THE UNIX-BASED DATA ACQUISITION SYSTEM FOR MULTICHANNEL NUCLEAR PHYSICS SETUPS

A. Yu. Isupov[1], V. E. Kovtun[2], A. G. Foshchan[2]

[1] *Joint Institute for Nuclear Research, Dubna, Moscow region, Russia*
[2] *V. N. Karazin Kharkiv National University, Kharkiv, Ukraine*

The data acquisition system (DAQ) for the Tandem setup (INR NASU) is proposed on base of the *ngdp* framework under the freely distributed UNIX-like operating system (OS) FreeBSD. The *ngdp* framework provides the software modules to organize an experimental data streams locally and remotely avoiding the intermediate data storages on the slow medii. The CAMAC subsystem is implemented on base of the camac package (DLNP JINR) using the ***kh*(*4*)** driver for the CC02 crate controller (PTP KhNU). The online visualization is based on the data event-by-event representation for the ROOT package. Such data could be feeded for more than one ***r2h*(*1*)** histogram servers, each of them is able to conversate with more than one ***histGUI*(*1*)** clients. This client depends on neither DAQ nor ngdp libraries so can be compiled under any OS equipped by the ROOT package.

1. Introduction

The data acquisition system (DAQ) for the Tandem setup (INR NASU) is proposed on base of the *ngdp* framework [1] under the freely distributed UNIX-like operating system (OS) FreeBSD. The *ngdp* framework provides the software modules to organize an experimental data transfer in the form of the so called packet streams and to distribute their locally and remotely in the very efficient manner, because the unnecessary intermediate data copying and storages as well as the context boundary crossings are avoided. This framework avoids the intermediate data storages on the medii slower than the computer operating memory. The *camac* package (DLNP JINR) [2] used as the CAMAC subsystem implementation separates the hardware specific drivers from another software parts, which could be written in the hardware independent manner. The *ngdp* framework and *camac* package using allows us to limit the software design efforts by the Tandem setup specific modules only: the CAMAC interrupt handler and the C++ class for the events representation for the ROOT package [3], because all other mentioned below software modules are already ready.

2. DAQ Tandem setup INR NASU

As a nuclear reaction result in the most cases the more two particles are born in the final state. For proper analysis of these many-body reactions the full kinematics reconstruction is required, which could be done by measurement of the time and energy correlation spectra of the reaction products. This brings to need of the many parameters experimental measurement by the semiconductor detector telescopes. Accordingly, the multivariate analysis of the coincidences is required to provide from the miscellaneous pair of the telescope detector $E_i \times \Delta E_i \times E_j \times \Delta E_j \times t_{ij} \times N_D$, where i,j - a telescope number, Ei, Ej - reaction products energy, which registered by these telescopes; ΔE_i, ΔE_j – energy loss of the particles in ΔE-detector; N_D – identifier code of the telescope pair was triggered; tij - a time spectrum of the coincidences. As a rule, the data rate in these experiments is too low. This leads to multichannel setup on base of the specific CAMAC modules, which is able to measure many values simultaneously, and off-line software for multiparametric sorting and analysis. Proper methodic for the multiparametric studies of the nuclear reactions in the full and incomplete kinematic is required. For the excitations and decay of the $^{6,7}Li^*$ unbounded states in three-particles reaction on the cyclotron U-120 and tandem-generator EGP-10K (INR NASU) this methodic was described in [4]. The proposed Tandem DAQ is able to fill and draw one- and two-dimensional histograms to show spectra and correlations during on-line.

3. Features of the proposed DAQ system

The CAMAC subsystem is implemented on base of the *camac* package (DLNP JINR) [2], which encapsulates the crate controller dependent code into the so called drivers. These drivers could be easily implemented from scratch and replaced without serious changes in another code parts. For the CC02 CAMAC crate controller [5] designed in the Karazin Kharkov National University the ***kh*(*4*)** driver is already implemented by authors. So, the only user suppliable software module to be designed is the CAMAC interrupt handler ***tandem*(*4*)**, which should encode the actual work with the CAMAC hardware of the Tandem setup. This module in the many aspects is close to the already written interrupt handler ***combas*(*4*)** for the COMBAS setup (FLNR JINR). For each serviced interrupt from CAMAC the ***tandem*(*4*)** checks an interrupt source and for the recognized trigger the handler reads the CAMAC hardware and encapsulates obtained data into the *ngdp* packet. According to the usual *ngdp* data flow scheme these packets are supplied to the ***ng_camacsrc*(*4*)** module for injection into ***netgraph*(*4*)** for the local and/or remote data transfer. The simplest graph, which solves our problem, is `ng_camacsrc` → `ng_fifos`. The ***ng_fifos*(*4*)** module is able to produce more than one identical output packet streams, for example, one of them for the ***writer*(*1*)** utility and another – for binary-to-ROOT converter ***b2r*(*1*)**.

The online data processing is based on the data event-by-event representation for the ROOT package. This representation is implemented in a form of the special C++ class(es), whose design principles and requirements are described in [6]. For the Tandem setup the single Etandem ROOT class representing the trigger events will be enough, because we should not handle the data coupled with the accelerator cycle. The Etandem class could be very similar to the Ekeyaf class already implemented for the QUADRO setup [7] (PTF KhNU). Note, that both ***b2r*(*1*)** and ***r2h*(*1*)** should be relinked (however not rewritten) with the *Etandem.o* object code after each Etandem class change.

So, each packet of the trigger event is converted into the Etandem class instance by the ***b2r*(*1*)** utility. These instances could be transferred locally or remotely to the runtime configurable ***r2h*(*1*)** histogram server(s). Of course, the data could be written onto a hard disk in the each of (or both) packet and ROOT event representations for the final storage by the ***writer*(*1*)** and ***b2r*(*1*)**, correspondingly. The ROOT event representation scheme allows us to handle also the data coupled with the accelerator cycle, if this is required. Each ***r2h*(*1*)** works according to the current own configuration and could serve the one or more ***histGUI*(*1*)** client modules.

This configuration done by the interaction protocol (see [6] for more details) between the histogram server and visualization client(s), which, in particular, allows one to:

- declare the value in terms of integer triplet “channel, module, group of modules” (details of correspondence between triplets and event data fields are hidden in the Etandem class implementation);
- add to any already declared value another triplet to form the list of values (each list of values allows to unite together the number of detector channels);
- declare the TH1F (TH2F) histograms in terms of one (two) value(s) (or list(s) of them);
- transfer to client any already declared histogram for drawing once or many times;
- delete any declared (list of) value(s);
- clear / delete any declared histogram.

Additionally to describe in [6] the calculated values (cells) as well as conditional histogram filling are implemented. The command Cvar for cell declaration is introduced:

Cvar – declares the name varname of the variable of type restype to be calculated at each event of type evtype obtaining by the supplied prog of calc. This program of cell result calculations is C-string, beginning after the evtype and ending at end of line. It is a right part of C expression (without assignment), where operands may be a constant value, already declared cell (including current cell) Cvar, variable Var, or other valid expression. Following operations are implemented:

unary: ‘+’', ‘-’, ‘~’, ‘!’

binary: ‘+’', ‘-’, ‘*’, ‘/’, ‘= =’, ‘!=’, ‘<’, ‘>’, ‘<=’, ‘>=’, ‘&&’, ‘||’, ‘<<’, ‘>>’, ‘&’, ‘|’, ‘^’ ternary: ‘? :’

Calculations performed in order, defined by C operation priorities, and may be changed by operands grouping in parentheses, ‘(‘and‘)’. The following functions of the one argument are implemented (see ***math*(*3*)**):

‘sin’, ‘cos’, ‘tan’, ‘asin’, ‘acos’, ‘atan’, ‘sinh’, ‘cosh’, ‘tanh’, ‘asinh’, ‘acosh’, ‘atanh’, ‘exp’, ‘expm1’, ‘log’, ‘log10’, ‘log1p’, ‘sqrt’, ‘cbrt’, ‘fabs’, ‘erf’, ‘erfc’, ‘j0’, ‘j1’, ‘y0’, ‘y1’,

as well as ‘pow’ with two arguments.

Note the fact, that all values are internally converted to double type, so calculations are performed with double values (see ***math*(*3*)** for details about guaranteed precisions, number of significant decimals, etc.). This can lead to some results, unexpected from the integer arithmetic point of view. We should try to avoid, in particular, long integer constants.

So, the Cvar has the following format:\\

Cvar varname restype evtype <prog for calc>,

for example: Cvar tdc0s1 type_Float TANDEM_DAT_0 (tdc0 - tdc1 + 1024)

The one optional parameter fdep is added for Book1d and Book2d commands. This is an already declared Cvar of type type_UChar or Var variable. The corresponding histogram will be filled, if such fdep is nonzero, and will not otherwise.

So, each client-server interaction as well as entry in the ***r2h.conf*(*5*)** configuration file is a command of the protocol mentioned above. Of course, if our setup has stable configuration, the routine value and histogram declarations could be written into the configuration file to be processed at the ***r2h*(*1*)** startup instead of to be entered by the user interactively again and again. The abilities to have several remote histogram servers and to delete the histograms during the runtime are very useful to overcome the operating memory and/or processor performance limitations of the single computer, particularly if we have huge TH2F histogram(s).

At the each arrival of the Etandem class instance from the ***b2r*(*1*)** the ***r2h*(*1*)** assigns all declared values (and lists of them) and Fill()s all declared ROOT histograms. At the client request obtaining the ***r2h*(*1*)** serves it or responds by error code.

The visualization and minimal graphical user interface (GUI) are provided by the ***histGUI*(*1*)** client, which depends on neither DAQ nor *ngdp* libraries, so can be compiled under any OS equipped by the ROOT package. Minimal GUI contains the “Exit'” button (TGTextButton), the command string input field (TGTextEntry), the window(s) for histogram drawing (TCanvas), and the output text viewer area (TGTextView) to report errors. This GUI could be a subject for additional design to satisfy the experimental setup requirements. For example, the panels of buttons could be added to simplify viewing of some frequently used histograms. The processed at startup configuration file for the

***histGUI*(*1*)** is supported and could contain commands of the client-server protocol, for example, some routine actions like connection to the ***r2h*(*1*)** server and starting the histograms visualization. Moreover, the minimal client could be a ROOT script.

Conclusions

The proposed DAQ system should allow to visualize online the all key plots of the Tandem setup, so the beam time could be used very efficiently.

REFERENCES

1. *Isupov A.Yu.* The *ngdp* framework for data acquisition systems. // JINR Communications. - E10-2010-34. - 2010. - P. 1 - 20.
2. *Gritsaj K.I., Olshevsky V.G.* Software package for work with CAMAC in Operating system FreeBSD // JINR Communications. - P10-98-163. - 1998. - P. 1 - 16 (in Russian).
3. *Brun R., Rademakers F.* ROOT - An Object Oriented Data Analysis Framework // Proc. of the AIHENP'96 Workshop (Lausanne, Switzerland, 1996). - Nucl. Instr. and Meth. in Phys. Res. - 1997. - Vol. A389. - P. 81 - 86. See also http://root.cern.ch/.
4. *Nemets O.F., Pavlenko Yu.N., Shablov V.L. et al*. Decay of $^{7}Li^{*}$(7.45 MeV) unbound state in reaction $^{7}Li(\alpha,\alpha)^{7}Li^{*}$ // Программа и тез. Докл. 54 Междунар. совещ. по ядерной спектроскопии и структуре атомного ядра (Белгород, 22 - 25 июня 2004 г.). - С.-Петербург, 2004. - С. 171.
5. *CC02*. Руководство пользователя. - Харьков: ООО “РиЭС”, 2004. - С. 1. - 84.
6. *Isupov A.Yu.* CAMAC subsystem and user context utilities in *ngdp* framework // JINR Communications. - E10-2010-35. - 2010. - P. 1 - 20.
7. *Isupov A.Yu., Kovtun V.E., Foshchan A.G.* Implementation trial of the DAQ system for the compact physics setup on base of the *ngdp* framework // Fundamental problems and applications of nuclear physics: from space to nanotechnologies. 59 Int. Meeting on Nuclear Spectroscopy and Nuclear Structure (NUCLEUS-2009) (Cheboksary, Russia, 2009): Book of Abstracts / Ed. by A.K. Vlasnikov. - S.-Petersburg, 2009. - P. 346 (in Russian).