

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Automatização de um Telescópio Newtoniano por meio de um sistema GoTo na ESP32

**Autores: Bruno Carmo Nunes
Marcos Vinícius Rodrigues da Conceição
Orientador: Prof. Dr. Renato Coral Sampaio**

**Brasília, DF
2023**



Bruno Carmo Nunes
Marcos Vinícius Rodrigues da Conceição

Automatização de um Telescópio Newtoniano por meio de um sistema GoTo na ESP32

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Renato Coral Sampaio

Brasília, DF

2023

Bruno Carmo Nunes
Marcos Vinícius Rodrigues da Conceição
Automatização de um Telescópio Newtoniano por meio de um sistema GoTo
na ESP32/ Bruno Carmo Nunes
Marcos Vinícius Rodrigues da Conceição. – Brasília, DF, 2023-
91 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Renato Coral Sampaio

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2023.

1. Telescópio GoTo. 2. ESP32. I. Prof. Dr. Renato Coral Sampaio. II.
Universidade de Brasília. III. Faculdade UnB Gama. IV. Automatização de um
Telescópio Newtoniano por meio de um sistema GoTo na ESP32

CDU 02:141:005.6

Bruno Carmo Nunes
Marcos Vinícius Rodrigues da Conceição

Automatização de um Telescópio Newtoniano por meio de um sistema GoTo na ESP32

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 26 de julho de 2023:

Prof. Dr. Renato Coral Sampaio
Orientador

Prof. Dr. Gerardo Antonio Idrobo Pizo
Convidado 1

Prof. Dr. Maurício Serrano
Convidado 2

Brasília, DF
2023

Este trabalho é dedicado a todos que olham para o céu, e se deslumbram com o dança das estrelas.

Agradecimentos

Agradeço com todo o amor e sinceridade à minha família, primeiramente aos meus pais, Odenilza e Nestor, que me incentivaram e apoiaram ao longo de toda minha trajetória acadêmica. Sem isso, não teria o vasto conhecimento que zelo como o presente mais importante da minha vida. Também as minhas queridas irmãs, Amanda, Maria Clara e Carolina, que me deram companhia, conforto e alegria;

Aos meus amigos, Caio, Gustavo, Ian, Lúcio, Michel e Tobias que me acompanharam desde o início da minha trajetória acadêmica. Nos tempos de alegria e aflição, sempre estiveram ali para me estender a mão;

Por fim, agradeço ao meu parceiro de trabalho Marcos Vinícius, que compartilhou comigo as frustrações e soluções que desenvolvemos juntos.

- Bruno

Agradeço aos meus pais, Maria da Graça e Onorino pelo apoio incondicional do início ao fim dessa jornada;

Agradeço a todos os amigos que consegui nesse período e tornaram essa jornada mais fácil e gratificante;

Agradeço ao meu amigo Bruno, meu parceiro neste trabalho, por toda cooperação durante as inúmeras madrugadas a fio, a fim de entregar o melhor trabalho possível.

- Marcos Vinícius

Agradecemos ao grupo de astronomia e astrofísica, Lucas, Maria Cristina, e Thompson, que ajudaram a elaborar e implementar o primeiro telescópio com lente própria produzida na Faculdade do Gama;

Ao professor Wytler Cordeiro dos Santos, que nos influenciou desde o começo da faculdade a sempre olhar para céu. Nos explicando os fenômenos físicos que regem o universo, e também nos instruindo a criar ferramentas para observá-los;

Agradecemos em especial ao professor Renato Coral Sampaio, que nos inspirou e compartilhou de seu conhecimento sobre sistemas embarcados, por sua orientação e disponibilidade no auxílio da elaboração deste trabalho.

Resumo

Este projeto visa converter um telescópio newtoniano com base equatorial em um telescópio com base dobsoniana, possibilitando a incorporação de um sistema GoTo. Para o funcionamento dessa base, utilizaremos o protocolo INDI (Instrument Neutral Distributed Interface), que viabiliza a comunicação entre o cliente e um software planetário, como o KStars. A ESP32, desenvolvida pela Expressif, será empregada para controlar o telescópio e estabelecer a comunicação via WiFi para o protocolo INDI. Esse controle possui motores de passo para movimento, sensores *Hall* e encoders para medir o comportamento da estrutura, além de um módulo GPS para obtenção de dados de latitude e longitude. Foram conduzidos três testes com astros próximos ao planeta Terra, obtendo resultados positivos no apontamento e rastreamento de Saturno, Júpiter e a Lua, sendo possível rastrear esses astros por longos períodos. Baseado nesses testes, identificamos a necessidade de aprimorar a precisão do telescópio, o que implicará em mudanças na estrutura e no software embarcado. Assim, a construção desse telescópio serve como referência para a criação de um telescópio GoTo, abrindo caminho para entusiastas que aspiram observar e explorar os astros com maior facilidade e precisão.

Palavras-chaves: Telescópio, GoTo, ESP32, INDI, motor de passo, GPS, KStars.

Abstract

This project aims to convert a Newtonian telescope with an equatorial base into a telescope with a Dobsonian base, enabling the incorporation of a GoTo system. For the operation of this structure, we will use the INDI protocol (Instrument Neutral Distributed Interface), which enables communication between the client and planetary software, such as KStars. The ESP32, developed by Expressif, will be used to control the telescope and establish communication via WiFi for the INDI protocol. This control involves stepper motors for movement, *Hall* sensors and encoders to measure the behavior of the structure, in addition to a GPS module for obtaining latitude and longitude data. Three tests were conducted with stars close to planet Earth, obtaining positive results in the pointing and tracking of Saturn, Jupiter and the Moon, where it was possible to track these stars for long periods. Based on these tests, we identified the need to improve the telescope's accuracy, which will imply changes in the structure and on-board software. Thus, the construction of this telescope serves as a reference for the creation of a GoTo telescope, paving the way for enthusiasts who aspire to observe and explore the stars with greater ease and precision.

Key-words: Telescope, GoTo, ESP32, INDI, stepper motor, GPS, KStars.

Lista de ilustrações

Figura 1 – Esfera celeste visível para um certo lugar no hemisfério sul. (SARAIVA, 2017)	27
Figura 2 – Coordenadas do sistema horizontal: altura (h) e azimute (A) (SARAIVA, 2017)	29
Figura 3 – Coordenadas equatoriais: ascensão reta (α) e declinação (δ) (SARAIVA, 2017)	30
Figura 4 – Telescópio Newtoniano (WIKIPEDIA, 2023)	30
Figura 5 – Estrutura básica de uma montagem equatorial (KELLY, 2014)	31
Figura 6 – Estrutura básica de uma montagem dobsoniana (KELLY, 2014)	32
Figura 7 – Montagem dobsoniana caseira (ANGELIS, 2019)	32
Figura 8 – Constelação GPS e seus segmentos	34
Figura 9 – Cones de atuação de cada satélite (a) Um satélite (b) Dois satélites (c) Três satélites (ZANOTTA; CAPPELLETTO; MATSUOKA, 2011)	34
Figura 10 – Motor de Ímã Permanente (FEIS.UNESP, 2013)	35
Figura 11 – Motor Relutância Variável (FEIS.UNESP, 2013)	36
Figura 12 – Motor de Passo Híbrido (LOCATELLI, 2021)	37
Figura 13 – Esquema ligação Unipolar (TAMAWAGA, 2021a)	37
Figura 14 – Esquema ligação Bipolar (TAMAWAGA, 2021b)	37
Figura 15 – Funcionamento básico Ponte H(WIKI, 2022)	38
Figura 16 – Encoder Incremental e seu disco (INSTRUMENTS, 2017)	38
Figura 17 – Encoder Absoluto e seu disco (INSTRUMENTS, 2017)	39
Figura 18 – Funcionamento do efeito <i>hall</i> (MANAVELLA, 2023)	40
Figura 19 – (a) Rede sem fio com um ponto de acesso. (b) Rede ocasional (S., 2003)	41
Figura 20 – Exemplo de dispositivos conectador I2C (SEMICONDUCTORS, 2021)	42
Figura 21 – Transferência completa de dados (SEMICONDUCTORS, 2021)	43
Figura 22 – Diagrama de blocos UART (WIKIPEDIA, 2022b)	43
Figura 23 – Quadro UART (ROHDE-SCHWARZ, 2022)	44
Figura 24 – Telescópio base equatorial (Fonte: De autoria própria.)	45
Figura 25 – Representação 3D Montagem Dobsoniana (Fonte: De autoria própria.)	46
Figura 26 – Projeto e caixa de redução confeccionada	47
Figura 27 – Estrutura Dobsoniana parcial e completa (Fonte: De autoria própria)	48
Figura 28 – DOIT ESP-WROOM-32 (ELETROGATE, 2022)	49
Figura 29 – Pinagem ESP32 WROOM DOIT (MOHANAN, 2022)	50
Figura 30 – Curva de Pulsos/s vs Torque (TAMAWAGA, 2021c)	52
Figura 31 – DVR8825 e seu esquemático (INSTRUMENTS, 2022)	53
Figura 32 – CNC Shield V3 (FILIPEFLOP, 2021)	54

Figura 33 – Pinagem CNC Shield V3 (MARTINS, 2022)	55
Figura 34 – Módulo GPS Neo-6M (PROJECTS, 2021)	56
Figura 35 – Encoder Magneto-resistivo e seu funcionamento (INSTRUMENTS, 2022)	57
Figura 36 – Encoder e conjunto do eixo de altitude (Fonte: De autoria própria) . .	58
Figura 37 – Fluxograma de Desenvolvimento (Fonte: De autoria própria)	59
Figura 38 – Relação entre componentes (Fonte: De autoria própria)	60
Figura 39 – Eletrônica acoplada à <i>protoboard</i> (Fonte: De autoria própria)	61
Figura 40 – Imãs na base inferior (Fonte: De autoria própria.)	63
Figura 41 – Sensor Hall A e Sensor Hall B (Fonte: De autoria própria.)	63
Figura 42 – Telescópio modo estacionamento (Fonte: De autoria própria.)	64
Figura 43 – Exibição do endereço de IP do cosmo finder (Fonte: De autoria própria.)	71
Figura 44 – Tela inicial do telescópio (Fonte: De autoria própria.)	72
Figura 45 – Alvo da localização desejada (Fonte: De autoria própria.)	73
Figura 46 – Mapa de inserção da localização (Fonte: De autoria própria.)	73
Figura 47 – Botões de movimento (Fonte: De autoria própria.)	74
Figura 48 – Diagrama de blocos do Cosmo Finder (Fonte: De autoria própria.) . . .	75
Figura 49 – Diagrama de Fluxo do Cosmo Finder (Fonte: De autoria própria.) . . .	76
Figura 50 – Ambiente de simulação (Fonte: De autoria própria.)	77
Figura 51 – Base completa e integrada (Fonte: De autoria própria.)	78
Figura 52 – Distância entre o alvo desejado e do alvo real (Fonte: De autoria própria.)	80
Figura 53 – Júpiter, Io, Europa, Ganímedes e Calisto (Fonte: De autoria própria.) .	81
Figura 54 – Imagem da Lua (Fonte: De autoria própria.)	81
Figura 55 – Diagrama Eletrônico do sistema (NUNES; RODRIGUES, 2023)	91

Lista de tabelas

Tabela 1 – Versões do 802.11 (IEEE, 2022)	41
Tabela 2 – Sentença GPGGA do protocolo NMEA	62
Tabela 3 – Protocolo Cosmo Finder	67
Tabela 4 – Teste de GoTo e rastreamento de Saturno	79
Tabela 5 – Teste de GoTo e rastreamento de Júpiter	80
Tabela 6 – Teste de GoTo e rastreamento da Lua	82

Lista de abreviaturas e siglas

A/D	Analógico/Digital
E/S	Entrada/Saída
EEPROM	Electrically Erasable Programmable Read-Only Memory
D/A	Digital/Analógico
FDM	Fused Deposition Modeling
GPGGA	Global Positioning System Fix Data
I2C	Inter-Integrated Circuit
I2S	Inter-Integrated Circuit Sound
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
LCD	Liquid Cristal Display
LED	Light Emitting Diode
MEMS	Micro-Electro-Mechanical System
GPS	Global Positioning System
NMOS	N-channel metal-oxide semiconductor
NMOS	N-channel metal-oxide semiconductor
NMEA	National Marine Electronics Association
PWM	Pulse Width Modulation
QFN	Quad Flat No Leads
RAM	Random Access Memory
RTC	Real Time Clock
ROM	Read Only Memory
SRAM	Static Random-Access Memory
SDA	Serial Data Line

SCL	Serial Clock Line
SSID	Service Set Identifier
SoC	System-on-a-chip
SPI	Serial Peripheral Interface
UART	Universal asynchronous receiver-transmitter

Lista de símbolos

ϕ	Letra grega phi
α	Letra grega alfa
Ω	Letra grega Omega
δ	Letra grega delta
λ	Letra grega lambda

Sumário

1	INTRODUÇÃO	23
1.1	Descrição do Problema	24
1.2	Objetivos	25
1.3	Organização do Documento	25
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	Astronomia	27
2.1.1	Esfera Celeste	27
2.1.2	Coordenadas Geográficas	28
2.1.3	Coordenadas Astronômicas	28
2.1.4	Telescópio	29
2.2	Montagens	30
2.2.1	Montagem Equatorial	31
2.2.2	Montagem Dobsoniana	31
2.3	Sensores Localização GPS	33
2.4	Motor de Passo	35
2.4.1	Ímã permanente	35
2.4.2	Relutância variável	36
2.4.3	Híbridos	36
2.5	Encoder	38
2.5.1	Encoders Incrementais	38
2.5.2	Encoders Absolutos	39
2.6	Sensores de efeito Hall	39
2.7	Protocolos de comunicação	40
2.7.1	IEEE 802.11	40
2.7.2	I2C	41
2.7.3	UART	42
2.7.4	Considerações finais do capítulo	44
3	ESPECIFICAÇÕES E CONSTRUÇÃO DO SISTEMA DO TELES- CÓPIO	45
3.1	Materiais	45
3.1.1	Telescópio Newtoniano com base Dobsoniana	45
3.1.2	ESP32-WROOM-32 DEVKIT V1	49
3.1.3	Motor de Passo TS3103N40	51
3.1.4	DVR8825	53

3.1.5	CNC SHIELD V3	54
3.1.6	NEO-6M	55
3.1.7	RMS20-256	57
3.2	Fluxo de Desenvolvimento	59
3.2.1	Relação entre componentes do sistema	60
3.2.2	Leitura dos sensores	61
3.2.3	Rotinas do Cosmo Finder	64
3.2.4	Servidor com o driver instanciado	68
3.2.5	Comunicação com o cliente KStars	70
3.2.6	Considerações finais do capítulo	73
4	RESULTADOS	77
4.1	Simulação do telescópio	77
4.2	Testes com o Telescópio	78
5	CONCLUSÃO	83
	REFERÊNCIAS	85
	ANEXOS	89
	ANEXO A – DIAGRAMA ELETRÔNICO DO SISTEMA	91

1 INTRODUÇÃO

Quando o primeiro homem decidiu olhar para cima e registrar as fases da lua ou a posição dos astros, a fim de procurar respostas para seus problemas mundanos, seja para marcar a passagem do tempo, decidir se era hora de plantar, colher ou saber da cheia de rios, de certa forma nascia ali a Astronomia, considerada a mais antiga das ciências.

Essa forma de astronomia primitiva que contava apenas com a capacidade do olho humano perdurou por milhares de anos até a grande revolução causada por um fabricante de lentes holandês em 1608, Hans Lippershey (KING, 1995) quando tentou patentear sem sucesso sua invenção, o Telescópio, que tinha fins primariamente militares, naquela época Holanda e Espanha estavam em guerra e não era de interesse do Estado holandês que aquilo caísse nas mãos inimigas, Hans queria manter sua invenção em segredo, mas a existência de tal objeto se espalhou rapidamente por toda a Europa e já em 1609 Galileu Galilei resolveu criar a sua versão aprimorada e olhar para o céu abrindo um novo universo de descobertas.

Desde então os telescópios não pararam de serem aprimorados, e hoje observamos o Universo em todo seu espectro, das ondas de rádio aos raios gama. O que todos esses grandes observatórios têm em comum são uma estrutura e mecânica refinadas para conseguirem apontar para qualquer lugar do céu e acompanhar o astro compensando a rotação do planeta ou sua órbita, seja por um conjunto de engrenagens ou por rodas de reação. Na astronomia amadora, o termo GoTo (VAUGHAN, 2016), refere-se a um tipo de base de telescópio que juntamente com instruções de um software irá guiar automaticamente o telescópio ao astro desejado, sendo que suas rotinas podem ser geradas por um microcontrolador ou por um computador externo. Os telescópios com o sistema GoTo, hoje são objetos de difícil acesso ao público, por terem manufatura complexa e falta de conhecimento do mesmo, porém com a popularização de plataformas como Arduino (ARDUINO, 2022), ESP32 (SYSTEMS, 2022) e outros microcontroladores, muitos projetos e bibliotecas colaborativas como o INDI (INDI, 2022) foram sendo criados para diminuir o valor de entrada nessa área.

Era impossível imaginar há uma década a possibilidade de possuir um SoC (System on Chip) tão barato, potente e com tantas funcionalidades em relação aos seus concorrentes como a ESP32 da EXPRESSIF. A construção caseira de um sistema completo GoTo ficou extremamente mais acessível financeiramente e utilizando da sua capacidade de conexão sem fio, astrônomos amadores podem controlar seus telescópios e observarem os objetos astronômicos em qualquer lugar do mundo, contribuindo cada vez mais para o avanço da ciência.

1.1 Descrição do Problema

Para os estudantes de física ou astronomia que desejam realizar observações e coletar dados relacionados aos astros, assim como para aqueles que têm o interesse em observar o cosmos como um passatempo, deparam-se com a grande dificuldade de acesso a telescópios profissionais. Essa dificuldade se deve ao fato do alto valor e de que muitos telescópios profissionais possuem códigos fechados, impedindo a modificação de certas funcionalidades por parte do público.

A utilização de telescópios em montagens sem acompanhamento torna bem mais difícil a busca e travamento para observação celeste, sendo preciso corrigir periodicamente a posição da montagem compensando a rotação do planeta.

Com o passar dos anos, o avanço tecnológico proporcionou o uso de microcontroladores mais potentes, o que facilitou a manufatura caseira de sistemas GoTo para astronomia. Muitos projetos foram desenvolvidos utilizando o microcontrolador Atmel AVR, em conjunto com a plataforma de prototipagem Arduino. No entanto, para implementar a conectividade de sistemas periféricos, como Wifi e Bluetooth, foram necessários acréscimos modulares conhecidos como Shields, o que gera custos adicionais para a construção do sistema.

Contudo, o cenário mudou com o surgimento da ESP32, uma plataforma completa em um único chip. A ESP32 oferece o equilíbrio perfeito entre custo e benefício para aplicações como essa. Com a ESP32, não é mais necessário recorrer a múltiplos módulos e Shields para obter as funcionalidades desejadas, reduzindo significativamente os custos e a complexidade de montagem do sistema GoTo.

A proposta consiste em montar um sistema GoTo, com o controle da base do telescópio pela ESP32, criando bibliotecas próprias para um telescópio de montagem Dobsoniana. Sendo que, a ESP32 já nos oferece um vasto uso de componentes integrados, o que tornariam a aquisição dos módulos adicionais por um Arduino muito mais custoso.

Para o controle desses dispositivos, temos o auxílio da biblioteca INDI, a qual é um software livre que faz o controle de equipamentos astronômicos. Seu protocolo é baseado no Instrument Neutral Distributed Interface (INDI) e faz uma ponte de comunicação entre os clientes (notebooks, smartphone, equipamentos), os quais são ambos software e hardware. (INDI, 2022)

Para utilização, iremos embarcar com a ESP32 (SYSTEMS, 2022) o controle dos motores de passo e sensores que fazem parte da base do telescópio, e também a comunicação dos comandos recebidos pelo protocolo INDI, e sua comunicação com o microcontrolador.

1.2 Objetivos

Nesse trabalho temos como visão projetar, simular e implementar um software embarcado para o funcionamento de um sistema GoTo na ESP32. Construir uma base de telescópio projetada para o comportamento do microcontrolador, e desenvolver a comunicação entre o sistema embarcado e o cliente de simulações gráficas de planetários. Feito isso, o objetivo é direcionar o alvo do telescópio para astros do nosso sistema solar e ter seu rastreamento. Sendo assim, temos como objetivos específicos:

- Especificação técnica dos componentes necessários para a implementação do sistema;
- Construção de uma base altazimutal do padrão dobsoniano;
- Integrar motores de passo e sensores à base para o controle do telescópio;
- Calibração dos sensores;
- Implementar uma rede de comunicação sem fio para transmissão de dados;
- Desenvolver uma solução computacional para o controle da comunicação entre um sistema astronômico de guia e os dispositivos embarcados na base e no telescópio.
- Realizar testes para verificar o atingimento do astro alvejado e seu rastreamento.
- Fazer comparações com outros telescópios caseiros.

1.3 Organização do Documento

Esse trabalho está dividido em cinco capítulos, primeiramente com o capítulo 1 informando o objeto de estudo do trabalho, descrevendo, problematizando e marcando os objetivos do mesmo. O capítulo 2, incrementa a fundamentação teórica necessária para o entendimento dos assuntos de astronomia, montagens, sensores, motores de passo e protocolos de comunicação. O Capítulo 3 descreve as especificações do sistema, sendo levantado materiais, fluxos de desenvolvimento. O capítulo 4, mostra os resultados obtidos pela do telescópio criado, fala sobre os protocolos utilizados, calibração, movimentação para o astro e o seguimento do mesmo. Por fim, o Capítulo 5 apresenta as conclusões desse mesmo trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Astronomia

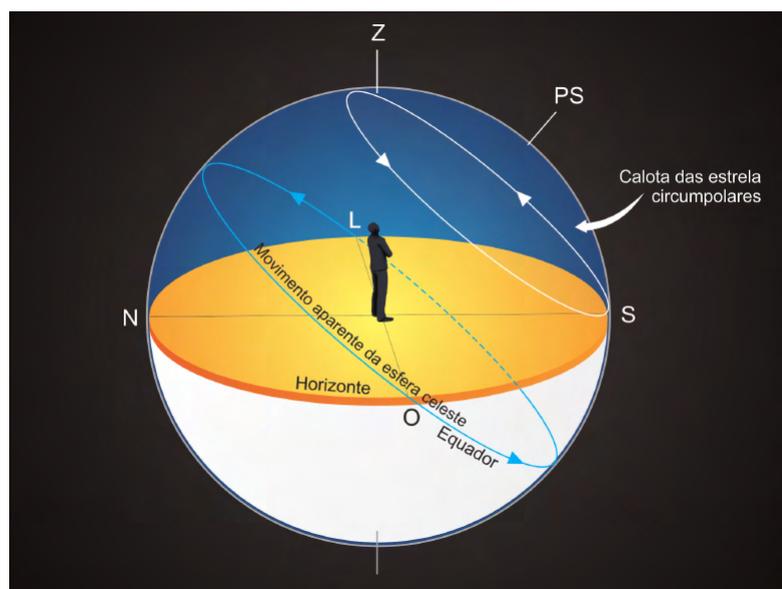
A astronomia é uma ciência que estuda os astros e fenômenos celestes, investigando o universo além da Terra. Seu objetivo é compreender a origem, evolução e comportamento dos corpos celestes, como estrelas, planetas, luas, cometas, asteroides e galáxias, bem como as interações entre eles. Sobre esse assunto, será explicado algumas noções necessárias para a utilização do projeto.

2.1.1 Esfera Celeste

Já usada e inspirada pelos antigos gregos, a esfera celeste é uma esfera imaginária, centrada na Terra, girando em torno de um eixo que é o prolongamento do eixo de rotação do planeta. Essa esfera causa a impressão de que está girando do leste para o oeste, a partir de um eixo imaginário que intercepta dois pontos fixos, o polo sul celeste e o polo norte celeste. (SARAIVA, 2017)

Na Figura 1, podemos ver como é imaginada a esfera celeste. A calota das estrelas circumpolares é a região limitada entre o polo sul celeste (PS) e o círculo branco.

Figura 1 – Esfera celeste visível para um certo lugar no hemisfério sul. (SARAIVA, 2017)



Alguns termos serão mencionados para o melhor entendimento da esfera celeste, sendo eles:

- Horizonte: plano tangente à Terra e perpendicular à vertical do lugar que se encontra o observador.
- Zênite: é o ponto onde a vertical do lugar intercepta a esfera celeste, logo acima do observador.
- Equador celeste: círculo máximo onde o prolongamento do equador da Terra intercepta a esfera celeste.
- Ponto Geográfico sul: é o ponto onde o círculo vertical que passa pelo polo celeste sul intercepta o horizonte.

2.1.2 Coordenadas Geográficas

Como devemos focar em algum astro, é necessário definir um sistema de coordenadas astronômicas. De acordo com (SARAIVA, 2017), temos o uso de coordenadas geográficas que servirão de auxílio para as coordenadas astronômicas. Para as coordenadas geográficas, usadas para medir posições sobre a superfície da terra, temos:

- Longitude geográfica (λ): sendo o arco do equador, com origem no meridiano de Greenwich, com angulação de 90° a 180° para leste ou oeste de Greenwich.
- Latitude geográfica (ϕ): é o arco do meridiano do lugar, com origem no equador, variando entre -90° e $+90^\circ$. Sendo o sinal negativo indicando latitudes do hemisfério sul e positivo do hemisfério norte.
- Ponto Geográfico sul: é o ponto onde o círculo vertical que passa pelo polo celeste sul intercepta o horizonte.

2.1.3 Coordenadas Astronômicas

Incluindo no tópico coordenadas astronômicas, (SARAIVA, 2017) diz sobre o sistema horizontal, que abrange o plano do horizonte celeste, onde é um sistema local, no sentido de que é fixo na Terra. Seus componentes são:

- Azimute (A): o arco medido sobre o horizonte, no sentido horário, com origem no ponto cardeal Norte e fim no círculo vertical do astro. Variando entre 0° a 360°
- Altura (h): sendo o arco medido sobre o círculo vertical do astro, com origem no horizonte e fim no astro. a altura varia entre -90° a $+90^\circ$

Na Figura 2, podemos então ter o entendimento sobre o azimute e a altura em relação a um corpo celeste.

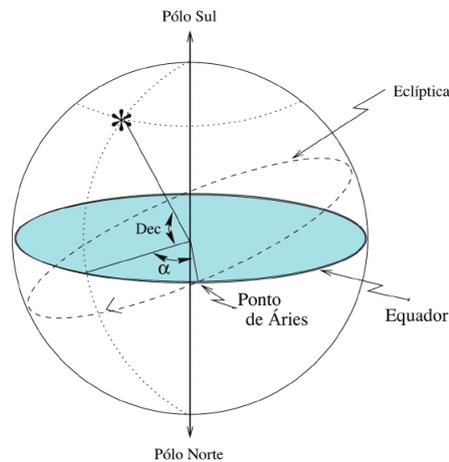
Figura 3 – Coordenadas equatoriais: ascensão reta (α) e declinação (δ) (SARAIVA, 2017)

Figura 4 – Telescópio Newtoniano (WIKIPEDIA, 2023)



2.2 Montagens

Uma montagem de telescópio é uma estrutura mecânica que suporta um telescópio. As montagens do telescópio são projetadas para suportar o peso do telescópio e permitir um apontamento preciso do instrumento. Muitos tipos de montagens foram desenvolvidos ao longo dos anos, com a maioria do esforço colocado em sistemas que podem rastrear o movimento das estrelas fixas à medida que a Terra gira. (WIKIPEDIA, 2022a)

Abaixo abordaremos dois dos tipos mais comuns de montagem com suas vantagens e desvantagens, a Montagem Equatorial e a Montagem Dobsoniana (altazimutal sem tripé).

2.2.1 Montagem Equatorial

Uma montagem para telescópios astronômicos com dois eixos, um dos quais gira em torno de um eixo paralelo ao eixo de rotação da Terra, como na Figura 5. Este eixo pode ser acionado manualmente ou por um motor a uma taxa igual ao movimento diurno do objeto celeste que está sendo visto, permitindo que o telescópio mantenha o objeto à vista por longos períodos sem ajuste, sendo considerada a melhor montagem para fotografias de céu profundo. O outro eixo, perpendicular ao primeiro, é o de declinação, uma vez que o movimento na mesma taxa com que o planeta gira, o astro será acompanhado.

Figura 5 – Estrutura básica de uma montagem equatorial (KELLY, 2014)



Tem como desvantagem ser uma montagem muito mais pesada, pois mesmo para sistemas relativamente leves precisa-se de contrapesos para diminuir o esforço nas engrenagens do sistema e também peca por ser menos estável que a montagem dobsoniana.

2.2.2 Montagem Dobsoniana

A montagem dobsoniana é uma montagem altazimutal que também pode ser chamada por azimute-elevação ou altitude-azimute que consiste em apoiar o telescópio em um sistema de dois eixos perpendiculares onde a rotação em torno do eixo horizontal varia a altitude (cima-baixo) e a rotação em torno do eixo vertical varia o azimute (leste-oeste), como na Figura 6, e um telescópio dobsoniano caseiro como na Figura 7.

John Dobson (DOBSON, 2008), foi quem popularizou a utilização desse tipo de montagem altazimutal com um projeto simplificado e mais leve para a utilização com telescópios refletores newtonianos. A grande facilidade de construção da base com materiais que podem ser facilmente encontrados em lojas de materiais de construção como madeira, plástico ou acrílico fez com que ela seja uma das opções mais baratas e rápidas para entrar na área da astronomia amadora.

Figura 6 – Estrutura básica de uma montagem dobsoniana (KELLY, 2014)

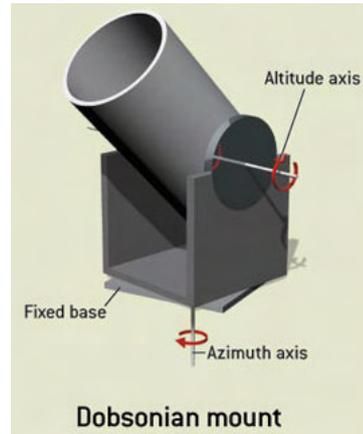


Figura 7 – Montagem dobsoniana caseira (ANGELIS, 2019)



A maior **desvantagem** de uma montagem dobsoniana frente a equatorial é a incapacidade de perseguir objetos astronômicos enquanto a Terra gira no seu eixo, precisando do auxílio computacional para o controle simultâneo dos dois eixos que se movimentam em taxas distintas, enquanto na equatorial uma vez calibrado o eixo polar conforme a localização (Latitude) do telescópio na Terra só é preciso controlar um eixo, ascensão reta em uma taxa fixa conforme a rotação do planeta para acompanhar um astro. Por esse motivo, a montagem dobsoniana não é recomendável para fotografias de espaço profundo onde é necessário acompanhar um objeto com extrema precisão por longos períodos, na casa de horas. Para se alcançar esse nível de precisão numa montagem dobsoniana o custo cresce muito e fica mais viável o investimento em uma montagem equatorial motorizada.

Em relação ao posicionamento da ocular, na montagem dobsoniana sempre é possível encontrar uma posição confortável para executar uma observação, enquanto na equatorial, dependendo da posição do astro, a posição da ocular pode ficar obstruída e a observação inviável sem a utilização de câmeras acopladas ou de um sistema capaz de girar o tubo do telescópio em torno do seu eixo longitudinal. Outra desvantagem em relação a equatorial é que devido à ocular ficar bem no alto da montagem é mais complicado acoplar instrumentos muito pesados como câmeras, por exemplo.

2.3 Sensores Localização GPS

O Sistema de Posicionamento Global (GPS) foi desenvolvido pelo Departamento de Defesa dos Estados Unidos, com o intuito de auxiliar o poder de suas forças armadas, guiando mísseis, localizando tropas e manejo de manobras de combate. Com o tempo, foi permitindo o seu uso para meios civis, onde teve grande aplicação num panorama comercial e científico.

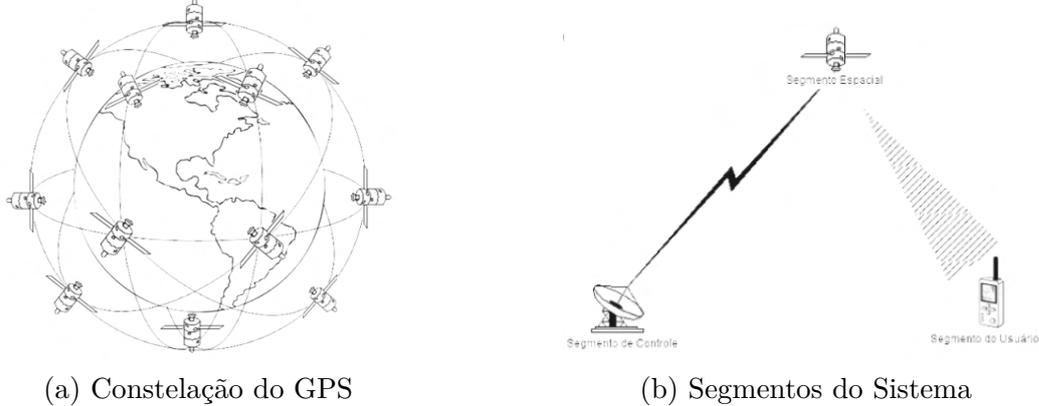
Contando com dois tipos de serviços de posicionamento, o GPS possui o padrão SPS (Standart Position Service) e o preciso PPS (Precise Positioning Service). O SPS está disponível a todos os habitantes do mundo, proporcionando precisões de 10 a 140 metros. Já o PPS, é de uso restrito militar, com precisão de 10 a 20 metros. O GPS é separado em 3 segmentos, o espacial, de controle e de usuários. (ZANOTTA; CAPPELLETTO; MATSUOKA, 2011)

O segmento espacial conta com 24 satélites distribuídos em seis planos orbitais igualmente espaçados, onde cada plano conta com 4 satélites. A partir desses planos, em qualquer local da superfície terrestre é possível que pelo menos 4 satélites estejam ao alcance de comunicação por um receptor. O segmento de controle é responsável pelo monitoramento, correção de relógios e mensagens periódicas de navegação dos satélites, composto por cinco estações terrestres distribuídas estrategicamente no globo. Por fim, o segmento de usuário, se caracteriza pelo receptor do GPS.

Na Figura 8a, mostra a distribuição de satélites pela órbita terrestre, e na Figura 8b, é exibido os segmentos do sistema de controle, que envia os dados para o segmento espacial, e também a transmissão da informação do segmento espacial para o segmento do usuário.

De acordo com (ZANOTTA; CAPPELLETTO; MATSUOKA, 2011), os 24 satélites transmitem sinais de rádio continuamente para a superfície do globo, sendo que nesse sinal contém informações sobre sua posição orbital, vinculado a um referencial geodésico, e o tempo marcado por um relógio atômico interno. Sendo assim, com a informação de um único satélite, o receptor consegue se localizar numa determinada área circular sobre a superfície terrestre. Com a adição da informação de outro satélite, duas áreas circulares

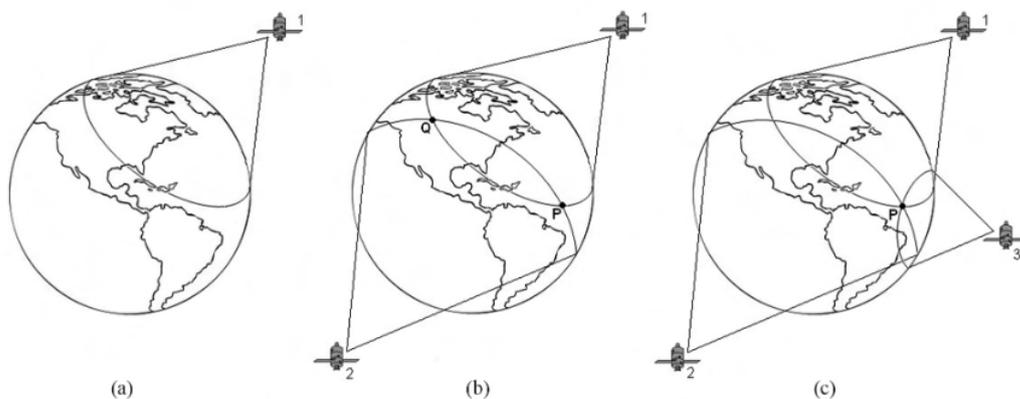
Figura 8 – Constelação GPS e seus segmentos
(ZANOTTA; CAPPELLETTO; MATSUOKA, 2011)



são projetadas na superfície, e com isso existe a interseção de dois pontos. Porém, não é o suficiente para definir a localização atual, resultando então na adição de mais uma informação do terceiro satélite. Com os três satélites, é possível definir um ponto em comum nas suas interseções, e com ele temos a localização atual de um receptor na superfície da Terra.

Na Figura 9, exibe os cones de projeção de comunicação dos satélites. No exemplo 9a, mostra apenas de um satélite, tornando a localização do receptor do GPS em uma área de gigante. Já na 9b, temos dois satélites enviando informações, com suas interseções gerando os pontos Q e P, para a possível localização do receptor. E por fim, na 9c, temos o envio de informação de três satélites, onde suas interseções produzem um ponto único de localização P.

Figura 9 – Cones de atuação de cada satélite (a) Um satélite (b) Dois satélites (c) Três satélites (ZANOTTA; CAPPELLETTO; MATSUOKA, 2011)



Com essas informações da posição orbital e tempo do relógio interno, os dispositivos receptores de GPS podem calcular, como base, a latitude, longitude e elevação em certos dispositivos, podendo então mudar a precisão de diferentes aparelhos utilizados.

2.4 Motor de Passo

Os motores de passo são motores que possuem um controle fino de sua posição de giro via um controle individual de suas bobinas. São motores precisos quando se tem a necessidade controlar o número de ciclos ou até mesmo o ângulo de giro do motor.

Eles são encontrados em equipamentos que exigem controle preciso de ângulos como impressoras 3D, scanners, fresadoras, robôs, etc. Algumas das principais características desse tipo de motor é que eles não possuem escovas e comutadores, usados nos motores convencionais e também necessitam da utilização de drivers para enviar os sinais de controle para o motor.

Para percorrer cada passo, o motor faz um controle entre um ímã e bobinas, que vão variando de acordo com seus polos.

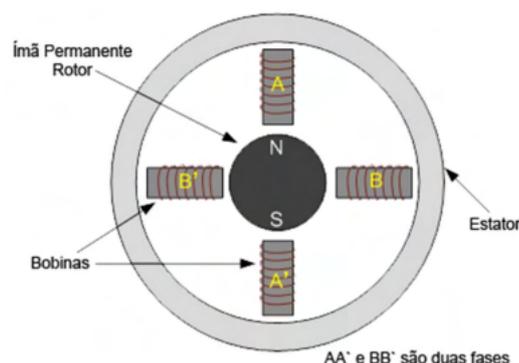
O controle é feito com a mudança de atuação de cada bobina, ou seja, elas se ligam alternadamente, e isso faz com que os ímãs sejam atraídos de acordo com seu campo magnético. Quando uma bobina é acionada, ela produz um campo magnético e o ímã é atraído com sua polaridade inversa, e enquanto as bobinas vão alternando, o motor vai girando.

Os motores de passo possuem 3 tipos básicos, os de ímã permanente, os de relutância variável e os híbridos.([JONES, 2004](#))

2.4.1 Ímã permanente

No caso dos motores de passo com ímã permanente, os ímãs são fixados em um eixo liso no centro do motor. São motores que possuem uma grande potência e um torque maior em sua partida, só que eles acabam perdendo com menos precisão. Quando a bobina do estator é ativada, o eixo se alinha com o campo magnético até o estator ser desligado e o estator seguinte ligado.

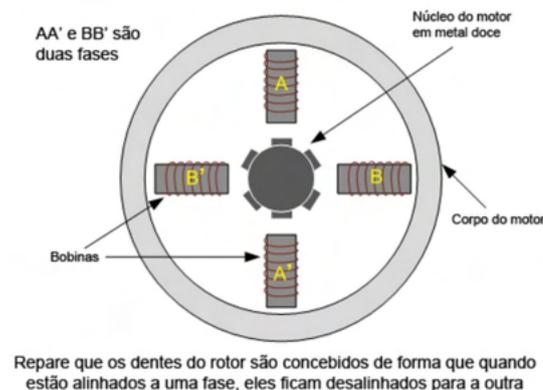
Figura 10 – Motor de Ímã Permanente ([FEIS.UNESP, 2013](#))



2.4.2 Relutância variável

Já os motores de relutância variável, também chamados de motores de relutância variável chaveada, possuem uma característica um pouco diferente do de ímã permanente. Seu eixo é feito de ferro e não com ímãs, e é o campo magnético formado pelas bobinas que faz com que se tenha o controle do eixo. Para ele funcionar é preciso ter seções com dentes no eixo de tal forma que quando eles se alinham com um conjunto de bobina, eles fiquem desalinhados com as outras. Quando o próximo grupo de bobinas é ligado, o eixo se alinha com o outro estator e assim vai repetidamente. O problema desses motores é que eles possuem um torque menor na sua saída.

Figura 11 – Motor Relutância Variável (FEIS.UNESP, 2013)



2.4.3 Híbridos

E por último, temos os motores híbridos, sendo basicamente uma mistura do motor de ímã permanente com o motor de relutância variável. Isso faz com que esse tipo de motor possua um alto torque e tenha uma alta precisão nos passos. O eixo do motor é construído com dois grupos de dentes, sendo um com o polo sul e outro com polo Norte, sendo que eles fiquem alternados em seu próprio eixo.

Por ter ímãs interpolados em seu eixo, isso faz com que se tenha uma precisão muito grande no seu eixo de rotação, podendo variar de 3,6 a 0,9 graus, isso sem precisar de caixa de redução.

Os motores unipolares usam dois enrolamentos por fase e costumam ter um contato em comum, geralmente com 5, 6 ou 8 conexões como mostra a figura abaixo.

Já os motores bipolares, usam uma ligação por polo, e precisam que um circuito controle e possa reverter o sentido da corrente para acionar as bobinas corretamente.

O fato de possuir apenas uma fase torna a ligação de um motor unipolar menos complexa e permite alcançar velocidades mais altas, com um torque menor e mais consumo

Figura 12 – Motor de Passo Híbrido (LOCATELLI, 2021)

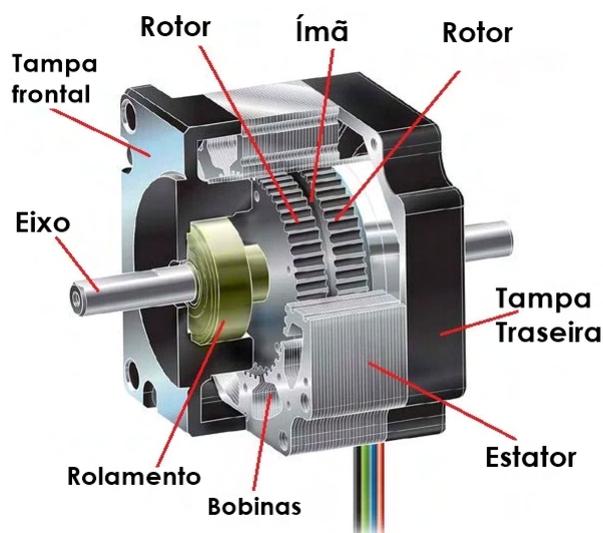


Figura 13 – Esquema ligação Unipolar (TAMAWAGA, 2021a)

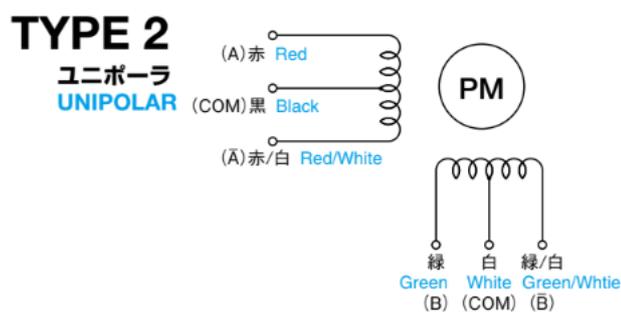
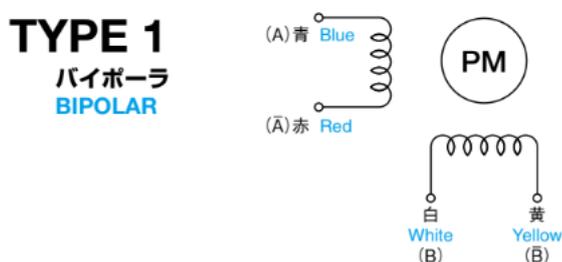


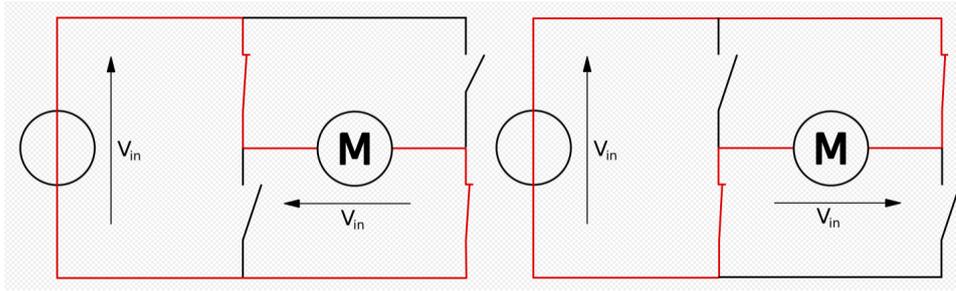
Figura 14 – Esquema ligação Bipolar (TAMAWAGA, 2021b)



de corrente se comparada a ligação bipolar que necessita de drivers mais complexos que permitem a inversão do sentido da corrente para o devido funcionamento do motor.

O circuito básico de controle de um motor bipolar é a ponte H, que pode ser configurada para permitir a passagem de corrente pela bobina em ambas direções. (JONES, 2004)

Figura 15 – Funcionamento básico Ponte H (WIKI, 2022)



2.5 Encoder

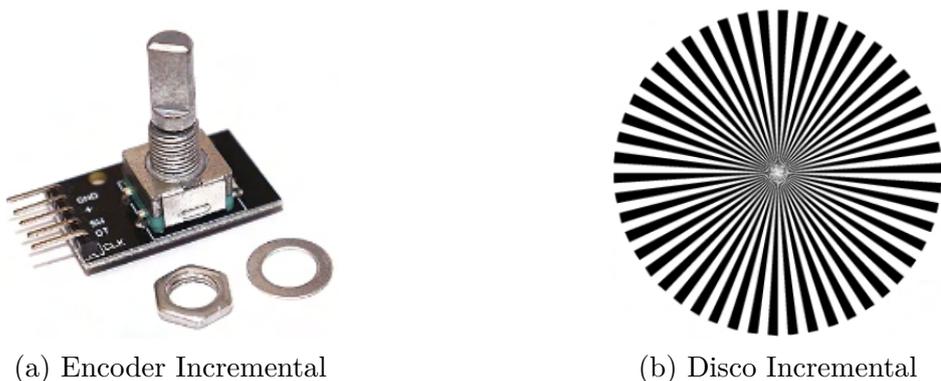
Encoders são dispositivos utilizados para converter movimento mecânico em sinais elétricos ou digitais. Os encoders são frequentemente utilizados para medir a posição, velocidade ou direção de rotação de um eixo, ou sistema mecânico. Eles são amplamente empregados em diversos equipamentos e máquinas, como motores, robôs, impressoras, máquinas CNC (Controle Numérico Computadorizado) e sistemas de controle de movimento.

Existem dois principais tipos de encoders na eletrônica: encoders incrementais e encoders absolutos.

2.5.1 Encoders Incrementais

Esses encoders geram pulsos elétricos ou digitais em resposta ao movimento do eixo, ou objeto a ser medido. Eles não fornecem informações absolutas sobre a posição, mas sim sobre o movimento relativo. O número de pulsos emitidos por um encoder incremental em um ciclo completo de rotação é chamado de resolução. Através da contagem e direção dos pulsos, é possível determinar a posição, velocidade e direção do movimento. Encoders incrementais são amplamente utilizados em aplicações que exigem controle de movimento, como robótica industrial, sistemas de posicionamento e motores de controle de velocidade.

Figura 16 – Encoder Incremental e seu disco (INSTRUMENTS, 2017)



(a) Encoder Incremental

(b) Disco Incremental

2.5.2 Encoders Absolutos

Ao contrário dos encoders incrementais, os encoders absolutos fornecem informações precisas e diretas sobre a posição do eixo sem a necessidade de contagem incremental. Cada posição angular tem um código único associado a ela, permitindo saber a posição exata do eixo em um determinado momento. Isso é particularmente útil em sistemas onde é crucial evitar perdas de informação de posição, mesmo após uma interrupção de energia. Encoders absolutos são frequentemente utilizados em aplicações como máquinas-ferramenta CNC, sistemas de levantamento, robótica médica e aplicações aeroespaciais.

Figura 17 – Encoder Absoluto e seu disco (INSTRUMENTS, 2017)



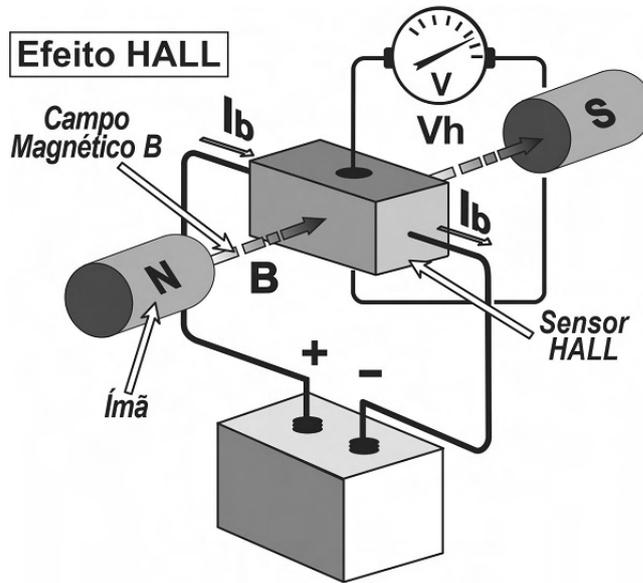
Ambos os tipos de encoders são vitais para o controle de movimento, automação e precisão em uma ampla gama de indústrias eletrônicas e mecânicas. Eles fornecem dados cruciais para sistemas de controle, ajudando a garantir a operação precisa e confiável de máquinas e dispositivos eletrônicos.

2.6 Sensores de efeito Hall

Os sensores de efeito Hall são dispositivos eletrônicos utilizados para medir a intensidade de um campo magnético. Eles são baseados no princípio do Efeito Hall, que foi descoberto por Edwin Hall em 1879. Quando um condutor elétrico (geralmente uma placa fina de material semiconductor) é submetido a um campo magnético perpendicular à direção da corrente elétrica que o atravessa, surgem diferenças de potencial (tensões) em suas extremidades, como pode ser verificado na Figura 18.

O fenômeno ocorre da seguinte forma: os portadores de carga (elétrons ou lacunas) que estão se movendo no condutor são desviados pela ação do campo magnético, acumulando mais carga de um lado do material do que do outro. Isso cria uma diferença de potencial entre as extremidades do condutor, resultando em uma tensão proporcional ao campo magnético aplicado.

Figura 18 – Funcionamento do efeito hall (MANAVELLA, 2023)



Os sensores de efeito Hall são comumente utilizados em diversas aplicações, como em sistemas de controle de motores (para medir a posição do rotor), na medição de correntes elétricas, em dispositivos de detecção de proximidade e também em instrumentos de medição de campo magnético.

2.7 Protocolos de comunicação

2.7.1 IEEE 802.11

Com o surgimento das LANs sem fios, surgiu a necessidade de padronização desse tipo de comunicação, sendo que a partir dos outros padrões LANs que começaram com 802.1, 802.2, até o 802.10; o padrão LAN sem fio recebeu a denominação 802.11, também comumente chamado de WiFi. (S., 2003)

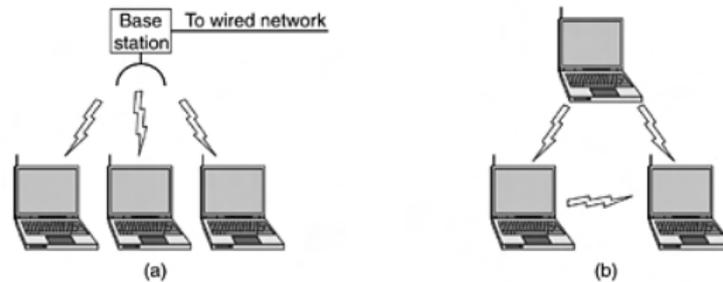
Trabalhando no espectro de bandas não licenciadas, como as bandas ISM (Industrial, Scientific, and Medical) definidas pela ITU-R, as mais conhecidas hoje usadas pelo protocolo a banda de 2,4-2,5 GHz e também a 5,725-5,825 GHz). Os dispositivos têm permissão ao uso de espectro, desde que limite a potência de transmissão para que diferentes dispositivos coexistam na área.

O 802.11 comportam clientes (smartphones, notebooks, Smart TVs, dentre outros dispositivos) e também infraestrutura chamadas pontos de acesso ou APs (Access Points), uma vez que todos os clientes passam por um ponto de acesso. E, outra infraestrutura conhecida como rede ocasional ou rede ad hoc, já que opera pela comunicação direta de um ou mais computadores.

Na Figura 19, a letra a caracteriza o modo da infraestrutura de ponto de acesso,

sendo que três computadores estão ligados ao um ponto de acesso (Base station), e ele direcionando a conectividade a um fio da rede (wired network). E na letra b, é o modo da estrutura rede ocasional (ad hoc), onde três computadores comunicam entre si sem a necessidade de um AP.

Figura 19 – (a) Rede sem fio com um ponto de acesso. (b) Rede ocasional (S., 2003)



Ao decorrer das gerações houve diversas mudanças no protocolo 802.11, onde na tabela abaixo é enfatizado a velocidade, e frequência do rádio.

Tabela 1 – Versões do 802.11 (IEEE, 2022)

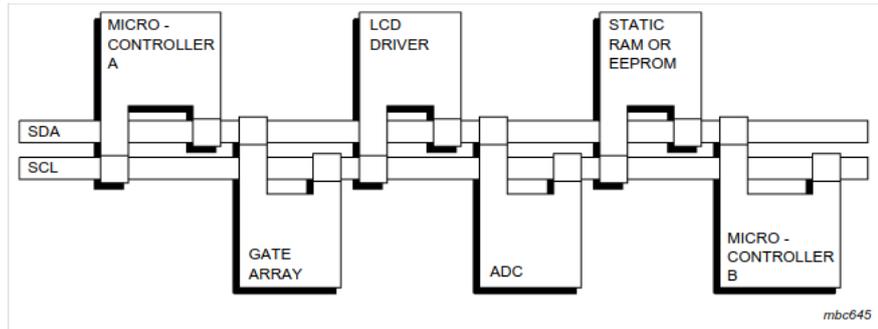
Padrão IEEE	Taxa de transmissão (Mbits/s)	Ano	Frequência (GHz)
802.11	1 - 2	1997	2.4
802.11b	1 - 11	1999	2.4
802.11a	6 - 54	1999	5
802.11g	6 - 54	2003	2.4
802.11n	72 - 600	2008	2.4/5
802.11ac	433 - 6933	2014	5
802.11ax	600 - 9608	2019	2.4/5

2.7.2 I2C

O I2C é um barramento serial usada para a comunicação de vários dispositivos externos com o mesmo tipo de barramento. Podendo ser sistemas de controle inteligente como um microprocessador, drivers de LCD e LED, portas de E/S, RAM, EEPROM, conversores A/D e D/A. Esse tipo de barramento que consiste em dois fios conectados, sendo eles o SDA (Serial Data Line) e o SCL (Serial Clock Line). Como é um barramento de múltiplos controles, existe a detecção de colisões e arbitrariedade para prevenir à corrupção dos dados, para que dois ou mais controladores iniciarem simultaneamente a transmissão. (SEMICONDUCTORS, 2021)

Na Figura 20, podemos ver um exemplo de múltiplos dispositivos conectados ao barramento I2C, pelo SDA e também com o SCL.

Geralmente o dispositivo trabalha ou no modo “master” ou no modo “slave”, sendo que podem ser um ou vários modos interagindo na mesma comunicação do barramento.

Figura 20 – Exemplo de dispositivos conectador I2C ([SEMICONDUCTORS, 2021](#))

Dependendo da variedade de dispositivos de tecnologia, os níveis lógicos '0' (BAIXO) e '1' (ALTO) para o SDA e SCL, não são fixos e dependem do nível associado ao Vdd.

A comunicação inicia-se quando o “master” envia a condição de início, enviando um sinal de borda de descida no SDA, e permanece ALTO no SCL, mandando então nove pulsos de clock para a linha SCL. Os oito primeiros pulsos são usados identificar um byte considerando 7 bits de endereço e 1 bit de escrita/leitura. Se o “slave” com esse endereço está ativado no barramento, ele somente poderá responder lançando o valor para BAIXO do SDA no nono pulso de clock como ACK (Confirmação) . O “master” pode então mandar mais 9 pulsos de clock sendo 8 bits de dados e 1 de ACK, e dependendo do valor enviado do bit de leitura/escrita, o modo “slave” ou “master” vai deslocar os dados para a linha SDA, com o outro lado recebendo a confirmação da transferência (ACK), jogando o valor do SDA para baixo no nono pulso de clock. Durante essa transferência de dados, a linha do SDA só muda quando a linha SCL está com o valor baixo. Portanto, quando o dispositivo “master” terminou a comunicação, ele vai mandar uma condição de parada para o barramento de comunicação, pela borda de subida do SDA, enquanto o SCL ainda está com o valor de ALTO. ([SYSTEMS, 2022](#))

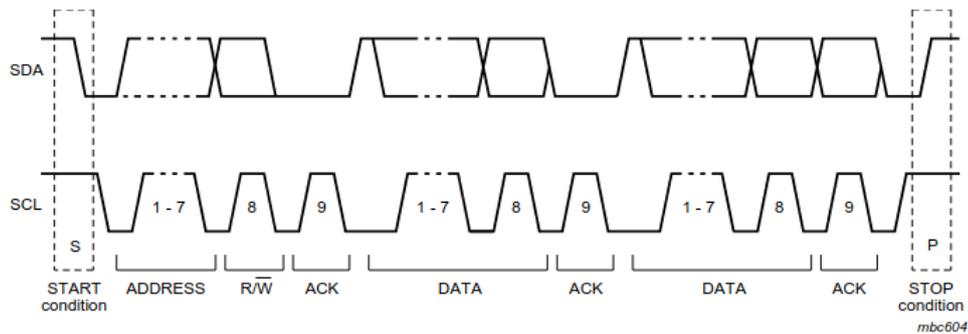
Na Figura 21, é disponível a transferência completa dos dados pelo sinal, onde temos a condição de Início no “START condition”, depois é passado o endereço “ADDRESS”, a leitura/escrita “R/W” e o ACK, depois é transferido os dados “DATA” de oito bits, e por final o ACK. Após transferido os dados, a condição de parada é lançada “Stop condition”.

2.7.3 UART

O protocolo UART, desenvolvido por Gordon Bell da Digital Equipment Corporation, veio com a inovação principal do uso de "sampling" para converter o sinal para o domínio digital, permitindo então mais confiabilidade da cronometragem do que em circuitos anteriores que usavam cronometragens analógicas com o ajuste manual de potenciômetros. ([WIKIPEDIA, 2022b](#))

A UART usa uma comunicação serial assíncrona, sendo que o formato dos dados

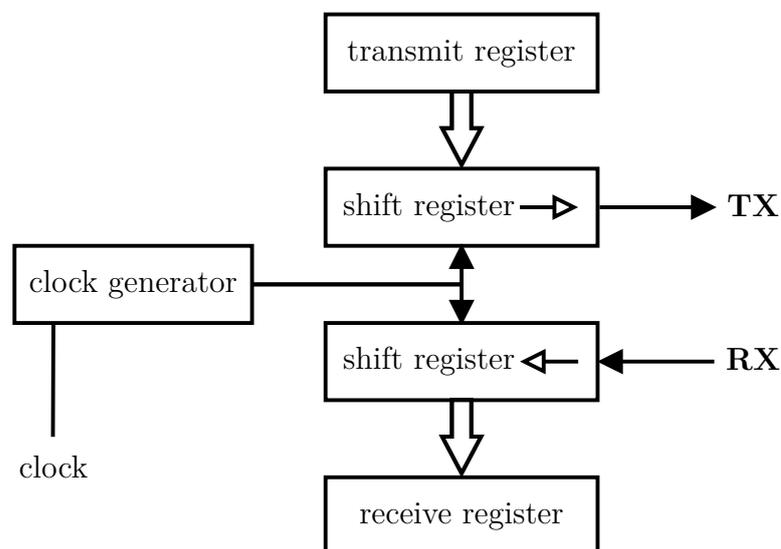
Figura 21 – Transferência completa de dados (SEMICONDUCTORS, 2021)



e a velocidade de transmissão são configuráveis. Manipulando então os bytes de dados e transmitindo eles em bits individuais de forma sequencial. No seu destino, a UART que receber esse sinal, reconstrói esses valores em bytes completos. A UART contém um shift register, utilizado na conversão entre formas paralelas e seriais, o que a faz ter a transmissão do dado serialmente via um fio ou paralelamente via múltiplos fios.

A Figura 22 mostra o diagrama de blocos do protocolo, tendo registradores de transmissão e recepção no seu início e fim, também registradores de deslocamento, sendo eles responsável pela transmissão de dado da saída serial TX, e o recebimento de dados da entrada serial RX.

Figura 22 – Diagrama de blocos UART (WIKIPEDIA, 2022b)



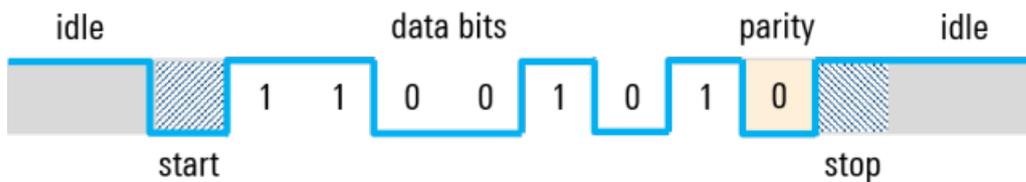
A comunicação pode vir com 3 modos, sendo eles simplex (somente uma direção), half duplex (toma turnas em transmitir e receber), e full duplex (ambos transmitem e recebem). Tendo como quadro de dados, a UART tem que seguir os padrões em ambos os dispositivos, o que recebe e o que transmite. O quadro consiste em 5 elementos:

- Espera (Idle)

- Bit início (Start bit)
- Bits de dados (Data bits)
- Bit de paridade (Parity bit)
- Parada (Stop)

A Figura 23 mostra como o quadro se comunica bit a bit, de acordo com o clock.

Figura 23 – Quadro UART (ROHDE-SCHWARZ, 2022)



2.7.4 Considerações finais do capítulo

Com os estudos da fundamentação teórica, podemos começar a unir as peças necessárias para a aplicação do sistema GoTo. Utilizar meios para realizar testes, e verificar a veracidade dos resultados pela comparação do que foi estudado.

O presente capítulo proporcionou uma sólida base de conhecimento através dos estudos da fundamentação teórica, essencial para aplicação do sistema GoTo no telescópio. Com o embasamento adquirido, tem-se o preparo para continuar o projeto.

O próximo passo consiste em realizar testes e fazer verificações para comprovar a eficácia e precisão do sistema desenvolvido. No processo de implementação, precisa-se considerar as especificações técnicas dos componentes, bem como a integração dos motores de passo e sensores à base do telescópio. Com a calibração dos sensores, tem-se a garantia de um funcionamento adequado do sistema, o que acarreta a correta orientação e rastreamento dos astros.

O próximo capítulo apresentará a metodologia detalhada que guiará as ações no desenvolvimento do projeto e concretização dos objetivos propostos.

3 ESPECIFICAÇÕES E CONSTRUÇÃO DO SISTEMA DO TELESCÓPIO

3.1 Materiais

3.1.1 Telescópio Newtoniano com base Dobsoniana

Utilizando um telescópio Newtoniano com espelho parabólico de 160 mm construído como projeto da disciplina de Introdução à Astronomia e Astrofísica com professor Wytler Cordeiro dos Santos que possuía uma montagem equatorial básica.

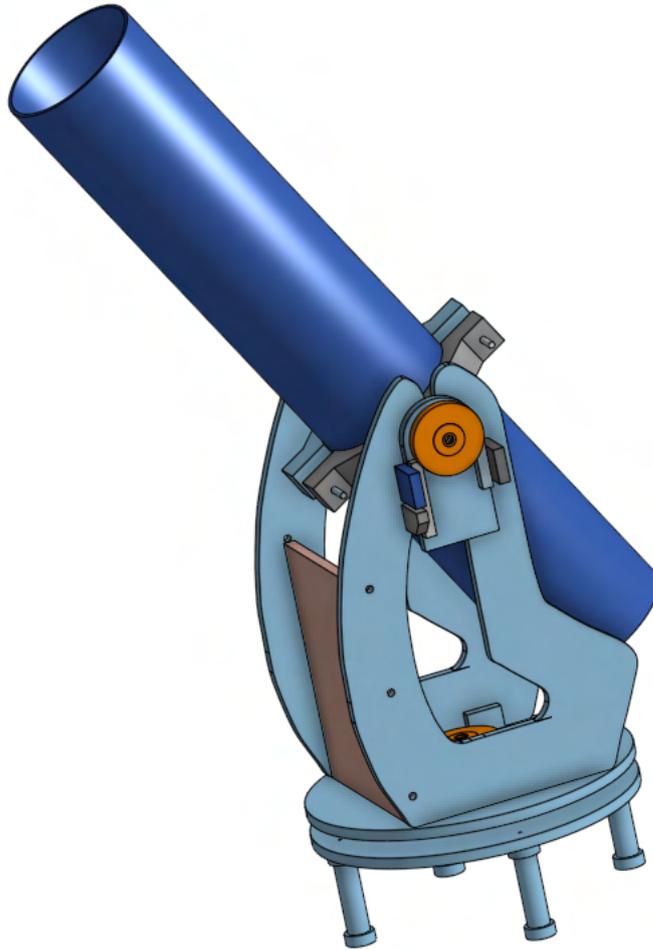
Figura 24 – Telescópio base equatorial (Fonte: De autoria própria.)



A base equatorial atual foi construída com caráter provisório, pois é muito pesada, com cerca de 30 kg, e é de difícil operação, pois necessita da utilização de ferramentas(chave de fenda, alicate) para destravar seus eixos e não possui nenhum mecanismo para auxiliar o observador sobre sua orientação espacial.

Um dos principais objetivos deste trabalho é desenvolver uma nova base Dobsoniana, seguindo um processo semelhante ao exemplificado na Figura 7. Para a criação dessa base, nos inspiramos no tutorial fornecido pelo pesquisador Eder Martioli ([MARTIOLI, 2009](#)), vinculado à divisão de Astrofísica do INPE (Instituto Nacional de Pesquisas Espaciais). A partir dessas referências, projetamos nossa própria base, detalhada na Figura 25. Para esse fim, foi utilizado o software CAD online ([ONSHAPE, 2023](#)). O projeto 3D completo pode ser acessado no GitHub ([NUNES; VINÍCIUS, 2023](#)) deste projeto.

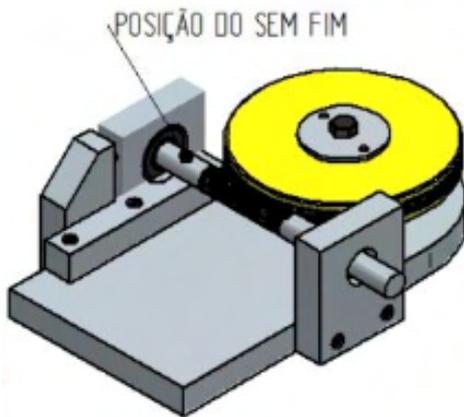
Figura 25 – Representação 3D Montagem Dobsoniana (Fonte: De autoria própria.)



Anteriormente seriam utilizadas engrenagens impressas em 3D para a movimentação dos eixos de altitude e azimute da montagem. No entanto, surgiram preocupações em relação ao desgaste e às folgas dessas engrenagens, o que poderia resultar em erros de calibração do sistema. Diante disso, optou-se por revisar nossa estratégia e buscar por outras abordagens de mecanismos. Durante essa busca, foi encontrado um projeto completo de duas caixas de redução no site do projeto FireGoTo ([LORENSINI, 2023b](#)).

A rosca sem fim é confeccionada em Aço 1020 e a coroa sem fim em poliacetal um polímero largamente utilizado na indústria, e possui ótimas características de resistência a impacto e tração, baixo coeficiente de atrito, elevada rigidez, e que pode ser facilmente usinado ([INTERSTATE, 2023](#)). Tanto a rosca e coroa sem fim foram encomendadas ao custo de R\$ 120,00 reais cada par, o resto das peças necessárias (parafusos, rolamentos, madeira) para finalizar as caixas somaram cerca de R\$ 60,00. O total para a construção das caixas foi de cerca de R\$ 300,00. Ambas as caixas possuem uma relação de 182:1, são necessárias 182 voltas da rosca sem fim para uma volta da coroa. Foram impressos utilizando uma impressora 3D o suporte do motor de passo NEMA 23 ([TAMAWAGA, 2021d](#)), o suporte e acoplador dos encoders que também vão acoplados nas caixas de

Figura 26 – Projeto e caixa de redução confeccionada



(a) Caixa de Redução - Eixo de Altitude (LORENSINI, 2023a)



(b) Caixa de Redução Confeccionada

Fonte: De autoria própria

redução.

Todas as relações entre engrenagens, correias e motores são fundamentais para a correta calibração e rastreamento do sistema, permitindo calcular a máxima resolução do nosso equipamento que atualmente é de $11,868131844''$ segundo de arco por passo.

Atualmente, foi utilizado o motor de passo no modo “full-step” para obter o maior torque e velocidade possíveis. Contudo, esse modo tem a desvantagem de gerar mais vibração na montagem, o que pode prejudicar as observações astronômicas. Embora se opte pelo modo “half-step” para reduzir a vibração, o motor já está operando com o menor ciclo de *clock* que a ESP-32 pode fornecer antes que o motor comece a perder passos. Mudar para o modo “half-step” dobraria o tempo necessário para movimentar o telescópio do ponto A ao ponto B.

O motor requer 200 passos para dar uma volta completa, sendo acoplado a uma polia dentada de 20 dentes. Uma correia dentada do tipo GT2 conecta essa polia de 20 dentes a outra polia de 60 dentes, que por sua vez está acoplada ao eixo da rosca sem fim. Essa configuração cria uma relação de 3:1 entre o eixo do motor e o eixo da rosca sem fim. Considerando que a relação da rosca com a coroa, ambas sem fim, é de 182:1, podemos calcular a quantidade de passos necessários no motor de passo para completar uma volta completa, ou seja, 360° no eixo da coroa sem fim.

Para isso, basta multiplicar o número de passos do motor para uma volta completa pelo maior fator da relação entre as duas polias e depois multiplicar pelo maior fator da relação entre a rosca sem fim e a coroa sem fim. A variável X , manterá esses números de

passos.

$$X = 200 \text{ passos} * 3 * 182 = 109.200 \text{ passos} \quad (3.1)$$

Portanto, são necessários 109.200 passos para que um eixo complete uma volta de 360°.

Com esse dado em mãos, também pode-se determinar quantos graus cada passo em cada motor gira durante o movimento dos eixos de altitude e azimute do telescópio, que será denominado de variável Y .

$$Y = 360^\circ / 109.200 = 0,00329670329^\circ \text{ por passo} \quad (3.2)$$

Para uma melhor compreensão, tem-se a conversão desse valor em graus para segundos de arco, que será atribuído à Z .

$$Z = 0,00329670329^\circ * 3600 = 11,868131844'' \text{ segundos de arco por passo} \quad (3.3)$$

O resto da estrutura foi confeccionada em madeira MDF de 15mm de espessura, seguindo o projeto presente na Figura 25. Os pés são de plástico e possuem regulagem de altura de modo que permitem deixar a base nivelada mesmo em locais um pouco acidentados.

Figura 27 – Estrutura Dobsoniana parcial e completa (Fonte: De autoria própria)



(a) Estrutura Dobsoniana



(b) Montagem Dobsoniana com tubo

3.1.2 ESP32-WROOM-32 DEVKIT V1

O ESP32-WROOM DevKit é uma plataforma de desenvolvimento baseada no chip ESP32 da Espressif Systems. O ESP32 é um microcontrolador de baixo custo e alto desempenho que integra Wi-Fi e Bluetooth em um único chip, tornando-o uma opção popular para projetos de IoT (Internet das Coisas). Ele oferece uma variedade de recursos e interfaces, incluindo GPIOs, UART, SPI, I2C e ADC, permitindo que desenvolvedores criem uma ampla gama de aplicações, desde dispositivos conectados à Internet até projetos de automação residencial e wearables. Além disso, o DevKit é compatível com diversas IDEs, como o Arduino IDE e o ESP-IDF (Espressif IoT Development Framework), tornando-o uma escolha versátil e acessível para entusiastas e profissionais de desenvolvimento de eletrônicos.

Baseado nela decidimos utilizar a placa CNC shield V3 [3.1.5](#) como driver de controle para realizar a ligação com os motores de passo dos eixos de altitude e azimute do telescópio.

Figura 28 – DOIT ESP-WROOM-32 ([ELETROGATE, 2022](#))



O chip ESP-WROOM-32 possui como principais características:

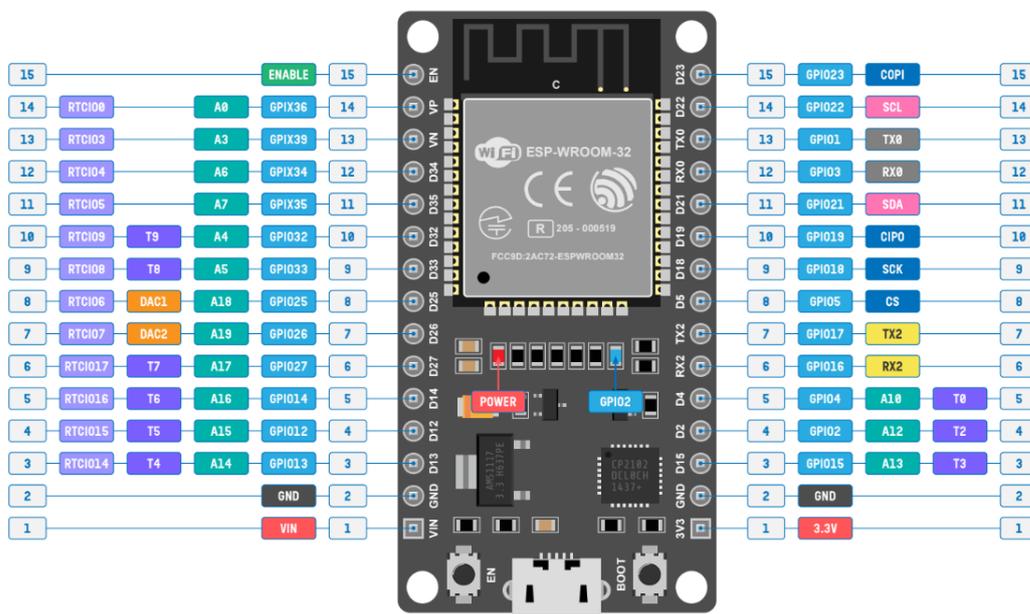
- Chip com WiFi embutido: padrão 802.11 B/G/N, operando na faixa de 2.4 a 2.5 GHz. No padrão 802.11/N capaz de atingir transferências de até 150 Mbps;
- Modos de operação: Client, Access Point, Station+Access Point;
- Coprocessador dual core Tensilica Xtensa 32-bit LX6;
- Clock ajustável de 80 MHz até 240 MHz;
- Tensão de operação: 3.3 V (DC);
- 448 KB de ROM para inicialização e funções principais;
- 520 KB de SRAM no chip para dados e instruções;
- 8 KB RTC FAST Memory que pode ser utilizada como armazenamento de dados e também é acessada pela CPU principal durante a inicialização RTC a partir do modo Deep-sleep;

- 8 KB RTC SLOW Memory e pode ser acessado pelo coprocessador durante o modo Deep-sleep;
- Possui memória flash externa de 32MB (4 megabytes);
- Corrente máxima por pino é de 12mA (recomenda-se usar 6mA);
- Possui 34 GPIOs;
- SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC;
- Possui Bluetooth v4.2 BR / EDR e BLE (Bluetooth Low Energy);

A integração de Bluetooth, Bluetooth LE (Low Energy) e Wi-Fi garante que uma ampla gama de aplicativos possa ser utilizada: o uso de Wi-Fi permite um grande alcance físico e conexão direta à Internet via um roteador Wi-Fi, enquanto o uso do Bluetooth permite que o usuário se conecte convenientemente ao um dispositivo e transmita dados sem gastar muita energia. Quando em modo “deep-sleep” a corrente no chip ESP32 é inferior a 5 mA, tornando-o adequado para aplicações eletrônicas vestíveis e alimentadas por bateria. O módulo suporta uma taxa de dados de até 150 Mbps, e potência de saída de 20 dBm na antena para garantir o maior alcance físico. Como tal, o módulo oferece especificações líderes do setor e o melhor desempenho para integração eletrônica, alcance, consumo de energia, e conectividade.(SYSTEMS, 2022)

Na Figura 29 é possível visualizar com mais clareza todas a portas presentes na ESP-WROOM-32 DOIT.

Figura 29 – Pinagem ESP32 WROOM DOIT (MOHANAN, 2022)



O software que foi embarcado na ESP-32 é baseado no FreeRTOS ([FREERTOS, 2023a](#)). O FreeRTOS (Real-Time Operating System) é um sistema operacional de tempo real de código aberto, projetado para sistemas embarcados e dispositivos com recursos limitados. Desenvolvido por Richard Barry, o FreeRTOS oferece um kernel leve e escalonável que permite a execução de múltiplas tarefas concorrentemente, com suporte a preempção e priorização de tarefas em tempo real.

O objetivo principal do FreeRTOS é fornecer uma plataforma de execução robusta e eficiente para dispositivos que necessitam de determinismo e sincronização em suas operações. Esse sistema operacional é amplamente utilizado em uma variedade de aplicações, incluindo microcontroladores, sistemas de comunicação, dispositivos médicos, automóveis, sistemas de segurança, entre outros.

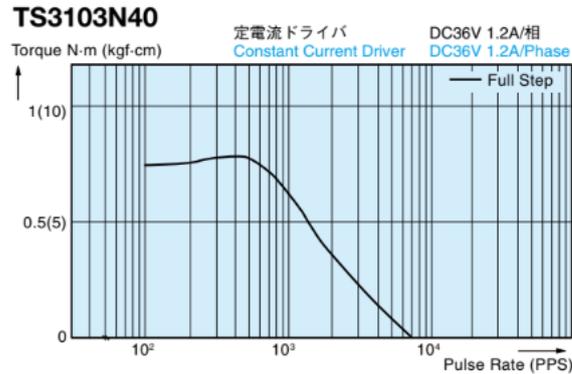
3.1.3 Motor de Passo TS3103N40

Os motores que serão utilizados para o controle altazimutal do telescópio serão desse tipo. Dois motores *Tamagawa TS3103N40 Nema 23*, suas características podem ser vistas com mais detalhes abaixo.

- Modelo: Tamawaga TS3103N40;
- Ângulo de Passo: 1.8° ;
- Tensão de Operação: 6 V / fase;
- Corrente de Operação: 1.2 A;
- Resistência do Enrolamento: 5Ω ;
- Indutância: 10 mH;
- Torque de retenção: 0.5 N.m / 5.0 kgf.cm;
- Comprimento do Motor: 56 mm;
- Tipo de Enrolamento: 2;

O motor possui 6 fios e pode ser utilizado em dois modos: unipolar e bipolar. Os motores serão conectados utilizando a ligação bipolar que segundo Dr. Douglas W. Jones e Reston Condit ([JONES, 2004](#)) oferece até 30% mais torque que a ligação unipolar, isso ocorre porque devido à derivação central na ligação unipolar em qualquer momento de tempo somente, metade do enrolamento das bobinas estão energizados e gerando campo magnético enquanto na ligação bipolar toda a bobina está energizada.

Figura 30 – Curva de Pulsos/s vs Torque (TAMAWAGA, 2021c)



Na Figura 30, é evidente a queda significativa do torque à medida que a velocidade de rotação do motor é aumentada. Inicialmente, o motor foi alimentado com uma fonte de 12V e corrente máxima de 1A, o que era suficiente para movimentos em baixas velocidades na base. Entretanto, ao considerando a necessidade de 109.200 passos para uma volta de 360°, foi decidido aumentar a velocidade para reduzir o tempo de movimentação. Esse aumento resultou em perdas de passo e diminuição do torque, como explicado no tópico 3.1.1.

Para resolver esse problema, foi adicionado um “*step-up*”, um tipo de conversor de energia que aumenta a tensão elétrica de uma fonte de energia para um valor mais elevado. Esse dispositivo é comumente utilizado para elevar a tensão de baterias ou fontes de alimentação de baixa voltagem, tornando-os adequados para dispositivos que exigem uma tensão mais alta para funcionar corretamente.

Ao aumentar a tensão para 32V e manter a corrente do *driver* DVR8825 em 1A, conseguimos solucionar o problema de baixo torque e atingir o limite da capacidade do *tickrate*(FREERTOS, 2023b) do FreeRTOS, que é de 1 milissegundo. A taxa de ticks no FreeRTOS refere-se à frequência com que ocorre a interrupção do tick do sistema operacional.

A interrupção de *tick* é uma interrupção periódica gerada por um temporizador de hardware e serve como mecanismo de cronometragem do sistema. Ele é usado para vários propósitos internos, incluindo agendamento de tarefas, gerenciamento de tempo e cronômetros de software.

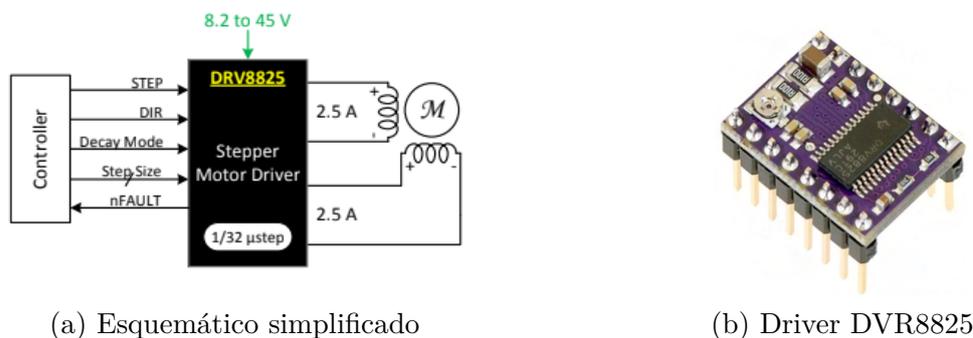
A taxa de *ticks* geralmente é configurável durante a inicialização do FreeRTOS e pode ser ajustada para atender aos requisitos do aplicativo e da plataforma de hardware. Por exemplo, uma *tickrate* de 1 milissegundo (1 kHz) significa que a interrupção do tique ocorre a cada milissegundo, fornecendo uma resolução de tempo de 1 ms para agendamento de tarefas e outras operações relacionadas ao tempo. Com essa configuração otimizada a desempenho do sistema melhorou, proporcionando movimentos mais precisos e eficientes na base do telescópio.

O aumento de torque da ligação bipolar não vem sem o preço de que para utilizar essa ligação o motor precisa de um controlador bem mais complexo e por consequência mais caro, que no nosso caso será o *driver* DVR8825. Esse *driver* é o responsável por receber o sinal da ESP32 para controlar a quantidade de corrente e os passos que o motor receberá, fazendo com que se tenha o controle fino necessário para nossa necessidade. Mais informações sobre as características desse componente podem ser vistas logo abaixo.

3.1.4 DVR8825

Para executar o controle dos motores de passo bipolares utilizaremos o *driver* DVR8825. Suas características encaixam perfeitamente para a nossa aplicação. O dispositivo contém dois NMOS Pontes H, sensor de corrente, circuitos de regulação e um indexador de micropasso. O DRV8825 pode ser alimentado com uma tensão de alimentação entre 8,2 V e 45 V e consegue fornecer uma corrente de saída de até 2,5 A em escala real. Uma interface STEP/DIR simples permite uma interface fácil com o circuito do controlador. Possui limitação de corrente ajustável, proteção contra sobretensão e seis resoluções de passo ($\frac{1}{1}$, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ e $\frac{1}{32}$).

Figura 31 – DVR8825 e seu esquemático (INSTRUMENTS, 2022)



Para configurar a corrente que será fornecida ao motor, é realizado o ajuste através da tensão de referência presente no *trimpot*. A tensão de referência respeita a seguinte fórmula.(INSTRUMENTS, 2022) onde V_{ref} é a tensão de referência, R_{sense} é o valor do resistor de sensibilidade presente no *driver* e I_{max} é a corrente máxima suportada pelo motor.

$$V_{ref} = 5 * R_{sense} * I_{max} \quad (3.4)$$

É limitado para evitar possível superaquecimento do motor a corrente máxima fornecida pelo *driver* em 70% da corrente máxima de operação do motor TS3103N40 que é 1,2 A com o R_{sense} igual a R250 (0.25 Ω), temos:

$$V_{ref} = 5 * 0.25 * 1,2 * 0,7 \quad (3.5)$$

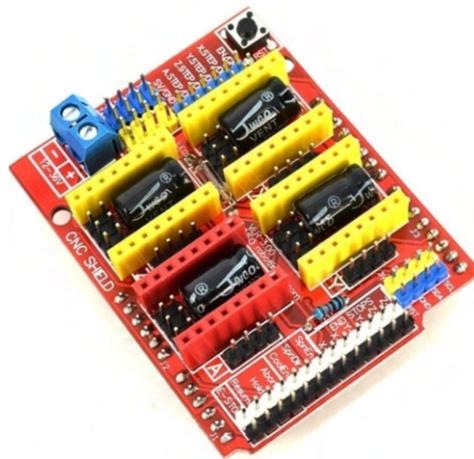
$$V_{ref} = 1,05 V \quad (3.6)$$

Os *drivers* serão conectados a um *shield* que intermediará a conexão com a ESP32.

3.1.5 CNC SHIELD V3

O CNC Shield foi projetado pela Protoneer.co.nz para aproveitar a demanda por um controlador de baixo custo para soluções de máquinas CNC caseiras. Ele foi projetado para ser 100% compatível com Grbl, o Open-source G-Code interpretador, e se encaixa no popular Arduino Uno. O CNC Shield pode ser usado para controlar vários tipos de máquinas CNC, incluindo fresadoras CNC, máquinas de gravação/corte a laser, máquinas de desenho, impressoras 3D ou qualquer projeto que precise de controle de precisão de motores de passo. (KRUGER, 2015)

Figura 32 – CNC Shield V3 (FILIPEFLOP, 2021)

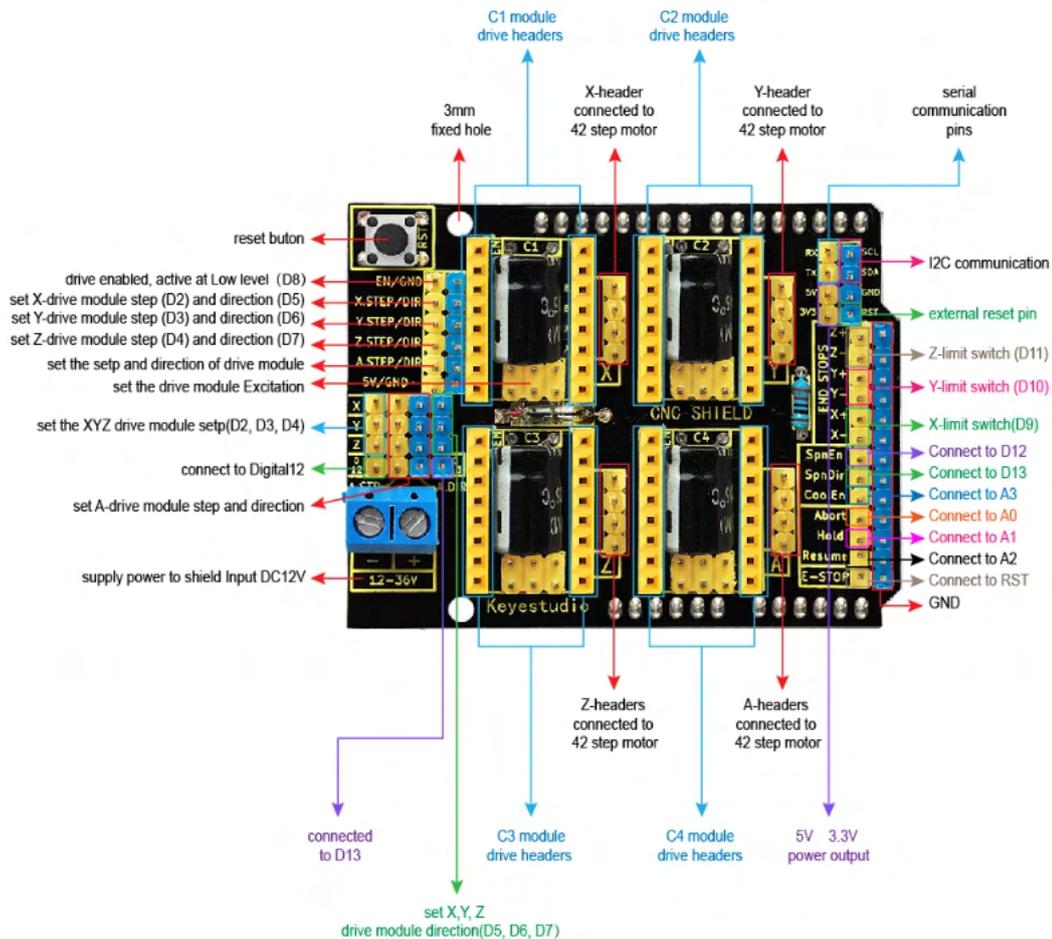


Especificações:

- Possui 4 conectores para drivers controladores de passo. (A4988, DRV8825, TMC2209 entre outros).
- Suporta fontes de alimentação entre 12 V e 36 V;
- Suporta 4 eixos (X, Y, Z, A), O eixo A pode duplicar X, Y, Z ou fazer um quarto eixo completo com firmware personalizado usando os pinos D12 e D13);
- 2 conectores fim de curso para cada eixo (cada par de eixo compartilhada por mesmo pino IO);
- Permite ligação e direção de um motor *Spindle* no caso de uso como CNC.
- Possui conector para *cooler*.

- Possui *jumpers* para definição do passo dos motores (*microstepping*).

Figura 33 – Pinagem CNC Shield V3 (MARTINS, 2022)



3.1.6 NEO-6M

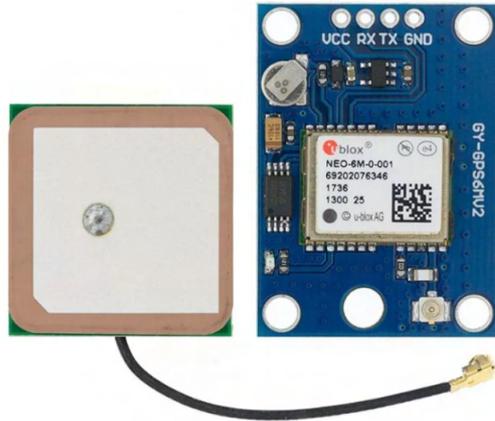
No coração do módulo GPS está um chip da U-blox – NEO-6M. O chip mede menos que um selo postal, mas contém uma quantidade surpreendente de recursos em sua pequena estrutura.

Ele possui a capacidade de rastrear até 22 satélites em 50 canais e alcançar o mais alto nível de sensibilidade de rastreamento do setor, ou seja, -161 dB, enquanto consome apenas 45 mA de corrente, podendo realizar 5 atualizações de localização em um segundo com precisão de posição horizontal de apenas 2,5 m. O mecanismo de posicionamento U-blox 6 também possui um tempo até a primeira correção (TTFF) de menos de 1 segundo.

Um dos melhores recursos oferecidos pelo chip é o Power Save Mode (PSM). Isso permite uma redução no consumo de energia do sistema ligando e desligando seletivamente certas partes do receptor. Isso reduz drasticamente o consumo de energia do módulo para apenas 11mA, tornando-o adequado para aplicações sensíveis à energia, como relógios de pulso com GPS.(U-BLOX, 2004)

O módulo suporta taxas de transmissão de 4800bps a 230400bps com um baud rate padrão de 9600.

Figura 34 – Módulo GPS Neo-6M (PROJECTS, 2021)



Seguem abaixo um conjunto das principais características desse sensor:

- Tipo de Receptor: 50 canais, GPS L1(1575.42 MHz)
- Acurácia de posição horizontal: 2,5 m
- Taxa: 1 Hz (5 Hz máximo)
- Tempo de Inicialização: Partida Fria: 27s, Partida quente: 1s
- Sensibilidade de navegação: -161 dBm
- Protocolo de comunicação: NMEA, UBX Binary, RTCM
- Baud Rate (Serial): 4800 ~ 30400 (padrão 9600)
- Temperatura de operação: $-40\text{ }^{\circ}\text{C} \sim 85\text{ }^{\circ}\text{C}$
- Tensão de operação: $2,7\text{ V} \sim 3,6\text{ V}$
- Corrente de operação: 45 mA
- Impedância TXD/RXD: $510\ \Omega$

O módulo GPS desempenha um papel fundamental neste projeto, sendo responsável por adquirir dados precisos de latitude e longitude para a calibração do sistema de controle do telescópio. A movimentação do eixo de altitude do telescópio é influenciada pela posição do observador na Terra, o que torna esses dados de localização essenciais para o rastreamento preciso dos astros. Embora atualmente seja possível obter os valores

de longitude e latitude por meio de pesquisas simples na internet, a incorporação desse módulo ao projeto elimina a necessidade de acesso à internet ou de inserção manual desses dados no sistema. Com essa implementação, a obtenção de dados geográficos ocorre de forma automatizada, garantindo um acompanhamento mais acurado dos astros durante as observações.

3.1.7 RMS20-256

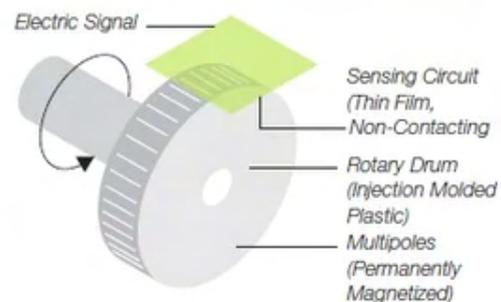
É um encoder magnetoresistivo fabricado pela NIDEK Copal. Um encoder magnetoresistivo é um dispositivo que utiliza o efeito magnetoresistivo para medir o movimento angular ou linear. O efeito magnetoresistivo é uma propriedade de certos materiais que faz com que sua resistência elétrica varie em resposta a um campo magnético aplicado.

No caso dos encoders magnetoresistivos, eles geralmente consistem em uma faixa ou disco codificado com padrões magnéticos e um sensor magnetoresistivo que detecta esses padrões. Quando o sensor se move em relação aos padrões magnéticos, sua resistência elétrica varia segundo as mudanças no campo magnético. Essa variação de resistência é então convertida em sinais elétricos que podem ser processados para determinar a posição angular ou linear do dispositivo. (DYNAPAR, 2023)

Figura 35 – Encoder Magnetoresistivo e seu funcionamento (INSTRUMENTS, 2022)



(a) Encoder Incremental (NIDEC, 2023)



(b) Funcionamento Encoder Magnético (DYNAPAR, 2023)

Seguem abaixo as principais características desse encoder.

- Resolução: 256 pulsos por revolução
- Frequência Máxima de Resposta: 5 kHz
- Forma de Onda de saída: Quadrada
- Fases de saída: A, B
- Defasagem entre os canais de saída: $90^\circ \pm 45^\circ$
- Tensão de alimentação: + 5V

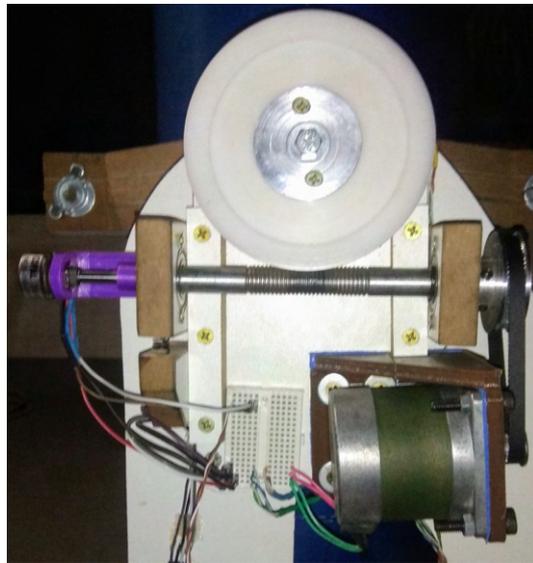
- Tensão de saída: $> 4,5$ V (HIGH), $< 0,5$ V (LOW)
- Corrente de operação: 15 mA

A base do telescópio está equipada com dois encoders: um conectado à rosca sem fim do eixo de azimute e outro ao eixo de altitude. Sua principal função é detectar possíveis perdas de passos nos motores. Quando um dos encoders identifica uma perda de passo, uma função é acionada para repetir esse passo perdido até que o encoder garanta que tudo esteja funcionando conforme o esperado. Essa abordagem garante um apontamento mais preciso do telescópio para os astros, minimizando os erros e assegurando um acompanhamento confiável.

Figura 36 – Encoder e conjunto do eixo de altitude (Fonte: De autoria própria)



(a) Encoder acoplado ao suporte

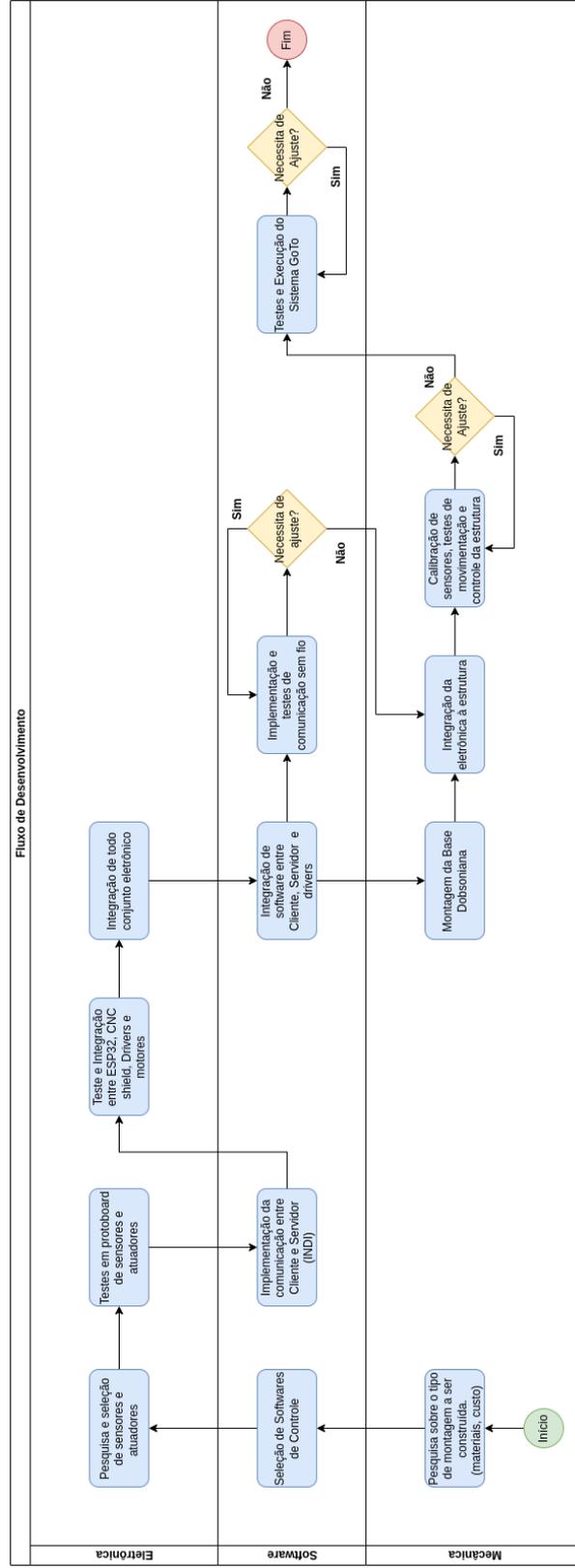


(b) Conjunto eixo altitude

3.2 Fluxo de Desenvolvimento

Neste trabalho foram desenvolvidas as etapas iniciais para a construção de um sistema GoTo em um telescópio Dobsonianiano, com testes individuais de todos os seus motores e sensores. A primeira etapa realizada foi o detalhamento dos blocos do sistema, seguida de um estudo teórico a respeito de cada um dos componentes necessários para estruturá-los. Na segunda etapa foram feitos os testes de integração, construção da montagem, integração eletrônica, calibração dos sensores e por fim testes e execução do sistema GoTo, como exibido na Figura 37.

Figura 37 – Fluxograma de Desenvolvimento (Fonte: De autoria própria)

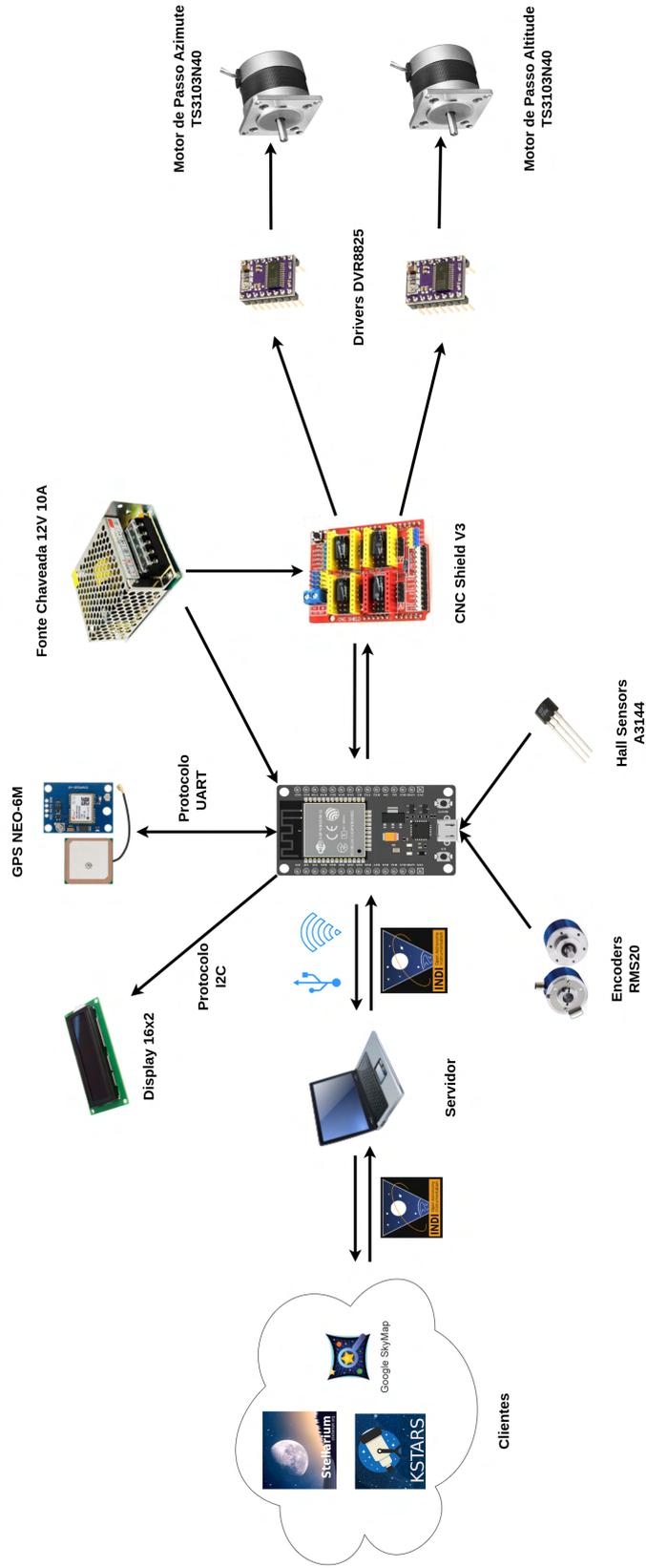


3.2.1 Relação entre componentes do sistema

Segue abaixo uma representação simplificada de todos os componentes do sistema e suas relações.

A comunicação entre um software planetário que contém todos os dados sobre a posição dos astros na esfera celeste será intermediado com o INDI que de maneira sem fio, por meio de WiFi, comunicará com a DOIT ESP32-WROOM. Sendo assim, o dispositivo manterá contato com todos os outros drivers do sistema, recebendo e enviando informações sobre a localização e posicionamento da estrutura. O sistema embarcado da estrutura do telescópio será alimentado utilizando uma fonte chaveada de 12V e 10A.

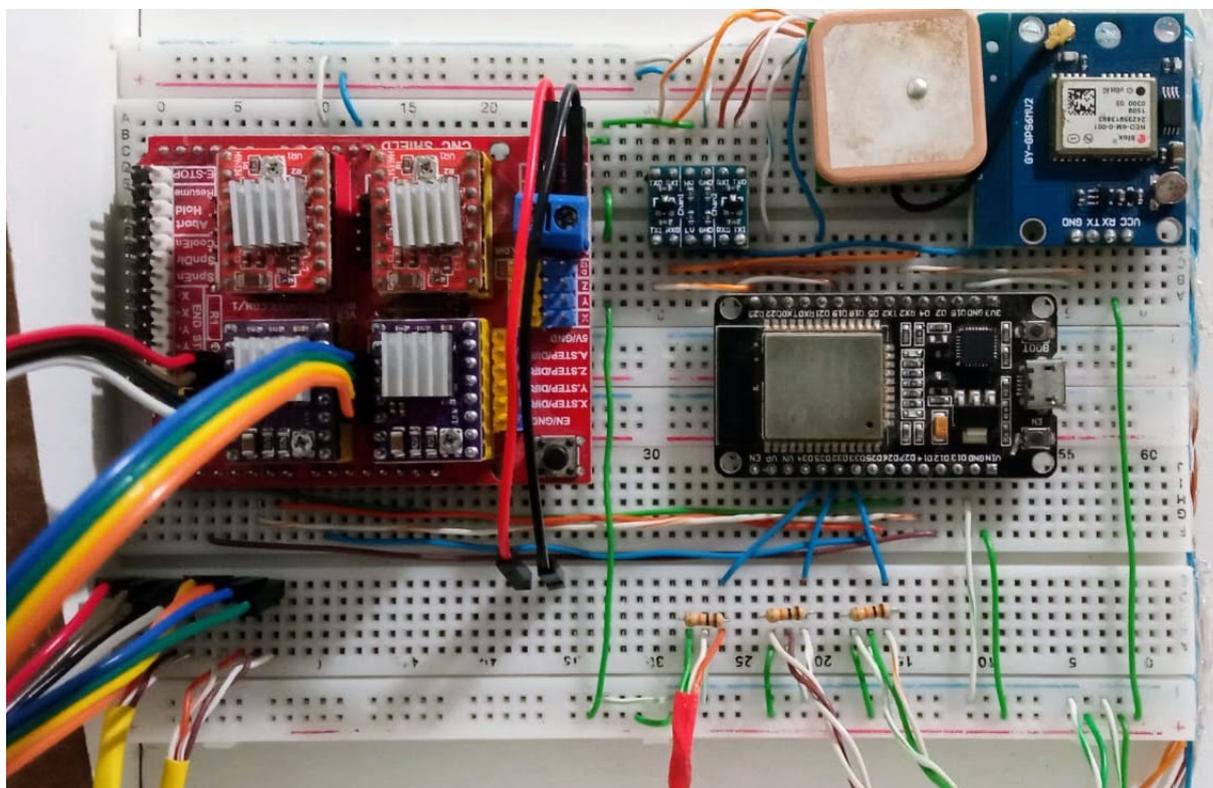
Figura 38 – Relação entre componentes (Fonte: De autoria própria)



3.2.2 Leitura dos sensores

Após a instalação dos sensores na base estrutural, conforme pode ser visto na Figura 39, é necessário interpretar os dados a partir dos sinais enviados para a ESP32. Cada sensor comporta o seu tipo de protocolo de comunicação, o módulo NEO-6M, utiliza o protocolo UART, e a tela LCD, o protocolo I2C. O sistema também possui sensores de efeito *hall* que permitem a calibração do telescópio e os encoders que verificam a perda de passo do mesmo. A partir da utilização e modificação das bibliotecas disponibilizadas pelo framework da ESP-IDF e também de repositórios públicos, é possível tratar as informações presentes na ESP-32 e assim mandar as informações através do protocolo INDI para um cliente escolhido.

Figura 39 – Eletrônica acoplada à *protoboard* (Fonte: De autoria própria)



- Módulo GPS

O módulo NEO-6M é essencial para que um cliente possa se referenciar na posição atual da esfera celeste. Ao conhecer a localização do polo sul celeste, especialmente no caso da região centro-oeste do Brasil, é possível determinar a demarcação do zênite e do horizonte, o que orienta o posicionamento do telescópio. Isso permite o rastreamento, apontamento e acompanhamento dos astros de acordo com a função desejada.

Pela Figura 34, é utilizada a saída TX conectada a ESP32 pela GPIO 16(RX2), o diagrama completo de ligação pode ser visto na Figura 55, que irá transferir os dados da

posição necessária. O protocolo NMEA nos entrega várias informações com diferentes tipos de sentença, porém a que importa é a Global Positioning System Fix Data (GPGGA). Essas sentenças chegam com seus dados espaçados a cada vírgula, segue um exemplo de como que é recebida a informação na Tabela 2.

Tabela 2 – Sentença GPGGA do protocolo NMEA

Nome	Exemplo de dados	Descrição
Sentence Identifier	\$GPGGA	Global Positioning System Fix Data
Time	170834	17:08:34 Z
Latitude	4124.8963, N	41d 24.8963' N or 41d 24' 54"N
Longitude	08151.6838, W	81d 51.6838' W or 81d 51' 41"W
Fix Quality:		
- 0 = Invalid		
- 1 = GPS fix		
- 2 = DGPS fix	1	Data is from a GPS fix
Number of Satellites	05	5 Satellites are in view
Horizontal Dilution of Precision (HDOP)	1.5	Relative accuracy of horizontal position
Altitude	280.2, M	280.2 meters above mean sea level
Height of geoid above WGS84 ellipsoid	-34.0, M	-34.0 meters
Time since last DGPS update	blank	No last update
DGPS reference station id	blank	No station id
Checksum	*75	Used by program to check for transmission errors

Usando uma expressão regular, conseguimos capturar essa linha de informação do GPGGA, e também pegar o grupo que importa, que é a latitude, longitude e altitude. Com isso irá informar para o driver instanciado com o INDI, em qual posição o telescópio está na esfera celeste, e assim tem-se uma precisão maior ao direcionar a trajetória do telescópio pelo cliente.

- Sensor Hall

Uma das rotinas que o telescópio necessita para a calibragem é o chamado modo estacionamento (*Park*), sendo que esse modo faz o telescópio direcionar o alvo para o polo

sul geográfico. A rotina precisa chegar em dois ímãs colocados no disco inferior, na direção que é a frente do telescópio apontado para o polo sul geográfico, como segue a Figura 40. Também para satisfazer a rotina, é necessário chegar em um ímã posicionado no eixo de altitude do telescópio, como na Figura 41.

Figura 40 – Ímãs na base inferior (Fonte: De autoria própria.)



Figura 41 – Sensor Hall A e Sensor Hall B (Fonte: De autoria própria.)



Dois sensores *hall* são posicionados no disco de superior, chamaremos o primeiro de sensor A e o segundo de sensor B, também um sensor *hall* posicionado na haste que segura o eixo de atitude do telescópio, que chamaremos de sensor C. Todos os sensores têm o critério de parada quando é percebido um campo magnético.

Conforme a rotina do telescópio, quando é acionado o modo estacionamento, o motor de passo gira o telescópio no sentido horário até encontrar um ímã no sensor A. Quando encontra um ímã no sensor A ele verifica se também existe um ímã no sensor B, caso isso aconteça o telescópio está posicionado corretamente no polo sul geográfico. Caso não, a rotina exige que o telescópio mude a direção do giro para o sentido anti-horário até satisfazer a condição de ter um ímã nos dois sensores.

Também existe uma condição de segurança, caso o telescópio não encontre nenhum dos ímãs durante um giro de 270° , tanto para o horário quanto para o anti-horário, a base para de girar.

Já no caso do terceiro sensor, a rotina de estacionamento faz com que o motor de passo de altitude gire para o sentido anti-horário, fazendo que o tubo do telescópio desça em relação ao zênite. Com isso, é necessário identificar quando o tubo está apontado para o horizonte, e assim entra o terceiro sensor hall para pegar esse dado com o ímã fixado na haste que segura o eixo de atitude.

Figura 42 – Telescópio modo estacionamento (Fonte: De autoria própria.)



3.2.3 Rotinas do Cosmo Finder

Nesse tópico, será discutido como é implementado as rotinas da ESP32 que é referenciado aqui como Cosmo Finder, para o uso do telescópio. O código-fonte dessa área de desenvolvimento está disponibilizado no repositório <<https://github.com/brunocmo/cosmo-finder>>.

A aplicação foi desenvolvida com o framework ESP-IDF, sendo utilizado um kernel de sistema operacional de tempo real chamado FreeRTOS. Com o FreeRTOS e a ESP-IDF, podemos utilizar alguns recursos que auxiliam na construção do sistema embarcado, como

funções prontas de temporização, chamadas de funções para leitura e escrita, protocolos de comunicação, entre outras inúmeras rotinas prontas e de código aberto. É utilizado a linguagem C++, que permite ter o uso do paradigma de orientação a objetos, o que ajuda na construção da arquitetura do código.

A seguir, foi dividido em 3 tópicos principais, sobre as rotinas da ESP32.

- Inicialização do WiFi

Primeiramente, é necessário configurar a rede WiFi que o sistema embarcado irá se conectar, pelo configuração de menu(*menuconfig*), é definindo então nome da rede e a senha para autenticação. Após essa configuração, é possível construir uma imagem para ser colocada na ESP32.

Uma das primeiras coisas que o microcontrolador faz ao inicializar, além de ter o seu boot iniciado com sucesso, é conectar com a rede WiFi definida anteriormente. A aplicação só irá iniciar os recursos periféricos como motor de passo, encoder, GPS e sensores halls, quando obter o êxito na autenticação da rede WiFi estabelecida. Enquanto não for estabelecido o acesso à rede, a rotina fica tentando se conectar sem parar a rede especificada. Conforme o código abaixo, arquivo **main.cpp** na linha 103), é visto uma dessas tarefas sendo criadas e a rotina que verifica se está conectado ou não com a rede.

```
setupWifi();
xTaskCreate( &runWifi, "Run Wifi", 4096, NULL, tskIDLE_PRIORITY, NULL );

while ( wifiState != WIFI::Wifi::state_e::CONNECTED )
{
    vTaskDelay( 1000 / portTICK_PERIOD_MS );
}
```

- Inicialização de todas as tarefas

Após o êxito da conexão com a rede, é exibido na tela de LCD, o endereço de IP necessário para se comunicar com a solução computacional. Até que seja conectado um cliente, a rotina fica pausada e esperando alguém se conectar, caso o contrário começa a instanciação de cada classe responsável pelo funcionamento de cada periférico. Dos periféricos, é inicializado as tarefas de funcionamento do Motor de Passo, Encoder e GPS, código encontrado no arquivo **Controller.cpp** na linha 23.

```
void Controller::Run()
{
```

```

xTaskCreate( &StepMotor::Run, "Run Motor", 4096, &m_motorPasso,
            tskIDLE_PRIORITY, &m_motorHandle );
xTaskCreate( &Comms::RunTCPSTerver, "Run TCP/IP Comms", 4096,
            &m_communication, tskIDLE_PRIORITY, &m_commsHandle );
xTaskCreate( &GPS::Gps6mv2::receiveData, "Run GPS", 32000, &m_gps,
            tskIDLE_PRIORITY, &m_gpsHandle);
}

```

Já na parte de controle, é inicializado uma tarefa para o tratamento dos protocolos de comunicação com o cliente e também uma tarefa que faz a referência de todas as classes para exercer o controle das mesmas. Essa tarefa, faz com que cada comando vindo do cliente seja distribuído pelo restante dos periféricos, tendo ênfase nos recursos mais utilizados que são os motores de passo.

- Espera de uma requisição pelo cliente

A partir da biblioteca do INDI, temos comandos padrões utilizados para o funcionamento dos clientes, são eles GoTo, Estacionamento(*Park*), Rastreamento(*Track*) e Parar(*Stop*), em conjunto desses comandos, tivemos a necessidade de criar um comando para a aquisição do posicionamento do GPS. Foi desenvolvido então, um protocolo que comportassem esses comandos, o que gerou um dicionário no código, sendo ele:

- 0x01 - Comando para GoTo
- 0x02 - Comando para Estacionamento
- 0x03 - Comando para Rastreamento
- 0x04 - Comando para pedir valores de GPS
- 0x05 - Comando para Parar

Como dito anteriormente, a tarefa de controle tem uma instância de cada classe que controla os periféricos e mantém os protocolos de comunicação. Sendo assim, cada requisição mandada pelo cliente é tratada por essa tarefa, e foi projetado um protocolo serial mostrado na Tabela 3, adaptada ao mapa de comandos que estipulamos. Com a tabela temos que o “passoX” e “passoY” são inteiros e “long”(longitude), “lati”(latitude) e “alti”(altitude), são pontos flutuantes.

Na solução computacional o mesmo protocolo é tratado para cada requisição que chega, será posteriormente instruído como o driver do INDI envia essa mensagem, mas por enquanto sabemos que o tempo de espera para cada mensagem é de 1 segundo. O

Tabela 3 – Protocolo Cosmo Finder

Comando	Dado	Verificação	Tamanho	Retorno
0x01	passoX, passoY	crc16	12 bytes	OK(2 bytes)
0x02		crc16	3 bytes	OK(2 bytes)
0x03	passoX, passoY	crc16	12 bytes	OK(2 bytes)
0x04		crc16	3 bytes	long, lati, alti(26 bytes)
0x05		crc16	3 bytes	OK(2 bytes)

que leva o programa na ESP32 estar em um laço de 1 segundo à 1 segundo. O código abaixo, exibe um trecho da camada de como é tratado os eventos de serialidade. Pode ser encontrado no arquivo **IndiDriverProtocol.cpp** na linha 19.

```

void IndiDriverProtocol::translateMessage( char* message )
{
    std::regex wordRegex;
    std::smatch matchRegex;
    m_commandId = static_cast< std::uint8_t >( message[0] );

    switch ( m_commandId )
    {
    case GoTo:
    case Track:
        m_azimuthSteps = getIntValues( &message[1] );
        m_altitudeSteps = getIntValues( &message[6] );
        break;
    case Park:
        m_needToPark = true;
        break;
    case GetLocation:
        break;
    case Stop:
        m_needToStop = true;
        break;
    default:
        break;
    }
}

```

Com esse código, basta construir a imagem para a ESP32, e fazer o carregamento da mesma no microcontrolador. Caso não tenha todas as conectividades, as rotinas do programa podem acabar não funcionando corretamente.

3.2.4 Servidor com o driver instanciado

Segundo o manual do *driver* INDI(INDI, 2023), é ensinado primeiramente a como criar telas, botões e exibições de valores, porém para o nosso caso, iremos aproveitar as bibliotecas de telas prontas para um *driver* de telescópio genérico. Apesar da biblioteca oferecer uma gama de customização de interface, ainda é necessário o desenvolvedor criar rotinas para a configuração de passos, aquisição de valores e diversas transformações de coordenadas.

Chamaremos essa instanciação do *driver* com o servidor de **indi-esp**. O código-fonte dessa aplicação pode ser obtida pelo repositório <<https://github.com/brunocmo/indi-esp>>.

Primeiramente, é indispensável a configuração da localização que o *driver* do telescópio reside, o que leva a ter a exatidão de cada cálculo feito na esfera celeste. Com a requisição do comando de solicitação de dados do GPS, tem-se a obtenção desses valores e também para a configuração no *driver*.

É aproveitado uma rotina pronta da biblioteca que consegue conectar com o endereço de IP e a porta nas janelas do cliente, com isso após a conectividade com o ESP32, já é possível utilizar outros comandos para o envio de mensagens. Outra rotina aproveitada, foi a da leitura do alvo do telescópio, que é atualmente simulada pelo próprio *driver*. Com essa rotina, tem-se a realizar os cálculos necessários para o envio de comandos de deslocamento para a ESP32.

Para o envio de mensagens, é implementado uma rotina que consegue envia os “bytes” de uma forma ordenada que segue os princípios do protocolo de comunicação criado, aqueles que estão mostrados na Tabela 3. Sendo assim, todos os comandos usados no indi-esp serão redirecionados para essa rotina que envia e espera uma resposta do Cosmo Finder.

A primeira calibração é feita pelo comando de estacionamento, que procura na esfera celeste a posição de 180° de azimute e 0° de altitude, sendo a mesma a localização do polo sul geográfico com o alvo no horizonte. Após colocar o alvo do telescópio nessa posição, é enviado a mensagem de estacionamento para o Cosmo Finder, que com a ajuda dos sensores halls, irá posicionar o telescópio na posição de estacionado.

Após estacionado, é possível utilizar a função GoTo, que irá fazer uma transformação da ascensão reta e da declinação do alvo, para os valores de azimute e altitude. Com isso, tem-se a diferença da posição anterior do alvo com a posição desejada, e assim é adquirido o valor resultante em graus dos movimentos de ambos os eixos do telescópio.

$$\Delta x = x_f - x_i \quad (3.7)$$

Sendo Δx o valor resultante em graus, x_f a posição desejada e x_i a posição inicial. Contudo, o protocolo só envia a quantidade de passos que os motores precisam trabalhar, o que leva a fazer a divisão da resolução do passo em graus, conforme calculado em 3.2, o que gera o resultado do número de passo necessário para o telescópio se movimentar, representado como z , visto na equação abaixo:

$$z = \Delta x / Y \quad (3.8)$$

Com isso, tem-se o valor do número de passos necessários que o motor de passo precisa executar. Esse número pode ser positivo ou negativo, o que é exigido para as regras submetidas para o movimento do motor de passo. Caso o motor de azimute seja positivo irá se deslocar para o sentido horário, o contrário leva o motor a se deslocar para o sentido anti-horário. Já para o motor de altitude, a lógica é invertida, caso positivo, é deslocado anti-horário e o oposto é deslocado no sentido horário. Respeitando, a movimentação do telescópio com o referencial obtido pelo cliente.

O algoritmo abaixo refere-se a rotina do cálculo de passos necessários do azimute e altitude, que serão posteriormente enviados ao Cosmo Finder. O código desse trecho está localizado no arquivo **indi_esp32.cpp** na linha 337.

```
std::tuple< int, int > Esp32Driver::stepsNeededToMove( double ra, double dec )
{
    double diffAzimute{0};
    double diffAltitude{0};

    double anglePerStep = 0.00329670329;

    INDI::IHorizontalCoordinates currentAltAz{ 0, 0 };
    INDI::IHorizontalCoordinates positionAltAz{ 0, 0 };

    TelescopeDirectionVector currentTDV;
    TelescopeDirectionVector positionTDV;
    TransformCelestialToTelescope( currentRA, currentDEC, 0.0, currentTDV);
    TransformCelestialToTelescope( ra, dec, 0.0, positionTDV);

    AltitudeAzimuthFromTelescopeDirectionVector(currentTDV, currentAltAz);
    AltitudeAzimuthFromTelescopeDirectionVector(positionTDV, positionAltAz);

    trackingAltAz = positionAltAz;

    diffAzimute = positionAltAz.azimuth - currentAltAz.azimuth;
    diffAltitude = positionAltAz.altitude - currentAltAz.altitude;
```

```

diffAzimute /= anglePerStep;
diffAltitude /= anglePerStep;

int stepsAzimute = (int)diffAzimute;
azimuthAcc = diffAzimute - stepsAzimute;
int stepsAltitude = (int)diffAltitude ;
altitudeAcc = diffAltitude - stepsAltitude;

return { stepsAzimute, stepsAltitude };
}

```

Após feito o comando GoTo, o driver executa automaticamente o comando de rastreamento do astro selecionado, mandando a cada segundo a mesma diferença de coordenadas de azimute e altitude, e transformando-lá para passos a ser enviado ao cosmo finder.

Além desses comandos, também persiste a rotina de quatro botões de movimento do telescópio, norte, sul, leste e oeste. Que ao pressioná-los, é enviado a ESP32 um movimento de 10 passos para a direção do botão desejado. Essa rotina foi feita para que caso o GoTo perca o alvo do astro, seja possível corrigi-los sem a necessidade de fazer o estacionamento novamente.

E por fim, temos o comando de “Abort”, que para todas as rotinas de movimento do telescópio, o que inclui GoTo e Rastreamento. Parando o movimento do mesmo, a escolha do usuário.

Com isso, é possível compilar o programa e instanciar o servidor em uma máquina que tenha as bibliotecas do indi necessárias, pelo comando:

```
indiserver -v indi_esp32
```

3.2.5 Comunicação com o cliente KStars

Para enviar as informações de posição, e também receber o alinhamento do telescópio, é necessário um software cliente que faça esses tipos de operações. A escolha foi o KStars para ser essa ferramenta, pelo fato de ela ser uma solução computacional de software livre, sendo sua documentação e código-fonte, de livre acesso aos interessados na sua modificação e uso.

O KStars é um software cross-platform, de graça e aberto, que prove uma simulação de interface gráfica de precisão para o céu noturno, sendo que sua utilização pode ser qualquer posição da terra em qualquer data e tempo. Tendo um vasto componente de objetos astronômicos, que incluem de asteroides a supernovas. Usados por muitos profes-

sores, estudantes, astrônomos amadores, que provem ferramentas que possam mensurar métricas astronômicas, calendários estelares, editores do campo de visão e muito mais. (COMMUNITY, 2022)

Com o driver inicializado e as tarefas da ESP32 sendo executadas, é possível acessar o telescópio pelo IP informado no LCD da ESP32, o que pode ser verificado na Figura 43.

Após receber essa informação, na aba de dispositivos o usuário necessita colocar as informações de conectividade do servidor instanciado pelo driver. No caso, se estiver executando o servidor na mesma máquina que o cliente, é necessário somente usar o endereço como “127.0.0.1” e a porta é exibida no momento que é instanciado o servidor.

Figura 43 – Exibição do endereço de IP do cosmo finder (Fonte: De autoria própria.)

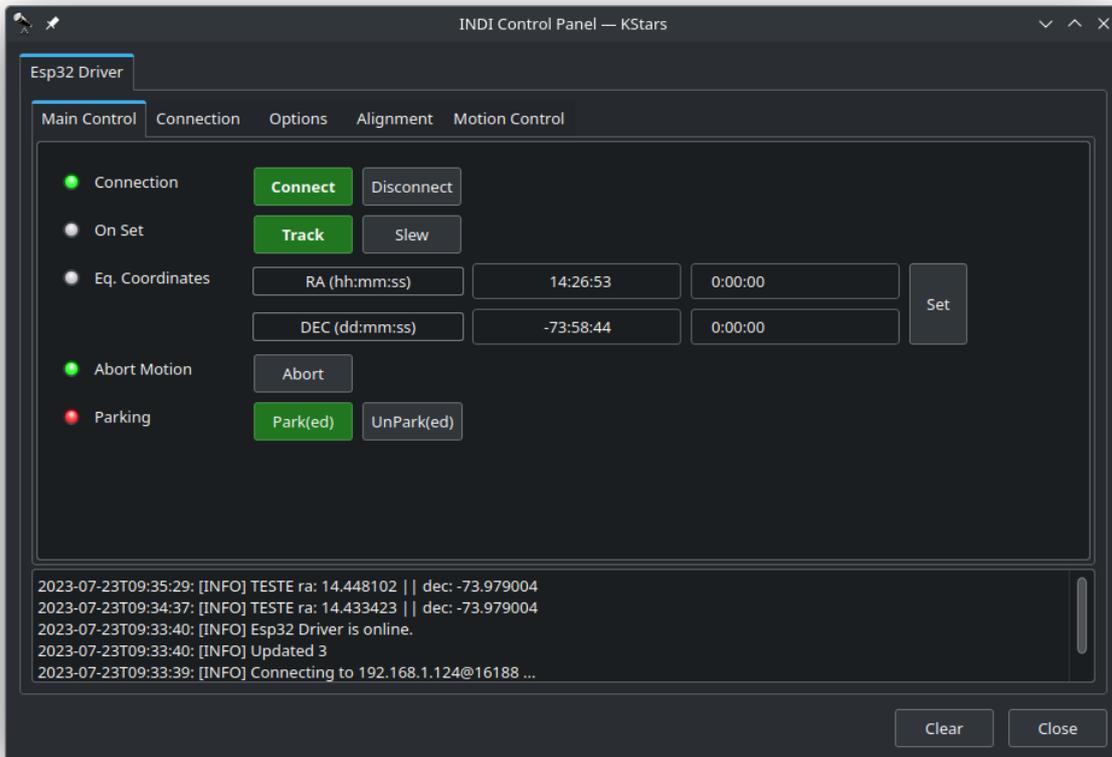


Após completar esse processo, pode-se acessar o menu de controle do telescópio. Antes de exibir esse menu, será necessário fornecer as informações de conectividade de rede da ESP32, que corresponde ao IP mencionado na Figura 43. Uma vez conectado, novos botões surgirão, como mostrado na Figura 44. Além disso, será adicionado um alvo referenciado pelo indi-esp, como evidenciado na Figura 45.

Antes de enviar qualquer comando para utilizar o telescópio, é crucial atualizar corretamente a localização atual do cliente, fornecendo informações precisas de latitude, longitude e elevação. Caso contrário, a movimentação do GoTo poderá apresentar erros. O GPS integrado à ESP32 é responsável por transmitir essas informações essenciais ao indi-esp, embora haja potencial para melhorias futuras com uma atualização de ambos os métodos. A Figura 46 exibe a janela de configurações da localização do cliente para referência.

Caso seja necessário corrigir algum passo errado, ou problemas que retirem o astro do alvo, é possível recorrer aos botões que quando acionados, enviam para o cosmo finder

Figura 44 – Tela inicial do telescópio (Fonte: De autoria própria.)



executar 100 passos na direção desejada. Esses botões são utilizados no campo Motion N/S que representa o Norte e o Sul e o Motion W/E que representa o Leste e o Oeste, conforme exibido na Figura 47.

Com essas opções disponíveis, é possível realizar testes para verificar a calibração, e a precisão do GoTo nos alvos requisitados.

Na Figura 48, temos o diagrama de blocos do Cosmo Finder. Para o Encoder, são recebidos os valores das entradas A e B, e os motores de passo com o módulo “StepMotor” enviando sinal de passo(step) e direção(dir). O módulo GPS NEO-6M, está dedicado à aquisição e procura da sentença *GPGGA*. O módulo Wi-Fi, faz a conexão com *SSID* especificado na placa e o módulo HD44780, pertença ao enviar os dados para exibir as mensagens na tela *LCD*.

Na Figura 49, temos o diagrama de fluxo do Cosmo Finder, onde pode ser observado de uma maneira visual as operações mencionadas nesse subtópico.

Figura 45 – Alvo da localização desejada (Fonte: De autoria própria.)

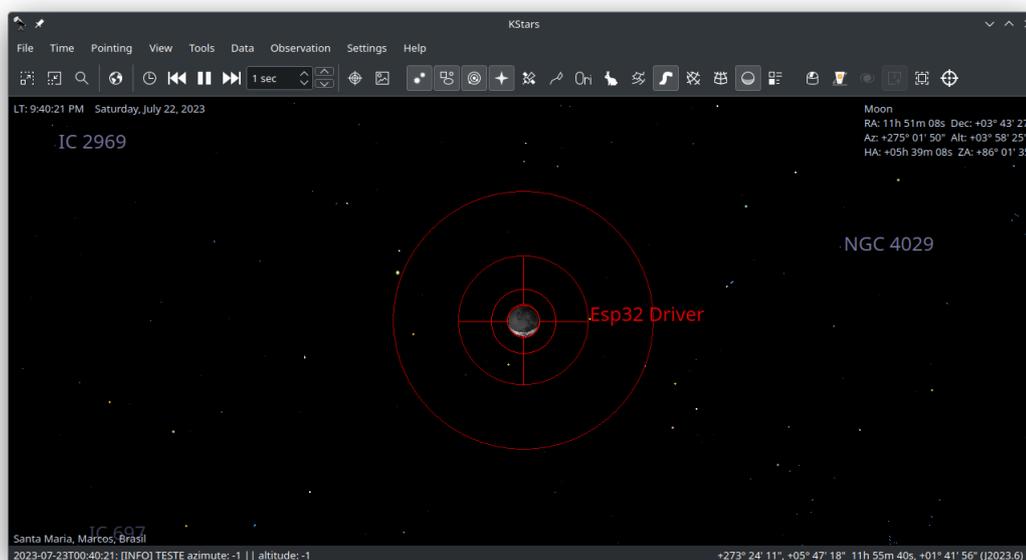
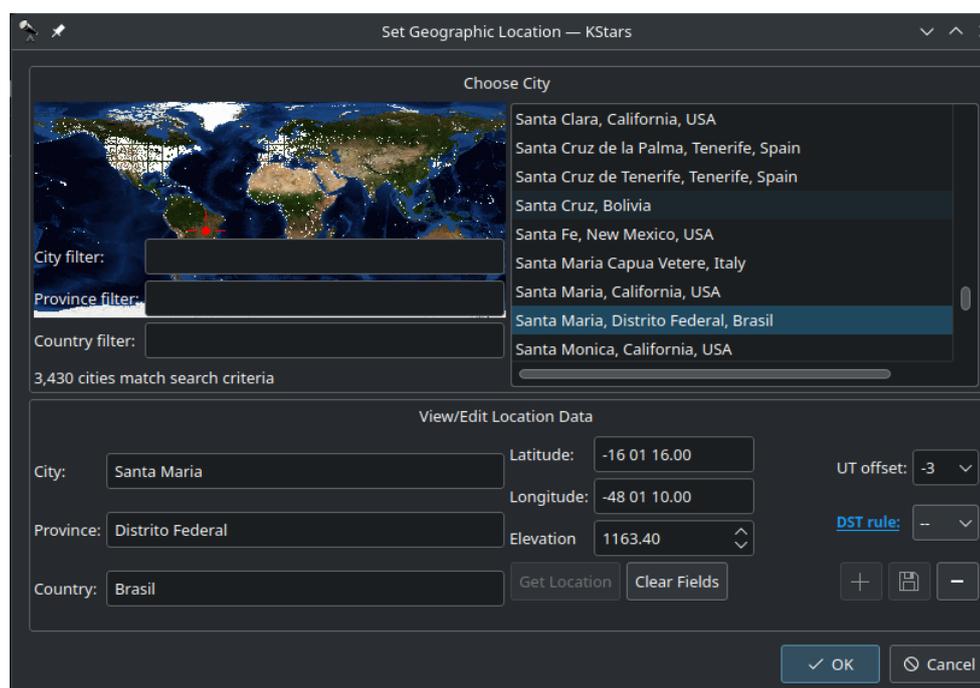


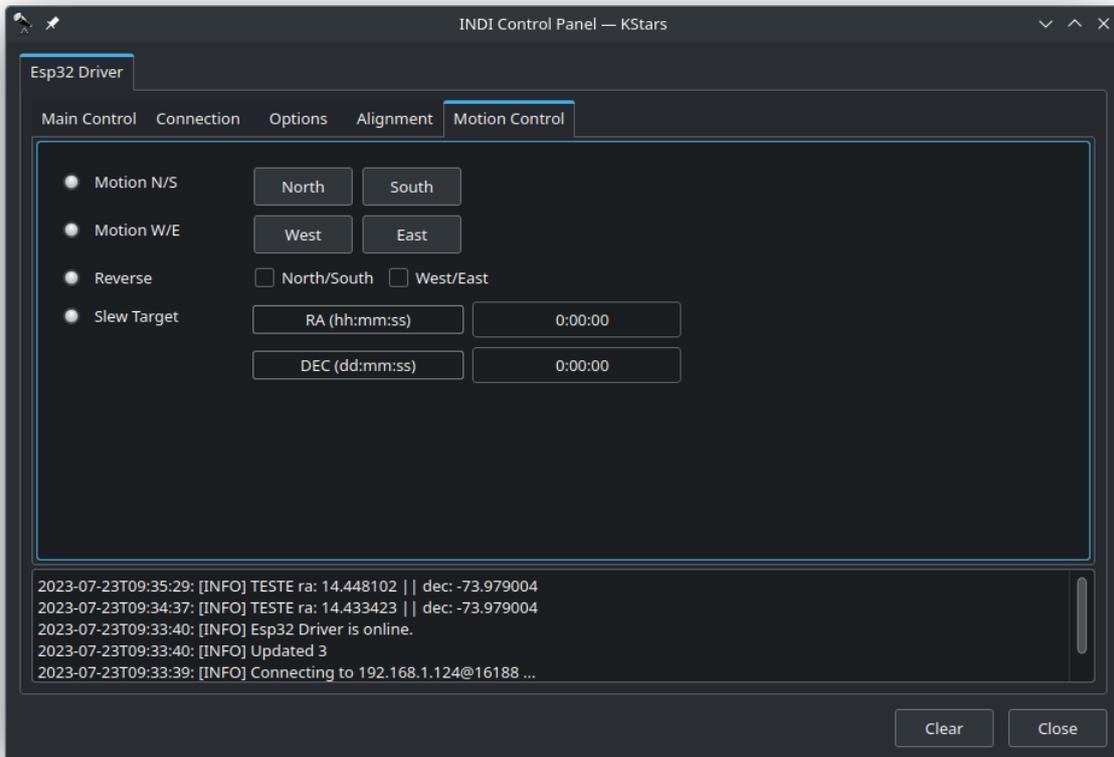
Figura 46 – Mapa de inserção da localização (Fonte: De autoria própria.)



3.2.6 Considerações finais do capítulo

Feito o estudo dos materiais necessários para a confecção do telescópio, sendo possível ter a construção de sua base. Com isso, têm-se os cálculos necessários para dimensionar a base que conecta o eixo azimutal e também os lados que fixam o eixo de altitude. Após feito, com a aquisição dos materiais eletrônicos é preciso fazer testes de integração com a estrutura dobsoniana.

Figura 47 – Botões de movimento (Fonte: De autoria própria.)



Foi possível também utilizar várias funcionalidades já empregadas pela ESP32, o que facilita o desenvolvimento do software para o sistema embarcado. Ao conectar motores de passos a *CNC Shield* e implementar as rotinas requisitadas do sistema, temos o movimento desejado pelo telescópio que garante parte do escopo do projeto. Com o GPS, é possível verificar a posição que se situa o cliente ao desejar observar o astro, o qual o módulo *NEO – 6M*, é conectado a portas de *RX/TX* da ESP32. E por fim, fixar os sensores *RMS20 – 256* a caixa de redução para verificar se os passos foram realmente executados pela rotina da ESP32.

Após a confecção da estrutura e equipamentos eletrônicos, é projetada a arquitetura do software embarcado *Cosmo Finder*, juntamente com o driver *indi-esp* para o funcionamento de todo o sistema. O *Cosmo Finder* se sujeita ao controle de passos, verificação dos sensores e comunicação para o cliente que auxilia o processo do telescópio *GoTo*. Já o *indi-esp*, faz a ponte de comunicação entre o cliente e o próprio sistema embarcado, garantindo os processos de movimentação, estacionamento e parada do telescópio.

Por fim, com essas especificações detalhadas e o desenvolvimento de todo o telescópio, é possível averiguar a mira do astro desejado e seu rastreamento, o que leva a

completitude de um sistema GoTo. No próximo capítulo, terá a realização da integração de todo o sistema em conjunto com os testes fixados em astros do sistema solar.

Figura 48 – Diagrama de blocos do Cosmo Finder (Fonte: De autoria própria.)

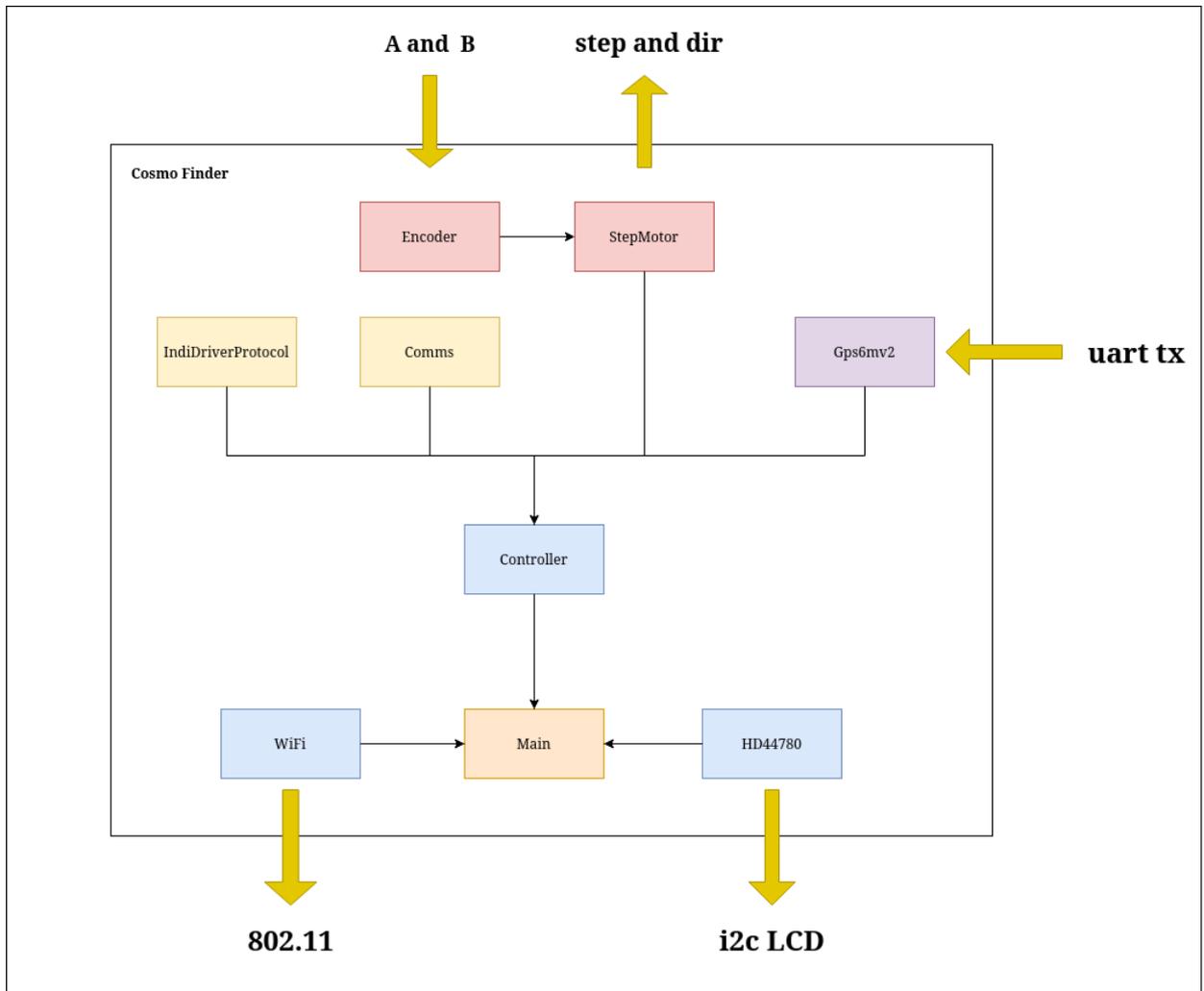
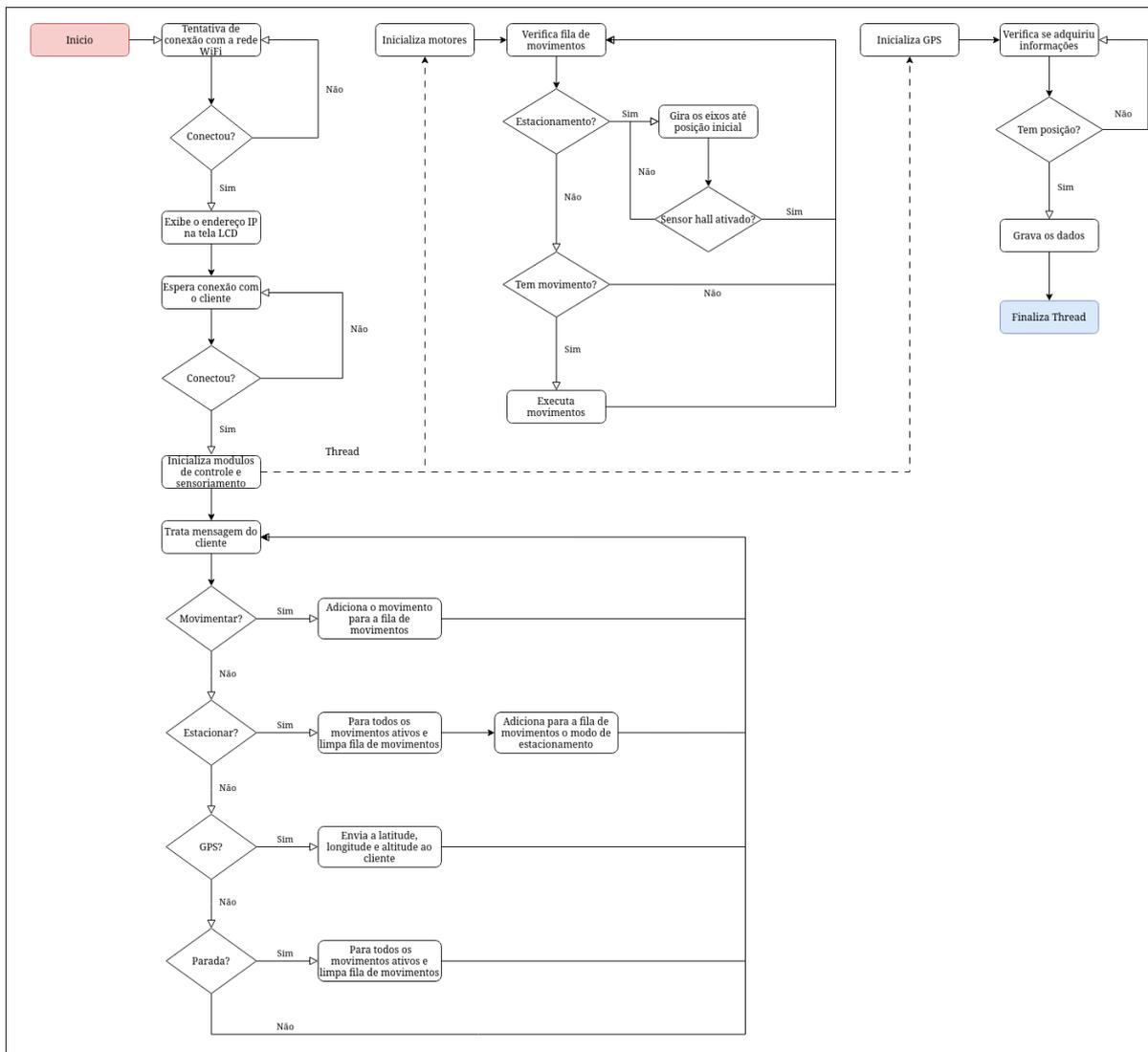


Figura 49 – Diagrama de Fluxo do Cosmo Finder (Fonte: De autoria própria.)



4 RESULTADOS

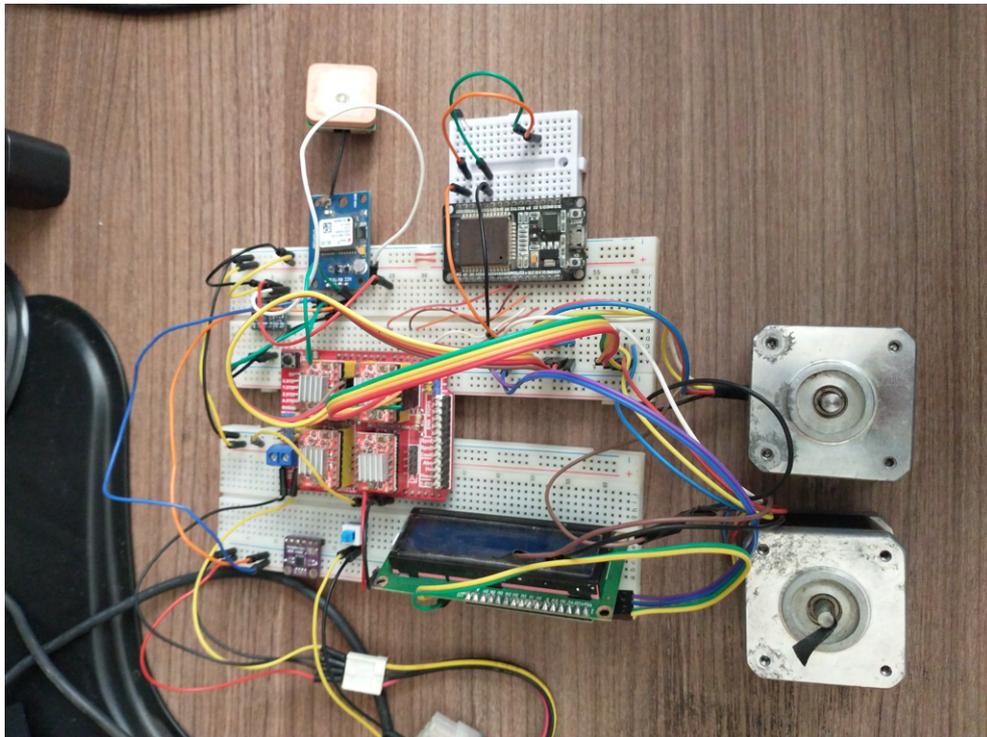
A seguir, apresentamos os resultados obtidos e, caso seja necessário reproduzir algum dos passos, os dois repositórios do projeto estão disponíveis:

- Cosmo Finder - <<https://github.com/brunocmo/cosmo-finder>>
- indi-esp - <<https://github.com/brunocmo/indi-esp>>

4.1 Simulação do telescópio

Foram conduzidas diversas simulações em bancada, permitindo a implementação remota do telescópio sem exigir a presença física do equipamento. Esse processo levou ao desenvolvimento mais ágil dos algoritmos para a realização de testes reais. Contudo, durante os testes do módulo GPS, foram identificados alguns desafios, possivelmente causados por problemas de contato dos componentes e falta de sinal, o que resultou em uma implementação incompleta do módulo. A bancada de simulações pode ser visualizada na Figura 50, que foi construída conforme o diagrama de conexão na Figura 55.

Figura 50 – Ambiente de simulação (Fonte: De autoria própria.)



Alguns componentes geraram dificuldades de implementação, e acabaram sendo descartados por falta de conhecimento e prazos necessários para lançar um protótipo que

funcionasse adequadamente. Como, por exemplo, um deles ser o MPU-9250, que poderia trazer um grande auxílio para a calibração do comando de estacionamento, porém com a falta de tempo, mudar a arquitetura, tanto física quanto em software, foi inviável para o momento.

Como resposta a falta de um meio de *feedback* sobre o estado do movimento do sistema, decidimos acoplar 2 encoders como método de correção de possíveis perdas de passos nos eixos. Principalmente pela dificuldade de unir as correias sem que ficassem muito apertadas e quebrassem ou frouxas e perdessem parte da tração fornecida pelo motor por deslizamento.

4.2 Testes com o Telescópio

Após realizar a calibragem do telescópio com o estacionamento como no vídeo a seguir <<https://youtu.be/4966ONk6xQM>>, é direcionado o olhar para os astros do sistema solar, claramente visíveis no céu noturno durante os dias do experimento. Tem-se a realização das observações de um astro grande como a Lua e também observar astros menores, como Saturno e Júpiter.

Figura 51 – Base completa e integrada (Fonte: De autoria própria.)



Nessa bancada de testes, verificamos o tempo de deslocamento do telescópio, a diferença da distância real para a distância desejada e o tempo de rastreamento até o objeto sair da focal.

Nos primeiros testes com Saturno com a ocular 32mm, conforme a Tabela 4, o telescópio se deslocou 89° à esquerda e 67° para cima, levando um tempo de 54 segundos para deslocar-se ao alvo desejado, sendo feito posteriormente 5 testes. No caso de Saturno, infelizmente não foram adquiridas fotos do astro, devido à vibração que deixava o astro borrado. Nesse primeiro teste, os encoders não estavam acoplados e o software devidamente implementado.

Tabela 4 – Teste de GoTo e rastreamento de Saturno

Tentativa	Distância do alvo(graus)	Tempo(minutos)
1	01°04'27"	07:02:66
2	02°66'98"	06:47:37
3	01°80'32"	07:05:39
4	01°43'11"	07:03:23
5	01°49'78"	06:58:92

Após a primeira observação de Saturno, foi verificado que o telescópio estava sujeito a uma considerável vibração, resultando em imagens borradas durante o processo de observação. Teve-se a suspeita que essa condição fosse ocasionada pela folga nos parafusos que conectam o motor à base. Apesar da pequena variação angular do astro, a localização precisa foi um desafio para realizar o rastreamento adequado. Além disso, notou-se que quanto mais distante o objeto de interesse estava do ponto de estacionamento inicial, maior a probabilidade de ocorrer perda de passos durante o movimento. Esses aspectos foram cuidadosamente analisados para aprimorar a precisão e estabilidade do sistema de rastreamento do telescópio.

Também foi constatado um tempo reduzido de visualização do astro na ocular durante o rastreamento, o que suscitou considerável incerteza sobre a causa desse comportamento. Isso conduziu a uma análise minuciosa do código, resultando na descoberta de que os motores estavam realizando menos passos do que o necessário devido a uma conversão inadequada de ponto flutuante para inteiro.

Para o segundo teste, o alvo selecionado foi Júpiter com ocular de 32mm, tendo o deslocamento de 134° a esquerda e 48° para cima, levando um tempo de 1 minuto e 23 segundos de deslocamento para o alvo desejado. Também foram realizados 5 testes para o alvo, com resultado na Tabela 5 onde se notou diferenças significativas nos testes.

Ao empregar o encoder nesta iteração, existia a expectativa de alcançar análises mais precisas para o GoTo. Contudo, ao realizar a primeira medição, constatou-se que, embora o rastreamento tenha apresentado melhorias, a distância até o alvo desejado foi consideravelmente maior do que o esperado. Surgiram suspeitas de que o encoder estava impactando negativamente o teste, devido à codificação inadequada que acabou comprometendo os movimentos do motor de passo.

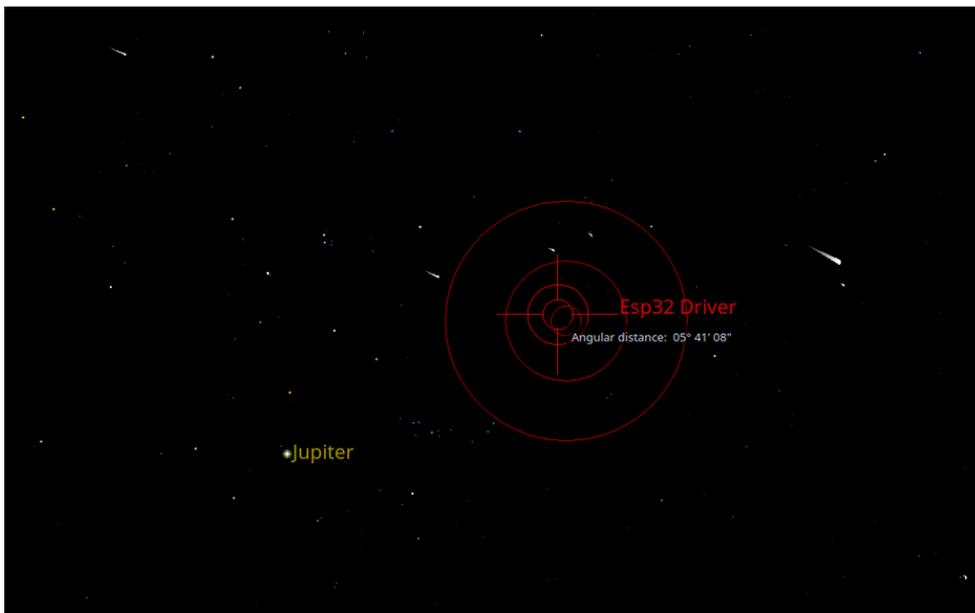
Tabela 5 – Teste de GoTo e rastreamento de Júpiter

Tentativa	Distância do alvo(graus)	Tempo(minutos)
1	12°06'32"	12:43:73
2	05°41'08"	20:01:26
3	05°58'87"	19:32:87
4	05°10'74"	19:12:01
5	05°79'29"	20:00:47

Verificou-se que a rotina que descartava pontos flutuantes resultou em uma notável redução do tempo de acompanhamento de um astro com o telescópio. O segundo teste registrou um tempo 2,5 vezes superior ao do primeiro teste. Essa observação levou à conclusão de que existia um problema na implementação do código anterior, devido ao problema de conversão de graus para passos do motor. Além disso, constatou-se que a distância até o alvo aumentou em 4° em comparação com o primeiro teste. Tal cenário sugere que, quanto mais próximo do ângulo de 180° de azimute em relação ao polo sul celeste (0°), mais erros são acumulados no caminho percorrido pelo telescópio. Conseqüentemente, a perda de passos se configura como a principal fragilidade da implementação do projeto.

A Figura 52 mostra como eram medidas as distâncias entre o astro e alvo pelo cliente KStars.

Figura 52 – Distância entre o alvo desejado e do alvo real (Fonte: De autoria própria.)



Os parafusos que seguram os motores de passos à base foram apertados, diminuindo a vibração e possibilitando tirar algumas imagens de Júpiter com suas luas (Io, Europa, Ganímedes e Calisto), a melhor foto obtida pode ser vista na Figura 53.

Figura 53 – Júpiter, Io, Europa, Ganímedes e Calisto (Fonte: De autoria própria.)



Outro teste foi feito com o alvo na Lua, como mostra a Figura 54, tendo um deslocamento 89° à direita e 14° para cima, levando um tempo de 52 segundos para o deslocamento ao alvo. Foram realizados 5 testes conforme mostra na Tabela 6.

Figura 54 – Imagem da Lua (Fonte: De autoria própria.)



A discrepância no tempo de acompanhamento da primeira tentativa pode ser explicada pelo uso de uma ocular de 10mm durante a observação. Essa ocular proporciona maior zoom, porém com a desvantagem de ter um campo de visão inferior, o que resultou em um tempo menor de acompanhamento.

Tabela 6 – Teste de GoTo e rastreamento da Lua

Tentativa	Distância do alvo(graus)	Tempo(minutos)
1	04°06'28"	04:30:43
2	04°03'73"	14:23:64
3	04°10'29"	13:37:41
4	03°57'74"	14:04:89
5	04°12'66"	13:53:77

Nos outros 4 testes, optamos por utilizar a ocular de 32mm, a mesma utilizada nos testes anteriores de Saturno e Júpiter. A ocular de 32mm é ideal para observar objetos amplos e extensos, proporcionando uma visão panorâmica do céu, enquanto a ocular de 10mm é mais adequada para observar detalhes e objetos mais compactos, com maior ampliação.

Esses testes nos mostram como é difícil acertar um astro na primeira tentativa sem travar um objeto já na esfera celeste. O que demanda a utilização de métodos como o do FireGoto (LORENSINI, 2023b) ou Onstep (ONSTEP, 2023) que conseguem calibrar o sistema de maneira personalizada quando este aponta para dois ou mais astros de modo a conseguir identificar o desvio angular entre o alvo e a posição atual e compensando isso nos próximos rastreamentos.

Como mencionado anteriormente, a perda de passo é acarretada principalmente pelas correias que não ficaram justas o suficiente para a transmissão correta do movimento entre as reduções, o que leva à perda de precisão na hora de rastrear a partir do modo estacionamento. Também foi averiguado perdas de passo por existir folgas entre certos componentes, e também por ter a dependência de bússolas externas para encontrar o polo sul geográfico. Isso diminui a precisão que o sistema necessita, contribuindo para o acúmulo de erros.

5 CONCLUSÃO

Neste trabalho, foi realizado o projeto e a construção de uma conversão completa de um telescópio. A montagem equatorial simples foi transformada em uma base altazimutal do padrão dobsoniano para permitir a integração de um sistema GoTo. Para definir o design do modelo, utilizamos o software CAD e construímos a estrutura do telescópio em MDF.

A fim de incorporar o sistema GoTo, foi fixado motores de passo nos dois eixos de movimentação, o que também exigiu a fabricação das caixas de redução para possibilitar o movimento. Posteriormente, procedeu-se com a calibração, utilizando sensores de efeito *hall* para criação de uma rotina de estacionamento, tornando o telescópio pronto para uso do GoTo e rastreamento.

A comunicação entre as duas soluções computacionais foi realizada por meio do framework ESP-IDF, juntamente com a biblioteca do INDI. Essa comunicação sem fio foi essencial para o funcionamento adequado do sistema GoTo. Além disso, teve o desenvolvimento da aplicação “Cosmo Finder”, incorporado na ESP32, que permite o controle dos sistemas físicos e a comunicação com o servidor.

Para a intermediação entre o “Cosmo Finder” e o cliente KStars, foi criado o driver “indi-esp”. Esse driver desempenha o papel de facilitador na comunicação entre os dispositivos, garantindo o correto funcionamento do sistema de forma integrada.

Com a conclusão de todo o trabalho, foi possível atingir o objetivo de transformar o telescópio, que anteriormente era uma montagem simples, em um instrumento avançado e automatizado, graças à implementação bem-sucedida do sistema GoTo e à eficiente utilização do ESP-IDF e do INDI.

Após finalizar a montagem do telescópio com o sistema GoTo funcionando, foram realizados testes para verificar a precisão do direcionamento aos alvos desejados. Durante esses testes, foram detectadas algumas pequenas discrepâncias, mas que, ao reposicionar manualmente o telescópio para um astro específico e acionar o sistema de rastreamento, existe um notável benefício comparado com um sistema sem GoTo. Especificamente, foi possível rastrear astros distantes, como Júpiter, mantendo o objeto dentro do campo de visão da ocular por até 20 minutos, utilizando uma ocular de 34mm.

Essa capacidade de rastreamento contínuo é um avanço em relação aos telescópios sem o sistema GoTo, onde o astro rapidamente sai do campo de visão devido à rotação do planeta Terra, exigindo correções manuais frequentes para mantê-lo adequadamente apontado. Com a implementação do GoTo, essa tarefa desafiadora foi simplificada, per-

mitindo uma observação mais detalhada e prolongada do céu estrelado sem perder o foco e o encanto dos astros distantes.

Conclui-se que, embora os objetivos propostos tenham sido alcançados, este projeto, que abrange diversas áreas do conhecimento, ainda apresenta amplo espaço para melhorias. Algumas sugestões para futuras implementações incluem:

1. Integrar módulos de giroscópio, acelerômetro e bússola à base para facilitar a calibração e verificação de desvios de rota.
2. Construir uma plataforma equatorial para a montagem dobsoniana, aproveitando o melhor dos dois mundos e combinando as vantagens de uma base equatorial.
3. Utilizar o módulo Bluetooth da ESP-32 para permitir a conexão com dispositivos móveis, como celulares e tablets, proporcionando controle e monitoramento remotos.
4. Adicionar um conjunto de baterias recarregáveis para tornar a base independente da oferta de eletricidade, permitindo sua utilização em locais remotos ou sem acesso a uma fonte de energia.
5. Criação de uma placa de circuito impresso para devida integração de toda a eletrônica do projeto.
6. Substituição dos Drivers DVR8825 por mais avançados como, por exemplo, TMC2208 para a diminuição de barulho e vibração dos motores.
7. Adição de um joystick para controle manual do telescópio.

Essas melhorias têm o potencial de aprimorar significativamente a funcionalidade e a versatilidade do projeto, tornando-o ainda mais completo e eficiente em suas aplicações.

Por fim, o trabalho abrange conhecimentos nas áreas de Astronomia, Mecânica, Eletrônica e Software, o que aumenta significativamente a complexidade da integração de todo o projeto. No entanto, essa jornada proporcionou aos autores um aprendizado profundo nessas áreas, permitindo-lhes adquirir habilidades sólidas em sistemas embarcados que envolvem calibrações, sincronizações e testes.

É esperado que esse projeto não apenas amplie o conhecimento nessas disciplinas, mas também se torne uma fonte valiosa de contribuição para futuras criações de entusiastas interessados em construir seus próprios telescópios amadores com sistema GoTo.

Ao compartilhar o trabalho e a experiência, é desejado inspirar e capacitar outras pessoas a se aventurarem nesse emocionante mundo da astronomia e eletrônica, proporcionando-lhes uma base sólida para explorar e expandir ainda mais os horizontes da ciência e da tecnologia aplicada à observação celestial. Compartilhar conhecimento é a chave para o crescimento contínuo da comunidade científica e o estímulo à inovação.

Referências

- ANGELIS, H. de. *Montagem Dobsoniana completa*. 2019. <<https://epistulaeastronomicae.wordpress.com/2019/04/28/dobsonian-base-skywatcher-200pds-newtonian-telescope/>>. [Online; accessed 05-September-2022]. Citado 2 vezes nas páginas 13 e 32.
- ARDUINO. *Arduino Hardware*. 2022. <<https://www.arduino.cc/en/hardware>>. [Online; accessed 20-September-2022]. Citado na página 23.
- COMMUNITY, K. *KStar Website*. 2022. <<https://edu.kde.org/kstars/>>. [Online; accessed 18-September-2022]. Citado na página 71.
- DOBSON, J. *John Lowry Dobson*. 2008. <https://pt.wikipedia.org/wiki/John_Lowry_Dobson>. [Online; accessed 05-September-2022]. Citado na página 31.
- DYNAPAR. *Magnetic Encoders*. 2023. <https://www.dynapar.com/technology/encoder_basics/magnetic_encoder/>. [Online; accessed 09-MAY-2023]. Citado na página 57.
- ELETROGATE. *ESP32 WROOM*. 2022. <<https://www.eletrogate.com/modulo-wifi-esp32-bluetooth-30-pinos>>. [Online; accessed 15-September-2022]. Citado 2 vezes nas páginas 13 e 49.
- FEIS.UNESP. *Motor de Passo*. 2013. <<https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf>>. [Online; accessed 06-September-2022]. Citado 3 vezes nas páginas 13, 35 e 36.
- FILIPEFLOP. *CNC Shield V3*. 2021. <<https://www.filipeflop.com/produto/cnc-shield-v3-para-arduino-impressora-3d/#tab-accessories>>. [Online; accessed 06-September-2022]. Citado 2 vezes nas páginas 13 e 54.
- FREERTOS. *FreeRTOS*. 2023. <<https://www.freertos.org/index.html>>. [Online; accessed 26-July-2023]. Citado na página 51.
- FREERTOS. *FreeRTOS tickrate*. 2023. <https://www.freertos.org/FreeRTOS_Support_Forum_Archive/November_2013/freertos_configTICK_RATE_HZ_8fc54ac4j.html>. [Online; accessed 26-July-2023]. Citado na página 52.
- IEEE. *IEEE 802.11*. 2022. <https://en.wikipedia.org/wiki/IEEE_802.11>. [Online; accessed 15-September-2022]. Citado 2 vezes nas páginas 15 e 41.
- INDI. *INDI Open Astronomy Instrumentation*. 2022. <<https://indilib.org/>>. [Online; accessed 21-September-2022]. Citado 2 vezes nas páginas 23 e 24.
- INDI. *Indi Driver Tutorial*. 2023. <<http://docs.indilib.org/drivers/>>. [Online; accessed 05-July-2023]. Citado na página 68.
- INSTRUMENTS, N. *Encoders*. 2017. <<https://docplayer.com.br/8741819-Ni-com-serie-de-conceitos-basicos-de-medicoes-com-sensores.html>>. [Online; accessed 20-April-2023]. Citado 3 vezes nas páginas 13, 38 e 39.

- INSTRUMENTS, T. *DVR8825 Datasheet*. 2022. <<https://www.ti.com/lit/gpn/drv8825>>. [Online; accessed 15-September-2022]. Citado 4 vezes nas páginas 13, 14, 53 e 57.
- INTERSTATE. *Poliacetal*. 2023. <<https://www.interstateplastics.com/acetal>>. [Online; accessed 11-April-2023]. Citado na página 46.
- JONES, D. D. W. *Stepping Motors Fundamentals*. 2004. <<http://ww1.microchip.com/downloads/en/AppNotes/00907a.pdf>>. [Online; accessed 12-September-2022]. Citado 3 vezes nas páginas 35, 37 e 51.
- KELLY, R. *Montagem Dobsoniana*. 2014. <<https://astronomy.com/observing/equipment-use/2014/04/telescopes-101?page=3>>. [Online; accessed 05-September-2022]. Citado 3 vezes nas páginas 13, 31 e 32.
- KING, H. C. *THE HISTORY OF THE TELESCOPE*. [S.l.]: Dover Publications, Inc., 1995. Citado na página 23.
- KRUGER, B. *Arduino CNC Shield*. 2015. <<https://blog.protoneer.co.nz/arduino-cnc-shield/>>. [Online; accessed 05-September-2022]. Citado na página 54.
- LOCATELLI, C. *Esqueleto Motor de Passo Híbrido*. 2021. <<https://www.curtocircuito.com.br/blog/motor-de-passo/introducao-ao-motor-de-passo>>. [Online; accessed 05-September-2022]. Citado 2 vezes nas páginas 13 e 37.
- LORENSINI, M. *Caixa Redução*. 2023. <<https://fregoto.com.br/mao-na-massa/caixa-de-reducao/>>. [Online; accessed 01-July-2023]. Citado na página 47.
- LORENSINI, M. *FireGoto*. 2023. <<https://fregoto.com.br/>>. [Online; accessed 01-July-2023]. Citado 2 vezes nas páginas 46 e 82.
- MANAVELLA, H. *Efeito Hall são os sensores sem contato e os mais conhecidos são os de posição e de rotação*. 2023. <<https://www.oficinabrasil.com.br/noticia/tecnicas/efeito-hall-sao-os-sensores-sem-contato-e-os-mais-conhecidos-sao-os-de-posicao-e-de-rotacao>>. [Online; accessed 26-July-2023]. Citado 2 vezes nas páginas 13 e 40.
- MARTINS, S. *Pinagem CNC Shield V3*. 2022. <<https://blog.eletrogate.com/cnc-shield-guia-pratico-e-introducao-ao-grbl/>>. [Online; accessed 14-September-2022]. Citado 2 vezes nas páginas 14 e 55.
- MARTIOLI, E. *Projeto Telescópio Dobsoniano*. 2009. <http://www.das.inpe.br/telescopia/pdfs/projeto.pdf?fbclid=IwAR2EQRZT_unWudUtYa8gQKtDnEnRBeYZQoiKGmmoR74Ag-6sOscAmuFZYqk>. [Online; accessed 16-September-2022]. Citado na página 45.
- MOHANAN, V. *Pinagem ESP32 WROOM*. 2022. <<https://www.circuitstate.com/pinouts/doit-esp32-devkit-v1-wifi-development-board-pinout-diagram-and-reference/>>. [Online; accessed 15-July-2023]. Citado 2 vezes nas páginas 13 e 50.
- NIDEC. *Datasheet Encoder RMS20-256*. 2023. <<https://www.nidec-copal-electronics.com/e/catalog/rotary-encoder/rms20.pdf>>. [Online; accessed 09-MAY-2023]. Citado na página 57.

- NUNES, B. C.; RODRIGUES, M. V. *circuito eletronico*. 2023. <https://github.com/brunocmo/indi-esp/tree/main/projeto_eletronico>. [Online; accessed 20-July-2023]. Citado 2 vezes nas páginas 14 e 91.
- NUNES, B. C.; VINÍCIUS, M. *indi-esp*. 2023. <<https://github.com/brunocmo/indi-esp>>. [Online; accessed 20-July-2023]. Citado na página 45.
- ONSHAPE. *onshape*. 2023. <<https://www.onshape.com>>. [Online; accessed 01-July-2023]. Citado na página 45.
- ONSTEP. *Onstep*. 2023. <<https://onstep.groups.io/g/main/wiki/Home>>. [Online; accessed 01-July-2023]. Citado na página 82.
- PROJECTS, S. *Módulo GPS NEO-6M*. 2021. <<https://www.smartprojectsbrasil.com.br/modulo-gps-gy-neo6m-v2>>. [Online; accessed 13-September-2022]. Citado 2 vezes nas páginas 14 e 56.
- ROHDE-SCHWARZ. *Formato de quadro UART*. 2022. <https://www.rohde-schwarz.com/br/produtos/teste-e-medicao/osciloscopios/educational-content/compreender-uart_254524.html>. [Online; accessed 05-September-2022]. Citado 2 vezes nas páginas 13 e 44.
- S., T. A. *Computer Networks*. 4. ed. [S.l.]: Prentice Hall PTR, 2003. Citado 3 vezes nas páginas 13, 40 e 41.
- SARAIVA, K. de Souza Oliveira Filho; Maria de F. O. *Astronomia e astrofísica*. [S.l.]: Editora Livraria da Física, 2017. Citado 5 vezes nas páginas 13, 27, 28, 29 e 30.
- SEMICONDUCTORS, N. *I2C-bus specification and user manual*. 2021. <<https://www.nxp.com/docs/en/user-guide/UM10204.pdf>>. [Online; accessed 08-September-2022]. Citado 4 vezes nas páginas 13, 41, 42 e 43.
- SYSTEMS, E. *ESP32 Technical Reference Manual*. 2022. <https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf>. [Online; accessed 05-September-2022]. Citado 4 vezes nas páginas 23, 24, 42 e 50.
- TAMAWAGA. *Datasheet Motor de Passo Tamagawa TS3103N40*. 2021. <<https://www.tjsolution.com/TAMAGAWA/1648n1ej.pdf>>. [Online; accessed 06-September-2022]. Citado 2 vezes nas páginas 13 e 37.
- TAMAWAGA. *Datasheet Motor de Passo Tamagawa TS3103N40*. 2021. <<https://www.tjsolution.com/TAMAGAWA/1648n1ej.pdf>>. [Online; accessed 06-September-2022]. Citado 2 vezes nas páginas 13 e 37.
- TAMAWAGA. *Datasheet Motor de Passo Tamagawa TS3103N40*. 2021. <<https://www.tjsolution.com/TAMAGAWA/1648n1ej.pdf>>. [Online; accessed 06-September-2022]. Citado 2 vezes nas páginas 13 e 52.
- TAMAWAGA. *Datasheet Motor de Passo Tamagawa TS3103N40*. 2021. <<https://www.tjsolution.com/TAMAGAWA/1648n1ej.pdf>>. [Online; accessed 06-September-2022]. Citado na página 46.

- U-BLOX. *Módulo GPS Neo-6m Datasheet*. 2004. <https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf>. [Online; accessed 12-September-2022]. Citado na página 55.
- VAUGHAN, C. *The Ultimate in Astronomy Gadgets: GOTO Telescopes*. 2016. <<https://www.space.com/33586-goto-telescopes-ultimate-astronomy-gadget.html>>. [Online; accessed 11-September-2022]. Citado na página 23.
- WIKI. *Ponte H*. 2022. <https://pt.wikipedia.org/wiki/Ponte_H>. [Online; accessed 10-September-2022]. Citado 2 vezes nas páginas 13 e 38.
- WIKIPEDIA. *Telescope Mount*. 2022. <https://en.wikipedia.org/wiki/Telescope_mount>. [Online; accessed 05-September-2022]. Citado na página 30.
- WIKIPEDIA. *Universal asynchronous receiver-transmitter*. 2022. <https://en.wikipedia.org/wiki/Universal_asynchronous_receiver-transmitter>. [Online; accessed 05-September-2022]. Citado 3 vezes nas páginas 13, 42 e 43.
- WIKIPEDIA. *Telescópio newtoniano*. 2023. <https://es.wikipedia.org/wiki/Telescopio_newtoniano#/media/Archivo:NewtonsTelescopeReplica.jpg>. [Online; accessed 07-July-2023]. Citado 2 vezes nas páginas 13 e 30.
- ZANOTTA, D. C.; CAPPELLETTO, E.; MATSUOKA, M. T. O GPS: unindo ciência e tecnologia em aulas de física. *Revista Brasileira de Ensino de Física*, FapUNIFESP (SciELO), v. 33, n. 2, jun 2011. Citado 3 vezes nas páginas 13, 33 e 34.

Anexos

ANEXO A – Diagrama Eletrônico do Sistema

Figura 55 – Diagrama Eletrônico do sistema (NUNES; RODRIGUES, 2023).

