AUTARQUIA ASSOCIADA À UNIVERSIDADE DE SÃO PAULO

**UNIDADE AUTÔNOMA DE MONITORAMENTO, SINALIZAÇÃO E REGISTRO PARA O
SISTEMA DE SEGURANÇA DO IRRADIADOR MULTIPROPÓSITO DE COBALTO-60**

RICARDO HOVACKER BALDACONI

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

Orientador:

Prof. Dr. Fabio Eduardo da Costa

São Paulo

2017

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

**UNIDADE AUTÔNOMA DE MONITORAMENTO, SINALIZAÇÃO E REGISTRO PARA O
SISTEMA DE SEGURANÇA DO IRRADIADOR MULTIPROPÓSITO DE COBALTO-60**

RICARDO HOVACKER BALDACONI

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

Orientador:
Prof. Dr. Fabio Eduardo da Costa

Versão Corrigida
Versão Original disponível no IPEN

São Paulo
2017

“Dedico esse trabalho ao meu avô, Laszlo Hovacker (in memoriam). Desde cedo me acostumei a olhar meu avô, que na verdade foi um pai, como se ele fosse um padrão de comportamento. Ao longo de sua vida travou duras batalhas mas nunca virou o rosto à luta, nunca se deixou ficar no chão. É um orgulho tê-lo como meu avô, um privilégio ter tido seu carinho e uma honra ter aprendido a viver sob seus ensinamentos. Para sempre viverá em meu coração...

Dedico também à minha mãe, Beti Julieta Hovacker, que sempre esteve e sempre estará ao meu lado, me apoiando incondicionalmente. Só o teu coração sabe compreender quando preciso de algo, obrigado é muito pouco...

Também dedico à minha avó, Julianna Bana Hovacker, que sempre cuidou de mim quando eu precisei, uma avó que também posso chamar de mãe. Eu gostaria de lhe agradecer pelas inúmeras vezes que você me enxergou melhor do que eu sou...

Dedico ainda à minha companheira, Mariane Columbari Alves, que perdeu uma parcela de minha atenção no decorrer do desenvolvimento desta dissertação, mas sempre entendeu e me apoiou, sendo a principal incentivadora para a conclusão deste trabalho.”

AGRADECIMENTOS

Ao meu orientador, Dr. Fábio Eduardo da Costa, por todas as orientações, apoio, transmissão de conhecimento e amizade. Um orientador diferenciado, com extremo domínio técnico sem deixar de lado a humildade e parceria.

Ao coordenador, Antônio Carlos Lemos Carvalho, um ser humano evoluído, que transmite paz, justiça e transparência. Que enxerga e sabe extrair o que há de melhor nas pessoas. Obrigado pela compreensão e apoio de sempre.

À todas as pessoas que, de alguma maneira, me apoiaram durante o desenvolvimento deste trabalho.

UNIDADE AUTÔNOMA DE MONITORAMENTO, SINALIZAÇÃO E REGISTRO PARA O SISTEMA DE SEGURANÇA DO IRRADIADOR MULTIPROPÓSITO DE COBALTO-60

Ricardo Hovacker Baldaconi

RESUMO

O Irradiador Multipropósito de Cobalto-60 é uma unidade construída no Instituto de Pesquisas Energéticas e Nucleares, órgão da Comissão Nacional de Energia Nuclear. Esta instalação utiliza dos efeitos que a radiação gama produz ao interagir com um meio material e suas consequências. A radiação gama emitida pelo radioisótopo cobalto-60 é ionizante, com elevado poder de penetração, que ao interagir com os produtos, inclusive no interior de suas embalagens, transfere sua energia por meio de colisões aos elétrons dos átomos que constituem os produtos. Ao mesmo tempo que este processo de ionização é desejável em função de suas características deletérias aos microrganismos, a exposição indiscriminada às radiações ionizantes pelo ser humano ou animais apresentará danos e em doses elevadas poderão levar a morte. Diante destas circunstâncias, para garantir a segurança, os equipamentos utilizados para irradiação são construídos e operados sob normas rígidas de construção e operação.

O sistema de segurança do Irradiador, constituído por um gerenciamento eletrônico de intertravamento de portas e exposição das fontes radioativas, é feito simultaneamente por um controlador lógico programável (CLP) e uma lógica de relés. Todas as informações, obtidas através das entradas e saídas do Irradiador, são monitoradas pelo sistema de segurança e enviadas para um computador com um programa supervisor.

O propósito do trabalho foi a construção de uma Unidade Autônoma de Monitoramento, Sinalização e Registro para o Sistema de Segurança do Irradiador Multipropósito de Cobalto-60. O desenvolvimento deste equipamento permitiu o monitoramento e registro de eventos, até mesmo de tempos muito curtos, não detectáveis pelo CLP. Cada evento foi registrado em um cartão de memória, de forma a permitir que estes eventos possam ser posteriormente analisados em qualquer computador, mantendo todo e qualquer histórico de ocorrências. Este é um equipamento de monitoramento totalmente independente, não interferindo no funcionamento do sistema atual já homologado.

AUTONOMOUS MONITORING UNIT, SIGNS AND REGISTRATION FOR COBALT-60 MULTIPURPOSE IRRADIATOR SAFETY SYSTEM

Ricardo Hovacker Baldaconi

ABSTRACT

The Cobalto-60 Multipurpose Irradiator is a unit built at the Nuclear and Energy Research Institute, an agency of the National Nuclear Energy Commission. This installation uses the effects that gamma radiation produces when interacting with a material medium and its consequences. The gamma radiation emitted by the cobalt-60 radioisotope is ionizing, with high penetration power, which when interacting with the products, even inside its packages, transfers its energy by means of collisions to the electrons of the atoms that make up the products. At the same time as this ionization process is desirable because of its deleterious characteristics to the microorganisms, indiscriminate exposure to ionizing radiation by humans or animals will present damages and in high doses could lead to death. In view of these circumstances, to ensure safety, the equipment used for irradiation is constructed and operated under strict construction and operation standards.

The safety system of the Irradiator, consisting of an electronic management of door interlocking and exposure of the radioactive sources, is made simultaneously by a programmable logic controller (PLC) and a relay logic. All information, obtained through the inputs and outputs of the Irradiator, is monitored by the security system and sent to a computer with a supervisory program.

The purpose of the work was the construction of an Autonomous Monitoring, Signaling and Registration Unit for the Cobalto-60 Multipurpose Irradiance Safety System. The development of this equipment allowed the monitoring and recording of events, even of very short times, not detectable by the CLP. Each event has been recorded on a memory card, so that these events can be later analyzed on any computer, maintaining any history of occurrences. This is a fully independent monitoring equipment, not interfering with the operation of the current system already approved.

SUMÁRIO

	Página
1 INTRODUÇÃO	13
2 OBJETIVOS	18
3 FUNDAMENTAÇÃO TEÓRICA	19
3.1 Radiação	19
3.1.1 Tipos de Radiação	Erro! Indicador não definido.
3.1.3 Radiação Beta	20
3.1.4 Radiação Gama	21
3.2 Interação da radiação com a matéria	23
3.2.1 Efeito fotoelétrico ou absorção fotoelétrica	24
3.2.2 Efeito Compton	25
3.2.3 Produção de pares	27
3.3 Irradiadores	29
3.4 Irradiador Multipropósito de Cobalto-60 do IPEN	31
3.4.1 Principais características do Irradiador Multipropósito	32
3.4.2 Gerenciamento eletrônico do Sistema de Segurança	33
3.4.3 Código de controle do CLP no Sistema de Segurança	36
3.4.4 Interface homem-máquina (IHM)	39
3.5 Sistemas Microcontrolados	42
3.5.1 Microcontrolador	42
4 MATERIAIS E MÉTODOS	44
4.1 Materiais	44
4.2 Diretrizes do Projeto da Unidade Autônoma	46
5 RESULTADOS	49
5.1 Placa dos optoacopladores	49
5.2 Placa Controladora Principal	51
5.2.1 Estrutura do código do programa no microcontrolador principal	55
5.2.2 Estrutura do código do programa no microcontrolador secundário	56
5.3 Gerador de data e hora RTCC	59
5.4 Microcontrolador auxiliar e conexão com o cartão de memória	61
5.5 Interface com o operador	62

5.6	Montagem da Unidade Autônoma	64
5.7	Simulador de eventos em tempos curtos	65
5.8	Teste de detecção de eventos pelo CLP do Painel de Controle	67
5.9	Teste de detecção e registro de eventos pelo computador com o software supervisório	68
5.10	Teste de detecção e registro de eventos pela Unidade Autônoma	70
6	CONCLUSÕES.....	71
	REFERÊNCIAS BIBLIOGRÁFICAS.....	74
APÊNDICE	A - Código do microcontrolador auxiliar	74
	APÊNDICE B – Código do microcontrolador principal	77
APÊNDICE	C – Código do microcontrolador secundário	100

LISTA DE FIGURAS

FIGURA 1:	Diagrama em blocos do Sistema de Segurança do Irradiador Multipropósito de Cobalto-60.....	15
FIGURA 2:	Diagrama em blocos da instalação da Unidade Autônoma junto ao Sistema de Segurança do Irradiador Multipropósito de Cobalto-60.....	17
FIGURA 3:	Gráfico das regiões onde predominam os três principais tipos de interação da radiação com a matéria em função da energia da radiação gama incidente e do número atômico do material sujeito a esta radiação.....	23
FIGURA 4:	Representação esquemática do efeito fotoelétrico.....	24
FIGURA 5:	Representação esquemática do efeito Compton.....	25
FIGURA 6:	Representação esquemática do efeito Produção de Pares.....	27
FIGURA 7:	Esquema de um irradiador gama categoria IV pela AIEA.....	28
FIGURA 8:	Diagrama esquemático do Irradiador Multipropósito do IPEN.....	29
FIGURA 9:	Árvore de falhas do sistema de segurança do Irradiador Multipropósito do IPEN.....	30
FIGURA 10:	Diagrama simplificado das conexões entre o processo e o CLP.....	33
FIGURA 11:	Diagrama em blocos do funcionamento de um CLP.....	33
FIGURA 12:	Diagrama de controle da porta, alarme e sistema de exaustão.....	35
FIGURA 13:	Diagrama de controle das fontes radioativas.....	36
FIGURA 14:	Diagrama da lógica do início das operações.....	36
FIGURA 15:	Tela do programa supervisor, na condição de procedimento de partida.....	38

FIGURA 16: Tela do programa supervisorio, registrando o estado das botoeiras de emergência na posição correta e porta de serviço aberta.....	38
FIGURA 17: Tela do programa supervisorio apresentando o registro dos alarmes ou eventos ocorridos.....	39
FIGURA 18: Tela do programa supervisorio com as condições gerais do Irradiador.....	39
FIGURA 19: Diagrama simplificado da Unidade Autônoma de Monitoramento, Sinalização e Registro desenvolvida para o Sistema de Segurança do Irradiador Multipropósito de Cobalto-60.....	45
FIGURA 20: Diagrama elétrico de um opto acoplador com informações sobre seu funcionamento.....	47
FIGURA 21: Placa dos optoacopladores da Unidade Autônoma. Este conjunto garantiu a isolamento galvânica entre os equipamentos e a direcionalidade dos sinais do Painel de Controle para a Unidade Autônoma.....	47
FIGURA 22: Esquema elétrico de parte da placa controladora principal, com o PIC18F8722 que é microcontrolador principal e circuitos de oscilação, reset e gravação ICSP.....	48
FIGURA 23: Trecho do esquema elétrico da placa controlador principal, exibindo a entrada para alimentação do circuito.....	49
FIGURA 24: Trecho do esquema elétrico da placa controlador principal, exibindo as conexões para o circuito do cartão de memória (a), o circuito gerador de data e hora (b) e a comunicação serial com o display (c).....	49
FIGURA 25: Trecho do esquema elétrico da placa controladora principal, exibindo o acionamento do relé para alarme externo (a), o acionamento do alarme sonoro local (b) e os LEDs de sinalização (c).....	50
FIGURA 26: Esquema elétrico dos conectores disponíveis na placa controladora principal...	50
FIGURA 27: Trecho do esquema elétrico da placa controladora principal, exibindo a comunicação com o microcontrolador secundário.....	51
FIGURA 28: Vista da placa controladora principal com todos os componentes que a compõe já soldados.....	52

FIGURA 29: Fluxograma do código do programa armazenado no microcontrolador principal.....	53
FIGURA 30: Diagrama em blocos dos dispositivos que compõe o sistema secundário para detecção de travamento do microcontrolador principal.....	55
FIGURA 31: Fluxograma do microcontrolador principal, trecho da interação com o microcontrolador secundário.....	55
FIGURA 32: Fluxograma do código do programa armazenado no microcontrolador dá-rio.....	56
FIGURA 33: Gerador de relógio com o circuito integrado DS1307, visto de ambas as faces	
FIGURA 34: Arquivo em DATALOG.TXT gravado pela Unidade Autônoma em um cartão de memória tipo SD Card.....	59
FIGURA 35: Telas da interface com o operador da Unidade Autônoma.....	60
FIGURA 36: Conjunto das placas e periféricos que constituem a Unidade Autônoma.....	61
FIGURA 37: Diagrama em blocos, o arranjo desenvolvido para a simulação de eventos utilizado nos testes da Unidade Autônoma.....	63

LISTA DE QUADROS

QUADRO 1: Porcentagem de detecção de eventos no CLP, do Painei de Controle, em função da duração do tempo do evento aplicado.....	64
QUADRO 2: Porcentagem de registros de eventos no computador em função da duração do tempo do evento aplicado.....	66
QUADRO 3: Porcentagem de registros de eventos no computador em função da duração do tempo do evento aplicado.....	67
QUADRO 4: Comparativo de porcentagem de registros de eventos nos sistemas isolados em função da duração do tempo do evento aplicado.....	68

1 INTRODUÇÃO

O Irradiador Multipropósito de Cobalto-60 é uma unidade construída no Instituto de Pesquisas Energéticas e Nucleares, órgão da Comissão Nacional de Energia Nuclear. Foi projetado e construído pelo Centro de Tecnologia das Radiações do próprio Instituto, com tecnologia nacional com financiamento da FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo e presta uma importante função de dar assistência técnica à iniciativa privada no uso de irradiadores para desenvolvimento de processos de radioesterilização em:

- Produtos médicos descartáveis, próteses, cosméticos, aditivos e componentes para indústrias farmacêuticas;
- Meios de cultura, soluções e ingredientes para laboratórios de prestação de serviços e pesquisas;
- Implantação de um banco de tecidos biológicos: ossos, peles e membranas diversas;
- Alimentos destinados a pacientes de risco.

Esta instalação utiliza dos efeitos que a radiação gama produz ao interagir com um meio material e suas consequências. Além das possibilidades de pesquisa, a utilização comercial da radiação está presente em vários processos industriais e diversos produtos. Santos, em sua dissertação [1], cita como exemplos de processos tradicionais, a esterilização de produtos da área da saúde, reticulação, cura e degradação de polímeros, irradiação de alimentos, e a indução de cor em gemas. O autor ainda reporta novas aplicações que estão surgindo como reação as necessidades globais, como no campo ambiental, onde a radiação ionizante tem sido utilizada para degradar compostos orgânicos em estruturas mais simples e desta forma diminuir o seu tempo de degradação. Nos últimos anos, a irradiação de tecidos humanos para transplante também foi pesquisada e hoje é aplicada de forma rotineira. Outras aplicações, envolvendo novos materiais de tecnologia avançada, como óxido de grafeno e nanopartículas também estão sendo estudadas [2-5].

A radiação gama emitida pelo radioisótopo cobalto-60, feixes de elétrons e os raios X produzidos em aceleradores de elétrons são radiações ionizantes, que ao interagir com

os produtos, inclusive no interior de suas embalagens, transfere sua energia através de colisões aos elétrons dos átomos que constituem os produtos. Este processo provoca a produção de grande quantidade de átomos excitados ou grupo de átomos eletricamente carregados conhecidos como íons. Os produtos da ionização: íons, elétrons secundários, átomos e moléculas excitadas irão também perder ou transferir sua energia para os átomos a sua volta, no caso o produto que está sendo irradiado, de modo a reestabelecer um equilíbrio [6]. Um dos resultados deste processo de transferência de energia é a inativação de microrganismos pela radiação, a qual é ocasionada em parte, pela ação da colisão direta da radiação com regiões sensíveis da célula e parcialmente, pela ação indireta, via formação de radicais químicos altamente ativos, produzidos no líquido da célula pela radiação. No caso da ação direta, a radiação ioniza uma parte da molécula de DNA, uma enzima ou qualquer outro componente vital da célula, levando-a a um estado de falência ou na inibição da sua reprodução. Pela via indireta, a radiação provoca nas moléculas de água presente nos microrganismos, a formação de radicais livres tais como: $\text{OH}^\bullet$, $\text{H}^\bullet$ e moléculas como H_2O_2 . Essas espécies químicas, sendo altamente reativas, podem interagir com os componentes vitais dos microorganismos causando indiretamente danos letais [6].

Ao mesmo tempo que este processo de ionização é desejável em função de suas características deletérias aos microrganismos, a exposição indiscriminada às radiações ionizantes pelo ser humano ou animais apresentará os mesmos danos e em valores elevados poderão levar a morte. Diante destas circunstâncias, os equipamentos utilizados para irradiação são construídos e operados sob normas rígidas de construção e operação.

O Irradiador Multipropósito de Cobalto-60 do IPEN foi projetado atendendo a Norma CNEN NN 6.02 [7] para Licenciamento de Instalações Radioativas bem como para o sistema de intertravamento e segurança a Norma Basic Safety Standards 115 [8] e a Specific Safety Guide N° SSG-8 [9], além de possuir um Plano de Radioproteção aprovado pela CNEN de acordo com a Norma CNEN NN 3.01 [10].

O sistema de segurança do Irradiador Multipropósito de Cobalto-60, constituído por um gerenciamento eletrônico de intertravamento de portas e exposição das fontes radioativas, é feito simultaneamente por um controlador lógico programável (CLP) modelo S7-200 da Siemens [11] e uma lógica de relés, dentro do Painel de Controle e Segurança.

A partir de um conjunto de informações dos sensores da instalação, muitas delas com funções de redundância, o sistema de segurança permite ou não a abertura da porta de serviço ou exposição das fontes. As informações de todos os sensores, chaves e atuadores

são monitoradas e controlados pelo sistema de segurança e seus estados enviados de forma serial, via comunicação EIA-232 [12], para um computador pessoal com plataforma Windows® a um programa supervisorio, com nome comercial Indusoft Web Studio® [13], que proporciona, além do acompanhamento de todo o processo de forma sinótica em um monitor, o registro de todos os eventos no disco rígido do computador. O diagrama de blocos do sistema de segurança é demonstrado na Figura 1.

FIGURA 1: Diagrama em blocos do Sistema de Segurança do Irradiador Multipropósito de Cobalto-60

O gerenciamento eletrônico do sistema de segurança tem-se mostrado eficiente e até a presente data não apresentou qualquer falha que houvesse comprometido a segurança. O CLP, junto com a lógica de relés, tem sempre tomado as decisões corretas assegurando a devida proteção radiológica dos operadores e a integridade das fontes radioativas. Entretanto, ao longo dos anos de operação (início em 2004), apresentou cerca de uma dezena de falhas no sistema de registro de eventos, que não foram registrados quando ocorridos de forma transitória em tempos muito curtos, na ordem de mili ou micro segundos.

A deficiência foi encontrada no processo de envio das informações, via comunicação serial (EIA-232), entre o CLP e o programa supervisorio residente no computador. Quando a falha ocorria em um tempo muito curto o CLP sempre tomava a decisão correta mas o processo de registro, que tinha que atravessar os tempos de acesso aos programas na plataforma Windows®, perdia a informação. Desta maneira, sem os dados do evento ocor-

rido registrados, não é possível analisar ou mesmo identificar a falha posteriormente. Mesmo sabendo que o CLP vem tomando as ações corretas impedindo qualquer tipo de incidente, os estudos sobre as falhas para melhorias contínuas, o aumento da segurança e a otimização de manutenções pontuais ficam todos comprometidos devido a essa deficiência.

Atualmente existem CLP's da linha *Safety* [14], como a CPU S7-1200F da Siemens [15], que permite, em conjunto com periféricos adicionais, o registro e armazenamento de informações e coleta de dados. Para esta linha de produtos mais atual, a aquisição de dados pode ainda ser feita através de módulos adicionais de entrada e saída, como por exemplo, os módulos *Fail-safe Digital Input* (entrada digital à prova de falhas), ou simplesmente F-DI [16] e *Fail Safe Digital Output* (saída digital à prova de falhas), ou simplesmente F-DO [17], ambos fabricados pela Siemens.

Entretanto, como o CLP S7-200 não é mais comercializado, o código atual residente no CLP que constitui o sistema de segurança do Irradiador Mutipropósito de Cobalto-60 do IPEN, não poderá ser reaproveitado, devendo ser completamente refeito devido a incompatibilidade das plataformas de desenvolvimento.

Desta maneira, a substituição da instalação atual por novos equipamentos exigiria, além de um longo tempo de implementação, um alto investimento financeiro bem como interferiria nos processos atuais do irradiador que teria sua produção paralisada para que a substituição pudesse ser realizada. Indo além, pode-se citar também a necessidade de nova homologação do sistema pela CNEN que demanda toda uma nova documentação para a instalação.

Para suplantar as dificuldades mencionadas foi proposto o desenvolvimento de uma unidade autônoma microcontrolada junto ao gerenciamento eletrônico do sistema de segurança. A grande vantagem dos sistemas microcontrolados é que estes sistemas podem ser desenvolvidos de forma customizada, dedicados à aplicação para a qual são projetados, a um custo extremamente baixo [18].

A Unidade Autônoma proposta foi configurada para registrar em um cartão de memória todos os eventos ocorridos, mesmo em tempos muito curtos, de forma a permitir que estes possam ser posteriormente analisados em qualquer computador. Para não interferir no sistema de segurança atual, a Unidade Autônoma foi instalada de forma paralela ao Painel de Controle e Segurança, ou seja, não houve qualquer modificação ou alteração nas instalações atuais e funciona de forma totalmente independente e apenas observadora, ou seja apenas registrando os eventos, sem interferir no controle.

Na Figura 2 é mostrado, no formato de diagrama de blocos, o local onde a unidade foi instalada. Pode-se perceber sua independência e sua característica apenas observadora das informações que chegam ao CLP através dos sensores do Irradiador.

FIGURA 2: Diagrama em blocos da instalação da Unidade Autônoma junto ao Sistema de Segurança do Irradiador Multipropósito de Cobalto-60

A unidade também espera a tomada de decisão correta do CLP e, em caso de uma decisão incorreta, além de também registrar este evento, sinaliza a ocorrência com um alarme independente e notifica em um display com interface própria.

2 OBJETIVOS

Desenvolver e construir uma Unidade Autônoma de Monitoramento, Sinalização e Registro para o sistema de segurança do Irradiador Multipropósito de Cobalto-60.

Registrar eventos locais, com intervalos de curta duração, na ordem de mili e micro segundos, na própria Unidade Autônoma.

Validar os resultados registrados na Unidade Autônoma e os comparar com os registros atuais feitos pelo computador.

Instalar a Unidade paralelamente ao sistema de segurança do Irradiador Multipropósito de Cobalto-60 no Centro de Tecnologia das Radiações do IPEN, sem interferir nas instalações já homologadas, aumentando a confiabilidade no registro de falhas em tempos curtos, otimizando processos preventivos de manutenção e segurança.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão reunidos temas essenciais por meio de uma abordagem resumida visando estabelecer uma relação entre os conceitos fundamentais que envolvem este trabalho, servindo como base para o entendimento e justificativa do projeto.

3.1 Radiação

As radiações (partículas ou ondas) podem ser não ionizantes ou ionizantes [19 -20]. Radiação ionizante ou não ionizante é a transferência direta de energia de uma fonte emissora até um ponto do espaço, independente do meio físico presente. Em física, o termo radiação refere-se usualmente a partículas e campos que se propagam (transferindo energia) no espaço, preenchido ou não pela matéria [19, 21, 23].

A radiação é dita não ionizante, quando sua energia não é suficiente para arrancar elétrons dos átomos. Neste caso, pode acontecer a excitação do átomo, onde elétrons são levados a camadas mais externas do átomo, sem serem ejetados.

A ionização acontece quando a energia da radiação incidente sobre um material é suficiente para arrancar elétrons dos seus átomos. Esta propriedade é que torna a radiação ionizante tão útil, pois ao mesmo tempo que é altamente deletéria à medida em que quebra ligações químicas, danifica células ou produz radicais livres, pode desde que adequadamente utilizada, produzir os desejados efeitos de esterilização ou mudança de propriedades [19- 21].

3.2 Fontes e tipos de radiação

A radiação pode ser de natureza particulada ou eletromagnética [19-20]. Nesse contexto, a radiação de natureza particulada é caracterizada por sua carga, massa e velocidade: pode ser carregada ou neutra, leve ou pesada, lenta ou rápida. Prótons, nêutrons e elétrons ejetados de átomos ou núcleos atômicos são exemplos de radiação particulada [19-21].

Já a radiação eletromagnética possui características físicas que incluem o comprimento e a frequência de onda. Estas características definem uma energia a esta radiação e permitem sua identificação [20-23].

As radiações podem ser emitidas por elementos químicos com núcleos atômicos instáveis (radioisótopos), que podem ser encontrados na natureza ou produzidas por meio de reações específicas em aceleradores de partículas, tubos de raio X, conversores de raio X ou reatores nucleares [23-24]. Neste trabalho, a radiação é proveniente de radioisótopos de cobalto-60 que são emissores de elétrons (partículas beta) e radiação eletromagnética (radiação gama).

3.2.1 Radiação Beta

As partículas beta são elétrons rápidos ou pósitrons (elétrons com carga positiva) emitidos por um núcleo radioativo. Possuem energias que variam de zero até um valor máximo que é característico do radioisótopo [21, 25].

A diferença de energias é explicada pelo fato de que, no decaimento beta, a energia total (máxima) é dividida entre as partículas beta e os antineutrinos, que também são emitidos. Os antineutrinos não possuem carga, massa e nem sofrem interação significativa com a matéria, sua existência foi postulada para permitir a conservação de energia, momento e spin no decaimento beta.

Em sua passagem pela matéria as partículas beta perdem energia predominantemente por colisões inelásticas com os elétrons [20-23]. Entretanto, devido ao fato da partícula beta e o elétron com o qual colide ter a mesma massa, a partícula beta pode perder até metade de sua energia em uma única colisão e pode ser desviada de sua trajetória. O desvio da trajetória também pode ocorrer quando esta partícula passa próxima ao núcleo atômico, devido as forças coulombianas. As partículas beta não têm um alcance fixo nos materiais, mas sim uma distância máxima de penetração ou alcance máximo. Apresentam baixa penetrabilidade e no caso das fontes de cobalto-60 utilizadas em irradiadores, o encamisamento de aço inoxidável que reveste as fontes já absorve todo este tipo de radiação.

A emissão da radiação beta é o mecanismo mais frequente de um núcleo efetuar o seu decaimento e acontece quando um núcleo atômico apresenta números excedentes de

prótons ou nêutrons. Dependendo do tipo de partícula em excesso no núcleo, acontecerá a emissão de um elétron (beta -), emissão de um pósitron (beta +) ou captura eletrônica [26].

3.2.2 Radiação Gama

A radiação gama ou raios gama são um tipo de radiação eletromagnética produzida em processos de decaimento nuclear. Por ser de origem nuclear (diferentemente dos raios-X, que são de origem atômica) recebem o nome de radiação gama. São altamente energéticas devido à sua elevada frequência e, conseqüentemente, baixo comprimento de onda [21, 23, 26].

Inicialmente era usual distinguir a radiação gama dos raios-X pela sua energia (os raios-X eram menos energéticos). No entanto, atualmente a distinção já não é efetuada desta forma, pois consegue-se produzir raios-X mais energéticos do que muitos raios gama, como nos aparelhos médicos de radiografia ou conversores de raio-X. A distinção entre estes tipos de radiação eletromagnética é efetuada através da sua origem: os raios-X são produzidos por excitação dos elétrons de camadas atômicas interiores para níveis energéticos mais elevados e posterior regresso ao estado fundamental (libertação de energia sob a forma de raios-X), enquanto a radiação gama provém do núcleo atômico, pois após a emissão de partículas alfa (que é um átomo de hélio duplamente ionizado) ou beta, o núcleo fica com um excesso de energia, que é libertada sob a forma de radiação gama [19-20, 23].

Os raios gama foram descobertos em 1900 pelo físico e químico francês Paul Villard quando estudava a radiação emitida por uma amostra de rádio. A radiação emitida incidia numa placa fotográfica protegida por uma fina camada de chumbo, suficiente para travar o avanço das partículas alfa. Villard mostrou que a radiação que atravessava a camada de chumbo era de dois tipos diferentes. Um deles era defletido quando submetido a um campo magnético, identificado como a radiação beta, descoberta um ano antes pelo físico e químico neozelandês Ernest Rutherford. No entanto, o segundo tipo de radiação não sofria qualquer tipo de deflexão, pelo que não deveria ter carga elétrica associada. Além disso, a radiação desconhecida tinha um poder penetrante muito superior à radiação alfa e beta e que Villard não conseguiu identificar. Efetivamente, foi Rutherford, no seguimento da descoberta de Villard, que identificou a radiação como o terceiro tipo de

radiação decorrente do decaimento radioativo e designou-a, em 1903, por radiação gama [20-22, 27].

A radiação gama, dos três tipos de radiação resultantes do decaimento nuclear (alfa, beta e gama), é a que tem o maior poder penetrante. Embora seja a que tem o menor poder de ionização, o seu elevado poder penetrante faz da radiação gama a mais perigosa para o ser humano.

Enquanto as radiações alfa e beta, geralmente já são absorvidas pela superfície da pele, a radiação gama consegue penetrar e, até, atravessar o corpo, causando danos a nível celular por onde passa [27].

A alta penetrabilidade da radiação gama e suas propriedades ionizantes permite sua aplicação em várias áreas, tais como [1, 19-20]:

- Esterilizar alimentos, acervos de livros, quadros e obras de arte, equipamento médico e de higiene para descontaminação ou esterilização.
- Trabalhos com gemas de pedras preciosas, permitindo a alteração de alguns parâmetros, como por exemplo a cor e opacidade da pedra.
- Em medicina é utilizada no tratamento de certos tipos de câncer (elimina as células cancerígenas) e, também, em diagnósticos (é administrada ao doente uma pequena quantidade de um radionuclídeo, que através de detectores específicos permite seu acompanhamento).

3.2.3 Radiação Alfa

As partículas α (alfa) são núcleos de hélio carregadas positivamente, emitidas por núcleos radioativos com energias discretas que são características do radioisótopo.

Ao atravessarem a matéria, as partículas α perdem energia principalmente por colisões inelásticas com os elétrons que se encontram em seu caminho, promovendo a excitação e a ionização de átomos e moléculas [21-22]. A grande diferença de massa entre o elétron e a partícula α faz com que ela perca uma pequena parte de sua energia e, praticamente, não seja desviada de sua trajetória durante a colisão com o elétron. Assim, como consequência as partículas α são gradativamente freadas como resultado de um grande número de colisões e percorrem uma trajetória retilínea.

A energia perdida por uma partícula α em sua passagem pela matéria produz um considerável número de íons e moléculas excitadas. Entretanto, o número de reações químicas que se segue não depende somente do número de espécies reativas formadas, mas

também da sua concentração no meio e, que por sua vez, depende da velocidade com que a energia é perdida [22-23].

3.3 Interação da radiação com a matéria

A radiação interage com a matéria de diferentes maneiras, variando de acordo com o tipo desta radiação, energia e composição do meio em que ela incide [21, 23, 28].

Considerando apenas as radiações ionizantes de natureza eletromagnética, as interações do fóton gama (partícula elementar mediadora da forma eletromagnética) ou X com a matéria podem ser divididas em efeito fotoelétrico, efeito Compton, produção de pares, espalhamento Rayleigh e reações fotonucleares. Entretanto, devido a maior parte das aplicações terem os três primeiros efeitos como os mais relevantes, somente estes serão abordados. A produção de pares consta desta parte teórica por ser um dos três processos principais da interação da radiação gama ou X, mas deve ser esclarecido que para a radiação gama emitida pelo radioisótopo cobalto-60, este processo não acontece devido a sua energia ser insuficiente para sua ocorrência [26].

De acordo com a energia da radiação incidente e do número atômico do material na qual incide, ocorre um tipo predominante de interação desse fóton com a matéria, como pode ser visto na Figura 3 e que serão discutidos na sequência.

FIGURA 3: Gráfico das regiões onde predominam os três principais tipos de interação da radiação com a matéria em função da energia da radiação gama incidente e do número atômico do material sujeito a esta radiação. Figura modificada [26]

3.3.1 Efeito fotoelétrico ou absorção fotoelétrica

Neste processo, um fóton de energia $h\nu$ interage com um átomo do material, transferindo toda sua energia a um dos elétrons orbitais e desaparecendo completamente. No seu lugar, o elétron, também conhecido como fotoelétron, é ejetado de uma das camadas do átomo, geralmente da camada K, com energia cinética E_c dada por [21,26].

$$E_c = h\nu - E_b$$

Em que:

$h\nu$ representa a energia do fóton incidente,

E_b é a energia de ligação do elétron ao seu orbital (energia que foi dispendida para desfazer a ligação do elétron ao átomo).

O elétron ejetado com energia E_c agora vai perdendo energia progressivamente pelo material provocando outras ionizações. Como este elétron é o mais fortemente ligado ao átomo, seu poder de ionização acaba sendo o maior entre os elétrons removidos por outros processos [23, 26].

A probabilidade de ocorrência (seção de choque) do efeito fotoelétrico é proporcional à Z^5 onde Z é o número atômico do material sujeito à radiação. Para energias $h\nu$ levemente superiores a E_b , para um dado átomo, a seção de choque é máxima e depois decresce rapidamente com o incremento de $h\nu$. [26, 28].

Com a ejeção de um elétron do átomo, há um rearranjo dos elétrons entre camadas, que consiste no preenchimento da vacância eletrônica formada, acompanhada da emissão de um ou vários raios X ou elétrons Auger. Este processo de rearranjo do ponto de vista de ionização de outros elétrons é pouco importante e não é levado em conta neste trabalho [21, 26, 33].

O efeito fotoelétrico é o mecanismo predominante de absorção para fótons de baixa

energia [21, 26]. Na Figura 4 é mostrado esquematicamente o efeito fotoelétrico.

FIGURA 4: Representação esquemática do efeito fotoelétrico. Figura modificada de [32]

3.3.2 Efeito Compton

O processo de interação por espalhamento Compton ocorre entre o fóton incidente e um elétron de orbitas mais externas do material onde o fóton incidiu. É o mecanismo de interação predominante para energias $h\nu$ médias, e pode ser tratado como um choque elástico entre o fóton e o elétron [26, 28].

No espalhamento Compton, o fóton transfere parte de sua energia para o elétron (assumindo que este esteja em repouso inicialmente), que é então ejetado (elétron de recuo). Um novo fóton, de menor energia aparece emergindo com um ângulo θ em relação à direção do fóton original [26, 28]. Na Figura 5 é mostrado esquematicamente o efeito

Compton.

FIGURA 5: Representação esquemática do efeito Compton. Figura modificada de [32]

Percebe-se que a energia do fóton incidente é repartida, sendo uma parte convertida na energia cinética do elétron ejetado e o restante na energia do fóton espalhado, que pode vir a interagir com outros elétrons presentes no meio [21, 26, 28].

Quando a energia do fóton incidente e o ângulo formado pelo fóton espalhado são conhecidos, é possível calcular a energia do fóton espalhado por meio da equação seguinte [26]:

$$h\nu' = \frac{h\nu}{1 + \frac{h\nu}{m_0c^2}(1 - \theta)}$$

Em que:

$h\nu'$ é a energia do fóton espalhado

$h\nu$ é a energia do fóton incidente

θ é o ângulo do fóton espalhado em relação à direção incidente

m_0c^2 é a energia do elétron em repouso (0,511 MeV)

Como ilustrado na Figura 3, o efeito Compton é predominante com fótons de energias intermediárias, faixa em que a energia de ligação E_b é desprezível quando comparada com a energia do fóton. A probabilidade do espalhamento Compton depende do número de elétrons disponíveis como espalhadores fazendo com que a probabilidade deste efeito ocorrer aumente linearmente com o número atômico do material na qual a radiação está incidindo [21, 26].

3.3.3 Produção de pares

O processo de produção de pares consiste na conversão da energia em matéria, de um fóton com o núcleo atômico, pela interação com o seu campo coulombiano, produzindo um elétron e um pósitron. Para que a produção de pares seja energeticamente possível, a energia do fóton deve exceder em duas vezes o equivalente em energia da massa de repouso do elétron (0,511 MeV). Todo o excesso de energia do fóton acima de 1,02 MeV, requerida para formar o par, se transforma em energia cinética do pósitron e do elétron [23, 26].

Em média, o pósitron possui energia cinética maior que a do elétron devido à ação coulombiana exercida pela carga positiva do núcleo atômico e as cargas do par elétron (-) e pósitron (+). Nessa interação o núcleo acelera o pósitron e desacelera o elétron [21, 23, 28].

O pósitron e o elétron perderão energia por ionização ao percorrer a matéria. Há uma probabilidade finita de que eles percam parte da energia por radiação Bremsstrahlung. Quando as partículas atingem o repouso, o elétron permanece na matéria coexistindo com os outros elétrons passando a pertencer a algum orbital disponível. Quanto ao pósitron, após transmitir por colisões sua energia cinética ao meio, interage com um elétron produzindo dois fótons de aniquilação, diametralmente opostos, cada um com energia de 0,511 MeV correspondente a energia da massa de repouso do elétron (m_0) relacionada pela fórmula clássica [23, 26]. Na Figura 6 é mostrado esquematicamente o efeito de Produção de Pares.

FIGURA 6: Representação esquemática do efeito Produção de Pares. Figura modificada de [32]

A produção de pares é predominante para energias elevadas e mais favorável para elementos com alto número atômico e proporcional a Z^2 [21, 23, 26].

3.4 Exposição e doses de radiação

A dose exposição é uma grandeza que exprime a carga elétrica produzida pela radiação ionizante, por unidade de massa de ar, em condições de pressão e temperaturas normais. A unidade do Sistema Internacional (SI) para a dose exposição é Coulomb / kg e é definida apenas para o ar e não pode ser usado para descrever a dose em seres humanos. Como definição geral usa-se o Gy (Gray) que a energia depositada por uma radiação ionizante em uma determinada massa, atendendo assim qualquer material sobre um campo de radiação. Especificamente para seres vivos se utiliza o Sv (Sievert) que além de levar em conta a energia depositada, considera também um fator de ponderação que leva em conta o tipo de radiação ionizante e o órgão que recebeu [29].

A exposição normal dos indivíduos deve ser restringida de tal modo que nem a dose efetiva nem a dose equivalente nos órgãos ou tecidos de interesse, causadas pela possível combinação de exposições originadas por práticas autorizadas, excedam o limite de dose especificado a seguir. Esses limites de dose não se aplicam às exposições médicas.

Os limites de dose anuais para o corpo inteiro para o indivíduo ocupacionalmente exposto é de 20 mSv e para um indivíduo do público 1 mSv [30].

3.4.1 Efeitos das radiações ionizantes na saúde humana

Azevedo [31] cita em seu trabalho que os efeitos biológicos das radiações ionizantes podem ser estocásticos ou determinísticos. Os efeitos estocásticos causam uma alteração aleatória no DNA de uma única célula que no entanto, continua a reproduzir-se e levam à transformação celular. Os efeitos hereditários são estocásticos. Não apresentam limiar de dose e o dano pode ser causado por uma dose mínima de radiação. O aumento da dose somente aumenta a probabilidade e não a severidade do dano. A severidade é determinada pelo tipo e localização do tumor ou pela anomalia resultante. No entanto, o organismo apresenta mecanismos de defesa muito eficientes. A maioria das transformações neoplásicas não evolui para câncer. Quando este mecanismo falha, após um longo período de latência, o câncer então, aparece. Os efeitos são cumulativos: quanto maior a dose, maior a probabilidade de ocorrência. Quando o dano ocorre em célula germinativa, efeitos hereditários podem ocorrer.

Ainda em seu trabalho, Azevedo [31] cita que os efeitos determinísticos levam à morte celular. A probabilidade de ocorrência e a severidade do dano estão diretamente relacionadas com o aumento da dose. Quando a destruição celular não pode ser compensada, efeitos clínicos podem aparecer, se a dose estiver acima do limiar. Por ex. 3-5 Gy eritema, 20 Gy necrose.

3.5 Irradiadores

Do ponto de vista operacional, as instalações de irradiação que utilizam raios gama, aceleradores de elétrons e raios X são similares. Basicamente os produtos são transferidos por sistemas automatizados de transporte para o interior da câmara de irradiação, onde são expostos por um determinado tempo e a seguir transferidos para fora, encerrando o processo. Nestas instalações as fontes são envoltas por espessas paredes de concreto que tem por finalidade a segurança radiológica dos operadores [33].

A Figura 7 apresenta um irradiador genérico que utiliza cobalto-60, sendo que a principal diferença com relação a um acelerador de elétrons é a blindagem com água. Com cobalto-60, as fontes radioativas são mergulhadas na piscina quando há necessidade de intervenção dentro da câmara de irradiação por parte dos operadores (manutenção, inspeção). No caso dos aceleradores a alta tensão da máquina é desligada cessando a fonte de radiação [33].

FIGURA 7: Esquema de um irradiador gama categoria IV pela AIEA. Figura modificada de [9]

Estes irradiadores atendem normas internacionais e são classificados segundo a Agência Internacional de Energia Atômica (AIEA) como categoria IV [8, 9], onde a fonte radiativa é armazenada em uma piscina com água que exerce a função de blindagem para a radiação. Para a Comissão Nacional de Energia Nuclear (CNEN), estes irradiadores gama estão classificados no Grupo I [7] para instalações de grande porte que utilizam fontes seladas em processos industriais induzidos por radiação. O termo seladas vem do processo de construção das fontes radioativas que são revestidas por aço inox, de forma estanque, para que na possibilidade de algum contato com a fonte radioativa, este seja apenas com o encamisamento de aço e não com o material radioativo em si. Este processo de fabricação também permite que as fontes fiquem submersas em água quando não usadas, sem que isso propicie alguma possibilidade de contaminação desta água. Assim, nesse tipo de irradiador, as fontes ficam armazenadas no fundo da piscina quando o equipamento está fora de uso e

são suspensas para a câmara de irradiação, por meio de um sistema mecânico apropriado, durante a irradiação de produtos [33].

3.6 Irradiador Multipropósito de Cobalto-60 do IPEN

O Irradiador Multipropósito do IPEN é do tipo compacto, não apresenta labirinto e a entrada/saída de produtos é feita através de uma porta giratória o que não permite, de forma intencional, a entrada de uma pessoa para o interior da câmara. Por sua vez, a porta de serviço é um ponto crítico do sistema de segurança pois a sua abertura indevida, com as fontes expostas, traria possibilidades de altas taxas de exposição à radiação dos operadores e pessoas no local, devido a não existência do labirinto. Como esclarecimento, o labirinto é construído de forma a trazer a taxa de radiação, nas portas de entrada e saída de produtos e de serviço, aos níveis ambientais, mas a sua implementação demanda em maiores áreas construídas e grande quantidade de concreto gasto na obra devido as espessuras elevadas das paredes do labirinto [33]. Um diagrama do Irradiador Multipropósito do IPEN é representado na Figura 8:

FIGURA 8: Diagrama esquemático do Irradiador Multipropósito do IPEN [33]

Durante o processamento por radiação a câmara de irradiação fica fechada por meio de sistemas de intertravamento e segurança. Como em toda instalação radioativa, o sistema de segurança trabalha de forma a evitar uma exposição à radiação do operador ou pessoas próximas. Este cuidado pode ser analisado através da árvore de falhas para uma exposição intencional como é demonstrado na Figura 9:

FIGURA 9: Árvore de falhas do sistema de segurança do Irradiador Multipropósito do IPEN. Figura modificada de [33]

3.6.1 Principais características do Irradiador Multipropósito

Algumas das principais características do Irradiador Multipropósito de Cobalto-60 construído no IPEN são [33]:

- Capacidade total licenciada pela CNEN: 37 PBq (1 MCi), Categoria IV;
- Atividade inicial de operação: 3.407,7 TBq (92.099 Ci, em 16/11/04);
- Piscina: 7,0 m de profundidade e 2,7 m de diâmetro;
- Portas de concreto: Uma deslizante, para montagem e manutenção da instalação, com 4,00 m (comprimento) x 4,00 m (altura) x 2,45 m (largura) e uma giratória, para entrada e saída dos produtos na câmara de irradiação, com 2,65 m (diâmetro) x 4,00 m (altura);
- Parede de concreto: 1,80 m de espessura (densidade = 2,35 g/cm³);

A instalação do Irradiador auxilia projetos de pesquisa e desenvolvimento do IPEN e outras instituições acadêmicas na área das aplicações da radiação em processos industriais por meio de suas diversas aplicações, como [33]:

- Esterilização de produtos médicos e farmacêuticos, tecidos biológicos para implantes, materiais utilizados em laboratórios e alimentos para pessoas com problemas imunológicos;
- Aumento do prazo de validade de alguns alimentos secos, sementes, vegetais, tubérculos e frutas;
- Controle de disseminação de insetos em frutas, sementes, flores ou folhas;
- Beneficiamento de pedras como topázios e quartzos;
- Desenvolvimento de novos irradiadores, dispositivos de irradiação, detectores e sensores de radiação e de novos materiais poliméricos.

As fontes de cobalto-60 são armazenadas em bastidores submersos no fundo da piscina abaixo do piso do irradiador e são removidas do fundo da piscina quando é desejado expor os materiais à radiação gama proveniente das fontes. O movimento destas fontes, bem como a abertura de portas é controlada, via gerenciamento eletrônico, pelo sistema de segurança, que garante condições seguras para a operação do Irradiador Multipropósito [1, 33].

3.6.2 Gerenciamento eletrônico do Sistema de Segurança

O gerenciamento eletrônico do Irradiador Multipropósito está a cargo de uma lógica de relés, que são chaves eletromecânicas, e um controlador lógico programável, definido pela *International Electrotechnical Commission (IEC)*, como o sistema eletrônico operando digitalmente, projetado para uso em um ambiente industrial, que usa uma memória programável para a armazenagem interna de instruções orientadas para o usuário, para implementar funções específicas, tais como, lógica sequencial, temporização, contagem e aritmética, para controlar, através de entradas e saídas digitais ou analógicas, vários tipos de máquinas ou processos.

Um controlador lógico programável, conhecido por sua sigla CLP e pela sua expressão inglesa *PLC (Programmable Logic Controller)*, é um computador especializado, de alta confiabilidade, baseado em um microprocessador que desempenha funções de

controle de diversos tipos e níveis de complexidade. Geralmente as famílias de controladores lógicos programáveis são definidas pela capacidade de processamento de um determinado número de pontos de entradas e ou saídas [34].

Em um sistema típico, toda a informação dos sensores é concentrada no controlador (CLP) que, de acordo com o programa em sua memória, define o estado dos pontos de saída conectados a atuadores. Os CLP's tem capacidade de comunicação de dados via canais seriais e paralelos, com isto podem ser supervisionados por computadores formando sistemas de controle integrados. Os canais de comunicação nos CLP's permitem que os mesmos sejam conectados à interface de operação ou interface homem-máquina (*IHM*), computadores, outros CLP's e até mesmo com unidades de entradas e saídas remotas. Cada fabricante estabelece um protocolo para fazer com que seus equipamentos troquem informações entre si [34, 35].

Os CLP's estão muito difundidos nas áreas de controle de processos ou de automação industrial, como nas indústrias do tipo contínuo, produtoras de líquidos, materiais gasosos e em linhas de montagem, como na indústria de automóveis [34]. Na Figura 10 é mostrado um diagrama simplificado das conexões entre a máquina e ou processo e o CLP.

FIGURA 10: Diagrama simplificado das conexões entre o processo e o CLP [35]

Dentro de um CLP existe o módulo da unidade central de processamento que é onde ficam o microprocessador, as memórias e os registradores que executam as operações lógicas do código de controle. Este código tem um tempo de execução, chamado de ciclo de varredura ou *scan time* que é contado desde a leitura das entradas, processamento até o acionamento das saídas [35]. Este código forma um laço fechado como pode ser observado pelo diagrama em blocos do funcionamento de um CLP, apresentado na Figura 11.

FIGURA 11: Diagrama em blocos do funcionamento de um CLP [35]

Para que tudo ocorra de maneira correta, ao ser ligado, o CLP executa um conjunto de instruções de inicialização [36], tais como:

- Limpeza das memórias imagens de entradas e saídas;
- Teste de escrita e leitura de memória;
- Teste de executabilidade do programa de controle;
- Execução de rotinas de inicialização, limpeza de registros auxiliares, limpeza do *display* e preparação do teclado se existir IHM.

Após estes procedimentos a unidade central de processamento executa, em laço fechado, o código de controle ininterruptamente até encontrar uma interrupção de *software* ou *hardware*. Os CLPs utilizados nestas aplicações atendem inclusive, especificações militares com alta confiabilidade, assegurando que só após a execução correta da rotina determinada, ou seja, as condições iniciais de funcionamento do equipamento, neste caso o Irradiador Multipropósito, serão passíveis de execução. Deve ser observado que o tempo de execução do processamento em laço fechado do CLP utilizado nesta instalação é da ordem de 60 μ s, como será demonstrado mais adiante nesta dissertação, assim a cada período deste, todas as condições de operação da instalação foram verificadas.

Apesar do tempo reduzido de execução, existe a possibilidade de que um evento com duração inferior a este tempo não seja percebido e o código de controle do CLP não atue por não ter verificado a falha de forma transitória. Até o momento, não foram detectados problemas neste sentido, e sim no registro dos eventos através do computador. Entretanto, a Unidade Autônoma prevê atuação em tempos ainda menores para poder auxiliar, também, em eventuais problemas deste tipo.

3.6.3 Código de controle do CLP no Sistema de Segurança

O código de controle foi gravado no CLP para atender as exigências da instalação do Irradiador Multipropósito e está escrito em linguagem Siemens S7 200. Este código não pode ser alterado ou atualizado sem o software STEP 7-Micro/WIN.

A principal função deste código é controlar os atuadores para abertura/fechamento de portas e exposição/recolhimento das fontes radioativas em função dos estados dos sensores que compõe a instalação do Irradiador Multipropósito.

O código foi desenvolvido de forma a não receber informações do computador, para evitar qualquer interferência deste com a segurança da instalação. O diagrama de blocos do código de controle, modificada de [33], pode ser observado nas Figuras 12 e 13.

FIGURA 12: Diagrama de controle da porta, alarme e sistema de exaustão

Na Figura 12 é mostrado o diagrama lógico da parte que libera ou não a abertura da porta de serviço, aciona alarmes e controla o sistema de exaustão para tentar, na eventualidade de um incêndio, minimizar os efeitos por abafamento. O diagrama da Figura 13 controla o movimento das fontes radioativas, seu tempo de trânsito e o controle do tempo de exposição das fontes. Paralelamente a esta lógica de funcionamento armazenada no CLP, existe uma lógica de reles que opera de forma conjunta para ampliar o sistema de redundância.

FIGURA 13: Diagrama de controle das fontes radioativas

Na Figura 14, modificada de [33], é mostrado o diagrama de blocos da lógica utilizada no procedimento de início de operações.

FIGURA 14: Diagrama da lógica do início das operações

A lógica utilizada no procedimento de início de operações é um sistema que exige, por parte do operador, que um procedimento pré-estabelecido de vistoria da instalação seja feito de modo que o sistema de segurança permita a exposição das fontes, removendo-as do fundo da piscina e iniciando o processo de irradiação.

Como todas as decisões realizadas pelo sistema de segurança, o procedimento de início de operações também é registrado no computador, através da interface homem máquina (IHM), tornando possível em uma eventual falha do sistema ou do procedimento, identificar a origem do problema. Todas as informações registradas podem ser visualizadas através da IHM.

3.6.4 Interface homem-máquina (IHM)

Interface homem-máquina (IHM) é um equipamento com um tipo de visor ou tela que serve para facilitar a comunicação entre as pessoas e as máquinas [37].

Para tornar o gerenciamento eletrônico do sistema de segurança um sistema amigável e permitir rápida visualização dos eventos em andamento por parte dos operadores da instalação, o CLP, através de sua saída serial, está conectado ao computador que executa um programa supervisorio, com nome comercial de Indusoft Web Studio®, que constitui a interface homem-máquina, no qual é possível verificar e acompanhar o funcionamento de todo o sistema bem como o estado atual de cada um dos sensores e demais elementos que compõem a planta do Irradiador Multipropósito. Este acompanhamento acontece através de quadros sinóticos na tela do computador com a plataforma Windows®.

A Unidade Autônoma, desenvolvida neste trabalho, atuará de forma redundante a este sistema, trabalhando de forma paralela às entradas e saídas do CLP, procurando registrar todos os eventos ocorridos, até aqueles que ocorrem em tempos muito curtos, que atualmente não são registrados.

Como exemplo, a Figura 15 representa o quadro sinótico referente as condições verificadas pelo programa supervisorio que permitem a partida do sistema (início das operações). Nessa tela, o diagrama de blocos indica que o irradiador está em condições

normais de operação, o alarme de procedimento de partida foi inicializado e ambas, fotocélula e corrente da porta de serviço, sinalizadas em vermelho, indicam ao operador que o processo ainda não está finalizado, em função de alguma obstrução ou presença do operador neste ponto [33]. Está é uma situação absolutamente normal, onde o procedimento de partida está sendo executado e o bloqueio da porta através da fotocélula e corrente da porta ainda não foi finalizado.

FIGURA 15: Tela do programa supervisorio, na condição de procedimento de partida

Ainda como exemplo dos quadros sinóticos, a Figura 16 representa a tela do programa supervisorio que permite visualizar o estado das botoeiras de emergência (em vermelho), e a porta de serviço ainda aberta, constituindo as últimas etapas do procedimento de partida.

FIGURA 16: Tela do programa supervisorio, registrando o estado das botoeiras de emergência na posição correta e porta de serviço aberta

Os quadros sinóticos ainda são complementados por uma tela de Listagem dos Alarmes, que é exibida na Figura 17, onde todos os eventos no irradiador são registrados com data, horário e descrição das ocorrências.

FIGURA 17: Tela do programa supervisorio apresentando o registro dos alarmes ou eventos ocorridos

Através de uma última tela, ainda é possível visualizar as condições gerais do Irradiador. É possível observar na Figura 18, através de cores, os estados das entradas e saídas de sensores e atuadores determinados pelo código de controle no CLP [33].

FIGURA 18: Tela do programa supervisorio com as condições gerais do Irradiador

3.7 Sistemas Microcontrolados

Sistemas microcontrolados, também denominados sistemas embarcados, são compostos por uma unidade de processamento com um *software* interno. Este *software*, embarcado na unidade de processamento é tecnicamente chamado de *firmware*. O *firmware* tem como função controlar o *hardware*, não passando de um conjunto de regras que rege o funcionamento do sistema, podendo ser escrito em diversas linguagens de programação e gravado em sua memória interna. Para desenvolvimento de um sistema microcontrolado, deve-se selecionar um microcontrolador comercial que atenda aos requisitos do projeto específico, tendo como principal característica o controle dedicado da aplicação. Os microcontroladores são componentes de baixo custo e que apresentam uma infinidade de periféricos internos, tornando seu campo de aplicações muito amplo[18, 38].

3.5.1 Microcontrolador

Microcontrolador é um pequeno computador em um único circuito integrado o qual contém um núcleo de processador, memória e periféricos programáveis de entrada e saída. São concebidos para aplicações dedicadas e embarcadas, em contraste com os microprocessadores utilizados em computadores pessoais ou outras aplicações de uso geral. Ao reduzir o tamanho e o custo são uma alternativa eficiente para controlar muitos processos e aplicações. São usados em produtos e dispositivos automatizados, como os sistemas de controle de automóveis, dispositivos médicos, controles remotos, máquinas de escritório, eletrodomésticos, ferramentas elétricas e outros sistemas embarcados. De forma diferente da programação para microprocessadores, que em geral contam com um sistema operacional, o programador ou projetista que desenvolve sistemas com microcontroladores geralmente cria todo programa que será executado pelo sistema [38, 39].

O mercado de microcontroladores conta com uma grande variedade de fabricantes tais como AMD[®], Atmel[®], Intel[®], Microchip[®], Texas[®], entre outras e conseqüentemente uma extensa lista de modelos. Frente a esta grande possibilidade de seleção, cabe ao desenvolvedor a escolha do componente adequado, levando em consideração sua

arquitetura, encapsulamento, velocidade de processamento, capacidade de armazenamento, quantidade de periféricos internos, consumo de energia e demais características relevantes para o desenvolvimento do projeto [18, 40].

4 MATERIAIS E MÉTODOS

Neste capítulo serão descritos os materiais utilizados, assim como os requisitos básicos da Unidade, de forma a justificar as escolhas dos componentes que constituíram a Unidade Autônoma de Monitoramento, Sinalização e Registro.

4.1 Materiais

Para execução desta dissertação, foi necessário a utilização de um *software* de esquematização eletrônica e *layout* de placa de circuito eletrônico, assim como um *softwares* para programação dos microcontroladores existentes no circuito. Todos os *softwares* usados no desenvolvimento construção da Unidade Autônoma são descritos a seguir.

1 – Proteus VSM (*Virtual System Modelling*)

É um *software* de desenvolvimento de esquemas eletrônicos que torna possível a validação do circuito criado através de simulações do projeto [41]. É possível criar e simular aplicações de diversos microcontroladores e periféricos.

Além da criação e simulação, o Proteus possibilita realizar a prototipagem da placa de circuito impresso, gerando visualização tridimensional e o aspecto físico do projeto. Para realizar todas essas tarefas o Proteus contempla duas interfaces: ISIS e ARES.

O ISIS é o ambiente de criação e simulação do esquema eletrônico. Através deste ambiente é possível inserir componentes animados ao circuito, permitindo assim uma simulação mais real e amigável.

De acordo com o fabricante, em relação a todos os concorrentes, o Proteus é o que oferece a maior biblioteca de microcontroladores, inclusive de várias empresas. Neste *software* de desenvolvimento, a inserção do microcontrolador acontece da mesma maneira que um outro componente qualquer, permitindo inclusive carregar o arquivo de código, no formato COF ou HEX. O arquivo com extensão no formato COF, permite sua execução em simuladores virtuais; já os arquivos com a extensão no formato HEX contém o código no padrão para serem gravados diretamente nas memórias dos microcontroladores. Desta

forma, antes mesmo da fabricação da Unidade Autônoma, seu comportamento já pode ser simulado e ao executar a simulação, é possível acompanhar os efeitos de seu código sobre o circuito a ser montado, a mudança de estados lógicos dos pinos e o acionamento dos demais periféricos que constituem o circuito eletrônico projetado, facilitando sobremaneira o desenvolvimento da unidade final e a redução de perdas por erros no projeto.

Este *software* permite ainda que se realize análises mais específicas, com a inserção de dois ou mais microcontroladores no mesmo esquema, tratamento de interrupções, temporizadores, comunicações USART, SPI, entre outras [42]. Para isso o ISIS oferece uma série de instrumentos virtuais, incluindo osciloscópio, analisador lógico, gerador de funções, voltímetro, amperímetro, terminal virtual, analisadores de protocolos, entre outros.

Com o ARES foi possível realizar a diagramação das placas de circuito impresso, determinar a largura das trilhas, o diâmetro das ilhas, o número de camadas da placa, inserir planos de terra, legendas, determinar furações, tamanho físico, tipos de encapsulamento, gerar arquivo *Gerber* [43, 44] e listas de componentes. O arquivo *Gerber* é um padrão universal que permite a industrialização do circuito eletrônico projetado através de máquinas de comando numérico computadorizado (CNC). Além disso, ainda é possível gerar uma visualização tridimensional de seu circuito antes da fabricação [45].

O programa Proteus VSM, desenvolvido pela empresa *Labcenter Electronics*, pode ser encontrado em uma versão de demonstração (completa) no site do fabricante [46].

2 – MPLAB IDE

É um *software* de programação para desenvolvimento de aplicações e sistemas embarcados, utilizando todos microcontroladores da família PIC. É fornecido gratuitamente pela empresa *Microchip Technology*® integrando diversos ambientes de trabalho para programação, simulação e gravação de microcontroladores. Pode ser encontrado no site do próprio fabricante [47].

3 – CCS C Compiler.

O código dos microcontroladores foi desenvolvido usando o ambiente MPLAB como descrito anteriormente, entretanto o MPLAB, nativamente, não permite a escrita de código em linguagem C, que é uma linguagem extremamente adaptada a sistemas embarcados. Para se desenvolver códigos em linguagem C na plataforma do MPLAB é necessário a integração com um compilador. Existem diversas opções de compiladores C para microcontroladores PIC tais como o C18, XC18, MikroC, Hitech e um dos mais

utilizados é o CCS C *Compiler* [48], produzido pela *Custom Computer Services*. O CCS C *Compiler* maximiza a reutilização do código permitindo a fácil portabilidade de um microcontrolador para outro. Oferece ainda funções internas específicas para os registradores, permitindo o acesso a recursos de *hardware* diretamente pelo código em linguagem C [49]. A *Custom Computer Services*, através de seu site, disponibiliza uma versão de demonstração, por um período de tempo definido [50].

Os componentes eletrônicos utilizados para o desenvolvimento da Unidade Autônoma foram:

- 40 optoacopladores (4N25);
- 1 Microcontrolador principal (PIC18F8722);
- 1 Microcontrolador secundário (PIC16F1503);
- 1 Gerador de data e hora de tempo real RTCC (DS1307);
- 1 Microcontrolador auxiliar (ATMEGA328 – placa Arduíno);
- 1 Módulo de interface com cartão de memória do tipo SD Card;
- 1 Display gráfico para interface entre o operador e a Unidade Autônoma;
- 1 Alarme piezoelétrico;
- 1 Relé com contato auxiliar.
- Componentes diversos tais como: placas de circuito impresso, conectores, resistores, capacitores, cristal oscilador de quartzo.

4.2 Diretrizes do Projeto da Unidade Autônoma

Todos os eventos ocorridos no Irradiador Multipropósito devem ser percebidos pela Unidade Autônoma e registrados localmente. Estes eventos são percebidos através da comutação de sinais elétricos vindos do Painel de Controle. Estes sinais, apesar de terem valores previsíveis, em um defeito elétrico, podem assumir valores superiores aos admissíveis pela Unidade Autônoma, podendo causar sua danificação. Também como condição, é desejado que a Unidade não venha a interferir nos sinais do Painel de Controle. Para atender estas duas situações foram utilizados isoladores ópticos, denominados optoacopladores, nos 40 sinais monitorados pela Unidade.

O componente da Unidade Autônoma desenvolvida, responsável pelo processamento dos sinais monitorados do Irradiador Multipropósito, é um microcontrolador, denominado principal. O microcontrolador principal permitiu o desenvolvimento de um sistema embarcado e dedicado, sem uso de sistemas operacionais.

Para aumentar a confiabilidade do sistema, um segundo microcontrolador, denominado secundário, foi aplicado ao projeto, com função exclusiva de monitorar o microcontrolador principal onde, na falha deste, um alarme interno será acionado alertando para o não funcionamento da Unidade Autônoma. O alarme interno foi incorporado com intuito de adicionar a função de alarme sonoro local caso uma falha seja detectada na Unidade Autônoma.

Para que os eventos possam ser registrados localmente, de forma permanente, e posteriormente analisados em qualquer computador, um módulo para integração com cartões de memória do tipo SD-Card (*Secure Digital Card*) foi implementado.

Devido a necessidade de se analisar os dados gravados posteriormente, foi necessária a inclusão de um gerador de data e hora em tempo real, que permitiu a inserção da data e hora atual no registro do evento ocorrido.

Um display com tela sensível ao toque foi inserido para permitir a simples interface do operador com a Unidade.

Um relé também foi implementado com a função de oferecer um contato auxiliar para comutação de alguma carga externa. Uma das possibilidades de uso é na instalação de um alarme sonoro externo de maior potência ou até mesmo no acionamento de um indicador luminoso.

A Figura 19 representa o diagrama de blocos da Unidade Autônoma desenvolvida. A direção das setas indica o sentido das trocas de informações entre os dispositivos. Observa-se que embora alguns periféricos permitam que sinais sejam transmitidos de forma bidirecional o microcontrolador principal apenas recebe (monitora) os sinais do Painel de Controle e Segurança.

FIGURA 19: Diagrama simplificado da Unidade Autônoma de Monitoramento, Sinalização e Registro desenvolvida para o Sistema de Segurança do Irradiador Multipropósito de Cobalto-60

5 RESULTADOS

Neste capítulo serão descritos como foram configurados os componentes utilizados para a aplicação proposta e os resultados obtidos com a Unidade Autônoma em funcionamento.

5.1 Placa dos optoacopladores

Atendendo as necessidades de isolamento e tratamento dos sinais, vindos do Painel de Controle para as entradas da Unidade Autônoma, foram utilizados optoacopladores que propiciam uma isolamento galvânica entre a entrada e saída dos sinais. Também conhecido como optoisolador, é simplesmente uma cápsula que contém um diodo emissor de infravermelho e um fotodetector [51].

O optoacoplador tem como característica fundamental a isolamento física entre a sua entrada e sua saída, impedindo assim que um sinal elétrico presente em sua entrada possa atingir sua saída em suas condições originais. Neste componente, um sinal elétrico detectado em sua entrada é convertido em luz infravermelha, que passa a ser reconhecido pelo elemento de saída fotodetector, comutando assim o seu estado e entregando um nível lógico de tensão em sua saída absolutamente adequado de ser monitorado pela Unidade Autônoma.

Para cada um dos sinais elétricos, vindos do Painel de Controle e recebidos pela Unidade Autônoma, foi inserido um opto acoplador independente. Para isso foram utilizados optoacopladores comerciais modelo 4N25 que apresentam uma isolamento entre a entrada e a saída de 2,5 kV, sem nenhum contato elétrico entre as partes [52]. Ainda como vantagem, este componente só permite o envio de sinais em apenas um sentido, que garante a individualidade dos sinais no Painel de Controle, de forma a não sofrerem qualquer interferência da Unidade Autônoma. A Figura 20 representa o esquema elétrico do opto- acoplador, onde pode ser observada de forma esquemática a isolamento física, neste caso também chamada de galvânica, entre a entrada e saída do sinal, demonstrando que a interação entre o sinal de entrada e a comutação da saída acontece de forma ótica através de luz infravermelha.

FIGURA 20: Diagrama elétrico de um optoacoplador com informações sobre seu funcionamento. Figura modificada de [53]

Na Unidade Autônoma foram previstas 40 entradas de sinal, para atender todas as situações possíveis de serem registradas, reunidas em uma placa de circuito eletrônico independente. Optou-se por construir a placa com os optoacopladores separados da placa de circuito eletrônico principal, unicamente para atender as dimensões de uma caixa escolhida para alojar a Unidade Autônoma. Na Figura 21 pode ser observada a placa dos optoacopladores finalizada.

FIGURA 21: Placa dos optoacopladores da Unidade Autônoma. Este conjunto garantiu a isolamento galvânica entre os equipamentos e a direcionalidade dos sinais do Painel de Controle para a Unidade Autônoma

5.2 Placa Controladora Principal

O componente principal, responsável por processar os sinais monitorados e armazenar o código do programa desenvolvido para gerenciamento é o microcontrolador, modelo PIC18F8722 [54], denominado principal. Este microcontrolador, bem como seu circuito de oscilação necessário para seu funcionamento, o circuito de reinicialização (*Reset*) e os componentes que auxiliam na gravação do programa através do modo *In Circuit Serial Programming* (ICSP) [55], foram montados no que foi chamado de placa controladora principal. Na Figura 22 pode ser observado parte dos circuitos necessários para funcionamento deste microcontrolador.

FIGURA 22: Esquema elétrico de parte da placa controladora principal, com o PIC18F8722 que é microcontrolador principal e circuitos de oscilação, *reset* e gravação ICSP

Na placa controladora principal, existe um circuito referente a tensão de alimentação de entrada, com destaque ao fusível de proteção e ao diodo emissor de luz para sinalização do estado da alimentação dos circuitos. Este circuito pode ser visto no esquema elétrico da Figura 23.

FIGURA 23: Trecho do esquema elétrico da placa controlador principal, exibindo a entrada para alimentação do circuito

A placa controladora principal dispõe de conexões e circuitos auxiliares para comunicação entre periféricos externos. Na Figura 24 (a), pode-se observar o esquema elétrico dos divisores resistivos nas linhas de dados que realizam a comunicação entre o microcontrolador principal, que trabalha em 5V, e a saída para o circuito auxiliar do cartão de memória, que trabalha em 3,3V. Na figura 24 (b), observa-se a conexão disponível para o circuito auxiliar gerador de data e hora (RTCC). Na Figura 24 (c) é demonstrado o esquema elétrico do trecho responsável por realizar a comunicação serial entre a tela de cristal líquido e a placa controladora principal.

Trecho do esquema elétrico da placa controlador principal, exibindo as conexões para o circuito do cartão de memória (a), o circuito gerador de data e hora (b) e a comunicação serial com o display (c)

Circuitos de sinalização, local e externo, também fizeram parte da placa controladora principal. Na Figura 25 (a) pode ser observado o esquema elétrico do trecho do circuito responsável por acionar o relé que controla um possível alarme externo. Na Figura 25 (b), pode-se observar o trecho do esquema elétrico responsável pelo acionamento do alarme sonoro local. Na Figura 25 (c) são mostrados dois diodos emissores de luz (LEDs) de sinalização utilizados durante a programação do circuito, responsáveis por indicar determinados estados do sistema que auxiliam o programador durante o desenvolvimento do código gravado na placa controladora principal.

FIGURA 25: Trecho do esquema elétrico da placa controladora principal, exibindo o acionamento do relé para alarme externo (a), o acionamento do alarme sonoro local (b) e os LEDs de sinalização (c)

A placa controladora principal recebe a conexão da placa de optoacoplamento através de dois conectores de 20 pinos cada, denominados J3 e J8. A placa de optoacoplamento é energizada apenas quando conectada a placa controladora principal, especificamente através do conector J9. Estes três conectores podem ser vistos através da figura 26 (a). A figura 26 (b) mostra o esquema elétrico de um conector extra, com o

objetivo de permitir a ampliação

de

funcionalidades da placa controladora caso necessário.

FIGURA 26: Esquema elétrico dos conectores disponíveis na placa controladora principal

A placa controladora principal contempla ainda um segundo microcontrolador, denominado secundário, responsável exclusivamente por monitorar o funcionamento do microcontrolador principal, aumentando assim a confiabilidade da Unidade Autônoma. A Figura 27 mostra o trecho do esquema elétrico que exhibe a conexão entre o microcontrolador principal e o microcontrolador secundário.

FIGURA 27: Trecho do esquema elétrico da placa controladora principal, exibindo a comunicação com o microcontrolador secundário

A placa controladora principal foi fabricada em dupla face e com legendas indicadoras de componentes e conectores, tornando rápida a identificação dos circuitos eletrônicos independentes que compõe a placa controladora principal da Unidade Autônoma.

A legenda facilita a correta conexão das tensões de alimentação e periféricos externos, com identificações de pinos, níveis e polaridade. Na Figura 28 pode ser observada a placa controladora principal finalizada já com todos seus componentes soldados.

FIGURA 28: Vista da placa controladora principal com todos os componentes que a compõe já soldados

5.2.1 Estrutura do código do programa no microcontrolador principal

O código do programa armazenado no microcontrolador principal contempla a identificação dos eventos ocorridos, otimizada de forma a permitir a identificação de qualquer mudança de estado nas condições do Painel de Controle, mesmo quando acontecem de forma transitória com período muito curto.

O código também contempla matrizes que indicam os estados lógicos de todas as condições de normalidade previstas. Estas matrizes foram construídas a partir dos diagramas de blocos do sistema de segurança como mostrados nas Figuras 12,13 e 14. Na sequência, este código executa a leitura dos estados das entradas do Painel de Controle e as armazena em uma nova tabela. Os dados desta nova tabela são comparados individualmente com dados contidos nas matrizes e em caso de não concordância, o código além de registrar as ocorrências no cartão de memória, sinaliza com um alarme local com a finalidade de alertar que o Painel de Controle não reagiu da forma esperada.

Com intuito de esclarecer esse procedimento, na Figura 29 são mostrados os procedimentos executados pelo código do programa desenvolvido, que foi armazenado no microcontrolador principal.

FIGURA 29: Fluxograma do código do programa armazenado no microcontrolador principal

5.2.2 Estrutura do código do programa no microcontrolador secundário

A Unidade Autônoma atua como um sistema de segurança redundante ao Painel de Controle. Para aumentar a confiabilidade da própria Unidade, foi incluído um segundo microcontrolador na placa controladora, denominado secundário, com intuito exclusivo de monitorar o microcontrolador principal a fim de se detectar possíveis paralizações que interromperiam o monitoramento executado pela Unidade Autônoma. Seu objetivo é

apenas monitorar um sinal periódico programado para ser enviado pelo microcontrolador principal.

O microcontrolador secundário foi escolhido de modo a atender os seguintes requisitos:

- a) Ser da mesma família do principal, permitindo que ambos fossem gravados através do mesmo gravador de PIC.
- b) Ter poucos pinos, neste caso, apenas 14 (em comparação com os 80 pinos do microcontrolador principal).
- c) Dispor de comunicação serial para acesso aos periféricos que compõe a Unidade Autônoma.

O microcontrolador PIC 16F1503 [56] atendeu a todas estas características.

A Unidade Autônoma tem por objetivo obter leituras de eventos no menor tempo possível, sendo assim, procurou-se em todas as etapas diminuir o tempo gasto do microcontrolador principal com outras atividades que não fossem o monitoramento das entradas do Painel de Controle. Desta forma, o microcontrolador secundário foi programado de maneira a não solicitar o envio de sinal do microcontrolador principal, para que este não gaste tempo na espera desta solicitação. Este último, periodicamente envia dados ao secundário para notificar sua condição. Sendo assim, o microcontrolador secundário aguarda um sinal elétrico que deve ser enviado pelo microcontrolador principal a cada minuto. A cada novo envio, este sinal alterna seu estado elétrico, entre 0 e 5 Volts. Essa alteração é percebida pelo microcontrolador secundário que interpreta que o microcontrolador principal está funcionando corretamente. Caso essa alternância não ocorra, uma nova leitura é feita no minuto seguinte, se mesmo assim o sinal elétrico continuar sendo o mesmo enviado na leitura anterior, o microcontrolador secundário interpreta que houve uma paralização do principal. Na Figura 30 pode ser observado o diagrama em blocos dos dispositivos que compõem este sistema.

No caso da existência desta ocorrência, um alarme sonoro local será disparado, alertando ao operador que a Unidade precisa ser reiniciada pois sofreu uma paralização. Este alarme sonoro local em nada se confunde com o alarme externo disparado em casos de falhas no Painel de Controle detectadas pela Unidade Autônoma.

FIGURA 30: Diagrama em blocos dos dispositivos que compõe o sistema secundário para detecção de travamento do microcontrolador principal

Com o objetivo de otimizar o processo de reinicialização, após a detecção de paralização, o microcontrolador secundário também foi programado para realizar tentativas sucessivas para reinicializar o microcontrolador principal. Um pulso é enviado pelo microcontrolador secundário, de forma automática, fazendo com que o sistema se recupere ou em casos mais específicos, através da intervenção do operador, ao pressionar o botão de *reset* disponível na Unidade Autônoma.

Como esta lógica está distribuída em dois códigos diferentes, um programado no microcontrolador principal e outro programado no microcontrolador secundário, as Figuras 31 e 32 representam os fluxogramas das lógicas do principal e secundário respectivamente, para entendimento do processo.

FIGURA 31: Fluxograma do microcontrolador principal, trecho da interação com o microcontrolador secundário

FIGURA 32: Fluxograma do código do programa armazenado no microcontrolador secundário

5.3 Gerador de data e hora RTCC

Após a ocorrência de uma falha, é necessário entender porque ela foi gerada. Sem um histórico de ocorrências anteriores à falha, não existem registros para análise, dificultando assim sua correta interpretação. A falta do histórico não necessariamente impede a correção da falha, embora a melhora contínua do processo de segurança, pode ajudar a encontrar o real problema através dos registros dos eventos e assim permitir a compreensão da falha, de seu comportamento, de sua geração, permitindo estudos analíticos para sua contenção.

Para se ter um histórico de ocorrências, se fez necessário um gerador de data e hora, denominado RTCC (*Real Time Clock and Calendar*). Estes RTCC's apresentam uma opção de energização independente, através de uma bateria específica, tornando-os eficientes em caso de uma possível desenergização do circuito principal, mantendo assim a data e hora programada em suas memórias.

O modelo de RTCC implementado na Unidade Autônoma é baseado no circuito integrado DS1307 [57]. Para funcionar, precisou ainda de um cristal de quartzo de 32.768 Hz. O circuito DS1307, o cristal e o conector para bateria individual são todos comercializados, já montados em um circuito eletrônico, que pode ser adicionado a qualquer projeto em que se necessite obter data e hora. Na Figura 33 podem ser observadas as duas faces do circuito gerador de relógio.

FIGURA 33: Gerador de relógio com o circuito integrado DS1307, visto de ambas as faces

Quando a Unidade Autônoma é inicializada o microcontrolador principal é responsável por capturar a data e hora inseridas pelo operador, através do display interativo, e enviá-las ao RTCC. Entretanto, os valores inseridos no formato numérico inteiro, precisam ser convertidos para o formato numérico BCD (*Binary Coded Decimal*) antes de serem enviados ao RTCC. Esta conversão precisou ser implementada no código inserido no microcontrolador principal. Logo após o recebimento dos valores devidamente convertidos, o RTCC os armazena em sua memória interna e passa a incrementá-los a cada segundo. Em uma eventual ocorrência, o microcontrolador principal envia um comando de solicitação de data e hora para o RTCC que responde imediatamente com um pacote de dados. Entretanto este pacote de dados não pode ser exibido ou registrado pois seu formato está no padrão BCD. Para que os dados possam ser interpretados em valores decimais, antes foi preciso realizar uma conversão de BCD para números inteiros. Esta conversão também precisou ser implementada no código inserido no microcontrolador principal.

5.4 Microcontrolador auxiliar e conexão com o cartão de memória

Para se obter a funcionalidade desejada da Unidade Autônoma foi necessário o armazenamento da data, hora e identificação do evento. Todas essas informações foram registradas de forma local na Unidade e como premissa deveriam permitir a análise em um computador com qualquer sistema operacional. Para atender este objetivo, os eventos foram gravados em um cartão de memória não volátil do tipo SD-Card [58] com capacidade de armazenamento de até 8 Gb, espaço suficiente para um grande número de eventos, e para que os dados pudessem ser analisados com facilidade em qualquer computador, as informações foram reunidas em um arquivo de texto, que pode ser acessado nos principais sistemas operacionais.

Nos primeiros testes que foram realizados, a própria placa controladora era responsável por enviar os dados diretamente para o cartão de memória. Os dados eram gravados em forma de *bytes* na memória do cartão, e não foram passíveis de interpretação de uma maneira prática para o usuário em um computador pessoal.

Como solução, foi incluído um microcontrolador auxiliar entre a placa controladora, que envia os dados a serem gravados, e o cartão de memória, que efetivamente armazena os dados em um arquivo de texto. Por facilidade, uma placa comercial que incorpora um microcontrolador, neste caso com o nome comercial de Arduíno Uno, foi incorporada a Unidade Autônoma.

A placa Arduíno [59] possui um conjunto de informações específicas que permite a criação de um arquivo de textos em um cartão de memória, em um formato adequado para ser lido em um computador pessoal. Previamente o cartão de memória deve estar formatado no padrão FAT32 [60], que é um processo corriqueiro de ser executado dentro de qualquer computador.

Ao realizar as gravações, observou-se que estas informações eram armazenadas na mesma linha, de forma sequencial, dificultando sensivelmente a análise dos dados pelo operador. Optou-se, por inserir uma instrução adicional no pacote de dados enviado pela placa controladora. Esta instrução indica o final do pacote e solicita que se execute uma simples formatação no arquivo, inserindo o comando de pular para a próxima linha. Desta maneira, quando o próximo pacote de dados é recebido, o mesmo é adicionado na linha seguinte. Esta formatação permitiu uma maior facilidade e é um formato mais utilizado para registro de informações no arquivo de eventos.

O procedimento de gravação seguiu os seguintes passos:

- 1) A placa controladora identifica o evento, faz a aquisição da data e hora atual e encapsula todas essas informações em um pacote de dados que é enviado a placa Arduino.
- 2) A placa Arduino, ao receber este pacote, cria um arquivo de textos, caso o mesmo já não tenha sido criado anteriormente, denominado DATALOG.txt. Neste arquivo são inseridas as informações enviadas pela placa controladora.
- 3) A cada novo evento identificado na placa controladora, um novo pacote de dados é enviado para a placa Arduino que o processa e o transfere para o arquivo no cartão de memória.

Na Figura 34 é observado o arquivo aberto em um computador pessoal com sistema operacional Windows®, e alguns registros de eventos, com suas devidas identificações, data e hora de suas respectivas ocorrências.

FIGURA 34: Arquivo em DATALOG.TXT gravado pela Unidade Autônoma em um cartão de memória tipo SD Card

5.5 Interface com o operador

A maneira escolhida de se exibir as informações processadas bem como permitir a interação entre a Unidade Autônoma e o operador foi através da implantação de um *display*, com recursos gráficos e de sensibilidade ao toque. Este *display* permitiu a exibição de textos e imagens monocromáticas além da navegação entre as telas com as funcionalidades disponíveis na Unidade Autônoma.

O modelo adquirido foi o TW12864 [61], que é constituído por 128 colunas por 64 linhas, cujo produto destes fatores determina o número de pixels deste dispositivo gráfico. A programação deste dispositivo permitiu a criação de uma tela de inicialização, Figura 35

(a), que exibe o logotipo do IPEN e o nome do projeto durante os 3 primeiros segundos. Logo após a inicialização do equipamento uma nova tela foi desenvolvida, com intuito de permitir o ajuste da data e hora atual, Figura 35 (b). Esta é uma operação feita na primeira vez em que o equipamento é instalado para atualização da data e hora e a mesma se mantém atualizada por meio do gerador de relógio RTCC incorporado na Unidade.

Sequencialmente foram criadas mais 3 telas com interfaces interativas, acessadas através de botões implementados graficamente nesta interface. A tela principal, Figura 35 (c), exibe 2 botões, um para executar um auto-teste da Unidade Autônoma e outro para iniciar o monitoramento do Painel de Controle. Na tela de auto-teste, Figura 35 (d), é possível observar os periféricos que estão sendo testados, começando pelo relê, passando pelo alarme sonoro e finalizando com o teste do microcontrolador redundante. Para finalizar, a última tela é acessada através do botão de monitoramento, onde é possível observar a data e hora em que o monitoramento se iniciou, além de um botão de retorno, que permite voltar a tela principal, Figura 35 (e).

FIGURA 35: Telas da interface com o operador da Unidade Autônoma

A cada interação entre a tela de monitoramento e a principal, atualiza-se o relógio. Optou-se por não atualizar o relógio em tempo real devido ao processamento gasto segundo a segundo. Desta forma, é possível manter o microcontrolador dedicando-se exclusivamente ao monitoramento das entradas do Painel de Controle, tornando o sistema mais eficiente em sua principal tarefa.

5.6 Montagem da Unidade Autônoma

Os periféricos como, placa dos optoacopladores, *display* para interface com o operador, gerador de data e hora, microcontrolador auxiliar e placa de conexão com o cartão de memória, são todos conectados a placa controladora principal de modo a constituir a Unidade Autônoma.

A Figura 36 mostra todo o conjunto de componentes e os periféricos da Unidade Autônoma.

FIGURA 36: Conjunto das placas e periféricos que constituem a Unidade Autônoma

5.7 Simulador de eventos em tempos curtos

Para que a eficiência da Unidade Autônoma pudesse ser comprovada foi necessária a sua comparação com os tempos de resposta a uma falha do Painel de Controle e Segurança do Irradiador. Entretanto, previamente, foi imprescindível o desenvolvimento de um simulador de eventos em tempos curtos. Durante a fase de projeto, foram definidas as premissas do simulador de eventos, sendo:

- 1-) Sistema compacto e flexível para facilitar sua inserção ao Painel de Controle e Segurança do Irradiador Multipropósito;
- 2-) Capacidade de ajuste de tempos de duração da falha, para poder verificar a confiabilidade de registros da Unidade Autônoma em tempos muito curtos;
- 3-) Configuração e operação através de *software* eliminando contatos eletromecânicos;
- 4-) Isolação entre o simulador de eventos e o Painel de Controle e Segurança do Irradiador Multipropósito.

Para atender a todos os requisitos citados, a solução encontrada foi desenvolver o simulador de eventos sob a plataforma da placa Arduíno Uno, que foi programada para gerar pulsos em tempos controlados, que simularam as falhas. Esses pulsos tiveram seus tempos de duração configurados e disparados através de um computador a qual o simulador de eventos está constantemente conectado. A comunicação entre o computador e o simulador de eventos foi feita de forma serial através do protocolo USB.

Para que o simulador de eventos estivesse completamente isolado do Painel de Controle e Segurança, foi também utilizado um optoacoplador na saída do simulador.

É importante destacar que, a denominação de evento, está sendo adotada neste trabalho para toda e qualquer ocorrência de mudança de estado lógico de um sinal elétrico, podendo esta ser uma alteração esperada, que faz parte da operação do Irradiador Multipropósito ou oriunda de uma falha de algum componente do sistema. Desta forma, entende-se evento como uma comutação que deve ser percebida e registrada, seja advinda de um procedimento aguardado ou de uma falha inesperada. A diferença entre ambos é que uma comutação de um procedimento padrão do sistema ocorre em um tempo relativamente longo, fácil de ser monitorado e registrado. Enquanto que uma falha, quando intermitente, pode apresentar tempos de comutação extremamente curtos, difíceis de serem percebidos e consequentemente registrados.

A Figura 37 demonstra, no formato de diagrama em blocos, o arranjo desenvolvido para a simulação de eventos utilizado nos testes da Unidade Autônoma.

FIGURA 37: Diagrama em blocos, o arranjo desenvolvido para a simulação de eventos utilizado nos testes da Unidade Autônoma

Com o arranjo do simulador de eventos montado, foi possível ajustar o tempo de duração do pulso inserido para ser detectado pelo Painel de Controle e pela Unidade Autônoma. Este ajuste do tempo de duração assim como o disparo do pulso, são todos realizados através do computador.

Para validação do procedimento de simulação de falhas ou eventos, utilizou-se um osciloscópio, para certificar-se que a duração do pulso configurada no simulador coincidia com o valor medido. Após o disparo do simulador de eventos, o osciloscópio exibia a forma de onda do sinal bem como o seu período coincidente com o ajustado no computador, caracterizando um sistema calibrado pronto para os testes.

5.8 Teste de detecção de eventos pelo CLP do Painel de Controle

No primeiro conjunto de testes, através do computador foram selecionados tempos de duração dos pulsos para simulação de falhas na ordem de μs . Neste momento a Unidade Autônoma não foi conectada ao Painel de Controle do Irradiador Multipropósito. O objetivo foi observar apenas a detecção do evento pelo CLP do Painel de Controle, sem levar em consideração o registro deste evento no computador ou na Unidade Autônoma.

Os testes foram realizados através de 1000 disparos do simulador, 100 em cada faixa de tempo selecionada. O Quadro 1 apresenta o percentual de detecção do CLP em função do tempo de duração do evento.

QUADRO 1: Porcentagem de detecção de eventos no CLP, do Painel de Controle, em função da duração do tempo do evento aplicado

Tempo de duração do evento [μs]	Percentual de detecção [%]
10	14
20	66
30	80
40	85
50	91
60	100
70	100
80	100
90	100
100	100

Como pode ser observado pelos dados apresentados no Quadro 1, o CLP do Painel de Controle apresenta 100% de confiabilidade para eventos com tempo de duração iguais ou maiores que 60 μs , indicando uma boa capacidade de detecção em tempos curtos. Entretanto, é importante frisar que estes eventos não foram registrados.

5.9 Teste de detecção e registro de eventos pelo computador com o *software* supervisor

Este teste foi realizado com o intuito de se verificar qual seria o tempo mínimo de um evento, para que o mesmo pudesse ser detectado pelo CLP do Painel de Controle e registrado no computador.

Após a detecção do evento, o CLP envia um sinal identificado para o computador com a plataforma *Windows*[®] através de comunicação serial, o que exige conversões de níveis de tensão e consequentemente atrasos [62]. Posteriormente, o seu registro ocorre através do *software Indusoft*[®], instalado no computador, o que demanda um certo tempo determinado pela própria plataforma *Windows*[®]. Embora estes tempos, de forma individual, tenham durações pequenas, quando somados acabam apresentando durações demasiadamente longas, que são propagados por meio da detecção do evento, conversão de nível de tensão da identificação do evento, envio da identificação do evento para o computador, identificação do recebimento do evento pela plataforma *Windows*[®], aquisição da data e hora de recebimento e registro deste evento no *software Indusoft*[®]. Todo este tempo de procedimento foi mensurado neste teste.

Neste momento a Unidade Autônoma ainda não foi conectada ao sistema, e o objetivo deste teste foi observar a detecção do evento pelo CLP do Painel de Controle e o seu registro no computador.

Seguindo o mesmo procedimento do teste anterior, através do computador, foram ajustados tempos de disparo para o simulador na ordem de 100 a 900 μ s, entretanto, nenhum evento foi registrado para esta faixa de tempo. Os tempos passaram para ordem de 1 a 100 ms e mesmo assim, ainda não puderam ser observados nos registros do computador. Um novo ajuste foi realizado e os tempos passaram para ordem de 0,5 até 2 s, permitindo desta vez, que fossem registrados com certa reincidência.

Os testes foram realizados através de 1600 disparos do simulador de eventos, 100 em cada uma das 16 faixas de tempo determinadas.

O Quadro 2 apresenta o percentual de registros (e evidentes detecções) em função do tempo de duração do evento.

QUADRO 2: Porcentagem de registros de eventos no computador em função da duração do tempo do evento aplicado

Tempo de duração do evento [s]	Percentual de registros [%]
0,5	6
0,6	5
0,7	13
0,8	20
0,9	26
1	54
1,1	85
1,2	84
1,3	84
1,4	84
1,5	84
1,6	90
1,7	90
1,8	94
1,9	98
2	98

De acordo com os dados apresentados no Quadro 2, a quantidade de falhas nos registros é muito grande, o que compromete a confiabilidade do sistema de registros para eventos ocorridos com duração menor ou igual a 1 s.

Embora o percentual de registros aumente consideravelmente para eventos com duração superior a 1 segundo, cerca de uma dezena de falhas intermitentes já ocorreram no sistema do Irradiador Multipropósito, com tempos inferiores a este, e não puderam ser registrados.

Deve ser observado que em sistemas mais modernos, os CLPs disponibilizam comunicação via *Ethernet* para o computador que tem o programa supervisor, apresentando alta confiabilidade e um índice de falhas menor que o de uma comunicação serial do tipo descrito neste capítulo, mas como discutido na introdução deste trabalho

demandam a substituição total do CLP do Pannel de Controle, assim como sua nova programação e homologação da instalação.

5.10 Teste de detecção e registro de eventos pela Unidade Autônoma

Este teste foi realizado com o intuito de se verificar qual seria o tempo mínimo de um evento, para que o mesmo pudesse ser detectado e registrado na Unidade Autônoma. Neste teste a Unidade Autônoma foi conectada de forma paralela ao Pannel de Controle, de forma a não provocar qualquer interferência ao sistema do Irradiador Multipropósito.

Todos os procedimentos relacionados ao arranjo do simulador de eventos mantiveram-se constantes, como nos testes anteriores. Através do computador, foram ajustados tempos de disparo para o simulador na ordem de 1 μ s a 2 s. Os testes foram realizados através de 700 disparos do simulador de eventos, 100 em cada uma das 7 faixas de tempo determinadas.

O Quadro 3 apresenta o percentual de registros em função do tempo de duração do evento.

QUADRO 3: Porcentagem de registros de eventos no computador em função da duração do tempo do evento aplicado

Tempo de duração do evento [μs]	Percentual de registros [%]
1	90
5	92
10	99
40	99
50	100
100	100
1s	100
2s	100

De acordo com os dados apresentados pelo Quadro 3, a detecção e registro da Unidade Autônoma supera inclusive o tempo de resposta do CLP.

6 CONCLUSÕES

A Unidade Autônoma de Monitoramento, Sinalização e Registro para o Sistema de Segurança do Irradiador Multipropósito de Cobalto 60, desenvolvida e apresentada nesta dissertação, demonstrou-se eficiente para as aplicações citadas, todas verificadas através de testes experimentais. Para efeito comparativo de análise, os dados obtidos na detecção de eventos, tanto pelo CLP, quanto pelo computador e a Unidade Autônoma foram inseridos no Quadro 4:

QUADRO 4: Comparativo de porcentagem de registros de eventos nos sistemas isolados em função da duração do tempo do evento aplicado

Tempo de duração do evento [μ s]	Percentual de registros [%]		
	Painel de Controle com CLP	Computador com Supervisório	Unidade Autônoma
1	0	0	90
10	14	0	99
20	66	0	99
40	85	0	99
50	91	0	100
60	100	0	100
100	100	0	100
200000	100	2	100
500000	100	6	100
800000	100	20	100
1000000	100	54	100
1500000	100	84	100
2000000	100	98	100

Analisando e comparando os dados do Quadro 4, observa-se que a detecção de eventos pelo CLP do Painel de Controle é confiável e com alto percentual de detecção para eventos a partir de 50 μ s.

Conclui-se que o CLP apresenta alta confiabilidade na detecção dos eventos, mesmo em tempos curtos, o que garante a segurança do Irradiador Multipropósito. Entretanto, embora o CLP detecte o evento curto e tome a ação correta, garantindo a

integridade da instalação e dos operadores, o evento não é registrado e a possível falha não será identificada.

Observa-se que a detecção de eventos pelo computador com software supervisorio não é satisfatória. Pode-se concluir que, o tempo gasto por todo o processo desde a detecção, passando pela transmissão e finalizando no registro do evento, impactavam diretamente na eficiência do sistema de registros dos eventos no Irradiador Multipropósito. Mesmo com o CLP do Painel de Controle apresentando boa resposta na detecção e na tomada de decisões, existia uma deficiência no sistema de registros que não permitia o armazenamento de todos os eventos de curta duração. Eventos com duração de 1 segundo, por exemplo, são detectados em 54% das vezes. Já eventos com duração de 500 ms, são detectados em apenas 6% das vezes. Mesmo para eventos longos, na ordem de 2 segundos de duração, o computador não apresenta 100% de eficiência na detecção e registro de tais eventos.

Com o evento não registrado, caso uma falha ocorresse, a mesma não poderia ser analisada, pois não existiriam registros que a precederam, dificultando o entendimento da causa real.

De acordo com os resultados obtidos nos testes de detecção e registro de eventos através da Unidade Autônoma, observa-se que seu desempenho é satisfatório, superando até mesmo o CLP do Painel de Controle (que apenas monitora e não registra). Para eventos curtos, com duração de apenas 1 μ s, o percentual de detecção da Unidade Autônoma já é de 90%. Eventos à partir de 10 μ s já são detectáveis em 99% das ocorrências.

Em comparação com o antigo sistema de registros de eventos, a Unidade Autônoma apresentou um melhor desempenho, garantindo uma alta confiabilidade dos registros, detectando e registrando eventos com durações menores em até 200 mil vezes.

A Unidade Autônoma de Monitoramento, Sinalização e Registro para o Sistema de Segurança do Irradiador Multipropósito de Cobalto 60, possibilitou que fossem registrados eventos com durações na ordem de 20^4 menores que os atuais e com as seguintes características:

Total independência e sentido dos sinais entre o Painel de Controle e Unidade Autônoma, evitando qualquer interferência no Painel pela introdução do equipamento desenvolvido;

Microcontrolador principal que identifica eventos e verifica se a decisão do Painel de Controle foi correta;

Microcontrolador secundário que periodicamente confirma o funcionamento do microcontrolador principal, aumentando a confiabilidade do próprio equipamento desenvolvido;

Código de programação otimizado para permitir a visualização de eventos de curta duração;

Interface com o operador de fácil uso e intuitiva;

Relógio interno com bateria de *back-up* para evitar perda de data e hora;

Gravação em cartão de memória tipo SD CARD no formato “.txt” compatível com a maioria dos computadores.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] SANTOS, P. S. Estudo e otimização dos parâmetros de processamento por radiação gama em escala industrial considerando fatores operacionais. São Paulo: 2017. Dissertação (Mestrado em Ciências na Área de Tecnologia Nuclear- Aplicações). IPEN/USP. Universidade de São Paulo.
- [2] INTERNATIONAL ATOMIC ENERGY AGENCY. Guidelines for Development, Validation and Routine Control of Industrial Irradiation Processes. Vienna. 2013. (Nº 4).
- [3] CHMIELEWSKI, A. G.; HAJI-SAEID, M. Radiation Technologies: past, present and future. Radiation Physical and Chemistry, 71, 2004. 17-21.
- [4] CHMIELEWSKI, A. G. Worldwide developments in the field of radiation processing of materials in the dawn of 21st century. NUKLEONIKA, 51, 2006. S3-S9.
- [5] KODAMA, Y.; RODRIGUES, O. JR.; GARCIA, R. H. L.; SANTOS, P. S.; VASQUEZ, P. A. S.. Study of Free Radicals in Gamma Irradiated Cellulose of Cultural Heritage Materials Using Electron Paramagnetic Resonance. Radiation Physics and Chemistry, 124, 2016. 169-173.
- [6] RELA, P.R. Utilização da Radiação Ionizante na esterilização de produtos médicos e farmacêuticos, Revista da Sociedade Brasileira de Controle da Contaminação(SBCC), edição 4, 10-16, 2001.
- [7] Norma CNEN NN 6.02, Resolução CNEN 166/14 Abril / 2014, LICENCIAMENTO DE INSTALAÇÕES RADIATIVAS, Resolução CNEN 112/11, Publicação: DOU 01.09.2011, Resolução CNEN 166/14, Publicação: DOU 29.04.2014.
- [8] BSS-115 International Basic Safety Standards for Protection against Ionizing Radiation and for Safety of Radiation Sources-IAEA, Vienna, 1966 –Safety Series Nº 115.
- [9] INTERNACIONAL ATOMIC ENERGY AGENCY. Radiation Safety of Gamma, Electron and X Ray Irradiation Facilities. Specific Safety Guide Nº SSG8. Austria, 2010.
- [10] Norma CNEN NN 3.01, Resolução CNEN 27/04 Janeiro/ 2005, DIRETRIZES BÁSICAS DE PROTEÇÃO RADIOLÓGICA, Resolução CNEN 164/14 (Alteração do item 5.4.3.1), Publicação: DOU 11.03.2014.

- [11] SIEMENS. Simatic S7-200 Programmable Controller, Alemanha, 1999.
- [12] SILVEIRA, J. L. Comunicação de Dados e Sistemas de Teleprocessamento. São Paulo: Makron Books, 1991.
- [13] INDUSOFT. Indusoft Web Ready Documentation and Literature, Estados Unidos, 2013.
- [14] SIEMENS. Catálogo da linha Safety. Disponível em: <https://w3.siemens.com.br/automation/br/pt/seguranca-de-maquinas/interfaces-de-seguranca/3rk3/Documents/Catálogo%20Safety%20Integrated.pdf>. Acessado em 04/07/16.
- [15] SIEMENS. Catálogo do Controlador S7-1200, Disponível em: <http://www.automation.siemens.com/salesmaterial-as/catalog/en/simatic-st70-chap03-english-2015.pdf>. Acessado em: 07/07/16.
- [16] SIEMENS. Fail-safe Digital Input Module – F-DI, 2016. Disponível em: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7136-6BA00-0CA0>. Acessado em: 10/08/2016.
- [17] SIEMENS. Fail-safe Digital Output Module – F-DI, 2016. Disponível em: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7136-6DB00-0CA0>. Acessado em: 10/08/2016.
- [18] WILMSHURST, T. Designing embedded systems with PIC microcontrollers – Principles and applications. 2ed. EUA: Butterworth-Heineman, 2009.
- [19] FRANÇA, M. B. A.; SILVA, C. F. Tecnologia Industrial e Radiações Ionizantes e Não Ionizantes. Goiânia: AB Editora, 2007.
- [20] SILVA. Abel A. Radiação Ionizante e Não Ionizante. Pontifícia Universidade Católica de Minas Gerais: Dep. de Física e Química. Belo Horizonte, 1997
- [21] OKUNO, Emico; YOSHIMURA, Elisabeth. Física das Radiações. São Paulo: Oficina de Textos, 2010.
- [22] MARTINS, Roberto de Andrade. Investigando o invisível: as pesquisas sobre Raios X logo após a sua descoberta por Rontgen. Revista da SBHC. São Paulo, 1997.
- [23] DEYLLLOT, M. E. C. Física das Radiações: Fundamentos e Construção de Imagens. São Paulo: Érica, 2014.

- [24] ISOLDI, Ana Beatriz Godoy. Estudo do efeito da dose de radioesterilização sobre as propriedades do poli (tereftalato de Etileno) - pet reciclado. São Paulo:, 2003. Dissertação (Mestrado em Ciências na Área de Tecnologia Nuclear- Aplicações). IPEN/USP. Universidade de São Paulo.
- [25] NUNES, Maíra Goes. Avaliação do desempenho dos detectores termoluminescentes de caso: DY e LIF: MG, Ti na dosimetria de feixes clínicos de elétrons. São Paulo: 2003. Dissertação (Mestrado em Ciências na Área de Tecnologia Nuclear- Aplicações). IPEN/USP. Universidade de São Paulo.
- [26] KNOLL, G. F..Radiation Detection Measurement. 3ed. New York: John Wiley & Sons, 1999.
- [27] FIOCRUZ. Radiações. Disponível em: http://www.fiocruz.br/biosseguranca/Bis/lab_virtual/radiacao.html. Acessado em 05/07/16.
- [28] YOSHIMURA, E. M.. Física das Radiações: Interação da radiação com a matéria. Revista Brasileira de Física Médica, v.3, p.57-67, fev. 2009.
- [29] VELUDO, Patrícia Carvalho. Efeitos da Radiação X e Níveis de Exposição em Exames Imagiológicos Inquéritos a Clínicos Gerais. Coimbra: 2011. Dissertação (Mestrado em Saúde Pública). Faculdade de Medicina da Universidade de Coimbra.
- [30] Norma CNEN – NN – 3.01 de 13/03/2014. “Diretrizes Básicas de Proteção Radiológica”.
- [31] AZEVEDO, Ana Cecília Pedrosa. Radioproteção em Serviços de Saúde. Rio de Janeiro:2001. Artigo FIOCRUZ. Escola Nacional de Saúde Pública-CESTEH e Programa de Radioproteção e Dosimetria.
- [32] KOTTWITZ, S. TeXample.net. Disponível em: <http://www.texample.net/tikz/examples/gamma-interaction/>. Acessado em: 23/07/2016.
- [33] CALVO, W.A.P. Desenvolvimento do Sistema de Irradiação em um Irradiador Multipropósito de Cobalto-60 tipo Compacto. São Paulo: 2005. Tese (Doutorado em Ciências na Área de Tecnologia Nuclear- Aplicações). IPEN/USP. Universidade de São Paulo.
- [34] SOMESSARI, Samir Luiz. Automação do processo de soldagem a laser (Nd: YAG) para confecção das sementes de iodo-125 utilizadas em braquiterapia. São Paulo: 2010. Dissertação (Mestrado em Ciências na Área de Tecnologia Nuclear- Aplicações). IPEN/USP. Universidade de São Paulo.

- [35] CORETTI, J. A. Manual de Treinamento Básico de Controlador Programável. Ser-tãozinho, Centro de Treinamento SMAR. 1998.
- [36] PUPO, Maurício Santos. Interface homem-máquina para supervisão de um CLP em controle de processos através da www. São Carlos: 2002. Dissertação (Mestrado em Engenharia Elétrica). Escola de Engenharia de São Carlos. Universidade de São Paulo.
- [37] SHARP, H.; REGERS, Y.; PREECE, J. Design de Interação - Além da Interação Homem-computador. Porto Alegre: Bookman, 2013.
- [38] ANDRADE, F. S.; OLIVEIRA A. S. Sistemas embarcados - hardware e firmware na prática. São Paulo: Érica, 2006.
- [39] BAILEY, O. Embedded Systems: Desktop Integration. Wordware, 2005.
- [40] ARNOLD, Ken. Embedded Controller Hardware Design. LLH Technology, 2000.
- [41] LABCENTER. Proteus VSM, 2015. Disponível em: <http://www.labcenter.com/>. Acessado em: 01/08/2016.
- [42] MAZIDI, M. A.; MAZIDI J. G. The 8051 Microcontroller and Embedded Systems. São Paulo: Pearson Prentice Hall, 1999.
- [43] MICROPRESS. Arquivo Gerber. Disponível em: <http://www.micropress.com.br/arquivos-gerber/>. Acessado em: 22/04/2017.
- [44] GOUVEA, M. Saiba qual é a importância dos arquivos de fabricação da PCI: formato Gerber. Disponível em: <http://produza.ind.br/tecnologia/formato-gerber/>. Acessado em: 22/04/2017.
- [45] MODOZERO. Proteus: Apresentação, 2015. Disponível em: <http://www.modozero.com/#!Proteus-Aula-0-Apresentação/c1kt3/55e45b440cf28ffc7eeced25>, Acessado em: 16/07/2016.
- [46] LABCENTER. Proteus VSM: Demo, 2015. Disponível em: http://www.labcenter.com/download/prodemo_download.cfm. Acessado em: 01/08/2016.
- [47] MICROCHIP. MPLAB-IDE, 2015. Disponível em: <http://www.microchip.com/pagehandler/en-us/devtools/dev-tools-parts.html>, Acessado em: 01/08/2015.
- [48] CUSTOM COMPUTER SERVICES, INC. CCS C Compiler Microchip PIC® MCU Development Tool Solutions. Disponível em: <http://www.ccsinfo.com/>. Acessado em: 01/08/2015.

- [49] MODOZERO. Programação: Como iniciar? Disponível em: <http://www.modozero.com/#!/Programação-Como-iniciar/c1kt3/55f449db0cf24e84f7681e72>, Acessado em: 16/07/2016.
- [50] CUSTOM COMPUTER SERVICES, INC. CCS C Compiler: Demo Download. Disponível em: <http://www.ccsinfo.com/downloads.php>. Acessado em: 01/08/2015.
- [51] BOYLESTAD, R. L.; NASHELSKY, L. Dispositivos eletrônicos e teoria de circuitos, 8ª Ed. São Paulo: Pearson Prentice Hall, 2004.
- [52] FAIRCHILD SEMICONDUCTOR General Purpose Optocouplers: Data Sheet, 2002. Disponível em: <http://pdf.datasheetcatalog.com/datasheet/fairchild/4N25.pdf>. Acessado em: 11/07/2015.
- [53] ASSIS, M. Controle PID de Potência em Corrente Alternada - Arduino e TRIAC - Parte II. Disponível em: http://automatobr.blogspot.com.br/2013/05/controle-de-potencia-em-corrente_17.html. Acessado em: 21/05/2017.
- [54] MICROCHIP. PIC18F8722: Family Data Sheet, 2008. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/39646c.pdf>. Acessado em: 11/07/2015.
- [55] PEREIRA, F. Microcontroladores PIC: programação em C. São Paulo: Érica, 2006.
- [56] MICROCHIP. PIC16F1503: Family Data Sheet, 2011. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/41607A.pdf>. Acessado em: 11/07/2015.
- [57] MAXIM INTEGRATED. DS1307: Serial I2C Real-Time Clock, 2015. Disponível em: <http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>. Acessado em: 04/07/2016.
- [58] ITEAD STUDIO. SD-Shield: Overview, 2014. Disponível em: <http://www.robotshop.com/media/files/pdf/datasheet-im140726001.pdf>. Acessado em: 15/07/2016.
- [59] ARDUÍNO. Arduino UNO Board, 2016. Disponível em: <https://www.arduino.cc/en/main/arduinoBoardUno>. Acessado em: 15/11/2016.
- [60] MICROSOFT CORPORATION. Hardware White Paper: FAT – General Overview of on-disk format, 2000. Disponível em: <https://staff.washington.edu/dittrich/misc/fatgen103.pdf>. Acessado em: 28/12/2016.

- [61] TATO. TW-12864 Display Gráfico Serial: Manual, 2011. Disponível em: <http://tato.ind.br/files/TW-12864.pdf>. Acessado em: 23/11/2016.
- [62] TANENBAUM, A. S. Sistemas Operacionais Modernos. 4ª Ed. São Paulo: Pearson Prentice Hall, 2016.

APÊNDICE A – Código do microcontrolador auxiliar

```

#include <SPI.h>
#include <SD.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX
const int chipSelect = 4;
char dadoRec[15];
int i=0;
File dataFile;

void setup(){
  Serial.begin(9600);
  if (!SD.begin(chipSelect)) {
    Serial.println("Leitor de cartao falhou ou SD Card nao esta inserido...");
    return;
  }
  Serial.println("LEITOR DE CARTAO PRONTO...");
}

void loop(){
  if(Serial.available()){
    dadoRec[i] = (char)Serial.read();
    Serial.print(dadoRec[i]);
    if(dadoRec[0]=='i'){
      i++;
    }
    else{
      i=0;
    }
  }
  if(i>=13){
    i=0;
    dataFile = SD.open("Datalog.txt", FILE_WRITE);
    if (dataFile){
      dataFile.print("Data: ");
      dataFile.print(dadoRec[1]); //dia
      dataFile.print(dadoRec[2]);
      dataFile.print("/");
      dataFile.print(dadoRec[3]); //mês
    }
  }
}

```



```

dataFile.print(dadoRec[4]);
dataFile.print("/");
dataFile.print(dadoRec[5]); //ano
dataFile.print(dadoRec[6]);
dataFile.print(" - Hora: ");
dataFile.print(dadoRec[7]); //hora
dataFile.print(dadoRec[8]);
dataFile.print(":");
dataFile.print(dadoRec[9]); //minuto
dataFile.print(dadoRec[10]);
dataFile.print(" - Evento: ");
trataDado(0);
dataFile.print(" - Estado: ");
dataFile.print(dadoRec[12]); //estado lógico
dataFile.println(" ");
dataFile.close();
Serial.println("");
Serial.print("Data: ");
Serial.print(dadoRec[1]); //dia
Serial.print(dadoRec[2]);
Serial.print("/");
Serial.print(dadoRec[3]); //mês
Serial.print(dadoRec[4]);
Serial.print("/");
Serial.print(dadoRec[5]); //ano
Serial.print(dadoRec[6]);
Serial.print(" - Hora: ");
Serial.print(dadoRec[7]); //hora
Serial.print(dadoRec[8]);
Serial.print(":");
Serial.print(dadoRec[9]); //minuto
Serial.print(dadoRec[10]);
Serial.print(" - Evento: ");
trataDado(1);
Serial.print(" - Estado: ");
Serial.print(dadoRec[12]); //estado
Serial.println(" ");
Serial.println(" ");
}
else{
    Serial.println("Erro ao tentar abrir/criar o arquivo Datalog.txt");
}
}
}
}

```

```
void trataDado(int modo){
    char EventoRecebido = dadoRec[11];
    if(EventoRecebido=='A'){
        if(modo==0){
            dataFile.print("Fotocelula 1 acionada");
        }else{
            Serial.print("Fotocelula 1 acionada ");
        }
    }
    else if(EventoRecebido=='B'){
        if(modo==0){
            dataFile.print("Fonte exposta");
        }else{
            Serial.print("Fonte exposta ");
        }
    }
    else if(EventoRecebido=='C'){
        if(modo==0){
            dataFile.print("Fonte recolhida");
        }else{
            Serial.print("Fonte recolhida ");
        }
    }
    else if(EventoRecebido=='D'){
        if(modo==0){
            dataFile.print("Porta principal aberta");
        }else{
            Serial.print("Porta principal aberta");
        }
    }
    else if(EventoRecebido=='E'){
        if(modo==0){
            dataFile.print("Porta principal aberta fechada");
        }else{
            Serial.print("Porta principal aberta fechada ");
        }
    }
    } //... demais casos
}
```

APÊNDICE B – Código do microcontrolador principal

```

/*****
                                CONFIGURAÇÕES INICIAIS
*****/
#include<18F8722.h>
#include<MatrizesDeEstados.h>
#fuses HS, NOWDT, NOLVP
#use delay (clock=20000000)
#use RS232(baud=9600, xmit=pin_j4, rcv=pin_j5)
#use RS232 (baud=9600, xmit=pin_c6, rcv=pin_c7, STREAM=arduino)
#use I2C(master, sda=pin_a1, scl=pin_a0)
#byte intcon = 0xff2
#bit T0IF = intcon.2
#byte T0CON = 0xfd5
#bit T0ON = T0CON.7
#bit T8bits = T0CON.6
/*****

                                DEFINES
*****/

#define SEGUNDOS           0x00
#define MINUTOS            0x01
#define HORAS              0x02
#define DIA_DA_SEMANA      0x03
#define DIA_DO_MES         0x04
#define MES                0x05
#define ANO                0x06
#define Ende_escrita       0xd0
#define Ende_leitura       0xd1
/*****

                                DEFINES DAS 40 ENTRADAS DIGITAIS OPTOACOPLADAS
*****/

#define en1    pin_b5
#define en2    pin_b4
#define en3    pin_b3
#define en4    pin_b2
#define en5    pin_b1
#define en6    pin_b0
#define en7    pin_j3
#define en8    pin_j2
#define en9    pin_j1
#define en10   pin_j0
#define en11   pin_d7

```

```

#define en12 pin_d6
#define en13 pin_d5
#define en14 pin_d4
#define en15 pin_d3
#define en16 pin_d2
#define en17 pin_d1
#define en18 pin_d0
#define en19 pin_e7
#define en20 pin_e6
#define en21 pin_e5
#define en22 pin_e4
#define en23 pin_e3
#define en24 pin_e2
#define en25 pin_h0
#define en26 pin_h1
#define en27 pin_h2
#define en28 pin_h3
#define en29 pin_e1
#define en30 pin_e0
#define en31 pin_g0
#define en32 pin_g1
#define en33 pin_g2
#define en34 pin_g3
#define en35 pin_f7
#define en36 pin_f6
#define en37 pin_f5
#define en38 pin_f4
#define en39 pin_f3
#define en40 pin_f2
/*****

```

VARIÁVEIS GLOBAIS

```

*****/
char codigoObtido[10];
char inicializador = 'i';
int bytesread = 0;
int val = 0;
int indice=0;
int digitoRTC[11];
int seglcd=0, minlcd=0, hrslcd=0, diaSemlcd=0, dialcd=0, mmlcd=0, anolcd=0, status=0,
time=154, time2=6;
int identificaEvento;
int dd=1, mm=1, aa=17, hh=0, mi=0;
int dateTimeLCD[10];
short int flagBack=FALSE;

```

```

short int flagFimAjuste=FALSE;
short int flagPrimeiraVez = TRUE;
short int flag1=TRUE, flag2=TRUE, flag3=TRUE, flag4=TRUE, flag5=TRUE,
flag6=TRUE, flag7=TRUE, flag8=TRUE, flag9=TRUE;
short int ent1, ent2, ent3, ent4, ent5, ent6, ent7, ent8, ent9, ent10, ent11, ent12, ent13,
ent14, ent15, ent16, ent17, ent18, ent19, ent20, ent21, ent22, ent23, ent24, ent25, ent26,
ent27, ent28, ent29, ent30, ent31, ent32, ent33, ent34, ent35, ent36, ent37, ent38, ent39,
ent40; //armazena estado lógico da primeira leitura.
short int ent1a, ent2a, ent3a, ent4a, ent5a, ent6a, ent7a, ent8a, ent9a, ent10a, ent11a, ent12a,
ent13a, ent14a, ent15a, ent16a, ent17a, ent18a, ent19a, ent20a, ent21a, ent22a, ent23a,
ent24a, ent25a, ent26a, ent27a, ent28a, ent29a, ent30a, ent31a, ent32a, ent33a, ent34a,
ent35a, ent36a, ent37a, ent38a, ent39a, ent40a; //armazena estado lógico atual.

```

```

/*****

```

PROTÓTIPOS DE FUNÇÕES

```

*****/

```

```

void verificaValor(void);
void display(void);
void telaPrincipal(void);
void autotest(void);
void monitoring(void);
void testaRele(void);
void testaBuzzer(void);
void testaRedundant(void);
void ajustaDataEHora(void);
int BINparaBCD(int valor);
void DS1307_ler_dados(void);
int BCDparaBIN(int valor);
void inicializa(void);
void le_EntradasDigitais(void);
void gravaSD(short int estado, int event);

```

```

/*****

```

INÍCIO

```

*****/

```

```

void main(){
    inicializa();           //inicializa o sistema.
    AjustaDataEHora();      //Solicita ao operador o ajuste de data e hora inicial.
    telaPrincipal();        //cria tela inicial com 2 botões (monitoring e autotest).
    t0on=1;                 //liga timer0.
    while(TRUE){
        if(kbhit()){         //monitora valor da serial do Display Touch.
            val=getc();       //se chegar um dado na usart, guarda em "val".
            if(val!=13){

```

```

        codigoObtido[bytesread] = val; //organiza os dados em um vetor.
        bytesread++;                  // incrementa o índice do vetor.
    }else{                            //Se receber 13 foi encerrado.
        bytesread=0;                  //volta o índice do vetor.
        verificaValor();              //Chama função para tratar o dado recebido.
    }
}
}
}
}
/*****

```

FUNÇÃO - VERIFICA VALOR

```

*****/
void verificaValor(){
    printf("bu 1");                  //DELETAR os botões! 1 é o nº do botão.
    putc(13);                        //13=enter(CR), necessário para o comando.
    delay_ms(2);
    printf("bu 2");
    putc(13);
    delay_ms(2);
    if(codigoObtido[0]=='B'){          //Se pressionou algum botão...
        output_high(pin_a3);          //Beep no buzzer
        delay_ms(60);
        output_low(pin_a3);
        if(codigoObtido[1]=='1'){      //Verifica se foi o botão 1.
            autotest();                //Função Autotest (testar periféricos).
            telaPrincipal();           //Retorna a tela principal novamente.
        }
        if(codigoObtido[1]=='2'){      //Ou, verifica se foi o botão 2.
            monitoring();               //Função Monitoring (ler entradas).
            if(flagBack){               //Se clicar em voltar, seta a flag.
                flagBack=FALSE;
                telaPrincipal();        //Retorna para tela principal.
            }
        }
    }
}
/*****

```

FUNÇÃO - DISPLAY

```

*****/
void display(){
    puts("T1");                      //exibe imagem 1 (logo IPEN).
    putc(13);
    putc('t');                        //instrução para enviar texto ao display.
    putc(34);                         //ASCII de aspas duplas (necessário).
}

```

```

    puts(" Unidade Autônoma");           //texto a ser enviado ao display
    putc(34);                             //fechamento de aspas duplas
    putc(13);                             //"enter" (CR), necessário para instruções.
    delay_ms(3000);
    puts("z");                             //apaga a tela
    delay_ms(100);
    putc(13);
}
/*****

FUNÇÃO - TELA PRINCIPAL (CRIA MENU PRINCIPAL COM 2 BOTÕES)
*****/

void telaPrincipal(){
    printf("bu 3");
    putc(13);
    delay_ms(20);
    puts("z");
    delay_ms(30);
    putc(13);
    delay_ms(50);
    putc('\t');                           //instrução necessária para enviar texto.
    putc(34);                             //ASCII de aspas duplas (necessário).
    puts(" UAMSR-SSIMC60");               //texto a ser enviado ao display.
    putc(34);                             //fechamento de aspas duplas.
    putc(13);                             //"enter" (CR).
    printf("bd 1 60 23 %cAutoteste%c", 34, 34); //Criando um botão! (1=id,
        //50=valor em x, 35=valor em y, 34=valor das aspas duplas(em ASCII)).
    putc(13);
    delay_ms(150);
    printf("bd 2 60 50 %cMonitorar%c", 34, 34);
    putc(13);
}
/*****

FUNÇÃO - MONITORING
*****/

void monitoring(){
    printf("bu 1");
    putc(13);
    delay_ms(40);
    printf("bu 2");
    putc(13);
    delay_ms(40);
    printf("bu 2");
    putc(13);
    delay_ms(40);

```

```

flagBack=FALSE;
puts("z");
delay_ms(50);
putc(13);
puts("z");
delay_ms(50);
putc(13);
putc('t');
putc(34);
puts("  Monitorando");
putc(34);
putc(13);
delay_ms(20);
DS1307_ler_datas();
putc(13);
printf("sc 10 17");           //pular de linha...10 em x e 17 em y.
putc(13);
putc('t');
putc(34);
printf("%d%d/%d%d/%d%d - %d%d:%d%d",
dateTimeLCD[0], dateTimeLCD[1],
dateTimeLCD[2], dateTimeLCD[3],
        dateTimeLCD[4], dateTimeLCD[5],
        dateTimeLCD[6], dateTimeLCD[7],
        dateTimeLCD[8], dateTimeLCD[9] );
putc(34);
putc(13);
delay_ms(20);
putc(13);
printf("bd 3 25 46 %cVoltar%c", 34, 34); //Criando um botão.
putc(13);
ent1=input_state(en1);       //Faz a leitura do estado inicial das entradas.
ent2=input_state(en2);
ent3=input_state(en3);
ent4=input_state(en4);
ent5=input_state(en5);
ent6=input_state(en6);
ent7=input_state(en7);
ent8=input_state(en8);
ent9=input_state(en9);
ent10=input_state(en10);
ent11=input_state(en11);
ent12=input_state(en12);
ent13=input_state(en13);

```



```
ent14=input_state(en14);
ent15=input_state(en15);
ent16=input_state(en16);
ent17=input_state(en17);
ent18=input_state(en18);
ent19=input_state(en19);
ent20=input_state(en20);
ent21=input_state(en21);
ent22=input_state(en22);
ent23=input_state(en23);
ent24=input_state(en24);
ent25=input_state(en25);
ent26=input_state(en26);
ent27=input_state(en27);
ent28=input_state(en28);
ent29=input_state(en29);
ent30=input_state(en30);
ent31=input_state(en31);
ent32=input_state(en32);
ent33=input_state(en33);
ent34=input_state(en34);
ent35=input_state(en35);
ent36=input_state(en36);
ent37=input_state(en37);
ent38=input_state(en38);
ent39=input_state(en39);
ent40=input_state(en40);
gravaSD(ent1a, 65); //65 é o char 'A'...O microcontrolador auxiliar
                    //recebe e trata os dados em char.

delay_ms(1);
gravaSD(ent2a, 66);
delay_ms(1);
gravaSD(ent3a, 67);
delay_ms(1);
gravaSD(ent4a, 68);
delay_ms(1);
gravaSD(ent5a, 69);
delay_ms(1);
gravaSD(ent6a, 70);
delay_ms(1);
gravaSD(ent7a, 71);
delay_ms(1);
gravaSD(ent8a, 72);
delay_ms(1);
```

```
gravaSD(ent9a, 73);
delay_ms(1);
gravaSD(ent10a, 74);
delay_ms(1);
gravaSD(ent11a, 75);
delay_ms(1);
gravaSD(ent12a, 76);
delay_ms(1);
gravaSD(ent13a, 77);
delay_ms(1);
gravaSD(ent14a, 78);
delay_ms(1);
gravaSD(ent15a, 79);
delay_ms(1);
gravaSD(ent16a, 80);
delay_ms(1);
gravaSD(ent17a, 81);
delay_ms(1);
gravaSD(ent18a, 82);
delay_ms(1);
gravaSD(ent19a, 83);
delay_ms(1);
gravaSD(ent20a, 84);
delay_ms(1);
gravaSD(ent21a, 85);
delay_ms(1);
gravaSD(ent22a, 86);
delay_ms(1);
gravaSD(ent23a, 87);
delay_ms(1);
gravaSD(ent24a, 88);
delay_ms(1);
gravaSD(ent25a, 89);
delay_ms(1);
gravaSD(ent26a, 90);
delay_ms(1);
gravaSD(ent27a, 91);
delay_ms(1);
gravaSD(ent28a, 92);
delay_ms(1);
gravaSD(ent29a, 93);
delay_ms(1);
gravaSD(ent30a, 94);
delay_ms(1);
```

```

    gravaSD(ent31a, 95);
    delay_ms(1);
    gravaSD(ent32a, 96);
    delay_ms(1);
    gravaSD(ent33a, 97);
    delay_ms(1);
    gravaSD(ent34a, 98);
    delay_ms(1);
    gravaSD(ent35a, 99);
    delay_ms(1);
    gravaSD(ent36a, 100);
    delay_ms(1);
    gravaSD(ent37a, 101);
    delay_ms(1);
    gravaSD(ent38a, 102);
    delay_ms(1);
    gravaSD(ent39a, 103);
    delay_ms(1);
    gravaSD(ent40a, 104);
    delay_ms(1);
    while(flagBack!=true){
        if(kbhit()){
            //Fica minitorando o Touch do Display.
            val=getc();
            //Se chegar um dado, guarda em "val".
            if(val!=13){
                //Se for diferente de 13 ainda não terminou.
                codigoObtido[bytesread] = val;
                bytesread++;
            }else{
                bytesread=0;
                if(codigoObtido[0]=='B'){
                    //Verifica se pressionou o botão.
                    output_high(pin_a3);
                    //Beep no buzzer
                    delay_ms(60);
                    output_low(pin_a3);
                    if(codigoObtido[1]=='3'){
                        flagBack=TRUE;
                    }
                }
            }
        }
        le_EntradasDigitais();
    }
    puts("z");
    delay_ms(300);
    putc(13);
}

```

```

/*****

```

FUNÇÃO - LÊ ENTRADAS DIGITAIS

```

*****/

```

```

void le_EntradasDigitais(){
    ent1a=input_state(en1);    //Faz a leitura do estado da Entrada 1 ATUAL para
                                //compará-la posteriormente com a leitura inicial.

    ent2a=input_state(en2);
    ent3a=input_state(en3);
    ent4a=input_state(en4);
    ent5a=input_state(en5);
    ent6a=input_state(en6);
    ent7a=input_state(en7);
    ent8a=input_state(en8);
    ent9a=input_state(en9);
    ent10a=input_state(en10);
    ent11a=input_state(en11);
    ent12a=input_state(en12);
    ent13a=input_state(en13);
    ent14a=input_state(en14);
    ent15a=input_state(en15);
    ent16a=input_state(en16);
    ent17a=input_state(en17);
    ent18a=input_state(en18);
    ent19a=input_state(en19);
    ent20a=input_state(en20);
    ent21a=input_state(en21);
    ent22a=input_state(en22);
    ent23a=input_state(en23);
    ent24a=input_state(en24);
    ent25a=input_state(en25);
    ent26a=input_state(en26);
    ent27a=input_state(en27);
    ent28a=input_state(en28);
    ent29a=input_state(en29);
    ent30a=input_state(en30);
    ent31a=input_state(en31);
    ent32a=input_state(en32);
    ent33a=input_state(en33);
    ent34a=input_state(en34);
    ent35a=input_state(en35);
    ent36a=input_state(en36);
    ent37a=input_state(en37);
    ent38a=input_state(en38);
    ent39a=input_state(en39);

```

```

ent40a=input_state(en40);
if(ent1a!=ent1){           //Se forem diferentes, grava o estado
                           //ATUAL. Se for igual, ignora...
    gravaSD(ent1a, 65);
    ent1=ent1a;           //Atualiza o valor inicial.
}
if(ent2a!=ent2){
    gravaSD(ent2a, 66);
    ent2=ent2a;
}
if(ent3a!=ent3){
    gravaSD(ent3a, 67);
    ent3=ent3a;
}
if(ent4a!=ent4){
    gravaSD(ent4a, 68);
    ent4=ent4a;
}
if(ent5a!=ent5){
    gravaSD(ent5a, 69);
    ent5=ent5a;
}
if(ent6a!=ent6){
    gravaSD(ent6a, 70);
    ent6=ent6a;
}
if(ent7a!=ent7){
    gravaSD(ent7a, 71);
    ent7=ent7a;
}
if(ent8a!=ent8){
    gravaSD(ent8a, 72);
    ent8=ent8a;
}
if(ent9a!=ent9){
    gravaSD(ent9a, 73);
    ent9=ent9a;
}
if(ent10a!=ent10){
    gravaSD(ent10a, 74);
    ent10=ent10a;
}
if(ent11a!=ent11){
    gravaSD(ent11a, 75);

```

```
        ent11=ent11a;
    }
    if(ent12a!=ent12){
        gravaSD(ent12a, 76);
        ent12=ent12a;
    }
    if(ent13a!=ent13){
        gravaSD(ent13a, 77);
        ent13=ent13a;
    }
    if(ent14a!=ent14){
        gravaSD(ent14a, 78);
        ent14=ent14a;
    }
    if(ent15a!=ent15){
        gravaSD(ent15a, 79);
        ent15=ent15a;
    }
    if(ent16a!=ent16){
        gravaSD(ent16a, 80);
        ent16=ent16a;
    }
    if(ent17a!=ent17){
        gravaSD(ent17a, 81);
        ent17=ent17a;
    }
    if(ent18a!=ent18){
        gravaSD(ent18a, 82);
        ent18=ent18a;
    }
    if(ent19a!=ent19){
        gravaSD(ent19a, 83);
        ent19=ent19a;
    }
    if(ent20a!=ent20){
        gravaSD(ent20a, 84);
        ent20=ent20a;
    }
    if(ent21a!=ent21){
        gravaSD(ent21a, 85);
        ent21=ent21a;
    }
    if(ent22a!=ent22){
        gravaSD(ent22a, 86);
```

```
        ent22=ent22a;
    }
    if(ent23a!=ent23){
        gravaSD(ent23a, 87);
        ent23=ent23a;
    }
    if(ent24a!=ent24){
        gravaSD(ent24a, 88);
        ent24=ent24a;
    }
    if(ent25a!=ent25){
        gravaSD(ent25a, 89);
        ent25=ent25a;
    }
    if(ent26a!=ent26){
        gravaSD(ent26a, 90);
        ent26=ent26a;
    }
    if(ent27a!=ent27){
        gravaSD(ent27a, 91);
        ent27=ent27a;
    }
    if(ent28a!=ent28){
        gravaSD(ent28a, 92);
        ent28=ent28a;
    }
    if(ent29a!=ent29){
        gravaSD(ent29a, 93);
        ent29=ent29a;
    }
    if(ent30a!=ent30){
        gravaSD(ent30a, 94);
        ent30=ent30a;
    }
    if(ent31a!=ent31){
        gravaSD(ent31a, 95);
        ent31=ent31a;
    }
    if(ent32a!=ent32){
        gravaSD(ent32a, 96);
        ent32=ent32a;
    }
    if(ent33a!=ent33){
        gravaSD(ent33a, 97);
```

```

        ent33=ent33a;
    }
    if(ent34a!=ent34){
        gravaSD(ent34a, 98);
        ent34=ent34a;
    }
    if(ent35a!=ent35){
        gravaSD(ent35a, 99);
        ent35=ent35a;
    }
    if(ent36a!=ent36){
        gravaSD(ent36a, 100);
        ent36=ent36a;
    }
    if(ent37a!=ent37){
        gravaSD(ent37a, 101);
        ent37=ent37a;
    }
    if(ent38a!=ent38){
        gravaSD(ent38a, 102);
        ent38=ent38a;
    }
    if(ent39a!=ent39){
        gravaSD(ent39a, 103);
        ent39=ent39a;
    }
    if(ent40a!=ent40){
        gravaSD(ent40a, 104);
        ent40=ent40a;
    }

    comparaValores(); //Compara valores obtidos com a resposta do Pannel de
                      //Controle. Se forem diferentes, dispara alarme.
}

/*****
FUNÇÃO - GRAVA_SD
*****/
void gravaSD(short int estadoLogico, int IdEvento){
    DS1307_ler_datas();
    identificaEvento = IdEvento;           //informa a identificação do evento.
    status = estadoLogico;                  //indica o estado lógico da entrada.
    fprintf(arduino, "%c%d%d%d%d%d%d%d%d%d%c", inicializador,
            dateTimeLCD[0], dateTimeLCD[1],

```



```

        dateTimeLCD[2], dateTimeLCD[3],
        dateTimeLCD[4], dateTimeLCD[5],
        dateTimeLCD[6], dateTimeLCD[7],
        dateTimeLCD[8], dateTimeLCD[9],
        identificaEvento, status);

    delay_ms(1000);
}
/*****

FUNÇÃO - AUTOTEST

*****/
void autotest(){
    delay_ms(40);
    printf("bu 1");
    putc(13);
    delay_ms(30);
    printf("bu 2");
    putc(13);
    delay_ms(30);
    printf("bu 3");
    putc(13);
    puts("z");
    putc('t');
    putc(34);
    puts("    Autoteste");
    putc(34);
    putc(13);
    delay_ms(10);
    printf("sc 25 20");           //colocando o cursor de texto na linha x e y.
    putc(13);
    putc('t');
    putc(34);
    puts(" RELE.....");
    putc(34);
    putc(13);
    testaRele();
    putc(13);
    printf("sc 25 35"); //colocando o cursor de texto na linha x e y.
    putc(13);
    putc('t');
    putc(34);
    puts(" BUZZER.....");
    putc(34);
    putc(13);
    testaBuzzer();

```

```

    putc(13);
    delay_ms(10);
    printf("sc 25 50"); //colocando o cursor de texto na linha x e y.
    putc(13);
    putc('t');
    putc(34);
    puts(" SECUNDARIO.");
    putc(34);
    putc(13);
    testaRedundant();
    delay_ms(1500);
    puts("z"); //apaga a tela
}
/*****
                                     FUNÇÃO - TESTA RELÊ
*****/
void testaRele(){
    int i;
    for(i=0; i<6; i++){
        output_high(pin_f0);
        delay_ms(300);
        output_low(pin_f0);
        delay_ms(300);
    }
}
/*****
                                     FUNÇÃO - TESTA BUZZER
*****/
void testaBuzzer(){
    int i;
    for(i=0; i<6; i++){
        output_high(pin_a3);
        delay_ms(300);
        output_low(pin_a3);
        delay_ms(300);
    }
}
/*****
                                     FUNÇÃO - TESTA PIC REDUNDANTE
*****/
void testaRedundant(){
    int i;
    for(i=0; i<6; i++){
        if(chamado()){ //se o pulso for respondido pelo mic. secundário...

```

```

        output_high(pin_j7);
        delay_ms(300);
        output_low(pin_j7);
        delay_ms(300);
    }
}

/*****
FUNÇÃO - AJUSTA DATA E HORA DO RTC
*****/
void ajustaDataEHora(){
    int bcd_sec, bcd_min, bcd_hrs, bcd_dia_semana, bcd_dia_mes, bcd_mes, bcd_ano;
    delay_ms(30);
    printf("bu 1");
    putc(13);
    delay_ms(20);
    printf("bu 2");
    putc(13);
    delay_ms(20);
    printf("bu 2");
    putc(13);
    delay_ms(20);
    printf("bu 3");
    putc(13);
    delay_ms(20);
    printf("bu 3");
    putc(13);
    puts("z");
    putc(13);
    putc('t');
    putc(34);
    puts("Ajuste:");
    putc(34);
    putc(13);
    delay_ms(70);
    printf("bd 4 32 22 %cMais %c", 34, 34);
    putc(13);
    delay_ms(20);
    printf("bd 5 32 52 %cMenos%c", 34, 34);
    putc(13);
    delay_ms(20);
    printf("bd 6 100 52 %c OK %c", 34, 34);
    putc(13);

```

```

printf("sc 43 0");           //colocando o cursor de texto na linha x e y.
putc(13);
delay_ms(20);
putc('t');
putc(34);
printf("01/01/17-00:00");   //data/hora padrão
putc(34);
putc(13);
delay_ms(20);
while(flagFimAjuste!=true){  //Enquanto não terminar de ajustar o relógio.
    if(kbhit()){              //Fica monitorando o Touch do Display.
        val=getc();           //Guarda na variável "val".
        if(val!=13){          //Se for diferente de 13 ainda não terminou.
            codigoObtido[bytesread] = val;
            bytesread++;
        }else{                //Se o dado for igual a 13, pode tratar.
            bytesread=0;       //retorna o índice do vetor para zero.
            if(codigoObtido[0]=='B'){ //Verifica o botão.
                output_high(pin_a3); //Beep no buzzer
                delay_ms(60);
                output_low(pin_a3);
                if(codigoObtido[1]=='4'){ //BOTÃO MAIS
                    if(indice==0){        //0 é o ajuste do DIA.
                        dd++;
                        if(dd>=31) dd=31;
                    }else if(indice==1){   //1 é o MÊS.
                        mm++;
                        if(mm>=12) mm=12;
                    }else if(indice==2){   //2 é o ANO.
                        aa++;
                        if(aa>=99) aa=99;
                    }else if(indice==3){   //3 é a HORA.
                        hh++;
                        if(hh>=23) hh=23;
                    }else if(indice==4){   //4 é o MINUTO.
                        mi++;
                        if(mi>=59) mi=59;
                    }
                }
            }else if(codigoObtido[1]=='5'){ //BOTÃO MENOS
                if(indice==0){ //índice=0 é o ajuste do DIA.
                    dd--;
                    if(dd<=1) dd=1;
                }else if(indice==1){      //1 é o MÊS.
                    mm--;

```

```

        if(mm<=1) mm=1;
    }else if(indice==2){          //2 é o ANO.
        aa--;
        if(aa<1) aa=0;
    }else if(indice==3){          //3 é a HORA.
        hh--;
        if(hh<1) hh=0;
    }else if(indice==4){          //4 é o MINUTO.
        mi--;
        if(mi<1) mi=0;
    }
}
}else if(codigoObtido[1]=='6'){    //BOTÃO OK.
    indice++;
    if(indice>=5){                  //5 =terminou.
        flagFimAjuste=TRUE;        //Sai do while.
    }
}
}
putc('t');
putc(34);
dateTimeLCD[0] = dd/10;            //dia
dateTimeLCD[1] = dd%10;           //dia
dateTimeLCD[2] = mm/10;           //mes
dateTimeLCD[3] = mm%10;           //mes
dateTimeLCD[4] = aa/10;           //ano
dateTimeLCD[5] = aa%10;           //ano
dateTimeLCD[6] = hh/10;           //hora
dateTimeLCD[7] = hh%10;           //hora
dateTimeLCD[8] = mi/10;           //minuto
dateTimeLCD[9] = mi%10;           //minuto

delay_ms(5);
printf("%d%d/%d%d/%d%d-%d%d:%d%d",
    dateTimeLCD[0], dateTimeLCD[1],
    dateTimeLCD[2], dateTimeLCD[3],
    dateTimeLCD[4], dateTimeLCD[5],
    dateTimeLCD[6], dateTimeLCD[7],
    dateTimeLCD[8], dateTimeLCD[9]);
putc(34);
putc(13);
    }
}
}
bcd_sec = BINparaBCD(0);          // Converte binario para bcd.

```

```

bcd_min = BINparaBCD(mi);
bcd_hrs = BINparaBCD(hh);
bcd_dia_semana = BINparaBCD(1);
bcd_dia_mes = BINparaBCD(dd);
bcd_mes = BINparaBCD(mm);
bcd_ano = BINparaBCD(aa);
i2c_start();
i2c_write(Ende_escrita);
i2c_write(SEGUNDOS);
i2c_write(bcd_sec);
i2c_write(bcd_min);
i2c_write(bcd_hrs);
i2c_write(bcd_dia_semana);
i2c_write(bcd_dia_mes);
i2c_write(bcd_mes);
i2c_write(bcd_ano);
i2c_stop();
output_high(pin_a3);
delay_ms(1500);
output_low(pin_a3);
printf("bu 4");
putc(13);
delay_ms(20);
printf("bu 5");
putc(13);
delay_ms(20);
printf("bu 6");
putc(13);
puts("z");
putc(13);
delay_ms(80);
}
/*****
FUNÇÃO - CONVERTE BINÁRIO PARA BCD
*****/
int BINparaBCD(int valor){
    int BCD=0;
    while(valor>=1){
        if(valor >= 10){                                     //incrementa casas decimais.
            valor -= 10;
            BCD += 0x10;
        }else {
            BCD += valor;
            valor=0;
        }
    }
}

```

```

    }
}
return(BCD);           //retorna o valor.
}
/*****
FUNÇÃO - LER DATA NO DS1307
*****/
void DS1307_ler_dados(void) {
    int bin_sec, bin_min, bin_hrs, bin_dia_semana, bin_dia_mes, bin_mes, bin_ano;
    i2c_start();
    i2c_write(Ende_escrita);
    i2c_write(SEGUNDOS);           // inicia lendo segundos.
    i2c_start();
    i2c_write(Ende_leitura);
    bin_sec      = i2c_read();
    bin_min      = i2c_read();
    bin_hrs      = i2c_read();
    bin_dia_semana = i2c_read();
    bin_dia_mes  = i2c_read();
    bin_mes      = i2c_read();
    bin_ano      = i2c_read();
    i2c_stop();
    seglcd       = BCDparaBIN(bin_sec);    // converte datas de bcd para inteiro.
    minlcd       = BCDparaBIN(bin_min);
    hrslcd       = BCDparaBIN(bin_hrs & 0x3F);
    diaSemlcd    = BCDparaBIN(bin_dia_semana);
    dialcd       = BCDparaBIN(bin_dia_mes);
    mmlcd        = BCDparaBIN(bin_mes);
    anolcd       = BCDparaBIN(bin_ano);
    dateTimeLCD[0] = dialcd/10;           //dia
    dateTimeLCD[1] = dialcd%10;           //dia
    dateTimeLCD[2] = mmlcd/10;           //mes
    dateTimeLCD[3] = mmlcd%10;           //mes
    dateTimeLCD[4] = anolcd/10;          //ano
    dateTimeLCD[5] = anolcd%10;          //ano
    dateTimeLCD[6] = hrslcd/10;          //hora
    dateTimeLCD[7] = hrslcd%10;          //hora
    dateTimeLCD[8] = minlcd/10;          //minuto
    dateTimeLCD[9] = minlcd%10;          //minuto
}
/*****
FUNÇÃO - CONVERTE BCD PARA BINÁRIO
*****/
int BCDparaBIN(int valor){

```

```

    int temp=15;
    int temp2=240;
    temp &= valor;
    temp2 &= valor;
    temp2 >>= 4;
    temp2 *= 10;
    valor=temp+temp2;
    return(valor);
}
/*****

FUNÇÃO - INICIALIZA O SISTEMA
*****/
void inicializa(){
    int i = 0, zera;
    t0on=0; //mantém desligado.
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_4);
    T8bits=1;
    set_timer0(130);
    enable_interrupts(GLOBAL|TMR0);
    putc(13); //inicialização do display.
    puts("z"); //apaga a tela do display.
    for(zera=11; zera!=0; zera--){ //zera as posições da matriz.
        digitoRTC[zera]=0;
    }
    for(i=0; i<5; i++){ //inicialização visual e sonora.
        output_high(pin_j6); //led
        output_high(pin_j7); //led
        output_high(pin_a3); //buzzer
        delay_ms(100);
        output_low(pin_j7);
        output_low(pin_j6);
        output_low(pin_a3);
        delay_ms(100);
    }
    display(); //Exibe o logo do IPEN ao ligar a unidade.
}

```



```
/******  
FUNÇÃO – TRATAMENTO TIMER 0  
*****/  
#INT_TMR0  
void TMR0_isr(){  
    set_timer0(0);  
    time--;  
    if(time<=0){  
        time=154;  
        time2--;  
        if(time2<=0){  
            output_toggle(pin_C0); //Envia pulso ao microcontrolador secundário...  
            time2=6;  
        }  
    }  
}  
/******  
FIM  
*****/
```

APÊNDICE C – Código do microcontrolador secundário

```
#include<16F1503.h>
#fuses HS, NOWDT, NOLVP, NOMCLR
#use delay (clock=20000000)
#byte intcon = 0xff2
#bit T0IF = intcon.2
#byte T0CON =0xfd5
#bit T0ON =T0CON.7
#bit T8bits = T0CON.6
#define alarme          pin_a1
#define resetPrincipal   pin_a3
#define entrada          pin_a5
int cont=154;
int cont=6;
int erro = 1;
short int pulsoRecebido = 0;
short int pulsoAnterior = 0;

void main(){
    t0on=1;
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_4);
    T8bits=1;
    set_timer0(0);
    pulsoRecebido = input_state(entrada);
    if(pulsoRecebido==1) {
        pulsoAnterior=0;
    }else{
        pulsoAnterior=1;
    }
    while(true){
        if(T0IF){
            set_timer0(0);
            T0IF = 0;
            cont--;
            if(cont>=0){
                cont=154;
                cont2--;
                if(cont2<=0){
                    cont2=6;
                    pulsoRecebido = input_state(entrada);
                    if(pulsoRecebido==pulsoAnterior) {
                        pulsoAnterior=~pulsoAnterior;
                        output_low(alarme);
                    }
                }
            }
        }
    }
}
```

```
        else{
            erro++;
        }
    }
}
}
if{erro>=2){
    output_high(alarme);
    output_low(resetPrincipal);
    delay_ms(1000);
    output_high(resetPrincipal);
    erro=1;
}
}
}
```