

ONLINE MONITORING SYSTEM FOR THE BM@N EXPERIMENT

I.R. Gabdrakhmanov^a, M.N. Kapishin^b, S.P. Merts^c

*Veksler and Baldin Laboratory of High Energy Physics, Joint Institute for Nuclear Research,
6 Joliot-Curie, Dubna, Moscow region, 141980, Russia*

E-mail: ^ailnur@jinr.ru, ^bkapishin@jinr.ru, ^cmerts@jinr.ru

The BM@N experiment is the crucial stage in the technical development of the NICA project. In order to effectively maintain experiment it is extremely important to have a uniform for all detectors, fast and convenient tool to monitor experimental facility. The system implements decoding of the incoming raw data during the run, preprocessing and visualization on the web-page. Users can monitor any detector subsystem, select specific detector plane/station, time or strip profile histograms in 1/2/3D view. The system is developed as a part of the BmnRoot package using the CERN jsROOT library. The lighttpd web server is used.

Keywords: BM@N, NICA, ROOT, jsROOT, C++, ZeroMQ

© 2017 Ilnur R. Gabdrakhmanov, Mikhail N. Kapishin, Sergei P. Merts

1. Introduction

The [BM@N](#) (Baryonic Matter at Nuclotron) is a fixed target experiment based on the Nuclotron-M [1]. It was developed to study heavy ion collision and search for phase transitions in the dense baryonic matter. It is built as the first stage of the NICA (Nuclotron based Ion Collider Facility) project. The experiment includes high precision tracking system with time-of-flight measurements for particle identification and calorimeters for total energy measurements. As a typical fixed-target experiment, [BM@N](#) includes a dipole analyzing magnet with variable magnetic field up to 1.2 T that allows searching for optimal acceptance and momentum resolution for every run configuration.

During experimental runs it was very important to have an online system for monitoring quality of experimental data. For this reason the fast decoding algorithm and monitoring system were implemented as a library into the BmnRoot framework. The BmnRoot is a program package for simulation, reconstruction and physical analysis for the [BM@N](#) experiment. It is based on the FairSoft (FairRoot) framework which was developed by GSI for experiments at the FAIR complex.

2. Structure overview

Functionally system consists of two separate parts as one can see in Fig. 1. The first one is a RawDataDecoder which gets raw data event-by-event from DAQ system, decodes it to the suitable ROOT object format – “digits” and transfers it to a ZeroMQ socket [2]. The second is the BmnMonitor – it reads data out from the ZeroMQ socket and fills the corresponding histograms. Histograms in turn are made accessible for external http requests by ROOT THttpServer [3] via FastCGI interface [4].

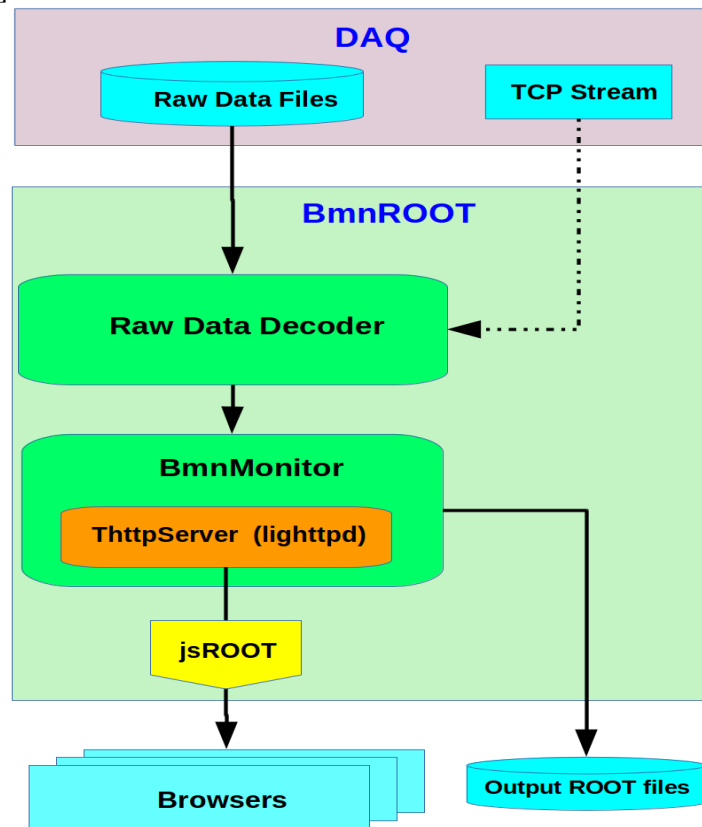


Figure 1. Online monitoring scheme

2.1. Decoding

The Decoding module consists of two main parts: a converter and a decoder. Fig. 2 shows this scheme.

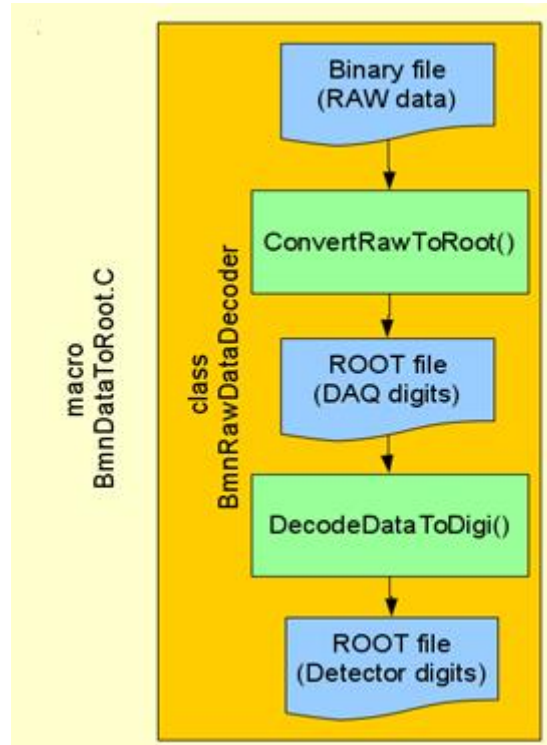


Figure 2. BmnRawDataDecoder scheme

The main goal of the converter is to gather useful information about so called DAQ-digits (ADC, TDC, HRB, TQDC, ...) from binary files according to the DAQ format and writing it into ROOT tree. Besides, the converter is reading out meta information about the current run which is stored in the binary file as well.

The main goal of the decoder is more complicated and consists in decoding of DAQ-digits using channel-strip mapping and converting them into the format appropriate for reconstruction program.

Let $ADC_i(k)$ is a raw signal on strip i for event k .

For signal decoding we have to know pedestal (average inner noise) for each channel P_i . It is calculated by special events without beam using the:

$$P_i = \frac{1}{M} \sum_{k=1}^M (ADC_i(k) - CMS_i(k)),$$

where $M = 1000$ is the number of events chosen for pedestals calculation, $CMS_i(k)$ is a common mode shift (average value of noise on chip) for data without beam. It is calculated using the following formula:

$$CMS(k) = \frac{1}{N} \sum_{i=1}^N ADC_i(k),$$

where $N = 32$ – the number of channels per one chip.

A corrected signal is a raw signal without pedestal and common mode shift:

$$S_i(k) = ADC_i(k) - P_i - CMS(k)$$

At the last step all signals below the threshold (15 ADC units) are excluded.

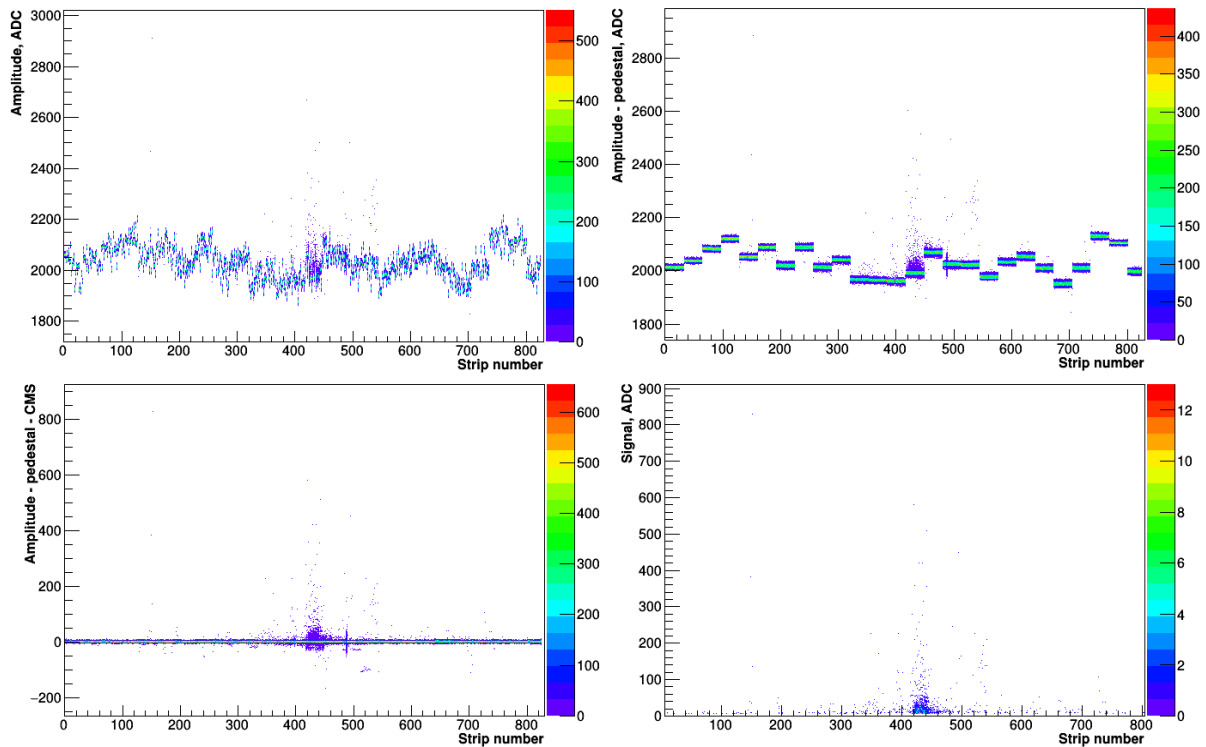


Figure 3 shows the all stages of the signal clearing from raw data to final signal for some plane of the inner tracker.

2.2. ZeroMQ library

ZeroMQ is a fast, lightweight and functional C++ library, which significantly simplifies network application development. Besides, it implements memory cached message queue, which in some cases greatly improves application performance.

In the [BM@N](#) monitoring system it gives a huge flexibility allowing us to distributively run decoder and monitor processes either on one machine (as well as in the different threads of one process) or on different machines located in different subnetworks. Such situation may arise if the information security is needed to be preserved and the DAQ subsystem is isolated from the outside network. Also several monitoring processes can get data from one decoding process, so shifters can use several monitoring processes without affecting each other work.

2.3. Frontend

The ROOT THttpServer class makes the ROOT object available for http requests particularly via FastCGI interface. One can request objects in the json or binary format. Here it should be noted that on the client machine there can be no ROOT installed in the system. So we need to convert a received object to the human readable form in a universal platform-independent way. The CERN JavaScript ROOT is a javascript library which converts ROOT objects to the html format allowing users to interact with them in the same way. We used [lighttpd](#) [5] web server for the web-page with the digest account authentication.

The system is capable to provide a detailed information about detector subsystems. The user is able to select the exact plane, strip and strip side for example for the ToF400 detector as one can see in Fig. 4. Also the system allows users to select reference run from past runs and impose it on the current run in order to detect possible deviations in the histogram form as can be seen in Fig. 5.

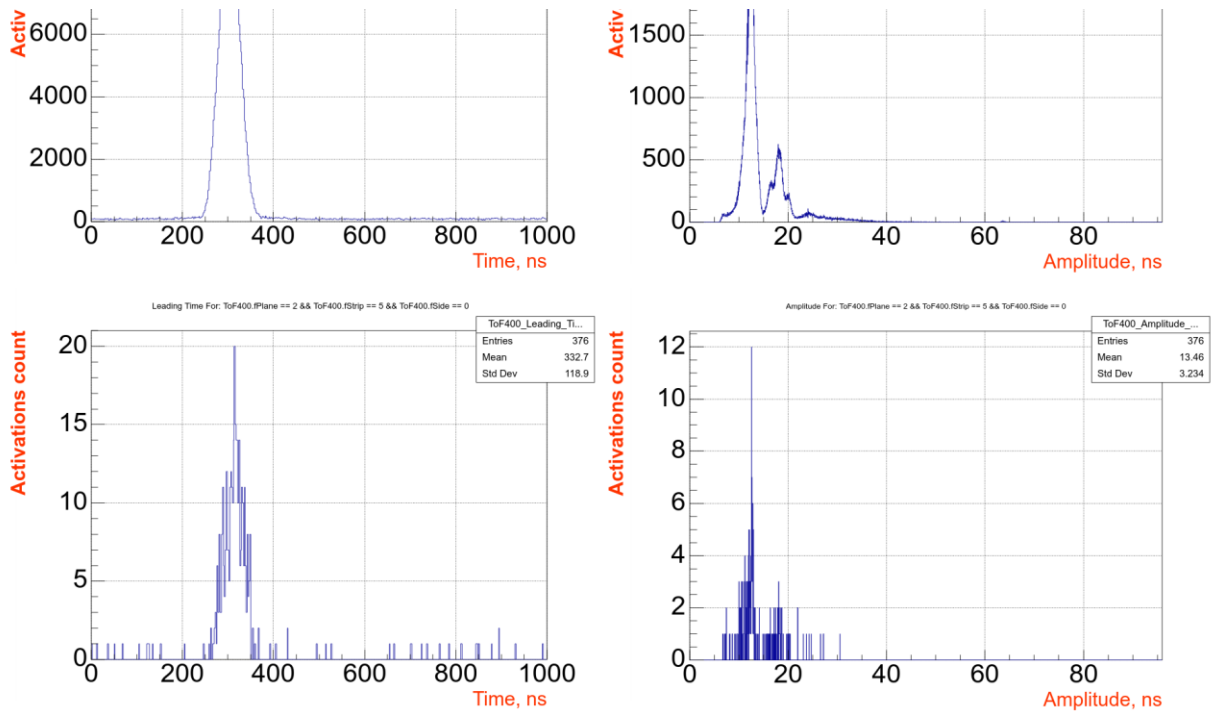


Figure 4. ToF400 overall and partial histograms

Authentication assumes two types of accounts – administrator and guest. Guests can only observe data while administrators (shifters conducting a run) can reset any data sets accumulated during the current run. Web based interface implies platform independence. Namely shifters can access monitoring (bmn-mon.jinr.ru) from any device connected to the JINR subnetwork: PCs, laptops, tablets or smartphones.

Run: 1502

Event: 92733

Run Type: target

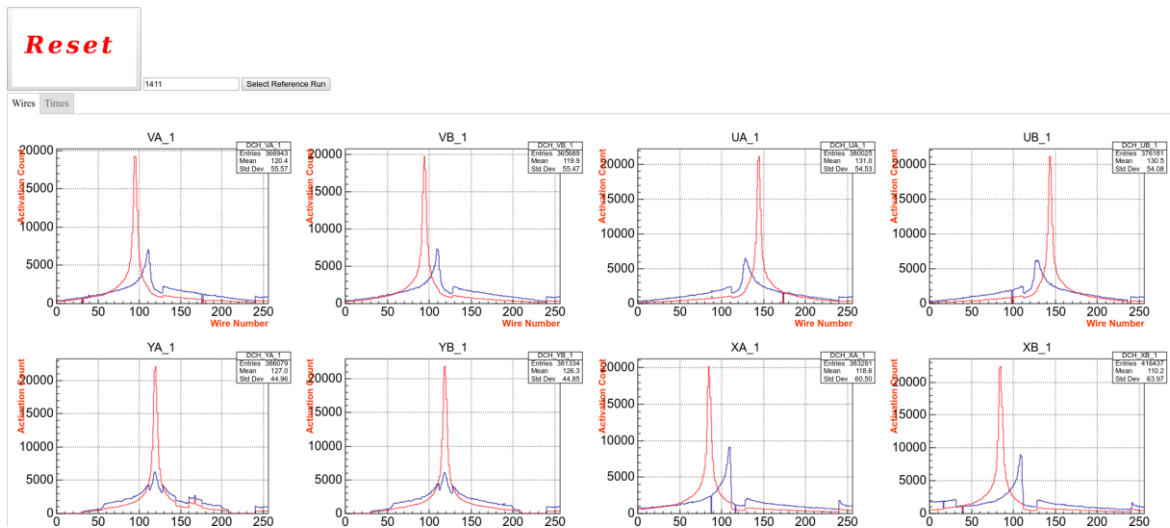


Figure 5. Drift chamber with imposed reference histograms

3. Conclusion

The developed [BM@N](#) online decoding and monitoring system is flexible and uniform for all detector subsystems. It can work distributively on several machines which allows choosing the most appropriate configuration for every experimental setup thus making it possible to use the system (with little changes) for other high energy experiments. At the same time it has convenient web based interface which makes it simple to observe experiment data flow from any device.

References

- [1] BM@N Conceptual Design Report (BM@N collaboration), http://nica.jinr.ru/files/BM@N/BMN_CDR.pdf
- [2] <http://zeromq.org/>
- [3] Adamczewski-Musch J. and Linev S. THttpServer class in ROOT, J. Phys. Conf. Ser. 664 (2015) no.6, 062032.
- [4] Bellenot B. and Linev S. JavaScript ROOT, J. Phys. Conf. Ser. 664, no. 6, 062033 (2015).
- [5] <https://www.lighttpd.net/>