

SISTEMA DE REFRIGERAÇÃO COM ESP32

Fernando Lucas Santos Cordeiro

Guskov da Silva Coelho

João Manoel de Souza Muniz

Marcos Corrêa Nunes

Marcos Vinícios da Silva Limas

Paulo Gabriel Sodré Meneses

RESUMO: O presente artigo visa evidenciar a criação e funcionamento do sistema de refrigeração que utiliza a placa microcontroladora ESP32 e atua juntamente com o sensor de temperatura DHT-11 para controle de ventoinhas que conduzem a corrente de refrigeração de um ambiente para o outro. A temperatura é visualizada e monitorada através de um display LCD, e todo este sistema é controlado remotamente pela plataforma Blynk.

Palavra-Chave: sistema de refrigeração, placa microcontroladora ESP32, temperatura, controle remoto.

ABSTRACT: This article aims to highlight the creation and operation of a refrigeration system that uses the ESP32 microcontroller board and works together with the DHT-11 temperature sensor to control fans that direct the cooling current from one environment to another. The temperature is visualized and monitored through an LCD display, and the entire system is remotely controlled via the Blynk platform.

Keywords: refrigeration system, ESP32 microcontroller board, temperature, remote control.

¹ Artigo elaborado para a disciplina Instrumentação Científica do Curso de Engenharia da Computação do Centro Universitário do Maranhão como requisito para obtenção de nota sob a orientação do prof. Madson Cruz Machado.

² Discentes do 7º período de Engenharia da Computação do Centro Universitário do Maranhão.

1. INTRODUÇÃO

Diante do cenário atual, torna-se cada vez maior a presença da automação em sistemas de controle de temperatura, principalmente para aplicações que exigem um alto grau de precisão e eficiência energética. Dessa forma, foi desenvolvido um projeto inovador de um sistema de refrigeração automático que utiliza a placa microcontroladora ESP32 para o controle da temperatura de um ambiente. Para o acionamento das ventoinhas, o sistema é integrado com o sensor de temperatura DHT11, com o qual a temperatura interna do ambiente é monitorada continuamente. Quando a temperatura medida atinge um valor predefinido, as ventoinhas são automaticamente acionadas para resfriar o ambiente interno, garantindo a manutenção de condições ideais de conservação.

Além do controle automatizado, o sistema terá a opção de controle remoto através da plataforma Blynk. O sistema permite que os usuários monitorem e ajustem a temperatura ideal em tempo real de qualquer lugar que tenha acesso à internet, proporcionando assim, um melhor controle e usabilidade do sistema. Desta forma, ao longo deste artigo, será discutido em detalhes o desenvolvimento e funcionalidade do sistema, desde a ideia inicial até a implementação final. Isso incluirá os componentes, a coordenação de hardware e software e os testes reais para determinar a eficácia do sistema.

1. METODOLOGIA

A metodologia adotada para o desenvolvimento do sistema de refrigeração automatizado utilizando a placa microcontroladora ESP32 foi estruturada em várias etapas, desde o planejamento e a concepção do projeto até a implementação e os testes finais.

Planejamento e Conceito do Projeto:

Revisão bibliográfica para identificar componentes adequados.

Escolha do ESP32 pela sua capacidade de processamento e conectividade Wi-Fi.

Seleção do sensor DHT11 por sua simplicidade e precisão.

Design do Sistema:

Elaboração do projeto em modelagem 3D para análise prévia.

Componentes principais: ESP32, DHT11, ventoinhas, display LCD, plataforma Blynk.

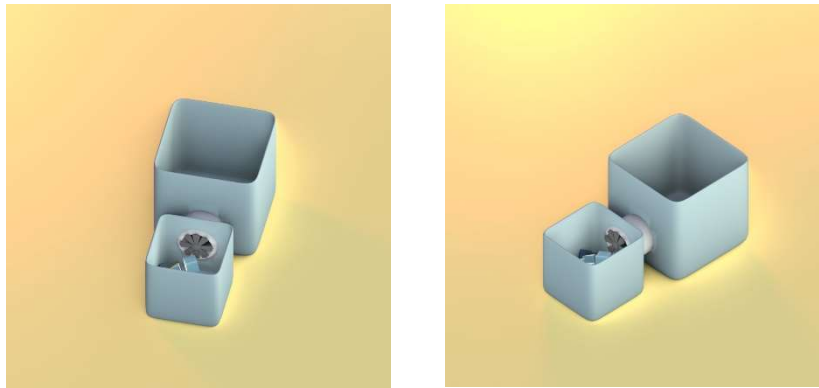


Figura 1 e 2– Modelo 3D do projeto

Integração de Hardware

Conexão do DHT11 ao ESP32.

Conexão das ventoinhas ao ESP32 pelas portas analógicas.

Conexão do display LCD ao ESP32.

Configuração do módulo Wi-Fi do ESP32 para integração com Blynk.

Desenvolvimento de Software

Desenvolvimento do código na IDE do Arduino.

Módulos de leitura do sensor, controle das ventoinhas, interface com o display e conectividade com Blynk.

```
// Definindo pinos e tipo do sensor DHT
const int DHTPIN = 4;
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE); // Inicializando o sensor DHT

// Endereço I2C do display OLED
#define i2c_Address 0x3c

// Definindo as dimensões do display OLED
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SH1106G display = Adafruit_SH1106G(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

//Pin das Ventoinhas
const int fanPin =16;
const int fanPinTwo =17;

// Variáveis para armazenar o estado das ventoinhas
```

```
bool fanPinState = LOW;
bool fanPinTwoState = LOW;

BLYNK_WRITE(V16){
    fanPinState = param.asInt();
    digitalWrite(fanPin,param.asInt());
}
BLYNK_WRITE(V17){
    fanPinTwoState = param.asInt();
    digitalWrite(fanPinTwo,param.asInt());
}
void setup() {
    // Inicializa a comunicação serial
    Serial.begin(9600);

    // Inicializa o Blynk
    Blynk.begin(auth, ssid, pass,"blynk.cloud",80);

    // Inicializa o sensor DHT
    dht.begin();

    // Inicializa o display com o endereço I2C correto
    if(!display.begin(i2c_Address, true)) { // Endereço 0x3C por padrão
        Serial.println(F("Falha na alocação do SSD1306"));
        while(true); // Não prossegue, fica em loop infinito
    }

    // Limpa o buffer do display
    display.clearDisplay();

    // Configura o tamanho e a cor do texto
    display.setTextSize(1);
    display.setTextColor(SH110X_WHITE);

    // Inicializa o pino como saída
    pinMode(fanPin, OUTPUT);
    pinMode(fanPinTwo, OUTPUT);
}

// Função para obter a temperatura
float getTemperature() {
    return dht.readTemperature();
}

// Função para atualizar o display LCD
void updateDisplay() {
    delay(100); // Espera o OLED ligar
```

```

// Limpa o buffer do display
display.clearDisplay();

// Configura a posição do cursor
display.setCursor(0, 10);

// Obtém a temperatura
float temperature = getTemperature();

// Imprime a temperatura no display
display.print(F("Temperatura: "));
display.print(temperature);
display.println(F(" C"));

// Envia os valores para o Blynk
Blynk.virtualWrite(V4, temperature);

// Mostra o buffer do display na tela
display.display();
}

//funcao controladora da ventoinha
void fanController(){
    if(getTemperature() <= 30){
        fanOn();
    }else{fanOff(); }
}

// Liga a ventoinha (GPIO HIGH)
void fanOn(){
    digitalWrite(fanPin, HIGH);
    digitalWrite(fanPinTwo, HIGH);
}

// Desliga a ventoinha (GPIO LOW)
void fanOff(){
    digitalWrite(fanPin, LOW);
    digitalWrite(fanPinTwo, LOW);
}

// Função principal de loop
void loop() {
    Blynk.run();
    // Atualiza o display LCD continuamente
    updateDisplay();
    //funcao controladora da ventoinha
    if (fanPinState == LOW && fanPinTwoState == LOW) {fanController();}
}

```

Testes e Validação

Testes integrados do sistema completo em condições reais.

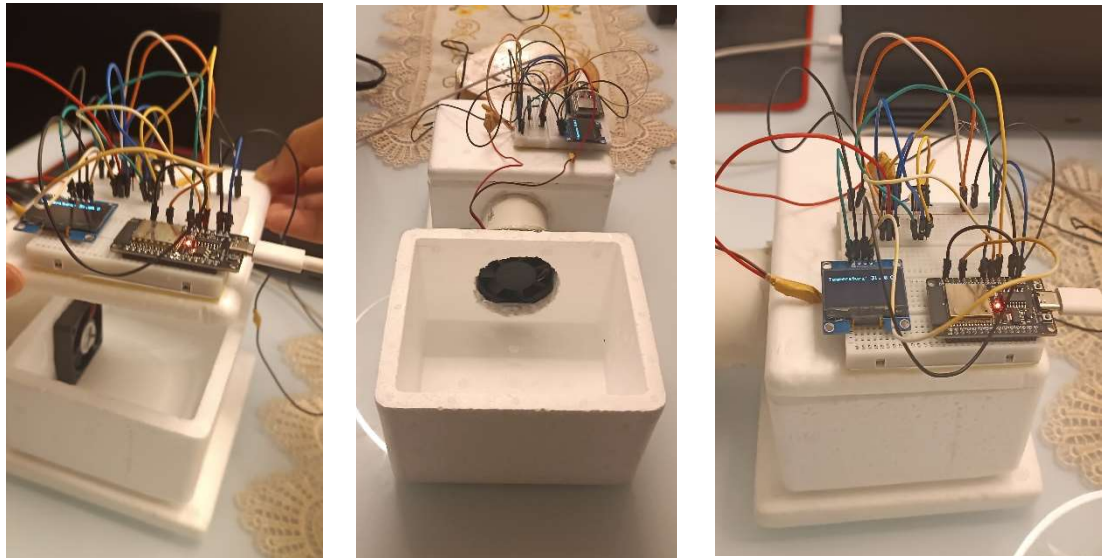


Figura 3,4 e 5– Testes finais do projeto

2. RESULTADOS

Os testes realizados demonstraram que o sistema de refrigeração automatizado funciona conforme o esperado, mantendo a temperatura interna da caixa de isopor dentro dos parâmetros predefinidos. A integração com a plataforma Blynk permitiu um controle remoto eficiente, possibilitando monitoramento e ajustes em tempo real.

O sistema mostrou-se eficiente na manutenção da temperatura desejada, acionando as ventoinhas de maneira responsiva quando necessário. A utilização do ESP32 e do DHT11 se revelou adequada para essa aplicação, proporcionando uma solução de baixo custo e alta funcionalidade.

Algumas limitações foram identificadas, como a necessidade de otimização do consumo de energia das ventoinhas e a melhoria na precisão de leitura do sensor DHT11 em condições extremas.

3. Considerações Finais

O projeto desenvolvido resultou em um sistema de refrigeração automatizado funcional e eficiente, capaz de manter a temperatura de uma caixa de isopor dentro dos parâmetros desejados. A integração com a plataforma Blynk adicionou uma camada de flexibilidade e

conveniência, permitindo o controle remoto do sistema. Este projeto não apenas demonstrou a viabilidade técnica da criação de um sistema de controle de temperatura eficaz para uma caixa de isopor, mas também destacou a integração de várias tecnologias modernas, incluindo sensores, atuadores e plataformas de controle remoto, como a Blynk.

Os resultados obtidos durante os testes confirmam que o sistema é capaz de manter a temperatura interna da caixa de isopor dentro dos parâmetros desejados de forma autônoma e eficiente. A capacidade de monitoramento e controle remoto via Blynk adiciona uma camada significativa de conveniência e flexibilidade, permitindo que usuários ajustem e monitorem a temperatura em tempo real a partir de qualquer lugar.

Além dos benefícios práticos, o projeto ofereceu uma oportunidade valiosa para os alunos aplicarem conhecimentos teóricos em um contexto real, enfrentando desafios e desenvolvendo soluções inovadoras. A escolha do ESP32 se mostrou particularmente acertada devido à sua versatilidade e capacidades de conectividade, enquanto o sensor DHT11, apesar de suas limitações em condições extremas, provou ser adequado para a maioria das situações.

REFERÊNCIAS

THOMSEN, Adilson. Monitorando Temperatura e Umidade com o sensor DHT11. Maker Hero, 2013. Disponível em: <https://www.makerhero.com/blog/monitorando-temperatura-e-umidade-com-o-sensor-dht11/>. Acesso em: 01 de junho de 2024.

SANTOS, Felipe. Aprenda a configurar a rede WiFi do ESP32 pelo smartphone. Maker Hero, 2020. Disponível em: <https://www.makerhero.com/blog/aprenda-a-configurar-a-rede-wifi-do-esp32-pelo-smartphone/>. Acesso em: 01 de junho de 2024.

BRINCANDO COM O ESP32 NO ARDUINO IDE. Tudo sobre IOT, 2022. Disponível em: https://www.tudosobreiot.com.br/blog/1098-iot-feito-facil_-brincando-com-o-esp32-no-arduino-ide. Acesso em: 01 de junho de 2024.

FREITAS, Gabriel. Blynk IoT – Monitoramento com ESP8266 e Sensor de Umidade do Solo (Higrômetro). Smart Kits, 2021. Disponível em: <https://blog.smartkits.com.br/blynk-iot-monitoramento-com-esp8266-e-sensor-de-umidade-do-solo-higrometro/>. Acesso em: 01 de junho de 2024.