DAΦNE BTF IMPROVEMENTS OF THE TRANSVERSE BEAM DIAGNOSTIC

B. Buonomo, C. Di Giulio, L.G. Foggetta, INFN – Laboratori Nazionali di Frascati, Frascati, (Italy)
P. Valente, INFN – Sezione di Roma, Rome (Italy)

Abstract

The DAΦNE BTF (beam-test facility) can provide electrons and positrons, tuning at runtime different beam parameters: energy (from about 50 MeV up to 750 MeV for e^- and 540 MeV for e^+), intensity (from single particle up to 10^{10}/bunch) and pulse length (in the range 1.5-40 ns) up to 49 Hz, depending on the operations of the DAΦNE collider. The beam spot and divergence can be adjusted, down to sub-mm sizes and 2 mrad (downstream of the vacuum beam-pipe exit window), matching the user needs. We describe of the BTF beam transverse monitor systems based on FitPIX detectors, operating in bus synchronization mode externally timed to the BTF beam. We also describe our custom software allowing the acquisition and synchronization of the beam diagnostics with the users data, using TCP/IP calls to MEMCACHED. The performance of the system in a variety of beam intensity, energy and focusing conditions is reported.

THE DAΦNE BEAM TEST FACILITY (BTF)

The DAΦNE accelerator complex hosts the BTF (Beam Test Facility). It is a fully equipped experimental hall where electrons or positrons provided by high intensity LINAC usually injected into to the DAΦNE damping ring, are driven by a pulsed magnet in a transfer line towards the experimental hall. The BTF performances and the upgrade are described in [1].

Runtime tunable electrons and positrons beams in a defined range of different parameters could be provided to the users, with energy of the beam that can be settled on 50 MeV up to 750 MeV, for electrons, and 540 MeV for positrons. The BTF usually works in parasitic way using the bunches not accumulated by the damping ring and the delivery rate is depending on the DAΦNE injection frequency (25 or 50 Hz) with a duty cycle also changing according to the DAΦNE injection status, up to 49 bunches/s.

In the dedicated mode working condition, the beam pulse length can also be adjusted from 1.5 to 250 ns. Two major modes of operations are possible, depending on the user needs: high and low intensity. In the high intensity mode the LINAC beam is directly steered in the BTF hall with a fixed energy (i.e. the LINAC one, fixed to 510 MeV during the DAΦNE operations) and with the capability in multiplicity selection from 10^{10} down to 10^4 particles/bunch. In the low intensity mode a stepped Copper target with three possible radiation lengths (1.7, 2 or 2.3 X_0), is introduced in the initial portion of the BTF line for the production of a secondary beam with a full-span energy selectable from LINAC energy until 50 MeV giving related setting current to the energy selector dipole and with the multiplicity selectable down to single particle/bunch managing the sets of horizontal and vertical collimators. The typical momentum uncertainty is well below 1% from 750 MeV up to 50 MeV.

THE BTF PIXEL DETECTOR LAYOUT

The BTF users during the beam test of their detectors require a fast transverse beam imaging during runtime, with some pre-analysis capability to control the multiplicity and transverse dimensions of the particle beam. For the same reasons, the related low-level software has to be easy and efficient integration in the plurality of user data acquisition codes. Concerning graphical user interface software, we decided to use LabVIEW®.

In [2, 3] are described the BTF detectors and the related software available for the users.

Silicon Pixel Detector in BTF

The FitPIX [4] detector is complying our requirements in timing and others performances as the transverse profile beam monitoring.

In order to allow more than one different consumers at the same time, accessing to the live data at the maximum frame-rate within the BTF duty cycle, a typical producer-consumer software architecture is implemented with data caching on MEMCACHED(MC) server.

This has been achievable developing all the available programming solutions: beginning with the python scripting capability by the vendor software kit, down to low-level programming as described in [5], which required some interaction with ADVACAM for a full exploitation of the Linux libraries.

Now a software version for the single FitPIX, the multidetector-single FitPIX, and more FitPIX devices in daisy chain is available and the same software architecture allowed us to use it for a GEMPIX tracker [6] (a four sensors detector with a software configuration similar to the four-stacked layers in a single FitPIX).

PRODUCER SOFTWARE LAYOUT

We reached the best performance of our low level code by using C/C++ compiled code in an Ubuntu 14.04LTS environment, using the multiple frame acquisition, and by interfacing to a modified API library, in strong collaboration with the ADVACAM software engineers.

For a single FitPIX, single thread handling the FitPIX USB interrupt manages a multiple frame acquisition. After each triggered frame a (intra-thread) function entertains the frame data on the library internal memory, invoking a callback function. This procedure is automatic in multiple frame acquisition, so we limit to 1000 frames per repetition with the aim to keep the memory usage below 2GB per FitPIX.

The callback function is programmable by the user code and it is use for handling the frame data in the user space. Our choice was to get directly the frame data at each interrupt as described in [5] where the performances were measured and justify this choice.

The Producer Data Structure

We have adjusted the data structure in order to deliver the minimum necessary information for each frame in the different use-cases. We have then implemented three possible data structures:

- array = send all the matrix data (65536 pixel value data, indirect pixel indexing),
- sparse mode 2dim array = a variable size bi-dimensional array [2,N]. Each slice is the couple of pixel data: pixel number and pixel value
- ultra sparse mode array = each array element stores the over threshold pixel number

All of these cases are headed by the following values:

– Case and FitPIX configuration identifier;
– Time tag (resolution 0.1 ms);
– number of fired pixel;
– frame number;

The data types change by the used FitPIX configuration: for single layer FitPIX (as imaging detector) are unsigned short int, with the header expressed in 6 array elements. For multiple-layers FitPIX we use unsigned int, for low data regime, with five header elements (still six for compatibility, one being spare). The FitPIX configuration file and data are labeled with the serial identifier of the chip, distinctive id for a multiple FitPIX configuration.

The producer software pushes the processed FitPIX data in these key-value couples on the MC server, where the key name has the FitPIX detector unique identifier, safeguarding so the source. The producer and the consumer C/C++ codes use standard MC calls provided by the POSIX, thread-safe libmemcached [7, 8] library. For the consumer software, written in LabVIEW, we use our custom MC API's. This choice is due to the high reliability obtained in a heavy data load environments of the DAΦNE Control System, BTF [2, 3] and the very good performance obtained within the !CHAOS project [9, 10], where this configuration via multiple, concurrent producer and consumer calls, has been tested.

An excellent stability of the producer software, running without any problem for months, while providing data to different (BTF or users) consumers is the main aim reached.

The performances of the producer of the system in a variety of beam intensity, energy and focusing conditions are reported in table 1 and table 2. We profiled the callback from its interrupt-starting to the data pushing on MC for a double FitPIX configuration both for the two data regime.

Normally, we stay within 1ms for low data one, as we need. It's clear that the bottleneck of the MC pushing in full matrix regime, however we still within or close to the BTF Ethernet bandwidth.

Table 1: Two Detectors, Low Data, and Timing Profile

Low	USB_0	USB_1	MC_0	MC_1
Mean[s]	5.9E-05	5.9E-05	8.62E-04	8.9E-04
Std[s]	1.7E-05	1.7E-05	1.79E-04	1.5E-04

Table 2: Two Detectors, Full Matrix and Timing Profile

Full	USB_0	USB_1	MC_0	MC_1
Mean[s]	5.8E-05	6.0E-05	2.23E-02	2.33E-02
Std[s]	1.6E-05	1.5E-05	2.09E-03	1.37E-03

OVERVIEW ON CONSUMER SOFTWARE

LabVIEW and C/ROOT Example Code

We have developed a LabVIEW runtime display of FiPIX data and extracted beam parameters display (such as 3-d transverse image, beam centroid and Gaussian fit data), working both as shot by shot (instantaneous) and in cumulative mode as shown in figures 1 and 2.

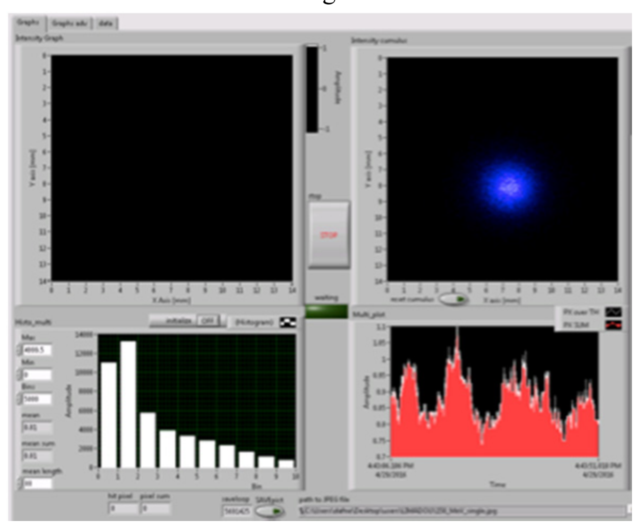

Figure 1: Upper left beam spot bunch per bunch, upper right the cumulative spot images, lower left the multiplicity distribution, lower right the mean multiplicity over 38 frame vs time.

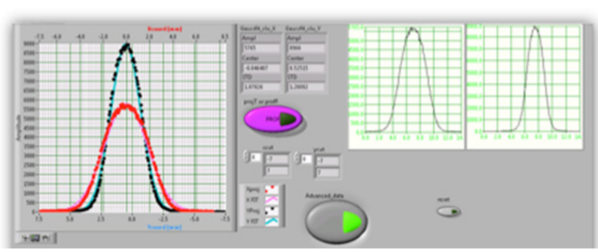

Figure 2: Cumulative or bunch per bunch transverse profile data and fitting.

ISBN 978-3-95450-182-3

In addition, we have developed a combined C/ROOT code, released freely on [11], that fetches MC keys and integrates ROOT[12] library to save ROOT trees, both in single and in multiple FitPIX configurations. This software makes a wide use of the information encapsulated in the data header, also allowing data integrity checking, e.g. by comparing the frame number vs. frame time of bunch position vs. time as shown in figure 3.

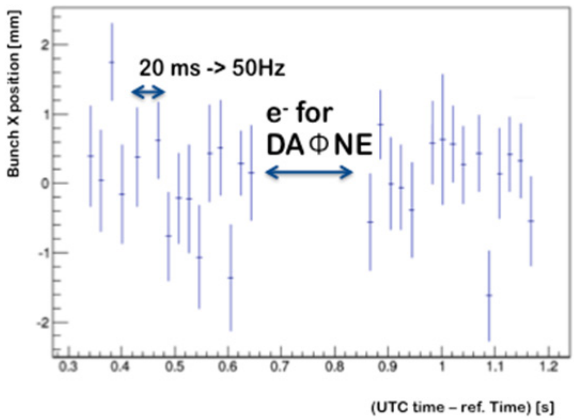

Figure 3: Bunch position monitored by the ROOT Cosumer software. The 50 Hz of the DAFNE LINAC bunch repetition rate and the timing used for the DAFNE dumping ring are marked.

This is a simple comparison but very useful when acquiring multiple FitPIX devices: after getting the MC Key, it gives the chance to discard aged, non-synchronous data. This code is provided as an example code for FitPIX integration in users DAQ software.

Data Synchronization with Users DAQ

Thanks to the usage of the MC functions, the integration in any user code is now just a matter of few plain C code include and calls. Allowing the BTF users such an easy integration in any, heterogeneous DAQ software, already during the first phases of the beam-time, was indeed one of the major objectives of this work.

From the point of view of consumer (C code) data fetching, Table 3, we show that the consumer software rejection function senses a single event jitter delay of 5ms (the minimum time-tag difference between two synchronized FitPIX MC keys) after 100000 calls, mainly due to producer thread time jitter.

Table 3: Two Detector, Low Data Regime, Consumer Timing

Frame	Max Delay [s]	framerate [Hz]	Rejected
100000	0.010	49.98	0
100000	0.005	49.95	1
100000	0.002	49.95	6
100000	0.001	49.63	486

This promises a complete software data alignment within the user DAQ cycle, even for the data coming for two FitPIX devices in the BTF typical tracking setup (shown in Fig. 4). Users can easily implement this integration after having set NTP synchronization on their PC, before performing one of these actions: fetching the MC keys in a separate database for a delayed offline data-matching; applying our sample code in their DAQ cycle, when providing a trigger signal to the FitPIXs; or, if the DAQ cycle is fast enough, even without any hardware synchronization, just matching the time of the two data streams (DAQ and FitPIX).

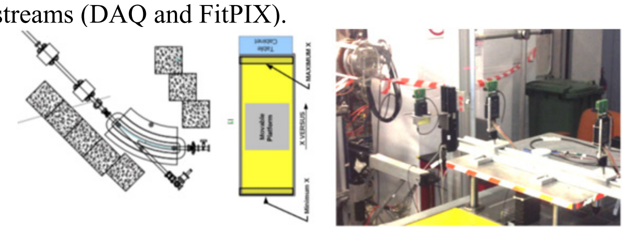

Figure 4: Three FitPIX Layout in BTF.

ACKNOWLEDGEMENT

This work is supported by the Horizon 2020 project AIDA-2020, GA no. 654168.

We would like to thank Daniel Tureček and Pavel Soukup, ADVACAM, for the productive involvement sharing in low-level libraries and hardware development. We are grateful also, among our users, especially to Francesco Renga and Giovanni Tassielli: their requirements are continuously go-ahead our facility for upgrading.

REFERENCES

[1] B. Buonomo *et al.*, “The Frascati LINAC beam test facility performances and upgrades”, *Incontri di Fisica delle Alte Energie - IFAE 2016* Il Nuovo Cimento C DOI: 10.1393/ncc/i2017-17069-6

[2] L.G. Foggetta, B. Buonomo, and P. Valente, "Evolution of Diagnostics and Services of the DAÎ¦NE Beam Test Facility", in *Proc. 6th International Particle Accelerator Conference*, Richmond, VA, USA, 2015, paper MOPHA049, pp. 904-906

[3] P. Valente *et al.*, "Frascati Beam-Test Facility (BTF) High Resolution Beam Spot Diagnostics”, in *International Beam Instrumentation Conference (IBIC16),* Barcelona, Spain, 2016, paper MOPG65, this conference.

[4] V. Kraus *et al.*, “FITPix — fast interface for Timepix pixel detectors”, Journal of Instrumentation, Volume 6, January 2011

[5] B. Buonomo *et al.*, in *International Beam Instrumentation Conference (IBIC16),* Barcelona, Spain, 2016, paper WEPG73.

[6] F.Murtas *et al*, “Applications of triple GEM detectors beyond particle and nuclear physics”, *Journal of Instrumentation*, Volume 9, January 2014

[7] https://memcached.org/

[8] http://libmemcached.org/

[9] http://chaos.infn.it/

[10] A. Stecchi *et al.*, "!CHAOS Status and Evolution", in *Proc. 6th International Particle Accelerator Conference*, Richmond, VA, USA, 2015, paper MOPHA046, pp. 894-896

[11]https://svn.lnf.infn.it/filedetails.php?repname=btf_science&path=/trunk/MC_btf/MC_read_tracker/readMC_tracker_Example.cc

[12] Rene Brun *et al.*, “ROOT - An Object Oriented Data Analysis Framework”, *Nucl. Inst. & Meth. in Phys. Res.* A 389 (1997) 81-86. See also http://root.cern.ch/