VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI, KARNATAKA-590018

INTERNSHIP REPORT

on

"EMBEDDED PRODUCT DESIGN ENGINEER"

at

SOFTHOUZ SOLUTIONS PVT LT D, NATIONAL SKILL DEVELEPMENT CORPORATION

(GOVERNMENT OF INDIA)

Submitted by

MOHAMMED MUSAIB 4DM21EC023

In the partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATION ENGINEERING

Carried out at: Yenepoya Institute of Technology, Moodbidri.

Duration: October 18,2024 to December 20,2024

YENEPOYA INSTITUTE OF TECHNOLOGY

N. H. 13, THODAR, MOODBIDRI-574225, MANGALORE, D.K 2023-2024

YENEPOYA INSTITUTE OF TECHNOLOGY

THODAR, MIJAR POST, MOODBIDRI-574225

(Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Internship/Professional Practice Report entitled **"EMBEDDED PRODUCT DESIGN ENGINEER"** is an authentic record of the work carried out by **Mr. MOHAMMED MUSAIB (USN:4DM21EC023)** student of Seventh Semester in partial fulfillment of requirements for the award of Bachelor's Degree in **Electronics and Communication Engineering** prescribed by **Visvesvaraya Technological University, Belagavi** during the year **2024-25**. It is certified that all corrections/suggestions indicated for assessment have been incorporated in the report. The internship report has been approved as it satisfies the academic requirements in respect of internship work prescribed by the University.

Mr. Gangadhara

(Internship Coordinator)

Mr. Rohith A R

(Skill Development Coordinator)

Dr. Prasanna Kumar C

(Head of the Department)

I, **MOHAMMED MUSAIB (4DM21EC023)** student of 7TH semester Bachelor of Engineering in **Electronics and Communication Engineering, Yenepoya Institute of Technology Moodbidri**, hereby declare that the internship work entitled **"EMBEDDED PRODUCT DESIGN ENGINEER"** submitted to **Visvesvaraya Technological University, Belagavi** during the academic year **2024- 25** is a record of work done by me during my internship under the

Guidance of **Ms**. **Farheen Shaikh,** Assistant Professor**,** Department of ECE, YIT. This work is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering.

ACKNOWLEDGEMENT

The successful completion of any work would be incomplete without mentioning the people who made it possible. I extend my gratitude to all the people who supported me during my internship.

I thank **the Softhouz Solutions PVT LTD** and **National Skill Development Corporation** for providing the internship.

It is my great fortune that I have got an opportunity to carry out this Internship work under the guidance of **Ms. Farheen Shaikh,** Assistant Professor**,** Department of Electronics and Communication Engineering, I express my sincere thanks for his Constant Support, unparalleled guidance and limitless encouragement.

I am also thankful to our Internship Coordinator, **Mr. Gangadhara,** Assistant Professor, Department of Electronics and Communication Engineering for hirs support and concern.

I wish to convey my gratitude to **Dr. Prasanna Kumar C,** Head of the Department of Electronics and Communication Engineering for his support and encouragement.

I express my sincere gratitude to **Mr. Rohit A R, Skill** Development Coordinator for his constant support and guidance.

I am indebted to the Vice Principal, **Dr. Prabhakara B K** and the Management of Yenepoya Institute of Technology, Moodbidri for providing an environment with all facilities that helped me to complete the internship.

I take this opportunity to convey my gratitude to all the teaching and non-teaching staff of the Department of Electronics and Communication Engineering for their help.

MOHAMMED MUSAIB
(4DM21EC023)

The Embedded Product Design Engineering internship under the National Skill Development Corporation (NSDC) provided a solid foundation in embedded systems, combining both theoretical

concepts and hands-on experience. The program covered essential topics such as microcontroller architecture, system design, and programming, starting with the 8051 microcontrollers. Through this, I was able to understand the fundamentals of embedded applications and programming. Exploring Arduino and ESP modules further enhanced my knowledge, especially in IoT applications and sensor-based systems. The internship also introduced me to Raspberry Pi, expanding my understanding of single-board computers and their integration in embedded projects. Proteus was an important tool for circuit simulation and debugging, allowing me to design, test, and troubleshoot circuits effectively. Akey aspect of the program involved learning Embedded C programming, which provided the skills to develop real-time applications for embedded systems. Additionally, the experience in PCB design using EasyEDA enabled me to create and implement printed circuit boards for various projects. Practical skills in soldering and assembling circuits were also developed, allowing me to test and refine my designs. This internship experience enhanced both my technical and practical skills, preparing me for future opportunities in embedded software engineering by providing the tools and knowledge to tackle real-world embedded system challenges.

INDEX

CHAPTER

CONTENT

PAGE NO

ACKNOWLEDGEMENT

i

ABSTRACT

ii

INDEX

iii

LIST OF FIGURES

v

1

INTRODUCTION

1

2

OBJECTIVES

2

3

HARDWARE USED

3

3.1

8051(AT89C51)

3

3.2

AVR MICROCONTROLLERS

5

3.3

NODEMCU

6

3.4

RASPBERRY PI

6

4

SOFTWARE USED

8

4.1

EMBEDDED C

8

4.2

Keil uVision

8

4.3

Python PROGRAMMING

8

4.4

Python IDLE

9

4.5

PROTEUS

9

4.6

EasyEDA

10
4.7 ARDUINO IDE 10
5 TASKS 11
5.1 INTERFACING LED AND LCD 11
5.2 INTERFACING LEDS TO DISPLAY NAME 11
5.3 INTERFACING 8-SEG LED DISPLAY 12
5.4 ALERT SYSTEM 13
5.5 INTERFACING ARDUINO WITH LDR 14
5.6 INTERFACING MOTOR 14
5.7 INTERFACING DHT22 14
5.8 INTERFACING LN35 15
5.9 INTERFACING ARDUINO WITH GPS

15

5.10 INTERFACING ULTRASONIC SENSOR WITH ARDUINO 15

5.11 INTERFACING GAS SENSOR WITH 8051 16

5.12 PCB DESIGNING OF REGULATED POWER SUPPLY 16

6 PROJECT **18**

6.1 INTRODUCTION 18

6.2 PROBLEM STATEMENT 18

6.3 OBJECTIVE 19

6.4 BLOCK DIAGRAM 20

6.5 RESULT 21

7 CONCLUSION **22**

REFERENCES 23

LIST OF FIGURES

FIGURE	CONTENT	PAGE NO
3.1.1	Pin Diagram of At89c51	3
3.2.1	Arduino Uno	5
3.3.1	NodeMCU Pinout	6
3.4.1	Raspberry Pi Pin Diagram	7
5.1.1	LED And LCD Interfacing	11
5.2.1	LEDs To Display Name	11
5.3.1	Interfacing 8-Segmant Led Display	12
5.3.2	PCB Design Of 8-Segment LED Display	12
5.3.3	PCB Layout, 2D And 3D View	12

5.4.1
Alert System
13
5.4.2
Schematic of Alert System In EasyEDA
13
5.4.3
PCB Layout, 2D And 3D View Of 5.4
13
5.5.1
Interfacing Arduino with LDR and LCD
14
5.6.1
Interfacing Motor
14
5.7.1
Interfacing Of DHT22 Temperature Sensor
14
5.8.1
Arduino With LM35
15
5.9.1
Arduino With GPS Module
15
5.10.1
Ultrasonic Sensor with Arduino
15
5.11.1
Gas Sensor With 8051
16
5.12.1
Schematic Of RPS
16
5.12.2
PCB Layout Design

16

5.12.3

2D View of RPS PCB

17

5.12.4

Back View of Soldered PCB

17

5.12.5

Front View of Soldered PCB

17

6.4.1

Block Diagram

20

6.5.1

Project Model

21

CHAPTER 1

INTRODUCTION

Embedded systems are specialized computing systems designed to perform specific tasks within a larger system. Unlike general-purpose computers, which can run a variety of applications, embedded systems are dedicated to particular functions and are optimized for performance, reliability, and efficiency. These systems are typically built around microcontrollers or microprocessors that are programmed to perform predefined functions. They combine hardware and software in an integrated manner to control, monitor, and manage the operation of various devices or machines.

Embedded systems are integral components in a vast range of industries, including automotive, telecommunications, medical devices, consumer electronics, aerospace, industrial automation, and many more. Common examples of embedded systems include washing machines, microwave ovens, digital cameras, smartphones, automotive engine control units (ECUs), pacemakers, and even smart home devices. Their primary appeal lies in their ability to function autonomously, often with minimal human intervention, and provide real-time processing, making them indispensable in applications that require high levels of accuracy and responsiveness.

The architecture of an embedded system typically includes a microcontroller (MCU) or microprocessor (MPU) at its core, along with memory components (RAM, ROM, flash memory), input/output interfaces (for sensors, displays, etc.), and communication modules (such as UART, SPI, I2C, or wireless communication). The software is written to directly interact with the hardware and is often developed using programming languages such as C, C++, or assembly to ensure high efficiency and performance within the constraints of the system's resources.

Key characteristics of embedded systems include low power consumption, real-time operation, reliability, and longevity. They are designed to operate continuously, often for years, with minimal maintenance. Additionally, embedded systems are typically constrained by size and cost, which makes the optimization of both hardware and software critical.

The rise of the Internet of Things (IoT) has significantly expanded the role of embedded systems in modern technology, allowing devices to communicate and work together in smart environments. As embedded systems evolve, they continue to enable advancements in automation, machine learning, and artificial intelligence, driving innovation across industries.

CHAPTER 2
OBJECTIVES

The primary objective of this internship was to gain a comprehensive understanding and hands-on experience in various aspects of embedded systems. The goal was to acquire with necessary skills to design, implement, and troubleshoot embedded applications. These objectives include:

- **Understand Embedded Systems Fundamentals:** To gain a solid understanding of the core concepts of embedded systems, including microcontroller architecture, real-time operating systems, and the integration of software and hardware.
- **Hands-on Experience with Microcontrollers:** To develop practical skills in programming and interfacing microcontrollers, particularly 8051, Arduino, and ESP modules, for a variety of embedded applications.
- **Learn Embedded C Programming:** To learn and apply Embedded C programming techniques to develop efficient, real-time applications for embedded systems, ensuring optimal performance and resource management.
- **Explore Circuit Simulation and Design:** To gain proficiency in using tools such as Proteus for circuit simulation, debugging, and designing electronic systems, and EasyEDA for creating and testing Printed Circuit Boards (PCBs).
- **Develop IoT-based Projects**: To explore and implement Internet of Things (IoT) applications by integrating sensors, actuators, and communication modules with embedded systems to build real- world IoT solutions.
- **Understand PCB Design and Soldering Techniques:** To acquire hands-on experience in designing and fabricating PCBs using EasyEDA and to develop skills in soldering and assembling embedded systems circuits.
- **Implement Real-Time Embedded Systems:** To learn the importance of real-time processing in embedded systems and apply this knowledge to design systems that meet specific time sensitive requirements.
- **Troubleshoot and Debug Embedded Systems:** To develop problem-solving skills in troubleshooting embedded systems, including diagnosing hardware and software issues, and ensuring reliable system performance.
- **Gain Industry-Relevant Skills:** To prepare for future career opportunities by gaining practical, industry-relevant skills in embedded system design, programming, and troubleshooting, and applying them to real-world projects.
- **Develop an Understanding of System Integration:** To understand the process of integrating

various components, such as sensors, actuators, and communication modules, into a cohesive embedded system that meets the project requirements

CHAPTER 3 HARDWARE USED

- 8051 (AT89C51):

The AT89c51 is a low-voltage, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable memory. The device is manufactured using Atmel's high density non-volatile memory technology and is compatible with the industry-standard MCS-51 instruction set. The on- chip flash allows the program memory to be reprogrammed in system or by a conventional non- volatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89c51is a powerful microcomputer, which provides a highly flexible and cost- effective solution to many embedded control applications. In addition, the AT89c51is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning.

- PIN DESCRIPTION:

Fig.3.1.1: pin diagram of at89c51

- **Pins 1 to 8** − These pins are known as Port 1. This port doesn't serve any other functions. It is internally pulled up, bi-directional I/O port.
- **Pin 9** − It is a RESET pin, which is used to reset the microcontroller to its initial values.
- **Pins 10 to 17** − These pins are known as Port 3. This port serves some functions like interrupts, timer input, control signals, serial communication signals RxD and TxD, etc.
- **Pins 18 & 19** − These pins are used for interfacing an external crystal to get the system clock.
- **Pin 20** − This pin provides the power supply to the circuit.
- **Pins 21 to 28** − These pins are known as Port 2. It serves as I/O port. Higher order address bus signals are also multiplexed using this port.
- **Pin 29** − This is PSEN pin which stands for Program Store Enable. It is used to read a signal from the external program memory. Page 2 of 2
- **Pin 30** − This is EA pin which stands for External Access input. It is used to enable/disable the external memory interfacing.
- **Pin 31** − This is ALE pin which stands for Address Latch Enable. It is used to demultiplex the address-data signal of port.

- **Pins 32 to 39** − These pins are known as Port 0. It serves as I/O port. Lower order address and data bus signals are multiplexed using this port.
- **Pin 40** − This pin is used to provide power supply to the circuit.

- Features of AT89c51:
  - 8K Bytes of Re-programmable Flash Memory.
  - RAM is 256 bytes.
  - 4.0V to 5.5V Operating Range.
  - Fully Static Operation: 0 Hz to 33 MHz's
  - Three-level Program Memory Lock.
  - 256 x 8-bit Internal RAM.
  - 32 Programmable I/O Lines.
  - Three 16-bit Timer/Counters.
  - Eight Interrupt Sources.
  - Full Duplex UART Serial Channel.
  - Low-power Idle and Power-down Modes.
  - Interrupt recovery from power down mode.
  - Dual data pointer.
  - Power-off flag.
  - Flexible ISP programming (byte and page mode).

- AVR MICROCONTROLLERS:

AVR is a family of microcontrollers developed since 1996 by Atmel, acquired by Microchip Technology in 2016. These are modified Harvard architecture 8-bit RISC single-chip microcontrollers. AVR was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to one-time programmable ROM, EPROM, or EEPROM used by other microcontrollers at the time.

AVR microcontrollers find many applications as embedded systems. They are especially common in hobbyist and educational embedded applications, popularized by their inclusion in many of the Arduino line of open hardware development boards. It includes microcontrollers such as ATmeg8, ATmeg16, ATmeg32, ATmeg328P, ATmeg48/8/16.

- ARDUINO UNO:

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

Fig.3.2.1 Arduino UNO

- NodeMCU:

NodeMCU V3 is an open-source firmware and development kit that plays a vital role in designing an IoT product using a few script lines. Multiple GPIO pins on the board allow us to connect the board with other peripherals and are capable of generating PWM, I2C, SPI, and UART serial communications.

The interface of the module is mainly divided into two parts including both Firmware and Hardware where former runs on the ESP8266 Wi-Fi SoC and later is based on the ESP-12 module. The firmware is based on Lua – A scripting language that is easy to learn, giving a simple programming environment layered with a fast-scripting language that connects you with a well- known developer community.

Fig.3.3.1 NodeMCU Pinout

- Raspberry Pi:

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside of its target market for uses such as robotics. There are many different raspberry Pi boards one among which we used for internship is raspberry Pi 4.

- Raspberry Pi 4:

Raspberry Pi 4 Model B is the latest product in the popular Raspberry Pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation Raspberry Pi 3 Model B+, while retaining backwards

compatibility and similar power consumption. For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems.

This product's key features include a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, hardware video decode at up to 4Kp60, up to 4GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on).

The dual-band wireless LAN and Bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time to market.

Fig.3.4.1 Raspberry Pi Pin Diagram

- **EMBEDDED C:**

CHAPTER 4 SOFTWARE USED

Embedded C Programming is the soul of the processor functioning inside each and every embedded system we come across in our daily life, such as mobile phone, washing machine, and digital camera. Each processor is associated with an embedded software. The first and foremost thing is the embedded software that decides functioning of the embedded system. Embedded C language is most frequently used to program the microcontroller.

Earlier, many embedded applications were developed using assembly level programming. However, they did not provide portability. This disadvantage was overcome by the advent of various high level languages like C, Pascal, and COBOL. However, it was the C language that got extensive acceptance for embedded systems, and it continues to do so. The C code written is more reliable, scalable, and portable; and in fact, much easier to understand.

- Keil uVision:

Keil uVision is a powerful integrated development environment (IDE) designed for embedded systems programming, particularly for ARM-based microcontrollers. It provides developers with a complete toolchain, including a code editor, compiler, and debugger, to simplify the creation of embedded software. The IDE supports a wide range of microcontrollers, such as ARM Cortex-M, ARM7, and ARM9 families, making it a versatile choice for diverse applications. With its ARM C/C++ compiler, Keil uVision ensures efficient code optimization and performance, while its integrated debugging tools, including support for ULINK and JTAG hardware debuggers, enable real-time debugging directly on target devices.

Additionally, Keil uVision offers a robust simulation environment, allowing developers to simulate microcontroller peripherals and I/O signals without the need for physical hardware. Its user-friendly interface and extensive device libraries make it an ideal platform for both beginners and experienced embedded developers. This versatility and ease of use have made Keil uVision a popular choice for firmware development in IoT, robotics, automotive, and industrial automation projects.

- Python programing:

Python is a high-level, interpreted, and general-purpose programming language known for its simplicity and readability. Developed by **Guido van Rossum** and first released in 1991, Python emphasizes code readability with its clean and straightforward syntax, making it an ideal language for beginners and experienced developers alike. It supports multiple programming paradigms,

including procedural, object-oriented, and functional programming.

Python's extensive standard library, combined with a vast ecosystem of third-party libraries, allows developers to build applications in diverse domains such as web development, data analysis, artificial intelligence, machine learning, automation, and scientific computing. Tools like Django and Flask enable web development, while libraries such as NumPy, Pandas, TensorFlow, and PyTorch cater to data science and AI applications. Its cross-platform compatibility and active community further contribute to Python's popularity, making it one of the most widely used programming languages in the world.

- Python IDLE:

Python IDLE (Integrated Development and Learning Environment) is the default IDE that comes bundled with Python installations, designed to help users write, edit, and execute Python code. It is lightweight, user-friendly, and suitable for beginners and casual programmers. IDLE provides a Python shell for interactive testing and debugging, alongside a simple text editor for writing and saving Python scripts. Its features include syntax highlighting, auto-indentation, and support for basic debugging with breakpoints and stepping through code.

As a cross-platform tool, Python IDLE runs on Windows, macOS, and Linux systems. Its simplicity makes it ideal for learning Python programming, but its limited features compared to more advanced IDEs like PyCharm or VS Code may not suit complex projects. Nevertheless, IDLE remains a reliable and accessible choice for those starting with Python or for quick scripting and testing tasks.

- PROTEUS:

Proteus is a software suite used for electronic circuit design, simulation, and PCB layout development. Developed by Labcenter Electronics, it is popular among engineers, students, and hobbyists for its comprehensive tools that support the design and testing of both analog and digital circuits. One of its key features is the ability to simulate microcontrollers like Arduino, PIC, and ARM-based processors alongside the electronic components in a circuit. This makes it particularly useful for embedded system projects, allowing developers to test their code and hardware virtually before implementing them physically.

Proteus offers an intuitive interface for designing schematics and a powerful PCB design module for creating custom layouts. It supports a wide range of electronic components and allows users to import libraries or create custom parts. Its real-time simulation capability enables users to visualize how their circuits perform under different conditions, reducing prototyping time and costs. This versatility makes Proteus a valuable tool for applications ranging from educational projects to

professional product development in the electronics field.

- EasyEDA:

EasyEDA is a cloud-based electronic design automation (EDA) software used for designing circuit schematics, simulating electronic circuits, and creating printed circuit board (PCB) layouts. It is widely popular due to its user-friendly interface, accessibility through web browsers, and integration of multiple tools in one platform. EasyEDA caters to beginners and professionals alike, offering a seamless environment to design, test, and produce PCBs. It includes a vast component library, supporting millions of parts from suppliers like LCSC and JLCPCB, which simplifies component selection and procurement.

One of EasyEDA's standout features is its ability to simulate circuits in real-time, enabling users to test functionality before manufacturing. The platform also supports exporting design files in various formats and provides a direct link to manufacturers for quick PCB prototyping and production. With

collaboration tools for sharing designs and an active community for support, EasyEDA has become a go-to tool for electronic engineers, students, and hobbyists.

- ARDUINO IDE:

The Arduino IDE (Integrated Development Environment) is an open-source software platform used for writing, compiling, and uploading code to Arduino microcontroller boards. Designed to be beginner-friendly, the IDE simplifies embedded programming with its straightforward interface and support for the Arduino programming language, which is based on simplified C/C++. It features a text editor for coding, a message area for notifications, and a serial monitor for debugging and communication with the connected Arduino board.

The Arduino IDE supports a wide range of microcontroller boards, including Arduino Uno, Nano, Mega, and many more. It includes a library manager to access pre-built libraries for various sensors, modules, and devices, making it easy to expand functionality. Additionally, it allows users to upload their code to boards via USB and supports third-party board packages, further increasing its versatility. The IDE is cross-platform (compatible with Windows, macOS, and Linux) and widely used in applications like IoT, robotics, and DIY electronics projects.

CHAPTER 5 TASKS

- ITERFACING LED AND LCD:

Fig.5.1.1 LED and LCD Interfacing

- INTERTFACING LEDS TO DISPLAY NAME:

Fig.5.2.1 LEDs to Display Name

- INTERFACING 8-SEGMANT LED DISPLAY:

Fig.5.3.1 Interfacing 8-Segmant Led Display

Fig.5.3.2 PCB Design of 8-Segment LED Display

Fig.5.3.3 PCB Layout, 2D and 3D view

- ALERT SYSTEM:

Fig.5.4.1 Alert system

Fig.5.4.2 schematic design of alert system in EasyEDA

Fig.5.4.3 PCB Layout, 2D and 3D view of 5.4

- INTERFACING ARDUINO WITH LDR AND LCD DISPLAY:

Fig.5.5.1 INTERFACING ARDUINO WITH LDR AND LCD

- INTERFACING MOTOR:

Fig.5.6.1 Interfacing Motor

- INTERFACING DHT22 TEMPERATURE SENSOR:

Fig.5.7.1 Interfacing of DHT22 Temperature Sensor

- INTERFACING OF LM35 TEMPERATURE SENSOR WITH ARDUINO

Fig.5.8.1 Arduino with LM35

- INTERFACING ARDUINO WITH GPS:

Fig.5.9.1 Arduino with GPS Module

- INTERFACING ULTRASONIC SENSOR WITH ARDUINO:

Fig.5.10.1 Ultrasonic Sensor with Arduino

- INTERFACING GAS SENSOR WITH AT89C52:

Fig.5.11.1 gas sensor with 8051

- PCB DESIGNING OF REGULATED POWER SUPPLY:

Fig.5.12.1 Schematic of RPS

Fig.5.11.2 PCB Layout Design

Fig.5.12.3 2D view of RPS PCB

Fig.5.12.4 Back View of Soldered PCB

Fig.5.12.5 Front View of Soldered PCB

CHAPTER 6 PROJECT

IOT BASED "VOICE BASED HOME AUTAMATION"

- **INTRODUCTION:**

Voice-based smart home automation using Arduino Uno, NodeMCU, and Alexa allows control of home devices via voice commands. The Arduino Uno manages sensors and actuators, while NodeMCU provides Wi-Fi connectivity to the system. Alexa processes voice commands, such as "Alexa, turn on the light," and sends them to NodeMCU. NodeMCU then communicates with the Arduino to control the devices. This system enhances convenience, energy efficiency, and security by enabling hands-free control. It also benefits accessibility, especially for the elderly or those with disabilities. The system is scalable, allowing easy integration of additional devices or sensors.

- PROBLEM STATEMENT:

In today's fast-paced world, managing household appliances manually can be inconvenient and inefficient. Traditional switches require physical interaction, which can be challenging for elderly individuals, people with disabilities, or those who are away from home. Additionally, energy wastage due to human negligence, such as forgetting to turn off lights or appliances, is a common issue.

While smart home solutions exist, many require complex setups, expensive hardware, or lack seamless voice control integration. There is a need for an affordable, user-friendly, and efficient home automation system that can be controlled remotely and hands-free.

This project addresses these challenges by developing a voice-controlled home automation system using Arduino Uno, Alexa, and a 4-channel relay module. The system allows users to control multiple electrical devices via simple voice commands, enhancing convenience, accessibility, and energy efficiency while providing a cost-effective solution for smart home automation.

- OBJECTIVES:

  - Develop a Voice-Controlled System – Enable hands-free operation of home appliances using Alexa voice commands.

  - Integrate Arduino Uno with IoT – Use Arduino Uno as the central controller to process voice commands and control appliances.

  - Implement a 4-Channel Relay Module – Control multiple electrical devices efficiently through relay switching.

  - Enhance User Convenience – Provide an easy-to-use home automation solution, reducing the need for manual switching.

- Improve Accessibility – Assist elderly individuals and people with disabilities by offering a voice- activated control system.

- Increase Energy Efficiency – Reduce energy wastage by enabling remote and automated control of appliances.

- Ensure Cost-Effectiveness – Design a budget-friendly smart home solution using readily available components.

- Provide Remote Control Functionality – Allow users to control appliances from anywhere via Alexa's cloud-based services.

- Ensure System Reliability and Safety – Implement secure and stable connections to prevent electrical hazards and malfunctions.

- Promote IoT-Based Automation – Encourage the adoption of smart home technologies by demonstrating a practical IoT application.

- BLOCK DIAGRAM:

Fig.6.4.1 Block Diagram

The block diagram illustrates a voice-based home automation integrating various components for intelligent mobility control. The Arduino uno and NodeMCU serves as the central communication hub, interfacing with the Alexa for live voice communication and connecting to Arduino uno to 4-channel relay for remote operation. The Arduino Uno processes inputs from the 4-channel relay and LCD display for controlling home appliances. It controls the blub, fan, refrigerator and many more home appliances which regulates two DC motors powered by a battery. Additionally, an LCD with an I2C module provides real-time feedback. This system enhances accessibility, independence, and monitoring for mobility-impaired users.

- RESULTS:

Fig.6.5.1 Project Model

CHAPTER 7
CONCLUSION

The Voice-Based Home Automation System using Arduino Uno, Alexa, and a 4-Channel Relay Module provides an efficient, cost-effective, and user-friendly solution for smart home automation. By integrating voice commands, the system enhances convenience, accessibility, and energy efficiency, making daily tasks easier for users, including the elderly and individuals with disabilities.

This project successfully demonstrates the potential of IoT and voice-controlled technology in modern home automation. With further improvements, such as adding more appliances, mobile app control, or integrating sensors for automation, this system can be expanded to create a more advanced smart home ecosystem. Ultimately, this project contributes to the growing demand for smart living solutions by offering a practical, scalable, and affordable approach to home automation.

.

REFERENCES

- **Arduino Official Documentation** – https://www.arduino.cc

(Provides technical details on Arduino Uno and its functionalities.)

- **Amazon Alexa Developer Documentation** – https://developer.amazon.com/alexa

(Covers Alexa integration and voice command processing.)

- **IoT-Based Home Automation Research Paper** – IEEE Xplore

(Discusses the role of IoT in smart home automation.)

- **Relay Module Datasheet** – Manufacturer's Website

(Technical specifications and working principles of a 4-channel relay module.)

- **Blynk IoT Platform Documentation** – https://blynk.io

(Explains how to integrate cloud-based control for IoT applications.)

- **ESP8266/ESP32 Wi-Fi Module Documentation** – https://www.espressif.com

(Details on using Wi-Fi modules for IoT connectivity in automation projects.)

- Smart Home Automation Using Voice Control – ResearchGate Paper

(Explores different methods of implementing voice-controlled home automation.)

- Energy Efficiency in Smart Homes – Journal of Sustainable Energy

(Analyzes the impact of home automation on energy savings.)

- Home Automation Using Arduino and IoT – Springer Link

(Discusses various applications of Arduino in smart home projects.)

- **YouTube Tutorials on Alexa and Arduino Integration** – Various sources

(Practical demonstrations of voice-controlled automation using Arduino and Alexa.)