

CENTRO PAULA SOUZA
FATEC SANTO ANDRÉ
Tecnologia em Mecatrônica Industrial

Sabrina Darim de Oliveira

Drive de Motor de Passo Realimentado por *Encoder*

Santo André

2021

Sabrina Darim de Oliveira

Drive de Motor de Passo Realimentado por *Encoder*

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Mecatrônica da Fatec Santo André como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial.

Orientador: Prof. Dr. Edson Caoru Kitani

Santo André

2021

FICHA CATALOGRÁFICA

O482d

Oliveira, Sabrina Darim de
Drive de motor de passo realimentado por encoder / Sabrina
Darim de Oliveira. - Santo André, 2021. – 54f: il.

Trabalho de Conclusão de Curso – FATEC Santo André.
Curso de Tecnologia em Mecatrônica Industrial, 2021.

Orientador: Prof. Dr. Edson Caoru Kitani

1. Mecatrônica. 2. Sistema. 3. Desenvolvimento. 4.
Monitoramento. 5. Controle. 6. Tecnologia. 7. Drive de potência.
8. Motor de passo. 9. Encoder. 10. Malha fechada. I. Drive de
motor de passo realimentado por encoder.

629.892

Sabrina Darim de Oliveira

Drive de Motor de Passo Realimentado por *Encoder*

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Mecatrônica da Fatec Santo André como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial.

Orientador: Prof. Dr. Edson Caoru Kitani

MEMBROS DA BANCA EXAMINADORA

Presidente da Banca

Prof. Dr. Edson Caoru Kitani

Fatec Santo André

Primeiro membro da Banca

Prof. Fernando Garup Dalbo

Fatec Santo André

Segundo membro da Banca

Prof. Wesley Torres

Fatec Santo André

Local: Fatec Santo André

Horário: 19h00

Data: 20/12/2021

Santo André

2021

A meus pais pela ajuda e apoio nas minhas conquistas e realizações.

AGRADECIMENTOS

Agradeço a Deus por ter me abençoado, protegido e capacitado durante toda a minha vida para hoje ter a oportunidade de realizar este trabalho e de ter superado todas as dificuldades para a conclusão dele.

A meus pais por todo o cuidado, carinho e apoio em todos os momentos sendo cruciais para a minha formação.

A meus familiares e amigos pelo apoio prestado durante a realização deste trabalho.

Ao meu orientador Prof. Dr. Edson Caoru Kitani por todo o tempo dedicado no auxílio a minha formação direcionados a conclusão deste trabalho.

Ao professor Fernando Garup Dalbo, pela ajuda e orientação para a organização do projeto e documentação.

“O passado serve para evidenciar as nossas falhas e dar-nos indicações para o progresso do futuro.”

HENRY FORD

RESUMO

Uma falha muito comum, que frequentemente se observa nos projetos com motores de passo, é a perda de referência do motor. Em equipamentos como impressoras 3D e *routers* CNC, esse tipo de falha produz uma enorme perda de horas trabalhadas e material. Baseado nessa observação, este projeto propõe um sistema que consiste em um circuito de controle e monitoramento de um motor de passo em malha fechada utilizando um drive de potência com realimentação por meio de um *encoder*, que fará a monitoração dos passos do motor. Esse sistema irá assegurar que o movimento solicitado ao motor será executado, e se houver algum erro entre a posição real e a desejada, o sistema tentará corrigir, caso contrário gerará um alarme. O circuito proposto usará um microcontrolador Arduino que compara as informações que são enviadas pelo drive ao motor de passo com as informações recebidas do *encoder* sobre o movimento realizado pelo motor. Ao analisar essas informações o Arduino fará a comparação do movimento do eixo do motor, com o que foi solicitado pelo drive, e havendo alguma diferença entre eles o sistema tentará compensar, e se não for possível gerará um alarme para o usuário.

Palavras chaves: Motor de passo. Drive de Motor de Passo. Malha Fechada por *Encoder*.

ABSTRACT

A widespread failure observed in stepper motor designs is the loss of reference of the motor. Equipment, such as 3D printers and CNC routers, this type of failure produces a huge loss of hours worked and material. Based on this observation, this project proposes a system consisting of a control and monitoring of a stepper motor with a closed-loop circuit. A drive with feedback through an optical encoder, will monitor the motor steps. This system will ensure that the movement requested will be executed, and if there is a discrepancy between the actual and the desired position, the system will try to correct it. Otherwise, it will generate an alarm. The proposed circuit will use an Arduino microcontroller that compares the information sent by the driver to the stepper motor with the information received from the encoder about the movement performed by the motor. When analyzing this information, the Arduino will compare what was done by the engine with what was requested by the driver. If there is any difference between them, the system will try to fix it, and if it is not possible, it will cause an alarm for the user, also assuring the constant reference of the stepper motor.

Keywords: Stepper motor. Drive. Encoder. Closed Loop. Feedback. Monitoring. Control.

LISTA DE ILUSTRAÇÕES

Figura 1.	Motor de passo de ímã permanente.....	19
Figura 2.	Princípio de funcionamento do motor de relutância variável.....	20
Figura 3.	Construção do eixo do motor de passo híbrido.	20
Figura 4.	Funcionamento do motor de passo híbrido.....	21
Figura 5.	Drives unipolares e bipolares	22
Figura 6.	Construção interna de um <i>encoder</i>	23
Figura 7.	Resolução de <i>encoders</i> incrementais	24
Figura 8.	Resolução de <i>encoders</i> absolutos	25
Figura 9.	Diagrama de blocos de um sistema de malha fechada.....	26
Figura 10.	Fluxograma do sistema em malha fechada	27
Figura 11.	Motor de passo Nema 17.....	28
Figura 12.	Drive TMC2209	28
Figura 13.	<i>Encoder</i> incremental	29
Figura 14.	Arduino MEGA	30
Figura 15.	Fonte 24V	30
Figura 16.	Fonte 5V	31
Figura 17.	Esquema elétrico.....	31
Figura 18.	Esquema elétrico detalhado (1).....	32
Figura 19.	Esquema elétrico detalhado (2).....	33
Figura 20.	Placa de montagem	34
Figura 21.	Motor e <i>encoder</i>	34
Figura 22.	Sinal de pulso e <i>encoder</i> A.....	35
Figura 23.	Sinais do <i>encoder</i> A e <i>encoder</i> B.....	36
Figura 24.	Sinais de pulso e do <i>encoder</i>	36
Figura 25.	Microcontrolador PIC 18F4550.....	37

Figura 26. Sentido horário	38
Figura 27. Sentido anti-horário.....	38
Figura 28. Pulsos enviados pelo PIC e pelo Arduino ao drive	39
Figura 29. Resolução do motor e do <i>encoder</i>	40
Figura 30. Atraso do <i>encoder</i> em relação aos pulsos no Arduino	41

LISTA DE TABELAS

Tabela 1 - Relação de pulsos.....	39
-----------------------------------	----

LISTA DE ABREVIATURAS E SIGLAS

UV – Ultravioleta

FDM – *Fused Deposition Modeling*

CNC – *Computer Numerical Control*

CAD – *Computer Aided Design*

SUMÁRIO

1. INTRODUÇÃO	15
1.1 Motivação	16
1.2 Objetivo	16
1.3 Metas e resultados desejados.....	16
2. FUNDAMENTAÇÃO TEÓRICA.....	17
2.1 Aplicações de motores de passo	17
2.1.1 Impressoras 3D.....	17
2.1.2 <i>Routers</i> CNC.....	17
2.1.3 Impressoras à jato de tinta e a laser	18
2.2 Motores de passo	18
2.2.1 Motores de Passo com Imã Permanente	19
2.2.2 Motor de Passo de Relutância Variável	19
2.2.3 Motor de Passo Híbrido	20
2.3 Drives	21
2.3.1 Tipos de drives	22
2.4 <i>Encoders</i>	22
2.4.1 Tipos de <i>encoder</i>	24
3. METODOLOGIA.....	26
3.1 Controle de motor de passo em malha fechada.....	26
3.2 Motor de Passo Nema 17	27
3.3 Drive TMC2209.....	28
3.4 <i>Encoder</i>	29
3.5 Arduino MEGA	29
3.6 Alimentação.....	30
3.7 Esquema elétrico	31
3.8 Montagem do <i>hardware</i>	33

4. TESTES E RESULTADOS	35
5. CONCLUSÃO.....	43
6. REFERÊNCIAS TEXTUAIS.....	44
7. REFERÊNCIAS DAS IMAGENS	48
APÊNDICE A – CÓDIGO ARDUINO.....	49
ANEXO A – CIRCUITO DRIVE TMC2209	53

1. INTRODUÇÃO

Em meados da década de 1980, uma técnica conhecida como estereolitografia foi desenvolvida pelo norte-americano Chuck Hull, e que permitia a criação de objetos 3D a partir de uma luz Ultravioleta (UV) direcionada a um ponto específico em um reservatório de uma resina que se solidificava, formando então uma impressão em 3D. Mas foi em 1989 que S. Scott Crump inventou e patenteou a forma mais comum das impressoras 3D que temos atualmente, a impressão FDM (*Fused Deposition Modeling*), conhecida como modelagem por deposição fundida. Esse método consiste em fundir um material e depositá-lo em camadas para modelar o objeto projetado (3D, [2020]).

Outra técnica semelhante às impressoras 3D são as *routers* CNC (*Computer Numerical Control*), onde um modelo 3D é produzido através de um *software* CAD (*Computer Aided Design*) e depois é transferido para a máquina executar o projeto. Porém, ao invés de modelar o objeto aplicando o material, as *routers* CNC usinam a matéria prima do produto através de cortes e remoções de material com uma alta precisão, podendo a matéria prima ser de: metal, madeira, acrílico, entre outros.

Não é só na utilização de *softwares* de desenhos em 3D que as impressoras se assemelham as *routers* CNC, outro ponto em comum entre eles é que para a produção das peças tem que se ter uma precisão muito grande para executar o que foi feito em *software*, por isso se tem a utilização de motores de passo para a movimentação pelos eixos x, y e z, tanto nas impressoras 3D como em *routers* CNC (CNC, C., 2021).

Desde então, houve grandes avanços da tecnologia em impressão 3D. Contudo, ainda há muitos problemas que os usuários encontram ao utilizarem uma impressora 3D. Um dos grandes problemas é quando, por algum motivo não previsto ou planejado, seja por queda de energia, por falta de filamento etc., a impressora não consegue continuar sua operação gerando uma peça inacabada, dado que a máquina não consegue continuar seu processo por perda da referência. Isto obriga o usuário a iniciar toda a sequência de operação, perdendo o material já utilizado e o tempo gasto, que dependendo da peça a ser feita, pode levar horas, e até mesmo dias. Entretanto, esse problema não é único das impressoras 3D, ocorre também em *routers* CNC.

Equipamentos de impressão ou usinagem de alto custo trabalham com servomotores AC e com *encoders* absolutos que ajudam a evitar o problema de perda de referência, mas a maioria dos equipamentos de baixo custo dispõe apenas de motores de passo sem realimentação da posição. Além do problema da perda de posição quando o equipamento é desligado, as impressoras de baixo custo trabalham em malha aberta, o que torna mais impreciso o controle de posição do eixo do motor. Contudo, uma possível solução para esses problemas seria o uso de um *encoder* para realimentar o drive do motor de passo com a informação da posição do eixo do motor. Com essa informação, o drive saberia os passos exatos que o motor girou e assim a máquina não perderia a sua referência de posição e, horas de trabalho poderiam ser recuperadas.

1.1 Motivação

Este trabalho utilizará uma malha fechada para o monitoramento do movimento do motor de passo tendo como motivação as perdas nos processos produtivos de *routers* CNC e impressoras 3D devido à falta de informação do movimento e posição do motor, ocasionada devido à queda de energia ou perda de sincronização dos passos, quando é normalmente utilizado em malha aberta. Por conta disso, esse projeto utilizará uma malha fechada ao invés de malha aberta, garantindo assim o monitoramento dos movimentos do motor e do eixo mecânico associado.

1.2 Objetivo

Desenvolver um sistema de malha fechada de um drive de motor de passo com realimentação por *encoder* buscando monitorar possíveis falhas do movimento do motor de passo em relação à referência de pulsos que foi enviado.

1.3 Metas e resultados desejados

- Melhorar a precisão de posicionamento do motor de passo;
- Garantir o posicionamento do motor de passo por malha fechada.

2. FUNDAMENTAÇÃO TEÓRICA

Para uma revisão bibliográfica consistente sobre o tema do projeto de pesquisa foram consultados vários trabalhos relacionados sobre o funcionamento das impressoras 3D, *routers* CNC, motores de passo, drives de motores de passo e *encoders*. As seções a seguir trazem um resumo dos principais aspectos que foram considerados para o desenvolvimento do projeto.

2.1 Aplicações de motores de passo

Discute-se nesta seção as principais aplicações atuais com motores passo. Naturalmente, a quantidade de aplicações comerciais que usam o motor de passo é muito maior do que os apresentados neste texto, mas estão destacados apenas aqueles que foram importantes para compreender as possibilidades de aplicação do projeto proposto.

2.1.1 Impressoras 3D

As impressoras 3D são uma das aplicações que podemos ter de motores de passo, devido à alta exigência de movimentos precisos. O princípio do funcionamento elétrico consiste em uma placa de controle que recebe as informações do *software* CAD sobre o projeto a ser impresso e transforma isso em informação para os drives que irão controlar o movimento dos motores de passo, deslocando o bico injetor, ou a mesa de depósito de material, dependendo do tipo de impressora, para assim o objeto ser formado. (RAZGRIZ, 2020)

2.1.2 *Routers* CNC

Sobre as máquinas *routers* CNC podemos dizer que:

É um equipamento CNC controlado por computador, construído em estrutura metálica e que é movimentado em vários eixos por fusos de esferas ou cremalheiras através de motores de passo ou servo motores sobre guias lineares que executa trabalhos de gravação e corte 2D e usinagem 3D com precisão decimal através de um *Spindle* com vários tipos de fresas em substratos como Madeira, Acrílico, MDF, ACM, PVC expandido, Policarbonato, Chapas metálicas, entre outros. (CNC, M., 2021)

Em relação aos motores utilizados em *routers* CNC os mais comuns são os motores de passo e os servomotores. Os motores de passo “*são a alternativa mais barata e oferecem bom desempenho e confiabilidade, mas geralmente não são um sistema de circuito fechado*” (SILVA, J., 2015). Já os servomotores trabalham em

malha fechada, o que significa que eles recebem a informação do movimento a ser feito, realizam o movimento e mandam para o sistema de controle o que foi realizado, assim o sistema consegue verificar se o que foi solicitado foi executado.

2.1.3 Impressoras à jato de tinta e a laser

Além das impressoras 3D, as impressoras à jato de tinta e as impressoras à laser também utilizam motores de passo em seus circuitos. As impressoras à jato de tinta utilizam o motor de passo para mover o conjunto da cabeça de impressão ao longo do papel (HAT, 2005), necessitando grande precisão e controle de velocidade. Quanto as impressoras à laser “o *micromotor de passo é muito atuante, pois há a necessidade de um controle mais rigoroso para rotacionar uma rede de difração, de modo que a frequência do laser possa ser ajustada.*” (IMPRESSORAJATO, 2021).

2.2 Motores de passo

O motor é uma máquina que consegue converter energia elétrica em energia mecânica, e o motor de passo, em específico, possui a característica de executar esse movimento de rotação de um modo bastante particular. O movimento do motor de passo depende de uma sequência ordenadas de pulsos que promovem um realinhamento entre dois campos magnéticos. Assim, o motor de passo trabalha dividindo um período de 2π radianos (360°) em vários passos.

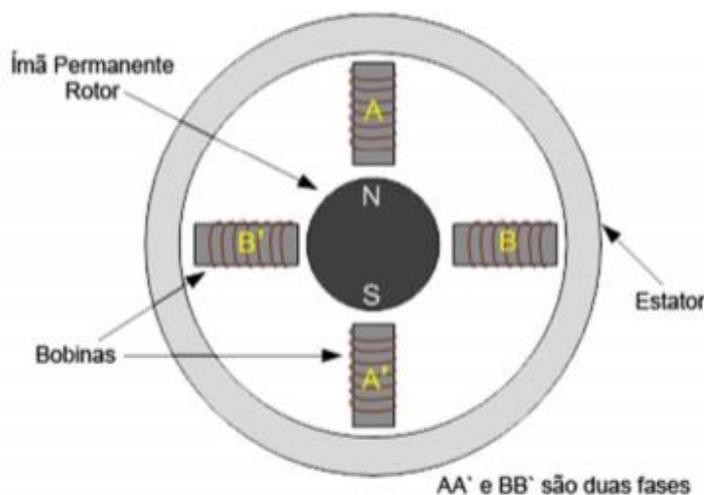
O motor de passo recebe um sinal elétrico na forma de pulso, o que equivale a um passo de movimento do rotor. Por conta dessa natureza de operação, o motor de passo necessita de um circuito de controle para que os sinais de pulsos recebidos estejam de acordo com a quantidade desejada de passos que o motor executará. E a precisão desse controle é fundamental, por isso os motores de passos são utilizados em equipamentos que precisem de um controle de posição preciso e estável. (ELECTRONICS, 2021)

Existem três tipos de motores de passo, os de ímã permanente, de relutância variável e os híbridos.

2.2.1 Motores de Passo com Ímã Permanente

Esse tipo de motor possui um rotor magnetizado ou de ímã permanente como mostra a Figura 1. O motor irá ter a bobina do estator acionada e fará com que o polo magnético do eixo se alinhe com o polo oposto das bobinas, fazendo o eixo do motor girar. Esse tipo de construção de motor possui a vantagem de *“ter um campo magnético permanente que se soma ao campo magnético das bobinas, dando uma potência, ou torque, maior na partida.”* (UNESP, 2013). A Figura 1 ilustra uma visão esquemática do motor de passo com ímã permanente. Observa-se na figura a existência de dois campos eletromagnéticos que serão formados pela corrente que passam nas bobinas A e B, bem como o rotor que é formado por um ímã permanente.

Figura 1. Motor de passo de ímã permanente.

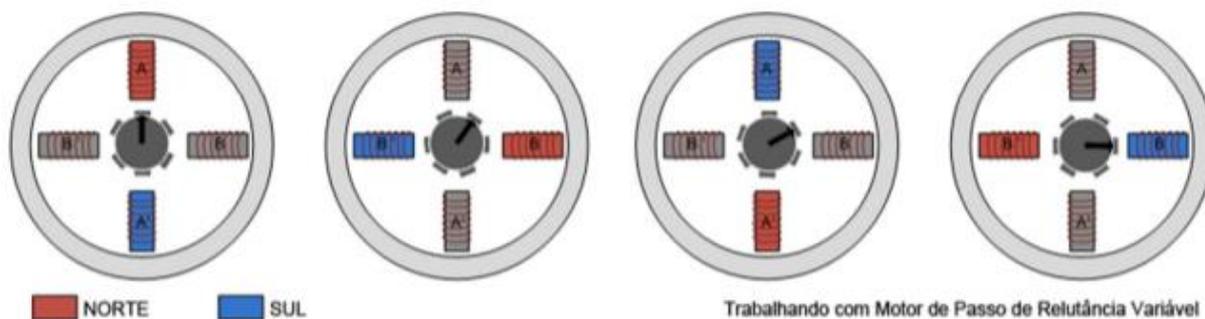


Fonte: UNESP, 2013

2.2.2 Motor de Passo de Relutância Variável

Esse tipo de construção de motor se difere de um motor de ímã permanente pelo fato de seu eixo ser de ferro e não um ímã, por tanto, não possui um campo magnético permanente e por consequência, apresenta um torque menor na saída. (UNESP, 2013). O seu eixo de ferro possui dentes que se alinham com uma das bobinas do estator e se desalinham com outras, fazendo com que o motor gire, como pode ser observado na Figura 2.

Figura 2. Princípio de funcionamento do motor de relutância variável.

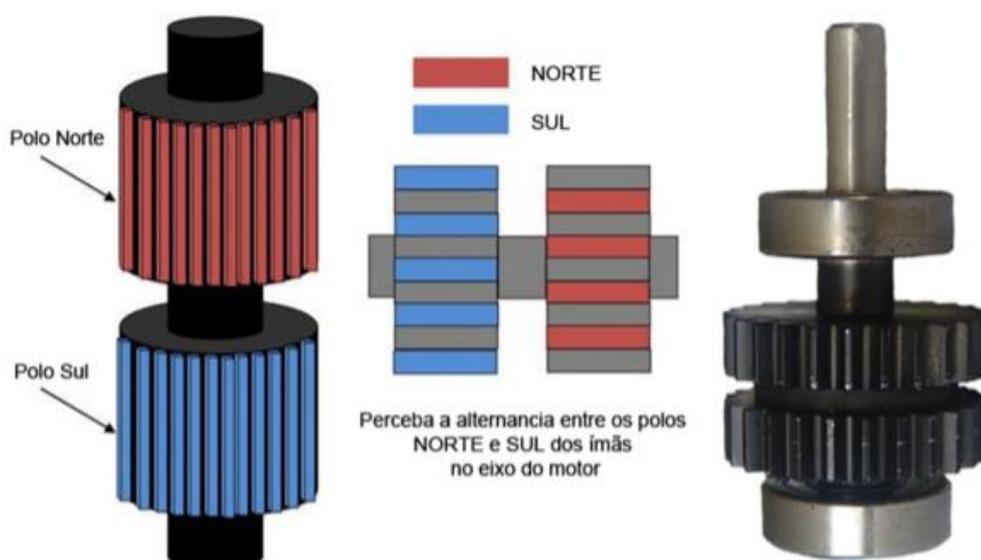


Fonte: UNESP, 2013

2.2.3 Motor de Passo Híbrido

O motor de passo híbrido é uma combinação tanto do motor de ímã permanente como o motor de relutância variável, tendo como característica maiores torques e precisão, e é atualmente o mais aplicado no mercado. Possui em seu eixo “dois grupos de dentes, um com o polo sul saliente e o outro com o polo norte, de modo que os dentes fiquem alternados” (UNESP, 2013). A Figura 3 ilustra o detalhe do rotor que é formado por dois polos de um ímã permanente, e os dentes que aumentam a densidade magnética no topo de cada dente da engrenagem que forma o rotor.

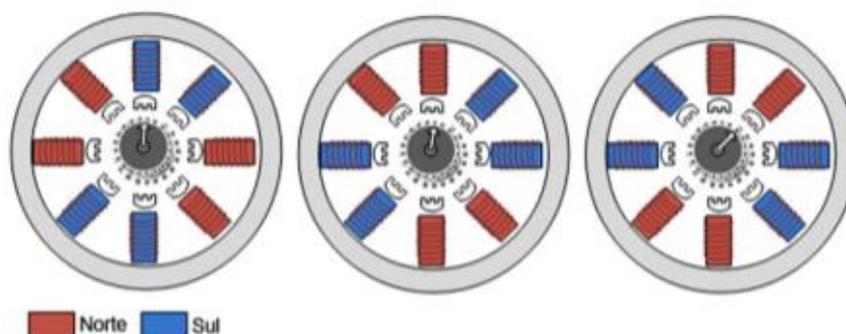
Figura 3. Construção do eixo do motor de passo híbrido.



Fonte: UNESP, 2013

Assim como nos outros dois tipos de motores de passo apresentados anteriormente, no motor híbrido a ligação das bobinas deve ser de forma sequencial para que o motor funcione adequadamente, como ilustrado na Figura 4.

Figura 4. Funcionamento do motor de passo híbrido



Fonte: UNESP, 2013

Nota-se que os motores de passo não funcionam como os motores de corrente contínua série, que conseguem produzir movimento rotativo quando são alimentados por uma fonte ou bateria. No caso dos motores de passo, é necessário criar um tipo de sequência de pulsos que gere um campo girante para criar uma força torcional entre o rotor e o campo girante. Assim, veremos que o rotor irá perseguir o campo magnético gerado pelas bobinas para tentar se alinhar. Logo, a produção desse campo magnético girante exigirá a construção de um circuito eletrônico que reproduza esse campo. O equipamento em questão é conhecido como Drive de Motor de Passo e será apresentado em detalhes na próxima seção.

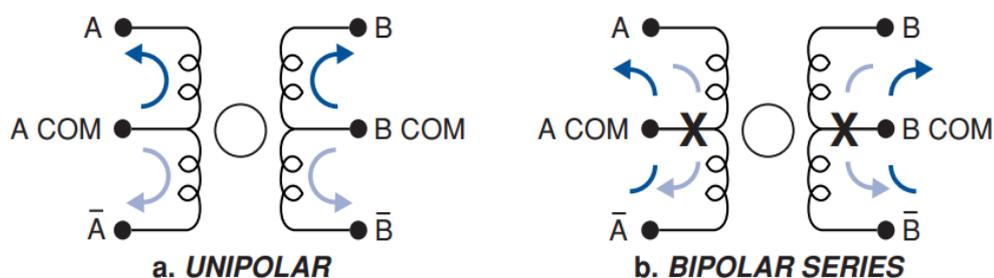
2.3 Drives

O drive de motor de passo tem como função receber os sinais relativamente baixos do sistema de controle com as informações dos passos exigidos ao motor e prover um valor de corrente elétrica suficiente e correspondente para a execução dos passos e o acionamento do motor de passo (CORPORATION, 2021).

2.3.1 Tipos de drives

Existem dois tipos de drives de motores de passo, os unipolares e os bipolares. A diferença entre eles se dá pelo fato que os drives unipolares fornecem a corrente elétrica para os enrolamentos do motor de passo em apenas uma direção, já os drives bipolares podem enviar a corrente em ambas as direções (CORPORATION, 2021). Podemos ver na Figura 5 a comparação do percurso da corrente elétrica pelos enrolamentos do motor com uso de drives unipolares e bipolares em motores de passo de seis fios.

Figura 5. Drives unipolares e bipolares



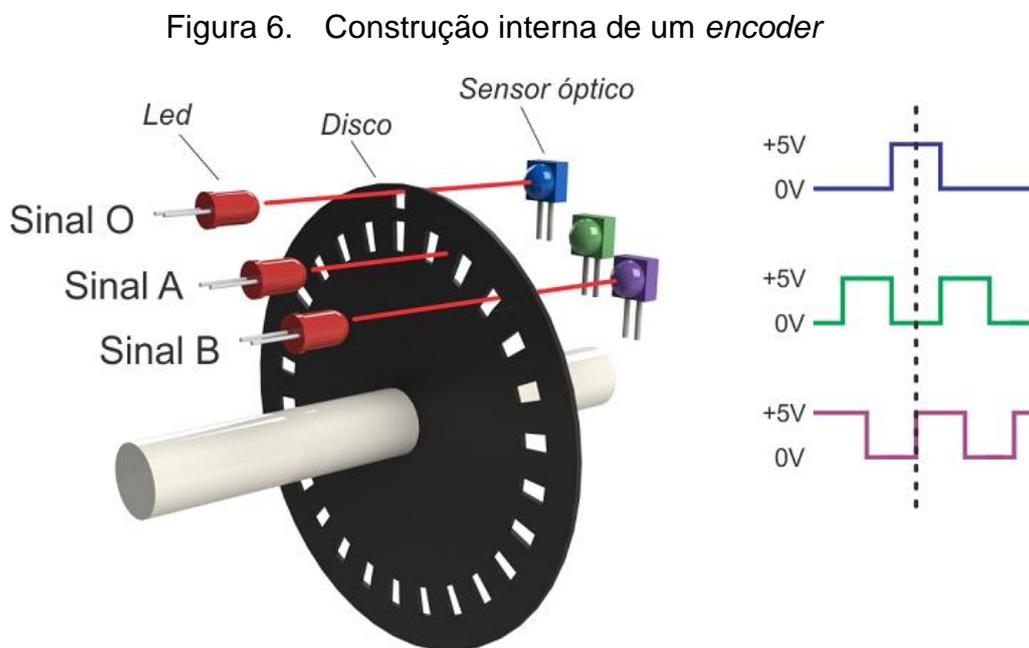
Fonte: CORPORATION, 2021

Esse conjunto formado pelo sistema de controle que envia a informação para os drives que convertem para a execução do movimento rotacional do motor de passo, é conhecido como um sistema de malha aberta, onde uma informação é enviada, processada e executada. Para o caso do sistema de malha fechada o último processo não seria a execução, e sim a comunicação do que foi executado para a comparação com o que foi solicitado, ou seja, o sistema de malha fechada controla e monitora a operação a ser realizada. No caso de motores de passo trabalhando em um sistema de malha fechada, é necessário o uso do dispositivo *encoder* para o monitoramento da rotação do motor. Esse dispositivo será apresentado na seção a seguir.

2.4 Encoders

Encoder é um dispositivo sensor responsável por monitorar a posição do motor e transformar o movimento de rotação em sinal elétrico digital. O *encoder* é formado

por um disco com marcações, um emissor e um receptor (ALMEIDA, 2017). Na Figura 6 podemos ver a representação de um *encoder* internamente.



Fonte: ALMEIDA, 2017

O sinal elétrico digital fornecido pelo *encoder* é gerado pela frequência que o feixe de luz enviado pelo emissor, passa pelas marcações do disco e é coletado pelo receptor do outro lado do disco. Esse sinal gerado é normalmente “*uma onda quadrada, proporcional ao número de marcações do encoder, de acordo com a resolução do mesmo.*” (ALMEIDA, 2017). A resolução do *encoder* equivale a quantidade de marcações presentes no disco, que é correspondente com a número de ondas quadradas que são geradas em uma volta do *encoder* (ALMEIDA, 2017). A Figura 7 apresenta alguns exemplos de discos de *encoders* incrementais em relação a suas resoluções.

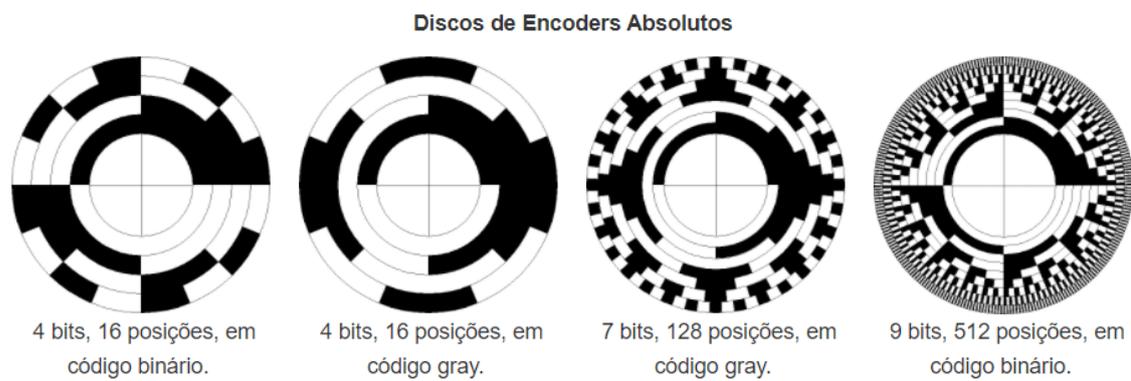
Figura 7. Resolução de *encoders* incrementais

Fonte: ALMEIDA, 2017

2.4.1 Tipos de *encoder*

Atualmente os tipos de *encoder* mais utilizados são os do tipo incremental e os absolutos. Como foi mencionado, na Figura 7 há a representação de alguns exemplos de resolução de *encoders* incrementais. Esses *encoders* possuem a característica de definir a posição do motor contando pulsos a partir do pulso zero, porém tendo como a principal desvantagem a perda de referência da posição inicial se for desligado ou se sofrer alguma queda de energia, tendo que ser feita a resincronização da posição inicial ao ser novamente ligado (CAPPELLATO, 2020).

Esse problema não encontramos nos *encoders* absolutos, onde a posição “é determinada pela leitura de um código e este é único para cada posição do seu curso” (CAPPELLATO, 2020), com isso ele não perde a referência de posição, mesmo que sofra queda de energia ou que seja desligado. Podemos analisar o disco de um *encoder* absoluto e suas resoluções na Figura 8, que nos mostra com mais clareza como cada posição tem seu código único, seja por meio de código binário ou *gray*.

Figura 8. Resolução de *encoders* absolutos

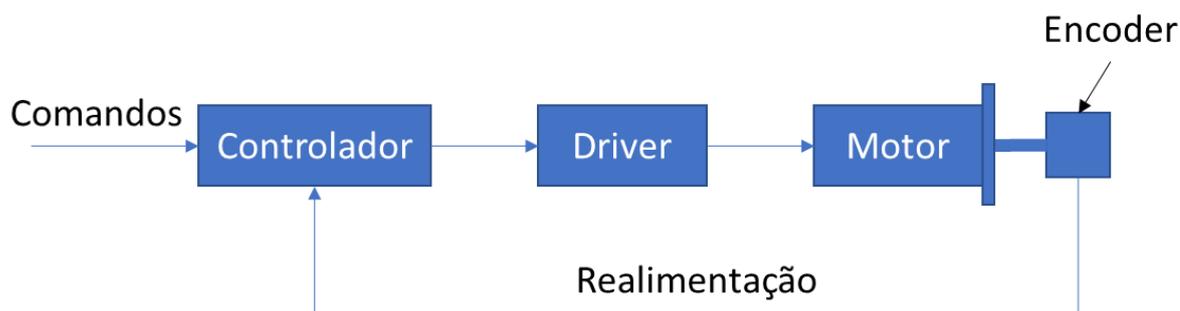
Fonte: ALMEIDA, 2017

3. METODOLOGIA

3.1 Controle de motor de passo em malha fechada

Os motores de passo são usados em sua grande maioria em circuitos de malha aberta, que são mais viáveis economicamente, porém quando se necessita um maior torque ou velocidade do motor nesse sistema pode haver uma perda de passos do motor por ocorrer a falta de sincronismo entre os pulsos gerados e execução desses pulsos pelo motor, tendo assim uma necessidade de se usar o motor em malha fechada, onde se é implementado, por exemplo, um *encoder* para a realimentação do sistema e controle de possíveis erros de posicionamento ou falhas do motor. Na Figura 9 vemos o diagrama de blocos de uma malha fechada. (SOUZA, 2007)

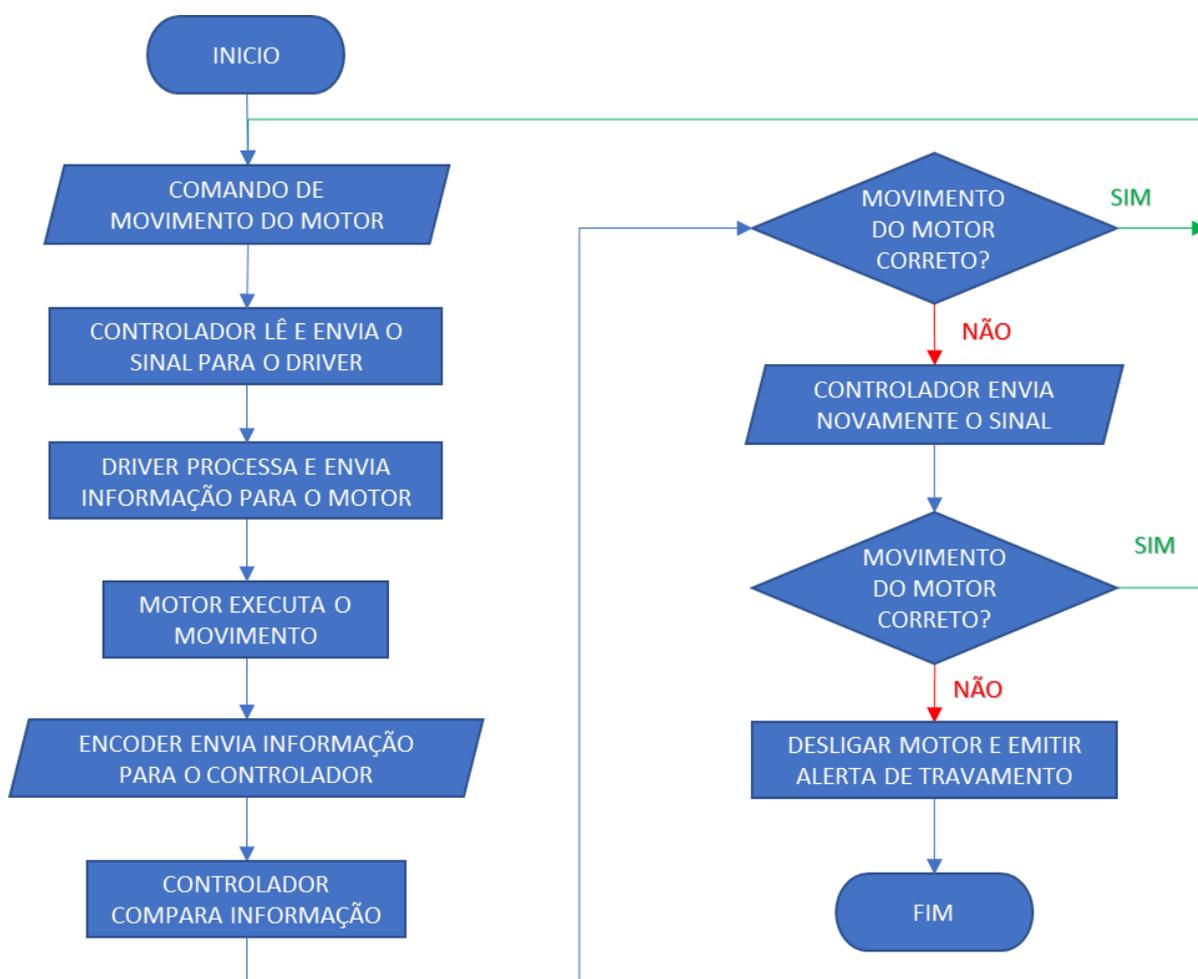
Figura 9. Diagrama de blocos de um sistema de malha fechada



Fonte: Próprio autor

Conforme o objetivo deste projeto, será utilizado um sistema de malha fechada como visto na Figura 9. Inicialmente o controlador receberá os comandos, fará a leitura do sinal e enviará um sinal equivalente ao recebido para o drive do motor de passo, que por sua vez processará o sinal e enviará os comandos para o motor. Será utilizado um *encoder* que enviará ao controlador a informação dos movimentos do motor de passo, sendo usado como realimentação, fechando a malha. A informação da realimentação será comparada pelo controlador com os comandos recebidos, e assim identificará se houve ou não perda de passos ou se o motor atuou conforme o pedido. A Figura 10 ilustra o fluxograma do sistema.

Figura 10. Fluxograma do sistema em malha fechada



Fonte: Próprio autor

Para a análise de dados foram adotados alguns equipamentos a fim de testar e verificar os resultados obtidos. Estes equipamentos serão apresentados nos tópicos a seguir com suas especificações.

3.2 Motor de Passo Nema 17

O motor de passo adotado foi o Nema 17, que possui como característica duas fases, $1,8^\circ$ de movimentação angular, o que significa que a cada pulso elétrico o motor irá rotacionar $1,8^\circ$, por tanto, são necessários 200 pulsos para a volta completa do motor, além de possuir uma tensão de alimentação de 8 a 35V. Na Figura 11 temos a imagem do motor Nema 17.

Figura 11. Motor de passo Nema 17



Fonte: MOTION, 2021

3.3 Drive TMC2209

O drive para o controle do motor de passo que será utilizado é o TMC2209 que possui tensão de alimentação 5.5 a 28V e consegue alimentar motores de até 2A atendendo o motor Nema 17 que possui 1A por fase. Na Figura 12 temos a imagem do drive.

Figura 12. Drive TMC2209



Fonte: ACELERA3D, [2021]

3.4 Encoder

Dado a escolha dos equipamentos acima, foi escolhido o *encoder* incremental PNP de 400 pulsos de resolução e alimentação de 5-26 VDC para a leitura e envio do sinal dos movimentos do motor de passo. Temos a ilustração do *encoder* na Figura 13.

Figura 13. *Encoder* incremental



Fonte: ACELERA3D, [2021]

3.5 Arduino MEGA

Para o controle e monitoramento do sistema, será utilizado o microcontrolador Arduino MEGA, que atende a quantidade de pinos de interrupção necessária do sistema. Sua função será receber os dados de movimentos e transferi-los para o drive, em seguida ele analisará a informação recebida do *encoder* sobre o movimento realizado pelo motor. Na Figura 14 temos a imagem do Arduino MEGA que será utilizado.

Figura 14. Arduino MEGA



Fonte: ARDUINO, [202-?]

3.6 Alimentação

A alimentação do sistema será realizada por uma fonte chaveada estabilizada bivolt com saída de 24V para alimentar o motor e o *encoder* apresentada na Figura 15.

Figura 15. Fonte 24V

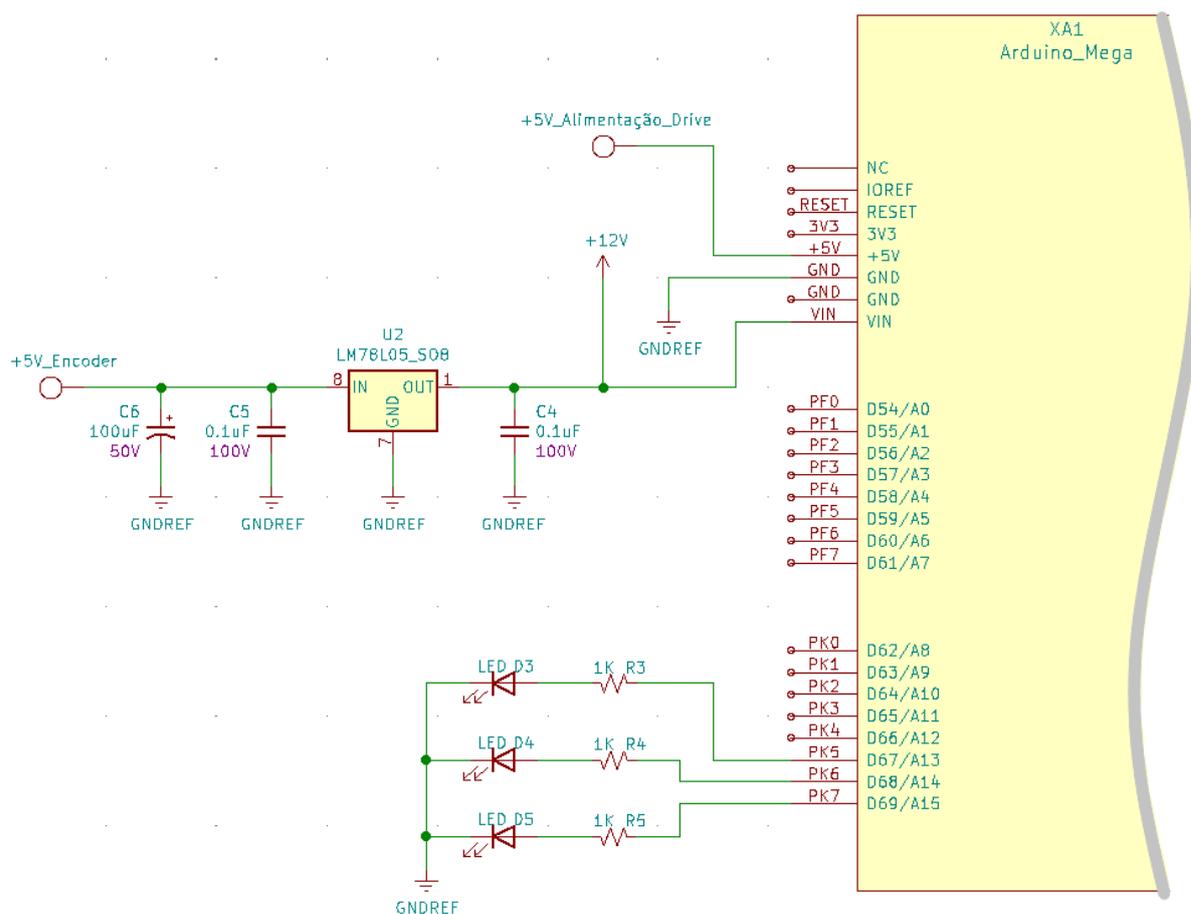


Fonte: ELECTRONICS, [202-?]

Para a alimentação do Arduino e drive será utilizada uma outra fonte que fornecerá na saída 5V que podemos ver na Figura 16.

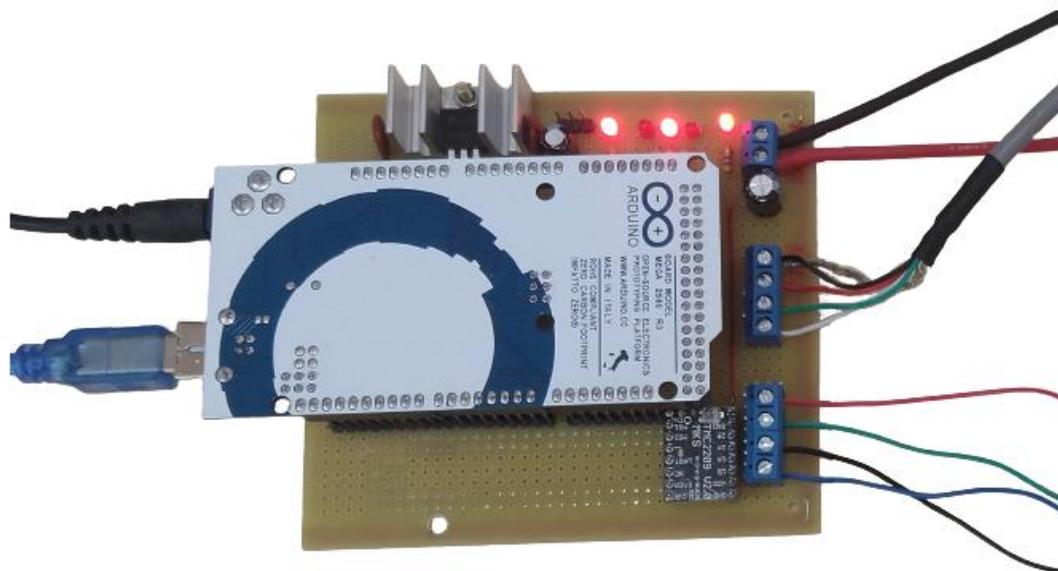
Além dos equipamentos já citados foram adicionados cinco LEDs para sinalização, dois deles para sinalizar a alimentação de 5V e 24V, e os outros três estão sendo utilizados para códigos de erro. A seguir vemos a Figura 18 e Figura 19 detalhando cada parte do esquema elétrico.

Figura 18. Esquema elétrico detalhado (1)



Fonte: Próprio autor

Figura 20. Placa de montagem



Fonte: Próprio autor

Além da placa de circuitos perfurada, foi montado também um suporte para o motor Nema e o *encoder* incremental que foram ligados por um eixo como vemos na Figura 21.

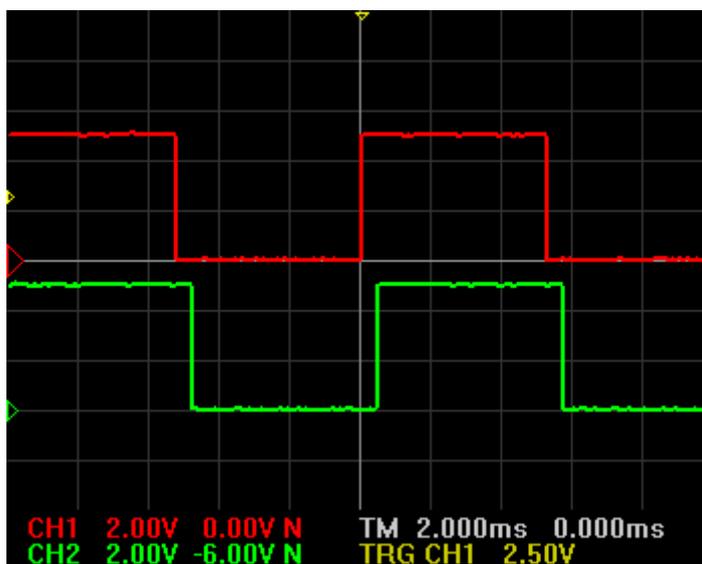
Figura 21. Motor e *encoder*

Fonte: Próprio autor

4. TESTES E RESULTADOS

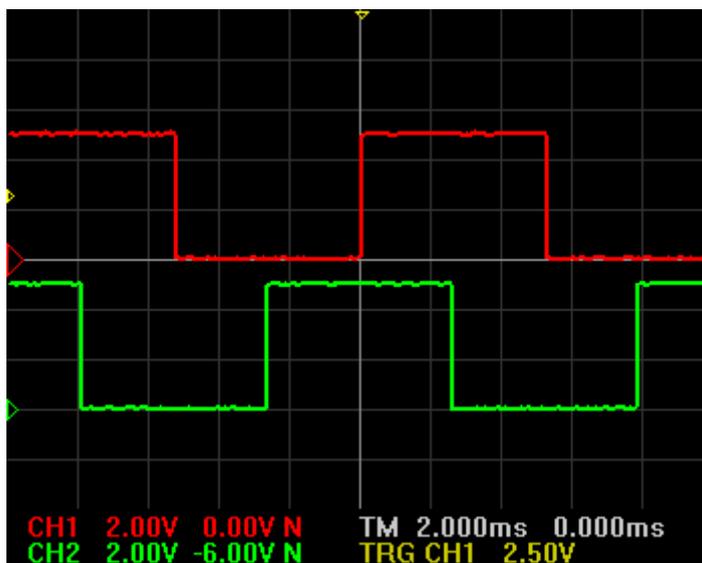
Para fins de testar a lógica empregada na programação do circuito, foi utilizado o *software* PicSimLab para simular os pulsos enviados ao drive e os sinais recebidos do *encoder*. Com isso, podemos ver na Figura 22 o sinal de pulso, destacado em vermelho, e logo abaixo a leitura do sinal A do *encoder*.

Figura 22. Sinal de pulso e *encoder* A.



Fonte: Próprio autor

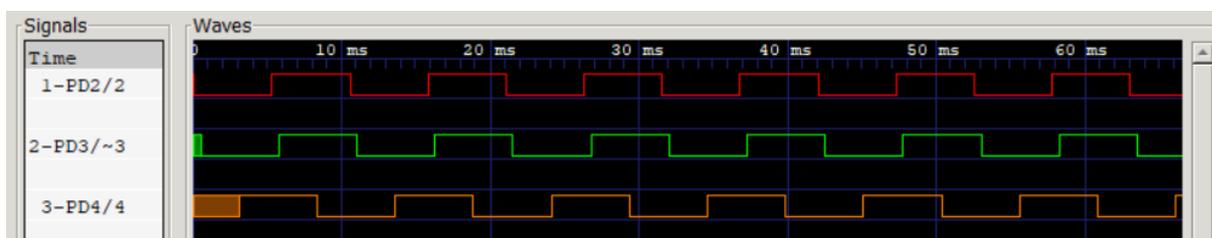
Ao analisar a Figura 22 notamos que o sinal do *encoder* A (verde) está com um formato de onda igual ao sinal do pulso (vermelho), apenas com um leve atraso de aproximadamente 0,57 segundos por conta do tempo de comando do sinal para o motor e a leitura dos movimentos pelo *encoder*. Na Figura 23 podemos ver os sinais do *encoder* A e *encoder* B.

Figura 23. Sinais do *encoder A* e *encoder B*

Fonte: Próprio autor

A Figura 23 nos apresenta os sinais A e B do *encoder*, onde notamos que o sinal B (verde) está 90° defasado do sinal A (vermelho), confirmando o funcionamento padrão do *encoder* incremental.

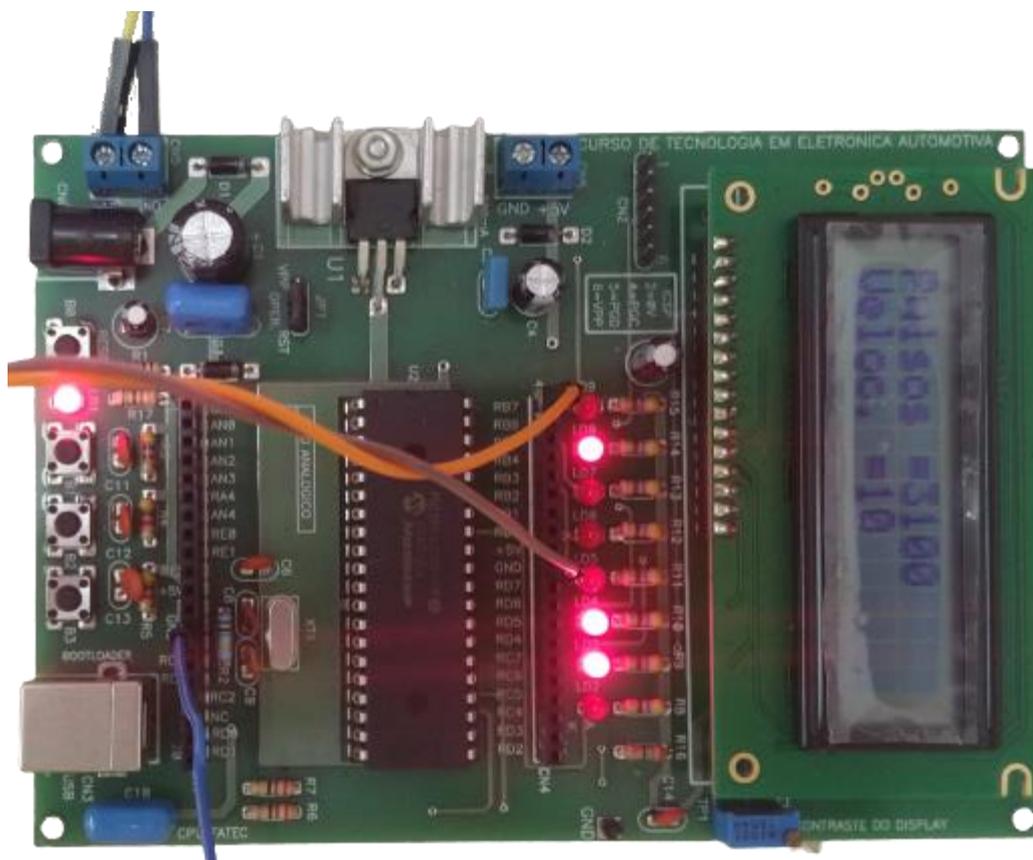
Para uma análise conjunta dos três sinais, foi utilizada uma ferramenta do *software* capaz de gravar as ondas de determinados pinos gerando um documento com as informações. Na Figura 24 temos a onda de pulso representada pela cor vermelha, a onda do *encoder A* na cor verde, e por fim a onda laranja representando o sinal do *encoder B*.

Figura 24. Sinais de pulso e do *encoder*

Fonte: Próprio autor

Para a realização dos testes foi utilizado um microcontrolador PIC 18F4550 fornecido pela Fatec para a geração de pulsos. A Figura 25 retrata a placa utilizada.

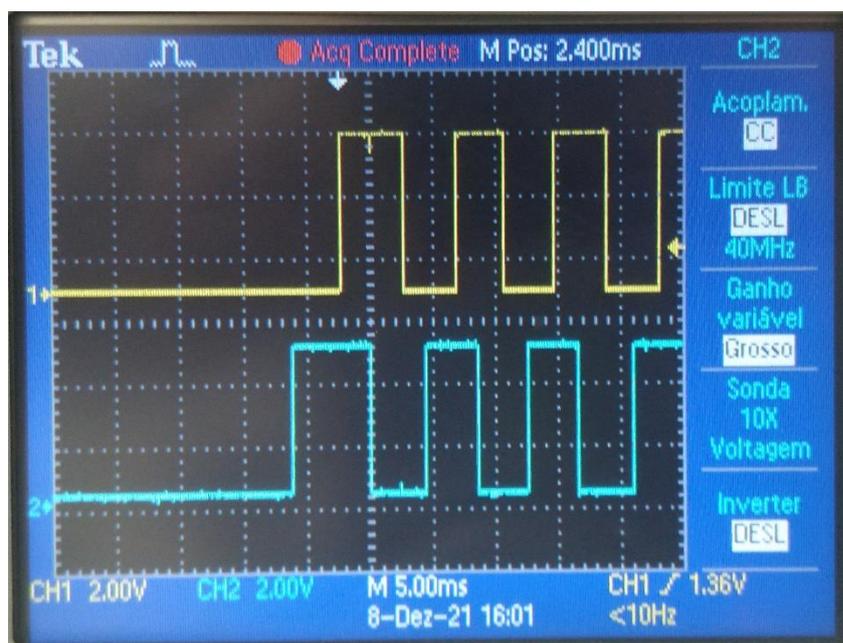
Figura 25. Microcontrolador PIC 18F4550



Fonte: Próprio autor

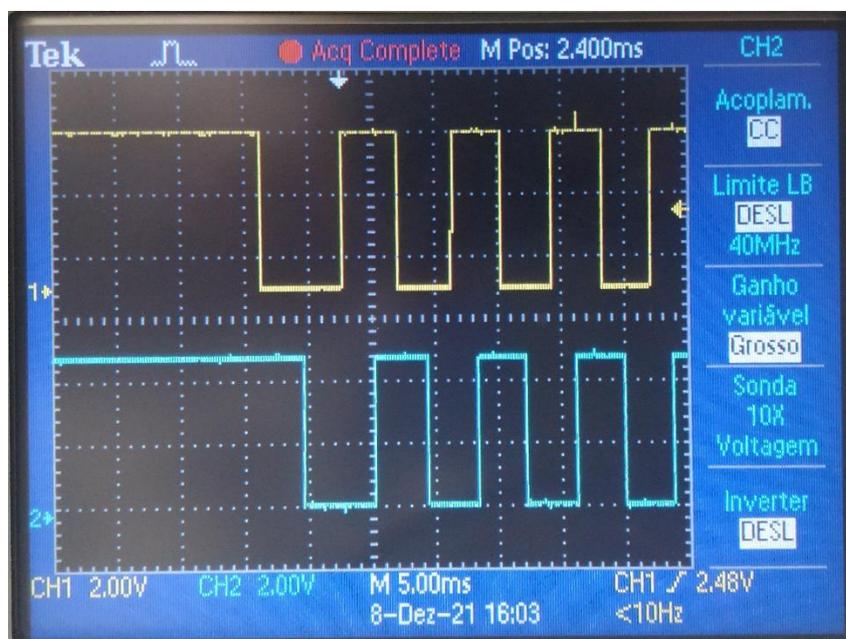
A primeira análise que podemos fazer é o sinal do *encoder* quando o motor está girando no sentido horário e anti-horário para a verificação da defasagem de 90° dos sinais A e B do *encoder*, como vemos na Figura 26 e Figura 27 sendo o sinal A o canal 1 do osciloscópio (amarelo) e o sinal B o canal 2 (azul).

Figura 26. Sentido horário



Fonte: Próprio autor

Figura 27. Sentido anti-horário



Fonte: Próprio autor

Logo depois foram analisados os pulsos enviados pelo PIC (onda amarela) e os pulsos que o Arduino enviou para o drive (onda azul) que podemos ver na Figura 28.

Figura 28. Pulsos enviados pelo PIC e pelo Arduino ao drive



Fonte: Próprio autor

Em seguida foi realizado o teste de verificação dos pulsos enviados pelo PIC, que foi ajustado em 3100 pulsos, próximo a uma volta do motor, e com os pulsos recebidos pelo Arduino e os lidos pelo *encoder*, na Tabela 1 podemos analisar os dados obtidos deste teste.

Tabela 1 - Relação de pulsos

	Arduino	Encoder		Arduino	Encoder		Arduino	Encoder
1	1615	404	6	1615	403	11	1582	396
2	1615	405	7	1583	396	12	1583	395
3	1648	412	8	1616	404	13	1615	404
4	1615	404	9	1583	396	14	1616	404
5	1615	405	10	1616	404	15	1615	407

Fonte: Próprio autor

Com isso foi possível comprovar que a resolução do *encoder* não era similar a resolução dos pulsos lidos pelo Arduino e aos enviados para o drive. O motor Nema 17 possui uma resolução de $1,8^\circ$ por pulso, o que significa que são necessários 200

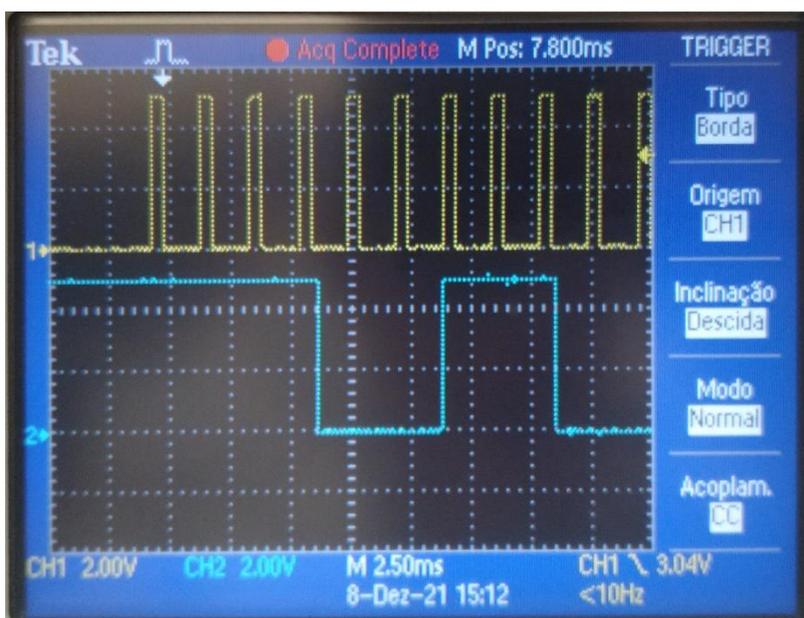
pulsos para dar uma volta completa, porém o drive TMC2209 possui a função de micropasso, sendo 8 o menor valor possível de ajuste de micropasso, com isso os 200 pulsos necessários para dar uma volta são multiplicados por 8, totalizando 1600 pulsos para uma volta completa, valor que podemos confirmar com os dados da Tabela 1. Já o *encoder* possui uma resolução de 400 pulsos por volta, com isso podemos calcular a relação de graus por pulsos em uma volta, tanto do motor quanto do *encoder*:

$$\text{Motor} = \frac{360^\circ}{(200 \cdot 8)} = \frac{360}{1600} = 0,225^\circ$$

$$\text{Encoder} = \frac{360^\circ}{400} = 0,9^\circ$$

A partir desses dados foi possível concluir que essa diferença de resolução interferia na comparação da quantidade de pulsos pela programação para a identificação de erro ou perda de pulsos. Na Figura 29 podemos visualizar melhor essa diferença de resolução analisando o sinal de pulso na saída do Arduino (onda amarela) e o sinal A do *encoder* (onda azul) através do osciloscópio.

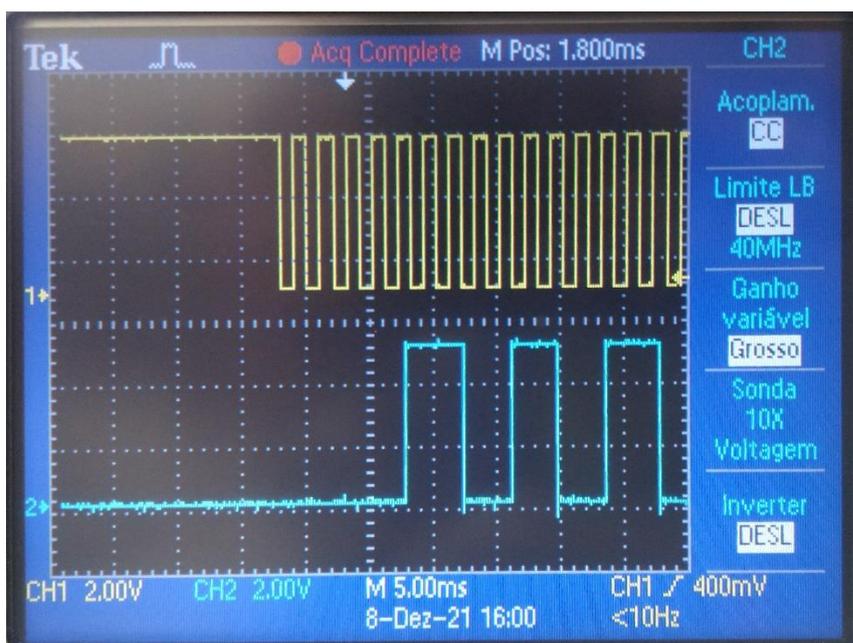
Figura 29. Resolução do motor e do *encoder*



Fonte: Próprio autor

Após analisar a diferença de resolução vista anteriormente, o próximo teste realizado foi a análise do atraso do envio do sinal do *encoder* com os pulsos recebidos pelo Arduino. Esse atraso pode ser visto na Figura 30 através do osciloscópio. A onda amarela é o sinal dos pulsos recebidos pelo Arduino e a onda azul é o sinal recebido do *encoder*.

Figura 30. Atraso do *encoder* em relação aos pulsos no Arduino



Fonte: Próprio autor

Foi a partir destes testes que a programação foi modificada para atender as características dos equipamentos utilizados. Uma das mudanças foi a multiplicação por quatro dos pulsos lidos do *encoder*, devido a diferença de resoluções, para que se aproximasse a quantidade de pulsos recebidos pelo Arduino. Esse valor foi definido a partir da média de pulsos recebidos pelo Arduino dividida pela média de pulsos que o *encoder* enviou. Podemos ver o cálculo a seguir:

$$\text{Relação} = \frac{\text{Média de pulsos do Arduino}}{\text{Média de pulsos do encoder}} = \frac{1608,8}{402,6} \cong 3,99$$

Assim como também foi necessário a utilização de uma tolerância de erro de pulsos, pois como foi visto na Tabela 1, existia uma variação da leitura no Arduino dos pulsos enviados pelo PIC, mesmo esse valor sendo constante no microcontrolador.

Para a tolerância foi utilizado o cálculo do desvio padrão dos pulsos do Arduino dividido pelo desvio padrão dos pulsos do *encoder* totalizando numa tolerância de 4 pulsos.

$$\text{Tolerância} = \frac{\text{Desvio padrão do Arduino}}{\text{Desvio padrão do encoder}} = \frac{18,276}{4,777} \cong 3,825$$

5. CONCLUSÃO

Este trabalho proporcionou a pesquisa e o desenvolvimento de um sistema de malha fechada capaz de identificar perdas de passo e possíveis falhas de um motor de passo, com o intuito de melhorar sistemas contra quedas de energia ou paradas acidentais, levando em conta o aumento e a grande utilização de motores de passo em sistemas como máquinas CNC e impressoras 3D.

Com o aumento exponencial desses equipamentos, principalmente no meio industrial, fica claro que é inviável utilizar um sistema em malha aberta, cedendo ao risco de perda de todo um trabalho por causa de uma falha de alimentação no sistema, sendo mais benéfico a utilização de um circuito de malha fechada.

Visando trabalhos futuros, algumas propostas podem ser implementadas como: a) a utilização de um sistema PID para um melhor controle dos movimentos do motor de passo; b) avaliar a dinâmica do sistema mecânico e o tempo de resposta; c) usar a medida de quadratura da onda do *encoder* dividida por quatro para aumentar a resolução, igualando-a a resolução do motor; d) implementar o projeto num sistema de CNC ou impressora 3D.

6. REFERÊNCIAS TEXTUAIS

3D, dOne. **História das Impressoras 3D**. [2020]. Disponível em: www.done3d.com.br/historia-das-impressoras-3d/. Acesso em: 12 mar 2021.

AGNIHOTRI, Nikhil. **Stepper Motor: basics, types and working**. Basics, Types and Working. 2011. Disponível em: www.engineersgarage.com/article_page/stepper-motor-basics-types-and-working/. Acesso em: 27 maio 2021.

ALMEIDA, Fernanda. **O que é Encoder? Para que serve? Como escolher? Como interfacear?** 2017. Disponível em: www.hitecnologia.com.br/blog/o-que-é-encoder-para-que-serve-como-escolher-como-interfacear/. Acesso em: 31 maio 2021.

CAPPELLATO, Frederico. **Encoders Industriais: heavy duty, incremental e absoluto**. Heavy Duty, Incremental e Absoluto. 2020. Disponível em: www.automacaoindustrial.info/encoders-industriais-heavy-duty-incremental-e-absoluto/. Acesso em: 31 maio 2021.

CNC, Cobra. **Router CNC**. Disponível em: www.cobracnc.com.br/router-cnc. Acesso em: 13 mar 2021.

CNC, Mach. **O que é Router CNC?**. Disponível em: www.machcnc.net/o-que-e-router-cnc. Acesso em: 29 maio 2021.

CORPORATION, Nmb Technologies. **Stepping Motor Engineering: driver & winding configurations**. Driver & Winding Configurations. Disponível em: nmbtc.com/resources/stepping-motor-engineering/. Acesso em: 31 maio 2021.

CRAVO, Edilson. **Motor de passo com encoder**: quando devo optar por um?. Disponível em: blog.kalatec.com.br/motor-de-passo-com-encoder-quando-devo-optar-por-um/. Acesso em: 29 maio 2021.

ELECTRONICS, Athos. **Motor de Passo – O que é e como funciona**. Disponível em: www.athoselectronics.com/motor-de-passo-como-funciona/. Acesso em: 25 maio 2021.

HAT, Red. **Introdução à Administração de Sistemas**: impressoras e impressão. Impressoras e Impressão. 2005. Disponível em: web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-pt_br-4/s1-printers-types-inkjet.html. Acesso em: 31 maio 2021.

IMPRESSÃO 3D. In: Wikipédia: a enciclopédia livre. Disponível em : pt.wikipedia.org/wiki/Impressão_3D. Acesso em: 12 mar 2021.

IMPRESSORAJATO. **Motor de passo**: saiba o que é e para que serve. Saiba o que é e para que serve. Disponível em: www.impressorajato.com.br/motor-de-passo. Acesso em: 31 maio 2021.

MOTION, Neo. **Datasheet de produto: Motor de passo**. Disponível em: www.neomotion.com.br/wp-content/uploads/2017/07/Catálogo-Datasheet-dos-motores-de-passo-R01.pdf. Acesso em: 25 maio 2021.

MOTOR, Oriental. **Tudo que você precisa saber sobre motores de passo**. Disponível em: www.orientalmotor.com.br/tecnologia/tudo-que-voce-precisa-saber-sobre-motores-de-passo.html. Acesso em: 27 maio 2021.

PRINTER, Pick 3D. **2021 List of The Best 3D Printer Brands**. 2021. Disponível em: www.pick3dprinter.com/3d-printer-brands/. Acesso em: 26 maio 2021.

RAZGRIZ, Guilherme. **Como funciona uma impressora 3D**. 2020. Disponível em: www.filipeflop.com/blog/como-funciona-uma-impressora-3d/. Acesso em: 29 maio 2021.

RIBEIRO, Lucas; MALAGUTTI, Thiago Francisco. Motores de passo: funcionamento e aplicações. **Linguagem Acadêmica**: Dossiê: Engenharias, Batatais, v. 10, n. 1, p. 115-131, jan./jun. 2020. Semestral. Disponível em: www.claretiano.edu.br/revista/language-academica/605b6ee828675c916d868d80. Acesso em: 27 maio 2021.

SILVA, Cleber Rodrigues et al. **Impressora de prototipagem rápida**: Impressora 3d. 2015. Disponível em: www.pt.slideshare.net/FranklinArisson/tcc-impressora-3d-unip-2013. Acesso em: 26 maio 2021.

SILVA, Jonathan. **Como trabalha um Router CNC**. 2015. Disponível em: maxdesign.com.br/corte-em-router-cnc/index.php/2015/04/27/como-trabalha-um-router-cnc/. Acesso em: 29 maio 2021.

SINCAK, Carlos Henrique *et al.* Projeto, construção e funcionamento de uma máquina CNC com plataforma livre - arduino. **Revista de Empreendedorismo, Inovação e Tecnologia**, [S.L.], v. 3, n. 2, p. 67-76, 23 dez. 2016. Complexo de Ensino Superior Meridional S.A.. <http://dx.doi.org/10.18256/2359-3539/reit-imed.v3n2p67-76>. Disponível em: <https://seer.imed.edu.br/index.php/revistas/article/view/1538>. Acesso em: 29 maio 2021.

SOUZA, Marco Antonio Alves de. **Implementação de sistema controlador de motor de passo em malha fechada utilizando tecnologia baseada em controlador digital de sinais**. 2007. 82 f. TCC (Graduação) - Curso de Engenharia Elétrica Com Ênfase em Eletrônica, Universidade de São Paulo, São Carlos, 2007. Disponível em:

www.tcc.sc.usp.br/tce/disponiveis/18/18042100/tce-14052010-111439/?&lang=br.

Acesso em: 13 jul. 2021.

STORE, HD. **Tipos de impressoras 3D**: conheça as mais importantes. conheça as mais importantes. Disponível em: www.blog.hdstore.com.br/tipos-de-impressoras-3d/.

Acesso em: 26 maio 2021.

TECHNOLOGIES, Wishbox. **Conheça a história e evolução das impressoras 3D**.

Disponível em: www.wishbox.net.br/blog/impressora-3d-historia/. Acesso em: 12 mar 2021.

UNESP, FEIS. **Motor de Passo**. 2013. 43 slides. Disponível em:

www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf. Acesso em: 25 maio 2021.

7. REFERÊNCIAS DAS IMAGENS

ACELERA3D. **Driver MKS TMC2209 V2.0:** Silencioso UART. [2021]. Disponível em: <https://acelera3d.com/produto/driver-tmc2209/#tab1>. Acesso em: 09 out. 2021.

ALMEIDA, Fernanda. **O que é Encoder? Para que serve? Como escolher? Como interfacear?** 2017. Disponível em: <https://www.hitecnologia.com.br/blog/o-que-é-encoder-para-que-serve-como-escolher-como-interfacear/>. Acesso em: 31 maio 2021.

ARDUINO. **Arduino Mega 2560 Rev3.** [202-?]. Disponível em: <https://store.arduino.cc/products/arduino-mega-2560-rev3>. Acesso em: 05 nov. 2021.

CORPORATION, Nmb Technologies. **Stepping Motor Engineering:** Driver & Winding Configurations. Disponível em: <https://nmbtc.com/resources/stepping-motor-engineering/>. Acesso em: 31 maio 2021.

ELECTRONICS, Athos. **Fonte chaveada:** Como funciona. [202-?]. Disponível em: <https://athoselectronics.com/fonte-chaveada/>. Acesso em: 02 nov. 2021.

ELETROGATE. **Fonte 5V 1A Bivolt.** [202-?]. Disponível em: <https://www.eletrogate.com/fonte-5v-1a>. Acesso em: 02 nov. 2021.

MOTION, Neo. **Datasheet de produto:** Motor de passo. Disponível em: www.neomotion.com.br/wp-content/uploads/2017/07/Catálogo-Datasheet-dos-motores-de-passo-R01.pdf. Acesso em: 25 maio 2021.

UNESP, FEIS. **Motor de Passo.** 2013. 43 slides. Disponível em: www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf. Acesso em: 25 maio 2021.

APÊNDICE A – CÓDIGO ARDUINO

```

#include <Arduino.h>

#define encoderA      21
#define encoderB      20
#define comandoPulsos 19
#define comandoDirecao 18
#define diag          9

#define ms1           10
#define ms2           11
#define stepPin       12
#define directionPin  13
#define enablePin     8

#define led1          67
#define led2          68
#define led3          69

volatile int microPasso = 0, pulsosComparacao = 0;
volatile int direcao = 0, leituraDirecao = 0, pulsos = 0, leituraPulsos = 0;
volatile int intPulso = 0, intDirecao = 0, intEncoder = 0; // flags para
interrupções
int valorTolerancia = 4; // valor de tolerância devido a diferença de
resoluções do encoder e motor
int maiorTolerancia = 0, menorTolerancia = 0; // variáveis para a comparação
dos pulsos com a tolerância aplicada

void setup(){

// CONFIGURAÇÃO DOS PINOS DE ENTRADA E SAÍDA //

pinMode(encoderA, INPUT);
pinMode(encoderB, INPUT);
pinMode(comandoPulsos, INPUT);
pinMode(diag, INPUT);

pinMode(ms1, OUTPUT);
pinMode(ms2, OUTPUT);
pinMode(stepPin, OUTPUT);
pinMode(directionPin, OUTPUT);
pinMode(enablePin, OUTPUT);

pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
pinMode(led3, OUTPUT);

// ATRIBUIÇÃO DAS INTERRUPÇÕES //

```

```

    attachInterrupt(digitalPinToInterrupt(comandoPulsos), interrupcaoPulsos,
RISING);
    attachInterrupt(digitalPinToInterrupt(comandoDirecao), interrupcaoDirecao,
CHANGE);
    delay(100);
    attachInterrupt(digitalPinToInterrupt(encoderA), interrupcaoEncoderA,
RISING);

    Serial.begin(9600);

    digitalWrite(enablePin, HIGH); // HIGH desativa o drive (LOW habilita)
    digitalWrite(directionPin, HIGH); // define direção de rotação inicial
    direcao = 1;

/*
DEFINIÇÃO DO MICROPASSO - DRIVE TMC2209

MS1      MS2      PASSO

GND      GND      8
VCC_IO   GND      32
GND      VCC_IO   64
VCC_IO   VCC_IO   16
*/

    digitalWrite(ms1, LOW);
    digitalWrite(ms2, LOW); // define micropasso

    microPasso = 4; // valor definido como a metade do micropasso ajustado

    digitalWrite(enablePin, LOW); // habilita o drive
}

void loop(){

    if(intPulso == 1 && intEncoder == 1){ // garante que as interrupções
foram acionadas

        intPulso = 0; // reseta Flag
        intEncoder = 0; // reseta Flag

        maiorTolerancia = pulsosComparacao + valorTolerancia; // aplica
tolerância
        menorTolerancia = pulsosComparacao - valorTolerancia; // aplica
tolerância

```

```

    if(pulsos > maiorTolerancia || pulsos < menorTolerancia){ // analisa se
o valor está dentro da tolerância
        digitalWrite(enablePin, HIGH); // se valor estiver acima ou abaixo
desabilita o drive
        Serial.println("Erro de pulsos"); // indica o erro de pulso no monitor
serial
        digitalWrite(led1, HIGH); // acende o Led de erro de pulso
    }
}
if(intDirecao == 1){ // garante que a interrupção de direção foi acionada
    if(direcao != leituraDirecao){ // verifica se possui erro de direção
        digitalWrite(enablePin, HIGH); // se a direção estiver errada,
desabilita o drive
        Serial.println("Erro no sentido de rotacao"); // indica o erro de
direção no monitor serial
        digitalWrite(led2, HIGH); // acende o Led de erro de direção
    }
}
if(digitalRead(diag) == HIGH){ // verifica se possui erro no drive
    digitalWrite(enablePin, HIGH); // se possui erro, desabilita o drive
    Serial.println("Erro no drive"); // indica o erro no drive no monitor
serial
    digitalWrite(led3, HIGH); // acende o Led de erro no drive
}
}

// INTERRUPTO DO COMANDO DE PULSOS - pino 19 //
void interrupcaoPulsos(){

    intPulso = 1; // aciona flag

    pulsos++; // incrementa a variável de contagem de pulsos
    digitalWrite(stepPin, HIGH); // realiza o pulso no pino Step do drive
    delayMicroseconds(500);
    digitalWrite(stepPin, LOW);
}

// INTERRUPTO DO COMANDO DE DIREÇÃO - pino 18 //
void interrupcaoDirecao(){

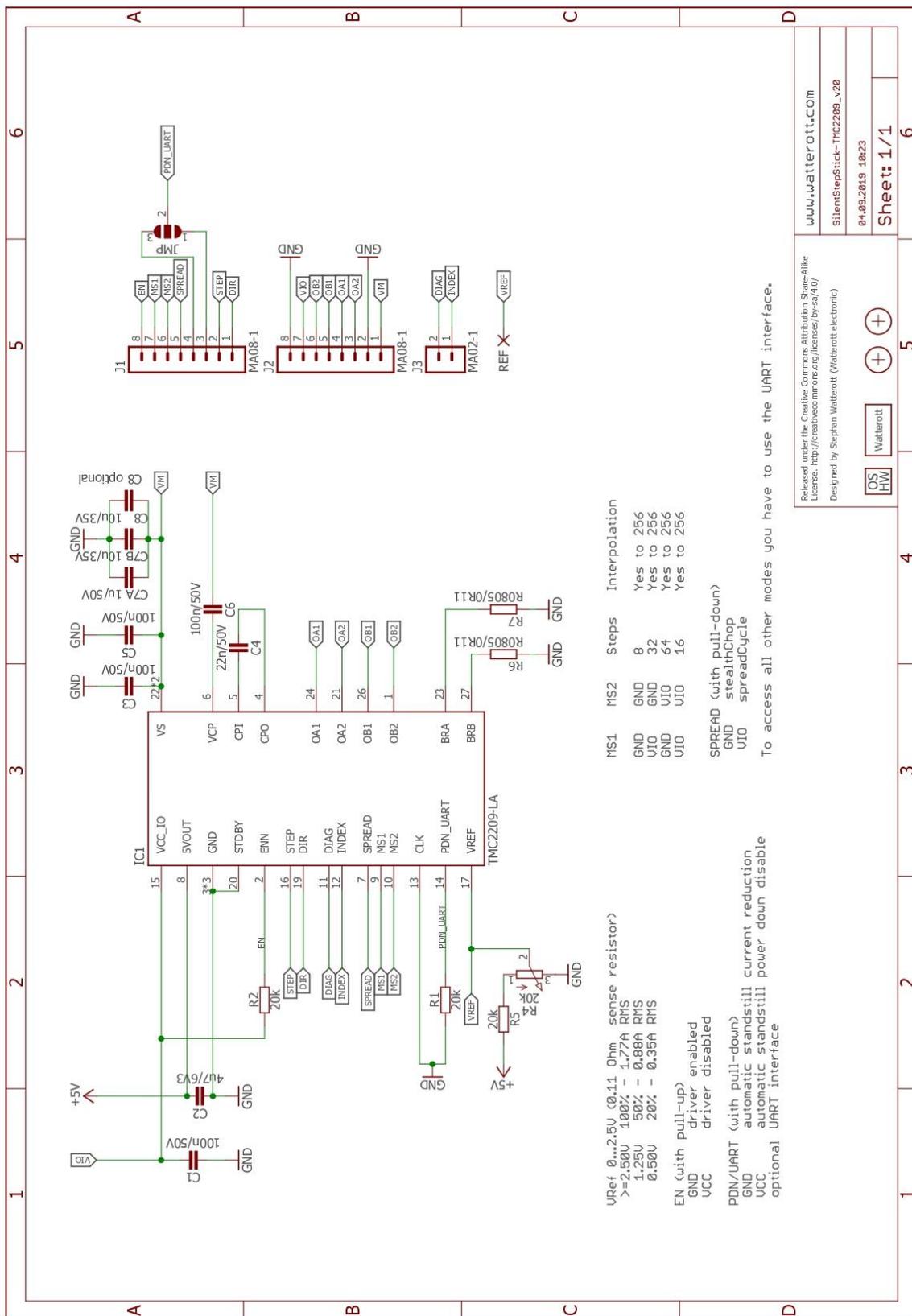
    intDirecao = 1; // aciona flag

    if(digitalRead(comandoDirecao) == HIGH){
        digitalWrite(directionPin, HIGH);
        direcao = 1; //horário
    }else{
        digitalWrite(directionPin, LOW);
        direcao = 0; //anti-horario
    }
}

```

```
}  
  
// INTERRUPTÃO DO ENCODER A - pino 21 //  
void interrupcaoEncoderA(){  
  
    intEncoder = 1;    // aciona flag  
  
    leituraPulsos++;    // incrementa a variável de leitura de pulsos  
  
    if(digitalRead(encoderB) == HIGH)leituraDirecao = 1; //horário  
    else leituraDirecao = 0; //anti-horario  
  
    pulsosComparacao = leituraPulsos*microPasso; // define a variável para  
    comparação de pulsos  
}
```

ANEXO A – CIRCUITO DRIVE TMC2209



Released under the Creative Commons Attribution Share-Alike License: <http://creativecommons.org/licenses/by-sa/4.0/>
 Designed by Stephan Watterott (WatterottElectronics)

www.watterott.com
 SilentStepStick-TMC2209_v20
 04.09.2019 18:23
 Sheet: 1/1

OS HW Watterott

Component Side View (Bottom)

