10th ICALEPCS Int. Conf. on Accelerator & Large Expt. Physics Control Systems. Geneva, 10 - 14 Oct 2005, PO1.091-8 (2005)

DESIGN AND DEVELOPMENT OF A CONTROL SYSTEM FOR INTENSE SOURCE OF RADIOACTIVE IONS PROTOTYPE

D. Bolkhovityanov, N. Lebedev, A. Tsyganov
The Budker Institute of Nuclear Physics, Novosibirsk, Russia

ABSTRACT

The experiment named "Proton Accelerator Based Intense Source of Radioactive Ions for Nuclear Physics Experiments" is carried out jointly by The Budker Institute of Nuclear Physics and INFN – Laboratori Nazionali di Legnaro, and is supported by ISTC (project #2257).

The hardware used for automation has diverse interfaces: CAMAC (digital oscilloscopes), CAN-bus (DACs and ADCs), Ethernet (CCD-camera and Tektronix oscilloscope) and RS485 (stepper motor and pyrometer).

Since the experiment is relatively small-scale, we had a choice to create several standalone programs, which perform all automation tasks. Fortunately, we have chosen another approach – to employ the same control system which is used on VEPP-5 complex – CX, which has 3-layer architecture.

One of the main reasons for this choice was experimental nature of the work, so that requirements were expected to change dramatically and unpredictably. And often physicists can't tell what would they need in the next days. In the course of experiment these expectations proved to be true.

This paper presents our experience from this work and solutions we used.

THE EXPERIMENT

The SPES project at LNL aim to produce intense radioactive ion beams by fast neutron inducing fission on uranium carbide targets. The fast neutrons are generated by proton beam in a thick graphite converter of sufficient thickness to stop all the protons. The fast neutrons impinge on a thick target of fissionable material to produce fission fragments. The converter is designed to dissipate more than 100 kW of beam power. The graphite material was selected because its excellent physical and chemical properties allow high beam intensities with a rotating wheel cooled mainly by thermal radiation.

BINP is responsible for design and production of the rotating target prototype. The rotating target is irradiated with electron beam, and target's behaviour under high temperatures (>2000°C) and at high temperature gradients (>100°C/mm) is evaluated. Photometric methods are employed: temperature distribution field is measured with either CCD-camera or a photodiode line via different colour filters.

A distinctive feature of the experiment is a ^{13}C target (predicted to have higher neutron yield), so the prototype experiment is nicknamed "C13".

AUTOMATION TASKS AND HARDWARE

The following hardware, in large part produced by BINP, is used:

- CCD-camera with Ethernet interface (custom protocol over UDP) is used for 2D diagnostics of the temperature distribution on the target surface.
- ADC333 CAMAC digital oscilloscope is employed in the second method of temperature distribution diagnostics (1D), reading a photodiode line. Plus, a pair of ADC333s were intended for "fast" monitoring of target wheel vibrations. BINP-designed CM5307-PPC intelligent CAMAC-controller[1] is used.
- CADC40 40-channel CAN-bus ADC performs measurement of thermocouples, beam parameters, vacuum and "slow" vibration data. CDAC16 16-channel CAN-bus DAC does control of wheel's linear motor, CCD-camera's objective control, and some more tasks. Each of these devices also includes 8-bit input and output registers, which are used in control.
- Colour filters' wheel is rotated by a stepper motor with a KShD-485 controller, which uses RS485 interface.
- Calibration if performed with Impac IS10 pyrometer, which also uses RS485 interface.

These hardware is connected to a PC, which both performs control and provides operator's interface.

CHOICE OF CONTROL SYSTEM

So, the list of hardware is short, and the experiment is controlled by a single PC. Such small-scale experiments are often automated with a single, standalone program, which deals with hardware and provides operator's interface. But there are two contras:

- First, the control hardware is too diverse, and such a standalone application, implementing "drivers" for all these devices, would be almost as complex as a general-purpose control system.
- Second, requirements for the control software were expected to change. But making changes[1] into such a big, complex and interweaved program inevitably leads to errors.

So, the choice of control system architecture wasn't obvious. 3 variants were considered:

1. A dedicated, standalone "do everything" program.

---

[1] Which are often unpredictable and even unjustified.

2. An industrial small-scale automation tool, like LabView.
3. A regular, full-featured 3-layer control system.

The 1st variant was used in the automation of the "prototype's prototype". It gave us a chance to understand and to feel that this way is too complicated and error-prone.

Use of a ready small-scale solution, available on the market, didn't look attractive either. First, drivers for not-so-common hardware we use had to be created for such a system anyway. And, second, we have no trained personnel for it.

A distributed 3-layer control system looked a bit excessive, since its use introduces some "implementation overhead", while many of its features are simply not needed in our case. On the other hand, this way has no other disadvantages (inherent in two previous approaches), and makes implementation of control tasks very straightforward.

So, finally the 3rd way was chosen. And the main reason was availability of such a system — CX[2], which is used at VEPP-5 complex in BINP, by the same people who are involved in the "C13" experiment.

CX runs under Linux, which stipulated use of Linux as a control PC's OS.

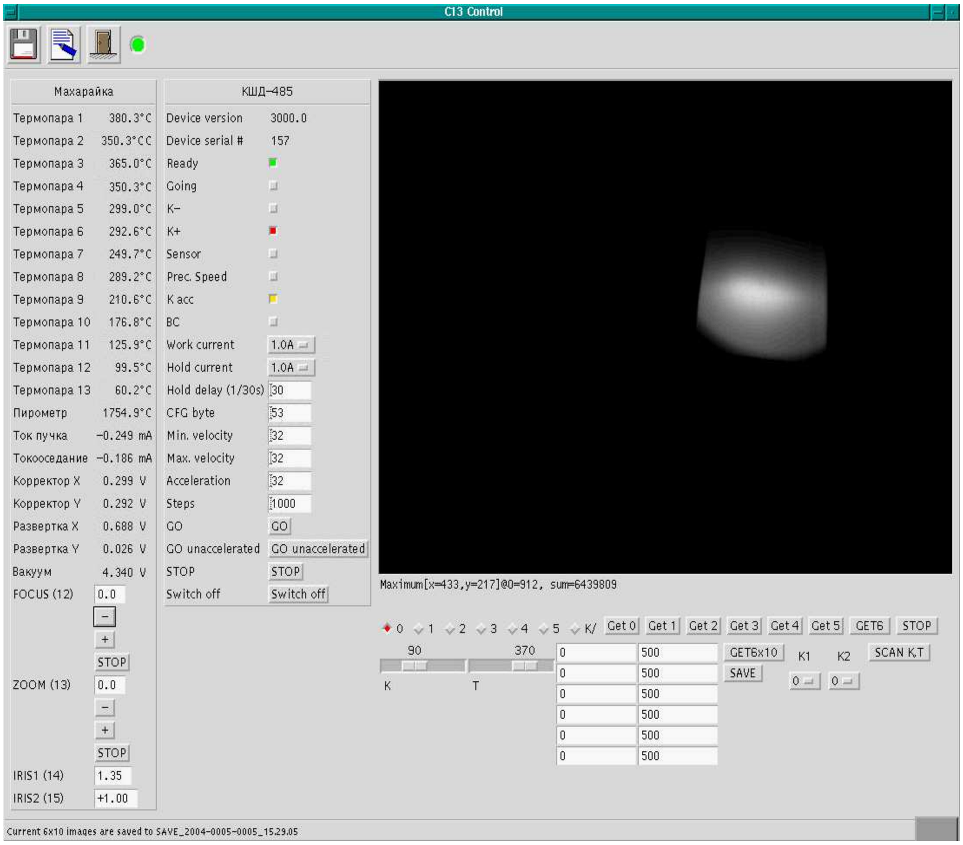
IMPLEMENTATION

Photometric measurements

Photometric program was designed to be the main application — a "control center", allowing to control most aspects of the experiment. It is the most complicated application. Its screen (see fig.1) is divided into three areas.

The left one provides miscellaneous measurements and controls: temperature measurements from thermocouples and pyrometer, electron beam parameters, target wheel parameters and control, CCD-camera objective control. The second, related to KShD485, allows manual control of the colour wheel. And the main place — to the right — is devoted to photometry: a picture from CCD or a calculated temperature field, tuning "handles", and automation control — "perform the measurements" buttons.

A second variant of this program is used for photometric measurements with photodiode line. It allows similar operations, with difference reflecting the specifics of a photodiode line measurements. Its user interface is improved, and a slightly different set of measurement and control channels is provided (since this application was used later in the course of experiment, and hardware channels have changed).



Махарайка	
Термопара 1	380.3°C
Термопара 2	350.3°CC
Термопара 3	365.0°C
Термопара 4	350.3°C
Термопара 5	299.0°C
Термопара 6	292.6°C
Термопара 7	249.7°C
Термопара 8	289.2°C
Термопара 9	210.6°C
Термопара 10	176.8°C
Термопара 11	125.9°C
Термопара 12	99.5°C
Термопара 13	60.2°C
Пирометр	1754.9°C
Ток пучка	−0.249 mA
Токооседание	−0.186 mA
Корректор X	0.299 V
Корректор Y	0.292 V
Развертка X	0.688 V
Развертка Y	0.026 V
Вакуум	4.340 V
FOCUS (12)	0.0
ZOOM (13)	0.0
IRIS1 (14)	1.35
IRIS2 (15)	+1.00

КШД−485	
Device version	3000.0
Device serial #	157
Ready	🟩
Going	☐
K−	☐
K+	🟥
Sensor	☐
Prec. Speed	☐
K acc	🟨
BC	☐
Work current	1.0A
Hold current	1.0A
Hold delay (1/30s)	30
CFG byte	53
Min. velocity	32
Max. velocity	32
Acceleration	32
Steps	1000
GO	GO
GO unaccelerated	GO unaccelerated
STOP	STOP
Switch off	Switch off

Maximum[x=433,y=217]@0=912, sum=6439809

● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ K/

0	500
0	500
0	500
0	500
0	500
0	500

K1: 0 K2: 0

Current 6x10 images are saved to SAVE_2004-0005-0005_15.29.05

Figure 1: Color wheel and CCD-camera control application

Vibration measurements

Two types of vibration measurements are performed. “Fast”, which give a real-time picture and allow early detection of target wheel cracks, and to stop the wheel before damage becomes fatal. “Slow” measurements (displayed in a time scale) allow visual evaluation of vibrations development and their correlation with other factors; plus, these allow post-mortem analysis.

In the course of experiment TDS3032 oscilloscope was found to be adequate for “fast” measurements, so only “slow” ones were left for a control system.

The “slow measurements” application reads vibration sensors data and displays it in both numeric and recorder form. Plus, for convenience, it displays some related data and allows wheel motor control. This application is simple and straightforward.

Pyrometer

Automation of Impac IS10 pyrometer consisted of a “general” and “custom” work. The former included creating a device driver for CX (see below) and writing a control application, similar to Impac’s Windows-only InfraWin tool.

The custom part consists of an application which employs IS10 for temperature distribution diagnostics, using some specific knowledge of its operation. And *that* is the program that should have better been written by physicists themselves: it has no programming specifics besides access to data via CX.

Unfortunately, currently CX programming is too difficult for most physicists. That’s a vivid and valuable lesson.

Device drivers

CX provided ready CAMAC and CAN support, so only KShD485 and IS10 drivers had to be written specifically for “C13”. This task consisted of two parts.

First, fitting devices’ resources and models of operation into a paradigm of channels. This was easy, albeit revealed that KShD485 protocol is a bit poorly designed.

The second part was implementation of drivers’ code, dealing with serial interface specifics. This includes intelligent message queueing, retransmits upon timeouts, dealing with connection loss and recovery.

While KShD485 uses a binary protocol and IS10 — a text one, server-side support for them is similar. And, in both cases implementation is more complicated than that for an average CAMAC or CAN device.

EVOLUTION

From the very beginning several types of changes were expected.

First, the photometric measurements had to be performed via different methods: CCD-camera and photodiode line. This was a simple engineering task.

Second, the hardware was constantly modified according to the intermediate results and changing requirements. This included changes in the set of control channels, calibration coefficients and formulae. To reflect this in software was a boring, but easy task.

Third, changes of algorithms and approaches during the course of experiment, according to change of understanding of the physical processes. And *that* changes were the most exhausting and time-consuming. Even the relatively small, specialized programs had changed dramatically, and in the case of a single, monolithic application this would be hardly doable.

Most changes applied to the "experiment control center" application. The first, CCD-camera variant, have undergone major changes, so finally its code became too complicated. The photodiode-line variant was written from scratch. But during its life the code was heavily modified too, so that now it is far from beauty.

On the one hand, many of these changes were unavoidable, since some requirements weren't known in advance, being discovered in the course of experiment.

But: much work was caused by the fact that physicists didn't took the trouble to think through the task thoroughly, preferring to lay the burden of continuous modification of software on programmers. Physicists' reason is: "it's just a program, it isn't hardware, it's so easy to change when we want!".

So, probably the best way for programmers to perform automation of such experiments[2] is to give physicists an ability to make control applications themselves, in the easiest and convenient way. But that's another story.

CONCLUSION

Application of a 3-layer distributed control system in such a small-scale experiment turned out to be a right decision. It allowed to escape many problems, unavoidable with other control system architectures. The set of control programs is larger than was planned at the beginning, and many of them access the same control data. While initial requirements specification didn't call for distributed and remote control, this ability turned out to be valuable. The only exception, which falls out of this scheme — the CCD-camera[3], confirms validity of the chosen way.

All future small-scale automation projects, conducted by our laboratory, will use this approach.

References

[1] D.Bolkhovityanov et al, "PowerPC-based CAMAC and CAN-bus controllers in VEPP-5 Control System", Proc. PCaPAC'2005, http://conference.kek.jp/pcapac2005/paper/WEB4.pdf

[2] D.Bolkhovityanov et al, "Evolution and Present Status of VEPP-5 Control System", Proc. PCaPAC'2002, http://www.lnf.infn.it/conference/pcapac2002/TALK/MO-P15/MO-P15.pdf

---

[2] And, probably, *any* automation

[3] Due to some problems it had to be accessed by the application directly.