REFERENCES

1. Rey U. Methods of controlling technological processes. – Moscow: Mir, 1983. – 368 p.
2. Myshlyaev L.P., Avdeev V.P., Kartashov V.Ya., Kupchik M.B. Algorithmization of controlling objects with delay. – Kemerovo: KemSU, 1989. – 83 p.
3. Guretskiy Kh. Analysis and synthesis of control systems with delay. – Moscow: Mashinostroenie, 1974. – 328 p.
4. Kochubievskiy I.D. Loading systems for studying and testing devices and mechanisms. – Moscow: Mashinostroenie, 1985. – 224 p.
5. Streits V. State space method in theory of discrete linear control systems. – Moscow: Nauka, 1985. – 296 p.
6. Kartashov V.Ya. Equivalence of discrete – reality // Industrial ACS and controllers. – 2006. – № 8. – P. 40–44.
7. Makarov I.M., Menskiy B.M. Linear automatic systems. – Moscow: Mashinostroenie, 1982. – 504 p.
8. Volgin L.N. Optimal discrete control of dynamic systems. – Moscow: Nauka, 1986 – 240 p.

Received on 06.10.2007

UDC 681.51

AUTOMATED SYSTEM FOR STUDYING FEEDBACK REGULATORS

V.S. Leschev, A.A. Shilin, A.A. Svetlakov

Tomsk State University of Control Systems and Radio electronics
E-mail: npowest@tomsk.ru

Bundled software intended for realization of different control algorithms constructed on the basis of functional units on medium-priced industrial controllers have been considered. The example of programming in a language of functional block diagrams of algorithm of real processing automation is given.

Introduction

Recently to create automated control systems of technological processes (TP ACS) different technological programming languages available not only to programmer understanding but industrial engineers as well have been widely adopted. As a result, at present there are program packages for developing MMI and software of operator stations of TP ACS (SCADA) [1]. Owing to the fact that technological languages are easily unified a great many of such program packages is developed and moreover, proper programming languages are developed for them. To order somehow this process the standard of International Electrotechnical Commission IEC-1131-3 was accepted in 1993 [2]. The standard describes five programming languages of the programmed logic controller (PLC): *Sequential Function Chart* (*SFC*), *Function Block Diagram* (*FBD*), *Ladder Diagrams* (*LD*), *Structured Text* (*ST*), *Instruction List* (*IL*) [3]. Languages ST and IL are the most popular among programmers as they included the most common operators of programming languages of the type *Pascal* and *Assembler*.

Practice showed that the language of **functional blocdiagram (FBD)** is the clearest for industrial engineers. Language FBD serves for constructing and detailed description of control algorithms of technological processes. It allows a user to construct block diagram of control algorithm consisting of library blocks for a system of any complexity. Software complex «AKIAR» developed by programmers of the enterprise «NPO VEST» (Tomsk) in collaboration with information-measuring technology department at Tomsk State University of Control Systems and Radio electronics allows working with this very programming language and has a number of peculiarities discriminating it from similar software products. These peculiarities will be considered later and now let us pay attention to the fact that standard of IEC IEC-1131 is of voluntary character therefore we tried to conform but nevertheless developed our own library of functional blocks.

Software complex «AKIAR»

Software complex «AKIAR» includes all main possibilities of SCADA-systems. The analogue of such program is development system of programming algorithms «KONGRAF» («MZTA», Moscow) for regulators of «MS» type constructed at expensive microcontrollers. Software complex «AKIAR» is intended for working with regulators constructed at inexpensive microcontrollers. «AKIAR» combines graphics editor of functional blocks and program of modeling TP ACS which are intuitively clear for users having general idea of SCADA-systems. The significant advantage of program complex «AKIAR» over the system «KONGRAF» is the system of input, output and indication of industrial regulator parameters independent on control algorithm. Such system has flexible logic, allows denoting parameters in English and Russian, has tree-like (structured) menu.

Such powerful program packages of SCADA-systems as «*Genesis*», «*Trace Mode*», «*Genie*» are multifunctional and intended first of all for large-scale TP ACS. Such systems require powerful and expensive industrial controllers as well as significant temporary expenditures for modeling transfer processes for their realization. Use of expensive controllers in the field of heat and power engineering and especially housing and communal services for heating systems, hot-water supply and ventilation is not quite justified. Financial viability of using ACS is on the first place in such systems. Therefore, at present construction of such TP ACS at inexpensive controllers meeting minimal requirements of SCADA-system functionality is more actual. On the basis of software complex «AKIAR» the automatic regulator of «VEST» type was developed and realized. It is based on flexible logic and meets requirements of locality; functionality sufficient for concerned systems and efficiency the urgency of which has been already stressed.

Software complex «AKIAR» allows developing various algorithms of controlling technological processes in the language of functional block diagrams which is clear first of all for industrial engineers. TP ACS constructed on the basis of «AKIAR» and inexpensive controllers of «AKIAR» type meet all requirements of modern SCADA-systems and they are cost-beneficial in the field of power supply.

Structure of software complex «AKIAR»

Diagram of software complex «AKIAR» constructed on the basis of industrial regulator of «VEST»type is given in Fig. 1. Complex consists of graphics editor and software modeling.

Graphics editor is intended for plotting block-diagram of control algorithm on the basis of the library of software functional blocks by industrial engineer. The library includes several hundreds of functional blocks which have different functionalities: from simple mathematical (addition, subtraction) and logical operations (AND, OR) to PID-control (proportional-integral-differential control). User has no opportunity to develop new blocks. But the developers may develop new element for the library depending on complexity of the constructed system.

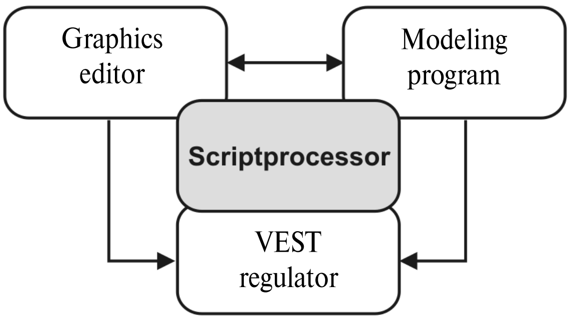

Fig. 1. *Diagram of the complex «AKIAR»*

Modeling program allows simulating the developed control algorithm. User may develop control object model or object simulator consisting of a proper set of digital inputs and analog outputs by the same graphics editor. The integral part of all SCADA-systems is the monitor of displaying the process variable behavior in the form of diagrams in real time mode [1]. When modeling control object the user has an opportunity to connect exterior modules, simulate complex functional nodes such as delay lines, noise sources, actuator hysteresis etc. As a result of modeling and obtaining video rendering about control object behavior in the form of diagrams the user may exclude all possible errors at design stage. It is especially useful at constructing large-scale block-diagrams of complex TP ACS where the error may be made through elementary lack of attention. And, finally, module of external connection with concrete equipment which realizes the possibility of exporting a ready-made block-diagram in the form of compiled control algorithm script into FLASH-memory of industrial controller microprocessor. Logical interface uses standard protocol *MODBUS* of *MODICON GOULD* company [5] therefore, software complex «AKIAR» may be used for displaying, archiving, testing and other operations of external device data base.

Language of functional block-diagram

Language of functional block-diagrams (FBD) is nothing other than transfer of ideas of relay logic language (LD) to another elemental base. Functional blocks, chips by appearance, are used instead of relay [3]. Algorithm of a certain device work expressed by this language means recalls functional diagram of electron device: elements of the type logical «AND», logical «OR» etc., connected with lines. FBD possess the advantage typical for languages transferring ideas of relay logic: easiness of initial studying. They give a rather natural opportunity of working with analog variables and minimal means of structuring (new functional blocks may be composed using the available ones) [4]. FBD language benefits significantly before the other technological languages in convenience of algorithm programming and structuring.

A functional block is the program object which fulfills a dedicated control function. Functional block output is connected to another block input by a communication line (variable). Several functional blocks connected by communication lines input/output form a program in FBD language. The language describes the law of program variable conversion. Graphic expression of FBD language in the form of blocks (rectangles) connected into block-diagram (program) by communication lines makes this programming language convenient for many application programs containing transfer of information or data between different components.

The majority of modern program packages meets the following programming rules at FBD:

1) functional blocks may be arranged randomly in program field;
2) there can not be free (disconnected) inputs and outputs of functional blocks;
3) any connection may have name and type;
4) inputs and outputs of functional blocks connected to links having the same names are considered to be connected;
5) order of block performance in the program: left to right, top-down;
6) in the case if one and the same variable is formed by several functional blocks the last value will be assigned to the variable by that block which was performed last of all.

Graphics editor «FBD-editor» and program «Modeling»

At present the complete library of functional blocks of software complex «AKIAR» numbers 43 elements (blocks). The symbols and performed functions of the main functional blocks which found application in various control algorithms are given in Table 1.

Table 1. *Library of functional blocks of «AKIAR»*

Symbol	Function
Ind	Block of specifying parameter displayed at regulator indicator (without possibility of adjustment)
GetPar	Block of specifying parameter displayed at regulator indicator (with possibility of adjustment)
SetCnst	Block of specifying constant value (real number)
LoWord	Formation of double-byte number (SHORT) at block output from four-byte number (LONG) at block input
HiWord	Function reverse to «LoWord»
Integr	Integrator
Inert	Block of forming relaxation circuit with transfer function $W(p)=1/(1+\tau p)$
Isodr	Block of forming isodromic link with transfer function $W(p)=\tau p/(1+\tau p)$
USWOd	Regulator forming block
Arch	Block of forming variable value archive
Signal forming at output (input) of the block	
SetDo	Digital output signal
SetAo	Analog output signal
GetDi	Digital input signal
GetAi	Analog input signal
Formation of real value of current time at block output by the formula Out=In/10	
Hour	Hour
Mins	Minutes
DayM	Day of the month
DayW	Day of the week
Year	Year
Blocks of performing mathematical and logical operations	
Add (A+B)	Addition
Mul (A×B)	Multiplication
Sub (A−B)	Subtraction
Div (A/B)	Division
More (A>B)	More
Equ (A=B)	Equal
Less (A<B)	Less
And (A&B)	Conjunction / «AND»
Or (A|B)	Disjunction / «OR»
Xor (A/\B)	Excluding «OR»
No (A→$\bar{A}$)	Inversion / «NO»
Sl_AB	Choice of «A» provided performance of «U» (TRUE)

Blocks performing dedicated functions, for example, for working with memory («*FLSH*» – *Flash-memory*; «*EEPR*» – volatile memory, «*TmMem*» – nonvolatile memory of time) or for performing operations with bit variables at «low» level («*LShft*» – formation of four-byte numbers and back «*RShft*») are given in Table 1.

The example of control algorithm realization

Let us consider functioning of graphics editor «FBD-editor» and program «Modeling» by the simplest practical example. Let us try to develop the algorithm of automation of hot water supply (HWS) system of a house.

The diagram of HWS automation is given in Fig. 2. The flow of delivery water enters HWS heat exchanger from delivery heating system pipeline. Cold water enters heat exchanger from water pipeline as a counter flow. Heat is transferred from warmer medium to a colder one through steel plates of heat exchanger water is heated up to the specified value (hot water) and come to consumers. Return water returns to the heat network at the output from heat exchanger. According to the building regulations 2.04.01-85 [6] hot water temperature should be equal to 65 °C. As a result, the main task of HWS automated control is supporting the specified temperature value of hot water at changing the temperature of delivery water supplied into heat exchanger.

A fragment of a window of graphics editor «FBD-editor» with block-diagram of algorithm of controlling HWS temperature according to the specified value is given in Fig. 3. Return water temperature restriction at the output from heat exchanger according to temperature heat supply graph is not considered in the given example.

Control action is formed by one input parameter (HWS temperature «HWS temp.») in the given block-diagram and it is enough for understanding principles of programming at FBD.

As it is seen from block-diagram (Fig. 3) functional blocks are arranged randomly in window field there are no vacant inputs and outputs; the diagram is formed left to right top-down for correct order of performance.

Graphics editor supports two types of variables: binary variables (*Boolean*) 1 (*true*) or 0 (*false*); integer type variable (*Integer*) in the range from -32768 to 32767. There are the following variables in the given diagram. The specified parameter «Зад.Тгвс» is the specified temperature of HWS; input variable «Темп.гвс» is the measured temperature of HWS; instruments settings of adjustment: «Вр.демп» is the damping time (integral time of measured parameters), «Время об» is the object time (object time constant), «Вр.возд» is the action time (in direct proportion to gain constant in closed circuit), «Зона неч» is the dead space, «Гистерез» is the hysteresis range.

Inlay – «Parameter properties» is given in Fig. 4. A name («Value») of parameter, parameter type, possibility of manual control, units, amount of sampling increments, minimal, maximal and specified value of the parameter are specified in the given delay. The possibility of user access restriction to parameter change («Access level») by means of password, service mode or service button is provided.

The control circuit operates in the following way (Fig. 3). The measured value of HWS temperature is subtracted from specified value. The error value obtained at subtraction block output (*Sub*) is displayed in indicator window (*Ind*), time averaged (*Inert*) and enter regulator input (*USWOd*). Depending on specified adjustment parameters and exceeding dead space and hysteresis at regulator block output the control signal which is compared to constant null (*SetCnst*) and supplied to corresponding digital output (*SetDo*), that is control valve, depending on error signal (*Less*), (*More*) is formed.

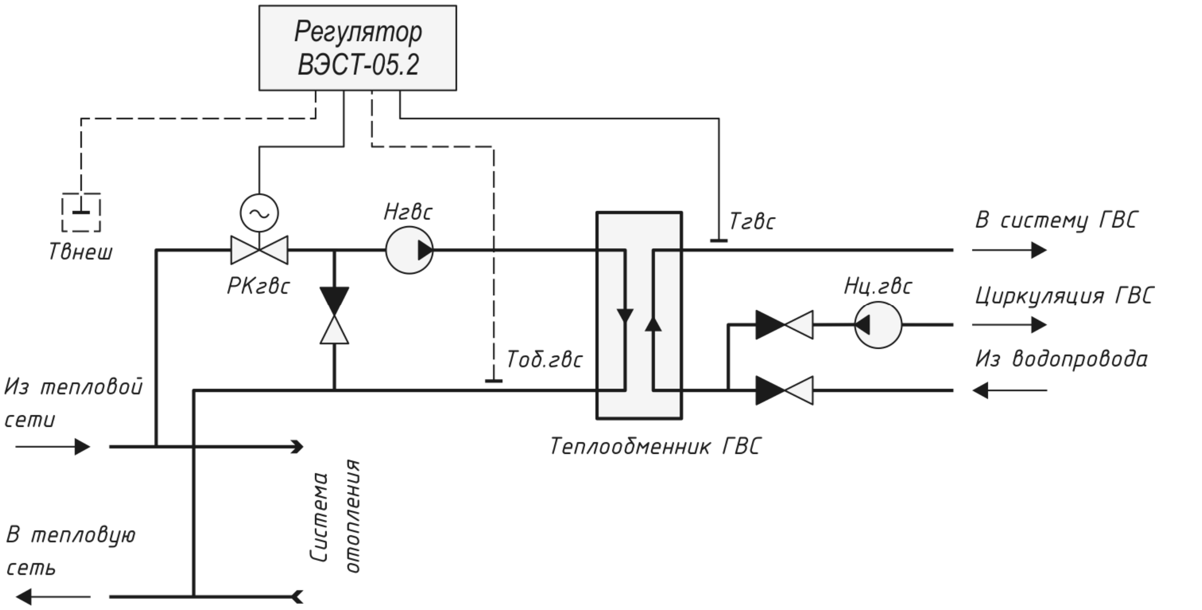

Fig. 2. *Diagram of connecting controlling complex of HWS system of a house.*
Регулятор ВЭСТ – VEST regulator; В систему ГВС – Into HWS system; Циркуляция ГВС – HWS circulation; Из водопровода – From water pipeline; Из тепловой сети – From heat network; Теплообменник ГВС – HWS heat exchanger; В тепловую сеть – Into heat network; Система отопления – Heating system

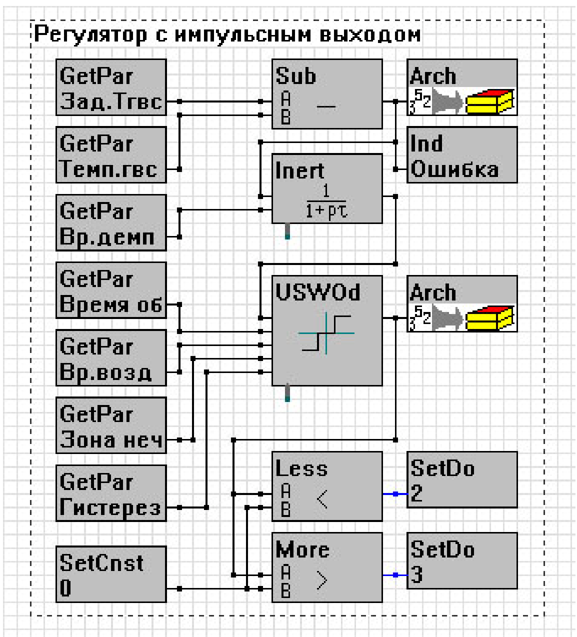

Fig. 3. *Block-diagram of algorithm of controlling HWS temperature*

Decreasing HWS temperature the regulator opens control valve increasing delivery water discharge to heat exchanger. Increasing HWS temperature the regulator closes control valve decreasing delivery water discharge to heat exchanger. The valve is opened or closed till HWS temperature is adjusted with specified value. Temperature is set with accuracy in the range of specified dead space usually not more than ±0,5 °C.

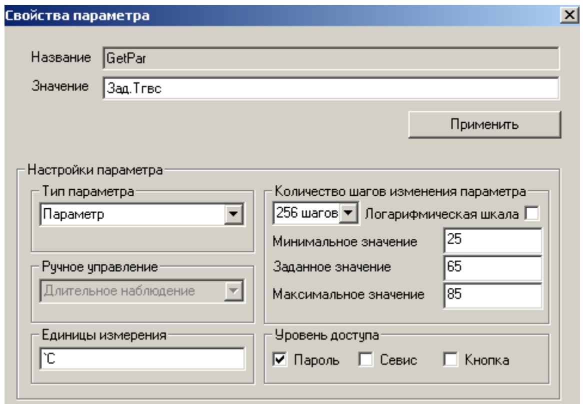

Fig. 4. *Inlay «Parameter properties»*

To check out availability of the given control circuit let us develop the control object (CO) model that is control valve with relaxation circuit (Fig. 5). There are two new variables in this circuit. This is «Вр.хода» is the run time (time of valve motion from completely open to completely close state or on the contrary) and «Время об» is the object time which characterizes control object inertia. Circuit operation is rather clear and there is no point in studying it in detail taking into account all that we have already analyzed.

When both block-diagrams are formed it is necessary to describe connections between regulator and CO in dedicated window «Connections» (Fig. 6). As it is seen from the Figure there are parameters of connection adjustment (left to right) in the given window: transferring device type («Simulator» or «External device»); transferring device number; number of transferring device output; transmission type (discrete or analogous); receiving device type; receiving device number; number of receiving device output; connection serial number.

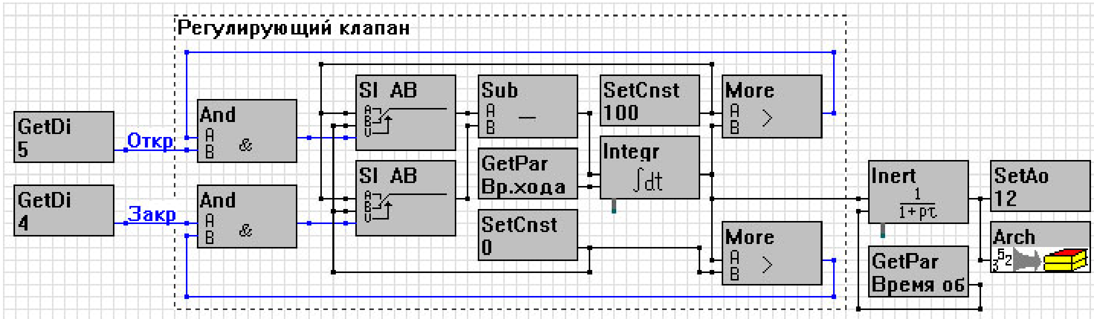

Fig. 5. *Block-diagram of control object model*

Тип прибора	№ прибор	№ выхода	Тип сигнала	Тип прибора	№ прибор	№ входа	Соединение
Имитатор	0	2	Дискрет.	Имитатор	1	4	0
Имитатор	0	3	Дискрет.	Имитатор	1	5	1
Имитатор	1	12	Аналог.	Имитатор	0	2	2

Fig. 6. *Window «Connections»*
Тип прибора – Device type; № прибора – device №; № выхода – output №; Тип сигнала – Signal type; Соединение – Connection; Имитатор – Simulator; Дискретный – Discrete; Аналоговый – Analogous

Then it is necessary to compile the developed program and start the window of regulator prototype (Fig. 7).

As a result user has the opportunity to observe and work out transfer processes in the form of diagrams both in the form of aggregate result and in real time mode changing the specified HWS temperature and parameters of regulator adjustment.

The aggregate result of modeling in the relative program «Modeling» is given in Fig. 8. The amount of taken off data (curves) corresponds to the quantity of archive

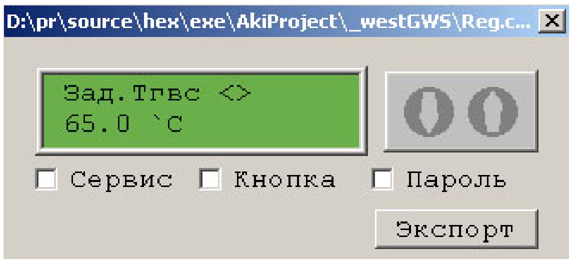

Fig. 7. *Window of regulator prototype*

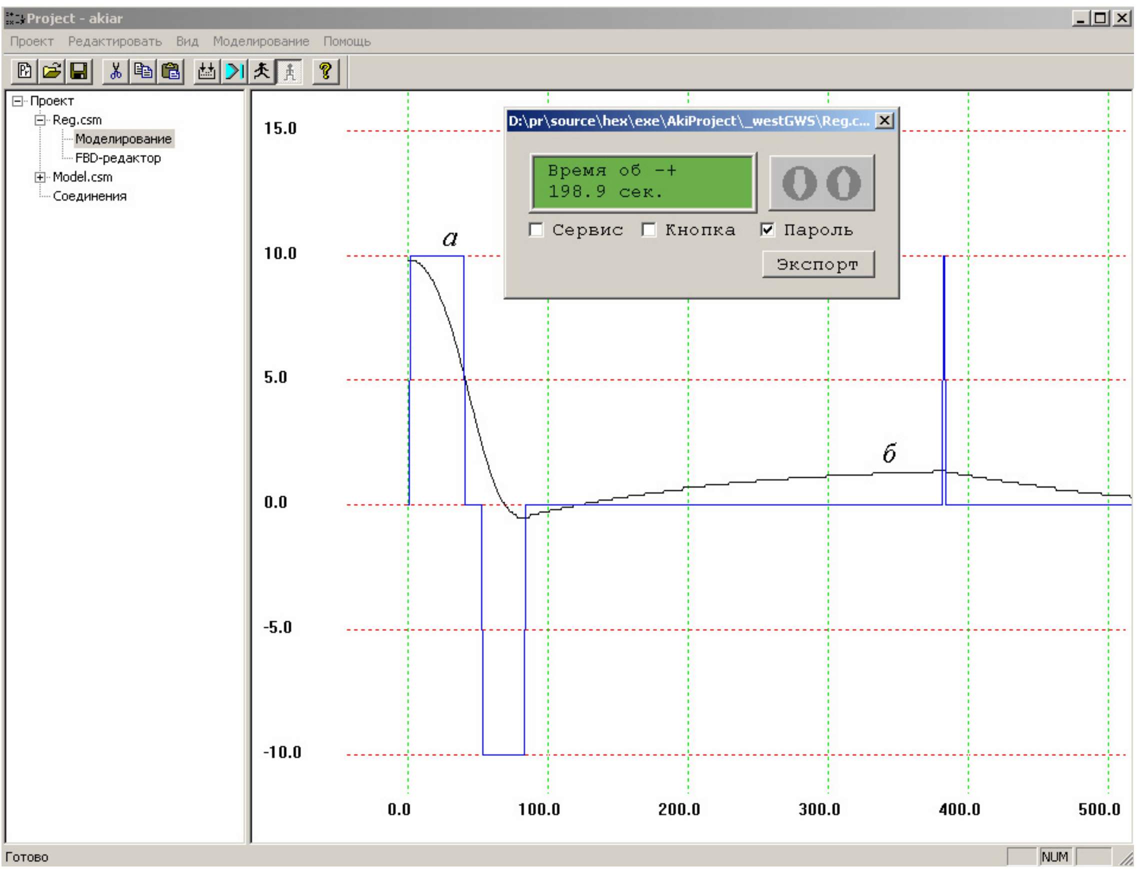

Fig. 8. *Window of modeling program*

blocks (*Arch*) set in the program at proper points (Fig. 3). User has the opportunity to set at his own discretion points of archive data taking off and keeping at block-diagram in graphics editor (up to 6 parameters).

In the given diagram curve *a* is the duration and sign (greater, less) of control pulses at regulator block output (*USWOd*), curve *б* is the error of measuring at output of block (*Sub*). CO characteristics and parameters of regulator adjustment for the given object are given in Table 2.

Table 2. *CO characteristics and parameters of regulator adjustment*

CO characteristics	Run time = 101,9 c Object time = 147,7 c
Parameters of regulator adjustment	Damping time = 10 c Object time = 198,9 c Action time = 65,9 c Dead space = 0,9 °C Hysteresis = 0,2 °C

As it is seen from the diagram in Fig. 8 the transient achieves stability at perturbation action, for example, at HWS temperature change by 10 °C. Changing such parameters of regulator adjustment as «Время об» and «Вр.возд» user has the opportunity to explore and achieve more optimal results of adjustment.

Conclusion

The software complex of studying automatic regulators «AKIAR» was presented. Its structure, features as well as advantages and a number of peculiarities distinguishing it from similar products were studied. The language of functional block-diagrams (FBD) and library of functional blocks of «AKIAR» were briefly described.

The example of programming the simplest algorithm of controlling real technological process in FBD language was given in conclusion. By the given example of HWS system automation the idea of complex «AKIAR» operation in general; block-diagrams construction at functional blocks, transient modeling and obtaining optimal results of adjustment in particular was obtained. The software complex «AKIAR» has capacities for improving developing algorithms of control and automation of technological processes of various complexity levels.

REFERENCES

1. Kuznetsov A. SCADA-systems: you can not be a programmer ... // STA. – 1996. – № 1. – P. 32–35.
2. IEC 65B/373/CD, Voting Draft – IEC 61131-3. Programmable controllers. Part 3: Programming languages, 2nd Ed – International Electrotechnic Commission, 1993.
3. Zyubin V.E. To fifth anniversary of the standard IEC 1131-1. Results and forecasts http://www.msclub.ce.cctpu.edu.ru/plc/iec1131.htm
4. Zyubin V.E. PLC programming: IEC languages 61131-3 and possible alternatives // Promyshlennye ACS i Kontrolery. – 2005. – № 11. – P. 31–35.
5. MODBUS protocol. Communication protocols of device «Vzlet» company. User Guide – Saint-Petersburg: «Vzlet». – 2000. – P. 39–52.
6. Building regulations 2.04.01-85. Inner water pipeline and sewage system.

Received on 09.10.2007