

IrrigaFácil: Sistema Automatizado de Irrigação para Agricultura Sustentável

Fernando Henrique Souza de Araújo, Prof. William França Costa

Faculdade de Computação e Informática
Universidade Presbiteriana Mackenzie (UPM) – São Paulo, SP – Brazil

`fernandohenriquesouza.araujo@mackenzista.com.br`

Abstract.

This paper proposes an automated irrigation system aimed at small-scale agricultural industries, focusing on optimizing water use and promoting sustainability. The system integrates soil moisture sensors and an actuator that controls water pumps, ensuring an efficient and tailored irrigation process. By employing MQTT protocol and an IoT-based architecture, this solution enables remote monitoring and management of the irrigation system. The project addresses key sustainability challenges, reducing water waste and improving productivity, contributing to the Sustainable Development Goal 9 (Industry, Innovation, and Infrastructure).

Resumo. *Este artigo propõe um sistema de irrigação automatizado voltado para pequenas indústrias agrícolas, com foco na otimização do uso de água e promoção da sustentabilidade. O sistema integra sensores de umidade do solo e um atuador que controla bombas de água, garantindo um processo de irrigação eficiente e personalizado. Utilizando o protocolo MQTT e uma arquitetura baseada em IoT, a solução permite o monitoramento e a gestão remotos do sistema de irrigação. O projeto visa resolver desafios importantes de sustentabilidade, reduzindo o desperdício de água e melhorando a produtividade, contribuindo para a Meta de Desenvolvimento Sustentável 9 (Indústria, Inovação e Infraestrutura).*

1. Introdução

A agricultura sustentável tem sido um tema central nas discussões sobre desenvolvimento econômico e bem-estar social, principalmente em países com escassez de recursos hídricos e demanda crescente por alimentos. A irrigação, uma prática crucial na agricultura, responde por grande parte do consumo de água no mundo. No entanto, o uso ineficiente desse recurso pode levar a desperdícios significativos, afetando tanto a produtividade quanto o meio ambiente. Nesse contexto, soluções tecnológicas que otimizem o uso da água tornam-se essenciais para o desenvolvimento sustentável.

Nos últimos anos, a aplicação da Internet das Coisas (IoT) em sistemas agrícolas vem se destacando como uma inovação crucial para melhorar a eficiência de processos produtivos. Projetos como o monitoramento de umidade do solo e controle automatizado de irrigação têm ganhado relevância, especialmente para pequenos produtores que buscam maximizar a eficiência de suas operações e minimizar os custos com recursos naturais. Trabalhos como o de **Jha et al. (2019)** demonstram o impacto da automação na agricultura de precisão, reduzindo o consumo de água e aumentando a produtividade.

Esse estudo se insere nesse contexto, propondo o desenvolvimento de um sistema automatizado de irrigação que utiliza sensores de umidade do solo para otimizar o uso da água em pequenas indústrias agrícolas. A inovação reside na capacidade de monitorar remotamente os níveis de umidade e controlar bombas de irrigação de forma autônoma e eficiente. Utilizando a comunicação via protocolo MQTT, o sistema permite a integração fácil com plataformas de gerenciamento remoto, o que aumenta sua aplicabilidade e escalabilidade.

O objetivo principal deste trabalho é apresentar uma solução sustentável e acessível para pequenos agricultores, contribuindo para os Objetivos de Desenvolvimento Sustentável (ODS), especialmente o ODS 9, que visa promover a inovação e a infraestrutura sustentável. Além disso, o projeto busca fortalecer a capacidade produtiva das pequenas indústrias agrícolas, oferecendo uma alternativa tecnológica viável para a otimização de recursos hídricos e energéticos.

2. Materiais e Métodos

Neste projeto, foi desenvolvido um sistema de irrigação automatizado utilizando componentes de hardware que permitem a coleta de dados em tempo real sobre as condições do solo e o acionamento automatizado do sistema de irrigação. A seguir, são descritos os materiais utilizados e os métodos implementados para a execução do sistema.

2.1 Plataforma de Prototipagem: NodeMCU (ESP8266)

A **NodeMCU** é uma plataforma de prototipagem baseada no microcontrolador **ESP8266**, que possui conectividade Wi-Fi integrada. Suas especificações técnicas incluem **Tensão de Operação:** 3.3V, **GPIOs:** 17 pinos de entrada/saída digitais, **Memória Flash:** 4 MB, **Frequência do Clock:** 80 MHz a 160 MHz, **Conectividade:** Wi-Fi IEEE 802.11 b/g/n, **Protocolos Suportados:** MQTT, HTTP, entre outros, **Dimensões:** 58 mm x 31 mm. A NodeMCU é ideal para aplicações IoT devido ao seu tamanho compacto, baixo consumo de energia e suporte a comunicação via Wi-Fi, permitindo o envio de dados de sensores para a internet.

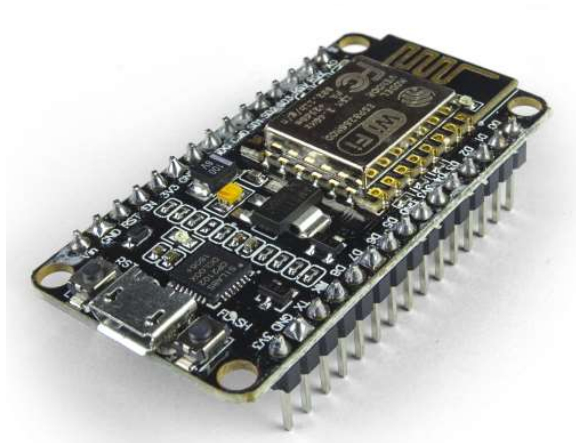


Figura 2.1 – Node MCU 8266 Fonte: <https://www.robocore.net/wifi/nodemcu-esp8266-12-v2>

2.2 Sensor de Umidade do Solo YL-69

O **YL-69** é um sensor que mede a umidade do solo através da variação da resistência elétrica entre dois eletrodos. As especificações técnicas incluem: **Tensão de Operação:** 3.3V a 5V, **Corrente de Operação:** 35mA, **Interface de Saída:** Analógica (0-1023), **Dimensões da Sonda:** 60 mm x 20 mm, **Módulo de Controle:** Potenciômetro ajustável para definir o limite de umidade. O YL-69 permite monitorar os níveis de umidade do solo, e o valor é enviado para a NodeMCU, que utiliza essas informações para determinar o acionamento da bomba de água.

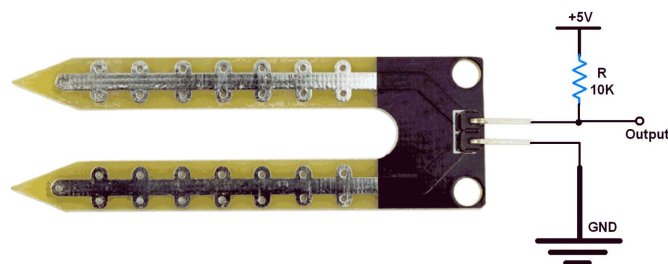


Figura 2.2 – Sensor de umidade do solo Fonte: <https://www.electronicwings.com/sensors-modules/soil-moisture-sensor>

2.3. Atuador: Mini Bomba de Água Motor 12V DC - RS-385

A bomba de água será responsável por irrigar o solo automaticamente, acionada de acordo com os níveis de umidade medidos. As especificações técnicas da bomba

utilizada são: **Tensão de Operação:** 12V, **Corrente Nominal:** 0.6A, **Fluxo de Água:** 240 L/h, **Altura Máxima de Elevação:** 3 metros, **Material:** Plástico resistente à corrosão, **Dimensões:** 90 mm * 40 mm * 35 mm. Esta bomba de água é eficiente e adequada para pequenos sistemas de irrigação, fornecendo uma solução compacta e de baixo custo.



Figura 2.3 – Atuador Bomba D'água Fonte: Eletrogate (2024)

2.4. Módulo Relé 4 Canais 3V 10A com Borne KRE para ESP32

O relé de 3V será utilizado para controlar o acionamento da bomba de água. Ele funciona como um interruptor eletrônico acionado pela NodeMCU. As especificações técnicas incluem: **Tensão de Operação:** 3V DC, **Corrente de Acionamento:** 15-20mA, **Capacidade de Contato:** 10A 250VAC ou 10A 30VDC, **Tempo de Resposta:** 10ms, **Dimensões:** 75 x 50 x 17mm. O relé é necessário para garantir o controle seguro e eficiente do fluxo de corrente que aciona a bomba de água.



Figura 2.4 – Módulo Relé 4 Canais Fonte: Eletrogate (2024)

2.5. Fonte 9V 1A Bivolt para Arduino

A fonte 9V 1A bivolt para Arduino é essencial para fornecer alimentação estável e confiável a projetos eletrônicos, garantindo o funcionamento adequado de dispositivos

como microcontroladores, sensores e módulos diversos. As especificações técnicas incluem: **Tensão de Entrada:** Bivolt 100/240VAC 4764Hz, **Tensão de Saída:** 9VDC0 **Corrente de Saída Máxima:** 1A, **Plugue:** P4 Macho, **Diâmetro do Furo Interno:** 2.1mm, **Diâmetro Externo:** 5.5mm. A utilização da fonte é fundamental para assegurar que todos os componentes eletrônicos recebam a energia necessária de forma consistente.



Figura 2.5 – Fonte 9V 1A Bivolt para Arduino Fonte: Eletrogate (2024)

2.6. Protoboard 830 Pontos

A protoboard de será utilizada para a montagem e teste de circuitos eletrônicos de forma prática e versátil. As especificações técnicas incluem: **Faixa de Temperatura:** -20 a 80°, **Furos:** 830,, **Para terminais e Condutores de 0,3 a 0,8 mm (20 a 29 AWG),** **Tensão máxima:** 500v AC por minuto, **Dimensões:** 165mm x 57mm x 10mm.

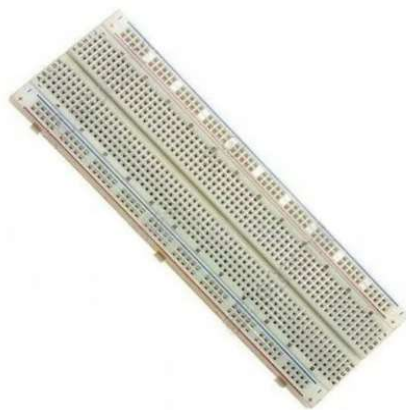


Figura 2.6 – Protoboard 830 Pontos Fonte: Eletrogate (2024)

2.7. Protocolo de Comunicação: MQTT

Para a comunicação entre o sistema de irrigação e a interface de monitoramento remoto, será utilizado o protocolo **MQTT**. Esse protocolo é ideal para dispositivos IoT devido ao seu baixo consumo de largura de banda e eficiência na transmissão de pequenos pacotes de dados. As características principais do MQTT incluem:

- **Modelo de Comunicação:** Publicador/Assinante
- **Segurança:** Suporte a SSL/TLS para criptografia de dados
- **Consumo de Banda:** Muito baixo, com overhead mínimo
- **Tamanho Máximo de Mensagem:** 256 MB
- **Portas Padrão:** 1883 (TCP) e 8883 (SSL)

O **MQTT** permite o envio de dados do sensor de umidade para uma interface de controle remoto, permitindo o monitoramento e o controle da irrigação a partir de qualquer lugar.

2.8. Prototipagem: Fritzing

Nesse projeto, foi utilizada a plataforma de prototipagem Fritzing para elaborar o diagrama do circuito elétrico e a disposição dos componentes na protoboard. O Fritzing é uma ferramenta de código aberto que permite aos usuários criar projetos eletrônicos de forma intuitiva, facilitando a visualização e documentação do hardware utilizado.

A utilização do Fritzing na prototipagem acelerou o processo de desenvolvimento, permitindo simular e testar diferentes configurações antes da montagem física. Isso contribuiu para a eficiência e sucesso do projeto de irrigador automático, garantindo que todos os componentes funcionassem corretamente.

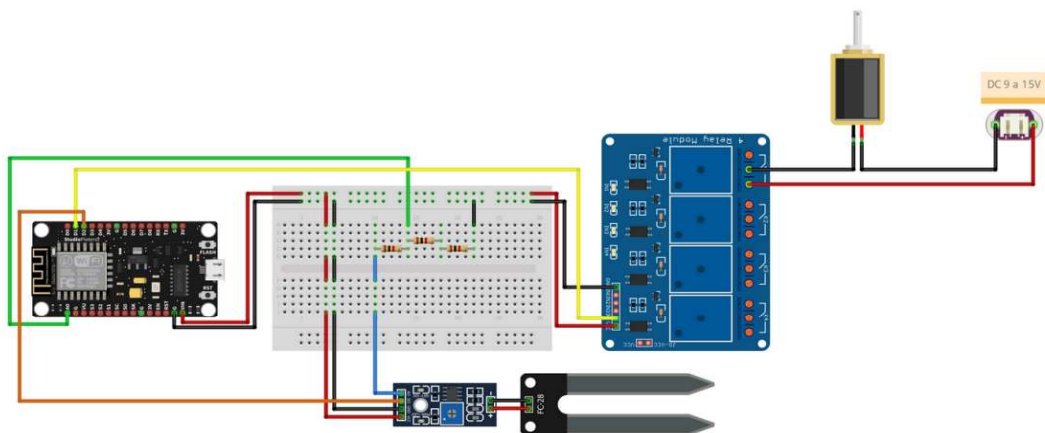


Figura 2.7 – Protótipo do circuito no Fritzing Fonte: Autor

2.9. Funcionamento do protótipo:

1. Leitura do Sensor de Umidade do Solo:

- O sensor FC-28 mede a umidade no solo e fornece um sinal proporcional ao nível de umidade.
- Esse sinal é enviado ao ESP8266 através de um dos pinos.

2. Processamento no ESP8266:

- O microcontrolador recebe os dados do sensor e avalia o nível de umidade.
- Com base em um limite de umidade definido no código, o ESP8266 decide se a bomba d'água deve ser ligada ou desligada.

3. Controle do Módulo de Relé:

- O ESP8266 envia um sinal digital ao módulo de relé:
 - Se o solo estiver seco (umidade baixa), o relé é ativado, ligando a bomba d'água.
 - Se o solo estiver úmido (umidade alta), o relé é desativado, desligando a bomba.

4. Operação da Bomba d'Água:

- Quando o relé é ativado, a bomba recebe energia da fonte externa e bombeia água para irrigar o solo.
- Quando o relé é desativado, a bomba é parada.

2.10. Fluxograma: Lucidchart

Nesse projeto, foi utilizado o Lucidchart para elaborar o fluxograma que representa o funcionamento lógico do sistema de irrigação automática. O Lucidchart é uma ferramenta online de criação de diagramas que permite aos usuários desenvolver fluxogramas, organogramas e outros tipos de diagramas de forma intuitiva.

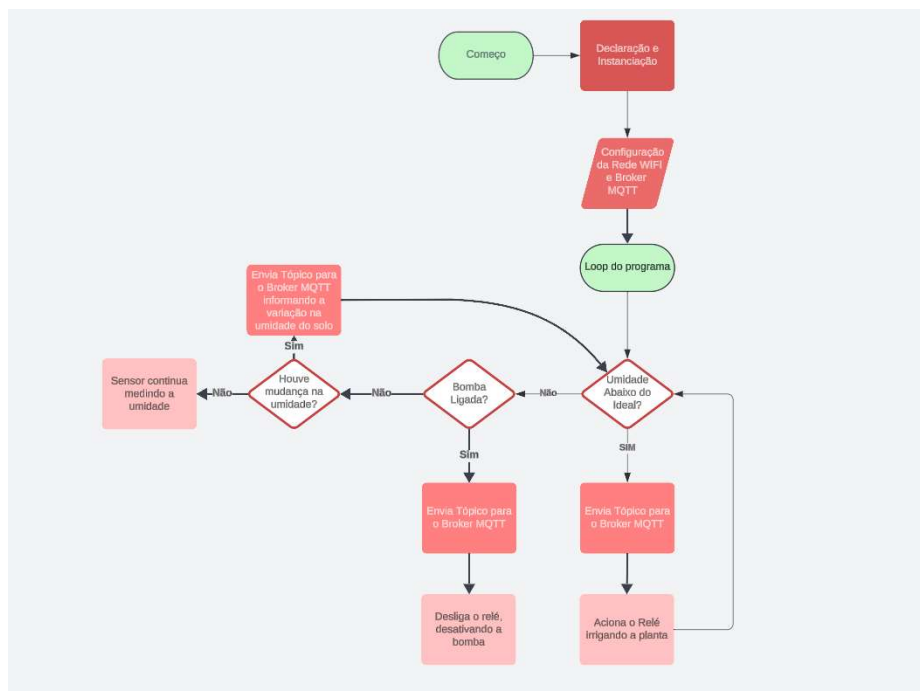


Figura 2.8 – Fluxograma de Funcionamento Geral Fonte: Autor

2.11. Broker MQTT: Mosquitto

No Irrigafácil foi utilizado o Mosquitto como broker MQTT para gerenciar a comunicação entre o ESP8266 e outros componentes do sistema. O Mosquitto é um broker MQTT de código aberto que implementa o protocolo MQTT versão 3.1 e 3.1.1, reconhecido por sua eficiência, simplicidade e confiabilidade na transmissão de mensagens em redes IoT.

MQTT Client Name: MQTTBoxClientIrigador

MQTT Client id: 8e87e8af-29f9-4c36-8a0f-da702fa8c13

Protocol: mqtt / tcp

Username: Username

Password: Password

Reconnect Period (milliseconds): 1000

Will - Topic: Will - Topic

Host: test.mosquitto.org

Append timestamp to MQTT client id? ☒ Yes

Clean Session? ☒ Yes

Reschedule Pings? ☒ Yes

KeepAlive (seconds): 60

Will - Retain: ☐ No

Will - QoS: 0 - Almost Once

Will - Payload:

Broker is MQTT v3.1.1 compliant? ☒ Yes

Auto connect on app launch? ☒ Yes

Queue outgoing QoS zero messages? ☒ Yes

Buttons: Save, Delete

Figura 2.9 – Configuração do Broker MQTT Fonte: Autor

2.12. Código Fonte

O código-fonte do IrrigaFácil foi desenvolvido para o microcontrolador ESP8266, utilizando a linguagem Arduino C++. A seguir, apresento uma descrição detalhada em forma de pseudocódigo que explica o funcionamento do sistema, facilitando a compreensão dos processos envolvidos.

1 - Importação de Bibliotecas Necessárias

Importar biblioteca ESP8266WiFi

Importar biblioteca PubSubClient

Importar biblioteca NTPClient

Importar biblioteca WiFiUdp

2 - Declaração de Constantes e Variáveis Globais

Definir SSID e senha da rede Wi-Fi

Definir endereço e porta do broker MQTT (Mosquitto)

Criar objetos WiFiClient e PubSubClient para conexão MQTT

Configurar NTPClient com servidor NTP, fuso horário e intervalo de atualização

Definir pinos para sensor de umidade (pino analógico) e relé (pino digital)

Inicializar variáveis para armazenar leituras de umidade atual e anterior

Definir constante THRESHOLD para diferença mínima de umidade

Definir limites de calibração para o sensor de umidade (SENSOR_SECO e SENSOR_UMIDO)

Configurar variáveis para controle de tempo de leitura do sensor (lastCheckTime e checkInterval)

3 - Função para Obter o Timestamp Atual

Função getUnixTimestamp():

Se timeClient conseguiu atualizar o tempo:

Obter o timestamp atual em segundos

Converter timestamp para milissegundos (multiplicar por 1000)

Imprimir timestamp sincronizado no monitor serial

Retornar timestamp em milissegundos

Senão:

Imprimir mensagem de erro no monitor serial

Retornar 0

4 - Função de Configuração (setup)

- Inicia comunicação serial a 9600 bps
- Configura pino do relé como saída
- Chama função `setup_wifi()` para conectar ao Wi-Fi
- Imprime endereço IP obtido no monitor serial
- Inicia `timeClient` para sincronização NTP
- Imprime mensagem de sincronização do tempo
- Enquanto o tempo não estiver sincronizado ou até atingir timeout:
 - Forçar atualização do `timeClient`
 - Aguardar 500 ms
- Se o tempo foi sincronizado:
 - Imprimir mensagem de sucesso e horário atual formatado
- Senão:
 - Imprimir mensagem de falha na sincronização
- Configurar servidor e callback do cliente MQTT

5 - Função para Configurar o Wi-Fi (setup_wifi)

- Imprimir mensagem de conexão ao Wi-Fi
- Iniciar conexão com o SSID e senha definidos
- Enquanto o Wi-Fi não estiver conectado:
 - Aguardar 500 ms
- Imprimir ponto no monitor serial
- Imprimir mensagem de Wi-Fi conectado

6 - Função de Reconexão ao MQTT (reconnect)

- Enquanto o cliente MQTT não estiver conectado:
 - Imprimir mensagem de tentativa de conexão
 - Se conseguir conectar ao broker MQTT:
 - Imprimir mensagem de sucesso
 - Inscrever-se no tópico "irrigador/controle"
 - Senão:
 - Imprimir código de erro e aguardar 5 segundos antes de tentar novamente

7 - Função de Callback do MQTT (callback)

Quando uma mensagem MQTT é recebida:

- Converter payload em string

- Imprimir tópico e mensagem recebida no monitor serial

- Se o tópico for "irrigador/controle":

 - Se a mensagem for "ligar":

 - Acionar o relé (ligar a bomba)

 - Publicar mensagem de status no tópico "irrigador/status"

 - Imprimir mensagem de relé ligado no monitor serial

 - Se a mensagem for "desligar":

 - Desligar o relé

 - Publicar mensagem de status no tópico "irrigador/status"

 - Imprimir mensagem de relé desligado no monitor serial

8 - Loop Principal do Programa (loop)

Se o cliente MQTT não estiver conectado:

- Chamar função reconnect() para reconectar

Executar loop do cliente MQTT

Obter tempo atual em milissegundos (currentTime)

Se o intervalo de tempo desde a última leitura for maior ou igual a checkInterval:

- Atualizar lastCheckTime com currentTime

- Ler valor analógico do sensor de umidade (ValAnalogIn)

- Mapear ValAnalogIn para umidade percentual (UmidadeAtual) usando calibração

- Restringir UmidadeAtual para valores entre 0 e 100%

- Obter timestamp atual chamando getUnixTimestamp()

- Se o timestamp for válido (diferente de 0):

 - Se a diferença entre UmidadeAtual e UmidadeAnterior for maior ou igual a THRESHOLD ou se for a primeira leitura:

 - Criar payload em formato JSON com umidade e timestamp

 - Publicar payload no tópico "sensor/umidade"

 - Imprimir payload publicado no monitor serial

 - Atualizar UmidadeAnterior com UmidadeAtual

 - Senão:

 - Imprimir mensagem de nenhuma alteração significativa

 - Senão:

 - Imprimir mensagem de erro de timestamp inválido

 - Imprimir valores de debug no monitor serial (Valor Analógico, Umidade Atual, Umidade Anterior)

2.13. Descrição Geral do Funcionamento

Inicialização:

- O dispositivo se conecta à rede Wi-Fi especificada.
- Sincroniza o tempo com um servidor NTP.
- Configura a conexão com o broker MQTT (Mosquitto) e se inscreve nos tópicos necessários.

Leitura do Sensor de Umidade:

- A cada intervalo definido (5 segundos), o sensor de umidade é lido.
- O valor analógico é convertido para umidade percentual, calibrado entre 0% (solo seco) e 100% (solo totalmente molhado).

Envio de Dados via MQTT:

- Se houver uma mudança significativa na umidade (definida pelo THRESHOLD), ou se for a primeira leitura, os dados são enviados ao broker MQTT.
- O payload inclui a umidade atual e o timestamp em milissegundos.

Recebimento de Comandos via MQTT:

- O dispositivo escuta por mensagens no tópico "irrigador/controle".
- Comandos como "ligar" ou "desligar" controlam o estado do relé, acionando ou desligando a bomba de irrigação.
- Respostas de status são publicadas no tópico "irrigador/status".

Manutenção da Conexão:

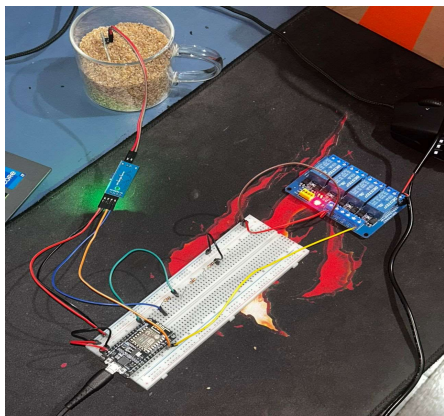
- O programa verifica constantemente se está conectado ao broker MQTT e reconecta se necessário.
- O loop do cliente MQTT é executado continuamente para processar mensagens.

3. Resultados

O circuito e os dispositivos do projeto Irrigafácil conseguiram se comunicar com sucesso utilizando os protocolos definidos. O sistema foi capaz de monitorar a umidade do solo e enviar os valores ao usuário via protocolo MQTT. Utilizando ferramentas como o Node-RED e aplicativos móveis MQTT (MQTT PANEL), foi possível testar a conexão com o broker e verificar os dados transmitidos, confirmando que o projeto funcionava dentro do esperado. Durante a execução, a maior dificuldade enfrentada foi relacionada à sincronização do tempo via NTP e à estabilidade da conexão MQTT, desafios que foram superados com ajustes no código e na configuração dos componentes.

Após entender plenamente a lógica do projeto, constatamos que ele é simples e eficiente. Adquirindo todos os componentes necessários, é possível replicar facilmente este sistema. A montagem deve seguir o diagrama do circuito e as orientações fornecidas. Para facilitar a reprodução e o entendimento, disponibilizamos um repositório com toda a documentação e códigos necessários para a execução deste projeto, que pode ser acessado em https://github.com/Fharaujok/ObjetosInteligentes_IrrigaFacil. Além disso, um vídeo de demonstração está disponível em <https://youtu.be/imMiZyLVVk> mostrando o funcionamento do sistema na prática e auxiliando interessados em desenvolver projetos semelhantes.

Fotos do Hardware montado:



Para avaliar o desempenho do sistema, realizei medições do tempo médio de resposta em dois aspectos fundamentais:

1. Tempo entre o envio de comandos e a ação do atuador (bomba de água):

Esta métrica avalia a rapidez com que o sistema responde aos comandos enviados pelo usuário através do protocolo MQTT, acionando o relé que controla a bomba de água.

2. Tempo entre a detecção pelo sensor de umidade e o recebimento dos dados na plataforma MQTT: Esta métrica verifica a eficiência na coleta e transmissão dos dados do sensor, garantindo que as leituras de umidade do solo sejam atualizadas em tempo hábil para tomadas de decisão.

Foram realizadas quatro medições para cada aspecto mencionado. Os valores obtidos foram registrados e utilizados para calcular o tempo médio de resposta. As tabelas com os resultados dessas medições estão apresentadas a seguir.

Núm. medida	Sensor/Atuador	Tempo de Resposta /MS
1	Atuador – Ligando	642
2	Atuador – Ligando	680
3	Atuador – Ligando	740
4	Atuador – Ligando	635
Tempo Médio de Resposta		647,25

Núm. medida	Sensor/Atuador	Tempo de Resposta /MS
1	Atuador – Desligando	551
2	Atuador – Desligando	557
3	Atuador – Desligando	601
4	Atuador – Desligando	532
Tempo Médio de Resposta		560,25

Núm. medida	Sensor/Atuador	Tempo de Resposta /MS
1	Sensor de Umidade	402
2	Sensor de Umidade	408
3	Sensor de Umidade	401
4	Sensor de Umidade	415
Tempo Médio de Resposta		406,5

4. Conclusão

O projeto **IrrigaFácil** alcançou com sucesso os objetivos estabelecidos, comprovando a eficácia de um sistema de irrigação automatizado baseado em tecnologias de Internet das Coisas (IoT). A implementação permitiu o monitoramento preciso da umidade do solo e o acionamento inteligente da irrigação, com controle remoto em tempo real através do protocolo MQTT. A integração harmoniosa entre hardware e software demonstrou que soluções acessíveis podem ser empregadas para otimizar o uso de recursos hídricos, facilitando o manejo de plantas em diversos contextos, desde jardins residenciais até aplicações agrícolas de maior escala.

Durante o desenvolvimento, desafios significativos foram enfrentados, especialmente relacionados à sincronização do tempo via NTP e à estabilidade da conexão com o broker MQTT. A resolução desses problemas envolveu a adoção de bibliotecas específicas, como a `NTPClient`, e ajustes no código para garantir que o dispositivo aguardasse a sincronização completa antes de prosseguir com as operações. Implementações de rotinas de reconexão automática ao MQTT também foram essenciais para assegurar a continuidade da comunicação, resultando em um sistema mais robusto e confiável.

O **IrrigaFácil** apresenta vantagens notáveis, como a automatização da irrigação baseada em dados reais, monitoramento remoto, baixo custo e escalabilidade para expansão futura. Contudo, também possui algumas limitações, como a dependência de uma conexão estável à internet e a necessidade de sensores mais duráveis para uso prolongado. Para aprimorar o projeto, futuras melhorias poderiam incluir a implementação de medidas de segurança na comunicação MQTT, o uso de sensores de umidade mais robustos, a incorporação de fontes de energia renovável e a integração com dados climáticos para otimizar ainda mais o sistema. Essas evoluções potencializariam o impacto do projeto, tornando-o mais resiliente, sustentável e adequado para uma ampla gama de aplicações.

5. Referências

- SOUZA, Rodrigo Otávio Rodrigues de Melo; BOTREL, Tarlei Arriel. Desenvolvimento e avaliação de um sistema de irrigação automatizado para áreas experimentais. *Irriga*, v. 14, n. 3, p. 365–382, 2009. Esse estudo aborda a criação e avaliação de um sistema de irrigação automatizado para uso experimental, com foco na eficiência do uso da água e baixo custo operacional. Disponível em: <https://doi.org/10.15809/irriga.2009v14n3p365-382>.
- SANTANA, Guilherme Bispo; SILVA, Thiago Claudio Alcantara da. Automatização de um sistema de irrigação com ESP32, com controle e monitoramento através do Blynk IoT. Trabalho de Conclusão de Curso (Bacharelado de Engenharia de Controle e Automação) - Instituto Federal de São Paulo, 2023. Este trabalho apresenta o desenvolvimento de um sistema de irrigação automatizado utilizando o microcontrolador ESP32 e a plataforma Blynk IoT, com foco na otimização do uso de água e monitoramento remoto. Disponível em: <https://repositorio.ifsp.edu.br/handle/123456789/843>.
- JHA, K. et al. Internet of Things (IoT)-based smart agriculture: Towards making the fields talk. *IEEE Access*, v. 7, p. 129551-129583, 2019. Disponível em: <https://ieeexplore.ieee.org/document/8790731>. Acesso em: 5 set. 2024.