

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

Bruno G. F. Sampaio

**DESENVOLVIMENTO DE UMA PLATAFORMA DE
HARDWARE ABERTO DE BAIXO CUSTO PARA
SISTEMAS DE RASTREAMENTO SOLAR PARA
PEQUENAS USINAS DE GERAÇÃO DE ENERGIA
FOTOVOLTAICA UTILIZANDO REDE MODBUS
TCP/IP**

Santa Maria, RS
2024

Bruno G. F. Sampaio

**DESENVOLVIMENTO DE UMA PLATAFORMA DE HARDWARE ABERTO DE
BAIXO CUSTO PARA SISTEMAS DE RASTREAMENTO SOLAR PARA
PEQUENAS USINAS DE GERAÇÃO DE ENERGIA FOTOVOLTAICA
UTILIZANDO REDE MODBUS TCP/IP**

Trabalho de conclusão de curso, apresentado
ao Curso de Engenharia de Controle e
Automação, da Universidade Federal de Santa
Maria (UFSM, RS).

Orientador: Prof. Dr. Frederico Meninen Schaf

Santa Maria, RS
2024

AGRADECIMENTO

Àqueles que vieram antes de mim e, através de suas mãos, me ensinaram a caminhar.

"Sometimes you've got to bleed to know that you're alive and have a soul."

(Twenty One Pilots, Tear in My Heart)

RESUMO

DESENVOLVIMENTO DE UMA PLATAFORMA DE HARDWARE ABERTO PARA SISTEMAS DE RASTREAMENTO SOLAR NA GERAÇÃO DE ENERGIA FOTOVOLTAICA UTILIZANDO REDE Modbus TCP/IP

AUTOR: BRUNO G. F. SAMPAIO
ORIENTADOR: FREDERICO MENINE SCHAF

O aumento significativo no uso de sistemas de rastreamento solar na geração de energia tem impulsionado uma escolha popular devido ao seu potencial para ampliar a eficiência e o rendimento das instalações solares. Contudo, a viabilidade econômica desses sistemas ainda permanece um desafio, uma vez que requerem não apenas uma quantidade substancial de componentes elétricos, mas também diversos circuitos eletrônicos auxiliares. Esses aspectos demandam uma engenharia de hardware cuidadosa para manter a estabilidade e o funcionamento eficaz do sistema. Neste contexto, busca-se incentivar a adoção desses métodos de geração solar por meio do desenvolvimento de uma plataforma de hardware aberta e replicável. A proposta visa impulsionar a criação de novos sistemas de rastreamento solar para painéis fotovoltaicos, com a finalidade de promover alternativas economicamente viáveis. A criação de novos hardwares a preços acessíveis para esse tipo de sistema visa fomentar pesquisas no campo da geração solar, visando aprimorar ainda mais a eficiência da energia solar no Brasil e tornar essa tecnologia acessível aos consumidores em geral.

Palavras-chave: Energia solar; Rastreamento solar; *Hardware* aberto; Rede de comunicação Modbus TCP/IP.

ABSTRACT

DESENVOLVIMENTO DE UMA PLATAFORMA DE HARDWARE ABERTO PARA SISTEMAS DE RASTREAMENTO SOLAR NA GERAÇÃO DE ENERGIA FOTOVOLTAICA UTILIZANDO REDE Modbus TCP/IP

AUTHOR: BRUNO G. F. SAMPAIO

ADVISOR: FREDERICO MENINE SCHAF

The significant increase in the use of solar tracking systems in power generation has driven a popular choice due to their potential to increase the efficiency and yield of solar installations. However, the economic viability of these systems still remains a challenge, as they require not only a substantial amount of electrical components, but also several auxiliary electronic circuits. These aspects require careful hardware engineering to maintain system stability and effective operation. In this context, the adoption of these solar generation methods is encouraged through the development of an open and replicable hardware platform. The proposal aims to boost the creation of new solar tracking systems for photovoltaic panels, with the aim of promoting economically viable alternatives. The creation of new hardware at affordable prices for this type of system aims to encourage research in the field of solar generation, aiming to further improve the efficiency of solar energy in Brazil and make this technology accessible to consumers in general.

Keywords: Solar energy; Solar tracking; Open hardware; Modbus TCP/IP communication network.

LISTA DE FIGURAS

Figura 1 : Mapa do potencial de geração solar fotovoltaica em termos do rendimento energético anual para todo o Brasil (medido em kWh/kWp.ano)	13
Figura 2 - Ângulos de Azimute e Zênite referentes a um seguidor solar	16
Figura 3 - Módulo de controle utilizado por GONÇALVES FILHO (2004).....	18
Figura 4 - Sensor ISS-A60 utilizado por ANGULO-CALDERÓN (2022).	18
Figura 5 - Módulo de controle utilizado por BARBOSA (2009).	19
Figura 6 - Esquema lógico da ponte H L298N.	22
Figura 7 - Configuração paralelo do CI L298N recomendado.	23
Figura 8 - Exemplos de motores DC escovados de baixa potência (A) Motor My4835 (B) 5840-31zy	24
Figura 9 - Encoder incremental ótico com disco perfurado.	25
Figura 10 - Exemplos de tacômetros (a) ópticos (b) de efeito <i>hall</i> (c) coroa dentada	26
Figura 11 - Encoders absolutos do tipo resistivo (a) construção (b) circuito equivalente.	27
Figura 12 - Encoders ópticos com discos codificados com codificação (a) binária (b) de <i>Gray</i>	28
Figura 13 - Arranjo do sensor magnético AS5600.	28
Figura 14 - configuração dos pinos de um micro cartão SD.	30
Figura 15 - Configuração de blocos do microcontrolador ESP32-WROOM32E.	31
Figura 16 - Módulo ESP32-WROOM32E.	32
Figura 17 : Fluxograma de <i>Hardware</i>	34
Figura 18 - Simulação do circuito de proteção contra inversão de polaridade utilizando <i>LTS spice</i>	36
Figura 19 - Resultado da simulação do circuito de proteção contra inversão de polaridade.	37
Figura 20 - Esquemático do circuito de proteção contra inversão de polaridade utilizado.	38
Figura 21 - Esquema de ligação aconselhado para o circuito Buck.	39
Figura 22 - Arranjo do conversor <i>Buck</i> utilizado no projeto.	39
Figura 23 - Circuito retificador LDO de 3.3V.	40
Figura 24 - Sistema de acionamento eletromecânico da alimentação do motor.	41
Figura 25 : Interface USB-UART e conversor CH340C.	42
Figura 26 : Conexões do microcontrolador ESP32-WRROM32 e botões push-pull de controle.	43
Figura 27 : Conexões do <i>driver</i> de controle de motor DC L298N.	44
Figura 28 : Conversor de nível lógico LLC.	45

Figura 29 : <i>Slot</i> de conexão para cartão SD e CI 74LVC125APW isolador.....	46
Figura 30 : Conectores para sensores ou GPIO.....	46
Figura 31 - Roteamento das conexões na PCB (a) Camada superior (b) Camada inferior ..	48
Figura 32 -Visualização 2D em CAD da PCB projetada. Vista (a) superior (b) inferior.....	48
Figura 33 - Visualização 3D em CAD da PCI projetada. (a) Vista superior (b) com secções em destaque e numerada.....	49
Figura 34 : Padrão Interface/Módulo/Controlador adotado.....	52
Figura 35 : Arquitetura de rede mestre/escravo Modbus TCP/IP.....	53
Figura 36 : Fluxograma do firmware dos módulos escravos Modbus TCP/IP.....	56
Figura 37 : Fluxograma da conexão Wi-Fi e criação da rede Modbus TCP/IP.....	57
Figura 38 : Fluxograma das mensagens Modbus para sincronismo de data e hora.....	59
Figura 39 : Placa de circuito impresso fabricada por JLCPCB (a) vista superior (b) vista inferior.....	60
Figura 40 : Placa de circuito impresso do sistema finalizada.....	61
Figura 41 : Teste de alimentação da placa. Alimentação (a) diretamente polarizada (b) inversamente polarizada.....	62
Figura 42 : Avaliação do sinal USB - CH340C antes e após resistor série.....	63
Figura 43 : Montagem para teste de acionamento de um motor BDC.....	64
Figura 44 : Teste de acionamento de motor BDC.....	65

LISTA DE TABELAS

Tabela 1	- Relação dos diferentes tipos de geração elétrica no Brasil.....	12
Tabela 2	- Eventos solares anuais de interesse.....	17
Tabela 3	- Comparação entre os componentes utilizados na simulação e no circuito real....	38
Tabela 4	- Mapa lógico de acionamento do CI L298N na configuração paralelo.	44
Tabela 5	- Identificador das secções da Figura 26-b.	49

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Aspectos gerais.....	10
1.2	Antecedente do problema	13
1.3	Descrição do problema	14
1.4	Objetivo	14
1.5	Objetivos específicos	15
2.	REVISÃO BIBLIOGRÁFICA	16
3.	MATERIAIS E MÉTODOS	20
3.1	Requisitos do sistema	20
3.1.1	Atuadores	21
3.1.2	Sensores	24
3.1.3	Registrador de dados	29
3.1.4	Microcontrolador	30
4.	DESENVOLVIMENTO	33
4.1	Sistema de Hardware	33
4.1.1	Sistema de gerenciamento e distribuição de energia	34
4.1.2	Sistema microprocessador e conversor serial para atualização de hardware;	41
4.1.3	Sistema de controle do motor DC	43
4.1.4	Sistema de armazenamento das variáveis de processo	45
4.1.5	Sistema de comunicação com sensor de posição angular;	46
4.2	Projeto da Placa de Circuito Impresso (PCB Design)	47
4.3	Projeto de firmware	50
4.4	Arquitetura de rede	52
5.	RESULTADOS	60
5.1	Placa de circuito impresso	60
5.1.1	Teste de alimentação	61
5.1.2	Upload de firmware	62
5.1.3	Teste do driver de motor L298N	64
5.1.4	Teste de conexão dos sensores	65
5.1.5	Aplicação de malha de controle de velocidade	70
5.1.6	Leitor de cartão SD	Error! Indicador Não Definido .
5.2	Firmware	71
5.3	Rede de comunicação Modbus TCP/IP	Error! Indicador Não Definido .
	CONCLUSÃO	77
	BIBLIOGRAFIAS	79
	APÊNDICE	82

1. INTRODUÇÃO

1.1 Aspectos gerais

Antes da aceitação da teoria heliocêntrica, prevalecia a visão geocêntrica do mundo, fundamentada nos argumentos de Aristóteles (384 a.C. - 322 a.C.) e Ptolomeu (90 d.C. - 168 d.C.). Eles sustentavam que a Terra ocupava o centro do universo, com todos os corpos celestes orbitando ao seu redor. Naquela época, não era incomum conceber a Terra como o epicentro do cosmos, pois as leis fundamentais da natureza ainda eram desconhecidas. As teorias se baseavam em experimentações e pura observação, realizadas com recursos extremamente limitados.

No entanto, desafiando as ideologias predominantes e contrariando a crença da época, Nicolau Copérnico (1473-1543) publicou, em 1543, durante seu leito de morte, a obra "Da revolução de esferas celestes" ou "*De Revolutionibus Orbium Coelestium*" em latim, um livro que representava anos de pesquisa de Copérnico sobre o movimento do Sol e dos corpos celestes conhecidos na época, fundamentado fortemente em princípios matemáticos. Pela primeira vez, Copérnico introduziu publicamente a ideia de que a Terra realiza três tipos de movimentos: rotação em torno de seu próprio eixo (rotação diária), translação ao redor do Sol (volta anual) e precessão em torno de seu eixo eclíptico (inclinação anual de seu eixo). Esse livro desencadeou uma revolução na visão do mundo da época, eventualmente derrubando as teorias geocêntricas predominantes (COPÉRNICO, 2003).

Servindo de inspiração para outros grandes nomes da física, como Johannes Kepler (1571 - 1630) que em seu livro “Astronomia Nova” publicado em 1610, apresentava as suas duas primeiras leis acerca do movimento dos planetas ao redor do sol (TOSSATO; MARICONDA, 2010) e Galileu Galilei (1564 – 1642) com seu folheto “Mensageiro Sideral”, tradução livre de “*Sidereus Nuncius*”, publicado em 1610 que apresentou observações feitas do espaço através de telescópios desenvolvidos pelo próprio Galileu, trazendo novos fatos acerca do que viria ser descoberto como a Via Láctea e o universo, trazendo consigo fortes argumentos que confrontavam o geocentrismo.

Esses cientistas foram muito importantes nos seus campos de pesquisa e dedicaram suas vidas, arriscando-se para fazer ciência em um tempo que isso não era permitido, pois iniciaram uma busca por conhecimento nos céus, estudando os astros que nos rodeiam e

trazendo informações a cerca deles, que formam a base de tudo que se é aprendido e aceito hoje.

Com as descobertas do comportamento dos astros, pode-se então compreender suas características de movimento, traçar trajetórias, calcular suas posições e fazer previsões de posição de cada um no céu de maneira precisa. Quando focamos no Sol como nosso astro de estudo, tais feitos se tornam ainda mais importantes uma vez que o sol é uma grande fonte de energia, principalmente nos dias de hoje, ao qual somos capazes de aproveitar sua energia, não apenas no âmbito da agricultura ou arquitetura, mas também como uma fonte geradora de energia elétrica inacabável.

Com as descobertas feitas por Alexandre Edmond Becquerel em 1839 quando descobriu o efeito fotovoltaico, efeito que transforma a energia dos raios solares em energia elétrica e Willoughby Smith que mais tarde, em 1873 descobriu a fotocondutividade do selênio, que originalmente era um isolante, mas se comportava como um condutor na presença de raios solares e não apenas conduzia eletricidade como também era capaz de gera-la, teve-se em 1883 a criação da primeira célula fotovoltaica por Charles Fritts e em 1958, Russell Ohl patenteou o primeiro sistema fotovoltaico, o mais próximo do que temos hoje (RICHARDSON, 2018), um sistema capaz de produzir energia elétrica através dos raios solares que poderia ser facilmente instalado em qualquer lugar.

Não se sabia na época que tais descobertas e invenções iriam mudar a forma como a geração de energia passou a ser feita ao redor do mundo, tornando a geração solar uma das principais e mais cobiçadas fontes geradoras de energia renováveis no planeta. Atualmente há um grande número de incentivos em escala global para essa prática de geração, um reflexo dos incentivos mencionados, foi a produção primária de energia solar por painéis fotovoltaicos aumentando 395% entre 2003 e 2013, frente a 56% das outras fontes renováveis. Somente o crescimento da geração de energia eólica superou a energia solar nesse período (SILVA, 2015).

No Brasil, em 2021 a marca de geração solar está para atingir 8GW de potência (sieBRASIL, sd), representando apenas 4% da geração de energia no país, como mostrado na Tabela 1, um valor baixo se comparado ao potencial de geração que esse tipo de tecnologia pode nos oferecer. Atualmente, a maior usina de geração solar do Brasil está localizada na cidade de São Gonçalo do Gurguéia, no Piauí. Contando com mais de 2.2 milhões de painéis solares em uma região semiárida do Brasil podendo chegar a gerar 2,2TW por ano de energia (GREEN POWER, 2021) sendo também considerada a maior Usina de geração solar da América do sul. A Usina de São Gonçalo por sua vez, conta com um sistema de rastreio do

sol com um grau de liberdade, sendo capaz de seguir o sol no seu movimento de azimute e zênite, variado sua inclinação com o solo devido sua posição geográfica favorecida.

Tabela 1 - Relação dos diferentes tipos de geração elétrica no Brasil.

Tipo de geração	Unidade	Produção	Percentagem
Hidroelétrica	MW	109294,05	60,89%
Eólica	MW	17146,13	9,55%
Solar	MW	7922,22	4,41%
Térmica	MW	43152,37	24,04%
Nuclear	MW	1990,00	1,11%
Total	MW	179504,77	100,00%
Valores referentes a 01 set 2021.			

Fonte 1: adaptação de sieBRASIL, sd.

Ao contrário dos painéis móveis, os painéis solares fixos, que possuem uma limitação de irradiação que se agrava na medida em que se sai das regiões próximas à linha do equador. Devido aos movimentos do sol ao longo do dia (rotação), somados aos movimentos do sol ao longo do ano (translação), os painéis fixos perdem grande parte da irradiação que poderiam receber no período do dia. Estima-se que 40% da energia solar é desperdiçada para configurações de painéis fixos em regiões afastadas da linha do equador quando comparados com painéis não fixos com 2 graus de liberdade e até 30% com apenas um grau de liberdade (VALDOREIX GREENPOWER, 2015).

Visando um notável crescimento no uso de sistemas de rastreamento solar na geração de energia, esses sistemas têm se tornado uma escolha popular devido ao seu potencial de aumentar significativamente a eficiência e o rendimento das instalações solares. No entanto essa solução ainda é bastante cara do ponto de vista econômico, pois necessita não apenas de uma quantidade maior de componentes elétricos, mas também de uma quantidade maior de circuitos eletrônicos auxiliares que demandam de uma engenharia de *hardware* cuidadosa, a fim de manter o sistema estável e em pleno funcionamento.

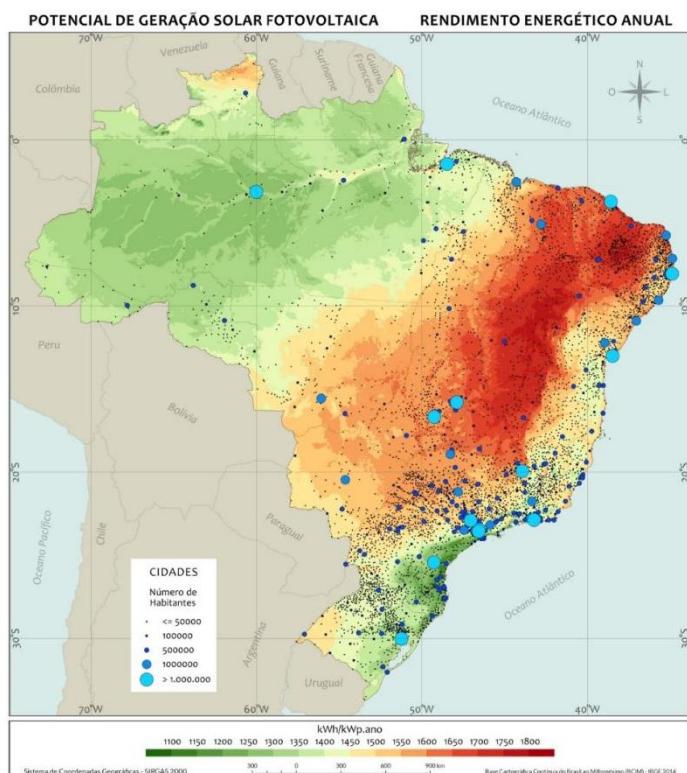
Como forma de incentivar a adoção desse método de geração solar e conseguir trazer um rendimento de geração superior aos tradicionais, estima-se que com o desenvolvimento de uma plataforma de *hardware* aberta e replicável, promover o desenvolvimento de novos sistemas de rastreamento solar para painéis fotovoltaicos. Com a promoção de novos *hardwares* como alternativas baratas para esse tipo de sistema. Além disso, novas pesquisas no campo da geração solar podem ser promovidas a fim melhorar ainda mais a eficiência da

energia solar no Brasil e tornar essa tecnologia, acessível para clientes residenciais, a fim de não restringir essa tecnologia apenas para parques de geração solar de larga escala.

1.2 Antecedente do problema

O Brasil é um país de grande área territorial, possuindo muitos climas e características próprias em cada região, possuindo predominantemente um clima tropical, semiárido e subtropical. Por possuir áreas que cortam desde a linha do equador até regiões subtropicais, a geração de energia solar torna-se um problema para as regiões mais afastadas da linha do equador, necessitando de sistemas mais inteligentes e capazes de gerar mais energia com menos irradiação solar. Na Figura 1 pode-se ver os níveis de irradiação anuais médios para irradiação solar normal no território brasileiro. Destacam-se duas regiões no mapa, sendo a primeira região, a região de clima semiárido em vermelho, localizados aproximadamente em 10°S 45°W, que correspondem a zonas ideais para a implementação de usinas de painéis solares fixos ou de um grau de liberdade como a de São Gonçalo e a segunda região localizada aproximadamente a 30°S 54°W, de clima subtropical correspondente ao estado do Rio Grande do Sul que será o ponto estudado.

Figura 1: Mapa do potencial de geração solar fotovoltaica em termos do rendimento energético anual para todo o Brasil (medido em kWh/kWp.ano)



Fonte: PEREIRA (2017).

Devido a grandes extensões territoriais e estando situado em grande parte na região subtropical, o Brasil possui grande potencial de geração de energia solar fazendo uso de sistemas de rastreamento solar. Com essa perspectiva de crescimento e visando a difusão da adoção dessa prática de geração, estima-se que com o desenvolvimento de um *hardware* aberto de controle desse sistema, de forma modular, possa-se incentivar a adoção dessa prática, contribuindo para o desenvolvimento sustentável e formando uma solução energética mais eficiente.

1.3 Descrição do problema

Devido ao complexo sistema de rastreamento, que permite o uso de motores de alta potência, inversor de frequência e uma gama de sensores para posicionamento, detecção de condições climáticas adversas, sistemas de controle em diferentes malhas e processamentos de dados, como valores de geração de energia dos painéis solares e consumo de energia por parte do sistema, o sistema *Tracker* se torna um sistema complexo para ser gerenciado e que agrupa um valor econômico muito grande. Dessa forma, a fim de viabilizar sistemas de rastreamento *Trackers* de baixa potência como solução viável, deve-se adotar um sistema de *hardware* barato e replicável, de forma a assegurar uma fácil manutenção e garantir a segurança e confiabilidade de operação do sistema, tornando essa, uma solução mais atrativa para consumidores residenciais.

1.4 Objetivo

Considerando os benefícios proporcionados pelo uso de sistemas de rastreamento solar para potencializar a geração de energia dos painéis solares convencionais, propõe-se a concepção de um sistema de *hardware* aberto, acessível e replicável. Esse sistema seria destinado ao controle de *trackers* que empregam motores de corrente contínua e realizam o sensoriamento da posição angular do eixo de saída por meio de sensores magnéticos, incrementais ou resistivos. A solução proposta consiste em um *hardware* capaz de assegurar o posicionamento dos eixos em um sistema de rastreamento solar. O projeto engloba a criação de um módulo eletrônico destinado ao controle de um único eixo do sistema. No entanto, esse módulo é projetado para integração com outros módulos semelhantes, formando uma rede de comunicação Modbus TCP/IP, a fim de estabelecer uma malha de controle única.

O módulo eletrônico desenvolvido foi criado visando-se a aplicação em sistemas de rastreamento com dois graus de liberdade, para oferecer ao sistema, a liberdade de movimento para acompanhamento dos ângulos de azimute e zênite do sol ao longo do dia. No entanto, ele não se restringe a essa aplicação unicamente, podendo ser aplicado em soluções de apenas um grau de liberdade ou aplicações genéricas que necessitem do controle de ângulo de motores DC com um posicionamento preciso de posição. O objetivo do presente trabalho não é o desenvolvimento do sistema de rastreamento solar mecânico, se limitando exclusivamente no desenvolvimento de *hardware* do módulo empregado no controle desse tipo de estrutura.

1.5 Objetivos específicos

Para o desenvolvimento de um *hardware* aberto voltado para o controle de estruturas de geração solar do tipo *tracker*, tem-se como objetivos específicos do projeto descrito no documento:

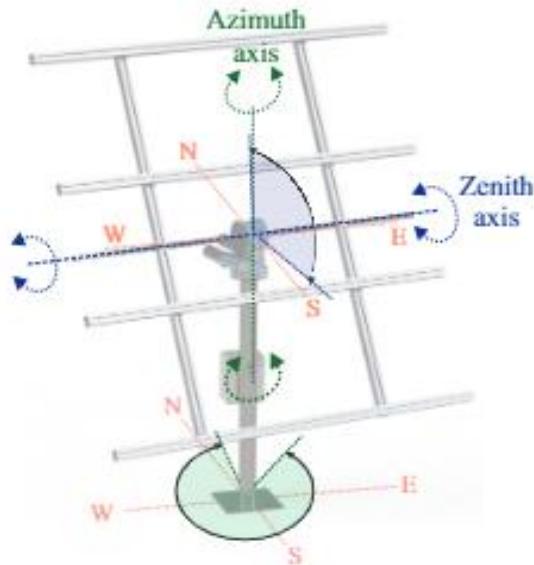
- I. Desenvolver um módulo eletrônico capaz de controlar sistemas de rastreamento solar, focado em operar em dois graus de liberdade para rastrear os ângulos de azimute e zênite do sol ao longo do dia;
- II. Garantir a adaptabilidade do módulo para ser aplicado não apenas em sistemas de dois graus de liberdade, mas também em configurações de um único grau de liberdade;
- III. Criar um *hardware* versátil, capaz de ser utilizado em aplicações genéricas que exijam controle de ângulo em motores de corrente contínua, oferecendo precisão de posicionamento.
- IV. Estabelecer a comunicação eficiente entre os módulos, visando a formação de uma rede de controle única por meio do protocolo Modbus TCP/IP.
- V. Assegurar a integração perfeita entre os módulos para formar uma malha de controle consistente, possibilitando o funcionamento conjunto e sincronizado em sistemas de rastreamento solar.
- VI. Criar um sistema de *hardware* aberto e replicável, para que possa se difundir a adoção dessa técnica de geração solar para residências do sul do Brasil, a fim de melhor o desempenho desses sistemas.

2. REVISÃO BIBLIOGRÁFICA

De acordo com Queiros, Souza e Andrade (2018), os rastreadores solares podem ser classificados quanto a quatro características principais, sendo elas: rastreados ativos ou passivos; de um eixo ou dois eixos; sistema de controle em malha aberta ou fechada; estratégia de rastreamento cronológica ou por sensores. Essa classificação também está presente na detalhada análise de Nsengiyumva (2018).

De acordo com essas características dos rastreadores, é pertinente se adotar estratégias de controle de posicionamento para cada característica do sistema adotado. No entanto as abordagens sempre seguiram as mesmas. Ainda de acordo com Nsengiyumva (2018), o papel principal de qualquer seguidor solar será compensar as mudanças de posição do sol, ou seja, os ângulos de azimute e zênite (vide Figura 2), ao decorrer do dia e principalmente ao longo das estações.

Figura 2 - Ângulos de Azimute e Zênite referentes a um seguidor solar



Fonte: NSENGIYUMVA (2018).

Como forma de validação da necessidade da utilização dessa compensação ao longo das estações do ano, através do site SunCal.org (SUNCACAL, sd) podem ser determinados os limites de movimento necessários para um seguidor solar tomando como referência a posição geográfica da Universidade Federal de Santa Maria, posicionada em Latitude - 29.71345° e Longitude -53.71719° durante o ano de 2024. Os dados coletados podem ser

vistos na Tabela 2 e correspondem aos principais eventos que ocorrem ao longo do ano que são os equinócios, quando os dias possuem a mesma quantidade de horas de luz e noite e os solstício que são os pontos máximos de diferença entre as horas de luz e noite. Esses pontos são importantes, pois marcam as transições entre as estações do ano e em consequência, os limites máximos e mínimos do sol na eclíptica solar de qualquer região. Dessa forma, fica evidente os limites mínimos que um rastreador deverá seguir em cada eixo a fim de se manter sempre alinhado ao sol.

Tabela 2 - Eventos solares anuais de interesse.

Dia	Estação	Azimute			Zênite		
		Min.	Máx.	Dif.	Min.	Máx.	Dif.
21/03/2023	Outono	90.28º	269.95º	180.33 º	0º	59.99º	59.99º
21/06/2023	Inverno	93.31º	296.69º	156.62º	0º	36.87º	36.87º
21/09/2023	Primavera	90.28º	269.95º	180.33º	0º	59.99º	59.99º
21/12/2023	Verão	117.84º	242.16º	235.68º	0º	83.73º	83.73º

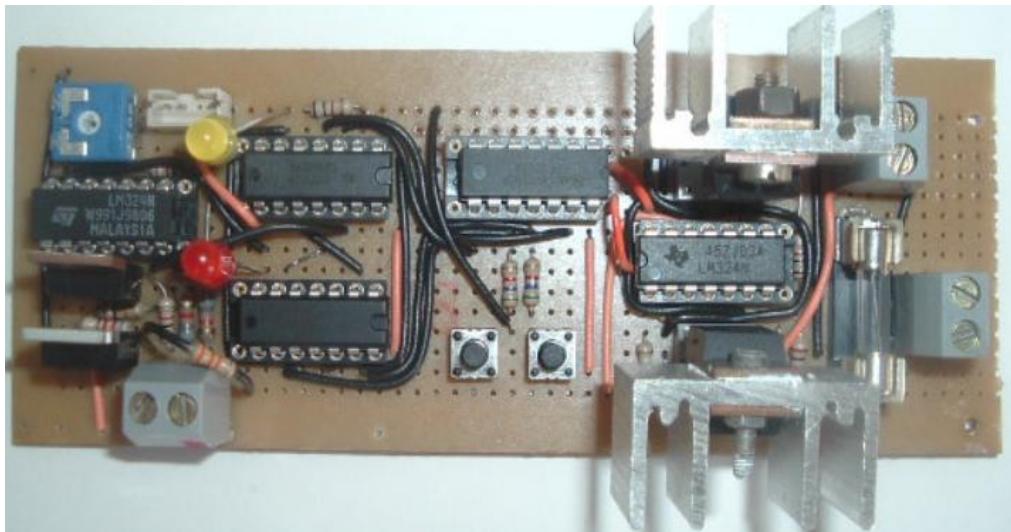
Fonte: Próprio autor.

Dessa forma, diversos autores desenvolvem seus sistemas de rastreamento de forma a conseguirem manter os painéis solares sempre alinhados à posição aparente do sol, sendo capazes de operar entre os limites de azimute e zênite de cada região, com sistemas que devem possuir o menor consumo possível a fim de se obter o maior rendimento energético.

Em Gonçalves Filho (2007), o autor propôs a criação de um sistema de rastreamento solar de um eixo móvel, ficando o eixo de azimute fixo na latitude da cidade de Recife - PE. Para o projeto, o autor utilizou de sensores de irradiação solar, utilizando resistores dependentes de luz, conhecidos popularmente como LDR (do inglês *Light Dependent Resistor*), distribuídos em 4 quadrantes para que seja possível se realizar o rastreamento do sol devido as diferenças de medição dos sensores. Devido ao uso desse método de sensoriamento, o sistema apresentou algumas falhas de rastreio devido a irradiação de luz solar por difusão, que ocorre quando o dia está nublado ou quando o sol é encoberto por nuvens densas. Além disso, por não se tratar de um sensor absoluto, foi necessário se adicionar chaves de fim de curso para se iniciar o movimento dos motores em posição conhecida e retorno ao fim do dia, sendo uma técnica que despende de maiores custos energéticos por realizar movimentos desnecessários e manter os atuadores em funcionamento por um tempo além do necessário. No entanto, mesmo assim o sistema conseguiu atingir um valor de rendimento com máximas de 31% quando comparados a sistemas fixos e mínimas de 15%, se mostrando uma solução viável.

O autor desenvolveu seu módulo de controle, como pode ser visto na Figura 3. De acordo com ele, os valores em eletrônica para o protótipo, ficaram em torno de 10% do valor total do sistema, sendo 90% em valores relacionados a estrutura mecânica.

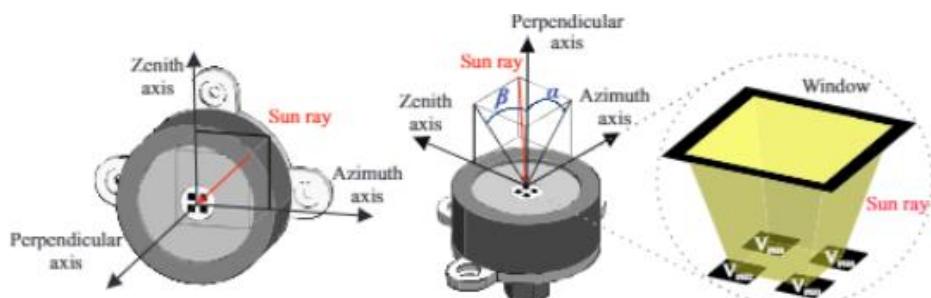
Figura 3 - Módulo de controle utilizado por GONÇALVES FILHO (2004).



Fonte: GONÇALVES FILHO (2004).

Já em Angulo-Calderón (2022), o autor propôs o desenvolvimento de um sistema de rastreamento solar de dois eixos para ser atuado na cidade de Aguascalientes - México. No trabalho proposto, fora utilizado um sensor capaz de medir o ângulo de incidência solar através do sensor ISS-A60 (Figura 4). O sensor solar ISS-A60 realiza a medição do ângulo de incidência de um raio solar nos dois eixos ortogonais, pois realiza as medições de incidência solar em 4 quadrantes. Sua alta sensibilidade é alcançada por meio das dimensões geométricas do *design* da lente do sensor e as características distintivas o tornam uma ferramenta adequada para sistemas de rastreamento e posicionamento solar de alta precisão, caracterizados por baixo consumo de energia e alta confiabilidade, utilizando tecnologia MEMS (do inglês *Micro-Electro-Mechanical Systems*).

Figura 4 - Sensor ISS-A60 utilizado por ANGULO-CALDERÓN (2022).



Fonte: ANGULO-CALDERÓN (2022).

Os resultados obtidos por Angulo-Calderón (2022) mostraram de forma experimental que utilizando o sistema proposto, houve um aumento da geração do sistema fotovoltaico de 37,5% em relação a um sistema fixo. No entanto, mesmo que o autor tenha utilizado um sensor com capacidades superiores aos LDRs, ele ainda não é absoluto, se depende de chaves fim de curso para dar inicio ao movimento do atuador, sendo uma solução que desperdiça parte da energia gerada em movimentos desnecessários do atuador.

Outros sistemas de rastreadores podem ser visto em Abreu Vieira (2014) e em Barbosa (2009), só que desta vez, sendo empregado no rastreamento do sol para concentradores, cuja função é aquecer um tambor, normalmente de água, para ser utilizado por diversos propósitos. No caso de Barbosa, foi utilizado um sensor LDR para se realizar o rastreio solar, e mesmo assim, o concentrador conseguiu atingir resultados consideráveis, atingindo diferenças de temperaturas de até 50°C dentro dos concentradores. No entanto, em dias com irradiação difusas, o concentrador se perdeu em algumas situações. Na Figura 5 pode ser visto a placa eletrônica para controle do rastreador utilizado por Barbosa.

Figura 5 - Módulo de controle utilizado por BARBOSA (2009).



Fonte: BARBOSA (2009).

3. MATERIAIS E MÉTODOS

Este tópico explora os materiais e métodos empregados no desenvolvimento do *hardware* do projeto. Será detalhado os requisitos do sistema de rastreamento solar para a implementação do *hardware*, incluindo as exigências para a criação de uma rede de controle Modbus TCP/IP. Além disso, serão apresentados os materiais utilizados para os sistemas de sensoriamento e atuação, bem como os *softwares* de simulação e *design* de *hardware* empregados ao longo do processo de desenvolvimento.

Em cada subdivisão, será analisado as características específicas do sistema em cada área e os requisitos necessários para seu funcionamento. Isso proporcionará clareza quanto às decisões tomadas em relação à seleção de componentes e recursos utilizados no decorrer do projeto.

3.1 Requisitos do sistema

Inicialmente, destaca-se os principais componentes necessários para a implementação do sistema de rastreamento solar, destacando-se as três principais áreas de interesse: atuação, sensoriamento e rede de controle.

Na etapa de atuação, é fundamental salientar a importância de controlar um dos eixos do sistema de rastreamento. Por meio de um circuito de controle, esse eixo é responsável por assegurar que o sistema esteja devidamente alinhado com seu objetivo, que é manter-se direcionado ao sol. Dado o propósito de criar um sistema versátil, aplicável a diversos métodos de rastreamento, e permitindo o uso de diferentes dispositivos de atuação, a seleção do atuador desempenha um papel crucial na garantia dessa característica do sistema.

Na etapa de sensoriamento, é crucial estabelecer um método eficaz para monitorar periodicamente e de maneira confiável a posição angular do motor, viabilizando a obtenção precisa do ângulo de saída do atuador. Essa fase é fundamental para assegurar o controle preciso da posição do atuador.

Finalmente, a etapa da rede de controle, será a responsável pelo estabelecimento da comunicação entre os módulos do sistema, sendo fundamental para o correto funcionamento do sistema como um todo. Tratando-se de uma rede de controle fundamentada no protocolo Modbus TCP/IP, é necessário de estabelecer os requisitos mínimos para se obter

essa rede que faz uso dos protocolos de Internet, utilizando dispositivos compatíveis com redes *Wi-Fi*.

3.1.1 Atuadores

Com o propósito de desenvolver um sistema versátil, capaz de se integrar a uma ampla variedade de *drivers* para atuação de motores em sistemas de rastreamento, a escolha recaiu sobre atuadores que operam com corrente contínua, utilizando motores comumente conhecidos como motores DC (do inglês *Direct Current*).

Considerando a aplicação desses motores em sistemas de controle de posição como os servo atuadores, que usam motores DC escovados, geralmente dotados de uma considerável redução na relação de velocidade de entrada por velocidade de saída do motor, a opção por esse tipo de atuador foi preferida. Além disso, eles possuem as seguintes características que se relacionam com a aplicação em questão:

- 1) Utilização de uma fonte de energia compatível com sistemas de geração de energia solar (fonte DC);
- 2) Amplo emprego de métodos para controlar a direção de rotação (direção é dada pelo sentido da corrente na armadura do motor);
- 3) Precisão no controle de velocidade (velocidade varia com a tensão);
- 4) Boa relação entre torque e velocidade.

Essas qualidades tornam os motores de corrente contínua uma escolha sólida e adequada para o propósito de implementação em sistemas de rastreamento, fornecendo controle preciso e eficiente para manter a orientação correta em direção ao sol.

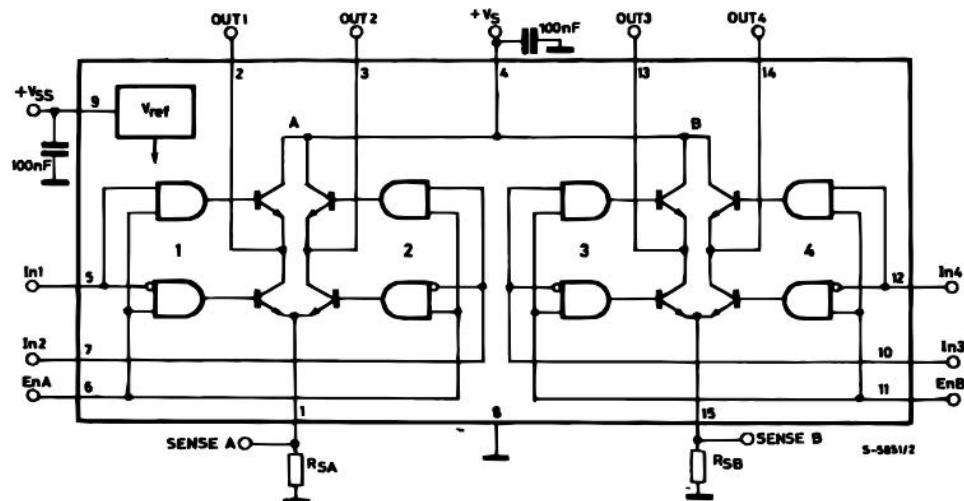
Partindo dessa premissa, o sistema de *hardware* deverá ser capaz de realizar o acionamento de um motor DC escovado e através de uma malha de controle, onde deverá ser capaz de controlar sua posição através da variação de velocidade e sentido de rotação.

Em Fitzgerald (1975) é apresentado alguns aspectos construtivos dos motores DC escovados com imãs permanentes e em Araújo (2022) são apresentados algumas técnicas de acionamentos desses motores nos quatro quadrantes de atuação, onde é possível se obter operações de operação direta e inversa do motor e frenagem direta e regenerativa do motor, através do uso de topologias de ponte completa ou como é comumente chamado, ponte H. Além disso, em Araújo (2022) também é comparado as técnicas de controle de velocidade do motor DC utilizando controladores proporcional-integral e lógica *fuzzy* sobre esta topologia.

Em Matos (2008) a topologia de ponte H é explorada com uma riqueza de detalhes, possuindo uma grande participação no trabalho desenvolvido, uma vez que o autor explora todos os aspectos funcionais e construtivos de uma ponte H.

Dessa forma, a fim de se realizar o controle de velocidade de um motor DC escovado, optou-se pelo uso de um circuito integrado de ponte-H que permitisse o seu controle de velocidade através da técnica de modulação PWM. O Circuito Integrado (CI) utilizado para esse controle foi o L298N. Esse mesmo CI fora utilizado por Belo (2004) para compor a construção de um robô sentinela do tipo *rover*, que utiliza duas rodas em uma característica de movimento diferencial com um apoio central. Para a construção do sentinela, o CI L298N foi empregado para realizar o controle das duas rodas, uma vez que o CI possui dois canais de controles para motores DC independentes, realizando o controle de velocidade e direção de giro através da modulação PWM. Na Figura 6 é possível se observar os aspectos construtivos do CI.

Figura 6 - Esquema lógico da ponte H L298N.



Fonte: Folha de dados do componente L298N.

De acordo com a folha de dados do componente L298N em STMicroelectronics (2024, pg. 1):

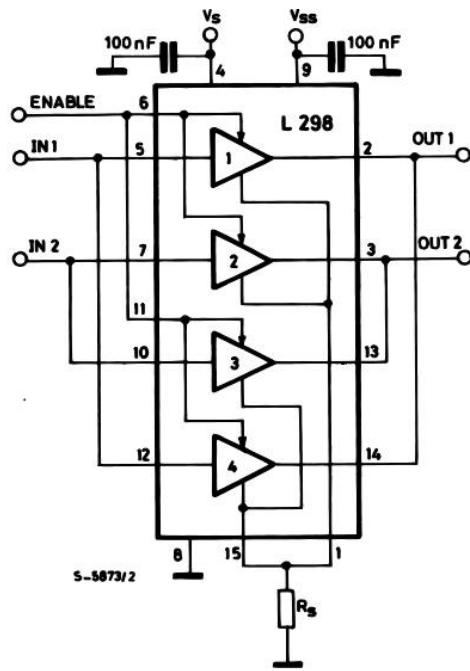
“O L298N é um circuito integrado monolítico em um invólucro *Multiwatt* de 15 pinos e no formato PowerSO20. Trata-se de um driver duplo de ponte completa de alta voltagem e alta corrente, projetado para aceitar níveis lógicos TTL padrão e controlar cargas indutivas, como relés, solenoides e motores CC e de passo. São fornecidas duas entradas de habilitação para ativar ou desativar o dispositivo independentemente dos sinais de entrada. Os emissores dos transístores inferiores de cada ponte são conectados juntos, e o terminal externo correspondente pode ser utilizado para conectar um resistor de sensoriamento externo. Uma entrada de alimentação adicional é disponibilizada para que a lógica funcione em uma voltagem mais baixa. (STMICROELECTRONICS, 2023, Pg. 1)”

Ainda de acordo com a folha de dados do componente, o CI L298N se mostraria suficiente para alimentar e controlar um motor DC de até 184W operando a 46V. Além disso, ele possui um preço atrativo e demanda de poucos componentes periféricos para seu correto funcionamento e possui um terminal dedicado para a medição de corrente de saída nos canais de motor através da medição da queda de tensão sobre um resistor *shunt*, permitindo que a corrente no motor seja monitorada pelo microcontrolador. Na Figura 7, é representado o arranjo com canais paralelos, com medição de corrente pelo resistor *shunt* representado como R_s .

Com a utilização desse *driver* para controle de motores DC, pode-se utilizar uma ampla gama de motores de pequena potência com compatibilidade garantida, estando dentro dos objetivos de utilização do projeto.

Motores DC utilizados em patinetes e *scooter* elétricos como o motor My4835 ou My6812 (vide Figura 8.a) com faixa de operação de 12 a 24V e potência de 100 a 150W, que chegam a 3350RPM podem ser utilizados a fim de satisfazer os aspectos elétricos e mecânicos do sistema de rastreamento requerido.

Figura 7 - Configuração paralelo do CI L298N recomendado.



Fonte: Folha de dados do CI L298N.

Outra solução de motor para uso é o motor 5840-31zy, disponível pela *RoboBuilders*, sendo um motor 12V com alto torque, sendo capaz de fornecer até 78W de potência (vide Figura 8.b). Esse motor se destaca por fazer o uso de uma redução integrada com valores de RPM que variam de 10 a 250RPM.

Figura 8 - Exemplos de motores DC escovados de baixa potência (A) Motor My4835 (B) 5840-31zy



Fonte: (a) Desconhecido (b) RoboBuilders.

Desta forma, a fim de se realizar a atuação deste tipo de motor DC escovado com potência menores que 184W, o módulo para controle dos atuadores deverá contar com uma saída de potência onde seja possível se conectar o atuador utilizado pelo sistema de rastreio de forma que o conector assegure a operação de até 184W. A solução adotada foi utilizar um conector do tipo *Borne KRE* de 90 graus macho com um conector do mesmo tipo fêmea. O modelo do conector é KF2EDGK-5,08-02P e fora escolhido pela facilidade de se realizar as conexões por permitir que o motor seja parafusado no conector fêmea e posteriormente conectado no conector macho da placa.

3.1.2 Sensores

Quando se trabalha com um sistema cujo desempenho depende essencialmente da precisão no alinhamento dos painéis com o sol, a etapa de sensoriamento torna-se um aspecto crucial do projeto. Essa fase é responsável por fornecer informações à malha de controle, que calcula o erro na posição do eixo em relação à referência e permite a correção do sistema com base nesse erro. Portanto, a qualidade da medição da posição do eixo e a precisão dos sensores utilizados desempenham um papel fundamental na determinação da qualidade geral do sistema de rastreamento.

Como o sistema de atuação foi projetado para operar exclusivamente com motores DC, é necessário monitorar a posição angular, o que exige a utilização de técnicas externas de medição. De acordo com Kilian (1996), existem diversas técnicas para medir a posição

angular e a velocidade do motor, que podem ser essencialmente divididas em dois grupos: sensores relativos; e sensores absolutos.

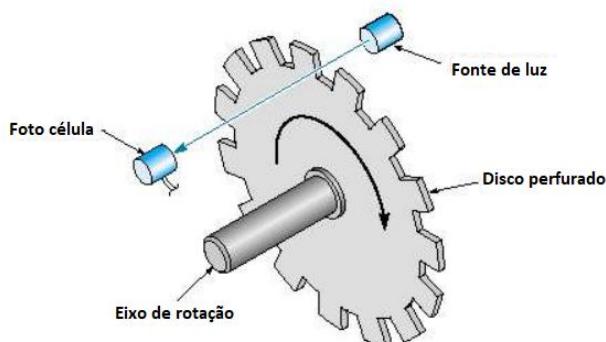
3.1.2.1 Sensores relativos

Os sensores relativos, são os sensores que não possuem um valor real associado a grandeza medida, necessitando de uma referência inicial, como a posição inicial que pode ser dada por uma chave fim de curso, obrigando o sistema a realizar uma etapa de calibração inicial.

Exemplos de *encoders* relativos são os sensores óticos rotativos (Figura 9). Esses sensores possuem um feixe de luz que ao serem interrompidos, geram uma interrupção que deve ser contada. Através desses pulsos, é possível se estipular o deslocamento do motor e/ou estipular parâmetros como velocidade e aceleração.

A vantagem desse tipo de sensoriamento é que não são necessários conversores analógicos para digital e são fáceis de serem instalados. No entanto, eles possuem uma precisão limitada ao número de furações do disco perfurado e necessitam de uma rotina de calibração inicial, além disso, não são capazes de medir o sentido de rotação do motor, para isso são necessários dois discos, como apresentado em Kilian (1996) e em Moreto (2007).

Figura 9 - Encoder incremental ótico com disco perfurado.

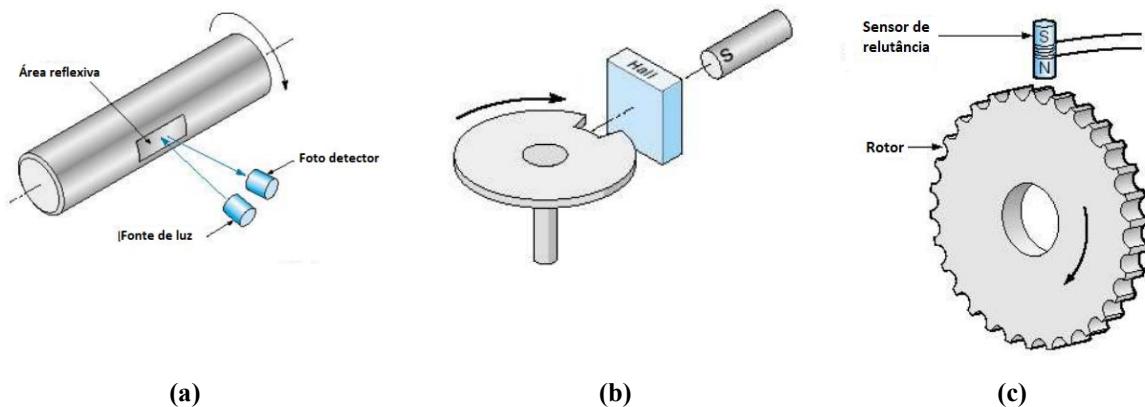


Fonte: Adaptado de KILIAN (1996).

Outra forma de se realizar a montagem de sensores de posição relativos é através do uso de tacômetros, que através da medição da velocidade do eixo do motor, podem estipular a posição do motor através da relação de deslocamento por tempo. No entanto essas técnicas devem ser aplicados no eixo de saída do motor antes de entrar na relação de redução, para conseguir medir a velocidade do eixo com maior precisão, sendo também um ponto de desvantagem dessa técnica, uma vez que a relação de saída deve ser bem conhecida, para que

não sejam carregados erros de medição ao longo do tempo. Na Figura 10 são mostrados exemplos de: Figura 10.a tacômetro ótico, Figura 10.b tacômetro de efeito hall e Figura 10.c tacômetro com coroa dentada.

Figura 10 - Exemplos de tacômetros (a) óticos (b) de efeito hall (c) coroa dentada



Fonte: Adaptado de KILIAN (1996).

A maioria dos sensores relativos possuem um método de contagem incremental, para realizarem o sensoriamento das variáveis de posição e/ou velocidade, dessa forma, o sistema deverá contar com terminais de entrada de sinais digitais, preferencialmente os que permitam interrupções do sistema microcontrolado. Além disso, para o uso de sistemas relativos, é necessário se avaliar a necessidade de calibração ou definição de rotinas de inicialização para que seja adotado uma referência de partida do sistema em um estado conhecido.

Apesar de não serem considerados os melhores sistemas de sensoriamento para a aplicação em questão, considerou-se em projeto a possibilidade de se utilizar esses métodos de medição devido a sua facilidade de aplicação e valor reduzido de montagem e manutenção.

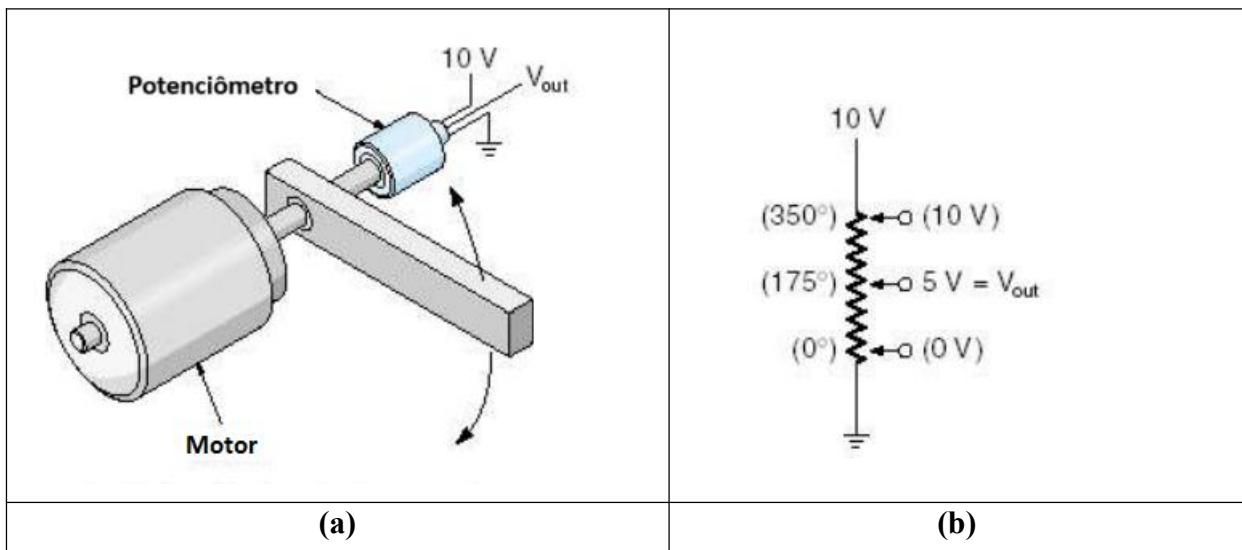
3.1.2.2 Sensores absolutos

Os sensores absolutos, são os sensores capazes de realizar a medição da posição angular do eixo de saída do motor de forma instantânea, ou seja, sem a necessidade de calibração ou referência inicial.

O primeiro exemplo de sensores absolutos são os potenciômetros. Esse tipo de sensor se baseia na conversão do deslocamento angular em um sinal de tensão, que tem seu valor dado através da variação proporcional da resistência desse sensor, trabalhando como um divisor resistivo. A Figura 11.a apresenta o modelo construtivo do emprego desse tipo de

sensor e na Figura 11.b está o circuito equivalente desse sensor, onde é apresentado a derivação central do potenciômetro que varia sua tensão em relação a alimentação do potenciômetro.

Figura 11 - Encoders absolutos do tipo resistivo (a) construção (b) circuito equivalente.



Fonte: Adaptado de KILIAN (1996).

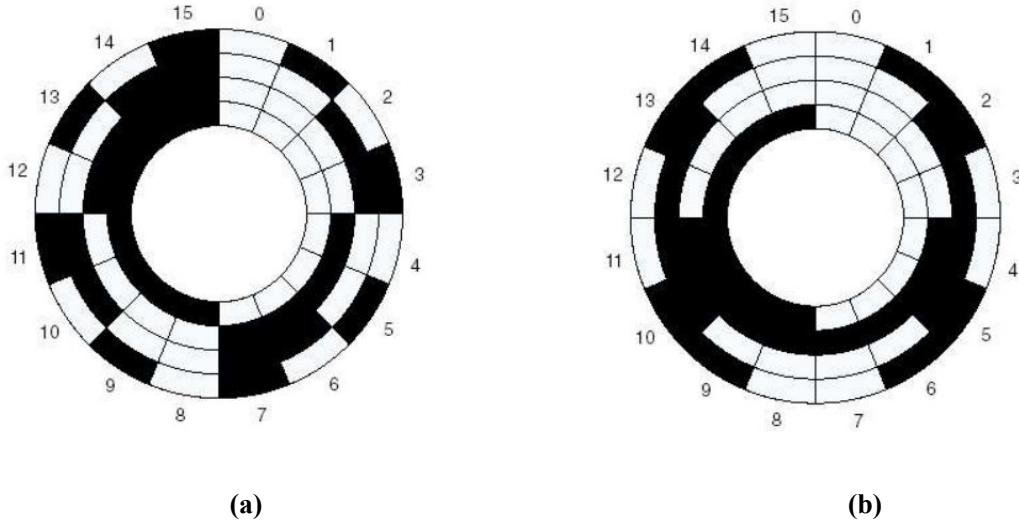
Por se tratar de um sensor que utiliza componentes passivos, ele é ideal para ser empregado em qualquer sistema de posicionamento absoluto, sendo uma solução de baixo custo e fácil manutenção.

Dentre as desvantagens desse tipo de sensor está a limitação de giro de muitos sensores, que são capazes de realizar apenas algumas voltas e necessitam de um bom conversor analógico/digital para sua medição precisa. Além disso, são sensores que possuem um consumo de energia que inviabiliza o seu emprego em sistema de baixo consumo e estão suscetíveis a ruídos do meio externo.

Além dos potenciômetros, pode-se citar os discos com marcação também chamados de *encoders* absolutos ópticos como apresentado em Moreto (2007) e Kilian (1996). Esse disco normalmente é feitos de vidro estampados com um padrão bem definido que pode ser lido por sensores ópticos de forma a representarem um código binário ou em código de *Gray*, representando a posição atual do motor. Na Figura 12.a é mostrado o padrão binário do disco e na Figura 12.b o padrão em código de *Gray* a título de comparação.

Esse método possui a desvantagem de necessitar de muitos sensores ópticos e possuir uma precisão limitada quando comparada à técnica de potenciômetro ou leitura digital. Na Figura 12 são mostrados dois discos com 4 bits de resolução, que permitem uma leitura de apenas 16 posições do eixo as custas de 4 sensores digitais.

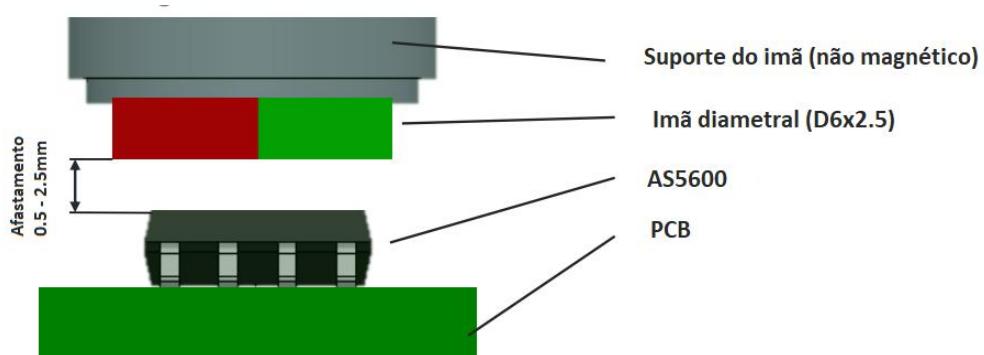
Figura 12 - Encoders óticos com discos codificados com codificação (a) binária (b) de Gray.



Fonte: Adaptado de KILIAN (1996).

Já em Tang (2022) é apresentado uma abordagem de sensor de posição para se realizar o controle em malha fechada de um motor de passo baseado em um *encoder* magnético. O sensor utilizado por Tang é o AS5600, um sensor magnético que faz o uso de um imã de polaridade diametral que deve ser acoplado ao eixo de saída do motor e estar em contato ou próximo o suficiente do circuito integrado AS5600. Na Figura 13 é apresentado o arranjo imã/sensor que representa a relação mecânica entre os componentes.

Figura 13 - Arranjo do sensor magnético AS5600.



Fonte: Folha de dados do componente AS5600A.

De acordo com a folha de dados do sensor AS5600, ele é um sensor de efeito *hall* programável com 12-bit de resolução, o que permite que ele faça medições de ângulos com precisão de 0,00549° (AMS, 2014). Além disso, ele é um sensor que possui comunicação I2C ou saída analógica, necessitando de apenas 2 fios de alimentação 5 ou 3.3V e dois fios de dados SDA e SCL para a comunicação digital via I2C ou 1 único fio para comunicação

analógica via PWM, sendo possível programar os pontos iniciais e finais do sensor de acordo com a necessidade do sistema.

A única desvantagem do sensor AS5600 é a necessidade de se utilizar um imã com polarização diametral em um suporte sobre o eixo de forma que seja possível mensurar o campo magnético gerado por esse imã pelo sensor e o alto custo agregado no sensor e na placa de circuito que será necessária para comportar o CI. Todavia, com todas as vantagens que esse método de medição oferece e em comparação aos demais sensores, essa abordagem de sensoriamento é a mais recomendada para ser utilizada com o projeto.

Em conclusão, existem muitos sensores de posição que podem ser usados de acordo com a necessidade de cada usuário. Sendo assim, para o desenvolvimento do *hardware*, foram disponibilizadas as entradas para todos os sensores citados acima, havendo destaque para o uso de sensores digitais através da interface I2C. Desta forma o sistema contará com diversos conectores genéricos para a conexão do sensor que melhor adaptar ao sistema e a configurações dele, ficará a cargo da programação do *firmware* do microcontrolador.

3.1.3 Registrador de dados

Para garantir que o sistema funcione corretamente e de forma eficiente, é importante poder monitorar seu desempenho e identificar possíveis problemas. Dessa forma, é necessário armazenar dados do sistema, como por exemplo os níveis de irradiação solar, estado de operação, temperatura do sistema e consumo elétrico dos motores. Uma das formas de se fazer isso é através de um *SD card* (do inglês *Secure Digital Card*) ou como é conhecido popularmente como cartão SD.

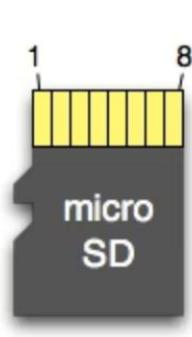
O cartão SD é uma mídia de armazenamento compacta e acessível, que pode armazenar grandes quantidades de dados. Ele é conectado ao microcontrolador do sistema por meio de um *slot* permitindo que esse cartão seja facilmente removido do sistema para se realizar as auditorias quando necessário.

Os dados armazenados no cartão SD podem ser usados para gerar relatórios que ajudam a acompanhar o desempenho do sistema, identificar problemas e tomar medidas corretivas, como por exemplo, se os dados de irradiação solar mostrarem que o sistema está operando abaixo da capacidade máxima, isso pode indicar um problema com a instalação dos painéis solares.

Além disso, os relatórios podem ajudar os operadores a identificar o problema e tomar medidas para corrigi-lo antes que ele cause perdas de energia ou então os dados de temperatura podem ser usados para monitorar o desempenho dos motores e identificar possíveis problemas de super aquecimento que podem levar à perda de eficiência do sistema. Dessa forma, o armazenamento dos dados do sistema podem se tornar uma medida útil para se monitorar o funcionamento do sistema e permitir que melhorias sejam tomadas e medidas de manutenção preventivas sejam tomadas no momento certo.

A Figura 14 mostra a configuração de pinos de um cartão SD. Esse tipo de cartão de memória pode ser utilizado através do protocolo de comunicação SPI (do inglês *Serial Peripheral Interface*), um tipo comum de comunicação em que os dados são transmitidos de forma sequencial e de forma síncrona com o microcontrolador, estabelecendo uma comunicação rápida e confiável, capaz de ler e escrever os dados em uma comunicação bidirecional de até 2Mbps (EMBARCADOS, sd).

Figura 14 - configuração dos pinos de um micro cartão SD.



Pin	SD	SPI
1	DAT2	X
2	CD/DAT3	CS
3	CMD	DI
4	VDD	VDD
5	CLK	SCLK
6	VSS	VSS
7	DAT0	DO
8	DAT1	X

Fonte: EMBARCADOS, sd.

3.1.4 Microcontrolador

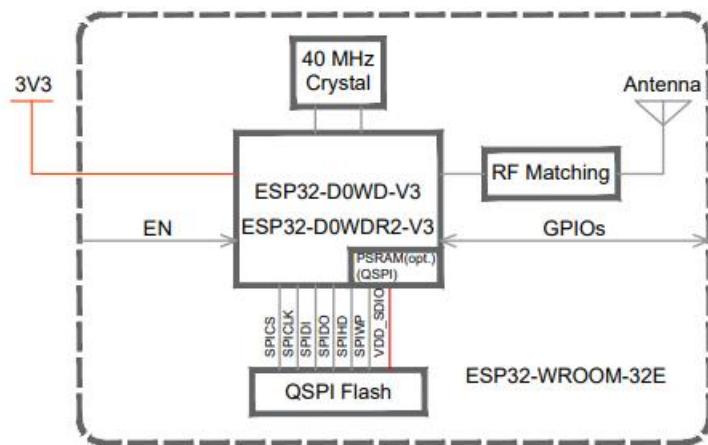
Para atender aos requisitos de comunicação do sistema, optou-se por adotar uma rede que proporcionasse uma implementação econômica e eficiente. O Modbus TCP/IP se destacou como a solução ideal devido às suas características que se alinham com essas necessidades, sendo um protocolo muito utilizado na indústria e possuindo a característica de ser estabelecido em redes *wireless*.

Dado que a solução escolhida é o Modbus TCP/IP, o microcontrolador selecionado para integrar o módulo precisava atender a certos critérios. Especificamente, o

microcontrolador deveria ser compatível com uma antena *Wi-Fi* externa ou possuir um módulo *Wi-Fi* interno. Além disso, o microcontrolador precisava incorporar periféricos suficientes para acionar os atuadores e ler os diversos tipos de sensores mencionados.

Para satisfazer esses requisitos, optou-se pelo microcontrolador ESP32-D0WDR2-V3, integrado na placa de desenvolvimento ESP32-WROOM-32E (vide Figura 15). Este microcontrolador é capaz de abranger todos os componentes periféricos necessários, garantindo que o sistema funcione conforme esperado e aproveite plenamente as vantagens do Modbus TCP/IP.

Figura 15 - Configuração de blocos do microcontrolador ESP32-WROOM32E.



Fonte: ESPRESSIF, sd.

O ESP32 constitui uma linha de micro controladores que se destacam por seu custo acessível e baixo consumo energético, além de funcionar como um sistema integrado em um único *chip*, este dispositivo combina um microcontrolador, conectividade *Wi-Fi* e *Bluetooth*, fundamentais para a aplicação em questão. O núcleo da série ESP32 é um processador *Tensilica Xtensa LX6*, disponível em duas versões *dual-core* (CPU com dois núcleos) e uma *single-core* (CPU com único núcleo). A versão *dual-core* foi escolhida para o projeto.

Possui uma configuração que incorpora uma antena RF de 2.4GHz com amplificadores de potência, o que permite que ele possua comunicação *Wi-Fi* e *Bluetooth* com o mesmo *chip*. Este componente é desenvolvido pela empresa chinesa *Espressif Systems*, sediada em Xangai e o ESP32 é fabricado pela *foundry TSMC* por meio de seu processo de fabricação de 40 nm (ESPRESSIF, sd).

Dentre as características do microcontrolador que são relevantes para o projeto estão o seu processador de 32bits com velocidade de *clock* configurável de 160 até 240MHz e dois

núcleos com arquitetura simétrica, possuindo conectividade *Wi-Fi* 802.11 b/g/n e *Bluetooth* v4.2 BR/EDR e BLE, com até 18 canais ADCs de 12 bits de resolução, 2 DACs de 8 bits, possui sensor de temperatura interno, possui 2 interfaces I2C utilizadas pelo sensor de posição magnético, possuir 3 interfaces UART com possibilidade de gravação *firmware*, possui criptografia da memória *flash* gravada, garantindo segurança ao *firmware* escrito no micro, além de possuir um baixíssimo consumo de energia no estado de sono profundo, consumindo apenas 5uA e com função acordar via GPIO.

Uma vantagem de destaque para o uso deste microcontrolador é o fato dele possuir certificações de funcionamento pela ANATEL (Agência Nacional de Telecomunicações), que permite que ele seja utilizado em produtos com caráter comercial. Dentre as certificações, podem ser citadas:

- 1) **Certificação de RF:** ANATEL (Anatel ID: 02152-20-11541) / FCC / CE-RED / IC / TELEC / KCC / SRRC / NCC;
- 2) **Certificação Wi-Fi:** *Wi-Fi Alliance*;
- 3) **Certificação Bluetooth:** BQB;
- 4) **Certificação "Green":** REACH / RoHS;

Na Figura 16 é possível se visualizar o módulo ESP32-WROOM32E.

Figura 16 - Módulo ESP32-WROOM32E.



Fonte: ESPRESSIF, sd.

4. DESENVOLVIMENTO

Neste capítulo, serão apresentadas as etapas de desenvolvimento do projeto de *hardware*, desenvolvimento da placa de circuito impresso, o *firmware* desenvolvido para o microcontrolador utilizado no sistema e as definições da arquitetura da rede Modbus TCP/IP para comunicação entre os módulos.

Na parte de desenvolvimento do sistema de *hardware*, serão descritos os blocos eletrônicos definidos no projeto, subdivididos de acordo com a função de cada processo, sendo eles:

1. Sistema de gerenciamento e distribuição de energia;
2. Sistema microprocessador e conversor serial para atualização de *hardware*;
3. Sistema de controle do motor DC;
4. Sistema de comunicação com sensor de posição angular;
5. Sistema de armazenamento das variáveis de processo;
6. Sistema de interface com conexões externas;

Em seguida, será apresentado o processo de design e fabricação das placas de circuito impresso, também conhecidas como PCB (do inglês *Printed Circuit Board*). Esta seção descreverá o processo de *layout* da placa, que é essencial para a materialização dos blocos de *hardware* em um circuito funcional.

Na sequência, será apresentado o *firmware* embarcado nos módulos, abordando os aspectos da programação de microcontroladores, incluindo tópicos específicos para a programação do microcontrolador utilizado, destacando as técnicas e padrões utilizados para o desenvolvimento e manutenção do *firmware* desenvolvido.

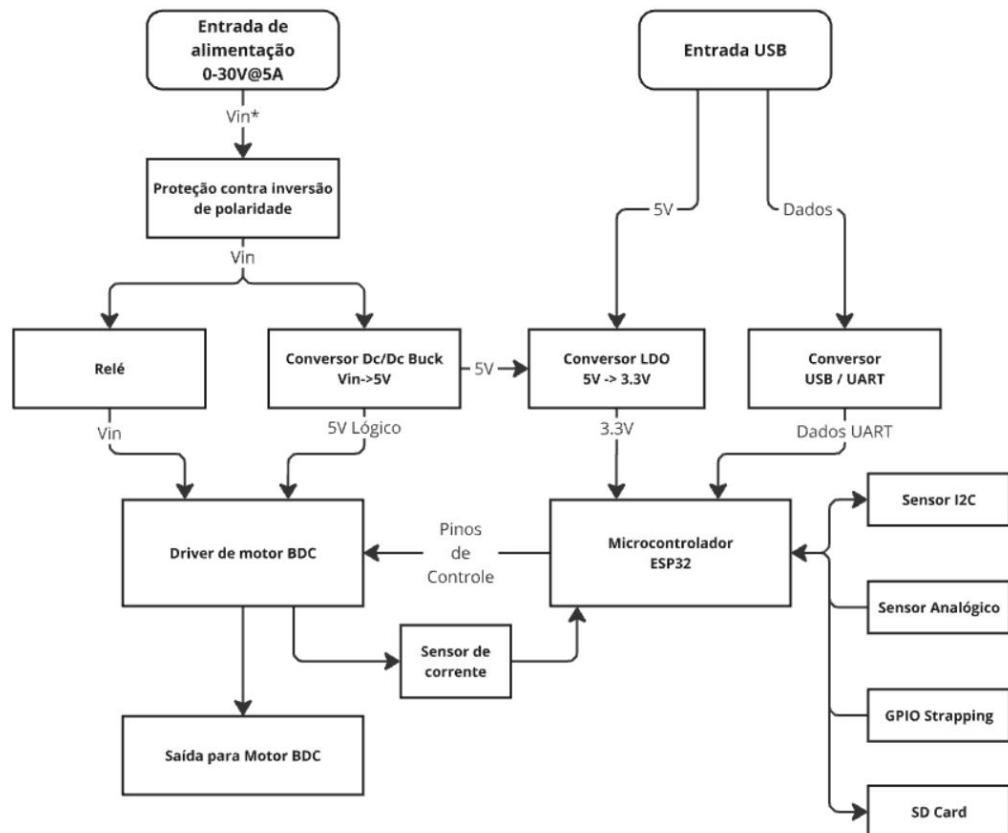
E por fim, na será apresentado a arquitetura da rede, onde serão discutidos aspectos relacionados à conexão entre os módulos utilizando a rede Modbus TCP/IP e como se espera que esses módulos interajam entre si a fim de se respeitar a arquitetura de rede escolhida.

4.1 Sistema de Hardware

Para a criação do *hardware* desenvolvido, foi necessário realizar uma etapa de idealização e ela esta apresentada no fluxograma apresentado na Figura 17, que ilustra a interconexão entre os principais componentes do projeto. O fluxograma oferece uma visão

mais detalhada do fluxo de informações e energia, destacando a integração dos diversos sistemas que compõem a solução e é possível identificar rapidamente onde ocorrem interações críticas entre componentes, facilitando a análise de desempenho e a identificação de possíveis falhas.

Figura 17: Fluxograma de Hardware.



Fonte: Próprio autor.

4.1.1 Sistema de gerenciamento e distribuição de energia

O sistema de gerenciamento de energia é a parte responsável pela entrada da energia no sistema e distribuição para os subsistemas. O sistema de energia foi pensado para ser condicionado para as seguintes funções:

- Proteger os subsistemas contra inversão de polaridade;
- Alimentar o sistema microcontrolador;
- Alimentar o sistema de controle do motor;
- Alimentar os sensores externos e periféricos.

Sendo assim, existirão diferentes nível de tensão dentro do sistema, sendo necessário algumas etapas de retificação da tensão, passando por alguns reguladores lineares de tensão e conversores de nível DC/DC rebaixadores de tensão, conhecidos como conversores *Buck*.

4.1.1.1 Sistema de proteção contra inversão de polaridade

Para aumentar a robustez e a segurança do sistema, foi implementado um mecanismo de proteção contra a inversão de polaridade na entrada de alimentação. A maneira mais simples de realizar essa proteção é através da inserção de um diodo diretamente polarizado na entrada do circuito, que impede uma conexão incorreta de polaridade. No entanto, essa abordagem apresenta algumas limitações, como a queda de tensão no diodo e a dissipação de calor devido à perda de potência, dada pela Equação 4.1.

$$P = V_f \cdot I \quad (4.1)$$

Onde P é a potência dissipada em forma de calor, V_f é a tensão direta no diodo e I é a corrente que é percorrida pelo diodo.

Em uma situação limite de operação, limitada pela corrente máxima de operação do circuito de ponte H sendo igual a 6A, considerando um diodo retificador comum com V_f igual a 0,6V, tem-se uma dissipação de 3,6W em forma de calor, que reduz consideravelmente a eficiência energética do sistema.

A alternativa adotada foi com a utilização de um transistor MOSFET na entrada do circuito como chave eletrônica que só poderia ser acionado com a polarização correta do circuito, bloqueando em casos de inversão de polaridade. Para garantir o correto acionamento do sistema de proteção, foi utilizado um MOSFET do tipo *P-Channel*, e alguns parâmetros essenciais foram verificados na folha de dados do componente que incluem:

1. Os valores de V_{GS} (tensão *Gate-Source*), que é a tensão mínima para se realizar o acionamento do transistor. Neste aspecto, fora adotado uma solução adicional que envolve o uso de um diodo *Zener* para limitar a tensão entre o *Gate* e a *Source*, possibilitando se escolher um MOSFET com V_{GS} menor.
2. A resistência parasita R_{DSon} , que representa a resistência interna do MOSFET quando ele está ativo e em condução. De acordo com a lei de Joule (4.2), quanto maior essa resistência, maior será a potência dissipada em forma de calor.

$$P = R_{DSon} \cdot I^2 \quad (4.2)$$

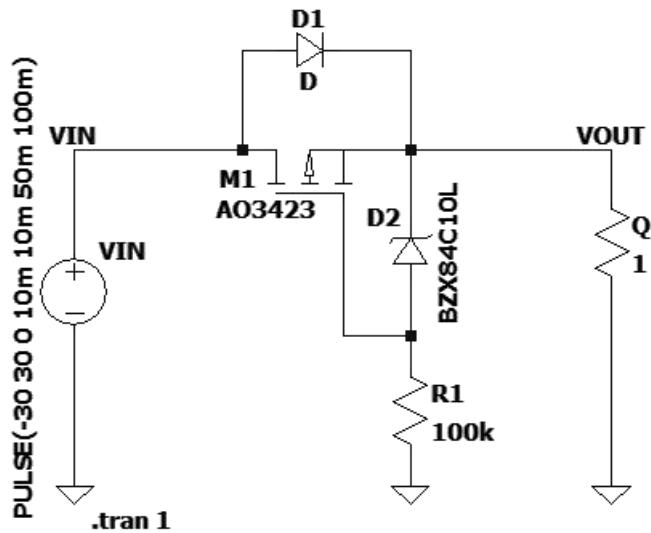
3. A tensão V_{DS} (tensão *Drain-Source*) que deve ser superior à tensão de alimentação. O valor V_{DS} determina a máxima diferença de potencial permitida entre o *Drain* e a *Source*.
4. O valor da Resistência térmica de junção R_{ja} e o valor máximo de temperatura de operação T_j , valores responsáveis pelo cálculo de dissipação de calor, avaliando a necessidade de se utilizar um dissipador de calor caso o valor de dissipação do MOSFET não seja suficiente para a dissipação de calor. A fórmula para calcular a potência máxima dissipada pelo MOSFET sem o uso de um dissipador (P_d) é dada pela Equação 4.3.

$$P_d = \frac{T_j - T_a}{R_{ja}} \quad (4.3)$$

Onde P_d é a potência máxima dissipada pelo MOSFET sem dissipador, T_j é a temperatura de junção máxima permitida, T_a é a temperatura ambiente e R_{ja} é a resistência térmica de junção com o ambiente. Todos valores utilizados estão disponíveis na folha de dados do MOSFET utilizado. A Equação 4.3 descreve a capacidade do MOSFET de dissipar calor para o ambiente.

A fim de se validar o sistema de proteção contra inversão de polaridade para se desenvolver, foi usado o simulador de eletrônica *LTS spice* para se simular o arranjo dado pela Figura 18.

Figura 18 - Simulação do circuito de proteção contra inversão de polaridade utilizando *LTS spice*.



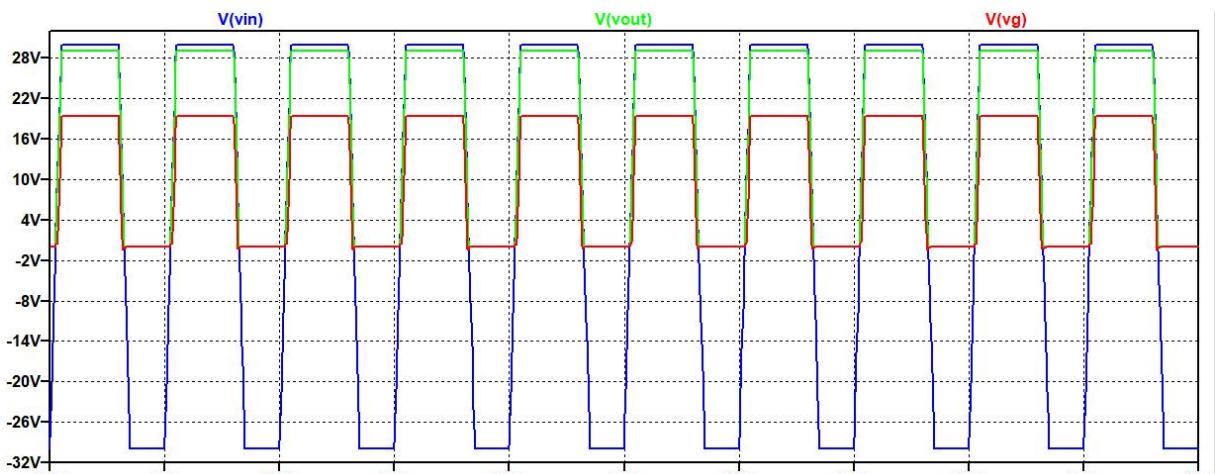
Fonte: Próprio autor.

Para a simulação foi usado um MOSFET *P-channel* com as características descritas na Tabela 4. Para o diodo Zener aplicado no *Gate* do MOSFET a fim de causar uma pequena

queda de potencial e garantir o correto acionamento do MOSFET foi utilizado o Diodo BZX84C1O com tensão *zener* de 10V.

Para a simulação fora usado uma fonte de tensão de -30 a 30V com onda cíclica com período de 100ms e um *duty-cycle* de 50%. Para a validação, foi medido a tensão de entrada na fonte (*vin*) e a tensão de saída em uma resistência que simula uma carga de 1Ohm para medir a tensão de saída (*vout*), além disso, foi feito a medição da tensão de *Gate* do MOSFET (*vg*) para avaliar os momentos de ativação da chave (vide Figura 19).

Figura 19 - Resultado da simulação do circuito de proteção contra inversão de polaridade.

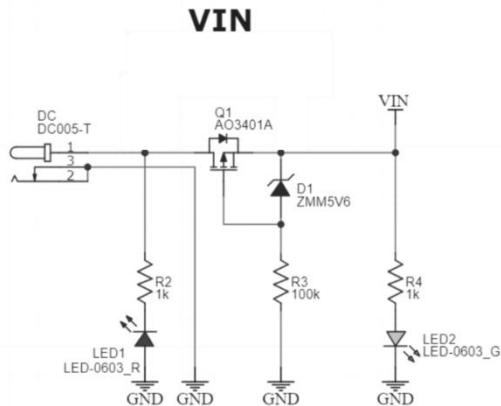


Fonte: Próprio autor.

Como mostrado na Figura 19 pode ser visto que a tensão de saída é bloqueada quando a tensão de entrada se torna negativa, o que simula uma inversão de polaridade, mas não bloqueia quando a tensão de entrada esta positiva, simulando a polarização direta esperada pelo circuito. Avaliando a tensão de *Gate* do MOSFET, pode-se confirmar que a chave não é ativa em tensões de entrada negativas. Dessa forma o sistema de proteção se mostra eficaz contra inversões de polaridade não esperadas, bloqueando a tensão de entrada no circuito, evitando queima de circuitos de forma não prevista.

Para a implementação do sistema de proteção fora usado o seguinte arranjo mostrado na Figura 20 e na Tabela 3 é feito um comparativo entre os componentes usados na simulação e os componentes usados no circuito real. Além disso, o circuito conta com um par de LED (do inglês *Light Emissor Diode*) identificados como LED1 e LED2, onde o primeiro, da cor vermelha, sinaliza a alimentação com polaridade invertida, servindo para sinalizar ao usuário do sistema que a fonte de alimentação não esta de acordo com o esperado. Já o LED2, de cor verde, sinaliza a correta polarização do sistema e indicando que ele esta ativo e funcionando corretamente.

Figura 20 - Esquemático do circuito de proteção contra inversão de polaridade utilizado.



Fonte: Próprio autor.

Tabela 3 - Comparação entre os componentes utilizados na simulação e no circuito real.

Nome	Componente Simulado		Componente Real		Diferença
MOSFET <i>p-channel</i>	Modelo	AO3423	Modelo	AO3401A	-
	VGS	$\pm 12V$	VGS	$\pm 12V$	-
	RDSon	$166m\Omega$	RDSon	$85m\Omega$	$81m\Omega$
	VDS	-20V	VDS	-30V	-10V
	Pd	0.9W	Pd	0.9W	-
Diodo Zener	Modelo	BZX84C1OL	Modelo	ZMM5V6	-
	Vd	10V	VD	5.6V	4.4V

Fonte: Próprio autor.

4.1.1.2 Sistema rebaixador de tensão - Conversor Buck

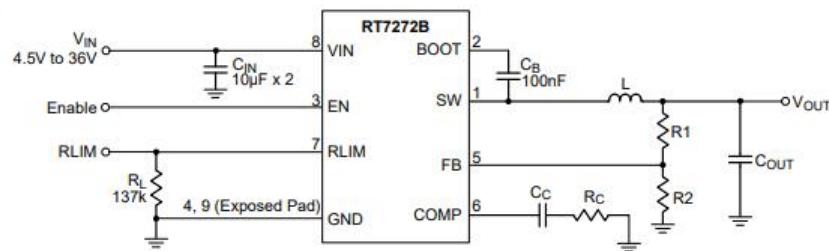
Como parte do sistema de gerenciamento de energia, a tensão de entrada foi projetada para atuar na faixa de 4,5 a 30V devendo passar por uma etapa de retificação, onde a tensão será condicionada para 5V afim de assegurar a tensão de nível lógico da ponte H e alimentar o circuito do sistema microcontrolador. Para realizar essa etapa de conversão de energia com diferenças que podem chegar a 25V, fora escolhido o componente RT7272BGSP, um conversor DC/DC rebaixador de tensão, também conhecido como conversor *Buck*.

O RT7272B é um conversor que possui uma alta eficiência de conversão, chegando na casa dos 95% podendo fornecer até 3A de corrente. O componente integra uma meia ponte utilizando MOSFET do tipo N com um *high side* de $150m\Omega$ e um *low side* de $80m\Omega$. Além

disso o componente possui *softstarter*, uma característica importante para evitar sobre tensões na hora da alimentação, que poderia acarretar na queima de componentes sensíveis a picos de tensão, como o caso do microcontrolador e o dispositivo de controle do motor. O conversor possui proteções contra sobre e sub tensão, além de bloquear seu funcionamento no caso de curtos circuitos de saída.

Na Figura 21 pode ser visto o arranjo aconselhado pela folha de dados do componente, seguido tabela de valores dos componentes sugeridos para cada nível de tensão desejado pelo *Buck*.

Figura 21 - Esquema de ligação aconselhado para o circuito Buck.

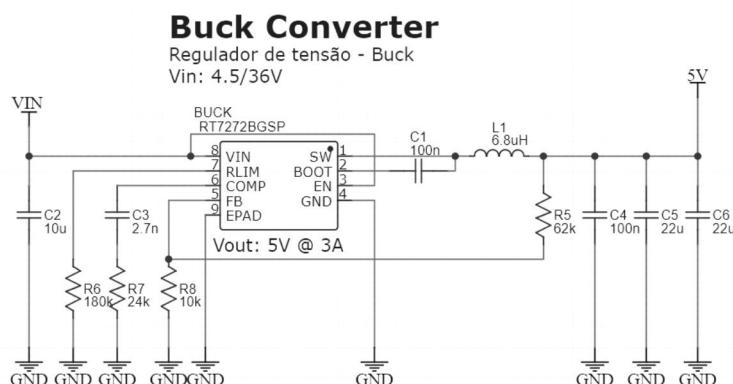


V _{OUT} (V)	R ₁ (kΩ)	R ₂ (kΩ)	R _c (kΩ)	L (μH)	C _c (nF)	C _{OUT} (μF)
12	47	3.35	47	10	2.7	22 x 2
8	27	3	36	8.2	2.7	22 x 2
5	62	11.8	24	6.8	2.7	22 x 2
3.3	75	24	16	4.7	2.7	22 x 2
2.5	25.5	12	12	3.6	2.7	22 x 2
1.2	30	60	6.8	2.2	2.7	22 x 2

Fonte: Folha de dados do componente.

Já no arranjo da Figura 21, a tensão de saída do conversor será de 5V com capacidade de entregar até 15W e os valores dos componentes para essa tensão de saída são dados pelo folha de dados do RT7272B.

Figura 22 - Arranjo do conversor Buck utilizado no projeto.



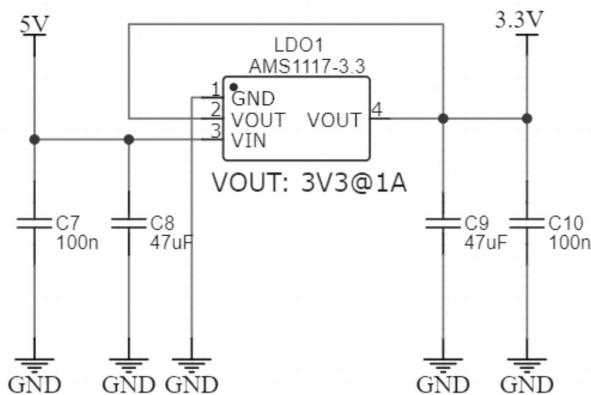
Fonte: Próprio autor.

A tensão de 5V será utilizada como tensão de nível lógico para o circuito de controle dos motores, alimentação dos sensores periféricos e do microcontrolador do sistema.

4.1.1.3 Circuito retificador

O microcontrolador possui uma tensão de alimentação de 3,3V, por esse motivo, antes de ser alimentado, a tensão será retificada novamente através de um retificador linear de tensão LDO (do inglês *Low Dropout Regulator*) dado pela Figura 23.

Figura 23 - Circuito retificador LDO de 3.3V.



Fonte: Próprio autor.

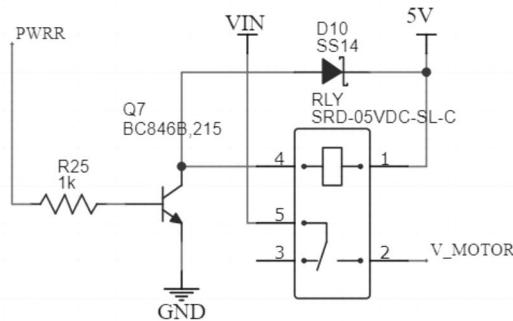
A escolha do retificador linear de tensão AMS1117-3.3 se deu pelo valor do dispositivo e a qualidade de retificação, além de necessitar de poucos componentes externos para funcionar. De acordo com a folha de dados, o regulador é capaz de entregar até 1A com tensão de saída de 3,3V, sendo suficiente para alimentar o microcontrolador e servir de tensão de nível lógico para os sensores periféricos.

4.1.1.4 Circuito de proteção eletromecânico auxiliar

Além da alimentação dos dispositivos, também fora previsto um sistema eletromecânico para alimentação do motor do módulo. Essa alimentação servirá como um sistema de proteção para desligamento do sistema em casos de emergências, detecção de falhas do sistema, sobre-corrente ou inversão de polaridade, uma vez que o relé só será ativado via *firmware* do microcontrolador. O esquemático do sistema de alimentação do motor pode ser visto na Figura 24.

O circuito é um arranjo básico para ativação de um relé SRD-05VDC que possui uma tensão de operação de 5V através da energização da bobina dada pelos terminais 1 e 4 da Figura 24. Para ser possível fazer a ativação do relé, fazendo uso de um microcontrolador, fora utilizado um transistor bipolar de junção BC846B, que fará a amplificação do sinal de saída do microcontrolador.

Figura 24 - Sistema de acionamento eletromecânico da alimentação do motor.



Fonte: Próprio autor.

Através da folha de dados do relé, é possível se estimar o valor de corrente mínima necessário para se ativar a bobina, além de se obter informações de resistência do enrolamento da bobina. Com esse valor em mãos, fora possível se calcular o valor da corrente de coletor emissor da chave, a fim de se dimensionar o BC845.

4.1.2 Sistema microprocessador e conversor serial para atualização de hardware;

A parte do sistema microcontrolado é o cérebro de toda operação, sendo a figura crucial para se realizar um sistema de controle da posição do motor e realizar a comunicação com os outros módulos utilizando o protocolo *Modbus* adotado neste projeto. Na aplicação, o microcontrolador possui as responsabilidades de:

- 1) Realizar os cálculos do algoritmo de controle do motor DC através do sensoriamento da posição com o sensor magnético em malha fechada;
- 2) Realizar a comunicação com o mestre Modbus TCP/IP e interfaces externas para relatar os estados do sistema e ser configurado de acordo com as necessidades de operação;
- 3) Realizar o acionamento do motor DC;
- 4) Realizar o sensoriamento do sensor magnético AS5600;
- 5) Realizar a gravação dos dados de operação em memória removível;
- 6) Otimização da energia do sistema;

7) Detecção de falhas do sistema.

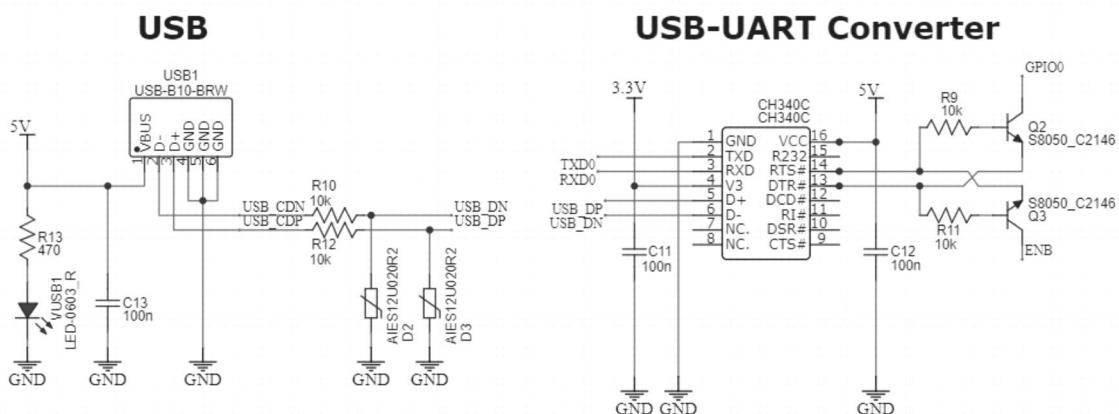
Contudo, para que o sistema microcontrolado possa realizar essas operações de forma eficaz, é imprescindível realizar o *upload* do *firmware* que será executado pelo microcontrolador, seguindo as etapas e procedimentos adequados.

Alguns métodos comuns de *upload* em microcontroladores são através de JTAG (do inglês *Joint Test Action Group*), SWD (do inglês *Serial Wire Debugging*) ou USB DFU (do inglês *Device Firmware Upgrade*). Comummente, o ESP32-WROOM32E utilizado no projeto, realiza o processo de *upload* através do uso de um módulo de programação USB para UART, como o popular módulo FTDI (do inglês *Future Technology Devices International*), sendo a última abordagem a utilizada.

Para realizar essa etapa de conversão do padrão USB para o UART, é utilizado um componente especializado nisso, o CH340C. Esse componente funciona como um MODEM (modulador e de-modulador) na comunicação USB para o UART, tendo uma comunicação *full-duplex* e com capacidade de transmissão de até 115200bps. Através do CH340C é possível não apenas realizar o *upload* do *firmware*, mas também utilizá-lo para *debug*, estabelecendo uma comunicação *serial* entre o dispositivo *host* USB e o sistema.

Na Figura 25 é possível visualizar a ligação entre o conector micro USB tipo B e o conversor CH340C que fará a interface com o microcontrolador.

Figura 25: Interface USB-UART e conversor CH340C.



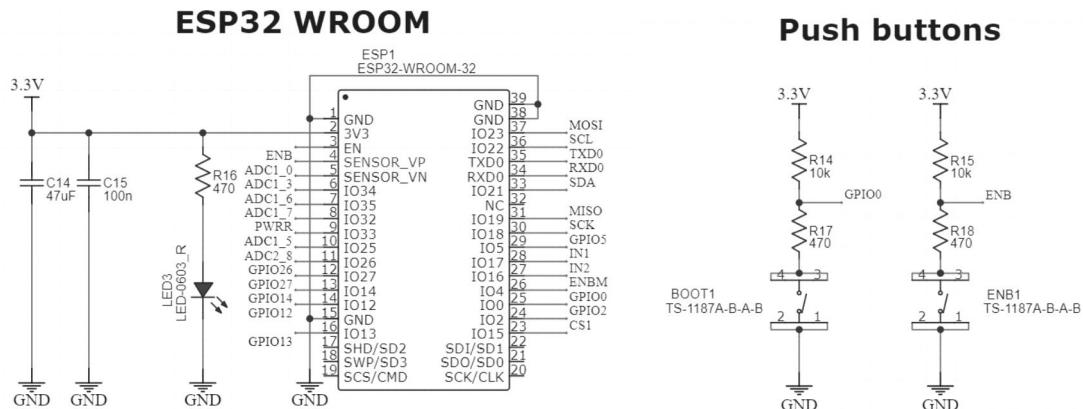
Fonte: Próprio autor.

O CI CH340C foi escolhido pois o microcontrolador ESP32-WROOM32E possui *strapping pins* que definem alguns estados de configuração inicial do ESP quando energizado. O microcontrolador possui no total 6 *strapping pins*, porém somente dois foram utilizados no projeto devido às necessidades, sendo eles: GPIO0; GPIO2.

Ao inicializar o ESP32-WROOM32E com o pino digital GPIO0 em nível lógico baixo e o pino digital GPIO2 também em nível lógico baixo, o microcontrolador entra em estado de programação, permitindo que seja transferido ou atualizado um novo *firmware* para ele. O *chip* CH340C possui dois pinos chamados de RTS (do inglês *Request to Send*) e DTR (do inglês *Data Terminal Ready*) que estabelecem um controle do fluxo de dados transmitidos na hora de se fazer o *upload* do *firmware* no microcontrolador. Através dos pinos de controle, o *chip* CH340C consegue satisfazer as condições de gravação do ESP32-WROOM32E ativando e desativando os transístores que operam como chaves *pull-up* e *pull-down* e inicializar a transmissão novo *firmware* de forma automática, trazendo uma maior praticidade ao sistema para se realizar atualizações de *firmware* via USB.

Além das conexões de *strapping pins*, na Figura 26 pode-se observar as conexões realizadas no *chip* do microcontrolador. O sistema conta também com dois botões *push-pull* que possuem as funções de reiniciar o sistema e levar o pino digital GPIO0 a nível lógico baixo para ser usado em gravações sem o uso do USB.

Figura 26: Conexões do microcontrolador ESP32-WRROM32 e botões *push-pull* de controle.



Fonte: Próprio autor.

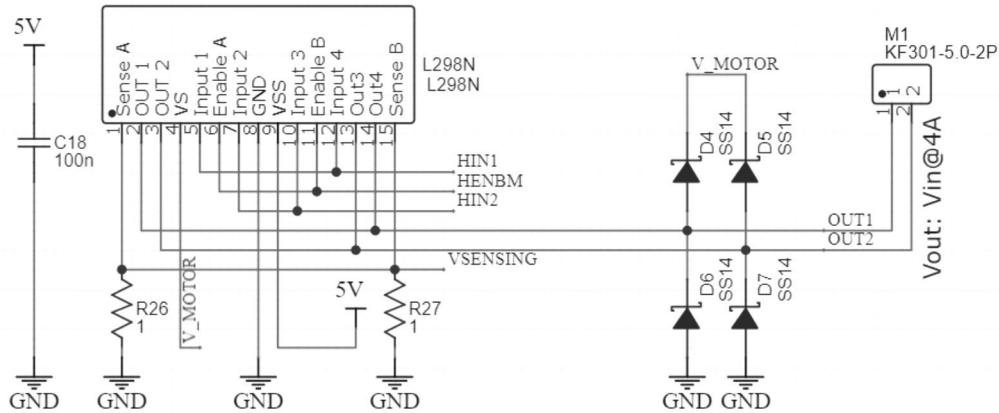
4.1.3 Sistema de controle do motor DC

Para o sistema de operação do motor DC que poderá ser utilizado no sistema, foi utilizado o *driver* de controle de motores de corrente contínua L298N. As conexões realizadas para o correto funcionamento do motor pode ser visto na Figura 27 e foi baseado nas informações contidas na folha de dados do componente.

O L298N pode ser alimentado com até 46V e possui um nível lógico para atuação ideal de 5V, sendo capaz de fornecer até 4A de corrente. Ele utiliza duas entradas de nível lógico para controle de direção de giro do canal 1 assinalados pelo nome IN1 e IN2 e do canal

2 assinalados pelo nome IN3 e IN4. Além disso, utiliza uma entrada PWM para o controle de velocidade para cada canal do motor assinalado pelos nomes ENA para o canal 1 e ENB para o canal 2. Além disso possui os terminais 1 e 15 dedicados para medição de corrente de saída.

Figura 27: Conexões do *driver* de controle de motor DC L298N.



Fonte: Próprio autor.

Para atingir maiores correntes de operação, o fabricante fornece um esquema de conexão que ativa ambos canais em paralelo, conectando os canais IN1 a IN3 e IN2 a IN4, além de conectar os pinos PWM ENA com ENB. Nessa configuração, os canais de saída se tornam únicos e operam em paralelo atuando na mesma carga. A Tabela 4 mostra o mapa lógico dos estados dos canais paralelizados de acordo com a entrada dos pinos IN1, IN2 e ENA.

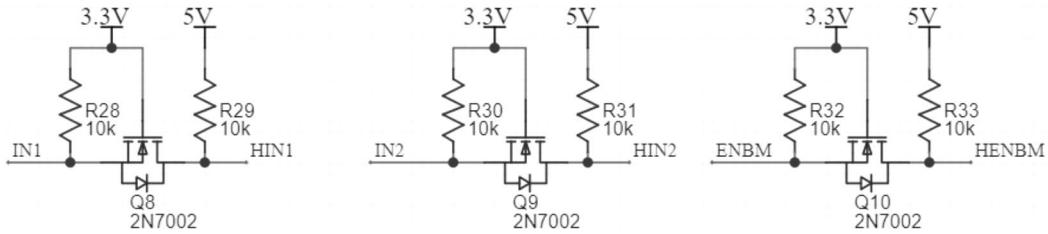
Tabela 4 - Mapa lógico de acionamento do CI L298N na configuração paralelo.

Entradas		Estado
ENA = ALTO	IN1 = ALTO & IN2 = BAIXO	Sentido horário
	IN1 = BAIXO & IN2 = ALTO	Sentido anti-horário
	IN1 = IN2	Frenagem
ENA = BAIXO	IN1 = X & IN2 = X	Roda livre

Fonte: Próprio autor.

Para o correto funcionamento da ponte-H, de acordo com a folha de dados do componente, ele deve operar com tensão lógica de 5V, no entanto, o microcontrolador utilizado possui uma tensão de nível lógico de 3,3V, sendo assim, foi necessário utilizar um conversor de nível lógico bidirecional, também conhecidos como conversores LLC (do inglês *Logic Level Converters*). Na Figura 28 é apresentado o arranjo de um LLC que faz a interface entre o microcontrolador e a ponte-H, o primeiro operando a 3,3V enquanto o segundo opera a 5V.

Figura 28: Conversor de nível lógico LLC.



Fonte: Próprio autor.

4.1.4 Sistema de armazenamento das variáveis de processo

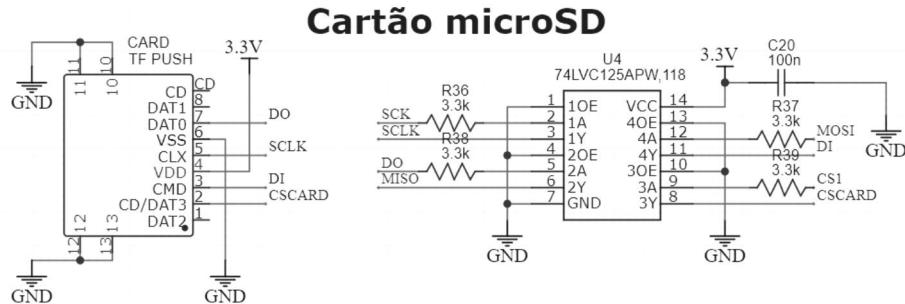
O microcontrolador possui a função de registrar os dados gerados pelo sistema no cartão SD, com o objetivo de criar um *data logger* e armazenar essas informações. A implementação desta função é facilitada pelo uso do protocolo SPI, que é amplamente utilizado na comunicação com cartões SD. A simplicidade dessa tarefa é assegurada pela compatibilidade do protocolo SPI com a maioria dos cartões SD, permitindo a integração e operação de maneira direta e eficiente.

A comunicação SPI é caracterizada por fazer uso de um pino de relógio para garantir o sincronismo na comunicação entre os periféricos, caracterizada pela sigla SCLK e dois pinos de dados que funcionam como dados de saída do mestre (controlador) para o escravo (módulo), e dados de entrada para o mestre e saída do escravo. Esses pinos são chamados respetivamente de pinos MISO (do inglês *Master In Slave Out*) e MOSI (do inglês *Mester Out Slave In*). Além disso, existe um pino auxiliar para ativar ou desativar o módulo, também chamado de seletor de *chip* ou *Chip Select* do inglês.

Para a instalação do cartão SD no sistema, foi utilizado um *slot* que serve como interface física para conexão do cartão com o sistema, onde através dele, as conexões são realizadas com o cartão até o microcontrolador. O cartão SD utiliza o protocolo SPI, que torna ele bastante fácil de ser utilizado. Além do *slot*, para garantir a integridade do cartão SD e evitar que falhas não previstas possam danificá-lo levando a perda de dados importantes, foi adotado o componente 74LVC125A que funciona como um *buffer* não inversor com o intuito de isolar o cartão SD do microcontrolador e dos demais circuitos da placa, tornando o cartão imune a ruídos ou surtos do sistema e também protegendo o microcontrolador contra dispositivos SD mal conectados ou com problemas de integridade.

Na Figura 29 é possível se observar as conexões necessárias para se conectar o *slot* do cartão SD com o dispositivo isolador que faz a interface com o microcontrolador através dos pinos SPI.

Figura 29: *Slot* de conexão para cartão SD e CI 74LVC125APW isolador.



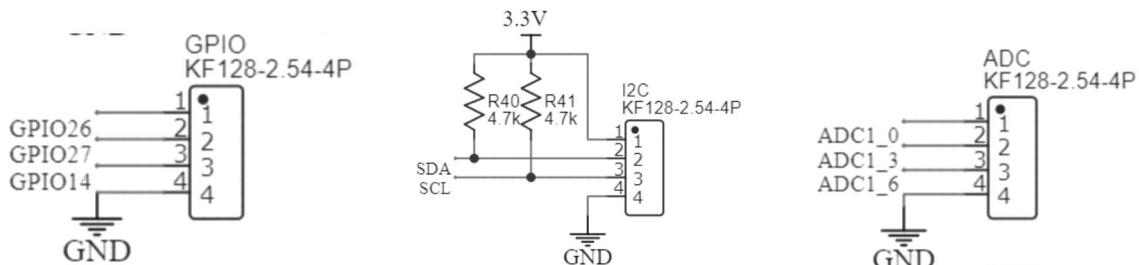
Fonte: Próprio autor.

4.1.5 Sistema de comunicação com sensor de posição angular;

Como o sistema fora projetado para ser genérico, ele conta com 3 conectores com o mundo físico externo para serem conectados a sensores angulares, podendo ser eles do tipo incremental através das conexões de GPIO, de potenciômetros através da conexão ADC ou utilizando *chips* dedicados para a medição de posição angular via I2C. Os três conectores são do tipo KFE com 2.54mm de *pitch* e foram escolhidos devido a facilidade de se realizar a conexão dos sensores com o conector. Além disso, os pinos são conectados em GPIO do microcontrolador podendo assumir outras funções além dos sensores de posição angular, servindo como sensores de chaves fim de curso ou conexões a sensores como temperatura ou pressão. Na Figura 30 é possível se visualizar as conexões nos conectores KFE.

Figura 30: Conectores para sensores ou GPIO.

CONECTORES



Fonte: Próprio autor.

4.2 Projeto da Placa de Circuito Impresso (PCB Design)

Após finalizada a etapa de definição de definição dos requisitos do sistema, simulações e projeto dos esquemáticos, pode-se então realizar a etapa de criação da placa de circuito impresso que iria embarcar todos os periféricos do sistema em uma única placa que servirá de módulo do sistema. Para a realização do *design* da PCB, fora utilizada a ferramenta de *design* de placas *EasyEDA*, uma ferramenta online e com versão gratuita de desenho de esquemáticos e *design* de placas.

Essa ferramenta conta com uma grande variedade de opções que visam facilitar o desenho de PCB, além fazer parte de uma linha de fabricação de PCB contando com empresas parceiras para produção das PCB pela JLCPCB e com uma das principais distribuidoras de componentes eletrônicos do mundo, a LCSC componentes. Dentre as principais vantagens do EasyEDA, podem ser destacados:

- 1) Possui ferramentas de desenho de esquemáticos;
- 2) Gerenciamento de *footprints* de componentes comerciais;
- 3) Auto roteamento e gestão de camadas de roteamento;
- 4) Simulação de circuitos *Spice*;
- 5) Gerenciamento de projeto com versionamentos;
- 6) Colaboração de times de projeto;
- 7) Integração com a linha produtiva de fabricação de PCI pela JLCPCB;
- 8) Possui *software online* e versão para *desktop*.

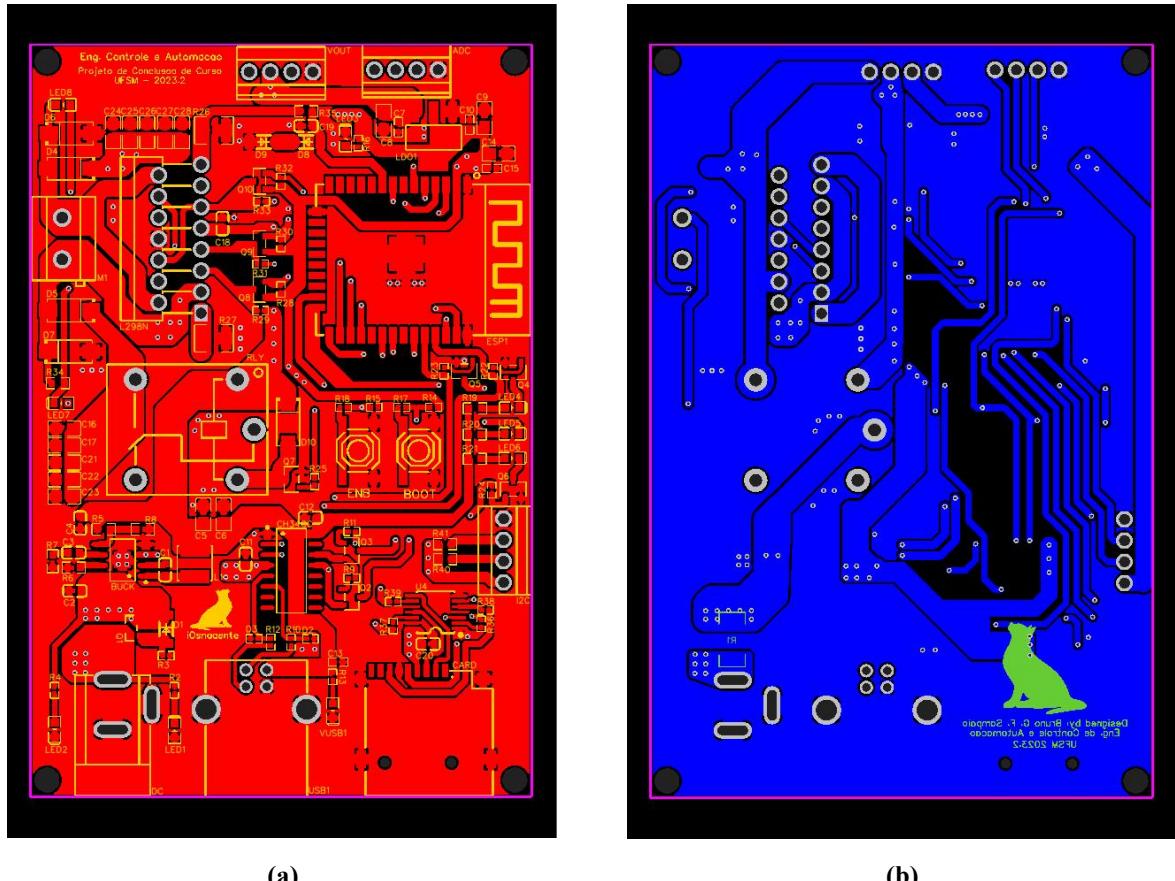
Essas são apenas algumas das vantagens que tornam o *EasyEDA* uma ferramenta atrativa para o desenho de PCB. Os esquemáticos resultantes do projeto podem ser encontrados em:

- a) Apêndice I - Esquemático do módulo / Gestão de energia do sistema.
- b) Apêndice II -Esquemático do módulo / Conversor USB - UART.
- c) Apêndice III - Esquemático do módulo - Microcontrolador.
- d) Apêndice IV -Esquemático do módulo / Circuito da ponte H.
- e) Apêndice V - Esquemático do módulo / *Data logger* do sistema.

Para realizar o roteamento da placa, foram utilizados duas camadas de cobre, onde a camada superior pode ser vista na Figura 31-a e camada inferior na Figura 31-b. As camadas podem ser diferenciadas pela cor que o *software* escolhe, sendo a camada inferior caracterizada pela cor vermelha e a camada inferior pela cor azul. Os furos são caracterizados

pela cor cinza e representam a transposição de uma camada para outra, também chamados de vias entre placas.

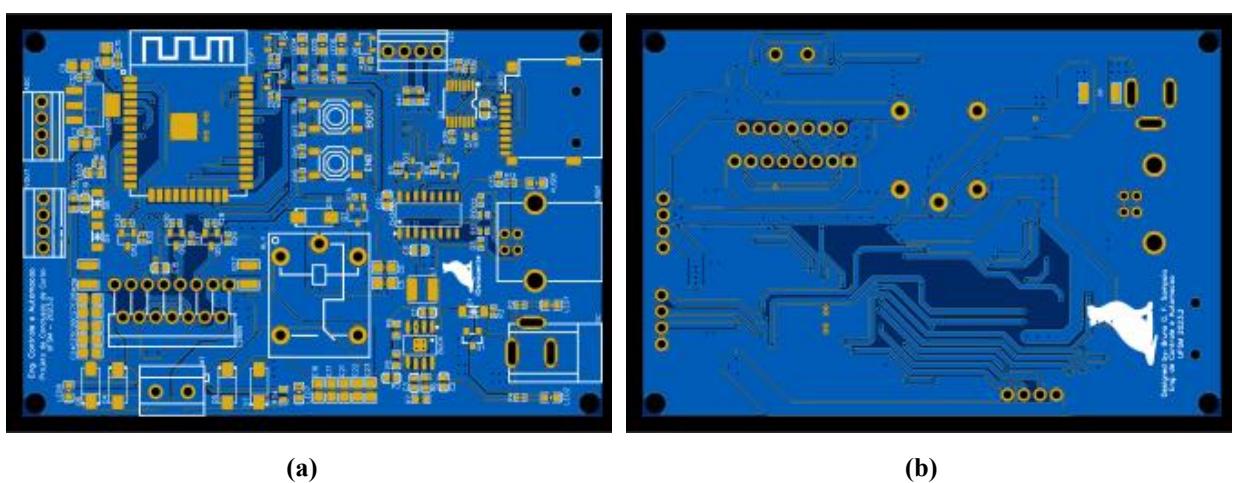
Figura 31 - Roteamento das conexões na PCB (a) Camada superior (b) Camada inferior.



Fonte: Próprio autor.

A visão final da PCB pode ser vista na Figura 32-a sendo a visualização 2D em CAD da camada superior da placa e a Figura 32-b a visualização 2D em CAD da camada inferior da placa.

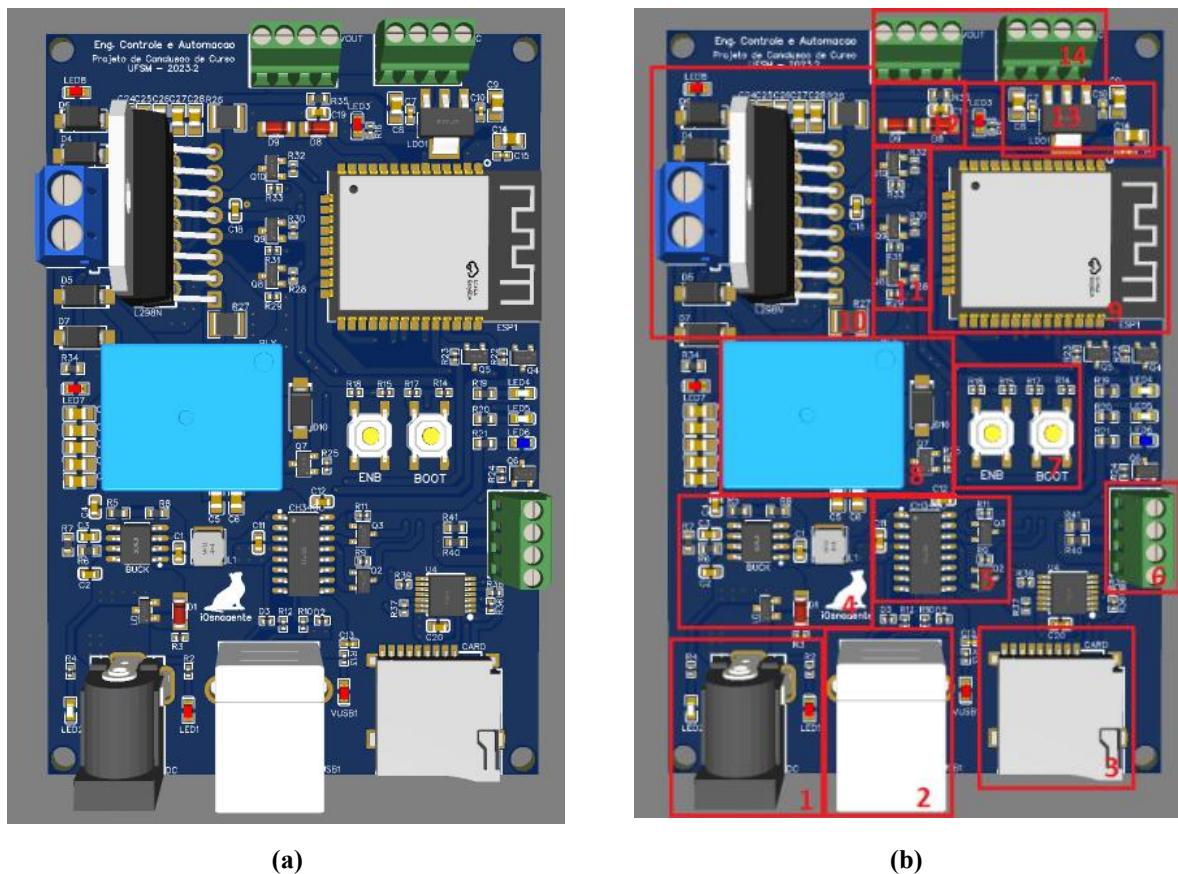
Figura 32 -Visualização 2D em CAD da PCB projetada. Vista (a) superior (b) inferior.



Fonte: Próprio autor.

Além disso, o programa permite que seja feito uma visualização 3D em CAD, para se estimar como será a placa com os componentes finais. Essa visualização pode ser vista na Figura 33-a.

Figura 33 - Visualização 3D em CAD da PCI projetada. (a) Vista superior (b) com secções em destaque e numeradas.



Fonte: Próprio autor.

Na Figura 33-b estão marcados e enumerados os setores da placa que correspondem aos itens descritos na Tabela 5.

Tabela 5 - Identificador das secções da Figura 26-b.

(continua)

Setor	Descrição
1	Conektor DC para entrada de energia do sistema 6 a 36V @ 6A.
2	Conektor USB B para entrada de dados no sistema e atualização de <i>Firmware</i> .
3	Slot para cartão micro SD para o <i>data logger</i> .
4	Conversor DC/DC rebaixador de tensão Buck para 5V.
5	Conversor USB para UART. Responsável pela atualização do <i>firmware</i> via USB B.
6	Conektor Borne 4 vias para interface I2C dos sensores de posição AS5600.

7	Botões de <i>Boot</i> e <i>Reset</i> do microcontrolador.
8	Relé de <i>by-pass</i> da energia do conector DC para o motores via ponte-H.
9	Microcontrolador ESP32-WROOM32E
10	Ponte-H L298N, diodos de roda livre e conector para conexão do motor DC.
11	Conversores de nível lógico 3,3V do microcontrolador para 5V da ponte-H.
12	Sensor de corrente.
13	Regulador de tensão 3,3V para alimentação do microcontrolador.
14	Conectores de saída para Tensão e GPIOs com ADCs integrados.

Fonte: Próprio autor.

4.3 Projeto de firmware

Para o projeto de *firmware*, foi utilizado o *framework* de desenvolvimento ESP-IDF desenvolvido pela empresa Espressif e fora projetado para os SoCs (do inglês *System-on-Chip*) ESP32. Ela oferece um SDK completo para o desenvolvimento de aplicativos genéricos nessas plataformas, utilizando linguagens de programação como C e C++. A ESP-IDF fora escolhida pois possui código aberto e é utilizada em milhões de dispositivos, permitindo o fácil acesso aos códigos de desenvolvimento. Além disso, ela suporta diversos componentes para serem utilizados com os periféricos do microcontrolador, incluindo RTOS, *drivers* de periféricos, pilha de rede, implementações de protocolos de rede e utilitários para casos de uso comuns. Isso facilita a lógica de negócios do *firmware*, pois o SDK fornece a maioria dos blocos de construção necessários para o uso do microcontrolador.

Para utilizar a ESP-IDF, fora utilizado a interface de programação amplamente conhecida do *Visual Studio Code*, através de um *pluggin* chamado *Platformio*, um ambiente de desenvolvimento integrado para a programação de microcontroladores de diversos fabricantes e dentre eles, os microcontroladores da Espressif. Esse ambiente de programação fora escolhido, pois ele suporta a ESP-IDF sendo um dos canais de divulgação de uso do *framework*.

Para garantir a flexibilidade e a adaptabilidade do *firmware*, foi adotada uma metodologia baseada em interfaces, módulos e controladores na arquitetura. Essa abordagem modular permite que os componentes do sistema, como sensores e atuadores, sejam facilmente intercambiáveis, facilitando futuras expansões ou modificações sem a necessidade de uma reestruturação completa do sistema. Esse padrão no *firmware* explora algumas teorias

da programação orientada a objetos da linguagem de programação C++, implementando modelos bem definidos de serviços como atuadores, sensores e relógios.

As interfaces são classes abstratas que possuem as definições dos sistemas permitidos (motores, sensores, relógios e comunicação) de modo a definir os contratos ou conjuntos de métodos que os módulos devem implementar a fim de padronizar as definições de cada parte do sistema. Não cabe as interfaces fazerem as definições específicas de cada módulo, mas sim a de restringir um conjunto mínimo necessário para que cada módulo possa funcionar de acordo com o esperado. Esse *design* promove a flexibilidade e sobretudo a intercambiabilidade de componentes, permitindo que diferentes implementações possam ser substituídas sem alterar o restante do sistema.

Os módulos por sua vez, são classes de implementação específicas para cada dispositivo que possa ser utilizado e implementam as definições das interfaces. Os módulos foram projetados para serem independentes e autos suficientes, encapsulando funcionalidades específicas de cada parte do sistema, como a aquisição de dados de sensores, acionamento dos atuadores, contagem do tempo ou processamento das mensagens da comunicação.

Já os controladores, por sua vez, são as classes de implementação de partes do sistema, como por exemplo, a implementação da malha de controle de posição do motor, que possui um módulo de motor, um sensor e a malha de controle que relaciona os dois módulos. Os controladores gerenciam a comunicação e a coordenação entre os módulos, garantindo que os dados sejam corretamente trocados e as operações sejam sincronizadas de acordo com as necessidades do sistema.

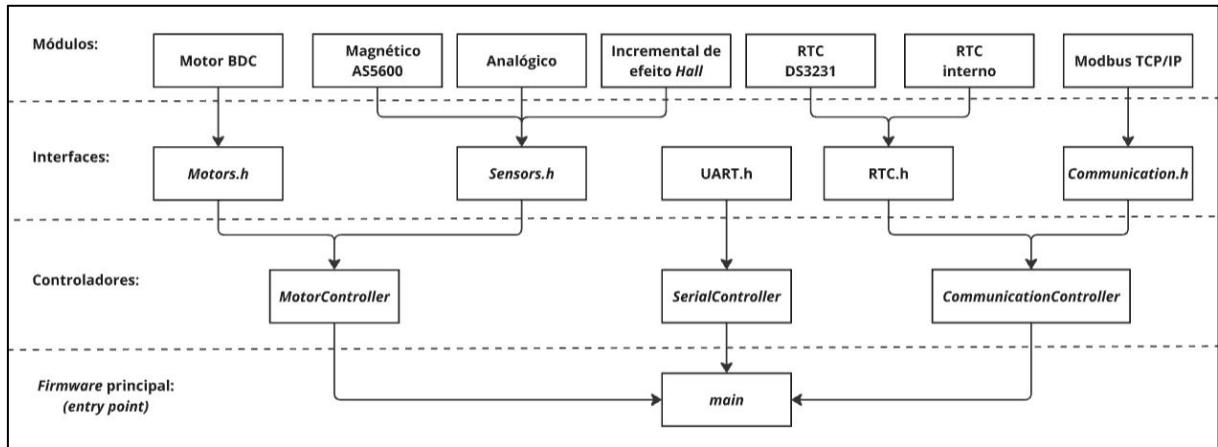
Com essa arquitetura, o sistema é capaz de integrar uma ampla gama de sensores e atuadores de diferentes fabricantes, desde que atendam às especificações das interfaces definidas. Isso proporciona uma grande versatilidade na escolha dos componentes, possibilitando a personalização do *hardware* para aplicações específicas.

Essa metodologia de *design* de *Firmware* é essencial para o desenvolvimento de sistemas robustos e escaláveis, permitindo a fácil atualização e adaptação do *hardware* sem comprometer a integridade do sistema como um todo. Na Figura 34 é apresentado as relações entre interfaces, módulos e controladores e suas relações dentro do sistema.

Além disso, o sistema de *firmware* desenvolvido para a placa eletrônica foi projetado para operar com o *FreeRTOS*, um sistema operacional de tempo real amplamente utilizado e suportado nativamente pelos microcontroladores ESP32 da Espressif. A escolha do *FreeRTOS* foi motivada por sua capacidade de gerenciar tarefas de forma independente,

proporcionando um controle preciso sobre a execução de múltiplos processos concorrentes dentro do microcontrolador, uma vez que o microcontrolador conta com dois núcleos de CPU.

Figura 34: Padrão Interface/Módulo/Controlador adotado.



Fonte: Próprio autor.

Um dos maiores benefícios do uso do *FreeRTOS* com o ESP32 é o escalonador de tarefas integrado no microcontrolador. O ESP32 possui suporte nativo para o *FreeRTOS*, o que significa que o sistema operacional pode aproveitar as capacidades de *hardware* do microcontrolador para gerenciar a execução de tarefas de maneira eficiente. O escalonador é responsável por determinar quais tarefas têm prioridade de execução, garantindo que tarefas críticas sejam atendidas conforme necessário.

Quando as duas abordagens: a utilização do padrão interface/módulo/controlador e o uso do *FreeRTOS*, são aplicadas em conjunto, elas se tornam extremamente complementares e poderosas. Essa combinação permite a criação de um sistema modular, flexível e eficiente, aproveitando o melhor das práticas de *design* de *firmware* e dos recursos de um sistema operacional de tempo real, uma vez que o escalonador integrado, garante periodicidade entre as tarefas em execução, definição de prioridades e permite a preempção temporal entre elas.

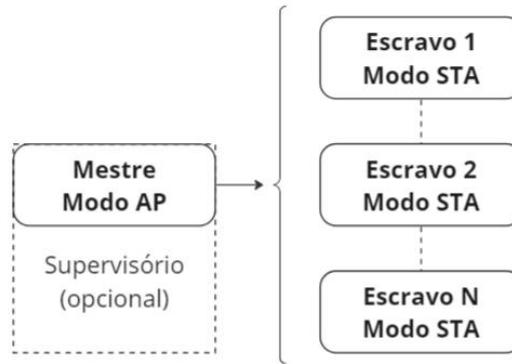
Para mais informações a respeito do projeto de *firmware*, o projeto pode ser encontrado no repositório do *GitHub* pelo link <https://github.com/iOsnaaente/TCC-Modulo-de-rastreamento-solar>.

4.4 Arquitetura de rede

Como o sistema será baseado em uma rede Modbus TCP/IP, é necessário se definir o comportamento de cada módulo de rastreamento solar perante o sistema como um todo. O sistema Modbus TCP/IP é baseado em um modelo de mestre escravo, sendo dessa

forma, necessário se haver dois tipos de dispositivos na mesma rede para se realizar a comunicação entre os sistemas. Na Figura 35 pode ser visto um modelo de arquitetura de rede que pode ser adotado no sistema.

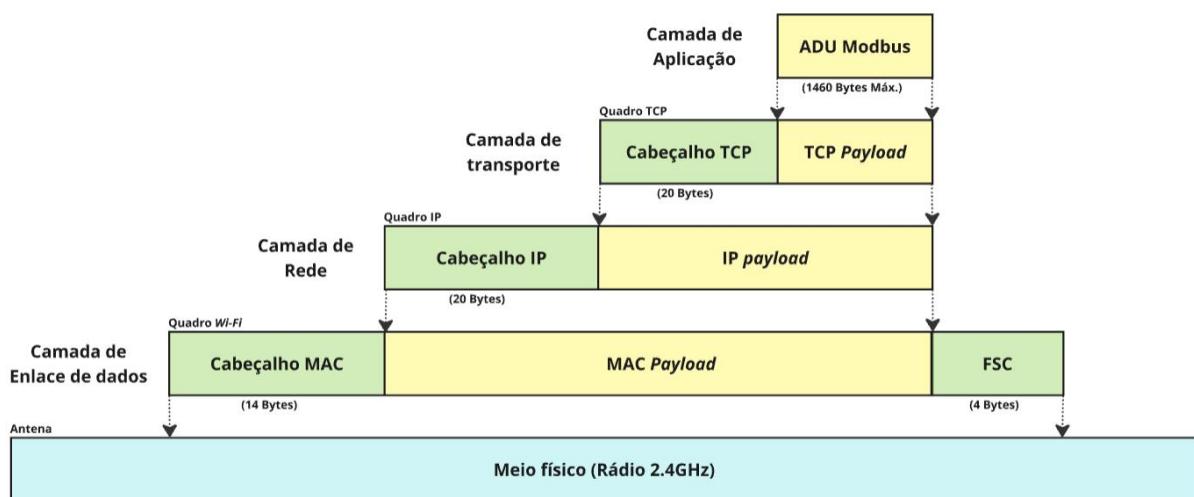
Figura 35: Arquitetura de rede mestre/escravo Modbus TCP/IP.



Fonte: Próprio autor.

O protocolo Modbus TCP/IP é um protocolo de aplicação definido sobre os protocolos TCP (*Transmission Control Protocol*) de rede e o IP (*Internet Protocol*) de transmissão e são usados juntos para transmitir dados de forma eficiente e confiável sobre a rede. O protocolo Modbus TCP/IP não define as camadas de acesso a rede, sendo essa, uma responsabilidade do microcontrolador ESP32, que define esse acesso através do *Wi-Fi* sobre o protocolo IEEE802.11, utilizando sua antena de 2.4GHz para estabelecer comunicação com os demais dispositivos. A Figura 36 mostra a relação dos pacotes e como um protocolo se relaciona com o outro.

Figura 36: Estrutura relacional dos protocolos empregados na comunicação Modbus TCP/IP.



Fonte: Próprio autor.

Os pacotes *Modbus* estão encapsulados no protocolo de comunicação TCP que garante o transporte desse pacote. O protocolo TCP é o responsável por definir a porta de

comunicação (neste caso, a porta *Modbus* padrão 502) e assegurar uma transmissão confiável dos dados. A camada de rede é dada pelo protocolo IP que gerencia o endereçamento e o roteamento dos pacotes entre os dispositivos de origem e destino, identificados por seus endereços IP. Finalmente, a camada física garante que os dados são transmitidos pelos respectivos endereços MAC, que faz a parte de enlace de dados para transmissão no meio físico.

No Modbus TCP/IP, a estrutura do pacote é dividida em *Application Data Unit* (ADU) e *Protocol Data Unit* (PDU), uma é responsável por carregar dados específicos do protocolo que carrega o *Modbus* (neste caso o TCP) e a outra sobre os dados *Modbus* propriamente dito.

A ADU envolve a PDU e adiciona um cabeçalho específico ao Modbus TCP, que facilita a comunicação na rede TCP/IP. Este cabeçalho inclui: o *Transaction Identifier* que é um identificador da transação, usado para diferenciar múltiplas solicitações simultâneas, o *Protocol Identifier* que define o protocolo utilizado, que é 0 para *Modbus*, o *Length* que indica o comprimento total dos campos subsequentes, ou seja, a PDU e o *Unit Identifier* que é um identificador do escravo no sistema *Modbus*, que é útil em redes que utilizam *gateways*.

Nesse contexto, o ESP32 utiliza o *Wi-Fi* para estabelecer uma comunicação sem fio e é responsável por gerenciar a transmissão e recepção dos pacotes Modbus TCP/IP sobre a rede. O ESP32 possui funções na ESP-IDF para implementar redes *Wi-Fi* via topologia *mesh*, que são malhas de comunicação, que permitem o dispositivo se comunicar com múltiplos dispositivos de forma descentralizada, retransmitindo os pacotes enviados sem um caminho específico, até que eles cheguem no seu destino correto, utilizando os protocolos de alto nível para endereçamento. Essa é uma abordagem válida para a aplicação, o que permitiria que os módulos fossem distribuídos em longas distâncias, utilizando apenas um mestre para múltiplos escravos.

Para a aplicação desenvolvida, foi utilizado uma abordagem mais simples do problema, utilizando uma topologia estrela, onde o dispositivo mestre assume a posição central e se conecta em todos os dispositivos escravos. Essa abordagem possui uma limitação de distância para as conexões entre os dispositivos, alcançando um raio limitado pela potência do ESP32.

Esse método de arquitetura de rede *Modbus* tratará o mestre como um servidor AP em uma rede *Wi-Fi*, no entanto, um das premissas do modelo de comunicação mestre/escravo, o mestre deve conhecer os endereços dos escravos para poder enviar requisições a eles, portanto, se o mestre não conhece os endereços, ele não pode se comunicar com eles diretamente.

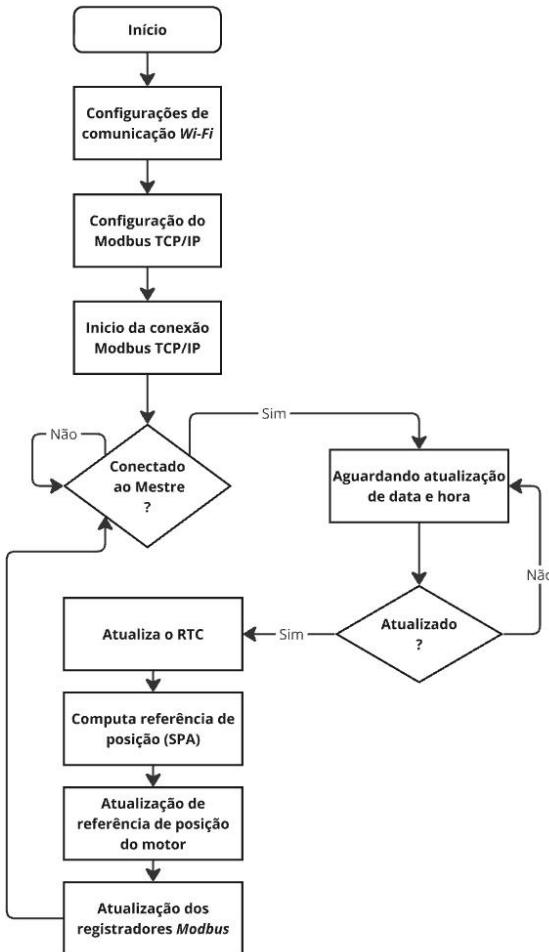
Com essa arquitetura, o mestre não conhece os endereços dos escravos até que eles estabeleçam conexão com ele e isso será um problema se os escravos não estiverem

configurados para estabelecer conexão com o mestre automaticamente. Para resolver esse problema, é possível configurar os escravos para estabelecerem uma conexão com o mestre automaticamente assim que iniciados. Com uma conexão estabelecida com o mestre, ele irá então armazenar o endereços IP únicos de cada escravo em sua tabela de conexões e fará o gerenciamento dos dispositivos conectados em sua rede.

Na Figura 36 é possível se visualizar o fluxograma do *firmware* utilizado para se realizar essas atividades de conexão em cada dispositivo escravo. O principal ponto que pode ser observado ao se visualizar o fluxograma de *firmware*, está na dependência que os módulos escravos têm do sistema que estará atuando como mestre, uma vez que os módulos não possuem a capacidade de atualizarem o relógio automaticamente. Uma vez conectados ao mestre, ele possui a responsabilidade de atualizar a hora de cada módulo e gerenciar as conexões dos seus escravos conectados.

Após possuir o horário sincronizado com o mestre, os dispositivos escravos podem assumir o controle do sistema e passam a operar de forma independente, uma vez que o microcontrolador utilizado possui um RTC interno com capacidade de manter o horário do sistema atualizado enquanto ele estiver energizado. Os dispositivos escravos só poderão operar se o mestre da rede permitir. O mestre pode intervir na operação do sistema, obrigando-o a entrar em modos de operação específicos através da manipulação de registradores Modbus que serão úteis caso alguma condição adversa ocorra com o sistema ou alguma condição climática desfavorável para a operação do rastreador seja identificada, para isso o mestre Modbus deverá conhecer a tabela de registradores do escravo, disponível no Apêndice VI.

Figura 37: Fluxograma do firmware dos módulos escravos Modbus TCP/IP.

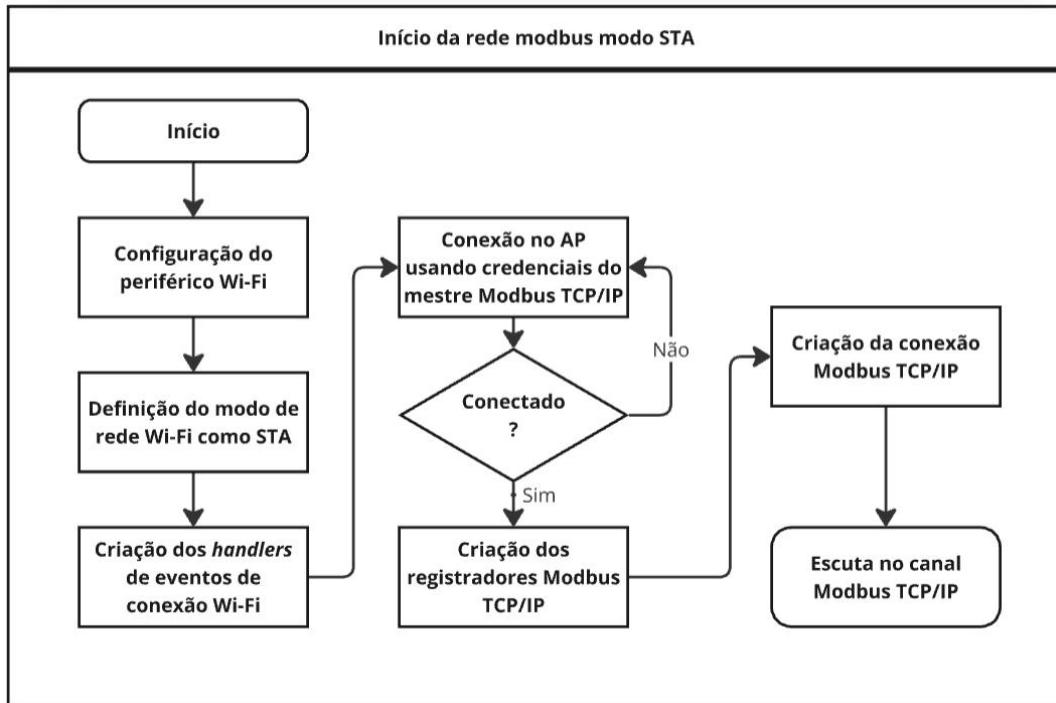


Fonte: Próprio autor.

Dessa forma, o sistema poderá operar de forma independente, realizando o controle de posição de forma local em cada módulo, aliviando as responsabilidades do mestre, que irá apenas ser responsável por gerenciar os dispositivos conectados no sistema e avaliar as condições de operação dos rastreadores, tornando essa, uma abordagem bastante modular.

Para realizar a conexão com o mestre Modbus TCP/IP, é realizada uma etapa de configuração da rede *Wi-Fi* e inicio da rede *Modbus*. Esta etapa é a primeira rotina a ser realizada a partir do momento em que o dispositivo é iniciado, devido ao fato de que é uma etapa lenta, devido ao fato de que o dispositivo deverá fazer uma varredura nas redes disponíveis para realizar a conexão *Wi-Fi* com as credenciais definidas. Essa rotina é realizada de forma concorrente ou paralela (no mesmo CPU ou em outro CPU) e gerenciado pelo próprio microcontrolador. Na Figura 39, é apresentado o fluxograma do dispositivo ao iniciar a sua rotina de conexão *Wi-Fi* do dispositivo escravo com o mestre.

Figura 38: Fluxograma da conexão *Wi-Fi* e criação da rede Modbus TCP/IP.



Fonte: Próprio autor.

O primeiro passo realizado esta na configuração do periférico *Wi-Fi* pelo microcontrolador, onde serão definidos os parâmetros de operação para o funcionamento do *Wi-Fi* de maneira geral. Em seguida é realizada a configuração do modo de operação da rede, que neste caso será uma rede STA, que significa que o dispositivo será tido como uma estação que possui como característica, a necessidade de se conectar em um ponto de acesso. Além dessas configurações, é possível se definir um *handler* de eventos *Wi-Fi*.

O *handler* de eventos é uma interrupção do sistema que é lançado em situações específicas sobre a conexão *Wi-Fi*, como por exemplo, quando o dispositivo se conecta em um ponto acesso, recebe um IP válido ou se desconecta de um AP, sendo muito importante para as lógicas de negócio do sistema.

Após configurado o periférico *Wi-Fi* e definido os *handlers*, pode-se então conectar em uma rede *Wi-Fi* válida, passando os parâmetros de SSID ou nome da rede *Wi-Fi* e a senha. Como o sistema irá se conectar em uma rede mestre *Modbus*, o nome e senha da rede será conhecido previamente. O sistema fica então verificando se a conexão foi bem sucedida ou não e entra em um laço de repetição até que a conexão seja efetuada com sucesso.

Na sequência inicia-se a etapa de configuração dos registradores *Modbus*, alocando os endereços de memória necessários pelo sistema e criando a tabela de endereços, que devem

ser conhecidos pelo mestre. Finalizado as etapas, o dispositivo estará visível para o mestre para trocar informações e ficará escutando no canal *Wi-Fi* em busca de pacotes *Modbus*.

As configurações de conexão que permitem que o sistema saiba se está conectado ou não, são realizadas pelo *handler Wi-Fi*. Esse *handler* está baseado em notificações de eventos ou mudanças de estados do microcontrolador e são lançados quando eventos ocorrem no microcontrolador como o caso dos eventos *Wi-Fi*.

Esses eventos recebem como argumentos: a origem da interrupção, o tipo de evento que foi lançado, a identificação do tipo de evento lançado e as informações da interrupção. Neste exemplo somente os eventos do tipo *Wi-Fi* foram considerados com as identificações:

- i. SYSTEM_EVENT_STA_START: notificação do evento de dispositivo do tipo STA configurado pronto para uso.
- ii. SYSTEM_EVENT_STA_CONNECTED: notificação do evento de dispositivo do tipo STA conectado em um ponto de acesso *Wi-Fi*.
- iii. SYSTEM_EVENT_STA_DISCONNECTED: notificação de evento de que um dispositivo do tipo STA perdeu a sua conexão ou é desconectado do ponto de acesso.

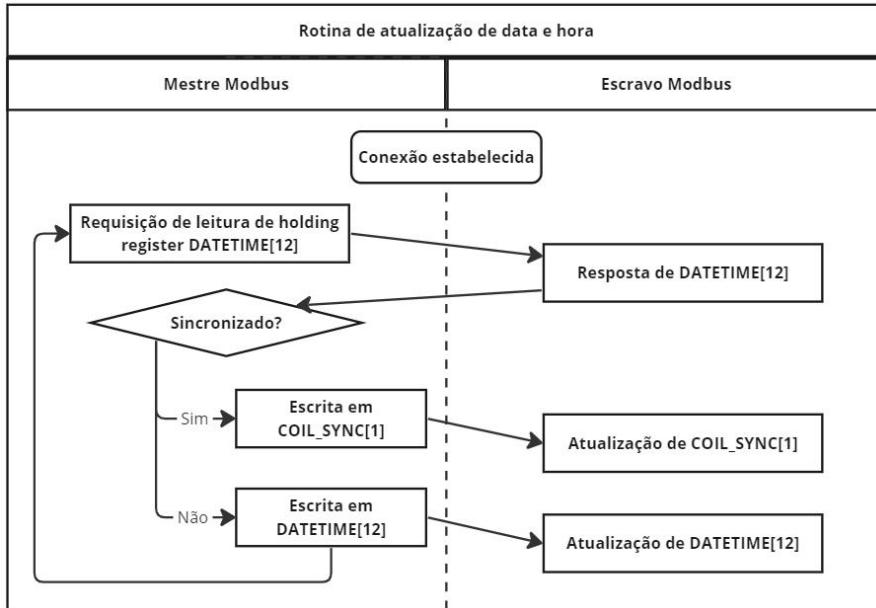
A ESP-IDF possui uma ampla documentação a cerca dos eventos disponíveis para uso que fazem parte do microcontrolador ESP32.

Após a configuração do *Wi-Fi*, o dispositivo tenta se conectar com o ponto de acesso para se tornar visível na rede e estabelecer uma relação com mestre Modbus TCP/IP. Uma vez conectado, o dispositivo fica aguardando que o mestre da rede atualize os registradores de data e hora, para que o dispositivo possa atualizar o seu RTC interno e realizar os cálculos de posicionamento solar através do SPA (do inglês *Solar Position Algorithm*).

O SPA é um algoritmo utilizado para calcular a posição do sol no céu em função de uma determinada data, hora e localização geográfica, além de fatores como latitude, longitude, altitude, e parâmetros atmosféricos para fornecer coordenadas solares precisas, como azimute e elevação.

A rotina de atualização das informações de data e hora se dá pelo lado do mestre do sistema, que possui a obrigação de verificar se os valores de data e hora do dispositivo estão atualizados e sincronizados, uma vez que ele deverá possuir um relógio atualizado. Caso a hora do dispositivo escravo não esteja sincronizada com a hora do mestre, ele irá escrever nos registradores de data e hora com o valor correto e em seguida irá refazer a verificação de sincronismo. Quando a hora estiver sincronizada, o mestre irá escrever em um registrador de sincronismo do tipo bobina, o valor verdadeiro (*true*) e o sistema irá prosseguir. Essa rotina pode ser vista através do fluxo de dados apresentado na Figura 38.

Figura 39: Fluxograma das mensagens Modbus para sincronismo de data e hora.



Fonte: Próprio autor.

A estrutura de *DATETIME* apresentada na Figura 38 possui na sua estrutura, o valor de 6 registradores do tipo *Holding Registers* de 16 bits cada. Dessa forma, o *payload* do pacote transmitido possuirá 12 bytes no formato: *Ano, Mês, Dia, Hora, Minuto, Segundo*.

Quando o dispositivo mestre recebe o pacote de *DATETIME* com a data e hora sincronizada do escravo, ele envia então, um pacote Modbus TCP/IP com um registrador do tipo *Coil Register* de 1 bit, que gera um pacote com *payload* de 1 byte, identificado por *COIL_SYNC*, que define a sincronia da hora entre os sistemas.

A atualização de data e hora pelo lado do dispositivo escravo é feita através RTC interno do microcontrolador e, uma vez que o sistema defina que a hora armazenada nos registradores esta atualizada, o RTC atualiza seu valor e começa a contar o tempo de forma sincronizada enquanto o dispositivo permanecer ativo. O RTC do sistema faz uso de uma estrutura de dados semelhante a estrutura de *DATETIME*.

Após possuir as informações de data e hora atualizada, o sistema é capaz de realizar o rastreio solar de maneira independente, através de uma malha de controle composta pela informação de referência do ângulo solar obtida pelo SPA embarcado no microcontrolador, as informações de ângulo do motor obtido pelo sensor magnético e a planta composta pelo motor BDC.

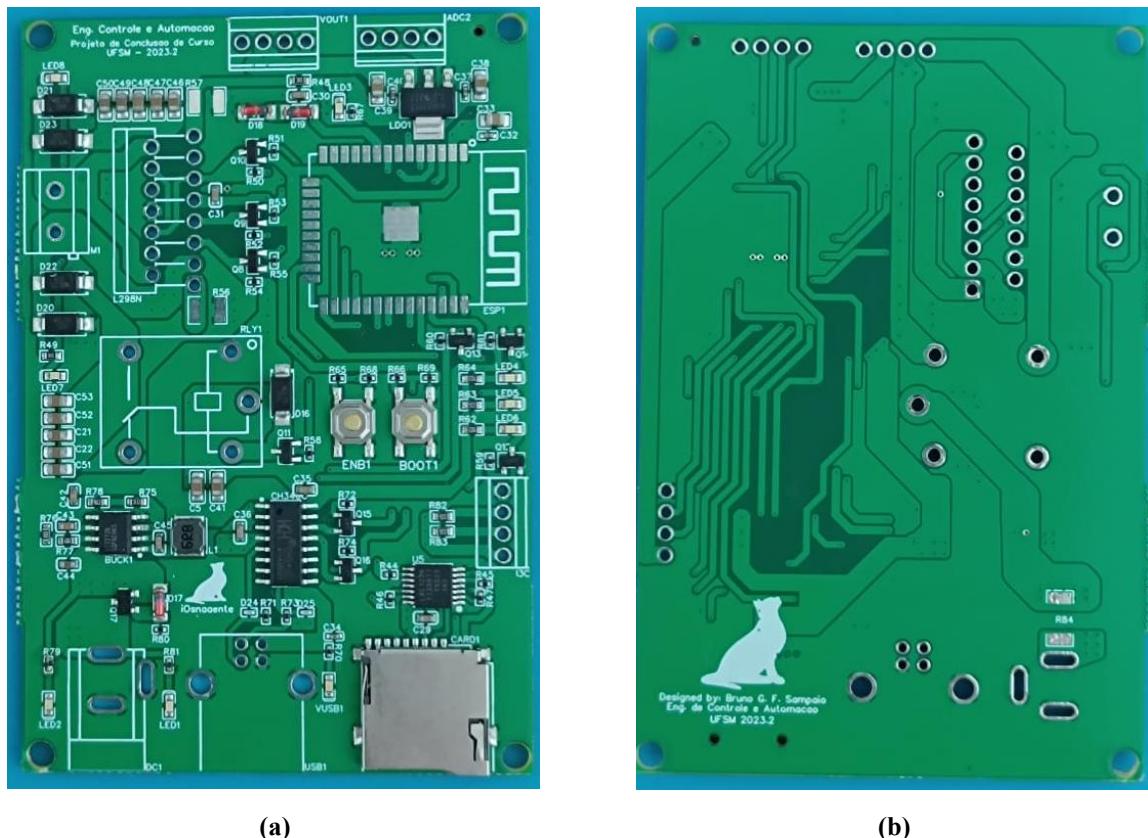
5. RESULTADOS

5.1 Placa de circuito impresso

Para a fabricação das placas de circuito impresso, a empresa JLCPCB foi responsável pela fabricação do projeto, sendo necessário enviar os *gerbers* do projeto digital em CAD e os arquivos de montagem que a própria ferramenta do *EasyEDA* gera, com a relação dos componentes utilizados. Na Figura 39-a e Figura 39-b pode ser visto as imagens das placas logo após a fabricação das mesmas.

Para o processo de montagem por parte da empresa, foram contratados os serviços de montagem apenas para os componentes SMD, que possuem um processo de montagem manual mais complexo e tem um custo de montagem mais barato pela empresa, dessa forma, através do custo benefício, optou-se por realizar a compra sem os componentes PTH.

Figura 40: Placa de circuito impresso fabricada por JLCPCB (a) vista superior (b) vista inferior.



Fonte: Próprio autor.

Com a placa de circuitos impresso em mãos, a próxima etapa foi a de realizar a etapa de soldagem dos componentes que não foram contratados no processo de montagem pela

JLCPCB. Na Figura 40 pode ser visto a placa de circuito impresso com os componentes PTH soldados e os demais componentes que faltavam, sendo apresentado o resultado final obtido.

Figura 41: Placa de circuito impresso do sistema finalizada.



Fonte: Próprio autor.

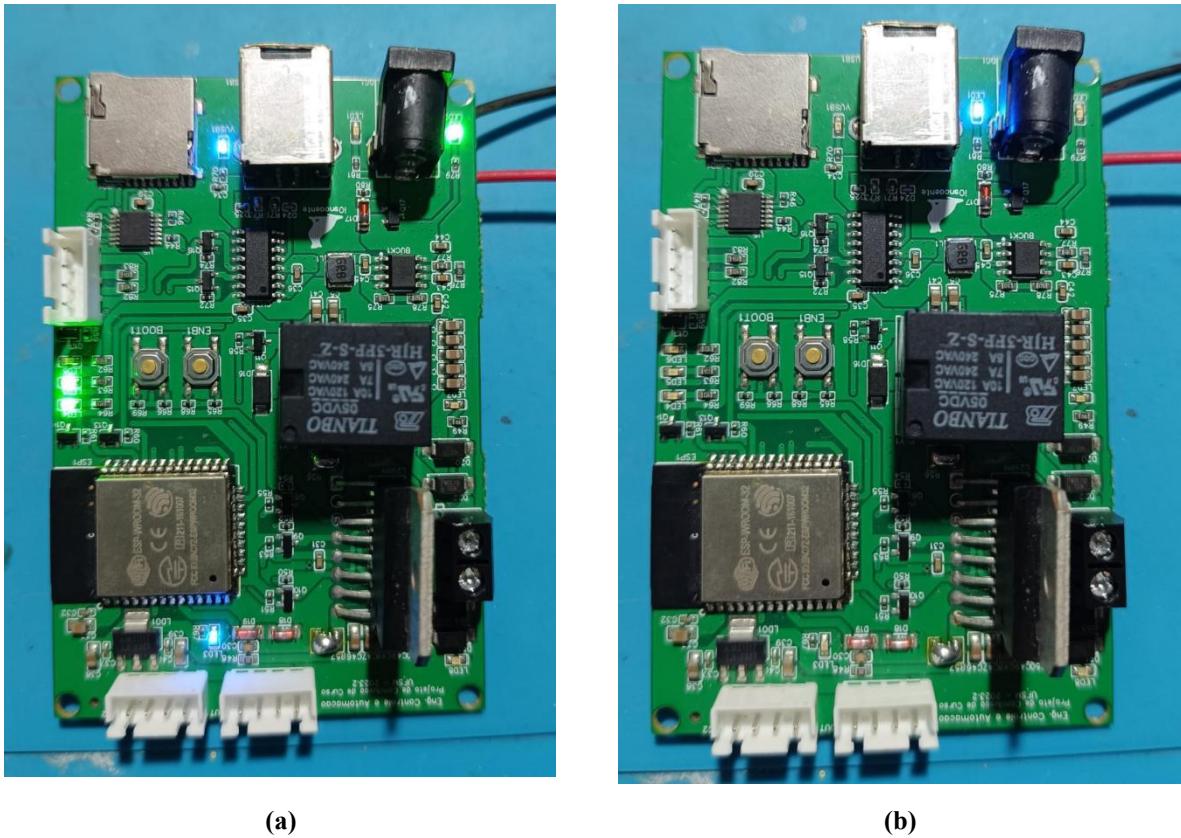
Foram feitas modificações nos conectores GPIO, I₂C e ADC que foram substituídos por conectores do tipo KK de 2,54mm ao invés dos KFE de 2,5mm especificados no projeto originalmente. Essa substituição se deu devido ao fato dos conectores KFE de *pitch* 2,54mm não serem tão acessíveis quando comparados aos conectores utilizados na placa final.

5.1.1 Teste de alimentação

Os primeiros testes foram realizados para avaliar o sistema de alimentação do módulo. O módulo foi alimentado por meio de uma fonte de bancada externa com tensão variável, operando em uma faixa de 9 a 30V. Dentro dessa faixa, o conversor *Buck* presente na placa conseguiu manter uma tensão estável de 5V, garantindo a operacionalidade da ponte H, que controla a tensão de operação dos motores em correlação com a tensão de entrada.

Adicionalmente, foi aplicada uma tensão invertida para testar a proteção contra inversão de polaridade. Durante esse teste, a placa ativou sua proteção, evitando danos aos componentes internos. Para indicar a alimentação incorreta, foi adicionado um LED azul como indicação visual, alertando para a necessidade de correção da fonte de alimentação. Na Figura 41, a placa é mostrada com uma fonte de alimentação correta (vide Figura 41-a) e durante o teste de inversão de polaridade (vide Figura 41-b).

Figura 42: Teste de alimentação da placa. Alimentação (a) diretamente polarizada (b) inversamente polarizada.



(a)

(b)

Fonte: Próprio autor.

Para a saída de tensão no conector VOUT destinado para alimentação de sensores e blocos externos, foi testado com um multímetro na opção voltímetro e acuram as tensões de saída esperada: tensão de entrada, 5V, 3,3V e GND (0V). Os limites de corrente adotados para cada nível tensão foram definidos pelos limites teóricos de cada fonte, sendo 3A menos a corrente de operação do módulo para a tensão de 5V e 1A menos a corrente de operação do microcontrolador do módulo para a tensão de 3,3V.

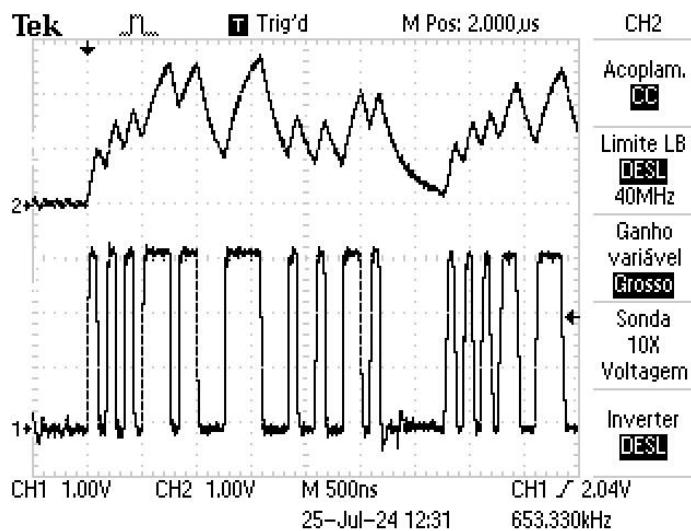
5.1.2 Upload de firmware

Um ponto relevante para a placa desenvolvida está relacionado à capacidade programação do microcontrolador embarcado, todavia, a placa eletrônica contou um com um erro de *design* que dificultou essa característica. Foi adicionado erroneamente de um resistor de valor 10.000Ohm em série entre a entrada USB nos pinos de dados e o conversor CH340C. Esse resistor foi colocado para limitar a corrente entre o módulo e o USB, para evitar sobre corrente em casos de falha, resguardando o computador de danos na porta USB utilizada,

todavia, os valores de resistores adequados para essa finalidade possuem valores abaixo de 100Ohm.

Para avaliar o comportamento esperado e o comportamento obtido do erro, foi analisado os sinais da conexão USB com e sem o resistor de 10.000Ohm. Para avaliar esse sinal, foi medido o sinal enviado antes e após o resistor fazendo uso de um osciloscópio e apresentados na Figura 42.

Figura 43: Avaliação do sinal USB - CH340C antes e após resistor série.



Fonte: Próprio autor.

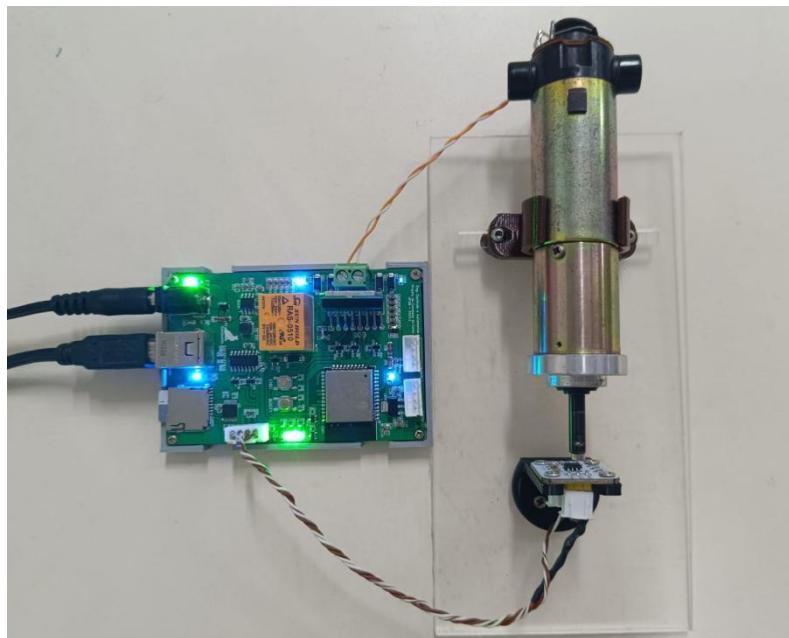
Percebe-se que o resistor de valor elevado, causou um efeito de filtro passa baixas com frequência de corte abaixo do *baudrate* do USB, causando uma forte perturbação no sinal e evitando o correto funcionamento. Como solução, os resistores foram removidos e a comunicação passou a funcionar sem mais problemas.

Com a correção, quando a placa é conectada a um computador por meio do terminal USB B, ele consegue ser reconhecido pelo sistema operacional *Windows*, versão 10 Pro, edição 22H2 sem problemas adicionais. Para programar o módulo, utilizou-se o *software Visual Studio Code*, empregando um código exemplo para testar o LED integrado à placa. Verificou-se que o sistema de *autoboot* do ESP32 funcionou de maneira adequada durante todo o processo, identificando a placa, executando o processo de gravação e colocando-a em modo de programação, possibilitando o carregamento do *firmware* e reinicializando o microcontrolador para iniciar a operação normal.

5.1.3 Teste do driver de motor L298N

Para se validar o funcionamento do *driver* de atuação do motor, foi realizado o teste das conexões da ponte H e da funcionalidade do relé de proteção. Para isso, foi utilizado um motor de corrente contínua BDC conectado ao sistema, um sensor de posição magnético AS5600 e foi desenvolvido uma estrutura para sustentação e alinhamento do aparato (vide Figura 43). Durante os testes, o sistema foi alimentado com uma fonte de 12V@2,5A.

Figura 44: Montagem para teste de acionamento de um motor BDC.



Fonte: Próprio autor.

Um *firmware* foi desenvolvido para controlar o motor BDC, visando se efetuar a atuação do motor através de um sinal de referência senoidal de frequência 0.025Hz (período de 40s) e amplitude unitária que define o *duty cycle* do motor para valores entre -1.0 e 1.0. Para sinais de referência negativos, foi adotado sentido de rotação anti-horário e para sinais positivos o sentido de rotação horário.

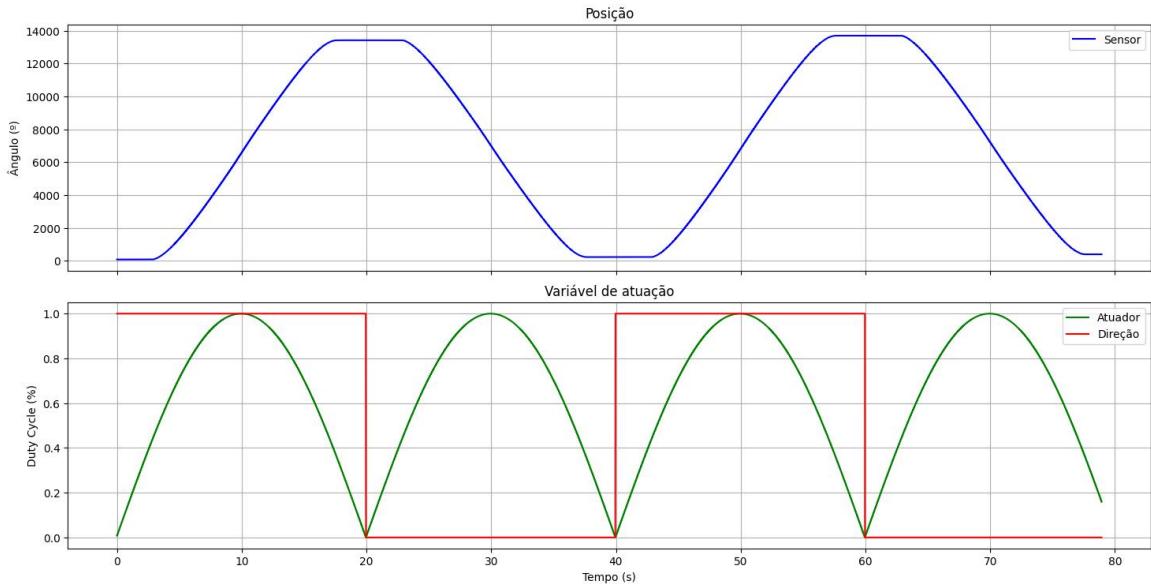
Para se obter os valores de medição, foi utilizado o periférico UART do módulo, e obtidos via conexão USB com o computador. Os dados foram coletados utilizando o *software* de captura chamado *Putty* e processados através de um *script* em *python*. Os valores obtidos podem ser vistos na Figura 44.

O primeiro gráfico descreve a posição do eixo do motor em graus de rotação para um sistema multi voltas. Como foi empregado um ciclo de 40s, nos primeiros 20s de teste, o sistema efetuou um sentido horário de rotação e nos 20s seguintes, um sentido anti horário,

retornando a sua posição inicial. A diferença de ângulo de partida e chegada se dá pela força inercial de partida do motor utilizado.

O segundo gráfico descreve o *duty cycle* de atuação do motor BDC em verde, partindo de 0 a 1, e em vermelho o sentido de rotação do motor, sendo 0 o sentido anti horário e 1 o sentido horário.

Figura 45: Teste de acionamento de motor BDC.



Fonte: Próprio autor.

Durante os testes, constatou-se que o sensor de corrente integrado ao módulo não obteve uma leitura coerente do valor de corrente aplicado ao motor. Embora a corrente pudesse ser visualizada de maneira geral, devido às características de partida de um motor DC, não foi possível medir com precisão devido à presença de ruídos significativos. Diante disso, o circuito do sensor de corrente requer revisão, com a adição de filtros e instrumentações adicionais para o condicionamento do sinal, visando alcançar uma leitura mais precisa e confiável.

5.1.4 Teste de conexão dos sensores

Para a avaliação dos conectores designados aos *encoders*, utilizou-se um osciloscópio a fim de verificar a existência de conexões entre os terminais de saída e o GPIO do microcontrolador e uma vez validado que as conexões estavam integras, desenvolveu-se um *firmware* específico para alternar os níveis lógicos dos pinos e a presença de sinal no osciloscópio permitiu validar que os pinos foram adequadamente implementados na placa desenvolvida.

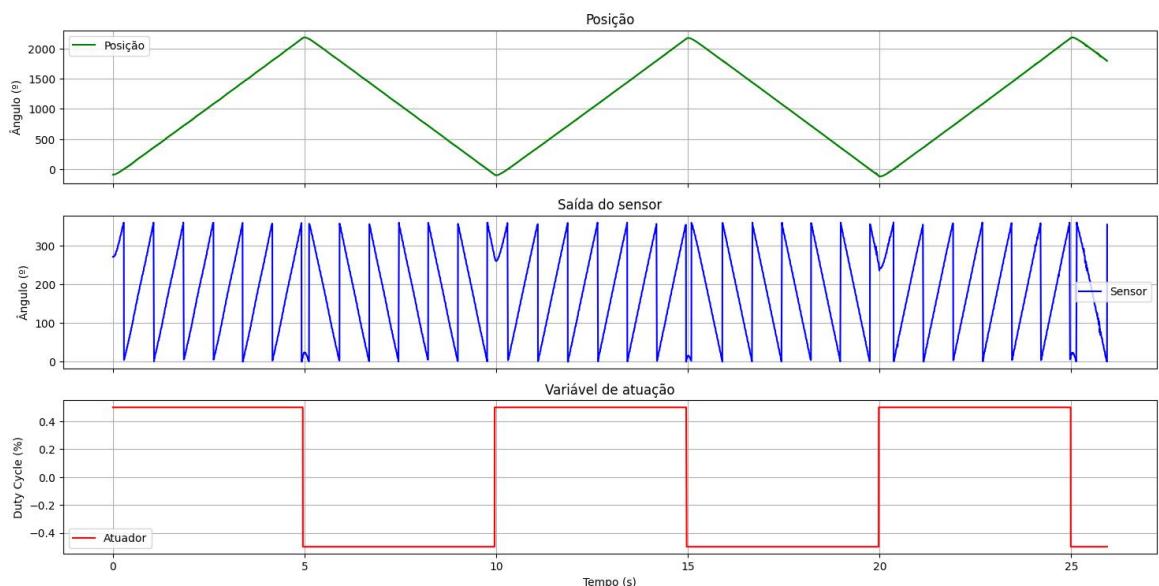
Em seguida, foi possível proceder ao teste dos *encoders* designados para o módulo. Os *encoders* testados incluíram: um *encoder* digital magnético AS5600 com interface I2C, um *encoder* incremental de efeito *Hall* e um potenciômetro para simular um encoder analógico. Com os testes, abrangeu-se assim, os três principais tipos de *encoders* utilizados nos módulos de *Tracker* estudados.

5.1.4.1 Encoder magnético I2C

Para testar o funcionamento do encoder magnético AS5600A, foi utilizado a mesma estrutura montada para o teste de acionamento do *driver* de motor BDC. O objetivo do teste foi o de comprovar o funcionamento da comunicação I2C com a interface disponível na placa, dessa forma, conectou-se o módulo sensor AS5600A e criou-se um firmware de teste do sensor.

Utilizando o modelo de programação do sistema embarcado, criou-se um módulo para operação do sensor, utilizando a interface já definida, chamada Sensor. Para o teste com o sensor, foi aplicado um degrau de tensão no atuador, operando com *duty cycle* de 0,5 e alternando entre os sentidos de rotação com períodos de 5s. Na Figura 45 é possível se avaliar o comportamento do sensor e os resultados obtidos no teste.

Figura 46: Resultados experimentais para módulo de sensor magnético AS5600A.



Fonte: Próprio autor.

No primeiro gráfico, em verde, apresentado na Figura 45, é mostrado a relação de posição do atuador. No segundo gráfico, em azul, é apresentado a saída do sensor AS5600A, que varia de 0 a 360°. No terceiro gráfico, em vermelho, é mostrado os degraus de atuação do

atuador, que varia entre 0,5 e -0,5, o que representa a inversão do sentido de rotação do atuador.

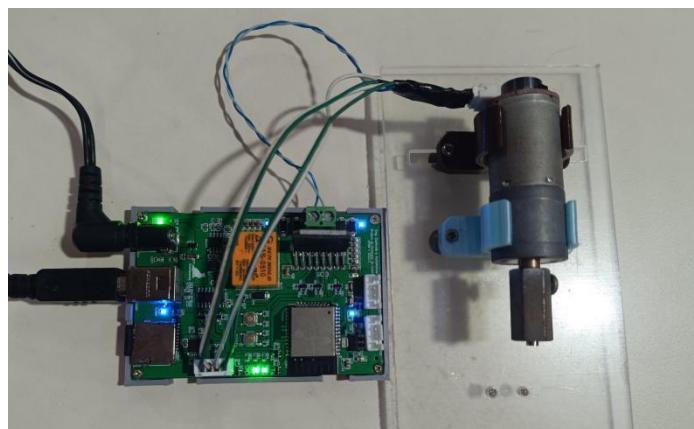
Inicialmente, o atuador opera com uma velocidade constante e um degrau de 50% da potência, girando no sentido horário. Isso resulta em um aumento linear na posição do atuador, visível como uma rampa crescente no gráfico verde. Após 5 segundos, o sentido de rotação do atuador é invertido, causando uma mudança na direção do movimento que é refletida nos gráficos, com a posição do atuador e a saída do sensor ajustando-se conforme o novo sentido de rotação.

Com os resultados obtidos, pode-se comprovar que o sensor operou como esperado, com o sistema embarcado sendo capaz de mensurar a posição angular do atuador com boa precisão. A interface I2C não apresentou nenhuma falha, garantindo uma comunicação estável e confiável entre o sensor e o sistema. A capacidade de detectar e responder às mudanças no sentido de rotação do atuador confirma a eficácia do encoder magnético AS5600A no monitoramento da posição angular.

5.1.4.2 Encoder incremental

Para o teste de sensor utilizando um *encoder* incremental, foi utilizado um modelo de motor DC de menor potência, que possui um *encoder* incremental de efeito *Hall* integrado no eixo do motor. O motor utilizado opera a 100RPM a 12V e possui um imã presente no eixo traseiro que interage com o sensor. O *encoder* possui uma resolução de 10 Pulsos Por Revolução (PPR), e, considerando a relação de engrenagens de 1:119 do motor, a resolução total de saída no motor é de 1.190PPR. Para o teste, foi montado uma estrutura específica para segurar o motor, como mostrado na Figura 46.

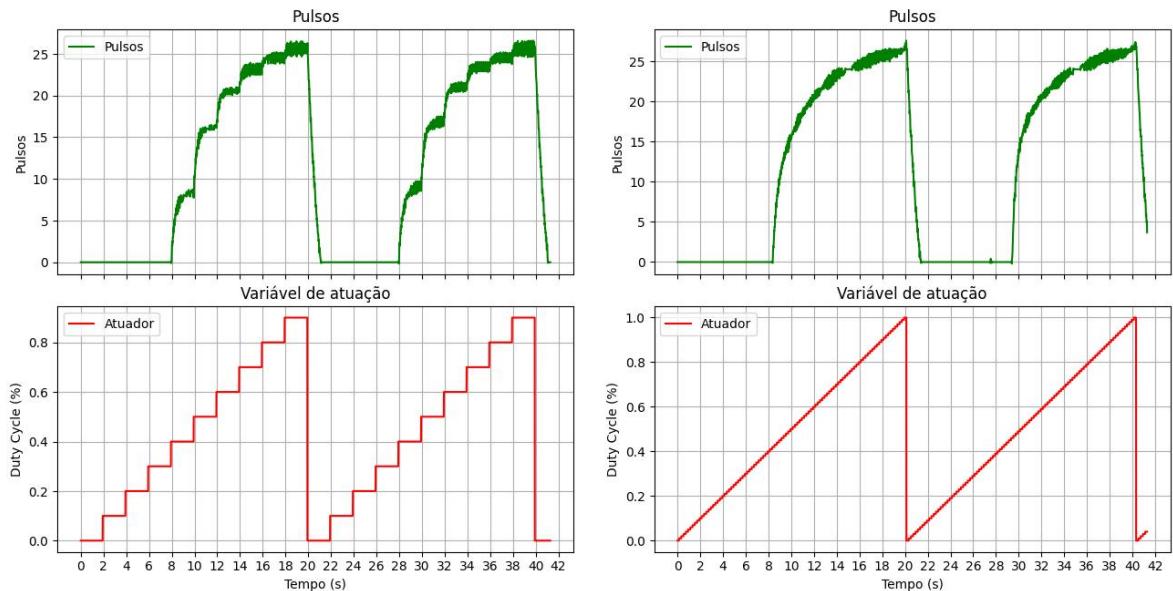
Figura 47: Montagem para teste de encoder incremental de efeito *Hall*.



Fonte: Próprio autor.

Para o teste, foi criado um *firmware* específico, onde se variou os passos de atuação de duas formas: em degraus de 10% a cada 2s e em degraus de 1% a cada 0,2s, como mostrado na Figura 47.

Figura 48: Resultado do sensor incremental de efeito Hall.



Fonte: Próprio autor.

Na Figura 47 é possível ver os resultados dos dois testes simultaneamente. Nos gráficos em verde, estão expostos a contagem de pulsos de cada teste e em vermelho, os degraus de *duty cycle* aplicados no atuador, com valores de 0 a 1. A contagem dos pulsos foi medida em intervalos de 15ms, atingindo o valor máximo de 28 pulsos em *duty cycle* igual a 1.

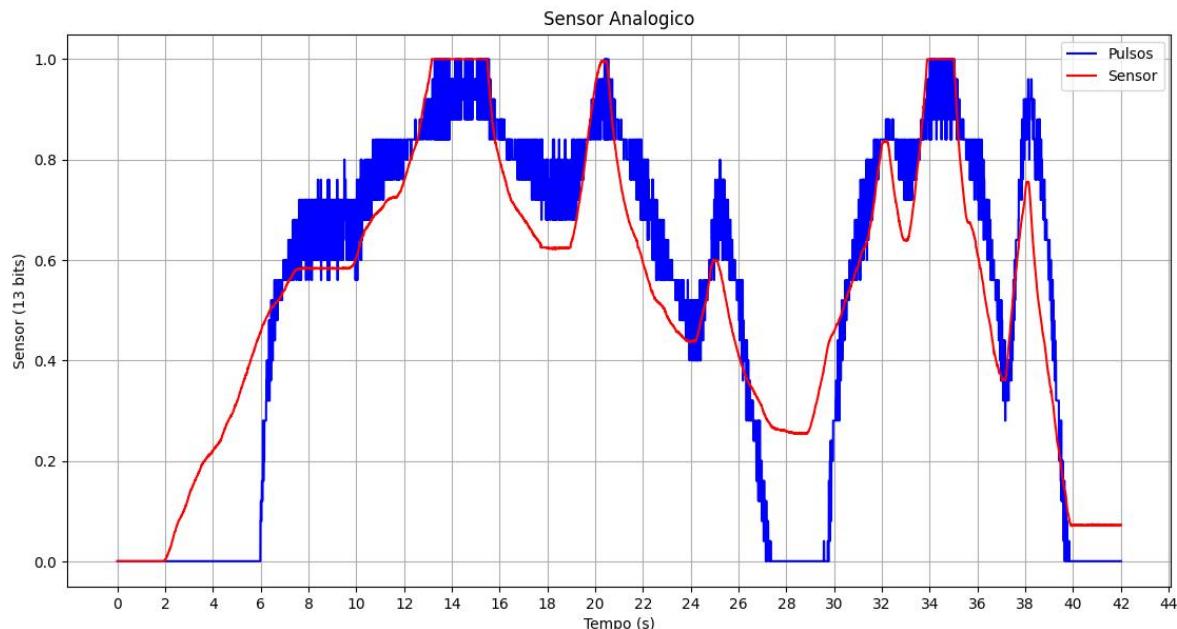
Os resultados também confirmaram um comportamento esperado, destacando uma característica do motor com uma relação de redução de engrenagens elevada. Em regiões onde o *duty cycle* é menor que 0,4, a força exercida pelo atuador não é suficiente para superar a inércia gerada pelas engrenagens, resultando em uma não linearidade na atuação do motor. Dessa forma, para se realizar a atuação linearmente, deve condicionar os valores de atuação para cada motor.

Com os valores obtidos, confirmou-se que o módulo desenvolvido é capaz de realizar a leitura e análises de velocidade para *encoders* incrementais, operando através de interrupções. Com a atualização do *freeRTOS*, pode-se garantir a periodicidade da leitura dos pulsos e se criar uma rotina de atualização e contagem dos pulsos, assegurando que nenhum pulso seja perdido, permitindo que seja realizada a integração dos pulsos para a contabilização da posição do dispositivo através do valor de PPR do sensor.

5.1.4.3 Encoder analógico do tipo potenciômetro

Para o teste de sensor utilizando um *encoder* analógico do tipo potenciômetro, como não havia disponível um sensor desse tipo, utilizou-se um potenciômetro simples desacoplado do motor para se realizar a medição do valor analógico do mesmo. Afim de se avaliar a qualidade da medição, foi realizada a medida do potenciômetro, condicionado o sinal para valores entre 0 e 1 e com o valor, atuou-se no valor de *duty cycle* do motor BDC que possui o *encoder* incremental, dessa forma, pode-se visualizar a variação de velocidade do motor dadas as variações no potenciômetro. Os resultados podem ser vistos na Figura 48.

Figura 49: Resultados do sensor analógico.



Fonte: Próprio autor.

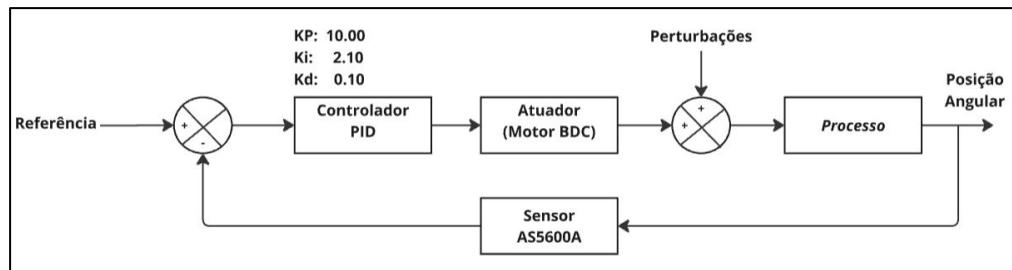
No gráfico apresentado na Figura 48, em vermelho é apresentado a medição do sensor analógico e em azul os valores de pulsos a cada 15ms, ambos com valores normalizados entre 0 e 1. Para a leitura do potenciômetro, foi utilizado 13 bits de resolução, permitindo leituras com valores entre 0 e 4095, medindo na faixa de 0 a 3,3V. Para a contagem dos pulsos, foi normalizado para valores entre 0 e 28 pulsos.

Com os resultados apresentados, foi possível comprovar o comportamento da leitura analógica do módulo. A leitura apresentou uma boa estabilidade de medição e permitiu que sensores analógicos possam ser utilizados como *encoder*. Além disso, a interface analógica não se restringe a *encoders* e pode ser utilizado com módulos ou sensores externos, como sensor de temperatura para monitoramento da temperatura dos atuadores, por exemplo.

5.1.5 Aplicação de malha de controle de velocidade

Com o intuito de validar a capacidade do sistema de executar uma malha de controle de posição, foi realizado um teste de controle utilizando um motor BDC e o sensor magnético AS5600A para se efetuar o controle efetivo do sistema em torno de uma referência de posição estipulada, aplicando um controlador Proporcional Integral Derivativo (PID) como mostrado na Figura 48. Com os resultados obtidos, pode-se avaliar a capacidade do sistema de manter uma resposta às variações de referencial de posição, sendo capaz de cumprir seu papel de rastreamento de referência.

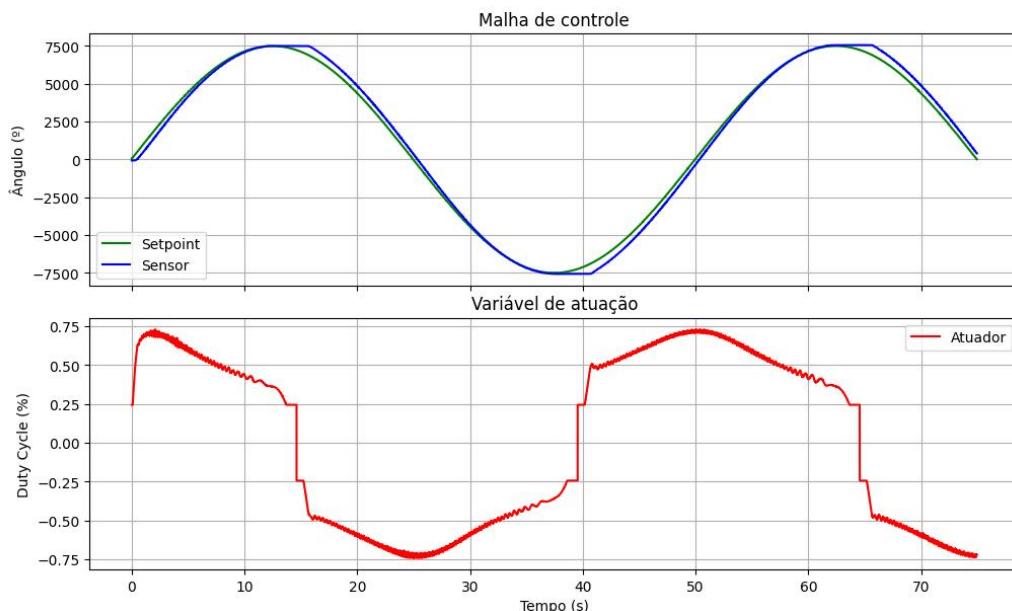
Figura 50: Diagrama de blocos da malha de controle.



Fonte: Próprio autor.

Para o teste de controle, foi desenvolvido um *firmware* para aplicar uma referência de posição senoidal, com amplitude relativa aos graus de rotação do motor, para se definir a posição de referência. O sinal senoidal possui frequência de 0.020Hz (período de 50s) e amplitude de 7.500° de pico (15.000 de pico a pico). O resultado do teste pode ser visto na Figura 49.

Figura 51: Malha de controle.



Fonte: Próprio autor.

Como mostrado na Figura 50, utilizando o sensor de posição magnético para se mensurar a posição angular do atuador, pode-se ver que a posição do motor foi capaz de seguir a referência aplicada pelo sinal senoidal através da malha de controle PID desenvolvido.

Dessa forma, pode-se concluir que o sistema de controle PID, em conjunto com o sensor magnético, proporcionou uma resposta eficaz, mantendo a posição do motor em conformidade com a referência desejada. A integração do sensor, atuador e a implementação com a malha de controle PID demonstrou que a combinação da estrutura de código baseada em interfaces, módulos e controladores foi eficaz.

5.2 Firmware

Para a realização do teste de *firmware* do sistema, foram utilizados as implementações dos módulos sensores, atuadores, RTCs e de rede Modbus TCP/IP descritos. O intuito do teste de *firmware*, é comprovar o correto funcionamento do sistema, integrando todos os módulos e validando a integração dos mesmos, através do padrão de programação utilizando interfaces/módulos/controladores. O presente tópico irá apresentar dois tipos principais de testes, sendo eles: o teste de conexão utilizando a rede Modbus TCP/IP e o teste de integração e controle dos motores para fazer o correto posicionamento do zênite e azimute, utilizando dois módulos conectados na mesma rede Modbus TCP/IP.

5.2.1 Conexão Modbus TCP/IP

Com o objetivo de ser capaz de estabelecer uma conexão Modbus TCP/IP entre os módulos, será validado a capacidade dos módulos de estabelecerem conexão entre si, através de uma rede mestre escravo. Para monitorar os resultados dessa conexão, foi utilizado um *software* de *sniffer* de pacotes TCP/IP. Um *sniffer* é um *software* ou *hardware* que permite que se monitore o tráfego de pacotes de dados transmitidos e recebidos.

O *sniffer* utilizado para a captura dos pacotes TCP/IP trafegados foi o *Wireshark*, um *software* de código aberto disponível para os principais sistemas operacionais. Para realizar essa análise, a versão utilizada do *Wireshark* foi a 4.2.6 disponível em <<https://www.wireshark.org/>> com acesso em 20 de out. de 2024.

Inicialmente, para se realizar o primeiro teste de conexão, foi utilizado apenas um módulo, um computador utilizando o sistema operacional *Windows*, versão 10 Pro, edição

22H2 e um roteador *Wi-Fi* como ponto de acesso (AP) para estabelecimento da rede. Para o teste, optou-se por utilizar o computador como mestre da rede Modbus TCP/IP, tendo como objetivo, atualizar a data e hora do sistema conectado na rede.

Dessa forma, pode-se capturar com facilidade os pacotes transitados na rede Modbus TCP/IP através do *software Wireshark* e avaliar a comunicação estabelecida. Os primeiros pacotes capturados podem ser visto nas Figura 52 e marcam o início da comunicação entre os dispositivos. O dispositivo do módulo possui o IP *10.0.0.107* e para capturar somente pacotes relacionados a ele, foi utilizado o filtro de busca por IP utilizando o IP equivalente.

Figura 52: Pacotes do estabelecimento da comunicação Modbus TCP/IP capturados via *Wireshark*.

No.	Time	Source	Destination	Protocol	Length	Info
594 *REF*		10.0.0.103	10.0.0.107	TCP	66	58433 → 502 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
607 0.237962		10.0.0.107	10.0.0.103	TCP	58	502 → 58433 [SYN, ACK] Seq=0 Ack=1 Win=5760 Len=0 MSS=1436
622 1.001546		10.0.0.103	10.0.0.107	TCP	66	58442 → 502 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
623 1.057420		10.0.0.107	10.0.0.103	TCP	58	502 → 58442 [SYN, ACK] Seq=0 Ack=1 Win=5760 Len=0 MSS=1436
624 1.057477		10.0.0.103	10.0.0.107	TCP	54	58442 → 502 [ACK] Seq=1 Ack=1 Win=64240 Len=0
625 1.070107		10.0.0.103	10.0.0.107	Modbus/TCP	79	Query: Trans: 1; Unit: 18, Func: 16: Write Multiple Registers
626 1.096703		10.0.0.107	10.0.0.103	Modbus/TCP	66	Response: Trans: 1; Unit: 18, Func: 16: Write Multiple Registers
627 1.097048		10.0.0.103	10.0.0.107	Modbus/TCP	66	Query: Trans: 2; Unit: 0, Func: 5: Write Single Coil
628 1.151144		10.0.0.107	10.0.0.103	Modbus/TCP	66	Response: Trans: 2; Unit: 0, Func: 5: Write Single Coil
629 1.193581		10.0.0.103	10.0.0.107	TCP	54	58442 → 502 [ACK] Seq=38 Ack=25 Win=64216 Len=0
630 2.152335		10.0.0.103	10.0.0.107	Modbus/TCP	79	Query: Trans: 3; Unit: 18, Func: 16: Write Multiple Registers
639 2.199033		10.0.0.107	10.0.0.103	Modbus/TCP	66	Response: Trans: 3; Unit: 18, Func: 16: Write Multiple Registers
640 2.199271		10.0.0.103	10.0.0.107	Modbus/TCP	66	Query: Trans: 4; Unit: 0, Func: 5: Write Single Coil
642 2.251598		10.0.0.107	10.0.0.103	Modbus/TCP	66	Response: Trans: 4; Unit: 0, Func: 5: Write Single Coil
643 2.295274		10.0.0.103	10.0.0.107	TCP	54	58442 → 502 [ACK] Seq=75 Ack=49 Win=64192 Len=0

Fonte: Próprio autor via *Wireshark*.

Baseado na primeira captura, pode-se concluir que:

1. Os pacotes 594 a 623 se referem à troca de mensagens TCP (*SYN, ACK*) entre os IPs *10.0.0.103* (computador) e *10.0.0.107* (módulo de rastreamento), estabelecendo uma conexão TCP/IP. O mestre (*10.0.0.103*) inicia a conexão enviando um pacote *SYN* para o escravo (*10.0.0.107*), que responde com um *SYN-ACK*, e o mestre finaliza o estabelecimento com um *ACK*. Esses pacotes configuram o "three-way handshake" do TCP.
2. Nos pacotes subsequentes, após o estabelecimento da conexão TCP/IP, começam as transações Modbus TCP/IP. O mestre (*10.0.0.103*) envia consultas *Modbus* para o escravo (*10.0.0.107*). Dentre as principais transações estão os pacotes:
 - a) Pacote 625: Primeira requisição do mestre para escrita de múltiplos registradores do tipo retenção iniciando nos registradores 18;
 - b) Pacote 626: Resposta de recebimento do comando de escrita por parte do escravo;
 - c) Pacote 627: Requisição do mestre para escrita de um único registrador do tipo bobina;
 - d) Pacote 628: Resposta de recebimento do comando de escrita por parte do escravo;
 - e) Pacote 629: Confirmação de recebimento de pacote por parte do protocolo TCP.

Essa ordem de transmissão é definida pelo sistema, que, ao possuir a data e hora atualizada pelo mestre (Escrita de múltiplos registradores do tipo retenção), verificam na sequência o estado da bobina de sincronia de data e hora para fazer a sincronia da hora do sistema, se a bobina estiver setada como verdadeiro, então a hora é atualizada no sistema (escrita de um único registrador do tipo bobina).

Para avaliar melhor os pacotes Modbus TCP/IP, o *Wireshark* permite que cada pacote seja analisado individualmente, dessa forma, pode-se explorar melhor os dados contidos na comunicação. Na Figura 53 é mostrado com mais detalhes, o conteúdo do pacote 625 mostrado da Figura 52.

Figura 53: Detalhamento de um pacote Modbus TCP/IP via *Wireshark*.

```

▶ Ethernet II, Src: Intel_d7:a9:1f (e0:d4:e8:d7:a9:1f), Dst: Espressif_24:ed:94 (30:ae:a4:24:ed:94)
▶ Internet Protocol Version 4, Src: 10.0.0.103, Dst: 10.0.0.107
▶ Transmission Control Protocol, Src Port: 58442, Dst Port: 502, Seq: 1, Ack: 1, Len: 25
└ Modbus/TCP
    Transaction Identifier: 1
    Protocol Identifier: 0
    Length: 19
    Unit Identifier: 18
└ Modbus
    .001 0000 = Function Code: Write Multiple Registers (16)
    Reference Number: 10
    Word Count: 6
    Byte Count: 12
    └ Register 10 (UINT16): 24
    └ Register 11 (UINT16): 3
    └ Register 12 (UINT16): 21
    └ Register 13 (UINT16): 0
    └ Register 14 (UINT16): 20
    └ Register 15 (UINT16): 0

```

Fonte: Próprio autor via *Wireshark*.

Com essa visão mais detalhada do quadro transmitido, observa-se que os pacotes *Modbus* estão encapsulados nos protocolos de comunicação TCP/IP e os dados transmitidos podem ser identificados como *Modbus TCP* para a ADU e *Modbus* para a PDU. Dentro da ADU estão as identificações que o Modbus TCP/IP utiliza para a transmissão dos pacotes e que fazem parte do protocolo, já na PDU, estão contidos os dados de data e hora transmitidos, que nesse caso, esta transmitindo a data do dia 21 de março de 2024 com a hora definida para 00:20:00 (zero horas com vinte minutos e zero segundos). Essa data foi escolhida para o teste, pois ela define o dia do equinócio de outono no hemisfério sul do planeta Terra, data em que o dia possui a mesma duração de tempo que a noite.

Para múltiplas conexões, pode ser visto na Figura 54 as capturas de dois dispositivos utilizando o protocolo Modbus TCP/IP. Os dois dispositivos representam os módulos de azimute e zênite. Como esperado, os pacotes possuem a característica de serem semelhantes entre si, sendo possível diferenciá-los pelos endereços IP e porta de cada conexão.

Figura 54: Pacotes de múltiplas conexões Modbus TCP/IP.

No.	Time	Source	Destination	Protocol	Length	Info
73	"REF"	10.0.0.103	10.0.0.107	TCP	66	63113 → 502 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
75	0.107078	10.0.0.107	10.0.0.103	TCP	58	502 → 63113 [SYN, ACK] Seq=0 Ack=1 Win=5760 Len=0 MSS=1436
76	0.107144	10.0.0.103	10.0.0.107	TCP	54	63113 → 502 [ACK] Seq=1 Ack=1 Win=64240 Len=0
78	0.123763	10.0.0.103	10.0.0.108	TCP	66	63114 → 502 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
79	0.213440	10.0.0.108	10.0.0.103	TCP	58	502 → 63114 [SYN, ACK] Seq=0 Ack=1 Win=5760 Len=0 MSS=1436
80	0.213494	10.0.0.103	10.0.0.108	TCP	54	63114 → 502 [ACK] Seq=1 Ack=1 Win=64240 Len=0
81	0.214029	10.0.0.103	10.0.0.107	Modbus/TCP	79	Query: Trans: 1; Unit: 18, Func: 16: Write Multiple Registers
83	0.327198	10.0.0.107	10.0.0.103	Modbus/TCP	66	Response: Trans: 1; Unit: 18, Func: 16: Write Multiple Registers
84	0.327431	10.0.0.103	10.0.0.107	Modbus/TCP	66	Query: Trans: 2; Unit: 0, Func: 5: Write Single Coil
85	0.368599	10.0.0.107	10.0.0.103	TCP	54	502 → 63113 [ACK] Seq=13 Ack=38 Win=5723 Len=0
86	0.373228	10.0.0.107	10.0.0.103	Modbus/TCP	66	Response: Trans: 2; Unit: 0, Func: 5: Write Single Coil
87	0.373480	10.0.0.103	10.0.0.108	Modbus/TCP	79	Query: Trans: 1; Unit: 18, Func: 16: Write Multiple Registers
89	0.420283	10.0.0.103	10.0.0.107	TCP	54	63113 → 502 [ACK] Seq=38 Ack=25 Win=64216 Len=0
92	0.457247	10.0.0.108	10.0.0.103	Modbus/TCP	66	Response: Trans: 1; Unit: 18, Func: 16: Write Multiple Registers
93	0.457503	10.0.0.103	10.0.0.108	Modbus/TCP	66	Query: Trans: 2; Unit: 0, Func: 5: Write Single Coil
97	0.510957	10.0.0.108	10.0.0.103	Modbus/TCP	66	Response: Trans: 2; Unit: 0, Func: 5: Write Single Coil
99	0.560206	10.0.0.103	10.0.0.108	TCP	54	63114 → 502 [ACK] Seq=38 Ack=25 Win=64216 Len=0

Fonte: Próprio autor via Wireshark.

Sendo assim, pode-se concluir que a comunicação esta sendo efetuada de acordo com o esperado e os dados de data e hora estão sendo transmitidos, juntamente com a confirmação de sincronia. Mesmo para múltiplas conexões, pode-se comprovar que a comunicação é estabelecida sem problemas. Sendo assim, pode-se afirmar que o sistema esta estabelecendo uma comunicação utilizando o Modbus TCP/IP.

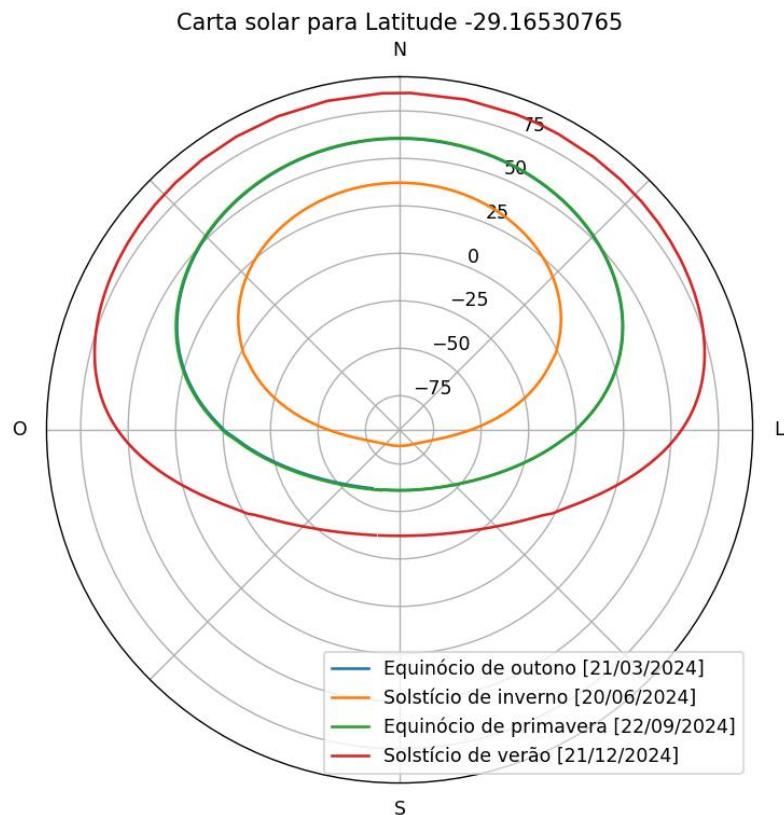
5.2.2 Leitura de registradores Modbus TCP/IP

Após confirmar que a conexão Modbus TCP/IP estava funcionando conforme o esperado, os testes das demais funcionalidades do módulo foram iniciados. Utilizando o protocolo *Modbus*, foi possível efetuar a escrita da data e hora, além de realizar a leitura de dados dos registradores do sistema para analisar pontos de interesse específicos. Para avaliar os cálculos de posicionamento solar, o relógio interno dos módulos foi atualizado de forma acelerada, avançando em intervalos de 2 minutos. Em seguida, os valores obtidos pelo algoritmo de cálculo solar (SPA) foram lidos dos registradores.

Dois módulos foram conectados ao sistema, um configurado como rastreador de azimute e o outro como rastreador de zênite. Esse procedimento permitiu observar os resultados obtidos para os ângulos calculados de azimute e zênite ao longo do tempo. A leitura dos dados desses registradores foi solicitada e, posteriormente, comparada com dados precisos obtidos em sites de rastreamento solar. Os dados para comparação utilizado, foi obtido através do site suncalc.org, um portal de domínio governamental que permite verificar com precisão os dados de movimentação aparente do sol, ao longo de datas e horas específicos.

Uma forma de se visualizar os dados de posicionamento solar, é através da Carta Solar, um método onde os valores de zênite e azimute são colocados em um gráfico polar onde os valores de azimute assumem o eixo dos ângulos no plano e os valores de zênite assumem a posição do raio do plano. Na Figura 55 é possível se observar essa carta solar ao longo de 4 dias de interesse para a região estudada.

Figura 55: Carta solar dos 4 dias de transição entre estações do ano de 2024.



Fonte: Próprio autor.

As datas escolhidas para estudo, apresentados na Figura 55 são:

- Dia 21 de março de 2024, marcado pelo Equinócio de outono;
- Dia 20 de junho de 2024, marcado pelo Solstício de inverno;
- Dia 22 de setembro de 2024, marcado pelo Equinócio de primavera;
- Dia 21 de dezembro de 2024, marcado pelo Solstício de verão;

Todas datas acima, descrevem os dias que marcam as transições entre estações e reforçam a importância de se investir em painéis solares com rastreamento solar para as regiões subtropicais. A comparação entre os dados de zênite calculados nos dias de interesse apresentados e os dados obtidos no site *Suncalc.org* podem ser vistos na Tabela 6.

Tabela 6: Comparação entre dados de Zênite do sistema e de *Suncalc.org*

Data	Zênite módulo	Zênite <i>Suncalc.org</i>	Diferença (°)	Diferença (%)
21/03/2024	60,42 °	59,69 °	0,73	1,223%
20/06/2024	37,17 °	36,87 °	0,30	0,814%
22/09/2024	60,69 °	60,34°	0,35	0,580%
21/12/2024	84,52 °	83,73 °	0,79	0,944%

Fonte: Próprio autor.

Dessa forma, pode-se comprovar que o sistema é capaz de se comunicar via Modbus TCP/IP, calcular corretamente os dados de posição solar e, utilizando uma malha de controle com os parâmetros calibrados para o motor utilizado, realizar o rastreamento de referência com base nos valores de zênite e azimute calculados pelo SPA de cada módulo.

Além dos pontos apresentados, é importante destacar a necessidade de adicionar lógicas de controle adicionais para o correto funcionamento do módulo. Por exemplo, deve-se considerar lógicas de retorno à posição inicial ao final de cada dia de rastreamento e verificações para iniciar uma nova jornada de rastreamento através da avaliação do ângulo de zênite. Também é fundamental implementar rotinas para monitorar emergências climáticas e funções de estado de falha para proteger o sistema.

CONCLUSÃO

O desenvolvimento, validação da placa de circuito impresso (PCB) e do *firmware* associados ao projeto foram realizados de maneira satisfatória, cumprindo com o papel desejado nas etapas de idealização, apesar de alguns desafios enfrentados durante o processo. A fabricação da PCB pela empresa JLCPCB e a subsequente montagem e soldagem dos componentes evidenciaram a eficácia do planejamento e execução do projeto, permitindo a criação de um sistema que além de funcional, possui uma estética profissional.

Durante a fase de testes, o sistema demonstrou excelente desempenho em vários fatores, desde a estabilização da tensão de alimentação até a proteção contra inversão da polaridade de alimentação. A placa passou por testes que garantiram a integridade e funcionalidade dos componentes integrados, mesmo quando submetidos a condições indesejadas de uso, sem danificar nenhum componente integrado, como microcontrolador, conversor USB ou *driver* de motor BDC.

O projeto contou com um erro de *design*, relacionado à resistência em série na linha de comunicação USB que foi identificado e corrigido, o que ressalta a importância da validação de projeto em etapas iniciais e sobretudo a revisão do projeto esquemático e das folhas de dados dos componentes. Após a correção, o sistema funcionou conforme o esperado, permitindo a programação e operação do microcontrolador sem problemas.

Durante os testes realizados com diferentes tipos de encoders: incremental, analógico e magnético, confirmaram a flexibilidade e precisão do sistema na leitura de posições angulares e velocidade do motor, permitindo a implementação de malhas de controle de posição do motor. A aplicação de uma malha de controle PID com o motor BDC demonstrou a capacidade do sistema de seguir referências de posição com precisão, validando a eficiência da integração entre *hardware* e *software* além de ser capaz de atingir o seu objetivo principal de rastreamento solar.

Como sistemas de rastreamento solar, contam com motores de baixa rotação e normalmente caracterizados pelo uso de técnicas de redução elevada de velocidade através de engrenagens, por exemplo, as técnicas de controle de posicionamento se tornam mais complexas, devido ao elevado atraso que essa redução acarreta na planta controlada. Dessa forma, será necessário em pesquisa futuras, se realizar uma malha de controle mais sofisticada para o sistema, todavia, foi evidenciado a capacidade de se desempenhar cálculos

matemáticos complexos com o microcontrolador selecionado, evidenciado pelos cálculos do SPA.

Por fim, a arquitetura de *firmware* desenvolvida, baseada em interfaces, módulos e controladores, mostrou-se eficaz para a integração e operação de todos os componentes do sistema, facilitando a comunicação através da rede Modbus TCP/IP e o controle dos atuadores. A abordagem modular permitiu uma validação eficiente do sistema como um todo, garantindo que o projeto atenda aos requisitos propostos e esteja pronto para implementações.

BIBLIOGRAFIAS

ABREU VIEIRA, Pedro Henrique, et al. SISTEMA DE RASTREAMENTO SOLAR DE UM EIXO PARA MÓDULOS FOTOVOLTAICOS UTILIZANDO A PLATAFORMA ARDUINO. In: Congresso Brasileiro de Energia Solar-CBENS. 2014.

AMS. **AS5600** [datasheet](#). 2014. Disponível em: <https://www.mouser.com/datasheet/2/588/AS5600_UG000254_2_00-1877354.pdf>. Acesso em: 13 de nov. de 2023.

ANGULO-CALDERÓN, Marthoz, et al. Development and accuracy assessment of a high-precision dual-axis pre-commercial solar tracker for concentrating photovoltaic modules. Applied Sciences, 2022, 12.5: 2625.

ARAÚJO, C. R. D. S. Estudo comparativo de controladores proporcional-integral e fuzzy para acionamento de máquinas CC, 2022.

BARBOSA, Erlison de Sousa. Desenvolvimento de um sistema de controle de baixo custo para rastreamento de concentradores cilíndrico-parabólicos. 2009.

BELO, F. A. W. Controle de Velocidade de um Sistema Robótico Rádio-Controlado Sentinel - Doctoral dissertation, PUC-Rio, 2004.

COPÉRNICO. N. Commentariolus: Pequeno comentário de Nicolau Copérnico sobre suas próprias hipóteses acerca dos movimentos celestes. Tradução por Roberto de Andrade Martins. 2. ed. São Paulo: Livraria da Física, 2003.

DA ROCHA QUEIROZ, Juliano; DA SILVA SOUZA, Anacreone; ANDRADE, Cid Marcos Gonçalves. CLASSIFICAÇÃO DE RASTREADORES SOLARES: UMA BREVE REVISÃO - III Simpósio Paranaense de Modelagem, simulação e controle de processos - Pag. 152 - 159, (2018).

E. FITZGERALD, C. KINGSLEY JR, A. KUSKO. **Máquinas Elétricas**. Ed. McGraw-Hill do Brasil, 1975.

Embarcados. **SPI (Serial Peripheral Interface) – Parte 1**. Disponível em: <<https://embarcados.com.br/spi-parte-1/>>. Acesso em: 15 de nov. De 2023.

ESPRESSIF. ESP32 Datasheet. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. Acesso em: 15 de nov. de 2023.

GONÇALVES FILHO, Carlos Antônio Pereira. **Desenvolvimento de um protótipo de rastreador solar de baixo custo e sem baterias**. 2007. Master's Thesis. Universidade Federal de Pernambuco.

GREEN POWER: **Parque solar São Gonçalo**. 2021. Disponível em: <<https://www.enelgreenpower.com/pt/nossos-projetos/highlights/parque-solar-sao-goncalo>>. Acesso em: 01 set 2021.

Kilian, C. T. **Modern control technology: components and systems**. West Publishing Co - (1996).

MATOS, B. G. G. D. **Controlador e accionador para motor DC em malha fechada** - Doctoral dissertation, 2008.

MORETO, Marcelo. **CONTROLE DE SERVOMOTOR**. 2007.

NSENGIYUMVA, Walter, et al. **Recent advancements and challenges in Solar Tracking Systems (STS): A review**. *Renewable and Sustainable Energy Reviews*, 2018, 81: 250-279.

PEREIRA, E. B.; et al: **Atlas brasileiro de energia solar. 2.ed.** São José dos Campos: INPE, 2017. 80p. Disponível em: <http://doi.org/10.34024/978851700089>. Acesso em: 2 nov de 2021.

sieBRASIL: **Ministério de minas e energia do Brasil – Sistema de Informação de Energias SIE - Capacidade Instalada de Geração Elétrica**: Disponível em:

<<https://www.mme.gov.br/SIEBRASIL/consultas/reporte-dato42-jerarquizado.aspx?oc=30181&V=30182&ss=2&v=1>>. Acesso em: 01 set 2021.

SILVA, Rutelly Marques da. Energia Solar no Brasil: **dos incentivos aos desafios. 2015.** Disponível em: <<http://www12.senado.gov.br/publicacoes/estudos-legislativos/tipos-de-estudos/textos-para-discussao/td166>>. Acesso em: 01 set. 2021

STMICROELECTRONICS. L298 datasheet. 2023. Disponível em: <<https://www.st.com/resource/en/datasheet/l298.pdf>>. Acesso em: 12 de nov. de 2023.

SunCalc. **Computation Path of the Sun.** Available online: www.suncalc.org (Acessor em: 14 de nov. de 2023).

Tang, S., & Yu, Y. **Research on Closed-loop Control of Step Motor Based on Magnetic Encoder - Highlights in Science, Engineering and Technology**, 1, 351–356. (2022).

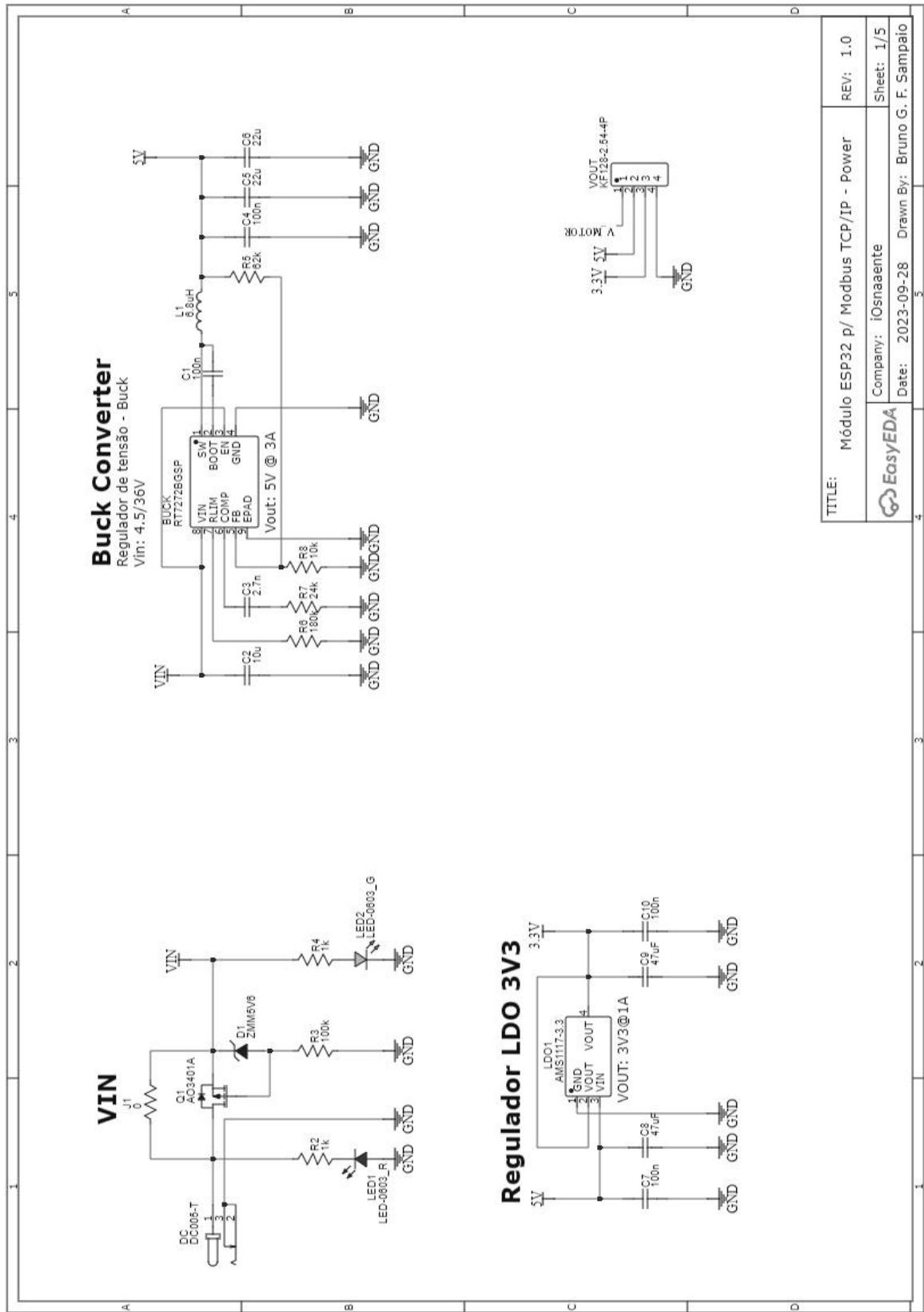
TOSSATO, Claudemir Roque; MARICONDA, Pablo Rubén: **O método da astronomia segundo Kepler.** Scientiae Studia, v. 8, p. 339-366, 2010.

RICHARDSON, L: **ENERGYSAGE.** mai 2018, Disponível em: <<https://news.energysage.com/the-history-and-invention-of-solar-panel-technology/>>. Acesso em: 01 set 2021.

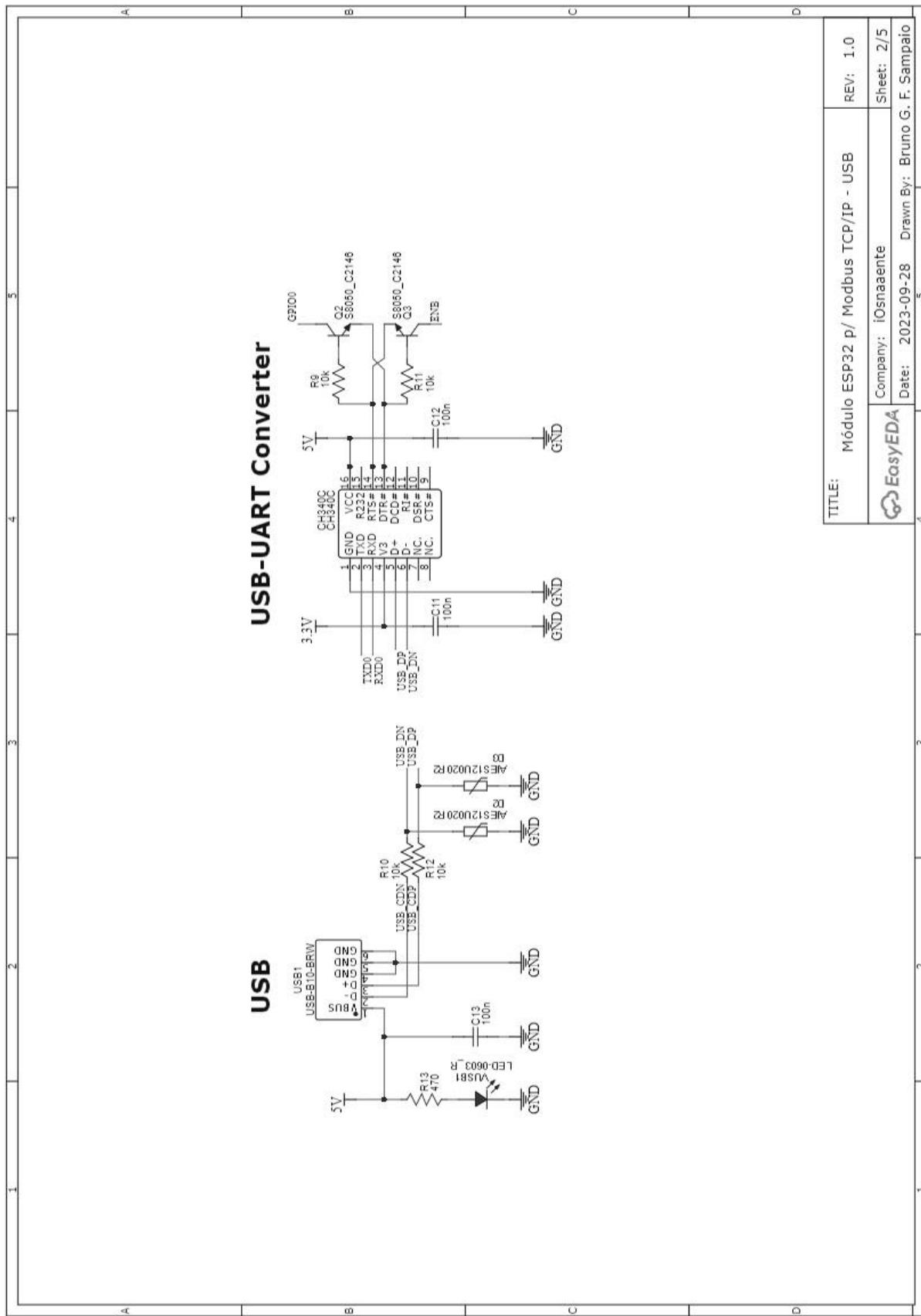
VALLDOREIX GREENPOWER: **The Benefits of Solar Trackers.** Julho de 2015. Disponível em: <<http://www.valldoreix-gp.com/the-benefits-of-solar-trackers/>>. Acesso em: 01 set 2021.

APÊNDICE

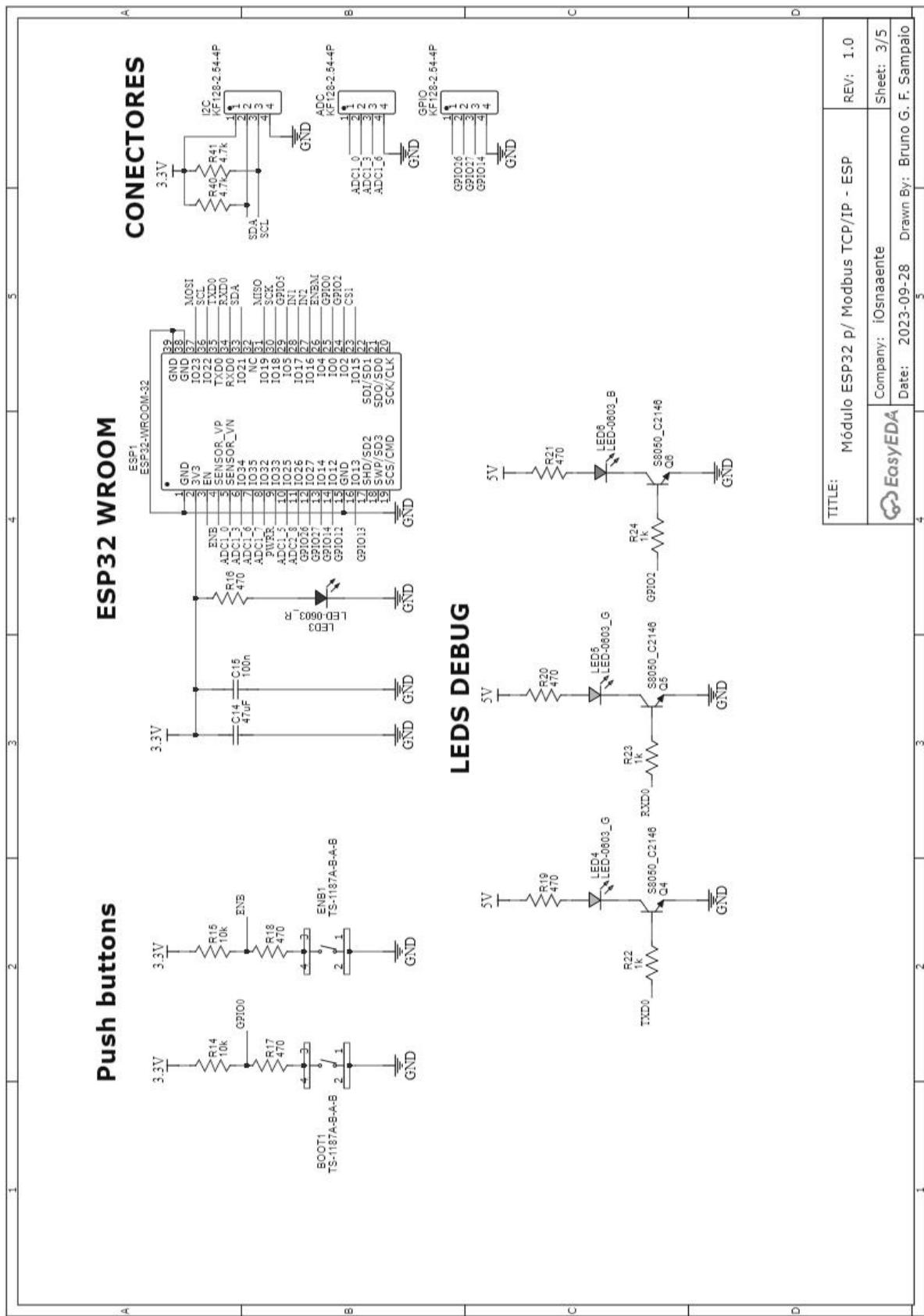
Apêndice I - Esquemático do módulo / Gestão de energia do sistema.



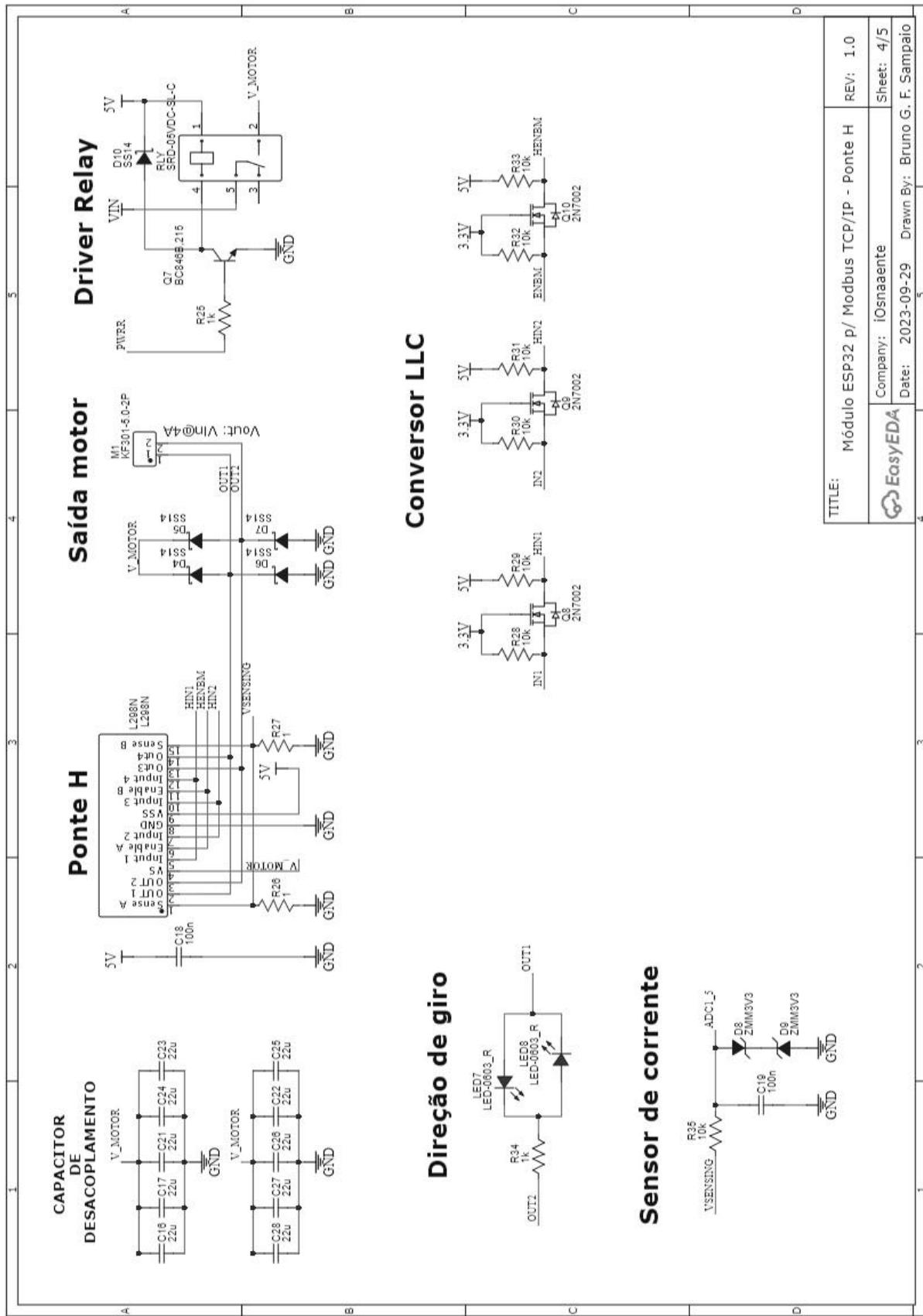
Apêndice II -Esquemático do módulo / Conversor USB - UART.



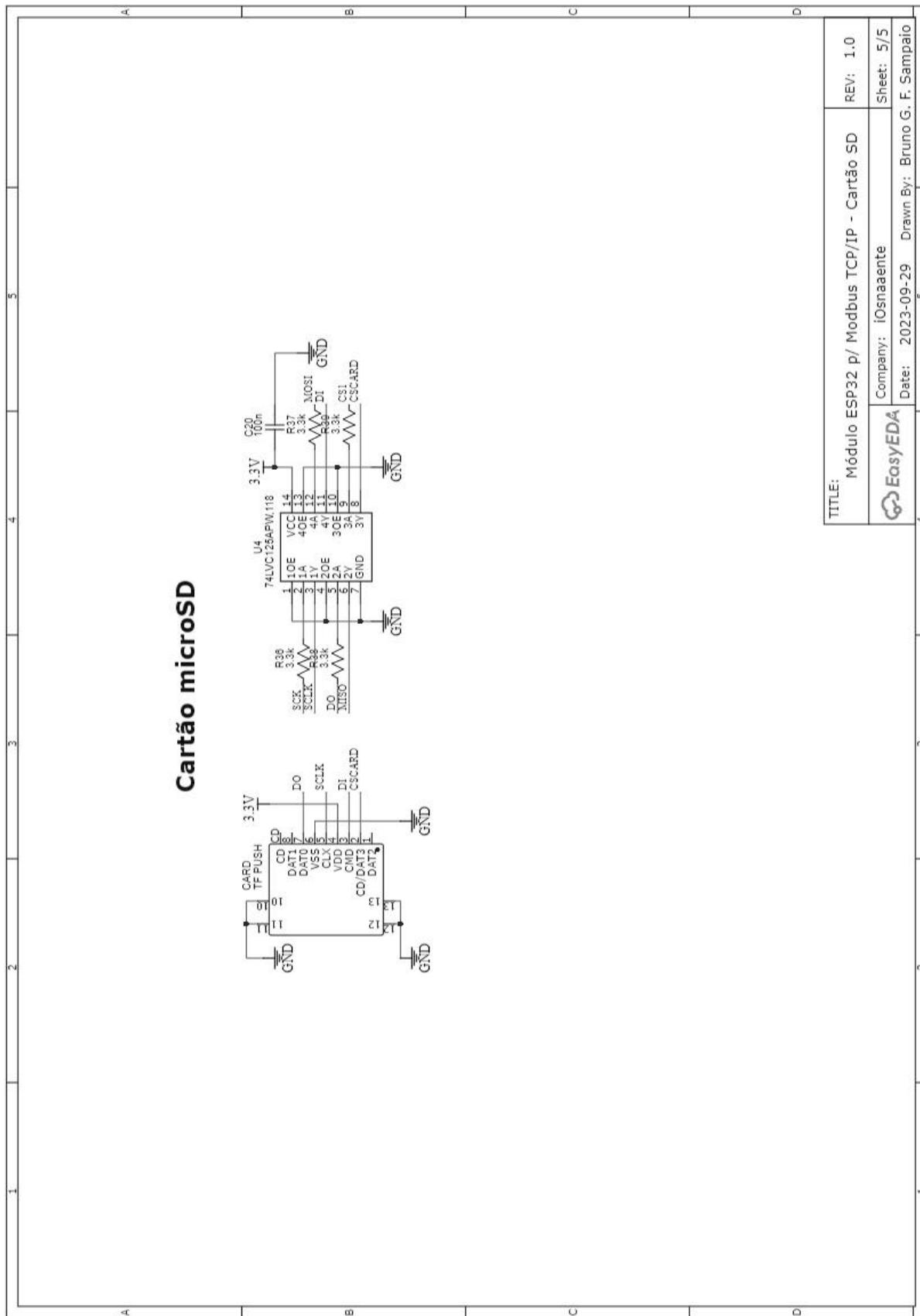
Apêndice III - Esquemático do módulo - Microcontrolador.



Apêndice IV -Esquemático do módulo / Circuito da ponte H.



Apêndice V - Esquemático do módulo / *Data logger* do sistema.



Apêndice VI - Mapa de registradores Modbus TCP/IP embarcado.

Nome do registrador	Tipo de registrador	Tamanho	Endereço
INPUT SYSTEM MODE	<i>Input Register</i>	2 bytes	0x00
INPUT SENSOR POS	<i>Input Register</i>	2 bytes	0x01
INPUT SENSOR STATUS	<i>Input Register</i>	2 bytes	0x02
INPUT SUN TARGET	<i>Input Register</i>	2 bytes	0x03
INPUT ZENITH	<i>Input Register</i>	2 bytes	0x04
INPUT AZIMUTH	<i>Input Register</i>	2 bytes	0x05
INPUT POWER GEN	<i>Input Register</i>	2 bytes	0x06
INPUT TEMPERATURE	<i>Input Register</i>	2 bytes	0x07
INPUT PRESURE	<i>Input Register</i>	2 bytes	0x08
INPUT YEAR	<i>Input Register</i>	2 bytes	0x09
INPUT MONTH	<i>Input Register</i>	2 bytes	0x0A
INPUT DAY	<i>Input Register</i>	2 bytes	0x0B
INPUT HOUR	<i>Input Register</i>	2 bytes	0x0C
INPUT MINUTE	<i>Input Register</i>	2 bytes	0x0D
INPUT SECOND	<i>Input Register</i>	2 bytes	0x0E
HR STATE	<i>Holding Register</i>	2 bytes	0x00
HR MOTOR PV	<i>Holding Register</i>	2 bytes	0x01
HR MOTOR KP	<i>Holding Register</i>	2 bytes	0x02
HR MOTOR KI	<i>Holding Register</i>	2 bytes	0x03
HR MOTOR KD	<i>Holding Register</i>	2 bytes	0x04
HR ALTITUDE	<i>Holding Register</i>	2 bytes	0x05
HR LATITUDE	<i>Holding Register</i>	2 bytes	0x06
HR LONGITUDE	<i>Holding Register</i>	2 bytes	0x08
HR YEAR	<i>Holding Register</i>	2 bytes	0x0A
HR MONTH	<i>Holding Register</i>	2 bytes	0x0B
HR DAY	<i>Holding Register</i>	2 bytes	0x0C
HR HOUR	<i>Holding Register</i>	2 bytes	0x0D
HR MINUTE	<i>Holding Register</i>	2 bytes	0x0E
HR SECOND	<i>Holding Register</i>	2 bytes	0x0F
DISCRETE ADC1_0	<i>Discrete Input</i>	1 bit	0x00
DISCRETE ADC1_3	<i>Discrete Input</i>	1 bit	0x01
DISCRETE ADC1_6	<i>Discrete Input</i>	1 bit	0x03
DISCRETE FAIL	<i>Discrete Input</i>	1 bit	0x04
DISCRETE SYNC	<i>Discrete Input</i>	1 bit	0x05
COIL POWER	<i>Coil Register</i>	1 bit	0x00
COIL LED	<i>Coil Register</i>	1 bit	0x01
COIL DT SYNC	<i>Coil Register</i>	1 bit	0x02
COIL PUSH BUTTON 1	<i>Coil Register</i>	1 bit	0x03
COIL PUSH BUTTON 2	<i>Coil Register</i>	1 bit	0x04
COIL PUSH BUTTON 3	<i>Coil Register</i>	1 bit	0x05