

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE FÍSICA

PATRICK SOARES MENDES

Desenvolvimento de um protótipo auxiliar
para pesquisas com plantas em estresse
hídrico.

GOIÂNIA
2023



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE FÍSICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome completo do autor: Patrick Soares Mendes

Título do trabalho: Desenvolvimento de um protótipo auxiliar para pesquisas com plantas em estresse hídrico

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [] NÃO¹

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)s autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.



Documento assinado eletronicamente por **Geyverson Teixeira De Paula, Professor do Magistério Superior**, em 22/08/2023, às 14:42, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº](#)



Documento assinado eletronicamente por **Patrick Soares Mendes, Discente**, em 22/08/2023, às 14:52, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3983409** e o código CRC **BC16F545**.

PATRICK SOARES MENDES

Desenvolvimento de um protótipo auxiliar
para pesquisas com plantas em estresse
hídrico.

Trabalho de Conclusão de Curso apresentado à banca examinadora do Instituto de Física da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Bacharel em Engenharia Física.

Área de concentração: Engenharia Física

Orientador: Prof. Dr. Geyverson Teixeira de Paula

Coorientador: Dr. Marcelo Gonçalves Narciso

GOIÂNIA
2023

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Mendes, Patrick Soares

Desenvolvimento de um protótipo auxiliar para pesquisas com plantas em estresse hídrico [manuscrito] / Patrick Soares Mendes. - 2023.

LXX, 70 f.: il.

Orientador: Prof. Dr. Geyverson Teixeira de Paula; co-orientador Dr. Marcelo Gonçalves Narciso.

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Física (IF), Engenharia Física, Goiânia, 2023.

Bibliografia. Anexos. Apêndice.

Inclui abreviaturas, gráfico, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. Eletrônica. 2. Agro 4.0. 3. Estresse hídrico. 4. Mudanças climáticas. 5. IoT. I. Paula, Geyverson Teixeira de, orient. II. Título.

CDU 621.3



UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Ao(s) 16 dia(s) do mês de agosto do ano de 2023 iniciou-se a sessão pública de defesa do Trabalho de Conclusão de Curso (TCC) intitulado “Desenvolvimento de um protótipo auxiliar para pesquisas com plantas em estresse hídrico”, de autoria de Patrick Soares Mendes, do curso de Engenharia Física, do(a) Instituto de Física da UFG. Os trabalhos foram instalados pelo(a) prof. Dr. Geyverson Teixeira de Paula, EMC-UFG com a participação dos demais membros da Banca Examinadora: Prof. Sílvio Leão Vieira, IF-UFG, Dr. Luiz Henrique Reis de Jesus, EMC-UFG, e Dr. Marcelo Gonçalves Narciso, EMBRAPA Arroz e Feijão. Após a apresentação, a banca examinadora realizou a arguição do(a) estudante. Posteriormente, de forma reservada, a Banca Examinadora atribuiu a nota final de 9,0 (nove vírgula zero), tendo sido o TCC considerado aprovado.

Proclamados os resultados, os trabalhos foram encerrados e, para constar, lavrou-se a presente ata que segue assinada pelos Membros da Banca Examinadora.



Documento assinado eletronicamente por **Geyverson Teixeira De Paula, Professor do Magistério Superior**, em 16/08/2023, às 11:39, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **LUIZ HENRIQUE REIS DE JESUS, Usuário Externo**, em 16/08/2023, às 11:40, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Marcelo Gonçalves Narciso, Usuário Externo**, em 16/08/2023, às 11:41, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Sílvio Leão Vieira, Coordenador**, em 22/08/2023, às 09:06, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3967187** e o código CRC **0FCF4C5E**.

AGRADECIMENTOS

Ao meu pai Adeijar, minha mãe Eliamar, meu irmão Caio César e demais familiares dedico a maior gratidão por sempre me apoiarem durante todos os anos de estudos e agirem como grandes amigos nos pontos mais difíceis da minha trajetória.

À Universidade Federal de Goiás, pelas as oportunidades abertas para a construção do meu conhecimento na minha formação profissional.

À Embrapa Arroz e Feijão, pela oportunidade de estudo e disponibilização dos materiais.

Aos professores, pelo conhecimento, ajuda e paciência com a qual guiaram minha formação profissional.

Ao professor Geyverson Teixeira e ao Marcelo Narciso, por terem sido meus orientadores e contribuírem com meu aprendizado.

Aos meus amigos que me ajudaram, diretamente ou indiretamente, durante todos os anos de estudo. Especialmente à Caíque Aiuê, Carlyne Valentin, Lucas Mendes, Millena Borges e Thayna Alves que durante diversas fases, foram suporte essencial para minha vida.

Aos meus colegas do Instituto de Física, pelo companheirismo durante os anos e pelo compartilhamento de conhecimento e bons momentos.

RESUMO

As mudanças climáticas têm o potencial de transformar a sociedade e a relação humana com o meio ambiente. Com a necessidade crescente de alimentos e as mudanças climáticas já impactando a produção global, estudos alertam para possíveis cenários preocupantes caso a exploração de recursos naturais continue. Uma solução crucial é aumentar a produção agrícola com uso eficiente de água para garantir segurança alimentar.

Desenvolver técnicas nesse sentido requer coleta precisa de dados climáticos, das plantas e do solo. A coleta indiscriminada e manual prejudica a confiabilidade das pesquisas realizadas. Para resolver isso, sistemas embarcados da rede Agro 4.0 são indicados para automatizar processos de pesquisa. Na Embrapa Arroz e Feijão, trabalhos sobre os impactos do estresse hídrico na produção das plantas são feitos, portanto, foi proposto um protótipo auxiliar, com a capacidade de coletar dados como peso do vaso de planta, perda de massa de água e umidade do solo. Desenvolveu-se um circuito eletrônico de baixo custo com o microcontrolador ESP32, a fim de executar a tarefa.

Experimentos com plantas de algodão em ambiente controlado foram conduzidos para validar o protótipo. Resultados demonstraram que os sensores selecionados, comunicação de dispositivos e armazenamento de dados funcionaram adequadamente, inclusive sendo registrados em uma planilha. Com a realização de adaptações, o protótipo tem potencial de aprimoramento para uso externo, monitorando pressão atmosférica, intensidade de exposição solar e temperatura, entre outros.

Palavras - chave: Mudanças climáticas, Agro 4.0, Estresse hídrico, IoT, Eletrônica.

ABSTRACT

Climate change has the potential to transform society and the human relationship with the environment. With the growing need for food and climate change already impacting global production, studies warn of possible worrying scenarios if the exploitation of natural resources continues. A crucial solution is to increase agricultural production with efficient use of water to ensure food security.

Developing techniques in this direction requires accurate collection of climate, plant and soil data. Indiscriminate and manual collection undermines the reliability of the research carried out. To solve this, embedded systems from the Agro 4.0 network are suitable for automating research processes. At Embrapa Rice and Beans, work is being done on the impact of water stress on plant production, so an auxiliary prototype was proposed, with the ability to collect data such as plant pot weight, water mass loss and soil moisture. A low-cost electronic circuit was developed with the ESP32 microcontroller in order to perform the task.

Experiments with cotton plants in a controlled environment were conducted to validate the prototype. Results showed that the sensors selected, device communication and data storage worked properly, including being recorded in a spreadsheet. With adaptations, the prototype has the potential to be improved for external use, monitoring atmospheric pressure, intensity of sun exposure and temperature, among others.

Key - words: Climate Changes, Agro 4.0, Hydric stress, IoT, Eletronics.

LISTA DE FIGURAS

Figura 2.1:	Representação genérica de um microcontrolador.	20
Figura 2.2:	Exemplo de um sinal em modo síncrono.	23
Figura 2.3:	Exemplo de sinal assíncrono.	23
Figura 2.4:	Figura ilustrativa dos tipos de transmissões seriais.	24
Figura 2.5:	Referência de pinos.	28
Figura 3.1:	Casa de vegetação protegido 8	29
Figura 3.2:	Balança WAAGEN KN P 100/10.	30
Figura 3.3:	Válvula solenoide.	31
Figura 3.4:	Ciclo de operação da válvula. O tempo de permanência da corrente aplicada são da ordem de milissegundos. Normalmente, um tempo de 500 ms funciona corretamente. Elaboração própria.	31
Figura 3.5:	LN298N Ponte-H Dupla.	32
Figura 3.6:	Sensor resistivo de umidade de solo.	32
Figura 3.7:	Módulo DB9 Conversor Serial RS232-TTL	33
Figura 3.8:	Fluxograma do código.	36
Figura 3.9:	Circuito referente às leituras dos dados das balanças.	37
Figura 3.10:	Circuito referente à leituras dos sensores de umidade de solo.	38
Figura 3.11:	Circuito referente à alimentação do circuito.	38
Figura 3.12:	Conexões na ESP32.	39
Figura 3.13:	Experimento de testes.	39
Figura 3.14:	Visão frontal e traseira do protótipo.	40
Figura 3.15:	Visão diagonal e interna do protótipo.	40
Figura 3.16:	Vasos com plantas de algodão no início do experimento 1.	42
Figura 3.17:	Válvula instalada para irrigação.	42
Figura 3.18:	Vasos com plantas de algodão no experimento 2.	43
Figura 4.1:	Plantas no final do experimento.	45
Figura 4.2:	Experimento com os vasos de algodão.	46
Figura 4.3:	Água perdida no vaso de estresse, em relação ao peso inicial. A regressão apresenta $R^2 = 0,987$	47

Figura 4.4:	Experimento 2 com os vasos de algodão.	48
Figura 4.5:	Progressão dos vasos.	49
Figura 4.6:	Umidade superficiais dos vasos.	50
Figura B.1:	Esquema elétrico do circuito.	68

LISTA DE TABELAS

Tabela 2.1:	Comunicação Serial vs Paralela.	21
Tabela 3.1:	Potência consumida pelo circuito elétrico.	41
Tabela 3.2:	Parâmetros iniciais do experimento com plantas de algodão.	41
Tabela 3.3:	Parâmetros iniciais do experimento 2 com plantas de algodão.	43
Tabela 4.1:	Dados obtidos no experimento do algodão.	45
Tabela 4.2:	Perda de água no período 1.	47
Tabela 4.3:	Perda de água no período 2.	47
Tabela 4.4:	Dados obtidos no experimento 2 do algodão.	48
Tabela 4.5:	Perda de água no período 1.	49
Tabela 4.6:	Perda de água no período 2.	49
Tabela C.1:	Tabela com dados obtidos a cada 5 minutos.	69
Tabela C.2:	Tabela com dados obtidos a cada 10 minutos.	70
Tabela C.3:	Tabela com dados obtidos a cada 30 minutos.	70

LISTA DE ABREVIACOES

ULA	Unidade Lgica e aritmtica
CPU	Central Processing Unit
ROM	Read Only Memory
RAM	Random Access Memory
RS	Recomended Standard
UART	Universal Asynchronous Receiver-Transmitter
USART	Universal Synchronous-Asynchronou Receiver and Transmitter
USB	Universal Serial Bus
I2C	Inter-Intergrated Circuit
SPI	Serial Peripheral Interface
CAN	Controller Area Network
PWM	Pulse Width Modulation
HDMI	High-Definition Multimedia Interface
Embrapa	Empresa Brasileira de Pesquisa Agropecuria
RTC	Real Time Clock
DAC	Digital-to-Analog converter
ADC	Analog-to-Digital converter
GPIO	General Purpose Input/Output

SUMÁRIO

Capítulo 1: Introdução	16
Capítulo 2: Plataforma e Desenvolvimento	19
2.1 Microcontroladores	19
2.2 Comunicação	21
2.2.1 Protocolos de comunicação	22
2.3 Banco de dados	25
2.4 Placas disponíveis	25
2.5 ESP32	27
Capítulo 3: Metodologia	29
3.1 Materiais	29
3.1.1 Balança	29
3.1.2 LN298 Ponte-H Dupla e Válvula Solenoide	30
3.1.3 Sensor de umidade	32
3.1.4 Módulo DB9 Conversor Serial RS232-TTL	33
3.1.5 Demais componentes	33
3.2 Programação	34
3.2.1 Google planilhas	37
3.3 Circuito eletrônico	37
3.3.1 Protótipo	39
3.4 Experimentos	41
Capítulo 4: Resultados	44
4.1 Experimento 1	44
4.2 Experimento 2	48
Capítulo 5: Conclusões	51
Referências Bibliográficas	52

APÊNDICES	55
Apêndice A: Códigos desenvolvidos	56
Apêndice B: Esquema elétrico do circuito	67
Apêndice C: Tabelas	69

INTRODUÇÃO

A garantia da segurança alimentar para a população é um dos grandes desafios que estão em pauta em diversos países pelo mundo. É definida pelo Ministério da Saúde do Brasil como:

"A Segurança Alimentar e Nutricional (SAN) consiste na realização do direito de todos ao acesso regular, permanente e irrestrito a alimentos de qualidade, quer diretamente ou por meio de aquisições financeiras, a alimentos seguros em quantidade e qualidade adequadas e suficientes, sem comprometer o acesso a outras necessidades essenciais." [Insegurança Alimentar e Nutricional — Ministério da Saúde]

Conforme o Panorama da Segurança Alimentar e Nutricional na América Latina e no Caribe (2020) [FAO et al. 2021], houve melhoras significativas nos níveis de subalimentação da população latina desde o início do século XXI até meados da década de 2010, porém foi posteriormente observado um aumento nestas taxas. Além disso, a baixa qualidade, pouca variedade, assim como a frequência e a quantidade acima do recomendado para uma refeição saudável, também agem como fatores que reduzem a segurança alimentar da população. Fatores externos como mudanças climáticas, fenômenos meteorológicos e climáticos extremos podem entrar como agentes decisivos na alimentação da população, principalmente das pessoas mais pobres e vulneráveis.

As perspectivas vislumbradas para o futuro mostram grandes desafios para a civilização humana. Um dos principais atuantes que se insere neste contexto são as mudanças climáticas. É previsto que até o ano de 2100, caso o ritmo das alterações climáticas continuem o mesmo, haverá grande declínio na produção de cultivos de cereais, com a estimativa na queda de produção podendo chegar a 40% para o caso do milho, 50% para o trigo e de 30% para o arroz [Arora 2019].

Atualmente a agropecuária é um dos grandes expoentes econômicos brasileiros. Segundo a CNA (Confederação da Agricultura e Pecuária do Brasil) [CNA 2023], agropecuária representou 21,6% do crescimento brasileiro no primeiro trimestre de 2023,

apresentando a maior alta desde 1996. Este dado ressalta a importância econômica dos cultivos e outras atividades do campo no cenário da economia brasileira. O relatório aponta dois pontos importantes: O crescimento de 3,5% no consumo de produtos agropecuários em relação ao período homólogo e a variação na produção das lavouras diversos produtos, apesar do crescimento registrado em relação ao período homólogo na lavoura de soja de 24,7%, milho 1º safra de 9,8% e cana-de-açúcar de 6,5%. Quedas foram registradas na produção do arroz (7,5%), batata (2,8%) e trigo (1,7%). Uma das justificativas para esta queda está em adversidades climáticas enfrentadas no campo no final do ano de 2022.

As mudanças climáticas enfrentadas pelo Brasil já são pauta de discussão entre os pesquisadores brasileiros. Em [Assad et al. 2020], é feito um estudo buscando identificar quais setores da agricultura no Cerrado serão afetados e como isto ocorreria. Observa-se nos cenários levantados que a cultura da soja e arroz sofreram as maiores reduções em potencial econômico. Foi neste contexto de demanda crescente de produtividade e desenvolvimento de novas técnicas aplicada nas ciências agrárias que surge a Embrapa (Empresa Brasileira de Pesquisa Agropecuária) no ano de 1973. Hoje a instituição conta com 43 unidades espalhadas pelo Brasil, tendo como foco de pesquisa áreas relevantes para agropecuária brasileira, entre elas: Embrapa Arroz e Feijão (Santo Antônio de Goiás, Goiás), Embrapa Caprinos e Ovinos (Sobral, Ceará), Embrapa Clima Temperado (Pelotas, Rio Grande do Sul), Embrapa Instrumentação (São Carlos, São Paulo), Embrapa Pesca e Aquicultura (Palmas, Tocantins), entre outras. Entre os anos de 2021 e 2022, a Embrapa foi responsável por 7982 publicações científicas abrangendo diversos temas, desde a relação das comunidades tradicionais com o cenário agropecuário, até desenvolvimento genético de plantas. Em [Costa et al. 2023], é descrito um estudo feito na Embrapa sobre o aprimoramento de cultivos de arroz de terras altas, é verificado se os efeitos do silício e de agentes biológicos, pré-tratados em ambiente em stress hídrico, auxiliam no desenvolvimento e produção da planta. Para a execução deste estudo, a automação de processos pode ser uma grande ferramenta na obtenção de dados, auxiliando os pesquisadores.

Esta integração tecnológica na agropecuária é normalmente chamada de Agro 4.0. Segundo [Massruhá e Leite 2017], Agro 4.0 pode ser definido como:

"A Agro 4.0 emprega métodos computacionais de alto desempenho, rede de sensores, comunicação máquina para máquina (M2M), conectividade entre dispositivos móveis, computação em nuvem, métodos e soluções analíticas para processar grandes volumes de dados e construir sistemas de suporte à tomada de decisões de manejo. Além disso, contribuirá para elevar os índices de produtividade, da eficiência do uso de insumos, da redução de custos com mão de obra, melhorar a qualidade do trabalho e a segurança dos trabalhadores e diminuir os impactos ao meio ambiente." [Massruhá e Leite 2017]

Esta nova abordagem do ambiente de produção agrícola, tornou-se um ponto focal de grande parte dos centros de pesquisa pelo mundo. Em [Chamara et al. 2022], é feita uma

revisão bibliográfica de pesquisas publicadas sobre sistemas embarcados em ambientes agrícolas entre os anos de 2011 e 2021. O método usado pelo estudo aponta um pico de publicações nos anos de 2017 a 2019, com mais de 50 publicações na área. Índia e China apresentam o maior número de publicações com 19 e 14, respectivamente, seguidas pelos Estados Unidos com 10 publicações. Há diversos protocolos de comunicação que podem ser utilizados em sistemas embarcados, cada método deverá ser escolhido consoante a aplicação e recursos disponíveis. Os protocolos mais utilizados são a comunicação Wi-Fi com 34%, LoRa (17%) e ZigBee (14%). Para o armazenamento destes dados, é amplamente utilizado serviços de nuvem, que por sua vez carregam vantagens e desvantagens em sua aplicação. Os serviços mais utilizados são o Thingspeak (28%), Microsoft Azure, Ubidots e sistemas próprios desenvolvidos com 12% e Firebase com 8% [Chamara et al. 2022].

Neste trabalho é proposto o desenvolvimento de um protótipo que seja capaz de monitorar plantas utilizadas em experimentos que compõem os estudos de estresse hídrico em plantas. Estes experimentos se baseiam em acompanhar a perda de massa de água das plantas, dado um período pré-definido, além de obter a umidade superficial do vaso da planta. Com estes objetivos, foi projetado um experimento com dois vasos de plantas de algodão, um vaso definido como referência. Este vaso de referência é irrigado periodicamente caso seu peso ultrapasse um mínimo determinado. A partir dos dados coletados, torna-se possível rastrear o impacto da escassez hídrica na produção das plantas, comparando o desempenho produtivo da planta de referência com a que esteve em escassez de água.

A utilização do protótipo na obtenção destes dados propostos anteriormente, permite ao pesquisador adicionar padronização e confiança nos dados obtidos, características essenciais que podem ser prejudicadas pelo registro manual. Nesta fase de desenvolvimento, o projeto foi concretizado para uma casa de vegetação, onde há as balanças onde são postos os vasos. Esta escolha foi feita devido ao ambiente ser controlado e por ser o local onde as pesquisas em estresse hídrico ocorrerem. O produto apresenta, realizando adaptações de seus componentes, potencial para ser expandido para aplicações além da casa de vegetação, como por exemplo, obter dados em tempo real das condições do solo e climáticas do campo remotamente, além de proporcionar informações para estudo do controle do consumo de água utilizado em irrigações, visto que é possível concluir a quantidade de água mínima necessária pela planta que permite a produtividade requerida. Outro caso de destaque, é na seleção genética das plantas que melhor se adapta a condições hostis no campo.

PLATAFORMA E DESENVOLVIMENTO

O circuito a ser desenvolvido deverá ser capaz de se comunicar com sensores de umidade e com a balança instalada na casa de vegetação, além de ter a capacidade de exportar estes dados para um banco de dados. Para alcançar este objetivo, os microcontroladores se mostram como boas ferramentas para desempenharem estas tarefas.

Há a disponibilidade no mercado de diversas placas que conseguem executar as tarefas exigidas para este trabalho, sendo necessário a realização de uma análise para a definição da placa mais adequada dentre elas. A ESP32 é um microcontrolador que se mostrou mais indicado para esta aplicação, visto que, além de conseguir comunicar com os dispositivos necessários e controlar a válvula que será utilizada na irrigação, a placa contém um módulo para comunicação Wi-Fi e Bluetooth embutidos, ferramentas que permitem a transferência e armazenamento de dados.

2.1 Microcontroladores

Os microcontroladores são dispositivos eletrônicos que integram em um único *chip* um processador, unidade lógica e aritmética (ULA) e periféricos. Eles são projetados para controlar sistemas embarcados e executar funções específicas de forma autônoma.

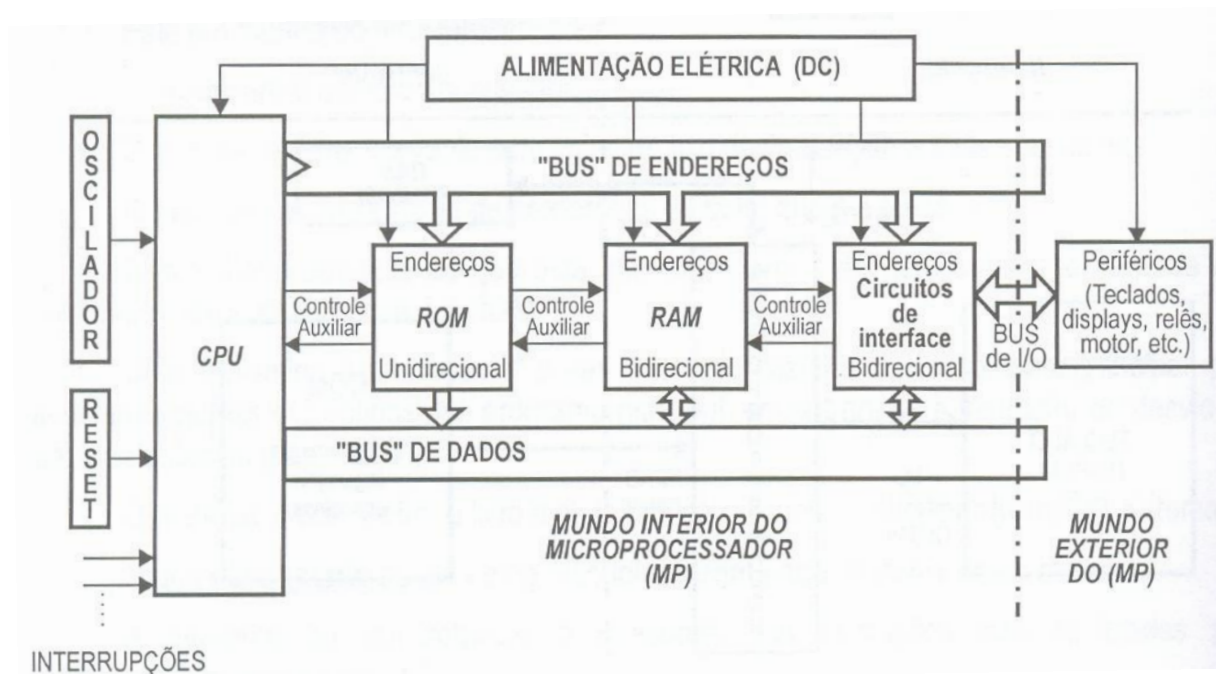
O surgimento dos microcontroladores remonta a década de 1970. O primeiro microcontrolador foi chamado de 8048 e foi desenvolvido pela Intel, mas foi seu sucessor o 8051, que teve maior popularidade entre os usuários. Este chip apresentava maior robustez que seu antecessor e sua programação era feita em Assembly. Outro microcontrolador popular que surgiu nesta época foi o PIC1650, desenvolvido pela General Instrument. Posteriormente, esta equipe de desenvolvimento com o nome de Microchip Technology continuou com o projeto, surgindo a partir de então toda a família de microcontroladores PIC.

Devido a esta capacidade de execução de tarefas, os microcontroladores estão presentes em diversas áreas de automação, sendo os principais responsáveis por monitora-

mento e controle de processos. Sua utilização se estende da automação residencial [Chong, Zhihao e Yifeng 2011], aumento de produtividade no ramo alimentício [Dadhaneeya, Nema e Arora 2023], aprimoramento e mitigação problemas relacionados a sistemas de reabilitação de pacientes clínicos [Fan et al. 2014], na agricultura por meio do sensoriamento das condições climáticas e de solo das plantações [Chamara et al. 2022], na implementação ou aprimoramento de processos industriais e de processos complexos [Xu et al. 2014].

Os responsáveis pelos cálculos e pelas tomadas de decisões dos microcontroladores são chamados de microprocessadores. São constituídos pela CPU (*Central Processing Unit*¹), ROM (*Read Only Memory*²), RAM (*Random access Memory*³), além dos barramentos (*Bus*) e o cristal oscilador. Estes componentes estão expostos na figura (2.1).

Figura 2.1: Representação genérica de um microcontrolador.



Fonte: [Nisolosi 2009]

As memórias ROM e RAM têm funções distintas em um sistema computacional. A ROM é responsável por armazenar as instruções que serão executadas pelo microprocessador. Já a RAM é utilizada temporariamente para armazenar informações que serão necessárias durante a execução dessas instruções. A CPU é o componente principal do microprocessador e funciona como o "cérebro" do sistema. É nela que os cálculos são re-

1. Tradução: Unidade central de processamento

2. Tradução: Memória apenas de leitura

3. Tradução: Memória de Acesso Aleatório

alizados com base nas instruções pré-gravadas na memória ROM. Juntos a CPU temos dois elementos, o oscilador e o *reset*. O oscilador é responsável de temporizar o processamento, assim gerando sincronidade dos processos e determinando sua velocidade. Já o *reset* determina o início das instruções internas da CPU e realoca o endereço da primeira instrução no endereço inicial (0000h).

Outro elemento importante para o funcionamento do microprocessador são barramentos de I/O (*Input/Output*⁴). São estes barramentos que se comunicam com o mundo exterior e assim permitindo o controle e monitoramento de periféricos. O conjunto destes elementos em uma única placa, é chamado de microcontrolador. Esta placa pode conter outros elementos que complementam o funcionamento do microprocessador, como a comunicação serial e um contador de tempo.

2.2 Comunicação

A comunicação entre diferentes dispositivos é essencial para o funcionamento de sistemas eletrônicos. Há duas formas principais de comunicação, a serial e paralela. A tabela abaixo mostra a comparação entre estes dois métodos:

Tabela 2.1: Comunicação Serial vs Paralela.

Comunicação serial	Comunicação paralela
Envia dado bit a bit em um pulso de clock	Vários bits de dados retransmitidos por vez
Requer um fio para transmitir dados	Requer "n" número de linhas para transmitir "n" bits
Velocidade de comunicação lenta	Velocidade de comunicação rápida
Baixo custo de instalação	Alto custo de instalação
Preferido para comunicação de longa distância	Usado para comunicação de baixa distância
Exemplo: Computador para computador	Computador para impressora multifuncional

Fonte: [Dawoud e Dawoud 2022], tradução nossa.

Os microcontroladores utilizam a comunicação serial para envio e recepção de dados, devido este método ser o mais vantajoso. Para a comunicação serial ser feita, é necessário definir qual padrão de comunicação serial será utilizado. O mais conhecido e usado é o RS-232, onde "RS" significa *Recommended standard*⁵. Para o correto funcionamento da comunicação de dados é preciso definir dois parâmetros, o protocolo de comunicação e a maneira que o sinal será enviado pelos fios.

4. Tradução: Entrada/Saída

5. Padrão Recomendado

2.2.1 Protocolos de comunicação

Os protocolos de comunicação são divididos em dois tipos, o protocolo inter-sistemas e intra-sistemas. O protocolo inter-sistemas são aqueles que comunicam entre diferentes dispositivos por meio de um barramento. Tem-se três categorias de protocolos dentro desta classificação:

- UART - *Universal Asynchronous Receiver-Transmitter*⁶.
- USART - *Universal Synchronous-Asynchronous Receiver and Transmitter*⁷.
- USB - *Universal Serial Bus*⁸.

Os protocolos intra-sistemas são aqueles que a comunicação é feita entre os dispositivos dentro do mesmo circuito eletrônico. Tem-se três categorias dentro desta classificação:

- I2C - *Inter-Integrated Circuit ou IIC*⁹.
- SPI - *Serial Peripheral Interface*¹⁰.
- CAN - *Controller Area Network*¹¹.

Os protocolos de comunicação operam em três modos: síncrono, assíncrono e bit-síncrono. Exemplos de comunicação assíncrona e síncrona podem ser vistos nas figuras (2.2) e (2.3), respectivamente. O modo síncrono utiliza um sinal de *clock* para temporizar a transferência de informações e devido a isso, surge como sendo o mais rápido. Por outro lado, tem-se a necessidade da adição de um fio extra (sinal do *clock*) para a comunicação ser completa. Dentro do modo síncrono, existe uma variação que utiliza o nível de um bit para sincronização além do sinal de *clock*, esta técnica é chamada de bit-sincronização.

Os protocolos de comunicação que utilizam este modo de operação são o USART, USB, I2C, SPI e CAN.

Já o modo assíncrono, os dados são sincronizados utilizando um bit inicial indicando o início do pacote de dado e outro bit indicando o final do pacote.

Além dos bits de início e parada, é possível utilizar um bit de paridade para identificação de erros na comunicação. Esse bit pode ser configurado como paridade

6. Receptor-Transmissor Assíncrono Universal

7. Receptor e Transmissor Universal Síncrono-Assíncrono

8. Barramento Serial Universal

9. Este termo é amplamente referenciado somente como I2C (I-dois-C) em qualquer língua utilizada, sua tradução literal para o português incorre em erro gramatical.

10. Interface Periférica Serial

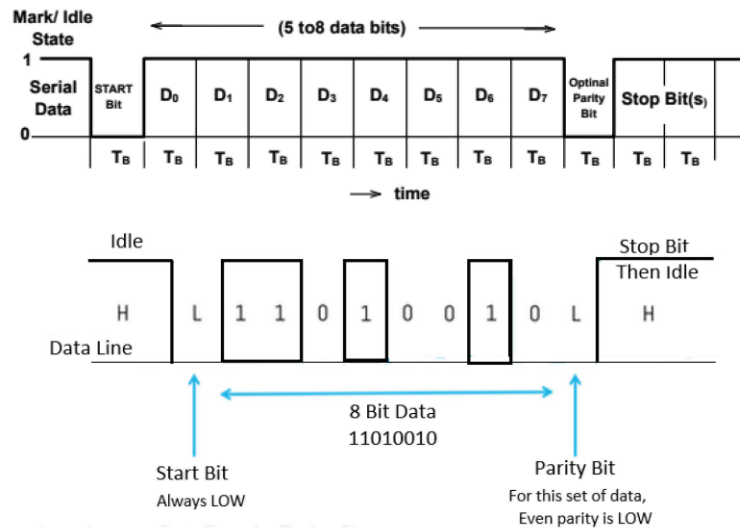
11. Rede de área de Controlador

Figura 2.2: Exemplo de um sinal em modo síncrono.



Fonte: [Dawoud e Dawoud 2022].

Figura 2.3: Exemplo de sinal assíncrono.



Fonte: [Dawoud e Dawoud 2022].

ímpar ou par. Ao receber um pacote de dados, o receptor verifica a quantidade de bits "1" presentes no pacote. Em seguida, compara essa quantidade com o valor esperado, definido previamente entre o transmissor e o receptor. Se a paridade for ímpar e a quantidade de bits "1" for par, o bit de paridade é definido como "1". Assim, o receptor pode comparar se o número de bits recebidos corresponde ao número de bits pré-definido. Caso não sejam iguais, indica-se que houve alguma corrupção nos dados enviados. Esse protocolo de verificação é muito simples, mas possui algumas limitações. Ele é capaz de detectar a ocorrência de um único bit alterado devido a erros na transmissão, porém não tem a capacidade de corrigir esse erro.

O protocolo UART é o mais popular que utiliza a transmissão assíncrona no modo de transmissão padrão RS232. Como visto, a grande vantagem da comunicação assíncrona é não depender de um sinal de clock para a sincronização, porém é necessário

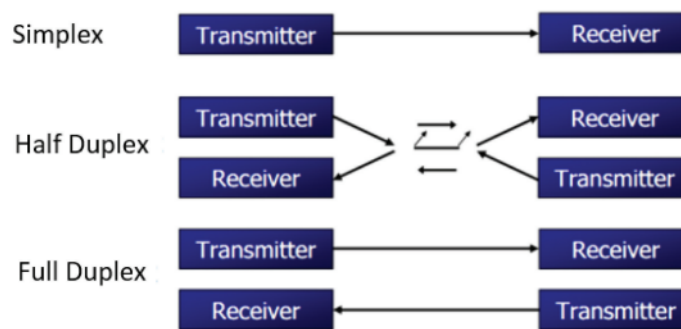
que ambos dispositivos estejam configurados para o mesmo *baud rate*. *Baud rate* é a taxa de transferência de bits por unidade de tempo, ou seja:

$$\text{Baud rate} = \frac{1}{T_B} \quad (2.1)$$

Onde T_B é o espaço de tempo do sinal do bit. Valores típicos desta comunicação são de 4800, 9600, 115200 e 256000 bits por segundo.

Por fim, definidos o modelo padrão e o protocolo de comunicação, parte-se para a seleção do tipo de transmissão que será feita. Existem três formas: *Simplex*, *Half-Duplex* e *Full-Duplex*.

Figura 2.4: Figura ilustrativa dos tipos de transmissões seriais.



Fonte: [Dawoud e Dawoud 2022].

Os dois fatores que diferenciam os três tipos de comunicação é a direção que o dado é enviado (unidirecional ou bidirecional) e o momento que o dado é enviado.

Nas transmissões *Half-Duplex* e *Full-Duplex* o dado é bidirecional, porém enquanto na *Full-Duplex* os dados podem ser enviados nas duas direções ao mesmo tempo, na *Half-Duplex* a transmissão só poderá ocorrer em uma direção por vez. Telefones celulares funcionam por meio da transmissão *Full-Duplex*, já *walkie-talkie*, ou transceptor de mão, funcionam por meio de *Half-Duplex*. A comunicação *Simplex* é a mais básica de todas, ela funciona de maneira unidirecional, exemplos desta comunicação são sinais de TV e Rádio. O diagrama de blocos expresso na figura (2.4) resume essas relações.

2.3 Banco de dados

O armazenamento de informações é uma dos principais atributos requeridos para sistemas embarcados, pois o registro de sensores e/ou outras variáveis de interesse será utilizado para tomada de decisões posteriores. Existem diversas formas de gravar estas informações, entre elas:

- **Gravação local utilizando MicroSD:** A comunicação do MicroSD com microcontroladores é simples, visto que já estão disponíveis no mercado módulos que realizam essa função. Apesar do baixo custo de implementação e da grande capacidade de armazenamento, este método sofre com a necessidade de aumento no *hardware* (caso o microcontrolador não tenha uma entrada de MicroSD na placa), risco de corrupção dos dados, além de não ser possível o acompanhamento em tempo real dos dados coletados.
- **Firebase:** Plataforma do Google especializada no desenvolvimento de aplicativos *mobile*. Esta ferramenta é bastante utilizada para sistemas embarcados devido a sua robustez na gravação e exposição dos dados. Uma das grandes vantagens desta ferramenta é em sua flexibilidade de aplicações, atualização em tempo real e sistema de usuário e autenticação. A desvantagem dessa ferramenta está em necessitar de conexão com a internet durante o tempo de operação, maior complexidade de implementação e possível aumento de custos, visto que caso seja necessário um armazenamento maior de informações, deve-se trocar o plano grátis para um pago [Firebase | Google's Mobile and Web App Development Platform].
- **Google Planilhas:** Esta ferramenta também desenvolvida pelo Google pode ser utilizada para armazenamento de dados. As vantagens desta ferramenta está em sua intuitividade e familiaridade de uso, grande capacidade de armazenamento de dados e o acesso de qualquer localidade. Porém, este método conta com algumas desvantagens, como o uso da conexão com internet para armazenar informações e a visualização dos dados não é feita de maneira automática.

2.4 Placas disponíveis

Com o passar do tempo e com a sofisticação dos sistemas embarcados, placas mais robustas foram desenvolvidas. Dentre elas, as mais populares são as placas Arduino [Arduino Hardware | Arduino], Raspberry Pi [Buy a Raspberry Pi – Raspberry Pi] e ESP32 [ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems]. O Arduino conta com três placas principais [Tipos de Placas Arduino: Conheça as Principais - Blog Eletrogate]:

- Uno: Com o processador ATmega238, esta placa é a mais utilizada e mais difundida em projetos, tanto amadores, quanto profissionais em eletrônica. A placa possui 14 pinos digitais de entrada e saída, sendo 6 destes configuráveis para saídas PWM (*Pulse Width Modulation*¹²) e 6 entradas analógicas.
- Nano: Utiliza o processador ATmega238. Esta placa é direcionada para projetos compactos. Devido a isso, esta placa não tem um conector CC (Corrente Contínua) de alimentação e a conexão é Mini-B USB. A placa possui 14 pinos digitais de entrada e saída, nas quais 6 podem ser usados para saídas PWM e 8 entradas analógicas.
- Mega: Esta placa é basicamente uma extensão da versão Uno. Nesta versão é utilizado o processador ATmega2560. Possui 54 pinos digitais de entrada e saída, nos quais 15 podem ser configurados para saídas PWM, 16 entradas analógicas e 4 UART.

Dentre os modelos mais populares da Raspberry Pi, tem-se: Raspberry Pi 4 B, Raspberry Pi 3 B+ e a Raspberry Pi Zero W.

- Raspberry Pi 4B: modelo mais recente lançado. A placa conta com um processador ARM-Cortex A72, quatro núcleos 64-bit, 1.5GHz. Além disso, é disponível duas portas Micro HDMI, conexão de Ethernet, opção de RAM de 2, 4 ou 8 GB, 4 portas USB (2 USB2 e 2 USB3), conexão 802.11 b/g/n Wireless LAN, Bluetooth v5.0 e processador gráfico VideoCore VI 3D, além de outros atributos. Nota-se que esta placa é direcionada a projetos mais complexos, principalmente aqueles que envolvem processamento de imagem [Raspiberry Pi 2019].
- Raspberry Pi 3 B+: modelo da terceira geração das placas raspiberry Pi. Com o processador Broadcom BCM2837B0 ARMv8 quatro núcleos de 1.4GHz, memória RAM de 1 GB, 4 portas USB2, conexão com cabo *Ethernet*, conexão Wi-Fi duas bandas (2.4GHz e 5GHz), bluetooth v4.2 e entrada HDMI [Raspiberry Pi].
- Raspberry Pi Zero W: Esta placa é a mais compacta entre as citadas, direcionada, portanto, para projetos de dimensões menores. A placa conta com o processador Broadcom BCM2835 ARMv6 com um núcleo de 1 GHz, memória RAM DE 512 MB, entrada micro USB, saída mini-HDMI e 40 *GPIOs* (Pinos). Esta placa é a versão aprimorada da Pi Zero, pois tem Wi-Fi 2.4 Ghz [Electronic Datasheets].

A ESP32 apresenta uma enorme variedade de placas, o que garante boa flexibilidade para aplicação em diversos sistemas. Dentre essas, destaca-se: ESP32 DevKit DOIT, ESP32 SX1278 e ESP32 com Bateria.

¹²Modulação por Largura de Pulso

- ESP32 DevKit DOIT: Placa de desenvolvimento mais popular entre os usuários. Esta placa apresenta o melhor custo benefício entre as aqui citadas, devido sua ampla quantidade de GPIOs (algumas placas têm 38 GPIOs) e por ter Wi-Fi e Bluetooth integrados.
- ESP32 SX1278: Esta placa é utilizada principalmente em projetos que exigem comunicação de longo alcance (LoRa¹³). Suas especificações são basicamente as mesmas da ESP32 DevKit DOIT, porém algumas placas contam com uma tela OLED integrada, o que pode ser útil para algumas aplicações. Usualmente, a comunicação pode ser realizada a distâncias de até 2 a 5 km em ambientes urbanos, para ambientes rurais a comunicação pode chegar a 10 km [Siddique et al. 2019].
- ESP32 com Bateria: A principal vantagem desta placa é ter o espaço para acoplar uma bateria recarregável que irá prover energia para o microcontrolador. Esta placa é indicada para projetos embarcados com alta mobilidade.

Com estas opções expostas, foi escolhida a ESP32 DevKit DOIT como placa que será utilizada no projeto, devido sua disponibilidade imediata, Wi-Fi e Bluetooth integrado, facilidade de uso e implementação.

2.5 ESP32

A placa ESP32 é um microcontrolador desenvolvido pela Espressif Company [ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems] a partir do ano de 2016. Desde então, o chip apresenta uma crescente popularidade em seu uso devido sua versatilidade para diversas aplicações em IoT (*Internet of Things*¹⁴) e seu baixo consumo de energia, característica muito desejada para sistemas embarcados.

O *chip* contém dois núcleos de baixa potência do processador Xtensa, 32-bit, LX6, e um cristal oscilador de 40 MHz. Além disso, sua memória interna conta com 448 KB de ROM para funções de inicialização e dos núcleos, 520 KB de SRAM (*Static Random Access Memory*¹⁵) para dados e instruções, 8 KB de SRAM direcionados ao RTC (*Real time Clock*)¹⁶ de memória rápida e 8 KB de SRAM ao RTC de memória lenta. A ESP32 utilizada têm 30 pinos ao todo, nos quais:

- 25 x GPIOs.

¹³Long Range Communication, ou seja, comunicação de longo alcance.

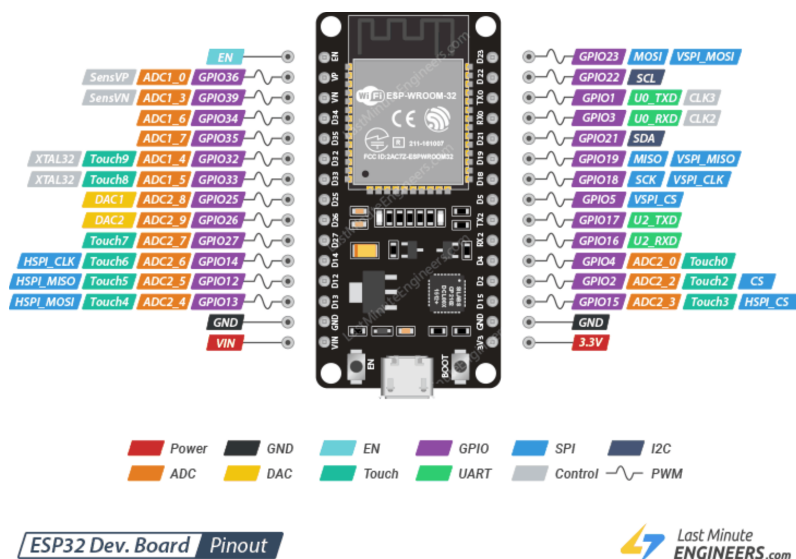
¹⁴Internet das Coisas

¹⁵Memória de Acesso Aleatório Estático

¹⁶Relógio de tempo real

- 15 x Canais ADC 12-bits.
- 3 x UART.
- 25 x Canais PWM.
- 2 x DAC 8-bits.
- 9 x sensores de toque.
- Comunicação: 4 x SPI, 2 x I2C e 2 x I2S.

Figura 2.5: Referência de pinos.



A disposição destes pinos está exposto na figura (2.5). Nota-se que cada pino tem a capacidade de exercer diferentes funções de cada vez, visto sua configuração. Para o monitoramento de grandezas físicas, dois circuitos eletrônicos presentes na ESP32 são essenciais: o ADC (*Analog-to-Digital converter*¹⁷) e o DAC (*Digital-to-Analog converter*¹⁸). Estes circuitos fazem parte da interface de comunicação do microcontrolador com o mundo externo.

Outra grande vantagem é o Wi-Fi 802.11 b/g/n com frequência de 2.4 GHz ~ 2.5GHz e Bluetooth v4.2 [Espressif System 2019].

¹⁷Conversor Analógico para Digital

¹⁸Conversor Digital para Analógico

METODOLOGIA

Este capítulo é dedicado a expor os materiais utilizados para o projeto, assim como descrever o protótipo montado. O projeto foi desenvolvido para a casa de vegetação 8, na Embrapa Arroz e Feijão, Santo Antônio de Goiás, Goiás.

Figura 3.1: Casa de vegetação protegido 8

(a) Foto externa da casa de vegetação. (b) Foto interna da casa de vegetação.



Fonte: Elaborado pelo autor



Fonte: Elaborado pelo autor.

Este ambiente foi construído especificamente para estudos sobre estresse hídrico. Na figura (3.1b), é possível ver as filas onde são colocadas as balanças. No total, a casa de vegetação conta com 12 filas, com cada uma tendo capacidade máxima de 16 balanças. Com esta configuração, a estufa é capaz de performar diferentes pesquisas simultaneamente.

3.1 Materiais

3.1.1 Balança

A balança utilizada na casa de vegetação é da marca Waagen KN P 100/10 modelo ID-03, com capacidade máxima de 100kg, resolução de 0,01 kg e linearidade de 0,01 kg.

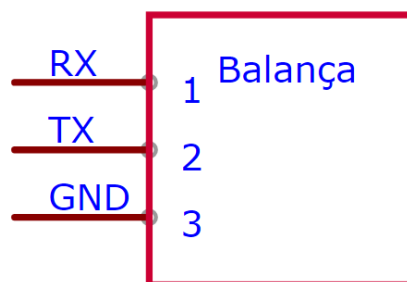
Figura 3.2: Balança WAAGEN KN P 100/10.

(a) Foto da balança.



Fonte: Elaborado pelo autor.

(b) Esquema representativo.



Fonte: Elaborado pelo autor.

Há a possibilidade de configurar duas formas de saída dos dados: impressão ou envio para um *PC* via porta serial RS232, havendo duas formas de transmissão:

- Transmissão contínua: O valor enviado pela balança é feito continuamente.
- Transmissão ao estabilizar: O valor enviado pela balança é enviado somente quando é detectada uma alteração e sua estabilização.

A taxa de transmissão (*baud rate*) pode ser alterada para 1200, 2400, 4800, 9600 e 19200 bits por segundo. Além disso, pode ser escolhido o número de bits dos dados (7 ou 8) e o tipo de paridade [KN WAAGEN].

3.1.2 LN298 Ponte-H Dupla e Válvula Solenoide

A válvula utilizada para irrigação é a GEM-BL-9EL 12V DC LATCH 9W [GEM-BL | Irrigation GEM Sol Latch Coil]. Como exposta na figura abaixo:

A válvula solenoide é um dispositivo composto por uma bobina que, ao ser energizada, provoca movimentação mecânica, sendo ela linear ou rotativa. Há três categorias de válvulas principais: solenoides do tipo *push*(empurra), *latching* e *proportional*. A escolha de cada configuração irá depender da aplicação desejada. Para sistemas que exigem uma baixa taxa de ativação da válvula recomenda-se a utilização de válvulas tipo *latching*, pois é possível ajustar e mantê-la na posição desejada sem a alimentação constante de corrente elétrica [Latching Solenoid Theory | TLX Technologies]. A válvula do tipo *Latch* é composta por um ímã permanente, uma mola e uma haste que efetuará a abertura para o fluxo de água. Quando aplicada uma corrente externa com a mesma polaridade do campo magnético já existente, estes campos se somam, movem a haste e por fim abrem

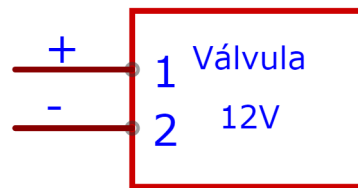
Figura 3.3: Válvula solenoide.

(a) Foto da válvula.



Fonte: Elaborado pelo autor.

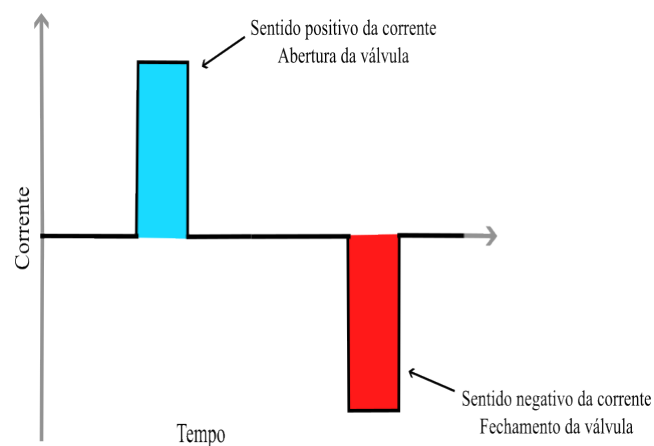
(b) Esquema representativo.



Fonte: Elaborado pelo autor.

a válvula. Ao aplicar a corrente na outra polaridade, a haste se move na outra direção, assim fechando a válvula. A figura (3.4) mostra de maneira genérica um ciclo de operação da válvula.

Figura 3.4: Ciclo de operação da válvula. O tempo de permanência da corrente aplicada são da ordem de milissegundos. Normalmente, um tempo de 500 ms funciona corretamente. Elaboração própria.

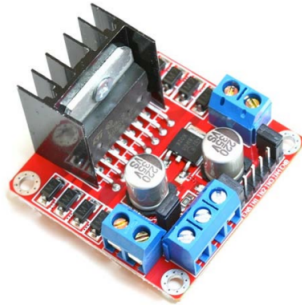


Fonte: Elaborado pelo autor.

Para a ativação da válvula foi utilizado o módulo LN298 Ponte-H Dupla. Este módulo é normalmente utilizado em projetos de robótica para ativação de motores de corrente contínua. Devido à ativação da válvula solenoide ser semelhante, o módulo foi reaproveitado neste projeto.

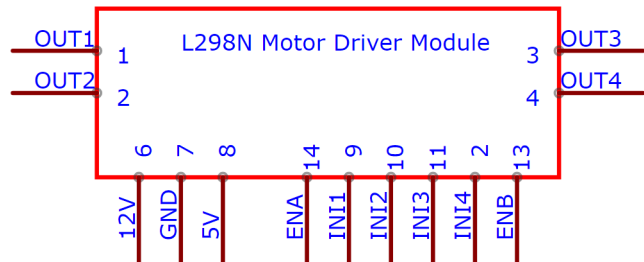
Figura 3.5: LN298N Ponte-H Dupla.

(a) Foto do Módulo L298N Ponte-H Dupla.



Fonte: [Handson Technology]

(b) Esquema representativo.



Fonte: Elaborado pelo autor

Este módulo tem como principais características [Handson Technology]:

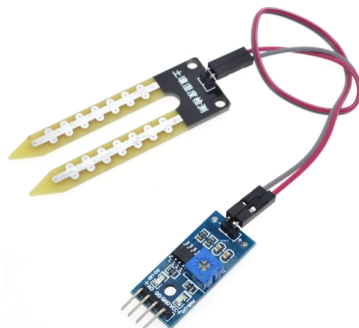
- Tensão de entrada: 3.2V ~ 40V.
- Corrente Máxima nos pinos de saída: 2 A.
- Consumo máximo de potência: 20 W (T = 75° C).
- Temperatura de operação: -25° C a +130° C.

3.1.3 Sensor de umidade

Para o monitoramento da umidade da superfície do vaso, foi utilizado o sensor de umidade de solo FC-28 exposto na figura (3.6):

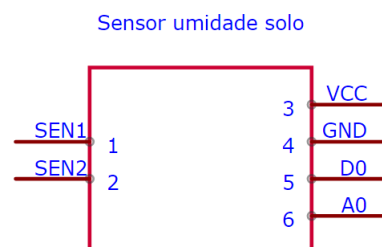
Figura 3.6: Sensor resistivo de umidade de solo.

(a) Foto sensor resistivo de umidade de solo



Fonte: [Soil moisture sensors: problems & solutions – Arduino, ESP8266, ESP32 & Raspberry Pi stuff]

(b) Esquema representativo.



Fonte: Elaborado pelo autor.

O sensor de umidade de solo resistivo é um dispositivo que permite medir a umidade do solo de forma eficiente. Ele é amplamente utilizado em aplicações agrícolas,

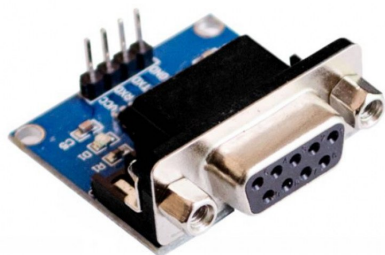
ambientais e de jardinagem, onde a medição da umidade do solo é essencial para o crescimento e saúde das plantas. O módulo do sensor opera na faixa de tensão de 3,3V a 5V e consome 15 mA de corrente, o que o torna uma opção econômica e de baixo consumo de energia. Ele possui saídas analógica e digital, além de um potenciômetro acoplado para definir o nível limite do sinal digital. Para medir a umidade do solo, o sensor é colocado no solo e uma corrente elétrica é aplicada. A presença de água no solo altera a intensidade da corrente elétrica conduzida entre as hastes do sensor. O solo seco apresenta a menor condução elétrica devido ao maior nível de porosidade em comparação com o solo úmido. Portanto, a leitura feita pelo conversor analógico-digital (ADC) de 12 bits da ESP-32 é o valor máximo de 4095 para o solo seco. Com base nessas informações, é possível determinar a frequência de irrigação. Neste projeto, serão utilizados dois sensores de umidade [Soil Moisture Sensor - Pinout, Specifications, Working].

3.1.4 Módulo DB9 Conversor Serial RS232-TTL

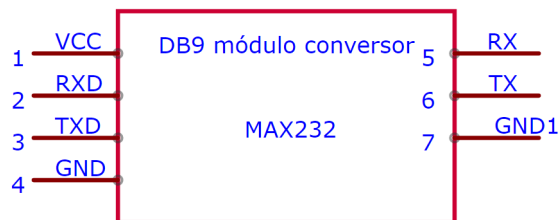
Este módulo foi utilizado para fazer a leitura dos valores da balança.

Figura 3.7: Módulo DB9 Conversor Serial RS232-TTL

(a) Foto do módulo conversor.



(b) Esquema representativo.



Fonte: Elaborado pelo autor.

Fonte: [Modulo Db9 Conversor Max3232
Rs232 Ttl Serial - Diverso Eletrônica 2023]

O módulo conta com o chip MAX3232. Este chip é responsável pela conversão do sinal RS232 para TTL (Lógica Transistor-Transistor) necessário para que a leitura possa ser feita pela ESP32. Utiliza-se somente as entradas RX/TX e da entrada RS232.

3.1.5 Demais componentes

Para a construção do circuito eletrônico, foram ainda utilizados:

- 1 x Placa universal perfurada 10x10 cm.
- 1 x Caixa de plástico 15x10x5 cm.
- 2 x Barra de 40 pinos fêmea.
- 1 x Fan 12V 0.04 A.
- 1 x Módulo Regulador Step-down LM2596.
- 4 x Conectores fêmea.
- 1 x Fonte 12V 1A.
- 1 x Conector borne fêmea P4.
- 1 x Chave on/off.
- 2 metros de cabo manga 4 vias.
- 2 x Conectores circulares 3 vias.
- 1 metro de cabo 1mm vermelho e 1 metro de cabo 1mm preto.

3.2 Programação

A programação na ESP32 foi feita em linguagem C, utilizando a IDE do Arduino. Apesar da Espressif System disponibilizar um ambiente próprio para a programação do microcontrolador, a Arduino IDE demonstra ser um ambiente mais intuitivo e apresenta suporte para as bibliotecas requisitadas. O fluxograma do código pode ser verificado na figura (3.8) e sua descrição segue da seguinte forma:

1. Configuração do programa (Início):
 - (a) Definição das variáveis temporárias e armazenamento no RTC.
 - (b) Verificação e garantia que a válvula estará fechada.
 - (c) Configuração das portas seriais RX/TX e RX2/TX2 com velocidade de transmissão de 9600 bits por segundo.
2. Etapa de coleta de dados:
 - (a) Realiza leituras das balanças e sensores de umidade.
 - (b) Se for a primeira medição, é armazenado os valores iniciais da balança referência e da balança em stress, bem como a umidade superficial dos dois vasos. Caso contrário, é gravado os valores atuais.
3. Cálculos e comparações:
 - (a) Calcula a diferença entre pesos atuais com os pesos iniciais.
 - (b) Calcula a umidade atual em relação à umidade inicial de cada vaso (em porcentagem).
4. Envio de informações:

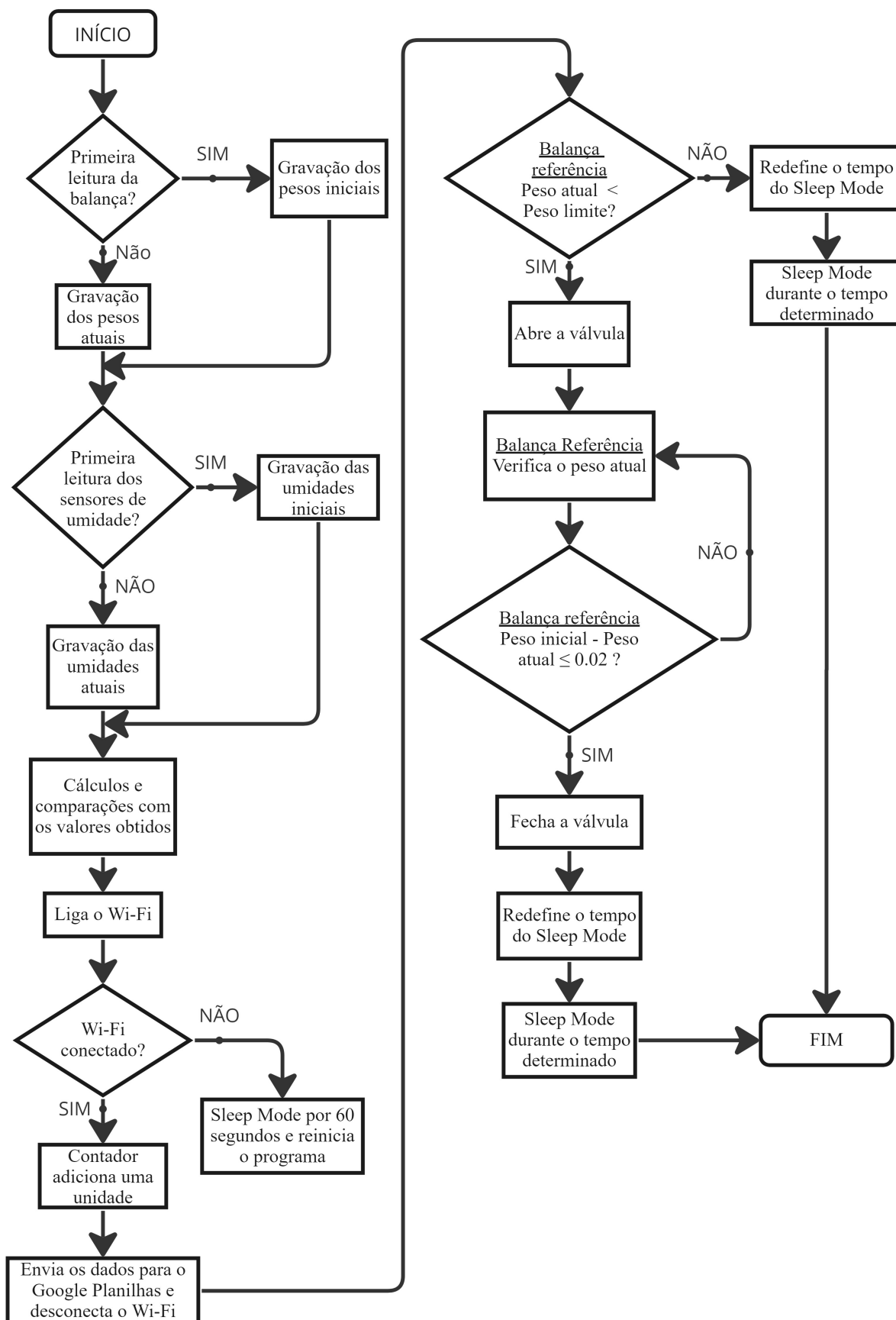
- (a) Liga o Wi-Fi e verifica se foi conectado.
- (b) Se 4a for verdadeiro, incrementa o contador de leituras e os dados coletados são enviados para a planilha dedicada no Google Planilhas. Caso contrário, é configurado o *Sleep Mode* para 60 segundos e é reiniciado o programa.
- (c) Desliga o Wi-Fi.
- (d) Verifica se o peso atual é menor do que um peso limite previamente definido. Se não, é redefinido o tempo para a próxima verificação e entra-se em *Sleep Mode*.
- (e) Se o item 4d for verdadeiro, abre-se a válvula para a irrigação do vaso de referência e é checado continuamente o peso na balança de referência.
- (f) Quando o peso de referência atingir o peso gatilho do fechamento da válvula, a válvula é fechada.

5. Preparação para o *Sleep Mode*:

- (a) Calcula-se o tempo que o programa permaneceu no modo ativo.
- (b) Redefine-se o tempo do *Sleep Mode* e o modo é ativado.

O fluxograma pode ser acessado na página seguinte.

Figura 3.8: Fluxograma do código.



Fonte: Elaborado pelo autor.

No item (4d), buscando maior precisão na retomada do peso original, foi definido que o valor do peso que agirá como gatilho para o fechamento da válvula será ligeiramente inferior ao peso inicial. De forma empírica, chegou-se que o valor de 0.02 kg abaixo do peso de referência inicial seja suficiente, ou seja, se, por exemplo, o peso inicial for de 10kg, o peso que irá produzir o fechamento da válvula será de $P_{gatilho} = 10kg - 0.02kg = 9.98kg$. Este método minimiza a interferência de gotejamentos após o fechamento da válvula.

3.2.1 Google planilhas

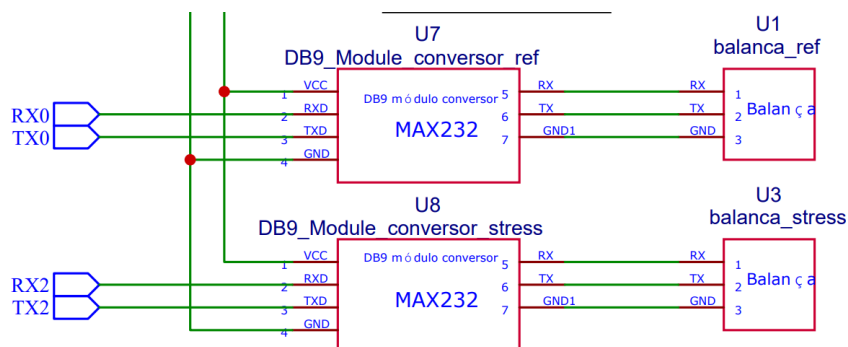
A exportação dos dados para o Google planilhas é simples. Primeiramente a ESP32 é responsável por construir uma *url* contendo os dados que serão exportados. Posteriormente, foi escrito um código no ambiente no *Apps Script* para controle dos aplicativos pertencentes ao Google *Workspace*. Este código é capaz de receber uma requisição contendo os dados exportados pela ESP32 e adicioná-los em uma planilha no Google Planilhas.

Este método foi escolhido devido sua rápida implementação e por ser a forma de análise de dados já usada pelos pesquisadores. Assim, sem a necessidade de grandes adaptações, o sistema já pode ser colocado em prática. O código feito no *App Script* está disposto no Apêndice A e por questões de segurança, a identidade da planilha e seu nome foram ocultados. No final, a planilha deverá ser preenchida com os valores dos pesos atuais dos dois vasos, a diferença destes valores do valor inicial do experimento e os valores das umidades superficiais atuais em relação à umidade inicial.

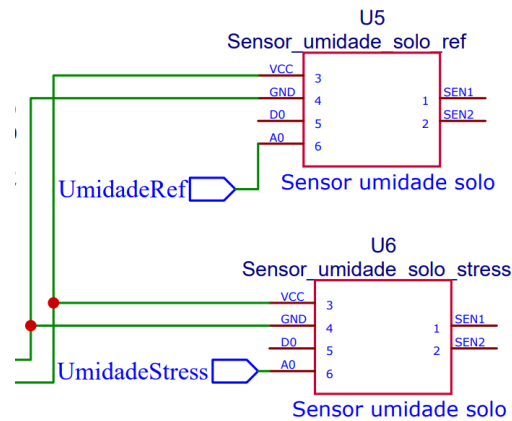
3.3 Circuito eletrônico

O circuito eletrônico completo desenvolvido pode ser consultado no Apêndice B. Abaixo é apresentado a descrição por partes do circuito.

Figura 3.9: Circuito referente às leituras dos dados das balanças.



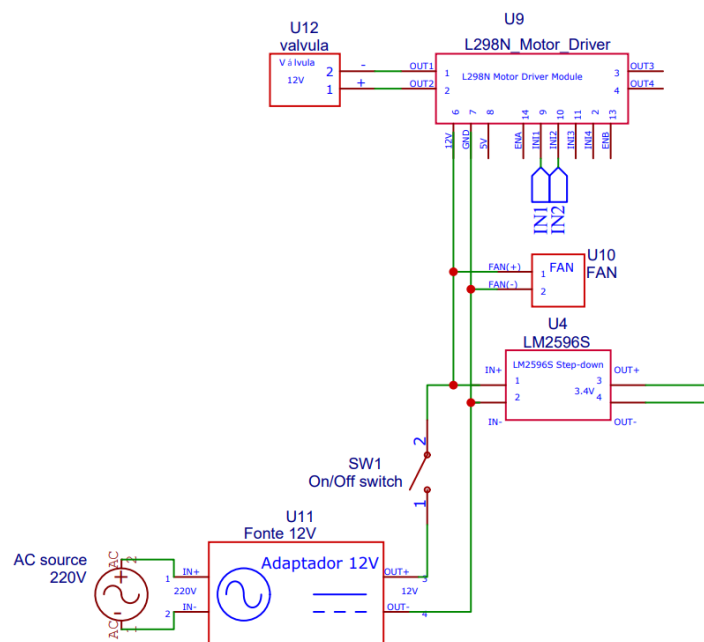
Fonte: Elaborado pelo autor.

Figura 3.10: Circuito referente à leituras dos sensores de umidade de solo.

Fonte: Elaborado pelo autor.

Na figura (3.9) estão expostos os módulos conversores DB9 RS232-TTL e suas conexões com as entradas seriais da ESP32 e com as balanças. Na figura (3.10), estão expostos as conexões nas saídas analógicas dos sensores de umidade de solo.

Para alimentação do circuito, foi utilizado uma fonte 12V 1A conectada diretamente com a rede.

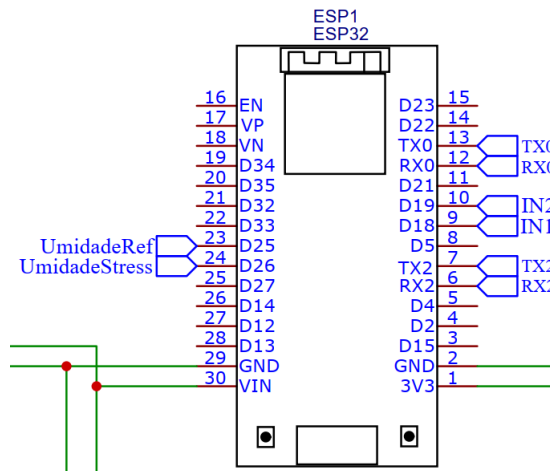
Figura 3.11: Circuito referente à alimentação do circuito.

Fonte: Elaborado pelo autor.

Na figura (3.11) é possível ver que a válvula solenoide, o módulo L298N, a FAN e o LM2596s sendo alimentados pela tensão de 12V.

Por fim, na figura (3.12) é disposto as conexões feitas na ESP32.

Figura 3.12: Conexões na ESP32.



Fonte: Elaborado pelo autor.

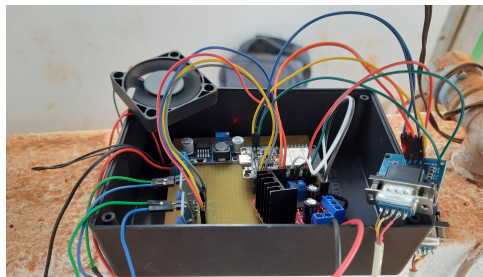
3.3.1 Protótipo

Na figura (3.13) esta disposta a foto do protótipo ainda não finalizado:

Figura 3.13: Experimento de testes.

(a) Protótipo aberto em fase de testes.

(b) Vasos utilizados sem plantas.



Fonte: Elaborado pelo autor.

Fonte: Elaborado pelo autor.

Os dois vasos foram preparados de forma semelhante. Devido o objetivo de calibração e validação do circuito, o peso inicial de cada vaso, assim como a quantidade de água inicialmente colocada não há necessidade em serem registrados. A condição principal do início do experimento é se ambos os vasos estivessem encharcados com água, alcançando a capacidade de retenção máxima do solo em questão. Foram feitos quatro experimentos, variando o tempo que o dado é coletado, a cada cinco minutos, dez minutos, trinta minutos e uma hora. Abaixo o protótipo finalizado pode ser visualizado nas figuras (3.14) e (3.15).

Figura 3.14: Visão frontal e traseira do protótipo.

(a) Visão frontal. Na imagem é possível a (b) Visão traseira. Na imagem é possível entrada da fonte de alimentação e o botão de ligar/desligar o circuito. ver as entradas dos módulos conversores DB9 RS232-TTL



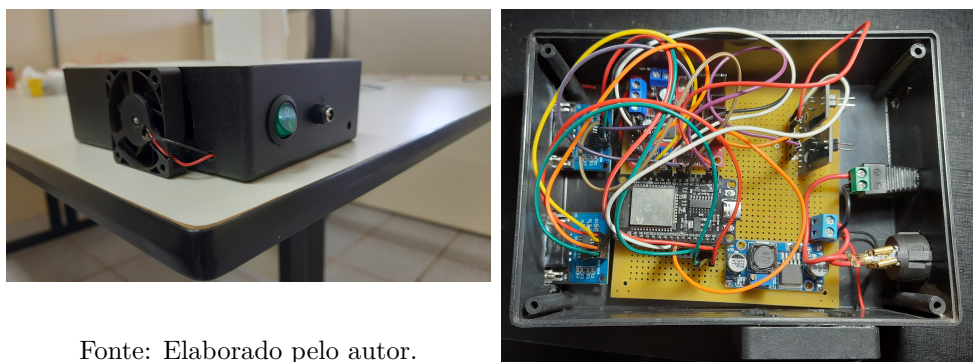
Fonte: Elaborado pelo autor.

Fonte: Elaborado pelo autor.

Figura 3.15: Visão diagonal e interna do protótipo.

(a) Visão diagonal.

(b) Visão interna.



Fonte: Elaborado pelo autor.

Fonte: Elaborado pelo autor.

Na figura (3.15a), pode-se ver a ventoinha colocada externamente, esta escolha foi feita devido à limitações de espaço da caixa.

Há diferenças na potência elétrica consumida pelo sistema total e acordo com as tarefas a serem executadas. O funcionamento do circuito pode ser separado em etapas:

- Etapa 1 - ESP32 em modo ativo¹ fazendo as leituras das balanças e sensores com o Wi-Fi desligado.
- Etapa 2 - ESP32 em modo ativo com Wi-Fi ligado, a fim de enviar as informações para a planilha

1. Chama-se modo ativo o estado de operação que todos os periféricos do microcontrolador estão ativados, também recebe o nome de modo normal.

- Etapa 3 - ESP32 em *Sleep Mode*.

Durante todas estas etapas, a FAN está ligada continuamente com potência de 0.48 W. Foi verificado a potência consumida em cada etapa e os valores podem ser conferidos na tabela (3.1) abaixo:

Tabela 3.1: Potência consumida pelo circuito elétrico.

	Consumo	
	Tensão 12V	
	Corrente (A)	Potência (W)
Etapa 1	0.185	2.22
Etapa 2	0.760	9.12
Etapa 3	0.145	1.74

Fonte: Elaborado pelo autor.

A válvula solenoide irá brevemente consumir a corrente de 750 mA para sua mudança de estado . É requisitada a abertura ou fechamento da válvula solenoide quando o circuito está na etapa 1, portanto caso seja necessário a utilização da válvula, o circuito irá ter dois picos de consumo relacionados à abertura/fechamento da válvula. Este pico será de 11.22 W. O consumo total diário irá depender de alguns fatores, como, por exemplo, se a válvula foi ativada e por quanto tempo e se houve conexão imediata do Wi-Fi.

Após a validação do circuito nestas configurações, os experimentos com plantas de algodão foram feitos.

3.4 Experimentos

Para o primeiro experimento, foi determinado o peso inicial de cada vaso, assim como a quantidade de água colocada em cada vaso. Mais uma vez, o limite de água colocada será quando ocorrer a saturação da retenção de líquidos do solo. Os parâmetros iniciais deste experimento estão expostos na tabela (3.2) abaixo.

Tabela 3.2: Parâmetros iniciais do experimento com plantas de algodão.

	Peso sem água adicional (kg)	Peso com água adicional (kg)	Água adicionada (kg)
Vaso Referência	5.32	6.11	0.79
Vaso Stress	5.25	6.09	0.84

Fonte: Elaborado pelo autor.

O sistema foi programado para verificação a cada 12 horas começando dia 17/07/2023 às 17:03. Foi definida a perda de 0.5 kg de água para a ativação da válvula.

Figura 3.16: Vasos com plantas de algodão no início do experimento 1.



Fonte: Elaborado pelo autor.

Figura 3.17: Válvula instalada para irrigação.



Fonte: Elaborado pelo autor.

Para o experimento 2, os mesmos processos de preparação tomados no experimento 1 foram aplicados. Na tabela (3.3) estão as condições iniciais.

O sistema foi programado para verificação a cada 12 horas, começando dia 09/08/2023 às 09:38:23. Foi definida a perda de 0.4 kg de água para ativação da válvula.

Tabela 3.3: Parâmetros iniciais do experimento 2 com plantas de algodão.

	Peso sem água adicional (kg)	Peso com água adicional (kg)	Água adicionada (kg)
Vaso referência	5.45	5.99	0.54
Vaso estresse	5.37	6.02	0.65

Fonte: Elaborado pelo autor.

Figura 3.18: Vasos com plantas de algodão no experimento 2.

Fonte: Elaborado pelo autor.

RESULTADOS

Foi planejado para ser executada uma fase de testes e um experimento com plantas de algodão. Definiu-se como objetivo da fase de testes identificar se o circuito estaria sendo capaz de registrar os valores dos pesos dos vasos e dos sensores de umidade e exportá-los para a planilha dedicada. Após a finalização dos testes, foi identificado que o sistema estava sendo capaz de gravar os valores dos pesos dos vasos e enviá-los para a planilha dedicada. Porém, foi identificado problemas na obtenção do valor da umidade do solo e na precisão dos valores da balança. Assim uma calibração na balança foi feita. As tabelas das fases de testes podem ser verificadas em Apêndices C, nas tabelas C.1, C.2 e C.3.

No experimento com plantas de algodão, definiu-se como objetivo registrar a perda de massa de água dos dois vasos a cada 12 horas e, a partir disto, obter e rastrear o desempenho hídrico das duas plantas. Os experimentos se deram conforme se planejava, os dados obtidos dos pesos dos vasos correspondem ao comportamento esperado, assim como a irrigação foi feita corretamente.

4.1 Experimento 1

Os resultados do experimento 1 podem ser verificados na tabela (4.1) abaixo. Nota-se um deslocamento temporal no tempo da leitura, ou seja, o sistema não entra em atividade exatamente em 12 horas, e sim de 11 ~ 12 horas. Esta diferença no tempo é explicada pela imprecisão da contagem do tempo do RTC para longos períodos. Fatores como temperatura e oscilações na alimentação da ESP32 estão como principais agentes geradores de ruídos que alteram a precisão do relógio, assim para evitar problemas de sincronização, é necessário realizar calibrações periódicas.

Tabela 4.1: Dados obtidos no experimento do algodão.

n	data	Peso referência (kg)	Peso estresse (kg)	Água perdida Referência(kg)	Água Perdida estresse (kg)
1	18/07/2023 17:03:50	6.11	6.09	0.00	0.00
2	19/07/2023 04:46:16	6.00	6.01	0.11	0.08
3	19/07/2023 16:51:46	5.67	5.87	0.44	0.22
4	20/07/2023 04:38:12	5.59	5.80	0.52	0.29
5	20/07/2023 16:42:08	5.74	5.66	0.37	0.43
6	21/07/2023 04:29:21	5.76	5.60	0.35	0.49
7	21/07/2023 16:33:26	5.66	5.50	0.45	0.59
8	22/07/2023 04:19:00	5.66	5.45	0.45	0.64
9	22/07/2023 16:22:13	5.53	5.37	0.58	0.72
10	23/07/2023 04:08:35	6.01	5.31	0.10	0.78
11	23/07/2023 16:13:57	5.80	5.23	0.31	0.86
12	24/07/2023 04:01:32	5.77	5.18	0.34	0.91
13	24/07/2023 16:06:39	5.60	5.14	0.51	0.95

Fonte: Elaborado pelo autor.

Figura 4.1: Plantas no final do experimento.

Fonte: Elaborado pelo autor.

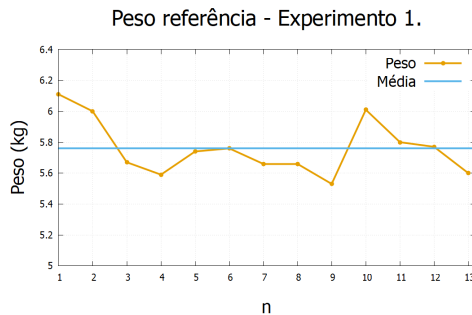
Para a aplicação que o circuito foi projetado, não há a necessidade de uma rotina de calibração frequente, pois o *offset* do tempo gerado entre as medições não irá afetar substancialmente os valores medidos. Porém, para experimentos de maior duração, é recomendado inserir ferramentas de calibração do relógio.

Analisando a tabela (4.1), nota-se que, no final das 13 medidas, o peso da água perdida pelo vaso de estresse foi de 0.91 kg, valor maior que o adicionado de água inicialmente (0.84 kg). Este valor deve-se ao fato da já existência de água no vaso de estresse antes do experimento começar. Além disso, é notado que o limite de ativação da irrigação é alcançado três vezes ($n = 4, 9$ e 13), somando a quantidade de 1,61 kg de água adicionada durante o experimento.

Na figura (4.4), é mostrado um conjunto de representações gráficas dos dados da tabela (4.1). No gráfico exposto na figura (4.2a) nota-se que o circuito com as configurações atuais é capaz de manter a balança de referência no peso médio de aproximadamente 5,8 kg. Já no vaso em estresse, percebe-se uma perda de massa seguindo um padrão linear, como pode ser verificado no gráfico mostrado na figura (4.2b). Este

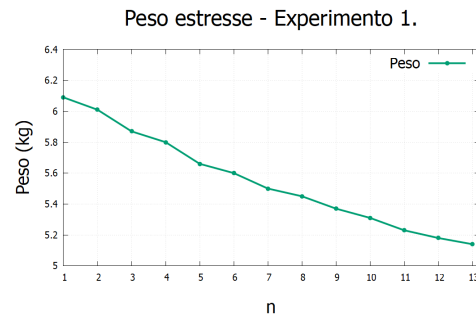
Figura 4.2: Experimento com os vasos de algodão.

(a) Progressão do vaso de referência.



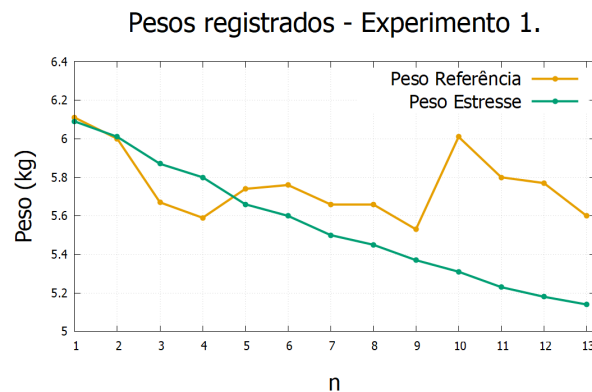
Fonte: Elaborado pelo autor.

(b) Progressão do vaso em estresse.



Fonte: Elaborado pelo autor.

(c) Progressão dos vasos.



Fonte: Elaborado pelo autor.

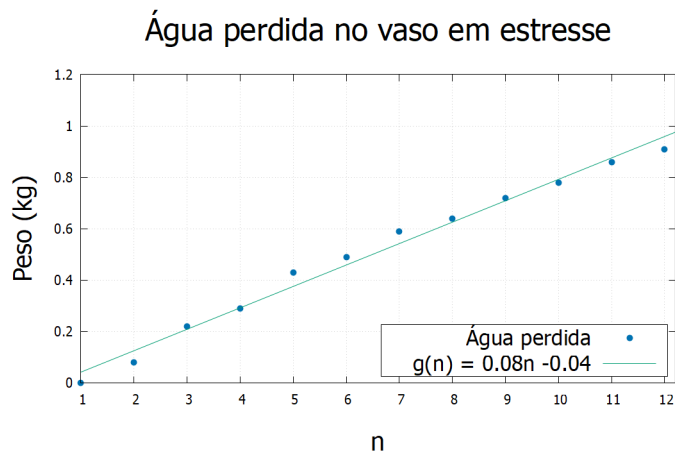
comportamento pode ser notado na regressão linear feita com a água perdida desse vaso, como pode ser visto abaixo na figura (4.3).

Para análise das taxas de perda de massa de água diária, é interessante separar entre dois períodos, nomeando-os de Período 1 e Período 2. No Período 1, há exposição reduzida das plantas ao sol, em relação ao Período 2. Este comportamento pode ser analisado nas tabelas (4.2) e (4.3) abaixo.

A influência da exposição solar fica evidente quando comparamos a perda de massa média entre os dois períodos, sendo que a maior perda ocorre no período 2. Além disso, no período 2, há uma diferença nos valores médios de água perdida entre os dois vasos. Pode ser observado que o vaso que estava sob estresse apresentou uma perda média menor do que o vaso de referência.

Essa diferença pode ser explicada pela disposição dos vasos na estufa, pois o vaso de referência sofre uma maior incidência de raios solares, justificando a maior perda média de água.

Figura 4.3: Água perdida no vaso de estresse, em relação ao peso inicial. A regressão apresenta $R^2 = 0,987$.



Fonte: Elaborado pelo autor.

Tabela 4.2: Perda de água no período 1.

Período 1			Água perdida (kg)	
Data	Referencia	Estresse		
18/07/2023 17:03:50 a 19/07/2023 04:46:16	0.11	0.08		
19/07/2023 16:51:46 a 20/07/2023 04:38:12	0.08	0.07		
20/07/2023 16:42:08 a 21/07/2023 04:29:21	0.02	0.06		
21/07/2023 16:33:26 a 22/07/2023 04:19:00	0.00	0.05		
22/07/2023 16:22:13 a 23/07/2023 04:08:35	0.10	0.06		
23/07/2023 16:13:57 a 24/07/2023 04:01:32	0.03	0.05		
Média	0.06	0.06		

Fonte: Elaborado pelo autor.

Tabela 4.3: Perda de água no período 2.

Período 2			Água perdida (kg)	
Data	Referência	Estresse		
19/07/2023 04:46:16 a 19/07/2023 16:51:46	0.33	0.14		
20/07/2023 04:38:12 a 20/07/2023 16:42:08	0.15	0.14		
21/07/2023 04:29:21 a 21/07/2023 16:33:26	0.10	0.10		
22/07/2023 04:19:00 a 22/07/2023 16:22:13	0.13	0.08		
23/07/2023 04:08:35 a 23/07/2023 16:13:57	0.20	0.08		
24/07/2023 04:01:32 a 24/07/2023 16:06:39	0.17	0.04		
Média	0.18	0.10		

Fonte: Elaborado pelo autor.

4.2 Experimento 2

Os resultados do experimento 2 podem ser verificados na tabela (4.4). O deslocamento temporal observado no experimento anterior também é verificado.

Tabela 4.4: Dados obtidos no experimento 2 do algodão.

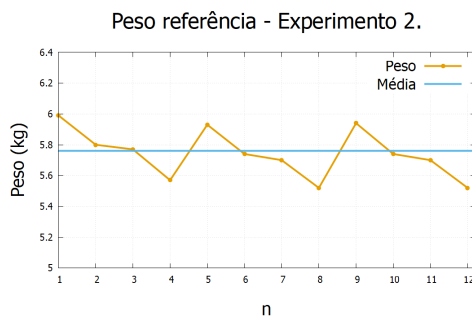
n	data	Peso referência (kg)	Peso estresse (kg)	Água perdida Referência(kg)	Água Perdida estresse (kg)	Umidade ref (%)	Umidade estresse (%)
1	09/08/2023 9:38:23	5.99	6.02	0	0	100	100
2	09/08/2023 21:40:48	5.80	5.89	0.19	0.13	27.54	100
3	10/08/2023 9:32:46	5.77	5.85	0.22	0.17	22.03	100
4	10/08/2023 21:35:08	5.57	5.75	0.42	0.27	24.13	100
5	11/08/2023 9:27:17	5.93	5.72	0.06	0.3	43.07	100
6	11/08/2023 21:29:09	5.74	5.80	0.25	0.22	14.79	100
7	12/08/2023 9:20:48	5.70	5.76	0.29	0.26	14.42	94.42
8	12/08/2023 21:21:46	5.52	5.65	0.47	0.37	18.75	95.21
9	13/08/2023 09:13:24	5.94	5.63	0.05	0.39	40.47	93.41
10	13/08/2023 21:17:47	5.74	5.53	0.25	0.49	18.44	86.86
11	14/08/2023 09:09:12	5.70	5.49	0.29	0.53	12.62	77.24
12	14/08/2023 21:10:54	5.52	5.48	0.47	0.54	14.00	78.00

Fonte: Elaborado pelo autor.

O aprimoramento alcançado pelo experimento 2 é na correção feita nos sensores de umidade de solo, pode-se então ver que o circuito está sendo capaz de fazer corretamente as leituras. Foi registrada três ativações da válvula de irrigação ($n = 4, 8$ e 12), somando 1,36 kg de água adicionada durante o experimento. O circuito foi capaz de manter o peso médio de aproximadamente 5.8 kg, resultado semelhante ao experimento anterior. Devido ao estado de debilitação da planta de algodão em estresse, foi realizada a irrigação manual no dia 11/08/2023.

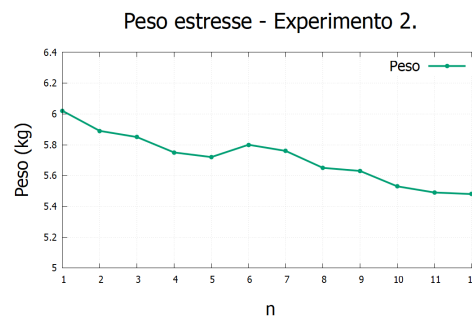
Figura 4.4: Experimento 2 com os vasos de algodão.

(a) Progressão do vaso de referência.



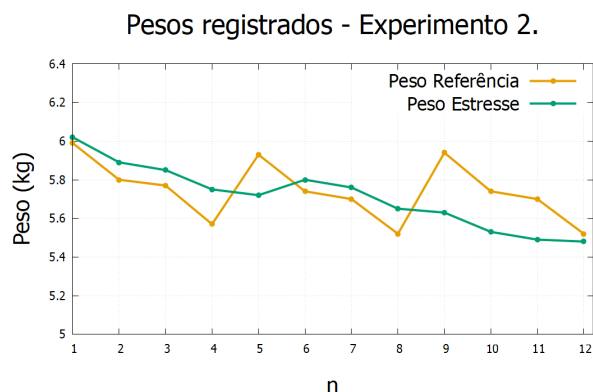
Fonte: Elaborado pelo autor.

(b) Progressão do vaso em estresse.



Fonte: Elaborado pelo autor.

Figura 4.5: Progressão dos vasos.



Fonte: Elaborado pelo autor.

Novamente, pode ser feita a análise da perda de massa de água separada em períodos de maior e menor exposição, períodos 1 e 2, respectivamente. Como pode ser visualizado nas tabelas (4.5) e (4.6) abaixo.

Tabela 4.5: Perda de água no período 1.

Período 1		
Data	Água perdida (kg)	
	Referencia	Estresse
09/08/2023 21:40:48 a 10/08/2023 09:32:46	0.03	0.04
10/08/2023 21:35:08 a 11/08/2023 09:27:17	-0.36	0.03
11/08/2023 21:29:09 a 12/08/2023 09:20:48	0.04	0.04
12/08/2023 21:21:46 a 13/08/2023 09:13:24	-0.42	0.02
13/08/2023 21:17:47 a 14/08/2023 09:09:12	0.04	0.04
Média	-0.13	0.03

Fonte: Elaborado pelo autor.

Tabela 4.6: Perda de água no período 2.

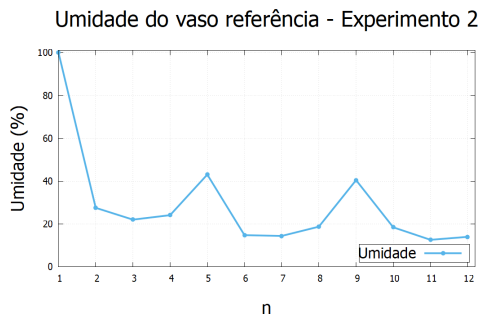
Período 2		
Data	Água perdida (kg)	
	Referencia	Estresse
09/08/2023 09:38:26 a 09/08/2023 21:40:48	0.19	0.13
10/08/2023 09:32:46 a 10/08/2023 21:35:08	0.20	0.10
11/08/2023 09:27:17 a 11/08/2023 21:29:09	0.19	0.08
12/07/2023 09:20:48 a 12/08/2023 21:21:46	0.18	0.11
13/08/2023 09:13:24 a 13/08/2023 21:17:47	0.20	0.10
Média	0.19	0.10

Fonte: Elaborado pelo autor.

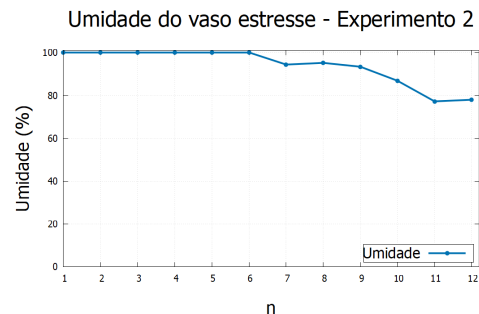
Na tabela (4.5), nota-se valores negativos para massa de água perdida. Este sinal está indicando que a água perdida durante este intervalo de tempo foi menor que o valor de água adicionada na irrigação. Vale ressaltar que a planta em estresse, em média, consumiu menos água do que anteriormente (os valores obtidos no experimento anterior podem ser verificados na tabela (4.2)). O comportamento das plantas durante os dois períodos foi o esperado, pois a planta que está sobre estresse hídrico adaptou-se para as condições de escassez e portanto o consumo de água foi diminuído. As umidades superficiais podem ser vistas nas figuras (4.6a) e (4.6b)

Figura 4.6: Umidade superficiais dos vasos.

(a) Progressão da umidade do vaso de referência. (b) Progressão da umidade do vaso em estresse.



Fonte: Elaborado pelo autor.



Fonte: Elaborado pelo autor.

Observando a figura (4.6a), é possível que, o circuito nas condições atuais, consegue manter a umidade em torno de 40% em relação à umidade inicial. Já a planta em estresse apresentou uma diminuição tardia da umidade superficial, como exposto na figura (4.6b).

CONCLUSÕES

Com o experimento das plantas de algodão, pode-se colocar o protótipo em operação em condições mais próximas da aplicação na qual fora desenvolvido. Com as calibrações feitas nas balanças. O protótipo se mostrou capaz de monitorar as balanças de referência e em estresse, além de controlar a irrigação do vaso de planta referência corretamente, retomando o peso inicial registrado no início do experimento.

A partir dos dados coletados, o pesquisador tem a capacidade de continuar os estudos com plantas em estresse hídrico, assim complementando o estudo do desempenho hídrico da planta, o desenvolvimento das raízes e o estudo da temperatura foliar.

A grande vantagem atingida por este protótipo se deu pela automação da irrigação e registro dos dados. O pesquisador, portanto, não precisará realizar visitas periódicas para registros manuais dessas medidas e para efetivar a irrigação dos vasos de referência. Com isso, conclui-se que houve um aprimoramento nas atividades exercidas pelo pesquisador.

Pode-se traçar perspectivas de melhora para o protótipo. Um fator importante é monitorar a umidade do solo em diferentes profundidades do vaso, proporcionando assim a visão mais completa da umidade em todo o vaso. Pode-se ressaltar outros pontos, como a expansão do monitoramento para mais vasos, o controle da ventoinha de resfriamento para controle da temperatura da caixa, o monitoramento de outros parâmetros que são relevantes para o experimento, como a pressão atmosférica, a temperatura da casa de cultivo, a umidade do ar, entre outros. Com estes sensores adicionais, a expansão do projeto para aplicações no campo se torna possível, pois é a partir das informações extras das condições ambientais em torno da planta, que torna-se possível a análise do desempenho hídrico das plantas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [Arduino Hardware | Arduino] ARDUINO Hardware | Arduino. <<https://www.arduino.cc/en/hardware>>. (Acessado em 03/08/2023). Citado na página 25.
- [Arora 2019] ARORA, N. K. Impact of climate change on agriculture production and its sustainable solutions. **Environmental Sustainability**, Springer, v. 2, n. 2, p. 95–96, 2019. Citado na página 16.
- [Assad et al. 2020] ASSAD, E. D. et al. Efeito das mudanças climáticas na agricultura do cerrado. In: _____. **Dinâmica Agrícola no Cerrado: Análises e Projeções**. [S.l.]: Embrapa, 2020. cap. 7, p. 213–226. Citado na página 17.
- [Buy a Raspberry Pi – Raspberry Pi] BUY a Raspberry Pi – Raspberry Pi. <<https://www.raspberrypi.com/products/>>. (Acessado em 03/08/2023). Citado na página 25.
- [Chamara et al. 2022] CHAMARA, N. et al. Ag-iot for crop and environment monitoring: Past, present, and future. **Agricultural systems**, Elsevier, v. 203, p. 103497, 2022. Citado 3 vezes nas páginas 17, 18 e 20.
- [Chong, Zhihao e Yifeng 2011] CHONG, G.; ZHIHAO, L.; YIFENG, Y. The research and implement of smart home system based on internet of things. In: IEEE. **2011 International Conference on Electronics, Communications and Control (ICECC)**. [S.l.], 2011. p. 2944–2947. Citado na página 20.
- [CNA 2023] CNA. **PIB da agropecuária cresce 21,6% no primeiro trimestre de 2023**. [S.l.], 2023. Edição 11/2023. Citado na página 16.
- [Costa et al. 2023] COSTA, N. et al. Silicon and bioagents pretreatments synergistically improve upland rice performance during water stress. **Plant Stress**, v. 7, p. 100142, 2023. ISSN 2667-064X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2667064X2300012X>>. Citado na página 17.
- [Dadhaneeya, Nema e Arora 2023] DADHANEYYA, H.; NEMA, P. K.; ARORA, V. K. Internet of things in food processing and its potential in industry 4.0 era: A review. **Trends in Food Science Technology**, 2023. ISSN 0924-2244. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0924224423002169>>. Citado na página 20.
- [Dawoud e Dawoud 2022] DAWOUD, D. S.; DAWOUD, P. **Serial communication protocols and standards**. [S.l.]: CRC Press, 2022. Citado 4 vezes nas páginas 21, 23 e 24.
- [Electronic Datasheets] ELECTRONIC DATASHEETS. **Raspberry Pi Zero e Raspberry Pi Zero W**. [S.l.]. Citado na página 26.

[ESP32 Pinout Reference] ESP32 Pinout Reference. <<https://lastminuteengineers.com/esp32-pinout-reference/>>. Acessado em 09/07/2023 às 16:58. Citado na página 28.

[ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems] ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems. <<https://www.espressif.com/en/products/socs/esp32>>. (Acessado em 03/08/2023). Citado na página 25.

[ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems] ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems. <<https://www.espressif.com/en/products/socs/esp32>>. (Acessado em 18/08/2023). Citado na página 27.

[Espressif System 2019] ESPRESSIF SYSTEM. **ESP32-WROOM-32 Datasheet**. [S.l.], 2019. Version 2.9. Citado na página 28.

[Fan et al. 2014] FAN, Y. J. et al. Iot-based smart rehabilitation system. **IEEE Transactions on Industrial Informatics**, v. 10, n. 2, p. 1568–1577, 2014. Citado na página 20.

[FAO et al. 2021] FAO et al. **Versão resumida de Panorama da segurança alimentar e nutricional na América Latina e no Caribe 2020 - Segurança alimentar e nutricional para os territórios mais atrasados**. Santiago, 2021. Disponível em: <<https://www.fao.org/documents/card/en/c/CB2330PT>>. Citado na página 16.

[Firebase | Google's Mobile and Web App Development Platform] FIREBASE | Google's Mobile and Web App Development Platform. <<https://firebase.google.com/?hl=pt>>. (Acessado em 03/08/2023). Citado na página 25.

[GEM-BL | Irrigation GEM Sol Latch Coil] GEM-BL | Irrigation GEM Sol Latch Coil. <<https://www.baccarastore.com.au/GEM-BL-Irrigation-GEM-Sol-Latch-Coil>>. (Acessado em 03/08/2023). Citado na página 30.

[Handson Technology] HANDSON TECHNOLOGY. **L298N Dual H-Bridge Motor Driver**. [S.l.]. Citado na página 32.

[Insegurança Alimentar e Nutricional — Ministério da Saúde] INSEGURANCA Alimentar e Nutricional — Ministério da Saúde. <<https://www.gov.br/saude/pt-br/assuntos/saude-brasil/glossario/inseguranca-alimentar-e-nutricional>>. (Acessado em 20/07/2023). Citado na página 16.

[KN WAAGEN] KN WAAGEN. **Manual de Usuário - KN WAAGEN**. [S.l.]. Citado na página 30.

[Latching Solenoid Theory | TLX Technologies] LATCHING Solenoid Theory | TLX Technologies. <<https://www.tlxtech.com/solenoid-theory/latching-solenoid-theory>>. Acessado em 10/07/2023 às 22:02. Citado na página 30.

[Massruhá e Leite 2017] MASSRUHÁ, S.; LEITE, M. d. A. Agro 4.0-rumo à agricultura digital. **JC na Escola Ciência, Tecnologia e Sociedade: mobilizar o conhecimento para alimentar o Brasil**, Centro Paula Souza São Paulo, v. 2, p. 28–35, 2017. Citado na página 17.

- [Modulo Db9 Conversor Max3232 Rs232 Ttl Serial - Diverso Eletrônica 2023] MODULO Db9 Conversor Max3232 Rs232 Ttl Serial - Diverso Eletrônica. 2023. <<https://www.diversoeletronica.com.br/loja/produto/modulodb9conversormax3232rs232ttlserial>>. (Acessado em 14/07/2023). Citado na página 33.
- [Nisolosi 2009] NISOLOSI, D. E. C. **Microcontrolador 8051: detalhado**. [S.l.]: Editora Érica, 2009. Citado na página 20.
- [Raspiberry Pi] RASPIBERRY PI. **Raspberry Pi 3 Model B+**. [S.l.]. Citado na página 26.
- [Raspiberry Pi 2019] RASPIBERRY PI. **Raspberry Pi 4 Model B**. [S.l.], 2019. Citado na página 26.
- [Siddique et al. 2019] SIDDIQUE, A. et al. A review on intelligent agriculture service platform with lora based wireless sensor network. **Life**, v. 100, p. 7000, 2019. Citado na página 27.
- [Soil Moisture Sensor - Pinout, Specifications, Working] SOIL Moisture Sensor - Pinout, Specifications, Working. <<https://techzeero.com/sensors-modules/soil-moisture-sensor/>>. (Acessado em 14/07/2023). Citado na página 33.
- [Soil moisture sensors: problems & solutions – Arduino, ESP8266, ESP32 & Raspberry Pi stuff] SOIL moisture sensors: problems & solutions – Arduino, ESP8266, ESP32 & Raspberry Pi stuff. <<https://arduino diy.wordpress.com/2020/08/24/soil-moisture-sensors/>>. (Acessado em 16/07/2023). Citado na página 32.
- [Tipos de Placas Arduino: Conheça as Principais - Blog Eletrogate] TIPOS de Placas Arduino: Conheça as Principais - Blog Eletrogate. <<https://blog.eletrogate.com/tipos-de-placas-arduino/>>. Acessado em 15/07/2023. Citado na página 25.
- [Xu et al. 2014] XU, L. D. et al. Object-oriented templates for automated assembly planning of complex products. **IEEE Transactions on Automation Science and Engineering**, v. 11, n. 2, p. 492–503, 2014. Citado na página 20.

APÊNDICES

CÓDIGOS DESENVOLVIDOS

Código ESP32

```
1 #include "WiFi.h"
2 #include <HTTPClient.h>
3 #include "time.h"
4 #include "esp_wpa2.h"
5
6 // CONFIGURANDO SLEEP MODE
7
8 #define uS_TO_S_FACTOR 1000000ULL // Fator de conversão de
   microseconds
9 uint64_t TIME_TO_SLEEP = 43200; // Segundos que irá
   dormir — 12 horas
10 //===== Variáveis
   =====
11
12 const char* ssid = ""; // SSID
13 const char* password = ""; // password
14 // Google script ID
15 String GOOGLE_SCRIPT_ID = "";
16
17 // SENSORES DE UMIDADE
   =====
18 const int sensor_umidade_ref = 25, sensor_umidade_stress = 26;
19 float umidade_ref, umidade_stress, sensor_analog_valv_ref,
   sensor_analog_valv_stress;
```

```
20 //
    =====
21
22 // BALANÇA
    =====

23 #define RXD2 16    //STRESS
24 #define TXD2 17
25
26 String dados_balanca_stress;
27 String dados_balanca_ref;
28 String dados_balanca_inicial;
29
30 float peso_stress , peso_ref , agua_perdida_ref ,
    agua_perdida_stress ;
31
32 // VÁLVULA
    =====

33 int pin1 = 18;    //GPIO18 (IN1)
34 int pin2 = 19;    //GPIO19 (IN2)
35
36
37 // VARIÁVEIS DE CONTROLE
    =====

38 int aux = 0;
39 int tempo_reconec;
40 unsigned long tempo_inicial , tempo_final;
41 //
    =====

42
43 //VARIÁVEIS GUARDADAS RTC
44
45 RTC_DATA_ATTR int n = 0;    //contagem de leituras
46 RTC_DATA_ATTR float peso_inicial_ref = 20;
47 RTC_DATA_ATTR float peso_inicial_stress = 20;
```

```
48
49 RTC_DATA_ATTR float umidade_inicial_ref;
50 RTC_DATA_ATTR float umidade_inicial_stress;
51 RTC_DATA_ATTR float delta_U_ref, delta_U_stress;
52
53
54 void setup() {
55     tempo_inicial = millis() / 1000;    //Momento do inicio do
        programa em segundos
56     Serial.begin(9600);
57     Serial2.begin(9600, SERIAL_8N1, RXD2, TXD2);
58     delay(500);
59
60
61     pinMode(LED_BUILTIN, OUTPUT);
62
63     pinMode(LED_BUILTIN, OUTPUT); //LED para indicar válvula
        aberta
64     pinMode(pin1, OUTPUT); // (GPIO18)
65     pinMode(pin2, OUTPUT); // (GPIO19)
66
67     digitalWrite(pin2, HIGH);
68     delay(500);
69     digitalWrite(pin2, LOW);
70     delay(500);
71     digitalWrite(LED_BUILTIN, LOW);
72
73
74     //Inicia o Monitoramento
75     leitura_balanca();
76     leitura_umidade();
77
78     initWifi();
79     ++n; // Incrementa o contador de leituras
80     envia_dados();
81
82     valvula();
83
84     // Redefinindo o tempo do Sleep Mode
```

```
85     tempo_final = millis() / 1000;
86     float tempo_exec = tempo_final - tempo_inicial;
87
88     // SLEEP MODE
89     TIME_TO_SLEEP = TIME_TO_SLEEP - tempo_exec;
90     esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * uS_TO_S_FACTOR);
91
92     // Encerrando a comunicação serial
93     Serial.flush();
94     Serial2.flush();
95     Serial.end();
96     Serial2.end();
97
98     esp_deep_sleep_start();
99 }
100
101 void loop()
102 {
103
104 }
105 void leitura_balanca() {
106     // PRIMEIRA LEITURA - OBTENÇÃO DO PESO INICIAL (REFERENCIA)
107     while (true) {
108         if (n == 0) {
109             while (true) {
110                 if (Serial.available()) {
111                     String incomingStr_ref_inicial = Serial.
112                         readStringUntil('\n');
113                     if (!(incomingStr_ref_inicial.startsWith("T", 0))) {
114                         String balanca_dados_ref_inicial =
115                             incomingStr_ref_inicial.substring(3, 9);
116                         balanca_dados_ref_inicial.trim();
117                         peso_inicial_ref = balanca_dados_ref_inicial.toFloat
118                             ();
119                         peso_ref = peso_inicial_ref;
120                         break;
121                     }
122                 }
123             }
124         }
125     }
```

```
121
122     while (true){
123         if (Serial2.available()){
124             String incomingStr_stress_inicial = Serial2.
                readStringUntil('\n');
125             if (!(incomingStr_stress_inicial.startsWith("T",0))){
126                 String balanca_dados_stress_inicial =
                    incomingStr_stress_inicial.substring(3,9);
127                 balanca_dados_stress_inicial.trim();
128                 peso_inicial_stress = balanca_dados_stress_inicial.
                    toFloat();
129                 peso_stress = peso_inicial_stress;
130                 break;
131             }
132         }
133     }
134
135     agua_perdida_ref = 0;
136     agua_perdida_stress = 0;
137 }
138
139 else{
140     while (true){
141         if (Serial2.available()){
142             String incomingStr_stress = Serial2.readStringUntil('\
                n');
143             if (!(incomingStr_stress.startsWith("T",0))){
144                 String balanca_dados_stress = incomingStr_stress.
                    substring(3,9);
145                 balanca_dados_stress.trim();
146                 peso_stress = balanca_dados_stress.toFloat();
147                 break;
148             }
149         }
150     }
151
152     while (true){
153         if (Serial.available()){
154             String incomingStr_ref = Serial.readStringUntil('\n');
```

```
155         if (!(incomingStr_ref.startsWith("T",0))){
156             String balanca_dados_ref = incomingStr_ref.substring
                (3,9);
157             balanca_dados_ref.trim();
158             peso_ref = balanca_dados_ref.toFloat();
159             break;
160         }
161     }
162 }
163
164 //Diferença de peso —> (VASO STRESS INICIAL – VASO
        STRESS)
165 agua_perdida_stress = peso_inicial_stress – peso_stress;
        // QUANTIDADE DE AGUA PERDIDA NA VASO DE STRESS
166
167 // tirando a diferença de peso —> (VASO INICIAL – VASO
        REF)
168 agua_perdida_ref = peso_inicial_ref – peso_ref; //
        QUANTIDADE DE ÁGUA PERDIDA NA VASO REFERENCIA
169 }
170
171 if (peso_inicial_stress != 0 && peso_stress != 0 &&
        peso_inicial_ref !=0 && peso_ref !=0 && peso_ref < 100 &&
        peso_stress < 100){
172     break;
173 }
174 }
175
176 }
177
178 void leitura_umidade(){
179
180     if (n == 0){ //Primeira medida
181         delay(1000);
182         umidade_inicial_ref = analogRead(sensor_umidade_ref);
183         umidade_inicial_stress = analogRead(sensor_umidade_stress);
184
185         delta_U_ref = (4095 – umidade_inicial_ref);
186         delta_U_stress = (4095 – umidade_inicial_stress);
```

```
187     umidade_ref = 100;
188     umidade_stress = 100;
189 }
190 else{
191     // LEITURA DOS VASOS
192     sensor_analog_valv_ref = analogRead(sensor_umidade_ref);
193     sensor_analog_valv_stress = analogRead(sensor_umidade_stress
194 );
195     if (sensor_analog_valv_ref < umidade_inicial_ref){
196         sensor_analog_valv_ref = umidade_inicial_ref;
197     }
198
199     if (sensor_analog_valv_stress < umidade_inicial_stress){
200         sensor_analog_valv_stress = umidade_inicial_stress;
201     }
202
203     float umidade_ref_p = ((sensor_analog_valv_ref -
204         umidade_inicial_ref))/(delta_U_ref);
205     float umidade_stress_p = ((sensor_analog_valv_stress -
206         umidade_inicial_stress))/(delta_U_stress);
207     umidade_ref = (1 - umidade_ref_p)*100;
208     umidade_stress = (1 - umidade_stress_p)*100;
209 }
210 void valvula () {
211
212     if (agua_perdida_ref >= 0.4) {
213         digitalWrite(pin1, HIGH); //Abre a válvula
214         delay(500);
215         digitalWrite(pin1, LOW);
216         aux = 1;
217     }
218
219     if (aux == 1){
220         unsigned long tempo_inicial = millis();
221         while (true){
222             digitalWrite(LED_BUILTIN, HIGH);
```

```
223     while (true){
224         if (Serial.available()){
225             String incomingStr_ref = Serial.readStringUntil('\n');
226             if (!(incomingStr_ref.startsWith("T",0))){
227                 String balanca_dados_ref = incomingStr_ref.substring
228                     (3,9);
229                 balanca_dados_ref.trim();
230                 peso_ref = balanca_dados_ref.toFloat();
231                 break;
232             }
233         }
234
235         float limite = peso_inicial_ref - peso_ref;
236         if (limite <= 0.02){
237             digitalWrite(pin2,HIGH);
238             delay(500);
239             digitalWrite(pin2,LOW);
240
241             //Serial.println("Válvula fechada!");
242             digitalWrite(LED_BUILTIN, LOW);
243             break;
244         }
245     }
246     aux = 0;
247 }
248
249 }
250
251 void initWifi() {
252     WiFi.disconnect(true);
253     WiFi.begin(ssid,password);
254     WiFi.mode(WIFI_STA);
255     digitalWrite(LED_BUILTIN, HIGH);
256
257
258     int timeout = 100;
259     while(WiFi.status() != WL_CONNECTED && (--timeout > 0)){
260         Serial.println(".");
```

```
261     delay(250);
262 }
263 if(WiFi.status() != WL_CONNECTED){
264     Serial.println("Wi-fi não conectado, entrando em sleep mode
        por 60 segundos");
265     //Limpaando a comunicação serial.
266     Serial.flush();
267     Serial2.flush();
268     Serial.end();
269     Serial2.end();
270
271     tempo_reconec = 60;
272     esp_sleep_enable_timer_wakeup(tempo_reconec * uS_TO_S_FACTOR
        );
273     esp_deep_sleep_start();
274 }
275 else{
276     digitalWrite(LED_BUILTIN, LOW);
277 }
278 }
279
280 void envia_dados(){
281     if (WiFi.status() == WL_CONNECTED)
282     {
283         String urlFinal;
284         // EXPORTANDO PARA PLANILHA (GOOGLE SHEETS)
285
286         urlFinal = "https://script.google.com/macros/s/"+
            GOOGLE_SCRIPT_ID+"/exec?"+"n="+String(n)+"&peso_ref="+
            String(peso_ref)+"&peso_stress="+String(peso_stress)+"&
            agua_perdida_ref="+String(agua_perdida_ref)+"&
            agua_perdida_stress="+String(agua_perdida_stress)+"&
            umidade_ref="+String(umidade_ref)+"&umidade_stress="+
            String(umidade_stress);
287
288         HTTPClient http;
289         http.begin(urlFinal.c_str());
290         http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
291         int httpCode = http.GET();
```

```
292
293     String payload;
294     if (httpCode > 0)
295     {
296         payload = http.getString();
297     }
298     //

299     http.end();
300
301     while(true){
302         WiFi.disconnect(true);
303         delay(1000);
304         if (WiFi.status() != WL_CONNECTED){
305             break;
306         }
307     }
308 }
309 }
```

Código Apps Script

```
1 var sheet_id = " ";
2 var sheet_name = " ";
3
4 function doGet(e){
5 var ss = SpreadsheetApp.openById(sheet_id);
6 var sheet = ss.getSheetByName(sheet_name)
7
8 var n = Number(e.parameter.n);
9 var peso_ref = Number(e.parameter.peso_ref);
10 var peso_stress = Number(e.parameter.peso_stress);
11 var agua_perdida_ref = Number(e.parameter.agua_perdida_ref);
12 var agua_perdida_stress = Number(e.parameter.agua_perdida_stress
    );
13 var porc_umidade_ref = Number(e.parameter.porc_umidade_ref);
14 var porc_umidade_stress = Number(e.parameter.porc_umidade_stress
    );
15
16 var data = new Date().toLocaleString('br',{timeZone: 'America/
    Sao_Paulo'});
17 sheet.appendRow([n, data, peso_ref, peso_stress, agua_perdida_ref,
    agua_perdida_stress, porc_umidade_ref, porc_umidade_stress]);
18 }
```

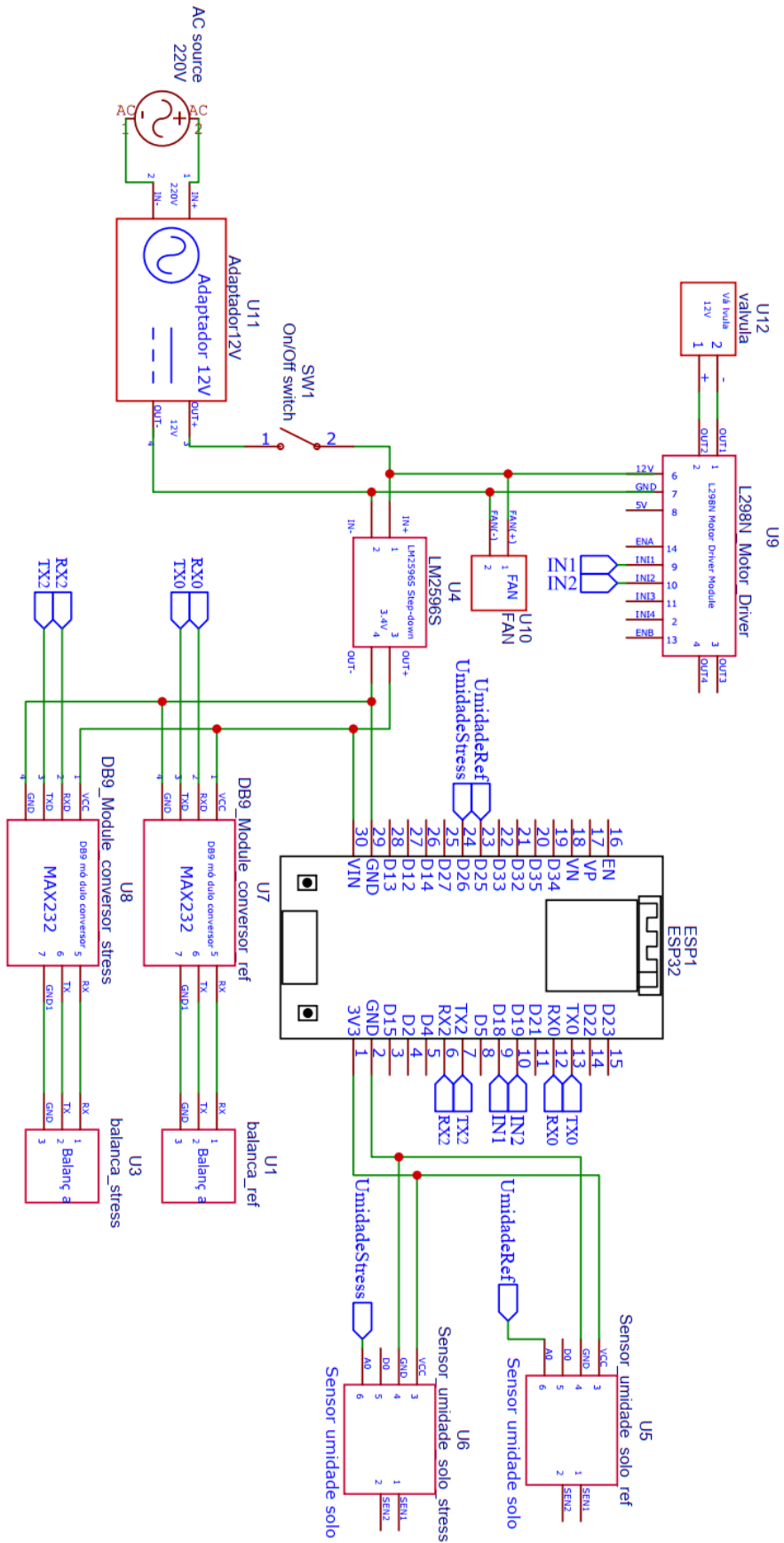
APÊNDICE

B

ESQUEMA ELÉTRICO DO CIRCUITO

O circuito elétrico desenvolvido está exposto na figura (B.1) na página seguinte.

Figura B.1: Esquema elétrico do circuito.



Fonte: Elaboração própria.

TABELAS

Tabela C.1: Tabela com dados obtidos a cada 5 minutos.

n	data	Peso referência (kg)	Peso stress (kg)	Água perdida Referência(L)	Água Perdida Stress	Umidade ref (%)	Umidade stress (%)
1	12/07/2023 15:37:38	12.91	12.89	0	0	100	100
2	12/07/2023 15:42:41	12.91	12.89	0	0	100	100
3	12/07/2023 15:47:45	12.91	12.89	0	0	100	100
4	12/07/2023 15:52:50	12.91	12.89	0	0	100.00	100
5	12/07/2023 15:57:55	12.91	12.89	0	0	100.00	100
7	12/07/2023 16:08:17	12.91	12.89	0	0	100	100
8	12/07/2023 16:13:22	12.91	12.89	0	0	100	100
9	12/07/2023 16:18:28	12.91	12.89	0	0	100	100
10	12/07/2023 16:23:34	12.91	12.89	0	0	100	100
11	12/07/2023 16:28:41	12.91	12.89	0	0	100	100
12	12/07/2023 16:33:47	12.91	12.89	0	0	100	100
13	12/07/2023 16:38:52	12.91	12.89	0	0	100	100
14	12/07/2023 16:43:56	12.91	12.88	0	0.01	100	100
15	12/07/2023 16:49:02	12.9	12.88	0.01	0.01	100	100
16	12/07/2023 16:54:08	12.9	12.88	0.01	0.01	100	100
17	12/07/2023 16:59:14	12.9	12.88	0.01	0.01	100	100
18	12/07/2023 17:04:20	12.9	12.88	0.01	0.01	100	100
19	12/07/2023 17:09:23	12.9	12.88	0.01	0.01	100	100
20	12/07/2023 17:14:27	12.9	12.88	0.01	0.01	100	100
21	12/07/2023 17:19:32	12.9	12.88	0.01	0.01	100	100
22	12/07/2023 17:24:36	12.9	12.88	0.01	0.01	100	100
23	12/07/2023 17:29:40	12.9	12.88	0.01	0.01	100	100
24	12/07/2023 17:34:44	12.9	12.88	0.01	0.01	100	100
25	12/07/2023 17:39:49	12.9	12.88	0.01	0.01	100	100
26	12/07/2023 17:44:53	12.9	12.88	0.01	0.01	100	100
27	12/07/2023 17:49:57	12.89	12.88	0.02	0.01	100	100
28	12/07/2023 17:55:01	12.91	12.88	0	0.01	100	100
29	12/07/2023 18:00:08	12.91	12.88	0	0.01	100	100
30	12/07/2023 18:05:12	12.92	12.88	-0.01	0.01	100	100
31	12/07/2023 18:10:21	12.92	12.88	-0.01	0.01	100	100
32	12/07/2023 18:15:26	12.9	12.88	0.01	0.01	100	100
33	12/07/2023 18:20:31	12.97	12.88	-0.06	0.01	100	100
34	12/07/2023 18:25:36	13	12.88	-0.09	0.01	100	100
35	12/07/2023 18:30:40	13.02	12.88	-0.11	0.01	100	100
36	12/07/2023 18:35:45	13.03	12.88	-0.12	0.01	100	100
37	12/07/2023 18:40:49	13.04	12.88	-0.13	0.01	100	100
38	12/07/2023 18:45:53	13.04	12.88	-0.13	0.01	100	100
39	12/07/2023 18:50:57	13.05	12.88	-0.14	0.01	100	100
40	12/07/2023 18:56:03	13.05	12.88	-0.14	0.01	100	100
41	12/07/2023 19:06:13	13.05	12.88	-0.14	0.01	100	100
42	12/07/2023 19:11:19	13.05	12.87	-0.14	0.02	100	100
43	12/07/2023 19:16:26	13.05	12.87	-0.14	0.02	100	100
44	12/07/2023 19:21:30	13.05	12.87	-0.14	0.02	100	100
45	12/07/2023 19:26:33	13.05	12.87	-0.14	0.02	100	100
46	12/07/2023 19:31:38	13.05	12.87	-0.14	0.02	100	100
47	12/07/2023 19:36:42	13.04	12.87	-0.13	0.02	100	100
48	12/07/2023 19:41:46	13.04	12.87	-0.13	0.02	100	100
49	12/07/2023 19:46:50	13.04	12.87	-0.13	0.02	100	100
50	12/07/2023 19:51:54	13.03	12.87	-0.12	0.02	100	100
51	12/07/2023 19:56:59	13.03	12.87	-0.12	0.02	100	100
52	12/07/2023 20:07:11	13.02	12.87	-0.11	0.02	100	100
53	12/07/2023 20:12:18	13.02	12.87	-0.11	0.02	100	100
54	12/07/2023 20:17:22	13.02	12.87	-0.11	0.02	100	100
56	12/07/2023 20:27:30	13.02	12.87	-0.11	0.03	100	100
57	12/07/2023 20:32:34	13.05	12.87	-0.14	0.02	100	100
58	12/07/2023 20:37:38	13.05	12.86	-0.14	0.03	100	100
59	12/07/2023 20:42:42	13.04	12.85	-0.13	0.04	100	100
60	12/07/2023 20:47:46	13.02	12.84	-0.11	0.05	100	100
61	12/07/2023 20:52:51	12.98	12.84	-0.07	0.05	100	100
62	12/07/2023 20:57:56	12.97	12.84	-0.06	0.05	100	100
63	12/07/2023 21:03:00	12.93	12.8	-0.02	0.09	100	100
64	12/07/2023 21:08:04	12.92	12.72	-0.01	0.17	100	100
65	12/07/2023 21:13:09	12.91	12.8	0	0.09	100	100
66	12/07/2023 21:18:14	12.88	12.06	0.03	0.83	100	100
67	12/07/2023 21:23:18	12.88	12.62	0.03	0.27	100	100
68	12/07/2023 21:28:23	12.86	12.29	0.05	0.6	100	100
69	12/07/2023 21:38:48	12.87	11.57	0.04	1.32	100	100
70	12/07/2023 21:43:53	12.87	12.31	0.04	0.58	100	100
71	12/07/2023 21:49:00	12.85	12.33	0.06	0.56	100	100
72	12/07/2023 21:54:05	12.83	12.24	0.08	0.65	100	100
73	12/07/2023 21:59:09	12.61	12.56	0.3	0.33	100	100

Fonte: Elaboração própria.

Tabela C.2: Tabela com dados obtidos a cada 10 minutos.

n	data	Peso referência (kg)	Peso stress (kg)	Água perdida Referência (L)	Água Perdida Stress (L)	Umidade ref (%)	Umidade stress (%)
1	13/07/2023 10:48:18	13.8	12.9	0.0	0.0	100.0	100.0
2	13/07/2023 10:58:21	13.8	12.8	0.0	0.0	100.0	100.0
3	13/07/2023 11:08:26	13.8	12.8	0.0	0.0	100.0	100.0
4	13/07/2023 11:18:30	13.8	12.8	0.0	0.0	100.0	100.0
5	13/07/2023 11:28:35	13.7	12.8	0.0	0.0	100.0	100.0
6	13/07/2023 11:38:39	13.7	12.8	0.1	0.0	100.0	100.0
7	13/07/2023 11:48:43	13.7	12.8	0.1	0.0	100.0	100.0
8	13/07/2023 11:58:49	13.7	12.8	0.1	0.0	100.0	100.0
9	13/07/2023 12:08:52	13.7	12.8	0.1	0.0	100.0	100.0
10	13/07/2023 12:18:59	13.7	12.8	0.1	0.0	100.0	100.0
12	13/07/2023 12:39:10	13.7	12.8	0.1	0.0	100.0	100.0
13	13/07/2023 12:49:14	13.7	12.8	0.1	0.0	100.0	100.0
14	13/07/2023 12:59:17	13.7	12.8	0.1	0.0	100.0	100.0
15	13/07/2023 13:09:26	13.7	12.8	0.1	0.0	100.0	100.0
16	13/07/2023 13:19:30	13.7	12.8	0.1	0.0	100.0	100.0
17	13/07/2023 13:29:32	13.0	12.8	0.8	0.0	100.0	100.0
18	13/07/2023 13:39:38	13.8	12.8	0.0	0.0	100.0	100.0
19	13/07/2023 13:49:41	13.8	12.8	0.0	0.0	100.0	100.0
20	13/07/2023 13:59:45	13.8	12.8	0.0	0.0	100.0	100.0
21	13/07/2023 14:09:48	13.8	12.8	0.0	0.0	100.0	100.0
22	13/07/2023 14:19:51	13.8	13.1	0.0	-0.3	100.0	100.0
23	13/07/2023 14:29:55	13.8	12.8	0.0	0.0	100.0	100.0
24	13/07/2023 14:39:59	13.7	12.5	0.0	0.4	100.0	100.0
25	13/07/2023 14:50:03	13.7	12.9	0.0	-0.1	100.0	100.0
26	13/07/2023 15:00:07	13.7	12.8	0.0	0.1	100.0	100.0
27	13/07/2023 15:10:10	13.7	12.8	0.0	0.1	100.0	100.0
28	13/07/2023 15:20:13	13.7	12.8	0.1	0.1	100.0	100.0
29	13/07/2023 15:30:19	13.7	12.8	0.1	0.1	100.0	100.0
30	13/07/2023 15:40:23	13.7	12.8	0.1	0.1	100.0	100.0

Fonte: Elaboração própria.

Tabela C.3: Tabela com dados obtidos a cada 30 minutos.

n	data	Peso referência (kg)	Peso stress (kg)	Água perdida Referência(L)	Água Perdida Stress (L)	Umidade ref (%)	Umidade stress (%)
1	13/07/2023 16:04:21	13.0	12.8	0.0	0.0	100.0	100.0
2	13/07/2023 16:34:07	13.0	12.8	0.0	0.0	100.0	100.0
3	13/07/2023 17:04:03	13.0	12.8	0.0	0.0	100.0	100.0
4	13/07/2023 17:33:56	13.0	12.7	0.0	0.1	100.0	100.0
5	13/07/2023 18:03:56	13.0	12.6	0.0	0.2	100.0	100.0
6	13/07/2023 18:33:51	13.0	12.8	0.0	0.0	100.0	100.0
7	13/07/2023 19:03:47	13.0	12.8	0.0	0.0	100.0	100.0
8	13/07/2023 19:33:44	13.0	75.4	0.1	-62.6	100.0	100.0
9	13/07/2023 20:03:45	12.98	12.78	0.06	0.02	100	100
10	13/07/2023 20:33:41	12.98	12.44	0.06	0.36	100	100
11	13/07/2023 21:03:36	13.0	12.4	0.1	0.4	100.0	100.0
12	13/07/2023 21:33:32	13.0	12.7	0.1	0.1	100.0	100.0
13	13/07/2023 22:03:27	13.0	91.2	0.1	-78.4	100.0	100.0
14	13/07/2023 22:33:26	13.0	12.9	0.1	-0.1	100.0	100.0
15	13/07/2023 23:03:20	13.0	35.0	0.1	-22.2	100.0	100.0
16	13/07/2023 23:45:04	13.0	99.8	0.1	-87.0	100.0	100.0
17	14/07/2023 00:15:00	13.0	12.8	0.1	0.0	100.0	100.0
18	14/07/2023 00:44:56	13.0	12.8	0.1	0.0	100.0	100.0
19	14/07/2023 01:14:51	13.0	12.8	0.1	0.1	100.0	100.0
20	14/07/2023 01:44:46	13.0	12.8	0.1	0.0	100.0	100.0
21	14/07/2023 02:14:43	13.0	12.7	0.1	0.1	100.0	100.0
22	14/07/2023 02:44:39	13.0	12.7	0.1	0.1	100.0	100.0
23	14/07/2023 03:14:36	13.0	12.7	0.1	0.1	100.0	100.0
24	14/07/2023 03:44:32	13.0	12.6	0.1	0.2	100.0	100.0
25	14/07/2023 04:14:27	13.0	12.8	0.1	0.0	100.0	100.0
26	14/07/2023 04:44:24	13.0	12.7	0.1	0.1	100.0	100.0
27	14/07/2023 05:14:20	13.0	12.8	0.1	0.0	100.0	100.0
28	14/07/2023 05:44:18	13.0	12.7	0.1	0.1	100.0	100.0
29	14/07/2023 06:14:14	13.0	12.7	0.1	0.1	100.0	100.0
30	14/07/2023 06:44:09	13.0	12.7	0.1	0.1	100.0	100.0
31	14/07/2023 07:14:09	13.0	12.7	0.1	0.1	100.0	100.0
32	14/07/2023 07:44:06	13.0	12.7	0.1	0.1	100.0	100.0
33	14/07/2023 08:14:02	13.0	12.8	0.1	0.1	100.0	100.0
34	14/07/2023 08:43:59	13.0	12.8	0.1	0.1	100.0	100.0
35	14/07/2023 09:13:55	12.9	12.8	0.1	0.1	100.0	100.0
36	14/07/2023 09:43:55	12.9	12.8	0.1	0.1	100.0	100.0
37	14/07/2023 10:13:53	13.4	12.8	-0.3	0.0	100.0	100.0
38	14/07/2023 10:43:51	13.3	9.1	-0.3	3.7	100.0	100.0
39	14/07/2023 11:13:48	13.3	12.8	-0.3	0.0	100.0	100.0
40	14/07/2023 11:43:43	13.3	12.8	-0.3	0.0	100.0	100.0
41	14/07/2023 12:13:40	13.3	12.8	-0.3	0.0	100.0	100.0

Fonte: Elaboração própria.