

Water plant

Renato V. Dantas¹, Caio A. Schmidt², Wilian França Costa³

¹Faculdade de Computação e Informatica – Universidade Presbiteriana Mackenzie (UPM)
01.302-907 – São Paulo – SP – Brasil

renato.srdantas@gmail.com, caio.alvares22@gmail.com

Abstract. *This article describes a project in Internet of things. The goal is to use a microcontroller that has internet connectivity, and through that connection we will use certain messaging protocols to generate data. In this project the main objective is to ensure that the plant is always at the correct humidity level.*

Resumo. *Este artigo descreve um projeto na área de Internet das coisas. O objetivo será utilizar um microcontrolador que tenha uma conectividade com a internet, e através dessa conexão utilizaremos determinados protocolos de mensageria para gerarmos dados. Neste projeto o principal objetivo é fazer com que a planta esteja sempre nos níveis corretos de umidade.*

1. Introdução

O objetivo será manter as plantas nos níveis corretos de umidade, a partir da utilização de um microcontrolador que estará conectado com a internet, neste microcontrolador estarão conectados um sensor de umidade e temperatura e uma bomba de água, para que tudo funcione de modo automático utiliza-se o protocolo MQTT de mensageria para obter os dados do sensor e dessa forma o microcontrolador irá disparar a bomba de água para regar as plantas, se necessário. O cliente terá todo o controle de medição do sensor através de um site que irá montar gráficos de umidade e temperatura, assim ele poderá ter um controle maior sobre o que está acontecendo na plantação.

2. Materiais e metodos

O microcontrolador que utilizado utilizar neste projeto e o ESP8266 Nodemcu, ele tem como foco os projetos de Internet das coisas. Ele vem com um firmware que funciona com ESP8266 Wi-Fi SoC, e um hardware que é baseado no ESP-12. Este firmware usa o projeto Lua de linguagem de script, e como podemos observar na Figura 1 as especificações dos pinos e entradas que podemos encontrar no microcontrolador. Este controlador possui entradas analógicas e digitais [Wan et al. 2019]. E consegue fornecer até 3.3V de energia [cityos air 2020].

Para conseguir ler os dados que pretende-se capturar, utilizar-se o módulo DHT11. Este módulo que tem o foco em umidade e temperatura, se utiliza um sinal digital calibrado para funcionar [mouser 2020]. Por usar um sinal digital de aquisição ele tem uma alta confiabilidade e estabilidade, este módulo inclui um meio resistivo de medir a umidade e utiliza um NTC para medir a temperatura, e dentro é conectado por um microcontrolador de 8-bit, dessa forma ele oferece uma boa qualidade e tem um tempo de resposta muito rápida [Tianlong 2010].

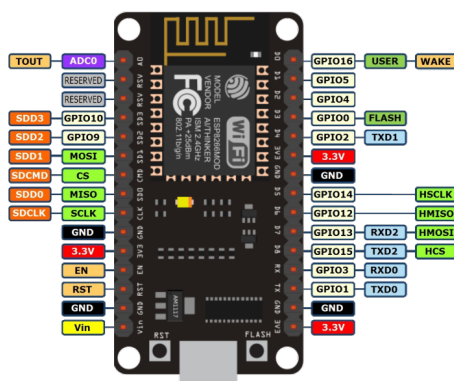


Figure 1. ESP8266 pinos

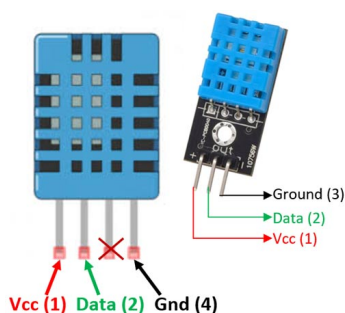


Figure 2. DHT11 pinos

Para enviar os dados para o módulo DHT11 utilizar-se um sensor de umidade e temperatura denominado FC-28. Este sensor pode ser usando com um output tanto analógico ou digital. Para fazer uma medição ele utiliza a resistência que a terra tem para conduzir a eletricidade, dessa forma quanto menor for essa resistência mais água tem na terra. Ele tem uma voltagem de input de 3.3V – 5V e 0 – 4.2V de output, este sensor também tem um potenciômetro que é usado para definir um valor limite. Esse valor limite é comparado cada vez que o sensor faz uma leitura [summerfuelrobots 2020].

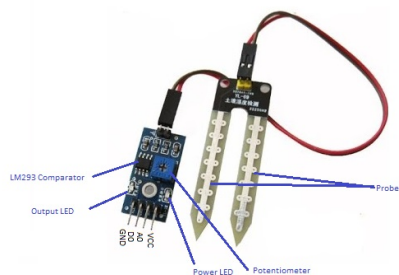


Figure 3. FC-28 sensor de umidade e temperatura

Para bombear água para as plantas utilizar-se uma bomba elétrica de 5V. Como a ESP8266 não tem alimentação elétrica de 5V, vamos utilizar um relé de 5V [circuitstoday 2020]. O seu uso é focado no controle de circuitos com sinais de baixa potência, dessa forma ele é capaz de replicar sinais que recebe para um outro circuito ou

aparelho [Bhadani and Vashisht 2019].



Figure 4. Bomba de agua 5v



Figure 5. Modulo rele 2 canais 5v

2.1. Protocolos

O protocolo de troca de envio de mensagens que utiliza-se é o MQTT. Este protocolo foi desenvolvido pela IBM e Eurotech, ele foi baseado na pilha TCP/IP. Este protocolo tem suporte para comunicação assíncrona, ou seja, a comunicação não ocorre ao mesmo tempo, porém esse protocolo se utiliza um modelo fundamentado na topologia de publicação de dados que seria assinatura/leitura para o envio das mensagens. Ele é um bom protocolo para se utilizar em IOT por ser um protocolo simples, de fácil implementação, por ser uma aplicação leve ela é ideal para a comunicação remota entre dispositivos. A base do funcionamento deste protocolo, são os cliente e um servidor que é responsável por rotear as mensagens para os clientes, este servidor que é responsável por receber, enfileirar e disparar esses dados é chamada de broker [Thangavel et al. 2014].

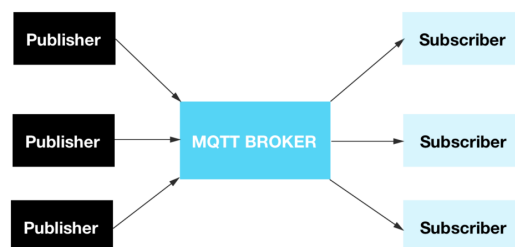


Figure 6. MQTT

3. Métodos

Para o desenvolvimento do software foi utilizada a Arduino IDE, com a utilização de algumas bibliotecas conseguimos criar uma conexão entre o NodeMcu e uma internet

residencial, após a conexão ter sido estabelecida iniciamos a tentativa de entrar em contato com um broker que está hospedado em um site chamado ThingSpeak, após se conectar o sensor de umidade passa a enviar os dados para o microcontrolador. Para enviar esses dados para o site foi utilizado o protocolo MQTT de mensageria, após estabelecer esta conexão o NodeMcu passar a publicar esses dados no broker o qual foi designado, e então o site passa a transformar esses dados em um gráfico, que ficará disponível sempre que alguém precise dar uma olhada nesses gráficos. Após essas conexão o sistema passa a trabalhar para manter a umidade do solo nos pontos escolhidos, sempre que o sensor captar uma umidade abaixo da desejada o microcontrolador irá transmitir um sinal para que uma bomba de água ligue, e dessa forma irá sempre manter o nível de umidade que foi estabelecido previamente.

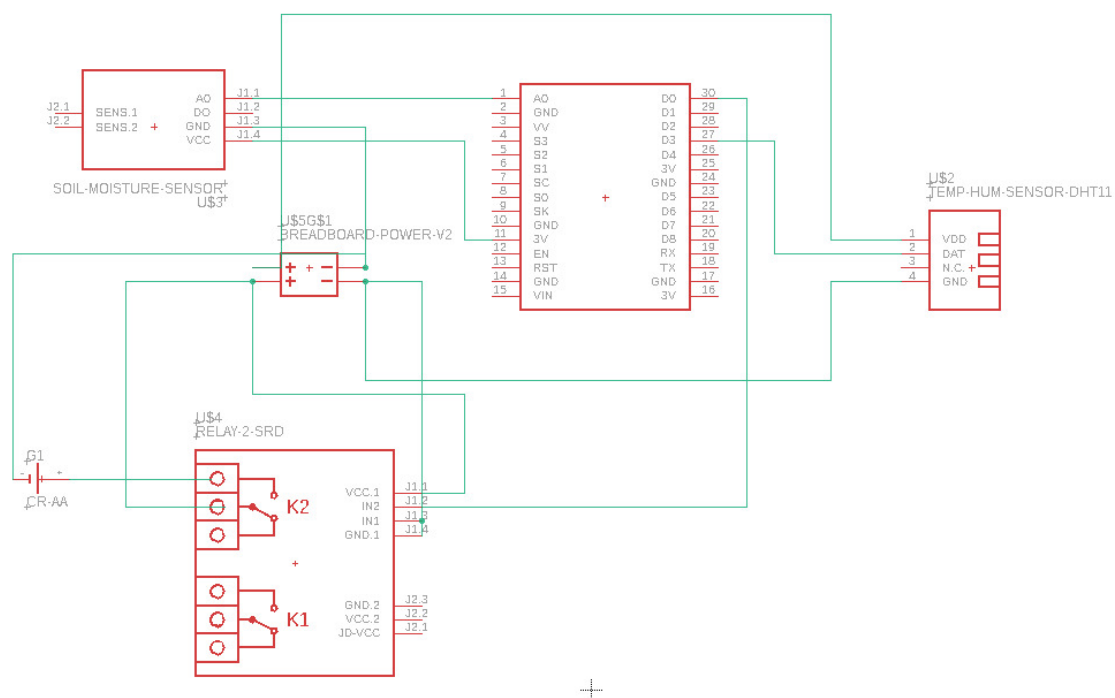


Figure 7. Schema do projeto

4. Resultados

O funcionamento do projeto segue o mesmo que foi descrito no capítulo acima, o microcontrolador NodeMcu, após realizar suas conexões, tanto com a internet quanto ao Broker do site ThingSpeak, passa a coletar os dados dos seus sensores, o sistema mantém a conexão MQTT sempre checada para não haver queda na conexão com o servidor, após algum período o NodeMcu envia um dado coletado para o Broker e então o servidor pública o dado em um gráfico.

Durante o desenvolvimento projeto foram utilizados dois brokers diferentes para o teste da publicação dos dados, EasyIoT Cloud e ThingSpeak. Ao utilizar o EasyIoT Cloud tivemos sucesso em publicar os dados porém o microcontrolador não era capaz de ligar a bomba de água para fazer a irrigação automática. Com o ThingSpeak o problema foi o oposto, a bomba de água era ligada quando a umidade do solo chegava a determinados níveis, porém os dados não estavam sendo publicados nos campos desejado. Após

um estudo minucioso da documentação do site na parte de publicação de dados em um campo específico utilizando MQTT, foi possível adaptar o código desenvolvido para que tal publicação fosse feita.

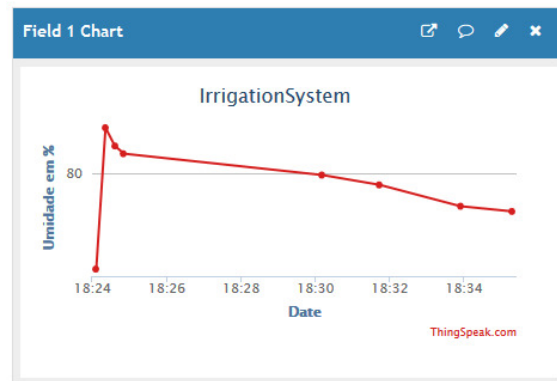


Figure 8. Grafico de medição do solo em escala de porcentagem

5. Conclusão

A Partir dos objetivos estabelecidos previamente neste documento, podemos concluir que o resultado obtido foi satisfatório, todos os testes feitos após a adaptação do sistema ao ThingSpeak apontaram que os dados estavam sendo gerados e enviados a partir do protocolo MQTT e quando era necessário o microcontrolador ligava a bomba de água para manter a terra nos níveis de umidade desejado. Com a grande quantidade de tarefas que realizamos todos os dias se torna difícil tomar conta de tudo, para as pessoas que possuem pequenas plantas ou vasos em casa, é recorrente esquecer de regar suas mudas, este projeto busca ajudar esse grupo de pessoas, para nunca deixarem suas plantas morrerem. A próxima etapa deste projeto seria incluir uma interface de usuário, a onde seria possível estabelecer o nível de umidade desejada e acrescentar botões para o controle da bomba de água, o usuário poderia manter a irrigação de modo automático ou manual, dessa forma ele mesmo seria capaz de ligar a bomba através do uso dessa interface [renatovieiradantas 2020].

References

- Bhadani, P. and Vashisht, V. (2019). Soil moisture, temperature and humidity measurement using arduino. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 567–571. IEEE.
- circuitstoday (2020). arduino-soil-moisture-sensor. <http://www.circuitstoday.com/arduino-soil-moisture-sensor> , Last accessed 8 Abr 2020.
- cityos air (2020). Esp8266 nodemcu. <https://cityos-air.readme.io/docs/esp8266-nodemcu>, Last accessed 8 Abr 2020.
- mouser (2020). Dht11. <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>, Last accessed 8 Abr 2020.
- renatovieiradantas, c. (2020). water-plant. <https://github.com/srdantas/water-plant>, Last accessed 22 Jun 2020.

summerfuelrobots (2020). 2-channel-relay-module.
<https://sites.google.com/site/summerfuelrobots/arduino-sensor-tutorials/2-channel-relay-module>, Last accessed 8 Apr 2020.

Thangavel, D., Ma, X., Valera, A., Tan, H.-X., and Tan, C. K.-Y. (2014). Performance evaluation of mqtt and coap via a common middleware. In *2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)*, pages 1–6. IEEE.

Tianlong, N. (2010). Application of single bus sensor dht11 in temperature humidity measure and control system [j]. *Microcontrollers & Embedded Systems*, 6:026.

Wan, Z., Song, Y., and Cao, Z. (2019). Environment dynamic monitoring and remote control of greenhouse with esp8266 nodemcu. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 377–382. IEEE.