UNIVERSIDADE PRESBITERIANA MACKENZIE
– Faculdade de Computação e Informática –

Sistema de Alerta de Temperatura e Umidade com Comunicação via MQTT

Ione Ribeiro, Leandro Carlos Fernandes

[1] Faculdade de Computação e Informática
Universidade Presbiteriana Mackenzie (UPM) – São Paulo, SP – Brazil
10407966@mackenzista.com.br

Abstract. *This project aims to create a simple temperature and humidity monitoring system using the DHT11 sensor and an ESP32 microcontroller. The system is designed for industrial applications where remote monitoring of environmental conditions is essential. Data will be sent to a remote server via MQTT, and local alerts will be activated when predefined thresholds are exceeded, promoting efficiency and sustainability in industrial operations.*

Resumo. *O projeto consiste na criação de um sistema de monitoramento de temperatura e umidade utilizando o sensor DHT11 e a placa ESP32, focado em aplicações industriais. O sistema visa monitorar remotamente as condições ambientais em instalações empresariais e industriais, onde o controle de temperatura e umidade é crucial para a otimização de processos e preservação de materiais. Os dados serão enviados para um servidor remoto via protocolo MQTT, e alertas locais serão acionados quando os limites de temperatura ou umidade forem ultrapassados.*

1. Introdução

Com o avanço da Internet das Coisas (IoT) no setor industrial, sistemas de monitoramento remoto têm se tornado essenciais para garantir a eficiência, produtividade e sustentabilidade de processos em instalações empresariais (Ayaz et al., 2019). Este projeto propõe a construção de um sistema de monitoramento de temperatura e umidade utilizando o sensor DHT11 e o microcontrolador ESP32, componentes acessíveis e eficientes para prototipagem IoT (Espressif Systems, ESP32 Datasheet). O uso de IoT em ambientes industriais permite o controle preciso de condições ambientais, como temperatura e umidade, fatores críticos para a preservação de equipamentos, materiais e produtos sensíveis (Santos et al., 2021).

A comunicação será realizada via MQTT (Message Queuing Telemetry Transport), um protocolo leve e confiável, amplamente utilizado na indústria por sua eficiência na transmissão de dados em tempo real entre dispositivos conectados (IBM, s.d.). Trabalhos semelhantes em ambientes industriais demonstraram que o monitoramento remoto por IoT pode otimizar processos produtivos, reduzir desperdícios e melhorar a

sustentabilidade, alinhando-se ao ODS 9, que visa promover infraestrutura resiliente e inovação industrial (Heath, 2021).

2. Materiais e Métodos

Componentes Utilizados:

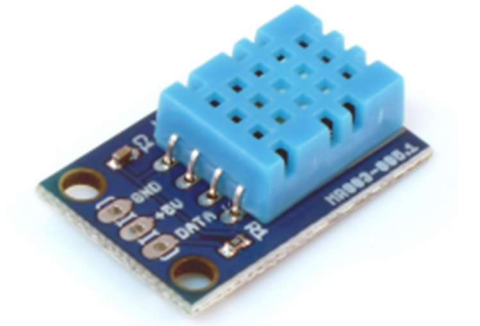
ESP32:

Descrição: O ESP32 é um microcontrolador que possui conectividade Wi-Fi e Bluetooth, sendo ideal para projetos de IoT. Ele será utilizado para conectar o sistema à internet e enviar os dados de temperatura e umidade via protocolo MQTT.

Fonte:https://lobodarobotica.com/blog/o-que-e-esp32-pra-que-serve-quando-usar/#google_vignette

Datasheet:https://www.theengineeringprojects.com/2020/12/esp32-pinout-datasheet-features-applications.html

Sensor de Temperatura e Umidade (DHT11):

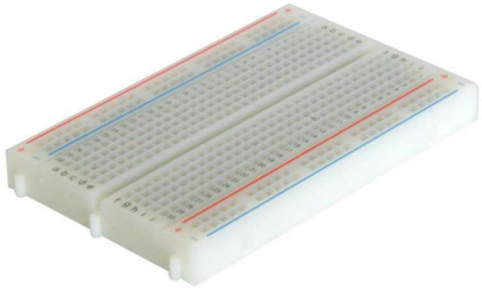


Descrição: O sensor DHT11 será utilizado para medir a temperatura e a umidade do ambiente. Ele é ideal para projetos simples que requerem monitoramento de condições ambientais.

Fonte: https://www.adafruit.com/product/386

Datasheet:
https://www.tme.eu/Document/7a4fd48d400b8c4c8309ef1e2b13cdd4/MR003-005-1.pdf

Protoboard:



Descrição: Usado para montar e testar os componentes sem a necessidade de solda, facilitando a prototipagem do sistema.

Fonte:https://www.mouser.com/datasheet/2/58/BPS-DAT-(BB400)-Datasheet-932623.pdf?srsltid=AfmBOoqtX55yf172-dUktIXSHqYnVHItWrZWKGQ5DwxfLnswUGctXfBs

Fios Jumper:

Descrição: Utilizados para conectar os diferentes componentes no protoboard e realizar a comunicação entre o ESP32 e os sensores.

Fonte: https://www.makerhero.com/categoria/prototipagem/jumpers/

Atuador Led:

Descrição: Atuam como atuadores visuais para indicar diferentes estados do sistema. O LED verde indica que a temperatura está dentro da faixa aceitável, enquanto o LED vermelho acende quando a temperatura ultrapassa os limites estabelecidos.

Fonte: https://www.makerhero.com/blog/aprenda-a-piscar-um-led-com-arduino/

Fonte de Alimentação:

Descrição: O sistema será alimentado via USB, conectado ao computador durante o desenvolvimento e testes.

Ferramentas Utilizadas:

IDE do Arduino:

Descrição: Ferramenta de desenvolvimento utilizada para programar o ESP32. A IDE permite a criação do código e a transferência do firmware para o microcontrolador.

Fonte: https://www.arduino.cc/en/software

Bibliotecas:

DHT sensor library: Utilizada para ler os dados do sensor DHT11.

PubSubClient: Biblioteca para comunicação via protocolo MQTT.

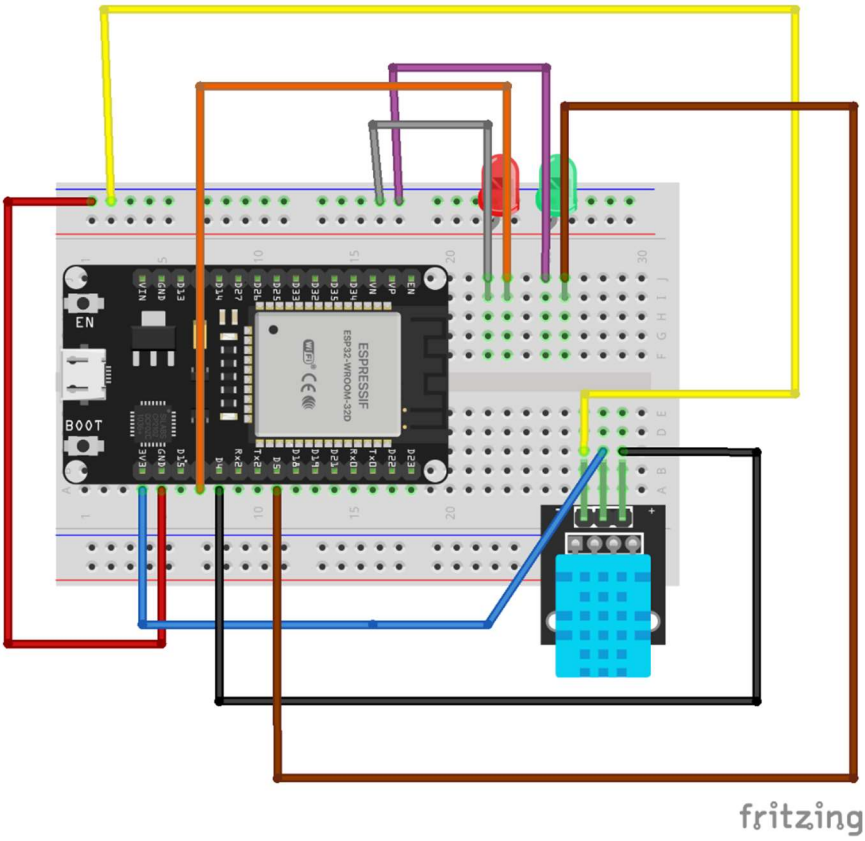
Protocolo MQTT:

Implementado para permitir a comunicação e o monitoramento remoto do sistema. O protocolo MQTT facilita o envio e recebimento de dados de dispositivos IoT em tempo real.

Software Fritzing:

Utilizado para criar o diagrama do protótipo.

Diagrama de Montagem:



3. Funcionamento

O sistema proposto monitora a temperatura ambiente e a umidade relativa em tempo real. O sensor DHT11, amplamente utilizado em projetos de IoT por sua simplicidade e eficiência (DHT11 Datasheet), realiza leituras periódicas e envia os dados ao microcontrolador ESP32. Os valores de temperatura e umidade são processados para:

- Acionar LEDs locais, que indicam se a temperatura está dentro ou fora do intervalo aceitável.
- Enviar as leituras ao broker MQTT, que distribui as informações para um servidor remoto onde os dados podem ser monitorados.

Lógica de Acionamento dos LEDs

A lógica de acionamento dos LEDs foi projetada para sinalizar visualmente o estado atual da temperatura:

- LED vermelho: Acende se a temperatura detectada for maior ou igual a 28°C, sinalizando que a temperatura ultrapassou o limite seguro para a preservação de equipamentos e materiais sensíveis.

- LED verde: Permanece aceso para temperaturas abaixo de 28°C, indicando condições dentro do intervalo aceitável.

Essa lógica foi implementada para garantir que somente um LED fique aceso por vez, desligando o LED oposto a cada nova leitura, com o objetivo de minimizar conflitos de sinal.

Integração com o Protocolo MQTT

A comunicação entre o sistema local e o servidor remoto foi implementada utilizando o protocolo MQTT, conhecido por ser leve e eficiente, especialmente em aplicações de IoT (IBM, s.d.). O fluxo da comunicação é o seguinte:

Conexão ao broker MQTT: Após estabelecer conexão com a rede Wi-Fi, o ESP32 se conecta ao broker MQTT, configurado no Adafruit IO, que gerencia o envio e recebimento de dados.

Publicação de dados: A cada leitura do sensor, os valores de temperatura e umidade são enviados aos feeds configurados no Adafruit IO:

- Feed de temperatura (idribeiro/feeds/temperatura): Recebe os valores em graus Celsius.
- Feed de umidade (idribeiro/feeds/umidade): Recebe os valores em porcentagem.

Monitoramento em tempo real: O Adafruit IO apresenta os valores recebidos em tempo real por meio de:

- Gráficos de linha (line charts) para análise histórica.
- Medidores (gauge) que exibem os valores instantâneos de forma visual.

Gestão de Conexões

Para garantir a robustez do sistema, o código implementa verificações contínuas da conexão com o broker MQTT. Em caso de falha, o ESP32 realiza tentativas automáticas de reconexão, assegurando a continuidade do envio dos dados.

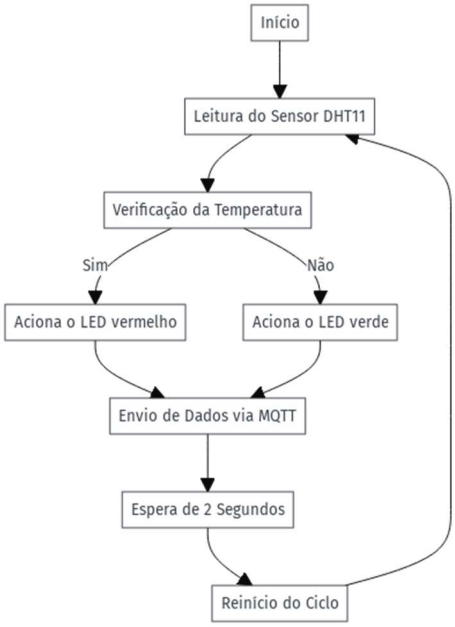
Monitoração em Tempo Real

No servidor Adafruit IO, o dashboard exibe:

- Um gráfico de linha para acompanhar as variações de temperatura e umidade ao longo do tempo.
- Medidores que apresentam os valores atuais de forma clara e intuitiva. Esses elementos facilitam a análise de tendências e permitem ações preventivas com base nos dados recebidos.

Fluxograma de funcionamento:

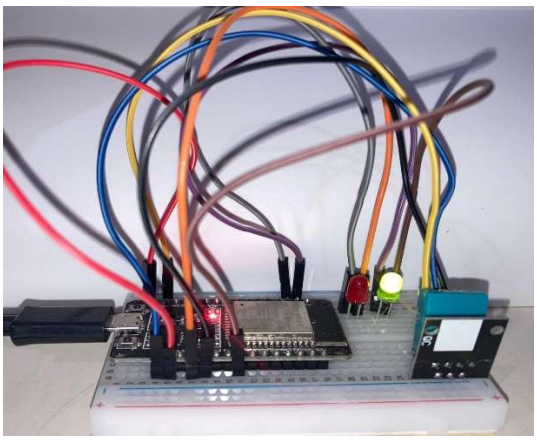
O fluxograma abaixo representa a lógica operacional do sistema de monitoramento de temperatura e umidade. Ele mostra, em sequência, as etapas do processo de coleta de dados, acionamento dos LEDs e envio de informações via MQTT.



O fluxo operacional do sistema pode ser resumido em cinco etapas principais:

- Leitura do sensor DHT11: Mede a temperatura e a umidade do ambiente.
- Verificação da temperatura: Determina se a temperatura está acima ou abaixo de 28°C.
- Acionamento dos LEDs: O LED correspondente (vermelho ou verde) é acionado com base no valor da temperatura.
- Envio dos dados via MQTT: Os valores são enviados aos feeds configurados no Adafruit IO.
- Reinício do ciclo: Após um intervalo de dois segundos, o ciclo é reiniciado para coletar novos dados.

Hardware Montado:



Descrição do Software Desenvolvido

O código desenvolvido para o projeto foi programado utilizando a IDE do Arduino e tem como objetivo monitorar a temperatura ambiente em tempo real, acionando LEDs para sinalização local e enviando os dados coletados via protocolo MQTT para um servidor remoto. Abaixo está uma descrição detalhada das principais funções e lógicas utilizadas no código:

Leitura de Dados do Sensor DHT11

O código utiliza a biblioteca DHT.h para gerenciar o sensor DHT11, que mede a temperatura e umidade. A função dht.readTemperature() realiza a leitura da temperatura, enquanto a função dht.readHumidity() lê a umidade. Esses dados são coletados em intervalos de dois segundos, definidos pelo comando delay(2000).

Lógica de Acionamento dos LEDs

Após a leitura da temperatura, o código verifica se a temperatura está abaixo de 28°C. Caso positivo, o LED verde é acionado utilizando o comando digitalWrite(LED_GREEN, HIGH), sinalizando que a temperatura está dentro do intervalo seguro. Caso contrário, o LED vermelho é acionado utilizando o comando

digitalWrite(LED_RED, HIGH), indicando que a temperatura ultrapassou o limite aceitável.

Para garantir que apenas um LED esteja aceso por vez, o código desativa o LED não necessário em cada leitura, utilizando digitalWrite(LED_GREEN, LOW) ou digitalWrite(LED_RED, LOW).

Envio de Dados via Protocolo MQTT

A comunicação MQTT foi configurada utilizando a biblioteca Adafruit_MQTT_Client, que gerencia a conexão com o broker MQTT. As leituras de temperatura e umidade são enviadas para dois feeds específicos no servidor remoto:

- Feed de Temperatura: Os valores em graus Celsius são publicados no tópico idribeiro/feeds/temperatura.
- Feed de Umidade: Os valores em percentual de umidade são publicados no tópico idribeiro/feeds/umidade.

A função temperaturaPub.publish(temperature) é responsável por enviar os valores de temperatura, enquanto umidadePub.publish(humidity) envia os valores de umidade.

Loop Principal

O loop principal do código executa as seguintes etapas:

- Realiza a leitura do sensor DHT11.
- Verifica os valores de temperatura e umidade.
- Aciona os LEDs com base no valor da temperatura.
- Publica os dados no broker MQTT.
- Reinicia o ciclo após um intervalo de dois segundos.

Integração com o Protocolo MQTT

A integração do protocolo MQTT no projeto foi realizada utilizando a biblioteca Adafruit MQTT, permitindo a conexão com o broker MQTT e a publicação dos dados coletados pelos sensores. O fluxo de envio dos dados funciona da seguinte forma:

Conexão ao Broker MQTT

Após a conexão do ESP32 à rede Wi-Fi, o código utiliza a biblioteca MQTT para estabelecer uma conexão com o broker da Adafruit IO. O broker atua como o

intermediário entre o dispositivo IoT e o sistema de monitoramento, gerenciando a comunicação e armazenando temporariamente os dados publicados.

Publicação dos Dados

- As leituras do sensor DHT11 são enviadas para dois feeds específicos:
- O feed de temperatura (idribeiro/feeds/temperatura) recebe os valores em graus Celsius a cada 2 segundos.

O feed de umidade (idribeiro/feeds/umidade) recebe os valores em percentual no mesmo intervalo.

Monitoração em Tempo Real

Os valores enviados ao broker são exibidos em tempo real no dashboard do Adafruit IO, que foi configurado com as seguintes visualizações:

- Gráficos de linha (Line Charts): Monitoram a variação dos valores ao longo do tempo.
- Medidores (Gauges): Exibem os valores instantâneos de temperatura e umidade.

Gestão de Conexões e Reconexões

O código verifica continuamente o estado da conexão com o broker MQTT. Caso a conexão seja perdida, o ESP32 tenta reconectar automaticamente, garantindo que o sistema permaneça funcional e confiável.

4. Resultados

Descrição dos resultados:

Os resultados obtidos demonstram a eficiência do sistema proposto para monitoramento de temperatura e umidade em tempo real. A partir das medições realizadas, os tempos de resposta do sensor, atuador e comunicação via MQTT foram registrados e avaliados. Conforme apresentado na Tabela 1, o tempo médio de resposta do sensor foi de 25 ms, do atuador 0 ms, e o tempo médio de envio de dados via MQTT foi de 2 ms. Esses tempos são altamente adequados para aplicações industriais, onde a agilidade na detecção de condições ambientais e na transmissão de dados é essencial para garantir a integridade de equipamentos e materiais.

Além disso, o acionamento dos LEDs e a publicação dos dados em tempo real demonstram a confiabilidade do sistema. O LED vermelho sinalizou condições fora do intervalo aceitável de temperatura, enquanto o LED verde indicou condições normais. Esses alertas locais complementam a monitoração remota, permitindo uma resposta visual imediata no local.

No contexto industrial, a integração com o protocolo MQTT foi fundamental para garantir a escalabilidade e eficiência do sistema. A transmissão de dados em tempo real para o dashboard do Adafruit IO permite uma visualização contínua e centralizada das informações coletadas, promovendo uma análise rápida e ações preditivas para evitar falhas operacionais.

Tabela de resultados:

Num. Medida	Sensor/Atuador	Tempo de Resposta(ms)
1	Envio via MQTT	2
2	Sensor (Leitura)	25
3	Atuador (LED)	0
4	Envio via MQTT	2
5	Sensor (Leitura)	25
6	Atuador (LED)	0
7	Envio via MQTT	2

Dash no Adafruit:

Monitoração Remota via Dashboard

A interface desenvolvida no Adafruit IO serve como uma ferramenta visual eficiente para o acompanhamento das condições monitoradas. O dashboard foi configurado com dois medidores (gauge) e dois gráficos de linha (line chart), exibindo as leituras de temperatura ambiente e umidade relativa em tempo real. A Figura 1 ilustra o dashboard em operação.

Descrição dos Elementos do Dashboard:

Medidores (Gauges):

- O medidor de temperatura apresenta os valores instantâneos em graus Celsius, com escala de 0 a 50°C. Ele inclui uma indicação visual clara, destacando valores críticos com mudança de cor.
- O medidor de umidade exibe os valores percentuais, com escala de 0 a 100%, proporcionando uma leitura direta do estado atual do ambiente monitorado.

Gráficos de Linha (Line Charts):

- Os gráficos de linha registram o histórico das leituras de temperatura e umidade ao longo do tempo. Essa funcionalidade é essencial para identificar tendências e mudanças nas condições ambientais.

A monitoração em tempo real no dashboard do Adafruit IO permite que os operadores tenham acesso imediato às condições do ambiente monitorado, mesmo à distância. Essa

visualização auxilia na tomada de decisão rápida e na implementação de ações corretivas ou preventivas, reduzindo possíveis riscos para equipamentos e processos.

Dashboard configurado no adafruit IO mostrando os dados de temperatura e umidade

5. Conclusões

Os objetivos propostos para este projeto foram alcançados com sucesso. O sistema demonstrou ser capaz de monitorar temperatura e umidade em tempo real, acionando LEDs locais para sinalização visual e enviando dados para a plataforma Adafruit IO via protocolo MQTT. A integração dos componentes de hardware e software garantiu o cumprimento dos requisitos, proporcionando uma solução funcional e escalável para aplicações industriais.

Os principais problemas enfrentados durante o desenvolvimento foram:

- Configuração da comunicação MQTT: Inicialmente, houve dificuldades para conectar o ESP32 ao broker MQTT, mas isso foi resolvido ao revisar as configurações de autenticação e corrigir inconsistências no código.
- Precisão do sensor DHT11: Apesar de ser acessível e simples, o DHT11 apresentou pequenas variações nas leituras. Isso foi mitigado realizando testes repetidos e ajustando os limites para os LEDs.

- Estabilidade da conexão Wi-Fi: Em ambientes com sinal de Wi-Fi instável, houve interrupções no envio de dados. Para resolver, foi implementada a reconexão automática no código.

As principais vantagens do projeto incluem:

- Simplicidade e custo baixo: O uso de componentes acessíveis e fáceis de configurar torna o sistema ideal para prototipagem.
- Conectividade em tempo real: O protocolo MQTT garante uma transmissão eficiente dos dados para o servidor remoto.
- Escalabilidade: A solução pode ser facilmente adaptada para outros sensores e atuadores.

Entre as desvantagens, destacam-se:

- A precisão limitada do sensor DHT11, que pode não ser adequada para aplicações de alta precisão.
- Dependência de uma conexão Wi-Fi estável para funcionamento contínuo.

Para melhorar o projeto, seria possível:

- Substituir o sensor DHT11 por sensores mais precisos, como o DHT22 ou BME280.
- Implementar redundância na comunicação, utilizando um protocolo alternativo em caso de falha no MQTT.
- Desenvolver uma interface web personalizada para monitoramento dos dados, aumentando a flexibilidade da solução.

Link para o repositório no github:

https://github.com/idribeir0/IOT_Project_Temperature_Sensor

Link para o vídeo no youtube:

https://youtu.be/wkVO98cpoe4

6. Referências

AYAZ, M. et al. Internet-of-Things (IoT)-Based Smart Agriculture: toward making the fields talk. IEEE Access, v. 7, p. 129551-129583, 1º ago. 2019. Disponível em: https://ieeexplore.ieee.org/document/8784034. Acesso em: 8 set. 2024.

SANTOS, B. et al. Internet das Coisas: da teoria à prática. Disponível em: https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf. Acesso em: 8 set. 2024.

IBM. Introducing MQTT: Why MQTT is the best messaging protocol for IoT. Disponível em: https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/. Acesso em: 8 set. 2024.

DHT11. Datasheet do sensor DHT11 – Sensor de temperatura e umidade. Disponível em: https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT11.pdf. Acesso em: 8 set. 2024.

HEATH, S. Embedded Systems Design. Disponível em: https://fdocuments.in/document/embedded-systems-design.html. Acesso em: 8 set. 2024.

HUNKAR, D. et al. Smart CEI Moncloa: An IoT-based Platform for People Flow and Environmental Monitoring on a Smart University Campus. Sensors, v. 17, n. 12, p. 2856, 2017. Disponível em: https://doi.org/10.3390/s17122856. Acesso em: 8 set. 2024.

TUTORIALPOINT. Embedded Systems - Basic Concepts. Disponível em: https://www.tutorialspoint.com/embedded_systems/es_processors.htm. Acesso em: 8 set. 2024.

MQTT. MQTT Protocol. Disponível em: http://mqtt.org/. Acesso em: 8 set. 2024.