

**UNIVERSIDADE FEDERAL DE MINAS GERAIS  
ESCOLA ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA NUCLEAR  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS E TÉCNICAS NUCLEARES**

**DIEGO DE CARVALHO DRUMMOND SALVADOR**

**Projeto e desenvolvimento de sistema de posicionamento  
automatizado para simulação de fonte de radiação infinita e  
calibração de detectores de radiação**

Belo Horizonte

2021

DIEGO DE CARVALHO DRUMMOND SALVADOR

**Projeto e desenvolvimento de sistema de posicionamento  
automatizado para simulação de fonte de radiação infinita e  
calibração de detectores de radiação**

Dissertação apresentada ao Programa de Pós-Graduação em Ciências e Técnicas Nucleares da Escola de Engenharia da Universidade Federal de Minas Gerais, como parte dos requisitos para obtenção do título de Mestre em Ciências e Técnicas Nucleares.

Orientador: Prof. Dr. Arno Heeren de Oliveira

Belo Horizonte

2021

S182p

Salvador, Diego de Carvalho Drummond.

Projeto e desenvolvimento de sistema de posicionamento automatizado para simulação de fonte de radiação infinita e calibração de detectores de radiação [recurso eletrônico] / Diego de Carvalho Drummond Salvador. - 2021.

1 recurso online (73 f. : il., color.) : pdf.

Orientador: Arno Heeren de Oliveira.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndice: f. 69 -73.

Bibliografia: f. 67-68.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia nuclear - Teses. 2. Detectores de radiação nuclear – Teses. 3. Monte Carlo, Método de – Teses. 4. Prototipagem rápida - Teses. I. Oliveira, Arno Heeren de. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.039(043)



**UNIVERSIDADE FEDERAL DE MINAS**

**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS E TÉCNICAS**



## FOLHA DE APROVAÇÃO

**Projeto e desenvolvimento de sistema de posicionamento automatizado para simulação de fonte de radiação infinita e calibração de detectores de radiação**

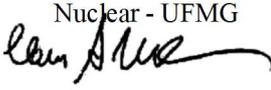
### **DIEGO DE CARVALHO DRUMMOND SALVADOR**

Dissertação submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em CIÊNCIAS E TÉCNICAS NUCLEARES, como requisito parcial para obtenção do grau de Mestre em CIÊNCIAS E TÉCNICAS NUCLEARES, área de concentração CIÊNCIAS DAS RADIAÇÕES.

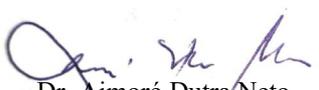
Aprovada em 24 de novembro de 2021, pela banca constituída pelos membros:

  
Prof. Arno Heeren de Oliveira - Orientador  
Departamento de Engenharia Nuclear -

  
UFMG  
Prof. Carlos Eduardo Velasquez  
Cabrera Departamento de Engenharia  
Nuclear - UFMG

  
Prof. Clarysson Alberto Mello da Silva  
Departamento de Engenharia Nuclear -  
UFMG

  
Dr. Thiago César de Oliveira  
Universidade Federal de Minas

  
Dr. Aimoré Dutra Neto  
CDTN  
Gerais

Belo Horizonte, 24 de novembro de 2021.

## **AGRADECIMENTOS**

Ao Prof. Arno Heeren, por compartilhar toda sua experiência e conhecimento, além de estar sempre disposto a contribuir para meu desenvolvimento acadêmico.

Ao meu amigo Cleber Vargas, que me ajudou imensamente em várias etapas deste projeto ao compartilhar seus anos de experiência como projetista e engenheiro mecânico para que o protótipo funcionasse adequadamente.

Ao Dr. Aimoré Dutra Neto, que deu as ideias iniciais para o projeto e se mostrou sempre disposto a ajudar a solucionar os pequenos problemas que apareceram pelo caminho.

À CAPES/CNPq pela concessão de bolsa e apoio financeiro à realização desse trabalho.

A todos os professores, funcionários e colegas do CCTN, que trabalharam para que eu tivesse todo o suporte necessário para meu aprendizado e desenvolvimento.

## RESUMO

A simulação computacional com o método Monte Carlo vem sendo usada por pesquisadores em diversos trabalhos relacionados à interação da radiação com a matéria. A fim de validar os resultados destas simulações, é necessário obter dados experimentais reais para que possam ser comparados com os resultados da simulação. Existe, porém, uma dificuldade em obter dados experimentais em laboratórios para casos em que a fonte de radiação tem dimensões maiores do que as fontes de laboratório, que são normalmente bem pequenas. O objetivo deste trabalho foi projetar, construir, programar e testar um protótipo de um equipamento que faz deslocamentos automatizados muito precisos para mudar a distância linear e o ângulo entre um conjunto de fonte e detector, com intuito de poder simular experimentalmente uma fonte de grandes dimensões a partir de uma fonte puntual de laboratório. Com diferentes medições da taxa de dose em diferentes posições, é possível interpretar a somatória dessas medições como o valor que seria lido pelo detector, caso a fonte de radiação tivesse dimensões maiores. O protótipo foi projetado em CAD, tendo diversos componentes impressos com utilização de impressora 3D. Para a operação do protótipo, utilizou-se motores de passo controlados por placas Arduino. O microcontrolador do Arduino foi programado utilizando-se linguagem de programação baseada em C/C++. Devido à precisão do deslocamento, o protótipo também pode ser usado para calibração de detectores de radiação. Para tal, é apenas necessário utilizar a função de deslocamento linear para posicionar precisamente o conjunto fonte e detector. Com uma fonte de laboratório conhecida, é possível ajustar o fator de calibração do detector para que a leitura esteja de acordo com o esperado para aquela fonte na distância determinada. O protótipo foi testado para garantir uma alta precisão no posicionamento. Para o deslocamento linear obteve-se precisão de 1 milímetro nos deslocamentos, enquanto no deslocamento angular obteve-se precisão de  $0,4^\circ$ . O protótipo foi usado para obtenção de dados experimentais com um conjunto de fonte e detector, os quais poderão já ser usados em outros trabalhos para validar resultados de simulações computacionais utilizando o método Monte Carlo.

**Palavras-chave:** Arduino. Monte Carlo. Detector de radiação. Calibração de detectores de radiação. Prototipagem.

## ABSTRACT

Computational simulation with the Monte Carlo method has been used by researchers in several studies related to the interaction of radiation with matter. In order to validate the results of these simulations, it is necessary to obtain real experimental data so that they can be compared with the simulation results. There is, however, a difficulty in obtaining experimental data in laboratories for cases where the radiation source has larger dimensions than laboratory sources, which are usually quite small. The objective of this work was to design, build, program and test a prototype of an equipment that makes very precise automated displacements to change the linear distance and angle between a set of source and detector, in order to be able to experimentally simulate a large source by using a very small laboratory source. With different dose rate measurements at different positions, it is possible to interpret the sum of these measurements as the value that would be read by the detector if the radiation source had larger dimensions. The prototype was designed using a CAD, having several components printed using a 3D printer. For the operation of the prototype, stepper motors controlled by Arduino boards were used. The Arduino microcontroller was programmed using a programming language based on C/C++. Due to the accuracy of the displacement, the prototype can also be used for calibration of radiation detectors. To do this, it is only necessary to use the linear displacement function to precisely position the set of source and detector. With a known laboratory source, it is possible to adjust the detector calibration factor so that the reading is as expected for that source at the given distance. The prototype has been tested to ensure high positioning accuracy. For the linear displacement, precision of 1 mm was obtained in the displacements, while in the angular displacement, precision of  $0.4^\circ$  was obtained. The prototype was used to obtain experimental data with a set of source and detector, which can already be used in other studies to validate results of computational simulations using the Monte Carlo method.

**Keywords:** Arduino. Monte Carlo. Radiation detector. Calibration of radiation detectors. Prototyping.

## LISTA DE FIGURAS

Figura 1 - Representação típica de pulsos de corrente em um detector .....	16
Figura 2 - Esquema de um detector a gás típico.....	17
Figura 3 - Gráfico Tensão x Número de íons coletados por unidade de tempo .....	17
Figura 4 - Esquema de um detector de cintilação.....	20
Figura 5 - Imagem de um microcontrolador.....	23
Figura 6 - Arduino UNO .....	25
Figura 7 - Arduino Mega 2560.....	25
Figura 8 - Exemplo de programação do Arduino utilizando o IDE .....	26
Figura 9 - Exemplo de motor de passo bipolar.....	27
Figura 10 - Interface do driver A4988.....	29
Figura 11 - Indicação do trimpot do driver A4988.....	30
Figura 12 - Impressora 3D.....	32
Figura 13 - Bico injetor de uma impressora 3D durante processo de impressão .....	32
Figura 14 - Esquema de fonte puntual emitindo feixe de radiação com ângulo $0^\circ$ no detector .....	35
Figura 15 - Esquema de fonte infinita emitindo feixes de radiação com diferentes ângulos e distâncias no detector.....	36
Figura 16 - Esquema de deslocamento angular com conseqüente deslocamento linear .....	37
Figura 17 - Vista isométrica do sistema de deslocamento do detector.....	38
Figura 18 - Vista lateral do sistema de deslocamento do detector .....	38
Figura 19 - Vista planta do sistema de deslocamento do detector.....	39
Figura 20 - Vista isométrica do sistema em posição de ângulo $0^\circ$ .....	39
Figura 21 - Vista isométrica do sistema em posição de ângulo $-45^\circ$ .....	40
Figura 22 - Vista isométrica do sistema com imagens sobrepostas em 2 diferentes posições.	40
Figura 23 - Detalhe do motor do deslocamento linear com o suporte do motor (esq.) e sem o suporte do motor (dir.).....	41
Figura 24 - Detalhe do motor do deslocamento angular (rotação) com o suporte do motor (acima) e sem o suporte do motor (abaixo) .....	41
Figura 25 - Detalhe do sistema de molas e esticador da correia dentada .....	42
Figura 26 - Detalhe da guia linear com patins de deslocamento .....	43

Figura 27 - Detalhe da polia dentada e correia dentada .....	44
Figura 28 - Esteira porta cabos .....	44
Figura 29 - Motor de passo bipolar NEMA 17HS4401 .....	45
Figura 30 - Desenho peça 'Ajuste Altura Detector' .....	46
Figura 31 - Desenho peça 'Ajuste Altura Fonte' .....	46
Figura 32 - Desenho peça 'Esticador' .....	47
Figura 33 - Desenho peça 'Placa Molas' .....	47
Figura 34 - Desenho peça 'Suporte Carrinho' .....	47
Figura 35 - Desenho peça ' Suporte Correia Angular 1' .....	47
Figura 36 - Desenho peça 'Suporte Correia Angular 2' .....	48
Figura 37 - Desenho peça 'Suporte Correia Linear 1' .....	48
Figura 38 - Desenho peça 'Suporte Correia Linear 2' .....	48
Figura 39 - Desenho peça 'Suporte Eixo Fonte' .....	48
Figura 40 - Desenho peça 'Suporte Motor Angular' .....	48
Figura 41 - Desenho peça 'Suporte Motor Linear' .....	48
Figura 42 - Desenho peça 'Suporte Perfil L' .....	49
Figura 43 - Esquema de ligação Arduino/Drivers A4988/Motores de passo .....	50
Figura 44 - Esquema de ligação com descrição dos pinos do Arduino e Drivers A4988 .....	51
Figura 45 - Peças impressas.....	52
Figura 46 - Peças impressas.....	52
Figura 47 - Peças impressas.....	52
Figura 48 - Peças impressas.....	53
Figura 49 - Peças impressas.....	53
Figura 50 - Peças impressas.....	53
Figura 51 - Peças impressas.....	54
Figura 52 - Foto do sistema montado (isométrica).....	54
Figura 53 - Foto sistema montado (planta).....	55
Figura 54 - Foto do detalhe da pista de teflon .....	55
Figura 55 - Foto do sistema montado (vista lateral direita).....	56
Figura 56 - Foto do sistema montado (vista lateral esquerda).....	56
Figura 57 - Foto do detalhe do subsistema do motor angular, eixo de giro e suporte da fonte	57

Figura 58 - Foto do subsistema do motor linear e carro de suporte do detector .....	57
Figura 59 - Foto da ligação elétrica Arduino/Drivers A4988/Motores de passo .....	58

## LISTA DE TABELAS

Tabela 1 - Modos de passo do driver A4988.....	29
Tabela 2 - Pulsos por revolução em cada modo de passo do driver A4988 .....	30
Tabela 3 - Componentes comprados .....	43
Tabela 4 - Componentes eletrônicos .....	45
Tabela 5 - Peças fabricadas em impressora 3D .....	46
Tabela 6 - Medições para ângulo 15° .....	60
Tabela 7 - Medições para ângulo -15° .....	60
Tabela 8 - Medições para ângulo 30° .....	61
Tabela 9 - Medições para ângulo -30° .....	61
Tabela 10 - Medições para ângulo 40° .....	61
Tabela 11 - Medições para ângulo -40° .....	62
Tabela 12 - Medições para deslocamento de 100 mm.....	62
Tabela 13 - Medições para deslocamento de 155 mm.....	63
Tabela 14 - Medições para deslocamento de 259 mm.....	63
Tabela 15 - Medições para deslocamento de 388 mm.....	63
Tabela 16 - Contagem de pulsos do detector em diferentes posições do sistema .....	64

## LISTA DE ABREVIATURAS E SIGLAS

ABS *Acrylonitrile Butadiene Styrene*

CAD *Computer Aided Design*

CNC *Computer Numerical Control*

CNEN *Comissão Nacional de Energia Nuclear*

CPU *Central Processing Unit*

DC *Direct Current*

GM *Geiger-Muller*

I/O *Input/Output*

IAEA *International Atomic Energy Agency*

IDE *Integrated Development Environment*

MCNP *Monte Carlo N-Particle Transport Code*

PPR *Passos Por Revolução*

PROM *Programmable Read Only Memory*

RAM *Random Access Memory*

USB *Universal Serial BUS*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>13</b>
1.1	OBJETIVOS .....	14
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA .....</b>	<b>15</b>
2.1	DETECTORES DE RADIAÇÃO .....	15
2.1.1	Detectores a gás .....	16
2.1.1.1	<i>Câmaras de Ionização .....</i>	<i>18</i>
2.1.1.2	<i>Detectors proporcionais .....</i>	<i>18</i>
2.1.1.3	<i>Detectors Geiger-Muller (GM).....</i>	<i>18</i>
2.1.2	Detectors de cintilação .....	19
2.1.3	Detectors semicondutores .....	20
2.2	CALIBRAÇÃO DE DETECTORES DE RADIAÇÃO .....	21
2.3	AUTOMAÇÃO E CONTROLE.....	22
2.3.1	Microcontroladores.....	22
2.3.2	Arduino .....	24
2.3.2.1	<i>Programação do Arduino.....</i>	<i>25</i>
2.3.3	Motores de passo .....	26
2.3.4	Driver do motor .....	28
2.3.4.1	<i>Driver A4988 .....</i>	<i>28</i>
2.3.4.1.1	<i>Modos de passo do driver A4988 .....</i>	<i>29</i>
2.3.4.1.2	<i>Ajuste de corrente do driver A4988.....</i>	<i>30</i>
2.4	IMPRESSÃO 3D .....	31
<b>3</b>	<b>METODOLOGIA E MATERIAIS.....</b>	<b>34</b>
3.1	PRINCÍPIO DE FUNCIONAMENTO DO SISTEMA.....	34
3.1.1	Utilizando o sistema como auxiliador de calibração de detector de radiação .....	34

3.1.2	Utilizando o sistema para simular fonte infinita a partir de fonte puntual .....	35
3.2	DESENHO E PROJETO DO SISTEMA .....	37
3.3	LISTA DE MATERIAIS E COMPONENTES DO SISTEMA.....	42
3.4	ESQUEMA DE LIGAÇÃO DOS MOTORES DE PASSO.....	50
3.5	PEÇAS FABRICADAS NA IMPRESSORA 3D.....	52
3.6	MONTAGEM DO SISTEMA.....	54
3.7	ELETRÔNICA E PROGRAMAÇÃO.....	57
<b>4</b>	<b>RESULTADOS .....</b>	<b>60</b>
4.1	TESTE DE POSICIONAMENTO DO SISTEMA .....	60
4.1.1	Testes de posicionamento angular .....	60
4.1.2	Testes de posicionamento linear.....	62
4.1.3	Medições com fonte Cobalto e detector Geiger-Muller .....	64
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>66</b>
	<b>REFERÊNCIAS.....</b>	<b>67</b>
	<b>APÊNDICE A – Código de programação do Arduino.....</b>	<b>69</b>

## 1 INTRODUÇÃO

No fim do século XIX, com a descoberta dos raios-X, da radioatividade e do rádio (por Roentgen, Becquerel e os Curie, respectivamente), cada vez mais os estudos e aplicações da radiação ionizante se ampliaram. Em poucos anos, estudos mostraram um aumento da probabilidade de ocorrência de efeitos biológicos danosos, devido à utilização das radiações ionizantes. Surgiu-se, então, a necessidade de determinar esses efeitos em função das taxas de dose devido aos diversos tipos de radiação, a fim de tomar medidas preventivas a saúde dos trabalhadores expostos à radiação por períodos prolongados e frequentes. (DIAS, 1996)

Os detectores de radiação começaram a ser largamente utilizados para mensurar a radiação ionizante há quase um século atrás. Diversos tipos de detectores de radiação foram desenvolvidos, juntamente com a eletrônica associada aos mesmos, para diferentes situações de exposição. Basicamente, um detector de radiação funciona através da leitura do efeito produzido em um gás (ou outro material) quando uma partícula ou uma onda eletromagnética, emitida por uma fonte de radiação, interage com o volume sensível do detector. (MANFRINI, 2007)

Para se ter uma leitura correta de um sistema de detecção, é necessário que o sistema esteja devidamente calibrado. A IAEA (*International Atomic Energy Agency*) recomenda que a calibração seja feita a cada 12-14 meses (IAEA – INTERNATIONAL ATOMIC ENERGY AGENCY, 2000), enquanto a CNEN (Comissão Nacional de Energia Nuclear) exige uma calibração a cada 24 meses. (CNEN – COMISSÃO NACIONAL DE ENERGIA NUCLEAR, NORMA, 2013) Antes mesmo do primeiro uso de um detector, é necessária uma calibração precisa. Muitas vezes o próprio fabricante do detector não faz essa calibração inicial. Os motivos podem variar, desde a falta de instalações apropriadas para tal calibração até a falta de tempo hábil para calibrar todos os instrumentos. Quando se adquire um detector, é comum que o comprador acredite que o fabricante fez a devida calibração do instrumento e acabe fazendo leituras errôneas. Além disso, pequenas imprecisões na calibração podem acarretar grandes erros de leituras de dose nos detectores. (IAEA – INTERNATIONAL ATOMIC ENERGY AGENCY, 2000)

A calibração destes instrumentos de medição de radiação ionizante é realizada utilizando-se fontes radioativas pontuais de referência fornecidas por laboratórios primários,

tais como  $^{137}\text{Cs}$  e  $^{60}\text{Co}$  (emissão  $\gamma$ ),  $^{90}\text{Sr}+^{90}\text{Y}$  e  $^{204}\text{Tl}$  (emissão  $\beta$ ) e Am-Be (emissão de nêutrons). A fabricação de fontes radioativas gama, beta e nêutrons de grandes diâmetros e homogêneas é difícil e ainda, estas fontes de grandes dimensões apresentam riscos importantes de irradiação e de manipulação. Os instrumentos utilizados para medir radiação ionizante são projetados para medir taxas de dose que variam de alguns  $\mu\text{Sv} \cdot \text{h}^{-1}$  a  $\text{Sv} \cdot \text{h}^{-1}$ , em operação normal e de algumas centenas de  $\text{Sv} \cdot \text{h}^{-1}$  em situação acidental. (BAKSHI, 2013)

Hoje, também utilizam-se simulações computacionais de radiação, principalmente com o método Monte Carlo e o código MCNP. Estas simulações são feitas com diferentes objetivos, podendo até serem usadas para realizar calibração de detectores. É necessário, porém, que a simulação computacional tenha seus resultados validados ao serem comparados com dados experimentais. Essa validação pode ser difícil de ser realizada em laboratórios, já que as fontes laboratoriais são pequenas, enquanto muitas das simulações computacionais buscam simular fontes de maiores dimensões. (ALVARENGA, 2017)

## 1.1 OBJETIVOS

O trabalho tem como objetivo desenvolver um protótipo de um sistema automatizado de posicionamento que trabalha com um conjunto de fonte e detector. O sistema fará, de maneira precisa e automatizada, o deslocamento de um detector em relação a uma fonte de radiação. Esse posicionamento permitirá que se simule uma fonte “infinita” a partir de uma fonte puntual. Também será possível utilizar o sistema para fazer calibrações em detectores de radiação em diversas geometrias fonte-detector.

Portanto, as etapas do trabalho ficam assim organizadas:

- Desenhar e projetar o sistema em um CAD,
- Imprimir em impressora 3D algumas das peças do sistema,
- Realizar montagem do sistema conforme o projeto,
- Escrever o código de programação da placa Arduino, que será usada para controlar o sistema,
- Testar a operação do sistema, com intuito de garantir a precisão dos deslocamentos,
- Realizar medições utilizando um conjunto de fonte e detector em diferentes posicionamentos, para simular experimentalmente uma fonte de radiação “infinita” a partir de uma fonte puntual de laboratório.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 DETECTORES DE RADIAÇÃO

Como a radiação ionizante é imperceptível ao ser humano, se faz necessária a utilização de instrumentos que consigam indicar a presença da radiação. Quando uma fonte emite radiação sobre um outro corpo, ocorre uma interação da radiação com a matéria. (EQUIPCO SERVICES)

Quando inserido em um meio onde haja um campo de radiação, um detector de radiação consegue indicar a presença sua presença. Um detector de radiação pode utilizar de diferentes processos para medir ou indicar características das radiações. Os processos mais comuns são a geração de cargas elétricas, geração de luz, sensibilização de películas fotográficas, criação de traços no material, geração de calor e alterações da dinâmica de processos químicos. Em geral, os detectores de radiação são constituídos de um material sensível à radiação ionizante e de um sistema que associa esses efeitos da radiação a uma grandeza de medição. (KNOLL, 2010)

Os detectores de radiação precisam apresentar algumas características nas suas sequências de medição. São elas:

- Repetitividade, grau de concordância dos resultados obtidos sob as mesmas condições de medição;
- Reprodutibilidade, grau de concordância dos resultados obtidos em diferentes condições de medição;
- Estabilidade, aptidão do instrumento conservar constantes suas características de medição ao longo do tempo;
- Exatidão, grau de concordância dos resultados com o “valor verdadeiro”;
- Precisão, grau de concordância dos resultados entre si, normalmente expresso pelo desvio padrão em relação à média;
- Sensibilidade, razão entre a variação da resposta de um instrumento e a correspondente variação do estímulo;
- Eficiência, capacidade de converter em sinais de medição os estímulos recebidos. (KNOLL, 2010)

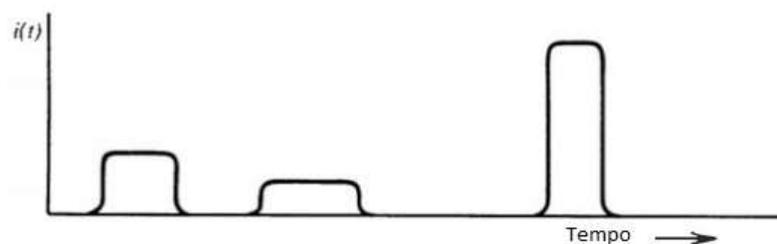
A seguir, alguns dos tipos mais comuns de detectores serão descritos.

### 2.1.1 Detectores a gás

Os detectores a gás utilizam a ionização resultante da radiação que atravessa um gás. Esses detectores são compostos de dois eletrodos, sobre os quais é aplicada uma tensão elétrica. O espaço entre os eletrodos é preenchido com o gás. Quando a radiação interage com o gás, o resultado dessa interação, no volume sensível de um detector, pode resultar no aparecimento de um par de íons, se a energia da radiação incidente for maior que a energia de ligação do meio. A carga elétrica resultante dessa interação migra para o eletrodo central do detector, quando se aplica uma diferença de potencial entre o anodo e o catodo, criando assim um campo elétrico. O sinal elétrico gerado (corrente ou pulso) é tratado pela eletrônica associada ao detector podendo fornecer os valores de dose ou taxa de dose correspondentes à radiação incidente. Todo o processo de interação, geração e coleta da carga tem uma duração muitíssimo pequena, na casa dos nanosegundos. Quando essa carga é coletada pelo circuito elétrico do detector, um pequeno aumento na corrente elétrica do circuito pode ser notado, também por um curtíssimo período de tempo. (KNOLL, 2010)

A Figura 1 mostra a distribuição de pulsos elétricos gerados no volume sensível de um detector. A magnitude e duração de cada pulso pode variar com o tipo de interação. Devido ao fato de a radioatividade ser um fenômeno aleatório, os intervalos de tempo entre os pulsos não são constantes. (KNOLL, 2010)

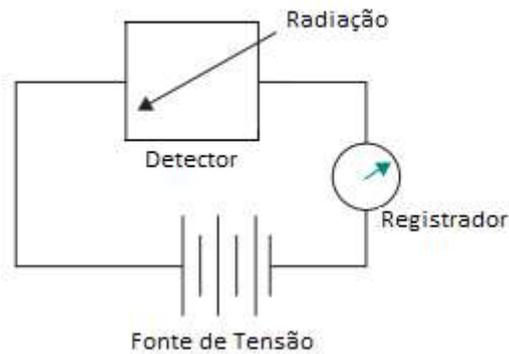
Figura 1 - Representação típica de pulsos de corrente em um detector



Fonte: "Radiation Detection and Measurement" 4th Ed., G. KNOLL

A Figura 2 mostra esquematicamente o sistema de um detector a gás.

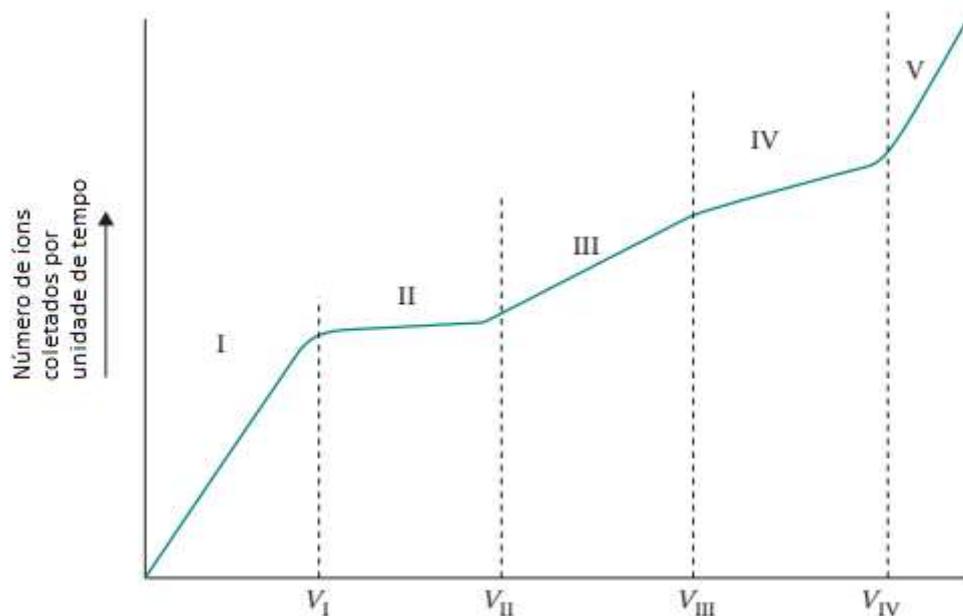
Figura 2 - Esquema de um detector a gás típico



Fonte: “Measurement & Detection of Radiation”, 4th Ed., N. TSOULFANIDIS

Existe uma relação entre a tensão aplicada ao circuito elétrico e o número de íons coletados por unidade de tempo. O gráfico na Figura 3 mostra essa relação, dividindo-se em 5 regiões que são explicadas a seguir. (CNEN – COMISSÃO NACIONAL DE ENERGIA NUCLEAR, NORMA, 2013)

Figura 3 - Gráfico Tensão x Número de íons coletados por unidade de tempo



Fonte: “Measurement & Detection of Radiation”, 4th Ed., N. TSOULFANIDIS

- Região I: A tensão não é alta o suficiente para coletar os pares e eles se recombinam.

- Região II: A carga coletada fica constante, independentemente do aumento da tensão nessa região. A recombinação é zero. É chamada de região de ionização.
- Região III: Os íons produzidos são acelerados e ionizam outras moléculas do gás. A quantidade de íons coletados aumenta proporcionalmente ao aumento da tensão. É chamada de região proporcional.
- Região IV: O campo elétrico é tão forte que um único par de elétron-íon é capaz de gerar um efeito de avalanche, criando vários outros pares, independente da energia ou tipo de radiação. É chamada de região Geiger-Muller (GM).
- Região V: Nessa região, um único evento ionizante inicia uma descarga contínua no gás. O detector perde sua função nessa região, portanto, tensões nessa região não são utilizadas. (CNEN – COMISSÃO NACIONAL DE ENERGIA NUCLEAR, NORMA, 2013)

Detectores a gás recebem os seus nomes de acordo com a região em que operam. A seguir, são discutidos os tipos de detectores a gás.

#### *2.1.1.1 Câmaras de Ionização*

As câmaras de ionização operam na região 2. Não há multiplicação de carga, pois os íons gerados pela radiação não geram outros íons. O sinal é proporcional a energia da partícula dissipada no detector. Logo, a energia da partícula é mensurável. É aplicável a partículas alfa, beta e fótons. A tensão utilizada é menor do que 1000V. (KNOLL, 2010)

#### *2.1.1.2 Detectores proporcionais*

Esses detectores são utilizados na região 3. Nela, há multiplicação de cargas, já que um par elétron-íon tem energia para gerar outros pares. O sinal é proporcional à energia depositada no detector, portanto a energia da partícula é também mensurável. Esse tipo de detector pode ser utilizado para qualquer tipo de partícula carregada. A tensão varia entre 800 e 2000V. (KNOLL, 2010)

#### *2.1.1.3 Detectores Geiger-Muller (GM)*

Operam na região 4. São detectores de simples operação. O sinal é bastante forte e eles podem ser usados para qualquer tipo de radiação ionizante. A desvantagem, porém, é que o sinal deles não depende do tipo ou energia da partícula. Isso significa que ele apenas fornece informação sobre o número de partículas, mas não identifica o tipo de radiação incidente ou sua energia. A tensão utilizada varia de 500 a 2000V. (KNOLL, 2010)

### 2.1.2 Detectores de cintilação

Alguns materiais emitem uma luz ou faísca como resultado de uma interação de uma radiação incidente. Os detectores de cintilação usam materiais que cintilam quando expostos à radiação. A quantidade de luz produzida como resultado é muito pequena e precisa ser ampliada através de tubos fotomultiplicadores. Esses detectores são capazes de medir altas taxas de contagem e são os mais apropriados para raios gama, sendo também usados para partículas alfa e beta. (INMETRO – INSTITUTO NACIONAL DE METROLOGIA, QUALIDADE E TECNOLOGIA, 2012)

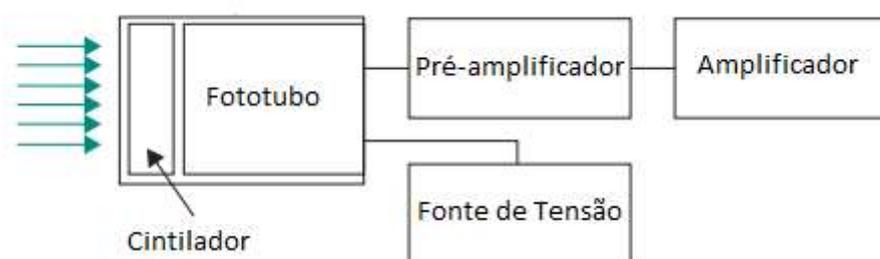
O processo de cintilação continua sendo um dos mais úteis métodos disponíveis para a detecção e espectroscopia de uma ampla variedade de radiações. O material ideal de cintilação deve possuir as seguintes propriedades:

- Deve converter a energia cinética das partículas carregadas em luz detectável com uma alta eficiência de cintilação.
- Esta conversão deve ser linear- o rendimento da luz deve ser proporcional ao depositado energia sobre o mais amplo alcance possível.
- O meio deve ser transparente ao comprimento de onda de sua própria emissão para uma boa luz coleção.
- O tempo de decomposição da luminescência induzida deve ser curto para que pulsos de sinal rápido possam ser gerado.
- O material deve ser de boa qualidade óptica e sujeito à fabricação em tamanhos grandes o suficiente para ser de interesse como um detector prático.
- Seu índice de refração deve ser próximo ao de vidro ( $n \approx 1.5$ ) para permitir o acoplamento eficiente de a luz de cintilação para um tubo fotomultiplier ou outro sensor de luz.

Nenhum material atende simultaneamente a todos esses critérios, e a escolha de um cintilante particular é sempre um equilíbrio entre esses e outros fatores. Para ser um bom cintilador, um material deve converter a fração maior possível da energia de radiação incidente para provocar fluorescência, minimizando as geralmente indesejáveis contribuições por fosforescência e fluorescência retardada. (KNOLL, 2010)

A Figura 4 mostra o esquema de um detector de cintilação.

Figura 4 - Esquema de um detector de cintilação



Fonte: “Measurement & Detection of Radiation”, 4th Ed., N. TSOULFANIDIS

### 2.1.3 Detectores semicondutores

São detectores que operam de forma semelhante às câmaras de ionização, mas com material em estado sólido. Os pares de elétron-buraco são criados ao longo do caminho tomado pela partícula carregada através do detector. O par de elétron-buraco é análogo ao par de íons criado em detectores a gás. Seu movimento em um campo elétrico aplicado gera o sinal elétrico básico do detector. (KNOLL, 2010) Os materiais mais comumente usados são o silício e o germânio. A maior vantagem em relação aos outros detectores é sua alta resolução para determinar a energia de radiação, reduzindo a incerteza na medição. (CNEN - COMISSÃO NACIONAL DE ENERGIA NUCLEAR, 2013)

Outras vantagens são as seguintes:

- Resposta linear (altura do pulso versus energia de partículas) sobre uma ampla faixa de energia;
- Maior eficiência para um determinado tamanho, devido à alta densidade de um sólido em relação a a de um gás;
- Possibilidade de configurações geométricas especiais;
- Rápido tempo de elevação do pulso (em relação aos detectores a gás);

- Capacidade de operar no vácuo;
- Insensibilidade aos campos magnéticos.

As características de um detector de semicondutores dependem não apenas do tipo de material utilizado, mas também da forma como o semicondutor foi feito, moldado e tratado. O tipo, tamanho, forma e tratamento do cristal desempenham um papel na operação e desempenho de um detector de semicondutores. (KNOLL, 2010)

## 2.2 CALIBRAÇÃO DE DETECTORES DE RADIAÇÃO

Calibração de detectores tem como propósito garantir que o instrumento está exibindo valores que estão de acordo com os fornecidos por padrões (sob condições especificadas), levando em consideração as incertezas de medição que podem estar associadas (INMETRO – INSTITUTO NACIONAL DE METROLOGIA, QUALIDADE E TECNOLOGIA, 2012). Sob condições de referência, todos parâmetros do instrumento de medição e sua eletrônica associada tem valores de referência. Fatores de calibração (que adequam as leituras do detector ao uso específico) são aplicados ao instrumento, dependendo das condições em que a medição será feita. Os fatores de calibração utilizados dependem de uma gama de parâmetros, como energia, ângulo de incidência, dose equivalente, tipo de radiação e uma variedade de condições ambientais. As calibrações de rotina devem ser feitas periodicamente para garantir que os fatores de calibração continuam adequados ao uso daquele detector. (KNOLL, 2010)

Um dos métodos mais comuns de calibração de um detector se dá utilizando fontes de radiação de referência produzidas por laboratórios primários. Para uma fonte que tem uma dose equivalente conhecida  $H$ , sob condições de referência, o fator de calibração é obtido pela equação 1:

$$N = H/M \quad (1)$$

onde  $N$  é o fator de calibração,  $H$  é o valor verdadeiro da dose a ser mensurada e  $M$  é o valor medido pelo instrumento que está sendo calibrado. (IAEA – INTERNATIONAL ATOMIC ENERGY AGENCY, 2000)

Um dos fatores que mais facilmente pode levar a um erro durante a calibração é a posição do detector em relação à fonte de radiação. Um pequeno desvio no posicionamento pode causar um grande erro de medição. (IAEA – INTERNATIONAL ATOMIC ENERGY AGENCY, 2000)

### 2.3 AUTOMAÇÃO E CONTROLE

O rápido avanço tecnológico tem possibilitado o desenvolvimento de sistemas automatizados para vários tipos de aplicação. Com os diversos tipos de computadores, microcontroladores e outros componentes, é possível automatizar processos antes feitos apenas de forma manual. A automação com componentes eletrônicos permite maior eficiência, confiabilidade, precisão e segurança em diversos processos, desde os mais simples que podem ser encontrados em qualquer casa hoje em dia, quanto os mais complexos que poderiam ser encontrados em grandes indústrias.

Os tópicos a seguir descrevem o funcionamento de vários componentes que serão usados neste trabalho para realizar a automação do sistema.

#### 2.3.1 Microcontroladores

Um microcontrolador é um pequeno computador num único circuito integrado. Os microcontroladores são importantes devido à alta demanda por sistemas embarcados, ou seja, sistemas computacionais que atendem apenas uma demanda específica. Por isso, são compactos e de baixo custo. Além disso, o consumo de energia dos microcontroladores é relativamente baixo e sua programação é relativamente simples, especialmente quando comparados a microprocessadores. (GRIDLING; WEISS, 2007)

Figura 5 - Imagem de um microcontrolador



Fonte: Banco de imagens da internet

Os microcontroladores são utilizados em diversos produtos e dispositivos automatizados, como eletrodomésticos, controles remotos, dispositivos médicos, ferramentas elétricas e muitos outros. Os microcontroladores foram uma grande revolução para várias indústrias e também para a sociedade como um todo, já que permitiram a evolução de sistemas que não se desenvolviam há anos. (GRIDLING; WEISS, 2007)

Internamente, um microcontrolador possui:

- Uma CPU (Unidade de Processamento Central), que interpreta as instruções do programa enviado ao controlador,
- Memória PROM (Memória Programável Somente de Leitura), que é usada para gravar a programação,
- Memória RAM (Memória de Acesso Aleatório), cuja função é memorizar as variáveis utilizadas na programação,
- Conjunto de portas I/O (Entrada e Saída de comandos), onde é possível conectar dispositivos que darão comandos ao sistema, bem como dispositivos que serão operados pelo sistema,
- Conjunto de dispositivos auxiliares que são necessários ao funcionamento do controlador, como contadores, dispositivos de comunicação serial, entre outros.

### 2.3.2 Arduino

Em 2005, um grupo de pesquisadores decidiu criar um dispositivo que fosse funcional, fácil de programar, que tivesse baixo custo e que fosse acessível para que muitos estudantes, projetistas amadores e qualquer outra pessoa pudesse utilizar em projetos eletrônicos e protótipos de automação. Assim surgiu o Arduino. (MCROBERTS, 2018)

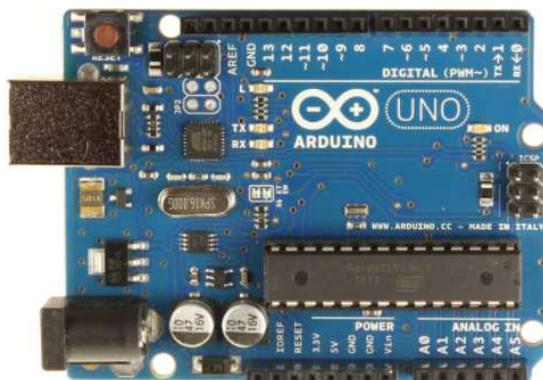
O Arduino é uma placa composta por um microcontrolador Atmel, com uma variedade de portas I/O e que pode ser facilmente conectada à maioria dos computadores pessoais através de uma conexão USB. O Arduino é programado através de um IDE (*Integrated Development Environment*), que é um tipo de software para se escrever código de programação, e utiliza uma linguagem de programação baseada em C/C++, uma das mais conhecidas e usada há anos. Dessa forma, sem necessidade de qualquer outro equipamento além de um computador e um cabo USB, é possível programar o Arduino para controlar e automatizar diversos tipos de sistemas, de maneira fácil, rápida e barata. (AVALCANTE; TAVOLARO; MOLISANI, 2011)

Após programado, o Arduino pode receber comandos de outros dispositivos conectados a ele, como botões, controles remotos, sensores e receptores de sinais bluetooth. E também pode enviar comandos para controlar dispositivos como lâmpadas, motores entre outros. (MCROBERTS, 2018)

O Arduino foi idealizado como um sistema de hardware e código abertos. Isso significa que qualquer um pode modificar ou personalizar o Arduino conforme a necessidade.

Existem vários modelos de Arduino disponíveis, sendo o Arduino Uno e o Arduino Mega 2560 os mais comuns. Todos os modelos têm funcionalidades muito parecidas, tendo como diferença algumas características como quantidade de portas I/O, quantidade de memórias e tamanho físico. Portanto, apesar de ser possível utilizar diferentes modelos em alguns projetos, certos modelos podem ser mais apropriados para determinadas situações. (MCROBERTS, 2018)

Figura 6 - Arduino UNO



Fonte: Banco de imagens da internet

Figura 7 - Arduino Mega 2560



Fonte: Banco de imagens da internet

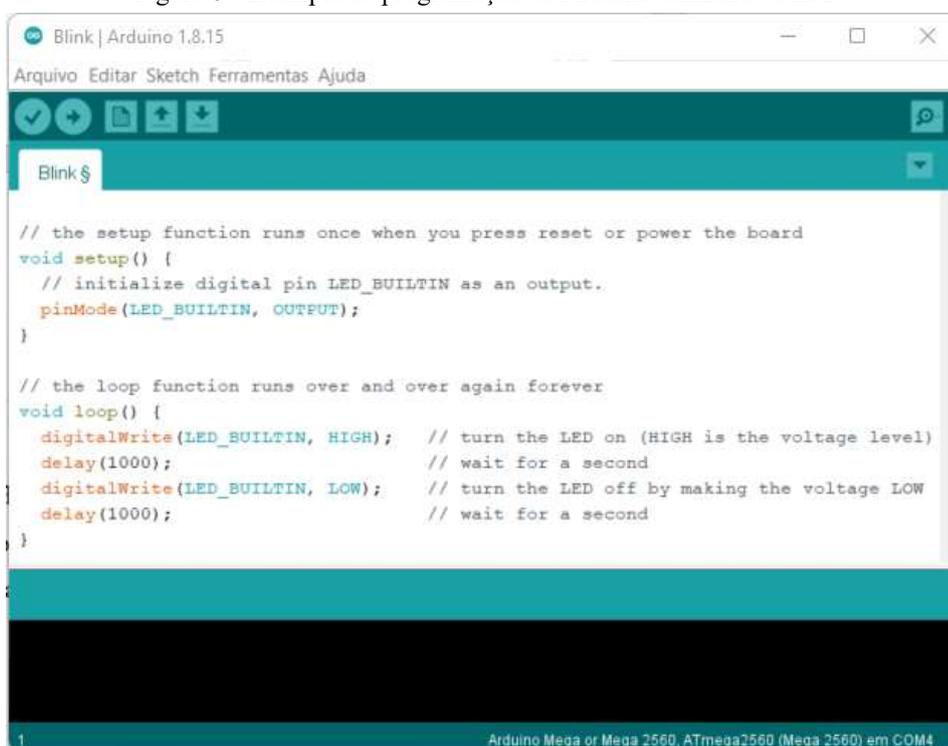
### 2.3.2.1 Programação do Arduino

A programação do Arduino é feita utilizando-se uma linguagem baseada em C/C++. Embora essa linguagem possa ser complexa e difícil de se dominar, em geral as programações no Arduino são bastante simples. Além disso, por ser encarado como um software livre e usado por vários hobbystas, é possível encontrar na internet diversos códigos prontos para certas aplicações, bastando apenas fazer pequenas alterações para adequá-lo ao uso específico. (EVANS; NOBLE; HOCHENBAUM, 2013)

Para escrever a programação e enviá-la para o Arduino, é utilizado um software chamado IDE, que é um ambiente de programação. Este software é gratuito e pode ser baixado no site oficial do Arduino (<https://www.arduino.cc/en/Main/Software>).

Em geral, a programação é feita em dois blocos. O primeira, chamado “setup()”, é onde se configura valores iniciais do sistema, define quais portas do Arduino serão utilizadas e como serão utilizadas e pode-se escrever mensagens para o usuário que utilizará o sistema. No segundo bloco, chamado “loop()”, define-se os comandos que o Arduino enviará para as portas que foram configuradas. Os comandos são repetidos até que algum outro comando faça a repetição ser encerrada. (MONK, 2013)

Figura 8 - Exemplo de programação do Arduino utilizando o IDE



```
Blink | Arduino 1.8.15
Arquivo Editar Sketch Ferramentas Ajuda

Blink $

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

1 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) em COM4
```

Fonte: Elaborado pelo autor

### 2.3.3 Motores de passo

O motor de passo é um motor elétrico que não possui escovas ou comutadores. O rotor é composto de um ou mais ímãs permanentes. O estator tem várias bobinas por onde passam corrente elétrica e criam um campo magnético. Para que o motor gire, um circuito eletrônico aciona rápida e repetidamente as bobinas numa sequência correta para que o rotor gire. Cada pulso elétrico enviado faz com que o motor avance um passo. A velocidade do motor varia de

acordo com a frequência e pulsos que o circuito de controle envia. O sentido de rotação também pode ser alterado através da sequência de polarização das bobinas. (BODSON, 1993)

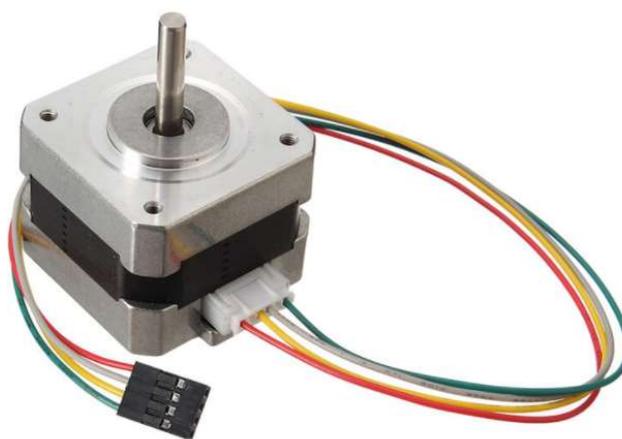
A maioria dos motores de passo precisa de 200 passos para dar um giro completo no rotor. Mas há motores com diferentes números de passos. Além disso, através do controlador associado ao motor, é possível fazer um maior controle dos passos, técnica chamada de microstepping. (BALUTA, 2007)

A vantagem de se utilizar um motor de passo é o controle que pode ser obtido com ele. Com o motor de passo é possível ter um grande controle da velocidade, aceleração e posição do giro do rotor, o que faz com que ele seja ideal para aplicações onde é necessária uma acurácia no posicionamento. Um exemplo de aplicação de motores de passo é no funcionamento de impressoras 3D e em CNCs. (BALUTA, 2007)

Os motores de passo podem ser unipolares ou bipolares. Os motores unipolares tem duas fases com enrolamentos com um terminal central. Nesses motores, cada uma das fases é energizada por um circuito de controle em apenas um sentido. Então, uma extremidade do enrolamento sempre será positiva enquanto a outra será negativa. Com isso, os motores unipolares acabam tendo menos torque do que um motor bipolar similar. (BODSON, 1993)

Já os motores bipolares têm duas fases, normalmente sendo um enrolamento por fase. Para se reverter um polo magnético, é necessário inverter a corrente no enrolamento. Os motores bipolares são mais potentes que outros motores unipolares de mesmo tamanho. (BODSON, 1993)

Figura 9 - Exemplo de motor de passo bipolar



Fonte: Banco de imagens da internet

### 2.3.4 Driver do motor

Em geral, uma placa microcontroladora, como o Arduino, não fornece corrente suficiente para o acionamento de um motor. Portanto, é preciso utilizar um circuito intermediário, chamado driver do motor. O microcontrolador é programado para dar comandos para o driver, que, por sua vez faz o acionamento do motor de passo.

Com o driver do motor, é possível controlar funções como habilitar ou desabilitar o motor, mudar o sentido de giro, controlar a velocidade e realizar o micro-stepping, que possibilita uma resolução ainda maior da posição do rotor. (ARDUFOCUS)

#### 2.3.4.1 Driver A4988

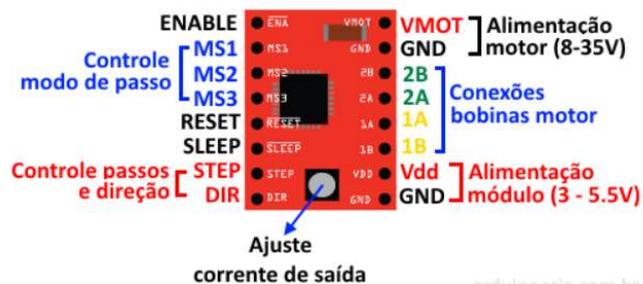
O driver A4988 é um driver bastante popular em projetos que utilizam motores de passo. Este é um driver de custo muito baixo e que permite um grande controle do motor. Ele possui duas pontes-H com transistores FET DMOS para fazer o controle de motores de passo bipolares. A resistência interna é baixa devido ao uso dos FETs, o que faz com que o calor dissipado pelo driver também seja baixo. Além disso, ele conta com proteção contra altas temperaturas e contra curtos-circuitos. (ELETROGATE, 2018)

Este driver também consegue realizar micro-stepping, permitindo 5 modos de controle de passo. Os modos de controle de passo são:

- Passo completo (Full step)
- Meio passo (Half step)
- Um quarto de passo (Quarter step)
- Um oitavo de passo (Eighth step)
- Um dezesseis-avos de passo (Sixteenth step)

No módulo, temos as portas de controle do motor de um lado e as conexões com o motor e pinos de monitoramento de corrente do outro.

Figura 10 - Interface do driver A4988



Fonte: Elaborado pelo autor

Cada pino do módulo tem uma diferente função:

- **STEP** (passo) – um sinal com transição de baixo para alto avança um passo,
- **DIR** – permite a alteração no sentido da rotação do motor,
- **ENABLE** – um sinal baixo ativa os drivers do motor,
- **RESET** – um sinal baixo dá um reset no chip, desativando todos os drivers de saída,
- **SLEEP** (dormir) – esse pino quando negativo, desabilita alguns circuitos internos para economizar energia.
- **MS1, MS2 e MS3** – são entradas para selecionar o modo da sequência dos passos (micro stepping).

#### 2.3.4.1.1 Modos de passo do driver A4988

Os pinos MS1, MS2 e MS3 devem ser acionados de acordo com a Tabela 1 para realizar o modo de passo desejado.

Tabela 1 - Modos de passo do driver A4988

Pinos			Resolução micropassos
MS1	MS2	MS3	
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	Quarter step (1/4)
High	High	Low	Eighth step (1/8)
High	High	High	Sixteenth step (1/16)

Os motores de passo têm, em sua maioria, 200 passos por volta. Isso significa que cada passo corresponde a  $1,8^\circ$  de rotação. Para avançar cada passo completo, o motor precisa receber 1 pulso. Portanto, quando operado em modo de passo completo, o motor recebe 200 pulsos por revolução (PPR).

No entanto, quanto menor o modo de passo, mais pulsos são necessários para cada revolução, já que serão necessários mais do que 1 pulso para o avanço de 1 passo completo.

Tabela 2 - Pulsos por revolução em cada modo de passo do driver A4988

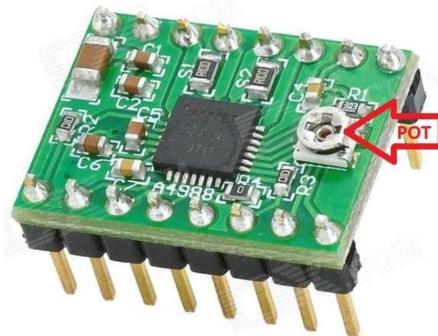
Modo de passo	PPR
Full Step	200
Half Step	400
Quarter Step	800
Eighth Step	1600
Sixteenth Step	3200

#### 2.3.4.1.2 Ajuste de corrente do driver A4988

O driver A4988 tem proteção contra curtos-circuitos, excesso de temperatura e excesso de corrente. A corrente máxima do driver precisa ser ajustada para o mesmo valor de corrente máxima do motor, de acordo com o datasheet. Para verificar a corrente nas bobinas do motor, mede-se a tensão nos resistores R1 e R2 do driver. Essas tensões são então comparadas com a tensão de referência do driver A4988. (ELETROGATE, 2018)

A tensão de referência pode ser ajustada girando-se o trimpot (POT) do driver, conforme é exibido na Figura 11.

Figura 11 - Indicação do trimpot do driver A4988



Fonte: Banco de imagens da internet

Para realizar o ajuste de corrente, o driver deve estar configurado no modo de passo completo. Com o driver alimentado pelo Arduino, mede-se a tensão entre o pino GND do driver A4988 e o trimpot, utilizando-se um voltímetro. O cálculo da tensão  $V_{ref}$  que deve ser medida no voltímetro segue a fórmula:

$$V_{ref} = I_{max} \times 8 \times RS \quad (2)$$

Onde  $V_{ref}$  é a tensão de referência a ser colocada no driver,  $I_{max}$  é a corrente máxima suportada pelo motor de acordo com o datasheet e  $RS$  é a resistência dos resistores  $RS1$  e  $RS2$  (não somados). Com a fórmula, descobre-se o valor da tensão de referência que deve ser lido no voltímetro. Caso a tensão não esteja adequada, gira-se o trimpot até que a tensão esteja correta. (ARDUFOCUS)

## 2.4 IMPRESSÃO 3D

Impressão 3D é um tipo de prototipagem rápida que permite fabricar peças desenhadas via software CAD. As impressoras 3D mais comumente usadas criam modelos tridimensionais usando uma técnica de sobreposição de camadas, injetando o material de um filamento que preenche cada uma das camadas da peça.

Em sua maioria, as peças feitas com impressão 3D são fabricadas com algum tipo de plástico, existindo diversos tipos de filamentos disponíveis. Cada um desses tipos pode ser usado em diferentes aplicações e dão as peças diferentes propriedades e acabamento.

A impressão 3D tem sido largamente utilizada por empresas e escolas que desejam desenvolver projetos de prototipagem de maneira relativamente fácil, rápida e barata. (3DPRINTING.COM)

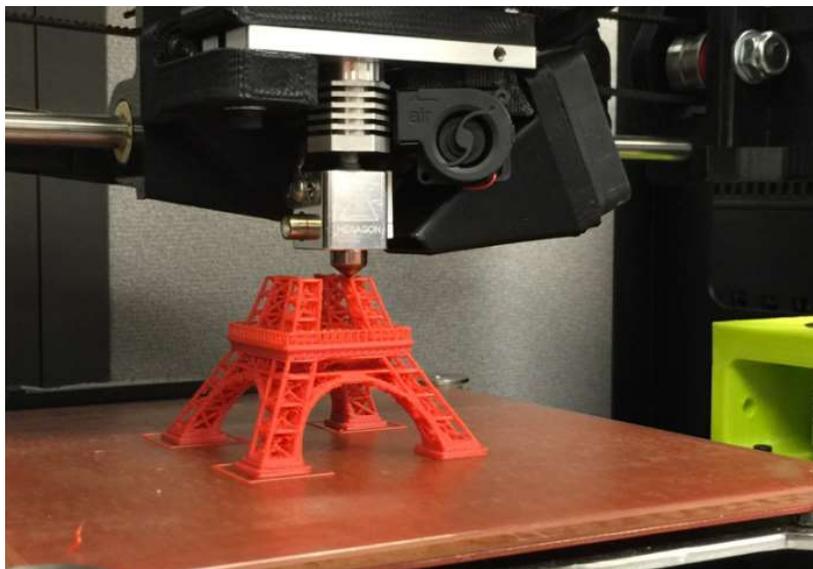
Figura 12 - Impressora 3D



Fonte: Banco de imagens da internet

Para se fazer uma impressão 3D, é utilizada uma impressora 3D. As impressoras 3D mais comuns são compostas de um compartimento em formato de caixa e uma bandeja de vidro, onde vai ser fabricada a peça. Acima da bandeja, encontra-se um bico injetor, que é alimentado com um filamento de material plástico. A posição do bico injetor é controlada por motores de passo, que por sua vez são controlados por microcontroladores. O bico injetor se esquentava até uma temperatura em que derrete o material do filamento, podendo assim despejar lentamente o material do filamento nos pontos que precisam ser preenchidos para a construção da peça. A fabricação sempre se dá de baixo para cima, e o material vai sendo injetado nas camadas mais baixas da peça até que se finalize com a camada mais acima. (WOODFORD, 2021)

Figura 13 - Bico injetor de uma impressora 3D durante processo de impressão



Fonte: Banco de imagens da internet

Existem também outros tipos de impressoras 3D, que podem trabalhar com mais variados tipos de materiais e utilizam diferentes técnicas de impressão, porém estas são, em geral, mais caras para se adquirir e utilizar, o que as faz menos populares.

### 3 METODOLOGIA E MATERIAIS

O sistema tem duas aplicações. A primeira é simular experimentalmente uma fonte “infinita”, ou seja, de grandes dimensões, a partir de uma fonte puntual de laboratório. Isso é possível através da aquisição de dados de diferentes posicionamentos do detector em relação a fonte, de maneira que a somatória dos dados represente a medição que o detector faria no caso de uma fonte de radiação de dimensão maior. Esses dados podem ser usados para validar simulações computacionais que utilizam o código MCNP em futuros trabalhos.

A segunda aplicação do sistema é atuar como um auxiliador para calibração de detectores de radiação. O sistema possibilita uma grande confiabilidade na distância fonte e detector. Portanto, conhecendo-se a fonte laboratorial que será usada para a calibração, o sistema vai assegurar um deslocamento preciso do detector para que a calibração seja feita corretamente.

Utilizando um software CAD (*Computer-Aided Design*), o sistema foi desenhado e projetado. O software utilizado foi o Siemens NX. Algumas das peças e componentes do projeto foram compradas anteriormente, enquanto as demais peças foram desenhadas no CAD para, posteriormente, serem impressas com impressão 3D.

Com as peças 3D impressas, foi realizada a montagem do sistema, seguida pelas ligações elétricas, programação do Arduino, testagem do funcionamento do sistema e aquisição de algumas medições com um conjunto fonte-detector.

#### 3.1 PRINCÍPIO DE FUNCIONAMENTO DO SISTEMA

O sistema conta com 2 motores de passo, controlados por Arduino, que fazem o deslocamento de um detector de radiação em relação a uma fonte de referência puntual. O deslocamento é linear, aproximando ou afastando o detector da fonte, e também angular, girando o detector no eixo da fonte.

##### 3.1.1 Utilizando o sistema como auxiliador de calibração de detector de radiação

Para se usar o sistema de posicionamento com objetivo de calibrar um detector, apenas o deslocamento linear é necessário. O Arduino, em conjunto com o motor de passo, consegue garantir um deslocamento muito preciso. Assim, a distância fonte-detector, será sempre

bastante acurada. Conhecendo-se a fonte de referência, basta posicionar o detector a uma distância específica e regular o fator de calibração até que a medição coincida com o valor esperado da fonte de referência.

Para averiguar se o deslocamento linear estava preciso, foram feitos testes com diferentes comandos de distâncias para o sistema. Utilizando-se uma trena (com precisão de 1 milímetro), mediu-se o deslocamento real que o sistema executou.

### 3.1.2 Utilizando o sistema para simular fonte infinita a partir de fonte puntual

As simulações computacionais de radiação, que são feitas por pesquisadores no DEN, precisam ser validadas, comparando-se seus resultados com alguns dados experimentais reais. Porém, em laboratório, temos apenas fontes de referência pequenas, que a partir de uma certa distância do detector, praticamente atuam como fontes pontuais. Nas simulações, assim como em situações reais, é provável que as fontes de radiação sejam bem maiores. Portanto, para se obter dados experimentais para uma fonte grande (chamada de “infinita” nesse trabalho), é possível mudar o posicionamento do detector em relação a fonte puntual, conseguindo-se assim várias medições diferentes que simulam uma fonte infinita.

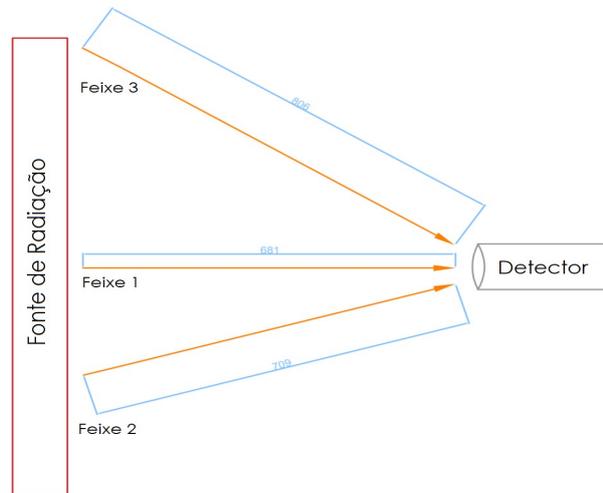
O sistema tem algumas limitações de posicionamento. Pode-se posicionar o detector nos ângulos extremos de  $-45^\circ$  e  $45^\circ$ . A distância máxima que o detector pode ficar da fonte de radiação é de 1027 milímetros. O ajuste de altura da fonte e do detector são feitos manualmente, o que significa que é necessário alinhar ambos antes de iniciar o posicionamento automatizado.

Figura 14 - Esquema de fonte puntual emitindo feixe de radiação com ângulo  $0^\circ$  no detector



Fonte: Elaborado pelo autor

Figura 15 - Esquema de fonte infinita emitindo feixes de radiação com diferentes ângulos e distâncias no detector

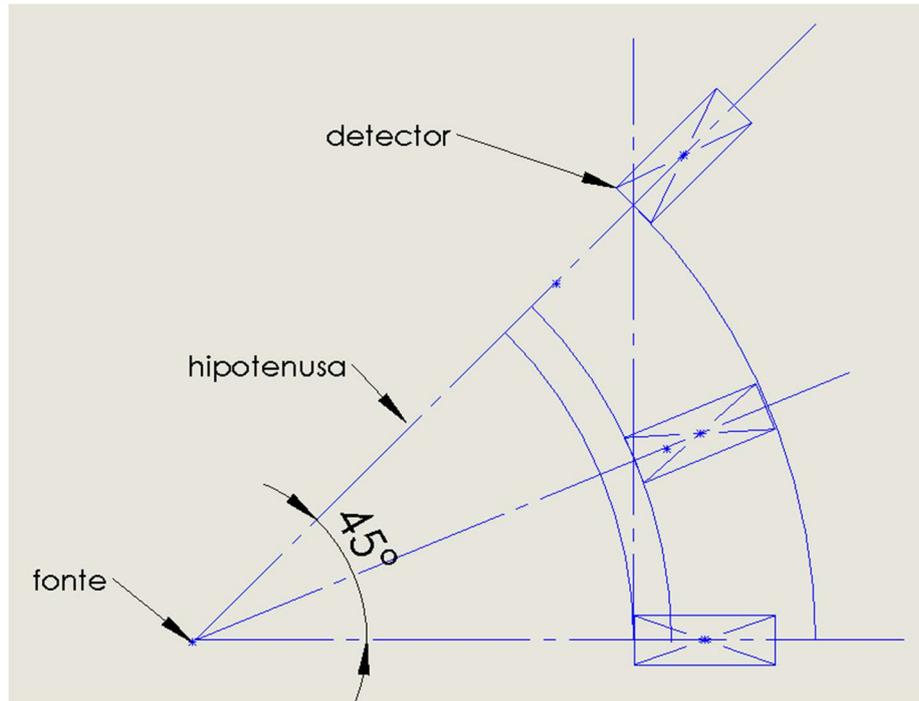


Fonte: Elaborado pelo autor

Para simular a fonte infinita, é necessário fazer medições em diferentes distâncias e ângulos. Primeiro, realizou-se uma medição com um ângulo de  $0^\circ$  entre fonte e detector. Conforme alterou-se o ângulo, a distância linear aumenta e novas medições foram feitas. Cada medição feita em um diferente ângulo (e conseqüente distância linear), representa um feixe emitido de um ponto diferente da fonte plana infinita que está sendo simulada. Após todas as medições serem feitas, os dados coletados são utilizados em conjunto para representar a leitura que o detector faria se estivesse parado e de frente a uma fonte plana infinita, que emitiria feixes vindos de diferentes direções.

Para realizar a medição do ângulo feito pelo sistema, mediu-se com uma trena (de precisão de 1 milímetro) a distância percorrida no eixo X depois de um deslocamento angular. Essa distância no eixo X representa o cateto oposto do triângulo formado pelo movimento do sistema. A hipotenusa desse triângulo é a distância entre o eixo de rotação do sistema e o fim da guia linear. Com essas duas medidas, é possível fazer o cálculo trigonométrico para a obtenção do ângulo que foi realmente feito pelo sistema. A Figura 16 mostra os triângulos formados pela rotação do sistema (e conseqüente deslocamento linear).

Figura 16 - Esquema de deslocamento angular com consequente deslocamento linear



Fonte: Elaborado pelo autor

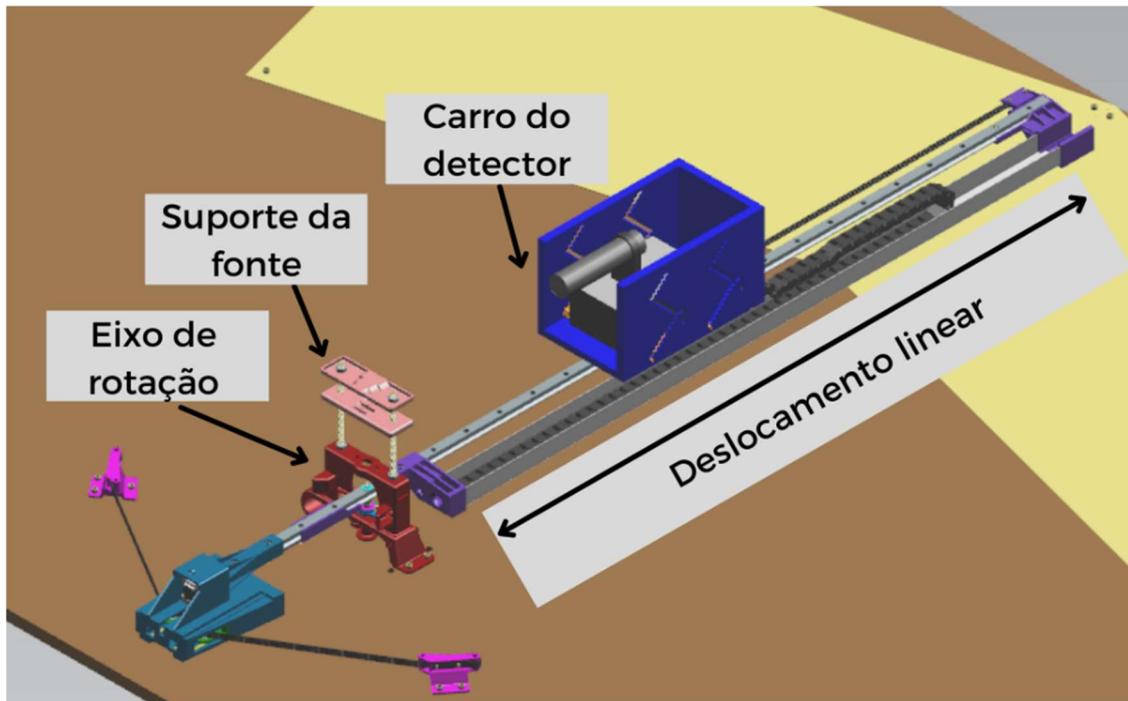
Esses resultados podem, por fim, ser comparados com os dados de simulações computacionais com fonte infinita, com objetivo de validar estas simulações.

A fim de fazer uma simulação de fonte infinita, foi testado um conjunto de detector Geiger-Muller e uma fonte puntual de Cobalto.

### 3.2 DESENHO E PROJETO DO SISTEMA

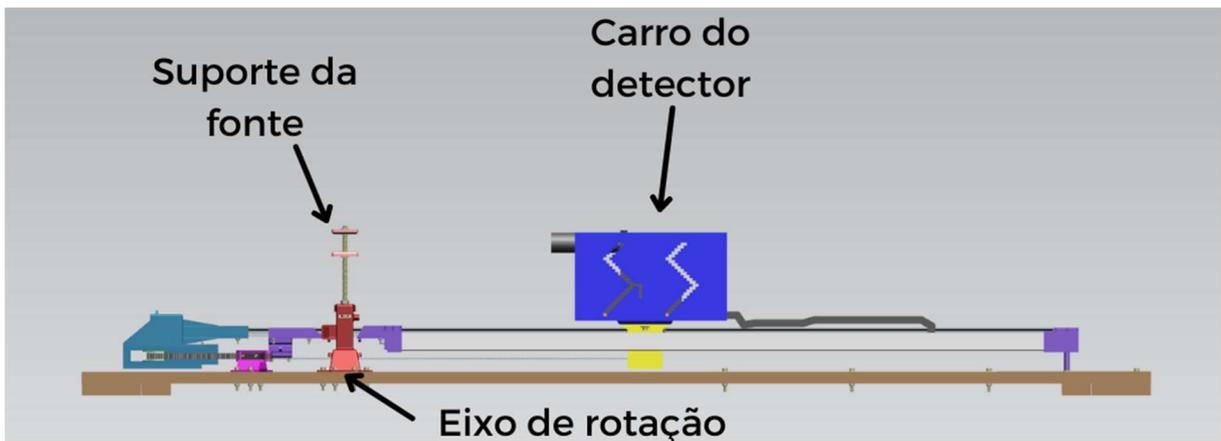
O projeto foi feito utilizando-se o software Siemens NX, que é um CAD 3D. As Figuras 17, 18, 19, 20, 21 e 22 mostram o sistema de diferentes ângulos de visão, com alguns detalhes de como funcionará o projeto quando construído.

Figura 17 - Vista isométrica do sistema de deslocamento do detector



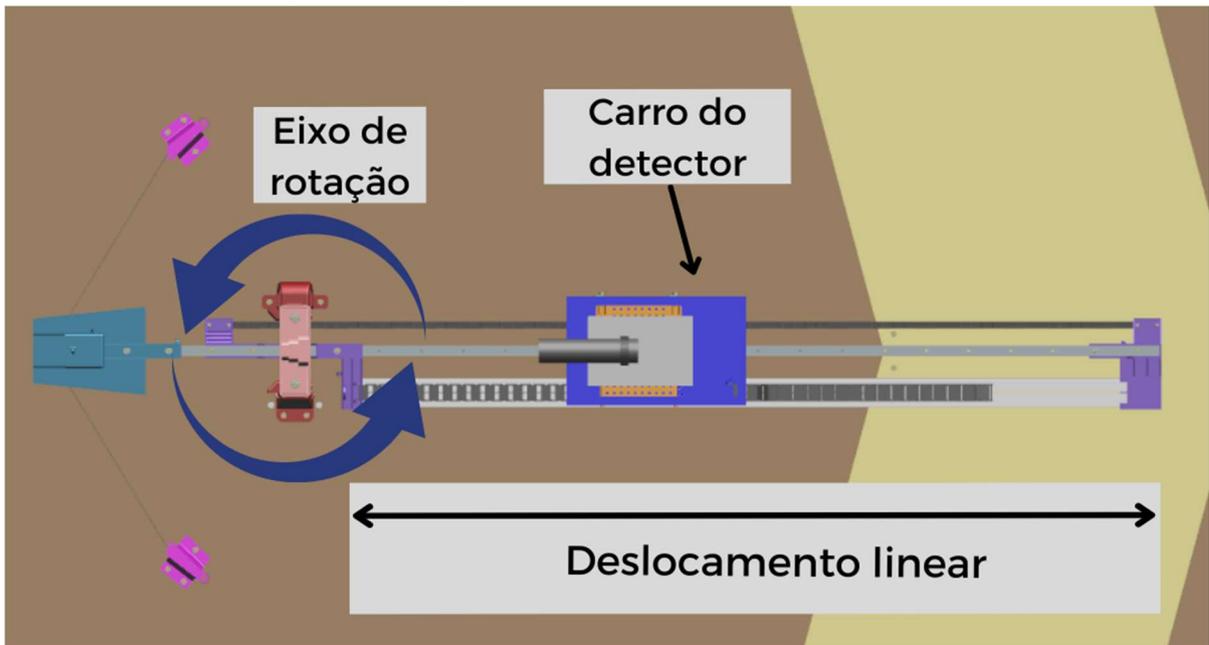
Fonte: Elaborado pelo autor

Figura 18 - Vista lateral do sistema de deslocamento do detector



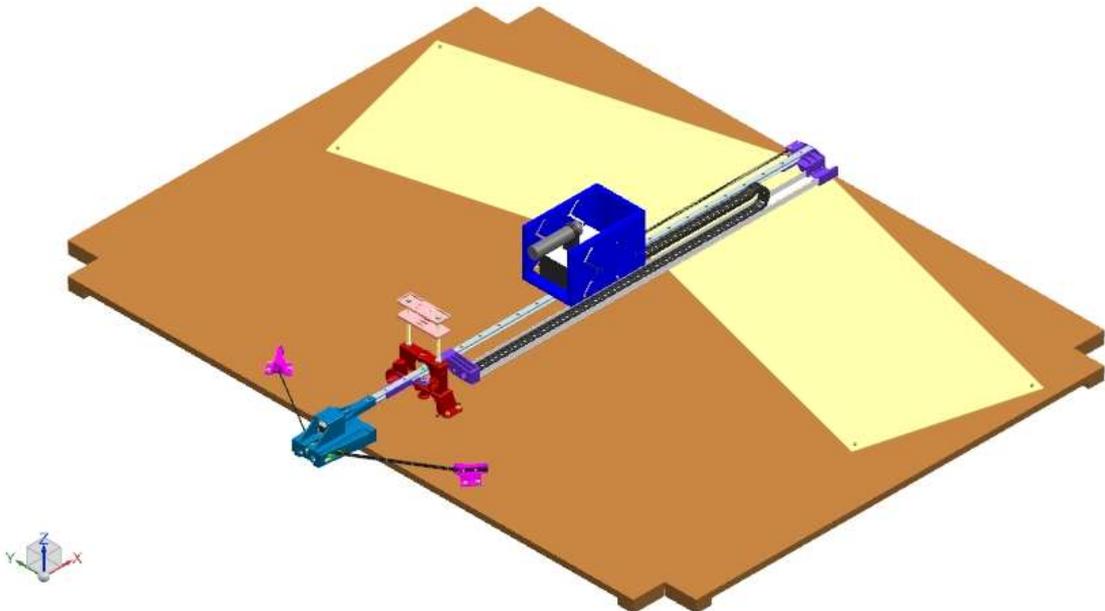
Fonte: Elaborado pelo autor

Figura 19 - Vista planta do sistema de deslocamento do detector

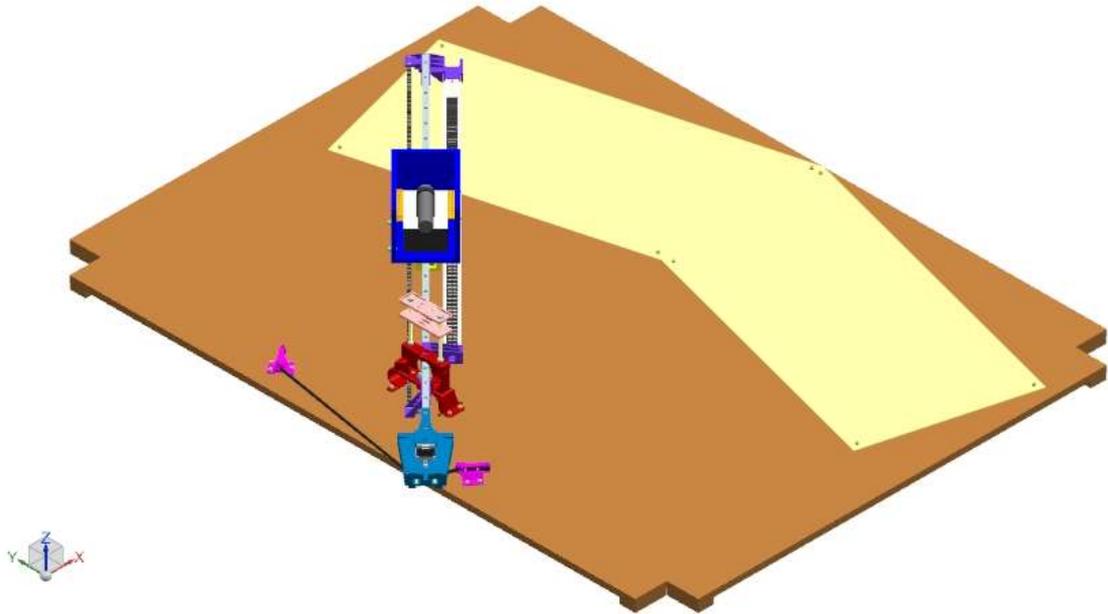


Fonte: Elaborado pelo autor

Figura 20 - Vista isométrica do sistema em posição de ângulo 0°

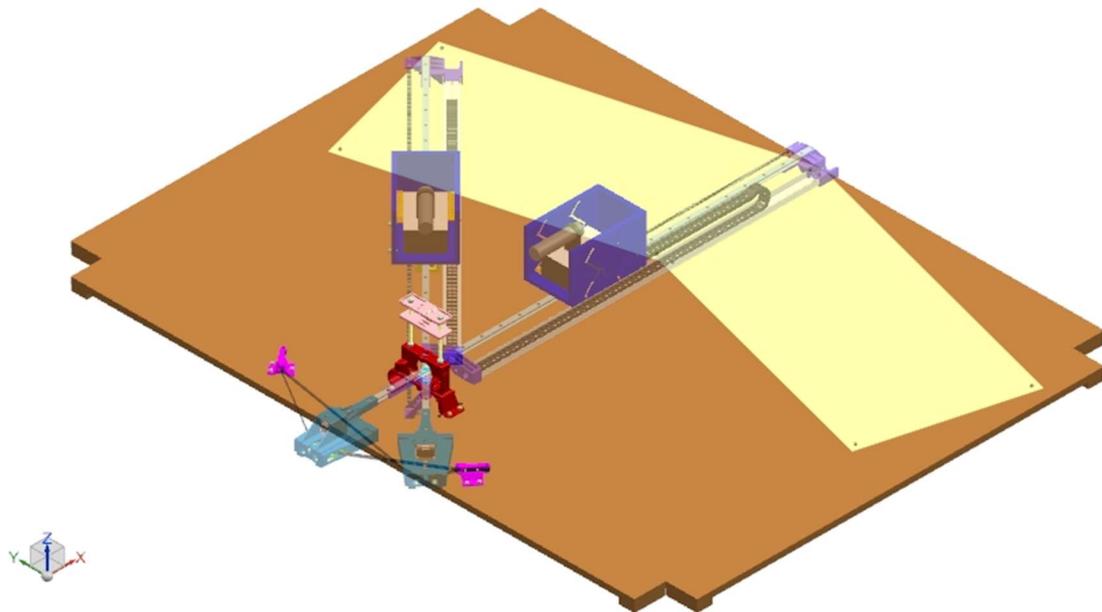


Fonte: Elaborado pelo autor

Figura 21 - Vista isométrica do sistema em posição de ângulo  $-45^\circ$ 

Fonte: Elaborado pelo autor

Figura 22 - Vista isométrica do sistema com imagens sobrepostas em 2 diferentes posições

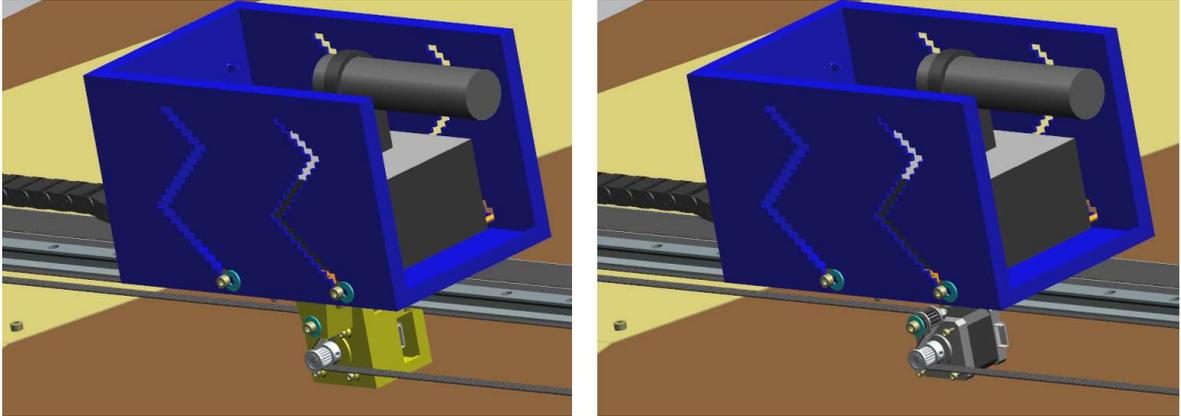


Fonte: Elaborado pelo autor

O detector fica acoplado em um carro que se desloca em uma guia linear, controlado por um dos motores de passo, como mostrado em detalhes na Figura 23. Próximo ao fim da guia linear, em um suporte fixo, fica posicionada a fonte de radiação. O suporte consegue

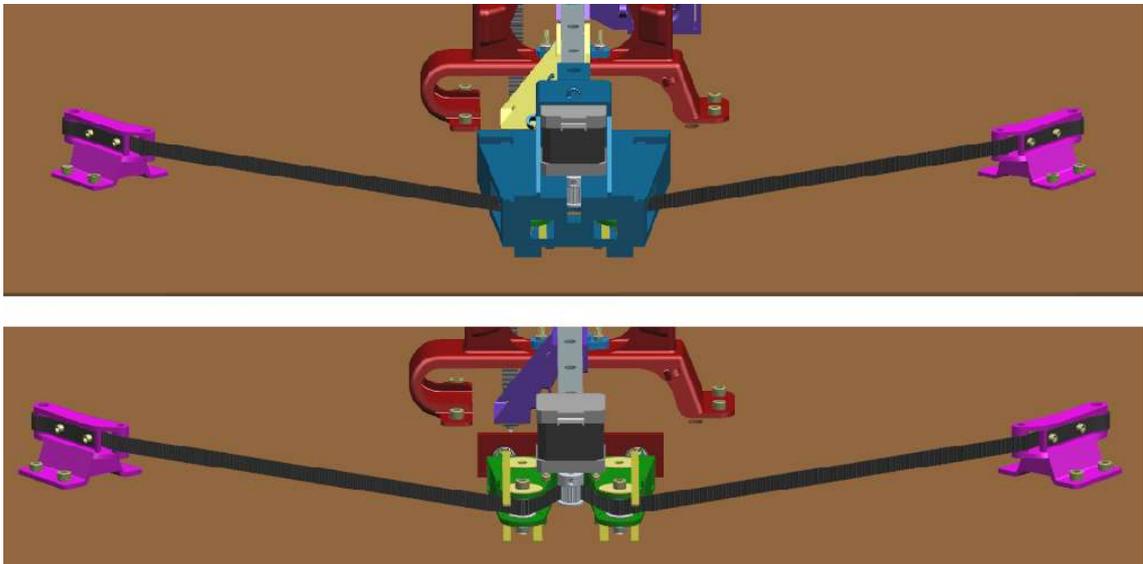
regular a altura da fonte, para que ela fique alinhada à altura do detector. E, neste mesmo suporte, fica um eixo, no qual a guia linear gira, podendo fazer um arco de 90°. A rotação é feita pelo segundo motor de passo, localizado na extremidade da guia.

Figura 23 - Detalhe do motor do deslocamento linear com o suporte do motor (esq.) e sem o suporte do motor (dir.)



Fonte: Elaborado pelo autor

Figura 24 - Detalhe do motor do deslocamento angular (rotação) com o suporte do motor (acima) e sem o suporte do motor (abaixo)

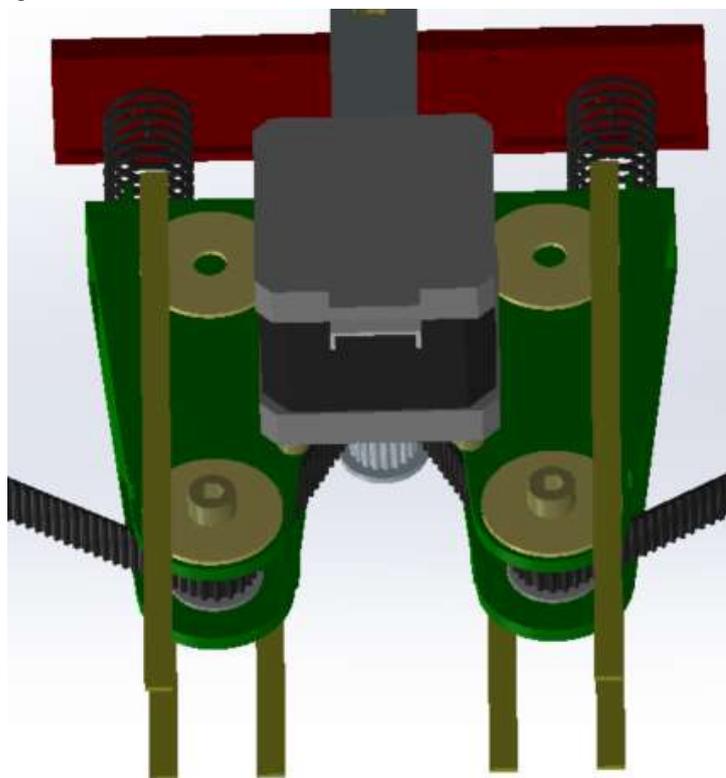


Fonte: Elaborado pelo autor

A correia dentada por onde corre o motor que faz o deslocamento angular precisa estar constantemente esticada. Caso contrário, após iniciar o movimento, a correia se soltaria. Para

isso, dentro do suporte do motor angular, foi projetado um sistema de molas e esticador, que mantém a correia sempre esticada e abraçando as polias.

Figura 25 - Detalhe do sistema de molas e esticador da correia dentada



Fonte: Elaborado pelo autor

### 3.3 LISTA DE MATERIAIS E COMPONENTES DO SISTEMA

Para a execução e montagem do projeto, foram necessários diversos componentes que foram comprados, além de outros que foram fabricados em impressora 3D. Abaixo estão as listas de componentes do sistema.

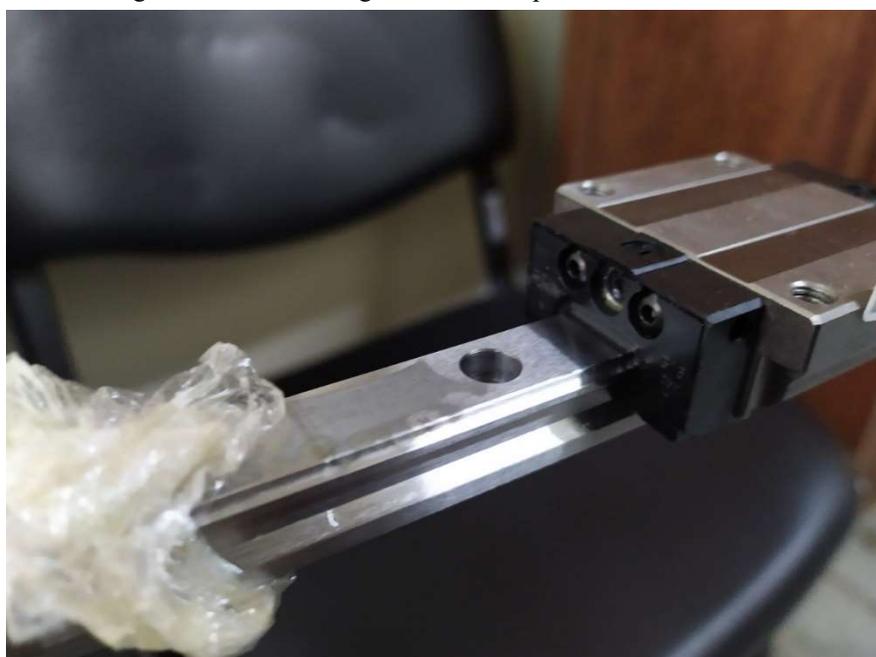
A Tabela 3 mostra os componentes da montagem do sistema que foram que foram comprados diretamente no mercado.

Tabela 3 - Componentes comprados

Quantidade	Componente
2	Motor de passo bipolar NEMA 17HS4401
1	Guia linear 15mm de 1,5 metros de comprimento, com patins
1	Esteira porta cabos de 1,5 metros, 10mm x 20mm
1	Calha em L de 3 metros
2	Correia dentada GT2, 10mm de largura, passo 2mm, 2 metros de comprimento
1	Chapa de teflon 1m x 1m x 1mm
1	Tábua compensado naval 1,60m x 2,20
3	Polia lisa GT2, 10mm de largura, eixo 5mm, flange dupla
2	Polia dentada GT2 com rolamento, 10mm de largura, passo 2mm, 20 dentes, eixo 5mm, flange dupla
2	Molas
1	Mancal com rolamento para eixo linear 8mm
1	Eixo para mancal
	Parafusos, Porcas e Arruelas

Abaixo, fotos de alguns dos componentes comprados diretamente no mercado.

Figura 26 - Detalhe da guia linear com patins de deslocamento



Fonte: Fotografado pelo autor

Figura 27 - Detalhe da polia dentada e correia dentada



Fonte: Fotografado pelo autor

Figura 28 - Esteira porta cabos



Fonte: Fotografado pelo autor

Figura 29 - Motor de passo bipolar NEMA 17HS4401



Fonte: Fotografado pelo autor

Também foram comprados alguns componentes eletrônicos, necessários para o funcionamento dos motores de passo com controle por Arduino. A Tabela 4 mostra a lista de componentes eletrônicos.

Tabela 4 - Componentes eletrônicos

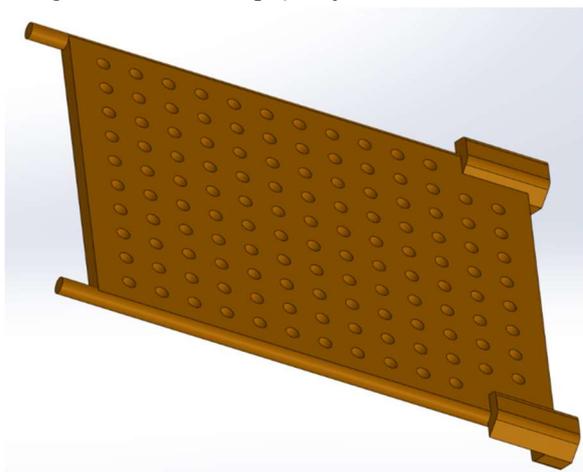
Quantidade	Componente
1	Arduino Mega 2560
1	Protoboard
2	Driver A4988
1	Capacitor 10 $\mu$ F
1	Capacitor 100 $\mu$ F
1	Fonte DC 12V
50	Cabos Jumper

Vários dos componentes do sistema projetado foram desenhados especificamente para este protótipo. A Tabela 5 com a lista de peças que serão fabricadas. Também são mostradas as imagens feitas no CAD de cada uma das peças a serem fabricadas.

Tabela 5 - Peças fabricadas em impressora 3D

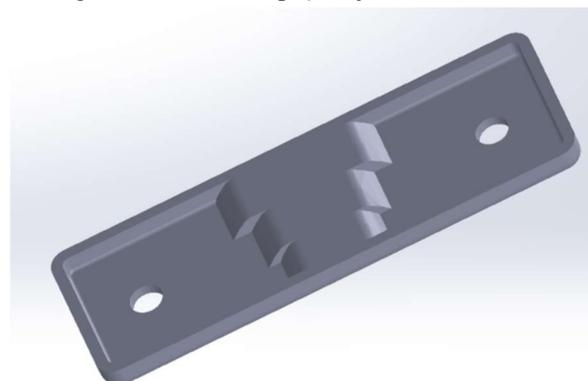
Quantidade	Componente
1	Ajuste Altura Detector
2	Ajuste Altura Fonte
1	Esticador
1	Placa Molas
1	Suporte Carrinho
1	Suporte Correia Angular 1
1	Suporte Correia Angular 2
1	Suporte Correia Linear 1
1	Suporte Correia Linear 2
1	Suporte Eixo Fonte
1	Suporte Motor Angular
1	Suporte Motor Linear
1	Suporte Perfil L

Figura 30 - Desenho peça 'Ajuste Altura Detector'



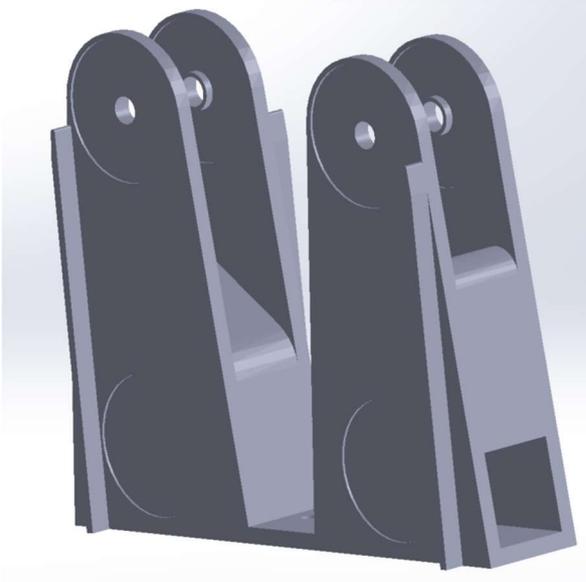
Fonte: Elaborado pelo autor

Figura 31 - Desenho peça 'Ajuste Altura Fonte'



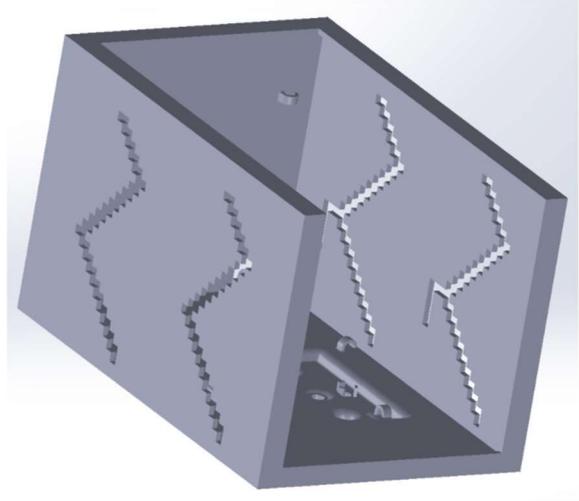
Fonte: Elaborado pelo autor

Figura 32 - Desenho peça 'Esticador'



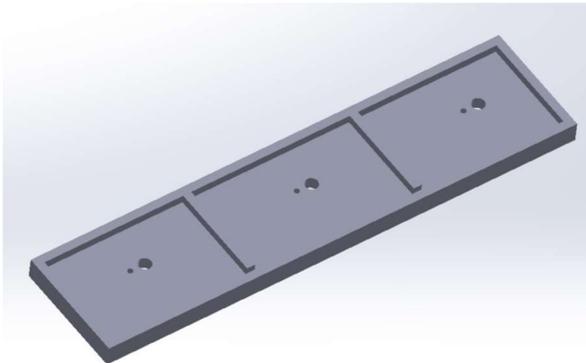
Fonte: Elaborado pelo autor

Figura 34 - Desenho peça 'Suporte Carrinho'



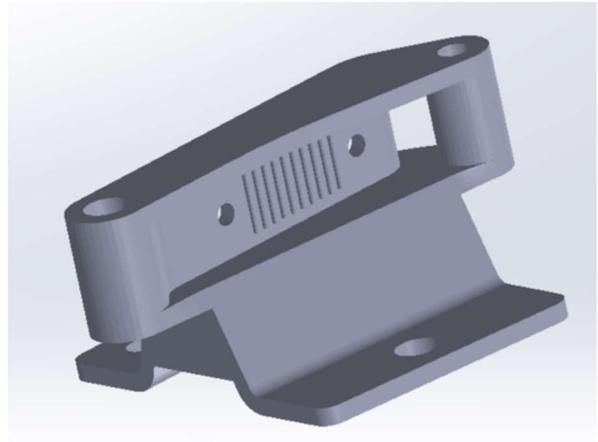
Fonte: Elaborado pelo autor

Figura 33 - Desenho peça 'Placa Molas'



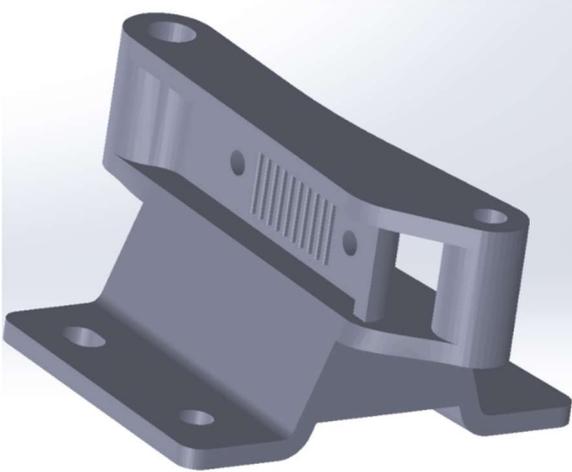
Fonte: Elaborado pelo autor

Figura 35 - Desenho peça 'Suporte Correia Angular 1'



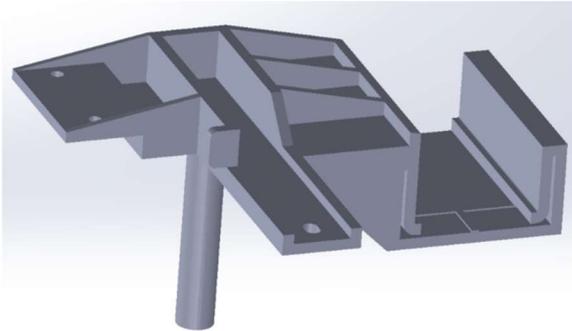
Fonte: Elaborado pelo autor

Figura 36 - Desenho peça 'Suporte Correia Angular 2'



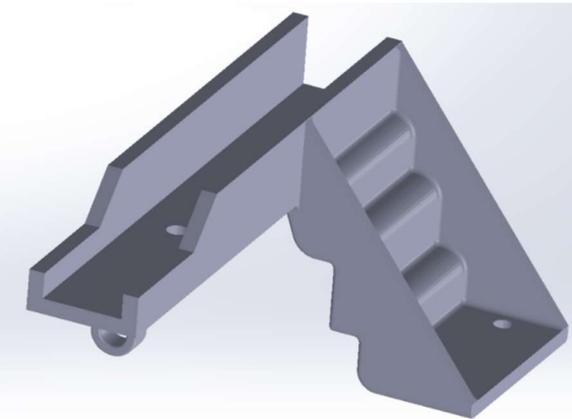
Fonte: Elaborado pelo autor

Figura 37 - Desenho peça 'Suporte Correia Linear 1'



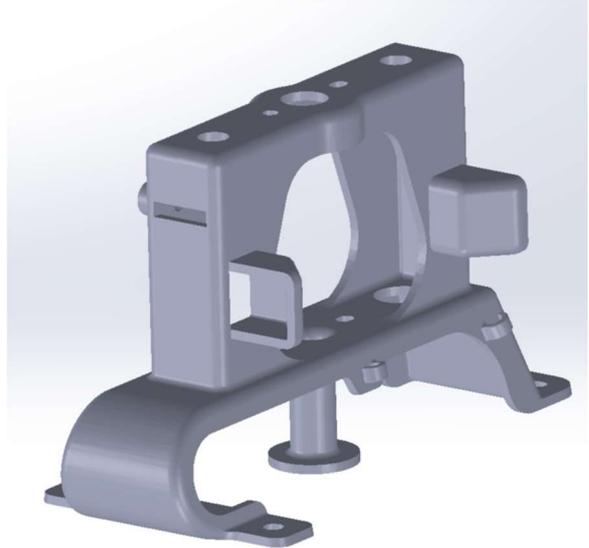
Fonte: Elaborado pelo autor

Figura 38 - Desenho peça 'Suporte Correia Linear 2'



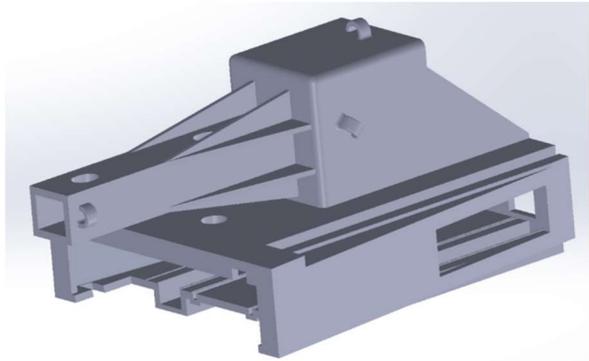
Fonte: Elaborado pelo autor

Figura 39 - Desenho peça 'Suporte Eixo Fonte'



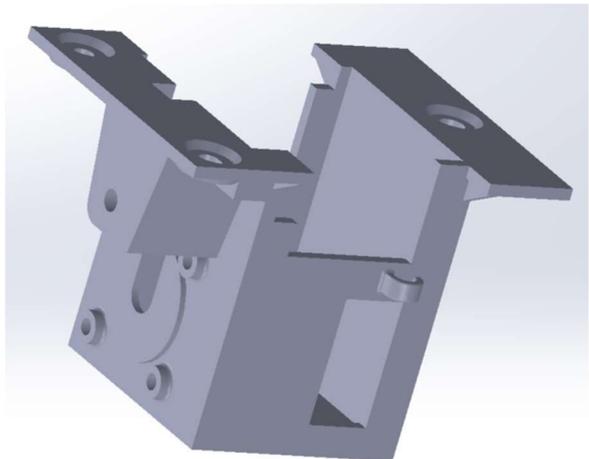
Fonte: Elaborado pelo autor

Figura 40 - Desenho peça 'Suporte Motor Angular'



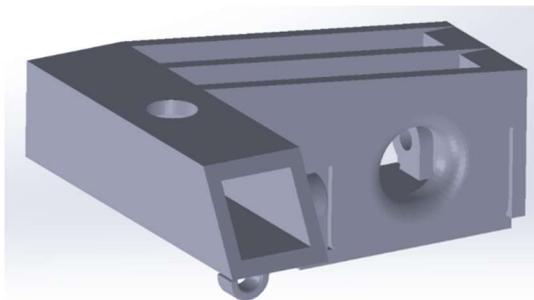
Fonte: Elaborado pelo autor

Figura 41 - Desenho peça 'Suporte Motor Linear'



Fonte: Elaborado pelo autor

Figura 42 - Desenho peça 'Suporte Perfil L'

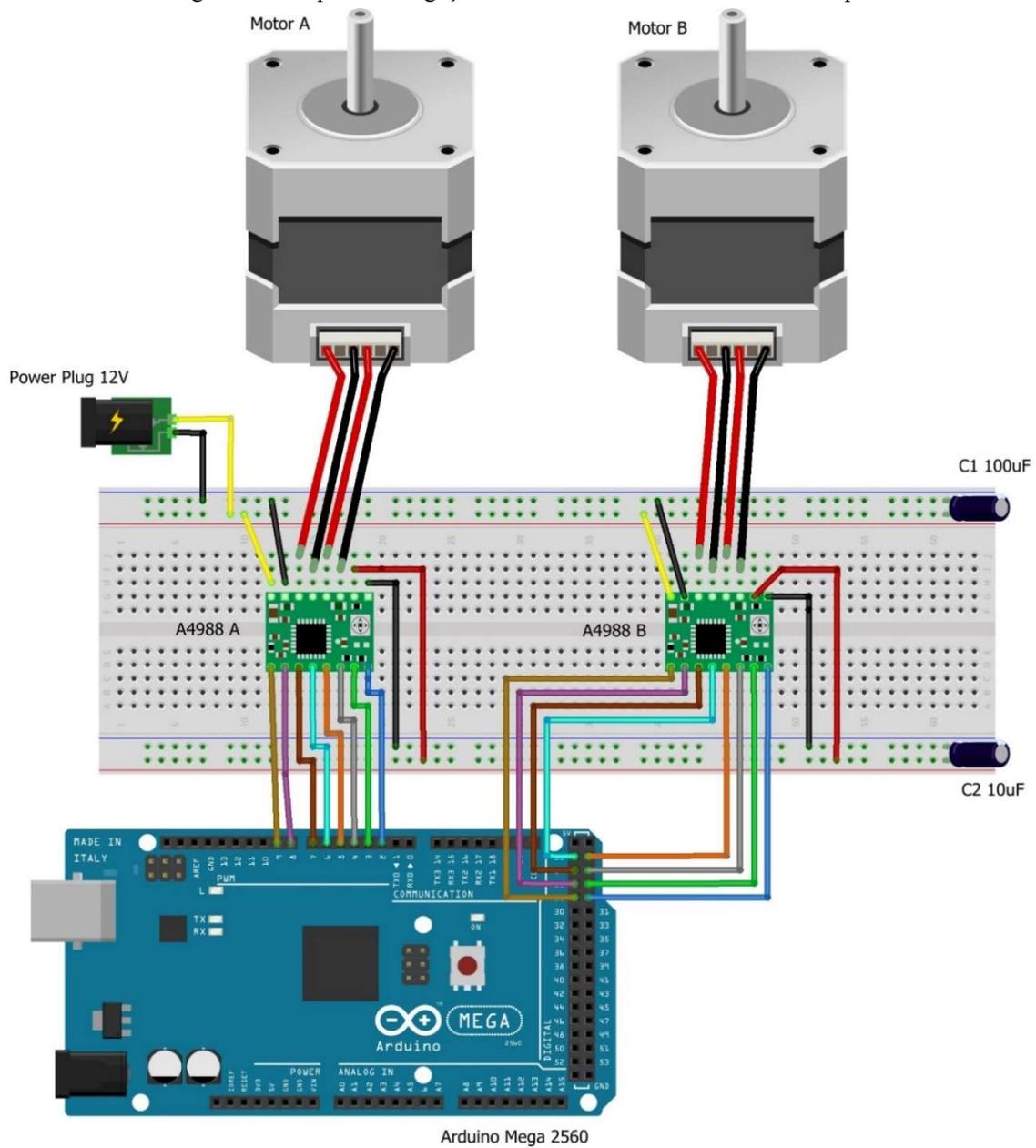


Fonte: Elaborado pelo autor

### 3.4 ESQUEMA DE LIGAÇÃO DOS MOTORES DE PASSO

Os dois motores de passo são controlados por uma mesma placa Arduino Mega 2560, juntamente com dois drivers A4988 (um para cada motor). Nas Figuras 43 e 44 estão dois esquemas de ligação dos motores de passo ao Arduino e aos drivers.

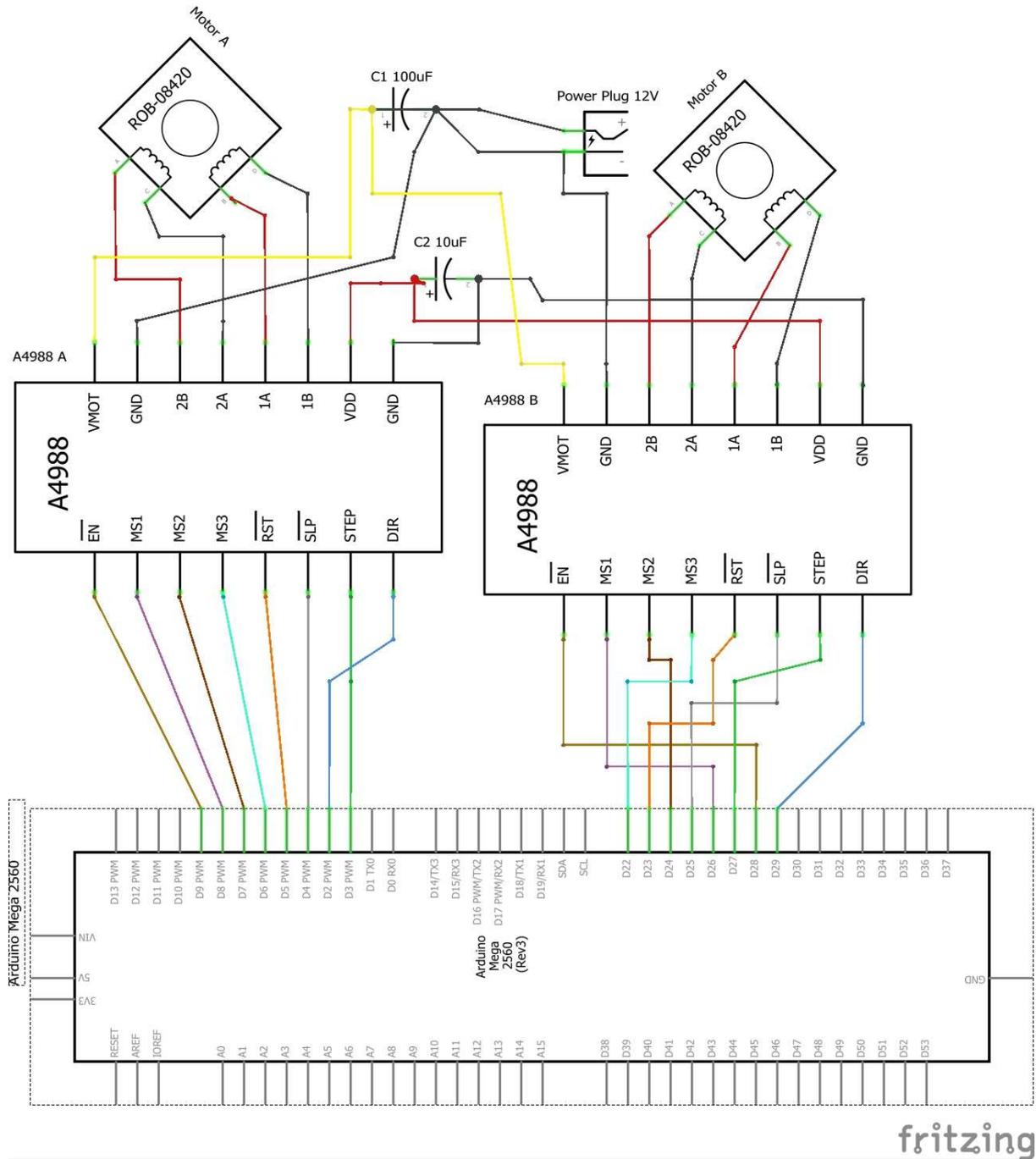
Figura 43 - Esquema de ligação Arduino/Drivers A4988/Motores de passo



fritzing

Fonte: Elaborado pelo autor

Figura 44 - Esquema de ligação com descrição dos pinos do Arduino e Drivers A4988



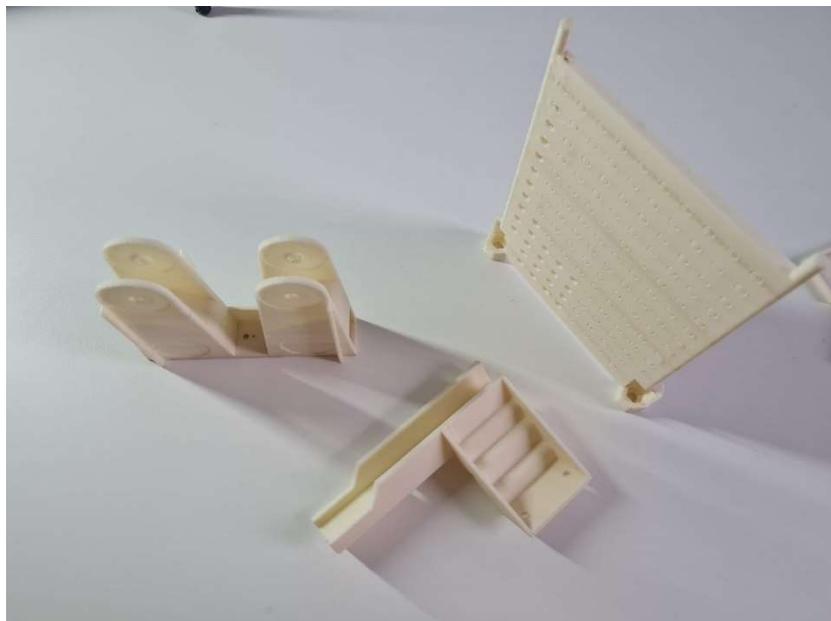
Fonte: Elaborado pelo autor

### 3.5 PEÇAS FABRICADAS NA IMPRESSORA 3D

A impressora 3D utilizada para a fabricação dessas peças foi a GTMax3D, modelo A1V1. Utilizou-se filamentos de material ABS. Cada peça levou entre 4 e 36 horas para ser fabricada, dependendo da complexidade da geometria de cada uma.

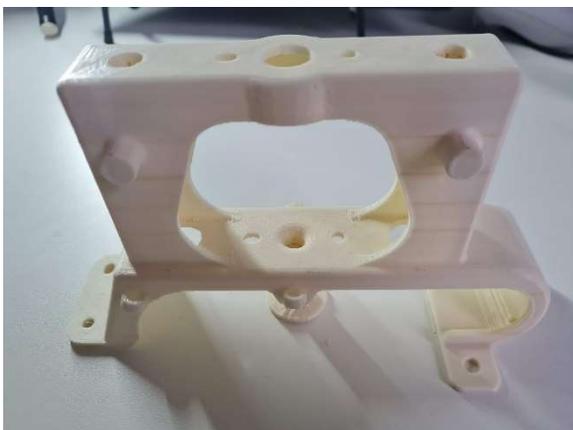
As peças impressas ficaram com ótimo acabamento, tendo medidas fiéis às do projeto. As Figuras 45, 46, 47, 48, 49, 50 e 51 mostram fotos das peças já impressas.

Figura 45 - Peças impressas



Fonte: Fotografado pelo autor

Figura 46 - Peças impressas



Fonte: Fotografado pelo autor

Figura 47 - Peças impressas



Fonte: Fotografado pelo autor

Figura 48 - Peças impressas



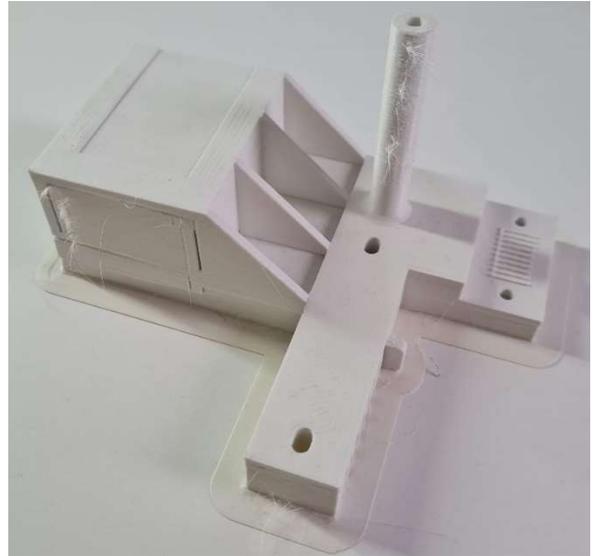
Fonte: Fotografado pelo autor

Figura 49 - Peças impressas



Fonte: Fotografado pelo autor

Figura 50 - Peças impressas



Fonte: Fotografado pelo autor

Figura 51 - Peças impressas



Fonte: Fotografado pelo autor

### 3.6 MONTAGEM DO SISTEMA

O sistema foi montado em uma bancada preparada para comportar toda a amplitude do movimento. A montagem completa foi realizada ao longo de 4 dias. A Figura 52 mostra a foto do sistema montado.

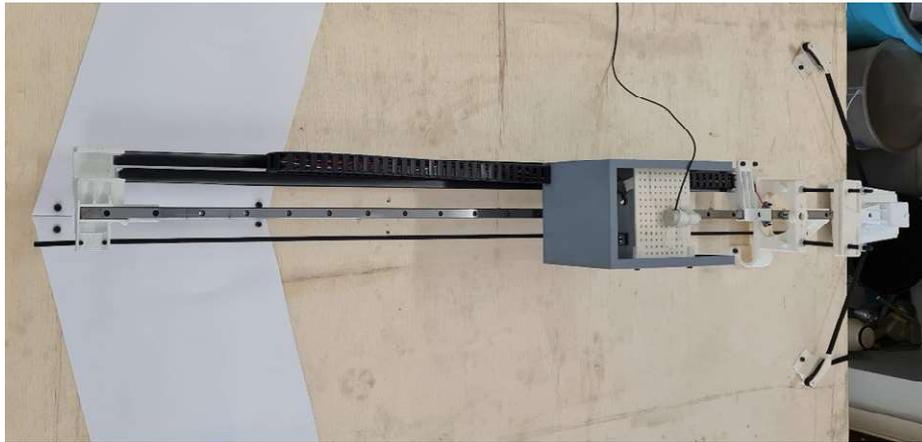
Figura 52 - Foto do sistema montado (isométrica)



Fonte: Fotografado pelo autor

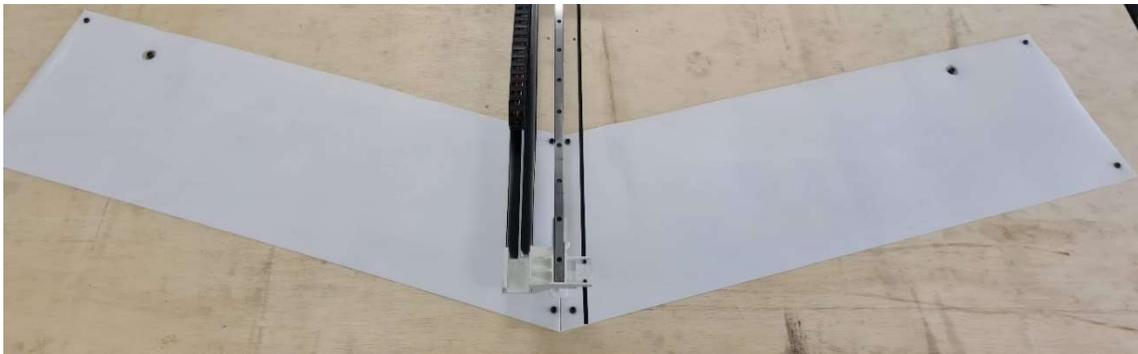
Tanto o suporte do detector quanto o suporte da fonte possuem regulagem manual de altura. O suporte do detector é regulado através de dentes que o posicionam dentro do carro que faz o deslocamento. O suporte da fonte é regulado através de um conjunto de roscas e parafusos.

Figura 53 - Foto sistema montado (planta)



Fonte: Fotografado pelo autor

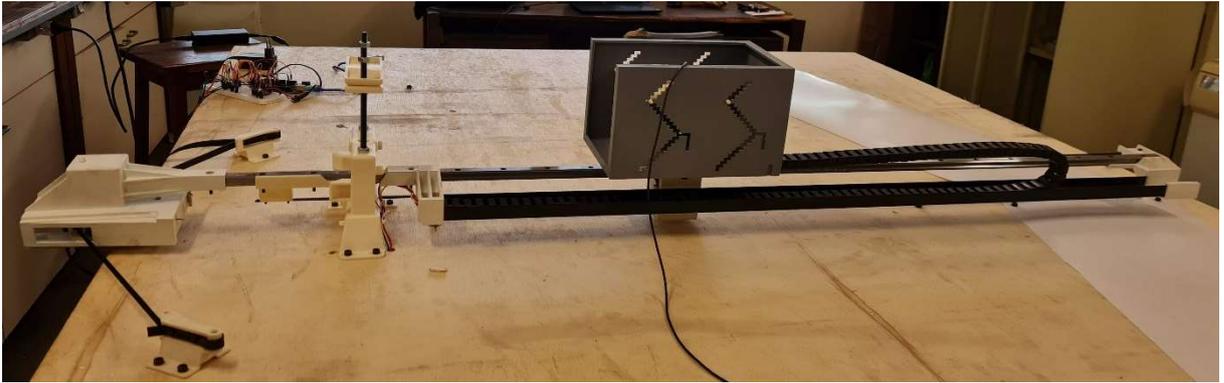
Figura 54 - Foto do detalhe da pista de teflon



Fonte: Fotografado pelo autor

A extremidade da guia linear é apoiada por uma haste em uma pista de teflon. A haste também tem uma peça de teflon. Quando o movimento linear é executado, a peça de teflon da haste desliza na pista de teflon com um baixo atrito, permitindo o movimento com um baixo atrito enquanto sustenta a guia.

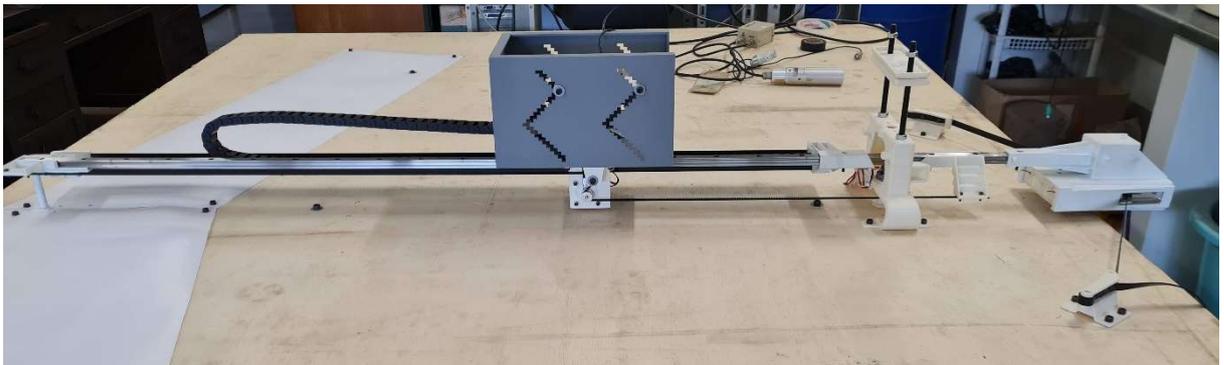
Figura 55 - Foto do sistema montado (vista lateral direita)



Fonte: Fotografado pelo autor

A esteira porta-cabos protege e sustenta os cabos que ligam o motor do deslocamento linear. Os cabos passam por dentro da esteira até chegarem ao fim da guia linear, onde são depois conectados ao Arduino.

Figura 56 - Foto do sistema montado (vista lateral esquerda)



Fonte: Fotografado pelo autor

Figura 57 - Foto do detalhe do subsistema do motor angular, eixo de giro e suporte da fonte



Fonte: Fotografado pelo autor

Figura 58 - Foto do subsistema do motor linear e carro de suporte do detector

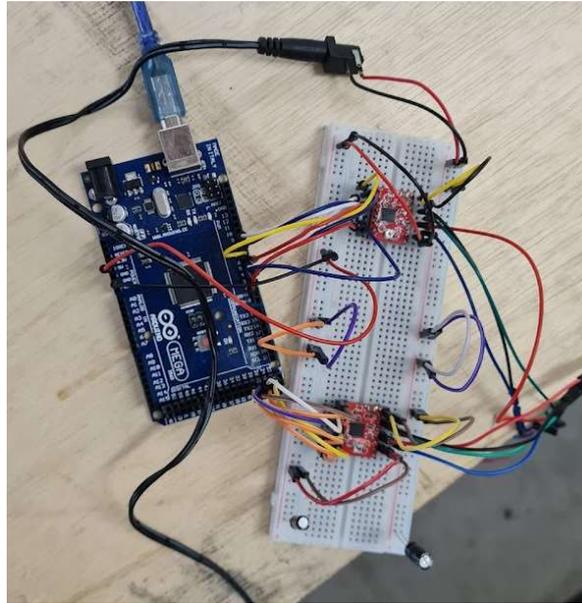


Fonte: Fotografado pelo autor

### 3.7 ELETRÔNICA E PROGRAMAÇÃO

Conforme o esquema de ligação apresentado na Figura 43, foi feita a ligação elétrica do Arduino, drivers A4988 e motores de passo. A Figura 59 mostra a ligação feita.

Figura 59 - Foto da ligação elétrica Arduino/Drivers A4988/Motores de passo



Fonte: Fotografado pelo autor

Após feita a ligação dos componentes, foi necessário realizar o ajuste de corrente dos drivers A4988, de acordo com a equação 2:

$$V_{ref} = I_{max} \times 8 \times RS \quad (2)$$

No caso desta montagem, os valores da corrente máxima dos motores e dos resistores dos drivers são:

$$I_{max} = 1,7A$$

$$RS = 0,1\Omega$$

Portanto o valor da tensão de referência que teve de ser ajustada no trimpot dos drivers A4988 foi de:

$$V_{ref} = 1,7 \times 8 \times 0,1 = 1,36V$$

Para que o Arduino opere corretamente os drivers e motores, foi realizada a programação. O software utilizado para a programação foi o Arduino IDE. O código completo está no APÊNDICE A – Código de programação do Arduino deste trabalho.

Os drivers A4988 tem função de micropasso, podendo ser programado para operar numa resolução de até um dezesseis-avos de passo. O motor que faz o deslocamento linear, chamado no código de programação de motor ‘a’, foi programado para operar em um quarto de passo. Já o motor que faz o deslocamento angular, chamado no código de programação de motor ‘b’, foi programado para operar em um dezesseis avos de passo. Esses modos de passo foram

definidos de modo a aumentar a precisão dos movimentos e tentar compensar possíveis atritos no movimento do sistema.

O código escrito foi testado em operação para que se refinasse os cálculos de distância linear e ângulo que o sistema precisa fazer, de acordo com cada comando que foi dado. Para iniciar uma trajetória, é necessário colocar o sistema num ângulo  $0^\circ$  e a uma distância de 0mm entre o carro que movimenta o detector linearmente e o suporte da fonte. Então, dentro do próprio código, digita-se o ângulo desejado e a distância desejada (quando o sistema está em ângulo  $0^\circ$ ). Quando o ângulo digitado não é  $0^\circ$ , o próprio código faz o cálculo de qual distância o carro que movimenta o detector linearmente precisa deslocar para compensar o ângulo e simular a fonte infinita. Com estes parâmetros definidos, basta enviar o código para o Arduino e a operação é iniciada, primeiro fazendo o deslocamento linear e, em seguida, o deslocamento angular.

## 4 RESULTADOS

### 4.1 TESTE DE POSICIONAMENTO DO SISTEMA

Com o sistema completamente montado e programado, realizou-se testes para verificar se o sistema estava operando com a precisão desejada.

#### 4.1.1 Testes de posicionamento angular

Todos os testes foram iniciados com o sistema no ângulo 0°. Os resultados dos testes são apresentados nas tabelas a seguir.

Tabela 6 - Medições para ângulo 15°

Ângulo desejado	Ângulo medido	Varição do ângulo	Varição percentual
15	14,7	-0,3	-1,82%
15	14,9	-0,1	-0,83%
15	14,9	-0,1	-0,50%

<b>MÉDIA</b>	<b>14,8</b>
<b>DESVIO PADRÃO</b>	<b>0,10088</b>

Resultado da medição:  $\delta = 14,8^\circ \pm 0,10088$

Tabela 7 - Medições para ângulo -15°

Ângulo desejado	Ângulo medido	Varição do ângulo	Varição percentual
-15	-14,9	0,1	-0,50%
-15	-15,0	0,0	-0,17%
-15	-14,9	0,1	-0,50%

<b>MÉDIA</b>	<b>-14,9</b>
<b>DESVIO PADRÃO</b>	<b>0,02800</b>

Resultado da medição:  $\delta = -14,9^\circ \pm 0,02800$

Tabela 8 - Medições para ângulo 30°

Ângulo desejado	Ângulo medido	Varição do ângulo	Varição percentual
30	29,9	-0,1	-0,45%
30	30,2	0,2	0,63%
30	29,4	-0,6	-1,92%

<b>MÉDIA</b>	<b>29,8</b>
<b>DESVIO PADRÃO</b>	<b>0,37921</b>

Resultado da medição:  $\delta = 29,8^\circ \pm 0,37921$

Tabela 9 - Medições para ângulo -30°

Ângulo desejado	Ângulo medido	Varição do ângulo	Varição percentual
-30	-30,1	-0,1	0,45%
-30	-30,3	-0,3	0,98%
-30	-30,2	-0,2	0,63%

<b>MÉDIA</b>	<b>-30,2</b>
<b>DESVIO PADRÃO</b>	<b>0,08282</b>

Resultado da medição:  $\delta = -30,2^\circ \pm 0,08282$

Tabela 10 - Medições para ângulo 40°

Ângulo desejado	Ângulo medido	Varição do ângulo	Varição percentual
40	39,9	-0,1	-0,17%
40	40,1	0,1	0,29%
40	39,7	-0,3	-0,63%

<b>MÉDIA</b>	<b>39,9</b>
<b>DESVIO PADRÃO</b>	<b>0,18329</b>

Resultado da medição:  $\delta = 39,9^\circ \pm 0,18329$

Tabela 11 - Medições para ângulo  $-40^\circ$ 

Ângulo desejado	Ângulo medido	Varição do ângulo	Varição percentual
-40	-40,0	0,0	-0,02%
-40	-40,1	-0,1	0,13%
-40	-40,0	0,0	-0,02%

<b>MÉDIA</b>	<b>-40,0</b>
<b>DESVIO PADRÃO</b>	<b>0,03532</b>

Resultado da medição:  $\delta = -40,0^\circ \pm 0,03532$

Todos os testes de posicionamento angular feitos tiveram resultados melhores que o esperado. A maior variação medida entre o ângulo desejado e o ângulo medido foi de  $0,6^\circ$ . No entanto, com o cálculo das médias e desvios, nota-se que a incerteza está bem baixa, sendo  $0,37921^\circ$  o maior desvio padrão (encontrado no teste para  $30^\circ$ ) e o menor sendo  $0,03532^\circ$  (encontrado no teste para  $-40^\circ$ ).

#### 4.1.2 Testes de posicionamento linear

Todos os testes foram iniciados com o carro do suporte do detector a uma distância linear de 0 mm em relação ao suporte da fonte. As medições foram feitas com instrumentos com precisão de 1 mm. Os resultados dos testes são apresentados nas tabelas a seguir.

Tabela 12 - Medições para deslocamento de 100 mm

Deslocamento desejado em mm	Deslocamento medido em mm	Varição do deslocamento	Varição percentual
100	100	0	0,00%
100	100	0	0,00%
100	100	0	0,00%

<b>MÉDIA</b>	<b>100</b>
--------------	------------

Resultado da medição:  $\delta = 100,0 \pm 0,5 \text{ mm}$

Tabela 13 - Medições para deslocamento de 155 mm

Deslocamento desejado em mm	Deslocamento medido em mm	Varição do deslocamento	Varição percentual
155	155	0	0,00%
155	155	0	0,00%
155	155	0	0,00%

<b>MÉDIA</b>	<b>155</b>
--------------	------------

Resultado da medição:  $\delta = 155,0 \pm 0,5 \text{ mm}$

Tabela 14 - Medições para deslocamento de 259 mm

Deslocamento desejado em mm	Deslocamento medido em mm	Varição do deslocamento	Varição percentual
259	259	0	0,00%
259	259	0	0,00%
259	259	0	0,00%

<b>MÉDIA</b>	<b>259</b>
--------------	------------

Resultado da medição:  $\delta = 259,0 \pm 0,5 \text{ mm}$

Tabela 15 - Medições para deslocamento de 388 mm

Deslocamento desejado em mm	Deslocamento medido em mm	Varição do deslocamento	Varição percentual
388	388	0	0,00%
388	388	0	0,00%
388	388	0	0,00%

<b>MÉDIA</b>	<b>388</b>
--------------	------------

Resultado da medição:  $\delta = 388,0 \pm 0,5 \text{ mm}$

Todas as medições realizadas nos testes de posicionamento linear tiveram uma precisão excelente. Em absolutamente todos os testes, as medições corresponderam exatamente a distância que havia sido programada para o sistema realizar.

Conclui-se então que o sistema está adequado para realizar o posicionamento fonte-detector com precisão esperada.

#### 4.1.3 Medições com fonte Cobalto e detector Geiger-Muller

Apenas a fim de pôr o sistema em uso para simular uma fonte infinita a partir de uma fonte puntual, foram feitas medições de fonte de Cobalto utilizando um detector do tipo Geiger-Muller. Para evitar erros de leitura que poderiam ser ocasionados pelo tempo morto do detector, as medições foram feitas em distâncias relativamente grandes, a partir de 300 mm, entre a fonte e detector. Para ângulos diferentes de  $0^\circ$  a distância fonte-detector foi corrigida automaticamente pelo código do Arduino para simular a fonte infinita plana. A Tabela 16 apresenta os resultados.

Tabela 16 - Contagem de pulsos do detector em diferentes posições do sistema

Ângulo desejado em graus	Ângulo medido em graus	Distância em mm da fonte infinita simulada	Distância em mm fonte-detector (já corrigida pelo sistema)	Tempo de aquisição em seg	Contagem de pulsos do detector						
					Medição 1	Medição 2	Medição 3	Medição 4	Medição 5	MÉDIA	DESVIO PADRÃO
0,0	<b>0,0</b>	300	<b>300</b>	50	927	858	824	914	837	<b>872</b>	<b>46</b>
30,0	<b>30,0</b>	300	<b>346</b>	50	820	806	849	814	820	<b>822</b>	<b>16</b>
-30,0	<b>-30,3</b>	300	<b>346</b>	50	840	843	827	795	798	<b>821</b>	<b>23</b>
0,0	<b>0,0</b>	350	<b>350</b>	80	1127	1124	1172	1151	1100	<b>1135</b>	<b>28</b>
15,0	<b>14,8</b>	350	<b>362</b>	80	1140	1090	1100	1101	1104	<b>1107</b>	<b>19</b>
-15,0	<b>-15,0</b>	350	<b>362</b>	80	1100	1140	1075	1108	1119	<b>1108</b>	<b>24</b>

Novamente, os posicionamentos feitos pelo sistema tiveram uma ótima precisão, conforme o esperado.

Também dentro do esperado estão os resultados das contagens de pulsos em diferentes ângulos. No primeiro teste, simulou-se uma fonte plana infinita que estaria posicionada a 300 mm do detector. O tempo de aquisição do detector foi configurado em 50 segundos. As

primeiras medições foram realizadas em ângulo  $0^\circ$ , com distância entre fonte e detector de 300 mm. Em média, o detector registrou 872 contagens. A seguir, foram realizadas medidas nos ângulos de  $30^\circ$  e  $-30^\circ$ . Para que fosse simulado a fonte plana infinita, o sistema adequou a distância fonte-detector para 346 mm (que seria a distância que o feixe neste ângulo percorreria). O detector registrou em média 822 contagens no ângulo  $30^\circ$  e 821 contagens no ângulo de  $-30^\circ$ . Os resultados estão de acordo com o esperado, já que em ângulo  $0^\circ$ , a distância fonte-detector é menor, e, portanto, era esperado uma contagem de pulsos maior; enquanto nas posições de ângulo  $30^\circ$  e  $-30^\circ$ , as distâncias são maiores e, portanto, era esperado uma contagem de pulsos menor em relação ao ângulo  $0^\circ$ . Também era esperado que as contagens das posições de ângulo  $30^\circ$  e  $-30^\circ$  ficassem próximas, o que também pode ser observado.

No segundo teste, os resultados também ficaram dentro do esperado. Neste teste, simulou-se uma fonte plana infinita posicionada a 350 mm do detector. O tempo de aquisição neste segundo teste foi configurado em 80 segundos. As primeiras medições foram realizadas em ângulo  $0^\circ$ , com distância entre fonte e detector de 350 mm. Em média, o detector registrou 1135 contagens. A seguir, foram realizadas medidas nos ângulos de  $15^\circ$  e  $-15^\circ$ . Para que fosse simulado a fonte plana infinita, o sistema adequou a distância fonte-detector para 362 mm (que seria a distância que o feixe neste ângulo percorreria). O detector registrou em média 1107 contagens no ângulo  $15^\circ$  e 1108 contagens no ângulo de  $-15^\circ$ . Como no primeiro teste, os resultados ficaram dentro do esperado, com a contagem de pulsos no ângulo  $0^\circ$  sendo maior do que nos ângulos de  $15^\circ$  e  $-15^\circ$ , que por sua vez foram bastante equivalentes.

## 5 CONCLUSÃO

Simulações computacionais com o código MCNP são feitas por vários pesquisadores (e com diferentes objetivos) e precisam ser validadas, confrontando-se os resultados destas simulações com dados experimentais reais. A obtenção de dados experimentais reais para fontes grandes (chamadas aqui de infinitas) é difícil, devido à dificuldade de fabricação e manuseio de fontes de referência de maior tamanho.

Este trabalho teve como objetivo, criar um protótipo de sistema de posicionamento automatizado para realizar simulação de uma fonte infinita a partir de uma fonte puntual de laboratório. O protótipo foi idealizado, projetado, fabricado, montado, programado e testado, apresentando ótimos resultados, com grande precisão nos posicionamentos.

Este protótipo pode ser usado para se obter dados experimentais reais para serem usados na validação de resultados de simulação computacional. Os resultados dos testes de posicionamento e medição da radiação que foram obtidos, mostram que o sistema tem uma boa confiabilidade e que é possível simular uma fonte infinita a partir de uma fonte puntual.

Como a precisão dos deslocamentos lineares foi bastante alta, também é possível utilizar o sistema para fazer uma calibração de detectores, tendo assegurada a distância fonte e detector no momento da calibração.

Trabalhos futuros poderão fazer mudanças no sistema para deixá-lo mais preciso, seguro e com maior aplicabilidade. Uma sugestão a ser implementada futuramente é a de implementar sensores de fim de curso e de aproximação, para que o sistema possa mudar de qualquer posição e ir para uma outra, sem a necessidade de deixar o sistema inicialmente em ângulo e distância 0. É possível também fazer futuras modificações nos suportes que regulam a altura do detector e da fonte, para que essas regulagens de altura também sejam automatizadas. O uso dos sensores também pode deixar o sistema menos suscetível a erros de operação que poderiam fazer com que o sistema forçasse a se mover além do que o projeto do sistema permite. Um erro de operação desse tipo, pode ocasionar a quebra de um ou mais componentes do sistema. Fica também sugerido uma automatização ainda maior do sistema, de forma a ser possível programá-lo para fazer diversas aquisições de dados em diferentes posições com apenas um comando. Em conjunto com uma programação que registra as medições do detector em uma planilha, seria possível deixar o sistema fazendo um grande volume de aquisições sem a necessidade de uma pessoa operar a programação a cada novo posicionamento.

## REFERÊNCIAS

3DPRINTING.COM. What is 3D Printing? **3DPrinting.com**. Disponível em: <https://3dprinting.com/what-is-3d-printing/>. Acesso em: 2021.

ALVARENGA, T. S. Estimativa computacional da radiação espalhada no laboratório de calibração com nêutrons do IPEN. **Brazilian Journal of Radiation Sciences**, 2017.

ARDUFOCUS. A4988 motor current tuning. **ArduFocus**. Disponível em: <https://ardufocus.com/howto/a4988-motor-current-tuning/>. Acesso em: 2021.

AVALCANTE, M. A.; TAVOLARO, C. R. C.; MOLISANI, E. Física com Arduino para iniciantes. **Revista Brasileira de Ensino de Física**, v. 33, p. 4503, 2011.

BAKSHI, A. K. **Measurement of the output of ISO recommended beat sources with**. Índia: Elsevier, 2013.

BALUTA, G. Microstepping mode for stepper motor control. **2007 International Symposium on Signals, Circuits and Systems**, p. 1-14, 2007.

BODSON, M. E. A. High-performance nonlinear feedback control of a permanent magnet stepper motor. **IEEE Transactions on Control Systems Technology**, v. 1, p. 5-14, 1993.

CNEN - COMISSÃO NACIONAL DE ENERGIA NUCLEAR. **Norma CNEN NN 6.04 - REQUISITOS DE SEGURANÇA E PROTEÇÃO RADIOLÓGICA PARA SERVIÇOS DE RADIOGRAFIA INDUSTRIAL**. Brasil: CNEN, 2013.

CNEN – COMISSÃO NACIONAL DE ENERGIA NUCLEAR, NORMA. **CNEN NN 3.05 - REQUISITOS DE SEGURANÇA E PROTEÇÃO RADIOLÓGICA PARA SERVIÇOS DE MEDICINA NUCLEAR**. Brasil: CNEN, 2013.

DIAS, S. K. **Desenvolvimento de uma Câmara de Extrapolação como instrumento de Referência para Dosimetria Beta**. Instituto de Pesquisas Energéticas e Nucleares. Belo Horizonte, p. 128. 1996.

ELETROGATE. Tudo sobre DRIVER A4988 e Motor de Passo > Usando o Arduino. **Blog Eletrogate**, 2018. Disponível em: <https://blog.eletrogate.com/driver-a4988-motor-de-passo-usando-o-arduino/>. Acesso em: 2021.

EQUIPCO SERVICES. Disponível em:  
<https://www.equipcoservices.com/support/tutorials/introduction-to-radiation-monitors/>.

Acesso em: 27 maio 2020.

EVANS, M.; NOBLE, J.; HOCHENBAUM, J. **Arduino em ação**. [S.l.]: Novatec Editora, 2013.

GRIDLING, G.; WEISS, B. **Introduction to microcontrollers**. Vienna University of Technology Institute of Computer Engineering Embedded Computing Systems Group. [S.l.]. 2007.

IAEA – INTERNATIONAL ATOMIC ENERGY AGENCY. **Calibration Of Radiation Protection Monitoring**. Viena: IAEA, 2000.

INMETRO – INSTITUTO NACIONAL DE METROLOGIA, QUALIDADE E TECNOLOGIA. **Vocabulário Internacional de Metrologia**. 1<sup>a</sup>. ed. Rio de Janeiro: INMETRO, 2012.

KNOLL, G. P. **Radiation Detection and Measurement**. 4th Edition. ed. Nova Iorque: John Wiley & Sons, 2010.

MANFRINI, F. A. L. **Projeto e construção de um sistema de posicionamento microcontrolado e de uma câmara de ionização à volume variável**. Escola de Engenharia da Universidade Federal de Minas Gerais. Belo Horizonte, p. 67. 2007.

MCROBERTS, M. **Arduino básico**. [S.l.]: Novatec Editora, 2018.

MONK, S. **Programação com Arduino: começando com Sketches**. [S.l.]: Bookman Editora, 2013.

WOODFORD, C. How do 3D printers work? **Explain That Stuff**, 2021. Disponível em: <https://www.explainthatstuff.com/how-3d-printers-work.html>. Acesso em: 2021.

## APÊNDICE A – Código de programação do Arduino

```

// Controle de Motor de Passo com Modulo driver A4988
//
// Modulo A4988 / Motor de Passo Bipolar / Arduino MEGA / IDE 1.8.15
//
// Motor "a" é o do carrinho e o "b" o da rotação
//
// Definições das Portas Digitais do Arduino

int RSTa = 5;           // Porta digital D05 - reset do A4988 motor 'a'
int SLPa = 4;          // Porta digital D04 - dormir (sleep) A4988 motor 'a'
int ENAa = 9;          // Porta digital D09 - ativa (enable) A4988 motor 'a'
int MS1a = 8;          // Porta digital D08 - MS1 do A4988 motor 'a'
int MS2a = 7;          // Porta digital D07 - MS2 do A4988 motor 'a'
int MS3a = 6;          // Porta digital D06 - MS3 do A4988 motor 'a'
int DIRa = 2;          // Porta digital D02 - direção (direction) do A4988 motor 'a'
int STPa = 3;          // Porta digital D03 - passo(step) do A4988 motor 'a'

int MeioPeriodoa = 1000; // MeioPeriodo do pulso STEP motor 'a' em microsegundos F= 1/T =
1/2000 uS = 500 Hz
float PPSa = 0;         // Pulsos por segundo motor 'a'
boolean sentidoa = true; // Variavel de sentido motor 'a'
long PPRa = 200;        // Número de passos por volta motor 'a'
long Pulsosa;           // Pulsos para o driver do motor motor 'a'
float Voltasa;          // voltas do motor motor 'a'
float RPMa;             // Rotacoes por minuto motor 'a'
int distancia;          // Distancia desejada entre fonte e detector em mm no ângulo 0
float deslocdetector;   // Deslocamento do detector em relação a frente do carrinho
float distanciapercorrida; // Distancia percorrida pelo carrinho já corrigida considerando
posição do detector em relação ao carrinho e posição no rasgo 2
float hipotenusa;       // Distancia da hipotenusa em mm

int RSTb = 23;          // Porta digital D23 - reset do A4988 motor 'b'
int SLPb = 25;          // Porta digital D25 - dormir (sleep) A4988 motor 'b'
int ENAb = 28;          // Porta digital D28 - ativa (enable) A4988 motor 'b'
int MS1b = 26;          // Porta digital D26 - MS1 do A4988 motor 'b'
int MS2b = 24;          // Porta digital D24 - MS2 do A4988 motor 'b'
int MS3b = 22;          // Porta digital D22 - MS3 do A4988 motor 'b'
int DIRb = 29;          // Porta digital D29 - direção (direction) do A4988 motor 'b'
int STPb = 27;          // Porta digital D27 - passo(step) do A4988 motor 'b'

int MeioPeriodob = 1000; // MeioPeriodo do pulso STEP motor 'b' em microsegundos F= 1/T =
1/2000 uS = 500 Hz
float PPSb = 0;         // Pulsos por segundo motor 'b'
boolean sentidob = true; // Variavel de sentido motor 'b'
long PPRb = 200;        // Número de passos por volta motor 'b'
long Pulsosb;           // Pulsos para o driver do motor motor 'b'
float Voltasb;          // voltas do motor motor 'b'
float RPMb;             // Rotacoes por minuto motor 'b'
float angulo;           // Ângulo desejado em graus entre -44 e 44 graus
float angrad;           // ângulo em radianos

void setup()
{
  Serial.begin(9600);

  pinMode(ENAa, OUTPUT); // Configura Porta D09 como saída motor 'a'
  pinMode(MS1a, OUTPUT); // Configura Porta D08 como saída motor 'a'
  pinMode(MS2a, OUTPUT); // Configura Porta D07 como saída motor 'a'
  pinMode(MS3a, OUTPUT); // Configura Porta D06 como saída motor 'a'
  pinMode(DIRa, OUTPUT); // Configura Porta D02 como saída motor 'a'
  pinMode(STPa, OUTPUT); // Configura Porta D03 como saída motor 'a'
  disa_A4988a();         // Desativa o chip A4988 motor 'a'

  pinMode(RSTa, OUTPUT); // Configura Porta D05 como saída motor 'a'
  pinMode(SLPa, OUTPUT); // Configura Porta D04 como saída motor 'a'
  digitalWrite(SLPa, HIGH); // Desativa modo sleep do A4988 motor 'a'

```

```

rst_A4988a();          // Reseta o chip A4988 motor 'a'
//ena_A4988a();       // Ativa o chip A4988 motor 'a'

pinMode(ENAb, OUTPUT); // Configura Porta D07 como saída motor 'b'
pinMode(MS1b, OUTPUT); // Configura Porta D04 como saída motor 'b'
pinMode(MS2b, OUTPUT); // Configura Porta D05 como saída motor 'b'
pinMode(MS3b, OUTPUT); // Configura Porta D06 como saída motor 'b'
pinMode(DIRb, OUTPUT); // Configura Porta D03 como saída motor 'b'
pinMode(STPb, OUTPUT); // Configura Porta D02 como saída motor 'b'
disa_A4988b();        // Desativa o chip A4988 motor 'b'

pinMode(RSTb, OUTPUT); // Configura Porta D08 como saída motor 'b'
pinMode(SLPb, OUTPUT); // Configura Porta D09 como saída motor 'b'
digitalWrite(SLPb, HIGH); // Desativa modo sleep do A4988 motor 'b'
rst_A4988b();          // Reseta o chip A4988 motor 'b'
//ena_A4988b();       // Ativa o chip A4988 motor 'b'
}

void rst_A4988a()
{
  digitalWrite(RSTa, LOW); // Realiza o reset do A4988 motor 'a'
  delay (100);             // Atraso de 10 milisegundos
  digitalWrite(RSTa, HIGH); // Libera o reset do A4988 motor 'a'
  delay (100);             // Atraso de 10 milisegundos
}

void rst_A4988b()
{
  digitalWrite(RSTb, LOW); // Realiza o reset do A4988 motor 'b'
  delay (100);             // Atraso de 10 milisegundos
  digitalWrite(RSTb, HIGH); // Libera o reset do A4988 motor 'b'
  delay (100);             // Atraso de 10 milisegundos
}

void disa_A4988a()
{
  digitalWrite(ENaa, HIGH); // Desativa o chip A4988 motor 'a'
  delay (100);             // Atraso de 10 milisegundos
}

void disa_A4988b()
{
  digitalWrite(ENAb, HIGH); // Desativa o chip A4988 motor 'b'
  delay (100);             // Atraso de 10 milisegundos
}

void ena_A4988a()
{
  digitalWrite(ENaa, LOW); // Ativa o chip A4988 motor 'a'
  delay (100);             // Atraso de 10 milisegundos
}

void ena_A4988b()
{
  digitalWrite(ENAb, LOW); // Ativa o chip A4988 motor 'b'
  delay (100);             // Atraso de 10 milisegundos
}

void HORa() // Configura o sentido de rotação do Motor motor 'a'
{
  Serial.println(" Sentido horário no motor 'a'");
  digitalWrite(DIRa, HIGH); // Configura o sentido HORÁRIO motor 'a'
}

void HORb() // Configura o sentido de rotação do Motor motor 'b'
{
  Serial.println(" Sentido horário no motor 'b'");
  digitalWrite(DIRb, HIGH); // Configura o sentido HORÁRIO motor 'b'
}

void AHRb() // Configura o sentido de rotação do Motor motor 'b'
{
  Serial.println(" Sentido anti-horário no motor 'b'");
  digitalWrite(DIRb, LOW); // Configura o sentido ANTI-HORÁRIO motor 'b'
}

```

```

void PASSOa() // Pulso do passo do Motor motor 'a'
{
    digitalWrite(STPa, LOW); // Pulso nível baixo motor 'a'
    delayMicroseconds (MeioPeriodoa); // MeioPeríodo de X microssegundos motor 'a'
    digitalWrite(STPa, HIGH); // Pulso nível alto motor 'a'
    delayMicroseconds (MeioPeriodoa); // MeioPeríodo de X microssegundos motor 'a'
}

void PASSOb() // Pulso do passo do Motor motor 'b'
{
    digitalWrite(STPb, LOW); // Pulso nível baixo motor 'b'
    delayMicroseconds (MeioPeriodob); // MeioPeríodo de X microssegundos motor 'b'
    digitalWrite(STPb, HIGH); // Pulso nível alto motor 'b'
    delayMicroseconds (MeioPeriodob); // MeioPeríodo de X microssegundos motor 'b'
}

void FREQUENCIAa() // calcula Pulsos, PPS e RPM motor 'a'
{
    Pulsosa = PPRa * Voltasa; // Quantidade total de Pulsos (PPR = pulsos por volta)
    motor 'a'
    PPSa = 1000000 / (2 * MeioPeriodoa); // Frequencia Pulsos por segundo motor 'a'
    RPMa = (PPSa * 60) / PPRa; // Calculo do RPM motor 'a'
}

void FREQUENCIAb() // calcula Pulsos, PPS e RPM motor 'b'
{
    if (Voltasb < 0){
        Pulsosb = -1 * PPRb * Voltasb; // Quantidade total de Pulsos (PPR = pulsos por volta)
    }
    else {
        Pulsosb = PPRb * Voltasb;
    }
    PPSb = 1000000 / (2 * MeioPeriodob); // Frequencia Pulsos por segundo motor 'b'
    RPMb = (PPSb * 60) / PPRb; // Calculo do RPM motor 'b'
}

void P1_4a()
{
    PPRa = 800; // PPR = pulsos por volta motor 'a'
    digitalWrite(MS1a, LOW); // Configura modo Micro Passo 1/4 motor 'a'
    digitalWrite(MS2a, HIGH);
    digitalWrite(MS3a, LOW);
}

void P1_16b()
{
    Serial.println(" Micro-passo 1/16 motor 'b' PPR = 3200 ");
    PPRb = 3200; // PPR = pulsos por volta motor 'b'
    digitalWrite(MS1b, HIGH); // Configura modo Micro Passo 1/16 motor 'b'
    digitalWrite(MS2b, HIGH);
    digitalWrite(MS3b, HIGH);
}

void CalcDist()
{
    angrad = angulo * 3.1415 / 180; // Calcula o ângulo em radianos
    hipotenusa = distancia / cos(angrad); // Calcula a hipotenusa
    distanciapercorrida = hipotenusa - deslodeDetector - 23; // Corrige a distancia que o
    carrinho precisa andar
    Voltasa = distanciapercorrida / 40; // Calcula quantas voltas o motor 'a' tem que dar
    para essa hipotenusa considerando que 1 volta desloca 40mm
    Voltasb = angulo / 7.65; // Calcula quantas voltas o motor 'b' tem que dar
    para chegar nesse ângulo considerando que 1 volta gira 7,65°
}

void TesteMotora()
{
    Print_RPMa (); // Print Voltas, PPS e RPM motor 'a'

    delay (500);
}

```

```

ena_A4988a();
delay (500);

HORa(); // Gira sentido Horario motor 'a'
for (int i = 0; i <= Pulsosa; i++) // Incrementa o Contador motor 'a'
{
  PASSOa(); // Avança um passo no Motor motor 'a'
}
disa_A4988a(); // Desativa o chip A4988 motor 'a'
delay (1000); // Atraso de 1 segundo
}

void TesteMotorb()
{
  Print_RPMb (); // Print Voltas, PPS e RPM motor 'b'

  delay (500);
  ena_A4988b();
  delay (500);

  if (Voltasb > 0){
    HORb(); // Gira sentido Horario motor 'b'
    for (int i = 0; i <= Pulsosb; i++) // Incrementa o Contador motor 'b'
    {
      PASSOb(); // Avança um passo no Motor motor 'b'
    }
    disa_A4988b(); // Desativa o chip A4988 motor 'b'
    delay (1000); // Atraso de 1 segundo
  }
  else {
    AHRb(); // Gira sentido anti-Horario motor 'b'
    for (int i = 0; i <= Pulsosb; i++) // Incrementa o Contador motor 'b'
    {
      PASSOb(); // Avança um passo no Motor motor 'b'
    }
    disa_A4988b(); // Desativa o chip A4988 motor 'b'
    delay (1000); // Atraso de 1 segundo
  }
}

void Print_RPMa ()
{
  FREQUENCIAa(); // calcula Distância, Hipotenusa e RPM motor 'a'
  Serial.print(" Distância no ângulo 0° = ");
  Serial.print(distancia);
  Serial.print("mm");
  Serial.println();
  Serial.print(" Distância da hipotenusa = ");
  Serial.print(hipotenusa);
  Serial.print("mm");
  Serial.println();
  Serial.print(" Distância percorrida pelo carrinho = ");
  Serial.print(distanciapercorrida);
  Serial.print("mm");
  Serial.println();
  Serial.print(" RPM motor 'a' = ");
  Serial.println(RPMa, 2);
}

void Print_RPMb ()
{
  FREQUENCIAb(); // calcula ângulo e RPM motor 'b'
  Serial.print(" Ângulo em relação à origem = ");
  Serial.print(angulo);
  Serial.print("°");
  Serial.println();
  Serial.print(" RPM motor 'b' = ");
  Serial.println(RPMb, 2);
}

void loop()
{
  deslocdetector = 43; // Distancia entre detector que foi posicionado e a frente do carrinho
  distancia = 150; // Definir distancia no angulo 0 em mm
}

```

```
angulo = 15;           // Definir angulo de -44 a 44 graus
CalcDist();           // Programa vai calcular o angulo em radianos e a hipotenusa

Serial.print(" INÍCIO DA OPERAÇÃO ");

Serial.println();
Serial.println();

Pl_4a();              // Define o passo do motor 'a' em 1/4
TesteMotora();       // Inicia teste do motor motor 'a'

Serial.println();

Pl_16b();             // Define o passo do motor 'b' em 1/16
TesteMotorb();       // Inicia teste do motor motor 'b'

Serial.println();

Serial.print (" FIM DA OPERAÇÃO");

while (1){}
}
```