INSTRUMENTATION FOR COMPUTER BASED MONITORING AND CONTROL SYSTEM DEPLOYMENT

V.G.Kurakin, V,M.Alekseev, V.P.Busygin, A.V.Koltsov, Lebedev Physical Institute, Moscow, Russia
P.V.Kurakin, Keldysh Institute of Applied Mathematics, Moscow, Russia

Abstract

Visual programming system and appropriate hardware are described to deploy and use automation monitoring and control system.

INTRODUCTION

In this paper, we describe the instrumentation that allows to anybody to organize a measurement console on the basis of a single computer for an experiment as well to deploy rather sophisticated automated control system for experimental or industrial complex. The kernel of this instrumentation is program code (FlexUsI) that controls digital and analog devices constituting hard ware interface between physical sensors and computers. The key features of the code mentioned are as follows. First of all, it provides data acquisition as well control signals generation in real time scale; this is quit usual function of any program code for computer based measurements and control. Second, it allows constructing necessary user interface consisting of virtual devices that corresponds to real devises. This visual programming is realized at program run time that means that console is ready for use immediately after building. Third, this code posses the auto configuration property similar to plug play technique in modern computers. The other part of the instrumentation is the collection of various interface devices for data processing and control signal generation.

VISUAL PROGRAMMING SYSTEMG

In this item, we will demonstrate the deployment of computer based monitoring and control system on the example. Suppose that the complex under automation consists of rf linear pulse accelerator and experimental area, control rooms for accelerator itself as well as for experiment being assumed. There are cables connecting accelerator sensors and drivers with control room devises, the same is assumed also for experimental area.

One has to have a computer in control room (at least one) to observe data from accelerator sensors as well as to produce driving signals for equipment installed. Interface devices between cables termination and computer are necessary as well. We use CAMAC standard (this is not the best solution, but this standard is available for many scientific laboratories). It is supposed that there is a computer at experiment control room as well. For data exchange, connection between computers is necessary, and LAN (local area network) is used for this purposes. Additional computers may be used. For example, to reduce noise of any nature, it is highly desirable to convert analog signal to digital one close to rf accelerator then transmitting digital code to control room computer over LAN.

In order to synchronize data acquisition from different sensors, a single computer process has to control all digital interface equipment installed. This means that a single program code has to be used at any individual computer. This in turn results in definite functionality of the code if the same code has to be used for different configurations. In particular, it has to possess the property to acquire any desired equipment in measurement computer console as well as to configure it. Thus, we arrive at the solution that a program code has not only to control hardware but to configure a measurement console too. We have realized such software. Here we describe its functionality, while a way how such functionality has been achieved will be discussed in details in appropriate item.

Waveform digitizer is used to monitor pulse analog signals, and appropriate component from component repository has to be used to acquire digitized data. With drag and drop procedure it has to be placed on the top of data module – special window that contains all devices used. After that special table of properties becomes available, and operator can configure virtual device by specifying these properties with specific values.

Several components are used for data visualization. These are numeric indicator, oscilloscope, histogram and spectrum. We do not go in details of data visualization step because the configuration is similar just described, while functionality of data representation is nearly the same like in commercial math software. Animation is foreseen, that is very helpful for multi dimensional measurements.

Network components make it possible data exchange between individual consoles in real time scale over network thus allowing building up large automated systems. To transmit data acquired from local hardware to remote computer or to the whole network, transmitter component has to be installed and configured at local computer in a manner just described. Similar receiving component has to be installed on remote computer. The data transmitted over network could be manipulated in ordinary manner similar to those obtained from local hardware. We use UDP protocol. This protocol is the fastest one among TCP/IP family and works in real time scale. It does not guarantee data lost less transmission and one has to take care of reliable data delivering in any particular case.

Interpreter component makes it possible acquired data preprocessing in real time scale. Preprocessing code may be written in text editor or loaded from a file.

Just built up and configured, the console may be turned immediately to measurement mode. Detail one can find in reference [1].

HARDWARE

We have developed a basic set of devices that makes it possible to deploy computer based monitoring and control system [2]. In this item, we will enumerate them as well as give brief description.

1. Multi channel waveform digitizer is built up on the basis of AD9050. It digitizes input signal continuously that guarantee optimal thermal condition and stable operation as the result. Synchronizing pulse starts filling in built in memory with digit code, record process being stopped when the memory is filed completely. After reading out necessary information the device is unlocked thus being preparing for next cycle. All digitizers but one installed in the crate are slave synchronized with muster one. This guarantees complete synchronization of multi channel measurements thus allowing correlation studying. The operation frequency (four sub ranges 33 MHz/16 – 33MHz/1) and memory size (32 K words) provides observation of signals with pulse duration in the range dozens nanoseconds – 16 milliseconds. Device noise does not exceed 3 channels, the whole channels number being equal to 1024.
2. 8-channel Digit-to-Analog Converter (DAC) united with multiplexer 8-channel Analog-to-Digit Converter (ADC) in the same module is built up on the basis of microcircuits AD7808 and AD976. The unit is used to produce precise dc voltage to control wide class of electronic circuits as well as to measure precisely dc and pulse signals.
3. Arbitrary waveform synthesizer – multi purposes computer controlled generator, providing any desired waveform signal generation in wide range from microseconds to dozens milliseconds. This device has been developed manly to modulate rf voltage of rf accelerators and functions as follows. Necessary voltage dependence is designed in form editor and appropriate program fills in device memory. Incoming pulse starts scanning generator that reads out memory with the frequency 2 MHz and puts digit code to DAC input. Time domain dependence of the signal from DAC output is similar to that designed in form editor.
4. 8-channel timer uses the technique just described. All memory with the exception of one byte is filled with zeros, one bit only of the selected byte being equal to unit. This unit arises at the output of the appropriate channel in the process of memory scanning. Thus after start pulse coming pulses are generated in all 8 channels being delayed relative start pulse. It is quite clear that the device might be used to generate pulse trains with any desired pulse distribution within train.
5. Multi purposes pulse generator is built on the bases of timer microcircuits 82C54-2 and can be used to control step motor driver, to synchronize experimental complex equipment and so on. Each channel has several inputs and outputs to realize all functions foreseen. Several operation modes are possible, continuous pulse generation with programmable switch on/off function as well as mode of pulse train being among these.
6. 16-channel charge-to-digit converter complements the devices just described. It built up on the basis 16ZCP and possesses the following specifications: 11 bit amplitude resolution with gate width in the range 50 – 600 nanoseconds. Conversion time interval is equal to 90 microseconds.
7. High stability rf synthesizer is built on the basis AD9851. This device is realized on the idea of DDS (Direct Digital Synthesis, Analog Device 1996). Practically continuous frequency changing is available, the frequency stability being the same as reference quartz one. The frequency range of our synthesizer is 0-70 MHz.

All devises described above are equipped with microcircuit 24LC21. The information stored in this read/write memory is used for device identification as well as for console configuration check at program start and devices installed searching. This simplifies significantly the configuration procedure and guarantees errors avoiding as well, especially in the case when many devices installed at CAMAC crate.

As it has been mentioned, we use CAMAC interface with crate controller on ISA bus interface. To use modern computers that have not ISA bus intermediate device with USB bus is foreseen.

PROGRAM CODE

To achieve functionality mentioned above one has to have an access to methods and properties of the graphical interface objects at program run time stage. We had chosen Borland C++ Builder as programming environment for several reasons [3]. Besides the well known feature of rapid application development this programming tool possesses also the remarkable properties to move properties and methods accessibility to executable file stage. We want to remind that C++ Builder environment allows to create desired user interface using drag-and-drop method and then filling out appropriate forms in so called object inspector. Visual Component Library (VCL) is used as the base of building up measurement console components. It is worth to point to the fact that VCL components are real objects rather than abstract representation of classes used to facilitate and automate programmer routine work in writing program code. What is important for us, VCL supports the next functionality during run time. It allows enumerate all properties, which has been declared as __published. Also there are functions to read values of these properties and to change them as well. So called component persistence is supported. This means that component properties set can be saved to a stream (file or computer memory) and an object can be created on base of this description at any time. Abstract class TDesigner is declared in VCL library for visual editing support. In designing mode, all events sent to component, first can be handled by an object of the class inherited from TDesigner. It allows to handle user input and to prevent (in some cases) standard behavior of component as well.

Thus, besides majority of auxiliary tasks inherent to any program code developing two key problems have to be solved to achieve program functionality under discussion.

User Interface is not some collection of elements for measured data visualization et al only. It may be an interaction between these elements – there are not “bricks” only but “glue” too. In other words, there must be some “lows”, embedded in real components that behave the components interaction, and this is the second task to be solved in addition to the first one just described above.

Two root object classes form the basis that determines the bases of program kernel. This are VBaseDisplay and VBaseDataModule. The first one correspond to a form (object, that has the window, that may contain displayable elements)and is derived class with the parent one TCustomForm. The second one corresponds to data module and is derived from base class TDataModule. An object of this class has not the window and as a result may contain other not displayable objects only. To provide designer mode at run time phase class VDesigner had been created and Borland C++ Builder TFormDesigner is the parent class of our class.

As has already mentioned necessary components property may be accessible at run time phase due to VCL library functionality. Necessary function set is available in Borland C++ Builder environment in order to get as well to change component properties. User interface had been developed to edit component properties, and this is similar to object inspector in C++ Builder environment. This interface becomes visible after switching software to designer-editing mode.

We follow usual programming style when standard functionality inherent to real component is determined by parent class, while components are created from derived classes with additional specific properties and methods. Listed below are the base (parent) classes that determine “interactions lows” between real components in user interface, responsible for data acquisition, preprocessing, representation and storing:

- Class VDevice determines interface with objects VStorage and VDeviceGroup. All classes that are responsible for data acquisition and have any interface with objects VStorage and VDeviceGroup have to be derived from this class.
- Class VStorage determines interface with objects VDeviceGroup and VDevice. All classes that are responsible for acquired data storing have to be derived from this class.
- Class VDeviseGroup determines interface of objects VStorage with VDevice in the case, when all devices behave as the group. This means, that only one object in the group initiates a process of data acquisition and storing.
- Class VOutputDevice determines interface of data channels enumerating as well as data themselves. It is derived class from parent one VDevice.
- Class VDACControl represents control component. We include it in base class set in order to have possibility for control object to be notified about data acquisition starting and ending.
- Class VChannelView represents component for data displaying. Component of this class may be data source as well.
- Class VBaseDisplay is the container for the components which are responsible for data displaying and interacting with user.
- Class VBaseDataModule is container for non-displayed components. E.g. component for data reading from physical device is an object of this class.

Components of three last classes mentioned above are notified about data acquisition starting and ending. These are notified about new data incoming as well.

- Class SDataValue is used for data transmitting between components. It contains data themselves as well additional information concerning this data – data identifier, their type and so on.

Data acquisition and processing algorithm looks as follows. Appropriate component is notified (over interrupt initiated by the signal of device controller and subsequent message sending in windows environment) about new data incoming. The component reads data, assigns new unique identifier to this data portion and calls global function that informs all objects of the classes VBaseDisplay and VBaseDataModule of this event. If appropriate component has a connection to this data source it checks if these are new data and then processes them. Data source in turn may have an interface with other data source, and if this takes place it sends appropriate request to this object.

CONCLUSION

The instrumentation under discussion proved to be convenient and power tool. We plan to continue this work to bring this tool to commercial level.

REFERENCES

[1] V.G.Kurakin, A.V.Koltsov, P.V.Kurakin. Flexible User Interface for Computer Measurement and Control. In Proceedings of the 2001 Particle Accelerator Conference, Chicago, Illinois, USA, June18-22, 2001, pp.1192-1194.

[2] V.M.Alekseev, V.P.Busygin, A.V.Koltsov, P.V.Kurakin, V.G.Kurakin. Computer based monitoring and control system for RF accelerators. In Proceedings the of 18th Conference on Charge Particle Accelerators RUPAC-2002, Obninsk, October 1-4, 2002, pp. 552-557 (in Russian).

[3] V.G.Kurakin, A.V.Koltsov, P.V.Kurakin. FlexUsI – Intrface Builder for Computer Based Accelerator Monitoring and Contol System. In Proceedings of the 19th Conference on Charge Particle Accelerators RUPAC2004, October 4-8, Dubna, Russia 2004, pp. 347-349.