DESIGN AND CONSTRUCTION OF A SMART REFRIGERATION SYSTEM

<u>Submitted by</u>

Tauhidur Rahman
Md. Sumon Hosssen
Md. Shohanur Rahman
Md. Raihan Ali
Md. Tanzir Hasan Taz

<u>Supervised by</u>

AMM. Shamsul Alam
Associate Professor
Department of Mechanical Engineering
Sonargaon University (SU)

Submitted to the
DEPARTMENT OF MECHANICAL ENGINEERING
SONARGAON UNIVERSITY (SU)

In partial fulfilment of the requirements for the Award of the Degree
Of
Bachelor of Science in Mechanical Engineering.

May 2026

DESIGN AND CONSTRUCTION OF A SMART REFRIGERATION SYSTEM

<u>Submitted by</u>

Tauhidur Rahman	ME2201026318
Md. Sumon Hosssen	ME2203028214
Md. Shohanur Rahman	ME2203028201
Md. Raihan Ali	ME2203028215
Md. Tanzir Hasan Taz	ME2203028267

<u>Supervised by</u>

AMM. Shamsul Alam
Associate Professor
Department of Mechanical Engineering
Sonargaon University (SU)

Submitted to the
DEPARTMENT OF MECHANICAL ENGINEERING
SONARGAON UNIVERSITY (SU)

In partial fulfilment of the requirements for the Award of the Degree
Of
Bachelor of Science in Mechanical Engineering.

May 2026

APPROVAL

This is to certify that the project work entitled "Design and Construction of a Smart Refrigeration System" has been carried out by Tauhidur Rahman (ME2201026318), Md. Sumon Hosssen (ME2203028214), Md. Shohanur Rahman (ME2203028201), Md. Raihan Ali (ME2203028215), and Md. Tanzir Hasan Taz (ME2203028267).

This project is an original work carried out by us and has not been previously published or submitted, in whole or in part, to any other institution for the award of any degree.

The work has been submitted in partial fulfillment of the requirements for the degree of Bachelor of Science (B.Sc.) in Mechanical Engineering in the year 2026, under the Department of Mechanical Engineering, Sonargaon University (SU), and has been carried out under the supervision of **AMM. Shamsul Alam, Associate Professor,** Department of Mechanical Engineering, Sonargaon University (SU).

AMM. Shamsul Alam
Associate Professor
Department of Mechanical Engineering
Sonargaon University (SU)

DECLARATION

We solemnly declare that the work presented in this project report has been carried out by us and has not been previously submitted to any university or organization for the award of any degree or certificate.

We hereby ensure that the work presented here does not breach any existing copyright.

We further indemnify the university against any loss or damage arising from a breach of the foregoing obligation.

…………………..	…………………..
Tauhidur Rahman	Md. Sumon Hosssen
ME2201026318	ME2203028214
…………………..	…………………..
Md. Shohanur Rahman	Md. Raihan Ali
ME2203028201	ME2203028215

…………………..
Md. Tanzir Hasan Taz
ME2203028267

ACKNOWLEDGEMENT

First, we started in the name of Almighty Allah. This thesis is accomplished under the supervision of **AMM. Shamsul Alam, Associate Professor**, Department of Mechanical Engineering, Sonargaon University. It is a great pleasure to acknowledge our profound gratitude and respect to our supervisor for this consistent guidance, encouragement, helpful suggestions, constructive criticism and endless patience through the progress of this work. The successful completion of this thesis would not have been possible without his persistent motivation and continuous guidance.
The authors are also grateful to **Prof. Md. Mostofa Hossain, Head of the Department of Mechanical Engineering** and all the respected teachers of the Mechanical Engineering Department for their cooperation and significant help in completing the thesis work successfully.

Tauhidur Rahman
Md. Sumon Hosssen
Md. Shohanur Rahman
Md. Raihan Ali
Md. Tanzir Hasan Taz

ABSTRACT

The advancement of embedded systems and smart automation technologies has created new opportunities for improving conventional household appliances. This thesis presents the design and implementation of a smart refrigeration system capable of real-time temperature monitoring and automated cooling control. The proposed system integrates an ESP32 microcontroller, DS18B20 digital temperature sensor, relay module, LCD display, and wireless communication technology to enhance the functionality and efficiency of a refrigerator.

The system continuously monitors the internal temperature of the refrigerator and displays real-time information through both an LCD display and a web-based interface. Based on user-defined temperature settings, the controller automatically activates or deactivates the compressor using a relay module to maintain the desired cooling condition. The design also allows users to control the refrigerator and adjust temperature settings through physical buttons or a mobile device connected via Wi-Fi.

The developed prototype emphasizes reliability, ease of implementation, and energy-efficient operation. Experimental observations demonstrate that the system can accurately maintain the required temperature range and respond effectively to internal temperature variations. The integration of embedded control and wireless monitoring significantly improves user convenience and refrigeration performance compared to conventional systems.

The proposed smart refrigeration system provides a practical and affordable solution suitable for household, laboratory, commercial, and pharmaceutical storage applications. Furthermore, this research establishes a foundation for future developments such as IoT-based cloud monitoring, intelligent food management, and advanced energy optimization techniques.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

CHAPTER-1
INTRODUCTION

1.1 Introduction

A smart refrigerator is an advanced refrigeration system that integrates embedded electronics, sensors, communication technologies, and automatic control mechanisms to enhance the traditional functions of a conventional refrigerator. Unlike ordinary refrigerators that operate with fixed cooling cycles and manual user interaction, a smart refrigerator is capable of monitoring internal conditions, analyzing data, and responding intelligently to changing environments and user needs.

At its core, a smart refrigerator employs various sensors to continuously observe the internal state of the appliance. These sensors provide real-time data to a microcontroller or embedded processor, which processes the information and makes decisions to optimize cooling performance. For example, the system can adjust compressor operation based on temperature fluctuations, door opening frequency, or food load conditions, thereby improving energy efficiency and maintaining food quality.

One of the key characteristics of a smart refrigerator is connectivity. Through wireless technologies such as Wi-Fi or Bluetooth, the refrigerator can communicate with external devices like smartphones or cloud platforms. This connectivity allows users to remotely monitor temperature, receive alerts for abnormal conditions (such as power failure or door left open), and control certain functions of the refrigerator without physical interaction. Such features significantly enhance user convenience and system reliability.

Smart refrigerators also contribute to food management and pharmaceutical operations. In more advanced implementations, smart refrigerators may incorporate inventory tracking, expiration reminders, and data logging features that support informed decision-making. These capabilities are particularly important in addressing the global issue of food wastage.

From an energy perspective, smart refrigerators play a vital role in sustainable household technology. Intelligent control algorithms enable efficient compressor usage, reduced power consumption, and adaptive operation during peak and off-peak energy periods. This not only lowers electricity costs for users but also reduces environmental impact. In regions with limited energy resources, such efficiency improvements are especially valuable.

In summary, a smart refrigerator is an intelligent, connected, and adaptive appliance that extends beyond basic cooling functionality. By combining sensing, automation, and communication technologies, it improves food preservation, enhances user convenience, and promotes energy efficiency. As part of the broader smart home ecosystem, smart refrigerators represent a significant step toward intelligent living environments and modern appliance design.

1.2 Smart Refrigerator

A smart refrigerator is an advanced household appliance that combines traditional refrigeration with modern embedded and communication technologies. It uses sensors and a microcontroller to monitor internal conditions such as temperature and door status in real time. Through wireless connectivity, it can provide remote monitoring by the user. Smart control mechanisms help improve energy efficiency and maintain optimal food storage conditions. As part of smart home systems, smart refrigerators enhance convenience, reduce food waste, and support sustainable energy use.

1.3 Background of the Study

Refrigeration systems play a crucial role in modern households by preserving food quality, reducing waste, and ensuring food safety. Traditional refrigerators operate as standalone appliances with limited intelligence, offering basic cooling and storage functions without awareness of usage patterns, food conditions, or energy efficiency optimization. With the rapid advancement of embedded systems, sensors, and Internet of Things (IoT) technologies, conventional home appliances are being transformed into smart systems capable of autonomous decision-making and real-time monitoring.

A smart refrigerator integrates sensing, communication, and control technologies to enhance food management, energy efficiency, and user convenience. By continuously monitoring internal operations, a smart refrigerator can provide alerts, optimize cooling cycles, and support intelligent decision-making. These capabilities align with the global demand for smart homes, sustainable energy usage, and reduced food wastage.

1.4 Problem Statement

Despite the widespread use of refrigerators, several challenges remain unresolved in traditional systems. Users often face issues such as unintentional food spoilage, inefficient energy consumption, lack of real-time temperature monitoring, and absence of remote control mechanisms. Additionally, manual monitoring of stored food items is inconvenient and error-prone, leading to unnecessary waste and increased household costs.

In regions where energy resources are limited and cost-sensitive, inefficient refrigeration further exacerbates economic and environmental concerns. Therefore, there is a need for an smartdigital refrigeration system that can monitor internal conditions accurately, provide timely alerts, and improve overall efficiency without significantly increasing system complexity or cost.

1.5 Objectives

The primary objective of this thesis is to design and develop a smart refrigerator system that enhances functionality beyond conventional refrigeration. The specific objectives include:

1.5.1 To design a smart refrigerator with real-time temperature monitoring and controlling system

1.5.2 To measure internal temperature accurately using sensor

1.5.3 To show the temperature via Display and Web interface

1.5.4 To control cooling operation precisely via control panel and also using phone

1.6 Methodology

The methodology used in this study is as follows:

1.6.1 Creating an idea for the design and construction of a circuit for the Smart Refrigerator. And designing a circuit diagram to know which components we need to construct it.

1.6.2 Collecting all the components and programming the microcontroller to control the whole system.

1.6.3 Setting up all the components in a board & integrating with a mini refrigerator.

1.7 Workflow Diagram

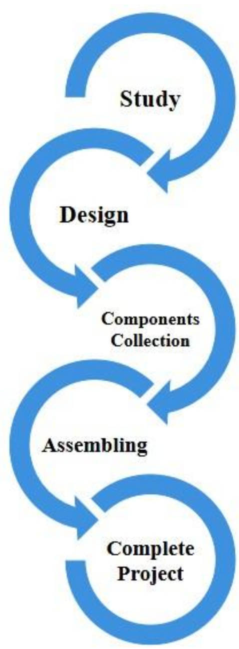


Figure 1.1: Working Flow Diagram

1.8 Research Outline

This project book consists of six chapters. The first chapter contains the statement of the Introduction, Background history, Problem Statement, Work Flow Diagram, objectives of the study and Methodology. Chapter two contains system Literature Review details. Chapter Three System Architecture, Discussed Block Diagrams, Circuit Diagram, Working principles and Cost analysis. Chapter Four describes the hardware implementation with component details and the software which we have used for our work. Chapter Five deals with the result Analysis and shows the complete prototype, Advantages, Applications and Limitations of the project that we have built. In the final chapter, we have discussed the Future Scope and conclusion of the project.

CHAPTER 2
LITERATURE REVIEW

2.1 Literature Review

Refrigeration and Air Conditioning by R. Arora and A. Kumar

This book explains the fundamental principles of refrigeration, including vapor compression cycles, temperature control methods, and performance parameters. It highlights the importance of maintaining stable temperature ranges for food and medical storage applications, which forms the theoretical foundation of this project.

Refrigeration and Air Conditioning by W. F. Stoecker and J. W. Jones

The authors discuss traditional refrigeration system design, heat transfer mechanisms, and control strategies. The text emphasizes efficiency and reliability, providing baseline knowledge against which smart and automated refrigeration systems can be compared.

Espressif Systems – ESP32 Technical Reference Manual

This manual details the architecture, GPIO configuration, communication protocols, and power management features of the ESP32 microcontroller. It supports the selection of ESP32 as a suitable controller for real-time monitoring and automation in smart refrigeration systems.

Maxim Integrated – DS18B20 Datasheet

The datasheet describes the working principle, accuracy, operating range, and 1-Wire communication protocol of the DS18B20 temperature sensor. It confirms the sensor's suitability for precise temperature measurement in refrigeration environments.

Hitachi Ltd. – Refrigeration Systems and Applications

This reference discusses industrial and commercial refrigeration applications, including cold storage and pharmaceutical use. It highlights the need for continuous temperature monitoring and system reliability in sensitive environments.

"Design of Smart Refrigerator Using IoT," International Journal of Engineering Research & Technology (IJERT)

This research presents an IoT-based smart refrigerator system using sensors and microcontrollers. It demonstrates real-time monitoring and automation, supporting the feasibility of smart control approaches used in this project.

A. Al-Fuqaha et al., IEEE Communications Surveys & Tutorials
This survey paper explains IoT architectures, communication models, and data processing techniques. It provides a conceptual framework for integrating embedded systems with smart monitoring solutions.

World Health Organization – Temperature Monitoring of Vaccines
The WHO document emphasizes strict temperature control requirements in pharmaceutical and medical storage. It reinforces the importance of accurate monitoring systems, validating the real-world relevance of smart refrigeration.

2.2 Summary

From the reviewed literature, it is evident that traditional refrigeration systems focus primarily on cooling performance, while modern applications demand precise monitoring and automation. Existing studies and technical manuals support the use of microcontrollers, digital temperature sensors, and automated control techniques to improve efficiency and reliability. However, many smart refrigeration solutions remain costly or complex. This project aims to bridge that gap by developing a **low-cost, reliable, and practical smart refrigerator system**, suitable for both domestic and sensitive commercial applications such as pharmaceutical storage.

CHAPTER 3
SYSTEM ARCHITECTURE

3.1 Proposed System

The proposed smart refrigerator system is designed to enhance conventional refrigeration by integrating sensing, control, and communication technologies. The system continuously monitors internal conditions and performs controlled actions to ensure optimal condition and energy efficiency. It also provides user interaction features through a communication interface.

The overall system consists of sensors for data acquisition, a microcontroller for processing and control, and output devices for indication and actuation. Wireless communication enables remote monitoring and notification. The proposed design emphasizes simplicity, low cost, and suitability for household & Commercial use.

3.2 Block Diagram

The block diagram of the proposed smart refrigerator system represents the functional relationship between different system components. Sensors placed inside the refrigerator collect temperature. This data is sent to the microcontroller, which processes the information and makes control decisions.

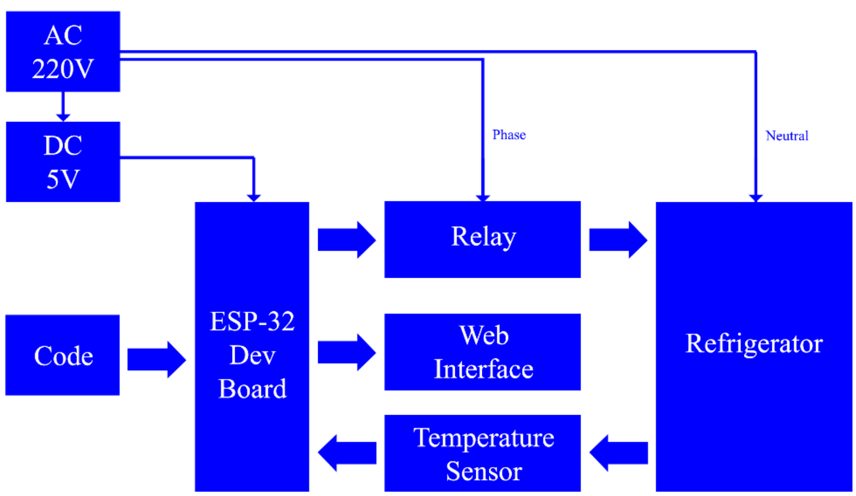


Figure 3.1: Block Diagram of Smart Refrigeration System.

3.3 Circuit Diagram

The circuit diagram illustrates the electrical connections between the microcontroller, sensors, communication module, and output devices. Sensors are connected to the appropriate input pins of the microcontroller, while output devices such as relays or displays are connected through suitable driver circuits.

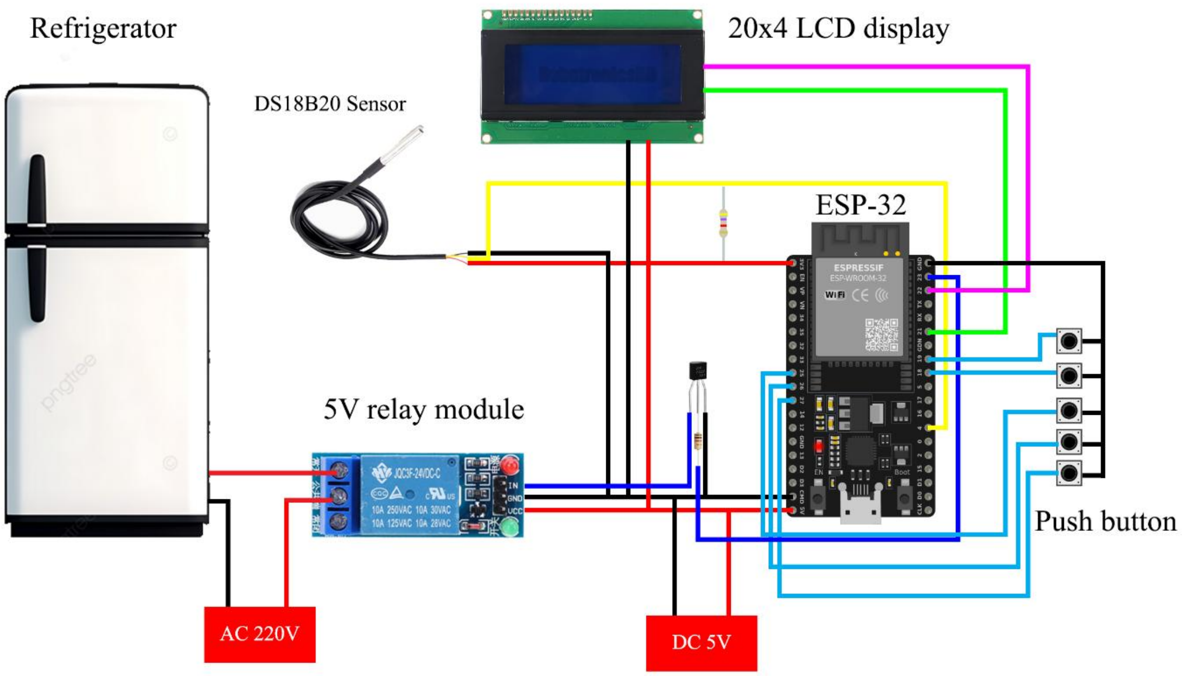


Figure 3.2: Circuit Diagram of Smart Refrigerator

3.4 Working Principle

The working principle of the smart refrigerator system begins with continuous data acquisition from the sensors. The microcontroller reads sensor values and compares them with predefined threshold values. If abnormal conditions are detected, appropriate actions are taken automatically. The system may activate or deactivate cooling components and generate alerts for the user. The monitoring and control process runs continuously to ensure reliable and efficient operation. The processes are:

3.4.1 Power Supply Initialization

When the system is powered ON, the 5V USB adapter supplies power to the ESP32, LCD display, relay module, and sensor system.

3.4.2 Temperature Measurement

The DS18B20 temperature sensor, mounted inside the refrigerator chamber, continuously measures the internal temperature and sends data to the ESP32 through the One-Wire communication protocol.

3.4.3 Data Processing
The ESP32 compares the measured temperature with the user-defined preset temperature stored in memory.
3.4.4 Compressor Control
If the measured temperature becomes higher than the preset threshold, the ESP32 sends a control signal to the relay module. The relay then connects the compressor power line and starts the cooling process.
When the temperature reaches the desired preset value, the ESP32 deactivates the relay module, disconnecting power from the compressor and stopping cooling operation.
3.4.5 User Interface and Monitoring
The LCD display continuously shows: current temperature, preset temperature, power status. The same information is also transmitted through Wi-Fi to the web interface, allowing remote monitoring and control through a mobile device.
3.4.6 User Adjustment
Users can: increase/decrease temperature, turn the refrigerator ON/OFF, save preset temperature values using either the physical push buttons or the web interface.

3.5 Connection Details
The circuit was completed by connecting and integrating the components together to complete the system and make it functional. The connections are shown below:
3.5.1 The Esp-32 is located inside the Control Box. It is connected to the Relay module, Temperature sensor and the Switches via breadboard. All the components are powered by an external 5V power supply.

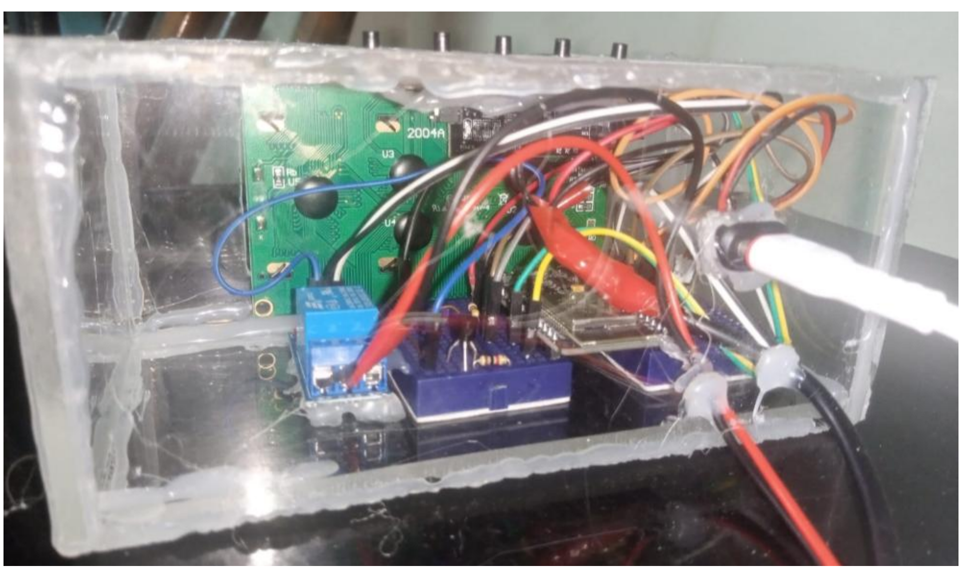


Figure 3.3: ESP-32 Connections

3.5.2 The Relay module is controlled by the Esp-32. So, the positive terminal of the compressor is connected through the relay. The microcontroller send signals to turn the compressor on or off and the relay does the work. The compressor is powered by a 220V connection.

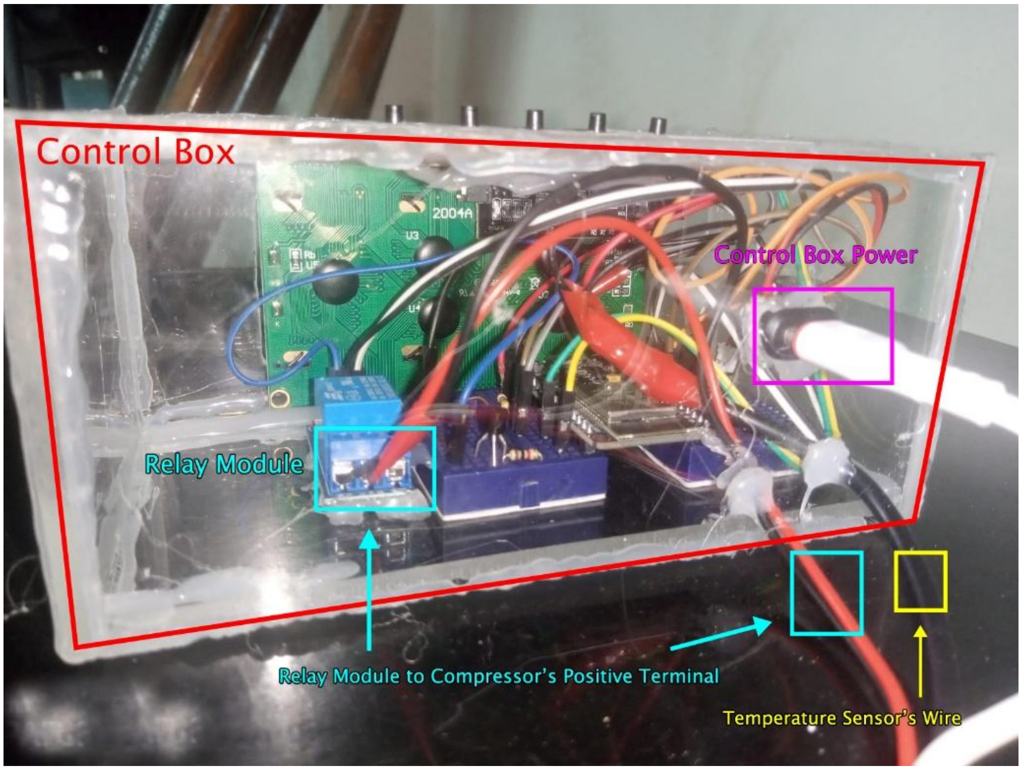


Figure 3.4: Relay and Sensor Connections

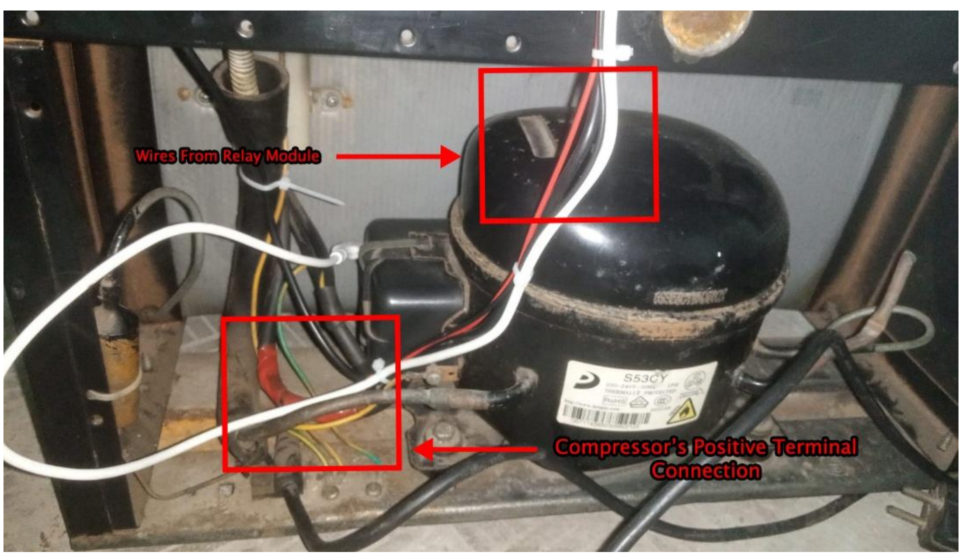


Figure 3.5: Relay and Compressor Connections

3.5.3 The DS18B20 Temperature sensor is connected to the Esp-32 by breadboard. The sensor is placed inside the fridge's cooling chamber.

Figure 3.6: Temperature Sensor (Inside Cooling chamber)

3.6 Cost Analysis

3.6.1 Project cost Analysis

No.	Product Name	Quantity	Unit Price	Total Price
1	Mini Refrigerator	1	12,650	12,650
2	Esp-32 Dev Board	1	880	880
3	5V Relay Module	1	95	95
4	20x4 LCD Display	1	500	500
5	I2C Module	1	150	150
6	Panel Mount Usb-C	1	70	70
7	Usb Power Adapter 5V	1	450	450
8	Waterproof DS18B20 Temperature Sensor	1	350	350
9	Bread Board	2	50	100
10	Push Buttons	5	5	25
11	Miscellaneous		2,000	2,000
Total				17,270

Table 1: Project Cost Analysis

CHAPTER 4
HARDWARE ANALYSIS

4.1 Components List

4.1.1 Mini Refrigerator
4.1.2 Esp-32 Dev Board
4.1.3 DS18B20 Temperature Sensor
4.1.4 20x4 LCD Display
4.1.5 I2C LCD Adapter Module
4.1.6 5V Relay Module
4.1.7 USB Power Adapter 5V

4.2 Mini Refrigerator

The mini refrigerator serves as the physical platform for implementing the proposed smart system. It operates as a conventional refrigeration unit with a built-in cooling mechanism, which is externally monitored and controlled using electronic components. The compact size of the mini refrigerator makes it suitable for experimental and prototype development.

The smart system is integrated without modifying the internal refrigeration mechanism, ensuring safety and practicality. Temperature sensing and control actions are applied externally, making the design adaptable to existing refrigerator units.

Figure 4.1: Mini Refrigerator

4.3 ESP32 Development Board

The ESP32 development board acts as the central processing unit of the proposed system. It is responsible for reading sensor data, executing control logic, and managing communication and display functions. The ESP32 is selected due to its high processing capability, low power consumption, and built-in wireless features.

Its multiple GPIO pins allow easy interfacing with sensors, display modules, and control devices. The ESP32 enables real-time data processing and ensures reliable system performance, making it suitable for smart appliance applications.

Figure 4.2: ESP-32 DEV Board

4.3.1 ESP-32 Dev Board Pinout Configuration:

ESP32 Pin	Pin Type	Connected Component	Function Description
VIN / 5V	Power Input	USB Power Adapter	Supplies 5V power to the ESP32
GND	Ground	All modules	Common ground reference
GPIO 4	Digital I/O	DS18B20 Temperature Sensor	Temperature data input (One-Wire)
GPIO 21	I2C (SDA)	I2C LCD Adapter	Serial data line for LCD
GPIO 22	I2C (SCL)	I2C LCD Adapter	Serial clock line for LCD
GPIO 23	Digital Output	5V Relay Module	Controls Refrigerator Compressor
GPIO 18	Digital Input	Push Button	Power Off
GPIO 19	Digital Input	Push Button	Power On
GPIO 27	Digital Input	Push Button	Set Temp
GPIO 26	Digital Input	Push Button	Temp Down
GPIO 25	Digital Input	Push Button	Temp Up

Table 2: Esp-32 Pinout Configuration

4.4 DS18B20 Temperature Sensor

The DS18B20 is a digital temperature sensor used to measure the internal temperature of the refrigerator. It provides accurate temperature readings and communicates with the microcontroller using a single-wire protocol, reducing wiring complexity.

The sensor is suitable for refrigeration environments due to its wide operating temperature range and stable performance. It only runs on 3.3V and perfect for integrating with esp-32 microcontroller. The Continuous temperature monitoring using this sensor allows the system to detect abnormal conditions and take appropriate control actions.

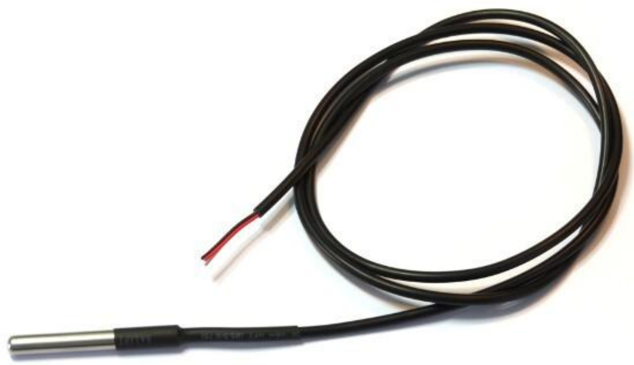


Figure 4.3: DS18B20 Temperature Sensor

4.5 20×4 LCD Display

The 20×4 LCD display is used to present real-time system information to the user. It can display multiple parameters such as current temperature, system status, and alert messages simultaneously. It works with 5V input voltage.

This display enhances user interaction by providing clear visual feedback without requiring external devices. Its character-based format makes it simple to control and suitable for embedded applications.

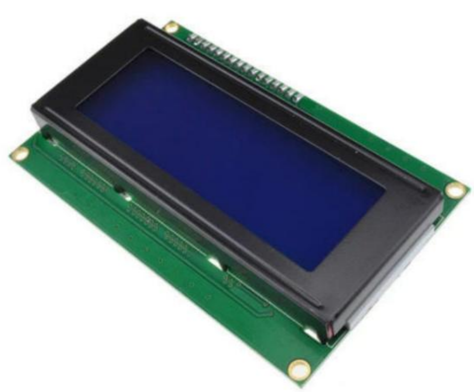


Figure 4.4: 20x4 LCD Display

4.6 I2C LCD Adapter Module

The I2C LCD adapter module is used to interface the LCD display with the ESP32 using the I2C communication protocol. This module significantly reduces the number of required connection pins, improving circuit simplicity and reliability.

By using I2C communication, the system minimizes wiring complexity and allows efficient data transfer between the microcontroller and the display. This makes the overall hardware design cleaner and more scalable.

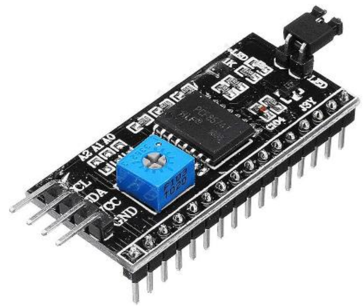


Figure 4.5: I2C LCD Adapter Module

4.7 5V Relay Module

The 5V relay module is used as an electrically controlled switching device in the proposed smart refrigerator system. It allows the low-voltage ESP32 microcontroller to safely control high-power components such as the refrigerator's compressor or external loads.

The relay module acts as an isolation interface between the control circuit and the high-voltage AC system, ensuring user and device safety. When the ESP32 provides a control signal to the relay input, the relay switches between normally open (NO) and normally closed (NC) states, enabling or disabling the connected load.

Figure 4.6: 5V Relay Module

4.8 USB Power Adapter (5V)

The USB power adapter provides a stable 5V power supply to the electronic components of the system. It ensures safe and consistent power delivery to the ESP32, sensors, display, and other modules.

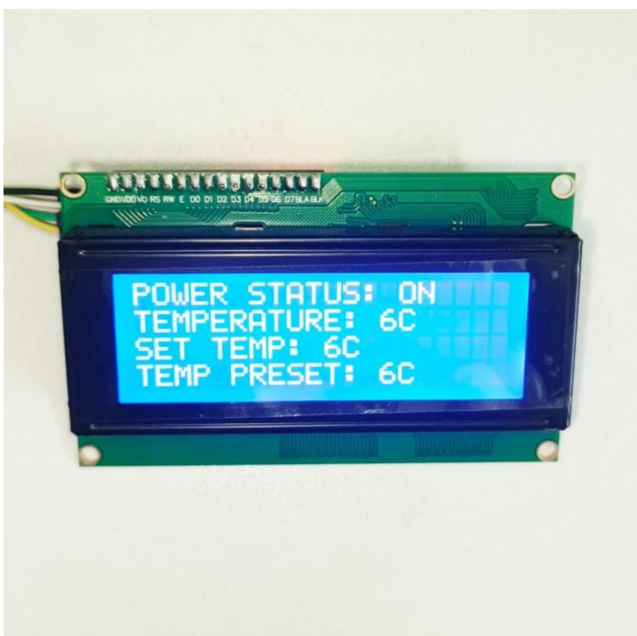
Using a standard 5V USB adapter improves safety and availability while eliminating the need for complex power regulation circuits. This choice contributes to the low-cost and user-friendly design of the system.

Figure 4.7: USB Power Adapter

CHAPTER 5
DISCUSSION

5.1 Introduction

This research successfully demonstrates the design and implementation of a smart refrigerator system using embedded technology. The proposed system integrates temperature sensing, microcontroller-based control, and user display to improve conventional refrigeration functionality. The experimental results indicate that the system can reliably monitor internal temperature and respond to changes based on predefined conditions. The overall performance confirms that monitoring and control can be achieved with low-cost hardware while maintaining system stability and efficiency.



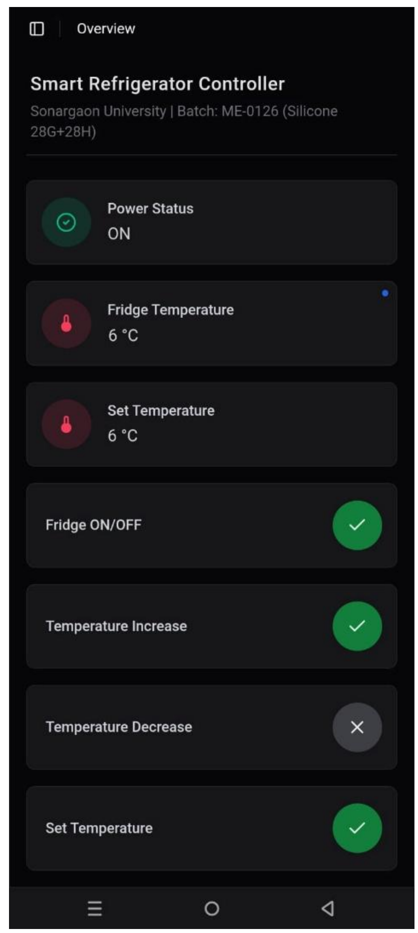


Figure 5.1: Temperature Reading on LCD Display & Web Interface

Here the Temperature is set to 6 °C. So, the compressor kept running until the temperature drops to the 6 °C. When it reached the target, the compressor shuts down and again starts when the temperature difference is 4 °C from the set value.

5.2 Project Details

The final project presents a functional prototype of a smart refrigerator system built on a mini refrigerator platform. The system utilizes an ESP32 development board, a digital temperature sensor, a relay module, and an LCD display to provide real-time temperature monitoring and automated control. The hardware and software components operate together to ensure accurate sensing, timely response, and user-friendly interaction. The project meets its intended objectives which are given bellow:

5.2.1 It drops the temperature as the user's requirement by keeping the compressor on.
5.2.2 It stops the compressor at that temperature and waits until the temperature rises to 4 °C more.
5.3.3 It updates the Live temperature and other parameters in both display and web interface
5.3.4 User can adjust the temperature as needed and also set presets via control box or mobile
5.3.5 User can turn off or on the system via the control box or the web interface

Figure 5.2: Smart Refrigerator

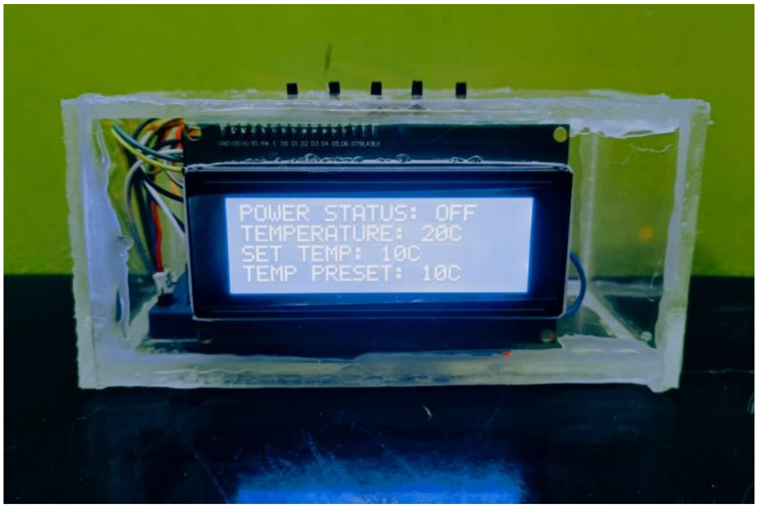


Figure 5.3: Control Box of the Smart Refrigerator

5.3 Data Calculation

5.3.1 Temperature Control Analysis

The system was configured with predefined threshold values to maintain optimal refrigeration conditions:

Threshold Type	Relay Status	Temperature Value (C)
Lower threshold	Relay ON	2°C
Upper threshold	Relay OFF	6°C

Table 3: Temperature analysis

When the temperature rises above 6°C, the ESP32 activates the relay to start the cooling process. When the temperature drops below 2°C, the relay is turned OFF to prevent overcooling. This creates a 4°C hysteresis range. This ensures stable temperature regulation within the desired range.

5.3.2 Cooling and Recovery Time Calculation

The system performance was evaluated based on cooling and warming cycles:

Initial room temperature: **25°C**

Time to reach 2°C: **30 minutes**

Time to rise from 2°C to 6°C (after relay OFF): **90 minutes**

Time to reach 2°C again (relay ON): **17 minutes**

These values indicate that the system maintains a controlled cooling cycle, balancing energy efficiency and temperature stability.

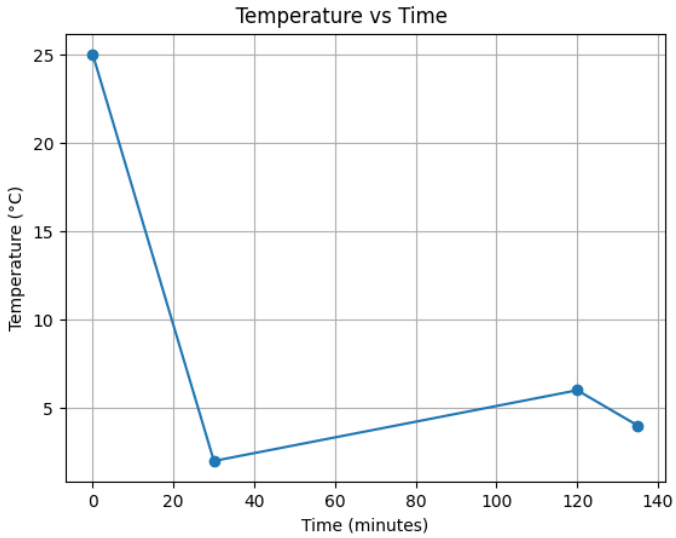


Figure 5.4: Cooling & recovery time calculation

5.4 Advantages

The proposed smart refrigerator system offers several advantages over traditional refrigerators:

5.4.1 Real-time temperature monitoring improves food preservation

5.4.2 Automated control enhances energy efficiency

5.4.3 Simple hardware architecture ensures reliability

5.4.4 Low-cost components make the system affordable

5.4.5 Expandable design allows future feature integration

5.4.6 The precise accuracy of maintaining the desired temperature

5.5 Limitations

Despite its successful implementation, the system has certain limitations:

5.5.1 The system focuses primarily on temperature monitoring

5.5.2 Advanced food recognition features are not included

5.5.3 Manual threshold configuration is required

5.5.4 Internet-based data analytics are not fully implemented

5.5.5 These limitations are mainly due to hardware constraints and project scope and can be addressed in future developments.

5.6 Applications

The proposed smart refrigerator system can be applied in various practical scenarios:

5.6.1 Household refrigerators for improved food storage

5.6.2 Small shops and restaurants for temperature monitoring

5.6.3 Laboratories requiring controlled storage conditions

5.6.4 Educational and research-based embedded system projects

5.6.5 Pharmaceutical Storage Systems management

5.6.6 Its flexibility and simplicity make it suitable for both domestic and commercial use.

5.7 Discussion

Overall, this project demonstrates that smart appliance functionality can be achieved using simple and cost-effective embedded systems. The design approach balances performance, affordability, and expandability. The system provides a strong foundation for future enhancements such as IoT-based monitoring and mobile application integration. The outcomes of this research contribute to the growing field of smart home technologies and highlight the potential of embedded systems in everyday appliances.

CHAPTER 6
CONCLUSION

6.1 Conclusion

This thesis presented the design and implementation of a smart refrigerator system using embedded technology. The proposed system successfully integrates a temperature sensor, ESP32 microcontroller, relay-based control, and LCD display to enhance conventional refrigeration functionality. The system provides real-time temperature monitoring and automated control, improving food preservation and energy efficiency.

The experimental results confirm that the system operates reliably under normal conditions and responds effectively to temperature variations. By using low-cost and readily available components, the project demonstrates that smart appliance solutions can be developed without significant complexity or expense. Overall, the objectives of the research have been achieved, and the proposed system offers a practical foundation for smart refrigeration applications.

6.2 Future Scope

Although the proposed system meets its primary objectives, several enhancements can be considered for future development. Internet of Things (IoT) integration can be implemented to enable remote monitoring through mobile or web applications. Advanced food management features such as inventory tracking and expiration alerts can be added using additional sensors or camera modules.

Machine learning techniques may be applied to analyze usage patterns and optimize energy consumption. Cloud-based data storage and analytics can further improve system intelligence. These future improvements can transform the proposed prototype into a fully autonomous and intelligent smart refrigerator system.

REFERENCES

1. R. Arora and A. Kumar, Refrigeration and Air Conditioning, 3rd ed., PHI Learning, New Delhi, 2015.
2. W. F. Stoecker and J. W. Jones, Refrigeration and Air Conditioning, McGraw-Hill, New York, 1982.
3. Espressif Systems, ESP32 Technical Reference Manual, Espressif Inc., 2023.
4. Maxim Integrated, DS18B20 Programmable Resolution 1-Wire Digital Thermometer Datasheet, Maxim Integrated, 2022.
5. Hitachi Ltd., Refrigeration Systems and Applications, Tata McGraw-Hill, 2010.
6. S. Pal and S. K. Bose, "Design of Smart Refrigerator Using IoT," International Journal of Engineering Research & Technology (IJERT), vol. 9, no. 6, 2020.
7. A.Al-Fuqaha et al., "Internet of Things: A Survey on Enabling Technologies," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.
8. Philips Electronics, Relay Module Application Notes, Philips Semiconductors, 2018.
9. World Health Organization (WHO), Technical Supplement: Temperature Monitoring of Vaccines, WHO Press, 2019.

INDEX
SOURCE CODE

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
#include <Preferences.h>
Preferences preferences;
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 4
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
#include <Arduino.h>
#if defined(ESP8266)
  /* ESP8266 Dependencies */
  #include <ESP8266WiFi.h>
  #include <ESPAsyncTCP.h>
  #include <ESPAsyncWebServer.h>
#elif defined(ESP32)
  /* ESP32 Dependencies */
  #include <WiFi.h>
  #include <AsyncTCP.h>
  #include <ESPAsyncWebServer.h>
#elif defined(TARGET_RP2040) || defined(PICO_RP2040) || defined(TARGET_RP2350) ||
defined(PICO_RP2350)
  /* RP2040 or RP2350 Dependencies */
  #include <WiFi.h>
  #include <RPAsyncTCP.h>
  #include <ESPAsyncWebServer.h>
#endif
#include <ESPDash.h>
/* Your SoftAP WiFi Credentials */
const char* ssid = "Smart Fridge"; // SSID
const char* password = "12345678"; // Password
/* Start Webserver */
AsyncWebServer server(80);
/* Attach ESP-DASH to AsyncWebServer */
ESPDash dashboard(server);
```

```
dash::SeparatorCard separator(dashboard, "Smart Refrigerator Controller", "Sonargaon
University | Batch: ME-0126 (Silicone 28G+28H)");
dash::FeedbackCard<const char*> feedback(dashboard, "Power Status");
dash::TemperatureCard<int> tempCard(dashboard, "Fridge Temperature", "C");
dash::TemperatureCard<int> setCard(dashboard, "Set Temperature", "C");
dash::ToggleButtonCard toggle(dashboard, "Fridge ON/OFF");
dash::ToggleButtonCard tempin(dashboard, "Temperature Increase");
dash::ToggleButtonCard tempde(dashboard, "Temperature Decrease");
dash::ToggleButtonCard tempset(dashboard, "Set Temperature");
 //web er code sesh
bool systemON = true;
bool oner = true;
bool offer = true;
const int off_pin = 18;
const int on_pin = 19;
////const int mode_pin = 18;
const int up_pin = 25;
const int down_pin = 26;
const int set_pin = 27;
const int relay_pin = 23;
int setvalue ;  //= 0
unsigned long previousMillis = 0;   // Stores the last time the sensor was read (modified...........)
const long interval = 1000;         // Interval at which to read (1 second)
bool secondfunction = false;
bool conversionStarted = false;
 static unsigned long lastDashUpdate = 0;
void setup()
{
  // Serial.begin(115200);
    sensors.begin();
     Wire.begin(21, 22);
     lcd.init();
  lcd.backlight();
    //web code shuru
      /* Start Access Point */
  WiFi.mode(WIFI_AP);
  WiFi.softAPConfig(IPAddress(192, 168, 4, 1), IPAddress(192, 168, 4, 1), IPAddress(255, 255,
255, 0));
  WiFi.softAP(ssid, password);
  Serial.print("IP Address: ");
```

```
 Serial.println(WiFi.softAPIP());
 /* Start AsyncWebServer */
 server.begin();
 sensors.begin();
 //extra code start
 tempCard.setUnit("°C");
 setCard.setUnit("°C");
 //toggle button
 feedback.setFeedback("ON", dash::Status::SUCCESS);
toggle.setValue(true);

  toggle.onChange([&](bool state){

   toggle.setValue(state);
if(state){feedback.setFeedback("ON", dash::Status::SUCCESS);
 if(oner) { systemON = true;
 oner = false ;
 offer = true;
 dashboard.sendUpdates();
}
// on button high, delay, button low
}
 else{feedback.setFeedback("OFF", dash::Status::DANGER);
if(offer) { systemON = false;
 oner = true ;
 offer = false;
dashboard.sendUpdates();
 }
 // off button high, delay, button low
 }
  });
    tempin.setValue(true);
    tempde.setValue(false);
    tempset.setValue(true);


    tempin.onChange([&](bool instate){

     if(!instate) {if(setvalue < 25 ) {setvalue++;}
     dashboard.sendUpdates();
```

```
    tempin.setValue(true);

    lcd.setCursor(0, 2);
lcd.print("SET TEMP: ");
lcd.print(setvalue);
lcd.print("C  ");
     }

  });

  tempde.onChange([&](bool destate){

   if(destate) {if(setvalue > -11 ) {setvalue--;}
   dashboard.sendUpdates();
    tempde.setValue(false);

    lcd.setCursor(0, 2);
lcd.print("SET TEMP: ");
lcd.print(setvalue);
lcd.print("C  ");
    }

  });

   tempset.onChange([&](bool setstate){

   if(!setstate) {
  preferences.begin("my-app", false);
  preferences.putInt("counter", setvalue);
  preferences.end();


   dashboard.sendUpdates();
    tempin.setValue(true);

    lcd.setCursor(0, 3);
lcd.print("TEMP PRESET: ");
lcd.print(setvalue);
lcd.print("C  ");
```

```
      }

    });

    //web code sesh

  pinMode(off_pin, INPUT_PULLUP);
  pinMode(on_pin, INPUT_PULLUP);
 // pinMode(mode_pin, INPUT_PULLUP);
  pinMode(up_pin, INPUT_PULLUP);
  pinMode(down_pin, INPUT_PULLUP);
  pinMode(set_pin, INPUT_PULLUP);
  pinMode(relay_pin, OUTPUT);

  preferences.begin("my-app", false);
   setvalue = preferences.getInt("counter", 0);
  preferences.end();

      lcd.setCursor(0, 0);
  lcd.print("POWER STATUS: ON ");

  lcd.setCursor(0, 1);
  lcd.print ("Loading...");

   lcd.setCursor(0, 2);
  lcd.print("SET TEMP: ");
  lcd.print(setvalue);
  lcd.print("C  ");

  lcd.setCursor(0, 3);
  lcd.print("TEMP PRESET: ");
  lcd.print(setvalue);
  lcd.print("C  ");

}

void loop() {
```

```
unsigned long currentMillis = millis();  // modified
static int tempC = 0;
if (!conversionStarted) {

sensors.requestTemperatures();


previousMillis = currentMillis;
  conversionStarted = true;
}

if ( !secondfunction && (currentMillis - previousMillis >= 800))

{

 tempC = sensors.getTempCByIndex(0);
 //Serial.print("Temperature: ");
 //Serial.print(tempC);
 //Serial.println("°C");
 //previousMillis = currentMillis;
if(tempC < -100)

{
  lcd.setCursor(0, 1);
lcd.print("SENSOR ERROR!        ");

}

if(tempC > -100){
 lcd.setCursor(0, 1);
lcd.print("TEMPERATURE: ");
lcd.print (tempC);
lcd.print ("C  "); }

if (!systemON)
{
  lcd.setCursor(0, 0);
lcd.print("POWER STATUS: OFF");}

if (systemON)
```

```
  {
   lcd.setCursor(0, 0);
 lcd.print("POWER STATUS: ON  ");}

  secondfunction = true;

 }

 if (currentMillis - previousMillis >= 1000) {


 conversionStarted = false;
 secondfunction = false;

 }

//ekhane copied code shesh

if (currentMillis - lastDashUpdate >= 1000) {

 lastDashUpdate = currentMillis;
 tempCard.setValue((float)tempC);
 setCard.setValue(setvalue);
  dashboard.sendUpdates();

}
 if (digitalRead(up_pin) == LOW)
 {
  if(setvalue < 25 ) {setvalue++;}
   delay(400);

  lcd.setCursor(0, 2);
 lcd.print("SET TEMP: ");
 lcd.print(setvalue);
 lcd.print("C  ");

 }

 if (digitalRead(down_pin) == LOW)
 {
```

```
  if(setvalue > -11 ) {setvalue--;}
  delay(400);

  lcd.setCursor(0, 2);
lcd.print("SET TEMP: ");
lcd.print(setvalue);
lcd.print("C  ");
}
if (digitalRead(on_pin) == LOW)
{ if(oner) {
  feedback.setFeedback("ON", dash::Status::SUCCESS);
  systemON = true;
oner = false ;
offer = true;
}
 toggle.setValue(true);
  dashboard.sendUpdates();


 }
if (digitalRead(off_pin) == LOW)
{
  if(offer) { systemON = false;
oner = true ;
offer = false;
}
     toggle.setValue(false);
 feedback.setFeedback("OFF", dash::Status::DANGER);
   digitalWrite (relay_pin, LOW);
    dashboard.sendUpdates();
}

if (systemON)
 {
  if ( tempC >= setvalue + 4 )
  {digitalWrite (relay_pin, HIGH); }
  if ( tempC <= setvalue  )
  {digitalWrite (relay_pin, LOW); }
 }
```

```
if (!systemON)
{
  digitalWrite (relay_pin, LOW);
 toggle.setValue(false);
 }

 if(digitalRead(set_pin) == LOW )
 {
  preferences.begin("my-app", false);
  preferences.putInt("counter", setvalue);
  preferences.end();
  delay(400);

  lcd.setCursor(0, 3);
lcd.print("TEMP PRESET: ");
lcd.print(setvalue);
lcd.print("C  ");
 }

 }
```