A NEW DAQ INSTALLATION FOR THE SIS18 BEAM POSITION MONITORING SYSTEM AT GSI

G. Jansa, J. Bobnar, S.Sah, M. Sekoranja, I. Verstovsek, Cosylab d.d., Ljubljana, Slovenia
T. Hoffmann, K. Höppner, P. Kowina, K. Lang, M. Schwickert, GSI Helmholtzzentrum für Schwerionenforschung, Darmstadt, Germany

Abstract

The BPM system for the heavy ion synchrotron SIS18 consists of 12 shoe-box pick-ups. Their analog signals are digitized by Libera Hadron units, manufactured by I-Tech, Slovenia. In addition, the Libera is used for a FPGA based online position calculation. Due to the high rate of 12 times 60 MB/s at 5 MHz bunch frequency, a dedicated 10 Gigabit Ethernet network is used to concentrate the data from all Liberas on two concentrator PCs. Besides all control actions these PCs are running the DAQ server applications, developed and produced within the CERN made Front-End Software Architecture (FESA) framework. These servers are also used for further data analysis such as tune and closed orbit measurements. As a mediator between the FESA front-end and the Java GUI application level the CMW/RDA middleware, also developed at CERN, is used. Due to the modular layout, the system is already extensible for the SIS100 and SIS300 BPM systems at the Facility of Antiproton and Ion Research (FAIR). The implementation of the high speed data transfer, the FESA servers and the Java GUI application is presented.

INTRODUCTION

The presented system will be used for beam position monitoring in the SIS18 synchrotron at GSI. The purpose of the project was not only to replace the old BPM system but also to study and evaluate new technologies (FESA, CWM/RDA, and JAPC), which are candidates to be used for the future FAIR control system. The project was accomplished in collaboration with the beam diagnostic and control system departments at GSI.

SYSTEM DESCRIPTION

The BPM system consists of 12 shoe-box pick-ups attached to 12 Liberas. A HP ProCurve 2900-24G switch transfers acquired data from Libera's 1GbE ports to 10GbE ports located on two concentrator and control PC servers (Fig. 1). The PC servers are systems with two 2.0 GHz Quad-Core Intel Xeon CPUs and 32 GB of DDR2 RAM with a Scientific Linux 5.0 (SL5) operating system. The control of the Liberas is done over a separate 1GbE accelerator network, which is also used to control a programmable timing interface (PTIF), signal amplifiers and a noise generator. The noise generator is used for tune measurements to produce bunch oscillations. The Java GUI application runs on a SL5 based PC with 3.0 GHz Intel Core 2 Duo CPU with 2GB of RAM.

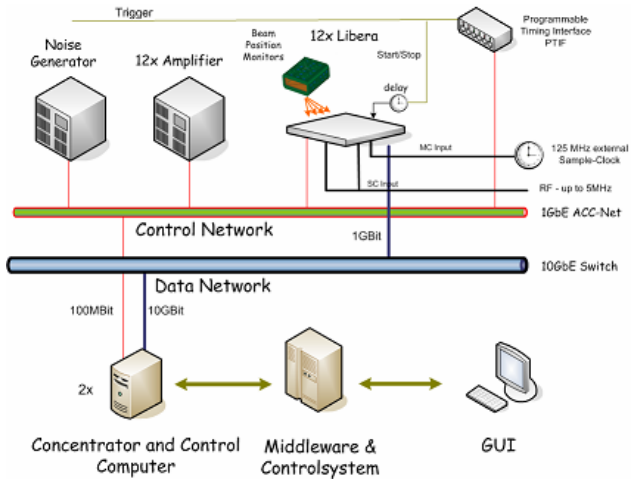

Figure 1: Hardware layout.

The bunch detection, position calculation and data transfer are done inside Libera's FPGA [1] whereas control functions are implemented by using a separate server running on the Libera's SBC. Each PC server hosts 6 BPM FESA [2] device servers. In addition to that, one of the PC servers is running also the BPM master FESA device server, which is used to orchestrate the operation of all 12 BPM FESA device servers (Fig. 2). The Java GUI application communicates with the FESA servers via CMW/RDA [3] access, which is encapsulated in the JAPC framework [4].

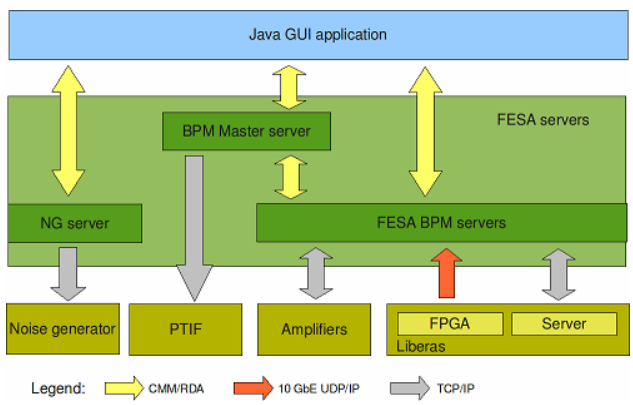

Figure 2: Software layout.

The operation of the BPM system is divided into three main modes: bunch to bunch mode, calibration mode and raw data mode. In bunch to bunch mode, the user selects start and stop acquisition events by configuring the PTIF, which is connected to the GSI timing. During the acquisition, the Liberas send data over the 10 GbE

network to the server PCs until a stop event is issued by the PTIF. On the server PCs, the acquired data is processed and in turn displayed by the GUI application.

To acquire meaningful data in bunch to bunch mode the system needs to be calibrated. Two types of calibration are distinguished. The zero line calibration is used to correct ADC drifts of each Libera unit. Within the zero line calibration no data is sent over the 10 GbE network since zero line calibration values are used inside Libera's FPGA and are read over the common 1GBit accelerator network for persistence only. For the position offset calibration the signals connected to the Libera ADC inputs are switched from the real detectors to test signals. This is done in order to remove slight differences in the analog signal amplification. During this calibration, position data is sent to the server PCs, where then position calibration values are calculated.

The raw data mode is meant to be used by experts. After the occurrence of a start event, the Liberas start acquiring raw data with 125 MHz sampling rate. The data rate acquired by each Libera is 8 GbE/s and therefore too high to be transported to the server PCs on the fly. That is why the acquired data is stored on the Libera itself and transported at the acquisition end. Since each Libera has only 256 MB of free memory storage the acquisition time in the raw data mode is limited to 125 ms.

DATA TRANSFER

In bunch to bunch mode a 96 bit BPM position data packet is generated for each detected bunch (Fig. 3). The BPM packet is composed of horizontal and vertical bunch position (22 bit each), RF timestamp (16 bit), error detection timestamps (start and end of bunch window relative to RF timestamp, 12 bit each), error (overload and underload) detection bits for each plane (4 bit each), and intensity (4 bits). The BPM packets are also used in the position offset calibration mode. For the raw data mode 64 bit long packets are used (Raw packet). This raw packet comprises four signals with each being 16 bits long (Fig. 3).

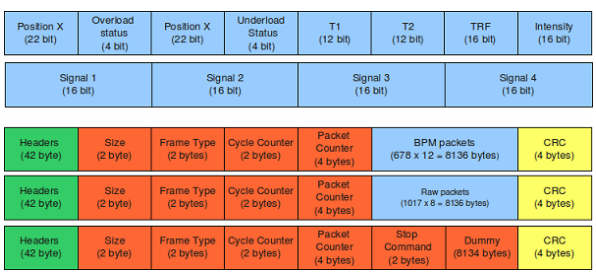

Figure 3: From top to bottom: BPM packet, Raw packet, BPM frame, Raw frame, and Stop frame.

Due to the high amount of data generated by each Libera the User Datagram Protocol (UDP) is used to transmit data to the server computers. To reduce the amount of UDP frames on the network and consequently also the number of interrupts on the server side, jumbo UDP frames with MTU of 8192 bytes are used. The headers of the UDP/IP-Ethernet protocols and checksum (CRC) account for 46 bytes, which leaves 8146 bytes for the payload. Since the UDP protocol does not offer any hand-shaking dialogues for guaranteeing reliability of the data transmission, two counter fields, cycle counter and packet counter, are used. The cycle counter is incremented with each new cycle and the packet counter with each frame, which guarantees detection of lost frames. For the server, to be able to distinguish between different UDP frame types, an additional frame type field is used. A size field is used to provide the number of valid BPM- or Raw packets in the payload (Fig. 3).

UDP frames are divided into three categories based on the data they contain, here the BPM frame, Raw frame, and Stop frame. The BPM frame is used in bunch to bunch and position offset calibration mode and can carry up to 678 BPM packets. The Raw frame is used in raw data mode and holds 1017 raw packets. The Stop frame is send at the end of the data transfer and indicates the end of the cycle. By replacing the stop command this type of frame can be used also for other purposes in the future.

DEVICE SERVERS

When the acquisition is started by means of a FESA server action, BPM FESA servers are waiting for the arrival of Libera data packets. To distinguish between packets sent from different Liberas each unit is sending data to its own unique UDP port. The acquisition process inside a BPM server is comprised of three threads (shown on Fig. 4), the acquisition thread, processing thread and FFT thread. The acquisition thread has the highest priority and its purpose is to acquire UDP frames from the network and to monitor the connection health. The inbound frame is at first placed into an intermediate ring buffer. If the processing thread is idle, it gets signalled by the arrival of a new frame and the acquisition thread returns to check for the arrivals of new frames.

When the processing thread is woken up, it retrieves frames from the ring buffer and checks if the packet and cycle counter are ok, meaning that no frames were lost. If frames were lost the acquisition is cancelled and the clients (GUI) are notified about it. If everything is ok, the data from the frame is extracted and put into an internal buffer. During acquisition clients can request for new data (e.g. operators zoom within the position chart and a new set of data must be fetched from the server by the client). To prevent conflicts and inconsistent data two buffers are used. One is used by the processing thread to write new data and the other is used by clients. At the end of any acquisition the roles are switched. To reduce the CPU's computation consumption an FFT can be calculated for only one BPM at a time. During an FFT computation for a dedicated BPM, the processing thread also monitors the number of received bunch positions. When this number exceeds the number needed to compute a new FFT chunk (usually 2048) the FFT thread is woken up. The acquisition ends when a stop frame is received from the Libera. When this occurs, the acquisition thread must first

wait for the FFT thread to finish the current calculation and after that the FESA real time action (named AcqRT) is triggered, which then notifies clients that new data is available (Fig. 4).

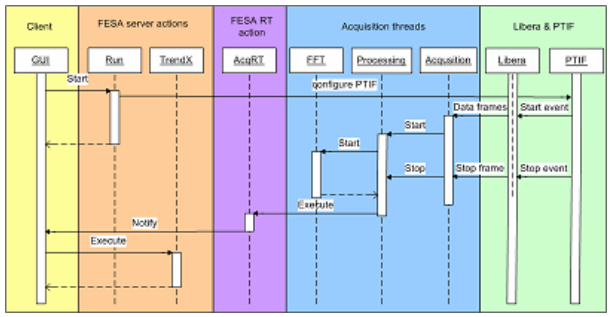

Figure 4: Acquisition cycle sequence with FFT calculation.

At the end of the cycle, the server contains a complete set of cycle data i.e. no data reduction is performed on the fly. For a 1s long cycle with bunch frequency of 5 MHz this sums up to around 540 MB of data per PC server. The data is reduced for specific client requests, which are implemented as FESA server actions as shown in Fig. 4. Here the TrendX is a server action that returns reduced data for position x over time. For example, a chart displaying bunch positions vs. time over a complete cycle is normally showing only a few thousand points whereas the total number of positions stored in the buffer is approximately for 5 million bunches.

Because FFT calculation is time consuming and hence can conflict with the data acquisition, it is calculated on a dedicated CPU core. This means that the cores 1 to 7 are used for acquisition and data processing and core 8 is used for FFT calculation only. Due to a multithreaded environment a special care has to be taken to prevent dead locks and to assure data integrity especially in exceptional events e.g. when the user stops the acquisition in the middle of the cycle.

GUI APPLICATION

The GUI application is written in Java and is based on CERN's graphical framework, which provides a main frame with already included error console and status line. On top of this, the application uses CERN's chart framework for presentation of all data obtained from the servers. To access the data on the server the CERN's JAPC framework is used. The JAPC calls were wrapped into a GUI device model, which provides all the necessary commands required by the user interface. The device model uses only asynchronous calls and is completely event based towards the GUI. This allows the model to be completely separated and independent of the GUI allowing easy changes if the device server's API is changed in the future.

The main window (Fig. 5) is divided into two sections. The upper half of the window provides a panel with accelerator information and for the control of the beam position monitors while the lower half is used to display various data viewers. In bunch to bunch mode the user can select between the following viewers closed orbit, overload/underload status, trending, trending mean value, intensity, tune, auxiliary status and consistency check. The tune panel is exemplarily shown in Fig. 5.

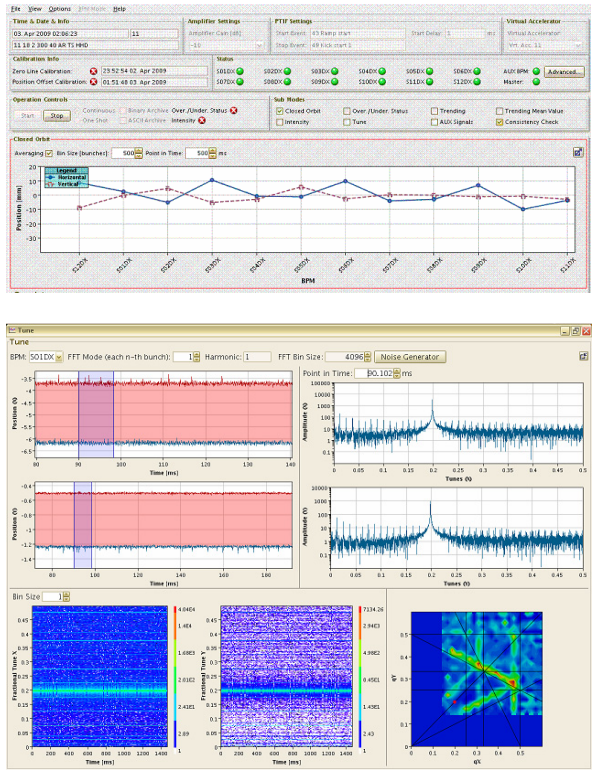

Figure 5: GUI main window with closed orbit viewer (top) and tune panel (bottom).

SUMMARY AND OUTLOOK

Extensive tests with generated test signals simulating bunches at 5MHz have shown that the system is stable and without frame losses. Currently, tests with real beam signals are in progress. Since the system is modular it can be extended to be used at the SIS100 synchrotron at the FAIR project with a minimum of effort. The BPM system at the SIS100 will probably use 84 Liberas and the highest bunch frequency will be 6 MHz. 7 Liberas per one server PC will be used, therefore the highest data rate will be around 500 MB/s per PC. Foregoing tests have shown that the system can handle this amount of data with negligible frame losses.

REFERENCES

[1] K. Lang et al. "Performance tests of digital signal processing for GSI synchrotron BPMs", in Proc. of PCaPAC08

[2] M. Arruat et al. "Front End Software Architecture", in Proc. of ICALEPS07

[3] CMW/RDA, http://cmw.web.cern.ch/cmw/products/cmw-rda/cmw-rda.html

[4] V. Baggiolini et al. "JAPC - the Java API for Parameter Control", in Proc. of ICALEPCS05