

PROJETO DE AUTOMAÇÃO RESIDENCIAL PARA UM JARDIM SUSPENSO COM SENSORES DE TEMPERATURA, UMIDADE DO AR E UMIDADE DO SOLO, CONTROLE E PROGRAMAÇÃO PARA ESP 8266

HOME AUTOMATION PROJECT FOR A HANGING GARDEN WITH TEMPERATURE, AIR HUMIDITY AND SOIL HUMIDITY SENSORS, CONTROL AND PROGRAMMING FOR ESP 8266

Francisco José da Costa Guimarães¹
Alex Franco Ferreira²

RESUMO: A tecnologia da informação assim como a internet das coisas (IoT) têm evoluído continuamente nas últimas décadas, ao ponto desse campo do conhecimento associado ao aumento do processamento de sinais e a inteligência artificial estarem empurrando a evolução natural do controle de processos, a domótica em aplicativos móveis via nuvem e a automatização como realidade social do amanhã. Sistemas de irrigação inteligentes associados ao controle automatizado e IoT, representam uma sofisticada e atual forma de utilização mais eficiente de recursos hídricos e essa variedade de aplicações pode ser utilizada em projetos urbanos. O atual projeto visou automatizar um jardim suspenso utilizando um microcontrolador ESP8266 para monitorar condições ambientais e controlar a irrigação. O sistema foi aplicado a um jardim suspenso com três fileiras de plantas. Cada fileira teve seu próprio sensor de umidade do solo. Um único sensor DHT mediu a temperatura e umidade do ambiente próximo ao jardim. Um único sistema de bomba/relé controlou a irrigação para todas as fileiras simultaneamente, controlando continuamente todas as filas. Com esse trabalho determinou-se o melhor suprimento de água requerido para as filas sem desperdício usando a bomba acoplada automaticamente ao controlador levando em consideração valores úteis em trabalhos bibliográficos pretéritos. Com isso apresentou-se um sistema de irrigação mais eficaz e inteligente capaz de auxiliar na economia de tempo, reduzir a perda de água que não é utilizada pelos vegetais na rega manual, sendo assim, pode-se melhorar a eficiência de um sistema de irrigação de jardins fazendo uso correto da automação mais simples e à baixo custo.

2721

Palavras-chave: Microcontrolador. ESP8266. Domótica. Automação Residencial. Irrigação de Jardins.

¹Engenheiro Eletricista, discente do curso de pós-graduação em Automação Industrial da Universidade de Vassouras.

²Engenheiro Eletricista, docente do curso de pós-graduação em Automação Industrial da Universidade de Vassouras.

ABSTRACT: Information technology, as well as the Internet of Things (IoT), has continuously evolved over the past decades to the point where this field of knowledge, combined with advances in signal processing and artificial intelligence, is driving the natural evolution of process control, home automation through cloud-based mobile applications, and automation as a social reality of the future. Smart irrigation systems integrated with automated control and IoT represent a sophisticated and up-to-date way of using water resources more efficiently, and this variety of applications can be used in urban projects. The present project aimed to automate a vertical garden using an ESP8266 microcontroller to monitor environmental conditions and control irrigation. The system was applied to a vertical garden with three rows of plants, each having its own soil moisture sensor. A single DHT sensor measured the temperature and humidity of the environment near the garden. A single pump and relay system controlled irrigation for all rows simultaneously, continuously managing all lines. This study determined the optimal water supply required for the rows without waste, using a pump automatically triggered by the controller, taking into account useful reference values from previous academic works. As a result, a more effective and intelligent irrigation system was developed, capable of saving time, reducing water loss that occurs with manual watering, and improving the efficiency of garden irrigation systems through the proper use of simple and low-cost automation.

Keywords: Microcontroller, ESP8266, Home Automation, Domotics System, Garden Irrigation.

INTRODUÇÃO

A tecnologia da informação assim como a internet das coisas (IoT) têm evoluído continuamente nas últimas décadas do século XXI determinando que no âmbito da automação residencial, qualquer melhoria no conforto residencial, pode prover informações disponíveis, utilizáveis e evolução do conforto e praticidade no processo de controle residencial. O aumento do processamento de sinais e a inteligência artificial também estão empurrando a evolução natural do controle de processos, a domótica em aplicativos móveis via nuvem e a automatização do mundo do amanhã, (Estevan Jr. 2019).

Com o desenvolvimento das cidades e aumento numérico da população global, sistemas automatizados, que permitam um menor desperdício de recursos cada vez mais escassos, seja de energia como de água como recursos reutilizáveis pode ser uma ótima nova proposta para desenvolver lado a lado a automação de lares. Vários experimentos já demonstraram que existe uma proposta de aproveitamento e redução de mais de 60% do total utilizado em volume hídrico comparando sistemas automatizados e irrigação por mecanismos tradicionais como rega manual por metro quadrado de área plantada o que mostra uma grande eficiência metodológica em automatizar a irrigação, (Pawar et. al. 2024).

A implementação de sistemas simples e sustentáveis que tragam benefícios e facilidades para as pessoas, junto com o desenvolvimento e a ampla utilização da IoT e automação em

sistemas de irrigação mais inteligentes surgem como uma solução para tais desafios. Paralelamente, o rápido avanço nas tecnologias de comunicação e sensores, incluindo redes de sensores sem fio, a relação direta entre comunicação de redes e a própria internet das coisas com a evolução assustadora da inteligência artificial, o aprendizado de máquina, e a quantidade cada vez maior de microcontroladores e *shields* associados a estes facilitaram o desenvolvimento do processo de controle/automação.

O artigo teve o objetivo de, a partir da programação no software Arduino IDE, propor um projeto de controle e irrigação de jardim através de monitoramento de variáveis de temperatura e umidade. O sistema foi aplicado a um jardim suspenso com três fileiras de plantas, porém pode ser amplamente dimensionado para outras quantidades. Cada fileira teve seu próprio sensor de umidade do solo. Um único sensor DHT mediu a temperatura e umidade do ambiente próximo ao jardim. Um único sistema de bomba/relé controlou a irrigação para todas as fileiras simultaneamente através da rede de abastecimento de água ou de um reservatório de água simples. E a medição em tempo real das variáveis a seguir: Monitoramento contínuo da temperatura do ar; Monitoramento contínuo da umidade do ar; Monitoramento contínuo da umidade do solo em diferentes pontos (fileiras unitárias de plantas no jardim); Acionamento automático de um sistema de irrigação baseado nos níveis de umidade do solo; Disponibilização dos dados dos sensores e status do sistema via interface web local.

2723

Mais especificamente o atual estudo visou propor um projeto de baixo custo baseado num sistema de irrigação inteligente associado ao controle lógico via microcontrolador em resposta a dados de sensibilidade fornecidos por sensores de umidade e temperatura do solo e do ar. Foi descrito o desenvolvimento e o desenho do projeto, inclusive com objetivo de aproveitamento de água da chuva. Os sensores utilizados, o microcontrolador, e o módulo Wi-Fi para automação do processo de controle de irrigação. A arquitetura básica do projeto foi proposta, os componentes de hardware, ferramentas de software e o código algoritmo usado para essa aplicação.

Tal tecnologia fez dos sistemas automatizados de irrigação de pequenos sistemas, uma ótima alternativa para aproveitamento de água de chuva, que como no atual projeto pode ser utilizada para irrigação. Essa proposta de utilizar sistemas de irrigação inteligente baseados em inovações de IoT para conservação de água já foi abordada em outras publicações, (Gawali et. al. 2024).

Como muitos outros trabalhos orientados à irrigação com sistemas de monitoramento e controle automatizado, a proposta inicial foi criar um projeto de sistema que consegue perceber

variações nas variáveis foco do sistema, que neste caso são ambas, temperatura e umidade do solo/ar por serem vitais ao bom desenvolvimento de plantas em jardins e garantir um controle simples e atuação segundo esse último para a demanda de irrigação.

Segundo VanDerZanden (2008), os principais fatores ambientais que afetam o crescimento das plantas incluem luz, temperatura, água, umidade e nutrição.

UMIDADE

A umidade do ar, do solo e a temperatura são fatores climáticos cruciais para o desenvolvimento saudável das plantas. A combinação ideal desses fatores varia de acordo com a espécie, mas em geral, a faixa de umidade do solo entre 20% e 60% é considerada adequada para a maioria das culturas (Nutrição de Safras, 2025). Temperaturas muito baixas ou muito altas podem prejudicar o crescimento, a floração e a frutificação. A umidade do ar também é importante, pois a falta ou o excesso podem afetar negativamente o desenvolvimento das plantas, (VanDerZanden, 2008).

UMIDADE DO SOLO

A umidade ideal do solo varia entre 20% e 60% para a maioria das culturas. A quantidade de água disponível no solo influencia a absorção de nutrientes e o crescimento das plantas. O excesso de água pode levar ao apodrecimento das raízes, enquanto a falta de água pode causar estresse hídrico, (Nutrição de Safras, 2025; VanDerZanden, 2008).

UMIDADE DO AR

A umidade relativa do ar ideal varia conforme a cultura, mas níveis muito altos ou muito baixos podem prejudicar o desenvolvimento. Baixa umidade relativa pode causar perda de água por evaporação, afetando a fotossíntese e o crescimento. Já a alta umidade relativa pode favorecer o desenvolvimento de doenças fúngicas, (Nutrição de Safras, 2025; VanDerZanden, 2008).

TEMPERATURA

A temperatura ideal para o desenvolvimento das plantas varia de acordo com a espécie, mas a faixa entre 20°C e 30°C é geralmente favorável. Temperaturas muito baixas podem retardar o crescimento e o desenvolvimento. Temperaturas muito altas podem causar estresse

térmico, afetando a fotossíntese e a produção de frutos, (Nutrição de Safras, 2025; VanDerZanden, 2008).

RELAÇÕES

A temperatura e a umidade do solo estão interligadas, com temperaturas mais altas levando a uma maior taxa de evaporação e, conseqüentemente, a uma redução da umidade do solo. A umidade do ar influencia a taxa de transpiração das plantas, que por sua vez afeta a absorção de água e nutrientes do solo. O controle adequado desses fatores é fundamental para otimizar o crescimento e a produtividade das plantas, (Portal Embrapa, 2025).

A umidade do ar, do solo e a temperatura são fatores ambientais cruciais para o desenvolvimento saudável de plantas em jardins. A faixa ideal de umidade do ar para a maioria das plantas de jardins situa-se entre 50% e 70%, (Senado Federal, 2024).

A umidade do solo ideal pode variar, mas geralmente situa-se entre 20% e 60% do volume total do solo (EOS Data Analytics, 2025).

Temperaturas ideais variam conforme a espécie, mas muitas plantas de jardim prosperam em temperaturas entre 23°C e 25°C, (Barros, 2012).

2725

MICROCONTROLADORES: ESP32 x ESP8266

Vale ressaltar aqui, as diferenças, porém, sem determinar qual é o melhor ou pior equipamento, e aqui, se evidencia o momento impar que a humanidade está vivendo, com a tecnologia sendo integrada às casas e vidas das pessoas comuns. As diferenças são:

Processamento: O ESP32 possui um processador dual-core com clock de até 240 MHz, enquanto o ESP8266 apresenta um single-core com 80 MHz ou 160 MHz de clock, (Batrinu, 2018).

Memória: O ESP32, uns microcontrolados de 32 bits, geralmente oferece mais memória RAM, já o ESP8266, um microcontrolador RISC de 8 a 32 bits, tem 64 KiB de RAM para instruções e 96 KiB para dados, o ESP32 permite lidar com tarefas mais complexas e projetos maiores, o ESP8266 é mais indicado para utilizações menores e mais simples, (Stevan Jr., 2019).

Portas GPIO: O ESP32 possui mais portas de entrada/saída (GPIO) do que o ESP8266, o que aumenta a flexibilidade para conectar diversos sensores e atuadores. O chip ESP32 possui maior poder de processamento do que o ESP8266. Também possui 36 portas GPIO contra 17 do ESP8266, além de 16 canais PWM ao invés de, mais canais analógicos e digitais e mais interfaces SPI e I2C, (Batrinu, 2018).

Conectividade: Ambos possuem Wi-Fi, o ESP32 oferece suporte a Bluetooth e Bluetooth Low Energy (BLE), (Oliveira, 2021).

Custo: O ESP8266 geralmente é mais barato que o ESP32, tornando-o uma boa opção para projetos com restrições de orçamento, porém esses custos entre as duas placas estão quase que pareados, estando os desenvolvedores num nível de escolha de complexidade baseada em demanda de projeto, não tato em relação à custos, (Batrinu, 2018).

Complexidade: O ESP32 pode ser mais complexo de programar devido à sua maior quantidade de recursos, mas também oferece mais opções para projetos avançados, (Svitla, 2020).

Enfim, o ESP8266 é um microcontrolador de baixo custo e baixo consumo de energia, que é amplamente utilizado em projetos IoT (Internet das coisas) e em projetos de automação residencial. Ele possui uma CPU de 32 bits, suporta WiFi e tem uma memória flash de até 4MB, já o ESP32 é um microcontrolador de maior desempenho e mais recursos, possui uma CPU de 32 bits dual-core, apresenta WiFi, Bluetooth e tem uma memória flash de até 8MB. Ele é geralmente utilizado em projetos mais complexos, (Svitla, 2020).

MICROCONTROLADOR ESP8266

2726

O ESP8266, foi desenvolvido pela empresa chinesa EspressIf Systems, pode implementar controles dos mais variados níveis via programação em Arduino IDE, isso o torna uma ferramenta muito útil pois abre possibilidade para liberdade de programação de diferentes códigos de acordo com a demanda de seus desenvolvedores que estão buscando maneiras melhores de acessar informações e controlar o ambiente residencial. Essa liberdade de programação é possível graças aos SDKs (Kits de desenvolvimento de softwares), tanto os oficiais da empresa que produz o chip ESP8266 quanto os SDKs sem sistema operacional (NonOS) e os SDKs FreeRTOS para o código de programação C/C++ largamente utilizado (Batrinu, 2018).

Esses kits de desenvolvimento de software são um conjunto de ferramentas de criação específicas da plataforma orientada para os desenvolvedores, eles carregam um conjunto de componentes (bibliotecas, compiladores e depuradores) específicos para auxiliar na criação de códigos que sejam utilizáveis e executáveis em sistemas operacionais, plataformas e linguagens de programação das mais diversas fontes e para as mais amplas utilizações (Amazon). Tanto os SDKs quanto o Arduino IDE (Ambiente de desenvolvimento integrado) foram ferramentas de

desenvolvimento de softwares que geraram enormes vantagens para a domótica ter se desenvolvido numa velocidade tão grande nos últimos anos.

Outro fator de grande importância nos dias de hoje é a gestão/conservação de recursos hídricos e o aproveitamento de água de diferentes fontes, (Gawali et al, 2024).

Recentemente, o uso do módulo Wi-Fi ESP 8266, tornou-se muito popular. O chip foi projetado para uso em tomadas inteligentes, redes mesh, câmeras IP, sensores sem fio, eletrônicos vestíveis e assim por diante. O consumo de energia do ESP8266 é comparativamente baixo para soluções com Wi-Fi. Além disso, ele possui opção para entrar no modo de hibernação, onde o consumo de energia é incrivelmente baixo (1,0 mW), (Batrinu, 2018).

Existem duas opções para usar o chip: Através da ponte UART-WIFI quando o módulo baseado em ESP8266 se conecta a uma solução existente baseada em qualquer outro microcontrolador e é controlado por comandos AT, proporcionando comunicação com a infraestrutura Wi-Fi; implementando uma nova solução utilizando o próprio chip ESP8266 como módulo de controle pelo microcontrolador. (Ideali, 2021)

As especificações para o ESP8266 são o suporte para protocolos WiFi 802.11 b/g/n; Wi-Fi Direct (P2P), Pilha TCP/IP integrada; Interruptor TR integrado, balun, LNA, amplificador de potência e correspondência de rede PLL, reguladores e sistema de gerenciamento de energia integrados; potência de saída +19,5 dBm no modo 802.11b; memória do programa: 1 MB; SDIO 2.0, SPI, UART; STBC, 1x1 MIMO, 2x1 MIMO; envio de pacotes: até 22 ms; consumo em modo Standby (extremamente baixo) até 1,0 mW (DTIM3); Dimensões de 24,5x14 mm; conexões Vcc - fonte de alimentação, + 3,3V (máximo 3,6V); GND - terra comum; TXD - transmissão de dados (nível 3,3 V); RXD - receptor de dados (nível 3,3 V); CH_PD - desliga o módulo (nível baixo ativo, para ligar basta enviar o sinal Vcc); GPIO0 - pino 0 de uso geral; GPIO2 - pino 2 de uso geral (LED azul, baixo); RST - reinicialização do módulo, (Ideali, 2021).

2727

Ao mesmo tempo, diferentes produtores podem variar os parâmetros do módulo, como contagem de pinos e fator de forma do pacote.

SENSOR DHT22 (AM2302)

O Sensor de Umidade e Temperatura DHT22 / AM2302 é um dos componentes mais utilizados em projetos que envolva medição de temperatura e umidade ambiente. Este sensor faz medições de temperatura de -40º até 80º celsius e mede a umidade do ar nas faixas de 0 a 100%. A precisão (margem de erro) do sensor para medição de temperatura é de aproximadamente 0,5º celsius e para umidade é de 2%, (Stevan Jr., 2015; Eletrogate, 2025).

Especificações e características:

Modelo: AM2302; Faixa de medição (umidade): 0 a 100%; Faixa de medição (temperatura): -40° a 80°C; Tensão de operação: 3 – 5VDC; Corrente de operação: 2,5mA; Corrente em stand by: 100uA a 150 uA; Precisão (umidade): $\pm 2\%$; Precisão (temperatura): $\pm 0,5^\circ$ celsius; Tempo de resposta: 2s, (Stevan Jr., 2019)

Aplicações:

Projetos com finalidade de medir umidade e temperatura ambiente utilizando Arduino ou outras plataformas microcontroladas. Utilizar o Sensor de Umidade e Temperatura DHT22 / AM2302 em conjunto com o Arduino e medir umidade e temperatura ambiente. O valor da umidade e da temperatura é exibido no monitor serial do ambiente de programação do Arduino.

SENSOR CAPACITIVO DE UMIDADE DO SOLO V1.2 (OU V2.0).

Monitoramento da umidade do solo em projetos de automação ou cultivo inteligente, Sensor Capacitivo de Umidade do Solo V2.0 ao contrário dos sensores resistivos comuns, esse modelo utiliza um princípio capacitivo, com máxima resistência à corrosão, prolongando assim a durabilidade e promovendo medições mais estáveis, (EOS Data Analytics, 2025).

Em resumo, um sensor capacitivo usa o solo como um elemento capacitor e usa a capacidade de armazenamento de carga do solo para calibrar o conteúdo de água. 2728

Então, a escolha por um sensor capacitivo se deve ao fato de o solo ser uma mistura de sólidos, líquidos e gases. Cada um deles tem um dielétrico diferente, mas, no geral, todos têm valores dielétricos baixos em comparação com a água. Desse modo, quando a capacidade de armazenamento de água na porção do solo é verificada por um sensor capacitivo ao invés de um resistivo, a água e o ar são os únicos componentes a mudarem de volume na mistura, dando uma relação de conteúdo de água mais assertiva.

O uso desse sensor é ideal para sistemas de irrigação automatizados, monitoramento de plantas em vasos e sistemas de irrigação mais complexos, consegue detectar com facilidade se o solo está seco ou úmido, ajudando a evitar tanto falta quanto excesso de água. Já apresenta regulador de tensão, para a integração com microcontroladores como Arduino, ESP32, NodeMCU e outros.

Vantagem do uso desse componente é sua compatibilidade com prototipagens rápidas e de baixo custo, ou desenvolvimento de soluções para uso profissional. Tensão de Operação de 3,3 - 6V; Saída Analógica de 0 - 3V; Dimensões de 22 × 102 x 7mm, (Oliveira, 2021).

MÓDULO RELÉ DE 1 CANAL, 5V, OPTOACOPLADO.

O Módulo Relé 1 Canal 5v é um componente eletrônico utilizado para controlar dispositivos de alta potência, como motores e luzes, a partir de sinais de baixa potência, como os gerados por microcontroladores. Ele funciona como um interruptor que pode ser acionado eletronicamente. O módulo relé permite assim, a integração com microcontroladores como Arduino, AVR, PIC, ARM e outros. A partir das saídas digitais pode-se controlar cargas maiores como motores AC ou DC, eletroímãs, solenóides e lâmpadas. O módulo tem um canal sendo assim concebido para ser integrado para controlar até 1 relé, com carga nominal 10A/250VAC, 10A/125VAC, 10A/30VDC. Cada canal possui um LED para indicar o estado da saída, (Eletrogate, 2025).

Especificações técnicas: Tensão de operação 5V DC (VCC e GND), corrente de operação 80 mA, capacidade do relé 30 V DC e 10A ou 250V AC e 10A, tempo de resposta 5~10ms, (Eletrogate, 2025).

MINI BOMBA DE ÁGUA SUBMERSÍVEL DC (EX: 3-6V OU 5V).

Componente que se adequa para uso de projetos de prototipagem, incluindo automação residencial (domótica) e protótipos robóticos baseados em plataformas microcontroladoras como Arduino e outros. O diferencial para a utilização desse componente é a possibilidade de uso submersa na água. Isso permite que envases utilizados para armazenar água pluviométrica possam ser utilizados para reservatórios de água de rega sem comprometer o uso de água de cisternas e aumentar o gasto mensal familiar com esse insumo vital. Com um motor de tamanho adequado essa bomba de água pode impulsionar até 120 litros por hora, com eficácia e precisão medianas, porém aceitáveis para tal demanda. Por ter um tamanho reduzido e baixo peso, pode ser aplicada na grande maioria dos projetos, operando com tensão recomendada de 2,5v a 5v, (Eletrogate, 2025).

Especificações: Tensão DC: 2.5-6 v; Corrente nominal: 130-220mA; Potência: 0.4-1.5 w; Elevação máxima: 40-110 cm/15.75"- 43.4"; Taxa de fluxo: 80-120L/H; Diâmetro de saída de água lado externo: aprox. 7.5mm/0.3"; diâmetro de saída de água lado interno: aprox. 4.7mm/0.18"; Diâmetro: aprox. 24mm/0.95"; Comprimento: aprox. 45mm/1.8"; Altura: aprox. 33mm/1.30"; Comprimento do fio: cerca de 15-20 cm; Material da blindagem: plástico; Modo de condução: brushless DC design, condução magnética; Vida de funcionamento contínuo: 500 horas, (Eletrogate, 2025).

MULTIPLEXADOR ANALÓGICO: CD4051BE OU 74HC4051

O Módulo Multiplexador Analógico de 8 Canais 74HC4051 - CJMCU-4051 é um componente eletrônico utilizado para expandir a capacidade de leitura de sinais analógicos em projetos com microcontroladores, como o Arduino, ESP32 e outros. Ele permite selecionar entre até 8 sinais analógicos diferentes utilizando apenas um pino de leitura analógica do microcontrolador, otimizando o uso dos recursos do sistema. (Smart Project, 2025)

CHIP PRINCIPAL: 74HC4051

Número de Canais: 8 entradas analógicas multiplexadas em uma única saída. (Smart Project, 2025). Tensão de Operação: 2V a 10V (tipicamente 5V com Arduino). Frequência Máxima de Operação: Até 6 MHz. Impedância de Entrada: Alta, permitindo o uso com uma variedade de sensores analógicos. Temperatura de Operação: -40°C a +85°C. Controle de Seleção de Canais: Seleção de canais através de três pinos digitais (S1, S2, S3). Aplicações: Expansão de entradas analógicas em microcontroladores. Sistemas de aquisição de dados analógicos. Controle de vários sensores analógicos com poucos pinos de I/O. Este módulo é ideal para projetos de automação, robótica e sistemas de monitoramento que exigem múltiplos canais analógicos pois ele tem um sistema de comutação para diferentes canais, oferecendo uma solução prática e eficiente para a limitação de entradas analógicas de controladores. No projeto em questão, esse componente será necessário para ler os 3 sensores de umidade do solo usando a única entrada analógica (Ao) do ESP8266, (Smart Project, 2025).

2730

CONSTRUÇÃO DO PROTÓTIPO

Um sistema de irrigação de jardim simples usando o ESP8266 pode ser programado para gerenciar automaticamente o processo, utilizando sensores de umidade do solo, do ar e de temperatura para determinar quando e quanto regar, e um módulo ESP8266 para controlar válvulas solenoides e acionar bombas d'água. O ESP8266, um microcontrolador, recebe os dados dos sensores, compara com valores pré-definidos e controla uma bomba d'água ou válvula solenóide para irrigar a área.

Componentes:

ESP8266: O computador do sistema, responsável por processar os dados e controlar a irrigação.

- Sensor de Umidade do Solo: Mede a umidade do solo, geralmente com um sensor resistivo que varia a resistência conforme a umidade, ou capacitivo como nesse caso por apresentar valores mais precisos num sistema mais simples e com custos mais baixos.
- Sensor de Temperatura e Umidade (DHT11/DHT22): Mede a temperatura e umidade do ar, com as informações necessárias para otimizar a irrigação.
- Bomba d'água ou Válvula Solenóide: Componente para liberar a água para a irrigação das fileiras de plantas.
- Fonte de Alimentação: Para alimentar o ESP8266 e os sensores.
- Módulo Relé: Para controlar a bomba ou válvula solenóide, pois o ESP8266 opera em baixa tensão e o dispositivo de irrigação pode exigir mais tensão e corrente.
- Fios e Cabos: Para conectar os componentes.

Funcionamento:

1. Coleta de Dados: O ESP8266 lê os dados dos sensores de umidade do solo e temperatura/umidade do ar.
2. Processamento: O código no ESP8266 compara os dados lidos com valores pré-definidos de umidade e temperatura para determinar se a irrigação é necessária.
3. Controle: Se a umidade do solo estiver abaixo do limiar definido ou a temperatura e umidade do ar indicarem alta evaporação, o ESP8266 ativa o relé, que aciona a bomba d'água ou válvula solenóide para iniciar a irrigação.
4. Monitoramento: O sistema pode enviar dados para um serviço de monitoramento (como o [2731](#) ThingSpeak) ou exibir as informações em um display.
5. Desligamento: Após um tempo programado ou quando a umidade do solo atingir o nível desejado, o ESP8266 desativa o relé, parando a irrigação.

Programação:

Inicialmente deve-se realizar a configuração dos Sensores para sua inicialização e leitura dos dados. Configuração da Conexão Wi-Fi que é a conexão do ESP8266 à rede Wi-Fi. Definição dos Limites para estabelecer os valores de umidade e temperatura para acionar a irrigação. Controle através da programação da ativação e desativação da bomba ou válvula. Comunicação: Envio de dados para serviços de monitoramento.

Vantagens:

Economia de água: Irrigação somente quando necessário, evitando o desperdício. Melhor aproveitamento dos recursos, redução de custos, elimina a necessidade de regar manualmente, e monitoramento remoto, permitindo acompanhamento do sistema. Flexibilidade que se reflete em poder adaptar o sistema para diferentes tipos de plantas e sistemas de irrigação.

Lista de materiais:

Esta lista detalha os componentes eletrônicos e materiais necessários para a construção do sistema de automação do jardim suspenso com 3 fileiras, conforme os requisitos definidos.

Componentes necessários ao projeto:

1. Microcontrolador:

Placa de desenvolvimento baseada no ESP8266 (Ex: NodeMCU V₂ ou V₃ LoLin, Wemos D1 Mini). Função: Responsável pelo processamento dos dados dos sensores, lógica de controle e comunicação (se implementada interface web).

2. Sensor de Temperatura e Umidade do Ar:

Sensor DHT22 (AM2302). Função: Mede a temperatura e umidade do ambiente próximo às plantas. Oferece maior precisão que o DHT11.

3. Sensores de Umidade do Solo:

Sensor Capacitivo de Umidade do Solo V_{1.2} (ou V_{2.0}). Função: Sensores capacitivos são preferíveis aos resistivos por terem maior durabilidade (não corroem). Medem a umidade diretamente no substrato de cada fileira.

4. Módulo Relé:

Módulo Relé de 1 Canal, 5V, optoacoplado. Função: Utilizado para acionar a bomba de água, que geralmente opera com tensão/corrente diferente e maior que a suportada diretamente pelo ESP8266. O optoacoplamento oferece isolamento elétrico.

5. Bomba de Água:

Mini Bomba de Água Submersível DC (Ex: 3-6V ou 5V).

6. Mangueiras e Conexões:

Tubulação fina (ex: 4mm ou 6mm de diâmetro interno), conectores T, cotovelos, tampões. Comprimento e número de conexões suficientes para alcançar as 3 fileiras a partir da bomba. Função: Distribuição da água da bomba até os pontos de irrigação.

7. Gotejadores/Emitters:

Gotejadores ajustáveis ou fixos. Pelo menos 3 (um por fileira), idealmente um por planta ou vaso. Função: Para liberar a água de forma controlada perto das raízes das plantas.

8. Fonte de Alimentação:

Adaptador AC/DC 5V (1 unidade - 3A).

Função: Para alimentar o ESP8266, os sensores e o módulo relé. Verificar se a corrente é suficiente caso a bomba de 5V seja alimentada pela mesma fonte. Ou fonte adicional unitária para bomba se essa for de tensão diferente (12V).

9. Fios de Conexão (Jumpers):

Conjunto de jumpers Macho-Macho, Macho-Fêmea, Fêmea-Fêmea.

Função: Para conectar os componentes na protoboard durante a montagem e teste.

10. Protoboard (Placa de Ensaio):

Breadboard de 400 ou 830 pontos (1 unidade). **Função:** Facilitar a montagem e teste do circuito sem necessidade de solda.

Componentes opcionais ao projeto:

1. Sensor de Nível de Água:

Sensor de nível de água (tipo boia ou eletrônico).

Função: Útil se for utilizado um reservatório, para monitorar o nível e evitar que a bomba funcione a seco.

2. Caixa/Gabinete:

Caixa plástica ou de outro material resistente.

Função: Para proteger os componentes eletrônicos da umidade, poeira e possíveis danos físicos.

3. Multiplexador Analógico:

Uma unidade de CD4051BE (ou similar, como 74HC4051).

Função: Necessário para ler os 3 sensores de umidade do solo usando a única entrada analógica (Ao) do ESP8266.

Requisitos de Hardware

Microcontrolador:

ESP8266 (NodeMCU ou similar).

Sensores:

Sensor de Temperatura e Umidade do Ar: DHT11 ou DHT22 (DHT22 preferível pela maior precisão).

Sensores de Umidade do Solo: Sensor Capacitivo de Umidade do Solo V1.2 (ou similar) - um por fileira (total de 3 sensores).

Atuador:

Módulo Relé (5V) para acionar a bomba de água.

Bomba de água submersível (compatível com a tensão/corrente do relé e fonte, ex: 5V ou 12V).

Alimentação: Fonte de alimentação adequada para o ESP8266, sensores e bomba de água.

Conexões: Jumpers, protoboard (para prototipagem), fiação adequada para instalação final.

Sistema de Irrigação: Mangueiras, conectores, gotejadores (adequados para jardim suspenso).

Requisitos de Software

- Firmware (ESP8266): Código em C++/Arduino para ler os sensores, aplicar a lógica de controle e acionar o relé.
- Lógica de Controle:
- Ler sensores periodicamente (ex: a cada 5-10 minutos).
- Calcular a média (ou usar leitura individual) da umidade do solo.
- Se a umidade do solo estiver abaixo de um limiar pré-definido, acionar a bomba de água por um período determinado.
- Evitar acionamentos muito frequentes.
- Interface (Opcional, mas recomendada):
- Servidor web embarcado no ESP8266 para exibir leituras dos sensores e status da bomba.
- Possibilidade de ajuste de parâmetros (limiar de umidade, tempo de irrigação) via interface web.

2734

INFORMAÇÕES ADICIONAIS DE CALIBRAÇÃO DE UMIDADE

Cálculo do Sistema de Irrigação por Sensor: O reservatório circular de cada planta possui uma área de 0,045 metros quadrados, calculada pela fórmula πr^2 . A vazão de saturação, Q_I , é de 7,67 mililitros por segundo para grandes culturas. O sensor no solo registra um valor setado de secura de 670, que excede o nível máximo de secura definido no programa, acionando a irrigação automática. Isso poderia até ser utilizado em sistemas maiores e mais robustos, porém aqui, é mais interessante acionar a irrigação por um pulso pois a umidade é medida em intervalos de tempo fixos, (Gawali et al., 2025).

Após 1 minuto de irrigação, espera-se que a demanda de água por área circular de cada planta em cada fileira seja atendida e o motor deve parar automaticamente. Se, alguns segundos ou minutos depois disso, o solo da planta ficar novamente seco e esse nível cair para 430, isso fará com que a bomba solenóide reinicie e pare a irrigação automaticamente, mantendo um valor constante de umidade. No presente trabalho, os valores de saída de tensão são convertidos analógico-digitalmente para um intervalo entre 0 e 1.023, então pode-se utilizar entre 750 a 1.023

como baixa umidade, 500 a 749 como média e o a 499 como alta, mas foi mais fácil e rápido definir os limiares em 350 = ambiente seco e 800 = ambiente úmido. Note que a automação permite uma gama de definições amplas para variáveis e limites que se quer controlar.

Área do vaso circular (tanque da planta), $= \pi r^2 = 0,045 \text{ m}^2$ ($\pm 12 \text{ cm}$ de raio) Descarga total por saturação, $Q_I = [V (\text{volume}) / T (\text{tempo})] \text{ ml} / \text{seg.} = 7,67 \text{ ml/s}$. Mais uma vez aqui reporta-se a não necessidade do estabelecimento de volume ficando tais valores como exemplos para níveis de comparação em razão do volume gasto no sistema mais simples do jardim.

Equação de potência da bomba: Esta equação é usada para calcular a potência necessária para que a bomba de água forneça uma determinada vazão de água. A equação de potência da bomba pode ser expressa como: Potência da bomba = (vazão x pressão) / eficiência da bomba, onde a vazão é o volume de água fornecido por unidade de tempo, a pressão é a força exercida sobre a água pela bomba e a eficiência da bomba é a razão entre a potência de saída e a potência de entrada. Equação da vazão: Esta equação é usada para calcular a vazão de água através de um cano ou canal. A equação da vazão pode ser expressa como: $Q = A \times V$, onde Q é a vazão, A é a área da seção transversal do cano ou canal e V é a velocidade da água. Equação da necessidade hídrica: Esta equação é usada para calcular a quantidade de água necessária para que uma planta cresça e produza um determinado rendimento, (Tolentino, 2023).

2735

A equação da necessidade hídrica pode ser expressa como: Necessidade hídrica = consumo de água pela planta x produtividade (crescimento), onde o consumo de água pela planta é a quantidade total de água necessária à planta durante seu ciclo de crescimento e a produtividade é a quantidade de crescimento por unidade de área.

Equação de potência da bomba: Potência da bomba = (vazão x pressão) / eficiência da bomba. Equação da vazão: $Q = A \times V$. Equação de necessidade de água: Necessidade de água = uso de água da cultura x rendimento. Usar um sistema de irrigação automático economiza água e tempo em comparação à irrigação manual, que requer mais água e é propensa a vazamentos, (Tolentino, 2023).

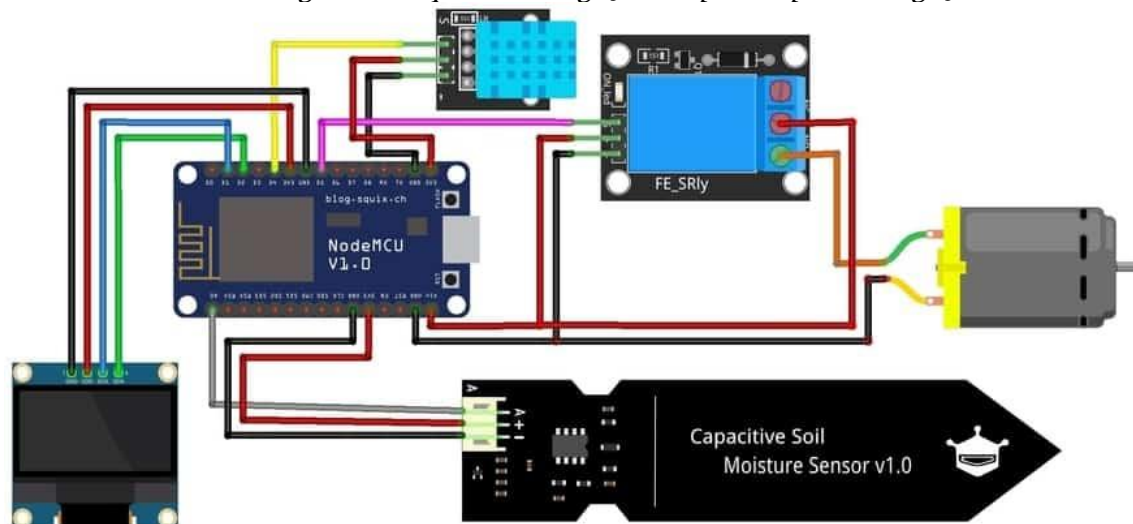
O planejamento ideal da irrigação é importante para maximizar a eficiência do uso da água e atingir outros objetivos, no caso proposto via projeto, o que importa é irrigar um pequeno sistema, porém uma alternativa seria medir o volume gasto no tempo e comparar. Em estudos mais complexos, o valor econômico das aplicações de irrigação pode ser avaliado por meio de funções de produção de água e métodos de análise. Aqui, a lógica é demanda de uso de sistema de domótica versus economia de gasto de água. Então essas informações e dimensionamentos

acima caberiam melhor em análises mais aprofundadas e para sistemas de jardins mais complexos.

ESQUEMA DE MONTAGEM

Na Figura 1 pode se observar o diagrama de ligação dos componentes do projeto:

Figura 1: Esquema de ligação do protótipo de irrigação.

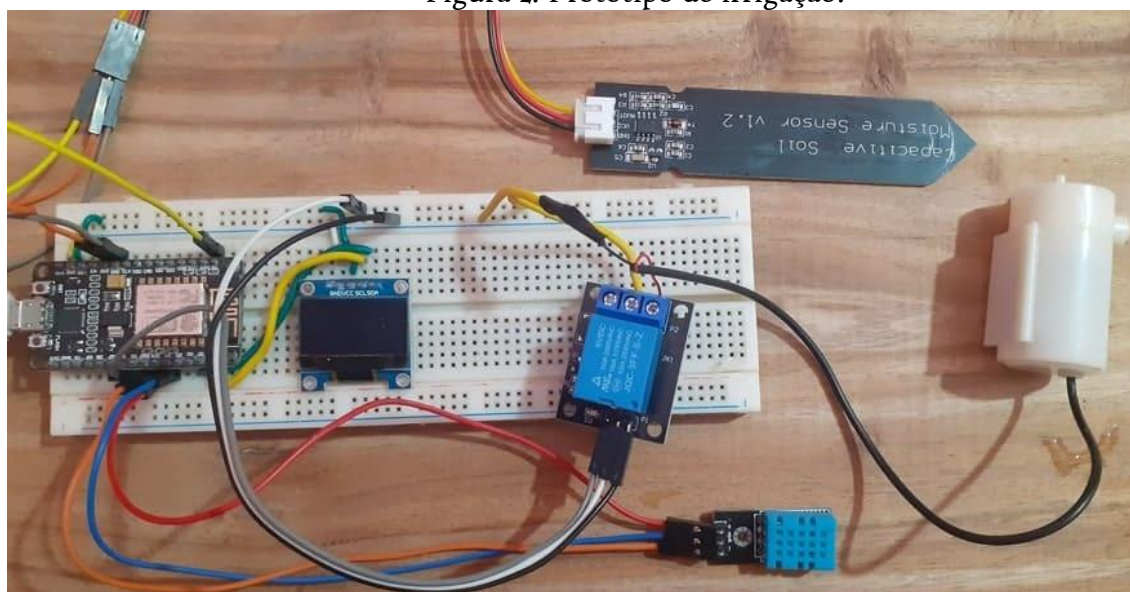


Fonte: Do Autor, 2025

Na Figura 2 pode se observar a imagem do protótipo montado em bancada

2736

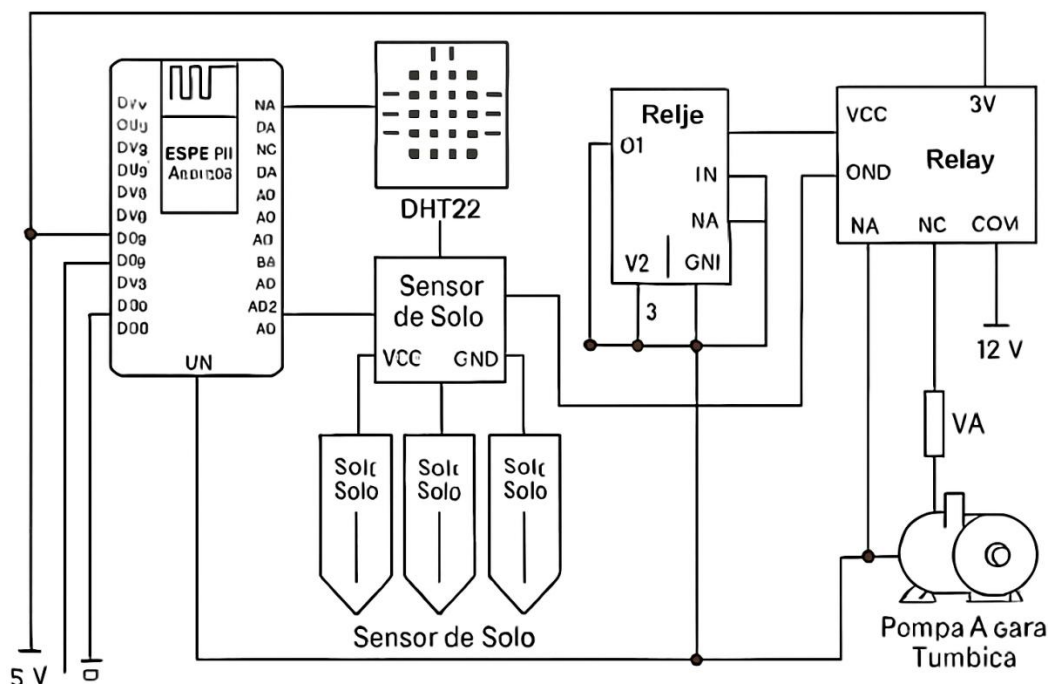
Figura 2: Protótipo de irrigação.



Fonte: Do Autor, 2025

Na Figura 3 observa-se o esquema elétrico de ligações entre sensores, controlador e atuadores.

Figura 3: Esquema elétrico de ligações dos componentes e controlador.



Fonte: Do Autor, 2025

CÓDIGO DE PROGRAMAÇÃO - ESP 8266

O código realizado na IDE do Arduino em C++ é apresentado a seguir.

2737

```
// Código para Automação de Jardim Suspenso com ESP8266
// Baseado no diagrama com DHT22, 3 Sensores de Umidade do Solo (via Mux CD4051BE) e
// Relé
// Inclusão de Bibliotecas
#include "DHT.h" // Biblioteca para o sensor DHT
// --- Definições de Pinos (Baseado no NodeMCU ESP8266) ---
#define DHTPIN D4 // Pino digital conectado ao sensor DHT22
#define RELAY_PIN D1 // Pino digital conectado ao módulo Relé (para a bomba)
// Pinos de Seleção do Multiplexador CD4051BE
#define MUX_So_PIN D5 // Pino So do Mux
#define MUX_S1_PIN D6 // Pino S1 do Mux
#define MUX_S2_PIN D7 // Pino S2 do Mux
// Pino Analógico conectado à saída Z do Multiplexador
#define MUX_Z_PIN A0 // Pino A0 do ESP8266 (única entrada analógica)
// --- Configurações do Sensor DHT ---
#define DHTTYPE DHT22 // Define o tipo de sensor DHT (DHT11, DHT21, DHT22)
DHT dht(DHTPIN, DHTTYPE);
// --- Configurações do Sistema de Irrigação ---
const int UMIDADE_SOLO_MINIMA = 40; // Limiar em % (Ajustar conforme calibração!)
// Valor menor indica solo mais seco.
const unsigned long TEMPO_IRRIGACAO_MS = 5000; // Tempo que a bomba fica ligada (5 segundos)
```

```
const unsigned long INTERVALO_LEITURA_MS = 600000; // Intervalo entre leituras (10 minutos)
const unsigned long INTERVALO_IRRIGACAO_MIN_MS = 3600000; // Intervalo mínimo entre irrigações (1 hora)
// --- Variáveis Globais ---
float temperatura = 0.0;
float umidadeAr = 0.0;
int umidadeSolo[3] = {0, 0, 0}; // Array para armazenar a umidade de cada sensor
unsigned long ultimaLeitura = 0;
unsigned long ultimaIrrigacao = 0;
bool bombaLigada = false;

// --- Função para Selecionar o Canal do Multiplexador ---
void selectMuxChannel(byte channel) {
    // Mapeia o número do canal (0-7) para os pinos de seleção So, S1, S2
    digitalWrite(MUX_So_PIN, bitRead(channel, 0)); // LSB
    digitalWrite(MUX_S1_PIN, bitRead(channel, 1));
    digitalWrite(MUX_S2_PIN, bitRead(channel, 2)); // MSB
    delay(5); // Pequeno delay para estabilização do Mux
}

// --- Função para Ler Umidade do Solo de um Canal Específico ---
int readSoilMoisture(byte channel) {
    selectMuxChannel(channel); // Seleciona o canal desejado no Mux
    int valorRaw = analogRead(MUX_Z_PIN); // Lê o valor analógico da saída do Mux
    // Mapeia o valor lido (0-1023) para uma porcentagem (0-100%)
    // ATENÇÃO: Os valores MIN e MAX dependem do sensor e precisam ser calibrados.
    // Valor MIN = Leitura com sensor no ar (seco)
    // Valor MAX = Leitura com sensor submerso em água
    // Inverta o mapeamento se o seu sensor der valores maiores para solo seco.
    int valorMinCalibrado = 800; // Exemplo: Valor lido no ar
    int valorMaxCalibrado = 350; // Exemplo: Valor lido na água (sensor capacitivo normalmente diminui com umidade)
    int porcentagem = map(valorRaw, valorMinCalibrado, valorMaxCalibrado, 0, 100);
    // Garante que a porcentagem fique entre 0 e 100
    porcentagem = constrain(porcentagem, 0, 100);
    return porcentagem;
}

// --- Função de Configuração Inicial ---
void setup() {
    Serial.begin(115200);
    Serial.println("\nIniciando Sistema de Jardim Suspenso Automatizado...");
    // Inicializa o sensor DHT
    dht.begin();
    Serial.println("Sensor DHT inicializado.");
    // Configura os pinos do Relé e do Multiplexador como SAÍDA
    pinMode(RELAY_PIN, OUTPUT);
    pinMode(MUX_So_PIN, OUTPUT);
    pinMode(MUX_S1_PIN, OUTPUT);
}
```

```
pinMode(MUX_S2_PIN, OUTPUT);
// Garante que a bomba comece desligada
digitalWrite(RELAY_PIN, HIGH); // Módulos relé comuns acionam com LOW
bombaLigada = false;
Serial.println("Pinos configurados. Bomba desligada.");
// Realiza uma leitura inicial
ultimaLeitura = millis() - INTERVALO_LEITURA_MS; // Força a primeira leitura no loop
ultimaIrrigacao = millis() - INTERVALO_IRRIGACAO_MIN_MS; // Permite irrigar logo
se necessário
}
// --- Loop Principal ---
void loop() {
    unsigned long agora = millis();
    // Verifica se é hora de fazer uma nova leitura
    if (agora - ultimaLeitura >= INTERVALO_LEITURA_MS) {
        ultimaLeitura = agora;
        Serial.println("\nRealizando leitura dos sensores...");
        // --- Leitura do Sensor DHT22 ---
        temperatura = dht.readTemperature();
        umidadeAr = dht.readHumidity();
        if (isnan(temperatura) || isnan(umidadeAr)) {
            Serial.println("Erro ao ler do sensor DHT!");
        } else {
            Serial.print("Temperatura: ");
            Serial.print(temperatura);
            Serial.println(" *C");
            Serial.print("Umididade do Ar: ");
            Serial.print(umidadeAr);
            Serial.println(" %");
        }
    }
    // --- Leitura dos Sensores de Umidade do Solo via Mux ---
    int umidadeMedia = 0;
    for (int i = 0; i < 3; i++) {
        umidadeSolo[i] = readSoilMoisture(i); // Lê canal 0, 1 e 2
        Serial.print("Umidade Solo [Fileira ");
        Serial.print(i + 1);
        Serial.print("]: ");
        Serial.print(umidadeSolo[i]);
        Serial.println(" %");
        umidadeMedia += umidadeSolo[i];
    }
    umidadeMedia /= 3; // Calcula a média
    Serial.print("Umidade Média do Solo: ");
    Serial.print(umidadeMedia);
    Serial.println(" %");
    // --- Lógica de Irrigação ---
    // Verifica se a umidade média está abaixo do limiar E
    // se já passou o intervalo mínimo desde a última irrigação
```

```
if (umidadeMedia < UMIDADE_SOLO_MINIMA && (agora - ultimaIrrigacao >=
INTERVALO_IRRIGACAO_MIN_MS)) {
    Serial.println("Umidade baixa detectada. Acionando a bomba...");
    digitalWrite(RELAY_PIN, LOW); // Liga a bomba (LOW para acionar o relé)
    bombaLigada = true;
    ultimaIrrigacao = agora; // registra o início da irrigação
    // Mantém a bomba ligada pelo tempo definido
    delay(TEMPO_IRRIGACAO_MS);
    Serial.println("Tempo de irrigação concluído. Desligando a bomba.");
    digitalWrite(RELAY_PIN, HIGH); // Desliga a bomba
    bombaLigada = false;
} else if (umidadeMedia >= UMIDADE_SOLO_MINIMA) {
    Serial.println("Umidade do solo OK.");
} else {
    Serial.println("Umidade baixa, aguardando intervalo mínimo para irrigar.");
}
} // Fim do bloco de leitura periódica
// Outras tarefas podem ser adicionadas aqui (ex: servidor web)
delay(100); // Pequeno delay para não sobrecarregar o ESP
```

CONCLUSÃO

Quando o programa é carregado no Arduino IDE, o sensor de umidade do solo exibirá o valor de secura do solo. Há a necessidade de definir o nível baixo de umidade do solo, supondo que foi definido o valor em 800 (nível baixo/21,79% de umidade), que é a referência da umidade do ar seco quando a baixa umidade ultrapassa esse valor, então, a bomba iniciará o funcionamento automaticamente e quando o valor de 350 (nível alto/65,78%) for atingido, a bomba será desligada automaticamente.

Se deseja-se inundar o sistema por qualquer motivo fora da avaliação de parâmetros pelos sensores, o sensor de umidade deve permanecer em um nível mais alto ou o valor de programação deve ser menor que o valor fixo.

O sistema está programado para mostrar que o motor começa a irrigar quando o valor de secura é maior que o limiar “baixa umidade/valor alto” que corresponde ao intervalo (800 – 1023) e desliga quando o solo atinge um estado equilibrado que, aqui, nesse sistema simples é o valor 800.

Qualquer valor entre 800 e 350 representa uma umidade média. O sistema também avalia o tempo de bomba ligada (5000ms) desligando-a após transcorrido esse intervalo, o intervalo de leitura que é de 10 minutos (600000ms) corresponde a um limiar, não permitindo que passe esse período sem uma nova leitura e o intervalo mínimo entre irrigações que é de 1 hora (3600000ms)

onde caso o sistema não seja acionado dentro desse intervalo, levando-se em consideração as condições tropicais, o sistema impõe mais um ciclo de rega de 5 segundos.

Adotar o sistema automático não só economiza água, mas também tempo, a verdade é que não existe um consenso entre valores adequados de irrigação, porém a condição média nos trópicos é que mais água leva a uma maior eficiência no crescimento de plantas de jardim. Com o devido cuidado em relação às espécies de plantas utilizadas, sempre haverá um nível de rega eficaz onde as plantas obterão uma quantidade próxima do ideal de água para o crescimento máximo, isso é difícil de mensurar e pode ser uma boa proposta para outros desdobramentos e trabalhos de domótica para irrigação residencial. Um sistema de irrigação automatizado pode ajudar as pessoas a cultivarem plantas em espaços mais reduzidos com diferentes finalidades e com um aproveitamento superior.

Um sistema de irrigação mais eficaz e inteligente pode auxiliar na economia de tempo, de recurso água e diminuindo a perda de água que não é utilizada pelos vegetais na rega manual, sendo assim, pode-se melhorar a eficiência de um sistema de irrigação de jardins fazendo uso correto da automação mais simples e à baixo custo.

REFERÊNCIAS

2741

Barros, Marcos José; da Silva, Cristiane Ribeiro (organizadores). *Formação Inicial e continuada em Jardinagem*. (2012). IFPR. Disponível em: <https://ifpr.edu.br/pronatec/wp-content/uploads/sites/46/2012/07/jardi.pdf>. Acesso em: 15 jul. 2025.

Basri, M. H., et al. (2024). *IoT based Smart Irrigation System using Soil Moisture Sensor and ESP8266 NodeMCU*. International Journal of Novel Research and Development (IJNRD), 9(2), 196-201. Disponível em: <https://www.ijnrd.org/papers/IJNRD2402026.pdf>. Acesso em: 15 jul. 2025.

Batrinu, Catalin. *Projetos de Automação Residencial com ESP8266* / Catalin Batrinu. – São Paulo: Novatec, 2018. 172 p.

Brasil. Congresso Nacional. Senado Federal. *Guia de Paisagismo Sustentável*. Senado Federal. [Brasília]: Senado Federal, 2024. 30 p.

Developing a Smart Irrigation System Using NodeMCU. Gawali, Palomi; Rode, Omkar; Saundane, Ashwin; Shaikh, Sahil; Shaikh, Arman. *International Journal of Internet of Things and Web Services*, 2024. Disponível em: [https://www.iaras.org/iaras/filedownloads/ijitws/2024/022-0002\(2024\).pdf](https://www.iaras.org/iaras/filedownloads/ijitws/2024/022-0002(2024).pdf). Acesso em: 14 jul. 2025.

Eletrogate. *Sensor capacitivo de umidade do solo V2.0 com regulador*. Disponível em: <https://www.eletrogate.com/sensor-capacitivo-de-umidade-do-solo-v20-com-regulador>. Acesso em: 8 jul. 2025.

Eletrogate. Disponível em: <https://www.eletrogate.com/>. Acesso em 21 jul. 2025.

EOS Data Analytics. Disponível em: <https://eos.com/>. Acesso em: 11 jul. 2025.

How to electronics. *IoT Smart Agriculture Automatic Irrigation System with ESP8266*. Disponível em: <https://how2electronics.com/iot-smart-agriculture-automatic-irrigation-system-with-esp8266/>. Acesso em: 14 jul. 2025.

Ideali, Wagner. *Conectividade em Automação e IoT: Protocolos I2C, SPI, USB, TCP-IP entre outros. Funcionalidade e interligação para automação e IoT / Wagner Ideali*. – Rio de Janeiro : Alta Books, 2021. 256 p.

Muthekar, R., et al. (2024). *Smart Irrigation System using ESP8266, NodeMCU & ThingSpeak*. International Journal of Innovative Technology and Web Sciences (IJITWS), 22(1), 8-12. Disponível em: [https://www.iaras.org/iaras/filedownloads/ijitws/2024/022-0002\(2024\).pdf](https://www.iaras.org/iaras/filedownloads/ijitws/2024/022-0002(2024).pdf). Acesso em: 15 jul. 2025.

Nutrição de Safras. Disponível em: <https://nutricaodesafras.com.br/>. Acesso em: 15 jul. 2025.

Oliveira, Sérgio de. *Internet das coisas com ESP8266, Arduino e RaspBerry PI / Sérgio de Oliveira*. – São Paulo: Novatec, 2021. 311 p.

Pawar, Y. S., et al. (2024). *Smart irrigation optimization through the integration of machine learning and IoT for sustainable agriculture*. Multimedia Tools and Applications, 1-24. DOI: 10.1007/s11042-024-19504-y. Disponível em: <https://link.springer.com/article/10.1007/s11042-024-19504-y>. Acesso em: 15 jul. 2025.

2742

Portal Embrapa. Disponível em: <https://www.embrapa.br/>. Acesso em: 15 jul. 2025.

Senado Federal. 2024. Disponível em: <https://www12.senado.leg.br/noticias/materias/2024/09/11/senado-tem-pec-sobre-qualidade-do-ar>. Acesso em: 14 jul. 2025.

Smart Project. 2025. Disponível em: <https://www.smartprojectsbrasil.com.br/>. Acesso em: 14 jul. 2025.

Smart Irrigation System using ESP8266, NodeMCU & ThingSpeak. Muthekar, Divya S.; Gadekar, Kishor V.; Deulkar, A. M.; Pardeshi, D. B. *8th International Conference on Inventive Systems and Control (ICISC)*, 2024. Disponível em: <https://ieeexplore.ieee.org/document/10677594/>. Acesso em: 21 jul. 2025.

Stevan Junior, Sergio Luiz. *Automação e instrumentação industrial com arduino: teoria e projetos / Sergio Luiz Stevan Junior, Rodrigo Adamshuk Silva*. – São Paulo : Érica, 2015. 296 p.

Stevan Junior, Sergio Luiz. *Domótica: automação residencial e casas inteligentes com Arduino e ESP8266 / Sergio Luiz Stevan Junior, Felipe Adalberto Farinelli*. – São Paulo: Érica, 2019. 206 p.

Svitla. (2020). *ESP8266 vs ESP8285 vs ESP32*. Disponível em: <https://svitla.com/blog/esp8266-vs-esp8285-vs-esp32/#ESP8266>. Acesso em: 14 jul. 2025.

Tolentino Jr., João B. Hidráulica Agrícola / João B. Tolentino Jr. – Maringá: 2023. Disponível em: <https://hidraulica.tolentino.pro.br/bombas-hidr%C3%A1ulicas.html>. Acesso em: 21 jul. 2025.

VanDerZanden, A. M. (2008). *Environmental factors affecting plant growth*. Oregon State University Press. Disponível em: <https://extension.oregonstate.edu/gardening/techniques/environmental-factors-affecting-plant-growth>. Acesso em: 15 jul. 2025.