

The module <WebCfgD> of subsystems “User Interfaces”

<i>Module:</i>	WebCfgD
<i>Name:</i>	Dynamic Web configurator
<i>Type:</i>	User Interfaces
<i>Source:</i>	ui_WebCfgD.so
<i>Version:</i>	0.8.0
<i>Author:</i>	Roman Savochenko
<i>Translated:</i>	Maxim Lysenko
<i>Description:</i>	Provides dynamic WEB based configurator. Uses XHTML, CSS and JavaScript technology.
<i>License:</i>	GPL

Contents table

The module <WebCfgD> of subsystems “User Interfaces”	1
Introduction	2
1. Configuration	4
2. Basic elements	5
3. Commands	6
4. Lists	7
5. Tables	8
6. Images	9
7. Errors	10

Introduction

The "WebCfgD" module provides the configurator of OpenSCADA system. Configurator is implemented on the basis of Web-technologies:

- *HTTP* — hypertext transfer protocol;
- *XHTML* — extended language of markup of the hypertext documents;
- *CSS* — cascading style sheets of hypertext documents;
- *JavaScript* — built-in into the hypertext document the browser programming language;
- *DOM* — document object model of the internal structure of the browser;
- *AJAX* — arrangement of asynchronous and synchronous requests from the JavaScript to the server;
- *XML* — eXtensible Markup Language.

Interface of the configurator is formed in the WEB-browser by reference to the WEB-server and getting from it the XHTML-document over HTTP. In this case there is the OpenSCADA system in the role of the WEB-server, which supports standard communication mechanisms of TCP-networks (module Transport.Sockets), hypertext transfer protocol (module Protocol.HTTP), as well as encryption of traffic between the browser and the server (Transport.SSL). Based on this to gain access to the interface configuration of the OpenSCADA, provided by this module, you need to configure the transport in the OpenSCADA (Transport.Sockets or Transport.SSL) in conjunction with the protocol HTTP (Protocol.HTTP). In the delivery of the OpenSCADA system there are configuration files containing settings of Transport.Sockets for ports 10002 and 10004. Hence the interface of the module in the configuration of the OpenSCADA by default will be available at URL: <http://localhost:10002> or <http://localhost:10004>.

After receiving the document XHTML the JavaScript program runs to create dynamic interface configurator.

At the core of the module there is the language of the management interface of the OpenSCADA system, and thus provides the uniform interface of configuration. Update of module may be required only in the case of updating the specification of the language of management.

The module was implemented and tested on three WEB-browsers, representatives of the three types of WEB-engines, as follows:

- Mozilla Firefox 3.0.4
- Opera 9.6.2
- Konqueror 3.5.10

Using the module starts with the opening of the session, the user authentication by the module of the protocol HTTP (Protocol.HTTP). For the operation of the authentication and the mechanism of saving of the session the browser must allow Cookies.

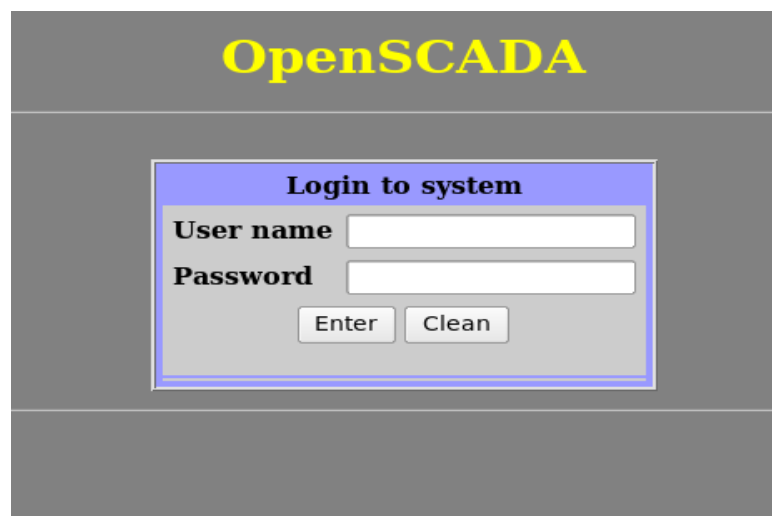


Fig.1. User authentication.

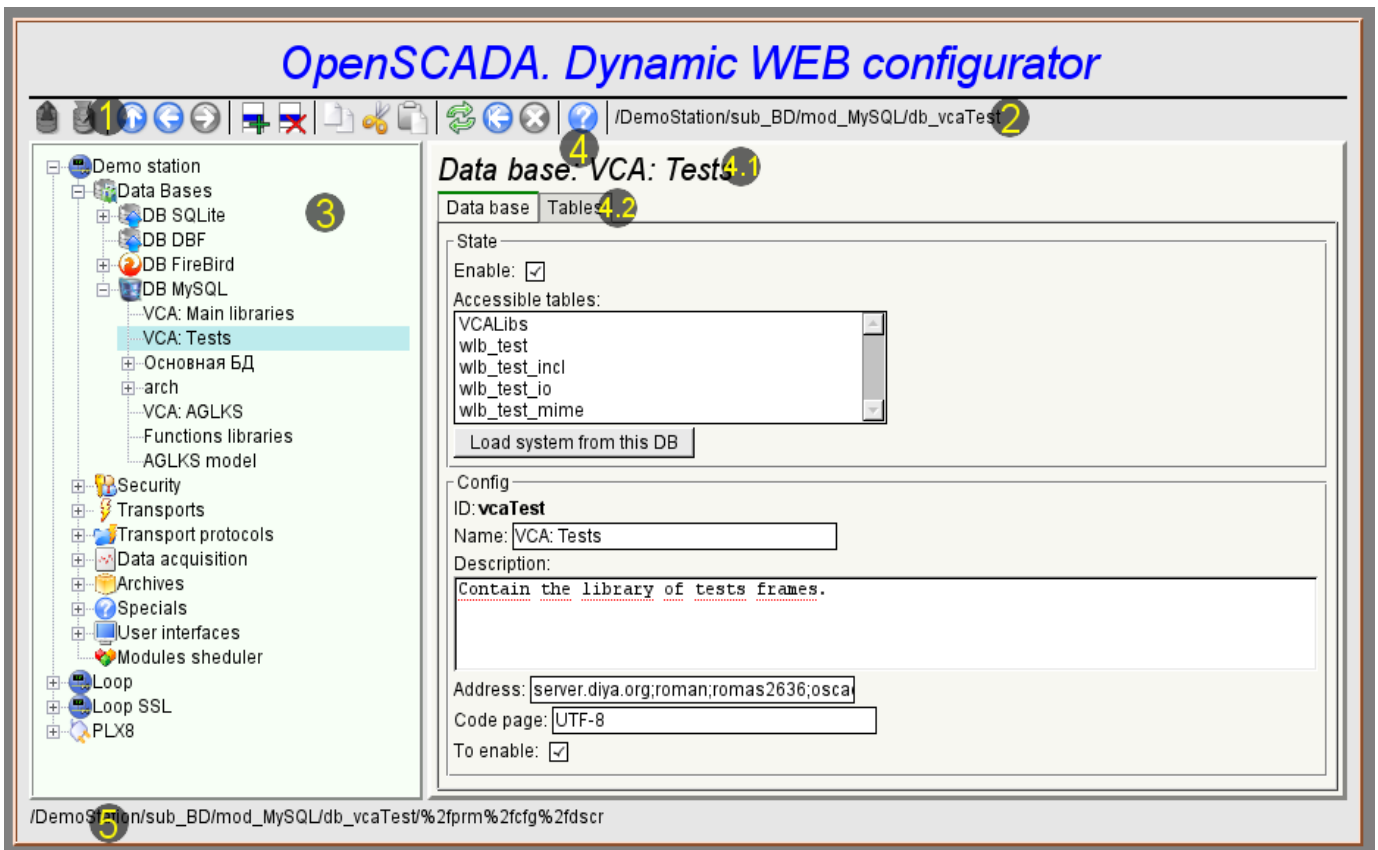


Fig.2. Working window of the configurator

Lets examine the working window of the configurator in Fig. 2.

Working window of the configurator consists of the following parts:

- 1 *Toolbar* — contains the control buttons.
- 2 *Address of the open node* — displays the current selected node.
- 3 *Navigator* — intended for direct navigation through the control tree.
- 4 *Working field* — divided into parts:
 - 4.1 *The name of the node* — contains the name of the current node.
 - 4.2 *Tabulator of the working areas* — the root pages (control areas) of the node are placed into the tabulator. The control areas of the following levels are placed on the information panel.
- 5 *Status line* — displaying the states of the configurator.

The toolbar contains the following control buttons (from left to right):

- *Load* — downloads the selected object or branch of object from the database.
- *Save* — save the selected object or branch of object to the database.
- *Up* — climb up the tree.
- *Previous* — open the previous page.
- *Next* — open the following page.
- *Add item* — add a new object to the container.
- *Delete item* — delete the selected object.
- *Copy item* — copy the selected object.
- *Cut item* — cut of the selected object. The original object is removed after paste.
- *Paste item* — paste of the copied or cut item.
- *Refresh item and the tree* — refresh the current page.
- *Start periodic update* — run periodically update of the contents of the current page with an interval of 5 second.
- *Stop periodic update* — stop periodically update od the contents of the current page with an interval of one second.
- *About* — the information about the module.

The control tools are divided into basic, commands, lists, tables and images. All items are displayed in the sequence strictly appropriate to their location in the description of language of management interface.

1. Configuration

To adjust your own behavior in the not obvious situations module provides the ability to customize individual settings through the management interface of the OpenSCADA (Fig. 3). These parameters are:

- The lifetime of the authentication session (min) — points during which time interval of user inactivity his session will be saved.
- The link to the configuration page of the external OpenSCADA stations used to enable remote configuration.

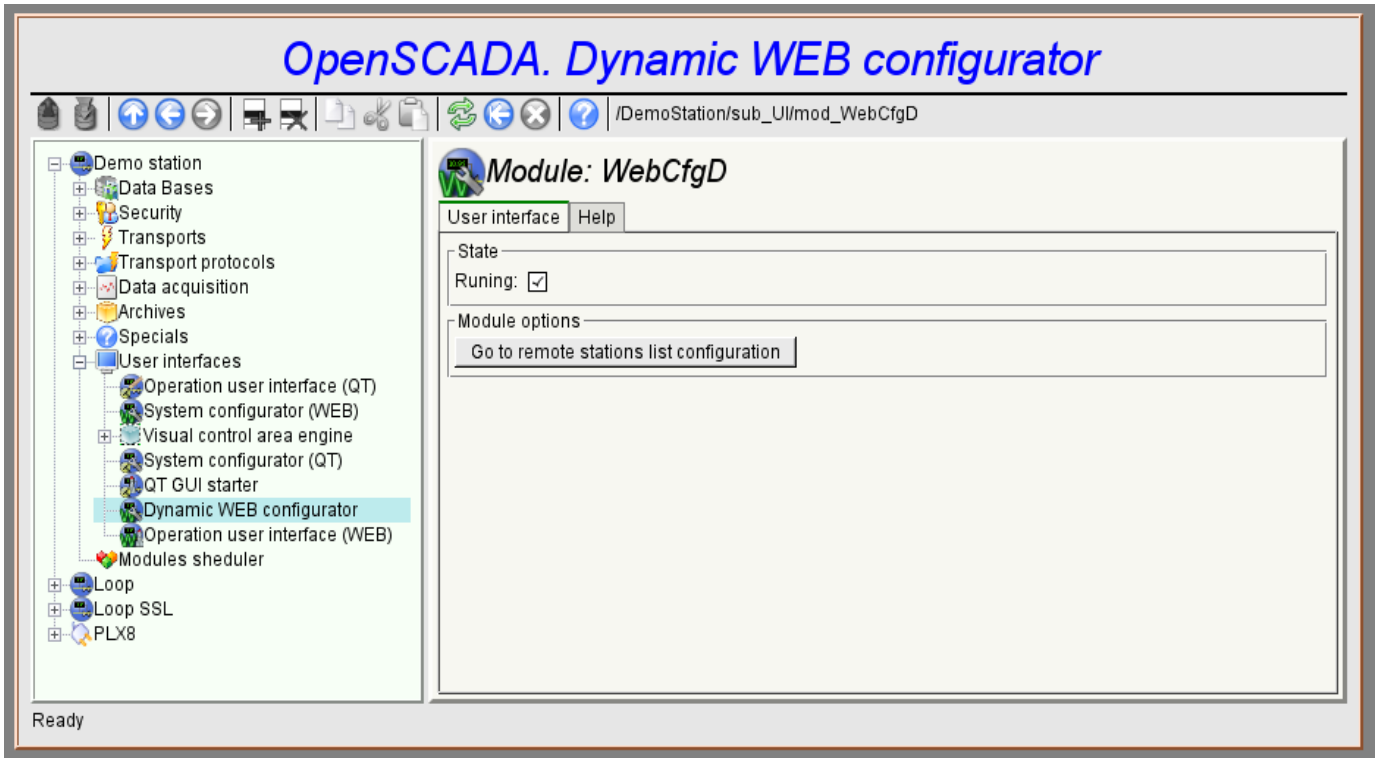


Fig.3. The configuration page of the configurator.

3. Commands

Commands are the elements for the transfer of the certain instructions of the action to the node and for the organization of the links on the page. Commands may contain parameters. The parameters are formed from the basic elements. Example of the commands with the parameters is shown in Fig.5.

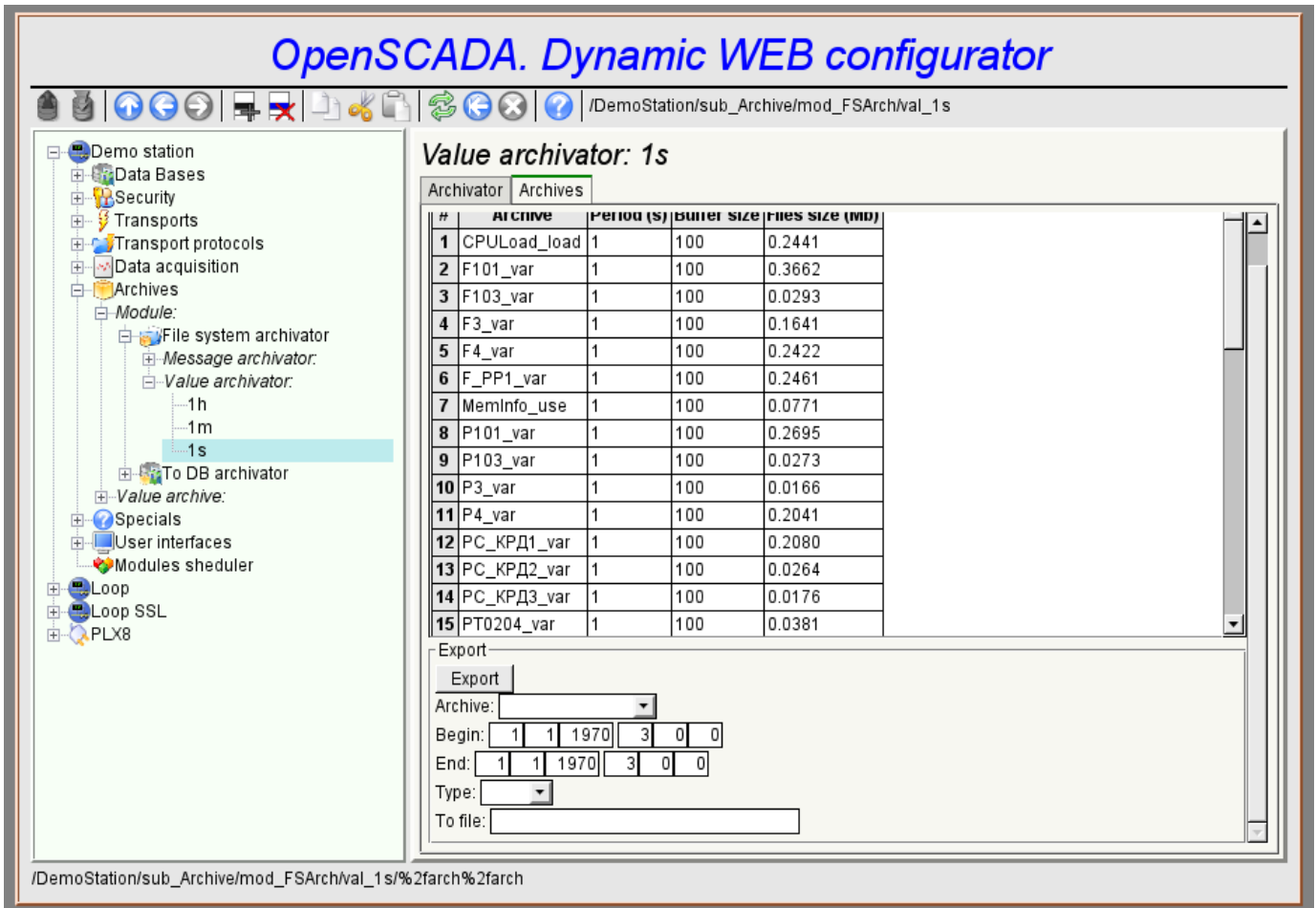


Fig.5. The command.

4. Lists

Lists contain a group of basic elements of the same type. Operations under the elements are accessible via the context menu by the mouse click on the list. Through the elements of the list can be performed the moving operations to other pages. Lists can be indexed. An example of the list is shown in Fig. 6.

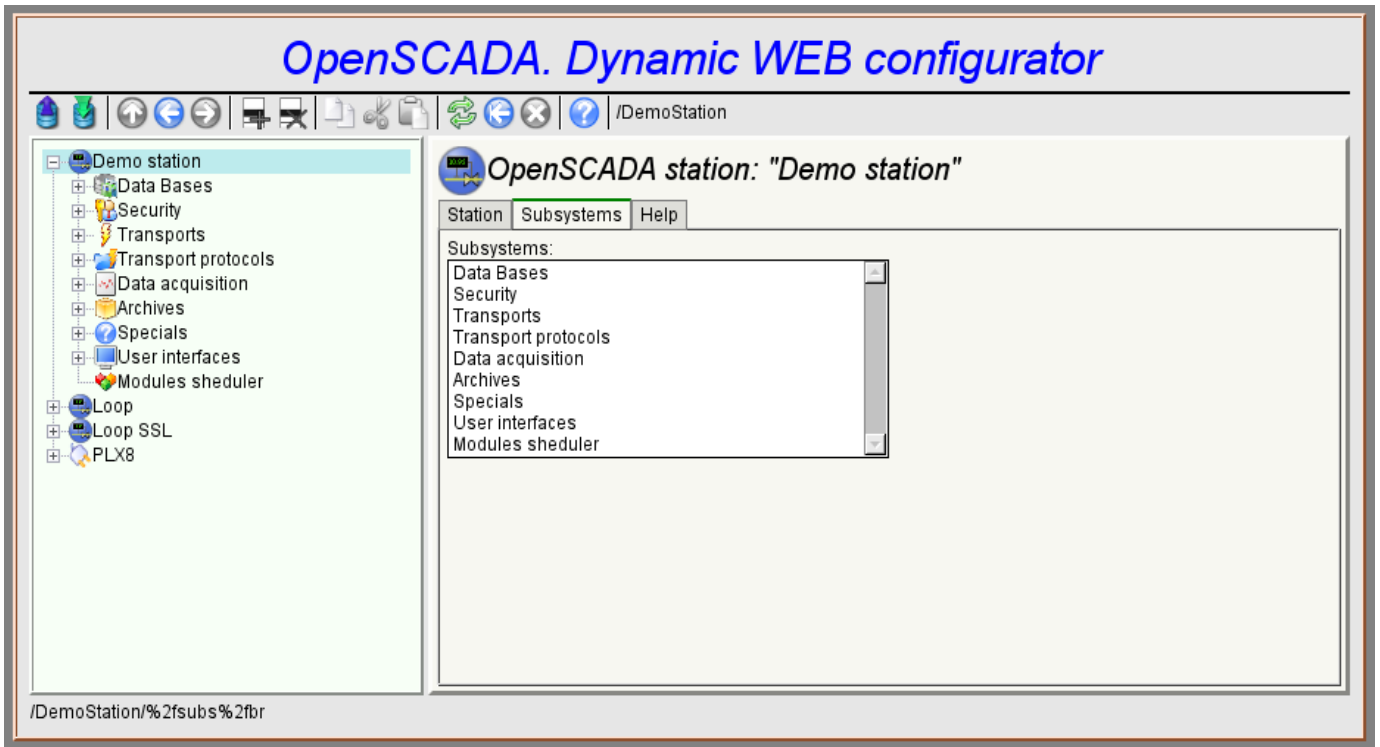


Fig.6. The list.

5. Tables

The tables contain values of the basic elements. Type of the basic element is an individual for each column. Example of the table is given in Fig. 7. Operations on the structure of the table for editable tables are accessible through the context menu by the mouse clicking on the service button with the line number. Editing of the table is done by double-clicking on the desired cell.

The screenshot shows the OpenSCADA Dynamic WEB configurator interface. The title bar reads "OpenSCADA. Dynamic WEB configurator". The address bar shows the path: "/DemoStation/sub_DQA/mod_JavaLikeCalc/lib_techApp/fnc_diafragma".

On the left is a tree view of the project structure. The "Diaphragm" function is selected and highlighted in blue.

The main window displays the "Function: Diaphragm" configuration. It has three tabs: "Function", "Program", and "Execute". The "Function" tab is active, showing an IO table.

#	Id	Name	Type	Mode	Hide	Default
1	Fi	Input flow (tones/h)	Real	Output	Off	0
2	Pi	Input pressure (ata)	Real	Input	Off	1
3	Fo	Output flow (tones/h)	Real	Input	Off	0
4	Po	Output pressure (ata)	Real	Output	Off	1
5	dP	Pressure differential (kPa)	Real	Output	Off	0
6	Sdf	Diaphragm cutset (m2)	Real	Input	Off	0.1
7	So	Output pipe cutset (m2)	Real	Input	Off	0.2
8	lo	Output pipe length (m)	Real	Input	Off	10
9	Q0	Norm density of environs (kg/m3)	Real	Input	Off	1
10	f_frq	Calc frequency (Hz)	Real	Input	On	100
11	Pot	Output pressure laged	Real	Output	On	1
12	Fit	Input flow laged	Real	Output	On	0

Below the table is the "Programm:" section containing the following code:

```
Pot+=(Po-Pot)/(0.005*lo*f_frq);
Qr=Q0*Pi+0.01;
Fi=4e3*Sdf*sigm(Pi-Pot)*pow(Q0*abs(pow(Pi,2)-pow(Pot,2))/293,0.5);
Fit+=(Fi-Fit)/(0.005*lo*f_frq);
Po+=0.27*(Fit-Fo)/(So*lo*Q0*f_frq);
Po=(Po<0)?0:(Po>200)?200:Po;
dP--=(dP-100.*(Pi-Po))/f_frq;
```

The status bar at the bottom shows the path: "/DemoStation/sub_DQA/mod_JavaLikeCalc/lib_techApp/fnc_diafragma/%2fio%2fio".

Fig.7. The table.

6. Images

The images are designed to transmit graphic information into the configurators. Example of the image is shown in Fig. 8.

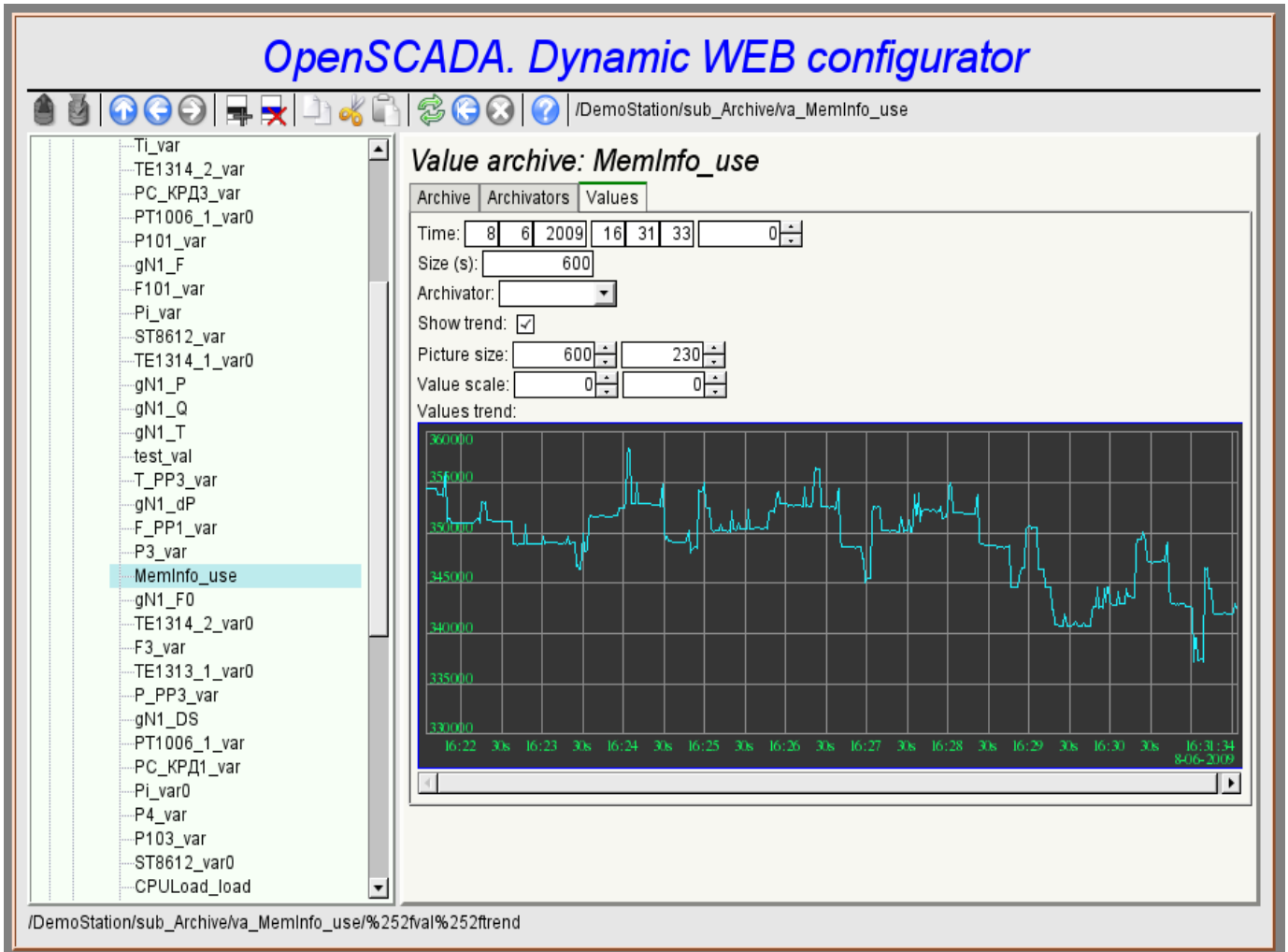


Fig.8. The image.

7. Errors

Performance of the configurator may differ for different types of browsers. This is due to the fact that the basis of this module is quite a lot of complex technologies, as well as differences between them on different types of WEB-engines.

In addition, each Web-browser has its own problems. Some errors were outflanked in the process of implementation, but part of them has stayed in sight of the significant difficulties in their outflanking and also of actual impossibility to do so.

This section contains a table listing the detected errors of the WEB-browsers, which are appeared in the configurator.

Error	Description	Correction
<i>Mozilla FireFox 3.0.4</i> (stable, few errors)		
Offset of the popup window of the editable combobox at 5 pixels up and left.	The problem lies in the fact that the calculation of the absolute position of the element of the document doesn't grab exactly 5 pixels. Error of 5 pixels is visible in relation to the coordinates of the mouse pointer and the position of the newly-created entirely-positioned window. The algorithm for computing the position: <i>for(; obj != null; obj = obj.offsetParent) posX += obj.offsetLeft;</i>	To correction of this error to the estimated value on this browser the 5 pixels are added.
In the element of the list (<select size="10"/>) the vertical scroller is always shows and never turned on the horizontal one.	This element is actively used for building the context menu and drop-down list of the editable combo-box.	To outflank the browser's error I had to include the list in the block with scroll of the block itself.
The image field is not updating.	In order to eliminate the need for restructuring of the configurable page while updating the values of fields in the tree of the structure objects of the pages which was get from the XMLHttpRequest, the properties are created with the links to the object of the tags of the fields (addr_lab, addr_val_w). In objects with the name of the tag "img", these properties are not created by the browser.	The problem is not solved.
<i>Opera</i> (stable, few errors)		
Scroller of the unit of the page does not turn on. For example when displaying large images of the trend.	The block is fixed with the parameters { overflow: auto; width: 600px; } however, in the case of the exceeding the size of the interior elements the scroller is not turned on.	The problem is not solved.
<i>Konqueror</i> (very unstable on the dynamic resources and contains many errors)		
Stable browser crashing.	Browser repeatedly and consistently crashes while the computation of JavaScript and when dealing with external windows.	The problem is not solved.
The skroll of the navigation tree doesn't returned.	If the navigation tree to expand until the vertical scroll is appeared, then scroll down it, then fold a large branch, the vertical scroll disappears, and a part of the tree remains invisible in the upper part of the block. Ie the contents of the block is not updated.	The problem is not solved.

Error	Description	Correction
The images do not update.	In the fields of images, to update the image from the server, the property "src" is to be changed. Browser does not feel it, or even updates the size of the frame, and the image is not updated. Methods to prevent caching of the images are used, but do not help.	The problem is not solved.
Capture of the images of the buttons	JavaScript modules use non asynchronous, but synchronous requests to a server to save the sequence of actions. In moments of such a request if it was caused by the event from the image (the image is a button), the image is captured as if to move, even for short mouse clicks.	The problem is not solved.
Impossible to insert a new element to the tree of objects obtained as a result of XMLHttpRequest	To monitor the modification of the configuration page the current tree structure to a new one, just received from XMLHttpRequest, reduction was used. When inserting a new element to the tree of the structure through the insertBefore() an error occurs "DOM error 4". If the paste is made to the tree created from zero (not from XMLHttpRequest), this error does not occur. It seems the problem lies in the contrast of object "document" — the owner of these trees. In such a tree it is not possible to add a node as document.createElement (). Only created as a mytree.ownerDocument.createElement() are inserted.	The procedure for verifying the structure was simplified to reduced to the determining the fact of changes.
The mechanisms of the formation of the context menu in the Konqueror 4 doesn't work.	Typically, to form a context menu handler oncontextmenu is used in Firefox and IE, or onmousedown in processing the right keys on the remaining browsers. In the Konqueror generally oncontextmenu does not work, but onmousedown only works in Konqueror 3.5.	The problem is not solved.