

Модуль подсистемы “Транспорты” <Serial>

Модуль:	Serial
Имя:	Последовательный интерфейс
Тип:	Транспорт
Источник:	tr_Serial.so
Версия:	0.5.0
Автор:	Роман Савоченко
Описание:	Предоставляет последовательный интерфейс. Используется для обмена данными через последовательные интерфейсы типа RS232, RS485, GSM и другое.
Лицензия:	GPL

Оглавление

Модуль подсистемы “Транспорты” <Serial>	1
Введение	1
1. Входящие транспорты	1
2. Исходящие транспорты	3

Введение

Модуль транспорта Serial предоставляет в систему поддержку транспортов основанных на последовательных интерфейсах типа RS232, RS485, GSM и другие. Поддерживаются входящие и исходящие транспорты. Добавить новые входящие и исходящие интерфейсы можно посредством конфигурации транспортной подсистемы в любом конфигураторе системы OpenSCADA.

1. Входящие транспорты

Сконфигурированный и запущенный входящий транспорт открывает порт последовательного интерфейса для ожидания запросов клиентов. Каждый входящий интерфейс обязательно связывается с одним из доступных транспортных протоколов и которому передаются входящие сообщения.

Диалог конфигурации входящего последовательного интерфейса изображён на рис.1.

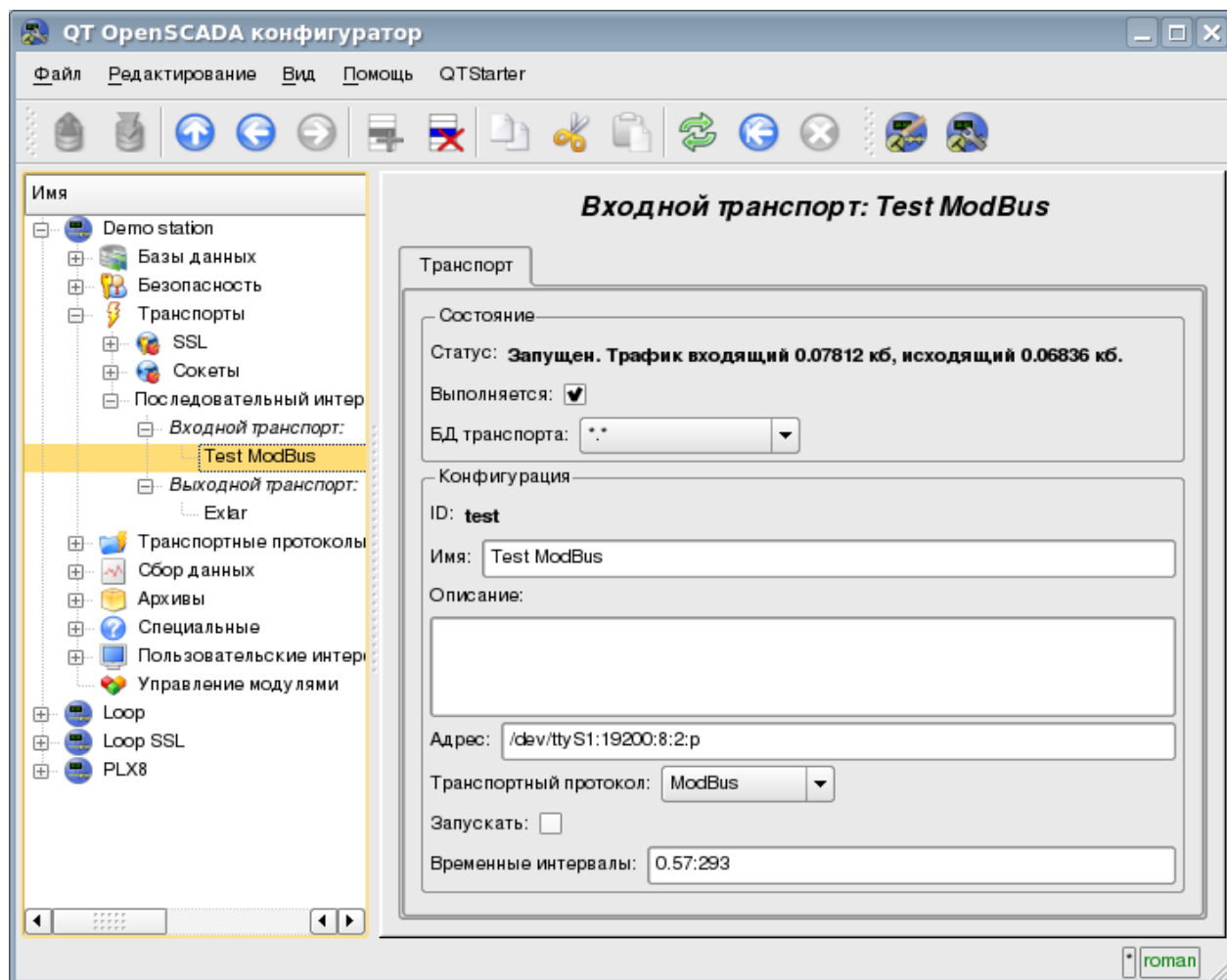


Рис.1. Диалог конфигурации входящего последовательного интерфейса.

С помощью этого диалога можно установить:

- Состояние транспорта, а именно: «Статус», «Выполняется» и имя БД содержащей конфигурацию.
- Идентификатор, имя и описание транспорта.
- Адрес интерфейса в формате строки: `'[dev]:[spd]:[len]:[stop]:[parity]'`. Где:
 - *dev* — адрес последовательного устройства (`/dev/ttyS0`);
 - *spd* — скорость последовательного устройства из ряда: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 500000, 576000 или 921600;
 - *len* — длина символа (7 или 8);
 - *stop* — число стоп битов (1 или 2);
 - *parity* — проверка целостности символа (р-чётность, n-нечётность, 0-отключено).
- Выбор транспортного протокола.
- Состояние, в которое переводить контроллер при загрузке: «Запущен».
- Временные интервалы интерфейса в формате строки: `'[symbol]:[frm]'`. Где:
 - *symbol* — время символа, в миллисекундах. Используется для контроля факта окончания фрейма;
 - *frm* — максимальное время фрейма, в миллисекундах. Используется для ограничение максимального размера пакета запроса (фрейма).

2. Исходящие транспорты

Сконфигурированный и запущенный исходящий транспорт открывает порт последовательного интерфейса для отправки запросов через него.

Главная вкладка страницы конфигурации исходящего последовательного интерфейса изображена на рис.2.

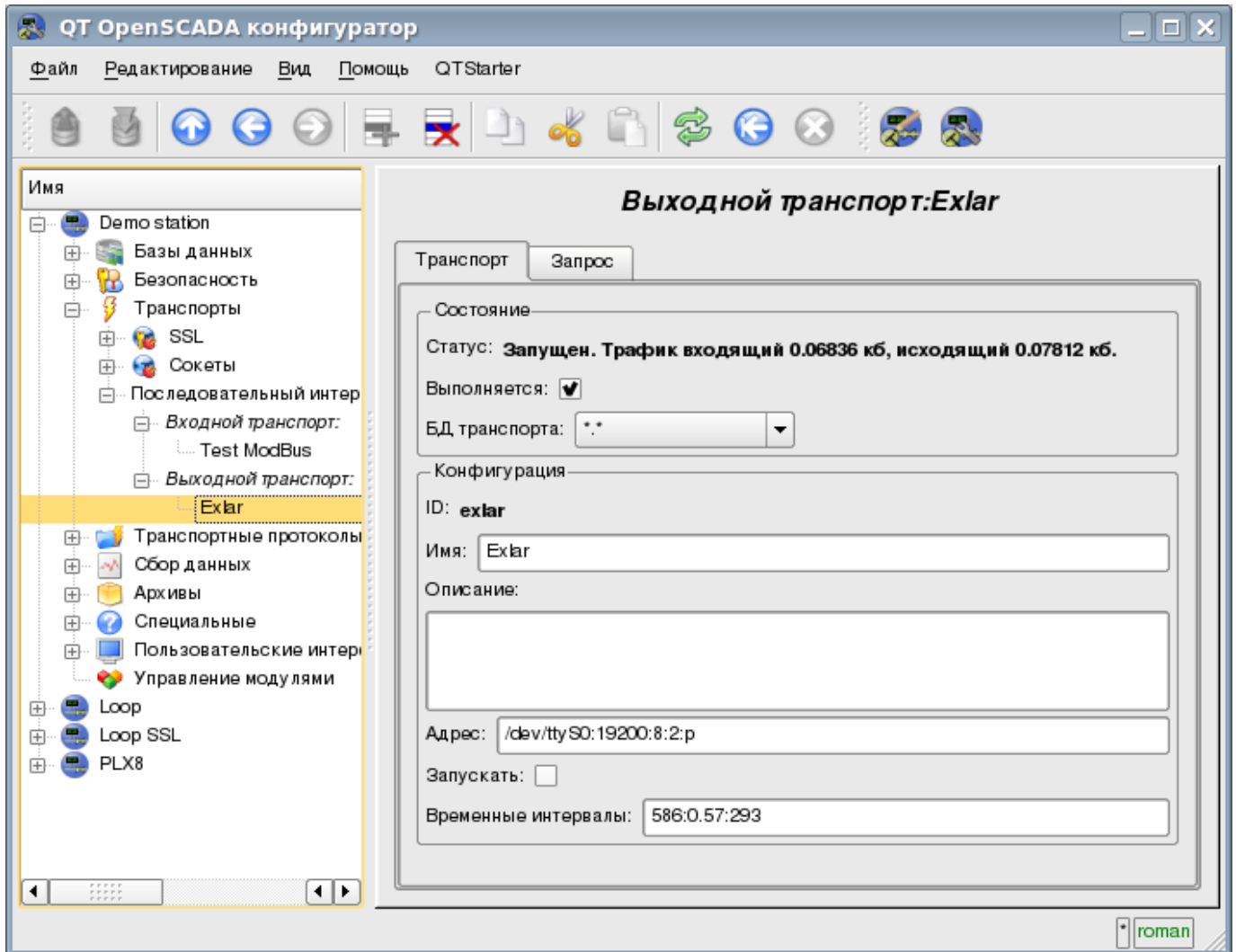


Рис.2. Главная вкладка страницы конфигурации исходящего последовательного интерфейса.

С помощью этого диалога можно установить:

- Состояние транспорта, а именно: «Статус», «Запущен» и имя БД содержащей конфигурацию.
- Идентификатор, имя и описание транспорта.
- Адрес интерфейса в формате строки: '*[dev]:[spd]:[len]:[stop]:[parity]*'. Где:
 - *dev* — адрес последовательного устройства (/dev/ttyS0);
 - *spd* — скорость последовательного устройства из ряда: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 500000, 576000 или 921600;
 - *len* — длина символа (7 или 8);
 - *stop* — число стоп битов (1 или 2);
 - *parity* — проверка целостности символа (p-чётность, n-нечётность, 0-отключено).
- Состояние, в которое переводить контроллер при загрузке: «Запущен».
- Временные интервалы интерфейса в формате строки: '*{conn}:[symbol]:[frm]*'. Где:
 - *conn* — время ожидания соединения т.е. ответа от удалённого устройства.
 - *symbol* — время символа, в миллисекундах. Используется для контроля факта окончания фрейма;
 - *frm* — максимальное время фрейма, в миллисекундах. Используется для ограничение максимального размера пакета ответа (фрейма).