SMART PARKING SYSTEM

A PROJECT REPORT

submitted by

ARUN CYRIAC(MAC20EC033)

JACOB BABY (LMAC20EC120)

NEIL KOSHY GEORGE (MAC20EC085)

PAUL CHERIYAN(LMAC20EC123)

to

the APJ Abdul Kalam Technological University

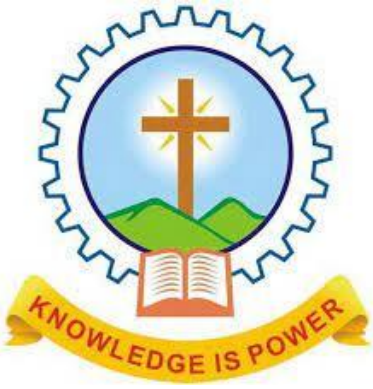
in partial fulfillment of requirements for the award of the Degree

of

Bachelor of Technology

in

Electronics and Communication Engineering



Department of Electronics and Communication Engineering
Mar Athanasius College of Engineering
Kothamangalam, Kerala, India 686666
MAY 2024

DECLARATION

We **ARUN CYRIAC**,**JACOB BABY**,**NEIL KOSHY GEORGE**,**PAUL CHERIYAN** hereby declare that the project report **SMART PARKING SYSTEM**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done under supervision of Dr. VinodKumar Jacob .

This submission represents our ideas in our own words and where ideas or words of others have been included, We have adequately and accurately cited and referenced the original sources.

We also declare that We have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kothamangalam
09-05-2024

ARUN CYRIAC
JACOB BABY
NEIL KOSHY GEORGE
PAUL CHERIYAN

DEPT. OF ELECTRONICS & COMMUNICATION ENGINEERING

MAR ATHANASIUS COLLEGE OF ENGINEERING, KOTHAMANGALAM

CERTIFICATE

This is to certify that the report entitled **SMART PARKING SYSTEM** submitted by **ARUN CYRIAC**(MAC20EC033) **JACOB BABY**(LMAC20EC120) **NEIL KOSHY GEORGE**(MAC20EC085) **PAUL CHERIYAN**(LMAC20EC123)to the APJ Abdul Kalam Technological University in partial fulfillment of the Bachelor of Technology in Electronics and Communication Engineering is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Dr. VinodKumar Jacob
(Project Guide)

Dr. Jiss Paul
(Project Coordinator)

Dr.Aji Joy
(Head of Department)

MAR ATHANASIUS COLLEGE OF ENGINEERING * KOTHAMANGALAM, KERALA * DEPT. OF ELECTRONICS & COMMUNICATION ENGINEERING

ACKNOWLEDGEMENT

We would like to express my profound gratitude to God, whose unwavering guidance and blessings have illuminated my path throughout this project work.We would like to place on record our sincere gratitude to our project guide **Dr. VinodKumar Jacob**, Electronics and Communication Engineering, Mar Athanasius College of Engineering for the guidance and mentorship throughout the course.

We would like to express sincere gratitude to **Dr.Aji Joy** , Head of Department, Electronics and Communication Engineering, Mar Athanasius College of Engineering Kothamangalam for providing us with all the necessary facilities and support.

We express our sincere thanks to **Dr. Jiss Paul** , department of Electronics and Communication Engineering, Mar Athanasius College of Engineering Kothamangalam for their support and co-operation.

We express our sincere thanks to all technical staffs for providing us with all support and necessary facilities.We express our sincere thank to all our friend and family for providing us with all support and cooperation

ARUN CYRIAC
JACOB BABY
NEIL KOSHY GEORGE
PAUL CHERIYAN

ABSTRACT

In this project, Smart Parking System created especially for retail centres is presented. The system uses data analytic and state-of-the-art sensor technology to deliver real-time parking availability information via digital displays or mobile apps. Predictive algorithms improve traffic flow, easing congestion and raising productivity levels.The parking process is further streamlined with automated payment alternatives and an intuitive interface that directs visitors to available spaces.In crowded retail settings, this intelligent parking solution seeks to increase operational effectiveness.

In this phase we created hardware prototype of a parking area.We use IR sensors to monitor the presence of the vehicle in each parking slot the parking area,whether the vehicle is present in that particular slot or not.Then we use the IR sensor to send this data to cloud via WiFi module.The user can access this information in real time through mobile app.Also through this application user can book the parking slot.This information is passed through the cloud. Then through the data stream between hardware and cloud the booking information is passed to microcontroller.Microcontroller is connected with booking indication LED.This informs other user that the slot is booked. By this I implemented realtime availability of parking slots and its booking

Contents

List of Figures

ABBREVIATIONS

PWM	Pluse Width Modulation
RST	Reset Pin
API	Application Interface
EDA	Electronics Design Automation
GND	Ground
LED	Light Emitting Diode
ML	Machine Learning
RX	Receiver Pin
PCB	Printed Circuit Board
TX	Transmitter Pin

CHAPTER 1

INTRODUCTION

This project introduces an advanced Smart Parking System designed specifically for larger retail centers. It utilizes cutting-edge sensor technology and data analytics to provide real-time updates on parking availability through digital displays or mobile apps. By utilizing predictive algorithms, the system aims to improve traffic flow, reduce congestion, and enhance operational efficiency in busy retail environments. Key components include IR sensors that monitor vehicle occupancy in parking slots, with data sent to the cloud via WiFi for instant access through mobile applications. Users can conveniently check parking availability, reserve spots, and make payments through a user-friendly mobile interface. The system also includes automated payment options and a user-friendly interface to guide visitors to open parking spaces, enhancing the overall user experience. This project demonstrates the potential of IoT technologies in transforming parking management strategies for larger retail settings, offering scalable solutions to optimize resource allocation and streamline parking operations. Future developments may focus on scalability, integration with navigation systems, and further enhancements to improve user interactions and system performance.

1.1 GENERAL BACKGROUND

This project introduces an innovative Smart Parking System designed specifically for retail centers, with a primary goal of optimizing parking management efficiency by leveraging data analytics and advanced sensor technology. The system provides real-time updates on parking availability through digital displays or mobile apps, utilizing sophisticated predictive algorithms to improve traffic flow, reduce congestion, and enhance productivity levels within parking facilities. By integrating automated payment options and an intuitive interface that directs visitors to open parking spaces, this intelligent solution aims to significantly enhance operational effectiveness in busy retail settings.

During the initial phase of development, a comprehensive hardware prototype of the parking area was created. This prototype incorporates advanced IR sensors strategically positioned to monitor vehicle presence in each parking slot. The data collected by these sensors is transmitted to the cloud infrastructure via a WiFi module, enabling users to access up-to-date parking information directly from their mobile devices through a dedicated app interface. Moreover, the mobile app enables users to conveniently reserve parking slots, with the booking information efficiently relayed to a microcontroller via the cloud platform. The microcontroller then activates intuitive LED indicators to notify other users when specific parking slots have been reserved.

In summary, this project demonstrates a transformative approach to implementing real-time parking availability updates and streamlined booking functionalities within retail environments. The integration of advanced sensor technologies, cloud connectivity, and user-centric mobile applications represents a significant advancement in smart parking systems, offering tangible benefits such as improved efficiency, reduced congestion, and enhanced user experiences. Moving forward, future developments may focus on scalability, seamless integration with navigation systems, and continuous refinements to optimize system performance and accessibility in retail settings.

1.2 OBJECTIVES

The objective of this project is to integrate multiple technologies to optimise parking management. Infrared (IR) sensors are used in each parking slot to track and monitor the overall number of parked cars as well as the number of open spots. The ESP8266 module is then used to send the acquired data to the cloud, guaranteeing effective parking resource use and real-time monitoring.

The objective are:

1. To use the IR sensor in each parking slot the maintain the data of the total number of vehicles parked and the free parking space.

2. To send the above data to cloud using the ESP32.

3.To maker the realtime data of parking slot available to the user through moblie app.

4.To make an interface between the user and web server which allows the user to book a parking slot according to the availability.

5. To verify the user at the entrance of the parking area using RFID card provided for each user.

6. To collect the parking fee using RFID card when the vehicle leaves the parking area

1.3 SCOPE

The creation of a smart parking system designed for larger retail centers represents a significant advancement in parking management technology. This innovative system utilizes cutting-edge sensor technology and data analytics to provide real-time parking availability updates to users through digital displays or mobile apps. By employing predictive algorithms, the system not only enhances traffic flow but also reduces congestion and boosts productivity within the parking infrastructure.

A key feature of this system is its ability to simplify the parking process with automated payment options and a user-friendly interface that guides visitors to open parking spaces. In busy retail environments, this intelligent parking solution aims to improve operational efficiency by offering precise and current parking information, ultimately minimizing search time and enhancing the overall visitor experience.

The incorporation of cloud connectivity facilitates efficient data exchange between the parking system and users' mobile devices, enabling remote access to parking availability details and the capability to reserve parking spaces instantly. Furthermore, the system is scalable to accommodate larger parking areas within retail centers, making it adaptable to diverse settings and parking requirements. Overall, the implementation of this smart parking system presents an innovative solution to optimize parking management and enhance convenience in larger retail environments.

1.4 SCHEME OF THE PROJECT WORK

This project introduces a specialized Smart Parking System for retail centers, utilizing advanced sensor technology and data analytics to provide real-time parking availability updates via digital displays or mobile apps. Predictive algorithms are employed to optimize traffic flow, reducing congestion and improving productivity in parking areas.

The parking process is enhanced with automated payment options and an intuitive interface that directs visitors to open parking spaces, aiming to increase operational efficiency in busy retail settings.

During this phase of development, a hardware prototype was constructed using IR sensors to monitor vehicle presence in parking slots. Occupancy data is transmitted to the cloud through a WiFi module, allowing users to access real-time information via a mobile app. Users can also book parking slots through the app, with booking details sent to a microcontroller via the cloud. The microcontroller controls an LED indicator to notify others when a slot is booked.

This implementation successfully demonstrates real-time parking availability and booking features within the project's scope, improving parking management efficiency and convenience for retail center visitors.

CHAPTER 2

LITERATURE SURVEY

To understand and learn about the background and evolution of the system,we referred various journals,conference paper and articles which are ststed below

[1]M. P. Thakre1 et. al,introduced parking system as important issue that has arisen in metropolitan areas as a result of the rise in automobile traffic is the lack of sufficient parking spots. This problem is especially apparent in places with high population density, such as retail centres, medical facilities, and educational institutions. In order to alleviate the parking issues, this project aims to implement an automated real-time vehicle parking system. The suggested solution uses the ESP8266 Node MCU to create a connection between the parking lot and the internet by utilising the Internet of Things (IoT), which enables communication between physical items.Each parking space has an infrared sensor installed to determine the slots' availability in order to deploy this system. The length of time the car is used is used by the system to determine how much parking will cost. An RFID tag that has been registered is given to each user, which helps with efficient parking space management. Users may also find available parking spots fast with the help of a smart car parking app. This all-inclusive system uses RFID tags and Internet of Things technology to automate the parking process.

[2]L. Kumar et. al,introduced a parking system for safe parking spaces as a result of growing urbanisation and an increase in the quantity of urban vehicles. This article suggests integrating RFID and GSM technologies as a means of addressing the problem of vehicle theft from parking lots. Automation that uses radio frequency

identification technology lowers transaction costs dramatically. A GSM kit, RFID readers, RFID tags, barrier gates, PCs, software, and LED lights are all necessary parts of this solution. The system is designed to work independently and adapt to different organisational contexts. Its main purpose is to stop car theft within an organisation, which improves security for its members.The system comprises the GSM kit operating in unison, obstacles operating, and LEDs activating under various scenarios. Because of its flexibility, it may function well in a variety of organisational settings. Smart cards and RFID vehicle tags are used to assure vehicle security, and specialised software is used to manage check-ins and check-outs. The technology immediately sends out an SMS message to authorised members in the event of any theft or unauthorised action, and it also sets off an alarm for heightened security. In addition to providing a strong mechanism to prevent vehicle theft in organisational settings, this all-inclusive solution not only addresses the growing need for secure parking but also improves general safety and peace of mind for its customers.

[3]J D Bachhav et. al, introduced a comprehensive and successful method for automating the management of parking systems, making excellent use of Internet of Things (IoT) technology to distribute parking spaces. The Internet of Things integration makes the system wirelessly accessible, allowing customers to easily keep an eye on the availability of parking spaces. Growing traffic in urban areas has resulted in heavy traffic congestion, which presents a major obstacle. This paper's main objective is to address and overcome this problem.Usually consumers have had difficulty finding parking spots in authorised zones, which causes traffic jams and inefficiency and requires significant time and effort on their part. By providing users with notifications that provide real-time parking information, the suggested solution aims to reduce this issue. The goal of this function is to shorten the wait times for users who are actively looking for parking spots.RFID technology is incorporated into the system as a security feature to discourage vehicle theft. This research offers a comprehensive solution to reduce urban traffic congestion and improve the general user experience in parking facilities by combining effective space allocation through IoT with increased security measures with RFID technology.

[4]A. O. Kotb et. al, introduced a model to solveparking problems, especially in those

with high traffic volumes, have a noticeable and direct effect on people's everyday life as well as traffic flow. In this study, a novel smart parking system based on dynamic pricing, intelligent resource allocation, and reservation methods is presented. The solution that is being suggested aims to tackle the present parking issues by guaranteeing parking reservations at the most affordable prices with the least amount of time that drivers have to hunt, while also optimising income and parking managers' resource utilisation. The research also proposes novel equitable pricing schemes that are feasible to put into practise in addition to being theoretical.This creative method uses mixed-integer linear programming (MILP), a type of mathematical modelling. The principal aim is to reduce the overall financial expenses incurred by drivers while concurrently maximising the efficiency of parking resources. By means of guaranteed reservations and equitable pricing strategies, the system aims to improve motorist satisfaction with parking while simultaneously increasing income production and resource efficiency for parking facility administrators.

[5]J.F. Zafar et. al,introduces human users gadgets' localization and location. It provides a thorough analysis of the benefits of current systems that have been suggested in the literature. This research, in contrast to previous surveys, assesses various systems based on their energy efficiency, availability, cost, reception range, latency, scalability, and tracking accuracy. The focus is on comparing the localization systems and summarising their underlying ideas, as opposed to comparing technology or methodologies. The study provides a thorough summary of the current status of the subject and also explores the remaining obstacles to precise indoor localization.Recently, there has been a greater focus on the topic of indoor localization due to the potential for a wide range of applications made possible by ubiquitous connection and the Internet of Things (IoT). In an effort to improve user experiences, a range of methods, wireless technologies, and mechanisms have been put forth in the literature to provide indoor localization services. In spite of these efforts, there is a lack of information in the literature regarding a current survey that includes the most recent reliable and accurate indoor localization systems. By offering a thorough analysis of various indoor localization methods, such as angle of arrival (AoA), time of flight (ToF), return time of flight (RTOF), and received signal strength (RSS), this research aims to close this gap. This paper intends to close the current knowledge

and understanding gap in this sector by performing a thorough examination of several indoor localization approaches, such as angle of arrival (AoA), time of flight (ToF), return time of flight (RTOF), and received signal strength (RSS). In order to contribute to a more thorough understanding of indoor localization technologies, the aim is to give a full evaluation and comparison of these diverse techniques. The goal of this analysis is to provide light on the advantages, disadvantages, and possible uses of each technique in order to inform and direct future advancements in indoor localization systems.

[6]X. Wang, L. Gao et. al,indoor localization on commodity 5-GHz WiFi networks using fingerprinting. Three hypotheses concerning the channel state information (CSI) data of 5-GHz orthogonal frequency division multiplexing (OFDM) channels are first validated theoretically and experimentally. The authors then present BiLoc, a system that uses widely accessible WiFi devices to localise users in indoor surroundings through the use of bi-modality deep learning. A deep learning-based system is created to take advantage of bi-modal data, namely average amplitudes and estimated angles of arrival. These amplitudes are acquired by calibrating CSI data using a number of suggested methods. The BiLoc system is implemented with off-the-shelf WiFi equipment and is intended to perform well in both the offline and online phases of indoor fingerprinting.

[7]P. P. Gaikwad et. al,introduced the problem faced by expantion ofthe Internet of Things (IoT). Particularly noticeable is this increase in areas like smart homes and industrial wireless sensor networks (WSN). A subset of Internet of Things applications known as ”smart homes” include appliances and household gadgets that may be remotely monitored and operated. This paper explores the architectural features of the Internet of Things, with a particular emphasis on how the technology is being used in smart homes.New obstacles arise when technology develops and architectural advancements take hold, such as server security, security issues in smart homes, and efficient management and control of the overall system. The article describes the Internet of Things' architecture and highlights how smart homes fit within it. IoT-based Smart Homes are systems that are created when standard protocols and suitable network architecture are used to connect household items in smart homes to the

internet.By enabling remote monitoring and control, smart homes help to simplify home automation activities. In addition to highlighting the difficulties IoT and smart home systems confront, the study offers possible solutions to these problems. The study intends to contribute to a thorough understanding of the complexities related to IoT-based smart homes by addressing both issues and solutions.

[8]S. Lee et. al,introduced a unique energy-conscious access point (eAP) system that is intended to improve the energy efficiency of IoT devices that are connected to IEEE 802.11 Wi-Fi networks. A cross-layer design strategy is used by the proposed eAP system to manage IoT devices' energy resources and increase their operational lifetime. Three crucial elements of the system contribute to increased energy efficiency: a numerous IoT data aggregate function, a caching-and-retransmission IoT data function, and a rapid TCP ACK transmit function.The eAP system has a device energy management module that accurately regulates a number of operational parameters, including the length of time that IoT packets transmit, the value that IoT devices assign to their delivery of traffic indication messages (DTIMs), and the power at which IoT devices transmit. These controls are essential for both achieving the service requirements for dependable IoT services and prolonging the battery life of IoT devices.

[9]A. D. Wankhade et. al,introduced a framework for cloud-based Internet of Things networks that is secure and makes use of lightweight cryptography and machine learning. The aim of this study is to investigate attack detection, attack mitigation, and lightweight cryptography algorithm implementation for secure end-to-end communication in Internet of Things (IoT) networks supported by clouds. The framework's goal is to offer a reliable and flexible security solution that is suited to the unique difficulties presented by IoT contexts.The suggested framework aims to recognise and react to new threats instantly by incorporating machine learning techniques, increasing its adaptability to changing assault situations. To make sure that security measures do not place an excessive computational load on IoT devices with limited resources, lightweight cryptography has been developed. As a result, the framework helps to build a more secure and robust Internet of Things network that is based in the cloud and meets the particular needs and difficulties presented by the

dynamic and diverse nature of IoT environments.

[10]Z. Li et. al,introduced an architecture makes use of edge computing technology to alleviate server load concerns. This improves the responsiveness and efficiency of the platform while reducing the possibility of overloading the cloud server with enormous amounts of data. Additionally, by including a cloud-native environment into the cloud computing center's design, the platform is optimised for efficient operation and maintenance management. This creative strategy guarantees that the platform is capable of handling the intricacies involved in setting up, running, and maintaining an IoT network over an entire city.The issues associated with traditional IoT platforms are effectively addressed by this city-level IoT integrated platform, which integrates edge computing and cloud-native environments. It reduces server overload, improves flexibility to a variety of IoT devices, and simplifies management, operation, and maintenance procedures in general. All of these features make it a more effective and scalable solution for the changing IoT technology landscape in urban settings.Even if Cloud computing is integrated into traditional IoT platforms for data management, problems occur when handling large datasets or when errors occur in directly uploaded data, which might result in server overload. Large-scale conventional IoT platform development and operation also need a lot of time and money. This study presents a revolutionary city-level IoT integrated platform that aims to seamlessly connect various IoT devices throughout the city as a solution to these difficulties.

CHAPTER 3

METHODOLOGY

3.1 WORKFLOW

The flowchart shows the overall process of the Smart Parking System project. The process starts with the booking of a parking slot. If a slot is booked by a person, a buffer time is given for the person to park their car in the slot. If the buffer time is exceeded, the booking is automatically canceled and it returns to the initial stage of booking the slot. If the car is parked within the buffer time, the slot is reserved for that person and the start time is recorded for payment purposes at the end. The system keeps checking whether the vehicle leaves the parking slot. If the vehicle has not left, the slot remains reserved. If it leaves, the departure time is recorded and used to calculate the amount to be paid for parking. Once the time-based payment is done, the process stops.

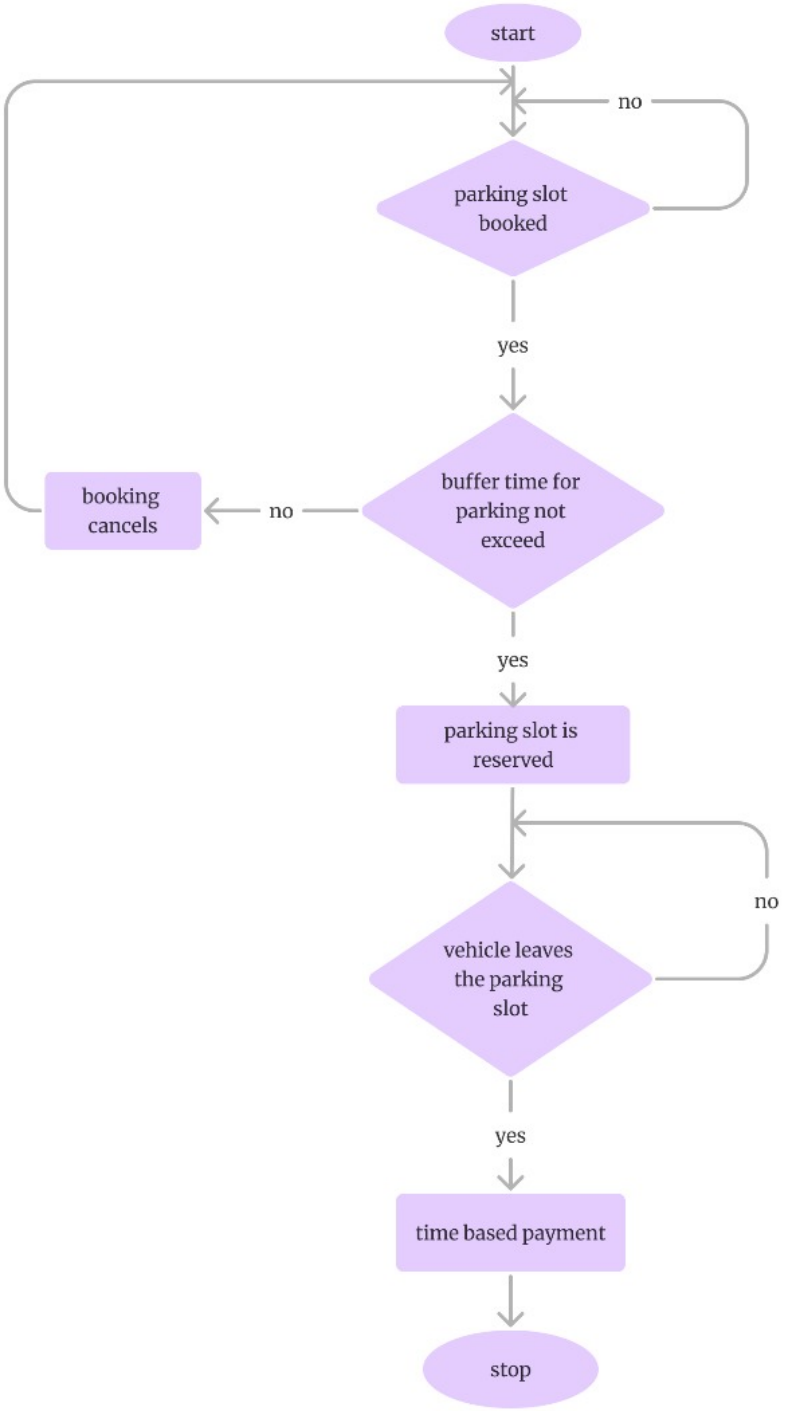


Figure 3.1: Workflow

3.2 BLOCK DIAGRAM

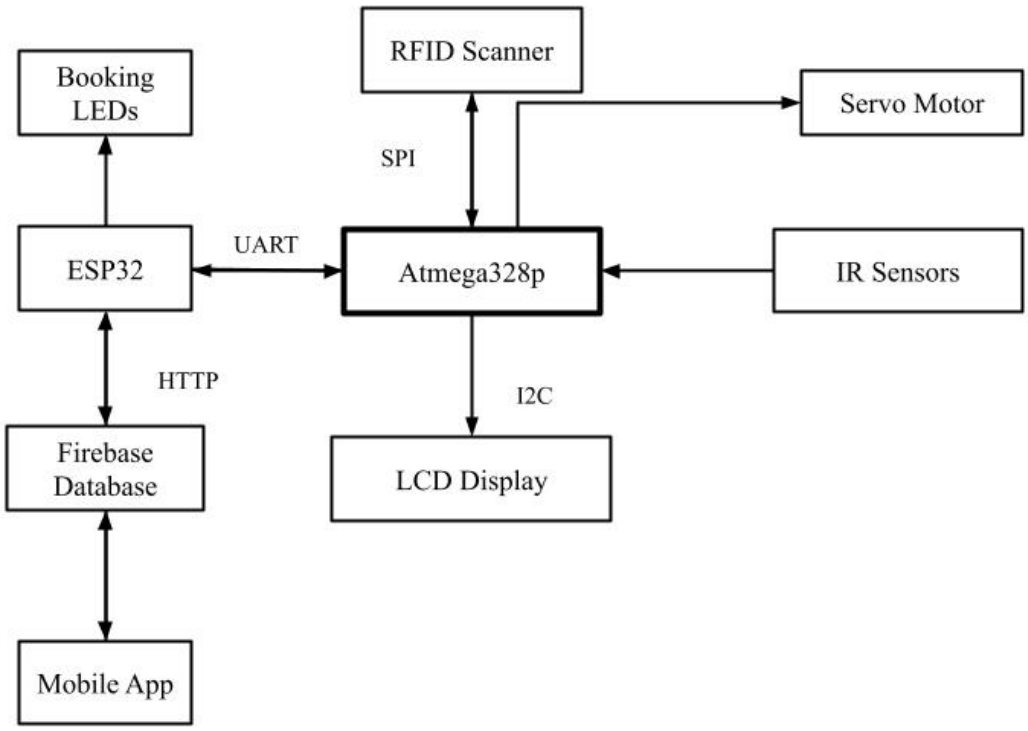


Figure 3.2: Block Diagram

From the block diagram, the servo motor controls the flow of vehicles entering and exiting the parking lot, which is managed by the Atmega. The Atmega communicates with the servo using data from the RFID, which is used for verification. If the verification matches, the servo is activated and opens for entry. The verification is done using an RFID tag and reader. The RFID communicates with the Atmega using the SPI protocol for transferring data. When the vehicle enters the parking slot, the IR sensor detects the presence of the vehicle in the parking slot to track the time that the vehicle spends in that slot. The data from the IR sensor is transferred to the Atmega for identifying empty slots for booking and cancellation purposes. The data received from the IR sensor is used to calculate the rate for parking and is also shared with the ESP32. The Atmega communicates with the ESP32 using the UART protocol, which allows communication between two microcontrollers at different speeds. The data shared by the Atmega with the ESP is then shared with Firebase, which is connected to the mobile application for booking the parking lot. The advantage of using Firebase

is that it helps in real-time data updates to all the devices connected to that Firebase. The mobile application, developed using MIT App Inventor and connected to Firebase, allows the user to book the parking slot, check availability, and cancel the booked slot. This data from the app is updated in Firebase, which is connected to the ESP32. Then, the ESP32 is utilized to control the booking LED to show whether the slot is reserved or not. When the vehicle is leaving the parking slot, the data from the IR sensor is used to calculate the rate, and it is displayed on a liquid crystal display, which is connected to the Atmega using the I2C Protocol. The same RFID used for identification is used to make the payment. After the payment is done, the servo is activated, and it opens for the vehicle to leave the parking lot.

3.3 COMPONENTS USED

3.3.1 ATmega32pu

A well-known microcontroller chip produced by Microchip Technology (previously Atmel Corporation) is the ATmega328PU. Its performance, usability, and portability make it a popular choice for DIY electronics projects and a variety of embedded systems.It is predicated on the well-known for its simplicity and efficiency 8-bit RISC architecture, the AVR.Programme code can be stored in the 32 KB of Flash memory of the ATmega328PU. This enables you to immediately write and save your programme code or firmware onto the chip.Its 2 KB of Static Random Access Memory (SRAM) is utilised to store data and variables while the programme is running.Additionally, the chip has one kilobyte of Electrically Erasable Programmable Read-Only Memory (EEPROM) for non-volatile data storage that must be kept even in the event of a power outage.Although it can be run at lower clock speeds (16 MHz) to save power, the ATmega328PU normally operates at up to 20 MHz.Digital I/O pins, analog-to-digital converters (ADC), serial communication interfaces (UART, SPI, and I2C), timers/counters, and other peripherals are among the many built-in features.Because it has modes with minimal power consumption, battery-powered applications can use it.For projects based on the ATmega328PU, Microchip offers a complete development environment, which includes integrated development environments (IDEs) like Atmel Studio and Arduino IDE.

3.3.2 ESP32

Espressif Systems created the adaptable microcontroller unit (MCU) known as the ESP32.A low-cost, low-power system-on-a-chip (SoC) with dual-mode Bluetooth and Wi-Fi capabilities is called the ESP32.It is the replacement for the well-liked ESP8266 SoC.Applications for the ESP32 include wearable electronics, Internet of Things (IoT) devices, and mobile devices. Both single-core and dual-core versions of Tensilica's 32-bit Xtensa LX6 CPU are available for the ESP32. Built-in support for Wi-Fi (2.4 GHz) is available. The ESP32 has an operational temperature range of -40°C to +125°C, which allows it to function dependably in industrial settings...Designed with energy economy in mind, it uses less power thanks to optimisations made with proprietary software. Filters, power management modules, RF balun, power amplifier, low-noise receive amplifier, and antenna switches are all incorporated into the ESP32.It can reduce communication stack overhead on the primary application processor by operating as a slave device to a host MCU or as a stand-alone system. The Arduino IDE or other development environments can be used to programme the ESP32.

3.3.3 Arduino Uno R3

The Arduino UNO is the best option since it offers the best combination of dependability and flexibility. For those who have never coded before or are just getting started with the Arduino platform, the UNO provides a solid platform for experimentation and project creation. The cornerstone of the Arduino family, the UNO is widely known for its extensive use and well-documented capabilities, which make it a convenient and approachable choice. Its versatility is increased by its smooth interaction with a range of sensors, actuators, and shields, enabling users to confidently take on a number of tasks. Beyond its technical skills, this board is significant because it represents a plethora of instructional materials and a friendly community. The Arduino UNO becomes more than simply a physical tool with the abundance of tutorials, forums, and shared projects available; it becomes a portal to a knowledge-rich and collaborative environment. Because of its intuitive design and active community, it's a great place for both beginners and experts to start learning about electronics and programming. It also fosters a creative and encouraging environment for learning and developing

creative ideas.

3.3.4 IR Sensor

An electrical device called an infrared sensor uses light emission to sense objects in its surroundings. It can detect motion and measure the heat of an item. Generally speaking, objects generate heat radiation in the infrared spectrum. This radiation is detectable by an infrared sensor even if it is invisible to the human eye.The basis for the operation of infrared (IR) sensors is the detection of infrared radiation, an electromagnetic radiation type that lies beyond the visible light spectrum. Usually consisting of an emitter and a receiver, the emitter—often an infrared LED—emits infrared light, and the receiver—often a photodiode—detects the radiation that is reflected or emits infrared. A portion of the light that is emitted is reflected back to the sensor when it comes into contact with an object, and the receiver detects this reflected light. The colour, substance, and distance of the item all affect how much light is reflected. Any alteration in the signal reflects a variation in the presence or distance of an item. The sensor evaluates the incoming signal.Due to their non-contact nature and adaptability in a variety of electronic systems, infrared (IR) sensors are widely used in applications such as motion detectors, proximity sensors, and temperature measurement devices. They can be used actively, where the sensor emits and detects its own infrared light, or passively, where it responds to changes in the infrared radiation emitted by warm objects.

3.3.5 LED

LEDs, or light-emitting diodes, are semiconductor technologies that have completely changed the lighting industry. LEDs use the electroluminescence principle to produce light when an electric current is supplied. A semiconductor material used in LEDs experiences electron mobility, wherein electrons travel from a higher energy area to a lower energy region. Visible light is produced when these electrons recombine with positively charged ”holes” in the material, releasing energy in the form of photons. The colour of the light emitted is determined by the particular semiconductor materials that are employed. Among the many benefits of LEDs are their extended lifespan, energy efficiency, and multicoloured light output. LEDs are widely used in a variety

of applications, from energy-efficient lighting solutions for homes and businesses to indicator lights in electronic gadgets, and they are still a major force behind advances in the lighting and display industries. Continuous developments in LED technology highlight how important a role they play in improving energy efficiency and advancing the development of contemporary lighting systems.

3.3.6 RFID RC522

A well-liked RFID (Radio Frequency Identification) module for reading and writing RFID tags is the RFID RC522. It uses the Serial Peripheral Interface (SPI) protocol to interface with microcontrollers such as Arduino, operating at a frequency of 13.56 MHz.It is compatible with multiple RFID tag types, such as ISO14443A standard MIFARE cards. SPI Interface: Arduino, ESP32, and other comparable platforms can be easily integrated because SPI is the interface used for communication with microcontrollers. An external antenna is not necessary because the module has an integrated antenna. Simple Interface: It is rather easy to use in projects because it offers basic instructions for both reading and writing RFID tags. functions at 3.3V most of the time, which makes it compatible with most microcontroller platforms.

3.3.7 16x2 LCD Display

In electronics projects, a 16x2 LCD (Liquid Crystal Display) module is a typical kind of alphanumeric display. As the name ”16x2” suggests, it is made up of a display screen that can display 16 characters in each of its two rows. There are two rows of 16 characters on the display, for a total of 32 characters.Depending on the controller being used, these displays can typically display symbols, alphanumeric characters, and some special characters. A built-in LED backlight that may be regulated independently of the display itself is a feature of several 16x2 LCD modules. Low-light readability is enhanced by this lighting. The HD44780 controller (or equivalent) serves as the foundation for the 16x2 LCD modules, making interface with microcontrollers such as Arduino easier. The low-level display control and communication are handled by this controller. Usually, a parallel interface is used to interface these displays with microcontrollers; this method necessitates many digital pins for data and control signals. Backpack modules for SPI and I2C are available that reduce the interface to

a few pins, making microcontrollers with a restricted number of GPIO pins easier to use. The majority of 16x2 LCD modules run at 5V, while some can also be used with 3.3V systems.

3.3.8 ESP8266

The widely used and adaptable Wi-Fi module ESP8266 was created by Espressif Systems. It is well-liked among makers, hobbyists, and Internet of Things developers due to its affordability, low power consumption, and simplicity of usage. Due to the ESP8266's integrated Wi-Fi capability, devices can connect to nearby Wi-Fi networks and use the internet to communicate with other devices or services. The ESP8266 is mostly a Wi-Fi module, but it also has a potent 32-bit microcontroller unit (MCU) that can be programmed to do a number of different things, like reading sensors, actuating actuators, and carrying out logic and algorithms.A number of General Purpose Input/Output (GPIO) pins on the module allow it to be interfaced with switches, LEDs, sensors, and other external devices.Many developers may use the Arduino IDE to programme the ESP8266, making it available to a larger audience. Additionally, it supports C and C++ programming with Espressif's ESP8266 SDK. Because of its low power consumption, the ESP8266 is a good choice for low-power or battery-powered applications, even with its impressive capabilities. Owing to its widespread popularity, the ESP8266 has a sizable and vibrant developer community that produces libraries, projects, and tutorials, facilitating the process of learning for novices and helping seasoned developers troubleshoot and solve issues. The ESP8266 module is available in a number of forms factors and feature combinations. The most popular versions are the ESP-01, ESP-12E, and ESP-12F, which differ in terms of form factors and GPIO pin counts.

3.3.9 LM7805 Voltage Regulator IC

A well-known voltage regulator integrated circuit (IC) produced by a number of businesses, including Texas Instruments, is the LM7805. It belongs to the LM78xx series of regulators, which also has regulators with various output voltages. In particular, the LM7805 has a set output voltage of +5 volts. Regardless of variations in input voltage or load conditions, the LM7805 produces a steady output voltage

of +5 volts DC. Typically, it can handle input voltages between 7 and 35 volts DC. To guarantee appropriate regulation, the input voltage must be at least 2.0 volts greater than the output voltage. The highest output current that the LM7805 can consistently produce is contingent upon various factors, including heat dissipation, ambient temperature, and input voltage. Usually, it's about 1 amp, although with the right heat sinking, various versions or designs might enable larger currents. Thermal overload protection integrated into the LM7805 shuts down the device if the internal temperature rises above a safe threshold. This function aids in preventing overheating from damaging the regulator. The voltage dropout of the LM7805 is the minimal voltage differential between the input and output that the regulator needs to function correctly. This dropout voltage for the LM7805 is usually approximately 2.0 volts.With an input pin (VIN), an output pin (VOUT), and a ground pin (GND), the LM7805 is a three-terminal device. It is often offered in TO-220 packages, which facilitate simple mounting and dissipation of heat. In many different electronic circuits and tasks, the LM7805 is utilised to control voltage and offer a steady +5V power supply. Microcontrollers, sensors, and other digital logic components are frequently powered by it.

3.3.10 16MHz Crystal Oscillator

An element used in electronic circuits to produce accurate clock signals at a frequency of 16 megahertz (MHz) is a 16MHz crystal oscillator. Microcontrollers, microprocessors, and other digital circuitry requiring precise time frequently use crystal oscillators. A quartz-based crystal resonator is the main part of a crystal oscillator. When exposed to an electric field, quartz crystals' piezoelectric qualities allow them to precisely oscillate a signal. When it comes to precision and frequency stability, crystal oscillators outperform other oscillator types. In applications like microcontroller-based systems, where exact timing is critical, this stability is vital. The crystal resonator is typically coupled in parallel with capacitors to create a parallel resonant circuit in crystal oscillator setups. A common frequency utilised in a lot of microcontroller and microprocessor circuits is 16MHz. However, crystal oscillators are offered in a range of frequencies, from a few kilohertz to several hundred megahertz, to suit a variety of applications. Numerous electronic devices, such as computers, embedded systems,

communication devices, and precision measuring tools, use crystal oscillators.

3.4 SOFTWARE USED

3.4.1 Arduino IDE

The Arduino IDE is a cutting-edge microcontroller programming environment that is renowned for its versatility and ease of use. Because it is open-source, users may customise the IDE to suit their own requirements and promote collaborative development. The Arduino IDE guarantees accessibility for a wide user base by being compatible with major operating systems, such as Windows, macOS, and Linux. This promotes a diverse and inclusive development community. The interface, which is typified by a simple text editor with capabilities like auto-indentation and syntax highlighting, makes coding easier and can be used by programmers of all skill levels.

The Library Manager, a prominent element of the IDE, simplifies the incorporation of external libraries into projects. Having a collection of prewritten functions and code snippets makes this feature more useful for projects. Furthermore, the Serial Monitor makes real-time communication easier, allowing users to easily monitor and troubleshoot sensor data. The IDE supports a range of Arduino-compatible boards with a separate Board Manager, giving users the freedom to select the hardware that best meets their project requirements. For those who want to learn more about physical computing and electronics, the Arduino IDE is a great resource because it includes integrated examples and a basic debugger. Its vibrant and helpful community, which offers guidance, materials, and other resources, solidifies its reputation as the go-to platform for microcontroller programming.

3.4.2 Firebase

Firebase is a comprehensive platform developed by Google that offers a variety of tools and services for building and managing web and mobile applications. It provides developers with a wide range of features to help them develop high-quality apps more efficiently. Firebase offers a NoSQL cloud database that allows developers to store and synchronize data in real-time across multiple clients. It's particularly useful for building collaborative applications and multiplayer games. Firebase provides authentication services, enabling developers to easily implement user authentication using email/password, phone number, social logins (e.g., Google, Facebook, Twitter), and more. Firestore is Firebase's newer, more scalable NoSQL database, offering features like real-time updates, offline support, and powerful querying capabilities. It's designed to scale with your application as it grows. Firebase offers cloud storage solutions for storing and serving user-generated content like images, videos, and other files. It integrates seamlessly with other Firebase services: Firebase Hosting allows developers to deploy web apps and static content quickly and securely to Google's global content delivery network (CDN) directly from the Firebase console. Cloud Functions: Firebase Cloud Functions lets developers run backend code in response to events triggered by Firebase features and HTTPS requests. It's a serverless compute platform that scales automatically. Firebase Analytics provides detailed insights into user behavior and app usage, helping developers understand how users interact with their apps and make data-driven decisions. Performance Monitoring: Firebase Performance Monitoring helps developers identify performance issues in their apps by tracking key metrics like app startup time, screen rendering time, and network performance. Firebase Remote Config allows developers to customize the behavior and appearance of their apps without requiring an app update. It enables A/B testing, staged rollouts, and targeted app configurations. Firebase Cloud Messaging (FCM) enables developers to send targeted push notifications and messages to users across platforms, including Android, iOS, and web. It is a powerful and versatile platform that can be used for a wide range of applications, from simple mobile apps to complex web applications and games.

3.4.3 EasyEDA

Engineers can design and prototype electronic circuits and PCBs (Printed Circuit Boards) online with EasyEDA, an Electronic Design Automation (EDA) tool package. It offers a variety of features and an intuitive UI to make the design process easier. With EasyEDA, users can drag and drop components onto a canvas to construct electronic schematics. It provides the option to develop bespoke components in addition to a library of frequently used components. With EasyEDA, users may generate physical representations of their circuits by routing traces and arranging components to design PCB layouts. Features like design rule verification, auto-routing, and 3D visualisation are offered by the programm... A built-in SPICE (Simulation Programme with Integrated Circuit Emphasis) simulator in EasyEDA enables users to model and analyse circuit behaviour prior to prototyping. This enhances circuit performance and helps find possible problems. Multiple people can collaborate on the same project at once using EasyEDA's support for these functionalities. Users can work together in real time, exchange projects with peers, and comment on designs. A sizable user base of EasyEDA contributes to its collection of PCB footprints, symbols, and components.It is simple to locate and utilise components in designs thanks to this large collection. Platforms and services provided by third parties, such as JLCPCB for PCB fabrication and LCSC for component procurement, are easily integrated with EasyEDA. Users can easily send their designs to these providers for assembly and production. EasyEDA is a web-based tool that can be accessed from any device that has a web browser and an internet connection.

3.4.4 MIT App Inventer

Without having to know how to programme in conventional languages like Java or Swift, users can create mobile applications for Android devices with the help of MIT App Inventor, a web-based integrated development environment (IDE). After being developed by Google at first, the Massachusetts Institute of Technology (MIT) took over its development. Building apps with App Inventor is done through a visual drag-and-drop interface. By choosing and placing elements on a design canvas, such as buttons, text boxes, and sensors, users can construct applications. They then

use blocks that reflect programming logic to specify these components' behaviours. Users programme their apps by combining blocks that represent actions, events, and procedures rather than manually coding code.The process of creating apps is made easier and more approachable for novices using this block-based method. A vast array of features and components, such as UI elements, multimedia capabilities, sensors (like GPS and accelerometer), and networking choices (like Bluetooth and web APIs), are available for users to incorporate into their apps using MIT App Inventor. Users can install and run their apps on an Android device by scanning a QR code, or they can use the built-in emulator to test their apps instantaneously. This makes it possible to debug and iterate quickly during the development process. A thriving user community exists for MIT App Inventor, where members provide tutorials, guidelines, and project examples.To assist customers in getting started and understanding app development concepts, the platform also provides a wealth of documentation and educational materials. Users can access data and functionality from online services including databases, social networking platforms, and cloud storage by integrating App Inventor with external web services and APIs.

3.4.5 Thinkercad

With a dual focus on electronic circuit simulation and 3D design, TinkerCAD emerges as a flexible and user-friendly online platform that appeals to a wide range of users, from enthusiasts to schools. Its ease of use is derived from its simple interface, which lets users create and simulate electrical circuits with ease by simply dragging and dropping electronic components. The platform's special strength is its ability to essentially link these parts and deliver instantaneous simulations of circuit behaviour, making it a priceless resource for electronics education and experimentation.

TinkerCAD's feature set extends beyond electronics; it also includes 3D design and modelling. The platform's use to computer-aided design (CAD) extends beyond electronics to include the ability for users to build and visualise complex three-dimensional constructions. Because TinkerCAD runs on the cloud, it is universally accessible and can be used from any internet-connected device. In addition to being more convenient, its cloud-based design promotes a collaborative atmosphere where

users can easily exchange ideas and projects. TinkerCAD is an amazing resource for novices and hobbyists who want to study and improve their abilities in electronics and 3D design. Its comprehensive approach combines these two subjects in an approachable online environment.

3.4.6 Blynk

One well-liked platform for creating Internet of Things (IoT) projects and apps is Blynk. Using smartphones or tablets, it offers a convenient way to monitor and control linked hardware equipment. With the help of Blynk's mobile app, users may design unique user interfaces (UIs) for their IoT projects on iOS and Android smartphones. The application offers drag-and-drop capabilities for including gauges, buttons, sliders, and more widgets into the user interface. The cloud-based architecture that powers Blynk facilitates communication between the mobile app and linked physical gadgets.In addition to receiving real-time data updates from sensors or other peripherals, users can remotely manage their equipment. Numerous hardware platforms and microcontrollers, such as Arduino, Raspberry Pi, ESP8266, ESP32, and others, are supported by Blynk. Users can use a variety of communication protocols, including Bluetooth, Ethernet, Wi-Fi, and GSM, to link their devices to the Blynk platform.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 SIMULATION RESULT

4.1.1 Simulation of IR Sensor

when we start doing our project important to start with object detection using IR sensor.we employed TinkerCad for simulating of the sensor's behavior for the designed for a parking slot.IR sensor has three pins Vcc,GND,OUT.When the object detects the output pin will give 5v.In TinkerCad there is only PIR sensor and ultrasonic sensor for object detection.For the simulations we use PIR sensor which is similar to IR senor that use in hardware implementation. In this simulation we connected the output of the IR is connected to the digital pin 10 of Arduino Uno.The digital pin 10 of Arduino is initialize as input pin.Also the digital pin 2 of Arduino Uno is initialize as output pin.We connect led to digital pin 2(fig6.1) .We had written the code such that when IR sensor detects the vehicle the LED will be ON,if not the LED will be off.

4.1.2 Simulation Five Parking Slot

when a vehicle occupies the parking slot, the sensor activates a sequence causing the LED to flash in a distinctive RED pattern. This serves as a visual cue for users or monitoring systems, indicating the current occupancy status of the parking slot.

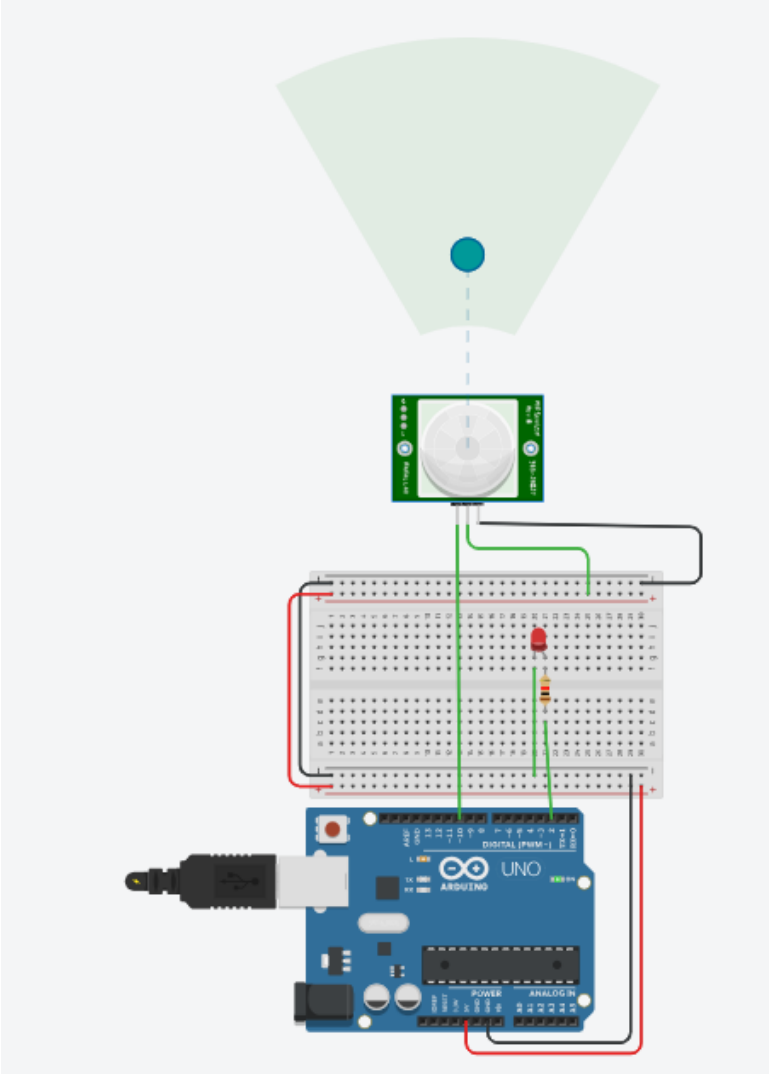


Figure 4.1: simulation of IR sensor

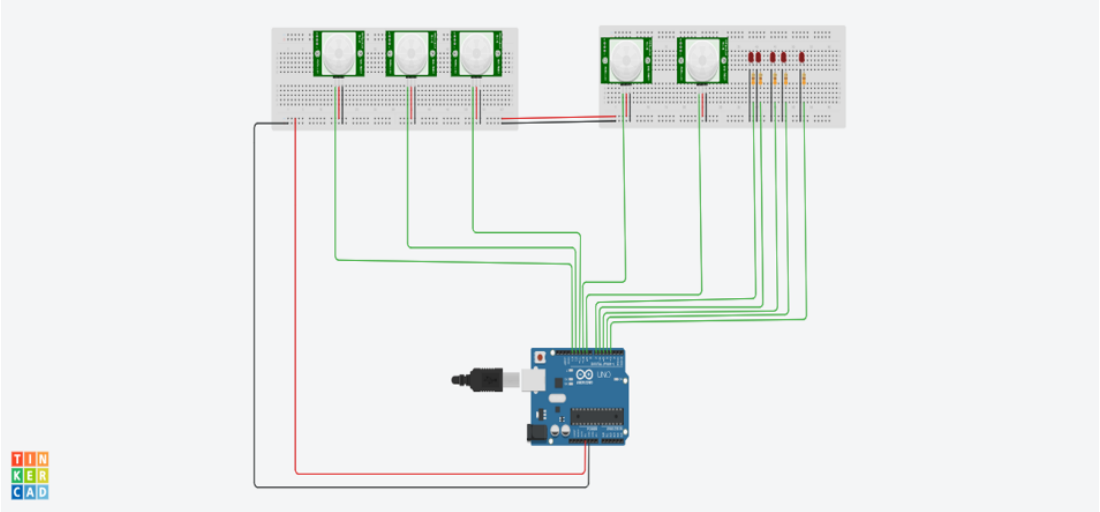


Figure 4.2: Simulation Result

Conversely, as the vehicle exits the parking slot, the sensor responds accordingly, and the LED alters its flashing pattern to OFF. This color change acts as a clear signal, notifying that the parking slot is now available for another vehicle. The incorporation with RED representing occupancy and LED off denoting vacancy, enhances the user experience by providing a quick and intuitive method for discerning the parking space's status.

4.1.3 Simulation of 16x2 LCD Screen

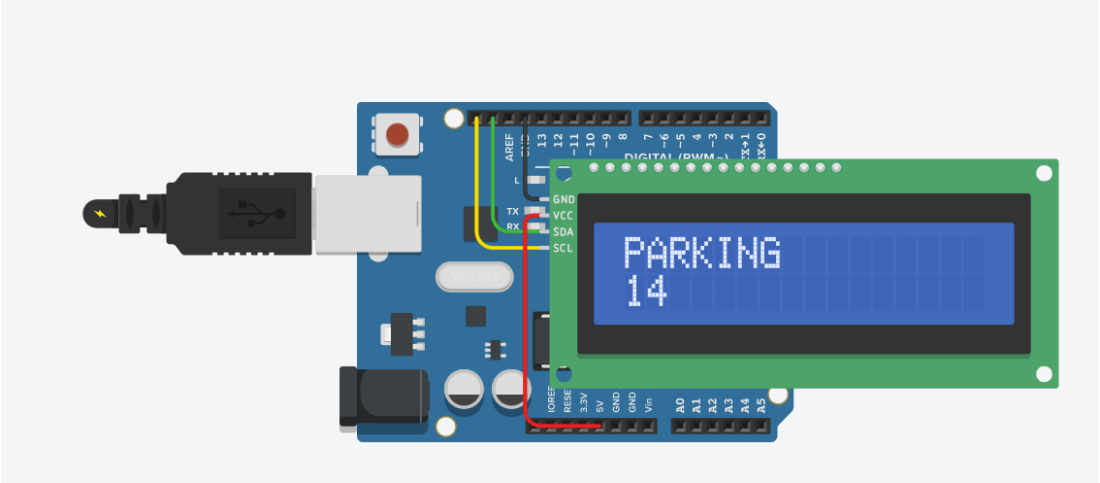


Figure 4.3: Simulation of 16x2 LCD Screen

The 16x2 LCD display is connected to Arduino Uno is by I2C module.So that the number wire connected to the Arduino Uno is reduced to four.In code we include "wire.h" library for implementing the I2C communication between the display and the Arduino Uno. We also include "LiquidCrystal.h" library to call the functions initializing the LCD display, function control the LCD display,function to on the light of the LCD and function to print on the LCD display.

The above figure we printed 'PARKING' on line one and '14' on line two.At first we place the cursor at the at the beginning of the display and print 'PARKING'.Then we place the cursor at the beginning of the line two of the display then print '14'.

4.1.4 Simulation of Servo Motor

In this simulation we implemented the working of servo motor.Servo motor has three pins:signal pin(orange colour wire),ground pin(brown colourwire),Vcc pin(red colour

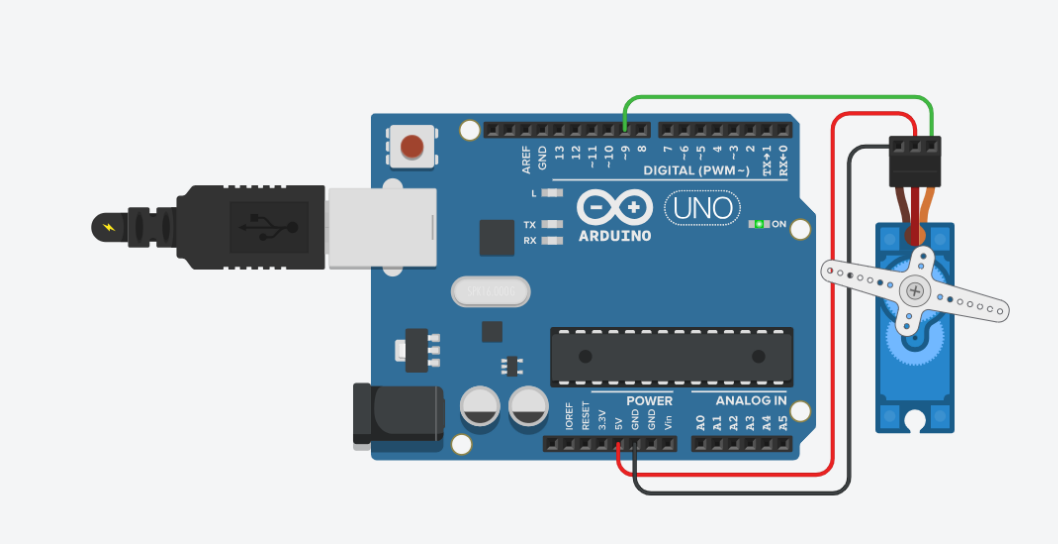


Figure 4.4: Simulation of Servo Motor

wire).The Vcc pin and the ground pin of the servo motor is connected to the 5v pin and ground pin of the Arduino Uno.The rotation is of servo is done by sending pulse width modulation signal to the signal pin of the servo.The pulse width determines the amount of rotation of the servo motor.

To implement this in the code we include the library Servo to the program.We connect the servo motor to the digital pin 9,which support PWM.We created a object for the library Servo so that we can call different functions like attach to initialise the pin to start digital modulation and write function to determine the width of PWM signal which determine the amount of rotation in degree.In the above simulation we rotated the servo to 120 degree.

4.2 INTERFACING OF SENSORS WITH ARDUINO

4.2.1 Arduino with IR Sensor

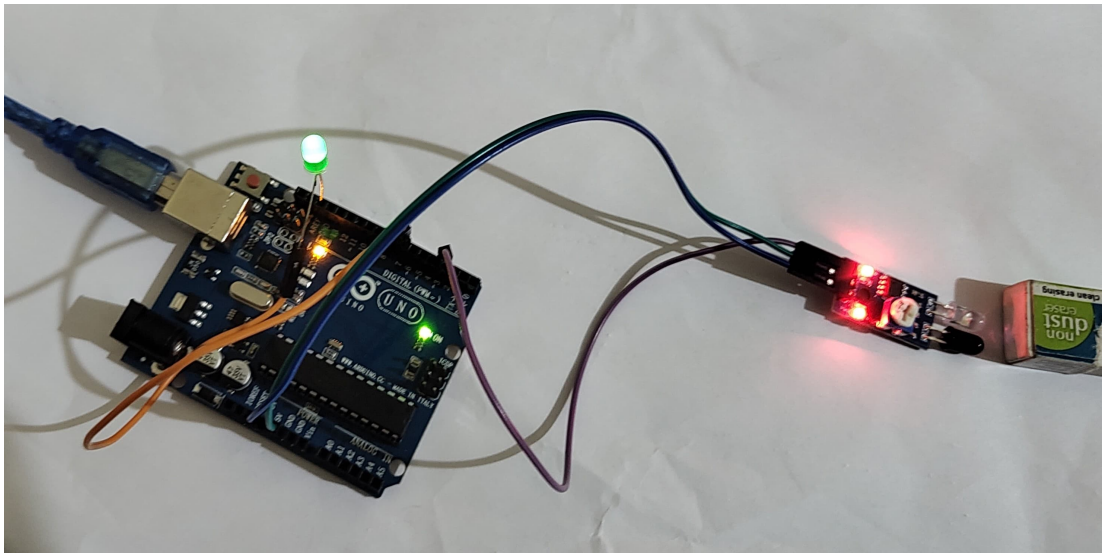


Figure 4.5: Interfacing of Arduino and IR Sensor

The hardware implementation interfacing of Arduino with IR sensor.We have done the the simulation and we use the same code here to implement this.In the we can see that when IR sensor detects the object the LED will be on else the LED will be off.By do this we can implement a parking slot.

4.2.2 Arduino with RC522

We implemented the connection of RC522, the RFID card reader with done with help of SPI protocol.The SCK pin of RC522 is connected to digital pin 13,MISO(Master In Slave Out) pin of RC522 is connected to digital pin 12,MISO(Master In Slave Out) pin of RC522 is connected to digital pin 11,the SDA pin of RC522 is connected to the digital pin 10,the rest pin of RC522 is connected to digital pin 9,the Vcc pin of RC522 is connected to 3.3V pin of the Arduino.Also the ground of RC522 should be connected to ground of the Arduino.

Before we start doing a program we should know the unique card ID number each RFID card.This number is in hexadecimal.We should first know the this ID.To run the sample code to get this ID and for calling the the functions to receive the the values from the RFID scanner RC522 we include a library ”RC522.h”.

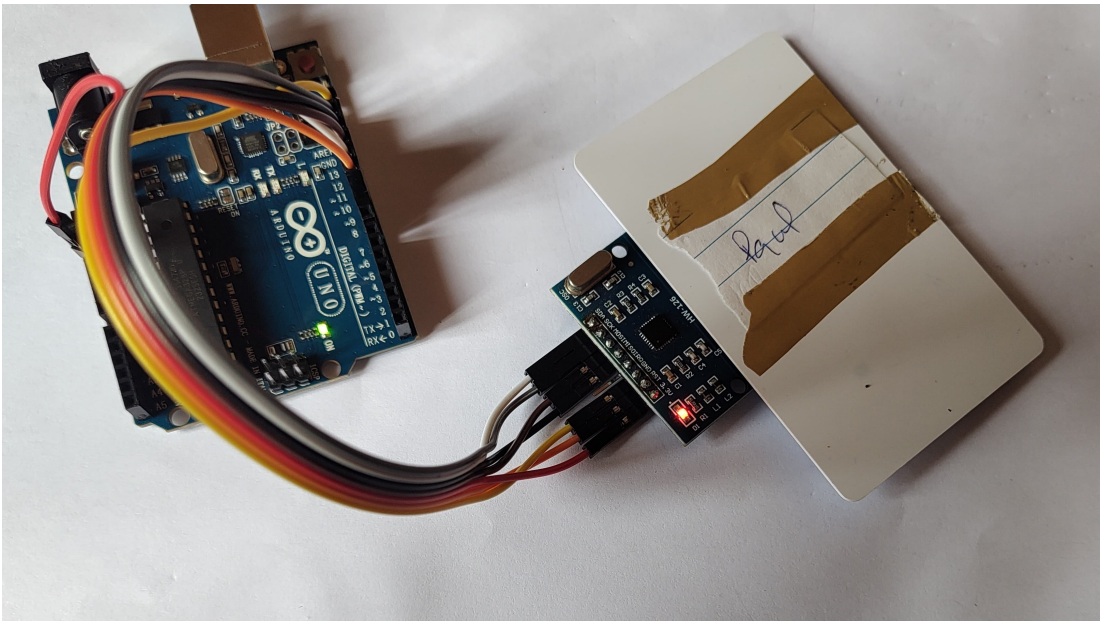


Figure 4.6: Interfacing of Arduino and RC522

```
Card UID: 33 06 45 FB
MIFARE 1KB
 FF FF FF FF FF FF
Block 0: 33 06 45 FB 8B 08 04 00 62 63 64 65 66 67 68 69

Card UID: C3 6F 45 FB
MIFARE 1KB
 FF FF FF FF FF FF
Block 0: C3 6F 45 FB 12 08 04 00 62 63 64 65 66 67 68 69

Card UID: 43 CF 50 FB
MIFARE 1KB
 FF FF FF FF FF FF
Block 0: 43 CF 50 FB 27 08 04 00 62 63 64 65 66 67 68 69

Card UID: 33 25 EB 1B
MIFARE 1KB
 FF FF FF FF FF FF
Block 0: 33 25 EB 1B E6 08 04 00 62 63 64 65 66 67 68 69

Card UID: 7C AB FA 37
MIFARE 1KB
 FF FF FF FF FF FF
Block 0: 7C AB FA 37 1A 08 04 00 62 63 64 65 66 67 68 69
```

Figure 4.7: RFID Card ID

The figure 6.7 shows the Card ID of the five RFID Card that that we are using .From this data only we get the card ID which we will assign to each user.So that we can uniquely identify each user and their individual payment. The RC522 works with SPI protocol so that should include ”SPI.h” library to call the necessary functions to carry of the data transfer through SPI protocol.

4.2.3 Arduino With 16x2 LCD Display

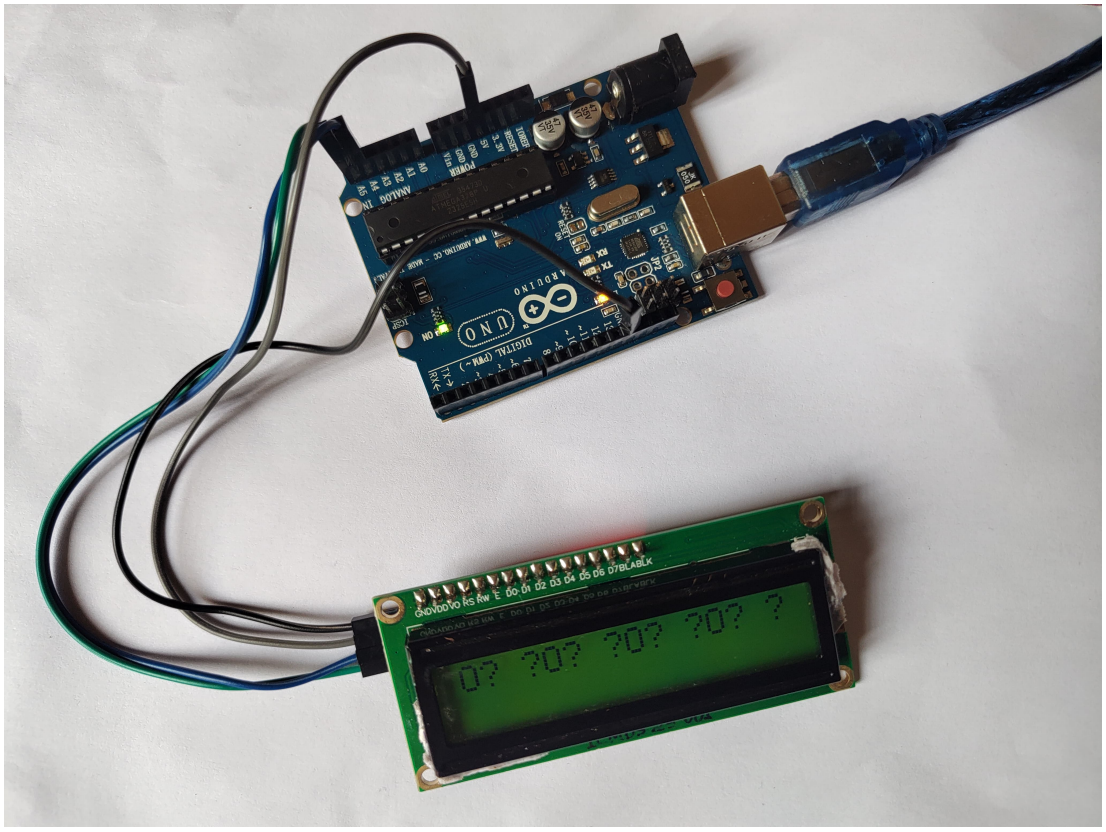


Figure 4.8: Arduino With 16x2 LCD Display with Issue

The figure 6.8 shows the result where we dirtectly enter the same code that we have done in the simulation which is an issue.After some research we found that for implementing communication via I2C protocol we need to know the address of the device.Here the device is 16x2 LCD display.By running a test code with the function to return the address of the 16x2 LCD display from the Library we get the adress as 0x27 for this display(usually it will be 0x16 or 0x27). After resloving this issue we get the same result as in simulation which is shown in the figure6.9

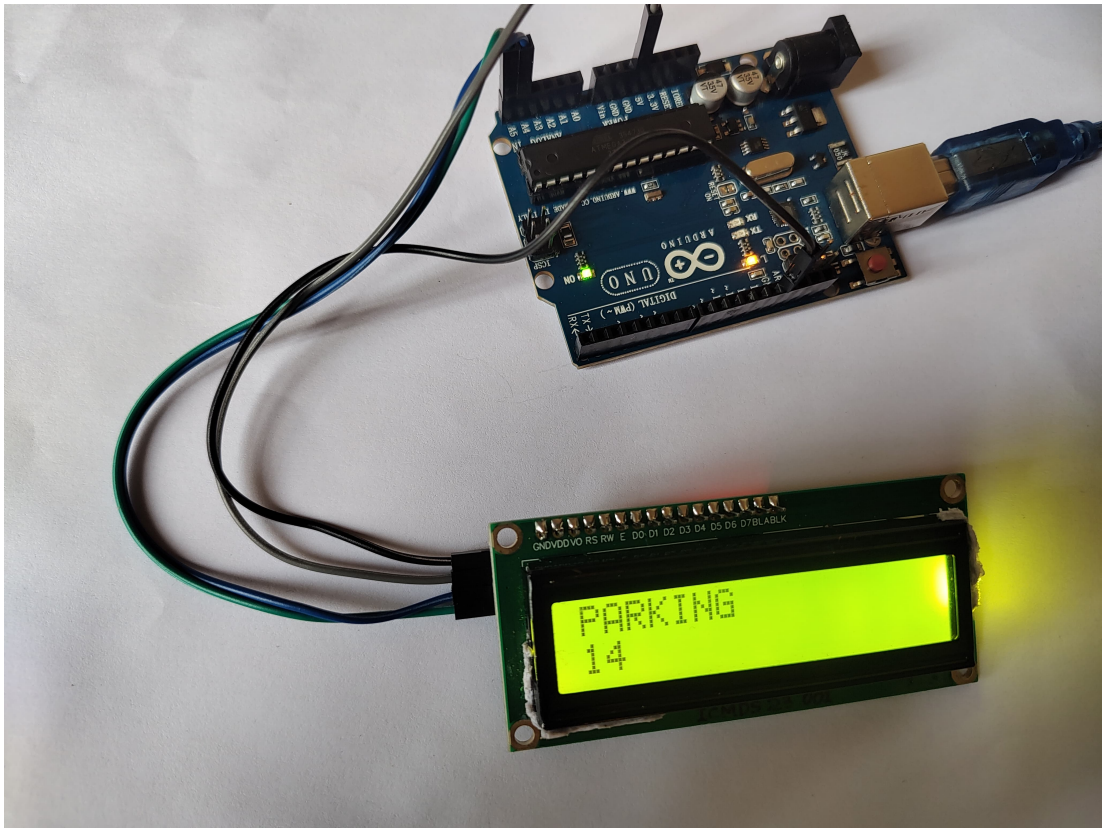


Figure 4.9: Arduino With 16x2 LCD Display without Issue

4.2.4 Arduino With Servo Motor

We implemented the hardware of the simulation of the servo motor.Here we rotated the servo motor by 120 degree .We were able to get exactly the same output as the simulated output

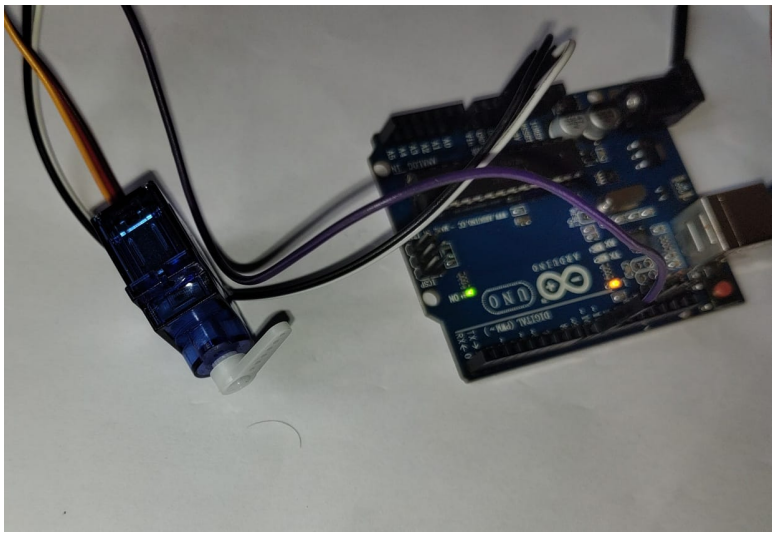


Figure 4.10: Arduino With Servo Motor

4.2.5 Arduino With Servo Motor and RC522

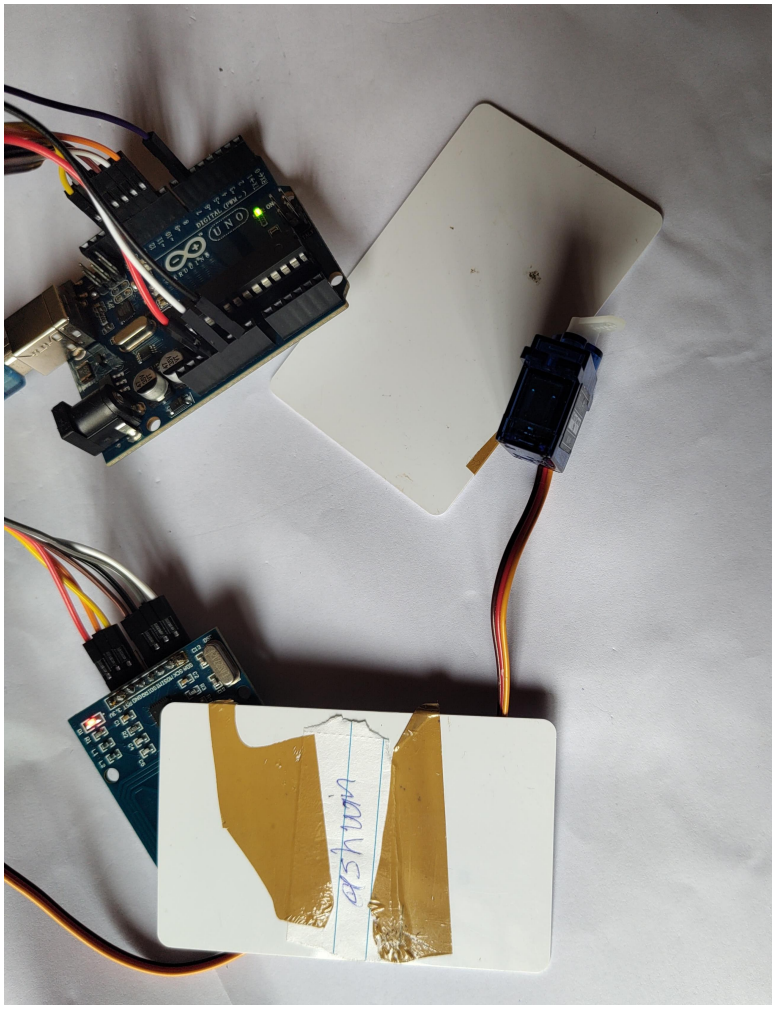


Figure 4.11: Arduino With Servo Motor and RC522

The servo rotates 180 degrees to open the gate in the event that a user with a valid RFID tag is identified, and 90 degrees to close the gate in the event that no valid tag is detected.As we covered in the simulation section, the servo motor and the RC522 RFID scanner are connected to Arduino.The sole distinction is that pin 6—which permits digital modulation—is connected to the servo motor's signal pin.

Here, we use the RC522 scanner to continuously check for the presence of RFID tags.Move the servo motor 180 degrees to open the gate if a valid tag is detected.Move the servo motor to close the gate (to its starting position of 90 degree) if no valid tag is detected.

4.3 Arduino with Servo Motor,RC522 and Display

4.3.1 Gate Opening and Closing

Here we expand the above result by additionally adding 16x2LCD display.When the verified user is comes with RFID.The RC522 scanner reads the data and if the card ID is verified the servo will rotates to 180 degree

The RC522 scanner will look for a new card. If the RC522 scanner detect the presence on new card then the Arduino will verify the card ID of that card if it matches ,the servo will rotates 180 degree else the servo will not rotates(We set the initial position of the Servo motor as 90 degree)

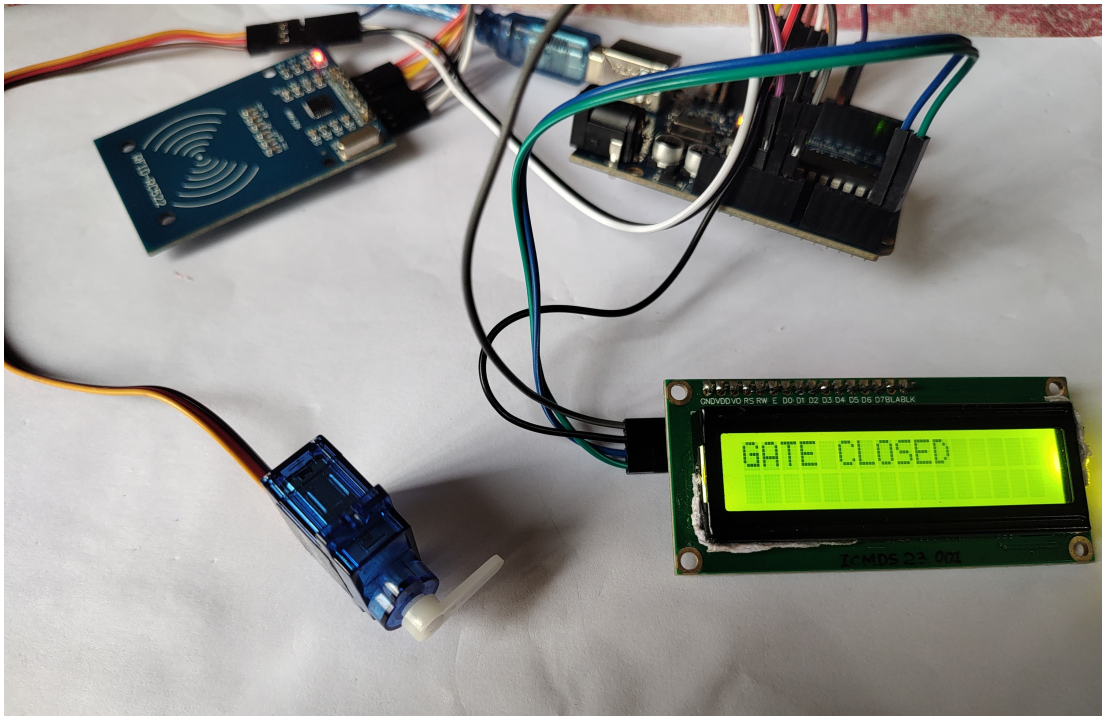


Figure 4.12: Gate Closed

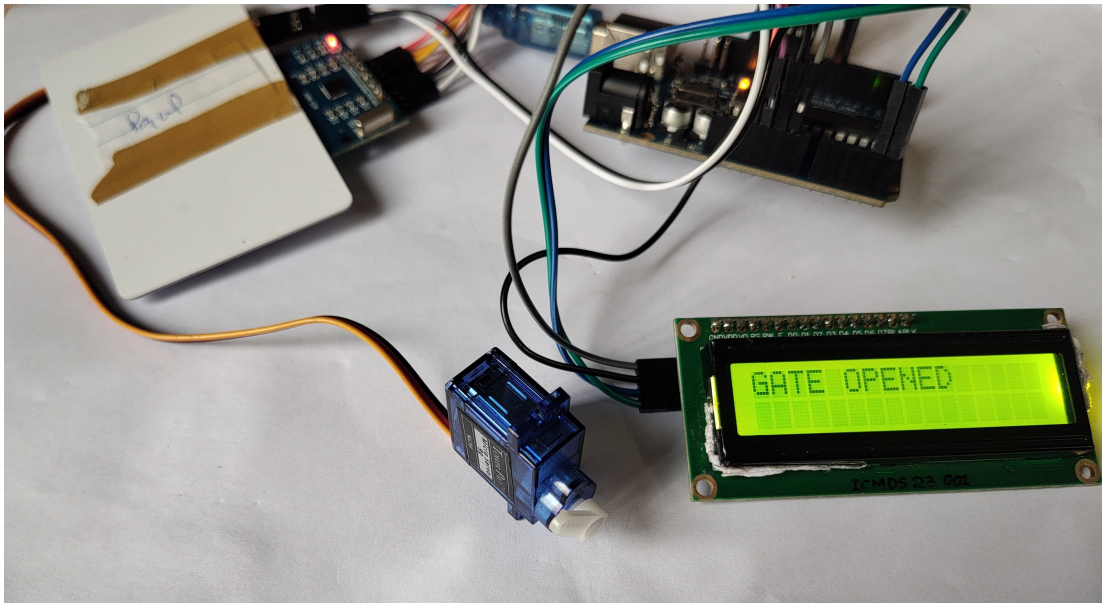


Figure 4.13: Gate Opened

4.3.2 Parking Occupancy

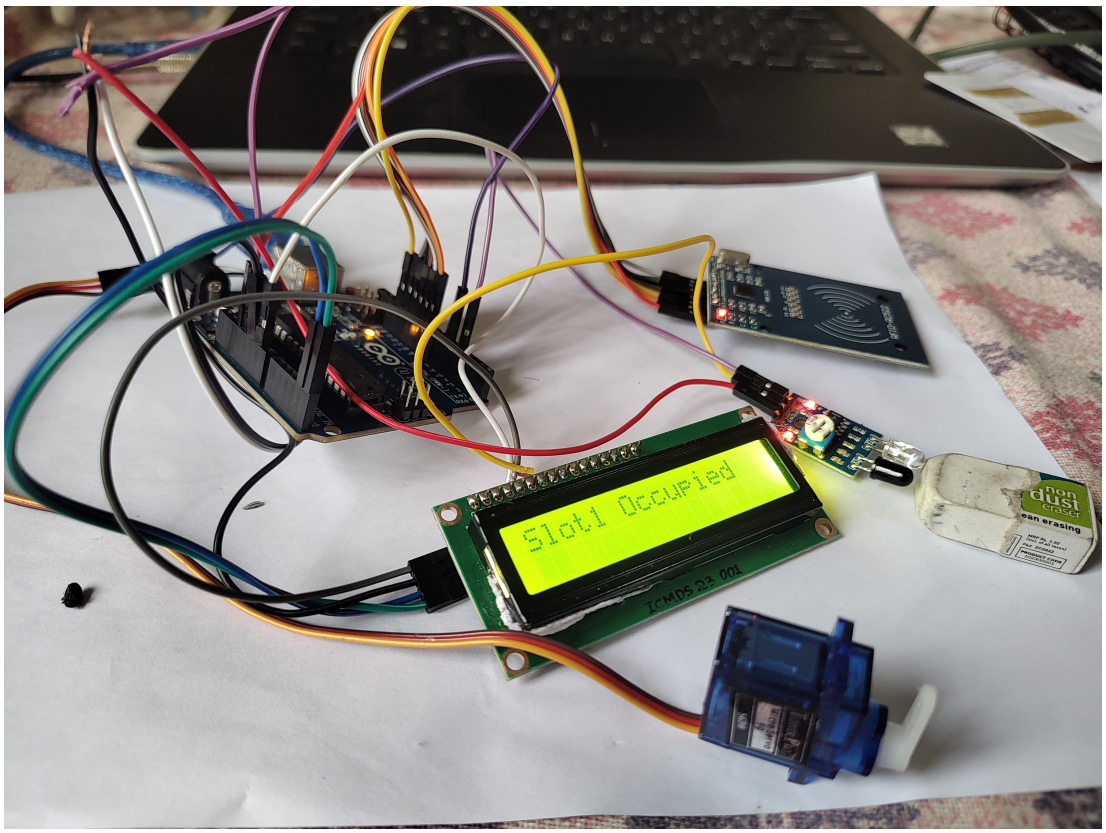


Figure 4.14: Slot Occupied with Issue

Here we try to implement the parking area with one parking slot. The vehicle can enter the parking slot using the RFID card at the entrance.We are implementing for

five user.The user can enter the parking area using the RFID card.If the card verifies the servo will rotate to open position(180 degree).Then after the user enters the servo will rotates back to close position(90 degree).

But from the figure6.14 we can see that the servo is not moving to closed postion,which is an issue

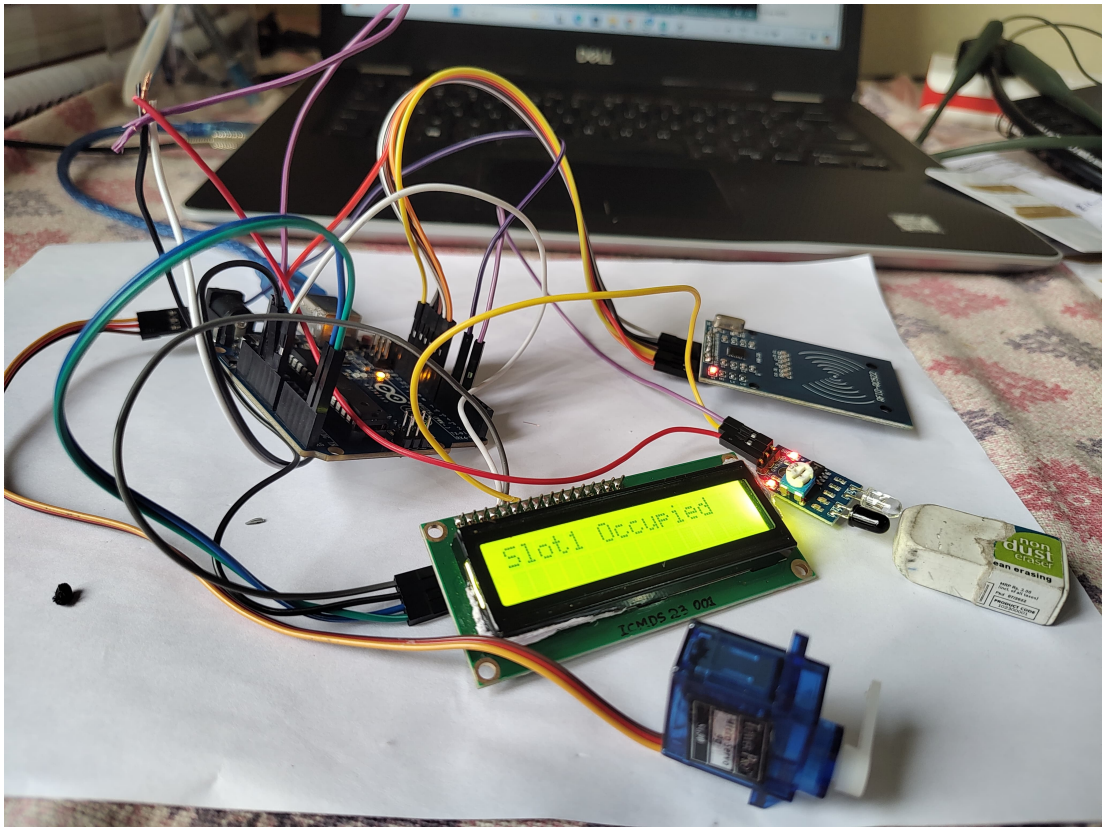


Figure 4.15: Slot Occupied without Issue

By debugging the code we are able solve that issue by adding a flag variable which if the user enters the parking area.We also check if this flag variable is flagged or not along with other condition.The resolved output is shown in figure6.15

If the is user in parking slot other users cannot enter the parking area.The servo motor will be in closed position

4.3.3 Calculation of Parking Time

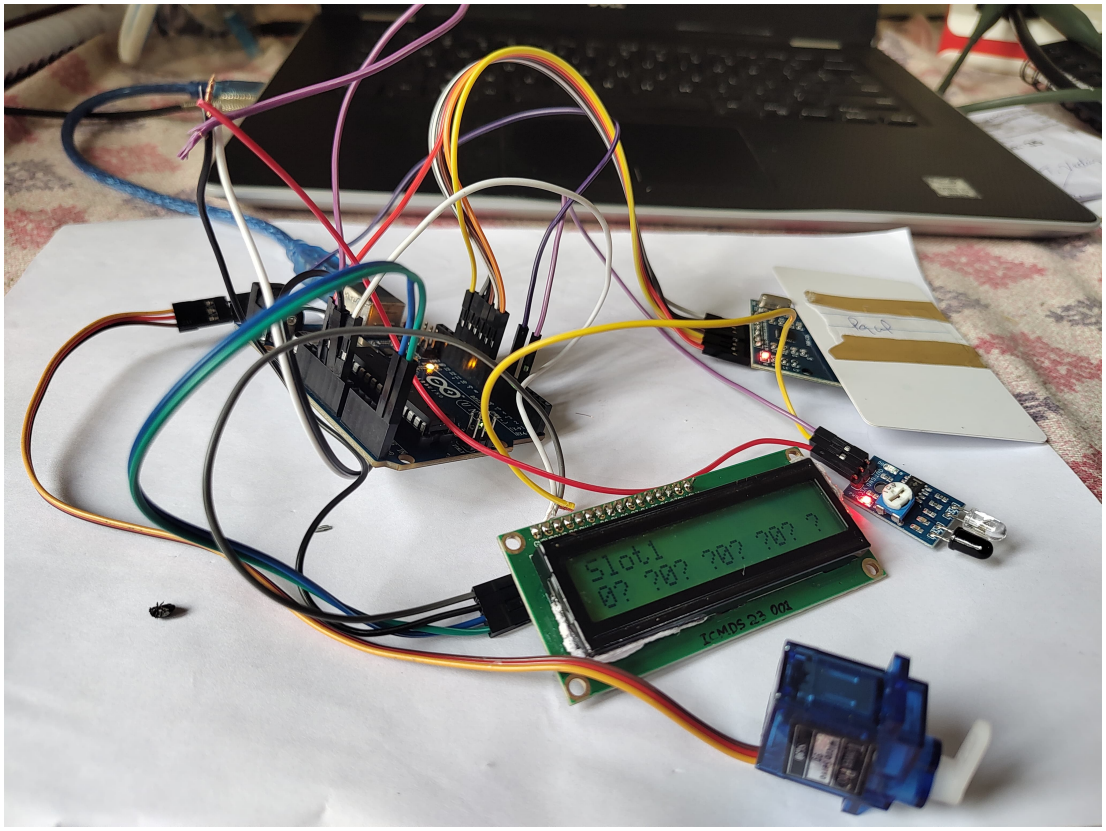


Figure 4.16: Time Displayed with Issue

Here we had tried to implement the payment for the parking to the user based on the time the have parked.In this prototype we giving 1 min free and the payment amount will calculate after 1min.We use millis () function to determine current status of the program timer and the return value of the function is in millisecond.By using that function we get start time when the IR sensor is on (user parked the car).We also get stop time when user exits the parking slot.From this start time and the stop time we can calculate the total time the parked the vehicle.Which will help us to determine the parking fee.

In figure 6.16 the line two display is not printing the data correctly.While debugging we found out that the value we a trying to print line two is a integer datatype but the display supports only string.So we typecaste the the payment amount variable to string.Thus resloved the issue

The figure 6.17 show the amount before 1 min and the figure 6.18 shows the amount after 3min.

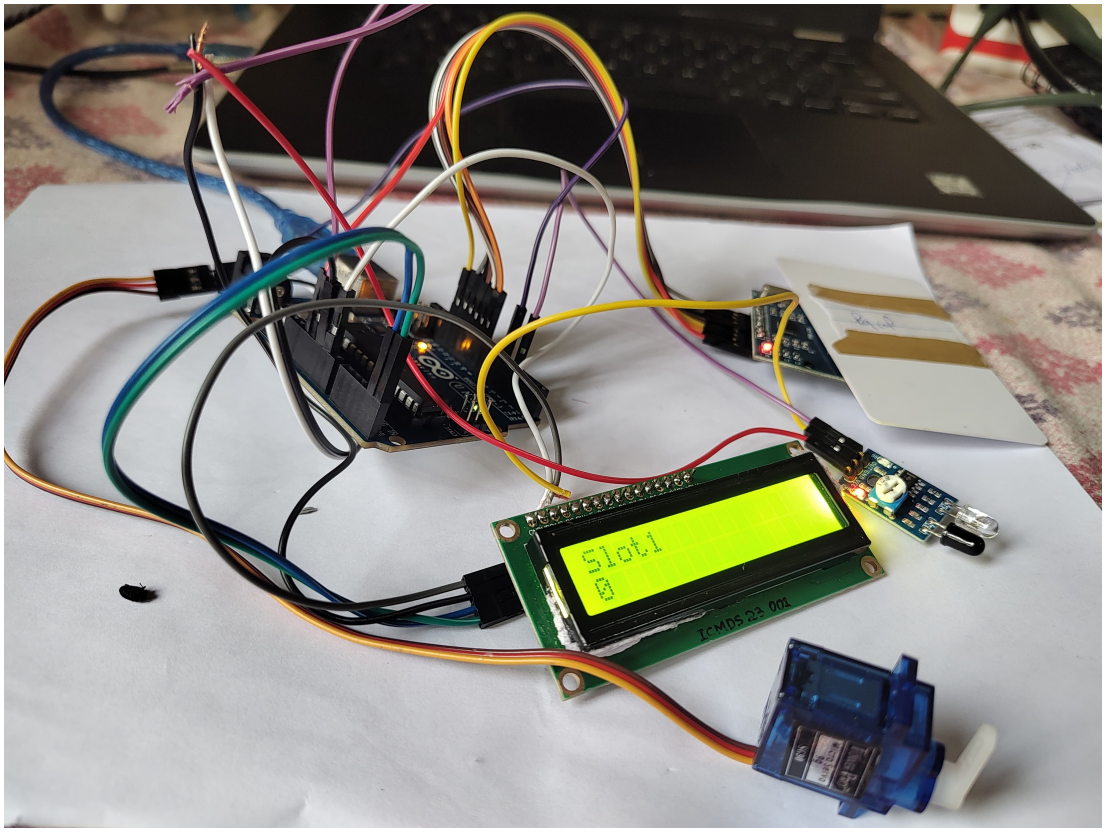


Figure 4.17: Time Displayed without Issue 1

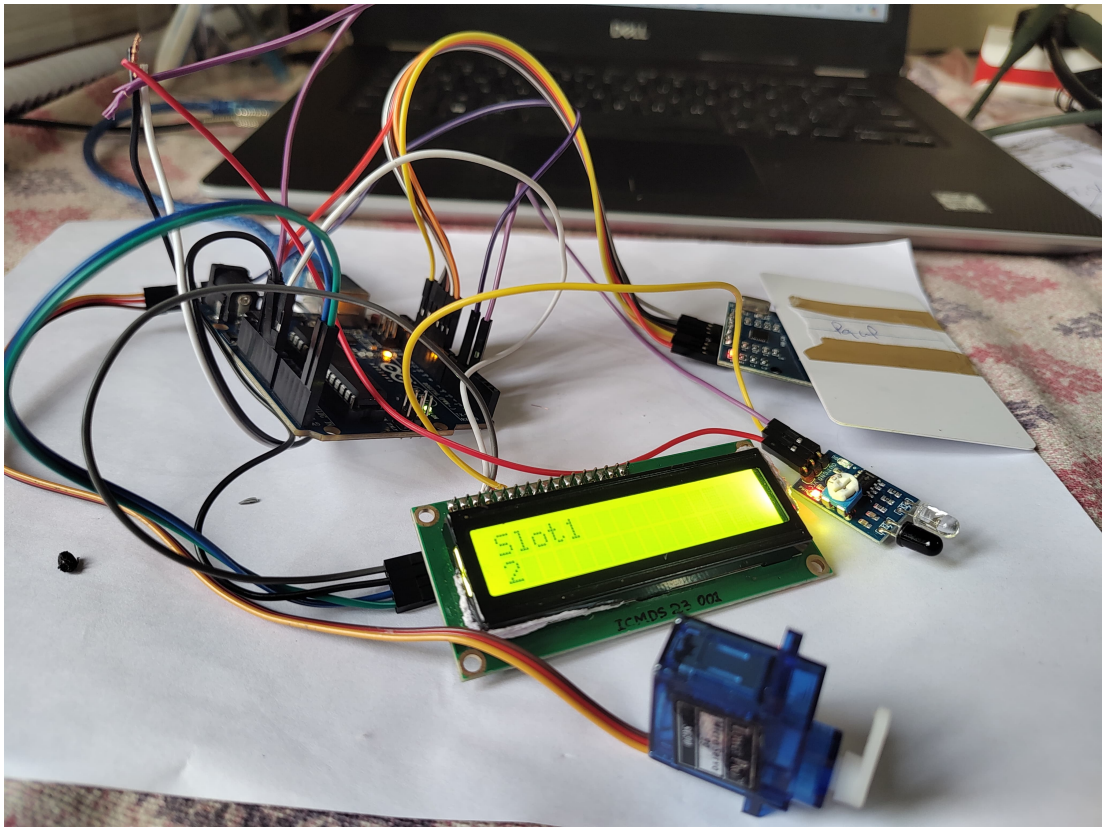


Figure 4.18: Time Displayed without Issue 1

4.4 SENDING AND RECIVING DATA FROM CLOUD

4.4.1 Sending IR Sensor value to Blynk Cloud

when a vehicle occupies the parking slot, the sensor activates a sequence causing the LED to flash in a distinctive RED pattern. This serves as a visual cue for users or monitoring systems, indicating the current occupancy status of the parking slot.

Conversely, as the vehicle exits the parking slot, the sensor responds accordingly, and the LED alters its flashing pattern to GREEN. This color change acts as a clear signal, notifying that the parking slot is now available for another vehicle. The incorporation of a dual-color LED system, with RED representing occupancy and GREEN denoting vacancy, enhances the user experience by providing a quick and intuitive method for discerning the parking space's status.

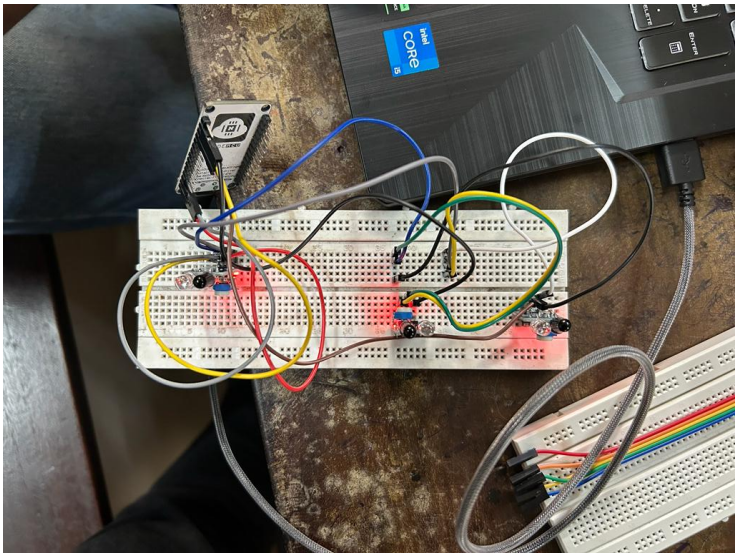


Figure 4.19: Esp8266 and IR sensors

In this prototype we implement the realtime avaliablity of parking slot and its booking.If the vehicle approaches the Vout IR sensor become 1 and it transferred to cloud(fig7.5).The switches in blynk web dashboard is for booking if press it the booking indication light in the prototype will on until the vehicle moves away from the IR sensor.

We send the parking slot data to Blynk Cloud using ESP8266 which enables the WiFi connectivity to our project.We can the cloud using the cloud URL and the

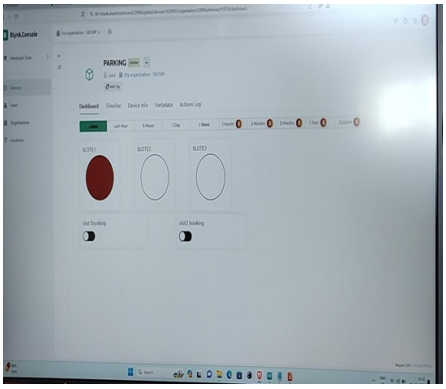


Figure 4.20: Realtime Data in Blynk Console

authentication key for getting authentication

4.4.2 User Interface

The user interface is made using MIT App Inventer which a website used to make moble app.This app is connected to Firebase database via database URL and the authentication key

Figure 4.21: App Interface 1

We should provide the user name which is linked to user Id.This user Id is sent to the fire base database while booking the parking slot

In the above figure the user 'paul' with the user Id '5' booked the parking slot2

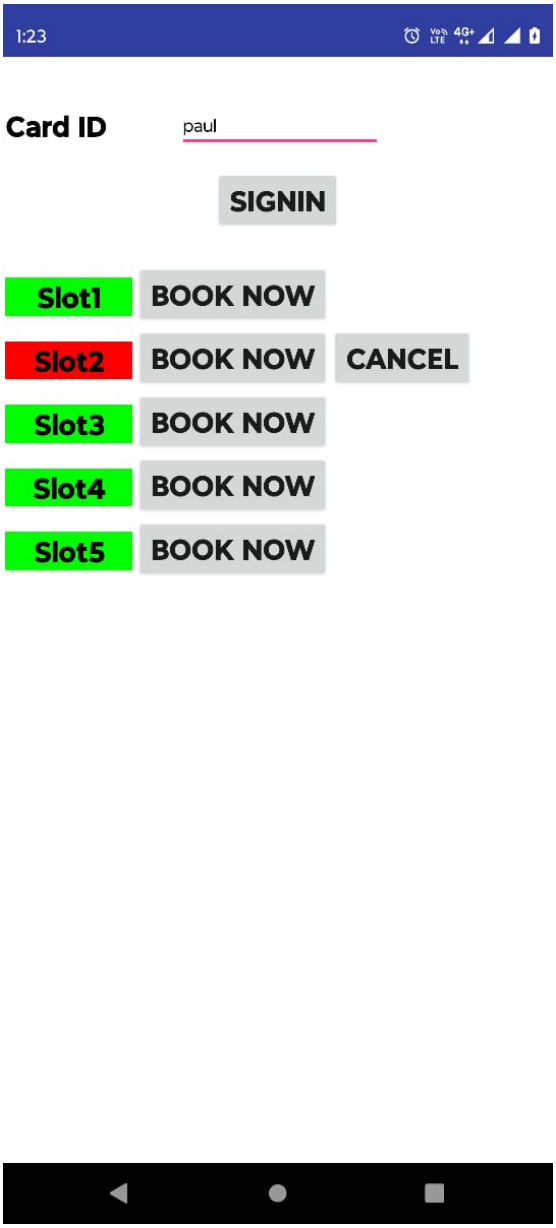


Figure 4.22: App Interface 2

In this figure the user 'arun' with user Id '1' booked parking slot 3.User'arun' cannot book the slot 2 because slot 2 is already booked by user 'paul'

The above figure shows the firebase database.B!,B2,B3,B4,B5 represents parking slots and the key value in the data base are the userId

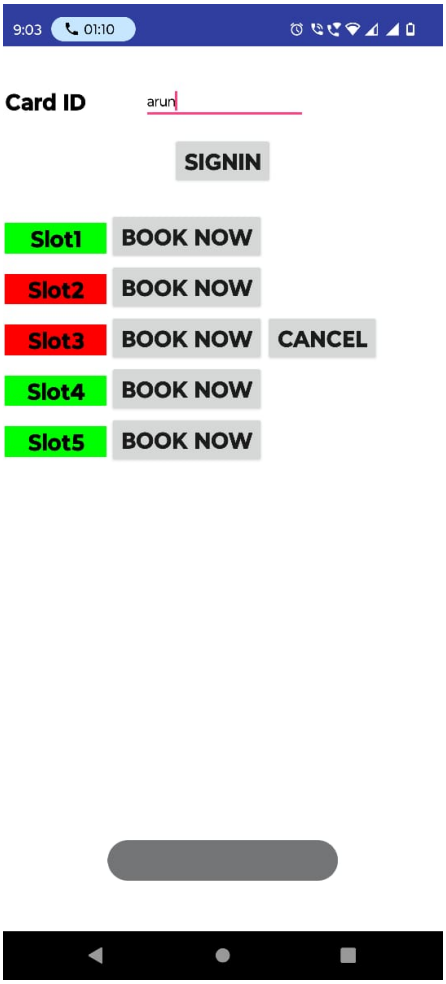


Figure 4.23: App Interfac 3

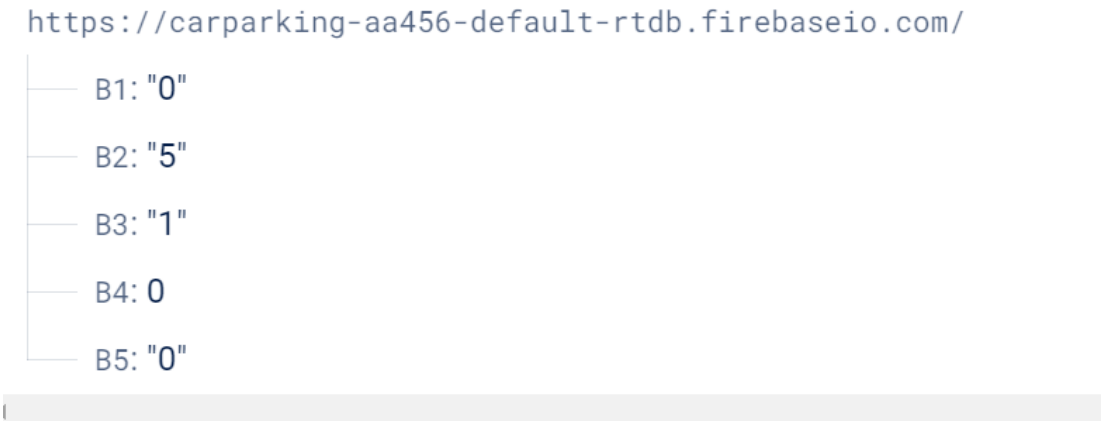


Figure 4.24: App interface 4

4.4.3 Connecting ESP32 to Firebase

We decided to upgrade the esp8266 with its successor esp32.The esp 32 access the real time data at regular intervals to check if any updates occurs at the database(figure6.27).the esp32 access the firebase base database using the database URL which is shown

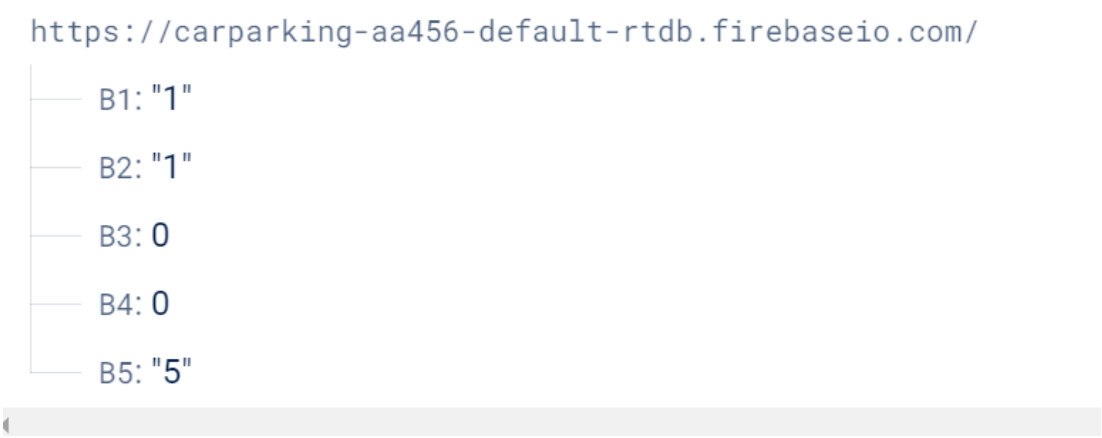


Figure 4.25: Realtime Database 1

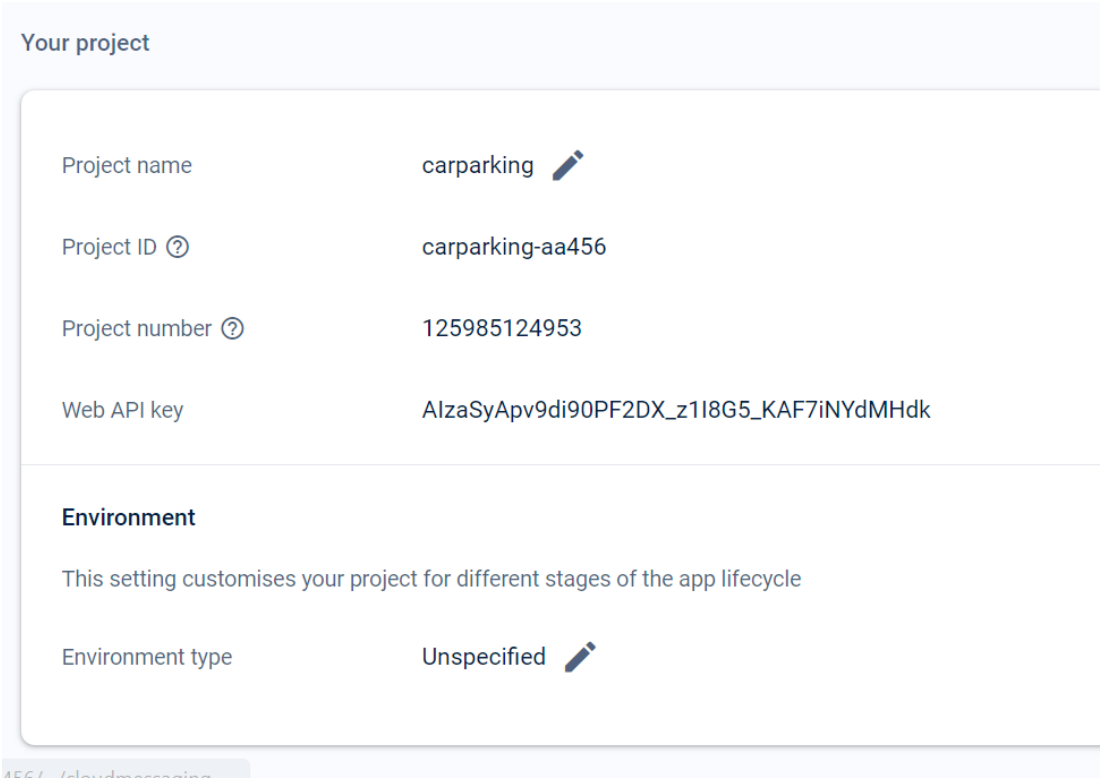


Figure 4.26: Realtime Database 2

figure6.25.The authentication is done using API Key which is shown in the figure 6.26 as web API Key.Only with this API key the esp can access the database.ESP can send and receive data from the firebase when request to access the database is authenticated ,which is shown the figure6.28

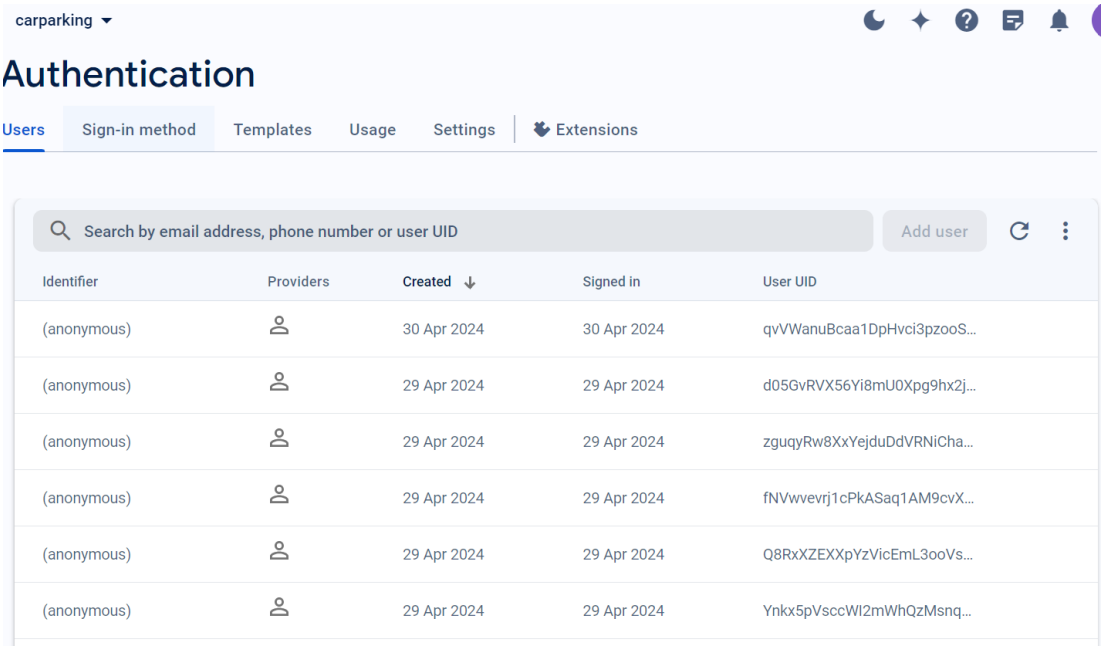


Figure 4.27: Realtime Database 3

Figure 4.28: Realtime Database 4

4.5 PCB DESIGN

4.5.1 Initial PCB Design

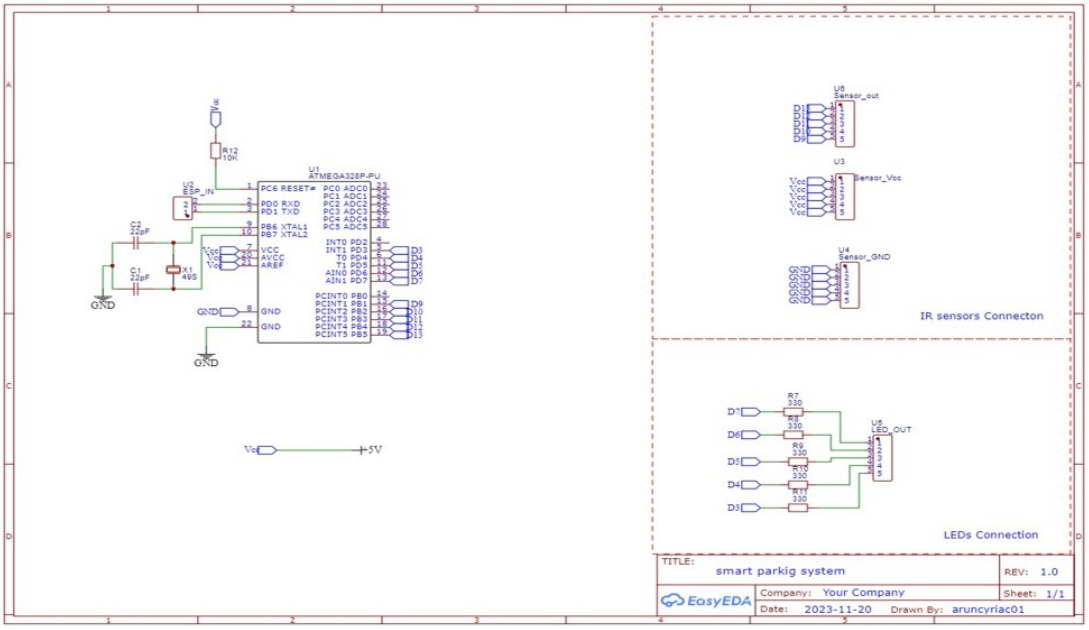


Figure 4.29: Initial PCB Schematic

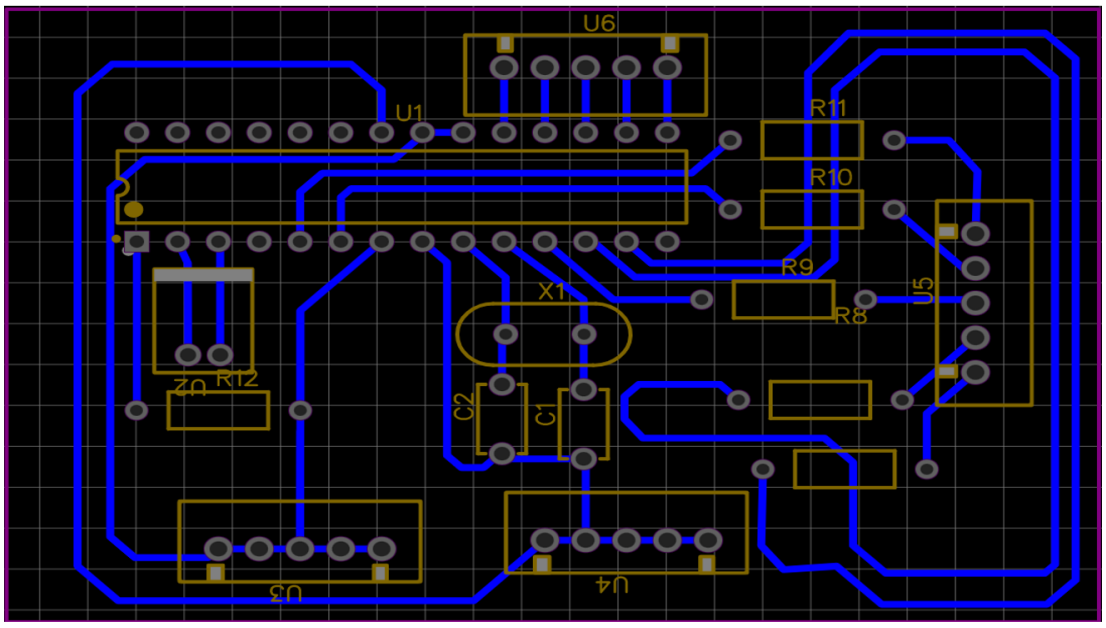


Figure 4.30: Initial PCB Layout

From above hardware implementation simulations we come across the PCB design,the schematic is show in the figure 6.29 and the single layer PCB is shown in the figure.

Here we use ATmega328 as the main controller.Pins D3-D7 is connected to header pins for LED connections U5,which indicate booking.The data pin of the IR sensor is connected to pins D9-D13 via header U4, the Vcc to header U3 and the ground to header U6.To receive and send data to cloud we connect TX and RX of ATmega328 to RX and TX pins of NodeMCU.The crystal oscillator with 16MHz with parallel capacitor provide the processor speed. The figure shows the PCB that we made from the above schematic and layout

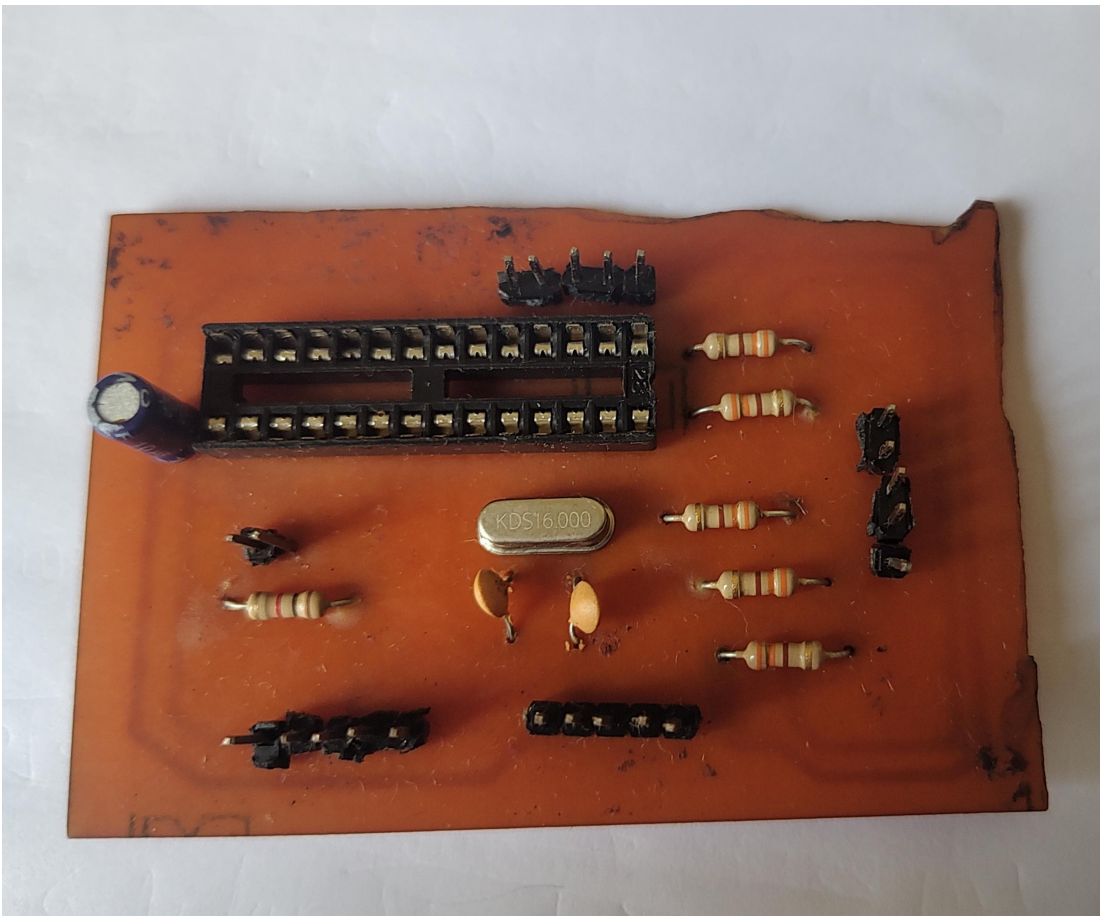


Figure 4.31: Initial PCB

4.6 FINAL PCB DESIGN

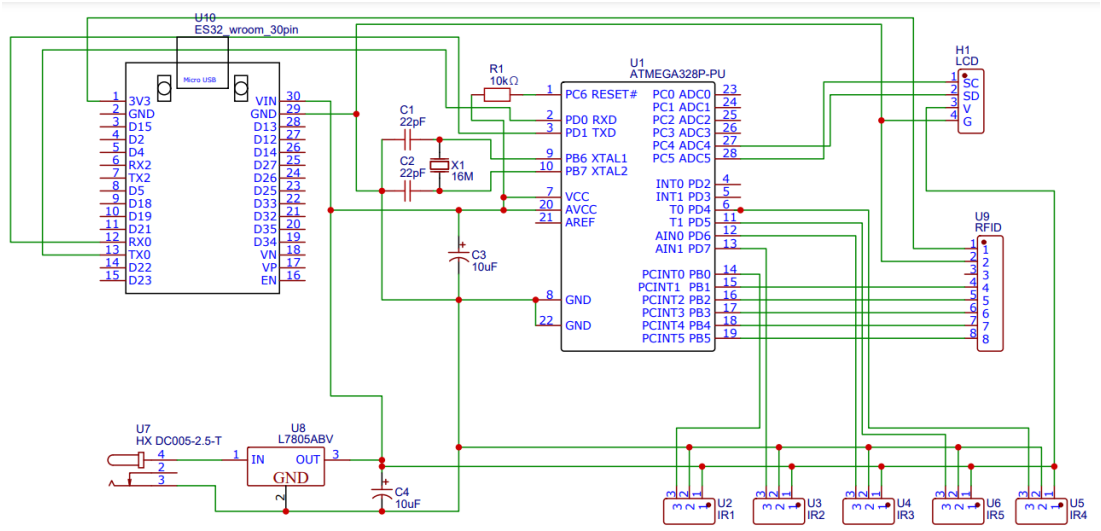


Figure 4.32: Final Schematic

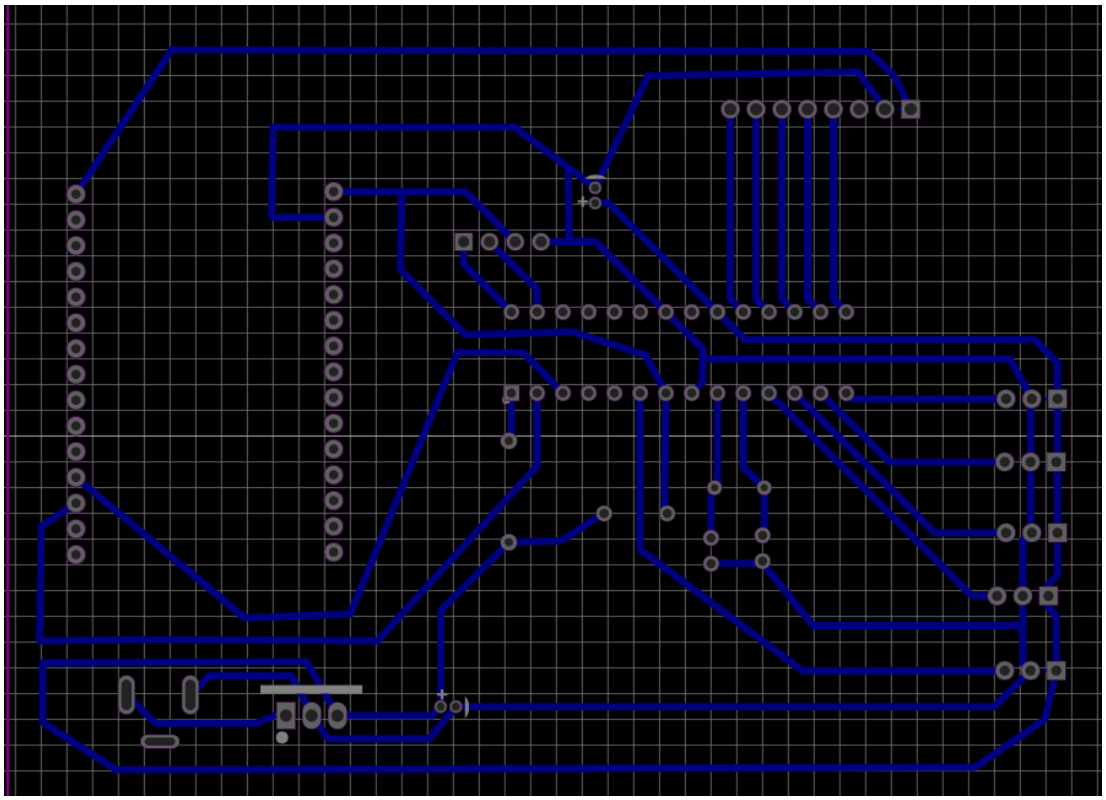


Figure 4.33: Final PCB Layout

The initial PCB works but has some issues like we cannot put the ESP32 into the board.Also we are taking the Vin from microcontroller.In the final schematic and layout we were able to implement the changes like adding LCD display,servo motor,ESP32 into PCB.

4.6.1 Final PCB

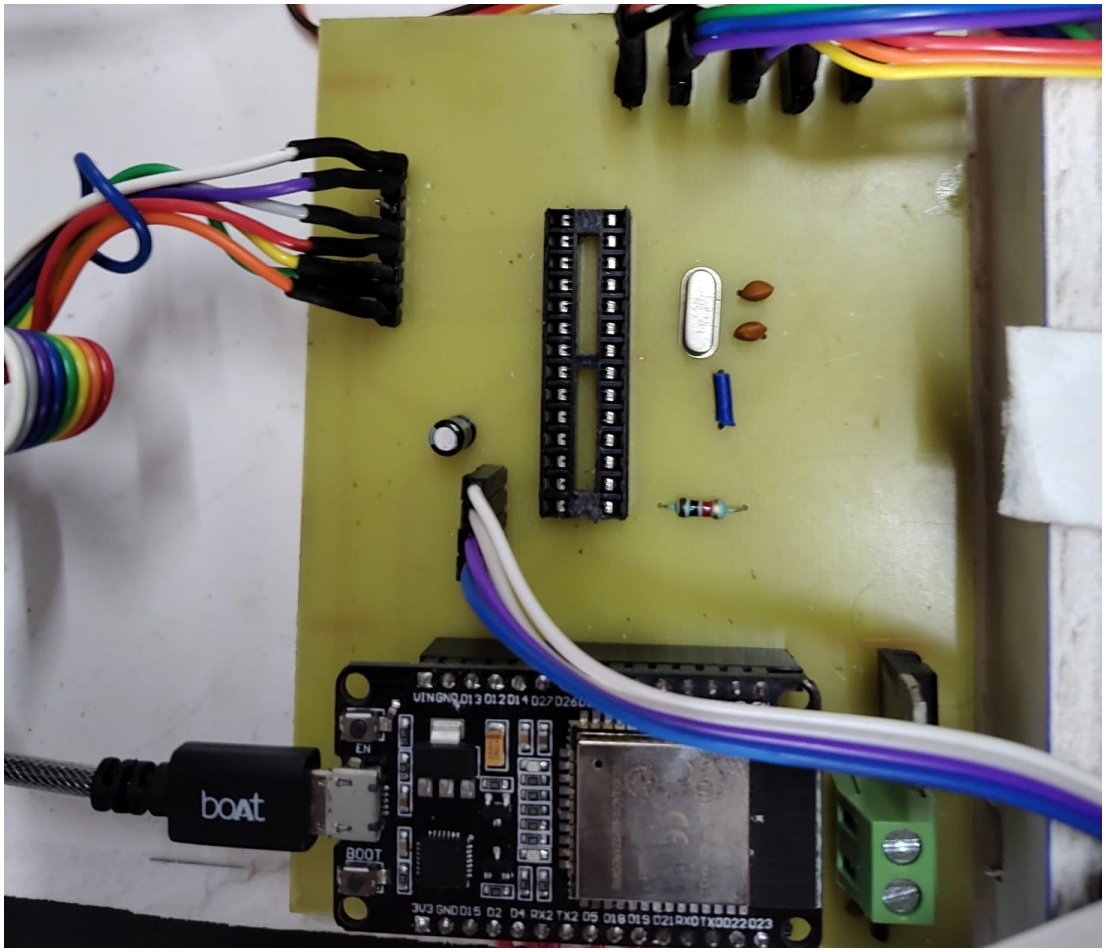


Figure 4.34: Final PCB

We implemented the final PCB which is shown in figure.In the PCB we add a wire through the top of the PCB to make the layout single layer.This from the pin 7 of the ATmega32 and the connection from the electrolytic capacitor.In this final PCB we have made power supply circuit using LM7805.We also checked output voltage of the power supply using multimeter.We had connected Vcc to to RST pin ATmega32 via 10k resistor.

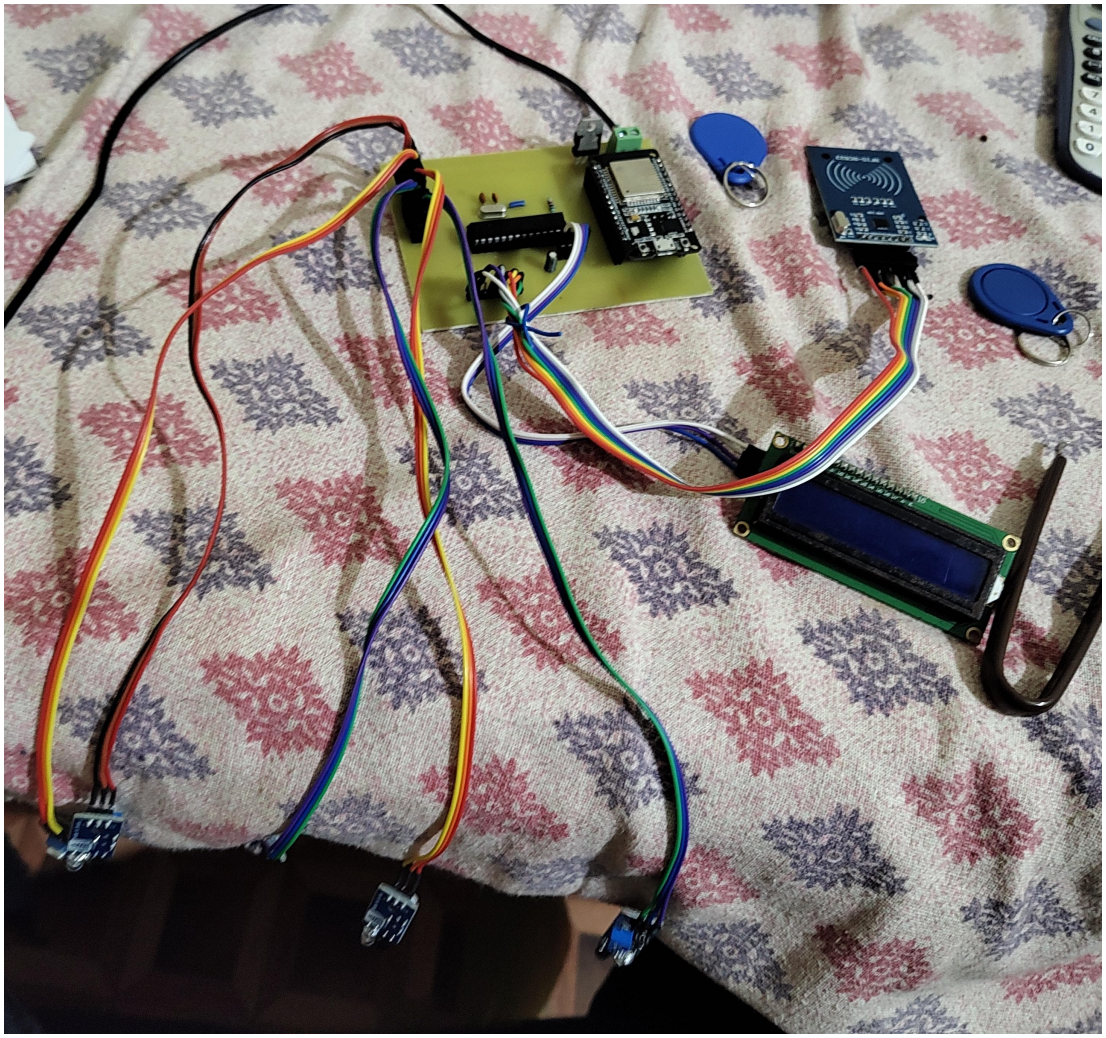


Figure 4.35: Final PCB Testing 1

At this point we faced an issue that slot five is not showing any response means,The detection of vehicle in the parking slot is not showing.So first checked the IR sensor but the IR sensor is working fine.While pin of the PCB and the code we find out that the pin for data out in code not the same as it in PCB .So we change the pin of IR sonsor from the code

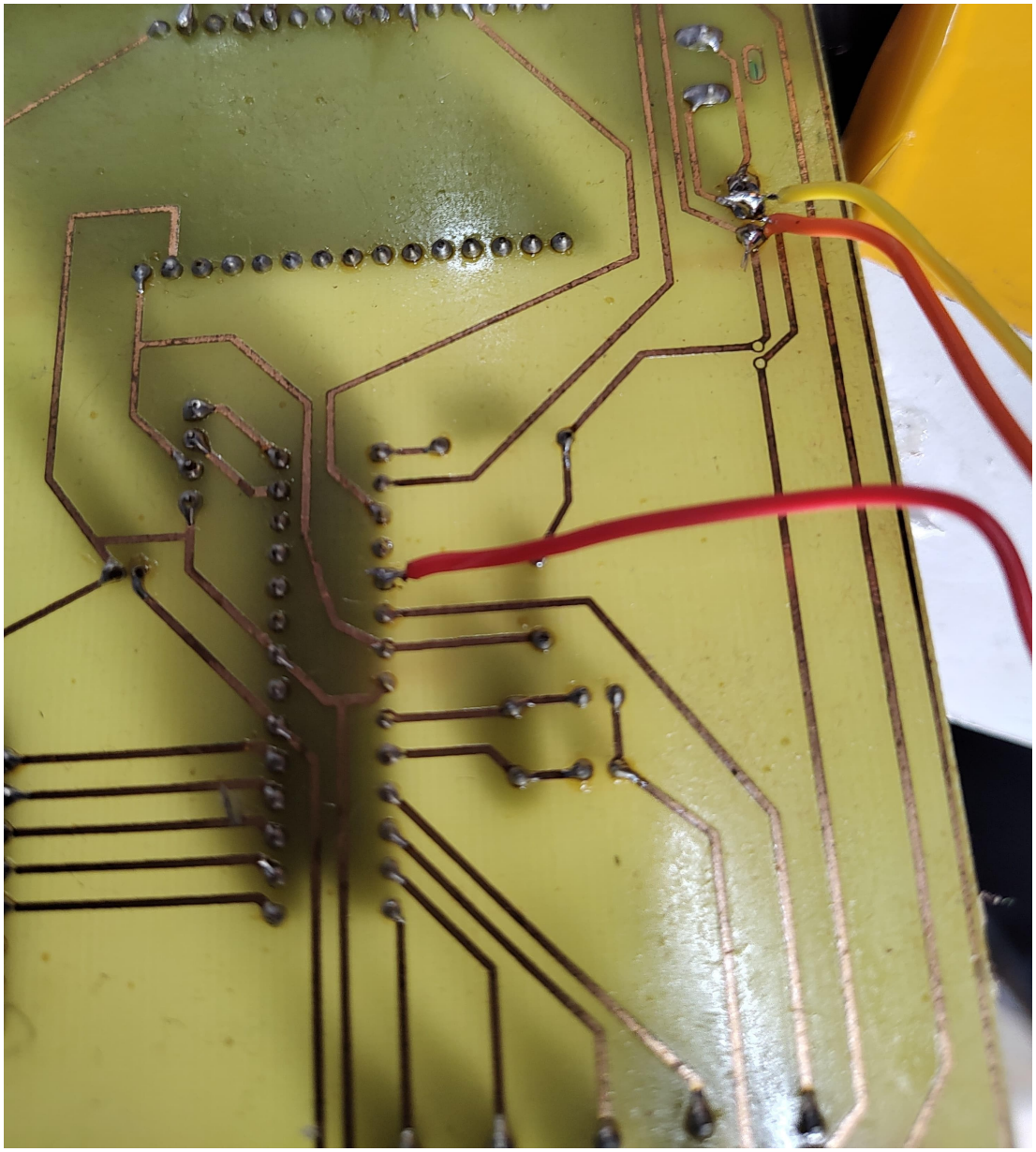


Figure 4.36: Final PCB

At first we impemented the PCB with out the servo motor but we were asked to add the servo by our supervisors.We add servo directly to PCB,slodering the signal pin to the PWM enable pin of the ATmega32 and connected the ground and Vcc of the servo motor to the PCB

4.7 Final Results

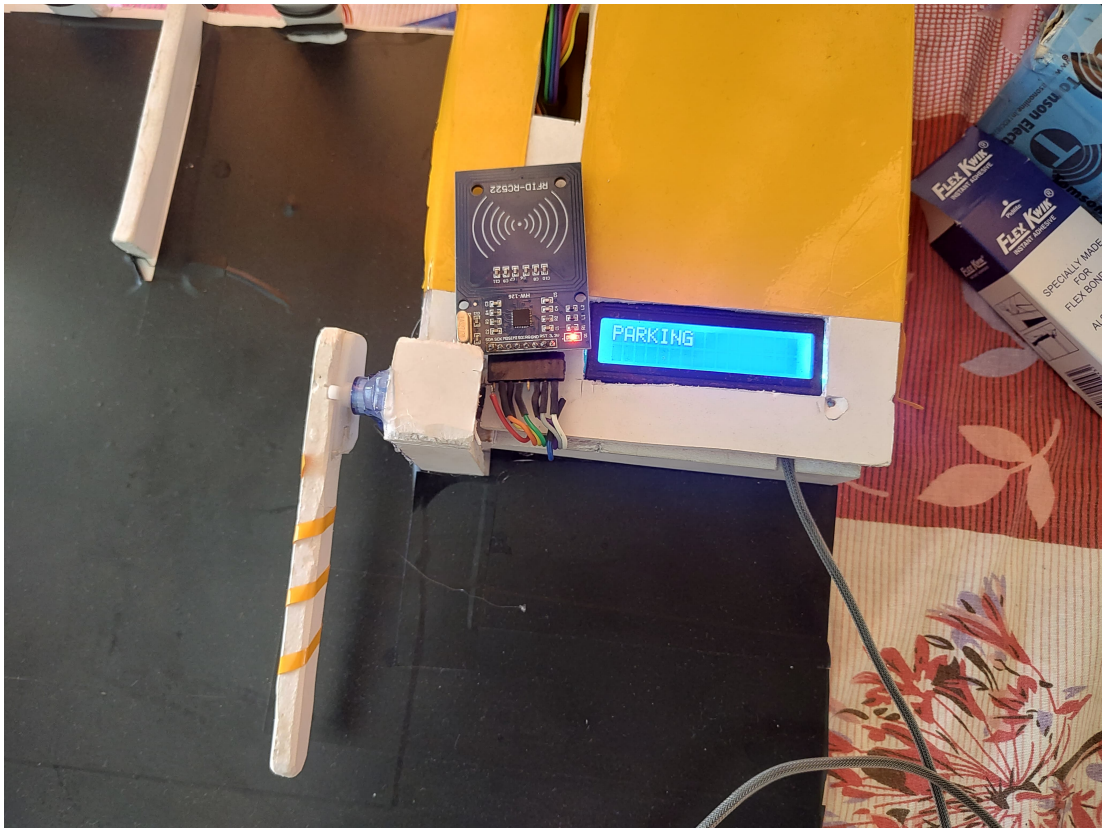


Figure 4.37: Final Result 1

The above figure show the starting of our parking system.The esp32 will receive the booking data,the parking slot and the user Id of the user who had booked the slot.The RFID scanner check for any RFID detection at the entrance.If detects that card the RC522 scanner will sent the data to ATmega32 processor for the user verification.

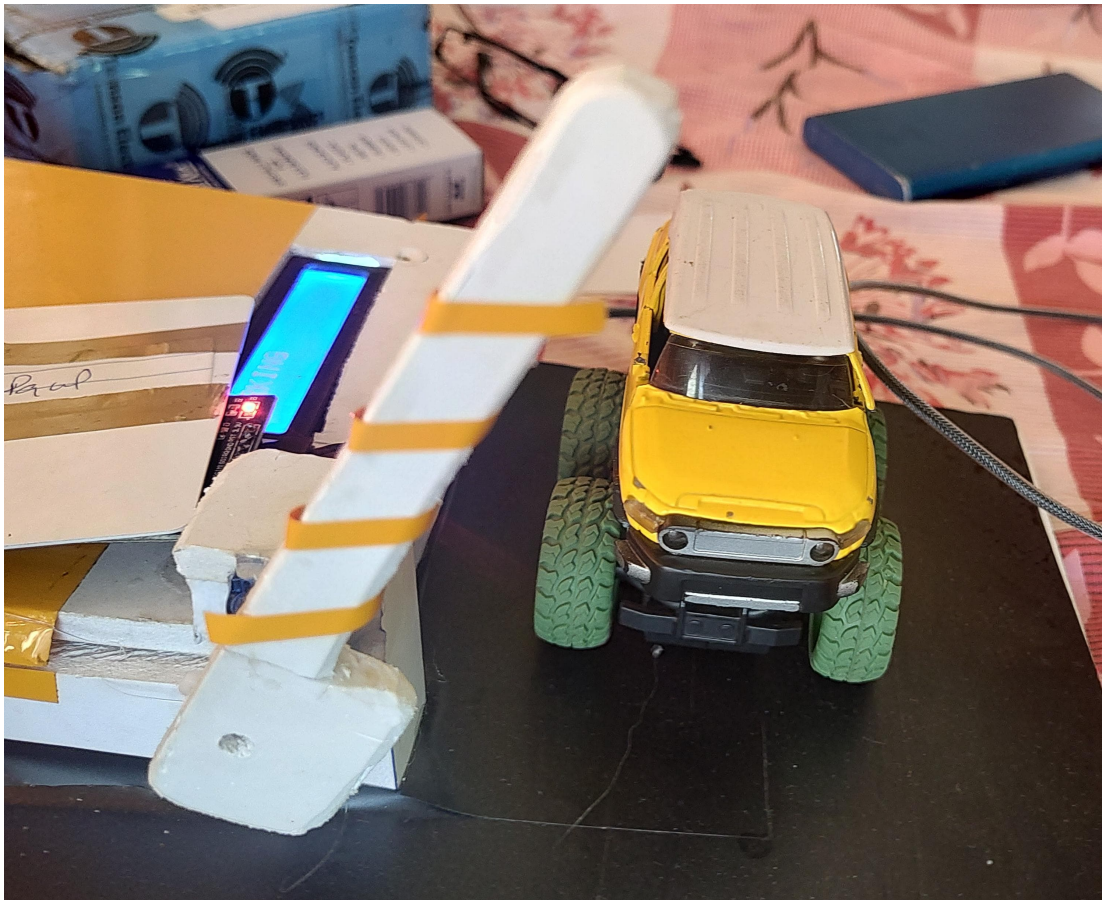


Figure 4.38: Final Result 2

The above figure indicates the gate opening of the parking area.If the RFID is verified by the ATmega from RC522.Then the ATmega32 will rotate the servo to 170 degree.If the user is not verified the servo will be in initial position,hence access denied to the parking area.

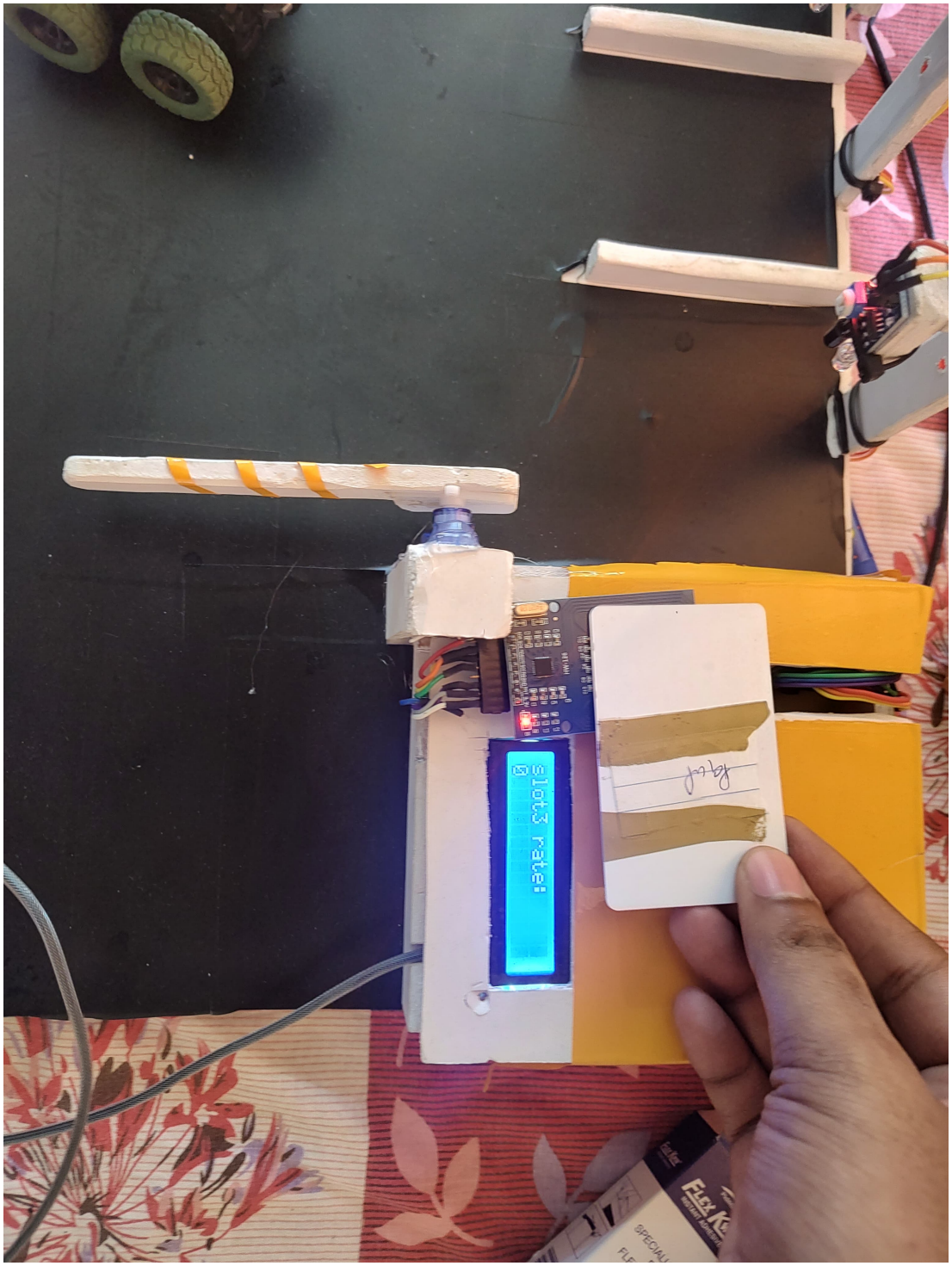


Figure 4.39: Final Result 3

This figure shows the charge of the parking.In this image shows that the user is parked for less than a min.So the parking fee is zero.

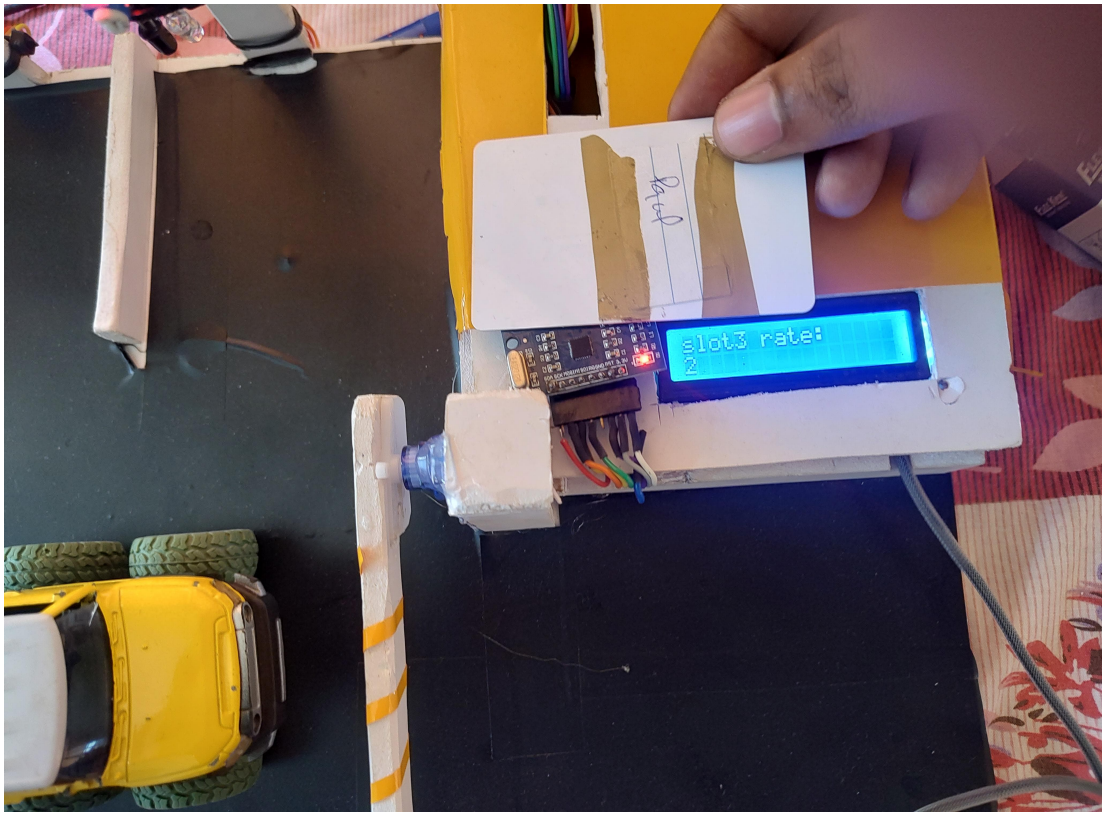


Figure 4.40: Final Result 4

This figure shows the parking fee collection at the exit.When the fee payment is completed the the ATmega will open the gate(rotates the servo by 170 degree).

Also to make the parking slot empty the ATmega will set all timer variable of that slot to and the flag bool variable to false(these bool variable flags is used to check whether we are calculating the time).

Atmega will also send the updates of the parking lot to the esp32 .The esp32 will turn off booking and send the updates received from the ATmega to firebase database

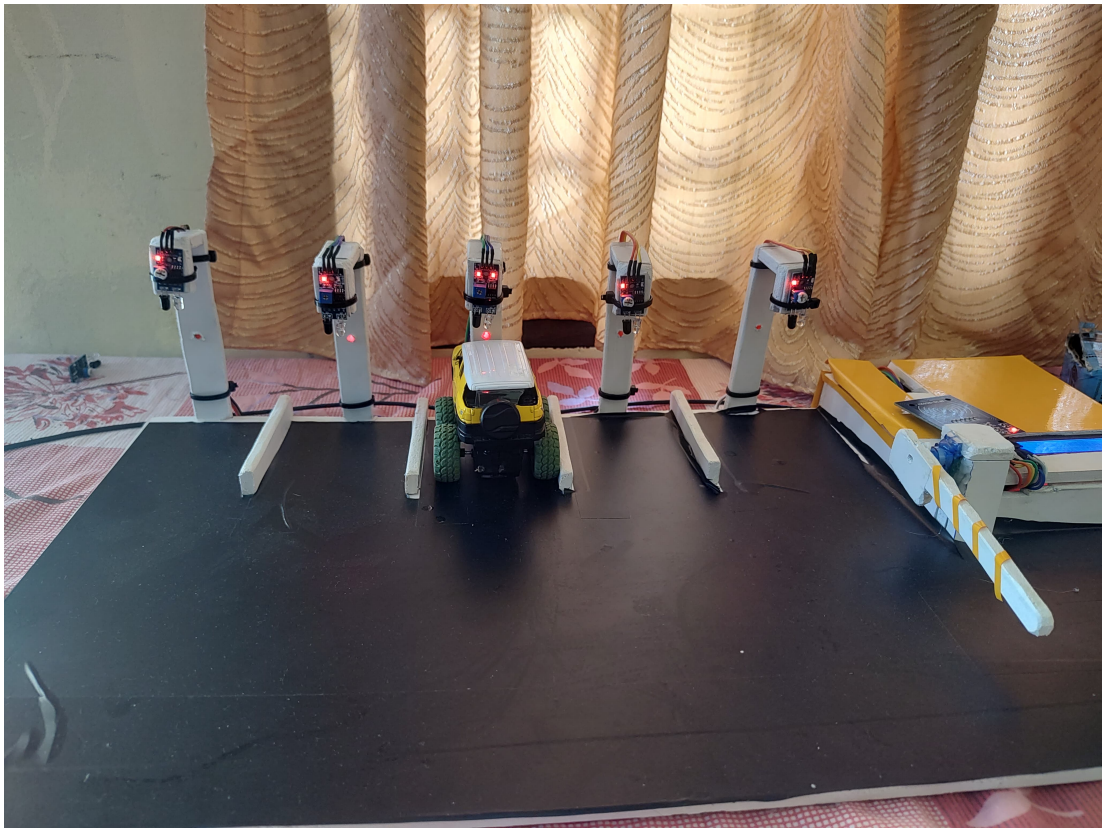


Figure 4.41: Front View

In this figure shows the slot three is occupied by by the vehicle of the user of a user.The parking slot four is not occupies.If this unoccupency is greater than 30 Sec(for the testing purpose).

The esp32 access the firebase database for the updating the booking in our project the esp32 access the database every 1o Sec.The firebase will sent the parking slot and the user Id in Json format to esp32.From figure 6.25 the data will be ”B1=”1”,B2=”1”,B3=”0”,B4=”0”,B5=”0””.The B1,B2 etc are the root of the Json data and the value after the equal sign is the keyvalue of the root.

Also the serial communication between ATmega and the esp32 is in Json format.In ATmega and esp32 we need to parse the received Json to the variable for holding the value of user Id for each parking slot.the parse of keyvalue will be string but the

To send data we convert the psrking slot status variable to string data type in ATmega.The we need to convert these into data form.This Json data will transfere to esp32 and then to firebase to update the status of parking slot at the database which will reflect in moblie app

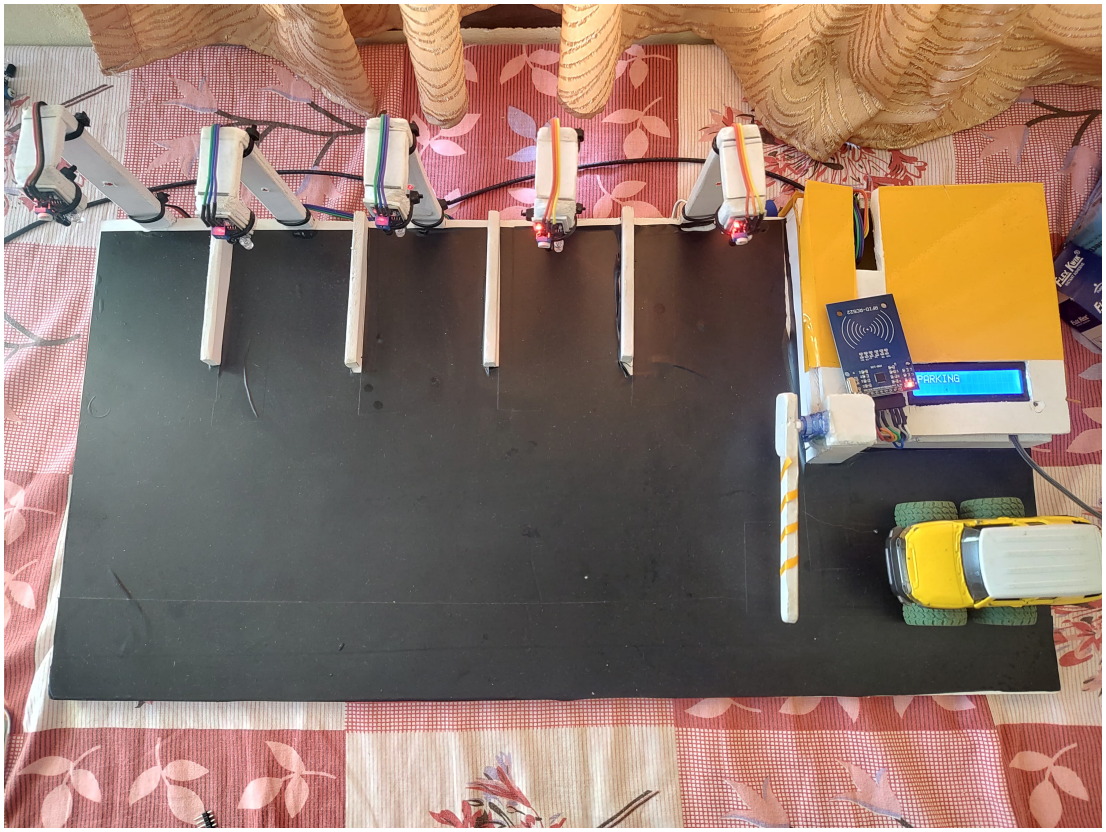


Figure 4.42: Top View

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this project, we've developed a Smart Parking System tailored for retail centers, utilizing advanced sensor technology and data analytics to provide up-to-date parking availability information. This information is accessible through digital displays or a dedicated mobile app, aiming to improve user convenience and alleviate congestion.

The system incorporates predictive algorithms to optimize traffic flow, thereby enhancing productivity within retail environments. To streamline the parking process, automated payment options and an intuitive interface guide visitors to available parking spaces.

In this hardware prototype phase, we've implemented IR sensors to monitor vehicle presence in each parking slot. These sensors transmit data to the cloud via a WiFi module, enabling real-time access to parking availability information through the mobile app. Users can also utilize the app to reserve parking slots, with booking data transmitted to the microcontroller through the cloud.

The microcontroller then interfaces with booking indication LEDs, providing visual cues to other users regarding slot availability. This seamless integration of hardware, cloud connectivity, and user interface ensures real-time updates on parking slot availability and facilitates convenient booking processes as well as cancelation of slots

booked within a given timeperiod

In summary, our prototype Smart Parking System offers real-time parking availability information, booking capabilities, and efficient management of parking slots in busy retail environments. This implementation aims to enhance operational efficiency, improve user experience, and address common parking challenges faced in retail settings.

ADVANTAGES

- Real-time Parking Availability: Instant access to current parking availability status via digital displays or mobile apps.
- Convenient Booking Process: Users can easily reserve parking slots through the mobile app, ensuring availability upon arrival.
- Enhanced Operational Efficiency: Predictive algorithms optimize traffic flow, reducing congestion and improving productivity in retail environments.
- Improved User Experience: Intuitive interface and real-time updates contribute to a positive overall experience for users.
- Efficient Space Utilization: Accurate parking availability information maximizes space usage in crowded retail settings, enhancing efficiency and customer satisfaction.

DISADVANTAGES

- Dependency on Technology: The system relies heavily on sensor technology and cloud connectivity, making it susceptible to malfunctions or disruptions in case of technical issues.
- Initial Setup Costs: Implementing state-of-the-art sensor technology and cloud infrastructure may require significant upfront investment, potentially limiting adoption by smaller retail centers.
- Connectivity Issues: Poor WiFi connectivity or network outages could disrupt

communication between the hardware prototype and the cloud, affecting the accuracy of parking availability information.

- User Accessibility Challenges: Not all users may be comfortable using mobile apps or digital displays to access parking availability information or book parking slots, potentially excluding certain demographics.
- Limited Flexibility: The system's reliance on IR sensors for vehicle detection may limit its effectiveness in environments with unconventional parking arrangements or obstacles obstructing sensor visibility.
- Security Concerns: Transmitting sensitive booking information over the cloud raises privacy and security concerns, necessitating robust data encryption measures to safeguard user data from potential cyber threats.
- Maintenance Requirements: Regular maintenance and calibration of IR sensors and other hardware components are essential to ensure accurate and reliable operation, adding to the system's overall maintenance costs and complexity.

5.1 Future Work

1. Advanced Sensor Technologies: Investing in more advanced sensor technologies, such as lidar and advanced camera systems, can improve the accuracy and reliability of parking space detection.These technologies could provide more comprehensive data on parking space occupancy and support a finer level of granularity in space allocation.

2. Enhanced User Experience: Future developments should focus on further enhancing the user experience through mobile applications. Features like real-time navigation within the parking structure, parking history, and personalized recommendations could make the overall parking experience more convenient and user-friendly.

3. Data Analytics for Business Insights: Utilizing data analytics tools can provide valuable insights into parking patterns, customer behavior, and overall mall traffic.This information can be used by mall management to make informed decisions about facility planning, marketing strategies, and resource allocation.

References

[1] M. P. Thakrel, P. S. Borse, N. P.Matale, P. Sharma "**IoT based smart vehicle parking system using RFID**" 2 International Conference on Computer Communication and Informatics (ICCCI)2021, Jan. 27 – 29, 2021, Coimbatore, INDIA .

[2] L. Kumar, M. H. Khan and M. S. Umar, "**Smart parking system using RFID and GSM technology**", International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)2017, Aligarh, India, 2017, pp. 180-184.

[3] J. D. Bachhav, M. A. Mechkul ."**IoT based smart parking system**" Published in: 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)2021 06-08 May 2021, Madurai, India.

[4] A. O. Kotb, Y. -C. Shen, X. Zhu and Y. Huang, "**iParker—A New Smart Car-Parking System Based on Dynamic Resource Allocation and Pricing**", IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 9, pp. 2637-2647, Sept. 2016

[5] J.F. Zafari, A. Gkelias and K. K. Leung, "**A Survey of Indoor Localization Systems and Technologies**", in IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2568-2599.

[6] X. Wang, L. Gao and S. Mao, "**BiLoc: Bi-Modal Deep Learning for Indoor Localization With Commodity 5GHz WiFi**", in IEEE Access 2017, vol. 5, pp. 4209-4220.

[7] P. P. Gaikwad, J. P. Gabhane and S. S. Golait, "**A survey based on Smart Homes system using Internet-of-Things**", International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)2015, Melmaruvathur, India,2015, pp. 0330-0335

[8] S. Lee, H. Choi, T. Kim, H. -S. Park and J. K. Choi, "**A Novel Energy-Conscious Access Point (eAP) System With Cross-Layer Design in Wi-Fi Networks for Reliable IoT Services**" in IEEE Access 2022, vol. 10, pp. 61228-61248.

[9] A. D. Wankhade and K. Wagh, "**Design Secure Cloud Based IoT Network for End to End Cloud Communication**," 11th International Conference on Emerging Trends in Engineering Technology - Signal and Information Processing (ICETET - SIP)2023, Nagpur, India, 2023, pp. 1-5

[10] Li, Z. Xie, L. Liu and Y. Wu, "**Design and Implementation of an Integrated City-Level IoT Platform Based on Edge Computing and Cloud Native**",IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)2022, Beijing, China, 2022, pp. 463-467

[11] E. Khan, D. Garg, R. Tiwari and S. Upadhyay, "**Automated Toll Tax Collection System using Cloud Database**",3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)2018, Bhimtal, India, 2018, pp. 1-5

[12] Y. Benazzouz, C. Munilla, O. Günalp, M. Gallissot and L. Gürgen, "**Sharing user IoT devices in the cloud**",IEEE World Forum on Internet of Things (WF-IoT)2014, Seoul, Korea (South), 2014, pp. 373-374

[13] A. Kumar, N. C. Narendra and U. Bellur, "Uploading and Replicating Internet of Things (IoT) Data on Distributed Cloud Storage", IEEE 9th International Conference on Cloud Computing (CLOUD)2016, San Francisco, CA, USA, 2016, pp. 670-677

[14] N. Nikolov and O. Nakov, "**Research of Communication Between IoT Cloud Structure, Android Application and IoT Device Using TCP Sockets**", X National Conference with International Participation (ELECTRONICA)2019, Sofia, Bulgaria, 2019, pp. 1-4

[15] M. Ishaq, M. H. Afzal, S. Tahir and K. Ullah, "**A Compact Study of Recent Trends of Challenges and Opportunities in Integrating Internet of Things (IoT) and Cloud Computing**", International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)2021, Quetta, Pakistan, 2021, pp. 1-4

[16] A. Alsaidi and F. Kausar, "**Security Attacks and Countermeasures on Cloud Assisted IoT Applications**", IEEE International Conference on Smart Cloud (SmartCloud)2018, New York, NY, USA, 2018, pp. 213-217

[17] S. Shekhar, "**Dynamic Data Driven Cloud Systems for Cloud-Hosted CPS**", IEEE International Conference on Cloud Engineering Workshop (IC2EW)2016, Berlin, Germany, 2016, pp. 195-197

[18] L. Mhatre and N. Rai, "**Integration between wireless sensor and cloud**" International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)2017, Palladam, India, 2017, pp. 779-782.

[19] D. Priyadarshi and A. Behura, "**Analysis of Different IoT Protocols for Heterogeneous Devices and Cloud Platform**", International Conference on Communication and Signal Processing (ICCSP)2018, Chennai, India, 2018, pp. 0868-0872

[20] V. R. Tadinada, "**Software Defined Networking: Redefining the Future of Internet in IoT and Cloud Era**", International Conference on Future Internet of Things and Cloud 2014, Barcelona, Spain, 2014, pp. 296-301

APPENDIX

SOURCE CODE

Code in Atmega328P

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>

#define IR1 4
#define IR2 5
#define IR3 6
#define IR4 7
#define IR5 8
#define SS_PIN 10
#define RST_PIN 9

MFRC522 rfid(SS_PIN, RST_PIN); // Instance of the class
MFRC522::MIFARE_Key key;

unsigned long startTime1 = 0;
unsigned long stopTime1 = 0;
bool timer1Running = false;

unsigned long startTime2 = 0;
unsigned long stopTime2 = 0;
bool timer2Running = false;

unsigned long startTime3 = 0;
unsigned long stopTime3 = 0;
bool timer3Running = false;

unsigned long startTime4 = 0;
unsigned long stopTime4 = 0;
```

```
bool timer4Running = false;

unsigned long startTime5 = 0;
unsigned long stopTime5 = 0;
bool timer5Running = false;

unsigned long timer1 = 0;
bool t1 = false;
unsigned long timer2 = 0;
bool t2 = false;
unsigned long timer3 = 0;
bool t3 = false;
unsigned long timer4 = 0;
bool t4 = false;
unsigned long timer5 = 0;
bool t5 = false;

// Define target IDs
byte targetID1[] = {0x7C, 0xAB, 0xFA, 0x37};
byte targetID2[] = {0xC3, 0x6F, 0x45, 0xFB};
byte targetID3[] = {0x33, 0x06, 0x45, 0xFB};
byte targetID4[] = {0x33, 0x25, 0xEB, 0x1B};
byte targetID5[] = {0x43, 0xCF, 0x50, 0xFB};
int S1, S2, S3, S4, S5;
int l1=1,l2=1,l3=1,l4=1,l5=1;
int r1=1,r2=1,r3=1,r4=1,r5=1;

Servo servoMotor;

void sendleddata();
void checkid(int i);

LiquidCrystal_I2C lcd(0x27, 16, 2); // Address 0x27, 16 columns, 2 rows

void setup() {
  Serial.begin(9600); // Set baud rate to match sender
  servoMotor.attach(3);
  servoMotor.write(90);
  delay(100);
  pinMode(IR1, INPUT);
  pinMode(IR2, INPUT);
  pinMode(IR3, INPUT);
  pinMode(IR4, INPUT);
  pinMode(IR5, INPUT);
  lcd.init(); // Initialize LCD
  lcd.backlight(); // Turn on backlight
  lcd.clear(); // Clear LCD
  lcd.print("PARKING ");
```

```
  SPI.begin(); // Init SPI bus
  rfid.PCD_Init(); // Init MFRC522
}



void loop()
{
  if (Serial.available() > 0)
  { // Check if data is available to read
    String receivedData = Serial.readStringUntil('\n');

    // Parsing data
    int parsedB1, parsedB2, parsedB3, parsedB4, parsedB5;
    sscanf(receivedData.c_str(), "B1: %d, B2: %d, B3: %d, B4: %d, B5: %d", &
    parsedB1, &parsedB2, &parsedB3, &parsedB4, &parsedB5);

    S1 = parsedB1;
    S2 = parsedB2;
    S3 = parsedB3;
    S4 = parsedB4;
    S5 = parsedB5;
  }
  if(S1!=0)
  {
   if (!t1)
   {
    timer1 = millis();
    t1 = true;
   }
   if (digitalRead(IR1) == LOW && !timer1Running)
   {
    startTime1 = millis();
    timer1Running = true;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Slot 1 occupied");
    delay(500);
    lcd.clear();
    l1=1;
    t1 = false;
    timer1 = 0;
   }
   else if(digitalRead(IR1) == HIGH && timer1Running)
   {
    l1=0;
    r1=0;
    sendleddata();
```

```
  servoMotor.write(90);
  delay(100);
 }
 if (t1 && !timer1Running && (millis() - timer1 >= 30000))
 {
  r1 = 0;
  l1 = 0;
  t1=false;
  timer1 = 0;
  sendleddata();
 }
}
else
{
  t1 = false;
}

if(S2!=0)
{
 if (!t2)
 {
  timer2 = millis();
  t2 = true;
 }
 if (digitalRead(IR2) == LOW && !timer2Running)
 {
  startTime2 = millis();
  timer2Running = true;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Slot 2 occupied");
  delay(500);
  lcd.clear();
  l2=1;
  t2 = false;
  timer2 = 0;
 }
 else if(digitalRead(IR2) == HIGH && timer2Running)
 {
  l2=0;
  r2=0;
  sendleddata();
  servoMotor.write(90);
  delay(100);
 }
 if (t2 && !timer2Running && (millis() - timer2 >= 30000))
 {
  r2 = 0;
```

```
  l2 = 0;
  t2=false;
  timer2 = 0;
  sendleddata();
 }
}
else
{
  t2 = false;
}

if(S3!=0)
{
 if (!t3)
 {
  timer3 = millis();
  t3 = true;
 }
 if (digitalRead(IR3) == LOW && !timer3Running)
 {
  startTime3 = millis();
  timer3Running = true;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Slot 3 occupied");
  delay(500);
  lcd.clear();
  l3=1;
  t3 = false;
  timer3 = 0;
 }
 else if(digitalRead(IR3) == HIGH && timer3Running)
 {
  l3=0;
  r3=0;
  sendleddata();
  servoMotor.write(90);
  delay(100);
 }
 if (t3 && !timer3Running && (millis() - timer3 >= 30000))
 {
  r3 = 0;
  l3 = 0;
  t3=false;
  timer3 = 0;
  sendleddata();
 }
}
```

```
else
{
  t3 = false;
}

if(S4!=0)
{
 if (!t4)
 {
  timer4 = millis();
  t4 = true;
 }
 if (digitalRead(IR4) == LOW && !timer4Running)
 {
  startTime4 = millis();
  timer4Running = true;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Slot 4 occupied");
  delay(500);
  lcd.clear();
  l4=1;
  t4 = false;
  timer4 = 0;
 }
 else if(digitalRead(IR4) == HIGH && timer4Running)
 {
  l4=0;
  r4=0;
  sendleddata();
  servoMotor.write(90);
  delay(100);
 }
 if (t4 && !timer4Running && (millis() - timer4 >= 30000))
 {
  r4 = 0;
  l4 = 0;
  t4=false;
  timer4 = 0;
  sendleddata();
 }
}
else
{
  t4 = false;
}

if(S5!=0)
```

```
  {
   if (!t5)
   {
    timer5 = millis();
    t5 = true;
   }
   if (digitalRead(IR5) == LOW && !timer5Running)
   {
    startTime5 = millis();
    timer5Running = true;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Slot 5 occupied");
    delay(500);
    lcd.clear();
    l5=1;
    t5 = false;
    timer5 = 0;
   }
   else if(digitalRead(IR5) == HIGH && timer5Running)
   {
    l5=0;
    r5=0;
    sendleddata();
    servoMotor.write(90);
    delay(100);
   }
   if (t5 && !timer5Running && (millis() - timer5 >= 30000))
   {
    r5 = 0;
    l5 = 0;
    t5=false;
    timer5 = 0;
    sendleddata();
   }
  }
  else
  {
    t5 = false;
  }
  rfidcheck();
}


void rfidcheck()
{
  if (!rfid.PICC_IsNewCardPresent())
        return;
```

```
    // Verify if the NUID has been read
    if (!rfid.PICC_ReadCardSerial())
        return;

    if (memcmp(rfid.uid.uidByte, targetID1, sizeof(targetID1)) == 0)
    {
//      lcd.setCursor(0, 0);
//      lcd.print("Target ID 1 matched.");
      checkid(1);
    }
    else if (memcmp(rfid.uid.uidByte, targetID2, sizeof(targetID2)) == 0)
    {
//      lcd.setCursor(0, 0);
//      lcd.print("Target ID 2 matched.");
      checkid(2);
    }
    else if (memcmp(rfid.uid.uidByte, targetID3, sizeof(targetID3)) == 0)
    {
//      lcd.setCursor(0, 0);
//      lcd.print("Target ID 3 matched.");
      checkid(3);
    }
    else if (memcmp(rfid.uid.uidByte, targetID4, sizeof(targetID4)) == 0)
    {
//      lcd.setCursor(0, 0);
//      lcd.print("Target ID 4 matched.");
      checkid(4);
    }
    else if (memcmp(rfid.uid.uidByte, targetID5, sizeof(targetID5)) == 0)
    {
//      lcd.setCursor(0, 0);
//      lcd.print("Target ID 5 matched.");
      checkid(5);
    } else
    {
//      lcd.setCursor(0, 0);
//      lcd.print("No matching ID found.");
    }

    // Halt PICC
    rfid.PICC_HaltA();
}



void checkid(int i)
{
```

```
    if(i==S1)
    {
//      lcd.clear();
//      lcd.setCursor(0, 1);
//      lcd.print("id is in S1");
      if (timer1Running)
      {
        stopTime1 = millis();
        timer1Running = false;
        unsigned long elapsedTime1 = (stopTime1 - startTime1) / 60000;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("slot1 rate: ");
        lcd.setCursor(0, 1);
        lcd.print(elapsedTime1);
        delay(1000);
        lcd.clear();
        startTime1 = 0;
        stopTime1 = 0;
        r1=0;
        l1=1;
        sendleddata();
        r1=1;
        servoMotor.write(170);
        delay(100);
      }
    }
    else if(i==S2)
    {
//      lcd.clear();
//      lcd.setCursor(0, 1);
//      lcd.print("id is in S2");
      if (timer2Running)
      {
        stopTime2 = millis();
        timer2Running = false;
        unsigned long elapsedTime2 = (stopTime2 - startTime2) / 60000;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("slot2 rate: ");
        lcd.setCursor(0, 1);
        lcd.print(elapsedTime2);
        delay(1000);
        lcd.clear();
        startTime2 = 0;
        stopTime2 = 0;
        r2=0;
        l2=1;
```

```
        sendleddata();
        r2=1;
        servoMotor.write(170);
        delay(100);
      }
    }
    else if(i==S3)
    {
//      lcd.clear();
//      lcd.setCursor(0, 1);
//      lcd.print("id is in S3");
      if (timer3Running)
      {
        stopTime3 = millis();
        timer3Running = false;
        unsigned long elapsedTime3 = (stopTime3 - startTime3) / 60000;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("slot3 rate: ");
        lcd.setCursor(0, 1);
        lcd.print(elapsedTime3);
        delay(1000);
        lcd.clear();
        startTime3 = 0;
        stopTime3 = 0;
        r3=0;
        l3=1;
        sendleddata();
        r3=1;
        servoMotor.write(170);
        delay(100);
      }
    }
    else if(i==S4)
    {
//      lcd.clear();
//      lcd.setCursor(0, 1);
//      lcd.print("id is in S4");
      if (timer4Running)
      {
        stopTime4 = millis();
        timer4Running = false;
        unsigned long elapsedTime4 = (stopTime4 - startTime4) / 60000;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("slot4 rate: ");
        lcd.setCursor(0, 1);
        lcd.print(elapsedTime4);
```

```
        delay(1000);
        lcd.clear();
        startTime4 = 0;
        stopTime4 = 0;
        r4=0;
        l4=1;
        sendleddata();
        r4=1;
        servoMotor.write(170);
        delay(100);
      }
    }
    else if(i==S5)
    {
//      lcd.clear();
//      lcd.setCursor(0, 1);
//      lcd.print("id is in S5");
      if (timer5Running)
      {
        stopTime5 = millis();
        timer5Running = false;
        unsigned long elapsedTime5 = (stopTime5 - startTime5) / 60000;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("slot5 rate: ");
        lcd.setCursor(0, 1);
        lcd.print(elapsedTime5);
        delay(1000);
        lcd.clear();
        startTime5 = 0;
        stopTime5 = 0;
        r5=0;
        l5=1;
        sendleddata();
        r5=1;
        servoMotor.write(170);
        delay(100);
      }
    }
  }

  void sendleddata()
  {
    Serial.print("l1: ");
    Serial.print(l1);
    Serial.print(", l2: ");
    Serial.print(l2);
    Serial.print(", l3: ");
```

```
    Serial.print(l3);
    Serial.print(", l4: ");
    Serial.print(l4);
    Serial.print(", l5: ");
    Serial.print(l5);
    Serial.print(", r1: ");
    Serial.print(r1);
    Serial.print(", r2: ");
    Serial.print(r2);
    Serial.print(", r3: ");
    Serial.print(r3);
    Serial.print(", r4: ");
    Serial.print(r4);
    Serial.print(", r5: ");
    Serial.println(r5);
  }
```

Code in ESP32

```
{
#include <ArduinoJson.h>
#include <Arduino.h>
#if defined(ESP32)
  #include <WiFi.h>
#elif defined(ESP8266)
  #include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>

//Provide the token generation process info.
#include "addons/TokenHelper.h"
//Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"

// Insert your network credentials
#define WIFI_SSID "realme"
#define WIFI_PASSWORD "12345678"

// Insert Firebase project API Key
#define API_KEY "AIzaSyApv9di90PF2DX_z1I8G5_KAF7iNYdMHdk"

// Insert RTDB URLefine the RTDB URL */
#define DATABASE_URL "https://carparking-aa456-default-rtdb.firebaseio.com/"
//Define Firebase Data object
FirebaseData fbdo;

#define led1 19
#define led2 18
```

```
#define led3 5
#define led4 4
#define led5 2

FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;
int intValue;
float floatValue;
bool signupOK = false;
int l1,l2,l3,l4,l5;
int s1,s2,s3,s4,s5;
int r1=1,r2=1,r3=1,r4=1,r5=1;

void setup() {
  Serial.begin(9600);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
//  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    //Serial.print(".");
//    delay(300);
  }
  //Serial.println();
  //Serial.print("Connected with IP: ");
//  Serial.println(WiFi.localIP());
//  Serial.println();

  /* Assign the api key (required) */
  config.api_key = API_KEY;

  /* Assign the RTDB URL (required) */
  config.database_url = DATABASE_URL;

  /* Sign up */
  if (Firebase.signUp(&config, &auth, "", "")) {
    //Serial.println("ok");
    signupOK = true;
  }
  else {
    //Serial.printf("%s\n", config.signer.signupError.message.c_str());
  }
```

```
  /* Assign the callback function for the long running token generation task
    */
  config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper
    .h

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
}



void loop()
{
  if (Serial.available() > 0)
  { // Check if data is available to read
   String receivedData = Serial.readStringUntil('\n');
   //Serial.println(receivedData);
   int parsedB1, parsedB2, parsedB3, parsedB4, parsedB5, parsedB6, parsedB7,
    parsedB8, parsedB9, parsedB10;
   sscanf(receivedData.c_str(), "l1: %d, l2: %d, l3: %d, l4: %d, l5: %d, r1: %
    d, r2: %d, r3: %d, r4: %d, r5: %d", &parsedB1, &parsedB2, &parsedB3, &
    parsedB4, &parsedB5, &parsedB6, &parsedB7, &parsedB8, &parsedB9, &
    parsedB10);

        l1=parsedB1;
        l2=parsedB2;
        l3=parsedB3;
        l4=parsedB4;
        l5=parsedB5;
        r1=parsedB6;
        r2=parsedB7;
        r3=parsedB8;
        r4=parsedB9;
        r5=parsedB10;

//        Serial.print("l1: ");
//        Serial.print(l1);
//        Serial.print(", l2: ");
//        Serial.print(l2);
//        Serial.print(", l3: ");
//        Serial.print(l3);
//        Serial.print(", l4: ");
//        Serial.print(l4);
//        Serial.print(", l5: ");
//        Serial.println(l5);

//        Serial.print("r1: ");
```

```
//        Serial.print(r1);
//        Serial.print(", r2: ");
//        Serial.print(r2);
//        Serial.print(", r3: ");
//        Serial.print(r3);
//        Serial.print(", r4: ");
//        Serial.print(r4);
//        Serial.print(", r5: ");
//        Serial.println(r5);

        ledoff();
  }
  // Check if Firebase is ready, signed up, and it's time to retrieve data
  if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 15000
    || sendDataPrevMillis == 0))
  {
    sendDataPrevMillis = millis();

    // Attempt to retrieve JSON data for the specified path ("/Time" in this
    case)
    if (Firebase.RTDB.getJSON(&fbdo, "/"))
    {
      // Check if the data was successfully retrieved
      if (fbdo.jsonObjectPtr())
      {
        // Convert JSON object to string
        String jsonData = fbdo.jsonString();

        // Parse JSON data
        StaticJsonDocument<200> doc;
        DeserializationError error = deserializeJson(doc, jsonData);

        // Check for parsing errors
        if (error)
        {
//          Serial.print("Failed to parse JSON: ");
//          Serial.println(error.c_str());
          return;
        }

        // Extract values for B1, B2, B3, B4, and B5
        int b1 = doc["B1"].as<int>();
        int b2 = doc["B2"].as<int>();
        int b3 = doc["B3"].as<int>();
        int b4 = doc["B4"].as<int>();
        int b5 = doc["B5"].as<int>();

        s1=b1;
```

```
        s2=b2;
        s3=b3;
        s4=b4;
        s5=b5;
        ledon();

        // Print the retrieved data
        Serial.print("B1: ");
        Serial.print(b1);
        Serial.print(", B2: ");
        Serial.print(b2);
        Serial.print(", B3: ");
        Serial.print(b3);
        Serial.print(", B4: ");
        Serial.print(b4);
        Serial.print(", B5: ");
        Serial.println(b5);
      } else {
//        Serial.println("Failed to get JSON data");
      }
    } else {
//      Serial.println("Failed to retrieve JSON data from Firebase");
    }
  }
}

void ledoff()
{
  if(l1==0)
  {
    digitalWrite(led1,LOW);
  }
  if(l2==0)
  {
    digitalWrite(led2,LOW);
  }
  if(l3==0)
  {
    digitalWrite(led3,LOW);
  }
  if(l4==0)
  {
    digitalWrite(led4,LOW);
  }
  if(l5==0)
  {
    digitalWrite(led5,LOW);
  }
```

```
  if(r1==0)
  {
    Firebase.RTDB.setInt(&fbdo, "B1", 0);
    r1=1;
  }
  if(r2==0)
  {
    Firebase.RTDB.setInt(&fbdo, "B2", 0);
    r2=1;
  }
  if(r3==0)
  {
    Firebase.RTDB.setInt(&fbdo, "B3", 0);
    r3=1;
  }
  if(r4==0)
  {
    Firebase.RTDB.setInt(&fbdo, "B4", 0);
    r4=1;
  }
  if(r5==0)
  {
    Firebase.RTDB.setInt(&fbdo, "B5", 0);
    r5=1;
  }
}

void ledon()
{
  if(s1!=0)
  {
    digitalWrite(led1,HIGH);
    l1=1;
  }
  if(s2!=0)
  {
    digitalWrite(led2,HIGH);
    l2=1;
  }
  if(s3!=0)
  {
    digitalWrite(led3,HIGH);
    l3=1;
  }
  if(s4!=0)
  {
    digitalWrite(led4,HIGH);
    l4=1;
```

```
  }
  if(s5!=0)
  {
    digitalWrite(led5,HIGH);
    l5=1;
  }
}

}

}
```