

# DEVELOPMENT OF AN EMBEDDED SYSTEM APPLICATION: ADVANCED TEMPERATURE MONITORING SYSTEM USING ESP32 & DHT22

*P.Thimmaiah*

Assistant Professor, Dept. of Electronics, Sri Krishnadevaraya University,  
Anantapuramu, AP, India

**Abstract:** In today's world, temperature monitoring has become an essential part of various industries such as healthcare, food processing, and manufacturing. With the help of embedded systems, it is now possible to design and implement advanced temperature monitoring systems that can provide accurate and real-time data. This article discusses the design and implementation of a temperature monitoring system using Embedded system including the hardware and software components (ESP32 & DHT22) and the benefits of using such a system.

**Keywords:** ESP 32 Microcontroller, DHT 22 Sensor, Arduino IDE and wired & wireless communication

## 1. INTRODUCTION:

Temperature monitoring is a critical aspect of many industries, and it's essential to have an accurate and reliable system in place to ensure the safety and quality of the products. Embedded systems have become a popular choice for designing temperature monitoring systems due to their ability to perform complex tasks, their small size, and their low power consumption.

## 2. BLOCK DIAGRAM & WORKING

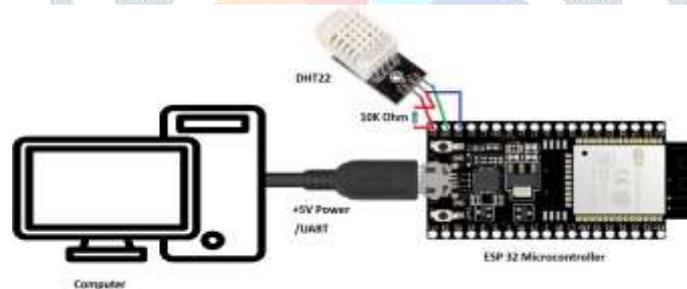


Figure 1: Block Diagram of the system

Fig:1 Demonstrates the block diagram of the advanced Temperature Monitoring System. It consists of ESP32 MCU and DHT22 Sensor. In a temperature monitoring system employing embedded systems, hardware components form the backbone of its functionality, comprising essential elements like a temperature sensor DHT22, microcontroller ESP32, and communication module UART.

### 2.1 HARDWARE COMPONENTS:

The temperature sensor serves as the frontline component, tasked with accurately measuring the ambient temperature of the environment under observation. This sensor type can vary depending on the specific requirements of the application, with options ranging from thermistors to thermocouples or resistance temperature detectors (RTDs). Regardless of the sensor choice, its primary function remains consistent: to provide reliable temperature data for further processing and analysis.

At the heart of the system lies the microcontroller, a miniature computing unit that orchestrates the entire operation. Endowed with processing power and programmable logic, the microcontroller interprets the temperature readings received from the sensor, executes predefined algorithms for data processing, and orchestrates system control mechanisms. Its role extends beyond mere data processing; the microcontroller ensures system stability, regulates power consumption, and facilitates seamless interaction between hardware and software components.

Complementing the sensor and microcontroller is the communication module, pivotal for enabling remote monitoring and data analysis. This module acts as the conduit for transmitting temperature data from the monitoring system to a centralized location, where it can be accessed, analyzed, and acted upon in real-time. Communication modules leverage diverse protocols

such as wired (UART), wireless (Wi-Fi, Bluetooth & Zigbee) each tailored to suit specific range, bandwidth, and power consumption requirements.

Together, these hardware components form a cohesive ecosystem, seamlessly integrating to deliver accurate, timely, and actionable insights for temperature monitoring and control applications. Their symbiotic relationship underscores the importance of hardware synergy in enabling efficient data acquisition, processing, and dissemination in embedded systems-based temperature monitoring setups.

## 2.2 TEMPERATURE SENSORS:

Temperature sensors encompass a diverse array of technologies, each tailored to meet specific application needs and accuracy requirements. Among the most commonly employed are thermistors, thermocouples, and resistance temperature detectors (RTDs), each offering unique advantages and performance characteristics. Thermistors, distinguished by their affordability and compact size, find widespread use in consumer electronics and household appliances. Their semiconductor-based design allows for rapid response times and precise temperature measurements within a defined range, making them ideal for applications where cost-effectiveness and space constraints are paramount.

Conversely, thermocouples represent a robust solution capable of withstanding extreme temperatures and harsh environments. Composed of two dissimilar metal wires, thermocouples generate a voltage proportional to the temperature difference between their junctions, enabling accurate temperature measurement across a broad spectrum of operating conditions. This versatility renders thermocouples indispensable in industries such as manufacturing, automotive, and aerospace, where temperature extremes are commonplace. Resistance temperature detectors (RTDs) embody the pinnacle of temperature sensing precision and stability. Constructed from pure metals or metal alloys with predictable resistance-temperature characteristics, RTDs deliver unparalleled accuracy and reliability, particularly in applications demanding precise temperature control or calibration. Their linear response and low drift make them indispensable in scientific research, pharmaceutical manufacturing, and laboratory environments where precise temperature measurement is paramount.

In essence, the selection of temperature sensors hinges upon a careful consideration of factors such as accuracy, operating conditions, and cost constraints. By leveraging the unique capabilities of thermistors, thermocouples, or RTDs, organizations can ensure optimal performance and reliability in temperature monitoring applications across diverse industries and environments. In this research discussion used the DHT 22 Temperature Sensor. DHT22 output calibrated digital signal utilizes exclusive digital signal sensing technology, assuring its reliability and stability. DHT22 is specified with the following operating voltage: 3.5V to 5.5V, operating current: [0.3mA (measuring), 60uA (stand by)], output serial data, temperature range from -40 degree Celsius to 80 degree Celsius.

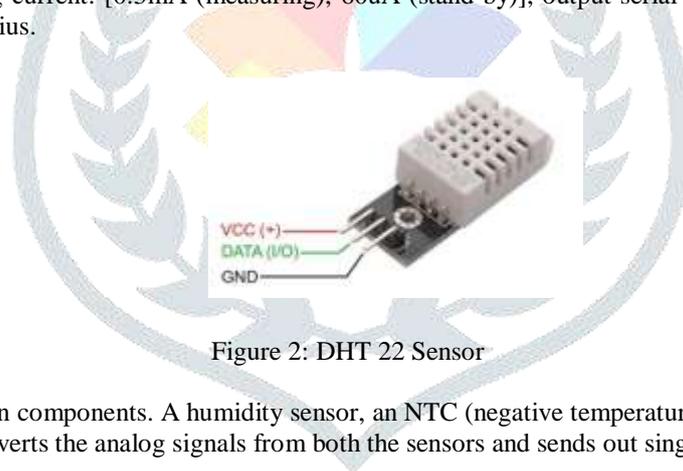


Figure 2: DHT 22 Sensor

DHT22 Sensor consists of 3 main components. A humidity sensor, an NTC (negative temperature coefficient) thermistor and an 8-bit microcontroller, which converts the analog signals from both the sensors and sends out single digital signal.

The data from the DHT22 sensor consists of 40 bits and the format is as follows:

1. 8 – Bit data for integral part of RH value,
2. 8 – Bit data for decimal part of RH value,
3. 8 – Bit data for integral part of Temperature value,
4. 8 – Bit data for decimal part of Temperature value and
5. 8 – Bit data for checksum.

If the data transmission is right, the check-sum should be the last 8 Bit of “8 Bit integral RH data + 8 Bit decimal RH data + 8 Bit integral Temperature data + 8 Bit decimal Temperature data.

## 2.3 MICROCONTROLLERS:

Microcontrollers serve as the brain of embedded systems, providing the computational power necessary to execute specific tasks with precision and efficiency. These compact computing devices are meticulously engineered to perform dedicated functions, making them indispensable in a myriad of applications, including temperature monitoring systems.

Within the realm of temperature monitoring, microcontrollers play a pivotal role in orchestrating the entire operation. Acting as the central processing unit, a microcontroller interfaces with temperature sensors, retrieves raw data, and initiates the

necessary computations. This data processing capability enables microcontrollers to analyze temperature readings in real-time, facilitating prompt responses to changing environmental conditions.

Moreover, microcontrollers assume the responsibility of system control, overseeing the operation of peripheral devices and ensuring seamless integration within the monitoring system. Through tailored algorithms and logic circuits, they orchestrate tasks such as activating cooling systems or triggering alarms based on predefined thresholds, thereby maintaining optimal temperature levels in controlled environments.

Additionally, microcontrollers serve as the conduit for data transmission in temperature monitoring systems. Leveraging communication protocols wired and wireless such as UART, Wi-Fi, Bluetooth, Zigbee, they facilitate the seamless transfer of temperature data to centralized locations for further analysis and monitoring. This connectivity enables remote access capabilities, empowering users to monitor temperature conditions from virtually anywhere, enhancing operational efficiency and responsiveness.

In essence, microcontrollers embody the technological backbone of temperature monitoring systems, harnessing their computational prowess to ensure precise data acquisition, efficient system control, and seamless data transmission. Their versatility, reliability, and performance make them indispensable components in the realm of embedded systems, empowering organizations to uphold stringent temperature control standards and safeguard critical assets. In this research discussion used the ESP32 Microcontroller

## 2.4 COMMUNICATION MODULES:

Communication modules play a vital role in the seamless transmission of data from temperature monitoring systems to centralized locations, facilitating real-time monitoring and in-depth analysis. These modules leverage various communication protocols such as Wi-Fi, Bluetooth, and ZigBee, each offering distinct advantages tailored to specific application needs.

UART is a serial communication protocol. UART uses two wires to transmit and receive data in both directions. Data is sent in frames, which typically include a start bit, data bits, an optional parity bit, and stop bits. The start bit indicates the beginning of a frame, and the stop bits signal the end.

Wi-Fi, known for its widespread availability and high bandwidth capabilities, is often preferred for applications requiring extensive data transfer over relatively long distances. It provides robust connectivity, making it suitable for environments with multiple monitoring points or complex infrastructures.

Bluetooth, on the other hand, is characterized by its low power consumption and short-range communication capabilities, making it ideal for scenarios where energy efficiency and proximity are prioritized. It is commonly used in wearable devices or localized monitoring systems where direct communication between devices is essential.

Zigbee offers a balance between range, data rate, and power consumption, making it suitable for applications that require reliable communication over medium distances while minimizing energy consumption. Its mesh networking capability allows for scalability and redundancy, making it well-suited for large-scale monitoring systems or environments with potential signal interference.

The choice of communication protocol depends on several factors including the specific requirements of the application, such as range, bandwidth, power consumption, and environmental considerations. By selecting the most appropriate communication module and protocol, organizations can ensure efficient and reliable data transmission for effective temperature monitoring and analysis.

## 3. SOFTWARE COMPONENTS:

In a temperature monitoring system employing embedded systems, software components are pivotal for seamless operation and efficient data management. These components primarily consist of a user interface, data processing module, and communication module.

The user interface serves as the primary interaction point between the system and users, facilitating the display of temperature data in a comprehensible format. Additionally, it allows users to control various aspects of the monitoring system, such as setting thresholds or configuring alerts. The user interface is designed to be intuitive and user-friendly, ensuring accessibility for operators of varying technical backgrounds.

The data processing module plays a critical role in handling the vast amounts of data collected by the monitoring system. This module is responsible for processing raw temperature data, performing necessary calculations or transformations, and generating insightful reports or visualizations. Through sophisticated algorithms and analysis techniques, the data processing module can extract valuable insights from the collected data, aiding in decision-making and trend analysis.

Furthermore, the communication module acts as the bridge between the monitoring system and external entities, facilitating seamless data transmission to a central location for monitoring and analysis purposes. Utilizing various communication protocols such as Wi-Fi, Bluetooth, or Zigbee, this module ensures reliable and efficient transfer of temperature data, enabling real-time monitoring and remote access capabilities.

By integrating these software components effectively, temperature monitoring systems can offer enhanced functionality, robust performance, and actionable insights for users. This holistic approach to software development ensures that the system meets the diverse needs of its users while maintaining reliability and scalability in temperature monitoring applications. In this research discussion used the opensource Arduino Integrated Development Environment (IDE) for developing the temperature monitoring software. Arduino IDE contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the ESP32 hardware to upload programs and communicate with them. Figure 3 demonstrates the Arduino IDE.

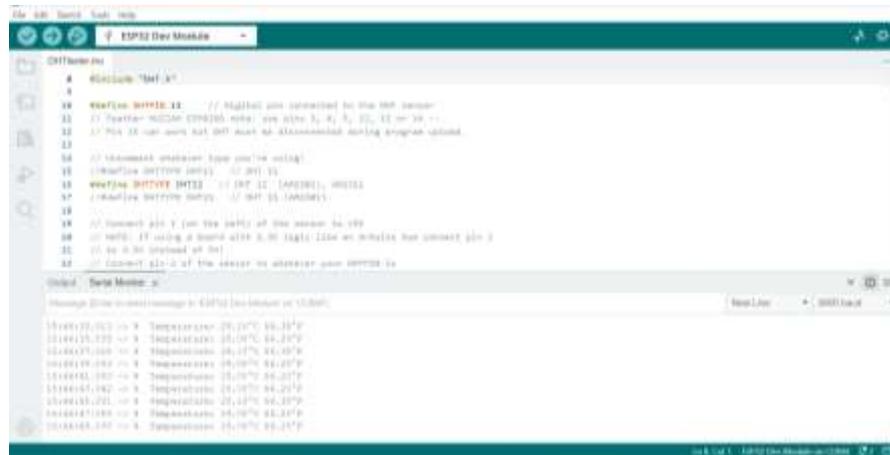


Figure 3: Arduino IDE

Figure 4 demonstrate the advance temperature monitoring system test setup. Temperature Result are Monitored in Serial terminal. This system further scalable to monitor the temperature remotely.

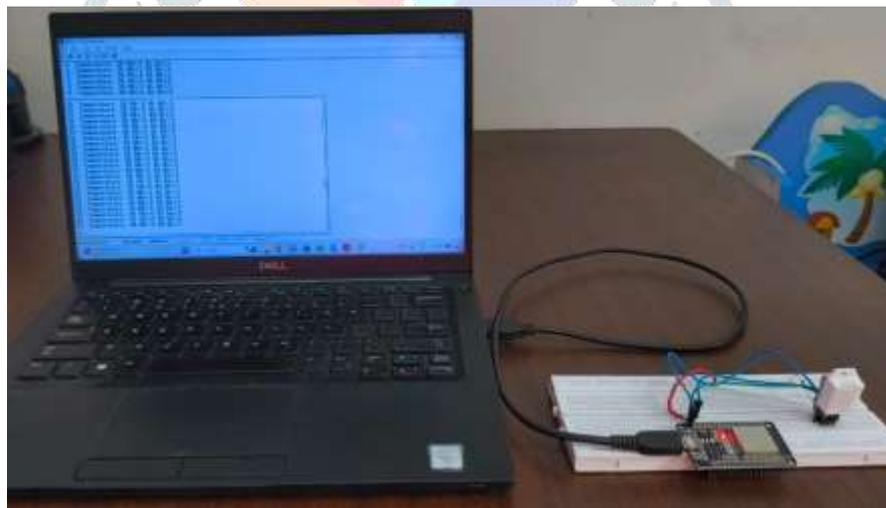


Figure 4: Temperature Monitoring Test Setup

### 3.1 DHT22 SENSOR WORKING PRINCIPLE:

When MCU send start signal, DHT22 change from low-power-consumption-mode to running-mode. When MCU finishes sending the start signal, DHT22 will send response signal of 40-bit data

- Step 1: MCU send out start signal to DHT22,
- Step 2: DHT22 send response signal to MCU
- Step 3: DHT22 send data to MCU

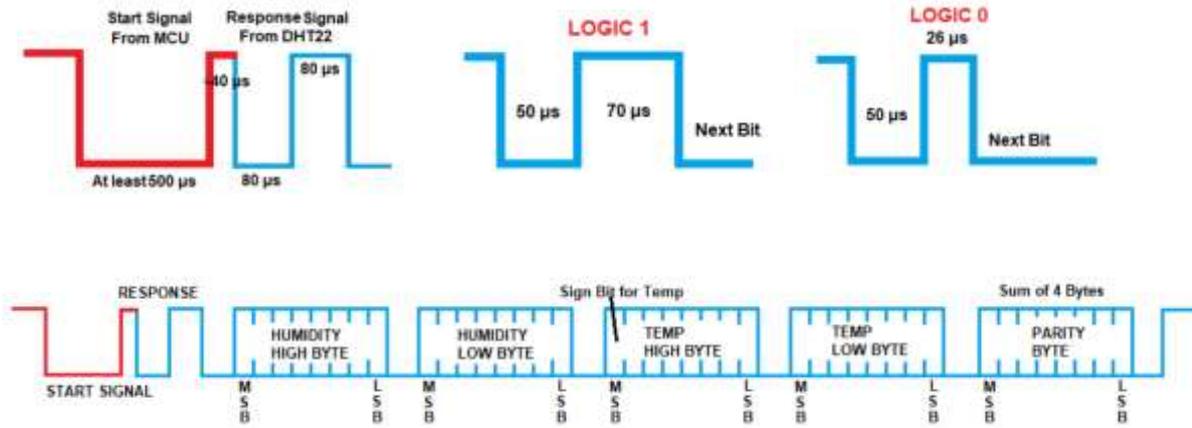


Figure 5: DHT 22 Timing diagram

When DHT22 is sending data to MCU, every bit's transmission begins with low-voltage-level that last 50us, the following high-voltage-level signal's length decide the bit is "1" or "0".

### 3.4 ARDUINO IDE PREPARATION

We need to download a couple of libraries, so that ESP32 will properly communicate with DHT22 Sensor. First is main DHT22 Sensor Library. Go to Tools -> Manage Libraries... in Arduino IDE.

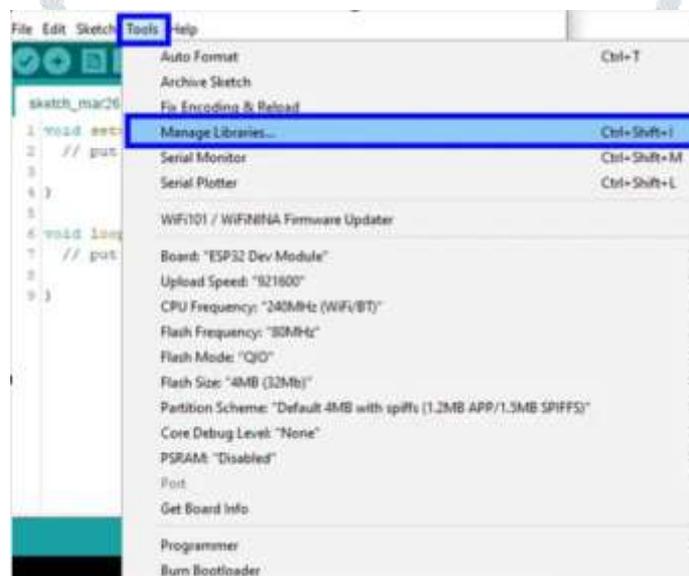


Figure 6: Arduino IDE Preparation

In the search bar, enter 'dht'. Scroll through the options and install "DHT sensor library" by Adafruit.



Figure 7: Arduino "DHT Sensor Library" Selection

The next library is associated with the Adafruit itself. Search for “adafruit unified”, scroll down and install “Adafruit Unified Sensor” library.



Figure 8: Arduino “Adafruit Unified Sensor” Library Selection

### 3.5 CODE

Write the Below code in Arduino Sketch and Save the Sketch as DHT22.ino file, After Saving the file, compile the code and upload the binary in to ESP32 Microcontroller. Program will be running and continuously reading the temperature every 2 seconds and printing the temperature data in the serial terminal.

```
#include "DHT.h"

#define DHTPIN 13 // Digital pin13 connected to the DHT sensor
#define DHTTYPE DHT22
// Initialize DHT sensor.
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println(F("DHT22 Monitoring"));
  dht.begin();
}

void loop() {
  // Wait for 2 seconds between measurements.
  delay(2000);

  // Read temperature as Celsius
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  Serial.print(F("% Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
  Serial.println(F("°F"));
}
```

### 4. BENEFITS OF USING EMBEDDED SYSTEMS FOR TEMPERATURE MONITORING:

There are several benefits of using embedded systems for temperature monitoring, including:

1. **Real-time data:** Embedded systems can provide real-time data, allowing for quick responses to changes in temperature.
2. **Accurate data:** Embedded systems can provide accurate data, ensuring that the temperature is monitored accurately.
3. **Low power consumption:** Embedded systems are designed to consume low power, making them ideal for battery-powered applications.

4. **Small size:** Embedded systems are small in size, making them easy to integrate into various devices and systems.
5. **Cost-effective:** Embedded systems are cost-effective, making them an attractive option for temperature monitoring applications.

## 5. CONCLUSION:

In conclusion, temperature monitoring systems using embedded systems are an effective and efficient way to monitor temperature in various industries. With the help of embedded systems, it is now possible to design and implement advanced temperature monitoring systems that can provide accurate and real-time data.

## REFERENCES:

- [1] “Embedded Systems - Architecture, Programming and Design”, Raj Kamal, Publs. McGraw-Hill Education
- [2] N. N. Prince, A. Theophilus, O. D.A. Onwuzulike, and N. Vincent. Design and Implementation of Microcontroller Based Automatic Fan Speed Regulator (Using Temperature Sensor). International Journal of Engineering Research and Management (IJERM), vol. 01, 2014, p. 205.
- [3] Nor Alina Khairi, Asral Bahari Jambek, Teoh Wei Boon, Uda Hashim, “Design and Analysis of a Wireless Temperature Monitoring System” on 2013, RSM2013 Proc.
- [4] Gungur Bal, Nihat Daldal, “Design and Implementation based Temperature Measurement and Control System using Powerline Communications” on 2013 at 2013 Fourth International Conference.
- [5] Bai, Y., & Ku, Y. (2008). Automatic Room Light Intensity Detection and Control Using a Microprocessor and Light Sensors. IEEE Transactions on Consumer Electronic, 1173.
- [6] A. Javed, Building Arduino Projects for the Internet of Things, Experiments with Real-World Applications. United States of America: Apress Media, LLC, 2016, pp. 15-34.
- [7] Deeksha Srivastava, Awanish Kesarwani, Shivani Dubey “Measurement of Temperature and Humidity by using Arduino” in International Research Journal of Engineering and Technology, Volume: 05 Issue: 12 | Dec 2018
- [8] S. Kalaiarasi, Shubham Gautam “Arduino Based Temperature and Humidity Sensor” in Journal of Network Communications and Emerging Technologies (JNCET), Volume 8, Issue 4, April (2018).