

Модули <ModBus> подсистемы “Сбор данных” и подсистемы “Транспортные протоколы”

Параметр	Модуль 1	Модуль 2
<i>ID:</i>	ModBus	
<i>Имя:</i>	ModBus	
<i>Тип:</i>	DAQ	Протокол
<i>Источник:</i>	daq_ModBus.so	
<i>Версия:</i>	1.2.0	0.6.3
<i>Автор:</i>	Роман Савоченко	
<i>Описание:</i>	Предоставляет реализацию клиентского сервиса протокола ModBus. Поддерживаются Modbus/TCP, Modbus/RTU и Modbus/ASCII протоколы.	Предоставляет реализацию протоколов ModBus. Поддерживаются Modbus/TCP, Modbus/RTU и Modbus/ASCII протоколы.
<i>Лицензия:</i>	GPL	

Оглавление

Модули <ModBus> подсистемы “Сбор данных” и подсистемы “Транспортные протоколы”	1
Введение	2
1. Общее описание протокола ModBus	3
1.1. Адресация	3
1.2. Стандартные коды функций	3
2. Модуль реализации протокола	4
2.1. API функции исходящих запросов	4
2.2. Обслуживание запросов по протоколу ModBus	5
2.3 Отчёт запросов ModBus	12
3. Модуль сбора данных	13
3.1. Контроллер данных	13
3.2. Параметры	15
3.3. API пользовательского программирования	20

Введение

ModBus — коммуникационный протокол, основанный на клиент-серверной архитектуре. Разработан фирмой Modicon для использования в контроллерах с программируемой логикой (PLC). Стал стандартом де-факто в промышленности и широко применяется для организации связи промышленного электронного оборудования. Использует для передачи данных через последовательные линии связи RS-485, RS-422, RS-232, а также сети TCP/IP. В настоящее время поддерживается некоммерческой организацией ModBus-IDA.

Существуют три режима протокола: ModBus/RTU, ModBus/ASCII и ModBus/TCP. Первые два используют последовательные линии связи (в основном RS-485, реже RS-422/RS-232), последний использует для передачи данных сети TCP/IP.

Модуль сбора данных предоставляет возможность собирать информацию у различных устройств по протоколу ModBus во всех режимах. Также модулем реализуются функции горизонтального резервирования, а именно совместной работы с удалённой станцией этого же уровня. В то же время модуль протокола позволяет сформировать и выдать данные по протоколу ModBus в различных режимах и через интерфейсы, поддерживаемые модулями подсистемы "Транспорты".

1. Общее описание протокола ModBus

Протокол ModBus/RTU предполагает одно ведущее (запрашивающее) устройство в линии (master), которое может передавать команды одному или нескольким ведомым устройствам (slave), обращаясь к ним по уникальному в линии адресу. Синтаксис команд протокола позволяет адресовать 247 устройств на одной линии связи стандарта RS-485 (реже RS-422 или RS-232). В случае с режимом TCP адресация исключена из протокола, поскольку выполняется на уровне TCP/IP стека.

Инициатива проведения обмена всегда исходит от ведущего устройства. Ведомые устройства прослушивают линию связи. Мастер подаёт запрос (посылка, последовательность байт) в линию и переходит в состояние прослушивания линии связи. Ведомое устройство отвечает на запрос, пришедший в его адрес.

Окончание ответной посылки определяется в зависимости от режима. В режиме RTU окончание посылки определяется по временному интервалу между окончанием приёма предыдущего байта и началом приёма следующего, время символа. Если этот интервал превысил время, необходимое для приёма полтора байта на заданной скорости передачи то приём фрейма ответа считается завершённым. В режиме ASCII критерием окончания посылки является символ '\r', а в режиме TCP — ожидаемый размер посылки, информация о котором присутствует в заголовке пакета.

1.1. Адресация

Все операции с данными привязаны к нулю, каждый вид данных (регистр, бит, регистр входа или бита входа) начинаются с адреса 0000 и заканчиваются 65535.

1.2. Стандартные коды функций

В протоколе ModBus можно выделить несколько подмножеств команд (Таблица 1).

Таблица 1: Подмножество команд протокола ModBus

Подмножество	Диапазон кодов
Стандартные	1–21
Резерв для расширенных функций	22–64
Пользовательские	65–119
Резерв для внутренних нужд	120–255

Модулем сбора данных используются команды 0x03 и 0x06(0x10) для чтения и записи регистров, 0x01 и 0x05(0x0F) для чтения и записи битов, 0x02 и 0x04 для чтения бита и регистра входа соответственно.

Модуль протокола обрабатывает запросы командами 0x03 и 0x06(0x10) для чтения и записи регистров, 0x01 и 0x05(0x0F) для чтения и записи битов.

2. Модуль реализации протокола

Модуль протокола ModBus содержит код реализации протокольной части ModBus, а именно особенности вариантов протоколов ModBus/TCP, ModBus/RTU и ModBus/ASCII. Модуль протокола, совместно с выбранным транспортом, активно используется модулем сбора данных для осуществления непосредственных запросов. Поскольку модуль протокола является автономным то, используя его, можно создавать дополнительные модули сбора данных посредством нестандартных функций расширения ModBus различного оборудования автоматизации.

2.1. API функции исходящих запросов

API функции исходящих запросов оперируют обменом блоками PDU, завернутыми в XML-пакеты со следующей структурой:

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

Где:

- *prt* — имя тега запроса с названием используемого варианта протокола (TCP, RTU или ASCII).
- *sId* — идентификатор источника запроса. Используется для помещения в протокол выходного протокола.
- *reqTm* — время запроса, а именно время, в течение которого ожидать ответа, в миллисекундах.
- *node* — номер узла назначения или идентификатор юнита ModBus/TCP.
- *reqTry* — количество попыток повторения запроса с ошибкой в ответе. Только для вариантов ModBus/RTU и ModBus/ASCII.
- *pdu* — непосредственно блок юнита данных протокола (PDU) ModBus.

Результирующий pdu заменяет pdu запроса в XML-пакете, а также устанавливается атрибут "err" с кодом и текстом ошибки, если таковая имела место.

2.2. Обслуживание запросов по протоколу ModBus

Входная часть обслуживания запросов к модулю протокола осуществляет проверку и обработку запросов посредством объектов узлов, предусмотренных модулем (рис.1). Фактически реализуется механизм, позволяющий выполнять системой OpenSCADA роль сервера ModBus/TCP или подчинённого устройства ModBus/RTU и ModBus/ASCII. Таким образом система OpenSCADA получает возможность использоваться в роли любого участника сетей ModBus.

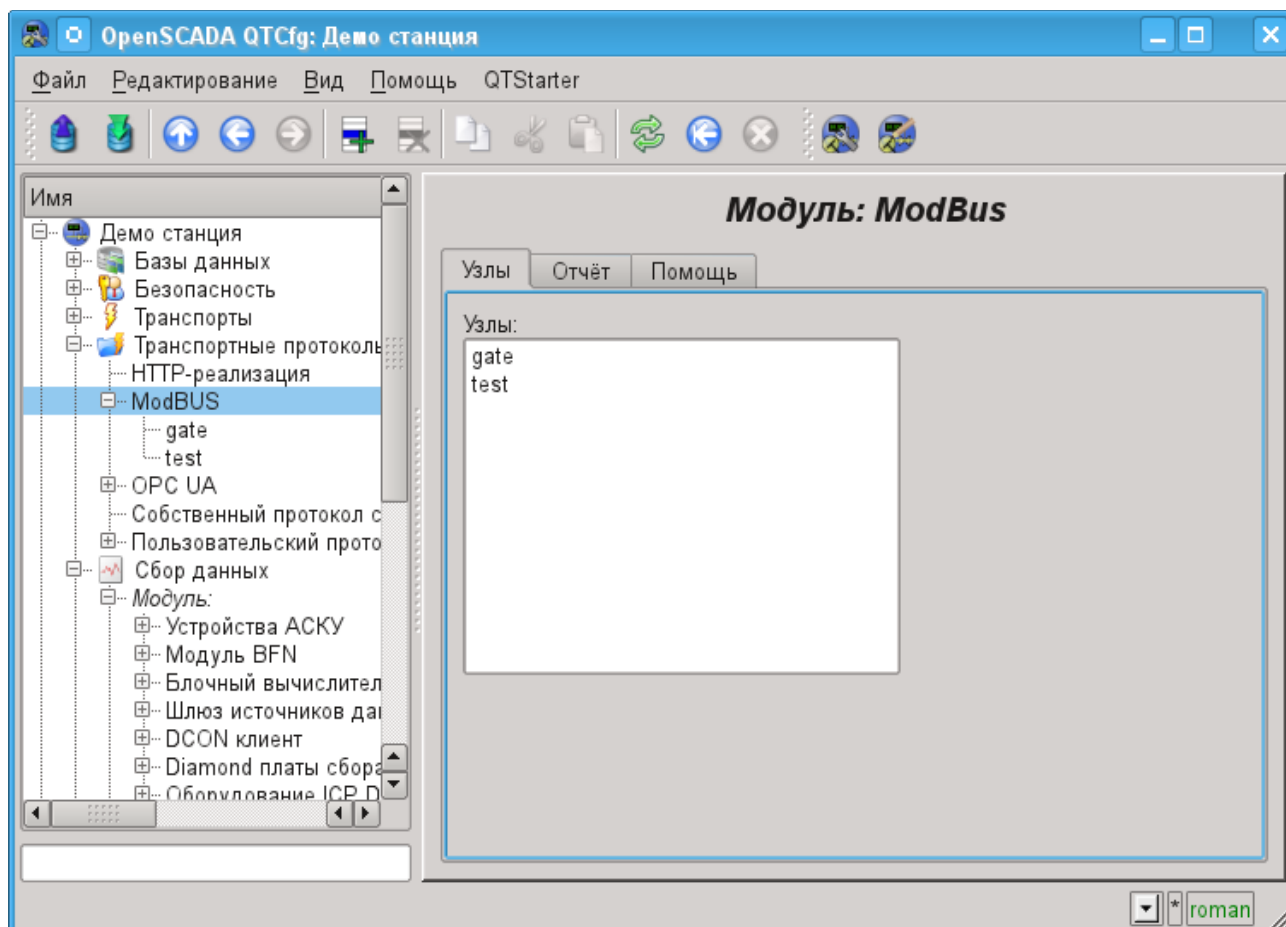


Рис.1. Вкладка перечня узлов обслуживания входящих запросов протокола.

Узел протокола эквивалентен физическому узлу устройства сети ModBus. Узел протокола может работать в трёх режимах:

- "Данные" — режим отражения данных системы OpenSCADA на массивы регистров и битов ModBus с передачей их по запросу клиентского узла или мастера.
- "Шлюз узла" — режим перенаправления (шлюзования) запросов к узлу в другой сети ModBus через данный узел.
- "Шлюз сети" — режим перенаправления запросов к любому узлу в другую сеть ModBus, фактически выполняя интеграцию нескольких сетей ModBus в одну.

Поскольку узлов протокола может быть создано множество, то получается, что на одном интерфейсе, т.е. в одной сети, одна станция на основе OpenSCADA может прозрачно представлять несколько узлов сети ModBus с различными данными.

Рассмотрим особенности конфигурации узла протокола в различных режимах.

Режим узла протокола «Данные»

Режим используется для отражения данных системы OpenSCADA на массивы регистров и битов ModBus. Общая конфигурация узла осуществляется во вкладке "Узел" (рис.2) параметрами:

- Состояние узла, а именно: статус, "Включен" и имя БД, содержащей конфигурацию.
- Идентификатор, имя и описание узла.
- Состояние, в которое переводить узел при загрузке: "Включен".
- Адрес узла сети ModBus от 1 до 247.
- Входящий транспорт, к сети которого относится узел. Выбирается из перечня входных транспортов подсистемы "Транспорты" OpenSCADA. Указание в качестве транспорта символа "*" делает данный узел участником любой сети ModBus с обработкой запросов от любого транспорта.
- Вариант протокола ModBus, запросы в котором должен обрабатывать узел из списка: Все, RTU, ASCII, TCP/IP.
- Выбор режима, в данном случае этот режим "Данные".
- Период обчёта данных в секундах. Указывает периодичность обработки формируемых для запросов данных, а именно таблицы данных ModBus, программы обчёта данных и обслуживание ссылок на данные OpenSCADA.

Узлом в этом режиме обрабатываются следующие стандартные команды протокола ModBus:

- *0x01* — чтение группы битов;
- *0x03* — чтение группы регистров;
- *0x05* — установка одного бита;
- *0x06* — установка одного регистра;
- *0x0F* — установка группы битов;
- *0x10* — установка группы регистров.

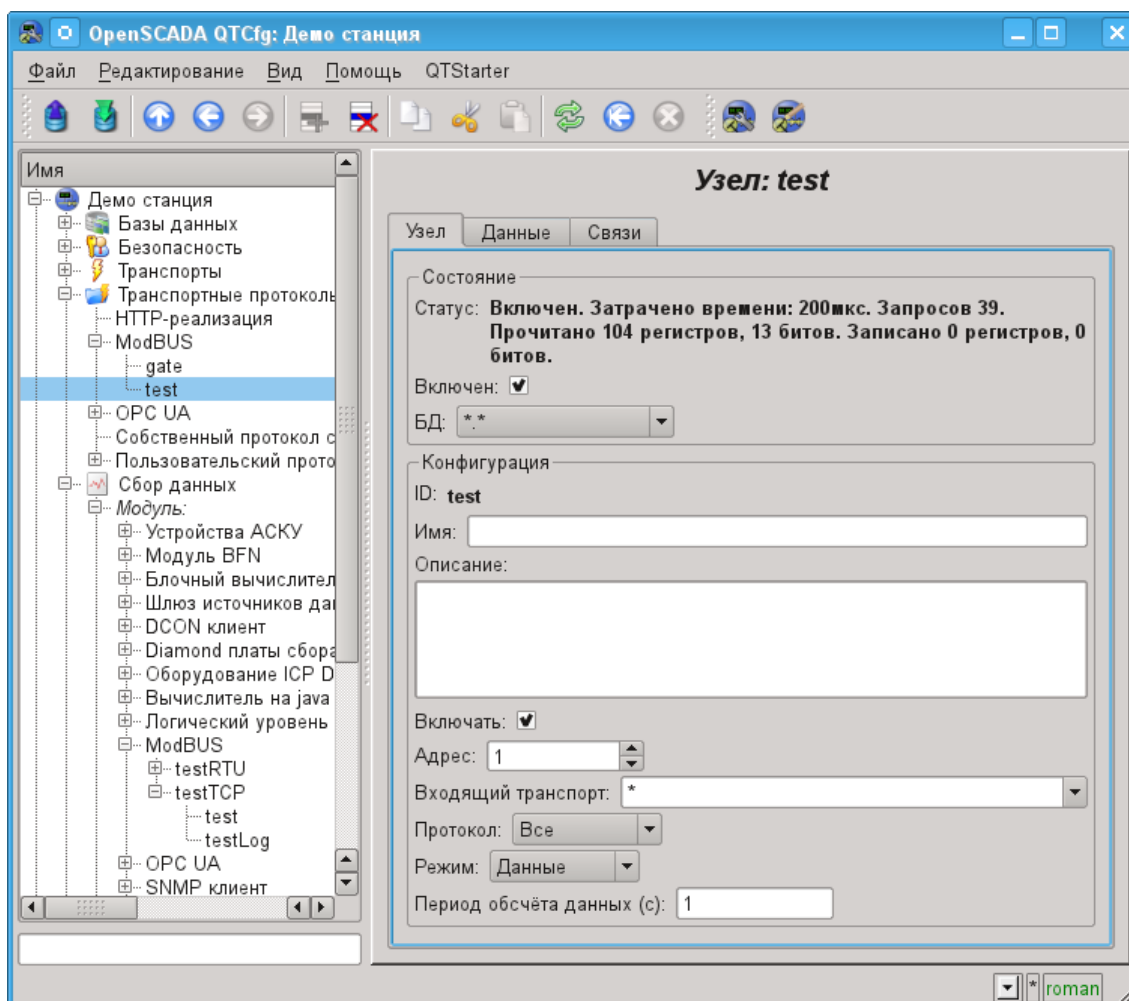


Рис.2. Вкладка «Узел» страницы конфигурации узла протокола в режиме «Данные».

Для формирования таблицы отражения данных сети ModBus, а именно регистров и битов предусматривается вкладка "Данные" (рис.3). Вкладка "Данные" содержит таблицу параметров и программу обработки параметров с указанным языком программирования, доступным в системе OpenSCADA. Таблица содержит параметры со свойствами:

- *Id* — идентификатор параметра. Является ключевым для формирования таблиц регистров и битов ModBus. Для указания того, что данный параметр является регистром ModBus, необходимо идентификатор записать в виде "R[N]w", где N — число номера регистра от 0 до 65535, а w — необязательный символ, указывающий на возможность установки его запросом ModBus, например: R23, R456, R239w. Для указания бита ModBus необходимо идентификатор записать в виде "C[N]w", где N — число номера бита от 0 до 65535, а w — необязательный символ, указывающий на возможность установки его запросом ModBus, например: C437, C0, C39w. Все остальные параметры, не попавшие под вышеуказанные правила, являются внутренними и используются для различных промежуточных вычислений, обработки и преобразований.
- *Имя* — Имя параметра, используется для именования связи.
- *Тип* — Тип параметра из списка: "Вещественный", "Целый", "Логический" и "Строка". Для регистров и битов ModBus имеет смысл устанавливать "Целый" и "Логический" тип соответственно.
- *Связь* — Признак того, что данный параметр должен связываться с атрибутом параметра подсистемы "Сбор данных". Указанные этим флагом связи устанавливаются во вкладке "Связи".
- *Значение* — Исходное или текущее, если узел включен, значение параметра.

В таблице по умолчанию определяются несколько параметров специального значения:

- *f_freq* — частота вычисления таблицы программой;
- *f_start* — признак первого исполнения, запуска, программы.
- *f_stop* — признак последнего исполнения, останова, программы.

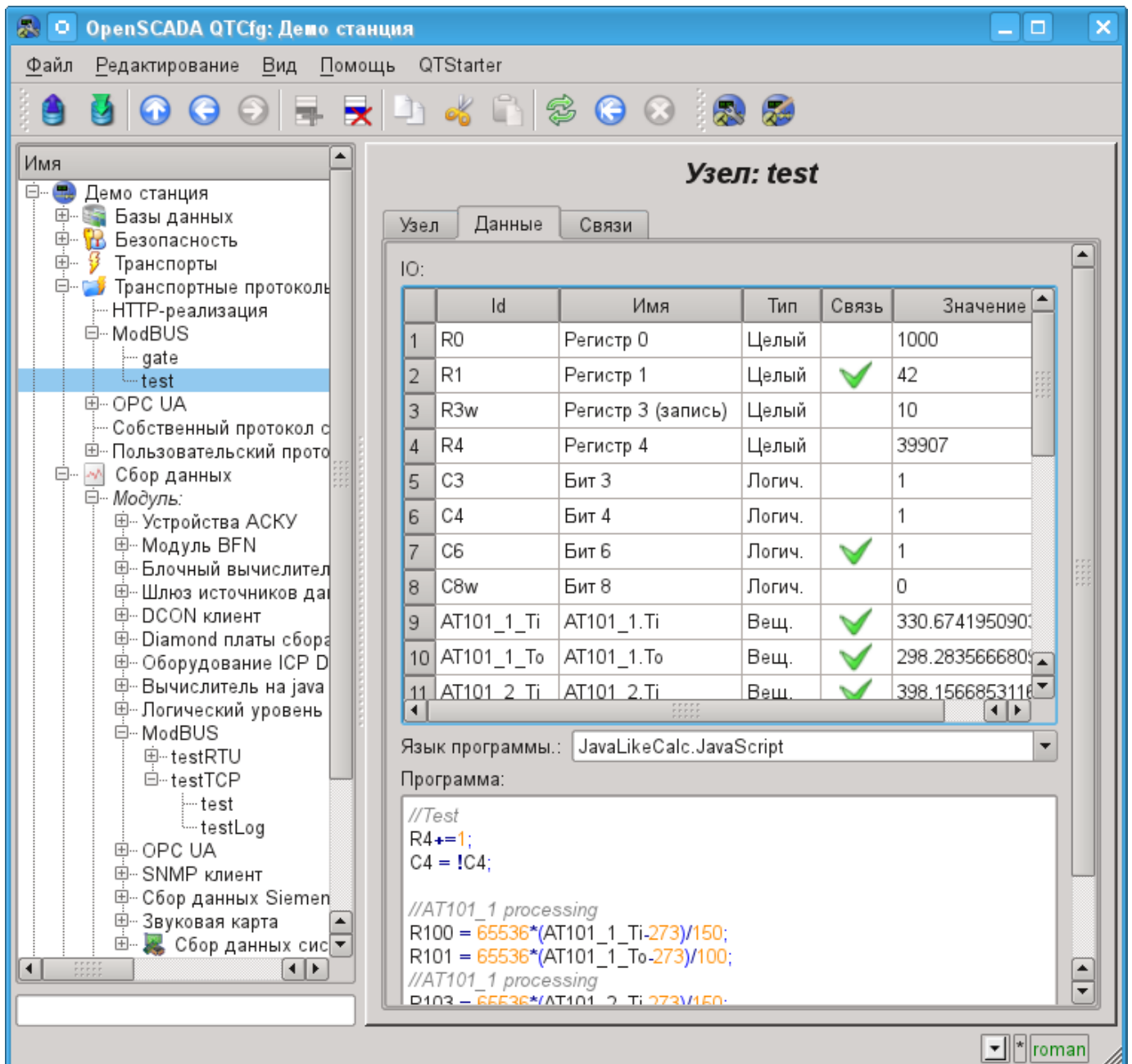


Рис.3. Вкладка «Данные» страницы конфигурации узла протокола в режиме «Данные».

Для указанных в качестве связей параметров можно установить связи только для выключенного узла протокола во вкладке "Связи" (рис.4).

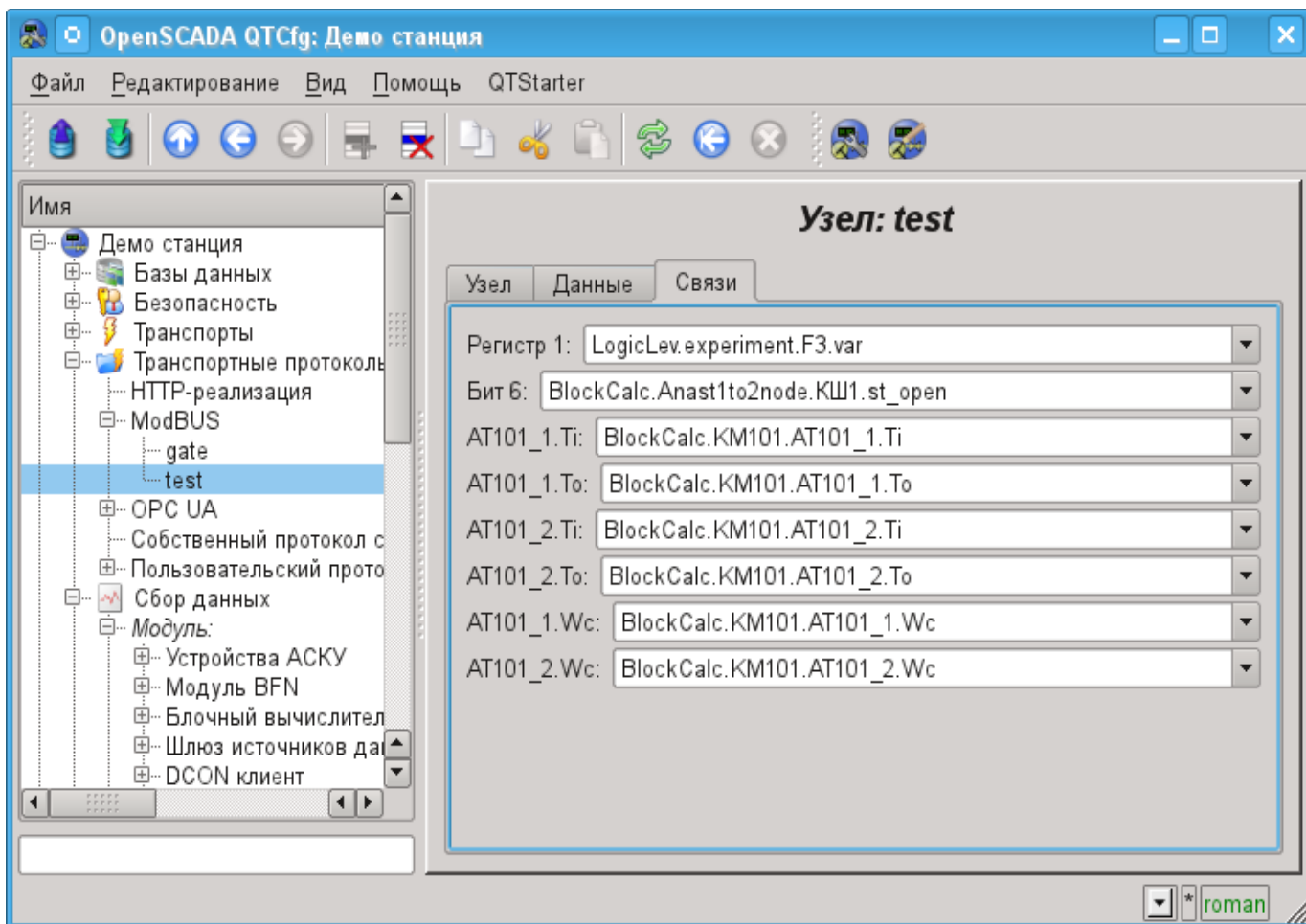


Рис.4. Вкладка «Связи» страницы конфигурации узла протокола в режиме «Данные».

Режим узла протокола «Шлюз узла»

Режим используется для проброса запросов к отдельному устройству в другой сети ModBus из сети ModBus, для которой сконфигурирован данный узел. Общая конфигурация узла осуществляется во вкладке "Узел" (рис.5) параметрами:

- Состояние узла, а именно: статус, "Включен" и имя БД, содержащей конфигурацию.
- Идентификатор, имя и описание узла.
- Состояние, в которое переводить узел при загрузке: "Включен".
- Адрес узла исходной сети ModBus от 1 до 247.
- Входящий транспорт, к сети которого относится узел. Выбирается из перечня входных транспортов подсистемы "Транспорты" OpenSCADA. Указание в качестве транспорта символа "*" делает данный узел участником любой сети ModBus с обработкой запросов от любого транспорта.
- Вариант протокола ModBus запросы в котором должен обрабатывать узел из списка: Все, RTU, ASCII, TCP/IP.
- Выбор режима, в данном случае этот режим "Шлюз узла".
- Транспорт, в который перенаправлять запрос, из перечня исходящих транспортов подсистемы "Транспорты".
- Протокол в котором перенаправлять запрос.
- Адрес узла сети ModBus, от 1 до 247, в которую перенаправляется запрос.

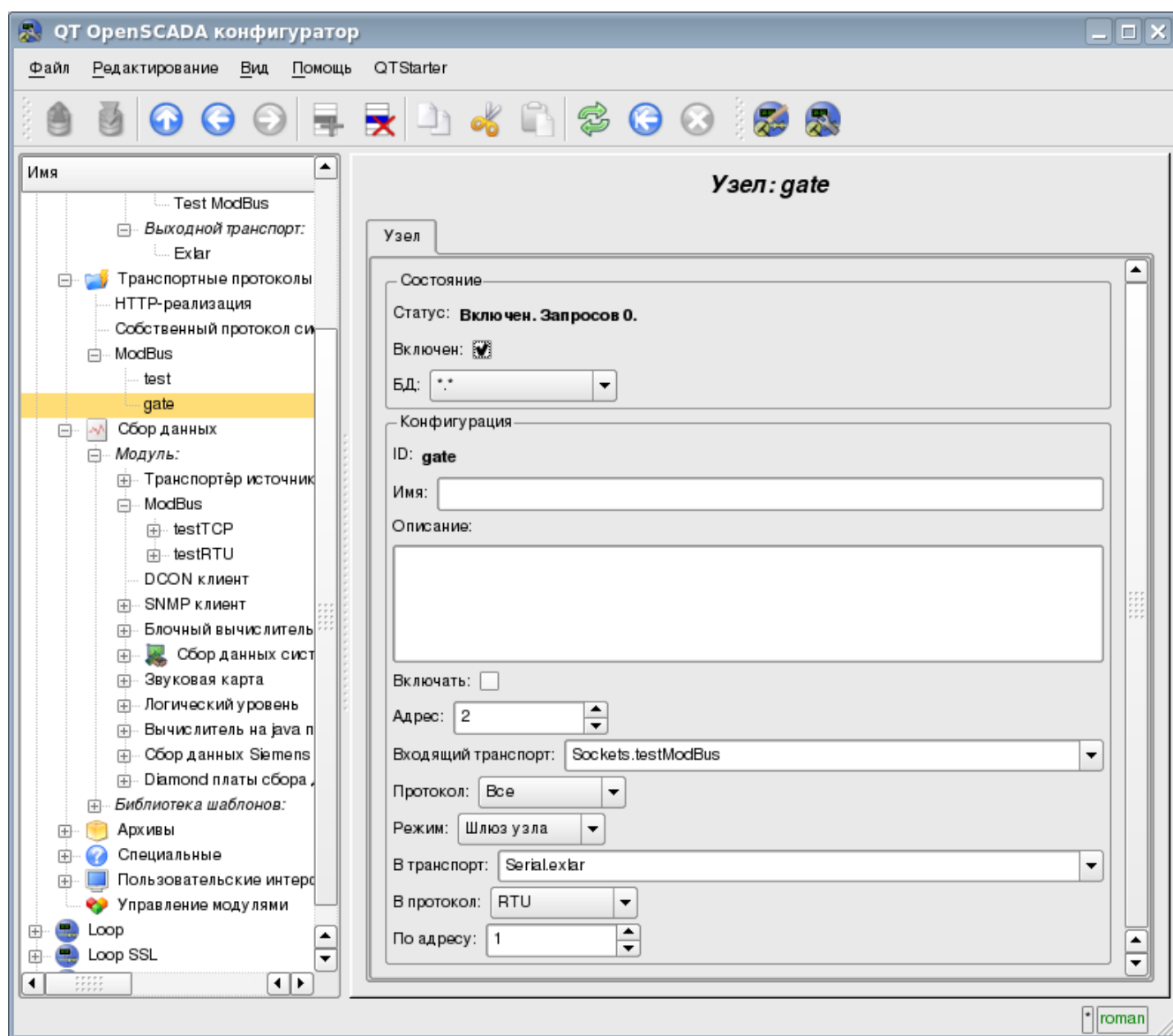


Рис.5. Вкладка «Узел» страницы конфигурации узла протокола в режиме «Шлюз узла».

Режим узла протокола «Шлюз сети»

Режим используется для проброса запросов сети целиком в другую сеть ModBus из сети ModBus для которой сконфигурирован данный узел протокола. Т.е. запрос на устройство с любым адресом будет направляться в другую сеть, без переадресовки. Общая конфигурация узла протокола осуществляется во вкладке "Узел" (рис.6) параметрами:

- Состояние узла, а именно: статус, "Включен" и имя БД, содержащей конфигурацию.
- Идентификатор, имя и описание узла.
- Состояние, в которое переводить узел при загрузке: "Включен".
- Входящий транспорт сети, из которой пробрасываются запросы. Выбирается из перечня входных транспортов подсистемы "Транспорты" OpenSCADA.
- Вариант протокола ModBus, запросы в котором должен обрабатывать узел из списка: Все, RTU, ASCII, TCP/IP.
- Выбор режима, в данном случае этот режим "Шлюз сети".
- Транспорт сети, в которую перенаправлять запрос, из перечня исходящих транспортов подсистемы "Транспорты".
- Протокол в котором перенаправлять запрос.

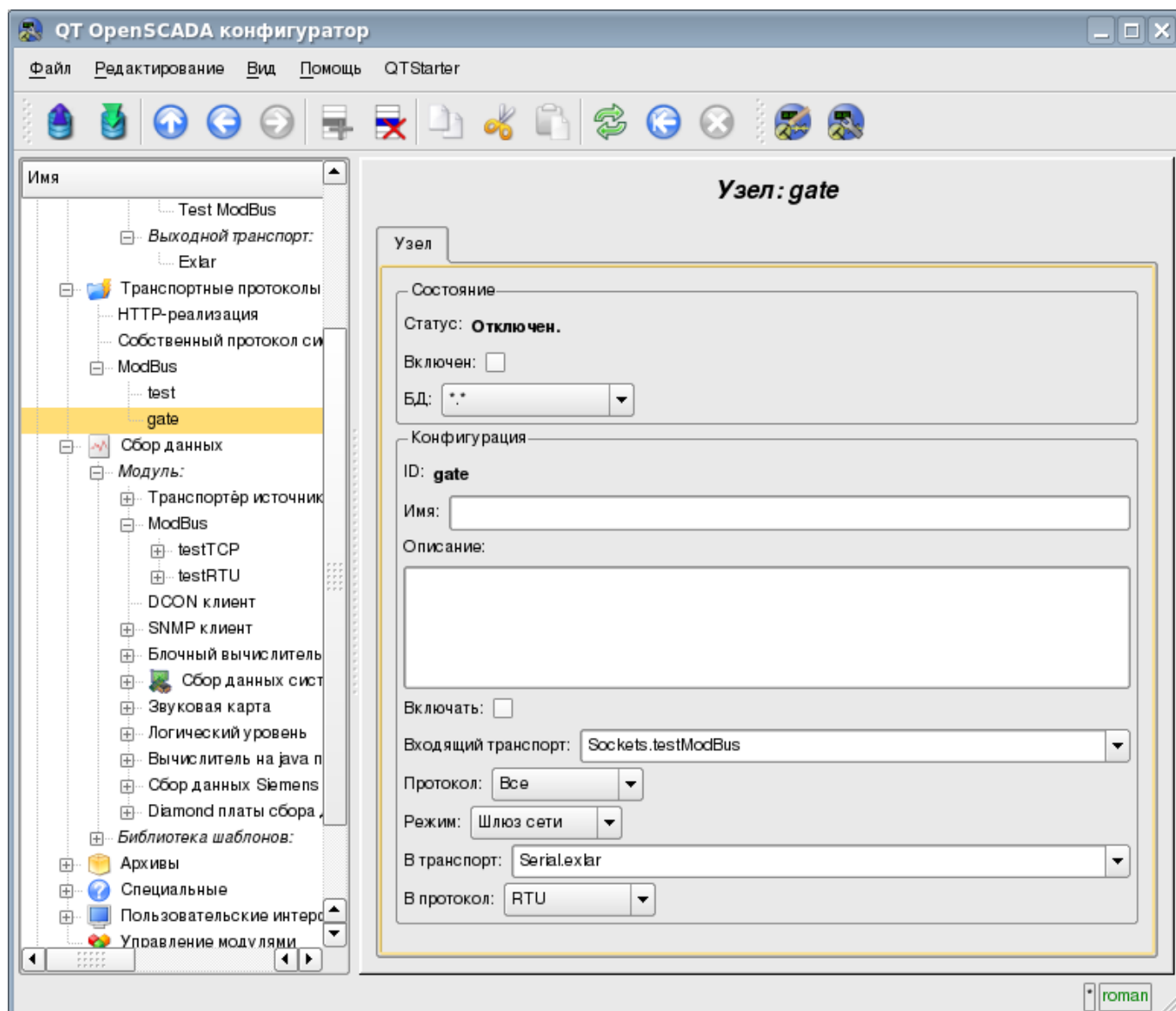


Рис.6. Вкладка «Узел» страницы конфигурации узла протокола в режиме «Шлюз сети».

2.3 Отчёт запросов ModBus

Для возможности контроля и диагностики за корректностью осуществления запросов различными компонентами модулем предоставляется возможность включения отчёта запросов, проходящих через модуль протокола. Отчёт включается указанием ненулевого количества записей во вкладке "Отчёт" страницы модуля протокола (рис.7).

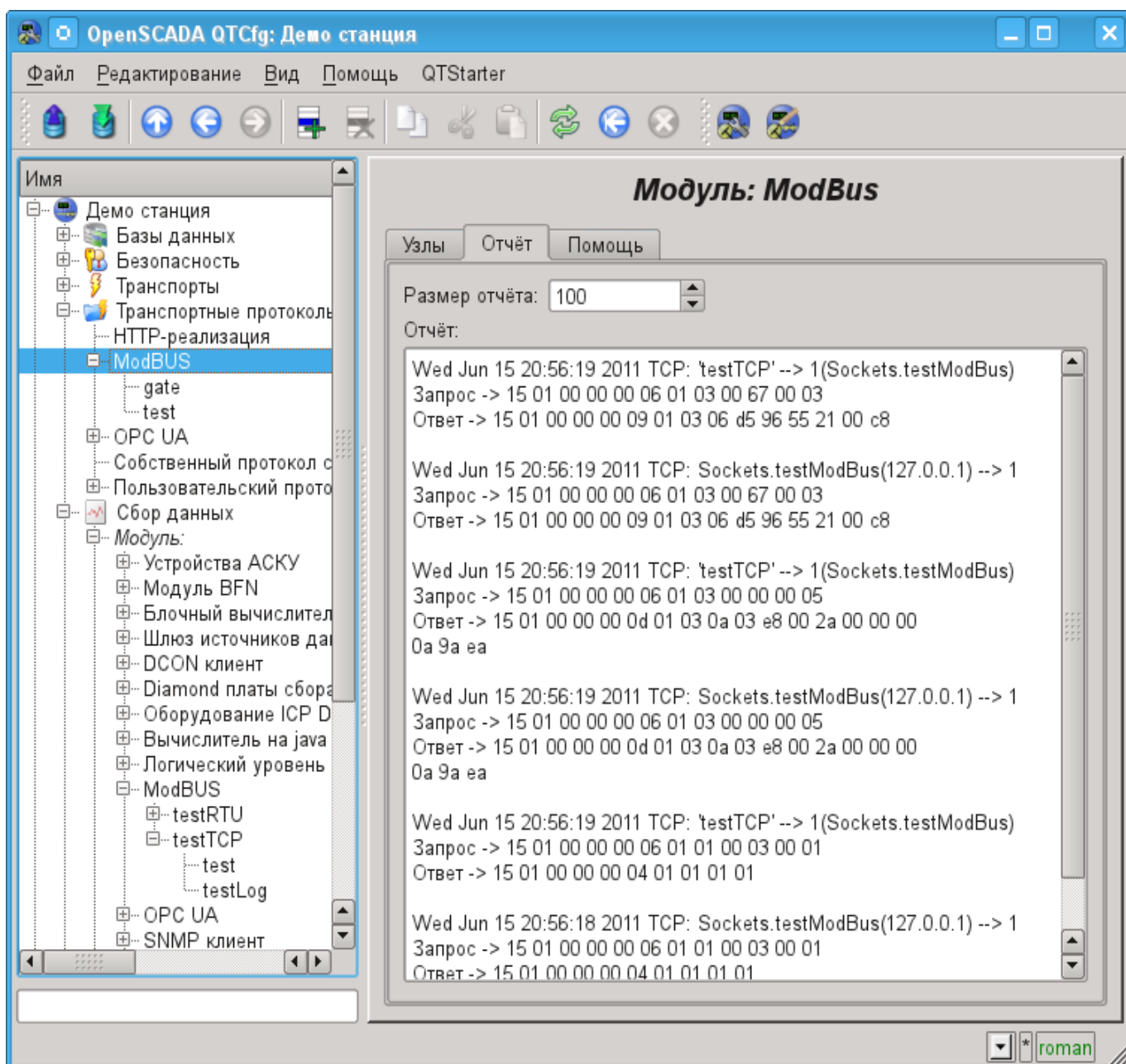


Рис.7. Вкладка «Отчёт» страницы модуля протокола.

3. Модуль сбора данных

Модуль сбора данных предоставляет возможность опроса и записи регистров и битов устройств посредством режима протоколов TCP, RTU, ASCII и команд запроса 0x01 - 0x06, 0x0F, 0x10.

3.1. Контроллер данных

Для добавления источника данных ModBus создаётся и конфигурируется контроллер в системе OpenSCADA. Пример вкладки конфигурации контроллера данного типа изображен на рис.8.

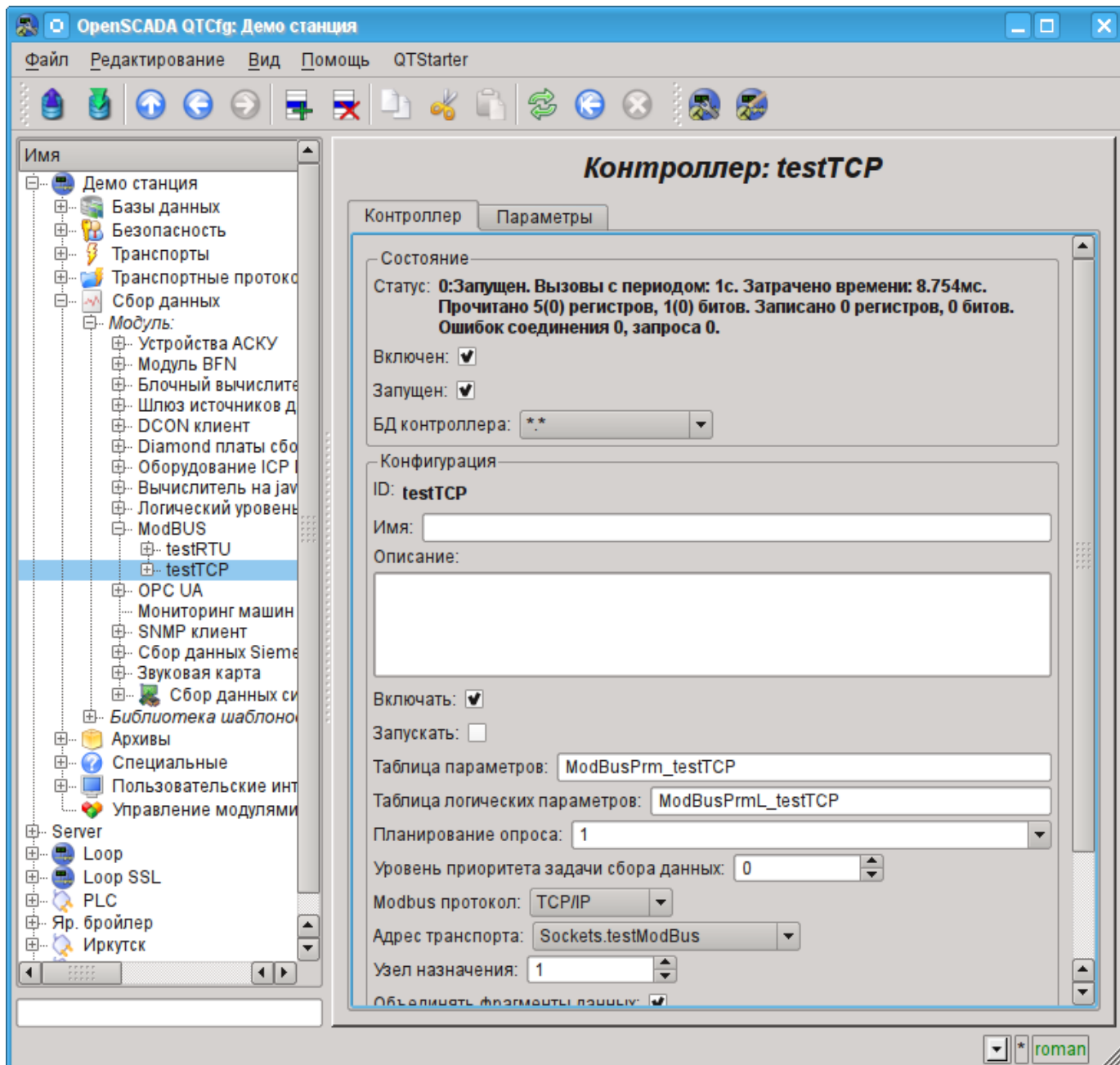


Рис.8. Вкладка конфигурации контроллера.

С помощью этой вкладки можно установить:

- Состояние контроллера, а именно: Статус, "Включен", Запущен" и имя БД, содержащей конфигурацию.
- Идентификатор, имя и описание контроллера.
- Состояние, в которое переводить контроллер при загрузке: "Включен" и "Запущен".
- Имя таблиц для хранения конфигурации параметров контроллера стандартного и логического типа.

- Политика планирования и приоритет задачи сбора данных.
- ModBus протокол, используемый для опроса физического устройства (TCP/IP, RTU или ASCII).
- Адрес исходящего транспорта из списка сконфигурированных исходящих транспортов в подсистеме "Транспорты" OpenSCADA.
- ModBus узел назначения. В случае с протоколами RTU и ASCII — это индивидуальный адрес физического устройства, а при TCP/IP — идентификатор единицы.
- Объединение фрагментов регистров. Стандартные функции 01-04 позволяют запросить сразу несколько смежных регистров или битов. Такая стратегия часто позволяет оптимизировать трафик и время. Однако нужные регистры не всегда расположены смежно друг к другу, данная опция позволяет собирать их в блоки до 100 регистров или 1600 битов. К установке данного параметра нужно подходить с осторожностью, поскольку не все устройства поддерживают доступ к регистрам между фрагментами!
- Использовать функции записи нескольких элементов (0x0F,0x10). Вместо функций одно-элементной записи будут использованы много-элементные.
- Время ожидания соединения в миллисекундах. Указывает промежуток времени, в течение которого ожидать ответа. В случае указания нулевого значения используется время ожидания по умолчанию в транспорте. Позволяет учитывать индивидуальные свойства контроллера в общей сети.
- Время восстановления соединения в секундах. Указывает промежуток времени по истечению которого осуществлять повторную попытку запроса к ранее недоступному устройству.
- Попыток запроса для протоколов RTU и ASCII. Указывает на количество попыток повторения запроса в случае получения неполного или повреждённого ответа.

3.2. Параметры

Модуль сбора данных предоставляет два типа параметра: "Стандартный"(std) и "Логический"(logic). Дополнительными конфигурационными полями параметров данного модуля являются:

- "Стандартный"(std):
 - **Перечень атрибутов** — содержит структурированный список конфигурации атрибутов ModBus.
- "Логический"(logic):
 - **Шаблон параметра** — адрес шаблона параметра DAQ.

Стандартный тип параметра(std)

Главная страница конфигурации параметра стандартного типа представлена на рисунке 9.

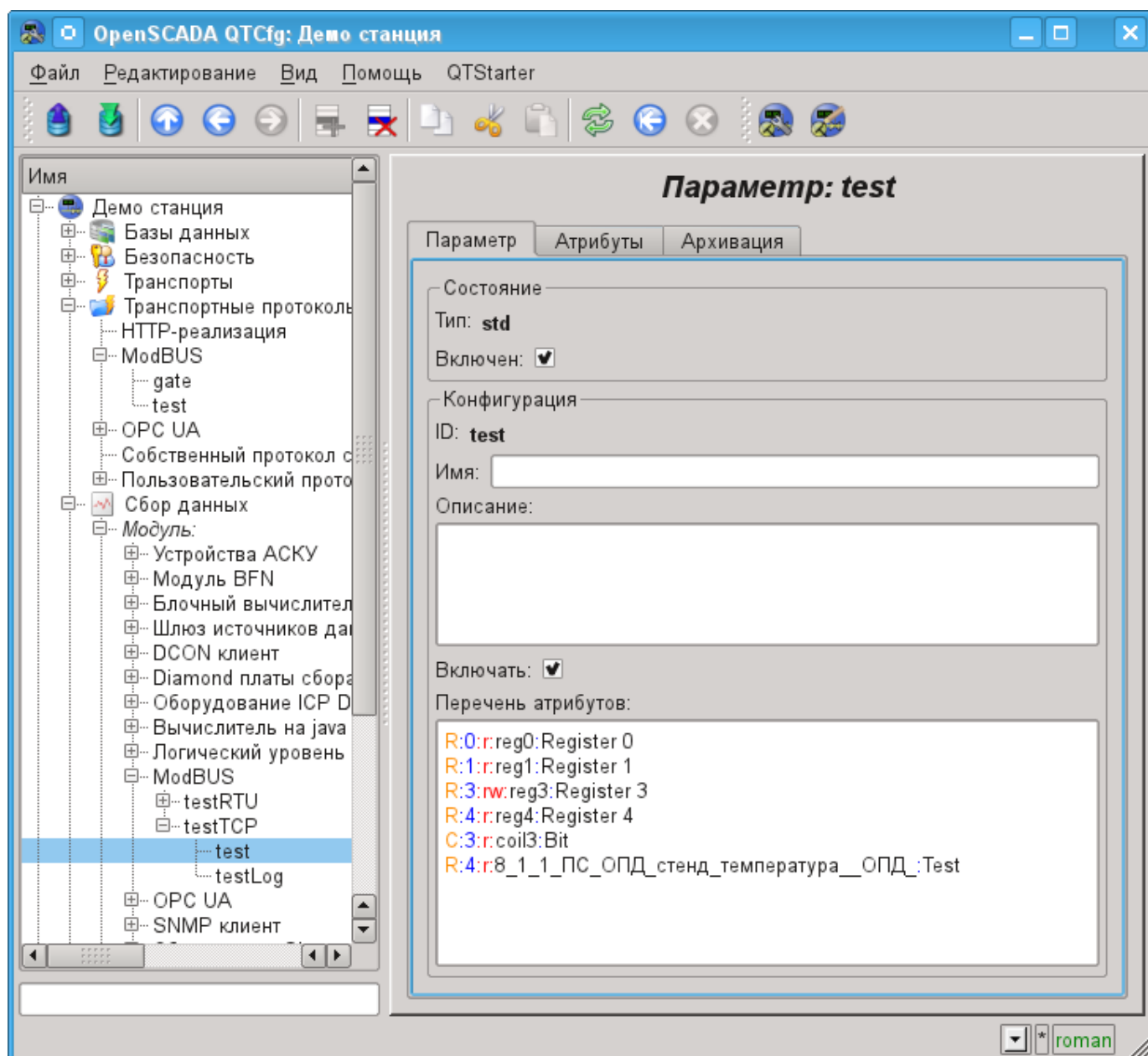


Рис.9. Вкладка конфигурации параметра стандартного типа.

Структура атрибута, в параметре списка атрибутов, записывается следующим образом:
`<dt>:<numb>:<wr>:<id>:<name>`.

Где:

dt — ModBus тип данных (R-регистр, C-бит, RI-регистр входа, CI-бит входа). R и RI могут быть расширены суффиксами: i2-Int16, i4-Int32, f-Float, b5-Bit5.

numb — номер регистра или бита ModBus устройства (десятичный, восьмеричный или шестнадцатеричный);

wr — режим чтения-записи (r-чтение, w-запись, rw-чтение и запись);

id — идентификатор атрибута OpenSCADA;

name — имя атрибута OpenSCADA.

В соответствии с указанным списком атрибутов выполняется опрос и создание атрибутов параметра (рис.10).

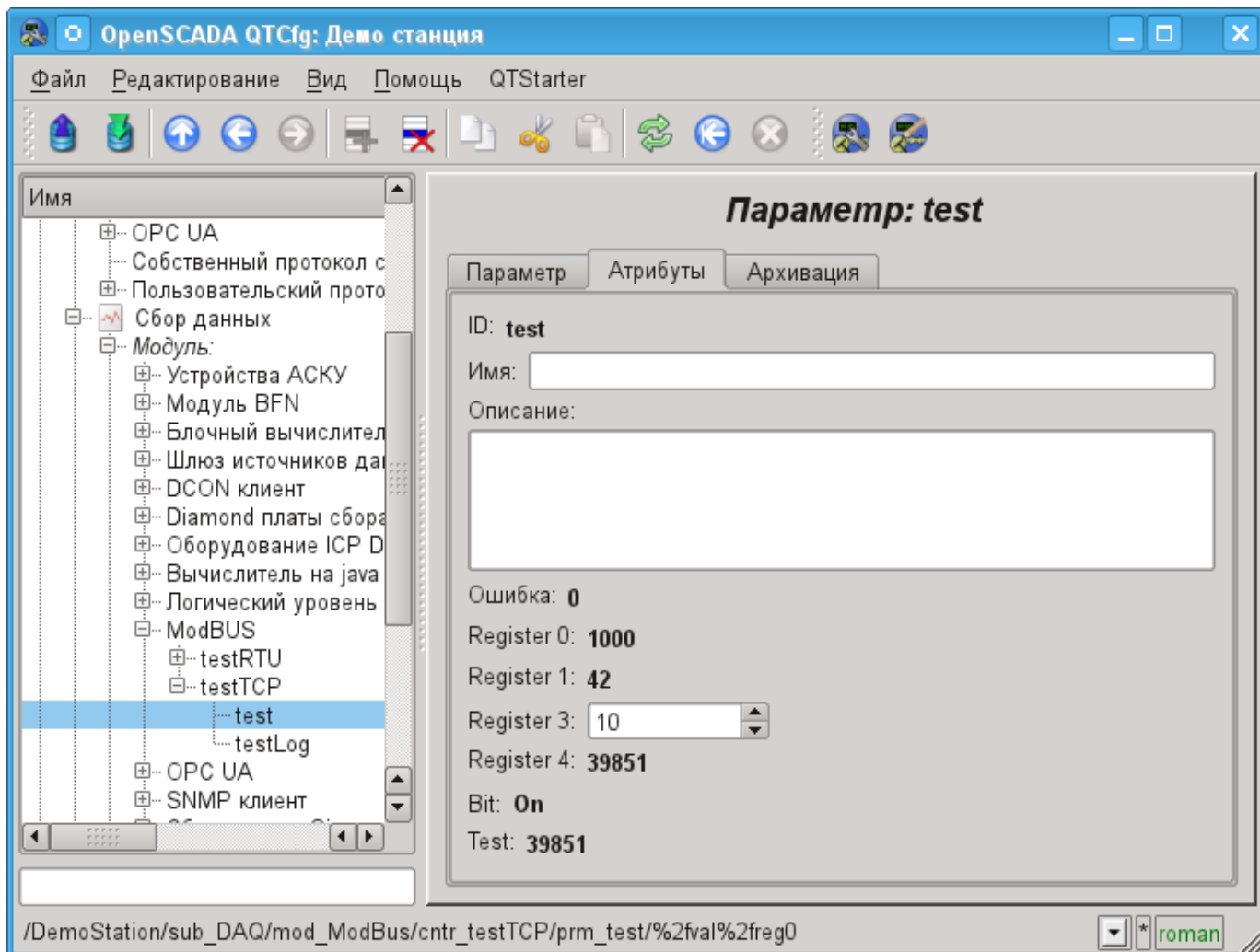


Рис.10. Вкладка атрибутов параметра стандартного типа.

Логический тип параметра(logic)

Главная страница конфигурации параметра логического типа представлена на рисунке 11.

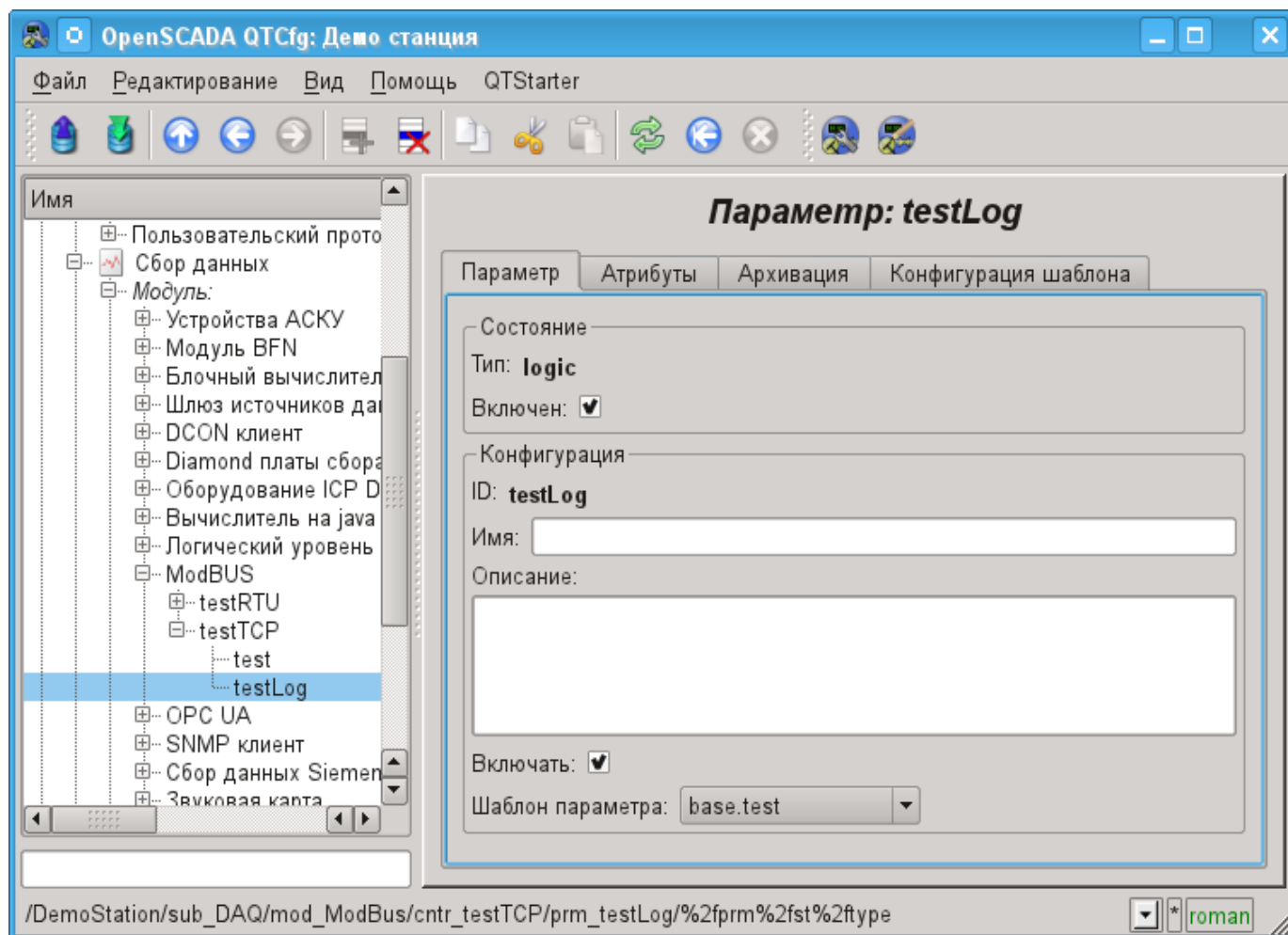


Рис.11. Вкладка конфигурации параметра логического типа.

При формировании шаблона, для логического типа параметра данного контроллера, не нужно учитывать формат ссылки шаблона, поскольку он не используется и может быть опущен. Само же значение ссылки, при конфигурации шаблона (рис.12), записывается в формате: **<dt>:<numb>:<wr>**.

Где:

- dt* — ModBus тип данных (R-регистр, C-бит, RI-регистр входа, CI-бит входа). R и RI могут быть расширены суффиксами: i2-Int16, i4-Int32, f-Float, b5-Bit5.
- numb* — номер регистра или бита ModBus устройства (десятичный, восьмеричный или шестнадцатеричный);
- wr* — режим чтения-записи (r-чтение, w-запись, gw-чтение и запись), пустое значение интерпретируется как "rw".

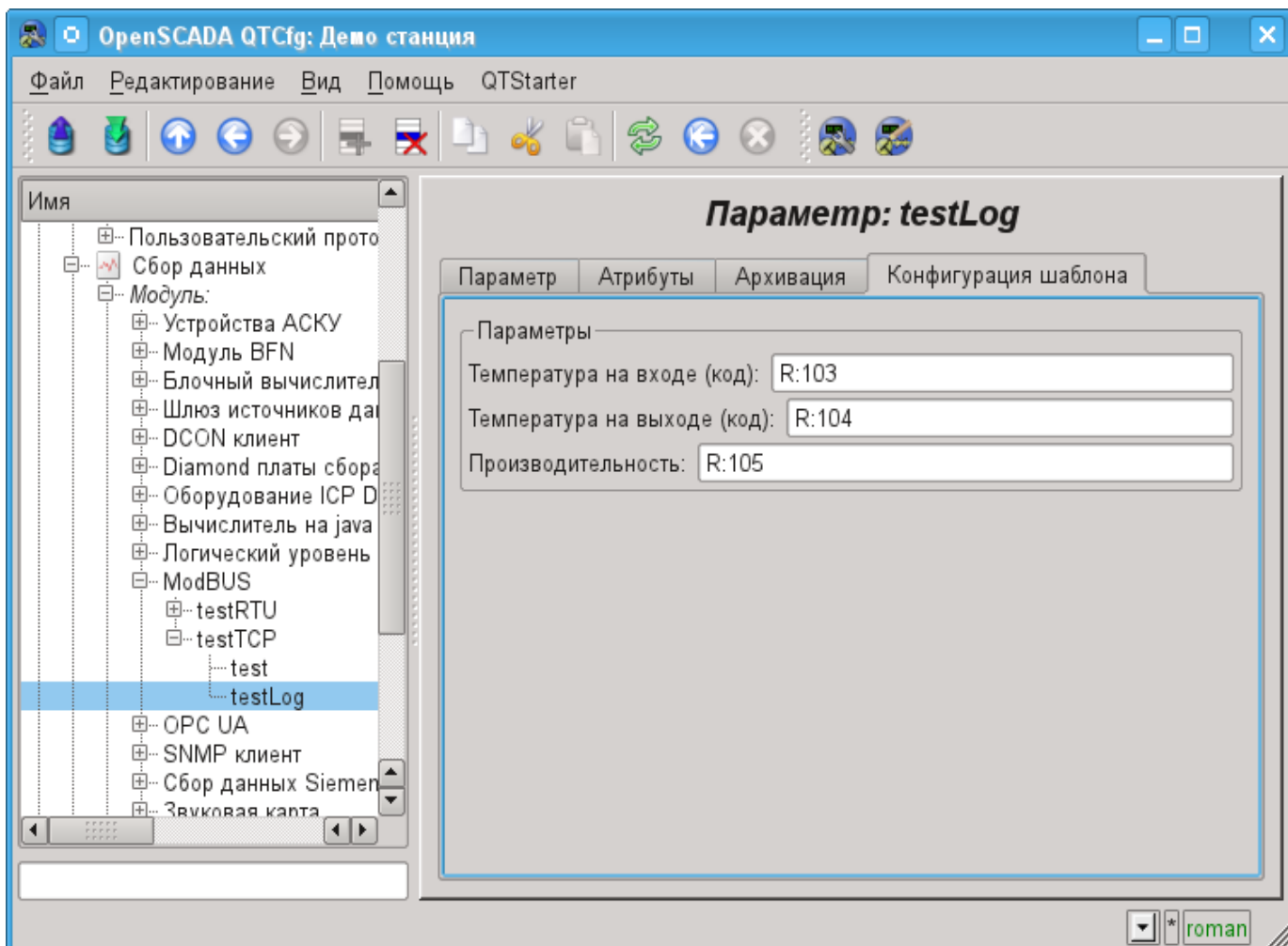


Рис.12. Вкладка "Конфигурация шаблона" параметра логического типа.

Модулем предусмотрена особая обработка ряда атрибутов шаблона:

- *f_freq* — Частота вычисления процедуры шаблона или время после последнего вычисления, отрицательное в секундах, для планирования по CRON, только чтение.
- *f_start* — Флаг первого выполнения процедуры шаблона, запуск, только чтение.
- *f_stop* — Флаг последнего выполнения процедуры шаблона, останов, только чтение.
- *f_err* — Ошибка параметра, полный доступ. Значение этого атрибута шаблона попадает в атрибут ошибки параметра "err".
- *SHIFR* — Значение шифра параметра, только чтение.
- *NAME* — Значение имени параметра, только чтение.
- *DESCR* — Значение описания параметра, только чтение.
- *this* — Объект данного параметра, позволяет получить доступ к атрибутам параметра, например, для доступа к архивам.

В соответствии с шаблоном, лежащим в основе параметра, мы получаем набор атрибутов параметра рис.13.

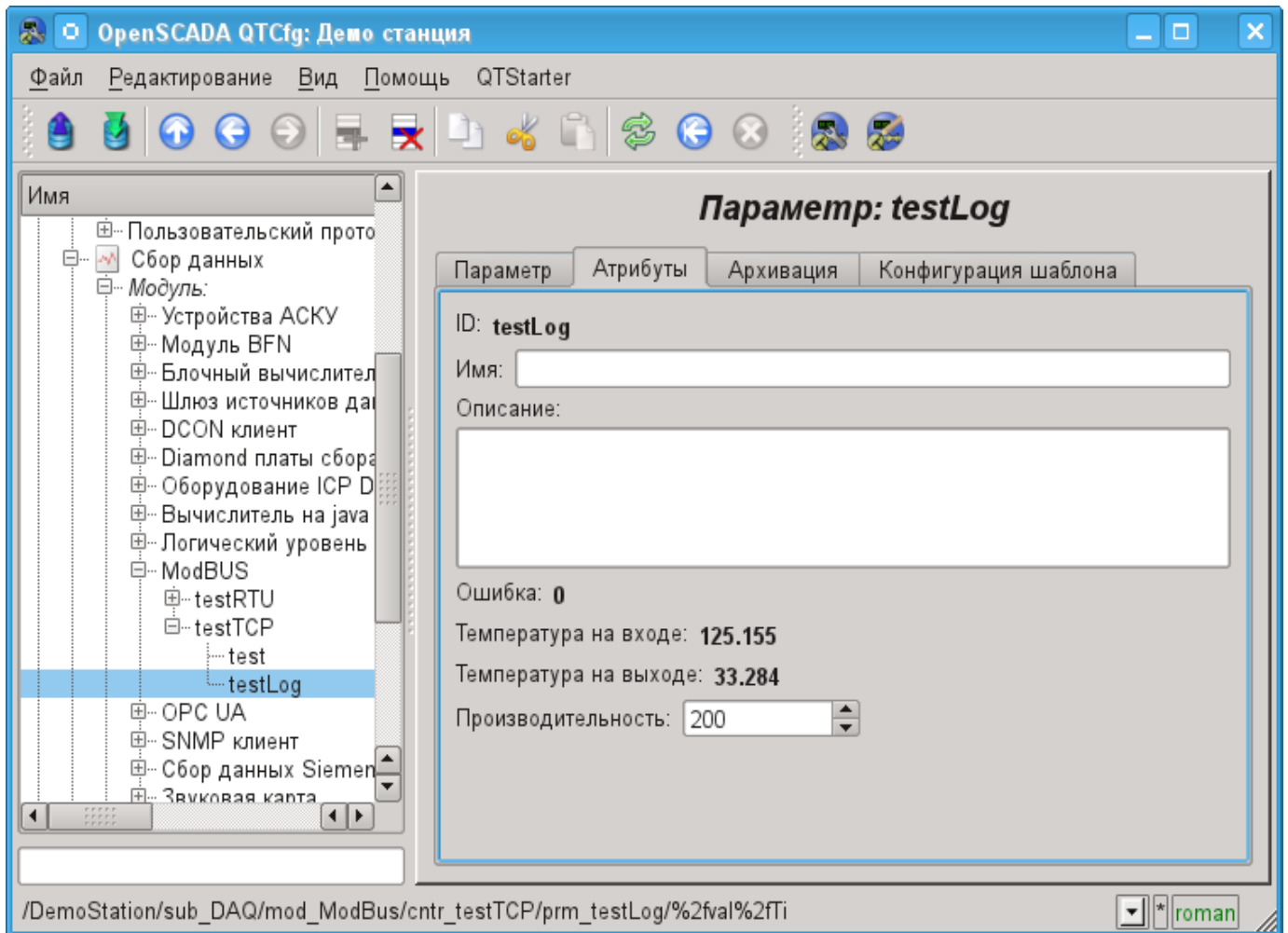


Рис.13. Вкладка атрибутов параметра логического типа.

3.3. API пользовательского программирования

В виду поддержки модулем параметров логического типа имеет смысл предоставление ряда функций пользовательского API для вызова из шаблона логического параметра.

Объект "Контроллер" (`this.nodePrev()`)

- *string* `messIO(string pdu)` — отправка PDU `<pdu>` через транспорт объекта контроллера посредством ModBus протокола. PDU результата запроса помещается вместо запроса в `<pdu>`, а ошибка возвращается в результате функции.