The module of subsystem "Data acquisition" <LogicLev>

Module:	LogicLev
Name:	Logic level
Type:	DAQ
Source:	daq_LogicLev.so
Version:	1.3.0
Author:	Roman Savochenko
Translated:	Maxim Lysenko
Description:	Provides the logical level of parameters.
License:	GPL

Contents table

Introduction

The module is a pure logic-level implementation mechanism, based on the templates of parameters of the subsystem "Data acquisition — DAQ". The implementation of the module is based on the "Logical level of the parameters of the system OpenSCADA". Practically, this module is an implementation of the subsystem "Options" of the project without templates and putting it into the module.

The module provides a mechanism for the formation of the parameters of subsystem "DAQ", based on other sources of the subsystem at the level of the user. Actually, the module uses templates of subsystem "DAQ" and the specific format for the description of references to the attributes of the parameters of subsystem "DAQ".

Also, the module implements the functions of the horizontal reservation, namely, working in conjunction with the remote station of the same level. In addition to the synchronization of the archives of values and archives of attributes of parameters the module implements synchronization of computational templates, in order to shockless catch of the algorithms.

1. Data controller

For addition of the data source of parameters of the logical level the controller in the system OpenSCADA is created and configured. Example of the configuration tab of the controller of the type is depicted in Figure 1.
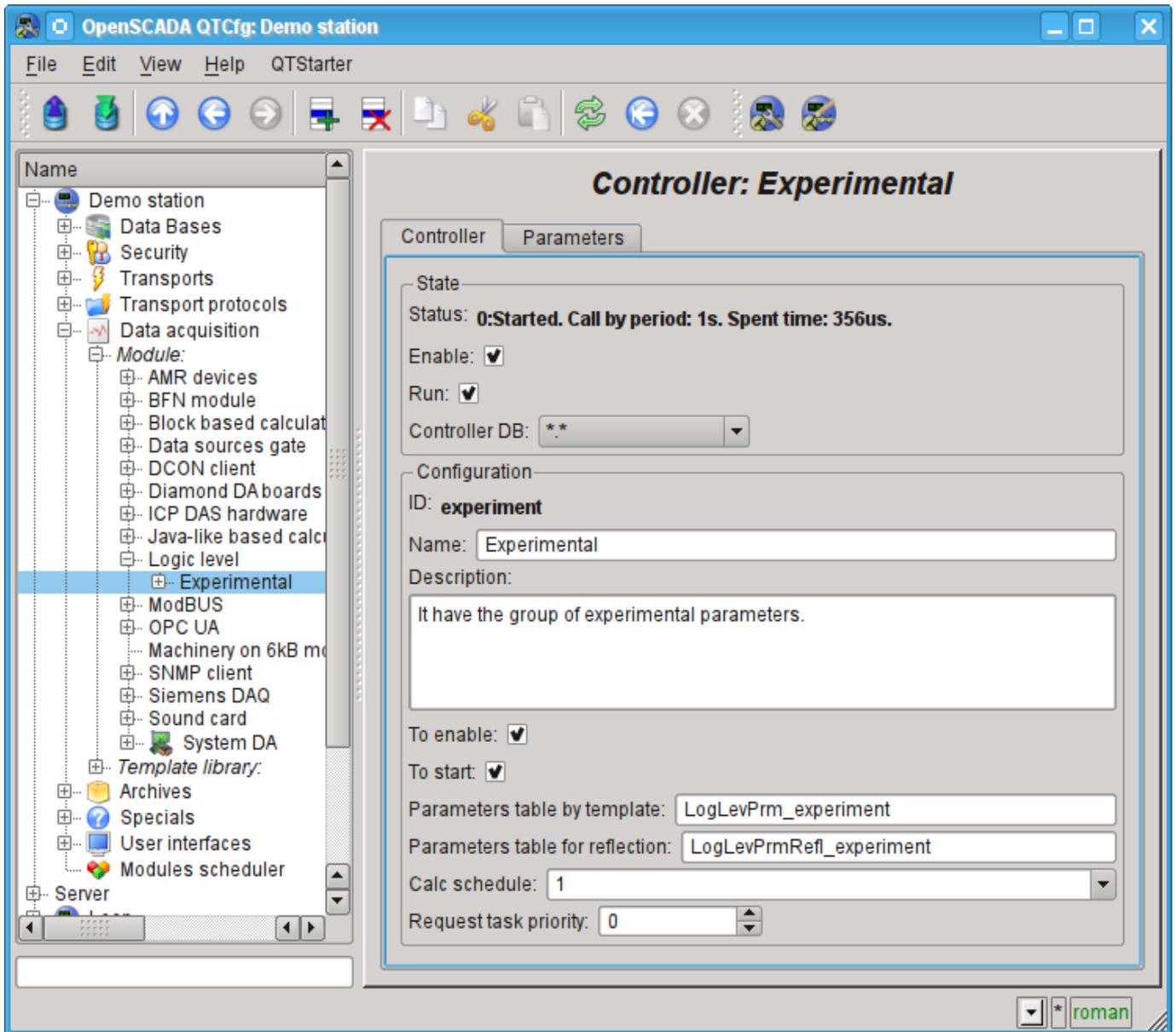

Fig.1. Configuration tab of the controller.

From this tab you can set:
- The state of the controller, as follows: Status, «Enable», «Run» and the name of the database containing the configuration.
- Id, name and description of the controller.
- The state, in which the controller must be translated at boot: «To enable» and «To start».
- Name of tables to store the settings based on templates and direct reflection external parameter of DAQ.
- The schedule policy and the priority of the task of the interrogation of data sources.

2. Parameters

The LogicLev module provides two types of parameters: "Logical"(std) and "Reflection parameter"(pRefl). Additional configuration fields, the parameters of the module (Fig. 2) are:

- *"Logical"(std)*:
  - **Parameter template** — template of DAQ parameter address.
- *"Reflection parameter"(pRefl)*:
  - **Source parameter** — source reflected parameter address.
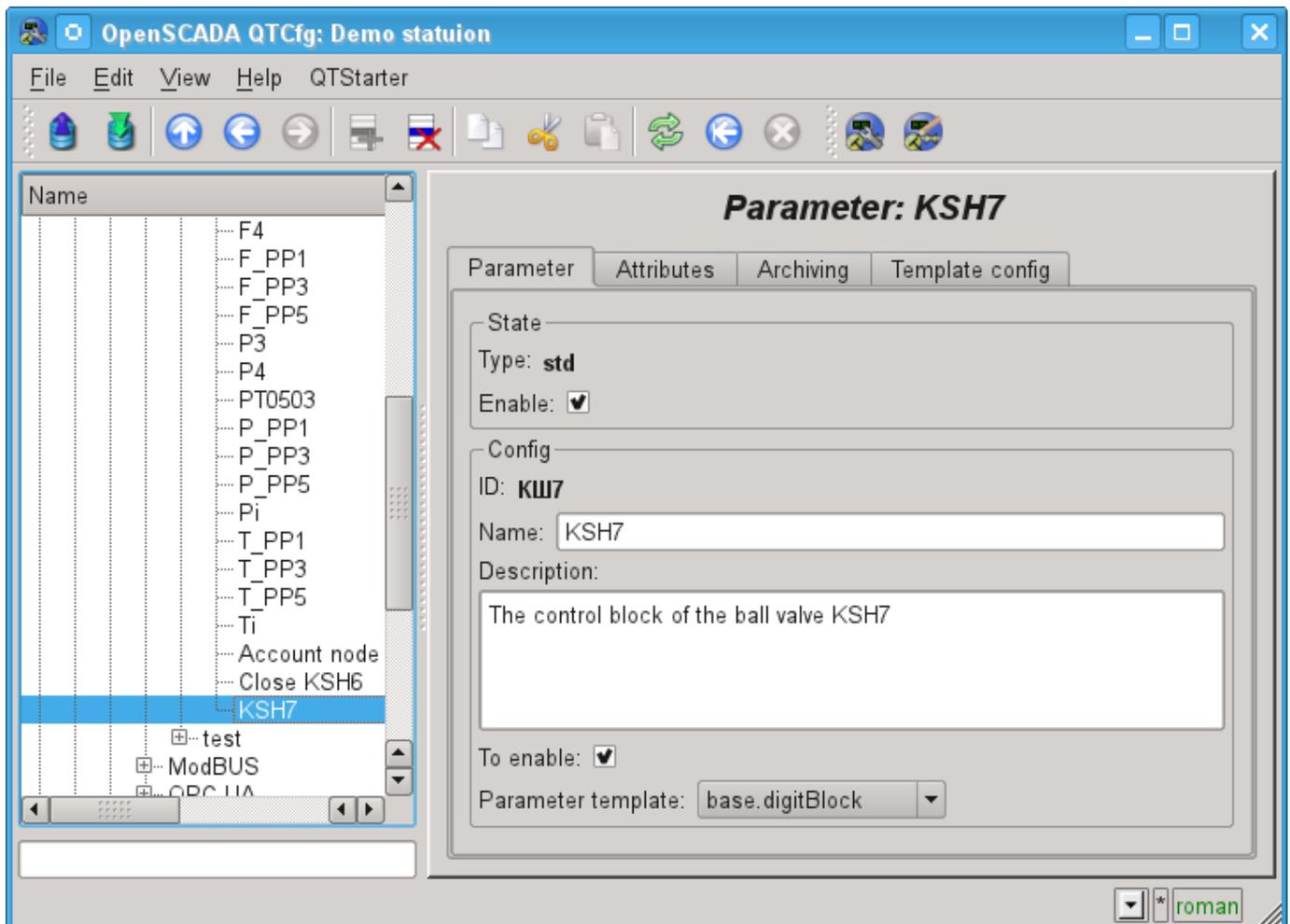
Fig.2. Configuration tab of the parameter.

Logical type parameter (std)

When building a template, for logical parameter type of the controller, the peculiarity of the link format of the template must take into account. Reference should be written in the form: **<Parameter>| <identifier>**, where:

        <Parameter> — line, characterizing the parameter;
        <Identifier> — id of the attribute of parameter.

This record allows to group multiple attributes of a source parameter and assign them only by the choice of the parameter. Ie in the configuration dialog of the template (Fig. 3) it will be shown only parameter. This does not preclude the possibility to assign the attributes of the parameter each separately, in addition, if you miss in the configuration of the template the description of the links in the specified format, it will be assigned an attribute of the parameter (Fig.4).

The module provides a special treatment of a number of attributes of the template:
- *f_frq* — Frequency computation procedure template, or the time after the last calculation, the negative, in seconds, for scheduling of CRON, read-only.
- *f_start* — First calculate of template's procedure, start, read-only.
- *f_stop* — Last calculate of template's procedure, stop, read-only.
- *f_err* — The parameter error, full access. Value of the attribute is set to the parameter's error attribute "err".
- *SHIFR* — The parameter code, read-only.
- *NAME* — The parameter name, read-only.
- *DESCR* — The parameter description, read-only.
- *this* — The parameter object, allow access to attributes of the parameter, for example to their archives access.

Sign "(+)" at the end of the address signals about successful linking and presence of the target.
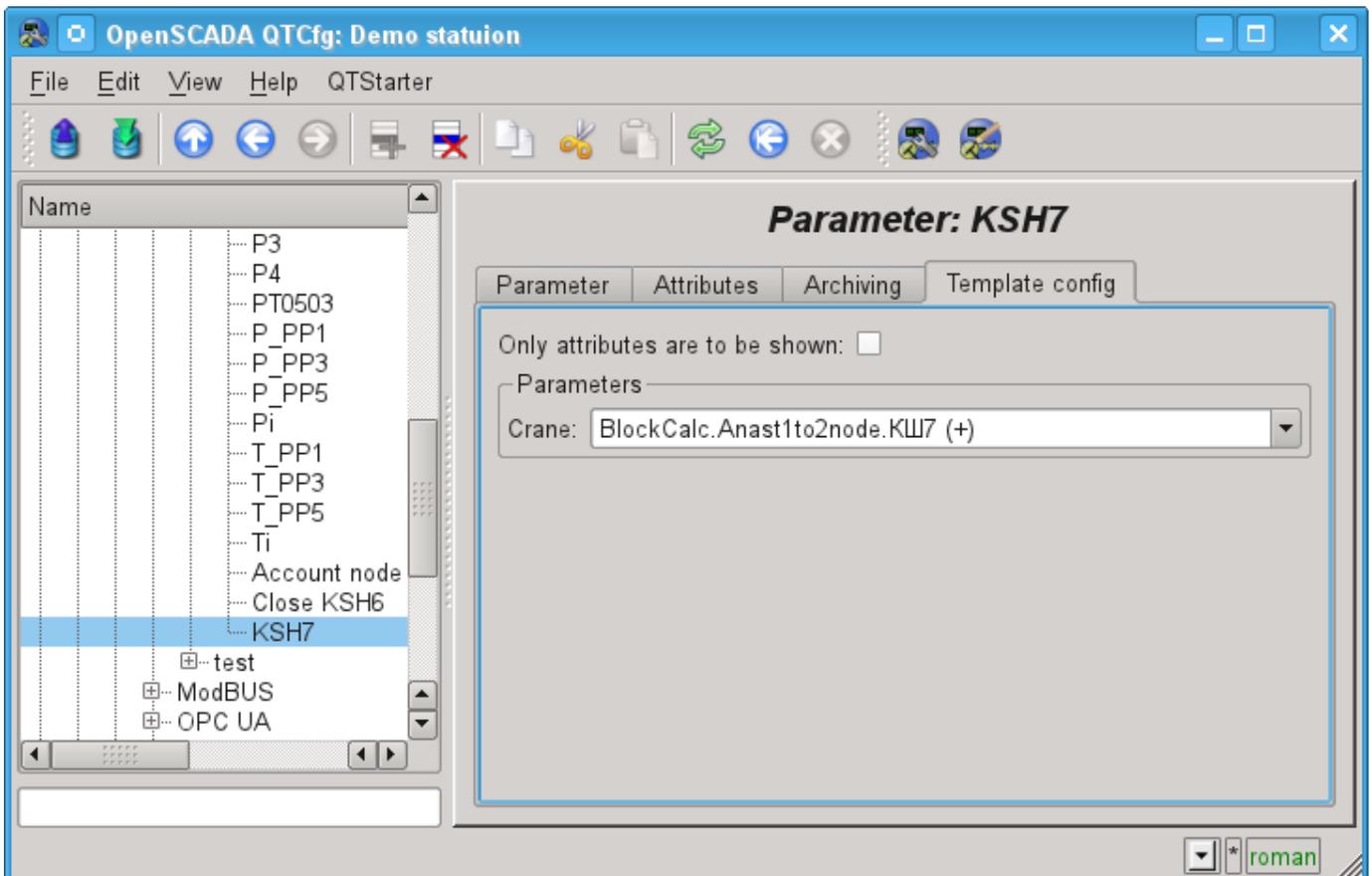

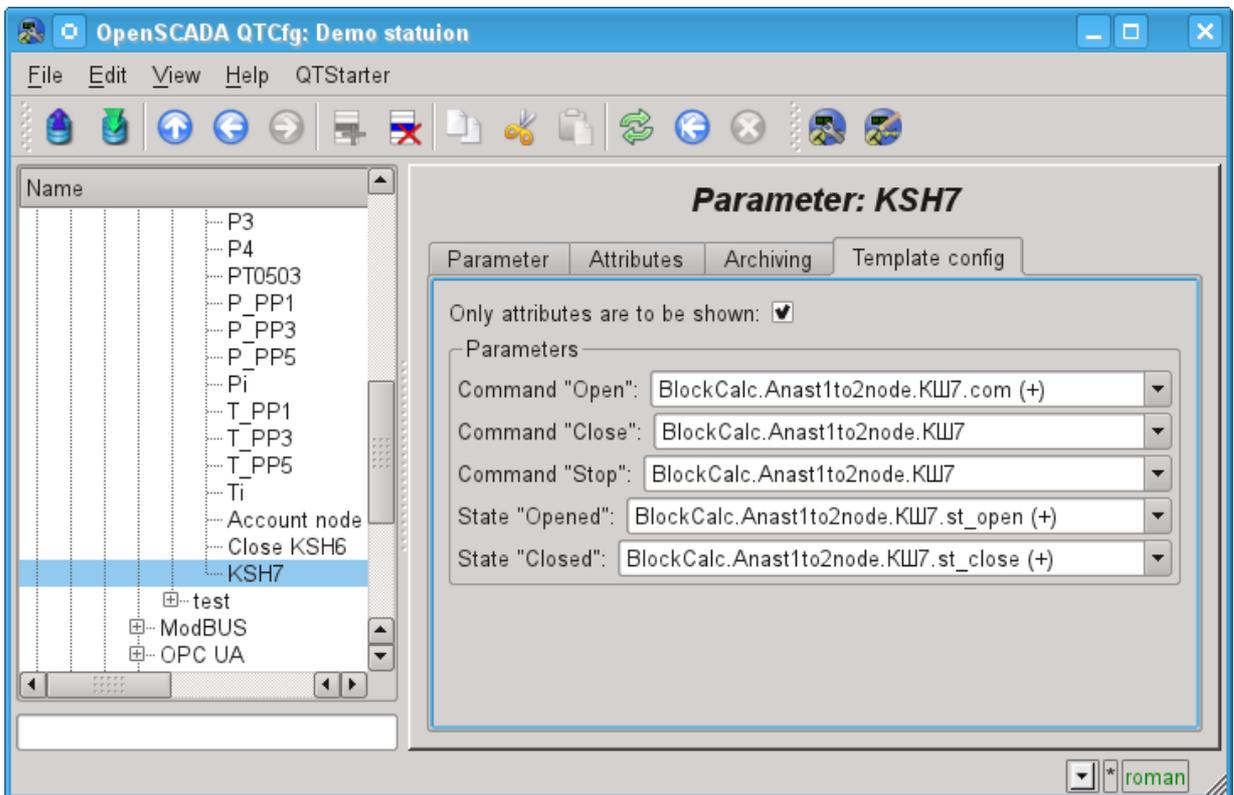Fig.3. Configuration tab of the template of parameter.

Fig.4. Configuration tab of the template of parameter. Show only attributes.

In accordance with the template underlying the parameter, we get the set of attributes of the parameter Fig.5.
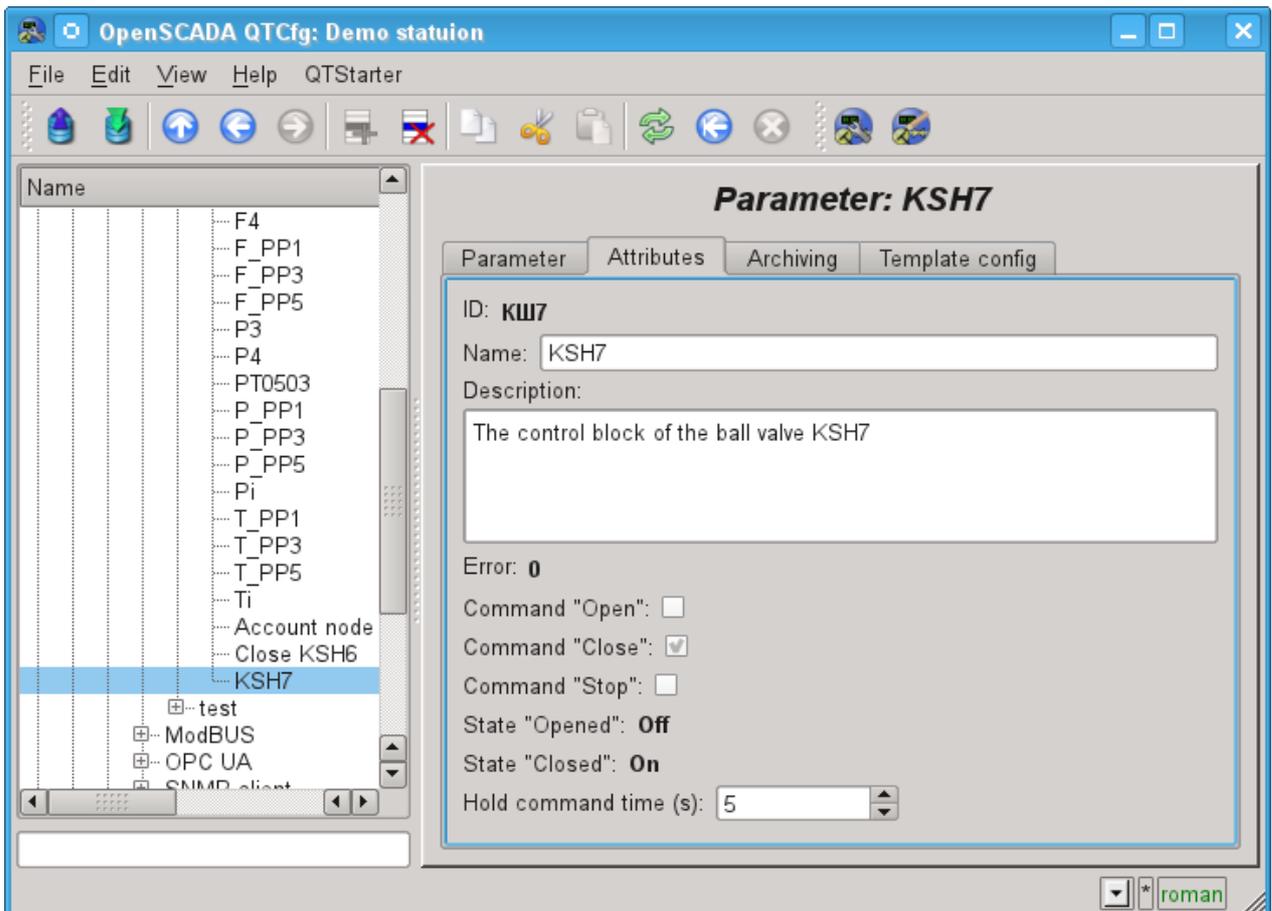

Fig.5. Tab of the attributes of the parameter.

Parameter reflection (pRef)

All attributes from specified to reflect parameter just become available in this parameter thereby realizing the function proxy, for example, for include the parameters from other sources into a single, export, object controller (PLC).