Getting started

Creating a simple IOlog SS3000 modules Modbus Rtu application

Document revision

Date	Edition	Comments
23/07/2010	1.0	-

Sielco Sistemi srl
via Roma, 24
I-22070 Guanzate (CO)

http://www.sielcosistemi.com

1. Introduction

In this guide we propose, as an example, the design of a simple supervision application that communicate with IOlog SS3000 data acquisition modules using Modbus RTU protocol; this example is a little step towards the design of more complex SCADA applications, but it can be useful for anybody who approaches for the first time to a SCADA, and in particular to **Winlog Lite/ Pro** software, to quickly understand how to communicate with theese devices.

Sielco Sistemi provides a series of flexible and easy to use data acquisition modules; available modules are:

SS 3014 4 RTD, Res, Pot Analog Inputs

SS 3015 4 V or mA Analog Inputs

SS 3016 4 Tc, mV, mA Analog Inputs

SS 3017 8 V or mA Analog Inputs

SS 3018 8 Tc, mV, mA Analog Inputs

SS 3024 4 0..10V - 0..20mA analog Outputs

SS 3130 4 Digital Inputs + 4 Relays

SS 3148 12 Digital Inputs

SS 3188 8 Digital Inputs + 8 Digital Ouputs

SS 3580 RS232 / RS485-422 Converter

SS 3580-USB USB / RS485-422 Converter

SS 3580-TCP Ethernet Modbus TCP / RS485 Modbus RTU Converter

SS 3590 RS485-422 Repeater

When you design a new application, it is necessary to know for each external device: the communication protocol, the address and the list of variables that you want read or write.

Let us suppose to create an application that communicates through a Modbus RTU protocol with the 2 following devices:

SS 3018 8 Thermocouple Inputs (address 5)

SS 3188 8 Digital Inputs + 8 Digital Outputs (address 9)

We have to read Input1,..,Input 8 numeric variables and to activate Out1,…,Out8 digital variables (Control) of these two devices.

Our aim is to realise a simple 8 loops temperature controller; for each loop, when temperature is under a low threshold, a control output becomes active (ON/OFF).

IOlog SS3000 acquisition modules are released with the following configuration:

Protocol: MODBUS RTU - **Baud Rate**: 38400 - **Address** : 1

To modify the configuration, please refer *"Configuring IOlog SS3000 modules using Winlog*" guide.

2. Creating the project

To create a new supervision project, it is necessary to use **Project Manager**, the **Winlog Lite/Pro** integrated development environment that provides different tools (**Gate Builder**, **Template Builder**, **Code Builder, Application Builder**).

Run **Project Manager** selecting own icon from Start menu

It is possible to create a new project in the standard way (see the guide *"Getting started - Creating a simple Modbus protocol application*") or using SS3000 instruments **Winlog Lite/Pro** internal library.

Winlog Lite/Pro can automatically create SCADA applications by using **Application Builder**, a powerful tool that allows a drastic reduction of development time. Any application can be built by simply taking up from a library and putting together objects that refer to various automation devices (PID controllers, indicators, data acquisition modules, motor drives, ...).

Objects include tags, protocols, templates and all information required for device supervision and programming (eg. device front view, configuration templates, ...). Events and alarms that can be managed by **Winlog Lite/Pro** are also included.

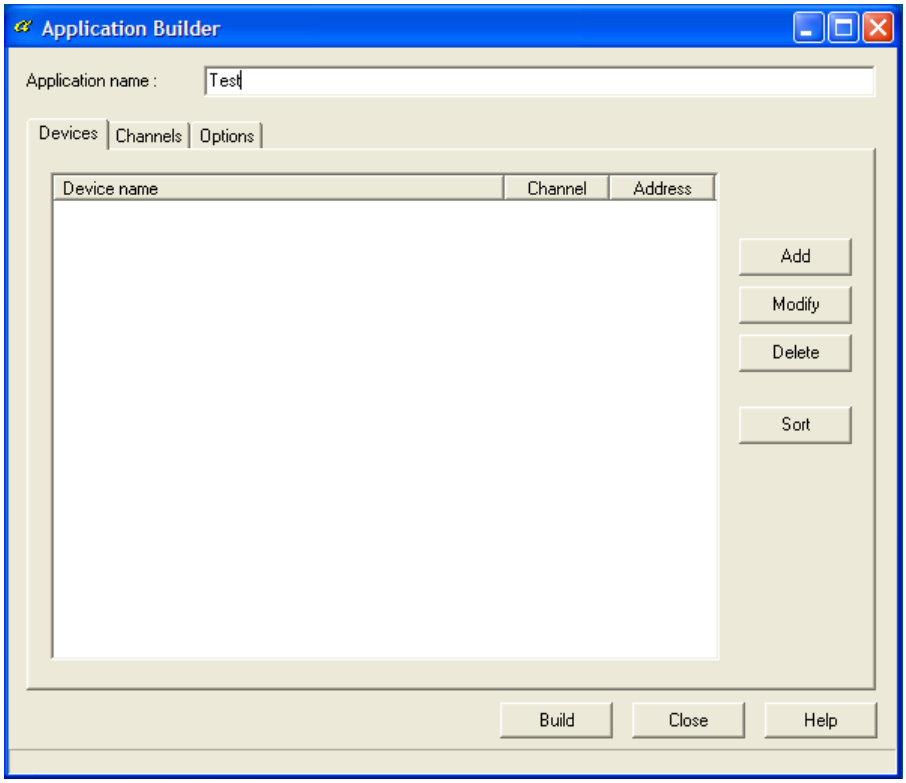
Winlog Lite/Pro includes a device library with IOlog SS3000 data acquisition modules.

Select `Application Builder` from `Tools` menu and insert the project name (foe example `Test`).

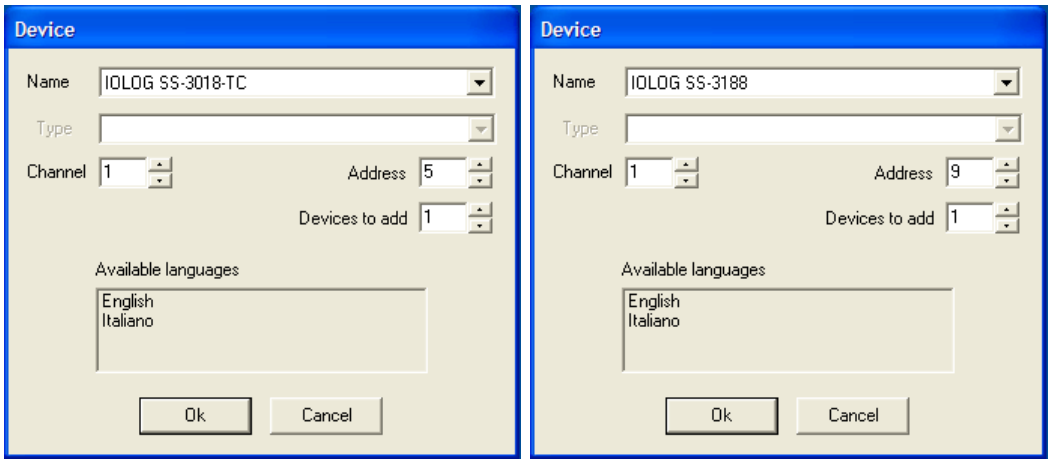
Add devices to the project picking them from the available ones clicking on `Add button`

Add devices on the channel 1 using different addresses:

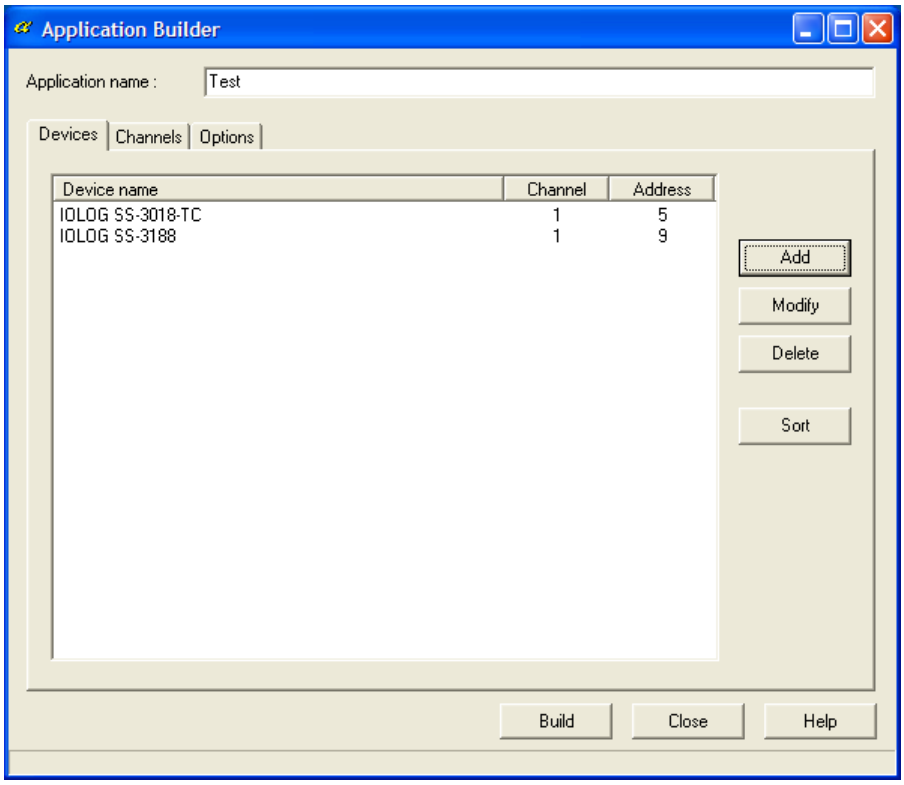
Device Name	Channel	Description	Address	Device to Add
IOlog SS3018	1	8 Thermocouple Inputs	5	1
IOlog SS3188	1	8 Digital Inputs e 8 Digital Outputs	9	1



Project name definition



Adding devices in the project



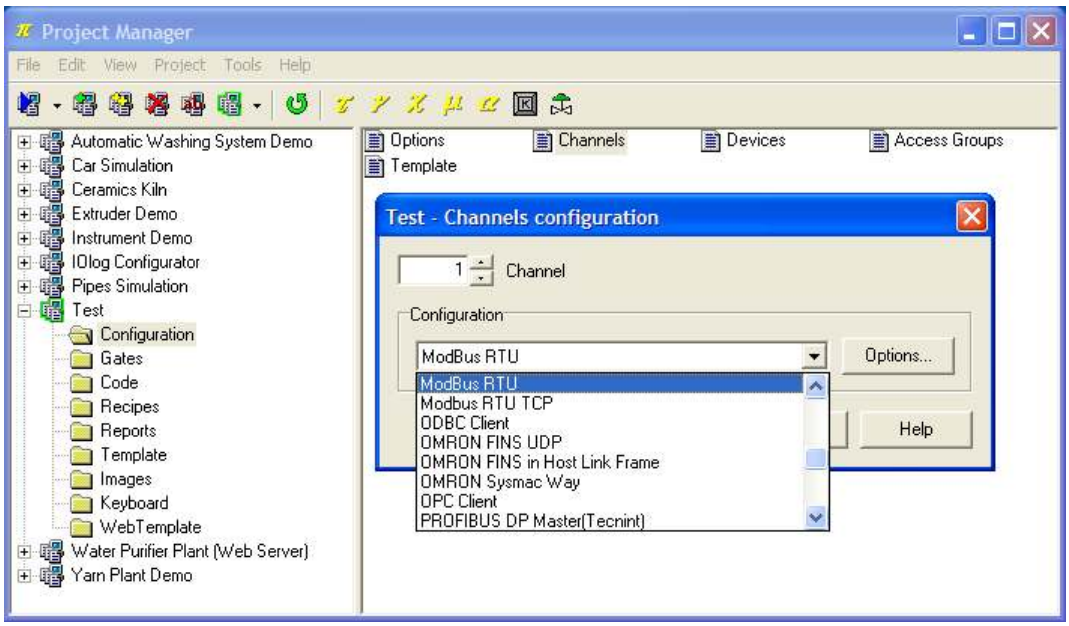
Project creation

Press `Ok` button and then `Create Application` button; in this way you create a tree structure with all supervision project elements.

3. Communication channel configuration

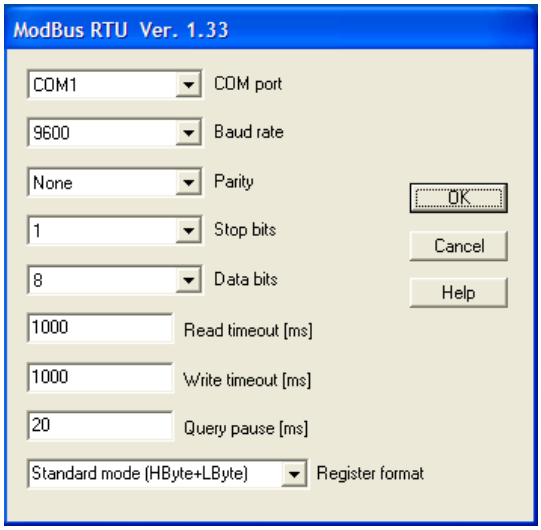
From elements in `Configuration` folder select `Channels`.

Define the logic channel 1 to communicate in `Modbus RTU`.



Protocol selection

Press button `Options...` and select the serial port to assign to the channel (for example COM1). It is necessary to set serial port specifying `Baud rate`, `Parity`, `Stop bits`, `Data bits`, `Time out` e `Query Pause` (for these parameters refer to manufacturer data device, an example is provided in the figure)



Protocol configuration

4. Creating local variables database

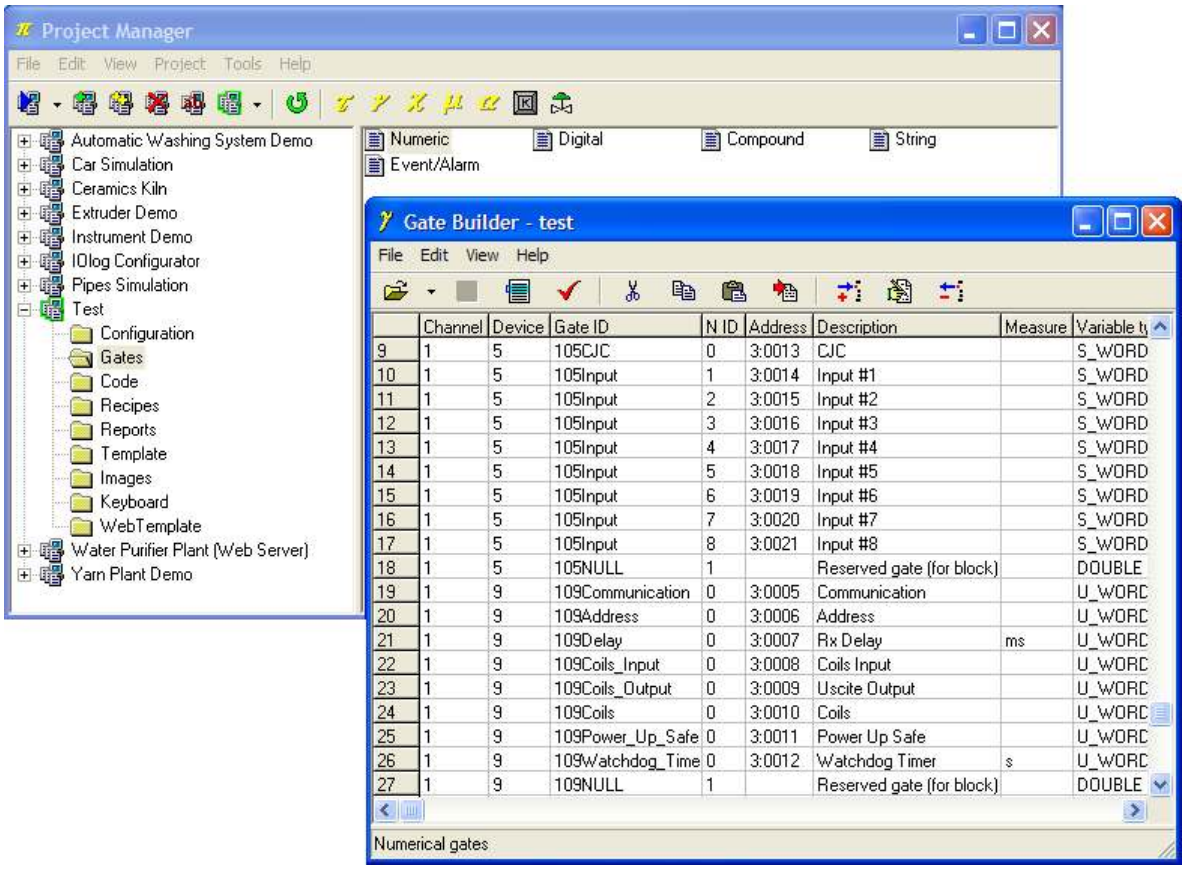
Now we can insert in gates database the local variables required to define 8 setpoint values necessary for ON/OFF control.

Numeric gates include all those variables that refer to an analog quantity (for example: measured variables, set-points, alarm threshold ..) and can be expressed by a byte, a word, a double word, an integer or by a floating-point variable.

For a more ehxaustive discussion, refer to the guide *"Getting started - Creating a simple Modbus protocol application*".

To edit the variables database, you need to run **Gate Builder**

From **Project Manager**, select Gates folder and double-click on each of icons (`Numeric, Digital`, ...).



	Channel	Device	Gate ID	N ID	Address	Description	Measure	Variable ty
9	1	5	105CJC	0	3:0013	CJC		S_WORD
10	1	5	105Input	1	3:0014	Input #1		S_WORD
11	1	5	105Input	2	3:0015	Input #2		S_WORD
12	1	5	105Input	3	3:0016	Input #3		S_WORD
13	1	5	105Input	4	3:0017	Input #4		S_WORD
14	1	5	105Input	5	3:0018	Input #5		S_WORD
15	1	5	105Input	6	3:0019	Input #6		S_WORD
16	1	5	105Input	7	3:0020	Input #7		S_WORD
17	1	5	105Input	8	3:0021	Input #8		S_WORD
18	1	5	105NULL	1		Reserved gate (for block)		DOUBLE
19	1	9	109Communication	0	3:0005	Communication		U_WORD
20	1	9	109Address	0	3:0006	Address		U_WORD
21	1	9	109Delay	0	3:0007	Rx Delay	ms	U_WORD
22	1	9	109Coils_Input	0	3:0008	Coils Input		U_WORD
23	1	9	109Coils_Output	0	3:0009	Uscite Output		U_WORD
24	1	9	109Coils	0	3:0010	Coils		U_WORD
25	1	9	109Power_Up_Safe	0	3:0011	Power Up Safe		U_WORD
26	1	9	109Watchdog_Time	0	3:0012	Watchdog Timer	s	U_WORD
27	1	9	109NULL	1		Reserved gate (for block)		DOUBLE

Variables database creation

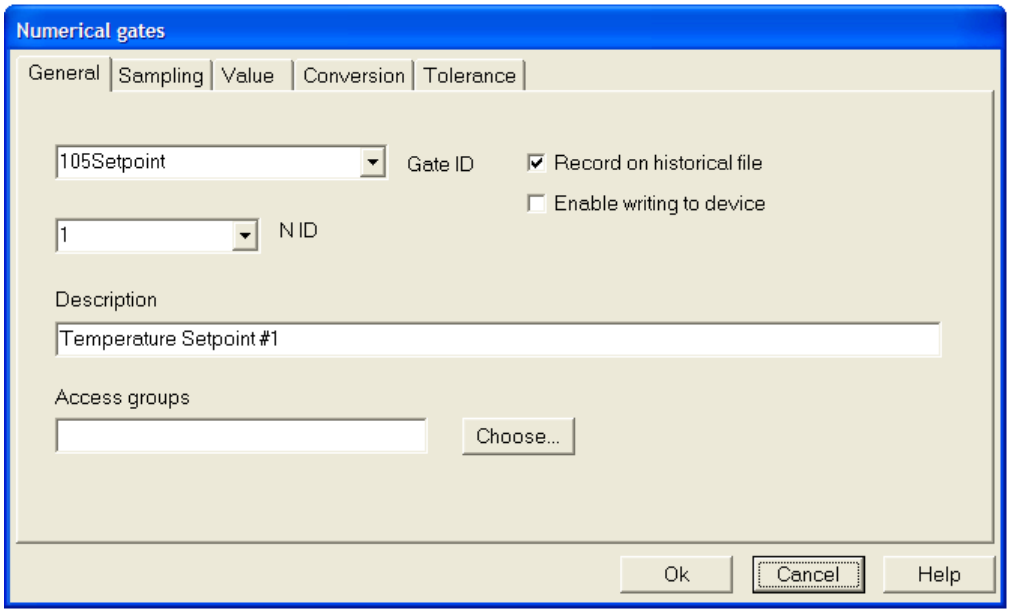
Suppose you need to add the following variables to the variables created by the *Application Builder*:

Name	Channel	Device	Address	Type	Unit	Description
105Setpoint 1	1	5	Internal Variable	Signed Word	°C	Temperature Setpoint Ch 1
105Setpoint 2	1	5	Internal Variable	Signed Word	°C	Temperature Setpoint Ch 2
105Setpoint 3	1	5	Internal Variable	Signed Word	°C	Temperature Setpoint Ch 3
105Setpoint 4	1	5	Internal Variable	Signed Word	°C	Temperature Setpoint Ch 4
105Setpoint 5	1	5	Internal Variable	Signed Word	°C	Temperature Setpoint Ch 5
105Setpoint 6	1	5	Internal Variable	Signed Word	°C	Temperature Setpoint Ch 6
105Setpoint 7	1	5	Internal Variable	Signed Word	°C	Temperature Setpoint Ch 7
105Setpoint 8	1	5	Internal Variable	Signed Word	°C	Temperature Setpoint Ch 8

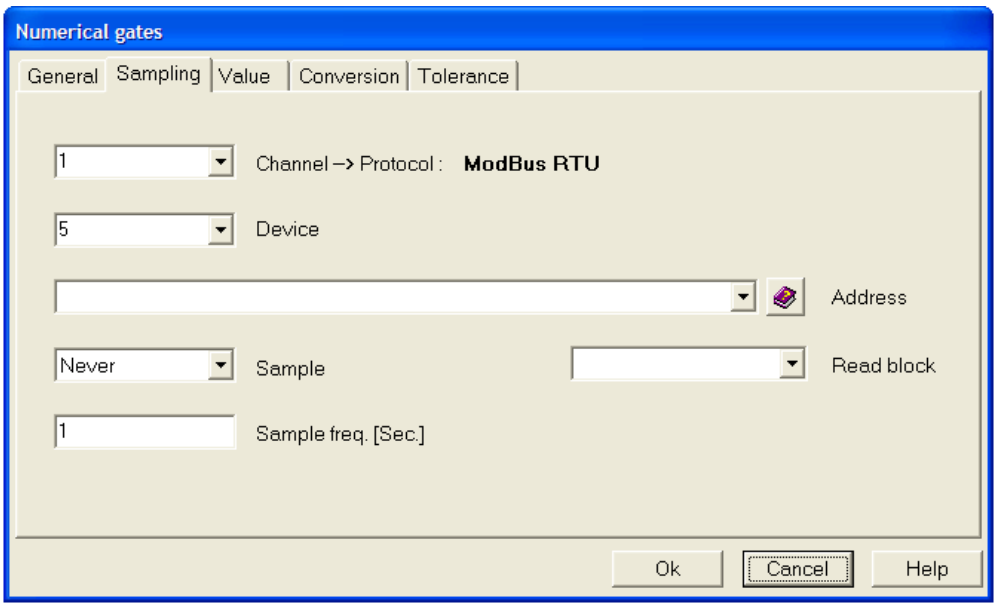
4.1 Numeric variables configuration

Execute numeric gates 105Setpoint 1 - 105Setpoint 8 configuration.

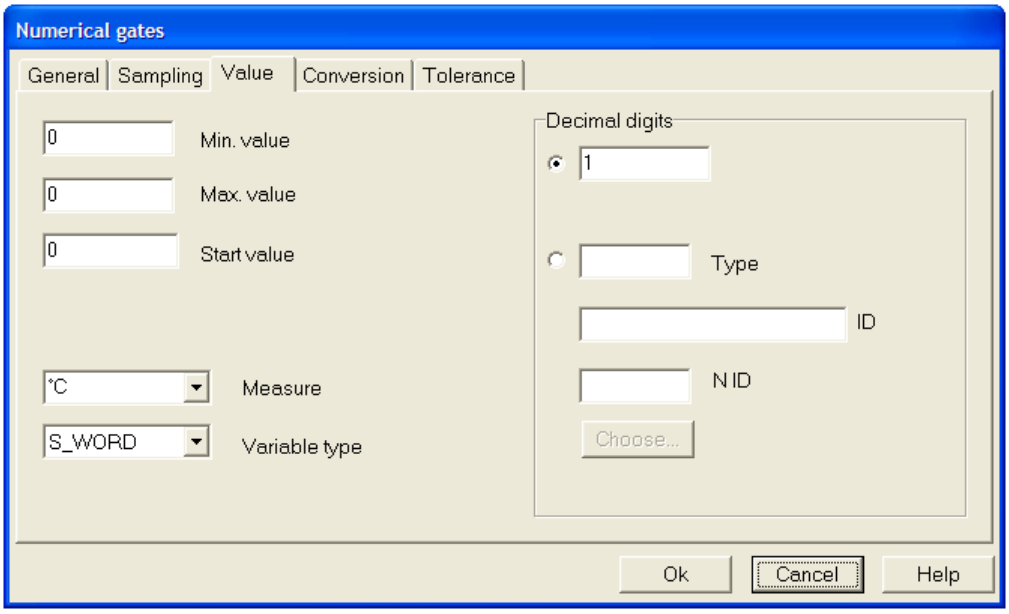
Configuration of numeric gate 105Setpoint 1



105Setpoint 1 *numeric variable configuration – General folder*



105Setpoint 1 *numeric variable configuration – Sampling folder*

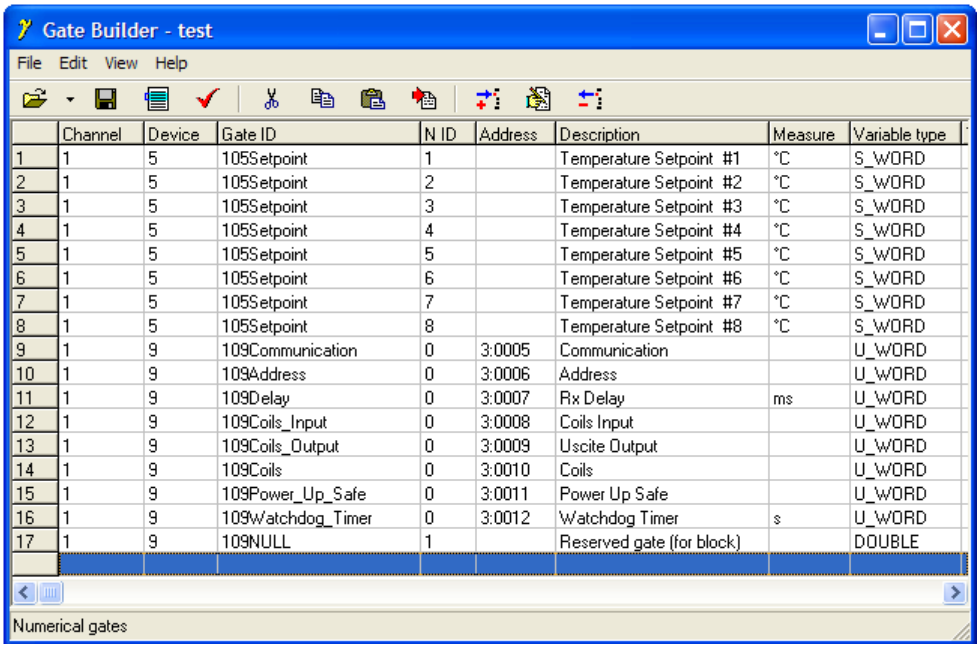


105Setpoint 1 *numeric variable configuration – Value folder*

After defining 105Setpoint 1 variable copy and paste 7 times in **Gate Builder** the database row related to this variable, verifying that automatically `N ID` parameter increases

End result

After you have defined all numeric variables, you should see the **Gate Builder** main page similar to the one shown below.



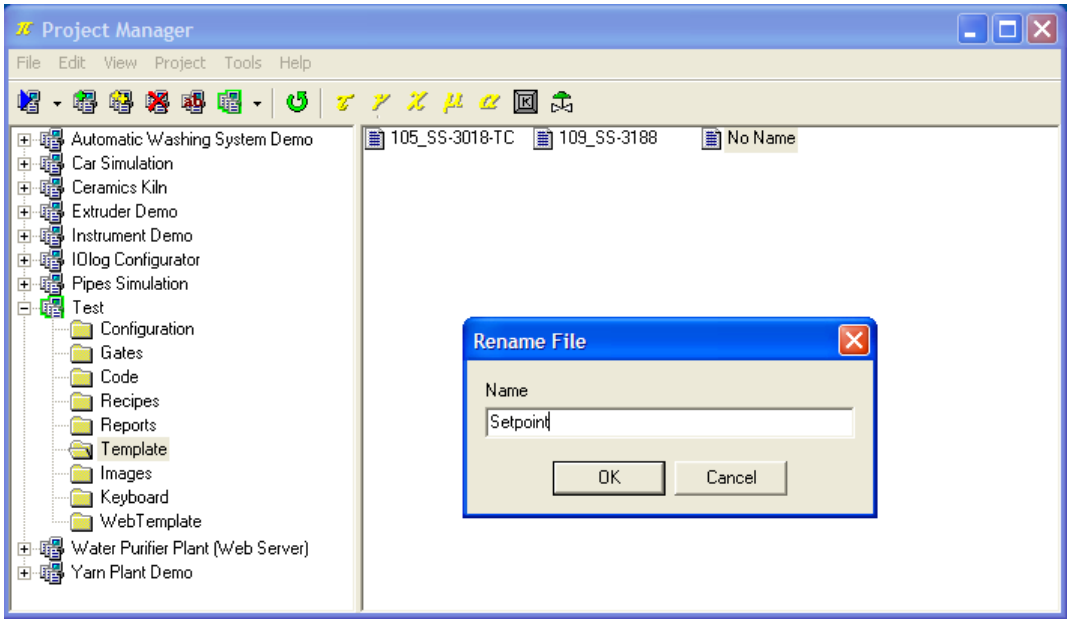
	Channel	Device	Gate ID	N ID	Address	Description	Measure	Variable type
1	1	5	105Setpoint	1		Temperature Setpoint #1	°C	S_WORD
2	1	5	105Setpoint	2		Temperature Setpoint #2	°C	S_WORD
3	1	5	105Setpoint	3		Temperature Setpoint #3	°C	S_WORD
4	1	5	105Setpoint	4		Temperature Setpoint #4	°C	S_WORD
5	1	5	105Setpoint	5		Temperature Setpoint #5	°C	S_WORD
6	1	5	105Setpoint	6		Temperature Setpoint #6	°C	S_WORD
7	1	5	105Setpoint	7		Temperature Setpoint #7	°C	S_WORD
8	1	5	105Setpoint	8		Temperature Setpoint #8	°C	S_WORD
9	1	9	109Communication	0	3:0005	Communication		U_WORD
10	1	9	109Address	0	3:0006	Address		U_WORD
11	1	9	109Delay	0	3:0007	Rx Delay	ms	U_WORD
12	1	9	109Coils_Input	0	3:0008	Coils Input		U_WORD
13	1	9	109Coils_Output	0	3:0009	Uscite Output		U_WORD
14	1	9	109Coils	0	3:0010	Coils		U_WORD
15	1	9	109Power_Up_Safe	0	3:0011	Power Up Safe		U_WORD
16	1	9	109Watchdog_Timer	0	3:0012	Watchdog Timer	s	U_WORD
17	1	9	109NULL	1		Reserved gate (for block)		DOUBLE

Numeric variable database

5. Creating the setpoints template

Now it is necessary to create a template to insert the control setpoint values .

Select `Template` folder and create a new template, selecting the item `New>File` from `Edit` menu. Rename the just created template using the name `Setpoint`, do this selecting it and then using `Rename` item from `Edit` menu



Template creating

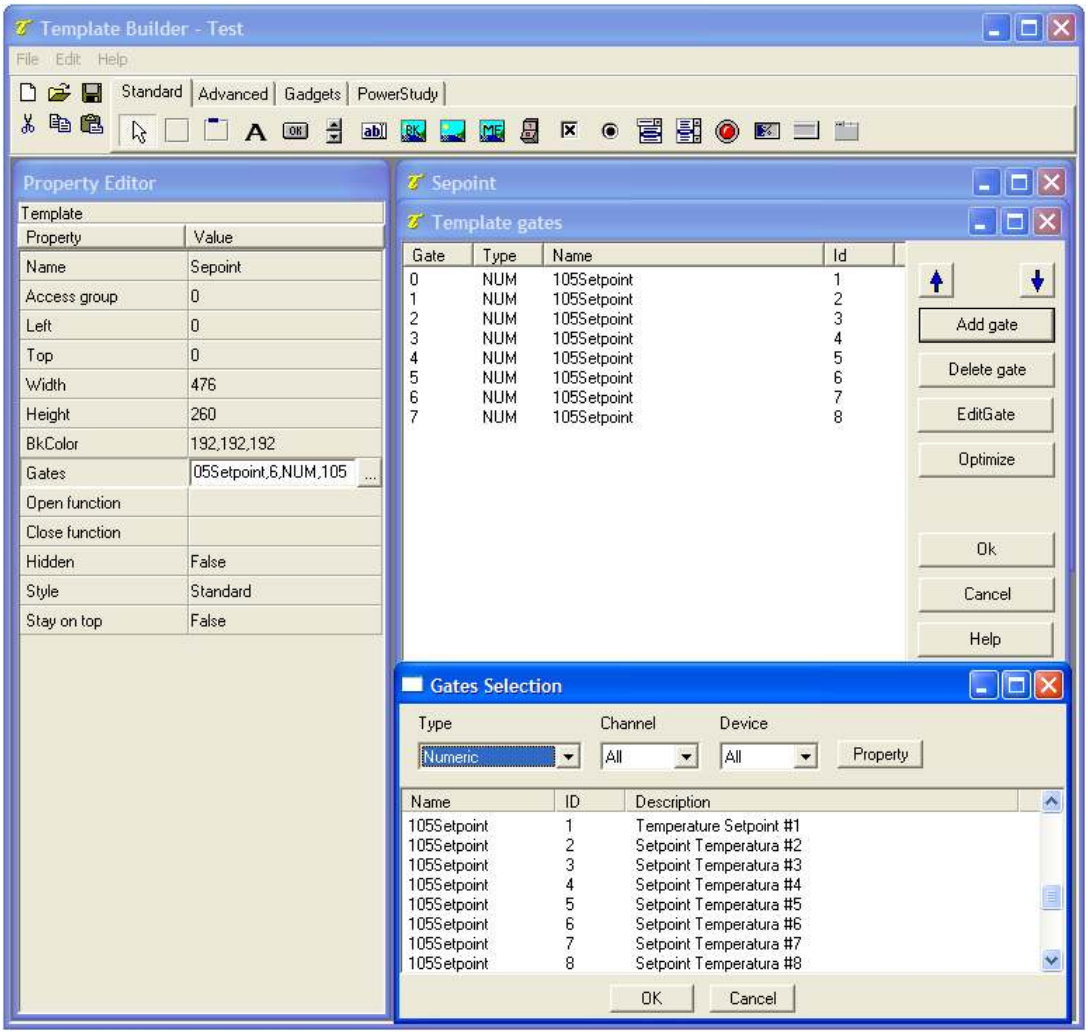
Double-clicking on created template, **Template Builder** start in order to build the graphic page.

5.1 Declaring template variables

First it is necessary to declare which variables we will use in the template; in this example we will use setpoint related variables.

Click on button  alongside of the Gates item in the *Property Editor* (Property Editor is the window on the left side of the screen that allows to modify template elements properties).

A new windows will appear; press `Add gate` button, select the first numeric gate and press `Ok`. Repeat this operation for each numeric, digital and alarm gate that belongs to the application.



Template variables declaration

5.2 Inserting a Label object

Firstly build a *Frame* that will contain all the elements that will be inserted later.

To do this, select *Frame* object among the ones on the upper bar (, it is the first on the left) and click on the template, a void rectangle will be displayed.

The next step is to insert into the created frame a static label that is a static text; select *Label* object among the ones on the upper bar ( ),then click into the frame. To modify the text displayed into the object, use Property Editor, click alongside of the property `Label` and digit `SETPOINT 1`.

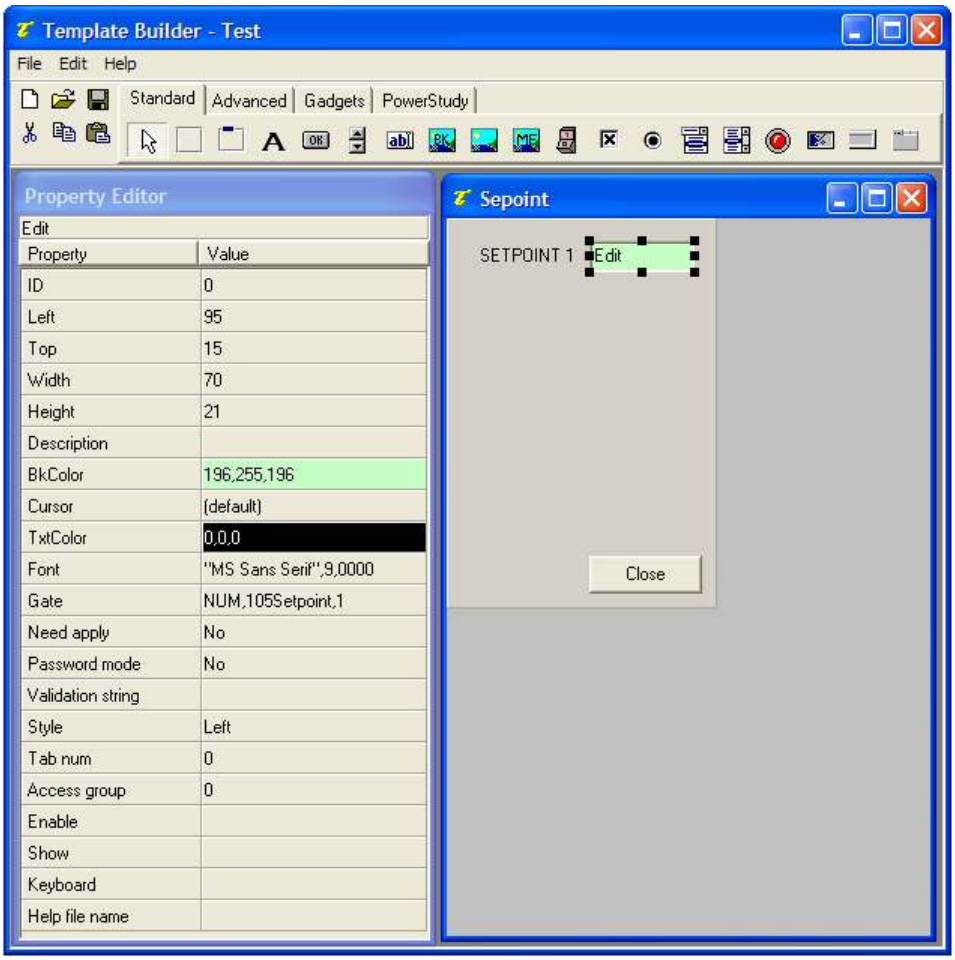
Every described object can be formatted and placed as you like using Property Editor.

5.3 Inserting an *Edit* object

A control will be inserted that will allow to modify the value of the 105*Setpoint1* gate and to send it to the software.

Select *Edit* ( );then click into the frame.

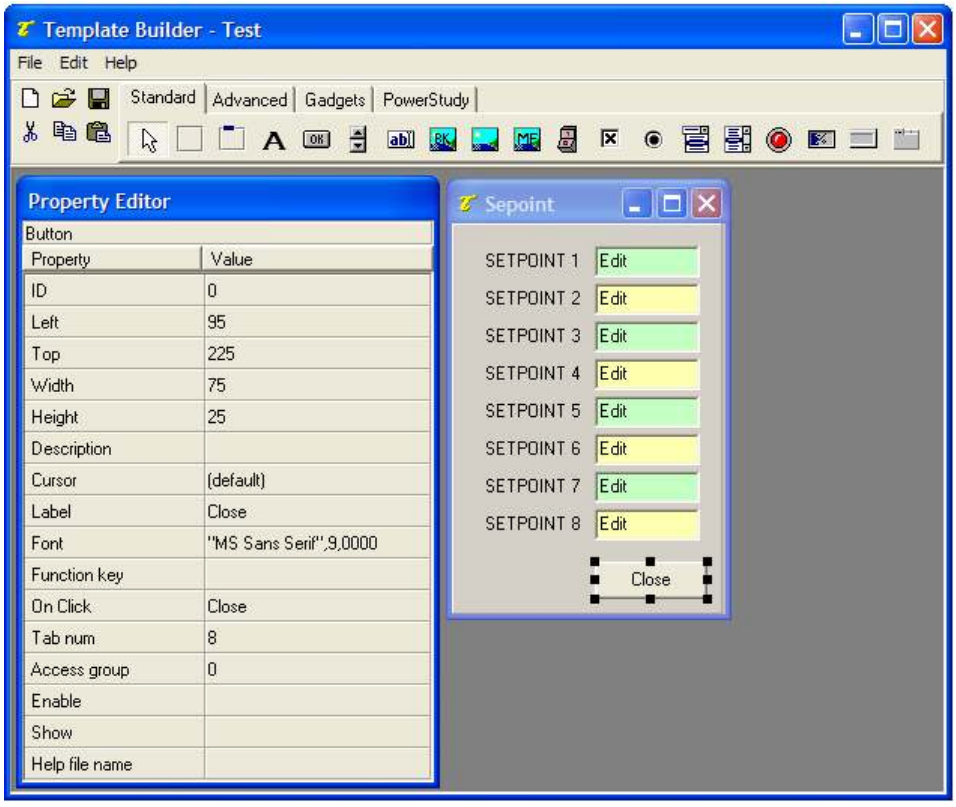
To link the *Edit object* to the numeric variable *105Setpoint1*, click on the button ... alongside of the item `Gate` in Property Editor and select `NUM,105Setpoint,1` among the available gates.



Edit object inserting

Copy and paste *Label* and *Edit* objects 7 times, taking care to change text in *Label*s and reference gate in *Edit.*

The template is now ready and can be refined, modifying object properties.

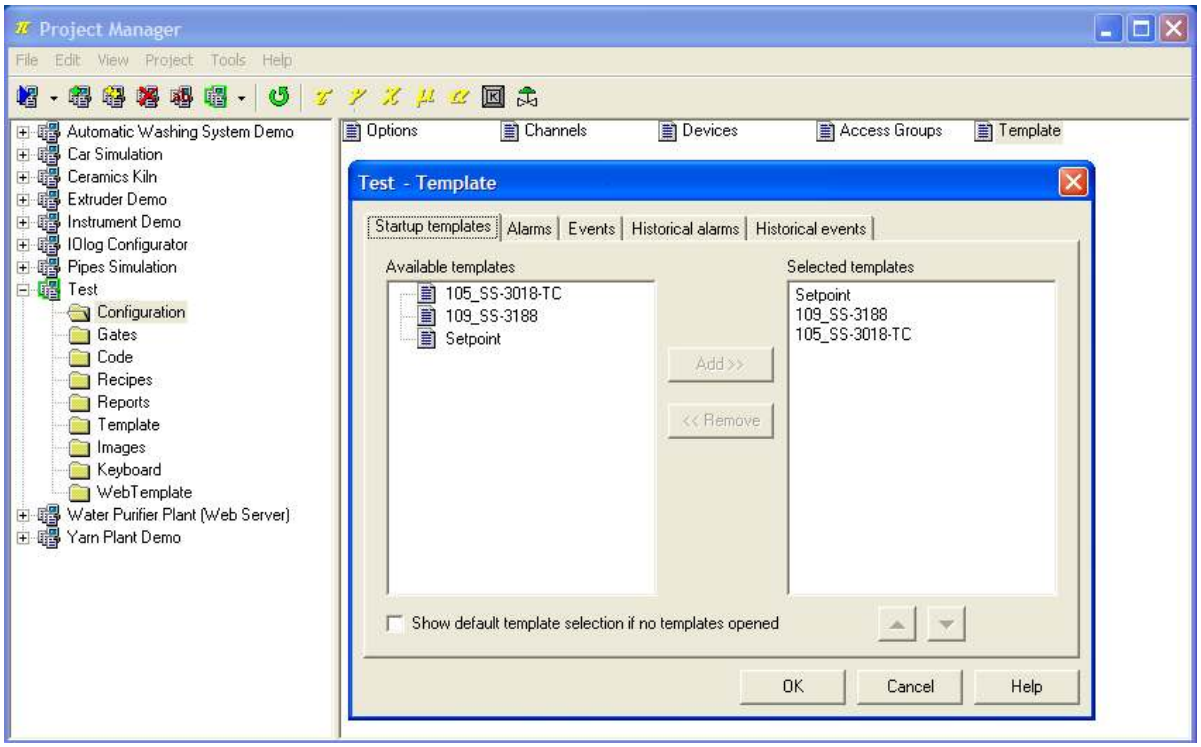


Template completion

6. Automatic display of templates

It is possible to show templates automatically at Winlog program startup.

From elements in `Configuration` folder select `Template`.



Choosing the startup templates

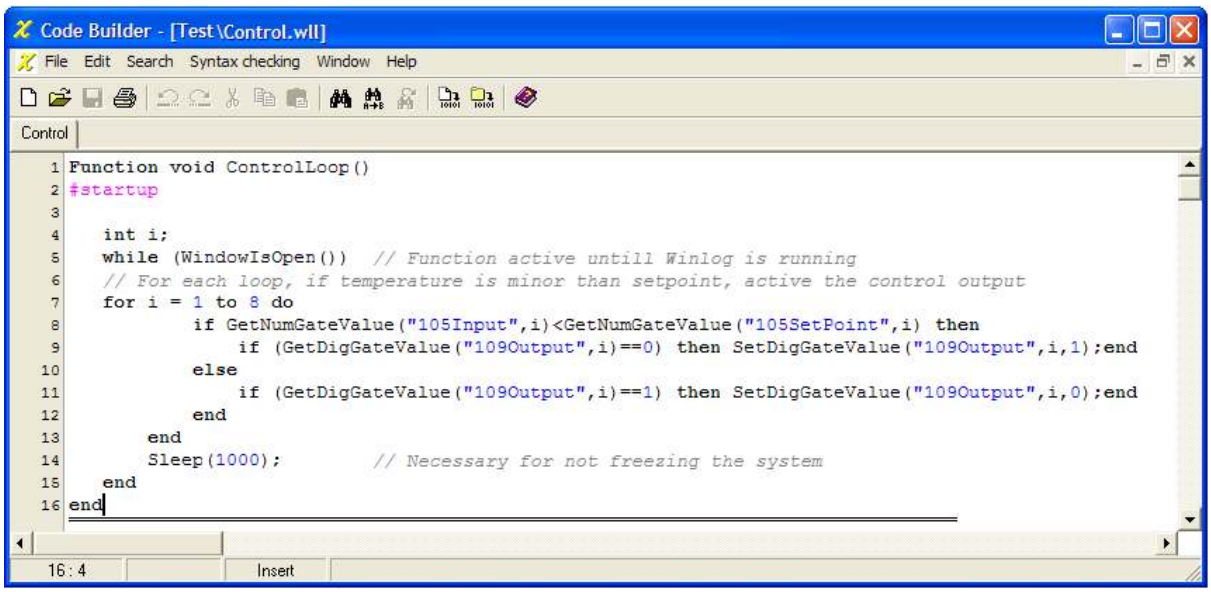
In `Startup templates` choose the templates that you want to start automatically.

At project startup, chosen templates will appear automatically.

7. Winlog Pro control code example

Now create the code function that allows to realize the ON/OFF control.

In `Code` folder create a file and rename it `Control`; opening it, **Code Builder** starts.



Creating a code file

Code Builder is the **Winlog Pro** programming environment; in this example we will use it only to define a function that will realize the ON/OFF control at the application startup.

Copy and paste the following code:

```
Function void ControlLoop()

#startup // Function called at Winlog startup
TPageOpen("Setpoint");

int i;

while (WindowIsOpen())  // Function active until Winlog is running

// For each loop, if temperature is minor than setpoint, active the control output

for i = 1 to 8 do

if GetNumGateValue("105Input",i)<GetNumGateValue("105SetPoint",i) then

if (GetDigGateValue("109Output",i)==0) then SetDigGateValue("109Output",i,1);end

else

if (GetDigGateValue("109Output",i)==1) then SetDigGateValue("109Output",i,0);end

end

end

Sleep(1000);        // Necessary for not freezing the system

end

end
```

To check syntax of the code use function `Check syntax` ().

8. Project execution

Our example is complete.

Link devices to the serial port; to run the project, in **Project Manager** select `Execute...` from `Project menu`.

Now we are entering in the "run-time" phase that is application execution mode. **Winlog Lite/Pro** samples variables from devices and processes results in graphical representations (trends and template) and in tabular representations (reports and historical data).

At project startup, main template will appear automatically.

From `Supervision` menu you can display graphical trends; select menu item `Charts...` and define the group of variables that you want to display as graphical trends.
Again in `Supervision` menu you can display both the online status (`Status>Alarms...`) and the story (`Historical>Alarms`...) of all alarms that have been created with **Gate Builder**.