


ARDUINO GARDEN: UM SISTEMA INTELIGENTE PARA MONITORAMENTO DE PLANTAS

 <https://doi.org/10.56238/arev7n5-187>

Data de submissão: 12/04/2025

Data de publicação: 12/05/2025

Lucas Santos Garreto

Doutorando em Engenharia Civil – UTFPR

lucasgarreto@alunos.utfpr.edu.br

<https://orcid.org/0009-0007-9330-0625>

Marcio Vinicius Carneiro Menezes

Graduando em Engenharia Civil – Estácio

marcio081010@outlook.com

Hiana Cristina Silva de Carvalho

Arquiteta e Urbanista – CEUMA

hianacarvalho@hotmail.com

Carlos Gustavo Martins Silva

Engenheiro Civil - UEMA

cgustavo.cgms@gmail.com

Icaro Gustavo Serra Santos

Arquiteto e Urbanista – UEMA

icaro.serra@hotmail.com

Ana Carolina Vale Porto Pereira

Mestranda em Gestão e Regulação de Recursos Hídricos - ANA

engcarolinavaleporto@gmail.com

Gleildson Oliveira dos Santos

Engenheiro Civil – UNINASSAU

gleildson1996@gmail.com

Amanda Caroline da Silva Buna

Engenheira Civil – UFMA

amandabuna4@gmail.com

RESUMO

A identificação rápida e precisa de poluentes e germes nas mãos emergiu como uma prioridade crítica no ambiente atual de crescentes preocupações com a higiene e o desenvolvimento de doenças infecciosas. Este artigo acadêmico investiga a aplicação criativa da plataforma Arduino como um dispositivo sensor transportável e de baixo custo. Devido ao arduino ser uma ferramenta acessível e dinâmico, ele permite a criação de sensores especializados unificados aos componentes de comunicação como visor LCD e display OLED para uma comunicação em tempo real de uma planta ou um jardim. A inclusão de condicionais no software do Arduino facilitam de forma precisa e eficácia do processo de umidade do solo e temperatura ambiente, ao mesmo tempo que fornece uma resposta

flexível e adaptativa com diferentes emoções emitidas em um display OLED. As conclusões deste estudo têm o potencial de alterar a forma como a jardins são cuidados, melhorando a qualidade nas colheitas e o desenvolvimento amigável entre máquina e homem.

Palavras-chave: Tecnologia. Sustentabilidade. Arduino.

1 INTRODUÇÃO

A combinação de sistemas inteligentes e agricultura abriu novos caminhos para formas criativas de melhorar a nossa relação com o mundo natural nesta era de rápido desenvolvimento tecnológico. A ideia de um “Jardim Inteligente” surgiu da incorporação da tecnologia Arduino nas técnicas convencionais de jardinagem.¹ Este método inovador utiliza microcontroladores para construir sistemas inteligentes e automatizados que maximizam o crescimento das plantas, o uso eficiente de recursos e o gerenciamento geral do jardim.

Durante esse artigo iremos mergulhar no mundo dos Jardins Inteligentes e descobrir as muitas possibilidades que existem quando a horticultura e a tecnologia colidem, como ponto de ignição o microcontrolador Arduino.² A plataforma eletrônica Arduino de código aberto é um excelente ponto de partida para a criação de soluções de jardinagem inteligentes porque é flexível e fácil de usar tanto para jardineiros novatos quanto para jardineiros experientes.

Fundamentalmente, um Smart Garden com Arduino consiste em uma rede de sensores, atuadores e microcontroladores que colaboram para monitorar e regular continuamente uma série de fatores ambientais. Essas características, que são elementos vitais que afetam a saúde e o crescimento das plantas, incluem o teor de umidade do solo, a temperatura ambiente, a umidade e a intensidade da luz.³ Os Smart Gardens controlados por Arduino usam uma variedade de sensores para coletar dados em tempo real, dando aos jardineiros a capacidade de tomar decisões dependendo dos requisitos específicos de suas plantas.

O ambiente de programação intuitivo da tecnologia Arduino é um dos seus principais benefícios. Com a ajuda da interface gráfica do Arduino, jardineiros novatos com pouco ou nenhum conhecimento de codificação podem criar programas exclusivos que controlam as ações de seu Smart Garden. Pessoas de diversas origens podem agora contribuir ativamente para o desenvolvimento de sistemas de jardinagem inteligentes, sustentáveis e eficazes devido à democratização da tecnologia.

2 MICROCONTROLADOR ARDUINO

A plataforma Arduino é um ponto brilhante para acessibilidade e criatividade na área de eletrônica e programação. Embora tenha sido criado inicialmente como uma placa de prototipagem de código aberto, o Arduino cresceu para representar uma tecnologia de microcontrolador fácil de usar, tornando-se um recurso valioso para desenvolvedores iniciantes e experientes. Seu ambiente de desenvolvimento integrado (IDE) e sua grande comunidade, que trocam uma infinidade de materiais, tutoriais e projetos, são as razões de sua simplicidade. Com o Arduino, os alunos podem agora

explorar o fascinante mundo da eletrônica e da programação e usar uma plataforma prática para concretizar suas ideias.⁴

Imagem 1 - Microcontrolador Arduino



Fonte: EMBARCADOS, 2013.⁵

O Arduino é o único que pode ser usado para uma ampla variedade de projetos, desde simples displays de LED até robótica complexa e aplicações de Internet das Coisas.⁶ O microcontrolador de uma placa Arduino, que é o cérebro por trás da execução das instruções programadas, é seu componente central. Normalmente, este chip é um modelo da série ATmega. A facilidade com que os alunos podem usar uma placa Arduino para fazer interface com sensores, atuadores e outros componentes eletrônicos lhes dá a liberdade de investigar e testar usos práticos.⁷ Com um ambiente de codificação fácil de usar e funcionalidade plug-and-play, a plataforma reduz a barreira de entrada para os alunos, liberando-os para se concentrarem nas partes criativas de seus projetos, em vez de resolverem os problemas técnicos. Além disso, a influência do Arduino vai além da sala de aula; ele é usado em instalações de arte interativas, desenvolvimento de startups e entusiastas de eletrônicos DIY. Por ser de código aberto, os usuários podem colaborar e inovar enquanto adicionam um código e uma biblioteca consideráveis que aprimoram o ambiente Arduino. Uma ferramenta essencial que incorpora o espírito de aprendizagem prática e democratiza o acesso ao fascinante mundo da eletrônica e da programação, o Arduino é usado tanto por programadores novatos quanto por desenvolvedores experientes que estão criando a próxima grande invenção.

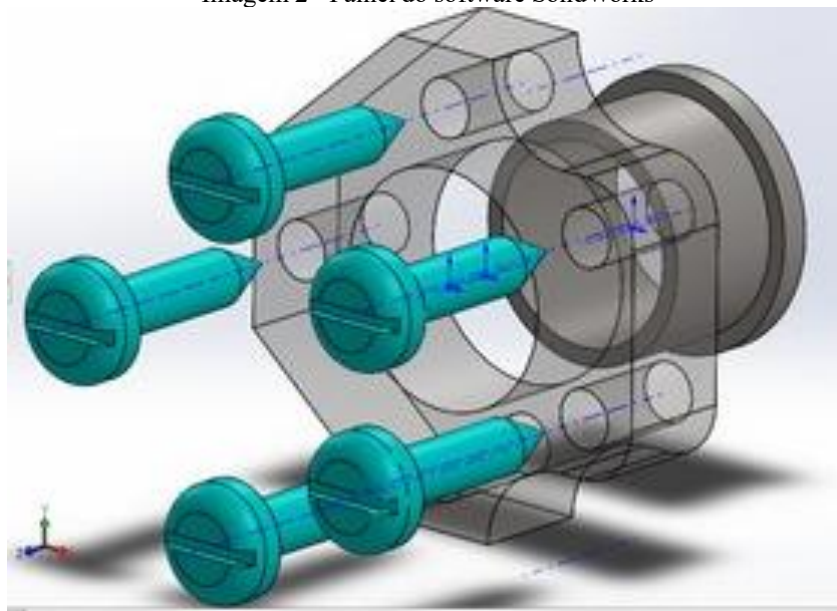
A linguagem de programação da plataforma Arduino é um poderoso facilitador tanto para desenvolvedores inexperientes quanto para alunos experientes. A linguagem de programação Arduino, que é construída em uma forma condensada de C e C++, equilibra cuidadosamente usabilidade e

acessibilidade.⁸ Essa linguagem simplificada é um excelente ponto de partida para o aprendizado de programação, pois permite que os alunos entendam rapidamente as ideias de codificação sem serem sobrecarregados pelas complexidades da gramática. Os alunos podem facilmente criar códigos para controlar o comportamento do microcontrolador, criando interações com sensores e coordenando o funcionamento de diversos componentes eletrônicos, graças ao Ambiente de Desenvolvimento Integrado (IDE) Arduino.⁹ Devido à facilidade de usar a linguagem e à abundância de bibliotecas e recursos disponíveis para a comunidade, os alunos podem transformar suas ideias em produtos concretos.

3 MODELAGEM 3D

Nas áreas de tecnologia e engenharia, a modelagem 3D é um componente fundamental para a criatividade e a resolução de problemas. Antes mesmo de os protótipos reais serem feitos, engenheiros e designers podem visualizar, avaliar e otimizar seus conceitos, empregando software especializado para criar representações tridimensionais de itens ou sistemas.¹⁰ A capacidade de criar protótipos virtualmente reduz muito o tempo necessário para construir um produto, permitindo iterações e refinamentos rápidos de design. Estudantes e especialistas agora podem imitar situações do mundo real, avaliar a integridade estrutural e prever resultados de desempenho com níveis de precisão e complexidade anteriormente impensáveis, graças à modelagem 3D. As equipes interdisciplinares podem colaborar de forma mais dinâmica e eficaz em ideias difíceis devido à natureza visual e interativa dos modelos 3D, que podem ajudar os membros da equipe a obter uma compreensão mais profunda dos tópicos técnicos.

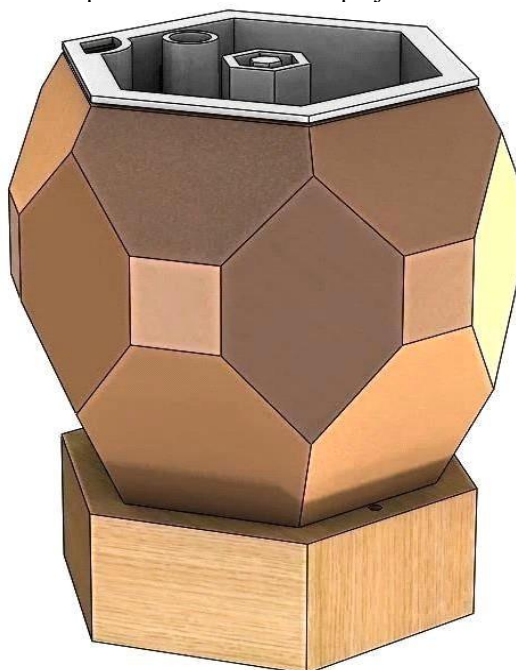
Imagem 2 - Painel do software SolidWorks



Fonte: PROGRAMAÇÃO PRÁTICA, 2025.¹¹

Além disso, a modelação 3D tem impacto numa vasta gama de indústrias, desde a indústria transformadora e aeroespacial até aos cuidados de saúde e à arquitetura, e não apenas na fase de conceptualização. Por exemplo, na indústria transformadora, os sistemas de produção assistida por computador (CAM) baseiam-se em modelos 3D para orientar máquinas automatizadas na produção de componentes muito precisos e complexos. Com o uso da modelagem 3D, os arquitetos podem produzir representações realistas de edifícios, o que auxilia na visualização do projeto e na comunicação com o cliente, além de agilizar o processo de projeto. A flexibilidade da modelagem 3D revolucionou a forma como lidamos com problemas de engenharia, incentivando a criatividade e permitindo-nos testar ideias num ambiente virtual antes de implementá-las no mundo real.¹²

Imagem 3 - Impressão 3D utilizada no projeto de Smart Garden



Fonte: THINGIVERSE, 2019.¹³

3.1 IRRIGAÇÃO SUSTENTAVEL

Incentivar técnicas de irrigação sustentáveis é essencial para exemplificar o uso racional da água, especialmente na agricultura, onde a água é um recurso vital. Um método criativo é reciclar garrafas plásticas para fazer um sistema de irrigação barato e ecologicamente correto. Os alunos podem criar um sistema de irrigação alimentado por gravidade que utiliza menos água usando garrafas plásticas como reservatórios. As caixas garrafas como recipientes eficazes de armazenamento de água, e a gravidade garante um fornecimento contínuo de água às plantas abaixo, posicionando-as em altitudes ideais.¹⁴ Esta técnica de simples tecnologia utiliza energia potencial gravitacional, uma propriedade da matéria para fornecer um sistema de irrigação sustentável com pouco uso de energia. Além disso, ao reaproveitar resíduos, a utilização de garrafas de plástico neste contexto promove uma economia circular e diminui o impacto ambiental dos sistemas de irrigação convencionais.

O método de gotejamento para a irrigação sustentável se baseia na Lei de Torricelli que descreve o fluxo de um líquido contido em um recipiente, através de um pequeno orifício, sob a ação da gravidade.¹⁵ A partir do teorema de Torricelli pode-se calcular o caudal de saída de um líquido por um orifício:

$$v = \sqrt{2gh}$$

onde v é a velocidade do fluido, g é a aceleração da gravidade com uma constante de $9,8 \text{ m/s}^2$ e h representa a altura do líquido acima do orifício.

O gotejamento ocorre à medida que a água atravessa o parafuso, seguindo a trajetória determinada pela gravidade. A força gravitacional atua sobre a água, permitindo que ela goteje consistentemente na planta abaixo. A altura do reservatório e a configuração do parafuso são fatores críticos que influenciam a pressão e, consequentemente, a taxa de gotejamento.

4 SENSORES UTILIZADOS

Quando se trata de sistemas de jardinagem inteligentes, adicionar sensores de umidade e temperatura do solo é essencial para melhorar a precisão e eficácia do monitoramento das plantas. Integrados no sistema Arduino Garden, os sensores de umidade do solo fornecem aos jardineiros informações em tempo real sobre o conteúdo de água do solo, permitindo-lhes ajustar os horários de irrigação e garantir que as plantas recebam a quantidade certa de água. Esta prática não só incentiva a conservação da água, mas também evita situações em que as plantas podem ser afetadas negativamente pelo excesso de água ou submersão. Ao permitir que os utilizadores façam julgamentos bem informados, a avaliação precisa do nível de humidade do solo promove um estilo de agricultura mais consciente dos recursos e da sustentabilidade.¹⁶

Ao mesmo tempo, a incorporação de sensores de temperatura do solo eleva as capacidades de monitoramento do Arduino Garden a um novo nível. A temperatura do solo tem um impacto significativo na atividade microbiana da zona radicular e no crescimento das plantas. Os jardineiros podem maximizar o metabolismo das plantas e a absorção de nutrientes usando formas de obter informações sobre as condições térmicas do solo através da integração de sensores de temperatura no sistema. Esta informação é especialmente útil para compreender as diferenças sazonais, escolher tipos de plantas adequados e diminuir os efeitos de variações bruscas de temperatura. Em essência, o Arduino Garden torna-se um sistema inteligente e adaptativo quando sensores de umidade e temperatura do solo são implantados em conjunto, garantindo que as plantas recebam as condições climáticas exatas necessárias para um crescimento robusto e saudável. Estes sensores revelam-se recursos inestimáveis à medida que aprendemos mais sobre as nuances da monitorização das plantas, ajudando a eliminar a divisão entre os métodos convencionais de jardinagem e a tecnologia de ponta.

A incorporação de sensores como sensores de umidade do solo e sensor de temperatura TMP36 é essencial para a operação e realização do projeto na área de sistemas de monitoramento de plantas baseados em Arduino.¹⁷ Parte essencial do sistema Arduino Garden, os sensores de umidade do solo são responsáveis por fornecer aos usuários informações em tempo real sobre o teor de umidade do

solo. Esses sensores medem a capacitância ou condutividade elétrica do solo e convertem os resultados em estimativas precisas do teor de umidade do solo. monitoram o nível de hidratação do solo incorporando sensores de umidade do solo, o que lhes confere uma gestão precisa da irrigação e garante que as plantas recebam a quantidade certa de água.

Imagem 5 - A imagem mostra o sensor de umidade do solo e temperatura



**SENSOR DE
UMIDADE DO
SOLO**

**SENSOR DE
TEMPERATURA
TMP36**

Fonte: os autores, 2023

A adição do sensor de temperatura TMP36 aos sensores de umidade do solo aprimora ainda mais as capacidades de monitoramento do Arduino Garden. Um sensor analógico barato e preciso com baixo consumo de energia, o TMP36 mede a temperatura com precisão. Os sensores TMP36 podem ser colocados estrategicamente no solo para fornecer aos jardineiros dados exatos de temperatura, essenciais para compreender as condições térmicas na zona radicular.

Fazer julgamentos fundamentados sobre os tipos de plantas, padrões de desenvolvimento e como reagir às flutuações de temperatura que podem afetar a saúde das plantas fica muito mais fácil com a ajuda dessas informações.¹⁸ Ao combinar sensores de umidade do solo com o sensor de temperatura TMP36, o Arduino Garden se torna um sistema inteligente e flexível que permite aos usuários projetar uma atmosfera que promova o melhor crescimento e saúde possível para suas plantas.

5 IMPLEMENTAÇÃO PRÁTICA

O passo inicial é escolher cuidadosamente e anexar sensores à placa Arduino, como sensores de temperatura TMP36 e sensores de umidade do solo. Como a coleta correta de dados depende de fiação adequada, é fundamental compreender as características de cada sensor.¹⁹ Os participantes no projeto ganham experiência prática com integração de hardware através deste método tátil, que os

ajuda a compreender como os sensores se comunicam com a plataforma Arduino em um nível mais profundo.

Assim que os componentes de hardware estiverem instalados, a próxima fase envolve a programação do microcontrolador Arduino para coletar, processar e transmitir dados dos sensores. Essa é uma ótima forma de mergulhar no mundo da programação Arduino, criando algoritmos que governam o comportamento do sistema. Esta etapa inclui a definição de limites para a umidade do solo, a implementação de lógica baseada na temperatura e o desenvolvimento de rotinas para atuadores. O aspecto da programação não apenas aprimora as habilidades de codificação, mas também incute uma mentalidade de resolução de problemas à medida que eles ajustam as respostas do sistema às diversas condições ambientais. O processo iterativo de teste e refinamento do código fornece informações valiosas sobre as complexidades do desenvolvimento de sistemas e reforça a natureza interconectada de hardware e software em aplicações do mundo real.

Implantar o sistema em um ambiente de jardinagem real é a última etapa da implementação prática à medida que o Arduino Garden toma forma. À medida que o Arduino Garden monitora e responde ativamente aos requisitos das plantas, os alunos poderão ver os resultados do seu trabalho. Este programa abrangente oferece uma fusão especial de proficiência técnica e conhecimento útil em horticultura. Os participantes aprenderão sobre as ramificações mais amplas dos sistemas inteligentes na agricultura, além dos componentes técnicos, estabelecendo as bases para uma época em que a tecnologia e o mundo natural possam coexistir pacificamente. À medida que os participantes veem em primeira mão como o seu sistema inteligente de monitorização de plantas afeta diretamente a saúde do jardim, este método de implementação prática não só fortalece as competências técnicas, mas também desperta o desejo de inovação e soluções sustentáveis.

5.1 APLICAÇÃO VIAVEL NA COLONIZAÇÃO DE MARTE

O uso do sistema Arduino Garden torna-se evidente como uma forma altamente relevante e criativa de manter a vida em Marte no contexto da colonização. O Arduino Garden torna-se um farol de autossuficiência, abordando questões vitais de produção de alimentos e monitoramento ambiental, fundindo os princípios da eletrônica e da sustentabilidade. Usando sensores de umidade do solo, displays dinâmicos e microcontroladores Arduino, o sistema inteligente de monitoramento de plantas é um exemplo de solução escalonável de cultivo para o rigoroso clima marciano. Com a automação baseada em Arduino, os recursos são utilizados de forma mais eficaz e a água é utilizada com sabedoria, o que apoia iniciativas de conservação de água que são essenciais para manter as operações agrícolas num ambiente marciano com recursos limitados.²⁰

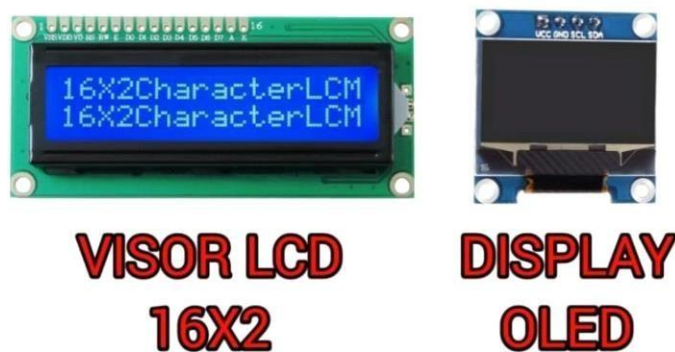
Além disso, o sistema Arduino Garden estende o seu impacto para além do mero monitoramento das plantas, tornando-se um componente vital no bem-estar psicológico e na adaptação humana ao habitat marciano. A capacidade do sistema de comunicar dados em tempo real através de exibições visuais não só fornece informações essenciais sobre a saúde das plantas, mas também promove uma ligação entre os colonizadores e o seu ambiente cultivado. No isolamento de Marte, onde o bem-estar mental é uma consideração crítica, o Jardim Arduino serve como uma ligação tangível às práticas terrenas, oferecendo uma sensação de continuidade e propósito através do cultivo de plantas. Esta abordagem abrangente à sustentabilidade reconhece os aspectos psicológicos da colonização espacial e mostra como a electrónica pode ajudar a criar uma atmosfera pacífica e de apoio para os futuros residentes de Marte.

6 COMUNICAÇÃO DINÂMICA

As exibições visuais desempenham um papel crítico na melhoria da experiência do usuário do sistema Arduino Garden por meio da comunicação dinâmica. Uma das principais ferramentas para fornecer aos usuários informações em tempo real sobre a umidade do solo é a integração de um display LCD 16x2.²¹ Os alunos podem projetar o display LCD para exibir mensagens fáceis de usar, como "Seco", "Pouco úmido", "Úmido" ou "Muito úmido", elaborando cuidadosamente o código, dando aos jardineiros conhecimento instantâneo sobre as condições do solo. Este mecanismo de comunicação dinâmico melhora a usabilidade do sistema, permitindo aos utilizadores avaliar rapidamente as condições do solo e reagir às exigências das suas plantas. O display LCD 16x2 não apenas adiciona uma dimensão prática ao Arduino Garden, mas também exemplifica como a tecnologia pode preencher a lacuna entre dados complexos e interfaces fáceis de usar em aplicações do mundo real.

Ao adicionar um display OLED, o sistema Arduino Garden ganha um componente visualmente atraente que avança a comunicação dinâmica. Os alunos podem configurar o display OLED para mostrar emoticons que correspondem a diferentes níveis de umidade e transmitir a reação da planta ao ambiente. Por exemplo, um rosto feliz poderia representar o teor de umidade ideal, mas um rosto triste sugeriria solo seco. Esse uso inovador do display OLED melhora a capacidade de comunicação do sistema e adiciona um toque de interatividade, o que torna o processo de monitoramento mais interessante e prazeroso para os usuários.²² Através do uso de um método de exibição criativo e fácil de usar, a incorporação de emoticons visuais não apenas ajuda na transmissão de informações, mas também aumenta o valor educacional do Arduino Garden, promovendo uma compreensão mais profunda do comportamento e dos cuidados com as plantas.

Imagem 7 - A imagem mostra dois componentes de comunicação



Fonte: os autores, 2023.

Dessa forma, o uso duplo de um display OLED e um display LCD 16x2 no sistema Arduino Garden ilustra uma estratégia abrangente para comunicação dinâmica. Embora o LCD 16x2 forneça informações precisas e fáceis de entender sobre os níveis de umidade do solo, o display OLED traz um toque artístico ao mostrar as “emoções” das plantas em relação aos níveis de umidade. Esta fusão de criatividade e pragmatismo não apenas melhora o funcionamento do sistema inteligente de monitoramento de plantas, mas também demonstra a amplitude e profundidade das opções de comunicação que podem ser alcançadas integrando cuidadosamente as tecnologias de exibição nos projetos Arduino.

6.1 IDENTIFICAÇÃO SONORA

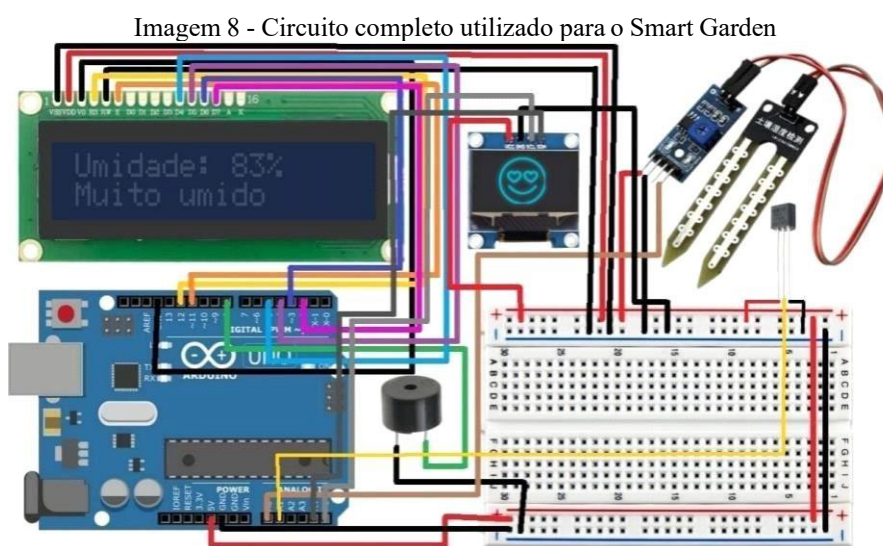
Para a identificação sonora foi incluso um dispositivo Buzzer, Um componente eletrônico básico, mas adaptável é essencial para sistemas de feedback e sinalização auditiva. O Buzzer, que é essencialmente um transdutor eletroacústico, emite um tom ou bipe característico quando os sinais elétricos são convertidos em ondas sonoras.²³ O Buzzer é um ótimo instrumento para ensinar aos alunos sobre processamento de sinais, modulação de frequência e criação de som em ambientes educacionais e projetos práticos de eletrônica. É um elemento de fácil compreensão para iniciantes que estão aprendendo a conectar componentes eletrônicos devido à sua simples integração com microcontroladores, como os encontrados na plataforma Arduino. A campanha é um excelente exemplo de como a engenharia elétrica e a percepção sensorial se encontram quando se trata de sistemas de alarme, projetos musicais ou protótipos interativos.

Durante o desenvolvimento do circuito foi codado uma condicional para o Buzzer aonde se a umidade de solo do jardim estiver abaixo de 16% será emitido dois bips curtos a cada dez segundos, com essa solução a atenção aos baixos níveis de umidade se tornam mais fáceis de identificar.

7 RESULTADOS

O uso duplo de um display OLED e um display LCD 16x2 no sistema Arduino Garden ilustra uma estratégia abrangente para comunicação dinâmica. Embora o LCD 16x2 forneça informações precisas e fáceis de entender sobre os níveis de umidade do solo, o display OLED traz um toque artístico ao mostrar as “emoções” das plantas em relação aos níveis de umidade. Esta fusão de criatividade e pragmatismo não só melhora o funcionamento do sistema inteligente de monitoramento da planta, mas também demonstra a amplitude e profundidade das opções de comunicação que podem ser alcançadas integrando cuidadosamente tecnologias de exibição em projetos Arduino.

Os resultados no mundo real destacam a competência tecnológica do sistema na recolha e análise de dados ambientais, bem como a sua influência palpável no bem-estar edesenvolvimento das plantas sob observação. Ao usar telas LCD e OLED para representar visualmente as condições do solo, o Arduino Garden não apenas facilita a interação do usuário, mas também ajuda os usuários a compreender melhor a complexa relação entre a umidade do solo e a saúde das plantas.



Fonte: os autores, 2023.

Além disso, os participantes ao observarem o processo do projeto desenvolvem suas habilidades de programação e integração de sensores, bem como uma compreensão abrangente de como a tecnologia pode apoiar atividades convencionais, como jardinagem, durante o processo de aplicação atual. Os resultados vão além da funcionalidade básica do sistema, eles representam a fusão de compreensão científica, conhecimento técnico e experiência prática, criando um ambiente de aprendizagem rico e envolvente.

Em última análise, o Arduino Garden permanece como um farol, mostrando não apenas a viabilidade técnica de um sistema inteligente de monitoramento de plantas, mas também destacando a

importância da educação prática e da aplicação criativa da tecnologia na promoção de práticas agrícolas sustentáveis. Os resultados alcançados englobam não apenas uma conquista tecnológica, mas também uma ponte entre a inovação e a natureza, cultivando uma ligação mais profunda entre os seres humanos e o mundo verde que eles cultivam.

8 CONCLUSÃO

Em conclusão, o projeto Arduino Garden é um exemplo incrível de combinação de consciência ecológica com inovação técnica. A integração da tecnologia Arduino com operações convencionais de jardinagem tem o potencial de revolucionar as práticas de jardinagem, conforme demonstrado pelo processo desde a concepção até a execução prática. Com o uso de sensores de umidade e temperatura do solo e comunicação de exibição dinâmica, o sistema permite que os jardineiros desenvolvam uma consciência mais sofisticada do ambiente ao redor de suas plantas.

Os resultados práticos não apenas ressaltam as proezas técnicas do Arduino Garden, mas também ressaltam as implicações educacionais mais amplas. Os alunos envolvidos no processo de implementação não apenas aprimoraram suas habilidades de programação e integração de hardware, mas também cultivaram uma perspectiva holística sobre a relação simbiótica entre tecnologia e natureza. Os displays dinâmicos, que transmitem informações sobre as condições do solo de forma intuitiva, aumentam o envolvimento do utilizador e contribuem para uma ligação mais profunda entre os indivíduos e os seus espaços verdes cultivados.

No futuro, o Arduino Garden será lembrado como um excelente exemplo de sistemas inteligentes de monitoramento de plantas, demonstrando como a tecnologia pode melhorar a sustentabilidade, incentivar a economia de água e aumentar a nossa compreensão dos complexos requisitos das plantas. O Arduino Garden é um monumento ao poder transformador das iniciativas educacionais que criam uma ligação entre os mundos virtual e físico à medida que atravessamos a encruzilhada da programação, engenharia e jardinagem. O Arduino Garden convida-nos para um futuro onde a tecnologia contribui harmoniosamente para a nossa compreensão e cultivo do mundo que nos rodeia, cultivando não apenas plantas, mas também um sentido de responsabilidade ambiental.

RÊFERÊNCIAS

- MENG, H. et al. The development and practice of urban smart garden. [S.l.]: [s.n.], [s.d.].
- BARRETT, Scott F. Arduino III. [S.l.]: Springer Nature, 2022.
- SOUZA, Vitor Amadeu. Introdução aos circuitos elétricos. [S.l.]: Clube de Autores, 2013.
- O'NEILL, Terence; WILLIAMS, Josh. Arduino. Ann Arbor: Cherry Lake Publishing, 2014.
- EMBARCADOS. Arduino Uno. Embarcados, [S.l.], [s.d.]. Disponível em: <https://embarcados.com.br/arduino-uno/>. Acesso em: 25 mar. 2025.
- MAGRANI, Eduardo. A internet das coisas. Rio de Janeiro: FGV Editora, 2018.
- EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. Arduino em ação. [S.l.]: Novatec Editora, 2013.
- PURDUM, Jack. Beginning C for Arduino. [S.l.]: Apress, 2013.
- ODOM, Chris D. Physical computing and robotics with the Arduino IDE - volume one. [S.l.]: [s.n.], [s.d.].
- SCATOLIN, Robson. SolidWorks 2016. [S.l.]: SESI SENAI Editora, 2017.
- PROGRAMAÇÃO PRÁTICA. SolidWorks essencial: o passo a passo da modelagem 3D. [S.l.], [s.d.]. Disponível em: <https://programacaopratica.com.br/produto/solidworks-essencial-o-passo-a-passo-da-modelagem-3d/>. Acesso em: 25 mar. 2025.
- OLIVEIRA, Adriano de. Autodesk® AutoCAD 2016: modelagem 3D. [S.l.]: [s.n.], [s.d.].
- THINGIVERSE. Automatic smart plant pot - (DIY, 3D printed, Arduino, self watering, project). [S.l.], [s.d.]. Disponível em: <https://www.thingiverse.com/thing:3537287>. Acesso em: 25 mar. 2025.
- COMO fazer irrigação por gotejamento usando um parafuso: simples e fácil de fazer. YouTube, 2023. Disponível em: <https://ro.pinterest.com/pin/504403227030896032/>. Acesso em: 20 dez. 2023.
- BISTAFA, Sergio Roberto. A lei de Torricelli $v=\sqrt{2gh}$: uma tradução comentada de sua origem no De Motu Aquarum (Do Movimento das Águas). Revista Brasileira de História da Ciência, v. 7, n. 1, p. 110-119, 2014.
- RAIMUNDO, C. et al. Desenvolvimento de um dispositivo eletrônico para calibração de sensores de umidade do solo. Engenharia Agrícola, v. 27, n. 1, p. 294-303, abr. 2007.
- SOUZA, Vitor Amadeu. Lendo um sensor de temperatura modelo TMP036 programado no Arduino. [S.l.]: Clube de Autores, 2019.
- MARÇAL, Danilo et al. Agricultura digital. [S.l.]: Oficina de Textos, 2022.

MORALES-HERRERA, Rafael et al. Integration of sensors in control and automation systems. Journal of Sensors, v. 2017, p. 1-3, 2017. Disponível em: <https://www.hindawi.com/journals/js/2017/6415876/>. Acesso em: 12 maio 2025.

ASTROBIOLOGIA e as missões tripuladas para a colonização de Marte. RECIMA21 - Revista Científica Multidisciplinar, v. [volume não especificado], p. [intervalo de páginas não especificado], 29 abr. 2021. Disponível em: <https://recima21.com.br>. Acesso em: 12 maio 2025.

LUFSEY, Tom. Arduino LCD connection. [S.l.]: [s.n.], [s.d.].

SOUZA, Vitor Amadeu. Apresentando no display OLED com controlador SSD1306 para medição de 4 a 20 mA programado no Arduino. [S.l.]: Clube de Autores, 2020.

AVCU, Mehmet. Buzzer alarm system with help of Arduino. [S.l.]: Arduino Instructor, 2021.

ANEXO

I. CODIGO UTILIZADO NO MICROCONTROLADOR ARDUINO:

```
#include <Wire.h> #include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h> #include <LiquidCrystal.h>

#define OLED_RESET 4 Adafruit_SSD1306 display(OLED_RESET);

// Pinos do LCD const int rs = 12; const int en = 11; const int d4 = 5; const int d5 = 4; const int
d6 = 3; const int d7 = 2;

// Pino do piezo
const int piezoPin = 8; // Escolha o pino que você conectou o piezo const int
umidade_muito_umido = 80;
const int umidade_umido = 65; const int umidade_pouco_umido = 30; const int
umidade_seco = 15;
// Inicialize o LCD
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// Pino do sensor de umidade do solo const int sensorPin = A0;
const int sensorTempPin = A1;

unsigned long lastBipTime = 0; // Variável para rastrear o último bip
// Ícones
const unsigned char love [] PROGMEM = {
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x7f,
0xff, 0x00, 0x00, 0x00, 0x03, 0xfd, 0x5f, 0xc0, 0x00, 0x00, 0x0f, 0xc0, 0x03, 0xf0, 0x00,
0x00, 0x1e, 0x00, 0x00, 0x78, 0x00, 0x00, 0x38, 0x00, 0x00, 0x3c, 0x00, 0x00, 0xf0, 0x00,
0x00, 0x0f, 0x00,
0x00, 0xe0, 0x00, 0x00, 0x07, 0x00, 0x01, 0xc0, 0x00, 0x00, 0x03, 0x80, 0x03, 0x80, 0x00,
0x00,
0x01, 0xc0, 0x07, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x07, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x0e,
0x00, 0x00, 0x00, 0x00, 0x70, 0x0c, 0x06, 0x38, 0x0c, 0x70, 0x30, 0x0c, 0x1f, 0xfc, 0x3f,
0xf8, 0x38, 0x1c, 0x3f, 0xde, 0x77, 0xdc, 0x38, 0x18, 0x31, 0xc6, 0x73, 0x8c, 0x18, 0x18,
0x70, 0x06, 0x60, 0x0e, 0x18, 0x38, 0x30, 0x06, 0x60, 0x0c, 0x1c, 0x30, 0x38, 0x0e, 0x70,
0x1c, 0x0c, 0x30, 0x18,
0x1c, 0x38, 0x18, 0x0c, 0x30, 0x1e, 0x38, 0x1c, 0x78, 0x0c, 0x30, 0x0f, 0xf0, 0x0f, 0xf0,
0x0c, 0x30, 0x03, 0xc0, 0x03, 0xc0, 0x0c, 0x30, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x30, 0x00,
0x00, 0x00,
0x00, 0x0c, 0x30, 0x10, 0x00, 0x00, 0x08, 0x0c, 0x38, 0x38, 0x00, 0x00, 0x18, 0x1c, 0x18,
0x1c,
0x00, 0x00, 0x38, 0x18, 0x18, 0x0e, 0x00, 0x00, 0x70, 0x18, 0x1c, 0x07, 0x80, 0x01, 0xe0,
0x38,
0x0c, 0x03, 0xc0, 0x03, 0xc0, 0x30, 0x0e, 0x00, 0xf8, 0x1f, 0x00, 0x70, 0x06, 0x00, 0x7f,
0xfe, 0x00, 0x60, 0x07, 0x00, 0x0f, 0xf0, 0x00, 0xe0, 0x03, 0x80, 0x02, 0xa0, 0x01, 0xc0,
0x03, 0x80,
0x00, 0x00, 0x01, 0xc0, 0x01, 0xc0, 0x00, 0x00, 0x03, 0x80, 0x00, 0xe0, 0x00, 0x00, 0x07,
0x00,
0x00, 0x78, 0x00, 0x00, 0x1e, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x7c, 0x00, 0x00, 0x0f, 0x80,
0x01,
0xf0, 0x00, 0x00, 0x07, 0xf0, 0x17, 0xe0, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00, 0x00, 0x00,
0x00,
0x1f, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
};

const unsigned char crying [] PROGMEM = {
```

```

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x7f,
0xff, 0x00, 0x00, 0x00, 0x03, 0xfd, 0x5f, 0xc0, 0x00, 0x00, 0x0f, 0xc0, 0x03, 0xf0, 0x00,
0x00, 0x1e, 0x00, 0x00, 0x78, 0x00, 0x00, 0x38, 0x00, 0x00, 0x3c, 0x00, 0x00, 0xf0, 0x00,
0x00, 0x0f, 0x00,
0x00, 0xe0, 0x00, 0x00, 0x07, 0x00, 0x01, 0xc0, 0x00, 0x00, 0x03, 0x80, 0x03, 0x80, 0x00,
0x00,
0x01, 0xc0, 0x07, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x07, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x0e,
0x07,
0x80, 0x03, 0xc0, 0x70, 0x0c, 0x1f, 0xf0, 0x07, 0xf8, 0x30, 0x0c, 0x3d, 0x78, 0x1f, 0x7c,
0x38, 0x1c, 0x30, 0x38, 0x1c, 0x0c, 0x38, 0x18, 0x00, 0x00, 0x00, 0x00, 0x18, 0x18, 0x00,
0x00, 0x00,
0x00, 0x18, 0x38, 0x00, 0x00, 0x00, 0x00, 0x1c, 0x30, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x30,
0x00, 0x00, 0x0c, 0x30, 0x00, 0x0c, 0x30, 0x00, 0x0f, 0xf0, 0x00, 0x0c, 0x30, 0x00, 0x3f, 0xfc,
0x00, 0x0c, 0x30, 0x00, 0xf0, 0x0f, 0x00, 0x0c, 0x30, 0x01, 0xe0, 0x07, 0x80, 0x0c, 0x30,
0x01, 0x80, 0x01,
0x80, 0x0c, 0x30, 0x03, 0x80, 0x01, 0xc0, 0x0c, 0x38, 0x03, 0x00, 0x00, 0xc0, 0x1c, 0x18,
0x07,
0x00, 0x00, 0xe0, 0x18, 0x18, 0x06, 0x00, 0x00, 0x60, 0x18, 0x1c, 0x07, 0x7f, 0xfe, 0xf0,
0x38,
0x0c, 0x0f, 0xff, 0xff, 0xe0, 0x30, 0x0e, 0x07, 0xd0, 0x0b, 0xf0, 0x70, 0x06, 0x00, 0x00,
0x00, 0x00, 0x60, 0x07, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x03, 0x80, 0x00, 0x00, 0x01, 0xc0,
0x03, 0x80,
0x00, 0x00, 0x01, 0xc0, 0x01, 0xc0, 0x00, 0x00, 0x03, 0x80, 0x00, 0xe0, 0x00, 0x00, 0x07,
0x00,
0x00, 0x78, 0x00, 0x00, 0x1e, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x0f, 0x80,
0x01, 0xf0, 0x00, 0x00, 0x07, 0xf0, 0x0f, 0xe0, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00, 0x00,
0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00
};
const unsigned char happy [] PROGMEM = {
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x7f,
0xff, 0x00, 0x00, 0x00, 0x03, 0xfd, 0x5f, 0xc0, 0x00, 0x00, 0x0f, 0xc0, 0x03, 0xf0, 0x00,
0x00, 0x1e, 0x00, 0x00, 0x78, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x3c, 0x00, 0x00, 0xf0, 0x00,
0x00, 0x0f, 0x00,
0x00, 0xe0, 0x00, 0x00, 0x07, 0x00, 0x01, 0xc0, 0x00, 0x00, 0x03, 0x80, 0x03, 0x80, 0x00,
0x00,
0x01, 0xc0, 0x07, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x07, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x0e,
0x00,
0x00, 0x00, 0x00, 0x70, 0x0c, 0x00, 0x00, 0x00, 0x00, 0x30, 0x0c, 0x00, 0x00, 0x00, 0x00,
0x38,
0x1c, 0x00, 0x00, 0x00, 0x00, 0x38, 0x18, 0x07, 0xc0, 0x03, 0xe0, 0x18, 0x18, 0x1f, 0xf0,
0x0f, 0xf8, 0x18, 0x38, 0x3c, 0xf8, 0x1e, 0x3c, 0x1c, 0x30, 0x30, 0x18, 0x18, 0x0c, 0x0c,
0x30, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x30, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x30, 0x00, 0x00,
0x00, 0x00, 0x0c,
0x30, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x30, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x30, 0x00, 0x00,
0x00,
0x00, 0x0c, 0x30, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x38, 0x04, 0x00, 0x00, 0x20, 0x1c, 0x18,
0x06,
0x00, 0x00, 0x60, 0x18, 0x18, 0x07, 0x80, 0x01, 0xe0, 0x18, 0x1c, 0x03, 0xc0, 0x03, 0xc0,
0x38,
0x0c, 0x00, 0xf8, 0x1f, 0x00, 0x30, 0x0e, 0x00, 0x7f, 0xfe, 0x00, 0x70, 0x06, 0x00, 0x0f,
0xf0, 0x00, 0x60, 0x07, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x03, 0x80, 0x00, 0x00, 0x01, 0xc0,
0x03, 0x80,
0x00, 0x00, 0x01, 0xc0, 0x01, 0xc0, 0x00, 0x00, 0x03, 0x80, 0x00, 0xe0, 0x00, 0x00, 0x07,
0x00,
0x00, 0x78, 0x00, 0x00, 0x1e, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x7c, 0x00, 0x00, 0x0f, 0x80,
0x01,
0xf0, 0x00, 0x00, 0x07, 0xf0, 0x17, 0xe0, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00, 0x00, 0x00,
0x00,

```

```

0x1f, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00
};
const unsigned char bad [] PROGMEM = {
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x7f,
0xff, 0x00, 0x00, 0x00, 0x03, 0xfd, 0x5f, 0xc0, 0x00, 0x00, 0x0f, 0xc0, 0x03, 0xf0, 0x00,
0x00, 0x1e, 0x00, 0x00, 0x78, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x3c, 0x00, 0x00, 0xf0, 0x00,
0x00, 0x0f, 0x00,
0x00, 0xe0, 0x00, 0x00, 0x07, 0x00, 0x01, 0xc0, 0x00, 0x00, 0x03, 0x80, 0x03, 0x80, 0x00,
0x00,
0x01, 0xc0, 0x07, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x07, 0x00, 0x00, 0x00, 0x00, 0xe0, 0x0e,
0x00,
0x00, 0x00, 0x00, 0x70, 0x0c, 0x00, 0x00, 0x00, 0x00, 0x30, 0x0c, 0x00, 0x00, 0x00, 0x00,
0x38,
0x1c, 0x00, 0x00, 0x00, 0x00, 0x38, 0x18, 0x00, 0x00, 0x00, 0x00, 0x18, 0x18, 0x00, 0x00,
0x00,
0x00, 0x18, 0x38, 0x00, 0x00, 0x00, 0x00, 0x1c, 0x30, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x30,
0x00,
0x18, 0x18, 0x00, 0x0c, 0x30, 0x00, 0x38, 0x1c, 0x00, 0x0c, 0x30, 0x00, 0xf0, 0x0e, 0x00,
0x0c, 0x30, 0x1f, 0xe0, 0x03, 0xf8, 0x0c, 0x30, 0x1f, 0x00, 0x01, 0xfc, 0x0c, 0x30, 0x14,
0x00, 0x00, 0x10, 0x0c, 0x30, 0x00, 0x00, 0x00, 0x0c, 0x38, 0x00, 0x00, 0x00, 0x00,
0x1c, 0x18, 0x00,
0x00, 0x00, 0x00, 0x18, 0x18, 0x00, 0x00, 0x00, 0x00, 0x18, 0x1c, 0x00, 0x07, 0xe0, 0x00,
0x38,
0x0c, 0x00, 0x1f, 0xf8, 0x00, 0x30, 0x0e, 0x00, 0x3c, 0x3c, 0x00, 0x70, 0x06, 0x00, 0x70,
0x0e,
0x00, 0x60, 0x07, 0x00, 0xe0, 0x07, 0x00, 0xe0, 0x03, 0x80, 0xc0, 0x07, 0x01, 0xc0, 0x03,
0x81,
0xc0, 0x03, 0x81, 0xc0, 0x01, 0xc0, 0x00, 0x00, 0x03, 0x80, 0x00, 0xe0, 0x00, 0x00, 0x07,
0x00,
0x00, 0x78, 0x00, 0x00, 0x1e, 0x00, 0x00, 0x3c, 0x00, 0x00, 0x7c, 0x00, 0x00, 0x0f, 0x80,
0x01,
0xf0, 0x00, 0x00, 0x07, 0xf0, 0x17, 0xe0, 0x00, 0x00, 0x00, 0xff, 0xff, 0x00, 0x00, 0x00,
0x00,
0x1f, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00
};

void setup()
{
// Inicialize o LCD com 16 colunas e 2 linhas lcd.begin(16, 2);

// Inicialize o pino do piezo como saída pinMode(piezoPin, OUTPUT);

lcd.print("Verificando solo");
}

void emitirBips(int quantidade, int duracao) { for (int i = 0; i < quantidade; ++i) {
tone(piezoPin, 1000, duracao);
delay(300); // Pausa entre os bips

// Inicialize o display OLED display.begin(SSD1306_SWITCHCAPVCC, 0x3c);
display.clearDisplay();
display.display(); delay(1000);

}
}

void loop()
{

```

```
// Leia o valor do sensor
int sensorUmidadeValue = analogRead(sensorPin); // Corrigir o nome do pino
// Converta o valor lido para um valor de umidade (ajuste conforme necessário) int umidade =
map(sensorUmidadeValue, 0, 1023, 0, 100);

// Limpe a tela do LCD lcd.clear();
// Mostre a umidade no LCD lcd.setCursor(0, 0); lcd.print("Umidade: "); lcd.print(umidade);
lcd.print("%");

// Verifique se a umidade está abaixo de um limite if (umidade <= 34)
{
  if (umidade <= umidade_pouco_umido)
  {
    lcd.setCursor(0, 1); lcd.print("Seco");

    // Exibir ícone "crying" no display OLED display.clearDisplay();
    display.drawBitmap(40, 10, crying, 48, 48, WHITE); display.display();

    if (millis() - lastBipTime >= 10000)
    {
      emitirBips(2, 200); // 2 bips curtos
      lastBipTime = millis(); // Atualize o tempo do último bip
    }
  }
  else if (umidade >= umidade_muito_umido)
  {
    lcd.setCursor(0, 1); lcd.print("Muito umido");

    // Exibir ícone "love" no display OLED display.clearDisplay();
    display.drawBitmap(40, 10, love, 48, 48, WHITE); display.display();
  }
  else if (umidade >= umidade_umido)
  {
    lcd.setCursor(0, 1); lcd.print("Umido");

    // Exibir ícone "happy" no display OLED display.clearDisplay();
    display.drawBitmap(40, 10, happy, 48, 48, WHITE); display.display();
  }
  else if (umidade >= umidade_seco)
  {
    lcd.setCursor(0, 1); lcd.print("Pouco umido");

    // Exibir ícone "bad" no display OLED display.clearDisplay();
    display.drawBitmap(40, 10, bad, 48, 48, WHITE); display.display();
  }

  delay(2000);
}

// Leia o valor do sensor de temperatura TMP36 int sensorTempValue =
analogRead(sensorTempPin);
float temperaturaCelsius = (sensorTempValue / 1024.0) * 500; // Conversão para graus Celsius
// Limpe a tela do LCD lcd.clear();

// Mostre a temperatura no LCD lcd.setCursor(0, 0); lcd.print("Temp(0C): ");
lcd.print(temperaturaCelsius); lcd.print("C");

// Aguarde mais 5 segundos antes de realizar a próxima leitura delay(2000);
}
```