1. **Desenvolvimento do Projeto**

O sistema de detecção de múons conta com 3 partes principais: o detector, a eletrônica de *Front-End* (FEE) e a unidade de aquisição de dados unificada, o DAQ (*Data Acquisition*), com a FEE e o DAQ integrados em uma única PCB.

A placa possui dois módulos conversores DC-DC, um com saída de 5 V e outro com saída de 3.3 V, que alimentam todos os demais módulos e ICs com suas respectivas tensões de entrada. Os conversores são alimentados por uma entrada de tensão da PCB, que pode receber de 12 até 24 V. Foram adicionados sensores de corrente na saída dos conversores para monitoramento da alimentação da placa pelo DAQ. A PCB conta ainda com módulos DAC (*Digital Analog Converter*), responsáveis por controlar o valor diversas tensões do circuito, como tensões de *offset* dos amplificadores operacionais e tensões de referência dos discriminadores. O controle de tensão dos DACs é realizado pela firmware da FPGA.

O detector é composto de um conjunto de tiras cintilantes plásticas, os cintiladores. Cada tira conta com um filamento de fibra ótica WLS (*Wavelength Shifter*) acoplado em seu interior e uma fotomultiplicadora SiPM (*Silicon Photomultiplier*) em sua extremidade. Os cintiladores são feitos de um material fluorescente, cujos átomos emitem fótons ao serem ionizados por uma partícula carregada que o atravessa, nesse caso, os múons a serem detectados. O sinal luminoso é então guiado até a extremidade da tira, chegando à SiPM, onde é transformado em um sinal elétrico correspondente. Na placa existe um módulo de alta tensão, responsável por gerar a tensão de polarização reversa das fotomultiplicadoras.

O sinal gerado é então tratado pela eletrônica de *Front-End*. O pulso de saída da SiPM é primeiro acoplado e amplificado, chegando então ao discriminador. Esse é basicamente um comparador, com uma tensão de referência previamente escolhida. Quando o sinal amplificado atinge esse limiar, a saída do comparador vai para alto. Isso é feito com o intuito de discernir uma detecção real, ou seja, um sinal com amplitude suficiente para caracterizar uma excitação do cintilador por uma partícula, dos sinais de ruído presentes no circuito. A saída do comparador é então alimentada em um buffer que, por fim, repete o sinal digital de detecção daquele canal e o alimenta numa entrada de uma unidade FPGA no DAQ, responsável pela contagem de pulsos.

O DAQ abrange sensores de temperatura, pressão, umidade, campo magnético e acelerômetro (para verificar a ortogonalidade do experimento com a superfície) para monitoramento das condições de contorno do experimento, leitura de tensões e correntes relevantes no circuito, relógio digital, GPS, um módulo de comunicação Ethernet, um módulo para leitura e escrita de cartão microSD para armazenamento de dados, um microcontrolador ESP32 e um FPGA.

O FPGA recebe os pulsos digitais que vêm da FEE em suas entradas, sendo responsável pela contagem de detecções individuais em cada canal, assim como detecções simultâneas entre dois ou mais canais. Cada uma dessas contagens é realizada por um tempo predeterminado de aquisição (TAQ). Ao fim desse tempo, o bloco de contagens é então enviado para um microcontrolador.

O microcontrolador é responsável pela aquisição, processamento, armazenamento e envio de todos os dados obtidos no experimento, como leituras de sensores e, principalmente, as contagens de pulsos. Ele se conecta a um PC via USB, recebendo e enviando leituras de dados, status do experimento e variáveis de configuração e controle.

A firmware executada pelo microcontrolador se comunica continuamente com o FPGA, esperando confirmação de que o último bloco de contagens foi processado. Quando isso ocorre, essas contagens são lidas e compiladas com os dados dos sensores e demais CIs de interesse a intervalos regulares, sendo elaborado um dataframe que é gravado no microSD e enviado para um servidor no CBPF em tempo real, via módulo Ethernet. Esses dados também são enviados ao PC, que pode ser monitorado remotamente através do software *TeamViewer*.

* 1. **Módulo Detector**
     1. **Tiras Cintilantes *SciTile***

Os cintiladores são materiais que exibem o fenômeno de cintilação, a emissão de luz mediante a excitação por radiação ionizante. Quando uma partícula carregada atravessa um cintilador e interage com elétrons de seus átomos, parte de sua energia é depositada nesses elétrons, elevando-os a níveis orbitais de maior energia, porém instáveis. Ao retornarem aos orbitais originais, a energia adicional é liberada através da emissão de um fóton.

As tiras cintilantes plásticas usados no detector do projeto foram desenvolvidas pelo Fermilab (*Fermi National Accelerator Laboratory*) e são feitas de um material fluorescente, que emite fótons com comprimento de onda na faixa do azul ao ser excitado por um raio cósmico. As tiras possuem um orifício ao longo de seu comprimento para a acoplação da fibra óptica WLS.

Os cintiladores são cobertos por uma casca opaca, que bloqueia a incidência da luz externa. O interior da casca é revestido de um material reflexivo, com o objetivo de refletir os fótons emitidos pelo processo de cintilação, aumentando a probabilidade de que eles atinjam e sejam conduzidos pela fibra WLS.



Figura: Montagem mecânica da tira cintilante com fibra *Wavelength Shifter*

* + 1. **Fibra Óptica WLS**

A fibra óptica *Wavelength Shifter* (WLS) acoplada no cintilador é do modelo Y-11(175)MSJ e é produzida pela *KURARAY*. Ela absorve os fótons de maior energia, no espectro do azul, emitidos pelas tiras cintilantes, e reemite múltiplos fótons de menor energia, no espectro do verde, conduzindo-os até a fotomultiplicadora SiPM. O maior número de fótons aumento a eficiência do detector, já que muitos se perdem no caminho até a SiPM. Além disso, a fotomultiplicadora apresenta maior rendimento na frequência do verde.



Figura – Fibras ópticas WLS.

* + 1. **Fotomultiplicadora SiPM**

A fotomultiplicadora SiPM (*Silicon Photomultiplier*) é um fotodetector de estado sólido que, em resposta à absorção de um fóton, produz um pulso de corrente com duração na ordem de dezenas de nanossegundos. Ela é composta de uma matriz de microcélulas, sendo cada célula um fotodiodo de avalanche operando em modo Geiger, com a tensão de polarização reversa um pouco acima da tensão de ruptura. Isso torna a SiPM sensível suficiente para detectar a passagem de um único fóton.



Figura: Fotomultiplicadora SiPM da série S13360.

Quando um fóton acerta uma microcélula, é produzida uma avalanche de portadores de carga, podendo chegar a um número de 105 a 106, gerando um pulso de corrente detectável. As células são arranjadas num circuito em paralelo umas com as outras. Com isso o sinal na saída da fotomultiplicadora é a soma dos sinais em cada célula e é proporcional ao número de fótons detectados, permitindo uma leitura de alta precisão.



Figura: Forma de onda do pulso de saída da S13360.

O ganho da fotomultiplicadora é o número de portadores de carga gerados em uma descarga, e depende linearmente da diferença entre a tensão de polarização VBIAS e a tensão de ruptura VBR (*breakdown voltage*):

A tensão de ruptura varia com a temperatura, o que pode afetar o ganho. Para garantir um ganho constante é necessário operar a SiPM com um VBIAS que garanta um constante apesar das variações de temperatura.

Nesse projeto, são usadas SiPMs da série S13360 em conjunto com uma fonte de tensão C11204-02, ambos da *Hamamatsu*. A fonte fornece a tensão de polarização da fotomultiplicadora, na faixa de 40 V a 90 V, e é programável via protocolo serial UART. Ela contém uma função de compensação de temperatura, que ajusta sua tensão de saída de acordo com as variações de temperatura, buscando manter o ganho da SiPM constante. Para isso, ela monitora o output analógico do sensor de temperatura LM94021.



Figura: Circuito da fonte de tensão de polarização da SiPM.

* 1. **Eletrônica de Front-End**

A amplitude e largura dos pulsos advindos da SiPM são muito pequenos para detecção pelas portas de entrada do FPGA. Para isso, o circuito de *Front-End* (FEE) é responsável por amplificar e alongar esse sinal. Além disso, para diferenciar o pulso proveniente da detecção de uma partícula de sinais de ruído, que podem ser de origem térmica, da fonte de alimentação ou de interferência externa, a saída do circuito amplificador passa por um discriminador, cuja saída é então alimentada no módulo contador, a unidade FPGA. Segue abaixo o esquemático do circuito:



O circuito possui dois estágio de amplificação, sendo o ganho total o produto do ganho de cada estágio. O discriminador se trata basicamente de um comparador ADCMP600. Quando o sinal de entrada ultrapassa a tensão de referência (*threshold*) na porta Vn, a saída vai para alto, transformando o pulso analógico em um pulso digital.

A tensão de referência pode ser gerada por um circuito manual de *threshold*, que usa um *trimpot* como divisor de tensão, ou a partir de uma saída de um dos módulos DAC, representada pela tensão DACthDisc no esquemático. A escolha da tensão é feita através da ligação via *jumper* entre o terminal do *header* conectado ao discriminador e o terminal com a tensão escolhida.

* 1. **Módulo DAQ**

A unidade de aquisição de dados possui dois componentes principais: a unidade FPGA e o microcontrolador ESP32. Ela também abrange sensores de temperatura, pressão, umidade, campo magnético e acelerômetro (para verificar a ortogonalidade do experimento com a superfície) para monitoramento das condições de contorno do experimento, conversores analógico-digitais (ADC) para a leitura de tensões relevantes no circuito, relógio digital, GPS, um módulo de comunicação Ethernet e um módulo para leitura e escrita de cartão microSD.

O FPGA desempenha o papel de módulo contador de eventos. Ele recebe os pulsos digitais dos 12 canais da Front-End, oriundos da amplificação, alongamento e discriminação dos pulsos analógicos gerados pelas SiPM de cada canal. Após cada período de contagem, esses dados são enviados para o microcontrolador.

O ESP32 age como um módulo de controle central do experimento. Ele é responsável pela aquisição, processamento, armazenamento e envio de todos os dados obtidos no experimento, como leituras de sensores e, principalmente, as contagens de pulsos. Ele se conecta a um PC via USB, recebendo e enviando leituras de dados, status do experimento e variáveis de configuração e controle.

**2.3.1. Módulo Contador de Eventos**

O módulo contador é implementado como um circuito programável através da firmware da unidade FPGA. Ele é responsável pela contagem do número de pulsos digitais vindos da eletrônica de Front-End, da saída do discriminador. Cada um desses pulsos corresponde a um sinal analógico na saída de uma SiPM que é processado por um dos doze canais da FEE.

Os cintiladores são organizados em três grupos de quatro tiras cintilantes. Os canais de 1 a 4 formam o grupo A, os canais de 5 a 8 formam o grupo B, e os canais de 9 a 12 formam o grupo C. A firmware realiza a contagem de pulsos individuais em cada um dos canais e também realiza contagens de todas as possíveis coincidências duplas, triplas e quádrupla entre os canais de um mesmos grupo.

O FPGA deriva seu sinal de clock de um cristal oscilador de 50 MHz, representado pelo nome LOCALCLK. Na firmware, é utilizado um módulo PLL, que a partir do LOCALCLK deriva como output um sinal de 100 kHz, usado como clock do bloco de controle dos módulos DAC do circuito, e um sinal repetido de 50 MHz, que tem o papel de sinal de clock de todos os demais blocos lógicos.

Devido a instabilidades no sinal da fotomultiplicadora e ruídos de chaveamento no discriminador é possível que um único pulso vindo do detector gere a detecção de diversas bordas de subida no pino de entrada correspondente da FPGA. Para evitar que isso aconteça, cada um dos doze canais digitais de entrada é alimentado em um bloco denominado Gate Timer. Quando ocorre uma borda de subida em um desses canais, a saída correspondente nesse bloco é mantida em alto por um tempo configurável chamado nT\_GATE. A duração do gate tem o valor de um número inteiro de ciclos de clock que vai de 1 a 255.

Os doze sinais de saída do Gate Timer então chegam no bloco de coincidências. Esse bloco é responsável por determinar as coincidências de detecção em canais de um mesmo grupo. Isso é implementado alimentando cada sinal de um grupo em portas AND organizadas de acordo com cada uma das combinações possíveis de dois, três ou quatro canais. Assim, por exemplo, para o grupo A, temos os canais A1, A2, A3 e A4. Com isso, as combinações possíveis são:

* Combinações duplas: A1A2, A1A3, A1A4, A2A3, A2A4 e A3A4;
* Combinações triplas: A1A2A3, A1A2A4, A1A3A4 e A2A3A4;
* Combinação quádrupla: A1A2A3A4;

Os grupos B e C apresentam combinações análogas entre seus canais. Assim temos 6 combinações duplas, 4 combinações triplas e uma quádrupla por grupo. A imagem abaixa detalha o esquema de lógica combinacional:

IMAGEM DO COINCIDENCE BLOCK

Finalmente, os pulsos individuais e os pulsos de coincidência são direcionados para a entrada do bloco de contagem. Com 11 combinações mais 4 pulsos individuais, são 15 contagens por grupo, num total de 45 contagens. Cada contagem é armazenada em 3 bytes de memória, com todas as contagens ocupando em todo 135 bytes.

Quando é detectada uma borda de subida em uma das 45 entradas, a contagem correspondente é incrementa em uma unidade. As contagens vão sendo atualizadas pela duração do ciclo de contagem atual. Uma entrada de RELEASE é responsável por sinalizar o fim do ciclo em curso. Quando RELEASE sofre uma borda de descida, cada valor de cada contagem é descarregado em saídas correspondentes do bloco, para depois serem armazenadas no bloco de memória da firmware. As contagens então voltam para zero, e inicia-se um novo ciclo.

O bloco de memória é formado pelos bancos A e B. Ele é acessível através de um bloco de comunicação serial via protocolo I2C para o usuário que, no caso deste experimento, é o microcontrolador ESP32. O banco A é acessível apenas para leitura, sendo responsável por armazenar as 45 contagens de pulsos vindas do bloco de contagem. Já o banco B é acessível para leitura e escrita, e armazena variáveis de configuração e de status, que são usadas para mediar e sincronizar a comunicação entre o usuário e a FPGA.

Cada banco possui 255 bytes de memória, podendo ser endereçados por um único byte. Assim os endereços de memória vão de 0x00 a 0xFF. Os cinco endereços mais altos, de 0xFF a 0xFB, são comuns aos dois bancos, representando, nessa ordem, a versão da firmware, o reset de software, a seleção de banco, e os endereços de FLAG e ACK. O banco A é completamente reservado para contagens de pulsos. O banco B armazena variáveis para configuração do período de um ciclo de contagem de pulsos, o TAQ (*time of acquisition*), o número de ciclos de clock em um período do Gate Timer, o nT\_GATE, o reset do bloco de controle dos módulos DAC, e o controle dos LEDs do módulo FPGA para testes.

Os endereços de FLAG e ACK servem para sincronizar a comunicação entre o FPGA e o microcontrolador. Quando um ciclo de aquisição, ou de contagem, chega ao fim, o valor do FLAG se torna 0x01. O ESP32, que monitora continuamente o FLAG, interpreta isso como sinal de que as últimas contagens estão prontas para serem lidas. O ACK é um endereço reservado para escrita pelo microcontrolador, e é setado para 0x01 quando todas as contagens já foram lidas. Com isso, a firmware seta o FLAG e, em seguida, o ACK, ambos de volta para 0x00.

A arbitragem da duração do ciclo de aquisição, da atualização de variáveis de configuração, e da comunicação com o microcontrolador é feita pelo bloco de controle. Ao começo de um novo ciclo, ele lê da memória o valor mais recente configurado para o TAQ e o nT\_GATE, sendo esse enviado para o Gate Timer. Então, um bloco interno de cronometragem marca a passagem do número de segundos desde o início do ciclo. Quando esse número se iguala ao valor do TAQ, o sinal de saída de RELEASE vai para alto, fazendo o bloco de contagem descarregar as contagens no banco A da memória. Por fim, o endereço de FLAG é setado, o RELEASE volta para zero, e um novo ciclo se inicia.

DESCREVER O BLOCO DE CONTROLES DOS DACS

**2.3.2. Firmware do ESP32**

O microcontrolador age como controlador mestre de todos os demais módulos no DAQ. Assim que o experimento é ligado, a firmware do ESP32 realiza a inicialização de seus módulos de comunicação serial, os quais são usados para comunicação com os sensores, módulos ADC, módulo de cartão SD, módulo Ethernet e com a unidade FPGA.

A temporização do experimento é realizada em conjunto pelo relógio interno do ESP32, pelo módulo Real-Time Clock (RTC) e pelo módulo GPS. Quando a PCB é ligada, o microcontrolador registra o horário UTC obtido via satélite pelo GPS. Esse horário é então usado como referência para medida do tempo a cada instante do experimento pelo relógio interno. Além disso, esse tempo é gravado no RTC, que é alimentado por uma bateria própria, e serve para garantir que o sincronismo temporal não seja perdido mesmo que a placa venha a ser desligada, por exemplo, por uma possível falha na alimentação energética do experimento. Isso é importante, já que a fixação do sinal de satélite pelo GPS pode levar até dezenas de minutos até ser obtida.

Para garantir que não haja *drift* na marcação do tempo depois de um período extenso desde a última fixação por satélite, considerando que o experimento deverá permanecer ativo sem interrupção por meses ou anos, o horário é corrigido periodicamente pela leitura do GPS. Esse período é programável pelo usuário, podendo ter um valor na ordem de dias.

Os dados adquiridos, processados, salvos e enviados pelo microcontrolador consistem nos valores de contagens vindos da FPGA, mais os valores de leituras dos sensores, referidos como dados de *slow control*. Esses dados são organizados em um *dataframe* que é posteriormente salvo no cartão o microSD e enviado para um servidor no CBPF via Ethernet.

O *dataframe* é dividido em *frames* e *blocks*. Cada *block* consiste na coleção de todas as contagens de pulsos feitas dentro da duração de um período de aquisição (TAQ), um tempo de início e um índice de bloco. O *frame* é composto por um conjunto de *blocks*, tendo a duração de um TAQ vezes o número de blocos que possui. Ele também agrega o conjunto de medidas de *slow control* feitas ao longo de sua duração. Cada medida, por exemplo, de temperatura, é feita uma vez no início de cada bloco, e ao final é armazenada a média dessas medidas. Isso é feito devido às medidas de slow controlserem quantidades de baixa variabilidade ao longo do tempo de medição característico de um frame, da ordem de minutos. Por fim, a cada frame são associados um tempo de início, um índice de frame, e um conjunto de configurações, de forma que, quando novas configurações são escritas no microcontrolador, essas são aplicadas somente ao início de um novo frame.

Assim que o experimento é ligado, o microcontrolador entra em uma fase de *setup* seguida do ciclo normal de execução. No setup, a firmware inicializa variáveis de código, estabelece a comunicação com os sensores, carrega as configurações mais recentes armazenadas no cartão SD, escreve as configurações do FPGA, escreve as configurações do módulo de alta tensão, e ajusta a temporização do experimento conforme descrito acima.

Ao entrar no ciclo normal de execução um novo frame é construído, com as últimas configurações sendo associadas a ele, e seu cabeçalho é escrito no cartão SD. O endereço de FLAG do FPGA é lido continuamente, esperando a sinalização de término do ciclo de contagem atual. Quando a FLAG vai para alto, é enviado um ACK. São lidas todas as contagens armazenadas no banco A da FPGA, é registrado o tempo associado à recepção desse block de contagens, e são feitas as leituras de slow control correspondentes. O bloco recebido é armazenado no frame atual e escrito no cartão SD. Esses passos se repetem até que o frame atual seja povoado com o número de blocos por frame configurado. Quando o último bloco é recebido, é feita média das leituras de slow control, e essas são escritas no SD. O frame inteiro é então enviado via Ethernet, um novo frame é construído, e repete-se o ciclo.

No ciclo, há ainda uma segunda fase, quando a FLAG está em baixo. É então aproveitado o tempo para verificar a existência se mensagens seriais vindas do PC. Essas mensagens podem incluir novas configurações, ou pedidos de envio de dados, e serão descritas em melhor detalhe na próxima seção.

**2.3.3. Software de controle via PC**

O microcontrolador é responsável pela aquisição