

Relatório Sistemas Microprocessados

Alexandre Augusto, 15/0056940

Pedro Venzi, 20/0042696

¹Faculdade de Tecnologia - Universidade de Brasília (UnB)

ENE0058 - Laboratório de Sistemas Microprocessados

25 de julho de 2023

150056940@aluno.unb.br, 200042696@aluno.unb.br

1. Introdução

Durante o semestre foi possível aprender sobre o microcontrolador MSP430 e como o mesmo funciona. Além disso, foi ensinado como funcionam protocolos de comunicação como o I2C, UART e outros.

Um desses protocolos, o UART (Universal Asynchronous Receiver/Transmitter), funciona assincronamente usando duas linhas de dados e um terra. O pino TX de um dispositivo é conectado ao RX do outro e vice-versa, como na figura 1, para que se estabeleça comunicação, dispensando a necessidade de um clock.

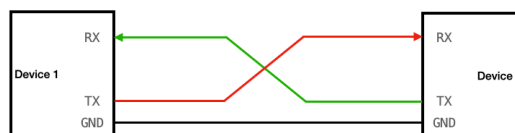


Figura 1. UART

Com o avanço da IoT (internet das coisas), dispositivos pequenos e de baixo consumo se tornaram cada dia mais essenciais, e um protocolo de mensagens leves se fez necessário para que a comunicação entre dispositivos ocorresse de forma a não sobrecarregar a rede dos usuários.

Para tanto, surgiu o MQTT, Message Queuing Telemetry Transport, que permitiu que vários sensores diferentes pudessem ser organizados de tal forma que os dados pudessem ser facilmente gerenciados. Usufruindo do padrão PubSub, em que canais no broker são usados para sensores publicarem e clientes se inscreverem, a estrutura do MQTT segue o padrão observado na figura 2. Um dos vários serviços disponíveis na internet para lidar com IoT é o [HiveMQ](#), que simplifica o gerenciamento dos dados e mantém os mesmos na nuvem.

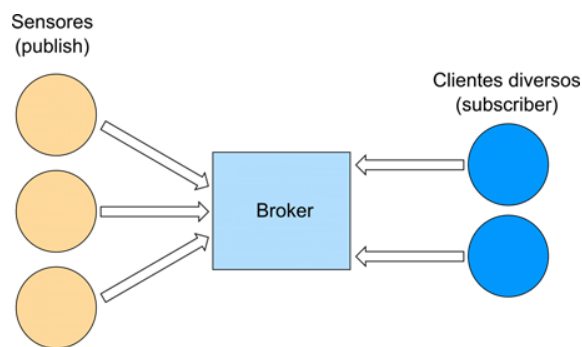


Figura 2. MQTT

1.1. Objetivos

Este projeto tem por objetivo implementar um sistema composto por um MSP430, um leitor de umidade e temperatura (DHT22) e um ESP8266 para que sejam feitas a leitura e transmissão via WiFi dos dados de um determinado recinto para um ambiente em nuvem que possibilite o acesso das informações em qualquer lugar. Desta forma, ao final do projeto teremos uma coleta de dados constante do clima com a capacidade de visualização os dados em qualquer dispositivo com acesso à internet.

1.2. Equipamentos

Neste projeto foram utilizados os seguintes equipamentos e softwares:

- [MSP430F5529](#) da Texas Instrument, um dispositivo barato, versátil e de baixo consumo elétrico.
- Sensor DHT22, um sensor de temperatura e umidade de baixo custo e alta precisão, podendo medir temperaturas de -40°C a 80°C, e umidade de 0% a 100%.
- Módulo ESP32, uma evolução do ESP8266, possuindo tanto conexão WiFi quanto Bluetooth 4.2. Apesar de ser possível usá-lo de forma independente, usou-se o mesmo atrelado ao MSP430 para fins educacionais.
- Protoboard, onde são fixados os dispositivos do sistema.
- Fios Conectores, usados para conectar o DHT e o ESP ao MSP430.
- [CodeComposer Studio](#) da TI para desenvolver o código usado no MSP430.
- [Arduino IDE](#) para desenvolver o código usado no ESP32.

2. Procedimentos e Resultados

Segue abaixo a descrição da montagem, desenvolvimento e seus resultados.

2.1. Montagem do equipamento

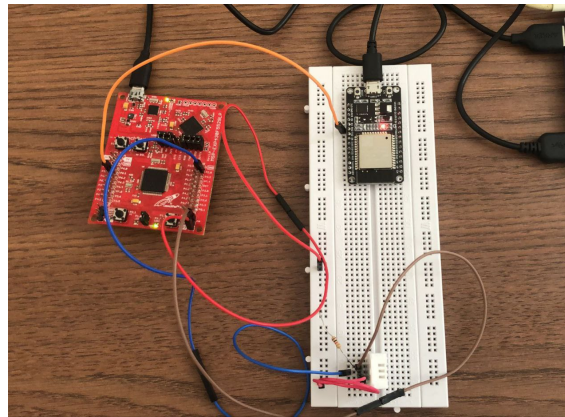


Figura 3. Implementação do projeto

Para este projeto, com sua montagem podendo ser observada na figura 3, foram definidas as seguintes portas para conexão dos dispositivos. O DHT22 foi conectado na P2.0, enquanto o ESP32 usou a P3.3 e P3.4 para a conexão via UART (TX e RX).

2.2. Desenvolvimento

Para que o sistema funcionasse de forma adequada, foi necessário instalar todos os softwares já listados, e ao programar foram feitos testes concomitantemente ao código para garantir que tudo desenvolvido estava funcionando corretamente. Iniciou-se por programar a captura de umidade e temperatura do DHT22, depois averiguou-se a conexão correta entre a rede WiFi e o ESP32, para logo em seguida implementar o envio dos dados do DHT para o ESP via MSP430, o que provou ser a parte mais complexa do sistema por envolver um padrão de comunicação, o UART, e necessitar de uma configuração correta e uso simultâneo de IDEs para averiguar a corretude dos dados transmitidos.

Uma vez implementado o sistema, foi necessário configurar a parte do HiveMQ, que, enquanto não necessária, é parte fundamental do projeto para torná-lo o mais próximo do encontrado no mundo real. Esta parte foi relativamente simples por estar bem documentada no próprio site do HiveMQ e existirem tutoriais mostrando como realizar a instalação e configuração do broker, assim como o uso da ferramenta nodeRED para manipulação e visualização dos dados como mostrado na figura 4.

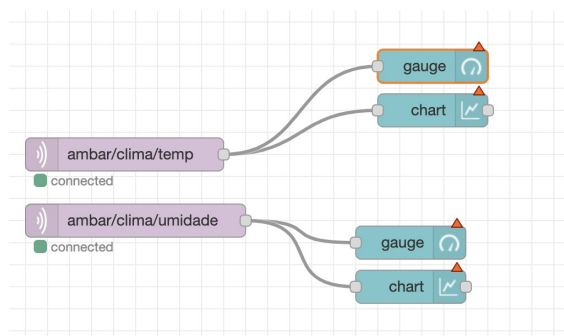


Figura 4. Manipulação visual dos dados no nodeRED

2.3. Resultado

Foi possível coletar dados de umidade e temperatura do ambiente a partir do DHT22 e enviar para um serviço externo, o HiveMQ, de forma consistente e precisa a nível de hardware. Já a nível de software, o serviço do HiveMQ apresentou instabilidades durante um dos dias do desenvolvimento do projeto mas mostrou-se estável nos outros dias.

O dashboard configurado no HiveMQ gerou um gráfico e possibilitou a averiguação dos dados de forma mais interativa, podendo o dashboard ser visualizado na figura 5.

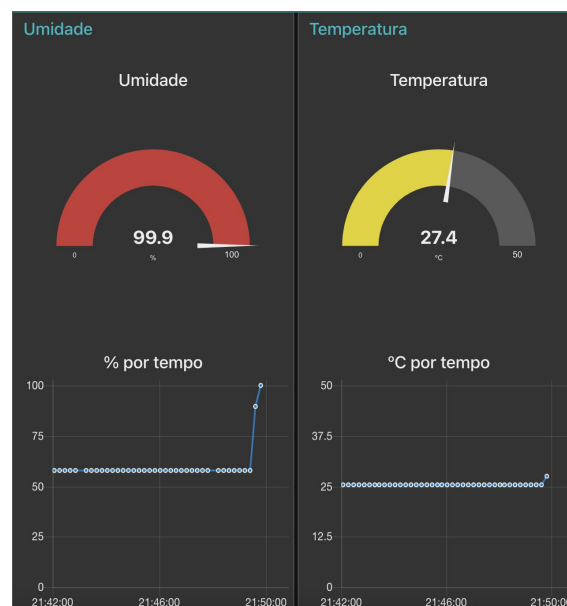


Figura 5. Dashboard do nodeRED

Apesar do uso de um serviço externo, foi implementada inicialmente uma API Rest em Django e um frontend em Vue que podem ser encontradas nos seguintes links para o [backend](#) e [frontend](#), que acabaram não sendo usados por ser mais fácil usar o HiveMQ tanto por conta de tempo, quanto para simplificar o projeto.

3. Conclusão

A partir do projeto realizado, foi possível compreender melhor e solidificar conhecimentos sobre comunicação entre dispositivos, UART e, apesar de estar fora do escopo do matéria, foi possível aprender sobre o mundo IoT e, de forma simplificada, como funciona o MQTT. Assim, conclui-se que o objetivo do projeto final foi alcançado e, salvo momentos de instabilidade de serviços externos, progrediu-se de forma adequada considerando o tempo à disposição e o objetivo proposto.