

# Модуль подсистемы “Протоколы” <SelfSystem>

Модуль:	SelfSystem
Имя:	Собственный протокол системы OpenSCADA
Тип:	Протокол
Источник:	prot_SelfSystem.so
Версия:	0.8.0
Автор:	Роман Савоченко
Описание:	Собственный протокол системы OpenSCADA, поддерживает основные функции.
Лицензия:	GPL

## Оглавление

Модуль подсистемы “Протоколы” <SelfSystem>.....	1
Введение.....	1
1 Синтаксис протокола.....	1
2 Внутренняя структура исходящего протокола.....	2

## Введение

Модуль транспортного протокола SelfSystem предназначен для отражения интерфейса управления системы OpenSCADA в сеть, с целью предоставления возможности, внешним системам, взаимодействовать с системой OpenSCADA, а также для взаимодействия станций построенных на основе OpenSCADA между собой.

Первым опытом использования функций данного модуля стала поддержка возможности удалённого конфигурирования одной OpenSCADA станции из другой, через сеть, посредством модуля конфигурирования [QTCfg](#).

## 1 Синтаксис протокола

Протокол построен по механизму запрос-ответ. Типы запросов и их структура сведены в таблице 1.

Таблица 1 Структура запроса.

Запросы
<b>REQ:</b> «SES_OPEN <user> <password>\n» <b>REZ OK:</b> «REZ 0 <ses_id>\n» <b>REZ ERR:</b> «REZ 1 Auth error. User or password error.\n» Запрос на открытие сессии от имени пользователя <user> с паролем <password>. В случае удачи будет получен идентификатор сессии, иначе код и сообщение об ошибке.
<b>REQ:</b> «SES_CLOSE <ses_id>\n» <b>REZ:</b> «REZ 0\n» Закрытие сессии. Результат всегда удачен.

### Запросы

**REQ 1:** «REQ <ses\_id> <req\_size> \n <control interface command>»

**REQ 2:** «REQDIR <user> <password> <req\_size> \n <control interface command>»

**REZ OK:** «REZ 0 <rez\_size> \n <control interface command result>»

**REZ ERR:** «REZ 1 Auth error. Session no valid.\n»

**REZ ERR:** «REZ 2 <control interface err>»

Основные запросы, сеансовый и прямой. Реализуется путём отправки стандартной команды [интерфейса управления OpenSCADA](#) в поле <control interface command>. В результате будет получен ответ интерфейса управления <control interface command result> или одна из ошибок.

**REQ:** «ERR REQUEST»

**REZ ERR:** «REZ 3 Command format error.\n»

Любой некорректный запрос.

Протоколом поддерживается возможность упаковки трафика. Пакуются только данные интерфейса управления <control interface command> и <control interface command result>. Факт прихода упакованного запроса или ответа определяется отрицательным значением размера запроса <req\_size> или ответа <rez\_size>.

Для управления параметрами упаковки модулем предоставляется форма конфигурации (рис.1).

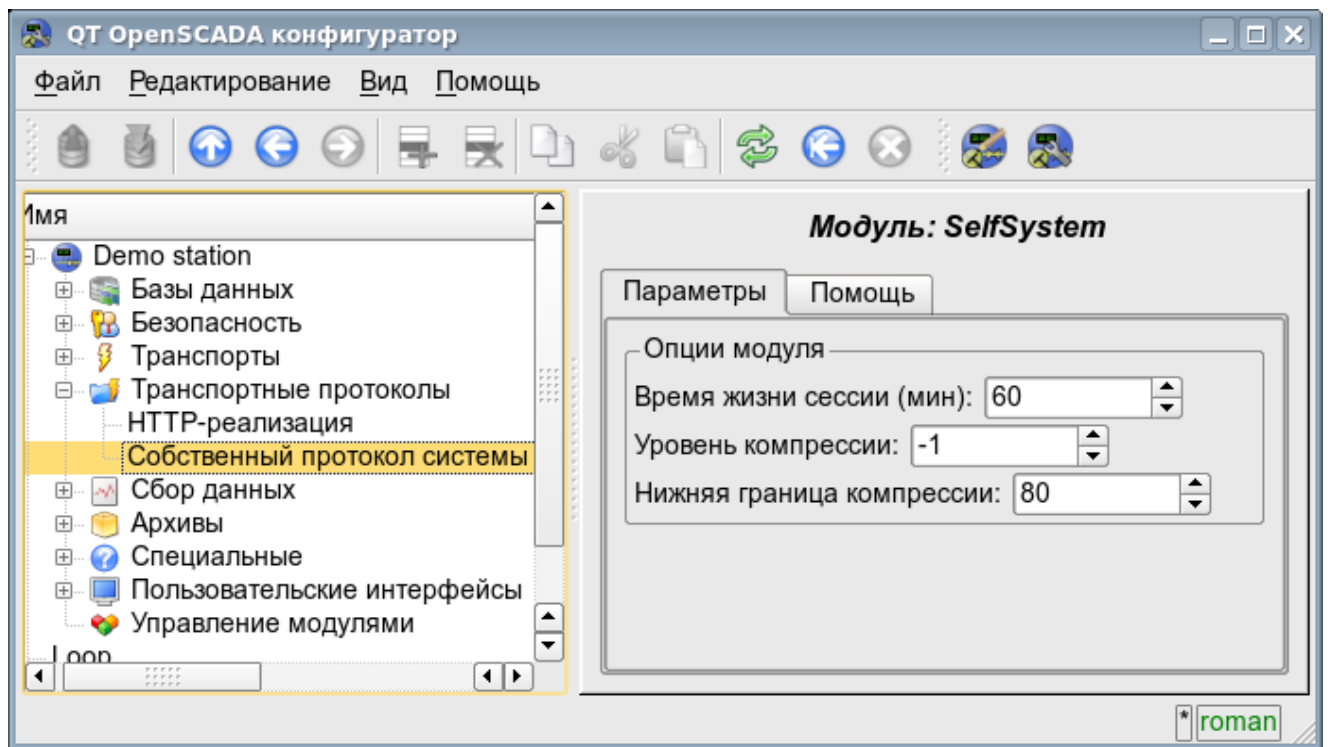


Рис.1. Форма конфигурации параметров упаковки.

На данной форме можно указать:

- время жизни сессии аутентификации;
- уровень компрессии протокола, в диапазоне от 0 до 9 (0-отключение компрессии, -1-оптимальный по производительности и качеству уровень компрессии);
- нижний порог использования компрессии; выключает компрессию мелких запросов.

## 2 Внутренняя структура исходящего протокола

Внутренняя структура описывается массивом данных в строке predetermined structure and is parsed by the outgoing protocol before exchange. The structure of the line of this protocol has the following form: [*direct*>\n<user>\n<pass>\n<data>] Where:

*direct* — признак отправки сообщения минуя процедуру открытия сеанса (0-открывать сеанс, 1-слать сразу); *user* — пользователь; *pass* — пароль; *data* — данные запроса.