VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

"WiFi Based Digital Notice Board"

Thesis submitted in partial fulfillment of the curriculum prescribed for the award of the degree of Bachelor of Engineering in Information Science & Engineering by

1CR14IS029	Bindushree J M
1CR14CS092	Ponnaganti Manendra
1CR14CS144	Suhas M S
1CR15CS412	HariKrishna P

Under the Guidance of

Mrs.Vaishali Deshmukh
Associate Professor
Department of ISE, CMRIT, Bengaluru

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
#132, AECS LAYOUT, IT PARK ROAD, BENGALURU - 560037

2017-18

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

Certificate

This is to certify that the project entitled "**WiFi Based Digital Notice Board**" is a bonafide work carried out by **Bindushree J M,Ponnaganti Manendra, Suhas M S and Hari Krishna P** in partial fulfillment of the award of the degree of Bachelor of Engineering in **Information Science & Engineering** of Visvesvaraya Technological University, Belgaum, during the year 2017-18. It is certified that all corrections / suggestions indicated during reviews have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

----------	----------	----------
Signature of Guide	Signature of HoD	Signature of Principal
Mrs.VaishaliDeshmukh	**Dr.PremKumarRamesh**	**Dr. Sanjay Jain**
Associate Professor	Professor & Head	Principal
Department of ISE	Department of ISE	CMRIT,
CMRIT, Bengaluru - 37	CMRIT, Bengaluru - 37	Bengaluru - 37

External Viva

	Name of the Examiners	Institution	Signature with Date
1.	----------	----------	----------
2.	----------	----------	----------

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI - 590018

Declaration

We **Ms.Bindushree J M,Mr. Ponnaganti Manendra,Mr. Suhas M S and Mr. HariKrishna P** ,bonafide students of CMR Institute of Technology,Bangalore,hereby declare that the dissertation entitled "WiFi Based Digital Notice Board"has been carried out by us under the guidance of **Mrs.Vaishali Deshmukh**,Associate Professor of ISE Department,CMRIT,Bangalore,in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in **Information Science Engineering**,of the Visvesvaraya Technological University,Belgaum during the academic year 2017-2018.The work done in this dissertation report is original and it has not been submitted for any other degree in any University.

Bindushree J M
Ponnganti Manendra
Suhas M S
Hari Krishna P

Acknowledgement

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without mentioning people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So,with gratitude we acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

We would like to thank **Dr. Sanjay Jain**, Principal, CMRIT, Bangalore for providing excellent academic environment in the college and his never-ending support for the B.E program.

We would like to express our gratitude towards **Dr. H.N.Shankar**, Professor and Dean Academics and Research, CMRIT, Bangalore who provided guidance and gave valuable suggestions regarding the project.

We would like to thank **Dr. Prem Kumar Ramesh**, Professor & HOD and **Mr. Manoj Challa**,Associate Professor & Program Coordinator, Department of Computer Science, CMRIT, Bangalore who shared their opinions and experiences through which we received the required information crucial for the project.

We consider it a privilege and honour to express my sincere gratitude to my internal guide **Mrs Vaishali Deshmukh**, Associate Professor, Department of Information Science & Engineering, for his valuable guidance throughout the tenure of this project work.

We would also like to thank all the faculty members who have always been very co-operative and generous. Conclusively, we also thank all the non-teaching staff and all others who have done immense help directly or indirectly during my project

Bindushree J M
Ponnganti Manendra
Suhas M S
Hari Krishna P

Table of Contents

List of Figures

List of Tables

Abstract

This project presents a digital notice board using WiFi module. The idea behind this project is to provide its users with a simple, fast and reliable way to put up important notices in an LCD where the user can send a message to be displayed in the LCD. The message can be sent through an android application designed in this project, through the WiFi module . So, notices can be put up in an LCD display from any location in the world. It uses a microcontroller for system control, WiFi based technology for communication and sends the message through the android application. The project consists of a 32-bit ARM based microcontroller LPC2148, WiFi module, an LCD, and an android application for user interface with the hardware.This device can be used anywhere irrespective of the place of deployment provided mobile network connectivity is available.

Chapter 1

PREAMBLE

1.1 Introduction

As technology improves, efficient, financially affordable and highly productive output becomes an absolute necessity, and this leads us to be more inclined towards using automated control systems. Human intervention, although it offers variety, adaptability and interactivity, could lead to errors, as it is a natural and inevitable result of this variability. Hence, automation of a system is an accepted means to minimize human error and its impact.

Applying this to the situation under scrutiny now, the traditional methods of writing typing the notice on paper, and having a man/woman deliver the notice to the respective groups, or having himher paste the notice on the notice board, is prone to errors. The person delivering could deliver it to the wrong group, or tamper with the information being sent, etc.

With the electronics industry moving at a fast pace, we are able to solve many such problems with digital replacements. Our project, Multi Electronic Notice Board, aims at eliminating the use of paper in offices, schools & colleges, and other institutions; also minimizing the risk of errors, by replacing paper with LCD displays.

In this project, a hardware capable of displaying notices electronically using an android application has been built. In order to display notices, a user can use the android application to type a notice and click on the send button to get it displayed. The functionality can be used only if wiFi module is connected to hot spot of the host. The hardware consists of an ARM based microcontroller LPC2148 that communicates to the application through a WiFi module to receive messages. LPC2148 itself retrieves message and sends signal to switch on/off a device or display a notice.

The motivation behind such a project is mainly to reduce physical effort for operating appliances especially for aged people. Another reason for this project is overusage of paper in educational institutions for printing notices. Due to mushrooming paper usage day by day, lot of trees are being cut which is harmful for the environment. So, if notices are displayed everywhere electronically, it would reduce paper usage and make communication easier and faster. A GSM based system is exible, durable without any risk of getting hacked. Such a system has a low cost of installation and maintenance.

1.2 Problem Statement

The idea behind this project is to provide its users with a simple, fast and reliable way to put up important notices in an LCD where the user can send a message to be displayed in the LCD. The message can be sent through an android application designed in this project

1.3 Objective of the project

Presently almost all electronic notice boards are designed using wired system. One of the drawbacks of the design is the system is inflexible in term of placement. The common notice board cannot be placed anywhere because of the messy wire.

The aim of this project is to develop a digital notice board that will be used at the faculty in order to display latest information through WiFi module.The message can be send through android application

1.4 Proposed System

The project overcomes this problem by introducing an electronic display notice board interfaced to an android device through WiFi connectivity. The WiFi module receives the message from the android device that is sent to a microcontroller of 8051 family. The microcontroller displays the message on a LCD screen. This project can be used in colleges, offices, railway stations or airports for displaying any information.

1.5 Phase Description

Table 1.1: Phase Description

Phase	Task	Description
Phase1	Analysis	Analyzing the core of the IEEE paper and provide Literature review based on analysis.
Phase2	Literature survey	Collect raw data and elaborate on literature surveys.
Phase3	System analysis	Analyses the requirements of the project and lists the specific requirements needed.
Phase4	Design	Object designing and Functional description .
Phase5	Implementation	Implement the code based on the object specification .
Phase6	Testing	Test the project according to Test Specification .
Phase7	Documentation	Prepare the document for this project with conclusion and future enhancement.

1.6 Organization of the project report

The project report is organized as follows:

Chapter 2: **Literature Review** - Gives a brief overview of the survey papers and the research sources that have been studied to establish a thorough understanding of the project under consideration.

Chapter 3: **Theoretical Background** - Establishes groundwork for the proposed project by giving a detailed analysis of the project topic, existing research relevant to the project, arguments in favor and against the existing solutions and finally explores the motivation behind the proposed solution.

Chapter 4:**System Requirement Specification** - Discusses in details about the different kinds of requirements needed to successfully complete the project.

Chapter 5: **System Analysis** - gives details about several analysis that are performed to facilitate taking decision of whether the project is feasible enough or not.

Chapter 6: **System Design** - Gives the design description of the project, conceptual and detailed design well supported with design diagrams.

Chapter 7: **Implementation** - Discusses the implementation details of the project and reasons the use of the programming language and development environment.

Chapter 8: **Testing** - Briefs the testing methods used for testing the different modules in the project.

Chapter 9: **Results and Performance Analysis** - Gives the snapshots and graphs of the proposed protocols.

Chapter 10: **Conclusion and Future Scope** - Gives the concluding remarks of the project, throwing light on its future aspects.

Chapter 2

LITERATURE SURVEY

Literature survey is mainly carried out in order to analyze the background of the current project which helps to find out flaws in the existing system and guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

GSM network is widely used today whether it is for calling or SMS. Also some of the places needs urgent notices like in college, railway stations share-market , and this notice should be in real-time , so we need a real-time notice [1]. This project is our experiment to give a start to the era of real-time noticing. This project is about writing the message which is to be displayed in mobile and send it as SMS to other side. This received message is fetched into Microcontroller and after authentication it is displayed on LCD screen. Also by interfacing a voice data recording IC with Microcontroller we can also do announcements in real-time.

This paper is designed using ARM-LPC2148 interfaced with Graphical Display. At present, when information has to be updated in a notice board, it has to bedone manually. Also in present electronic systems, no matter how many displays are present, only a single notice can be sent to all of the notice boards irrespective of their places. In order to overcome this disadvantage, multiple displays along with a decoder are used to select a particular display and the corresponding information is sent through an ARM controller by using GSM technology [2]. The entries can be documented and a record may be maintained for future use by using visual basic. The controller has internal a real time clock used for synchronization of data. A resistive touch screen is used to access the previous notices and also progress details. The monitoring system consists of an image sensor which captures the images for the specified amount of time and the images can be transferred through an USB port to

a PC for storage purposes.

This paper is developed a GSM based notice board display using ARM7 controller along with LED array. The microcontrollers provide all the functionality of the display notices and wireless control. The Display is obtained on a 7X96 Light Emitting Diode (LED) dot matrix display. A desired text message from a mobile phone is sent via a Global System for Mobile Communication (GSM) to the GSM module located at the receiving end [3]. The GSM modem is connected, through MAX 232 Integrated Circuit (MAX 32IC), to the ARM7 microcontroller. The message that is stored in the Electrically Erasable Programmable Read Only Memory (EEPROM) is then displayed on the LED dot matrix display. This hardware uses regulated 5V, 500mA power supply. A three-terminal LM7805 is employed for regulation of the voltage. A bridge type full-wave rectifier is used to rectify the AC output of the secondary of 230/12V step down transformer. The system was tested to work according to specification.

In the last couple of decades, communication technology has developed by leaps and bounds. The use of Embedded System in Communication has given rise to many interesting applications. One of such applications is public addressing system (PAS). Many companies are manufacturing audio / video systems like public announcement system, CCTV, programmable sign boards etc. But all these systems are generally hardwired, complex in nature and difficult to expand. So, by adding wireless communication interface such as GSM to these systems, we can overcome their limitations .Now a days LED Message Scrolling Displays are becoming very popular .These displays are used in shopping malls, theaters, public transportation, traffic signs, highways signs, etc.This paper describes the GSM based LED display [4].

Now a days, Scrolling LED Displays are normally used in stationaries, railway stations, banks, etc. everywhere in the daily occupational life. This LEDs are preprogrammed in sense that they are already programmed to display a particular message; in case of editing or manipulating the message a person is needed around the display either by leased media or some kind of wireless media(within a limited area) which itself is an disadvantage because a the person cannot be always present at the location of the display board; a person might be at some other place and it is urgent for the person to display the message on the LED display board which is at a distant place, so this type of Scrolling LED Displays are not effective in all situations and also this display board cannot be placed anywhere because of complex and delicate wiring. GSM based LED Scrolling Display Board is a model for displaying notices or messages within any networked area through SMS which can be send by mobiles

[5]. The Project aims to provide the services of communication with LED displays remotely using GSM (that is by using SIMs SMSs via mobile phones) and updating message instantly on display board unlike a desk bound device such as PC or laptop. The user can update it even from a remote distant with simple user interaction.

Everything around us is becoming smart such as smart phones, smart televisions, smart refrigerators, so why not smart displays boards for advertisements and notices. Display boards are primary thing in any institute, organization, public utility places like bus stops, railway stations, parks, shopping malls to display information regarding platforms, various advertisements about the products, or important notices. People are now adapted to the idea of the world at its fingertips. The old wired display boards are controlled by microcontroller. To change message, we need to change the microcontroller program code again and again. By adding GSM wireless communication interface, we can overcome these limitations [6]. It is a start to the era of smart and real-time displaying of messages on display boards. This paper explains the development of GSM based Smart LED Display Boards using Short Message Service(SMS).

Summary

Literature survey is mainly carried out in order to analyze the background of the current project which helps to find out flaws in the existing system and guides on which unsolved problems we can work out.

Chapter 3

THEORETICAL BACKGROUND

Theoretical background highlighting some topics related to project work. The description contains several topics which are worth to discuss and also highlight some of their limitation that encourage going on finding solution as well as highlights some of their advantages for which reason these topics and their features are used in this project.

3.1 Arduino

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

There are various types of arduinos.Of all the type of arduinos we are using NodeMCU10 in this project.This Node arduino consist of 12pins.It also has an inbuilt wiFi module in it.It has a storage of 32kb of memory.

3.2 NodeMCU10

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. Since NodeMCU is open source platform, their hardware design is open for edit/-modify/build. NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip. The ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. F There is Version2 (V2) available for NodeMCU Dev Kit i.e. NodeMCU Development Board v1.0 (Version2), which usually comes in black colored PCB.

ESP8266 is Wi-Fi enabled system on chip (SoC) module developed by Espressif system. It is mostly used for development of IoT (Internet of Things) embedded applications. ESP8266 comes with capabilities of

→ 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2).
→ general-purpose input/output (16 GPIO).
→ Inter-Integrated Circuit (IC) serial communication protocol.
→ analog-to-digital conversion (10-bit ADC).
→ Serial Peripheral Interface (SPI) serial communication protocol.
→ IS (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO).
→ UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2).
→ pulse-width modulation (PWM).

It employs a 32-bit RISC CPU based on the Tensilica Xtensa L106 running at 80 MHz (or overclocked to 160 MHz). It has a 64 KB boot ROM, 64 KB instruction RAM and 96 KB data RAM. External flash memory can be accessed through SPI.

ESP8266 module is low cost standalone wireless transceiver that can be used for end-point IoT developments. To communicate with the ESP8266 module, microcontroller needs to use set of AT commands. Microcontroller communicates with ESP8266-01 module using UART having specified Baud rate. There are many third-party manufacturers that produce different modules based on this chip. So, the module comes with different pin availability options like

→ ESP-01 comes with 8 pins (2 GPIO pins) PCB trace antenna.
→ ESP-02 comes with 8 pins, (3 GPIO pins) U-FL antenna connector.
→ ESP-03 comes with 14 pins, (7 GPIO pins) Ceramic antenna.
→ ESP-04 comes with 14 pins, (7 GPIO pins) No ant
.

3.3 Dot Matrix

LED dot matrix display is used to display any messages that are key-in by user. The LED dot matrix that is used in this project consists of 2 blocks and 8X8 matrixes. The position of the least significant bit (LSB) and most significant bit (MSB) of the display need to be initialized.

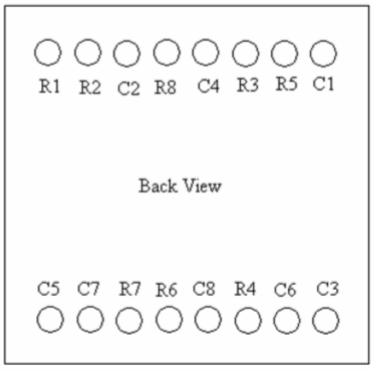

Figure 3.1: Dot Matrix Pin Configuration

The scrolling mechanism of the light on LED from the first column to the last column is applied in this board so as to display fixed or scroll characters. It is controlled by microcontroller system through decoder by enabling and disabling certain

latches. Basically, only one column for each LED dot matrix will lit up at one time. But due to the eye insensitivity, user cannot tell the different. The scrolling mechanism means the LED lighting up simultaneously to display fixed or scrolling messages.

Initially, the first column for LED display blocks is lit up. It is followed by second column while the other column is lit off. The process will continue until the last column. Then the process is repeated to produce a character or word in accordance to user request. The lit off the LED is so fast, so that all of the LED looks like lit up at the same time.

The latch played the major role in controlling the data into rows and columns for the LED. This process is controlled by software written in assembly language for microcontroller MC68HC11A1.

For example, to display B in the dot matrix, a set of hex numbers instructions must be given to the microcontroller for rows and columns. The numbers are $81, $B5, $B5, $CB for rows and $C3 for columns. Logic 0 will lit up the LED on the Dot Matrix. Figure 3.2 is an example of dot matrix display.

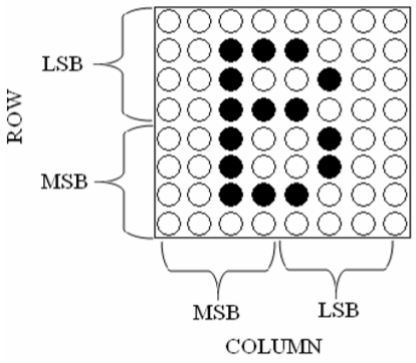

Figure 3.2: How Dot Matrix Works

A dot-matrix display is a display device used to display information on machines, clocks, railway departure indicators and many other devices requiring a simple display device of limited resolution. The display consists of a dot matrix of lights or mechanical indicators arranged in a rectangular configuration (other shapes are also possible, although not common) such that by switching on or off selected lights, text or graphics can be displayed. A dot matrix controller converts instructions from a processor into signals which turns on or off lights in the matrix so that the required

display is produced.

A dot matrix is a 2-dimensional patterned array, used to represent characters, symbols and images. Every type of modern technology uses dot matrices for display of information, including cell phones, televisions, and printers. They are also used in textiles with sewing, knitting, and weaving. An alternate form of information display using lines and curves is known as a vector display, was used with early computing devices such as air traffic control radar displays and pen-based plotters but is no longer used. Electronic vector displays were typically monochrome only, and either don't fill in the interiors of closed vector shapes, or shape-filling is slow, time-consuming, and often non-uniform, as on pen-based plotters.

3.4 Voltage Transformer

Voltage transformer gets used in electrical power system for stepping down the system voltage to a safe value which can be fed to low ratings meters and relays. Commercially available relays and meters used for protection and metering, are designed for low voltage. This is a simplest form of potential transformer definition.

The Voltage Transformer can be thought of as an electrical component rather than an electronic component. A transformer basically is very simple static (or stationary) electro-magnetic passive electrical device that works on the principle of Faradays law of induction by converting electrical energy from one value to another. The transformer does this by linking together two or more electrical circuits using a common oscillating magnetic circuit which is produced by the transformer itself. A transformer operates on the principals of electromagnetic induction, in the form of Mutual Induction.

3.5 Voltage Regulator

A voltage regulator is an electronic circuit that provides a stable DC voltage independent of the load current, temperature and AC line voltage variations. A voltage regulator may use a simple feed-forward design or may include negative feedback. It may use an electro mechanical mechanism, or electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages.

Electronic voltage regulators are found in devices such as computer power supplies where they stabilize the DC voltages used by the processor and other elements. In automobile alternators and central power station generator plants, voltage regulators control the output of the plant. In an electric power distribution system, voltage regulators may be installed at a substation or along distribution lines so that all customers receive steady voltage independent of how much power is drawn from the line.

3.6 Embedded C Language

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability. An Embedded System can be best described as a system which has both the hardware and software and is designed to do a specific task. A good example for an Embedded System, which many households have, is a Washing Machine.

3.7 Arduino 1.8.5 Software

The Arduino Integrated Development Environment - or Arduino Software (IDE) - 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 and Genuino hardware to upload programs and communicate with them. Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor 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 Software (IDE), including complete error messages and other information.

Summary

This chapter mainly concentrates on the basic theoretical background related to the topic of focus. It gives information about the platform on which this application has been developed in this chapter.

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

Software requirement Specification is a fundamental document, which forms the foundation of the software development process. It not only lists the requirements of a system but also has a description of its major feature. An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. It is important to note that an SRS contains functional and nonfunctional requirements only; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements to be.

4.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may

include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

→ Input test case must not have compilation and runtime errors.
→ The application must not stop working when kept running for even a long time.
→ The application must function as expected for every set of test cases provided.
→ The application should generate the output for given input test case and input parameters.
→ The application should generate on-demand services.

4.2 Non-Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:-

→ Product Requirements.
→ Organizational Requirements.
→ Basic Operational Requirements.

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. This should be contrasted with functional requirements that define specific behaviour or functions. The plan for implementing non-functional requirements is detailed in the system architecture. Broadly, functional requirements define what a system is supposed to do and non- functional requirements define how a system is supposed to be. Functional requirements are usually in the form of system shall do ¡requirement¿, an individual action of part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of system shall be ¡requirement¿, an overall property of the system as a whole or of a particular aspect and not a specific function. The systems' overall properties commonly mark the difference between whether the development project has succeeded or failed.

Non-functional requirements of our project include:

→ Response time The time the system takes to load and the time for responses on any action the user does.
→ Processing time - How long is acceptable to perform key functions or export / import data?
→ Throughput The number of transactions the system needs to handle must be kept in mind.
→ Storage - The amount of data to be stored for the system to function.
→ Architectural Standards The standards needed for the system to work and sustain.

4.2.1 Product Requirements

→**Correctness:** It follows a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.

→**Ease of Use:** The front end is designed in such a way that it provides an interface which allows the user to interact in an easy manner.

Non functional requirements are also called the qualities of a system. These qualities can be divided into execution quality & evolution quality. Execution qualities are security & usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

4.2.2 Organizational Requirements

Process Standards: IEEE standards are used to develop the application which is the standard used by the most of the standard software developers all over the world.

Design Methods: Design is one of the important stages in the software engineering process. This stage is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs.

4.2.3 Basic Operational Requirements

Operational requirement is the process of linking strategic goals and objectives to tactic goals and objectives. It describes milestones, conditions for success and explains how, or what portion of, a strategic plan will be put into operation during a given operational period, in the case of, a strategic plan will be put into operation during a given operational period, in the case of commercial application, a fiscal year or another given budgetary term. An operational plan is the basis for, and justification of an annual operating budget request. Therefore, a five-year strategic plan would typically require five operational plans funded by five operating budgets.

Operational plans should establish the activities and budgets for each part of the organization for the next 1-3 years. They link the strategic plan with the activities the organization will deliver and the resources required to deliver them.

An operational plan draws directly from agency and program strategic plans to describe agency and program missions and goals, program objectives, and program activities. Like a strategic plan, an operational plan addresses four questions:

→ Where are we now?
→ Where do we want to be?
→ How do we get there?

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points:

Mission profile or scenario: It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.

Performance and related parameters: It points out the critical system parameters to accomplish the mission

Utilization environments: It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.

Operational life cycle: It defines the system lifetime.

4.3 Hardware Requirements

→ Micro controller unit [8051 Family]
→ Regulator
→ Transformer
→ WiFi module
→ Matrix Display (88)

4.4 Software Requirements

→ Keil Vision IDE
→ MC Programming Language: Embedded C

Summary

This chapter gives details of the functional requirements, non-functional requirements, resource requirements, hardware requirements, software requirements etc. Again the non-functional requirements in turn contain product requirements, organizational requirements, user requirements, basic operational requirements etc.

Chapter 5

SYSTEM ANALYSIS

5.1 Overview

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

5.2 Feasibility Study

All systems are feasible when provided with unlimited resource and infinite time. But unfortunately, this condition does not prevail in practical world. So it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. Months or years of effort, thousands of rupees and untold professional embarrassment can be averted if an illconceived system is recognized early in the definition phase. Feasibility & risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. In this case there are three primary areas of interest:-

5.2.1 Performance Analysis

For the complete functionality of the project work, the project is run with the help of healthy networking environment. Normally, the OS is windows 7. The main theme of this project is to design a system that correctly identifies plant diseases and generates output specific data. Performance analysis is done to find out whether the

proposed system is time efficient and accurate. It is essential that the process of performance analysis and definition must be conducted in parallel.

5.2.2 Technical Analysis

System is only beneficial only if it can be turned into information systems that will meet the organizations technical requirement.Simply stated this test of feasibility asks whether the system will work or not when developed & installed, whether there are any major barriers to implementation. Regarding all these issues in technical analysis there are several points to focus on:-

Changes to bring in the system: All changes should be in positive direction, there will be increased level of efficiency and better customer service.

Required skills: Platforms & tools used in this project are widely used. So the skilled manpower is readily available in the industry.

Acceptability: The structure of the system is kept feasible enough so that there should not be any problem from the users point of view.

5.2.3 Economical Analysis

Economical analysis is performed to evaluate the development cost weighed against the ultimate income or benefits derived from the developed system. For running this system, we simply need a computer. All the features in this system run even on the other Operating Systems. So the system is economically feasible enough.

Summary

The main aim of this chapter is to find out whether the system is feasible enough or not. For these reasons different kinds of analysis, such as performance analysis, technical analysis,economical analysis etc is performed.

Chapter 6

SYSTEM DESIGN

Overview

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customers requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

6.1 System development methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

6.1.1 Model phases

The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of

Requirement initiation, Analysis, Design, Implementation, Testing and maintenance.

Requirement Analysis: This phase is concerned about collection of requirement of the system. This process involves generating document and requirement review.

System Design: Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:- algorithm, data structure, software architecture etc.

Coding: In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine readable compute code. Implementation: The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executables, user manuals and additional software documentation .

Testing: In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.

Maintenance: The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software.

6.1.2 Reason for choosing Waterfall Model as development method

1.Clear project objectives.
2.Stable project requirements.
3. Progress of system is measurable.
4.Strict sign-off requirements.
5.Helps you to be perfect.
6. Logic of software development is clearly understood.
7.Production of a formal specification.
8.Better resource allocation.
9.Improves quality. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage.

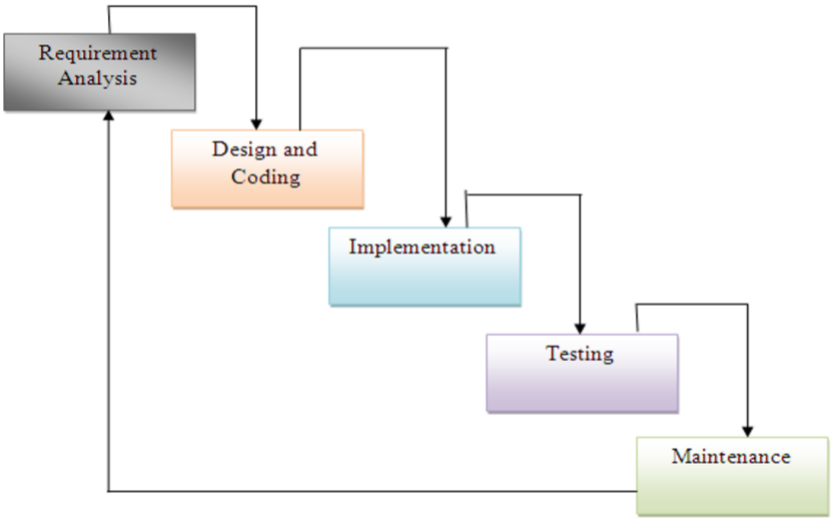

Figure 6.1: Waterfall model

6.2 Design Using UML

Designing UML diagram specifies, how the process within the system communicates along with how the objects with in the process collaborate using both static as well as dynamic UML diagrams since in this ever-changing world of Object Oriented application development, it has been getting harder and harder to develop and manage high quality applications in reasonable amount of time. As a result of this challenge and the need for a universal object modeling language every one could use, the Unified Modeling Language (UML) is the Information industries version of blue print. It is a method for describing the systems architecture in detail. Easier to build or maintains system, and to ensure that the system will hold up to the requirement changes.

6.2.1 Architectural Design

The overall logical structure of the project is divided into processing modules and a conceptual data structure is defined as Architectural Design.

From the above figure 6.2 we can observe that the dot matrix is connected to NodeMCU10 by 3 wired SPI protocol connection.The NodeMCU10 has one wifi module inbuilt in it.The NodeMCU10 has 12pins.Here in this we are using only 3 pins

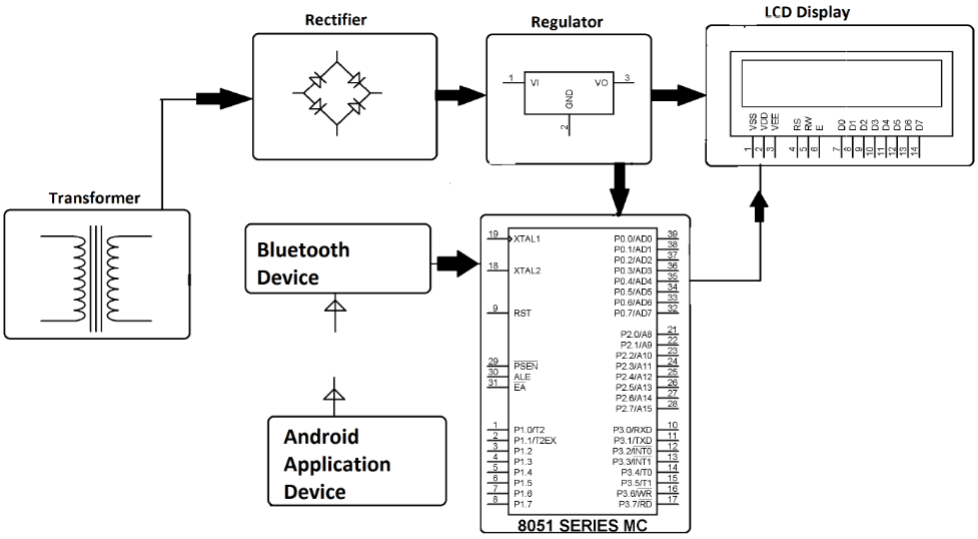

Figure 6.2: System Architecture

rest of the pins we are leaving free.The user connect the device and android application with the same hot spot.the user then open the app and press the ip address of the hot spot then after establishment of the connection between device and android application ,User will type the content to be displayed in the text box provided for text in the android application and then user will press enter then the message will be transferred from application to device through wifi and it will be displayed in dot matrix display.

6.2.2 Class Diagram

UML class diagram shows the static structure of the model. The class diagram is a collection of static modeling elements, such as classes and their relationships, connected as a graph to each other and to their contents.

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed.Figure 6.3 shows the Class Diagram.

Here the class diagram consist of three classes.They are NodeMCU,matrix Displayand server .Here the class NodeMCU was used for checking the status and for

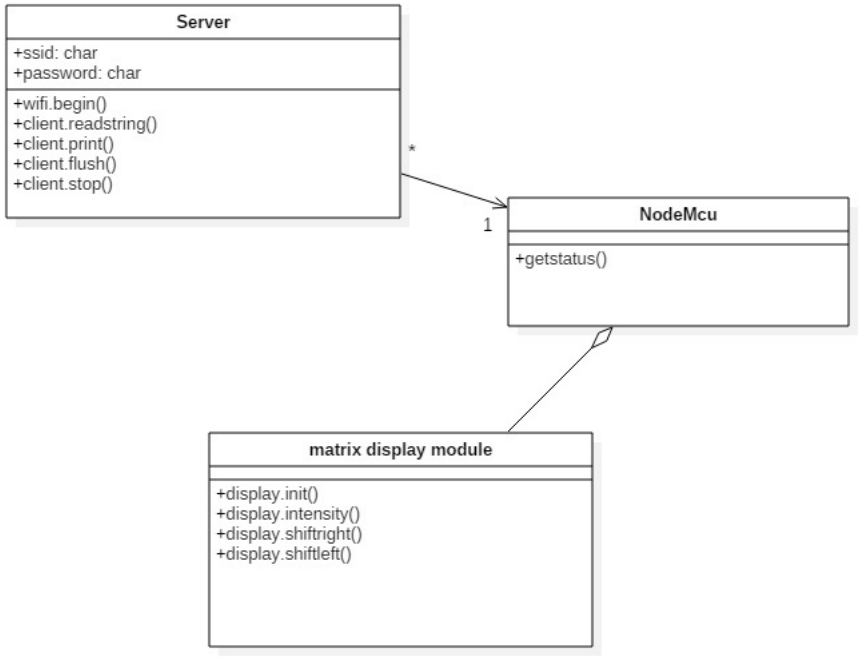

Figure 6.3: Class Diagram

transmission of information.The class matrix display is used for fecthing the user input from the user and displaying it on display.Where as server class is used for storing the local host user name and password .It was also used for connecting of the device to local host.

6.2.3 Use Case Diagram

A use case defines a goal-oriented set of interactions between external entities and the system under consideration. The external entities which interact with the system are its actors. A set of use cases describe the complete functionality of the system at a particular level of detail and it can be graphically denoted by the use case diagram.

The use case diagram of the proposed system is shown in Figure 6.4 , the system broadly classifies the functionality of the system into components that perform the actions :-connecting to network,display ip address,enter the text,fetech the input and display it on matrix.Connecting to the net is common action performed by app and lcd.Display ip,take input and display the text will be performed by lcd.enter the text will be performed by app.

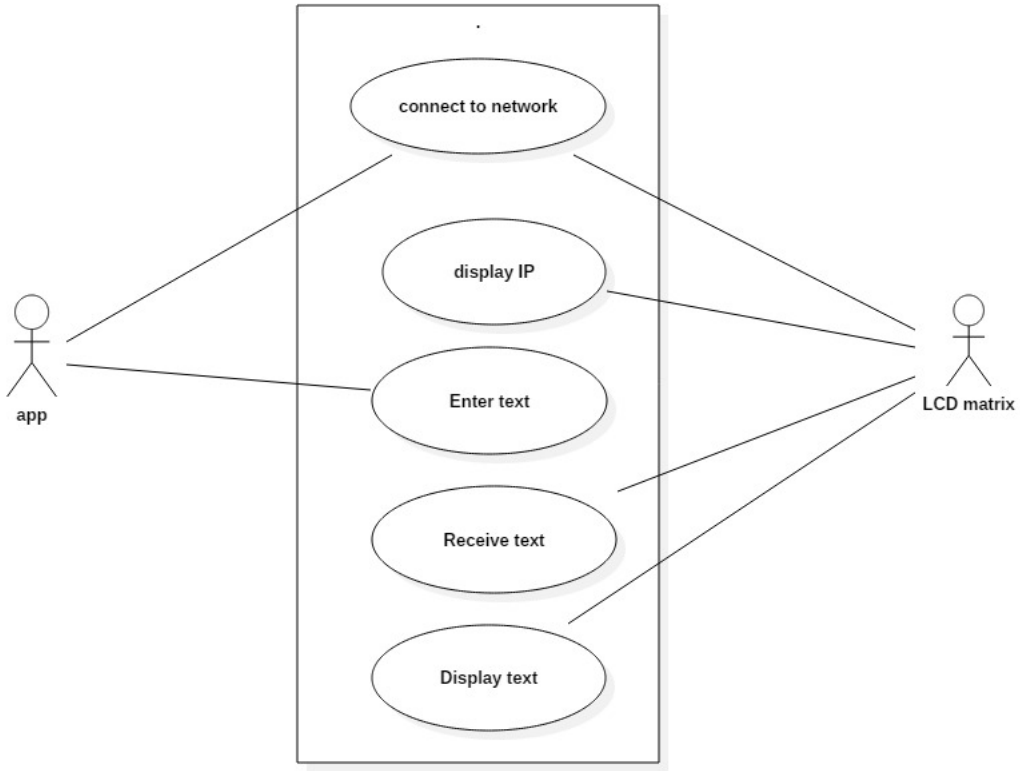

Figure 6.4: Use Case Diagram

The following are the actors and use cases:

Actors
1.app
2.Lcd

Use Cases
1.Connect to network
2.Display IP
3.Enter text
4.Receive text
5.Display text

6.2.4 Activity Diagram

An activity diagram shows the sequence of steps that make up a complex process. An activity is shown as a round box containing the name of the operation. An outgoing solid arrow attached to the end of the activity symbol indicates a transition

triggered by the completion.Figure 6.5 shows the Activity Diagram.

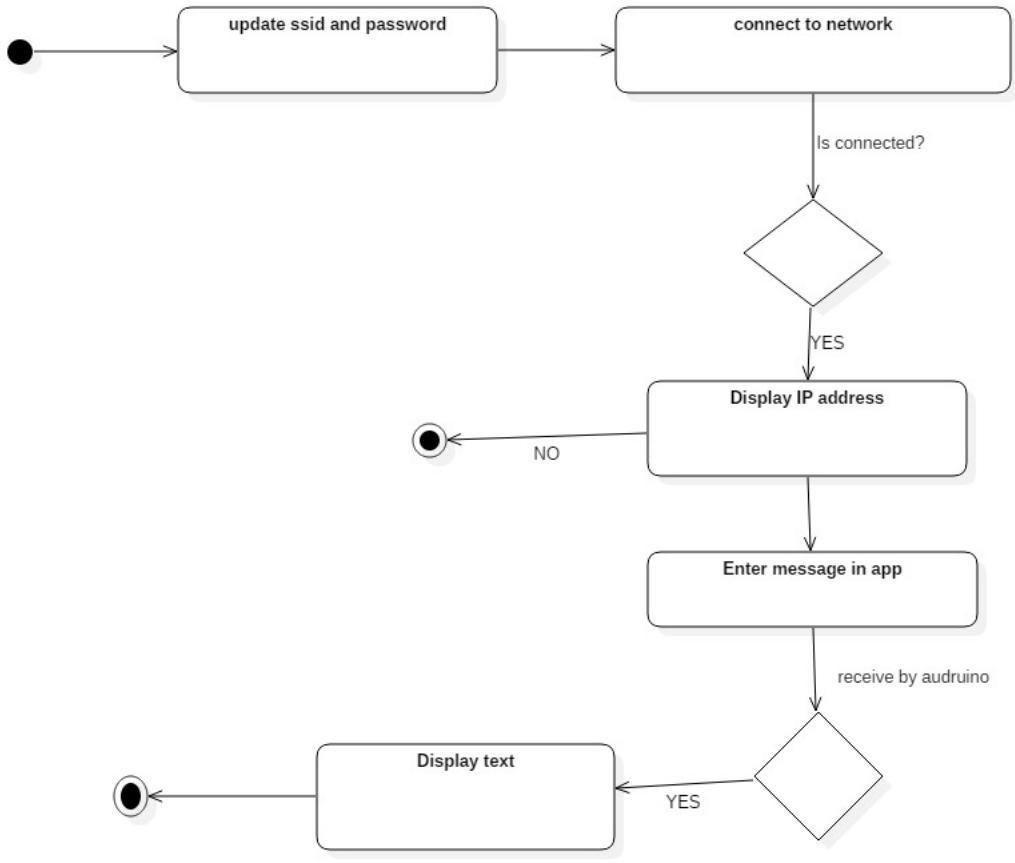

Figure 6.5: Activity Diagram

In the activity diagram first we will update the ssid and password of the wifi ,then we will check whether connection is established or not.If established we will then enter the ip address which is displayed on the lcd into the app and then we will type the text which has to be displayed on the lcd.If connection is not established then we will go back to check whether ssid and password are correct or not.

6.2.5 Sequence Diagram

Sequence diagrams are an easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and the environment. A sequence diagram shows an interaction arranged in a time sequence. A sequence diagram has two dimensions: vertical dimension represents time, the horizontal dimension represents the objects existence during the interaction.Figure 6.6 shows the

Sequence Diagram.

Basic elements:

→ **Vertical rectangle:** represent the object is active (method is being performed).

→ **Vertical dashed line:** represent the life of the object.

→ **X:** represent the life end of an object. (Being destroyed from memory).

→ **Horizontal line with arrows:** messages from one object to another.

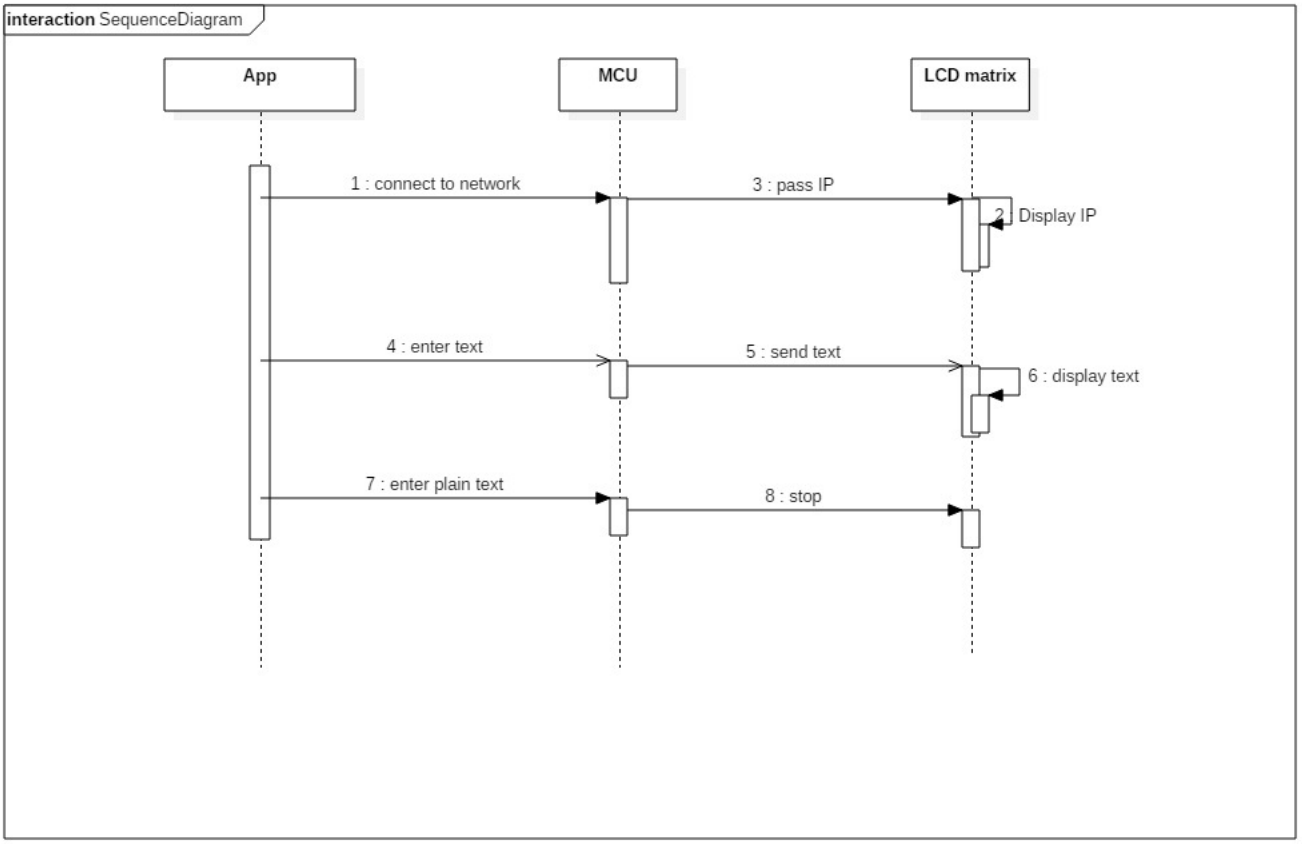

Figure 6.6: Sequence Diagram

In this sequence diagram there are 3 objects named as app,Mcu and Lcd matrix respectively.Here the user has to enter the ip address displayed on lcd in app to get connected to it.After establishing connection the user can type any text whatever he wants to display on the lcd screen in the app and then he press enter.This data is transmitted to the Mcu through Wifi then it is transferred to the lcd display via SPI protocol.Now Lcd display will fetech the content and it will display on the Lcd matrix.If the user want to stop the textthen user press plain text and enter ,automatically the displayed will be stopped.

Summary

This chapter mainly concentrates on few fundamental design concepts such as system development methodology, system architecture, class diagram, flowchart, sequence diagram, use-case diagram, activity diagram, data flow diagram etc.

Chapter 7

IMPLEMENTATION

The implementation phase of the project is where the detailed design is actually transformed into working code. Aim of the phase is to translate the design into a best possible solution in a suitable programming language. This chapter covers the implementation aspects of the project, giving details of the programming language and development environment used. It also gives an overview of the core modules of the project with their step by step flow. The implementation stage requires the following tasks.

→ Careful planning.
→ Investigation of system and constraints.
→ Design of methods to achieve the changeover.
→ Evaluation of the changeover method.
→ Correct decisions regarding selection of the platform.
→ Appropriate selection of the language for application development.

7.1 SPI

Serial Peripheral Interface (SPI) Synchronous serial communications comprises CLK, DATA IN, DATA OUT and optionally chip selects. As well as using specific SPI devices, it is very easy to expand I/O using the SPI and standard logic devices (e.g. 74HC595 and 74HC165). The SPI also enables synchronous communication between the microcontroller and peripheral, devices such as:

→ Shift registers.
→ Liquid Crystal Display (LCD) drivers.
→ Analog to Digital Converters.

→ Other microprocessors.

Microcontrollers employ multiple approaches to communicate synchronously with peripheral devices and other microcontrollers. The Serial Peripheral Interface or SPI is one of such protocols. Two devices communicating with synchronous serial interfaces operate from the same clock. The clock signal is included in the interface cable between devices. Typically, the master device creates the clock, and the slave device(s) uses the clock to latch the data (in or out.) The SSI protocol includes four I/O lines. The CS is a logic control signal from master to slave signal signifying the channel is active. The CLK line is a clock generated by the master. The DOUT is a data line driven by the master and received by the slave. The DIN is a data line driven by the slave and received by the master. In order to work properly, the transmitting device uses one edge of the clock to change its output, and the receiving device uses the other edge to accept the data. In general, the SSI interface allows data to flow both directions.

7.2 Dot Matrix

A dot-matrix display is a display device used to display information on machines, clocks, railway departure indicators and many other devices requiring a simple display device of limited resolution. The display consists of a dot matrix of lights or mechanical indicators arranged in a rectangular configuration (other shapes are also possible, although not common) such that by switching on or off selected lights, text or graphics can be displayed. A dot matrix controller converts instructions from a processor into signals which turns on or off lights in the matrix so that the required display is produced.

A dot matrix is a 2-dimensional patterned array, used to represent characters, symbols and images. Every type of modern technology uses dot matrices for display of information, including cell phones, televisions, and printers. They are also used in textiles with sewing, knitting, and weaving. An alternate form of information display using lines and curves is known as a vector display, was used with early computing devices such as air traffic control radar displays and pen-based plotters but is no longer used. Electronic vector displays were typically monochrome only, and either don't fill in the interiors of closed vector shapes, or shape-filling is slow, time-consuming, and often non-uniform, as on pen-based plotters.

7.3 Arduino IDE

The NodeMcu is the wifi module required for uploading the APP data to the Smatrix display. NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the dev kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and spiffs.

7.4 ESP8266 Arduino Core

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate tool chains to allow Arduino C/C++ to be compiled down to these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file down to the target MCU's machine language. Some creative ESP8266 enthusiasts have developed an Arduino core for the ESP8266 WiFi SoC. This is what is popularly called the "ESP8266 Core for the Arduino IDE" and it has become one of the leading software development platforms for the various ESP8266 based modules and development boards, including NodeMCUs. The nodemcu will provide power to the matrix display. The matrix display will upload its reading in the wireless module, the wireless module will upload the data to the app.

7.5 Android Application

App Inventor for Android is an open-source web application originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT). It allows newcomers to computer programming to create software applications for the Android operating system (OS). It uses a graphical interface, very similar to Scratch and the StarLogo TNG user interface, which allows users to drag-and-drop visual objects to create an application that can run on Android devices. In creating App Inventor, Google drew upon significant prior research in educational computing, as

well as work done within Google on online development environments. App Inventor and the projects on which it is based are informed by constructionist learning theories, which emphasizes that programming can be a vehicle for engaging powerful ideas through active learning. MIT App Inventor is also supported with the Firebase Database extension. This allows people to store data on Google's firebase.

Android software development is the process by which new applications are created for devices running the Android operating system. The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials.

7.6 hardware implementation

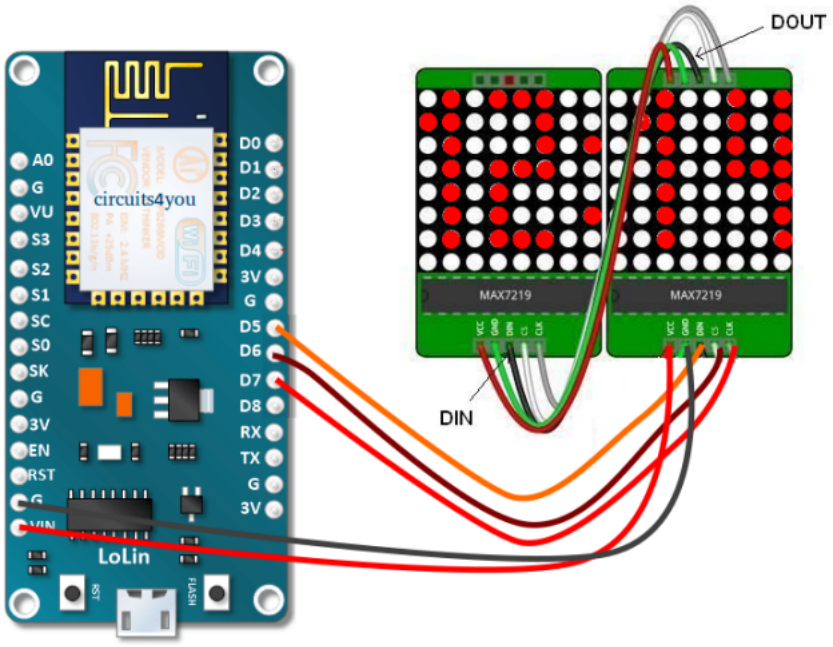

Figure 7.1: hardware wiring

Hardware wiring from Arduino to dot matrix display

- data pin D5 from arduino is connected to the display pin Din.
- data pin D6 from arduino is connected to the display pin CS
- data pin D7 from arduino is connected to the display pin CLK.
- VCC pin from arduino connected to VCC pin in display.

- Gnd pin from arduino connected to Gnd pin in display

Using data output pin in display we can connect to another display in serial fashion as Shown in above figure.It has inbuilt Wifi module, its is used for communication purpose.Separate power module is used to power the arduino and display, because arduino has 5v has output it is used to power only one display module and input is give to this from android app.

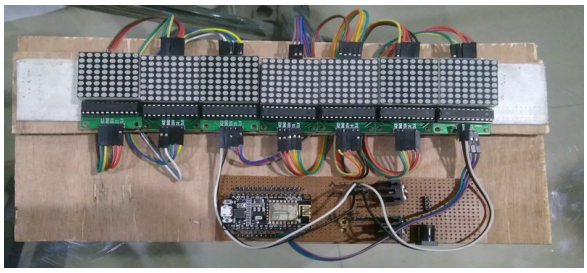

Figure 7.2: Proto Type

Summary

This chapter discusses the implementation details of the different modules of the system and gives the step by step flow of each of them.Along with these,this chapter also highlights some of the important features of the platform and language used for implementation purpose.

Chapter 8

TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve:-

→ To affirm the quality of the project.
→ To find and eliminate any residual errors from previous stages.
→ To validate the software as a solution to the original problem.
→ To provide operational reliability of the system.

8.1 Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

8.1.1 White box testing

White box testing (clear box testing, glass box testing, and transparent box testing or structural testing) uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. While white box testing is applicable at the unit, integration and system levels of the software testing process, it is typically applied to the unit. While it normally tests paths within a unit, it can also

test paths between units during integration, and between subsystems during a system level test

Though this method of test design can uncover an overwhelming number of test cases, it might not detect unimplemented parts of the specification or missing requirements, but one can be sure that all paths through the test object are executed. Using white box testing we can derive test cases that:

→ Guarantee that all independent paths within a module have been exercised at least once.
→ Exercise all logical decisions on their true and false sides.
→ Execute all loops at their boundaries and within their operational bounds.
→ Execute internal data structure to assure their validity .

8.1.2 Black box testing

Black box testing focuses on the functional requirements of the software. It is also known as functional testing. It is a software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs.

The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications. It enables us to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing is an alternative to white box technique. Rather it is a complementary approach that is likely to uncover a different class of errors in the following categories:-

→ Incorrect or missing function.
→ Interface errors.
→ Performance errors.
→ Initialization and termination errors.

Advantages
→ The test is unbiased as the designer and the tester are independent of each other.
→ The tester does not need knowledge of any specific programming languages.
→ The test is done from the point of view of the user, not the designer.

8.2 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.3 Integration Testing

Upon completion of unit testing, integration testing begins. Individual modules are combined and tested as a group. Integration testing is black box testing. The purpose of integration testing is to ensure distinct components of the application still work in accordance to user requirements. Integration testing is considered complete, when actual results and expected results are either in line or differences are explainable based on client input. It concentrates on data transfer between modules. Integration testing is a logical extension of unit testing. Two units that have already been tested are combined into a component and the interface between them is tested. Integration testing identifies problems that occur when units are combined .The errors that arise can be attributed to those occurring due to the combination of modules, resulting from errors across interface.

8.4 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "interassemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specifications.

The following examples are different types of testing that should be considered during System testing:

→ Graphical user interface testing.
→ Usability testing .
→ Software performance testing.
→ Compatibility testing .
→ Exception handling .
→ Load testing .
→ Volume testing.

Although different testing organizations may prescribe different tests as part of System testing, this list serves as a general framework or foundation to begin with.

8.5 Quality Assurance

Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confident that the product quality is meeting its goals. This is an umbrella activity that is applied throughout the engineering process. Software quality assurance encompasses:-

→ Analysis, design, coding and testing methods and tools.
→ Formal technical reviews that are applied during each software engineering.

→ Multitier testing strategy.
→ Control of software documentation and the change made to it.
→ A procedure to ensure compliance with software development standards.
→ Measurement and reporting mechanisms.

Quality Assurance (QA) is a way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers. QA is applied to physical products in pre-production to verify what will be made meets specifications and requirements, and during manufacturing production runs by validating lot samples meet specified quality controls. QA is also applied to software to verify that features and functionality meet business objectives, and that code is relatively bug free prior to shipping or releasing new software products and versions. Quality Assurance refers to administrative and procedural activities implemented in a quality system so that requirements and goals for a product, service or activity will be fulfilled. It is the systematic measurement, comparison with a standard, monitoring of processes and an associated feedback loop that confers error prevention. This can be contrasted with quality control, which is focused on process output.

Two principles included in Quality Assurance are: "Fit for purpose", the product should be suitable for the intended purpose; and "Right first time", mistakes should be eliminated. QA includes management of the quality of raw materials, assemblies, products and components, services related to production, and management, production and inspection processes. Suitable quality is determined by product users, clients or customers, not by society in general. It is not related to cost, and adjectives or descriptors such as "high" and "poor" are not applicable. For example, a low priced product may be viewed as having high quality because it is disposable, where another may be viewed as having poor quality because it is not disposable.

Software quality assurance (SQA) consists of a means of monitoring the software engineering processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards, such as ISO 9000 or a model such as CMMI.SQA encompasses the entire software development process, which includes processes such as requirements definition, software design, coding, source code control, code reviews, software configuration management, testing, release management, and product integration.

8.5.1 Quality Factor

An important objective of quality assurance is to track the software quality and assess the impact of methodological and procedural changes on improved software quality. The factors that affect the quality can be categorized into two broad groups:

1.Factors that can be directly measured.

2.Factors that can be indirectly measured.

These factors focus on three important aspects of a software product

→ Its operational characteristics.
→ Its ability to undergo changes.
→ Its adaptability to a new environment.
→ Effectiveness or efficiency in performing its mission.
→ Duration of its use by its customer.

In the context of software engineering, software quality refers to two related but distinct notions that exist wherever quality is defined in a business context: Software functional quality reflects how well it complies with or conforms to a given design, based on functional requirements or specifications. That attribute can also be described as the fitness for purpose of a piece of software or how it compares to competitors in the marketplace as a worthwhile product; Software structural quality refers to how it meets non-functional requirements that support the delivery of the functional requirements, such as robustness or maintainability, the degree to which the software was produced correctly.

Structural quality is evaluated through the analysis of the software inner structure, its source code, at the unit level, the technology level and the system level, which is in ffect how its architecture adheres to sound principles of software architecture outlined in a paper on the topic by OMG. In contrast, functional quality is typically enforced and measured through software testing. Historically, the structure, classification and terminology of attributes and metrics applicable to software quality management have been derived or extracted from the ISO 9126-3 and the subsequent ISO 25000:2005 quality model, also known as SQuaRE. Based on these models, the Consortium for IT Software Quality (CISQ) has defined five major desirable structural characteristics needed for a piece of software to provide business value: Reliability, Efficiency, Security, Maintainability and (adequate) Size.

Software quality measurement quantifies to what extent a software or system rates along each of these five dimensions. An aggregated measure of software quality can be computed through a qualitative or a quantitative scoring scheme or a mix of both and then a weighting system reflecting the priorities. This view of software quality being positioned on a linear continuum is supplemented by the analysis of "critical programming errors" that under specific circumstances can lead to catastrophic outages or performance degradations that make a given system unsuitable for use regardless of rating based on aggregated measurements.

Such programming errors found at the system level represent up to 90 percent of production issues, whilst at the unit-level, even if far more numerous, programming errors account for less than 10 percent of production issues. As a consequence, code quality without the context of the whole system, as W. Edwards Deming described it, has limited value.

To view, explore, analyze, and communicate software quality measurements, concepts and techniques of information visualization provide visual, interactive means useful, in particular, if several software quality measures have to be related to each other or to components of a software or system.

8.6 Test Cases

There are two cases:-
1.Left to Right
2.Right to left.
The following test cases were generated for case 1.

Table 8.1: Test case table1

Test case id	**Input**	**Expected 0/p**	**Status**
1	Ip address	Displaying Ip address	Pass
2	Single word	Displaying Word	Pass
3	Sentence	Displaying Sentence	Pass
4	Special Symbols	Displaying Symbols	Pass
5	Sentence including Symbols	displaying Content	Pass

The following test cases were generated for case 2.

Table 8.2: Test case table2

Test case id	**Input**	**Expected 0/p**	**Status**
1	Ip address	Displaying Ip address	Pass
2	Single word	Displaying Word	Pass
3	Sentence	Displaying Sentence	Pass
4	Special Symbols	Displaying Symbols	Pass
5	Sentence including Symbols	displaying Content	Pass

Summary

The chapter discusses the tests that are done on the system to check its functionality. Testing is carried out at three different levels from the module level to the system level checking for errors at each stage. The remarks have also been documented.

Chapter 9

RESULTS & EXECUTION

The following snapshots and graphs define the results or outputs that we will get after step by step execution of each proposed protocol for different values.

9.1 Snapshots

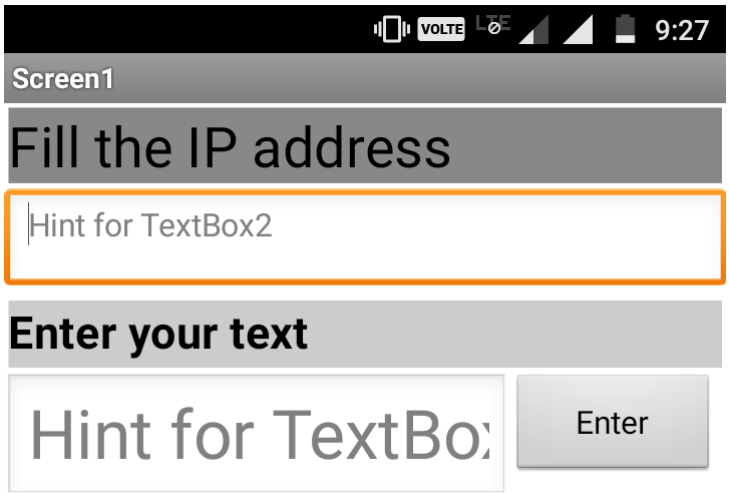

Figure 9.1: Front end of the app

The above screenshot shows the front end of the android application.It consist of two text boxes.One for entering the ip address of the local host to get connected and other box is used to type text which text user like to display on matrix display.

The screenshot in Figure 9.2 shows us how a simple word was displayed in dotmatrix from left to right.Here the user had typed the msg which to be displayed in the textbox in the app which we can observer in the figure.Here both mobile and hardware device

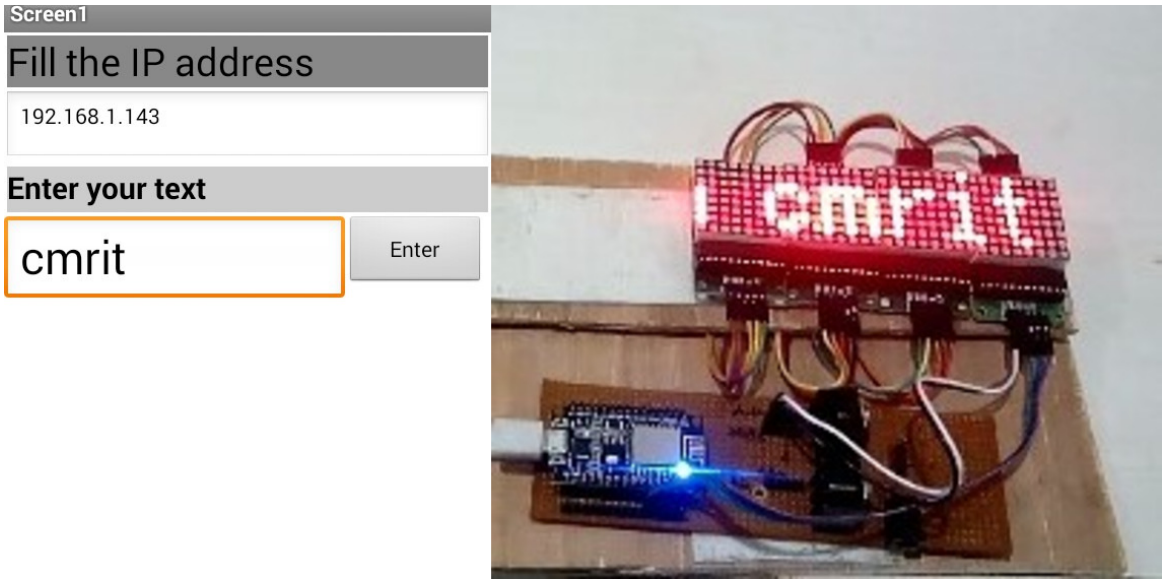

Figure 9.2: Display of sample text

are connected to same hot spot.the message transfer was happened here by using SPI protocol.

The screenshot in Figure 9.3 shows us how a digits were displayed in dotmatrix from

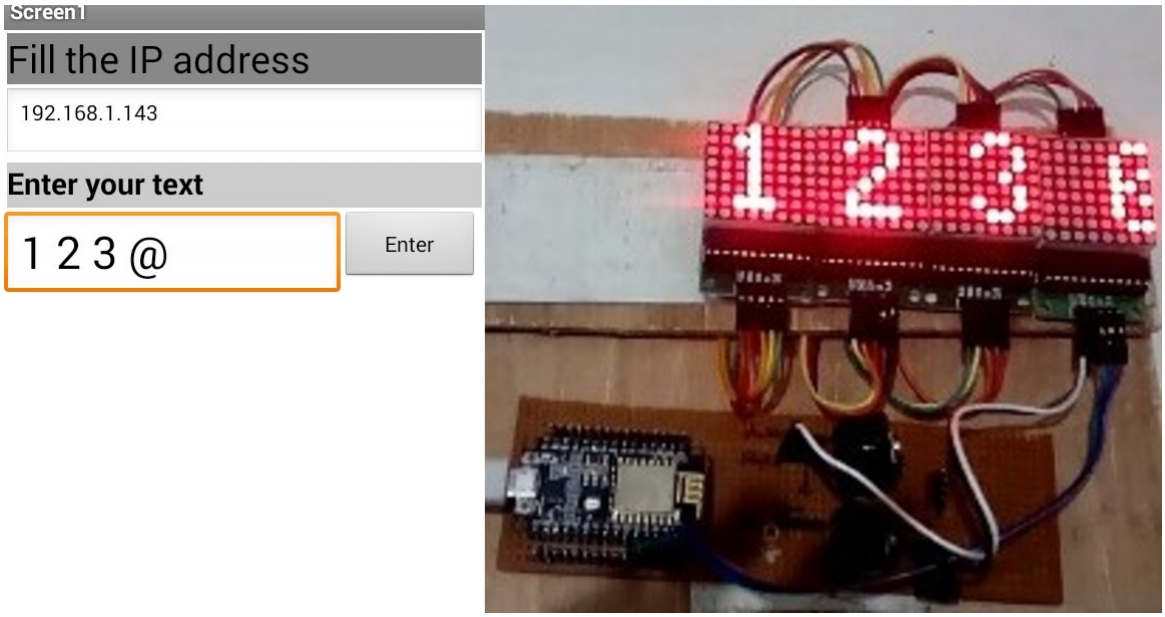

Figure 9.3: Display of Digits

left to right.Here the user had typed the msg which to be displayed in the textbox in the app which we can observer in the figure.Here both mobile and hardware device are connected to same hot spot.the message transfer was happened here by using SPI protocol.

The screenshot in Figure 9.4 shows us how a message was displayed in dotmatrix

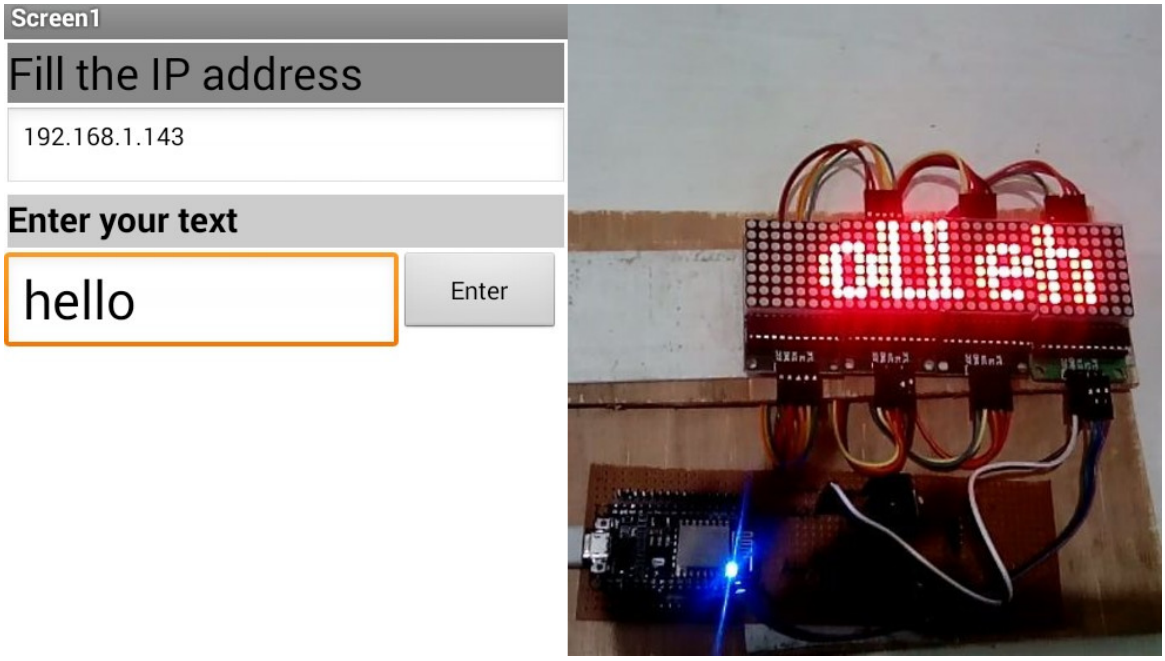

Figure 9.4: Display of text from right to left

from right to left.Here the user had typed the msg which to be displayed in the textbox in the app which we can observer in the figure.Here both mobile and hardware device are connected to same hot spot.the message transfer was happened here by using SPI protocol.

Summary

WiFi based Digital Notice Board system makes user easy to type and display the required content .It works based on SPI protocal.

Chapter 10

CONCLUSION & FUTURE ENHANCEMENT

10.1 Conclusion

The prototype of the proposed WI-FI based electronic notice board was successfully designed. It can be easily integrated with all general-purpose display board thus proving its mobility. The message is transferred using wireless technology and is eventually obtained on the LED matrix. Thus we are using modern technology to replace conventional display boards the android app interface can make this system even more user friendly and popular.

The system accepts the message from app to be displayed in the form of Short Message Service (SMS) stores it, checks for its validation and then displays it on the display unit if it meant for that particular display unit, it decided based on IP address. This system supports only one message at a time. The proposed system can be efficiently used for transfer of message instantly on campus.

The WI-FI based smart electronic notice board is efficiently designed. the smart electronic notice board system accepts new SMS, validate it, store it and display it on 8X8 LED panel. It reduces the overall development cost and also minimizes the complexity.Therefore, smart electronic notice board system becomes smarter, efficient, robust and portable.

10.2 Future Enhancement

The display unit can range from LED scrolling displays to LCD monitors. The LED scrolling displays can be set up at public transport places such as bus stations, railway stations and airports. They can also be used in offices and similar organizations for sending notices. The LCD monitors can be setup on school and college campuses for sending out notices. Also, as an extension to the current message displaying template, multiple messages can be displayed at a time, by dividing the screen to the required number of parts.

A commercial model can be able to display more than one message at a time.In our project we are sending messages via WI-FI network and displaying on a LED by utilizing AT-WF commands. The same principle can be applied to control electrical appliances at a distant location.

Robots can be controlled in a similar fashion by sending the commands to the robots. These commands are read by using AT-WF commands and appropriate action is taken. This can be used for spy robots at distant locations, utilized by the military to monitor movement of enemy troops.

REFERENCES

[1] Darshankumar C. Dalwadi, Ninad Trivedi , Amit Kasundra ,(2011) Wireless notice board our real-time solution, National Conference on Recent Trends in Engineering & Technology.

[2] Nivetha S. R, Pujitha. R, Preethi Selvaraj & Yashvanthini S.M,(2012) SMS based Wireless Notice board with monitoring system ,International Journal of Advanced Electrical and Electronics Engineering, (IJAEEE) ISSN (Print) : 2278-8948, Volume-2, Issue-3, 2013.

[3] Pawan Kumar, VikasBhrdwaj, Kiran Pal, Narayan Singh Rathor & Amit Mishra, (2012) GSMbased e-notice board: Wireless Communication, International Journal of Soft Computing andEngineering (IJSCE), ISSN: 2231-2307, Volume 2, Issue-3, pp 601-605.

[4] Prachee U. Ketkar, Kunal P. Tayade, Akash P. Kulkarni & Rajkishor M. Tugnayat, (2013) GSMmobile phone based led scrolling message display system, International Journal of ScientificEngineering and Technology (ISSN : 2277-1581), Volume 2 Issue 3, pp 149.

[5] AnuradhaMujumdar, VaishaliNiranjane & Deepika Sagne, (2014) Scrolling LED display using wireless transmission, International Journal of Engineering Development and Research (ISSN: 2321-9939), Volume 2, Issue 1, pp 475-478.

[6] BhawnaSaini,RachnaDevi, ShilpiDhankhar, Mohammad -ziaul-Haque and JagandeepKaur, (2014) Smart LED display boards, International Journal of Electronic and Electrical Engineering (ISSN0974-2174), Volume 7, Number 10, pp 1057-1067, International Research Publication House.

Appendix A

Code

```
#include <SPI.h>
#include <ESP8266WiFi.h>

byte ledPin = 2;
char ssid[] = "mani";                    // SSID of your home WiFi
char pass[] = "";                    // password of your home WiFi
WiFiServer server(80);

#include "MaxMatrix.h"
#if (defined(__AVR__))
#include <avr\pgmspace.h>
#else
#include <pgmspace.h>
#endif


PROGMEM const unsigned char CH[] = {
3, 8, B00000000, B00000000, B00000000, B00000000, B00000000, // space
1, 8, B01011111, B00000000, B00000000, B00000000, B00000000, // !
3, 8, B00000011, B00000000, B00000011, B00000000, B00000000, // "
5, 8, B00010100, B00111110, B00010100, B00111110, B00010100, // #
4, 8, B00100100, B01101010, B00101011, B00010010, B00000000, // $
5, 8, B01100011, B00010011, B00001000, B01100100, B01100011, // %
5, 8, B00110110, B01001001, B01010110, B00100000, B01010000, // &
1, 8, B00000011, B00000000, B00000000, B00000000, B00000000, // '
3, 8, B00011100, B00100010, B01000001, B00000000, B00000000, // (
3, 8, B01000001, B00100010, B00011100, B00000000, B00000000, // )
```

```
5, 8, B00101000, B00011000, B00001110, B00011000, B00101000, // *
5, 8, B00001000, B00001000, B00111110, B00001000, B00001000, // +
2, 8, B10110000, B01110000, B00000000, B00000000, B00000000, // ,
4, 8, B00001000, B00001000, B00001000, B00001000, B00000000, // -
2, 8, B01100000, B01100000, B00000000, B00000000, B00000000, // .
4, 8, B01100000, B00011000, B00000110, B00000001, B00000000, // /
4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // 0
3, 8, B01000010, B01111111, B01000000, B00000000, B00000000, // 1
4, 8, B01100010, B01010001, B01001001, B01000110, B00000000, // 2
4, 8, B00100010, B01000001, B01001001, B00110110, B00000000, // 3
4, 8, B00011000, B00010100, B00010010, B01111111, B00000000, // 4
4, 8, B00100111, B01000101, B01000101, B00111001, B00000000, // 5
4, 8, B00111110, B01001001, B01001001, B00110000, B00000000, // 6
4, 8, B01100001, B00010001, B00001001, B00000111, B00000000, // 7
4, 8, B00110110, B01001001, B01001001, B00110110, B00000000, // 8
4, 8, B00000110, B01001001, B01001001, B00111110, B00000000, // 9
2, 8, B01010000, B00000000, B00000000, B00000000, B00000000, // :
2, 8, B10000000, B01010000, B00000000, B00000000, B00000000, // ;
3, 8, B00010000, B00101000, B01000100, B00000000, B00000000, // <
3, 8, B00010100, B00010100, B00010100, B00000000, B00000000, // =
3, 8, B01000100, B00101000, B00010000, B00000000, B00000000, // >
4, 8, B00000010, B01011001, B00001001, B00000110, B00000000, // ?
5, 8, B00111110, B01001001, B01010101, B01011101, B00001110, // @
4, 8, B01111110, B00010001, B00010001, B01111110, B00000000, // A
4, 8, B01111111, B01001001, B01001001, B00110110, B00000000, // B
4, 8, B00111110, B01000001, B01000001, B00100010, B00000000, // C
4, 8, B01111111, B01000001, B01000001, B00111110, B00000000, // D
4, 8, B01111111, B01001001, B01001001, B01000001, B00000000, // E
4, 8, B01111111, B00001001, B00001001, B00000001, B00000000, // F
4, 8, B00111110, B01000001, B01001001, B01111010, B00000000, // G
4, 8, B01111111, B00001000, B00001000, B01111111, B00000000, // H
3, 8, B01000001, B01111111, B01000001, B00000000, B00000000, // I
4, 8, B00110000, B01000000, B01000001, B00111111, B00000000, // J
4, 8, B01111111, B00001000, B00010100, B01100011, B00000000, // K
4, 8, B01111111, B01000000, B01000000, B01000000, B00000000, // L
5, 8, B01111111, B00000010, B00001100, B00000010, B01111111, // M
5, 8, B01111111, B00000100, B00001000, B00010000, B01111111, // N
4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // O
4, 8, B01111111, B00001001, B00001001, B00000110, B00000000, // P
```

```
4, 8, B00111110, B01000001, B01000001, B10111110, B00000000, // Q
4, 8, B01111111, B00001001, B00001001, B01110110, B00000000, // R
4, 8, B01000110, B01001001, B01001001, B00110010, B00000000, // S
5, 8, B00000001, B00000001, B01111111, B00000001, B00000001, // T
4, 8, B00111111, B01000000, B01000000, B00111111, B00000000, // U
5, 8, B00001111, B00110000, B01000000, B00110000, B00001111, // V
5, 8, B00111111, B01000000, B00111000, B01000000, B00111111, // W
5, 8, B01100011, B00010100, B00001000, B00010100, B01100011, // X
5, 8, B00000111, B00001000, B01110000, B00001000, B00000111, // Y
4, 8, B01100001, B01010001, B01001001, B01000111, B00000000, // Z
2, 8, B01111111, B01000001, B00000000, B00000000, B00000000, // [
4, 8, B00000001, B00000110, B00011000, B01100000, B00000000, // \ backsl
2, 8, B01000001, B01111111, B00000000, B00000000, B00000000, // ]
3, 8, B00000010, B00000001, B00000010, B00000000, B00000000, // hat
4, 8, B01000000, B01000000, B01000000, B01000000, B00000000, // _
2, 8, B00000001, B00000010, B00000000, B00000000, B00000000, // ‘
4, 8, B00100000, B01010100, B01010100, B01111000, B00000000, // a
4, 8, B01111111, B01000100, B01000100, B00111000, B00000000, // b
4, 8, B00111000, B01000100, B01000100, B00101000, B00000000, // c
4, 8, B00111000, B01000100, B01000100, B01111111, B00000000, // d
4, 8, B00111000, B01010100, B01010100, B00011000, B00000000, // e
3, 8, B00000100, B01111110, B00000101, B00000000, B00000000, // f
4, 8, B10011000, B10100100, B10100100, B01111000, B00000000, // g
4, 8, B01111111, B00000100, B00000100, B01111000, B00000000, // h
3, 8, B01000100, B01111101, B01000000, B00000000, B00000000, // i
4, 8, B01000000, B10000000, B10000100, B01111101, B00000000, // j
4, 8, B01111111, B00010000, B00101000, B01000100, B00000000, // k
3, 8, B01000001, B01111111, B01000000, B00000000, B00000000, // l
5, 8, B01111100, B00000100, B01111100, B00000100, B01111000, // m
4, 8, B01111100, B00000100, B00000100, B01111000, B00000000, // n
4, 8, B00111000, B01000100, B01000100, B00111000, B00000000, // o
4, 8, B11111100, B00100100, B00100100, B00011000, B00000000, // p
4, 8, B00011000, B00100100, B00100100, B11111100, B00000000, // q
4, 8, B01111100, B00001000, B00000100, B00000100, B00000000, // r
4, 8, B01001000, B01010100, B01010100, B00100100, B00000000, // s
3, 8, B00000100, B00111111, B01000100, B00000000, B00000000, // t
4, 8, B00111100, B01000000, B01000000, B01111100, B00000000, // u
5, 8, B00011100, B00100000, B01000000, B00100000, B00011100, // v
5, 8, B00111100, B01000000, B00111100, B01000000, B00111100, // w
```

```
5, 8, B01000100, B00101000, B00010000, B00101000, B01000100, // x
4, 8, B10011100, B10100000, B10100000, B01111100, B00000000, // y
3, 8, B01100100, B01010100, B01001100, B00000000, B00000000, // z
3, 8, B00001000, B00110110, B01000001, B00000000, B00000000, // {
1, 8, B01111111, B00000000, B00000000, B00000000, B00000000, // |
3, 8, B01000001, B00110110, B00001000, B00000000, B00000000, // }
4, 8, B00001000, B00000100, B00001000, B00000100, B00000000, // ~
};

int data = D4;     // DIN pin of MAX7219 module
int load = D2;     // CS pin of MAX7219 module
int clock = D3;   // CLK pin of MAX7219 module

int maxInUse = 4;   //how many MAX7219 are connected

MaxMatrix m(data, load, clock, maxInUse); // define Library

byte buffer[8];

char string1[] = " Sourab Sharma      "; // Scrolling Text
IPAddress i;
IPAddress* x;
char p[10];
char* a;
byte c;
String s;
void setup() {
  Serial.begin(115200);                    // only for debug
   Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
```

```
  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.println(WiFi.localIP());
  s=WiFi.localIP().toString();
  s+="      ";
   Serial.println(s);
    a=&s[0];
  m.init(); // module MAX7219
  m.setIntensity(10);

}
String req;
void loop()
{
  delay(100);
  m.shiftLeft(false, true);
  printStringWithShift(a, 100);
   WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");
  while(!client.available()){
    delay(1);
  }
  // Read the first line of the request
  req = client.readStringUntil('H');
  Serial.println(req);
  client.flush();
  // Prepare the response
  String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYP
  //s += (val)?"high":"low";
  s += "</html>\n";
```

```
  // Send the response to the client
  client.print(s);
  delay(1);
  Serial.println("Client disonnected");
  delay(100);
  a=&req[6];
  m.shiftLeft(false, true);
  printStringWithShift(a, 100);
}
void printCharWithShift(char c, int shift_speed){
  if (c < 32) return;
  c -= 32;
  memcpy_P(buffer, CH + 7*c, 7);
  m.writeSprite(maxInUse*8, 0, buffer);
  m.setColumn(maxInUse*8 + buffer[0], 0);

  for (int i = 0; i < buffer[0]+1; i++)
  {
    delay(shift_speed);
    m.shiftLeft(false, false);
  }
}

// Extract characters from Scrolling text
void printStringWithShift(char* s, int shift_speed){
  while (*s != 0){
    printCharWithShift(*s, shift_speed);
    s++;
  }
}
```