

# ADJUSTABLE COMPUTER INTERFACE FOR MEASUREMENTS AND CONTROL

V.G.Kurakin, V.P.Busygin, A.V.Koltsov, Lebedev Physical Institute, Moscow, Russia

## Abstract

Novel programming tools such as Borland Delphi and C++ Builder allow to shorten considerably time for application development even with complicated graphical computer interface. We suggest making next step in rapid application development using Borland Visual Component Technology for adjusting user interface by user himself at the stage of program initialisation before real measurement. Appropriate computer code for real time diagnostic and control has been developed at Lebedev Physical Institute Radiation Complex. Three modes of physical data representation can be used. These are indicator, pulse recorder (oscilloscope mode) and spectrum representation. The last mode mentioned above can be used with animation process, thus allowing exploring dynamical properties of multi channel processes. Hard ware consists of fast digit to analogue converters and CAMAC interface. Detailed description of the program and computer code followed by example of measurement interface adjusting is given in paper.

## 1 INTRODUCTION

Nowadays, automation is the necessary attribute of any more or less complicated experiment. As a rule, the latter includes apparatus control as well as physical data collection and processing in real time scale. In the simplest case this means the visualisation the data being collected in various forms, but to have a possibility to affect experiment actively in real time scale, it is desirable sometimes to convert some data before visualisation. Up to recent time computer user interface that serves for measured data display and data entering was the most complicated and time consuming element of computer code. The appearance of Borland programming tools for rapid application development such as Borland Delphi and Borland C++ Builder had simplified significantly the program code developing process for interface components mentioned above. Nevertheless, code development for sophisticated measurement is still experts in C++ prerogative and such a situation very often imposes a limitation for fast experimental check of new ideas. To adjust program in use (executable file) to new experiment condition one have to make some changes in original code and to recompile it, and the process results very often in the errors that are very difficult for finding out. The are many reasons for this and almost every professional programmer was in similar situation. We

have made next step in Borland Visual Component Technology, having moved the drag-and-drop interface assembling to the level of executable file. All this is very similar of form filling in Borland Delphi or Borland Builder environment. The substantial difference is that the built interface does not needed in compiling and linking, while errors processing are including in program and cannot result in serious mistake during program functioning.

## 2 PROGRAM INTERFACE

After loading the program the menu bar appears (Fig. 1),

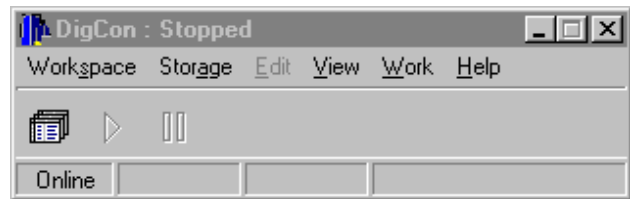


Fig. 1: Program starting menu

and user can start measurements with the interface being used in previous experiment run, or modify interface with environment, or built up completely new interface. We describe in details the last option. Any measurement deals with real devices as well as with data collection, and the first step assumes the choice of appropriate devices from the list or from tool bar. All these are collected on the top of Display Module, organised as standard window (Fig. 2). This and consequent steps of interface designing do not differ from appropriate procedures of interface

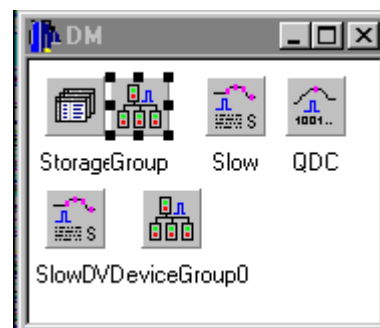


Fig. 2: Graphical menu for device choice

designing in Borland Delphi or C++ Builder environment. The next step is the necessary device property fixing in Property Editor. We use standard CAMAC interface, and together with many other attributes one has first of all to fix CAMAC crate station number, where the device is

installed and to activate device channel, that will be used for measurements (Fig. 3,4). After that step one can organise all equipment installed as different groups and individual

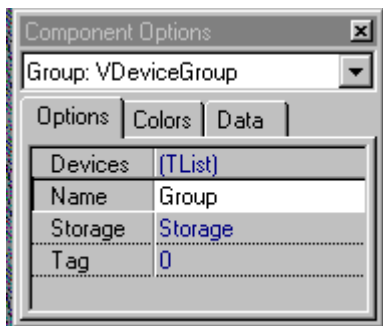


Fig. 3: Interface for selecting device and graphical components properties

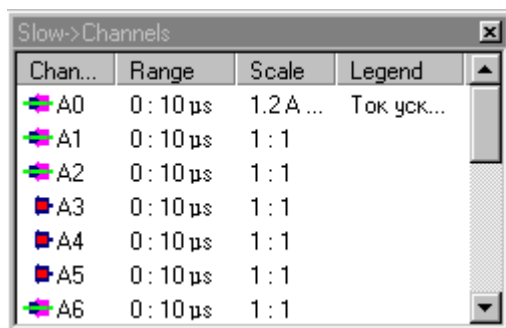


Fig. 4: Interface for unit channel selection

devices. The main difference between group and stay alone devices is the way of physical data collection and storage (saving for consequent use and processing). Grouped devices read out data from primary analogue devices synchronously with master device from the same group, and the collected and saved data marked appropriately. Data from individual device collected independently from group. Measurements are triggered by interrupt process or by timer.

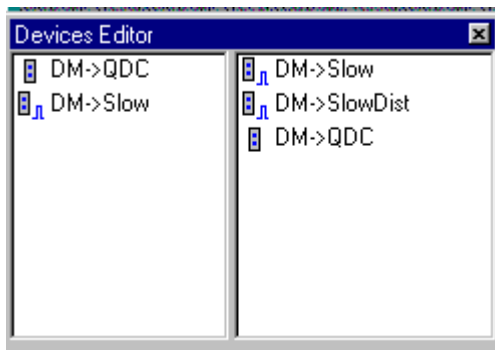


Fig. 5: Devices group creation

Fig. 5 demonstrates device editor. Icons in left columns indicates the devices that form group, and small pulse sign displayed as subscript of device icon indicates, that

appropriate device is master, that is used as trigger for other devices from the same group.

After the data collection item of the measurement interface has been completed one can proceed to data display interface designing. Several data display ways can be used. These are numeric indicator mode, oscilloscope mode, spectrum and analyser mode and spectrum animation mode. With the exception of the last one mentioned, these data representation on computer monitor are more or less evident. Indicator is simply the named physical value that can be placed in frame and accompanied by appropriate text (label). Oscilloscope representation serves to display one channel time dependent process, several such process can be displayed on the same grid. Spectrum as well as analyser mode are used to visualise multi channel processes. The difference between them is that in latter mode data may stored in individual channel, while in former case instant data from different channel displayed. We decided to use spectrum animation mode instead three-dimensional representation of time dependent multi channel process. For our opinion this is much more convenient. The data in this mode of operation can be displayed as usual spectrum at fix moment or as animation process. In latter case the animation process is triggered by additional event selection pulse, and user sees the time dependent process in slowed down way, that can be adjusted individually. The process can be played back manually as well. Of course, the rest-displayed data indicates the values corresponding to the moment chosen.

As usually in component designing technology used in visual programming, all components have to be placed on the top of form, the number of forms is not practically limited. Colour, axis labels and so on can be formatted in desired way.

We foresee a function converter in our interface to process data on line that is in real time scale. Since it is desirable very often to see not only measured data from primary sensors but to watch some secondary functional dependencies, user has to have the tools for such data transformation. At interface designing stage one enters appropriate dependencies in special windows, that are used for data transformation at the measurement stage.

### 3 PROGRAM CODE

Program code consists of several relatively independent parts described below:

- The first one is data driver that controls data acceptance from digital device. Of course, this part of code is device dependent and has to be written for any digital device (converter) used in experiment. Our strategy is to use limited number of different digital devices that are described below in appropriate subsection. This part of code includes dynamical library of .bpl type as well.

- System dynamical library forms the second main part. It includes the primary components that are parent for the real components used to form appropriate interface (data collection and displaying). These primary components define the standard functionality as well as appropriate interaction interface and a large number of auxiliary functions.
- Dynamical libraries, consisting of components, that are necessary to build up any desired interface. It is clear that the library variety define the flexibility of the application that is in what extent user is free to create the interface he wants. Of course, to add any new component in library, one has to recompile as well as to link again the main program module.
- The main program module, that is responsible for user interface. This part serves to adjust appropriate interface as described above.

The program code has been developed for Windows 95 platform.

#### 4 HARDWARE

We use hard ware in CAMAC standard, although digital devices might be complemented in ISA bus interface. The latter may be convenient if one uses limited hard ware and has not scientific laboratory environment. There are several digital devices to control our experimental installation. All equipment developed is used for multi purpose radiation complex control [1,2] The latter consists of far infrared free electron laser and high current racetrack. Detailed description of this facility one can find elsewhere in this Proceedings [3].

The most modern and base unit for physical data collection under adjusting now is multi channel wave form digitizer. One unit consists of 16 analogue-to-digit converters of type 1446ПВ1 (Russian production) and electronic memory; appropriate analogue and digital circuits integrated in unit for necessary functionality. Normal and fast modes can be used. In normal mode pulse processor converts input analogue signals into digital 10-bit code every 320 nsec and stores these in device memory. Microcircuits used allow to process analogue signals during 80  $\mu$ sec from any of 16 channels. This time interval is quite sufficient for our purposes. Analogue signals of both negative and positive polarities up to amplitude 2.5 volts can be digitised. In fast mode we have 4 channel that process analogue signal with the rate 100 nsec/per point. After digitising is completed, LAM-signal is installed in device, starting the following digital codes processing by computer. Two trigger input (the main and selection event) control the unit start and mark appropriate data for further selection. The device is designed as CAMAC unit 2M.

Multi channel charge to digit converter (QDC) in CAMAC standard, controlled before by its own software [2], is also included into interface under discussion. This unit is based on sampled/held with 16 independent

channels and multiplexer that consequently scans sample/held capacitance's and directs read out analogue signals to one analogue-to-digit converter of the type AD7892. Up to now we used it for linac and FEL data collection. Fig. 6 is a cut-off from computer monitor screen during linac tuning.

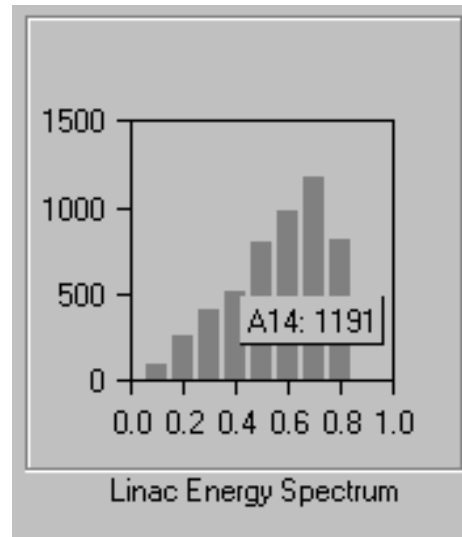


Fig 6. An example of real time measurements with interface under discussion. Label on the spectrum top appears after mouse pointer positioning on appropriate spectrum channel

#### ACKNOWLEDGEMENTS

The work is made in part in the frame of Russian Scientific Program "Microwave Physics" and the authors thanks Local Program Co-ordinators Prof. A.N.Lebedev and Dr. A.V.Agafonov for financial support. The authors are very grateful to our colleague V.M Alekseev for auxiliary code development for waveform digitizer adjustment and tuning.

#### REFERENCES

- [1] K.A.Belovintsev, A.I.Bukin, E.B.Gaskevich, A.V.Koltsov, V.A.Kuznetsov, V.G.Kurakin and S.V.Sidorov The Radiation Complex for Fundamental Research, in Proceedings of the 4th European Particle Accelerator Conference, London, 27 June to 1 July 1994, pp. 861-863.
- [2] A.V.Agafonov, A.I.Bukin, A.V.Koltsov, V.G.Kurakin, A.N.Lebedev. Current Status of Lebedev Physical Institute Far Infrared Free Electron Laser. Problem of Atomic Science and Technology, 4, 1999, p. 3-5.
- [3] V.G.Kurakin, A.V.Agafonov, A.I.Bukin, A.V.Koltsov, A.N.Lebedev. Far Infrared Free Electron Laser Commissioning at Lebedev Physical Institute, this Proceedings.