Getting started

Creating a simple OPC Client support application

Revisioni del documento

Data	Edizione	Commenti
24/11/2009	1.0	-
24/11/2009	2.0	-

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

http://www.sielcosistemi.com

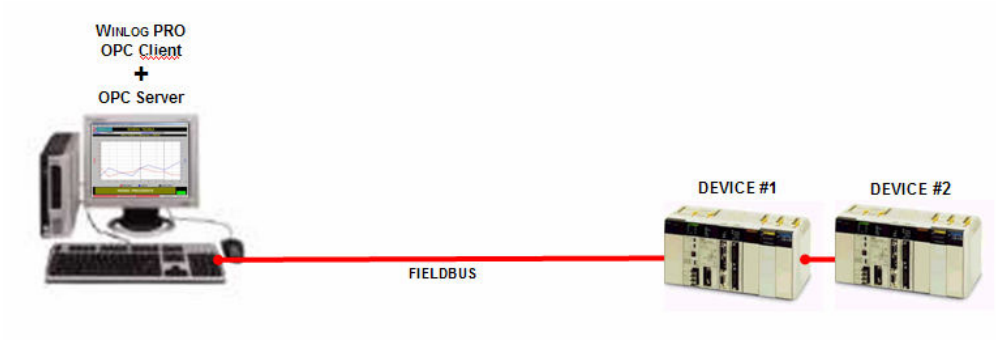
1. Introduzione

In this guide we propose, as an example, the design of a simple supervision application with OPC Client support; 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 Pro** software, to quickly understand how to communicate with external devices

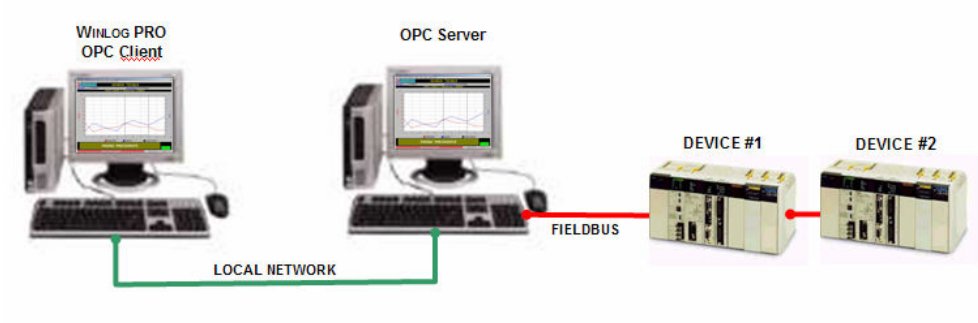
OPC (OLE for Process Control) is an industry standard created in collaboration with a number of worldwide leading automation hardware and software manufacturers. OPC allows software components such as software connectors to be combined and enables these components to intercommunicate with no need of special adaptions.

Winlog Pro OPC Client driver support data access (DA) to OPC servers DA 1.0 and 2.0; It can connect to local OPC servers through COM object or to remote OPC servers (in the local network) trough DCOM object. In case of remote server, you must ensure that DCOM is properly configured.

DCOMCNFG is a Windows tool that allows users to configure the DCOM settings.Before you can access a COM component via DCOM, you must provide the authentication credentials of a user who has been granted permission to access/launch the component.



Local OPC Server



Remote OPC Server

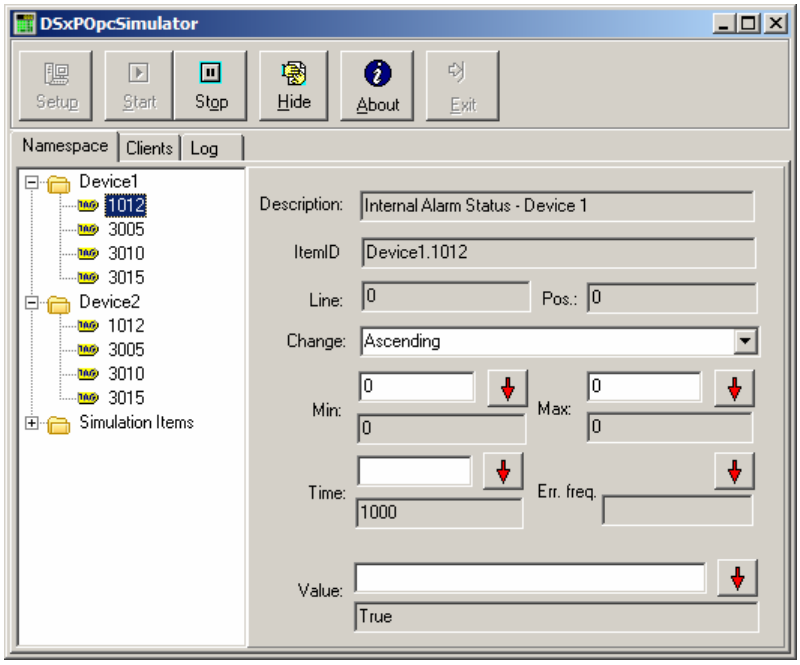
DA OPC Server is organized in a structure of groups and items that are directly connected to device or PLC internal variables, so the problem of communication protocol with them is solved by the specific OPC server.

OPC Client communicates with OPC servers always in the same mode without need to know the specific device or PLC communication protocol.

The first thing to do is to install the OPC server on the computer, and configure it defining communication parameters and items (devices read or write variables); Item full name (usually compound by DeviceName+GroupName+ItemId) does not be more than 80 chars.

After that open ProjectManager and create a new project, then select ProjectManager->Configuration->Channel and choose OPC Client protocol and configure it by selecting the computer and OPC server name to connect to

In our example we will use an OPC Server simulator installed in the same Winlog Pro computer; in particular we used DSXP OPC Simulator Ver1.2 by DET Informatica sas, that can be freely downloaded from the website http://www.dsxp.com and can be freely redistributed. This software can simulate variables trends in an easy way.

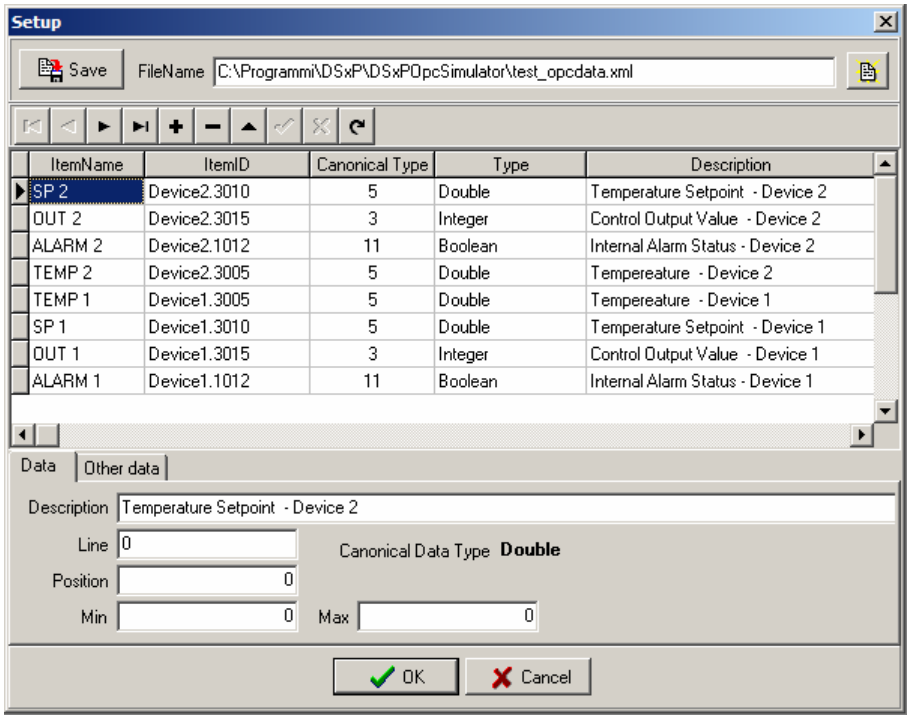


OPC Server simulator

Every time 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.

We suppose to have to communicate, using an OPC Modbus server, with two devices (Test Device#1 e Test Device#2) whose address are 1 and 2; for each device we will simulate 3 numeric variables (Temp, Sp e Out) and 1 digital variable (Alarm).

We set OPC server simulator to provide these variables as it obtain them, using a Modbus protocol, from two field devices; you can operate in a similar way for each different protocol.



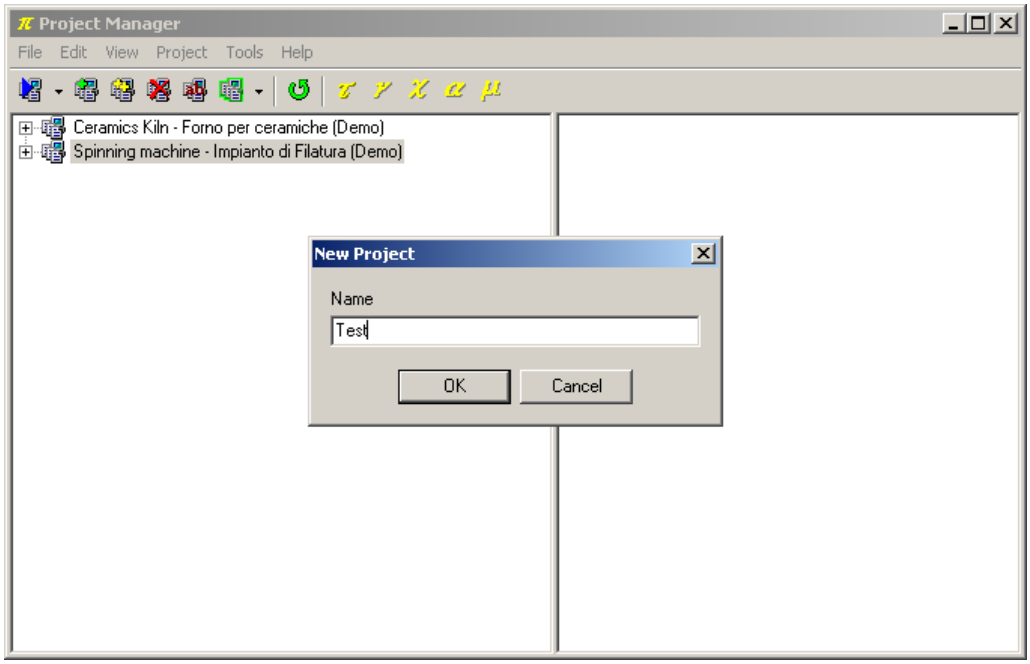
OPC Server simulator configuration

2. Creating the project

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

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

Select `New` from `Project` menu and insert the project name (for example `Test`).



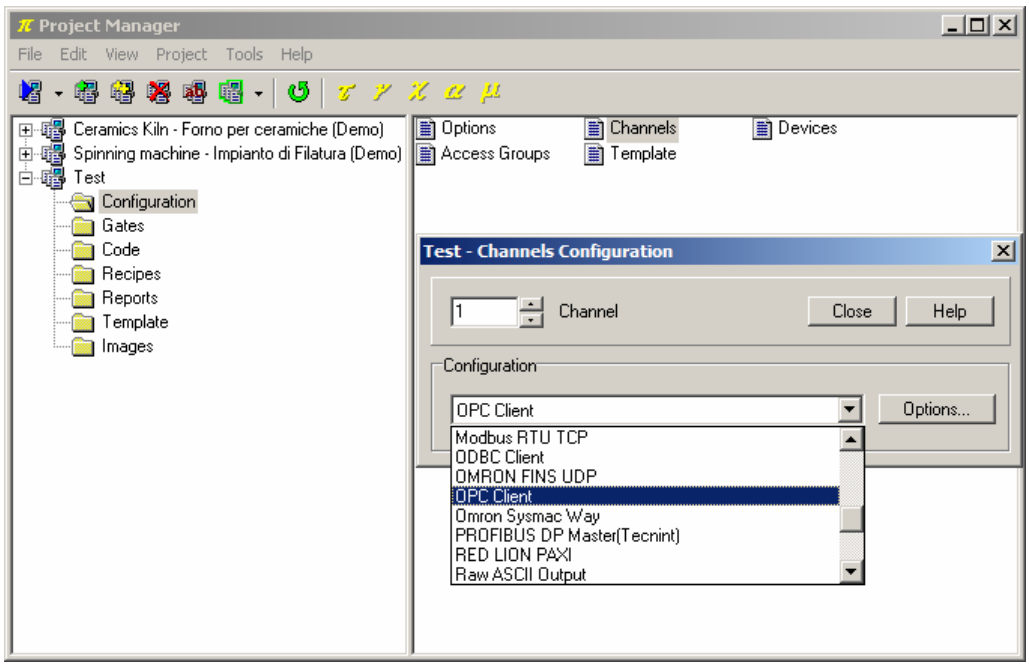
Project creation

In this way you create a tree structure with all supervision project elements.

3. Communication channel configuration

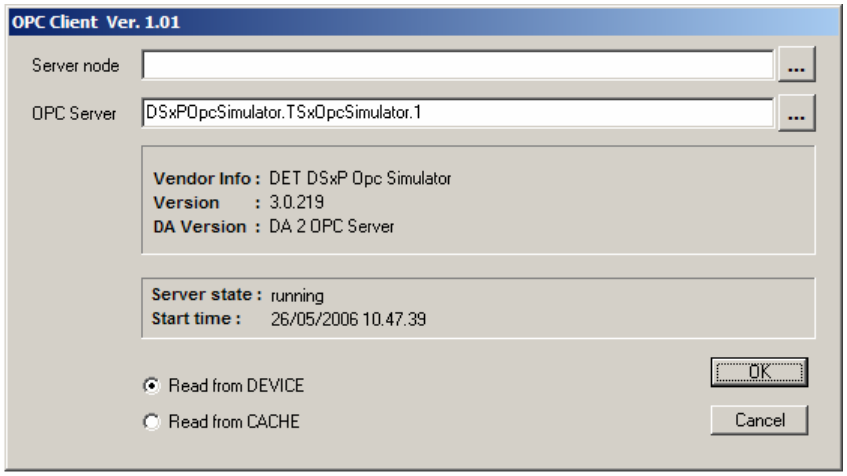
From elements in `Configuration` folder select `Channels`.

Define the logic channel 1 to communicate `OPC Client support`.



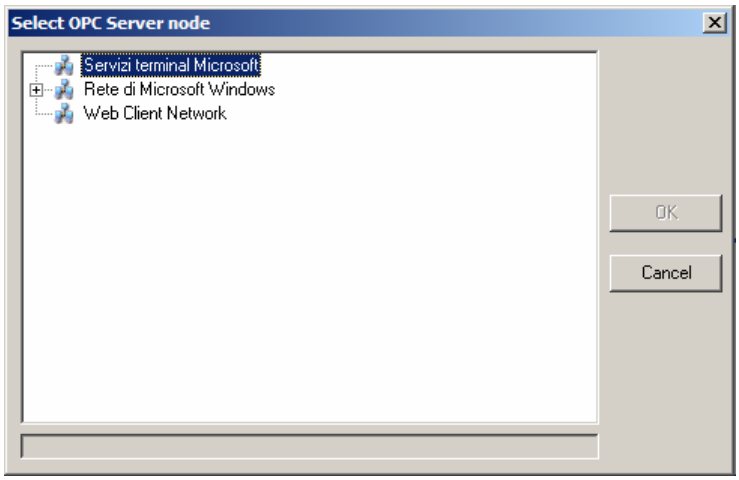
Protocol selection

Press button `Options`.

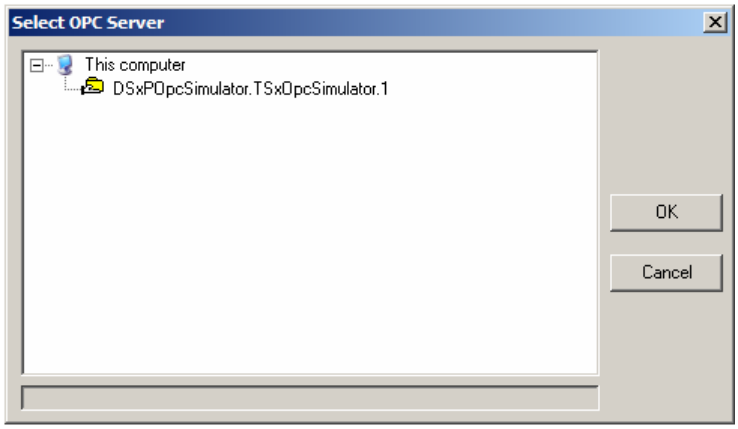


Protocol configuration

Choose the `Server node` that is the network node on which the OPC server is installed (for example the local computer) and choose the `OPC Server` between the available ones on the node.



OPC Server network node choice



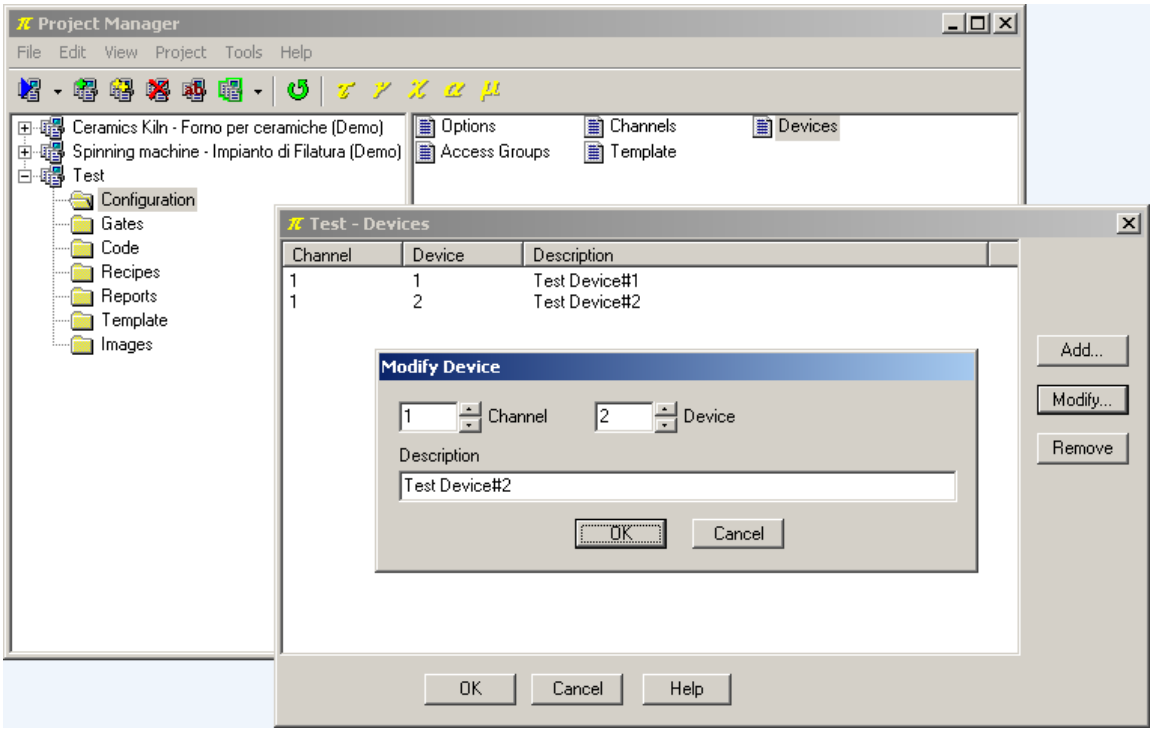
OPC Server choice

It is also necessary to choose whether to read data directly from devices (`Read from DEVICE`) or OPC Server cache memory (`Read from CACHE`)

4. Devices declaration

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

Insert `Test Device#1` and `Test Device#2`, respectively at address 1 and 2 on logic channel 1 previously set.



Devices declaration

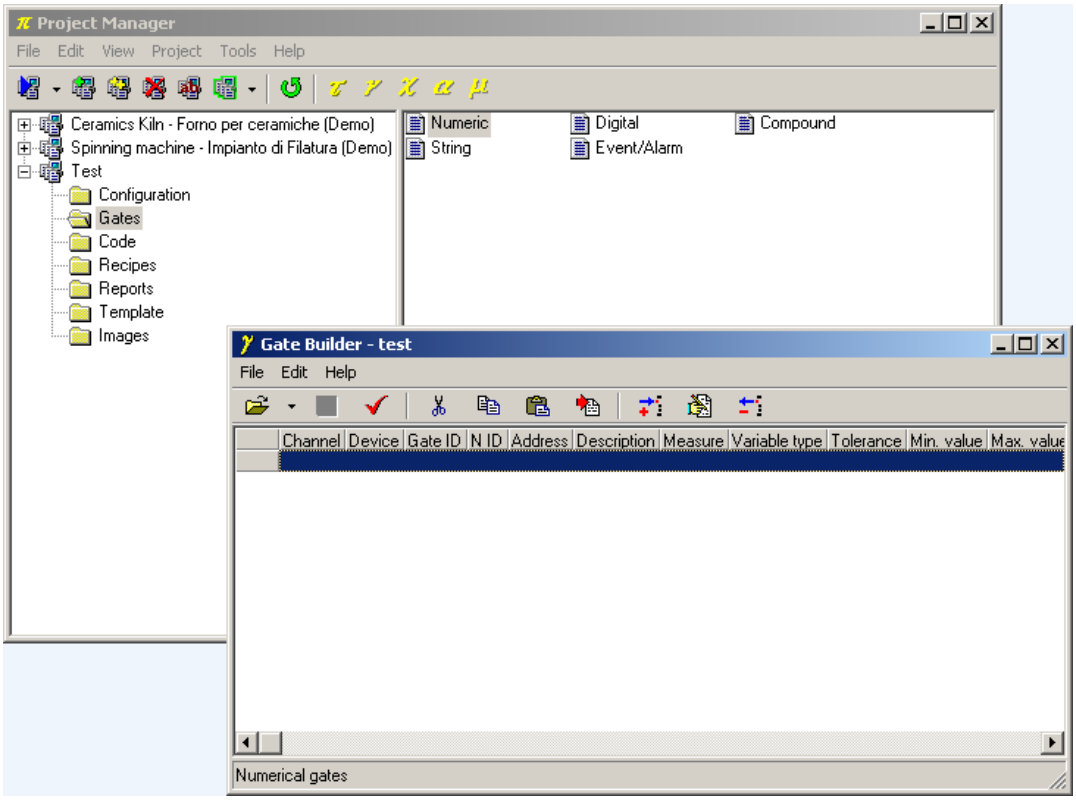
5. Creating variables database

Now we can insert the devices variables in gates database

In this example we only consider numeric and digital variables (gates).
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.
Digital gates include all those variables that refer to digital status (for example an alarm conditions, a configuration option, ...) and can be expressed by a single bit.
Sometimes more digital conditions can be gathered in a single numeric variable, but this case will not be explained in this example.

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`, ...).



Variables database creation

Suppose you need to read the following variables (to adapt this example to a real case it is enough to modify the gates details below).

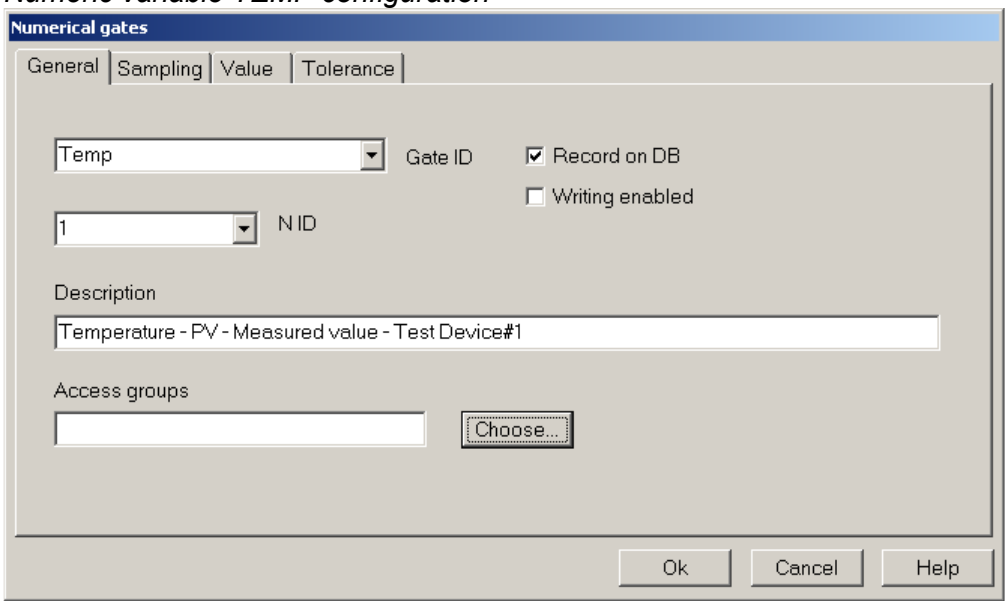
Name	Channel	Device	OPC address*	Variabile type	Gate Type	Unit	Description
TEMP	1	1	Device1.3005	Signed Word	Numeric	°C	Temperature - Measure
SP	1	1	Device1.3010	Signed Word	Numeric	°C	Temperature - Setpoint
OUT	1	1	Device1.3015	Unsigned Word	Numeric	%	Control Output - Value
ALARM	1	1	Device1.1012	Bit	Digital		Internal alarm status
TEMP	1	2	Device2.3005	Signed Word	Numeric	°C	Temperature - Measure
SP	1	2	Device2.3010	Signed Word	Numeric	°C	Temperature - Setpoint
OUT	1	2	Device2.3015	Unsigned Word	Numeric	%	Control Output - Value
ALARM	1	2	Device2.1012	Bit	Digital		Internal alarm status

* OPC address can be already defined in the OPC Server or user defined; it must be no longer than 80 characters (please read the protocol guide in Project Manager Help).

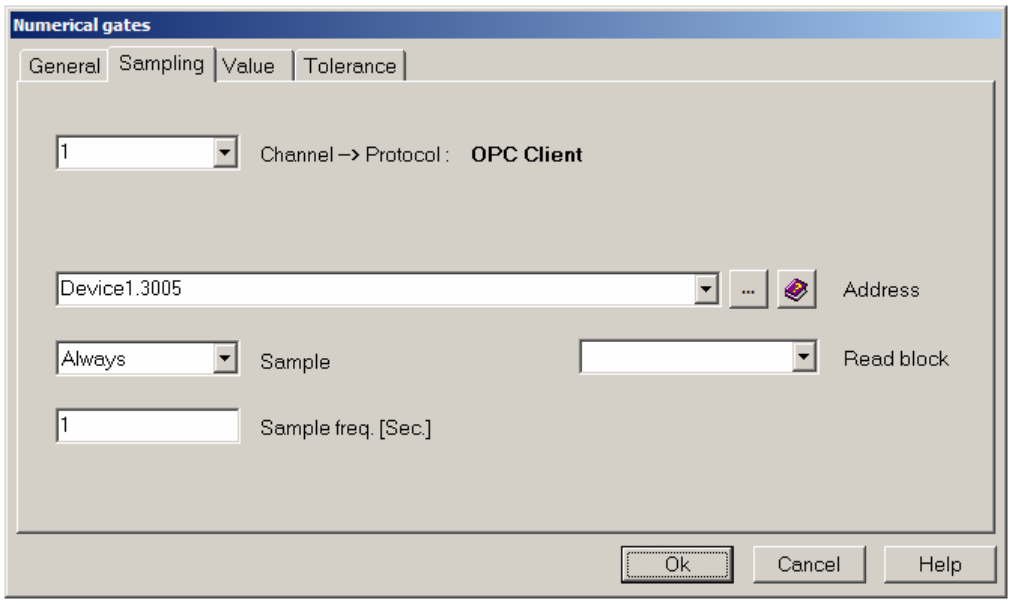
5.1 Numeric variables configuration

Repeat numeric gates configuration for both devices, having care to change device number (`Device`) and `N ID`.

Numeric variable TEMP configuration

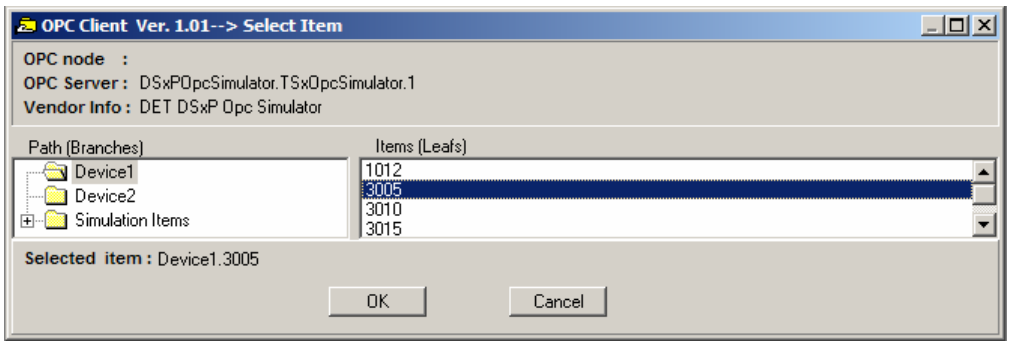


TEMP numeric variable configuration – General folder

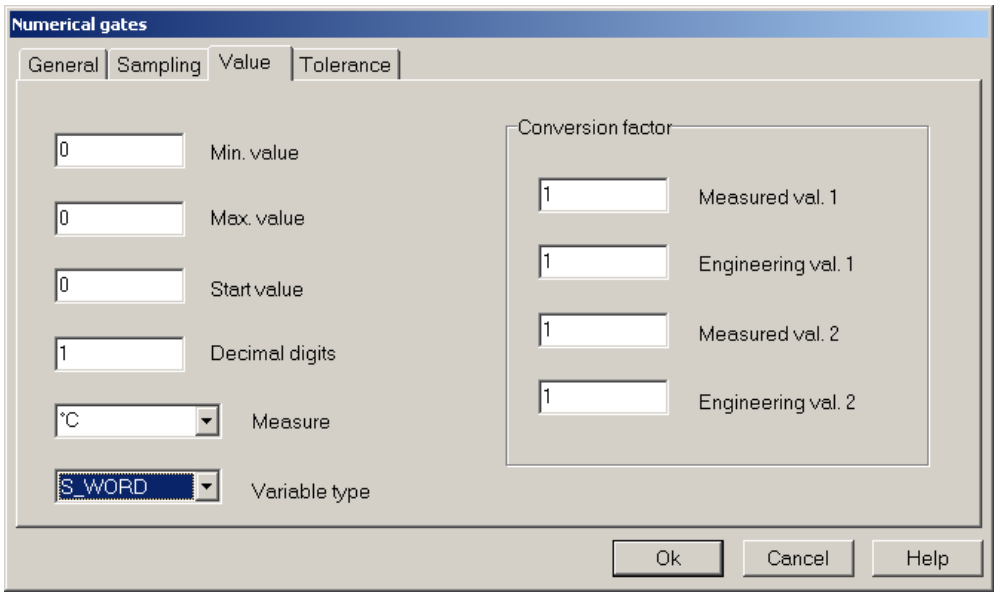


TEMP numeric variable configuration – Sampling folder

Click on button ... and choose the variable among the ones available on the OPC Server

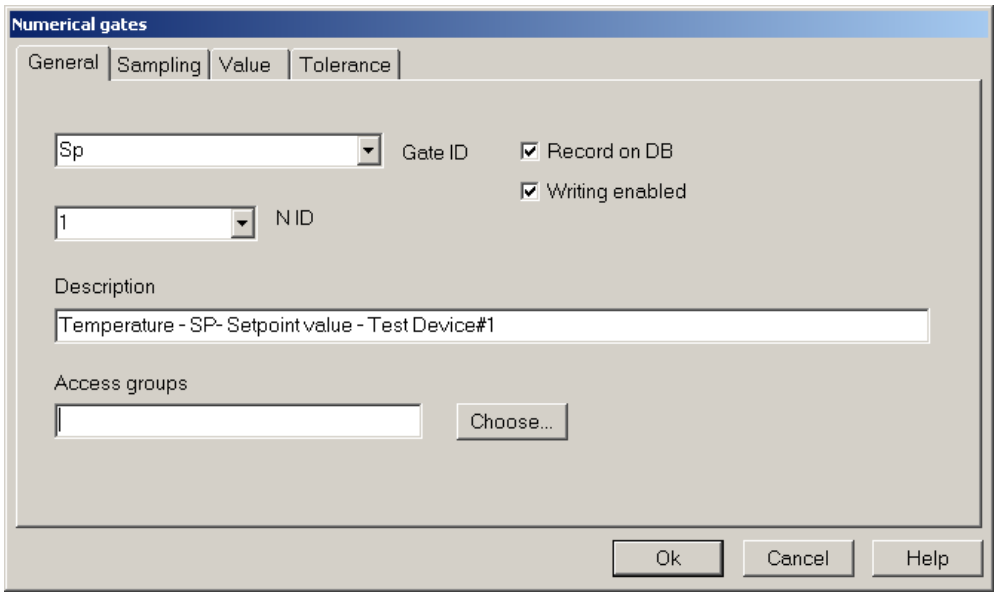


TEMP numeric variable configuration – variable choice from OPC Server

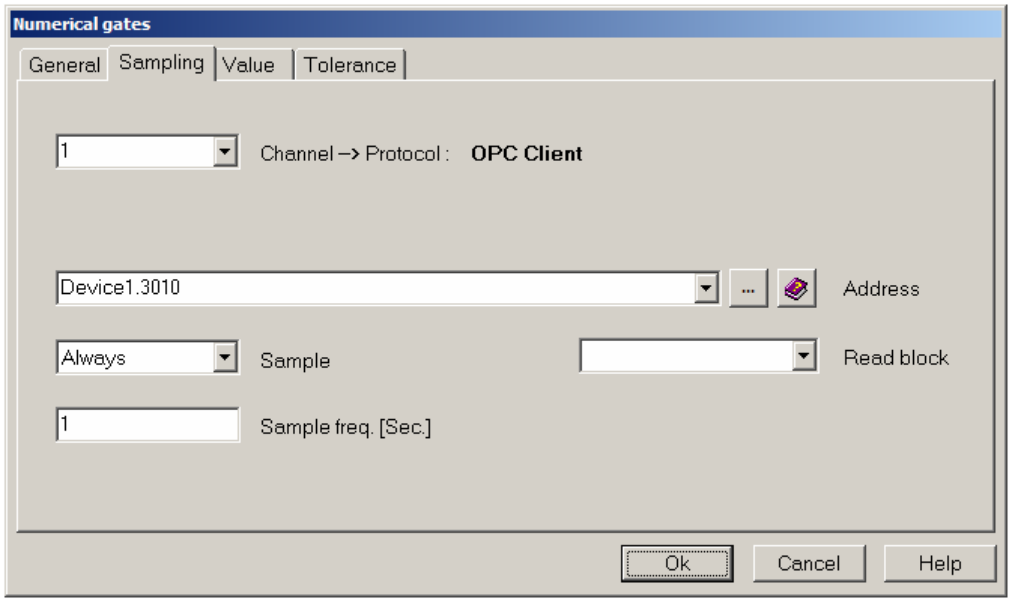


TEMP numeric variable configuration – Value folder

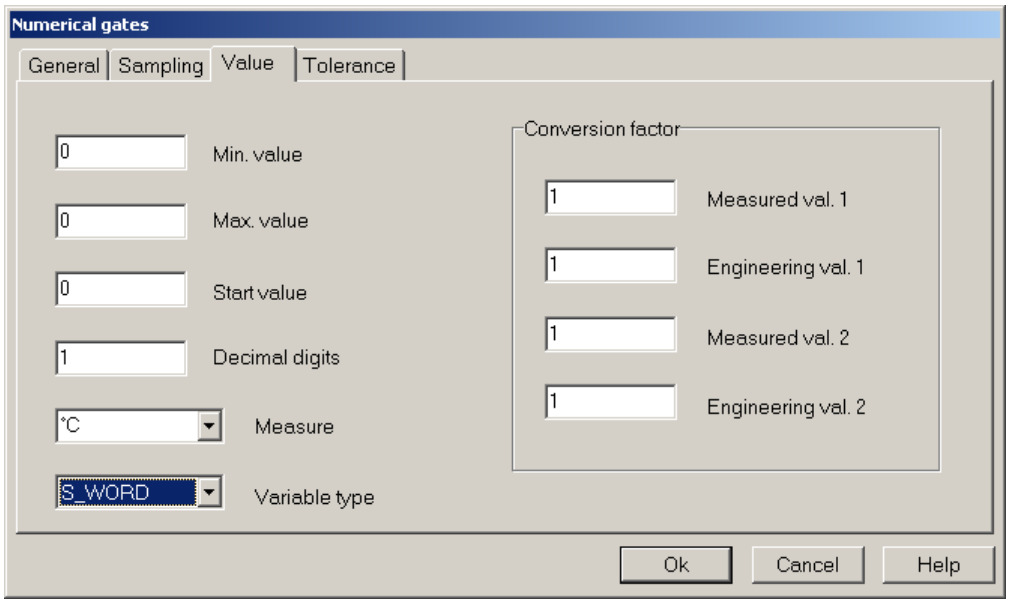
Numeric variable SP configuration



SP numeric variable configuration – General folder

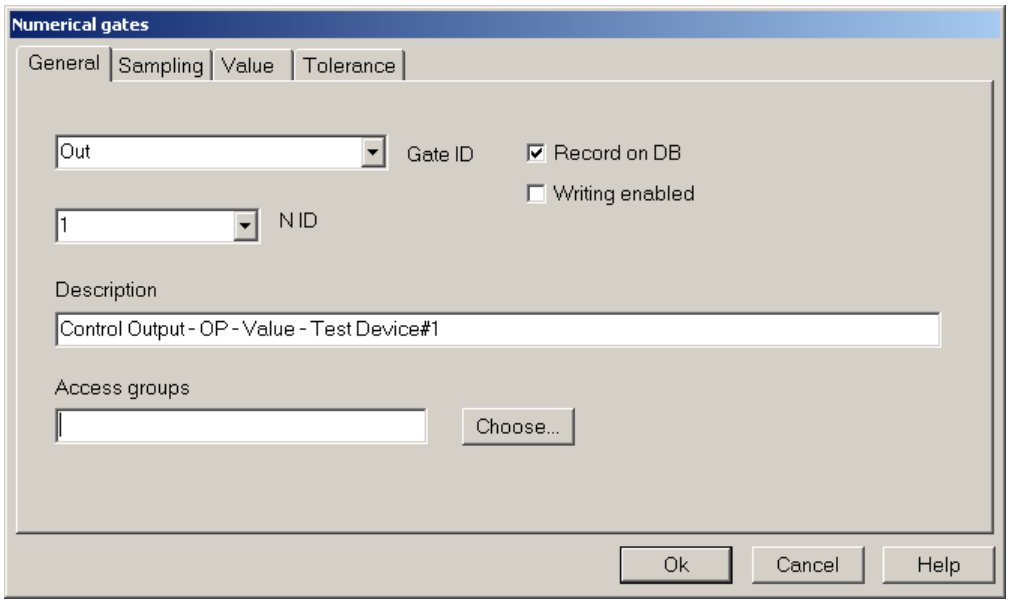


SP numeric variable configuration – Sampling folder

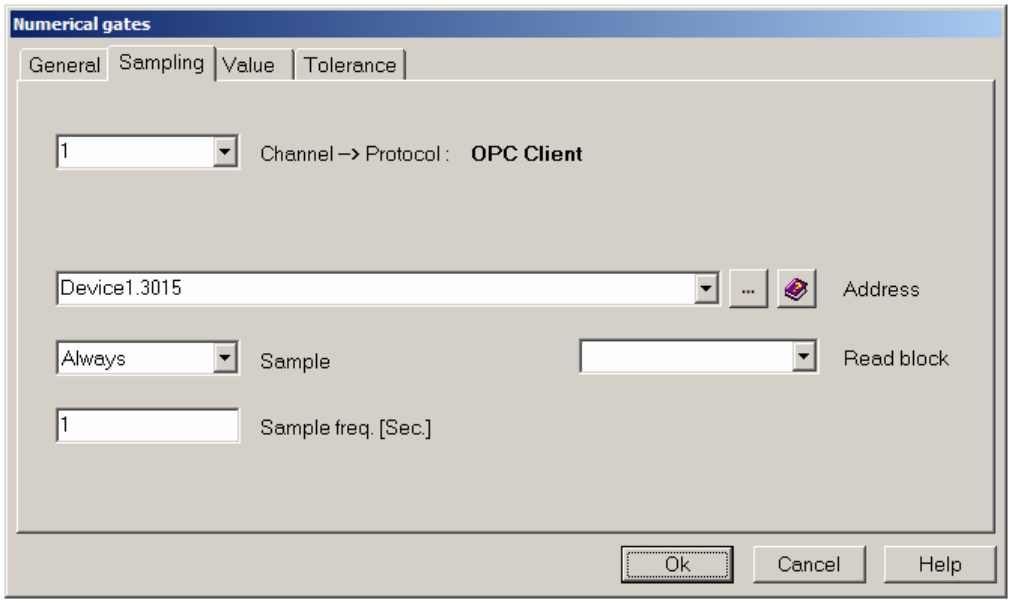


SP numeric variable configuration – Value folder

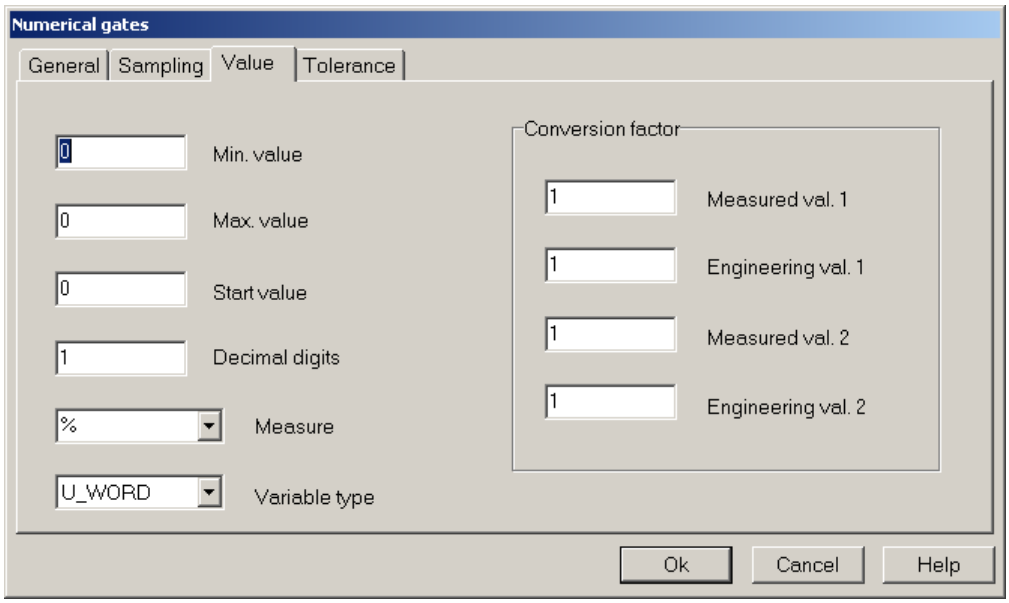
Numeric variable OUT configuration



OUT numeric variable configuration – General *folder*



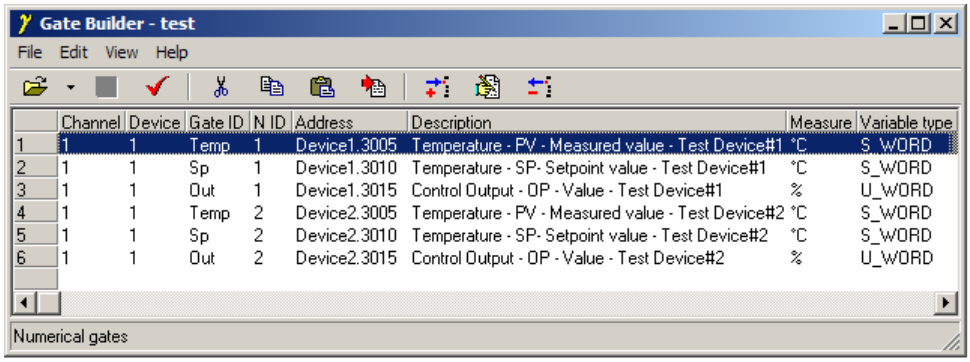
OUT numeric variable configuration –Sampling *folder*



OUT numeric variable configuration – Value folder

End result

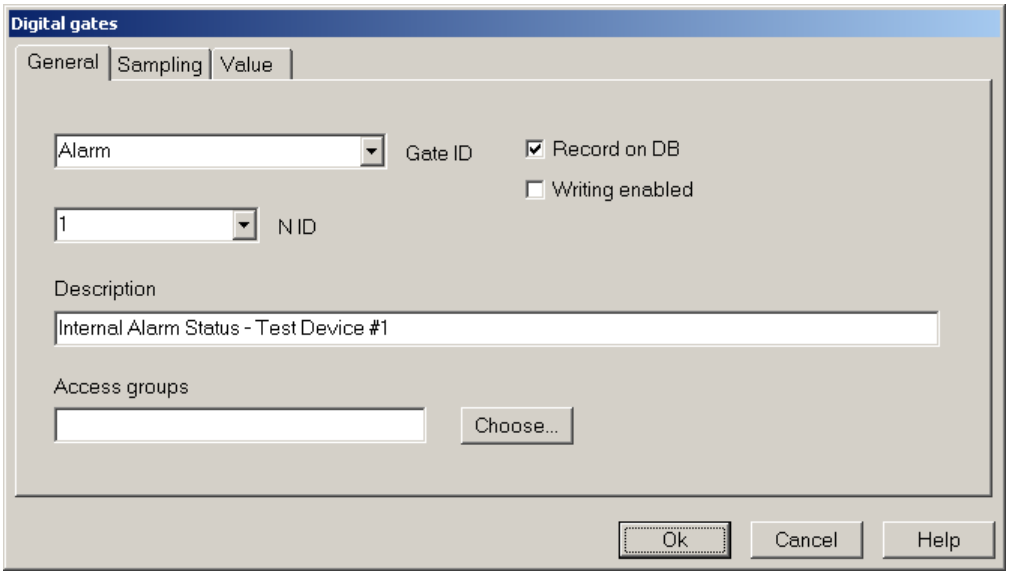
After you have defined all numeric variables, you should see the **Gate Builder** main page similar to the one shown below.



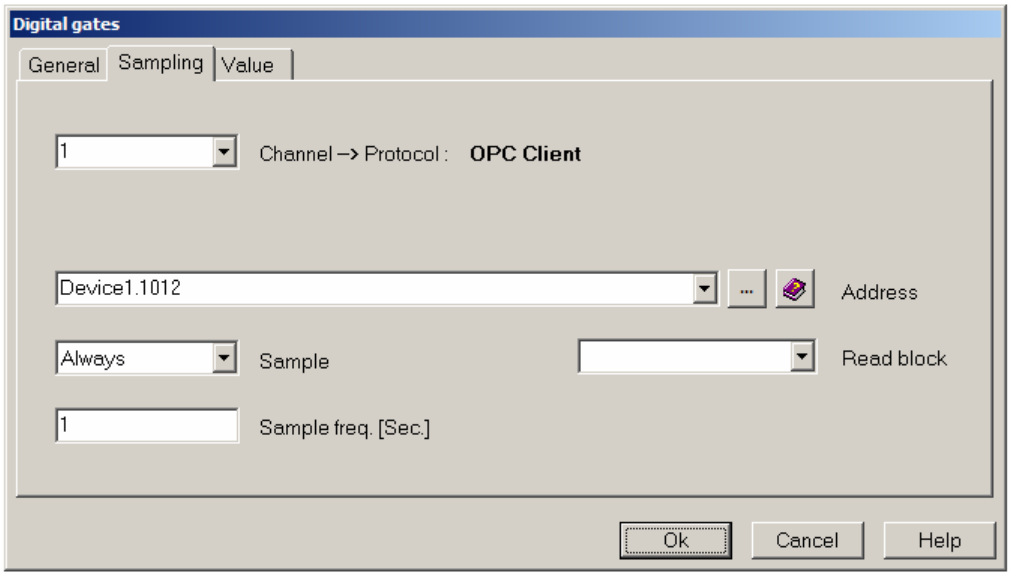
Numeric variable database

5.2 Digital variables configuration

Repeat numeric gates configuration for both devices, having care to change device number (`Device`) and `N ID`.



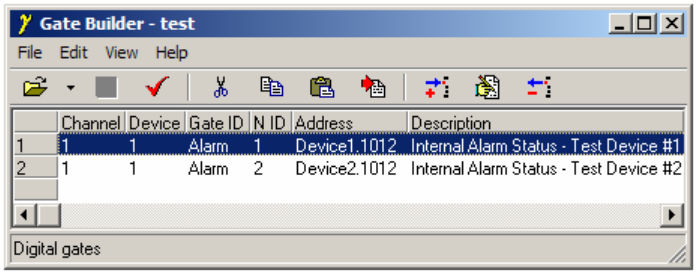
ALARM digital variable configuration – General folder



ALARM digital variable configuration – Sampling folder

Final result

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



Digital variables database

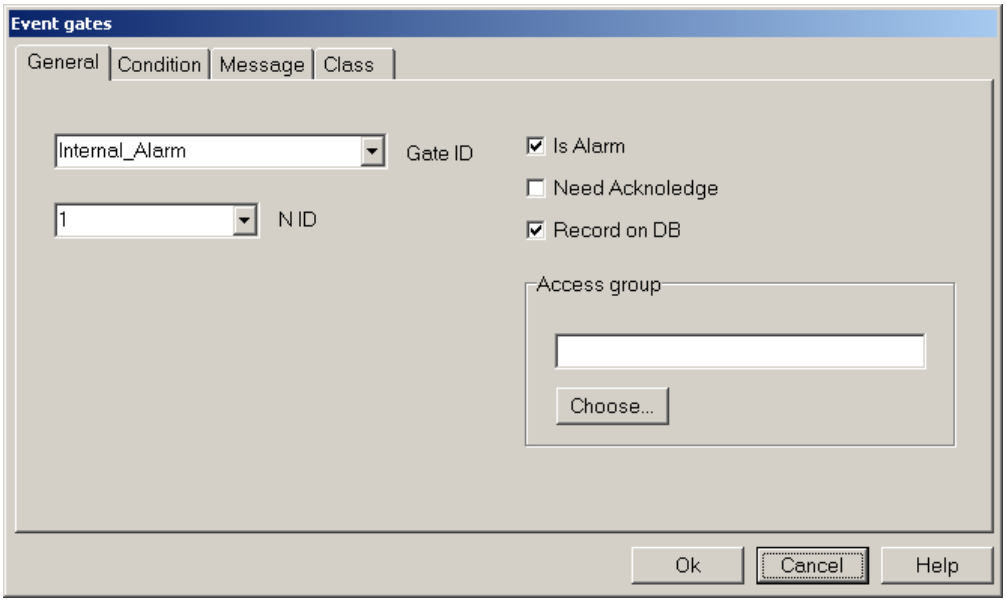
5.3 Alarms gates configuration

So we have created numeric and digital gates database; now we will create as example an event/alarm gate for each device.
These gates are not read from devices but are software generated and their status wil be displayed in runtime as "event and alarm status" and "event and alarm history".

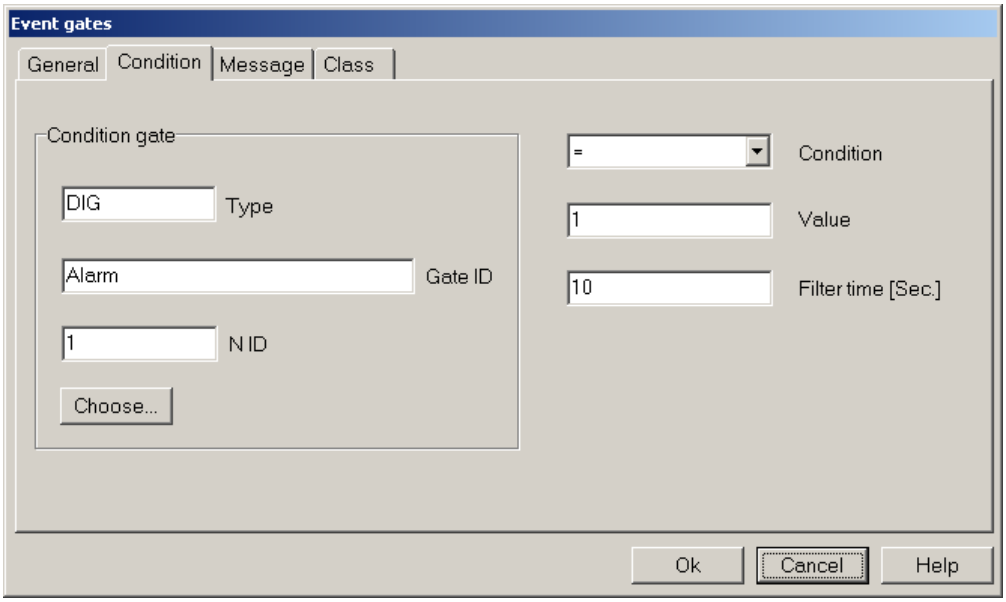
Let's create alarm gates with the following conditions.

Name	Condition	Filter time	Message	Registration
Internal_Alarm,1	Alarm,1 = 1	10 s	Attention! Internal Alarm Test Device#1	yes
Internal_Alarm,2	Alarm,2 = 1	10 s	Attention! Internal Alarm Test Device#2	yes

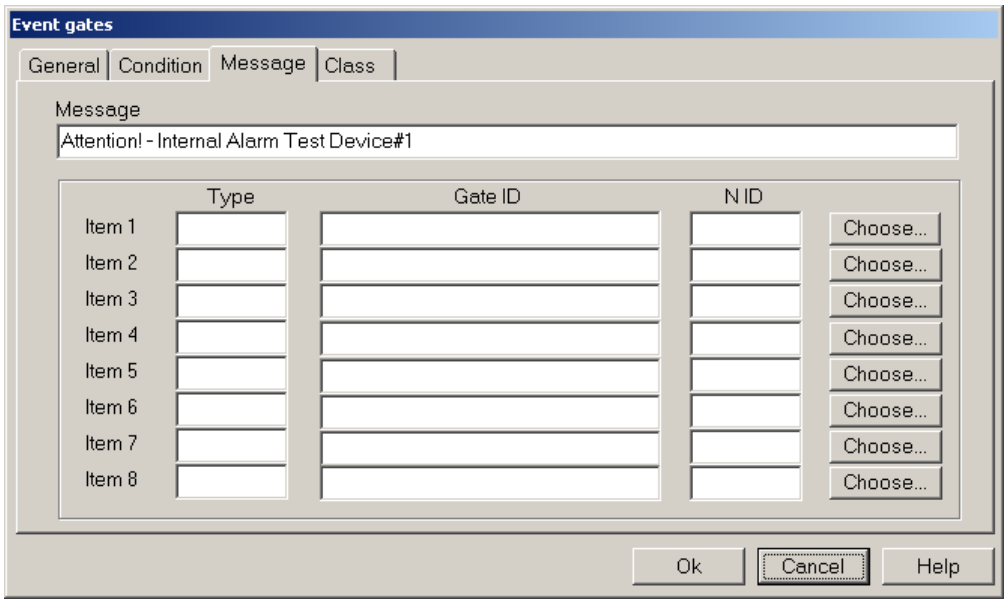
Configuration of alarm gate Internal_Alarm



Internal_Alarm ALARM gate configuration – General folder



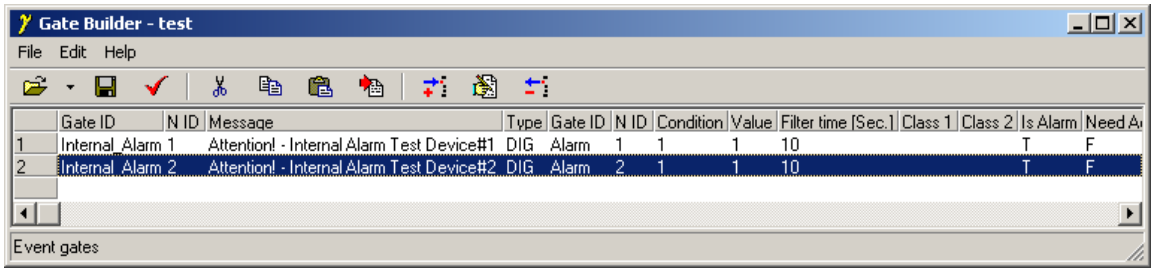
Internal_Alarm ALARM gate configuration – Condition folder



Internal_Alarm ALARM gate configuration –Message folder

Final result

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



Alarm gates database

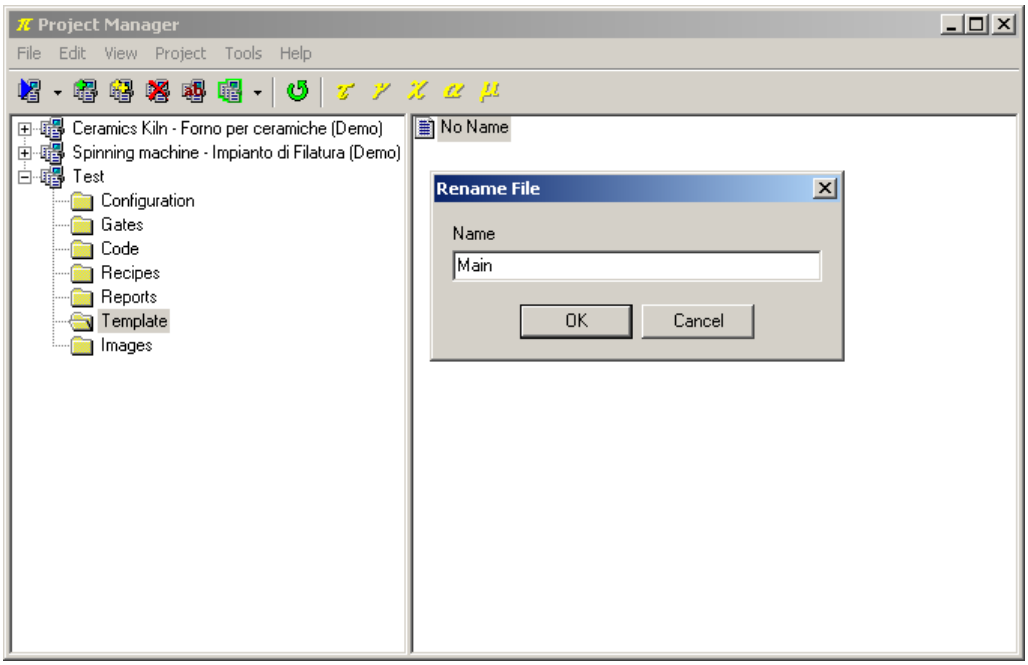
6. Creating a template

Now supervision network has been set; we have defined the logical channel connected to the OPC Client support; we have linked to this channel two devices using the OPC Server (Test Device#1 e Test Device#2); for both we have declared sampling variables and alarm/event internal variables.

si è caratterizzato il canale logico al quale è stato associato il supporto OPC Client; su questo canale tramite OPC Server si sono collegati due dispositivi

Now it is the moment to build a template for the application.

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



Template creating

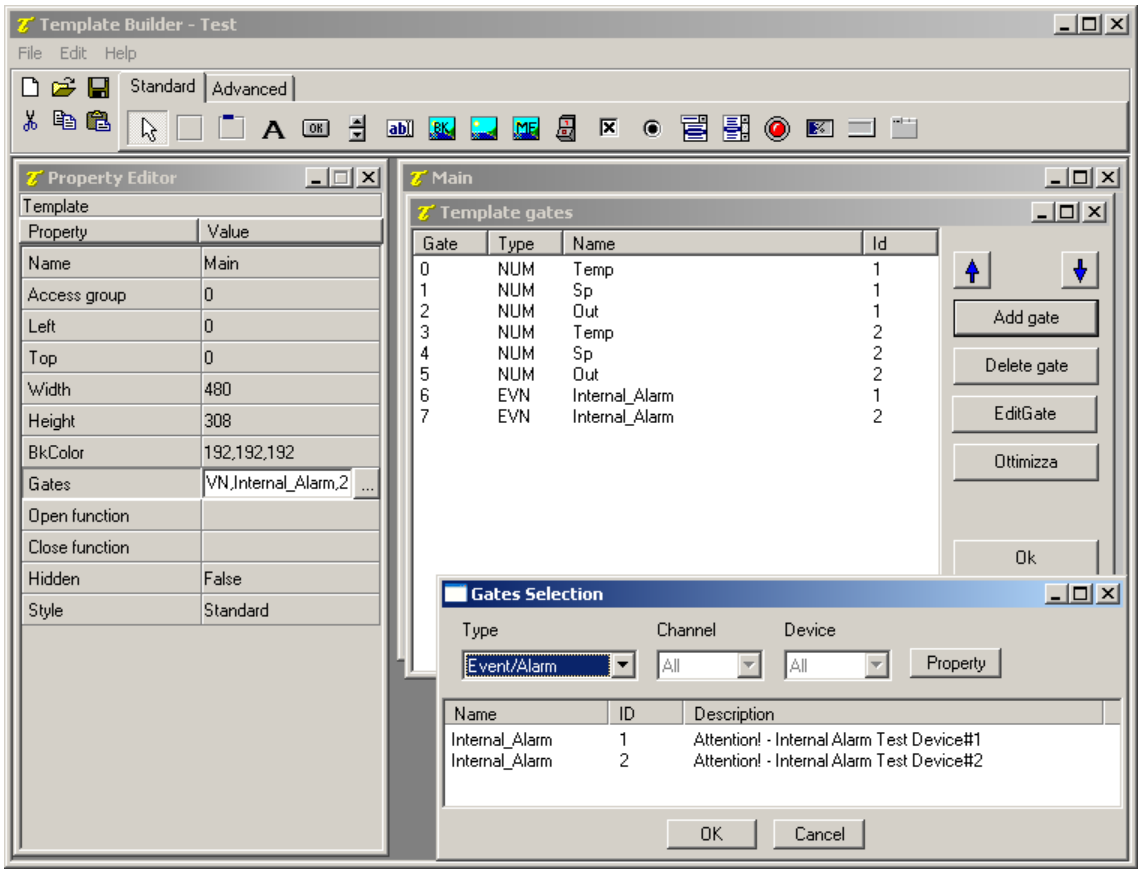
Double-clicking on created template, **Template Builder** start in order to build the graphic page.

6.1 Declaring template variables

First it is necessary to declare which variables we will use in the template; in this example we will use all of them.

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

6.2 Inserting a Label object

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

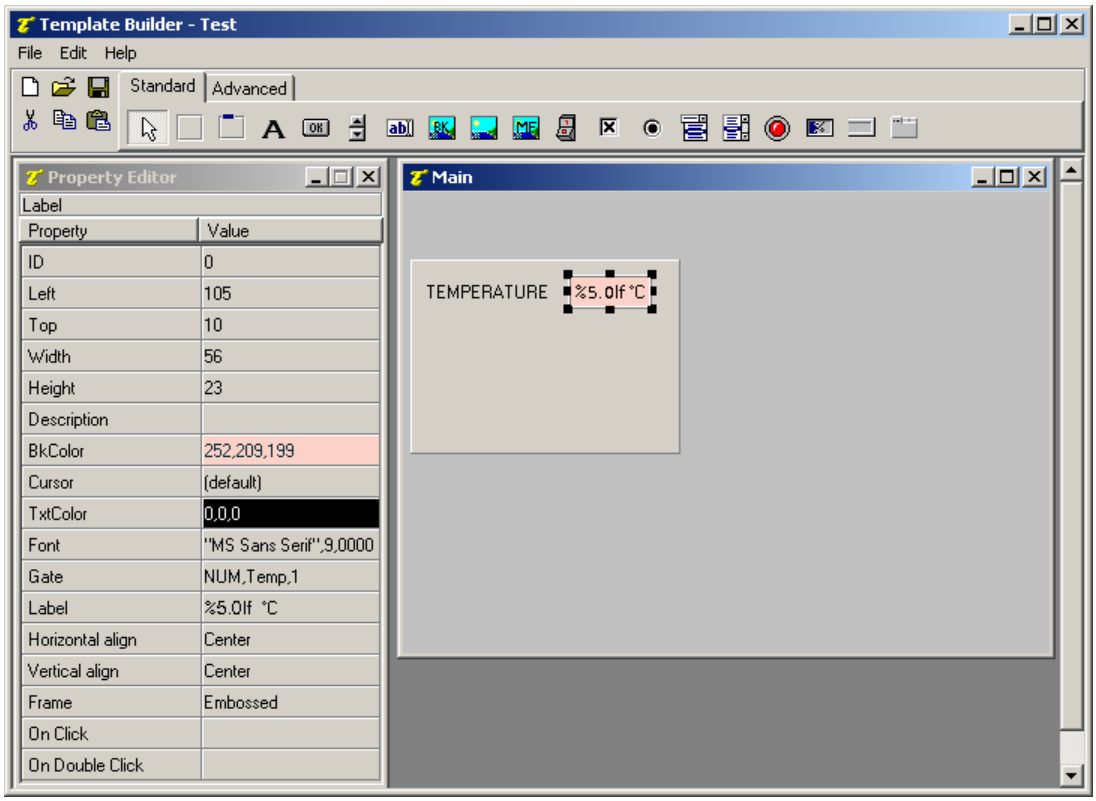
To do this, select *Frame* object among the ones on the upper bar (, it is the first on the left) and click on the template, a void rectangle will be displayed.

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 `TEMPERATURE`.

Alongside of just inserted label, position another one to visualise temperature read form the device.

To link the Label to the numeric variable *TEMP*, click on the button ... alongside of the item `Gate` in Property Editor and select `NUM,Temp,1` among the available gates.
Modify in addition the property `Label` inserting `%5.0lf °C`.

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



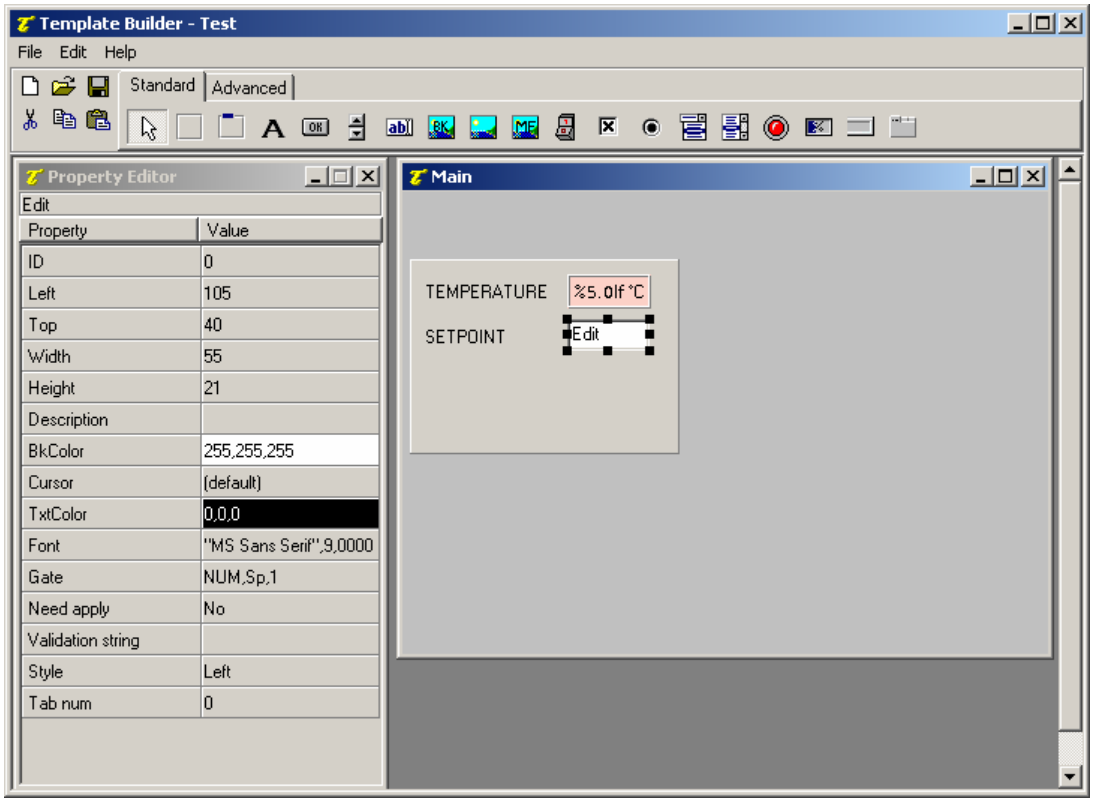
Label object inserting

6.3 Inserting an *Edit* object

Insert another Label, positioning it below `TEMPERATURE` and modify the text in `SETPOINT`

A control will be inserted that will allow to modify the value of the *SP* gate and to send it to the device.

Select *Edit* () object from tool bar; and, as done before, link it to `NUM,Sp,1` gate using the Property Editor.



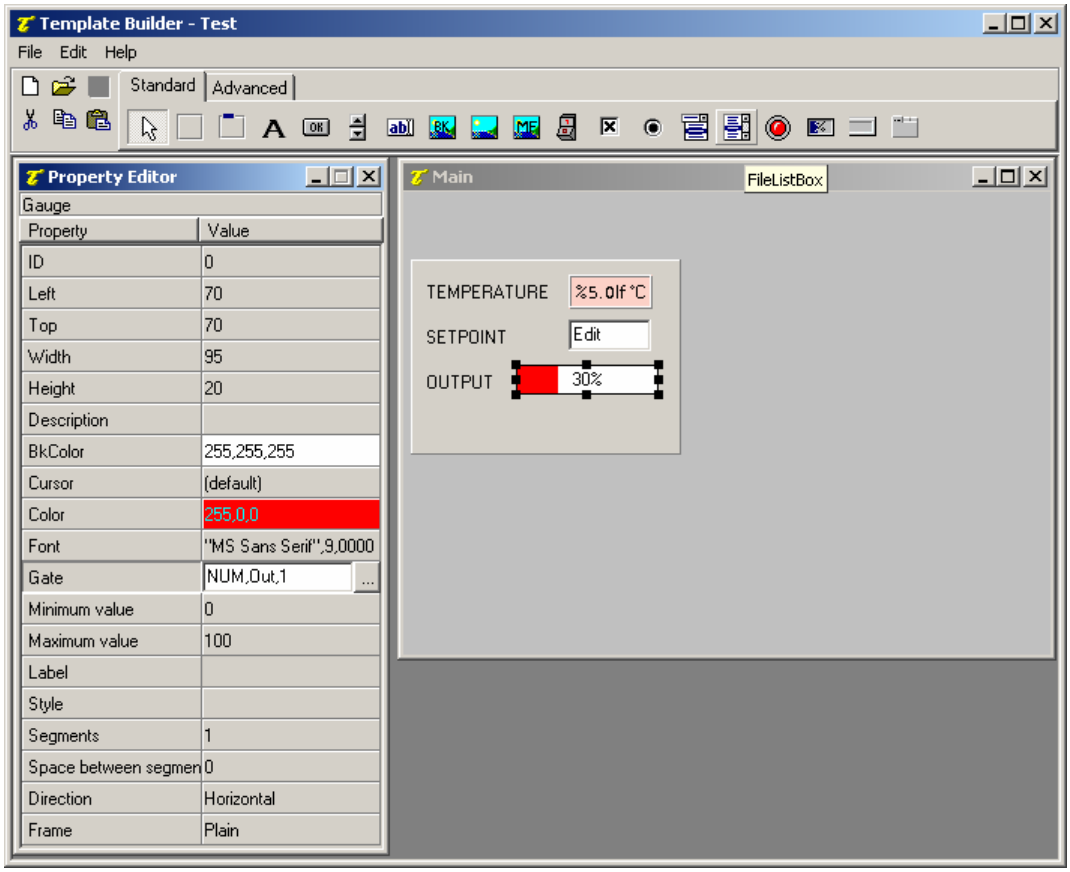
Edit object inserting

6.4 Inserting a Gauge object

Insert another Label, positioning it below `SETPOINT` and modify the text in `OUTPUT`.

Insert now a *Gauge* object ( ) alongside of the previous Label; link it to `NUM,Out,1` gate using the Property Editor.

In this way the value of the device output power will be displayed in bar format.

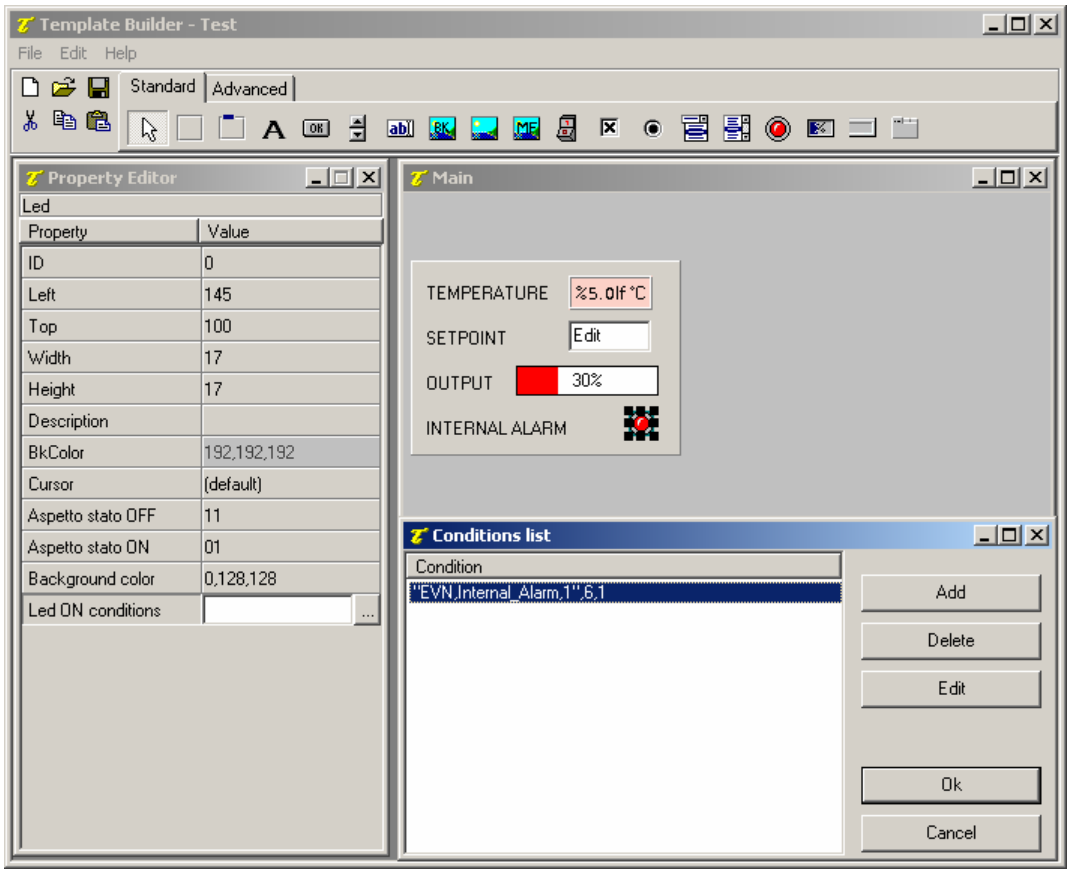


Gauge object inserting

6.5 Inserting a Led object

Insert another Label, positioning it below `OUTPUT` and modify the text in `INTERNAL ALARM`.

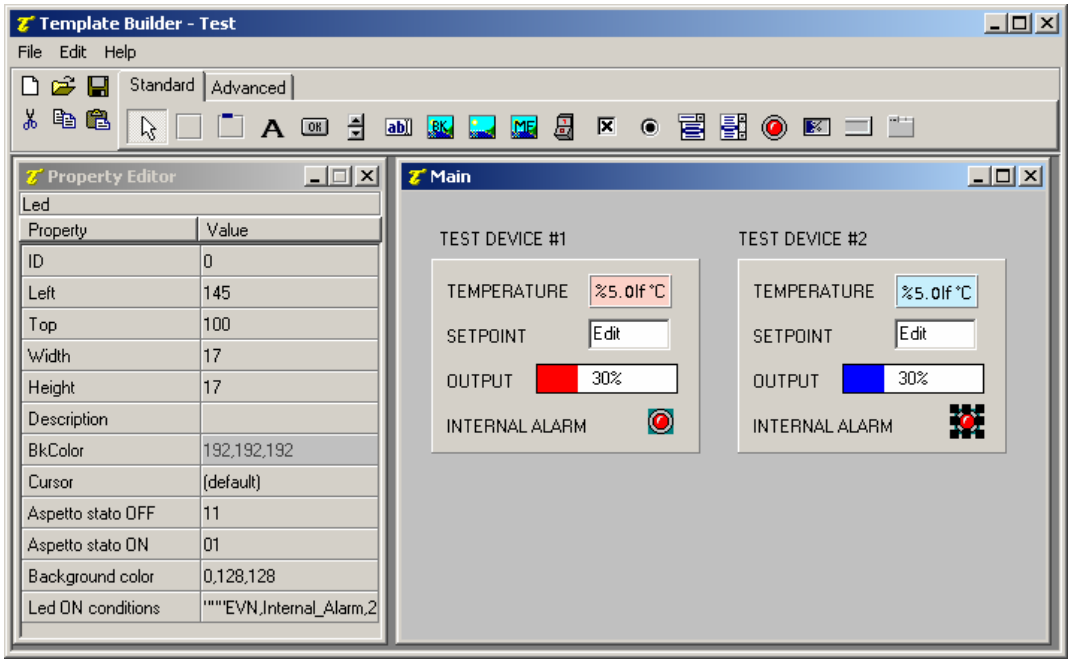
Insert now a *Led* ( ).alongside of the previous Label. To "give animation" to the object it is necessary to specify which is the condition that make it change colour; modify `Led ON conditions` property linking led activation condition to *Internal_Alarm,1* (`Internal_Alarm,1 == true`) alarm activation. A red led will be shown in presence of the alarm, otherwise led will be green.



Led object inserting

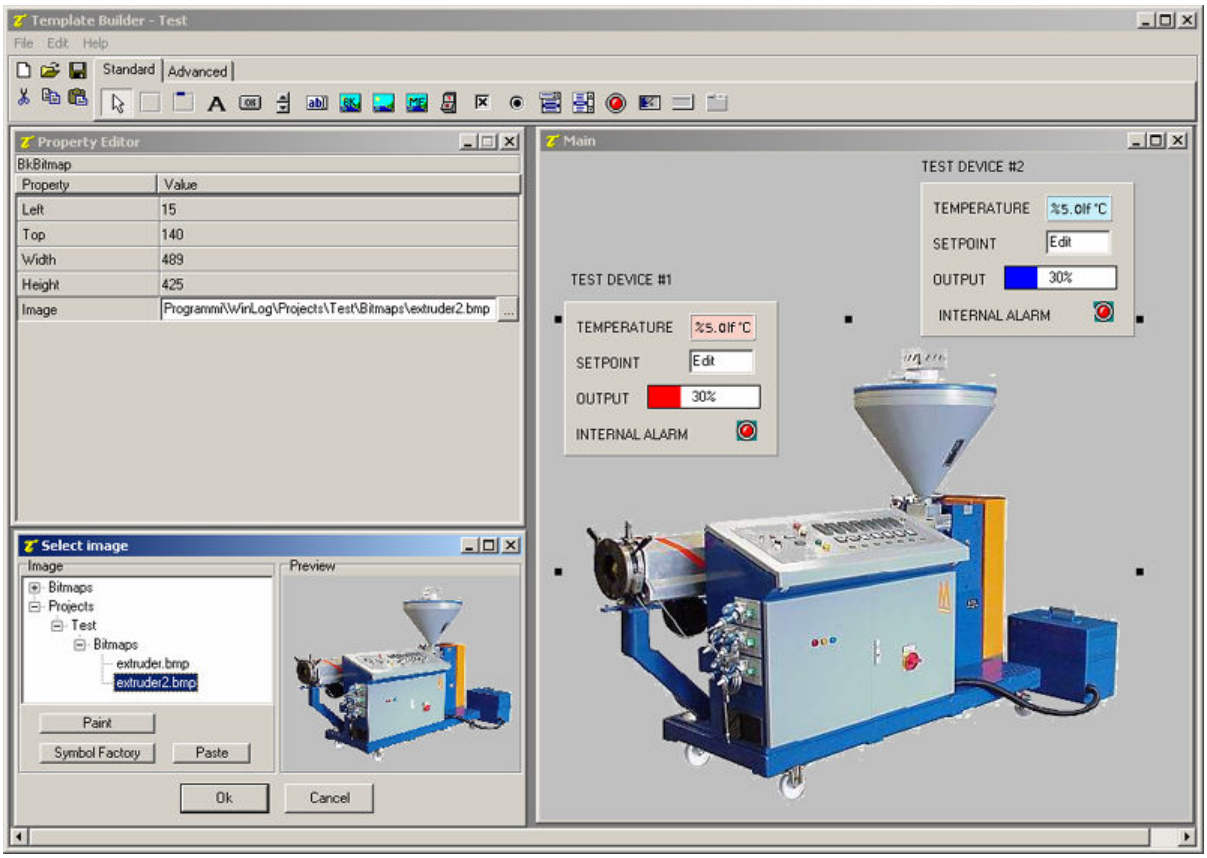
6.6 Completing template

All variables read from device 1 are now displayed; to display also device 2 variables it is enough to select the Frame we have created, copy and paste it in the template. Be careful to not paste it in the source frame; to avoid this mistake click in a free object area of the template before pasting it. Now we have only to modify variables links in Label, Edit, Gauge and Led objects to obtain a supervision interface for the *Test Device #2*.



Two devices supervision template

To complete the template, insert now a *BkBitmap* object (background bitmap, ) previously created using any graphic design software (for example *Paint*) and saved in project `Bitmaps` folder.

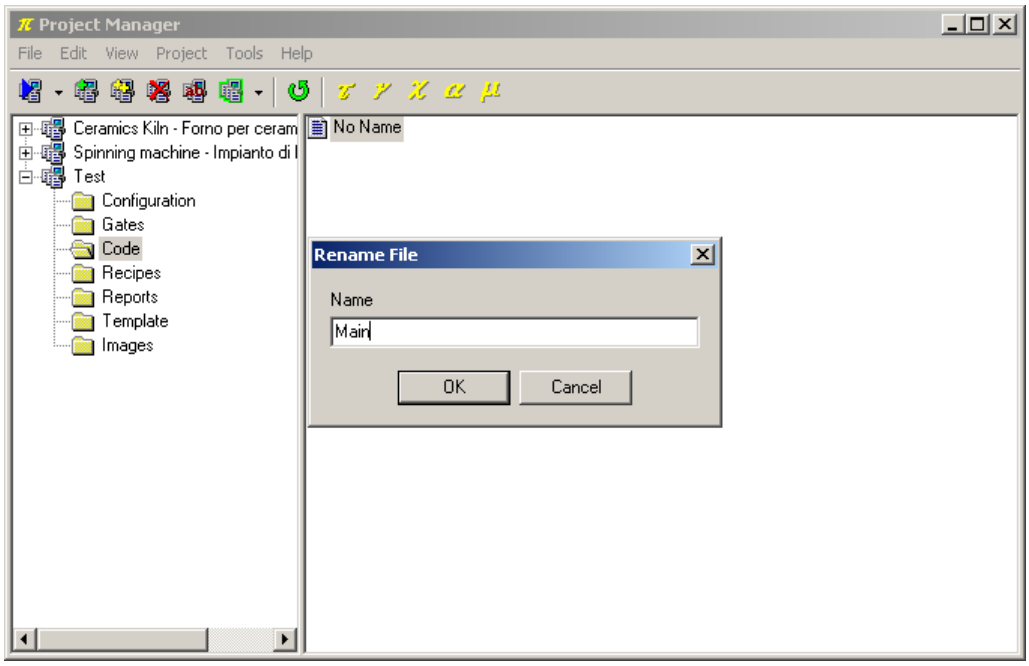


Background bitmap inserting

7. Winlog Pro code example

Now create the code function that allows showing the template at runtime startup.

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



Creazione di un file di codice

Code Builder is the **Winlog Pro** programming environment; we will use it only to define a function that will open the main template at the application startup.

Copy and paste the following code:

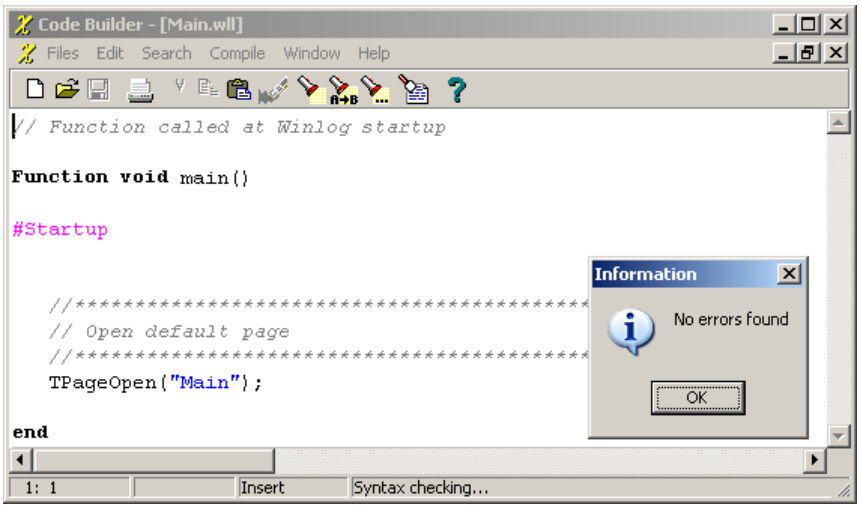
```
// Function called at Winlog startup

Function void Main()
#Startup

    //**************************************
    // Open default page
    //**************************************
    TPageOpen("Main");

end
```

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



Code syntax checking

8. Project execution

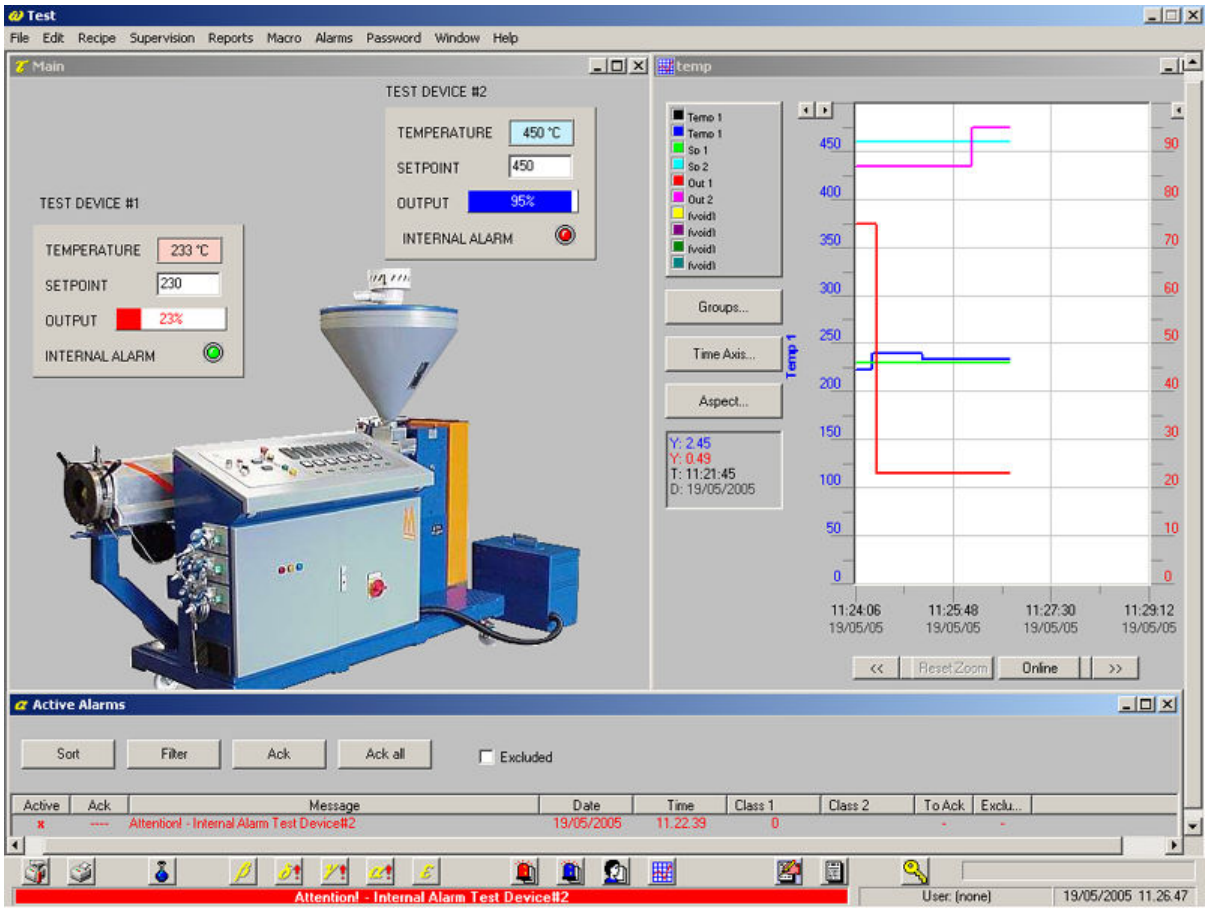
Our example is complete.

Wire 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 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**.



Project execution