

MINISTÉRIO DA DEFESA
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
Seção de Engenharia Elétrica e de Engenharia da Comunicação (SE/3)

PROJETO DE SISTEMAS EMBARCADO

Sensor de Temperatura e Umidade

Cap Renan Gelatti Chalegre
1º Ten Djalma Teixeira dos Santos Junior

Rio de Janeiro, RJ
Maio de 2025

1 Introdução

1.1 Objetivos

Este trabalho teve como objetivo implementar e operacionalizar, através da plataforma ESP32 e o sensor DHT11, um sistema de sensoriamento de temperatura e umidade, realizando o envio automatizado dos dados capturados para um canal criado na plataforma ThingSpeak, com visualização gráfica em tempo real.

1.2 Introdução Teórica

O sensor DHT11 é um dispositivo de medição de temperatura e umidade amplamente utilizado devido à sua simplicidade e custo acessível. Internamente, conta com um termistor para temperatura e um sensor capacitivo para umidade, entregando leituras digitais entre 0 a 50°C, com precisão de $\pm 2^\circ\text{C}$, e umidade de 20 a 90%, com precisão de $\pm 5\%$.

Para processar e transmitir essas informações, utilizou-se o microcontrolador ESP32 DEV KIT V1, que integra conexão Wi-Fi, capacidade de processamento robusta e suporte a diversas bibliotecas para aplicações de Internet das Coisas (IoT).

1.3 Configuração da Conta no ThingSpeak

Foi criada uma conta na plataforma ThingSpeak para configurar um canal exclusivo de recebimento de dados. O canal foi parametrizado com dois campos: temperatura e umidade. A chave de API gerada foi inserida no código do ESP32, garantindo a autenticação segura e o correto encaminhamento dos dados.

2 Desenvolvimento

2.1 Esquemático do Projeto

O esquemático do circuito foi elaborado utilizando a ferramenta CAD **EasyEDA**, conforme recomendado. A diagramação inclui:

- Conexão do pino de sinal do DHT11 ao GPIO4 (D4) do ESP32.
- Alimentação do sensor com 5V e GND.
- ESP32 alimentado via porta micro-USB.

2.2 Montagem do Protótipo em Protoboard

A montagem do circuito foi realizada em protoboard, utilizando fios do tipo “fêmea-fêmea” para facilitar as conexões. Esta montagem permitiu ajustes rápidos e validação do sistema.

2.3 Código-Fonte e Implementação

O código foi desenvolvido na plataforma Arduino IDE, utilizando as bibliotecas `WiFi.h`, `DHT.h` e `ThingSpeak.h`. Abaixo, apresentamos os principais trechos, devidamente comentados:

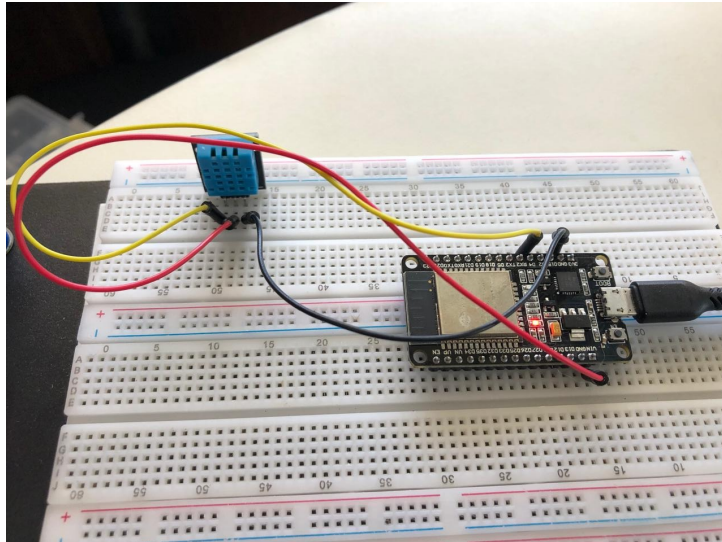


Figura 1: Protótipo montado com ESP32 e sensor DHT11

```
// Inclusão das bibliotecas
#include <WiFi.h>
#include "DHT.h"
#include "ThingSpeak.h"

// Configuração da rede Wi-Fi
const char* ssid = "IME-SE3-MICROCON";
const char* password = "ime04871";

// Configuração do ThingSpeak
const char* apiKey = "IG6EGDGD0IOZAFWE";
const unsigned long channelID = 2960853;

// Inicialização do sensor DHT11
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// Objeto WiFiClient
WiFiClient client;

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  // Aguarda conexão
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Conectando ao WiFi...");
  }
}
```

```

    Serial.println("Conectado!");
    ThingSpeak.begin(client);
    dht.begin();
}

void loop() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    if (!isnan(h) && !isnan(t)) {
        ThingSpeak.setField(1, t);
        ThingSpeak.setField(2, h);
        int x = ThingSpeak.writeFields(channelID, apiKey);

        if (x == 200) {
            Serial.println("Dados enviados com sucesso!");
        } else {
            Serial.println("Erro ao enviar dados: " + String(x));
        }
    }

    delay(15000); // Envia a cada 15 segundos
}

```

2.4 Desafios Enfrentados e Soluções

Durante a implementação do projeto, foram encontrados alguns desafios:

- **Problemas na leitura do sensor:** Foram observados valores de "NaN" (Not a Number) nas primeiras leituras. A solução foi aguardar alguns segundos após a inicialização do sensor para estabilização.
- **Instabilidade na conexão Wi-Fi:** Foi implementada uma rotina de reconexão automática caso a conexão fosse perdida.
- **Limitação de tempo de envio:** O intervalo mínimo entre envios foi ajustado para 15 segundos, conforme as boas práticas e limitações do ThingSpeak.

2.5 Projeto da Placa de Circuito Impresso (PCB)

A figura abaixo apresenta o layout do projeto de Placa de Circuito Impresso (PCI) desenvolvido na plataforma EasyEDA. A placa foi projetada para realizar a leitura de dados de um sensor de umidade e temperatura, integrando-se a um microcontrolador ESP32, responsável pela coleta, processamento e envio das informações para uma aplicação externa.

O roteamento das trilhas foi feito de forma a garantir um bom desempenho do sinal e uma alimentação estável para os componentes. O sensor foi posicionado estrategicamente na PCI, permitindo uma leitura precisa do ambiente ao redor. O ESP32 foi incluído como

principal unidade de controle, devido à sua versatilidade, conectividade Wi-Fi integrada e compatibilidade com plataformas de automação e IoT.

O uso da ferramenta EasyEDA facilitou o processo de criação, simulação e visualização do circuito em 2D e 3D, além de permitir a geração dos arquivos Gerber necessários para a produção física da placa.

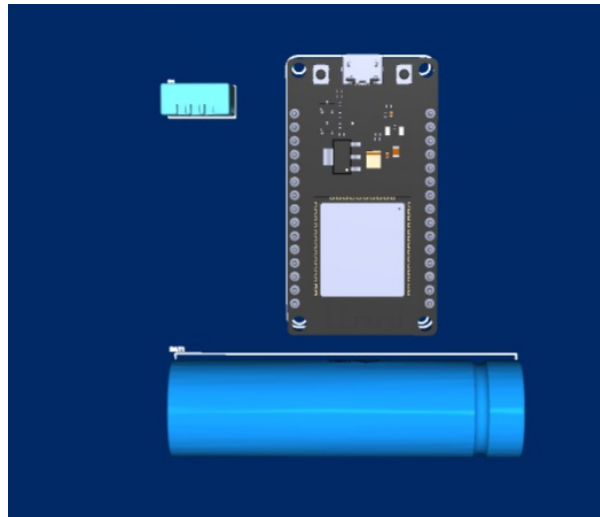


Figura 2: Visão frontal da placa

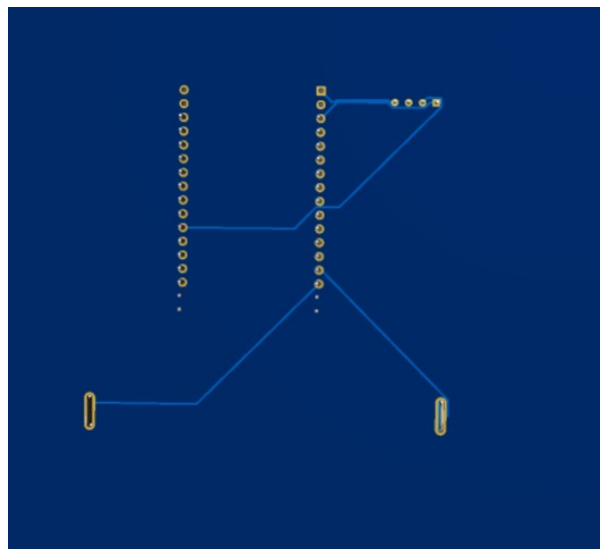


Figura 3: Visão de trás da placa

3 Resultados

Após o carregamento do código e a estabilização da conexão Wi-Fi, o ESP32 passou a realizar as leituras periódicas e enviá-las automaticamente para o ThingSpeak. A figura a seguir mostra a interface gráfica com os dados coletados:

As medições demonstraram a eficiência e estabilidade do sistema, com dados coerentes com as condições ambientais do local de teste.

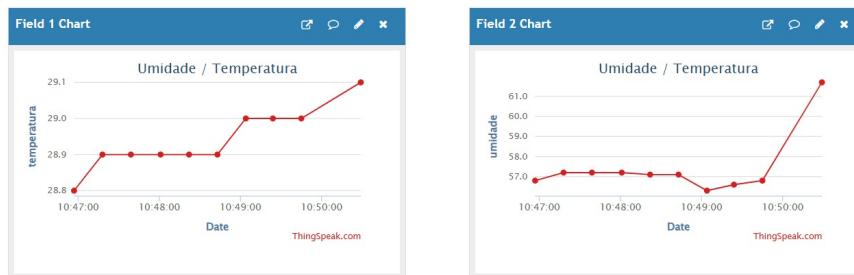


Figura 4: Gráficos de temperatura e umidade no ThingSpeak

4 Conclusão

O desenvolvimento deste projeto atendeu integralmente aos requisitos propostos. A solução apresentou-se eficaz, com potencial para aplicações reais de monitoramento ambiental.

O uso de ferramentas como EasyEDA para esquemático e PCB facilitou a documentação e permitiu a projeção de um produto final mais profissional.

Além disso, o trabalho contribuiu significativamente para o aprofundamento em práticas de sistemas embarcados, redes de comunicação e IoT.