

A Data Acquisition System for the Test Beam

During the ongoing process of the development of the DAQ system, the electronics has been tested in two test beams organised in Fall 2014 and Fall 2015. The first test beam, ran with the first prototype of the DAQ, aimed at proving the feasibility of the architecture of the system, not focusing on data taking from the provided pion and muon beam but rather on usability. As the results were encouraging, the second version of the electronics was developed involving a complete redesign of the hardware, firmware, and software. Therefore, we will mainly focus on the second test beam, which electronics is described in the previous chapter, during which abundant data was recorded showing both excellent results for the detectors and the DAQ system.

In this chapter, we present the firmware and software developments done for the DAQ system for the test beams followed by the analysis of the recorded data. First, we will present the firmware architecture of the OH and the GLIB to better understand the global layout of the system and the features that have been implemented to control the components. Then we will move on to the back-end applications developed to control and monitor the DAQ system and read out data. Finally, after a presentation of the layout of the test beam setup and its characteristics, we will show the results obtained after analysis of the recorded data.

1.1 Architecture of the OptoHybrid Firmware

The most important function of the OptoHybrid is to transfer data between the VFAT2s and the GLIB. Downwards, from the off-detector to the on-detector electronics, it transfers slow control requests and fast commands while upwards it sends the trigger and tracking data back to the back-end system. Next to the handling of the basic VFAT2 functionalities, it must also handle the optical links and a couple of programmable registers which control the system. Furthermore, procedures that were previously implemented in software and required extensive computational time have been moved to firmware in order to speed up the system.

Although rather complex, the firmware of the OptoHybrid can be decomposed in six blocks: the fast commands, the trigger data, the tracking data, the slow control, the calibration routines, and the optical links. Eventough these blocks separately will be presented, they are tightly interconnected using a wishbone-like architecture for intercommunication.

1.1.1 Internal Communication Through a Wishbone-Like Architecture

Wishbone is an open-source communication protocol used in many projects to enable data transfer between ICs. A light version of the protocol has been implemented in the OptoHybrid to link the various modules of the system. The latter are divided in two categories: masters which initiate requests, and slaves which provide responses. The link between the two is done through a switching hub which redirects the requests to the appropriate slaves by using a 32-bit-long address space mapped to the various modules. By design, multiple masters can interact simultaneously with different slaves allowing for parallelism in the system.

A request from a master is composed of four signals: a flag signaling the presence of a request, a write-enable bit to indicate the nature of the request (read or write), a 32-bit-long address to which the request will be redirected, and an optional 32-bit-long data field in case the request is a write operation. The response from a slave consists of: an acknowledgment signal, a 4-bit-long error status in case the operation failed, and an optional 32-bit-long data field holding the response to read requests. A programmable timeout has also been implemented to avoid blocking operation. Some components implement both a master and a slave module in order to be able to receive requests and propagate them to various other modules.

Communication done through the wishbone-like protocol is used mainly for slow control or non-time-critical operations as the latency of a transaction is not fixed. Indeed, the switching hub implements a waiting list functionality which allows two masters to address the same slave at the same time by storing one of the requests in memory and awaiting for the slave to finish the other transaction.

1.1.2 Encoding Fast Commands

Fast commands such as L1As, Resyncs, BC0s, etc can originate from various sources. In normal data taking runs, the AMC13 receives the TTC signals, forwards them to the microTCA AMC which in turn transmits them to the OptoHybrid. When data taking is stopped and calibration runs are performed, the routines implemented in the OptoHybrid are run and require fast commands. To this end, the TTC signals can also be generated locally using the T1 generator block. This entity can generate L1As, Resyncs, BC0s, and CalPulses in three different ways: send a single command a given number of times at fixed interval, send a CalPulse followed by a L1A with a fixed delay, or send a programmable pattern involving all the commands. Finally, the two remaining sources of TTC commands, and more precisely L1As, are either a loopback from a given VFAT2 trigger bits directly to all the VFAT2s (self-triggering) or the signals coming from an external component through a debugging header on

the OptoHybrid.

The switching between the various sources is done by setting a register through slow control operations. Once the source has been selected, the commands are forwarded to the VFAT2s and encoded on the T1 signals. Each command is composed of three bits clocked at 40 MHz. An additionnal feature has been added to the L1A line to throttle the trigger signals when working in high rate environments. Through registers, the operator can select to send only a fraction of the received commands to not overload the VFAT2 buffers and allow for correct readout of the chip.

1.1.3 Formatting Trigger Data

Each of the 24 VFAT2s transmits eight trigger bits per BX. These are regrouped using a logic OR in the DAQ system of the test beam thus yielding 24 bits. The trigger information is used for calibration routines and forwarded over debugging headers to an external electronics crate. As the number of external connections is limited, only six trigger bits can be send thus requiring the need for a selection mechanism controller by programmable registers.

1.1.4 Acquiring Tracking Data

Upon reception of a L1A, the VFAT2 transfers the selected event from its SRAM1 to SRAM2 and then to an encoder which serialises the data. Each event is 192 bits long followed by two idle bits and clocked at 40 MHz. This means that it takes 194 BXs to read out one event and that the VFAT2 cannot handle trigger rates higher than ≈ 200 kHz without experiencing an overflow of the buffers. The data is pushed out of the VFAT2 automatically and formatted according to the pattern shown in Table 1.1. The top left bits are Most Significant Bits (MSBs) which are pushed out first and the bottom right bits or the Least Significant Bits (LSBs) which come in last. The first four bits of the first three 16-bit-long words are constant values. These are completed by the BC which increments at each clock cycle, the EC which counts the number of received L1As, four flags which hold information on the status of the buffers, and a chip ID unique to each VFAT2. Following are the 128 bits reflecting the hit information on each channel. Finally, the VFAT2 uses a Cyclic Redundancy Check (CRC) on 16 bits to detect errors. The CRC uses all other 176 bits to encode its data but does not provide a way to correct errors.

Next to the data line, each VFAT2 provides a DataValid line which is pulled high when the bits coming out of the VFAT2 are valid. However, due to the limited number of pins connecting the GEB and the OptoHybrid, this signal has been left unconnected for all but six VFAT2s. Therefore, data is constantly shifted in a 194 bits (192 bits of data and 2 idle bits) serial-to-parrallel converter and analyzed. When the fixed pattern of 12 bits is seen, a flag is raised signaling the presence of potential data. The data packet is split up in its various elements and the CRC is recomputed

VFAT2 tracking data

15	0	
1010	BC[11:0]	
1100	EC[7:0]	Flags[3:0]
1110	Chip ID[11:0]	
Channel data x8		
CRC		

Table 1.1 – ??

and compared against the received CRC. Two additionnal flags respectively hold the results of the comparation of both CRCs indicating if the data is valid or not, and a logic OR of all 128 channel to indicate if the packet contains a hit or not. In theory, only packets with valid CRCs could be transmitted to the back-end electronics. However, sending all packets offer the possibility to identify recurring errors in the CRCs computation and offers the possibility to correct them offline.

In parallel to the decoding of the data packets, the OptoHybrid maintains its own BC and stores the value of the counter in a buffer each time a L1A is received. When the decoding modules report they have detected data, a concentrator module aggragates the information from all available VFAT2s. Every packet received within a window of 10 BX is assigned to the same event to which the value of the corresponding BC is appended. The assembled event is stored in a large buffer to be latter sent over the optical links to the GLIB.

To prevent positions either not equipped with VFAT2 Hybrids or that are noisy to generate fake data, a 24-bit-long register allows to mask individual positions to ignore any packets it generates. Next to this, each decoding module is equipped with two counter respectivly counting the number of valid and invalid packets.

1.1.5 Controlling and Monitoring the Systems

The OptoHybrid is used to control itself through wishbone and the VFAT2s through I2C. The slow control of the OptoHybrid mainly consists in selecting TTC command sources, setting the trigger throttling, reading out counters, etc. These operations are performed to control the data flow and the functioning of the DAQ system itself. Commands sent to the VFAT2s on the other hand have direct impact on the physics of the data taking through the biais of the analog front-end readout.

To communicate with the VFAT2s, six I2C controllers are implemented in the firmware of the OptoHybrid, one per sector on the GEB. Each controller can access

four VFAT2s which are identified using three resistors that can be installed on the GEB. VFAT2 uses a modified I2C protocol which uses the same frame format as the official one but a different addressing scheme. The official I2C data frame is shown in Figure 1.1. The master starts by sending seven address bits which are used to identify the slave it wants to talk to, followed by a read/write bit. After the slave acknowledged the request, eight bits are sent from the master to the slave in case of a write operation followed by a slave acknowledgment, or the opposite in case of a read operation. For the VFAT2, the first three bits of the address are used to select the chip that is addressed by the master while the four remaining bits are used to select the register that needs to be accessed. This addressing scheme allows the OptoHybrid to access up to 16 registers on the VFAT2s. However, each VFAT2 holds 16 primary registers and 136 extended registers. The extended registers are accessed using two primary pointer registers: one set to point to an extended register, and one to read/write the data in said register. Thus, in order to perform a transaction on a primary register, only one operation is necessary, while two are required to access extended registers.

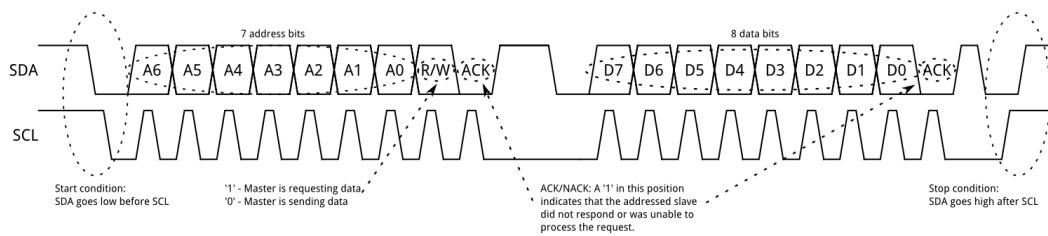


Figure 1.1 – ??? [1].

To make this process transparent to the user and flatten out the address space of the VFAT2s, the OptoHybrid maps the primary registers from addresses 0 to 16 and the extended registers from addresses 17 to 152. When performing a transaction with the extended registers, the I2C controllers automatically run the double addressing scheme.

Next to the basic I2C controllers, an extended I2C controller was developed to abstract individual addressing and allow request broadcasting. This module forwards I2C requests to all selected VFAT2s at once to configure the entire system in parallel. A programmable register allows to leave out given VFAT2s from the broadcast while a buffer stores the result of the operations.

1.1.6 Calibrating the Systems

The calibration routines have been ported from software to firmware in order to increase their speed of operation and reduce the number of requests that need to be performed. The routines are as follows: a threshold scan which measures the noise on the strips as a function of the threshold of the VFAT2s, a latency scan which allows

to select the correct BX when receiving L1As, and a s-curve scan which characterize the response of the strips as a function of the collected charge and threshold.

The threshold scan is used to scan each VFAT2 for noise. For each threshold value set on the VFAT2, the percentage of events displaying a hit is recorded and taken as the percentage of noise. A graph can be obtained showing the noise decrease as the threshold increases yielding a point at which the system can be operated with minimal noise. The threshold scan can be operated using trigger information on the 128 strips as a whole or on individual strips using tracking data. For the latter, the T1 generator is used as trigger source to generate data as these runs being performed when the beam is off. No relation between a L1A and physical event is needed to study the noise on the system.

The latency scan allows to determine the best latency value to be set in a VFAT2. The latency is the time difference, in number of BX, between the time of arrival of a L1A and the time at which the related event was stored in the VFAT2 buffer. This module is operated when the particle beam is on and triggers are generated by an external source such as a Photomultiplier (PM) placed in front of the detector. For each value of the latency, the OptoHybrid counts the ratio of events with hits over the total number of events. For a noiseless VFAT2 with 100% detection efficiency, the ratio would be 0% outside the correct latency window and 100% inside. The size of the window can be adjusted by changing the length of the monostables output in the VFAT2s.

The s-curve scan is part of the calibration routines performed for the qualification of the VFAT2s. It yields the response of the VFAT2 strips to an injected charge pulse according to the threshold. It is used in conjunction with the T1 generator which send a CalPulse followed by a L1A at fixed interval, thus with known latency. The use and results obtained for this module are further detailed in Chapter ??.

1.1.7 Optical Communication with the Off-Detector Electronics

The link between the OptoHybrid and the off-detector electronics are the optical links from which two are used: one for the trigger data and fast commands, and one for the tracking data and slow control. Each optical fibre is connected to an SFP+ transceiver which contains the Light Emitting Diode (LED) that drives the link. The SFP+ module is in turn connected to a Gigabit Transceiver X (GTX) block inside the FPGA which is a dedicated component that can handle high-speed serial communications.

The GTX module is in charge of the serialization/deserialization of high-speed data. A schematic diagram of the functions it provides is shown in Figure 1.2. The GTX

handles both the transmission and reception of data. From the FPGA logic, it receives a parallel vector of bits containing the raw data. The raw data enters the Transmit FIFO which acts as a buffer before the Line Encoding module. The latter uses an eight-bit/ten-bit (8b/10b) encoding scheming which is a standard in high-speed communications. 8b/10b encodes every eight bit word on ten bits according to a predefined table. It has the particularity that whatever combination of words might be sent, no more than five successive '0's or '1's will arise in the serial stream. This ensures line balancing: the fact that the electric lines between the GTX and the SFP+ module will remain at an average voltage and not drift to a logic '1' or '0' over time, thus requiring a longer fall or rise time of the signals. 8b/10b also offers a set of K-characters which are unique patterns of bit that can never arise randomly in the stream. Those characters are used to detect the beginning of a frame and allow the receiver to align correctly. Finally, after the encoding of the data frame is done, it is serialized and transmitted to the SFP+ module over differential pairs.

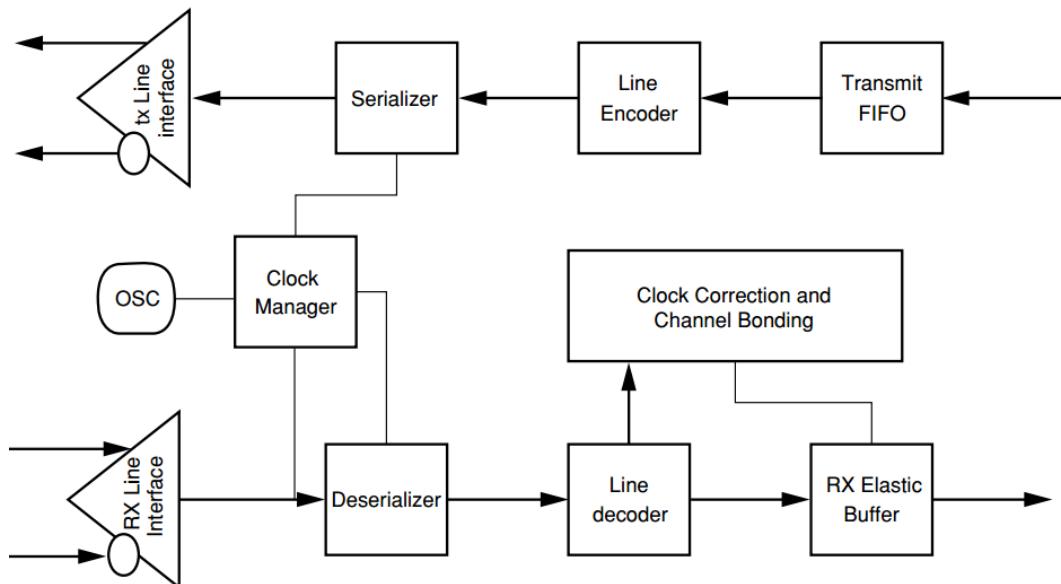


Figure 1.2 – ??? [2].

The receiver essentially follows the opposite path as the transmitter except for the Clock Management block. The 8b/10b encoding offers one more advantage which is the possibility to recover a clock from the data stream itself, technique called Clock Data Recovery (CDR). In high-speed signal design, small variations in the clock frequency between the transmitter and the receiver can lead to discrepancy in the read out data. To avoid this, the receiver can tune the frequency of the clock of its deserializer using the transitions of the bits in the incoming stream to align its phase to the data. The GTX block further provides the obtained recovered clock to the FPGA to use in the logic. Correct alignment of the data is checked using the K-characters in the stream.

GLIB to OptoHybrid frame		OptoHybrid to GLIB frame	
15	0	15	0
Header		Header	
Request address (MSB)		VFAT2 data x14	
Request address (LSB)		Response data (MSB)	
Request data (MSB)		Response data (LSB)	
Request data (LSB)		CRC	
CRC			

Table 1.2 – ??

The fixed latency link is used to transmit trigger data from the OptoHybrid to the GLIB and receive the fast commands. Of the 64 bits that the downlink transmits, 16 are used to align the serial bit stream on the receiving side and four others are used to encode the four fast commands that the VFAT2 understands. The other 44 bits are not used and left to constant zeros. The uplink uses a 16 bit header to align the stream to which it appends the 24 triggers bits coming from the logic-OR of the VFAT2s. These patterns repeat at 40 MHz as data changes every BX.

The variable latency link is used to transmit and receive slow control commands and to transmit the tracking data of one VFAT2 at the time to the GLIB. As only one VFAT2 was exposed to the beam at any given time during the test beam, reconstructing and sending the full event was not a necessity for the DAQ system. In both directions, a given frame format is constantly sent using two bits in the header to indicate the presence of a valid slow control request and valid VFAT2 tracking data in case of the uplink. The two data formats are shown in Table 1.2. Both send a 16 bit header to align the link followed by the frame data. As can be seen in the table on the left, the OptoHybrid receives requests from the optical link which are then converted to wishbone-like requests by the logic. The optical link module is one of the wishbone masters that controls the system and which responses are forwarded back to the GLIB.

1.1.8 System Summary

The aim of the OptoHybrid firmware is to provide a communication medium between the VFAT2s and the off-detector electronics and to speed up the calibration routines by implementing them closer to the detector. The developments that have been listed in this chapter are targeted at the OptoHybrid v2a and the configuration of the system ran during the test beams. Modifications have to be brought for the final system although the foundations of the firmware have been solidly laid out.

1.2 Architecture of the GLIB Firmware

The GLIB development group at CERN provides a system firmware which controls the onboard components, handles the communication with the MCH, and implements the logic to decode IPBus requests. This code is similar to the operating system in which the users runs their applications by adding custom firmware. The latter is system specific and up to the user to develop. In our application, the firmware we developed for the GLIB is in charge of the buffering of the tracking data until it is read out by the application, and the forwarding of the IPBus requests over the optical link to the OptoHybrid. Other than that, the GLIB holds a couple of counters which monitor the status of the optical links and the number of requests performed. However, the firmware of the GLIB has been designed to be able to handle two OptoHybrids in parallel, seemlessly allowing the user to switch from one to another at any time.

1.2.1 The Tracking Data Readout

The GLIB acts as buffer for the tracking data it reads out from the OptoHybrids and decodes from the optical links. It stores the VFAT2 events until the control and monitoring application reads them out through IPBus. In case the buffers reach their full occupancy, new events are discarded and flags are raised. The buffers can hold up to 16,383 VFAT2 events per OptoHybrid connected and the system has been tested and proven to work without overflows up until L1A rates of 200 kHz, the maximum rate the VFAT2s can handle.

1.2.2 The IPBus Controller and Slaves

The IPBus decoding procedure is much similar to the wishbone system implemented on the OptoHybrid. According to the address contained in the request, the GLIB selects the slave it needs to address. Each request is made of a flag signaling the presence of a request, a read/write bit reporting the type of operation, a 32-bit-long address field, and an optionnal 32-bit-long data field populated with data in case of a write operation. The IPBus response is composed of an acknoledgment and error flag to signal the validity of a response, and an optionnal 32-bit-long data field containing the read data. By design, each IPBus request is decomposed in single read or write operations. Therefore, even if the software performs a read request of multiple addresses at once, IPBus slaves will still see this as multiple sequential read operations on single addresses. This in term propagates to the OptoHybrid wishbone requests which originate from IPBus requests meaning that no more than one wishbone request coming from the GLIB is present at any time in the system. As a majority of the requests have to be forwarded to the OptoHybrid, an entire address space of the GLIB is mapped to the former. These are handled by an IPBus slave connected to the GTX blocks of the GLIB. This module encodes the request according to the data format on the left in Table 1.2 and awaits for a response before signaling the IPBus controller.

1.3 The Control and Monitoring Application

The first version of the control and monitoring application that communicated with the system was based on a series of Python scripts running PyChips, a Python implementation of IPBus provided with the GLIB source code. Each script was developed to run a specific task and needed to be ran over and over again. This was feasible for the first test beam during which only the system was tested for small amounts of time, but not for the second one that was meant to collect data. Therefore, we developed a web application based on NodeJS [3] as server technology, SocketIO [4] for the client/server exchange, AngularJS [5] as front-end framework, and Bootstrap [6] as design framework.

1.3.1 Architecture of the Application

NodeJS is a JavaScript (JS) runtime that has the ability to run JS code outside the context of a web browser. It extended JavaScript from its original use in web content to a wide range of applications by enabling it to manipulate files, data streams, etc. The most common use for NodeJS is the implementation of a web server that has the property to be much more reactive than other infrastructures. Where classic server structures display a blocking behavior, meaning they only handle one request at the time blocking the queue of incoming demands, NodeJS is event driven and thus non-blocking. For example, if a client requests a file, NodeJS will transfer the request to the operating system and continue to respond to other client as long as the file is not served. Interactions are thus more reactive and allow to easily implement a real-time control and monitoring system. Along side the NodeJS server that serves the content of our web application, we also implemented a version of IPBus in JS. The latter allows the server to execute IPBus requests on the GLIB and readout the response.

To link the client side controlled by the user and the server side which sends requests to the GLIB, the SocketIO project is used. It is an abstraction layer on the fairly new WebSocket technology which adds sockets to web browsers and allows for fast communication with servers by bypassing the HTTP protocol. When the client changes settings in its web browser, SocketIO transfers the request to NodeJS which in turn creates an IPBus packet. When the GLIB responded to the request, the server transfers the response back to the client which can handle the data as it likes.

To handle data in a dynamic and convenient way, the web pages use Bootstrap and AngularJS. Bootstrap is a design framework which provides simple and readable themes to layout the pages. The content of the page on the other hand is managed by AngularJS which is a front-end JS framework. It helps rendering dynamic content in function of the data received from the server. Actions can be triggered or disabled according to the state of the system, data can be updated live, etc. It renders the

user experience more fluid and enjoyable by providing a usable set of actions and an efficient navigation through the control pages.

1.3.2 Controlling the Systems

The control over the GLIB, OptoHybrids, and VFAT2s is done through four pages which screenshots are shown in Figure 1.3. The top-left picture displays the home page of the application which provides an overview of the status of each system. The left panel displays the firmware version of the GLIB and OptoHybrids; the middle panel gives the occupancy of the GLIB readout buffers, which are here signaled as full through the use of a red flag, and the status of the various modules of the OptoHybrid; finally, the right panel lists the VFAT2s that are turned on in green, turned off in red, and missing in gray. By clicking on any of the elements, the user is redirected to the corresponding system control page.

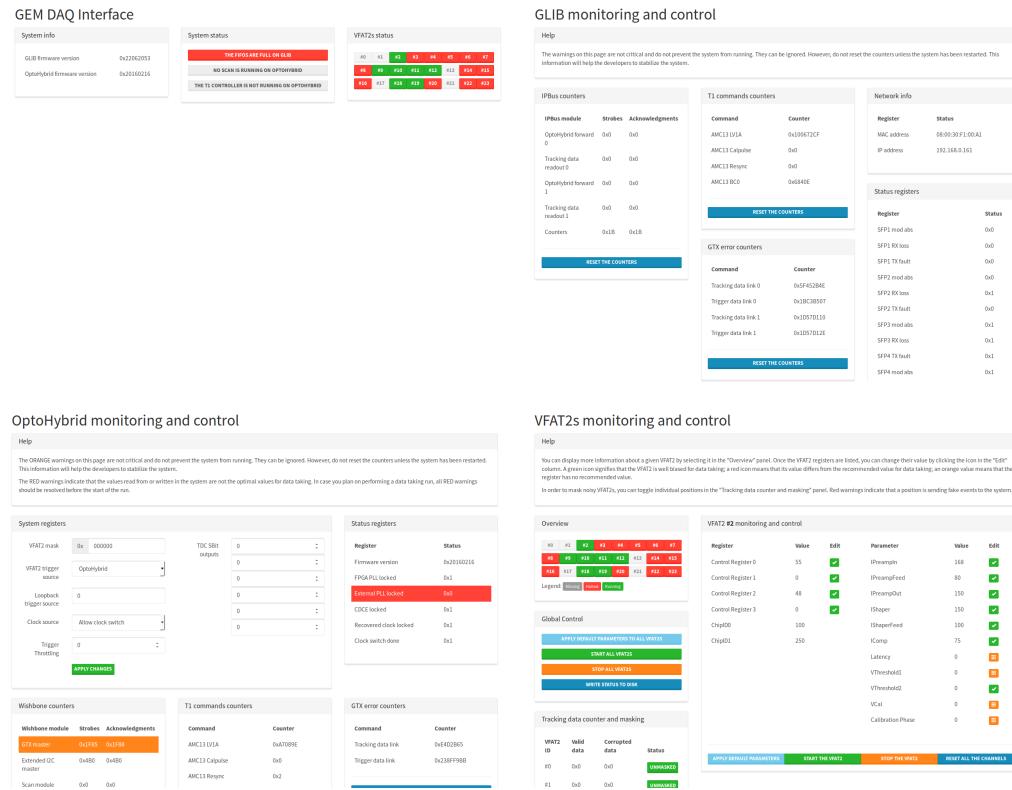


Figure 1.3 – ???

The control page of the GLIB is shown in the top-right corner. The information it contains is mainly related to counters which allow to see, for example, the number of requests sent to the OptoHybrid and the number of responds received. It also counts the number of errors detected on the optical link providing a quantitative value of the health of the link.

The OptoHybrid page, shown in the bottom-left corner, provides more information and control options to the user. The top-left part of the page is used to modify system register controlling the trigger source, clock source, trigger information sent over the debugging header, etc. This is where the user can change the type of ongoing run by going from a data taking run using external triggers to a calibration run using internally generated triggers. The panel directly on the right provides status information on the various clocking resources in the FPGA. By default, the OptoHybrid uses the onboard oscillator to power the system and the optical links. As communication is established with the GLIB, the GTXs provide the recovered clock from the link which is the system clock. When the latter is stable, the OptoHybrid can decide to switch from its onboard clock to the system clock in order to be synchronous with the whole DAQ system. This operation can be prohibited, forced, or allowed by the user, which can also choose to use an external clock provided through the debugging header. These clocking options are useful for the test beam setup where the clock can be provided externally to synchronize with other electronics equipment. Finally, the bottom part of the page displays counter values for the wishbone request, fast commands, link errors, etc. Note that when a discrepancy is observed between two correlated counters, a warning flag is raised as can be seen in the bottom-left of the picture.

Finally, the control page of the VFAT2s, shown in the bottom-right of the figure, gives access to the most significant registers of the front-end chips. It provides a summary of the on/off/absent status of each VFAT2 on the GEB and, by clicking on an individual slot, the value of each register as shown in the panel on the right. Default settings for the registers are saved in memory and compared to the current values, raising warnings when they differ and allowing the user to correct them. The user can also control all the VFAT2s at once by applying the default settings or turning them on or off at the same time. In the bottom panel, control buttons allow to mask individual VFAT2s in case they generate noise or send corrupted tracking data. Two counters monitor the number of valid (with a correct CRC) and invalid packets each position has sent.

1.3.3 Fast and Slow Control of the VFAT2s

The fast control or T1 generator is controlled by a dedicated page, displayed in the left of Figure 1.4, which allows to select the type of commands to generate, the pattern to send, and the delay between two pulses. The slow control of the VFAT2s is handled by the I2C modules and the extended I2C controller is managed by another page shown in the right. The slow control commands can either target a single VFAT2 as shown in the left panel or an entire set of chips as can be seen in the right panel. In case of a single command, the response is shown inline. For multiple operations, a table of results is shown indicating the value and the corresponding

VFAT2. To help the user, the accessible registers are named rather than listed using their address.

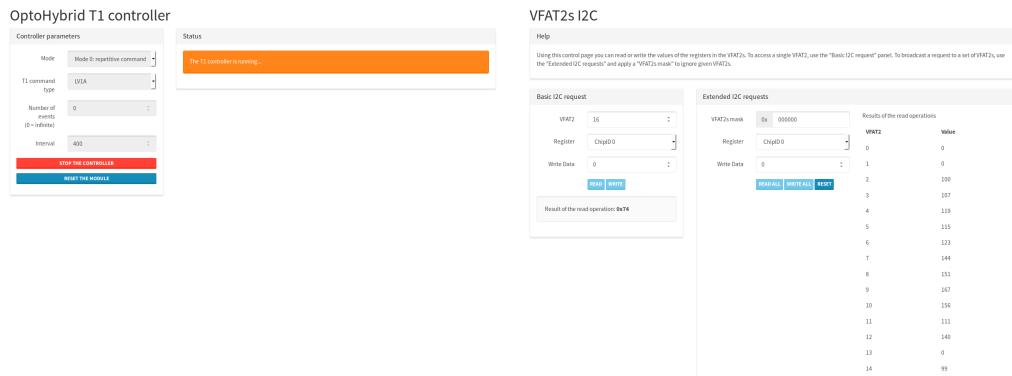


Figure 1.4 – ???

1.3.4 Calibrating the VFAT2s

The calibration of the VFAT2s is done through a dedicated page which regroups the various operations that can be performed on the chip. Figure 1.5 is a screenshot of the application running a threshold scan on one of the VFAT2s. The left panel allows the user to select all the required parameters such as the type of scan, the range of the scan, the number of events per point, etc. Once the parameters are entered, the user starts the scan and waits for it to finish in firmware. When done, a graph displays the results in the right panel. The user can then dynamically change VFAT2 settings by clicking on a point of the graph to, for example, select the most appropriate threshold or latency value. Furthermore, the user can save the data points to disk for offline analysis.

VFAT2 scans



Figure 1.5 – ???

1.3.5 Data Quality Monitoring

Data Quality Monitoring (DQM) is used online to check the consistency of data recorded. This is the last component of the application shown in Figure 1.6. This page monitors the status of the readout buffers of the GLIB and allows the user to acquire data. Once acquisition has started, plots begin to fill with the various parameters extracted from the VFAT2 data format. The user can visualise the number of data packets, the BC, EC, flags, and chip ID distribution, the occupancy of the strips, etc. The user can monitor the validity of the data and if required change the parameters of the run. Although this application has the ability to readout data and store it to disk, it was not used to this intend during the test beam. A XDAQ based program developed by the software development team of the GEM collaboration was used to acquire raw data from the GLIB and store it on disk in text files for later analysis.

Tracking data readout

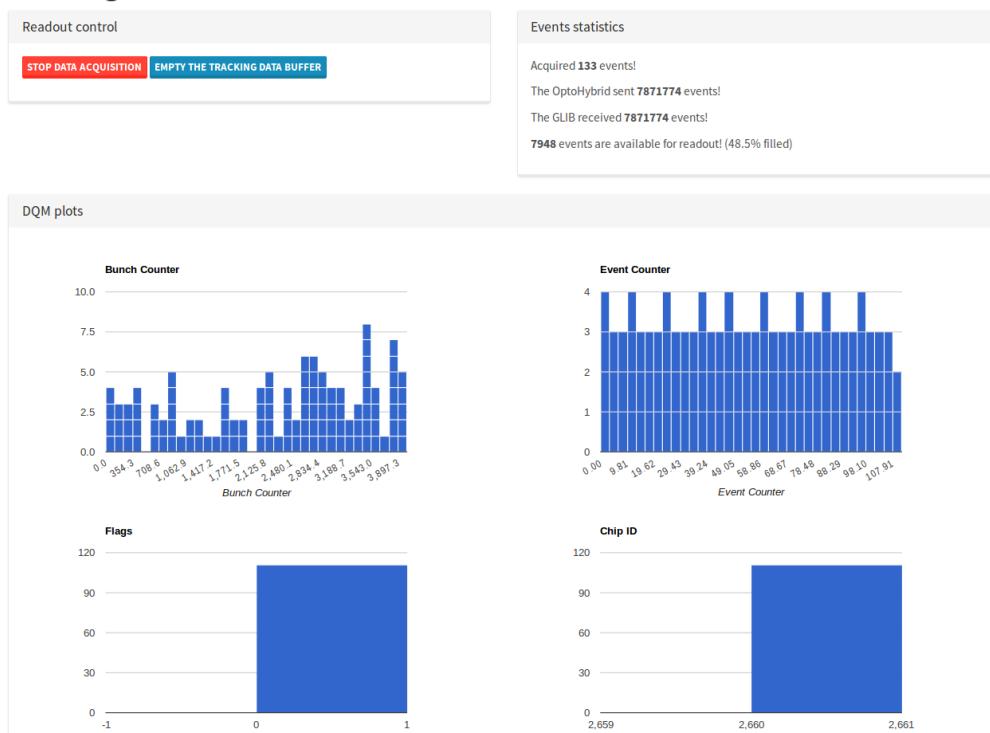


Figure 1.6 – ???

1.4 The Test Beam Setup

The test beam campaigns take place at CERN in the North Area of Prevessin, one of the secondary sites. The beam provided to the experiments originates from the SPS proton bunches which are extracted from the accelerator at 400 GeV and interact with four primary targets (T2, T4, T6, and T10) to produce the secondary beams mainly composed of electrons and hadrons. A selection on the type of particles and their momentum is performed and the beam is guided towards the experimental

setups, namely H2 and H4, using dipole magnets. In the transfer tunnels, collimators can be adjusted to increase or reduce the flux of particles and an optional secondary target can be placed in the beam path to create a tertiary beam. Figure 1.7 shows the monitoring page of the SPS indicating the presence of spills of particles in yellow. Spills last between 4.8 s and 9.6 s and repeat every 14 s to 48 s depending on the SPS operation.

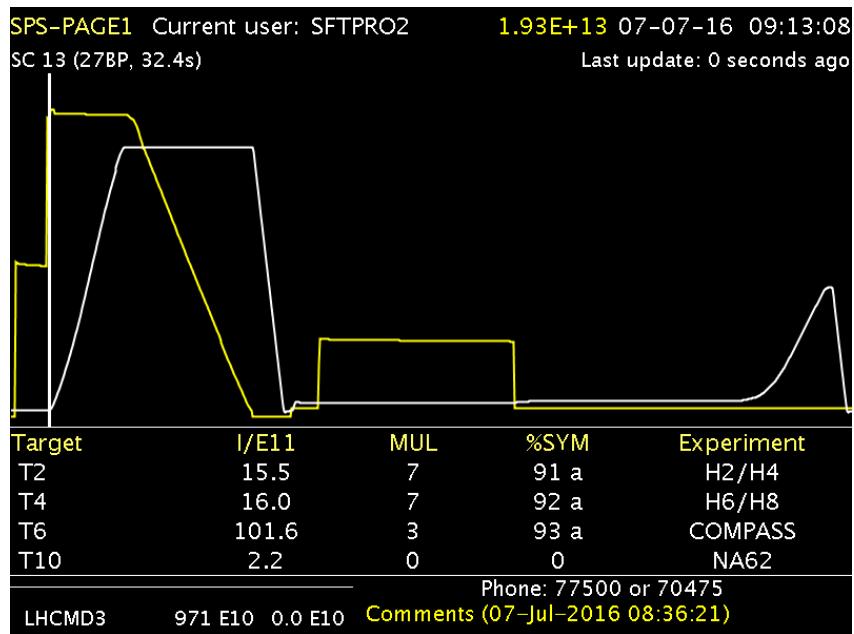


Figure 1.7 – ???

The two types of particles used during the test beam were pions and muons. Pions are obtained from the secondary beam by inserting a thin absorber which removes the electronic component of the beam. They can reach intensities between 10^4 and 10^7 particles per spill and produce a very narrow beam. From their decay, a muon beam can be obtained by using moderate intensity secondary beams and closing the collimators. The muon beam reaching the experimental setup is a wider and less intense.

1.4.1 The GEM Detectors Setup

During the November 2015 test beam campaign, a superchamber of GE1/1-V detectors has been used to collect data. Each detector was equipped with 12 VFAT2 Hybrids mounted in the central four rows. The remaining slots, too far away from the beam spot, were covered with grounding equipment to reduce noise. Each GEM was mounted with a GEB v2 and a OptoHybrid v2a. A picture of the setup is represented in Figure 1.8. Both OptoHybrids were connected to the same GLIB placed in a microTCA crate.



Figure 1.8 – ???

The GEM detectors were connected to a gas system providing a mixture of Ar/CO₂/CF₄ at 45%, 15%, and 40% respectively. The high-voltage was delivered through a ceramic high-voltage divider for one chamber and by separate channels connected directly to the GEM foils and planes for the other. The high-voltage applied is expressed in terms of current flowing through the divider which, by summing up the individual resistors that it is made of, yields a value in volts. Table 1.3 enumerates the different values of the voltages according to the current flowing through the divider. The difference between the divider and drift plane values is due to an input filter placed before the latter which smoothes out fluctuations in the voltage. The low voltages for the OptoHybrids (1.8 V and 4 V) and for the GEBs (2.5 V) were delivered by two power boards mounted on top of the detector stand along with the necessary tools to reprogram the FPGA in case of failure and the debugging headers.

Current (μ A)	Divider (V)	Drift plane (V)	GEM1 top (V)	GEM1 bottom (V)	GEM2 top (V)	GEM2 bottom (V)	GEM3 top (V)	GEM3 bottom (V)
700	3921	3290	2503	2107	1801	1416	805	437
710	3977	3337	2538	2137	1827	1437	816	443
720	4033	3384	2574	2167	1853	1457	828	450
730	4089	3431	2610	2198	1879	1477	839	456
740	4145	3478	2646	2228	1904	1497	851	462
750	4201	3525	2682	2258	1930	1518	862	468
760	4257	3572	2717	2288	1956	1538	874	475
770	4313	3619	2753	2318	1981	1558	885	481
780	4369	3666	2789	2348	2007	1578	897	487
790	4425	3713	2825	2378	2033	1598	908	493
800	4481	3760	2860	2408	2059	1619	820	500
810	4537	3807	2896	2438	2084	1639	931	506

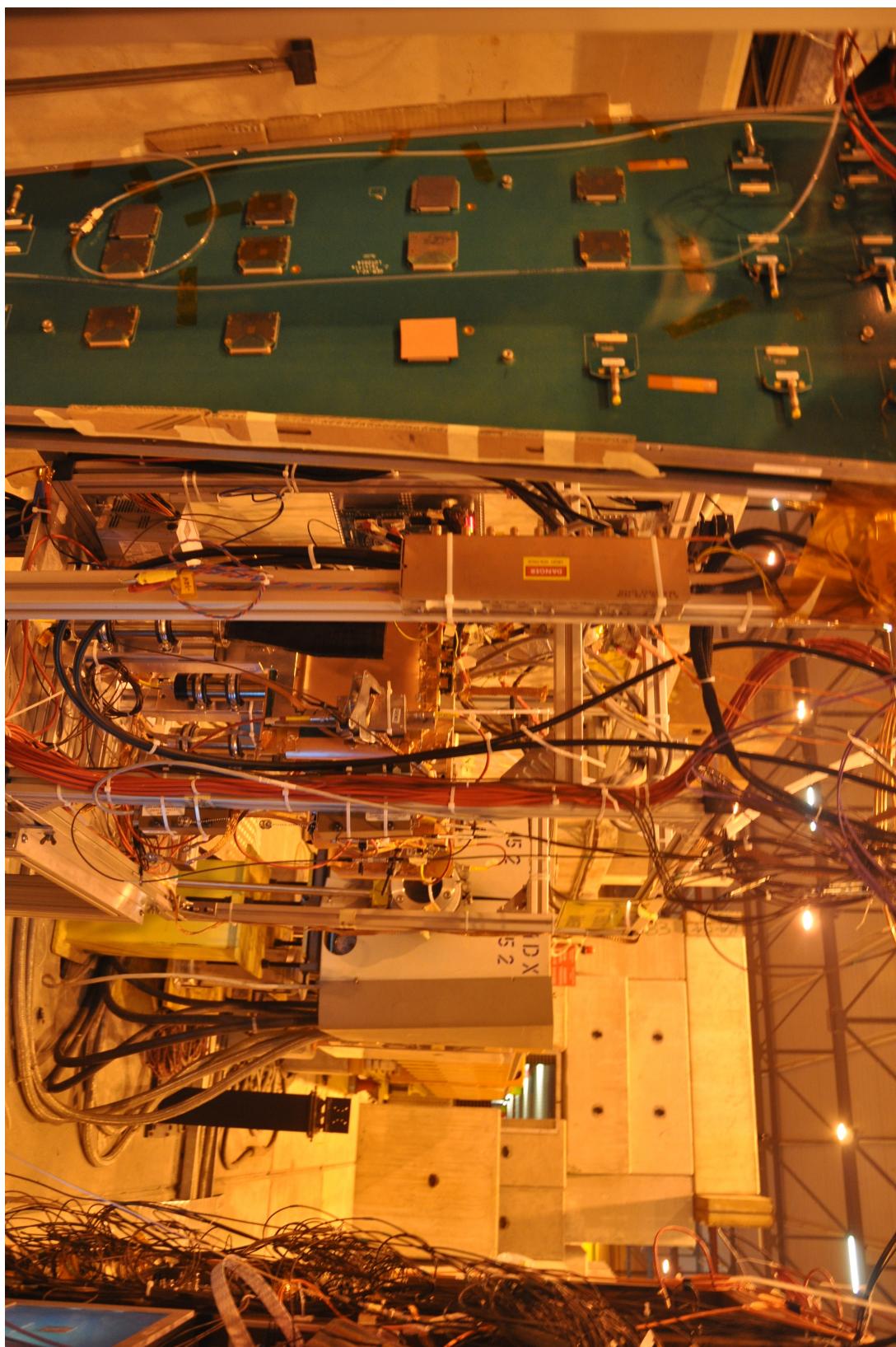
Table 1.3 – ??

1.4.2 The Beam Area Setup

The GE1/1 superchamber has been inserted in the beam path and adjusted so that solely one VFAT2 in the middle column would be exposed to the beam spot. Figure 1.9 shows the setup on the left placed in front of the beam transfer tunnel on the right side. Four scintillators, three placed in front of the detector and visible in the picture, and one placed in the back of the detector were used to provide triggers to the system. The output of the four scintillators was converted to Nuclear Instrumentation Module (NIM) logic and their coincidence used as the trigger source of the system.

NIM uses current driven logic levels which produce -0.8 V over 50 Ω for a logic '1' and 0 V for a logic '0'. As the OptoHybrids use TTL voltage driven levels, namely 2.5

Figure 1.9 – ???



V for a logic '1' and 0 V for a logic '0', a converter had to be used. It was observed that the output levels of the NIM to TTL module were close to the switching voltage and might affect the number of triggers seen by the OptoHybrids. It was computed that this situation occurred in less than 1/10 000 events and could be solved by the offline analysis. Checks were performed to ensure the source of the problem did not originate from the DAQ system itself, which was proven not to be the case.

1.5 Data Analysis

The data collected during the test beam was analysed to study the detector performance according to various parameters. Data was taken at multiple high-voltage values, VFAT2 threshold settings, and particle rates. An analysis framework has been developed in order to combine the readout data from the two GE1/1 chambers. Alignment of the events was non-trivial and based on the difference between two consecutive BC values of the events. Indeed, due to the problem exposed hereabove with the trigger logic and the fact that the two VFAT2s are not guaranteed to have the same value of the BC counter at the same time, matching of the events based on the absolute value of the BC could not be done. Therefore, the difference between the BCs of two consecutive events in the data stream was computed and matched between the two datasets. When a discrepancy arose, one of the datasets had to be realigned. Once realignment was performed, analysis tools had access to both chambers for each event.

1.5.1 VFAT2 Threshold Scans

The first parameter to study is the noise on the VFAT2s induced by the system and by the antenna effect of the strips. To this end, threshold scans have been produced using the web application. These scans are run when the beam is not present. They use the trigger bits of the VFAT2s which do not need a L1A source to be produced. They are sampled at 40 MHz and for each threshold value the ratio of events containing hits against the total amount of collected events is defined to be the noise. Figure 1.10 plots the noise level as a function of the VFAT2 threshold in terms of VFAT2 units, the decimal value written in the 8 bit register afterwards converted to a voltage level, for GEM0 on the left and GEM1 on the right with muons. Although the noise level quickly diminishes, it never reaches the expected value of 0 even at very high threshold. Typically, previous test beams have shown that a value of 15 kills the noise. Furthermore, it can be seen that noise level is higher in GEM1, which reaches 20% at a threshold value of 10 against 15% for GEM0. From the threshold scans, it was decided to run both chambers at a VFAT2 threshold level of 25 for data taking runs. This value is considered to be the default threshold except if otherwise specified.

After the end of the test beam, the noise issue has been investigated thoroughly and two sources have been isolated: the I2C clock used for the slow communication and the LEMO cables used for the trigger input. In the version of the firmware

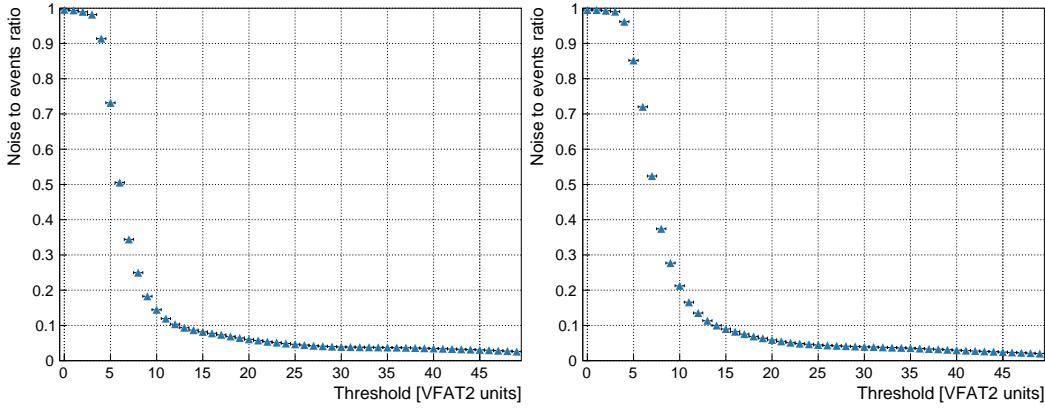


Figure 1.10 – Plots of the noise level as a function of the VFAT2 threshold in terms of VFAT2 units, the decimal value written in the 8 bit register afterwards converted to a voltage level, for GEM0 on the left and GEM1 on the right with muons.

used during the test beam, the I2C clock runs continuously which induces noise on the trigger bits of the VFAT2s at a frequency of 100 kHz, the same as the clock itself. This problem has been solved by activating the I2C clock solely when a slow control operation takes place, limiting the number of corrupted trigger bits. The second issue originated from current loops formed between the low power cables of the OptoHybrids and the LEMO cable connecting the system to the NIM crate. The ground of the latter was connected to the OptoHybrid ground through the shielding of the LEMO cables. Once the two ground systems were isolated, noise level comparable to previous observations were obtained.

It was also observed that GEM1 experienced a leakage current through the GEM foils of about $10 \mu\text{A}$. This had as consequence to reduce the voltage on the GEM foils and thus decrease the gain and efficiency of the detector. The detector also suffered from water condensation inside the gas pipes which caused some problems during the first days of run. These issues can explain the difference in noise observed between the two chambers.

1.5.2 VFAT2 Latency Scans

Once the threshold on the VFAT2s has been set and the beam has been turned on, a latency scan has to be done in order to capture the events corresponding to the triggers coming from the scintillators. Again the web application is used to run the scan on the VFAT2s. Similarly to the threshold scan, the latency scan counts the ration of events with a hit against the total number of events. However, it uses the tracking data provided by the VFAT2s and not the trigger bits. Figure 1.11 plots the ratio of hit events as a function of the VFAT2 latency in terms of BXs for a monostable pulse length of four clock cycles for GEM0 on the left and GEM1 on the right with muons. When the latency is too low or too high, the VFAT2 misses the hit and a baseline value of 0.05 is observed corresponding to noise. Once the latency enters the correct windows, it rises to a value close to 0.98. These plots illustrate

the usefulness of the settable monostable pulse length. Indeed, the two points at approximately 0.5 and 0.6 indicate that hits are falling in two adjacent latency values, each collecting half the events. In case of a non-adjustable setting, the maximum efficiency of the system would be reduced by half. From these results, a VFAT2 latency value of 11 has been used for data taking runs. This value is considered to be the default latency except if otherwise specified.

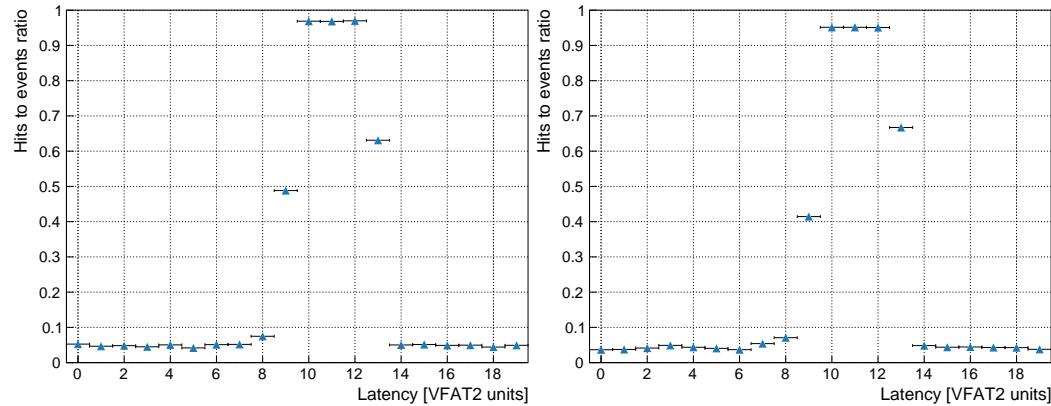


Figure 1.11 – Plots of the ratio of hit events as a function of the VFAT2 latency in terms of BXs for a monostable pulse length of four clock cycles for GEM0 on the left and GEM1 on the right with muons.

Latency scan are also used to measure the noise and the efficiency of the detector. When looking at out-of-time events, the only contribution to the hits is noise. On the other hand, when considering the window in which the latency is well adjusted, both the noise and the particle hits play a role in the measurement. Taking this value to be the measured efficiency, it corresponds to the ratio between the number of events detected by the GEM detector and those detected by the scintillator system. However, the measured efficiency is the superposition of two poisson processes: the noise and the real detector efficiency. In order to obtain the computed or real efficiency, simulations were developed to compute upper and lower limits on the real efficiency according to the noise levels. This was achieved by running a thousand pseudo-experiments for each possible efficiency value and comparing the result to the measured value. With a noise level of 100%, the real efficiency could be anything between 0% and 100%. However, with a noise level of 1%, the computed efficiency will be close to the measured efficiency.

1.5.3 Efficiency against High-Voltage

Using the latency scan to compute noise and efficiency values, the evolution of these parameters has been studied according to the high-voltage applied on the detector. Figure 1.12 plots the measured noise (green), measured efficiency (blue), and computed efficiency (orange) as a function of the high-voltage expressed in μA for GEM0 on the left and GEM1 on the right with muons. Both results show an increase in efficiency with the high-voltage, reaching a plateau around 98%. This

is caused by the higher gain at which the GEM foils start to operate which in turn amplifies the signals and allows for it to be detected by the VFAT2 ASIC. It can be noted that a shift exists between the results for GEM0 and GEM1. Values for the latter are shift to the left by approximately $10 \mu\text{A}$. While GEM0 reaches a plateau at $800 \mu\text{A}$, GEM1 requires $810 \mu\text{A}$ to reach an efficiency around 98%. This is due to the leakage current observed during installation which decreases the voltage applied to the GEM foils and in turn performance. It can also be noted that the noise is constant with the high-voltage. These results were compared against previous test beam campaign results [7] and are in excellent accordance with each other. From this, it was decided to operate both chambers at $800 \mu\text{A}$ as sparks were sometimes observed at $810 \mu\text{A}$. This value is considered to be the default high-voltage except if otherwise specified.

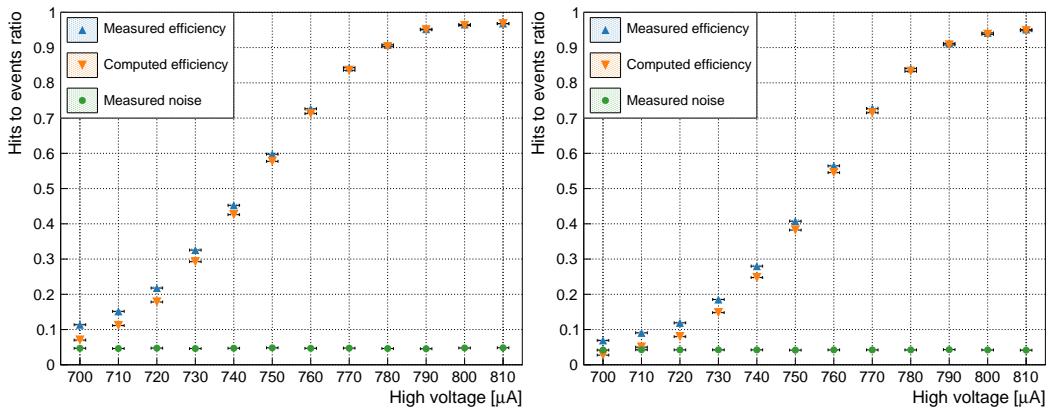


Figure 1.12 – Plots of the measured noise (green), measured efficiency (blue), and computed efficiency (orange) as a function of the high-voltage expressed in μA for GEM0 on the left and GEM1 on the right with muons.

1.5.4 Efficiency against VFAT2 Threshold

Although it had been decided to run the system with a VFAT2 threshold of 25, a study of the evolution of the noise and efficiency according to the threshold was performed. It aimed at quantifying the amount of noise and signal that was cut when increasing its value allowing to select the configuration yielding the best signal over noise ratio. Contrary to the previously operated scan which used trigger bits to measure the noise, this procedure uses tracking data and the full granularity information to analyze events. Figure 1.13 plots the measured noise (green), measured efficiency (blue), and computed efficiency (orange) as a function of the VFAT2 threshold in terms of VFAT2 units for GEM0 on the left and GEM1 on the right with muons. Additional points in purple have been extracted from the tracking data. Using the two GEM detectors, it is possible to measure the efficiency of one using the other as reference. For each event, the following cuts were applied on the first GEM in order to reduce the dataset:

- only one cluster, group of adjacent strips that have been hit, is present in the event;
- the sole cluster must contain exactly two active strips;
- the center of the cluster must be at least ten strips away from the sides of the chip.

The first condition eliminates events where a hit is accompanied by noise, creating an ambiguity on the cluster associated with the particles. The second condition helps to remove noise related events as they have a cluster size distribution peaked at one strip. Therefore, combining these two criteria creates a stringent condition for the selection of particle generated signals. The final condition ensures that the hit in the second GEM detector will be located inside the VFAT2 thus eliminate acceptance problems. Using the reduced dataset, the algorithm computed the efficiency by looking for hits in the second GEM detector in a window of five strips around the center of the cluster of the first GEM.

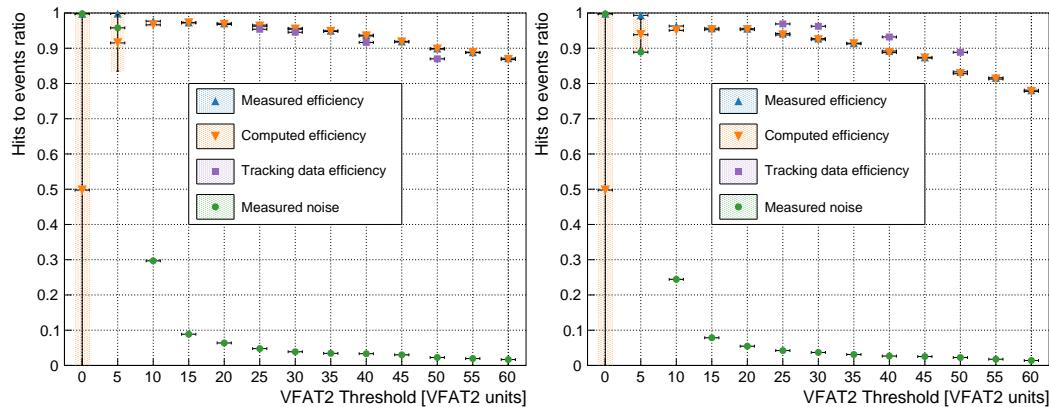


Figure 1.13 – Plots of the measured noise (green), measured efficiency (blue), computed efficiency (orange), and tracking data based efficiency (purple) as a function of the VFAT2 threshold in terms of VFAT2 units for GEM0 on the left and GEM1 on the right with muons.

It can be seen that at very low thresholds the computed efficiency displays a large error due to the high noise. As the latter decreases with the increase in the threshold, the algorithm computing the real efficiency is able to tighten its limits ultimately providing a computed efficiency close to the measured value. From a threshold value of 15, efficiency starts to slowly drop reaching 90% at 50. This shows how the threshold cuts the signal while only slightly affecting noise. From these results, the choice to operate the system at a threshold of 25 is validated. Although displaying a slightly lower efficiency, still around 97-98%, than results at a threshold of 20, the lower noise is non-negligible and reaches a local plateau.

TODO: Deviation with tracking data

1.5.5 Efficiency against Particle Rate

Similarly to the measurements and computations made for the previous study, the effect of particle rate on the efficiency was measured. Focus is given on the latter as it has been noted that noise levels are not influenced by the increase in rate. Figure 1.14 plots the measured efficiency (blue), computed efficiency (orange), and tracking data based efficiency (purple) as a function of the particle rates for GEM0 on the left and GEM1 on the right with pions. Control over the particle rates were done using collimators in front of the beam setup.

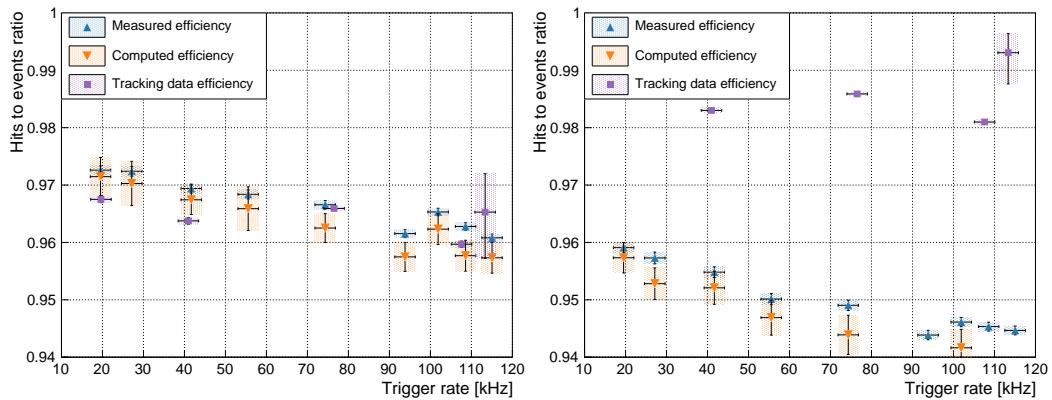


Figure 1.14 – Plots of the measured efficiency (blue), computed efficiency (orange), and tracking data based efficiency (purple) as a function of the particle rates for GEM0 on the left and GEM1 on the right with pions

These results support the claim that GEM detectors can sustain rates up to several tens of kHz. When going from 20 kHz to 120 kHz, a drop in efficiency of 1% is observed for GEM0 and of roughly 2% for GEM1.

TODO: Deviation with tracking data

1.5.6 Cluster Multiplicity and Size

Two studies that can be performed offline using the recorded data are the evolution of the cluster multiplicity, the number of clusters, and the cluster size, the number of strips per cluster, with the VFAT2 threshold. Figures 1.15 and 1.16 respectively plot the cluster multiplicity and cluster size for muons (green) and pions (blue) as a function of the VFAT2 threshold in terms of VFAT2 units for GEM0 on the left and GEM1 on the right. The results have been compared against previous test beam data [8, 9] and show good accordance.

The cluster multiplicity reflects the number of clusters present in the event, some of which are due to noise other to signal. Its evolution is tightly linked to the evolution of the efficiency as a function of the VFAT2 threshold. It was observed that for threshold values of 25 and 30, the noise and efficiency were only slightly decreasing, while doing so at a faster rate at higher thresholds. This explains the proximity of

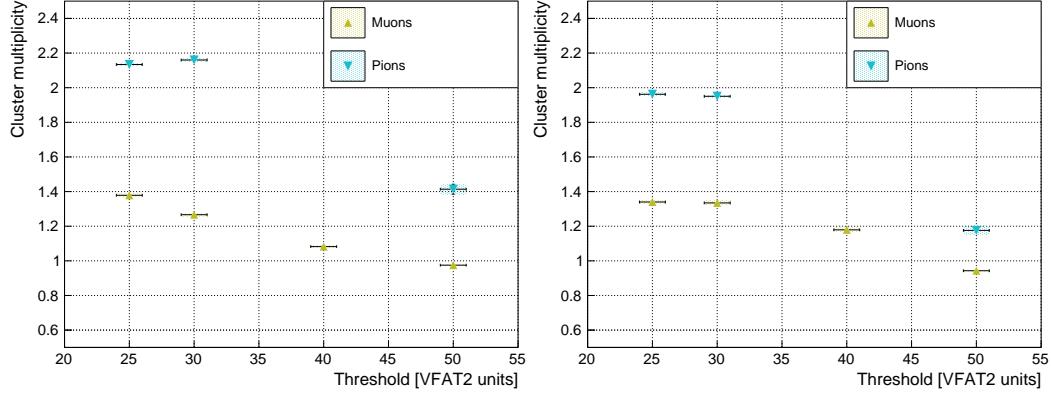


Figure 1.15 – Plots of the cluster multiplicity for muons (green) and pions (blue) as a function of the VFAT2 threshold in terms of VFAT2 units for GEM0 on the left and GEM1 on the right.

the points at 25 and 30 for both pions and muons in GEM0 and GEM1. When the threshold is further increased, the effects start to appear, cutting both the noise and the signal. Furthermore, the counting of cluster is not directly affected by the cluster size when the latter is rather small. This means that two clusters of size two are equivalent to two clusters of size one. Even if by increasing the threshold, some strips are being deactivated, other strips in the cluster might still remain active thus leaving the cluster statistic unchanged. When cluster sizes increase too much, clusters start to overlap and the previous assumption becomes invalid.

TODO: Why more clusters with pions

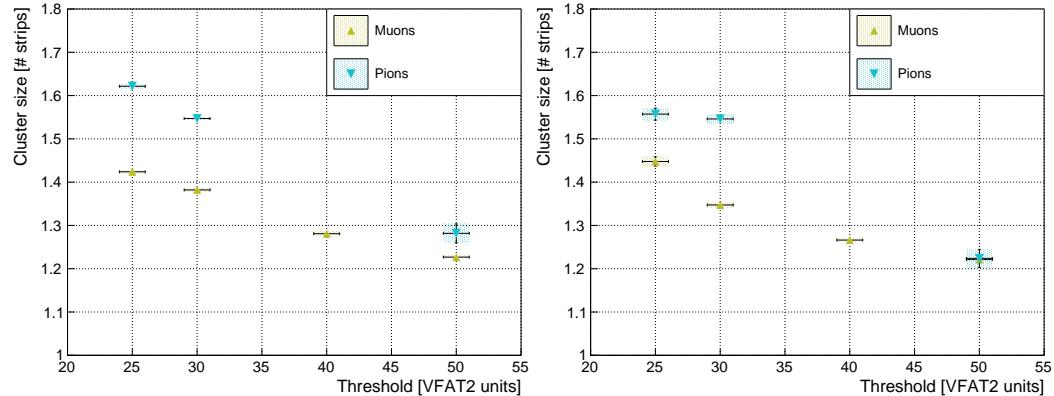


Figure 1.16 – Plots of the cluster size for muons (green) and pions (blue) as a function of the VFAT2 threshold in terms of VFAT2 units for GEM0 on the left and GEM1 on the right.

The previous explanation is supported by the plots of the cluster size as a function of the threshold. It is clear that the cluster size decreases as the threshold increases, meaning that less and less strips are being fired. At low threshold, clusters contain roughly 1.5 strips and an average of 1.6 clusters are present per event. When

increasing the threshold, the average number of strips fired decreases slightly which does not immediately affect the cluster multiplicity: clusters are reduced in size but not in numbers.

1.5.7 Beam Profile

Finally, to illustrate the structure of the beam, one can create a beam profile obtained by superimposing the collected data and counting the number of hits for each strip. Figure 1.17 plots the beam profile for muons (green) and pions (blue) for GEM0 on the left and GEM1 on the right. The pion beam is well defined and centered on the VFAT2 strips as it is guided and collimated by the magnets of the transfer tunnels from the SPS to the experimental area. The muon beam on the other side displays a large spread due to the decay process of the pions and the collimators installed to stop them. The sharp cut-off on the left side of the profile is due to the geometrical acceptance of the PM placed in the back of the detector. These plots show that the chips do not have any dead channels and that the accumulation follows the expected gaussian shape.

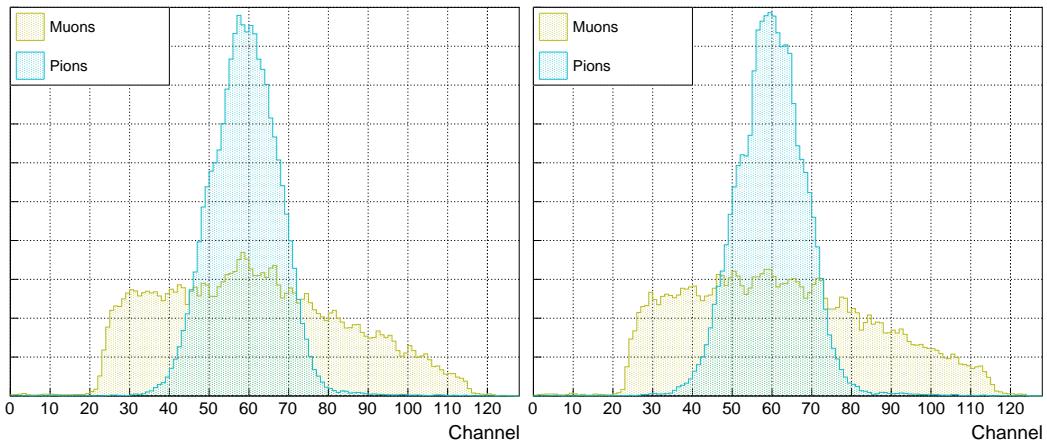


Figure 1.17 – Plots of the beam profile for muons (green) and pions (blue) for GEM0 on the left and GEM1 on the right.

1.6 Conclusion

The November 2015 test beam campaign has shown that the DAQ system composed of the VFAT2 ASIC, GEB v2, OptoHybrid v2a, and GLIB runs flawlessly providing quality data which has been used to compute the noise and efficiency on the system as a function of various parameters. The system is able to handle a superchamber of GE1/1 detectors with ease.

The flexible firmware architecture we developed for the OptoHybrid which provides the user with numerous monitoring and control options ran for the duration of the test beam and carried out all required tasks. The wishbone-like communication protocol implemented in the FPGA allows future developments to be easily integrated

and modules to communicate with one another.

The firmware of the GLIB and the communication through the optical links have ran smoothly showing no errors on the transmitted data. The readout rate was high enough to prevent buffer overflow and well integrated with the software.

The web application has been used extensivly and is still in use for gain and uniformity studies at CERN. It has proven to be a complete solution to control and monitor the DAQ system allowing users to run all the necessary scans.

The recorded data has been analyzed. Threshold scans have shown an elevated noise level which has since then been understood and solved. Using the latency scans and the tracking data, various studies of the various of noise and efficiency levels have been performed against high-voltage, threshold, and particle rates. Results have been compared to previous test beams and are consistant. An efficiency of 97% have been obtained and rate capability has been tested up until 120 kHz with a resulting efficiency of 96%.

