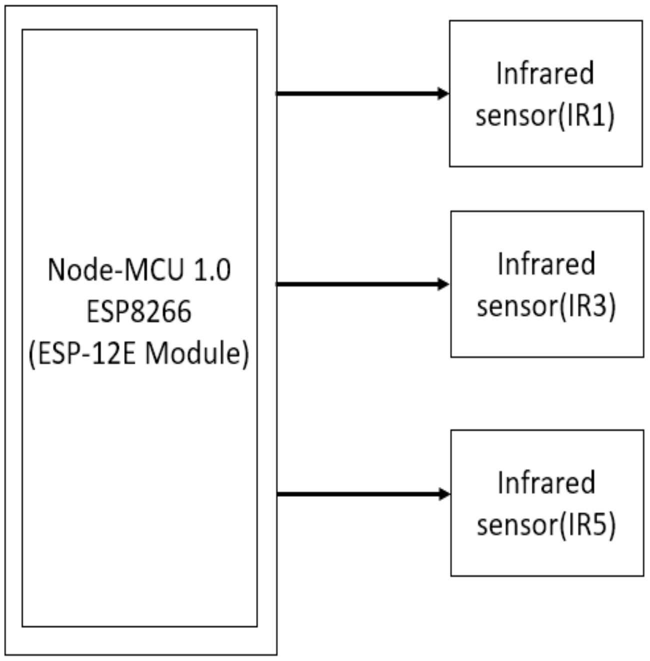


Fig 1.2 Block Diagram Node-MCU System with IR sensor

1.6 Scope of Projects

The scope of a Smart Car Parking System is expansive, encompassing various aspects that contribute to the efficiency, convenience, and sustainability of urban mobility. Hereare key dimensions of the scope for a Smart Car Parking System:

- Optimized parking – Users find the best spot available, saving time, resources and effort. The parking lot fills up efficiently and space can be utilized properly by commercial and corporate entities.
- Reduced pollution – Searching for parking burns around one million barrels of oil a day. An optimal parking solution will significantly decrease driving time, thus lowering the amount of daily vehicle emissions and ultimately reducing the global environmental footprint.
- Real-Time Data and Trend Insight – Over time, a smart parking solution can produce datathat uncovers correlations and trends of users and lots. These trends can prove to be invaluable to lot owners as to how to make adjustments and improvements to drivers.
- Decreased Management Costs – More automation and less manual activity saves on laborcost and resource exhaustion.

CHAPTER 2

LITERATURE REVIEW

The author of this paper [1] face a problem for looking for a parking spot can be quite a hassle, taking up a lot of valuable time and energy, and leading to significant financial expenses. This is especially true for individuals who are constantly feeling the pressure to be punctual. In order to address this issue, smart cities have implemented various cutting-edge technologies to effectively manage and optimize resources. As a response to this challenge, we have developed a smart parking management system (SPMS) as a modern solution that can efficiently handle parking concerns, ultimately saving users precious time, effort, and money. Instead of relying on the uncertainty of finding parking at a physical location, it is much more convenient to search for or reserve available parking spaces online in advance.

This paper [2] shows the huge proliferation in the number of vehicles on the road along with mismanagement of the available parking space has created parking related problems as well as increased the traffic congestion in urban areas. Thus, it is required to develop an automated smart parking management system that would not only help a driver to locate a suitable parking space for his/her vehicle, but also it would reduce fuel consumptions well as air pollution. It has been found that a drivers search for a suitable parking facility takes almost 15 minutes which increases the fuel consumption by the vehicle, traffic congestionand air pollution. A significant amount of research works exist in the area of design and development of smart parking system. Various features of smart parking system are listed below. • Inquiry on availability of parking space and reservation of parking lot. Real-time parking navigation and route guidance Vehicle occupancy detection and management of parking lots. Most of the smart parking systems (SPS) proposed in literature over the past few years provides solution to the design of parking availability information system, parking reservation system, occupancy detection and management of parking lot, real-time navigation within the parking facility etc. However, very few works have paid attention to the real time detection of improper parking and automatic collection of parking charges. Thus, this paper presents an internet-of thing (Iota) based E-parking system that employs an integrated component called parking meter (PM) to address the following issues.

• Real-time detection of improper parking.

• Estimation of each vehicles duration of parking lotusage.

• Automatic collection of parking charges The E-parking system proposed in thispaper also provides city-wide smart parking management solution via providing parking facility availability information and parking lot reservation system and it is named as parking meter (PM) based E-parking (PM-EP).

The view of author of this paper [3] is about India is getting motorized i.e. the rate of private vehicles is more as compared to public transports. As the rate of people owningtheir vehicles increases, the need of parking slots to park vehicles also increases. But currently the scenario is that there are not sufficient parking slots available or there is also possibility that people are not now aware about the legal parking slots available in their locality. This situation leads to the unnecessary crowding of vehicles on the road and also results in inconveniency of people walking on the road. To overcome above problems. We are proposing the solution in the form of a multilingual android application which will be helpful for the people to find their parking slots digitally. By digitally wemean that this particular system will assign the parking slot based on the current location of the user and the parking slot which the user wants according to his/her ease. Ease in terms of finding the exact slot. The payments can be done digitally or through vending machines. The end user can register and login with his/her account which will help the system to find the location and displaying the nearest parking area and nearest parking slot, whether it is available or not. If not then it will direct user to the next nearest slot and so on. The existing system comprises of both traditional and application based approach for parking. If we talk about the traditional approach it utilizes manual method of parking i.e user has to find the spot for parking by traveling to far distances and paying extra money. An application based approach consist of the applications which provides the parking slots for the particular locality.

This paper [4] talks about transportation is the key-success for any of the country. Nowaday, many people have options to use their own vehicle for travelling. This will surelyincrease the demand in trading but one of the problems created by road traffic is "parking". To park all these vehicles in the major metro cities is quite tedious and difficult task and it became problematic to park vehicles. Lot of research and development is being done all over the world to implement better and smarter parking management mechanisms. The current smart parking systems or Wireless Sensors Network Parking requires the combination of wireless sensor networks module, Embedded web-server, Central Web Server. Sensor networks make use of Infrared (IR) Sensor nodes to check the parking slot state and send this information to embedded web-server. It thereby displays the information on a LED screen with which the user can check for empty vehicle slots. These systems not guide the users to reach to the parking lot. If the slot isnot available at that time than drivers will start searching for another parking zone so that this process is time consuming and will increase the traffic congestion. This paperproposes a Reservation- based Smart Parking System for avoiding the traffic problemsthat provides the pre- booking of slots through the use of the mobile application. This application is expected to provide an efficient and cost- effective solution to the vehicle parking problems. Application must be installed in the user's mobile. Unlike the existing system, our idea is to use client- server architecture where client request for the reservation of slots and server responds with the slots which are available at that time. Oursystem is that the user has an option to go for the parking area according to his/her convenience. The advantage of this will greatly reduce the time taken by the vehicle to search for a parking area. Advanced payment modules are also included like e- wallet, debit card, credit card from which the user can pay. Penalty will be added on late exit as well as an over use of the slot after user specified entry and exit time. The refund will be given on cancelation of parking slot and early exit. The supervisor is required to monitor the area. Many of the vehicles parking facilities are unable to cope with the influx of vehicles on roads and parking area. The current smart parking systems or Wireless Sensors Network Parking requires the combination of wireless sensor networks module, Embedded web-server, Central Web- Server. Sensor networks makeuse of Infrared (IR) Sensor nodes to check the parking slot state and send this information to embedded web-server. It thereby displays the information on a LED screen with which the user can check for empty vehicle slots. Also image capturing devices are used for continuously clicking pictures of parking area to ensure empty slotswhich results in high power consumption and also high maintenance cost is required. There are some systems in the market like the smart parking services which are based on the wireless sensor networks which uses wireless sensors to effectively find the available parking space. But to use this system, additional hardware needs to be installed in the car which is not feasible. Finding a parking slot in a congested city is very hard. In many cases people go to a parking station and they find it full and there is no space available for parking. Then in search of parking space they have to again roam with their vehicle to find available parking.

Today, the parking industry is being transformed by new technologies that are allowing cities to reduce rates of congestion significantly. Sensor networks that sense vehicle occupancy are providing the basic intelligence behind smart parking systems. Thanks to the Smart Parking technology, it is now possible in real-time the location of free parking spaces and to help drivers to get to their ultimate destination. This paper [5] shows a variety type of vehicle detectors has been used in parking information acquisition. These vehicle detectors mainly include the inductive loop, acoustic sensor, infrared sensor, or ultrasonic sensor. System using video camera sensor technologies have been proposed to collect the information in vehicle parking field. However, video camera sensor is vulnerable to bad weather and night time operation. Furthermore, it is expensive, and can generate a large amount of data that can be difficult to transmit in a wireless network. The magneto-resistive based detection systems combined with a wireless area network are the most popular technique due to their high accuracy. Yet, this type of sensor is facing different issues, i.e. it can be divided by electromagnetic interference, which affects the accuracy, the reading from sensor needs to be collected constantly which will result in wearing out the battery. To extend the battery lifetime and increase the vehicle detection accuracy, a parking sensor system has been proposed. While power management technique has been implemented to optimize energy consumption, high occupancy monitoring accuracy is Achieved using two-fold sensing approach. It is a sequence of darkness and Signal Strength Indicator (RSSI) measurement based techniques. The wireless sensors are still intrusive, they are embedded in the pavement, or taped to the surface of each individual parking lot. Existing sensors, such as ground based parking sensors costs up to $200 per parking lot. As consequence, smart-parking technology using wireless sensors for outdoor parking is costly due to the large number of sensors units required to cover the entire parking lot. Although, parking occupancy monitoring systems have made a significant progress, smart parking payments rarely studied in smart parking research. Yet, there are companies working on the patents of parking systems for payments. A first approach consists in using a camera or an RFID transceiver for vehicle detection and identification. A limitation of this solution lies in that the system is complex and its implementation is expensive when a detection device is installed on each parking lot. Furthermore, when only RFID transceiver is used for vehicle detection and identification, the system can be bedeviled by electromagnetic interference, which affects the accuracy. Moreover, this system is designed to detect vehicle when entering a parking and seek payment, whereas information on vacant parking lots is not provided. A technique for monitoring vehicle parking using one camera to record the entrance of a vehicle and a second camera to record the vehicle leaving the parking has been proposed. Moreover, in a system and method for obtaining and displaying information on vacant parking space is described. When a user occupies a parking space designated with an individual ID, he enters this ID into a parking meter or via as mart phone mobile app., and pays the parking fees. The database processes thereceived data and changes the status of the parking space with its ID from unpaid to paid. These data are used as information on the occupation of a parking space. In this paper, we propose a smart sensor system allowing outdoor parking monitoring and payment without requiring any user/driver interaction. It will be deployed without having to install new components on each parking lot. The proposed sensor has benefits in terms of detection and payment reliability, and reduced expense by reducing the system complexity and installation, and extending batteries lifetime through the reduction of the system power consumption.

The paper[5] describes the development of an Internet of Things (IoT) based system designed to address parking issues in urban areas. The system utilizes sensors and microcontrollers to monitor parking space availability in real-time. It enables users to find and reserve parking spots via a mobile application, reducing the time spent searching for parking and helping to alleviate traffic congestion. This system aims to increase efficiency, convenience, and security in parking management.

The review paper[6] on Smart Parking Systems discusses various technologies and methodologies used to enhance parking efficiency in urban areas. It highlights the integration of IoT, machine learning, and deep learning to monitor and manage parking spaces. The paper reviews different systems, including sensor-based and camera-based solutions that utilize data analytics to provide real-time information on parking availability. It also explores the potential benefits of smart parking systems, such as reduced traffic congestion and improved user convenience, by facilitating efficient parking space utilization and reservation capabilities(IJERT).

This is an article[7] about a smart vehicle parking system using RFID and IoT . It discusses the problems of finding parking spaces and traffic congestion in metropolitan cities. The system uses RFID tags to identify cars and track their location. This allows drivers to find available parking spots more easily and reduces wasted time searching for parking [1]. The system also helps to reduce traffic congestion and pollution by minimizing the amount of time cars spend driving around looking for parking.

This is an article[8] about a smart parking system using ultrasonic sensors. It discusses the problems of finding parking spaces and wasting time looking for them. The system uses ultrasonic sensors to detect available parking spots and relays this information to a dashboard. Users can view this information on a mobile app or website. The system can also be used to book parking spots.

The paper[9] discusses an automated system that uses Infrared (IR) sensors to detect vehicles and manage the number of vehicles moving in and out of parking structures[1]. The system provides a visual output indicating available parking spaces, using a row of LED lights. The two main colors used are red and yellow, indicating occupied and free spaces respectively. This system aims to reduce the search time for parking spaces in large facilities such as malls, multi-storey parking structures, and IT hubs[1]. It is designed to be simple and cost-effective, not requiring heavy lines of code or expensive equipment. The authors argue that this smart car parking system can help address the growing problem of vehicle management in densely populated areas, making parking a less arduous task.

This is an article about a smart parking system using IoT technology [10]. It discusses the problems with conventional parking systems, which require a lot of manpower and waste time. The proposed system uses sensors and controllers to find available parking spaces. Lights are used to direct drivers to empty spaces. This system reduces energy use and improves the aesthetics of the parking area.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 Functional Requirement

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs/or conditions. These may include calculations, data manipulation and processing and other specific functionality. In these systems following are the functional requirements:-

- The application should not display in-appropriate message for valid condition.
- The application must not stop working when kept running for even longtime.
- The application should process information for any kind of input case.
- The application should generate the output for a given input test case.

3.2 Non-Functional Requirement

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. Given below are the non-functional requirements:

- Product requirements
- Organizational requirements
- Basic operational requirements

3.3 Hardware Requirement

- ARDUINO UNO
- INFRARED SENSORS
- 20*4 LCD DISPLAY
- SERVO MOTOR
- Node-MCU 8266

3.4 Software Specification

- ARDUINO IDE
- C LANGUAGE
- VS Code

CHAPTER 4

DESCRIPTION OF HARDWARE AND SOFTWARE COMPONENT USED

4.1. Arduino Uno

Arduino Uno is user-friendly and designed for beginners. It has a simple and easy-to-understand interface, making it suitable for individuals who may not have extensive programming or electronics experience. Although Arduino Uno is mainly used for prototyping, its ecosystem also includes other Arduino boards with different features and functions. This makes it possible to increase capacity when moving from a standard parking system to a smarter parking system. Arduino Uno is versatile and can be easily expanded with additional components and expansion boards. This change is important for smart parking systems because they will need to communicate with various sensors (such as ultrasonic sensors for distance measurement), actuators (such as servo motors for control issues) and Mode integration. Arduino Uno is compatible with many sensors and actuators. This relationship facilitates the integration of different products into smart stations. Many libraries are also available that facilitate interaction with various devices. Although Arduino Uno is primarily used for prototyping, its ecosystem also includes other Arduino boards with many features and capabilities.

4.1.2 Features

- **ATMega328P Processor**
  - **Memory**
    - AVR CPU at up to 16 MHz
    - 32KB Flash
    - 2KB SRAM
    - 1KB EEPROM
  - **Security**
    - Power On Reset (POR)
    - Brown Out Detection (BOD)

- **Peripherals**
  - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
  - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
  - 1x USART with fractional baud rate generator and start-of-frame detection
  - 1x controller/peripheral Serial Peripheral Interface (SPI)
  - 1x Dual mode controller/peripheral I2C
  - 1x Analog Comparator (AC) with a scalable reference input
  - Watchdog Timer with separate on-chip oscillator
  - Six PWM channels
  - Interrupt and wake-up on pin change

- **ATMega16U2 Processor**
  - 8-bit AVR® RISC-based microcontroller

- **Memory**
  - 16 KB ISP Flash
  - 512B EEPROM
  - 512B SRAM
  - Debug WIRE interface for on-chip debugging and programming
- **Power**
  - 2.7-5.5 volts

4.1.3 Functional Overview

4.1.3.1 Board Topology

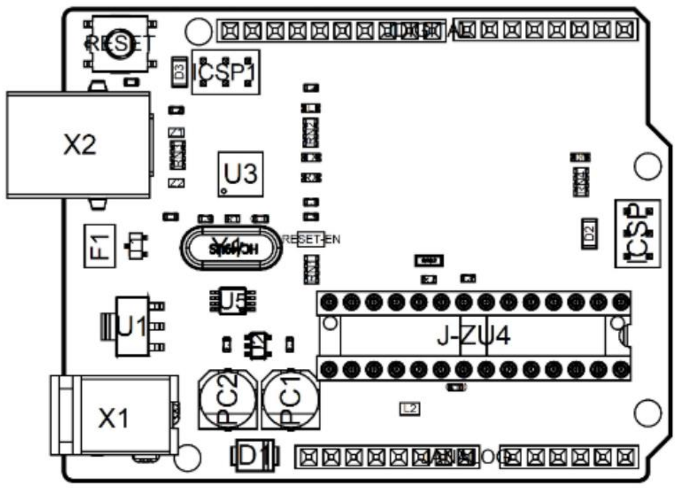


Fig 4.1 Arduino Uno Board Topology

Table 4.1 Parts Description

Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		

4.1.3.2 Arduino Uno R3 – Pin Layout

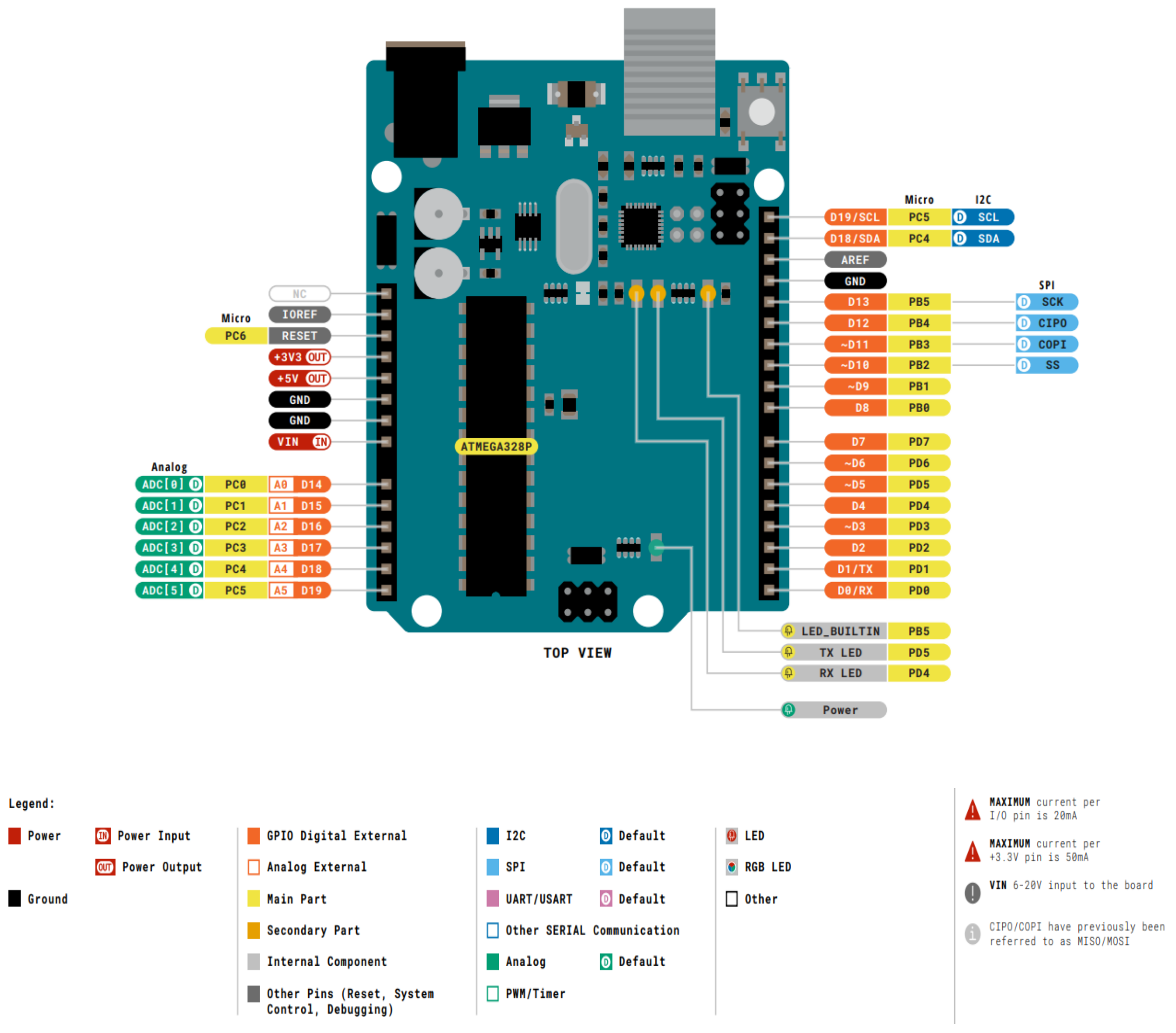


Fig 4.2 Pin layout of Arduino Uno

4.2 Node-MCU ESP8266

- Node-MCU is an open-source LUA based firmware developed for the ESP8266 Wi-Fi chip. By exploring functionality with the ESP8266 chip, Node-MCU firmware comes with the ESP8266Development board/kit i.e. Node-MCU Development board.
- Since Node-MCU is an open-source platform, its hardware design is open for edit. Node-MCU Dev Kit/board consist of ESP8266 Wi-Fi enabled chip. The ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer to the ESP8266 Wi-Fi Module. There is Version2 (V2) available for Node-MCU Dev Kit i.e. **Node-MCU Development Board v1.0 (Version2)**, which usually comes in black colored PCB.
- It supports serial communication protocols i.e. UART, SPI, I2C, etc.
- Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards, etc.

4.2.1 Tech specification

- Model: ESP8266-12E
- Node-MCU Model: Amica
- Wireless Standard: 802.11 b/g/n
- Frequency range: 2.4 GHz - 2.5 GHz (2400M-2483.5M)
- Wi-Fi mode: Station / SoftAP / SoftAP+station
- Stack: Integrated TCP/IP
- Output power: 19.5dBm in 802.11b mode
- Data interface: UART / HSPI / I2C / I2S / Ir
- Remote Control GPIO / PWM
- Supports protection mode: WPA / WP
- Encryption: WEP / TKIP / AES
- Power supply: from 4.5 VDC to 9 VDC (VIN) or via micro USB connector
- Consumption: with continuous Wi-Fi transmission about 70 mA (200 mA MAX) - in standby < 200µA
- Operating temperature: from -40°C to +125°C
- Dimensions (mm): 58×31.20×13

- Weight: 10 grams

4.2.2 Pin Definition:

4.2.2.1 Pin Layout

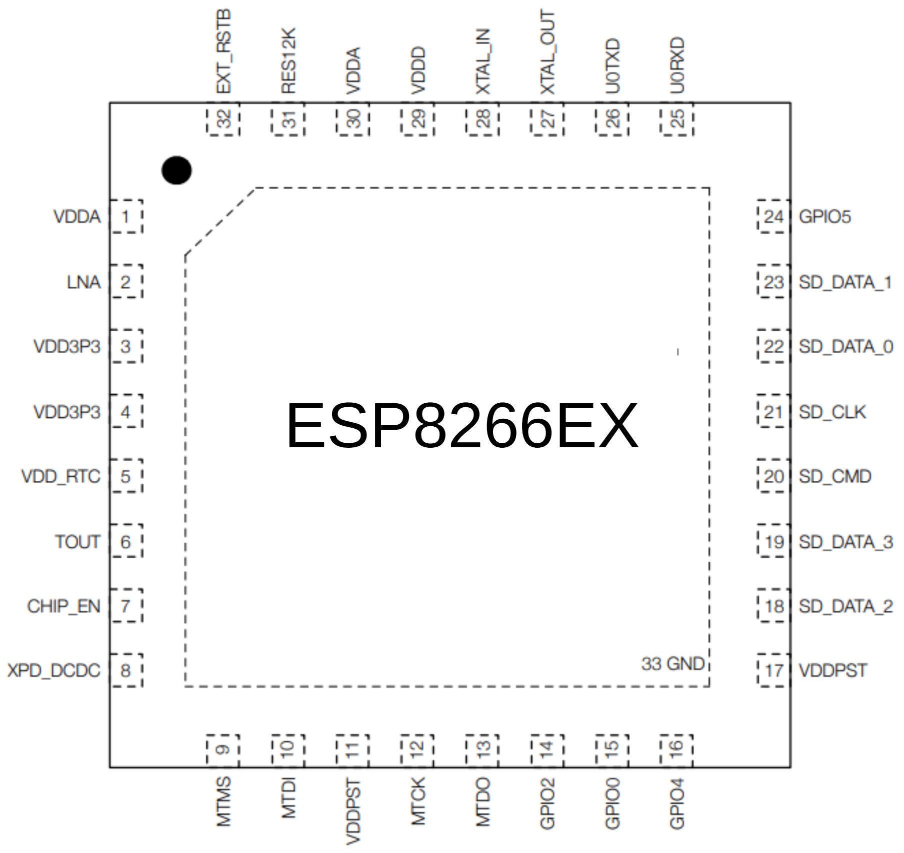


Fig 4.3 ESP8266XX Pin Layout

4.2.2.2 Pin Description:-

Table 4.2 ESP8266EX Pin Definitions

Pin	Name	Type	Function
1	VDDA	P	Analog Power 2.5 V ~ 3.6 V
2	LNA	I/O	RF antenna interface Chip output impedance = 39 + j6 Ω. It is suggested to retain the π-type matching network to match the antenna.
3	VDD3P3	P	Amplifier Power 2.5 V ~ 3.6 V

Pin	Name	Type	Function
4	VDD3P3	P	Amplifier Power 2.5 V ~ 3.6 V
5	VDD_RTC	P	NC (1.1 V)
6	TOUT	I	ADC pin. It can be used to test the power-supply voltage of VDD3P3 (Pin3 and Pin4) and the input power voltage of TOUT (Pin 6). However, these two functions cannot be used simultaneously.
7	CHIP_EN	I	Chip Enable High: On, chip works properly Low: Off, small current consumed
8	XPD_DCDC	I/O	Deep-sleep wakeup (need to be connected to EXT_RSTB); GPIO16
9	MTMS	I/O	GPIO 14; HSPI_CLK
10	MTDI	I/O	GPIO 12; HSPI_MISO
11	VDDPST	P	Digital/IO Power Supply (1.8 V ~ 3.6 V)
12	MTCK	I/O	GPIO 13; HSPI_MOSI; UART0_CTS
13	MTDO	I/O	GPIO 15; HSPI_CS; UART0_RTS
14	GPIO2	I/O	UART TX during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPI_CS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/IO Power Supply (1.8 V ~ 3.6 V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R: 20 Ω); SPIHD; HSPIHD; GPIO9
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R: 200 Ω); SPIWP; HSPIWP; GPIO10
20	SDIO_CMD	I/O	Connect to SD_CMD (Series R: 200 Ω); SPI_CS0; GPIO11
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R: 200 Ω); SPI_CLK; GPIO6
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R: 200 Ω); SPI_MISO; GPIO7
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R: 200 Ω); SPI_MOSI; GPIO8
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART TX during flash programming; GPIO1; SPI_CS1
27	XTAL_OUT	I/O	Connect to crystal oscillator output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal oscillator input
29	VDDD	P	Analog Power 2.5 V ~ 3.6 V
30	VDDA	P	Analog Power 2.5 V ~ 3.6 V

4.2.3. Applications

- Home appliances
- Home automation
- Smart plugs and lights
- Industrial wireless control
- Baby monitors
- IP cameras
- Sensor networks
- Wearable electronics
- Wi-Fi location-aware devices
- Security ID tags
- Wi-Fi position system beacons

4.2.4 Node-MCU 1.0 ESP8266 (ESP-12E Module) Pin Diagram

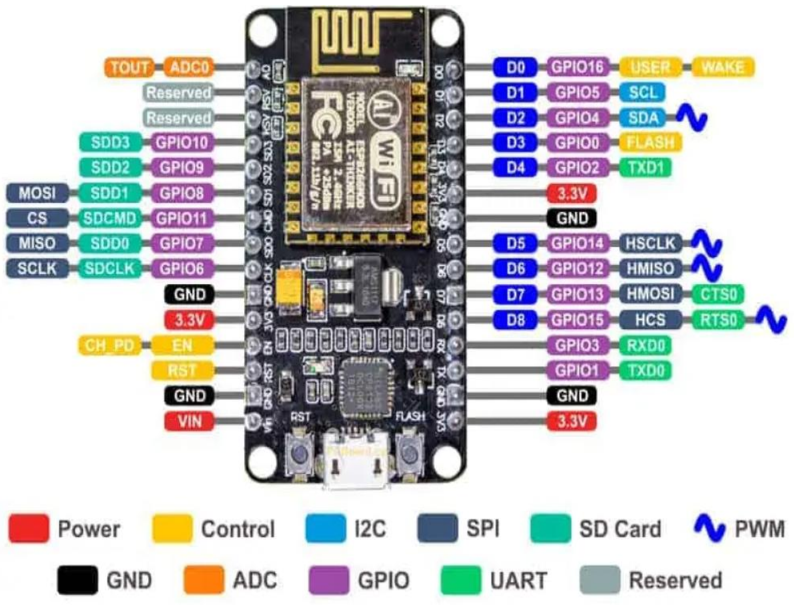


Fig 4.4 Node-MCU ESP8266 Pin Diagram

4.3 Infrared Sensor

An infrared (IR) sensor is an electronic device that measures and detects infrared radiation in its surrounding environment. Infrared radiation was accidentally discovered by an astronomer named William Herschel in 1800. While measuring the temperature of each color of light (separated by a prism), he noticed that the temperature just beyond the red light was highest. IR is invisible to the human eye, as its wavelength is longer than that of visible light (though it's still on the same electromagnetic Wearer using three IR detect sensor in our project ,one IR detect sensor is used to sense the vehicle near the parking sensor and other two IR detect sensor is used to send data to the Arduino Uno which is the brain of our system whether a vehicles parked in that slot or is unmarked .

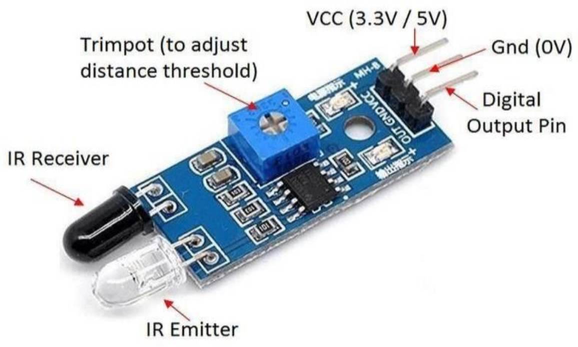


Fig 4.5 IR Sensor

4.4 Servo Motor

A servomotor is a rotary actuator or linear actuator that allows for precise control of angularor linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller (Arduino UNO in our case), often a dedicated module designed specifically for use with servomotors. A Servomotor is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is a signal (either analogue or digital) representing the position commanded for the output shaft. The motor is paired with some typeof position encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signalreduces to zero and the motor stops.

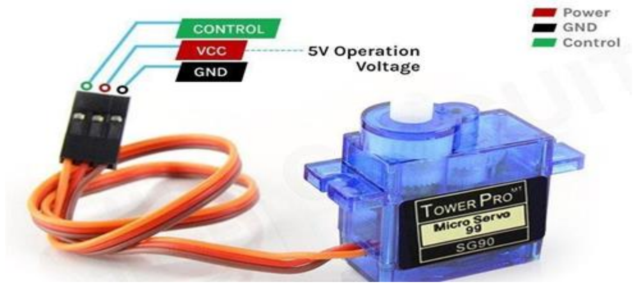


Fig 4.6 Servo Motor

4.5 20x4 I2C LCD

A 20x4 LCD (Liquid Crystal Display) is a display module commonly used in electronics projects and devices. It has 20 characters per row and 4 rows, allowing it to display up to 80 characters of text or custom symbols. The use of a 20x4 LCD (Liquid Crystal Display) with I2C (Inter-Integrated Circuit) communication in a smart car parking system plays a crucial role in enhancing the functionality, user experience, and efficiency of the system. The LCD can display the status of each parking space, indicating whether it's occupied or vacant. This can be achieved by connecting the LCD to a microcontroller that reads data from sensors like infrared sensors installed in each parking space.

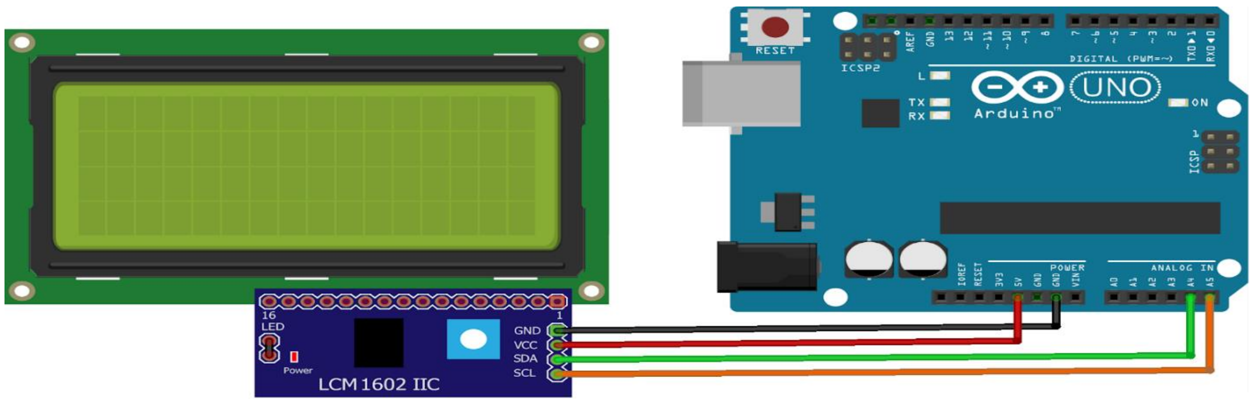


Fig 4.7 20*4 I2C LCD with Arduino Uno

4.6 Software Languages Used

4.6.1 HTML

HTML (Hypertext Mark-up Language) is used to create document on the World Wide Web. It is simply a collection of certain key words called ‘Tags’ that are helpful in writing the document to be displayed using a browser on Internet. It is a platform independent language that can be used on any platform such as Windows, Linux, Macintosh, and so on. To display a document in web it is essential to mark-up the different elements (headings, paragraph, tables, and so on) of the document with the HTML tags. A browser understands and interpret the HTML tags, identifies the structure of the document (which part are which) and makes decision about presentation (how the parts look) of the document.

4.6.2 CSS

- CSS stands for cascading style sheets
- Markup language used in the web document for presentation purpose.
- Various elements like text, font, and color are used in CSS for presentation.
- Can be used to bring styles in the web documents.
- By combining with HTML document, flexibility of contents is achieved**.**
- Cascading Style Sheets **(**CSS**)** is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

4.6.3 Java Script

What is JavaScript?

- JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an **interpreted** programming language with object-oriented capabilities.

- JavaScript is a **single-threaded** programming language that we can use for client-side or server-side development. It is a **dynamically** typed programming language, which means that we don't care about variable data types while writing the JavaScript code. Also, it contains the control statements, operators, and objects like Array, Math, Data, etc.
- JavaScript was first known as **Live Script**, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **Live Script**. The general-purpose core of the language has been embedded in Netscape and other web browsers.

4.7 Proposed System Architecture

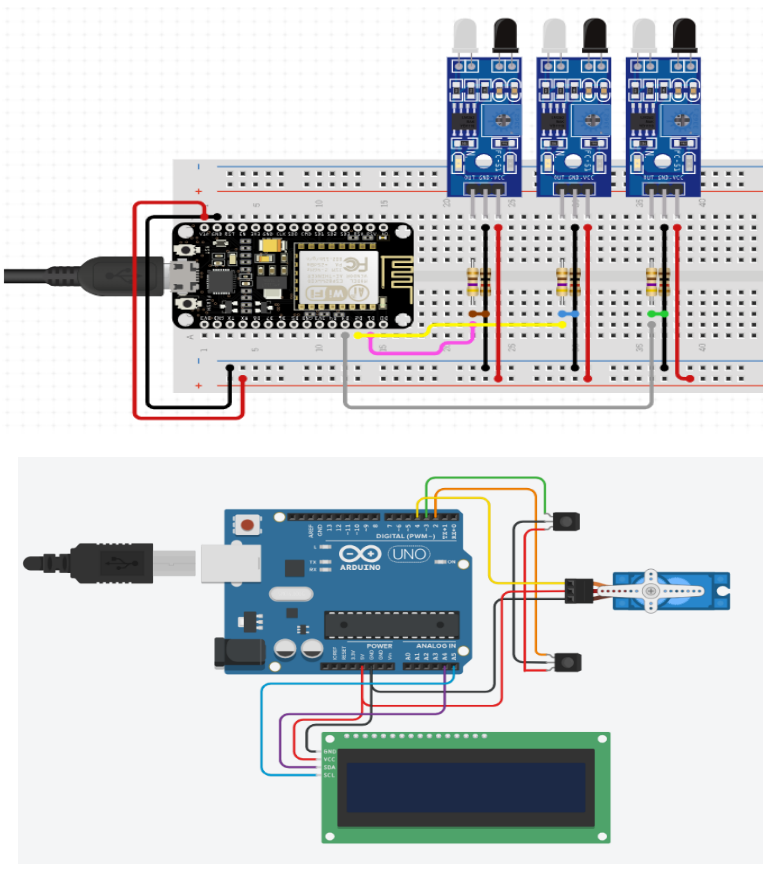


Fig 4.7 System Architecture

CHAPTER 5
IMPLEMENTATION AND CODING

5.1 Implementation

Step 1:- Collect all the hardware components required for this project.

Step 2:- First take Arduino Uno R3which controls the barricade, IR sensor, and I2C LCD display, and connecting wire, soldering iron.

Step 3:- Connection of I2C Liquid Crystal Display (LCD) with Arduino Uno R3

a) Connect SCL pin of I2C LCD to A5 of Arduino Uno
b) Connect SDA pin of I2C LCD to A4 of Arduino Uno
c) Connect VCC pin of I2C LCD to 5V terminal of Arduino Uno
d) Connect GND pin of I2C LCD to GND terminal of Arduino Uno

Step 4:- Connection two Infrared Sensors (IR sensor) to Arduino Uno

a) Connect VCC pin of both the Infrared sensor to 5V terminal of Arduino Uno
b) Connect GND pin of both the Infrared sensor to GND terminal of Arduino Uno
c) Connect OUTPUT pin of First infrared sensor to D2 terminal of Arduino Uno
d) Connect OUTPUT pin of Second Infrared sensor to D3 terminal of Arduino Uno

Step 5:- Connection of Servo Motor (Micro Servo 9g SG90) to Arduino Uno R3

a) Connect the VCC terminal of Servo motor to 5V terminal of Arduino Uno
b) Connect GND terminal of Servo Motor to GND terminal of Arduino Uno
c) Connect CONTROL terminal of Servo Motor to D4 Terminal of Arduino Uno

Step 6:- Now take Node-mcu which will use to show the status of parking spot on local webserver by using IR sensor on parking spot.

Step 7:- Connect six IR sensor to Node-mcu

a) Connect VCC pin of six IR sensor to 3.3V pin of Node-mcu
b) Connect GND pin of six IR sensor to GND pin of Node-mcu
c) Connect OUTPUT pin of six IR sensor to digital output pin of Node-mcu.
i.e. D0, D1, D2, D3, D4, D5.

5.2 Connection Schematic Diagram

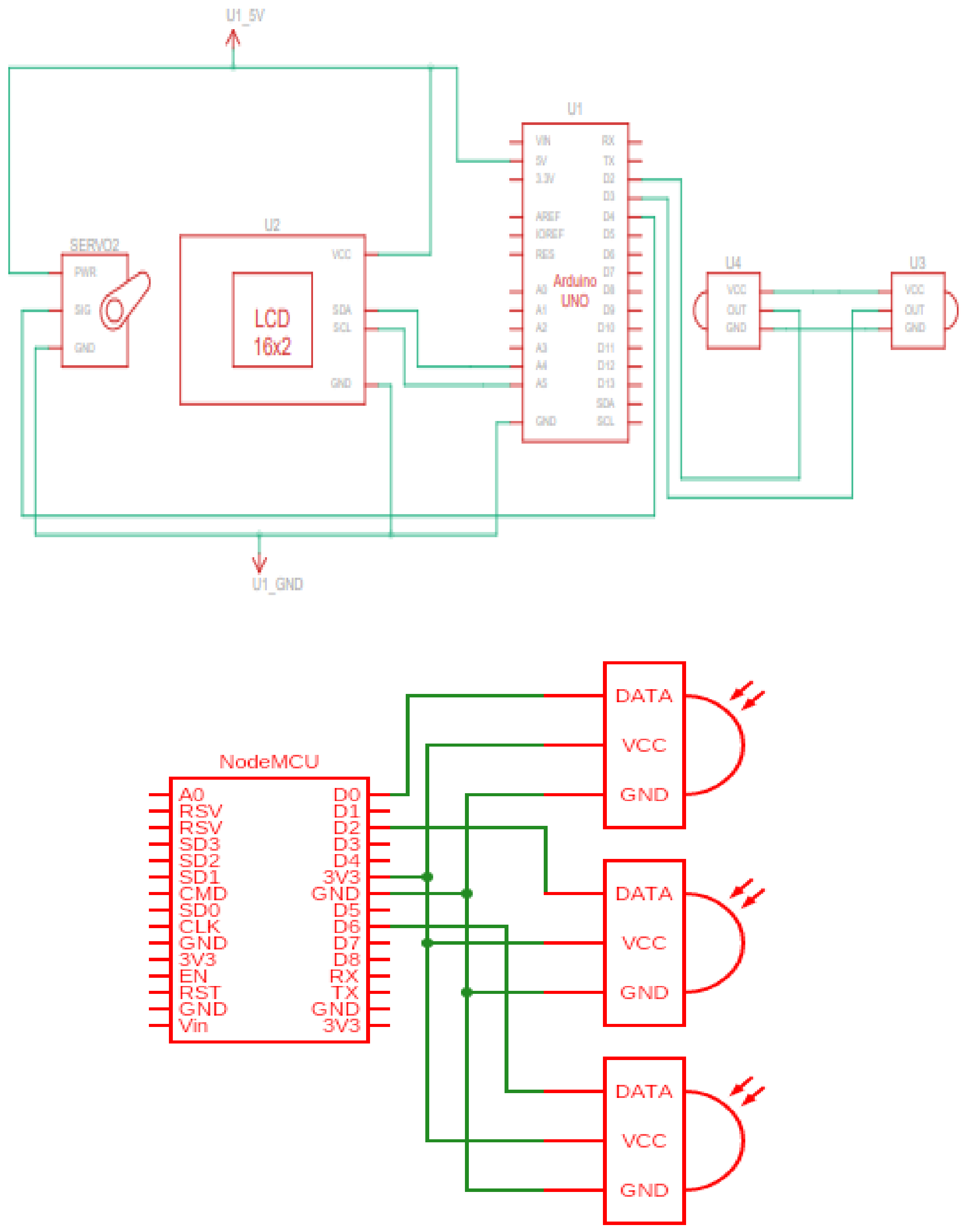


Fig. 5.1. Arduino Uno and Node-MCU System Schematic Diagram

5.3 System Code :-

5.3.1 Arduino Uno Code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);
#include <Servo.h>
Servo myservo;
int IR1 = 2;
int IR2 = 3;
int Slot = 3;          //Total number of parking Slots
int flag1 = 0;
int flag2 = 0;
void setup() {
Serial.begin(9600);
lcd.init(); //initialize the lcd
lcd.backlight(); //open the backlight
pinMode(IR1, INPUT);
pinMode(IR2, INPUT);
myservo.attach(4);
myservo.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){
```

```
myservo.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){
myservo.write(0);
Slot = Slot+1;
}
}
if(flag1==1 && flag2==1){
delay (1000);
myservo.write(100);
flag1=0, flag2=0;
}
lcd.setCursor (0,0);
lcd.print("    WELCOME!    ");
lcd.setCursor (0,1);
lcd.print("Slot Left: ");
lcd.print(Slot);
}
```

5.3.2 Node-MCU ESP8266

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include "webcode.h"
#include <Wire.h>
// #include <LiquidCrystal_I2C.h>
const char *ssid = "Alspr-wifi";
const char *password = "alsprwifi404f";
const int irSensorPin1 = D0; // Replace with your actual pin number
const int irSensorPin2 = D1; // Replace with your actual pin number
const int irSensorPin3 = D2; // Replace with your actual pin number
ESP8266WebServer server(80);
// bool ledStatus = false;
void webpagecode(){
server.send(200, "text/html", webcode);
}
void setup() {
Serial.begin(9600);
pinMode(irSensorPin1, INPUT);
pinMode(irSensorPin2, INPUT);
pinMode(irSensorPin3, INPUT);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(1000);
Serial.println("...");
}
Serial.println("Connected to WiFi");
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
```

```
server.on("/", webpagecode);
// server.on("/", HTTP_GET, handleRoot);
server.on("/status", HTTP_GET, handleStatus);
// server.on("/toggleLED", HTTP_GET, handleToggleLED);
server.begin();
}
void loop() {
server.handleClient();
 // if(isSensorEnabled){
 int sensorValue1 = digitalRead(irSensorPin1);
int sensorValue2 = digitalRead(irSensorPin2);
 int sensorValue3 = digitalRead(irSensorPin3);
updateSensorStatus(sensorValue1, sensorValue2, sensorValue3);
delay(100); // Adjust delay as needed   //
}
}
void handleStatus() {
int sensorValue1 = digitalRead(irSensorPin1);
int sensorValue2 = digitalRead(irSensorPin2);
int sensorValue3 = digitalRead(irSensorPin3);
String status = "Slot1:  " + getStatusText(sensorValue1) + "<br>";
status += "Slot2:  " + getStatusText(sensorValue2) + "<br>";
status += "Slot3:  " + getStatusText(sensorValue3);
server.send(200, "text/plain", status);
}
void updateSensorStatus(int sensorValue1, int sensorValue2, int sensorValue3 ) {
String status = "Slot1:  " + getStatusText(sensorValue1) + "<br>";
status += "Slot2:  " + getStatusText(sensorValue2) + "<br>";
status += "Slot3:  " + getStatusText(sensorValue3);
server.send(200, "text/plain", status);
}
String getStatusText(int sensorValue) {
```

```
// Use logical AND to determine if the parking space is occupied
if (sensorValue == LOW) {
return "Occupied";
} else {
return "Empty";
}
}
```

5.3.3 WebpageCode Programme :-

```
const char webcode[] =
R"=====(
<!DOCTYPE html>
<head>
<title>SMP</title>
<style>
.reserved-button {
padding: 10px 20px;
background-color: rgb(blue);
color: white;
border: none;
cursor: pointer;
transition: background-color 0.3s ease;
}
.reserved-button:hover {
background-color:rgb(blue);
}
#reservation-status {
display: inline-block;
margin-left: 10px;
font-weight: bold;
}
.body{
background-color: aliceblue;
font-family:Georgia, 'Times New Roman', Times, serif;
}
.container{
margin: 172px 492px 91px 471px;
border: 3px solid black;
border-radius: 10px;
background-color:yellow ;
}
.container .container-1{
background-color: aqua;
border: 0px solid black;
border-bottom: 3px solid black;
border-top-left-radius: 8px;
border-top-right-radius: 8px;
border-top: 0.5px solid black;
border-left: 0.4px solid black;
border-right: 0.4px solid black
}
.container .container-1 .header{
padding-left: 94px;
}
.container-2{
display: flex;
}
.showstatus{
margin: 11px 89px 1px 90px
}
.container-3{
border: 0px solid black;
border-left: 1px solid black;
padding: 17px 1px 1px 67px;
}
```

```
@media  (min-width: 400px){
.body{
background-color: blue;
}
.container{
margin: 172px 37px 91px 52px; ;
border: 3px solid black;
border-radius: 10px;
background-color:yellow ;
}
.container .container-1{
background-color: rgb(17, 222, 222);
border: 0px solid black;
border-bottom: 3px solid black;
border-top-left-radius: 8px;
border-top-right-radius: 8px;
border-top: 0.5px solid black;
border-left: 0.4px solid black
border-right: 0.4px solid black
}
.container .container-1 .header{
padding-left: 95px;
font-size: 5em;
}
.container-2{
display: flex;
}
.showstatus{
margin: 21px 107px 1px 108px;
font-size: 3em;
}
```

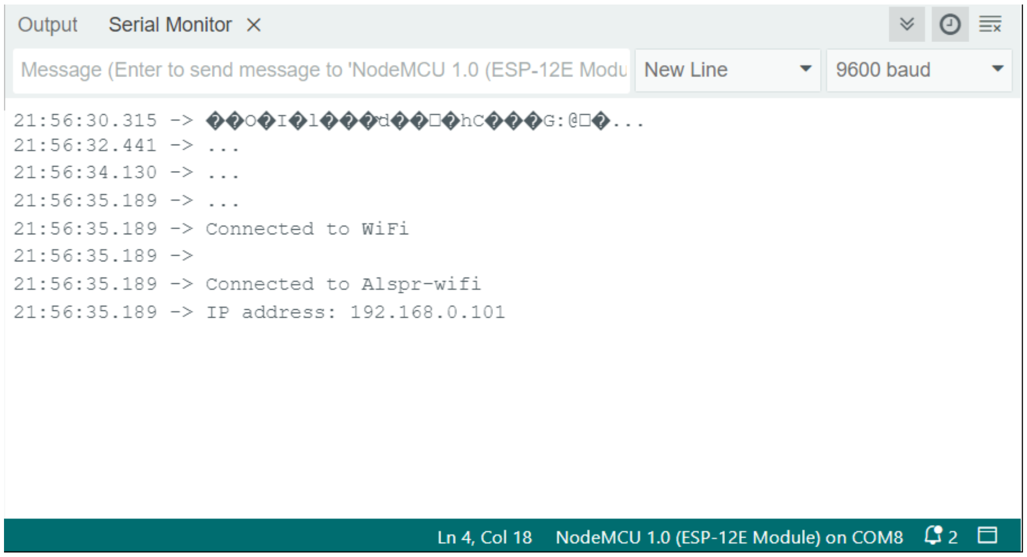
```
.container-3{
 border: 0px solid black;
 border-left: 3px solid black;
 padding: 17px 52px 1px 67px;
}
.reserved-button1 {
font-size: 2em;
}
#reserveButton1{
font-size: 2em;
}
.reserved-button2{
font-size: 2em;
}
#reserveButton2{
font-size: 2em;
.reserved-button3{
font-size: 2em;
}
#reserveButton3{
font-size: 2em;
}
}
</style>
</head>
<body>
<div class="container">
<div class="container-1">
<h1 class="header">Smart Parking Space</h1>
</div>
<div class="container-2">
<p class="showstatus" id='sensorStatus'>Loading...</p>
```

```
<div class="container-3">
<button id="reserveButton1" class="reserved-button1"
onclick="changeColorAndText1()">Reserve</button>
<! -- <span id="reservation-status">Not Reserved</span> -->
<button id='reserveButton1' onclick='reserveSpot1()'>Reserve</button>
 <p id='timer1'></p>
<button id="reserveButton2" class="reserved-button2"
onclick="changeColorAndText2()">Reserve</button>
 <button id='reserveButton2' onclick='reserveSpot2()'>Reserve</button>
<p id='timer2'></p>
<button id="reserveButton3" class="reserved-button3"
onclick="changeColorAndText3()">Reserve</button>
<button id='reserveButton3' onclick='reserveSpot3()'>Reserve</button>
<p id='timer3'></p>
</div>
</div>
</div>
<script>
function updateSensorStatus() {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById('sensorStatus').innerHTML = this.responseText;
}
};
xhttp.open('GET', '/status', true);
xhttp.send();
}
function reserveSpot1() {
var button1 = document.getElementById('reserveButton1');
button1.style.backgroundColor = 'red';
button1.innerHTML = 'Reserved';
```

```
var count1 = 20;
var timer1 = setInterval(function() {
var timerDisplay1= document.getElementById('timer1');
timerDisplay1.innerHTML = 'Timer: ' + count1 + 's';
count1--;
    if (count1 < 0) {
    clearInterval(timer1);
    button1.style.backgroundColor = '';
    button1.innerHTML = 'Reserve';
    timerDisplay1.innerHTML = '';
}
}, 1000);
}
function reserveSpot2() {
var button2 = document.getElementById('reserveButton2');
button2.style.backgroundColor = 'red';
button2.innerHTML = 'Reserved';
var count2 = 20;
var timer2 = setInterval(function() {
var timerDisplay2 = document.getElementById('timer2');
timerDisplay2.innerHTML = 'Timer: ' + count2 + 's';
count2--;
if (count2 < 0) {
clearInterval(timer2);
button2.style.backgroundColor = '';
button2.innerHTML = 'Reserve';
timerDisplay2.innerHTML = '';
}
}, 1000);
}
function reserveSpot3() {
var button3 = document.getElementById('reserveButton3');
```

```
button3.style.backgroundColor = 'red';
button3.innerHTML = 'Reserved';
var count3 = 20;
var timer3 = setInterval(function() {
var timerDisplay3 = document.getElementById('timer3');
timerDisplay3.innerHTML = 'Timer: ' + count3 + 's';
count3--;
if (count3 < 0) {
clearInterval(timer3);
button3.style.backgroundColor = '';
button3.innerHTML = 'Reserve';
timerDisplay3.innerHTML = '';
}
}, 1000);
}
setInterval(updateSensorStatus, 1000); // Update every 1 second
function changeColorAndText1() {
var button = document.getElementById("reserveButton1");
var status = document.getElementById("reservation-status1");
if (button.style.backgroundColor === "rgb(0, 123, 255)") {
button.style.backgroundColor = "#dc3545"; // Change color to red
status.textContent = "Reserved"; // Change text
} else {
button.style.backgroundColor = "#007bff"; // Change color back to original blue
status.textContent = "Not Reserved"; // Change text
}
}
function changeColorAndText2() {
var button = document.getElementById("reserveButton2");
var status = document.getElementById("reservation-status2");
if (button.style.backgroundColor === "rgb(0, 123, 255)") {
button.style.backgroundColor = "#dc3545"; // Change color to red
status.textContent = "Reserved"; // Change text
} else {
button.style.backgroundColor = "#007bff"; // Change color back to original blue
status.textContent = "Not Reserved"; // Change text
}
}
function changeColorAndText3() {
var button = document.getElementById("reserveButton3");
var status = document.getElementById("reservation-status3");
if (button.style.backgroundColor === "rgb(0, 123, 255)") {
button.style.backgroundColor = "#dc3545"; // Change color to red
status.textContent = "Reserved"; // Change text
} else {
button.style.backgroundColor = "#007bff"; // Change color back to original blue
status.textContent = "Not Reserved"; // Change text
}
}
</script>
</body>
</html>
)=====";
```

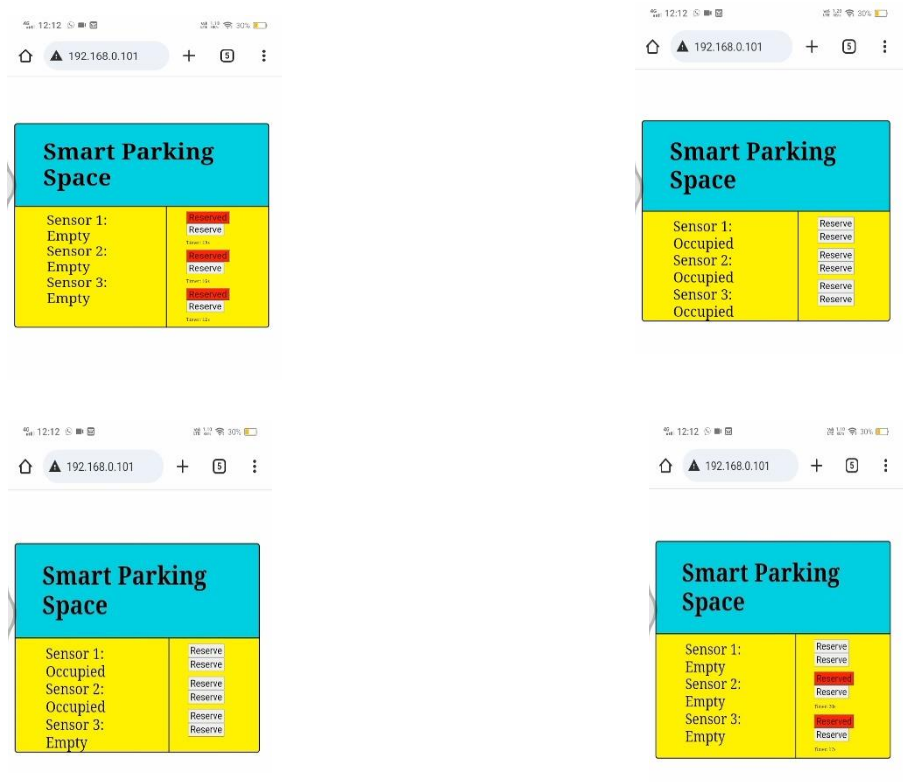
5.4 Output


Output
Serial Monitor
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Modu
New Line
9600 baud

```
21:56:30.315 -> ��o�I�l���d��□�hC���G:@□�...
21:56:32.441 -> ...
21:56:34.130 -> ...
21:56:35.189 -> ...
21:56:35.189 -> Connected to WiFi
21:56:35.189 ->
21:56:35.189 -> Connected to Alspr-wifi
21:56:35.189 -> IP address: 192.168.0.101
```

Ln 4, Col 18   NodeMCU 1.0 (ESP-12E Module) on COM8

Fig.5.2. Local Testing IP Address



Smart Parking Space

Sensor 1: Empty
Sensor 2: Empty
Sensor 3: Empty

Reserved Reserve
Reserved Reserve
Reserved Reserve

Smart Parking Space

Sensor 1: Occupied
Sensor 2: Occupied
Sensor 3: Occupied

Reserve Reserve
Reserve Reserve
Reserve Reserve

Smart Parking Space

Sensor 1: Occupied
Sensor 2: Occupied
Sensor 3: Empty

Reserve Reserve
Reserve Reserve
Reserve Reserve

Smart Parking Space

Sensor 1: Empty
Sensor 2: Empty
Sensor 3: Empty

Reserve Reserve
Reserved Reserve
Reserved Reserve

Fig.5.3. Outputs of Node-MCU local webserver

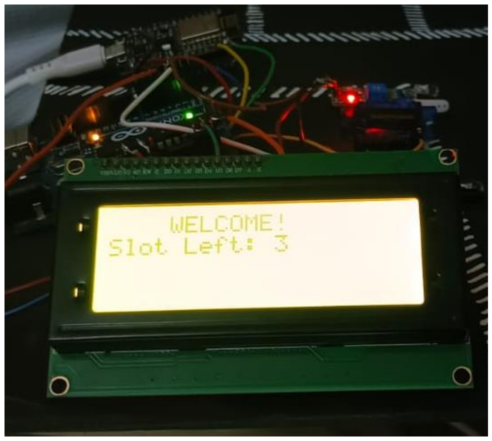


Fig.5.4. When all spots are Empty

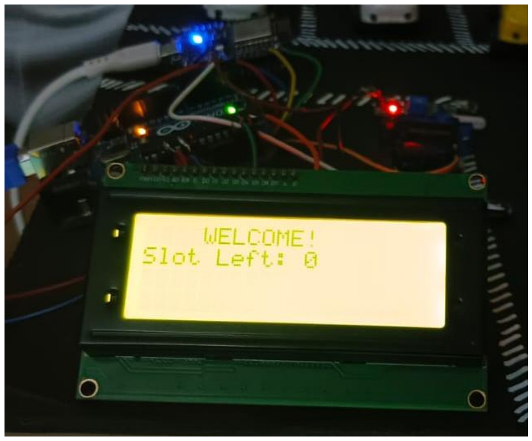


Fig.5.5. When all spots are Filled

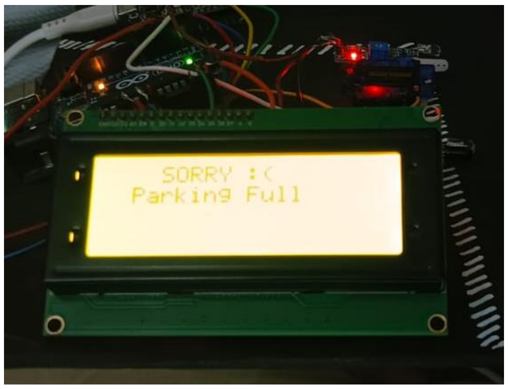


Fig.5.6. When all spots are Filled
and No Space For Another Car

CHAPTER 6

SOFTWARE TESTING

6.1Hardware Assembly:

- Connect the IR sensors to Arduino Uno, ensuring proper alignment and placement in the parking spaces.
- Interface the I2C LCD with Arduino Uno, adjusting settings for display clarity.
- Connect and calibrate the servo motor to the gate mechanism, allowing forprecise control.

6.2 Testing and Calibration:

- Conduct initial tests to ensure individual components function correctly.
- Calibrate IR sensors to accurately detect the presence or absence of vehicles.
- Verify the servo motor's responsiveness and accuracy in controlling the gate.
- Test the integration of all components to ensure seamless communication and functionality.

6.3 User Interface Integration:

- Implement user-friendly messages and indicators on the I2C LCD to conveyparking space status.
- Design an intuitive interface that provides clear information to users regarding available parking spaces.

6.4 System Integration:

- Integrate all components into a cohesive system, ensuring proper connections and communication.
- Address any compatibility issues or conflicts that may arise during the integration process.

6.5 Deployment and Field Testing:

- Install the Smart Car Parking System in a real-world setting, considering environmental factors and user accessibility.
- Conduct field tests to validate the system's performance in detecting parking space occupancy and controlling the gate.

6.6 Optimization and Refinement:

- Gather feedback from initial deployments and identify areas for improvement.
- Refine the Arduino code and system settings to optimize performance and address any observed issues.

CHAPTER 7

CONCLUSION

This study has proposed a parking system that improves performance by reducing the number ofusers that fail to find a parking space and minimizes the costs of moving to the parking space. Our proposed architecture and system has been successfully simulated and implemented in a realsituation. The results show that our algorithm significantly reduces the average waiting timeof users for parking. Our results closely agree with those of our proposed mathematical models. The simulation of our system achieved the optimal solution when most of the vehicles successfully found a free parking space. The average waiting time of each car parkfor service becomes minimal, and the total time of each vehicle in each car park is reduced.In our future study, we will consider the security aspects of our system as well as implementour proposed system in large scales in the real world. If this smart parking system with all the mentioned modifications is implemented, then surely this would be a revolution in the field of parking. This system would save time, fuel, labor and thus would decrease the cost.All we need is just a single time investment. We would be able to check the availability of parking space a well as book a slot just by a single click. Since the details of the vehicle as well as the driver are being collected, so they can be referred for any discrepancy in the parking slots, thus improving the security of our vehicles too.

FUTURE WORK

- Multi-factor authentication combine RFID tags with additional security measures like facial recognition or PIN codes for more robust access control, preventing unauthorized access even if credentials are compromised. Data encryption and anonymization Implement strong data encryption protocols to protect sensitive vehicle and user information collected through RFID tags. Anonymization techniques can further enhanceprivacy by decoupling individual vehicle data from user identities.
- We would make the mobile app or WebApp to display the real time status of parking lots inthe parking area in nearby locations. The main components of the Mobile app can be List ofParking Areas in the location Users can see the list of all the parking areas available in the nearby locations. Location can be automatically detected by the app (depends on the user).
- An automatic ticket generating system can be added to this project. The parts of to be added forimplementing this extension are as follows:
  - Object detection: The object detection model will analyze the size of the vehicle and detect the type of it. And accordingly it will provide a suitable parking slot for your vehicle.
  - Optical Character Recognition: For an automatic ticketing system, it needs to have a registration number and brand of the vehicle on the parking ticket. So for extracting the registration number and the brand, we will use the OCR model to detect the number plate and the text on it. To generate tickets.
- We will make the Parking space occupancy detection detector is Ultrasonic radiators emit high- frequency sound waves that bounce off parked vehicles and return to the sensor, enablingthe system to determine if a space is occupied. This data is then used to guide drivers to availablespots and optimize parking space utilization. By analyzing the reflected sound waves, ultrasonicsensors can differentiate between cars, motorcycles, bicycles, and even pedestrians occupying parking spaces. This information can be used for targeted guidance, fee calculation, and security purposes.

REFERENCES

[1] N. Farooqi, L. Najmi, S. Alshehri, Ghaidaa, Sahar, A. Alrashedi, "UParking: Developing a Smart Parking Management System Using the Internet of Things", *2019 Sixth HCT Information Technology Trends (ITT),* United Arab Emirates, November 2019.

[2] P. Sadhukhan, "An IoT-based E-Parking System for Smart Cities" *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, India, September 2017.

[3] S. Kazi, S. Nuzhat, A. Nashrah, Q. Rameeza, "Smart Parking System to Reduce Traffic Congestion" *2018 International Conference on Smart City and Emerging Technology (ICSCET),* IEEE, India, January 2018.

[4] R. Kanan, H. Arbess, "An IoT-Based Intelligent System for Real-Time Parking Monitoring and Automatic Billing" *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, IEEE, Qatar, Februray 2020.

[5] L. F. H. Quintero, J. V. Alfonso, D. Bermúdez, L. A. Marentes, K. Banse, "ITS for Smart Parking Systems, towards the creation of smart city services using IoT and cloud approaches", *2019 Smart City Symposium Prague (SCSP)*, IEEE, Czech Republic, May 2019.

[6] D. H. Rangrej, S. Patel, P. Kasodariya, H. Tanti, "Smart Parking System based on IOT", *International Journal of Engineering Research and V9(05),* Vol. 9, Issue 5, May 2020, India.

[7]. Anusha, Arshitha M S, Anushri, G. Bishtannavar, "Review Paper on Smart Parking System", *International Journal of Engineering Research & Technology (IJERT)*, Vol. 7, Issue 8, June 2019, India

[8] D. Balmiki, M. Singhal , A. Singh , D. Tyagi, "A Research on Smart Vehicle Parking System", *International Journal of Scientific Research and Management Studies (IJSRMS),* Vol. 4, Issue 7, pg: 124-127.

[9] N. M. F. A. B. Azmi1, M. B. Ismail," Smart Parking System Using IoT with Ultrasonic Sensor", *Journal of Engineering Technology*, Vol. 10(1): 93-97, 2022.

[10] D. Ashok, A. Tiwari, V. Jirge, "Smart Parking System using IoT Technology*", 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE),* IEEE, Vellore, India, February 2020.