

Monitoramento sem fio de ambientes climatizados

Andrew Lucas de Abreu Nascimento

Mateus Ladika

Orientador: Júlio Ferreira

RESUMO

Resumo—Este artigo tem por objetivo estudar e desenvolver um sistema de monitoramento de temperatura para ambientes climatizados. Utilizando microcontroladores Arduíno ESP32 e sensores de temperatura DHT22, o principal foco do projeto é projetar um sistema autónomo e de baixo custo, podendo monitorar as variações de temperatura em determinadas áreas.

Com base nos dados coletados pelos protótipos, concluímos que o ESP32 é uma ótima escolha para essa finalidade, porém existem modelos diferentes que podem desempenhar ainda melhor na questão de economia de energia e eficiência. Nos últimos testes realizados, ambos os dispositivos conseguiram superar oitenta horas ininterruptas de funcionamento.

Index Terms—DHT22, Arduino, ESP32, Custo.

Introdução

Hoje, a climatização se tornou uma necessidade em diversos ramos comerciais, desde pequenos empreendimentos até centros de armazenagem e distribuição, sejam estes de medicamentos ou alimentos. Porém, diversos locais não contam com um sistema de monitoramento dessa climatização e não controlam a temperatura do ambiente com precisão.

Como exemplo, temos uma notícia de outubro de 2020, publicada pelo G1 que demonstra uma fração desse problema.

“cerca de 800 doses de vacina contra febre aftosa impróprias para uso foram apreendidas durante inspeção realizada nos municípios de Abaetetuba e Tailândia, nordeste do Pará.”

“Em uma das revendas vistoriadas, foram encontradas vacinas mantidas em geladeira comum, de forma inadequada e sem termômetro para registro de temperaturas máxima e mínima.”

“As vacinas contra a febre aftosa devem ser armazenadas em temperatura entre 2°C e 8°C, em refrigerador com termômetro aprovado pelo Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro).”

Avaliando essa situação e outros casos similares, o objetivo desse artigo é automatizar este processo, coletando os dados e enviando para uma plataforma em nuvem, onde as informações coletadas podem ser consultadas a qualquer momento.

Referencial teórico

1. IOT (Internet das Coisas)

É o conceito da integração de vários tipos de equipamentos como por exemplo sensores, carros, smartwath. Funcionando como suporte para esse tipo de tecnologia. São dispositivos que tem como função coletar vários tipos de dados e se comunicar entre si de forma autônoma utilizando a internet. Esse termo nasceu em 1999 proposto por Kevin Ashton do MIT (Instituto de tecnologia de Massachusetts).

Com o avanço da internet 5G que é uma banda larga móvel sem fio com alta velocidade de comunicação, juntamente com o avanço da indústria com equipamentos de alta tecnologia embarcada colaboram para a maior eficiência da internet das coisas.

Vários tipos de equipamentos podem ser integrados a esse tipo de conceito. Porém não são todos que fazem parte pois dependem de alguns fatores como uma fonte de energia para alimentar o equipamento, uma bateria por exemplo. Uma unidade de comunicação para habilitar a comunicação com outras tecnologias. Microprocessador ou microcontrolador que gerencia o funcionamento do equipamento e pode processar alguns dos dados obtidos, a coleta desses dados fica de responsabilidade dos sensores que podem ser de proximidade, pressão, umidade e temperatura entre outros. Alguns desses equipamentos só coletam dados e transmite para outro equipamento ou dashboard.

Esse tipo de conceito ainda tem alguns pontos fracos, um deles é a baixa segurança devido a comunicação entre vários tipos de equipamento ser muito fácil e de amplo acesso tanto a informações importantes e equipamentos, se tornando alvo de malwares e criminosos.

Depende também do avanço da tecnologia e do investimento em equipamentos.

Outro aspecto desse tipo de tecnologia ter tanto acesso a informações pessoais dos usuários não é tão bem aceito por todos.

2. Cibersegurança

Com o avanço da indústria 4.0 o conceito de cibersegurança se torna indispensável e se baseia em práticas de proteção de todo o universo digital contra ameaças externas. Esse tipo de ameaça ataca principalmente servidores, equipamentos, computadores e outros que podem ser explorados por sua vulnerabilidade.

Um exemplo dessas ameaças são os malware que significa software malicioso, uma das ameaças mais comuns presentes na atualidade pois pode ser acessada sem o usuário perceber que está sendo invadido. Esse tipo de software é desenvolvido por hackers em sua maioria com o objetivo de danificar ou roubar dados privados.

Ransomware é outro tipo de ataque que rouba dados do usuário e exige resgate por meio de criptomoedas que são um tipo de cibermoeda que possui alto nível de criptografia.

O terceiro exemplo de ataque mais comum é o Phishing que consiste na tentativa de obter informações pessoais das vítimas por meio de e-mails e sites falsos.

A cibersegurança conta com algumas ferramentas contra esse tipo de ameaça como os antivírus que atuam na máquina escaneando e combatendo frequentemente vírus e malwares.

Firewall é outro exemplo e funciona como uma parede que restringe o acesso externo da internet com a rede privada, esse tipo de defesa decide se o acesso é perigoso ou não.

3. Cloud Computing

Computação em nuvem é um tipo de tecnologia de armazenamento e processamento de dados na nuvem em vez de servidores locais, tanto para empresas como para pessoas físicas. Os requisitos para a utilização desse serviço é uma boa conexão com a internet.

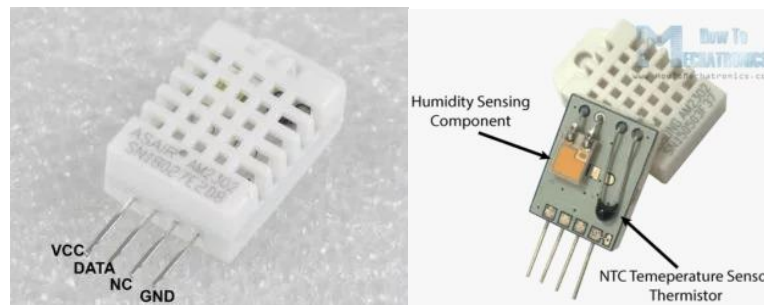
Esse ciberespaço ainda depende de uma grande estrutura para armazenar todos esses dados, para realizar esse serviço entram as grandes empresas que possuem uma imensurável capacidade de armazenamento e processamento que podem estar localizadas em qualquer parte do mundo. Esse tipo de armazenamento possui várias vantagens como possuir a função de backup e recuperação de dados não tendo a necessidade de o usuário ter custos com uma infraestrutura para armazenamento e de manutenção próprio.

MATERIAIS E MÉTODOS

1. MATERIAIS

A. DHT 22

O DHT 22 foi escolhido para esse projeto devido a seu custo e as suas especificações técnicas, as quais podemos encontrar características como sua faixa de leitura de umidade relativa do ar, que varia de 0 a 100% com resolução de até 0,1% e precisão de 2% e a faixa de leitura de temperatura, que varia de -40° a 80 °C, com resolução de 0,1 °C e precisão de aproximadamente 1% para mais ou para menos. O tempo de leitura é geralmente em torno de 250 milissegundos e pode ser alimentado com uma tensão de 3,3 a 5 volts. Seu consumo máximo de corrente é de 0,5mA durante a leitura e é importante utilizar um resistor pullup que possua de 4,7 a 10 kΩ de resistência elétrica.



B. ESP32-WROOM-32

O ESP32 é uma placa de desenvolvimento que possui Wi-Fi e Bluetooth integrados, utilizado para uma ampla área de aplicações. Seu design, segundo o site do fabricante, é robusto e pode suportar ambientes com temperaturas que variam de -40° a 85°C dependendo do modelo, além de possuir um regulador de tensão de entrada que corrige valores de 5v até 12v para uma estável tensão de 3,3v. Como é desenvolvido para plataformas móveis e IoT, possui um baixíssimo consumo de energia, além de dispor de diversas configurações que melhoram ainda mais a sua economia.

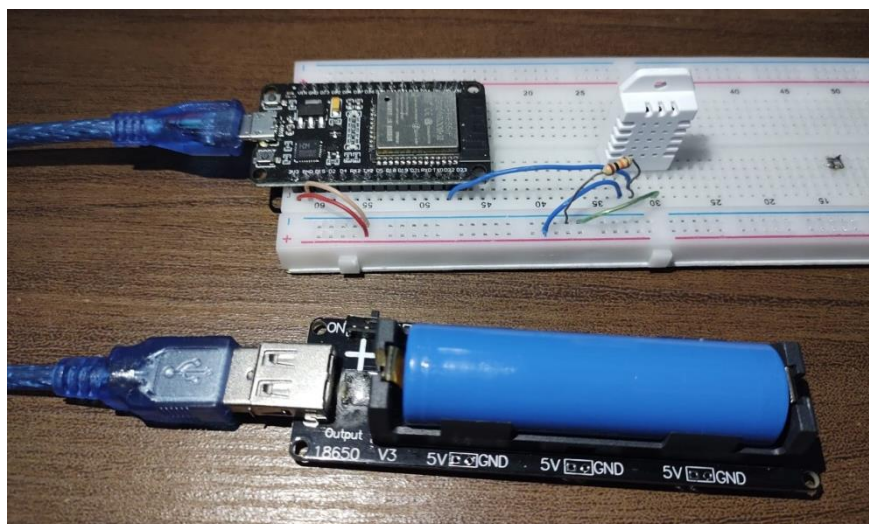
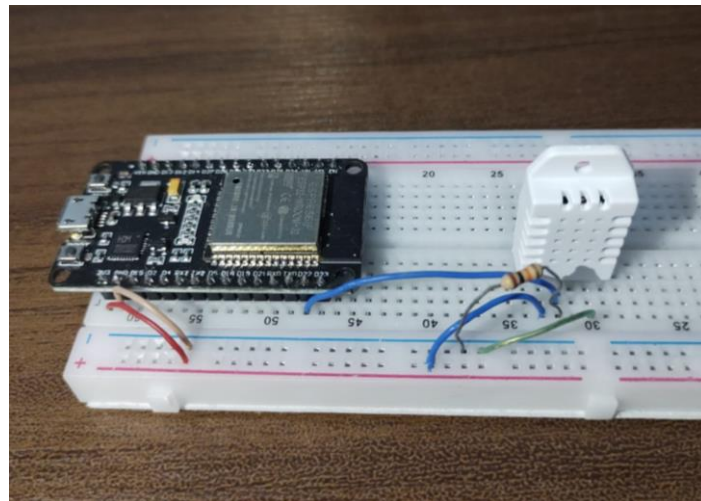
As especificações internas do modelo utilizado no projeto contam com um chip base ESP32-D0WDQ6 e um processador Xtensa 32bit Dual Core, com 520kb de memória SRAM.



2. MÉTODOS

A. Procedimentos:

Foi montado em uma protoboard o ESP32 e o sensor DHT22 e utilizados “jumpers” para as conexões de alimentação e sinal enviados do sensor para o controlador. Já o ESP32, foi conectado via cabo USB à bateria modelo 18650 com capacidade de 5800mA/h e aproximadamente 3,7V de alimentação, conectada a um módulo Shield com saída USB e cabo USB – micro USB, além de conter um circuito de proteção contra sobrecarga e subcarga.



Em seguida, utilizamos a Arduino Cloud, que é uma plataforma fornecida pela própria empresa Arduino, onde podemos criar, controlar e gerenciar as aplicações utilizadas nos projetos.

Para esse artigo, criamos duas “things” como chamado pela plataforma, que na prática cada um representa um dos ESP32 utilizados, e para cada “thing” criamos as variáveis utilizadas para leitura e envio dos dados à nuvem. Depois, geramos uma ID e uma senha de acesso para cada ESP, que serão validados no momento do envio dos dados e em seguida foi configurada a rede WiFi utilizada para envio.

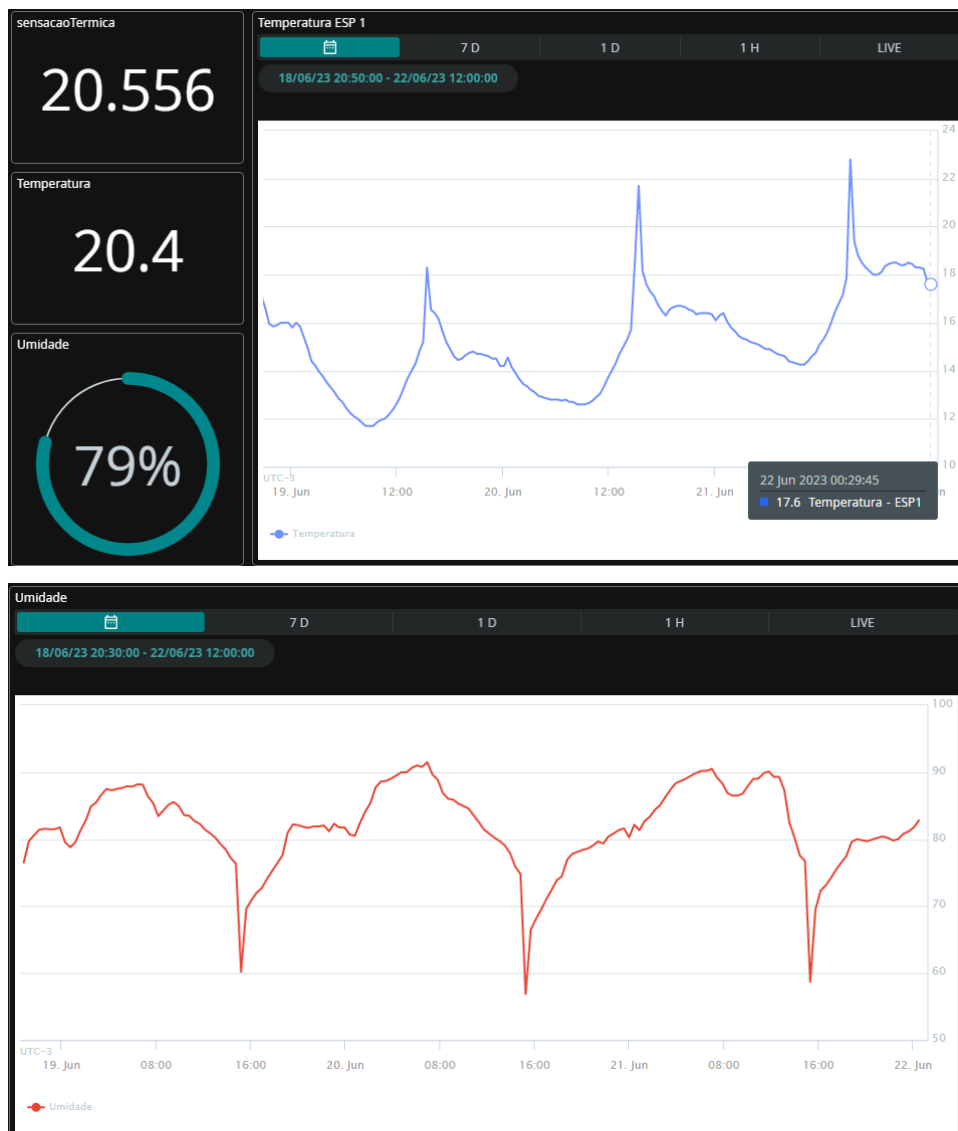
Com essas configurações, a Arduino Cloud gera um código em C++ com essa parametrização inicial e configuração do dispositivo para comunicação.

B. Medição e envio de dados:

Quando alimentado, o ESP32 inicializará e executará as rotinas de conexão com a internet e com a Arduino Cloud. Quando a conexão é confirmada, o ESP solicita ao sensor para realizar a leitura dos dados, e os envia para a nuvem. Em seguida, o ESP entra no modo Deep Sleep, onde apenas o Coprocessador ULP e o sistema RTC são mantidos ligados. Através do relógio interno do sistema RTC, o ESP32 calculará quanto tempo deve ficar no modo de baixo consumo até que seja necessária uma nova leitura do sensor, tempo esse que é configurado diretamente no código do ESP.

A Arduino Cloud conta com um dashboard online para armazenar os dados coletados e apresentá-los graficamente, e com base nos dados coletados nessa última versão do código, foi gerado os gráficos abaixo:

CENTRO UNIVERSITÁRIO UNICURITIBA



RESULTADOS E DISCUSSÃO

Foram necessários alguns testes e atualizações do código para melhorar mais a eficiência da bateria, totalizando até o presente momento, cinco versões diferentes com vários testes para cada uma.

Na primeira versão do código a leitura era realizada e os dados enviados para a nuvem a cada cinco minutos, além de não contar com um gerenciamento de energia mais complexo, fazendo com que a bateria durasse menos de duas horas nesse teste.

Para a segunda versão do código, foi adicionada a rotina Deep Sleep do ESP32 e ajustado o intervalo de leituras para quinze minutos. Com essa

configuração o projeto chegou a durar vinte e uma horas funcionando sem problemas.

A terceira versão teve pequenas alterações no código, reduzindo alguns “delays” e alterando o intervalo de leitura para trinta minutos, resultando em aproximadamente trinta e quatro horas de bateria.

A versão número quatro teve diversas melhorias em relação à anterior, sendo as principais alterações referentes à quantidade de leituras realizadas quando o ESP era reativado, e a redução do clock do processador para o mínimo disponível para esse modelo, que é de 80MHz. Essa atualização fez com que a bateria durasse cinquenta e nove horas aproximadamente.

Para a versão cinco, foram consultados alguns fóruns com outros utilizadores de dispositivos ESP32 onde foram relatados alguns “bugs” no processo de Deep Sleep do dispositivo. Segundo os usuários, randomicamente o ESP poderia não desligar os módulos de WiFi e Bluetooth, fazendo com que eles ficassem ligados mesmo no modo de economia de energia e acabassem consumindo muito mais corrente que o necessário. Para evitar esse tipo de problema, foi adicionada na rotina de Deep Sleep o desligamento manual dos módulos de WiFi, Bluetooth e conversores AD internos do ESP além da desativação da memória RTC que por padrão também ficaria ativada mesmo no modo de Deep Sleep. Ambos os sensores conseguiram ultrapassar setenta e duas horas ativos, mas o sensor que faz a leitura a cada trinta minutos teve o melhor desempenho, chegando a oitenta e cinco horas de duração de bateria.

Vale ressaltar que nessa última versão há um pequeno atraso no intervalo de leitura do ESP32 devido ao fato de não ser contabilizado o tempo de conexão e leitura no temporizador interno do dispositivo, fazendo que seja perdido entre quinze e trinta segundos por leitura.

CONCLUSÃO

Com a análise dos dados coletados, concluímos que é possível sim incluir esse tipo de monitoramento em ambientes climatizados, e que o ESP32 utilizado é um microcontrolador ótimo para essa aplicação, porém, quando

pesquisávamos mais formas de reduzir o consumo do dispositivo, encontramos outras versões do ESP32 que melhoram ainda mais a eficiência da bateria, pois seu hardware é mais simples, consumindo menos energia para executar as funções, como por exemplo o ESP32-S2, que possui apenas um núcleo de processamento, e tecnologias mais modernas do coprocessador ULP, que administra ainda melhor a energia utilizada.

Uma sugestão de Estudo futuro seria melhorar ainda mais a eficiência de energia do projeto, alterando alguns componentes como a plataforma ESP32 para a ESP32-S2 como foi citada anteriormente, ou algum outro modelo de microcontrolador compatível. Também seria possível alterar a bateria para algum modelo mais moderno, ou diferente do utilizado.

Devido à versatilidade da plataforma, também é possível adaptar esse projeto para substituir ou incluir novos sensores, podendo monitorar ainda mais dados de diferentes tipos de ambientes.

Outra sugestão seria desenvolver um sistema de controle integrado ao monitoramento, para ter um gerenciamento melhor da temperatura do ambiente além do já desenvolvido monitoramento.

REFERENCIAS

- [1] A. Loureiro, J. Nogueira, L. Ruiz, R. Mini, E. Nakamura, C. Figueiredo, "Redes de Sensores Sem Fio" UFMG, Belo Horizonte.
- [2] Poonam, Y. Mulge, "Remote Temperature Monitoring Using LM35 sensor and Intimate Android user via C2DM Service ", Department of Computer Science and Engineering, PDM College of Engineering for Women, MDU, Rohtak, Junho 2013
- [3] M. Silva "REDE DE SENSORES SEM FIO DE BAIXO CUSTO PARA MONITORAMENTO AMBIENTAL", Unicamp, Campinas, 2013
- [4] J. Moraes "Conhecendo o ESP32", acessado em 02/12/2019, Disponível em <https://portal.vidadesilicio.com.br/conhecendo-o-esp32/>
- [5] "Adepará apreende 800 doses de vacina contra febre aftosa impróprias para uso", disponível em: <https://g1.globo.com/pa/para/noticia/2020/10/08/adepara->

[apreende-800-doses-de-vacina-contr-a-febre-aftosa-improprias-para-uso.gh.html](#).

Acesso em: 15 abr. 2023.

[6] MISCHIANI, R. “ESP32 practical power saving: manage WiFi and CPU – 1 - Renzo Mischianti”. Disponível em: <https://mischianti.org/2021/03/06/esp32-practical-power-saving-manage-wifi-and-cpu-1>. Acesso em: 21 mai. 2023.