The modules <ModBus> of subsystem "Data acquisition" and subsystem "Transport protocols"

Parameter	Module 1	Module 2
ID:	ModBus	ModBus
Name:	ModBus	ModBus
Type:	DAQ	Protocol
Source:	daq_ModBus.so	daq_ModBus.so
Version:	1.0.0	0.5.0
Author:	Roman Savochenko	Roman Savochenko
Translated:	Maxim Lysenko	
Description:	Provides implementation of client service of the protocol ModBus. Modbus/TCP, Modbus/RTU and Modbus/ASCII protocols are supported.	Provides implementation of protocols ModBus. Modbus/TCP, Modbus/RTU and Modbus/ASCII protocols are supported.
License:	GPL	GPL

Contents table

Introduction

ModBus – communication protocol based on the client-server architecture. It was developed by Modicon for using in the programmable logic controllers (PLC). It became the de facto standard in the industry and is widely used for the connection of industrial electronic equipment. Used to transfer data via serial line RS-485, RS-422, RS-232, and network TCP/IP. Currently supported non-profit organization ModBus-IDA.

There are three modes of the protocol: ModBus/RTU, ModBus/ASCII and ModBus/TCP. The first two use the serial communication lines (mostly RS-485, less RS-422/RS-232), the last uses TCP/IP network to transfer data.

Module of the data acquisition provides an opportunity to gather the information from various devices by means of the protocol ModBus in all modes. Also, the module implements the functions of the horizontal reservation, namely, working in conjunction with the remote station of the same level. At the same time, the module of the protocol allows you to create and issue data by means of the protocol ModBus in various modes, and through interfaces that are supported by modules of subsystem "Transports".

1. General description of the ModBus protocol

Protocol ModBus/RTU requires one lead(requesting) device in the line(master), which can send commands to one or more driven devices(slave), referring to them by a unique in the line address. Syntax of the commands of the protocol allows to address 247 devices on the one connection line of standard RS-485(less RS-422 or RS-232). In the case of TCP addressing mode is excluded from the protocol, as it is implemented in the TCP/IP stack.

Initiative of exchange always comes from the leading device. Slave devices listen the line. Master request (package, the sequence of bytes) in the line and turns into a listening line. Slave device responds to the request, which came to him. The end of the response package the master determines by the time interval between the end of the reception the previous byte and the beginning of the reception of the following. If this interval exceeds the time required to receive one and a half of the byte on the given speed, receiving of the response frame is considered complete. Frames of the request and reply in the ModBus protocol have the fixed format.

1.1. Addressing

All data operations are tied to zero, each type of data (register, bit, register of input or bit of input) addresses begin with 0000 and ends 65535.

1.2. Standard codes of functions

In ModBus protocol it can be divided into several subsets of commands(Table 1).

Table 1: **The subset of commands of ModBus protocol**

Subset	Range of codes
Standard	1–21
Reserve for advanced features	22–64
Custom	65–119
Reserve for own needs	120–255

Module of the protocol process the requests by the commands 0x03 and 0x06 for reading and writing registers, 0x01 and 0x05 for reading and writing bits.

2. Module of the implementation of the protocol

ModBus protocol module contains the code implementing of the protocol part of ModBus, namely particular variants of protocols ModBus/TCP, ModBus/RTU and ModBus/ASCII. Module of the protocol in conjunction with the selected transport is actively used by the data acquisition module for direct queries implementation. Because of the module of the protocol is independent, by using of it you can create additional modules for data acquisition by non-standard functions of the expansion of ModBus of various automation equipment.

2.1. API functions of outgoing requests

API functions for outgoing queries operate with the exchange of blocks PDU, XML-wrapped in packages with the following structure:

      <prt id="sId" reqTm="reqTm" node="node" reqTry="reqTry">[pdu]</prt>

Where:

- *prt* --name of the tag with the name of the used variant of the protocol (TCP, RTU or ASCII).
- *sId* — identifier of the source of the query. Used for placing to the protocol the output protocol.
- *reqTm* — time of the request, namely the time during which the answer is expected.
- *node* — number of the destination node or the identifier of the unit ModBus/TCP.
- *reqTry* — number of attempts of repeating the request with the error in the answer. Only for ModBus/RTU and ModBus/ASCII.
- *pdu* — directly block of the unit of the protocol data (PDU) ModBus.

The resulting pdu replaces the request pdu in the XML-package, and set the attribute "err" with the code and text of the errors, if it is took place.

2.2. Servicing of the requests for ModBus protocol

Input part of servicing of the requests to the module of the protocol realizes verification and processing of the requests through objects of the nodes, provided by the module(Fig. 1). Actually, the mechanism is implemented, that allow the system OpenSCADA to perform the role of the ModBus/TCP server or the slave device of ModBus/RTU and ModBus/ASCII. Thus the system OpenSCADA gets an opportunity to serve the role of any participant of the ModBus networks.
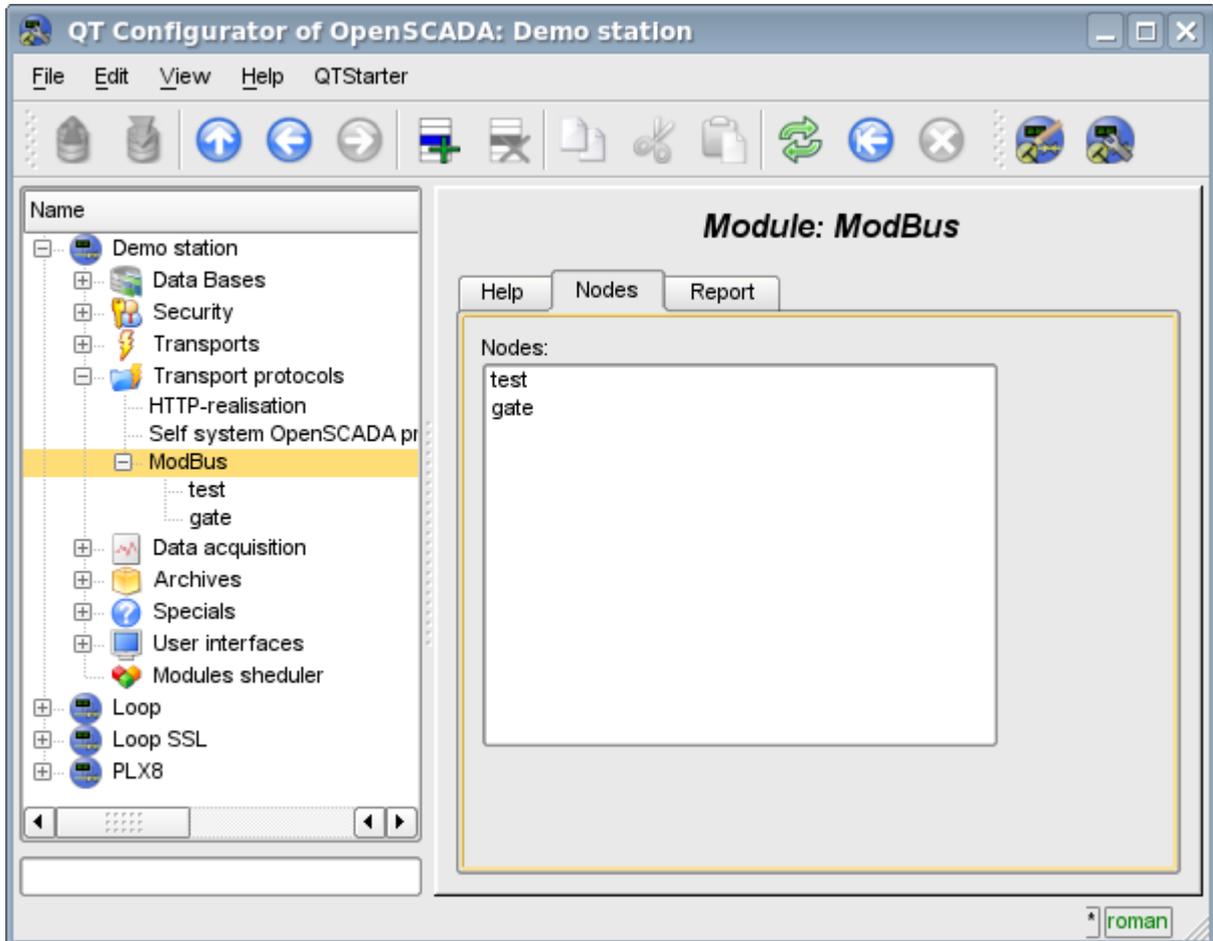
Fig.1. Tab of the list of the nodes of servicing incoming requests of the protocol.

The node of the protocol is equivalent to the physical node of the device of ModBus network. Node of the protocol can operate in three modes:
• "Data" — mode of the reflection of data of OpenSCADA on arrays of registers and bits of ModBus to transfer them at the request of the client node or master.
• "Gateway of the node" — mode of the redirecting of the requests to the node of the another ModBus network through this node.
• "Gateway of the network" — mode of the redirecting of the requests to any node in another ModBus network, actually carrying out the integration of several ModBus networks into one.

Since the protocol nodes can be created a great number, it turns out that on the one interface, ie in the one network, one station on the basis of OpenSCADA can clear provide multiple nodes of ModBus network with different data.

Lets consider particular configuration of the node of the protocol in various modes.

The mode of the node of the protocol "Data"

Mode is used to reflect the data of OpenSCADA on arrays of registers and bits of ModBus. The overall configuration of the node is made in the tab "Node"(Fig. 2) by the parameters:
- The state of the node, as follows: «Enabled» and the name of the database containing the configuration.
- Id, name and description of the node.
- The state, in which the node must be translated at boot: «Enabled».
- Address of the node in the ModBus network from 1 to 247.
- Inbound traffic, to the network of which the node is belonged to. It is selected from the list of input transport of the subsystem "Transports" of OpenSCADA. Specifying as the transport the symbol "*" makes this node a participant of any ModBus network with the processing of requests from any transport.
- Variant of the ModBus protocol, requests in which must be processed by the node from the list: All, RTU, ASCII, TCP/IP.
- The choice of the mode, in this case the mode "Data".
- Period of calculation of data in seconds. Specifies the frequency of processing of forming for the requests data, namely, data tables of ModBus, calculation of data programs and servicing of links to the data of OpenSCADA.

Node in this mode process the following standars commands of the ModBus protocol:
- *01* — reading of the group of bits;
- *03* — reading of the group of registers;
- *05* — settig of the single bit;
- *06* — settig of the single register.
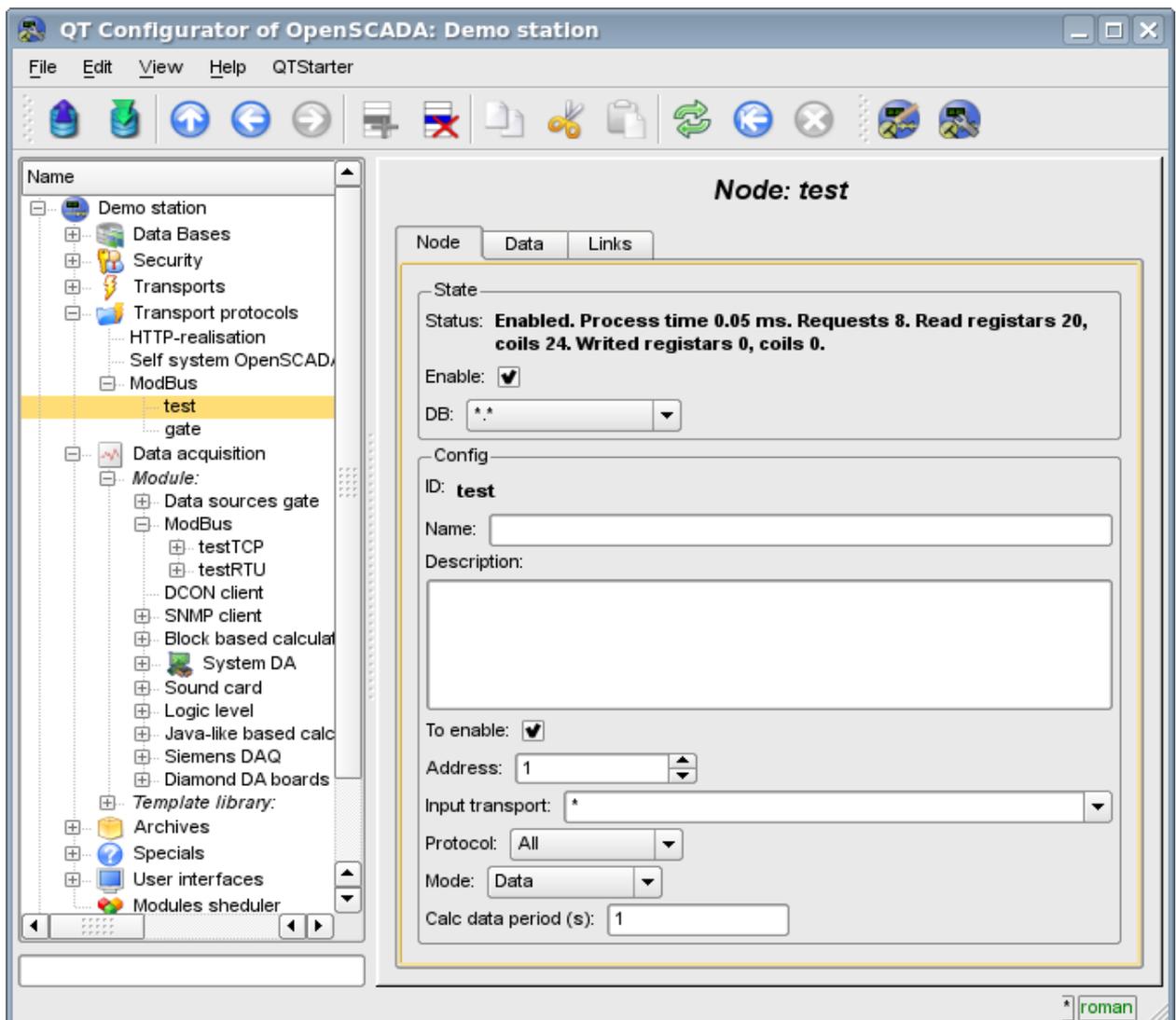
Fig.2. The tab "Node" of the configuration page of the node of the protocol in the "Data" mode.

To form the table of the reflection of the data of ModBus network, namely, registers and bits the tab "Data" is provided(Fig.3). The tab "Data" contains a table of parameters and program for processing of the parameters with the specified programming language, which is available in the system OpenSCADA. Table contains the parameters with the properties:

- *Id* — ID of the parameter. It is the key for the formation of the tables of registers and bits of ModBus. To specify that this parameter is the register of the ModBus, identifier must be written as «R[N]w», where N – number of the register's number from 0 to 65535, and w – optional character indicating the possibility of setting of it by the ModBus request eg: R23, R456, R239w. For the ModBus bit specifying, ID must be written as «C[N]w», where N – number of bits from 0 to 65535, and w – optional character indicating the possibility of setting of it by the ModBus request, eg: C437, C0, C39w. All other parameters are internal and are used for a variety of intermediate calculations, processing and conversion.
- *Name* — The name of the parameter is used for the naming of the connection.
- *Type* — Type of the parameter from the list: "Real", "Integer", "Boolean" and "String". For the registers and bits of ModBus it makes sense to set "Integer" and "Boolean" type, respectively.
- *Connection* — Sign that this option should be to connect with the attribute of the parameter of subsystem "Data acquisition". These connections are set in the "Connection" tab.
- *Value* — The original or current, if the node is switched on, the value of the parameter.

The table by default identifies several parameters of special significance:
- *f_frq* — frequency of computing the table by the program;
- *f_start* — sign of the first computing, the start up of the program.
- *f_stop* — sign of the last execution, the stop of the program.
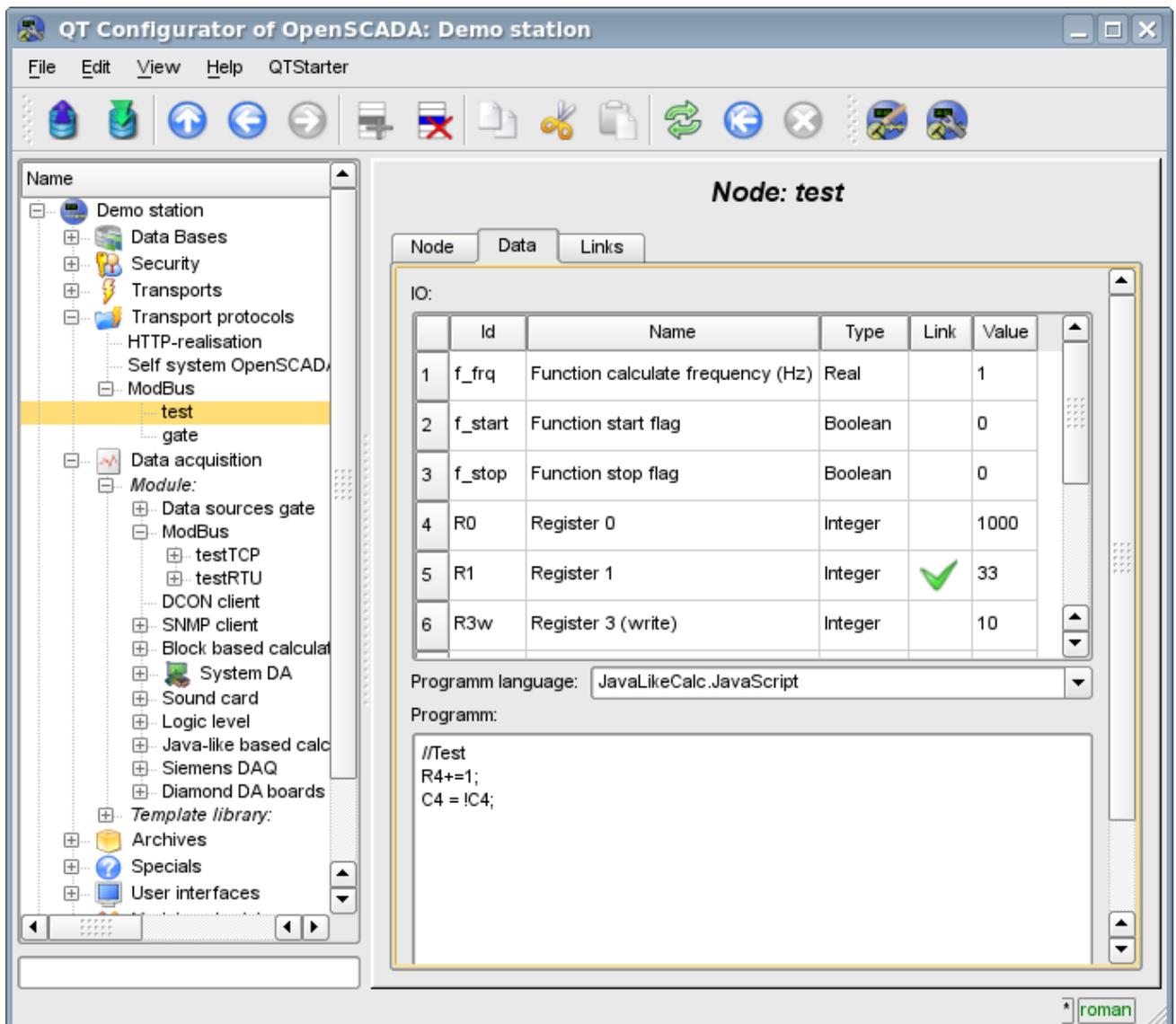
Fig.3. The tab "Data" of the configuration page of the node of the protocol in the "Data" mode.

For the parameter which are signed as links above it can be set the links only to switched off node of the protocol in the tab "Connections"(Figure 4).
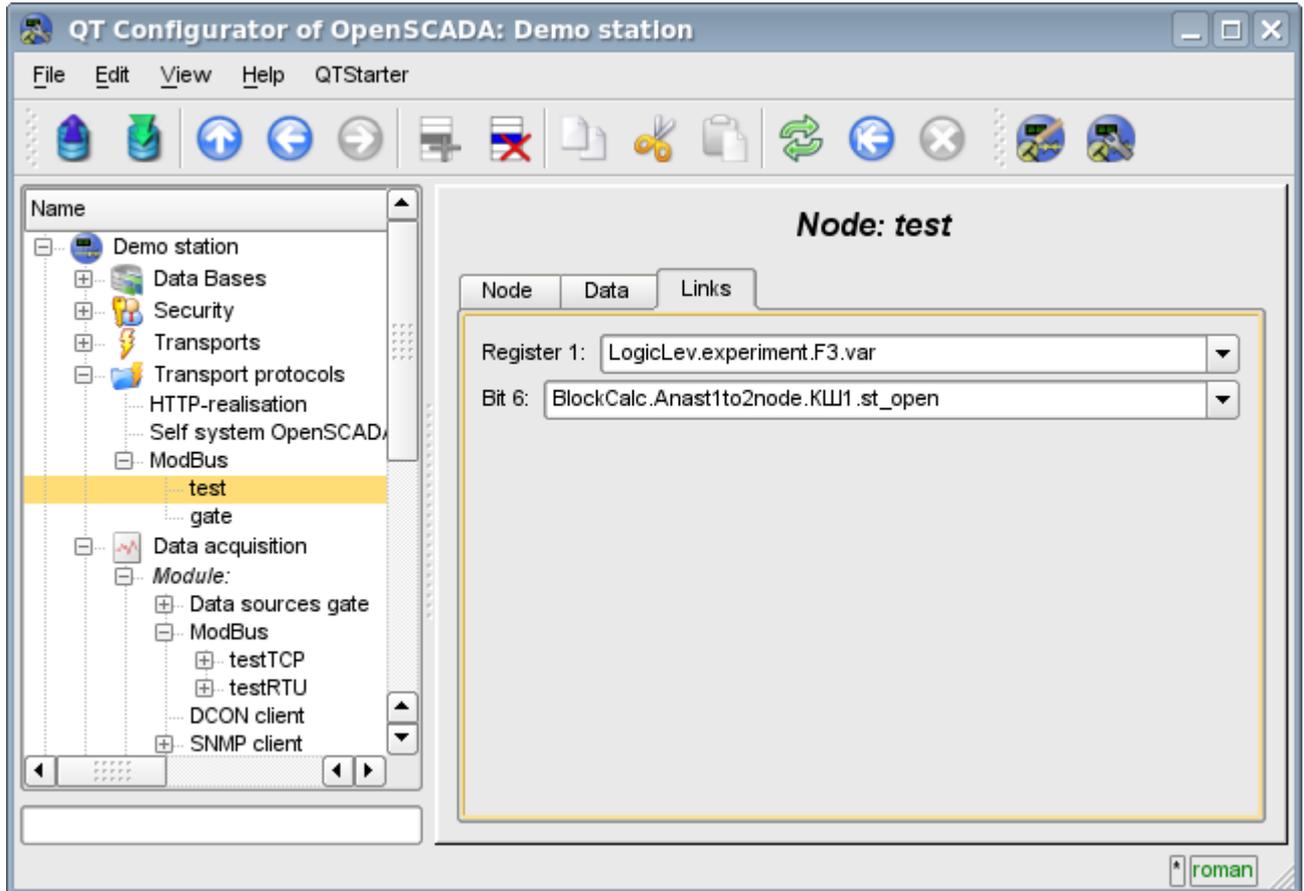
Fig.4. The tab "Links" of the configuration page of the node of the protocol in the "Data" mode.

The mode of the node of the protocol "Gateway of the node"

Mode is used to carryover the requests to a separate device in the other ModBus network to the ModBus network, in which this node is configured. The overall configuration of the node is made in the tab "Node"(Fig. 2) by the parameters:

- The state of the node, as follows: Status, «Enabled» and the name of the database containing the configuration.
- Id, name and description of the node.
- The state, in which the node must be translated at boot: «Enabled».
- Address of the node in the ModBus network from 1 to 247.
- Inbound traffic, to the network of which the node is belonged to. It is selected from the list of input transport of the subsystem "Transports" of OpenSCADA. Specifying as the transport the symbol "*" makes this node a participant of any ModBus network with the processing of requests from any transport.
- Variant of the ModBus protocol, requests in which must be processed by the node from the list: All, RTU, ASCII, TCP/IP.
- The choice of the mode, in this case the mode "Gateway of the node".
- Transport, in which the request must be redirected, from the list of outgoing transports of subsystem "Transports".
- Protocol in which to redirect the request.
- Address of the node of ModBus network from 1 to 247, in which the request is forwarded to.
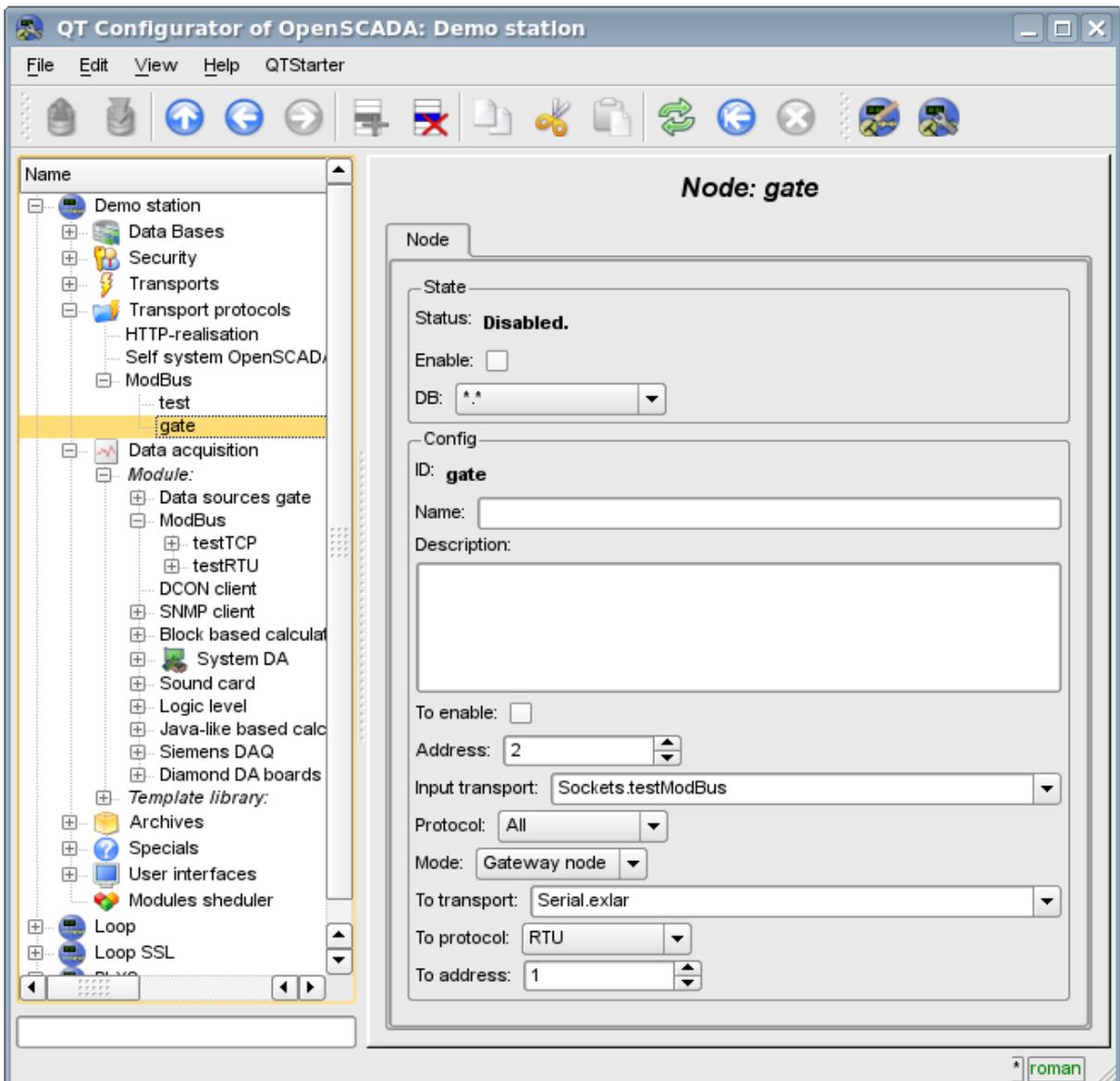
Fig.5. The tab "Node" of the configuration page of the node of the protocol in the "Gateway of the node" mode.

The mode of the node of the protocol "Gateway of the network"

Mode is used to carryover the requests of the network at whole to the other ModBus network from the ModBus network, in which this node is configured. The overall configuration of the node is made in the tab "Node"(Fig. 2) by the parameters:

• The state of the node, as follows: «Enabled» and the name of the database containing the configuration.
• Id, name and description of the node.
• The state, in which the node must be translated at boot: «Enabled».
• Incoming transport of the network, from which the requests are transfered.It is selected from the list of input transport of the subsystem "Transports" of OpenSCADA.
• Variant of the ModBus protocol, requests in which must be processed by the node from the list: All, RTU, ASCII, TCP/IP.
• The choice of the mode, in this case the mode "Gateway of the network".
• Transport, in which the request must be redirected, from the list of outgoing transports of subsystem "Transports".
• Protocol in which to redirect the request.
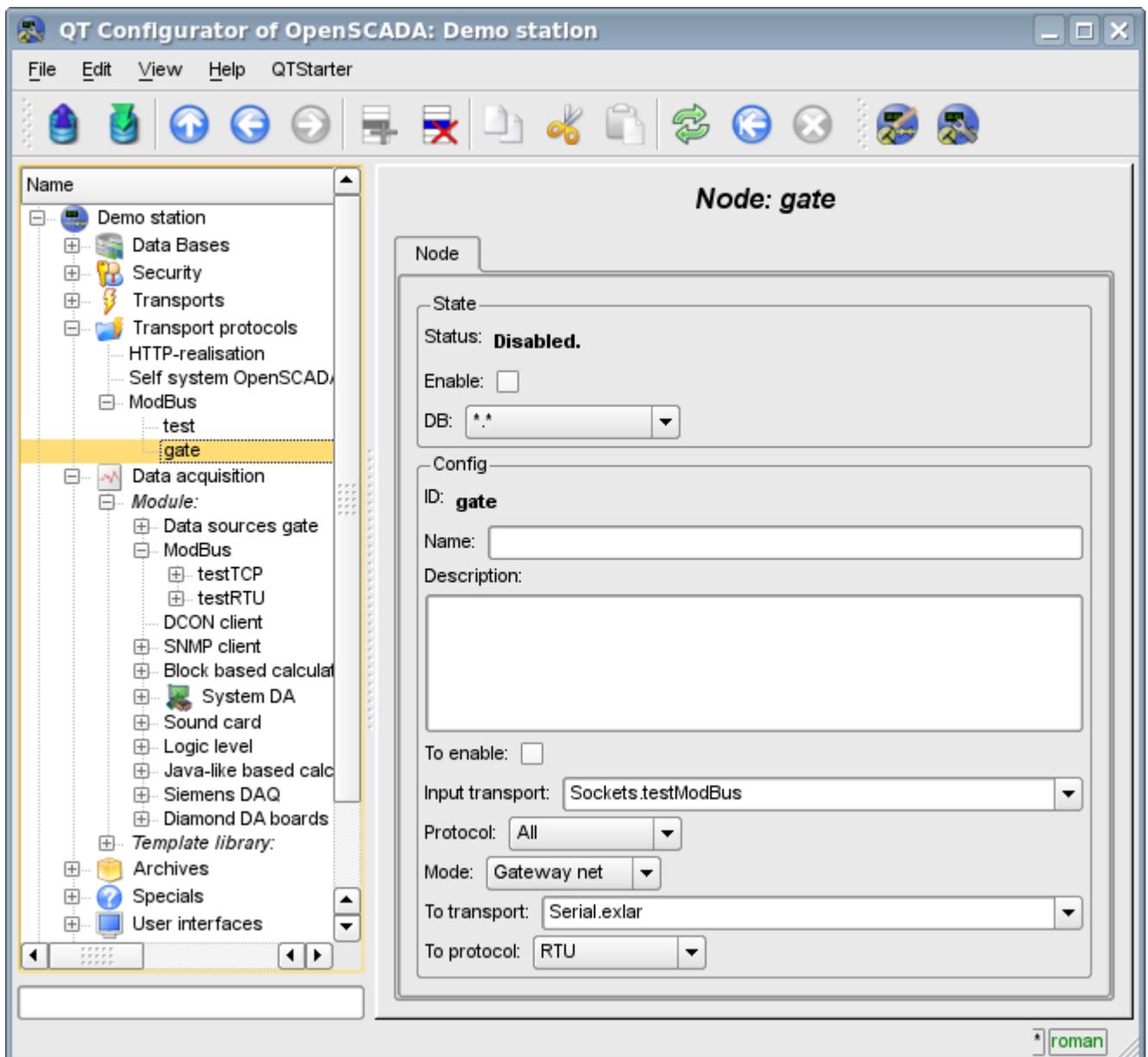
Fig.6. The tab "Node" of the configuration page of the node of the protocol in the "Gateway of the network" mode.

2.3 Report of the ModBus requests

To be able to monitor the correct implementation of requests to the various components the a module provides an opportunity to incorporate the report of the requests that pass through the protocol module. The report included by indication of non zero number of entries in the tab "Report" of the page of the module of the protocol(Fig.7).
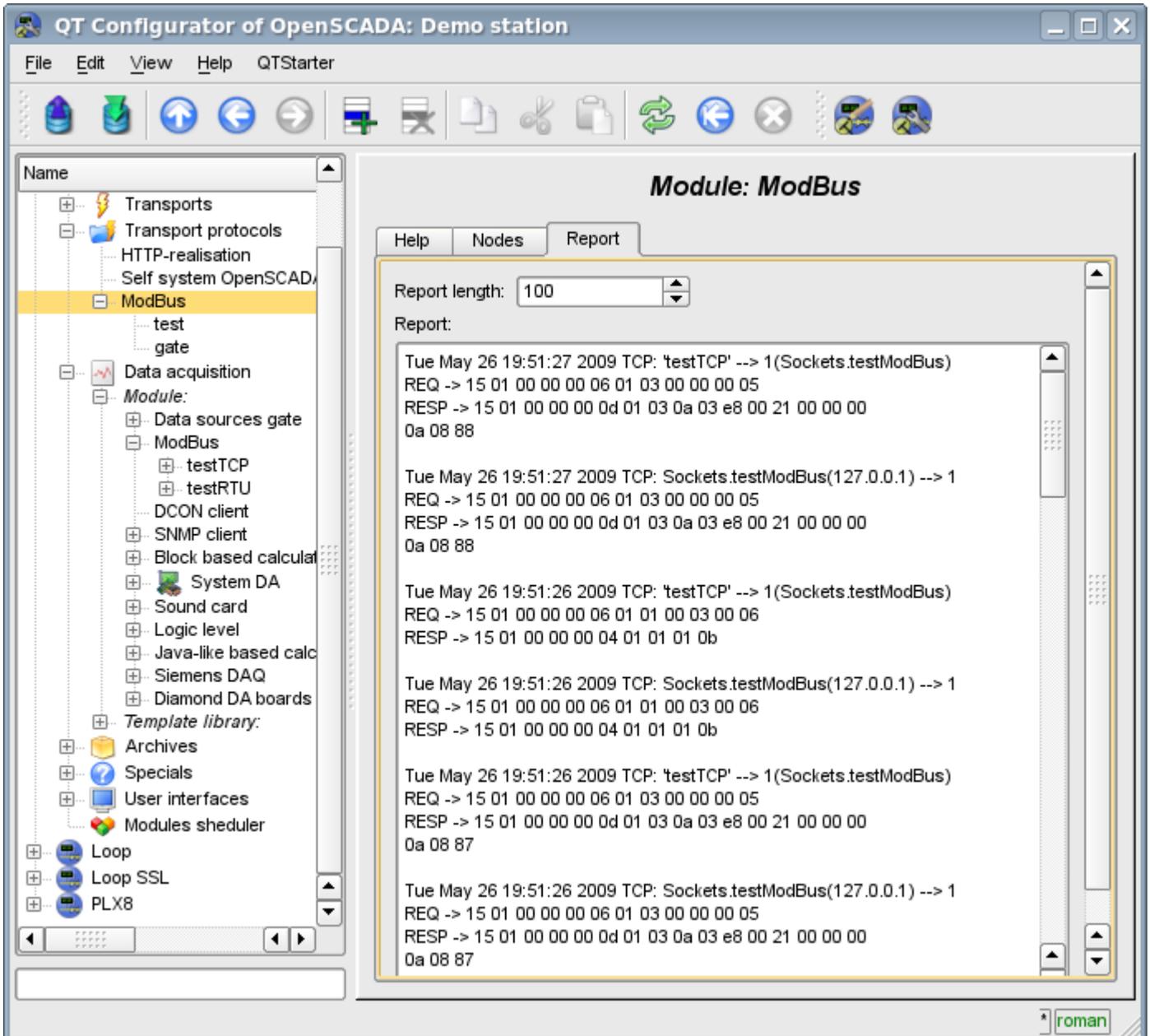

Fig.7. "Report" tab of the page of the module of the protocol.

3. Data acquisition module

Module of the data acquisition provides an opportunity to interrogate and write registers and bits of devices through protocol modes TCP, RTU, ASCII and commands of request 0x01 – 0x06.

3.1. Controller of data

For addition of a ModBus data source the controller is created and configured in the system OpenSCADA. Example of the configuration tab of the controller is depicted in Fig.8.
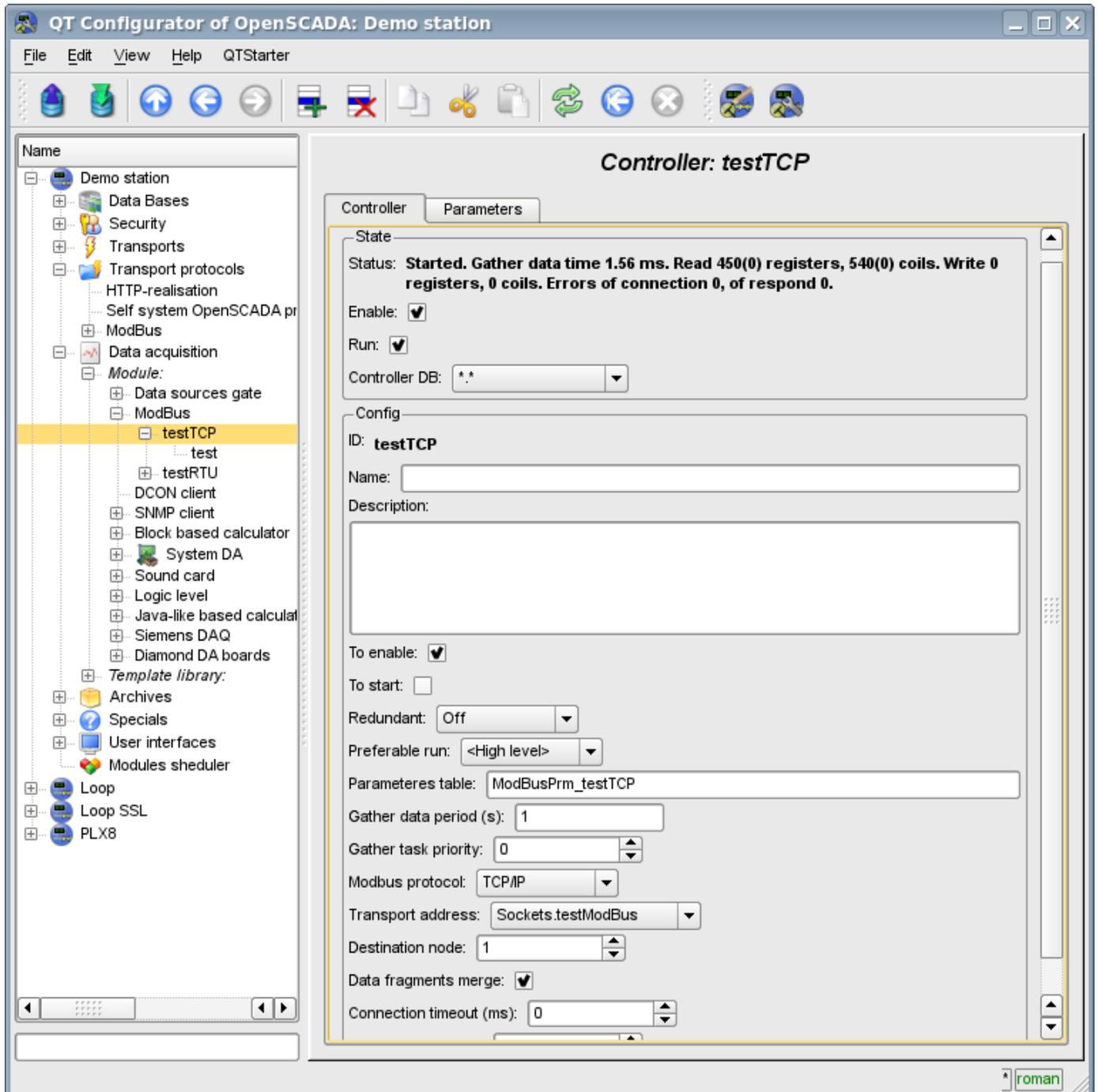
Fig.8. Configuration tab of the controller.

Using this tab you can set:
- The state of the controller, as follows: Status, «Enabled», «Running» 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: «Enabled» and «Running».
- Horizontal mode of redundancy and performance preference of the controller.

- Name of table to store the configuration of the parameters of the controller.
- The period and the priority of the task of data acquisition.
- ModBus протокол, используемый для опроса физического устройства (TCP/IP, RTU или ASCII).
- Address of outbound transport from the list of configured outbound transports in the subsystem "Transports" of OpenSCADA.
- ModBus destination node. In the case of protocols RTU and ASCII – this is the unique address of the physical device, and when TCP/IP – the identifier of the unity.
- Combining fragments of registers. Standard functions 01–04 let to request at once multiple adjacent registers or bits. This strategy often allows to optimize the traffic. However, the required registers are not always located adjacent to each other, this option allows you to collect them in blocks of up to 100 registers, or 1600 bits. The installing of this parameter must be approached with caution, since not all devices support access to registers between fragments.
- Connection timeout in milliseconds. Specifies the time interval during which the answer is expected. If there is zero waiting time by default the transport waiting time is used.
- Time of connection recovery in seconds. Specifies the time interval after which the re-attempt of the request to previously inaccessible device is done.
- Attempts of request for the protocols RTU and ASCII. Indicates the number of attempts by the repetition of the request in case of incomplete or damaged answer.

3.2. Parameters

Module of data acquisition provides only one type of parameters – "Standard". An additional configuration parameter field of the module(Fig.9) is a list of processed attributes(registers ModBus). Attribute in the list is written as follows: **<dt>:<numb>:<wr>:<id>:<name>**.

Where:

dt — Type of ModBus data (R-register, C-bit, RI- input register, CI-input bit);

numb — number of reguster or bit of ModBus device (decimal, octal or hexadecimal);

wr — read-write mode (r-read, w-write, rw-read and write);

id — ID of the attribute OpenSCADA;

name — name of the attribute OpenSCADA.
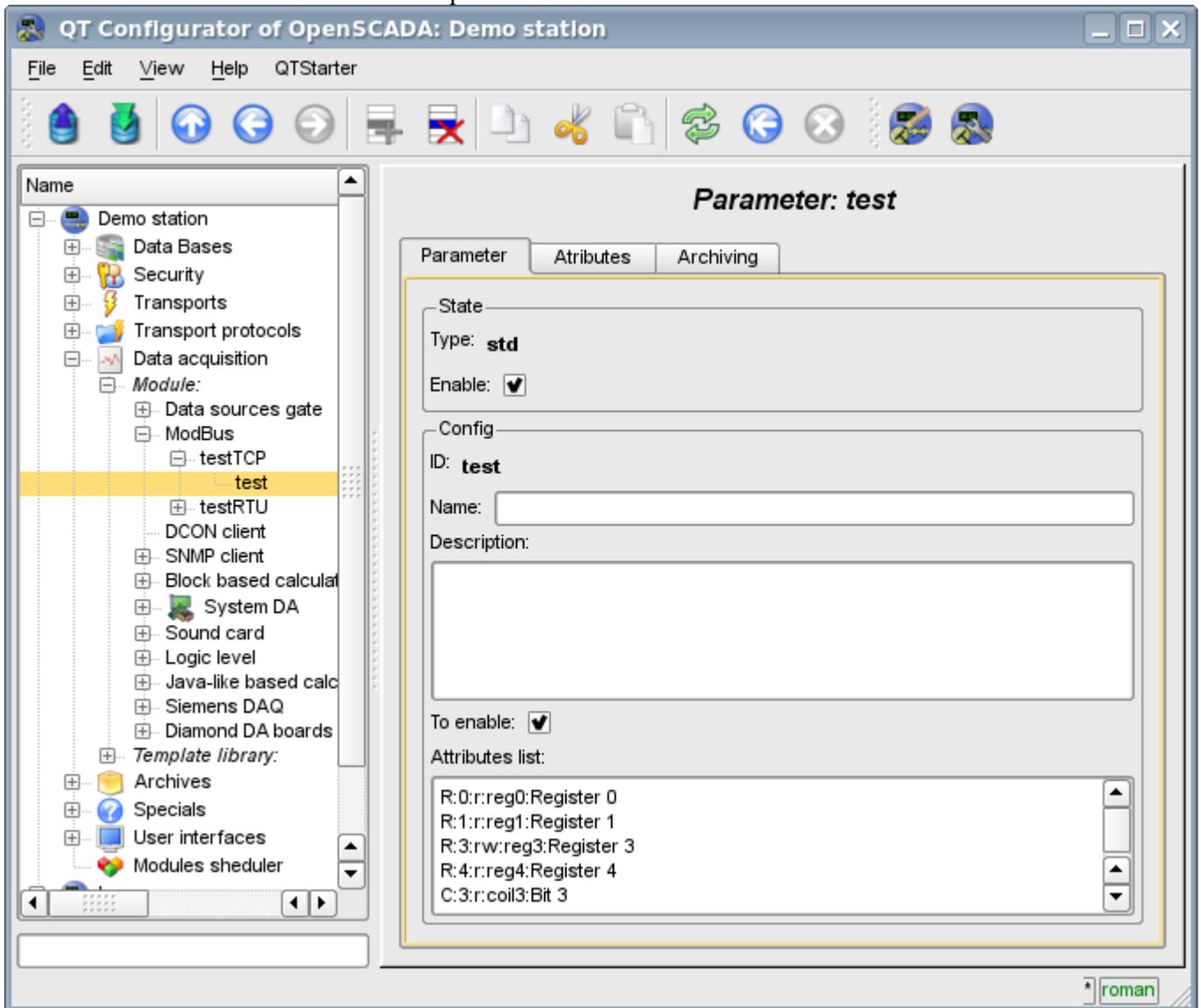
Fig.9. Configuration tab of the parameter.

In accordance with a specified list of attributes interrogation and the creation of the attributes of the parameter is carried out(Figure 10).
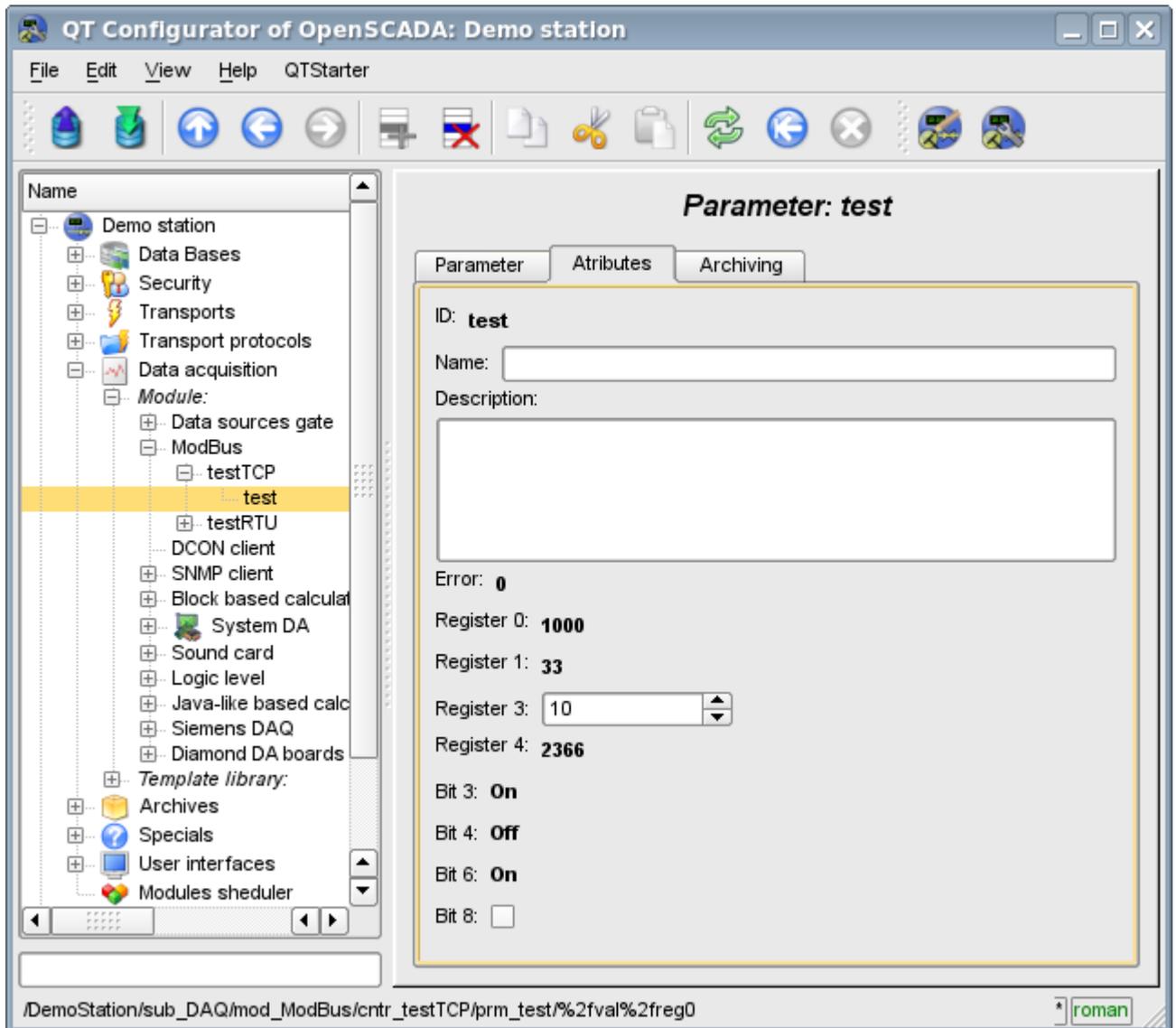
Fig.10. Tab of the attributes of the parameter.