



Smart Home: economia e controle residencial através da internet

Pamela Taga, Wilian Franca Costa

Universidade Presbiteriana Mackenzie (UPM)

Rua da Consolação, 930 Consolação, São Paulo - SP, 01302-907 – Brazil

10919008181@mackenzista.com.br

Abstract. *This article describes the project that aims to demonstrate how the Internet of Things can be used in our day-to-day. Through the mqtt communication protocol, the user will be able to remotely control the lighting and have access to the temperature of their home or office.*

Resumo. *Este artigo descreve o projeto que tem como objetivo demonstrar como a Internet das Coisas pode ser utilizada no nosso dia a dia. Através do protocolo de comunicação MQTT, o usuário poderá controlar de forma remota a iluminação e ter acesso a temperatura de sua residência ou escritório.*

1. Introdução

Tecnologias para casas inteligentes não é algo novo, a primeira surgiu em 1975, uma plataforma de automação residencial – X10 – apta a enviar informações digitais por meio de rajadas de radiofrequência. Já em 1990, temos uma torradeira controlada pela internet. Com os avanços tecnológicos, a automação residencial deixou de ser algo para pessoas com alto poder aquisitivo e se tornou algo mais acessível. Um exemplo disso é o Google Home Smart Speaker que oferece recursos de controle remoto, proporcionando praticidade, segurança e sustentabilidade. Atualmente há diversas empresas que oferecem tecnologias para casa inteligente no Brasil, à Intelbras é uma delas. Importante lembrar que esse projeto não é utilizado apenas em residências mas também muito utilizado no setor rural, como por exemplo, medir e controlar temperatura em estufas.

Nesse artigo será mostrado um projeto onde o usuário poderá acender ou apagar uma lâmpada através do seu celular e verificar a temperatura daquele ambiente. É um projeto simples porém de grande importância, pois, através dele, podemos ter uma breve noção de como é a comunicação de objetos com a web através do protocolo MQTT.

2. Materiais e métodos

Para o desenvolvimento deste projeto, será utilizado os seguintes materiais:

- 01 – Placa ESP8266;
- 01 – Cabo USB;
- 01 – Protoboard 400 pontos;
- 04 – Jumpers Macho-Fêmea;
- 01 – Jumper Macho-Macho;
- 01 – Sensor DHT22
- 01 – Led;
- 01 – Resistor 220 ohms

Abaixo está o link da imagem dos componentes obtidas pelo datasheet

ESP 8266

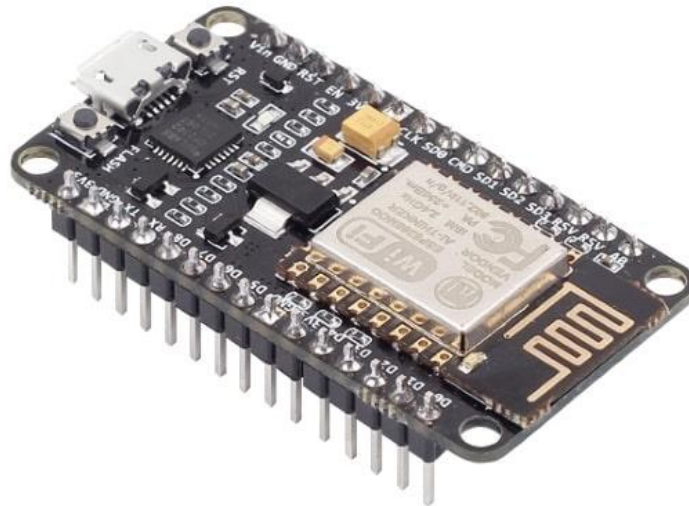


Figura 1: ESP8266 - Fonte: https://www.filipeflop.com/produto/modulo-wifi-esp8266-nodemcu-esp-12/?utm_source=google&utm_medium=organic&utm_campaign=shopping&utm_content=surfaces_across_google&gclid=CjwKCAiAm7OMBhAQEiwArvGi3DEJqqU6tXGORHwKMSDTHmpfjq0fZ6o7JsmI3Fbl6Kj3jue8gTXaGBoC2cEQAvD_BwE

Sensor DHT22

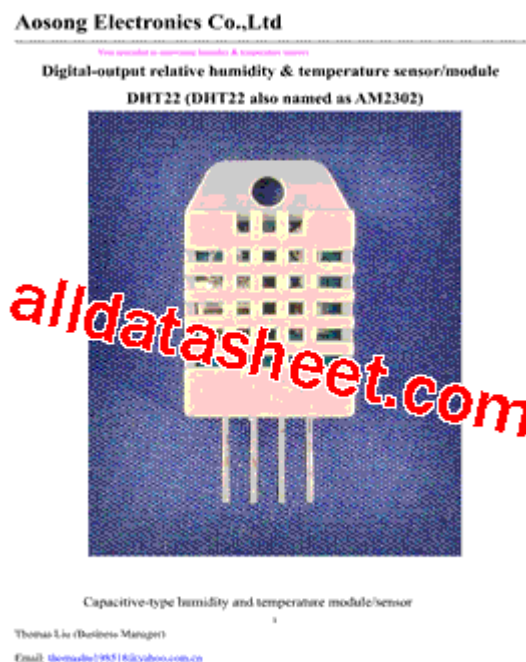
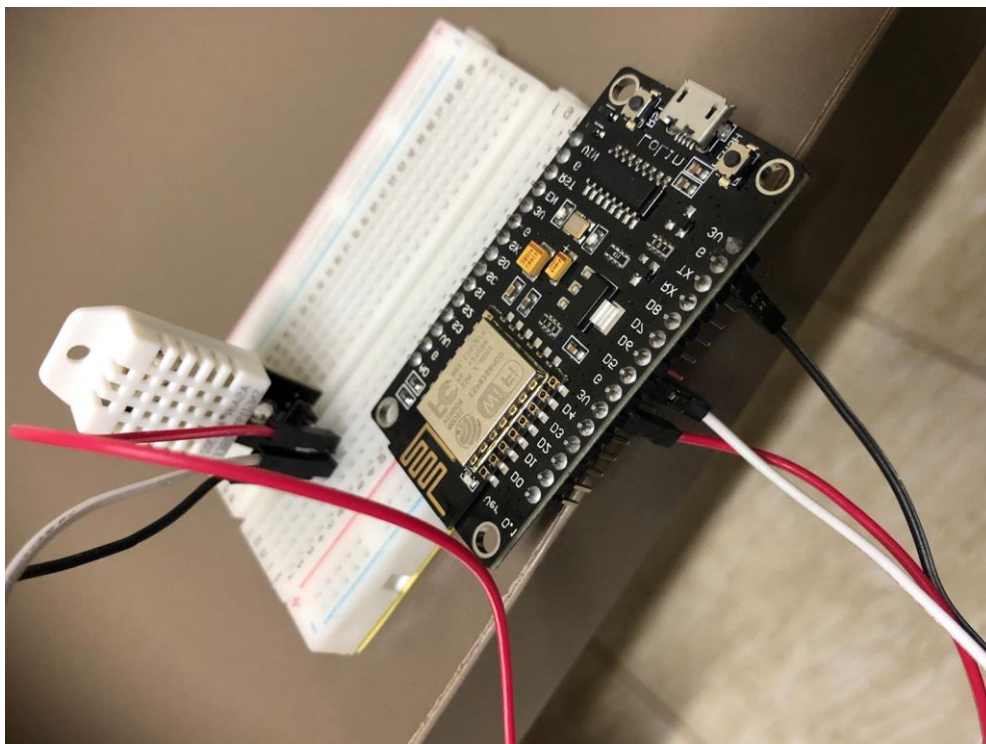
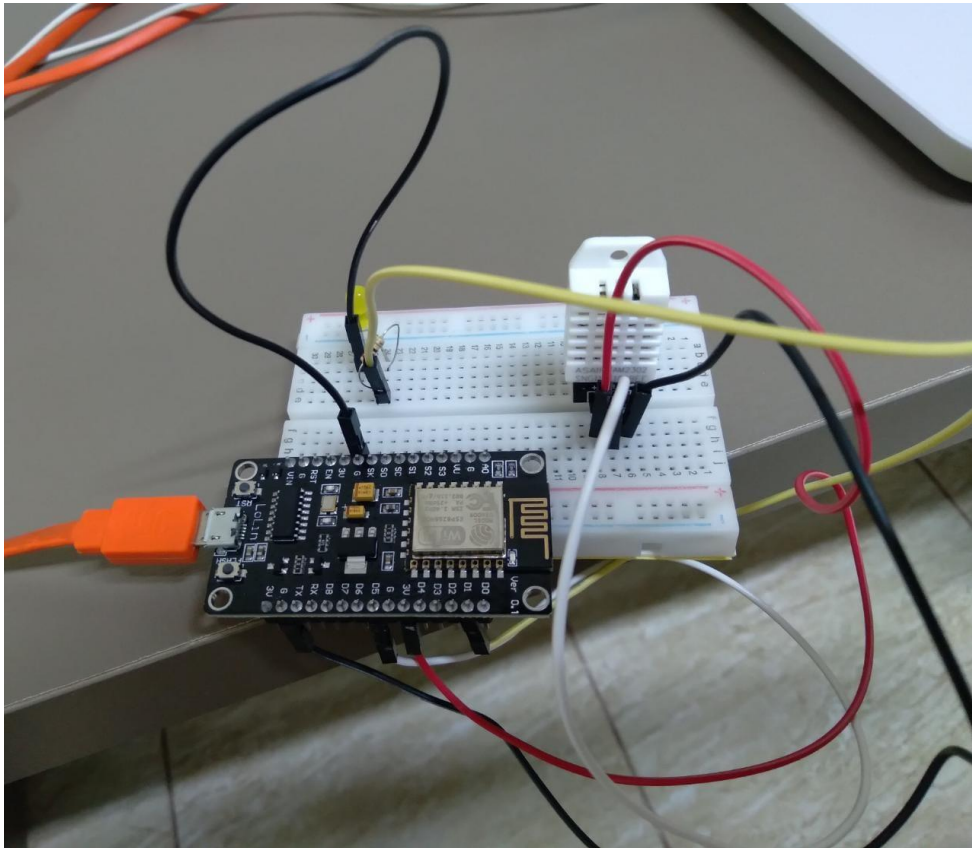


Figura 2: Sensor DHT 22 - <https://www.alldatasheet.com/datasheet-pdf/pdf/1132459/ETC2/DHT22.html>

Protótipo

Nas imagens abaixo, podemos ver a montagem de 2 protótipos, sendo o primeiro o protótipo completo

e o segundo, apenas da parte do sensor funcionando. Todos os componentes utilizados foram citados acima



A plataforma de prototipagem eletrônica utilizada será o ESP 8266 e o ambiente para desenvolvimento da programação será Arduino IDE

Especificações da placa ESP8266:

- Encapsulamento de 32 pinos QFN com 11.5mm por 11.5mm;

- Switch de Radiofrequência;
- Processadores de MAC e Baseband integrados;
- Gestão de Qualidade de Serviço (QoS) integrado;
- Interface I2S para aplicações de áudio;
- Regulador on-chip com baixo dropout para fornecimento de energia aos periféricos integrados;
- Arquitetura proprietária para geração de clock free-spurious;
- Engine integrada para criptografias WEP, TKIP, AES e WPA;
- Suporte aos protocolos 802.11 b/g/n;
- Wi-Fi Direct (P2P), soft-AP;
- Pilha de protocolo TCP/IP integrada com suporte a IPv4;
- Wi-Fi em frequência de 2.4GHz com suporte a WPA e WPA2;
- Potência de saída em +20dBm no modo 802.11b;
- Conversor ADC integrado com resolução de 10 bits;
- Suporte a uma variedade de antenas;
- Energia de consumo em modo sleep menor que 10uA;
- Tempo para sair de sleep e transmitir pacotes menor que 2ms;
- Potência de standby menor que 1.0mW;

O sensor DHT22 através do pino out, envia os dados para a placa ESP8266. A placa envia os dados via wifi para o servidor mqtt.eclipseprojects.io.

O MQTT (do inglês Message Queue Telemetry Transport) é um protocolo de comunicação entre máquinas (Machine to Machine - M2M). Ele utiliza dois conceitos: subscribed e publish. Um dispositivo pode ser “inscrito= subscribed” ou uma “publicação=publish”. Quando tal dispositivo é um Publish, ele publica informações pré-estabelecidas. Já o dispositivo Subscribe recebe todas essas informações que estão sendo publicadas.

O Broker é um servidor que gere as informações aos inscritos e vice-versa, funcionando, então, como um intermediário entre Subscribers e Publishers. Neste projeto, o broker utilizado é o <https://mqtt.eclipseprojects.io/> e porta: 1883



Com o protótipo feito, é necessário fazer a configuração do aplicativo de celular. Aplicativo utilizado: MQTT Dashboard: Iot and Node-RED.

1º Passo: Fazer a configuração do servidor broker:

Clicar no Botão +;

- Preencher os campos nome do servidor = "Nome livre para o seu projeto", Address = "mqtt.eclipseprojects.io", Port = "1883", selecionar a opção QOS1 e clicar no ícone salvar.

2º Passo: Criar os botões para as funções do projeto

- Clicar no Botão +;

- Temperatura:

- Escolher o botão "text";
- Preencher os campos: Nome do Bloco = "Temperatura", tópico para inscrever = "lab/temp", Desabilitar a opção "habilitar publicação" (Caso esteja selecionada), personalizar o ícone com imagem e cor conforme tua preferencia e clicar no botão salvar;

- Humidade:

- Escolher o botão "text";
- Preencher os campos: Nome do Bloco = "Humidade", tópico para inscrever = "lab/humid", Desabilitar a opção "habilitar publicação" (Caso esteja selecionada), personalizar o ícone com imagem e cor conforme tua preferencia e clicar no botão salvar;

- Led:

- Escolher o botão "button";
- Preencher os campos: Nome do Bloco = "Led", tópico para publicar = "casa/sala", personalizar o ícone com imagem e cor conforme tua preferencia e clicar no botão salvar;

Agora é necessário fazer a programação, mas, antes disso, para que o ambiente do arduino consiga se comunicar com a placa é necessário fazer as seguintes configurações:

1º Baixar o driver da placa: Dentro da IDE Arduino, acessar: arquivo – preferencias – copiar e colar este link: http://arduino.esp8266.com/stable/package_esp8266com_index.json em URLS adicionais para gerenciamento de placas – ok;

2º ir em: Ferramentas – placa – gerenciador de placas – digitar “ESP8266” – instalar- ok;

3º Selecionar a versão da placa: Acessar: Ferramentas – placa – ESP8266 Boards (3.0.2) NodeMCU 1.0 (ESP 12E Module);

4º Selecionar a porta: Ferramentas – Portas – COM4;

5º Baixar as seguintes bibliotecas: ESP8266WiFi.h, PubSubClient.h e DHT.h. Para isso, basta digitar o nome decada uma em: Sketch – incluir biblioteca – gerenciar bibliotecas – digitar o nome da biblioteca desejada – clicar em instalar e depois em ok

Agora é só baixar o código fonte que está no github (link em “Conclusão”), abrir na IDE Arduino e executar. Seu projeto já estará funcionando.

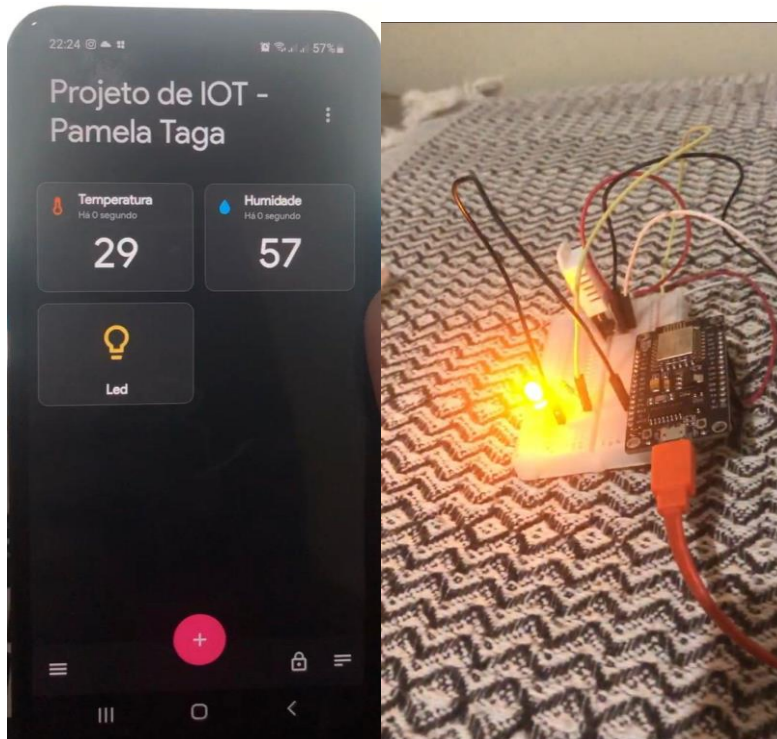
Conclusão

Link para o vídeo-demonstração no YouTube: <https://youtu.be/h0I-T72zli0>

Link para o github: <https://github.com/pamelataga/projeto-iot>

| Núm. Medida | Sensor/atuador | Tempo de resposta |
|-------------|----------------------|-------------------|
| - | Led | Não funcionou |
| Segundos | DHT 22 – Temperatura | 5 |
| Segundos | DHT 22 – Temperatura | 6 |
| Segundos | DHT 22 – Temperatura | 6 |
| Segundos | DHT 22 – Temperatura | 5 |
| Segundos | DHT 22 – Humidade | 5 |
| Segundos | DHT 22 – Humidade | 6 |
| Segundos | DHT 22 – Humidade | 6 |

Imagens do projeto funcionando:



Questões:

i) Os objetivos propostos foram alcançados?

R: Não completamente, o objetivo final era ver a temperatura, humidade e acender o led através de um aplicativo smartphone porém, a programação na parte do led não está funcionando corretamente, não sendo possível realizar essa parte que foi proposta

ii) Quais são os principais problemas enfrentados e como foram resolvidos?

R: Houve vários problemas enfrentados os 2 principais foram:

- Fazer a placa ESP8266 se comunicar com o notebook: a solução foi pesquisar muito sobre as configurações na IDE Arduino e trocar o cabo USB que estava com mal contato;

- Acender o LED pelo aplicativo: Consegui fazer a montagem corretamente do circuito e configurar o aplicativo celular, porém, alguma coisa na programação não está funcionando

iii) Quais são as vantagens e desvantagens do projeto?

R: Vantagens: Poder monitorar temperatura e humidade de vários ambientes, seja ele residencial ou rural;

Desvantagem: Mal funcionamento da função do led.

iv) O que deveria/poderia ser feito para melhorar o projeto?

R: Primeiro é necessário corrigir a programação para que o led funcione corretamente, depois, para melhorar, seria interessante acender uma lâmpada através do aplicativo de celular no lugar do led.

5. Referências

Locatelli, Caroline. Guia rápido — Monitoramento e Controle por Aplicativo - MQTT. Disponível em: <https://www.curtocircuito.com.br/blog/Categoria%20IoT/monitoramento-e-controle-por-aplicativo-mqtt>. Acesso em 20/11/2021.

Em 5 minutos, Automação. Guia rápido — Ligando e desligando um led com o ESP8266 NodeMCU. Disponível em: <https://medium.com/@automacaoem5minutos/guia-rápido-ligando-e-desligando-um-led-com-o-esp8266-nodemcu-5844fd4e4992>. Acesso em 01/12/2021.

Lúcio, Sérgio. ESP8266 NODEMCU #04: CONFIGURANDO O PUBLISH DO MQTT (Termômetro IoT). Youtube, 21 DE JULHO DE 2020. Disponível em: <<https://www.youtube.com/watch?v=vj2i1fSIInx0&list=RDCMUChymjuTJ1BIsKai5RVNIF2A&index=4>>.

Lúcio, Sérgio. ESP8266 NODEMCU #05: CONFIGURANDO O SUBSCRIBE DO MQTT (TERMÔMETRO IoT). Youtube, 23 DE JULHO DE 2020. Disponível em: <<https://www.youtube.com/watch?v=CvcTzarqQbI&list=RDCMUChymjuTJ1BIsKai5RVNIF2A&index=3>>.

Curvello, André. Apresentando o módulo ESP8266. Disponível em: <<https://www.embarcados.com.br/modulo-esp8266/>>. Acesso em 11/11/2021.

GREENGARD, Samuel. The Internet of Things. Mit Press, 27 de março de 2015.

TUDO o que você precisa saber sobre Casa Inteligente e suas tecnologias. INTERBRAS, 24 de novembro de 2020. Disponível em: <<https://blog.intelbras.com.br/casa-inteligente/>>. Acesso em 31/08/2021

