

Desenvolvimento de um Sistema de Controle e Aquisição de Dados para Testes do Detector de Fibras Cintilantes do Experimento LHCb

Development of a control system and data acquisition for Tests of the Scintillating Fibers Detector at the LHCb experiment

Maurício Féo Pereira Rivello de Carvalho*

Centro Brasileiro de Pesquisas Físicas

André Massafferri Rodrigues†

Centro Brasileiro de Pesquisas Físicas

Submetido em 13/06/2016 Aceito: 18/11/2016

Resumo: O experimento LHCb será submetido a um upgrade para se adequar aos novos regimes de luminosidade que o LHC oferecerá. As estações de traço atuais não são adequadas para funcionar sob os níveis de ocupação esperado, e por isto o upgrade também compreenderá a troca dos atuais detectores das estações T1, T2 e T3. O novo detector será composto por fibras cintilantes, enquanto o antigo compreendia tubos de *Straw* e tiras de silício. A atualização requer não somente o desenvolvimento do novo detector em si como também o desenvolvimento de nova eletrônica de front-end. Neste contexto, o CBPF se insere como responsável por desenvolver um sistema de teste para validação dos módulos eletrônicos durante a etapa de produção da eletrônica e validação do detector durante a etapa de montagem. O presente trabalho consiste no desenvolvimento de um sistema de controle para os testes mencionados, que compreendem a validação de toda a eletrônica de aquisição, a configuração da mesma para funcionar juntamente com o sistema supervisório (SCADA) do projeto e o desenvolvimento de painéis para controle e monitoramento de registradores internos à front-end.

Palavras chave: LHCb; LHC; Upgrade; Detector de Fibras Cintilantes; SCADA; Sistema de Controle.

Abstract: The LHCb experiment

The LHCb experiment will undergo an upgrade to suit the new levels of luminosity that the LHC will offer. The current tracking stations are not suitable to work under the occupancy levels expected, and this upgrade will include the exchange of current detectors of stations T1, T2 and T3. The new detector will consist of scintillating fibers while the former comprised of Straw Tubes and silicon strips. The update requires not only the development of the new detector itself as well as the development of the front-end electronics.

In this context, CBPF is inserted the responsible for developing a test system to validate the electronic modules during the production step of the electronics and validation of the detector during the assembly step.

This work consists in the development of a control system for the mentioned tests, that includes validation of all acquisition electronics, its configuration to work with the supervisory system (SCADA) used in the design and development of panels for control and monitoring of internal registers in the front end.

Keywords: LHCb; LHC; Upgrade; Scintillating Fibers Detector; SCADA; Control system.

*Electronic address: mauriciofeo@gmail.com

†Electronic address: massafferri@cbpf.br

1. O EXPERIMENTO LHCb E O UPGRADE

O experimento *Large Hadron Collider beauty* (LHCb) [4] é um dos experimentos do colisor de partículas LHC [1]. O LHCb se trata de um espectrômetro de braço único que cobre a região limitada pelos ângulos de 10mrad até 300mrad no plano horizontal e 10mrad até 250mrad no plano vertical a partir do ponto de colisão. O LHCb se encontra no local denominado como Ponto 8 do LHC, em uma região experimental escavada em torno de um dos 4 pontos em que ocorrem colisões no LHC. Ele é composto por um conjunto de detectores e elementos que permitem a identificação de partículas e reconstrução de sua trajetória. Dentre estes, os principais são: o Localizador de Vértice (VeLo) [23], o Sistema de Traço [24] [25], o Sistema de Múons [10], o Magneto [26], os Calorímetros Hadrônicos e Eletromagnéticos [9], e os detectores *Ring-Imaging Cherenkov* (RICH) [8]. O Sistema de Traço é subdividido em 4 estações de traço: TT (*Trigger Tracker*), T1, T2 e T3. Na Figura 1 podemos ver a geometria do experimento LHCb e a posição de seus sistemas e detectores. As Estações T1 a T3 são formadas ainda por 2 tipos de detectores diferentes. O IT (*Inner Tracker*) [25], formado de tiras de silício, e o OT (*Outer Tracker*) [24], formado por tubos de *Straw*.

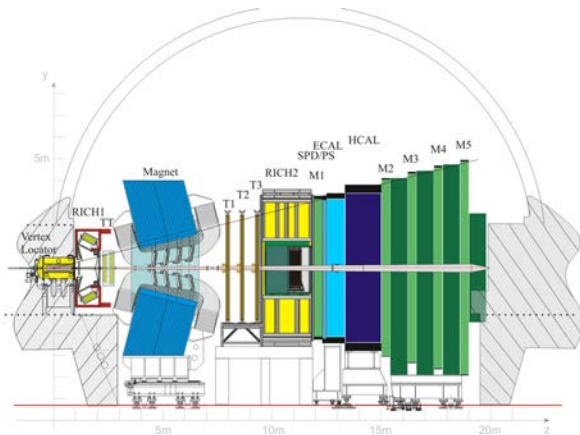


Figura 1: Vista lateral do detector LHCb.

O objetivo do LHCb é estudar a violação de simetria de carga-paridade e processos raros no decaimento de b-hádrons produzidos a partir das colisões do LHC.

O experimento LHCb passará por um upgrade durante a próxima interrupção longa (LS2) do LHC, que está previsto para os anos de 2019 e 2020, tendo seu retorno em 2021. O upgrade do LHCb visa a adequação do experimento aos regimes de maior luminosidade que serão oferecidos pelo acelerador LHC, que terá um aumento de 5 a 7 vezes na luminosidade nominal para a qual foi projetado.

Com o upgrade do LHCb, espera-se elevar o seu rendimento físico em decaimentos com muon por um fator de 10, o rendimento dos canais hadrônicos por um fator de

20, coletando dados a uma luminosidade aproximada de $2 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$. Isto corresponde a dez vezes a luminosidade do projeto atual e permitirá ao LHCb elevar o seu volume de dados coletados anualmente de 1 fb^{-1} de luminosidade integrada para 5 a 10 fb^{-1} . Uma Carta de Interesse [12], um TDR de Framework [13] e um TDR de Trigger e On-line [14] documentam os planos para atualização dos componentes do experimento.

O upgrade do LHCb será baseado na extinção do trigger de nível 0 e aquisição de dados a uma taxa de 40MHz. O aumento na luminosidade do LHCb trará como consequência a elevação significativa do nível de radiação incidente, do volume de dados gerados e da ocupação nos canais dos detectores. Os novos níveis de radiação exigem a validação de todos os componentes do sistema e requer o uso de eletrônica apropriada. O volume de dados exige substituição de toda eletrônica de aquisição do experimento e também da eletrônica de front-end (FEE) dos detectores. Por ter a FEE embutida no módulo de detecção, os detectores VELO, IT, TT e RICH também terão de ser substituídos. O novo nível de ocupação no detector OT exige uma reformulação completa de sua arquitetura. Por estas razões, as estações de traço T1 a T3 (compostas pelo IT e OT) serão completamente substituídas por um novo detector, que utiliza tecnologia de fibras cintilantes.

2. DETECTOR DE FIBRAS CINTILANTES

O Detector de Fibras Cintilantes (SciFi) é o detector responsável por determinar as coordenadas em que partículas carregadas cruzaram as estações de traço T1, T2 e T3. O objetivo do SciFi é fornecer as informações necessárias para reconstrução dos traçados, determinação da massa e momento de partículas carregadas que atravessam o detector, e prover informações de posição das partículas necessárias ao detector RICH para identificação das mesmas.

O SciFi detecta a passagem de partículas pelo fenômeno de cintilação que ocorre na fibra. Ao ser atravessado por uma partícula, esta causa a excitação dos átomos do material cintilante das fibras, gerando fótons. Estes fótons são guiados pela fibra até suas extremidades, onde são convertidos em sinal elétrico por fotomultiplicadoras de silício (SiPM).

O detector de fibras cintilantes consistirá de três estações de traço, entre o magneto e o detector RICH2. Cada estação do SciFi será composta por 4 camadas de detecção. As 3 estações são centradas no mesmo ponto no eixo z, que é o tubo do LHC. As camadas de detecção de cada estação se encontram, em sequência, nos ângulos 0° , $+5^\circ$, -5° e 0° , correspondendo aos planos X, U, V e X respectivamente. Há um espaço de 20cm entre cada camada de cada estação, e há um orifício circular no centro do detector pelo qual passa o tubo do feixe do LHC. Cada camada de detecção é subdividida em 12 módulos de fibras cintilantes. Cada módulo possui 5m de altura por 51cm de largura. Cada módulo do detector é subdividido em oito *mats* (esteiras) de fibras cintilantes com 2,5m de comprimento.

O detector completo cobre uma área de 5m de altura por 6m de largura, no plano X-Y. Ao total, o detector SciFi compreende 144 módulos. Nas extremidades de cada módulo

há 16 fotomultiplicadoras de silício (SiPM), que são responsáveis por converter a luz proveniente das fibras em sinais elétricos. Na parte interna do módulo, nas extremidades centrais de cada *mat*, são anexados espelhos, para que a luz que for em direção ao centro dos módulos seja refletida de volta às extremidades contendo as SiPMs. A Figura 3 mostra como é o arranjo dos módulos e um comparativo de tamanho com uma pessoa de estatura média.

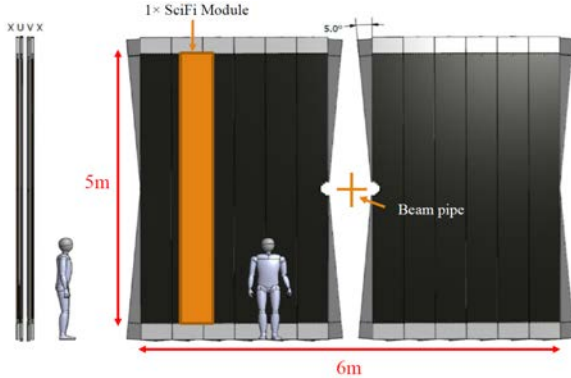


Figura 2: Modelo 3D do arranjo dos módulos do detector SciFi.

As fibras cintilantes são o elemento ativo de detecção do projeto SciFi. São fibras de 250 μ m de diâmetro com núcleo de poliestireno revestido por duas camadas cujos índices de refração são decrescentes de dentro para fora. A camada de revestimento interior é formada por polimetil-metacrilato (PMMA), que compõe o núcleo cintilante da fibra, e a camada exterior é feita de um polímero fluoretado. O projeto das camadas de revestimento das fibras são feitos de modo a otimizar a reflexão interna de luz, fazendo com que esta seja guiada pela fibra para as extremidades, onde são lidas pelas SiPM.

Anexados às extremidades dos módulos de fibras se encontram os módulos de leitura chamados de *Read-Out Box* (ROB). São nos ROBs que se encontram as SiPMs. O ROB provê o alinhamento entre as fibras e as SiPM, estrutura para a eletrônica de *front-end* e as chamadas *ColdBox*, caixa responsável por fornecer resfriamento às SiPM (que operam a -40°C). Cada ROB possui uma *ColdBox* com 16 SiPMs e 2 eletrônicas de *front-end*.

2.1. Eletrônica de Front-End do SciFi

A eletrônica de Front-End é a responsável pela leitura dos sinais elétricos provenientes das fotomultiplicadoras, pela digitalização, processamento e envio destes através de transmissores ópticos. A Front-End do SciFi é modularizada e composta por 3 tipos de módulos. Cada Front-End contém 4 módulos PACIFIC, 4 módulos de clusterização e um módulo Master. O arranjo dos módulos na *front-end* pode ser visto na Figura 3.

O módulo PACIFIC é onde a saída da fotomultiplicadora é conectada através de um cabo *flat*. Esta recebe até duas fotomultiplicadoras e possui um chip de leitura chamado PACIFIC [19], responsável por amplificar, modelar, integrar e

digitalizar os sinais elétricos provenientes das fotomultiplicadoras. Após o processamento pelo chip PACIFIC, o sinal é lido pelo módulo de Clusterização, onde o módulo PACIFIC é conectado diretamente.

O módulo de clusterização é responsável pela formação de clusters, possuindo 2 FPGAs que processam os dados provenientes do chip PACIFIC e os agrupa de forma otimizada para transmissão. Os FPGAs responsáveis pela clusterização recebem sinais de controle de um chip GBT SCA também contido na mesma placa.

O módulo Master é o responsável pela interface de comunicação entre os FPGAs de clusterização e o sistema de aquisição de dados. Ele contém um link óptico bidirecional utilizado para controle e oito links ópticos de saída utilizados para a transmissão dos dados. Os dados são serializados e transmitidos através de um conjunto de chips GBTx [17] a 4.8 Gbps no protocolo GBT WideBus.

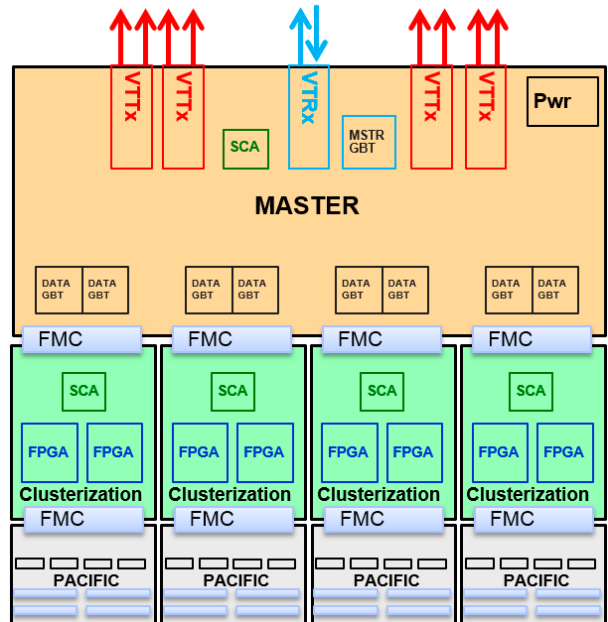


Figura 3: Diagrama ilustrando módulos que compõem a eletrônica de *front-end*.

3. O SISTEMA DE TESTES PARA O SCIFI

Após a etapa de produção em massa a eletrônica de leitura precisará ser submetida a um processo de controle de qualidade antes da etapa de instalação. Um total de aproximadamente 320 módulos de *front-end* (com 10% sobressalente) deverão ser produzidos. O objetivo de um sistema de teste automático é garantir a qualidade individual de cada *Read-Out Box* (ROB) em todos os postos de produção.

O sistema será capaz de executar as seguintes tarefas:

- detectar canais não funcionais;
- detectar canais ruidosos;

- verificar as funcionalidades do chip PACIFIC, como: uniformidade de ganho, características de *threshold*, DAC da tensão da SiPM, cancelamento de zeros e polos e a linha de base do *Shaper*;
- verificar *crosstalk* entre canais;
- detectar diferenças de fase intrínsecas de canais;
- verificar a integridade de fragmento de evento

Um diagrama funcional do sistema de teste é mostrado na Figura 4. Este consiste de três componentes principais em adição ao ROB sendo testado: um sistema de injeção, uma unidade de controle e aquisição, e um computador. O sistema de injeção consistirá de um gerador de sinais, capaz de emitir pulsos em todos os canais, variando o tempo de fase do sinal de entrada e a quantidade de carga injetada. O MiniDAQ (abordado na seção 3.1) será utilizado como unidade de controle e aquisição. O firmware dedicado do FPGA compreenderá ambos os blocos de TFC e ECS, e um software de controle WinCC proverá a interface para execução dos testes de parâmetros na *front-end* (*threshold*, temporização, etc.) e armazenará os dados no PC. Um pacote ROOT automático de análise proverá um diagnóstico completo de caracterização do ROB.

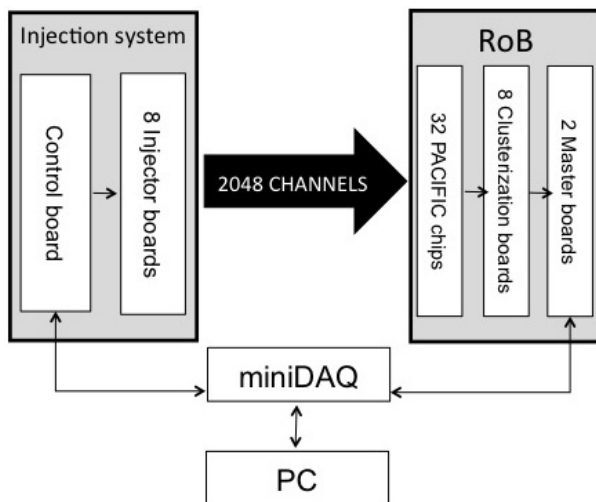


Figura 4: Diagrama de blocos do sistema de teste de ROB.

3.1. O MiniDAQ

O MiniDAQ (*Mini-Data Acquisition*) é um módulo de aquisição de dados composto por duas placas montadas em um frame metálico. A primeira, chamada AMC40, é usada para aquisição de dados e a segunda, chamada AMC_TP, usada para controle e interface da AMC40. O MiniDAQ é mostrado na Figura 5.

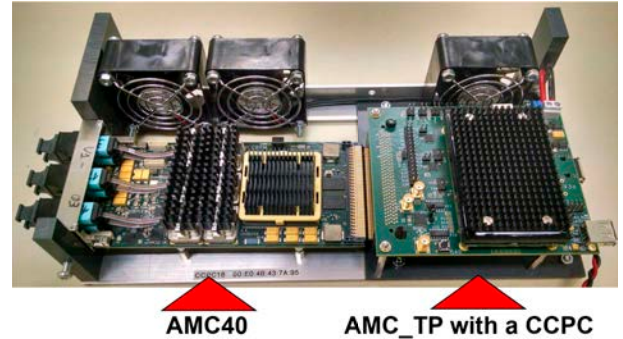


Figura 5: Caption

3.1.1. A placa AMC40

A AMC40 é a placa responsável pela aquisição de dados. Ela possui 36 links ópticos bidirecionais que operam até 10 Gbps. Destes 36 links, 24 são usados para aquisição de dados e controle da eletrônica de front-end enquanto os outros 12 são usados para a transmissão de dados para o computador via ethernet a 10 Gbps. Todos estes links são ligados a um FPGA da família Stratix V, da Altera. Além do FPGA se encontra uma memória DDR3.

3.1.2. A placa AMC_TP

A AMC_TP é a placa responsável pelo controle da AMC40 assim como por fornecer a interface com o usuário, visto que todo acesso ao FPGA da AMC40 é feito através da AMC_TP. Esta serve também como fonte de alimentação para a AMC40, convertendo níveis da entrada de 12V para os níveis usados pela AMC40, e como sistema de clock para a AMC40 gerado a partir de oscilador interno de 80,158MHz ou clock externo. A AMC_TP contém um CCPC (Credit Card PC) previamente configurado para fazer boot pela rede e acessível via SSH. Os desenvolvedores fornecem juntamente uma imagem de sistema operacional com Scientific Linux 6.2 instalado. O servidor de boot deve ser configurado pelo usuário. A placa contém um conector AMC no qual é inserido a AMC40. Todo o controle da AMC40 é feito a partir do CCPC. Através dele, com a biblioteca de software fornecida pela colaboração, é possível ler e escrever em registradores no FPGA da AMC40, assim como executar programas de configuração e controle da AMC40. Na AMC_TP também é executado o driver para interface entre o hardware de aquisição (AMC40) e o sistema SCADA utilizado.

3.1.3. Bloco SOL40_SCA do firmware do MiniDAQ

O bloco SOL40_SCA é a parte do firmware do MiniDAQ que recebe os comandos do CCPC referente ao Slow Control e os envia para a front-end através do link GBT.

Ele interpreta comandos vindos do CCPC, e gera e insere na palavra GBT a ser enviada pelo SOL40 os bits correspondentes a estes comandos que serão executados nos GBT

SCAs contidos na front-end. A palavra GBT enviada pelo SOL40 para o GBT Master contém vários conjuntos de 2 bits chamados de e-links. Cada GBT SCA da front-end é ligado a um destes e-links e é através destes que recebe comandos.

Os comandos podem significar escrita ou leitura de registradores de configuração dos GBT SCAs ou execução de comandos referentes aos periféricos contidos no GBT SCA, como leitura do ADC, envio de dados via I2C, etc. Mesmo os comandos aos periféricos são na verdade comando de escrita em registradores que por sua vez são interpretados pelo GBT SCA. Isto significa que todos os comandos passados ao GBT SCA podem ser simplificados em comandos de escrita em registradores.

Na front-end existem também componentes com os quais nos comunicamos através das portas I2C do GBT SCA. Para estes casos nós enviamos um comando de escrita I2C determinando em qual porta o componente alvo está ligado, e os dados a serem enviados pela I2C variam de acordo com a configuração do protocolo do dispositivo. Um protocolo I2C típico, por exemplo, recebe 7 bits de endereço, seguido de 1 bit que define se o comando é de leitura ou de escrita, o número de bytes a serem lidos ou escritos e então os dados a serem enviados.

Para enviar um comando ao bloco SOL40_SCA se deve fazê-lo através de seus registradores. Primeiro se escreve em um conjunto de 6 registradores o comando que se deseja enviar, e após isto se escreve em um bit específico de um registrador de controle que engatilha a leitura do comando contido nos 6 registradores mencionados. Após a leitura, o bloco SOL40_SCA processa o comando e inicia o processo de formação das palavras GBT para envio do comando aos dispositivos da front-end.

Os registradores onde se escreve o comando são 6 registradores de 32 bits, dos quais os dois primeiros definem o comando e os demais são reservados para dados. Os registradores para dados são utilizados em casos onde o comando exige uma carga de dados, como por exemplo escrita de dados através da porta I2C, JTAG e etc.

Uma ilustração destes registradores pode ser vista na Figura 6.

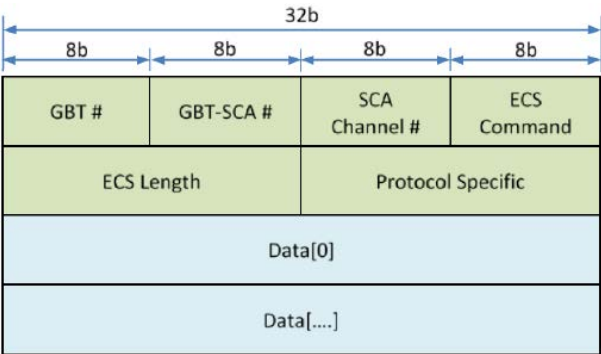


Figura 6: Ilustração dos campos contidos nos registradores de comando do bloco SOL40_SCA.

Cada linha da Figura 6 corresponde a um registrador. No momento da elaboração deste texto são utilizados 6 registradores, portanto a linha Data[....] corresponderia a 3 linhas adicionais, formando 4 registradores de dados (Data[0]

ao Data[3]), além dos 2 registradores contendo o comando. Uma breve explicação dos campos podem ser encontradas a seguir:

- GBT# é um número de 8 bits que corresponde ao número do link GBT através do qual se deseja enviar o comando. Como cada link é ligado a um único GBT Master, que por sua vez é único por front-end, na prática este campo define para qual front-end o comando está sendo enviado. No caso do sistema de teste do SciFi será utilizado sempre um único link ligado a uma única front-end, portanto este campo é sempre 0. Mas o bloco SOL40_SCA foi desenvolvido considerando a possibilidade de gerenciar diversas front-ends através de diversos links GBT.
- GBT-SCA# é um número de 8 bits que se refere a qual GBT SCA se deseja enviar o comando. Como no sistema de chips GBT cada GBT SCA é ligado a um único e-link, na prática este número seleciona para qual e-link do GBT Master o comando será enviado.
- SCA Channel# é um número de 8 bits que se refere a qual canal do GBT SCA será enviado o comando. Note que internamente ao GBT SCA existem vários blocos periféricos e um bloco de configuração, e são todos referidos como canais. Este número define, por exemplo, se o comando é para escrita em um registrador interno de configuração, ou para registradores relacionados aos periféricos como ADC, transceptor I2C e etc.
- ECS Command é um número de 8 bits que se refere ao tipo de comando a ser enviado. Os comandos disponíveis estão listados na documentação do GBT SCA ?citeSCA. É neste campo que se define, por exemplo, se um comando de acesso à porta I2C é de escrita ou leitura.
- ECS Length é um número de 16 bits que corresponde ao volume de dados (em bytes) contidos nos registradores de dados. Com 4 registradores de dados, o volume total é de 16 bytes. Quando este campo especifica um volume de dados inferior ao máximo, o conteúdo adicional é ignorado.
- Protocol Specific é um campo reservado que depende do protocolo utilizado no GBT SCA.
- Data[0..3] corresponde aos registradores que contém os dados a serem enviados junto com o comando, quando o comando é um do tipo que requer dados. Caso contrário estes registradores são ignorados.

Com os registradores cujo o conteúdo foi explicado acima é possível escrever o comando em que se deseja enviar para o bloco SOL40_SCA.

Após escrito o comando para acionar o envio deve-se escrever '1' no primeiro bit do registrador de controle.

Vemos uma ilustração do registrador de controle na Figura 7.

Adicionalmente, através do registrador de controle também se pode ver o status de resposta para os comandos

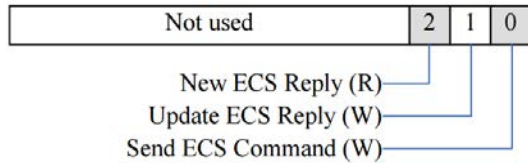


Figura 7: Registrador de controle utilizado pelo bloco SOL40_SCA.

de leitura, cujos dados de resposta são guardados também em um conjunto de 6 registradores. O bit 2 indica quando há uma nova resposta a ser lida, e quando já foi efetuada a leitura, pode-se limpar o registrador para a próxima resposta escrevendo '1' no bit 1 do registrador de controle.

4. SISTEMA DE CONTROLE

Para desenvolvimento do sistema de controle utilizamos o software SCADA (*Supervisory Control and Data Acquisition*), fornecido pela Siemens, chamado WinCC. Este já é um sistema amplamente utilizado no CERN e responsável pelo controle do LHC e seus quatro principais experimentos.

O WinCC é usado para se conectar a dispositivos de hardware do nosso sistema, adquirir dados e utilizá-los para supervisão, ou seja, monitorar o comportamento dos dispositivos do sistema e inicializá-los, configurá-los e operá-los.

4.1. Arquitetura de um Sistema WinCC

A arquitetura de um sistema WinCC é distribuída e modularizada. Um sistema WinCC é composto por diversos processos, chamados de "Managers" na nomenclatura WinCC. Um sistema típico WinCC é composto pelos seguintes *managers* (ilustrados na Figura 8):

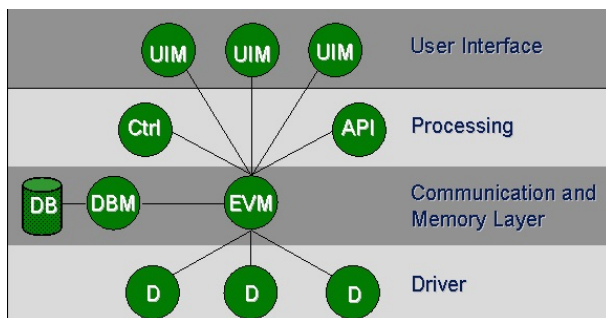


Figura 8: Estrutura de *managers* de um projeto em WinCC.

- **Event Manager (EVM):** É o núcleo do sistema e um *Manager* essencial, ou seja, requisito para funcionamento do sistema. Este é responsável por toda a comunicação entre os diferentes *managers*. Ele recebe dados dos drivers (D) e armazena no banco de dados (DB). Outros *managers* podem solicitar dados

ao *Event Manager*, que busca no banco de dados ou solicita aos Drivers, mas não podem se comunicar entre eles, ou seja, toda comunicação é feita através do *Event Manager*.

- **Database Manager (DBM):** É o processo responsável por gerenciar o banco de dados e que fornece a interface para que outros *managers* possam acessá-lo.
- **User Interface Manager (UIM):** São processos criados que fornecem uma interface gráfica para o usuário, no formato de painéis, que podem ser criados pelo próprio WinCC. Através destes painéis o usuário pode monitorar visualmente e atuar nos componentes do sistema.
- **Drivers (D):** Os drivers são processos que fornecem a interface entre o sistema e os dispositivos a serem controlados. Estes são específicos para cada dispositivo e precisam ser fornecido pelo fabricante do aparelho ou criado de forma personalizado.

O *Database Manager* (DBM) junto com o *Event Manager* (EVM) são os requisitos mínimos para um sistema WinCC poder operar. Sem estes dois processos sendo executados, nenhum outro processo funcionará. Os *managers* são todos processos independentes que se comunicam por protocolo TCP/IP. Mesmo quando sendo executados na mesma máquina a comunicação entre quaisquer dois managers é realizada por TCP/IP. Isto permite a distribuição do sistema por diferentes computadores. Desta forma podemos usar os painéis de controle (*Manager* UIM) de qualquer computador da rede, e ainda os Drivers que fazem a interface com os dispositivos a serem controlados não precisam estar no mesmo computador onde está o core do sistema, permitindo o uso de diversos computadores dedicados para interface de cada dispositivo.

No sistema de controle utilizaremos dois *managers* customizados: O Driver, pois por utilizarmos uma placa de fabricação própria do grupo, foi necessária criação também do componente driver, e os painéis com a interface gráfica de usuário.

Para o sistema de controle foi criado um novo projeto WinCC. Os *Event Manager* e *Database Manager* usados são os gerados por padrão na criação do projeto. Estes Managers, ou seja, o núcleo do projeto, ficará em execução em um computador que está ligado diretamente ao CCPC do MiniDAQ por cabo Ethernet. Este computador será usado para controle e aquisição de dados. O CCPC é ligado à AMC40 por interface PCIexpress. Para interface do nosso dispositivo, neste caso a AMC40, com o restante do sistema é necessário um componente Driver. O Driver utilizado é um processo chamado CCSERV que é executado no CCPC, acessa a AMC40 via PCIexpress e se comunica via Ethernet com o computador de controle fazendo uso do protocolo DIM. A Figura 9 mostra um diagrama ilustrando os componentes CCSERV e *Event Manager* executados no CCPC e no PC de controle respectivamente.

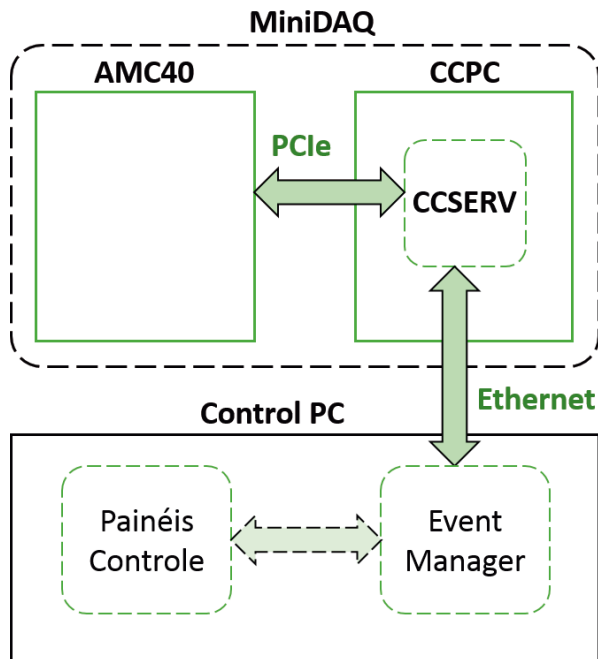


Figura 9: Diagrama de controle do sistema de teste.

4.2. Desenvolvimento das Bibliotecas de Acesso à Front-End

A biblioteca fwlbSCIFI foi criada para fornecer ao usuário funções de acesso aos registradores que se encontram na front-end. Ela utiliza o Framework LHCb para acessar os registradores do firmware do MiniDAQ e através destes enviar comandos ao bloco SOL40_SCA, a partir do qual é possível controlar os registradores da front-end através do link GBT conectado à front-end.

O objetivo final da biblioteca fwlbSCIFI é provêr funções para abstrair o procedimento de envio de comandos para a front-end, de modo que o usuário final precise apenas ler e escrever nos *datapoints* do sistema, com as funções e controles gráficos nativos do WinCC. Um *manager* utilizando a biblioteca fwlbSCIFI se encarregaria da interface com o bloco SOL40_SCA.

Até a data de elaboração deste documento a biblioteca fwlbSCIFI compreendia 9 funções, das quais cinco foram criadas com o intuito de serem usadas internamente pelas funções da API, que ficam disponíveis ao usuário. Estas cinco funções internas são:

- SOL40SCA_SetCommand
- SOL40SCA_Go
- SOL40SCA_SetReplyAddress
- SOL40SCA_NewReply
- SOL40SCA_GetReply

4.2.1. Funções de acesso direto ao bloco SOL40_SCA.

As funções com o prefixo “SOL40SCA” são aquelas que atuam diretamente nos registradores do bloco SOL40_SCA. Estas funções, a princípio, não precisam ser utilizadas pelo usuário final, uma vez que são chamadas pelas demais funções.

A função SOL40SCA_SetCommand recebe como argumento uma string em hexadecimal. A função retornará um aviso caso contenha um caractere não hexadecimal e será interrompida. Esta função apenas verifica os dados, os divide em 6 pedaços de 32 bits e os escreve nos 6 registradores de comando do bloco SOL40_SCA. São os registradores representados na Figura 6.

A função SOL40SCA_Go é responsável por dizer ao bloco SOL40_SCA que há um novo comando a ser executado. Na prática ela escreve '1' no bit 0 do registrador de controle do bloco SOL40_SCA, que é um bit apenas de escrita. Como se pode ver na Figura 7, escrever neste bit engatilha o envio do comando ECS. Para cada novo comando a ser enviado, deve-se escrevê-los nos registradores e após isto acionar o envio com a escrita de '1' no registrador de controle.

Para cada comando enviado ao bloco SOL40_SCA é gerada uma resposta. Esta resposta é armazenada em uma memória e o formato é semelhante à palavra de comando. Os primeiros 32 bits se referem ao endereço do comando, que contém informação do número do Master GBT, SCA e canal I2C para o qual foi enviado o comando. A resposta é lida a partir de 6 registradores de 32 bits, mas não fica imediatamente disponível. Supondo que foram enviados dois comandos para dispositivos I2C diferentes, com tempos de resposta diferentes, é possível selecionar qual resposta será visualizada primeiro, filtrando o valor de saída da memória onde as respostas são guardadas. Portanto, caso sejam enviados múltiplos comandos, pode-se esperar a resposta de um dispositivo específico cujo endereço corresponda ao comando enviado.

Para selecionar de qual dispositivo se deseja receber a resposta, utiliza-se a função SOL40SCA_SetReplyAddress. Ela seleciona o endereço de onde provém a resposta a ser lida pela memória. Isto ocorre escrevendo o valor desejado no registrador de endereço da memória. Esta função recebe uma string hexadecimal de 32 bits, interpreta e escreve no registrador mencionado.

Após escrever o endereço do comando cuja resposta se deseja ler, esta não se torna imediatamente disponível nos 6 registradores. É necessário solicitar uma nova resposta ao bloco SOL40_SCA. Isto é feito escrevendo '1' no bit 1 do registrador de controle. O bit 1 é um bit somente de escrita e é responsável por atualizar o valor dos 6 registradores com o último comando recebido correspondente ao endereço do registrador de endereço.

A função SOL40SCA_NewReply escreve '1' no bit 1 do registrador de controle a fim de solicitar nova resposta. Quando há uma nova resposta disponível, o bit 2 do registrador de controle fica ativado. Solicitar nova resposta nesta ocasião resetará o referido bit, indicando que a resposta do comando foi lida e não há outra disponível.

Após verificar a chegada de uma resposta e solicitá-la com a função SOL40SCA_NewReply, pode-se ler a resposta uti-

lizando a função `SOL40SCA_GetReply`. É possível ler a resposta completa acessando diretamente os registradores onde a resposta é armazenada, mas para fins de simplificação, a função `SOL40SCA_GetReply` faz isto pelo usuário. Ela lê a resposta e retorna uma string hexadecimal correspondente ao valor da resposta.

Estas são as funções internas da biblioteca `fwlbSCIFI`. Estas funções não foram criadas com o intuito de serem chamadas diretamente pelo usuário, mas são disponibilizadas da mesma forma que as outras para fins de desenvolvimento e teste.

4.2.2. Funções de envio de comandos à eletrônica de front-end.

As quatro funções restantes têm intuito de criar uma camada de abstração do processo de envio de comandos para a front-end, visando simplificar a utilização do sistema de controle. As referidas funções:

- `read_from_SCA`
- `write_to_SCA`
- `read_I2C_device`
- `write_I2C_device`

As funções de leitura `read_from_SCA` e `read_I2C_device` são utilizadas para ler um registrador de um chip GBT SCA ou de um dos dispositivos conectados aos canais I2C do GBT SCA, respectivamente. As funções `write_to_SCA` e `write_I2C_device` são utilizadas para escrever em algum dos registradores ou enviar comandos para os mesmos chips.

a. Funções de acesso ao SCA. Para ler ou escrever nos registradores do GBT SCA ou enviar comandos, as funções `read_from_SCA` e `write_to_SCA` são utilizadas. Ambas recebem dois parâmetros em comum, que é o número do link GBT o qual o bloco SOL40 deve utilizar para comunicação com a front-end, e o datapoint referente ao registrador que se deseja escrever.

A função de escrita recebe um parâmetro adicional, que são os dados a serem enviados, no formato de string hexadecimal. A função de leitura, por sua vez, retorna os dados solicitados, também no formato de uma string hexadecimal.

Ambas as funções verificam os endereços dos dispositivos a serem utilizados, e a partir destes criam a string de comando a ser passada para o bloco SOL40_SCA. Como exemplo, suponhamos que o registrador que se deseja acessar seja o “Control Register A” (CRA) do canal Controller do chip SCA Master. O datapoint correspondente seria:

```
GBT_Master0.SCA_Master.Device
    .Channel.Controller.Registers.CRA
```

O endereço do GBT Master será o valor do datapoint:

```
GBT_Master0.Device.config.address
```

O endereço do GBT SCA será o valor do datapoint:

```
GBT_Master0.SCA_Master.Device.config.address
```

O número do canal utilizado do GBT SCA será:

```
GBT_Master0.SCA_Master.Device.Channel
    .Controller.config.address
```

Após separar cada um destes valores, a função monta a palavra de comando baseada no formato descrito na seção 3.1.3, conforme se vê na Figura 6, e envia o comando ao bloco SOL40_SCA utilizando as funções `SOL40SCA_SetCommand` e `SOL40SCA_Go`.

Caso seja a função de leitura, após a função “Go” disparar o envio do comando, esta ainda realiza a leitura da resposta do comando, executando as funções “SetReplyAddress”, “NewReply” e “GetReply”.

b. Funções de acesso aos dispositivos I2C. Para ler ou escrever nos registradores dos dispositivos conectado em uma das portas I2C do GBT SCA, utiliza-se as funções `read_I2C_device` e `write_I2C_device`.

Assim como as funções para acesso ao SCA, ambas também recebem os parâmetros de link GBT, o qual o bloco SOL40 deve utilizar para comunicação com a front-end, e o datapoint referente ao registrador que se deseja escrever.

A função de escrita recebe um parâmetro adicional, que corresponde aos dados a serem enviados, no formato de string hexadecimal. A função de leitura, por sua vez, retorna os dados solicitados, também no formato de uma string hexadecimal.

De forma semelhante às funções de acesso ao GBT SCA, estas também verificam os endereços dos dispositivos a serem utilizados, mas adicionando também o endereço I2C do registrador final a qual o comando é direcionado.

4.3. Painéis de Controle

Para monitoramento e controle do sistema pelo usuário, utilizamos painéis desenvolvidos com a ferramenta GEDI (*Graphical Editor*) do WinCC. Estes painéis podem ser acessados pelo próprio computador de controle, como também de algum outro computador na rede, contanto que o computador onde o painel será executado e o computador que hospeda o projeto WinCC estejam conectados pela rede.

Na Figura 10 se pode encontrar um painel criado para suporte à utilização e teste do bloco SOL40_SCA. Ele utiliza as funções de acesso direto ao bloco SOL40_SCA para envio de comando e monitoramento das respostas.

Na parte superior se pode ver um campo longo de texto chamado “To be sent:”, que é onde se escreve o comando a ser enviado. Ao clicar no botão “Set Command”, a função `SOL40SCA_SetCommand` é chamada com o parâmetro do campo de texto. A função irá interpretar a string e escrever nos 6 registradores de comando do bloco SOL40_SCA.

O seis campos mais curtos de texto, logo abaixo do botão “Set Command”, são os valores reais dos registradores de comando do bloco SOL40_SCA.

O botão “Go” chama a função de mesmo nome, da biblioteca `fwlbSCIFI`, que envia o comando à front-end.

A resposta do comando se encontrará então disponível para leitura nos campos de texto da parte inferior do painel.

Há uma caixa de texto no centro do painel, onde se lê “Reply ready!” com fundo amarelo, que corresponde ao bit que indica se há uma nova resposta referente ao comando contido no registrador de endereço do bloco. Pode-se definir o valor do registrador de endereço do SOL40_SCA através dos pequenos campos de texto marcados pela borda com o texto “Reply Address”.

Este painel foi essencial durante o desenvolvimento do sistema e também do teste do firmware SOL40_SCA, e é muito útil para depuração uma vez que o usuário tem acesso direto ao comando enviado ao bloco SOL40_SCA.

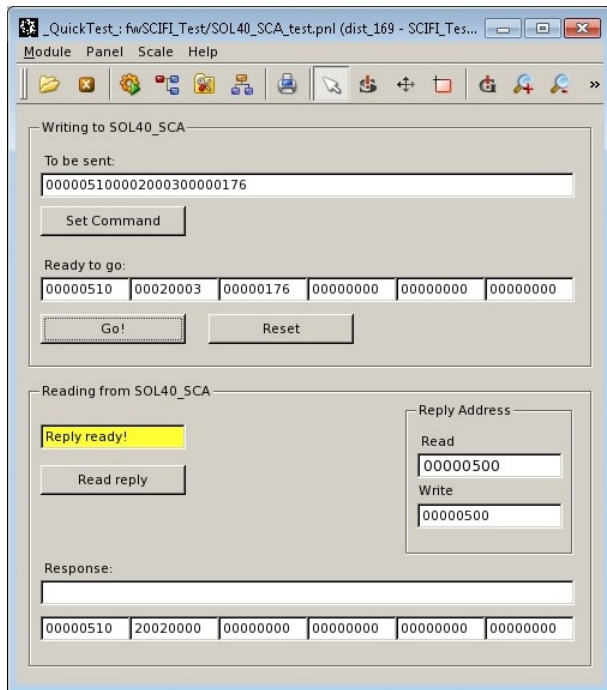


Figura 10: Painel desenvolvido para envio de comandos ao bloco SOL40_SCA.

5. CONCLUSÃO

O desenvolvimento de um sistema de controle em WinCC para a eletrônica de *front-end* do SciFi foi realizado com sucesso. Embora ainda não estejam prontos todos os componentes que integrarão a *front-end* e o sistema de testes, as funções desenvolvidas foram testadas com placas de desenvolvimento contendo o chip GBT e, através das funções da biblioteca fwlbSCIFI, foi possível configurar os chips GBTx e GBT-SCA.

Semanas após o desenvolvimento do sistema aqui descrito, os primeiros protótipos da *Master Board* ficaram prontos e o sistema foi crucial para os testes dos protótipos, uma vez que este era de fato a única ferramenta capaz de se comunicar com todos os dispositivos presentes. A biblioteca fwlbSCIFI e os painéis criados para o sistema de controle permitiram ao grupo responsável pelo desenvolvimento da *Master Board* a realização dos testes de validação dos primeiros protótipos da *Master Board*.

O trabalho aqui apresentado é um sistema de controle preliminar. À medida que os demais componentes do sistema de teste e da *front-end* fiquem prontos, a intenção é que mais painéis sejam criados e as funções de controle também se estendam para o módulo gerador de sinais e às funções de software do computador que receberá e analisará os dados.

-
- [1] L. Evans and P. Bryant. Lhc machine. *JINST*, 3:S08001, 2008.
 - [2] ATLAS Collaboration, G. Asd, et al. The atlas experiment at the cern large hadron collider. *JINST*, 3:S08003, 2008.
 - [3] CMS Collaboration, S. Chatrchyan, et al. The cms experiment at the cern lhc. *JINST*, 3:S08004, 2008.
 - [4] LHCb Collaboration, A. Augusto Alves Jr, et al. The lhcb detector at the lhc. *JINST*, 3:S08005, 2008.
 - [5] ALICE Collaboration, K. Aamodt, et al. The alice experiment at the cern lhc. *JINST*, 3:S08002, 2008.
 - [6] TOTEM Collaboration, G. Anelli, et al. The totem experiment at the cern large hadron collider. *JINST*, 3:S08007, 2008.
 - [7] LHCf Collaboration, O. Adriani, et al. The lhcf detector at the cern large hadron collider. *JINST*, 3:S08006, 2008.
 - [8] LHCb Collaboration et al. Lhcb rich. Technical report, European Organization for Nuclear Research - CERN, 2000.
 - [9] LHCb Collaboration et al. Lhcb calorimeters. Technical report, European Organization for Nuclear Research - CERN, 2000.
 - [10] LHCb Collaboration et al. Lhcb muon system. Technical report, European Organization for Nuclear Research - CERN, 2001.
 - [11] Cian O’Luanaigh. New technologies for the High-Luminosity

- LHC. Nov 2015.
- [12] Letter of Intent for the LHCb Upgrade. Technical Report CERN-LHCC-2011-001. LHCC-I-018, CERN, Geneva, Mar 2011.
 - [13] I Bediaga, J M De Miranda, F Ferreira Rodrigues, J Magnin, A Massafferri, I Nasteva, A C dos Reis, S Amato, K Carvalho Akiba, L De Paula, O Francisco, M Gandelman, A Gomes, J H Lopes, J M Otalora Goicochea, E Polycarpo, M S Rangel, B Souza De Paula, D Vieira, C Gobel, J Molina Rodriguez, and et al. Framework TDR for the LHCb Upgrade: Technical Design Report. Technical Report CERN-LHCC-2012-007. LHCb-TDR-12, CERN, Geneva, Apr 2012.
 - [14] LHCb Trigger and Online Upgrade Technical Design Report. Technical Report CERN-LHCC-2014-016. LHCb-TDR-016, CERN, Geneva, May 2014.
 - [15] Luis Lopez March and Matthias Karacson. Radiation studies for the LHCb tracker upgrade. Technical Report LHCb-PUB-2014-022. CERN-LHCb-PUB-2014-022. LHCb-INT-2013-003, CERN, Geneva, Feb 2014.
 - [16] Gbt project home page. Disponivel em: <https://espace.cern.ch/GBT-Project/default.aspx>. [Acessado em 13/Maio/2016].
 - [17] Giulia Papotti. *Architectural Studies of a Radiation-hard Transceiver ASIC in 0.13 um CMOS for Digital Optical Links in High Energy Physics Applications*. PhD thesis, Universita Degli Studi di Parma, 2007.
 - [18] A Gabrielli, S Bonacini, K Kloukinas, A Marchioro, P Moreira, A Ranieri, and D De Robertis. The GBT-SCA, a radiation tolerant ASIC for detector control applications in SLHC experiments. 2009.
 - [19] Jose Mazorra De Cos. PACIFIC : The readout ASIC for the SciFi Tracker planned for the upgrade of the LHCb detector. Sep 2015.
 - [20] A Affolder, B Allongue, G Blanchot, F Faccio, C Fuentes, A Greenall, and S Michelis. Dc-dc converters with reduced mass for trackers at the hl-lhc. *Journal of Instrumentation*, 6(11):C11035, 2011.
 - [21] G Vouters, F Alessio, J Cachemiche, S Cap, C Drancourt, P Durante, P Duval, L Fournier, M Jevaud, F Hachon, J Mendez, F Rethore, and S. T'Jampens. LHCb upgrade MiniDAQ Handbook. Technical report, CERN, Geneva, Nov 2013.
 - [22] Tektronix. Understanding Oscilloscope Bandwidth, Rise Time and Signal Fidelity. Technical Report 55W-18024-3.
 - [23] LHCb Collaboration et al. Lhcb vertex locator. Technical report, European Organization for Nuclear Research - CERN, 2001.
 - [24] LHCb Collaboration et al. Lhcb outer tracker. Technical report, European Organization for Nuclear Research - CERN, 2001.
 - [25] LHCb Collaboration et al. Lhcb inner tracker. Technical report, European Organization for Nuclear Research - CERN, 2002.
 - [26] LHCb Collaboration et al. Lhcb magnet. Technical report, European Organization for Nuclear Research - CERN, 2000.