IOT-ENABLED TEMPERATURE-CONTROLLED FAN SYSTEM

MINI PROJECT REPORT

Submitted to the Department of Computer Applications, Bharathiar University in partial fulfillment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Submitted by

VIKTAMATHITHAN M (23CSEA76)

Under the guidance of

Dr. J. SATHEESHKUMAR, MCA., Ph.D.,
Professor,
Department of Computer Applications

DEPARTMENT OF COMPUTER APPLICATIONS

BHARATHIAR UNIVERSITY

COIMBATORE – 641 046

DECEMBER – 2024

DECLARATION

I hereby affirm that the project entitled **"IOT-ENABLED TEMPERATURE-CONTROLLED FAN,"** which is being submitted to the Department of Computer Applications, Bharathiar University, Coimbatore, represents the original work conducted by **VIKRAMATHITHAN M (23CSEA76)**. This work was carried out under the supervision and guidance of **Dr. J SATHEESHKUMAR, MCA., Ph.D.,** from the Department of Computer Applications at Bharathiar University, Coimbatore. Furthermore, I confirm that this project has not been used as the basis for the conferral of a Degree, Diploma, Associate Ship, Fellowship, or any similar title to any candidate from any other university prior to this submission.

Place: Coimbatore Signature of Candidate

Date: **(VIKRAMATHITHAN M)**

Countersigned by

Dr. J. SATHEESHKUMAR, MCA., Ph.D.,

Professor,

Department of Computer Applications

CERTIFICATE

This is to certify that the project work titled **"IOT-ENABLED TEMPERATURE-CONTROLLED FAN,"** submitted to the Department of Computer Applications, Bharathiar University, in partial fulfillment of the requirements for the award of a Degree in Master of Computer Applications, is a record of the original work done by **VIKRAMATHITHAN M (23CSEA76)**. under my supervision and guidance. I confirm that this project work has not formed the basis of the award of any Degree/Diploma/Associate Ship/Fellowship or similar title to any candidate of any university.

Place: Coimbatore
Date:

Project Guide **Head of the Department**

Submitted for the University Viva-Voce Examination held on …………………

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

I thank the Almighty God who showered His grace to make this project successful. The success and final outcome of this study required a lot of guidance and assistance from many people and I am extremely fortunate to have got those people all along the work and completion of the project .

I express my sincere gratitude to our Professor & Head of the Department **Dr. M PUNITHAVALLI, MCA., Ph.D.,** Department of Computer Applications, Bharathiar University, Coimbatore for giving me the opportunity to do this project work .

I really deem it a special privilege to professor **Dr. J. SATHEESHKUMAR, MCA., Ph.D.,,** Department of Computer Application, Bharathiar University for valuable guidance and interest in my project work from the beginning to the end.My heartfelt thanks to my dear parents, family, and friends for their support and encouragement toward the successful completion of this project. I am highly obliged to all those who have helped me directly and indirectly in making this project a success .

TABLE OF CONTENT

ABSTRACT

The Temperature-Controlled Fan System is a microcontroller-based project designed to automate fan control using temperature feedback. This system incorporates a DS18B20 temperature sensor to monitor ambient temperature accurately and a relay module to control the fan operation. A potentiometer allows users to set a customizable temperature threshold, enabling the system to adjust its behaviour dynamically based on environmental conditions.

The microcontroller processes temperature readings from the sensor and compares them to the user-defined threshold. When the ambient temperature exceeds the threshold, the fan is activated via the relay module; otherwise, it remains off. An LCD display provides real-time feedback, showing the current temperature and the threshold set by the user. Additionally, the system outputs debugging information via a serial monitor for enhanced troubleshooting and development purposes.

This project offers a cost-effective, user-friendly solution for automating cooling systems in residential, industrial, or data center environments. By reducing energy consumption and optimizing performance, the Temperature-Controlled Fan System represents a step toward smarter and more efficient environmental control solutions.

CHAPTER – 1

INTRODUCTION

The Temperature-Controlled Fan System is an innovative approach to automating the operation of fans based on real-time environmental temperature readings. In today’s world, the demand for energy-efficient and automated systems is growing rapidly, especially in domains such as smart homes, industrial automation, and data centers. This project addresses the need for efficient cooling by integrating a temperature sensor, a relay-controlled fan, and user-configurable threshold settings.

The system utilizes a DS18B20 temperature sensor to accurately monitor ambient temperature, an Arduino microcontroller for processing, and a relay module to control the fan. The user can define the temperature threshold using a potentiometer, allowing flexibility in fan operation. A LiquidCrystal I2C-based LCD provides a convenient interface for real-time monitoring of the current temperature and the set threshold.

This project aims to simplify temperature management while optimizing energy usage. By automating fan operation, it eliminates the need for manual intervention, offering a practical and scalable solution for temperature regulation. The simplicity of its design, combined with its cost-effectiveness, makes this system ideal for small-scale and mid-scale applications requiring automated cooling.

CHAPTER – 2

SYSTEM ANALYSIS

2.1 IMPLEMENTED SYSTEM

2.1.1 PROPOSED SYSTEM

The proposed system aims to automate the operation of a fan based on ambient temperature. Using a DS18B20 temperature sensor, the system monitors environmental conditions and adjusts the fan operation through a relay module. A potentiometer allows users to define a threshold temperature, and an LCD provides real-time feedback. This system ensures energy-efficient cooling without manual intervention.

2.1.2 FEATURES

The Temperature-Controlled Fan System offers several key features designed to enhance automation, efficiency, and user convenience.

1. **Real-Time Temperature Monitoring:**

   The system continuously measures the ambient temperature using the DS18B20 temperature sensor. This ensures accurate and up-to-date information about the environmental conditions.

2. **User-Defined Threshold Adjustment:**

   A potentiometer allows users to easily set the desired temperature threshold. Once the threshold is exceeded, the fan is activated, ensuring that cooling is provided when needed.

3. **Automatic Fan Control:**

   Based on the real-time temperature and the user-defined threshold, the system automatically adjusts the fan's operation. The fan is turned on when the temperature exceeds the threshold and turned off when it is below the threshold.

4. **LCD Display for Monitoring:**

The LiquidCrystal I2C display provides real-time feedback, showing the current temperature and the set threshold. This allows users to monitor the system's status at a glance.

5. **Energy Efficiency:**

By automating fan control based on temperature, the system ensures that the fan operates only when necessary, optimizing energy consumption.

6. **Simple and Scalable Design:**

The system's design is simple, with only a few components required. This makes it easy to scale the system for different applications, from home environments to industrial settings.

7. **Real-Time Debugging and Monitoring:**

Serial monitoring allows for easy debugging and verification of the system's performance, providing insights into the fan's operation and the temperature readings.

8. **Cost-Effective:**

The components used in the system are widely available and affordable, making the solution cost-effective for both personal and commercial use.

2.1.3 ADVANTAGES

The Temperature-Controlled Fan System offers several advantages that contribute to its effectiveness and practicality:

1. **Automation and Convenience:**

The system automates the fan's operation based on temperature, eliminating the need for manual intervention. This provides users with hands-free control, improving convenience.

2. **Energy Efficiency:**

The fan operates only when the temperature exceeds the user-defined threshold, preventing unnecessary energy consumption. This results in lower energy costs and a more eco-friendly solution.

3. **Improved Cooling Performance:**

By using temperature data to control fan speed, the system ensures that the cooling mechanism is always operating at the right time, optimizing cooling efficiency.

4. **Cost-Effective:**

The components used, such as the DS18B20 sensor, Arduino microcontroller, and relay, are affordable and widely available. This makes the system a low-cost solution for automated fan control.

5. **Scalability and Flexibility:**

The system is easily scalable and can be adapted for various applications, including homes, offices, industrial settings, and data centers. It can also be customized to control multiple fans or devices.

6. **Real-Time Monitoring:**

The system provides real-time feedback on temperature and fan status via the LCD display, allowing users to monitor the system’s performance instantly. Additionally, the serial monitor can be used for debugging and performance analysis.

7. **Simple Setup and Operation:**

The system is simple to set up and operate, making it accessible even to those with minimal technical experience. The user-friendly interface and clear display make it easy to adjust settings and monitor results.

8. **Reliable Performance:**

The use of the DS18B20 sensor ensures accurate temperature readings, providing reliable performance. The system maintains consistent fan operation based on real-time conditions.

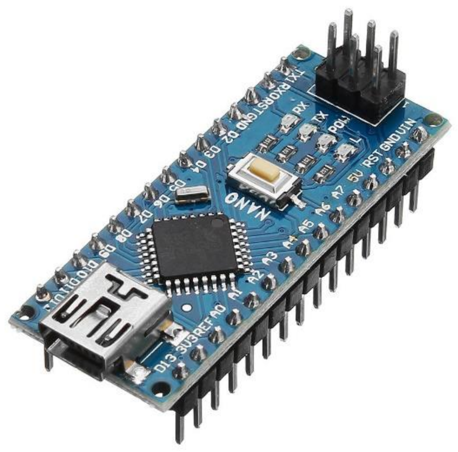
9. **Compact and Space-Efficient:**

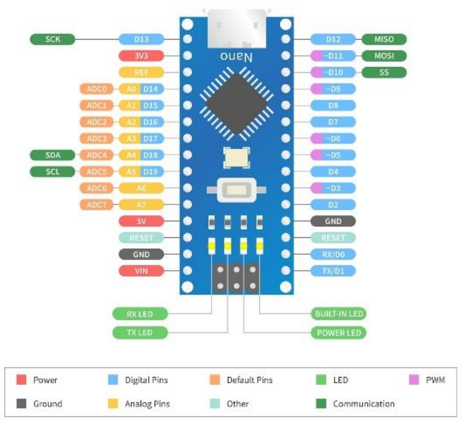
The system uses compact components, making it easy to integrate into existing setups without requiring significant space or infrastructure changes.

2.2 SYSTEM SPECIFICATION

2.2.1 HARDWARE REQUIREMENTS

1. **Microcontroller**





Model: Arduino Nano

Processor: ATmega328P

Operating Voltage: 5V

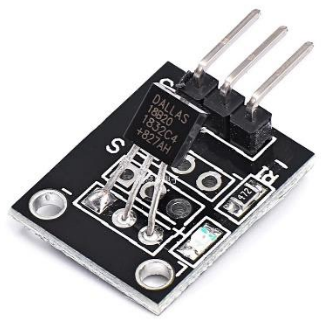
Input Voltage: 7-12V

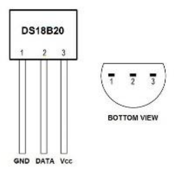
Digital I/O Pins: 14 (6 PWM outputs)

Analog Input Pins: 8

Purpose: To control the connected sensors, relays, and display.

2. Temperature Sensor





Model: DS18B20

Operating Voltage: 3V - 5.5V

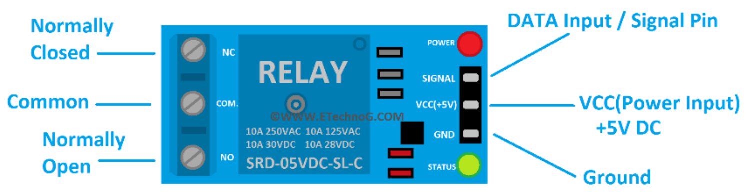
Temperature Range: -55°C to +125°C

Accuracy: ±0.5°C from -10°C to +85°C

Communication Protocol: OneWire

Purpose: To measure and provide real-time temperature data.

3. Relay Module



Model: Single-channel 5V Relay Module

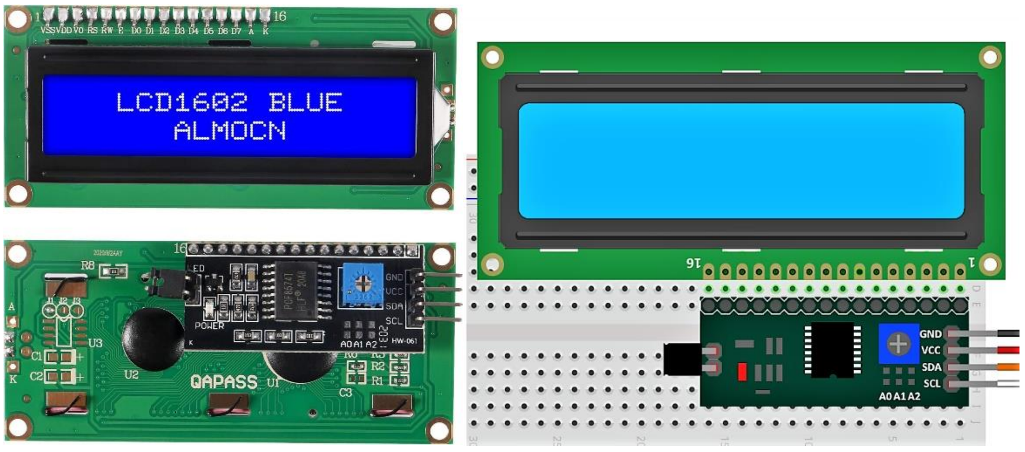
Operating Voltage: 5V DC

Relay Rating: 10A @ 250VAC or 10A @ 30VDC

Trigger Voltage: Low-level trigger (0V to 2.5V)

Purpose: To control high-voltage appliances such as a fan or air conditioner.

4. **LCD Display**



Model: 16x2 LCD with I2C Adapter

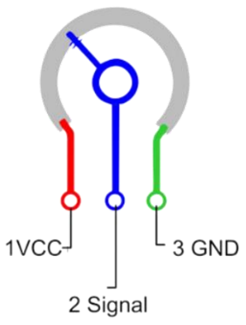
Operating Voltage: 5V DC

I2C Address: 0x27 or 0x3F (depending on the module)

Display: 2 rows, 16 characters per row

Purpose: To display temperature readings and threshold values.

5. **Potentiometer**



Resistance Rating: 10 kΩ

Type: Rotary potentiometer

Purpose: To set the desired temperature threshold for controlling the relay.

6. **Power Supply**

Type: DC Adapter or USB power supply

Voltage: 5V (for Arduino and peripherals)

Purpose: To power the microcontroller and other connected components.

7. **Fan/Appliance**

Type: AC Fan or similar load

Operating Voltage: 220V AC (or equivalent)

Purpose: To demonstrate relay-controlled temperature-based operation.

8. **Connecting Wires**

    Type: Male-to-male, male-to-female jumper wires

    Material: Copper or tinned copper for low resistance

    Purpose: To connect components in the circuit.

9. **Breadboard or PCB**

    Type: Standard solderless breadboard or pre-drilled PCB

    Purpose: For prototyping the circuit connections.

10. **USB Cable**

    Type: Mini-B or Micro-B (depending on Arduino Nano model)

    Purpose: For programming the microcontroller and initial power supply.

2.2.2 Software Requirements

1. **Arduino IDE**

    Used to write, edit, and upload the code to the Arduino board.

Includes tools for debugging and testing the hardware components.

2. **Libraries**

    OneWire: For communication with the DS18B20 temperature sensor.

    DallasTemperature: To process temperature readings from the sensor.

LiquidCrystal_I2C: To control the LCD display.

3. **Serial Monitor**

    Integrated with the Arduino IDE.

    Used to monitor temperature readings and relay status.

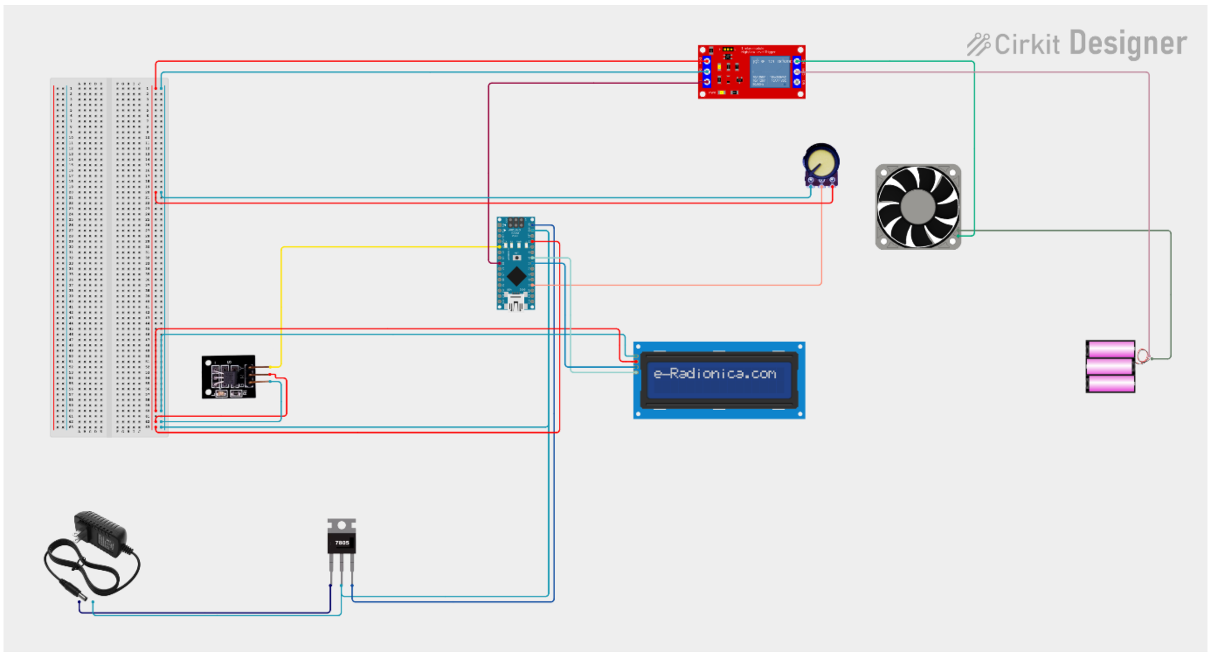
4. **Operating System**

    Compatible with Windows, macOS, or Linux for running the Arduino IDE.

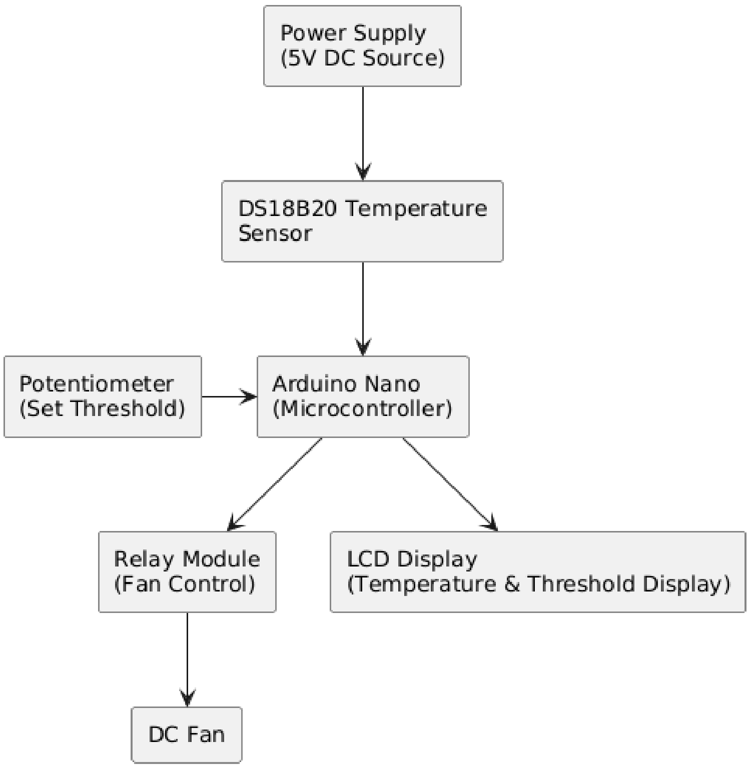
CHAPTER – 3

3.1 METHODOLOGY

3.1.1 CIRCUIT DIAGRAM



3.1.2 BLOCK DIAGRAM



3.2 Working Principles

1. **Powering the System:**

   The system is powered by a 5V DC source, which supplies power to the Arduino, DS18B20 temperature sensor, relay module, and LCD display.

2. **Temperature Sensing:**

   The DS18B20 temperature sensor continuously measures the ambient temperature.

   This temperature value is sent to the Arduino via the OneWire protocol.

3. **User-Defined Threshold:**

   A potentiometer is used to set the desired threshold temperature.

   The Arduino reads the analog value from the potentiometer and maps it to a temperature range (e.g., 20°C to 40°C).

4. **Temperature Comparison:**

   The Arduino compares the measured temperature from the DS18B20 with the user-defined threshold.

   If the temperature exceeds the threshold, the fan is activated by turning on the relay.

   If the temperature falls below the threshold, the relay is turned off, stopping the fan.

5. **Relay Operation:**

   The relay module acts as a switch to control the fan.

   When the relay receives a HIGH signal from the Arduino, it completes the circuit to power the fan.

   When the relay receives a LOW signal, the circuit is broken, and the fan is turned off.

6. **Feedback Display:**

An **LCD display shows:**

The real-time ambient temperature.

The current threshold temperature set by the user.

This provides the user with clear feedback about the system's status.

CHAPTER – 4

IMPLEMENTATION

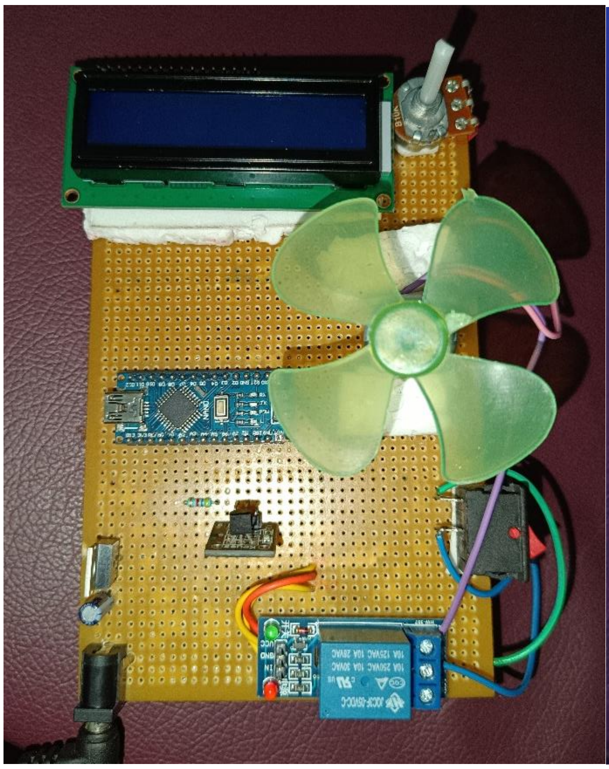
4.1 SOURCE CODE

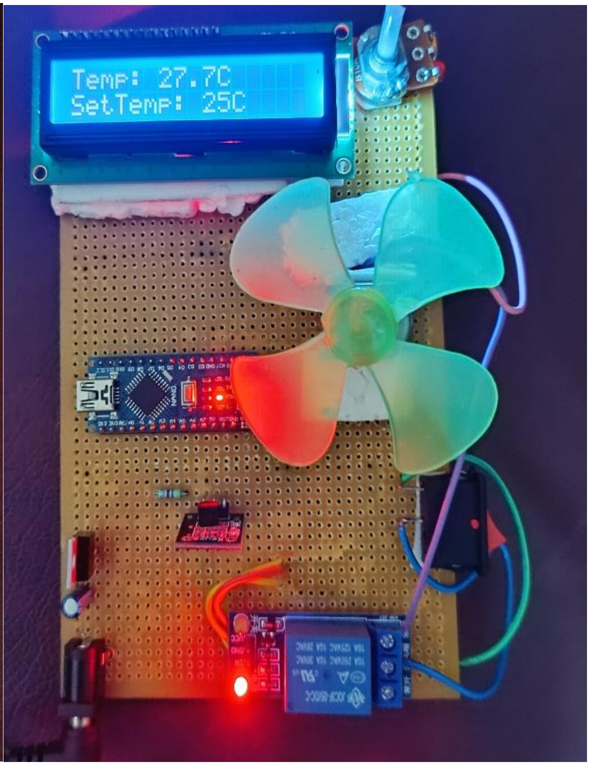
```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define ONE_WIRE_BUS 2
const int potPin = A0;
const int RelayPin = 5;

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  sensors.begin();
  pinMode(RelayPin, OUTPUT);
  digitalWrite(RelayPin, LOW);
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Temp Fan Control");
  lcd.setCursor(0, 1);
  lcd.print("Initializing...");
  delay(2000);
  lcd.clear();
}
```

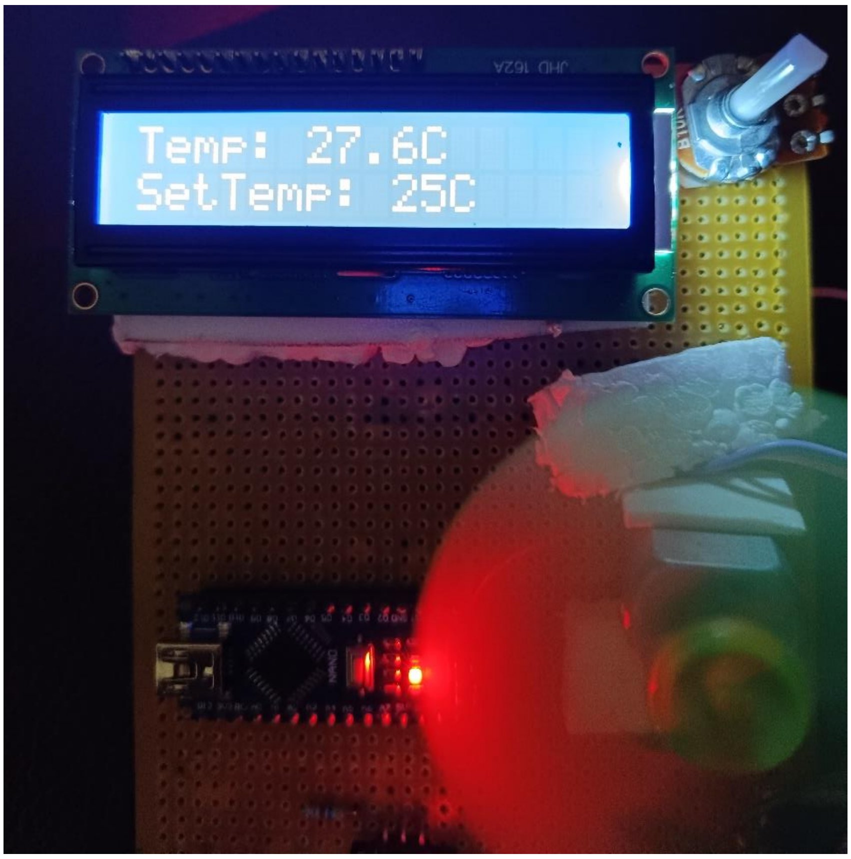
```
void loop() {
  int threshold = map(analogRead(potPin), 0, 1023, 20, 40);
  sensors.requestTemperatures();
  float temperature = sensors.getTempCByIndex(0);

  Serial.print("Temperature: ");
  Serial.println(temperature);
  Serial.print("Threshold: ");
  Serial.println(threshold);

  if (temperature == DEVICE_DISCONNECTED_C) {
    lcd.setCursor(0, 0);
    lcd.print("Sensor Error");
    digitalWrite(RelayPin, LOW);
    delay(1000);
    return;
  }

  if (temperature > threshold) {
    digitalWrite(RelayPin, HIGH);
  } else {
    digitalWrite(RelayPin, LOW);
  }

  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temperature, 1);
  lcd.print("C");
  lcd.setCursor(0, 1);
  lcd.print("SetTemp: ");
  lcd.print(threshold);
  lcd.print("C");
  delay(1000);
}
```

CHAPTER – 5

EXPERIMENTAL RESULT







CHAPTER – 6

CONCLUSION

In this project, we successfully designed and implemented a temperature-based fan control system using an ESP32 (or any compatible microcontroller), a DS18B20 temperature sensor, a relay, a potentiometer for adjusting the temperature threshold, and an LCD display for real-time feedback.

The system continuously monitors the temperature via the DS18B20 sensor. Based on the temperature readings, the relay controls the fan's operation. The temperature threshold can be adjusted using the potentiometer, allowing the system to turn the fan on or off according to the desired temperature range. The LCD provides a clear display of the current temperature and the set threshold, offering the user valuable real-time information about the system's status.

Key accomplishments of the project:

1. **Temperature Sensing:** Accurate temperature readings from the DS18B20 sensor are displayed on the LCD.
2. **Fan Control:** The relay switches the fan on/off based on the temperature threshold set by the potentiometer.
3. **User Interface:** The LCD provides feedback on both the current temperature and the set threshold, ensuring ease of use and monitoring.

This system can be used in various applications such as automated cooling systems, smart home projects, or any scenario where temperature control is essential. By adjusting the threshold, the system becomes adaptable to different cooling requirements.

FUTURE WORK

In the future, the temperature-based fan control system can be expanded to offer more advanced features and capabilities. One possible improvement is integrating Wi-Fi functionality, allowing the system to be controlled remotely through a mobile app or web interface using a device like the ESP32. Additionally, support for controlling multiple fans, each with individual temperature thresholds, can be added, providing more flexibility for larger environments. Data logging features could be incorporated to monitor temperature trends and optimize fan performance over time. To further enhance control, advanced algorithms such as Pulse Width Modulation (PWM) could be implemented, allowing for smoother fan speed regulation based on temperature. Finally, integrating voice control with platforms like Amazon Alexa or Google Assistant would enable hands-free operation, making the system even more user-friendly and accessible. These upgrades would significantly increase the versatility and intelligence of the system, making it suitable for a wider range of applications.

REFERENCES :

OneWire Library: Used for communication with the DS18B20 temperature sensor.
Available at: https://www.pjrc.com/teensy/td_libs_OneWire.html

DallasTemperature Library: A library designed to interface with Dallas Semiconductor's DS18B20 and other temperature sensors.
Available at: https://github.com/milesburton/Arduino-Temperature-Control-Library

Wire Library: A standard Arduino library used for I2C communication, allowing the interface with the LCD display.
Available at: https://www.arduino.cc/en/Reference/Wire

LiquidCrystal_I2C Library: Used for controlling I2C-based LCD displays, making it easy to display text on the LCD screen.
Available at: https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library

Arduino IDE: The platform used for writing and uploading code to the Arduino board.
Available at: https://www.arduino.cc/en/software