

FLEXIBLE USER INTERFACE FOR COMPUTER MEASUREMENTS AND CONTROL

V.G.Kurakin, A.V.Koltsov

Lebedev Physical Institute, Moscow, Russia

P.V.Kurakin

Keldysh Institute of Applied Mathematics, Moscow, Russia

Abstract

We have developed computer code for physical data acquisition and physical equipment control in real time scale with the interface that user can adjust at the level of executable. Borland C++ Builder has been used for this code development, and the concept of interface building is the same that software mentioned uses. The essential difference is that form filling with the necessary components is moved to the executable level. Components used for physical data representation are numeric indicator, plot, histogram and spectrum. These can be adjusted during experiment and the procedure does not require compiling and linking. This is crucial for our approach since often the most dangerous errors occur after correction of the code written some time before. There are also the components that are responsible for data processing before their visualization as well as for animation of visualized data. The later means that any time dependent component used for data representation may be played back in slower time after triggering by so called external selection pulse. This procedure can be used only if waveform digitizer based on analogue-to-digit converter is used in the appropriate channel. The adjustment of interface is achieved in a manner similar to Borland technology that is due to adjustment of appropriate properties in property editor and the possibility of movement and sizing of the components on the top of the form. There is a possibility of importing picture onto the form, and this we use for creation appropriate mnemonic scheme on computer display. The developed code is used for automation of Lebedev Physical Institute Radiation Complex and for experiments on creation of η -mesic nuclei at Lebedev Physical Institute electron synchrotron. There are also plans to use this code for automation of the synchrotron mentioned.

1 INTRODUCTION

In computer based measurement systems hard ware interface that serves for input signal digitizing is rather conservative system part and limited number of hardware units is usually quit sufficient to carry out an experiment. It is not the case for data measured pre-processing and visualization. Depending on many requirements these are desired to be changed even during the experiment. In other words, the pre-processing data transforming as well as the way of data displaying are desired to be flexible. Up to recent time user interface that serves for measured

data graphical representation and apparatus controlling data entering was the most complicated and time consuming part of computer code. With the appearance of power programming tools at the end of ninetieth based on Borland visual component technology (Borland C++ Builder and Borland Delphi) programmers had got access to rapid application development technology. The time necessary for the development of user interface had dropped drastically even for rather sophisticated applications. On the other hand, even with this technology one has to create new interface for any new experiment. To adjust program in use for new experiment one has to modify existing code and to recompile it, and this procedure is very often an origin of difficult-to-locate errors. To avoid this, we have moved an interface modification to the executable file level thus having made next step in Borland Visual Component Technology. To adjust interface one has to fill in forms in a manner very similar to form filling in Borland Delphi or Borland C++ Builder environment. The essential difference is the configuration built does not need to be compiled and linked and the program can be started immediately.

2 USER INTERFACE

After launching the program under discussion the interface used last time is loaded. The main menu allows starting measurements or doing one of the following: to load any desired interface built up and saved previously, to modify current interface or design completely new interface. We will describe in details the last options mentioned. First of all one has to choose the devices that we be used for data collections as well as for the appropriate equipment control. These have to be chosen from component palette and placed on the top of the special windows (Data Module) that also has to chosen from component palette. Fig.1 is an example of interface designing – component palette is marked with number 2 and data module with number 3. When any component is chosen special editor (Component Option, similar to those marked with number 4 on Fig.1) is available and user can fix desired properties. For the current step of the interface under designing one can chose station number and unit active channel as well as fix other properties necessary for proper hard ware unit functioning. (In this paper we describe the situation, when CAMAC interface is used in hardware environment.).

Two mode of operation of hardware units installed are available – synchronous and asynchronous. To organize the synchronous mode of operation one has to create device group – first to drag group icon from component palette to Data Module (position 3 on Fig. 1) and then activate it in any way available in Windows operation system environment. Devices editor will appear, and one

has to drag all necessary icons from right to left. The difference between two modes of operation is the way of data collection and storage. Data from grouped devices are acquired and stored together those of master device from the same group. The master device is the unit with activated property “start” – its icon is marked with pulse symbol in Device Editor.

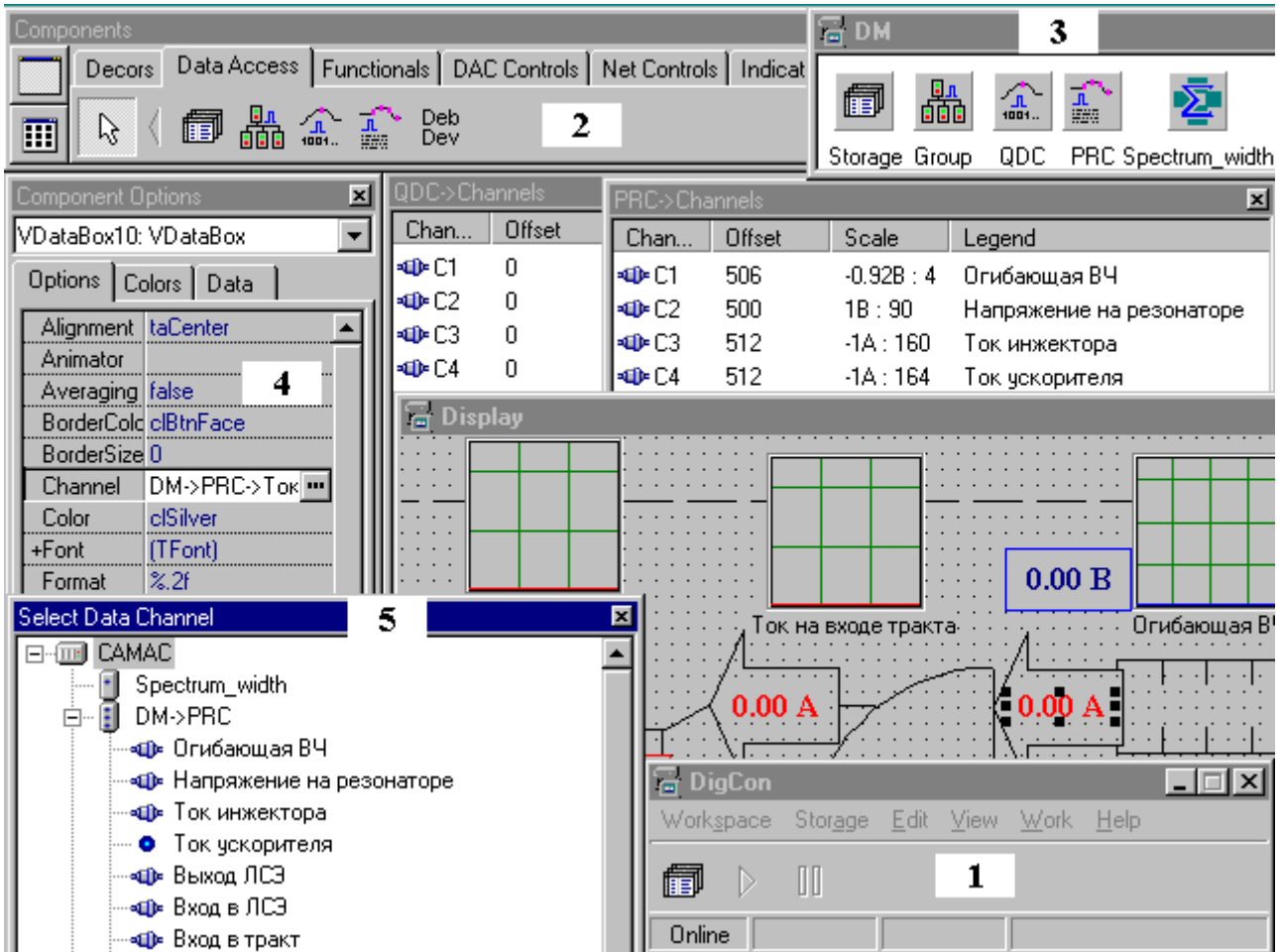


Figure 1: Program interface in design mode. 1 – the main menu, 2 – component palette, 3 – data module, 4 – component properties editor, 5 – selection of data channel.

The next step is the data visualization and hardware control interface building up. Several ways of measured data display are foreseen. These are numeric indicator, oscilloscope, histogram and spectrum. A numeric indicator is dimensional quantity measured at fixed moment. It can be arranged appropriately – for example, placed in frame and accompanied by the text (labeled). Oscilloscope is a plot of single channel process in coordinates amplitude – time. Histogram and spectrum serve to display multi channel data representation. The difference between them is that the former stores data in individual channels during some interval, while the latter displays a spectrum – data distribution among different channel at any fixed moment.

As usual in component based design technology used in visual programming, all components have to be placed on the top of a form, the number of forms being not limited practically. To create a component with desired functionality one has to activate it and fix necessary properties in Component Option editor (position 4 on Fig. 1). The window marked with number 5 is activated when one links the component under edition with data channel. Double clicking appropriate connector icon creates necessary link.

Animation mode can be used for any way of data visualization with the exception of histogram representation. We have chosen animation in favor of three-dimensional plot for time dependent multi channel process (spectrum) because find the former more visual.

This is especially true for low repetition time and for single events. The scenario for animation is as follows. Animation can be stimulated by so-called selection pulse that comes from a trigger to the plug socket of the unit faceplate (externally) or with any new impulse of synchronization (internally). Like in cinema sequent spectrums are displayed in slower rate. After animation the process can be continued or stopped – in last case one can play back the spectrum step by step (manually). All components under animation display the measured values at the moment chosen.

In order to process some data before visualization one has to chose functional component in Component Palette and drag it into Data Module. Activating this component brings one to Component Option window and to Code Editor (Fig.2). Following definite rules one can define transformation expressions that are compiled before closing editor. There is also the possibility to load necessary expressions from file as well to save expressions in use in file. After compiling the appropriate virtual channel is created (for instance, "Spectrum_width" in the window 5 on Fig. 1) and one can use it for data visualization as usual real digital data source.

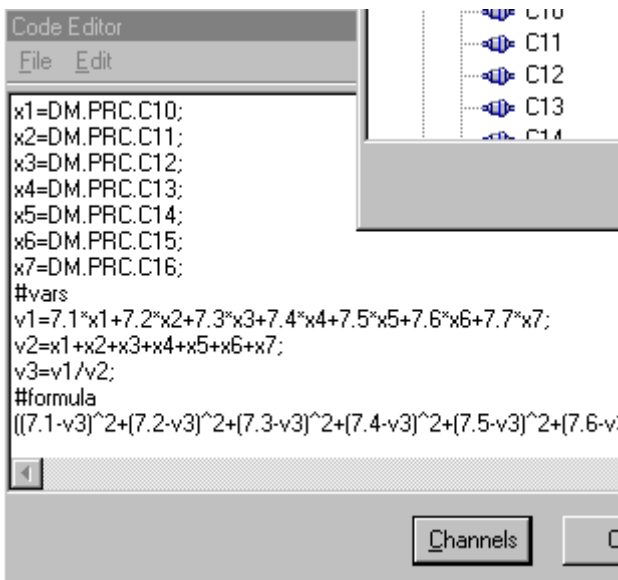


Figure 2: Code editor

3 PROGRAM CODE

Program code consists of several independent parts:

- The first one is the driver that controls data acquisition from digital devices as well as digital control. This part of code is device dependent and needs to be written for any digital device used in experiment. This part of the code includes also dynamic libraries of "*" .bpl type".
- System dynamic library forms the second main part. It includes the primary components that are parent for real components used to create appropriate interface for measurements and control. These

primary components determine the standard functionality as well as appropriate interaction interface and large number of auxiliary functions.

- Dynamic libraries consisting of components that are necessary to build up any desired interface. It is clear that the variety of components determines the flexibility of application that is determines to what extent one is free to create the interface he wants. In order to add any new component in library one has to add new code and to recompile and to link it together with existing code. This procedure is a programmer prerogative, and it is used to extent program functionality.
- The main program module that is responsible for user interface creation. This part of the code serves to adjust appropriate interface as described above.

The program code had been created for the use in Windows9* platform.

4 CONCLUSION

We have developed computer code that proved to be very convenient for computer based measurements. The essence of the code or the most powerful its feature is that user himself is able to build any desired interface of measured data representation on computer display as well as to process these data before their visualization according the desired laws. All these features are available at executable file level that is the work done does not require compiling and linking. This code is used for automation of data acquisition and processing at Lebedev Physical Institute Radiation Complex [1] as well as in the experiments on creation of η -mesic nuclei being carried out at the our electron synchrotron to the energy 1.3 GeV. There are plans to use the code for synchrotron mentioned automation [2].

5 ACKNOWLEDGEMENTS

The work in part had been made in the frame of the Russian Scientific Program "Microwave Physics", Prof. A.N.Lebedev and Dr. A.V.Agafonov being Program Coordinators. Hardware interface for the computer based measurements and control system has developed by V.P.Busygin and V.M.Alekseev.

6 REFERENCES

[1] V.G.Kurakin, A.V.Agafonov, A.I.Bukin, A.V.Koltsov, A.N.Lebedev. Far Infrared FEL Commissioning at Lebedev Physical Institute, in Proceedings of 7th European Particle Accelerator Conference, Vienna, 26 - 30 June 2000, p.p. 749-751.

[2] V.G.Kurakin, V.M.Alekseev, V.P.Busygin, A.V.Koltsov, P.V.Kurakin, G.G.Subbotin, E.I.Tamm, Yu.P.Yanusis. Automation of the Lebedev Physical Institute Synchrotron as the First Stage of the Accelerator Upgrade, this Proceedings.