

The module <FLibComplex1> of the subsystem “Specials”

<i>Module:</i>	FLibComplex1
<i>Name:</i>	Library of functions compatible with SCADA Complex1.
<i>Tun:</i>	Specials
<i>Source:</i>	spec_FLibComplex1.so
<i>Version:</i>	1.1.0
<i>Author:</i>	Roman Savochenko
<i>Translated:</i>	Maxim Lysenko
<i>Description:</i>	Provides the library of functions compatible with SCADA Complex1 of the firm SIC “DIYA”.
<i>License:</i>	GPL

Contents table

The module <FLibComplex1> of the subsystem “Specials”	1
Introduction	2
1. Alarm (alarm) <111>	2
2. Condition '<' (cond_lt) <239>	2
3. Condition '>' (cond_gt) <240>	2
4. Full condition (cond_full) <513>	2
5. Digital block (digitBlock) <252>	3
6. Division (div) <526>	3
7. Exponent (exp) <476>	3
8. Flow (flow) <235>	3
9. Iterator (increment) <181>	3
10. Lag (lag) <121>	3
11. Simple multiplication(mult) <259>	4
12. Multiplication + Division(multDiv) <468>	4
13. PID regulator (pid) <745>	4
14. Power (pow) <564>	5
15. Selection (select) <156>	5
16. Simple integrator (sum) <404>	5
17. Sum with the division (sum_div) <518>	5
18. Sum with the multiplication. (sum_mult) <483>	6
19. User programming API	6

Introduction

Special module FLibComplex1 provides the OpenSCADA system with the static library of functions compatible with SCADA Complex1 of firm SIC "DIYA". These functions are used in the SCADA system Complex1 in the form of algoblocks to create inner-computings on the virtual controller. Provision of the library of these functions lets to do the transfer of computational algorithms from the system Complex1.

To address the functions of the library you can use static call address "**Special.FLibComplex1.{Func}()**" or dynamic "**SYS.Special.FLibComplex1["{Func}"].call()**", "**SYS.Special.FLibComplex1.{Func}()**". Where *{Func}* — function identifier in the library.

Below is the description of each function of the library. For each function it was evaluated the execution time. Measurements were made on the system with the following parameters: Athlon 64 3000 + (ALTLinux 4.0 (32bit)) by measuring the total execution time of the function when you call it 1000 times. Sampling was carried out of the five calculations, rounded to integer. Time is in angle brackets and is measured in microseconds.

1. Alarm (alarm) <111>

Description: Set alarm sign in the case of going out of the variable for the specified boundary.

Formula:

```
out = if(val>max || val<min) true; else false;
```

2. Condition '<' (cond_lt) <239>

Description: Operation of branching in accordance with the condition "<".

Formula:

```
out = if(in1<(in2_1*in2_2*in2_3*in2_4)) in3_1*in3_2*in3_3*in3_4;  
      else in4_1*in4_2*in4_3*in4_4;
```

3. Condition '>' (cond_gt) <240>

Description: Operation of branching in accordance with the condition ">".

Formula:

```
out = if(in1>(in2_1*in2_2*in2_3*in2_4)) in3_1*in3_2*in3_3*in3_4;  
      else in4_1*in4_2*in4_3*in4_4;
```

4. Full condition (cond_full) <513>

Description: Full check of the conditions, including more, less and equal.

Formula:

```
out = if(in1<(in2_1*in2_2*in2_3*in2_4)) in3_1*in3_2*in3_3*in3_4;  
      else if( in1>(in4_1*in4_2*in4_3*in4_4) in5_1*in5_2*in5_3*in5_4;  
      else in6_1*in6_2*in6_3*in6_4;
```

5. Digital block (digitBlock) <252>

Description: Function contains the control algorithm of digital signals acquisition for valves and pumps that contain: signs of “Open”, “Close” and the command “Open”, “Close”, “Stop”. Supports work with pulse commands, i.e. can read the signal through the specified period of time.

Parameters:

ID	Parameter	Type	Mode
cmdOpen	Command “Open”	Bool	Out
cmdClose	Command “Close”	Bool	Out
cmdStop	Command “Stop”	Bool	Out
stOpen	Position “Opened”	Bool	In
stClose	Position “Closed”	Bool	In
tCmd	Command hold time (s)	Integer	In
frq	Frequency of calculation (milliseconds)	Integer	In

6. Division (div) <526>

Description: Makes division of the set of variables.

Formula:

$$\text{out} = \frac{(\text{in1_1} * \text{in1_2} * \text{in1_3} * \text{in1_4} * \text{in1_5} + \text{in2_1} * \text{in2_2} * \text{in2_3} * \text{in2_4} * \text{in2_5} + \text{in3})}{(\text{in4_1} * \text{in4_2} * \text{in4_3} * \text{in4_4} * \text{in4_5} + \text{in5_1} * \text{in5_2} * \text{in5_3} * \text{in5_4} * \text{in5_5} + \text{in6});}$$

7. Exponent (exp) <476>

Description: Calculating the exponent under the group of variables.

Formula:

$$\text{out} = \exp \left(\frac{(\text{in1_1} * \text{in1_2} * \text{in1_3} * \text{in1_4} * \text{in1_5} + (\text{in2_1} * \text{in2_2} * \text{in2_3} * \text{in2_4} * \text{in2_5} + \text{in3})}{(\text{in4_1} * \text{in4_2} * \text{in4_3} * \text{in4_4} * \text{in4_5} + \text{in5})} \right)$$

8. Flow (flow) <235>

Description: Calculation of the gas flow.

Formula:

$$f = K1 * ((K3 + K4 * x) ^ K2);$$

9. Iterator (increment) <181>

Description: Iterative calculation with the increment specifying. Gain ratio for different directions is different.

Formula:

$$\text{out} = \text{if}(\text{in1} > \text{in2}) \text{in2} + \text{in3} * (\text{in1} - \text{in2}); \\ \text{else } \text{in2} - \text{in4} * (\text{in2} - \text{in1});$$

10. Lag (lag) <121>

Description: Variation of the variable lag. Practice, this is the filter without reference to time.

Formula:

$$y = y - K_{\text{lag}} * (y - x);$$

11. Simple multiplication(mult) <259>

Description: Simple multiplication with division.

Formula:

$$\text{out} = (\text{in1_1} * \text{in1_2} * \text{in1_3} * \text{in1_4} * \text{in1_5} * \text{in1_6}) / (\text{in2_1} * \text{in2_2} * \text{in2_3} * \text{in2_4});$$

12. Multiplication + Division(multDiv) <468>

Description: Branched multiplication + division.

Formula:

$$\text{out} = \text{in1_1} * \text{in1_2} * \text{in1_3} * \text{in1_4} * \text{in1_5} * (\text{in2_1} * \text{in2_2} * \text{in2_3} * \text{in2_4} * \text{in2_5} + (\text{in3_1} * \text{in3_2} * \text{in3_3} * \text{in3_4} * \text{in3_5}) / (\text{in4_1} * \text{in4_2} * \text{in4_3} * \text{in4_4} * \text{in4_5}));$$

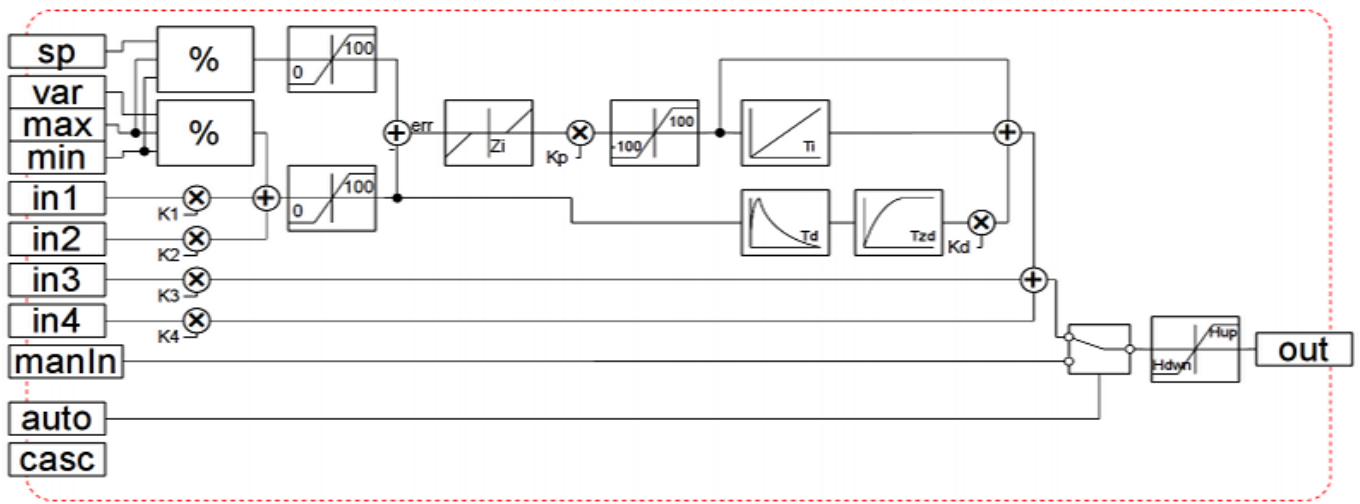
13. PID regulator (pid) <745>

Description: Proportional-integral-differential regulator.

Parameters:

ID	Parameter	Type	Mode	By defaults
var	Variable	Real	In	0
sp	Set point	Real	Out	0
max	Maximum of scale	Real	In	100
min	Minimum of scale	Real	In	0
manIn	Manual input (%)	Real	In	0
out	Out (%)	Real	Return	0
auto	Auto	Bool	In	0
cas	Cascade	Bool	In	0
Kp	Kp	Real	In	1
Ti	Ti (ms)	Integer	In	1000
Kd	Kd	Real	In	1
Td	Td (ms)	Integer	In	0
Tzd	Td lag (ms)	Integer	In	0
Hup	Upper limit of the out (%)	Real	In	100
Hdwn	Lower limit of the out (%)	Real	In	0
Zi	Insensitivity (%)	Real	In	1
followSp	Follow sp from var on manual	Bool	In	1
K1	Koef. of the input 1	Real	In	0
in1	Input 1	Real	In	0
K2	Koef. of the input 2	Real	In	0
in2	Input 2	Real	In	0
K3	Koef. of the input 3	Real	In	0
in3	Input 3	Real	In	0
K4	Koef. of the input 4	Real	In	0
in4	Input 4	Real	In	0
f_frq	Frequency of calculation (Hz)	Real	In	1

Structure:



14. Power (pow) <564>

Description: Raising to the power

Formula:

```
out = (in1_1*in1_2*in1_3*in1_4*in1_5) ^
      (in2_1*in2_2*in2_3*in2_4*in2_5 +
       (in3_1*in3_2*in3_3*in3_4*in3_5) /
       (in4_1*in4_2*in4_3*in4_4*in4_5));
```

15. Selection (select) <156>

Description: Selection of the one from four options.

Formula:

```
out = if( sel = 1 ) in1_1*in1_2*in1_3*in1_4;
      if( sel = 2 ) in2_1*in2_2*in2_3*in2_4;
      if( sel = 3 ) in3_1*in3_2*in3_3*in3_4;
      if( sel = 4 ) in4_1*in4_2*in4_3*in4_4;
```

16. Simple integrator (sum) <404>

Description: A simple summation with the multiplication.

Formula:

```
out = in1_1*in1_2+in2_1*in2_2 + in3_1*in3_2+in4_1*in4_2 +
      in5_1*in5_2+in6_1*in6_2 + in7_1*in7_2+in8_1*in8_2;
```

17. Sum with the division (sum_div) <518>

Description: The summation the set of values with the division.

Formula:

```
out = in1_1*in1_2*(in1_3+in1_4/in1_5) +
      in2_1*in2_2*(in2_3+in2_4/in2_5) +
      in3_1*in3_2*(in3_3+in3_4/in3_5) +
      in4_1*in4_2*(in4_3+in4_4/in4_5);
```

18. Sum with the multiplication. (sum_mult) <483>

Description: The summation the set of values with the multiplication.

Formula:

$$\begin{aligned} \text{out} = & \text{in1_1} * \text{in1_2} * (\text{in1_3} * \text{in1_4} + \text{in1_5}) + \\ & \text{in2_1} * \text{in2_2} * (\text{in2_3} * \text{in2_4} + \text{in2_5}) + \\ & \text{in3_1} * \text{in3_2} * (\text{in3_3} * \text{in3_4} + \text{in3_5}) + \\ & \text{in4_1} * \text{in4_2} * (\text{in4_3} * \text{in4_4} + \text{in4_5}); \end{aligned}$$

19. User programming API

Some objects of the module provides functions for user's programming.

The object "Functions library" (SYS.Special.FLibComplex1)

- *ElTp {funcID}(ElTp prm1, ...)* — call the library function *{funcID}*. Return result of the called function.

The object "User function" (SYS.Special.FLibComplex1["funcID"])

- *ElTp call(ElTp prm1, ...)* — call the function with parameters *<prm{N}>*. Return result of the called function.