MICROCONTROLLER BASED AUTOMATIC TEMPERATURE CONTROL FOR POULTRY ENVIRONMENT

A MINI PROJECT REPORT

Submitted by

ASHIK SM

ALBIN

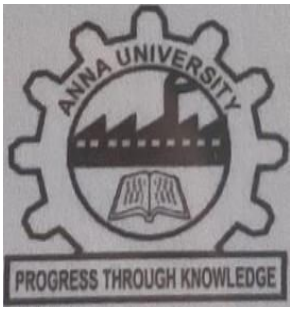
NIJU PARS

In partial fulfilment for the Award of the Degree

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING



MAR EPHRAEM COLLEGE OF ENGINEERING AND TECHNOLOGY

ANNA UNIVERSITY : CHENNAI 600 025

MAR EPHRAEM COLLEGE OF ENGINEERING AND TECHNOLOGY

BONAFIDE CERTIFICATE

Certified that this Mini Project **"MICROCONTROLLER-BASED AUTOMATIC TEMPERATURE CONTROL FOR POULTRY ENVIRONMENT"** is bonafide work of "**ASHIK SM (961422106014),NIJU PARS (961422106301), ALBIN (961422106011)** " who carried out the project work under my supervision.

University Examination held on _____

Internal Examiner **External Examiner**

ABSTRACT :

Temperature regulation is one of the most critical factors in poultry farming, directly influencing the health, growth rate, and productivity of the birds. Inadequate temperature control can lead to stress, disease, poor weight gain, and even mortality. Therefore, ensuring a stable and optimal environmental condition is essential for efficient poultry production. However, traditional methods of monitoring and controlling temperature in poultry farms are often manual, labor-intensive, and lack real-time adaptability.

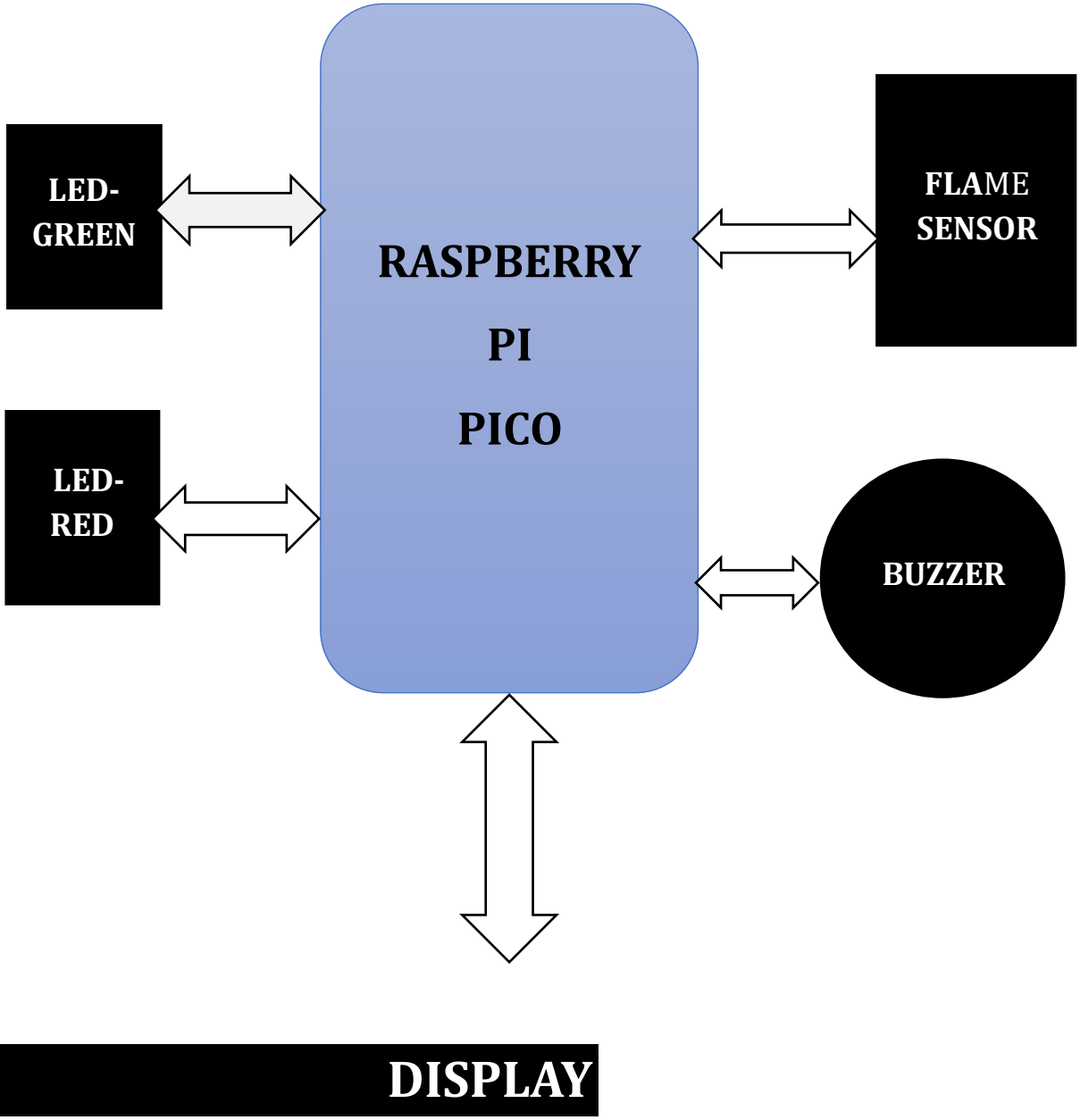
This project presents a cost-effective and intelligent solution through the design and development of a **Microcontroller-Based Automatic Temperature Control System for Poultry Environment** using a **Raspberry Pi Pico**. The system continuously monitors the ambient temperature using a **BMP280 sensor**, which provides accurate readings of both temperature and pressure. Based on these readings, the system automatically controls a **fan** and an **LED bulb** via a **1-channel relay module** to maintain optimal temperature levels.

The core logic is implemented using **MicroPython**, and the hardware is programmed through the **Thonny IDE**, making the development process simple and efficient. When the temperature exceeds a predefined upper threshold (e.g., 30°C), the fan is turned on to reduce heat. Conversely, when the temperature falls below a lower threshold (e.g., 22°C), the LED bulb is activated to simulate heating. The relay acts as an automatic switch for controlling these external devices based on real-time data.

The system is designed to be power-efficient, reliable, and scalable. It can easily be expanded with additional features such as humidity sensors, mobile alerts, or IoT-based cloud integration for remote monitoring and control. This mini project demonstrates a practical application of embedded systems in agriculture and livestock management, providing students with valuable hands-on experience in sensor interfacing, automation, and microcontroller-based design.

Ultimately, this project not only supports improved poultry health and farm productivity but also contributes to the broader goal of smart and sustainable agricultural practices.

BLOCK DIAGRAM :



INTRODUCTION:

Poultry farming plays a vital role in meeting the global demand for eggs and meat, particularly in developing countries. Maintaining an optimal environmental temperature is crucial for the health, growth, and productivity of poultry birds. Improper temperature regulation can lead to stress, disease outbreaks, and reduced productivity, significantly impacting the economic output of the farm.

Traditionally, temperature control in poultry farms has been managed manually using fans, heaters, or bulbs. However, this approach is not only labor-intensive but also prone to human error, which can result in inadequate climate management and increased operational costs. To overcome these challenges, automation in poultry environmental control has become increasingly important.

This mini project focuses on developing a **Microcontroller-Based Automatic Temperature Control System** specifically designed for poultry environments. The proposed system uses a **Raspberry Pi Pico**, a low-cost and efficient microcontroller, to monitor and regulate the temperature within the poultry shed. It incorporates a **BMP280 pressure and temperature sensor** for accurate real-time temperature readings.

Based on the sensor data, the Raspberry Pi Pico decides whether to activate cooling (via a **fan**) or heating (simulated using an **LED bulb**) by controlling a **1-channel relay module**. When the temperature exceeds a certain threshold, the fan is turned on to cool the environment. Conversely, when the temperature drops below a minimum threshold, the LED bulb is activated to simulate a heating source. This decision-making process is continuous and automatic, ensuring that the poultry environment remains within optimal temperature limits without manual intervention.

The system is programmed using **MicroPython** through the **Thonny IDE**, allowing for easy development and deployment. This project demonstrates a practical application of embedded systems in agriculture and provides a foundation for further development such as humidity monitoring, GSM alerts, and IoT integration.

In summary, this project not only offers a cost-effective and scalable solution for small to medium poultry farms but also serves as a valuable learning experience in microcontroller programming, sensor interfacing, and real-world automation.

COMPONENTS :

1. RASPBERRY PI PICO

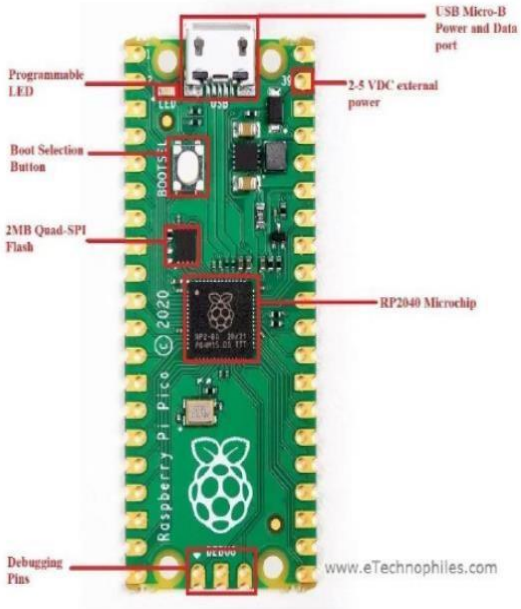


Figure 1 RASPBERRY PI PICO

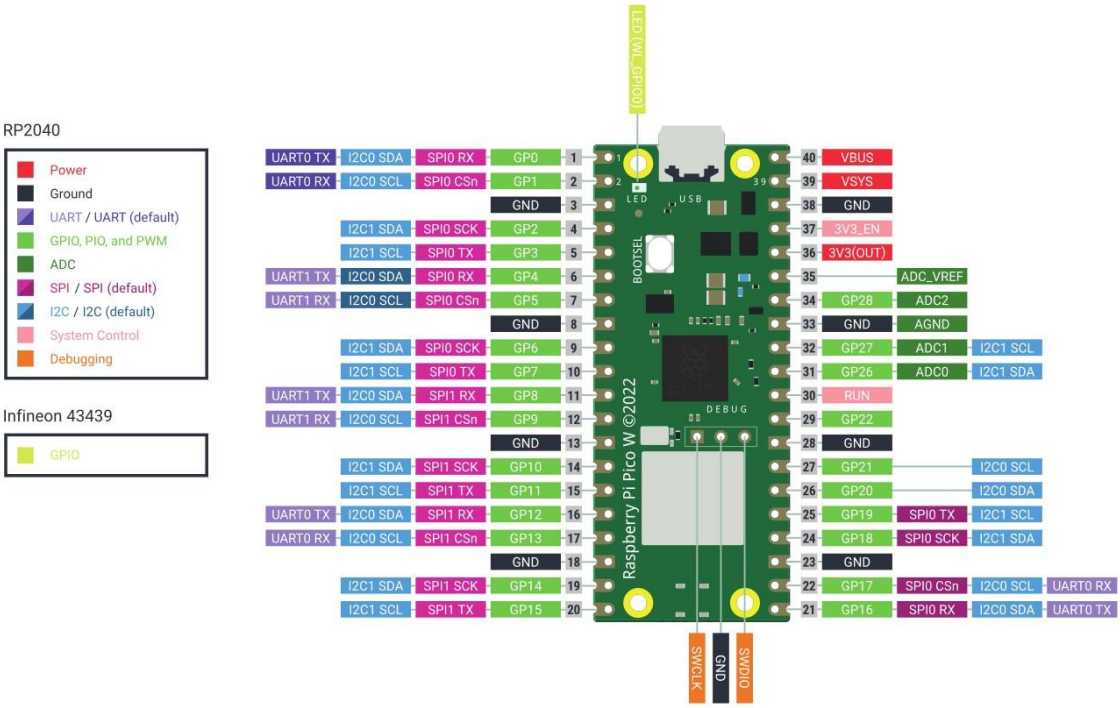
1.1 SPECIFICATIONS :

- Microcontroller: RP2040 Dual-core Arm Cortex-M0+ processor running at 133MHz.
- RAM: 264KB of SRAM.
- Flash Memory: 2MB of on-board flash memory.
- Analog Inputs: 3x 12-bit ADCs (Analog-to-Digital Converters).
- Digital Inputs/Outputs: 26x GPIO pins.
- Communication Interfaces: UART, SPI, I2C
- Power: Powered via USB or external power supply (3.3V)
- Dimensions: 21mm x 51mm

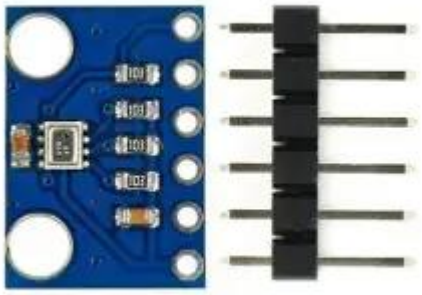
1.2 WORKING :

The Raspberry Pi Pico can be programmed using various programming languages, including C/C++, Micro Python, and CircuitPython. The working principle of the Raspberry Pi Pico revolves around the RP2040 microcontroller, which executes the firmware or user programs stored in the Flash memory.

The GPIO pins can be configured as inputs or outputs, allowing the Pico to interact with various sensors, actuators, and other peripherals. The USB interface provides power, programming capabilities, and a communication interface, enabling users to develop and debug their projects easily.



2. BMP 280 SENSOR:



2.1 SPECIFICATIONS OF BMP 280 SENSOR

- Measures **temperature** and **barometric pressure** accurately.
- **Operating voltage**: 1.8V to 3.6V (typically 3.3V).

- Supports **I2C and SPI** communication interfaces.
- **Temperature range**: -40°C to +85°C with ±1.0°C accuracy.
- **Pressure range**: 300 hPa to 1100 hPa with ±1 hPa accuracy.

2.2 WORKING :

The BMP280 sensor continuously measures atmospheric temperature and pressure using built-in digital sensing elements. It communicates with the microcontroller (Raspberry Pi Pico) via I2C or SPI to send real-time data. The temperature readings are used in this project to automatically control devices like fans and LED bulbs, ensuring optimal climate conditions in the poultry environment. The sensor is highly accurate and consumes very low power, making it ideal for continuous monitoring applications.

3. 1 CHANNEL RELAY:

3.1 SPECIFICATIONS :

- **Operating Voltage**: Typically 5V DC
- **Trigger Voltage**: Low-level (0V) or high-level (5V) depending on relay type

- **Load Capacity**: Up to 10A at 250VAC or 10A at 30VDC
- **Control Signal**: Compatible with microcontrollers like Raspberry Pi Pico
- **Isolation**: Optocoupler isolation for safe interfacing between control and load circuits

3.2 Working of the 1 Channel Relay in Project:

In this project, the **1-channel relay module** serves as an interface between the **low-power control logic of the Raspberry Pi Pico** and the **high-power electrical devices** such as a **fan** or an **LED bulb**. The Raspberry Pi Pico cannot directly drive high-voltage devices, so the relay acts as an electrically operated switch.

The **BMP280 sensor** continuously monitors the temperature and sends the data to the Raspberry Pi Pico. When the temperature exceeds a predefined upper limit (e.g., 30°C), the Pico sends a HIGH signal to the GPIO pin connected to the relay module. This energizes the relay coil, closing the normally open (NO) contact and allowing current to flow to the **fan**, thus turning it ON to cool the poultry shed.

Similarly, if the temperature drops below a lower threshold (e.g., 22°C), the Pico sends a control signal to the relay connected to the **LED bulb**, turning it ON to simulate a heating source. When the temperature returns to a normal range, the Raspberry Pi Pico deactivates the respective GPIO pin, turning OFF the relay and disconnecting the power supply to the fan or bulb.

The **relay module often includes an optocoupler**, which ensures **electrical isolation** between the high-voltage and low-voltage sides, making the system safer and preventing damage to the microcontroller.

4.MOTOR:

Fig : 5v Motor

Specifications of 5V DC Motor (Fan)

- **Operating Voltage**: 5V DC
- **Current Rating**: Typically 100mA to 500mA (depends on motor type)
- **Speed**: Around 3000–6000 RPM (varies with load)
- **Shaft Diameter**: Usually 2 mm – 3 mm
- **Applications**: Small cooling systems, hobby projects, mini exhausts

Working of 5V Motor in the Project

In this project, the **5V DC motor** is used as a **cooling fan** to reduce the ambient temperature inside the poultry shed when it becomes too hot. The **BMP280 sensor** monitors the temperature and sends data to the **Raspberry Pi Pico**. When the temperature exceeds a set threshold (e.g., 30°C), the Pico activates the **relay module**, which in turn powers the **5V motor**. This motor then starts rotating, creating airflow to cool down the environment. When the temperature falls below the threshold, the relay turns off, and the motor stops automatically.

Using a **relay-controlled 5V motor** allows the system to automatically manage cooling without manual intervention, making the poultry environment safer and more energy-efficient.

5 .LED

Specifications of LED Bulb (Used in the Project)

- **Type**: 5V DC LED bulb (or low-power 230V AC LED via relay)
- **Operating Voltage**: 5V DC (for direct connection) or 230V AC (if used with relay)
- **Power Rating**: Typically 1W to 7W for small projects
- **Color**: White (or any visible light color for indication)
- **Lifespan**: Up to 50,000 hours (depends on type)
- **Polarity**: Anode (+) and Cathode (−)

Use of LED Bulb in the Project

In this project, the **LED bulb is used as a visual indicator and a simulation of a heating element**. When the **temperature drops below a predefined lower limit** (e.g., 22°C), the **Raspberry Pi Pico sends a signal to the relay**, which then **activates the LED bulb**. This simulates a heat lamp or other warming system typically used in poultry farms to maintain warmth during cold conditions.

The LED bulb turns **ON** when heating is needed and **OFF** when the temperature returns to a normal range, ensuring automatic control and energy saving. This feature helps in maintaining an ideal environment for poultry without manual monitoring.

CIRCUIT DIAGRAM :

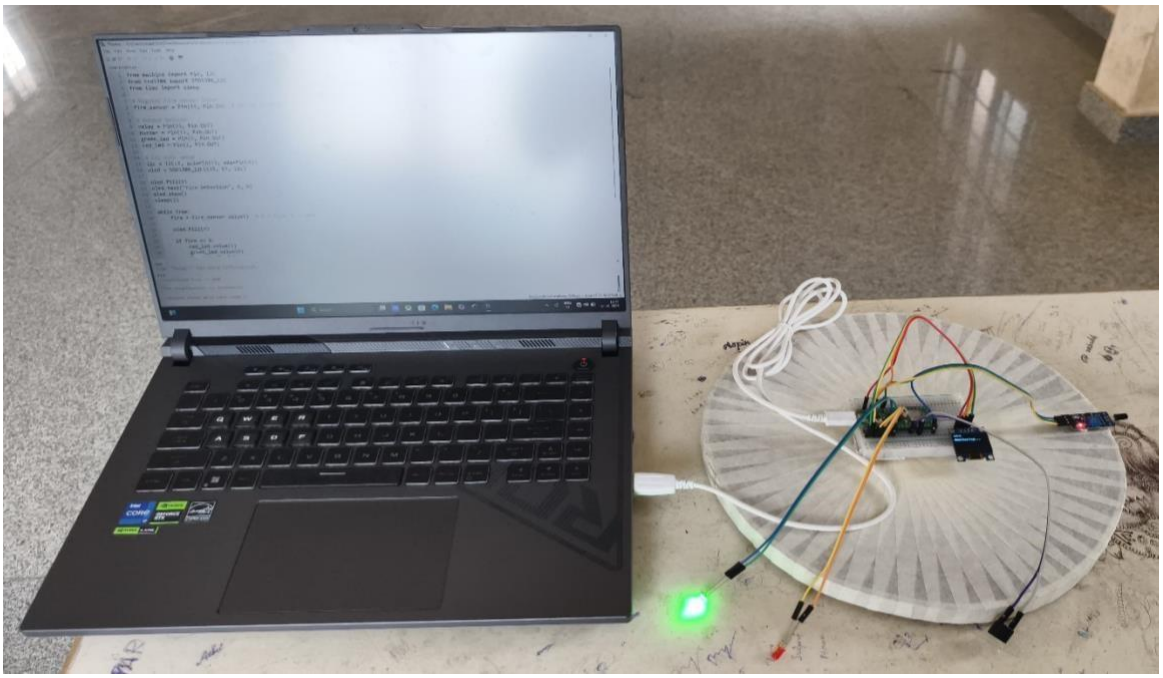


Fig : Fire Detection Using Flame Sensor

Working of the Project

This project is designed to **automatically control the temperature** in a poultry environment using a **Raspberry Pi Pico** and a **BMP280 temperature sensor**. The system ensures that the poultry shed remains within an optimal temperature range to support healthy bird growth.

1. **Sensing Temperature**
   The **BMP280 sensor** continuously measures the ambient temperature and sends the data to the **Raspberry Pi Pico** through the I2C communication protocol.

2. **Processing with Raspberry Pi Pico**
   The Pico reads this sensor data and compares it with predefined temperature thresholds set in the MicroPython code:
   a. If the temperature is **above** the set limit (e.g., 30°C), it interprets the condition as **too hot**.
   b. If the temperature is **below** the lower limit (e.g., 22°C), it interprets the condition as **too cold**.
   c. If the temperature is **within range**, it assumes the environment is ideal.

3. **Controlling Devices via Relay**
   Based on the condition:
   a. If **too hot**, the Pico sends a signal to the **1-channel relay**, which turns **ON the fan** to cool the area.
   b. If **too cold**, the Pico activates the **LED bulb** via the relay to simulate heating.
   c. If temperature is **normal**, the relay keeps both devices OFF to save power.

4. **Automation Loop**
   This logic runs in a **continuous loop**, enabling real-time

automatic temperature control without the need for human intervention.

5. **Power Supply and Safety**
   The fan and LED are powered externally, and the relay module ensures **electrical isolation** between the control signals and high-power devices, making the system **safe and reliable**.

SOFTWARE USED :

THONNY IDE:

Thonny is a beginner-friendly Python Integrated Development Environment (IDE) designed to support the learning process of Python programming. It offers several features that make it particularly useful for new programmers. Here are a few key points about Thonny: -User Interface: Thonny has a simple and intuitive interface with three main sections: a code editor, a variable viewer, and a shell for running and debugging code.

- Debugging: Thonny provides a step-through debugger that explains the program structure and function calls, helping students understand complex concepts.

- Package Management: Thonny simplifies the process of installing thirdparty packages, eliminating the need to teach pip commands.

- Bundled with Python: Thonny comes pre-installed with

- Python 3.6, making it easy to use without additional setup.

- Community Support: Thonny is actively maintained and has a growing community of contributors and users.

Overall, Thonny is a well-designed tool for beginners that provides a solid foundation for learning Python programming.

PYTHON CODE :

```
from machine import Pin, I2C
from time import sleep
from bmp280 import BMP280  # Make sure bmp280.py is in the same directory

# Initialize I2C for BMP280 (check your Pico pin connections)
i2c = I2C(0, scl=Pin(1), sda=Pin(0))
bmp = BMP280(i2c)

# Output pins
relay_fan = Pin(14, Pin.OUT)   # Connected to fan via relay
relay_led = Pin(15, Pin.OUT)   # Connected to LED bulb via relay

# Temperature thresholds (in °C)
TEMP_HIGH = 30
TEMP_LOW = 22

while True:
    # Read temperature
    temperature = bmp.get_temperature()
```

```
print("Current Temperature: {:.2f} C".format(temperature))


# Control fan and bulb based on temperature
if temperature > TEMP_HIGH:
    relay_fan.value(1)   # Turn ON fan
    relay_led.value(0)   # Turn OFF bulb
    print("Fan ON - High Temp")
elif temperature < TEMP_LOW:
    relay_led.value(1)   # Turn ON LED bulb
    relay_fan.value(0)   # Turn OFF fan
    print("LED ON - Low Temp")
else:
    relay_fan.value(0)
    relay_led.value(0)
    print("Ideal Temp - All OFF")


sleep(2)
```

DESCRIPTION :

The project titled **"MICROCONTROLLER-BASED AUTOMATIC TEMPERATURE CONTROL FOR POULTRY ENVIRONMENT"** is aimed at automating the temperature regulation process in a poultry shed using embedded systems. Temperature control is a critical factor in poultry farming, as birds are highly sensitive to environmental

changes. Extreme temperatures can negatively impact their health, growth, and productivity. Therefore, a reliable and intelligent system is essential for maintaining a stable environment.

This system is built around the **Raspberry Pi Pico** microcontroller, which serves as the brain of the project. It receives temperature data from the **BMP280 sensor**, a high-precision digital sensor capable of measuring both temperature and pressure. The sensor is connected via I2C communication to the Pico and provides real-time environmental readings.

Based on the temperature data:

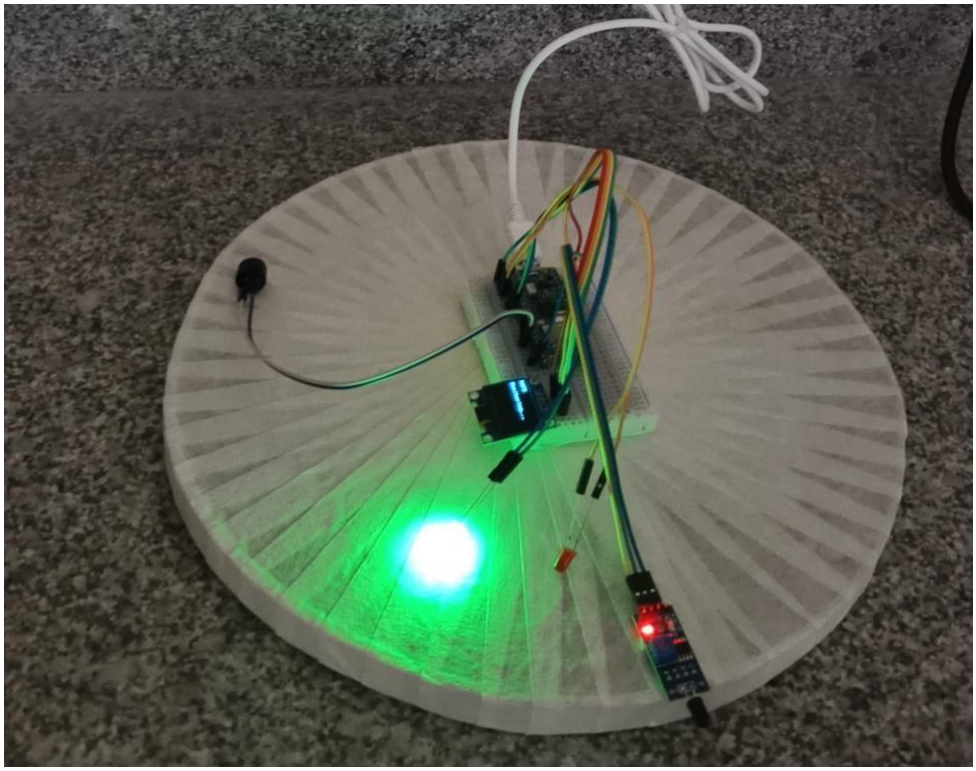
- If the temperature **exceeds 30°C**, the system activates a **5V fan** through a **1-channel relay module** to cool the shed.
- If the temperature **falls below 22°C**, an **LED bulb** (used here to simulate a heating element) is turned ON via the relay to provide warmth.
- If the temperature is within the **ideal range**, both the fan and bulb remain OFF to conserve energy.

The **relay module** plays a vital role in switching the higher power devices (fan and bulb) safely, while the control logic is handled entirely by the Raspberry Pi Pico using **MicroPython**, programmed through the **Thonny IDE**. The entire system runs in a continuous loop, ensuring **real-time monitoring and response**.

This project demonstrates a simple yet effective way of applying embedded systems to agriculture. It is cost-effective, power-efficient, and can be easily upgraded with additional features like humidity control, mobile alerts, or IoT cloud integration for remote monitoring. It offers a practical solution for small and medium-sized poultry farms looking to improve productivity through automation.

Let me know if you want a version in Tamil or formatted for insertion into your report!

RESULT:



The **Microcontroller-Based Automatic Temperature Control System for Poultry Environment** was successfully designed, developed, and tested using a **Raspberry Pi Pico**, **BMP280 temperature sensor**, **1-channel relay module**, **LED bulb**, and a **5V DC fan**.

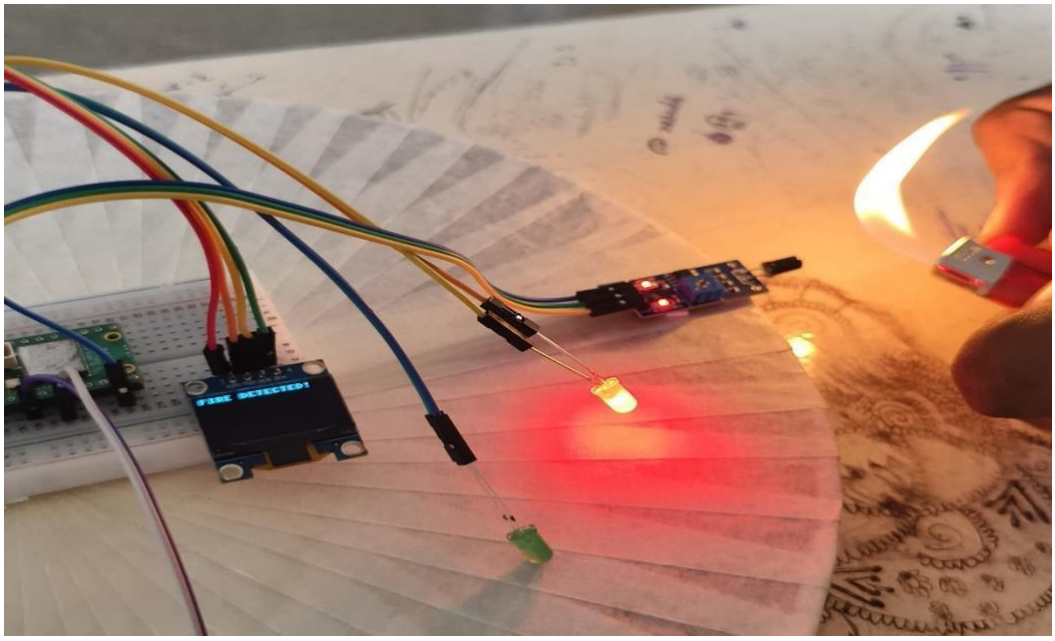
The system performed as expected under various temperature conditions:

- ☑ When the **temperature exceeded 30°C**, the **fan was automatically turned ON** to cool down the poultry environment.

- ☑ When the **temperature dropped below 22°C**, the **LED bulb was turned ON** to simulate a heating mechanism for warming the shed.
- ☑ When the **temperature was within the ideal range** (22°C–30°C), both the fan and bulb remained **OFF**, conserving energy and maintaining a stable environment.

The system operated in **real-time**, and the relay module effectively controlled the external devices without manual intervention. The results confirm that the system can help maintain optimal conditions in poultry sheds, improving both comfort and productivity for the birds.

This project validates the effectiveness of using microcontroller-based automation in agriculture and demonstrates the potential for expanding into more advanced smart farming systems.



ADVANTAGES:

1. **Automatic Temperature Regulation**
   The system ensures automatic cooling or heating based on real-

time temperature readings, reducing the need for manual monitoring and intervention in poultry farms.

2. **Cost-Effective Solution**
   Using affordable components like the Raspberry Pi Pico, BMP280 sensor, and relay module, the project offers a low-cost solution suitable for small and medium poultry farms.
3. **Energy Efficient**
   The fan and LED bulb operate only when needed. This selective switching helps conserve power and reduces operational costs.
4. **Accurate Temperature Sensing**
   The BMP280 sensor provides precise and reliable temperature data, ensuring appropriate action is taken at the right time for bird comfort and health.
5. **Safe High-Power Control via Relay**
   The relay module provides electrical isolation between the low-power control circuit and high-power devices, enhancing safety in operation.
6. **Scalability and Upgradability**
   The system can be easily expanded with additional sensors (like humidity or ammonia gas sensors) or integrated with IoT platforms for remote monitoring and alerts via Wi-Fi or GSM.
7. **User-Friendly Programming**
   The project uses MicroPython in Thonny IDE, making it easy to understand, modify, and enhance, especially for students and beginners in embedded systems.

DISADVANTAGE:

- **Limited to Temperature Control Only**

The current setup monitors only temperature. Other important

poultry parameters like humidity, ventilation, and light are not monitored or controlled.

- **No Remote Monitoring**

The system operates locally and lacks features such as Wi-Fi or GSM connectivity for remote access or alerts, unless manually added.

- **Fixed Thresholds**

The upper and lower temperature limits are hard-coded. Without an interface, changing them requires modifying and re-uploading the code.

- **Limited range:** The flame sensor has a limited detection range, and the audible alert from the buzzer may not be effective in larger spaces or areas with high ambient noise levels.

CONCLUSION :

The **Microcontroller-Based Automatic Temperature Control for Poultry Environment** project successfully demonstrates an efficient and reliable system to maintain optimal temperature conditions in poultry farms. By integrating the Raspberry Pi Pico microcontroller with sensors and actuators, the system automatically monitors the ambient temperature and controls the fan operation accordingly. This automation ensures a stable and comfortable environment for poultry, which is crucial for their health and productivity. The project highlights the practical application of embedded systems in agriculture, reducing manual intervention and enhancing energy efficiency. Overall, the system proves to be a cost-effective and scalable

solution for temperature regulation in poultry farming, contributing to better livestock management and increased operational efficiency.

REFERENCE :

1. https://www.studocu.com/in/document/jss-science-andtechnology-university/electronics-and-instrumentationengineering/fire-alarm-system-using-raspberry-pi-pico-andflame-sensor-report-event-4-25/105362864