ABSTRACT

This project aims to create a sophisticated vehicle access authentication system by combining fingerprint and face recognition technologies while incorporating IoT connectivity for remote monitoring and control. The methodology involves designing the system architecture, curating a diverse dataset for algorithm training, developing robust biometric algorithms, and integrating hardware components, such as the ESP32-CAM and Nodemcu, to ensure seamless communication and IoT connectivity. Rigorous testing and validation are conducted to verify the system's accuracy and reliability.

The result is a multi-layered security approach that enhances vehicle security by allowing only authorized users to access vehicles, thereby reducing the risk of theft and misuse. Users can conveniently manage access permissions and receive real-time alerts through the IoT connectivity. Stringent security measures are implemented to protect data and communication channels, adhering to privacy regulations. This system offers a secure and user-friendly solution that enhances vehicle security while providing a seamless and efficient experience for users.

CHAPTER 1

INTRODUCTION

In today's rapidly evolving world, the integration of technology into our daily lives is more pronounced than ever before. The realm of transportation is no exception, as it has witnessed transformative innovations aimed at improving convenience and security for vehicle owners and users. In this context, the development of a comprehensive Vehicle Access Authentication System that combines biometric recognition, specifically fingerprint and face recognition, with Internet of Things (IoT) connectivity, represents a significant leap forward. This multifaceted system not only addresses the imperative need for enhancing the security of personal vehicles but also promises a seamless and user-friendly experience.

As vehicle technology continues to advance, so do the methods employed by those with malicious intent to gain unauthorized access to automobiles. Car theft and unauthorized usage remain pressing concerns for vehicle owners and manufacturers. Traditional methods of securing vehicles, such as keys and key fobs, while effective to a certain extent, are not foolproof. As a response to these challenges, biometric authentication methods like fingerprint and face recognition have gained prominence due to their ability to provide a higher level of security.

Additionally, the rise of the Internet of Things (IoT) has opened new frontiers in vehicle access control. The ability to remotely monitor and manage vehicle access through IoT connectivity provides a range of benefits, from tracking the location of the vehicle to authorizing temporary access for others, thereby increasing the control and convenience for vehicle owners.

This document presents a comprehensive exploration of the design and development of a robust Vehicle Access Authentication System that effectively addresses these challenges and leverages the power of biometric recognition and IoT technology. Through a detailed analysis of the system's architecture, data collection and preparation, algorithm development, hardware integration, testing, and security measures, this document sheds light on how this innovative system can revolutionize vehicle security while enhancing user experience. It is a testament to the potential of technology in providing solutions that balance the ever-present need for security with the expectations of user convenience in our modern world.

With the growing significance of technology in modern life, the development of a comprehensive Vehicle Access Authentication System is poised to redefine the way we secure our vehicles and manage access. This system represents a synergy of cutting-edge biometric recognition technologies, such as fingerprint and face recognition, and the power of Internet of Things (IoT) connectivity, all aimed at striking a harmonious balance between heightened security and user convenience.

At the core of this system lies a meticulously designed architecture. It integrates components like the ESP32-CAM for face recognition, Nodemcu with a fingerprint sensor, an IoT module, and a web application. These components work in tandem to create a seamless, secure ecosystem for vehicle access.

The foundation for accurate biometric recognition lies in a diverse dataset. Collecting and preparing this data is pivotal for training the fingerprint and face recognition algorithms. Diversity ensures that the system can reliably identify authorized users across various conditions.

The efficacy of the system hinges on the precision of the biometric recognition algorithms. These algorithms are meticulously crafted using appropriate libraries to ensure their accuracy in distinguishing authorized users from unauthorized individuals.

Configuring microcontrollers, establishing communication protocols, and enabling IoT connections are integral parts of hardware integration. The success of the system depends on the efficient interplay between these hardware components, ensuring a seamless user experience. The harmonious interaction between the biometric recognition algorithms, IoT module, and web application is pivotal. This ensures that the system functions seamlessly, offering users a reliable and convenient means of accessing their vehicles. Once the system's components are in place, rigorous testing procedures are undertaken to validate its accuracy, robustness, and efficiency. This ensures the system functions as intended under various conditions. Moreover, security measures, including data encryption and compliance with privacy regulations, are implemented to safeguard user information. User experience is paramount in the design of the Vehicle Access Authentication System. The system simplifies user enrollment and management, allowing vehicle owners to easily add or remove users and authorize temporary access through the web application. Real-time alerts further enhance user convenience, notifying owners of unauthorized access attempts and vehicle movements. The system eliminates the need for physical keys, enhancing the overall convenience and efficiency of vehicle access.

EMBEDDED SYSTEMS:

Introduction to Embedded Systems:

Embedded systems play a pivotal role in the success of our project aimed at non-invasively monitoring glucose, hemoglobin, and sodium levels. These specialized computing systems are at the heart of our monitoring device, ensuring its functionality, efficiency, and real-time data processing capabilities. In this section, we provide an introduction to embedded systems and discuss their critical role in our project.

Embedded systems are computer systems designed to perform specific tasks or functions within a larger system, often with a focus on real-time operation and resource efficiency. Unlike general-purpose computers, which are versatile and capable of running a wide range of applications, embedded systems are tailored to perform dedicated functions with a high degree of reliability and predictability.

In our project, the embedded system serves as the central processing unit of the monitoring device, responsible for interfacing with the Infrared (IR), Photoplethysmography (PPG), and pH sensors, collecting data from them, processing this data, and providing meaningful outputs to the user. Here's how embedded systems contribute to the success of our project:

Real-Time Data Processing: Embedded systems excel at processing data in real-time, ensuring that measurements of glucose, hemoglobin, and sodium levels are continuously and accurately analyzed. This capability is crucial for providing timely feedback to users and healthcare professionals.

Sensor Integration: Embedded systems serve as the interface between the various sensors used in the project and ensure that data from these sensors are collected, synchronized, and processed seamlessly. This integration is essential for the overall functionality of the monitoring device.

Efficiency: Embedded systems are designed for efficiency, both in terms of power consumption and computational resources. This is critical for ensuring that the monitoring device can operate for extended periods on battery power and provide a user-friendly experience.

Reliability and Stability: Embedded systems are known for their stability and reliability, ensuring that the monitoring device operates consistently without crashes or disruptions. This reliability is crucial for the accuracy of health parameter measurements.

User Interface: The embedded system also manages the user interface of the monitoring device, allowing users to interact with the device, view their health data, and receive alerts or recommendations.

In summary, embedded systems form the backbone of our non-invasive health monitoring project, enabling the seamless integration of sensors, real-time data processing, and user interaction. The success of our project hinges on the effective design and implementation of the embedded system, ensuring that it delivers on its promise of making health monitoring accessible, non-invasive, and user-friendly.

CHAPTER 2

LITERATURE REVIEW

Title: Biometric car security and monitoring system using IOT

Authors:

Publication: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)

"Vehicle security is a crucial issue recently because of the rising range of thefts.Also another issue with vehicles is handling its keys. Keys have to be compelled to be carried and misplacing keys or loosing them can cause a heavy issue. Here we tend to propose an answer to the current drawback is by employing a fingerprint vehicle starter system. The system provides a secure and trouble free thanks to start/stop vehicle engine.User simply has to scan finger,no need to carry any key.The system permits licensed users to begin the vehicle.Users will initial register into the system by scanning fingerprints. The system permits multiple users to register.Here we use an Atmega 328 microcontroller and esp8266 wifimodule.The fingerprint detector is connected to the microcontroller and LCD display together with push buttons and starting motor.The whole system uses IOT technology to update the vehicle information on the webapp.",

Title: Intelligent Car Security System Based on Fingerprint Identification and Internet of Things

Authors: Shuai LV,Lele Wu,Zeyu Li,Qi Xu,Xinxin Zhou,

Publication: 2021 13th International Conference on Computational Intelligence and Communication Networks (CICN)

"With the improvement of people's economic level, many families have already owned compact cars. As a convenient tool for daily travel, cars are important private property, thus the safety of car is a problem that can't be ignored. And the design in this article adopts the working mode that combines single chip microcomputer with App. Employing 52 single chip microcomputer as detection center and control core of the antitheft system, new automobile has two kinds of unlocking method: password and fingerprint. Between them, fingerprint unlocking is safe and quick, which realizes Internet of things through communication module, enabling users to receive accurate and real-time security information of the car on APP.",

Title: Implementation of Biometric Access Control Using Fingerprint for Safety and Security System of Electric Vehicle

Authors: Marey Jemima Raj,Sajitha Gadde,Ramesh Jayaraman,

Publication: 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)

"This paper presents the implementation of biometric access control using a fingerprint for the safety and security system of Electric Vehicles (EV). The EVs play a vital role in the minimization of the emission of carbon footprint into the atmosphere. Right now, researchers are giving more attention to the design and development of advanced EVs with more safety and security systems. As the population is gradually increasing, there is a positive correlation between the number of road traffic accidents deaths. These road accidents are unpredictable. The main reasons for road traffic accidents are negligence in driving, drunk and drive also a minor cause is being on the stand and driving. At the same time, the vehicle owner has afraid more because day-by-day vehicle robbery is keep on increasing, which intends to develop anti-theft devices for EVs. In the next few years, with tremendous growth in the electric vehicle market, it is important to design EVs with high security and safety systems. In this paper, an advanced safety and security system is proposed for EVs which ensures the stand removing and vehicle owner through fingerprint recognition. The proposed safety and security systems are implemented and tested in E-Bike the efficacy of the system was found satisfactory.",

Title: A Biometric Vehicle Theft Identification and Prevention Scheme using GPS Location Tracking

Authors: K. Jeevitha,J. Venkatesh,V. Indhumathi,K. Krishna Veni,R. Prem Kumar,Devadarshini. M,

Publication: 2023 International Conference on Advances in Computing Communication and Applied Informatics (ACCAI)

"The demand for security is rising in every sector of society as the pace of technological development and scientific discovery quickens. Right now, having access to a car is essential. At the same time, preventing theft from happening is crucial. The expense and complexity of conventional methods of protecting vehicles are substantial. There is no further action or option that might help the car's owner recover their vehicle once it has been stolen. The usage of biometrics such as fingerprints is widespread and is now routine in many settings, including businesses, public buildings, educational institutions, and more. The primary objective of this study is to secure the car against unwanted entry using fingerprint recognition technology that is both quick and simple to implement, as well as clear, reliable, and cost-effective. A method was necessary to track where each vehicle was at all times and how far it had gone. These days, active vehicle monitoring and GPS technologies are used to keep tabs on our motor vehicles. Images of fingerprints are taken by the sensor, and the sensor then compares each fingerprint it reads to a database of recorded fingerprints or to a module inside the system. A GPS and GSM-based anti-theft car monitoring system would be the most cost-effective means of tracing a vehicle's whereabouts and could even be used to locate stolen vehicles. It is a built-in unit that utilizes GPS and the GSM network to pinpoint the exact location and trajectory of a moving vehicle (GSM)",

Title: Implementation of Vehicle Security System using GPSGSM and Biometric

Authors: Mridhula Ramesh,S Akruthi,K Nandhini,S Meena,S Joseph Gladwin,R Rajavel,

Publication: 2019 Women Institute of Technology Conference on Electrical and Computer Engineering (WITCON ECE)

"As the variety of urban vehicles is growing apace with the event of the economy, individuals are becoming a lot concerned about vehicle thievery, which creates broader market prospects for vehicle anti-theft products. Various vehicle antitheft devices have been developed latterly, but the outcome is still unsatisfactory since every kind of device has its drawbacks. Therefore, an enhanced system has been proposed in this paper to ensure vehicle safety and track the vehicle in the event of theft. This proposed system includes a fingerprint-based authentication to enable the engine ON in addition to the key mechanism. The owner has to use both fingerprint and key, to access the vehicle. Even if one input out of the two is available, the vehicle cannot be turned on. When an unauthorized person tries to operate the vehicle by an alternate mechanism by bypassing the finger print authentication and key, an alert SMS with the location coordinates is sent to the owner which will enable him/her to track the vehicle with the help of GPS and GSM technologies.",

Title: Privacy-Preserving Biometric-based Authentication Scheme for Electric Vehicles Charging System

Authors: Ahmed Sherif,Mohamed Elsersy,Mahmoud Nabil,Mohamed Mahmoud,Khaled H. Almotairi,

Publication: 2021 3rd IEEE Middle East and North Africa COMMunications Conference (MENACOMM)

"Nowadays, with the continuous increase in oil prices and the worldwide shift towards clean energy, all-electric vehicles are booming. Thence, these vehicles need widespread charging systems operating securely and reliably. Consequently, these charging systems need the most robust cybersecurity measures and strong authentication mechanisms to protect its user. This paper presents a new security scheme leveraging human biometrics in terms of iris recognition to defend against multiple types of cyber-attacks such as fraudulent identities, man-in-the-middle attacks, or unauthorized access to electric vehicle charging stations. Fundamentally, the proposed scheme implements a security mechanism based on the inherently unique characteristics of human eye biometric. The objective of the proposed scheme is to enhance the security of electric vehicle charging stations by using a low-cost and efficient authentication using k-Nearest Neighbours (KNN), which is a lightweight encryption algorithm.We tested our system on high-quality images obtained from the standard IITD iris database to search over the encrypted database and authenticate a legitimate user. The results showed that our proposed technique had minimal communication and computation overhead, which is quite suitable for the resource-limited charging station devices. Furthermore, we proved that our scheme outperforms other existing techniques.",

Title: Prototype of a fingerprint based licensing system for driving

Authors:

Publication: 2013 International Conference on Information Communication and Embedded Systems (ICICES)

"To prevent non-licensees from driving and therefore causing accidents, a new system is proposed. An important and very reliable human identification method is fingerprint identification. Fingerprint identification is one of the most popular and reliable personal biometric identification methods. The proposed system consists of a smart card capable of storing the fingerprint of particular person. While issuing the license, the specific person's fingerprint is to be stored in the card. Vehicles such as cars, bikes etc should have a card reader capable of reading the particular license. The same automobile should have the facility of fingerprint reader device. A person, who wishes to drive the vehicle, should insert the card (license) in the vehicle and then swipe his/her finger. If the finger print stored in the card and fingerprint swiped in the device matches, he/she can proceed for ignition, otherwise ignition will not work. Moreover, the seat belt detector verifies and then prompts the user to wear the seat belt before driving. This increases the security of vehicles and also ensures safe driving by preventing accidents.",

Title: Fingerprint recognition-based access controlling system for automobiles

Authors: Zhaoxia Zhu,Fulong Chen,

Publication: 2011 4th International Congress on Image and Signal Processing

"Traditional identifying system of automobiles includes key system, door-controlling system, Ultra High Frequency (UHF) sending/receiving system, alarm system and so on. Once the key is lost or stolen, those persons that have not the ownership can also open the door of the car with it. Biometric recognition technologies can overcome this shortcoming. As an Automatic Fingerprint Identification System (AFIS), fingerprint recognition-based access control system of automobiles, in which fingerprint encryption technique is utilized, has some advantages such as smartness, security, low power, low cost, etc.",

Title: Anti-theft protection of vehicle by GSM & GPS with fingerprint verification

Authors: Mrinmoy Dey,Md. Akteruzzaman Arif,Md. Asif Mahmud,

Publication: 2017 International Conference on Electrical Computer and Communication Engineering (ECCE)

"Recently vehicle tracking system is getting vast popularity because of the rising number of the stolen vehicles. Vehicle theft is happening on parking and sometimes driving in unsecured places. This research work explores how to avoid this kind of stealing and provides more security to the vehicles. The implemented system contains single-board embedded system which is equipped with global system for mobile (GSM) and global positioning system (GPS) along with a microcontroller installed in the vehicle. The use of GSM and GPS technologies allows the system to track the object and provides the most up-to-date information about on-going trips. Moreover, fingerprint verification is done in the implemented system to ensure the driving of correct person. The implemented system is very simple with greater security for vehicle anti-theft protection and low cost technique compared to others.",

Title: Authenticated Access Control for Vehicle Ignition System by Driveru2019s License and Fingerprint Technology

Authors: Arwa M. Ali,Heisum M. Awad,Ibrahim K. Abdalgader,

Publication: 2020 International Conference on Computer Control Electrical and Electronics Engineering (ICCCEEE)

"The vehicles theft increase day by day rapidly and the ratio of unlicensed drivers also increase, which creates a major responsibility towards manufactures as well as owners of luxury automobile to inbuilt the antitheft system which prevent the cars from stole. In this thesis it designed security system for vehicles. This system includes arduino, driver's license card (DL), Radio Frequency Identification reader (RFID), Fingerprint module (FP) and Global System for Mobile Communication modem (GSM). The arduino used to serve as entire brain of the system. It can add authorized driver's license and fingerprints of trusted people into program burn in arduino. When any person enters driver's license in RFID reader then will be compare to data existing in program if DL is authorized then move to place the finger in FP scanner to verify if match occur that will be ignition the system and this person can drive the car and if not Short Message Services (SMS) via GSM modem send to owner of vehicle and block the ignition system. Also SMS send to driver's license holder to renewal DL before expiration.",

Title: Analysis of IoT-based Vehicle Anti-Theft Security

Authors: Kgaotsang Thamoethata,Bassey Isong,Nosipho Dladlu,Adnan M. Abu-Mahfouz,

Publication: 2021 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)

"A vehicle anti-theft system is a security system that prevents authorized use of a vehicle and its theft. Vehicle theft and hijacking are skyrocketing on daily basis despite the swift technological advancements the world is witnessing. Several vehicle anti-theft systems have been proposed, developed, and deployed using several technologies such as Internet of Things based, biometric-based or hybrid. Therefore, this paper performed the analysis of some of the existing systems to identify the solutions offered, technologies utilized, limitations and provide research directions for improvements. We considered 11 papers and the findings obtained revealed the existence of several vehicle antitheft approaches employing common technologies and approaches to protect vehicles from theft and hijacking. However, the techniques employed are prone to manipulation and destruction thereby not preventing or reducing vehicle theft since the systems are embedded in the vehicle. Consequently, there is the need to design or develop a vehicle anti-theft and monitoring system that is not a component of the vehicle. This could go a long way to reduce the rate of vehicle theft in society.",

Title: Fingerprint Authentication-based Traffic Offence Control and Enforcement System on Smart Mobile Devices for Smart City

Authors: Kirtika A P S Parameswaran,Mazlina Abdul Majid,Md. Shohidul Islam,

Publication: 2022 International Conference on Intelligent Technology System and Service for Internet of Everything (ITSS-IoE)

"The evolution of communication and information technology in today's time should not despise the various practical aspects of daily life, regardless of economic, education, health, or other government services. Various functions of smart mobile devices meet the needs of users in many ways, where different mechanisms like templates, locks, fingerprints, and passwords are used to protect those functions. Due to the increased number of vehicles in smart cities, it becomes difficult for traffic officers with less manpower to complete many of their tasks related to registration, license, and issuance of summons in time, even reviewing the traffic violation's history. In addition, existing traffic systems are not real-time, data related to traffic management can be lost at any time, leading to the wastage of money and resources. To overcome these difficulties in smart cities, this paper proposes a fingerprint authentication-based traffic offence control-and-enforcement system on smart mobile devices. This scheme introduces a security framework to facilitate many tasks related to identification, registration, licensing, and issuance of summons to traffic violators by implementing fingerprint authentication. Functionality tests and user acceptance tests related to traffic offence problems have been conducted on the proposed system by analyzing biometric data of vehicle users' fingerprints.",

Title: IoT Based Embedded System for Vehicle Security and Driver Surveillance

Authors: Mahesh R. Pawar,Imdad Rizvi,

Publication: 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)

"Toady due to rapid increase in vehicles, there is an exponential increase in crime and accidents hence it has become challenge for governments to limit such crimes especially from professional thieves. This paper proposes designing and development of anti-theft as well as driver surveillance embedded system that uses biometric authentication to access the vehicle. This system contains camera which take the image of a person trying to get access of vehicle and compare with authorized person's image and then allowing or denying access. In The case of denial of vehicle access or even if there is an accident occurs, camera will capture the images and email it to the owner or authorizer. This will help to catch thieves, also allows the surveillance of driver and also the inner part of vehicle. The recent work on proposed embedded system is written in this paper. The system is designed and developed using raspberry pi, high resolution camera, vibration sensor and open source software.",

Title: Electric Bike Security: Biometric & GPS Integration for Intrusion Detection

Authors: Nikhil Gala,Anshu Poswalia,Riya Gharat,

Publication: 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)

"The proliferation of the e-bike industry has highlighted the need for a robust and secure tracking system. The traditional mechanical key-based vehicle security systems have limitations that include vulnerability to theft, limited access control, lack of personalization, and real-time tracking. This research paper outlines the creation and deployment of a sophisticated GPS/GSM-based vehicle tracking system that incorporates biometric authentication for access control, addressing the aforementioned challenges. The system leverages the Global Positioning System (GPS) to unceasingly monitor the real-time location of vehicles and facilitates the integration of Global Navigation Satellite System (GNSS) technology, in the form of a NEO-6M module, with a Global System for Mobile Communications (GSM) network, expedited by a SIM900A module integrated with TCP/IP stack. The core of the system is the ATMEGA16 Microcontroller, which serves as the central processing unit, managing the incoming data from the GPS (latitude, longitude and time) and GSM (phone number, signal strength and mobile network provider) modules and governing the systemu2019s overall functionality. The biometric module is powered by an ATMega328p chip incorporating fingerprint recognition technology, which adds an additional layer of security by permitting only authorized access. The proposed solution provides a cost-effective and secure means of monitoring that can be integrated into existing fleet management systems and used in various applications. Presently, the system is being utilized as a part of an electric bicycle kit and has been tested in controlled environments, demonstrating its reliability and efficiency.",

Title: Raspberry Pi based biometric authentication vehicle door locking system

Authors: N.N. Nagamma,M.V. Lakshmaiah,T. Narmada,

Publication: 2017 IEEE International Conference on Power Control Signals and Instrumentation Engineering (ICPCSI)

"In biometric authentication based vehicle door locking system, the biometric authentication plays an important role to provide high security. Now a days security is very important in order to keep our data confidentially from unauthorized persons. The main aim of this paper is to protect the car from unauthorized people by using the unique id that is finger print authentication. At the place of car door locking system, the finger print scanner is placed to lock and unlock the doors in place of the conventional door locking system. Which gives more protection to the car owner. The other indication system is also developed by using GSM module to send the message to the car owner mobile. The entire system is controlled by the Raspberry pi 3 processor.",

CHAPTER 3

SYSTEM DESIGN

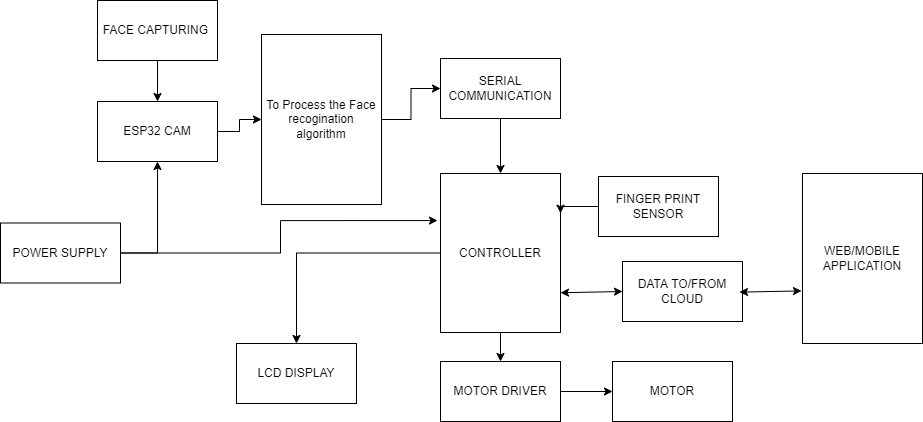
EXSISITNG SYSTEM:

Traditionally, vehicle access and security have primarily relied on physical methods such as keys and key fobs. These methods, while effective to some extent, have their limitations. The existing system lacks robust security, as keys can be lost, stolen, or duplicated. Moreover, remote monitoring and control of vehicle access are virtually non-existent, leaving vehicle owners with limited options for tracking and managing their vehicles. In a world where technology plays an ever-increasing role, the existing system falls short in offering the level of security and convenience that modern vehicle owners expect and need. It is essential to recognize these limitations as we move towards a more advanced and comprehensive solution.

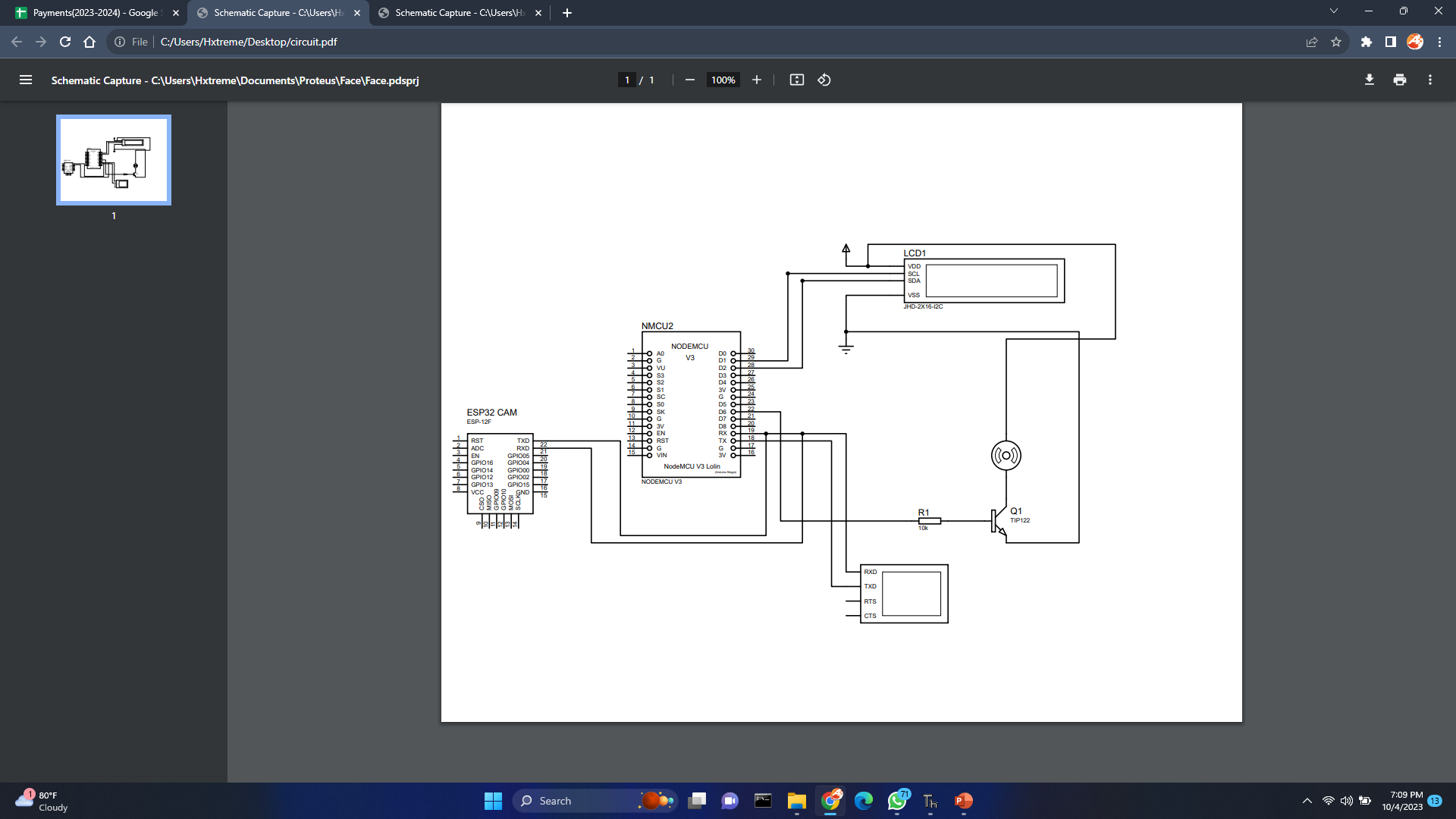
PROPOSED SYSTEM:

The proposed Vehicle Access Authentication System represents a significant leap forward in vehicle security and access management. It combines the power of biometric recognition with IoT connectivity to offer a multifaceted approach to these challenges. The system addresses the limitations of the existing system by providing highly secure biometric recognition methods, specifically fingerprint and face recognition, which are difficult to compromise. Additionally, the integration of IoT technology allows for remote monitoring and control, giving vehicle owners the ability to manage access permissions, track their vehicle's location, and receive real-time alerts in case of unauthorized access attempts. The proposed system not only enhances vehicle security but also significantly improves user convenience. It is a comprehensive solution that aligns with the demands of our technology-driven world, offering a new level of security and user-friendly experience for vehicle owners.

BLOCK DIAGRAM:



CIRCUIT DIAGRAM:



METHODOLOGY:

The methodology for implementing the proposed Vehicle Access Authentication System is a multi-faceted process that encompasses several key stages, each designed to ensure the system's reliability, effectiveness, and adherence to privacy and security standards.

The methodology begins with the design of the system's architecture, which integrates various components, including the ESP32-CAM for face recognition, Nodemcu with a fingerprint sensor, an IoT module, and a user-friendly web application. This architecture forms the backbone of the system, enabling the seamless interaction and communication between its different elements.

Data collection and preparation represent a critical step in the process, involving the curation of a diverse dataset to train the fingerprint and face recognition algorithms. The quality and diversity of this dataset are essential to ensure that the biometric recognition algorithms can reliably identify authorized users in various real-world scenarios.

Algorithm development follows, wherein the fingerprint and face recognition algorithms are meticulously crafted using suitable libraries. The goal is to achieve a high level of accuracy in distinguishing authorized users from unauthorized individuals, reducing the risk of false positives and false negatives during the authentication process.

Hardware integration involves configuring microcontrollers, establishing communication protocols, and enabling IoT connectivity. This step ensures that the system's hardware components work cohesively, enabling smooth and secure interactions between the biometric recognition methods, IoT module, and web application.

Rigorous testing and validation procedures are then carried out to verify the accuracy, robustness, and efficiency of the system. This stage involves subjecting the system to various test scenarios and conditions to identify and rectify potential issues, ensuring that it functions as intended in the real world.

Finally, the implementation of stringent security measures is crucial to safeguard data and communication channels, ensuring that user information remains confidential and in compliance with privacy regulations. Measures such as data encryption, secure communication protocols, and privacy controls are put in place to protect sensitive user data.

This methodology is designed to guide the development and implementation of the Vehicle Access Authentication System, emphasizing a comprehensive and rigorous approach to create a highly secure, reliable, and user-friendly solution that addresses the challenges of modern vehicle security and access management.

CHAPTER 4

HARDWARE DESCRIPTION:

HARDWARE DESCRIPTION:

4.1 NODEMCU (ESP8266)

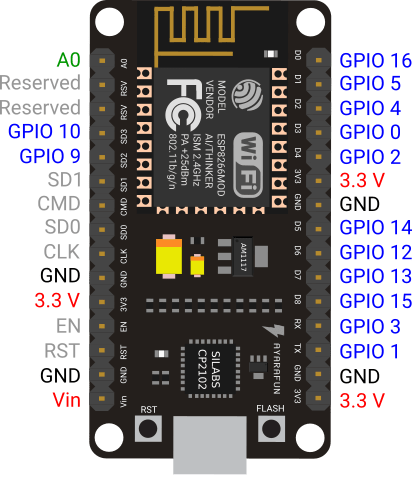


The Atmel AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in a single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers. The ATmega328/P provides the following features: 32Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 1Kbytes EEPROM, 2Kbytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), three flexible Timer/Counters with compare modes and PWM, 1 serial programmable USARTs , 1 byte-oriented 2-wire Serial Interface (I2C), a 6- channel 10- bit ADC (8 channels in TQFP and QFN/MLF packages) , a programmable Watchdog Timer with internal Oscillator, an SPI serial port, and six software selectable power saving modes.

This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main oscillator and the asynchronous timer continue to run. Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use Q Touch Suite toolchain allows you to explore, develop and debug your own touch applications. The device is manufactured using Atmel’s high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core.

Pin diagram of ATMEGA328

The ATmega328/P is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, and Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits. Below figure 3.2.1 pin diagram of ATMEGA328.



FEATURES OF ATMEGA328

28-pin AVR Microcontroller

Flash Program Memory: 32 kilo bytes

EEPROM Data Memory: 1 kilo bytes

SRAM Data Memory: 2 kilo bytes

I/O Pins: 23

Timers: Two 8-bit / One 16-bit

A/D Converter: 10-bit Six Channel

PWM: Six Channels

RTC: Yes with Separate Oscillator

MSSP: SPI and I²C Master and Slave Support

USART: Yes

External Oscillator: up to 20MHz

ADVANTAGES/ IMPROVEMENTS IN ATMEGA328

Still runs on 5 V, so legacy 5 V stuff interfaces cleaner

Even though it's 5 V capable, newer parts can run to 1.8 V. This wide range is very rare.

Nice instruction set, very good instruction throughput compared to other processors (HCS08, PIC12/16/18).

High quality GCC port (no proprietary crappy compilers!)

"PA" variants have good sleep mode capabilities, in micro-amperes.

Well rounded peripheral set

Q Touch capability

Pin Descriptions table

VCC

Digital supply voltage.

GND

Ground.

Port B (PB [7:0]) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up

resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB [7:6] is used as TOSC [2:1] input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

Port C (PC [5:0])

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC [5:0] output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port ‘C’ are elaborated in the Alternate Functions of Port C

section.

Port D (PD [7:0])

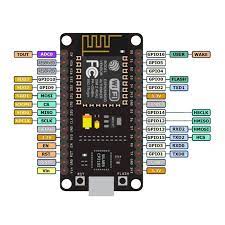
Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

AVCC

AVCC is the supply voltage pin for the A/D Converter, PC [3:0], and PE [3:2]. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter. Note that PC [6:4] use digital supply voltage, VCC.

NODEMCU Uno Board Description

We will learn about the different components on the NODEMCU board. We will study the NODEMCU UNO board because it is the most popular board in the NODEMCU board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below figure 3.2.2, but most NODEMCUs have majority of these components in common.



NODEMCU board can be powered by using the USB cable from computer. All we need to do is connect the USB cable to the USB connection (1).

Power (Barrel Jack)

NODEMCU boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack.

Voltage Regulator

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

Crystal Oscillator

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

NODEMCU Reset

We can reset wer NODEMCU board, i.e., start wer program from the beginning. We can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, we can connect an external reset button to the NODEMCU pin labelled RESET (5).

Pins (3.3, 5, GND, Vin)

3.3V (6) − Supply 3.3 output volt

5V (7) − Supply 5 output volt

Most of the components used with NODEMCU board works fine with 3.3 volt and 5 volt.

GND (8) (Ground) − There are several GND pins on the NODEMCU, any of which can be used to ground wer circuit.

Vin (9) − This pin also can be used to power the NODEMCU board from an external power source, like AC mains power supply.

Analog pins

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

POWER SUPPLY:

TRANSFORMER:

This document presents the solution for a 12V 1A flyback converter based on the Infineon OPTIREG™ TLE8386-2EL controller and IPD50N08S4-13 OptiMOS™-T2. The user is guided through the component selections, the circuit design and, finally, an overview of the experimental results are presented. The TLE8386-2EL is part of the Automotive OPTIREG™ family and it implements a low-side-sense current mode controller with built in protection features. The device is AECQ-100 qualified. The IPD50N08S4-13 is an AEC-Q101 qualified 80V N-channel enhanced mode MOSFET, it is part of the OptiMOS™-T2 family. Intended audience This document is intended for power supply design engineers, application engineers, students, etc., who need to design a Flyback converter for automotive power applications where a galvanic isolation between two voltage domains is required. In particular the focus is on a battery connected flyback that delivers up to 12W at 12V output voltage; the intention is to provide the user with all of the needed information to fully design and characterize the SMPS bringing it from an engineering concept to its production. Specific features and applications are: - 48V to 12V Automotive applications - Isolated current mode SMPS - Flyback regulators with auxiliary sensing

Centre Tapped Transformer Specifications

Step-down Centre tapped Transformer

Input Voltage: 220V AC at 50Hz

Output Voltage: 24V, 12V or 0V

Output Current: 1A

Vertical mount type

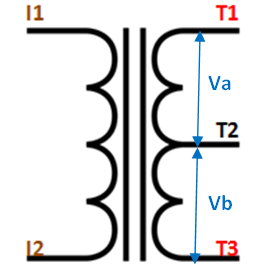
Low cost and small package

A centre-tapped transformer also known as two phase three wire transformer is normally used for rectifier circuits. When a digital project has to work with AC mains a Transformer is used to step-down the voltage (in our case, to 24V or 12V) and then convert it to DC by using a rectifier circuit. In a center-tapped transformer the peak inverse voltage is twice as in bridge rectifier hence this transformer is commonly used in full wave rectifier circuits.

The operation and theory behind a Center tapped transformer is very similar to a normal secondary transformer. A primary voltage will be induced in the primary coil (I1 and I3) and due to magnetic induction the voltage will be transferred to the secondary coil. Here in the secondary coil of a centre tapped transformer, there will be an additional wire (T2) which will be placed exactly at the center of the secondary coil, hence the voltage here will always be zero.

If we combine this zero potential wire (T2) with either T1 or T2, we will get a voltage of 12V AC. If this wire is ignored and voltage across T1 and T2 is considered then we will get a voltage of 24V AC. This feature is very useful for the function of a full wave rectifier.

Let us consider the voltage given by the first half of the secondary coil as Va and the voltage across the second half of the secondary coil as Vb as shown



RECTIFER CIRCUIT:

We have learnt in rectifier circuits about converting a sinusoidal ac voltage into its corresponding pulsating dc. Apart from the dc component, this pulsating dc voltage will have unwanted ac components like the components of its supply frequency along with its harmonics (together called ripples). These ripples will be the highest for a single-phase half wave rectifier and will reduce further for a single-phase full wave rectifier. The ripples will be minimum for 3-phase rectifier circuits. Such supply is not useful for driving complex electronic circuits. For most supply purposes constant dc voltage is required than the pulsating output of the rectifier. For most applications the supply from a rectifier will make the operation of the circuit poor. If the rectifier output is smoothened and steady and then passed on as the supply voltage, then the overall operation of the circuit becomes better. Thus, the output of the rectifier has to be passed though a filter circuit to filter the ac components. The filter is a device that allows passing the dc component of the load and blocks the ac component of the rectifier output. Thus the output of the filter circuit will be a steady dc voltage. The filter circuit can be constructed by the combination of components like capacitors, resistors, and inductors. Inductor is used for its property that it allows only dc components to pass and blocks ac signals. Capacitor is used so as to block the dc and allows ac to pass. All the combinations and their working are explained in detail below. Series Inductor Filter The circuit diagram of a full wave rectifier with a series inductor filter is given below. As the name of the filter circuit suggests, the Inductor L is connected in series between the rectifier circuit and the load. The inductor carries the property of opposing the change in current that flows through it. In other words, the inductor offers high impedance to the ripples and no impedance to the desired dc components. Thus the ripple components will be eliminated. When the rectifier output current increases above a certain value, energy is stored in it in the form of a magnetic field and this energy is given up when the output current falls below the average value. Thus all the sudden changes in current that occurs in the circuit will be smoothened by placing the inductor in series between the rectifier and the load. The waveform below shows the use of inductor in the circuit. From the circuit, for zero frequency dc voltage, the choke resistance Ri in series with the load resistance RL forms a voltage divider circuit, and thus the dc voltage across the load is Vdc = RL/(Ri + RL) Vdc is the output from a full wave rectifier. In this case, the value of Ri is negligibly small when compared to RL. The effect of higher harmonic voltages can be easily neglected as better filtering for the higher harmonic components take place. This is because of the fact that with the increase in frequency, the reactance of the inductor also increases. It should be noted that a decrease in the value of load resistance or an increase in the value of load current will decrease the amount of ripples in the circuit. So, the series inductor filter is mostly used in cases of high load current or small load resistance. A simple series inductor filter may not be properly used. It is always better to use a shunt capacitor (C) with series inductor (L) to form an LC Filter. Shunt Capacitor Filter As the name suggests, a capacitor is used as the filter and this high value capacitor is shunted or placed across the load impedance. This capacitor, when placed across a rectifier gets charged and stores the charged energy during the conduction period. When the rectifier is not conducting, this energy charged by the capacitor is delivered back to the load. Through this energy storage and delivery process, the time duration during which the current flows through the load resistor gets increased and the ripples are decreased by a great amount. Thus for the ripple component with a frequency of ‘f’ megahertz, the capacitor ‘C’ will offer a very low impedance. The value of this impedance can be written as: Shunt Capacitor Impedance = 1/2 fC Thus the dc components of the input signal along with the few residual ripple components, is only allowed to go through the load resistance RLoad. The high amount of ripple components of current gets bypassed through the capacitor C. Now let us look at the working of Half-wave rectifier and Full-wave rectifier with Capacitor filters, their output filtered waveform, ripple factor, merits and demerits in detail.

LIQUID CRYSTAL DISPLAY (LCD)

A Liquid Crystal Display (LCD) is an electronically-modulated optical device shaped into a thin, flat panel made up of any number of colour or monochrome pixels filled with liquid crystals and arrayed in front of a light source (backlight) or reflector. It is often utilized in battery-powered electronic devices because it uses very small amounts of electric power. LCD has material, which continues the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered from similar to a crystal. They are used in similar applications where LEDs are used. These applications are display of display of numeric and alphanumeric characters in dot matrix and segmental displays.

LCD consists of two glass panels, with the liquid crystal materials sandwiched in between them. The inner surface of the glass plates is coated with transparent electrodes which define in between the electrodes and the crystal, which makes the liquid crystal molecules to maintain a defined orientation angle. When a potential is applied across the cell, charge carriers flowing through the liquid will disrupt the molecular alignment and produce turbulence.

When the liquid is not activated, it is transparent. When the liquid is activated the molecular turbulence causes light to be scattered in all directions and the cell appears to be bright. Thus the required message is displayed. When the LCD is in the off state, the two polarizer’s and the liquid crystal rotate the light rays, such that they come out of the LCD without any orientation, and hence the LCD appears transparent. The fig. 6.1 shows the LCD display.

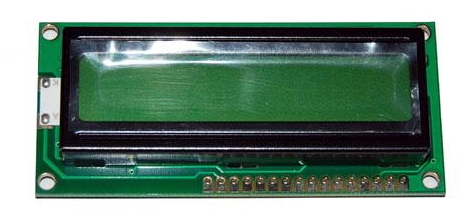


Fig. 6.1 LCD Display

6.2 WORKING OF LCD DISPLAY

When sufficient voltage is applied to the electrodes the liquid crystal molecules would be aligned in a specific direction. The light rays passing through the LCD would be rotated by the polarizer, which would result in activating/highlighting the desired characters. The power supply should be of +5V, with maximum allowable transients of 10mV. To achieve a better/suitable contrast for the display the voltage (V) at pin 3 should be adjusted properly. A module should not be removed from a live circuit.

The ground terminal of the power supply must be isolated properly so that voltage is induced in it. The module should be isolated properly so that stray voltages are not induced, which could cause a flicking display. LCD is lightweight with only a few, millimetres thickness since the LCD consumes less power, they are compatible with low power electronic circuits, and can be powered for long durations. LCD does not generate light and so light is needed to read the display. By using backlighting, reading is possible in the dark. LCDs have long life and a wide operating temperature range. Before LCD is used for displaying proper initialization should be done. LCD is used to display the blood group and blood glucose level.

6.2.1 LCD Pin description

The function of each pins of LCD is described below VCC, VSS and VEE while VDD and VSS provide +5V and ground, respectively, VEE is used for controlling LCD contrast.

6.2.2 Register select

There are two important registers inside the LCD. The RS pin is used for selection as follows. If RS=0, the instruction code register is selected, allowing the user to send a command such as clear display, cursor at home, etc. If RS=1 the data register is selected, allowing the user to send data to be displayed on the LCD.

6.2.3 Read/Write

R/W input allows the user to write information to the LCD or read information from it. R/W=1 when reading; R/W=0 when writing.

6.2.4 Enable

The enable pin is used by the LCD to latch information presented on its data pins. When data is supplied to data pins, a high to low pulse must be applied to this pin in order for the LCD to latch in the data present at the data pins.

6.2.5 D0 - D7

The 8-bit data pins, D0 – D7, are used to send information to the LCD or read contents of the LCD’S internal registers. There are also instruction codes that can be sent to the LCD to clear the display or force the cursor to the home position or blink the cursor. RS=0 is used to check the busy flag bit to see if the LCD is ready to receive information. The busy flag is D7 and can be read when R/W=1 and RS=0, as follows: if R/W=1, RS=0.when D7=1, the LCD is busy taking care of internal operation and will not accept any new information, when D7=0, the LCD is ready to receive new information.

6.3 PIN DESCRIPTION FOR LCD

The table 6.1 illustrate the pin descriptions of LCD.

Table: 6.1 Pin Descriptions of LCD

|  |  |  |
| --- | --- | --- |
| Pin No. | Symbol | Function |
| 1 | Vss | Ground terminal of Module |
| 2 | Vdd | Supply terminal of Module, +5v |
| 3 | Vo | Power supply for liquid crystal drive |
| 4 | RS | Register select  RS=0…Instruction register  RS=1…Data register |
| 5 | R/W | Read/Write  R/W=1…Read  R/W=0…Write |
| 6 | EN | Enable |
| 7-14 | DB0-DB7 | Bi-directional Data Bus. Data Transfer is performed once, through DB0-DB7,incase of interface data length is 8-bits;and twice, thru DB4-DB7 in the case of interface data length is 4-bits.Upper four bits first then lower four bits. |
| 15 | LAMP-(L-) | LED or EL lamp power supply terminals |
| 16 | LAMP+(L+) (E2) | Enable |

Features of 5-Pin 5V Relay

Trigger Voltage (Voltage across coil) : 5V DC

Trigger Current (Nominal current) : 70mA

Maximum AC load current: 10A @ 250/125V AC

Maximum DC load current: 10A @ 30/28V DC

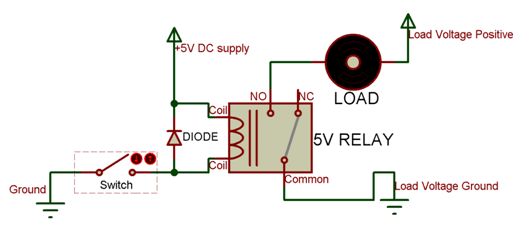
Compact 5-pin configuration with plastic moulding

Operating time: 10msec Release time: 5msec

Maximum switching: 300 operating/minute (mechanically)

Relays are most commonly used switching device in electronics. Let us learn how to use one in our circuits based on the requirement of our project.

Before we proceed with the circuit to drive the relay we have to consider two important parameter of the relay. Once is the Trigger Voltage, this is the voltage required to turn on the relay that is to change the contact from Common->NC to Common->NO. Our relay here has 5V trigger voltage, but you can also find relays of values 3V, 6V and even 12V so select one based on the available voltage in your project. The other parameter is your Load Voltage & Current, this is the amount of voltage or current that the NC,NO or Common terminal of the relay could withstand, in our case for DC it is maximum of 30V and 10A. Make sure the load you are using falls into this range.



The above circuit shows a bare-minimum concept for a relay to operate. Since the relay has 5V trigger voltage we have used a +5V DC supply to one end of the coil and the other end to ground through a switch. This switch can be anything from a small [transistor](https://components101.com/transistors) to a [microcontroller](https://components101.com/microcontrollers) or a microprocessor which can perform switching operating. You can also notice a diode connected across the coil of the relay, this diode is called the Fly back Diode. The purpose of the diode is to protect the switch from high voltage spike that can produced by the relay coil. As shown one end of the load can be connected to the Common pin and the other end is either connected to NO or NC. If connected to NO the load remains disconnected before trigger and if connected to NC the load remains connected before trigger.

Relay is one kind of [electro-mechanical component](https://www.elprocus.com/electromechanical-relay-working-with-applications/) that functions as a switch. The relay coil is energized by DC so that contact switches can be opened or closed. A single channel 5V relay module generally includes a coil, and two contacts like normally open (NO) and normally closed (NC). This article discusses an overview of the 5V relay module & its working but before going to discuss what is [relay](https://www.elprocus.com/relay-circuit-with-working/) module is, first we have to know what is relay and its pin configuration.

in1 (End 1): It is used to activate the relay; usually this pin one end is connected to 5Volts whereas another end is connected to the ground.

Pin2 (End 2): This pin is used to activate the Relay.

Pin3 (Common (COM)): This pin is connected to the main terminal of the Load to make it active.

Pin4 (Normally Closed (NC)): This second terminal of the load is connected to either NC/ NO pins. If this pin is connected to the load then it will be ON before the switch.

Pin5 (Normally Open (NO)): If the second terminal of the load is allied to the NO pin, then the load will be turned off before the switch.

Normal Voltage is 5V DC

Normal Current is 70mA

AC load current Max is 10A at 250VAC or 125V AC

DC load current Max is 10A at 30V DC or 28V DC

It includes 5-pins & designed with plastic material

Operating time is 10msec

Release time is 5msec

Maximum switching is 300 operating per minute

5V Relay Module

The relay module with a single channel board is used to manage high voltage, current loads like [solenoid](https://www.elprocus.com/different-types-of-solenoid-working-applications/) valves, motor, AC load & lamps. This module is mainly designed to interface through different microcontrollers like PIC, Arduino, etc.

5V Relay Module Pin Configuration

The pin configuration of the 5V relay module is shown below. This module includes 6-pins where each pin and its functionality are discussed below.

Applications of Relay

Commonly used in switching circuits.

For Home Automation projects to switch AC loads

To Control (On/Off) Heavy loads at a pre-determined time/condition

Used in safety circuits to disconnect the load from supply in event of failure

Used in Automobiles electronics for controlling indicators glass motors etc.

FINGER PRINT SENSOR:

The R307s fingerprint sensor is a state-of-the-art biometric recognition device designed to provide reliable and secure fingerprint authentication. It offers advanced features and functionality, making it a versatile and effective solution for a wide range of applications, from access control systems to time and attendance tracking.

At the heart of the R307s sensor is its capacitive fingerprint scanning technology, which is known for its high precision and accuracy in capturing fingerprint patterns. This technology relies on the electrical capacitance changes that occur when a finger is placed on the sensor's surface. It can detect even the most intricate details of a fingerprint, including ridge patterns and minutiae points, making it extremely difficult for unauthorized users to replicate or forge fingerprint data.

The sensor boasts an impressive resolution, typically operating at 500 DPI (dots per inch). This high resolution ensures that it can capture and process fingerprint data with exceptional clarity, resulting in reliable and accurate authentication results. Users can be confident that the R307s sensor will consistently recognize authorized individuals while minimizing false positives and false negatives.

In addition to its outstanding fingerprint scanning capabilities, the R307s sensor is designed with versatility in mind. It supports various modes of operation, including fingerprint registration and matching, and it can store a substantial number of fingerprint templates in its internal memory. This flexibility makes it suitable for both single-user and multi-user scenarios, where it can efficiently manage a database of authorized users.

The R307s fingerprint sensor also offers a fast recognition speed. Its efficient matching algorithm ensures quick verification and authentication, making it well-suited for time-sensitive applications such as secure access control or attendance tracking in busy environments.

Furthermore, the sensor is equipped with features that enhance its durability and reliability. It often includes protective measures against electrostatic discharge (ESD) and power supply fluctuations, safeguarding the sensor from potential damage. This durability ensures the sensor can maintain consistent performance over an extended lifespan, even in challenging environments.

The sensor's design typically includes a compact and user-friendly form factor, allowing for easy integration into a variety of devices and systems. It connects through a standard interface, such as UART (Universal Asynchronous Receiver-Transmitter) or USB, simplifying the integration process for developers and engineers.

The R307s fingerprint sensor is known for its ease of use. It often comes with a comprehensive software development kit (SDK) that facilitates the integration of fingerprint recognition into applications and systems. This SDK typically includes libraries and APIs (Application Programming Interfaces) that allow developers to communicate with the sensor and incorporate fingerprint recognition features into their software or hardware solutions.

Overall, the R307s fingerprint sensor represents a robust and reliable biometric recognition solution with its high-resolution scanning capabilities, fast recognition speed, versatility, and user-friendly design. It is a popular choice for applications requiring secure and convenient user authentication, and it continues to play a vital role in enhancing security and access control across various industries.

Top of Form



ESP32 CAM:



The ESP32-CAM is a versatile and powerful microcontroller module that combines an ESP32 microcontroller with a camera, making it an excellent choice for projects that require Wi-Fi connectivity and image or video capture capabilities. This compact module is highly popular in the field of embedded systems, IoT, and DIY electronics due to its extensive features and ease of use.

The ESP32-CAM is powered by the Espressif Systems' ESP32 microcontroller, which offers dual-core processing, Wi-Fi and Bluetooth connectivity, ample memory, and a wide range of I/O pins for flexibility in project development. This combination of features allows it to serve as both a controller and a communication hub for various applications.

One of the standout features of the ESP32-CAM is its integrated OV2640 camera module, which provides the ability to capture still images and videos. With a 2-megapixel resolution and support for various image formats, it can deliver high-quality visuals for projects ranging from surveillance cameras to image recognition applications.

Furthermore, the ESP32-CAM supports microSD card storage, enabling the saving and retrieval of captured images and videos. This is essential for applications that require data logging and archiving, as it ensures data integrity and facilitates easy access to recorded content.

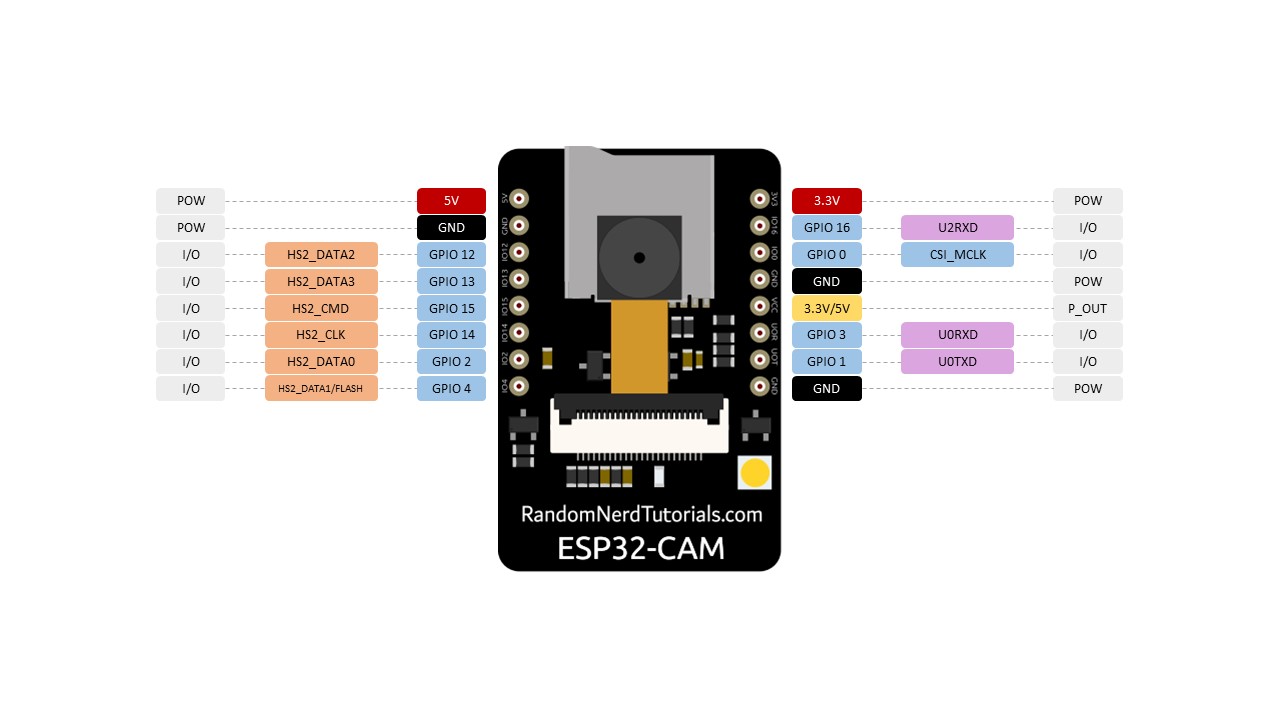
The module can be programmed using the popular Arduino IDE or the Espressif IDF (IoT Development Framework), which gives developers the flexibility to work with familiar tools and libraries. Its compatibility with the Arduino ecosystem simplifies the development process, making it accessible to a broad range of enthusiasts and professionals.

Wi-Fi and Bluetooth connectivity are integral to the ESP32-CAM, allowing it to connect to the internet or other devices. This capability is particularly valuable for IoT applications, as it enables remote control and data transfer, making the module suitable for home automation, smart security systems, and more.

In terms of power supply, the ESP32-CAM typically operates on a 5V power source, but it is essential to pay attention to power management to avoid issues with the camera's current requirements and voltage levels. Additionally, its low-power modes are beneficial for battery-powered applications, conserving energy and extending operational durations.

The ESP32-CAM's compact form factor makes it a versatile choice for various projects. It features a standard OV2640 camera lens interface, facilitating upgrades or modifications to suit specific project requirements. Its pinout and mounting holes further enhance its adaptability, allowing for integration into custom enclosures or PCB designs.

In summary, the ESP32-CAM is a highly capable and adaptable microcontroller module, offering a comprehensive feature set suitable for a wide range of applications. Its integration of Wi-Fi, Bluetooth, and camera capabilities, combined with ease of programming and a strong developer community, make it a top choice for IoT, image capture, and communication projects. Its versatility, compact design, and compatibility with widely used development tools position it as a powerful solution for both hobbyists and professionals in the embedded systems and IoT domains.



CHAPTER 5

SOFTWARE DESCRIPTION

Arduino Development Environment

The Arduino development environment contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

Writing Sketches

Software written using Arduino are called sketches. These sketches are written in the text editor. Sketches are saved with the file extension .ino. It has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino environment including complete error messages and other information. The bottom righthand corner of the window displays the current board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the IDE prior to 1.0 saved sketches with the extension pde It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

The Arduino environment uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

'''Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the Tools Board and Tools Serial Portmenus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241(for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or/dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyUSB0,/dev/ttyUSB1 or similar.

Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the File menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino environment will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch Import Library menu. This will insert one or more  statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete it from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

PROGRAMMING

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

• On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.

• On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of theATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

PYTHON:

PYTHON 3.7:

Python is an interpreter, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object- oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in manya reason most platforms and

may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications. This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off- line as well. For a description of standard objects and modules, see library-index. Reference-index gives a more formal definition of the language. To write extensions in C or C++, read extending-index and c-api-index. There are also several books covering Python in depth. This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python’s most notes worthy features, and will give you a good idea of the language’s flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in library-index. If you do much work on computers, eventually you find that there’s some task you’d like

to automate. For example, you may wish to perform a search-and-replace over a large number of text files, or rename and rearrange a bunch of photo files in a complicated way. Perhaps you’d like to write a small custom database, or a specialized

GUI application or a simple game. If you’re a professional software developer, you may have to work with several C/C++/Java libraries but find the usual write/compile/test/re-compile cycle is too slow. Perhaps you’re writing a test suite for such a library and find writing the testing code a tedious task. Or maybe you’ve written a program that could use an extension language, and you don’t want to design and implement a whole new language for your application.

Typing an end-of-file character (Control-D on Unix, Control-Z on Windows) at the primary prompt causes the interpreter to exit with a zero exit status. If that doesn’t work, you can exit the interpreter by typing the following command: quit(). The interpreter’s line-editing features include interactive editing, history substitution and code completion on systems that support read line. Perhaps the quickest check to see whether command line editing is supported is typing Control-P to the first Python prompt you get. If it beeps, you have command line editing; see Appendix Interactive Input Editing and History Substitution for an introduction to the keys. Ifnothing appears to happen, or if ^P is echoed, command line editing isn’t available; you’ll only be able to use backspace to remove characters from the current line. The interpreter operates somewhat like the Unix shell: when called with standard input connected to a tty device, it reads and executes commands interactively; when called with a file name argument or with a file as standard input, it reads and executes a script from that file. A second way of starting the interpreter is python -c command [arg] ..., which executes the statement(s) in command, analogous to the shell’s -c option. Since Python statements often contain spaces or other characters that are special to the shell, it is usually advised to quote commands in its entirety with single quotes.Some Python modules are also useful as scripts. These can be invoked using python-m module [arg]...,which executes the source file for the module as if you had spelled out its full name on the command line. When a script file is used, it is sometimes useful to be able to run the script and enter interactive mode afterwards. This can be done by passing -i before the script.

There are tools which use doc strings to automatically produce online or printed documentation or to let the user interactively browse through code; it’s good practice to include doc strings in code that you write, so make a habit of it. The execution of a function introduces a new symbol table used for the local variables of the function. More precisely, all variable assignments in a functions to read the value in the local symbol table; whereas variable references first look in the local symbol table, then in the local symbol tables of enclosing functions, then in the global symbol table, and finally in the table of built-in names. Thus, global variables cannot be directly assigned a value within a function (unless named in a global statement), although they may be referenced. The actual parameters (arguments) to a function call are introduced in the local symbol table of the called function when it is called; thus, arguments are passed using call by value (where the value is always an object reference, not the value of the object).1 When a function calls another function, a new local symbol table is created for that call. A function definition introduces the function name in the current symbol table. The value of the function name has a type that is recognized by the interpreter as a user-defined function. This value can be assigned to another name which can then also be used as a function.

Annotations are stored in the annotations attribute of the function as a dictionary and haven o effect on any other part of the function. Parameter annotations are defined by a colon after the parameter name, followed by an expression evaluating to the value of the annotation. Return annotationsare defined by a literal ->, followed by an expression, between the parameter list and the colon denoting the end of the def statement.

The comparison operators in and not in check whether a value occurs (does not occur) in a sequence. The operator is and does not compare whether two objects are really the same object; this only matters for mutable objects like lists. All comparison operators have the same priority, which is lower than that of all numerical operators. Comparisons can be chained. For example,a<b==ctestswhetheraislessthanbandmoreoverbequalsc. Comparisons may be combined using the Boolean operators and the outcome of a comparison (or of any other Boolean expression) may be negated with not. These have lower priorities than comparison operators; between them, not has the highest priority and or the lowest, so that A and not B or C is equivalent to (A and (not B)) or C. As always, parentheses can be used to express the desired composition. The Boolean operators and are so-called short-circuit operators: their arguments are evaluated from left to right, and evaluation stops as soon as the outcome is determined. For example, if A and C are true but Bis false, A and B and C does not evaluate the expression C. When used as a general value and not as a Boolean, the return value of a short-circuit operator is the last evaluated argument.

Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by its class) for modifying its state. Compared with other programming languages, Python’s class mechanism adds classes with a minimum of new syntax and semantics. It is a mixture of the class mechanisms found in C++ and Modula-3. Python classes provide all the standard features of Object Oriented Programming: the class inheritance mechanism allows multiple base classes, a derived class can override any methods of its base class or classes, and a method can call the method of a base class with the same name. Objects can contain arbitrary amounts and kinds of data. As is true for modules, classes partake of the dynamic nature of Python: they are created at runtime, and can be modified further after creation. In C++ terminology, normally class members (including the data members) are public (except see below Private Variables), and all member functions are virtual. A sin Modula-3, there are no short hands for referencing the object’s members from its methods: the method function is declared with an explicit first argument representing the object, which is provided implicitly by the call. A sin Small talk, classes themselves are objects. This providesSemantics for importing and renaming. Unlike C++ and Modula-3, built-in types can be used as base classes for extension by the user. Also, like in C++, most built-in operators with special syntax (arithmetic operators, sub scripting etc.) can be redefined for class instances.(Lacking universally accepted terminology to talk about classes, I will make occasional use of Smalltalk and C++ terms. I would use Modula-3 terms, since its object- oriented semantics are closer to those of Python than C++, but I expect that few readers have heard of it.)

Objects have individuality, and multiple names (in multiple scopes) can be bound to the same object. This is known as aliasing in other languages. This is usually not appreciated on a first glance at Python, and can be safely ignored when dealing with immutable basic types (numbers, strings, tuples).However, aliasing has a possibly surprising effect on these mantic of Python code involving mutable objects such as lists, dictionaries, and most other types. This is usually used to the benefit of the program, since aliases behave like pointers in some respects. For example, passing an object is cheap since only a pointer is passed by the implementation; and if a function modifies an object passed as an argument, the caller will see the change — this eliminates the need for two different argument passing mechanisms as in Pascal.

A namespace is a mapping from names to objects. Most name spaces are currently implemented as Python dictionaries, but that’s normally not noticeable in any way (except for performance), and it may change in the future. Examples of name spaces are: these to f built-in names (containing functions such as abs(), and built-in exception names); the global names in a module; and the local names in a function invocation. In a sense the set of attributes of an object also form a namespace. The important thing to know about namespaces is that there is absolutely no relation between names in different namespaces; for instance, two different modules may both define a function maximize without confusion — users of the modules must prefix it with the module name. By the way, I use the word attribute for any name following a dot — for example, in the expression z. real, real is an attribute of the object z. Strictly speaking, references to names in modules are attribute references: in the expression modname.funcname, modname is a module object and funcname is an attribute of it. In this case there happens to be a straight forward mapping between the module’s attributes and the global names defined in the module: they share the same namespace!1 Attributes may be read-only or writable. In the latter case, assignment to attributes is

possible. Module attributes are writable: you can

write modname.the\_answer = 42. Writable attributes may also be deleted with the del statement. For example, del mod name .the\_ answer will remove the attribute the\_answer from the object named by mod name. Namespaces are created at different moments and have different lifetimes. The namespace containing the built-in names is created when the Python interpreter starts up, and is never deleted. The global namespace for a module is created when the module definition is read in; normally, module namespaces also last until the interpreter quits.The statements executed by the top-level invocation of the interpreter, either read from a script file or interactively, are considered part of a module called main, so they have their own global namespace.(The built-in names actually also live in a module; this is called built ins.) The local namespace for a function is created when the function is called, and deleted when the function returns or raises an exception that is not handled within the function. (Actually, forgetting would be a better way to describe what actually happens.) Of course, recursive invocations each have their own local namespace.

To speed uploading modules, Python caches the compiled version

of each module in the pycache directory under the name

module.version.pyc, where the version encodes the format of the compiled

file; it generally contains the Python version number. For example, in CPython release 3.3 the compiled version of spam.py would be cached as

pycache/spam.cpython-33.pyc. This naming

convention allows compiled modules from different releases and different versions of Python to coexist. Python checks the modification date of the source against the compiled version to see if it’s out of date and needs to be recompiled. This is a completely automatic process. Also, the compiled modules are platform-independent, so the same library can be shared among systems with different architectures. Python does not check the cache in two circumstances. First, it always recompiles and does not store the result for the module that’s loaded directly from the command line. Second, it does not check the cache if there is no source module. To support anon-source (compiled only) distribution, the compiled module must be in the source directory, and there must not be a source module. Some tips for experts:

You can use the -O or -OO switches on the Python command to reduce the size of a compiled module. The -O switch removes assert statements, the -OO switch removes both assert statements and doc

strings. Since some programs may rely on having these available, you should only use this option if you know what you’re doing. “Optimized”

modules have an opt- tag and are usually smaller. Future releases may change the effects of optimization.

A program doesn’t run any faster when it is read from a .pyc file than when it is read from a .py file; the only thing that’s faster about .pyc files is the speed with which they are loaded.

The module compile all can create .pyc files for all modules in a directory.

There is more detail on this process, including a flow chart of the decisions

THONNY IDE:

Thonny is as mall and light weight Integrated Development Environment. It was developed to provide a small and fast IDE, which has only a few dependencies from other packages. Another goal was to be as independent as possible from a special Desktop Environment like KDE or GNOME, so Thonny only requires the GTK2 toolkit and therefore you only need the GTK2 runtime libraries installd to run it.

For compiling Thonny yourself, you will need the GTK (>= 2.6.0) libraries and header files. You will also need the Pango, Gliband ATK libraries and header files. All these files are available at [http://www.gtk.org.](http://www.gtk.org/) Furthermore you need, of course, a C compiler and the Make tool; a C++ compiler is also required for the included Scintilla library. The GNU versions of these tools are recommended.

Compiling Thonny is quite easy. The following should do it:

% ./configure

% make

% make install

The configure script supports several common options, for a detailed list, type

% ./configure --help

There are also some compile time options which can be found in src/Thonny .h. Please see Appendix C for more information. In the case that your system lacks dynamic linking loader support, you probably want to pass the option --disable-vte to the configure script. This prevents

compiling Thonny with dynamic linking loader support to automatically load libvte.so.4 if available. Thonny has been successfully compiled and tested under Debian 3.1 Sarge, Debian 4.0 Etch, Fedora Core 3/4/5, Linux From Scratch and FreeBSD 6.0. It also compiles under Microsoft Windows

At startup, Thonny loads all files from the last time Thonny was launched. You can disable this feature in the preferences dialog (see Figure 3-4). If you specify some files on the command line, only these files will be opened, but you can find the files from the last session in the file menu under the "Recent files" item. By default this contains the last 10 recently opened files. You can change the amount of recently opened files in the preferences dialog. You can start several instances of Thonny , but only the first will load files from the last session. To run a second instance of Thonny , do not specify any file names on the command-line, or disable opening files in a running instance using the appropriate command line option.

Thonny detects an already running instance of itself and opens files from the command-line in the already running instance. So, Thonny can be used to view and edit files by opening them from other programs such as a file

manager. If you do not like this for some reason, you can disable using the first instance by using the appropriate command line option

If you have installed libvte.so in your system, it is loaded automatically by Thonny , and you will have a terminal widget in the notebook at the bottom. If Thonny cannot find libvte.so at startup, the terminal widget will not be loaded. So there is no need to install the package containing this file in order to run Thonny . Additionally, you can disable the use of the terminal widget by command line option, for more information see Section3.2.You can use this terminal (from now on called VTE) nearly as an usual terminal program like xterm. There is basic clipboard support. You can paste the contents of the clipboard by pressing the right mouse button to open the popup menu and choosing Paste. To copy text from the VTE, just select the desired text and then press the right mouse button and choose Copy from the pop up menu. On systems running the X Window System you can paste the last selected text by pressing the middle mouse button in the VTE (on 2-button mice, the middle button can often be simulated by pressing both mouse buttons together).

As long as a project is open, the Make and Run commands will use the project’s settings, instead of the defaults. These will be used whichever document is currently displayed. The current project’s settings

are saved when it is closed, or when Thonny is shut down. When restarting Thonny , the previously opened project file that was in use at the end of the last session will be reopened.

Execute will run the corresponding executable file, shell script or interpreted script in a terminal window. Note that the Terminal tool path must be correctly set in the Tools tab of the Preferences dialog - you can use any terminal program that runs a Bourne compatible shell and accept the "-e" command line argument to start a command. After your program or script has finished executing, you will be prompted to press the return key. This allows you to review any text output from the program before the terminal window is closed.

By default the Compile and Build commands invoke the compiler and linker with only the basic arguments needed by all programs. Using Set Includes and Arguments you can add any include paths and compile flags for the compiler, any library names and paths for the linker, and any arguments you want to use when running Execute.

Thonny has basic printing support. This means you can print a file by passing the filename of the current file to a command which actually prints the file.

However, the printed document contains no syntax highlighting.

STEP 1: Building of sensor network

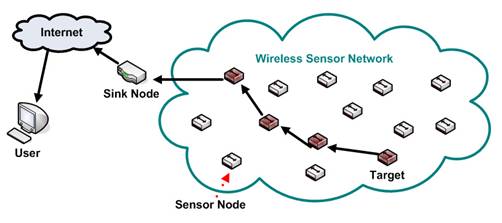


Figure 3.1: Conventional Sensor Network

A conventional sensor network is a radio network of sensor nodes with ability to sense physical parameters, store sensed data, carry out simple processing on data and forward the data through radio interface. The objective of such network is to push the data to a sink node which can then forward the data to server ( or cloud) is shown in the figure 3.1.

STEP 2: Connection of sensor network to the cloud.

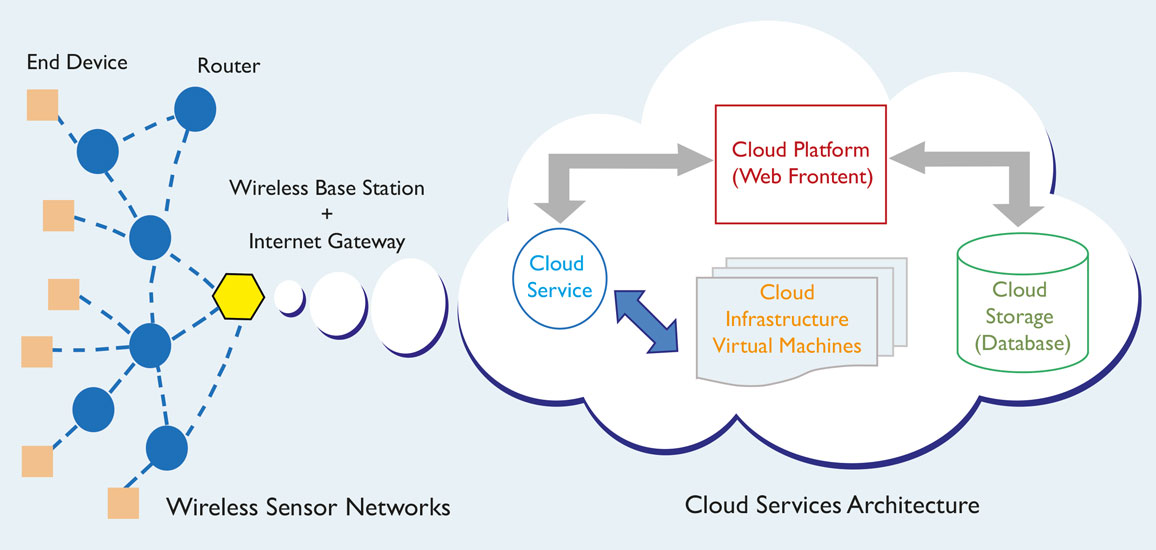


Figure 3.2: Sensor Network over cloud

Figure 3.2 shows the connection between sensor network and the cloud. However many real time applications includes sensors spread over long areas. As such they are treated as independent networks. Internet of Things is a new paradigm of connecting devices like micocontrollers and smart objects to cloud. Using IoT services, we can now connect sensors to internet directly. One of common design of sensor network includes cluster based methods where clusters are at formed by group of nodes. These are also called coordinator nodes. These nodes gather data from all neihboring nodes. If these nodes can be linked to internet with their unique Ip addresses, then the sensor network can be infinitely scaled( theoretically).

STEP 3: Interfacing with the embedded system.

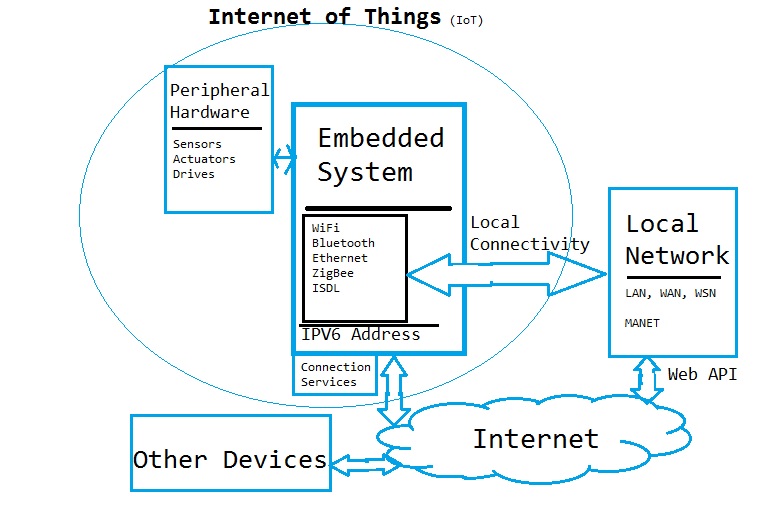


Figure 3.3: Structure Of IOT

The basic structure of IoT is presented above. Our work would include following innovative research extensions to the exising WSN and IoT framework.

Firstly we would focus purely on sensors and that too on coordinator nodes in Peripheral hardware. Rather than working on integrating individual hardware over cloud, our method would assume entire standalone sensor network as a single peripheral and would connect that to cloud though IoT. We would focus mainly on ZigBee as wireless technology as that is most accepted WSN standard.Our methods would provide not only communication services but also data gathering and analysis services. We would integrate both simulation as well as real time test beds to prove the designed concepts.

STEP 4: Possible outcome of the IoT:

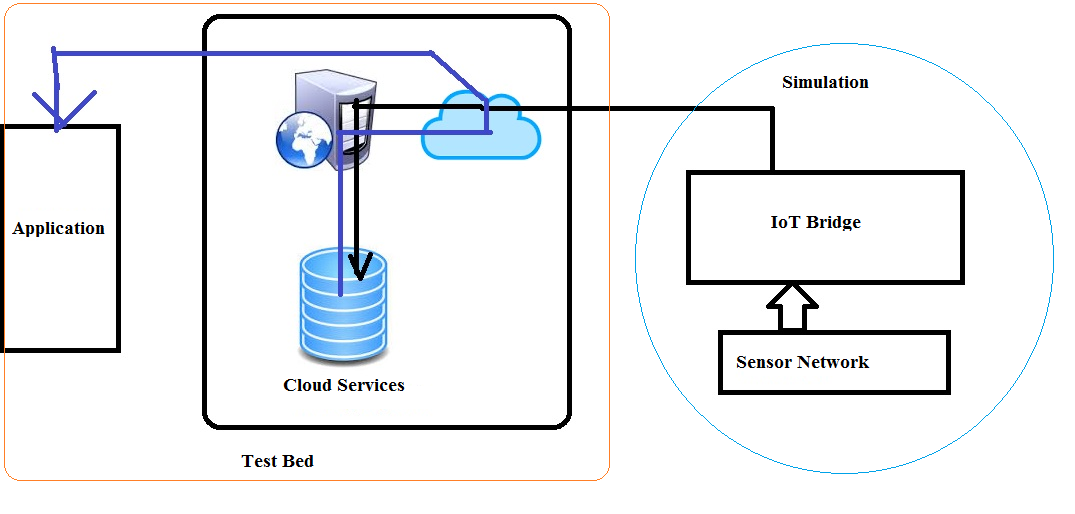


Figure 3.4: possible outcome of project

In the project work we mainly focus on building more comprehensive state of art cloud extension of WSN through IoT. Firstly the research would focus towards bettering each of the current state of art building blocks including but not limited to sensor network, coordinator protocol, data analysis in sensor network, cloud services, IoT protocols and so on. One of the first expected result would be a unique framework to connect existing WSN to cloud. Then system should prove the advantage of such extension by demonstrating the scale of improvement in data analysis services .Results should prove that IoT can be used to create mesh sensor networks and enhanced bandwidth can be used to connect sensor networks with other control system.

All the information given by the sensor network should be passed to the cloud through the javascriptnode.js . Here the cloud is a private application which can be accessed by the public and this application can be easily installed by any user which will helps the user to collect the data from the sensor network and also helps to analyze and visualize the data from the sensor network . This can be done by the matlab visualization which build in option provided in this apps. According to the data given by the apps the user can act based on the requirements ..

STEP 5: Sending data to THING SPEAK APPS:

The [Internet of Things](http://www.mathworks.com/solutions/internet-of-things/) provides access to a broad range of embedded devices and web services. [Thing Speak](http://www.thingspeak.com/) is an open data platform and API for the Internet of Things that enables us to collect, store, analyze, visualize, and act on data from sensors or actuators, such as Arduino®, and other hardware. For example, with Thing Speak one can create sensor-logging applications, location-tracking applications, and a social network of things with status updates, so that you could have your home thermostat control itself based on our current location.

The primary element of Thing Speak activity is the channel, which contains data fields, location fields, and a status field. After you create a Thing Speak channel, you can write data to the channel, process and view the data with MATLAB® code, and react to the data with tweets and other alerts.

The typical Thing Speak workflow lets you:

Create a [Channel](https://thingspeak.com/channels) and collect data

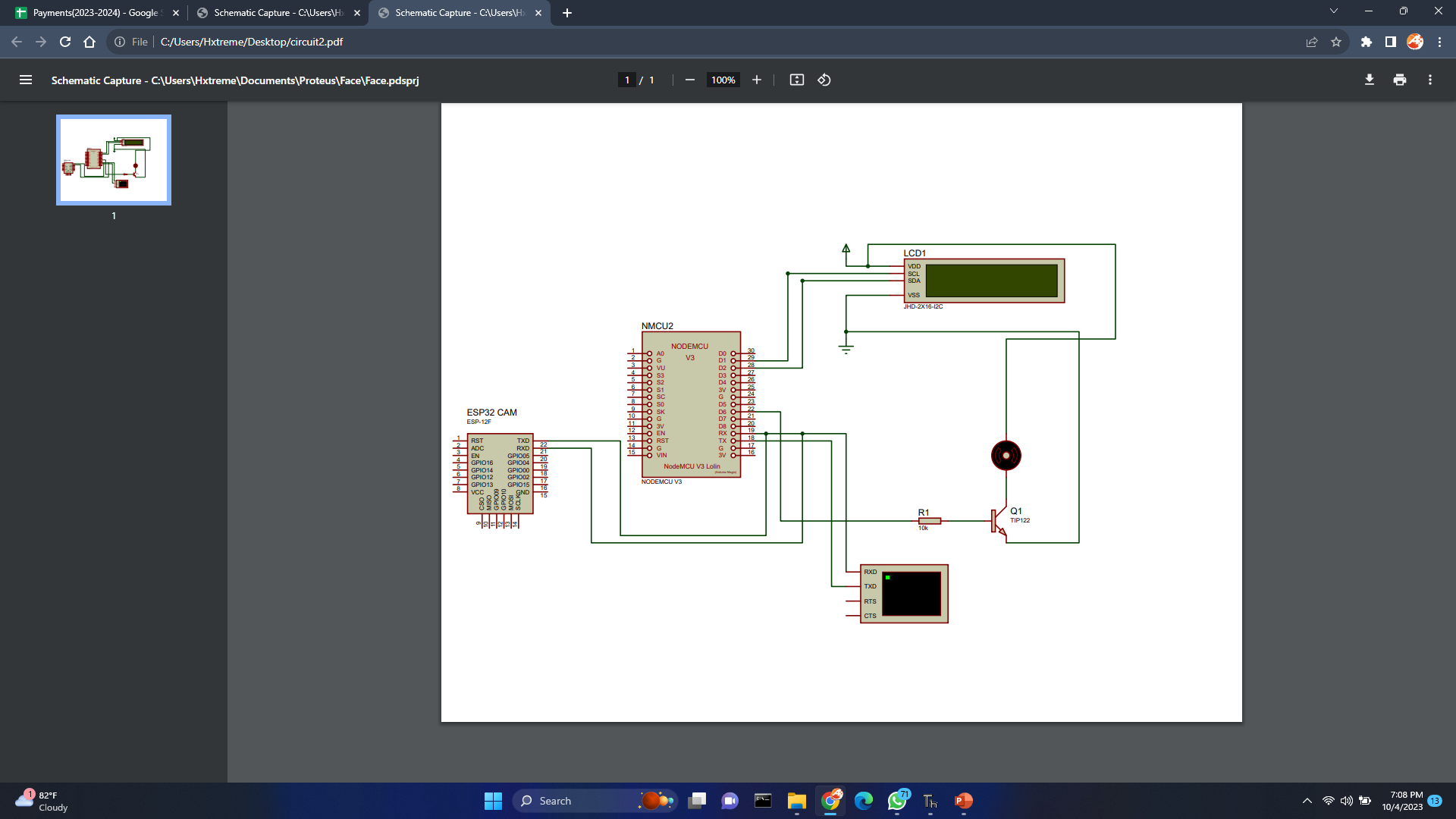
[Analyze](https://thingspeak.com/apps/matlab_analyses) and [Visualize](https://thingspeak.com/apps/matlab_visualizations) the data

Act on the data using any of several [Apps](https://thingspeak.com/apps)

CHAPTER 6

RESULTS

SIMULATION RESULTS:



CHAPTER 7

CONCLUSION:

In conclusion, this project represents a significant achievement in the development of a comprehensive Vehicle Access Authentication System. By seamlessly integrating fingerprint and face recognition technologies with IoT connectivity, it has successfully addressed the critical need for enhanced vehicle security while ensuring user convenience. The meticulously designed system architecture, robust biometric algorithms, and rigorous testing procedures have resulted in a highly reliable and efficient solution. Real-time alerts and user-friendly features, such as remote access management, contribute to an elevated user experience. Moreover, stringent security measures ensure data privacy and compliance with regulations. The project stands as a testament to the potential of technology to revolutionize vehicle security and access control, offering a solution that not only enhances security but also simplifies the way users interact with access control systems, ensuring a safer and more convenient experience.

REFERENCES:

* "Biometric car security and monitoring system using IOT",2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)
* Shuai LV,Lele Wu,Zeyu Li,Qi Xu,Xinxin Zhou,"Intelligent Car Security System Based on Fingerprint Identification and Internet of Things",2021 13th International Conference on Computational Intelligence and Communication Networks (CICN)
* Marey Jemima Raj,Sajitha Gadde,Ramesh Jayaraman,"Implementation of Biometric Access Control Using Fingerprint for Safety and Security System of Electric Vehicle",2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)
* K. Jeevitha,J. Venkatesh,V. Indhumathi,K. Krishna Veni,R. Prem Kumar,Devadarshini. M,"A Biometric Vehicle Theft Identification and Prevention Scheme using GPS Location Tracking",2023 International Conference on Advances in Computing Communication and Applied Informatics (ACCAI)
* Mridhula Ramesh,S Akruthi,K Nandhini,S Meena,S Joseph Gladwin,R Rajavel,"Implementation of Vehicle Security System using GPSGSM and Biometric",2019 Women Institute of Technology Conference on Electrical and Computer Engineering (WITCON ECE)
* Ahmed Sherif,Mohamed Elsersy,Mahmoud Nabil,Mohamed Mahmoud,Khaled H. Almotairi,"Privacy-Preserving Biometric-based Authentication Scheme for Electric Vehicles Charging System",2021 3rd IEEE Middle East and North Africa COMMunications Conference (MENACOMM)
* "Prototype of a fingerprint based licensing system for driving",2013 International Conference on Information Communication and Embedded Systems (ICICES)
* Zhaoxia Zhu,Fulong Chen,"Fingerprint recognition-based access controlling system for automobiles",2011 4th International Congress on Image and Signal Processing
* Mrinmoy Dey,Md. Akteruzzaman Arif,Md. Asif Mahmud,"Anti-theft protection of vehicle by GSM & GPS with fingerprint verification",2017 International Conference on Electrical Computer and Communication Engineering (ECCE)
* Arwa M. Ali,Heisum M. Awad,Ibrahim K. Abdalgader,"Authenticated Access Control for Vehicle Ignition System by Driveru2019s License and Fingerprint Technology",2020 International Conference on Computer Control Electrical and Electronics Engineering (ICCCEEE)
* Kgaotsang Thamoethata,Bassey Isong,Nosipho Dladlu,Adnan M. Abu-Mahfouz,"Analysis of IoT-based Vehicle Anti-Theft Security",2021 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC)
* Kirtika A P S Parameswaran,Mazlina Abdul Majid,Md. Shohidul Islam,"Fingerprint Authentication-based Traffic Offence Control and Enforcement System on Smart Mobile Devices for Smart City",2022 International Conference on Intelligent Technology System and Service for Internet of Everything (ITSS-IoE)
* Mahesh R. Pawar,Imdad Rizvi,"IoT Based Embedded System for Vehicle Security and Driver Surveillance",2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)
* Nikhil Gala,Anshu Poswalia,Riya Gharat,"Electric Bike Security: Biometric & GPS Integration for Intrusion Detection",2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)
* N.N. Nagamma,M.V. Lakshmaiah,T. Narmada,"Raspberry Pi based biometric authentication vehicle door locking system",2017 IEEE International Conference on Power Control Signals and Instrumentation Engineering (ICPCSI)