Getting started

Creating a simple Siemens PPI protocol application
### Document revision

<table>
<thead>
<tr>
<th>Date</th>
<th>Edition</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>24/11/2009</td>
<td>1.0</td>
<td>-</td>
</tr>
<tr>
<td>24/11/2009</td>
<td>1.0</td>
<td>-</td>
</tr>
</tbody>
</table>
1. Introduction

In this guide we propose, as an example, the design of a simple supervision application Siemens PPI protocol based; 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 (Siemens Simatic S7-200 PLCs).

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.

In our example we have to communicate using Siemens PPI protocol with a device (Test Device#1) whose address is 1 and from which we want to read 3 numeric variables (Temp, Spec Out) and 1 digital variable (Alarm).
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](image)

*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 using Siemens PPI protocol with Siemens S7-200 PLCs.

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, PLC/PPI Address, Time out e Query Pause (for these parameters refer to manufacturer data device, an example is provided in the figure)

Serial channel configuration

Communication between Personal Computer and PLC is provided by using SIEMENS RS232 / PPI Multi - Master Cable - Codice 6ES7 901-3CB30-0XA0 device configured (using dip-switch) in the following way: - PPI / Freeport - Local/DCE - 11 Bit
4. Devices declaration

From elements in Configuration folder select Devices.

Insert Test Device#1 at address 1 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, setpoints, 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, ...).

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).

<table>
<thead>
<tr>
<th>Name</th>
<th>Channel</th>
<th>Device</th>
<th>PPI address*</th>
<th>Variable type</th>
<th>Gate type</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMP</td>
<td>1</td>
<td>1</td>
<td>(VW) Word 5</td>
<td>Signed Word</td>
<td>Numeric</td>
<td>°C</td>
<td>Temperature - Measure</td>
</tr>
<tr>
<td>SP</td>
<td>1</td>
<td>1</td>
<td>(VW) Word 10</td>
<td>Signed Word</td>
<td>Numeric</td>
<td>°C</td>
<td>Temperature - Setpoint</td>
</tr>
<tr>
<td>OUT</td>
<td>1</td>
<td>1</td>
<td>(VW) Word 15</td>
<td>Unsigned Word</td>
<td>Numeric</td>
<td>%</td>
<td>Control Output - Value</td>
</tr>
<tr>
<td>ALARM</td>
<td>1</td>
<td>1</td>
<td>(VB) Byte 1 Bit 1</td>
<td>Bit</td>
<td>Digita</td>
<td></td>
<td>Internal alarm status</td>
</tr>
</tbody>
</table>

- The code between brackets represents the PPI function used to read the variable (refer to protocol manual in Project Manager Help).
5.1 Numeric variables configuration

Repeat numeric gates configuration (PLC Siemens S7200 V area).

**Numeric variable TEMP configuration**

TEMP numeric variable configuration – General folder

TEMP numeric variable configuration – Sampling folder
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**
After you have defined all numeric variables, you should see the Gate Builder main page similar to the one shown below.
5.2 Digital variables configuration

Repeat numeric gates configuration for both devices (PLC Siemens S7200 V area.

**ALARM digital variable configuration – General folder**

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

**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 will be displayed in runtime as "event and alarm status" and "event and alarm history".

Let's create alarm gates with the following conditions.

<table>
<thead>
<tr>
<th>Name</th>
<th>Condition</th>
<th>Filter time</th>
<th>Message</th>
<th>Registration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal_Alarm,1</td>
<td>Alarm,1 = 1</td>
<td>10 s</td>
<td>Attention! Internal Alarm Test Device#1</td>
<td>yes</td>
</tr>
</tbody>
</table>

*Configuration of alarm gate Internal_Alarm*

![Internal_Alarm ALARM gate configuration – General folder](image1)

![Internal_Alarm ALARM gate configuration – Condition folder](image2)
**Final result**

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

Now supervision network has been set; we have defined the logical channel and its link to PC COM port and we have connected it to Siemens PPI protocol; we have linked to this channel a devices (Test Device#1); we have declared sampling variables and alarm/event internal variables.

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.

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.
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 from 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.
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 gate NUM, Sp, 1 using the Property Editor.
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](image)

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.
6.6 Completing template

All variables read from device 1 are now displayed.

To complete the template, insert now a BkBitmap object (background bitmap, \[\text{Background bitmap}\]) 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.

Code file creating

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:

```cpp
// 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

```c
// Function called at Minlog startup

function void main()
    #Startup
        // Open default page
        //personaOpen("Main");
    end
```

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.