The <OPC_UA> module of "Data acquisition" and "Transport protocols" subsystems

Parameter	Module 1	Module 2
ID:	OPC_UA	OPC_UA
Name:	OPC_UA	OPC_UA
Type:	DAQ	Protocol
Source:	daq_daq_OPC_UA.so	
Version:	0.6.0	0.6.0
Author:	Roman Savochenko	
Translated:	Maxim Lysenko	
Description:	Provides the implementation of client service of OPC UA.	Provides the implementation of the OPC UA protocol.
License:	GPL	

Contents table

Introduction

OPC (OLE for Process Control) - it is the family of protocols and technologies that provide the single interface to control the objects of automation and technological processes. The creating and support of specifications of OPC coordinates an international nonprofit organization OPC Foundation, established in 1994 by the leading manufacturers of industrial automation.

In view of the fact that a significant influence in the OPC Foundation organization has the Microsoft corporation, OPC protocols, until recently, was single platform and closed, due to binding to the closed technologies of MS Windows. However, more recently, the OPC Foundation organization has created multi-platform interfaces such as OPC XML DA and OPC UA. The most interesting of them is the OPC UA interface, as unifying all the earlier interfaces in an open and multi-platform technology.

This module implements the interface and protocol support for OPC UA in the form of client service, and as the OPC UA server. Client service of OPC UA is implemented by the same name module of the subsystem "Data acquisition", and the server is implemented by the subsystem's "Protocols" module.

In the current version of these modules it is implemented the binary part of the protocol and basic services in unsafe mode and safe mode of policies "Base128Rsa15" и "Base256". Later it is planned to extend the module to work via HTTP/SOAP and implementation of other OPC UA services.

Although the OPC UA protocol is multi-platform, its specification and SDK are not freely available, but are provided only to members of the OPC Foundation organization. For this reason, the implementation of these modules has faced significant obstacles and problems.

First, the protocol OPC UA is complex and its realization in general without specification an extremely laborious. For this reason, the work on these modules for a long time was not started, and only thanks to sponsorship by an organization-member of OPC Foundation the OpenSCADA project received documentation of the specification. The SDK and source code ANSIC-API of the OPC-UA protocol have not been received due to their incompatibility with the GPL license and as a consequence, the potential threat of violation of the license when working with source code, which could lead to subsequent legal problems with the free distribution of these modules.

Secondly, even the presence of specification does not allow to solve some technical question without an example of implementation and the possibility of test the working prototype of the client and server of OPC UA. For example, it is the technical features of the implementation of symmetric encryption algorithms and the keys for them do not allowed to make the implementation of support for security policy at once.

To debug the operation of modules the demonstration software of company Unified Automation consisting of the OPC UA client - UA Expert and Server - OPC UA Demo Server, from SDK package, was used.

1. OPC UA protocol

OPC UA - is the platform-independent standard by the means of which the systems and devices of various types can interact by sending messages between the client and the server through various types of networks. The protocol supports secure communication through the validation of clients and servers, as well as the counteraction to attacks. OPC UA defines the concept *Services* that the servers can provide, as well as services that the server supports for the client. Information is transmitted as the data types defined by OPC UA and producer, in addition the servers define object model, for which the clients can implement the dynamic review.

OPC UA provides the combination of integrated address space with service model. This allows the server to integrate data alarms and events, the history in this address space, as well as provide access to them through integrated services. Services also provide an integrated security model.

OPC UA allows servers to provide for clients the definitions of types for access to the objects of the address space. OPC UA supports the provision of data in various formats, including binary structures and XML-documents. Through the address space clients can request the server metadata that describe the data format.

OPC UA adds the support for multiple connections between nodes instead of a simple hierarchy. Such flexibility in combination with types' definition allows to use OPC UA for solving problems in the wide problem area.

OPC_UA is designed to provide the reliable output of data. The main feature of all OPC servers - the ability to issue the data and events.

OPC_UA is designed to support the wide range of servers, from simple PLC to industrial servers. These servers are characterized by the wide range of sizes, performance, platforms and functional capacity. Consequently, the OPC UA defines the comprehensive set of possibilities, and the server can implement the subset of these possibilities. To ensure the interoperability between OPC UA defines the subsets, named the *Profiles* that the server can indicate for agreement Clients may subsequently make the review of server's profiles and make the interaction with the server, based on the profiles.

OPC UA specification is designed as the core in the layer, isolated from the underlying computer technologies and network transports. This allows OPC UA if necessary to expand on the future technologies without exclusion the framework of design. Currently, the specification defines two ways to data encode: XML/text and UA Binary. In addition, the two types of transport layer are defined: TCP and HTTP/SOAP.

OPC UA is designed as the solution for migration from OPC clients and servers, which are based on Microsoft COM technologies. OPC COM servers (DA, HDA and A&E) can be easily reflected in the OPC UA. Producers can independently make such migration or recommend users to use wrappers and converters between these protocols. OPC UA unifies the previous models in the single address space with the single set of services.

2. The module of the protocol implementation

Protocol module contains the implementation code for the protocol part of the OPC UA for both the client and for server services. To construct the OPC UA server it is enough to create an incoming transport, for ordinarily this TCP-transport of module Sockets, and select in it the module of the protocol, and also configure although one enpoint node of protocol module, about it bellow.

2.1. Service the requests on the OPC UA protocol

Incoming requests to the module-protocol are processed by the module in accordance with configured end points of OPC UA (EndPoints) (Fig. 1).
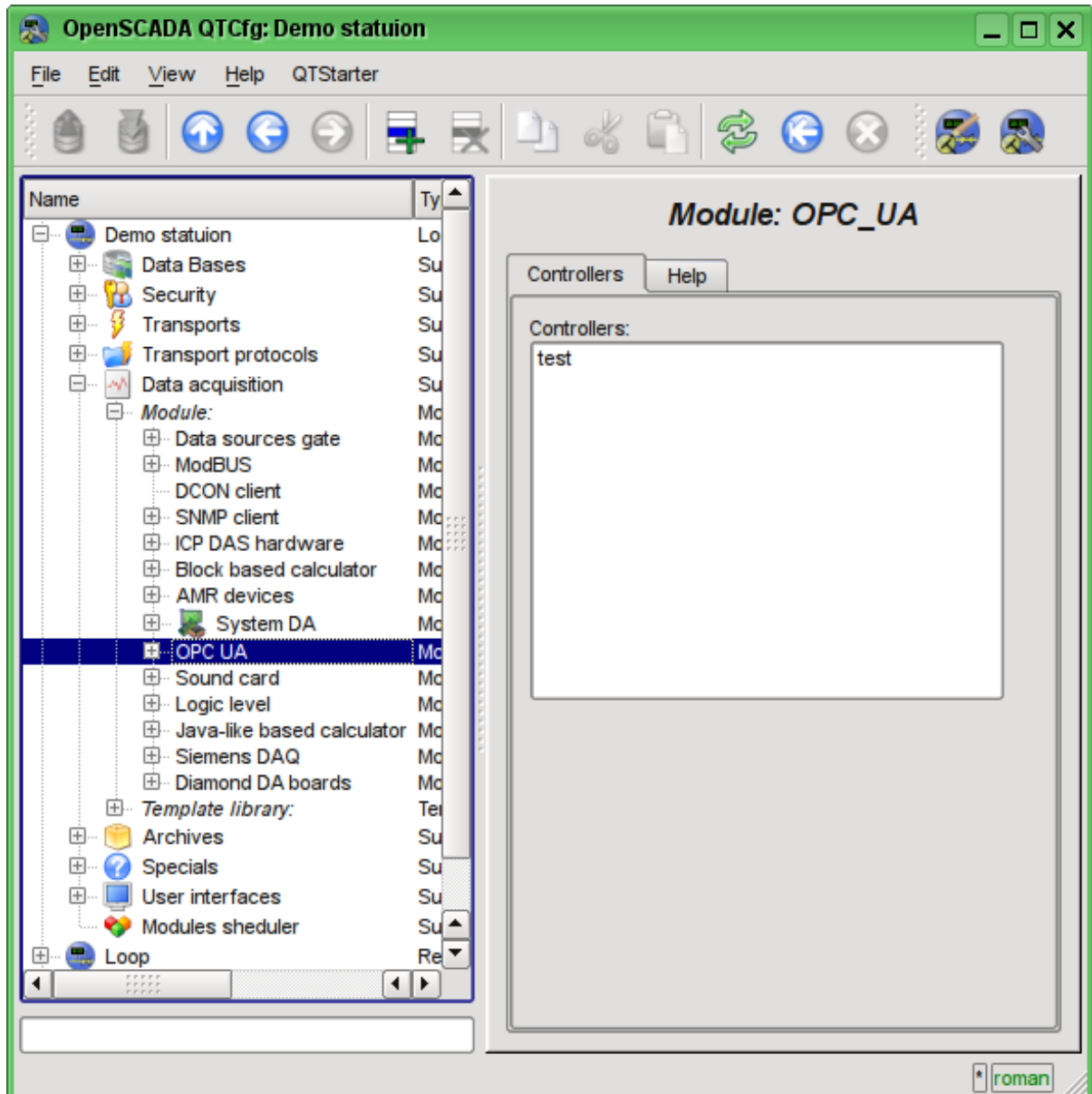

Fig.1. End points of the protocol.

Endpoint of the OPC UA protocol is actually the server object of OPC UA. End points in OPC UA can be either local or remote. The local one is designed to provide the resources of OpenSCADA station to protocol OPC UA, while the remote end points are both for the service and review of available OPC-UA units, and for locking requests to remote stations. In this version of the module is only supported the configuration of the local endpoints.

The general configuration of the endpoint is made on the main tab of the endpoint page (Fig. 2) with the parameters:

- Node status, namely: the state, "Enable" and the name of the database containing the configuration.
- ID, name and description of the node.
- The state, in which to transfer the node at start: "Enable".
- Type of the protocol coding. At the moment it is only "binary".
- URL of the end point.
- The server certificate in the PEM format.
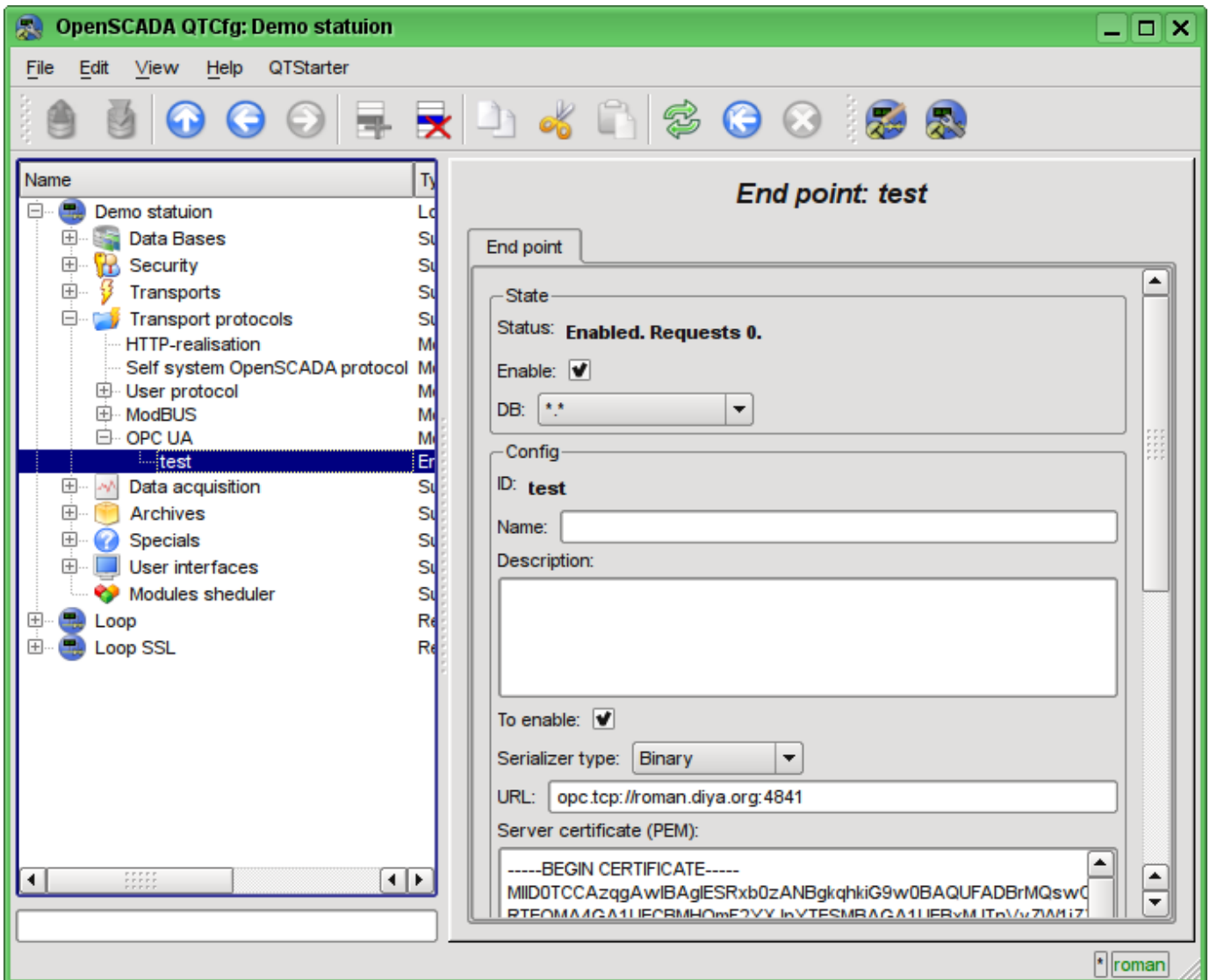- The private key in the PEM format.
- Security Server Policy.

Fig.2. The main tab of the end node.

3. Data acquisition module

Data acquisition module provides the ability of inquiry and recording the value's attributes (13) of points with the "Variable" type.

3.1. Data controller

To add the OPC UA data source controller in the OpenSCADA system is created and configured. An example of the configuration tab of the controller is shown in Figure 3.
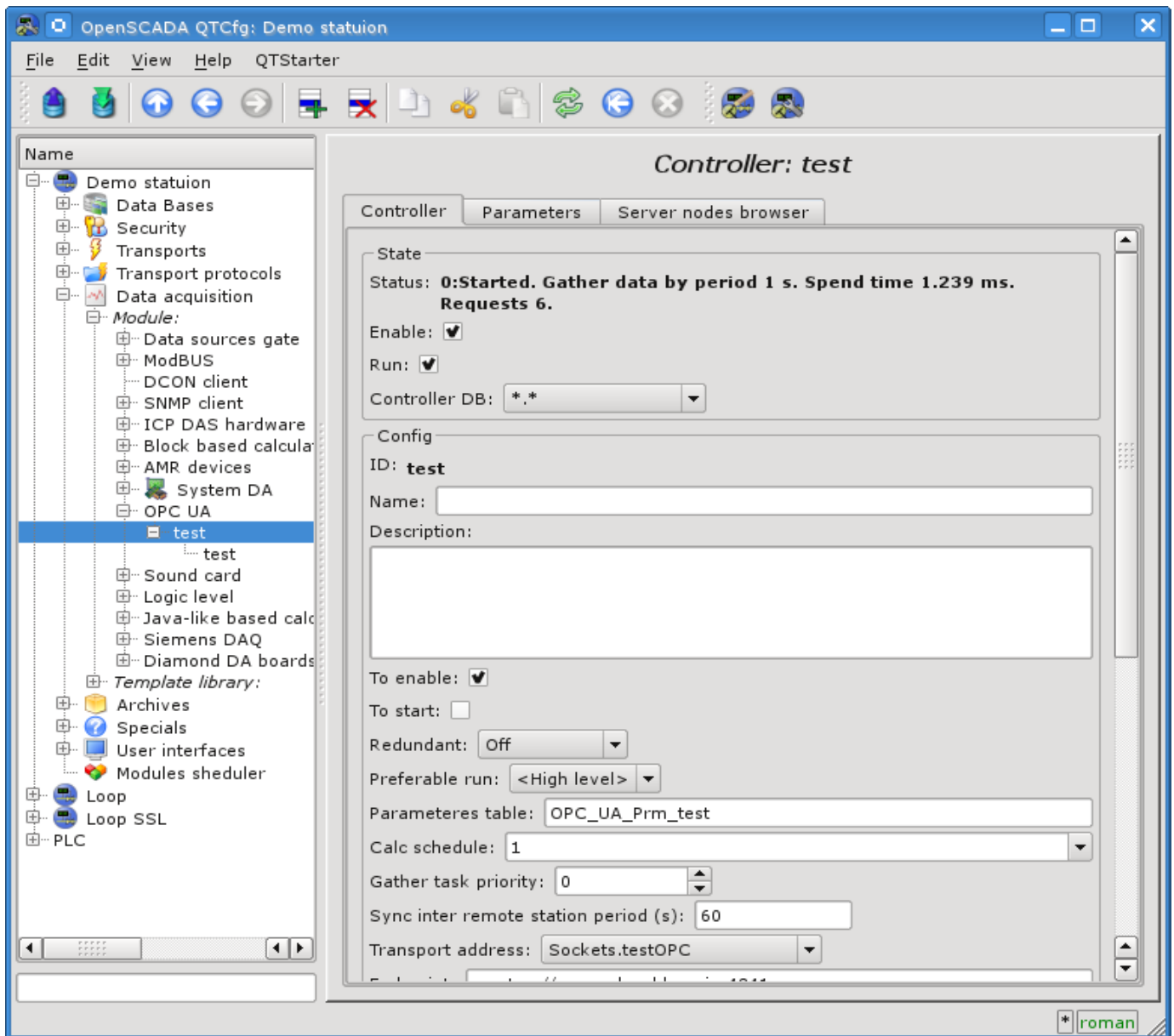

Fig.3. Controller's configuration tab.

From this tab you can set:
- The state controller, namely: Status, "Enable", "Run"and the name of the database containing the configuration.
- ID, name and description of the controller.
- The state, in which to transfer the controller at start: "Enable", "Run".
- The mode of horizontal redundancy and preference for the execution of the controller.
- The name of the table to store the configuration of parameters of the controller.
- The parameters of the execution the acquisition task scheduler and its priority.

• The period of the synchronization of the configuration of attributes of the parameters with the remote station, and try time for connection restore.

• The address of the outgoing transport from the list of configured outgoing transports in the subsystem "Transports" of OpenSCADA.

• The URL of the endpoint of remote station.

• Security policy and the mode of messaging security.

• The client certificate and private key in PEM format.

• The limit of the number of attributes in the parameter for the import mode of all the attributes belonging to the object.

To facilitate the identification of nodes on the remote station, as well as their choice to be inserted in the parameter of the controller in the controller's object it is provided the navigation on the remote station's nodes tab, where you can walk through the tree of objects and familiar with their attributes (Figure 4).
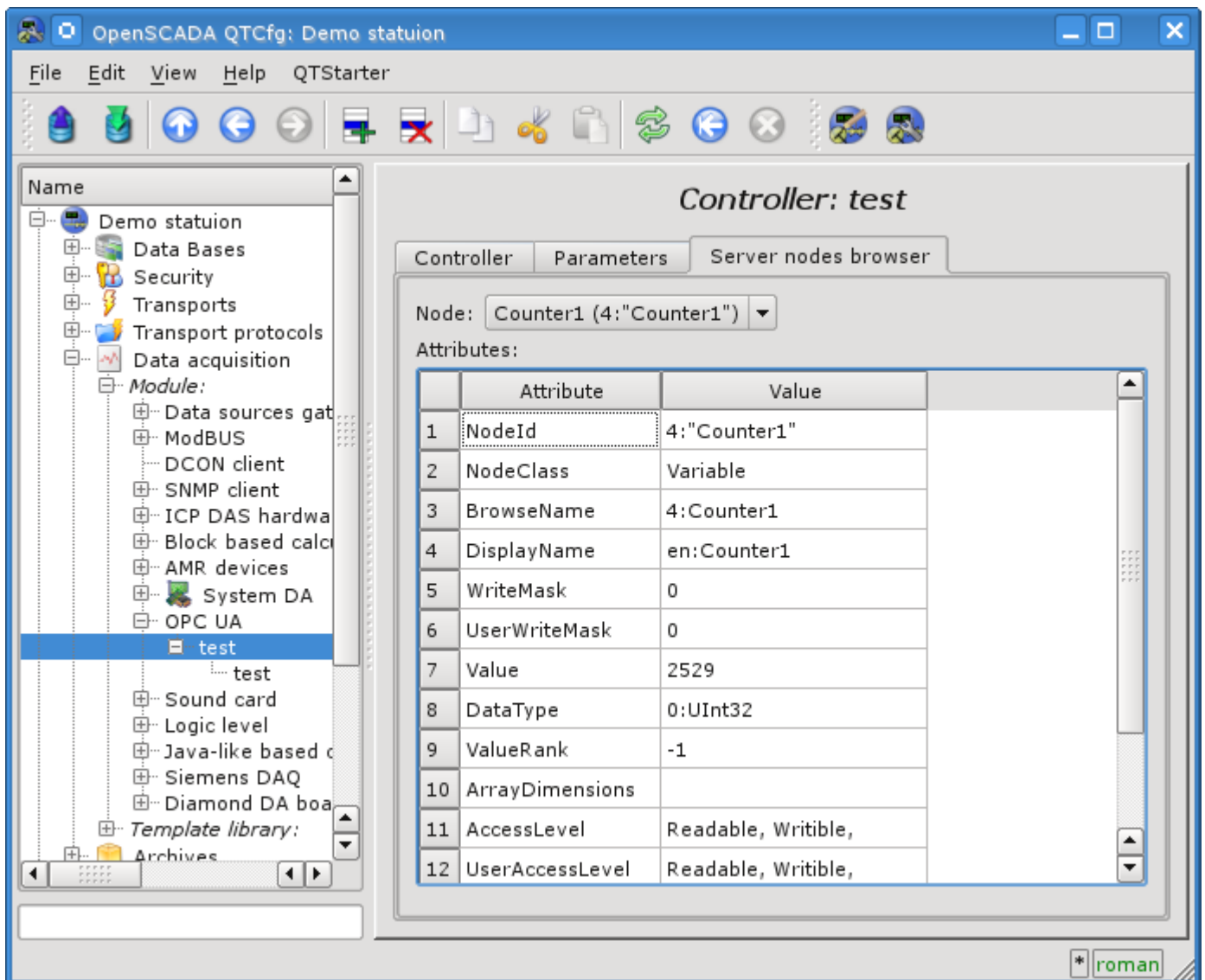

Fig.4. The "Server nodes browser" tab of the controller's page.

3.2. Parameters

Data acquisition module provides only one type of parameters - "Standard". Additional configuration field of the parameter of the module (Fig. 5) is the list of OPC UA nodes. Attribute in this list is written as follows: [ns:id].

Where:
  ns - names scope, number, zero value can missed;
  id - node identifier, number, string, bytes string and GUID.

Example:
  84 - root directory;
  3:"BasicDevices2" - basic devices node in the names scope 3 and string view;
  4:"61626364" - node in the names scope 4 and byte string view;
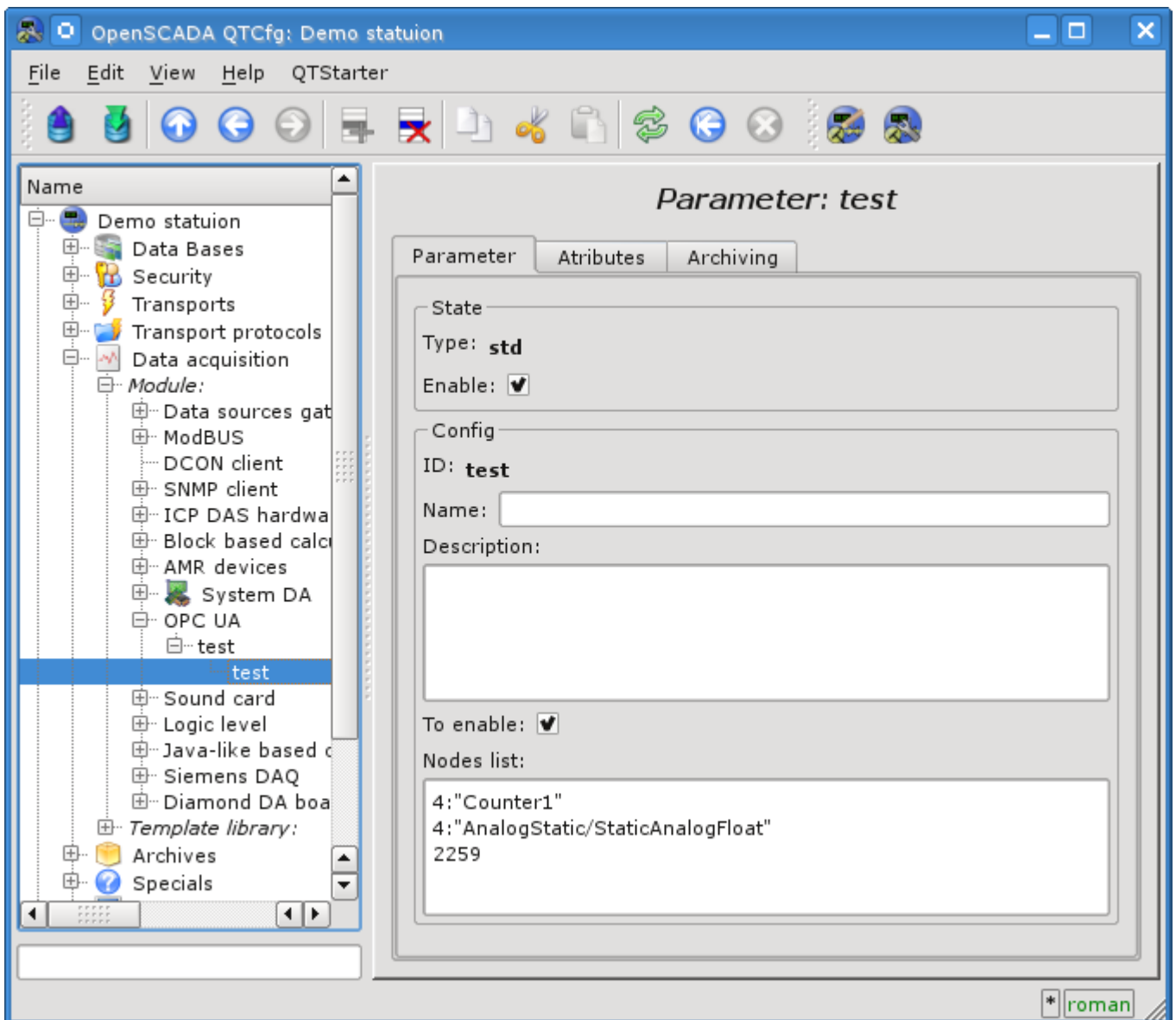 4:{40d95ab0-50d6-46d3-bffd-f55639b853d4} - node in the names scope 4 and GUID view.

Fig.5. The configuration tab of the parameter.

In accordance with the specified list of nodes the inquiry and the creation of the parameter's attributes is made (Fig. 6).
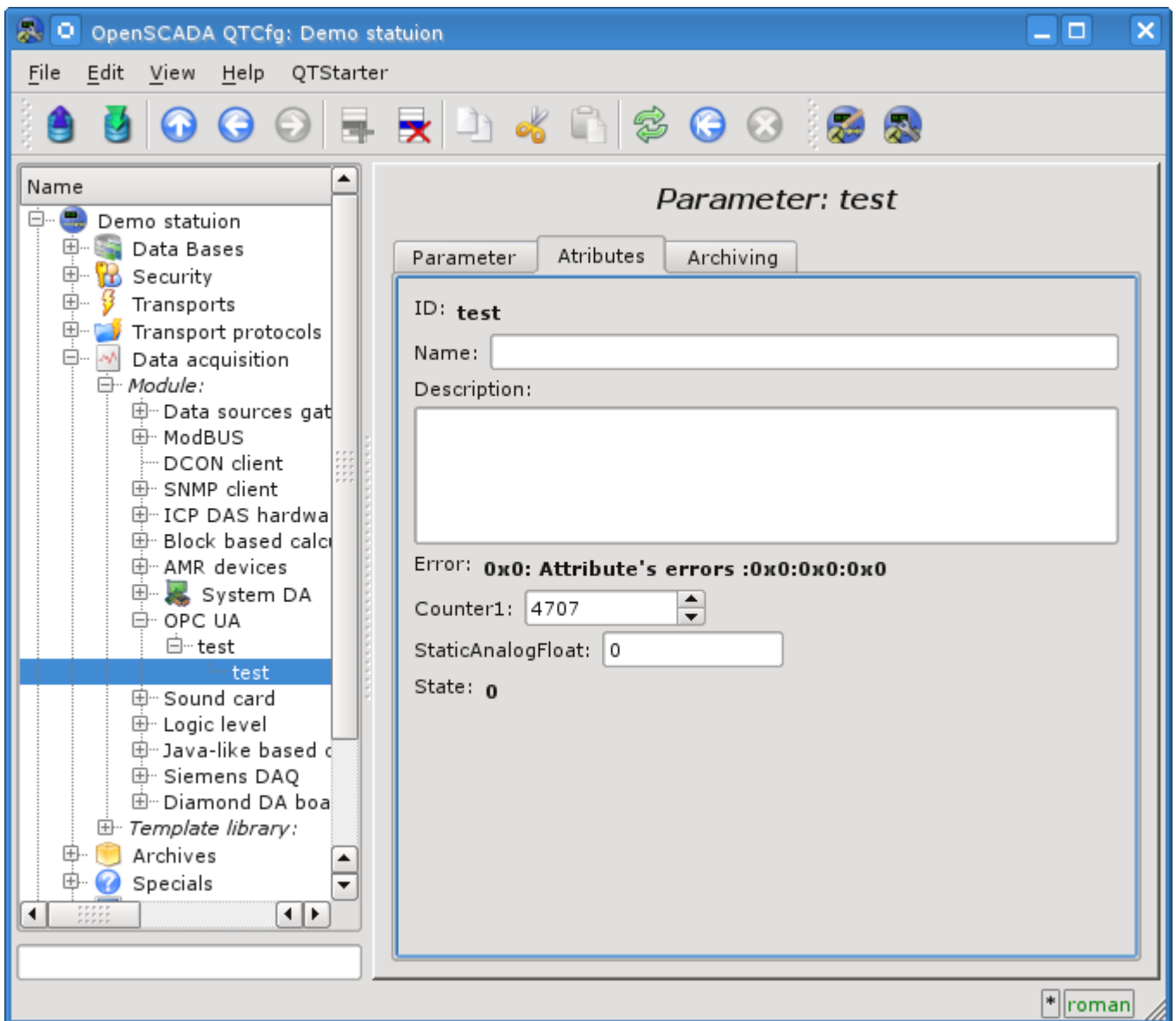

Fig.6. The parameter's attributes tab.

4. Notes

During the implementation of modules supporting OPC UA was detected several inconsistencies with the official SDK specification OPC UA:

- OPC UA Part 6 on page 27 contains an image of a handshake to establish a secure channel. The message of session create is signed by the client symmetric key and encrypted by server. In fact, both signature and encryption of the server key made.

- OPC UA Part 4 on page 141 contains a description of data structure signatures, which are the first data signature, and then the string algorithm. In fact, the reverse order is implemented.