

# Модуль подсистемы “DAQ” <ModBus>

Модуль:	ModBus
Имя:	ModBus клиент
Тип:	DAQ
Источник:	daq_ModBus.so
Версия:	0.9.0
Автор:	Роман Савоченко
Описание:	Предоставляет реализацию клиента ModBus-протокола. Поддерживает протоколы ModBus/TCP, ModBus/RTU и ModBus/ASCII.
Лицензия:	GPL

## Оглавление

<a href="#">Модуль подсистемы “DAQ” &lt;ModBus&gt;</a> .....	1
<a href="#">Введение</a> .....	2
<a href="#">1 Общее описание протокола ModBus</a> .....	2
<a href="#">1.1 Адресация</a> .....	2
<a href="#">1.2 Стандартные коды функций</a> .....	2
<a href="#">2 Модуль</a> .....	3
<a href="#">2.1 Контроллер данных</a> .....	3
<a href="#">2.2 Параметры</a> .....	5
<a href="#">2.3 Последовательные порты</a> .....	7

## 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.

Данный модуль предоставляет возможность собирать информацию у различных устройств по протоколу ModBus во всех режимах.

## 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 мастер определяет, по временному интервалу между окончанием приёма предыдущего байта и началом приёма следующего. Если этот интервал превысил время, необходимое для приёма двух байт на заданной скорости передачи, приём кадра ответа считается завершённым. Кадры запроса и ответа по протоколу ModBus имеют фиксированный формат.

### 1.1 Адресация

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

### 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

Данным модулем используются команды 0x03 и 0x06 для чтения и записи регистров, 0x01 и 0x05 для чтения и записи битов, 0x02 и 0x04 для чтения регистра и бита входа соответственно.

## 2 Модуль

Данный модуль предоставляет возможность опроса и записи регистров и битов устройств посредством режима протоколов TCP, RTU, ASCII и команд запроса 0x01 — 0x06.

### 2.1 Контроллер данных

Для добавления источника данных ModBus создаётся и конфигурируется контроллер в системе OpenSCADA. Пример вкладки конфигурации контроллера данного типа изображен на рис.1.

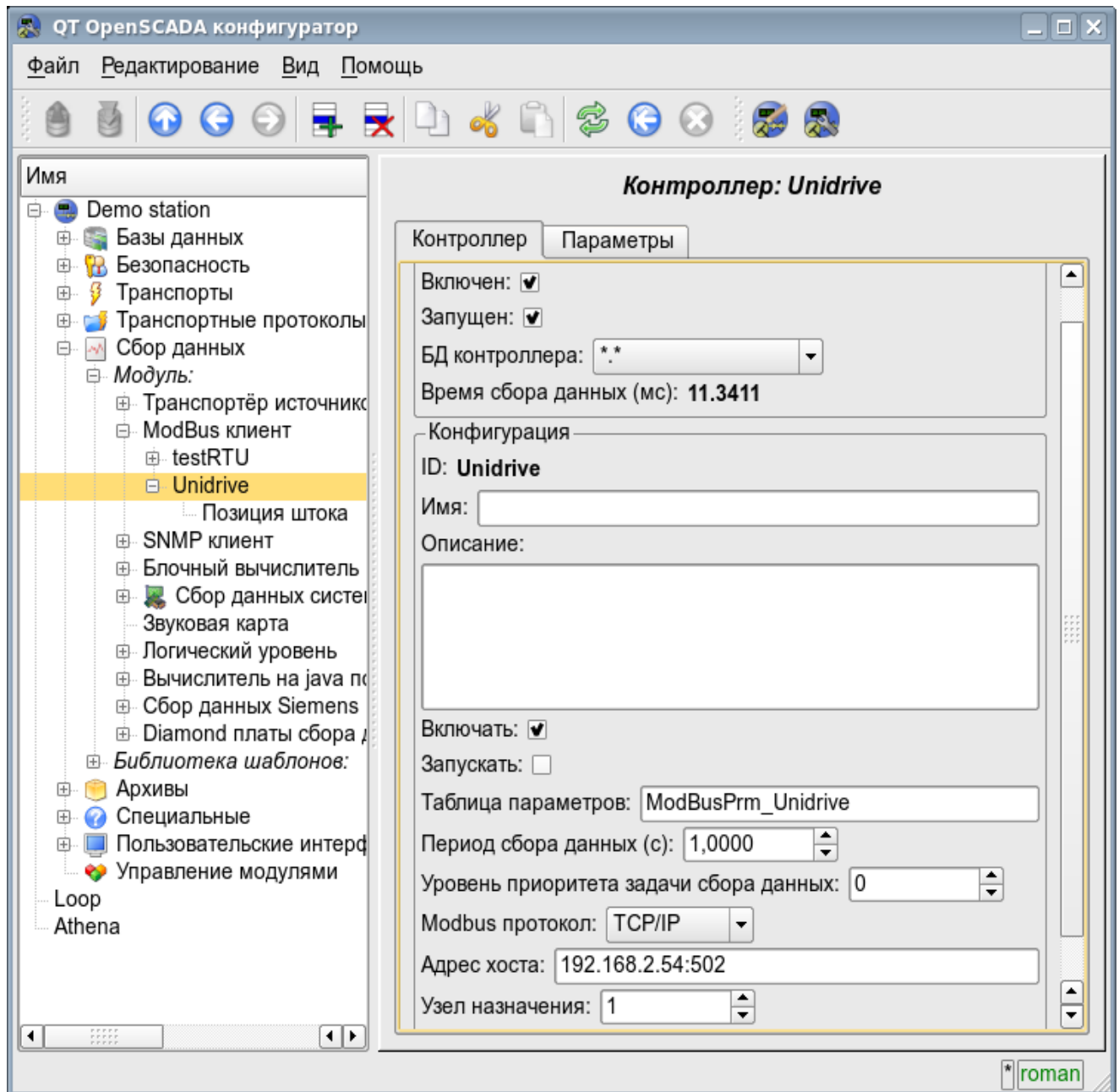


Рис.1. Вкладка конфигурации контроллера.

С помощью этой вкладки можно установить:

- Состояние контроллера, а именно: «Включен», «Запущен», имя БД содержащей конфигурацию и время сбора данных.
- Идентификатор, имя и описание контроллера.
- Состояние в которое переводить контроллер при загрузке: «Включен» и «Запущен».
- Имя таблицы для хранения конфигурации параметров контроллера.

- Период и приоритет задачи сбора данных.
- ModBus протокол, используемый для опроса физического устройства.
- Адрес устройства. В случае с протоколами RTU и ASCII адрес записывается в виде пути к устройству последовательно порта (/dev/ttyS0). В случае с протоколом TCP в адресе записывается хост и порт удалённого устройства (rem.host.org:502).
- Узел назначения. В случае с протоколами RTU и ASCII это индивидуальный адрес физического устройства.
- Объединение фрагментов регистров. Стандартные функции 01–04 позволяет запросить сразу несколько смежных регистров или битов. Такая стратегия часто позволяет оптимизировать трафик. 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.
- Времена интервалов ожидания последовательного интерфейса для протоколов RTU и ASCII. А именно: «Время фрейма», «Время символа» и «Время запроса».

## 2.2 Параметры

Модуль *ModBus* предоставляет только один тип параметров – “Стандарт”. Дополнительным конфигурационным полем параметра данного модуля (рис.2) является перечень обрабатываемых атрибутов(регистров ModBus). Атрибут в этом перечне записывается следующим образом: **<dt>:<numb>:<wr>:<id>:<name>**.

Где:

- dt* — [Mod Bus](#) тип данных (R-регистр, C-бит, RI-регистр входа, CI-бит входа);
- 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.

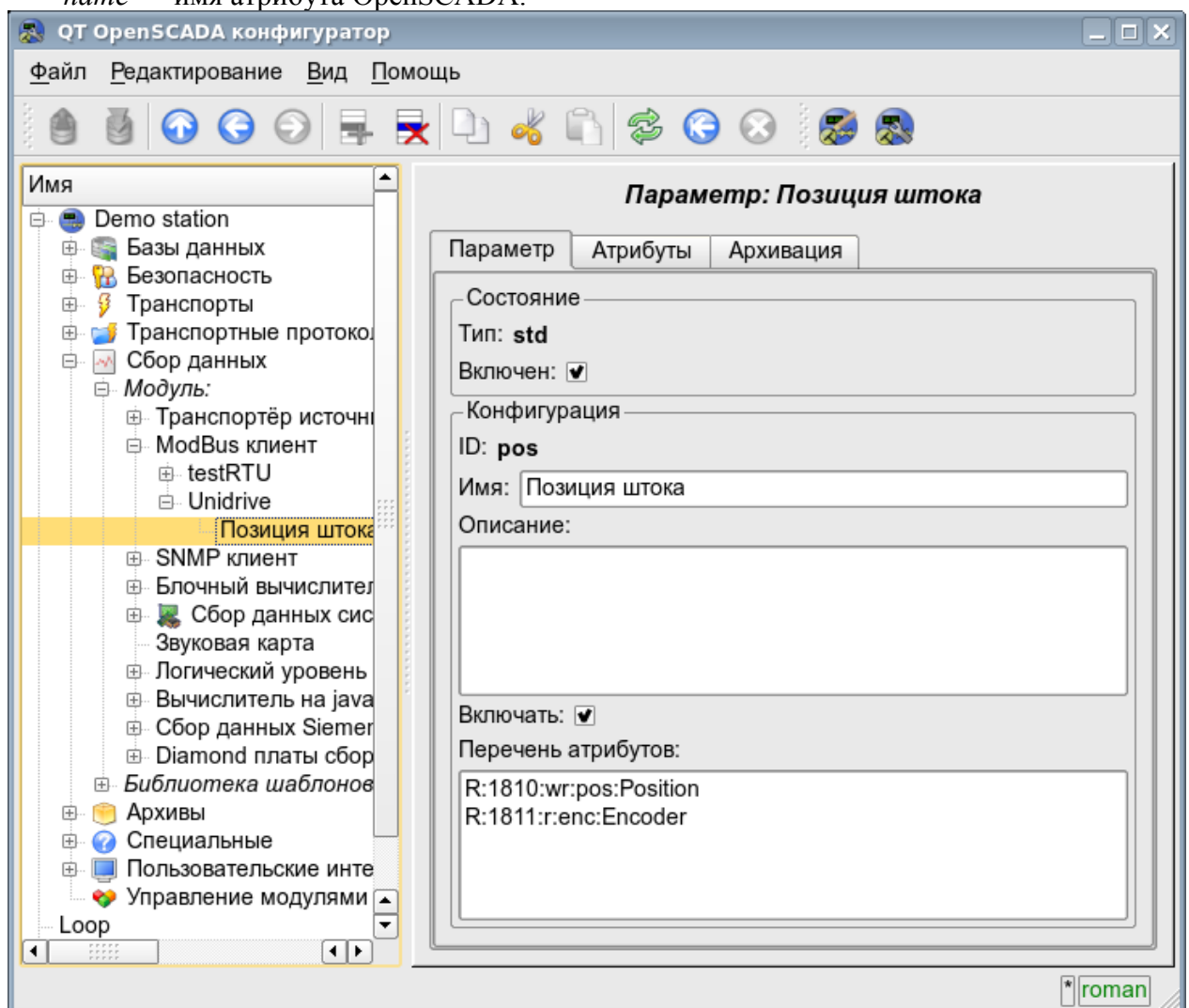


Рис.2. Вкладка конфигурации параметра.

В соответствии с указанным списком атрибутов выполняется опрос и создание атрибутов параметра (рис.3).

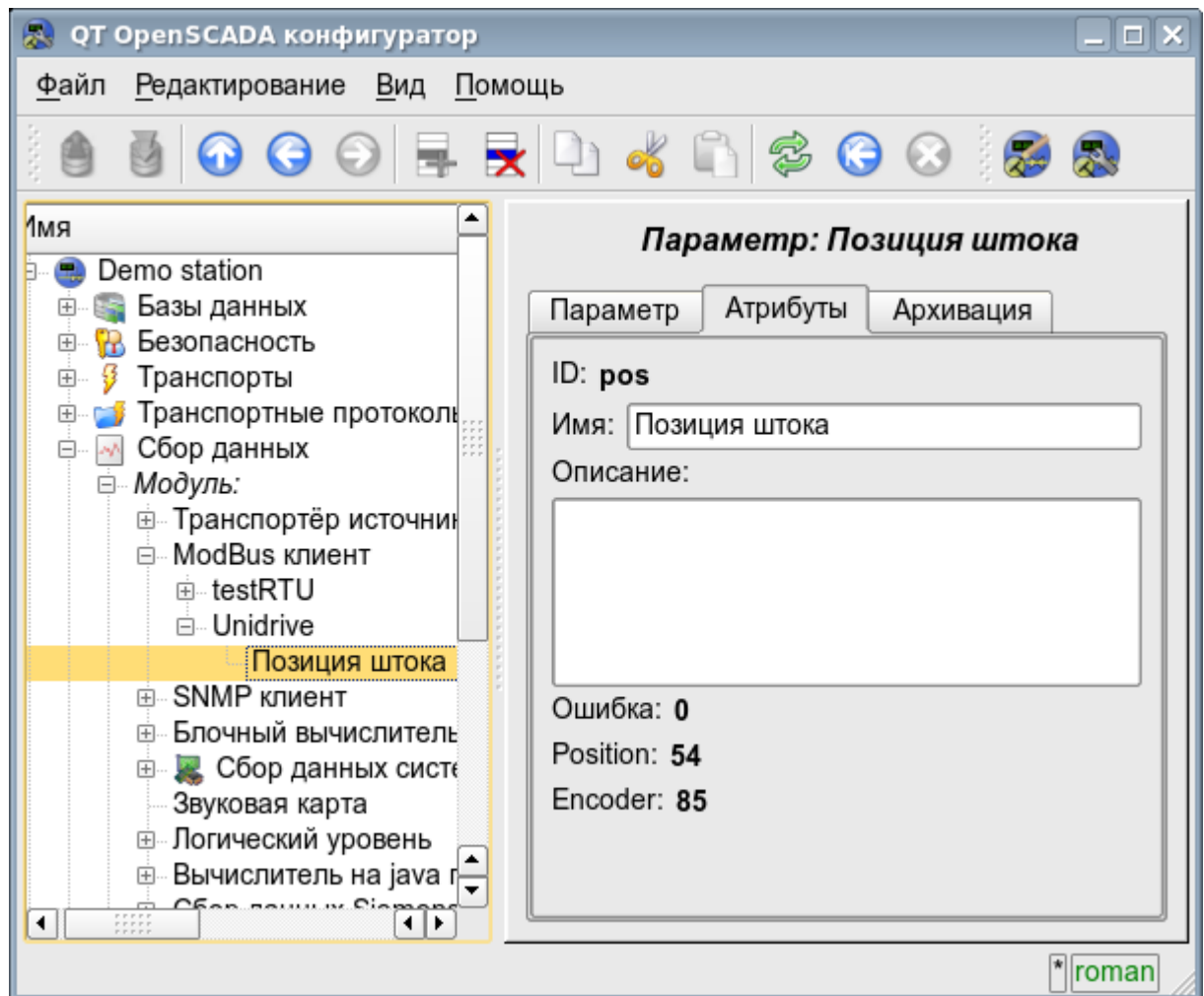


Рис.3. Вкладка атрибутов параметра.

## 2.3 Последовательные порты

В случае использования протоколов RTU и ASCII каждый контроллер данного модуля может работать с собственным реальным контроллером, через общий последовательный интерфейс. А учитывая тот факт, что по последовательному интерфейсу одновременно может работать только один клиент необходимо предусмотреть монопольный ресурс доступа к последовательному интерфейсу. Для обеспечения этого требования, а также для предоставления возможности конфигурации параметров порта, каждый последовательный интерфейс реализован в собственном объекте. Конфигурация последовательных интерфейсов производится в главной вкладке модуля (рис.4). Где в строке таблицы размещается конфигурация последовательного интерфейса. На этой же вкладке можно включить протокол коммуникации ModBus указав размерность протокола более нуля.

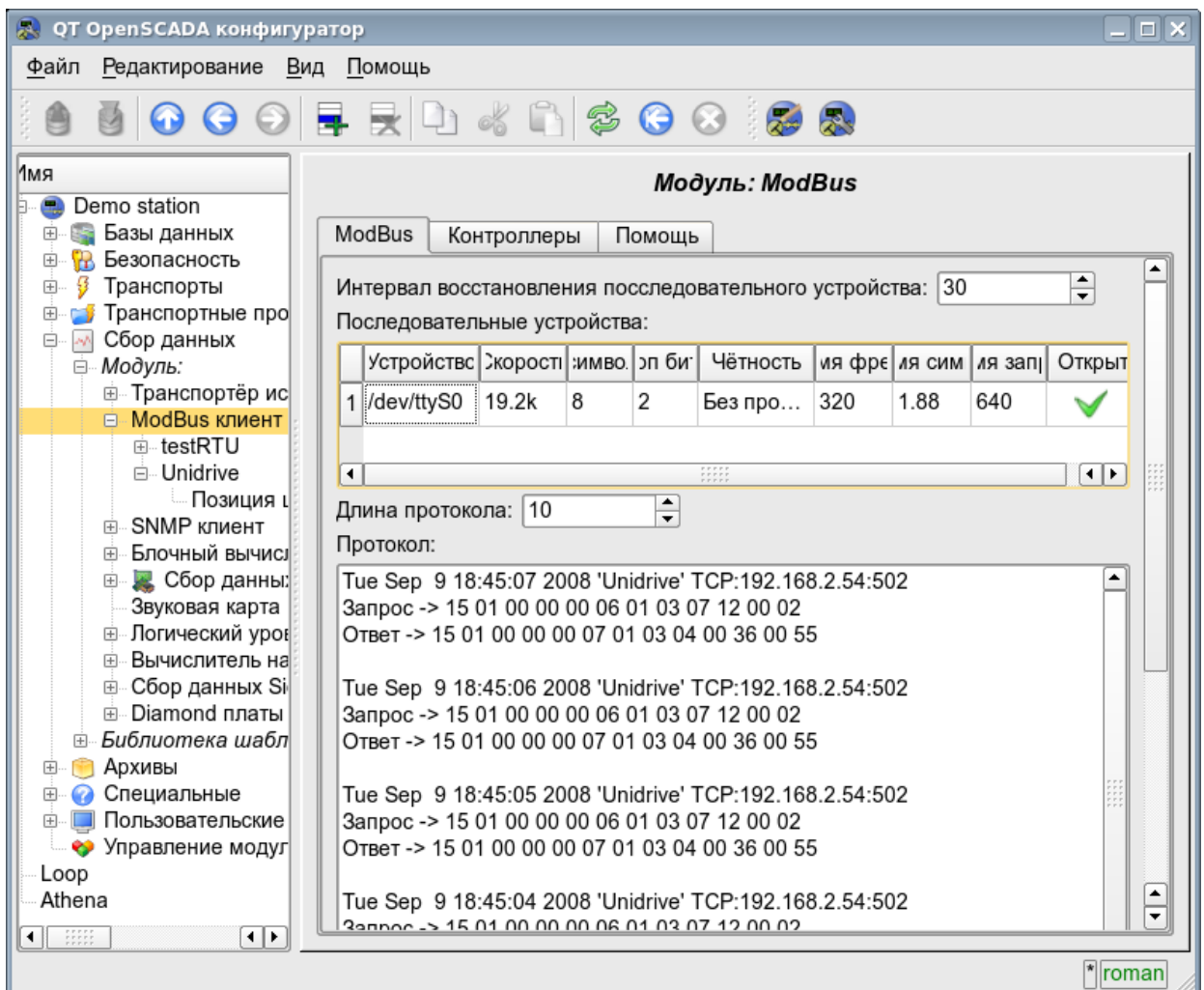


Рис.4. Главная вкладка модуля.

С помощью этой вкладки можно установить такие параметры последовательного интерфейса:

- Идентификатор последовательного интерфейса, совпадает с путём к устройству интерфейса.
- Скорость передачи данных.
- Количество битов в одном символе передачи.
- Количество стоп битов символа.
- Наличие проверки на чётность.
- Таймаут фрейма ModBus запроса (мс). Для ограничения размера ModBus блока.
- Таймаут ожидания символа блока запроса (мс).
- Таймаут запроса. Время ожидания ответа от устройства (мс).
- Состояние интерфейса «Открыт».