

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA  
CELSO SUCKOW DA FONSECA – CEFET/RJ**

**DESENVOLVIMENTO DO EXPERIMENTO  
ANTÁRTICO DE MONITORAÇÃO DE RAIOS  
CÓSMICOS PARA O MÓDULO CRIOSFERA I**

Por:

Ulisses de Freitas Carneiro da Graça

Orientadores:

André Luis Costa Canella (CEFET)

André Massafferri Rodrigues (CBPF)

**Rio de Janeiro  
Maio de 2015**

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA  
CELSO SUCKOW DA FONSECA – CEFET/RJ**

**DESENVOLVIMENTO DO EXPERIMENTO  
ANTÁRTICO DE MONITORAÇÃO DE RAIOS  
CÓSMICOS PARA O MÓDULO CRIOSFERA I**

Por:

Ulisses de Freitas Carneiro da Graça

Projeto final de curso apresentado em cumprimento às  
normas do Departamento de Educação Superior  
do CEFET/RJ, como parte dos requisitos para obtenção  
do título de Bacharel em Engenharia Eletrônica

Aprovado por banca examinadora mediante defesa oral  
pública realizada às 14h30m do dia 27 de maio de 2015, no  
auditório VIII do CEFET/RJ (unidade sede - Maracanã)

Orientadores:

André Luis Costa Canella (CEFET)

André Massafferri Rodrigues (CBPF)

Banca examinadora:

Dra. Ana Lucia Ferreira de Barros (CEFET)

Dr. André Luis Costa Canella (CEFET)

Dr. André Massafferri Rodrigues (CBPF)

Dr. Mauro Sandro dos Reis (CEFET)

Prof. Paulo Cesar Bittencourt (CEFET)

**Rio de Janeiro  
Maio de 2015**

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

G729 Graça, Ulisses de Freitas Carneiro da  
Desenvolvimento do experimento antártico de monitoração de  
raios cósmicos para o módulo Criosfera I / Ulisses de Freitas  
Carneiro da Graça.—2015.  
xii, 70f. + apêndices e anexo : il.color. , grafs. , tabs. ; enc.

Projeto Final (Graduação) Centro Federal de Educação  
Tecnológica Celso Suckow da Fonseca , 2015.

Bibliografia : f.69-70

Orientadores : André Luis Costa Canella  
André Massafferri

1. Engenharia eletrônica. 2. Raios cósmicos. 3. Mudanças  
climáticas. 4. Radiação – Medidas de segurança. 5. Pesquisa  
espacial. I. Canella, André Luis Costa (Orient.). II. Massafferri,  
André (Orient.). III. Título.

CDD 621.38

## **DEDICATÓRIA**

Dedico este trabalho a todos que me ajudaram a chegar até aqui, em especial a minha família e a todos os professores que, desde os níveis mais fundamentais até os mais específicos, me educaram. Dedico também aos amigos que fiz ao longo da graduação, que me motivaram a concluí-la e que, certamente, transformaram-se em grandes amigos para toda a vida.

## **AGRADECIMENTOS**

Agradeço ao Centro Brasileiro de Pesquisas Físicas (CBPF), por toda a infraestrutura disponibilizada para a execução deste projeto, ao pesquisador André Massafferri pela orientação dada, ao professor Heitor Evangelista (UERJ) que coordena o módulo Criosfera I, ao professor Jesse Costa (CEFET) que abriu as portas para que este trabalho fosse possível, ao INCT da Criosfera, CNPq e PROANTAR, por viabilizar a logística da missão de implementação do experimento, e a todos os que contribuíram direta ou indiretamente com a conclusão deste trabalho.

## RESUMO

Alterações climáticas globais têm atraído grande atenção da comunidade científica internacional devido ao elevado potencial de interferência que apresentam à atividade humana. Entender o que provoca tamanhas alterações é fundamental para que se possam compreender as variações climáticas e estabelecer a contribuição humana à mesma. Recentemente foi proposta a existência de relação direta entre o fluxo de radiação cósmica galáctica e a formação de núcleos de condensação, o primeiro estágio para a formação de nuvens. A plataforma continental antártica apresenta características únicas que fazem com que o fluxo de radiação cósmica tenha um efeito potencialmente diferenciado e efetivo em relação a outras regiões do planeta. O objetivo deste trabalho é desenvolver um dispositivo eletrônico capaz de efetuar monitoração de fluxo da radiação cósmica galáctica, operando de forma totalmente autônoma e ininterrupta em condições climáticas extremas no módulo Criosfera I – situado no interior da Criosfera Antártica (latitude 84°S) – executando assim a primeira fase do projeto *CRE@AT* (*Cosmic Rays Experiment at Antarctica*).

Palavras-chave: Clima; Radiação Cósmica Galáctica; Instrumentação Eletrônica

## **ABSTRACT**

*International scientific communities have shown great interest in climate change due to the intense interference of human activities. It is essential to understand what causes strong climate variations and analyze the influence of human activities in this process. Recently, studies indicate that there maybe a direct relation between cosmic radiation flow and the formation of condensation nucleus, which is the first stage of the formation of clouds. Antarctic continental shelves present unique characteristics that allow cosmic radiation flow to have a potentially unique and powerful effect in relation to other areas of the planet. The aim of this work is to develop an electronic device capable of monitoring autonomously and uninterruptedly cosmic radiation in severe climate conditions of Module Cryosphere I – situated in the interior of the Antarctic continental shelf (latitude 84°S) – which is the first part of this project CRE@AT (Cosmic Rays Experiment at Antarctica).*

*Keywords: Climate, Galactic Cosmic Rays, Electronic Instrumentation*

# SUMÁRIO

1. Introdução.....	1
1.1. Motivação .....	2
1.1.1. Projeto <i>CRE@AT</i> .....	5
1.2. Objetivos .....	6
1.3. Justificativa .....	6
1.4. Metodologia e Trabalho Realizado.....	7
1.5. Organização do Trabalho .....	9
2. Contexto do Trabalho.....	10
2.1. Os Raios Cósmicos .....	11
2.2. O Programa Antártico Brasileiro .....	12
2.3. O Módulo Criosfera 1 .....	13
3. Desenvolvimento do projeto .....	16
3.1. Módulos Incorporados .....	18
3.1.1. Módulo Detector.....	19
3.1.1.1. Tiras plásticas cintilantes ( <i>SciTiles</i> ) .....	24
3.1.1.2. Fibra óptica <i>Wavelength Shifter</i> .....	25
3.1.1.3. Fotomultiplicadora <i>MaPMT</i> .....	26
3.1.1.4. Eletrônica <i>Front-End</i> .....	28
3.1.2. Módulo de Alta Tensão .....	30
3.1.3. Módulo de Interface ao Barramento <i>VMEbus</i> .....	31
3.2. Módulos Desenvolvidos.....	32
3.2.1. Módulo Contador de Eventos Coincidentes .....	32
3.2.1.1. Temporizador de Entrada ( <i>Gate Timer</i> ) .....	34
3.2.1.2. Detecção de eventos coincidentes .....	37
3.2.1.3. Contagem de eventos.....	37
3.2.1.4. Mapa de registradores.....	39
3.2.1.5. Transmissão serial dos registradores .....	43
3.2.1.6. Acesso aos registradores pelo <i>VMEbus</i> .....	44
3.2.2. Módulo de Aquisição de Dados .....	45
3.2.3. Módulo de Calibração .....	46
3.2.3.1. Algoritmo de Calibração .....	47
3.2.4. Módulo de Alimentação Elétrica.....	48
3.2.4.1. Requisitos do Projeto.....	49



3.2.4.2.	Dimensionamento de Potência .....	49
3.2.4.3.	Importância das Características de Isolamento .....	51
3.2.4.4.	Conversores de Tensão .....	52
3.2.4.5.	Diagramas Esquemáticos.....	52
3.2.5.	Módulo Controlador de Carga.....	53
4.	Resultados Obtidos.....	54
5.	Conclusão .....	62
5.1.	Continuidade do Projeto <i>CRE@AT</i> .....	62
5.2.	Citações do Projeto na Mídia.....	64
	Referências Bibliográficas.....	69
	APENDICE A: Problemas na geração de energia eólica do módulo Criosfera I.....	71
	A.1. Problemas com as Turbinas FORGEN.....	72
	A.2. Estudo do Problema .....	72
	A.3. Solução Proposta .....	75
	APENDICE B: Diagrama esquemático do Módulo Contador de Eventos Coincidentes..	78
	APENDICE C: Diagramas esquemáticos do Módulo de Alimentação Elétrica .....	81
	APENDICE D: Código C do Módulo de Aquisição de Dados .....	86
	APENDICE E: Código C do Módulo Controlador de Carga.....	98
	APENDICE F: Caderno de códigos do Módulo Contador.....	113
	APENDICE G: Código C para análise de dados no ROOT .....	187
	ANEXO A: Relatório da missão Criosfera 2014 .....	195

## LISTA DE FIGURAS

Figura 1 – Interação dos raios cósmicos galácticos com a atmosfera [1].....	2
Figura 2 – Interação simplificada dos raios cósmicos galácticos com a atmosfera .....	2
Figura 3 – Interação do vento solar com os raios cósmicos galácticos .....	3
Figura 4 – Formação de nuvens e variação da temperatura .....	3
Figura 5 – Mecanismo de ação dos raios cósmicos na formação de nuvens [5] .....	4
Figura 6 – Cronograma do projeto <i>CRE@AT</i> .....	6
Figura 7 – Localização do módulo Criosfera I .....	13
Figura 8 – Laboratório do módulo Criosfera I .....	14
Figura 9 – Face Sul do módulo Criosfera I .....	14
Figura 10 – <i>Layout</i> básico do experimento .....	16
Figura 11 – Diagrama em blocos do sistema <i>CRE@AT</i> .....	18
Figura 12 – Diagrama em blocos com ênfase ao que foi incorporado ao projeto .....	19
Figura 13 – Diagrama em blocos com destaque ao detector .....	19
Figura 14 – Transformações de energia no detector.....	20
Figura 15 – <i>Layout</i> de montagem das tiras cintilantes .....	20
Figura 16 – <i>Layout</i> dos grupos de coincidência .....	21
Figura 17 – Detector de partículas cósmicas .....	22
Figura 18 – Detalhe do detector .....	22
Figura 19 – Máscara de acoplamento óptico entre as fibras <i>WLS</i> e a foto multiplicadora.....	23
Figura 20 – Alinhamento da máscara de fibras ópticas.....	23
Figura 21 – Montagem mecânica da tira cintilante com fibra <i>wavelength shifter</i> .....	25
Figura 22 – Espectro de absorção e emissão - <i>KURARAY WLS Fibers</i> [16].....	26
Figura 23 – Fibras ópticas <i>WLS</i> .....	26
Figura 24 – Aplicação de Alta Tensão numa fotomultiplicadora [17] .....	27
Figura 25 – <i>MaPMT</i> HAMAMATSU H6568MOD .....	27
Figura 26 – Diagrama em blocos da <i>Front-End</i> [18] .....	28
Figura 27 – Estágio amplificador do módulo de <i>front-end</i> [18].....	29
Figura 28 – Estágio discriminador do módulo <i>front-end</i> [18].....	29
Figura 29 – Diagrama em blocos com destaque ao módulo de alta tensão .....	30
Figura 30 – Diagrama em blocos com ênfase ao que foi desenvolvido neste projeto.....	32
Figura 31 – Diagrama em blocos com destaque ao módulo contador.....	33

Figura 32 – Diagrama em blocos do módulo contador de eventos coincidentes .....	33
Figura 33 – Resposta ao impulso na entrada do <i>FPGA</i> .....	35
Figura 34 – Diagrama de tempo do circuito <i>gate timer</i> .....	36
Figura 35 – Circuito <i>gate timer</i> .....	36
Figura 36 – Circuito detector de coincidências .....	37
Figura 37 – Circuito contador .....	38
Figura 38 – Diagrama em blocos com destaque ao módulo de aquisição de dados .....	45
Figura 39 – Fluxograma lógico do módulo de aquisição de dados .....	46
Figura 40 – Diagrama em blocos com destaque ao módulo de calibração .....	47
Figura 41 – Diagrama em blocos com destaque ao módulo de alimentação elétrica .....	49
Figura 42 – Sistema de controle das turbinas eólicas .....	53
Figura 43 – Histograma de fluxo no Brasil para o grupo de coincidência 1 .....	54
Figura 44 – Histograma de fluxo no Brasil para o grupo de coincidência 2 .....	55
Figura 45 – Histograma de fluxo no Brasil para o grupo de coincidência 3 .....	55
Figura 46 – Histograma de fluxo no Brasil para o grupo de coincidência 4 .....	56
Figura 47 – Histograma de fluxo no Brasil para o grupo de coincidência 5 .....	56
Figura 48 – Fluxo observado no CBPF .....	57
Figura 49 – Histograma de fluxo na Antártica para o grupo de coincidência 1 .....	58
Figura 50 – Histograma de fluxo na Antártica para o grupo de coincidência 2 .....	58
Figura 51 – Histograma de fluxo na Antártica para o grupo de coincidência 3 .....	59
Figura 52 – Histograma de fluxo na Antártica para o grupo de coincidência 4 .....	59
Figura 53 – Histograma de fluxo na Antártica para o grupo de coincidência 5 .....	60
Figura 54 – Fluxo observado na Antártica .....	60
Figura 55 – Captura de Tela do Portal Brasil .....	64
Figura 56 – Captura de Tela do Portal CBPF .....	65
Figura 57 – Captura de Tela do Portal Agência CT&I .....	66
Figura 58 – Captura de Tela do Portal Inovação Tecnológica .....	67
Figura 59 – Fotografia oficial da missão Criosfera 2014/2015 .....	68
Figura 60 – Incidência solar e velocidade do vento no módulo Criosfera 1 em 2014 .....	71
Figura 61 – Controlador de Carga fornecido com o gerador eólico VENTUZ 70 .....	73
Figura 62 – Estágio de potência do controlador FORGEN .....	73
Figura 63 – Estágio de alimentação do controlador FORGEN .....	74
Figura 64 – Estágio de acionamento do controlador FORGEN .....	74
Figura 65 – Estágio de controle (simplificado) do controlador FORGEN .....	74

Figura 66 – Circuito para medição da velocidade de rotação da turbina .....	76
Figura 67 – Malhas de controle do circuito construído .....	76

## LISTA DE TABELAS

Tabela 1 – Módulos da fase 1 do experimento antártico de raios cósmicos .....	18
Tabela 2 – Mapa de endereços .....	39
Tabela 3 – Conversores de Energia DC-DC.....	50
Tabela 4 – Consumo por linha de alimentação.....	51
Tabela 5 – Consumo total e eficiência da conversão.....	51
Tabela 6 – Conversores de Energia DC-DC.....	52

# 1. Introdução

Recentes estudos sugerem que os raios cósmicos – partículas de alta energia que atravessam continuamente a atmosfera terrestre – exercem papel fundamental no clima de nosso planeta. É muito limitada ainda, entretanto, a observação deste fenômeno a nível global e a aquisição de dados que possam relacioná-lo de maneira direta aos modelos climatológicos existentes. O estabelecimento de estações de monitoramento contínuo da atividade de partículas cósmicas permitirá a integração destes dados a modelos matemáticos tornando os estudos de climatologia e as previsões meteorológicas mais precisas e contribuindo, assim, para maior um entendimento sobre o clima e um incremento na precisão da previsão de fenômenos naturais, como tornados, chuvas e secas. Desta forma, é de grande importância o desenvolvimento de plataformas de monitoramento contínuo do fluxo de raios cósmicos.

O presente trabalho tem como objetivo desenvolver a primeira fase do experimento brasileiro de monitoração de raios cósmicos na Criosfera Antártica (*CRE@AT*), instalando uma plataforma de monitoração do fluxo de radiação cósmica galáctica no módulo de pesquisas Criosfera I, próximo ao polo Sul terrestre.

Projetada para operar de forma contínua e desassistida, esta plataforma transmitirá pequena quantidade de informações via satélite e manterá registro dos dados em mídia não-volátil a ser recuperada na próxima missão, permitindo a análise da influência que a atividade cósmica exerce sobre padrões atmosféricos de modo a viabilizar, assim, os mais recentes estudos climatológicos.

Neste sentido, diversos recursos tiveram de ser empregados de modo a tornar o experimento autossuficiente e, na medida do possível, imune a falhas. A proposta atende assim a requisitos de operação remota desassistida em ambiente inóspito, monitoração telemétrica via satélite de parte dos resultados e alta disponibilidade.

Os resultados preliminares de calibração obtidos foram bastante satisfatórios e condizentes com a atividade dos raios cósmicos esperada para aquela região, indicando que o experimento funciona. Entretanto, os principais resultados científicos virão após a próxima missão, com o recolhimento dos dados brutos gravados na mídia local.

## 1.1. Motivação

Os Raios Cósmicos Galácticos (RCG) são partículas estáveis que interagem com a atmosfera provocando chuveis, com múons dominando o estado final. Este processo é ilustrado nas Figuras 1 e 2.

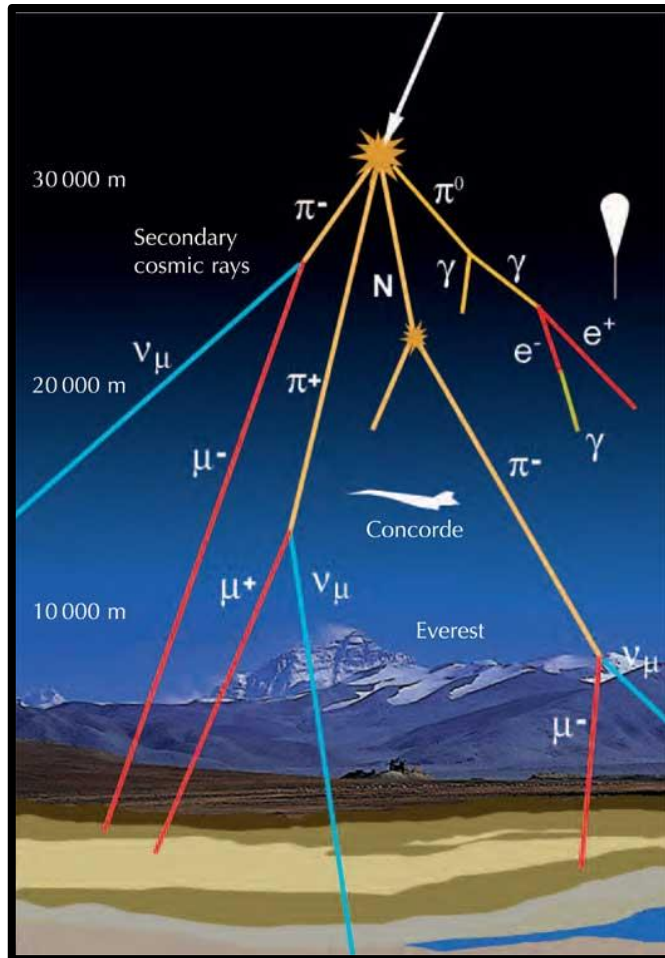


Figura 1 – Interação dos raios cósmicos galácticos com a atmosfera [1]

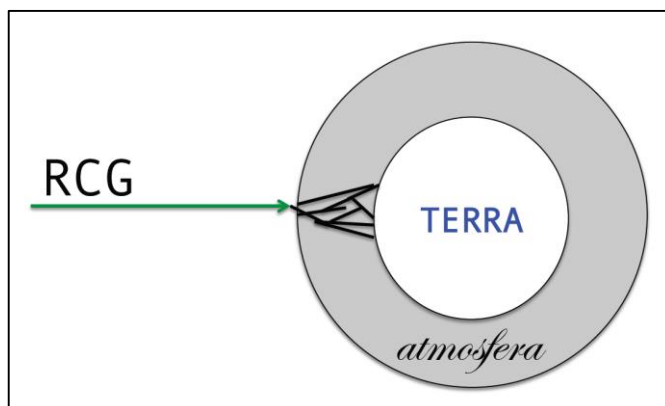


Figura 2 – Interação simplificada dos raios cósmicos galácticos com a atmosfera

Ventos Solares são o fluxo de partículas ionizantes de baixa energia emitidas pelo sol. A interação entre o campo magnético terrestre e o campo magnético associado aos ventos solares blindam a Terra da RCG. Este processo é ilustrado na Figura 3.

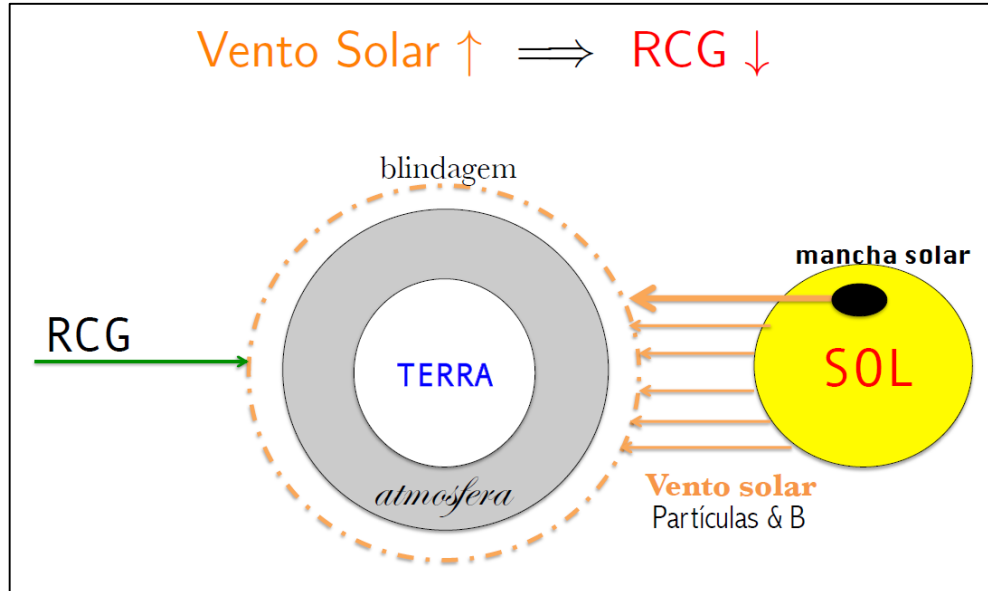


Figura 3 – Interação do vento solar com os raios cósmicos galácticos

Já na atmosfera terrestre, a formação de nuvens provoca reflexão da radiação solar reduzindo a temperatura na superfície do planeta. Esta característica é dominante em relação ao efeito estufa associado, fazendo com que se observe a relação ilustrada na Figura 4.

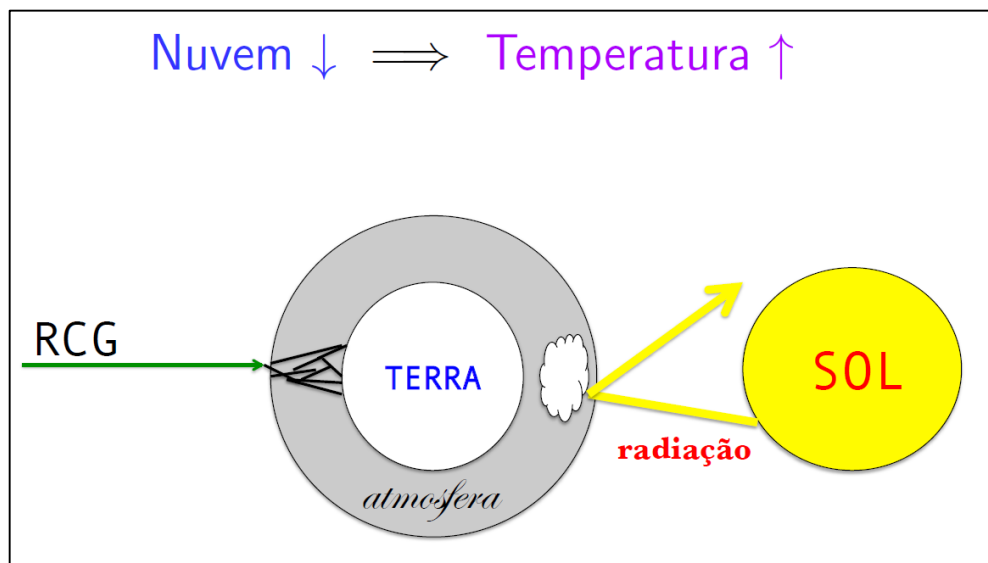


Figura 4 – Formação de nuvens e variação da temperatura



Os raios cósmicos galácticos são a fonte dominante de íons na troposfera/estratosfera e criam radicais livres. Tem sido proposto [2], [3], [4] que íons formam pontos de nucleação para formar aerossóis, que geram núcleos de condensação que são, por sua vez, um gatilho para a formação de nuvens. Tal cadeia implica na seguinte relação de consequências:

Vento Solar ↑ → RCG ↓ → Nuvem ↓ → Temperatura ↑

Atualmente, o ponto fraco desta cadeia é a relação entre radiação cósmica galáctica e formação de núcleos de condensação de nuvens vista na Figura 5, pois o estudo experimental em larga escala da radiação cósmica galáctica associada à atividade climática é relativamente recente. A correlação entre vento solar, em particular nos ciclos de 11 anos das manchas solares, e clima global já é bem conhecida a muito tempo e vem causando divergências entre cientistas sobre ação humana como causa do aquecimento global observado nos últimos séculos.

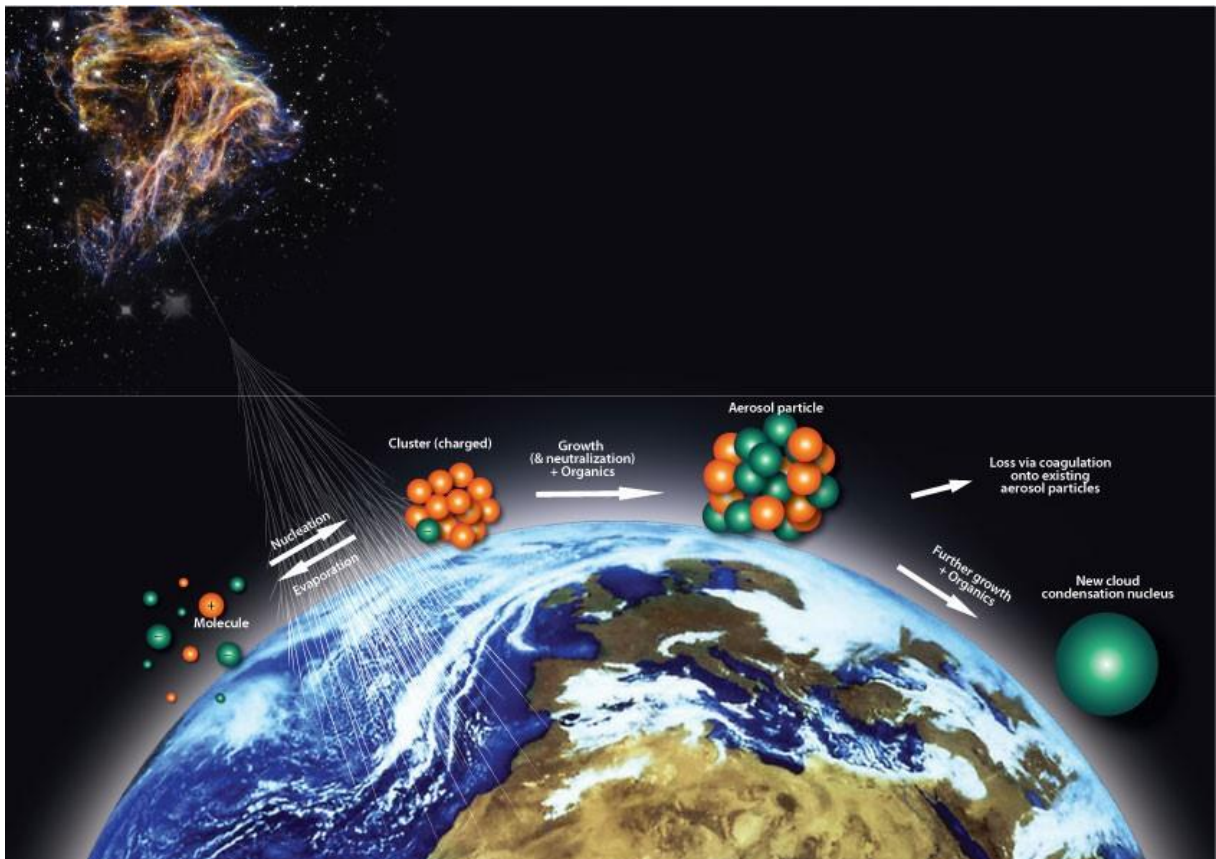


Figura 5 – Mecanismo de ação dos raios cósmicos na formação de nuvens [5]

Experimentos – como *CLOUD*, no *CERN* [6], em que uma grande câmara foi construída para observar os efeitos de um feixe de íons em detector de nuvens emulando atmosfera – têm buscado esclarecer esta relação.

Implementar, porém, o monitoramento dos Raios Cósmicos em atmosfera real, livre de poluentes, em local sob grande influência da radiação cósmica galáctica e verificar sua possível correlação com a formação de nuvens, permitindo ainda sua integração com outros sistemas de detecção de raios cósmicos para observação de chuveiros globais, é um grande atrativo para esta pesquisa. Neste contexto, e devido às características únicas presentes na região antártica, surge no começo de 2014 o projeto *CRE@AT*.

### **1.1.1. Projeto *CRE@AT***

O grupo de altas energias do CBPF foi convidado a participar do ProAntar com o intuito de desenvolver a área de física de altas energias no módulo de pesquisas Criosfera I. O projeto *CRE@AT* foi proposto pelo CBPF para aproveitar as condições únicas presentes no polo sul terrestre. São elas:

- As linhas do campo magnético terrestre são ortogonais às existentes nos trópicos, fazendo com que o fluxo de radiação cósmica tenha um efeito diferenciado e efetivo em relação a outros parâmetros meteorológicos.
- A existência de um manto de gelo com aproximadamente 1 km de espessura diminui fortemente os níveis de radioatividade proveniente de minerais contendo núcleos pesados presentes no solo.
- A diminuição do efeito da rotação no movimento linear terrestre pode contribuir para diversos estudos de física de partículas atuais, como os modelos envolvendo matéria escura.
- A redução do tamanho das camadas atmosféricas na região polar – devido à própria geometria da Terra – altera a dinâmica dos chuveiros cósmicos de forma significativa.

O projeto *CRE@AT*, dividido em três etapas, prevê a realização de experimento para monitoração de raios cósmicos, seu fluxo, distribuição angular e energia. Foi ressaltado que para que se viabilize tal projeto seria necessário desenvolver um sistema robusto e autossuficiente para aquisição remota de dados, especialmente desenhado para as condições ambientais extremas do local. O projeto *CRE@AT* tem como objetivo estabelecer as correlações entre a presença do fluxo cósmico e a alteração de alguns parâmetros climáticos observados através dos sensores meteorológicos presentes no Criosfera I.

A Figura 6 ilustra o cronograma de três fases do projeto *CRE@AT*.

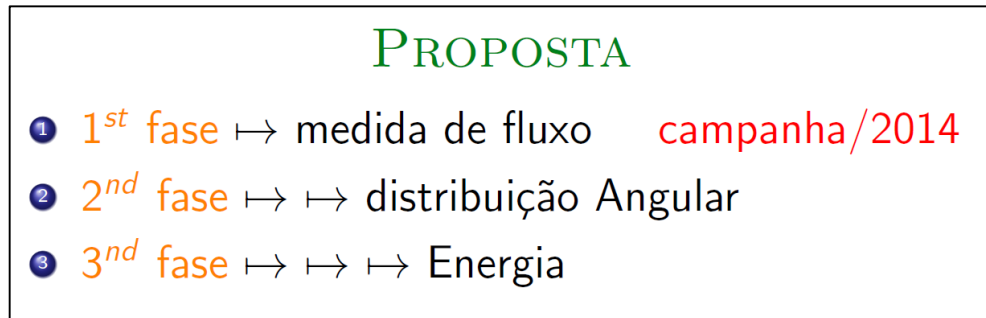


Figura 6 – Cronograma do projeto *CRE@AT*

O objetivo da primeira fase do experimento antártico de raios cósmicos é instalar no módulo Criosfera I um detector de fluxo cósmico baseado em coincidência de pulsos obtidos com tiras cintilantes. Este é um experimento piloto que abrirá caminho para a instalação futura do espectrômetro de raios cósmicos completo.

## 1.2. Objetivos

O objetivo deste trabalho é desenvolver um experimento de monitoração de fluxo da radiação cósmica galáctica para instalação no módulo Criosfera I, de modo a viabilizar a execução da primeira fase do projeto *CRE@AT*. Para tanto foi necessário desenvolver os sistemas de aquisição de dados e infraestrutura elétrica do experimento e incorporar um sistema detector de múons ao que foi desenvolvido.

## 1.3. Justificativa

Justamente pelo ineditismo deste estudo, não é possível encontrar no mercado um produto de instrumentação científica pronto capaz de viabilizá-lo. Por esta razão, fez-se necessário seu desenvolvimento por meio da integração de dispositivos pré-concebidos com módulos que, eventualmente, precisaram ser desenvolvidos do zero.

## 1.4. Metodologia e Trabalho Realizado

Para atingir os objetivos, concebeu-se uma plataforma autônoma de aquisição, processamento e transmissão em tempo real dos dados provenientes das detecções de partículas abrangendo, portanto, desde a captura do impulso elétrico oriundo da passagem das partículas por cintiladores plásticos, aquisição, digitalização, processamento, envio ao satélite e armazenagem local dos dados. Outros elementos do detector de partículas cósmicas – desenvolvidos pela equipe de altas energias do CBPF – foram posteriormente agregados a este trabalho.

Tendo em vista que o experimento ficará desassistido de pessoal local ao longo de todo o inverno antártico – sendo visitado apenas uma vez ao ano durante o verão – foram implementadas diversas rotinas automáticas de reinicialização e controle do mesmo. Foram projetados também algoritmos para tratamento de erros no caso de eventuais períodos de baixa disponibilidade de energia e/ou travamento do software.

A metodologia aplicada originou-se no estudo preliminar dos objetivos do experimento. Iniciado no final do ano de 2013, este estudo foi conduzido pelo aluno e pelo físico André Massafferri Rodrigues, pesquisador do Centro Brasileiro de Pesquisas Físicas (CBPF). Posteriormente, elaborou-se um cronograma de trabalho e o diagrama em blocos inicial do sistema, pilares fundamentais do trabalho posteriormente realizado. Organizou-se, portanto, o trabalho de modo a permitir o cumprimento do cronograma, o atingimento dos objetivos sob o ponto de vista científico e a posterior continuidade do experimento por outros grupos de pesquisa. Ao longo do ano de 2014 o experimento foi aos poucos sendo montado em bancada no laboratório de física experimental de altas energias (LAFEX) do CBPF e minuciosamente documentado de modo a permitir seu desmonte, transporte e posterior remontagem no módulo Criosfera I, continente antártico. Em 26 de novembro de 2014 o experimento foi transportado pela missão brasileira que saiu com destino ao Criosfera I. Em 28 de dezembro de 2014 o experimento entrou em operação.

Todos os recursos de hardware e software e facilidades necessárias à execução do projeto foram disponibilizados pelo CBPF, que disponibilizou:

- Laboratório Multiusuário de Física Experimental de Altas Energias (5º andar do edifício Cesar Lattes, Botafogo – Rio de Janeiro) com facilidades de:
  - » Instrumentação em geral: *VME*, *NIM*, osciloscópio, gerador de sinais e outros
  - » Espaço físico (bancadas, etc...)
  - » Pessoal de apoio
  - » Parceria para fabricação de Placa de Circuito Impresso no CBPF
- Instrumentação e sensores utilizados nos detectores:
  - » Fotomultiplicadora *MaPMT*
  - » Cintiladores plásticos e fibras ópticas *WLS*
  - » Caixa de polietileno preta para acondicionamento do detector
- Apoio do laboratório de mecânica
  - » Projeto e fabricação de componentes internos de fixação do detector
- Verbas diversas para aquisição de material de consumo
  - » Compra dos componentes eletrônicos necessários ao projeto
  - » Compra de miscelâneas
- Viabilização da Missão
  - » Cooperação interinstitucional para viabilizar a implementação do projeto

## 1.5. Organização do Trabalho

Este trabalho foi organizado de modo a propiciar ao leitor no Capítulo 2 (Contexto do Trabalho) um breve histórico introdutório da pesquisa de raios cósmicos e de sua interação com parâmetros climáticos. O programa antártico brasileiro e o módulo de pesquisas Criosfera I também são introduzidos neste capítulo com o objetivo de contextualizar o leitor no ambiente em que o experimento foi instalado. No Capítulo 3 (Desenvolvimento do projeto) cada um dos módulos nos quais o projeto foi dividido são apresentados, descritos e explicados. O Capítulo 4 (Resultados Obtidos) mostra os dados experimentais obtidos durante a fase de calibração, tanto em laboratório no Rio de Janeiro quanto em sua localização definitiva no módulo Criosfera I. Na sequência, o Capítulo 5 (Conclusão) faz um balanço do projeto e aponta as expectativas para as próximas fases do projeto *CRE@AT*. Seguem-se ainda sete apêndices e um anexo, onde são apresentados códigos e diagramas desenvolvidos e é descrito um trabalho auxiliar desenvolvido pelo aluno para contornar problemas com o controle de carga do banco de baterias do módulo Criosfera I durante a implementação do projeto.

## 2. Contexto do Trabalho

Sabe-se a algumas décadas que grande quantidade de partículas cósmicas subatômicas elementares constantemente atravessa e interage com nossa atmosfera. Ainda são preliminares, entretanto, os estudos acerca do processo microfísico por trás da formação dos núcleos de condensação de nuvens [7] e dos efeitos climáticos [8] e biológicos [9] consequentes da interação de partículas cósmicas com o ar atmosférico e com os seres vivos.

É sabido que grandes campos magnéticos oriundos da atividade solar exercem efeito repulsivo aos raios cósmicos que atingem a Terra e que existe, portanto, um efeito de modulação da atividade cósmica que é inversamente proporcional à atividade solar. Esta modulação provoca uma variação de aproximadamente 15% no fluxo médio da radiação cósmica observada em todo o planeta, sendo que a variação é de aproximadamente 5% em torno do equador e de 50% nos polos [8]. Tamanha variação justifica-se pela interação do fluxo cósmico com o campo magnético da própria Terra, que é ortogonal a seu eixo quando observado próximo aos polos e ao tamanho reduzido da atmosfera polar, aumentando consideravelmente o fluxo cósmico na região. Tais características fazem com que a região polar antártica seja de elevado interesse para os estudos sobre radiação cósmica.

Entretanto, esta é uma região extrema em todos os sentidos. Considerada a última fronteira na exploração do planeta, a região do círculo polar antártico (latitudes maiores que 80° Sul) apresenta as menores temperaturas do planeta. No inverno chegam-se a registrar temperaturas inferiores a -80°C em condições que não sustentam qualquer tipo de vida macrobiótica. A Antártica é considerada um deserto, sendo o único continente no qual não há população residente fixa [10]. Foi declarado em 1991 pelo protocolo de Madrid território internacional não-governado, e quaisquer atividades exploratórias e/ou militares foram proibidas destinando-se a presença humana apenas a fins pacíficos de pesquisa científica e lazer.

A instalação de qualquer experimento científico nesta região é extremamente complexa pois não existem quaisquer facilidades logísticas e operacionais permanentes de uso comum. Todo experimento deve ser então autossuficiente em termos energéticos, resistente às condições climáticas e capaz de operar longos períodos sem intervenção humana, pois não há acessibilidade durante o inverno polar.

## 2.1. Os Raios Cósmitos

Por volta de 1900 Charles Wilson, Julius Elster e Hans Geitel notaram [11] que a condutividade do ar contido num eletroscópio de folhas de ouro permanecia constante. O fenômeno foi então atribuído a radiação ionizante ambiental, relacionada com a recém-descoberta radioatividade. Entre 1911 e 1913 Victor Hess e Werner Kolhorster observaram que a partir dos 1000 metros de altitude a contagem de radiação aumentava fortemente. Victor Hess concluiu então que a ionização observada era devida à ação de uma radiação altamente penetrante provinda do espaço, recebendo o prêmio Nobel de Física em 1936 por essa descoberta.

Raios cósmicos são constituídos por partículas de alta energia – formadas por elementos carregados e por núcleos com meias-vidas superiores a  $10^6$  anos – que colidem com átomos da camada superior da atmosfera provocando chuviscos atmosféricos de diferentes dimensões, dependendo da energia das partículas iniciais. Grande parte destas partículas tem origem extragaláctica, sendo denominadas por “fontes primárias” são compostas por elétrons, prótons, hélio e por núcleos sintetizados por estrelas, como carbono, oxigênio e ferro. Outra parte é originada por fontes secundárias, compostas por certos núcleos, como lítio, berílio e boro que são gerados por interações entre fontes primárias e poeira interestelar. As partículas carregadas são afetadas pelo campo magnético terrestre e pela expansão do plasma magnetizado proveniente do sol – fenômeno conhecido como vento solar – que desaceleram e excluem parcialmente raios cósmicos galácticos de baixa energia. Estudos [8] indicam que existe uma anti-correlação entre a atividade solar, com ciclos de 11 anos, e a intensidade dos raios cósmicos.

Vários experimentos, como o Laboratório Pierre Auger [12] na Argentina, o experimento *IceCube* [13] no Polo Sul e o experimento *AMS-02 (Alpha Magnetic Spectrometer)* [14] realizado na estação espacial internacional, vem se dedicando a investigar a vasta gama de tópicos relacionados aos raios cósmicos, raios gama e neutrinos com o intuito de extrair informações acerca do Universo e de interações fundamentais da matéria num regime de energia ainda não alcançado pelos aceleradores de partículas. Segundo resultados recentes os raios cósmicos estão distribuídos isotopicamente devido a propagação difusa no campo magnético galáctico, o que dificulta enormemente a identificação de suas fontes. Neste contexto a pesquisa com radiação gama, apesar do fluxo reduzido em relação aos raios cósmicos, pode contribuir de forma significativa dada a ausência de deflexão magnética. Outras medidas, como



a fração relativa entre partículas e antipartículas na composição da radiação cósmica realizadas pelo experimento AMS, vêm recebendo destaque pois inserem elementos experimentais importantes acerca da elucidação da relação entre assimetria de matéria-antimatéria e a intensidade da radiação cósmica de fundo observada no Universo. Até agora nenhuma evidência de componentes primárias compostas por anti-próton, anti-hélio e anti-deutérium foi encontrada.

Com exceção dos prótons e elétrons, todas as demais partículas observadas em diferentes altitudes são produzidas pela interação dos raios cósmicos primários e secundários com a atmosfera. Mésons e neutrinos são produtos da cadeia de decaimentos de mésons carregados, enquanto elétrons e fótons são produtos de mésons neutros. Mésons são de longe [8] as partículas mais numerosas observadas abaixo de 6 km de altitude. A maior parte é produzida tipicamente a 15 km de altitude, e perdem aproximadamente 2 GeV por processos de ionização antes de atingir o solo. Sua energia e distribuição angular refletem a convolução entre o espectro de produção, a perda de energia na atmosfera e seu decaimento. Desta forma, os raios cósmicos são fortemente afetados pelo campo magnético, pela atividade solar, além da composição e dimensões atmosféricas.

## **2.2. O Programa Antártico Brasileiro**

O programa antártico brasileiro realiza pesquisas na região desde 1983 com a instalação na ilha Rei George da base Comandante Ferraz. A criação desta base proporcionou ao país tornar-se membro pleno de um grupo seleto de países responsáveis pelo futuro do continente antártico. O Programa é o resultado da soma de esforços de diversos órgãos do governo federal, envolvendo os ministérios de Defesa, das Relações Exteriores, do Meio Ambiente, da Ciência, Tecnologia e Inovação, das Minas Energias e da Educação. Para investigar aspectos relacionados às mudanças ambientais globais, uma rede de pesquisa foi formada por 12 equipes de pesquisadores, reunidas em sete grupos temáticos, formado por 16 instituições brasileiras. São realizados estudos integrados da atmosfera, do gelo, do solo e do oceano, estudos da composição atmosférica antiga através da análise de testemunhos de gelo retirados de grandes profundidades, além do monitoramento de parâmetros físicos, químicos e biológicos.

## 2.3. O Módulo Criosfera 1

O módulo avançado de pesquisa científica Criosfera 1, localizado na latitude 84°S (interior do continente antártico, a 640 km do polo sul geográfico como visto na Figura 7) foi inaugurado em 12 de janeiro de 2012 e é resultado dos mais de 30 anos de pesquisa e aprendizado do programa antártico brasileiro. Fruto de iniciativa do ministério de Ciência, Tecnologia e Inovação e do CNPq o módulo é a primeira (e, na atual data, única) iniciativa de pesquisa brasileira a operar de forma contínua, autônoma e remota no interior do continente antártico.



Figura 7 – Localização do módulo Criosfera I

Sua concepção estrutural é o estado da arte dos módulos de pesquisa polar, contando com espaço interno de 6,3 x 2,5 x 2,6 [m<sup>3</sup>] e um sistema híbrido eólico/solar autossuficiente para geração de energia elétrica que permitem mantê-lo em funcionamento ininterrupto, inclusive durante o inverno polar. Tem como objetivo estabelecer-se como plataforma de pesquisa multiusuária dado seu grande potencial para estudos voltados para biotecnologia, meteorologia, paleoclima, física, química da atmosfera, e astrofísica de altas energias, oferecendo condições de infraestrutura a pesquisa com o mínimo de alteração do ecossistema. As Figuras 8 e 9 mostram, respectivamente, um pequeno laboratório montado dentro do módulo Criosfera I e a face externa sul no momento em que recebeu nosso experimento.



Figura 8 – Laboratório do módulo Criosfera I



Figura 9 – Face Sul do módulo Criosfera I

Durante as missões de 2011, 2012 e 2013 foram instalados experimentos voltados para o estudo da atmosfera antártica e sua relação com o manto de gelo. Neste estudo são monitorados aerossóis de origem terrígena, antropogênica, cósmica e biogênica. O módulo é dotado de uma estação meteorológica que monitora temperatura do ar, pressão atmosférica, umidade relativa, intensidade e direção do vento e intensidade da radiação solar. Um sistema

ultrassônico, instalado no exterior do módulo, mede a dinâmica de deposição de neve na região em tempo real. Todos os dados meteorológicos, das concentrações de CO<sub>2</sub>, da deposição de neve, entre outros, são enviados via satélite em tempo real para o Brasil. Estes dados – reunidos e interpretados a luz de modelos computacionais de transporte atmosférico – permitem aumentar nossa compreensão sobre a relação climática Antártica-América do Sul, o impacto da redução da camada de ozônio, da atividade vulcânica no hemisfério sul, da evolução dos processos globais de desertificação, do transporte atmosférico global de poluentes e de microrganismos, bem como aprofundar nosso conhecimento sobre a história climática contada pelos testemunhos de gelo.

### 3. Desenvolvimento do projeto

O desenvolvimento de instrumentação para física de altas energias não é inédito. Entretanto o desenvolvimento de um experimento capaz de operar na Criosfera antártica demanda instrumentação capaz de operar de forma autônoma em condições muito reduzidas de suprimento energético. O layout básico desenhado para o projeto é visto na Figura 10.

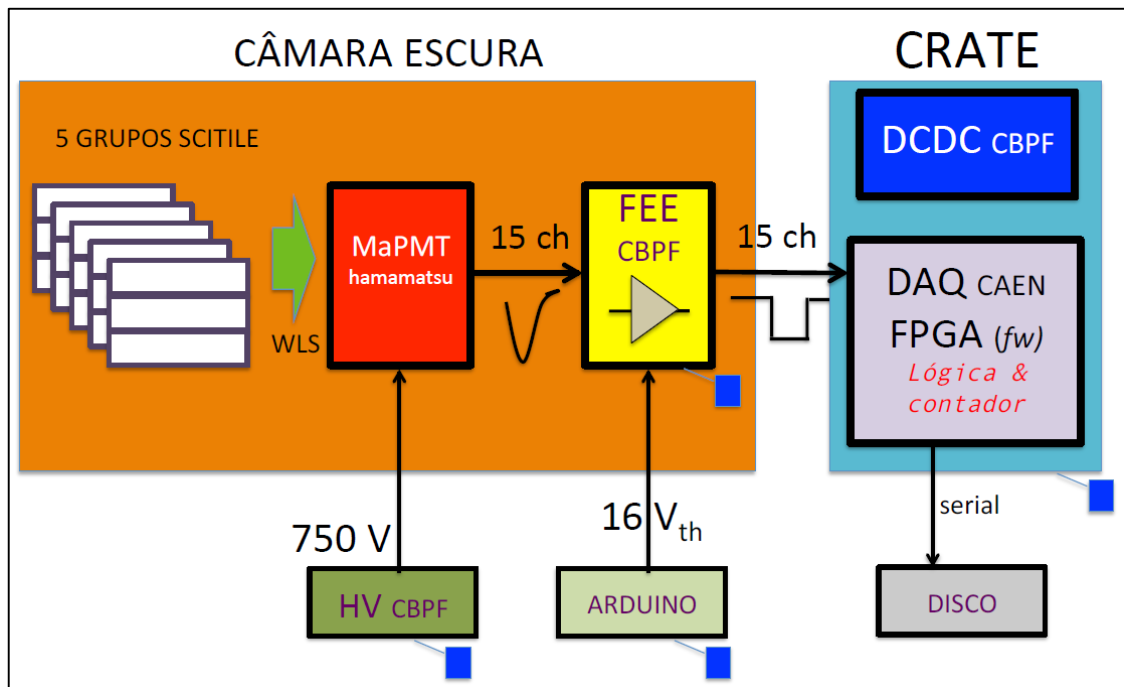


Figura 10 – Layout básico do experimento

Neste layout observamos que a área sensível é composta por cinco grupos cintiladores *SciTile* (3.1.1.1) conectados por meio de fibras ópticas *WLS* (3.1.1.2) a uma fotomultiplicadora multi anodo (3.1.1.3). O impulso elétrico gerado é então tratado pelo circuito eletrônico *Front-End* (3.1.1.4), analisado pelo circuito contador de coincidências numa *FPGA* (3.2.1) e armazenado em mídia não volátil (3.2.2). Todo o conjunto ótico necessita ser acondicionado em câmara escura pois é sensível à luz ambiente, e todos os dispositivos eletrônicos necessitam de alimentação elétrica, fornecida pelo conversor *DCDC* em azul (3.2.4).

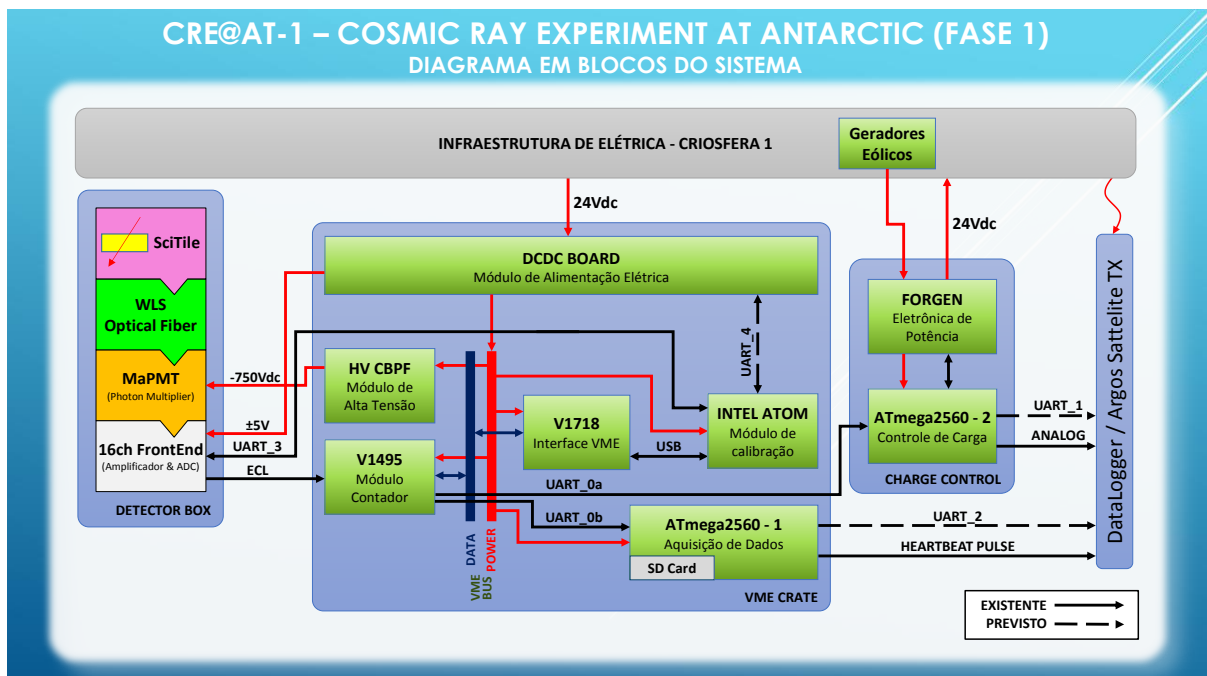
Sabendo desde o início que a missão para implementação deste projeto seria extremamente crítica, optou-se por sua montagem em módulos desmontáveis que pudessem ser apropriadamente acondicionados para transporte e facilmente remontáveis. Conceberam-se, então, nove módulos subdivididos em três grandes grupos:

- Detecção
  - » Realiza a detecção de passagem de partículas oriundas da radiação cósmica galáctica por múltiplos planos cintiladores, geração de pulsos elétricos de baixa impedância correspondentes às passagens de partículas pelos planos de detecção.
- Aquisição
  - » Realiza aquisição de eventos com base em coincidências de pulsos, contagem destes eventos por unidade de tempo, gravação em mídia não-volátil dos dados brutos, cálculo dos parâmetros de correção de eficiência, produção de sequência de dados a transmitir por satélite e calibração do detector.
- Infraestrutura
  - » Provê alimentação elétrica regulada de baixa tensão para o experimento, alimentação elétrica de alta tensão para fotomultiplicadora e realiza o controle dos geradores eólicos de energia.

Dois destes módulos foram incorporados a este projeto e, embora o compoñham, não foram desenvolvidos nele. São eles o módulo detector e o módulo de alta tensão, legados de trabalhos anteriores desenvolvidos pelo grupo de altas energias do CBPF. Outro módulo utilizado, de interface ao barramento *VME*, foi adquirido no mercado já como solução final. Coube, portanto, a definição do layout do experimento e o desenvolvimento dos módulos de contagem, de calibração, de aquisição de dados, de alimentação elétrica e de controle de carga. A Tabela 1 resume esta organização. O diagrama em blocos completo da primeira fase do projeto *CRE@AT* pode ser visto na Figura 11.

Tabela 1 – Módulos da fase 1 do experimento antártico de raios cósmicos

Item	Origem	Módulo	Grupo
3.1.1	Incorporado	Detector	Detecção
3.2.1	Desenvolvido	Contador de Coincências	Aquisição
3.2.2	Desenvolvido	Aquisição de Dados	
3.1.3	Incorporado	Interface VME	
3.2.3	Desenvolvido	Calibração	
3.2.4	Desenvolvido	Alimentação Elétrica	Infraestrutura
3.1.2	Incorporado	Alta Tensão	
3.2.5	Desenvolvido	Controlador de Carga	

Figura 11 – Diagrama em blocos do sistema *CRE@AT*

### 3.1. Módulos Incorporados

Nesta seção serão apresentados os módulos desenvolvidos por outras equipes e incorporados a este projeto. A Figura 12 os destaca no diagrama em blocos do experimento *CRE@AT*.

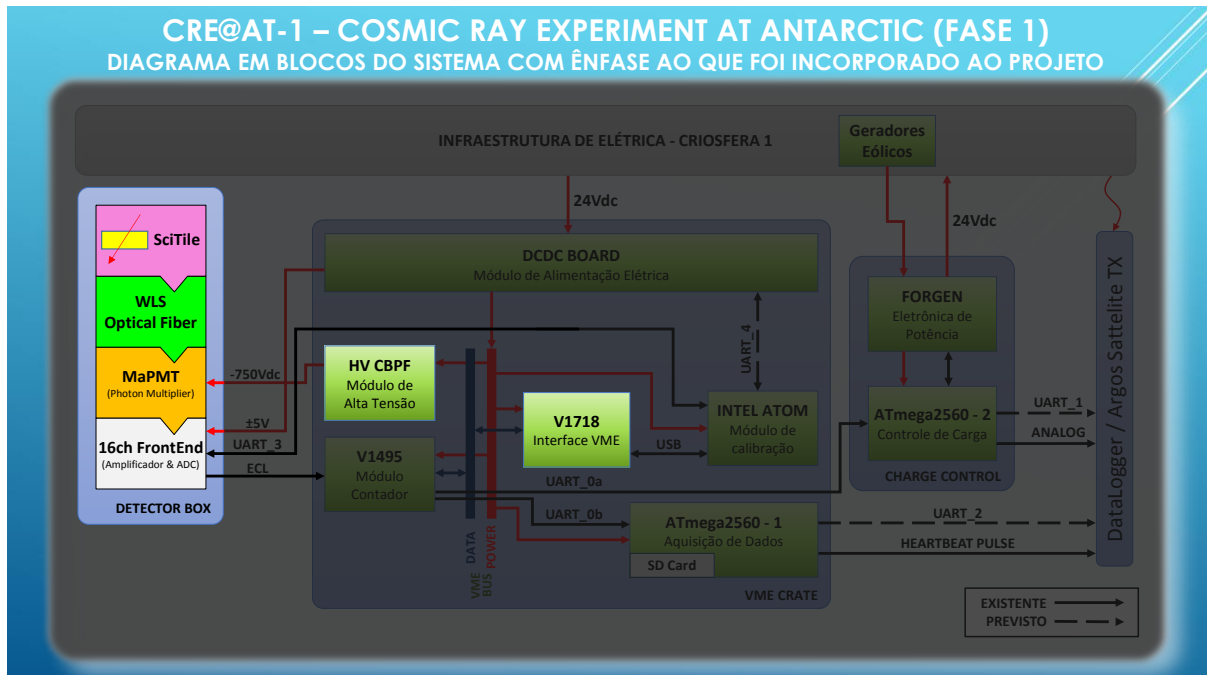


Figura 12 – Diagrama em blocos com ênfase ao que foi incorporado ao projeto

### 3.1.1. Módulo Detector

Este módulo é responsável pela geração de pulsos elétricos correspondentes à passagem por elementos cintilantes de múons oriundos dos raios cósmicos galácticos.

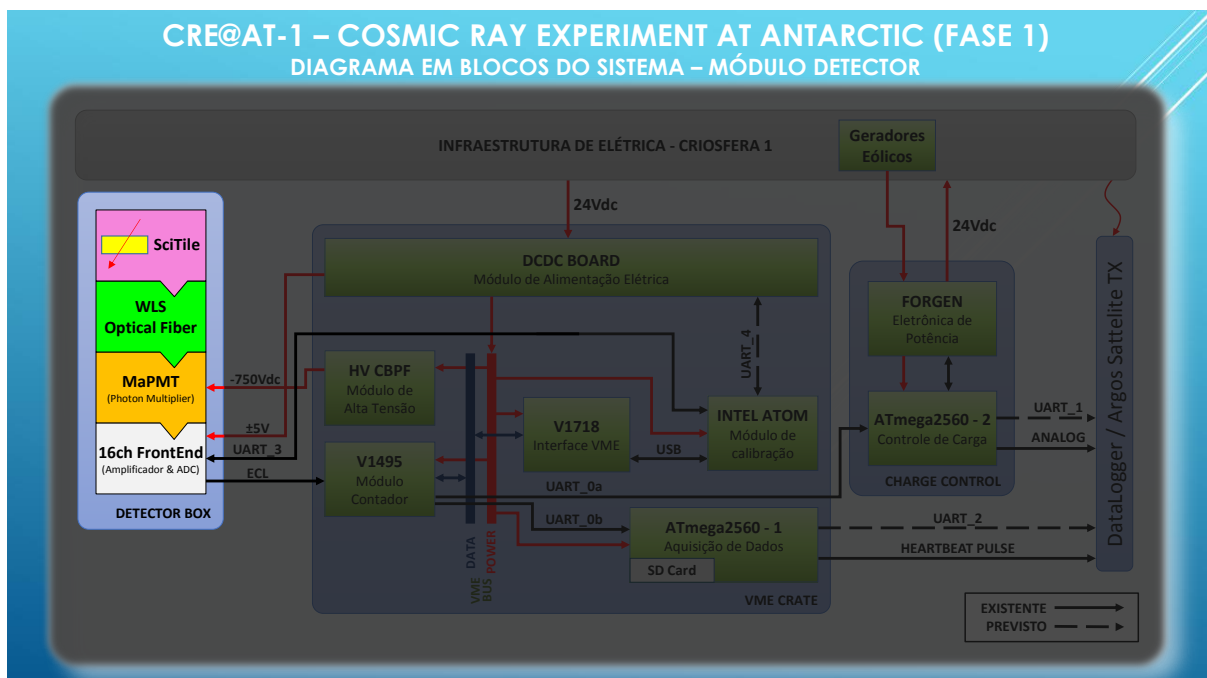


Figura 13 – Diagrama em blocos com destaque ao detector



Na Figura 13 o módulo detector encontra-se destacado no diagrama em blocos do projeto *CRE@AT*. A detecção de partículas é resultado de diversas transformações de energia, tal como pode ser visto na Figura 14.

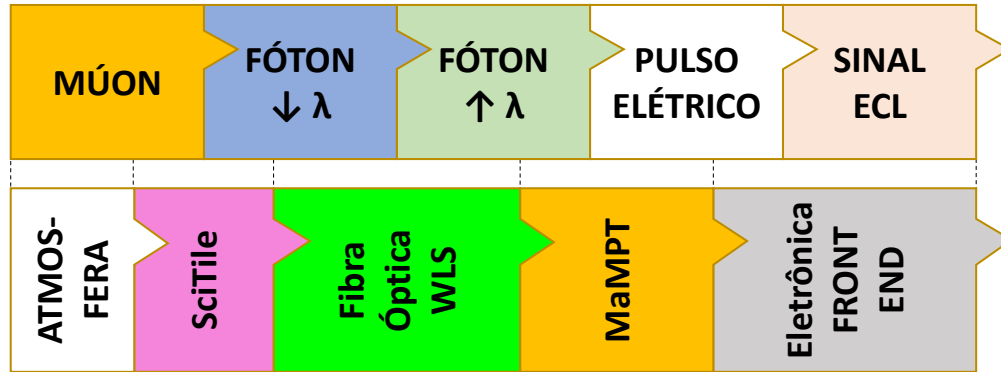


Figura 14 – Transformações de energia no detector

Primeiramente, múons originados da radiação cósmica galáctica atravessam uma tira de plástico cintilante (*SciTile*) provocando a emissão de fótons. Uma fibra óptica fluorescente instalada ao longo do meio cintilante captura os fótons ali presentes e os transforma em fótons de maior comprimento, conduzindo-os até um dispositivo foto multiplicador (*MaPMT*). Este, quando estimulado, produz um pulso elétrico que é amplificado, discriminado e transmitido ao módulo contador de eventos coincidentes.

O conjunto detector foi concebido de modo a agrupar quinze tiras cintilantes em cinco grupos de coincidência, sendo cada grupo composto por três planos de coincidência. A Figura 15 representa em seção transversal esta montagem.

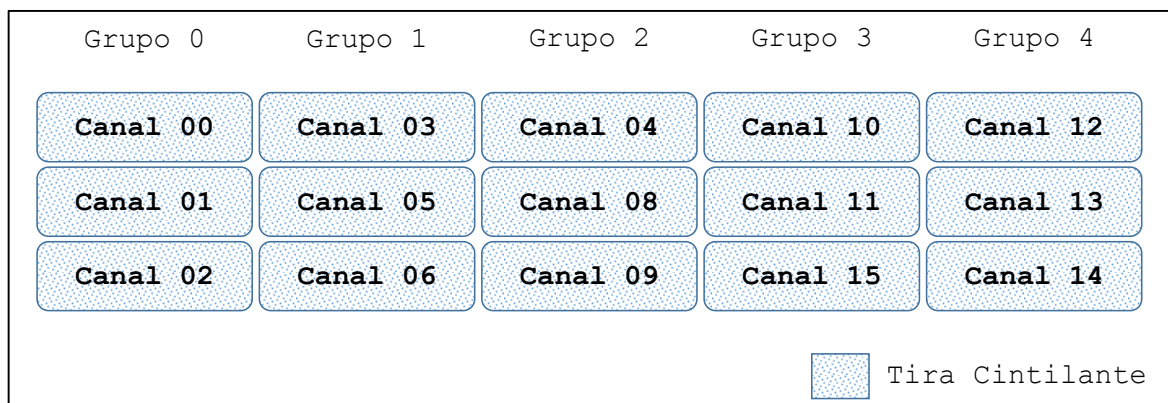


Figura 15 – Layout de montagem das tiras cintilantes

Cinco grupos resultam na tomada de cinco medidas independentes, permitindo que sejam avaliados efeitos sistemáticos sobre o experimento e comprovando sua estabilidade. A inclusão de três planos em cada grupo, como ilustrado na Figura 16, permite que sejam aplicadas técnicas de correção de fluxo por coincidência de eventos.

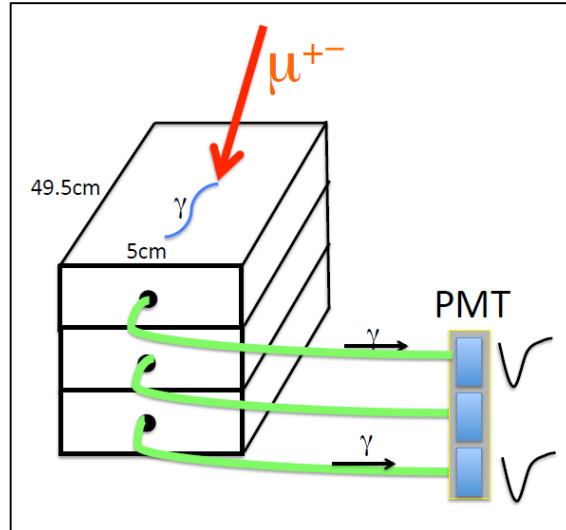


Figura 16 – *Layout* dos grupos de coincidência

A coincidência de pulsos é um método efetivo para selecionar eventos de passagem de partículas, pois o ruído eletrônico associado ao sistema é incoerente e sua ocorrência simultânea em múltiplos canais é estatisticamente pouco provável. O *layout* do detector em planos de coincidência tripla permite ainda a tomada instantânea da medida de eficiência de cada um de seus canais, aplicando a seguinte fórmula:

$$Eff(Canal_x) = \frac{Coincidências\ Triplas\ (todos\ os\ canais\ do\ grupo)}{Coincidências\ Duplas\ (outros\ canais\ do\ grupo)}$$

A correção da medida de fluxo pela eficiência relativa de cada canal mitiga problemas causados pela passagem de partículas que eventualmente tenham deixado de excitar apenas um dos canais de determinado grupo.

A Figura 17 mostra a montagem final do módulo detector de partículas. Nela podemos observar, da esquerda para a direita, as tiras plásticas cintilantes (*SciTiles*), as fibras ópticas (*WLS*), a foto multiplicadora (*MaPMT*) e o circuito eletrônico de *Front End*. A Figura 18 detalha as fibras cintilantes, a foto multiplicadora e o circuito eletrônico de *Front End*.



Figura 17 – Detector de partículas cósmicas

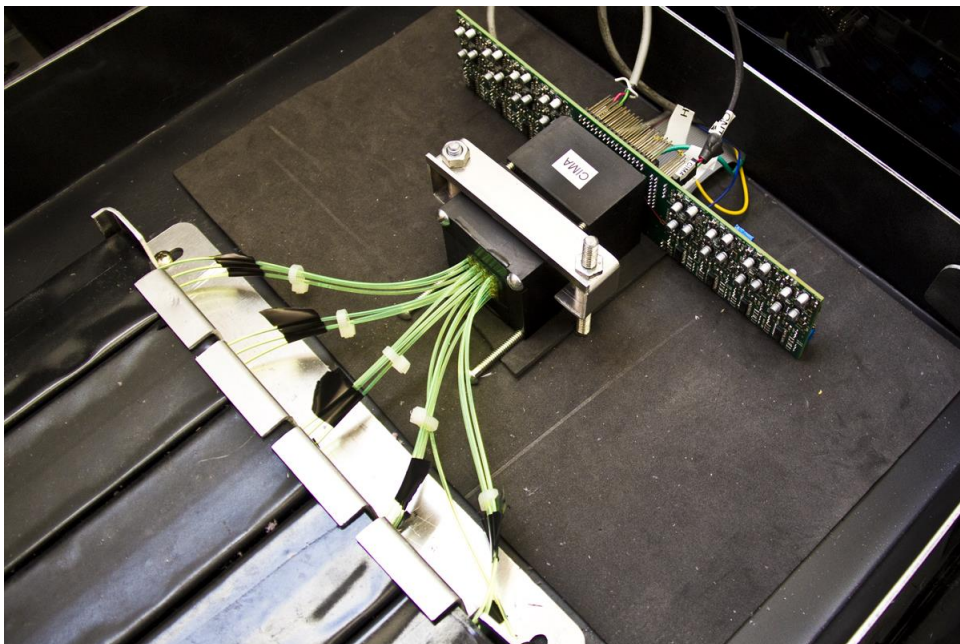


Figura 18 – Detalhe do detector

Cada tira cintilante possui uma fibra óptica *WLS* inserida através de seu núcleo e cada uma destas fibras é acoplada à foto multiplicadora em uma de suas extremidades. A máscara usinada em latão que pode ser vista na Figura 19 é utilizada para garantir o alinhamento deste acoplamento.

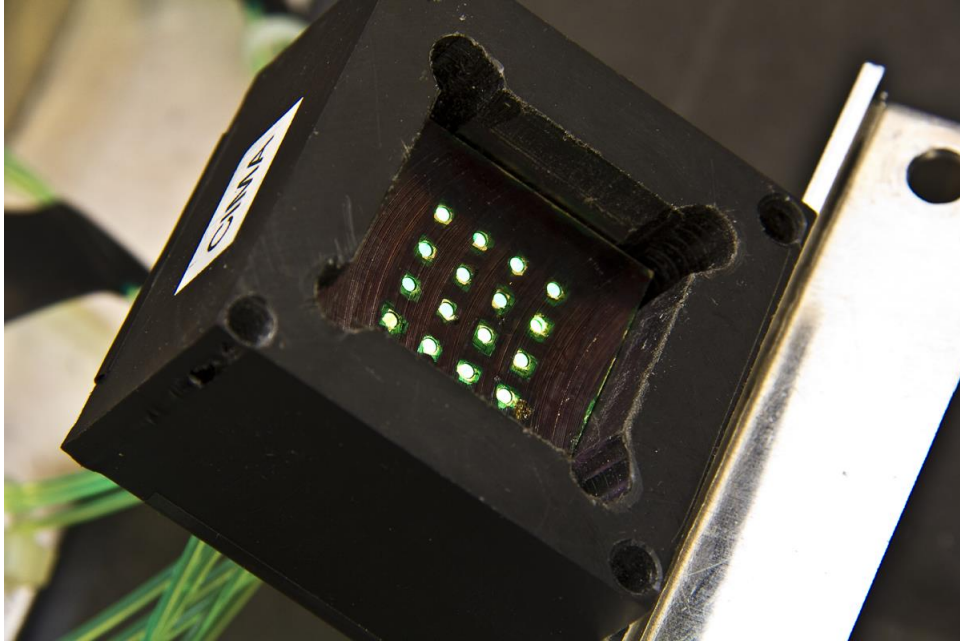


Figura 19 – Máscara de acoplamento óptico entre as fibras WLS e a foto multiplicadora

O acabamento preto fosco aplicado à face interna da máscara tem como objetivo reduzir a reflexão na superfície de vidro da *PMT*. O alinhamento entre a superfície da foto multiplicadora e a seção transversal da fibra óptica é extremamente crítico, e está diretamente relacionado aos parâmetros de eficiência do detector. Uma cola especial foi utilizada para garantir que o alinhamento se mantivesse durante a montagem do conjunto. Na Figura 20, observa-se que a fibra óptica indicada pela seta vermelha não está bem alinhada. Esta fibra, entretanto, não está associada a nenhuma das quinze tiras cintilantes.



Figura 20 – Alinhamento da máscara de fibras ópticas

### 3.1.1.1. Tiras plásticas cintilantes (*SciTiles*)

A emissão de luz como consequência da absorção de energia é propriedade presente em ampla gama de materiais, denominados cintiladores. Parte da energia depositada por partículas que os atravessam é absorvida pelos elétrons de seus átomos, o que os leva a níveis orbitais mais energéticos, porém, instáveis. O retorno destes elétrons a seus níveis originais ocorre na forma da emissão de um fóton.

Espera-se que um material cintilador utilizado em detecção de partículas [15] possua as seguintes propriedades:

- A quantidade de fótons gerados por cintilação deve ser proporcional a energia absorvida;
- Deve ser capaz de converter, com alta eficiência quântica, a energia cinética das partículas carregadas que o atravessam;
- Deve ser transparente ao comprimento de onda do fóton emitido, de modo que a luz possa ser transmitida a outros dispositivos;
- Para que seja viável a contagem de partículas sem grandes atrasos o processo de emissão deve ter curta duração;
- O material deve possuir boa qualidade óptica e ser viável para produção nas dimensões necessárias em um detector;

As tiras cintilantes utilizadas neste projeto foram desenvolvidas pelo *Fermilab* (*Fermi National Accelerator Laboratory*) e contém um orifício para que seja alocada a fibra óptica *WLS*. Possuem uma casca reflexiva que faz com que os fótons gerados sejam, com alta probabilidade, conduzidos até a fibra óptica *WLS* e são opacas para absorver luz no espectro visível. A construção da tira e a ilustração deste processo podem ser vistos na Figura 21, onde as unidades encontram-se em milímetros.

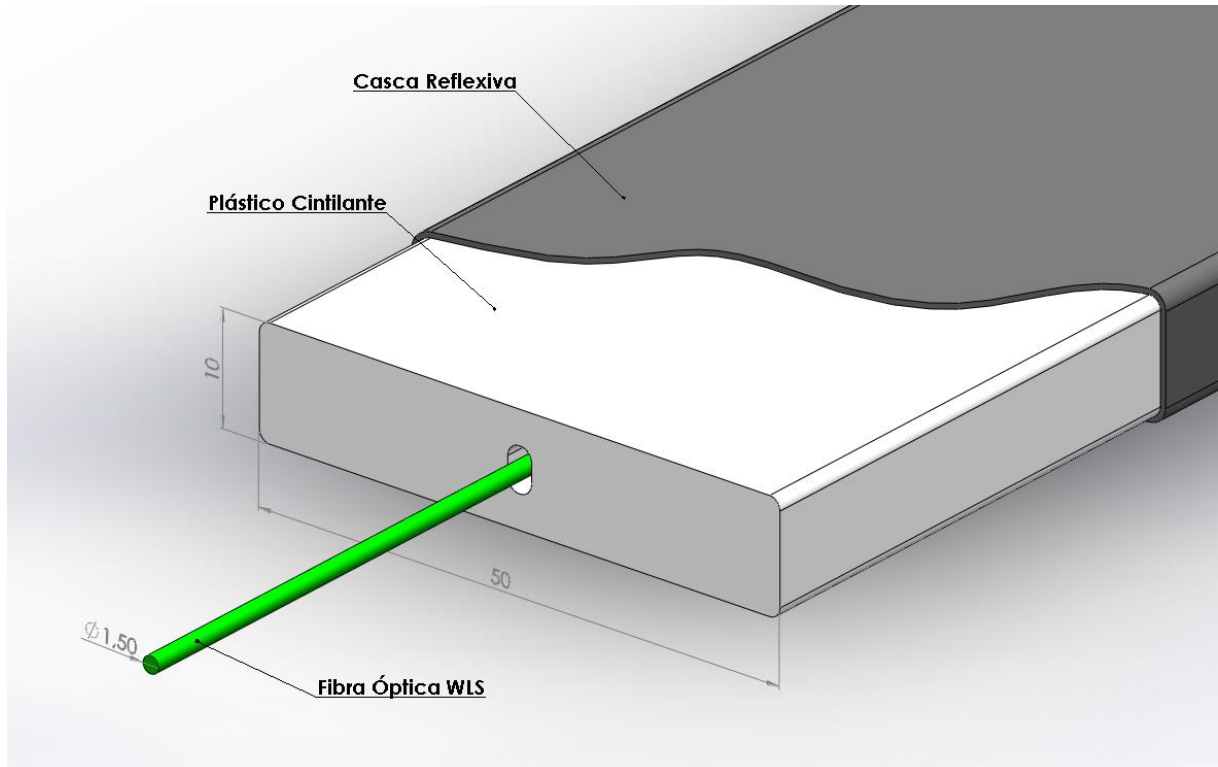


Figura 21 – Montagem mecânica da tira cintilante com fibra *wavelength shifter*

### 3.1.1.2. Fibra óptica *Wavelength Shifter*

As fibras ópticas *wavelength shifter* (WLS) modelo Y-11(175)MSJ utilizadas neste projeto são produzidas pela *KURARAY* com material fluorescente, capaz de absorver os fótons produzidos em frequências mais altas pela tira cintilante e reemitir-los em frequência mais baixa, numa região espectral onde há maior rendimento da foto multiplicadora. O gráfico da Figura 22 relaciona a função de transferência desta transformação de frequência. Além disto, a absorção de um fóton de energia mais alta pelo material da fibra óptica emite múltiplos fótons de energia mais baixa, ajudando também a aumentar a eficiência do sistema de detecção de partículas. A Figura 23 exibe parte das fibras ópticas utilizadas neste projeto.

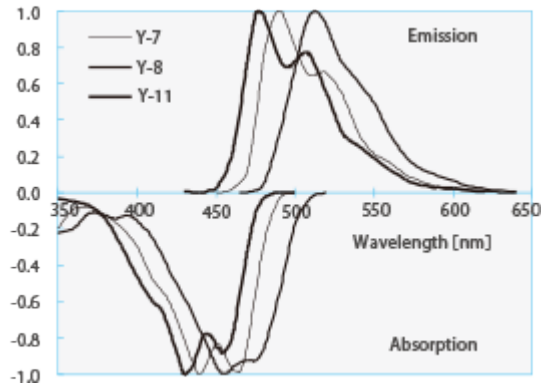


Figura 22 – Espectro de absorção e emissão - *KURARAY WLS Fibers* [16]

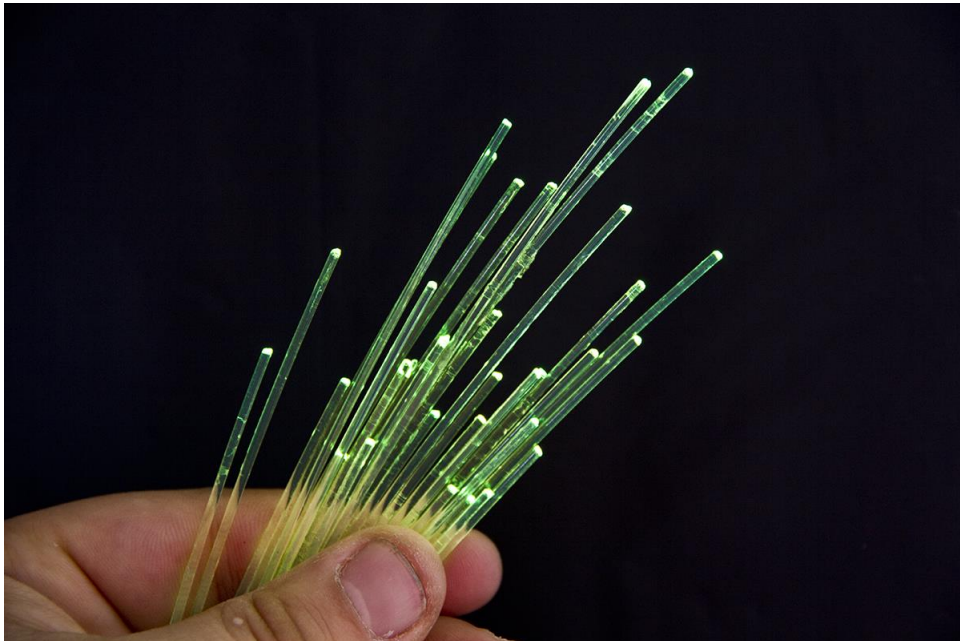


Figura 23 – Fibras ópticas *WLS*

### 3.1.1.3. Fotomultiplicadora *MaPMT*

Uma *MaPMT* (*Multianode Photomultiplier*) é um dispositivo transdutor foto sensível de múltiplos canais que possui a característica básica de gerar pulsos elétricos correspondentes à detecção de fótons em suas células. A foto multiplicação é o fenômeno de detecção de fótons através da avalanche de elétrons produzida devido a aceleração dos mesmos em múltiplos campos elétricos, como pode ser visto na Figura 24.

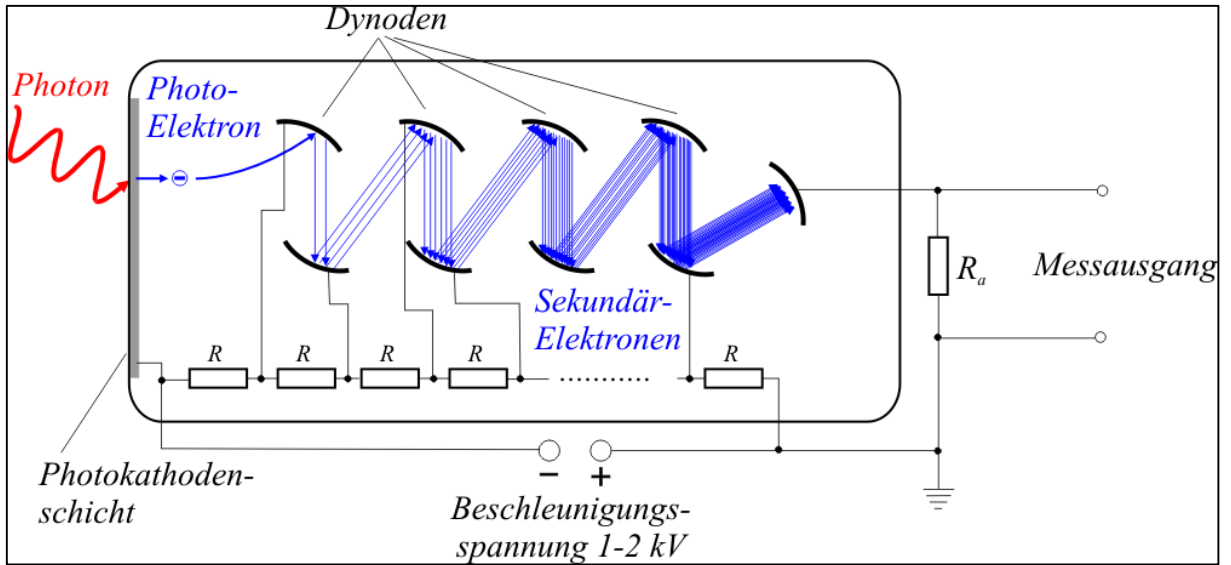


Figura 24 – Aplicação de Alta Tensão numa fotomultiplicadora [17]

É necessário, portanto, produzir campo elétrico suficiente para que haja aceleração dos elétrons e conseqüente foto multiplicação. A *MaPMT* HAMAMATSU H6568MOD vista na Figura 25 foi utilizada neste trabalho. Ela possui dezesseis anodos e é necessário aplicar alta tensão, 750 volts negativos em relação à massa, em seu catodo.



Figura 25 – *MaPMT* HAMAMATSU H6568MOD

O valor de pico do pulso de corrente gerado na saída de cada um dos canais desta *MaPMT* é especificado entre  $-150\mu\text{A}$  e  $-300\mu\text{A}$ , apresentando valor típico de  $-200\mu\text{A}$ . Quando injetado num circuito amplificador com impedância de entrada de 50 ohms obtemos então um pulso com tensão de pico de aproximadamente  $-10\text{mV}$ .



### 3.1.1.4. Eletrônica *Front-End*

Os pulsos oriundos da *MaPMT* apresentam amplitude e largura muito pequenos, tornando inviável sua transmissão direta ao módulo contador de eventos. Montado junto à foto multiplicadora foi introduzido então o circuito eletrônico amplificador e discriminador *ECL* de dezesseis canais produzido no projeto de fim de curso do aluno Leonardo Chaves Ruiz Guedes, do curso de Engenharia Eletrônica do CEFET/RJ, em 2013 [18]. O diagrama em blocos deste circuito pode ser visto na Figura 26. Algumas modificações no estágio de alimentação deste circuito se fizeram necessárias devido as restrições do presente projeto.

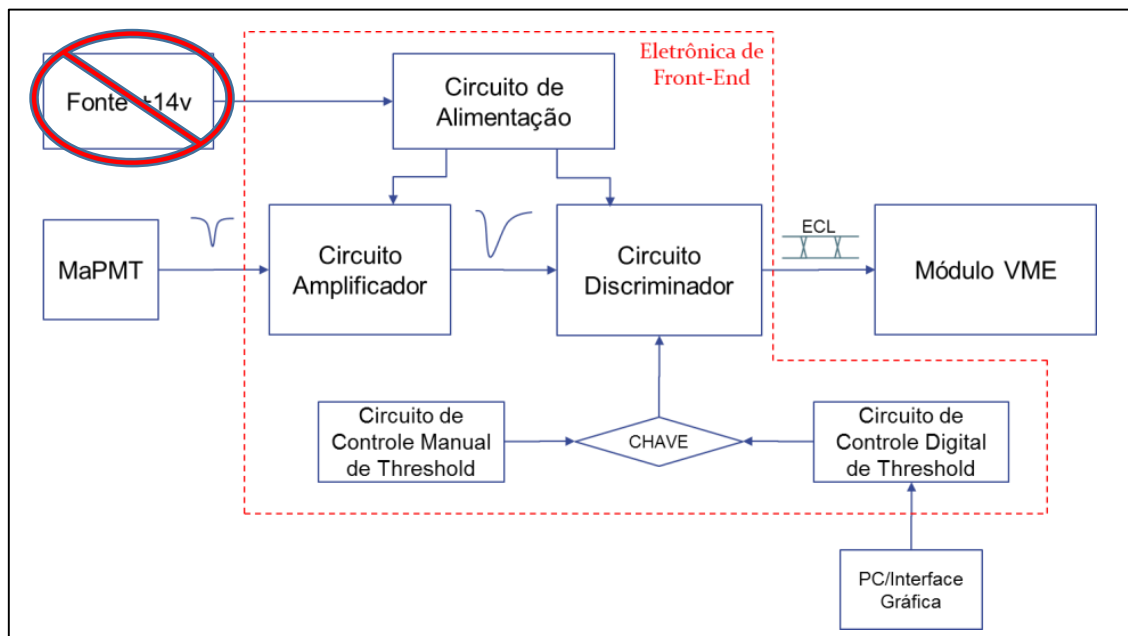


Figura 26 – Diagrama em blocos da *Front-End* [18]

Seu estágio amplificador apresenta ganho de aproximadamente 13 V/V e é dotado de filtro passa faixa integrado para eliminar o nível DC do sinal e aumentar de 2 ns para aproximadamente 20 ns a largura do pulso. O diagrama elétrico do estágio amplificador pode ser visto na Figura 27.

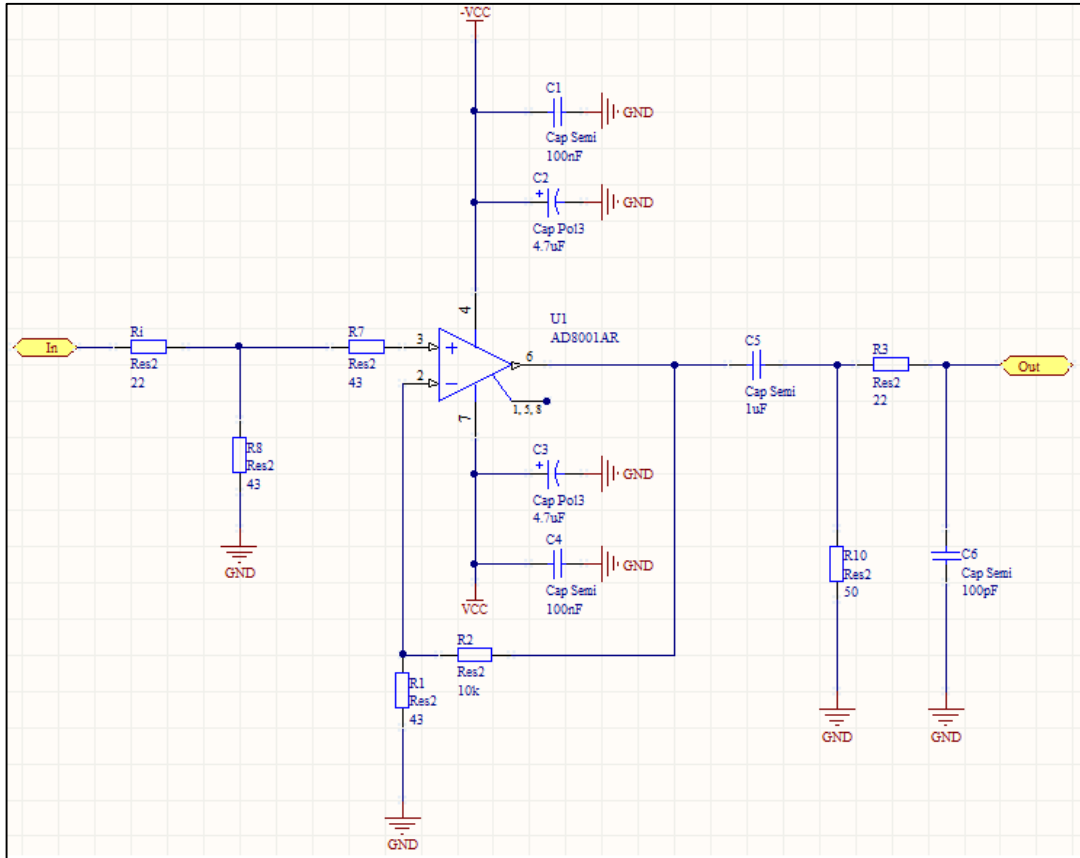


Figura 27 – Estágio amplificador do módulo de *front-end* [18]

Quando o nível do sinal amplificado ultrapassa uma tensão de referência (*threshold*) entende-se que houve um evento e é gerado um pulso digital diferencial *ECL* na saída do discriminador. Este é composto basicamente por um comparador de alta velocidade e baixa tensão capaz de operar até 800 MHz, possibilitando assim a leitura de pulsos com larguras a partir de 625 ps. O diagrama elétrico do estágio discriminador pode ser visto na Figura 28.

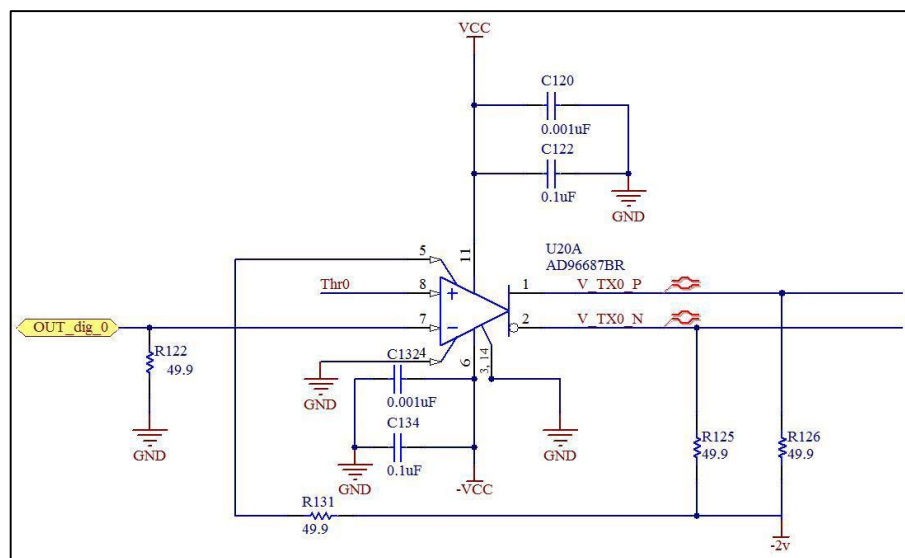


Figura 28 – Estágio discriminador do módulo *front-end* [18]

O *FPGA (Field Programmable Gate Array)* que compõe a módulo contador de eventos coincidentes é sensível a pulsos da ordem de 5 ns, tornando a transmissão dos pulsos oriundos da *Front-End* potencialmente vulnerável a ruídos eletromagnéticos. Para contornar tal vulnerabilidade optou-se pela transmissão diferencial destes sinais baseada no padrão *ECL*. Este padrão utiliza ainda baixos níveis de tensão com o objetivo de minimizar os efeitos capacitivos da linha de transmissão na propagação do sinal e, conseqüentemente, possibilitar um acréscimo na resposta em frequência da linha.

### 3.1.2. Módulo de Alta Tensão

A Figura 29 destaca a posição de alta tensão no diagrama em blocos do sistema *CRE@AT*.

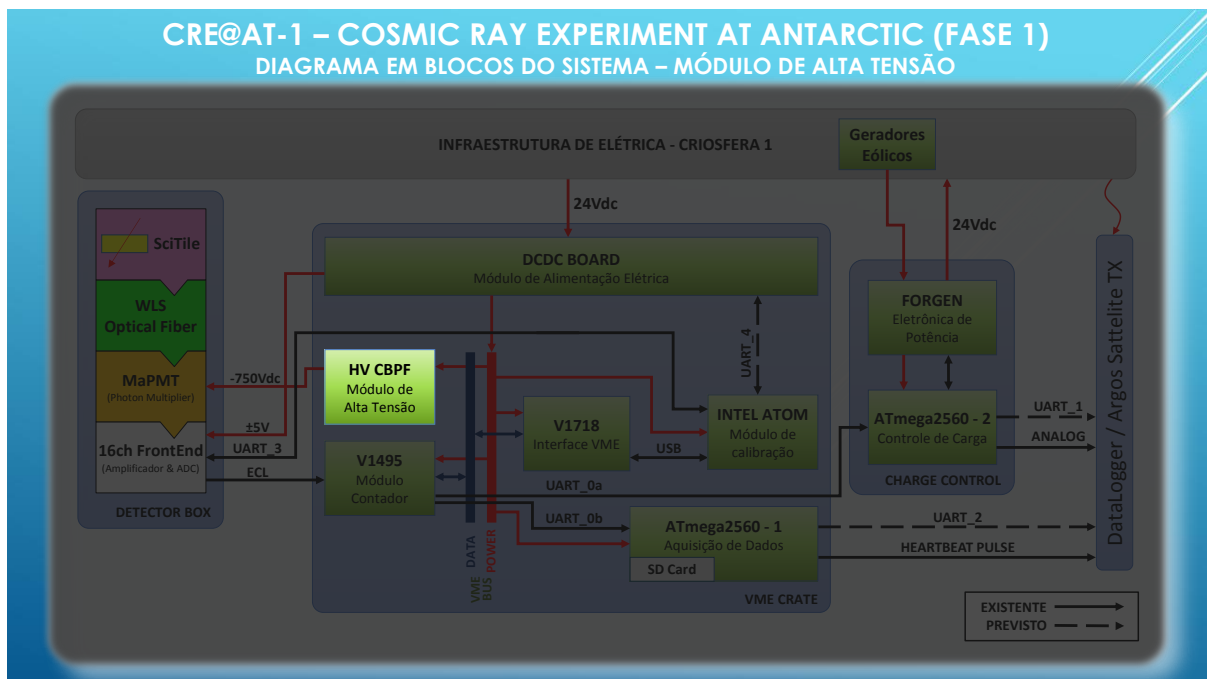


Figura 29 – Diagrama em blocos com destaque ao módulo de alta tensão

Foi utilizada uma fonte de alta tensão produzida alguns anos atrás pelo grupo de altas energias do CBPF que é alimentada por 12 volts em corrente contínua e possui saída de alta tensão negativa regulável de zero a mil volts.

### 3.1.3. Módulo de Interface ao Barramento *VMEbus*

A tecnologia *VMEbus* (*Versa Module Europa bus*) é um padrão de barramento de computador, originalmente desenvolvido pela Motorola para a linha 68000 de *CPUs*, porém foi largamente utilizada para muitos outros propósitos [19] incluindo várias aplicações em instrumentação para pesquisa em física de altas energias. Este é um barramento essencialmente paralelo com 32 bits para endereços e outros 32 bits para dados que permitem a comunicação entre um ou mais dispositivos *MASTER* e um ou mais dispositivos *SLAVE*. Obrigatoriamente também um dispositivo *MASTER* deve assumir a função de arbitragem do barramento (definida por sua inserção no primeiro *SLOT* do *BACKPLANE*).

O módulo *CAEN V1718*, utilizado em nosso experimento, é um dispositivo *VME MASTER* capaz de realizar arbitragem do barramento e que possui uma interface *USB* para permitir a troca de dados entre um computador do módulo de calibração e o barramento *VME*.

## 3.2. Módulos Desenvolvidos

Nesta seção serão apresentados os módulos inteiramente desenvolvidos neste projeto. A Figura 30 os destaca no diagrama em blocos do experimento *CRE@AT*.

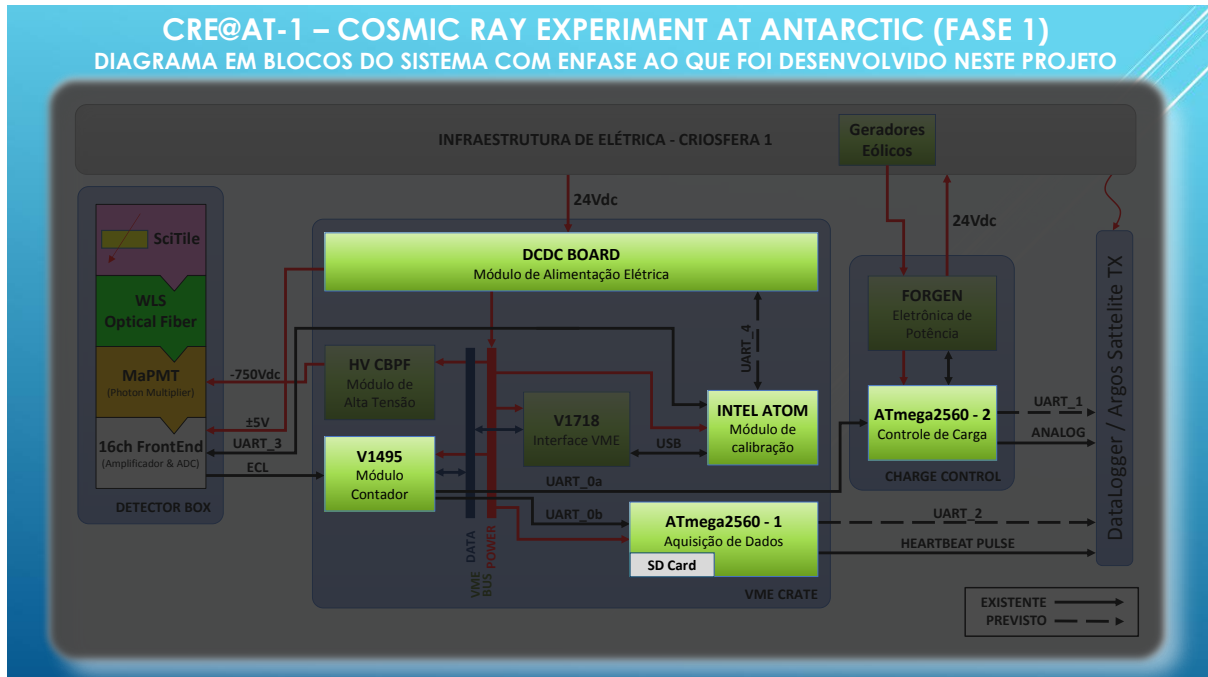


Figura 30 – Diagrama em blocos com ênfase ao que foi desenvolvido neste projeto

### 3.2.1. Módulo Contador de Eventos Coincidentes

Os pulsos *ECL* gerados em alta velocidade pelo detector devem ser contados de modo a obtermos dados de interesse científico que possibilitem análise do fluxo da radiação cósmica galáctica. O módulo contador de eventos coincidentes – destacado no diagrama em blocos da Figura 31 – é responsável por esta tarefa.

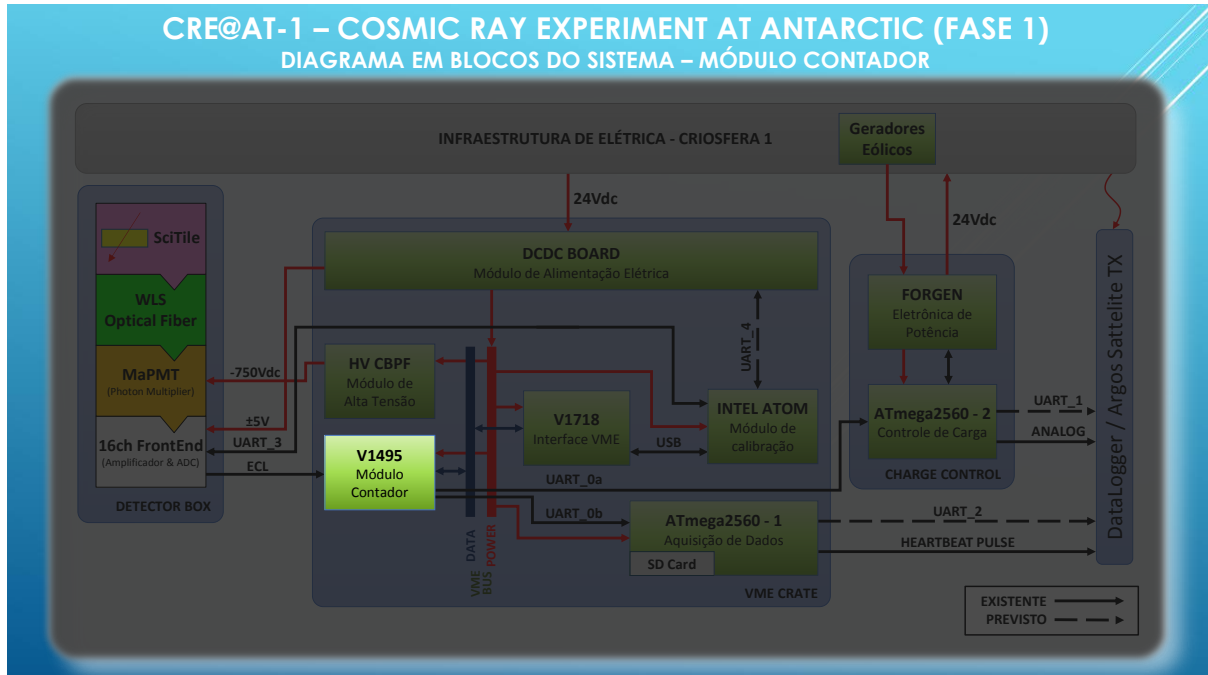


Figura 31 – Diagrama em blocos com destaque ao módulo contador

Tomando como base a geometria de construção do detector, ou seja, cinco grupos de três elementos cintiladores cada, o projeto deste módulo levou em consideração a necessidade de registro da taxa de ocorrência de fenômenos de coincidência dupla entre canais de um mesmo grupo, a taxa de ocorrência de coincidência tripla (ou total) num mesmo grupo e o registro dos eventos individuais em cada um dos elementos para caracterização de ruído. O diagrama em blocos do módulo é visto na Figura 32.

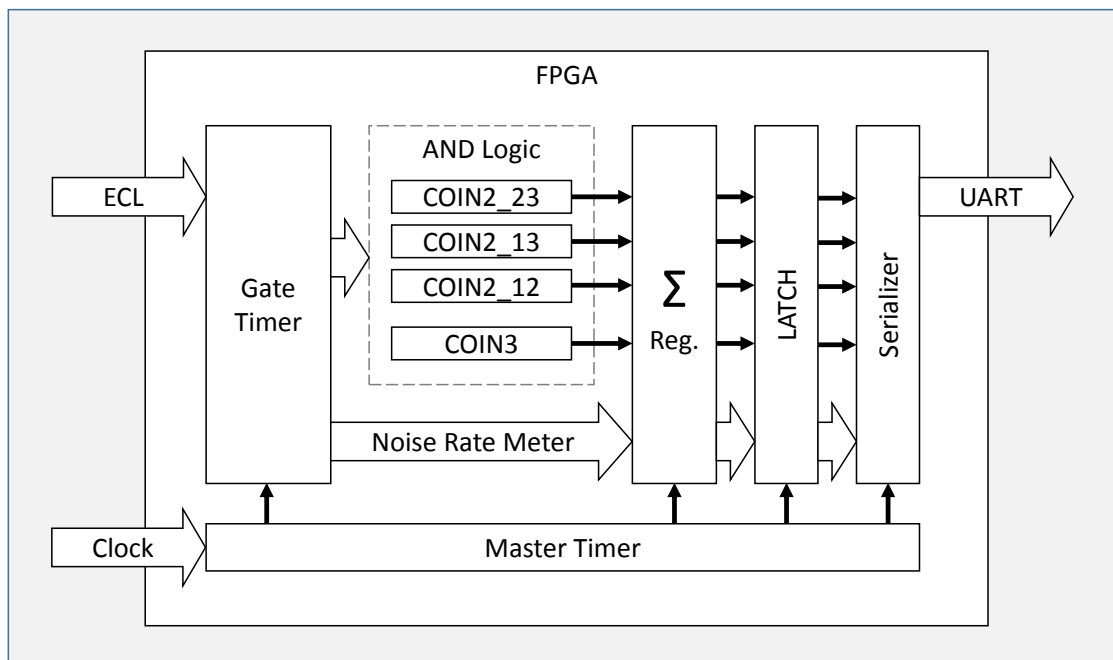


Figura 32 – Diagrama em blocos do módulo contador de eventos coincidentes

Um diagrama elétrico correspondente aos circuitos desenvolvidos neste módulo foi desenhado para facilitar seu entendimento e pode ser consultado no APENDICE B: Diagrama esquemático do Módulo Contador de Eventos Coincidentes.

Todos os circuitos lógicos foram implementados através de descrição literal do hardware em *VHDL* – vista no APENDICE F: Caderno de códigos do Módulo **Contador** – num *FPGA ALTERA Cyclone EP1C20F400C6* incluído no módulo *CAEN V1495*, específico para instrumentação científica. O módulo *CAEN V1495* também segue os padrões *VME (ANSI/IEEE 1014-1987)* e possui todas as facilidades de alimentação elétrica, geração de *clock* e montagem mecânica disponibilizadas num sistema *VMEBus* compatível.

### 3.2.1.1. Temporizador de Entrada (*Gate Timer*)

Cada pulso diferencial *ECL* recebido pelo *FPGA* representa o acionamento de um discriminador do módulo *Front-End* e deve ser comparado com os pulsos recebidos pelos demais canais de seu grupo para que se aplique a lógica de detecção de coincidências e contagem. Entretanto observou-se grande instabilidade nas primeiras montagens deste experimento devido a presença de ruído de chaveamento no discriminador e pequenas instabilidades no sinal da fotomultiplicadora. Estas características, associadas com a altíssima velocidade de resposta do *FPGA*, faziam com que os resultados das medidas de fluxo fossem mascarados devido à frequente contagem de falsas coincidências.

Para solucionar este problema foi introduzido em cada um dos canais de entrada do *FPGA* um temporizador digital (*gate timer*), acionado a cada bordo de subida do sinal *ECL* e ativo por período de tempo programável. Durante seu período ativo, qualquer bordo de subida em outro canal *ECL* do mesmo grupo será considerado evento coincidente. No esgotamento do temporizador os registradores temporários são transferidos para registradores de leitura e assumem valores iniciais.

Utilizando um osciloscópio de 20 Gs/s foi possível observar esta instabilidade e mensurar o tempo mínimo de disparo do *gate timer*. Observa-se na Figura 33 a captura de tela do instrumento no instante em que um pulso é gerado pela fotomultiplicadora, onde o canal 1 (amarelo) é o estado da linha de transmissão *ECL*, o canal 2 (verde) é o pulso corrigido pelo *gate timer*, o canal 3 (azul) é o pulso analógico extraído da saída do estágio amplificador da

eletrônica de front-end e o canal 4 (vermelho) está desconectado. Nota-se que, durante o tempo de acionamento do sinal gate timer (canal 2) ocorrem quatro pulsos na linha *ECL* (canal 1) mas que apenas o primeiro é verdadeiro e corresponde ao momento em que o nível analógico (canal 3) ultrapassa a tensão de *threshold*. Neste procedimento a fonte de trigger do osciloscópio era o sinal analógico e seu nível foi ajustado de modo a ficar igual a tensão de *threshold*. Durante o procedimento de medição a largura do pulso gate timer foi ajustada no limite inferior de funcionamento do circuito entretanto, para a versão final do projeto, assumiu-se uma margem de tolerância e este valor foi incrementado para 750ns.

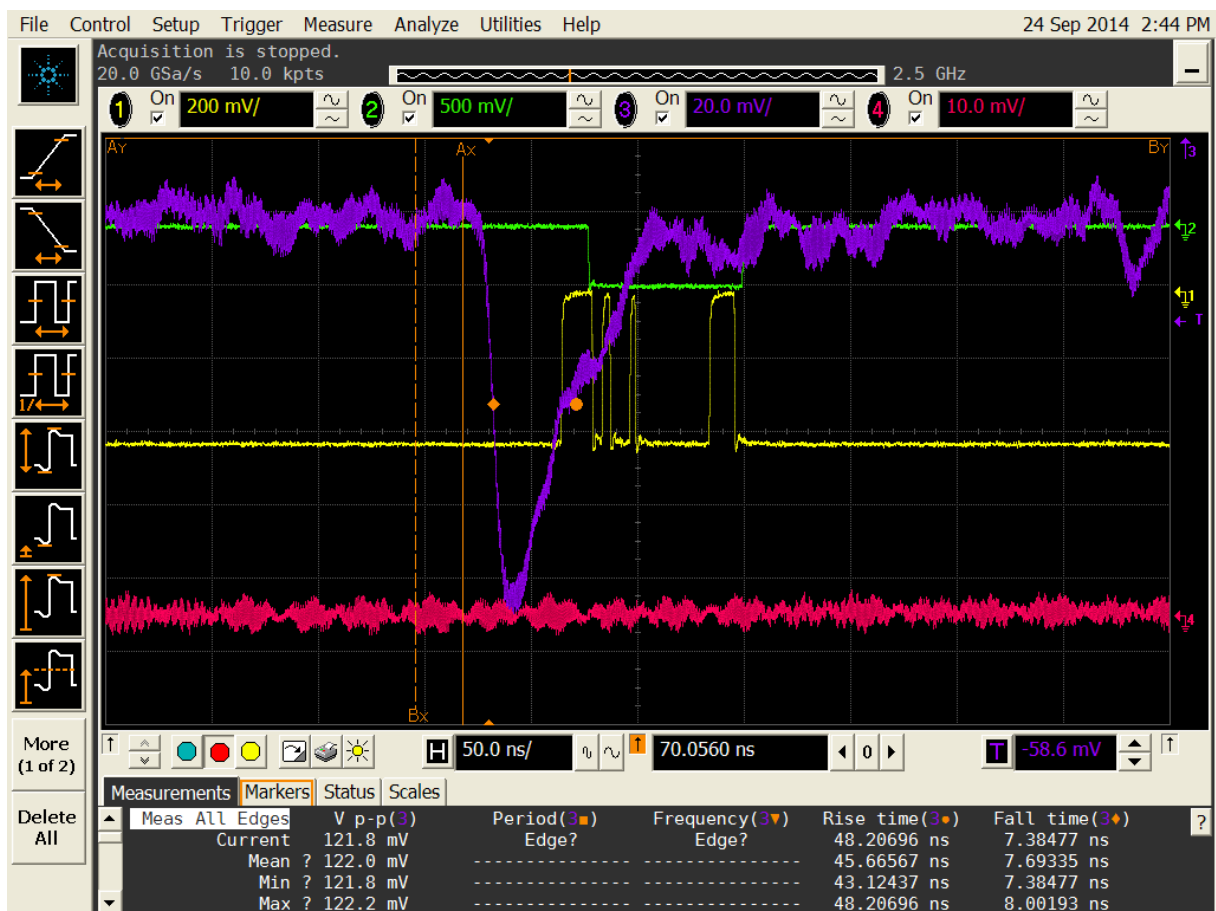


Figura 33 – Resposta ao impulso na entrada do *FPGA*

O diagrama de tempo da Figura 34 ilustra a mecânica de funcionamento deste circuito. Esta funcionalidade está implementada entre as linhas 765 e 774 e entre 861 e 870 do código *VHDL*, na forma de um circuito digital cuja representação ilustrativa simplificada segue na Figura 35.



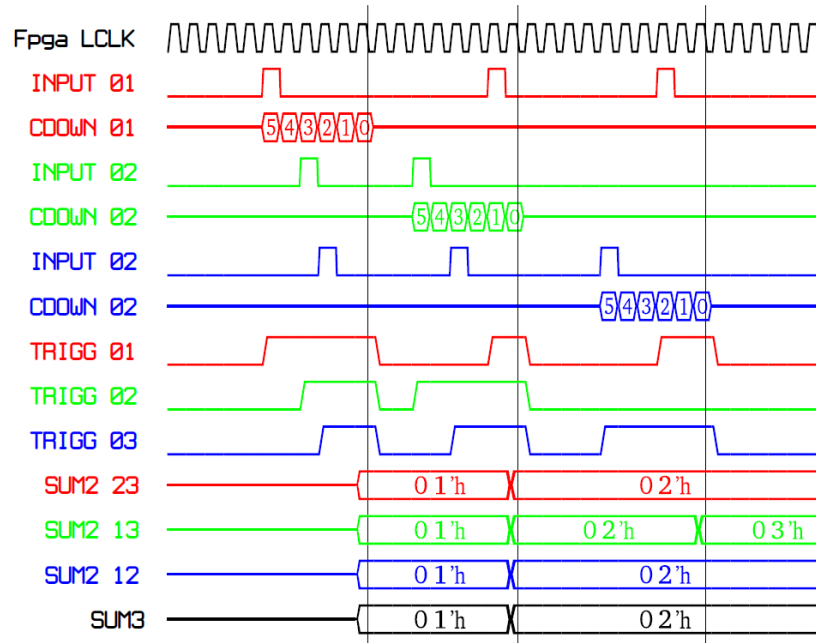


Figura 34 – Diagrama de tempo do circuito *gate timer*

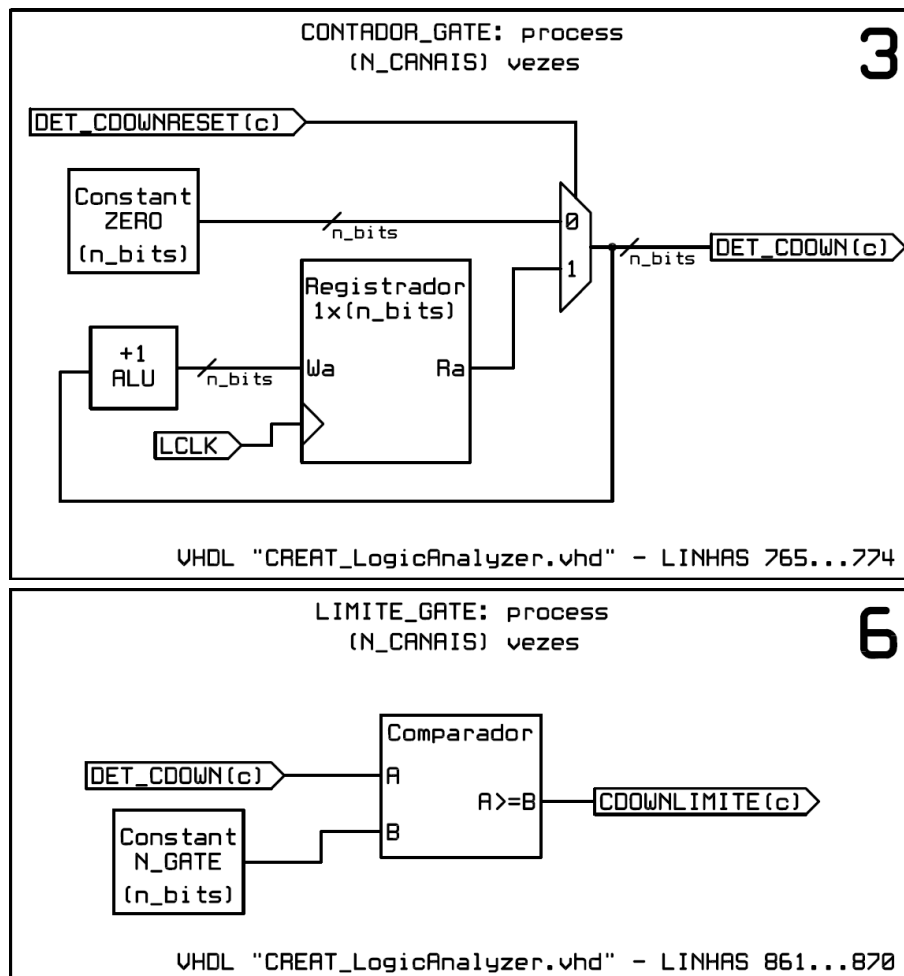


Figura 35 – Circuito *gate timer*

### 3.2.1.2. Detecção de eventos coincidentes

A detecção de eventos coincidentes é realizada através de lógica combinacional, ou seja, portas *AND* declaradas através do código *VHDL*. Por ser de natureza puramente combinacional, esta seção de código sempre entrega ao estágio seguinte o resultado da comparação de variáveis. Cabe, portanto, ao circuito temporizador do *gate timer* definir o instante em que o registrador de soma será acionado para a contagem de um evento. Esta funcionalidade está implementada entre as linhas 824 e 835 do código *VHDL* em anexo, na forma de um circuito digital cuja representação ilustrativa simplificada segue na Figura 36.

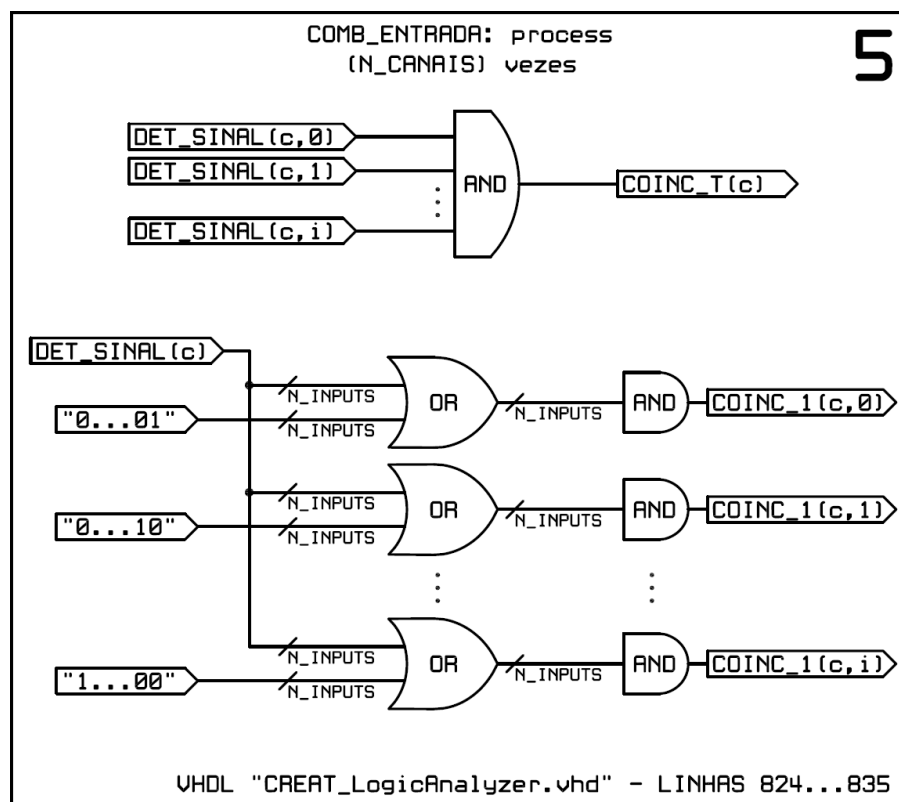


Figura 36 – Circuito detector de coincidências

### 3.2.1.3. Contagem de eventos

A contagem de eventos em cada um dos grupos de detecção é efetuada por um banco de registradores de soma, cujo valor é incrementado em uma unidade a cada esgotamento do *gate timer* e desde que, neste instante, esteja sinalizado um evento no registrador correspondente. Esta funcionalidade está implementada entre as linhas 875 e 952 do código *VHDL* em anexo, na forma de um circuito digital cuja representação ilustrativa simplificada segue na Figura 37.

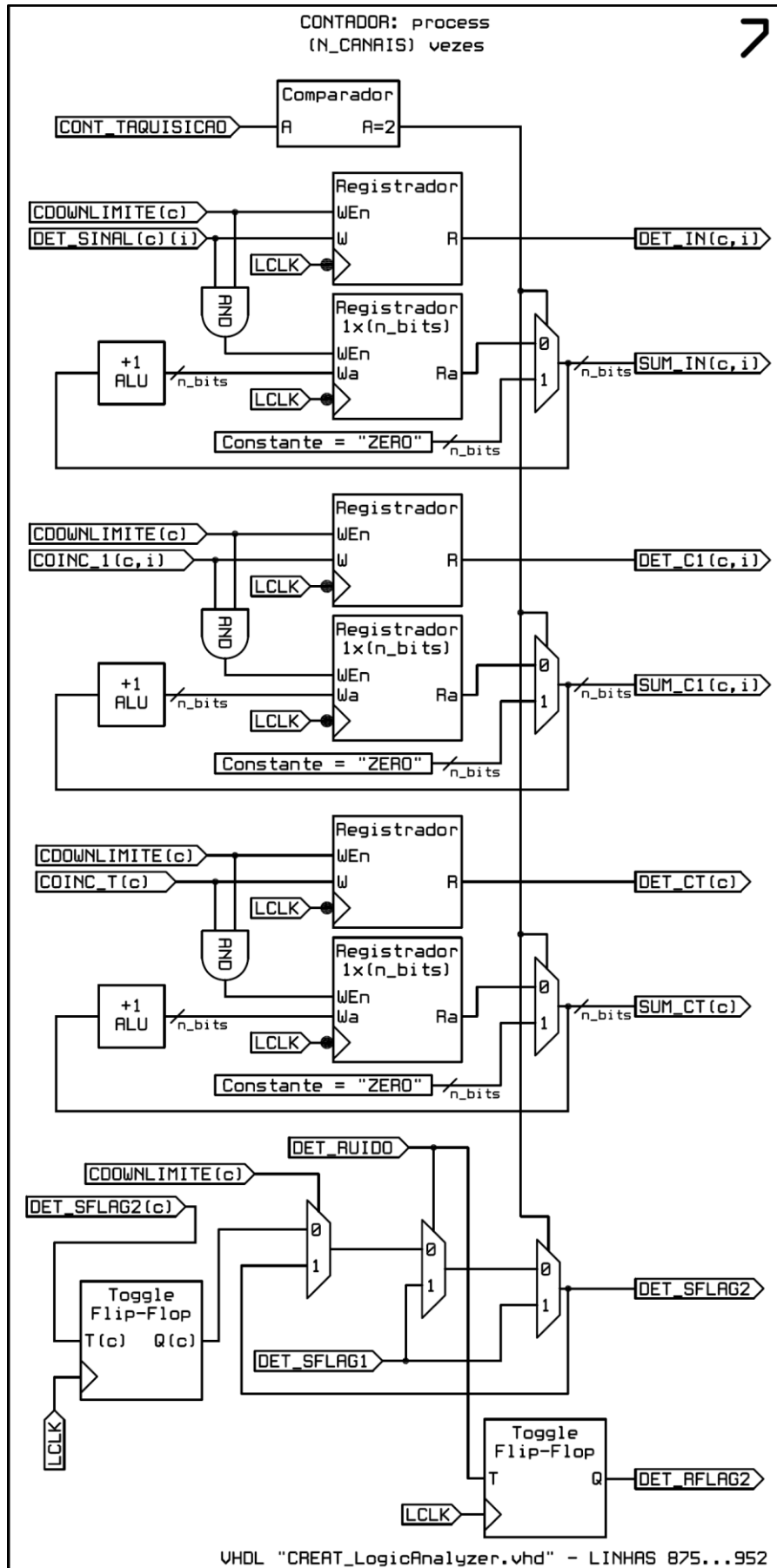


Figura 37 – Circuito contador

### 3.2.1.4. Mapa de registradores

Cada registrador de contagem pode ser lido pelo barramento *VME* e recebe, para tanto, um endereço único. Este endereço é constituído por 32 bits, sendo os primeiros 16 bits relativos à identificação do módulo (podem haver múltiplos dispositivos conectado ao mesmo barramento *VME*) e os 16 bits menos significativos identificadores dos registradores. Além dos registradores de contagem (somente leitura) foram incluídos registradores que podem ser escritos através do barramento *VME* para controle do módulo. A Tabela 2 discrimina os endereços e funções de cada um dos registradores.

Tabela 2 – Mapa de endereços

Endereço <sup>1</sup>	Descrição	Tipo <sup>2</sup>
<b>0x1000</b>	Valor instantâneo do somatório COINC_1a23	R
<b>0x1002</b>	Valor instantâneo do somatório COINC_1a13	R
<b>0x1004</b>	Valor instantâneo do somatório COINC_1a12	R
<b>0x1006</b>	Valor instantâneo do somatório COINC_Ta	R
<b>0x1008</b>	Valor instantâneo do somatório COINC_1b23	R
<b>0x100A</b>	Valor instantâneo do somatório COINC_1b13	R
<b>0x100C</b>	Valor instantâneo do somatório COINC_1b12	R
<b>0x100E</b>	Valor instantâneo do somatório COINC_Tb	R
<b>0x1010</b>	Valor instantâneo do somatório COINC_1c23	R
<b>0x1012</b>	Valor instantâneo do somatório COINC_1c13	R
<b>0x1014</b>	Valor instantâneo do somatório COINC_1c12	R
<b>0x1016</b>	Valor instantâneo do somatório COINC_Tc	R
<b>0x1018</b>	Valor instantâneo do somatório COINC_1d23	R
<b>0x101A</b>	Valor instantâneo do somatório COINC_1d13	R
<b>0x101C</b>	Valor instantâneo do somatório COINC_1d12	R
<b>0x101E</b>	Valor instantâneo do somatório COINC_Td	R
<b>0x1020</b>	Valor instantâneo do somatório COINC_1e23	R
<b>0x1022</b>	Valor instantâneo do somatório COINC_1e13	R
<b>0x1024</b>	Valor instantâneo do somatório COINC_1e12	R
<b>0x1026</b>	Valor instantâneo do somatório COINC_Te	R
<b>0x1028</b>	Valor instantâneo do somatório COINC_1f23	R
<b>0x102A</b>	Valor instantâneo do somatório COINC_1f13	R
<b>0x102C</b>	Valor instantâneo do somatório COINC_1f12	R
<b>0x102E</b>	Valor instantâneo do somatório COINC_Tf	R

<sup>1</sup> 16 bits menos significativos de identificação dos registradores. Acrescentar à esquerda os 16 bits mais significativos de identificação do módulo.

<sup>2</sup> R = Somente leitura. R/W = Leitura e Escrita

<b>0x1100</b>	Valor armazenado em LATCH2a23 (COINC_1a23 no ultimo T <sup>3</sup> )	R
<b>0x1102</b>	Valor armazenado em LATCH2a13 (COINC_1a13 no ultimo T <sup>4</sup> )	R
<b>0x1104</b>	Valor armazenado em LATCH2a12 (COINC_1a12 no ultimo T <sup>4</sup> )	R
<b>0x1106</b>	Valor armazenado em LATCH3a (COINC_Ta no ultimo T <sup>4</sup> )	R
<b>0x1108</b>	Valor armazenado em LATCH2b23 (COINC_1b23 no ultimo T <sup>4</sup> )	R
<b>0x110A</b>	Valor armazenado em LATCH2b13 (COINC_1b13 no ultimo T <sup>4</sup> )	R
<b>0x110C</b>	Valor armazenado em LATCH2b12 (COINC_1b12 no ultimo T <sup>4</sup> )	R
<b>0x110E</b>	Valor armazenado em LATCH3b (COINC_Tb no ultimo T <sup>4</sup> )	R
<b>0x1110</b>	Valor armazenado em LATCH2c23 (COINC_1c23 no ultimo T <sup>4</sup> )	R
<b>0x1112</b>	Valor armazenado em LATCH2c13 (COINC_1c13 no ultimo T <sup>4</sup> )	R
<b>0x1114</b>	Valor armazenado em LATCH2c12 (COINC_1c12 no ultimo T <sup>4</sup> )	R
<b>0x1116</b>	Valor armazenado em LATCH3c (COINC_Tc no ultimo T <sup>4</sup> )	R
<b>0x1118</b>	Valor armazenado em LATCH2d23 (COINC_1d23 no ultimo T <sup>4</sup> )	R
<b>0x111A</b>	Valor armazenado em LATCH2d13 (COINC_1d13 no ultimo T <sup>4</sup> )	R
<b>0x111C</b>	Valor armazenado em LATCH2d12 (COINC_1d12 no ultimo T <sup>4</sup> )	R
<b>0x111E</b>	Valor armazenado em LATCH3d (COINC_Td no ultimo T <sup>4</sup> )	R
<b>0x1120</b>	Valor armazenado em LATCH2e23 (COINC_1e23 no ultimo T <sup>4</sup> )	R
<b>0x1122</b>	Valor armazenado em LATCH2e13 (COINC_1e13 no ultimo T <sup>4</sup> )	R
<b>0x1124</b>	Valor armazenado em LATCH2e12 (COINC_1e12 no ultimo T <sup>4</sup> )	R
<b>0x1126</b>	Valor armazenado em LATCH3e (COINC_Te no ultimo T <sup>4</sup> )	R
<b>0x1128</b>	Valor armazenado em LATCH2f23 (COINC_1f23 no ultimo T <sup>4</sup> )	R
<b>0x112A</b>	Valor armazenado em LATCH2f13 (COINC_1f13 no ultimo T <sup>4</sup> )	R
<b>0x112C</b>	Valor armazenado em LATCH2f12 (COINC_1f12 no ultimo T <sup>4</sup> )	R
<b>0x112E</b>	Valor armazenado em LATCH3f (COINC_Tf no ultimo T <sup>4</sup> )	R
<b>0x1200</b>	Somatório instantâneo de eventos na entrada 1a	R
<b>0x1202</b>	Somatório instantâneo de eventos na entrada 2a	R
<b>0x1204</b>	Somatório instantâneo de eventos na entrada 3a	R
<b>0x1208</b>	Somatório instantâneo de eventos na entrada 1b	R
<b>0x120A</b>	Somatório instantâneo de eventos na entrada 2b	R
<b>0x120C</b>	Somatório instantâneo de eventos na entrada 3b	R
<b>0x1210</b>	Somatório instantâneo de eventos na entrada 1c	R
<b>0x1212</b>	Somatório instantâneo de eventos na entrada 2c	R
<b>0x1214</b>	Somatório instantâneo de eventos na entrada 3c	R
<b>0x1218</b>	Somatório instantâneo de eventos na entrada 1d	R
<b>0x121A</b>	Somatório instantâneo de eventos na entrada 2d	R
<b>0x121C</b>	Somatório instantâneo de eventos na entrada 3d	R
<b>0x1220</b>	Somatório instantâneo de eventos na entrada 1e	R
<b>0x1222</b>	Somatório instantâneo de eventos na entrada 2e	R
<b>0x1224</b>	Somatório instantâneo de eventos na entrada 3e	R
<b>0x1228</b>	Somatório instantâneo de eventos na entrada 1f	R
<b>0x122A</b>	Somatório instantâneo de eventos na entrada 2f	R
<b>0x122C</b>	Somatório instantâneo de eventos na entrada 3f	R

<sup>3</sup> Em que T é o instante de tempo em que se completa um intervalo de aquisição. O intervalo de aquisição é definido pelo registrador 0x2000, em segundos.

<b>0x1300</b>	Quantidade de eventos na entrada 1a no último R <sup>4</sup>	R
<b>0x1302</b>	Quantidade de eventos na entrada 2a no último R <sup>5</sup>	R
<b>0x1304</b>	Quantidade de eventos na entrada 3a no último R <sup>5</sup>	R
<b>0x1308</b>	Quantidade de eventos na entrada 1b no último R <sup>5</sup>	R
<b>0x130A</b>	Quantidade de eventos na entrada 2b no último R <sup>5</sup>	R
<b>0x130C</b>	Quantidade de eventos na entrada 3b no último R <sup>5</sup>	R
<b>0x1310</b>	Quantidade de eventos na entrada 1c no último R <sup>5</sup>	R
<b>0x1312</b>	Quantidade de eventos na entrada 2c no último R <sup>5</sup>	R
<b>0x1314</b>	Quantidade de eventos na entrada 3c no último R <sup>5</sup>	R
<b>0x1318</b>	Quantidade de eventos na entrada 1d no último R <sup>5</sup>	R
<b>0x131A</b>	Quantidade de eventos na entrada 2d no último R <sup>5</sup>	R
<b>0x131C</b>	Quantidade de eventos na entrada 3d no último R <sup>5</sup>	R
<b>0x1320</b>	Quantidade de eventos na entrada 1e no último R <sup>5</sup>	R
<b>0x1322</b>	Quantidade de eventos na entrada 2e no último R <sup>5</sup>	R
<b>0x1324</b>	Quantidade de eventos na entrada 3e no último R <sup>5</sup>	R
<b>0x1328</b>	Quantidade de eventos na entrada 1f no último R <sup>5</sup>	R
<b>0x132A</b>	Quantidade de eventos na entrada 2f no último R <sup>5</sup>	R
<b>0x132C</b>	Quantidade de eventos na entrada 3f no último R <sup>5</sup>	R
<b>0x2000</b>	Tempo do intervalo de aquisição em segundos	R/W
<b>0x2002</b>	Tempo do <i>gate</i> das entradas em unidades de <i>clock</i>	R/W
<b>0x2004</b>	Controle do PATCH de saída dos conectores NIM	R/W
<b>0x2006</b>	UNIQUE ID do último conjunto de registradores transmitido	R
<b>0x2008</b>	Palavra de controle do canal de análise de dados	R/W

Os registradores podem ser divididos em cinco grupos:

1. 0x10XX – Registros instantâneos de contagem de coincidência

A leitura destes endereços retorna o valor instantâneo dos registradores de soma e tem como finalidade principal analisar o funcionamento dos contadores. Como os registradores de soma tem seus valores continuamente alterados, a leitura destes registros pode não ser estável (seus valores podem ser alterados durante o ciclo de leitura)

2. 0x11XX – *Latch* de registros da última contagem de coincidência

Neste intervalo de endereços são lidos os *latch's* dos registradores de contagem de coincidências. Estes *latch's* são carregados no instante imediatamente anterior ao encerramento do período de aquisição com os valores dos

---

<sup>4</sup> Em que R é o instante de tempo em que se completa um intervalo de aquisição de ruído. O intervalo de aquisição de ruído é definido pelo registrador não manipulável N\_MED\_RUIDO em unidades de *clock*.

registradores de soma e são mantidos inalterados durante o período atual de aquisição.

3. 0x12XX – Registros instantâneos de contagem de ruído

Análogo ao intervalo 0x10XX, porém retorna valores de medida de ruído.

4. 0x13XX – *Latch* de registros da última contagem de ruído

Análogo ao intervalo 0x11XX, porém retorna valores de medida de ruído.

5. 0x20XX – Registradores de controle

São utilizados para controle do contador.

O valor do registrador 0x2000 define o período de aquisição em segundos e pode ser alterado a qualquer momento, mesmo com o contador em funcionamento. Neste caso, se seu valor for definido abaixo da quantidade em segundos transcorrida no período de aquisição vigente o mesmo será encerrado no próximo segundo cheio. Esta característica permite que um software em execução num microcomputador conectado ao barramento *VME* reinicie a qualquer momento a contagem. Vale lembrar que a transmissão serial sempre informará o *Unique ID* e o tempo efetivamente transcorrido do período de aquisição transmitido.

O registrador 0x2002 define o período do *gate*, em unidades de *clock*, sendo cada unidade igual a 25ns (40MHz) no caso do módulo V1495.

A funcionalidade implementada pelo registrador 0x2004 permite que os conectores *NIM* do módulo V1495 sejam utilizados para debug do contador. Seu valor define o grupo cujos sinais internos são direcionados para os conectores *NIM*.

O registrador 0x2006 retorna um valor único para cada período de aquisição, incrementado em uma unidade sempre que se inicia um período de aquisição novo. Este registrador pode ser utilizado pelo software para identificar o encerramento dos períodos de aquisição.

O registrador 0x2008 controla o modo de funcionamento e as fontes de sinal do canal de reentradas.

### 3.2.1.5. Transmissão serial dos registradores

A metodologia trivial aplicada à aquisição de dados em experimentos deste tipo inclui um microcomputador efetuando acessos aos registradores de contagem via barramento *VME*. Entretanto, esta abordagem implica na manutenção do microcomputador em funcionamento ininterrupto, condição indesejada para este projeto devido ao elevado consumo energético (o módulo de calibração, p. ex. consome cerca de 24W) e ao fator de risco do software altamente complexo. Optou-se, portanto, desenvolver o módulo de aquisição de dados embarcado num micro controlador e, de modo a prover comunicação direta entre o micro controlador e a *FPGA* desenvolveu-se, nesta última, um circuito capaz de efetuar transmissões seriais.

Após cada período de aquisição todos os registradores acumulados são então transmitidos pela saída serial incluída no *FPGA* na forma de uma sequência de caracteres *ASCII*. A cada segundo uma sequência reduzida é transmitida com a finalidade de reportar o funcionamento do circuito. As sequências obedecem às especificações de 9600 *baud/seg*, 8 *bits* de dados, paridade par (positiva), 2 *stop bits* e são compostas pelas seguintes informações:

1. Sequência de *STATUS* (uma por segundo):
  - a. *Unique ID* do período de aquisição
  - b. Tempo decorrido, em segundos, do período de aquisição atual
2. Sequência completa (uma a cada período de aquisição):
  - a. *Unique ID* do período de aquisição
  - b. Intervalo, em segundos, do período de aquisição transmitido
  - c. Largura do *gate* (parâmetro  $n_{gate}$ )
  - d. Palavra de controle do canal de análise de dados
  - e. Registradores de contagem, na ordem abaixo, para cada canal:
    - i. Contagens de coincidências duplas
    - ii. Contagens de medição de ruído
    - iii. Contagem de coincidência
  - f. *Checksum*



Tanto para as sequencias de status quanto sequencias completas foi implementado um protocolo de transmissão de modo que é transmitida a seguinte sequência:

1. Caractere de início de transmissão (apenas um por sequência)

- a. 7Bh (*ASCII* "{") para transmissões de status
- b. 7Ch (*ASCII* "|") para transmissões completas

2. Transmissão dos registradores:

- 2.1. Caractere identificador de registrador

20h (*ASCII* "SPACE") para o primeiro registrador, sendo incrementado em uma unidade para cada registrador posterior, ou seja, 20h, 21h, 22h...

- 2.2. Transmissão do valor do registrador

O valor do registrador é transmitido em base hexadecimal codificada em caracteres *ASCII*. Para cada *nibble* do registrador haverá um caractere *ASCII* correspondente, cujo valor varia de 0 a F.

- 2.3. Caractere indicador de fim do registrador

Após cada registrador é inserido o caractere 7Dh (*ASCII* "}")

3. Sequência de fim de transmissão

Para marcar o final de uma transmissão são enviados, na sequência, os símbolos *ASCII* 7Eh, 0Dh e 0Ah, respectivamente, "~", retorno de carro e alimentação de linha.

### **3.2.1.6. Acesso aos registradores pelo *VMEbus***

Registradores internos ao *FPGA* também podem ser acessados através do barramento *VME*, permitindo que alguns parâmetros de contagem (como tempo de aquisição e *gate timer*) sejam alterados sem que seja necessário reprogramá-lo. Além desta, outra principal diferença em relação à transmissão serial é que pelo barramento *VME* o computador deve iniciar as operações de leitura e escrita, não sendo permitido ao *FPGA* gerar uma interrupção no computador. No contexto deste projeto, este método de acesso é utilizado para controle e

calibração do experimento, mas não foi intencionado que seja utilizado para sua operação normal, embora isto seja possível. Um módulo de interface (3.1.3) é necessário para conectar o computador ao barramento *VME* através de um cabo USB.

### 3.2.2. Módulo de Aquisição de Dados

O recebimento, a verificação de erros, o processamento e o armazenamento dos registradores contidos na *STRING* serial enviada pelo módulo contador de eventos coincidentes são efetuados neste módulo, que também é responsável por transmitir ao sistema de satélite um sinal vital indicador de funcionamento do experimento. Desenvolvido em torno de um micro controlador *AVR* da fabricante *ATMEL*, dotado de um relógio de tempo real (*RTC*) e de um cartão de memória não volátil o módulo de aquisição de dados foi programado com o código que pode ser visto no APENDICE D: Código *C* do Módulo de Aquisição de Dados.

A Figura 38 destaca a posição do módulo de processamento de dados no diagrama em blocos do sistema *CRE@AT* e a Figura 39 contém o fluxograma lógico deste módulo.

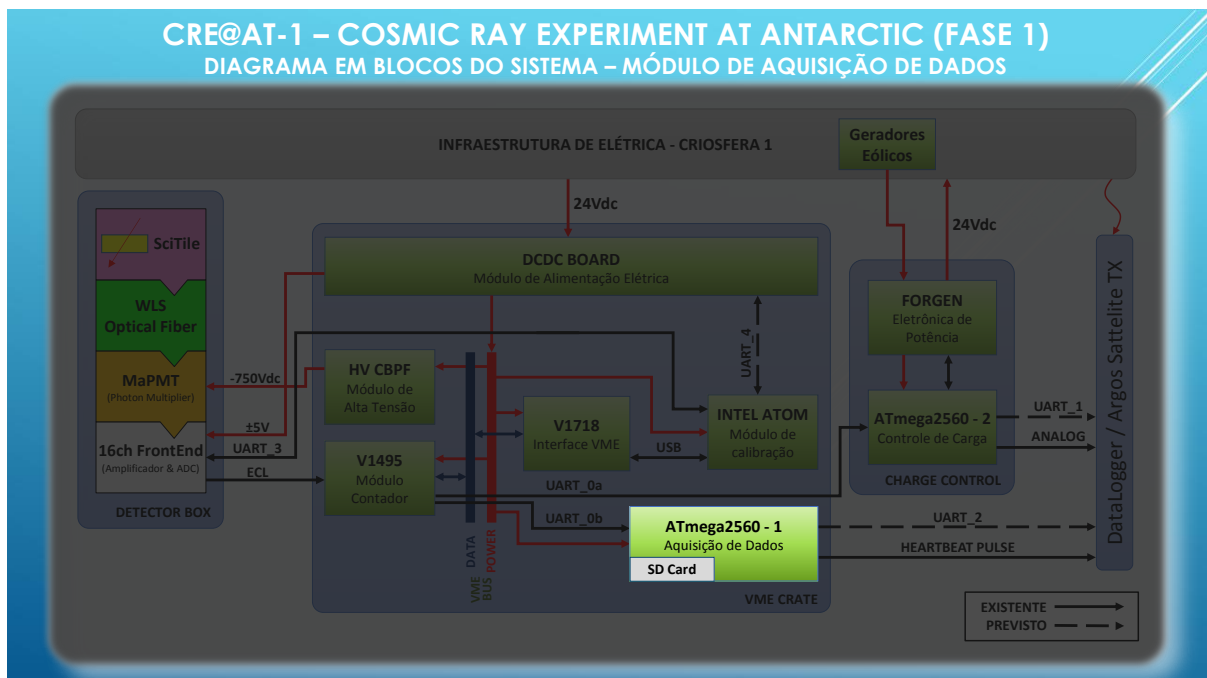


Figura 38 – Diagrama em blocos com destaque ao módulo de aquisição de dados

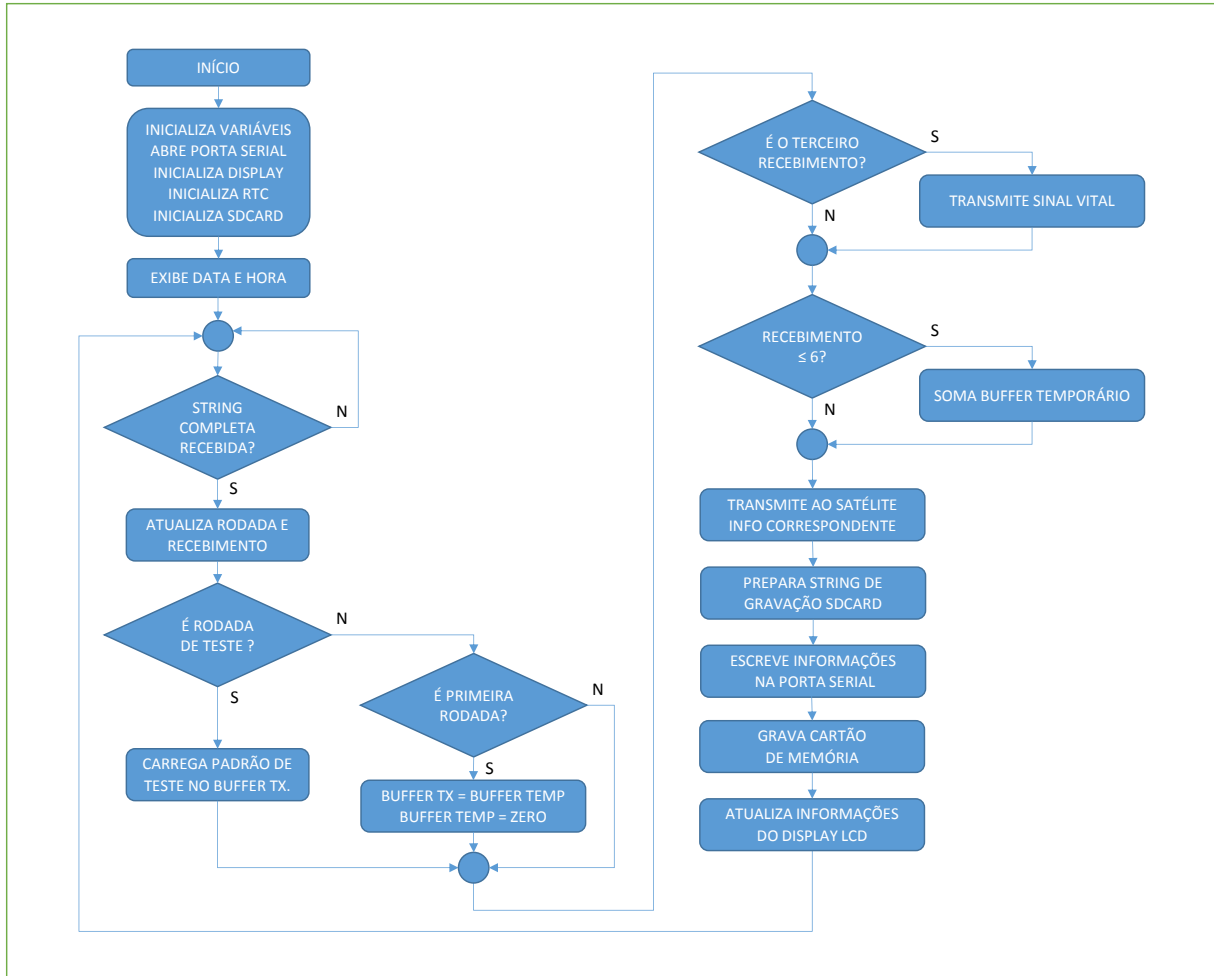


Figura 39 – Fluxograma lógico do módulo de aquisição de dados

### 3.2.3. Módulo de Calibração

Um computador baseado no sistema operacional *LINUX UBUNTU* executado na plataforma *INTEL ATOM* foi incluído no experimento para execução testes e reprogramações durante seu desenvolvimento e sua posterior calibração, após concluído. A Figura 40 destaca a posição do módulo de calibração no diagrama em blocos do sistema *CRE@AT*.

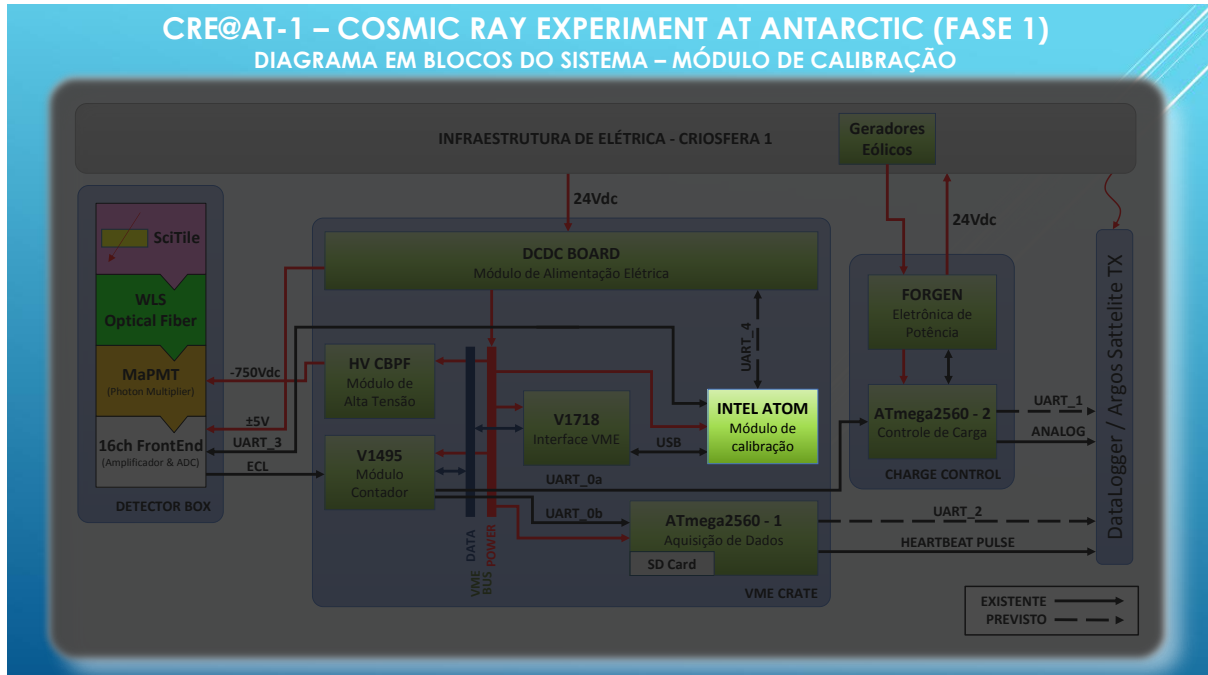


Figura 40 – Diagrama em blocos com destaque ao módulo de calibração

O módulo de acesso ao barramento *VME* descrito no item 3.1.3 permite que o módulo de calibração realize leituras e escritas no banco de registradores descrito no item 3.2.1.4. Através destas operações de leitura e escrita é possível obter acesso total ao funcionamento do módulo de trigger, além de ser possível também reprogramar seu *FPGA*. A aplicação *ROOT*, desenvolvida pelo *CERN*, também foi utilizada para permitir análise gráfica dos dados de calibração e testes do detector.

### 3.2.3.1. Algoritmo de Calibração

O somatório da contagem de eventos por período de aquisição é função de quatro parâmetros fundamentais: da taxa de atividade (fluxo) da radiação cósmica galáctica, do tempo de aquisição, do nível de alta-tensão aplicado à fotomultiplicadora (pois afeta o ganho da foto multiplicação) e do nível da tensão de discriminação (*threshold*)

Como o fluxo de partículas é de natureza aleatória, o tempo de exposição costuma se relacionar de maneira diretamente proporcional com o número total de contagens. Sua especificação está intimamente relacionada com a resolução temporal pretendida para o sistema e com o valor de fundo de escala dos contadores.

O ganho da fotomultiplicadora está relacionado com a alta-tensão nela aplicada para aceleração dos elétrons. Existe um ganho ideal (a ser calibrado) acima do qual observam-se falsos eventos devido ao ruído intrínseco do processo de foto multiplicação. O nível da alta tensão aplicada obedece à relação não linear com o número de contagens.

A tensão de discriminação estabelece o patamar mínimo de ruído para a eletrônica analógica do detector, não obedece à relação linear com o número de contagens e também possui valor ideal a ser calibrado. Este é um parâmetro crítico.

Foi então concebida – em comandos *Shell Script* do Linux –rotina de calibração para estes dois últimos parâmetros, consistindo basicamente em executar repetidas vezes o programa de aquisição variando determinado parâmetro a cada execução. Posteriormente os dados coletados são formatados em arquivo específico e introduzidos no programa *ROOT* para análise gráfica.

### **3.2.4. Módulo de Alimentação Elétrica**

Para viabilizar a alimentação elétrica deste projeto foi disponibilizado acesso ao barramento de 24 Vdc do módulo Criosfera 1. Este barramento está conectado ao banco de baterias do módulo e é alimentado por dois sistemas locais de geração de energia, sendo um por fonte solar e outro por fonte eólica.

Algumas características deste sistema, entretanto, fizeram com que fosse necessária a adoção de uma solução específica de alimentação elétrica para todo o experimento pois havia necessidade de alimentar cargas em diferentes níveis de tensão com elevado grau de estabilidade. Esta solução foi baseada na integração de dispositivos conversores DC/DC com circuitos periféricos de monitoração e controle, montados em uma placa de circuito impresso padrão *VME* para instalação em *CRATE*.

A Figura 41 destaca a posição do módulo de alimentação elétrica no diagrama em blocos do sistema *CRE@AT*.

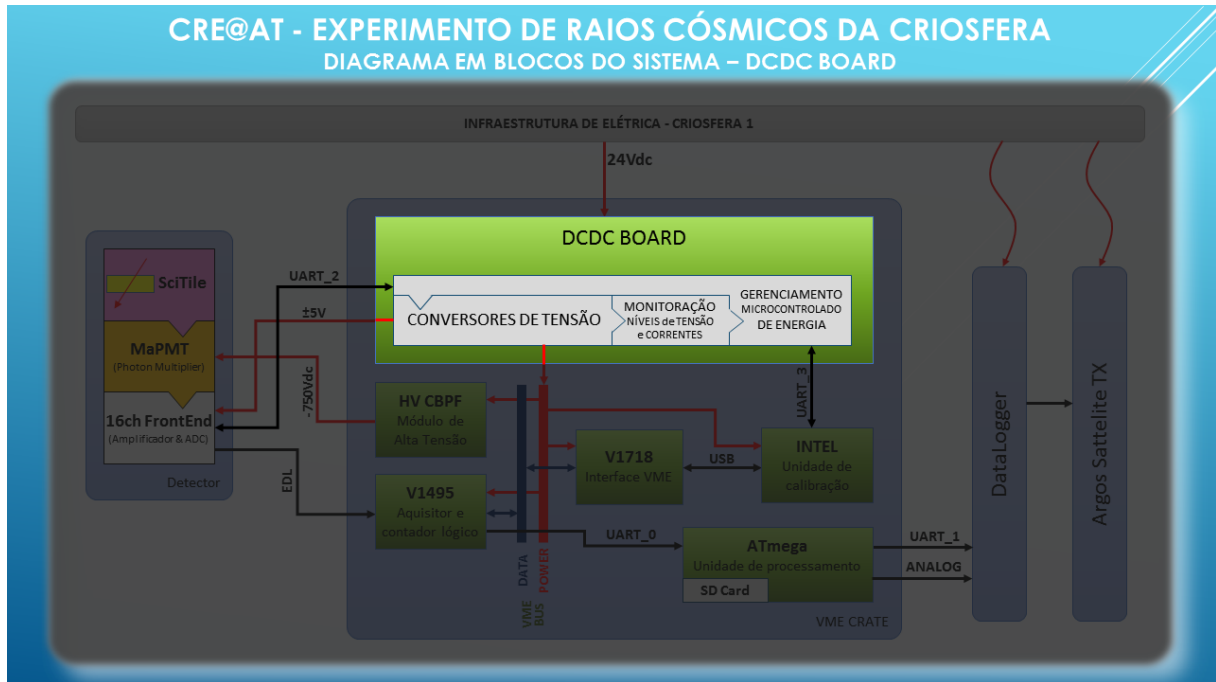


Figura 41 – Diagrama em blocos com destaque ao módulo de alimentação elétrica

### 3.2.4.1. Requisitos do Projeto

Foram definidos como requisitos deste projeto:

- Possibilidade de entrada de força com 12 ou 24 volts nominais;
- Acondicionamento de entrada de força não regulada (prevista variação até 36V);
- Alta qualidade do barramento de *ground*;
- Possibilidade de isolamento elétrico total entre os *grounds* da fonte de potência.

### 3.2.4.2. Dimensionamento de Potência

O projeto do módulo de alimentação elétrica levou em consideração o consumo máximo declarado em *datasheet* de cada elemento a ser alimentado, como pode ser visto na Tabela 3.

Tabela 3 – Conversores de Energia DC-DC

Espectrômetro de Raios Cósmitas da Criosfera - Consumo Elétrico do Experimento							
Assumindo HV com 6 cargas de 2kV a 2mA							
Consumo de Energia pelos Dispositivos do Experimento							
Finalidade	Dispositivo	Consumo de Corrente					Potência Teórica
		-12v	-5v	-2v	+5v	+12v	
Gerador de Alta Tensão	V6533	2,15 A	0,00 A	0,00 A	0,20 A	2,15 A	52,6 W
Lógica de Processamento	V1495	0,00 A	0,00 A	0,00 A	1,00 A	0,00 A	5,0 W
Expansor I/O p/ V1495	A395A	0,00 A	0,00 A	0,00 A	0,10 A	0,00 A	0,5 W
VMEBus	Crate	0,00 A	0,00 A	0,00 A	1,24 A	0,00 A	6,2 W
Interface Computacional	V1718	0,15 A	0,00 A	0,00 A	0,80 A	0,00 A	5,8 W
Computador de Aquisição	INTEL	0,00 A	0,00 A	0,00 A	0,00 A	2,00 A	24,0 W
Consumo Parcial do Sub-Sistema 1 ->		2,30 A	0,00 A	0,00 A	3,34 A	4,15 A	94,1 W
obs: Assumindo os valores nominais de máximo consumo dos dispositivos.							
Eletrônica Front-End	Guedes	0,00 A	2,57 A	2,40 A	1,42 A	0,00 A	19,9 W
Consumo Parcial do Sub-Sistema 2 ->		0,00 A	2,57 A	2,40 A	1,42 A	0,00 A	19,9 W
obs: Assumindo os valores nominais de máximo consumo dos dispositivos.							
TOTAL GERAL CONSUMIDO ->							114,0 W
Energia drenada da infraestrutura 24Vdc do módulo Criosfera 1							
Sub-Sistema	Volts	Output		Eficiência	Input 24 Vdc		Potência Sub-Sist.
		Ampéres	Watts		Ampéres	Watts	
Crate Modules	-12v	2,30 A	27,60 W	0,91	1,27 A	30,5 W	104,6 W
	+5v	3,34 A	16,70 W	0,86	0,81 A	19,4 W	
	+12v	4,15 A	49,80 W	0,91	2,28 A	54,7 W	
Elet. Front-End	-5v	2,57 A	12,84 W	0,84	0,64 A	15,3 W	30,9 W
	-2v	2,40 A	4,80 W	0,67	0,30 A	7,1 W	
	+5v	1,42 A	7,08 W	0,84	0,35 A	8,4 W	
obs: Assumindo os valores nominais de máximo consumo dos dispositivos.							
TOTAL GERAL DRENADO ->					5,65 A	135,5 W	
Dimensionamento e Coeficiente de Segurança dos Conversores DC-DC							
Conversor DC-DC	Output Projeto		Output Máximo	Carga do conversor	Eficiência		
	Volts	Ampéres			Máxima	Pto. Op.	
MURATA UHE-12/2500-Q12-C	-12v	2,30 A	2,50 A	92%	0,875	0,905	
MURATA UHE-5/5000-Q12-C	+5v	3,34 A	5,00 A	67%	0,875	0,860	
MURATA UQQ-12/8-Q12PB-C	+12v	4,15 A	8,00 A	52%	0,915	0,910	
MURATA UHE-5/5000-Q12-C	+5v	1,42 A	5,00 A	28%	0,875	0,840	
MURATA UHE-5/5000-Q12-C	-5v	2,57 A	5,00 A	51%	0,875	0,860	
MURATA UHE-5/5000-Q12-C	-2v	2,40 A	5,00 A	48%	0,875	0,860	

A Tabela 4 detalha a potência entregue a cada uma das linhas de alimentação e a Tabela 5 resume o consumo total e a eficiência de conversão do módulo.

Tabela 4 – Consumo por linha de alimentação

Sub-Sistema 1 - Crate Modules -				Sub-Sistema 2 - Eletrônica de Front-End -		
-12v	2,30 A	27,60 W		-5v	2,57 A	12,84 W
+5v	3,34 A	16,70 W		-2v	2,40 A	4,80 W
+12v	4,15 A	49,80 W		+5v	1,42 A	7,08 W

Tabela 5 – Consumo total e eficiência da conversão

Espectrômetro de Raios Cósmicos da Criosfera - Total				Eficiência da Conversão		
+24v	5,65 A	135,50 W	Input 24v	135,5 W	<b>84%</b>	
ASSUMINDO VALORES DE CONSUMO MÁXIMO			Perda	-21,5 W		
			Output	114,0 W		

É importante considerar que o projeto do módulo de alimentação foi executado antes da definição do layout final do experimento e que a característica de algumas cargas foi modificada:

- O modelo da fonte de alta tensão foi substituído

Não houve tempo hábil de trâmite para compra da fonte V6533 (CAEN) e foi utilizada fonte fabricada no próprio laboratório

- O fornecimento de energia para a eletrônica de *front-end* (3.1.1.4) foi alterado

A regulagem de -2 V para a eletrônica de *Front-End* passou a ser efetuada diretamente em seu circuito, fazendo com que este necessitasse apenas do fornecimento de +5 V e - 5 V. No projeto original já era considerada a adoção de um regulador linear intermediário, portanto não houve alteração quanto aos conversores DC-DC.

### 3.2.4.3. Importância das Características de Isolamento

O projeto deste experimento levou em consideração que o módulo Criosfera I não possui sistema de aterramento e que, portanto, poderiam haver diversos problemas não previstos relacionados a diferenças entre potenciais das malhas de referência pois suas interconexões é



realizada através do próprio barramento de potência. Tendo em vista então a falta de detalhamento técnico acerca do sistema de distribuição de energia do módulo Criosfera I optou-se por um design de alimentação capaz de isolar galvanicamente primário e secundário das fontes de alimentação.

### 3.2.4.4. Conversores de Tensão

Devido à necessidade de prover alta disponibilidade ao experimento (baixo *MTBF* - *Mean Time Between Failures*), à sua própria complexidade e ao curto tempo disponível para seu desenvolvimento optou-se por adquirir no mercado solução pré-fabricada para conversão de tensões do tipo DC-DC. A Tabela 6 descreve os dispositivos utilizados:

Tabela 6 – Conversores de Energia DC-DC

Linha	Part Number	Corrente Máxima	Tensão de Entrada	Fabricante
+12V_A	UQQ-12/8-Q12PB-C	8 A	9 a 36 Volts	<i>Murata Power Supply</i>
+5V_B	UHE-5/5000-Q12-C	5 A		
-12V_C	UHE-12/2500-Q12-C	2,5 A		
+5V_D	UHE-5/5000-Q12-C	5 A		
-5V_E	UHE-5/5000-Q12-C	5 A		
-5V_F	UHE-5/5000-Q12-C	5 A		

### 3.2.4.5. Diagramas Esquemáticos

Os diagramas esquemáticos do módulo de alimentação elétrica podem ser vistos no Apêndice C. Para facilitar sua visualização, o documento possui uma folha principal (folha 1 – mãe) que faz referência a outras três folhas (folhas 2, 3 e 4 – filhas).

### 3.2.5. Módulo Controlador de Carga

Não era escopo inicial deste projeto desenvolver circuitos eletrônicos de controle para as turbinas eólicas. Entretanto, devido a dificuldades observadas em campo e explicadas no APENDICE A: Problemas na geração de energia eólica do módulo Criosfera I, optou-se por desenvolver um novo circuito controlador para carga de baterias e controle das turbinas e embarcar nele a parte de transmissão de dados via satélite do sistema de aquisição. A montagem do conjunto mecânico de circuitos eletrônicos de potência e de controle do sistema de geração de energia eólica do módulo Criosfera I, após as intervenções efetuadas, é visto na Figura 42.

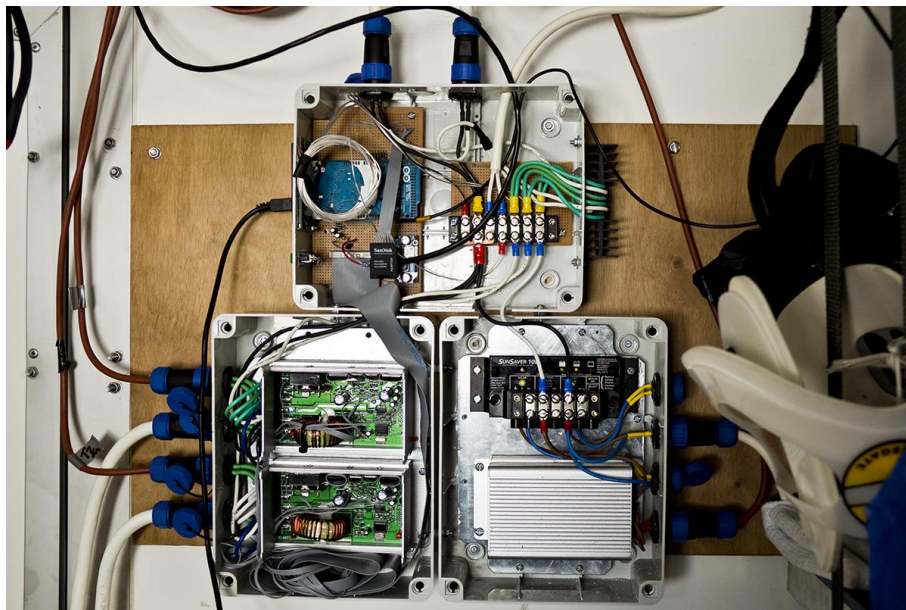


Figura 42 – Sistema de controle das turbinas eólicas

Tal decisão baseou-se na evidência de que não haveria – com a utilização do sistema fornecido – geração eólica de energia suficiente para manter o experimento operacional em regime permanente (como se havia previsto), e na decisão de mantê-lo operacional por apenas 30 minutos diários. Como a transmissão de dados por satélite é extremamente limitada (em apenas um sinal cujo nível de tensão quantizado em 7 bits é transmitido de hora em hora) ficou evidente que caso a interface de transmissão por satélite fosse acionada apenas 30 minutos por dia não haveria tempo hábil para transmissão de informação útil.

Entretanto, o controlador de carga proposto foi projetado para funcionar ininterruptamente, tornando oportuno embarcar neste dispositivo uma rotina que permitisse transmitir uma vez por dia os resultados do experimento, fragmentados a cada hora.

O diagrama de funcionamento deste controlador será oportunamente explicado em A.3

## 4. Resultados Obtidos

O algoritmo de calibração do experimento (3.2.3.1) foi executado para tomada de dados e análise de performance em dois momentos diferentes, no laboratório do CBPF e na posição de instalação definitiva, no módulo Criosfera I. Em todas as ocasiões o experimento mostrou-se operacional dentro da faixa de tolerância esperada.

Com o experimento ainda no CBPF foram considerados 80 períodos de aquisição – de 600 segundos cada – para a calibração de *threshold*. Para cada período de aquisição foi registrada a taxa de fluxo corrigida de cada um dos grupos de coincidência, correspondente a quantidade de contagens por segundo após ser corrigida pelo método das coincidências. O arquivo de dados obtido recebeu então tratamento estatístico com auxílio do software *ROOT*, sendo aplicado o algoritmo descrito no APENDICE G de modo a gerar os histogramas vistos nas Figuras 43, 44, 45, 46 e 47. Estes histogramas relacionam o fluxo corrigido expresso em contagens por grupo *SciTile* por segundo (eixo horizontal) com a quantidade de períodos nos quais tal fluxo foi observado (eixo vertical).

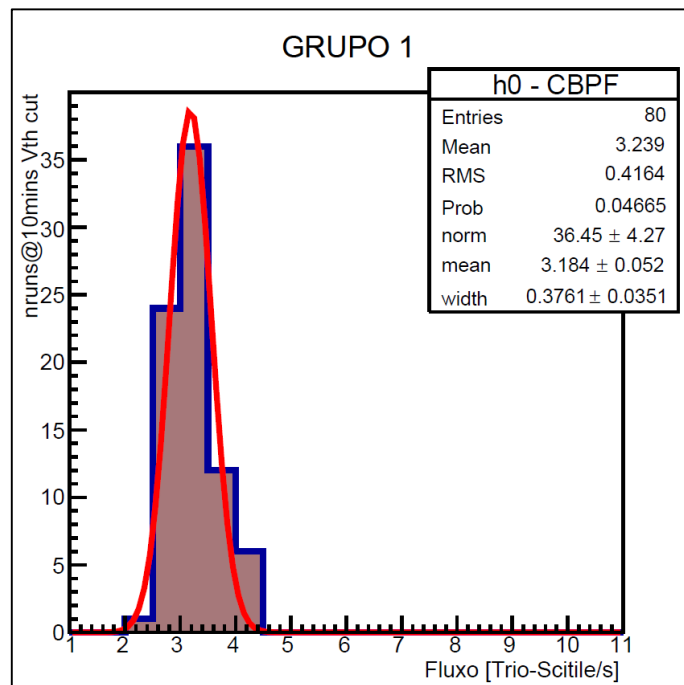


Figura 43 – Histograma de fluxo no Brasil para o grupo de coincidência 1

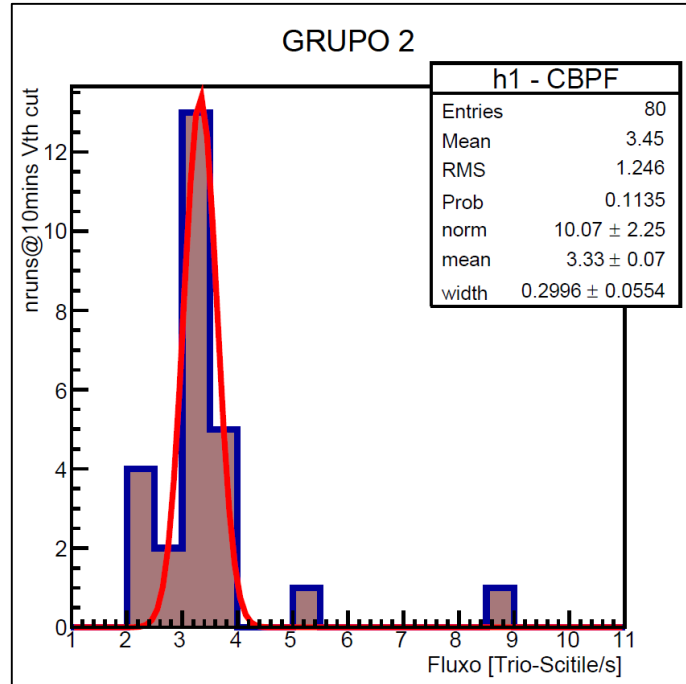


Figura 44 – Histograma de fluxo no Brasil para o grupo de coincidência 2

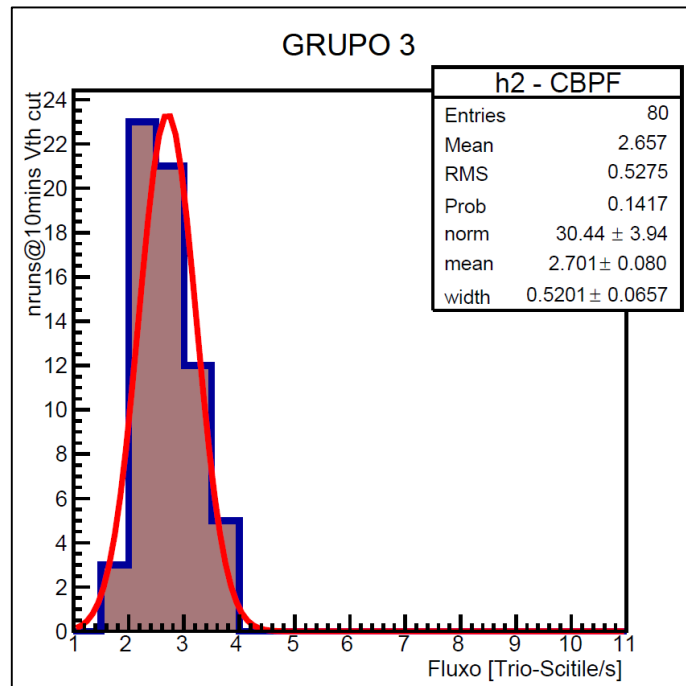


Figura 45 – Histograma de fluxo no Brasil para o grupo de coincidência 3

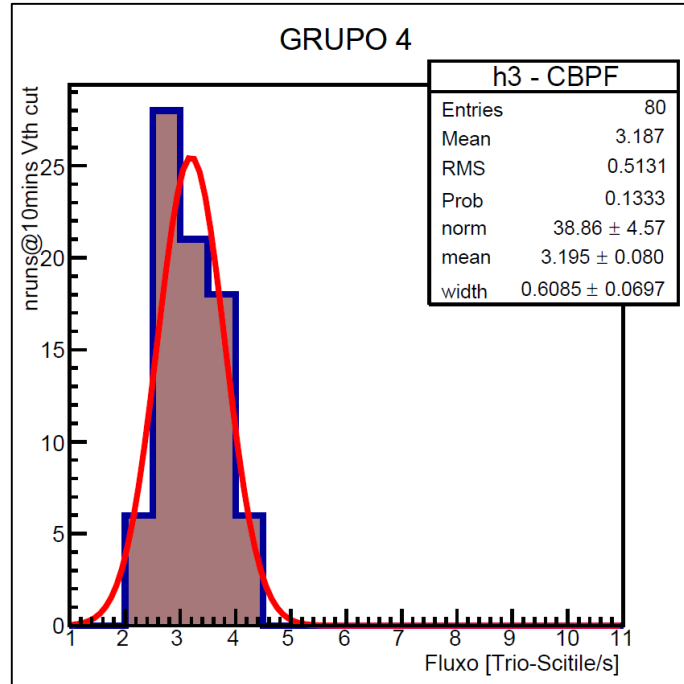


Figura 46 – Histograma de fluxo no Brasil para o grupo de coincidência 4

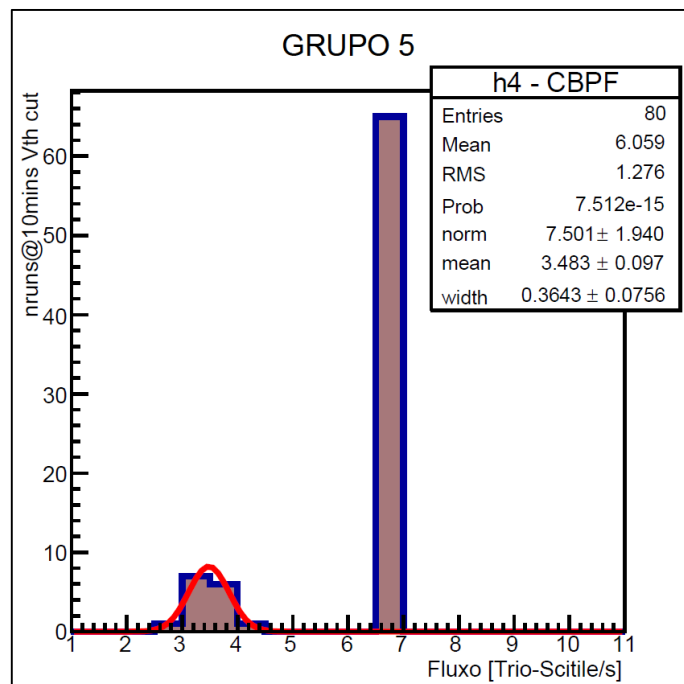


Figura 47 – Histograma de fluxo no Brasil para o grupo de coincidência 5

Em seguida, produziu-se a análise gráfica tipo *box plot* simplificado vista na Figura 48 para observação geral do fluxo na área sensível de *threshold*, ou seja, para a integração das medidas de fluxo observadas nos 80 períodos de aquisição de 10 minutos cada em que o valor de *threshold* foi superior a 130 [unidades internas de *threshold*]  $\approx 1,019$  Volts.

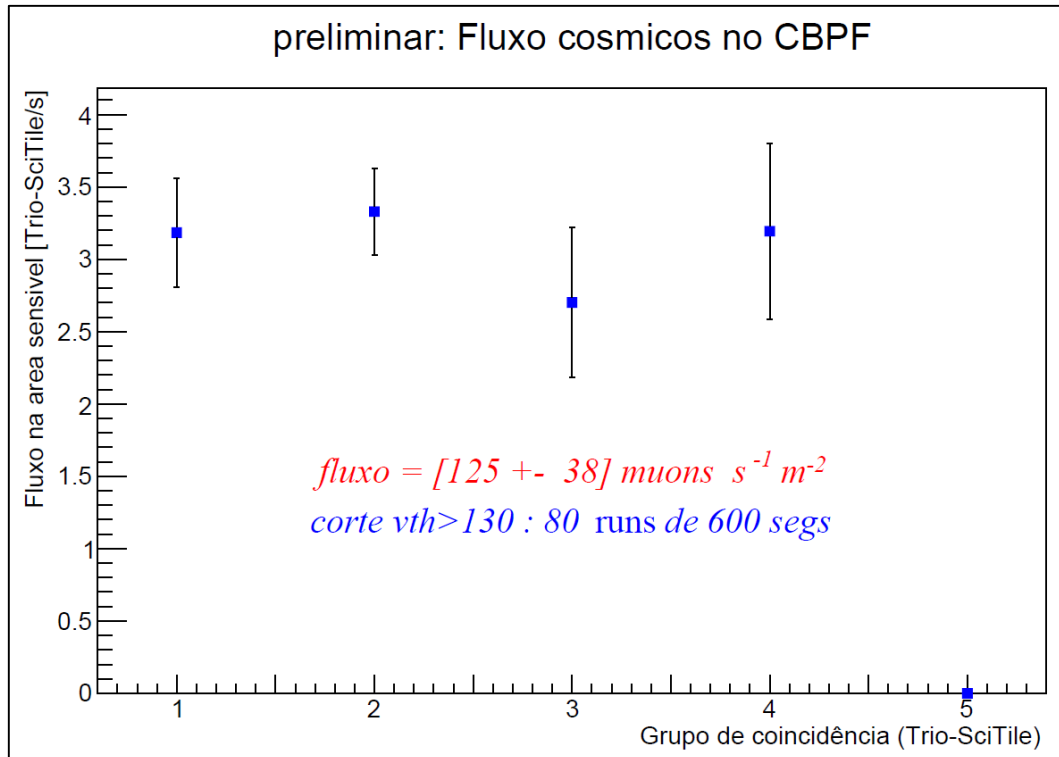


Figura 48 – Fluxo observado no CBPF

A análise gráfica destes histogramas nos indica claramente que há um padrão normal de fluxo nos grupos 1, 2, 3 e 4 e que há presença de ruído no quinto grupo, na faixa de 6,5 eventos por segundo. A aquisição de cinco medidas independentes objetivou justamente a percepção do efeito sistemático na medida, e deixa claro que há um problema no grupo 5. Entretanto, o experimento foi dimensionado para operar satisfatoriamente mesmo que dois grupos de coincidência necessitem ser descartados, e não houve tempo hábil para solucionar a presença de ruído no grupo 5. Deu-se então como satisfatório o resultado deste teste e prosseguiu-se com sua montagem do experimento.

Após seu transporte e instalação na posição definitiva, no módulo Criosfera I, os dados obtidos durante a calibração de *threshold* do experimento também foram tratados estatisticamente com auxílio do software *ROOT* e geraram os histogramas análogos aos anteriores, que podem ser vistos nas Figuras 49, 50, 51, 52 e 53.

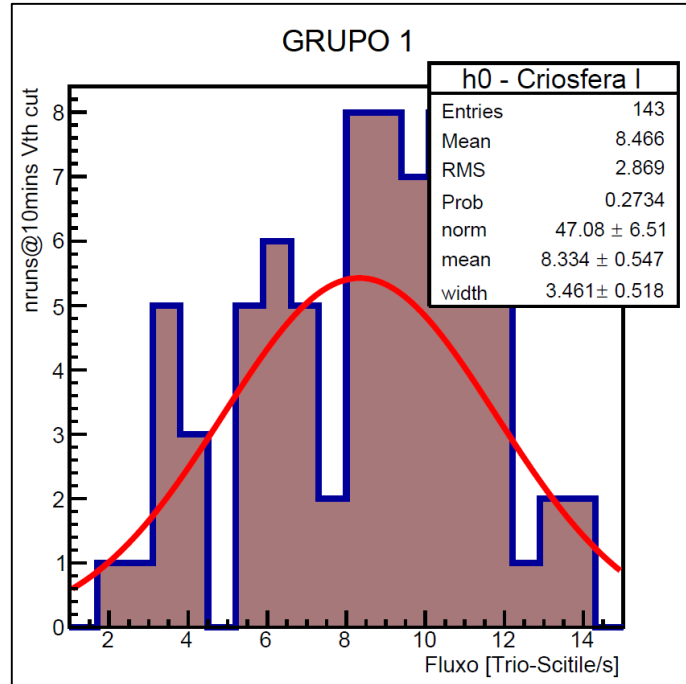


Figura 49 – Histograma de fluxo na Antártica para o grupo de coincidência 1

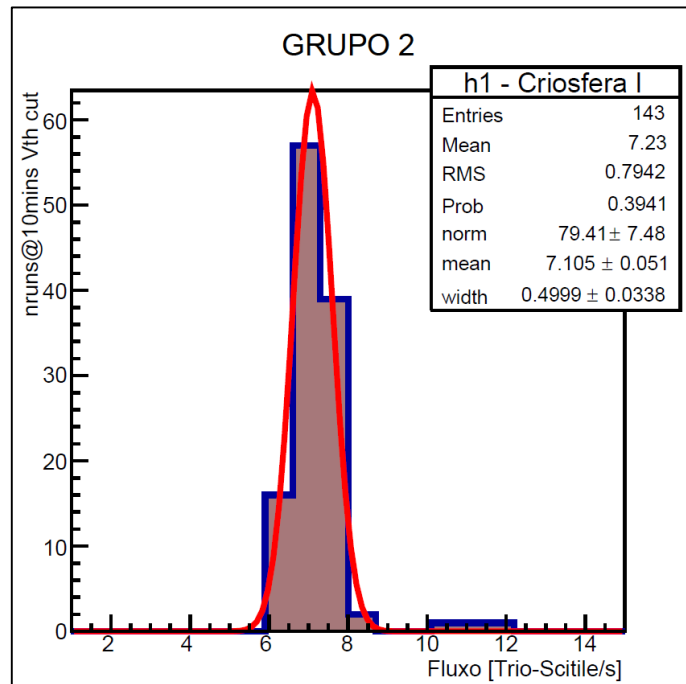


Figura 50 – Histograma de fluxo na Antártica para o grupo de coincidência 2

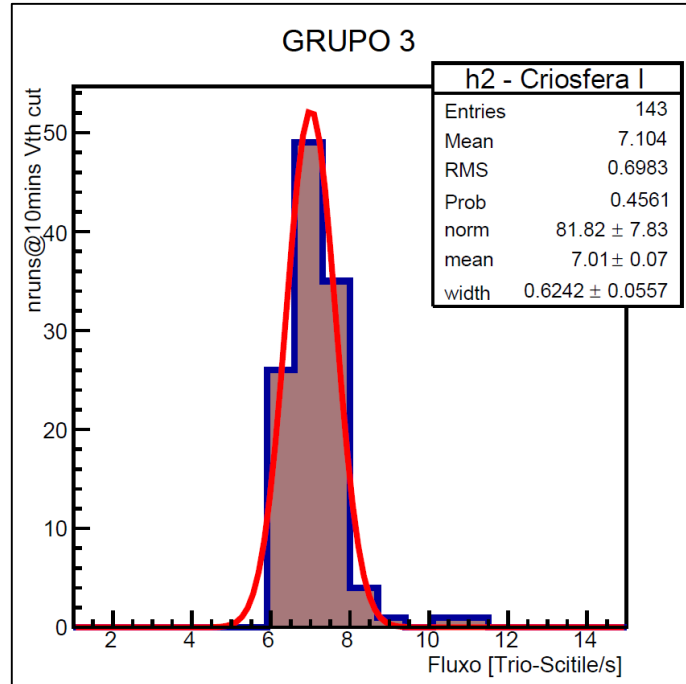


Figura 51 – Histograma de fluxo na Antártica para o grupo de coincidência 3

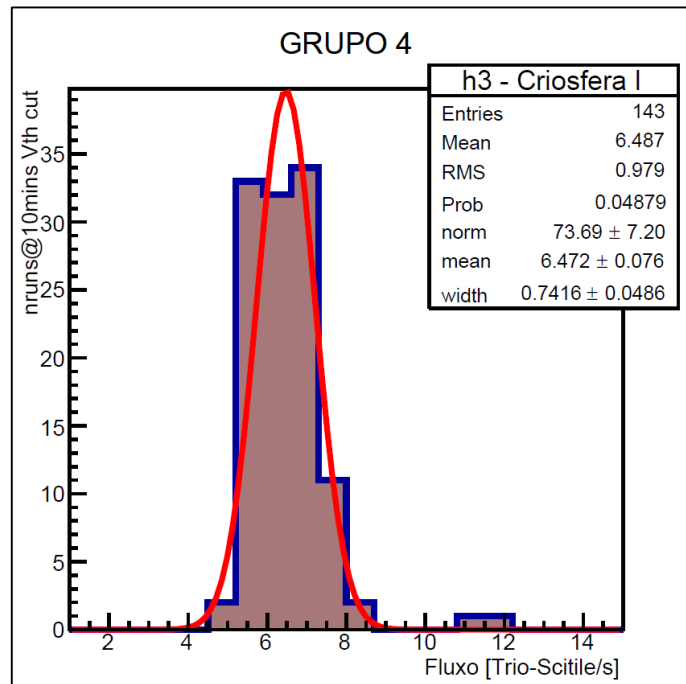


Figura 52 – Histograma de fluxo na Antártica para o grupo de coincidência 4



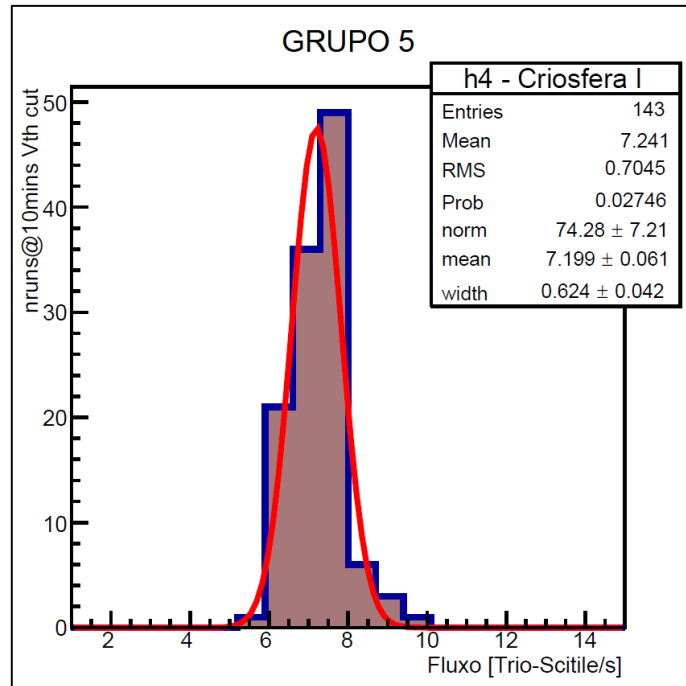


Figura 53 – Histograma de fluxo na Antártica para o grupo de coincidência 5

Em seguida, também se produziu a análise gráfica tipo *box plot* simplificado vista na Figura 54, e análoga à anterior, para observação geral do fluxo na área sensível de *threshold* nos 143 períodos de aquisição de 10 minutos cada.

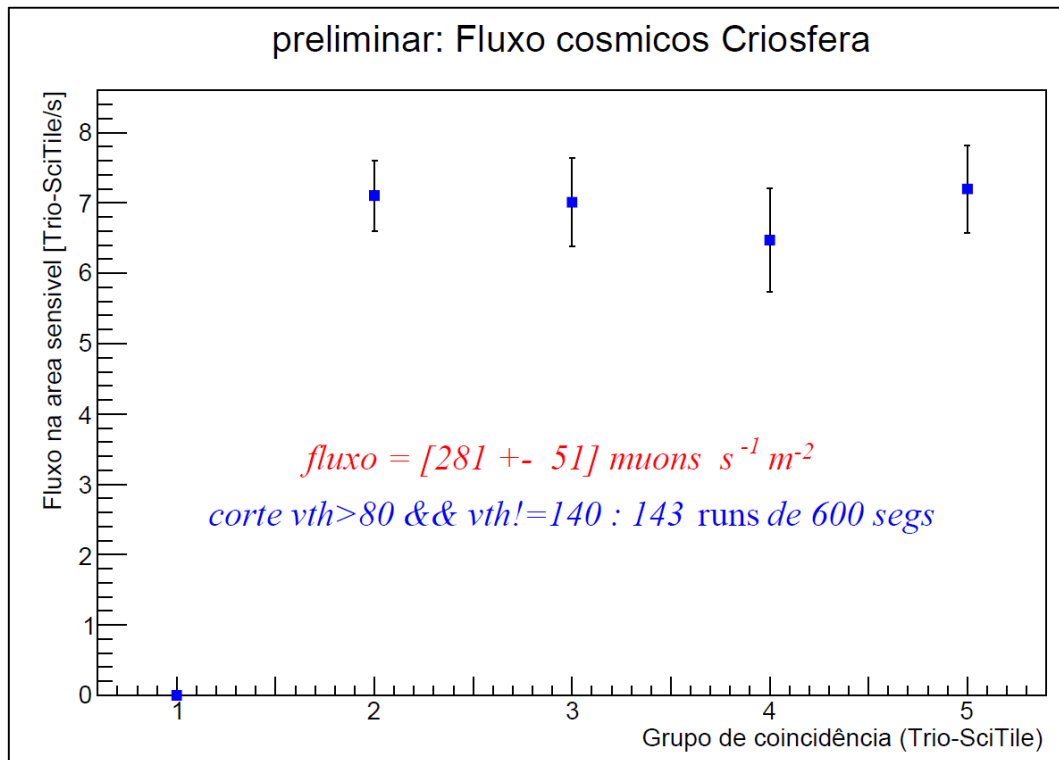


Figura 54 – Fluxo observado na Antártica

A análise gráfica destes histogramas nos indicou que após a desmontagem, transporte e remontagem do experimento houve alteração dos resultados sendo observado um padrão normal de fluxo nos grupos 2, 3, 4 e 5 e presença de ruído de natureza aleatória no primeiro grupo. A aquisição de cinco medidas independentes mais uma vez nos permite a percepção do efeito sistemático na medida, deixando claro que há um problema no grupo 1. Entretanto, deu-se então como satisfatório o resultado deste teste, validando a montagem do experimento.

Observou-se ainda, conforme esperado, que a região do círculo polar antártico apresenta fluxo cósmico superior àquele observado no Brasil. Nossas medições preliminares apontaram, para as calibrações, os seguintes fluxos de múons:

- Brasil, laboratório no 5º andar do edifício César Lattes (CBPF), setembro de 2014:
  - » Fluxo =  $[125 \pm 38]$  múons  $s^{-1} m^{-2}$
- Antártica,  $84^{\circ}00'00''S / 79^{\circ}29'39''W$  (Criosfera I), em dezembro de 2014:
  - » Fluxo =  $[281 \pm 51]$  múons  $s^{-1} m^{-2}$

Estas medições preliminares indicam o funcionamento apropriado do experimento.

## 5. Conclusão

O projeto de um experimento de monitoração de raios cósmicos para a Criosfera antártica demandou a utilização de métodos e técnicas multidisciplinares, envolvendo desde o projeto de circuitos digitais de alta velocidade em *FPGA* até a integração de circuitos eletrônicos de potência com sistemas de controle de malha fechada. Muitos imprevistos e adversidades foram experimentados durante a missão de instalação do projeto, promovendo sua adaptação às condições adversas encontradas no ambiente antártico. Algumas adaptações acabaram sendo incorporadas de maneira definitiva como, por exemplo, o módulo de controle de carga que inicialmente não havia sido previsto.

Os resultados obtidos com este experimento foram considerados satisfatórios desde o início, observando-se pleno funcionamento do mesmo em todas as fases do projeto. Não houve retrabalho significativo em nenhum de seus módulos.

As medidas de fluxo obtidas em ambiente antártico, quando comparadas com as medidas de fluxo obtidas em laboratório no Brasil, e com a expectativa teórica de fluxo polar indicam efetivo funcionamento do experimento, dando credibilidade aos resultados que serão obtidos no próximo ano.

Infelizmente o sistema de geração eólica de energia do módulo Criosfera I, cuja manutenção é de responsabilidade do INCT da Criosfera e do Instituto Nacional de Pesquisas Espaciais, não funcionou a contento inviabilizando o funcionamento ininterrupto do experimento. O sistema de comunicação por satélite presente no módulo Criosfera I também se mostrou bastante aquém das necessidades de transmissão de dados deste projeto.

Maiores informações a respeito da missão de instalação deste projeto, como seu detalhamento logístico e resultados dos demais projetos podem ser vistas no relatório de missão Criosfera 2014/2015, que se encontra anexado a este documento.

### 5.1. Continuidade do Projeto *CRE@AT*

Para as fases seguintes do projeto *CRE@AT* é urgente a solução definitiva dos problemas com geração de energia eólica no módulo Criosfera I. Também é urgente ao aprimoramento do experimento o estabelecimento de comunicação TCP/IP através do sistema

de comunicação por satélite presente no módulo de pesquisas. Sugere-se para tal a aquisição do sistema *IRIDIUM Pilot* [20].

Foi instalado no Criosfera I um *CRATE VME* e toda a infraestrutura de fontes de alimentação necessária ao desenvolvimento das demais fases do projeto *CRE@AT*. Devido à inexistência de comunicação bidirecional via satélite no Criosfera I, o módulo de alimentação elétrica não foi totalmente implementado pois ficou pendente a integração dos sensores de tensão e corrente com o sistema de comunicação. O desenvolvimento desta interface é necessário para o total controle remoto do experimento.

O hardware do módulo de contagem e dos módulos de aquisição e controle de carga foram utilizados com folga deixando, portanto, a possibilidade de que sejam aproveitados pelas demais fases do projeto *CRE@AT* bastando que apenas sua programação seja alterada.

## 5.2. Citações do Projeto na Mídia

Até a data do fechamento desta monografia este projeto havia sido citado em algumas páginas de notícias de agências públicas e sites especializados:

- Portal Brasil – Ciência e Tecnologia: Detector de raios cósmicos, instalado na Antártica, permitirá aprofundar pesquisas. (Figura 55)

*<http://www.brasil.gov.br/ciencia-e-tecnologia/2015/01/detector-de-raios-instalado-na-antartica-permitira-aprofundar-pesquisas%20>*



Figura 55 – Captura de Tela do Portal Brasil

- Portal CBPF – Notícias: CBPF vai ao Polo Sul. (Figura 56)

<http://portal.cbpf.br/noticia/cbpf-vai-ao-polo-sul/755>

The image is a screenshot of a web browser displaying a news article on the CBPF website. The browser's address bar shows the URL [portal.cbpf.br/noticia/cbpf-vai-ao-polo-sul/755](http://portal.cbpf.br/noticia/cbpf-vai-ao-polo-sul/755). The website header includes the CBPF logo and the text 'Centro Brasileiro de Pesquisas Físicas'. A navigation menu at the top contains links for 'BRASIL', 'Acesso à informação', 'Participe', 'Serviços', 'Legislação', and 'Canais'. A left sidebar lists various site sections like 'Quisa', 'Graduação', 'Idores', 'oteca', 'bratórios Multiusuários', 'gos Publicados', and 'riedade Intelectual'. The main content area features the article title 'CBPF vai ao Polo Sul' with a sub-header 'Postado em: 18/09/2014'. Below the title is a photograph of a red and white scientific structure on a snowy, icy landscape. The article text describes the 'Cosmic Ray Experiment at Antartida (CRE@AT)' project, its location at the 'Estação Criosfera 1' in Antarctica, and the role of student Ulisses de Freitas Carneiro da Graça. It also mentions the involvement of the ProAntar program and the CERN organization.

**CBPF vai ao Polo Sul**

Postado em: 18/09/2014



O Centro Brasileiro de Pesquisas Físicas (CBPF) desenvolve o projeto *Cosmic Ray Experiment at Antartida (CRE@AT)* na Coordenação de Física Experimental de Altas Energias do CBPF (LAFEX). Trata-se de missão para instalar a primeira fase de um experimento de raios cósmicos na base brasileira Criosfera 1, na Antártica, localizada a 670 quilômetros do polo sul geográfico.

Este trabalho faz parte do desenvolvimento de detectores de partículas para medidas de fluxo de raios cósmicos para estudar possíveis correlações entre raios cósmicos e parâmetros climáticos, como a cobertura de nuvens na Terra durante um ciclo solar, a formação de nuvens, atividade solar e composição atmosférica.

A instalação do experimento no Polo Sul será realizada pelo aluno de iniciação científica Ulisses de Freitas Carneiro da Graça.

De tão frio, o interior da Antártica só pode ser visitado durante o mês de dezembro. Um dos maiores desafios é projetar um experimento que seja operado de forma autônoma, remota e com baixo consumo de energia elétrica.

O mais frio dos continentes tem temperatura mínima de 75°C negativos no inverno, e o fluxo de radiação cósmica tem um padrão diferenciado e efetivo, já que as linhas do campo magnético terrestre ficam ortogonais às existentes nos trópicos. Além disso, o Polo Sul tem características especiais, como a ausência de elementos pesados no solo - cuja espessura da camada de gelo mede um quilômetro.

**CBPF, Cern e ProAntar: soma de competência científica**

Pela experiência e flexibilidade institucional do CBPF, o pesquisador André Massafferri foi convidado a participar do Programa Antártico Brasileiro (ProAntar) com o intuito de desenvolver a área de física de altas energias na estação Criosfera 1. Após conversas com o professor da Universidade do Estado do Rio de Janeiro (UERJ) Heitor Evangelista da Silva, representante da estação, destacou-se a necessidade da instalação de um sistema robusto para a realização de medidas precisas de raios cósmicos.

Do ponto de vista experimental, o convite do ProAntar surge em um momento oportuno, no qual o grupo de altas energias está envolvido em projetos de instrumentação de alta tecnologia, em particular no desenvolvimento do experimento LHCb, localizado no Organização Europeia para a Pesquisa Nuclear (CERN). A aplicação de novas técnicas, como as utilizadas neste projeto, pode ter consequências interessantes para a transferência de tecnologia para o Brasil.

\*\*\*\*\*

Núcleo de Comunicação Social - NCS/CCI/CBPF  
E-mail: [ncs\\_cbpf@cbpf.br](mailto:ncs_cbpf@cbpf.br)  
Ramais: 7291 / 7114

Figura 56 – Captura de Tela do Portal CBPF

- Associação Brasileira das Instituições de Pesquisa Tecnológica e Inovação – Agência Gestão CT&I: Criosfera recebe detector de raios cósmicos para aprofundar pesquisas na Antártica (Figura 57)

*[http://www.agenciacti.com.br/index.php?option=com\\_content&view=article&id=6737:criosfera-recebe-detector-de-raios-cosmicos-para-aprofundar-pesquisas-na-antartica&catid=1:latest-news](http://www.agenciacti.com.br/index.php?option=com_content&view=article&id=6737:criosfera-recebe-detector-de-raios-cosmicos-para-aprofundar-pesquisas-na-antartica&catid=1:latest-news)*

The screenshot shows a web browser window displaying the website of the Agência Gestão CT&I. The page title is "Criosfera recebe detector de raios cósmicos para aprofundar pesquisas na Antártica". The article is dated "TER, 13 DE JANEIRO DE 2015 15:09" and was written by "AGÊNCIA GESTÃO CT&I". The text describes the installation of the CRE@AT detector at the Criosfera 1 research station in Antarctica, aimed at studying cosmic rays and their impact on climate change. It mentions the collaboration between the CBPF, the University of Rio de Janeiro, and the INCT Criosfera, with support from the CERN. The article also includes a quote from researcher André Massafferri and a link for more information.

Últimas: Portal oferece explicações sobre a energia heliotérmica Quinta, 21 Mai 2015 10:14:11 AM AM

Principal SNCTI Oportunidades de fomento Análise de especialistas Seção Entrevista Gestão C&T online CT&I Multimídia Expediente

. C&T Inovação . C&T Economia . C&T Gestão . C&T Internacional . C&T Meio ambiente . C&T Educação . C&T Saúde . C&T Política . C&T Sistemas estaduais . Agenda . Mídia Kt .

## Criosfera recebe detector de raios cósmicos para aprofundar pesquisas na Antártica

C&T Inovação - BR

TER, 13 DE JANEIRO DE 2015 15:09 | ESCRITO POR AGÊNCIA GESTÃO CT&I

O módulo de pesquisa Criosfera 1, construído pelo Brasil no centro da Antártica, adquiriu um detector de raios cósmicos apelidado de CRE@AT (Cosmic Rays Experiment at Antarctica, na sigla em inglês). O objetivo é medir o efeito desses elementos na superfície da Terra e entender melhor as mudanças climáticas na Antártica.

Os raios cósmicos são partículas energéticas que se deslocam a velocidades próximas à da luz (299.792.458 metros por segundo) e fornecem aos cientistas informações importantes sobre as condições físicas do sistema solar e do universo. As características da Antártica, como ter um tamanho reduzido da atmosfera em relação a regiões fora dos polos e a ausência de crosta terrestre facilitam os estudos destas partículas.

A iniciativa é resultado da parceria entre o Centro Brasileiro de Pesquisas Físicas (CBPF), a Universidade do Estado do Rio de Janeiro (Uerj) e o Instituto Nacional de Ciência e Tecnologia da Criosfera (INCT Criosfera). O CBPF conta, ainda, com a colaboração da Cern (Organização Europeia para a Pesquisa Nuclear, na sigla em francês).

De acordo com o pesquisador do departamento de Física Experimental de Altas Energias do CBPF, André Massafferri, as variações no sistema antártico têm grandes implicações em várias partes do mundo, o que serve como uma espécie de termômetro das mudanças climáticas. Segundo ele, estudos recentes, ainda não conclusivos, indicam uma relação entre o clima e os raios cósmicos, tendo por princípio a formação de nuvens partindo de um ponto inicial, seja uma partícula de poeira mineral terrestre ou cósmica.

"Um dos objetos do estudo será justamente a relação de formação de nuvens com o clima e com partículas cósmicas", explica o cientista.

Saiba mais sobre o módulo de pesquisa [neste link](#).

(Agência Gestão CT&I, com informações d MCTI)

Tweet Share Curtir 0 Share 0

Figura 57 – Captura de Tela do Portal Agência CT&I

- Portal Inovação Tecnológica: Detector de raios cósmicos é instalado na Antártica (Figura 58)

<http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=detector-raios-cosmicos-instalado-antartica&id=010175150118#.VV3fprmrRQK>



Figura 58 – Captura de Tela do Portal Inovação Tecnológica



A missão Criosfera 2014 foi encerrada com sucesso no dia 28/12/2014. A Figura 59 é a fotografia oficial da missão.



Figura 59 – Fotografia oficial da missão Criosfera 2014/2015

## Referências Bibliográficas

- [1] A. Izquierdo, "Astromart News - Ultrahigh Energy Cosmic Rays Found Emanating From Hotspot Near Ursa Major," [Online]. Available: [http://www.astromart.com/news/news.asp?news\\_id=1425](http://www.astromart.com/news/news.asp?news_id=1425). [Acesso em 21 05 2015].
- [2] F. Arnold, "Ion nucleation - a potential source for stratospheric aerosols," *Nature*, no. 299, p. 134, 1982.
- [3] J.-K. Z. a. F. Y. R.P. Turco, "A new source of tropospheric aerosols: ion-ion," *Geophys. Res. Lett.*, no. 25, p. 635, 1998.
- [4] F. Y. a. R. Turco, "Ultrafine aerosol formation via ion-mediated nucleation," *Geophys. Res. Lett.*, no. 27, p. 883, 2000.
- [5] F. Marcastel, "CERN CLOUD research team adds new pieces to puzzle of cloud formation," 25 08 2011. [Online]. Available: <http://phys.org/news/2011-08-cern-cloud-team-pieces-puzzle.html>. [Acesso em 19 05 2015].
- [6] "CLOUD: AN ATMOSPHERIC RESEARCH FACILITY AT CERN," *CERN/SPSC 2000*, nº 041, pp. 01-15, October 17, 2000.
- [7] M. B. Baker, "Cloud Microphysics and Climate," *SCIENCE*, vol. 276, pp. 1072-1078, 1997.
- [8] K. S. Carslaw, R. G. Harrison and J. Kirkby, "Cosmic Rays, Clouds, and Climate," *SCIENCE*, vol. 298, pp. 1732-1737, 2002.
- [9] M. V. Medvedev and A. L. Melott, "Do extragalactic cosmic rays induce cycles in fossil diversity?," Department of Physics and Astronomy, University of Kansas, KS 66045, 2007.
- [10] C. Faria, "Antártida - Continente Gelado - Geografia - InfoEscola," Info Escola, [Online]. Available: <http://www.infoescola.com/geografia/antartica-antartida/>. [Acesso em 21 05 2015].

- [11] “Raio cósmico – Wikipédia, a enciclopédia livre,” [Online]. Available: [http://pt.wikipedia.org/wiki/Raio\\_c%C3%B3smico](http://pt.wikipedia.org/wiki/Raio_c%C3%B3smico). [Acesso em 21 05 2015].
- [12] P. A. Observatory, “Pierre Auger Observatory - Home Page,” [Online]. Available: <https://www.auger.org/>. [Acesso em 27 05 2015].
- [13] “IceCube Neutrino Observatory,” [Online]. Available: <https://icecube.wisc.edu/>. [Acesso em 27 05 2015].
- [14] “AMS-02,” [Online]. Available: <http://www.ams02.org/>. [Acesso em 27 05 2015].
- [15] C. C. Conti, “Cintiladores. Instrumentação Nuclear,” [Online]. Available: <http://www.ccconti.com/Instnuc/Aula/Aula9.pdf>. [Acesso em 5 Dezembro 2013].
- [16] K. C. Japan, “Plastic Scintillating Fibers - kuraray | Wavelength Shifting Fibers,” KURARAY Co. Japan, [Online]. Available: <http://kuraraypsf.jp/psf/ws.html>. [Accessed 24 02 2015].
- [17] “Fotomultiplicador - Wikipedia, la enciclopedia libre,” Wikipedia, [Online]. Available: <http://es.wikipedia.org/wiki/Fotomultiplicador>. [Acesso em 15 05 2015].
- [18] L. Guedes, “DESENVOLVIMENTO DE DETECTOR DE TIRAS DE CINTILADORES PLÁSTICOS PARA TRIGGER DO SISTEMA DE TRACKING DE PARTÍCULAS ELEMENTARES,” CEFET/RJ, RIO DE JANEIRO, 2013.
- [19] t. f. e. Wikipedia, “VMEbus - Wikipedia, the free encyclopedia,” [Online]. Available: <http://en.wikipedia.org/wiki/VMEbus>. [Acesso em 08 março 2015].
- [20] “Iridium | Products | Iridium Pilot,” [Online]. Available: <https://www.iridium.com/products/Iridium-Pilot.aspx>. [Acesso em 27 05 2015].
- [21] I. Barbi, “Eletrônica de Potência,” Florianópolis, Ed. do Autor, 2005, pp. 57-58.

## APENDICE A: Problemas na geração de energia eólica do módulo Criosfera I

O módulo Criosfera 1 é dotado de um sistema híbrido (solar / eólico) para geração de energia elétrica. Entretanto, dadas as características únicas do inverno polar, o módulo permanece sem receber radiação solar entre os meses de abril a outubro, como pode ser visto no gráfico da Figura 60.

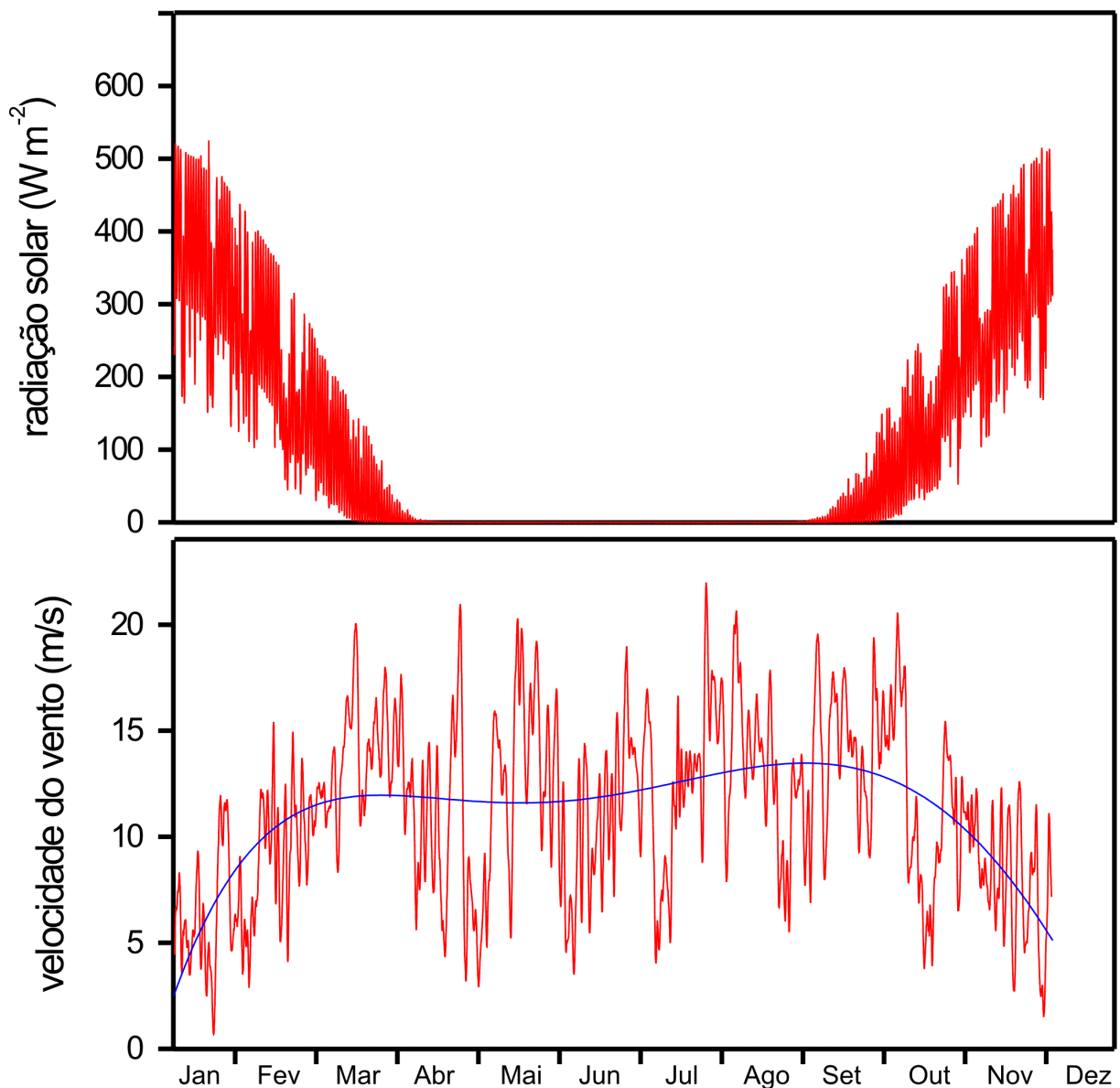


Figura 60 – Incidência solar e velocidade do vento no módulo Criosfera 1 em 2014

Portanto, nos períodos em que a radiação solar é insuficiente ou nula, o sistema eólico deve ser autossuficiente.

Devido ao total colapso da única turbina eólico-geradora que havia sido deixada em operação na missão do verão de 2013/2014<sup>5</sup> nossa missão previa a substituição dos quatro aero geradores do projeto original – e que até então haviam falhado – por aero geradores recém adquiridos. Todos os subsistemas relacionados à geração de energia do módulo eram até então baseados em dispositivos comerciais com algumas customizações de integração.

## **A.1. Problemas com as Turbinas FORGEN**

Os aero geradores *FORGEN Ventuz 70* adquiridos não foram previamente testados pela equipe brasileira, pois foram despachados diretamente do fornecedor (Reino Unido) para o galpão de facilidades logísticas da ALE em *Punta Arenas* (Chile) e, de lá, para o módulo Criosfera 1.

Segundo o fabricante a instalação dos aero geradores e de seus respectivos sistemas de controle seria um procedimento trivial, apenas a conexão de cabos conforme o diagrama fornecido. Entretanto observamos que não havia geração de energia para carga das baterias mesmo após a execução de todos os procedimentos listados e havendo, de acordo com a própria documentação do fabricante, oferta de vento suficiente para que esta se iniciasse.

Após investigação inicial observou-se que o controlador fornecido reduzia consideravelmente a rotação da turbina quando esta era conectada, fazendo com que a oferta de energia fosse reduzida.

## **A.2. Estudo do Problema**

Após vários contatos por telefone via satélite com o fabricante resultarem em tentativas frustradas de solução do problema decidimos abrir uma das quatro unidades de controle do gerador eólico, vista na Figura 61 para investigar seu circuito.

---

<sup>5</sup> O módulo CRIOSFERA 1 apresenta longo histórico de problemas no sistema eólico, tendo havido colapso total neste sistema em todos os invernos desde sua instalação (2012, 2013 e 2014).

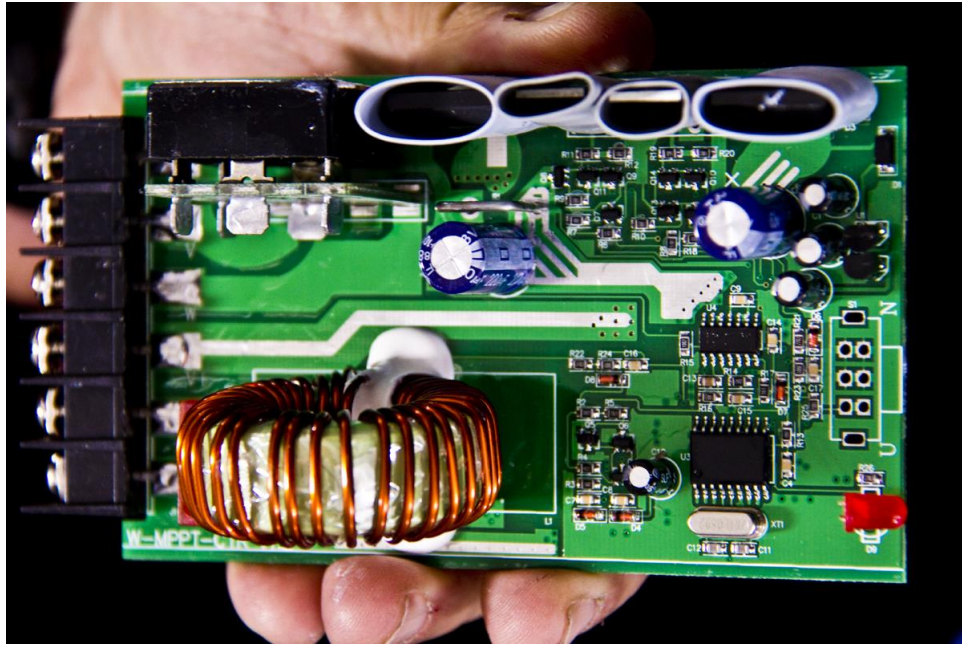


Figura 61 – Controlador de Carga fornecido com o gerador eólico VENTUZ 70

A ausência de documentação técnica sobre o controlador, a oferta limitada de bibliografia adicional e a ausência de conexão com a Internet tornaram a tarefa substancialmente complexa, mas, investigando as conexões elétricas da placa de circuito impresso, pudemos obter seu diagrama elétrico simplificado para – a partir dele – elaborar um *set-up* de testes que permitisse traçar um plano de intervenção. As Figuras 62, 63, 64 e 65 mostram, respectivamente, os diagramas elétricos do estágio de potência, de alimentação, de acionamento e, de forma simplificada, de controle do controlador de carga.

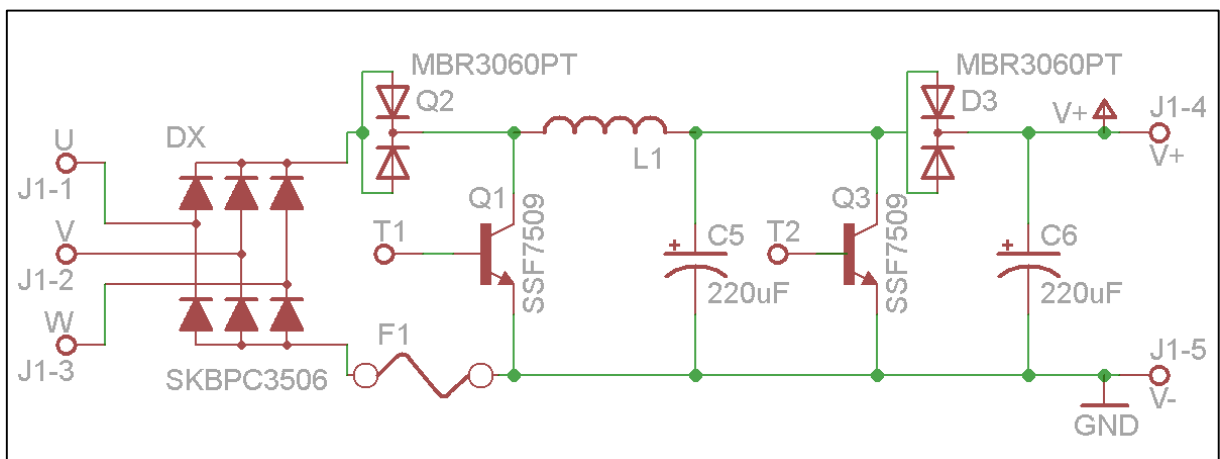


Figura 62 – Estágio de potência do controlador FORGEN

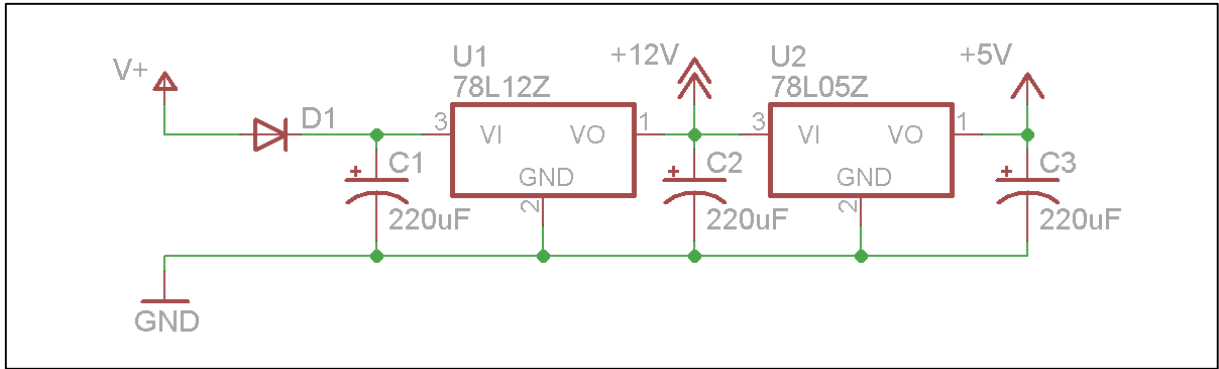


Figura 63 – Estágio de alimentação do controlador FORGEN

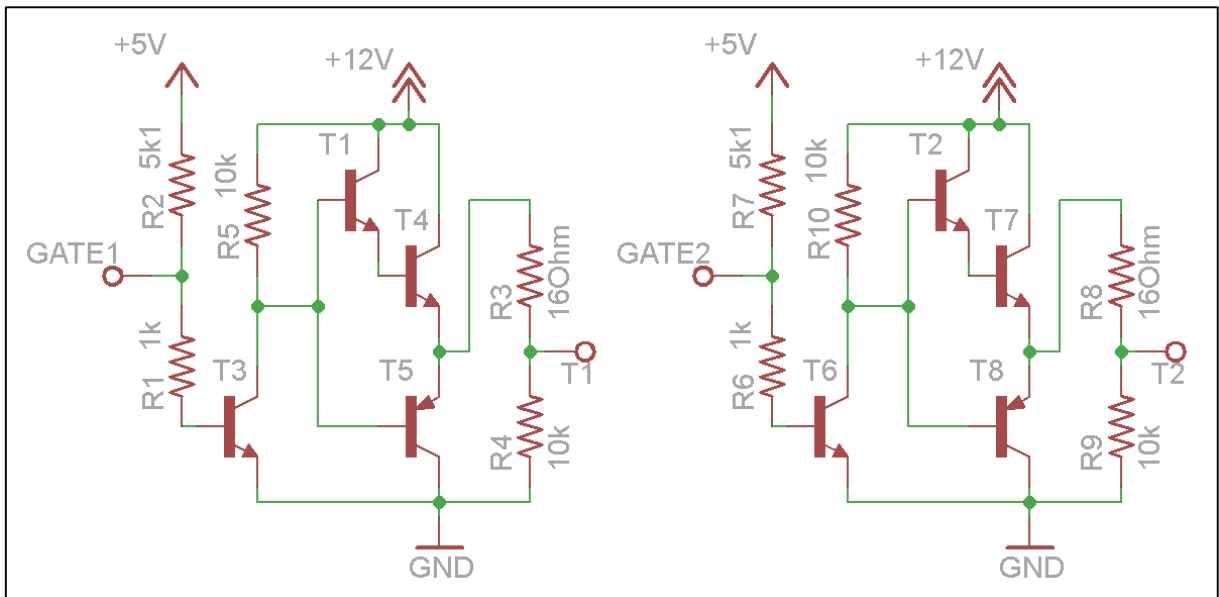


Figura 64 – Estágio de acionamento do controlador FORGEN

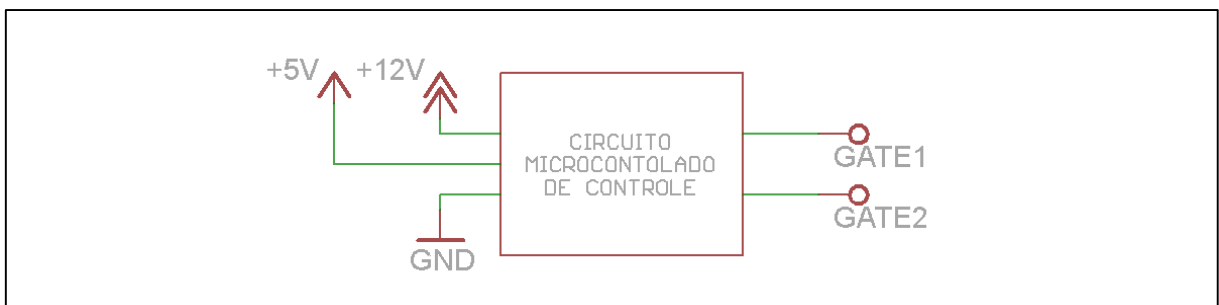


Figura 65 – Estágio de controle (simplificado) do controlador FORGEN

Inicialmente, por inspeção, pôde-se concluir que o estágio de potência deste controlador é composto de um retificador trifásico de onda completa (ponte de Graetz) [21] que alimenta

com corrente contínua um conversor elevador de tensão do tipo *Boost* formado pelos componentes L1, C5, Q3, D3 e C6. A topologia utilizada neste conversor utiliza o indutor como elemento armazenador de energia para elevar a tensão retificada proveniente da turbina e permitir a carga da bateria (24V no Criosfera 1) mesmo que a turbina esteja gerando níveis de tensão inferiores.

Concluiu-se também por inspeção que a função do transistor Q1 é drenar corrente do gerador e, conseqüentemente, frear por força contra eletromotriz a turbina eólica quando a geração de potência por esta é superior à necessária para carregar as baterias.

Os sinais de disparo (T1 e T2) observados num osciloscópio indicavam que o transistor Q1 não era acionado e que o transistor Q2 sofria acionamento modulado por largura de pulso em aproximadamente com frequência de 21kHz e ciclo ativo variando em loops muito bem definidos a cada 10 segundos, sugerindo que o circuito de controle era incapaz de estabelecer o ponto ideal de operação do estágio de potência. Observou-se também que a velocidade de rotação da turbina do aero gerador caía consideravelmente quando este era conectado ao circuito (não havia instrumentação disponível para mensurar esta queda).

### **A.3. Solução Proposta**

Decidiu-se então montar um *set-up* experimental com circuito de controle externo para estudar a influência da queda de velocidade de rotação da turbina na perda de potência do aero gerador. Estavam disponíveis uma placa de prototipagem micro controlada modelo “*Arduino MEGA2560*” e alguns poucos componentes para a montagem deste circuito.

Sabe-se que, pelas características de sua construção, a frequência da saída elétrica do aero gerador é diretamente proporcional à velocidade de rotação de sua turbina eólica (ainda que a razão desta relação fosse desconhecida). O circuito diagramado na Figura 66 foi então projetado com a finalidade de prover ao micro controlador um sinal que permitisse medir a frequência de saída do aero gerador. Este sinal foi conectado a uma entrada de interrupção externa do micro controlador programada com sensibilidade a bordo de descida.



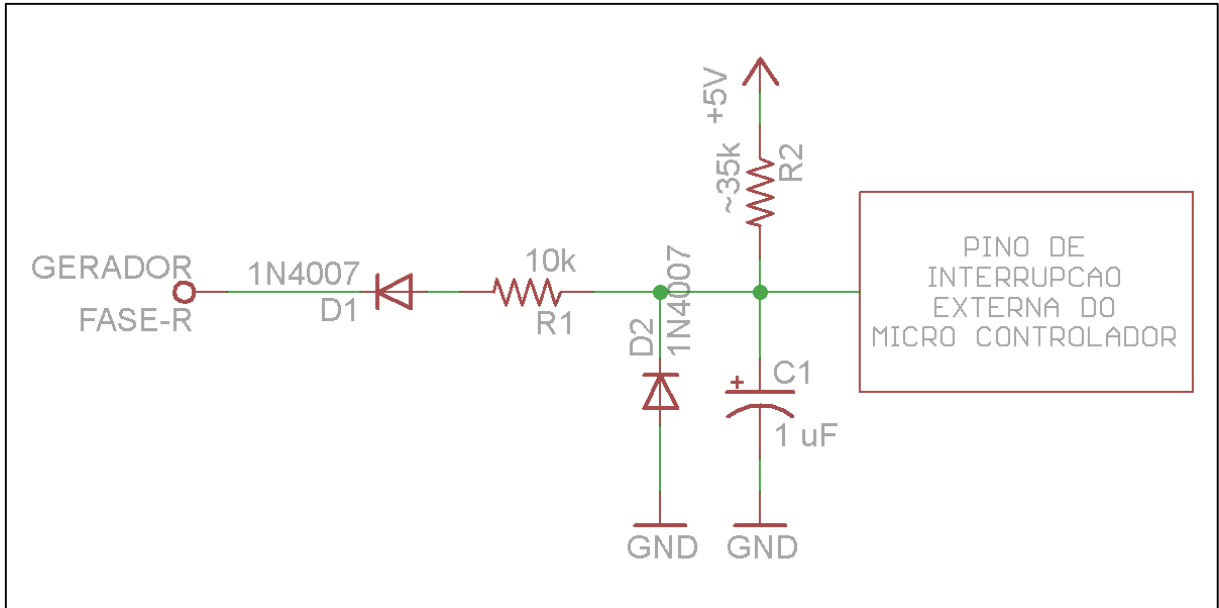


Figura 66 – Circuito para medição da velocidade de rotação da turbina

No âmbito da programação deste micro controlador foram implementados três dispositivos de controle do tipo PID (proporcional, integrativo e diferencial) para que se obtivesse controle sobre a carga da bateria (tensão máxima de flutuação), rotação mínima da turbina (quando em carga) e rotação máxima da turbina (para bateria em carga e completamente carregada). A Figura 67 ilustra as malhas de controle adotadas. O código C escrito para esta aplicação pode ser visto no Apêndice E.

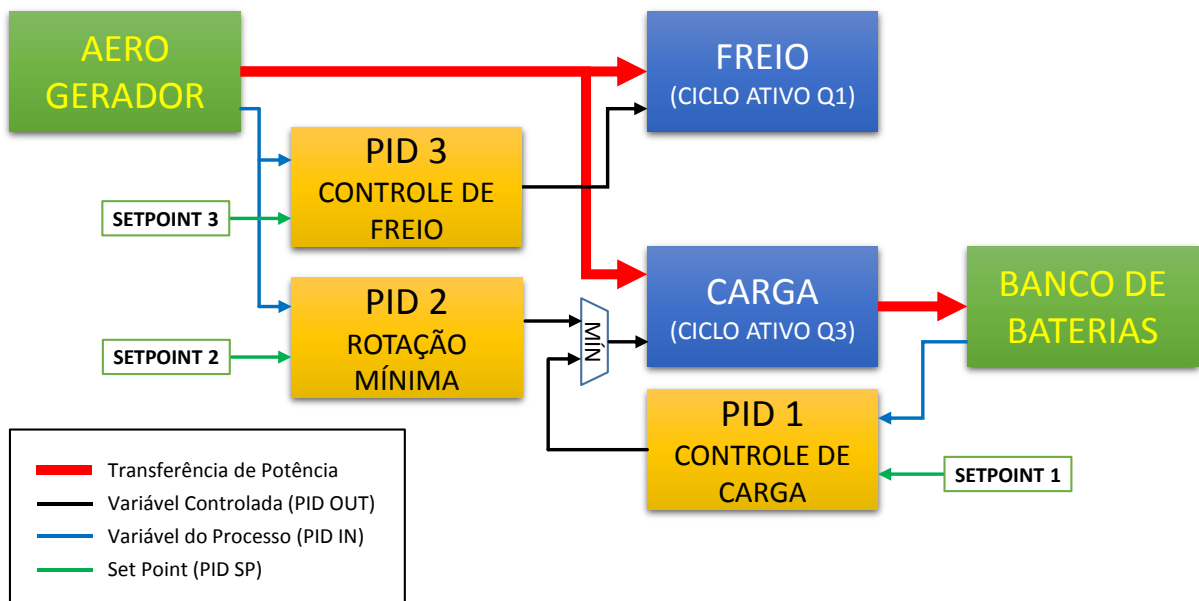
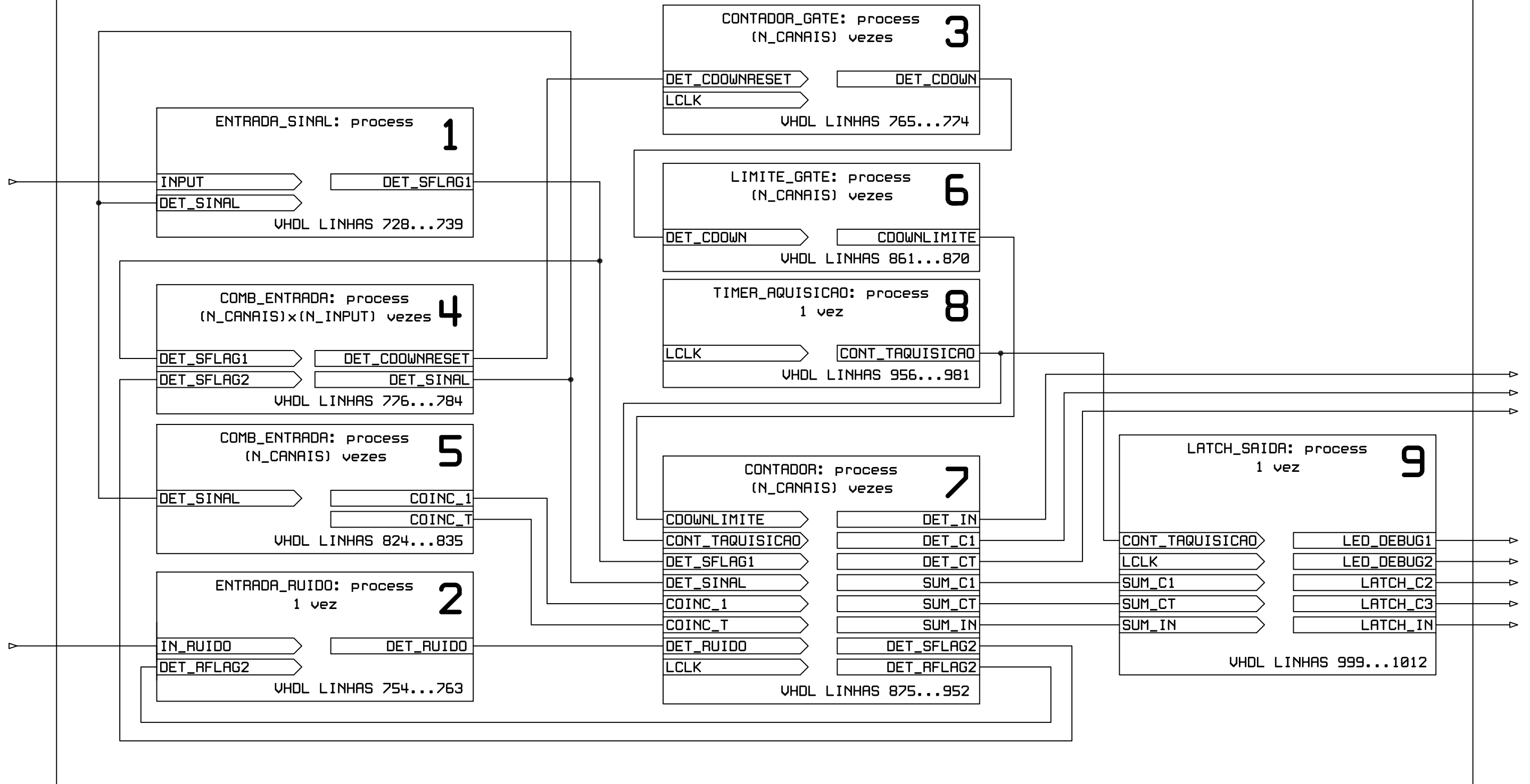


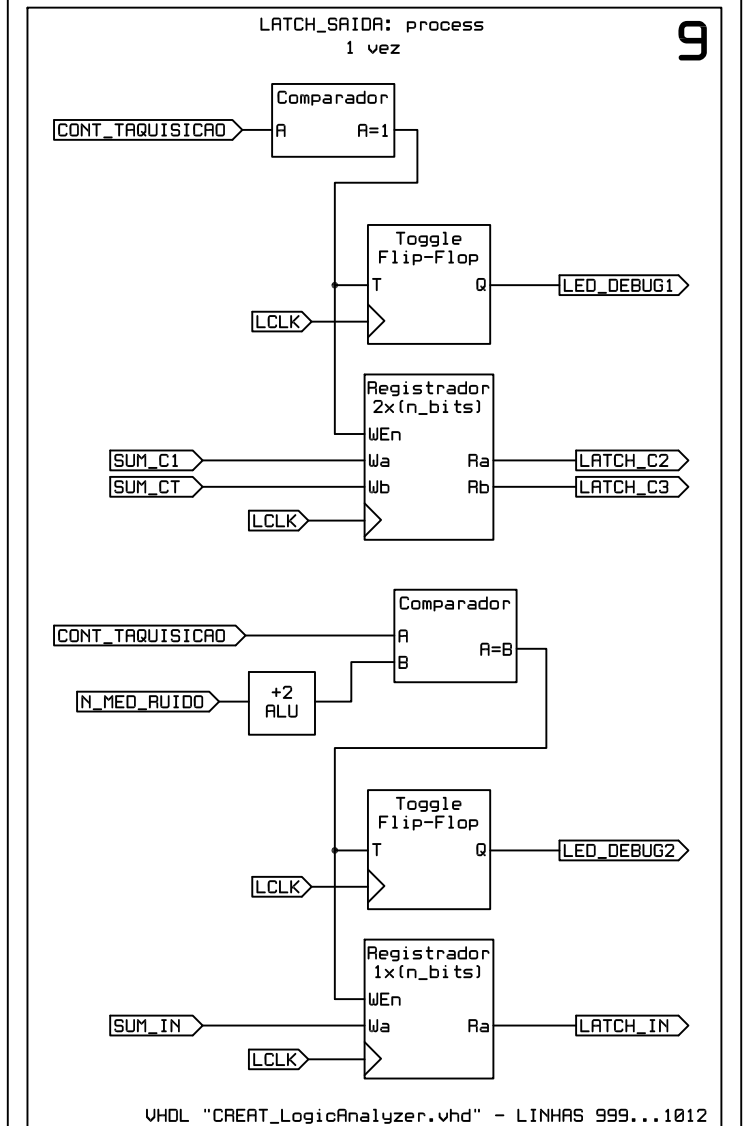
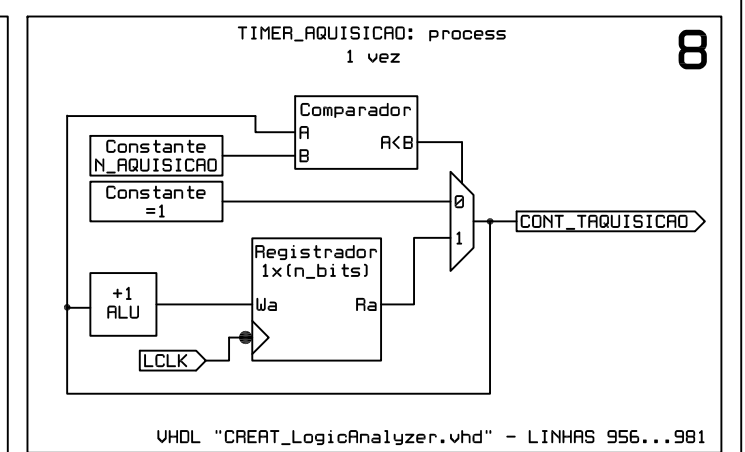
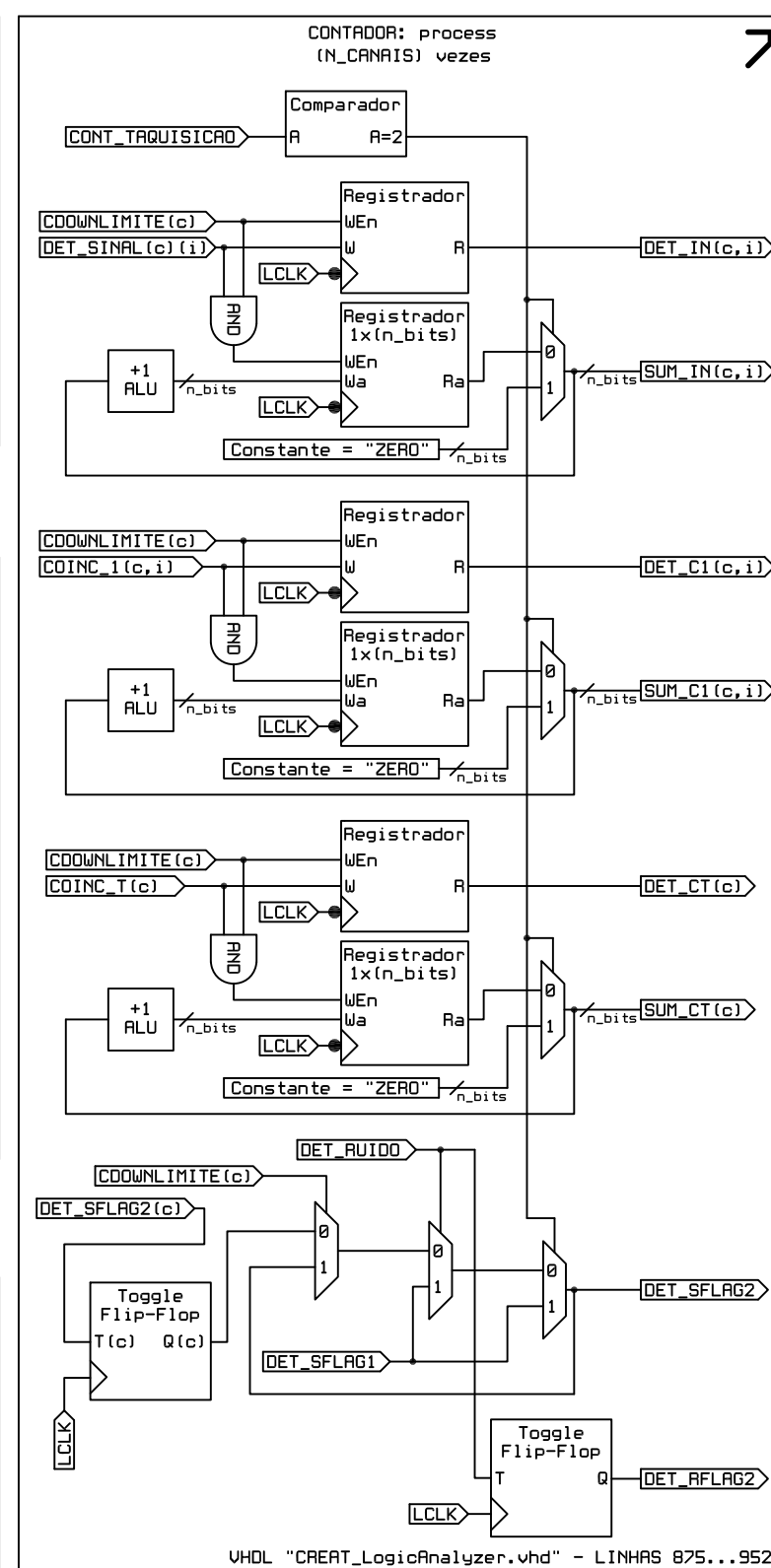
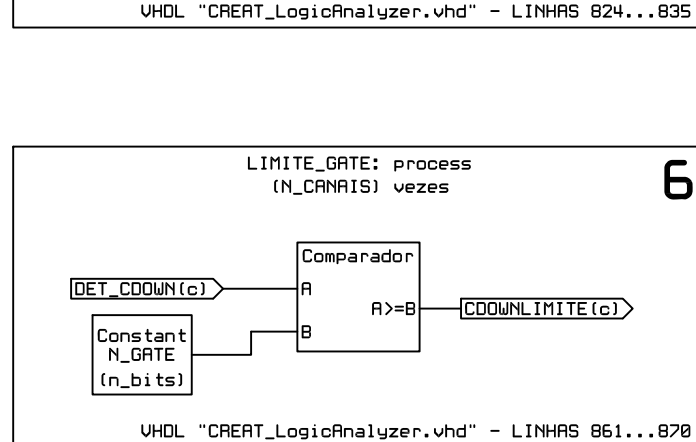
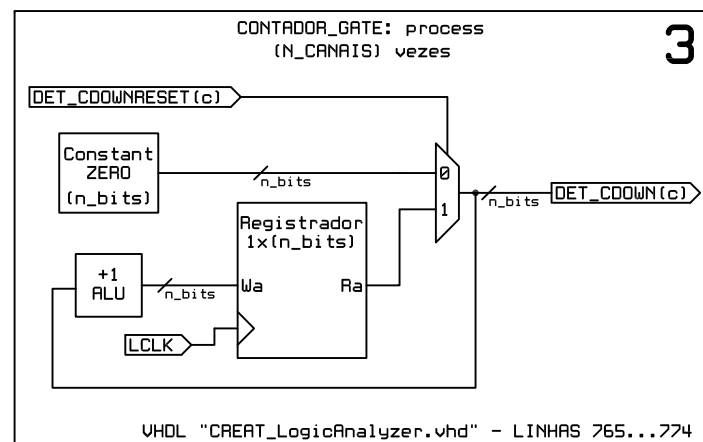
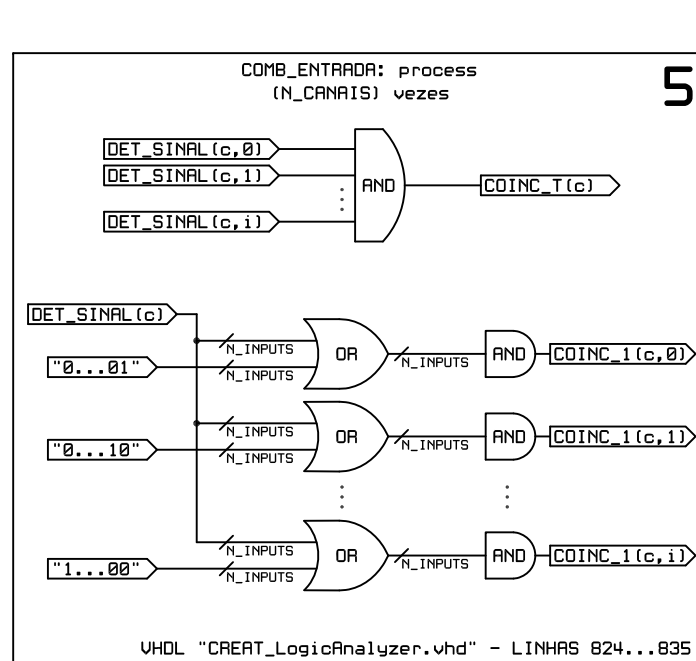
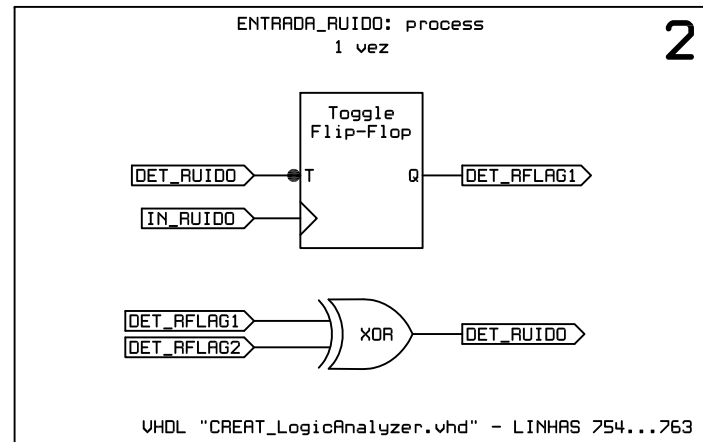
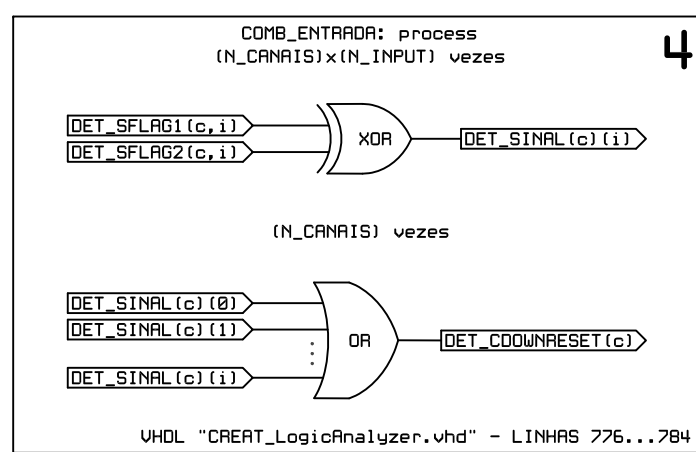
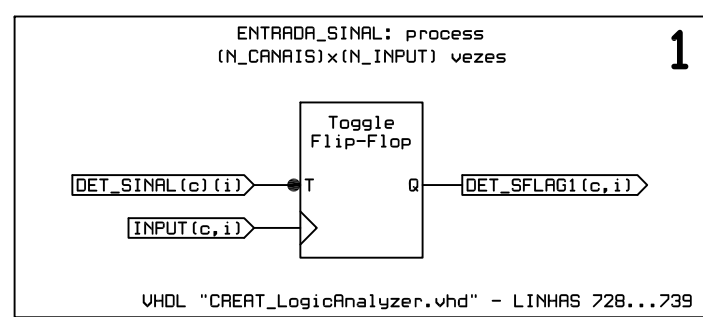
Figura 67 – Malhas de controle do circuito construído

Após ensaiar o aero gerador com o controlador de carga controlado pelo circuito experimental externo obteve-se em torno de 50% de aumento na potência de carga das baterias. Entretanto, ainda assim, esta encontrava-se muito aquém do esperado (para ventos de 10m/s obteve-se em torno de 12W numa turbina especificada para 50W nesta situação).

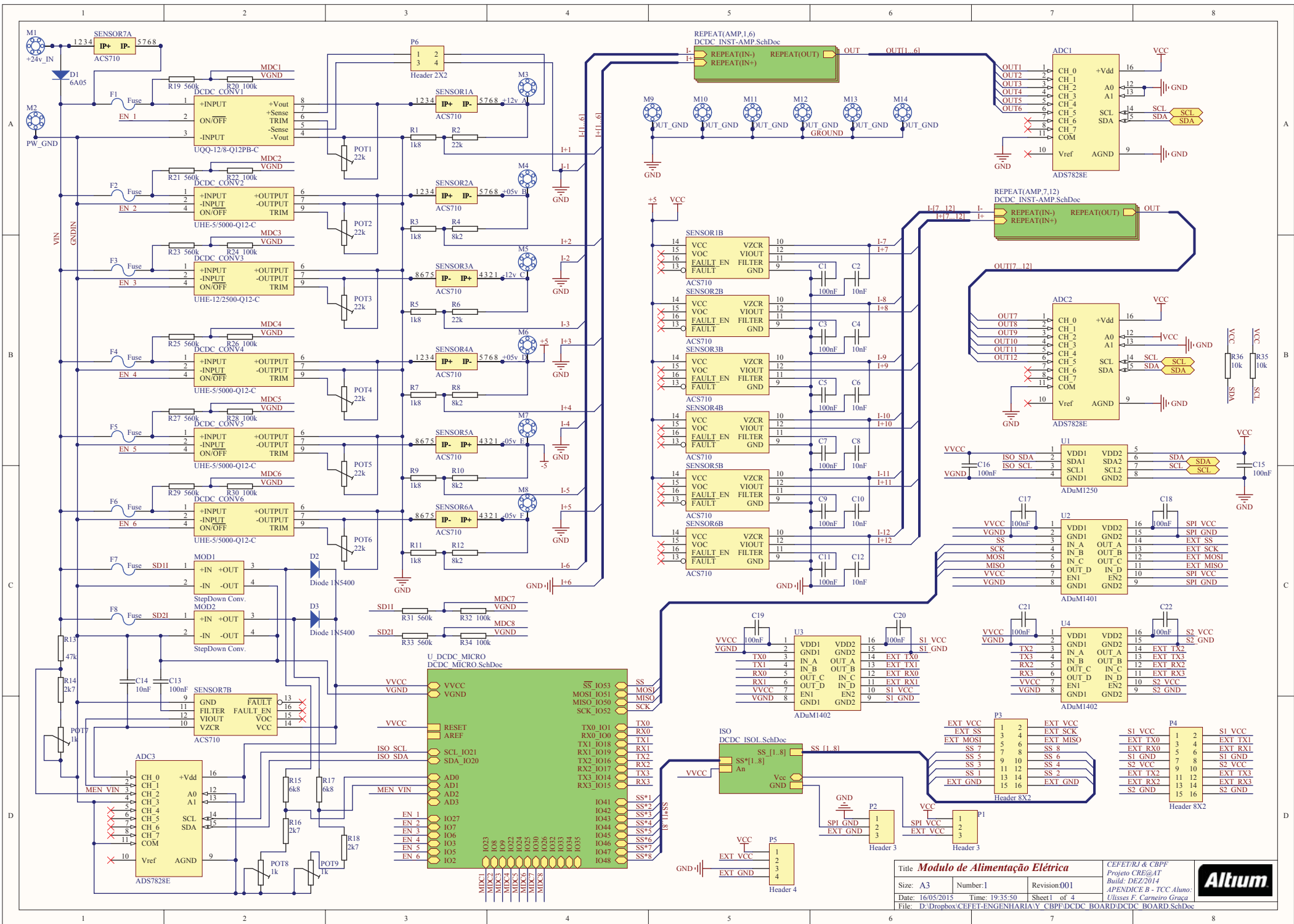
**APENDICE B:** Diagrama esquemático do Módulo Contador de Eventos Coincidentes

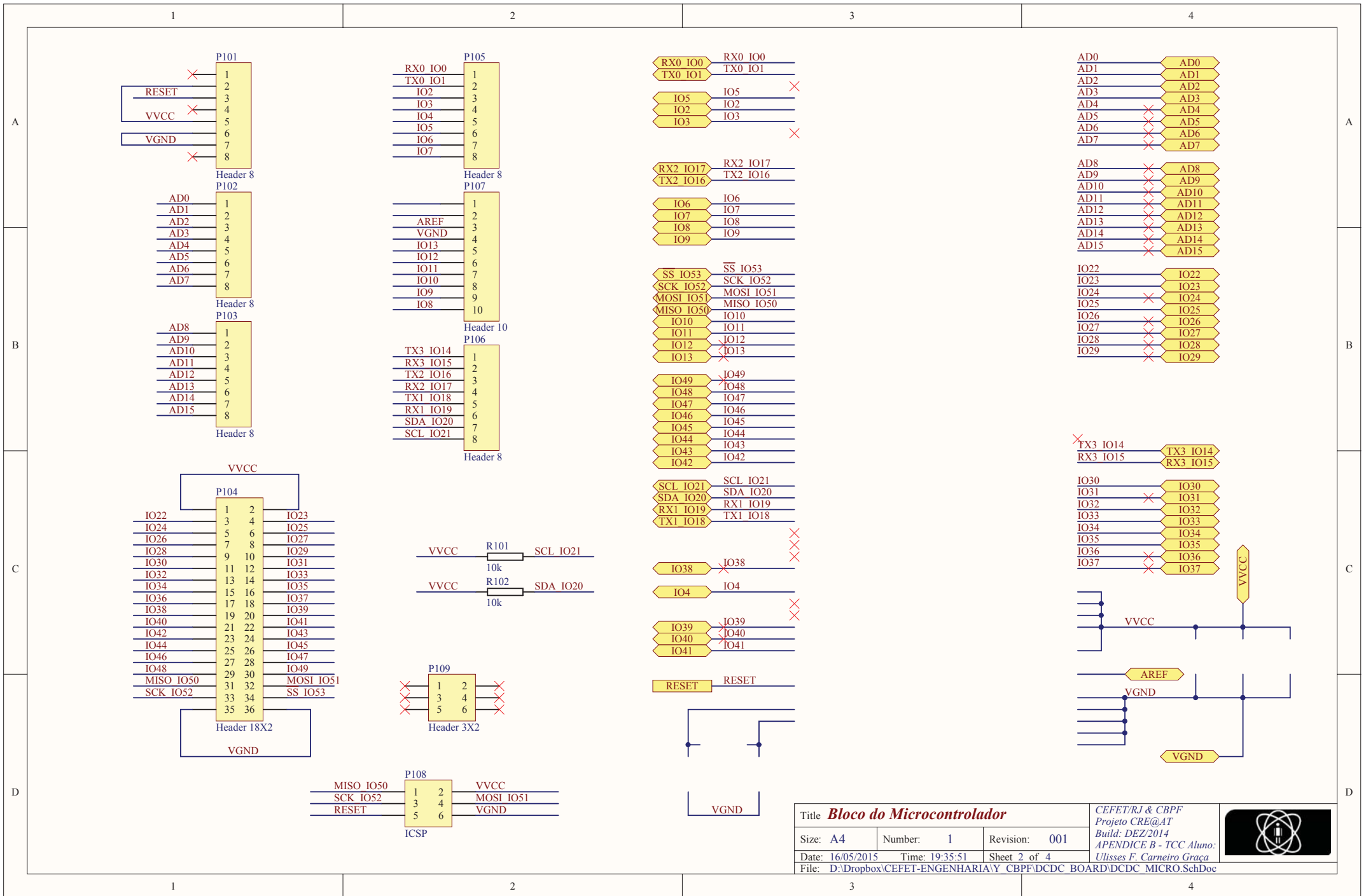
# Circuito Logico Detetor e Contador de Eventos Coincidentes





## **APENDICE C: Diagramas esquemáticos do Módulo de Alimentação Elétrica**





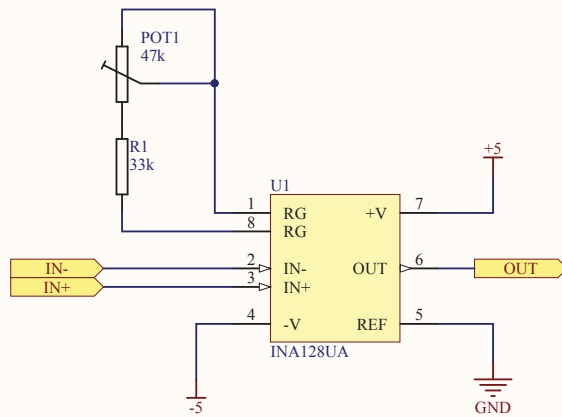
Title **Bloco do Microcontrolador**

Size: A4	Number: 1	Revision: 001
Date: 16/05/2015	Time: 19:35:51	Sheet 2 of 4
File: D:\Dropbox\CEFET-ENGENHARIA\Y CBPF\DCDC BOARD\DCDC_MICRO.SchDoc		

CEFET/RJ & CBPF  
 Projeto CRE@AT  
 Build: DEZ/2014  
 APENDICE B - TCC Aluno:  
 Ulisses F. Carneiro Graça

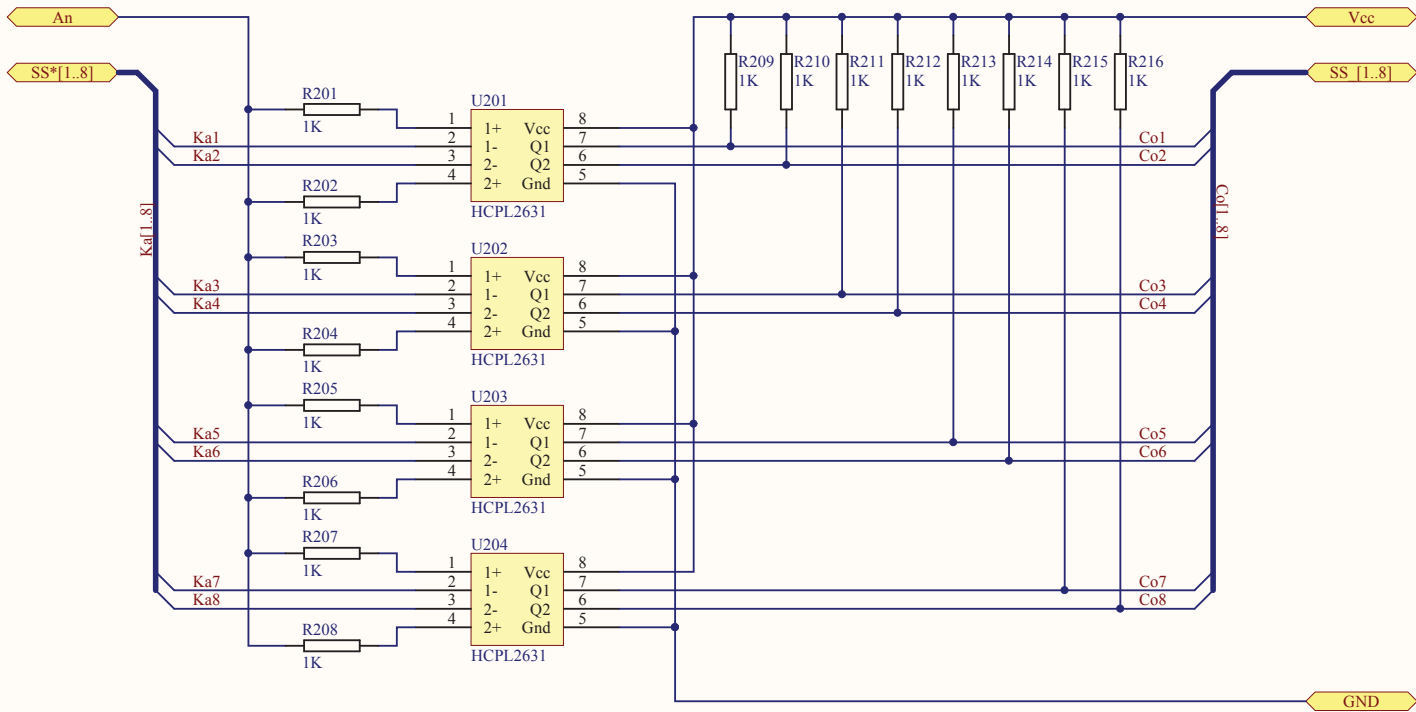






Title <b>Bloco do Amp. de Instrumentação</b>			CEFET/RJ & CBPF Projeto CRE@AT
Size: A4	Number: 3	Revision: 001	Build: DEZ/2014
Date: 16/05/2015	Time: 19:35:51	Sheet 3 of 4	APENDICE B - TCC Aluno: Ulisses F. Carneiro Graça
File: D:\Dropbox\CEFET-ENGENHARIA\Y CBPF\DCDC BOARD\DCDC INST-AMP.SchDoc			





Title <b>Bloco do Isolador Óptico</b>		
Size: A4	Number: 4	Revision: 001
Date: 16/05/2015	Time: 19:35:51	Sheet 4 of 4
File: D:\Dropbox\CEFET-ENGENHARIA\Y\CBPF\DCDC BOARD\DCDC ISOL.SchDoc		

CEFET/RJ & CBPF  
 Projeto CRE@AT  
 Build: DEZ/2014  
 APENDICE B - TCC Aluno:  
 Ulisses F. Carneiro Graça



## **APENDICE D: Código C do Módulo de Aquisição de Dados**

```

1 // Descrição do projeto:
2 // *****
3 // Company:          CENTRO BRASILEIRO DE PESQUISAS FISICAS - CBPF BRASIL
4 //                  CENTRO FEDERAL DE EDUCACAO TEC. CELSO SUCKOW DA FONSECA
5 // Project:         EXPERIMENTO DE RAIOS COSMICOS DA CRIOSFERA
6 //                  CRE@AT - Cosmic Ray Experiment at Antartida
7 // ATmega Proj. Name: CREAT_ATMegaSoftware DATALOGGER
8 // Device:          ATMEL ATmega2560
9 // Author:          Ulisses Carneiro (ulissestv@gmail.com)
10 // Date:           03-11-2014
11 // -----
12 // Description:     Recebe strings seriais UART do FPGA "CREAT_LogicAnalyzer"
13 //                  processa os dados, gera visualizacao local no LCD, registra
14 //                  dados brutos (RAW Data) e gera palavra de 16-bits para
15 //                  transmitir pelo satelite.
16 //                  Projeto desenvolvido para aplicacao especifica no
17 //                  experimento detector de raios cosmicos a ser instalado no
18 //                  modulo de pesquisas cientificas CRIOSFERA I, do INCT da
19 //                  CRIOSFERA (LAT S 84°00'00" - LON W 79°29'39") localizado na
20 //                  plataforma continental antartica.
21 // *****
22 //
23 // Revision History:
24 // #####
25 //   Date      Author          Rev.   Comments
26 // 03 Nov 14   Ulisses Carneiro 0001   Criacao - Primeira versao
27 // 08 Dez 14   Ulisses Carneiro 0002   Diversas alteracoes devido a modificacoes
28 //              na conexao com o satelite
29 // 27 Dez 14   Ulisses Carneiro 0005   Produzida versao definitiva
30
31
32 // Bibliotecas em uso:
33 #include <DS1307RTC.h> // Instrucoes de acesso ao RTC externo
34 #include <Time.h>     // Tratamento de variaveis temporais
35 #include <Wire.h>    // Uso do I2C para conexao com RTC
36 #include <LiquidCrystal.h> // Instrucoes de acesso ao LCD externo
37 #include <SD.h>      // Instrucoes de acesso ao SDcard externo
38
39 // Selecao de endereco do SDCard no SPI BUS:
40 const unsigned short SD01_CS = 4;
41 // Declaracao da nossa variavel de TEMPO ATUAL
42 tmElements_t tm;
43 // Variáveis necessárias p/ Biblioteca LCD:
44 LiquidCrystal lcd(23, 25, 27, 29, 31, 33);
45
46 //Pino de Saída de tensão analógica (Atenção para timer!!!)
47 const unsigned short pin_analog = 9;
48
49 //Níveis de tensão analógica:
50 const unsigned short mV_MAX = 4500; //mV_MIN<mV_MAX< 4500 ;
51 const unsigned short mV_MIN = 0;    // 0 <mV_MIN<mV_MAX;
52
53 //Variáveis necessárias para a rotina de recebimento SERIAL:
54 boolean stringComplete = false; // Variavel booleana que indica string completa
55 boolean serial_en = false;      // Flag para evitar recebimento de strings
56 // durante boot
57 unsigned short recebimento;

```

```
57 unsigned short rodada;
58
59 //Variáveis necessárias para gravacao no SDcard:
60 String inputString = "";           // Aqui se armazena a string recebida
61 String SDdataString = "";         // Aqui se monta a string a ser gravada no SDCard
62 String SDauxString = "";          // Aqui se monta a string a ser gravada no SDCard
63 boolean falhou;                   // Variavel booleana que indica falha na gravacao
64
65 //Escolha dos canais para transmissão:
66 const unsigned short g1 = 1;
67 const unsigned short g2 = 2;
68 const unsigned short g3 = 7;
69
70 //Variáveis do buffer de temporário de integração:
71 unsigned long T_Latch23g1, T_Latch13g1, T_Latch12g1, T_Latch3g1,
72             T_Latch23g2, T_Latch13g2, T_Latch12g2, T_Latch3g2,
73             T_Latch3g3;
74
75 //Variáveis do buffer de transmissão do satélite:
76 unsigned long B_Latch23g1, B_Latch13g1, B_Latch12g1, B_Latch3g1,
77             B_Latch23g2, B_Latch13g2, B_Latch12g2, B_Latch3g2,
78             B_Latch3g3;
79
80 /*Este programa foi desenvolvido para um tempo de aquisição de 10 min
81 no FPGA, integrando 6 tempos de aquisição num buffer temporário (T_Latch).
82 No instante em que é recebido o primeiro STRING SERIAL após um RESET do
83 microcontrolador o padrão de sincronismo é carregado no buffer de trans-
84 missão e uma rodada completa de sincronismo é transmitida. Cada rodada
85 completa é composta de 9 registros por período de aquisição vezes os 6
86 períodos de aquisição que são integrados por hora (um a cada 10 minutos)
87 para que se componha uma transmissão, ou seja, uma rodada acontece a cada
88 9 x 6 = 54 recepções de STRING serial. Após transcorrida a primeira rodada
89 (na qual foi transmitido o padrão de sincronismo) os 6 tempos de aquisição
90 */
91
92 void setup() {
93     TCCR2B = TCCR2B & 0b11111000 | 0x01; // Frequencia PWM pino 9 = 32kHz
94     Serial.begin(9600);
95     Serial1.begin(9600,SERIAL_8E2);
96     // reserve 300 bytes for the inputString:
97     inputString.reserve(300);
98     recebimento = 0;
99     rodada = 1;
100    inputString = "";
101    pinMode(53, OUTPUT);
102    pinMode(10, OUTPUT);
103    pinMode(4, OUTPUT);
104    pinMode(pin_analog, OUTPUT);
105    lcd.begin(16, 2);
106    lcd.setCursor(0, 0);
107    lcd.print(" -- CBPF -- ");
108    lcd.setCursor(0, 1);
109    lcd.print(" Projeto CRE@AT ");
110    delay(1500);
111    lcd.setCursor(0, 0);
112    lcd.print(" Inicializando ");
113    lcd.setCursor(0, 1);
```

```
114     lcd.print("  modulo RTC  ");
115     Serial.print("Iniciando modulo RTC...");
116     if (RTC.read(tm)) {
117         if (makeTime(tm) == 0) {
118             lcd.setCursor(0, 0);
119             lcd.print("  Falha no  ");
120             lcd.setCursor(0, 1);
121             lcd.print("  modulo RTC  ");
122             delay(1000);
123         }
124     }
125     lcd.setCursor(0, 0);
126     lcd.print(" Inicializando ");
127     lcd.setCursor(0, 1);
128     lcd.print("SDCard Primario");
129     Serial.print("Initializing SD card...");
130     if (!SD.begin(SD01_CS)) {
131         falhou = true;
132         Serial.println("Card failed, or not present");
133         lcd.setCursor(0, 0);
134         lcd.print("  Falha no  ");
135         lcd.setCursor(0, 1);
136         lcd.print("SDCard Primario");
137         delay(1000);
138     } else {
139         falhou = false;
140         Serial.println("card initialized.");
141         lcd.setCursor(0, 0);
142         lcd.print("SDCard1 iniciado");
143         lcd.setCursor(0, 1);
144         lcd.print("  com sucesso  ");
145         delay(1000);
146         escreveSDinicio(0);
147     }
148     lcd.setCursor(0, 0);
149     lcd.print(" ");
150     ExibeDataLCD();
151     lcd.print(" ");
152     lcd.setCursor(0, 1);
153     lcd.print(" ");
154     ExibeHoraLCD();
155     lcd.print(" ");
156     delay(1000);
157     atualizarTM();
158     lcd.setCursor(0, 0);
159     lcd.print(" ");
160     ExibeDataLCD();
161     lcd.print(" ");
162     lcd.setCursor(0, 1);
163     lcd.print(" ");
164     ExibeHoraLCD();
165     lcd.print(" ");
166     delay(1000);
167     serial_en = true;
168     lcd.setCursor(0, 0);
169     lcd.print("  Wait  Wait  ");
170     lcd.setCursor(0, 1);
```

```
171     lcd.print("  STRING  STATUS");
172   }
173
174   void loop() {
175     SDdataString = "";
176     // print the string when a newline arrives:
177     if (stringComplete) {
178       if (inputString.substring(0,1) == "{") {
179         atualizarTM();
180         if(recebimento >= 54){
181           recebimento = 1;
182           escreveSDinicio(3);
183           if (rodada >= 10){
184             rodada = 1;
185             Serial.println("Contagem de Rodadas Reiniciada");
186           } else {
187             rodada++;
188             Serial.println("Nova Rodada Iniciada");
189           }
190         } else {
191           recebimento++;
192         }
193         Serial.print("Rodada/Recebimento = ");
194         Serial.print(rodada);
195         Serial.print("/");
196         Serial.println(recebimento);
197         if (rodada == 1){
198           CarregaPadraoTesteBuffer();
199           Serial.println("Carregado Padrao Sincronismo");
200         } else {
201           if (recebimento == 1){
202             ZeraBufferSatelite();
203             Serial.println("BuffSat carregado / BuffTemp reiniciado");
204           }
205         }
206
207         //Incluido para gerar pulso de tudo OK:
208         if (recebimento == 3) digitalWrite(pin_analog, HIGH);
209         if (recebimento != 3) digitalWrite(pin_analog, LOW);
210
211         if (recebimento <= 6){
212           SomaBufferSatelite(g1,g2,g3);
213           Serial.print("Regs. dos canais ");
214           Serial.print(g1);
215           Serial.print(", ");
216           Serial.print(g2);
217           Serial.print(", ");
218           Serial.print(g3);
219           Serial.println(" somados no BuffTemp:");
220           Serial.println("-----");
221           Serial.print(" Registro   Trio");
222           Serial.print(g1);
223           Serial.print(" Trio");
224           Serial.print(g2);
225           Serial.print(" Trio");
226           Serial.println(g3);
227           Serial.print("T_Latch23: ");
```

```

228     EscreveLargura5(T_Latch23g1);
229     EscreveLargura5(T_Latch23g2);
230     Serial.println("    x ");
231     Serial.print("T_Latch13: ");
232     EscreveLargura5(T_Latch13g1);
233     EscreveLargura5(T_Latch13g2);
234     Serial.println("    x ");
235     Serial.print("T_Latch12: ");
236     EscreveLargura5(T_Latch12g1);
237     EscreveLargura5(T_Latch12g2);
238     Serial.println("    x ");
239     Serial.print("T_Latch3 : ");
240     EscreveLargura5(T_Latch3g1);
241     EscreveLargura5(T_Latch3g2);
242     EscreveLargura5(T_Latch3g3);
243     Serial.println("");
244     Serial.println("-----");
245 }
246 if      (recebimento <= 6){
247     Serial.print("Enviado B_Latch23g1 ");
248     TransmiteSatelite(B_Latch23g1);
249 } else if (recebimento <= 12){
250     Serial.print("Enviado B_Latch13g1 ");
251     TransmiteSatelite(B_Latch13g1);
252 } else if (recebimento <= 18){
253     Serial.print("Enviado B_Latch12g1 ");
254     TransmiteSatelite(B_Latch12g1);
255 } else if (recebimento <= 24){
256     Serial.print("Enviado B_Latch3g1 ");
257     TransmiteSatelite(B_Latch3g1);
258 } else if (recebimento <= 30){
259     Serial.print("Enviado B_Latch23g2 ");
260     TransmiteSatelite(B_Latch23g2);
261 } else if (recebimento <= 36){
262     Serial.print("Enviado B_Latch13g2 ");
263     TransmiteSatelite(B_Latch13g2);
264 } else if (recebimento <= 42){
265     Serial.print("Enviado B_Latch12g2 ");
266     TransmiteSatelite(B_Latch12g2);
267 } else if (recebimento <= 48){
268     Serial.print("Enviado B_Latch3g2 ");
269     TransmiteSatelite(B_Latch3g2);
270 } else if (recebimento <= 54){
271     Serial.print("Enviado B_Latch3g3 ");
272     TransmiteSatelite(B_Latch3g3);
273 }
274 SDdataString += "1,";
275 SDdataString += recebimento;
276 SDdataString += ">>";
277 SDdataString += String(tmYearToCalendar(tm.Year));
278 SDdataString += "/";
279 SDdataString += String(tm.Month);
280 SDdataString += "/";
281 SDdataString += String(tm.Day);
282 SDdataString += "@";
283 SDdataString += String(tm.Hour);
284 SDdataString += ":";

```





```

334     Serial.println("");
335     Serial.print("FREQ3:      ");
336     for (int i = 0; i<8 ; i++){
337         EscreveLargura5(conversao(inputString.substring(5*6+7*6*i+26,5*6+7*6*i
338             +30)));
339     }
340     Serial.println("");
341     Serial.println(
342         "-----");
343     if (!SD.begin(SD01_CS)) {
344         falhou = true;
345         Serial.println("Card failed, or not present");
346         lcd.setCursor(0, 0);
347         lcd.print("RX SdFail      ");
348     } else {
349         if(falhou) {
350             escreveSDinicio(2);
351         }
352         escreveSD(SDdataString);
353         escreveRawSD(inputString);
354         falhou = false;
355         lcd.setCursor(0, 0);
356         lcd.print("  RX OK      ");
357     }
358     lcd.setCursor(0, 1);
359     ExibeHoraLCD();
360 }
361     lcd.setCursor(10, 0);
362     lcd.print(conversao(inputString.substring(2,6)));
363     lcd.print("      ");
364     lcd.setCursor(10, 1);
365     lcd.print(conversao(inputString.substring(8,12)));
366     lcd.print("      ");
367     inputString = "";
368     stringComplete = false;
369 }
370 void serialEvent1() {
371     if (serial_en == true){
372         while (Serial1.available()) {
373             char inChar = (char)Serial1.read();
374             if (inChar == '\n') {
375                 stringComplete = true;
376             }
377             else {
378                 inputString += inChar;
379             }
380         }
381     }
382 }
383
384 unsigned long conversao(String novo_valor) {
385     char conv_char[novo_valor.length() + 1];
386     novo_valor.toCharArray(conv_char, (novo_valor.length() + 1));
387     return strtoul(conv_char, NULL, 16);
388 }

```

```
389
390 void print2digits(int number) {
391     if (number >= 0 && number < 10) {
392         Serial.write('0');
393     }
394     Serial.print(number);
395 }
396
397 void LCDprint2dig(int number) {
398     if (number >= 0 && number < 10) {
399         lcd.print("0");
400     }
401     lcd.print(number);
402 }
403
404 void ExibeDataLCD(){
405     lcd.print(" ");
406     LCDprint2dig(tm.Day);
407     lcd.print("/");
408     LCDprint2dig(tm.Month);
409     lcd.print("/");
410     LCDprint2dig(tmYearToCalendar(tm.Year));
411     lcd.print(" ");
412 }
413
414 void ExibeHoraLCD(){
415     lcd.print(" ");
416     LCDprint2dig(tm.Hour);
417     lcd.print(":");
418     LCDprint2dig(tm.Minute);
419     lcd.print(":");
420     LCDprint2dig(tm.Second);
421     lcd.print(" ");
422 }
423
424 void atualizarTM(){
425     if (RTC.read(tm)) {
426         Serial.print("RTC lido em ");
427         print2digits(tm.Hour);
428         Serial.write(':');
429         print2digits(tm.Minute);
430         Serial.write(':');
431         print2digits(tm.Second);
432         Serial.print(", ");
433         Serial.print(tm.Day);
434         Serial.write('/');
435         Serial.print(tm.Month);
436         Serial.write('/');
437         Serial.print(tmYearToCalendar(tm.Year));
438         Serial.println();
439     } else {
440         if (RTC.chipPresent()) {
441             Serial.println("The DS1307 is stopped. Please run the SetTime");
442             Serial.println("example to initialize the time and begin running.");
443             Serial.println();
444         } else {
445             Serial.println("DS1307 read error! Please check the circuitry.");
```

```

446     Serial.println();
447     }
448 }
449 }
450
451 void escreveSDinicio(int id){
452     SDauxString = "";
453     SDauxString += id;
454     SDauxString += ",0>>";
455     SDauxString += String(tmYearToCalendar(tm.Year));
456     SDauxString += ",";
457     SDauxString += String(tm.Month);
458     SDauxString += ",";
459     SDauxString += String(tm.Day);
460     SDauxString += ",";
461     SDauxString += String(tm.Hour);
462     SDauxString += ",";
463     SDauxString += String(tm.Minute);
464     SDauxString += ",";
465     SDauxString += String(tm.Second);
466     SDauxString += ">> ";
467     SDauxString += "L2_23a L2_13a L2_12a FRQ_1a FRQ_2a FRQ_3a LATC3a ";
468     SDauxString += "L2_23b L2_13b L2_12b FRQ_1b FRQ_2b FRQ_3b LATC3b ";
469     SDauxString += "L2_23c L2_13c L2_12c FRQ_1c FRQ_2c FRQ_3c LATC3c ";
470     SDauxString += "L2_23d L2_13d L2_12d FRQ_1d FRQ_2d FRQ_3d LATC3d ";
471     SDauxString += "L2_23e L2_13e L2_12e FRQ_1e FRQ_2e FRQ_3e LATC3e ";
472     SDauxString += "L2_23f L2_13f L2_12f FRQ_1f FRQ_2f FRQ_3f LATC3f ";
473     SDauxString += "L2_23g L2_13g L2_12g FRQ_1g FRQ_2g FRQ_3g LATC3g ";
474     escreveSD(SDauxString);
475 }
476
477 void escreveSD(String StrAGravar){
478     // open the file. note that only one file can be open at a time,
479     // so you have to close this one before opening another.
480     File dataFile = SD.open("datalog.txt", FILE_WRITE);
481     // if the file is available, write to it:
482     if (dataFile) {
483         dataFile.println(StrAGravar);
484         dataFile.close();
485         // print to the serial port too:
486         Serial.println(StrAGravar);
487         lcd.setCursor(0, 0);
488         lcd.print(" Grav. OK ");
489         lcd.setCursor(0, 1);
490         ExibeHoraLCD();
491     }
492     // if the file isn't open, pop up an error:
493     else {
494         Serial.println("erro ao abrir datalog.txt");
495         lcd.setCursor(0, 0);
496         lcd.print("Falhou a escrita");
497         lcd.setCursor(0, 1);
498         lcd.print(" LOG no SDCard1 ");
499     }
500 }
501
502 void escreveRawSD(String StrAGravar){

```

```
503 // open the file. note that only one file can be open at a time,
504 // so you have to close this one before opening another.
505 File dataFile = SD.open("RAWdata.txt", FILE_WRITE);
506 // if the file is available, write to it:
507 if (dataFile) {
508     dataFile.println(StrAGravar);
509     dataFile.close();
510     // print to the serial port too:
511     Serial.println(StrAGravar);
512 }
513 // if the file isn't open, pop up an error:
514 else {
515     Serial.println("erro ao abrir RAWdata.txt");
516     lcd.setCursor(0, 0);
517     lcd.print("Falhou a escrita");
518     lcd.setCursor(0, 1);
519     lcd.print(" RAW no SDCard1 ");
520 }
521 }
522
523 void ZeraBufferSatelite(){
524     B_Latch23g1 = T_Latch23g1;
525     B_Latch13g1 = T_Latch13g1;
526     B_Latch12g1 = T_Latch12g1;
527     B_Latch3g1 = T_Latch3g1;
528     B_Latch23g2 = T_Latch23g2;
529     B_Latch13g2 = T_Latch13g2;
530     B_Latch12g2 = T_Latch12g2;
531     B_Latch3g2 = T_Latch3g2;
532     B_Latch3g3 = T_Latch3g3;
533     T_Latch23g1 = 0;
534     T_Latch13g1 = 0;
535     T_Latch12g1 = 0;
536     T_Latch3g1 = 0;
537     T_Latch23g2 = 0;
538     T_Latch13g2 = 0;
539     T_Latch12g2 = 0;
540     T_Latch3g2 = 0;
541     T_Latch3g3 = 0;
542 }
543
544 void SomaBufferSatelite(int g1, int g2, int g3){
545     T_Latch23g1 += conversao(inputString.substring(0*6+7*6*g1+26,0*6+7*6*g1+30));
546     T_Latch13g1 += conversao(inputString.substring(1*6+7*6*g1+26,1*6+7*6*g1+30));
547     T_Latch12g1 += conversao(inputString.substring(2*6+7*6*g1+26,2*6+7*6*g1+30));
548     T_Latch3g1 += conversao(inputString.substring(6*6+7*6*g1+26,6*6+7*6*g1+30));
549     T_Latch23g2 += conversao(inputString.substring(0*6+7*6*g2+26,0*6+7*6*g2+30));
550     T_Latch13g2 += conversao(inputString.substring(1*6+7*6*g2+26,1*6+7*6*g2+30));
551     T_Latch12g2 += conversao(inputString.substring(2*6+7*6*g2+26,2*6+7*6*g2+30));
552     T_Latch3g2 += conversao(inputString.substring(6*6+7*6*g2+26,6*6+7*6*g2+30));
553     T_Latch3g3 += conversao(inputString.substring(6*6+7*6*g3+26,6*6+7*6*g3+30));
554 }
555
556 void TransmiteSatelite(unsigned long tx_val){
557     if (tx_val < 10000) Serial2.print("0");
558     if (tx_val < 1000) Serial2.print("0");
559     if (tx_val < 100) Serial2.print("0");
```

```
560     if (tx_val < 10) Serial2.print("0");
561     Serial2.println(tx_val);
562
563     Serial.print("para satelite = ");
564     if (tx_val < 10000) Serial.print("0");
565     if (tx_val < 1000) Serial.print("0");
566     if (tx_val < 100) Serial.print("0");
567     if (tx_val < 10) Serial.print("0");
568     Serial.print(tx_val);
569 }
570
571 void EscreveLargura5(unsigned long value){
572     Serial.print(" ");
573     if (value < 10000) Serial.print(" ");
574     if (value < 1000) Serial.print(" ");
575     if (value < 100) Serial.print(" ");
576     if (value < 10) Serial.print(" ");
577     Serial.print(value);
578 }
579
580 void CarregaPadraoTesteBuffer(){
581     B_Latch23g1 = 6554;
582     B_Latch13g1 = 13107;
583     B_Latch12g1 = 19661;
584     B_Latch3g1 = 26214;
585     B_Latch23g2 = 32768;
586     B_Latch13g2 = 39321;
587     B_Latch12g2 = 45875;
588     B_Latch3g2 = 52429;
589     B_Latch3g3 = 58982;
590 }
```

## **APENDICE E: Código C do Módulo Controlador de Carga**

```

1 // Descrição do projeto:
2 // *****
3 // Company:          CENTRO BRASILEIRO DE PESQUISAS FISICAS - CBPF BRASIL
4 //                  CENTRO FEDERAL DE EDUCACAO TEC. CELSO SUCKOW DA FONSECA
5 // Project:          EXPERIMENTO DE RAIOS COSMICOS DA CRIOSFERA
6 //                  CRE@AT - Cosmic Ray Experiment at Antartida
7 // ATmega Proj. Name: FORGEN_Controller
8 // Device:           ATMEL ATmega2560
9 // Author:           Ulisses Carneiro (ulissestv@gmail.com)
10 // Date:            27-12-2014
11 // -----
12 // Description:      Controla carga das baterias fornecida pelas turbinas FORGEN
13 //                  e recebe, decodifica e envia para transmissão por satélite
14 //                  os dados recebidos do FPGA e selecionados.
15 //                  Projeto desenvolvido para aplicacao especifica no
16 //                  experimento detector de raios cosmicos a ser instalado no
17 //                  modulo de pesquisas cientificas CRIOSFERA I, do INCT da
18 //                  CRIOSFERA (LAT S 84°00'00" - LON W 79°29'39") localizado na
19 //                  plataforma continental antartica.
20 // *****
21 //
22 // Revision History:
23 // #####
24 //   Date      Author          Rev.   Comments
25 // 03 Nov 14   Ulisses Carneiro 0001   Criacao - Primeira versao
26 // 08 Dez 14   Ulisses Carneiro 0002   Diversas alteracoes devido a modificacoes
27 //            na conexao com o satélite
28 // 27 Dez 14   Ulisses Carneiro 0003   Produzida versao definitiva
29
30
31 // Bibliotecas em uso:
32 #include <PID_v1.h>
33
34 unsigned long tamanho_serial = 0;
35 // Prototipo da funcao de conversao STRING >> INTEIRO
36 unsigned long conversao(String novo_valor);
37
38 //Variáveis necessárias para a rotina de recebimento SERIAL:
39 String inputString = ""; // Aqui se armazena a string recebida
40 boolean stringComplete = false; // Variavel booleana que indica string completa
41 String inputString2 = ""; // Aqui se armazena a string recebida
42 boolean stringComplete2 = false; // Variavel booleana que indica string completa
43 boolean serial_en2 = false; // Flag para evitar recebimento de strings
44 // durante boot
45 boolean enviarOFF = false;
46 boolean desligou = false;
47
48 unsigned short recebimento;
49 unsigned short rodada;
50 unsigned short envioOFF;
51
52 //Pino de Saída de tensão analógica (Atenção para timer!!!)
53 const unsigned short pin_analog = 9;
54
55 //Níveis de tensão analógica:
56 const unsigned short mV_MAX = 4500; //mV_MIN<mV_MAX< 4500 ;
57 const unsigned short mV_MIN = 0; // 0 <mV_MIN<mV_MAX;

```



```
57
58 //Escolha dos canais para transmissão:
59 const unsigned short g1 = 1;
60 const unsigned short g2 = 2;
61 const unsigned short g3 = 7;
62
63 //Variáveis do buffer de temporário de integração:
64 unsigned long T_Latch23g1, T_Latch13g1, T_Latch12g1, T_Latch3g1,
65             T_Latch23g2, T_Latch13g2, T_Latch12g2, T_Latch3g2,
66             T_Latch3g3;
67
68 //Variáveis do buffer de transmissão do satélite:
69 unsigned long B_Latch23g1, B_Latch13g1, B_Latch12g1, B_Latch3g1,
70             B_Latch23g2, B_Latch13g2, B_Latch12g2, B_Latch3g2,
71             B_Latch3g3;
72
73
74
75 //Define Variables we'll be connecting to
76 double SP_Vinp1, IP_Vinp1, OP_PWM_1inp,
77        SP_Vout1, IP_Vout1, OP_PWM_1out,
78        SP_Vide1,           OP_PWM_1ide,
79        SP_Vinp2, IP_Vinp2, OP_PWM_2inp,
80        SP_Vout2, IP_Vout2, OP_PWM_2out,
81        SP_Vide2,           OP_PWM_2ide;
82
83 double IP_Vout3;
84
85 //Effective program variables:
86 int   PWM_1a, PWM_1b,
87       PWM_2a, PWM_2b;
88
89 double rpm1, rpm2;
90
91 double   B_rpm1,
92         B_rpm2,
93         B_IP_Vout1,
94         B_IP_Vout2,
95         B_IP_Vout3;
96
97 int     B_PWM_1a,
98        B_PWM_1b,
99        B_PWM_2a,
100       B_PWM_2b;
101
102 int     MA_PWM_1a, MA_PWM_1b,
103        MA_PWM_2a, MA_PWM_2b;
104
105 const double fundoEscala = 33.28;
106
107 int conv_int = 0;
108 int modo = 0;
109
110 int rpm1_last, rpm1_now, rpm2_last, rpm2_now;
111
112 unsigned long last_rpmtime, rpm1_count,
113             rpm2_count,
```

```
114         last_recorder, last_OFFtime;
115
116     const unsigned long intervalo_rpm = 1000,
117         OFF_time = 3600000;
118
119
120     //Specify the links and initial tuning parameters
121     PID myPID_1a(&IP_Vout1, &OP_PWM_lout, &SP_Vout1, 2, 5, 1, DIRECT);
122     //Carga em funcao da Bateria
123     PID myPID_1b(&rpm1, &OP_PWM_lide, &SP_Vide1, 1, 0.02, 0.5, REVERSE);
124     //Carga em funcao do RPM
125     PID myPID_1c(&rpm1, &OP_PWM_linp, &SP_Vinp1, 2, 5, 1, REVERSE);
126     //Freio em funcao do RPM
127     PID myPID_2a(&IP_Vout2, &OP_PWM_2out, &SP_Vout2, 2, 5, 1, DIRECT);
128     PID myPID_2b(&rpm2, &OP_PWM_2ide, &SP_Vide2, 1, 0.02, 0.5, REVERSE);
129     PID myPID_2c(&rpm2, &OP_PWM_2inp, &SP_Vinp2, 2, 5, 1, REVERSE);
130
131     void setup()
132     {
133         analogReference(INTERNAL2V56);
134
135         TCCR1B = TCCR1B & 0b11111000 | 0x01; // Frequencia PWM pino 9 = 32kHz
136         TCCR2B = TCCR2B & 0b11111000 | 0x01; // Frequencia PWM pino 9 = 32kHz
137         TCCR3B = TCCR3B & 0b11111000 | 0x01; // Frequencia PWM pino 9 = 32kHz
138         TCCR4B = TCCR4B & 0b11111000 | 0x01; // Frequencia PWM pino 9 = 32kHz
139
140         Serial.begin(115200);
141         Serial2.begin(9600, SERIAL_8E2);
142         // reserve 200 bytes for the inputString:
143         inputString2.reserve(300);
144         recebimento = 0;
145         rodada = 1;
146
147         pinMode(2, INPUT_PULLUP);
148         pinMode(3, INPUT_PULLUP);
149         pinMode(15, OUTPUT);
150         pinMode(53, OUTPUT);
151
152         //initialize the variables we're linked to
153         IP_Vout1 = analogRead(7); //Leitura da tensao da bateria
154         IP_Vout2 = analogRead(4);
155         IP_Vout3 = analogRead(2);
156         SP_Vinp1 = 600.00; //SP do freio para bateria descarregada
157         SP_Vinp2 = 600.00;
158         SP_Vout1 = 866.84; //SP da tensao de carga da bateria
159         SP_Vout2 = 866.84;
160         SP_Vide1 = 400.00; //SP da rotacao minima para carga
161         SP_Vide2 = 400.00;
162
163         //turn the PID on
164         myPID_1a.SetMode(AUTOMATIC);
165         myPID_1b.SetMode(AUTOMATIC);
166         myPID_1c.SetMode(AUTOMATIC);
167         myPID_2a.SetMode(AUTOMATIC);
168         myPID_2b.SetMode(AUTOMATIC);
169         myPID_2c.SetMode(AUTOMATIC);
```

```
168
169     myPID_1a.SetOutputLimits(0, 191); //Carga em funcao da Bateria
170     myPID_1b.SetOutputLimits(0, 191); //Carga em funcao do RPM
171     myPID_1c.SetOutputLimits(0, 191); //Freio em funcao do RPM
172     myPID_2a.SetOutputLimits(0, 191); //Carga em funcao da Bateria
173     myPID_2b.SetOutputLimits(0, 191); //Carga em funcao do RPM
174     myPID_2c.SetOutputLimits(0, 191); //Freio em funcao do RPM
175
176     serial_en2 = true;
177
178     exibeInformacoes();
179     Serial.println("Aguardando proximo comando...");
180 }
181
182 void loop()
183 {
184
185     if (millis() > 259200000) {
186
187         pinMode(14, OUTPUT);
188         digitalWrite(14, LOW);
189     }
190
191     IP_Vout1 = analogRead(7); //Leitura da tensao da bateria
192     IP_Vout2 = analogRead(4);
193     IP_Vout3 = analogRead(2);
194
195     if (millis() > last_rpmtime + intervalo_rpm){
196         last_rpmtime = millis();
197         rpml = rpml_count*6*1.25;
198         rpml_count=0;
199         rpm2 = rpm2_count*6*1.25;
200         rpm2_count=0;
201     }
202
203     rpml_now = digitalRead(2);
204     if (rpml_last != rpml_now){
205         rpml_last = rpml_now;
206         rpml_count++;
207     }
208     rpm2_now = digitalRead(3);
209     if (rpm2_last != rpm2_now){
210         rpm2_last = rpm2_now;
211         rpm2_count++;
212     }
213
214     myPID_1a.Compute();
215     if (OP_PWM_1out != 0) {
216         myPID_1c.Compute();
217     } else {
218         OP_PWM_linp = 200;
219     }
220     myPID_1b.Compute();
221
222     myPID_2a.Compute();
223     if (OP_PWM_2out != 0) {
224         myPID_2c.Compute();
```

```

225     } else {
226         OP_PWM_2inp = 200;
227     }
228     myPID_2b.Compute();
229
230     //Escolhe o PWM da saída baseado no mínimo entre PID out da Vin e Vout:
231     if (modo == 0 ){
232         if(OP_PWM_1ide < OP_PWM_1out){
233             PWM_1b = OP_PWM_1ide;
234         } else {
235             PWM_1b = OP_PWM_1out;
236         }
237         if(OP_PWM_2ide < OP_PWM_2out){
238             PWM_2b = OP_PWM_2ide;
239         } else {
240             PWM_2b = OP_PWM_2out;
241         }
242
243         PWM_1a = OP_PWM_1inp;
244         PWM_2a = OP_PWM_2inp;
245     } else {
246         PWM_1a = MA_PWM_1a;
247         PWM_2a = MA_PWM_2a;
248         PWM_1b = MA_PWM_1b;
249         PWM_2b = MA_PWM_2b;
250     }
251
252     analogWrite(11, 255-PWM_1a); //Freio turbina 1
253     analogWrite(12, 255-PWM_1b); //Carga bateria 1
254     analogWrite(5, 255-PWM_2a); //Freio turbina 2
255     analogWrite(6, 255-PWM_2b); //Carga bateria 2
256
257     if (stringComplete) {
258         if (inputString.substring(0,4) == String("SP1I")) {
259             ConversaoDireta();
260             SP_Vinp1 = conv_int;
261             Serial.print("Set Point SP_Vinp1 (RPM) = ");
262             Serial.println(SP_Vinp1);
263         } else if (inputString.substring(0,4) == String("SP2I")) {
264             ConversaoDireta();
265             SP_Vinp2 = vTensao1024(conv_int);
266             Serial.print("Set Point SP_Vinp2 (RPM) = ");
267             Serial.println(SP_Vinp2);
268         } else if (inputString.substring(0,4) == String("SP1O")) {
269             ConversaoDireta();
270             SP_Vout1 = vTensao1024(conv_int);
271             Serial.print("Set Point SP_Vout1 = ");
272             Serial.print(SP_Vout1);
273             Serial.print(" (");
274             Serial.print(vTensao(SP_Vout1));
275             Serial.print(" Volts)");
276         } else if (inputString.substring(0,4) == String("SP2O")) {
277             ConversaoDireta();
278             SP_Vout2 = vTensao1024(conv_int);
279             Serial.print("Set Point SP_Vout2 = ");
280             Serial.print(SP_Vout2);
281             Serial.print(" (");

```

```
282     Serial.print(vTensao(SP_Vout2));
283     Serial.print(" Volts");
284 } else if (inputString.substring(0,4) == String("SP1M")) {
285     ConversaoDireta();
286     SP_Vide1 = conv_int;
287     Serial.print("Set Point SP_Vide1 (RPM) = ");
288     Serial.println(SP_Vide1);
289 } else if (inputString.substring(0,4) == String("SP2M")) {
290     ConversaoDireta();
291     SP_Vide2 = vTensao1024(conv_int);
292     Serial.print("Set Point SP_Vide2 (RPM) = ");
293     Serial.println(SP_Vide2);
294 } else if (inputString.substring(0,4) == String("MODE")) {
295     ConversaoDireta();
296     modo = conv_int;
297     Serial.print("Modo de Operacao = ");
298     Serial.print(modo);
299     if (modo == 0) Serial.println(" (Automatico)");
300     if (modo != 0) Serial.println(" (Manual)");
301 } else if (inputString.substring(0,4) == String("MA1I")) {
302     ConversaoDireta();
303     MA_PWM_1a = conv_int;
304     Serial.print("Valor MA_PWM_1a = ");
305     Serial.println(MA_PWM_1a);
306 } else if (inputString.substring(0,4) == String("MA2I")) {
307     ConversaoDireta();
308     MA_PWM_2a = conv_int;
309     Serial.print("Valor MA_PWM_2a = ");
310     Serial.println(MA_PWM_2a);
311 } else if (inputString.substring(0,4) == String("MA1O")) {
312     ConversaoDireta();
313     MA_PWM_1b = conv_int;
314     Serial.print("Valor MA_PWM_1b = ");
315     Serial.println(MA_PWM_1b);
316 } else if (inputString.substring(0,4) == String("MA2O")) {
317     ConversaoDireta();
318     MA_PWM_2b = conv_int;
319     Serial.print("Valor MA_PWM_2b = ");
320     Serial.println(MA_PWM_2b);
321 } else {
322
323     exibeInformacoes();
324
325     Serial.println("Rotacao (RPM):");
326     Serial.print("Turbinas: ");
327     Serial.println(rpm1);
328     Serial.print("Turbinas2: ");
329     Serial.println(rpm2);
330     Serial.println("");
331
332     Serial.println("Tensoes de saida:");
333     Serial.print("Bateria: ");
334     Serial.println(vTensao(IP_Vout1));
335     Serial.print("Bateria2: ");
336     Serial.println(vTensao(IP_Vout2));
337     Serial.print("Bateria3: ");
338     Serial.println(vTensao(IP_Vout3));
```

```
339     Serial.println("");
340
341     Serial.println("Set Points de Freio (RPM):");
342     Serial.print("SP_Vinpl: ");
343     Serial.println(SP_Vinpl);
344     Serial.print("SP_Vinp2: ");
345     Serial.println(SP_Vinp2);
346     Serial.println("");
347
348     Serial.println("Set Points tensao flutuacao:");
349     Serial.print("SP_Vout1: ");
350     Serial.print(SP_Vout1);
351     Serial.print(" (");
352     Serial.print(vTensao(SP_Vout1));
353     Serial.println(" Volts)");
354     Serial.print("SP_Vout2: ");
355     Serial.print(SP_Vout2);
356     Serial.print(" (");
357     Serial.print(vTensao(SP_Vout2));
358     Serial.println(" Volts)");
359     Serial.println("");
360
361     Serial.println("Set Points rotacao minima da turbina:");
362     Serial.print("SP_Vide1: ");
363     Serial.println(SP_Vide1);
364     Serial.print("SP_Vide2: ");
365     Serial.println(SP_Vide2);
366     Serial.println("");
367
368     Serial.print("Modo de Operacao = ");
369     Serial.print(modo);
370     if (modo == 0) Serial.println(" (Automatico)");
371     if (modo != 0) Serial.println(" (Manual)");
372     Serial.println("");
373
374     Serial.println("Variaveis de controle do Freio (0 a 255):");
375     Serial.print("Turbinal: ");
376     Serial.println(PWM_1a);
377     Serial.print("Turbina2: ");
378     Serial.println(PWM_2a);
379
380     Serial.println("Variaveis de controle da Carga (0 a 255):");
381     Serial.print("Baterial: ");
382     Serial.println(PWM_1b);
383     Serial.print("Bateria2: ");
384     Serial.println(PWM_2b);
385 }
386
387 Serial.println("");
388 Serial.println("Aguardando proximo comando...");
389 // clear the string:
390 inputString = "";
391 stringComplete = false;
392 }
393
394 if (!desligou){
395     if (analogRead(0) < 245.91) { //Se tensao de alimentacao do CRATE < 8 Volts
```

```
396     desligou = true;
397     enviarOFF = true;
398     envioOFF = 0;
399     last_OFFtime = 0;
400
401     B_rpm1 = rpm1;
402     B_rpm2 = rpm2;
403     B_IP_Vout1 = IP_Vout1;
404     B_IP_Vout2 = IP_Vout2;
405     B_IP_Vout3 = IP_Vout3;
406     B_PWM_1a = PWM_1a;
407     B_PWM_1b = PWM_1b;
408     B_PWM_2a = PWM_2a;
409     B_PWM_2b = PWM_2b;
410
411     Serial.println("Detectado desligamento");
412 }
413 }
414
415 if (enviarOFF){
416     if (millis() > last_OFFtime + OFF_time){
417         last_OFFtime = millis();
418         envioOFF ++;
419         if (envioOFF <= 1){
420             ZeraBufferSatelite();
421             Serial.print("Enviado em OFF B_Latch23g1 ");
422             TransmiteSatelite(B_Latch23g1);
423         } else if (envioOFF <= 2){
424             Serial.print("Enviado em OFF B_Latch13g1 ");
425             TransmiteSatelite(B_Latch13g1);
426         } else if (envioOFF <= 3){
427             Serial.print("Enviado em OFF B_Latch12g1 ");
428             TransmiteSatelite(B_Latch12g1);
429         } else if (envioOFF <= 4){
430             Serial.print("Enviado em OFF B_Latch3g1 ");
431             TransmiteSatelite(B_Latch3g1);
432         } else if (envioOFF <= 5){
433             Serial.print("Enviado em OFF B_Latch23g2 ");
434             TransmiteSatelite(B_Latch23g2);
435         } else if (envioOFF <= 6){
436             Serial.print("Enviado em OFF B_Latch13g2 ");
437             TransmiteSatelite(B_Latch13g2);
438         } else if (envioOFF <= 7){
439             Serial.print("Enviado em OFF B_Latch12g2 ");
440             TransmiteSatelite(B_Latch12g2);
441         } else if (envioOFF <= 8){
442             Serial.print("Enviado em OFF B_Latch3g2 ");
443             TransmiteSatelite(B_Latch3g2);
444         } else if (envioOFF <= 9){
445             Serial.print("Enviado em OFF B_Latch3g3 ");
446             TransmiteSatelite(B_Latch3g3);
447         } else if (envioOFF <= 10){
448             Serial.print("Enviado em OFF RPM1 ");
449             TransmiteSatelite(to_longuns(B_rpm1));
450         } else if (envioOFF <= 11){
451             Serial.print("Enviado em OFF RPM2 ");
452             TransmiteSatelite(to_longuns(B_rpm2));
```

```
453     } else if (envioOFF <= 12){
454         Serial.print("Enviado em OFF Vbat1 ");
455         TransmiteSatelite(20*to_longuns(vTensao(B_IP_Vout1)));
456     } else if (envioOFF <= 13){
457         Serial.print("Enviado em OFF Vbat2 ");
458         TransmiteSatelite(20*to_longuns(vTensao(B_IP_Vout2)));
459     } else if (envioOFF <= 14){
460         Serial.print("Enviado em OFF Vbat3 ");
461         TransmiteSatelite(20*to_longuns(vTensao(B_IP_Vout3)));
462     } else if (envioOFF <= 15){
463         Serial.print("Enviado em OFF PWM1a ");
464         TransmiteSatelite(300*B_PWM_1a);
465     } else if (envioOFF <= 16){
466         Serial.print("Enviado em OFF PWM1b ");
467         TransmiteSatelite(300*B_PWM_1b);
468     } else if (envioOFF <= 17){
469         Serial.print("Enviado em OFF PWM2a ");
470         TransmiteSatelite(300*B_PWM_2a);
471     } else if (envioOFF <= 18){
472         Serial.print("Enviado em OFF PWM2b ");
473         TransmiteSatelite(300*B_PWM_2b);
474     } else {
475         Serial.println("Envio em OFF finalizado");
476         TransmiteSatelite(0);
477         envioOFF = 0;
478         enviarOFF = false;
479     }
480 }
481 }
482
483 if (stringComplete2) {
484     if (inputString2.substring(0,1) == "{") { //Teste 08/12/2014
485         if (desligou) {
486             recebimento = 0;
487             rodada = 1;
488         }
489         desligou = false;
490         enviarOFF = false;
491         Serial.println("Recebidos dados do CRE@T");
492         if(recebimento >= 54){
493             recebimento = 1;
494             if (rodada >= 10){
495                 rodada = 1;
496                 Serial.println("Contagem de Rodadas Reiniciada");
497             } else {
498                 rodada++;
499                 Serial.println("Nova Rodada Iniciada");
500             }
501         } else {
502             recebimento++;
503         }
504         Serial.print("Rodada/Recebimento = ");
505         Serial.print(rodada);
506         Serial.print("/");
507         Serial.println(recebimento);
508         if (rodada == 1){
509             CarregaPadraoTesteBuffer();
```



```
510     Serial.println("Carregado Padrao Sincronismo");
511   } else {
512     if (recebimento == 1){
513       ZeraBufferSatelite();
514       Serial.println("BuffSat carregado / BuffTemp reiniciado");
515     }
516   }
517   if (recebimento <= 6){
518     SomaBufferSatelite(g1,g2,g3);
519     Serial.print("Regs. dos canais ");
520     Serial.print(g1);
521     Serial.print(", ");
522     Serial.print(g2);
523     Serial.print(", ");
524     Serial.print(g3);
525     Serial.println(" somados no BuffTemp:");
526     Serial.println("-----");
527     Serial.print(" Registro   Trio");
528     Serial.print(g1);
529     Serial.print(" Trio");
530     Serial.print(g2);
531     Serial.print(" Trio");
532     Serial.println(g3);
533     Serial.print("T_Latch23: ");
534     EscreveLargura5(T_Latch23g1);
535     EscreveLargura5(T_Latch23g2);
536     Serial.println("   x   ");
537     Serial.print("T_Latch13: ");
538     EscreveLargura5(T_Latch13g1);
539     EscreveLargura5(T_Latch13g2);
540     Serial.println("   x   ");
541     Serial.print("T_Latch12: ");
542     EscreveLargura5(T_Latch12g1);
543     EscreveLargura5(T_Latch12g2);
544     Serial.println("   x   ");
545     Serial.print("T_Latch3 : ");
546     EscreveLargura5(T_Latch3g1);
547     EscreveLargura5(T_Latch3g2);
548     EscreveLargura5(T_Latch3g3);
549     Serial.println("");
550     Serial.println("-----");
551   }
552   if      (recebimento <= 6){
553     Serial.print("Enviado B_Latch23g1 ");
554     TransmiteSatelite(B_Latch23g1);
555   } else if (recebimento <= 12){
556     Serial.print("Enviado B_Latch13g1 ");
557     TransmiteSatelite(B_Latch13g1);
558   } else if (recebimento <= 18){
559     Serial.print("Enviado B_Latch12g1 ");
560     TransmiteSatelite(B_Latch12g1);
561   } else if (recebimento <= 24){
562     Serial.print("Enviado B_Latch3g1 ");
563     TransmiteSatelite(B_Latch3g1);
564   } else if (recebimento <= 30){
565     Serial.print("Enviado B_Latch23g2 ");
566     TransmiteSatelite(B_Latch23g2);
```

```
567         } else if (recebimento <= 36){
568             Serial.print("Enviado B_Latch13g2 ");
569             TransmiteSatelite(B_Latch13g2);
570         } else if (recebimento <= 42){
571             Serial.print("Enviado B_Latch12g2 ");
572             TransmiteSatelite(B_Latch12g2);
573         } else if (recebimento <= 48){
574             Serial.print("Enviado B_Latch3g2 ");
575             TransmiteSatelite(B_Latch3g2);
576         } else if (recebimento <= 54){
577             Serial.print("Enviado B_Latch3g3 ");
578             TransmiteSatelite(B_Latch3g3);
579         }
580
581         Serial.println("Dados recebidos:");
582         Serial.println(
583             "-----");
584         Serial.println(" Registro   Trio0 Trio1 Trio2 Trio3 Trio4 Trio5 Trio6
585         Trio7");
586         Serial.print("T_Latch23: ");
587         for (int i = 0; i<8 ; i++){
588             EscreveLargura5(conversao(inputString2.substring(0*6+7*6*i+26,0*6+7*6*
589             i+30)));
590         }
591         Serial.println("");
592         Serial.print("T_Latch13: ");
593         for (int i = 0; i<8 ; i++){
594             EscreveLargura5(conversao(inputString2.substring(1*6+7*6*i+26,1*6+7*6*
595             i+30)));
596         }
597         Serial.println("");
598         Serial.print("T_Latch12: ");
599         for (int i = 0; i<8 ; i++){
600             EscreveLargura5(conversao(inputString2.substring(2*6+7*6*i+26,2*6+7*6*
601             i+30)));
602         }
603         Serial.println("");
604         Serial.print("T_Latch3: ");
605         for (int i = 0; i<8 ; i++){
606             EscreveLargura5(conversao(inputString2.substring(6*6+7*6*i+26,6*6+7*6*
607             i+30)));
608         }
609         Serial.println("");
610         Serial.print("FREQ1: ");
611         for (int i = 0; i<8 ; i++){
612             EscreveLargura5(conversao(inputString2.substring(3*6+7*6*i+26,3*6+7*6*
613             i+30)));
614         }
615         Serial.println("");
616         Serial.print("FREQ2: ");
```

```
616     for (int i = 0; i<8 ; i++){
617         EscreveLargura5(conversao(inputString2.substring(5*6+7*6*i+26,5*6+7*6*
        i+30)));
618     }
619     Serial.println("");
620     Serial.println(
        "-----");
621
622     }
623     inputString2 = "";
624     stringComplete2 = false;
625 }
626 }
627
628 void serialEvent2() {
629     if (serial_en2 == true){
630         while (Serial2.available()) {
631             char inChar2 = (char)Serial2.read();
632             if (inChar2 == '\n') {
633                 stringComplete2 = true;
634             }
635             else {
636                 inputString2 += inChar2;
637             }
638         }
639     }
640 }
641
642 void ZeraBufferSatelite(){
643     B_Latch23g1 = T_Latch23g1;
644     B_Latch13g1 = T_Latch13g1;
645     B_Latch12g1 = T_Latch12g1;
646     B_Latch3g1  = T_Latch3g1;
647     B_Latch23g2 = T_Latch23g2;
648     B_Latch13g2 = T_Latch13g2;
649     B_Latch12g2 = T_Latch12g2;
650     B_Latch3g2  = T_Latch3g2;
651     B_Latch3g3  = T_Latch3g3;
652     T_Latch23g1 = 0;
653     T_Latch13g1 = 0;
654     T_Latch12g1 = 0;
655     T_Latch3g1  = 0;
656     T_Latch23g2 = 0;
657     T_Latch13g2 = 0;
658     T_Latch12g2 = 0;
659     T_Latch3g2  = 0;
660     T_Latch3g3  = 0;
661 }
662
663 void SomaBufferSatelite(int g1, int g2, int g3){
664     T_Latch23g1 += conversao(inputString2.substring(0*6+7*6*g1+26,0*6+7*6*g1+30));
665     T_Latch13g1 += conversao(inputString2.substring(1*6+7*6*g1+26,1*6+7*6*g1+30));
666     T_Latch12g1 += conversao(inputString2.substring(2*6+7*6*g1+26,2*6+7*6*g1+30));
667     T_Latch3g1  += conversao(inputString2.substring(6*6+7*6*g1+26,6*6+7*6*g1+30));
668     T_Latch23g2 += conversao(inputString2.substring(0*6+7*6*g2+26,0*6+7*6*g2+30));
669     T_Latch13g2 += conversao(inputString2.substring(1*6+7*6*g2+26,1*6+7*6*g2+30));
670     T_Latch12g2 += conversao(inputString2.substring(2*6+7*6*g2+26,2*6+7*6*g2+30));
```

```
671     T_Latch3g2 += conversao(inputString2.substring(6*6+7*6*g2+26,6*6+7*6*g2+30));
672     T_Latch3g3 += conversao(inputString2.substring(6*6+7*6*g3+26,6*6+7*6*g3+30));
673 }
674
675 void TransmiteSatelite(unsigned long tx_val){
676
677     if (tx_val > 65535) tx_val = 65535;
678
679     Serial.print("para satelite = ");
680     if (tx_val < 10000) Serial.print("0");
681     if (tx_val < 1000) Serial.print("0");
682     if (tx_val < 100) Serial.print("0");
683     if (tx_val < 10) Serial.print("0");
684     Serial.print(tx_val);
685
686     double anal_write_val = 55.73+(tx_val/1000)*1.26;
687     analogWrite(pin_analog, anal_write_val);
688     Serial.print(" PWM=");
689     Serial.print(anal_write_val);
690
691     double mV_OUT = (5000*0.9)*anal_write_val/256;
692     Serial.print(" (");
693     Serial.print(mV_OUT);
694     Serial.println(" mV)");
695 }
696
697 void EscreveLargura5(unsigned long value){
698     Serial.print(" ");
699     if (value < 10000) Serial.print(" ");
700     if (value < 1000) Serial.print(" ");
701     if (value < 100) Serial.print(" ");
702     if (value < 10) Serial.print(" ");
703     Serial.print(value);
704 }
705
706 void CarregaPadraoTesteBuffer(){
707     B_Latch23g1 = 6554;
708     B_Latch13g1 = 13107;
709     B_Latch12g1 = 19661;
710     B_Latch3g1 = 26214;
711     B_Latch23g2 = 32768;
712     B_Latch13g2 = 39321;
713     B_Latch12g2 = 45875;
714     B_Latch3g2 = 52429;
715     B_Latch3g3 = 58982;
716 }
717
718 void serialEvent() {
719     while (Serial.available()) {
720         char inChar = (char)Serial.read();
721         if (inChar == '\n') {
722             stringComplete = true;
723         }
724         else {
725             inputString += inChar;
726         }
727     }
```

```
728 }
729
730 double vTensao(double admInp){
731     return admInp*(fundoEscala/1023);
732 }
733
734 double vTensao1024(double admInp){
735     return (admInp/100)*(1023/fundoEscala);
736 }
737
738 unsigned long conversao(String novo_valor2) {
739     char conv_char2[novo_valor2.length() + 1];
740     novo_valor2.toCharArray(conv_char2, (novo_valor2.length() + 1));
741     return strtoul(conv_char2, NULL, 16);
742 }
743
744 void ConversaoDireta() {
745     String novo_valor = inputString.substring(4);
746     char conv_char[novo_valor.length() + 1];
747     novo_valor.toCharArray(conv_char, (novo_valor.length() + 1));
748     conv_int = atoi(conv_char);
749 }
750
751 void exibeInformacoes(){
752     Serial.println("x-x  INCT da CRIOSFERA - Modulo Criosfera 1  x-x");
753     Serial.println("Controlador integrado do sistema Eolico de Energia");
754     Serial.println("Desenv. por Ulisses Carneiro (ulissestv@gmail.com)");
755     Serial.println("");
756     Serial.println("Comandos Disponiveis");
757     Serial.println("INFO      - Exibe informacoes sobre o sistema");
758     Serial.println("SP1Ixxxx - Set Point freio Turbina 1 (RPM)");
759     Serial.println("SP2Ixxxx - Set Point freio Turbina 2 (RPM)");
760     Serial.println("SP1Oxxxx - Set Point tensao carga da Bateria 1");
761     Serial.println("SP2Oxxxx - Set Point tensao carga da Bateria 2");
762     Serial.println("SP1Mxxxx - Set Point rotacao minima Turbina 1");
763     Serial.println("SP2Mxxxx - Set Point rotacao minima Turbina 2");
764     Serial.println("MODEx    - Modo: 0=Automatico, 1=Manual");
765     Serial.println("MA1Oxxxx - PWM Manual Estagio de Carga Bateria 1");
766     Serial.println("MA2Oxxxx - PWM Manual Estagio de Carga Bateria 2");
767     Serial.println("MA1Ixxxx - PWM Manual Estagio de Freio Turbina 1");
768     Serial.println("MA2Ixxxx - PWM Manual Estagio de Freio Turbina 2");
769     Serial.println("");
770 }
771
772
773 long unsigned to_longuns (double input){
774     long unsigned resultado = input*100;
775     return resultado;
776 }
```

## APENDICE F: Caderno de códigos do Módulo Contador

Este apêndice é composto pelos códigos necessários à compilação no software QUARTUS II do projeto de firmware desenvolvido para o FPGA do módulo contador de eventos coincidentes. Os arquivos estão apensados na seguinte ordem:

- `CREAT_FpgaFirmware_README.txt`
  - » Descreve a árvore de conteúdo da pasta do projeto, listando os mínimos arquivos necessários para efetuar-se a compilação.
- `CREAT_LogicAnalyzer.vhd`
  - » Código VHDL principal, nele estão implementadas todas as funções lógicas que caracterizam o funcionamento do firmware.
- `CREAT_FpgaFirmware.vhd`
  - » Código necessário à implementação da camada de abstração de hardware da CAEN, integrando o FPGA ao board V1495.
- `v1495usr_pkg.vhd`
  - » Pacote de variáveis necessárias à compilação do projeto. Aqui estão descritos os endereços correspondentes a cada registrador no barramento VME.
- `spare_if_rtl.vhd`
  - » Código necessário à implementação da camada de abstração de hardware da CAEN.
- `tristate_if_rtl.vhd`
  - » Código necessário à implementação da camada de abstração de hardware da CAEN.
- `CREAT_FpgaFirmware.qpf`; `CREAT_FpgaFirmware.qsf`; `CREAT_FpgaFirmware.pin`
  - » Arquivos de configuração do projeto gerados pelo software QUARTUS II.

```

1 =====
2
3 ===          CENTRO BRASILEIRO DE PESQUISAS FÍSICAS (CBPF)          ===
4 ===CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA  (CEFET/RJ)===
5
6 =====
7
8   Rio de Janeiro, Maio de 2015
9
10 =====
11
12   Arquivos reunidos em 27/05/2015 na revisão 0027. Neste momento todo o
13   sistema de aquisição e contagem estava operacional, o FPGA era capaz de
14   enviar dados pela porta serial (G0) e o gerador LVDS era capaz de gerar os
15   pulsos (sincronizados em fase) de 100Hz, 66Hz e 33Hz. Esta versão corresponde
16   ao que foi de fato implementado no experimento antártico, com todos os
17   valores finais de calibração definidos.
18
19 =====
20
21   Esta pasta deve conter todos os arquivos referentes ao projeto no software
22   QUARTUS II do Firmware FPGA customizado para Espectômetro de Raios Cósmitos
23   (1a fase) do projeto CREAT (Cosmic Rays Experiment for Weather) a ser
24   instalado na criosfera antártica (LAT 84 SUL).
25
26   Projeto final de curso
27   Aluno: Ulisses de Freitas Carneiro da Graca (ulissestv@gmail.com)
28   Engenharia Eletrônica
29   CENTRO FEDERAL DE EDUCACAO TECNOLOGICA CELSO SUCKOW DA FONSECA
30
31 =====
32
33   Conteúdo (mínimo):
34
35   CREAT_FpgaFirmware_README.txt      : This file.
36
37   SRC\
38     CREAT_FpgaFirmware\              : VHDL source files folder;
39     CREAT_FpgaFirmware.vhd           : Top Level Entity of project;
40     CREAT_LogicAnalyzer.vhd          : Core signal processing logic;
41     spare_if_rtl.vhd                 : Interface entity to the Spare pins;
42     tristate_if_rtl.vhd              : Entity of bidirectional bus;
43     v1495usr_pkg.vhd                 : Package of global project constants;
44     v1495usr_hal.vqm                 : Hardware Abstraction Layer netlist;
45     OLD_Releases\                    : Directory for old core VHDL files;
46
47   FIT\
48     CREAT_FpgaFirmware.qpf           : QuartusII project file.
49     CREAT_FpgaFirmware.qsf           : QuartusII settings file.
50     CREAT_FpgaFirmware.rbf           : Raw Binary File (to be used for
51     user FPGA firmware upgrade).
52     CREAT_FpgaFirmware.pin           : Pinout report from latest fitting.
53     OLD_RBF_Releases\                : Dir. of old Raw Binary File releases;
54
55
56
57

```

```
58 =====
59
60   Requisitos do Sistema:
61   -----
62
63   - Altera QuartusII 11.0 (Web/Full edition)
64     Atencao: Versoes mais novas NAO FUNCIONAM !!!!
65
66 =====
67
68   V1495 BOARD FPGA ARCHITECTURE:
69   -----
70
71   The board hosts two Altera Cyclone FPGA:
72
73   1) The VME FPGA is a EP1C6 device. It handles:
74   * VME interface
75   * USER FPGA interface
76
77   2) The USER FPGA is a EP1C4 device. It handles:
78   * VME FPGA Interface
79   * Delay lines interface
80   * I/O interface
81
82   Each FPGA can be upgraded via VME using the cvUpgrade utility ditributed
83   by CAEN SpA at url http://www.caen.it/csite/CaenProd.jsp?idmod=484&parent=11
84
```



```

1  -- Descricao do projeto:
2  -- *****
3  -- Company:          CENTRO BRASILEIRO DE PESQUISAS FISICAS - CBPF BRASIL
4  -- Project:          ESPECTRMETRO DE RAIOS CSMICOS DA CRIOSFERA
5  --                   CRE@AT - Cosmic Ray Experiment at Antartida
6  -- Model:            V1495 - Multipurpose Programmable Trigger Unit
7  -- FPGA Proj. Name:  CREAT_FpgaFirmware
8  -- Device:           ALTERA EP1C4F400C6
9  -- Author:           Ulisses Carneiro (ulissestv@gmail.com)
10 -- Date:             28-08-2014
11 -- -----
12 -- Based on demo version written by Luca Colombini and available to download by
13 -- CAEN SpA at url http://www.caen.it/csite/CaenProd.jsp?idmod=484&parent=11
14 -- -----
15 -- Module:           CREAT_LogicAnalyzer
16 -- Description:      Detecta coincidencias de eventos em intervalos definidos de
17 --                   tempo entre suas entradas e registra o somatorio da
18 --                   quantidade de eventos ocorridos em determinado intervalo
19 --                   para posterior leitura pelo bus VME. Tambem e capaz de
20 --                   gerar sinais de teste e debug para auto-calibracao.
21 --                   Projeto desenvolvido para aplicacao especifica no
22 --                   experimento detector de raios cosmicos a ser instalado no
23 --                   modulo de pesquisas cientificas CRIOSFERA I, do INCT da
24 --                   CRIOSFERA (LAT S 8400'00" - LON W 7929'39") localizado na
25 --                   plataforma continental antartica.
26 -- *****
27 --
28 -- Revision History:
29 -- #####
30 --   Date      Author          Rev.   Comments
31 -- 28 Ago 14   Ulisses Carneiro  0001   Criacao - Primeira versao
32 -- 29 Ago 14   Ulisses Carneiro  0002   Migrado para estrutura de design HAL da
33 --                                     CAEN e revisao geral e incl. comentarios
34 -- 29 Ago 14   Ulisses Carneiro  0003   Blocos de entrada triplicados,
35 --                                     suporte a I/O NIM e revisao geral
36 -- 02 Set 14   Ulisses Carneiro  0004   Incl. sinais de debug em C_DOUT
37 -- 02 Set 14   Ulisses Carneiro  0005   Aumentada frequencia do gerador
38 -- 02 Set 14   Ulisses Carneiro  0006   Testes com portas NIM
39 -- 02 Set 14   Ulisses Carneiro  0007   Alteracoes no gerador de teste:
40 --                                     - Incluido estado intermediario para
41 --                                     gerar 3 frequencias diferentes
42 --                                     - Incluido parametro de largura do
43 --                                     pulso do sinal de teste
44 -- 03 Set 14   Ulisses Carneiro  0008   Testes com novo metodo para trigger
45 -- 03 Set 14   Ulisses Carneiro  0009   Desfeitos testes do novo metodo de
46 --                                     trigger (rev.0008) e corrigido problema
47 --                                     das portas de entrada.
48 -- 03 Set 14   Ulisses Carneiro  0010   Incluída leitura de registradores
49 --                                     pelo barramento VME (tabela abaixo)
50 -- 03 Set 14   Ulisses Carneiro  0011   Adequacao dos arquivos da HAL ao projeto
51 -- 09 Set 14   Ulisses Carneiro  0012   Implementacao de novo trigger no canal A
52 --                                     (definitivo). GRANDES MODIFICACOES!!!
53 -- 10 Set 14   Ulisses Carneiro  0013   Implementacao de novo metodo de trigger
54 --                                     para todos os canais A,B,C. Incluso de
55 --                                     contadores individuais por entrada para
56 --                                     medicao de rudo. GRANDES MODIFICACOES!!
57 -- 11 Set 14   Ulisses Carneiro  0014   Testes com serial (niveis de tenso TTL)

```

```

58  -- 11 Set 14  Ulisses Carneiro  0015  Implementao do serial
59  -- 16 Set 14  Ulisses Carneiro  0016  Incremento de informaes no serial
60  -- 16 Set 14  Ulisses Carneiro  0017  Mudancas na contagem do tempo de
61  --                                     aquisio. GRANDES MODIFICACOES!!
62  -- 18 Set 14  Ulisses Carneiro  0018  Implementao do std_logic_array para
63  --                                     os sinais de entrada.
64  -- 18 Set 14  Ulisses Carneiro  0019  Incluídos "FOR LOOP" em todos os blocos
65  --                                     de deteo de entrada e contagem.
66  --                                     Firmware 100% escalvel PROFUNDAS
67  --                                     MODIFICACOES EM TODO O CODIGO!!
68  -- 20 Set 14  Ulisses Carneiro  0020  Incluídos "FOR LOOP" na interface UART
69  --                                     do projeto, ordenao no codigo, atuali-
70  --                                     zao de comentarios, reordenacao de
71  --                                     processos em todo o codigo. Alteracoes no
72  --                                     detector de coincidencias para permitir a
73  --                                     existencia de mais de 3 entradas por canal.
74  --                                     Alterados os nomes de varios sinais.
75  --                                     PROFUNDAS MODIFICACOES EM TODO O CODIGO!!
76  -- 07 Nov 14  Ulisses Carneiro  0021  Correcao de bugs na interface UART, defi-
77  --                                     nicao de limite superior nos contadores
78  --                                     (se atinge fundo de escala, para de con-
79  --                                     tar em [(2**(N_NIB*4))-1])
80  -- 07 Nov 14  Ulisses Carneiro  0022  Implementacao de geradores diferenciados
81  --                                     para todos os sinais de saida
82  -- 13 Nov 14  Ulisses Carneiro  0023  Mudanca na contagem do tempo de aquisicao
83  --                                     implementado um subdivisor de tempo antes
84  --                                     do contador de tempo em si. Incluídos
85  --                                     registradores R/W para alteracao "on air"
86  --                                     dos parametros de temporizacao. Incluídas
87  --                                     alteracao dinamica nas saidas NIM (OUT
88  --                                     PATCH) atraves de registrador VME R/W.
89  -- 15 Nov 14  Ulisses Carneiro  0024  Revisao de comentarios, implementacao de
90  --                                     contador para "Unique ID" da amostragem
91  --                                     (ID), inclusao do ID na transmissao
92  --                                     serial, inclusao de transmissoes seriais
93  --                                     a cada segundo para monitoracao do funcio-
94  --                                     namento interno da firmware, inclusao de
95  --                                     vetor de enderecamento dos canais da FE
96  --                                     para facilitar manipulacoes no ordenamen-
97  --                                     to dos canais fisicos de E/S.
98  -- 21 Nov 14  Ulisses Carneiro  0025  Inclusao do somatorio de crosstalk
99  -- 22 Nov 14  Ulisses Carneiro  0026  Inclusao do canal extra de analises
100 -- 09 Dez 14  Ulisses Carneiro  0027  Definição dos valores-padrão definitivos
101 --
102 --
103 -- #####
104 -- #                               DESCRICAO DO PROTOCOLO SERIAL                               #
105 -- #####
106 -- #                               #
107 -- # 1) TODA A INFORMACAO SEMPRE E TRANSMITIDA EM CARACTERES ASCII                               #
108 -- # 2) O TAMANHO TOTAL DAS TRANSMISSOES VARIA EM FUNCAO DOS PARAMETROS                               #
109 -- #   N_NIB, N_CANAIS E N_INPUTS DO FIRMWARE                               #
110 -- # 3) A TRANSMISSAO SEGUE O SEGUINTE PROTOCOLO:                               #
111 -- #   3.1) START BYTE DA TRANSMISSAO:                               #
112 -- #     3.1.1) START DE UMA TRANSMISSAO DE STATUS:                               #
113 -- #       UMA TRANSMISSAO DE STATUS (AQUELA QUE OCORRE A CADA SEGUNDO)                               #
114 -- #       SEMPRE SE INICIA PELO CARACTERE x"7B" - ASCII "{ "                               #

```

```

115 -- # 3.1.1) START DE UMA TRANSMISSAO DE REGISTRADORES: #
116 -- # UMA TRANSMISSAO DE REGISTRADORES (OCORRE A CADA PERIODO DE #
117 -- # AQUISICAO) SEMPRE SE INICIA PELO CARACTERE x"7B" - ASCII "|" #
118 -- # 3.2) TRANSMISSAO DOS REGISTRADORES: #
119 -- # 3.2.1) ID BYTE DE UM REGISTRADOR: #
120 -- # A TRANSMISSAO DE UM REGISTRADOR DE "N_NIB" BYTES SEMPRE SE INICIA #
121 -- # POR UM BYTE DE IDENTIFICACAO, QUE COMECA EM x"20" - ASCII "SPACE" #
122 -- # PARA O PRIMEIRO BYTE A SER TRANSMITIDO E RECEBE INCREMENTO +1 A #
123 -- # BYTE QUE E TRANSMITIDO, OU SEJA, x"20", x"21", x"22"... #
124 -- # 3.2.2) TRANSMISSAO DO REGISTRADOR EM SI: #
125 -- # UM REGISTRADOR COM LARGURA DE N_NIB*4 BITS SERA SEMPRE TRANSMI- #
126 -- # TIDO DE QUATRO EM QUATRO BITS, INICIANDO PELO NIBBLE MAIS SIGNI- #
127 -- # FICATIVO SENDO O VALOR DE CADA NIBBLE REPRESENTADO POR UM SIGNO #
128 -- # ASCII CORRESPONDENTE, PODENDO ESTE VARIAR DE "0" A "F" PARA #
129 -- # INDICAR, RESPECTIVAMENTE, NIBBLES CUJOS VALORES VARIAM DE "0000" #
130 -- # A "1111".A TRANSMISSAO DE UM REGISTRADOR DE N_NIB*4 BITS SEMPRE #
131 -- # SIGNIFICARA A TRANSMISSAO DE N_NIBs BYTES PELA INTERFACE SERIAL #
132 -- # 3.2.3) STOP BYTE DE UM REGISTRADOR: #
133 -- # APOS A TRANSMISSAO DO ULTIMO BYTE DE UM REGISTRADOR E TRANSMITIDO #
134 -- # O CARACTERE x"7D" - ASCII "}". SE HOVEREM AINDA MAIS REGISTRA- #
135 -- # DORES A SEREM TRANSMITIDOS RETORNA-SE AO ITEM 3.2 E SE REPETE A #
136 -- # TRANSMISSAO PARA TODOS OS PROXIMOS REGISTRADORES #
137 -- # 3.5) ENCERRAMENTO DE UMA TRANSMISSAO #
138 -- # APOS A TRANSMISSAO DO ULTIMO BYTE DO ULTIMO REGISTRADOR, E APOS A #
139 -- # TRANSMISSAO DO STOP BYTE REFERENTE A ESTE, SERAO TRANSMITIDOS 3 #
140 -- # CARACTERES ASCII EM SEQUENCIA: #
141 -- # - x"7E" - ASCII "~" - MARCA O FIM DA TRANSMISSAO #
142 -- # - x"0D" - ASCII \r - RETORNO DE CARRO NO ASCII #
143 -- # - x"0A" - ASCII \n - ALIMENTACAO DE LINHA NO ASCII #
144 -- # 4) UMA TRANSMISSAO DE STATUS (A CADA UM SEGUNDO) SE DA NA SEGINTE ORDEM: #
145 -- # 4.1) UNIQUE ID DO PERIODO DE AQUISICAO #
146 -- # 4.2) TEMPO DECORRIDO, EM SEGUNDOS, DO PERIODO DE AQUISICAO ATUAL #
147 -- # 5) UMA TRANSMISSAO COMPLETA SE DA NA SEGINTE ORDEM: #
148 -- # 5.1) UNIQUE ID DO PERIODO DE AQUISICAO #
149 -- # 5.2) INTERVALO, EM SEGUNDOS, DO ULTIMO PERIODO DE AQUISICAO #
150 -- # 5.3) LARGURA DO GATE (PARAMETRO N_GATE) #
151 -- # 5.4) PALAVRA DE CONTROLE DO CANAL DE ANALISE DE DADOS (CTL_ANALISE) #
152 -- # 5.5) REGISTRADORES DE CONTAGEM, NA SEGUINTE ORDEM, E REPETINDO #
153 -- # POR N_CANAIS VEZES #
154 -- # 5.5.1) N_INPUTS VALORES DE COINCIDENCIA [N-1] - 1o AO ULTIMO INPUT #
155 -- # 5.5.2) N_INPUTS VALORES DE MEDIDA DE RUIDO - 1o AO ULTIMO INPUT #
156 -- # 5.5.3) 1 VALOR DE COINCIDENCIA TOTAL #
157 -- # 5.6) CHECKSUM #
158 -- # #
159 -- ##### #
160 -- # MAPA DE REGISTRADORES (VME BUS) #
161 -- ##### #
162 -- #-----#
163 -- # REGISTRADOR* | FUNCAO #
164 -- # #
165 -- # Registradores do Somatorio instantaneo de eventos coincidentes: #
166 -- # 0x1000 | Produto do somatorio COINC_1a23 (instantaneo) #
167 -- # 0x1002 | Produto do somatorio COINC_1a13 (instantaneo) #
168 -- # 0x1004 | Produto do somatorio COINC_1a12 (instantaneo) #
169 -- # 0x1006 | Produto do somatorio COINC_Ta (instantaneo) #
170 -- # 0x1008 | Produto do somatorio COINC_1b23 (instantaneo) #
171 -- # 0x100A | Produto do somatorio COINC_1b13 (instantaneo) #

```

```

172  -- # 0x100C | Produto do somatorio COINC_1b12 (instantaneo) #
173  -- # 0x100E | Produto do somatorio COINC_Tb (instantaneo) #
174  -- # 0x1010 | Produto do somatorio COINC_1c23 (instantaneo) #
175  -- # 0x1012 | Produto do somatorio COINC_1c13 (instantaneo) #
176  -- # 0x1014 | Produto do somatorio COINC_1c12 (instantaneo) #
177  -- # 0x1016 | Produto do somatorio COINC_Tc (instantaneo) #
178  -- # 0x1018 | Produto do somatorio COINC_1d23 (instantaneo) #
179  -- # 0x101A | Produto do somatorio COINC_1d13 (instantaneo) #
180  -- # 0x101C | Produto do somatorio COINC_1d12 (instantaneo) #
181  -- # 0x101E | Produto do somatorio COINC_Td (instantaneo) #
182  -- # 0x1020 | Produto do somatorio COINC_1e23 (instantaneo) #
183  -- # 0x1022 | Produto do somatorio COINC_1e13 (instantaneo) #
184  -- # 0x1024 | Produto do somatorio COINC_1e12 (instantaneo) #
185  -- # 0x1026 | Produto do somatorio COINC_Te (instantaneo) #
186  -- # 0x1028 | Produto do somatorio COINC_1f23 (instantaneo) #
187  -- # 0x102A | Produto do somatorio COINC_1f13 (instantaneo) #
188  -- # 0x102C | Produto do somatorio COINC_1f12 (instantaneo) #
189  -- # 0x102E | Produto do somatorio COINC_Tf (instantaneo) #
190  -- # #
191  -- # Registradores da soma de coincidencias no ultimo periodo de aquisicao: #
192  -- # 0x1100 | Valor armazenado em LATCH2a23 (COINC_1a23 no ultimo T*) #
193  -- # 0x1102 | Valor armazenado em LATCH2a13 (COINC_1a13 no ultimo T*) #
194  -- # 0x1104 | Valor armazenado em LATCH2a12 (COINC_1a12 no ultimo T*) #
195  -- # 0x1106 | Valor armazenado em LATCH3a (COINC-Ta no ultimo T*) #
196  -- # 0x1108 | Valor armazenado em LATCH2b23 (COINC_1b23 no ultimo T*) #
197  -- # 0x110A | Valor armazenado em LATCH2b13 (COINC_1b13 no ultimo T*) #
198  -- # 0x110C | Valor armazenado em LATCH2b12 (COINC_1b12 no ultimo T*) #
199  -- # 0x110E | Valor armazenado em LATCH3b (COINC_Tb no ultimo T*) #
200  -- # 0x1110 | Valor armazenado em LATCH2c23 (COINC_1c23 no ultimo T*) #
201  -- # 0x1112 | Valor armazenado em LATCH2c13 (COINC_1c13 no ultimo T*) #
202  -- # 0x1114 | Valor armazenado em LATCH2c12 (COINC_1c12 no ultimo T*) #
203  -- # 0x1116 | Valor armazenado em LATCH3c (COINC_Tc no ultimo T*) #
204  -- # 0x1118 | Valor armazenado em LATCH2d23 (COINC_1d23 no ultimo T*) #
205  -- # 0x111A | Valor armazenado em LATCH2d13 (COINC_1d13 no ultimo T*) #
206  -- # 0x112C | Valor armazenado em LATCH2d12 (COINC_1d12 no ultimo T*) #
207  -- # 0x112E | Valor armazenado em LATCH3d (COINC_Td no ultimo T*) #
208  -- # 0x1120 | Valor armazenado em LATCH2e23 (COINC_1e23 no ultimo T*) #
209  -- # 0x1122 | Valor armazenado em LATCH2e13 (COINC_1e13 no ultimo T*) #
210  -- # 0x1124 | Valor armazenado em LATCH2e12 (COINC_1e12 no ultimo T*) #
211  -- # 0x1126 | Valor armazenado em LATCH3e (COINC_Te no ultimo T*) #
212  -- # 0x1128 | Valor armazenado em LATCH2f23 (COINC_1f23 no ultimo T*) #
213  -- # 0x112A | Valor armazenado em LATCH2f13 (COINC_1f13 no ultimo T*) #
214  -- # 0x112C | Valor armazenado em LATCH2f12 (COINC_1f12 no ultimo T*) #
215  -- # 0x112E | Valor armazenado em LATCH3f (COINC_Tf no ultimo T*) #
216  -- # #
217  -- # Registradores do Somatorio instantaneo de eventos por entrada: #
218  -- # 0x1200 | Somatorio instantaneo de eventos na entrada 1a #
219  -- # 0x1202 | Somatorio instantaneo de eventos na entrada 2a #
220  -- # 0x1204 | Somatorio instantaneo de eventos na entrada 3a #
221  -- # 0x1208 | Somatorio instantaneo de eventos na entrada 1b #
222  -- # 0x120A | Somatorio instantaneo de eventos na entrada 2b #
223  -- # 0x120C | Somatorio instantaneo de eventos na entrada 3b #
224  -- # 0x1210 | Somatorio instantaneo de eventos na entrada 1c #
225  -- # 0x1212 | Somatorio instantaneo de eventos na entrada 2c #
226  -- # 0x1214 | Somatorio instantaneo de eventos na entrada 3c #
227  -- # 0x1218 | Somatorio instantaneo de eventos na entrada 1d #
228  -- # 0x121A | Somatorio instantaneo de eventos na entrada 2d #

```

```

229  -- # 0x121C | Somatorio instantaneo de eventos na entrada 3d #
230  -- # 0x1220 | Somatorio instantaneo de eventos na entrada 1e #
231  -- # 0x1222 | Somatorio instantaneo de eventos na entrada 2e #
232  -- # 0x1224 | Somatorio instantaneo de eventos na entrada 3e #
233  -- # 0x1228 | Somatorio instantaneo de eventos na entrada 1f #
234  -- # 0x122A | Somatorio instantaneo de eventos na entrada 2f #
235  -- # 0x122C | Somatorio instantaneo de eventos na entrada 3f #
236  -- # # #
237  -- # Registradores da soma de eventos por entrada no ultimo periodo de ruido: #
238  -- # 0x1300 | Quantidade de eventos na entrada 1a no ultimo R* #
239  -- # 0x1302 | Quantidade de eventos na entrada 2a no ultimo R* #
240  -- # 0x1304 | Quantidade de eventos na entrada 3a no ultimo R* #
241  -- # 0x1308 | Quantidade de eventos na entrada 1b no ultimo R* #
242  -- # 0x130A | Quantidade de eventos na entrada 2b no ultimo R* #
243  -- # 0x130C | Quantidade de eventos na entrada 3b no ultimo R* #
244  -- # 0x1310 | Quantidade de eventos na entrada 1c no ultimo R* #
245  -- # 0x1312 | Quantidade de eventos na entrada 2c no ultimo R* #
246  -- # 0x1314 | Quantidade de eventos na entrada 3c no ultimo R* #
247  -- # 0x1318 | Quantidade de eventos na entrada 1d no ultimo R* #
248  -- # 0x131A | Quantidade de eventos na entrada 2d no ultimo R* #
249  -- # 0x131C | Quantidade de eventos na entrada 3d no ultimo R* #
250  -- # 0x1320 | Quantidade de eventos na entrada 1e no ultimo R* #
251  -- # 0x1322 | Quantidade de eventos na entrada 2e no ultimo R* #
252  -- # 0x1324 | Quantidade de eventos na entrada 3e no ultimo R* #
253  -- # 0x1328 | Quantidade de eventos na entrada 1f no ultimo R* #
254  -- # 0x132A | Quantidade de eventos na entrada 2f no ultimo R* #
255  -- # 0x132C | Quantidade de eventos na entrada 3f no ultimo R* #
256  -- # # #
257  -- # REGISTRADORES DE LEITURA E ESCRITA (R/W): #
258  -- # # #
259  -- # 0x2000 | Tempo do intervalo de aquisicao em segundos #
260  -- # 0x2002 | Tempo do gate das entradas em unidades de clock #
261  -- # 0x2004 | Controle do PATCH de saida dos conectores NIM #
262  -- # 0x2006 | UNIQUE ID do ultimo conjunto de registradores transmitido #
263  -- # 0x2008 | Palavra de controle do canal de analise de dados #
264  -- # # #
265  -- # ----- #
266  -- # * Endereco completo = base address do board + registrador de tabela #
267  -- # T* T = Intervalo de aquisicao definido pelo periodo [N_AQUISICAO]*25[ns]#
268  -- # R* R = Intervalo de aquisicao definido pelo periodo [N_MED_RUIDO]*25[ns]#
269  -- # #####
270
271  LIBRARY ieee;
272  USE ieee.std_logic_1164.all;
273  USE ieee.std_logic_arith.all;
274  USE ieee.std_logic_unsigned.all;
275  USE ieee.std_logic_misc.all; -- (Para utilizar a funo OR_REDUCE)
276  USE work.v1495pkg.all; -- Contem o mapa de registradores do BUS VME
277
278  ENTITY CREAT_LogicAnalyzer IS
279  PORT(
280  -- ORIGINAL DO CODIGO DA CAEN (NAO MODIFICAR!!!):
281  nLBRES : IN std_logic; -- Async Reset (act L)
282  LCLK : IN std_logic; -- Local Bus Clock
283  -----
284  -- REGISTER INTERFACE
285  -----

```

```

286     REG_WREN      : IN      std_logic;           -- Write pulse (act H)
287     REG_RDEN      : IN      std_logic;           -- Read pulse (act H)
288     REG_ADDR      : IN      std_logic_vector (15 DOWNT0 0);-- Register address
289     REG_DIN       : IN      std_logic_vector (15 DOWNT0 0);-- Data from Local Bus
290     REG_DOUT      : OUT     std_logic_vector (15 DOWNT0 0);-- Data to Local Bus
291     USR_ACCESS    : IN      std_logic;           -- Current reg. access
292                                           -- at user address space
293     --*****
294     -- V1495 Front Panel Ports (PORT A,B,C,G)
295     --*****
296     A_DIN         : IN      std_logic_vector (31 DOWNT0 0);-- In A (32 x LVDS/ECL)
297     B_DIN         : IN      std_logic_vector (31 DOWNT0 0);-- In B (32 x LVDS/ECL)
298     C_DOUT        : OUT     std_logic_vector (31 DOWNT0 0);-- Out C (32 x LVDS)
299     G_LEV         : OUT     std_logic;           -- Output Level(NIM/TTL)
300     G_DIR         : OUT     std_logic;           -- Output Enable
301     G_DOUT        : OUT     std_logic_vector (1 DOWNT0 0); -- Out G - (2 x NIM/TTL)
302     G_DIN         : IN      std_logic_vector (1 DOWNT0 0); -- In G - (2 x NIM/TTL)
303     --*****
304     -- A395x MEZZANINES INTERFACES (PORT D,E,F)
305     --*****
306     -- Expansion Mezzanine Identifier:
307     -- x_IDCODE :
308     -- 000 : A395A (32 x IN LVDS/ECL)
309     -- 001 : A395B (32 x OUT LVDS)
310     -- 010 : A395C (32 x OUT ECL)
311     -- 011 : A395D (8 x IN/OUT NIM/TTL)
312
313     -- Expansion Mezzanine Port Signal Standard Select
314     -- x_LEV :
315     -- 0=>TTL,1=>NIM
316
317     -- Expansion Mezzanine Port Direction
318     -- x_DIR :
319     -- 0=>OUT,1=>IN
320
321     -- In/Out D (I/O Expansion)
322     D_IDCODE      : IN      std_logic_vector ( 2 DOWNT0 0);-- D slot mezzanine ID
323     D_LEV         : OUT     std_logic;           -- D slot Port Level Sel
324     D_DIR         : OUT     std_logic;           -- D slot Port Direction
325     D_DIN         : IN      std_logic_vector (31 DOWNT0 0);-- D slot Data In Bus
326     D_DOUT        : OUT     std_logic_vector (31 DOWNT0 0);-- D slot Data Out Bus
327     -- In/Out E (I/O Expansion)
328     E_IDCODE      : IN      std_logic_vector ( 2 DOWNT0 0);-- E slot mezzanine ID
329     E_LEV         : OUT     std_logic;           -- E slot Port Level Sel
330     E_DIR         : OUT     std_logic;           -- E slot Port Direction
331     E_DIN         : IN      std_logic_vector (31 DOWNT0 0);-- E slot Data In Bus
332     E_DOUT        : OUT     std_logic_vector (31 DOWNT0 0);-- E slot Data Out Bus
333     -- In/Out F (I/O Expansion)
334     F_IDCODE      : IN      std_logic_vector ( 2 DOWNT0 0);-- F slot mezzanine ID
335     F_LEV         : OUT     std_logic;           -- F slot Port Level Sel
336     F_DIR         : OUT     std_logic;           -- F slot Port Direction
337     F_DIN         : IN      std_logic_vector (31 DOWNT0 0);-- F slot Data In Bus
338     F_DOUT        : OUT     std_logic_vector (31 DOWNT0 0);-- F slot Data Out Bus
339     --*****
340     -- DELAY LINES
341     --*****
342     -- PDL = Programmable Delay Lines (Step = 0.25ns / FSR = 64ns)

```

```

343     -- DLO = Delay Line Oscillator      (Half Period ~ 10 ns)
344     -- 3D3428 PDL (PROGRAMMABLE DELAY LINE) CONFIGURATION
345     PDL_WR      : OUT    std_logic;          -- Write Enable
346     PDL_SEL     : OUT    std_logic;          -- PDL Sel. (PDL0/PDL1)
347     PDL_READ    : IN     std_logic_vector ( 7 DOWNT0 0); -- Read Data
348     PDL_WRITE   : OUT    std_logic_vector ( 7 DOWNT0 0); -- Write Data
349     PDL_DIR     : OUT    std_logic;          -- Direction (0>W 1>R)
350     -- DELAY I/O
351     PDL0_OUT    : IN     std_logic;          -- Signal from PDL0 Out
352     PDL1_OUT    : IN     std_logic;          -- Signal from PDL1 Out
353     DLO0_OUT    : IN     std_logic;          -- Signal from DLO0 Out
354     DLO1_OUT    : IN     std_logic;          -- Signal from DLO1 Out
355     PDL0_IN     : OUT    std_logic;          -- Signal to PDL0 In
356     PDL1_IN     : OUT    std_logic;          -- Signal to PDL1 In
357     DLO0_GATE  : OUT    std_logic;          -- DLO0 Gate (active H)
358     DLO1_GATE  : OUT    std_logic;          -- DLO1 Gate (active H)
359     --*****
360     -- SPARE PORTS
361     --*****
362     SPARE_OUT   : OUT    std_logic_vector(11 downto 0); -- SPARE Data Out
363     SPARE_IN    : IN     std_logic_vector(11 downto 0); -- SPARE Data In
364     SPARE_DIR   : OUT    std_logic_vector(11 downto 0); -- SPARE Dir(0>OUT 1>IN)
365     --*****
366     -- LED
367     --*****
368     RED_PULSE   : OUT    std_logic;          -- RED led (active H)
369     GREEN_PULSE : OUT    std_logic;          -- GREEN led(active H)
370 );
371
372 -- Declarations
373
374 END CREAT_LogicAnalyzer ;
375
376 ARCHITECTURE rtl OF CREAT_LogicAnalyzer IS
377
378 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
379 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
380 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
381 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
382 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
383
384 -- CONSTANTES:
385
386 type in_ch_array is array(0 to 2, 0 to 4) of integer range 0 to 100;
387
388 constant IN_PATCH :in_ch_array := ((15-00, 15-03, 15-04, 15-10, 15-12),
389                                     (15-01, 15-05, 15-08, 15-11, 15-13),
390                                     (15-02, 15-06, 15-09, 15-15, 15-14));
391
392 constant N_NIB :integer := 4; -- Preciso dos contadores. l nibble = 4bits
393 constant N_CANAIS :integer := 8; -- Define qte. de canais
394 constant N_INPUTS :integer := 3; -- Define qte. de entradas por canal
395 constant C_GATE :unsigned(N_NIB*4-1 downto 0):= conv_unsigned(30, N_NIB*4);
396 signal N_GATE :unsigned(N_NIB*4-1 downto 0):= C_GATE;
397 -- TEMPO DE AQUISICAO:
398 constant N_T_1SEG :unsigned:= conv_unsigned(4000000, 32);
399 -- 1 unidade de N_T_1SEG equivale a 25ns.

```

```

400 -- Valor tipico: 40000000 = 1seg
401 constant C_TAQUIS :unsigned:= conv_unsigned(00600, 16);
402 -- 1 unidade de N_TAQUIS equivale a N_T_1SEG*25ns (IDEALMENTE 1s).
403 -- Valores tipicos: 00010 = 10seg
404 --                      00020 = 20seg
405 --                      00120 = 2min
406 --                      00600 = 10min
407 --                      01800 = 30min
408 --                      03600 = 60min
409 signal N_TAQUIS :unsigned(15 DOWNTO 0):=C_TAQUIS;
410 signal N_TAQUIS_STD:std_logic_vector(15 DOWNTO 0);--:=C_TAQUIS;
411 constant N_MED_RUIDO :integer := 1; -- 1 N_MED_RUIDO = N_T_1SEG*25ns (1s)
412 constant N_GEN_TTOTAL:integer := 400000; -- Intervalo entre pulsos do gerador
413 constant N_GEN_LPULSO:integer := 2; -- Largura de pulso do gerador
414 constant N_LCLK_FREQ :integer := 40000000;
415 constant N_SER_BRATE :integer := 4167; -- 4166,666666666667 p/ 9600 baud/seg
416 signal N_BYTES :integer;
417
418 constant C_CTL_ANAL :std_logic_vector(15 downto 0):= x"2123";
419
420 constant BYTES_EXTRAS :integer := 4; -- U_ID, T_AQUIS, N_GATE e CTL_ANALISE
421
422
423 -- EXEMPLO DE CALCULO DA FREQUENCIA DO GERADOR:
424 -- fLCLK = 40 MHz >>> fGEN = 40MHz / (N_GEN_TTOTAL)
425 -- N_GEN_TTOTAL = fLCLK / fGEN = 40x10^6 / 10
426 -- Para fGEN = 10 Hz temos que N_GEN_TTOTAL = 4x10^6 = 4000000
427
428 -- SINAIS:
429
430 -- Declaraes de tipos especificos para nosso circuito::
431 -- Foram criados vrios tipos em array, formando matrizes em que o numero da
432 -- linha representa o CANAL (a, b, c, d, e...) e o nmero da coluna representa
433 -- a ENTRADA (0, 1, 2...). A combinacao COLUNA-LINHA resulta no identificador de
434 -- sinal (0a, 1a, 2a, 0b, 1b...) utilizado nos demais equipamentos do projeto.
435 --
436 -- |----ID CANAL----| |---ID ENTRADA---|
437 type std_logic_array is array(N_CANAIS-1 downto 0, N_INPUTS-1 downto 0)
438 of std_logic;
439 type unsigned_array is array(N_CANAIS-1 downto 0, N_INPUTS-1 downto 0)
440 of unsigned(N_NIB*4-1 downto 0);
441 type unsigned_vector is array(N_CANAIS-1 downto 0
442 of unsigned(N_NIB*4-1 downto 0);
443 type vector_array is array( N_CANAIS-1
444 of std_logic_vector(N_INPUTS-1 downto 0);
445 type serial_buffer is array(BYTES_EXTRAS+(N_CANAIS*(2*N_INPUTS+1)-1) downto 0)
446 of unsigned(N_NIB*4-1 downto 0);
447 type ascii_array is array( N_NIB-1
448 of unsigned(7 downto 0);
449
450
451 -- Sinais individuais de entrada direta:
452 -- diretamente aqui que so injetados os sinais LVDS que vem da Front-End:
453 signal INPUT : std_logic_array;
454 signal IN_RUIDO : std_logic;
455

```



```
456 -- Sinal de RESET dos contadores de GATE TIME:
457 -- responsvel por resetar o GATE de cada uma dos canais, quando o contador GATE
458 -- de qualquer uma das entradas de determinado canal acionado haver o acrescimo
459 -- de +1 a cada LCLK, at chegar ao limite N_GATE. Nesta condio este sinal ir
460 -- ordenar o reinicio de contagem.
461 signal DET_CDOWNRESET      : std_logic_vector(N_CANAIS-1 downto 0);
462
463 -- Mquina lgica combinacional de deteo de sinal:
464 -- Para implementar o circuito de GATE de temporizao assncrona foi preciso
465 -- criar um circuito auxiliar capaz de detectar a presena ou no de um sinal
466 -- vlido em contagem pelo canal. Para isso foi utilizada lgica combinacional
467 -- simples do tipo ou-exclusivo. Como cada uma das entradas de cada um dos canais
468 -- deve ser monitorada, este um sinal do tipo (type) std_logic_array, definido
469 -- por ns como uma matriz de CANAIS vs. ENTRADAS, como dito acima.
470 signal DET_SFLAG1        : std_logic_array;
471 signal DET_SFLAG2        : std_logic_array;
472 signal DET_RFLAG1        : std_logic;
473 signal DET_RFLAG2        : std_logic;
474 signal DET_SINAL         : vector_array;
475 signal DET_RUIDO         : std_logic;
476
477 -- Sinais de detecao de coincidencias:
478 -- So os sinais que apontam (instantaneamente) a informao se houve ou no
479 -- coincidncia de eventos entre as entradas de determinado canal:
480 signal COINC_T           : std_logic_vector(N_CANAIS-1 downto 0); -- Coincidencia total
481 signal COINC_1          : std_logic_array;                       -- Coincidencia N-1
482
483 -- DEBBUGER dos Sinais de detecao de evento individual por canal:
484 -- Vai a nvel alto quando houve evento na entrada especificada por sua linha e
485 -- coluna. atualizado toda vez que ocorre uma deteco (no final do GATE).
486 signal DET_IN           : std_logic_array; -- Evento individual
487
488
489 -- DEBBUGER dos Sinais de detecao de coincidencias:
490 -- Mesma operao do sinal acima, porm, para as deteces de coincidncia:
491 signal DET_CT           : std_logic_vector(N_CANAIS-1 downto 0); -- Coincidencia total
492 signal DET_C1          : std_logic_array;                       -- Coincidencia N-1
493
494 -- Registradores de somatorio dos eventos coincidentes:
495 signal DET_CDOWN        : unsigned_vector;
496 signal CDOWNLIMITE      : std_logic_vector(N_CANAIS-1 downto 0);
497
498 -- Registradores de somatorio dos eventos brutos:
499 signal SUM_IN           : unsigned_array;
500 signal SUM_CROSS        : unsigned_array;
501
502 -- Registradores de somatorio dos eventos coincidentes:
503 signal SUM_CT           : unsigned_vector;
504 signal SUM_C1          : unsigned_array;
505
506 -- Registradores (Latch's) para captura da informacao pelo BUS VME:
507 signal LATCH_C3         : unsigned_vector;
508 signal LATCH_C2        : unsigned_array;
509
510 -- Registradores (Latch's) para captura da informacao pelo BUS VME:
511 signal LATCH_IN         : unsigned_array;
512
```

```

513  -- SERIAL UART OUTPUT:
514  signal SER_CONT_BRATE      : integer range 0 to 1000000000:=0;
515  signal SER_CONT_BIT       : integer range 0 to 1000000000:=0;
516  signal SER_CONT_BYTE      : integer range 1 to 1000000001:=1;
517
518  signal SER_TXGATE          : std_logic:='0';
519  signal SER_TXACTIVE        : std_logic:='1';
520  signal SER_DOUT            : std_logic;
521  signal SER_TXACTIVE_FLAG1  : std_logic:='1';
522  signal SER_TXACTIVE_FLAG2  : std_logic:='0';
523
524  signal SER_DBUFFER         : unsigned(7 downto 0);
525  signal SER_DIN             : unsigned(N_NIB*4-1 downto 0);
526  signal U_ID                : unsigned(N_NIB*4-1 downto 0):=conv_unsigned(0, N_NIB*4
);
527  signal CHECKSUM            : unsigned(N_NIB*4-1 downto 0);
528  signal TEMPO_AQUIS         : unsigned(N_NIB*4-1 downto 0);
529
530  signal SER_PARIDADE        : std_logic;
531
532  signal SER_ASCII_CONV      : ascii_array;
533
534  signal BUFFER_SERIAL       : serial_buffer;
535
536  -- SINAIS DE CONTROLE:
537
538  -- Processamento da contagem:
539  signal LED_DEBUG1          : std_logic; -- LED_DEBUG
540  signal LED_DEBUG2          : std_logic; -- LED_DEBUG
541
542  -- Gerador de sinais:
543  signal CLK_STATE           : std_logic; -- Clock da Maquina de Estados do Gerador
544  signal GO_GEN              : std_logic; -- Pulso de disparo do gerador
545  signal GEN_FLAG1           : std_logic:='1'; -- Elemento da maquina de estados do gerador
546  signal GEN_FLAG2           : std_logic:='0'; -- Elemento da maquina de estados do gerador
547  signal GEN1                : std_logic; -- Canal 1a da saida do gerador.
548  signal GEN2                : std_logic; -- Canal 2a da saida do gerador.
549  signal GEN3                : std_logic; -- Canal 3a da saida do gerador.
550  signal GEN1a               : std_logic; -- Canal 1a da saida do gerador.
551  signal GEN2a               : std_logic; -- Canal 2a da saida do gerador.
552  signal GEN3a               : std_logic; -- Canal 3a da saida do gerador.
553  signal GEN1b               : std_logic; -- Canal 1b da saida do gerador.
554  signal GEN2b               : std_logic; -- Canal 2b da saida do gerador.
555  signal GEN3b               : std_logic; -- Canal 3b da saida do gerador.
556  signal GEN1c               : std_logic; -- Canal 1c da saida do gerador.
557  signal GEN2c               : std_logic; -- Canal 2c da saida do gerador.
558  signal GEN3c               : std_logic; -- Canal 3c da saida do gerador.
559  signal GEN1d               : std_logic; -- Canal 1d da saida do gerador.
560  signal GEN2d               : std_logic; -- Canal 2d da saida do gerador.
561  signal GEN3d               : std_logic; -- Canal 3d da saida do gerador.
562  signal GEN1e               : std_logic; -- Canal 1e da saida do gerador.
563  signal GEN2e               : std_logic; -- Canal 2e da saida do gerador.
564  signal GEN3e               : std_logic; -- Canal 3e da saida do gerador.
565  signal CONT_GEN_POSICAO    : integer range 1 to 1000000001:=1; -- Perodo do gerador
566
567  -- Definicao de tipo de estados para gerador de sinais:
568  type STATE_type is

```

```

569         (S0, S1, S2);
570     signal STATE: STATE_type;
571
572     -- DIVISORES DE CLOCK:
573     signal CONT_T_1SEG      : unsigned(31 DOWNT0 0):=conv_unsigned(0, 32); -- Tempo de
aquisio
574     signal CONT_TAQUIS     : unsigned(15 DOWNT0 0):=conv_unsigned(0, 16); -- Tempo de
aquisio
575     signal CONT_GEN_TTOTAL : integer range 1 to 1000000001:=1; -- Perodo do gerador
576     signal CONT_GEN_LPULSO : integer range 0 to 1000000000:=0; -- Larg. pulso gerador
577
578     signal GO_1SEG        : std_logic:='1'; -- Pulso de disparo do contador de T.
Aquisicao
579
580     signal PATCH_NIM      : integer range 0 to N_CANAIS-1 := 0;
581     signal CTL_ANALISE    : std_logic_vector(15 downto 0):=C_CTL_ANAL;
582
583     begin -- incio da arquitetura
584
585     -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
586     -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
587     -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
588
589
590     --*****
591     -- PORT Levels: '0'=TTL '1'=NIM
592     D_LEV <= '1';
593     E_LEV <= '1';
594     F_LEV <= '1';
595     G_LEV <= '0';
596
597     --*****
598     -- PORT Directions: '0'=Output '1'=Input
599     D_DIR <= '0';
600     E_DIR <= '0'; ----ATENCAO!!!!!!!
601     F_DIR <= '1';
602     G_DIR <= '0';
603
604
605     -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
606     -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
607     -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
608
609     RED_PULSE <= LED_DEBUG1;
610     GREEN_PULSE <= LED_DEBUG2;
611
612     -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
613     -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
614     -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
615
616     INPUT(0,0) <= A_DIN(IN_PATCH(0,0)); -- Entrada 1a
617     INPUT(0,1) <= A_DIN(IN_PATCH(1,0)); -- Entrada 2a
618     INPUT(0,2) <= A_DIN(IN_PATCH(2,0)); -- Entrada 3a
619     INPUT(1,0) <= A_DIN(IN_PATCH(0,1)); -- Entrada 1b
620     INPUT(1,1) <= A_DIN(IN_PATCH(1,1)); -- Entrada 2b
621     INPUT(1,2) <= A_DIN(IN_PATCH(2,1)); -- Entrada 3b
622     INPUT(2,0) <= A_DIN(IN_PATCH(0,2)); -- Entrada 1c

```

```

623 INPUT(2,1) <= A_DIN(IN_PATCH(1,2)); -- Entrada 2c
624 INPUT(2,2) <= A_DIN(IN_PATCH(2,2)); -- Entrada 3c
625 INPUT(3,0) <= A_DIN(IN_PATCH(0,3)); -- Entrada 1d
626 INPUT(3,1) <= A_DIN(IN_PATCH(1,3)); -- Entrada 2d
627 INPUT(3,2) <= A_DIN(IN_PATCH(2,3)); -- Entrada 3d
628 INPUT(4,0) <= A_DIN(IN_PATCH(0,4)); -- Entrada 1e
629 INPUT(4,1) <= A_DIN(IN_PATCH(1,4)); -- Entrada 2e
630 INPUT(4,2) <= A_DIN(IN_PATCH(2,4)); -- Entrada 3e
631 INPUT(5,0) <= A_DIN(16); -- Entrada 1f
632 INPUT(5,1) <= A_DIN(17); -- Entrada 2f
633 INPUT(5,2) <= A_DIN(18); -- Entrada 3f
634
635 -- Conexoes eletricas estaticas das saidas do nosso gerador:
636 C_DOUT(IN_PATCH(0,0)) <= GEN1a; -- Saida 1 do Gerador
637 C_DOUT(IN_PATCH(1,0)) <= GEN2a; -- Saida 2 do Gerador
638 C_DOUT(IN_PATCH(2,0)) <= GEN3a; -- Saida 3 do Gerador
639 C_DOUT(IN_PATCH(0,1)) <= GEN1b; -- Saida 1 do Gerador
640 C_DOUT(IN_PATCH(1,1)) <= GEN2b; -- Saida 2 do Gerador
641 C_DOUT(IN_PATCH(2,1)) <= GEN3b; -- Saida 3 do Gerador
642 C_DOUT(IN_PATCH(0,2)) <= GEN1c; -- Saida 1 do Gerador
643 C_DOUT(IN_PATCH(1,2)) <= GEN2c; -- Saida 2 do Gerador
644 C_DOUT(IN_PATCH(2,2)) <= GEN3c; -- Saida 3 do Gerador
645 C_DOUT(IN_PATCH(0,3)) <= GEN1d; -- Saida 1 do Gerador
646 C_DOUT(IN_PATCH(1,3)) <= GEN2d; -- Saida 2 do Gerador
647 C_DOUT(IN_PATCH(2,3)) <= GEN3d; -- Saida 3 do Gerador
648 C_DOUT(IN_PATCH(0,4)) <= GEN1e; -- Saida 1 do Gerador
649 C_DOUT(IN_PATCH(1,4)) <= GEN2e; -- Saida 2 do Gerador
650 C_DOUT(IN_PATCH(2,4)) <= GEN3e; -- Saida 3 do Gerador
651 C_DOUT(16) <= GEN1; -- Saida 1 do Gerador
652 C_DOUT(17) <= GEN2; -- Saida 2 do Gerador
653 C_DOUT(18) <= GEN3; -- Saida 3 do Gerador
654
655 -- NIM OUTPUT PATCH:
656 D_DOUT(7) <= INPUT(PATCH_NIM,0);
657 D_DOUT(6) <= INPUT(PATCH_NIM,1);
658 D_DOUT(5) <= INPUT(PATCH_NIM,2);
659 D_DOUT(4) <= DET_SINAL(PATCH_NIM)(0);
660 D_DOUT(3) <= DET_SINAL(PATCH_NIM)(1);
661 D_DOUT(2) <= DET_SINAL(PATCH_NIM)(1);
662 D_DOUT(1) <= IN_RUIDO;
663 D_DOUT(0) <= LCLK;
664
665 E_DOUT(7) <= COINC_1(PATCH_NIM, 0);
666 E_DOUT(6) <= COINC_1(PATCH_NIM, 1);
667 E_DOUT(5) <= COINC_1(PATCH_NIM, 2);
668 E_DOUT(4) <= COINC_T(PATCH_NIM);
669 E_DOUT(3) <= DET_C1(PATCH_NIM, 0);
670 E_DOUT(2) <= DET_C1(PATCH_NIM, 1);
671 E_DOUT(1) <= DET_C1(PATCH_NIM, 2);
672 E_DOUT(0) <= SER_DOUT;
673
674 IN_RUIDO <= F_DIN(7); -- Entrada de Rudo (VETO do contador)
675 INPUT(6,0) <= F_DIN(6); -- Entrada de teste
676 INPUT(6,1) <= F_DIN(5); -- Entrada de teste
677 INPUT(6,2) <= F_DIN(4); -- Entrada de teste
678
679 process (COINC_T,COINC_1,DET_CDOWNRESET,CTL_ANALISE)

```

```

680  begin
681  if (CTL_ANALISE(15 downto 12) = "0000") then
682      INPUT(7,0) <= DET_CDOWNRESET(conv_integer(CTL_ANALISE(11 downto 8)));
683      INPUT(7,1) <= DET_CDOWNRESET(conv_integer(CTL_ANALISE( 7 downto 4)));
684      INPUT(7,2) <= DET_CDOWNRESET(conv_integer(CTL_ANALISE( 3 downto 0)));
685  elsif (CTL_ANALISE(15 downto 12) = "0001") then
686      INPUT(7,0) <= COINC_1(conv_integer(CTL_ANALISE(11 downto 8)),0) or
687          COINC_1(conv_integer(CTL_ANALISE(11 downto 8)),1) or
688          COINC_1(conv_integer(CTL_ANALISE(11 downto 8)),2);
689      INPUT(7,1) <= COINC_1(conv_integer(CTL_ANALISE( 7 downto 4)),0) or
690          COINC_1(conv_integer(CTL_ANALISE( 7 downto 4)),1) or
691          COINC_1(conv_integer(CTL_ANALISE( 7 downto 4)),2);
692      INPUT(7,2) <= COINC_1(conv_integer(CTL_ANALISE( 3 downto 0)),0) or
693          COINC_1(conv_integer(CTL_ANALISE( 3 downto 0)),1) or
694          COINC_1(conv_integer(CTL_ANALISE( 3 downto 0)),2);
695  elsif (CTL_ANALISE(15 downto 12) = "0010") then
696      INPUT(7,0) <= COINC_T(conv_integer(CTL_ANALISE(11 downto 8)));
697      INPUT(7,1) <= COINC_T(conv_integer(CTL_ANALISE( 7 downto 4)));
698      INPUT(7,2) <= COINC_T(conv_integer(CTL_ANALISE( 3 downto 0)));
699  elsif (CTL_ANALISE(15 downto 12) = "0011") then
700      INPUT(7,0) <= GEN1;
701      INPUT(7,1) <= GEN2;
702      INPUT(7,2) <= GEN3;
703  end if;
704  end process;
705
706  G_DOUT(0) <= SER_DOUT;
707  G_DOUT(1) <= SER_DOUT;
708
709  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
710  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
711  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
712  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
713  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
714  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
715  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
716
717  -- TRIGGER DO SINAL DE ENTRADA:
718  -- Implementamos um mecanismo de deteo do sinal de entrada por bordo com o
719  -- objetivo de estabilizar o contador e eliminar efeitos de atraso de propagacao
720  -- provenientes da eletronica de FRON-END do detector. Isto foi feito incluindo
721  -- uma especie de flip-flop tipo T, cuja entrada T est permanentemente conectada
722  -- ao nivel logico 1. Os clocks destes F-Fs esto ligados diretamente s portas de
723  -- entrada LVDS e suas sadas so ligadas a portas XOR que recebem tambm sinais
724  -- que se alteram sempre que o contador "DET_CDOWN" atinge o valor de "N_GATE".
725  -- Observe que estes dois sinais ("DET_SFLAG1" xor "DET_SFLAG2") so combinados de
726  -- modo a criar um terceiro ("DET_SINAL"), que o sinal de entrada j tratado
727  -- propriamente dito.
728  process(INPUT,DET_SINAL)
729  begin
730  for c in 0 to (N_CANAIS-1) loop --Para todos os canais
731      for i in 0 to (N_INPUTS-1) loop --E para todas as entradas
732          if (INPUT(c,i)'event) and (INPUT(c,i)='1') then --Se ocorre um evento
733              if (DET_SINAL(c)(i) = '0') then --E entrada no est ativa
734                  DET_SFLAG1(c,i) <= not DET_SFLAG1(c,i); --A FLAG1 muda de estado
735              end if;
736          end if;

```

```

737     end loop;
738 end loop;
739 end process;
740
741 -- Amostragem de rudo:
742 -- Adicionamos uma entrada no sistema com a inteno de eliminar os efeitos
743 -- da gerao de rudo da fonte de alta tenso na contagem de coincidncia. A
744 -- fonte HV emite, devido a caractersticas intrnsecas a seu projeto eletrnico,
745 -- pulsos de grande amplitude numa frequencia relativamente estvel. Estes acabam
746 -- interferindo nos circuitos comparadores ultra-sensveis da eletrnica FRONTEND
747 -- e manifestam-se no sinal LVDS como eventos de coincidncia tripla. Quando
748 -- ocorre um bordo de subida no sinal "IN_RUIDO" uma flag acionada e aguarda
749 -- at a prxima ocorrncia de um bordo de SUBIDA do LCLK para zerar DET_CDOWN e
750 -- todo o sinal "DET_SINAL". Com isso, VETADA a contagem de coincidncia sempre
751 -- que houver um evento de subida do sinal "IN_RUIDO". A janela de veto inicia
752 -- no processo abaixo e termina na prxima subida do LCLK, portanto, pode durar
753 -- at 25ns.
754 process(IN_RUIDO)
755 begin
756 if (IN_RUIDO'event) and (IN_RUIDO='1') then
757     if DET_RUIDO = '0' then
758         DET_RFLAG1 <= not DET_RFLAG1;
759     end if;
760 end if;
761 end process;
762
763 DET_RUIDO <= DET_RFLAG1 xor DET_RFLAG2;
764
765 process(LCLK,DET_CDOWNRESET,DET_SINAL)
766 begin
767 for c in 0 to (N_CANAIS-1) loop           --Para todos os canais
768     if (DET_CDOWNRESET(c)='0') then      --Se o reset est ativo (0)
769         DET_CDOWN(c) <= conv_unsigned(0, N_NIB*4); --Zera o contador
770     elsif (LCLK'event) and (LCLK='1') then --Se no, e se ocorre evento
771         DET_CDOWN(c) <= DET_CDOWN(c) + 1; --E incrementa o DET_CDOWN
772     end if;
773 end loop;
774 end process;
775
776 process(DET_SFLAG1,DET_SFLAG2,DET_SINAL)
777 begin
778 for c in 0 to (N_CANAIS-1) loop           --Para todos os canais
779     for i in 0 to (N_INPUTS-1) loop      --E para todas as entradas
780         DET_SINAL(c)(i) <= DET_SFLAG1(c,i) xor DET_SFLAG2(c,i);--XOR (ver acima)
781     end loop;
782     DET_CDOWNRESET(c) <= or_REDUCE(DET_SINAL(c)); --Produz sinal de RESET
783 end loop;
784 end process;
785 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
786 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
787 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
788 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
789 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
790 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
791 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
792
793

```

```
794
795
796 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
797 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
798 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
799 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
800 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
801 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
802 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
803
804
805 -- DETECTOR DE COINCIDENCIA:
806 -- Esta seo do circuito responsvel por detectar eventos coincidentes. Foram
807 -- definidos dois tipos de coincidncia: 1) Coincidncia Total e 2) Coincidncia
808 -- todos menos um (N-1). A coincidncia total obedece a condio na qual ocorreram
809 -- eventos em TODAS as N_INPUTS entradas de um determinado canal. A coincidncia
810 -- N-1 obedece a condio de ocorrerenca de eventos coincidentes PELO MENOS entre
811 -- N-1 entradas de determinado canal. Ou seja, sempre que a coin. total for vlida
812 -- todas as coin. N-1 sero vlidas, mas o contrrio no se aplica. Por exemplo,
813 -- uma abertura de GATE na qual ocorreram eventos nas entradas 0 e 1 do canal 0
814 -- para um parametro N_INPUTS = 3 resultar em COINC_T(0)='1', COINC_1(0,0)='0',
815 -- COINC_1(0,1)='0' e COINC_1(0,2)='1'. Releases anteriores deste FIRMWARE
816 -- utilizavam as seguintes nomeclaturas para estes sinais:
817 -- COINC_T(0) -> COIN3a
818 -- COINC_1(0,0) -> COIN2a_23
819 -- COINC_1(0,1) -> COIN2a_13
820 -- COINC_1(0,2) -> COIN2a_12
821 -- A extino desta nomeclatura e adoo da atual estrutura de arrays e funes
822 -- FOR LOOP foram resultado do esforo em tornar esta FIRMWARE parametrizvel por
823 -- constantes e viabilizar seu crescimento.
824 process (DET_SINAL)
825 begin
826 for c in 0 to (N_CANAIS-1) loop
827 -- Coincidencia TOTAL:
828 COINC_T(c) <= and_reduce(DET_SINAL(c));
829 for i in 0 to (N_INPUTS-1) loop
830 -- Coincidencia N-1:
831 COINC_1(c,i) <=
832 and_reduce(DET_SINAL(c) or (conv_std_logic_vector(2**i, N_INPUTS)));
833 end loop;
834 end loop;
835 end process;
836 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
837 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
838 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
839 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
840 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
841 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
842 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
843
844
845
846 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
847 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
848 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
849 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
850 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```

851  -- XXXXXXXXXXXXXXXXXXXX          INCIO DO CDIGO VHDL          XXXXXXXXXXXXXXXXXXXX
852  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
853  --
854  -- O trecho de codigo que se segue e responsavel por implementar os blocos de
855  -- contagem propriamente ditos. Podemos observar que o somatorio incrementado
856  -- do se, e somente se, na ocorrencia de um bordo de subida do clock fundamental
857  -- houver nivel logico '1' no detector de coincidencia do respectivo contador.
858  -- Observa-se tambem se o contador "CONT_TAQUIS" = 0 e "CONT_T_1SEG" = 2 para
859  -- reiniciar todas as contagens. Isto ocorre a cada N_TAQUIS * N_T_1SEG pulsos
860  -- do clock fundamental (LCLK).
861  process (DET_CDOWN,N_GATE)
862  begin
863  for c in 0 to (N_CANAIS-1) loop
864      if (DET_CDOWN(c) >= N_GATE) then
865          CDOWNLIMITE(c) <= '1';
866      else
867          CDOWNLIMITE(c) <= '0';
868      end if;
869  end loop;
870  end process;
871
872
873
874
875  process (CONT_TAQUIS,CONT_T_1SEG,LCLK,CDOWNLIMITE,DET_SINAL,COINC_1,COINC_T,
DET_SFLAG1)
876  begin
877  if (CONT_TAQUIS = 0) and (CONT_T_1SEG = 2) then -- ++ SE O CONTADOR DO TEMPO DE
AQUISICAO++
878      for c in 0 to (N_CANAIS-1) loop -- ++ TIVER REINICIADO E FOR
IGUAL A 2 ++
879          for i in 0 to (N_INPUTS-1) loop -- ++ ZERA TODOS OS CONTADORES DE
EVENTOS++
880              SUM_C1(c,i) <= conv_unsigned(0, N_NIB*4); -- Zera SUM_C1
881              SUM_IN(c,i) <= conv_unsigned(0, N_NIB*4); -- Zera SUM_IN
882              SUM_CROSS(c,i) <= conv_unsigned(0, N_NIB*4); -- Zera SUM_IN
883          end loop;
884          SUM_CT(c) <= conv_unsigned(0, N_NIB*4); -- Zera SUM_CT
885      end loop;
886  -- Reinicia o sinal do GATE (signal DET_SINAL (= FLAG1 xor FLAG2)):
887  DET_SFLAG2 <= DET_SFLAG1;
888  else
889  if (LCLK'event) and (LCLK='0') then
890      for c in 0 to (N_CANAIS-1) loop
891          if (CDOWNLIMITE(c) = '1') then
892              for i in 0 to (N_INPUTS-1) loop
893                  if (DET_SINAL(c)(i) = '1') then -- Se detecta evento na entrada
894                      if (conv_unsigned(SUM_IN(c,i), N_NIB*4) <
conv_unsigned(2**(N_NIB*4)-1, N_NIB*4) ) then
895                          SUM_IN(c,i) <= SUM_IN(c,i)+1; -- Incrementa +1 na soma SUM_IN
896                      end if;
897                  end if;
898
899
900
901                  if (or_REDUCE(DET_CDOWNRESET(N_CANAIS-1 downto 0) and (not(
conv_std_logic_vector(2**c, N_CANAIS)))) = '1') then
902                      SUM_CROSS(c,i) <= SUM_CROSS(c,i)+1; -- Incrementa +1 na

```



```

        soma SUM_CROSS
903     end if;
904
905
906
907
908     DET_IN(c,i) <= '1'; -- Sinal para DEBUG do contador
909     else
910     DET_IN(c,i) <= '0'; -- Sinal para DEBUG do contador
911     end if;
912     if (COINC_l(c,i) = '1') then      -- Se detecta coincidência N-1
913     if (conv_unsigned(SUM_Cl(c,i),    N_NIB*4) <
914         conv_unsigned(2**(N_NIB*4)-1, N_NIB*4) ) then
915     SUM_Cl(c,i) <= SUM_Cl(c,i)+1; -- Incrementa +1 na soma SUM_Cl
916     end if;
917     DET_Cl(c,i) <= '1'; -- Sinal para DEBUG do contador
918     else
919     DET_Cl(c,i) <= '0'; -- Sinal para DEBUG do contador
920     end if;
921 end loop;
922 if (COINC_T(c) = '1') then      -- Se detecta coincidência total
923 if (conv_unsigned(SUM_CT(c),    N_NIB*4) <
924     conv_unsigned(2**(N_NIB*4)-1, N_NIB*4) ) then
925 SUM_CT(c) <= SUM_CT(c)+1; -- Incrementa +1 na soma SUM_CT
926 end if;
927 DET_CT(c) <= '1'; -- Sinal para DEBUG do contador
928 else
929 DET_CT(c) <= '0'; -- Sinal para DEBUG do contador
930 end if;
931 end if;
932 end loop;
933 end if;
934 if (LCLK'event) and (LCLK='1') then
935 if DET_RUIDO='1' then
936     DET_SFLAG2 <=     DET_SFLAG1;
937     DET_RFLAG2 <= not DET_RFLAG2;
938     else
939     for c in 0 to (N_CANAIS-1) loop
940     if (CDOWNLIMITE(c) = '1') then
941     for i in 0 to (N_INPUTS-1) loop
942     if (DET_SINAL(c)(i)='1') then
943     -- Reinicia o sinal do GATE (DET_SINAL = FLAG1 xor FLAG2):
944     DET_SFLAG2(c,i) <= not DET_SFLAG2(c,i);
945     end if;
946     end loop;
947     end if;
948     end loop;
949     end if;
950     end if;
951 end if;
952 end process;
953
954
955
956 process(LCLK)
957 begin
958     if (LCLK'event) and (LCLK='0') then

```

```

959     if CONT_T_1SEG < (N_T_1SEG - 1) then
960         CONT_T_1SEG <= CONT_T_1SEG + 1;
961         GO_1SEG <= '1';
962     else
963         CONT_T_1SEG <= conv_unsigned(0, 32);
964         GO_1SEG <= '0';
965     end if;
966 end if;
967 end process;
968
969
970 process(GO_1SEG)
971 begin
972     if (GO_1SEG'event) and (GO_1SEG='0') then
973         if CONT_TAQUIS < (N_TAQUIS - 1) then
974             CONT_TAQUIS <= CONT_TAQUIS + 1;
975             TEMPO_AQUIS <= conv_unsigned(CONT_TAQUIS+1, N_NIB*4);
976         else
977             CONT_TAQUIS <= conv_unsigned(0, 16);
978             U_ID <= U_ID +1;
979         end if;
980     end if;
981 end process;
982 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
983 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
984 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
985 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
986 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
987 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
988 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
989
990
991
992 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
993 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
994 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
995 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
996 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
997 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
998 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
999 process (LCLK)
1000 begin
1001 if (LCLK'event) and (LCLK='1') then
1002     if (CONT_TAQUIS = 0) and (CONT_T_1SEG = 1) then
1003         LED_DEBUG1 <= not(LED_DEBUG1); -- Aciona LED indicador de atividade
1004         LATCH_C2 <= SUM_C1; -- Faz o registro da quantidade de eventos no
1005         LATCH_C3 <= SUM_CT; -- momento em que atinge o tempo de aquisicao
1006     end if;
1007     if (CONT_TAQUIS = N_MED_RUIDO) and (CONT_T_1SEG = 1) then
1008         LED_DEBUG2 <= not(LED_DEBUG2); -- Aciona LED indicador de atividade
1009         LATCH_IN <= SUM_IN; -- Registra a frequencia do rudo
1010     end if;
1011 end if;
1012 end process;
1013 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1014 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1015 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

```
1016 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1017 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1018 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1019 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1020
1021
1022
1023
1024 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1025 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1026 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1027 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1028 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1029 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1030 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1031 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1032 --
1033 P_RREG: process(LCLK, nLBRES)
1034 begin
1035 if (nLBRES = '0') then
1036     REG_DOUT <= (others => '0');
1037 elseif LCLK'event and LCLK = '1' then
1038     if (REG_RDEN = '1') and (USR_ACCESS = '1') then
1039         case REG_ADDR is
1040             -- Registradores do Somatorio instantaneo de eventos coincidentes:
1041             when A_SUM2a23 => REG_DOUT <= conv_std_logic_vector(SUM_C1(0,0), 16);
1042             when A_SUM2a13 => REG_DOUT <= conv_std_logic_vector(SUM_C1(0,1), 16);
1043             when A_SUM2a12 => REG_DOUT <= conv_std_logic_vector(SUM_C1(0,2), 16);
1044             when A_SUM3a    => REG_DOUT <= conv_std_logic_vector(SUM_CT(0),    16);
1045             when A_SUM2b23 => REG_DOUT <= conv_std_logic_vector(SUM_C1(1,0), 16);
1046             when A_SUM2b13 => REG_DOUT <= conv_std_logic_vector(SUM_C1(1,1), 16);
1047             when A_SUM2b12 => REG_DOUT <= conv_std_logic_vector(SUM_C1(1,2), 16);
1048             when A_SUM3b    => REG_DOUT <= conv_std_logic_vector(SUM_CT(1),    16);
1049             when A_SUM2c23 => REG_DOUT <= conv_std_logic_vector(SUM_C1(2,0), 16);
1050             when A_SUM2c13 => REG_DOUT <= conv_std_logic_vector(SUM_C1(2,1), 16);
1051             when A_SUM2c12 => REG_DOUT <= conv_std_logic_vector(SUM_C1(2,2), 16);
1052             when A_SUM3c    => REG_DOUT <= conv_std_logic_vector(SUM_CT(2),    16);
1053             when A_SUM2d23 => REG_DOUT <= conv_std_logic_vector(SUM_C1(3,0), 16);
1054             when A_SUM2d13 => REG_DOUT <= conv_std_logic_vector(SUM_C1(3,1), 16);
1055             when A_SUM2d12 => REG_DOUT <= conv_std_logic_vector(SUM_C1(3,2), 16);
1056             when A_SUM3d    => REG_DOUT <= conv_std_logic_vector(SUM_CT(3),    16);
1057             when A_SUM2e23 => REG_DOUT <= conv_std_logic_vector(SUM_C1(4,0), 16);
1058             when A_SUM2e13 => REG_DOUT <= conv_std_logic_vector(SUM_C1(4,1), 16);
1059             when A_SUM2e12 => REG_DOUT <= conv_std_logic_vector(SUM_C1(4,2), 16);
1060             when A_SUM3e    => REG_DOUT <= conv_std_logic_vector(SUM_CT(4),    16);
1061             when A_SUM2f23 => REG_DOUT <= conv_std_logic_vector(SUM_C1(5,0), 16);
1062             when A_SUM2f13 => REG_DOUT <= conv_std_logic_vector(SUM_C1(5,1), 16);
1063             when A_SUM2f12 => REG_DOUT <= conv_std_logic_vector(SUM_C1(5,2), 16);
1064             when A_SUM3f    => REG_DOUT <= conv_std_logic_vector(SUM_CT(5),    16);
1065             when A_SUM2g23 => REG_DOUT <= conv_std_logic_vector(SUM_C1(6,0), 16);
1066             when A_SUM2g13 => REG_DOUT <= conv_std_logic_vector(SUM_C1(6,1), 16);
1067             when A_SUM2g12 => REG_DOUT <= conv_std_logic_vector(SUM_C1(6,2), 16);
1068             when A_SUM3g    => REG_DOUT <= conv_std_logic_vector(SUM_CT(6),    16);
1069             when A_SUM2h23 => REG_DOUT <= conv_std_logic_vector(SUM_C1(7,0), 16);
1070             when A_SUM2h13 => REG_DOUT <= conv_std_logic_vector(SUM_C1(7,1), 16);
1071             when A_SUM2h12 => REG_DOUT <= conv_std_logic_vector(SUM_C1(7,2), 16);
1072             when A_SUM3h    => REG_DOUT <= conv_std_logic_vector(SUM_CT(7),    16);
```

```
1073 -- Registradores da soma de coincidncias no ltimo periodo de aquisio:
1074 when A_LATCH2a23 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(0,0), 16);
1075 when A_LATCH2a13 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(0,1), 16);
1076 when A_LATCH2a12 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(0,2), 16);
1077 when A_LATCH3a => REG_DOUT <= conv_std_logic_vector(LATCH_C3(0), 16);
1078 when A_LATCH2b23 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(1,0), 16);
1079 when A_LATCH2b13 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(1,1), 16);
1080 when A_LATCH2b12 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(1,2), 16);
1081 when A_LATCH3b => REG_DOUT <= conv_std_logic_vector(LATCH_C3(1), 16);
1082 when A_LATCH2c23 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(2,0), 16);
1083 when A_LATCH2c13 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(2,1), 16);
1084 when A_LATCH2c12 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(2,2), 16);
1085 when A_LATCH3c => REG_DOUT <= conv_std_logic_vector(LATCH_C3(2), 16);
1086 when A_LATCH2d23 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(3,0), 16);
1087 when A_LATCH2d13 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(3,1), 16);
1088 when A_LATCH2d12 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(3,2), 16);
1089 when A_LATCH3d => REG_DOUT <= conv_std_logic_vector(LATCH_C3(3), 16);
1090 when A_LATCH2e23 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(4,0), 16);
1091 when A_LATCH2e13 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(4,1), 16);
1092 when A_LATCH2e12 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(4,2), 16);
1093 when A_LATCH3e => REG_DOUT <= conv_std_logic_vector(LATCH_C3(4), 16);
1094 when A_LATCH2f23 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(5,0), 16);
1095 when A_LATCH2f13 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(5,1), 16);
1096 when A_LATCH2f12 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(5,2), 16);
1097 when A_LATCH3f => REG_DOUT <= conv_std_logic_vector(LATCH_C3(5), 16);
1098 when A_LATCH2g23 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(6,0), 16);
1099 when A_LATCH2g13 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(6,1), 16);
1100 when A_LATCH2g12 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(6,2), 16);
1101 when A_LATCH3g => REG_DOUT <= conv_std_logic_vector(LATCH_C3(6), 16);
1102 when A_LATCH2h23 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(7,0), 16);
1103 when A_LATCH2h13 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(7,1), 16);
1104 when A_LATCH2h12 => REG_DOUT <= conv_std_logic_vector(LATCH_C2(7,2), 16);
1105 when A_LATCH3h => REG_DOUT <= conv_std_logic_vector(LATCH_C3(7), 16);
1106 -- Registradores do Somatorio instantaneo de eventos por entrada:
1107 when A_SUM_IN1a => REG_DOUT <= conv_std_logic_vector(SUM_IN(0,0), 16);
1108 when A_SUM_IN2a => REG_DOUT <= conv_std_logic_vector(SUM_IN(0,1), 16);
1109 when A_SUM_IN3a => REG_DOUT <= conv_std_logic_vector(SUM_IN(0,2), 16);
1110 when A_SUM_IN1b => REG_DOUT <= conv_std_logic_vector(SUM_IN(1,0), 16);
1111 when A_SUM_IN2b => REG_DOUT <= conv_std_logic_vector(SUM_IN(1,1), 16);
1112 when A_SUM_IN3b => REG_DOUT <= conv_std_logic_vector(SUM_IN(1,2), 16);
1113 when A_SUM_IN1c => REG_DOUT <= conv_std_logic_vector(SUM_IN(2,0), 16);
1114 when A_SUM_IN2c => REG_DOUT <= conv_std_logic_vector(SUM_IN(2,1), 16);
1115 when A_SUM_IN3c => REG_DOUT <= conv_std_logic_vector(SUM_IN(2,2), 16);
1116 when A_SUM_IN1d => REG_DOUT <= conv_std_logic_vector(SUM_IN(3,0), 16);
1117 when A_SUM_IN2d => REG_DOUT <= conv_std_logic_vector(SUM_IN(3,1), 16);
1118 when A_SUM_IN3d => REG_DOUT <= conv_std_logic_vector(SUM_IN(3,2), 16);
1119 when A_SUM_IN1e => REG_DOUT <= conv_std_logic_vector(SUM_IN(4,0), 16);
1120 when A_SUM_IN2e => REG_DOUT <= conv_std_logic_vector(SUM_IN(4,1), 16);
1121 when A_SUM_IN3e => REG_DOUT <= conv_std_logic_vector(SUM_IN(4,2), 16);
1122 when A_SUM_IN1f => REG_DOUT <= conv_std_logic_vector(SUM_IN(5,0), 16);
1123 when A_SUM_IN2f => REG_DOUT <= conv_std_logic_vector(SUM_IN(5,1), 16);
1124 when A_SUM_IN3f => REG_DOUT <= conv_std_logic_vector(SUM_IN(5,2), 16);
1125 when A_SUM_IN1g => REG_DOUT <= conv_std_logic_vector(SUM_IN(6,0), 16);
1126 when A_SUM_IN2g => REG_DOUT <= conv_std_logic_vector(SUM_IN(6,1), 16);
1127 when A_SUM_IN3g => REG_DOUT <= conv_std_logic_vector(SUM_IN(6,2), 16);
1128 when A_SUM_IN1h => REG_DOUT <= conv_std_logic_vector(SUM_IN(7,0), 16);
1129 when A_SUM_IN2h => REG_DOUT <= conv_std_logic_vector(SUM_IN(7,1), 16);
```

```
1130     when A_SUM_IN3h => REG_DOUT <= conv_std_logic_vector(SUM_IN(7,2), 16);
1131 -- Registradores da soma de eventos por entrada no ltimo periodo rudo:
1132     when A_LATCH_IN1a => REG_DOUT <= conv_std_logic_vector(LATCH_IN(0,0),16);
1133     when A_LATCH_IN2a => REG_DOUT <= conv_std_logic_vector(LATCH_IN(0,1),16);
1134     when A_LATCH_IN3a => REG_DOUT <= conv_std_logic_vector(LATCH_IN(0,2),16);
1135     when A_LATCH_IN1b => REG_DOUT <= conv_std_logic_vector(LATCH_IN(1,0),16);
1136     when A_LATCH_IN2b => REG_DOUT <= conv_std_logic_vector(LATCH_IN(1,1),16);
1137     when A_LATCH_IN3b => REG_DOUT <= conv_std_logic_vector(LATCH_IN(1,2),16);
1138     when A_LATCH_IN1c => REG_DOUT <= conv_std_logic_vector(LATCH_IN(2,0),16);
1139     when A_LATCH_IN2c => REG_DOUT <= conv_std_logic_vector(LATCH_IN(2,1),16);
1140     when A_LATCH_IN3c => REG_DOUT <= conv_std_logic_vector(LATCH_IN(2,2),16);
1141     when A_LATCH_IN1d => REG_DOUT <= conv_std_logic_vector(LATCH_IN(3,0),16);
1142     when A_LATCH_IN2d => REG_DOUT <= conv_std_logic_vector(LATCH_IN(3,1),16);
1143     when A_LATCH_IN3d => REG_DOUT <= conv_std_logic_vector(LATCH_IN(3,2),16);
1144     when A_LATCH_IN1e => REG_DOUT <= conv_std_logic_vector(LATCH_IN(4,0),16);
1145     when A_LATCH_IN2e => REG_DOUT <= conv_std_logic_vector(LATCH_IN(4,1),16);
1146     when A_LATCH_IN3e => REG_DOUT <= conv_std_logic_vector(LATCH_IN(4,2),16);
1147     when A_LATCH_IN1f => REG_DOUT <= conv_std_logic_vector(LATCH_IN(5,0),16);
1148     when A_LATCH_IN2f => REG_DOUT <= conv_std_logic_vector(LATCH_IN(5,1),16);
1149     when A_LATCH_IN3f => REG_DOUT <= conv_std_logic_vector(LATCH_IN(5,2),16);
1150     when A_LATCH_IN1g => REG_DOUT <= conv_std_logic_vector(LATCH_IN(6,0),16);
1151     when A_LATCH_IN2g => REG_DOUT <= conv_std_logic_vector(LATCH_IN(6,1),16);
1152     when A_LATCH_IN3g => REG_DOUT <= conv_std_logic_vector(LATCH_IN(6,2),16);
1153     when A_LATCH_IN1h => REG_DOUT <= conv_std_logic_vector(LATCH_IN(7,0),16);
1154     when A_LATCH_IN2h => REG_DOUT <= conv_std_logic_vector(LATCH_IN(7,1),16);
1155     when A_LATCH_IN3h => REG_DOUT <= conv_std_logic_vector(LATCH_IN(7,2),16);
1156 -- Registradores do Somatorio instantaneo de eventos por entrada:
1157     when A_SUM_CROSS1a => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(0,0),
1158     16);
1159     when A_SUM_CROSS2a => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(0,1),
1160     16);
1161     when A_SUM_CROSS3a => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(0,2),
1162     16);
1163     when A_SUM_CROSS1b => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(1,0),
1164     16);
1165     when A_SUM_CROSS2b => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(1,1),
1166     16);
1167     when A_SUM_CROSS3b => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(1,2),
1168     16);
1169     when A_SUM_CROSS1c => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(2,0),
1170     16);
1171     when A_SUM_CROSS2c => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(2,1),
1172     16);
1173     when A_SUM_CROSS3c => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(2,2),
1174     16);
1175     when A_SUM_CROSS1d => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(3,0),
1176     16);
1177     when A_SUM_CROSS2d => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(3,1),
1178     16);
1179     when A_SUM_CROSS3d => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(3,2),
1180     16);
1181     when A_SUM_CROSS1e => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(4,0),
1182     16);
1183     when A_SUM_CROSS2e => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(4,1),
1184     16);
1185     when A_SUM_CROSS3e => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(4,2),
1186     16);
```

```

1172     when A_SUM_CROSS1f => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(5,0),
1173     16);
1173     when A_SUM_CROSS2f => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(5,1),
1174     16);
1174     when A_SUM_CROSS3f => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(5,2),
1175     16);
1175     when A_SUM_CROSS1g => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(6,0),
1176     16);
1176     when A_SUM_CROSS2g => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(6,1),
1177     16);
1177     when A_SUM_CROSS3g => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(6,2),
1178     16);
1178     when A_SUM_CROSS1h => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(7,0),
1179     16);
1179     when A_SUM_CROSS2h => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(7,1),
1180     16);
1180     when A_SUM_CROSS3h => REG_DOUT <= conv_std_logic_vector(SUM_CROSS(7,2),
1181     16);
1181     -- Registradores de leitura e escrita:
1182     when A_TAQUIS      => REG_DOUT <= conv_std_logic_vector(N_TAQUIS , 16);
1183     when A_TGATE      => REG_DOUT <= conv_std_logic_vector(N_GATE   , 16);
1184     when A_PATCH_NIM  => REG_DOUT <= conv_std_logic_vector(PATCH_NIM, 16);
1185     when A_U_ID       => REG_DOUT <= conv_std_logic_vector(U_ID    , 16);
1186     when A_CTL_ANAL   => REG_DOUT <= CTL_ANALISE;
1187     -- Outros registradores:
1188     when others       => REG_DOUT <= (others => '0');
1189     end case;
1190     end if;
1191     end if;
1192     end process;
1193
1194     -- WRITE REGISTERS
1195     P_WREG : process(LCLK, nLBRES)
1196     begin
1197     if (nLBRES = '0') then
1198     N_TAQUIS      <= C_TAQUIS; -- Valor default para o tempo de aquisicao
1199     N_GATE        <= C_GATE;   -- Valor default para o tempo de aquisicao
1200     PATCH_NIM     <= 0;
1201     CTL_ANALISE   <= C_CTL_ANAL;
1202     elsif LCLK'event and LCLK = '1' then
1203     if (REG_WREN = '1') and (USR_ACCESS = '1') then
1204     case REG_ADDR is
1205     when A_TAQUIS      => N_TAQUIS      <= unsigned(REG_DIN);
1206     when A_TGATE      => N_GATE        <= unsigned(REG_DIN);
1207     when A_PATCH_NIM  => PATCH_NIM     <= conv_integer(REG_DIN);
1208     when A_CTL_ANAL   => CTL_ANALISE   <= REG_DIN;
1209     when others       => null;
1210     end case;
1211     end if;
1212     end if;
1213     end process;
1214
1215     --N_TAQUIS <= conv_unsigned(N_TAQUIS_STD, 16);
1216
1217     -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1218     -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1219     -- XXXXXXXXXXXXXXXXXXXXXXXX LEITURA DE REGISTRADORES PELO BUS VME XXXXXXXXXXXXXXXXXXXXXXXX

```

```

1220  -- XXXXXXXXXXXXXXXXXXXX                               XXXXXXXXXXXXXXXXXXXX
1221  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1222  -- XXXXXXXXXXXXXXXXXXXX                                FIM DO CODIGO  VHDL      XXXXXXXXXXXXXXXXXXXX
1223  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1224
1225
1226
1227  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1228  -- XXXXXXXXXXXXXXXXXXXX                               XXXXXXXXXXXXXXXXXXXX
1229  -- XXXXXXXXXXXXXXXXXXXX    TRANSMISSO SERIAL DE  REGISTRADORES  XXXXXXXXXXXXXXXXXXXX
1230  -- XXXXXXXXXXXXXXXXXXXX                               XXXXXXXXXXXXXXXXXXXX
1231  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1232  -- XXXXXXXXXXXXXXXXXXXX                                INICIO DO CODIGO VHDL      XXXXXXXXXXXXXXXXXXXX
1233  -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1234  process (LCLK,CONT_TAQUIS,CONT_T_1SEG)
1235  begin
1236  if (LCLK'event) and (LCLK='1') then
1237    if (CONT_T_1SEG = 2) then -- (CONT_TAQUIS = 0) and
1238      SER_TXACTIVE_FLAG1 <= not (SER_TXACTIVE_FLAG1);
1239    end if;
1240  end if;
1241  end process;
1242
1243  process (LCLK)
1244  begin
1245  if (LCLK'event) and (LCLK='0') then
1246    if (SER_CONT_BRATE < N_SER_BRATE) then
1247      SER_CONT_BRATE <= SER_CONT_BRATE + 1;
1248      SER_TXGATE <= '0';
1249    else
1250      SER_CONT_BRATE <= 0;
1251      SER_TXGATE <= '1';
1252    end if;
1253  end if;
1254  end process;
1255
1256  process (CONT_TAQUIS,SER_TXACTIVE_FLAG1,SER_TXACTIVE_FLAG2)
1257  begin
1258  if (CONT_TAQUIS = 0) then
1259    N_BYTES <= BYTES_EXTRAS*(2+N_NIB)             -- U_ID, T_AQUIS, N_GATE e
1260    CTL_ANALISE                                     -- U_ID, T_AQUIS, N_GATE e
1261            +((2+N_NIB)*((2*N_INPUTS+1)*N_CANAIS)) -- COUNTs
1262            +(2+N_NIB)                             -- CHECKSUM
1263            +4;                                     -- START/END
1264  else
1265    N_BYTES <= 2*(2+N_NIB)                         -- U_ID e T_AQUIS
1266            +4;                                     -- START/END
1267  end if;
1268  SER_TXACTIVE <= SER_TXACTIVE_FLAG1 xor SER_TXACTIVE_FLAG2;
1269  end process;
1270
1271  process (SER_TXGATE,SER_TXACTIVE)
1272  begin
1273  if (SER_TXACTIVE='0') then
1274    SER_CONT_BIT <= 0;
1275    SER_CONT_BYTE <= 1;
1276    CHECKSUM <= conv_unsigned(0, N_NIB*4);

```

```

1276     else
1277         if (SER_TXGATE'event) and (SER_TXGATE='0') then
1278             if (SER_CONT_BIT < (12)) then
1279                 SER_CONT_BIT <= SER_CONT_BIT + 1;
1280             else
1281                 SER_CONT_BIT <= 1;
1282                 SER_CONT_BYTE <= SER_CONT_BYTE + 1;
1283             end if;
1284         end if;
1285         if (SER_TXGATE'event) and (SER_TXGATE='1') then
1286             if (SER_CONT_BIT = (12)) and (SER_CONT_BYTE < (N_BYTES-(3+(2+N_NIB)))) then
1287                 CHECKSUM <= CHECKSUM + SER_DBUFFER;
1288             end if;
1289         end if;
1290     end if;
1291 end process;
1292
1293 process (SER_TXGATE,SER_TXACTIVE)
1294 begin
1295     if (SER_TXACTIVE='1') then
1296         if (SER_TXGATE'event) and (SER_TXGATE='1') then
1297             if (SER_CONT_BYTE = 1) and (SER_CONT_BIT = 1) then
1298                 BUFFER_SERIAL(0) <= U_ID;
1299                 if (CONT_TAQUIS = 0) then
1300                     BUFFER_SERIAL(1) <= TEMPO_AQUIS +1;
1301                 else
1302                     BUFFER_SERIAL(1) <= TEMPO_AQUIS;
1303                 end if;
1304                 BUFFER_SERIAL(2) <= N_GATE;
1305                 BUFFER_SERIAL(3) <= unsigned(CTL_ANALISE);
1306                 for c in 0 to (N_CANAIS-1) loop
1307                     for i in 0 to (N_INPUTS-1) loop
1308                         BUFFER_SERIAL(BYTES_EXTRAS+(c*(2*N_INPUTS+1))+i) <=
1309                             LATCH_C2(c,i);
1310                         BUFFER_SERIAL(BYTES_EXTRAS+(c*(2*N_INPUTS+1))+i+(N_INPUTS)) <=
1311                             LATCH_IN(c,i);
1312                     end loop;
1313                     BUFFER_SERIAL(BYTES_EXTRAS+(2*N_INPUTS)+(c*(2*N_INPUTS+1))) <=
1314                         LATCH_C3(c);
1315                 end loop;
1316             end if;
1317         end if;
1318     end if;
1319 end process;
1320
1321 process (SER_TXGATE,SER_TXACTIVE)
1322 begin
1323     if (SER_TXACTIVE='1') then
1324         if (SER_TXGATE'event) and (SER_TXGATE='1') then
1325             if (SER_CONT_BYTE = N_BYTES) and (SER_CONT_BIT = 12) then
1326                 SER_TXACTIVE_FLAG2 <= not SER_TXACTIVE_FLAG2;
1327             end if;
1328         end if;
1329     end if;
1330 end process;
1331
1332 process (SER_CONT_BYTE,BUFFER_SERIAL,CHECKSUM)

```



```

1333 begin
1334 for n in 0 to (BYTES_EXTRAS+((2*N_INPUTS+1)*N_CANAIS)-1) loop
1335     if (SER_CONT_BYTE > (1+(N_NIB+2)*n)) and
1336         (SER_CONT_BYTE < (7+(N_NIB+2)*n)) then
1337         SER_DIN <= BUFFER_SERIAL(n);
1338     elsif ((SER_CONT_BYTE) >
1339         (1+(N_NIB+2)*(BYTES_EXTRAS+((2*N_INPUTS+1)*N_CANAIS)-1))) then
1340         SER_DIN <= CHECKSUM;
1341     end if;
1342 end loop;
1343 end process;
1344
1345 process (SER_DIN)
1346 begin
1347 for n in 0 to (N_NIB-1) loop
1348     case SER_DIN(((n*4)+3) downto (n*4)) is
1349         when "0000" => SER_ASCII_CONV(n) <= "00110000"; -- ASCII "0"
1350         when "0001" => SER_ASCII_CONV(n) <= "00110001"; -- ASCII "1"
1351         when "0010" => SER_ASCII_CONV(n) <= "00110010"; -- ASCII "2"
1352         when "0011" => SER_ASCII_CONV(n) <= "00110011"; -- ASCII "3"
1353         when "0100" => SER_ASCII_CONV(n) <= "00110100"; -- ASCII "4"
1354         when "0101" => SER_ASCII_CONV(n) <= "00110101"; -- ASCII "5"
1355         when "0110" => SER_ASCII_CONV(n) <= "00110110"; -- ASCII "6"
1356         when "0111" => SER_ASCII_CONV(n) <= "00110111"; -- ASCII "7"
1357         when "1000" => SER_ASCII_CONV(n) <= "00111000"; -- ASCII "8"
1358         when "1001" => SER_ASCII_CONV(n) <= "00111001"; -- ASCII "9"
1359         when "1010" => SER_ASCII_CONV(n) <= "01000001"; -- ASCII "A"
1360         when "1011" => SER_ASCII_CONV(n) <= "01000010"; -- ASCII "B"
1361         when "1100" => SER_ASCII_CONV(n) <= "01000011"; -- ASCII "C"
1362         when "1101" => SER_ASCII_CONV(n) <= "01000100"; -- ASCII "D"
1363         when "1110" => SER_ASCII_CONV(n) <= "01000101"; -- ASCII "E"
1364         when "1111" => SER_ASCII_CONV(n) <= "01000110"; -- ASCII "F"
1365         when others => SER_ASCII_CONV(n) <= "00000000"; -- ASCII null
1366     end case;
1367 end loop;
1368 end process;
1369
1370 process (SER_CONT_BYTE,N_BYTES,SER_ASCII_CONV)
1371 begin
1372 if (SER_CONT_BYTE=1) then
1373     if (CONT_TAQUIS = 0) then
1374         SER_DBUFFER <= x"7B"; -- ASCII "{" (Inicio da TX)
1375     else
1376         SER_DBUFFER <= x"7C"; -- ASCII "|" (Inicio da TX)
1377     end if;
1378 end if;
1379 for a in 0 to (1+BYTES_EXTRAS+((2*N_INPUTS+1)*N_CANAIS)-1) loop -- CHECKSUM +
BYTES_EXTRAS
1380     if (SER_CONT_BYTE=(a*6)+2) then
1381         SER_DBUFFER <= conv_unsigned(32+a, 8); -- ASCII ID_BYTE
1382     end if;
1383     for b in 1 to N_NIB loop
1384         if (SER_CONT_BYTE=(a*6)+2+b) then
1385             SER_DBUFFER <= SER_ASCII_CONV(N_NIB-b); -- Data BYTES
1386         end if;
1387     end loop;
1388     if (SER_CONT_BYTE=(a*6)+7) then

```

```

1389     SER_DBUFFER <= x"7D";           -- ASCII "]" (Fim do BYTE)
1390     end if;
1391 end loop;
1392 if (SER_CONT_BYTE=(N_BYTES-2)) then
1393     SER_DBUFFER <= x"7E";           -- ASCII "~" (Fim da TX)
1394 end if;
1395 if (SER_CONT_BYTE=(N_BYTES-1)) then
1396     SER_DBUFFER <= x"0D";           -- ASCII \r (Ret. Carro)
1397 end if;
1398 if (SER_CONT_BYTE=N_BYTES) then
1399     SER_DBUFFER <= x"0A";           -- ASCII \n (Alim. Linha)
1400 end if;
1401 end process;
1402
1403 SER_PARIDADE <= (((((((SER_DBUFFER(7) xor SER_DBUFFER(6)) -- Construcao do BIT
1404     xor SER_DBUFFER(5)) xor SER_DBUFFER(4)) -- de paridade para
1405     xor SER_DBUFFER(3)) xor SER_DBUFFER(2)) -- a transmissao
1406     xor SER_DBUFFER(1)) xor SER_DBUFFER(0)); -- serial.
1407
1408 -- Montagem do pino de sada do transmissor serial:
1409 -- O cdigo abaixo atua como um "seletor lgico combinacional", escolhendo qual
1410 -- sinal eltrico vai estar presente na sada serial em funo da posio de
1411 -- transmissao do BIT atual (contador SER_CONT_BIT).
1412 SER_DOUT <= '1'           when SER_TXACTIVE = '0' else -- Linha Ociosa
1413     '0'           when SER_CONT_BIT = 1 else -- START BIT
1414     SER_DBUFFER(0) when SER_CONT_BIT = 2 else -- SERIAL BIT0 (LSB)
1415     SER_DBUFFER(1) when SER_CONT_BIT = 3 else -- SERIAL BIT1
1416     SER_DBUFFER(2) when SER_CONT_BIT = 4 else -- SERIAL BIT2
1417     SER_DBUFFER(3) when SER_CONT_BIT = 5 else -- SERIAL BIT3
1418     SER_DBUFFER(4) when SER_CONT_BIT = 6 else -- SERIAL BIT4
1419     SER_DBUFFER(5) when SER_CONT_BIT = 7 else -- SERIAL BIT5
1420     SER_DBUFFER(6) when SER_CONT_BIT = 8 else -- SERIAL BIT6
1421     SER_DBUFFER(7) when SER_CONT_BIT = 9 else -- SERIAL BIT7 (MSB)
1422     SER_PARIDADE when SER_CONT_BIT = 10 else -- BIT de Paridade
1423     '1'           when SER_CONT_BIT = 11 else -- STOP BIT 1
1424     '1'           when SER_CONT_BIT = 12 else -- STOP BIT 2
1425     '1';
1426
1427
1428 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1429 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1430 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1431 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1432 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1433 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1434 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1435
1436
1437 -----
1438 -----
1439 -----
1440 -----
1441 -----
1442 -----
1443 -----
1444 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1445 -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```



```
1503         if(GEN_FLAG1 xor GEN_FLAG2) = '1' then
1504             if STATE=S0 then
1505                 GEN1 <= '1';
1506                 GEN2 <= '1';
1507                 GEN3 <= '1';
1508             elsif STATE=S1 then
1509                 GEN1 <= '1';
1510                 GEN2 <= '1';
1511                 GEN3 <= '0';
1512             elsif STATE=S2 then
1513                 GEN1 <= '1';
1514                 GEN2 <= '0';
1515                 GEN3 <= '0';
1516             end if;
1517             GEN_FLAG1 <= not GEN_FLAG1;
1518         else
1519             GEN1 <= '0';
1520             GEN2 <= '0';
1521             GEN3 <= '0';
1522         end if;
1523     end if;
1524 end process;
1525
1526 GEN1a <= GEN1 when CONT_GEN_POSICAO < 6 else -- Linha Ociosa
1527     '0';
1528 GEN2a <= GEN2 when CONT_GEN_POSICAO < 6 else -- Linha Ociosa
1529     '0';
1530 GEN3a <= GEN3 when CONT_GEN_POSICAO < 6 else -- Linha Ociosa
1531     '0';
1532
1533 GEN1b <= GEN1 when CONT_GEN_POSICAO < 5 else -- Linha Ociosa
1534     '0';
1535 GEN2b <= GEN2 when CONT_GEN_POSICAO < 5 else -- Linha Ociosa
1536     '0';
1537 GEN3b <= GEN3 when CONT_GEN_POSICAO < 5 else -- Linha Ociosa
1538     '0';
1539
1540 GEN1c <= GEN1 when CONT_GEN_POSICAO < 4 else -- Linha Ociosa
1541     '0';
1542 GEN2c <= GEN2 when CONT_GEN_POSICAO < 4 else -- Linha Ociosa
1543     '0';
1544 GEN3c <= GEN3 when CONT_GEN_POSICAO < 4 else -- Linha Ociosa
1545     '0';
1546
1547 GEN1d <= GEN1 when CONT_GEN_POSICAO < 3 else -- Linha Ociosa
1548     '0';
1549 GEN2d <= GEN2 when CONT_GEN_POSICAO < 3 else -- Linha Ociosa
1550     '0';
1551 GEN3d <= GEN3 when CONT_GEN_POSICAO < 3 else -- Linha Ociosa
1552     '0';
1553
1554 GEN1e <= GEN1 when CONT_GEN_POSICAO < 2 else -- Linha Ociosa
1555     '0';
1556 GEN2e <= GEN2 when CONT_GEN_POSICAO < 2 else -- Linha Ociosa
1557     '0';
1558 GEN3e <= GEN3 when CONT_GEN_POSICAO < 2 else -- Linha Ociosa
1559     '0';
```



```

1  -- *****
2  -- Company:                CENTRO BRASILEIRO DE PESQUISAS FISICAS - CBPF BRASIL
3  -- Project:                ESPECTÔMETRO DE RAIOS CÔSMICOS DA CRIOSFERA
4  -- Model:                  V1495 - Multipurpose Programmable Trigger Unit
5  -- FPGA Proj. Name:       CREAT_FpgaFirmware
6  -- Device:                 ALTERA EP1C4F400C6
7  -- Author:                 Ulisses Carneiro (ulissestv@gmail.com)
8  -- Date:                   03-09-2014
9  -- -----
10 -- Based on demo version written by Luca Colombini and available to download by
11 -- CAEN SpA at url http://www.caen.it/csite/CaenProd.jsp?idmod=484&parent=11
12 -- -----
13 -- Module:                  CREAT_FpgaFirmware
14 -- Description:             Top level entity of the coincidence counter in CREAT
15 --                          (Cosmic Ray Experiment at Antartida) launched in
16 --                          South Pole (LATITUDE 84°S) on November 2014.
17 -- *****
18
19 -- #####
20 -- Revision History:
21 --   Date      Author          Rev.   Comments
22 -- 03 Set 14   Ulisses Carneiro 0001   Substituido o nome da top level entity e
23 --                                     criados comentários neste arquivo.
24 -- #####
25
26 LIBRARY ieee;
27 USE ieee.std_logic_1164.all;
28
29 ENTITY CREAT_FpgaFirmware IS
30     PORT(
31         A      : IN      std_logic_vector (31 DOWNTO 0);
32         B      : IN      std_logic_vector (31 DOWNTO 0);
33         GIN     : IN      std_logic_vector (1  DOWNTO 0);
34         IDD     : IN      std_logic_vector (2  DOWNTO 0);
35         IDE     : IN      std_logic_vector (2  DOWNTO 0);
36         IDF     : IN      std_logic_vector (2  DOWNTO 0);
37         LCLK    : IN      std_logic;
38         PULSE   : IN      std_logic_vector (3  DOWNTO 0);
39         WnR     : IN      std_logic;
40         nADS    : IN      std_logic;
41         nBLAST  : IN      std_logic;
42         nLRESET : IN      std_logic;
43         C       : OUT     std_logic_vector (31 DOWNTO 0);
44         DIRDDLY : OUT     std_logic;
45         GOUT    : OUT     std_logic_vector (1  DOWNTO 0);
46         SELD    : OUT     std_logic;
47         SELE    : OUT     std_logic;
48         SELF    : OUT     std_logic;
49         SELG    : OUT     std_logic;
50         START   : OUT     std_logic_vector (1  DOWNTO 0);
51         WR_DLY0 : OUT     std_logic;
52         WR_DLY1 : OUT     std_logic;
53         nINT    : OUT     std_logic;
54         nLEDG   : OUT     std_logic;
55         nLEDR   : OUT     std_logic;
56         nOED    : OUT     std_logic;
57         nOEDDLY0 : OUT     std_logic;

```

```
58     nOEDDLy1 : OUT    std_logic;
59     nOEE     : OUT    std_logic;
60     nOEF     : OUT    std_logic;
61     nOEG     : OUT    std_logic;
62     nREADY   : OUT    std_logic;
63     nSTART   : OUT    std_logic_vector (3 DOWNT0 2);
64     D        : INOUT  std_logic_vector (31 DOWNT0 0);
65     DDLY     : INOUT  std_logic_vector (7 DOWNT0 0);
66     E        : INOUT  std_logic_vector (31 DOWNT0 0);
67     F        : INOUT  std_logic_vector (31 DOWNT0 0);
68     FPGA     : INOUT  std_logic_vector (3 DOWNT0 0);
69     LAD      : INOUT  std_logic_vector (15 DOWNT0 0);
70     SPARE    : INOUT  std_logic_vector (11 DOWNT0 0)
71 );
72
73 -- Declarations
74
75 END CREAT_FpgaFirmware ;
76 LIBRARY ieee;
77 USE ieee.std_logic_1164.all;
78 USE ieee.std_logic_arith.all;
79 USE ieee.std_logic_misc.all;
80 USE ieee.std_logic_unsigned.all;
81
82
83 ARCHITECTURE struct OF CREAT_FpgaFirmware IS
84
85     -- Architecture declarations
86
87     -- Internal signal declarations
88     SIGNAL A_DIN      : std_logic_vector(31 DOWNT0 0);
89     SIGNAL B_DIN      : std_logic_vector(31 DOWNT0 0);
90     SIGNAL C_DOUT     : std_logic_vector(31 DOWNT0 0);
91     SIGNAL DDLY_IN    : std_logic_vector(7 DOWNT0 0);
92     SIGNAL DDLY_OE    : std_logic;
93     SIGNAL DDLY_OUT   : std_logic_vector(7 DOWNT0 0);
94     SIGNAL DLO0_GATE  : std_logic;
95     SIGNAL DLO0_OUT   : std_logic;
96     SIGNAL DLO1_GATE  : std_logic;
97     SIGNAL DLO1_OUT   : std_logic;
98     SIGNAL D_DATA_IN  : std_logic_vector(31 DOWNT0 0);
99     SIGNAL D_DATA_OE  : std_logic_vector(31 DOWNT0 0);
100    SIGNAL D_DATA_OUT  : std_logic_vector(31 DOWNT0 0);
101    SIGNAL D_DIN       : std_logic_vector(31 DOWNT0 0);
102    SIGNAL D_DIR       : std_logic;
103    SIGNAL D_DOUT      : std_logic_vector(31 DOWNT0 0);
104    SIGNAL D_IDCODE    : std_logic_vector( 2 DOWNT0 0);
105    SIGNAL D_LEV       : std_logic;
106    SIGNAL E_DATA_IN   : std_logic_vector(31 DOWNT0 0);
107    SIGNAL E_DATA_OE   : std_logic_vector(31 DOWNT0 0);
108    SIGNAL E_DATA_OUT  : std_logic_vector(31 DOWNT0 0);
109    SIGNAL E_DIN       : std_logic_vector(31 DOWNT0 0);
110    SIGNAL E_DIR       : std_logic;
111    SIGNAL E_DOUT      : std_logic_vector(31 DOWNT0 0);
112    SIGNAL E_IDCODE    : std_logic_vector(2 DOWNT0 0);
113    SIGNAL E_LEV       : std_logic;
114    SIGNAL FPGA_DIR    : std_logic_vector(3 DOWNT0 0);
```

```

115     SIGNAL FPGA_IN       : std_logic_vector(3 DOWNTO 0);
116     SIGNAL FPGA_OUT      : std_logic_vector(3 DOWNTO 0);
117     SIGNAL F_DATA_IN     : std_logic_vector(31 DOWNTO 0);
118     SIGNAL F_DATA_OE     : std_logic_vector(31 DOWNTO 0);
119     SIGNAL F_DATA_OUT    : std_logic_vector(31 DOWNTO 0);
120     SIGNAL F_DIN         : std_logic_vector(31 DOWNTO 0);
121     SIGNAL F_DIR         : std_logic;
122     SIGNAL F_DOUT        : std_logic_vector(31 DOWNTO 0);
123     SIGNAL F_IDCODE      : std_logic_vector( 2 DOWNTO 0);
124     SIGNAL F_LEV         : std_logic;
125     SIGNAL GREEN_PULSE   : std_logic;
126     SIGNAL G_DIN         : std_logic_vector(1 DOWNTO 0);
127     SIGNAL G_DIR         : std_logic;
128     SIGNAL G_DOUT        : std_logic_vector(1 DOWNTO 0);
129     SIGNAL G_LEV         : std_logic;
130     SIGNAL LAD_IN        : std_logic_vector(15 DOWNTO 0);
131     SIGNAL LAD_OE        : std_logic;
132     SIGNAL LAD_OUT       : std_logic_vector(15 DOWNTO 0);
133     SIGNAL PDL0_IN       : std_logic;
134     SIGNAL PDL0_OUT      : std_logic;
135     SIGNAL PDL1_IN       : std_logic;
136     SIGNAL PDL1_OUT      : std_logic;
137     SIGNAL PDL_DIR       : std_logic;
138     SIGNAL PDL_READ      : std_logic_vector( 7 DOWNTO 0);
139     SIGNAL PDL_SEL       : std_logic;
140     SIGNAL PDL_WR        : std_logic;
141     SIGNAL PDL_WRITE     : std_logic_vector( 7 DOWNTO 0);
142     SIGNAL RED_PULSE     : std_logic;
143     SIGNAL REG_ADDR      : std_logic_vector(15 DOWNTO 0);
144     SIGNAL REG_DIN       : std_logic_vector(15 DOWNTO 0);
145     SIGNAL REG_DOUT      : std_logic_vector(15 DOWNTO 0);
146     SIGNAL REG_RDEN      : std_logic;
147     SIGNAL REG_WREN      : std_logic;
148     SIGNAL SPARE_DIR     : std_logic_vector(11 DOWNTO 0);
149     SIGNAL SPARE_IN      : std_logic_vector(11 DOWNTO 0);
150     SIGNAL SPARE_OUT     : std_logic_vector(11 DOWNTO 0);
151     SIGNAL USR_ACCESS    : std_logic;
152
153
154     -- Component Declarations
155     COMPONENT CREAT_LogicAnalyzer
156     PORT (
157         nLBRES          : IN          std_logic ;           -- Async Reset (active
158         LCLK            : IN          std_logic ;           -- Local Bus Clock
159         --*****
160         -- REGISTER INTERFACE
161         --*****
162         REG_WREN        : IN          std_logic ;           -- Write pulse (active
163         REG_RDEN        : IN          std_logic ;           -- Read  pulse (active
164         REG_ADDR        : IN          std_logic_vector (15 DOWNTO 0); -- Register address
165         REG_DIN         : IN          std_logic_vector (15 DOWNTO 0); -- Data from CAEN
166         REG_DOUT        : OUT         std_logic_vector (15 DOWNTO 0); -- Data TO  CAEN
167         Local Bus

```



```

167     USR_ACCESS   : IN      std_logic ;           -- Current register
        access is
168     -- at user address space(Active high)
169     ___*****
170     -- V1495 Front Panel Ports (PORT A,B,C,G)
171     ___*****
172     A_DIN       : IN      std_logic_vector (31 DOWNT0 0); -- In A (32 x LVDS/ECL)
173     B_DIN       : IN      std_logic_vector (31 DOWNT0 0); -- In B (32 x LVDS/ECL)
174     C_DOUT      : OUT     std_logic_vector (31 DOWNT0 0); -- Out C (32 x LVDS)
175     G_LEV       : OUT     std_logic ;           -- Output Level Select
        (NIM/TTL)
176     G_DIR       : OUT     std_logic ;           -- Output Enable
177     G_DOUT      : OUT     std_logic_vector (1 DOWNT0 0); -- Out G - LEMO (2 x
        NIM/TTL)
178     G_DIN       : IN      std_logic_vector (1 DOWNT0 0); -- In G - LEMO (2 x
        NIM/TTL)
179     ___*****
180     -- A395x MEZZANINES INTERFACES (PORT D,E,F)
181     ___*****
182     -- Expansion Mezzanine Identifier:
183     -- x_IDCODE :
184     -- 000 : A395A (32 x IN LVDS/ECL)
185     -- 001 : A395B (32 x OUT LVDS)
186     -- 010 : A395C (32 x OUT ECL)
187     -- 011 : A395D (8 x IN/OUT NIM/TTL)
188
189     -- Expansion Mezzanine Port Signal Standard Select
190     -- x_LEV :
191     -- 0=>TTL,1=>NIM
192
193     -- Expansion Mezzanine Port Direction
194     -- x_DIR :
195     -- 0=>OUT,1=>IN
196
197     -- In/Out D (I/O Expansion)
198     D_IDCODE    : IN      std_logic_vector ( 2 DOWNT0 0); -- D slot mezzanine
        Identifier
199     D_LEV       : OUT     std_logic ;           -- D slot Port Signal
        Standard Select
200     D_DIR       : OUT     std_logic ;           -- D slot Port Direction
201     D_DIN       : IN      std_logic_vector (31 DOWNT0 0); -- D slot Data In Bus
202     D_DOUT      : OUT     std_logic_vector (31 DOWNT0 0); -- D slot Data Out Bus
203     -- In/Out E (I/O Expansion)
204     E_IDCODE    : IN      std_logic_vector ( 2 DOWNT0 0); -- E slot mezzanine
        Identifier
205     E_LEV       : OUT     std_logic ;           -- E slot Port Signal
        Standard Select
206     E_DIR       : OUT     std_logic ;           -- E slot Port Direction
207     E_DIN       : IN      std_logic_vector (31 DOWNT0 0); -- E slot Data In Bus
208     E_DOUT      : OUT     std_logic_vector (31 DOWNT0 0); -- E slot Data Out Bus
209     -- In/Out F (I/O Expansion)
210     F_IDCODE    : IN      std_logic_vector ( 2 DOWNT0 0); -- F slot mezzanine
        Identifier
211     F_LEV       : OUT     std_logic ;           -- F slot Port Signal
        Standard Select
212     F_DIR       : OUT     std_logic ;           -- F slot Port Direction
213     F_DIN       : IN      std_logic_vector (31 DOWNT0 0); -- F slot Data In Bus

```

```

214     F_DOUT      : OUT      std_logic_vector (31 DOWNTO 0); -- F slot Data Out Bus
215     --*****
216     -- DELAY LINES
217     --*****
218     -- PDL = Programmable Delay Lines (Step = 0.25ns / FSR = 64ns)
219     -- DLO = Delay Line Oscillator (Half Period ~ 10 ns)
220     -- 3D3428 PDL (PROGRAMMABLE DELAY LINE) CONFIGURATION
221     PDL_WR      : OUT      std_logic ; -- Write Enable
222     PDL_SEL     : OUT      std_logic ; -- PDL Selection
223     (0=>PDL0, 1=>PDL1)
224     PDL_READ    : IN       std_logic_vector ( 7 DOWNTO 0); -- Read Data
225     PDL_WRITE   : OUT      std_logic_vector ( 7 DOWNTO 0); -- Write Data
226     PDL_DIR     : OUT      std_logic ; -- Direction
227     (0=>Write, 1=>Read)
228     -- DELAY I/O
229     PDL0_OUT    : IN       std_logic ; -- Signal from PDL0
230     Output
231     PDL1_OUT    : IN       std_logic ; -- Signal from PDL1
232     Output
233     DLO0_OUT    : IN       std_logic ; -- Signal from DLO0
234     Output
235     DLO1_OUT    : IN       std_logic ; -- Signal from DLO1
236     Output
237     PDL0_IN     : OUT      std_logic ; -- Signal TO PDL0
238     Input
239     PDL1_IN     : OUT      std_logic ; -- Signal TO PDL1
240     Input
241     DLO0_GATE   : OUT      std_logic ; -- DLO0 Gate (active
242     high)
243     DLO1_GATE   : OUT      std_logic ; -- DLO1 Gate (active
244     high)
245     --*****
246     -- SPARE PORTS
247     --*****
248     SPARE_OUT   : OUT      std_logic_vector (11 DOWNTO 0); -- SPARE Data Out
249     SPARE_IN    : IN       std_logic_vector (11 DOWNTO 0); -- SPARE Data In
250     SPARE_DIR   : OUT      std_logic_vector (11 DOWNTO 0); -- SPARE Direction (0
251     => OUT, 1 => IN)
252     --*****
253     -- LED
254     --*****
255     RED_PULSE   : OUT      std_logic ; -- RED Led Pulse
256     (active high)
257     GREEN_PULSE : OUT      std_logic ; -- GREEN Led Pulse
258     (active high)
259 );
260 END COMPONENT;
261 COMPONENT spare_if
262 PORT (
263     SPARE_DIR : IN       std_logic_vector (11 DOWNTO 0);
264     SPARE_OUT : IN       std_logic_vector (11 DOWNTO 0);
265     SPARE_IN  : OUT      std_logic_vector (11 DOWNTO 0);
266     SPARE     : INOUT    std_logic_vector (11 DOWNTO 0)
267 );
268 END COMPONENT;
269 COMPONENT tristate_if
270 PORT (

```

```
258     DDLY_OE      : IN      std_logic;
259     DDLY_OUT     : IN      std_logic_vector ( 7 DOWNTO 0 );
260     D_DATA_OE    : IN      std_logic_vector (31 DOWNTO 0 );
261     D_DATA_OUT   : IN      std_logic_vector (31 DOWNTO 0 );
262     E_DATA_OE    : IN      std_logic_vector (31 DOWNTO 0 );
263     E_DATA_OUT   : IN      std_logic_vector (31 DOWNTO 0 );
264     FPGA_DIR     : IN      std_logic_vector ( 3 DOWNTO 0 );
265     FPGA_OUT     : IN      std_logic_vector ( 3 DOWNTO 0 );
266     F_DATA_OE    : IN      std_logic_vector (31 DOWNTO 0 );
267     F_DATA_OUT   : IN      std_logic_vector (31 DOWNTO 0 );
268     LAD_OE       : IN      std_logic;
269     LAD_OUT      : IN      std_logic_vector (15 DOWNTO 0 );
270     DDLY_IN      : OUT     std_logic_vector ( 7 DOWNTO 0 );
271     D_DATA_IN    : OUT     std_logic_vector (31 DOWNTO 0 );
272     E_DATA_IN    : OUT     std_logic_vector (31 DOWNTO 0 );
273     FPGA_IN      : OUT     std_logic_vector ( 3 DOWNTO 0 );
274     F_DATA_IN    : OUT     std_logic_vector (31 DOWNTO 0 );
275     LAD_IN       : OUT     std_logic_vector (15 DOWNTO 0 );
276     D            : INOUT   std_logic_vector (31 DOWNTO 0 );
277     DDLY         : INOUT   std_logic_vector ( 7 DOWNTO 0 );
278     E            : INOUT   std_logic_vector (31 DOWNTO 0 );
279     F            : INOUT   std_logic_vector (31 DOWNTO 0 );
280     FPGA         : INOUT   std_logic_vector ( 3 DOWNTO 0 );
281     LAD          : INOUT   std_logic_vector (15 DOWNTO 0 );
282 );
283 END COMPONENT;
284 COMPONENT v1495usr_hal
285 PORT (
286     A            : IN      std_logic_vector (31 DOWNTO 0 );
287     B            : IN      std_logic_vector (31 DOWNTO 0 );
288     C_DOUT       : IN      std_logic_vector (31 DOWNTO 0 );
289     DDLY_IN      : IN      std_logic_vector (7 DOWNTO 0 );
290     DLOO_GATE    : IN      std_logic ;
291     DLO1_GATE    : IN      std_logic ;
292     D_DATA_IN    : IN      std_logic_vector (31 DOWNTO 0 );
293     D_DIR        : IN      std_logic ;
294     D_DOUT       : IN      std_logic_vector (31 DOWNTO 0 );
295     D_LEV        : IN      std_logic ;
296     E_DATA_IN    : IN      std_logic_vector (31 DOWNTO 0 );
297     E_DIR        : IN      std_logic ;
298     E_DOUT       : IN      std_logic_vector (31 DOWNTO 0 );
299     E_LEV        : IN      std_logic ;
300     FPGA_IN      : IN      std_logic_vector ( 3 DOWNTO 0 );
301     F_DATA_IN    : IN      std_logic_vector (31 DOWNTO 0 );
302     F_DIR        : IN      std_logic ;
303     F_DOUT       : IN      std_logic_vector (31 DOWNTO 0 );
304     F_LEV        : IN      std_logic ;
305     GIN          : IN      std_logic_vector (1 DOWNTO 0 );
306     GREEN_PULSE  : IN      std_logic ;
307     G_DIR        : IN      std_logic ;
308     G_DOUT       : IN      std_logic_vector (1 DOWNTO 0 );
309     G_LEV        : IN      std_logic ;
310     IDD          : IN      std_logic_vector (2 DOWNTO 0 );
311     IDE          : IN      std_logic_vector (2 DOWNTO 0 );
312     IDF          : IN      std_logic_vector (2 DOWNTO 0 );
313     LAD_IN       : IN      std_logic_vector (15 DOWNTO 0 );
314     LCLK         : IN      std_logic ;
```

```
315     PDL0_IN      : IN      std_logic ;
316     PDL1_IN      : IN      std_logic ;
317     PDL_DIR      : IN      std_logic ;
318     PDL_SEL      : IN      std_logic ;
319     PDL_WR       : IN      std_logic ;
320     PDL_WRITE    : IN      std_logic_vector ( 7 DOWNT0 0 );
321     PULSE        : IN      std_logic_vector ( 3 DOWNT0 0 );
322     RED_PULSE    : IN      std_logic ;
323     REG_DOUT     : IN      std_logic_vector (15 DOWNT0 0 );
324     WnR          : IN      std_logic ;
325     nADS         : IN      std_logic ;
326     nBLAST       : IN      std_logic ;
327     nLRESET      : IN      std_logic ;
328     A_DIN        : OUT     std_logic_vector (31 DOWNT0 0 );
329     B_DIN        : OUT     std_logic_vector (31 DOWNT0 0 );
330     C            : OUT     std_logic_vector (31 DOWNT0 0 );
331     DDLY_OE     : OUT     std_logic ;
332     DDLY_OUT    : OUT     std_logic_vector ( 7 DOWNT0 0 );
333     DIRDDLY     : OUT     std_logic ;
334     DLO0_OUT    : OUT     std_logic ;
335     DLO1_OUT    : OUT     std_logic ;
336     D_DATA_OE   : OUT     std_logic_vector (31 DOWNT0 0 );
337     D_DATA_OUT  : OUT     std_logic_vector (31 DOWNT0 0 );
338     D_DIN       : OUT     std_logic_vector (31 DOWNT0 0 );
339     D_IDCODE    : OUT     std_logic_vector ( 2 DOWNT0 0 );
340     E_DATA_OE   : OUT     std_logic_vector (31 DOWNT0 0 );
341     E_DATA_OUT  : OUT     std_logic_vector (31 DOWNT0 0 );
342     E_DIN       : OUT     std_logic_vector (31 DOWNT0 0 );
343     E_IDCODE    : OUT     std_logic_vector ( 2 DOWNT0 0 );
344     FPGA_DIR    : OUT     std_logic_vector ( 3 DOWNT0 0 );
345     FPGA_OUT    : OUT     std_logic_vector ( 3 DOWNT0 0 );
346     F_DATA_OE   : OUT     std_logic_vector (31 DOWNT0 0 );
347     F_DATA_OUT  : OUT     std_logic_vector (31 DOWNT0 0 );
348     F_DIN       : OUT     std_logic_vector (31 DOWNT0 0 );
349     F_IDCODE    : OUT     std_logic_vector ( 2 DOWNT0 0 );
350     GOUT        : OUT     std_logic_vector (1 DOWNT0 0 );
351     G_DIN       : OUT     std_logic_vector (1 DOWNT0 0 );
352     LAD_OE     : OUT     std_logic ;
353     LAD_OUT    : OUT     std_logic_vector (15 DOWNT0 0 );
354     PDL0_OUT   : OUT     std_logic ;
355     PDL1_OUT   : OUT     std_logic ;
356     PDL_READ   : OUT     std_logic_vector ( 7 DOWNT0 0 );
357     REG_ADDR   : OUT     std_logic_vector (15 DOWNT0 0 );
358     REG_DIN    : OUT     std_logic_vector (15 DOWNT0 0 );
359     REG_RDEN   : OUT     std_logic ;
360     REG_WREN   : OUT     std_logic ;
361     SELD       : OUT     std_logic ;
362     SELE       : OUT     std_logic ;
363     SELF       : OUT     std_logic ;
364     SELG       : OUT     std_logic ;
365     START      : OUT     std_logic_vector (1 DOWNT0 0 );
366     USR_ACCESS : OUT     std_logic ;
367     WR_DLY0    : OUT     std_logic ;
368     WR_DLY1    : OUT     std_logic ;
369     nINT       : OUT     std_logic ;
370     nLEDG      : OUT     std_logic ;
371     nLEDR      : OUT     std_logic ;
```

```
372     noED          : OUT    std_logic ;
373     noEDDLY0      : OUT    std_logic ;
374     noEDDLY1      : OUT    std_logic ;
375     noEE          : OUT    std_logic ;
376     noEF          : OUT    std_logic ;
377     noEG          : OUT    std_logic ;
378     nREADY        : OUT    std_logic ;
379     nSTART        : OUT    std_logic_vector (3 DOWNTO 2)
380 );
381 END COMPONENT;
382
383
384 BEGIN
385
386     -- Instance port mappings.
387     I0 : CREAT_LogicAnalyzer
388         PORT MAP (
389             nLBRES      => nLRESET,
390             LCLK        => LCLK,
391             REG_WREN    => REG_WREN,
392             REG_RDEN    => REG_RDEN,
393             REG_ADDR    => REG_ADDR,
394             REG_DIN     => REG_DIN,
395             REG_DOUT    => REG_DOUT,
396             USR_ACCESS  => USR_ACCESS,
397             A_DIN       => A_DIN,
398             B_DIN       => B_DIN,
399             C_DOUT      => C_DOUT,
400             G_LEV      => G_LEV,
401             G_DIR       => G_DIR,
402             G_DOUT      => G_DOUT,
403             G_DIN       => G_DIN,
404             D_IDCODE    => D_IDCODE,
405             D_LEV      => D_LEV,
406             D_DIR       => D_DIR,
407             D_DIN       => D_DIN,
408             D_DOUT      => D_DOUT,
409             E_IDCODE    => E_IDCODE,
410             E_LEV      => E_LEV,
411             E_DIR       => E_DIR,
412             E_DIN       => E_DIN,
413             E_DOUT      => E_DOUT,
414             F_IDCODE    => F_IDCODE,
415             F_LEV      => F_LEV,
416             F_DIR       => F_DIR,
417             F_DIN       => F_DIN,
418             F_DOUT      => F_DOUT,
419             PDL_WR      => PDL_WR,
420             PDL_SEL     => PDL_SEL,
421             PDL_READ    => PDL_READ,
422             PDL_WRITE   => PDL_WRITE,
423             PDL_DIR     => PDL_DIR,
424             PDL0_OUT    => PDL0_OUT,
425             PDL1_OUT    => PDL1_OUT,
426             DLO0_OUT    => DLO0_OUT,
427             DLO1_OUT    => DLO1_OUT,
428             PDL0_IN     => PDL0_IN,
```

```
429         PDL1_IN      => PDL1_IN,
430         DLO0_GATE    => DLO0_GATE,
431         DLO1_GATE    => DLO1_GATE,
432         SPARE_OUT    => SPARE_OUT,
433         SPARE_IN     => SPARE_IN,
434         SPARE_DIR    => SPARE_DIR,
435         RED_PULSE    => RED_PULSE,
436         GREEN_PULSE => GREEN_PULSE
437     );
438 I2 : spare_if
439     PORT MAP (
440         SPARE_OUT => SPARE_OUT,
441         SPARE_IN  => SPARE_IN,
442         SPARE_DIR => SPARE_DIR,
443         SPARE     => SPARE
444     );
445 I3 : tristate_if
446     PORT MAP (
447         D_DATA_IN  => D_DATA_IN,
448         D_DATA_OE  => D_DATA_OE,
449         D_DATA_OUT => D_DATA_OUT,
450         E_DATA_IN  => E_DATA_IN,
451         E_DATA_OE  => E_DATA_OE,
452         E_DATA_OUT => E_DATA_OUT,
453         F_DATA_IN  => F_DATA_IN,
454         F_DATA_OE  => F_DATA_OE,
455         F_DATA_OUT => F_DATA_OUT,
456         LAD_IN     => LAD_IN,
457         LAD_OE     => LAD_OE,
458         LAD_OUT    => LAD_OUT,
459         DDLY_OE    => DDLY_OE,
460         DDLY_OUT   => DDLY_OUT,
461         DDLY_IN    => DDLY_IN,
462         FPGA_IN    => FPGA_IN,
463         FPGA_OUT   => FPGA_OUT,
464         FPGA_DIR   => FPGA_DIR,
465         D          => D,
466         DDLY       => DDLY,
467         E          => E,
468         F          => F,
469         LAD        => LAD,
470         FPGA       => FPGA
471     );
472 I1 : v1495usr_hal
473     PORT MAP (
474         A          => A,
475         B          => B,
476         C_DOUT    => C_DOUT,
477         DDLY_IN   => DDLY_IN,
478         DLO0_GATE => DLO0_GATE,
479         DLO1_GATE => DLO1_GATE,
480         D_DATA_IN => D_DATA_IN,
481         D_DIR     => D_DIR,
482         D_DOUT    => D_DOUT,
483         D_LEV     => D_LEV,
484         E_DATA_IN => E_DATA_IN,
485         E_DIR     => E_DIR,
```

```
486     E_DOUT      => E_DOUT ,
487     E_LEV      => E_LEV ,
488     FPGA_IN    => FPGA_IN ,
489     F_DATA_IN  => F_DATA_IN ,
490     F_DIR      => F_DIR ,
491     F_DOUT     => F_DOUT ,
492     F_LEV      => F_LEV ,
493     GIN        => GIN ,
494     GREEN_PULSE => GREEN_PULSE ,
495     G_DIR      => G_DIR ,
496     G_DOUT     => G_DOUT ,
497     G_LEV      => G_LEV ,
498     IDD        => IDD ,
499     IDE        => IDE ,
500     IDF        => IDF ,
501     LAD_IN     => LAD_IN ,
502     LCLK       => LCLK ,
503     PDL0_IN   => PDL0_IN ,
504     PDL1_IN   => PDL1_IN ,
505     PDL_DIR   => PDL_DIR ,
506     PDL_SEL   => PDL_SEL ,
507     PDL_WR    => PDL_WR ,
508     PDL_WRITE => PDL_WRITE ,
509     PULSE     => PULSE ,
510     RED_PULSE => RED_PULSE ,
511     REG_DOUT  => REG_DOUT ,
512     WnR       => WnR ,
513     nADS      => nADS ,
514     nBLAST    => nBLAST ,
515     nLRESET   => nLRESET ,
516     A_DIN     => A_DIN ,
517     B_DIN     => B_DIN ,
518     C         => C ,
519     DDLY_OE   => DDLY_OE ,
520     DDLY_OUT  => DDLY_OUT ,
521     DIRDDLY   => DIRDDLY ,
522     DLO0_OUT  => DLO0_OUT ,
523     DLO1_OUT  => DLO1_OUT ,
524     D_DATA_OE => D_DATA_OE ,
525     D_DATA_OUT => D_DATA_OUT ,
526     D_DIN     => D_DIN ,
527     D_IDCODE  => D_IDCODE ,
528     E_DATA_OE => E_DATA_OE ,
529     E_DATA_OUT => E_DATA_OUT ,
530     E_DIN     => E_DIN ,
531     E_IDCODE  => E_IDCODE ,
532     FPGA_DIR  => FPGA_DIR ,
533     FPGA_OUT  => FPGA_OUT ,
534     F_DATA_OE => F_DATA_OE ,
535     F_DATA_OUT => F_DATA_OUT ,
536     F_DIN     => F_DIN ,
537     F_IDCODE  => F_IDCODE ,
538     GOUT      => GOUT ,
539     G_DIN     => G_DIN ,
540     LAD_OE    => LAD_OE ,
541     LAD_OUT   => LAD_OUT ,
542     PDL0_OUT  => PDL0_OUT ,
```

```
543         PDL1_OUT      => PDL1_OUT ,
544         PDL_READ      => PDL_READ ,
545         REG_ADDR      => REG_ADDR ,
546         REG_DIN       => REG_DIN ,
547         REG_RDEN      => REG_RDEN ,
548         REG_WREN      => REG_WREN ,
549         SELD          => SELD ,
550         SELE          => SELE ,
551         SELF          => SELF ,
552         SELG          => SELG ,
553         START         => START ,
554         USR_ACCESS    => USR_ACCESS ,
555         WR_DLY0       => WR_DLY0 ,
556         WR_DLY1       => WR_DLY1 ,
557         nINT          => nINT ,
558         nLEDG         => nLEDG ,
559         nLEDR         => nLEDR ,
560         nOED          => nOED ,
561         nOEDDLY0     => nOEDDLY0 ,
562         nOEDDLY1     => nOEDDLY1 ,
563         nOEE          => nOEE ,
564         nOEF          => nOEF ,
565         nOEG          => nOEG ,
566         nREADY       => nREADY ,
567         nSTART       => nSTART
568     );
569
570 END struct;
571
```



```

1  -- *****
2  -- Company:          CENTRO BRASILEIRO DE PESQUISAS FISICAS - CBPF BRASIL
3  -- Project:         ESPECTMETRO DE RAIOS CSMICOS DA CRIOSFERA
4  -- Model:           V1495 - Multipurpose Programmable Trigger Unit
5  -- FPGA Proj. Name: CREAT_FpgaFirmware
6  -- Device:          ALTERA EP1C4F400C6
7  -- Author:          Ulisses Carneiro (ulissestv@gmail.com)
8  -- Date:            03-09-2014
9  -- -----
10 -- Based on demo version written by Luca Colombini and available to download by
11 -- CAEN SpA at url http://www.caen.it/csite/CaenProd.jsp?idmod=484&parent=11
12 -- -----
13 -- Module:           V1495USR_PKG
14 -- Description:      Package that implements global constants (ID Codes,
15 --                   Register Addresses).
16 -- *****
17
18 -- #####
19 -- Revision History:
20 --   Date      Author          Rev.   Comments
21 -- 03 Set 14   Ulisses Carneiro  0001   Substituidos nomes de registradores
22 --           de leitura do barramento VME e inseridos
23 --           comentarios em geral.
24 -- 18 Set 14   Ulisses Carneiro  0002   Atualizacao dos registradores para 6*in
25 -- 22 Set 14   Ulisses Carneiro  0003   Inclusao dos registradores de R/W, med.
26 --           de CrossTalk e canais auxiliares
27 -- #####
28 LIBRARY ieee;
29 USE ieee.std_logic_1164.all;
30 PACKAGE v1495pkg IS
31
32 -- Constants
33
34 -- DEMO Revision
35 constant REVISION : std_logic_vector(15 downto 0) := X"0200";
36
37 -- Expansion Mezzanine Type ID-Codes
38 constant A395A : std_logic_vector(2 downto 0) := "000"; -- 32CH IN LVDS/ECL
39 constant A395B : std_logic_vector(2 downto 0) := "001"; -- 32CH OUT LVDS INTERFACE
40 constant A395C : std_logic_vector(2 downto 0) := "010"; -- 32CH OUT ECL INTERFACE
41 constant A395D : std_logic_vector(2 downto 0) := "011"; -- 8CH I/O SELECT NIM/TTL
42
43 -- Register Address Map
44 constant A_SUM2a23 : std_logic_vector(15 downto 0) := X"1000";
45 constant A_SUM2a13 : std_logic_vector(15 downto 0) := X"1002";
46 constant A_SUM2a12 : std_logic_vector(15 downto 0) := X"1004";
47 constant A_SUM3a    : std_logic_vector(15 downto 0) := X"1006";
48 constant A_SUM2b23 : std_logic_vector(15 downto 0) := X"1008";
49 constant A_SUM2b13 : std_logic_vector(15 downto 0) := X"100A";
50 constant A_SUM2b12 : std_logic_vector(15 downto 0) := X"100C";
51 constant A_SUM3b    : std_logic_vector(15 downto 0) := X"100E";
52 constant A_SUM2c23 : std_logic_vector(15 downto 0) := X"1010";
53 constant A_SUM2c13 : std_logic_vector(15 downto 0) := X"1012";
54 constant A_SUM2c12 : std_logic_vector(15 downto 0) := X"1014";
55 constant A_SUM3c    : std_logic_vector(15 downto 0) := X"1016";

```

```
56  constant A_SUM2d23      : std_logic_vector(15 downto 0) := X"1018";
57  constant A_SUM2d13      : std_logic_vector(15 downto 0) := X"101A";
58  constant A_SUM2d12      : std_logic_vector(15 downto 0) := X"101C";
59  constant A_SUM3d        : std_logic_vector(15 downto 0) := X"101E";
60  constant A_SUM2e23      : std_logic_vector(15 downto 0) := X"1020";
61  constant A_SUM2e13      : std_logic_vector(15 downto 0) := X"1022";
62  constant A_SUM2e12      : std_logic_vector(15 downto 0) := X"1024";
63  constant A_SUM3e        : std_logic_vector(15 downto 0) := X"1026";
64  constant A_SUM2f23      : std_logic_vector(15 downto 0) := X"1028";
65  constant A_SUM2f13      : std_logic_vector(15 downto 0) := X"102A";
66  constant A_SUM2f12      : std_logic_vector(15 downto 0) := X"102C";
67  constant A_SUM3f        : std_logic_vector(15 downto 0) := X"102E";
68  constant A_SUM2g23      : std_logic_vector(15 downto 0) := X"1030";
69  constant A_SUM2g13      : std_logic_vector(15 downto 0) := X"1032";
70  constant A_SUM2g12      : std_logic_vector(15 downto 0) := X"1034";
71  constant A_SUM3g        : std_logic_vector(15 downto 0) := X"1036";
72  constant A_SUM2h23      : std_logic_vector(15 downto 0) := X"1038";
73  constant A_SUM2h13      : std_logic_vector(15 downto 0) := X"103A";
74  constant A_SUM2h12      : std_logic_vector(15 downto 0) := X"103C";
75  constant A_SUM3h        : std_logic_vector(15 downto 0) := X"103E";
76
77  constant A_LATCH2a23    : std_logic_vector(15 downto 0) := X"1100";
78  constant A_LATCH2a13    : std_logic_vector(15 downto 0) := X"1102";
79  constant A_LATCH2a12    : std_logic_vector(15 downto 0) := X"1104";
80  constant A_LATCH3a      : std_logic_vector(15 downto 0) := X"1106";
81  constant A_LATCH2b23    : std_logic_vector(15 downto 0) := X"1108";
82  constant A_LATCH2b13    : std_logic_vector(15 downto 0) := X"110A";
83  constant A_LATCH2b12    : std_logic_vector(15 downto 0) := X"110C";
84  constant A_LATCH3b      : std_logic_vector(15 downto 0) := X"110E";
85  constant A_LATCH2c23    : std_logic_vector(15 downto 0) := X"1110";
86  constant A_LATCH2c13    : std_logic_vector(15 downto 0) := X"1112";
87  constant A_LATCH2c12    : std_logic_vector(15 downto 0) := X"1114";
88  constant A_LATCH3c      : std_logic_vector(15 downto 0) := X"1116";
89  constant A_LATCH2d23    : std_logic_vector(15 downto 0) := X"1118";
90  constant A_LATCH2d13    : std_logic_vector(15 downto 0) := X"111A";
91  constant A_LATCH2d12    : std_logic_vector(15 downto 0) := X"111C";
92  constant A_LATCH3d      : std_logic_vector(15 downto 0) := X"111E";
93  constant A_LATCH2e23    : std_logic_vector(15 downto 0) := X"1120";
94  constant A_LATCH2e13    : std_logic_vector(15 downto 0) := X"1122";
95  constant A_LATCH2e12    : std_logic_vector(15 downto 0) := X"1124";
96  constant A_LATCH3e      : std_logic_vector(15 downto 0) := X"1126";
97  constant A_LATCH2f23    : std_logic_vector(15 downto 0) := X"1128";
98  constant A_LATCH2f13    : std_logic_vector(15 downto 0) := X"112A";
99  constant A_LATCH2f12    : std_logic_vector(15 downto 0) := X"112C";
100 constant A_LATCH3f      : std_logic_vector(15 downto 0) := X"112E";
101 constant A_LATCH2g23    : std_logic_vector(15 downto 0) := X"1130";
102 constant A_LATCH2g13    : std_logic_vector(15 downto 0) := X"1132";
103 constant A_LATCH2g12    : std_logic_vector(15 downto 0) := X"1134";
104 constant A_LATCH3g      : std_logic_vector(15 downto 0) := X"1136";
105 constant A_LATCH2h23    : std_logic_vector(15 downto 0) := X"1138";
106 constant A_LATCH2h13    : std_logic_vector(15 downto 0) := X"113A";
107 constant A_LATCH2h12    : std_logic_vector(15 downto 0) := X"113C";
108 constant A_LATCH3h      : std_logic_vector(15 downto 0) := X"113E";
109
110 constant A_SUM_IN1a     : std_logic_vector(15 downto 0) := X"1200";
111 constant A_SUM_IN2a     : std_logic_vector(15 downto 0) := X"1202";
112 constant A_SUM_IN3a     : std_logic_vector(15 downto 0) := X"1204";
```

```
113 constant A_SUM_IN1b : std_logic_vector(15 downto 0) := X"1208";
114 constant A_SUM_IN2b : std_logic_vector(15 downto 0) := X"120A";
115 constant A_SUM_IN3b : std_logic_vector(15 downto 0) := X"120C";
116 constant A_SUM_IN1c : std_logic_vector(15 downto 0) := X"1210";
117 constant A_SUM_IN2c : std_logic_vector(15 downto 0) := X"1212";
118 constant A_SUM_IN3c : std_logic_vector(15 downto 0) := X"1214";
119 constant A_SUM_IN1d : std_logic_vector(15 downto 0) := X"1218";
120 constant A_SUM_IN2d : std_logic_vector(15 downto 0) := X"121A";
121 constant A_SUM_IN3d : std_logic_vector(15 downto 0) := X"121C";
122 constant A_SUM_IN1e : std_logic_vector(15 downto 0) := X"1220";
123 constant A_SUM_IN2e : std_logic_vector(15 downto 0) := X"1222";
124 constant A_SUM_IN3e : std_logic_vector(15 downto 0) := X"1224";
125 constant A_SUM_IN1f : std_logic_vector(15 downto 0) := X"1228";
126 constant A_SUM_IN2f : std_logic_vector(15 downto 0) := X"122A";
127 constant A_SUM_IN3f : std_logic_vector(15 downto 0) := X"122C";
128 constant A_SUM_IN1g : std_logic_vector(15 downto 0) := X"1230";
129 constant A_SUM_IN2g : std_logic_vector(15 downto 0) := X"1232";
130 constant A_SUM_IN3g : std_logic_vector(15 downto 0) := X"1234";
131 constant A_SUM_IN1h : std_logic_vector(15 downto 0) := X"1238";
132 constant A_SUM_IN2h : std_logic_vector(15 downto 0) := X"123A";
133 constant A_SUM_IN3h : std_logic_vector(15 downto 0) := X"123C";
134
135 constant A_LATCH_IN1a : std_logic_vector(15 downto 0) := X"1300";
136 constant A_LATCH_IN2a : std_logic_vector(15 downto 0) := X"1302";
137 constant A_LATCH_IN3a : std_logic_vector(15 downto 0) := X"1304";
138 constant A_LATCH_IN1b : std_logic_vector(15 downto 0) := X"1308";
139 constant A_LATCH_IN2b : std_logic_vector(15 downto 0) := X"130A";
140 constant A_LATCH_IN3b : std_logic_vector(15 downto 0) := X"130C";
141 constant A_LATCH_IN1c : std_logic_vector(15 downto 0) := X"1310";
142 constant A_LATCH_IN2c : std_logic_vector(15 downto 0) := X"1312";
143 constant A_LATCH_IN3c : std_logic_vector(15 downto 0) := X"1314";
144 constant A_LATCH_IN1d : std_logic_vector(15 downto 0) := X"1318";
145 constant A_LATCH_IN2d : std_logic_vector(15 downto 0) := X"131A";
146 constant A_LATCH_IN3d : std_logic_vector(15 downto 0) := X"131C";
147 constant A_LATCH_IN1e : std_logic_vector(15 downto 0) := X"1320";
148 constant A_LATCH_IN2e : std_logic_vector(15 downto 0) := X"1322";
149 constant A_LATCH_IN3e : std_logic_vector(15 downto 0) := X"1324";
150 constant A_LATCH_IN1f : std_logic_vector(15 downto 0) := X"1328";
151 constant A_LATCH_IN2f : std_logic_vector(15 downto 0) := X"132A";
152 constant A_LATCH_IN3f : std_logic_vector(15 downto 0) := X"132C";
153 constant A_LATCH_IN1g : std_logic_vector(15 downto 0) := X"1330";
154 constant A_LATCH_IN2g : std_logic_vector(15 downto 0) := X"1332";
155 constant A_LATCH_IN3g : std_logic_vector(15 downto 0) := X"1334";
156 constant A_LATCH_IN1h : std_logic_vector(15 downto 0) := X"1338";
157 constant A_LATCH_IN2h : std_logic_vector(15 downto 0) := X"133A";
158 constant A_LATCH_IN3h : std_logic_vector(15 downto 0) := X"133C";
159
160 constant A_SUM_CROSS1a : std_logic_vector(15 downto 0) := X"1400";
161 constant A_SUM_CROSS2a : std_logic_vector(15 downto 0) := X"1402";
162 constant A_SUM_CROSS3a : std_logic_vector(15 downto 0) := X"1404";
163 constant A_SUM_CROSS1b : std_logic_vector(15 downto 0) := X"1408";
164 constant A_SUM_CROSS2b : std_logic_vector(15 downto 0) := X"140A";
165 constant A_SUM_CROSS3b : std_logic_vector(15 downto 0) := X"140C";
166 constant A_SUM_CROSS1c : std_logic_vector(15 downto 0) := X"1410";
167 constant A_SUM_CROSS2c : std_logic_vector(15 downto 0) := X"1412";
168 constant A_SUM_CROSS3c : std_logic_vector(15 downto 0) := X"1414";
169 constant A_SUM_CROSS1d : std_logic_vector(15 downto 0) := X"1418";
```

```
170  constant A_SUM_CROSS2d : std_logic_vector(15 downto 0) := X"141A";
171  constant A_SUM_CROSS3d : std_logic_vector(15 downto 0) := X"141C";
172  constant A_SUM_CROSS1e : std_logic_vector(15 downto 0) := X"1420";
173  constant A_SUM_CROSS2e : std_logic_vector(15 downto 0) := X"1422";
174  constant A_SUM_CROSS3e : std_logic_vector(15 downto 0) := X"1424";
175  constant A_SUM_CROSS1f : std_logic_vector(15 downto 0) := X"1428";
176  constant A_SUM_CROSS2f : std_logic_vector(15 downto 0) := X"142A";
177  constant A_SUM_CROSS3f : std_logic_vector(15 downto 0) := X"142C";
178  constant A_SUM_CROSS1g : std_logic_vector(15 downto 0) := X"1430";
179  constant A_SUM_CROSS2g : std_logic_vector(15 downto 0) := X"1432";
180  constant A_SUM_CROSS3g : std_logic_vector(15 downto 0) := X"1434";
181  constant A_SUM_CROSS1h : std_logic_vector(15 downto 0) := X"1438";
182  constant A_SUM_CROSS2h : std_logic_vector(15 downto 0) := X"143A";
183  constant A_SUM_CROSS3h : std_logic_vector(15 downto 0) := X"143C";
184
185  constant A_TAQUIS       : std_logic_vector(15 downto 0) := X"2000";
186  constant A_TGATE       : std_logic_vector(15 downto 0) := X"2002";
187  constant A_PATCH_NIM   : std_logic_vector(15 downto 0) := X"2004";
188  constant A_U_ID        : std_logic_vector(15 downto 0) := X"2006";
189  constant A_CTL_ANAL    : std_logic_vector(15 downto 0) := X"2008";
190
191  END v1495pkg;
192
```

```

1  -- *****
2  -- Company:          CENTRO BRASILEIRO DE PESQUISAS FISICAS - CBPF BRASIL
3  -- Project:         ESPECTÔMETRO DE RAIOS CÔSMICOS DA CRIOSFERA
4  -- Model:           V1495 - Multipurpose Programmable Trigger Unit
5  -- FPGA Proj. Name: CREAT_FpgaFirmware
6  -- Device:          ALTERA EP1C4F400C6
7  -- Author:          Ulisses Carneiro (ulissestv@gmail.com)
8  -- Date:            03-09-2014
9  -- -----
10 -- Based on demo version written by Luca Colombini and available to download by
11 -- CAEN SpA at url http://www.caen.it/csite/CaenProd.jsp?idmod=484&parent=11
12 -- -----
13 -- Module:           SPARE_IF
14 -- Description:      Interface entity to the Spare pins.
15 --                  It implements three-state buffering of spare pins.
16 -- *****
17
18 -- #####
19 -- Revision History:
20 -- #####
21
22 LIBRARY ieee;
23 USE ieee.std_logic_1164.all;
24 USE ieee.std_logic_arith.all;
25
26 ENTITY spare_if IS
27     PORT(
28         --*****
29         -- SPARE PORTS (USER SIDE)
30         --*****
31         SPARE_OUT    : IN    std_logic_vector(11 downto 0);
32         SPARE_IN     : OUT   std_logic_vector(11 downto 0);
33         SPARE_DIR    : IN    std_logic_vector(11 downto 0);
34         --*****
35         -- SPARE PORTS (I/O PAD)
36         --*****
37         SPARE        : INOUT std_logic_vector(11 downto 0)
38     );
39 END spare_if;
40 --
41 ARCHITECTURE rtl OF spare_if IS
42 BEGIN
43
44     process(SPARE_DIR, SPARE_OUT)
45     begin
46         for i in 0 to 11 loop
47             if SPARE_DIR(i) = '0' then
48                 SPARE(i)    <= SPARE_OUT(i);
49             else
50                 SPARE(i)    <= 'Z';
51             end if;
52         end loop;
53     end process;
54
55     SPARE_IN <= SPARE;
56
57 END rtl;

```

```

1  -- *****
2  -- Company:          CENTRO BRASILEIRO DE PESQUISAS FISICAS - CBPF BRASIL
3  -- Project:         ESPECTÔMETRO DE RAIOS CÔSMICOS DA CRIOSFERA
4  -- Model:           V1495 - Multipurpose Programmable Trigger Unit
5  -- FPGA Proj. Name: CREAT_FpgaFirmware
6  -- Device:          ALTERA EP1C4F400C6
7  -- Author:          Ulisses Carneiro (ulissestv@gmail.com)
8  -- Date:            03-09-2014
9  -- -----
10 -- Based on demo version written by Luca Colombini and available to download by
11 -- CAEN SpA at url http://www.caen.it/csite/CaenProd.jsp?idmod=484&parent=11
12 -- -----
13 -- Module:           TRISTATE_IF
14 -- Description:      Three-state buffers on bidirectional bus.
15 -- *****
16
17 -- #####
18 -- Revision History:
19 -- #####
20
21 LIBRARY ieee;
22 USE ieee.std_logic_1164.all;
23 USE ieee.std_logic_arith.all;
24
25 ENTITY tristate_if IS
26     PORT(
27         -- *****
28         -- DATA & CONTROLS (FROM V1495HAL)
29         -- *****
30         -- D MEZZANINE DATA BUS
31         D_DATA_IN   : OUT   std_logic_vector (31 DOWNTO 0);
32         D_DATA_OE   : IN    std_logic_vector (31 DOWNTO 0);
33         D_DATA_OUT  : IN    std_logic_vector (31 DOWNTO 0);
34         -- E MEZZANINE DATA BUS
35         E_DATA_IN   : OUT   std_logic_vector (31 DOWNTO 0);
36         E_DATA_OE   : IN    std_logic_vector (31 DOWNTO 0);
37         E_DATA_OUT  : IN    std_logic_vector (31 DOWNTO 0);
38         -- F MEZZANINE DATA BUS
39         F_DATA_IN   : OUT   std_logic_vector (31 DOWNTO 0);
40         F_DATA_OE   : IN    std_logic_vector (31 DOWNTO 0);
41         F_DATA_OUT  : IN    std_logic_vector (31 DOWNTO 0);
42         -- LOCAL BUS DATA BUS
43         LAD_IN      : OUT   std_logic_vector (15 DOWNTO 0);
44         LAD_OE      : IN    std_logic;
45         LAD_OUT     : IN    std_logic_vector (15 DOWNTO 0);
46         -- PDL DELAY LINES PARALLEL PROGRAMMING DATA BUS
47         DDLY_OE     : IN    std_logic;
48         DDLY_OUT    : IN    std_logic_vector ( 7 DOWNTO 0);
49         DDLY_IN     : OUT   std_logic_vector ( 7 DOWNTO 0);
50         -- FPGA LINK
51         FPGA_IN     : OUT   std_logic_vector ( 3 downto 0);
52         FPGA_OUT    : IN    std_logic_vector ( 3 downto 0);
53         FPGA_DIR    : IN    std_logic_vector ( 3 downto 0);
54         -- *****
55         -- BIDIR PORTS
56         -- *****
57         D           : INOUT std_logic_vector (31 DOWNTO 0);

```

```
58     DDLY      : INOUT  std_logic_vector ( 7 DOWNTO 0 );
59     E         : INOUT  std_logic_vector (31 DOWNTO 0 );
60     F         : INOUT  std_logic_vector (31 DOWNTO 0 );
61     LAD       : INOUT  std_logic_vector (15 DOWNTO 0 );
62     FPGA      : INOUT  std_logic_vector ( 3 downto 0 )
63 );
64 END tristate_if;
65
66 --
67 ARCHITECTURE rtl OF tristate_if IS
68 BEGIN
69
70     D_DATA_IN <= D;
71     E_DATA_IN <= E;
72     F_DATA_IN <= F;
73     LAD_IN    <= LAD;
74     DDLY_IN   <= DDLY;
75     FPGA_IN   <= FPGA;
76
77     process(D_DATA_OE, D_DATA_OUT)
78     begin
79         for i in 0 to 31 loop
80             if D_DATA_OE(i) = '1' then
81                 D(i)    <= D_DATA_OUT(i);
82             else
83                 D(i)    <= 'Z';
84             end if;
85         end loop;
86     end process;
87
88     process(E_DATA_OE, E_DATA_OUT)
89     begin
90         for i in 0 to 31 loop
91             if E_DATA_OE(i) = '1' then
92                 E(i)    <= E_DATA_OUT(i);
93             else
94                 E(i)    <= 'Z';
95             end if;
96         end loop;
97     end process;
98
99     process(F_DATA_OE, F_DATA_OUT)
100    begin
101        for i in 0 to 31 loop
102            if F_DATA_OE(i) = '1' then
103                F(i)    <= F_DATA_OUT(i);
104            else
105                F(i)    <= 'Z';
106            end if;
107        end loop;
108    end process;
109
110    process(FPGA_DIR, FPGA_OUT)
111    begin
112        for i in 0 to 3 loop
113            if FPGA_DIR(i) = '1' then
114                FPGA(i) <= FPGA_OUT(i);
```

```
115         else
116             FPGA(i)    <= 'Z';
117         end if;
118     end loop;
119 end process;
120
121
122 LAD  <= LAD_OUT  when LAD_OE   = '1' else (others => 'Z');
123 DDLY <= DDLY_OUT when DDLY_OE  = '1' else (others => 'Z');
124
125 END rtl;
126
127
```



```
1  # ----- #
2  #
3  # Copyright (C) 1991-2011 Altera Corporation
4  # Your use of Altera Corporation's design tools, logic functions
5  # and other software and tools, and its AMPP partner logic
6  # functions, and any output files from any of the foregoing
7  # (including device programming or simulation files), and any
8  # associated documentation or information are expressly subject
9  # to the terms and conditions of the Altera Program License
10 # Subscription Agreement, Altera MegaCore Function License
11 # Agreement, or other applicable license agreement, including,
12 # without limitation, that your use is for the sole purpose of
13 # programming logic devices manufactured by Altera and sold by
14 # Altera or its authorized distributors. Please refer to the
15 # applicable agreement for further details.
16 #
17 # ----- #
18 #
19 # Quartus II
20 # Version 11.0 Build 208 07/03/2011 Service Pack 1 SJ Web Edition
21 # Date created = 22:08:59 September 20, 2014
22 #
23 # ----- #
24
25 QUARTUS_VERSION = "11.0"
26 DATE = "22:08:59 September 20, 2014"
27
28 # Revisions
29
30 PROJECT_REVISION = "CREAT_FpgaFirmware"
31 PROJECT_REVISION = "CREAT_FpgaFirmware_r0022"
32 PROJECT_REVISION = "CREAT_FpgaFirmware_r0020"
33 PROJECT_REVISION = "CREAT_FpgaFirmware_r0021"
34
```

```
1  # Copyright (C) 1991-2007 Altera Corporation
2  # Your use of Altera Corporation's design tools, logic functions
3  # and other software and tools, and its AMPP partner logic
4  # functions, and any output files from any of the foregoing
5  # (including device programming or simulation files), and any
6  # associated documentation or information are expressly subject
7  # to the terms and conditions of the Altera Program License
8  # Subscription Agreement, Altera MegaCore Function License
9  # Agreement, or other applicable license agreement, including,
10 # without limitation, that your use is for the sole purpose of
11 # programming logic devices manufactured by Altera and sold by
12 # Altera or its authorized distributors. Please refer to the
13 # applicable agreement for further details.
14
15
16 # The default values for assignments are stored in the file
17 #   vl495usr_demo_assignment_defaults.qdf
18 # If this file doesn't exist, and for assignments not listed, see file
19 #   assignment_defaults.qdf
20
21 # Altera recommends that you do not modify this file. This
22 # file is updated automatically by the Quartus II software
23 # and any changes you make may be lost or overwritten.
24
25
26
27 # Project-Wide Assignments
28 # =====
29 set_global_assignment -name ORIGINAL_QUARTUS_VERSION 5.1
30 set_global_assignment -name PROJECT_CREATION_TIME_DATE "14:48:14  MARCH 02, 2006"
31 set_global_assignment -name LAST_QUARTUS_VERSION "11.0 SP1"
32 set_global_assignment -name VHDL_FILE ../SRC/CREAT_FpgaFirmware/vl495usr_pkg.vhd
33 set_global_assignment -name VHDL_FILE
34   ../SRC/CREAT_FpgaFirmware/CREAT_LogicAnalyzer.vhd
35 set_global_assignment -name VHDL_FILE ../SRC/CREAT_FpgaFirmware/spare_if_rtl.vhd
36 set_global_assignment -name VHDL_FILE ../SRC/CREAT_FpgaFirmware/tristate_if_rtl.vhd
37 set_global_assignment -name VHDL_FILE
38   ../SRC/CREAT_FpgaFirmware/CREAT_FpgaFirmware.vhd
39 set_global_assignment -name VQM_FILE ../SRC/CREAT_FpgaFirmware/vl495usr_hal.vqm
40
41 # Pin & Location Assignments
42 # =====
43 set_location_assignment PIN_V3 -to B[17]
44 set_location_assignment PIN_U2 -to C[5]
45 set_location_assignment PIN_V2 -to C[11]
46 set_location_assignment PIN_R4 -to C[17]
47 set_location_assignment PIN_U3 -to C[22]
48 set_location_assignment PIN_T3 -to E[21]
49 set_location_assignment PIN_T2 -to F[15]
50 set_location_assignment PIN_K5 -to LCLK
51 set_location_assignment PIN_F3 -to PULSE[0]
52 set_location_assignment PIN_T4 -to PULSE[2]
53 set_location_assignment PIN_G3 -to A[0]
54 set_location_assignment PIN_F2 -to A[1]
55 set_location_assignment PIN_F6 -to A[3]
56 set_location_assignment PIN_J5 -to A[8]
57 set_location_assignment PIN_H1 -to A[11]
```

```
56 set_location_assignment PIN_H2 -to A[12]
57 set_location_assignment PIN_H3 -to A[13]
58 set_location_assignment PIN_G5 -to A[14]
59 set_location_assignment PIN_H6 -to A[18]
60 set_location_assignment PIN_F5 -to A[26]
61 set_location_assignment PIN_J3 -to A[29]
62 set_location_assignment PIN_P5 -to B[0]
63 set_location_assignment PIN_P2 -to B[1]
64 set_location_assignment PIN_M4 -to B[6]
65 set_location_assignment PIN_M2 -to B[7]
66 set_location_assignment PIN_H5 -to B[9]
67 set_location_assignment PIN_G6 -to B[10]
68 set_location_assignment PIN_P3 -to B[16]
69 set_location_assignment PIN_P6 -to B[20]
70 set_location_assignment PIN_M7 -to B[21]
71 set_location_assignment PIN_N3 -to B[22]
72 set_location_assignment PIN_M1 -to B[23]
73 set_location_assignment PIN_H4 -to B[24]
74 set_location_assignment PIN_G4 -to B[25]
75 set_location_assignment PIN_F4 -to B[26]
76 set_location_assignment PIN_G2 -to B[27]
77 set_location_assignment PIN_M3 -to C[4]
78 set_location_assignment PIN_R6 -to C[6]
79 set_location_assignment PIN_R2 -to C[7]
80 set_location_assignment PIN_R1 -to C[10]
81 set_location_assignment PIN_R5 -to C[14]
82 set_location_assignment PIN_N4 -to C[18]
83 set_location_assignment PIN_R3 -to C[23]
84 set_location_assignment PIN_P4 -to C[26]
85 set_location_assignment PIN_M5 -to C[31]
86 set_location_assignment PIN_P1 -to E[7]
87 set_location_assignment PIN_J4 -to E[14]
88 set_location_assignment PIN_P7 -to E[17]
89 set_location_assignment PIN_N2 -to E[22]
90 set_location_assignment PIN_N1 -to E[23]
91 set_location_assignment PIN_F1 -to E[24]
92 set_location_assignment PIN_G1 -to E[25]
93 set_location_assignment PIN_E2 -to E[26]
94 set_location_assignment PIN_M6 -to F[2]
95 set_location_assignment PIN_N7 -to F[10]
96 set_location_assignment PIN_N6 -to F[11]
97 set_location_assignment PIN_E3 -to FPGA[1]
98 set_location_assignment PIN_J6 -to IDD[1]
99 set_location_assignment PIN_J1 -to LAD[3]
100 set_location_assignment PIN_G7 -to LAD[9]
101 set_location_assignment PIN_J2 -to nOED
102 set_location_assignment PIN_C2 -to A[2]
103 set_location_assignment PIN_C4 -to A[5]
104 set_location_assignment PIN_D1 -to A[9]
105 set_location_assignment PIN_C5 -to A[15]
106 set_location_assignment PIN_D5 -to A[16]
107 set_location_assignment PIN_F7 -to A[17]
108 set_location_assignment PIN_D6 -to A[20]
109 set_location_assignment PIN_E5 -to A[22]
110 set_location_assignment PIN_B6 -to A[28]
111 set_location_assignment PIN_B4 -to B[11]
112 set_location_assignment PIN_E7 -to B[29]
```

```
113 set_location_assignment PIN_E4 -to FPGA[0]
114 set_location_assignment PIN_B3 -to FPGA[2]
115 set_location_assignment PIN_B5 -to LAD[4]
116 set_location_assignment PIN_B7 -to LAD[10]
117 set_location_assignment PIN_A7 -to nINT
118 set_location_assignment PIN_A6 -to nREADY
119 set_location_assignment PIN_A4 -to WnR
120 set_location_assignment PIN_D8 -to A[4]
121 set_location_assignment PIN_A8 -to A[6]
122 set_location_assignment PIN_C8 -to A[10]
123 set_location_assignment PIN_A9 -to A[24]
124 set_location_assignment PIN_F8 -to B[28]
125 set_location_assignment PIN_D7 -to B[30]
126 set_location_assignment PIN_C7 -to B[31]
127 set_location_assignment PIN_B8 -to FPGA[3]
128 set_location_assignment PIN_E8 -to LAD[11]
129 set_location_assignment PIN_E11 -to LAD[12]
130 set_location_assignment PIN_C6 -to PULSE[1]
131 set_location_assignment PIN_C15 -to PULSE[3]
132 set_location_assignment PIN_B11 -to A[7]
133 set_location_assignment PIN_C9 -to A[19]
134 set_location_assignment PIN_D9 -to A[21]
135 set_location_assignment PIN_A12 -to A[23]
136 set_location_assignment PIN_C14 -to A[25]
137 set_location_assignment PIN_B12 -to A[27]
138 set_location_assignment PIN_C12 -to A[30]
139 set_location_assignment PIN_A13 -to A[31]
140 set_location_assignment PIN_C11 -to B[12]
141 set_location_assignment PIN_D11 -to B[13]
142 set_location_assignment PIN_C10 -to B[14]
143 set_location_assignment PIN_D10 -to B[15]
144 set_location_assignment PIN_E14 -to D[0]
145 set_location_assignment PIN_E13 -to D[1]
146 set_location_assignment PIN_D12 -to D[2]
147 set_location_assignment PIN_F12 -to D[3]
148 set_location_assignment PIN_E9 -to D[4]
149 set_location_assignment PIN_C13 -to D[5]
150 set_location_assignment PIN_F14 -to D[6]
151 set_location_assignment PIN_D16 -to D[10]
152 set_location_assignment PIN_E10 -to D[11]
153 set_location_assignment PIN_E15 -to D[14]
154 set_location_assignment PIN_D14 -to D[15]
155 set_location_assignment PIN_A17 -to E[27]
156 set_location_assignment PIN_D13 -to E[29]
157 set_location_assignment PIN_C16 -to E[30]
158 set_location_assignment PIN_E12 -to E[31]
159 set_location_assignment PIN_E16 -to IDD[0]
160 set_location_assignment PIN_C17 -to IDD[2]
161 set_location_assignment PIN_B17 -to LAD[0]
162 set_location_assignment PIN_B15 -to LAD[1]
163 set_location_assignment PIN_B18 -to LAD[2]
164 set_location_assignment PIN_A11 -to LAD[5]
165 set_location_assignment PIN_B14 -to LAD[6]
166 set_location_assignment PIN_B10 -to LAD[7]
167 set_location_assignment PIN_B13 -to LAD[8]
168 set_location_assignment PIN_B9 -to LAD[14]
169 set_location_assignment PIN_B16 -to LAD[15]
```

```
170 set_location_assignment PIN_A10 -to nADS
171 set_location_assignment PIN_A15 -to nBLAST
172 set_location_assignment PIN_A14 -to nLRESET
173 set_location_assignment PIN_F16 -to D[7]
174 set_location_assignment PIN_F19 -to D[8]
175 set_location_assignment PIN_H15 -to D[9]
176 set_location_assignment PIN_C19 -to D[12]
177 set_location_assignment PIN_C18 -to D[13]
178 set_location_assignment PIN_G14 -to D[16]
179 set_location_assignment PIN_G17 -to D[17]
180 set_location_assignment PIN_G16 -to D[18]
181 set_location_assignment PIN_F17 -to D[19]
182 set_location_assignment PIN_D19 -to D[20]
183 set_location_assignment PIN_E17 -to D[21]
184 set_location_assignment PIN_D20 -to D[22]
185 set_location_assignment PIN_F20 -to D[23]
186 set_location_assignment PIN_E18 -to D[24]
187 set_location_assignment PIN_E19 -to D[25]
188 set_location_assignment PIN_D18 -to DDLY[4]
189 set_location_assignment PIN_G15 -to E[8]
190 set_location_assignment PIN_G19 -to E[11]
191 set_location_assignment PIN_G20 -to E[13]
192 set_location_assignment PIN_L14 -to GIN[0]
193 set_location_assignment PIN_F18 -to GIN[1]
194 set_location_assignment PIN_P20 -to GOUT[0]
195 set_location_assignment PIN_H16 -to IDE[0]
196 set_location_assignment PIN_J13 -to IDE[1]
197 set_location_assignment PIN_M14 -to IDE[2]
198 set_location_assignment PIN_K19 -to nLEDG
199 set_location_assignment PIN_D17 -to nLEDR
200 set_location_assignment PIN_N14 -to nOEE
201 set_location_assignment PIN_F15 -to SELD
202 set_location_assignment PIN_T18 -to SELE
203 set_location_assignment PIN_J16 -to D[28]
204 set_location_assignment PIN_H19 -to E[9]
205 set_location_assignment PIN_H18 -to E[10]
206 set_location_assignment PIN_H17 -to E[12]
207 set_location_assignment PIN_G18 -to E[15]
208 set_location_assignment PIN_J15 -to E[28]
209 set_location_assignment PIN_J14 -to B[8]
210 set_location_assignment PIN_T16 -to C[9]
211 set_location_assignment PIN_J18 -to D[26]
212 set_location_assignment PIN_K15 -to D[27]
213 set_location_assignment PIN_M15 -to D[29]
214 set_location_assignment PIN_K16 -to D[30]
215 set_location_assignment PIN_J17 -to D[31]
216 set_location_assignment PIN_U15 -to DDLY[0]
217 set_location_assignment PIN_M16 -to DDLY[1]
218 set_location_assignment PIN_N17 -to DDLY[2]
219 set_location_assignment PIN_W16 -to DDLY[3]
220 set_location_assignment PIN_N16 -to DDLY[5]
221 set_location_assignment PIN_U16 -to DDLY[6]
222 set_location_assignment PIN_V15 -to DDLY[7]
223 set_location_assignment PIN_W14 -to DIRDDLY
224 set_location_assignment PIN_T17 -to F[4]
225 set_location_assignment PIN_R16 -to F[5]
226 set_location_assignment PIN_P14 -to F[7]
```

```
227 set_location_assignment PIN_P18 -to F[8]
228 set_location_assignment PIN_R17 -to F[9]
229 set_location_assignment PIN_V18 -to F[12]
230 set_location_assignment PIN_U18 -to F[13]
231 set_location_assignment PIN_N15 -to F[14]
232 set_location_assignment PIN_U14 -to F[16]
233 set_location_assignment PIN_T15 -to F[17]
234 set_location_assignment PIN_V14 -to F[18]
235 set_location_assignment PIN_R14 -to F[19]
236 set_location_assignment PIN_T14 -to F[24]
237 set_location_assignment PIN_Y14 -to F[25]
238 set_location_assignment PIN_L13 -to GOUT[1]
239 set_location_assignment PIN_R18 -to IDF[1]
240 set_location_assignment PIN_H14 -to LAD[13]
241 set_location_assignment PIN_P17 -to nOEDDLY0
242 set_location_assignment PIN_R15 -to nOEDDLY1
243 set_location_assignment PIN_P15 -to nOEG
244 set_location_assignment PIN_R20 -to nSTART[2]
245 set_location_assignment PIN_R19 -to nSTART[3]
246 set_location_assignment PIN_P16 -to SELG
247 set_location_assignment PIN_V17 -to START[0]
248 set_location_assignment PIN_V16 -to START[1]
249 set_location_assignment PIN_Y17 -to WR_DLY0
250 set_location_assignment PIN_W17 -to WR_DLY1
251 set_location_assignment PIN_W9 -to B[19]
252 set_location_assignment PIN_U13 -to C[0]
253 set_location_assignment PIN_Y9 -to C[21]
254 set_location_assignment PIN_W10 -to C[25]
255 set_location_assignment PIN_Y10 -to C[30]
256 set_location_assignment PIN_V10 -to E[0]
257 set_location_assignment PIN_Y15 -to E[1]
258 set_location_assignment PIN_T6 -to E[2]
259 set_location_assignment PIN_W15 -to E[3]
260 set_location_assignment PIN_R11 -to E[4]
261 set_location_assignment PIN_U12 -to E[5]
262 set_location_assignment PIN_R13 -to E[6]
263 set_location_assignment PIN_U10 -to E[18]
264 set_location_assignment PIN_V13 -to E[20]
265 set_location_assignment PIN_T13 -to F[0]
266 set_location_assignment PIN_Y12 -to F[1]
267 set_location_assignment PIN_W13 -to F[3]
268 set_location_assignment PIN_T10 -to F[6]
269 set_location_assignment PIN_T11 -to F[20]
270 set_location_assignment PIN_U11 -to F[21]
271 set_location_assignment PIN_V11 -to F[22]
272 set_location_assignment PIN_R9 -to F[23]
273 set_location_assignment PIN_W11 -to F[26]
274 set_location_assignment PIN_Y11 -to F[27]
275 set_location_assignment PIN_T9 -to F[28]
276 set_location_assignment PIN_T12 -to F[29]
277 set_location_assignment PIN_W12 -to IDF[0]
278 set_location_assignment PIN_V12 -to IDF[2]
279 set_location_assignment PIN_Y13 -to nOEF
280 set_location_assignment PIN_W6 -to B[2]
281 set_location_assignment PIN_Y8 -to B[3]
282 set_location_assignment PIN_U8 -to B[4]
283 set_location_assignment PIN_U6 -to B[5]
```

```
284 set_location_assignment PIN_W8 -to B[18]
285 set_location_assignment PIN_Y4 -to C[1]
286 set_location_assignment PIN_V8 -to C[2]
287 set_location_assignment PIN_W7 -to C[3]
288 set_location_assignment PIN_R7 -to C[8]
289 set_location_assignment PIN_T5 -to C[12]
290 set_location_assignment PIN_U5 -to C[13]
291 set_location_assignment PIN_Y7 -to C[15]
292 set_location_assignment PIN_V5 -to C[16]
293 set_location_assignment PIN_Y6 -to C[19]
294 set_location_assignment PIN_V6 -to C[20]
295 set_location_assignment PIN_U9 -to C[24]
296 set_location_assignment PIN_W3 -to C[27]
297 set_location_assignment PIN_U7 -to C[28]
298 set_location_assignment PIN_V7 -to C[29]
299 set_location_assignment PIN_V4 -to E[16]
300 set_location_assignment PIN_W4 -to E[19]
301 set_location_assignment PIN_V9 -to F[30]
302 set_location_assignment PIN_T8 -to F[31]
303 set_location_assignment PIN_W5 -to SELF
304 set_location_assignment PIN_H7 -to SPARE[11]
305 set_location_assignment PIN_J7 -to SPARE[10]
306 set_location_assignment PIN_N5 -to SPARE[9]
307 set_location_assignment PIN_U4 -to SPARE[8]
308 set_location_assignment PIN_D3 -to SPARE[6]
309 set_location_assignment PIN_D2 -to SPARE[7]
310 set_location_assignment PIN_U19 -to SPARE[0]
311 set_location_assignment PIN_V19 -to SPARE[3]
312 set_location_assignment PIN_U20 -to SPARE[2]
313 set_location_assignment PIN_W18 -to SPARE[1]
314 set_location_assignment PIN_N20 -to SPARE[5]
315 set_location_assignment PIN_P19 -to SPARE[4]
316
317 # Analysis & Synthesis Assignments
318 # =====
319 set_global_assignment -name FAMILY Cyclone
320 set_global_assignment -name DEVICE_FILTER_PACKAGE FBGA
321 set_global_assignment -name DEVICE_FILTER_PIN_COUNT 400
322 set_global_assignment -name DEVICE_FILTER_SPEED_GRADE FASTEST
323 set_global_assignment -name TOP_LEVEL_ENTITY CREAT_FpgaFirmware
324
325 # Fitter Assignments
326 # =====
327 set_global_assignment -name RESERVE_ALL_UNUSED_PINS "AS INPUT TRI-STATED"
328 set_global_assignment -name CYCLONE_CONFIGURATION_SCHEME "PASSIVE SERIAL"
329 set_global_assignment -name DEVICE EP1C20F400C6
330 set_global_assignment -name STRATIX_DEVICE_IO_STANDARD LVTTL
331 set_instance_assignment -name IO_STANDARD LVTTL -to A[9]
332 set_instance_assignment -name IO_STANDARD LVTTL -to SPARE[11]
333 set_instance_assignment -name IO_STANDARD LVTTL -to SPARE[10]
334 set_instance_assignment -name IO_STANDARD LVTTL -to SPARE[9]
335 set_instance_assignment -name IO_STANDARD LVTTL -to SPARE[8]
336 set_instance_assignment -name IO_STANDARD LVTTL -to SPARE[6]
337 set_instance_assignment -name IO_STANDARD LVTTL -to SPARE[7]
338 set_instance_assignment -name IO_STANDARD LVTTL -to SPARE[0]
339 set_instance_assignment -name IO_STANDARD LVTTL -to SPARE[3]
340 set_instance_assignment -name IO_STANDARD LVTTL -to SPARE[2]
```

```

341 set_instance_assignment -name IO_STANDARD LVTTTL -to SPARE[1]
342 set_instance_assignment -name IO_STANDARD LVTTTL -to SPARE[5]
343 set_instance_assignment -name IO_STANDARD LVTTTL -to SPARE[4]
344 set_instance_assignment -name IO_STANDARD LVTTTL -to B[17]
345 set_global_assignment -name ENABLE_INIT_DONE_OUTPUT ON
346
347 # EDA Netlist Writer Assignments
348 # =====
349 set_global_assignment -name EDA_SIMULATION_TOOL "Active-HDL (Verilog)"
350
351 # Assembler Assignments
352 # =====
353 set_global_assignment -name GENERATE_RBF_FILE ON
354 set_global_assignment -name RESERVE_ALL_UNUSED_PINS_NO_OUTPUT_GND "AS INPUT
TRI-STATED"
355
356 # start EDA_TOOL_SETTINGS(eda_simulation)
357 # -----
358
359     # EDA Netlist Writer Assignments
360     # =====
361 set_global_assignment -name EDA_INCLUDE_VHDL_CONFIGURATION_DECLARATION OFF
-section_id eda_simulation
362 set_global_assignment -name EDA_MAP_ILLEGAL_CHARACTERS OFF -section_id
eda_simulation
363 set_global_assignment -name EDA_OUTPUT_DATA_FORMAT "VERILOG HDL" -section_id
eda_simulation
364
365 # end EDA_TOOL_SETTINGS(eda_simulation)
366 # -----
367
368 # start CLOCK(LOCAL_CLOCK)
369 # -----
370
371     # Classic Timing Assignments
372     # =====
373 set_global_assignment -name FMAX_REQUIREMENT "50 MHz" -section_id LOCAL_CLOCK
374 set_global_assignment -name INCLUDE_EXTERNAL_PIN_DELAYS_IN_FMAX_CALCULATIONS OFF
-section_id LOCAL_CLOCK
375 set_global_assignment -name DUTY_CYCLE 50 -section_id LOCAL_CLOCK
376
377 # end CLOCK(LOCAL_CLOCK)
378 # -----
379
380 # start CLOCK(PULSE)
381 # -----
382
383     # Classic Timing Assignments
384     # =====
385 set_global_assignment -name FMAX_REQUIREMENT "200 MHz" -section_id PULSE
386 set_global_assignment -name INCLUDE_EXTERNAL_PIN_DELAYS_IN_FMAX_CALCULATIONS OFF
-section_id PULSE
387 set_global_assignment -name DUTY_CYCLE 50 -section_id PULSE
388
389 # end CLOCK(PULSE)
390 # -----
391

```



```
392 # -----
393 # start ENTITY(v1495usr_demo)
394
395 # Pin & Location Assignments
396 # =====
397 set_instance_assignment -name FAST_OUTPUT_REGISTER ON -to LAD
398 set_instance_assignment -name FAST_INPUT_REGISTER ON -to LAD
399
400 # Classic Timing Assignments
401 # =====
402 set_instance_assignment -name CLOCK_SETTINGS PULSE -to PULSE[0]
403 set_instance_assignment -name CLOCK_SETTINGS LOCAL_CLOCK -to LCLK
404 set_instance_assignment -name CUT ON -from "coin_unit:I0|MODE" -to *
405 set_instance_assignment -name CUT ON -from "coin_unit:I0|A_EN_MASK" -to *
406 set_instance_assignment -name CUT ON -from "coin_unit:I0|B_EN_MASK" -to *
407 set_instance_assignment -name CUT ON -from "coin_unit:I0|C_EN_MASK" -to *
408 set_instance_assignment -name TCO_REQUIREMENT "10 ns" -from LCLK -to LAD
409 set_instance_assignment -name REPORT_DELAY ON -from nBLAST -to LCLK
410
411 # Fitter Assignments
412 # =====
413 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to C
414 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to DIRDDLY
415 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to GOUT
416 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to GOUT[0]
417 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to GOUT[1]
418 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nINT
419 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nLEDG
420 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nLEDR
421 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nOED
422 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nOEDDLY0
423 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nOEDDLY1
424 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nOEE
425 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nOEF
426 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nOEG
427 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nREADY
428 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nSTART
429 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nSTART[2]
430 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to nSTART[3]
431 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to SELD
432 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to SELE
433 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to SELF
434 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to SELG
435 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to SPARE
436 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to START
437 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to START[0]
438 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to START[1]
439 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to WR_DLY0
440 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to WR_DLY1
441 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to D
442 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to E
443 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to F
444 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to FPGA
445 set_instance_assignment -name CURRENT_STRENGTH_NEW "STRENGTH 4MA" -to LAD
446 set_instance_assignment -name WEAK_PULL_UP_RESISTOR ON -to IDD
447 set_instance_assignment -name WEAK_PULL_UP_RESISTOR ON -to IDE
448 set_instance_assignment -name WEAK_PULL_UP_RESISTOR ON -to IDF
```

```
449 set_instance_assignment -name WEAK_PULL_UP_RESISTOR ON -to D
450 set_instance_assignment -name WEAK_PULL_UP_RESISTOR ON -to E
451 set_instance_assignment -name WEAK_PULL_UP_RESISTOR ON -to F
452
453 # end ENTITY(v1495usr_demo)
454 # -----
455 set_global_assignment -name LL_ROOT_REGION ON -section_id "Root Region"
456 set_global_assignment -name LL_MEMBER_STATE LOCKED -section_id "Root Region"
457 set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id Top
458 set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL
PLACEMENT_AND_ROUTING -section_id Top
459 set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top
460 set_global_assignment -name EDA_TIME_SCALE "1 ps" -section_id eda_simulation
461
462 set_global_assignment -name VHDL_FILE
../SRC/CREAT_FpgaFirmware/v1495usr_pkg_r0001.vhd
463 set_global_assignment -name VHDL_FILE
../SRC/CREAT_FpgaFirmware/v1495usr_pkg_r0002.vhd
464 set_global_assignment -name POWER_USE_PVA OFF
465 set_instance_assignment -name PARTITION_HIERARCHY root_partition -to |
-section_id Top
```

```
1  -- Copyright (C) 1991-2011 Altera Corporation
2  -- Your use of Altera Corporation's design tools, logic functions
3  -- and other software and tools, and its AMPP partner logic
4  -- functions, and any output files from any of the foregoing
5  -- (including device programming or simulation files), and any
6  -- associated documentation or information are expressly subject
7  -- to the terms and conditions of the Altera Program License
8  -- Subscription Agreement, Altera MegaCore Function License
9  -- Agreement, or other applicable license agreement, including,
10 -- without limitation, that your use is for the sole purpose of
11 -- programming logic devices manufactured by Altera and sold by
12 -- Altera or its authorized distributors. Please refer to the
13 -- applicable agreement for further details.
14 --
15 -- This is a Quartus II output file. It is for reporting purposes only, and is
16 -- not intended for use as a Quartus II input file. This file cannot be used
17 -- to make Quartus II pin assignments - for instructions on how to make pin
18 -- assignments, please see Quartus II help.
19 -----
20
21
22
23 -----
24 -- NC          : No Connect. This pin has no internal connection to the device.
25 -- DNU         : Do Not Use. This pin MUST NOT be connected.
26 -- VCCINT      : Dedicated power pin, which MUST be connected to VCC (1.5V).
27 -- VCCIO       : Dedicated power pin, which MUST be connected to VCC
28 --             of its bank.
29 --             Bank 1:    3.3V
30 --             Bank 2:    3.3V
31 --             Bank 3:    3.3V
32 --             Bank 4:    3.3V
33 -- GND         : Dedicated ground pin. Dedicated GND pins MUST be connected to GND.
34 --             It can also be used to report unused dedicated pins. The connection
35 --             on the board for unused dedicated pins depends on whether this will
36 --             be used in a future design. One example is device migration. When
37 --             using device migration, refer to the device pin-tables. If it is a
38 --             GND pin in the pin table or if it will not be used in a future design
```

```

39  --      for another purpose the it MUST be connected to GND. If it is an unused
40  --      dedicated pin, then it can be connected to a valid signal on the board
41  --      (low, high, or toggling) if that signal is required for a different
42  --      revision of the design.
43  -- GND+      : Unused input pin. It can also be used to report unused dual-purpose pins.
44  --      This pin should be connected to GND. It may also be connected to a
45  --      valid signal on the board (low, high, or toggling) if that signal
46  --      is required for a different revision of the design.
47  -- GND*      : Unused I/O pin. Connect each pin marked GND* directly to GND
48  --      or leave it unconnected.
49  -- RESERVED   : Unused I/O pin, which MUST be left unconnected.
50  -- RESERVED_INPUT   : Pin is tri-stated and should be connected to the board.
51  -- RESERVED_INPUT_WITH_WEAK_PULLUP   : Pin is tri-stated with internal weak pull-up resistor.
52  -- RESERVED_INPUT_WITH_BUS_HOLD      : Pin is tri-stated with bus-hold circuitry.
53  -- RESERVED_OUTPUT_DRIVEN_HIGH       : Pin is output driven high.

```

```

54  -----

```

```

55

```

```

56

```

```

57

```

```

58  -----

```

```

59  -- Pin directions (input, output or bidir) are based on device operating in user mode.

```

```

60  -----

```

```

61

```

```

62  Quartus II Version 11.0 Build 208 07/03/2011 Service Pack 1 SJ Web Edition

```

```

63  CHIP "CREAT_FpgaFirmware" ASSIGNED TO AN: EP1C20F400C6

```

```

64

```

Pin Name/Usage	Location	Dir.	I/O Standard	Voltage	I/O Bank	User Assignment
GND	A1	gnd				
VCCINT	A2	power		1.5V		
GND	A3	gnd				
WnR	A4	input	3.3-V LVTTL		2	Y
VCCIO2	A5	power		3.3V	2	
nREADY	A6	output	3.3-V LVTTL		2	Y
nINT	A7	output	3.3-V LVTTL		2	Y
A[6]	A8	input	3.3-V LVTTL		2	Y
A[24]	A9	input	3.3-V LVTTL		2	Y
nADS	A10	input	3.3-V LVTTL		2	Y

77	LAD[5]	: A11	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
78	A[23]	: A12	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y
79	A[31]	: A13	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y
80	nLRESET	: A14	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y
81	nBLAST	: A15	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y
82	VCCIO2	: A16	: power	:	:	3.3V	: 2	:	:
83	E[27]	: A17	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
84	GND	: A18	: gnd	:	:	:	:	:	:
85	VCCINT	: A19	: power	:	:	1.5V	:	:	:
86	GND	: A20	: gnd	:	:	:	:	:	:
87	VCCINT	: B1	: power	:	:	1.5V	:	:	:
88	GND	: B2	: gnd	:	:	:	:	:	:
89	FPGA[2]	: B3	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
90	B[11]	: B4	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y
91	LAD[4]	: B5	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
92	A[28]	: B6	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y
93	LAD[10]	: B7	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
94	FPGA[3]	: B8	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
95	LAD[14]	: B9	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
96	LAD[7]	: B10	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
97	A[7]	: B11	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y
98	A[27]	: B12	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y
99	LAD[8]	: B13	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
100	LAD[6]	: B14	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
101	LAD[1]	: B15	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
102	LAD[15]	: B16	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
103	LAD[0]	: B17	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
104	LAD[2]	: B18	: bidir	: 3.3-V LVTTL	:	:	: 2	:	: Y
105	GND	: B19	: gnd	:	:	:	:	:	:
106	VCCINT	: B20	: power	:	:	1.5V	:	:	:
107	GND	: C1	: gnd	:	:	:	:	:	:
108	A[2]	: C2	: input	: 3.3-V LVTTL	:	:	: 1	:	: Y
109	~LVDS31p/INIT_DONE~ / RESERVED_OUTPUT_OPEN_DRAIN	: C3	: output	: 3.3-V LVTTL	:	:	: 1	:	:
	N								
110	A[5]	: C4	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y
111	A[15]	: C5	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y
112	PULSE[1]	: C6	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y
113	B[31]	: C7	: input	: 3.3-V LVTTL	:	:	: 2	:	: Y

114	A[10]	: C8	: input	: 3.3-V LVTTL	:	:	: 2	: Y
115	A[19]	: C9	: input	: 3.3-V LVTTL	:	:	: 2	: Y
116	B[14]	: C10	: input	: 3.3-V LVTTL	:	:	: 2	: Y
117	B[12]	: C11	: input	: 3.3-V LVTTL	:	:	: 2	: Y
118	A[30]	: C12	: input	: 3.3-V LVTTL	:	:	: 2	: Y
119	D[5]	: C13	: bidir	: 3.3-V LVTTL	:	:	: 2	: Y
120	A[25]	: C14	: input	: 3.3-V LVTTL	:	:	: 2	: Y
121	PULSE[3]	: C15	: input	: 3.3-V LVTTL	:	:	: 2	: Y
122	E[30]	: C16	: bidir	: 3.3-V LVTTL	:	:	: 2	: Y
123	IDD[2]	: C17	: input	: 3.3-V LVTTL	:	:	: 2	: Y
124	D[13]	: C18	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
125	D[12]	: C19	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
126	GND	: C20	: gnd	:	:	:	:	:
127	A[9]	: D1	: input	: 3.3-V LVTTL	:	:	: 1	: Y
128	SPARE[7]	: D2	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
129	SPARE[6]	: D3	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
130	RESERVED_INPUT	: D4	:	:	:	:	: 1	:
131	A[16]	: D5	: input	: 3.3-V LVTTL	:	:	: 2	: Y
132	A[20]	: D6	: input	: 3.3-V LVTTL	:	:	: 2	: Y
133	B[30]	: D7	: input	: 3.3-V LVTTL	:	:	: 2	: Y
134	A[4]	: D8	: input	: 3.3-V LVTTL	:	:	: 2	: Y
135	A[21]	: D9	: input	: 3.3-V LVTTL	:	:	: 2	: Y
136	B[15]	: D10	: input	: 3.3-V LVTTL	:	:	: 2	: Y
137	B[13]	: D11	: input	: 3.3-V LVTTL	:	:	: 2	: Y
138	D[2]	: D12	: bidir	: 3.3-V LVTTL	:	:	: 2	: Y
139	E[29]	: D13	: bidir	: 3.3-V LVTTL	:	:	: 2	: Y
140	D[15]	: D14	: bidir	: 3.3-V LVTTL	:	:	: 2	: Y
141	RESERVED_INPUT	: D15	:	:	:	:	: 2	:
142	D[10]	: D16	: bidir	: 3.3-V LVTTL	:	:	: 2	: Y
143	nLEDR	: D17	: output	: 3.3-V LVTTL	:	:	: 3	: Y
144	DDLY[4]	: D18	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
145	D[20]	: D19	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
146	D[22]	: D20	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
147	VCCIO1	: E1	: power	:	: 3.3V	:	: 1	:
148	E[26]	: E2	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
149	FPGA[1]	: E3	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
150	FPGA[0]	: E4	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
151	A[22]	: E5	: input	: 3.3-V LVTTL	:	:	: 1	: Y

152	RESERVED_INPUT	: E6	:	:	:	: 2	:
153	B[29]	: E7	: input	: 3.3-V LVTTL	:	: 2	: Y
154	LAD[11]	: E8	: bidir	: 3.3-V LVTTL	:	: 2	: Y
155	D[4]	: E9	: bidir	: 3.3-V LVTTL	:	: 2	: Y
156	D[11]	: E10	: bidir	: 3.3-V LVTTL	:	: 2	: Y
157	LAD[12]	: E11	: bidir	: 3.3-V LVTTL	:	: 2	: Y
158	E[31]	: E12	: bidir	: 3.3-V LVTTL	:	: 2	: Y
159	D[1]	: E13	: bidir	: 3.3-V LVTTL	:	: 2	: Y
160	D[0]	: E14	: bidir	: 3.3-V LVTTL	:	: 2	: Y
161	D[14]	: E15	: bidir	: 3.3-V LVTTL	:	: 2	: Y
162	IDD[0]	: E16	: input	: 3.3-V LVTTL	:	: 2	: Y
163	D[21]	: E17	: bidir	: 3.3-V LVTTL	:	: 3	: Y
164	D[24]	: E18	: bidir	: 3.3-V LVTTL	:	: 3	: Y
165	D[25]	: E19	: bidir	: 3.3-V LVTTL	:	: 3	: Y
166	VCCIO3	: E20	: power	:	: 3.3V	: 3	:
167	E[24]	: F1	: bidir	: 3.3-V LVTTL	:	: 1	: Y
168	A[1]	: F2	: input	: 3.3-V LVTTL	:	: 1	: Y
169	PULSE[0]	: F3	: input	: 3.3-V LVTTL	:	: 1	: Y
170	B[26]	: F4	: input	: 3.3-V LVTTL	:	: 1	: Y
171	A[26]	: F5	: input	: 3.3-V LVTTL	:	: 1	: Y
172	A[3]	: F6	: input	: 3.3-V LVTTL	:	: 1	: Y
173	A[17]	: F7	: input	: 3.3-V LVTTL	:	: 2	: Y
174	B[28]	: F8	: input	: 3.3-V LVTTL	:	: 2	: Y
175	VCCIO2	: F9	: power	:	: 3.3V	: 2	:
176	RESERVED_INPUT	: F10	:	:	:	: 2	:
177	VCCIO2	: F11	: power	:	: 3.3V	: 2	:
178	D[3]	: F12	: bidir	: 3.3-V LVTTL	:	: 2	: Y
179	VCCIO2	: F13	: power	:	: 3.3V	: 2	:
180	D[6]	: F14	: bidir	: 3.3-V LVTTL	:	: 2	: Y
181	SELD	: F15	: output	: 3.3-V LVTTL	:	: 3	: Y
182	D[7]	: F16	: bidir	: 3.3-V LVTTL	:	: 3	: Y
183	D[19]	: F17	: bidir	: 3.3-V LVTTL	:	: 3	: Y
184	GIN[1]	: F18	: input	: 3.3-V LVTTL	:	: 3	: Y
185	D[8]	: F19	: bidir	: 3.3-V LVTTL	:	: 3	: Y
186	D[23]	: F20	: bidir	: 3.3-V LVTTL	:	: 3	: Y
187	E[25]	: G1	: bidir	: 3.3-V LVTTL	:	: 1	: Y
188	B[27]	: G2	: input	: 3.3-V LVTTL	:	: 1	: Y
189	A[0]	: G3	: input	: 3.3-V LVTTL	:	: 1	: Y

190	B[25]	: G4	: input	: 3.3-V LVTTL	:	:	: 1	: Y
191	A[14]	: G5	: input	: 3.3-V LVTTL	:	:	: 1	: Y
192	B[10]	: G6	: input	: 3.3-V LVTTL	:	:	: 1	: Y
193	LAD[9]	: G7	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
194	GND	: G8	: gnd	:	:	:	:	:
195	GND	: G9	: gnd	:	:	:	:	:
196	GND	: G10	: gnd	:	:	:	:	:
197	GND	: G11	: gnd	:	:	:	:	:
198	GND	: G12	: gnd	:	:	:	:	:
199	GND	: G13	: gnd	:	:	:	:	:
200	D[16]	: G14	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
201	E[8]	: G15	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
202	D[18]	: G16	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
203	D[17]	: G17	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
204	E[15]	: G18	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
205	E[11]	: G19	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
206	E[13]	: G20	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
207	A[11]	: H1	: input	: 3.3-V LVTTL	:	:	: 1	: Y
208	A[12]	: H2	: input	: 3.3-V LVTTL	:	:	: 1	: Y
209	A[13]	: H3	: input	: 3.3-V LVTTL	:	:	: 1	: Y
210	B[24]	: H4	: input	: 3.3-V LVTTL	:	:	: 1	: Y
211	B[9]	: H5	: input	: 3.3-V LVTTL	:	:	: 1	: Y
212	A[18]	: H6	: input	: 3.3-V LVTTL	:	:	: 1	: Y
213	SPARE[11]	: H7	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
214	VCCIO1	: H8	: power	:	:	: 3.3V	: 1	:
215	GND	: H9	: gnd	:	:	:	:	:
216	VCCINT	: H10	: power	:	:	: 1.5V	:	:
217	GND	: H11	: gnd	:	:	:	:	:
218	VCCINT	: H12	: power	:	:	: 1.5V	:	:
219	VCCIO3	: H13	: power	:	:	: 3.3V	: 3	:
220	LAD[13]	: H14	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
221	D[9]	: H15	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
222	IDE[0]	: H16	: input	: 3.3-V LVTTL	:	:	: 3	: Y
223	E[12]	: H17	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
224	E[10]	: H18	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
225	E[9]	: H19	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
226	RESERVED_INPUT	: H20	:	:	:	:	: 3	:
227	LAD[3]	: J1	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y



228	nOED	: J2	: output	: 3.3-V LVTTL	:	:	: 1	: Y
229	A[29]	: J3	: input	: 3.3-V LVTTL	:	:	: 1	: Y
230	E[14]	: J4	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
231	A[8]	: J5	: input	: 3.3-V LVTTL	:	:	: 1	: Y
232	IDD[1]	: J6	: input	: 3.3-V LVTTL	:	:	: 1	: Y
233	SPARE[10]	: J7	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
234	RESERVED_INPUT	: J8	:	:	:	:	: 1	:
235	VCCINT	: J9	: power	:	:	: 1.5V	:	:
236	GND	: J10	: gnd	:	:	:	:	:
237	VCCINT	: J11	: power	:	:	: 1.5V	:	:
238	GND	: J12	: gnd	:	:	:	:	:
239	IDE[1]	: J13	: input	: 3.3-V LVTTL	:	:	: 3	: Y
240	B[8]	: J14	: input	: 3.3-V LVTTL	:	:	: 3	: Y
241	E[28]	: J15	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
242	D[28]	: J16	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
243	D[31]	: J17	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
244	D[26]	: J18	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
245	RESERVED_INPUT	: J19	:	:	:	:	: 3	:
246	RESERVED_INPUT	: J20	:	:	:	:	: 3	:
247	nCONFIG	: K1	:	:	:	:	: 1	:
248	RESERVED_INPUT	: K2	:	:	:	:	: 1	:
249	DATA0	: K3	: input	:	:	:	: 1	:
250	VCCA_PLL1	: K4	: power	:	:	: 1.5V	:	:
251	LCLK	: K5	: input	: 3.3-V LVTTL	:	:	: 1	: Y
252	GND+	: K6	:	:	:	:	: 1	:
253	GND_A_PLL1	: K7	: gnd	:	:	:	:	:
254	VCCIO1	: K8	: power	:	:	: 3.3V	: 1	:
255	GND	: K9	: gnd	:	:	:	:	:
256	VCCINT	: K10	: power	:	:	: 1.5V	:	:
257	GND	: K11	: gnd	:	:	:	:	:
258	VCCINT	: K12	: power	:	:	: 1.5V	:	:
259	VCCIO3	: K13	: power	:	:	: 3.3V	: 3	:
260	GND+	: K14	:	:	:	:	: 3	:
261	D[27]	: K15	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
262	D[30]	: K16	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
263	VCCA_PLL2	: K17	: power	:	:	: 1.5V	:	:
264	TDI	: K18	: input	:	:	:	: 3	:
265	nLEDG	: K19	: output	: 3.3-V LVTTL	:	:	: 3	: Y

266	GND_A_PLL2	: K20	: gnd	:	:	:	:
267	MSEL0	: L1	:	:	:	: 1	:
268	nCEO	: L2	:	:	:	: 1	:
269	DCLK	: L3	: bidir	:	:	: 1	:
270	RESERVED_INPUT	: L4	:	:	:	: 1	:
271	nCE	: L5	:	:	:	: 1	:
272	MSEL1	: L6	:	:	:	: 1	:
273	GNDG_PLL1	: L7	: gnd	:	:	:	:
274	RESERVED_INPUT	: L8	:	:	:	: 1	:
275	VCCINT	: L9	: power	:	: 1.5V	:	:
276	GND	: L10	: gnd	:	:	:	:
277	VCCINT	: L11	: power	:	: 1.5V	:	:
278	GND	: L12	: gnd	:	:	:	:
279	GOUT[1]	: L13	: output	: 3.3-V LVTTL	:	: 3	: Y
280	GIN[0]	: L14	: input	: 3.3-V LVTTL	:	: 3	: Y
281	GNDG_PLL2	: L15	: gnd	:	:	:	:
282	TMS	: L16	: input	:	:	: 3	:
283	nSTATUS	: L17	:	:	:	: 3	:
284	CONF_DONE	: L18	:	:	:	: 3	:
285	TCK	: L19	: input	:	:	: 3	:
286	TDO	: L20	: output	:	:	: 3	:
287	B[23]	: M1	: input	: 3.3-V LVTTL	:	: 1	: Y
288	B[7]	: M2	: input	: 3.3-V LVTTL	:	: 1	: Y
289	C[4]	: M3	: output	: 3.3-V LVTTL	:	: 1	: Y
290	B[6]	: M4	: input	: 3.3-V LVTTL	:	: 1	: Y
291	C[31]	: M5	: output	: 3.3-V LVTTL	:	: 1	: Y
292	F[2]	: M6	: bidir	: 3.3-V LVTTL	:	: 1	: Y
293	B[21]	: M7	: input	: 3.3-V LVTTL	:	: 1	: Y
294	RESERVED_INPUT	: M8	:	:	:	: 1	:
295	GND	: M9	: gnd	:	:	:	:
296	VCCINT	: M10	: power	:	: 1.5V	:	:
297	GND	: M11	: gnd	:	:	:	:
298	VCCINT	: M12	: power	:	: 1.5V	:	:
299	RESERVED_INPUT	: M13	:	:	:	: 3	:
300	IDE[2]	: M14	: input	: 3.3-V LVTTL	:	: 3	: Y
301	D[29]	: M15	: bidir	: 3.3-V LVTTL	:	: 3	: Y
302	DDLY[1]	: M16	: bidir	: 3.3-V LVTTL	:	: 3	: Y
303	RESERVED_INPUT	: M17	:	:	:	: 3	:

304	RESERVED_INPUT	: M18	:	:	:	:	3	:
305	RESERVED_INPUT	: M19	:	:	:	:	3	:
306	RESERVED_INPUT	: M20	:	:	:	:	3	:
307	E[23]	: N1	: bidir	: 3.3-V LVTTL	:	:	1	: Y
308	E[22]	: N2	: bidir	: 3.3-V LVTTL	:	:	1	: Y
309	B[22]	: N3	: input	: 3.3-V LVTTL	:	:	1	: Y
310	C[18]	: N4	: output	: 3.3-V LVTTL	:	:	1	: Y
311	SPARE[9]	: N5	: bidir	: 3.3-V LVTTL	:	:	1	: Y
312	F[11]	: N6	: bidir	: 3.3-V LVTTL	:	:	1	: Y
313	F[10]	: N7	: bidir	: 3.3-V LVTTL	:	:	1	: Y
314	VCCIO1	: N8	: power	:	:	3.3V	1	:
315	VCCINT	: N9	: power	:	:	1.5V	:	:
316	GND	: N10	: gnd	:	:	:	:	:
317	VCCINT	: N11	: power	:	:	1.5V	:	:
318	GND	: N12	: gnd	:	:	:	:	:
319	VCCIO3	: N13	: power	:	:	3.3V	3	:
320	nOEE	: N14	: output	: 3.3-V LVTTL	:	:	3	: Y
321	F[14]	: N15	: bidir	: 3.3-V LVTTL	:	:	3	: Y
322	DDLY[5]	: N16	: bidir	: 3.3-V LVTTL	:	:	3	: Y
323	DDLY[2]	: N17	: bidir	: 3.3-V LVTTL	:	:	3	: Y
324	RESERVED_INPUT	: N18	:	:	:	:	3	:
325	RESERVED_INPUT	: N19	:	:	:	:	3	:
326	SPARE[5]	: N20	: bidir	: 3.3-V LVTTL	:	:	3	: Y
327	E[7]	: P1	: bidir	: 3.3-V LVTTL	:	:	1	: Y
328	B[1]	: P2	: input	: 3.3-V LVTTL	:	:	1	: Y
329	B[16]	: P3	: input	: 3.3-V LVTTL	:	:	1	: Y
330	C[26]	: P4	: output	: 3.3-V LVTTL	:	:	1	: Y
331	B[0]	: P5	: input	: 3.3-V LVTTL	:	:	1	: Y
332	B[20]	: P6	: input	: 3.3-V LVTTL	:	:	1	: Y
333	E[17]	: P7	: bidir	: 3.3-V LVTTL	:	:	1	: Y
334	GND	: P8	: gnd	:	:	:	:	:
335	GND	: P9	: gnd	:	:	:	:	:
336	GND	: P10	: gnd	:	:	:	:	:
337	GND	: P11	: gnd	:	:	:	:	:
338	GND	: P12	: gnd	:	:	:	:	:
339	GND	: P13	: gnd	:	:	:	:	:
340	F[7]	: P14	: bidir	: 3.3-V LVTTL	:	:	3	: Y
341	nOEG	: P15	: output	: 3.3-V LVTTL	:	:	3	: Y

342	SELG	: P16	: output	: 3.3-V LVTTL	:	:	: 3	: Y
343	noEDDLY0	: P17	: output	: 3.3-V LVTTL	:	:	: 3	: Y
344	F[8]	: P18	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
345	SPARE[4]	: P19	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
346	GOUT[0]	: P20	: output	: 3.3-V LVTTL	:	:	: 3	: Y
347	C[10]	: R1	: output	: 3.3-V LVTTL	:	:	: 1	: Y
348	C[7]	: R2	: output	: 3.3-V LVTTL	:	:	: 1	: Y
349	C[23]	: R3	: output	: 3.3-V LVTTL	:	:	: 1	: Y
350	C[17]	: R4	: output	: 3.3-V LVTTL	:	:	: 1	: Y
351	C[14]	: R5	: output	: 3.3-V LVTTL	:	:	: 1	: Y
352	C[6]	: R6	: output	: 3.3-V LVTTL	:	:	: 1	: Y
353	C[8]	: R7	: output	: 3.3-V LVTTL	:	:	: 4	: Y
354	VCCIO4	: R8	: power	:	: 3.3V	:	: 4	:
355	F[23]	: R9	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
356	VCCIO4	: R10	: power	:	: 3.3V	:	: 4	:
357	E[4]	: R11	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
358	VCCIO4	: R12	: power	:	: 3.3V	:	: 4	:
359	E[6]	: R13	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
360	F[19]	: R14	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
361	noEDDLY1	: R15	: output	: 3.3-V LVTTL	:	:	: 3	: Y
362	F[5]	: R16	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
363	F[9]	: R17	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
364	IDF[1]	: R18	: input	: 3.3-V LVTTL	:	:	: 3	: Y
365	nSTART[3]	: R19	: output	: 3.3-V LVTTL	:	:	: 3	: Y
366	nSTART[2]	: R20	: output	: 3.3-V LVTTL	:	:	: 3	: Y
367	VCCIO1	: T1	: power	:	: 3.3V	:	: 1	:
368	F[15]	: T2	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
369	E[21]	: T3	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
370	PULSE[2]	: T4	: input	: 3.3-V LVTTL	:	:	: 1	: Y
371	C[12]	: T5	: output	: 3.3-V LVTTL	:	:	: 4	: Y
372	E[2]	: T6	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
373	RESERVED_INPUT	: T7	:	:	:	:	: 4	:
374	F[31]	: T8	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
375	F[28]	: T9	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
376	F[6]	: T10	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
377	F[20]	: T11	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
378	F[29]	: T12	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
379	F[0]	: T13	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y

380	F[24]	: T14	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
381	F[17]	: T15	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
382	C[9]	: T16	: output	: 3.3-V LVTTL	:	:	: 4	: Y
383	F[4]	: T17	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
384	SELE	: T18	: output	: 3.3-V LVTTL	:	:	: 3	: Y
385	RESERVED_INPUT	: T19	:	:	:	:	: 3	:
386	VCCIO3	: T20	: power	:	: 3.3V	:	: 3	:
387	RESERVED_INPUT	: U1	:	:	:	:	: 1	:
388	C[5]	: U2	: output	: 3.3-V LVTTL	:	:	: 1	: Y
389	C[22]	: U3	: output	: 3.3-V LVTTL	:	:	: 1	: Y
390	SPARE[8]	: U4	: bidir	: 3.3-V LVTTL	:	:	: 1	: Y
391	C[13]	: U5	: output	: 3.3-V LVTTL	:	:	: 4	: Y
392	B[5]	: U6	: input	: 3.3-V LVTTL	:	:	: 4	: Y
393	C[28]	: U7	: output	: 3.3-V LVTTL	:	:	: 4	: Y
394	B[4]	: U8	: input	: 3.3-V LVTTL	:	:	: 4	: Y
395	C[24]	: U9	: output	: 3.3-V LVTTL	:	:	: 4	: Y
396	E[18]	: U10	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
397	F[21]	: U11	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
398	E[5]	: U12	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
399	C[0]	: U13	: output	: 3.3-V LVTTL	:	:	: 4	: Y
400	F[16]	: U14	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
401	DDLY[0]	: U15	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
402	DDLY[6]	: U16	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
403	RESERVED_INPUT	: U17	:	:	:	:	: 3	:
404	F[13]	: U18	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
405	SPARE[0]	: U19	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
406	SPARE[2]	: U20	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
407	GND	: V1	: gnd	:	:	:	:	:
408	C[11]	: V2	: output	: 3.3-V LVTTL	:	:	: 1	: Y
409	B[17]	: V3	: input	: 3.3-V LVTTL	:	:	: 1	: Y
410	E[16]	: V4	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
411	C[16]	: V5	: output	: 3.3-V LVTTL	:	:	: 4	: Y
412	C[20]	: V6	: output	: 3.3-V LVTTL	:	:	: 4	: Y
413	C[29]	: V7	: output	: 3.3-V LVTTL	:	:	: 4	: Y
414	C[2]	: V8	: output	: 3.3-V LVTTL	:	:	: 4	: Y
415	F[30]	: V9	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
416	E[0]	: V10	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
417	F[22]	: V11	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y

418	IDF[2]	: V12	: input	: 3.3-V LVTTL	:	:	: 4	: Y
419	E[20]	: V13	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
420	F[18]	: V14	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
421	DDL[7]	: V15	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
422	START[1]	: V16	: output	: 3.3-V LVTTL	:	:	: 4	: Y
423	START[0]	: V17	: output	: 3.3-V LVTTL	:	:	: 4	: Y
424	F[12]	: V18	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
425	SPARE[3]	: V19	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
426	GND	: V20	: gnd	:	:	:	:	:
427	VCCINT	: W1	: power	:	:	: 1.5V	:	:
428	GND	: W2	: gnd	:	:	:	:	:
429	C[27]	: W3	: output	: 3.3-V LVTTL	:	:	: 4	: Y
430	E[19]	: W4	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
431	SELF	: W5	: output	: 3.3-V LVTTL	:	:	: 4	: Y
432	B[2]	: W6	: input	: 3.3-V LVTTL	:	:	: 4	: Y
433	C[3]	: W7	: output	: 3.3-V LVTTL	:	:	: 4	: Y
434	B[18]	: W8	: input	: 3.3-V LVTTL	:	:	: 4	: Y
435	B[19]	: W9	: input	: 3.3-V LVTTL	:	:	: 4	: Y
436	C[25]	: W10	: output	: 3.3-V LVTTL	:	:	: 4	: Y
437	F[26]	: W11	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
438	IDF[0]	: W12	: input	: 3.3-V LVTTL	:	:	: 4	: Y
439	F[3]	: W13	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
440	DIRDDL	: W14	: output	: 3.3-V LVTTL	:	:	: 4	: Y
441	E[3]	: W15	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
442	DDL[3]	: W16	: bidir	: 3.3-V LVTTL	:	:	: 4	: Y
443	WR_DLY1	: W17	: output	: 3.3-V LVTTL	:	:	: 4	: Y
444	SPARE[1]	: W18	: bidir	: 3.3-V LVTTL	:	:	: 3	: Y
445	GND	: W19	: gnd	:	:	:	:	:
446	VCCINT	: W20	: power	:	:	: 1.5V	:	:
447	GND	: Y1	: gnd	:	:	:	:	:
448	VCCINT	: Y2	: power	:	:	: 1.5V	:	:
449	GND	: Y3	: gnd	:	:	:	:	:
450	C[1]	: Y4	: output	: 3.3-V LVTTL	:	:	: 4	: Y
451	VCCIO4	: Y5	: power	:	:	: 3.3V	: 4	:
452	C[19]	: Y6	: output	: 3.3-V LVTTL	:	:	: 4	: Y
453	C[15]	: Y7	: output	: 3.3-V LVTTL	:	:	: 4	: Y
454	B[3]	: Y8	: input	: 3.3-V LVTTL	:	:	: 4	: Y
455	C[21]	: Y9	: output	: 3.3-V LVTTL	:	:	: 4	: Y

456	C[30]	: Y10	: output	: 3.3-V LVTTL	:	:	: 4	:	: Y
457	F[27]	: Y11	: bidir	: 3.3-V LVTTL	:	:	: 4	:	: Y
458	F[1]	: Y12	: bidir	: 3.3-V LVTTL	:	:	: 4	:	: Y
459	nOEF	: Y13	: output	: 3.3-V LVTTL	:	:	: 4	:	: Y
460	F[25]	: Y14	: bidir	: 3.3-V LVTTL	:	:	: 4	:	: Y
461	E[1]	: Y15	: bidir	: 3.3-V LVTTL	:	:	: 4	:	: Y
462	VCCIO4	: Y16	: power	:	:	: 3.3V	: 4	:	:
463	WR_DLY0	: Y17	: output	: 3.3-V LVTTL	:	:	: 4	:	: Y
464	GND	: Y18	: gnd	:	:	:	:	:	:
465	VCCINT	: Y19	: power	:	:	: 1.5V	:	:	:
466	GND	: Y20	: gnd	:	:	:	:	:	:
467									

## **APENDICE G: Código C para análise de dados no ROOT**



```

1  // #include "Riostream.h"
2
3  // ===== Analysis code for CRE@AT experiment
4  // ===== Ulisses Carneiro (ulissestv@gmail)
5  // ===== Baseado na obra de André Massafferri (massafferri@cbpf.br)
6  // > .L Analysis_CREAT.C
7
8  // for data (2 runs) collected in september in CBPF
9  // > makeTree("data/RUN-land2-CBPF-2014.dat", "tree_cbpf.root")
10
11 // for data (2 runs) collected in december Criosfera/Antartida
12 // > makeTree("data/RUN-land2-CRIOSFERA-2014.dat", "tree_criosfera.root")
13
14 // arguments: nbin, fluxo_min, fluxo_max, "cut", norm0, mean0, width0
15 // for CBPF data: sugestion
16 // > results("tree_cbpf.root", 20, 1, 11, "vth>130", 10, 3, 0.5)
17 // for Criosfera data
18 // > results("tree_criosfera.root", 20, 1, 15, "vth>80 && vth!=140", 30, 7, 0.5)
19
20 Double_t funGauss(Double_t *x, Double_t *par)
21 {
22     Double_t G;
23     Double_t cte = 2.5066297;
24     Double_t N, Function;
25     N = 1.0/(cte*par[2]);
26     Function = N*(TMath::Exp(- (par[1]-x[0]) * (par[1] - x[0]) / (2.0 * par[2] * par
27 [2]) ) );
28     G = par[0]*Function;
29     return G;
30 }
31
32 void makeTree(TString namedata, TString nametree = "tree.root") {
33     ifstream in;
34     in.open(namedata.Data());
35
36     Int_t nrun = 0;
37     TFile *f = new TFile(nametree, "RECREATE");
38     Char_t evtId[10];
39     Int_t dt;
40     Char_t dt_string[50] ;
41
42     Float_t vThr, counter;
43     Float_t l_23_0, l_23_1, l_23_2, l_23_3, l_23_4, l_23_5;
44     Float_t l_13_0, l_13_1, l_13_2, l_13_3, l_13_4, l_13_5;
45     Float_t l_12_0, l_12_1, l_12_2, l_12_3, l_12_4, l_12_5;
46     Float_t l_3_0, l_3_1, l_3_2, l_3_3, l_3_4, l_3_5;
47     Float_t f1_0, f1_1, f1_2, f1_3, f1_4, f1_5;
48     Float_t f2_0, f2_1, f2_2, f2_3, f2_4, f2_5;
49     Float_t f3_0, f3_1, f3_2, f3_3, f3_4, f3_5;
50     Float_t e1_0, e1_1, e1_2, e1_3, e1_4, e1_5;
51     Float_t e2_0, e2_1, e2_2, e2_3, e2_4, e2_5;
52     Float_t e3_0, e3_1, e3_2, e3_3, e3_4, e3_5;
53     Float_t fl_0, fl_1, fl_2, fl_3, fl_4, fl_5;
54     Float_t efl_0, efl_1, efl_2, efl_3, efl_4, efl_5;
55
56     // TNtuple *creat = new TNtuple("creat", "cosmics

```

```

    flux", "fl_0:fl_1:fl_2:fl_3:fl_4:fl_5:nrun");
57 TTree creat("creat", "creat");
58 creat.Branch("fl_0", &fl_0, "fl_0/F");
59 creat.Branch("fl_1", &fl_1, "fl_1/F");
60 creat.Branch("fl_2", &fl_2, "fl_2/F");
61 creat.Branch("fl_3", &fl_3, "fl_3/F");
62 creat.Branch("fl_4", &fl_4, "fl_4/F");
63 creat.Branch("nrun", &nrun, "nrun/I");
64 creat.Branch("vth", &vThr, "vth/F");
65
66 while (1) {
67     in >> vThr >> dt >> dt_string >> evtId >> counter >> l_23_0 >> l_23_1 >>
68     l_23_2 >> l_23_3 >> l_23_4 >> l_23_5 >> l_13_0 >> l_13_1 >> l_13_2 >>
69     l_13_3 >> l_13_4 >> l_13_5 >> l_12_0 >> l_12_1 >> l_12_2 >> l_12_3 >>
70     l_12_4 >> l_12_5 >> l_3_0 >> l_3_1 >> l_3_2 >> l_3_3 >> l_3_4 >>
71     l_3_5 >> fl_0 >> fl_1 >> fl_2 >> fl_3 >> fl_4 >> fl_5 >> f2_0 >>
72     f2_1 >> f2_2 >> f2_3 >> f2_4 >> f2_5 >> f3_0 >> f3_1 >> f3_2 >>
73     f3_3 >> f3_4 >> f3_5 >> e1_0 >> e1_1 >> e1_2 >> e1_3 >> e1_4 >>
74     e1_5 >> e2_0 >> e2_1 >> e2_2 >> e2_3 >> e2_4 >> e2_5 >> e3_0 >>
75     e3_1 >> e3_2 >> e3_3 >> e3_4 >> e3_5 >> fl_0 >> fl_1 >> fl_2 >>
76     fl_3 >> fl_4 >> fl_5 >> efl_0 >> efl_1 >> efl_2 >> efl_3 >> efl_4 >> efl_5;
77
78     if (!in.good()) break;
79     if (nrun < 30) printf("%d: %8f -- %8f, %8f, %8f, %8f, %8f \n", nrun, vThr, fl_0,
80     fl_1, fl_2, fl_3, fl_4);
81     //creat->Fill(nrun, fl_0, fl_1, fl_2, fl_3, fl_4, fl_5);
82     creat.Fill();
83     nrun++;
84 }
85 printf(" found %d lines \n", nrun);
86 in.close();
87 f->Write();
88 }
89
90 Double_t results(TString namefile = "tree.root", Int_t nbin = 20, Float_t min =
91 4.0, Float_t max = 14.0,
92     TString cut = "vth>80 && vth!=140", Double_t norm0 = 50, Double_t
93     mean0 = 7, Double_t width0 = 0.5)
94 {
95     int flag_print = 1; // 0=CBPF, 1=CRIOSFERA
96     Int_t doPrint = 1;
97
98     Float_t fl_0, fl_1, fl_2, fl_3, fl_4, fl_5, vth;
99     Int_t nrun;
100     Float_t area = 5.0 * 49.5 / 10000.0; //area in m2
101     Int_t tacq = 600; //sec
102     Float_t clevel = 0.02;
103     Float_t maxrms = 3.0;
104
105     TString nameGraph, namepdf1, namepdf2;
106
107     if (flag_print==0){
108         nameGraph = "preliminar: Fluxo cosmicos no CBPF";
109         namepdf1 = "creat_fits_CBPF.pdf";
110         namepdf2 = "creat_grafico_CBPF.pdf";

```

```

110     } else {
111         nameGraph = "preliminar: Fluxo cosmicos Criosfera";
112         namepdf1 = "creat_fits_Criosfera.pdf";
113         namepdf2 = "creat_grafico_Criosfera.pdf";
114     }
115
116     TFile *file = new TFile(namefile,"UPDATE");
117
118     TTree *creat = (TTree*)file->Get("creat");
119     creat->SetBranchAddress("fl_0",&fl_0);
120     creat->SetBranchAddress("fl_1",&fl_1);
121     creat->SetBranchAddress("fl_2",&fl_2);
122     creat->SetBranchAddress("fl_3",&fl_3);
123     creat->SetBranchAddress("fl_4",&fl_4);
124     creat->SetBranchAddress("vth",&vth);
125     creat->SetBranchAddress("nrun",&nrun);
126
127     Int_t nentries = (TTree*)creat->GetEntries(cut);
128
129     ///// c1: flux x Vth
130
131     TH2 *h0_2,*h1_2,*h2_2,*h3_2,*h4_2;
132     h0_2 = new TH2F ("h0_2","fl_0 x vth",nbin,30,220,nbin,min,max);
133     h1_2 = new TH2F ("h1_2","fl_1 x vth",nbin,30,220,nbin,min,max);
134     h2_2 = new TH2F ("h2_2","fl_2 x vth",nbin,30,220,nbin,min,max);
135     h3_2 = new TH2F ("h3_2","fl_3 x vth",nbin,30,220,nbin,min,max);
136     h4_2 = new TH2F ("h4_2","fl_4 x vth",nbin,30,220,nbin,min,max);
137
138     creat->Project("h0_2","fl_0:vth");
139     creat->Project("h1_2","fl_1:vth");
140     creat->Project("h2_2","fl_2:vth");
141     creat->Project("h3_2","fl_3:vth");
142     creat->Project("h4_2","fl_4:vth");
143
144     c1 = new TCanvas ("c1", "creat", 200, 10, 600, 700);
145     c1->Divide(2,3);
146     c1->cd(1);
147     h0_2->Draw("BOX");
148     c1->cd(2);
149     h1_2->Draw("BOX");
150     c1->cd(3);
151     h2_2->Draw("BOX");
152     c1->cd(4);
153     h3_2->Draw("BOX");
154     c1->cd(5);
155     h4_2->Draw("BOX");
156
157     ///// c2: fit of each group integrating in Vth given a Vthcut
158
159     TH1 *h0,*h1,*h2,*h3,*h4;
160     h0 = new TH1F ("h0 - CBPF","GRUPO 1",nbin,min,max);
161     h1 = new TH1F ("h1 - CBPF","GRUPO 2",nbin,min,max);
162     h2 = new TH1F ("h2 - CBPF","GRUPO 3",nbin,min,max);
163     h3 = new TH1F ("h3 - CBPF","GRUPO 4",nbin,min,max);
164     h4 = new TH1F ("h4 - CBPF","GRUPO 5",nbin,min,max);
165
166     creat->Project("h0 - CBPF","fl_0",cut);

```

```
167 creat->Project("h1 - CBPF", "fl_1", cut);
168 creat->Project("h2 - CBPF", "fl_2", cut);
169 creat->Project("h3 - CBPF", "fl_3", cut);
170 creat->Project("h4 - CBPF", "fl_4", cut);
171
172 TF1 *f1 = new TF1("f1", funGauss, min, max, 3);
173 f1->SetParNames("norm", "mean", "width");
174 f1->SetLineColor(2);
175
176 h0->GetXaxis()->SetTitle("Fluxo [Trio-Scitile/s]");
177 h1->GetXaxis()->SetTitle("Fluxo [Trio-Scitile/s]");
178 h2->GetXaxis()->SetTitle("Fluxo [Trio-Scitile/s]");
179 h3->GetXaxis()->SetTitle("Fluxo [Trio-Scitile/s]");
180 h4->GetXaxis()->SetTitle("Fluxo [Trio-Scitile/s]");
181
182 h0->GetYaxis()->SetTitle("nruns@10mins Vth cut");
183 h1->GetYaxis()->SetTitle("nruns@10mins Vth cut");
184 h2->GetYaxis()->SetTitle("nruns@10mins Vth cut");
185 h3->GetYaxis()->SetTitle("nruns@10mins Vth cut");
186 h4->GetYaxis()->SetTitle("nruns@10mins Vth cut");
187
188 gStyle->SetOptFit(1011);
189 //gPad->SetFrameFillColor(33);
190
191 h0->SetFillColor(48);
192 h1->SetFillColor(48);
193 h2->SetFillColor(48);
194 h3->SetFillColor(48);
195 h4->SetFillColor(48);
196
197 h0->SetLineWidth(2);
198 h1->SetLineWidth(2);
199 h2->SetLineWidth(2);
200 h3->SetLineWidth(2);
201 h4->SetLineWidth(2);
202
203 const int num = 5;
204 Double_t pars0[3], pars1[3], pars2[3], pars3[3], pars4[3], prob[5];
205
206 c2 = new TCanvas("c2", "creat", 200, 10, 900, 600);
207 c2->Divide(3, 2);
208
209 c2->cd(1);
210 f1->SetParameter(0, norm0);
211 f1->SetParameter(1, mean0);
212 f1->SetParameter(2, width0);
213 h0->Fit(f1, "R");
214 f1->GetParameters(&pars0[0]);
215 prob[0] = f1->GetProb();
216
217 c2->cd(2);
218 f1->SetParameter(0, norm0);
219 f1->SetParameter(1, mean0);
220 f1->SetParameter(2, width0);
221 h1->Fit(f1, "R");
222 f1->GetParameters(&pars1[0]);
223 prob[1] = f1->GetProb();
```

```
224
225     c2->cd(3);
226     f1->SetParameter(0,norm0);
227     f1->SetParameter(1,mean0);
228     f1->SetParameter(2,width0);
229     h2->Fit(f1,"R");
230     f1->GetParameters(&pars2[0]);
231     prob[2] = f1->GetProb();
232
233     c2->cd(4);
234     f1->SetParameter(0,norm0);
235     f1->SetParameter(1,mean0);
236     f1->SetParameter(2,width0);
237     h3->Fit(f1,"R");
238     f1->GetParameters(&pars3[0]);
239     prob[3] = f1->GetProb();
240
241     c2->cd(5);
242     f1->SetParameter(0,norm0);
243     f1->SetParameter(1,mean0);
244     f1->SetParameter(2,width0);
245     h4->Fit(f1,"R");
246     f1->GetParameters(&pars4[0]);
247     prob[4] = f1->GetProb();
248
249     if (doPrint==1) c2->SaveAs(namepdf1);
250
251     ///// c3: graph, final result
252
253     Double_t fluxo[num],rms[num];
254     Double_t media_fluxo = 0;
255     Double_t rms_quad = 0;
256     Double_t Grupos[num] = {1.0,2.0,3.0,4.0,5.0};
257     Double_t eGrupos[num] = {0.0,0.0,0.0,0.0,0.0};
258
259     fluxo[0] = pars0[1];
260     fluxo[1] = pars1[1];
261     fluxo[2] = pars2[1];
262     fluxo[3] = pars3[1];
263     fluxo[4] = pars4[1];
264
265     rms[0] = pars0[2];
266     rms[1] = pars1[2];
267     rms[2] = pars2[2];
268     rms[3] = pars3[2];
269     rms[4] = pars4[2];
270
271     Int_t goodfit=0;
272     for (int i=0; i<num; i++){
273         if (prob[i]<clevel || rms[i]>maxrms){
274             fluxo[i] = 0.0;
275             rms[i] = 0.0;
276         } else {
277             goodfit++;
278         }
279         media_fluxo = media_fluxo + fluxo[i];
280         rms_quad = rms_quad + rms[i]**2;
```

```

281     }
282     media_fluxo = media_fluxo/(goodfit*area);
283     rms_quad = sqrt(rms_quad)/area;
284
285     cout << " ----- " << endl;
286     cout << " ----- Results ----- " << endl;
287     cout << media_fluxo << " +- " << rms_quad << " muons s-1 m-2" << endl;
288     cout << " number of good fits@CL=" << clevel << " = " << goodfit << endl;
289     cout << " CUT: " << cut << " nentries = " << nentries << " of TAC = " << tacq
<< endl;
290     cout << " ----- " << endl;
291
292     Char_t resultado[150];
293     sprintf(resultado, "#font[12]{#color[2]{fluxo = [%3.0f +- %3.0f] muons s^{-1}
m^{-2}}}", media_fluxo, rms_quad);
294     Char_t cuttext[150];
295     sprintf(cuttext, "#font[12]{#color[4]{corte %s : %d #it{runs} de %3d segs}}", cut.
Data(), nentries, tacq);
296
297     c3 = new TCanvas ("c3", "Resultado", 200, 10, 900, 600);
298     //c3->SetFillColor(23);
299     TPad* current_pad;
300
301     TGraphErrors *g_fluxo;
302     g_fluxo = new TGraphErrors(num, Grupos, fluxo, eGrupos, rms);
303     g_fluxo->SetTitle(nameGraph);
304     g_fluxo->SetMarkerColor(4);
305     g_fluxo->SetMarkerStyle(21);
306     g_fluxo->SetMinimum(0);
307     g_fluxo->GetYaxis()->SetTitle("Fluxo na area sensivel [Trio-SciTile/s]");
308     g_fluxo->GetXaxis()->SetTitle("Grupo de coincidência (Trio-SciTile)");
309     g_fluxo->Draw("AP");
310
311     TLatex res;
312     res.SetTextAlign(1);
313     res.SetTextSize(0.05);
314     res.DrawLatex(1.7, 1.5, resultado);
315     res.SetTextSize(0.05);
316     res.DrawLatex(1.7, 1, cuttext);
317
318     c3->Update();
319     if (doPrint==1) c3->SaveAs(namepdf2);
320
321     return media_fluxo;
322     //return rms_quad;
323 }
324
325 //void run(Char_t name[50] = "error.dat"){
326 void run(){
327
328     //FILE *pFile;
329     //pFile = fopen (name, "w");
330     const int dim = 19;
331     Char_t vth_cut[30];
332     Double_t spot[dim];
333     Double_t vth[dim];
334     Double_t results(TString, Int_t, Float_t, Float_t, TString, Double_t, Double_t,

```

```
    Double_t);
335    TGraph *gerr;
336
337    for (Int_t i=0; i<dim; i++){
338        Float_t temp = 10*i+30;
339        sprintf(vth_cut, "vth>%f", temp);
340        spot[i] = results("tree_criosfera.root", 40, 1, 16, vth_cut, 30, 7, 1);
341        //spot[i] = results("tree_cbpf.root", 20, 1, 11, vth_cut, 10, 3, 0.5);
342        vth[i] = temp;
343        //fprintf(pFile, "%f \n", error);
344    }
345
346    c_e = new TCanvas ("c_e", "spot dist", 200, 10, 600, 700);
347    c_e ->SetFillColor(23);
348    gerr = new TGraph(dim, vth, spot);
349    gerr->SetTitle("spot dist");
350    gerr->SetMarkerColor(4);
351    gerr->SetMarkerStyle(21);
352    gerr->SetMinimum(0);
353    Double_t max = 300;
354    gerr->SetMaximum(1.3*max);
355    gerr->GetYaxis()->SetTitle("spot value");
356    gerr->GetXaxis()->SetTitle("Vth(reg)");
357    gerr->Draw("AP");
358
359 }
360
361
362
363
```

## **ANEXO A: Relatório da missão Criosfera 2014**



# RELATÓRIO DE ATIVIDADES DA MISSÃO CRIOSFERA 1 (Dez 2014)

## PROANTAR-CNPq-MCTI

### Áreas de atuação durante a campanha:

1. Química da atmosfera
2. Meteorologia
3. Astrofísica de alta energia
4. Microbiologia polar



### RESUMO SOBRE O Criosfera 1

Operacional desde 2012, O Criosfera 1 é o resultado de 30 anos de pesquisa e aprendizado do programa Antártico Brasileiro/CNPq e de uma iniciativa única do Ministério de Ciência, Tecnologia e Inovação: a criação do INCT-Criosfera. Estas iniciativas, juntamente com o empenho particular da UERJ, UFRGS e INPE tornaram factível este desafio para a Ciência e a Engenharia nacionais. O Criosfera 1 é o primeiro módulo de pesquisa do Brasil a operar de forma autônoma na Antártica Central (Latitude 84°S). Sua concepção estrutural é o estado da arte em módulos de pesquisa polar, sendo dotado de sistemas eólico e solar que permitem mantê-lo em funcionamento ininterrupto durante as estações de verão e inverno. Seu conceito inicial visa o estudo da meteorologia e química da atmosfera antártica e sua relação com o manto de gelo. Neste contexto são (atualmente) monitorados aerossóis de origem terrígena, antropogênica, cósmica e biogênica (ex: Al, Ca, Ti, Si, Fe, S, Mg, Mn, As, K, U, Pb, Cd, Cu, Zn, Sr, Cr, V, Na, Rb, <sup>7</sup>Be, <sup>22</sup>Na), íons marinhos entre os quais Cl<sup>-</sup>, Na<sup>+</sup>, Mg<sup>2+</sup>, SO<sub>4</sub><sup>2-</sup>, Ca<sup>2+</sup>, K<sup>+</sup>, H<sup>+</sup>, NH<sub>4</sub><sup>+</sup>, NO<sub>3</sub><sup>-</sup> e Br<sup>-</sup>; a variabilidade de deposição de neve sobre o manto de gelo, diversos compostos orgânicos voláteis e gases do efeito estufa, com ênfase para o CO<sub>2</sub>. Os dados dos sensores meteorológicos e geofísicos como também do desempenho das fontes de energia do módulo serão enviados via satélite para o Brasil em tempo quase-real. Estes dados reunidos e interpretados a luz dos modelos computacionais do clima permitirão aumentar nossa compreensão sobre o papel da Antártica sobre a América do Sul, o impacto em larga escala da redução da camada de ozônio, a atividade vulcânica, a evolução dos processos globais de desertificação, o transporte atmosférico de poluentes e microorganismos ao redor do globo e aprofundar nosso conhecimento sobre a história climática contada pelos testemunhos de gelo. **A partir de 2014 torna-se efetivamente uma plataforma de pesquisa multi-usuária com estudos na área de astrofísica de alta energia (raios cósmicos) e microbiologia polar com ensaios *in situ*.**

## Principais objetivos da missão:

1. Avaliar as condições físicas e operacionais gerais do módulo de pesquisa polar avançado Criosfera 1, localizado na latitude 84°S, quanto a aquisição de dados meteorológicos, dados geofísicos e da química da atmosfera, após sua terceira etapa anual de funcionamento autônomo;
2. Instalar 4 novas turbinas eólicas do tipo eixo vertical, como substitutas das versões com eixo horizontal, que apresentaram mau desempenho nas campanhas anteriores;
3. Construir um pequeno bunker (~2 m abaixo da superfície) para amostrar, de forma estratificada, grandes quantidades de neve recente visando uma melhor compreensão do caráter sazonal das razões isotópicas de  $\delta^{18}\text{O}$  e  $\delta\text{D}$  e dos íons  $\text{Na}^+$ ,  $\text{Mg}^{2+}$ ,  $\text{SO}_4^{2-}$ ,  $\text{Ca}^{2+}$ ,  $\text{K}^+$ , MSA,  $\text{NH}_4^+$ ,  $\text{NO}_3^-$  e Br<sup>-</sup>. Adicionalmente amostrar neve recente para fins de estudo microbiológico, composição elementar, análise de nutrientes e da razão dos isótopos radiogênicos de Sr/Nd;
4. Instalar um detector de raios cósmicos para *múons*, experimento **CRE@AT** (Cosmic Ray Experiment in Antarctica) um consórcio entre o CBPF, Uerj e INCT-Criosfera;
5. Reativar o sistema de coleta de aerossóis atmosféricos e o analisador de  $\text{CO}_2$  que foram interrompidos em maio de 2014 em função de problemas com as turbinas eólicas;
6. Instalar 2 sistemas de coleta de aerossóis que amostram em 9 diferentes frações granulométricas, visando análise de composição elementar;
7. Realizar amostragem de Compostos Orgânicos Voláteis (COVs) a partir de canisters a vácuo;

## ATIVIDADES

**Translado até a latitude 84°S:** Todo o pessoal e parte de nossa carga (aprox. 700 kg) foram transportados de Punta Arenas ao acampamento da ALE (Antarctic Logistics and Expeditions), localizado na geleira Union, latitude 80°S, no dia 2 de Dezembro de 2014, por meio de um avião cargueiro Ilyushin (IL-76) (Figura 1).



Figura 1 – Chegada da equipe a geleira Union em Ilyushin (IL-76) fretado pela ALE.

Outros aprox. 700 kg já haviam sido transportados diretamente para o Criosfera 1 através de um comboio com trator polar até o Polo Sul no mês de Novembro, a partir de um contrato com a empresa ALE. Permanecemos no acampamento da ALE até o dia 5 de Dezembro em virtude das más condições de visibilidade nas proximidades do Criosfera 1. A ALE sugeriu dividir nossa carga em 2 partes de forma a transportar a equipe de Twin-Otter, uma aeronave de menor porte. Assim, cerca de 300 kg da carga de trabalho e comida permaneceram na ALE para que fosse embarcada num segundo voo para o Criosfera 1. Tal fato ocorreu em 9 de Dezembro. Solicitamos a ALE, por precaução, 1 botijão de gás na carga do Criosfera 1.

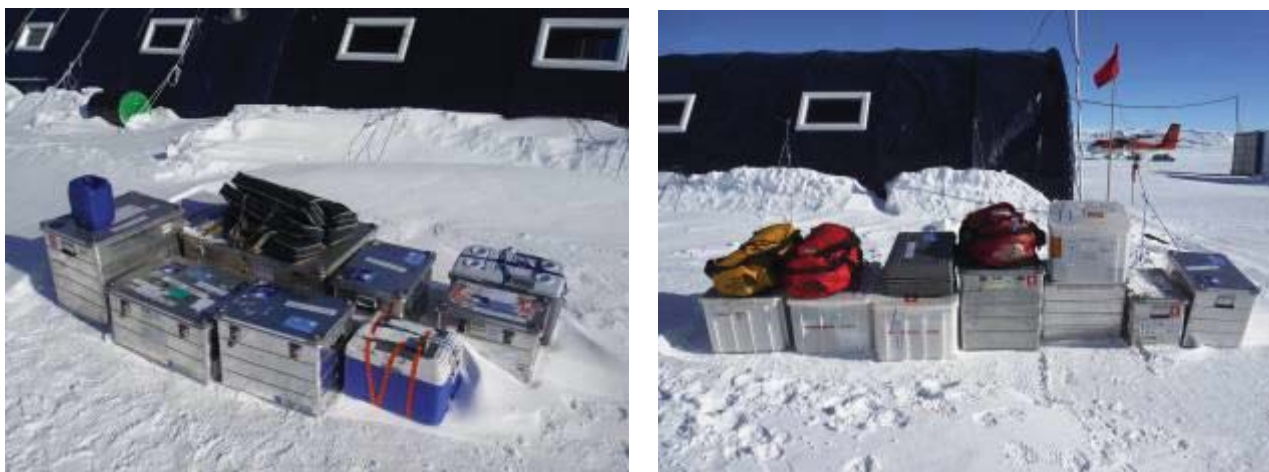


Figura 2 – (Esquerda) parte da carga transportada pelos pesquisadores diretamente para o Criosfera 1; (direita) carga transportada posteriormente ao Criosfera 1 pela ALE.



Figura 3 – Pouso de Twin-Otter na nova pista do Criosfera 1.

O vôo ao Criosfera 1 ocorreu em 5 de Dezembro de 2014, na parte da manhã, por meio de um avião Twin-Otter. O pouso ocorreu na nova pista sobre a neve na direção da face Norte do módulo, a jusante do fluxo de vento predominante na região (sentido de Sul para Norte). A pista de pouso foi previamente feita por tratores da ALE que passaram pelo local no início da temporada de verão e já consta com sinalização a cada 100 m. A aeronave estacionou a alguns metros do Criosfera 1 para facilitar o transporte de carga. Não observamos qualquer entrada de neve dentro do laboratório referente ao inverno de 2014. A deriva da neve ao redor do módulo foi considerável, observando-se maior incremento de neve na face sul do módulo, atingindo praticamente o nível do piso do Criosfera 1.

Durante as primeiras 48 horas que seguiram a chegada, foram montadas 2 barracas tipo Endurance/"Weatherhaven" e 1 snow-sledge (pirâmide); foram organizadas as caixas de comida, a cozinha, o banheiro e iniciou-se a organização do laboratório. Foi programado, e executado, contato diário, por sistema telefônico satelital iridium, com a ALE na geleira Union, diariamente às 19:40 h (horário de Punta Arenas/Chile). A ESANTAR equivocadamente nos cedeu uma Jabuti pequena ao invés do modelo estendido solicitado. Assim 3 pesquisadores dormiram na Jabuti e 2 no módulo Criosfera 1 por falta de espaço para dormitório. Para a campanha de 2014, os pesquisadores compraram camas individuais portáteis o que ofertou melhor qualidade de sono.

#### **Observação inicial das condições operacionais no Criosfera 1 durante a chegada:**

Um aerogerador foi deixado operacional no Criosfera 1 em Janeiro de 2014. Durante 4 meses o aerogerador se manteve funcionando. Aparentemente devido aos fortes e constantes ventos no inverno, o mesmo foi danificado e parou de gerar energia. Desta forma, as baterias de 24 V perderam carga levando a interrupção do sistema de amostragem de CO<sub>2</sub> e o sistema de coleta de aerossóis. O sistema de meteorologia, que durante o inverno opera com as baterias de 12 V, operou continuamente e de forma ininterrupta, incluindo-se o sensor de deposição de neve.

**Síntese da monitoração meteorológica no Criosfera 1 durante 2014:** É apresentado abaixo uma síntese dos dados meteorológicos básicos obtidos no Criosfera 1 durante a campanha 2014.

**Temperatura do ar, umidade relativa, velocidade do vento, radiação solar global e pressão atmosférica:** Temperatura do ar, umidade relativa, velocidade do vento e radiação solar global covariaram ao longo do ano. A radiação solar é um forte modulador dos 3 parâmetros meteorológicos. Durante os meses de Janeiro e Fevereiro e Novembro e Dezembro do verão austral, os 3 parâmetros apresentaram menor variabilidade e padrão de variabilidade semelhante a radiação solar global. Durante o inverno quando os níveis de radiação solar chegam a zero, observa-se maior variabilidade dos 3 parâmetros. Durante o inverno a temperatura do ar é basicamente governada pela advecção de frentes de latitudes mais baixas que transportam calor para o Platô Antártico. Durante estes episódios observamos aumento da umidade relativa o que retrata perfeitamente tal situação, pois as frentes que migram em torno da borda continental antártica tem alta influência marítima. A velocidade do vento tem padrão sazonal, mas sua variabilidade é mais acentuada provavelmente por estar relacionada não somente a advecção das massas de ar, mas também pela influência de um forte "vento catabático" que sopra do Pólo Sul em direção a base do Mar de Weddell, Figura 4.

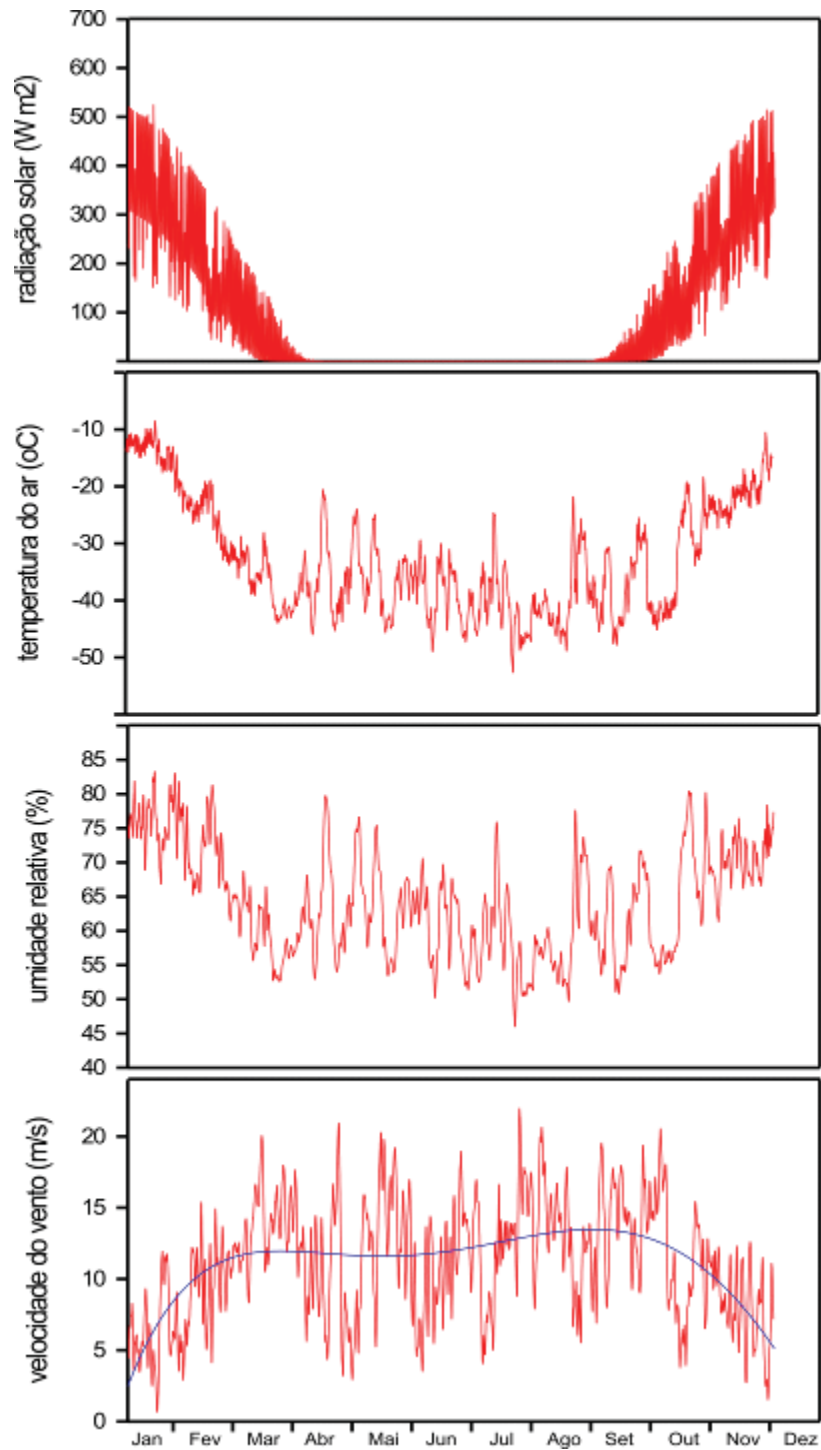
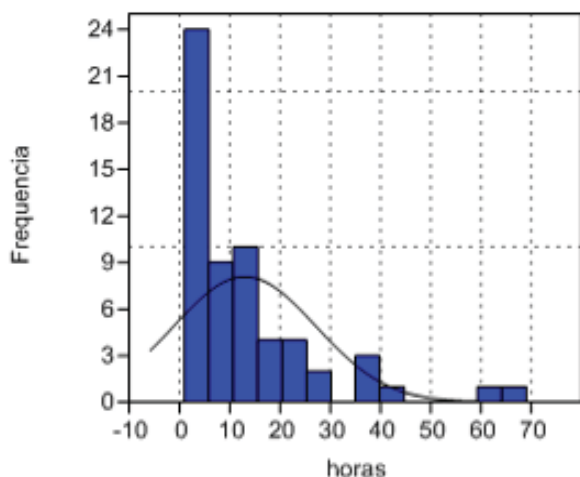


Figura 4 – Séries temporais para 2014 (velocidade do vento, umidade relativa, temperatura do ar e radiação solar global).

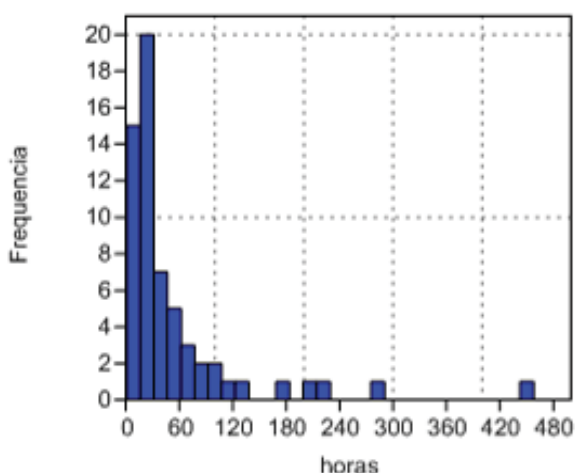
Em relação a duração de períodos com calmaria, aqui definido como períodos com ventos inferiores a  $5 \text{ ms}^{-1}$  observou-se duração anual de  $\sim 13$  horas, com máximo na ordem de 3 dias, Figura 5.



Média	12,8
Erro padrão	1,83
Mediana	10
Desvio padrão	14,2
Variância	202
Curtose	5,23
Assimetria	2,14
Duração Mínima	1
Duração Máxima	69
Eventos de Calmaria	60

Figura 5 – Estatística descritiva dos eventos de calma no Criosfera 1 (velocidade do vento <math>< 5 \text{ ms}^{-1}</math>).

Em relação ao uso dos aerogeradores Forgen que iniciam, efetivamente, a geração de energia a partir de ventos na ordem de  $10 \text{ ms}^{-1}$ , os dados do Criosfera 1, Figura 6, mostram que episódios com velocidade do vento inferiores a este patamar ocorrem com duração média de  $\sim 2,5$  dias, podendo chegar a quase 20 dias. Tais eventos representam cerca de 39% do período total observado aqui. Estes dados foram utilizados como referência



Média	53,9
Erro padrão	9,75
Mediana	28
Desvio padrão	76,1
Variância	5798
Curtose	13,8
Assimetria	3,37
Duração Mínima	1
Duração Máxima	458
Eventos com $vv < 10 \text{ ms}^{-1}$	61

Figura 6 – Estatística descritiva dos eventos de calma no Criosfera 1 (velocidade do vento <math>< 5 \text{ ms}^{-1}</math>).

A pressão atmosférica não apresentou um marcante padrão sazonal, mostrando uma suave tendência de declínio em direção ao inverno, Figura 7.

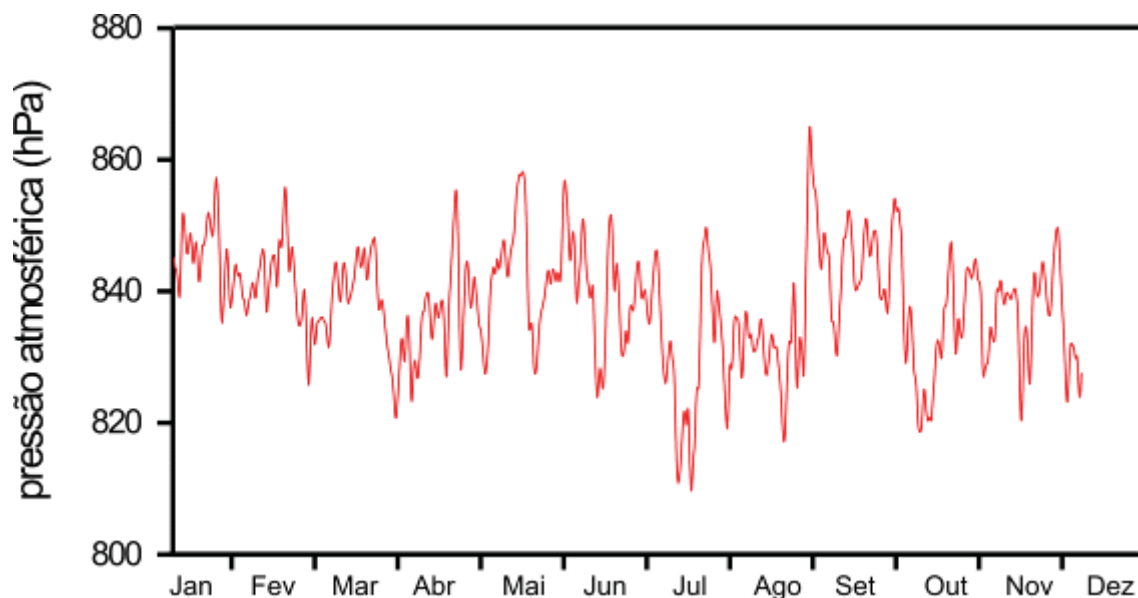


Figura 7. Pressão atmosférica em 2014 no Criosfera 1.

**Direção do vento:** Tal como observado nas campanhas anteriores, em 2014 a direção predominante do vento foi do Setor Sul, Figura 8. Entretanto importantes intrusões de massas de ar nos Setores SE-E-NE são observadas ao longo do ano, o que mostra uma importante influência das massas do Weddell sobre a região advectando através de "Dronning Maund Land". Eventos relacionados à influência dos Mares de Amundsen e Bellingshousen mostraram-se raros, ocorrendo basicamente no período de verão.

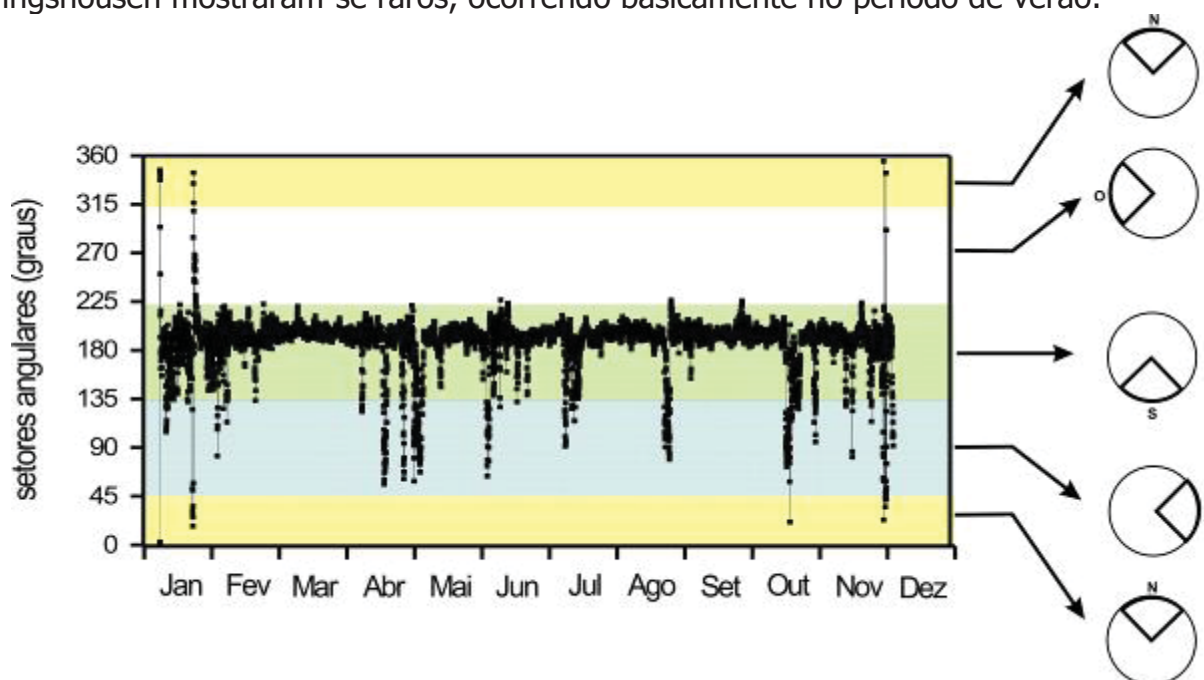


Figura 8 – Série temporal de direção do vento em 2014 no Criosfera 1. Círculos à direita ilustram os setores angulares em torno dos pontos cardinais.

Uma análise ilustrativa da questão é apresentada na Figura 9. O Criosfera 1 está localizado numa região de alicve que se inicia na base do Mar de Weddell e se estende até o platô antártico, estando numa altitude de aproximadamente 1270 m. Sua localização na latitude 84°S e 079°W faz com que fique situado na porção de menor elevação das Montanhas

Trans-Antártica, recebendo influência direta do platô antártico. Desta forma, a orografia atua como agente facilitador para a quase-permanente ação dos ventos do Setor Sul (Figura 8B). Os eventos do Setor Leste são interpretados como advecções originadas nas proximidades do Mar de Weddell como já mencionado (Figura 9A). É muito provável que tais advecções para o Criosfera 1 também sejam favorecidas pela orografia do próprio platô antártico cuja isolinha de 2500-3000 m é orientada no sentido Leste-Oeste em relação ao Criosfera.

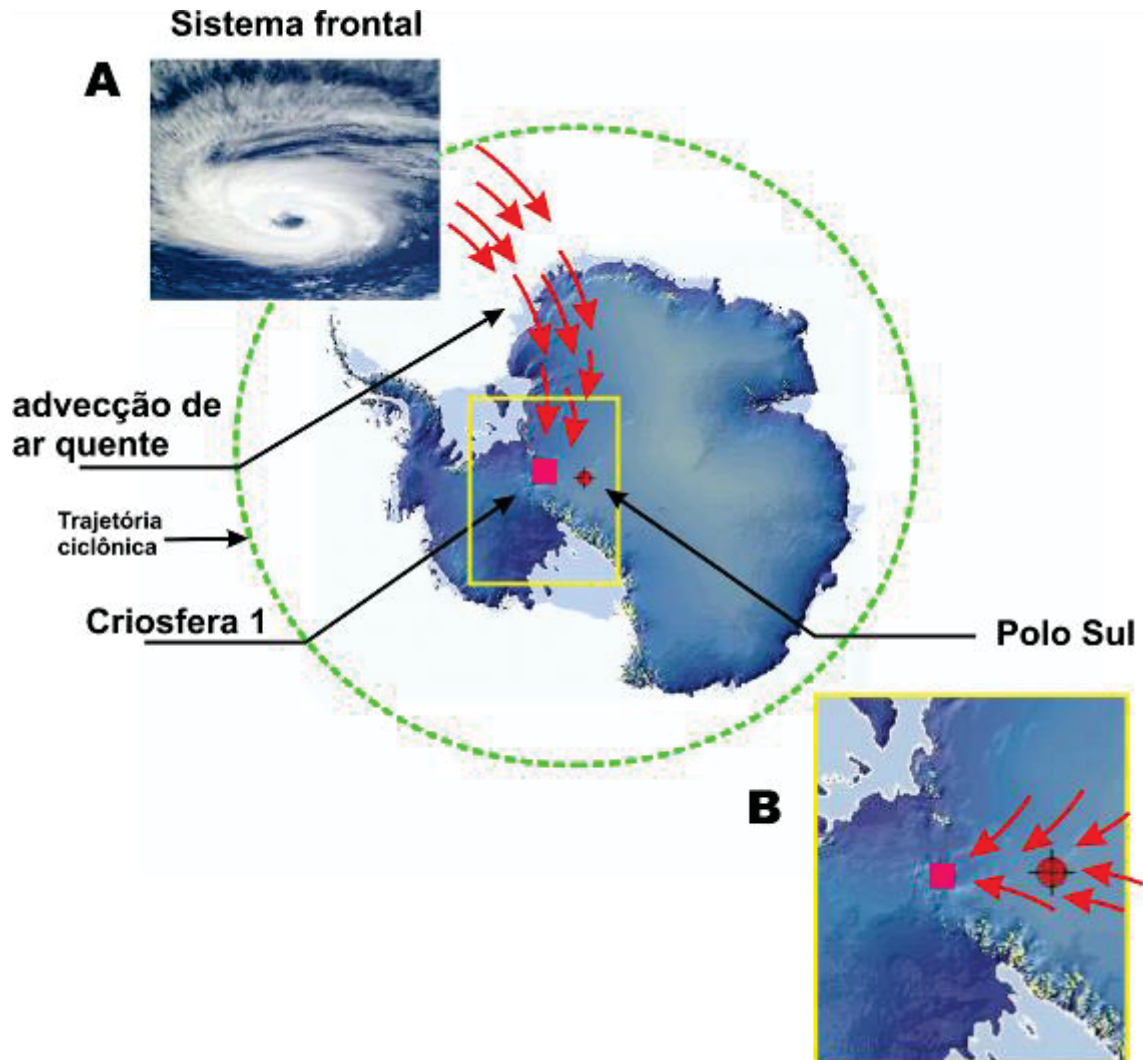


Figura 9 – Modelo conceitual para os principais sistema atmosféricos que atuam sobre o Criosfera 1: (A) Sistema de advecção quente proveniente do Mar de Weddell que adentra o continente Antártico; (B) Sistema de ventos que se origina no platô antártico e que flui em direção ao Mar de Weddell atravessando a região onde está instalado o Criosfera 1.

É possível observar que os eventos mais significativos de aumento de temperatura do ar e umidade relativa registrados no Criosfera 1 estão relacionados a mudança de direção do vento do padrão Sul para os Setores SE-E-NE tal como realçado na Figura 10.



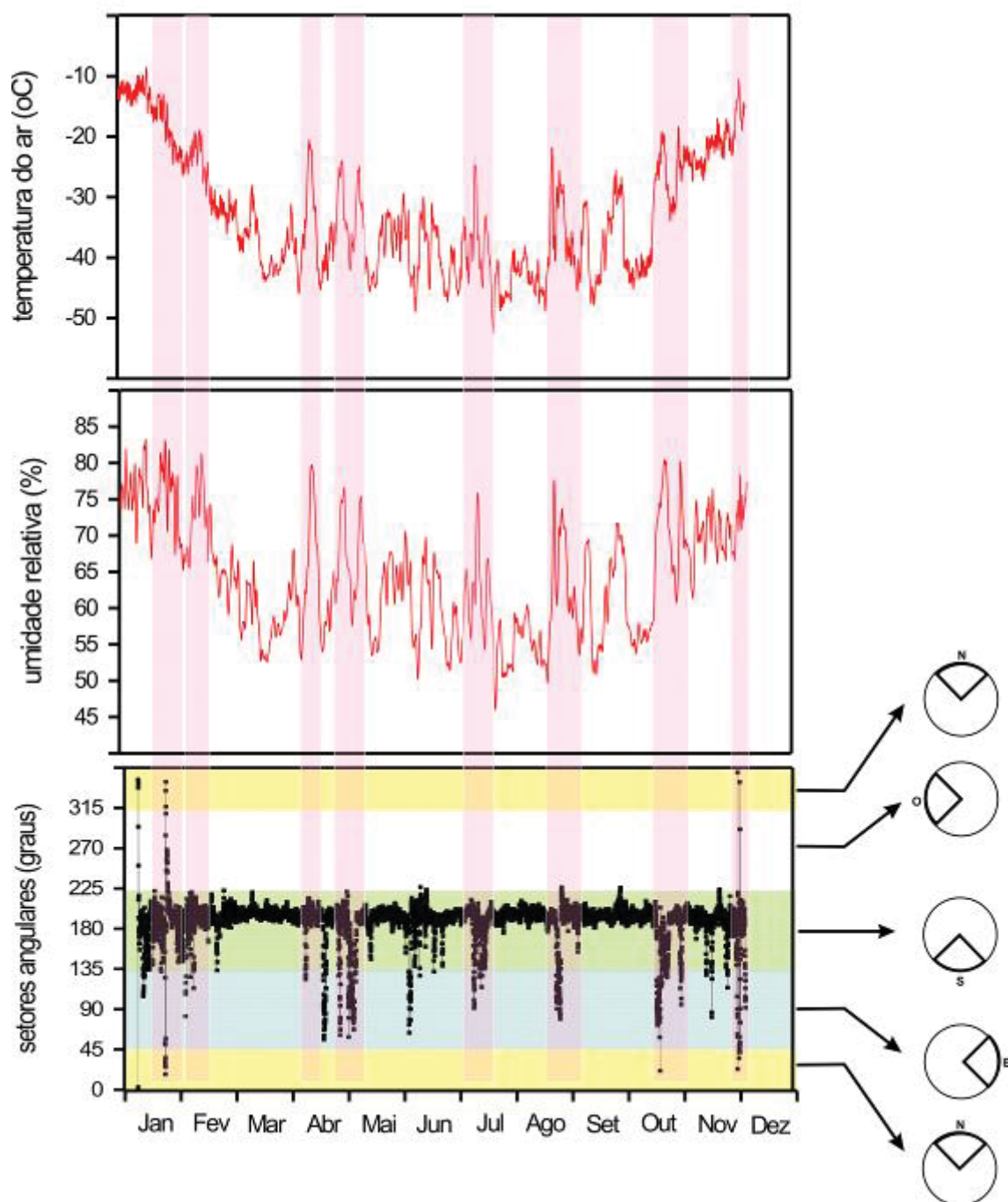


Figura 10 – Séries de temperatura do ar, umidade relativa e direção do vento para o ano de 2014 no Criosfera 1.

**Acumulação de neve:** Para a campanha 2014, o sensor ultrasônico de acumulação de neve (SR50A produzido pela Campbell Scientific, Inc.) operou normalmente durante todo o ano. Entretanto, para a validação dos dados adotamos o critério de qualidade de sinal sônico sugerido pelo fabricante em conformidade com a Tabela 1. Desta forma todos os dados fora do intervalo 152-210 foram descartados para se obter uma série temporal compilada de maior confiabilidade. Não foi observado um padrão sazonal marcante para a precipitação de neve no Criosfera 1 em 2014.

Tabela 1 – Nível de qualidade de sinal para o sensor sônico SR50A utilizado no Criosfera 1.

Intervalo do Nível de Qualidade do sinal sônico	Descrição
0	Sistema não é capaz de ler o sinal
152-210	Mensuração aceitável
210-300	Baixa intensidade de sinal sônico
300 to 600	Alta incerteza na medida

Durante o ano de 2014, os dados do sensor ultrasônico de acumulação de neve apresentaram alta variabilidade entre 0 (zero) e 1 m. Entretanto, a acumulação líquida observada no início de Dezembro de 2014 foi praticamente nula, sugerindo uma elevada dinâmica no processo de precipitação, acumulação e remoção da neve recente. Nossa observação de campo mostra que o "snow drift" na superfície ocorre para ventos acima de  $\sim 12 \text{ ms}^{-1}$ . Comparando-se as séries temporais de precipitação de neve e direção do vento observa-se claramente que ocorre aumento da precipitação na ocasião da mudança de direção do vento para o sentido SE-E-NE. A partir desta associação, pode-se reafirmar que a deposição de neve no Criosfera 1 é baseada em 2 processos: ventos quentes (decorrentes de sistemas ciclônicos da região sub-Antártica) trazem umidade do Setor Mar de Weddell-Dronning Maund Land aumentando a precipitação de neve na região. Após a passagem de tais sistemas, os fortes ventos que sopram do platô antártico removem a neve recém-depositada, tornando este balanço próximo de zero.

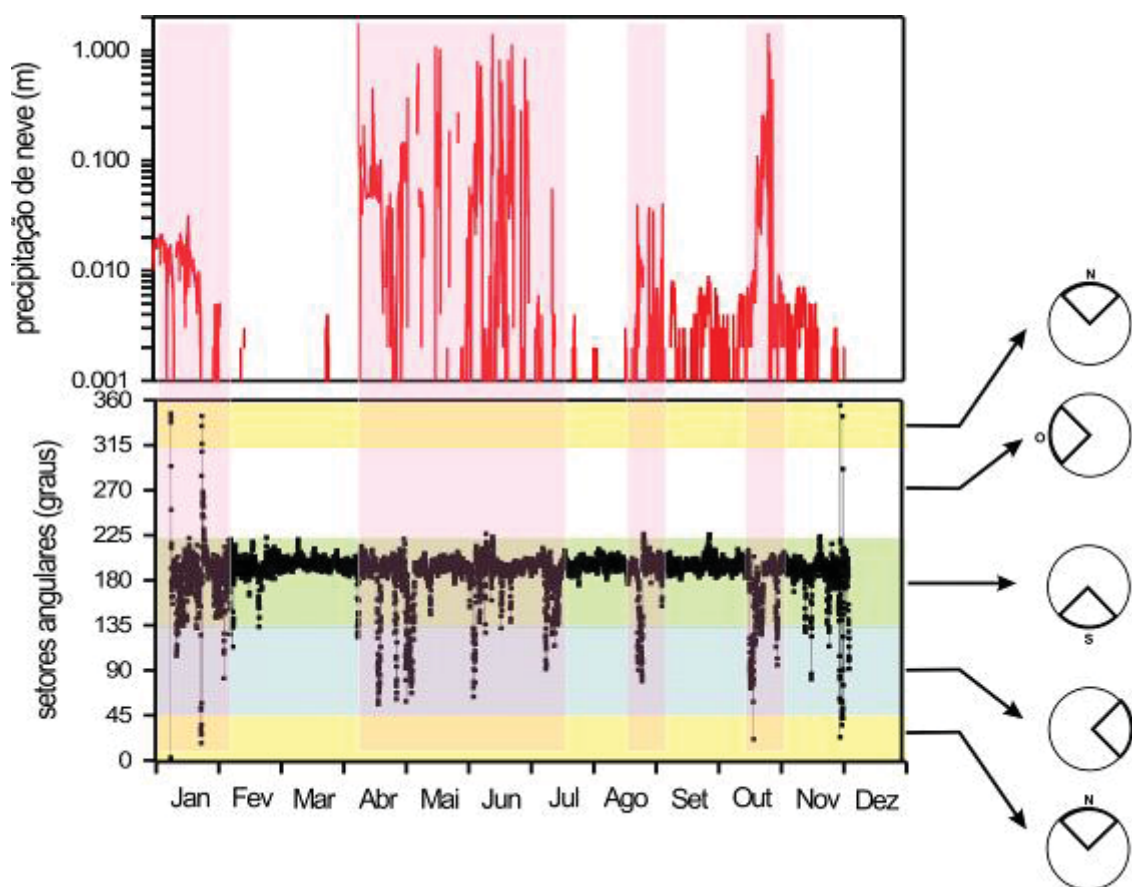
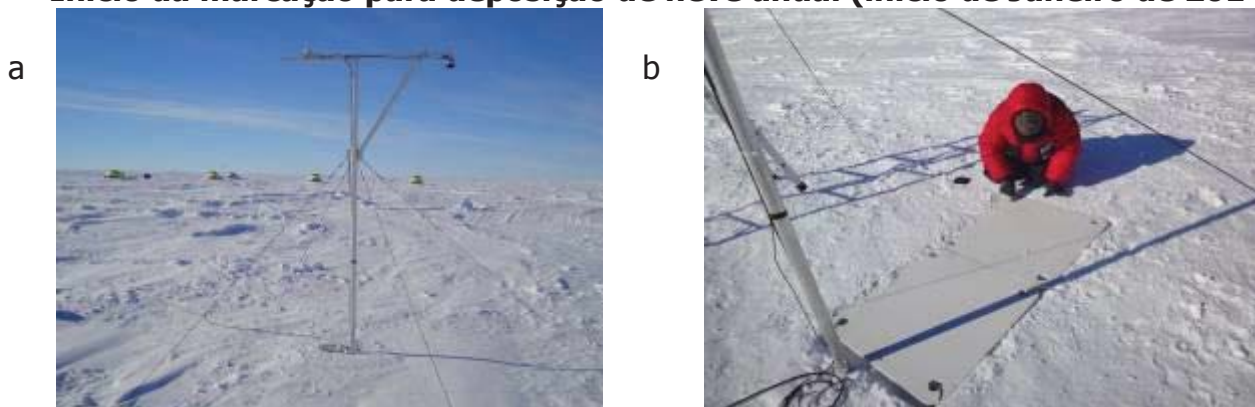


Figura 11 – Séries de temperatura de precipitação de neve e direção do vento para 2014.

O dado de altura de acumulação de neve na tábua de referência anual instalada diretamente abaixo do sensor sônico (Figura 12) apresentou acumulação praticamente nula, corroborando o dado de início de Dezembro do sensor sônico.

### Início da marcação para deposição de neve anual (início de Janeiro de 2014)



### Observação da marcação de deposição de neve em Dezembro de 2014



Figura 12 – (acima) Colocação da placa para acúmulo anual de neve sob o sensor SR50A; (abaixo) acúmulo anual de neve em Dezembro de 2014.

**Análise dos dados de CO<sub>2</sub> para 2014 e perspectivas para 2015:** O sensor "Carbon Dioxide Probe GMP343"/Vaisala operou nos meses de Janeiro, Fevereiro, Março e parte de Abril de 2014, em virtude da ausência de sistemas eólicos no Criosfera 1 após estes meses. Uma comparação entre os dados de CO<sub>2</sub>, temperatura e umidade mostra excelente covariabilidade (Figura 13). Como os dados de direção do vento mostram forte associação com os aumentos de temperatura e umidade na região, pode-se sugerir que as mesmas massas de ar que trazem calor, umidade e precipitação também transportam o CO<sub>2</sub> para o interior do continente antártico. A variabilidade destes parâmetros é concomitante.

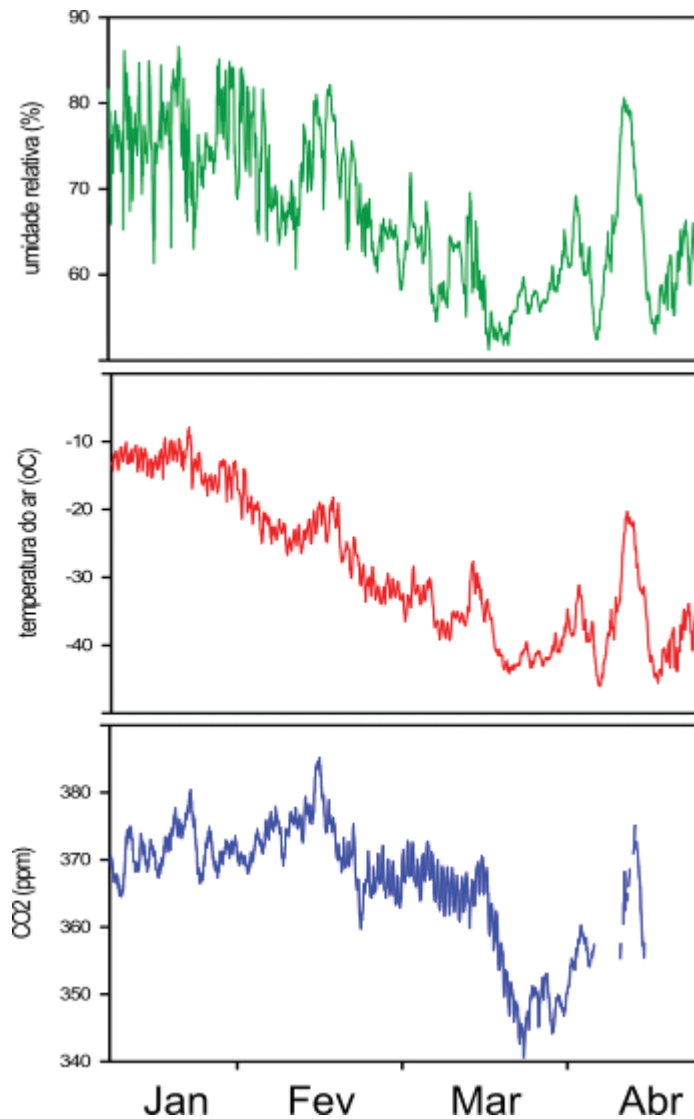


Figura 13 – Série temporal de CO<sub>2</sub>(não calibrado), temperatura do ar e umidade para 2014 no Criosfera 1.

Em decorrência dos problemas com as novas turbinas Forgen instaladas na campanha 2014, o novo sistema de baterias com 8 novas unidades (trazido na atual campanha), será de grande relevância para a manutenção dos sensores e sistemas de amostragem no inverno 2015. Desta forma, o sistema de CO<sub>2</sub> deverá operar com 4 medições diárias no inverno de 2015.

**Sistema de coleta de aerossóis:** Para a campanha de 2014, três novos amostradores foram instalados na torre meteorológica do Criosfera 1. Os três sistemas visam a caracterização granulométrica do material particulado que aporta na região do Criosfera 1. Eles são um impactador em cascata da PIXE com 9 estágios (apenas 8 foram utilizados, pois 2 foi danificado), que opera com vazão de 1 L/min, um impactador tipo "May Impactor" que opera com vazão de 8 L/min e um impactador tipo "Bertelli" que opera com vazão de 1 L/min. Todos utilizaram bomba de amostragem marca KNF de 12 L/min com regulador de vazão. As Figuras 14 e 15 mostram detalhes dos 3 sistemas. Os impactadores PIXE e May iniciaram operação a partir de 17 de Dezembro e o impactador Bertelli em 19 do mesmo mês. Os 3 sistemas integrarão uma amostragem durante parte da estadia dos pesquisadores

no Criosfera 1, o que corresponderá a aproximadamente 10 dias. O impactador Bertelli retornará ao Brasil com a amostra. Os integradores PIXE e May receberão novos filtros e lâminas (respectivamente) para integrarem o período de inverno. Pelo mesmo motivo que as mensurações de CO<sub>2</sub> no inverno 2015 só ocorrerão 4 vezes ao dia, pela limitação de energia no módulo, as amostragens de material particulado com os 2 impactadores em cascata ocorrerão da mesma forma, ou seja, a cada 3 horas (durante 10 minutos) ocorrerá amostragem. As técnicas analíticas deste projeto estão sob a supervisão da UFPR e UERJ.

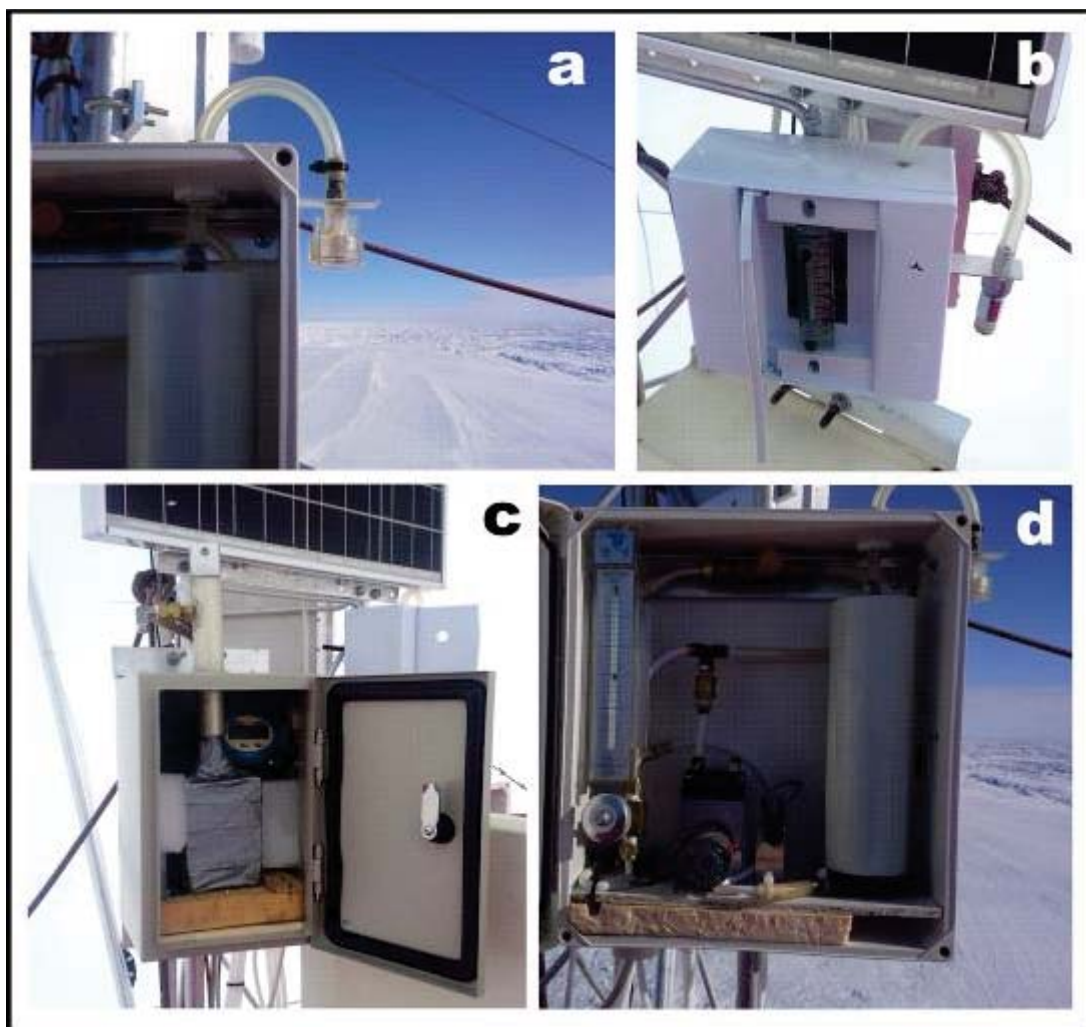


Figura 14 – (a) “inlet” do impactador PIXE-8-estágios; (b) impactador Bertelli; (c) impactador “May”; (d) sistema de amostragem do impactador PIXE.



Figura 15 – Vista geral dos sistemas de amostragem instalados no teto do Criosfera 1.

**Sistema de coleta de neve/firn:** Durante a campanha de 2014, foi construído um pit de 2 m objetivando a amostragem estratificada de neve. Foram coletadas 3 tipos de amostras: (1) amostras para a análise de  $\delta^{18}\text{O}$ , a cada 5 cm, visando a observação do sinal de variabilidade interanual nas estratificações da neve e firn; (2) amostras para análise iônica (cátions e ânions) estes representantes da variabilidade do gelo marinho (íons de Na, Cl e Mg) produtividade primária ao redor da Antártica (MSA) ou do aporte mineral nssCa (fração não marinha do Cálcio) e nssK (fração não marinha do Potássio). Além destas análises serão mensurados as concentrações de metais de origens terrestre, concentrados tecnologicamente e de diversas outras origens; (3) amostras para a razão de isótopos radiogênicos de Sr e Nd visando identificar o aporte atmosférico em escala continental para a região do Criosfera 1. Tais isótopos estão fortemente associados a sua matriz geológica de origem de uma comparação de nossos dados com inventários destes isótopos para a América do Sul, África e Austrália, será possível investigar com maior precisão a origem das massas de ar. Como a detectabilidade destes isótopos é algo complexo na matriz gelo, coletamos 8 amostras a cada estrato de 50 cm. As técnicas a serem utilizadas nesta análise serão a espectrometria e a cromatografia líquida. As instituições envolvidas na análise são a UERJ (análise de isótopos radiogênicos de Sr/Nd), UFRGS (isótopos e íons) e University of Maine (metais).



Figura 16 – Amostras do pacote anual de neve visando análises glacioquímicas.

**Instalação do novo conjunto de geradores eólicos:** Em decorrência do fracasso no uso de aerogeradores do tipo Air Breeze, de eixo horizontal, durante a campanha de 2014, foram trazidas ao Criosfera 4 novas turbinas eólicas. A marca Forgen foi sugerida pela ALE que utiliza unidades de 15 Watts para suas estações meteorológicas. Optou-se pela compra do modelo Ventus 70, cuja expectativa era gerar, em cada unidade, cerca de 35 Watts considerando-se um vento médio de inverno de 12 a 13  $\text{ms}^{-1}$ . Os resultados dos testes das turbinas tanto em bancada no Criosfera 1 como em condições de ventos fracos, moderados e fortes após sua instalação foram absolutamente frustrantes. Na melhor das hipóteses, as turbinas geraram no máximo 5 Watts, em condições de vento de 11 a 12  $\text{ms}^{-1}$ . Uma das quatro turbinas foi entregue inoperante e nenhuma das turbinas acompanhou sua curva de desempenho (velocidade do vento x potência gerada) dada pelo fabricante. Este resultado de 5 Watts representa uma melhoria do desempenho das turbinas Forgen após os engenheiros do Criosfera 1 realizarem testes eletrônicos e até alterações mecânicas tal como o encurtamento da distância entre os ímãs e os induzidos. De um modo geral, inspecionando-se internamente as turbinas, se verificou visíveis precariedades na montagem e até na qualidade dos componentes utilizados. Após várias discussões técnicas decidiu-se em manter 3 das quatro turbinas eólicas instaladas e operacionais com o intuito de manter minimamente as tensões nas baterias e conseqüentemente evitar maior dano ao novo banco de baterias. Ao longo do ano de 2015 será necessário reavaliar a questão do sistema eólico e provavelmente o uso de um terceiro sistema gerador. O aprendizado desta questão é que o sistema eólico que funcionará definitivamente no Criosfera 1 só deverá ser levado a Antártica após exaustivos testes de performance.



Figura 17 – Teste e montagem das turbinas eólicas Forgen Ventus 70 no Criosfera 1.

**Elevação do módulo Criosfera 1:** O Criosfera 1 foi projetado de forma que seus pés possam ser elevados em função da deposição de neve, mantendo-o na mesma coordenada acima do manto de gelo. De forma auxiliar há ainda um conjunto de 4 pés de apoio construído no INPE para o projeto, localizados ao lado dos pés originais que têm a função de elevar o módulo a partir do uso de macacos individuais. Desta forma, ao elevar os pés auxiliares, os pés originais, que possuem prolongamentos embutidos, podem ser fixados em diferentes alturas de forma manual. Pela primeira vez, durante a campanha de 2014, tal sistema foi utilizado na Antártica e o módulo pode ser elevado em 40 cm. Esta elevação garantirá o módulo acima do manto de gelo por mais aproximadamente 3 anos. O Criosfera 1, com seu sistema de pés, faz com que a neve, que sopra em alta velocidade próximo a superfície, ao encontrar o módulo escoe por abaixo de sua estrutura e desta forma promove baixa acumulação ao seu redor. A elevação do módulo em 2014 também sinalizou para a necessidade de pequenos aperfeiçoamentos no método de elevação e uso de materiais específicos para tal tarefa.





Figura 18 – (esquerda) colocação de macaco em pé auxiliar visando elevação do Criosfera 1 (40 cm); (direita) estruturas de elevação do módulo.

**Instalação do compartimento de Raios Cósmitos no Criosfera 1:** O continente antártico, principalmente a Antártica Central, é um dos melhores ambientes do globo para a observação astronômica e astrofísica. Aproveitando-se desta vantagem, o Criosfera 1 inaugurou em 2014 o primeiro compartimento do **CRE@AT** (Cosmic Ray Experiment in Antarctica), um consórcio entre o CBPF, a UERJ e o INCT-Criosfera. A idéia central é de ao longo dos próximos anos ter instalado o primeiro espectrômetro de raios cósmitos do Brasil na Antártica Central. Com isso o Criosfera 1 passa a monitorar em tempo real o fluxo de múons na Antártica Central. Objetiva-se para as próximas campanhas a ampliação do espectrômetro de raios cósmitos já instalado e a correlação de seus dados com um sensor de cobertura de nuvens para a região, já que raios cósmitos atuam como fonte geradora de núcleo de condensação que formam nuvens. Além disso, objetiva-se tornar o Criosfera 1 como parte do Global Cosmic Ray Network.

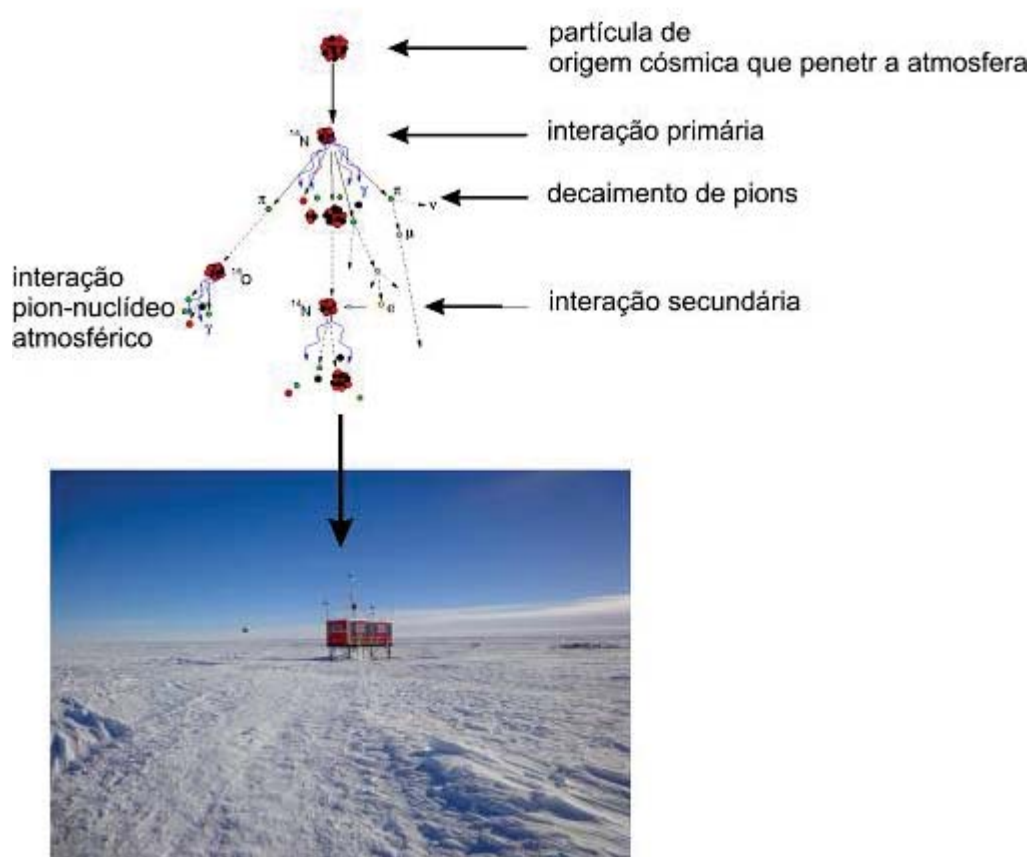
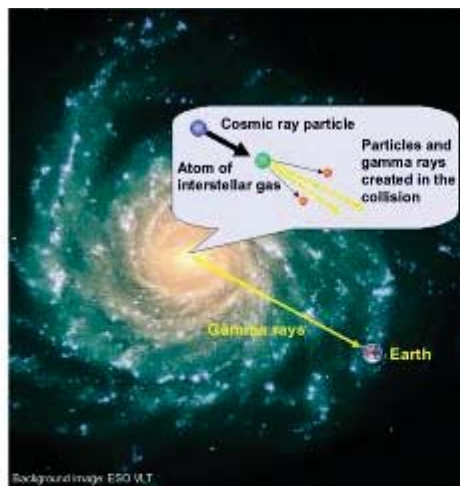


Figura 19 – Diagrama ilustrativo do processo de formação de sub-partículas atômicos durante interação dos raios cósmicos com a atmosfera terrestre.

Na Figura 20, é ilustrado parte do sistema eletrônico de detecção de múons instalado no Criosfera 1 em 2014.



Figura 20 – Eletrônica de detecção e sinal para múons no Criosfera 1.

Na Figura 21, é ilustrado o diagrama de blocos básico do sistema de detecção de múons no Criosfera 1.

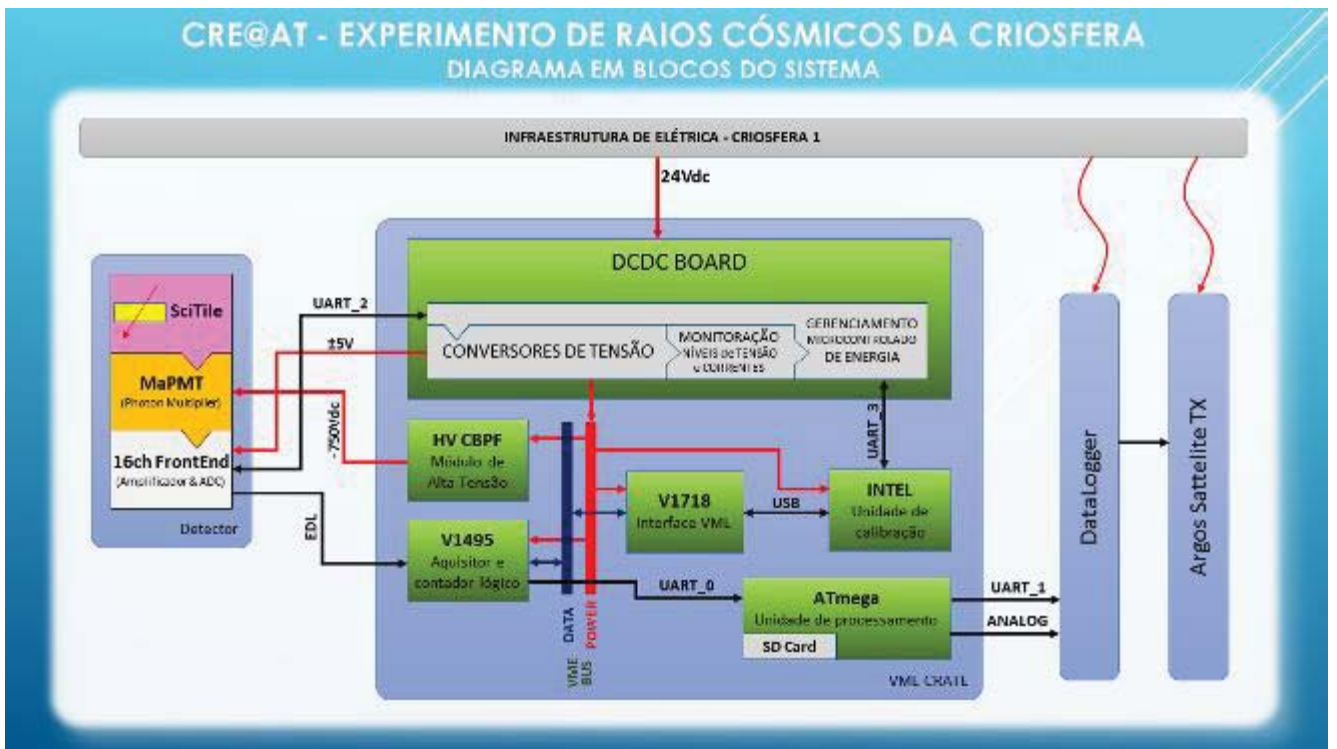


Figura 21 – Diagrama de blocos do sistema de detecção de múons e aquisição de dados no Criosfera 1.

**Instalação do compartimento CriLab 1 no Criosfera 1:** Em 2014, foi instalado dentro do Criosfera 1 um compartimento para atender demandas mínimas analíticas e de pré-concentração na área de microbiologia polar.



Figura 22 – Infra-estrutura instalada para microbiologia polar no Criosfera 1 em 2014.

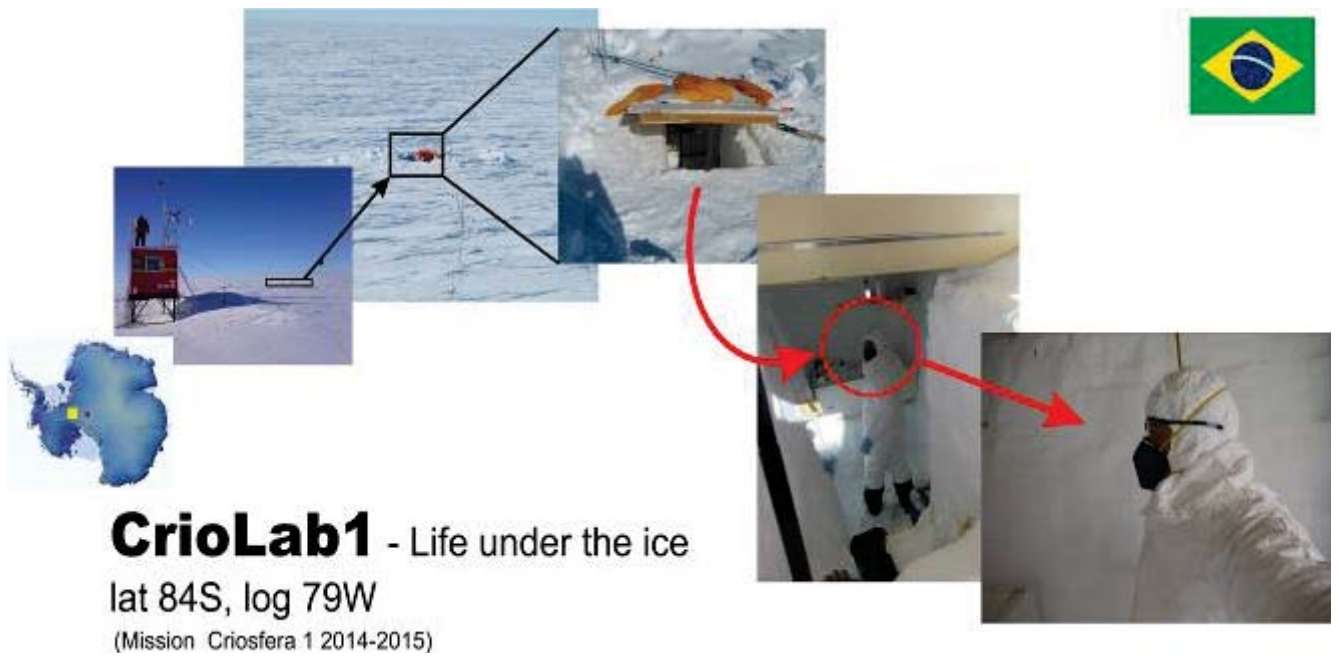


Figura 23 – Localização e características do CrioLab1, situado ao lado do Criosfera 1.

**Instalação do CrioLab no Criosfera 1:** O pit de 2 m de profundidade foi escavado com o objetivo de iniciarmos a montagem do CrioLab (Figura 23). A exposição das paredes do pit facilita a coleta de maiores volumes de amostra de um mesmo extrato. O objetivo é que o pit seja aumentado nas próximas missões tanto para as laterais (devido a coleta das amostras) como para baixo, aumentando em profundidade.

Dentro do Criosfera 1, foi instalado um compartimento para atender demandas mínimas para o processamento asséptico das amostras para análises microbiológicas (Figura 24). O canto leste do Criosfera 1 foi isolado por placas de isolantes. O chão e as paredes eram limpos com hipoclorito 2%, peróxido de hidrogênio 3% e álcool 70% todos os dias antes do início dos procedimentos. Não era permitida a circulação de pessoas e uso de botas na área isolada. Todos os procedimentos de manipulação das amostras foram realizados dentro do fluxo laminar, previamente esterilizado por luz ultravioleta, e dentro da área de segurança produzida pela chama de uma lamparina.



**Figura 24** - Sistema de filtragem instalado no Criosfera 1 para concentração de microorganismos e particulados da amostras de neve/firn coletadas no pit.

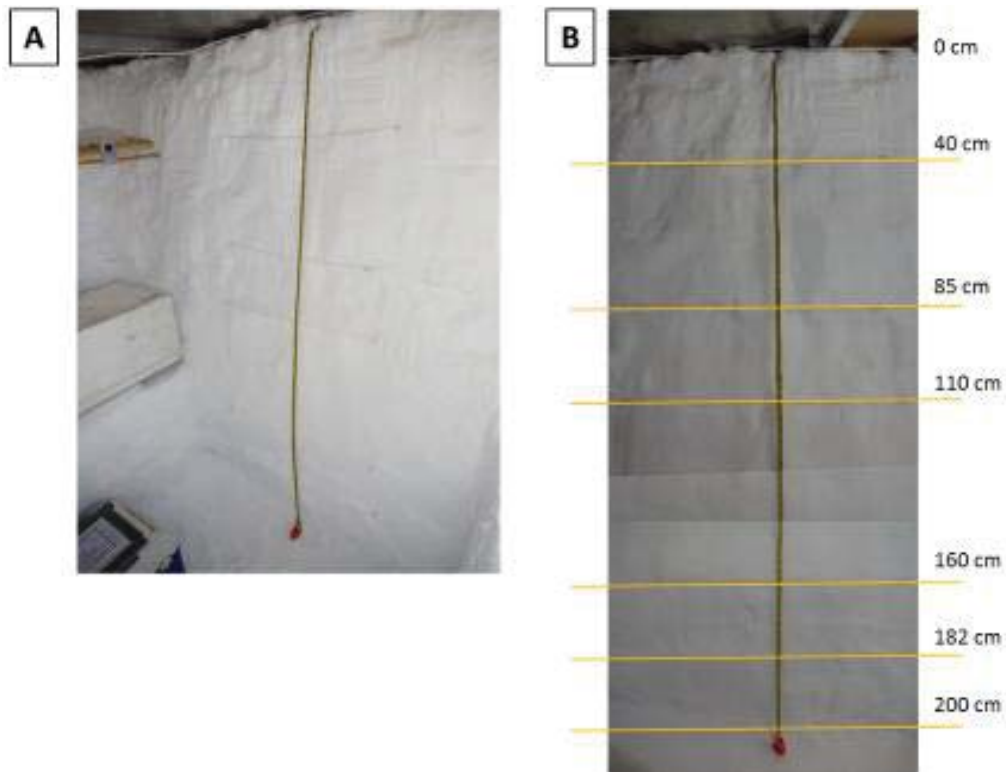
**Coletas e processamento de amostras de neve/firn para estudos microbiológicos:** O objetivo da coleta de material para estudos microbiológicos na Operação Criosfera 1 de número IV é de detecção, isolamento e caracterização de microorganismos que foram transportados para o interior do continente antártico.

Os procedimentos de coleta para estudos microbiológicos foram organizados em 4 etapas: (1) escavação do pit e coleta das amostras, (2) filtragem das amostras em membranas com poros de tamanho de 0.22  $\mu\text{m}$  e preparo de inóculos bacterianos, (3) filtragem das amostras em membranas com poros de tamanho de 0.1  $\mu\text{m}$ , e (4) coleta de amostra de neve/firn para serem transportadas congeladas para a USP e descrição das estratificações na parede do pit. A tabela 2 apresenta o período necessário para a conclusão de cada etapa.

**Tabela 2** - Período utilizado para a conclusão das etapas de coletas para estudos microbiológicos.

<b>Etapas</b>	<b>Período</b>
Etapa 1	de 6 a 14 de dezembro de 2014 (9 dias)
Etapa 2	de 15 a 22 de dezembro de 2014 (8 dias)
Etapa 3	de 18 a 24 de dezembro de 2014 (7 dias)
Etapa 4	25 de dezembro de 2014 (1 dia)

A escavação do pit foi essencial para a coleta de um volume grande (litros) de amostra de neve/firn de um mesmo estrato de deposição. A parede ao oeste do pit foi utilizada para a extração das amostras para análises microbiológicas. Seis estratificações principais (0-40 cm, 40-85 cm, 85-110 cm, 110-160 cm, 160-182 cm e 182-200 cm) foram observadas nos 200 cm de profundidade escavados (Figura 25A e B).



**Figura 25** - Parede oeste do pit com a marcação das estratificações principais. (A) Visualização geral dos estratos; (B) Esquema mostrando os estratos identificados e as profundidades de demarcação dos estratos.

Blocos de neve/firn foram removidos da parede com o uso de um serrote esterilizado com álcool 70% (Figura 26A). As seis faces do bloco coletado foram descontaminados pela remoção de 3 a 5 cm de neve/firn utilizando uma espátula de inóx esterilizada com álcool 70% (Figura 26B). As amostras foram armazenadas em sacos whirl-pak®, transferidas para o módulo Criosfera 1 e mantidos em coolers. As amostras foram derretidas a temperaturas entre 4 e 8°C para posteriormente serem filtradas. O volume de 36,186 L de amostra foi coletado em membranas de poliethersulfona de poros com tamanho de 0.22  $\mu\text{m}$  para extração de DNA e proteínas. O total de 8,246 L de amostra filtrada em 0.22  $\mu\text{m}$  foi filtrada novamente em membranas com poros de tamanho de 0.1  $\mu\text{m}$  para retenção de células e partíulas que compõe a fração entre 0.1 e 0.22  $\mu\text{m}$ . DNA total será extraído das membranas de 0.1  $\mu\text{m}$  para análise da composição da assembléia microbiana na neve/firn.



**Figura 26** - Coleta das amostras de neve/firn no CrioLab. (A) Bloco de neve/firn extraído da parede do pit; (B) Processo de descontaminação do bloco para armazenamento em sacos estéreis.

Em paralelo à coleta de células para extração de DNA e proteínas, foram preparadas membranas de 0.22  $\mu\text{m}$  de policarbonato para observações em microscópio eletrônico e de epifluorescência. Tivemos um contra-tempo com a bomba de vácuo para o procedimento de preparo das amostras para microscopia no Criosfera 1. Sendo assim, as amostras foram fixadas com glutaraldeído (concentração final de 0.5%) no dia 15 de dezembro de 2014 e processadas apenas no dia 8 de janeiro de 2015 no Instituto Chileno de Pesquisas Antárticas (INACH) em Punta Arenas, Chile. O INACH forneceu o espaço (capela de exaustão) e a bomba de vácuo para o procedimento de filtragem das amostras (Figura 27).



**Figura 27** - Sistema de filtragem de amostras para observação de microorganismos e particulados em microscópios eletrônico e de epifluorescência. O procedimento foi realizado no INACH, Punta Arenas, Chile.



Alíquotas de 15 mL de amostra concentrada por centrifugação a 13000 rpm por 5 minutos foram inoculadas em três meios de cultivo distintos para a obtenção de isolados bacterianos. Os meios utilizados foram: R2B 100%, R2B 10% e Caldo Marinho 10%. O inóculo ocorreu no dia 15 de dezembro de 2014 e desde esta data, os meios de cultivo estão sendo incubados a temperaturas entre 2-5°C.

**Mudanças necessárias para o aprimoramento de coletas para estudos de microorganismos no Criosfera 1:** Mesmo com o levantamento de uma parede móvel dentro do Criosfera 1 para o isolamento de uma área limpa, será necessário para próxima expedição levar uma barraca para ser utilizada como laboratório limpo. Esta barraca (sugestão de ser uma Polar-Heaven) deverá ser instalada ao lado do CrioLab e será necessária a instalação de um aquecedor a gás dentro da barraca para atingirmos temperaturas internas de pelo menos 4 °C. O objetivo é que a porta do pit esteja localizada dentro da barraca. Esta barraca será restrita para circulação de pessoas e será utilizada para os procedimentos de filtragem, inóculo, extração de DNA e outros. O CrioLab precisará ser estendido e aprofundado para novas coletas. Para isso, será necessária a instalação de placas maiores que cobrem o pit (placas que servem como o teto/cobertura do pit). As temperaturas negativas dentro do CrioLab (em geral de -12°C) impossibilitam a filtragem das amostras, sendo que estas congelam em poucos segundos quando em contato com o suporte de membranas. Sendo assim, um melhoramento de infra-estrutura no CrioLab é necessário na próxima missão.

**Amostragem para COVs:** COVs foram amostrados em garrafas metálicas sob vácuo (canisters). A Tabela 3 mostra as datas de coleta de 6 unidades nas proximidades do Criosfera 1.

**Tabela 3** – Datas de coleta de gases no Criosfera 1.

Data	Hora UTC	Pressão hPa	VV (ms <sup>-1</sup> )	Dir.	UR(%)	T(°C)	RAD (Wm <sup>-2</sup> )	CO <sub>2</sub> (ppm)	Nota
10Dez14	23:58h	834.25	3.3	109°	81	-16.5	267	335.2	Nublado/neve fraca
11Dez14	16:25h	833.9	7.0	170°	67.6	-16.7	531		Dentro de PIT fechado por 4 dias
11Dez14	16:25h	833.9	7.0	170°	67.6	-16.7	531		Dia claro sem nuvens
17 Dez14	23h								Vento Leste com "whiteout"
25Dez14	18h	839.4	3.9	184.8°	76.3	-12.8	532.9		Tempo claro
27Dez14	17h	842	2.9	164.2°	77.8	-12.3	492.7		Tempo claro

**Impactos ambientais da missão:** O modelo de missão científica empregado para o Criosfera 1 é o de baixo impacto ambiental, principalmente em decorrência do uso de energia solar e eólica e pelo número de pesquisadores envolvidos. Nos quase 20 dias de acampamento, a equipe fez uso de ZERO litro de combustível. Todas as fezes foram transportadas ao acampamento da ALE, sem processamento químico. Grande parte da urina (em estado congelado) foi encaminhada a ALE em um tonel congelado. O mesmo aconteceu com todos os resíduos de conservantes líquidos resultantes das refeições diárias e escovação de dente, ou seja, a missão não deixou qualquer resíduo líquido ou sólido na área do acampamento. Os métodos de remoção de resíduos empregados na campanha pela equipe foram simplórios e eficientes, porém pouco práticos. Assim é necessário, o desenvolvimento e teste de sistemas portáteis e práticos que possam auxiliar no dia a dia. Abaixo as tentativas implementadas na campanha.



Figura 28 – (esquerda) remanescente de urina excretada em um único ponto sobre a neve (ao lado da barraca banheiro), após enchimento de tonel de 30 L; (direita) acondicionamento de resíduo de cozinha. O material é ensacado e enviado a ALE.



Figura 29 – Lixo acondicionado para ser removido do acampamento. Caixas de marfinito contêm lixo convencional (incluindo-se comida fora do prazo de validade); parte dos sacos pretos se referem a dejetos humanos.

O uso e transporte de baterias para o Criosfera 1 são importantes fontes de impacto ambiental para a região. Apesar de serem seladas, as baterias são do tipo chumbo-ácido. Um dos parâmetros que promovem desgaste nas baterias utilizadas na Antártica é seu uso em ambiente de alto gradiente térmico. Este é o caso da Antártica Central. No Criosfera 1, o gradiente térmico entre verão e inverno pode atingir a marca de 60°C. Observamos que durante o inverno também há alta variabilidade ocorrendo eventos com gradiente térmico de

até 30°C numa escala de tempo de 10 dias. Desta forma, observando o comportamento da atmosfera e em 30 cm abaixo da superfície ao longo do ano, Figura 30, fica evidente o benefício para o uso das baterias quando enterradas na neve, em virtude de sua inegável isolamento térmica. Desta forma procedemos a relocação de um conjunto de 8 baterias para um compartimento na neve em 1 m de profundidade.

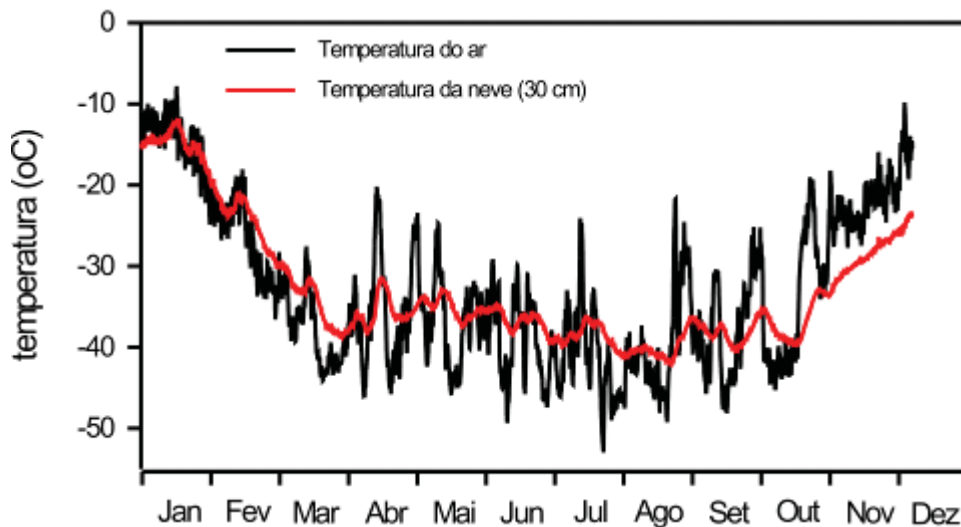


Figura 30 – Temperaturas do ar e da neve (30 cm de profundidade no Criosfera 1). Dados para o ano de 2014.

Para o repositório das baterias (localizado há 1 metro do módulo), as baterias foram acondicionadas em caixas de marfinite (2 unidades por caixa), com isolamento térmico interno (5 cm), sendo totalmente lacrada com cola de silicone, Figura 31.

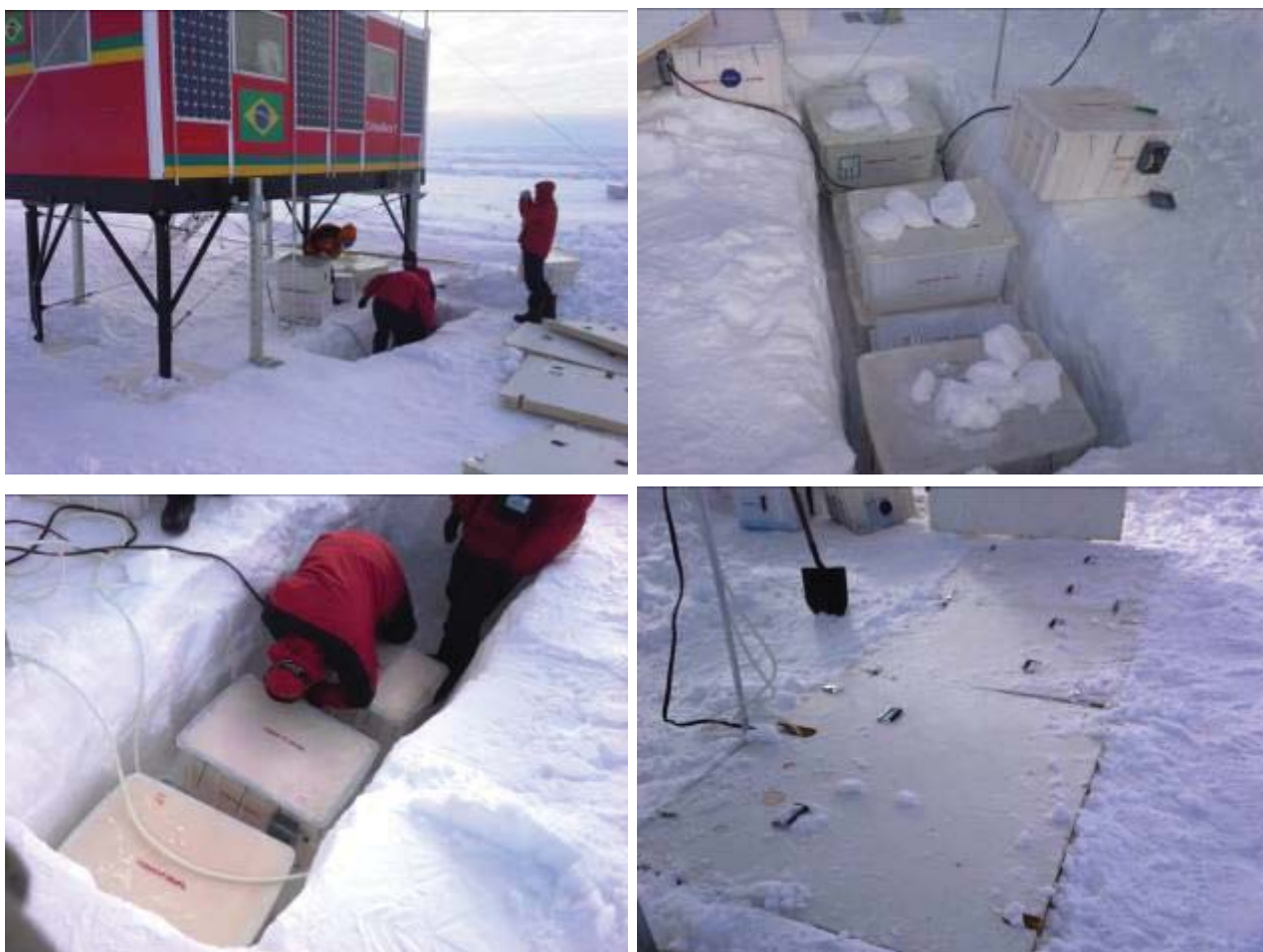


Figura 31 – Soterramento de banco de baterias do Criosfera 1.

**Itens diversos necessários para a próxima missão:**

<b>Item</b>	<b>Justificativa</b>
persianas	Controle luminosidade no interior do Criosfera
ebulidor	Geração de água quente
cadeiras para a cozinha	Material de uso diário
Extintor de CO <sub>2</sub>	Combate a incêndio
tupperware (potes organizadores)	Organizar conectores, material de laboratório e pequenos itens em geral
secador de botas	Material de uso diário
tesoura	Atividades diversas
Aspirador de pó	Evitar geração e acúmulo de poeira no interior do Criosfera 1, mantendo-o o mais limpo possível
Inversor 24VDC / 220 VAC (1500 – 2000 watts)	Funcionamento de equipamentos 220 V
Kit de Fusíveis entre 1 e 10 A	Manutenção de equipamentos
Trenas de vários tamanhos	Atividades diversas
Medidor de pressão arterial, desfibrilador e nebulizador.	Componentes da mala de primeiros socorros

Alicate de bico grande	Atividades diversas
Conjunto de brocas, chaves Phillips, fenda, laminas de corte, persintas, cordim, parafusos diversos, sepos,	Atividades diversas
Auto-Transformador de 1000 watts (110-220V)	Capela e outras atividades
Garrafa térmica	Unidade atual com defeito
Etiquetas diversas	Identificação de itens
Tonéis diversos	Para armazenar urina e restos de resíduo de lavagem de louças e escovação de dente.
Rolamento do aerogerador Forgen (SKF 10106L; 6001-22/LT). Diam. Interno: ~12mm e externo : ~28 mm)	Pecas de manutenção
Tonéis plásticos	Acondicionamento de cabos elétricos
Bússula Magnética Eletrônica	Correção do dado de direção do vento, devido ao movimento do módulo em relação ao manto de gelo
Sistema de recepção de imagem de satélite NOAA	Ter conhecimento em tempo real da condição sinótica da região
Sacos de ráfia (~30 unidades)	Acondicionamento de lixo
Revestimento para o casulo de instrumentação	Melhoria do isolamento térmico e aparência
Fogão com forno	Substituição do pequeno forno existente
Máquina de fazer pão	Valorização do bem estar dos pesquisadores
Microondas	Processar amostras e uso eventual para cozinha
Bateria do Hobbo de temperatura modelo u12-006 4 canais	Material de reposição
Furadeira a bateria 36V	Atividades diversas
Tonéis para separação do lixo em cores diferentes	Coleta seletiva de lixo
Receptor de sinal de satélite	Teste do sinal de transmissão de dados
Regua de tomadas com 10 saídas	Atividades diversas
Caixa de marfinite tamanho grande	Suporte de privada
Kit de reparo de barracas	Manutenção em caso de acidente

### Inventário de materiais do Criosfera 1:

#### 1. CARGA EXTERNA AO MÓDULO P/ VERÃO 2015/2016:

-----

1 unid. Barraca Tipo Pirâmide "Snowsled" (junto ao módulo)

1 unid. Barraca Jabuti "Weatherhaven" (junto ao módulo)

7 unid. Caixa de isopor de alta densidade (**ver detalhamento abaixo no item "LISTAGENS"**)

CPC C-044 (comida)  
CPC C-042 (comida)  
CPC C-040 (vazia)  
CPC C-039 (utilidades/comida pronto uso)  
LARANG #7 (comida)  
LARANG #4 (comida)  
LARANG #2 (comida)

7 unid. Caixas tipo marfinitite  
PROANTAR 12/1398 (Utilidades/Limpeza)  
PROANTAR 11/0168 (utilidades/limpeza)  
PROANTAR 0822 (cozinha)  
PROANTAR 12/0211 (dormitório)  
PROANTAR 3243 (gerador)  
PROANTAR 2241 (Banheiro/Leleco)  
PROANTAR 11/0146 (banheiro)

-----  
2. CARGA ENTERRADA (BANCO DE BATERIAS):  
-----

3 unid. Caixas tipo marfinitite (2 baterias 12V/145Ah, sist. 24V novo)  
PROANTAR 12/2106  
PROANTAR 12/0166  
PROANTAR 11/0674

5 unid. Caixas tipo marfinitite vazias não identificadas:  
PROANTAR 7600  
PROANTAR ...  
PROANTAR ...  
PROANTAR ...  
PROANTAR ...  
{provavelmente 7160, 3243, 12/1395, 12/0976}

-----  
3. CARGA INTERNA NO MÓDULO:  
-----

8 unid. Caixas tipo marfinitite:  
PROANTAR 7490 (Cabos elétricos)  
PROANTAR 10/0358 (Forno Solar)  
PROANTAR 12/1833 (Material de Cozinha)  
PROANTAR 13/0373 (Fogão c/ forno)  
PROANTAR 11/0297 (2 baterias 12V/92Ah, sistema 12V, antigo)  
PROANTAR 7122 (2 baterias 12V/92Ah, sistema 12V, antigo)

PROANTAR # N.A. (2 baterias 12V/92Ah, sistema 24V, antigo)  
PROANTAR 10/0472 (2 baterias 12V/145Ah, sistema 24V, novo)

-----  
4. CARGA RECOLHIDA POR TWIN OTTER NA SAÍDA (28/12/2014):  
-----

ZARGES/LARANG C-189 - Material científico (microbiologia)  
ZARGES/INPE #1 - Material técnico-científico (meteorologia)  
ZARGES/CPC #3 - Vazia.  
ZARGES/CPC #4 - Material técnico (peças de turbinas eólicas)  
ZARGES/CPC #5 - Vazia.  
ZARGES/CPC #6 - Vazia.  
ZARGES/LARANG C-183 - Material científico (química)  
ZARGES/LARANG C-173 - (grande)  
ZARGES/LARANG C-176 - (pequena)  
PROANTAR 2157 - Vazia.  
PROANTAR 3918 - Vazia.  
PROANTAR 7358 - Vazia.  
PROANTAR 10/0460 - Vazia.  
PROANTAR 12/1841 - Vazia.  
PROANTAR 13/0574 - Rádio HF.  
PROANTAR 12/1460 - Turbina Eólica Danificada.  
PROANTAR 12/1866 - Contém zarges C-176.  
PROANTAR 9432 - Vazia.  
PROANTAR N.A. - Vazia.  
INPE (CINZA) - Material técnico.  
PROANTAR Barraca Jabuti "Weatherhaven".  
8 Malas tipo Duffel c/ itens pessoais.

\* RESÍDUOS REMOVIDOS ATÉ ESTAÇÃO UNION GLACIER (ALE):

Cozinha:

4 sacos de resíduo úmido/sólido.  
3 sacos de resíduo líquido congelado.

Banheiro:

2 sacos de dejetos sólidos.  
2 tonéis de dejetos líquidos congelados.

Módulo:

2 sacos de resíduo sólido/tecnológico.

-----  
LISTAGENS:  
-----

C-044 [COMIDA]:

-----

5 pacotes	Corn Flakes (500g)	Venc. 06/2015
2 potes	Café Solúvel (200g)	Venc. 04/2015
20 pacotes	Torrinhas (160g)	Venc. 03/2015
3 pacotes	Papel Filtro nº103	Venc. N.A.

C-042 [COMIDA]:

-----

43 pacotes	Massa Espaguete	Venc. 09/2015
23 pacotes	Massa Riguatone	Venc. 07/2015
5 pacotes	Arroz Tio João (1kg)	Venc. 06/2014
1 pacote	Sal Fino (1kg)	Venc. N.A.
2 pacotes	Sal Marinho (1kg)	Venc. N.A.
1 pacote	Arroz Tucapel (1kg)	Venc. 08/2016
17 pacotes	Massa Instantânea (85g)	Venc. 02/2014

C-040 [VAZIA]:

-----  
Vazia.

C-039 [COZINHA PRONTO USO]:

-----  
Utilidades, material de cozinha e comida do último dia.

#7 [COMIDA]:

-----

7 potes	Leite em pó (650g)	Venc. 07/2015
3 pacotes	Torrinhas (160g)	Venc. 03/2015
3 caixas/8 un.	Barra de cereais	Venc. N.A.
1 pacote	Leite em pó (650g)	Venc. 10/2015
3 potes	Achocolatado (400g)	Venc. 07/2014
9 potes	Café solúvel (170g)	Venc. 04/2015
8 caixas/100un.	Chá Mildred	Venc. 2019
2 pacotes	Café em pó (500g)	Venc. 2012

#2 [COMIDA]:

-----

6 tabletes	Manteiga (250g)	Venc. 10/2013
24 sacos	Castanha de Caju (50g)	Venc. 04/2014
1 saco	Amendoim (200g)	Venc. 10/2014
3 pacotes	Biscoito Agua e Sal (310g)	Venc. 05/2015
1 pacote	Biscoito Agua e Sal (410g)	Venc. 09/2014
1 pacote	Barra de Mantecol (110g)	Venc. 08/2012



3 pacotes	Biscoito Maizena (200g)	Venc. 07/2014
1 saco	Arroz (1kg)	Venc. 05/2014
12 sacos	Batata Liofilizada (80g)	Venc. N.A.
2 pacotes	Massa instantanea	Venc. 04/2014
4 pacotes	Batata Palha	Venc. 04/2014
1 pacote	Massa Espaguete	Venc. 04/2014
1 saco (muito)	Sucos e chás	Venc. N.A.
1 frasco	Syrup (710mL)	Venc. 2016
18 pacotes	Queijo mussarela (500g)	Venc. 01/2015
10 pacotes	Queijo ralado (80g)	Venc. 02/2015
3 pacotes	Queijo Cheedar (144g)	Venc. 02/2015
1 pote	Nescau (400g)	Venc. 07/2014
4 pacotes	Torradinhas (160g)	Venc. 07/2014

#### #4 [COMIDA]:

-----

1 frasco	Adoçante Stevia (180mL)	Venc. 11/2016
2 frascos	Sal marinho (tempero)	Venc. N.A.
1 frasco	Pimenta Negra (50g)	Venc. N.A.
1 frasco	Aliño Completo (25g)	Venc. N.A.
1 frasco	Alecrim (11g)	Venc. 08/2017
1 frasco	Aji de Color (25g)	Venc. N.A.
8 sacos	Aliño completo (100g)	Venc. N.A.
1 saco	Oregano (50g)	Venc. N.A.
1 saco	Curry (15g)	Venc. N.A.
1 saco	Cogumelos Secos (35g)	Venc. N.A.
1 saco	Canela em casca (20g)	Venc. N.A.
7 sacos	Sopão Maggi (200g)	Venc. 07/2014
1 frasco	Ketchup (1kg)	Venc. 2016
4 frascos	Azeite vegetal (1L)	Venc. 09/2013
4 latas	Feijão Doce (400g)	Venc. 11/2015
6 frascos	Cebolinha em conserva (100g)	Venc. 04/2015
3 latas	Palmito em conserva (500g)	Venc. 03/2018
1 vidro	Azeitonas em conserva (500g)	Venc. 03/2015
2 latas	Almondega Bobina (250g)	Venc. 09/2017
1 lata	Sardinha Enlatada (100g)	Venc. 03/2017
15 latas	Atum Enlatado (120g)	Venc. 08/2015
1 lata	Cereja Enlatada (230g)	Venc. 12/2017
5 caixas	Leite condensado (400g)	Venc. 03/2014
6 latas	Leite condensado (400g)	Venc. 05/2014
1 lata	Milho em broto (230g)	Venc. 06/2016
1 lata	Morango enlatado (410g)	Venc. 12/2017
1 lata	Salada de frutas (560g)	Venc. 10/2016
1 saco	Milho p/ pipoca (500g)	Venc. 11/2013
1 saco	Ervilha seca (1 kg)	Venc. 01/2015
2 sacos	Lentilha seca (1 kg)	Venc. 03/2015
11 sacos	Açucar (1kg)	Venc. N.A.

MARFINITE 12/1398 [UTILIDADES/LIMPEZA]:

-----

49 rolos	Papel Higiênico	Venc. N.A.
1 pote	Detergente em pó	Venc. N.A.
8 potes	Removedor "Vanish" (450g)	Venc. N.A.
1,5L	Hipoclorito de sódio 2%	Venc. N.A.
2 frascos	Álcool Gel 46°GL (500mL)	Venc. N.A.
1 pacote	Toalhas umedecidas	Venc. N.A.
2 frascos	Protetor solar FPS 50 (190mL)	Venc. N.A.
1 frasco	Desodorante "Men Care" (50mL)	Venc. N.A.
25 unidades	Saco de lixo 200L grosso	Venc. N.A.
90 unidades	Saco de lixo 50L (azul)	Venc. N.A.
50 unidades	Saco de lixo 30L (preto)	Venc. N.A.
30 unidades	Saco de lixo 100L (azul)	Venc. N.A.

MARFINITE 11/0168 [UTILIDADES/LIMPEZA]:

-----

620 caixas	Fósforos	Venc. N.A.
18 unidades	Esponjas	Venc. N.A.
15 unidades	Isqueiros	Venc. N.A.
12 unidades	Papel Toalha	Venc. N.A.
15 unidades	Saco 100L grande/grosso	Venc. N.A.
2,5L	Detergente líquido louça	Venc. N.A.
12 pares	Luvax Látex p/ limpeza	Venc. N.A.
13 metros	Papel Alumínio	Venc. N.A.
3 sacos	Esponja de ferro "Bombril"	Venc. N.A.
2 unidades	Filtro p/ água	Venc. N.A.
10 unidades	Sacos c/ fecho grandes	Venc. N.A.
20 unidades	Sacos de lixo 30Litros	Venc. N.A.
20 unidades	Sacos plástico p/ comida	Venc. N.A.
1 frasco	Álcool Gel 46°GL (500mL)	Venc. N.A.
1 frasco	Álcool líquido 94°GL (1L)	Venc. N.A.

MARFINITE 12/0211 [DORMITÓRIO]:

-----

5 unidades	Esteira Isolante térmica	Venc. N.A.
5 unidades	Colchão de ar	Venc. N.A.

MARFINITE 11/0146 [BANHEIRO]:

-----

Estocagem de dejetos externamente ao banheiro.

MARFINITE 2241 [LELECO/BANHEIRO]:

-----

Assento p/ privada  
Estrutura p/ assento

Balde 20L

MARFINITE 0822 [COZINHA]

-----  
Louças diversas.

MARFINITE 3243 [GERADOR]:

-----  
1 Gerador Portátil  
2 funis  
60m corda dinâmica

Abaixo uma relação de itens deixados no Criosfera 1 deixados no local pela campanha 2014.



Figura 32 – Tonéis de combustível (2 lacrados e 1 já utilizado) e 1 botijão de gás (lacre rompido durante sua escavação da neve, o mesmo se encontra cheio) deixados na antiga pista de pouso do Criosfera 1 em 2014.