AUTARQUIA ASSOCIADA À UNIVERSIDADE DE SÃO PAULO

Desenvolvimento de um instrumento de movimentação e troca de amostras
de radioisótopos para detectores HPGe - Germânio Hiper-Puro.

ROBERTO SIMPLICIO GUIMARÃES

Dissertação apresentada como parte dos
requisitos para obtenção do Grau de
Mestre em Ciências na Área de
Tecnologia Nuclear - Aplicações

Orientador:

Prof. Dr. Frederico Antônio Genezini

São Paulo

2020

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia Associada à Universidade de São Paulo

Desenvolvimento de um instrumento de movimentação e troca de amostras de radioisótopos para detectores HPGe - Germânio Hiper-Puro.

Versão Corrigida

Versão Original disponível no IPEN

ROBERTO SIMPLICIO GUIMARÃES

**Dissertação apresentada como parte
dos requisitos para obtenção do Grau
de
Mestre em Ciências na Área
de Tecnologia Nuclear - Aplicações**

**Orientador:
Prof. Dr. Frederico Antônio Genezini**

**São Paulo
2020**

Fonte de Financiamento: próprio

Autorizo a reprodução e divulgação total ou parcial deste trabalho,
para fins de estudo e pesquisa, desde que citada a fonte

Como citar:

GUIMARAES, R. S. ***Development of a radioisotope sample movement and exchange instrument for HPGe - Hyper-Pure Germanium detectors***. 2020. 104 p. Dissertação (Mestrado em Tecnologia Nuclear), Instituto de Pesquisas Energéticas e Nucleares, IPEN-CNEN/SP, São Paulo. Disponível em: (data de consulta no formato: dd/mm/aaaa)

Ficha catalográfica elaborada pelo Sistema de geração automática da Biblioteca IPEN/USP,
com os dados fornecidos pelo(a) autor(a)

GUIMARAES, ROBERTO SIMPLICIO

Development of a radioisotope sample movement and exchange
instrument for HPGe - Hyper-Pure Germanium detectors / ROBERTO
SIMPLICIO GUIMARAES; orientador Frederico Antonio Genezini.

-- São Paulo, 2020.

104 p.

Dissertação (Mestrado) - Programa de Pós-Graduação em
Tecnologia Nuclear (Aplicações) -- Instituto de Pesquisas
Energéticas e Nucleares, São Paulo, 2020.

1. mecanismo. 2. Arduino. 3. Detector HPGe. 4. precisão.
5. baixo custo. I. Genezini, Frederico Antonio , orient. II.
Título.

AGRADECIMENTOS

Ao instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), pela oportunidade de ofertar o grau de mestre os seus docentes e técnicos.

Aos meus queridos pais Gentil e Aster(in memória) pelos valores, princípios e ensinamentos que me norteiam na vida pessoal e profissional.

A minha amada esposa Edite pela sua infinita paciência e apoio incondicional.

Aos meus colegas de trabalho, Aremilson, Juliano Fischer, Gleyser, Clayton Ferraz e Ewerton pelo apoio, incentivo e ajuda ao longo de todo o percurso do mestrado.

Aos professores de informática do curso Tecnólogo em Análise e Desenvolvimento de Sistemas, campus Vilhena, pelo incentivo e colaboração.

Ao Dr. Frederico Antonio Genezini pela sua orientação e confiança ao longo do trabalho e de suas valiosas sugestões na execução das atividades, em especial nos laboratórios do IPEN.

Ao Dr. Guliherme Soares Zahn pela ajuda, orientações e sugestões.

Ao Dr. Edson Gonçalves Moraes pela disponibilidade do uso do laboratório de ativação neutrônica.

Ao Dr. Iberê pela ajuda e apoio em todos os momentos de dúvida.

Aos demais pesquisadores do CRPq/IPEN pela acolhida e companheirismo que usufrui durante os meses que estive em sua valiosa companhia.

RESUMO

GUIMARÃES, R.S.. **Development of a radioisotope sample movement and exchange instrument for HPGe - Hyper-Pure Germanium detectors**.2020.104 p. Dissertação (Mestrado em Tecnologia Nuclear), Instituto de Pesquisas Energéticas e Nucleares, IPEN-CNEN/SP, São Paulo.

Um conjunto mecânico comandado por um microcontrolador que movimenta e posiciona com precisão uma série de amostras irradiadas foi montado. Este conjunto é personalizado para um arranjo com um detector de HPGe e blindagem, e traz ganhos dentre os quais se destacam desobrigar a presença do operador durante a aquisição de dados, minimizando sua exposição às radiações oriundas da amostra e proporcionar medidas com repetibilidade. O arranjo onde o equipamento foi montado é localizado no laboratório de ativação neutrônica na CRPq no IPEN/CNEN. Esse mecanismo, controlado por uma placa denominada Arduino MEGA 2560, possui uma interface amigável desenvolvida em C#, interage com o sistema de aquisição denominado Genie2000 e armazena os dados da contagem em um banco de dados MySQL. Em todo o conjunto mecânico existem peças desenvolvidas e manufaturadas em impressora 3D. Nesse arranjo o conjunto das amostras fica posicionado na lateral externa da blindagem, sendo necessária blindagem adicional em torno do conjunto para evitar aumento da radiação de fundo no laboratório e que a radiação emitida por elas alcance o detector. A amostra aproxima-se do detector por uma escotilha superior que abre e fecha conforme a movimentação do braço mecânico que contem a amostra na sua extremidade. Sensores de fim de curso asseguram a calibração do sistema em todos os seus movimentos, sendo que a amostra pode ser posicionada em frações de mm em relação ao detector HPGe, conforme programado pelo usuário, utilizando todo o curso do interior da blindagem. O instrumento pode ser replicado e customizado para outros sistemas de detecção na espectrometria gama, mantendo a filosofia de utilizar matérias acessíveis e ter o custo menor que os sistemas comerciais. Testes de repetibilidade foram executados demonstrando assim a eficiência do sistema.

Palavras chave: mecanismo, Arduino, Detector HPGe, precisão, baixo custo

ABSTRACT

GUIMARÃES, R.S.. **Development of a radioisotope sample movement and exchange instrument for HPGe - Hyper-Pure Germanium detectors**. 2020. 104 p. Dissertação (Mestrado em Tecnologia Nuclear), Instituto de Pesquisas Energéticas e Nucleares, IPEN-CNEN/SP, São Paulo.

A micro controller driven mechanical assembly that accurately moves and positions a series of irradiated samples to HPGe detectors to minimize user exposure to gamma radiation from the sample and provide long-term repeatability measurements. The purpose of this work was to develop a custom mechanism for the arrangement called Canberra 6, located in the neutral activation laboratory at CRPq at IPEN / CNEN. This mechanism controlled by a board called Arduino MEGA 2560, has a user-friendly interface developed in C # that interacts with the acquisition system called Genie2000 and stores counting data in a MySQL database. Throughout the mechanical assembly there are unique parts developed. and manufactured in 3D printer. In this arrangement the sample set is positioned on the outside of the HPGe detector shield, and additional shielding around the assembly is required. The sample approaches the detector through an upper hatch that opens and closes as the mechanical arm that contains the sample at its end moves. Limit sensors ensure the system is calibrated in all its movements, and the sample can be positioned in fractions of mm to the HPGe detector as programmed by the user using the full stroke inside the shield. be replicated and customized for other detection systems in gamma spectrometry, maintaining its low cost and efficiency, including for other equipment installed in the referred IPEN laboratory.

Keywords: engine, Arduino, HPGe Detector, precision, low cost

ÍNDICE DE FIGURAS

Figura 1 - Gráfico de energia e número atômico das regiões predominantes dos três tipos de interação da radiação com a matéria.....	17
Figura 2 - Diagrama de ligação do DAS 1000.....	21
Figura 3 - Trocador de amostras.....	27
Figura 4 - Visão geral do trocador de amostras.....	28
Figura 5 - a) Sistema gaiola; b) mecanismo com seringa.....	29
Figura 6 - Movimentador de Amostra – Mara.....	30
Figura 7 - a) Suporte fixo, b) Prateleira de acrílico, c) Prateleira encaixada no suporte fixo.....	31
Figura 8 - Suporte acoplado no detector HPGe.....	31
Figura 9 - Detector HPGe dentro da blindagem de chumbo.....	32
Figura 10 - Modelo conceitual do IMAR sobre Canberra 6.....	33
Figura 11 - Modelo Conceitual do IMAR destacando abertura da blindagem no Canberra 6.....	33
Figura 12 - Distância das fontes do detector HPGe.....	34
Figura 13 - Suporte da amostra.....	35
Figura 14 - movimentador de amostras.....	36
Figura 15 - Atuador linear com motor DC.....	37
Figura 16 - Garra do suporte, em destaque o fim de curso.....	37
Figura 17 - Mecanismo movimentador do recipiente da amostra.....	38
Figura 18 - Painel de aço.....	39
Figura 19 - Sistema de abertura e fechamento da blindagem.....	40
Figura 20 - Tela principal de inserção de dados pelo usuário.....	41

Figura 21 - Mesa de coordenadas.....	42
Figura 22 - Gráfico atenuação x espessura Pb	44
Figura 23 - Espectro de medidas de BG e medida com fonte de Cs e Co utilizando 5 cm de blindagem de chumbo.	45
Figura 24 - Espectro de medidas de BG e medida com fonte de Cs e Co utilizando 15 cm de blindagem de chumbo.	46
Figura 25 - Espectro de medidas de BG e medida com fonte de Cs e Co utilizando 15 cm de blindagem de chumbo.	47
Figura 26 - a) Suporte da amostra; b) Diâmetro do suporte	51
Figura 27 - Motor de passo Kalatec	52
Figura 28 - Descrição técnica do motor de passo Kalatec	53
Figura 29 - Dimensões do suporte dos eixos	54
Figura 30 - a) Rolamento, b) Medidas rolamento e eixo liso, c) peça impressa e montada	55
Figura 31 - Suporte do motor de passo.....	55
Figura 32 - a) Eixo linear, b) Rolamento linear, c) Conjunto eixo e rolamento .	56
Figura 33 - Suporte do rack das amostras	57
Figura 34 - Conjunto rack das amostras	57
Figura 35 - Sensor foto interruptor	58
Figura 36 - Prototipagem do circuito do reset.....	59
Figura 37 - Esquemático do circuito do reset	59
Figura 38 - Placa PCB do reset.....	60
Figura 39 - Conjunto reset montado.....	60
Figura 40 - Conjunto reset montado.....	61

Figura 41 - a) projeto da placa, b) placa confeccionada, c) placa Arduino Mega 2560	62
Figura 42 - Diagrama das portas do arduino mega2560	63
Figura 43 - Diagrama de atividades do sistema	65

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVOS.....	12
2	FUNDAMENTOS TEÓRICOS	13
2.1	ANÁLISE POR ATIVAÇÃO NEUTRÔNICA	13
2.2	INTERAÇÃO PARTÍCULAS CARREGADAS COM A MATÉRIA.....	13
2.2.1	Partículas carregadas pesadas	14
2.2.2	Partículas carregadas leves.....	15
2.2.3	Interação da radiação eletromagnética com a matéria	16
2.2.4	Absorção fotoelétrica	16
2.2.5	Espalhamento Compton	17
2.2.6	Produção de Pares	17
2.3	ELETRÔNICA UTILIZADOS NA ESPECTROMETRIA GAMA	18
2.3.1	Detectores (HPGe)	18
2.3.2	Eficiência na detecção.....	19
2.3.3	Analizador multicanal DSA 1000 e Genie 2000	20
2.4	PROTEÇÃO RADIOLÓGICA.....	21
2.4.1	Blindagem de raios gama	22
2.4.2	Ângulo Sólido.....	23
2.5	MECANISMOS DE MOVIMENTAÇÃO DE AMOSTRAS	25
2.5.1	Posicionador de amostra de Austin	26
2.5.2	Trocador de amostras de Jonhnesburg	26
2.5.3	Sistema mecânico-pneumático de Dubna	27

2.5.4	Movimentador automático na Austrália.....	28
2.5.5	Laboratório de Ativação Neutrônica do IPEN	29
3	CONFEÇÃO DO TROCADOR DE AMOSTRAS	30
4	ANÁLISE E DISCUSSÃO DOS RESULTADOS.....	35
4.1	SUPORTE DA AMOSTRA.....	35
4.2	MOVIMENTADOR DE AMOSTRAS	35
4.3	ATUADOR LINEAR	36
4.4	GARRA DO SUPORTE DA AMOSTRA.....	37
4.5	MECANISMO MOVIMENTADOR DO SUPORTE DE AMOSTRA	38
4.6	PAINEL DE CONTROLE	38
4.7	SISTEMA DE ABERTURA DA BLINDAGEM.....	39
4.8	SOFTWARE DE INTERAÇÃO.....	40
4.9	BANCO DE DADOS	42
4.10	MECANISMO POSICIONADOR DO RECIPIENTE DA AMOSTRA	42
4.11	PAINEL DE CONTROLE	42
5	TESTES EFETUADOS E RESULTADOS OBTIDOS	43
5.1	MESA DE AÇO E TESTES DE BLINDAGEM.....	44
5.2	BANCO DE DADOS	47
5.3	CUSTO DO IMAR.....	48
6	CONCLUSÕES	49
7	APÊNDICE A - SUPORTE DA AMOSTRA.....	51
8	APÊNDICE B - MOVIMENTADOR DE AMOSTRAS	52
8.2	BASE DO SUPORTE.....	53

8.3	SUORTE DO MOTOR DE PASSO.....	55
8.4	EIXO LISO LINEAR	56
8.5	SUORTE PARA O <i>RACK</i> DAS AMOSTRAS.....	56
8.6	CONJUNTO RACK DAS AMOSTRAS.....	57
8.7	SENSOR DE FIM DE CURSO E RESET	58
8.8	SENSOR DE PRESENÇA DO SUORTE	61
9	APÊNDICE C - PAINEL DE CONTROLE	61
10	REFERÊNCIAS.....	66
11	ANEXOS	68

1 INTRODUÇÃO

A Análise por Ativação Neutrônica (NAA, da sigla em inglês) é uma técnica analítica amplamente utilizada na determinação de concentração de elementos, e tornou-se um método prático de análise com o desenvolvimento de reatores nucleares como uma fonte de nêutrons de alta intensidade. A introdução dos detectores de raios gama por cintilação Na(Tl) e mais tarde os de alta resolução como o Ge(Li) e o HPGe estabeleceram a NAA como o procedimento analítico sensível e de boa precisão e exatidão, útil para análises multielementares quali e quantitativas de elementos de baixa concentração. (ALFASSI, Z., 1990).

Hoje, a NAA constitui-se em uma das aplicações mais importantes dos reatores nucleares de pesquisa, e esta técnica tem sido amplamente utilizada no Laboratório de Ativação Neutrônica (LAN), do IPEN, para a determinação das concentrações de elementos em diversos tipos de matrizes, sejam elas de origem geológicas, biológicas ou ambientais. Esta técnica foi impulsionada pelos avanços tecnológicos, dentre outros, podemos destacar a confecção de detectores de germânio hiperpuro (HPGe) na década de 1980 e o avanço da eletrônica digital, nos anos 2000.

A grande maioria das análises de NAA do LAN usa uma metodologia chamada de comparativa, onde a amostra e um padrão de um elemento químico são irradiados em conjunto nas mesmas condições de tempo e fluxo de nêutrons. A concentração de um elemento de interesse particular é obtida comparando-se as taxas de contagens dos espectros de amostra e padrão, adquiridas a partir das mesmas condições de geometria de contagem, feita a correção devido ao decaimento.

Atualmente no LAN do IPEN, o posicionamento das amostras e do padrão é realizado manualmente, e o tempo de contagem utilizado na aquisição de dados pode variar de uma a três horas em média, demandando muito tempo de dedicação do usuário, que deve acompanhar as medidas devido à necessidade de trocar manualmente as amostras,.

Para equacionar a questão da exigência do usuário acompanhar todo o processo de aquisição da série de amostras, uma solução é a automatização

da troca e movimentação das amostras irradiadas. Algumas soluções comerciais estão disponíveis no mercado, no entanto a maior parte é feita para blindagens padrão ou fornecem o próprio arranjo blindagem + movimentador de amostras. Além disso, os equipamentos que executam essa tarefa fornecidos pelas empresas comerciais são muito caros, da ordem de dezenas de milhares de dólares, tornando sua aquisição quase proibitiva para laboratórios com poucos recursos. Assim, ao longo das últimas décadas, laboratórios de NAA em diversas partes do mundo, buscam mitigar essa questão com soluções caseiras. Alguns desses casos são:

Em 1974, na universidade do Texas, EUA, um sistema automático foi desenvolvido para movimentar múltiplas amostras de dois diferentes tamanhos, que eram acondicionadas uma plataforma de 18 polegadas.

Em 2001, outro dispositivo pneumático para 200 amostras foi desenvolvido em Johannesburg, África do Sul, o qual posiciona a amostra selecionada e após aquisição faz o seu descarte em local devidamente blindado.

Em 2014, o dispositivo desenvolvido em Dubna, foi um aparato que demandava equipamentos e recursos elevados por se tratar de um sistema mecânico-pneumático. (PAVLOV, S. S.; DMITRIEV A.Y.; FRONTASYEVA M.V., 2006)

Em 2018, foi desenvolvido um mecanismo simples de baixo custo composto por um trocador de amostras que utiliza uma seringa acoplada em ventosas para a movimentação e o posterior posicionamento das amostras. (CARVALHO, M.C.; SANDERS, C.; HOLLOWAY, C., 2018)

.A proposta desse trabalho foi projetar e construir um sistema automático robusto, preciso e de valor de manufatura inferior ao obtido no mercado, que posicione e troque a fonte radioativa em posições programadas. Sua instalação foi feita no LAN do IPEN-CNEN/SP, no equipamento denominado Canberra 6.

1.1 OBJETIVOS

O instrumento de troca e posicionamento de amostras, objeto desse trabalho denomina-se IMAR (Instrumento de Movimentação e Troca de Amostras Radioativas), e foi instalado no LAN (Canberra 6).

O IMAR foi concebido de modo a realizar as seguintes atividades:

- a) Automatizar a movimentação e as trocas de radioisótopos nas medições de espectroscopia gama no detector HPGe, instalado no LAN do Centro do Reator de Pesquisa (CRPq) do Instituto de Pesquisas Energéticas e Nucleares (IPEN- CNEN/SP).
- b) Diminuir as possíveis exposições humanas às fontes de radiação manipuladas no referido laboratório;
- c) Auxiliar nas rotinas de reprodutibilidade - por fazer a troca e movimentação dos radioisótopos de forma automatizada - minimizando o tempo dedicado.
- d) Facilidade de acesso nos dados coletados – os dados coletados serão enviados e armazenados em um banco de dados, possibilitando que eles possam ser comparados posteriormente com os dados de outros experimentos.
- e) Integração com o programa de aquisição de dados (Genie 2000).

2 FUNDAMENTOS TEÓRICOS

2.1 ANÁLISE POR ATIVAÇÃO NEUTRÔNICA

O método de Análise por Ativação Neutrônica (NAA) baseia-se na formação de isótopos radiativos, por meio de reações nucleares e na subsequente medida de radiação emitida por sua desexcitação. Este método se diferencia da maior parte dos métodos analíticos usuais, visto que ele se baseia em reações que ocorrem nos núcleos atômicos e não com os elétrons orbitais é considerado um método de referência para análise de elementos em diferentes matrizes e é multielementar, ou seja, possibilita a determinação simultânea das concentrações de vários elementos.

O método de NAA utiliza um bombardeio com nêutrons que ao interagirem com núcleos alvo podem produzir novos núcleos compostos em estado altamente excitado e que perdem sua energia por meio da emissão de um fóton chamado de gama pronto ou pela emissão de partículas, com meia vida de 10^{-16} a 10^{-18} s. No entanto, esse novo núcleo pode ainda estar fora da linha de estabilidade e, por vários processos de transmutação (emissões de partículas), voltará à linha de estabilidade. O tipo de partícula emitida depende da reação nuclear ocorrida, ou seja, é dependente da energia dos nêutrons e das características do núcleo alvo.

2.2 INTERAÇÃO PARTÍCULAS CARREGADAS COM A MATÉRIA

A radiação ionizante é caracterizada como uma partícula que ao atravessar um meio material possui energia suficiente para ionizar o meio com o qual interage. As ionizações ocorrem pela transferência de energia dessas partículas para elétrons dos átomos com o qual a radiação interage, que por sua vez podem ser ejetados, no entanto, a partícula que atravessa o meio também pode depositar parte de sua energia por meio de excitações dos átomos. Portanto as partículas podem perder energia de duas formas ionização e excitação.

A maneira como a radiação perde energia depende de fatores como:

- 1) Natureza do meio com o qual a radiação interage;
- 2) Energia da partícula incidente;
- 3) Tipo da radiação ionizante incidente.

Segundo Okuno (OKUNO, 2010), as radiações ionizantes se dividem em duas categorias, sendo:

- *radiações diretamente ionizantes*: que compreendem as radiações com carga elétrica, dentre elas, elétrons (que são partículas leves); prótons e alfa (partículas pesadas);
- *radiações indiretamente ionizantes*: que compreendem o conjunto de radiação sem carga elétrica (fótons e nêutrons).

O objetivo final desse trabalho é o desenvolvimento de um trocador de amostras automatizado para ser utilizados em medidas de NAA, portanto, é fundamental considerar os mecanismos de interação da radiação com a matéria a fim de se minimizar interferências oriundas dos elementos que são utilizados na NAA. Os radioisótopos utilizados na NAA normalmente decaem emitindo radiação gama (indiretamente ionizante) e partículas beta menos (que são partículas leves carregadas).

Nas próximas subseções serão descritas as principais formas de interação das partículas leves carregadas (beta menos), partículas carregadas pesadas e fótons com a matéria, que devem ser blindadas para não interferirem nas medidas em NAA.

2.2.1 Partículas carregadas pesadas

A maneira de detectar partículas carregadas (prótons, alfas e íons pesados) se dá pela sua interação com o material detector. As colisões íon-átomo é o processo dominante, com perda de energia do íon em processos de ionização e excitação dos elétrons atômicos. A transferência de energia do íon para os elétrons ocorre em pequenas quantidades em cada colisão, devido à grande diferença de massas, porém quanto maior for o número de colisões, maior será o freamento, até que o íon seja totalmente freado. A medida que um íon altamente energético penetra num meio material adquire um alto estado de ionização independente de seu estado anterior à incidência sobre meio. À

medida que o íon perde velocidade, a probabilidade de captura de elétrons orbitais aumenta e o íon é gradualmente neutralizado. Sua trajetória até este momento é aproximadamente reta. Quando a velocidade do íon é suficientemente baixa, os processos de ionização e excitação de átomos do meio já não são mais importantes e as perdas de energia por colisões elásticas com os átomos do meio começam a ser dominantes, dando origem a espalhamentos de ângulos grandes.

Os processos de perda de energia por colisões inelásticas com elétrons (freamento eletrônico, S_e) e por colisões átomo-átomo (freamento nuclear, S_n) são praticamente independentes, de tal maneira que a taxa de perda de energia por unidade de comprimento pode ser expressa como a soma das duas contribuições, conforme equação:

$$-\frac{dE}{dx} = S_e(v) + S_n(v) \quad (1)$$

2.2.2 Partículas carregadas leves

Para que um elétron (ou beta menos, no caso do decaimento nuclear) possa ser considerado uma radiação ionizante ele deve possuir uma energia cinética maior que a energia de ligação dos átomos do meio com o qual ele interage (CARVALHO, M.C.; SANDERS, C.; HOLLOWAY, C., 2018), (PIZZINI, S, 2017)

Os mecanismos de interação das partículas leve carregadas são por meio de interação *coloumbiana* com os elétrons do meio com o qual ele interage, no entanto, devido a sua baixa massa de repouso, essas partículas podem sofrer mudanças bruscas de direção durante as interações, fato esse que resulta em trajetórias tortuosas, em consequência desse fato a penetração de elétrons de mesma energia pode variar bastante em um mesmo meio de interação. Essas partículas possuem alcance maiores na matéria quando comparado com as partículas pesadas em virtude de provocarem menos ionizações durante sua trajetória.

As partículas leves carregadas também podem interagir com os núcleos atômicos, originando radiação de freimento, ou *bremssstrahlung*. Em projetos

de blindagem esse tipo de radiação também deve ser considerada, uma vez que podem interferir na espectroscopia da amostra analisada.

2.2.3 Interação da radiação eletromagnética com a matéria

Os mecanismos principais pelos quais os raios gama interagem com a matéria são: a) absorção fotoelétrica, b) espalhamento Compton e c) produção de pares (KNOLL, G.F., 2000). Nesses processos, há a transferência, total ou parcial, de energia do fóton gama para a elétrons. As maneiras como a radiação eletromagnética perde energia serão descritas nas próximas subseções

2.2.4 Absorção fotoelétrica

Nesse processo a energia do fóton gama é completamente absorvida por um elétron orbital que é ejetado, que por sua vez, é desligando do átomo tornando-se um elétron livre, o fotoelétron. Sua energia ($E_{cinética}$) corresponde à energia do fóton incidente ($E_{fóton}$) subtraída pela energia de ligação do elétron no seu nível de energia original (E_{lig}), conforme equação:

$$E_{cinética} = E_{fóton} - E_{lig} \quad (2)$$

O efeito fotoelétrico (Figura 1), ocorre de forma predominante para fótons que carregam baixo valor de energia e que interajam com materiais de alto número atômico devido à absorção fotoelétrica ($\sigma_{fotoelétrica}$) ser proporcional ao número atômico (Z) e inversamente proporcional ao valor de energia do fóton (E), conforme equação abaixo (KNOLL, G.F., 2000):

$$\sigma_{fotoelétrica} \propto \frac{Z^5}{E^{3,5}} \quad (3)$$

Figura 1 - Gráfico de energia e número atômico das regiões predominantes dos três tipos de interação da radiação com a matéria.

FONTE:(Knoll, 2000)

2.2.5 Espalhamento Compton

No processo do espalhamento Compton a energia do fóton gama é parcialmente absorvida por um elétron orbital, que é ejetado de sua orbita atômica tornando-se um elétron de recuo. O fóton, agora com menor energia que o original, é espalhado em um ângulo θ , sendo possível calcular a energia do fóton espalhado por meio da equação 4 (KNOLL, G.F., 2000).

$$hv' = \frac{hv}{1 + \frac{hv}{m_0c^2}(1 - \cos \theta)} \quad (4)$$

Como ilustrado na figura 1, o espalhamento Compton é predominante para fótons de energia intermediária, que interajam em materiais com baixo valor de número atômico.

2.2.6 Produção de Pares

No processo de produção de pares o fóton incidente interage com o campo elétrico do núcleo e sua energia é convertida em massa, produzindo o par: elétron –pósitron. Para o surgimento desse fenômeno, é necessário que a

energia do fóton incidente seja superior a 1,022 MeV, que corresponde a soma das massas de repouso do elétron (511 KeV) com o do pósitron (511 Kev) (KNOLL, G.F., 2000). Quando a energia do fóton for excedente a 1,022 MeV, tal excesso é repartido em energia cinética entre as partículas criadas na forma de energia cinética, como mostra a equação 5:

$$E_y = K_{\beta^-} + K_{\beta^+} + 2m_0c^2 \quad (5)$$

Sendo:

$E_y \rightarrow$ Energia do fóton incidente;

$K_{\beta^-} \rightarrow$ Energia cinética do elétron;

$K_{\beta^+} \rightarrow$ Energia cinética do pósitron;

$m_0c^2 \rightarrow$ Energia de repouso, responsável pela criação da massa de repouso das partículas;

2.3 ELETRÔNICA UTILIZADOS NA ESPECTROMETRIA GAMA

Tipicamente o sistema de detecção de raios gama é constituído por um detector semicondutor de germânio hiperpuro (HPGe) associado ao analisador de multicanal (MCA) e a um computador onde o software encarregado de controlar a aquisição de dados, exibição do espectro com os resultados da medição dos raios gama e frequentemente analisar os dados coletados durante o processo de medição é executado.

2.3.1 Detectores (HPGe)

Os detectores de HPGe são atualmente os mais utilizados em pesquisas com raios gama de alta resolução e uma eficiência relativa com valores em torno de 20% a 60% (KNOLL, G.F., 2000). Sua eficiência está relacionada ao seu tamanho (eficiência geométrica) e à capacidade de absorver os raios gama que o atingem (eficiência intrínseca). Outro fator que influencia a aquisição de

dados é o “tempo morto”, e refere-se ao tempo finito requerido para o espectrômetro processar fótons individuais.

A metodologia desse sistema consiste na coleta e medida de cargas elétricas produzidas pela interação da radiação gama incidente no cristal de germânio hiperpuro, conhecido como HPGe (“High Purity Germanium”). Nos modernos detectores de germânio hiperpuro o grau de concentração de impurezas pode atingir valores tão baixos como de 1000 átomos/cm³ (GERMANO, R.V., 2012.). Cristais grandes, de até 8 cm de diâmetro por 12 cm de comprimento útil, podem ser obtidos por processo de crescimento e posteriormente usinados para finalizar sua confecção. Para diminuir a corrente de fuga, causada pelo elevado número de excitações térmicas, os detectores HPGe operam a baixas temperaturas, em geral à temperatura de ebulição do nitrogênio líquido, mas podem ser estocados à temperatura ambiente desde que sem polarização.

Os fótons emitidos pela fonte radioativa interagem com o cristal de germânio hiperpuro produzindo pares de elétron-lacuna na região de depleção da junção pn do mesmo. Os elétrons e as lacunas são coletados em terminais opostos devido à diferença de potencial nesses terminais. Essa coleta gera pulso cuja amplitude é proporcional à energia liberada na interação que é posteriormente pré amplificado em um pré-amplificador sensível à carga.

2.3.2 Eficiência na detecção

Geralmente a sensibilidade está diretamente e proporcionalmente associada à eficiência do detector, cuja eficiência é habilidade de medir pulsos que ocorrem devido a incidência de fótons, ou seja, a fração de todos os fótons que são emitidos para fonte ou amostra(A), que causa um evento no detector(N). Esta é a eficiência ε do detector, conforme equação:

$$\varepsilon = \frac{A \cdot T}{N} \quad (6)$$

Onde T é o tempo em que o detector registrou os pulsos.

Conforme dito, existe a componente da eficiência intrínseca e geométrica, de modo que a eficiência é obtida por:

$$\varepsilon_T = \varepsilon_{int} \cdot \varepsilon_{geo}$$

A medida de eficiência pode ser referente à eficiência total, ou seja, o número total de fótons registrado dividido pelo número total de fótons adquiridos. No entanto para muitas aplicações, usa-se a eficiência no fotopico, que diz respeito à capacidade de absorção total dos fótons. Como a absorção total é fortemente dependente da energia, essa medida de eficiência também é. Ela é obtida pela divisão do número de fótons, com determinada energia, pelo número de fótons emitidos pela fonte.

2.3.3 Analisador multicanal DSA 1000 e Genie 2000

O analisador multicanal utilizado no LAN do IPEN/CNEN é o modelo DSA1000 fabricado pela Canberra e trata-se de um analisador digital de pulso multicanal (MCA) integrado de 16K canais e sua função é digitalizar os pulsos analógicos coletados pelo detector HPGe, integrando a carga de cada um deles, e distribuir estes pulsos em função de sua amplitude nos canais do MCA formando um histograma que representa o número de fótons absorvidos em função de suas energias. O software que controla o sistema de aquisição utilizado é o Genie 2000, também fabricado pela Canberra. O resultado final mostrará o espectro em energia dos fótons emitidos pelo radionuclídeo em estudo, com um ou mais picos (cada um com uma distribuição aproximadamente gaussiana), correspondentes aos fótons que depositaram toda a energia no interior do cristal (picos de absorção total), além de uma estrutura proveniente das interações com perda parcial de energia (figura 2).

A identificação de um radionuclídeo é feita a partir da calibração do espectro em energia localizando os picos de absorção total.

Figura 2 - Diagrama de ligação do DAS 1000

FONTE: (Manual DAS 1000)

2.4 PROTEÇÃO RADIOLÓGICA

Proteção radiológica é a ciência que tem por objetivo oferecer proteção aos indivíduos e ao meio ambiente contra os efeitos nocivos à saúde oriunda da radiação ionizante. A proteção radiológica deve atender, entre outros, ao princípio da otimização, também conhecido como princípio ALARA (As Low As Reasonably Achievable) que preconiza: as doses devem ser mantidas tão pequenas quanto é racionalmente exequível, considerando os aspectos econômicos e sociais.

A proteção radiológica ocupacional tem o objetivo de proteger o trabalhador cuja atuação profissional esteja ligada à exposição à radiação ionizante e contato com materiais radiativos, e visa manter condições de trabalho seguras e satisfatórias combinando fatores de proteção a saber: a) tempo, b) distância e c) blindagem.

Para se minimizar a exposição às radiações ionizantes é necessário minimizar o tempo de exposição e maximizar a distância. A dose recebida é diretamente proporcional ao tempo de exposição e inversamente proporcional ao quadrado da distância entre o usuário e a fonte. Outra medida bastante

relevante é a blindagem, ou seja, a colocação de matérias entre a fonte de radiação e o usuário para diminuir a dose neste ultimo.

2.4.1 Blindagem de raios gama

A blindagem tem por objetivo atenuar a radiação por meios de anteparos de materiais com seu número atômico(Z) elevado, como concreto, chumbo, aço, entre outros materiais. Sua justificativa de uso se torna necessária porque uma das principais características da radiação eletromagnética é sua grande capacidade de penetração, podendo percorrer milhares de metros no ar e ser detida somente por uma parede espessa de concreto ou metal [FRIEDLANDER *et al*, 1981]. Isso faz da radiação gama, dentre os tipos de radiação conhecidos, uma das que mais oferece riscos de efeitos biológicos prejudiciais no caso de exposições externas. Não é por acaso que uma das maiores preocupações dos órgãos de controle de radiação ionizante – Agência Internacional de Energia Atômica (IAEA- *international Atomic Energy Agency*) e Comissão Nacional de Energia Nuclear (CNEN) - tem sido com exposição externa devido à emissão gama [JÚNIOR *et al*, 2009].

A atenuação da radiação eletromagnética é determinada por:

$$I = I_0 e^{-\mu x} \quad (7)$$

Onde I é a intensidade final do feixe após interagir com um material de espessura x ; I_0 é a intensidade inicial do feixe e μ é o coeficiente de atenuação linear do material (que é dependente da energia da radiação gama e do número atômico do material).

Mecanismos de movimentação de amostras fornecem uma resposta apropriada para que o usuário no laboratório do IPEN-CNEN/SP não manipule manualmente amostras irradiadas, limitando sua exposição à radiação ionizante.

2.4.2 Ângulo Sólido

Para que seja definido o conceito de ângulo sólido, pode-se partir de uma esfera de raio R , conforme a indicada na figura 3. Um ponto na superfície desta esfera pode ser descrito usando as coordenadas esféricas R , θ e ϕ .

Figura 3 - Ângulo Sólido

FONTE: (Knoll, 2000)

Caso se deseje calcular a área compreendida pela calota superior, pode-se imaginar uma superfície infinitesimal de área da superfície desta esfera, dA , tendo sua projeção no ponto central da esfera, conforme é ilustrado na figura 4.

Figura 4 - Área calota superior

FONTE: (Knoll, 2000)

Pode-se, em primeira aproximação, considerar o elemento dA como um quadrilátero, cuja área é definida por

$$dA = R \cdot \text{sen}\phi \cdot d\theta \cdot R \cdot d\phi \quad (8)$$

Integrando-se esse elemento de área, pode-se obter a área em uma região desejada.

$$A = \iint dA = \int_{\phi_i}^{\phi_f} \int_{\theta_i}^{\theta_f} R^2 \cdot \text{sen}\phi \cdot d\theta \cdot d\phi \quad (9)$$

No caso da calota, pode-se integrar o ângulo θ da figura X ao longo de todo o círculo, ou seja, com os limites de integração entre 0 e 2π .

$$A = R^2 \int_{\phi_i}^{\phi_f} \theta \Big|_0^{2\pi} \text{sen}\phi \cdot d\phi = 2\pi R^2 \int_{\phi_i}^{\phi_f} \text{sen}\phi \cdot d\phi \quad (10)$$

Para que não haja dependência do raio, considera-se uma esfera unitária. Portanto, a definição de ângulo sólido, representado pela letra Ω é a área de uma circunferência unitária.

$$A = \Omega_{calota} = 2\pi \int_{\theta_i}^{\theta_f} \sin\theta \cdot d\theta = 2\pi(1 - \cos\theta) \quad (11)$$

No caso de uma esfera, o ângulo sólido é $\Omega=4\pi$.

Como R é bastante grande com relação às dimensões do cristal, pode-se assumir que o cristal do detector é uma esfera. Assim, tem-se:

$$\Omega_{calota} = 2\pi \left(1 - \frac{R}{\sqrt{R^2 + (a/2)^2}} \right) \quad (12)$$

Sendo R a distância da fonte ao detector e R seu raio. Note-se que $R \gg a/2$.

A fração de fótons que alcançam a calota é dada por:

$$\frac{\Omega_{calota}}{\Omega_{esfera}} = \frac{2\pi \left(1 - \frac{R}{\sqrt{R^2 + (a/2)^2}} \right)}{4\pi} = \frac{1}{2} \left(1 - \frac{R}{\sqrt{R^2 + (a/2)^2}} \right) \quad (13)$$

2.5 MECANISMOS DE MOVIMENTAÇÃO DE AMOSTRAS

Por décadas, os Laboratórios de Ativação Neutrônica em todo o mundo buscam automatizar o processo de medição na NAA. Esse trabalho selecionou

alguns desses mecanismos automatizados, já apresentados em ordem cronológica na introdução.

2.5.1 Posicionador de amostra de Austin

Pesquisadores da Universidade de Austin, Texas(EUA), desenvolveram em 1974 um sistema automatizado de troca de amostras. O trocador consistia num disco de alumínio de 18 polegadas de diâmetro que girava para o posicionamento da amostra pretendida. Após a aquisição a amostra era descartada por meio de uma calha e deslizava para um recipiente no chão. (MAcADAMS, 1974). Com as limitações da eletrônica na época, o software possuía poucas funções de comando e seu armazenamento de dados era bastante reduzido.

2.5.2 Trocador de amostras de Jonhnnesburg

Um trocador automático de amostras para espectrometria de raios gama Ge(Li) foi desenvolvido na África do Sul com capacidade de 200 amostras. Era um conjunto de duas partes distintas, o armazenamento de amostras e a estação de contagem. O transporte entre essas partes era efetuado através de tubos pneumáticos. O trocador de amostras consistia em um cinto de alumínio de fluxo contínuo (Figura 5). O conjunto exigia um espaço físico considerável, sendo a distância das partes era de 180 metros. A utilização de sistema pneumático eleva consideravelmente os custos do equipamento.

Figura 5 - Trocador de amostras

FONTE: (Andewerg,1980)

2.5.3 Sistema mecânico-pneumático de Dubna

O LAN da cidade de Dubna, Rússia opera uma grande quantidade de amostras ambientais e tecnológicas em numerosos projetos nacionais e internacionais. Por isso o seu sistema de automação consiste em três detectores HPGe com eletrônica espectrométrica, três trocadores de amostras e o software de controle. Cada trocador possui 45 *slots* para recipientes com amostras. O controle da posição do recipiente durante a medição do espectro é realizado por *encoders* incrementais (Figura 6).

Uma limitação deste trocador é ele que ele permite que a amostra seja medida apenas em 3 posições fixas acima do detector, ou seja, o usuário não pode programar uma medição em uma altura customizada diferente das 3 pré definidas.

Figura 6 - Visão geral do trocador de amostras

FONTE: (Pavlov, 2014)

2.5.4 Movimentador automático na Austrália

Trata-se de uma estrutura metálica tipo gaiola de alumínio estrutural que circula o detector HPGe sobre o qual um sistema móvel que movimenta a amostra é colocado (Figura 7a). O sistema móvel é um mecanismo conectada a uma seringa de 20 ml com uma ventosa de 5 cm de diâmetro (Figura 7b). Esse sistema de ventosa pode ser um problema para amostras mais pesadas ou para recipientes muito menores. Outro aspecto é que o movimentador automático apenas deposita a amostra dentro do detector não sendo possível sua movimentação durante a aquisição (CARVALHO, M.C.; SANDERS, C.; HOLLOWAY, C., 2018)

Figura 7 - a) Sistema gaiola; b) mecanismo com seringa

FONTE: (Carvalho, 2018)

2.5.5 Laboratório de Ativação Neutrônica do IPEN

No LAN, do IPEN, nas NAA utiliza-se o método comparativo que consiste em irradiar simultaneamente, dentro de um mesmo dispositivo, uma amostra que se deseja analisar e um padrão com massa conhecida do elemento que se deseja determinar a concentração. Por esse método, a aquisição de dados da amostra e do padrão é realizada pelo mesmo equipamento, mantendo a mesma geometria de modo que a eficiência de detecção seja a mesma para a medição das atividades da amostra e do padrão.

Todo o processo de troca de amostras e de seu posicionamento em relação ao detector HPGe é realizado manualmente pelo usuário e no passado idealizou-se uma automação somente para o posicionamento da amostra.

Em 2015 foi desenvolvido um sistema de posicionamento automatizado visando aumentar a precisão das medidas, nominado de MARA (Movimentador de Amostras Radioativas Automatizado) - Figura 8. Tal mecanismo movimenta a amostra em apenas 4 posições fixas, sendo a sua principal utilização em medidas de radioisótopos de meias-vida curtas. Após realizar o teste de repetibilidade foram comparados os resultados obtidos dos espectros de energia emitidos pela fonte de ^{60}Co quando se utilizou o suporte fixo com os

resultados obtidos ao se utilizar o suporte automatizado MARA. Foi observado que as CPS (Contagem Por Segundo) obtidas como o suporte fixo para cada uma das 4 prateleiras estiveram na mesma ordem de grandeza das CPS obtidas com o suporte automatizado (SECCO, 2016).

Figura 8 - Movimentador de Amostra – Mara.

FONTE: (Secco, 2016)

Um modo de resolver a necessidade do operador estar presente durante a aquisição e portanto minimizar sua exposição à radiação é automatizar um sistema de posicionamento e troca de amostras, o que minimizaria a manipulação do usuário, que tem que levar a amostra ao detector e após a contagem levá-la à blindagem para armazená-la. Esse tipo de sistema automatizado também tem a vantagem de possibilitar que o usuário posicione a amostra na posição que ele deseje garantindo a reprodutibilidade na medição das amostras.

3 CONFEÇÃO DO TROCADOR DE AMOSTRAS

O sistema atualmente empregado nas medidas de análise por ativação no LAN (particularmente no Canberra 6) é apresentado na figura 9. A figura 9a mostra o suporte que se encaixa ao detector com prateleiras em 4 posições. A Figura 9b apresenta uma estrutura em forma de disco, com 0,58 cm de altura e diâmetro de 8,46 cm, confeccionada em acrílico, para o posicionamento da

amostra radioativa dentro de um rebaixo na peça mostrada na figura 5a (prateleiras), com diâmetro de 3,23 cm. Na Figura 9c é ilustrado o disco de acrílico encaixado em uma das posições do suporte fixo. O dispositivo apresenta comprimento total de 20,31 cm, sendo de 8,00 cm para fazer o acoplamento com o corpo do detector e 12,31 cm disponíveis para o posicionamento da amostra, sendo as distâncias fonte-detector de quatro posições fixas (0 cm, 3,5 cm, 7,0 cm e 10,5 cm), além de apresentar na parte inferior o diâmetro externo de 7,94 cm e diâmetro interno de 7,68 cm. O suporte fixo acoplado no detector como se vê na Figura 10.

Figura 9 - a) Suporte fixo, b) Prateleira de acrílico, c) Prateleira encaixada no suporte fixo

FONTE: (SECCO 2016)

Figura 10 - Suporte acoplado no detector HPGe.

FONTE: (IBERÊ, 2014)

No arranjo do CANBERRA 06, o detector de HPGe está posicionado dentro de uma blindagem de chumbo, conforme ilustrado na Figura 11. Nesse arranjo a distância entre o topo do detector até o teto da cabine de chumbo é de aproximadamente 235 mm.

Figura 11 - Detector HPGe dentro da blindagem de chumbo.

FONTE: (Secco, 2016)

Dependendo da atividade da fonte radioativa se faz necessário escolher a distância mais adequada para posicioná-la em relação ao detector para se realizar a aquisição de dados, pois se a fonte estiver emitindo muita radiação é necessário deixá-la mais afastada do detector para evitar a perda de informações com os efeitos de tempo morto e empilhamento. Porém, se ela apresentar uma quantidade muito baixa na emissão de radiação, ela deve ser posicionada próxima ao detector para obter mais eventos registrados no detector, resultando numa melhora estatística dos dados obtidos.

O movimentador de amostra proposto neste trabalho, ora nomeado IMAR (Instrumento de Movimentação de Amostras Radioativas) foi instalado sobre a blindagem de chumbo do detector de HPGe do sistema CANBERRA 06, localizado no LAN-CRPq, do IPEN.

Com o objetivo de proporcionar uma visão detalhada do IMAR e sua posição de instalação em relação ao Canberra 6, um modelo conceitual foi projetado no sketchup, programa de modelagem 3D, conforme ilustrado nas

Figuras 12,13 e 14. As imagens ilustram que no conceito proposto, o trocador de amostras tem duas partes principais: a) um mecanismo movimentador das amostras, posicionado sobre a blindagem; b) um suporte com as amostras, posicionado ao lado do arranjo.

Figura 12 - Modelo conceitual do IMAR sobre Canberra 6

FONTE: (Autor da dissertação, 2019)

Figura 13 - Modelo Conceitual do IMAR destacando abertura da blindagem no Canberra 6

FONTE: (Autor da dissertação, 2019)

Figura 14 - Distância das fontes do detector HPGe

FONTE: (Autor da dissertação, 2019)

O IMAR aproxima a amostra do detector no interior da câmara de blindagem, por uma janela na blindagem superior da câmara. Essa janela é fechada após a entrada da amostra. Como a distância máxima entre o detector e o teto da câmara é de 235 mm e o IMAR possui ao longo do seu curso a precisão milimétrica, o pesquisador terá a sua disposição 230 posições em vez das 4 disponíveis no sistema antigo.

O IMAR é composto de um trocador de amostra, um atuador linear, uma garra, um mecanismo movimentador do recipiente da amostra, sistema de abertura da blindagem, painel de controle e software de interação com o usuário, além de uma blindagem adicional de chumbo.

A elaboração e desenvolvimento de todos os componentes do IMAR possuem aspectos de originalidade, pois não são vendidos comercialmente, e a descrição técnica desses componentes será abordada em detalhes a seguir.

Em todo o entorno do IMAR será montada uma adequada blindagem de chumbo e a atual estrutura metálica da blindagem do detector será reforçada.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

4.1 SUPORTE DA AMOSTRA

Para que o sistema possa transportar as amostras, foi desenvolvido um suporte plástico de amostras projetado no software gratuito 3DBUILDER. As amostras, nas medições executadas no LAN, são normalmente fixadas em um recipiente de inox conhecido como “panelinha”. O suporte desenvolvido permite que tais “panelinhas” sejam encaixadas no seu interior e possui uma estrutura na parte superior que permite que uma garra seja encaixada possibilitando seu transporte seguro, como pode-se ver na Figura 15. Os detalhes do suporte podem ser vistos do APÊNDICE A.

Figura 15 - Suporte da amostra

FONTE: (Autor da dissertação, 2019)

4.2 MOVIMENTADOR DE AMOSTRAS

Trata-se de um módulo automatizado que posiciona o porta amostras com a amostra selecionada para que possa ser acoplada ao movimentador por meio da garra . Ele é composto de um *rack* para 6 suportes de porta amostras, um conjunto mecânico, sensores foto interruptores para atuarem como fim de

curso do *rack* e um circuito eletrônico para o usuário reiniciar o sistema, conforme figura 16. Os detalhes dos seus componentes estão descritos no APÊNDICE B.

Figura 16 - movimentador de amostras

FONTE: (Autor da dissertação, 2019)

4.3 ATUADOR LINEAR

O atuador linear da marca kalatec, permite a movimentação vertical da garra, possibilitando que o suporte de amostras seja retirado ou colocado no rack e posicionas obre o detector. Ele foi adaptado, substituindo o motor DC (como pode ser encontrado comercialmente) por um motor de passo, modelo nema 23 de 200 passos por volta, proporcionando precisão na movimentação da garra, conforme figura 17. O curso total do atuador é de 30 cm, assim a amostra pode ser posicionada ao longo dos 23,5 cm que separam o detector de HPGe do teto da blindagem. Sua mecânica permite 9900 posições programáveis ao longo do curso do atuador, sendo possível programar 330 posições distintas para cada centímetro. Atualmente, como já mencionado, o usuário possui apenas 4 posições para a amostra.

Figura 17 - Atuador linear com motor DC

FONTE: (Kalatec, 2019)

4.4 GARRA DO SUPORTE DA AMOSTRA

A garra do suporte da amostra está fixada no extremo do embolo do atuador linear e é composto por um conjunto de duas garras acionadas por um servo motor modelo MG 995 que é o mais utilizado em projetos de robótica de precisão. Possui engrenagem de alumínio e é capaz de suportar até 11 Kg/cm e tem velocidade de 0,17 seg/60° com 6 V de alimentação. Suas dimensões se ajustaram bem ao projeto. O conjunto de garras e o seu suporte foram idealizados e impressos em PLA. Um fim de curso, em destaque na figura 18, posiciona o suporte da amostra na posição inicial da contagem, correspondendo a posição 0 no arranjo anterior de 4 posições fixas, quando fizer contato com um anel plástico, encaixado no detector.

Figura 18 - Garra do suporte, em destaque o fim de curso

FONTE:(Autor da dissertação)

4.5 MECANISMO MOVIMENTADOR DO SUPORTE DE AMOSTRA

O mecanismo consiste de uma mesa de coordenadas fabricado pela empresa KALATEC S/A sob medida para este projeto, acionado por motor de passo, modelo NEMA 23, de 200 passos por volta, conforme figura 19. Uma mesa corre sobre o fuso com curso de movimentação de 70 cm. O atuador linear, descrito em 4.3, é fixado nessa mesa.

Figura 19 - Mecanismo movimentador do recipiente da amostra.

FONTE: (Kalatec, 2018)

4.6 PAINEL DE CONTROLE

O painel de controle é uma caixa de montagem metálica de 40 cm de altura, 40 cm de largura e 25 cm de profundidade, o qual comporta todos os componentes de acionamento, controle e alimentação do conjunto mecânico, conforme figura 20. Uma descrição mais detalhada dos componentes no painel está no APENDICE C.

Figura 20 - Painel de aço

FONTE: (Autor da dissertação, 2019)

4.7 SISTEMA DE ABERTURA DA BLINDAGEM

O atuador linear do trocador tem acesso ao detector pela parte superior da blindagem e para isso, uma abertura de 12 cm x 12 cm foi feita. No entanto, após a passagem da garra, a abertura é parcialmente fechada com um sistema de blindagem móvel, permanecendo ainda uma abertura de 4 cm x 6cm, necessária para o atuador linear. Com essa abertura menor, é possível movimentar a posição do suporte de amostras com relação à face do detector, no entanto para retirar a amostra da blindagem é necessária a abertura total da blindagem móvel. A abertura e fechamento da blindagem móvel ocorrem automaticamente pelo sistema, sincronizado com o movimento dos demais eixos. O tempo típico para a realização do ciclo da movimentação de cada amostra é de cerca de 2 minutos. Cada ciclo da movimentação compreende o período da coleta da amostra, sua movimentação ao longo de 70 cm de curso do movimentador, abertura e fechamento da blindagem e por fim de sua aproximação em relação ao detector HPGe. O sistema é composto por um mecanismo acionado por motores de passo modelo 23, de 200 passos por volta, que movimentará duas peças de chumbo, localizadas sobre a abertura superior da blindagem da câmara do detector HPGe, conforme Figura 21.

Figura 21 - Sistema de abertura e fechamento da blindagem

FONTE: (Autor da dissertação, 2019)

4.8 SOFTWARE DE INTERAÇÃO

Trata-se de uma *interface* amigável composta de uma série de telas interativas para inserção de parâmetros pelo usuário e que fornecerá posteriormente uma saída de dados por meio de relatórios que poderão ser impressos ou visualizados em PDF. Desenvolvido em C# no VisualStudio 2019. (VISUALSTUDIO®, 2019) o software envia e recebe dados do Arduino para poder interagir com o Genie 2000. Seu desenvolvimento foi por um programa na linguagem C# em seu apropriado ambiente de desenvolvimento, garantindo a compatibilidade com o ambiente Windows do computador que realiza as aquisições do Canberra 6.

O *software* de interação controla os motores do mecanismo, da blindagem móvel, a garra da amostra e a leitura nos diversos sensores do sistema.

A interação com o Genie 2000 ocorrerá através de uma codificação compatível com a sua linguagem, onde os dados armazenados nas variáveis de interesse do usuário serão utilizadas pelo IMAR. Essa interação está em fase de implantação, o que não compromete nenhuma das suas funcionalidades, basta o usuário abrir os dois programas na tela do computador (IMAR e Gennie2000). Após essa implantação o sistema IMAR irá fornecer os comandos para disparar a aquisição no software do Genie2000.

No programa do IMAR, atualmente, o usuário seleciona individualmente os parâmetros de cada amostra. Basta para isso nomear os dados da amostra(a) e inserir seus parâmetros: slot da amostra(b). tempo de aquisição(c), e posição de aquisição(d). A aquisição de dados da amostra pode ser repetida individualmente ou repetir a aquisição de todo as amostras disponíveis(e), podendo o início dessa aquisição ser imediata ou programada(f). O usuário pode setar a posição das amostras por prateleiras (0,1,2,3) ou pode inserir um valor customizado(d), respeitando a altura máxima interna da blindagem, conforme figura 22. O início da contagem e a inserção dos valores do tempo morto e tempo real ocorrem pelo usuário por meio do Genie2000.

Figura 22 - Tela principal de inserção de dados pelo usuário

A imagem mostra a janela 'Incluir' do software, com os seguintes campos e controles:

- (a)** Campo de texto para 'Lote:'.
- (b)** Campo de texto para 'Slot Amostra:'.
- (c)** Campo de texto para 'Tempo de Aquisicao (s):'.
- (d)** Grupo de opções para 'Prateleiras', incluindo 'Posição 0', 'Posição 1', 'Posição 2', 'Posição 3' e 'Customizada:' com um campo de texto adjacente.
- (e)** Campo de texto para 'Repetir Programação:'.
- (f)** Grupo de opções para 'Início', incluindo 'Imediata' (selecionada) e 'Programada'.

Outros campos visíveis incluem: 'Amostra:', 'Observacao:', 'Massa (g):', 'Tempo Morto (%)', 'Tempo Real (s):', 'Iniciar em:' (com data e hora) e 'Horário:'.

FONTE: (Autor da dissertação, 2019)

Ao termino da inserção dos parâmetros de cada amostra basta clicar em confirmar, onde essa programação fica alistada sequencialmente num *grid* de atividades e o usuário faz a próxima programação.

4.9 BANCO DE DADOS

O banco de dados será projetado em MySQL *community edition* – ou edição comunitária, ou seja, apenas para fins de estudos e/ou particulares, não sendo permitido o uso comercial. Ele irá armazenar os parâmetros recebidos do hardware da controladora e do Genie 2000. As restrições de capacidade de armazenamento serão determinadas pelo sistema operacional em uso no computador, ou seja, pela arquitetura em uso e o sistema de arquivos do sistema operacional instalado.

4.10 MECANISMO POSICIONADOR DO RECIPIENTE DA AMOSTRA

O sistema mecânico consiste de uma mesa de coordenadas (Figura 23) e um atuador linear, ambos fabricados pela empresa KALATEC S/A para este trabalho, ambos acionados por motor de passo, modelo NEMA 23, de altíssima precisão.

Figura 23 - Mesa de coordenadas.

FONTE: (KALATEC, 2018)

4.11 PAINEL DE CONTROLE

No painel de controle estão todos os componentes de acionamento, controle e alimentação do conjunto mecânico. Inicialmente um projeto da placa

de circuito impresso (*Printed Circuit Board* - PCB), tipo *Shield* foi idealizado no programa open-source Fritzing versão 0.9.3, disponível no site www.fritzing.com.org. No PCB os componentes eletrônicos são fixados por meio de solda e acoplada a uma placa *open-source* Arduino Mega 2560 R3, de 256KB memória *flash* (8KB usado no *bootloader*), velocidade do *clock* de 16MHz (EVANS, M.; NOBLE, J.; HOCHENBAUM, J., 2013). O sketch gravado na placa controladora Arduino Mega 2560 R3 foi desenvolvido em C++, na própria IDE Arduino - *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado e se comunica por meio da conexão via USB – *Universal Serial Bus* com o computador onde está instalado o Genie2000. O Arduino Mega 2560 R3 é uma placa de fácil aquisição e que dispõe de amplo conjunto de bibliotecas, possibilitando a interação com diversos sensores.

Foi utilizado para a alimentação do painel uma fonte de 12 volts de 30 Amperes para os controladores dos motores de passo e outra fonte de 5 volts de 2 Amperes para os sensores. Seu acionamento e seu desligamento ocorrem pelo usuário, através de botoeiras liga/desliga, obedecendo a norma técnica NR10, que no seu item 10.3.1 preconiza a obrigatoriedade do circuito a não voltar a ser energizado automaticamente, afim de não oferecer risco às pessoas ou equipamento. Assim, em caso da falta de energia, o sistema só será reiniciado com a interferência do usuário.

5 TESTES EFETUADOS E RESULTADOS OBTIDOS

Para que o sistema seja considerado robusto testes experimentais foram feitos. Para tanto, diversas fontes radioativas usando soluções calibradas (ou seja, com atividade conhecida) de ^{60}Co produzidas pelo Laboratório de Metrologia Nuclear do CRPq. A primeira rodada de testes foi usada para verificar a operacionalidade do sistema. Para tanto, diversas medidas foram feitas usando o trocador de amostras e não foi observado qualquer ocorrência de falha.

Caso o usuário observe a necessidade de inserir novas funcionalidades ou de alterar as existentes essas alterações poderão ser implementadas em novas versões. A segunda rodada de testes foi para se verificar a reprodutibilidade do posicionamento da amostra. Testes estatísticos de

reprodutibilidade foram aplicados após 10 medidas em uma posição para verificar se um índice aceitável foi obtido.

5.1 MESA DE AÇO E TESTES DE BLINDAGEM

Uma mesa de aço foi idealizada e confeccionada para comportar o movimentador de amostras ao lado do detector, bem como a blindagem das fontes colocadas no movimentador. Estimou-se que essa mesa deve suportar pelo menos uma massa total de aproximadamente 350 kg. Em virtude dessa aproximação, entre as fontes e o detector, a eficiência da blindagem para atenuar a possibilidade de interferência de raios gama foi calculada pela equação (ver na parte de blindagem) (OKUNO, 2010).

. Estimando-se a atenuação do chumbo variando a espessura e a energia do fóton, por meio dos valores de coeficiente de atenuação mássico obtidos com o software XCOM, foi montado o gráfico da figura 24. É possível ver que com 15 cm de chumbo, menos de 0,1% dos fótons de 3,5 MeV são transmitidos.

Figura 24 - Gráfico atenuação x espessura Pb

FONTE: (Autor da dissertação, 2019)

Além da blindagem, o pequeno ângulo sólido reduz ainda mais a chance de medidas. Aplicando a equação para a distância fonte-detector a 60 cm e o raio do detector de 3 cm, obtém-se a porcentagem de apenas 0,0625% dos fótons atingindo o detector. O produto dessas duas probabilidades leva a um fator redução de $6,24 \times 10^{-5} \%$. No caso de 6 fontes, tem-se $3,24 \times 10^{-4} \%$. Portanto com 6 fontes de 10 kBq cada, contando-se por 1 hora apenas 137 fótons atingirão o detector, o que não garante que eles sejam registrados no fotopico.

Foram realizados 3 testes experimentais para verificação da eficiência da blindagem e da geometria na supressão de fótons na região do carrinho dos suportes. Primeiro foi feita uma medida de *background* por um período de 24 h sem a presença de nenhuma fonte, e posteriormente uma medida com uma fonte de ^{137}Cs e ^{60}Co de atividades 343 e 34kBq, respectivamente, utilizando apenas a blindagem original do detector (uma parede de chumbo de 5 cm). A Figura 25 mostra os espectros obtidos pelas duas medidas.

Figura 25 - Espectro de medidas de BG e medida com fonte de Cs e Co utilizando 5 cm de blindagem de chumbo.

FONTE: (Autor da dissertação, 2019)

Verifica-se pela análise da Figura 18 que a espessura de chumbo foi insuficiente para proteger o detector da radiação gama dessas duas amostras, conforme destacado nos espectros, onde é possível verificar a presença dos fotopicos do ^{137}Cs e ^{60}Co .

Para medidas tipicamente aplicadas em Análise por Ativação Neutrônica (NAA) as maiores energias envolvidas são em torno de 2000 keV, o material disponível para blindagem constitui em 15 cm de chumbo e o fator de atenuação obtido para a energia de 2000 keV para essa quantidade de chumbo foi de 0,045%.

Após a determinação desse valor foi realizada a mesma medida com uma blindagem de 15 cm de chumbo entre as fontes e o detector, e nessas condições verificou-se que não houve interferência das fontes de cobalto e célio, conforme mostrado na Figura 26.

Figura 26 - Espectro de medidas de BG e medida com fonte de Cs e Co utilizando 15 cm de blindagem de chumbo.

FONTE: (Autor da dissertação, 2019)

Foi realizado um terceiro teste com uma situação real empregada na NAA, onde irradiou-se por 8 horas 6 amostras de aproximadamente 150 mg de

materiais de referência¹, e os espectros obtidos para tempo de contagem de 74000 s são apresentados na Figura 27.

Figura 27 - Espectro de medidas de BG e medida com fonte de Cs e Co utilizando 15 cm de blindagem de chumbo.

FONTE: (Autor da dissertação, 2019)

Verifica-se que a blindagem de 15 cm e a geometria do arranjo foram suficientes para minimizar problemas de interferências espectrais das amostras irradiadas, mostrando que para essa situação os 15 cm de blindagem foram suficientes.

5.2 BANCO DE DADOS

O banco de dados foi projetado em MySQL *community edition* (MySQL®, 2019) – ou edição comunitária, ou seja, apenas para fins acadêmicos e/ou particulares, não sendo permitido o uso comercial. Ele é o responsável pelo armazenamento dos parâmetros recebidos do hardware da

¹ NIST SRM 166 Oyster tissue (NIST); BCR 668 (Mussel Tissue-IRMM) eBCR 670 (Aquatic Plant-IRMM); IAEA 331 (Spinach) e IAEA 881.

controladora e do Genie 2000. As restrições de capacidade de armazenamento são determinadas pelo sistema operacional em uso no computador, ou seja, pela arquitetura em uso e o sistema de arquivos do sistema operacional instalado.

5.3 CUSTO DO IMAR

A Tabela 1 planifica os custos aproximados dos itens utilizados, separados por empresa ou item. Para a conversão da moeda americana, utilizou-se a cotação do dólar do dia 17 de outubro de 2019 (U\$1=R\$ 4,17). Os itens de menor valor foram agrupados e apresentados como 'diversos' na tabela.

Tabela 1 - Custo aproximado dos componentes essenciais para confecção da estação de monitoramento do detector HPGe.

Empresa/ Item	Valor R\$	Data	Cotação Dólar US\$	Valor US\$
Kalatec	R\$ 3.555,69	17/10/2019	4,17	R\$ 852,68
Omega	R\$ 621,50	17/10/2019	4,17	R\$ 149,04
Controladores	R\$ 320,00	17/10/2019	4,17	R\$ 76,74
Motor c/ fuso	R\$ 280,00	17/10/2019	4,17	R\$ 67,15
Arduino mega	R\$ 70,00	17/10/2019	4,17	R\$ 16,79
Painel	R\$ 100,00	17/10/2019	4,17	R\$ 23,98
Mesa /aço	R\$ 480,00	17/10/2019	4,17	R\$ 115,11
Motores blindagem	R\$ 300,00	17/10/2019	4,17	R\$ 71,94
Diversos	R\$ 500,00	17/10/2019	4,17	R\$ 119,90
Totais	R\$ 6.227,19			R\$ 1.493,33

FONTE: (Autor da dissertação, 2019)

6 CONCLUSÕES

A proposta deste trabalho foi desenvolver um mecanismo customizado para o arranjo denominado Canberra 6, localizado no Laboratório de Ativação Neurônica na CRPq no IPEN/CNEN, que movimenta e posiciona com precisão uma série de amostras irradiadas, para detectores HPGe, visando minimizar a exposição do usuário às radiações gama oriundas da amostra e proporcionar um conjunto de medidas com repetibilidade, inclusive de longa duração, sem a necessidade da presença de um operador.

Com respeito à construção do mecanismo, embora haja vários arranjos em funcionamento em diversos países do mundo, o dispositivo desenvolvido tem vantagens por ser baseado em componentes de fácil acesso. Esse projeto pode ser replicado para outros sistemas similares, desde que sejam feitas as adaptações necessárias. A maior limitação deste sistema é o mecanismo do movimentador de amostras, que só conseguiu comportar seis slots para as amostras, um número inferior ao número normalmente irradiado pelo usuário em cada experimento. Assim, em projetos futuros, um novo movimentador de amostrar para comportar até 10 ou 12 slots de amostras poderá ser redesenhado, sem alterar o espaço ocupado pelo atual movimentador e sua blindagem.

A interface foi idealizada para ser bastante intuitiva e de fácil visualização por parte do usuário, as funcionalidades na comunicação com o programa Genie2000 podem ser implementados à medida que os usuários utilizem o sistema e com isso identifiquem os requisitos funcionais do sistema que desejam tratar no processo de aquisição de dados das amostras analisadas. Vale ressaltar que esses implementos só são possíveis por se tratar de um instrumento com o código fonte aberto (*Open Source*)

O mecanismo de aproximação da amostra em relação ao detector HPGe, mostrou-se bastante preciso nos testes de implantação, sendo que não houve quaisquer diferenças na distância entre a amostra e o detector HPGe que pudesse ser aferida na escala de mm.

Uma serie de teste para aferir a precisão do mecanismo também é tópico para um novo estudo. Assim, testes de precisão, repetibilidade, fadiga entre outros, poderão ser executados.

7 APÊNDICE A - SUPORTE DA AMOSTRA

O suporte foi modelado pelo software 3D Builder, sendo exportado para o software Simplify3D, que tem a função de selecionar as configurações e gerar suas instruções em G-Code, conforme podemos observar na figura 28. Posteriormente a peça foi confeccionada em PLA (polilactida) por meio de impressora 3D de mesa modelo RepRap.

Figura 28 - a) Suporte da amostra; b) Diâmetro do suporte

a)

b)

FONTE: (Autor da dissertação, 2019)

O suporte possui a altura total de 56 mm (2), 36 mm de diâmetro externo (4) e 32 mm de diâmetro interno. A abertura frontal, de 27 mm de altura (3) e 34 mm de largura, facilita a inserção e remoção da “panelinha” dentro do suporte. Suas dimensões possibilitam o acondicionamento de amostra líquida, desde que esteja num recipiente adequado.

A parte superior da haste do suporte conta com uma área de 13 mm (1) para a atuação da garra do movimentador de amostras.

8 APÊNDICE B - MOVIMENTADOR DE AMOSTRAS

8.1 MOTOR DE PASSO

O motor de passo utilizado para o movimentador de amostras é da Kalatec automação Ltda., por ser uma empresa nacional líder no fornecimento de sistemas de movimento e equipamentos para a fabricação de máquinas. O produto que mais se ajustou ao projeto foi o motor de passo Nema 17 bipolar modelo KTC42HS48-1004-YAL8 com fuso de 8x8mm e curso linear de 340 mm e castanha de bronze (Figura 29).

Figura 29 - Motor de passo Kalatec

Fonte: (KALATEC, 2018)

Os motores de passo são por definição um dispositivo eletro-mecânico de posicionamento que converte um sinal digital em torque, rotacionando em incrementos precisos, conhecido como steps, uma carga. O motor de passo utilizado é do tipo magneto permanente e de construção bem simples, sendo sua constituição de um rotor, um estator, tampa de montagem, flange e rolamentos. O rotor é constituído de duas secções, cada seção possui 50 dentes. O estator, que é a parte externa do motor mede 48 mm de comprimento e 42 mm de altura e largura, onde internamente as bobinas são enroladas sobre os dentes do estator e estão conectadas aos pares. Quando a corrente atravessa um par de espiras, elas atrairão os dentes de polaridade opostas em cada extremidade do rotor. À medida que as bobinas são

energizadas alternadamente uma e depois duas espiras, o rotor se desloca num incremento de $1,8^\circ$ em cada estagio (Figura 30).

Figura 30 - Descrição técnica do motor de passo Kalatec

Fonte:(KALATEC 2018)

O conjunto movimentador de amostras foi construído sobre uma base de MDF branca com espessura de 13 mm e dimensões de 400x100 mm. Um conjunto de 3 peças completa o mecanismo de movimento do movimentador de amostras. As peças foram projetas no software gratuito 3DBUILDER, e posteriormente, confeccionado em PLA (polilactida) numa impressora 3D de mesa modelo RepRap. O conjunto é composto de um suporte dos eixos, um suporte do motor de passo com fuso e um suporte para o *rack* das amostras.

8.2 BASE DO SUPORTE

A base do suporte dos eixos possui um total de seis orifícios de 5 mm de diâmetro, dispostos em duas fileiras de 3 orifícios cada, servem para fixação da peça na base de MDF branca. A base possui uma espessura de 5 mm com densidade de 40% fornece uma excelente estrutura para peça. O quadrado de fixação do eixo liso possui 12 mm de espessura, conforme a Figura 31.

Figura 31 - Dimensões do suporte dos eixos

FONTE: (Autor da dissertação, 2019)

A extremidade do fuso do motor de passo é fixada na parte interna de um rolamento de esfera de dupla blindagem, modelo 608RS com 8 mm de medida interna e 22 mm de medida externa e 7 mm de profundidade (Figura 32a). O rolamento é encaixado dentro de um anel modelado numa das laterais do suporte de 22 mm de diâmetro (Figura 32b). Paralelo ao anel de fixação do rolamento um orifício de 8 mm para o encaixe do eixo liso de aço. O rolamento de esfera é encaixado no suporte plástico depois de projetado e impresso (figura 32c).

Figura 32 - a) Rolamento, b) Medidas rolamento e eixo liso, c) peça impressa e montada

a)

b)

c)

FONTE: (Autor da dissertação, 2019)

8.3 SUPORTE DO MOTOR DE PASSO

O suporte do motor de passo é similar ao suporte dos eixos, com a mesmas dimensões de base e lateral, dispõe da mesma disposição dos 6 furos de fixação. Na lateral do suporte o motor de passo é fixado com quatro parafusos de 3 mm e o eixo liso de 8 mm é fixado paralelo ao fuso do motor de passo. Um orifício de 24,5 mm ao centro da lateral serve para centralizar o motor de passo (Figura 33).

Figura 33 - Suporte do motor de passo

FONTE: (Autor da dissertação, 2019)

8.4 EIXO LISO LINEAR

Paralelo ao fuso do motor de passo, um eixo liso de aço com 8 mm de diâmetro e 350 mm de comprimento foi posicionado para fornecer estabilidade mecânica à mesa. O eixo liso linear de aço temperado e retificado com tolerância h6 garantido precisão e durabilidade ao sistema. Suas dimensões são de 8 mm de espessura e 342 mm de comprimento, sendo 2 mm mais comprido que o fuso linear ele encaixa com mais profundidade nos suportes do motor de passo e do suportes dos eixos como mostrado na Figura 34a. O eixo linear está posicionado perpendicularmente ao fuso do motor de passo e 2 rolamentos lineares de 8 mm de medida interna e 15 mm de medida externa tipo LM8UU e modelo L7BTT9YJ servem para sustentar a mesa do rack conforme é mostrado na Figura 34b. Sua utilização é amplamente conhecida em impressoras 3D. A figura 34c ilustra a montagem do eixo com o rolamento linear.

Figura 34 - a) Eixo linear, b) Rolamento linear, c) Conjunto eixo e rolamento

FONTE: (GTMAX, 2018)

8.5 SUPORTE PARA O RACK DAS AMOSTRAS.

O suporte para o rack das amostras é uma peça projetada no software gratuito 3DBUILDER, e posteriormente, confeccionado em PLA (polilactida) numa impressora 3D de mesa modelo RepRap. A peça plástica que está fixada

na parte superior da castanha do fuso do motor de passo e sobre os rolamentos lineares do eixo. Sobre ele uma peça de acrílico compõe o rack com 7 *slots* para o suporte da amostra conforme a Figura 35.

Figura 35 - Suporte do rack das amostras

FONTE: (Autor da dissertação, 2019)

8.6 CONJUNTO RACK DAS AMOSTRAS

O conjunto rack das amostras foi projetada no software gratuito 3DBUILDER (figura 36a), e posteriormente, confeccionado em uma peça de acrílico transparente de 10 mm de espessura, com 95 mm de largura, 38 mm de altura e 340 mm de comprimento, com 7 orifícios de 40 mm de diâmetro para o acondicionamento dos suportes das amostras (figura 36b). O conjunto suporte do *rack* das amostras é montado sobre o suporte do rack das amostras conforme figura 36c.

Figura 36 - Conjunto rack das amostras

FONTE: (Autor da dissertação, 2019)

8.7 SENSOR DE FIM DE CURSO E RESET

Dois sensores tipo foto interruptor estão instalados nas extremidades da base do movimentador de amostras atuando como fim de curso do *rack* das amostras conforme a Figura 37.

Figura 37 - Sensor foto interruptor

FONTE: (Autor da dissertação, 2019)

Um pequeno interruptor plástico fixado na parte inferior do suporte do *rack* aciona esses sensores. A finalidade desses sensores é garantir que o movimento do *rack* não extrapole seu curso projetado. Um pequeno circuito eletrônico foi prototipado conforme a figura 38, composto de um botão de acionamento e *leds* sinalizadores que indicam o status do reset no mecanismo.

acionado pelo interruptor plástico. Nesse momento a luz vermelha se apaga e uma luz verde acende, indicando que o movimentador de amostra está pronto para ser utilizado. A fim de proporcionar um acondicionamento apropriado e robustez ao circuito, uma pequena placa de circuito impresso (PCB) foi confeccionada conforme a Figura 40.

Figura 40 - Placa PCB do reset

FONTE: (Autor da dissertação, 2019)

Um conjunto de duas peças projetadas no software gratuito 3DBUILDER, e posteriormente, confeccionado em PLA (polilactida) numa impressora 3D de mesa modelo RepRap acondicionam o conjunto do circuito do reset conforme a figura 41.

Figura 41 - Conjunto reset montado

FONTE: (Autor da dissertação, 2019)

8.8 SENSOR DE PRESENÇA DO SUPORTE

Um sensor de distância laser, modelo VL53L0X, instalado ao lado do rack das amostras detectará a presença do suporte de amostras no slot antes que a garra feche, conforme com a figura 42. O sensor é composto de uma fonte de luz infravermelha, fina e invisível, e de um circuito para detectar quanto tempo à luz leva para atingir um objeto e retornar para o sensor. Caso o sensor detecte o slot selecionado como ausente da amostra, um alerta visual será apresentado pelo sistema na tela e o sistema interrompe o processo de movimentação até a intervenção do usuário. Esse processo supervisiona que os suportes das amostras estejam inseridos nos slots do rack.

Figura 42 - Conjunto reset montado

FONTE: (Autor da dissertação, 2019)

9 APÊNDICE C - PAINEL DE CONTROLE

Inicialmente um projeto da placa de circuito impresso (*Printed Circuit Board- PCB*), tipo *Shield* (Figura 43a) foi idealizado no programa open-source Fritzing versão 0.9.3.(FRITZING®,2019). OPCB, onde os componentes eletrônicos foram fixados por meio de solda, é acoplados a uma placa *open-source* Arduino Mega 2560 R3, de 256KB memória *flash* (8KB usado no

bootloader), velocidade do *clock* de 16MHz (EVANS, MARTIN; NOBLE, JOSHUA; HOCHENBAUM, JORDAN, 2013) (Figura 43b).

A programação da placa controladora Arduino Mega 2560 R3 foi desenvolvida em C++, na própria IDE Arduino – *Integrated Development Environment* e se comunica por meio da conexão via USB – *Universal Serial Bus* (MONK, 2013) com o computador onde está instalado o Genie2000. O Arduino Mega 2560 R3 (Figura 43c) é uma placa de fácil aquisição e que dispõe de amplo conjunto de bibliotecas, possibilitando a interação com diversos sensores.

Figura 43 - a) projeto da placa, b) placa confeccionada, c) placa Arduino Mega 2560

FONTE: (Autor da dissertação, 2019)

A placa Arduino Mega 2560, utilizada nesse projeto possui 54 portas digitais de entrada/saída de dados, 14 das quais podem fornecer saídas analógicas PWM, e 16 portas de entrada analógica. Com o intuito de fornecer uma visão clara da alimentação da placa e da utilização dessas portas, bem como os componentes à elas associadas, elaborou-se um diagrama geral das portas utilizadas, conforme apresentado na figura 44.

Figura 44 - Diagrama das portas do arduino mega2560

FONTE: (Autor da dissertação, 2019)

Os 4 motores de passo são acionados individualmente por controladores modelo TB6600, que mostrou-se de melhor dimensionamento de corrente e alimentação.

Foi utilizada para a alimentação do painel uma fonte de 12 volts de 30 Amperes para os controladores dos motores de passo e outra fonte de 5 volts de 2 Amperes para os sensores. Seu acionamento e seu desligamento obedecem à norma técnica NBR10, que preconiza a obrigatoriedade nos projetos elétricos de dispositivos de desligamento de circuitos que possuam

recursos para impedimento de reenergização do sistema, assim, em caso da falta de energia, o sistema só será reiniciado com a interferência do usuário. Um painel de aço protege os componentes de agentes externos, conforme já apresentado anteriormente.

O IMAR conforme anteriormente apresentado é composto por diversos componentes e interage com o Genie2000. A fim de apresentar a sequência dessas interações, foi elaborado um diagrama de atividades que apresenta de forma sucinta esse fluxo de ações, conforme figura 45.

Figura 45 - Diagrama de atividades do sistema

FONTE: (Autor da dissertação, 2019)

10 REFERÊNCIAS

ABNT, A. B. D. N. TÉCNICAS. **ISO /IEC NR-10: Segurança em Instalações e Serviços em Eletricidade**. Rio de Janeiro. 2004.

ALFASSI, Z. **Activation Analysis**. Boca Raton: CRC Press, 1990.

ANDEWERG, A. H.; WATTERSON, J.I. **A LARGE CAPACITY SAMPLE CHANGER FOR FULLY AUTOMATED GAMMA RAY SPECTROSCOPY**, Johannesburg, Vol. NS-27, n. 1, 1 February 1980.

CARVALHO, M.C.; SANDERS, C.; HOLLOWAY, C. HardwareX 4. **Auto-HPGe, an autosampler for gamma-ray spectroscopy using high-purity germanium (HPGe) detectors and heavy shields**, 19 agosto 2018. Disponível em: <<https://doi.org/10.1016/j.ohx.2018.e00040>>. Acesso em: 19 agosto 2019.

EVANS, M.; NOBLE, J.; HOCHENBAUM, J. **Arduino em ação**. Primeira Edição. ed. Greenwich, CT: Novatec, 2013.

EVANS, MARTIN; NOBLE, JOSHUA; HOCHENBAUM, JORDAN. **Arduino em ação**. Primeira Edição. ed. Greenwich, CT: Novatec, 2013.

FRIEDLANDER, M.; KENNEDY, J. W.; MILLER, J. M. J. **Nuclear and radiochemistry**. New York: ohn Wiley & Sons, 1981.

GERMANO, R.V. **Germanium Propriedades, Produção e Aplicações**. [S.I.]: Nova Science Publishers, Inc, 2012.

JÚNIOR, J. A. S; AMARAL, R.S.; SILVA, C. M.; MENEZES, R. S. C.; BEZERRA, J, D. **Estudo comparativo entre os detectores HPGe e NaI(Tl) na determinação de U-238, Th-232 e K-40 em amostras de solo. , 2009. ISSN 044102.**, IV, Abril 2009. p.1-16.

KALATEC S/A, Campinas. Disponível em: <<https://www.kalatec.com.br/mesa-linear-com-fuso-de-esferas/>>. Acesso em: 23 Maio 2019.

KNOLL, G.F. **Radiation detection and measurement**. 3ª Edição. ed. New York: John Wiley and Sons, Inc, 2000.

MCADAMS, V. D.; ATKINSON, S. D.; GAGE, S. J. **A Programmable and Automated System for Gamma and X-Ray Analysis**. [S.l.]. 1974.

MONK, S. **Programação com Arduino começando com Sketches**. Porto Alegre: Bookman, 2013.

MÜLLER, G.; DINIZ, A. C. G. C. **Entendendo a norma abnt iso/iec 17025:2005**. NBR. Uberlândia. 2007.

OKUNO, E. **Física das radiações**. São Paulo: Oficina de Textos, 2010.

PAVLOV, S. S.; DMITRIEV A.Y.; FRONTASYEVA M.V. **Automation system for measurement of gamma-ray spectra of induced activity for multi-element high volume neutron activation analysis at the reactor IBR-2 of Frank Laboratory of Neutron Physics at the joint institute for nuclear research.**, Dubna, 2006.

PIZZINI, S. **Point Defects in Group IV Semiconductors**. Millersville, PA: Materials Research Forum LLC, v. 10, 2017.

SECCO, M. **Desenvolvimento de dispositivo movimentador automatizado de amostras com vista à aplicação em medidas de radioisótopos que possuem curto tempo de meia-vida**. IPEN. São Paulo. 2016.

WATTERSON, A H ANDEWEG AND J I W. **A LARGE CAPACITY SAMPLE CHANGER FOR FULLY**, Johannesburg, February 1980.

11 ANEXOS

Códigos arduino

```
#include <Arduino.h>
#include "Motor.h"
#include "Adafruit_VL53L0X.h"
#include <Servo.h>
#include <Bounce2.h>

// ----- MOTOR TROCADOR DE AMOSTRAS -----//
#define pinPulso 5          // define Pulse pin
#define pinDirecao 6        // define Direction pin
#define pinEnable 7         // define Enable Pin
#define fimDeCursoDir 24     // define Fim de Curso Pin 22
#define fimDeCursoEsq 22     // define Fim de Curso Pin 24
#define ledPin 52           // led de sinalização
#define passos_por_loop 50 //50    // numero de passos por cada interação
/*
//-----MOTOR EIXO -----//
#define pinPulso 11          // define Pulse pin
#define pinDirecao 12        // define Direction pin
#define pinEnable 13         // define Enable Pin
#define fimDeCursoDir 32     // define Fim de Curso Pin 34 (Lado Motor)
#define fimDeCursoEsq 34     // define Fim de Curso Pin 32 (Lado Abertura)
#define ledPin 52           // led de sinalização
#define passos_por_loop 50 //numero de passos por cada interação
//-----MOTOR DO ELEVADOR-----//

#define pinPulso 8          // define Pulse pin
#define pinDirecao 9        // define Direction pin
#define pinEnable 10         // define Enable Pin
#define fimDeCursoDir 36     // define Fim de Curso Pin 36
#define fimDeCursoEsq 38     // define Fim de Curso Pin 38
#define ledPin 48           // led de sinalização
#define passos_por_loop 50    // numero de passos por cada interação
//-----MOTOR DO BLINDAGEM-----//
//
#define pinPulso 2          // define Pulse pin
```

```

#define pinDirecao 3          // define Direction pin
#define pinEnable 4          // define Enable Pin
#define fimDeCursoDir 33     // define Fim de Curso Pin 33
#define fimDeCursoEsq 35     // define Fim de Curso Pin 35
#define ledPin 31            // led de sinalização
#define passos_por_loop 50   // numero de passos por cada interação
*/
//-----painel trocador de amostras-----//

int led_Verm= 30;            // acesso indica que o rack está na posição 1 e operacional
int led_Verde=28;            // acesso quando o rack em movimento
int chave_Reset=26;          // botão vermelho para reset- pressionado posiciona o rack
para a amostra 1
//-----chave trocador -----//
#define chave_Trocador 44    // chave fim de curso lado do trocador de amostras
Bounce debouncer = Bounce(); // instancia o objeto Bounce
//-----Painel de leds -----//
#define LedVerdeEixo 52      // led Verde eixo
#define ledVerdeTrocador 50  // led Verde trocador
#define ledVerdeElevador 48  // led Verde elevador
#define ledVerdePresenca 46  // led Verde sensor presença amostra

#define LedVermelhoEixo 53   // led Verde eixo
#define ledVermelhoTrocador 51 // led Verde trocador
#define ledVermelhoElevador 49 // led Verde elevador
#define ledVermelhoAusencia 47 // led Verde sensor ausencia amostra

//----- LASER &GARRA -----//

#define ledLaser 39          // led laser
const int servoPin =A0;      // Servo motor
Servo garra;                 // cria um objeto da classe Servo
int num =0;                  // variável para a comunicação serial
int pos = 0;                  // variável para inicio da posição da garra
int var=195;                  // variável para final da posição da garra
//-----fim de curso -----//

int valDir,ValEsq;
char numero,valor;
int posicao_motor=0;
int posicao_amostra = 0;

```



```
// -----variaveis de tempo de execucao-----//
char buffer[80];

Motor motor_amostra(pinPulso,pinDirecao,pinEnable,fimDeCursoEsq,fimDeCursoDir,1000);

//--Motor motor_eixo(pinPulso,pinDirecao,pinEnable,fimDeCursoEsq,fimDeCursoDir,1600)--//
Motor motor_eixo(11,12,13,32,34,1200);

//Motor motor_elevador(pinPulso,pinDirecao,pinEnable,fimDeCursoEsq,fimDeCursoDir,1600)//
Motor motor_elevador(8,9,10,36,40,1200);

//Motor motor_blindagem(pinPulso,pinDirecao,pinEnable,fimDeCursoEsq,fimDeCursoDir,1600)
Motor motor_blindagem(2,3,4,35,33,1200);

// -----protocolo da comunicacao Arduino-----//

// -----comunicação teste-----//
const char COD_PING = '@';

// -----controle motor de passo ( que sobe e desce a prateleira )-----//
const char COD_POSICAO_AMOSTRA = 'a';
const char COD_POSICAO_EIXO = 'b';
const char COD_ELEVADOR = 'c';
const char COD_BLINDAGEM = 'd';
const char COD_LER_POSICAO = 'l';
const char COD_PARAR_MOTOR = 'p';
const char COD_POSICAO1='1';
const char LASER_LIG='x';
const char LASER_DESLIG='z';
const char GARRA_ABRIR = 'w';
const char GARRA_FECHAR = 'q';
// -----Constantes de verificação-----//
const char COD_ESPACO = ' ';
const char COD_FIM = '\n';
const String COD_VERSAO = "2.0.0";
const String COD_OK = "OK";
const String COD_P1= "POSIÇÃO 1";
const String LASER_LIGAR= "LASER LIGADO";
const String LASER_DESLIGAR= "LASER DESLIGADO";
const String AMOSTRA_PRESENTE= "AMOSTRA PRESENTE";
```

```

const String AMOSTRA_AUSENTE= "AMOSTRA AUSENTE !!";
const String GARRA_ABERTA = "Garra aberta";
const String GARRA_FECHADA = "Garra fechada";
//-----SENSOR DE DISTANCIA TROCADOR DE AMOSTRAS-----
----//
Adafruit_VL53L0X lox = Adafruit_VL53L0X();

void setup() {

//-----Inicia a Serial -----//
Serial.begin(9600);

//----- PAINEL DE LEDS -----//
pinMode(LedVerdeEixo, OUTPUT);    // led Verde eixo
pinMode(ledVerdeTrocador, OUTPUT); // led Verde trocador
pinMode(ledVerdeElevador, OUTPUT); // led Verde elevador
pinMode(ledVerdePresenca, OUTPUT); // led Verde sensor presença amostra

pinMode(LedVermelhoEixo, OUTPUT); // led Verde eixo
pinMode(ledVermelhoTrocador, OUTPUT); // led Verde trocador
pinMode(ledVermelhoElevador, OUTPUT); // led Verde elevador
pinMode(ledVermelhoAusencia, OUTPUT); // led Verde sensor ausencia amostra

pinMode(ledLaser,OUTPUT); // led laser do braço

debouncer.attach(chave_Trocador, INPUT_PULLUP);
debouncer.interval(25);
//pinMode(chave_Trocador,INPUT);    // chave fim de curso lado trocador

while (! Serial) {
  // delay(1);
}

Serial.println("Leitura de Presença da amostra");
if (!lox.begin()) {
  Serial.println(F("Falha no sensor Leitura amostra!"));
while(1);
}

```

```

Serial.println(F(" Leitura distancia"));

//-----Declaração dos pinos garra servo -----//
Serial.println("Digite q para abrir e w para fechar a garra");
garra.attach(servoPin); // attaches the servo on pin 9
//----- MOTOR AMOSTRAS-----//
// -----iniciar_motor();-----//

pinMode(ledPin, OUTPUT);
Serial.println("Teste Seleccion -> 6 amostras");
Serial.println("Digite o número do slot -> use números de 1 a 6");
Serial.println("DESEJA RESETAR O SELETOR? -> DIGITE 'r'");

// necessário para que o raspberry aceite a comunicação serial
// versão?
Serial.print(COD_OK);
Serial.print(COD_ESPACO);
Serial.print(COD_VERSAO);
Serial.print(COD_ESPACO);
Serial.print(COD_FIM);

//-----fim de curso -----//
pinMode(led_Verde,OUTPUT);
pinMode(led_Verm,OUTPUT);
pinMode(chave_Reset,INPUT);

}

int i;

int parseNumber()
{
    char r;
    int num = 0;
    bool negativo = false;
    while((r = buffer[i++]) != 0)

```

```

{
    switch(r)
    {
        case '-':
            negativo = true;
            break;
        case '0':
        case '1':
        case '2':
        case '3':
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':
            num *= 10;
    num += r - '0';
    break;
    default:
    if(negativo)
        return -num;
        else
            return num;
        break;
    }
}

if(negativo)
return -num;
else
return num;
}

int readline(int readch, char *buffer, int len)
{
    static int pos = 0;
    int rpos;

    if (readch > 0) {
        switch (readch) {
            case '\r': // Ignore CR

```

```

        break;
    case '\n': // Return on new-line
        rpos = pos;
        pos = 0; // Reset position index ready for next time
        return rpos;
    default:
        if (pos < len-1) {
            buffer[pos++] = readch;
            buffer[pos] = 0;
        }
    }
}

// No end of line has been found, so return -1.
return -1;
}

void ler_instrucao()
{
    i = 0;
    char cod = buffer[i++];
    int q_motor = -1;
    char data [60];
    switch(cod)
    {
        case COD_PING:
            Serial.print(COD_OK);
            Serial.print(COD_FIM);
            break;
        case COD_POSICAO_AMOSTRA:
            //----- pula o caractere de espaço-----//
            i++;
            motor_amostra.mover_para(parseNumber());
            //Serial.print(COD_OK);
            //Serial.print(COD_FIM);
            break;
        case COD_POSICAO_EIXO:
            // ----- pula o caractere de espaço-----//
            i++;
            motor_eixo.mover_para(parseNumber());
            //Serial.print(COD_OK);
            //Serial.print(COD_FIM);
    }
}

```

```

break;
case COD_ELEVADOR:
    //----- pula o caractere de espaço-----//
    i++;
    motor_elevador.mover_para(parseNumber());
    //Serial.print(COD_OK);
    //Serial.print(COD_FIM);
    break;
case COD_BLINDAGEM:
    //----- pula o caractere de espaço-----//
    i++;
    motor_blindagem.mover_para(parseNumber());
    //Serial.print(COD_OK);
    //Serial.print(COD_FIM);
    break;
case COD_LER_POSICAO:
    //Serial.print("STATUS");
    //Serial.print(motor_amostra.posicao);
    //Serial.print("S");
    //Serial.print(digitalRead(motor_amostra.fimDir));
    //Serial.print(digitalRead(motor_amostra.fimEsq));
    //Serial.print(";");
    //Serial.print(motor_eixo.posicao);
    //Serial.print("S");
    //Serial.print(digitalRead(motor_eixo.fimDir));
    //Serial.print(digitalRead(motor_eixo.fimEsq));
    //Serial.print(";");
    //Serial.print(motor_elevador.posicao);
    //Serial.print("S");
    //Serial.print(digitalRead(motor_elevador.fimDir));
    //Serial.print(digitalRead(motor_elevador.fimEsq));
    sprintf(data, "STATUS%dS%d%d;%dS%d%d;%dS%d%d;%dS%d%d"
, motor_amostra.posicao
, digitalRead(motor_amostra.fimDir)
, digitalRead(motor_amostra.fimEsq)
, motor_eixo.posicao
, digitalRead(motor_eixo.fimDir)
, digitalRead(motor_eixo.fimEsq)
, motor_elevador.posicao
, digitalRead(motor_elevador.fimDir)
, digitalRead(motor_elevador.fimEsq)

```

```

,digitalRead(chave_Trocador)
,motor_blindagem.posicao
,digitalRead(motor_blindagem.fimDir)
,digitalRead(motor_blindagem.fimEsq));
Serial.print(data);
    break;
    case COD_PARAR_MOTOR:
motor_amostra.parar();
motor_eixo.parar();
motor_elevador.parar();
motor_blindagem.parar();
Serial.print(COD_OK);
    Serial.print(COD_FIM);
    break;
    case LASER_LIG:
        digitalWrite(ledLaser,HIGH);
        Serial.println(LASER_LIGAR);
    break;
    case LASER_DESLIG:
        digitalWrite(ledLaser,LOW);
        Serial.println(LASER_DESLIGAR);
    break;
    case GARRA_ABRIR :
garra_abrir();
        Serial.println(GARRA_ABERTA);
    break;
    case GARRA_FECHAR :
garra_fechar();
        Serial.println(GARRA_FECHADA);
    break;
}
}
void loop()
{
    //-----chave de fim de curso do trocador -----//
    debouncer.update();

    if(debouncer.fell())
    {
        motor_elevador.parar();
    }
}

```

```

//-----SENSOR DISTANCIA TROCADOR -----//

/* VL53L0X_RangingMeasurementData_t measure;

lox.rangingTest(&measure, false);
int a=measure.RangeMilliMeter;
if (a<110 && a>50 )
{
    digitalWrite(ledVerdePresenca,HIGH);
digitalWrite(ledVermelhoAusencia, LOW);
    //Serial.print( AMOSTRA_PRESENTE);
}
else {
digitalWrite(ledVermelhoAusencia, HIGH);
    digitalWrite(ledVerdePresenca,LOW);
    //Serial.print( AMOSTRA_AUSENTE);
}

delayMicroseconds (1);
///copiado para a função amostra_Presente()fim do codigo

// -----CONTROLE DO MOTOR DO EIXO (MOVIMENTADOR) -----
---//
*/

valDir=digitalRead(fimDeCursoDir); // SENSOR DE FIM E CURSO DO LADO DO BOTÃO
RESET
ValEsq=digitalRead(fimDeCursoEsq); // SENSOR DE FIM E CURSO DO LADO DO MOTOR
DE PASSO

if(valDir == 1 ) // SENSOR FECHADO estava 1 mudei para 0
{
    digitalWrite(led_Verde,HIGH);//HIGH
    digitalWrite(led_Verm,LOW);

}

else if(digitalRead(chave_Reset)==LOW) ///BOTÃO PRESSIONADO
{
    digitalWrite(led_Verm,HIGH); //high
    digitalWrite(led_Verde,LOW); //LOW

```



```

motor_amostra.passos(-10);

motor_amostra.parar();

}

// -----se nao esta apertando o botao e nao esta no fim de curso-----//
if(digitalRead(chave_Reset)==HIGH)
{
    // -----tenta mover o motor em direcao ao objetivo-----//
    // -----aqui é que o motor mexe, o mover_para só dá o alvo.-----//

    if(!motor_amostra.mover(passos_por_loop))
    {
        if(!motor_eixo.mover(passos_por_loop))
        {
            if(!motor_elevador.mover(passos_por_loop))
            {
                motor_blindagem.mover(passos_por_loop);
            }
        }
    }
}

//-----MOTOR DO MOVIMENTADOR -----//

/*
// Aqui é a parte que lê o serial sem bloquear a execução
// isso é importante porque se a função mover() do motor não for chamada
// em um espaço curto de tempo, pode acontecer do motor não chegar a velocidade desejada
// ou se não chamar a função por um intervalo grande de tempo o que acontece é que o motor
para

// então aqui checa se chegou 1 byte ou mais na entrada serial
// e lê 1 byte e adiciona a variável buffer
// se o byte for um \n ( fim de linha ) passa pela função ler_instrucao()
*/
if(Serial.available() > 0) // se tem algo a ser lido
{
    // digitalWrite(redled, HIGH); // liga o led para indicar a comunicação chegando

```

```

if(readline(Serial.read(), buffer, 80) > 0) // le so mais um caractere, e retorna > 0 se tiver lido \n
{
    // interpreta a instrucao contida no buffer
ler_instrucao();
    //digitalWrite(redled, LOW);
}
}

void amostra_Presente()
{
    VL53L0X_RangingMeasurementData_t measure;
    lox.rangingTest(&measure, false);
    int a=measure.RangeMilliMeter;
    if (a<110 && a>50 )
    {
        digitalWrite(ledVerdePresenca,HIGH);
        digitalWrite(ledVermelhoAusencia, LOW);
        digitalWrite(ledVerdePresenca, HIGH);
    }
    else {
        digitalWrite(ledVermelhoAusencia, HIGH);
        digitalWrite(ledVerdePresenca,LOW);
        digitalWrite(ledVerdePresenca,LOW);
    }

}

//delayMicroseconds(1);
//delay(1);
void garra_fechar()
{
    Serial.println("Número 1 recebido - Fechando a garra");
    for (pos = 140; pos <= var; pos += 1)
    {
        garra.write(pos-15);
        delay(10);
    }
}

```

```

void garra_abrir()
{
    Serial.println("Número 2 recebido - Abrindo a garra");
    for (pos = var; pos >= 50; pos -= 1)//pos >=16
    {
        garra.write(pos-15);
        delay(5);
    }
}

```

Código Motor.cpp

```

/*
    Morse.cpp - Library for flashing Morse code.
    Created by David A. Mellis, November 2, 2007.
    Released into the public domain.
*/

#include "Arduino.h"
#include "Motor.h"

Motor::Motor(int _pinPulso, int _pinDirecao, int _pinEnable, int _fimEsq, int _fimDir, int _vel)
{
    pinPulso = _pinPulso;
    pinDirecao = _pinDirecao;
    pinEnable = _pinEnable;
    fimEsq = _fimEsq;
    fimDir = _fimDir;
    vel = _vel;

    pinMode (pinPulso, OUTPUT); //pino pulso - clock
    pinMode (pinDirecao, OUTPUT); // pino direção -
    pinMode (pinEnable, OUTPUT); // pino enable

    if(fimDir >= 0)
        pinMode (fimDir, INPUT );
    if(fimEsq >= 0)
        pinMode (fimEsq, INPUT);
    digitalWrite(pinEnable, LOW );

```

```

digitalWrite( pinDirecao, LOW );    // inicio do motor ( sentido horario (HIGH); sentido
antihorario(LOW))
digitalWrite( pinPulso,  LOW );    // Inicio dos passos do eixo
digitalWrite(fimDir,  HIGH );      // Sensor de fim de curso para reset carrocel
digitalWrite(fimEsq,  HIGH );      // Sensor de fim de curso para reset carrocel
}

```

```

bool Motor::bateu_fim(int direcao)
{
    // checagem do fim de curso
    if(fimDir >= 0)
    {
        if(direcao == LOW && digitalRead(fimDir) == 1)
        {
            return true;
        }
    }
}

```

```

if(fimEsq >= 0)
{
    if(direcao == HIGH && digitalRead(fimEsq) == 1)
    {
        return false;
    }
}
}

```

```

void Motor::_passos(int passos)
{
    int direcao = passos >= 0 ? HIGH : LOW;

    if(bateu_fim(direcao))
    {
        digitalWrite(pinEnable, LOW);
        return;
    }
}

```

```

digitalWrite(pinEnable,  HIGH );

```

```

digitalWrite(pinDirecao, direcao);

```

```

passos = passos >= 0 ? passos : -passos;
for(int i=0; i<passos; i++)
{
    // checagem do fim de curso
    if(fimDir >= 0) // fim de curso Direito ( ao Lado do botão do reset)
    {
        if(direcao == LOW && digitalRead(fimDir) == 1)
        {
            posicao = 0;
            parar();
            break;
        }
    }

    if(fimEsq >= 0) // fim de curso esquerdo
    {
        if(direcao == HIGH && digitalRead(fimEsq) == 1)
        {
            parar();
            break;
        }
    }

    // um passo
    digitalWrite( pinPulso, HIGH );
    delayMicroseconds(vel);
    digitalWrite( pinPulso, LOW );
    delayMicroseconds(vel);
    posicao += direcao == HIGH ? 1 : -1;
}

void Motor::passos(int passos)
{
    _passos(passos);
    digitalWrite(pinEnable, LOW);
}

```

```

void Motor::mover_para(int p)
{
    posicaoAlvo = p;
}

void Motor::parar()
{
    posicaoAlvo = posicao;
}

bool Motor::mover(int maximo)
{
    int posDiff = posicaoAlvo - posicao;
    if(posDiff == 0){
        digitalWrite(pinEnable,LOW);
        return false;
    }

    if(posDiff > maximo) _passos(maximo);
    else if(posDiff < -maximo) _passos(-maximo);
    else _passos(posDiff);

    return true;
}

```

Código Motor.h

```

/*
    Motor.h - Biblioteca para controlar um motor de passo
*/

#ifndef Motor_h
#define Motor_h

#include "Arduino.h"

class Motor
{
public:

```

```

Motor(int pinPulso, int pinDirecao, int pinEnable, int fimEsq, int fimDir,int vel);

    void passos(int passos);

    void mover_para(int posicao);

    bool mover(int maximo);

    bool bateu_fim(int dir);

    void parar();

    int pinPulso;

    int pinDirecao;

    int pinEnable;

    int fimEsq;

    int fimDir;

    int posicao;

    int vel;

    int posicaoAlvo;

private:

    void _passos(int passos);

};

#endif

```

Códigos C#

Código classe AcaoInterator

```

using System;

using System.Collections.Generic;

using System.Linq;

```

```

using System.Text;

using System.Threading.Tasks;

namespace Configurador.Models
{
    class AcaoIterator
    {
        List<Acao> acoes = new List<Acao>();
        Controlador controlador;

        int index;

        public string status { get; set; }
        public int idAmostra { get; }
        public string descricaoAmostra { get; }
        public string strDtIniciado { get; set; }
        public string strDtEncerrado { get; set; }
        public string strProgramado { get; set; }
        private Action<string> actionAtualizarStatus { get; }

        Action atualizar;

        public AcaoIterator(int idAmostra, string descricaoAmostra, Action atualizar, DateTime
            strProgramado, Controlador arduino)
        {
            this.idAmostra = idAmostra;
            this.descricaoAmostra = descricaoAmostra;
            this.status = "Nao Iniciado";
            this.atualizar = atualizar;
            actionAtualizarStatus = AtualizarStatus;
            if (strProgramado != DateTime.MinValue)
                this.strProgramado = strProgramado.ToLongDateString();
            else
                this.strProgramado = "Não Programado";
            controlador = arduino;
        }
    }

```



```

public void ExecutarAcoes()
{
    index = -1;

    strDtIniciado = DateTime.Now.ToShortDateString();

    foreach (var acao in acoes)
    {
        acao.tarefa.ExecutarAcao();
    }

    status = "Finalizado";

    strDtEncerrado = DateTime.Now.ToShortDateString();

    atualizar();
}

public void AtualizarStatus(string statusReceive)
{
    status = statusReceive;

    atualizar();
}

public void SelecionarAmostra()
{
    acoes.Clear();

    acoes.Add(new Acao
    {
        id = acoes.Count + 1,
        tarefa = new TarefaPosicionarNaAmostra(AtualizarStatus, controlador, idAmostra)
    });

    acoes.Add(new Acao
    {
        id = acoes.Count + 1,

```

```
    tarefa = new TarefaAbrirPegador(AtualizarStatus, controlador, idAmostra)
  });
```

```
    acoes.Add(new Acao
    {
        id = acoes.Count + 1,
    tarefa = new TarefaBaixarPegadorNoRack(AtualizarStatus, controlador, idAmostra)
  });
```

```
    acoes.Add(new Acao
    {
        id = acoes.Count + 1,
    tarefa = new TarefaFecharPegador(AtualizarStatus, controlador, idAmostra)
  });
```

```
    acoes.Add(new Acao
    {
        id = acoes.Count + 1,
    tarefa = new TarefaSubirPegador(AtualizarStatus, controlador, idAmostra)
  });
```

```
    acoes.Add(new Acao
    {
        id = acoes.Count + 1,
    tarefa = new TarefaPosicionarNoProvador(AtualizarStatus, controlador, idAmostra)
  });
```

```
    acoes.Add(new Acao
    {
        id = acoes.Count + 1,
    tarefa = new TarefaBaixarPegadorNoDetector(AtualizarStatus, controlador, idAmostra)
  });
```

```
acoes.Add(new Acao
{
    id = acoes.Count + 1,
    tarefa = new TarefaSubirPegador(AtualizarStatus, controlador, idAmostra)
});
```

```
acoes.Add(new Acao
{
    id = acoes.Count + 1,
    tarefa = new TarefaPosicionarNaAmostra(AtualizarStatus, controlador, idAmostra)
});
```

```
acoes.Add(new Acao
{
    id = acoes.Count + 1,
    tarefa = new TarefaBaixarPegadorNoRack(AtualizarStatus, controlador, idAmostra)
});
```

```
acoes.Add(new Acao
{
    id = acoes.Count + 1,
    tarefa = new TarefaAbrirPegador(AtualizarStatus, controlador, idAmostra)
});
```

```
acoes.Add(new Acao
{
    id = acoes.Count + 1,
    tarefa = new TarefaSubirPegador(AtualizarStatus, controlador, idAmostra)
});
```

```
acoes.Add(new Acao
{
    id = acoes.Count + 1,
```

```

    tarefa = new TarefaPosicionarNoProvador(AtualizarStatus, controlador, idAmostra)
        });
    }
}
}

```

Código da Classe TarefaPosicionadorNaAmostra

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace Configurador.Models
{
    class TarefaPosicionarNaAmostra : TarefaTemplate
    {
        const string posicionado_amostra = "01";

        public TarefaPosicionarNaAmostra(Action<string> atualizarStatus, Controlador controlador, int
            indentificadorDaAmostra)
            : base(atualizarStatus, controlador, indentificadorDaAmostra)
        {
        }

        }

        public override void ExecutarAcao()
        {

```

```

        Console.WriteLine("Posicionando na Amostra...");

        atualizarStatus("Posicionando Rack da Amostra");

        //Posicionar rack para retirada da amostra

        controlador.EnviarComando(controlador.motorMesa.name + " " +
        rackDeAmostras.GetSlotAmostra(indetificadorDaAmostra).posicao);

        atualizarStatus("Movimentando Eixo para amostra");

        controlador.EnviarComando($"{controlador.motorEixo.name} -100000");

        while (!controlador.motorEixo.fim_curso)

        {

        controlador.GetStatusMotores();

            Console.WriteLine($"Posicao Motor b: {controlador.motorEixo.posicaoAtual}");

        }

        }

    }
}

```

Código classe TarefaBaixarPegadorNoDetector

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace Configurador.Models
{
    class TarefaBaixarPegadorNoDetector : TarefaTemplate
    {
        public TarefaBaixarPegadorNoDetector(Action<string> atualizarStatus, Controlador controlador,
        int indentificadorDaAmostra)
            : base(atualizarStatus, controlador, indentificadorDaAmostra)
        {

```

```

    }

    public override void ExecutarAcao()
    {
        Console.WriteLine("Baixando garra no Detector....");
        atualizarStatus("Baixando garra no Detector");
        controlador.EnviaComando($"{controlador.motorElevador.name} 0");

        while (!controlador.motorElevador.inicio_curso)
        {
            controlador.GetStatusMotores();

            Console.WriteLine($"Posicao Motor c: {controlador.motorElevador.posicaoAtual} ");
        }
    }
}

```

Código classe FrmTarefas

```

using Configurador.Models;
using Syncfusion.Windows.Forms.Tools;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace Configurador
{
    public partial class FrmTarefas : Form
    {
        List<AcaoIterator> acaoIterators = new List<AcaoIterator>();

        Controlador arduino;

        public FrmTarefas()
        {
            InitializeComponent();

            arduino = new Controlador();

            arduino.porta = "COM3";

            arduino.taxaDeTransmissao = 9600;

            arduino.conectar();

            toolStripStatusLabelPort.Text = string.Format("Controlador: {0} | Porta: {1} | Status: {2}",
                "Arduino", arduino.porta, arduino.Status) ? "Conetado" : "Nao Conectado");

            AddAcoes();

            CarregarGrid();

            for (int i = 1; i < 6; i++)
            {
                cbSlotAmostra.Items.Add(i);
            }

            dtInicio.MinDate = DateTime.Now;

            cbSlotAmostra.SelectedIndex = 0;

        }

        public void CarregarGrid()
        {
            RecarregarGrid();

            dataGridViewAcoes.AutoGenerateColumns = false;

            dataGridViewAcoes.AutoSize = false;

            dataGridViewAcoes.DataSource = bindingSourceAcoes;
        }
    }
}

```

```

        DataGridViewColumn column = new DataGridViewTextBoxColumn();
        column.DataPropertyName = "idAmostra";
        column.Name = "Codigo da Amostra";
        column.AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
        dataGridViewAcoes.Columns.Add(column);

```

```

        column = new DataGridViewTextBoxColumn();
        column.DataPropertyName = "descricaoAmostra";
        column.Name = "Descricao da Amostra";
        column.AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
        dataGridViewAcoes.Columns.Add(column);

```

```

        column = new DataGridViewTextBoxColumn();
        column.DataPropertyName = "strProgramado";
        column.Name = "Programado?";
        column.AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
        dataGridViewAcoes.Columns.Add(column);

```

```

        column = new DataGridViewTextBoxColumn();
        column.DataPropertyName = "strDtIniciado";
        column.Name = "Inicio";
        column.AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
        dataGridViewAcoes.Columns.Add(column);

```

```

        column = new DataGridViewTextBoxColumn();
        column.DataPropertyName = "strDtEncerrado";
        column.Name = "Fim";
        column.AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
        dataGridViewAcoes.Columns.Add(column);

```

```

        column = new DataGridViewTextBoxColumn();
        column.DataPropertyName = "status";

```



```

column.Name = "Status do processo";
column.AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;
dataGridViewAcoes.Columns.Add(column);
    }

```

```

public void RecarregarGrid()
{
    bindingSourceAcoes.Clear();
    foreach (var item in acaoInterators)
    {
        bindingSourceAcoes.Add(item);
    }
}

```

```

public void AddAcoes()
{
    //acaoInterators.Add(new AcaoInterator(1, "Teste Amostra 1", AtualizarGrid));
    //acaoInterators.Add(new AcaoInterator(2, "Teste Amostra 2", AtualizarGrid));
    //acaoInterators.Add(new AcaoInterator(3, "Teste Amostra 3", AtualizarGrid));
}

```

```

private void rbTipolInicio_Checked(object sender, EventArgs e)
{
    if (rbProgramada.Checked)
    {
        dtInicio.Enabled = true;
        txtRepetir.Text = "1";
        txtRepetir.Enabled = false;
    }
    else
    {
        dtInicio.Enabled = false;
        txtRepetir.Enabled = true;
    }
}

```

```
    }
}
```

```
private void btnIniciar_Click(object sender, EventArgs e)
```

```
{
    arduino.SincronizarMotores();
    foreach (var item in acaoIterators)
    {
        item.SelecionarAmostra();
        item.ExecutarAcoes();
    }
}
```

```
private void AtualizarGrid()
```

```
{
    dataGridViewAcoes.Refresh();
}
```

```
public Panel GetPanel()
```

```
{
    return panelContent;
}
```

```
public string GetTitle()
```

```
{
    return Text;
}
```

```
private void button1_Click(object sender, EventArgs e)
```

```
{
    if (Validar())
    {
```

```

for (int i = 0; i < Int32.Parse(txtRepetir.Text); i++)
{
    acaoIterators.Add(new AcaoIterator
        (Int32.Parse(cbSlotAmostra.Items[cbSlotAmostra.SelectedIndex].ToString())
        , txtObservacao.Text
        , AtualizarGrid
        , rbProgramada.Checked ? dtInicio.Value : DateTime.MinValue
        , arduino));
}

```

```

RecarregarGrid();

```

```

    cbSlotAmostra.SelectedIndex = 0;
    txtObservacao.Text = string.Empty;
    //dtInicio.Enabled = false;
    rblmadiato.Checked = true;
    txtRepetir.Text = "1";
}
}

```

```

private bool Validar()
{
    if (cbSlotAmostra.SelectedIndex == -1)
    {
        MessageBox.Show("Selecione o slot da amostra!", "Erro ao programar", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }

    int qtdExecucao = 0;
    if (Int32.TryParse(txtRepetir.Text, out qtdExecucao))
    {
        if (qtdExecucao > 1 && rbProgramada.Checked)
        {

```

```
        MessageBox.Show("Tarefas programadas são configuradas individualmente e  
quantidade de execucao deve ser 1 por tarefa!", "Erro ao programar", MessageBoxButtons.OK,  
MessageBoxIcon.Error);
```

```
    return false;
```

```
    }
```

```
    }
```

```
else
```

```
{
```

```
        MessageBox.Show("Quantidade de Execucao é inválida!", "Erro ao programar",  
MessageBoxButtons.OK, MessageBoxIcon.Error);
```

```
    return false;
```

```
    }
```

```
return true;
```

```
}
```

```
private void DtInicio_ValueChanged(object sender, EventArgs e)
```

```
{
```

```
}
```

```
private void Label1_Click(object sender, EventArgs e)
```

```
{
```

```
}
```

```
private void CbSlotAmostra_SelectedIndexChanged(object sender, EventArgs e)
```

```
{
```

```
}
```

```
}
```

```
}
```

Código classe TarefaPosicionarNoProvador

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

namespace Configurador.Models
{
    class TarefaPosicionarNoProvador : TarefaTemplate
    {
        const string posicionado_provador = "10";

        public TarefaPosicionarNoProvador(Action<string> atualizarStatus, Controlador controlador, int
            indentificadorDaAmostra)
            : base(atualizarStatus, controlador, indentificadorDaAmostra)
        {
        }

        public override void ExecutarAcao()
        {
            Console.WriteLine("Posicionando no Detector....");
            atualizarStatus("Movimentando Eixo no Detector");
            //Posicionar rack para retirada da amostra
            controlador.EnviaComando($"{controlador.motorEixo.name} 0");
            while (!controlador.motorEixo.inicio_curso)
            {
                controlador.GetStatusMotores();
                Console.WriteLine($"Posicao Motor b: {controlador.motorEixo.posicaoAtual} ");
            }
        }
    }
}
```

```
}
```

Código classe Motor

```
using System;
using System.Threading;

namespace Configurador.Models
{
    public class Motor
    {
        public string name { get; protected set; }
        public string posicaoAtual { get; protected set; }
        public bool inicio_curso { get; protected set; }
        public bool fim_curso { get; protected set; }
        protected string resultMsgStatus { get; set; }
        public Motor(string name)
        {
            this.name = name;
        }

        public virtual void Sincronizar(Controlador controlador)
        {
            int irParaPosicao = -100000; //move para sentido negativo
            int auxPosicaoAtual = 0;
            int repetiuPosicao = 0;
            while (!inicio_curso)
            {
                controlador.EnviaComando($"{name}{irParaPosicao}");
                Thread.Sleep(1000);
            }
        }
    }
}
```

```
//Consulta status do motores até a posicao inicial ou até parar
```

```
while (!inicio_curso)
{
    controlador.GetStatusMotores();
    //se repetir posicao aguarda
    if (auxPosicaoAtual == Int32.Parse(posicaoAtual))
        ++repetiuPosicao;
    //se repetir mais 3x sai do loop e muda de direcao
    if (repetiuPosicao > 3)
    {
        repetiuPosicao = 0;
        irParaPosicao = (irParaPosicao * -1);
        controlador.RestartCommunication();
        Console.WriteLine("Mudou de direcao");
        break;
    }
    auxPosicaoAtual = Int32.Parse(posicaoAtual);
}
controlador.RestartCommunication();
}
```

```
publicvirtualvoidUpdStatusMotor(string receive)
{
    if (receive.Length> 6 && receive.Substring(0, 6).Equals("STATUS"))
    {
        receive = receive.Substring(6, receive.Length - 6);
        if (receive.Length> 0)
        {
            resultMsgStatus = receive;
            UpdatePosition();
        }
    }
}
```

```

        }
    }
protected virtual void UpdatePosition()
{
    }
}
}

```

Código classe BaseForm

```

using Syncfusion.Windows.Forms.Tools;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Configurador
{
    public partial class BaseForm : Form
    {
        TabControlAdv tabcontrol;

        public BaseForm()
        {
            InitializeComponent();
            TabMDIManager.TabControlAdded += TabMDIManager_TabControlAdded;
        }
    }
}

```



```
private void TabMDIManager_TabControlAdded(object sender,
TabbedMDITabControlEventArgs args)
```

```
{
    tabcontrol = args.TabControl;
}
```

```
private void sobreToolStripMenuItem_Click(object sender, EventArgs e)
```

```
{
    MessageBox.Show("Goodbye");
    Application.Exit();
}
```

```
private void MenuExecutarProcesso_Click(object sender, EventArgs e)
```

```
{
    pictureBox1.Visible = false;
    pictureBox2.Visible = false;
try
    {
int index = -1;
for (int i = 0; i < TabMDIManager.MdiChildren.Count(); i++)
    {
if (TabMDIManager.MdiChildren[i].GetType().Name == "FrmTarefas")
        index = i;
    }

if (index < 0)
    {
        FrmTarefas frmTarefas = new FrmTarefas();
        AbrirSubForm(frmTarefas);
    }
else
    tabcontrol.SelectedTab = tabcontrol.TabPages[index];
    }
```

```
catch (Exception ex)
```

```
    {
        MessageBox.Show("Erro: " + ex.Message);
    }

}
```

```
private void AbrirSubForm(Form form)
```

```
    {
        form.MdiParent = this;
        form.Show();
    }
```

```
private void menuItemConfigurar_Click(object sender, EventArgs e)
```

```
    {
        try
        {
            int index = -1;
            for (int i = 0; i < TabMDIManager.MdiChildren.Count(); i++)
            {
                if (TabMDIManager.MdiChildren[i].GetType().Name == "FrmConfiguracao")
                {
                    index = i;
                }
            }

            if (index < 0)
            {
                FrmConfiguracao frmConfiguracao = new FrmConfiguracao();
                AbrirSubForm(frmConfiguracao);
            }

            else
            {
                tabcontrol.SelectedTab = tabcontrol.TabPages[index];
            }
        }
        catch (Exception ex)
```

```
{  
    MessageBox.Show("Erro: " + ex.Message);  
}  
}
```

```
private void BaseForm_Load(object sender, EventArgs e)  
{  
  
}
```

```
private void GerenciarToolStripMenuItem_Click(object sender, EventArgs e)  
{  
  
}
```

```
private void Label1_Click(object sender, EventArgs e)  
{  
  
}
```

```
private void PictureBox2_Click(object sender, EventArgs e)  
{  
  
}  
}
```

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Diretoria de Pesquisa, Desenvolvimento e Ensino
Av. Prof. Lineu Prestes, 2242 – Cidade Universitária CEP: 05508-000
Fone/Fax(0XX11) 3133-8908
SÃO PAULO – São Paulo – Brasil
<http://www.ipen.br>

O IPEN é uma Autarquia vinculada à Secretaria de Desenvolvimento, associada à Universidade de São Paulo e gerida técnica e administrativamente pela Comissão Nacional de Energia Nuclear, órgão do Ministério da Ciência, Tecnologia, Inovações e Comunicações.