

ПРОГРАММНЫЙ КОМПЛЕКС ASTRAREGUL



РГДП.58.29.14.000-001-02 РП

# Инфраструктура

---

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

# СПИСОК ИЗМЕНЕНИЙ

Редакция	Список изменений
Редакция 2	<ul style="list-style-type: none"><li>- Обновлен раздел <a href="#">Astra.Imitator</a></li><li>- Обновлено описание модулей компонента Astra.Server: "<a href="#">Клиент BACnet</a>", "<a href="#">Опросчик Modbus TCP</a>", "<a href="#">Опросчик Modbus RTU</a>", "<a href="#">ОПС HDA Сервер</a>", "<a href="#">ОПС HDA Клиент</a>", "<a href="#">ОПС АЕ Сервер</a>", "<a href="#">Модуль NetDiag</a>", "<a href="#">Модуль NetDiag2</a>", "<a href="#">Менеджер SNMP</a>", "<a href="#">SQL Connector</a>", "<a href="#">Siemens S7 Client</a>", "<a href="#">Модуль SMTP</a>", "<a href="#">Syslog Клиент</a>", "<a href="#">Опросчик МЭК 60870-5-104</a>", "<a href="#">Станция МЭК 60870-5-104</a>", "<a href="#">Модуль истории</a>".</li></ul>

# ОГЛАВЛЕНИЕ

СПИСОК ИЗМЕНЕНИЙ .....	2
1. Инфраструктура .....	10
1.1. Astra.Server .....	11
1.1.1. Конфигурирование .....	18
1.1.1.1. Ручная настройка .....	21
1.1.1.2. Защита от несанкционированного доступа .....	23
1.1.2. Модули .....	26
1.1.2.1. ВАСnet .....	30
1.1.2.1.1. Клиент ВАСnet .....	31
1.1.2.1.1.1. Устройство ВАСnet .....	32
1.1.2.1.1.2. Настройка .....	35
1.1.2.1.1.3. Протокольные типы .....	39
1.1.2.1.1.4. Строковые идентификаторы объектов и свойств ...	42
1.1.2.1.1.5. Карта адресов .....	44
1.1.2.1.1.6. Значения сигнала доставки .....	46
1.1.2.1.1.7. Диагностика работы модуля .....	47
1.1.2.2. EtherNet/IP .....	49
1.1.2.2.1. Сканер EtherNet/IP .....	50
1.1.2.2.1.1. Настройка .....	51
1.1.2.2.1.2. Функции модуля .....	53
1.1.2.2.1.3. Карта адресов .....	57
1.1.2.2.1.4. Сигналы доставки .....	59
1.1.2.2.1.5. Диагностика работы модуля .....	60
1.1.2.3. Modbus .....	62
1.1.2.3.1. Опросчик Modbus TCP .....	63
1.1.2.3.1.1. Настройка .....	66
1.1.2.3.1.2. Функциональные возможности .....	69
1.1.2.3.1.3. Типы данных .....	82
1.1.2.3.1.4. Карта адресов .....	85
1.1.2.3.1.5. Использование сигналов .....	87
1.1.2.3.1.6. Диагностика работы модуля .....	92
1.1.2.3.2. Опросчик Modbus RTU .....	95

1.1.2.3.2.1. Настройка .....	98
1.1.2.3.2.2. Функциональные возможности .....	101
1.1.2.3.2.3. Типы данных .....	116
1.1.2.3.2.4. Карта адресов .....	119
1.1.2.3.2.5. Конфигурирование сигналов .....	121
1.1.2.3.2.6. Диагностика работы модуля .....	123
1.1.2.3.3. Станция Modbus TCP .....	129
1.1.2.3.3.1. Настройка .....	131
1.1.2.3.3.2. Типы данных .....	136
1.1.2.3.3.3. Карта адресов .....	139
1.1.2.3.3.4. Настройка сигналов .....	141
1.1.2.3.3.5. Диагностика работы модуля .....	142
1.1.2.3.4. Станция Modbus RTU .....	143
1.1.2.3.4.1. Настройка .....	144
1.1.2.3.4.2. Карта адресов .....	149
1.1.2.3.4.3. Настройка сигналов .....	151
1.1.2.3.4.4. Типы данных .....	153
1.1.2.3.4.5. Возвращаемые ошибки .....	156
1.1.2.4. OPC .....	157
1.1.2.4.1. OPC DA Сервер .....	158
1.1.2.4.1.1. Настройка .....	160
1.1.2.4.1.2. Карта адресов .....	163
1.1.2.4.1.3. Диагностика работы модуля .....	165
1.1.2.4.2. OPC DA Клиент .....	166
1.1.2.4.2.1. Настройка .....	168
1.1.2.4.2.2. Карта адресов .....	171
1.1.2.4.2.3. Качество сигналов .....	173
1.1.2.4.2.4. Диагностика работы модуля .....	175
1.1.2.4.3. OPC HDA Сервер .....	176
1.1.2.4.3.1. Настройка .....	178
1.1.2.4.3.2. Карта адресов .....	182
1.1.2.4.3.3. Диагностика работы модуля .....	183
1.1.2.4.4. OPC HDA Клиент .....	184
1.1.2.4.4.1. Настройка .....	186

1.1.2.4.4.2. Карта адресов .....	188
1.1.2.4.4.3. Диагностика работы модуля .....	189
1.1.2.4.5. OPC AE Сервер .....	190
1.1.2.4.5.1. Настройка .....	193
1.1.2.4.5.2. Функциональные возможности .....	198
1.1.2.4.5.3. Карта адресов .....	206
1.1.2.4.5.4. Диагностика работы модуля .....	207
1.1.2.4.6. OPC UA Сервер .....	208
1.1.2.4.6.1. Настройка .....	209
1.1.2.4.6.2. Способы аутентификации .....	214
1.1.2.4.6.2.1. Работа с клиентскими сертификатами .....	218
1.1.2.4.6.3. Карта адресов .....	222
1.1.2.4.6.4. Диагностика работы .....	224
1.1.2.4.7. OPC UA Клиент .....	225
1.1.2.4.7.1. Настройка .....	226
1.1.2.4.7.2. Настройка сервера .....	228
1.1.2.4.7.3. Карта адресов .....	229
1.1.2.4.7.4. Диагностика работы модуля .....	231
1.1.2.5. Коммуникационные модули .....	236
1.1.2.5.1. TCP Server .....	237
1.1.2.5.1.1. Настройка .....	238
1.1.2.5.1.2. Диагностика работы модуля .....	243
1.1.2.5.2. Модуль NetDiag .....	244
1.1.2.5.2.1. Настройка .....	248
1.1.2.5.2.2. Карта адресов .....	251
1.1.2.5.2.3. Диагностика работы модуля .....	252
1.1.2.5.3. Модуль NetDiag2 .....	253
1.1.2.5.3.1. Настройка .....	257
1.1.2.5.3.2. Карта адресов .....	260
1.1.2.5.3.3. Диагностика работы модуля .....	261
1.1.2.5.4. Менеджер SNMP .....	262
1.1.2.5.4.1. Настройка .....	265
1.1.2.5.4.2. Карта адресов .....	267
1.1.2.5.4.3. Преобразование типов данных .....	269

1.1.2.5.4.4. Диагностика работы модуля .....	271
1.1.2.5.5. HUB Модуль .....	276
1.1.2.5.5.1. Настройка .....	277
1.1.2.5.6. SQL Connector .....	280
1.1.2.5.6.1. Настройка .....	281
1.1.2.5.6.2. Карта адресов .....	284
1.1.2.5.6.3. Добавление базы данных .....	286
1.1.2.5.6.4. Диагностика работы модуля .....	296
1.1.2.5.7. Клиент FINS .....	297
1.1.2.5.7.1. Настройка .....	298
1.1.2.5.8. Клиент ПОРТАЛ .....	300
1.1.2.5.8.1. Настройка .....	301
1.1.2.5.9. Siemens S7 Client .....	303
1.1.2.5.9.1. Настройка .....	309
1.1.2.5.9.2. Карта адресов .....	312
1.1.2.5.9.3. Диагностика работы .....	314
1.1.2.5.10. Модуль SMTP .....	319
1.1.2.5.10.1. Настройка .....	320
1.1.2.5.10.2. Настройка списка получателей событий .....	323
1.1.2.5.10.3. Карта адресов .....	325
1.1.2.5.10.4. Диагностика работы модуля .....	326
1.1.2.5.11. Syslog Сервер .....	327
1.1.2.5.11.1. Настройка .....	334
1.1.2.5.11.2. Передача данных .....	337
1.1.2.5.11.3. Карта адресов .....	339
1.1.2.5.11.4. Диагностика работы модуля .....	341
1.1.2.5.12. Syslog Клиент .....	342
1.1.2.5.12.1. Настройка .....	345
1.1.2.5.12.2. Карта адресов .....	349
1.1.2.5.12.3. Диагностика работы модуля .....	352
1.1.2.5.13. ТЭМ-104 .....	353
1.1.2.5.13.1. Настройка .....	356
1.1.2.5.13.2. Карта адресов .....	359
1.1.2.5.13.3. Теплосчетчик .....	361

1.1.2.5.13.4. Модуль: Запросы .....	367
1.1.2.5.13.5. Диагностика работы модуля .....	373
1.1.2.6. МЭК-60870-5.....	374
1.1.2.6.1. Опросчик МЭК 60870-5-101 .....	375
1.1.2.6.1.1. Настройка .....	380
1.1.2.6.1.2. Настройка СОМ порта .....	384
1.1.2.6.1.3. Настройка станции.....	386
1.1.2.6.1.4. Карта адресов .....	389
1.1.2.6.1.5. Диагностика работы модуля .....	391
1.1.2.6.2. Станция МЭК 60870-5-101.....	394
1.1.2.6.2.1. Настройка .....	397
1.1.2.6.2.2. Карта адресов .....	401
1.1.2.6.2.3. Диагностика работы модуля .....	403
1.1.2.6.3. Опросчик МЭК 60870-5-104 .....	404
1.1.2.6.3.1. Настройка .....	410
1.1.2.6.3.2. Добавление станций .....	413
1.1.2.6.3.3. Функциональное содержание .....	419
1.1.2.6.3.4. Карта адресов .....	430
1.1.2.6.3.5. Диагностика работы модуля .....	432
1.1.2.6.4. Станция МЭК 60870-5-104.....	433
1.1.2.6.4.1. Настройка .....	435
1.1.2.6.4.2. Функциональное содержание .....	439
1.1.2.6.4.3. Карта адресов .....	442
1.1.2.6.4.4. Диагностика работы модуля .....	444
1.1.2.6.5. Стандартные типы данных.....	445
1.1.2.6.6. Частные типы данных.....	449
1.1.2.7. МЭК-61850.....	463
1.1.2.7.1. Опросчик МЭК 61850.....	464
1.1.2.7.1.1. Настройка .....	466
1.1.2.7.1.2. Обмен данными с устройством.....	468
1.1.2.7.1.3. Набор данных устройства и сигналы сервера .....	476
1.1.2.7.1.4. Типы данных .....	484
1.1.2.7.1.5. Диагностика работы модуля .....	486
1.1.2.8. Служебные модули.....	487

1.1.2.8.1. Модуль истории .....	488
1.1.2.8.1.1. Настройка .....	491
1.1.2.8.1.2. Диагностика работы модуля .....	494
1.1.2.8.1.3. Устранение неполадок .....	500
1.1.2.8.2. Модуль SnapShot .....	504
1.1.2.8.2.1. Настройка .....	505
1.1.2.8.2.2. Диагностика работы модуля .....	508
1.1.2.8.3. Модуль ESU TU.....	510
1.1.2.8.3.1. Настройка .....	511
1.1.2.8.4. Модуль Data Buffer .....	513
1.1.2.8.4.1. Настройка .....	518
1.1.2.8.4.2. Диагностика работы модуля .....	521
1.1.2.8.5. Модуль Write-VQT.....	522
1.1.2.8.5.1. Настройка .....	524
1.1.2.8.5.2. Диагностика работы модуля .....	526
1.1.2.8.6. Клиент безопасности.....	527
1.1.2.8.6.1. Настройка .....	528
1.1.2.8.7. Контроллер кластера безопасности.....	530
1.1.2.8.7.1. Настройка .....	531
1.1.2.8.8. Модуль вычислений .....	533
1.1.2.8.8.1. Настройка .....	535
1.1.2.8.9. Модуль резервирования.....	537
1.1.2.8.9.1. Настройка .....	539
1.1.2.8.9.2. Диагностика работы модуля .....	541
1.1.2.8.10. Модуль ProcessMonitor .....	542
1.1.2.8.10.1. Настройка .....	543
1.1.2.8.10.2. Диагностика работы модуля .....	545
1.1.3. Адресное пространство .....	546
1.1.3.1. Типы сигналов.....	547
1.1.3.2. Типы данных .....	548
1.1.3.3. Качество сигналов .....	549
1.2. Astra.AccessPoint .....	551
1.2.1. Настройка .....	555
1.2.1.1. Настройки HubModule.....	556



1.2.1.2. Настройка источника данных .....	558
1.2.2. Получение данных .....	564
1.2.2.1. Получение данных по OPC DA .....	565
1.2.2.2. Получение данных по OPC AE .....	566
1.2.2.3. Получение данных по OPC UA .....	567
1.2.2.3.1. Получение оперативных данных.....	568
1.2.2.3.2. Получение исторических данных .....	572
1.2.2.4. Квитирование событий .....	577
1.2.3. Диагностика работы .....	578
1.3. Astra.Imitator .....	580
1.3.1. Настройка конфигурационных файлов .....	581
1.3.1.1. Astra.Historian .....	582
1.3.1.2. Astra.Domain .....	583
1.3.2. Добавление Astra.Imitator в проект .....	585
1.3.3. Воспроизведение истории.....	599
1.3.3.1. Подключение к серверу .....	600
1.3.3.2. Сервисные сигналы .....	604
1.3.3.3. Просмотр истории .....	611
1.4. Astra.Domain.....	624
1.4.1. Конфигурирование .....	625
1.4.2. Диагностика работы .....	627

# 1. Инфраструктура

**Инфраструктура** – это программные компоненты (службы), которые обеспечивают функционирование системы.

К инфраструктурным компонентам относятся:

Компонент	Версия	Описание
<a href="#">Astra.Server</a>	2.1.0.10	Компонент для сбора, обработки и предоставления обработанных данных
<a href="#">Astra.AccessPoint</a>	2.1.0.10	Компонент реализует функции сервера приложений и межуровневого транспорта
<a href="#">Astra.Imitator</a>	2.1.0.10	Компонент для воспроизведения истории технологического процесса на мнемосхемах
<a href="#">Astra.Domain</a>	1.2.8.3	Компонент для выполнения инфраструктурных функций и объединения узлов исполняющих компонентов в единую сеть Astra.Net

# 1.1. Astra.Server

**Astra.Server** – программный компонент, выполняющий следующие задачи:

- › сбор данных;
- › логическая обработка данных в режиме реального времени;
- › сохранение полученной технологической информации в архивах;
- › предоставление данных клиентам и сторонним системам по различным протоколам и спецификациям;
- › генерация событий и тревог на основе полученных данных.
- › сохранение текущих значений сигналов в файл-срезы XML-формата;
- › сохранение текущих значений сигналов в файл-срезы бинарного формата;
- › диагностика сетевых устройств;
- › предоставление данных для записи в сервер истории.

## Принцип работы

Astra.Server построен по модульному принципу, что позволяет конфигурировать его в зависимости от выполняемых задач и не создавать лишней нагрузки. Astra.Server может содержать до 64 модулей.

На одном компьютере возможно функционирование нескольких экземпляров Astra.Server.

Установка нескольких экземпляров Astra.Server на одном компьютере решает задачу конвертации протоколов. Количество серверов на одной машине ограничено её производительностью.



Совместная нагрузка процессора при наличии нескольких экземпляров сервера не должна превышать 70%.

Работа Astra.Server возможна без открытия сеанса пользователя.

Полноценное функционирование Astra.Server не ограничено временными границами. Перезапуск сервера требуется только для выполнения конфигурационных настроек.

# Ядро

Ядро Astra.Server является центральным компонентом сервера. Предназначено для реализации инфраструктуры сервера, интерфейсов работы с модулями, сигналами и их свойствами, остальными подсистемами. Ядро может производить значимые логические вычисления, требующие наибольшей скорости вычислений. Все вычисления производятся по описанным при конфигурировании алгоритмам.

Основные функции ядра:

- › пересчет значений из физических значений в инженерные и в обратном направлении. При пересчете используются линейная и линейная с изломом зависимости;
- › выполнение алгоритмов по событию, таймеру и расписаниям;
- › управление запуском и остановом модулей при старте и в процессе работы сервера;
- › управление состоянием сервера в рамках резервирования;
- › запись и чтение данных из ОБД;
- › управление модулями, отправка и принятие уведомлений об изменении значений сигналов.

## Сбор данных

Astra.Server обеспечивает опрос источников данных по различным протоколам и спецификациям.

Протокол/спецификация	Модуль сервера	Поддержка в ОС	
		Windows	Linux
ГОСТ Р МЭК 60870-5-101	IEC-101 Master	+	+
ГОСТ Р МЭК 60870-5-104	IEC-104 Master	+	+
ГОСТ Р МЭК 61850	IEC-61850 Client	+	+
Modbus TCP	Modbus TCP Master	+	+
Modbus RTU	Modbus RTU Master	+	+
OPC DA	OPC DA Client	+	
OPC HDA	OPC HDA Client	+	
OPC UA	OPC UA Client	+	+
SQL	SQL Connector	+	+
SNMP	SNMP Manager	+	+
Syslog	Syslog Server	+	+

## Логическая обработка данных

Возможности:

- › пересчет значений из физических в инженерные и обратно (по линейной и линейной с изломом зависимостям);
- › пересчет значений сигналов по формуле;
- › выполнение алгоритмов по событию, таймеру или расписанию;
- › вызов функций из внешних динамических библиотек;
- › перехват генерируемых событий и тревог.
- › разбор буфера для выделения кода технологического объекта и кода события/тревоги (модуль Data Buffer);
- › опциональное изменение свойств сигнала Value, Quality или Timestamp (модуль Write VQT).

Модуль сервера	Поддержка в ОС	
	Windows	Linux
Logics Module	+	+
Data Buffer	+	+
Write VQT	+	+

Алгоритмы модуля логики составляются на специальном скриптовом языке Astra.Om.

## Предоставление данных

Astra.Server способен предоставлять данные по различным коммуникационным протоколам и спецификациям.

Протокол/спецификация	Модуль сервера	Поддержка в ОС	
		Windows	Linux
ГОСТ Р МЭК 60870-5-101	IEC-101 Slave	+	+
ГОСТ Р МЭК 60870-5-104	IEC Slave	+	+
Modbus TCP	Modbus TCP Slave	+	+
Modbus RTU	Modbus RTU Slave	+	+
OPC DA	OPC DA Server	+	
OPC HDA	OPC HDA Server	+	
OPC AE	OPC AE Server	+	
OPC UA	OPC UA	+	+
TCP	TCP Server	+	+
Файловый интерфейс	TCP Server	+	+

## Генерация событий

Сервер может генерировать и предоставлять пользователям сообщения о событиях и тревогах.

Модуль сервера	Поддержка в ОС	
	Windows	Linux
OPC AE Server	+	+
Модуль рассылки событий	+	+

Возможности:

- › генерация событий в рамках спецификации OPC AE;
- › предоставление информации о событиях в рамках спецификации OPC DA;
- › отправка информации о событиях по электронной почте (модуль рассылки событий).

Сервер генерирует события по нескольким алгоритмам срабатывания:

- › дискретный переключатель;
- › перечисление;
- › отклонение;
- › по уровню.



В ОС Linux модуль OPC AE Server генерирует события, но не предоставляет их.



# Резервирование



Astra.Server реализует горячее резервирование.

При горячем резервировании основной сервер работает в полнофункциональном режиме, резервный сервер не отправляет управляющих команд на контроллеры. Резервный сервер находится в состоянии готовности и при Резервном переходе берёт на себя функции основного сервера.

Благодаря резервированию серверов повышается их надёжность и обеспечивается сохранность собранных технологических данных в случае сбоев в работе. Включение в работу резервного сервера может производиться вручную, либо автоматически в случае отсутствия связи с одним из серверов.

При горячем резервировании система позволяет настроить репликацию данных между резервируемыми серверами для поддержания оперативной базы данных резервного сервера в актуальном состоянии.

## 1.1.1. Конфигурирование

Настройка Astra.Server осуществляется конфигуратором Astra.AStudio.



Конфигурация Astra.Server хранится в бинарном файле AstraServer.cfg в директории установки.

Конфигурация Astra.Server характеризуется:

- › номером версии конфигурации;
- › идентификатором конфигурации.

Номер версии конфигурации увеличивается при любых изменениях конфигурации сервера.

Текущий номер версии конфигурации можно узнать через:

- › сервисное приложение Статистика;
- › сигнал с тегом Service.Config.Version, значение которого можно посмотреть любым OPC клиентом.

Конфигурация сервера имеет уникальный идентификатор, который содержится в сигнале с тегом Service.Id.Str. Данные сигналы можно смотреть любым OPC клиентом.

Также идентификатор можно узнать, открыв экспортированный файл конфигурации в текстовом редакторе.

Конфигурация Astra.Server содержит:

- › структура дерева сигналов - полный список сигналов, которые добавлены в конфигурацию;
- › инициализирующие значения сигналов - значения, которые сигналы принимают при запуске сервера;
- › модульный состав сервера - полный список модулей, которые добавлены в конфигурацию;
- › конфигурационные параметры модулей и их значения;

› пароль доступа к серверу.



Все изменения значений сигналов и свойств сигналов, которые произведены с помощью OPC клиента, не записываются в файл AstraServer.cfg. После перезапуска Astra.Server сигналы примут свои исходные инициализирующие значения.



В OPC клиентах измененная конфигурация сервера будет доступна только после перезапуска Astra.Server.

Чтобы защитить конфигурацию сервера от потери данных, создавайте резервные копии текущей конфигурации. Сохраненные конфигурационные файлы могут быть обратно импортированы.

Файлы формата \*.csv открываются с помощью MS Excel и имеют вид структурированной таблицы.



После всех изменений конфигурации или после импорта конфигурации необходимо перезапустить Astra.Server.

# Резервное копирование конфигурации сервера

Создание резервной копии конфигурации Astra.Server возможно выполнить вручную и/или автоматически.

В автоматическом режиме резервное копирование выполняется в соответствии с настройками расписания, указанными в файле настроек сервера Astra.Server.xml.

Чтобы создать резервную копию конфигурации через сервисное приложение Конфигуратор, в окне приложения выберите команду Сервер → Создать архивную копию конфигурации.

Чтобы создать резервную копию конфигурации сервера с помощью сервисного приложения Управляющий, в окне приложения нажмите кнопку Резервное копирование.

Настройки сохранения резервных копий в ручном и в автоматическом режиме задаются в файле настроек сервера Astra.Server.xml теге <Backup>:

- Path - название каталога, в который сохраняются резервные копии;
- Time - время автоматического выполнения резервного копирования;
- StorageDepth - длительность хранения резервных копий на диске. Указывается в сутках, значение по умолчанию 14 суток.

Чтобы восстановить конфигурацию Astra.Server из резервной копии, выполните следующие действия:

1. файлу резервной копии задайте имя, указанное в файле Astra.Server.xml (в директории установки Astra.Server) в атрибуте Filename элемента Storage (по умолчанию AstraServer.cfg);
2. замените текущий файл конфигурации Astra.Server.

## 1.1.1.1. Ручная настройка

Ручная настройка может потребоваться после установки Astra.Server для изменения стандартных параметров:

- › расположение и имя бинарного файла конфигурации сервера;
- › номер порта подключения к серверу;
- › время выполнения автоматического резервного копирования;
- › папка хранения резервных копий конфигурации сервера;
- › папка хранения журналов работы модулей.

Параметры настройки сервера задаются в файле Astra.Server.xml, расположенном:

- › в ОС Windows в папке C:\Program Files\AstraRegul\Astra.Server\Server;
- › в ОС Linux в директории /opt/AstraRegul/Astra.Server/Server.

Файл по умолчанию имеет следующий вид:



```
<?xml version="1.0" encoding="windows-1251" ?>
<configuration>
  <install ServiceName="Astra.Server" ExeName="Astra.Server.exe">
    <ComServers>
      <OPCDA ProgID="Astra.OPCDA Server" CLSID="{4AF8C105-174A-429D-9DCC-F440...}>
      <OPCAE ProgID="Astra.OPCAE Server" CLSID="{D09EE44A-B3F9-4BC9-B033-71CA...}>
      <HDA ProgID="Astra.HDA Server" CLSID="{8002740A-886F-1488-2280-42058D6D...}>
    </ComServers>
  </install>
  <Storage Filename="AstraServer.cfg" />
  <Connection Port="4572" />
  <Backup Path="..\Backups" Time="00:00" StorageDepth="14" />
  <Log Path="..\Logs" />
  <Instance ID="0CBE366F-93A6-40E5-A9AE-F5CB2E8C3B28" />
  <MasterPassword Cipher="icj+2PDqNPRFKuxi6q+XmyvU9QSEyS3iDtChbTKBIBWi...
  Jz47D5UB02xeBurGxCB0bPbZi8KSeX09CAZAIw8lyRYkPICNhWA+NNq5X+ADZScDJ...
  g7+2t3PiCjJWoxZAwKpLWu1hmw9/QdikYQ9E" />
</configuration>
```

Допустимо изменение значений следующих атрибутов:

Родительский элемент	Атрибут	Описание
Storage	Filename	Полный путь или название файла конфигурации сервера
Connection	Port	Номер порта подключения к серверу
Backup	Path	Папка хранения автоматически создаваемых резервных копий конфигурации. <u>По умолчанию:</u> C:\Program Files\AstraRegul\Astra.Server\Backups  <u>Формат имени файла:</u> [Год]_[Месяц]_[Дата] [номер копии] [имя пользователь].backup
Backup	Time	Время автоматического создания резервной копии текущей конфигурации
Backup	StorageDepth	Длительность хранения резервных копий конфигурации сервера, указывается в сутках.
Log	Path	Папка хранения журналов работы модулей сервера.
MasterPassword	Cipher	Запись о пароле доступа к экземпляру сервера в зашифрованном виде, если пароль был задан в сервисном приложении Конфигуратор.



После внесения изменений в файл Astra.Server.xml сохраните его и перезапустите службу Astra.Server.

## 1.1.1.2. Защита от несанкционированного доступа

Установка пароля доступа к Astra.Server предотвращает следующие несанкционированные действия:

- › управление сервером или резервной парой серверов через сервисное приложение Управляющий;
- › модификацию конфигурации через сервисное приложение Конфигуратор;
- › просмотр статистической информации через сервисное приложение Статистика;
- › обмен данными с Astra.AccessPoint.



Пароль хранится в файле Astra.Server.xml в зашифрованном виде.

## Шифрование паролей

Пароль доступа к серверу, а также пароли, указываемые в параметрах некоторых модулей, хранятся в зашифрованном виде.

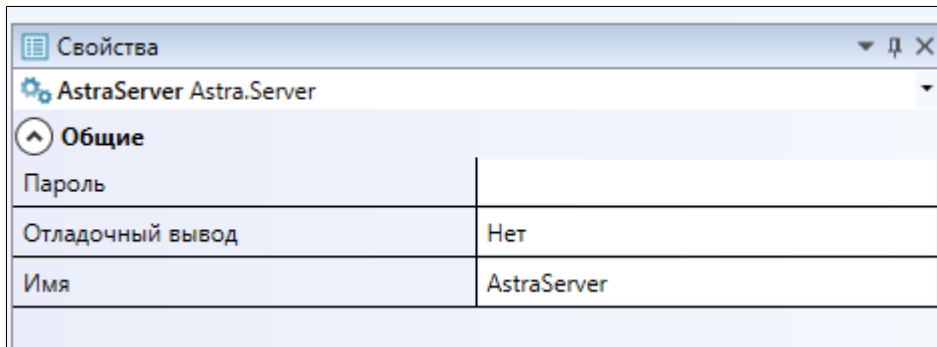
Для шифрования паролей используется асимметричный алгоритм шифрования RSA с ключом размером 1024 бита:

- › введённый пароль шифруется с помощью открытого ключа и сохраняется в зашифрованном виде;
- › сохранённый пароль расшифровывается только когда он нужен в открытом виде (например, при сравнении паролей). Расшифровка выполняется с помощью закрытого ключа. Закрытый ключ зашифрован алгоритмом blowfish и в открытом виде нигде не хранится.

## Настройка доступа к серверу по паролю

Для предотвращения несанкционированной модификации конфигурации Astra.Server необходимо настроить доступ к серверу по паролю.

Пароль для экземпляра Astra.Server можно задать в свойствах объекта в конфигураторе Astra.AStudio.



Пароль хранится в файле Astra.Server.xml в зашифрованном виде.



Чтобы сбросить пароль, удалите в конфигурационном файле Astra.Server.xml значение атрибута Cipher элемента MasterPassword (показано на рисунке ниже).



Чтобы убрать парольную защиту для сервера, полностью удалите в конфигурационном файле строку с элементом MasterPassword.



```
Astra.Server — Блокнот
Файл Правка Формат Вид Справка
<configuration>
  <install ServiceName="Astra.Server" ExeName="Astra.Server.exe">
    <ComServers>
      <OPCDA ProgID="Astra.OPCDAServer" CLSID="{28A2AD9C-C45E-4C6b-A0C3-6E363F99CA72}" />
      <OPCAE ProgID="Astra.OPCAEServer" CLSID="{0CAEA48A-D7E6-44A4-85FD-C27836727D07}" />
      <HDA ProgID="Astra.HDAServer" CLSID="{8002740A-886F-1488-2280-42058D6D5CA8}" />
    </ComServers>
  </install>
  <Storage Filename="AstraServer.cfg" />
  <Connection Port="4572" />
  <Backup Path="..\Backups" Time="00:00" StorageDepth="14" />
  <Log Path="..\Logs" />
  <Instance ID="6A2960B6-9C82-44C3-801A-DA2E268926AC" />
  <Culture LangID="ru-RU" />
</configuration>
```

## 1.1.2. Модули

Все модули обладают общим и уникальным наборами параметров. Уникальные параметры узла конфигурации модуля содержатся в группе Дополнительные и настраиваются для каждого модуля индивидуально.

Группа модулей	Описание
<a href="#">BACnet</a>	Коммуникационные модули для передачи данных по протоколу BACnet
<a href="#">EtherNet/IP</a>	Коммуникационный модуль для передачи данных по протоколу EtherNet/IP.
<a href="#">Modbus</a>	Коммуникационные модули для передачи данных по протоколу Modbus
<a href="#">OPC</a>	Модули для осуществления обмена данными по протоколу OPC
<a href="#">Коммуникационные модули</a>	Модули для осуществления обмена данными по различным протоколам
<a href="#">МЭК-60870-5</a>	Модули для осуществления обмена данными по протоколам стандарта МЭК-60870-5
<a href="#">МЭК-61850</a>	Модули для осуществления обмена данными по протоколам стандарта МЭК-61850
<a href="#">Служебные модули</a>	Служебные модули для осуществления диагностики и мониторинга сервера

## Активность модуля

Активность модуля задается при настройке общего параметра работы модуля Активность.

Активность модуля:

- › Да – модуль запущен
- › Нет – модуль остановлен

Текущая активность модуля отображается в OPC клиенте в сигнале с тегом Service.Modules.[Имя модуля].Active:

- › true – модуль запущен;
- › false – модуль остановлен.

## Общие параметры

Общие параметры узла конфигурации модуля содержатся в группе Общие:

- › Параметр.
- › Описание.
- › Имя модуля.
- › Название модуля.
- › Идентификатор модуля.
- › Идентификатор модуля в Astra.Server.

Значения общих параметров можно посмотреть в любом OPC-клиенте по тегу: Service.Modules.[Имя модуля].[Имя параметра].

## Параметры журналирования

У каждого модуля можно настроить параметры журналирования.

Для каждого модуля создается отдельный файл:

- › в ОС Windows в папке C:\Program Files\AstraRegul\Astra.Server\Logs;
- › в ОС Linux в директории /opt/AstraRegul/Astra.Server/Logs.

Ведение журнала работы для каждого модуля настраивается при конфигурировании модуля. Для каждого модуля создается отдельный файл. Расположение файлов журналов работы указывается в файле Astra.Server.xml в атрибуте Path элемента Log (по умолчанию Logs в папке/директории установки Astra.Server).

Имя файла совпадает с названием модуля. Расширение файлов журнала \*.aplog. Размер файла журнала работы модуля устанавливается вручную в конфигурации каждого модуля и по умолчанию равен 10 Мбайт. При заполнении заданного размера файла журнала создается другой файл журнала (количество дополнительных журналов зависит от количества журналов, настроенных при конфигурации модуля), в который сохраняются все записи текущего журнала. Текущий журнал, в свою очередь, очистится, и запись работы модуля возобновится в нем. При последующем заполнении текущего файла журнала создается еще один файл журнала, если в конфигурации модуля он был предусмотрен, либо уже созданный файл журнала будет перезаписан. После перезагрузки Astra.Server записи в журнале работы модуля не удаляются, а добавляются к уже имеющимся в журнале.

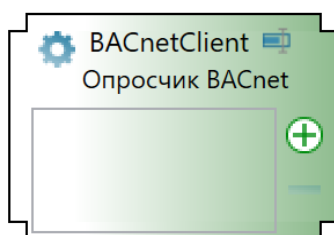
Параметр	Описание
Вести журнал работы модуля	Ведётся ли журнал работы модуля: <ul style="list-style-type: none"><li>› Да</li><li>› Нет</li></ul>

<p>Размер журнала работы модуля, МБ</p>	<p>Ограничение на размер файла журнала работы модуля в мегабайтах. При достижении максимального размера создается новый файл, копия старого файла хранится на рабочем диске.</p>
<p>Количество дополнительных журналов работы</p>	<p>Количество файлов заполненных журналов работы модуля. Значение: от 1 до 255.</p>
<p>Уровень трассировки в журнал приложений</p>	<p>Типы сообщений, которые фиксируются в журнал приложений:</p> <ul style="list-style-type: none"> <li>➤ Предупреждения и аварийные сообщения – логические ошибки, ошибки работы модуля. Предупреждения содержат некритичные ошибки. Аварийные сообщения информируют об ошибках, которые влияют на работоспособность службы</li> <li>➤ Информационные сообщения – основная информация о работе модуля.</li> <li>➤ Отладочные сообщения – детальная информация о работе модуля.</li> </ul> <p>Вышестоящий уровень входит в состав нижестоящего: если выбрано Информационные сообщения, то в журнал фиксируются Предупреждения и аварийные сообщения и Информационные сообщения</p>

## 1.1.2.1. BACnet

Модуль	Описание
<a href="#">Клиент BACnet</a>	Модуль для обмена данными между Astra.Server и устройствами, поддерживающими протокол BACnet

## 1.1.2.1.1. Клиент ВАСnet



Модуль ВАСnet Client предназначен для обмена данными между Astra.Server и устройствами, поддерживающими протокол ВАСnet.

Функции модуля ВАСnet Client:

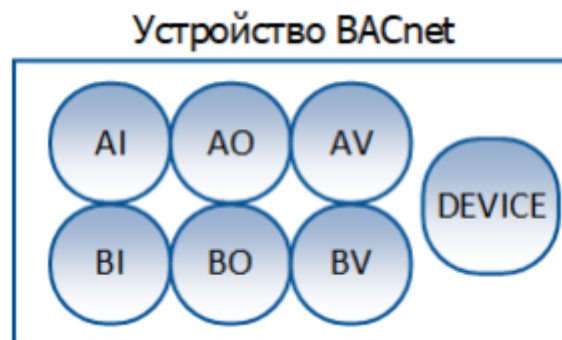
- › сбор данных с устройств ВАСnet и сохранение полученных значений в сигналы Astra.Server;
- › подача команд управления устройствам ВАСnet.

## 1.1.2.1.1.1. Устройство ВАСnet

Устройство ВАСnet имеет логическую структуру, состоящую из набора объектов ВАСnet. Полный перечень объектов приведен в спецификации ВАСnet. Набор объектов для каждого устройства ВАСnet индивидуален и приведен в документации на устройство.



Устройство состоит из набора объектов: «Analog\_Input» («AI»), «Analog\_Output» («AO»), «Analog\_Value» («AV»), «Binary\_Input» («BI»), «Binary\_Output» («BO»), «Binary\_Value» («BV») и устройство «DEVICE».

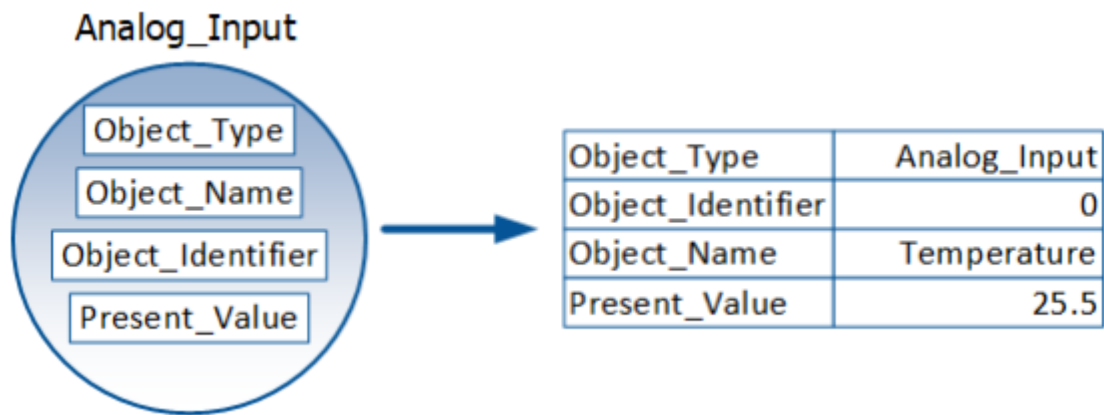


Каждый объект ВАСnet имеет набор свойств, которые содержат информацию об объекте и управляют его работой. Полный перечень свойств объектов приведен в спецификации ВАСnet. Набор используемых свойств объекта для каждого устройства ВАСnet индивидуален и приведен в документации на устройство.



Объект «AI» имеет набор свойств: тип объекта «Object\_Type», идентификатор объекта «Object\_Identifier», название объекта «Object\_Name» и текущее значение «Present\_Value». Данный объект соответствует измеряемой температуре.





## Обмен данными с устройством

Обмен данными между модулем BACnet Client и устройством BACnet выполняется по сети Ethernet (протокол UDP) в режиме запрос-ответ. Инициатором запроса является модуль BACnet Client. Запросы и ответы представляют собой кадры данных. Для модуля BACnet Client запрос, отправляемый устройству, является исходящим кадром, а ответ, полученный от устройства – входящим кадром.

## Обнаружение устройства

После запуска модуль BACnet Client начинает поиск в сети устройства BACnet, отправляя запрос обнаружения устройства Who-Is с заданной периодичностью. После получения от устройства ответа I-Am устанавливается связь с устройством и модуль готов к обмену данными.

После установления связи с устройством модуль BACnet Client отправляет устройству запросы Who-Is для подтверждения наличия связи в периоды, когда опрос устройства не ведётся.

## Получение данных от устройства

Модуль выполняет опрос устройства циклически с паузой между циклами опроса. Для получения данных от устройства модуль BACnet Client отправляет устройству запрос ReadPropertyMultiple-Request, содержащий список свойств,

значения которых требуется получить. Для каждого объекта отправляется отдельный запрос.

От устройства модуль получает ответ ReadPropertyMultiple-ACK, содержащий значения запрошенных свойств. Полученные значения модуль записывает в сигналы Astra.Server. Если ответ от устройства не получен за заданный период ожидания ответа от устройства, то связь с устройством считается потерянной.

## **Подача команд управления**

Подача команды управления - это отправка значения сигнала Astra.Server в устройство. Команды имеют более высокий приоритет, чем опрос устройства, поэтому при подаче команды модуль прерывает опрос и передаёт команду управления устройству.

Для отправки значения сигнала Astra.Server в устройство модуль BACnet Client отправляет устройству запрос WriteProperty-Request, содержащий свойство объекта и значение, которое требуется записать в устройство.

От устройства модуль получает ответ WriteProperty-ACK, который в случае успешной записи значения в устройство не содержит ошибок. Если значение в устройство записать не удалось, то ответ WriteProperty-ACK содержит информацию об ошибке.

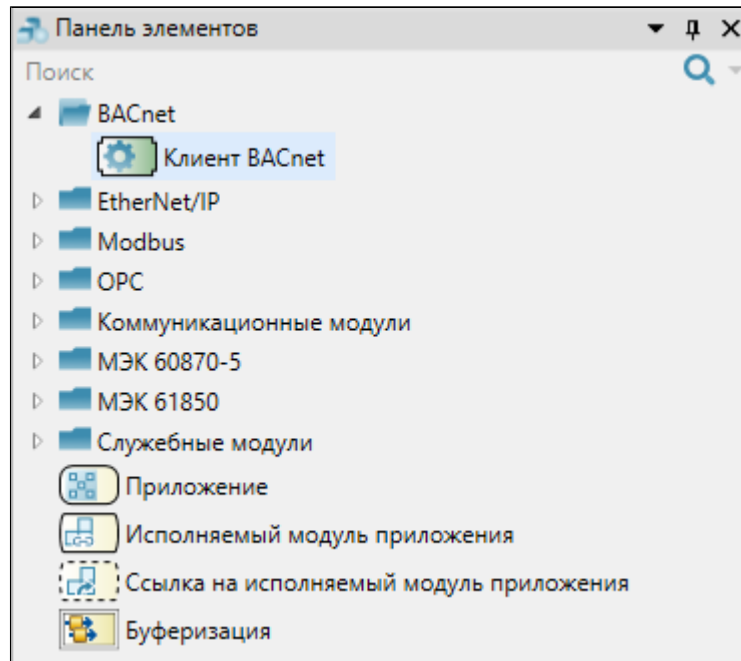
Для команд управления возможна настройка сигналов доставки. Значение сигнала доставки определяет состояние отправленной команды.

## **Работа модуля в резерве**

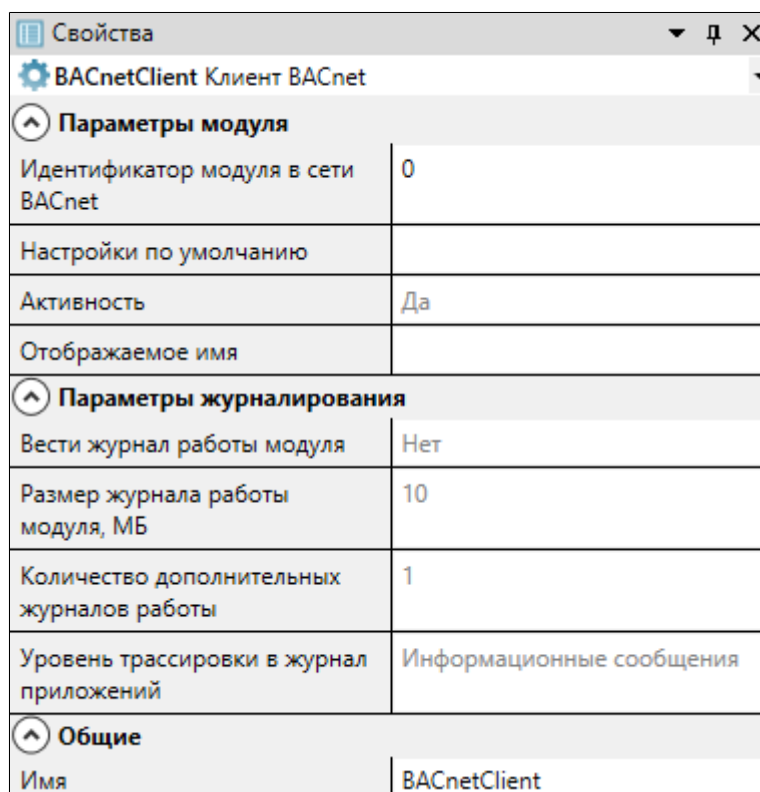
В режиме РЕЗЕРВ модуль BACnet Client не ведёт опрос устройств и не отправляет команды управления, но отправляет запросы обнаружения устройства "Who-Is" и принимает входящие уведомления "I-Am".

## 1.1.2.1.1.2. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.



## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: > Да – модуль запущен; > Нет – модуль остановлен. Управляется служебным сигналом «Active.Set»
Отображаемое имя	Отображаемое имя модуля (отображается в тегах служебных сигналов)
Идентификатор модуля в сети ВАСnet	Идентификатор, указанный в настройках устройства, к которому выполняется подключение
Настройки по умолчанию	Выбор файла с готовой конфигурацией

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.1.1.2.1. Настройка обмена данными с устройством

Чтобы настроить обмен данными между Astra.Server и устройством ВАСnet, необходимы следующие исходные данные:

1. Для настройки параметров устройства ВАСnet в конфигурации - идентификатор устройства. Информация об устройстве содержится в свойствах обязательного для всех устройств объекта Device. Идентификатор устройства - значение свойства Object\_Identifier объекта Device.

2. Для настройки адресов сигналов в карте адресов или редакторе адреса:

2.1 Идентификатор типа объекта. Для настройки адреса сигнала можно использовать одно из обозначений типа объекта:

- целочисленный идентификатор типа объекта - указан в спецификации ВАСnet;
- строковый идентификатор - для некоторых типов объектов модулем ВАСnet Client поддерживаны строковые идентификаторы.

2.2 Номер экземпляра объекта. Содержится в значении свойства Object\_Identifier объекта.

2.3 Идентификатор свойства объекта. Для настройки адреса сигнала можно использовать одно из обозначений свойства объекта:

- целочисленный идентификатор свойства объекта - указан в спецификации ВАСnet;
- строковый идентификатор - для некоторых свойств объектов модулем ВАСnet Client поддерживаны строковые идентификаторы.

2.4 Тип данных свойства объекта - приведен в описании свойств объекта в спецификации ВАСnet. В зависимости от типа данных свойства объекта выбирается тип сигнала и протокольный тип Astra.Server для получения

значения или отправки команды. Для настройки адреса используется соответствующий протокольный тип.

## 1.1.2.1.1.3. Протокольные типы

В таблицах приведены типы свойств объектов в спецификации BACnet, соответствующие им типы сигналов в Astra.Server и протокольные типы для настройки адресов сигналов Astra.Server:

- входящих сигналов - для получения значений свойств объектов устройства BACnet;
- исходящих сигналов - для отправки команд управления в устройство BACnet.

### Входящие сигналы

Протокольный тип	Тип сигнала в Astra.Server	Тип свойства, значение которого требуется получить от устройства	Описание
BOOLEAN	bool	BOOLEAN	Логическое значение (true; false)
Signed	int4	INTEGER, INTEGER16	Целое знаковое 4 байта (от -2 147 483 648 до 2 147 483 647)
Unsigned	uint4	Unsigned, Unsigned8, Unsigned16, Unsigned32	Целое беззнаковое 4 байта (от 0 до 4 294 967 295)
Enum	uint4	ENUMERATED	Численные перечисления
EnumStr	string		Строковые перечисления
REAL	float	REAL	Вещественное 4 байта ([ $\pm 1.5 \times 10^{-45}$ ; $\pm 3.4 \times 10^{38}$ ]. Точность 6-9 цифр)
double	double	Double	Вещественное 8 байт ([ $\pm 5.0 \times 10^{-324}$ ; $\pm 1.7 \times 10^{308}$ ]. Точность 15-17 цифр)

CharacterString	string	CharacterString	Текстовая строка
OctetString	string	OCTET STRING	Массив байтов в виде строки
BitString	string	BIT STRING	Массив битов в виде строки
ObjectIdStr	string	BACnetObjectIdentifier	Идентификатор объекта в формате <object_type>:<instance_num> > object_type - идентификатор типа объекта; > instance_num - номер экземпляра объекта.
TimeStr	string	Time	Время в формате HH:MM:SS.ss > HH - часы; > MM - минуты; > SS - секунды; > ss - миллисекунды.
DateStr	string	Date	Дата в формате YYYY:MM:DD > YYYY - год; > MM - месяц; > DD - день.

## Исходящие сигналы

Протокольный тип	Тип сигнала в Astra.Server	Тип свойства, значение которого требуется получить от устройства	Описание
BOOLEAN	bool	BOOLEAN	Логическое значение (true; false)
Signed	int4	INTEGER, INTEGER16	Целое знаковое 4 байта (от -2 147 483 648 до 2 147 483 647)



Unsigned	uint4	Unsigned, Unsigned8, Unsigned16, Unsigned32	Целое беззнаковое 4 байта (от 0 до 4 294 967 295)
Enum	uint4	ENUMERATED	Перечисления (от 0 до 4 294 967 295)
REAL	float	REAL	Вещественное 4 байта ([ $\pm 1.5 \times 10^{-45}$ ; $\pm 3.4 \times 10^{38}$ ]. Точность 6-9 цифр)
double	double	Double	Вещественное 8 байт ([ $\pm 5.0 \times 10^{-324}$ ; $\pm 1.7 \times 10^{308}$ ]. Точность 15-17 цифр)
CharacterString	string	CharacterString	Текстовая строка

## 1.1.2.1.1.4. Строковые идентификаторы объектов и свойств

### Типы объектов

Тип объекта в спецификации BACnet	Идентификатор объекта в спецификации BACnet	Строковый идентификатор объекта в Astra.Server
«Analog_Input»	0	AI
«Analog_Output»	1	AO
«Analog_Value»	2	AV
«Binary_Input»	3	BI
«Binary_Output»	4	BO
«Binary_Value»	5	BV
«Device»	8	DEVICE

### Свойства объекта

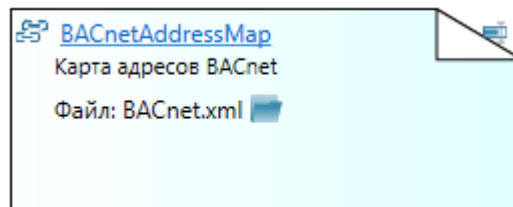
Свойство объекта в спецификации BACnet	Идентификатор свойства объекта в спецификации BACnet	Строковый идентификатор свойства в Astra.Server
«APDU_Timeout»	11	APDU_TIMEOUT
«Application_Software_Version»	12	APPLICATION_SOFTWARE_VERSION
«Database_Revision»	155	DATABASE_REVISION
«Description»	28	DESCRIPTION
«Device_Address_Binding»	30	DEVICE_ADDRESS_BINDING
«Event_State»	36	EVENT_STATE
«Firmware_Revision»	44	FIRMWARE_REVISION

«Location»	58	LOCATION
«Max_APDU_Length_Accepted»	62	MAX_APDU_LENGTH_ACCEPTED
«Model_Name»	70	MODEL_NAME
«Number_Of_APDU_Retries»	73	NUMBER_OF_APDU_RETRIES
«Object_Identifier»	75	OBJECT_IDENTIFIER
«Object_List»	76	OBJECT_LIST
«Object_Name»	77	OBJECT_NAME
«Object_Type»	79	OBJECT_TYPE
«Out_Of_Service»	81	OUT_OF_SERVICE
«Polarity»	84	POLARITY
«Present_Value»	85	PRESENT_VALUE
«Priority_Array»	87	PRIORITY_ARRAY
«Protocol_Object_Types_Supported»	96	PROTOCOL_OBJECT_TYPES_SUPPORTED
«Protocol_Revision»	139	PROTOCOL_REVISION
«Protocol_Services_Supported»	97	PROTOCOL_SERVICES_SUPPORTED
«Protocol_Version»	98	PROTOCOL_VERSION
«Reliability»	103	RELIABILITY
«Relinquish_Default»	104	RELINQUISH_DEFAULT
«Segmentation_Supported»	107	SEGMENTATION_SUPPORTED
«State_Text»	110	STATE_TEXT
«Status_Flags»	111	STATUS_FLAGS
«System_Status»	112	SYSTEM_STATUS
«Units»	117	UNITS
«Vendor_Identifier»	120	VENDOR_IDENTIFIER
«Vendor_Name»	121	VENDOR_NAME

## 1.1.2.1.1.5. Карта адресов

Для настройки сигналов модуля BACnet Client используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов BACnet в исполняемое приложение;



2. Откройте редактор карты адресов.

	Сигнал	Тип	Привязка	Тип объекта	Экземпляр объекта	Свойство объекта	Индекс	Протокольный тип
	uint4	uint4	непосредственно	Analog Input	0	OBJECT_NAME	1	Unsigned
	int4	int4	непосредственно	Analog Input	0	DESCRIPTION	2	Signed

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>› Непосредственно;</li><li>› Только чтение;</li><li>› Не привязан.</li></ul>
Тип объекта	Указание типа объекта по спецификации BACnet: <ul style="list-style-type: none"><li>› Analog Input</li><li>› Analog Output</li><li>› Analog Value</li><li>› Binary Input</li><li>› Binary Output</li><li>› Binary Value</li><li>› Device</li></ul>

Экземпляр объекта	Номер экземпляра объекта в устройстве
Свойство объекта	Идентификатор свойства объекта
Индекс	Индекс в массиве данных свойств объекта (необязательный параметр)
Протокольный тип	<p>Тип, значение которого выбирается в зависимости от направления сигнала и типа данных свойства объекта в устройстве:</p> <ul style="list-style-type: none"> <li>➤ IN_* - входящий сигнал;</li> <li>➤ OUT_* - исходящий сигнал;</li> <li>➤ INTERNAL_CTRLRES - сигнал доставки;</li> </ul> <p>где * - протокольный тип в зависимости от типа данных свойства объекта.</p>

## 1.1.2.1.1.6. Значения сигнала доставки

Значение сигнала доставки определяет состояние отправленной команды. Возможные значения сигнала доставки приведены в таблице:

<b>Значение</b>	<b>Состояние команды управления</b>
1	Команда успешно помещена в очередь на отправку
3	Команда исполнена
-1	Команда не исполнена (плохое качество, неверный формат данных, несоответствующий режим работы модуля)
-2	Нет связи с устройством
-3	Переполнена очередь данных на отправку
-4	Протокольная ошибка подачи команды управления
-5	Внутренняя ошибка модуля

## 1.1.2.1.1.7. Диагностика работы модуля

### Журнал работы

Модуль ВАСnet Client ведёт журнал работы, в который записывается информация об обмене данными с устройствами и работе модуля.



Чтобы модуль вёл журнал работы, в общих параметрах модуля в Конфигураторе или в свойствах опросчика в Astra.AStudio установите параметру Вести журнал работы модуля значение «Да» или установите сервисному сигналу модуля «FrameLogEnable.Set» значение «true».

Журнал работы модуля сохраняется в файл <имя модуля>.aplog по умолчанию:

› в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

› в Linux системах в директории:



/opt/AstraRegul/Astra.Server/Logs.

Для просмотра журнала работы модуля используется сервисное приложение Просмотрщик лога кадров.

№	Дата	Время	Описание	У.	Отправитель	Получатель	Object Identifier	Property	Value
47948	27.07.2023	16:24:41:323	DEV 1 Обработка ответа на запрос чтения (requestID=...				OBJECT_ANALOG_INP...	LOW_LIMIT (59)	65
47949	27.07.2023	16:24:42:303	Исходящий кадр (ReadPropertyMultiple-Request)	1	172.16.145.81:47808	172.16.145.90:4	OBJECT_ANALOG_INP...	HIGH_LIMIT (45)	14
47950	27.07.2023	16:24:42:303	Interface: Отправлен запрос на чтение (requestID=143,...				OBJECT_ANALOG_INP...	PRESENT_VALUE (85)	22
47951	27.07.2023	16:24:42:318	Входящий кадр (ReadPropertyMultiple-ACK)		172.16.145.90:47808	172.16.145.81:4			
47952	27.07.2023	16:24:42:318	Interface: Получен ответ с данными на запрос (request...						
47953	27.07.2023	16:24:42:320	DEV 1 Обработка ответа на запрос чтения (requestID=...						
47954	27.07.2023	16:24:43:305	Исходящий кадр (ReadPropertyMultiple-Request)	1	172.16.145.81:47808	172.16.145.90:4			
47955	27.07.2023	16:24:43:305	Interface: Отправлен запрос на чтение (requestID=144,...						
47956	27.07.2023	16:24:43:340	Входящий кадр (ReadPropertyMultiple-ACK)		172.16.145.90:47808	172.16.145.81:4			
47957	27.07.2023	16:24:43:340	Interface: Получен ответ с данными на запрос (request...						
47958	27.07.2023	16:24:43:342	DEV 1 Обработка ответа на запрос чтения (requestID=...						
47959	27.07.2023	16:24:44:308	Исходящий кадр (ReadPropertyMultiple-Request)	1	172.16.145.81:47808	172.16.145.90:4			
47960	27.07.2023	16:24:44:308	Interface: Отправлен запрос на чтение (requestID=145,...						
47961	27.07.2023	16:24:44:331	Входящий кадр (ReadPropertyMultiple-ACK)		172.16.145.90:47808	172.16.145.81:4			
47962	27.07.2023	16:24:44:331	Interface: Получен ответ с данными на запрос (request...						
47963	27.07.2023	16:24:44:332	DEV 1 Обработка ответа на запрос чтения (requestID=...						
47964	27.07.2023	16:24:45:311	Исходящий кадр (ReadPropertyMultiple-Request)	1	172.16.145.81:47808	172.16.145.90:4			
47965	27.07.2023	16:24:45:311	Interface: Отправлен запрос на чтение (requestID=146,...						
47966	27.07.2023	16:24:45:335	Входящий кадр (ReadPropertyMultiple-ACK)		172.16.145.90:47808	172.16.145.81:4			
47967	27.07.2023	16:24:45:335	Interface: Получен ответ с данными на запрос (request...						
47968	27.07.2023	16:24:45:336	DEV 1 Обработка ответа на запрос чтения (requestID=...						
47969	27.07.2023	16:24:46:314	Исходящий кадр (ReadPropertyMultiple-Request)	1	172.16.145.81:47808	172.16.145.90:4			
47970	27.07.2023	16:24:46:314	Interface: Отправлен запрос на чтение (requestID=147,...						
47971	27.07.2023	16:24:46:337	Входящий кадр (ReadPropertyMultiple-ACK)		172.16.145.90:47808	172.16.145.81:4			

Каждая запись журнала имеет порядковый номер, дату, время и описание. Записи исходящих и входящих кадров дополнительно содержат IP-адреса отправителя и получателя.

Побайтовое представление и данные кадров модуля отображаются в соответствующих полях окна сервисного приложения Просмотрщик лога кадров.

Object Identifier	Property	Value
OBJECT_ANALOG_INPUT:0	LOW_LIMIT (59)	65
OBJECT_ANALOG_INPUT:0	HIGH_LIMIT (45)	14
OBJECT_ANALOG_INPUT:0	PRESENT_VALUE (85)	22

Данные кадра

---

Запись журнала

Входящий кадр (ReadPropertyMultiple-ACK)

	0	1	2	3	4	5	6	7	8	9
0000	81	0A	00	34	01	08	00	14	0E	01
0001	00	00	00	00	00	30	91	0E	0C	00
0002	00	00	00	1E	29	3B	4E	44	42	82
0003	00	00	4F	29	2D	4E	44	41	60	00
0004	00	4F	29	55	4E	44	41	B0	00	00
0005	4F	1F								

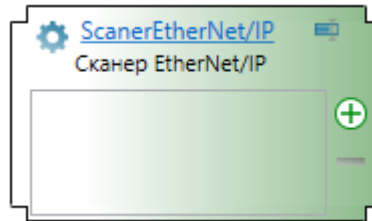
Побайтовое представление в шестнадцатиричном формате



## 1.1.2.2. EtherNet/IP

Модуль	Описание
<a href="#">Сканер EtherNet/IP</a>	Коммуникационный модуль, предназначенный для обмена данными по протоколу EtherNet/IP.

## 1.1.2.2.1. Сканер EtherNet/IP



Модуль EtherNet/IP Scanner – коммуникационный модуль, предназначенный для обмена данными между Astra.Server и ПЛК по протоколу EtherNet/IP.

### Используемые запросы EtherNet/IP (CIP)

Для обмена данными с ПЛК модуль EtherNet/IP Scanner использует следующие запросы EtherNet/IP (CIP):

- › Get\_Attribute\_Single (чтение значения атрибута объекта);
- › Set\_Attribute\_Single (запись значения атрибута объекта).

Для оптимизации обмена данными с ПЛК (скорость обмена, объём трафика) модуль EtherNet/IP Scanner может использовать запрос Multiple\_Service\_Packet (объединение нескольких запросов в один).

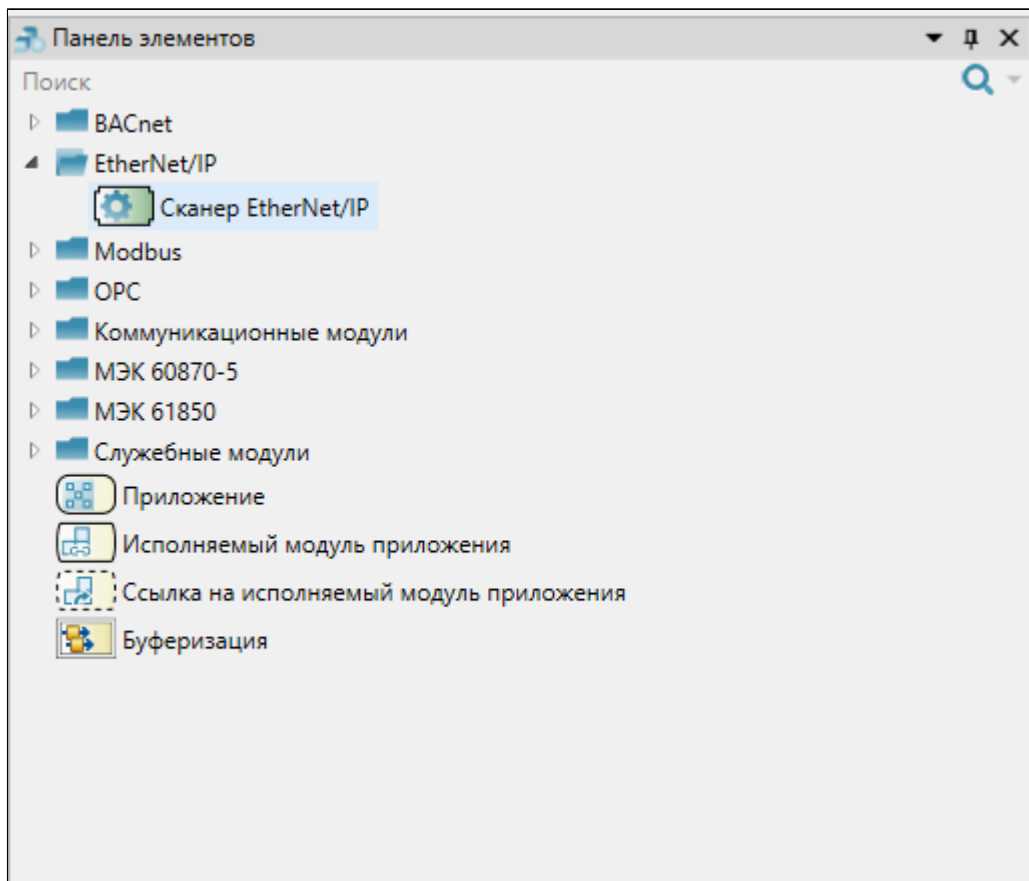
Другие запросы протокола EtherNet/IP (CIP) модуль EtherNet/IP Scanner не использует.

### Лицензирование

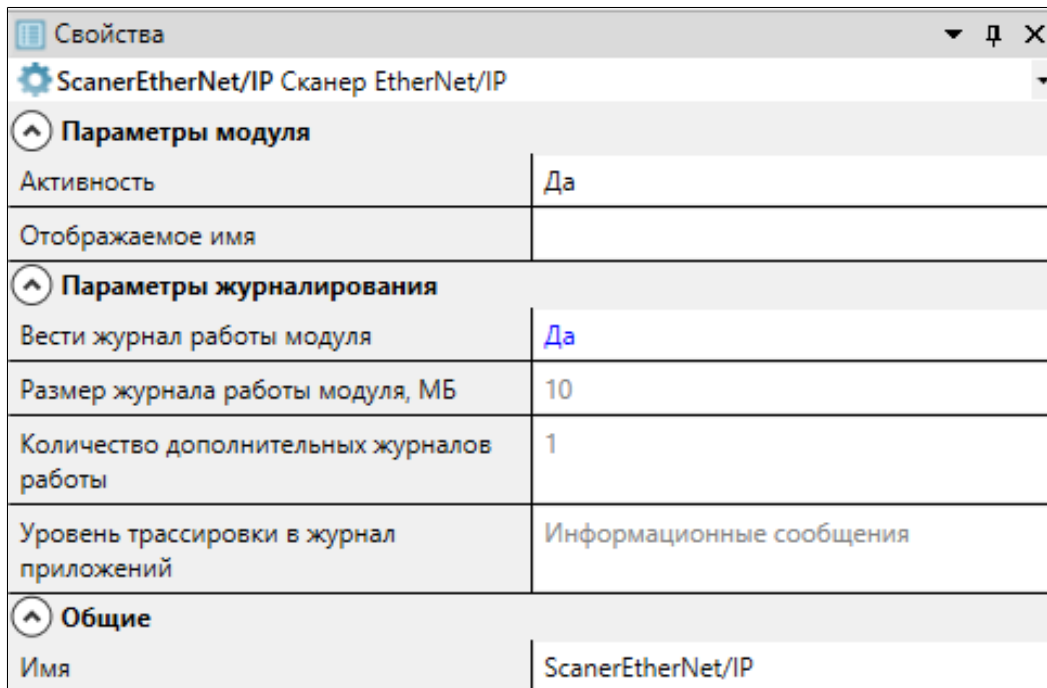
Лицензируется использование модуля EtherNet/IP Scanner и количество сигналов, обслуживаемых модулем.

## 1.1.2.2.1.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.



## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен;</li> <li>&gt; Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом «Active.Set»
Отображаемое имя	Отображаемое имя модуля (отображается в тегах служебных сигналов)

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.2.1.2. Функции модуля

Основные функции модуля EtherNet/IP Scanner:

- › сбор данных: модуль записывает в сигналы Astra.Server значения, полученные от ПЛК;
- › подача команд: модуль передаёт значения сигналов Astra.Server в ПЛК.

Специализированные функции модуля EtherNet/IP Scanner:

- › синхронизация времени ПЛК;
- › снижение нагрузки на канал;
- › подача импульсных команд;
- › обеспечение генерации события при подаче команды.

### Обмен данными с ПЛК

Модуль EtherNet/IP Scanner обменивается данными с ПЛК по сети Ethernet (протокол TCP) в режиме запрос-ответ. Инициатором запроса является модуль EtherNet/IP Scanner.

Для получения данных модуль EtherNet/IP Scanner отправляет в ПЛК запрос чтения тега или атрибута. В ответ ПЛК предоставляет значение запрашиваемого тега или атрибута. Полученное значение модуль EtherNet/IP Scanner записывает в сигнал Astra.Server.

Для подачи команды модуль EtherNet/IP Scanner отправляет в ПЛК запрос записи тега или атрибута. ПЛК выполняет запись значения в соответствующий тег или атрибут, после чего отвечает на запрос подтверждением выполнения команды.

### Обмен данными с несколькими ПЛК

Модуль EtherNet/IP Scanner может обмениваться данными одновременно и параллельно с несколькими ПЛК.

### Обмен данными при резервировании ПЛК

Модуль EtherNet/IP Scanner обеспечивает надёжный обмен данными при резервировании ПЛК. При наличии двух и более одинаковых экземпляров ПЛК модуль EtherNet/IP Scanner в процессе работы самостоятельно выбирает ПЛК для обмена данными.

## **Обмен данными при резервировании каналов связи**

Если ПЛК имеет несколько каналов связи, то модуль EtherNet/IP Scanner в процессе работы отслеживает доступность каналов связи ПЛК и самостоятельно выбирает канал для обмена данными.

## **Обмен данными по основному каналу**

При наличии основного и резервного каналов связи между Astra.Server и ПЛК обмен данными всегда ведётся по основному каналу. При обрыве связи по основному каналу EtherNet/IP Scanner начинает обмен данными по резервному каналу связи. При восстановлении связи по основному каналу EtherNet/IP Scanner возобновляет обмен данными по основному каналу.

Какой из каналов связи является основным, а какой резервным, модуль EtherNet/IP Scanner определяет по значению параметра Метрика каждого из Адаптеров Ethernet. Основным каналом считается тот канал, значение Метрики которого меньше. Если каналы имеют одинаковое значение параметра Метрика, то такие каналы считаются равнозначными и связь будет установлена по первому доступному из них.

## **Получение всех изменений в ПЛК и оптимизация трафика**

Данные в ПЛК изменяются с некоторой частотой, при этом разные данные могут изменяться с разными частотами. EtherNet/IP Scanner позволяет запрашивать данные с частотой их обновления в ПЛК.

Для получения изменений ПЛК с частотой обновления данных EtherNet/IP Scanner использует группы опроса. Для каждой частоты изменения данных настраивается отдельная группа опроса, в которой указывается период опроса данных. В карте адресов для сигналов, значения которых требуется получать с определенной частотой, указывается соответствующая группа опроса.

Опрос каждой группы ведётся не чаще, чем требуется для обнаружения изменений, что позволяет не допустить избыточного трафика.

## **Снижение нагрузки на канал связи**

Если физические каналы связи Astra.Server с ПЛК имеют разную скорость передачи данных, то при переключении на более медленный физический канал может потребоваться уменьшение объёма трафика. EtherNet/IP Scanner позволяет изменять частоту опроса группы сигналов во время работы Astra.Server.

Astra.Server не определяет переключение с одного физического канала на другой. Поэтому при переключении на более медленный физический канал частоту опроса групп сигналов задаёт пользователь с помощью служебных сигналов.

## **Оптимизация скорости обмена и объёма трафика**

ПЛК может предоставлять данные в виде массивов. EtherNet/IP Scanner позволяет запрашивать значения требуемых элементов массива за один запрос, вместо нескольких запросов к отдельным элементам массива.

## **Подача команд импульсного типа**

EtherNet/IP Scanner позволяет подавать в ПЛК команды импульсного типа. При подаче команды импульсного типа сначала требуемый бит устанавливается или сбрасывается, а спустя заданное время значение бита меняется на противоположное, т.е. сбрасывается или устанавливается.

## **Генерация события при подаче команды импульсного типа**

EtherNet/IP Scanner позволяет обеспечивать генерацию событий при каждой подаче команды импульсного типа в ПЛК, даже если команда подана повторно. Для этого EtherNet/IP Scanner после подачи команды импульсного типа сбрасывает значение сигнала в исходное.

## **Информирование о результате подачи команды в ПЛК**

EtherNet/IP Scanner позволяет получать информацию о результате подачи команды в ПЛК. При подаче команды в определенный сигнал записывается значение, которое определяет результат подачи команды - помещена в очередь, исполнена или не исполнена.

## **Синхронизация времени ПЛК**

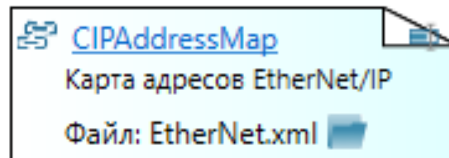
EtherNet/IP Scanner позволяет синхронизировать время ПЛК со временем Astra.Server. Синхронизация выполняется путём записи текущего времени Astra.Server в часовом поясе UTC+0 в некоторый тег ПЛК



## 1.1.2.2.1.3. Карта адресов

Для настройки сигналов модуля сканер EtherNet/IP используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов EtherNet/IP в исполняемое приложение;



2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	Сервис	Класс	Экземпляр	Атрибут	Смещение	Тег	Протокольный тип	Маска	Бит	Длительность	Категория дан
uint4	uint4	непосредственн	Тег					Tag1	UDINT				
int4	int4	непосредственн	Атрибут	245	1	5	8		DINT				

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>➤ Непосредственно;</li><li>➤ Только чтение;</li><li>➤ Не привязан.</li></ul>
Сервис	Требуемый режим: <ul style="list-style-type: none"><li>➤ Тег;</li><li>➤ Изменение тега;</li><li>➤ Атрибут.</li></ul>
Класс	Идентификатор класса объекта
Экземпляр	Идентификатор экземпляра объекта
Атрибут	Идентификатор (номер) атрибута
Смещение	Смещение в байтах относительно начала данных при чтении структур данных

Тег	Тег в ПЛК, бит которого требуется изменять
Протокольный тип	Тип тега в ПЛК
Маска	<p>Требуемый режим записи:</p> <ul style="list-style-type: none"> <li>➤ "PULSE_0" - если требуется сбрасывать бит ("0"), а затем устанавливать ("1") по истечении интервала времени, указанного в параметре Длительность;</li> <li>➤ "PULSE_1" - если требуется устанавливать бит ("1"), а затем сбрасывать ("0") по истечении интервала времени, указанного в параметре Длительность;</li> <li>➤ "BIT" - запись отдельного бита;</li> <li>➤ "AND" - запись результата побитового И между текущим значением тега ПЛК и значением сигнала Astra.Server;</li> <li>➤ "OR" - запись результата побитового ИЛИ между текущим значением тега ПЛК и значением сигнала Astra.Server.</li> </ul>
Бит	Номер бита, значение которого требуется изменять
Длительность	Интервал времени в миллисекундах, через который устанавливается исходное значение
Категория данных	Категории данных нужны для группировки сигналов по интервалу опроса. Категории могут быть удобны, когда требуется оптимизировать нагрузку на сеть или сократить объем трафика.

При пересечении или совпадении адресов (например, при создании управляющего воздействия на один и тот же адрес) происходит запись в журнал модуля о наличии пересечений адресов, но к отказу в постановке сигналов на обслуживание не приводит. Проверка на совпадение и пересечение адресов происходит при старте модуля.

## 1.1.2.2.1.4. Сигналы доставки

Значение сигнала доставки определяет состояние отправленной команды. Возможные значения сигнала доставки приведены в таблице.

<b>Значение</b>	<b>Состояние команды управления</b>
«1»	Команда успешно помещена в очередь на отправку.
«3»	Команда исполнена.
«-1»	Команда не исполнена.


# 1.1.2.2.1.5. Диагностика работы модуля

## Журнал работы


Модуль EtherNet/IP Scanner ведёт журнал работы, в который записывается информация о работе модуля и обмене данными с ПЛК.

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:

 C:\Program Files\AstraRegul\Astra.Server\Logs;

» в Linux системах в директории:

 /opt/AstraRegul/Astra.Server/Logs.

Для просмотра журнала работы модуля используется сервисное приложение Просмотрщик лога кадров.

№	Дата	Время	Источник	Устройство	Канал	Описание	Тип запроса	Путь	Тип	Статус	Значение
8079	05.07.2023	10:31:38:737	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега	Чтение тега	DINTArray3D[0][1][3]	DINT	Success	55
8080	05.07.2023	10:31:38:847	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8081	05.07.2023	10:31:38:848	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега					
8082	05.07.2023	10:31:38:999	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8083	05.07.2023	10:31:38:999	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега					
8084	05.07.2023	10:31:39:150	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8085	05.07.2023	10:31:39:150	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега					
8086	05.07.2023	10:31:39:302	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8087	05.07.2023	10:31:39:302	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега					
8088	05.07.2023	10:31:39:454	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8089	05.07.2023	10:31:39:454	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега					
8090	05.07.2023	10:31:39:607	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8091	05.07.2023	10:31:39:607	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега					
8092	05.07.2023	10:31:39:758	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8093	05.07.2023	10:31:39:758	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега					
8094	05.07.2023	10:31:39:869	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8095	05.07.2023	10:31:39:869	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега					
8096	05.07.2023	10:31:40:021	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8097	05.07.2023	10:31:40:021	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега					
8098	05.07.2023	10:31:40:173	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8099	05.07.2023	10:31:40:173	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега					
8100	05.07.2023	10:31:40:325	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8101	05.07.2023	10:31:40:325	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение тега					
8102	05.07.2023	10:31:40:477	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение тега					
8103	05.07.2023	10:31:40:843	PLC	Crate1_CPU	Канал 1	Опрос группы 3 sec.					
8104	05.07.2023	10:31:40:843	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение атрибу					
8105	05.07.2023	10:31:40:995	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение атрибу					
8106	05.07.2023	10:31:40:995	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение атрибу					
8107	05.07.2023	10:31:41:146	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение атрибу					
8108	05.07.2023	10:31:41:146	PLC	Crate1_CPU	Канал 1	Исходящий кадр: Чтение атрибу					
8109	05.07.2023	10:31:41:296	PLC	Crate1_CPU	Канал 1	Входящий кадр: Чтение атрибу					

0	1	2	3	4	5	6	7	8	9	
0000	EF	00	1A	00	01	00	40	00	00	00
0001	00	00	00	00	00	00	00	00	00	00
0002	00	00	00	00	00	00	00	00	00	00
0003	02	00	00	00	00	00	B2	00	0A	00
0004	CC	00	00	00	C4	00	37	00	00	00

Каждая запись журнала имеет порядковый номер, дату, время, источник, устройство, канал и описание. Побайтовое представление и данные кадров модуля отображаются в соответствующих полях окна сервисного приложения Просмотрщик лога кадров.

The screenshot displays three sections of a log viewer interface:

- Данные кадра (Frame Data):** A table with columns: Тип запроса (Request Type), Путь (Path), Тип (Type), Статус (Status), and Значение (Value). The row shows: Чтение тега (Tag Reading), DIntArray3D[0][1][3], DINT, Success, and 55.
- Запись журнала (Journal Entry):** A text field containing "Входящий кадр: Чтение тега" (Incoming frame: Tag Reading).
- Побайтовое представление в шестнадцатиричном формате (Hexadecimal Representation):** A table with columns 0-9 and rows 0000-0004. The row 0004 is highlighted in blue.

Тип запроса	Путь	Тип	Статус	Значение
Чтение тега	DIntArray3D[0][1][3]	DINT	Success	55

Входящий кадр: Чтение тега

	0	1	2	3	4	5	6	7	8	9
0000	6F	00	1A	00	01	00	40	00	00	00
0001	00	00	00	00	00	00	00	00	00	00
0002	00	00	00	00	00	00	00	00	00	00
0003	02	00	00	00	00	00	B2	00	0A	00
0004	CC	00	00	00	C4	00	37	00	00	00

## 1.1.2.3. Modbus

Модуль	Описание
<a href="#">Опросчик Modbus TCP</a>	Коммуникационный модуль для опроса данных по протоколу Modbus TCP
<a href="#">Опросчик Modbus RTU</a>	Коммуникационный модуль для опроса данных по протоколу Modbus RTU
<a href="#">Станция Modbus TCP</a>	Коммуникационный модуль для передачи данных по протоколу Modbus TCP
<a href="#">Станция Modbus RTU</a>	Коммуникационный модуль для передачи данных по протоколу Modbus RTU

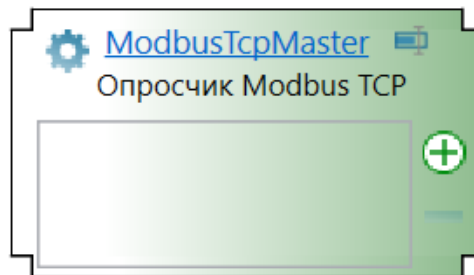
### Адресное пространство

Адресное пространство протокола Modbus представляет собой 4 сегмента памяти.

Сегмент	Описание	Размер	Чтение/Запись
Discrete Inputs	дискретные входы	одиночный бит	чтение
Coils	ячейки	одиночный бит	чтение и запись
Input Registers	входные регистры	16-битное слово	чтение
Holding Registers	регистры хранения	16-битное слово	чтение и запись

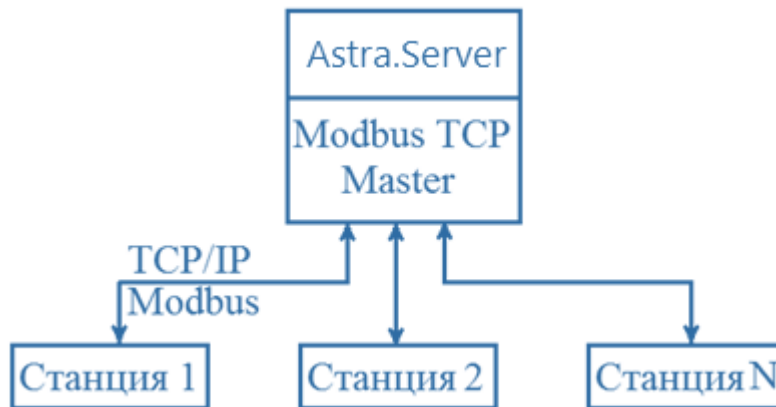
Доступ к элементам каждого сегмента осуществляется с помощью 16-битной адресации. При такой адресации, каждый из четырех сегментов может вмещать до 65536 элементов (адресов).

## 1.1.2.3.1. Опросчик Modbus TCP



Модуль Modbus TCP Master работает в составе Astra.Server, который является частью системы для сбора, анализа и регулирования параметров технологического процесса.

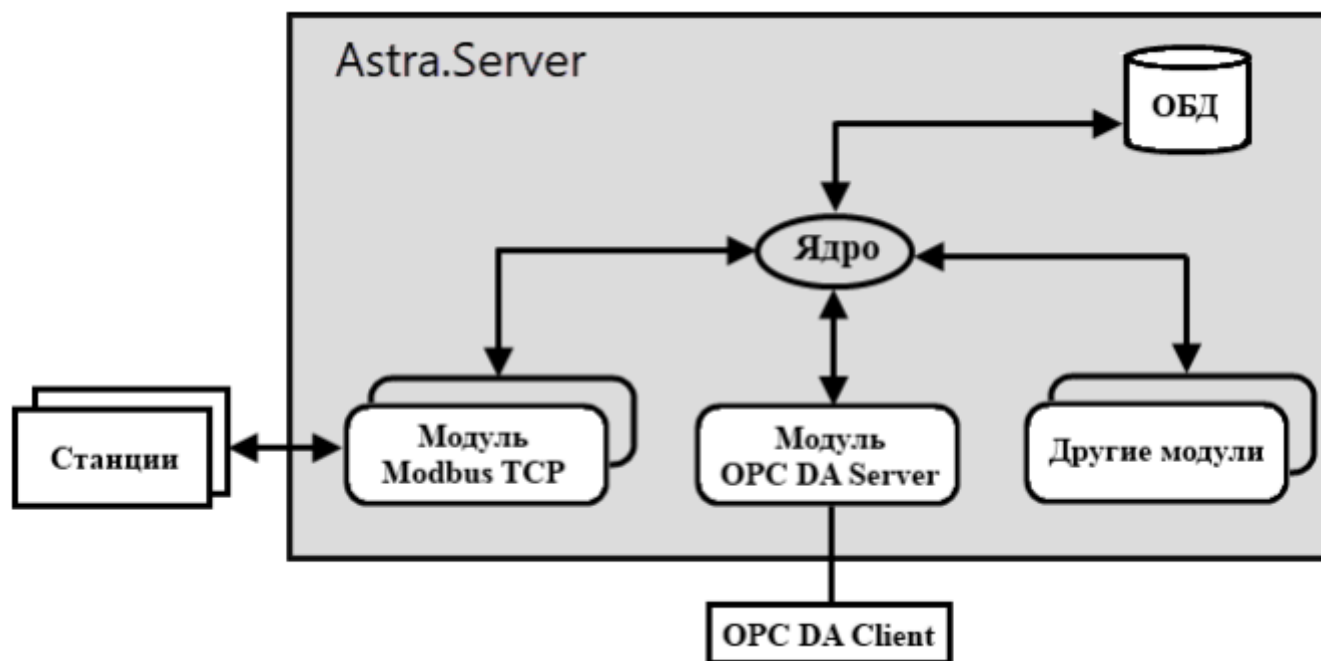
Модуль Modbus TCP Master предназначен для опроса подчиненных станций в сетях TCP/IP по протоколу Modbus в соответствии со спецификацией [Modbus Application Protocol Specification](#). Astra.Server поддерживает до 64-х модулей Modbus TCP Master.



Основными функциями модуля Modbus TCP Master являются сбор данных и подача управляющих воздействий подчиненным станциям.

Работая в составе Astra.Server, модуль Modbus TCP Master опрашивает подчиненные станции, циклически отправляя им запросы. Полученные данные через интерфейсы ядра сохраняются в оперативную базу данных (ОБД) (рисунок ниже). Протокол Modbus не поддерживает передачу качества и метки времени, но при этом модуль сам выставляет эти свойства для сигналов при записи в ядро на основании алгоритмов, описанных ниже.

Ядро Astra.Server при получении изменившихся сигналов уведомляет об изменениях сигналов другие модули, которые обслуживают эти сигналы, в том числе модуль OPC DA Server. Далее модуль OPC DA Server передает данные клиентам, подписанным на изменившиеся сигналы.



Модуль Modbus TCP Master отправляет запросы подчиненным станциям, т.е. выполняет роль TCP клиента, а станция поставляет данные в ответ на запрос, т.е. выполняет роль TCP сервера. Обмен данными всегда инициируется модулем Modbus TCP Master.

При старте модуля происходит инициализация модуля. Процесс инициализации включает в себя:

- › чтение конфигурационных данных модуля;
- › подписка у ядра на получение уведомлений об изменениях сигналов;
- › проверка корректности настройки адресов для сигналов уведомлений и постановка их на обслуживание;
- › формирование таблицы поллинга.

Затем в соответствии с режимом работы модуль устанавливает соединение и начинает работу в соответствии со своими настройками.



При установленном соединении модуль начинает опрос станций по заданным адресам.

Адресное пространство протокола Modbus представляет собой четыре массива данных:

- Coils - ячейки. Одиночный бит. Тип доступа на чтение и запись;
- Discrete Inputs - дискретные входы. Одиночный бит. Тип доступа только на чтение;
- Holding Registers - регистры хранения. 16-битное слово. Тип доступа на чтение и запись;
- Input Registers - входные регистры. 16-битное слово. Тип доступа только на чтение.

Опрос каждой подчиненной станции ведется по таблице поллинга, сформированной при старте модуля. Формирование таблицы поллинга ведется по использованным при конфигурировании адресам в массивах данных.

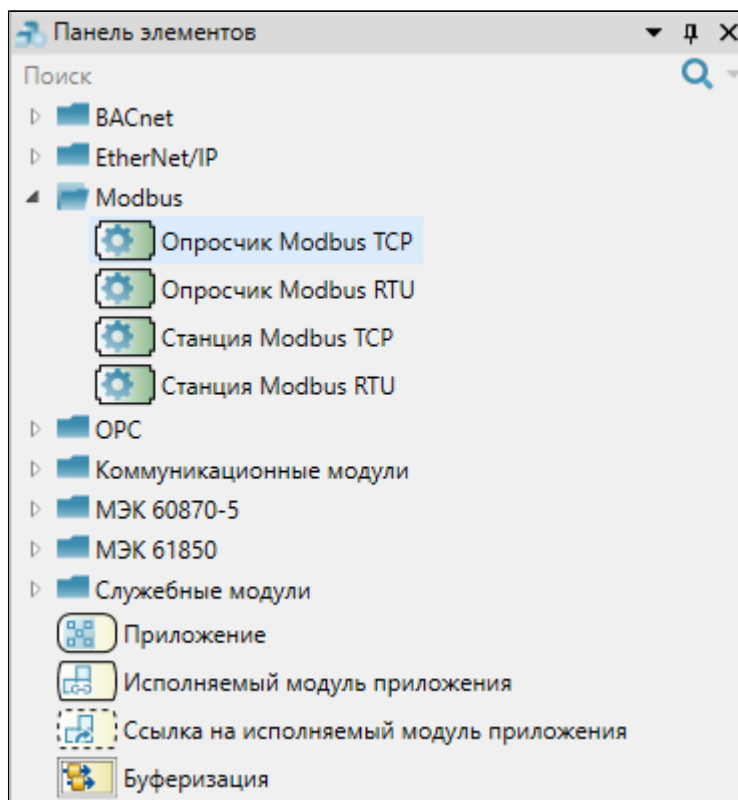
В модуле реализовано расширение стандартных функций протокола Modbus:

- для передачи времени возникновения данных.
- для увеличения производительности при работе с оборудованием от ЭМИКОН.

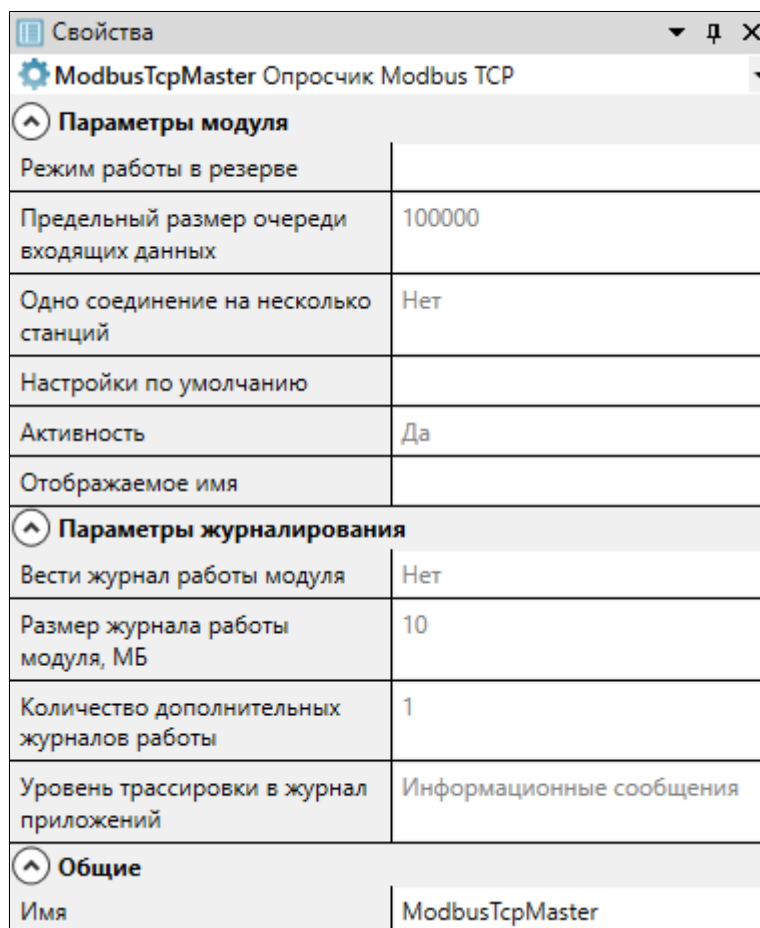
Опрос станций модулем Modbus TCP Master может производиться по резервируемым каналам связи.

## 1.1.2.3.1.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.



## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Режим работы в резерве	<p><b>Работа модуля в резерве:</b></p> <ul style="list-style-type: none"> <li>&gt; Вести опрос – модуль ведет опрос и выставляет сигналам полученные значения;</li> <li>&gt; Поддерживать соединения - модуль устанавливает и поддерживает связь с подчиненной станцией. Опрос и отправка команд управления не ведутся;</li> </ul>

	<p>› Закрывать соединения - модуль не устанавливает соединение с подчиненной станцией, пока не перейдет в режим РАБОТА</p> <p>Значение по умолчанию: Поддерживать соединения</p>
Предельный размер очереди входящих данных	Размер очереди значений для записи в Astra.Server, при достижении которого приостанавливается опрос станций. Значение по умолчанию 100000 значений
Одно соединение на несколько станций	Использование одного TCP соединения для станций, расположенных по одному адресу. Значение по умолчанию: Нет
Настройки по умолчанию	Выбор файла с готовой конфигурацией
Активность	<p>Активность модуля при запуске/перезапуске Astra.Server:</p> <ul style="list-style-type: none"> <li>› Да – модуль запущен;</li> <li>› Нет – модуль остановлен.</li> </ul> <p>Управляется служебным сигналом Active.Set</p>
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.3.1.2. Функциональные возможности

### Режимы работы модуля

Модуль Modbus TCP Master может работать в двух режимах: РАБОТА или РЕЗЕРВ. Режим работы модуля в Astra.Server устанавливает ядро. Режим работы модуля напрямую зависит от режима работы Astra.Server. В процессе работы режим работы Astra.Server может измениться.

Резервирование предназначено для повышения надежности системы управления. Резервирование Astra.Server необходимо для минимизации времени отсутствия основной функции систем управления, а именно времени потери функции управления станциями. Повышение надежности происходит благодаря тому, что при выходе из строя Astra.Server, содержащего модуль, происходит быстрое переключение на другой Astra.Server.

Время перехода модуля между режимами максимально оптимизировано. Время перехода увеличивается при использовании функций ограничения работы в режиме РЕЗЕРВ.

В режиме РАБОТА модуль выполняет функции:

- › ведет опрос подчиненной станции;
- › отправляет команды управления подчиненным станциям.

Отличия режима РЕЗЕРВ от режима РАБОТА:

- › управляющие воздействия подчиненным станциям не отправляются;
- › есть возможность регулирования ограничений нагрузки работы на подчиненную станцию. Нагрузка на подчиненную станцию увеличивается при множественных подключениях, а также при выполнении функций опроса и управления по множественным подключениям.

Ограничение режима работы в режиме РЕЗЕРВ предназначено для того, чтобы обеспечить корректную работу с устройствами, которые не поддерживают множественных подключений или не обеспечивают необходимой скорости работы с транзакциями в режиме постоянного опроса несколькими мастерами.

Скорость переключения между модулями зависит от настроек работы модуля в режиме РЕЗЕРВ. Имеются три варианта работы модуля в режиме РЕЗЕРВ:

- "вести опрос" - в данном режиме модуль устанавливает связь с подчиненной станцией и отправляет запросы на чтение данных с подчиненной станции. При переключении между режимами не теряется время на установку соединения со станцией, а также на полный опрос этой станции. Рекомендованный вариант работы резервного модуля;
- "поддерживать соединение" – в данном режиме модуль устанавливает связь с подчиненной станцией и проводит проверку соединения. При переключении между режимами не теряется время на установку соединения с подчиненной станцией;
- "закрыть соединение" - в данном режиме модуль не устанавливает соединение и не проверяет состояние связи с подчиненной станцией, пока не перейдет в режим РАБОТА. Такой вариант работы используется, если подчиненная станция не поддерживает множественных подключений.

## Стратегия формирования запросов на чтение

Модуль Modbus TCP Master предоставляет выбор варианта составления таблицы поллинга, т.е. возможность регулировки максимальной длины запроса на чтение, и выбор варианта учитывать или не учитывать разрывы данных в адресном пространстве протокола.

Существуют два варианта формирования таблицы поллинга. Один из них позволяет оптимизировать количество запросов, другой объем запрашиваемых данных. Выбор варианта формирования таблицы поллинга производится при конфигурировании модуля:

- "По максимуму" - строится карта запросов максимальной длины, не учитывая фрагментацию запрашиваемых данных – объединение в один запрос расположенных не подряд данных. Например, если есть 2 сигнала с адресом 1 и 125, они объединяются в один запрос, лишние данные при получении игнорируются (т.е. игнорируются данные с адресами от 2 до 124). Таким образом, оптимизируется количество отправляемых запросов за счет увеличения трафика (количества переданной информации);

➤ "На сплошные данные" - строится карта запросов только по необходимым данным. Таким образом, оптимизируется трафик за счет потери времени. Потеря времени происходит из-за необходимости построения и отправки нескольких запросов.

Максимальная длина запроса равна 125 регистров. Длина запроса задается в параметрах конфигурации модуля.



#### Формирование карты поллинга

Если заданы адреса на чтение 1, 7-25, 47, 130 то для стратегии формирования запросов:

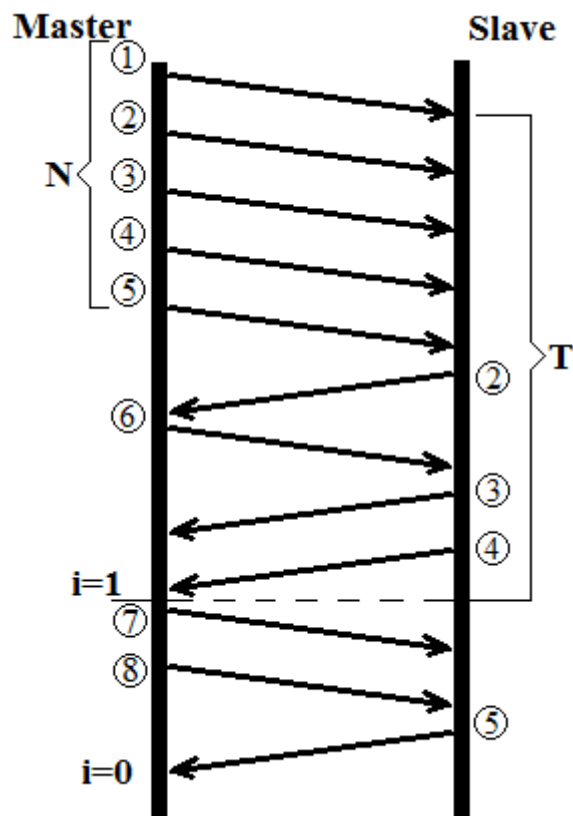
- по максимуму будут сформированы два запроса. Один из запросов по адресам от 1 до 125, второй запрос от 126 до 130;
- на сплошные данные будет сформировано 3 запроса: по адресу 1, по адресам от 7 до 25, по адресу 47 и по адресу 130.

## Опрос подчиненной станции

Опрос нескольких подчиненных станций ведется модулем параллельно, т.е. опрос одной станции не зависит от опроса другой станции.

При опросе подчиненной станции модуль отправляет запросы на чтение, сформированные в таблице поллинга. Количество одновременно активных запросов настраивается в параметрах модуля.

В начале опроса модуль в соответствии с настройками параметра "Максимальное количество одновременных запросов" отправляет станции N первых запросов по таблице поллинга. По мере поступления ответов модуль отправляет новые запросы, поддерживая количество одновременно доступных запросов не более, чем N.



На рисунке обозначено:

- > "T" - время ожидания ответа от станции (параметр конфигурации) промежуток времени, в течение которого модуль ждет ответа на запрос от подчиненной станции;
- > "i" - счетчик количества неуспешных запросов. Используется для определения состояния связи со станцией. С момента запуска модуля имеет значение равное 0 ( $i=0$ );
- > "N" - максимальное количество одновременных запросов (параметр конфигурации).

По истечении указанного времени T счетчик количества неуспешных запросов увеличивается на 1 ( $i=i+1$ ). Время отчитывается с момента отправки запроса на станцию. Если количество неуспешных запросов больше максимального количества неуспешных запросов (параметр конфигурации), то связь со станцией считается потерянной. Счетчик сбрасывается в 0 при нормальном завершении хотя бы одной операции запроса ( $i=0$ ).

Модуль имеет четыре состояния связи со станцией: соединение отсутствует, ожидание подключения, соединение есть, запрашиваются или передаются



данные. Отображение состояния связи ведется в служебном сигнале связи. Значениям служебного сигнала связи соответствуют следующие состояния связи:

- соединение со станцией отсутствует, значение сигнала связи равно 0. Состояние устанавливается при запуске модуля и при разрыве связи со станцией;
- ожидание подключения, значение сигнала связи равно 1. Это значение служебного сигнала связи принимается при попытках установить связь и при кратковременной потере связи. Если связь установлена, то выполняется переход в следующее состояние со значением сигнала связи 2, иначе выполняется переход в состояние со значением сигнала связи 0;
- соединение со станцией установлено, значение сигнала связи равно 2. Значение устанавливается при появлении соединения со станцией;
- запрашиваются или передаются данные, значение сигнала связи равно 3. Значение принимается при опросе станции или при отправке команд управления подчиненной станции. При кратковременной потере связи сигнал состояния связи принимает значение 1.

Если полученные данные с подчиненной станции не успевают обрабатываться и очередь данных достигает максимального размера, то опрос станции приостанавливается, происходит запись в журнал работы модуля об этом событии. Опрос подчиненной станции продолжается, когда уменьшится размер очереди данных.

Функция изменения порядка байт необходима для работы с определенными устройствами, которые на аппаратном уровне меняют местами байты в словах.

При опросе и выдаче управляющих воздействий модуль имеет возможность изменять порядок байт, параметры настраиваются в конфигурации модуля:

- Байт в слове - включает/отключает изменение порядка следования байтов в регистре. В случае если флаг установлен, то байты регистра меняются местами.



Действие флага распространяется на все протокольные типы сигналов, кроме TS и TC.

›Слов в сигналах TMC, TR4 — включает/отключает изменение порядка следования слов в сигналах TMC, TR4. В случае если флаг установлен, то слова данных меняются местами.



Если одновременно установлены флаги Слов в сигналах TMC, TR4 и Байт в слове, то слова данных меняются местами и байты в словах меняются местами;



›Слов в сигналах TMF4, TMF8, TRF4 — включает/выключает изменение порядка слов в сигналах TMF4, TMF8, TRF4. В случае если флаг установлен, изменение порядка следования слов происходит аналогично изменению порядка следования слов в сигналах TMC, TR4. Если одновременно установлены флаги Слов в сигналах TMF4, TMF8, TRF4 и Байт в слове, то пары байтов меняются местами и в каждой паре байты меняются местами (рисунок выше).

## Качество сигналов

В протоколе Modbus отсутствует понятие качества данных, и поэтому качество устанавливается модулем, исходя из результатов выполнения запросов. При старте модуля все сигналы имеют плохое качество.

Качества сигналов, выставляемых коммуникационным модулем, приведены в таблице:

Значение качества	Идентификатор качества	Расшифровка
-------------------	------------------------	-------------

4	CONFIG_ERROR	Сигнал неправильно сконфигурирован
8	NOT_CONNECTED	Сигнал принят на обслуживание, но устройство еще не инициализировано
12	DEVICE_FAILURE	Ошибка инициализации устройства
20	LAST_KNOWN	КП нет на связи, последнее пришедшее значение
24	COMM_FAILURE	КП нет на связи
28	OUT_OF_SERVICE	Модуль не запущен
64	UNCERTAIN	Связь установлена, но значения еще не пришли
192	GOOD	Значение сигнала достоверно
216	LOCAL_OVERRIDE	Значение достоверно, введено вручную

При переходе между резервными каналами и кратковременными пропаданиями связи качество сигналов не изменяется, так как предусмотрен параметр Таймаут потери связи (параметр конфигурации, рисунок ниже). Если связь со станцией потеряна, то начинается отчет времени, в течение которого модуль пытается восстановить соединение с подчиненной станцией – таймаут потери связи  $T_0$ . По истечению этого времени если соединение не удалось установить, то сигналам выставляется качество COMM\_FAILURE.



Модуль имеет возможность обрабатывать ошибки, для того чтобы не выставлять плохое качество данных при возникновении единичных сбоев в поведении устройства, а также чтобы сохранить целостную картину при резервных переходах устройств. Алгоритмы обработки ошибок следующие:

- если устройство при запросе данных возвращает строку SLAVE DEVICE BUSY (запись строки происходит в поле Качество сигнала), то запрос данных не повторяется, связь со станцией не меняется, модуль переходит к запросу следующих данных. Недоставленные данные перезапрашиваются в следующем цикле опроса;
- если устройство при запросе данных возвращает строку ILLEGAL DATA ADDRESS (запись строки происходит в поле Качество сигнала), то по данному запросу инкрементируется счетчик ошибочных запросов. Если в течение трех циклов опроса устройства подряд возвращается ошибка ILLEGAL DATA ADDRESS, то данным запроса выставляется качество OPC\_QUALITY\_BAD. Любой корректный ответ по этим данным обнуляет счетчик. Счетчик взводится отдельно для каждого запроса в карте поллинга;
- если устройство при запросе данных возвращает любой другой код ошибки, то по данному запросу инкрементируется счетчик ошибочных запросов. Если в течение трех циклов опроса устройства подряд возвращается данная ошибка, то данным запроса выставляется качество OPC\_QUALITY\_BAD и инкрементируется счетчик неуспешных запросов. При достижении максимального количества неуспешных запросов связь со станцией считается потерянной. Любой корректный ответ по этим данным обнуляет счетчик.

## Метка времени

Протокол Modbus не оперирует понятием время возникновения данных, обычно сигналам присваивается время сервера на момент получения данных, свойство 4 (TimeStamp).

Модуль Modbus TCP Master реализует расширение спецификации в части передачи метки времени. Эта возможность может быть использована как для формирования в контроллере времени изменения значения, так и для формирования однозначной последовательности событий, возникающих на

уровне контроллера, что невозможно в обычных условиях, так как опросчик может не успеть прочитать данные, если они очень быстро изменяются.

Для передачи метки времени параметров вводятся дополнительные типы данных в модуле Modbus TCP Master: TM2\_TIME, TMF4\_TIME, TMC\_TIME, TMF8\_TIME. Протокольные типы данных с меткой времени представляются логической структурой, представленной в таблице:

Рег./Поз.	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
1-4	Данные занимают от 1 до 4 регистров в зависимости от типа сигнала															
5	секунды						миллисекунды									
6	час								минуты							
7	год						месяц				день					

Для возможности взаимодействия модуля и контроллера на стороне контроллера следует поддерживать формирование описанной структуры.

Модуль, работая с сигналом, который имеет тип с меткой времени, формирует запрос на чтение с учетом типа сигнала и дополнительной структуры длиной три регистра. При получении данных метка времени, лежащая в дополнительной структуре, преобразуется к стандартному времени и записана в свойство 4 (TimeStamp) одновременно с записью значения в свойство 2 (Value).

Для получения метки времени, у подчиненной станции должен быть поддержан формат структур (таблица выше) по сигналам, которые должны содержать метку времени. Поля структуры должны заполняться на стороне подчиненной станции. В Astra.Server должны быть указаны протокольные типы данных с меткой времени для сигналов.

## Отправка управляющих и регулирующих воздействий

Отправка команд телеуправления и телерегулирования выполняется, только когда пара резервируемых серверов активна и Astra.Server находится в режиме РАБОТА.

Отправка управляющих воздействий имеет больший приоритет, чем опрос станций. При необходимости отправки управляющего воздействия модуль вставляет пакет, содержащий команду в начало очереди запросов на отправку подчиненной станции.

Отправка команд по резервным каналам происходит также как при опросе станций.

Модуль позволяет указать количество повторов отправки команд в случае неуспешной отправки. Повторы отправки команд происходят, если:

- устройство вернуло в ответ исключение с кодом SLAVE DEVICE BUSY. Если в итоге запись так и не была выполнена, состояние связи не меняется;
- устройство вернуло в ответ исключение с кодом ILLEGAL DATA ADDRESS, ILLEGAL DATA VALUE, состояние связи не меняется.

Если устройство вернуло в ответ исключение с любым кодом, кроме кодов, перечисленных выше, то отправка команд не повторяется, счетчик количества неуспешных запросов увеличивается на 1.

Если вышел таймаут ожидания ответа от станции, то отправка команд не повторяется, увеличивается счетчик количества неуспешных запросов.

Качество данных является показателем достоверности информации. Управляющие сигналы с недостоверной информацией отправлять недопустимо. Сигналы плохого качества не отправляются.

Для исходящих сигналов модуль позволяет настраивать сигналы доставки управляющих и регулирующих воздействий до подчинённой станции, которые формируются по правилам из таблицы:

Значение	Описание
2	Доставка подтверждена

1	Ожидание подтверждения доставки
-1	Нет связи со станцией
-2	Вышел таймаут подтверждения запроса
-1000 + код исключения	Устройство вернуло код исключения

Модуль имеет возможность отправлять все сигналы телерегулирования, при проявлении подчиненной станции на связи. При этом может произойти отправка сигнала, содержащего устаревшую информацию, из-за этого функцию необходимо использовать осторожно.

Отправка сигнала телеуправления (ТС) может проводиться в сегменте Holding Registers адресного пространства. Для возможности подачи телеуправления при конфигурировании сигнала следует указать номер бита, который будет устанавливаться в 1 при отправке телеуправления. При подаче телеуправления модуль формирует регистр для записи в устройство таким образом, что все биты, кроме указанного в настройках адреса сигнала телеуправления, будут сброшены в ноль. Например, если в адресе сигнала записано "BitPosition=(5)", то при отправке ТС в Holding Registers запишется число "00000000 00010000" или "32" в зависимости от настроек подчиненной станции.

## Резервируемые каналы связи

Резервирование каналов связи с устройством является одним из способов повышения надежности системы автоматизации. Разные производители устройств строят системы работы с резервируемыми каналами по-разному. В связи с этим модуль реализует несколько вариантов работы с подчиненными станциями по резервируемым каналам. Максимальное количество каналов связи Master-Slave равно 4. Количество настраивается в параметрах конфигурации.

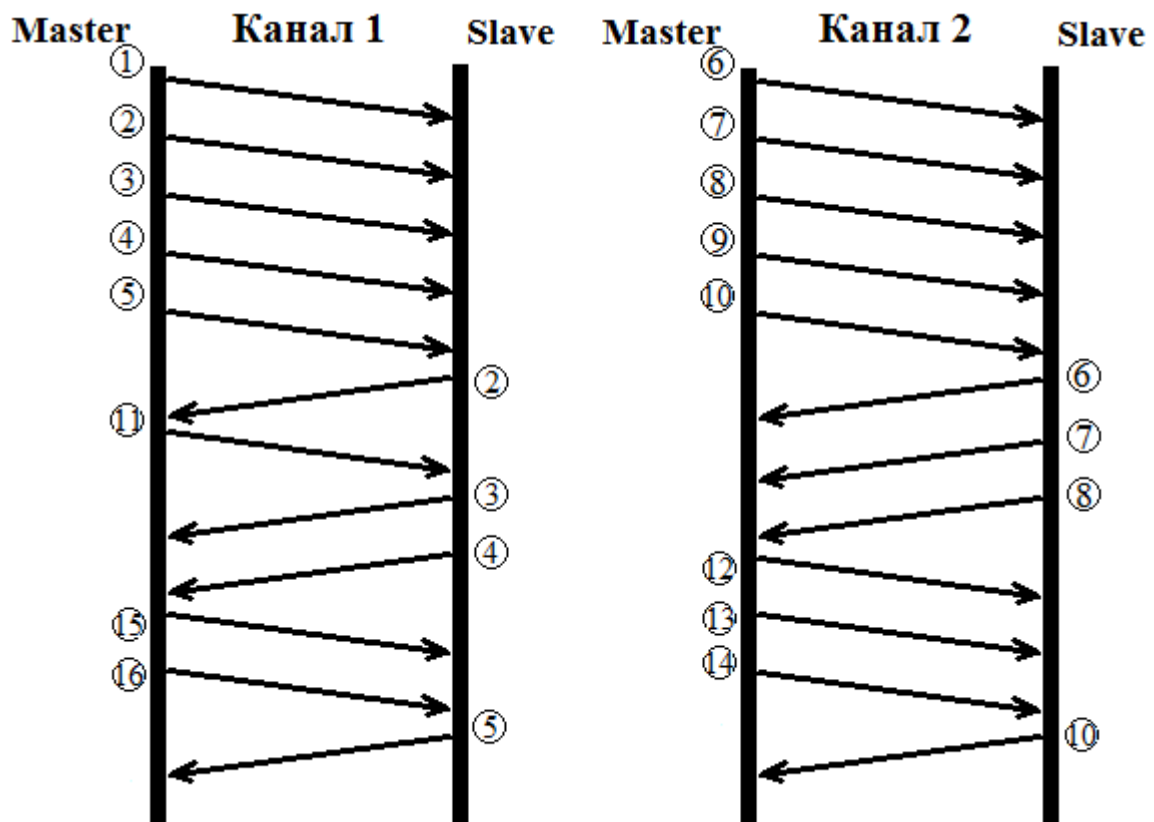
Имеются три режима работы резервных каналов, предназначенных для того, чтобы:

- максимально сократить время задержки доставки данных при переходе между резервными каналами связи;
- сократить время отсутствия возможности передачи управляющих воздействий;
- увеличить скорость опроса. Увеличение скорости опроса происходит благодаря тому, что запросы отправляются по каналам параллельно и за одно время работы пройдет в N раз больше запросов, где N – количество каналов связи.

Режимы работы резервных каналов следующие:

- все каналы выполняют опрос станции. Для каждого канала, по которому ведется опрос устройства, модуль резервирует максимальное количество одновременно возможных транзакций. В данном режиме работы модуль считает все резервные каналы единым каналом с большой пропускной способностью. Количество свободных транзакций суммируется (рисунок ниже). При отказе работы одного из каналов весь трафик делится между оставшимися каналами;
- один канал выполняет опрос, по остальным каналам поддерживается связь со станцией. Этот вариант используется для того, чтобы не перегружать устройство. В тоже время при переключении между каналами время переключения минимально. При отказе работы канала, выполняющего опрос, функция опроса передается другому каналу, выбор которого осуществляет модуль;
- один канал выполняет опрос, связь со станцией по остальным каналам не поддерживается. Этот вариант используется в случае, когда устройство не поддерживает множественные подключения опросчиков, при необходимости переключения между каналами время перехода будет больше чем в предыдущих случаях.





## 1.1.2.3.1.3. Типы данных

Модуль Modbus TCP Master, оперируя данными в соответствии со спецификацией, может предоставлять их в виде следующих типов. Такие типы данных называются протокольными.

Протокольный тип	Тип значения	Направление передачи	Описание
TS	bool	Slave → Master	Телесигнализация
TC	bool	Master → Slave	Телеуправление
TM2	int2/uint2	Slave → Master	Телеизмерение
TMF4	float	Slave → Master	Телеизмерение
TMC	int4/uint4	Slave → Master	Телесчет
TMF8	double	Slave → Master	Телеизмерение
TM2_TIME	int2/uint2	Slave → Master	Телеизмерение с меткой времени
TMF4_TIME	float	Slave → Master	Телеизмерение с меткой времени
TMC_TIME	int4/uint4	Slave → Master	Телесчет с меткой времени
TMF8_TIME	double	Slave → Master	Телеизмерение с меткой времени
TR2	int2/uint2	Master → Slave	Телерегулирование
TRF4	float	Master → Slave	Телерегулирование
TR4	int4/uint4	Master → Slave	Телерегулирование
STR	string	Slave → Master	Телеизмерение
STR-COMMAND	string	Master → Slave	Телерегулирование

Как видно из таблицы сигналы по отношению к модулю делятся по направлению передачи на две группы:

- входящий сигнал, в который будет записываться полученное значение с подчиненной станции. Направление передачи для сигнала от Slave к Master, где Slave – подчиненная станция, Master – модуль Modbus TCP Master;
- исходящий сигнал – сигнал, значение которого при изменении будет передаваться подчиненной станции. Направление передачи для сигнала от Master к Slave.

Модуль позволяет настроить сигналы доставки, которые необходимы для подтверждения доставки исходящих сигналов до подчиненной станции и содержат результат выполнения операции. Значения сигналов доставки представлены в таблице.

Модуль позволяет использовать сегмент адресного пространства Holding Registers для передачи сигналов телеуправления и телесигнализации. Регистр адресного пространства протокола Modbus используется как набор в качестве 16 сигналов телеуправления или телесигнализации. При конфигурировании такого способа доставки сигналов в адресе сигнала указывается номер бита (BitPosition).

Так как элементы адресного пространства Holding Registers и Coils имеют тип доступа на чтение и запись данных, то имеется возможность отправлять сигналы телерегулирования (телеуправления) и получать сигналы телеизмерения (телесчет, телесигнализация) на один регистр памяти. Для этого необходимо создать два сигнала типов – входящий и исходящий. При этом не будет ошибки пересечения адресов. Запись и чтение с одного регистра применяется для проверки полученных данных подчиненной станцией и для проверки текущих значений параметров объекта телерегулирования.

Протокольный тип данных, указанный для сигнала, должен быть согласован с типом данных сигнала в соответствии с таблицей. Если типы данных будут не согласованы, то сигнал не будет принят на обслуживание модулем. Правило пользования таблицей: в первом столбце выбрать тип используемых данных и соотнести ему тип сигнала.

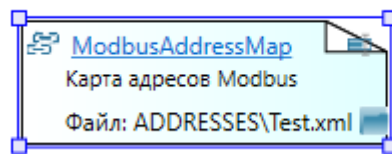
Протокольный тип	Тип в Astra.Server											
	int1	uint1	int2	uint2	int4	uint4	int8	uint8	float	double	string	bool
TS												да
TC												да
TM2			да	да*	да	да*	да	да*	да	да		
TMF4									да	да		
TMC					да	да*	да	да*				
TMF8										да		
TM2_TIME			да	да*	да	да*	да	да*				
TMF4_TIME									да	да		
TMC_TIME					да	да*	да	да*				
TMF8_TIME										да		
TR2			да	да*								
TRF4									да			
TR4					да	да*						
STR											да	
STR-COMMAND											да	

Где да\* - выполняется, если тип данных беззнаковый, иначе сигнал не принимается на обслуживание. Например, при преобразовании из протокольного типа данных TM2 в канонический тип данных uint2 необходимо в адресе сигнала указать "Signed=(False)".

## 1.1.2.3.1.4. Карта адресов

Для настройки сигналов модуля Опросчик Modbus TCP используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов Modbus в исполняемое приложение;



2. Откройте редактор карты адресов.

Тип	Привязка	Сегмент	Адрес	Номер бита	Номер записи в файле	Метка времени	Размер строки	Категория данных
int2	непосредственн	Input Registers	4			Нет		
uint2	непосредственн	Holding Registers	3			Нет		

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>› Непосредственно;</li><li>› Только чтение;</li><li>› Не привязан.</li></ul>
Сегмент	Указание, к какому сегменту памяти протокола Modbus привязан сигнал. Принимает одно из следующих значений: <ul style="list-style-type: none"><li>› Discrete Inputs (дискретные входы, один бит, только чтение);</li><li>› Coils (ячейки, один бит, чтение и запись);</li><li>› Input Registers (входные регистры, 16-битное слово, только чтение);</li><li>› Holding Registers (регистры хранения, 16-битное слово, чтение и запись).</li><li>› Files</li></ul>

Адрес	Адрес элемента данных в выбранной области памяти протокола. Значение в диапазоне от 0 до 65535
Номер бита	Номер бита в байте. Применяется с типом TS для Input Registers и Holding Registers и для TC в Holding Registers. Значения в диапазоне от 0 до 15
Номер записи в файле	Указывается при выборе сегмента Files, значение от 0 до 9999
Метка времени	Указание метки времени для протокольных типов TM2, TMF4: <ul style="list-style-type: none"> <li>&gt; Да;</li> <li>&gt; Нет.</li> </ul>
Размер строки	Размер строки в ASCII кодах. Применяется с типами STR и STR-COMMAND. Значения в диапазоне от 1 до 123
Категория данных	Категории данных нужны для группировки сигналов по интервалу опроса. Категории могут быть удобны, когда требуется оптимизировать нагрузку на сеть или сократить объем трафика.

При пересечении или совпадении адресов (например, при создании управляющего воздействия на один и тот же адрес) происходит запись в журнал модуля о наличии пересечений адресов, но к отказу в постановке сигналов на обслуживание не приводит. Проверка на совпадение и пересечение адресов происходит при старте модуля.

## 1.1.2.3.1.5. Использование сигналов

Сигналы Astra.Server используются для передачи значений параметров технологических объектов пользователю и для передачи управляющих воздействий пользователя к технологическим объектам. Сигналы привязываются к модулям через свойство адрес сигнала.

Сигнал состоит из набора свойств. Часть свойств описывает параметры технологического объекта (к ним относятся значение, качество, метка времени, а так же опциональные свойства – свойства пересчета), а другая часть описывает поведение сигнала в Astra.Server (к ним относится обязательное свойство CDT).

С точки зрения коммуникационного модуля необходимы следующие свойства сигнала:

- › 1 (CDT) – канонический тип данных;
- › 2 (Value) – инженерное значение;
- › 3 (Quality) – качество сигнала;
- › 4 (Timestamp) – метка времени;
- › 5000 (Address) – адрес сигнала;
- › 5001 (Active) – активный протокол обмена;
- › 5002 (RawValue) – физическое значение;
- › 5100 - 5108 – свойства пересчета.

### Свойства сигналов

Свойства 1 (CDT), 2 (Value), 3 (Quality), 4 (Timestamp), 5 (AccRight), 6 (ScanRate) являются обязательными. Если в конфигурации сервера обязательные свойства не заданы пользователем, то при старте сервера, содержащего модуль, данные свойства создаются динамически. Инициализирующие значения зависят от свойства, например, 2 (Value) – EMPTY, 4 (Timestamp) – NOW и т.п.

Общие свойства сигнала для всех модулей:

›1 (CDT) – канонический тип данных. Принимает значение равное коду указывающему тип сигнала;

Код	Тип сигнала
1	Int1
3	UInt1
9	Int2
8	UInt2
7	Int4
6	UInt4
13	Int8
12	UInt8
14	Float
15	Double
5	Bool
17	String

›2 (Value) – инженерное значение. Имеет тот же тип, что и сигнал. Содержит значение, с которым работает пользователь;

›3 (Quality) – качество сигнала. Тип свойства uint4. Показывает степень достоверности данных сигнала;

›4 (Timestamp) – метка времени. Содержит значение времени появления события.

Свойства сигнала для коммуникационного модуля:

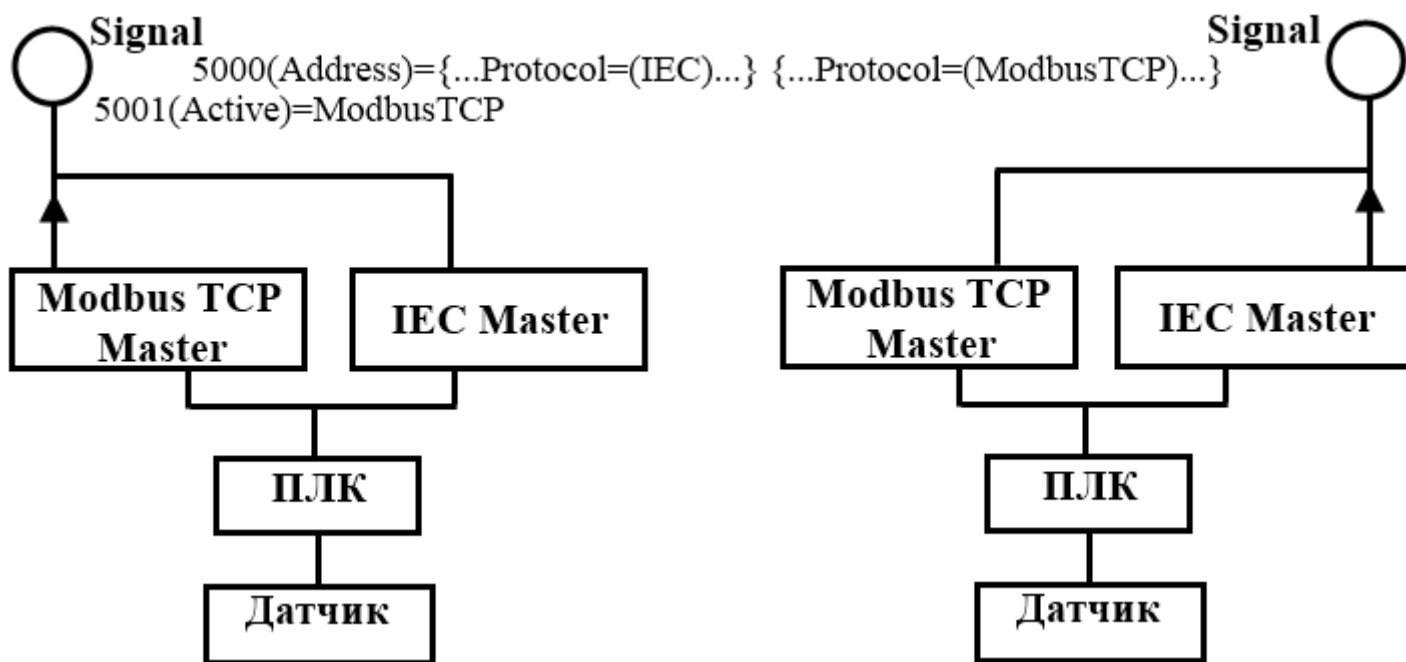
›5000 (Address) – адрес сигнала. Свойство создается пользователем. Сигнал может быть привязан к одному или нескольким коммуникационным



модулям. Связь осуществляется через указание адреса в рамках специфики протокола в свойстве 5000 (Address). Значение свойства адрес сигнала для каждого модуля заключается в фигурные скобки {}. Значение параметра помещается в круглые скобки, а между именем параметра и значением ставится знак "=".

В одном свойстве может находиться несколько адресов, записанных в формате {адрес\_1}{адрес\_2}...{адрес\_N}.

>5001 (Active) – активный протокол обмена. Свойство создается пользователем. Сигнал может быть привязан к одному или нескольким коммуникационным модулям, для этого необходимо указать, по какому протоколу принимать и сохранять данные (рисунок ниже). Если активный протокол не указан, то таким будет первый по списку в адресе сигнала;

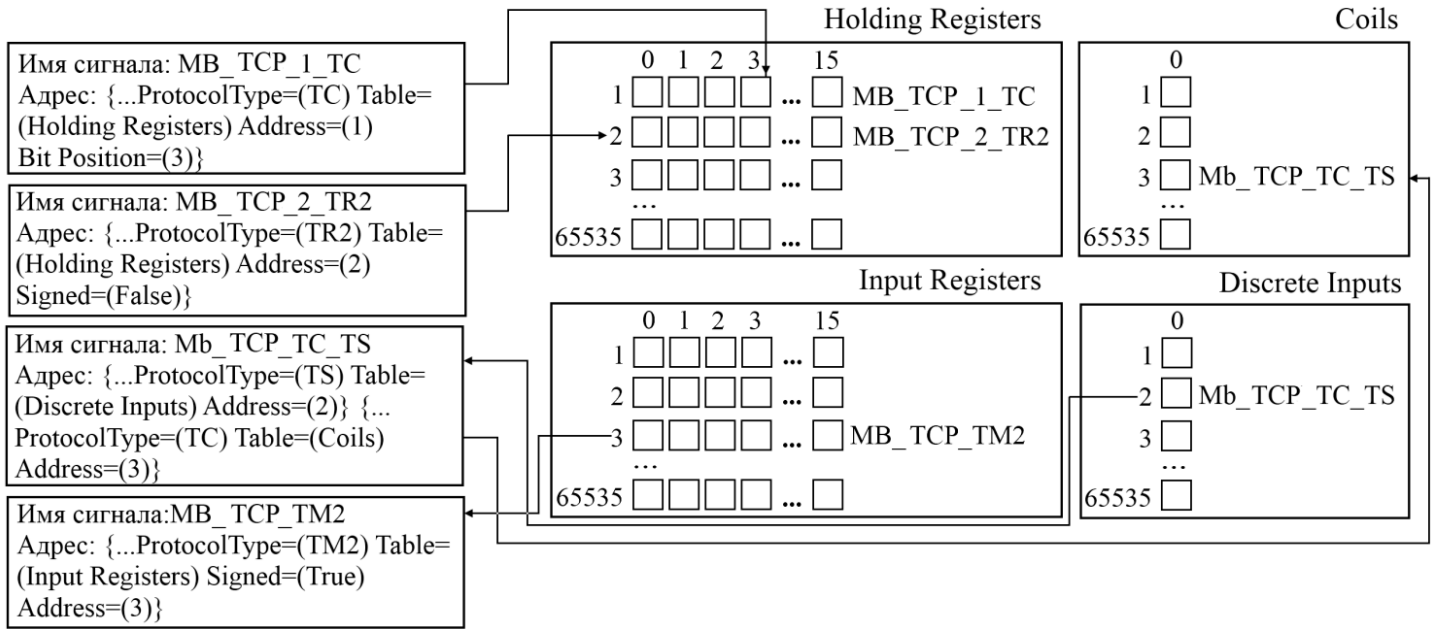


>5002 (RawValue) – физическое значение. Создается сервером динамически. Тип свойства должен соответствовать каноническому типу сигнала. При создании свойства активируются функции пересчета в инженерное значение (свойство 2 (Value)) и обратно.

Структура формата адреса для модуля Modbus TCP Master представлена в таблице:

Параметр	Значение
ModuleId	Идентификатор модуля
Protocol	ModbusTCP
Station	Номер станции
Type	Тип адреса сигнала. Принимает одно из двух значений: <ul style="list-style-type: none"> <li>› Protocol - значение указывает, что тип относится к типу, передаваемому в параметре ProtocolType;</li> <li>› DeliveryStatus - сигнал доставки. Тип сигнала доставки должен быть определен в параметре DeliveryStatusType.</li> </ul>
Table	Указание, к какому сегменту памяти протокола Modbus привязан сигнал. <ul style="list-style-type: none"> <li>› Discrete Inputs (дискретные входы, один бит, только чтение);</li> <li>› Coils (ячейки, один бит, чтение и запись);</li> <li>› Input Registers (входные регистры, 16-битное слово, только чтение);</li> <li>› Holding Registers (регистры хранения, 16-битное слово, чтение и запись).</li> </ul>
Address	Адрес элемента данных в выбранной области памяти протокола. Значение в диапазоне от 0 до 65535
ProtocolType	Протокольный тип данных по спецификации Modbus
Signed	Признак того что тип является знаковым. Применяется с типами TM2 и TM4
BitPosition	Номер бита в байте. Применяется с типом TS для Input Registers и Holding Registers. Значения в диапазоне от 0 до 15
Length	Размер строки в ASCII кодах. Применяется с типами STR и STR-COMM

При пересечении или совпадении адресов (например, при создании управляющего воздействия на один и тот же адрес) происходит запись в журнал модуля о наличии пересечений адресов, но к отказу в постановке сигналов на обслуживание не приводит. Проверка на совпадение и пересечение адресов происходит при старте модуля.



## 1.1.2.3.1.6. Диагностика работы модуля

### Диагностика станций через сигналы

Модуль Modbus TCP Master динамически создает служебные сигналы для подключенных станций. Через служебные сигналы, созданные модулем, пользователь имеет возможность наблюдать за работой станций. Права доступа к служебным сигналам только на чтение

Журнал работы модуля сохраняется в файл <имя модуля>.aplog по умолчанию:

› в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

› в Linux системах в директории:



/opt/AstraRegul/Astra.Server/Logs.

### Сигнал состояния связи по каналу

Тип: bool.

Тег сигнала:



Astra.OPCDAServer.Service.Modules.<Идентификатор модуля>.Station<номер станции>.Channels.<IP-адрес канала>:<порт>.ConnectionState

Значения:

- › True – связь по каналу установлена.
- › False – связь по каналу отсутствует.

### Сигнал состояния связи со станцией

Тип: int4.

Тег сигнала:



Astra.OPCDAServer.Service.Modules.<Идентификатор модуля>.Station  
<номер станции>. ConnectionState

Значения:

- › 0 – соединение отсутствует.
- › 1 – ожидание переподключения в случае кратковременной потери связи.
- › 2 – соединение есть.
- › 3 – запрашиваются или передаются данные.

## Сигнал физического соединения со станцией

Тип: bool.

Тег сигнала:



Astra.OPCDAServer.Service.Modules.<Идентификатор модуля>.Station  
<номер станции>. ConnectionState.ConnectionState

Значения:

- › True – соединение со станцией установлено.
- › False – соединение со станцией отсутствует.

## Сигнал логического соединения со станцией

Тип: bool.

Тег сигнала:



Astra.OPCDAServer.Service.Modules.<Идентификатор модуля>.Station  
<номер станции>. ConnectionState.Logical

Значения:

- › True – соединение со станцией установлено.
- › False – соединение со станцией не восстановлено после истечения таймаута потери связи.

## Количество принятых ответов от подчиненной станции

Значение инкрементируется при получении ответа от станции, вне зависимости от данных, находящихся в ответе.

Тег сигнала:



Astra.OPCDAServer\Service.Modules.<Идентификатор модуля>.Station <номер станции>.ReceivedResponseCount

## Количество отправленных запросов подчиненной станции

Тег сигнала:



Astra.OPCDAServer\Service.Modules.<Идентификатор модуля>.Station <номер станции>. SentRequestCount

## Журнал работы модуля

Каждый модуль ведет журнал работы. В журнал работы модуля сохраняется вся информация о работе модуля и об обмене данными с подчиненными станциями.

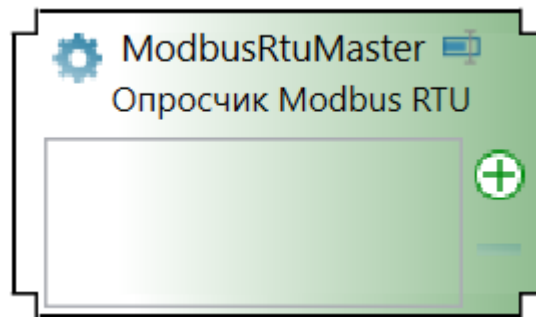
Журнал работы модуля предназначен для контроля работы модуля в режиме реального времени, а так же просмотра прошедших событий модуля. В окне приложения Просмотрщик лога кадров отобразятся записи журнала работы модуля в виде таблицы со столбцами, которые соответствуют набору свойств модуля.

№	Дата	Время	Описание	Станция	У.	Адрес	Hex	Знаковое	Беззнаковое
19419	04.07.2023	17:14:39:114	Входящий кадр Транзакция 37739, Функция 3	1	1..	0	0	0	0
19420	04.07.2023	17:14:39:145	Исходящий кадр Транзакция 37740, Функция 3, Адрес 0	1	1..				
19421	04.07.2023	17:14:39:145	Входящий кадр Транзакция 37740, Функция 3	1	1..				
19422	04.07.2023	17:14:39:177	Исходящий кадр Транзакция 37741, Функция 3, Адрес 0	1	1..				
19423	04.07.2023	17:14:39:177	Входящий кадр Транзакция 37741, Функция 3	1	1..				
19424	04.07.2023	17:14:39:208	Исходящий кадр Транзакция 37742, Функция 3, Адрес 0	1	1..				
19425	04.07.2023	17:14:39:208	Входящий кадр Транзакция 37742, Функция 3	1	1..				
19426	04.07.2023	17:14:39:239	Исходящий кадр Транзакция 37743, Функция 3, Адрес 0	1	1..				
19427	04.07.2023	17:14:39:239	Входящий кадр Транзакция 37743, Функция 3	1	1..				
19428	04.07.2023	17:14:39:270	Исходящий кадр Транзакция 37744, Функция 3, Адрес 0	1	1..				
19429	04.07.2023	17:14:39:270	Входящий кадр Транзакция 37744, Функция 3	1	1..				
19430	04.07.2023	17:14:39:301	Исходящий кадр Транзакция 37745, Функция 3, Адрес 0	1	1..				

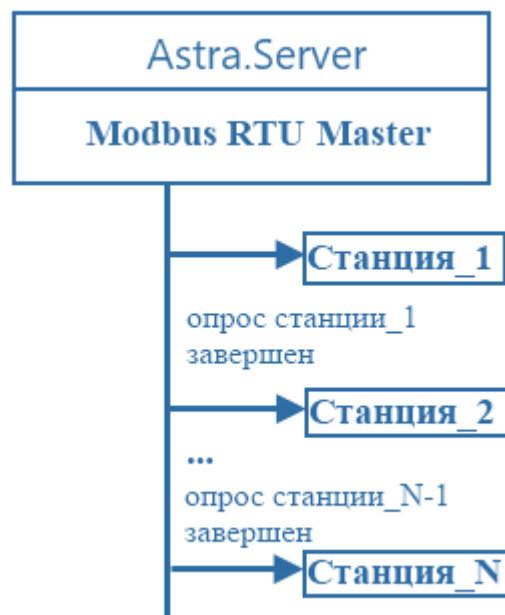
  

Адрес	Hex	Знаковое	Беззнаковое
0000	93 €D 00 00 00 05 01 03 02		
0001	00		

## 1.1.2.3.2. Опросчик Modbus RTU



Модуль Modbus RTU Master работает в составе Astra.Server, который является частью системы для сбора, анализа и регулирования параметров технологического процесса.

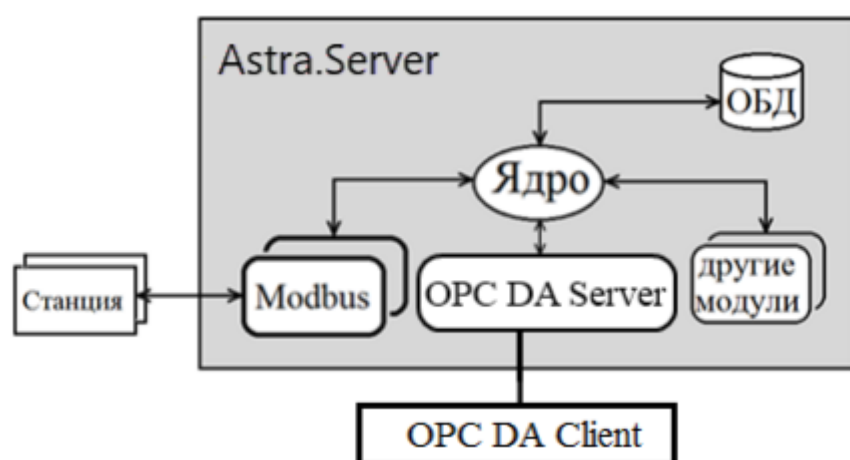


Модуль Modbus RTU Master предназначен для последовательного опроса подчиненных станций, подключенных к COM-порту компьютера, в соответствии с протоколом Modbus over Serial Line. Astra.Server поддерживает до 64-х модулей Modbus RTU Master.

Основными функциями модуля Modbus RTU Master являются сбор данных и подача управляющих воздействий подчиненным станциям.

Работая в составе Astra.Server, модуль Modbus RTU Master опрашивает подчиненные станции, циклически отправляя им запросы. Полученные данные через интерфейсы ядра сохраняются в оперативную базу данных (ОБД). Ядро Astra.Server при получении изменившихся сигналов уведомляет о получении изменений другие модули, которые обслуживают эти сигналы, в том числе и модуль OPC DA Server. Модуль OPC DA Server, в свою очередь, передает данные клиентам, подписанным на изменившиеся сигналы.

Отправка управляющих воздействий происходит в обратной последовательности.



Модуль Modbus RTU Master отправляет запросы подчиненным устройствам по последовательным линиям связи согласно протоколу Modbus RTU. Модуль имеет возможность работать только с одной станцией одновременно. Запросив данные и получив ответ, модуль разрывает соединение с одной станцией и подключается к другой. Подчиненное устройство отвечает на запрос, адресованный именно ему. В случае если через определенный промежуток времени (Время ожидания ответа от станции) ответ модулю от подчиненной станции не приходит, то соединение считается разорванным. При отправке запроса модуль имеет возможность получать сигнал доставки сообщения.

Модуль работает только в режиме РАБОТА. В режиме РЕЗЕРВ модуль не работает, исходя из того, что модуль работает по последовательным линиям связи и при резерве тратится большой трафик и замедляется работа модуля.



При старте модуля происходит инициализация модуля. Процесс инициализации включает в себя:

- › чтение конфигурационных данных модуля;
- › подписка у ядра на получение уведомлений об изменениях сигналов;
- › проверка корректности настройки адресов сигналов принадлежащих модулю и постановка их на обслуживание;
- › формирование таблицы поллинга.

После вышеперечисленного модуль устанавливает соединение и начинает работу в соответствии со своими настройками.

При установленном соединении модуль начинает опрос станции по заданным адресам.

Адресное пространство протокола представляет собой четыре массива данных:

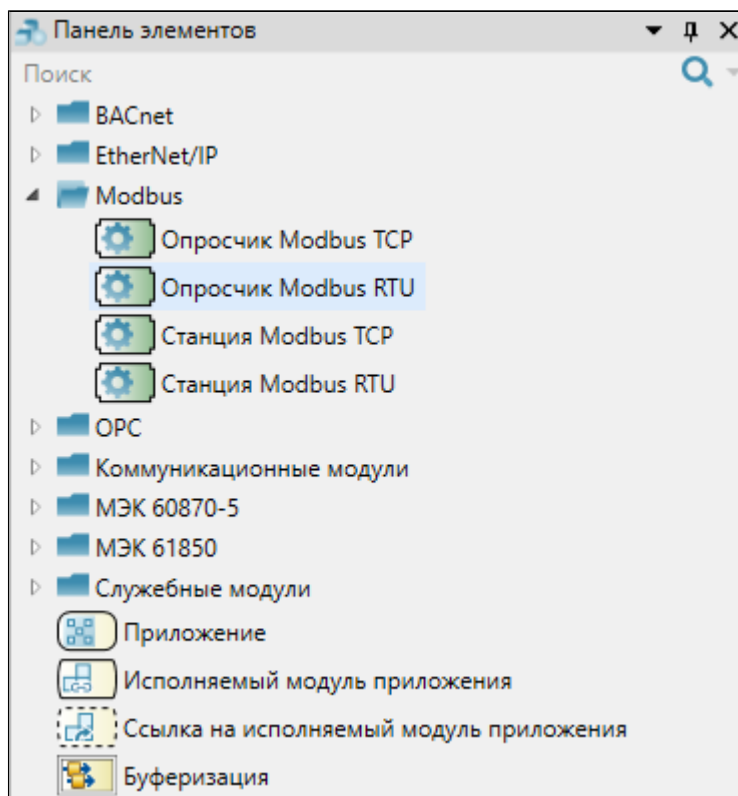
- › Coils - ячейки. Одиночный бит. Тип доступа на чтение и запись;
- › Discrete Inputs - дискретные входы. Одиночный бит. Тип доступа только на чтение;
- › Holding Registers - регистры хранения. 16-битное слово. Тип доступа на чтение и запись;
- › Input Registers - входные регистры. 16-битное слово. Тип доступа только на чтение.

Опрос каждой подчиненной станции ведется по таблице поллинга, сформированной при старте модуля. Формирование таблицы поллинга ведется по использованным при конфигурировании адресам в массивах данных.

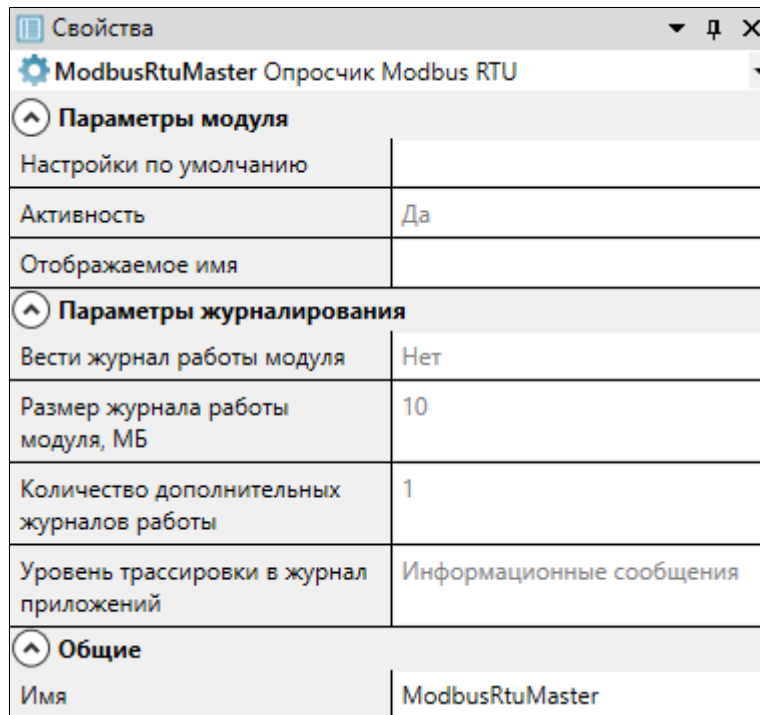
В модуле реализовано расширение функций Modbus для передачи времени возникновения данных.

## 1.1.2.3.2.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуриатора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.



## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Настройки по умолчанию	Выбор файла с готовой конфигурацией
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен;</li> <li>&gt; Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.3.2.2. Функциональные возможности

### Обмен данными по протоколу Modbus

Обмен данными по протоколу Modbus основан на архитектуре Master – Slave. Модуль Modbus RTU Master выполняет обмен данными с подчиненными станциями, реализуя функции Master в соответствии со спецификацией MODBUS APPLICATION PROTOCOL SPECIFICATION.

Модуль инициирует обмен данными, отправляя запрос подчиненной станции. Подчиненная станция отвечает на запрос, предоставляя запрошенные данные или выполняя команду модуля. Самостоятельно подчиненная станция данные не передает.

Запросы и ответы представляют собой кадры данных. Для модуля Modbus RTU Master запрос, отправляемый подчиненной станции, является исходящим кадром, а ответ, полученный от подчиненной станции – входящим кадром.

Исходящий кадр содержит код функции и данные, необходимые для выполнения функции. Входящий кадр содержит код функции и данные, полученные в результате выполнения функции. Подробное описание кадров данных модуля Modbus RTU Master приведено в разделе Журнал работы модуля.

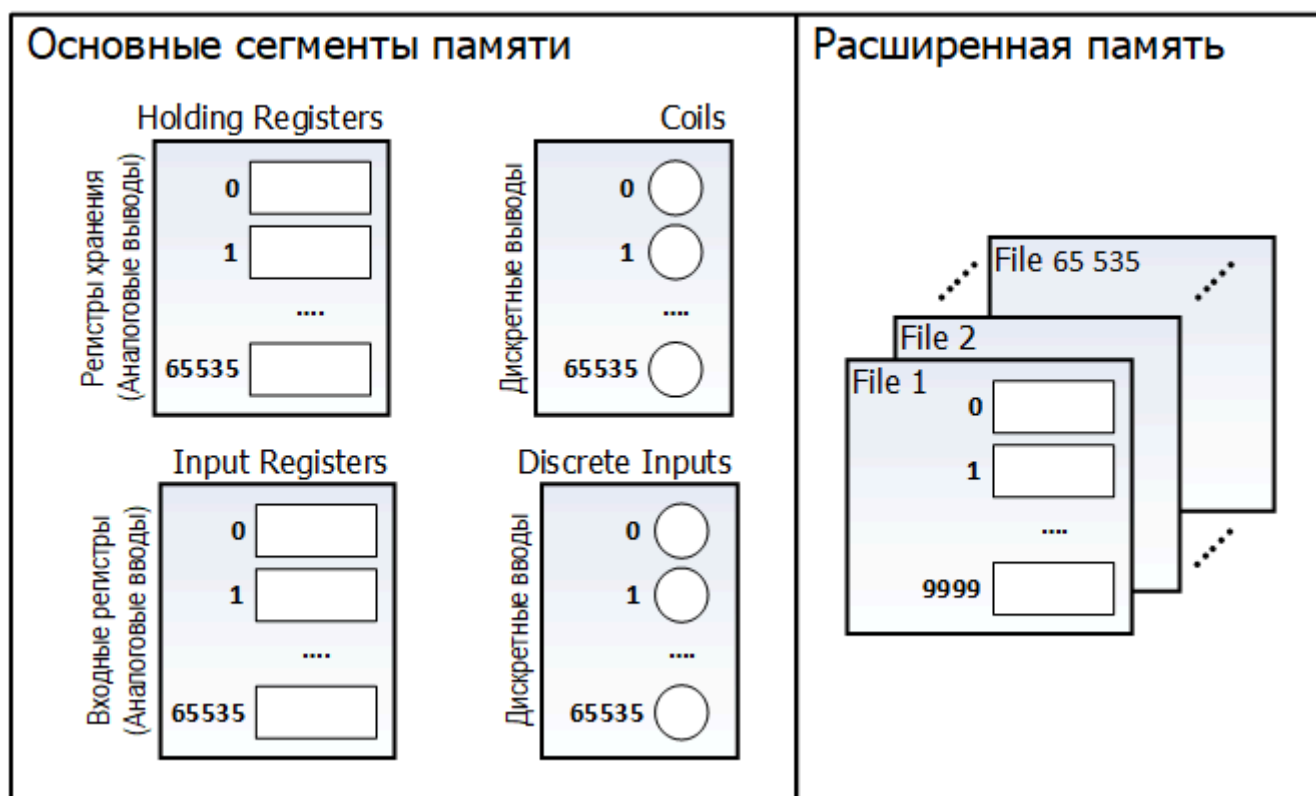
Код функции определяет сегмент памяти подчиненной станции и действие, которое требуется выполнить в данном сегменте.

### Сегменты памяти подчиненной станции

Модуль Modbus RTU Master выполняет чтение и запись данных в четыре основных сегмента памяти, а также чтение данных из файлов расширенной памяти подчиненной станции. В таблице приведены сегменты памяти подчиненной станции, тип и размер элементов сегмента, а также действия, допустимые для сегментов.

Сегмент	Описание	Тип и размер элемента
Coils	Дискретные выходы (Телеуправление)	Ячейка 1 бит
Discrete Inputs	Дискретные входы (Телесигнализация)	Ячейка 1 бит
Holding Registers	Регистры хранения (Телерегулирование)	Регистр 2 байта
Input Registers	Регистры ввода (Телеизмерения)	Регистр 2 байта
Files	16-битные регистры, организованные в файлы, состоящие из записей произвольной длины	Регистр 2 байта

Каждый сегмент памяти содержит 65 536 элементов, адреса которых от «0» до «65535». Расширенная память содержит файлы с номерами от «1» до «65535», каждый из которых содержит от 0 до 9999 записей.



## Поддерживаемые функции протокола Modbus

Стандартные функции протокола Modbus, используемые модулем Modbus RTU Master для обмена данными с подчиненной станцией, приведены в таблице.

Код функции	Название	Действие	Сегмент
01 (0x01)	Read Coils	Чтение состояния дискретных выводов	Coils
02 (0x02)	Read Discrete Inputs	Чтение состояния дискретных вводов	Discrete Inputs
03 (0x03)	Read Holding Registers	Чтение регистров хранения	Holding Registers
04 (0x04)	Read Input Registers	Чтение входных регистров	Input Registers
05 (0x05)	Write Single Coil	Запись одного дискретного вывода	Coils
06 (0x06)	Write Single Register	Запись одного регистра хранения	Holding Registers
16 (0x10)	Write Multiple registers	Запись нескольких регистров хранения	Holding Registers
20 (0x14)	Read File Record	Чтение данных из файла	Files

### Чтение данных

Для чтения данных из сегментов памяти подчиненной станции используются стандартные функции протокола Modbus:

- › «01 (0x01) Read Coils» – чтение состояния дискретных выводов;
- › «02 (0x02) Read Discrete Inputs» – чтение состояния дискретных вводов;
- › «03 (0x03) Read Holding Registers» – чтение регистров хранения;

- › «04 (0x04) Read Input Registers» – чтение входных регистров;
- › «20 (0x14) Read File Record» – чтение данных из файла.

Максимальное количество считываемых элементов за один запрос для каждого сегмента задаётся в параметрах модуля. Чтобы полученные от подчиненной станции данные записывались в сигнал сервера, в адресе сигнала (свойство 5000) необходимо указать параметры:

- › номер подчиненной станции;
- › сегмент памяти, из которого считываются данные;
- › адрес элемента сегмента памяти (для сегмента Files – номер файла и номер записи);
- › протокольный тип.

## Запись одного значения

Для записи одного значения в сегмент памяти подчиненной станции используются стандартные функции протокола Modbus:

- › «05 (0x05) Write Single Coil» – запись значения одного дискретного вывода;
- › «06 (0x06) Write Single Register» – запись значения в один регистр хранения.



Чтобы использовать функцию «06 (0x06) Write Single Register» для записи значений в регистры хранения, в настройках станции параметру Используемая функция записи значений в регистры установите значение «0x06 Write Single Register».

Чтобы значение сигнала сервера записывалось в сегмент данных подчиненной станции, в адресе сигнала (свойство 5000) необходимо указать параметры:

- › номер подчиненной станции;
- › сегмент памяти, в который требуется записать данные;
- › адрес изменяемого элемента сегмента памяти;
- › протокольный тип.

## Запись нескольких значений



Для записи значений в несколько регистров памяти подчиненной станции используется стандартная функция протокола Modbus «16 (0x10) Write Multiple Registers».



Чтобы использовать функцию «16 (0x10) Write Multiple Registers» для записи значений в регистры хранения, в настройках станции параметру Используемая функция записи значений в регистры установите значение «0x10 Write Multiple Registers».

Максимальное количество записываемых элементов за одну команду задаётся в параметрах модуля.

Чтобы значение сигнала сервера записывалось в сегмент данных подчиненной станции, в адресе сигнала (свойство 5000) необходимо указать параметры:

- › номер подчиненной станции;
- › сегмент памяти, в который требуется записать данные;
- › адрес первого элемента сегмента памяти, значение которого требуется изменить;
- › протокольный тип.

## Стратегия формирования запросов на чтение

Модуль Modbus RTU Master предоставляет выбор варианта составления таблицы поллинга, т.е. возможность регулировки максимальной длины запроса на чтение, и выбор варианта учитывать или не учитывать разрывы данных в адресном пространстве протокола.

Существуют два варианта формирования таблицы поллинга. Один из них позволяет оптимизировать количество запросов, другой объем запрашиваемых данных. Выбор варианта формирования таблицы поллинга производится при конфигурировании модуля:

- › **По максимуму** - строится карта запросов максимальной длины, не учитывая фрагментацию запрашиваемых данных – объединение в один запрос расположенных не подряд данных. Например, если есть 2 сигнала с адресом 1 и 125, они объединяются в один запрос, лишние данные при

получении игнорируются (т.е. игнорируются данные с адресами от 2 до 124). Таким образом, оптимизируется количество отправляемых запросов за счет увеличения трафика (количества переданной информации);

➤ **На сплошные данные** - строится карта запросов только по необходимым данным. Таким образом, оптимизируется трафик за счет потери времени. Потеря времени происходит из-за необходимости построения и отправки нескольких запросов.

Максимальная длина запроса равна 125 регистров. Длина запроса задается в параметрах конфигурации модуля.



#### Формирование карты поллинга

Если заданы адреса на чтение 1, 7-25, 47, 130 то для стратегии формирования запросов:

➤ **"по максимуму"** будут сформированы два запроса. Один из запросов по адресам от 1 до 125, второй запрос от 126 до 130;

➤ **"на сплошные данные"** будет сформировано 3 запроса: по адресу 1, по адресам от 7 до 25, по адресу 47 и по адресу 130.

Максимальная длина запроса равна 125 регистров. Длина запроса задается в параметрах конфигурации модуля. Сигналы с типами больше одного регистра отправляются и принимаются за один запрос независимо от настроек длины запроса.

## Опрос подчиненной станции

Опрос нескольких подчиненных станций ведется модулем по очереди, порядок станций определяется таблицей поллинга.

Модуль отправляет запросы каждой подчиненной станции последовательно, то есть отправляет станции один запрос и ждет ответа на запрос. Время между отправкой запросов задается в параметрах конфигурации.

При запуске модуля, он находится в состоянии Покоя – служебный сигнал состояния связи со станцией принимает значение 2 – соединение есть, т.е. не отправляет запросов и не ожидает ответов. Запрос может быть отправлен только из состояния Покоя. После отправки запроса модуль выходит из состояния Покоя (служебный сигнал состояния связи со станцией принимает значение 3 запрашиваются или передаются данные) и не может отправлять повторные запросы.

Когда запрос отправлен подчиненной станции, модуль переходит в состояние Ожидание ответа и запускается счетчик Времени ожидания ответа от станции. При получении ответа от подчиненной станции модуль проверяет его на ошибки. В случае ошибки Время ожидания ответа от станции истекает и выставляется состояние потеря связи со станцией качество сигнала принимает значение COMM\_FAILURE. Затем модуль переходит в состояние Покоя, что позволяет ему повторить запрос.

Для того чтобы выдержать паузу между приемом и передачей предусмотрен параметр Пауза между запросами. Функция предназначена для устройств, не отвечающих на запросы без паузы, т.е. параметр является устройством зависимым.

Максимальное количество повторов запросов на чтение и максимальное количество повторов на отправку команд задается в конфигурации модуля. Когда количество повторов превышает заданное максимальное количество, то регистрируется потеря связи со станцией.

Если в ответ на запрос подчиненная станция вернула строку SLAVE\_DEVICE\_BUSY, то повтор отправки запроса будет произведен после Паузы опроса занятого устройства. При этом необходимо чтобы Пауза опроса занятого устройства была меньше, чем Время ожидания ответа от станции.

Модуль Modbus RTU Master динамически создает служебные сигналы для каждой станции, предназначенные для обеспечения контроля работы модуля. Через служебные сигналы созданные модулем пользователь имеет возможность наблюдать за работой модуля.

При опросе и выдаче управляющих воздействий модуль имеет возможность изменять порядок байт:

- Байт в слове - включает/отключает изменение порядка следования байтов в регистре. В случае если флаг установлен, то байты регистра меняются местами.



Действие флага распространяется на все протокольные типы сигналов, кроме TS и TC.

- Слов в сигналах TMC, TR4 — включает/отключает изменение порядка следования слов в сигналах TMC, TR4. В случае если флаг установлен, то слова данных меняются местами.



Если одновременно установлены флаги Слов в сигналах TMC, TR4 и Байт в слове, то слова данных меняются местами и байты в словах меняются местами;



- Слов в сигналах TMF4, TMF8, TRF4 — включает/выключает изменение порядка слов в сигналах TMF4, TMF8, TRF4. В случае если флаг установлен, изменение порядка следования слов происходит аналогично изменению порядка следования слов в сигналах TMC, TR4. Если одновременно установлены флаги Слов в сигналах TMF4, TMF8, TRF4 и Байт в слове, то пары байтов меняются местами и в каждой паре байты меняются местами (рисунок выше).

Функция изменения порядка байт необходима для работы с определенными устройствами, которые на аппаратном уровне меняют местами байты в словах.

## Качество сигналов

В протоколе Modbus отсутствует понятие качества данных, и поэтому качество устанавливается модулем, исходя из результатов выполнения запросов. При старте модуля все сигналы имеют плохое качество.

Качества сигналов, выставляемых коммуникационным модулем, приведены в таблице:

<b>Значение качества</b>	<b>Идентификатор качества</b>	
4	CONFIG_ERROR	Сигнал неправильно сконфи
8	NOT_CONNECTED	Сигнал принят на обслужива
12	DEVICE_FAILURE	Ошибка инициализации устр
20	LAST_KNOWN	КП нет на связи, последнее п
24	COMM_FAILURE	КП нет на связи
28	OUT_OF_SERVICE	Модуль не запущен
64	UNCERTAIN	Связь установлена, но значе
192	GOOD	Значение сигнала достоверн
216	LOCAL_OVERRIDE	Значение достоверно, введе

Пропадание связи регистрируется, если за Время ожидания ответа от станции подчиненная станция не отвечает. При отсутствии ответа от подчиненной станции происходит повтор отправки команд (запросов). По истечению максимального числа повторов связь с подчиненной станцией считается потерянной и выставляется плохое качество сигналов (COMM\_FAILURE). При кратковременном пропадании связи станция успевает ответить за отведенное Количество повторов отправки команд (запросов) и качество сигналов при кратковременном пропадании связи не изменяется.



Модуль имеет возможность обрабатывать ошибки, чтобы не выставлять плохое качество данных при возникновении единичных сбоев в поведении устройства и чтобы сохранить целостную картину при резервных переходах устройств. Алгоритмы обработки ошибок следующие:

- если устройство при запросе данных возвращает строку SLAVE DEVICE BUSY (запись строки происходит в поле качество сигнала), то запрос данных не повторяется, связь со станцией не меняется, модуль переходит к запросу следующих данных. Недоставленные данные будут перезапрошены в следующем цикле опроса;
- если устройство при запросе данных возвращает строку ILLEGAL DATA ADDRESS (запись строки происходит в поле качество сигнала), либо другой код ошибки, то по данному запросу выставляется качество OPC\_QUALITY\_BAD;
- если подчиненная станция не отвечает и количество повторов отправки команд (запросов) достигло максимального, то сигналам по данному запросу выставляется качество OPC\_QUALITY\_BAD.

## Метка времени

Протокол Modbus не оперирует понятием время возникновения данных, обычно сигналам присваивается время сервера на момент получения данных, свойство 4 (TimeStamp).

Модуль Modbus RTU Master реализует расширение спецификации в части передачи метки времени. Метка времени может быть использована как для формирования в контроллере времени изменения значения, так и для формирования однозначной последовательности событий, возникающих на уровне контроллера, что невозможно в обычных условиях, так как опросчик может не успеть прочитать данные, если данные очень быстро изменяются.

Для передачи метки времени параметров вводятся дополнительные типы данных в модуле Modbus RTU Master: TM2\_TIME, TMF4\_TIME, TMC\_TIME, TMF8\_TIME.

Протокольные типы данных с меткой времени представляются логической структурой, представленной в таблице:

Рег./Поз.	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
1-4	Данные занимают от 1 до 4 регистров в зависимости от типа сигнала															
5	секунды						миллисекунды									
6	час								минуты							
7	год							месяц				день				

Для возможности взаимодействия модуля и контроллера на стороне контроллера следует поддерживать формирование описанной структуры.

Модуль, работая с сигналом, который имеет тип с меткой времени, формирует запрос на чтение с учетом типа сигнала и дополнительной структуры длиной три регистра. При получении данных метка времени, лежащая в дополнительной структуре, преобразуется к стандартному времени и записана в свойство 4 (TimeStamp) одновременно с записью значения в свойство 2 (Value).

Для получения метки времени, у подчиненной станции должен быть поддержан формат структур (таблица выше) по сигналам, которые должны содержать метку времени. Поля структуры должны заполняться на стороне подчиненной станции. В Astr.Server должны быть указаны протокольные типы данных с меткой времени для сигналов.

## Отправка управляющих и регулирующих воздействий

Отправка команд телеуправления и телерегулирования выполняется, только когда пара резервируемых серверов активна и Astra.Server находится в режиме РАБОТА.

Отправка управляющих воздействий имеет больший приоритет, чем опрос станций. При необходимости отправки управляющего воздействия модуль вставляет пакет, содержащий команду в начало очереди запросов на отправку подчиненной станции.

Модуль позволяет указать количество повторов отправки команд в случае неуспешной отправки. Повторы отправки команд происходят, если:

- устройство вернуло в ответ исключение с кодом SLAVE DEVICE BUSY. Если в итоге запись так и не была выполнена, то состояние связи выставляется в положение отсутствует;
- вышло время ожидания ответа от станции, т.е. произошел разрыв связи. После истечения максимального количества повторов отправки команд (запросов) сигналам выставляется плохое качество.

Качество данных является показателем достоверности информации. Управляющие сигналы с недостоверной информацией отправлять недопустимо. Сигналы управления плохого качества не отправляются.

Для исходящих сигналов модуль позволяет настраивать сигналы доставки управляющих и регулирующих воздействий до подчинённой станции, которые формируются по правилам из таблицы:

Значение	Описание
2	Доставка подтверждена
1	Ожидание подтверждения доставки
-1	Нет связи со станцией
-2	Вышел таймаут подтверждения запроса



-1000 + код исключения	Устройство вернуло код исключения
---------------------------	-----------------------------------

В случае если подчиненная станция получает запрос, но не может его обработать (например, чтение несуществующей ячейки или регистра), то подчиненная станция возвращает ответ модулю с кодом ошибки.

Модуль имеет возможность отправлять все сигналы телерегулирования, при проявлении подчиненной станции на связи. При этом может произойти отправка сигнала, содержащего устаревшую информацию, из-за этого функцию необходимо использовать осторожно.

Отправка сигнала телеуправления (ТС) может проводиться в сегменте Holding Registers адресного пространства. Для возможности подачи телеуправления при конфигурировании сигнала следует указать номер бита, который будет устанавливаться в 1 при отправке телеуправления. При подаче телеуправления модуль формирует регистр для записи в устройство таким образом, что все биты, кроме указанного в настройках адреса сигнала телеуправления, будут сброшены в ноль. Например, если в адресе сигнала записано "BitPosition=(5)", то при отправке ТС в Holding Registers запишется число "00000000 00010000" или "32" в зависимости от настроек подчиненной станции.

## Параметры СОМ порта

Для подключения к подчиненной станции необходимо настроить параметры СОМ порта, при этом значения параметров СОМ порта подчиненной станции должны совпадать со значениями параметров СОМ порта модуля. Настройка параметров производится в конфигураторе.

К параметрам СОМ порта относятся:

› Номер СОМ порта - параметр предназначен для определения СОМ порта;

› Скорость (бит/с) - параметр предназначен для установки максимальной скорости передачи данных;

› Четность - параметр предназначен для обнаружения ошибок при приеме/передаче данных. Для обнаружения ошибок производится подсчет бита четности, т.е. определяется устанавливать бит при четном или нечетном числе единиц. Проверка на четность позволяет определить возникновение ошибок в одном бите, при наличии ошибок в двух битах ошибка не обнаружится. Контроль на четность не указывает на ошибочный бит;

› Стоповые биты - параметр предназначен для обнаружения ошибок при приеме/передаче данных, позволяет провести синхронизацию при возникновении сбоев. Например, помеха на линии скрыла Старт бит. Период между старт и стоп битами постоянен, согласно значению скорости обмена, числу бит данных и бита четности. Стоп бит всегда включен. Если приемник определяет выключенное состояние, когда должен присутствовать стоп бит, фиксируется появление ошибки;

› Управление потоком. Управление потоком позволяет управлять передаваемыми данными. Если при приеме/передаче данные не могут обработаться приемным устройством, то используется управление потоком для прекращения передачи данных. Модуль имеет возможность использовать аппаратное или программное управление потоком:

› Аппаратное управление потоком. При аппаратном управлении потоком используется дополнительно два провода в кабеле. Поэтому аппаратное управление потоком не замедляет обмен. При необходимости послать данные передатчик устанавливает сигнал на линии RTS. Если приемник готов к приему данных, то он отвечает установкой сигнала на линии CTS, и передатчик начинает посылку данных. При неготовности к приему сигнал CTS не устанавливается.

› Программное управление потоком. При этом используется два символа: Хон и Хoff. Код ASCII символа Хон 17, а ASCII код Хoff 19. Если принимающее устройство не успевает обрабатывать данные, то посылает символ

Xoff передающему для прекращения посылки данных. При появлении возможности приема данных посылается символ Xon и передающий продолжит пересылку данных. Этот тип управления имеет преимущество в том, что не требует дополнительных линий, т.к. символы передаются по линиям TD/RD. Но на медленных соединениях приводит к значительному замедлению соединения, т.к. каждый символ требует 10 бит.

## 1.1.2.3.2.3. Типы данных

Модуль Modbus RTU Master, оперируя данными в соответствии с протоколом, может предоставлять их в виде следующих типов (таблица ниже). Такие типы данных называются протокольными.

Протокольный тип	Тип значения	Направление передачи	Описание
TS	bool	Slave → Master	Телесигнализация
TC	bool	Master → Slave	Телеуправление
TM2	int2/uint2	Slave → Master	Телеизмерение
TMF4	float	Slave → Master	Телеизмерение
TMC	int4/uint4	Slave → Master	Телесчет
TMF8	double	Slave → Master	Телеизмерение
TM2_TIME	int2/uint2	Slave → Master	Телеизмерение с меткой времени
TMF4_TIME	float	Slave → Master	Телеизмерение с меткой времени
TMC_TIME	int4/uint4	Slave → Master	Телесчет с меткой времени
TMF8_TIME	double	Slave → Master	Телеизмерение с меткой времени
TR2	int2/uint2	Master → Slave	Телерегулирование
TRF4	float	Master → Slave	Телерегулирование
TR4	int4/uint4	Master → Slave	Телерегулирование
STR	string	Slave → Master	Телеизмерение
STR-COMMAND	string	Master → Slave	Телерегулирование

Как видно из таблицы сигналы по отношению к модулю делятся по направлению передачи на две группы:

- входящий сигнал, в который будет записываться полученное значение с подчиненной станции. Направление передачи для сигнала от Slave к Master, где Slave – подчиненная станция, Master – модуль Modbus RTU Master;
- исходящий сигнал – сигнал, значение которого при изменении будет передаваться подчиненной станции. Направление передачи для сигнала от Master к Slave.

Модуль позволяет настроить сигналы доставки, которые необходимы для подтверждения доставки исходящих сигналов до подчиненной станции и содержат результат выполнения операции. Значения сигналов доставки представлены в таблице.

Модуль позволяет использовать сегмент адресного пространства **Holding Registers** для передачи сигналов телеуправления и телесигнализации. Регистр адресного пространства протокола Modbus используется как набор в качестве 16 сигналов телеуправления или телесигнализации. При конфигурировании такого способа доставки сигналов в адресе сигнала указывается номер бита (BitPosition).

Так как элементы адресного пространства **Holding Registers** и **Coils** имеют тип доступа на чтение и запись данных, то имеется возможность отправлять сигналы телерегулирования (телеуправления) и получать сигналы телеизмерения (телесчет, телесигнализация) на один регистр памяти. Для этого необходимо создать два сигнала типов – входящий и исходящий. При этом не будет ошибки пересечения адресов. Запись и чтение с одного регистра применяется для проверки полученных данных подчиненной станцией и для проверки текущих значений параметров объекта телерегулирования.

Протокольный тип данных, указанный для сигнала, должен быть согласован с типом данных сигнала в соответствии с таблицей. Если типы данных будут не согласованы, то сигнал не будет принят на обслуживание модулем. Правило пользования таблицей: в первом столбце выбрать тип используемых данных и соотнести ему тип сигнала.

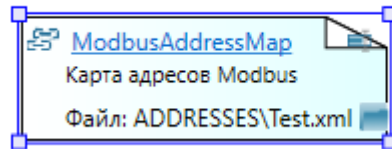
Протокольный тип	Тип в Astra.Server											
	int1	uint1	int2	uint2	int4	uint4	int8	uint8	float	double	string	bool
TS												да
TC												да
TM2			да	да*	да	да*	да	да*	да	да		
TMF4									да	да		
TMC					да	да*	да	да*				
TMF8										да		
TM2_TIME			да	да*	да	да*	да	да*				
TMF4_TIME									да	да		
TMC_TIME					да	да*	да	да*				
TMF8_TIME										да		
TR2			да	да*								
TRF4									да			
TR4					да	да*						
STR											да	
STR-COMMAND											да	

Где да\* - выполняется, если тип данных беззнаковый, иначе сигнал не принимается на обслуживание. Например, при преобразовании из протокольного типа данных TM2 в канонический тип данных uint2 необходимо в адресе сигнала указать "Signed=(False)".

## 1.1.2.3.2.4. Карта адресов

Для настройки сигналов модуля Опросчик Modbus RTU используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. В исполняемом приложении создайте сигналы необходимого типа;
2. Добавьте карту адресов Modbus в исполняемое приложение;



3. Задайте необходимые параметры для сигналов в карте адресов.

Тип	Привязка	Сегмент	Адрес	Номер бита	Номер записи в файле	Метка времени	Размер строки	Категория данных
int2	непосредственн	Input Registers	4			Нет		
uint2	непосредственн	Holding Registers	3			Нет		

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>› Непосредственно;</li><li>› Только чтение;</li><li>› Не привязан.</li></ul>
Сегмент	Указание, к какому сегменту памяти протокола Modbus привязан сигнал. Принимает одно из следующих значений: <ul style="list-style-type: none"><li>› Discrete Inputs (дискретные входы, один бит, только чтение);</li><li>› Coils (ячейки, один бит, чтение и запись);</li><li>› Input Registers (входные регистры, 16-битное слово, только чтение);</li><li>› Holding Registers (регистры хранения, 16-битное слово, чтение и запись).</li><li>› Files</li></ul>

Адрес	Адрес элемента данных в выбранной области памяти протокола. Значение в диапазоне от 0 до 65535
Номер бита	Номер бита в байте. Применяется с типом TS для Input Registers и Holding Registers и для TC в Holding Registers. Значения в диапазоне от 0 до 15
Номер записи в файле	Указывается при выборе сегмента Files, значение от 0 до 9999
Метка времени	Указание метки времени для протокольных типов TM2, TMF4: <ul style="list-style-type: none"> <li>&gt; Да;</li> <li>&gt; Нет.</li> </ul>
Размер строки	Размер строки в ASCII кодах. Применяется с типами STR и STR-COMMAND. Значения в диапазоне от 1 до 123
Категория данных	Категории данных нужны для группировки сигналов по интервалу опроса. Категории могут быть удобны, когда требуется оптимизировать нагрузку на сеть или сократить объем трафика.

При пересечении или совпадении адресов (например, при создании управляющего воздействия на один и тот же адрес) происходит запись в журнал модуля о наличии пересечений адресов, но к отказу в постановке сигналов на обслуживание не приводит. Проверка на совпадение и пересечение адресов происходит при старте модуля.

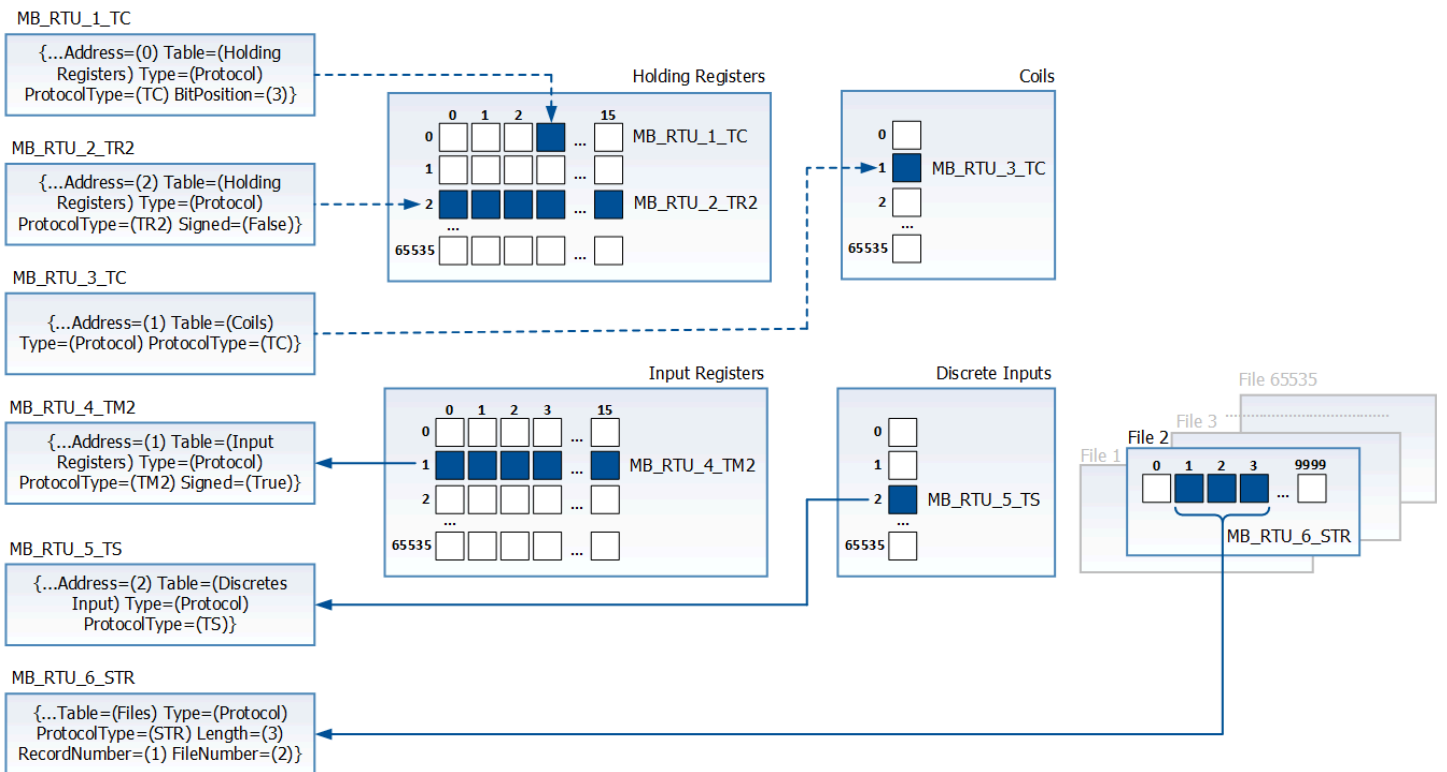


## 1.1.2.3.2.5. Конфигурирование сигналов

### Адрес сигнала

Параметры, включаемые в адрес сигнала для модуля Modbus RTU Master, представлены в таблице ниже.

Параметр	Значение
ModuleId	Идентификатор модуля
Protocol	ModbusRTU
Station	Номер станции
Type	Тип адреса сигнала. Принимает одно из двух значений: <ul style="list-style-type: none"><li>› Protocol - значение указывает, что тип относится к протоколу Modbus; тип указан в параметре ProtocolType;</li><li>› DeliveryStatus - сигнал доставки. Тип сигнала доставки указан в параметре DeliveryStatusType.</li></ul>
Table	Указание, к какому сегменту памяти протокола Modbus относится сигнал: <ul style="list-style-type: none"><li>› Discrete Inputs (дискретные входы, один бит, только чтение);</li><li>› Coils (ячейки, один бит, чтение и запись);</li><li>› Input Registers (входные регистры, 16-битное слово, только чтение);</li><li>› Holding Registers (регистры хранения, 16-битное слово, чтение и запись);</li><li>› Files (файлы).</li></ul>
Address	Адрес элемента данных в выбранной области памяти
ProtocolType	Протокольный тип данных по спецификации Modbus
Signed	Признак того что тип является знаковым. Принимает значения: True или False
BitPosition	Номер бита в байте. Применяется с типом TS для Input Registers. Значения в диапазоне от 0 до 15
Length	Размер строки в ASCII кодах. Применяется с типами TS и TS123
FileNumber	Номер файла. Диапазон значений от 1 до 65535



## 1.1.2.3.2.6. Диагностика работы модуля

### Журнал работы модуля

Модуль Modbus RTU Master ведёт журнал работы, в который записывается информация о работе модуля и обмене данными с подчиненными станциями.



Чтобы Modbus RTU Master вёл журнал работы, в общих параметрах модуля установите параметру Вести журнал работы модуля значение Да или установите сервисному сигналу модуля FrameLogEnable.Set значение true.

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

› в ОС Windows в папке:



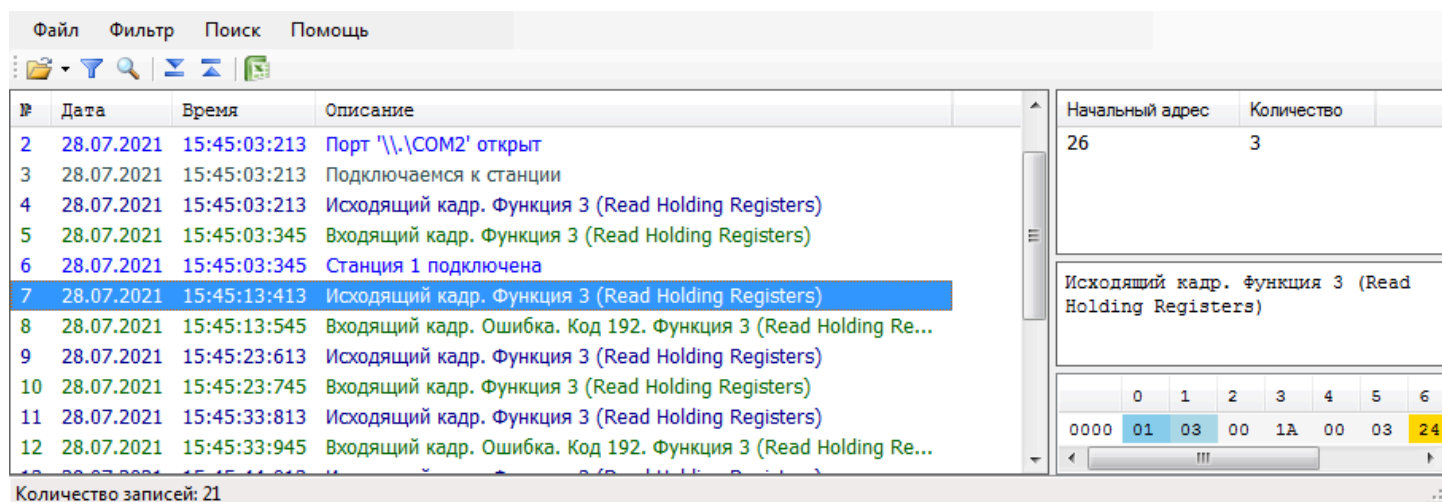
C:\Program Files\AstraRegul\Astra.Server\Logs;

› в Linux системах в директории:



/opt/AstraRegul/Astra.Server/Logs.

Для просмотра журнала работы модуля используется сервисное приложение Просмотрщик лога кадров.



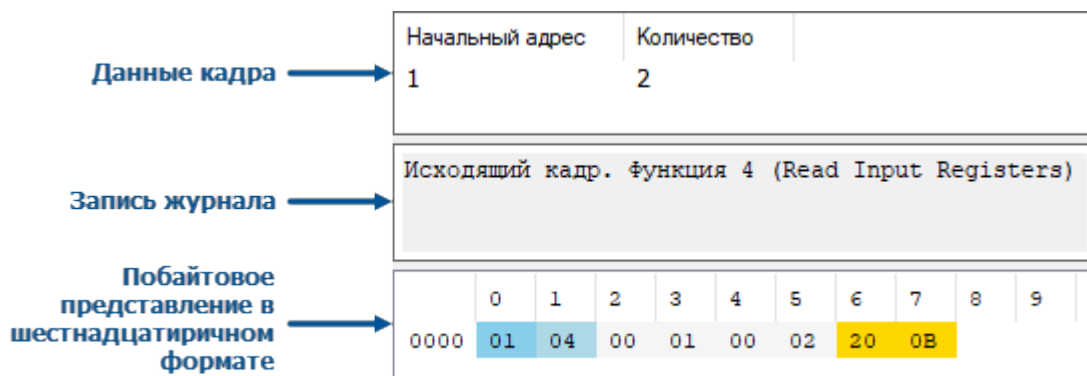
## Структура кадров модуля

Исходящий и входящий кадры модуля Modbus RTU Master имеют структуру:

Адрес	Код функции	Данные	Контрольная сумма
1 байт	1 байт	до 253 байт	2 байта

- Адрес – номер подчиненной станции в диапазоне от 1 до 255, с которой выполняется обмен данными.
- Код функции – функция записи или чтения данных сегмента памяти подчиненной станции.
- Данные (формат и длина поля зависит от кода функции):
  - в исходящем кадре – данные, необходимые для выполнения функции;
  - во входящем кадре – данные, предоставленные подчиненной станцией.
- Контрольная сумма – проверка отсутствия ошибок в кадре.

Побайтовое представление и данные кадров модуля отображаются в соответствующих полях окна сервисного приложения Просмотрщик лога кадров.



Индикация байт в поле побайтового представления кадра данных приведена в таблице:

Байт	Цвет	Описание
0	Голубой	Адрес подчиненной станции
1	Светло голубой	Код функции
от 2 до N	Светло-серый	Данные (формат и длина зависит от кода функции)
N+1, N+2	Желтый	Контрольная сумма

## Кадры функций чтения

Исходящий кадр для функций 01 (0x01) – 04 (0x04) имеет структуру:

Адрес	Код функции	Начальный адрес	Количество	Контрольная сумма
1 байт	1 байт	2 байта	2 байта	2 байта

- › Начальный адрес – адрес первого элемента сегмента памяти подчиненной станции, значение которого требуется прочитать.
- › Количество – количество считываемых элементов.

Входящий кадр для функций 01 (0x01) – 04 (0x04) имеет структуру:

Адрес	Код функции	Число байт	Данные	Контрольная сумма
1 байт	1 байт	1 байт	N байт	2 байта

- › Число байт – количество байт полученных данных.
- › Данные – полученные значения.

Исходящий кадр функции 20 (0x14) имеет структуру:

Адрес	Код функции	Число байт	Подзапрос				Контрольная сумма
			Тип ссылки	Номер файла	Номер записи	Длина записи	
1 байт	1 байт	1 байт	1 байт	2 байта	2 байта	2 байта	2 байта

- › Число байт – количество байт данных в подзапросе.
- › Подзапрос – группа для чтения, содержащая 7 байт данных для запроса:
  - › Тип ссылки – тип ссылки на данные, спецификацией определен код 0x06.
  - › Номер файла – номер файла расширенной памяти.
  - › Номер записи – адрес регистра внутри файла расширенной памяти.
  - › Длина записи – количество регистров для чтения.

Входящий кадр функции 20 (0x14) имеет структуру:

Адрес	Код функции	Число байт	Ответ			Контрольная сумма
			Число байт	Тип ссылки	Данные	
1 байт	1 байт	1 байт	1 байт	1 байт	N байт	2 байта

- › Число байт – общее количество байт полученных данных.
- › Ответ – группа полученных данных:
  - › Число байт – количество байт данных ответа.

› Тип ссылки – тип ссылки на данные, спецификацией определен код 0x06.

› Данные – полученные значения.

## Кадры функций записи

Исходящий кадр для функций 05 (0x05) и 06 (0x06) имеет структуру:

Адрес	Код функции	Начальный адрес	Значение	Контрольная сумма
1 байт	1 байт	2 байта	2 байта	2 байта

› Адрес элемента – адрес элемента сегмента памяти подчиненной станции, значение которого требуется изменить.

› Значение – устанавливаемое значение.

Если функция выполнена успешно, то входящий кадр содержит копию исходящего кадра.

Исходящий кадр функции 16 (0x10) имеет структуру:

Адрес	Код функции	Начальный адрес	Количество	Число байт	Значения	Контрольная сумма
1 байт	1 байт	2 байта	2 байта	1 байт	N байт	2 байта

› Начальный адрес – адрес первого элемента сегмента памяти подчиненной станции, значение которого требуется изменить.

› Количество – количество изменяемых элементов.

› Число байт – количество передаваемых байт устанавливаемых значений.

› Значения – устанавливаемые значения.

Если функция выполнена успешно, то входящий кадр функции 16 (0x10) имеет структуру:

Адрес	Код функции	Начальный адрес	Количество	Контрольная сумма
1 байт	1 байт	2 байта	N байт	2 байта

- › Начальный адрес – адрес первого измененного элемента.
- › Количество – количество измененных элементов.

## Кадры ошибок

Если подчиненная станция не может выполнить функцию, входящий кадр имеет структуру:

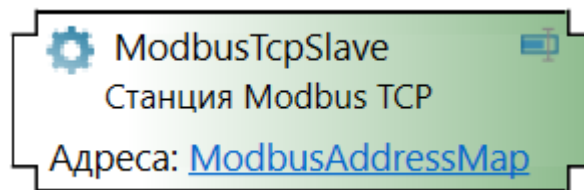
Адрес	Код функции + 0x80	Код ошибки	Контрольная сумма
1 байт	1 байт	1 байт	2 байта

В таблице приведены поддерживаемые модулем Modbus RTU Master коды ошибок.

Код	Название	
01	ILLEGAL FUNCTION (Недопустимая функция)	Принятый код функции не может быть
02	ILLEGAL DATA ADDRESS (Недопустимый адрес данных)	Адрес регистра подчиненной станции
03	ILLEGAL DATA VALUE (Недопустимое значение данных)	Значение, указанное в поле данных за
04	SERVER DEVICE FAILURE (Сбой подчиненной станции)	Подчиненная станция не может обраб
06	SERVER DEVICE BUSY (Подчиненная станция занята)	Подчиненная станция занята обработкой когда подчиненная станция освободит



### 1.1.2.3.3. Станция Modbus TCP



Модуль Modbus TCP Slave — коммуникационный модуль Astra.Server, предназначенный для передачи данных сервера по протоколу Modbus TCP.

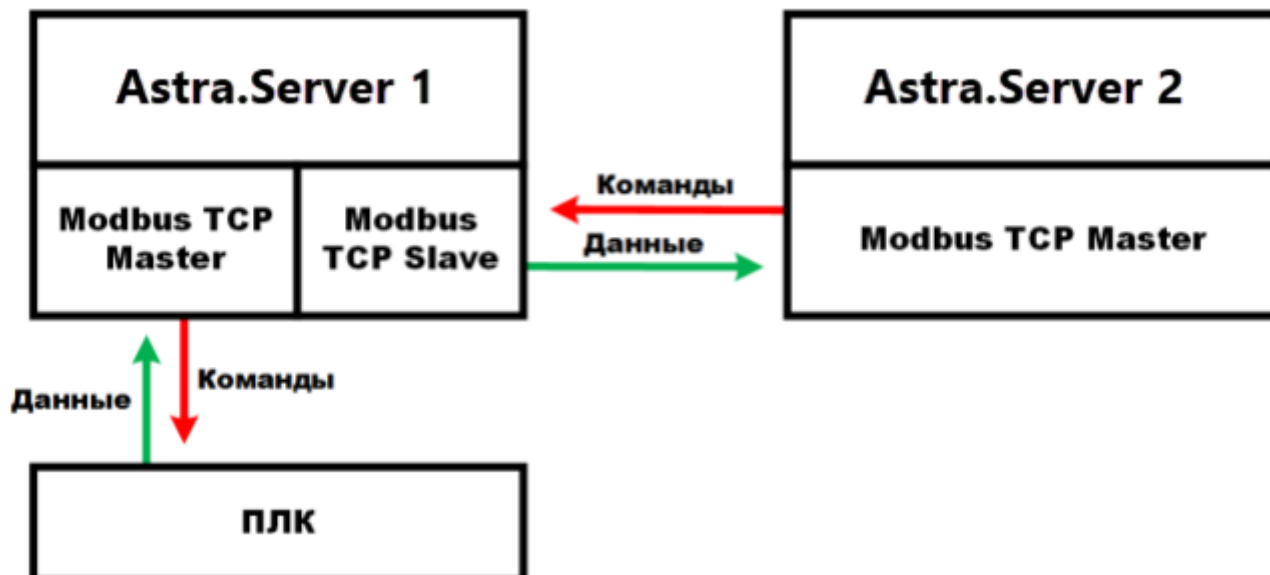
Модуль Modbus TCP Slave реализует функции подчиненной станции (или набора подчиненных станций) в соответствии со спецификацией [Modbus Application Protocol Specification](#).

Функции:

- › передача данных по запросу со стороны управления;
- › приём команд со стороны управления.

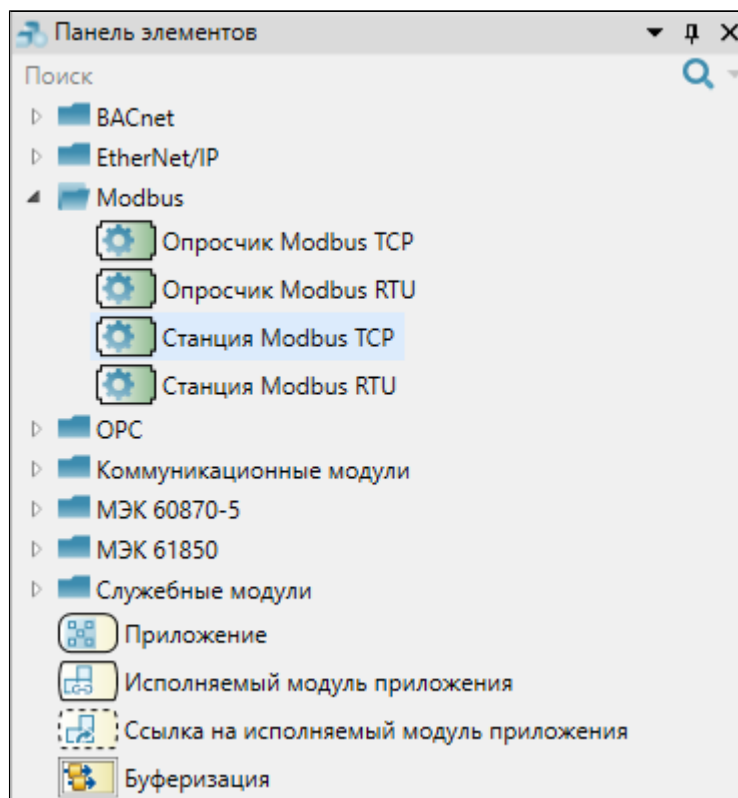
Модуль Modbus TCP Master отправляет команды и запрашивает данные у подчиненной станции (модуль Modbus TCP Slave), т.е. выполняет роль TCP клиента. Подчиненная станция (модуль Modbus TCP Slave) выполняет полученные команды или поставляет запрашиваемые данные, т.е. выполняет роль TCP сервера. Инициатором обмена данными всегда является модуль Modbus TCP Master.

Управляющие команды (ТУ, ТР) имеют больший приоритет перед командами опроса данных (ТС, ТИ, телесчет) и выполняются в первую очередь. Схема взаимодействия и потоков данных/управления между управляющей и подчиненной станцией показана на схеме ниже.



## 1.1.2.3.3.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

Свойства	
ModbusTcpSlave Станция Modbus TCP	
<b>Параметры модуля</b>	
Разрешено подключение с любых IP-адресов	Да
Список разрешенных IP-адресов	
Настройки по умолчанию	
Активность	Да
Отображаемое имя	
<b>Параметры станции</b>	
Номер станции	1
Номер TCP порта	502
Предельное количество одновременных запросов	5
Кодировка текста	windows-1251
<b>Общие</b>	
Карта адресов	Application.ModbusAddressMap
Имя	ModbusTcpSlave
<b>Изменение порядка байт</b>	
Байт в слове	
Слов в сигналах TMF4, TRF4, TMF8	
Слов в сигналах TMC, TR4, TM8, TR8	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения

## Общие


Параметр	Описание
Имя	Имя модуля
Карта адресов	Карта адресов Modbus, которая будет использована для данного адаптера

## Параметры модуля

Параметр	Описание
Разрешено подключение с любых IP-адресов	Значения: > Да > Нет
Список разрешенных IP-адресов	По умолчанию список пуст, что означает полный доступ к модулю с любого IP-адреса. Если нужно разрешить доступ только с некоторых IP-адресов, то перечислите их в этом параметре через точку с запятой или пробел
Настройки по умолчанию	Выбор файла с готовой конфигурацией
Активность	Активность модуля при запуске/перезапуске Astra.Server: > Да – модуль запущен; > Нет – модуль остановлен. Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

## Параметры станции

Параметр	Описание
Номер станции	Значение по умолчанию: 0
Номер TCP Порта	Порт, который будет прослушиваться модулем. По умолчанию: 502.
Список разрешенных IP-адресов	По умолчанию список пуст (полный доступ к модулю с любого IP-адреса). Если нужно разрешить доступ только с некоторых IP-адресов, то перечислите их в этом параметре через точку с запятой или пробел.

Предельное количество одновременных запросов	Максимальное число запросов с одного TCP-соединения, которые могут находиться в обработке.
Кодировка текста	<p>Выбор кодировки передаваемых текстовых данных:</p> <ul style="list-style-type: none"> <li>&gt; UTF-8</li> <li>&gt; windows-1251</li> </ul> <p>По умолчанию: windows-1251.</p> <div style="border: 1px solid #add8e6; padding: 10px; margin-top: 10px;"> <p> В UTF-8 символы Кириллицы и символы таблицы ASCII, выходящие за диапазон 0-127, занимают более 1 байта. Поэтому при настройке адреса сигнала для приёма/передачи строкового значения необходимо указывать достаточное количество регистров (параметр Размер строки), иначе строка может приниматься/передаваться не полностью.</p> </div>

## Изменение порядка байт

Параметр	Описание
Байт в слове	<p>Включает/отключает изменение порядка следования байтов в регистре. В случае если флаг установлен, то байты регистра меняются местами. Действие флага распространяется на все протокольные типы сигналов, кроме TS и TC</p> <p>Значения:</p> <ul style="list-style-type: none"> <li>&gt; Да</li> <li>&gt; Нет</li> </ul>
Слов в сигналах TMF4, TRF4, TMF8	<p>Включает/отключает изменение порядка следования слов в сигналах TMC, TR4. В случае если флаг установлен, то слова данных меняются местами. Если одновременно установлены флаги</p>

	<p>Слов в сигналах TMC, TR4 и Байт в слове, то слова данных меняются местами и байты в словах меняются местами.</p> <p>Значения:</p> <ul style="list-style-type: none"> <li>➤ Да</li> <li>➤ Нет</li> </ul>
Слов в сигналах TMC, TR4	<p>Включает/выключает изменение порядка слов в сигналах TMF4, TRF4. В случае если флаг установлен, изменение порядка следования слов происходит аналогично изменению порядка следования слов в сигналах TMC, TR4. Если одновременно установлены флаги Слов в сигналах TMF4, TRF4 и Байт в слове, то пары байтов меняются местами и в каждой паре байты меняются местами.</p> <p>Значения:</p> <ul style="list-style-type: none"> <li>➤ Да</li> <li>➤ Нет</li> </ul>

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.3.3.2. Типы данных

Для работы с данными в рамках протокола Modbus реализованы протокольные типы данных. В таблице ниже приведены протокольные типы и показано направление передачи для каждого из них.

Протокольный тип	Направление передачи	Описание
TS	Slave → Master	Телесигнализация
TC	Master → Slave	Телеуправление
TM2	Slave → Master	Телеизмерение
TMF4	Slave → Master	Телеизмерение
TMC	Slave → Master	Телесчет
TM2_TIME	Slave → Master	Телеизмерение с меткой времени
TMF4_TIME	Slave → Master	Телеизмерение с меткой времени
TMC_TIME	Slave → Master	Телесчет с меткой времени
TR2	Master → Slave	Телерегулирование
TRF4	Master → Slave	Телерегулирование
TR4	Master → Slave	Телерегулирование
STR	Slave → Master	Телеизмерение
STR-COMMAND	Master → Slave	Телерегулирование



В столбце Направление передачи показано, в каких случаях применяется тот или иной протокольный тип. Направление Master → Slave означает, что протокольный тип используется для отправки команд от управляющей станции. Направление Slave → Master —



данные передаются от подчиненной станции, в ответ на запрос от управляющей станции.

Адресное пространство протокола Modbus представляет собой 4 сегмента памяти (таблицы данных):

- › Discrete Inputs — дискретные входы. Размер: одиночный бит. Элементы доступны только на чтение;
- › Coils — ячейки. Размер: одиночный бит. Элементы доступны на чтение и запись;
- › Input Registers — входные регистры. Размер: 16-битное слово. Элементы доступны только на чтение;
- › Holding Registers — регистры хранения. Размер: 16-битное слово. Элементы доступны на чтение и запись.

Доступ к элементам каждого сегмента осуществляется с помощью 16-битной адресации. При такой адресации, каждый из четырех сегментов может вмещать до 65536 элементов (адресов).

Чтобы выбрать протокольный тип для работы с определенным сегментом памяти воспользуйтесь таблицей ниже.

Сегмент	Допустимые протокольные типы
Discrete Inputs	TS
Coils	TS, TC
Input Registers	TS, TM2, TMF4, TMC, TM2_TIME, TMF4_TIME, TMC_TIME, STR
Holding Registers	TS, TC, TM2, TMF4, TMC, TMF8, TM2_TIME, TMF4_TIME, TMC_TIME, TMF8_TIME, TR2, TRF4, TR4, STR, STR-COMMAND

Ставя сигналы на обслуживание модулю Modbus TCP Slave, необходимо учитывать совместимость между протокольными типами и встроенными типами Astra.Server. Таблица совместимости показана ниже.

Протокольный тип	Тип в Astra.Server											
	int1	uint1	int2	uint2	int4	uint4	int8	uint8	float	double	string	bool
TS												да
TC												да
TM2			да	да*	да	да*	да	да*	да	да		
TMF4									да	да		
TMC					да	да*	да	да*				
TM2_TIME			да	да*	да	да*	да	да*				
TMF4_TIME									да	да		
TMC_TIME					да	да*	да	да*				
TR2			да	да*								
TRF4									да			
TR4					да	да*						
STR											да	
STR-COMMAND											да	

Где да\* — выполняется, если тип данных беззнаковый, иначе сигнал не принимается на обслуживание.

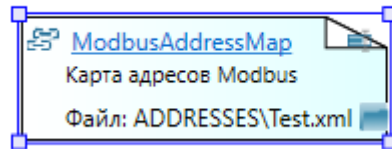


Если необходим мониторинг на стороне подчиненной станции однобитовых управляющих команд (телеуправление, протокольный тип TC), то в дереве сигналов следует создать сигнал типа bool, который будет носителем информации о поступающих командах.

## 1.1.2.3.3. Карта адресов

Для настройки сигналов модуля Станция Modbus TCP используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. В исполняемом приложении создайте сигналы необходимого типа;
2. Добавьте карту адресов Modbus в исполняемое приложение;



3. Задайте необходимые параметры для сигналов в карте адресов.

Тип	Привязка	Сегмент	Адрес	Номер бита	Номер записи в файле	Метка времени	Размер строки	Категория данных
int2	непосредственн	Input Registers	4			Нет		
uint2	непосредственн	Holding Registers	3			Нет		

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>› Непосредственно;</li><li>› Только чтение;</li><li>› Не привязан.</li></ul>
Сегмент	Указание, к какому сегменту памяти протокола Modbus привязан сигнал. Принимает одно из следующих значений: <ul style="list-style-type: none"><li>› Discrete Inputs (дискретные входы, один бит, только чтение);</li><li>› Coils (ячейки, один бит, чтение и запись);</li><li>› Input Registers (входные регистры, 16-битное слово, только чтение);</li><li>› Holding Registers (регистры хранения, 16-битное слово, чтение и запись).</li></ul>

	<p>› Files</p>
Адрес	Адрес элемента данных в выбранной области памяти протокола. Значение в диапазоне от 0 до 65535
Номер бита	Номер бита в байте. Применяется с типом TS для Input Registers и Holding Registers и для TC в Holding Registers. Значения в диапазоне от 0 до 15
Номер записи в файле	Указывается при выборе сегмента Files, значение от 0 до 9999
Метка времени	<p>Указание метки времени для протокольных типов TM2, TMF4:</p> <ul style="list-style-type: none"> <li>› Да;</li> <li>› Нет.</li> </ul>
Размер строки	Размер строки в ASCII кодах. Применяется с типами STR и STR-COMMAND. Значения в диапазоне от 1 до 123
Категория данных	Категории данных нужны для группировки сигналов по интервалу опроса. Категории могут быть удобны, когда требуется оптимизировать нагрузку на сеть или сократить объем трафика.

## 1.1.2.3.3.4. Настройка сигналов

Для того, чтобы подчиненные станции (реализуемые в рамках модуля Modbus TCP Slave) взаимодействовали с управляющей стороной (принимали команды и передавали данные по опросу управляющей стороны) нужно:

- создать сигналы подходящего типа, которые будут являться носителями информации для обмена между Modbus TCP Slave и управляющей станцией;
- созданные сигналы поставить на обслуживание предварительно настроенному модулю Modbus TCP Slave;
- для каждого из созданных сигналов настроить адрес в свойстве 5000.

## 1.1.2.3.3.5. Диагностика работы модуля

Для более детального анализа работы модуля Modbus TCP Slave воспользуйтесь сервисным приложением Просмотрщик лога кадров. Сервисные приложения включены в клиентскую часть дистрибутива Astra.Server.

С помощью Просмотрщика лога кадров можно просмотреть содержимое входящих и исходящих кадров каждой подчиненной станции.

№	Дата	Время	Описание	Станция	Т.	Адрес	Hex	Знаковое	Беззнаковое
26980	04.07.2023	17:14:39:375	Входящий кадр Функция 3 (Read Holding Registers)	1	3..	0	0	0	0
26981	04.07.2023	17:14:39:375	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				
26982	04.07.2023	17:14:39:386	Входящий кадр Функция 3 (Read Holding Registers)	1	3..				
26983	04.07.2023	17:14:39:386	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				
26984	04.07.2023	17:14:39:397	Входящий кадр Функция 3 (Read Holding Registers)	1	3..				
26985	04.07.2023	17:14:39:397	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				
26986	04.07.2023	17:14:39:428	Входящий кадр Функция 3 (Read Holding Registers)	1	3..				
26987	04.07.2023	17:14:39:428	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				
26988	04.07.2023	17:14:39:460	Входящий кадр Функция 3 (Read Holding Registers)	1	3..				
26989	04.07.2023	17:14:39:460	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				
26990	04.07.2023	17:14:39:491	Входящий кадр Функция 3 (Read Holding Registers)	1	3..				
26991	04.07.2023	17:14:39:491	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				

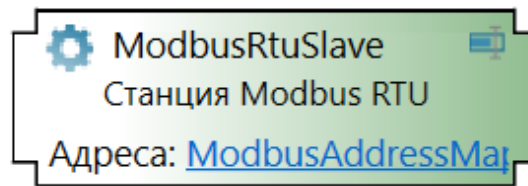
  

Адрес	Hex	Знаковое	Беззнаковое
0	0	0	0

Исходящий кадр функция 3 (Read Holding Registers)									
	0	1	2	3	4	5	6	7	8
0000	93	76	00	00	00	05	01	03	02
0001	00								

## 1.1.2.3.4. Станция Modbus RTU



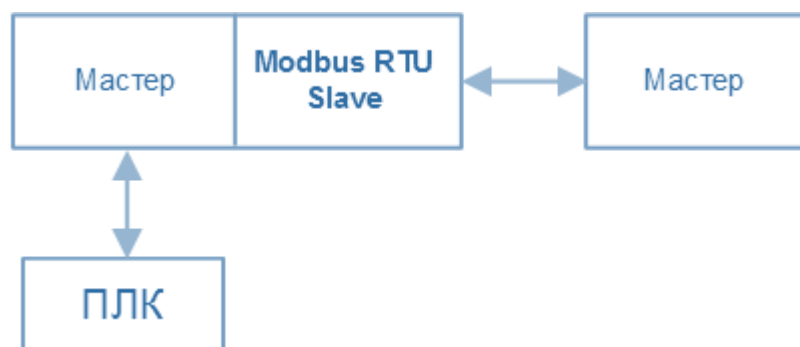
Модуль Modbus RTU Slave предназначен для взаимодействия с удаленным опрашивающим устройством (далее опросчиком, управляющей станцией, мастером) в соответствии с протоколом Modbus over Serial Line. Основными функциями модуля являются передача опросчику данных, полученных с подчиненной станции, и прием команд от управляющей станции.



Модуль Modbus RTU Slave в составе Astra.Server функционирует на компьютере только под управлением ОС Windows.

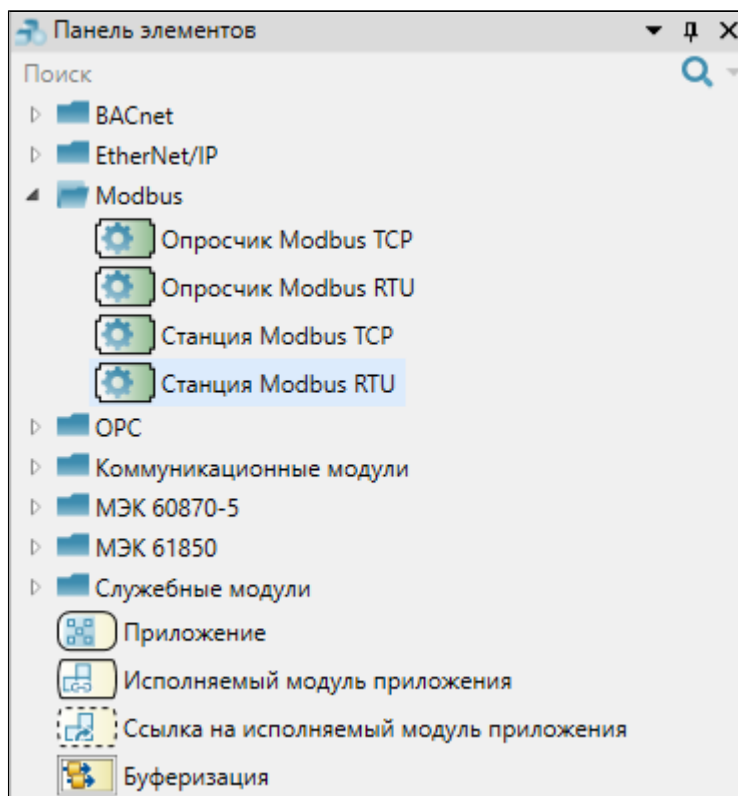
Модуль работает в качестве подчиненной станции для опросчика (к примеру, модуля Modbus RTU Master). Отправка данных опросчику по сигналам телеизмерения, телесигнализации, телесчета выполняется после получения запроса по этим сигналам. Команды телеуправления и телерегулирования имеют больший приоритет, чем команды опроса и выполняются вне очереди.

Схема передачи данных при взаимодействии модуля с опросчиком и подчиненной станцией показана на рисунке ниже.



## 1.1.2.3.4.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.




Свойства	
ModbusRtuSlave Станция Modbus RTU	
^ Параметры станции	
Номер станции	1
Кодировка текста	windows-1251
^ Параметры модуля	
Настройки по умолчанию	
Задержка перед отправкой ответа, мс	0
Активность	Да
Отображаемое имя	
^ Общие	
Карта адресов	Application.ModbusAddressMap
Имя	ModbusRtuSlave
^ Изменение порядка байт	
Байт в слове	
Слов в сигналах TMF4, TRF4, TMF8	
Слов в сигналах TMC, TR4, TM8, TR8	
^ Параметры журналирования	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения

## Общие

Параметр	Описание
Имя	Имя модуля
Карта адресов	Карта адресов Modbus, которая будет использована для данного адаптера

## Параметры станции

Параметр	Описание
----------	----------

Номер станции	Значение по умолчанию: 1
Кодировка текста	<p>Выбор кодировки передаваемых текстовых данных. Значение по умолчанию: windows-1251.</p> <div style="border: 1px solid #add8e6; padding: 10px; margin: 10px 0;">  В UTF-8 символы Кириллицы и символы таблицы ASCII, выходящие за диапазон 0-127, занимают более 1 байта. Поэтому при настройке адреса сигнала для приёма/передачи строкового значения необходимо указывать достаточное количество регистров (параметр Размер строки), иначе строка может приниматься/передаваться не полностью. </div> <p>Значения:</p> <ul style="list-style-type: none"> <li>&gt; UTF-8</li> <li>&gt; windows-1251</li> </ul>

## Параметры модуля

Параметр	Описание
Настройки по умолчанию	Выбор файла с готовой конфигурацией
Задержка перед отправкой ответа, мс	Время задержки в мс отправки сигнала
Активность	<p>Активность модуля при запуске/перезапуске Astra.Server:</p> <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен;</li> <li>&gt; Нет – модуль остановлен.</li> </ul> <p>Управляется служебным сигналом Active.Set</p>
Отображаемое имя	Отображаемое имя модуля

## Изменение порядка байт

Параметр	Описание
Байт в слове	<p>Включает/отключает изменение порядка следования байтов в регистре. В случае если флаг установлен, то байты регистра меняются местами. Действие флага распространяется на все протокольные типы сигналов, кроме TS и TC</p> <p>Значения:</p> <ul style="list-style-type: none"> <li>➤ Да</li> <li>➤ Нет</li> </ul>
Слов в сигналах TMF4, TRF4, TMF8	<p>Включает/отключает изменение порядка следования слов в сигналах TMC, TR4. В случае если флаг установлен, то слова данных меняются местами. Если одновременно установлены флаги Слов в сигналах TMC, TR4 и Байт в слове, то слова данных меняются местами и байты в словах меняются местами.</p> <p>Значения:</p> <ul style="list-style-type: none"> <li>➤ Да</li> <li>➤ Нет</li> </ul>
Слов в сигналах TMC, TR4	<p>Включает/выключает изменение порядка слов в сигналах TMF4, TRF4. В случае если флаг установлен, изменение порядка следования слов происходит аналогично изменению порядка следования слов в сигналах TMC, TR4. Если одновременно установлены флаги Слов в сигналах TMF4, TRF4 и Байт в слове, то пары байтов меняются местами и в каждой паре байты меняются местами.</p> <p>Значения:</p> <ul style="list-style-type: none"> <li>➤ Да</li> <li>➤ Нет</li> </ul>

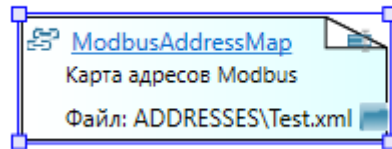
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.3.4.2. Карта адресов

Для настройки сигналов модуля Станция Modbus RTU используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. В исполняемом приложении создайте сигналы необходимого типа;
2. Добавьте карту адресов Modbus в исполняемое приложение;



3. Задайте необходимые параметры для сигналов в карте адресов.

Тип	Привязка	Сегмент	Адрес	Номер бита	Номер записи в файле	Метка времени	Размер строки	Категория данных
int2	непосредственн	Input Registers	4			Нет		
uint2	непосредственн	Holding Registers	3			Нет		

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>› Непосредственно;</li><li>› Только чтение;</li><li>› Не привязан.</li></ul>
Сегмент	Указание, к какому сегменту памяти протокола Modbus привязан сигнал. Принимает одно из следующих значений: <ul style="list-style-type: none"><li>› Discrete Inputs (дискретные входы, один бит, только чтение);</li><li>› Coils (ячейки, один бит, чтение и запись);</li><li>› Input Registers (входные регистры, 16-битное слово, только чтение);</li><li>› Holding Registers (регистры хранения, 16-битное слово, чтение и запись).</li></ul>

	<p>› Files</p>
Адрес	Адрес элемента данных в выбранной области памяти протокола. Значение в диапазоне от 0 до 65535
Номер бита	Номер бита в байте. Применяется с типом TS для Input Registers и Holding Registers и для TC в Holding Registers. Значения в диапазоне от 0 до 15
Номер записи в файле	Указывается при выборе сегмента Files, значение от 0 до 9999
Метка времени	<p>Указание метки времени для протокольных типов TM2,TMF4:</p> <ul style="list-style-type: none"> <li>› Да;</li> <li>› Нет.</li> </ul>
Размер строки	Размер строки в ASCII кодах. Применяется с типами STR и STR-COMMAND. Значения в диапазоне от 1 до 123
Категория данных	Категории данных нужны для группировки сигналов по интервалу опроса. Категории могут быть удобны, когда требуется оптимизировать нагрузку на сеть или сократить объем трафика.

## 1.1.2.3.4.3. Настройка сигналов

### Адрес сигнала

Параметры, включаемые в адрес сигнала для модуля Modbus RTU Slave, представлены в таблице ниже.

Параметр	Значение
ModuleId	Идентификатор модуля
Protocol	ModbusRTU
Station	Номер станции
Address	Адрес элемента данных в выбранной области памяти протокола
Table	Указание, к какому сегменту памяти протокола Modbus привязаны значения: <ul style="list-style-type: none"><li>› Discrete Inputs (дискретные входы, один бит, только чтение)</li><li>› Coils (ячейки, один бит, чтение и запись);</li><li>› Input Registers (входные регистры, 16-битное слово, только чтение)</li><li>› Holding Registers (регистры хранения, 16-битное слово, чтение и запись)</li></ul>
Type	Protocol — значение указывает, что тип относится к типу, переданному в параметре ProtocolType
ProtocolType	Протокольный тип данных по спецификации Modbus
Signed	Признак того что тип является знаковым. Применяется с типами TS и TR
BitPosition	Номер бита в байте. Применяется с типом TS для Input Registers Значения в диапазоне от 0 до 15
Length	Размер строки в ASCII кодах. Применяется с типами STR и STR-CC

Пример адреса сигнала для отправки данных в ответ на команду телеизмерения.



```
{ModuleId=(Modbus RTU Slave 1) Protocol=(ModbusRTU)  
Station=(1) Address=(24) Table=(Holding Registers) Type=(Protocol)  
ProtocolType=(TM2) Signed=(True)}
```



При некорректном составлении адреса сигнала в журнал приложений выводится соответствующее сообщение.



## 1.1.2.3.4.4. Типы данных

Для работы с данными в рамках протокола Modbus реализованы протокольные типы данных. В таблице ниже приведены протокольные типы и показано направление передачи.

Протокольный тип	Направление передачи
TS	Slave → Master
TC	Master → Slave
TM2	Slave → Master
TMF4	Slave → Master
TMC	Slave → Master
TM2_TIME	Slave → Master
TMF4_TIME	Slave → Master
TMC_TIME	Slave → Master
TR2	Master → Slave
TRF4	Master → Slave
TR4	Master → Slave
STR	Slave → Master
STR-COMMAND	Master → Slave



В столбце Направление передачи показано, в каких случаях применяется тот или иной протокольный тип. Направление Master → Slave означает, что протокольный тип используется для отправки команд от управляющей станции. Направление Slave → Master — данные передаются от подчиненной станции, в ответ на запрос от управляющей станции.

Адресное пространство протокола Modbus представляет собой 4 сегмента памяти (таблицы данных):

- › Discrete Inputs — дискретные входы. Размер: одиночный бит. Элементы доступны только на чтение;
- › Coils — ячейки. Размер: одиночный бит. Элементы доступны на чтение и запись;
- › Input Registers — входные регистры. Размер: 16-битное слово. Элементы доступны только на чтение;
- › Holding Registers — регистры хранения. Размер: 16-битное слово. Элементы доступны на чтение и запись.

Доступ к элементам каждого сегмента осуществляется с помощью 16-битной адресации. При такой адресации, каждый из четырех сегментов может вмещать до 65536 элементов (адресов).

Чтобы выбрать протокольный тип для работы с определенным сегментом памяти воспользуйтесь таблицей ниже.

Сегмент	Допустимые протоколы
Discrete Inputs	TS
Coils	TS, TC
Input Registers	TS, TM2, TMF4, TMC, TM2_TIME, TMF4_TIME, TMC_TIME, ST
Holding Registers	TS, TC, TM2, TMF4, TMC, TMF8, TM2_TIME, TMF4_TIME, TMC_COMMAND

Протокольный тип данных, указанный для сигнала, должен быть согласован с типом данных сигнала в соответствии с таблицей ниже. Если типы данных будут не согласованы, то сигнал не будет принят на обслуживание модулем. Правило пользования таблицей: в первом столбце выбрать тип используемых данных и соотнести ему тип сигнала.

Протокольный тип	Тип в Astra.Server											
	int1	uint1	int2	uint2	int4	uint4	int8	uint8	float	double	string	bool

TS												да
TC												да
TM2			да	да*	да	да*	да	да*	да	да		
TMF4									да	да		
TMC					да	да*	да	да*				
TM2_TIME			да	да*	да	да*	да	да*				
TMF4_TIME									да	да		
TMC_TIME					да	да*	да	да*				
TR2			да	да*								
TRF4									да			
TR4					да	да*						
STR												да
STR-COMMAND												да

Где да\* — выполняется, если тип данных беззнаковый, иначе сигнал не принимается на обслуживание.

## 1.1.2.3.4.5. Возвращаемые ошибки

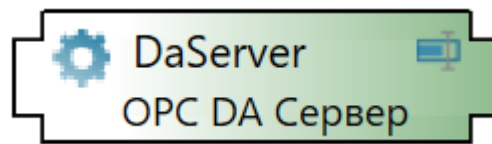
Модуль Modbus RTU Slave может передавать следующие строки в ответ на запрос опросчика:

- › SLAVE DEVICE BUSY — модуль занят обработкой другого запроса. В результате модуль получит повторный запрос данных;
- › ILLEGAL DATA ADDRESS — отсутствует адрес данных, переданный в запросе по сигналу.

## 1.1.2.4. OPC

Модуль	Описание
<a href="#">OPC DA Сервер</a>	Коммуникационный модуль для опроса данных по стандарту OPC DA
<a href="#">OPC DA Клиент</a>	Коммуникационный модуль для передачи данных по стандарту OPC DA
<a href="#">OPC HDA Сервер</a>	Коммуникационный модуль для опроса данных истории по стандарту OPC HDA
<a href="#">OPC HDA Клиент</a>	Коммуникационный модуль для передачи данных истории по стандарту OPC HDA
<a href="#">OPC AE Сервер</a>	Коммуникационный модуль для опроса данных различных событий по стандарту OPC AE
<a href="#">OPC UA Сервер</a>	Коммуникационный модуль для опроса данных по стандарту OPC UA
<a href="#">OPC UA Клиент</a>	Коммуникационный модуль для передачи данных по стандарту OPC UA

## 1.1.2.4.1. OPC DA Сервер



Модуль OPC DA Server предназначен для предоставления данных Astra.Server по спецификации OPC DA.

Функции:

- доступ к адресному пространству Astra.Server для сторонних OPC DA клиентов;
- предоставление данных о значениях сигналов и сигнальных свойств Astra.Server;
- возможность изменения значений сигналов и свойств сигналов в адресном пространстве Astra.Server.



Модуль OPC DA Server в составе Astra.Server функционирует на компьютере только под управлением ОС Windows.

Модуль OPC DA Server реализует серверную часть интерфейсов спецификации OPC DA. Модуль позволяет OPC DA клиентам подписываться на получение значений сигналов и свойств сигналов адресного пространства Astra.Server, а также изменять значения сигналов и свойств. Для подписки на получение сигналов Astra.Server используется тег сигнала.



Не рекомендуется создавать сигналы с именами, совпадающими с короткими именами свойств, т.к. при подписке сигналы имеют более высокий приоритет, чем свойства сигналов.

Модуль OPC DA Server реализует серверную часть интерфейсов спецификации OPC DA. Модуль позволяет OPC DA клиентам подписываться на получение

значений сигналов и свойств сигналов адресного пространства Astra.Server, а также изменять значения сигналов и свойств.

Если в сервере изменилось значение сигнала или свойства, то модуль OPC DA Server уведомляет подписанных OPC DA клиентов о новом значении. Уведомления, обрабатываемые модулем, выстраиваются в очередь для рассылки клиентам, размер которой задаётся в настройках модуля. Время, через которое модуль рассылает уведомление о новом значении, задаётся в настройках OPC DA клиента. Если за указанный интервал времени значение сигнала в сервере изменилось и вернулось к исходному значению, которое уже было передано клиенту, уведомление OPC DA клиенту не отсылается.



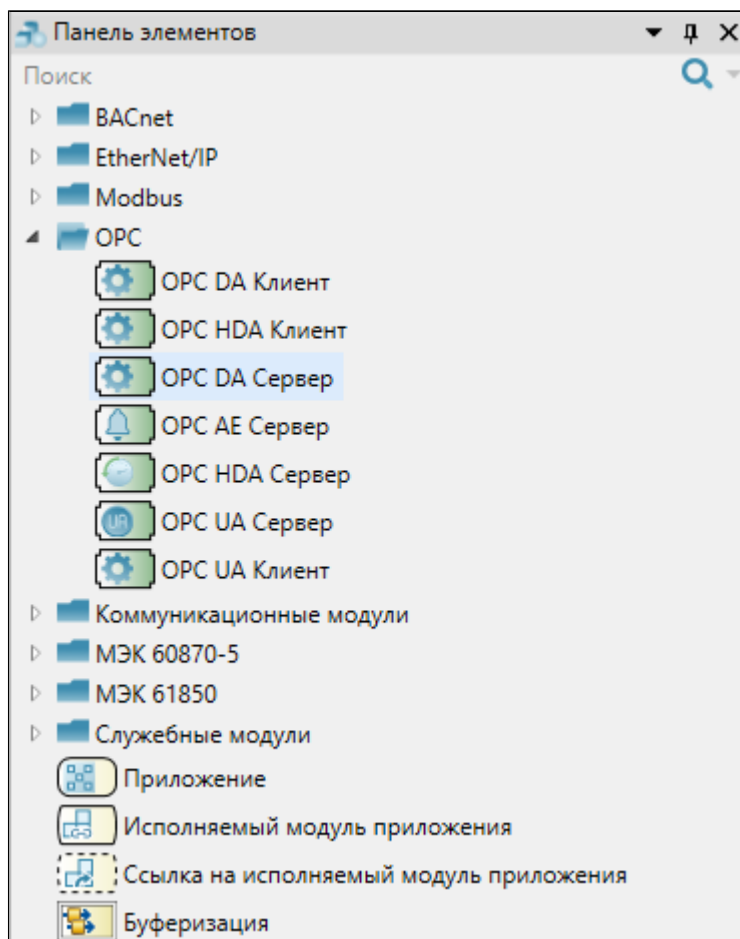
Конфигурация Astra.Server может содержать только один экземпляр модуля OPC DA Server. Программный идентификатор (ProgID) OPC DA сервера по умолчанию: Astra.OPCDAServer.



Если модуль OPC DA Server неактивен или отсутствует в конфигурации Astra.Server, то к серверу невозможно подключиться OPC DA клиентом.

# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.



Свойства	
DaServer OPC DA Сервер	
Общие	
Идентификатор сервера	AP.OPCDAServer
Имя	DaServer
Параметры модуля	
Максимальный размер очереди уведомлений	500000
Активность	Да
Отображаемое имя	
Параметры журналирования	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения

## Общие

Параметр	Описание
Имя	Имя модуля
Идентификатор сервера	Используется для подключения к серверу

## Параметры модуля

Параметр	Описание
Максимальный размер очереди уведомлений	Размер очереди уведомлений, при достижении которого новые уведомления отбрасываются. Если включена запись в журнал работы модуля, то выдается сообщение в журнал и сообщения об отброшенных уведомлениях сохраняются с пометкой Отброшено по причине переполнения очереди.
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен;</li> <li>&gt; Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом Active.Set

Отображаемое имя	Отображаемое имя модуля
------------------	-------------------------

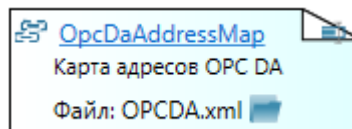
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.




## 1.1.2.4.1.2. Карта адресов

Для настройки сигналов модуля OPC DA Сервер используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов OPC DA в исполняемое приложение;



2. Откройте редактор карты адресов.

	Сигнал	Тип	Привязка	Тег в сервере	Номер бита	Позиция в мас	Категория дан
	▼	▼	▼	▼	▼	▼	▼
	uint4	uint4	непосредственн	1	2	3 4	
	int4	int4	непосредственн	1	2	3 4	

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>› Непосредственно;</li><li>› Только чтение;</li><li>› Не привязан.</li></ul>
Тег в сервере	Тег в сервере — адрес сигнала или свойства в адресном пространстве OPC DA сервера, к которому подключается модуль. Чтобы настроить сигнал на получение значения свойства сигнала со стороннего DA сервера, в параметре укажите через точку имя свойства сигнала
Номер бита	Для входящих сигналов типа bool можно указать Бит, чтобы при приеме целочисленного значения в сигнал записывалось значение указанного бита. Разрешенные значения [0, 63]

Позиция в массиве	Для входящих сигналов любого типа можно указать параметр Позиция, чтобы при приеме массива значений данного типа в сигнал записывалось значение из указанной позиции в массиве. Значения в массиве считаются с нуля
Категория данных	Отнесение сигнала к категориям данных

# Диагностика работы модуля

## Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

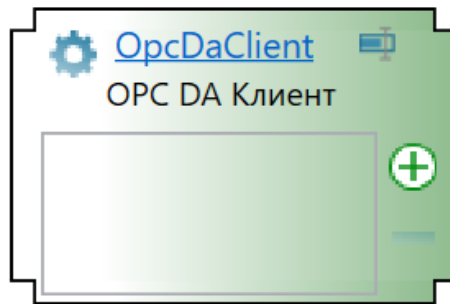
Для анализа ошибок, возникающих в процессе работы модуля, воспользуйтесь сервисным приложением Просмотрщик лога кадров.

№	Дата	Время	Описание	Object
10882	19.06.2021	15:38:15:185	Удалена группа ServiceStateServerSet_...	
10885	19.06.2021	15:38:15:185	Отключился клиент - "Unknown Client" ...	
10887	19.06.2021	15:38:15:185	<-- COPCServer::~~COPCServer	0x000000000261B3E0
10894	19.06.2021	15:38:15:226	Разрегистрирована фабрика классов	
10910	19.06.2021	15:38:15:226	Удалена группа Group 0 [Client Name - ...	
10917	19.06.2021	15:38:15:226	Удалена группа ServiceStateServerSet_...	
10920	19.06.2021	15:38:15:226	Отключился клиент - "Unknown Client" ...	
10933	19.06.2021	15:38:15:226	Удалена группа Group 0 [Client Name - ...	
10936	19.06.2021	15:38:15:226	Отключился клиент - "OPCtools" (ID - 6)	
10944	19.06.2021	15:38:15:226	Журнал закрыт	
10945	19.06.2021	15:38:16:157	Журнал открыт для записи	
10955	19.06.2021	15:38:16:167	Зарегистрирована фабрика классов	
10961	19.06.2021	15:38:16:297	Подключился клиент - "Unknown Client..."	

Зарегистрирована фабрика классов

0 1 2

## 1.1.2.4.2. OPC DA Клиент



Модуль OPC DA Клиент предназначен для сбора данных с серверов по спецификации OPC DA.

Модуль OPC DA Клиент предоставляет следующие функциональные возможности:

- › чтение значений сигналов различных OPC DA серверов;
- › запись значений сигналов в различные OPC DA серверы.



Модуль OPC DA Клиент в составе Astra.Server функционирует на компьютере только под управлением ОС Windows.

Модуль OPC DA реализует клиентскую часть интерфейсов спецификации OPC DA. В составе Astra.Server может присутствовать несколько модулей OPC DA Клиента.

Модуль собирает данные со сторонних OPC DA серверов по спецификации OPC DA и передает в Astra.Server.



Если Astra.Server переходит в состояние РЕЗЕРВ, то модуль OPC DA Клиент перестает записывать значения сигналов и сигнальных свойств в сторонние серверы и начинает работать только на чтение данных.

Принцип формирования значений сигнала (VQT) модуля OPC DA Client:

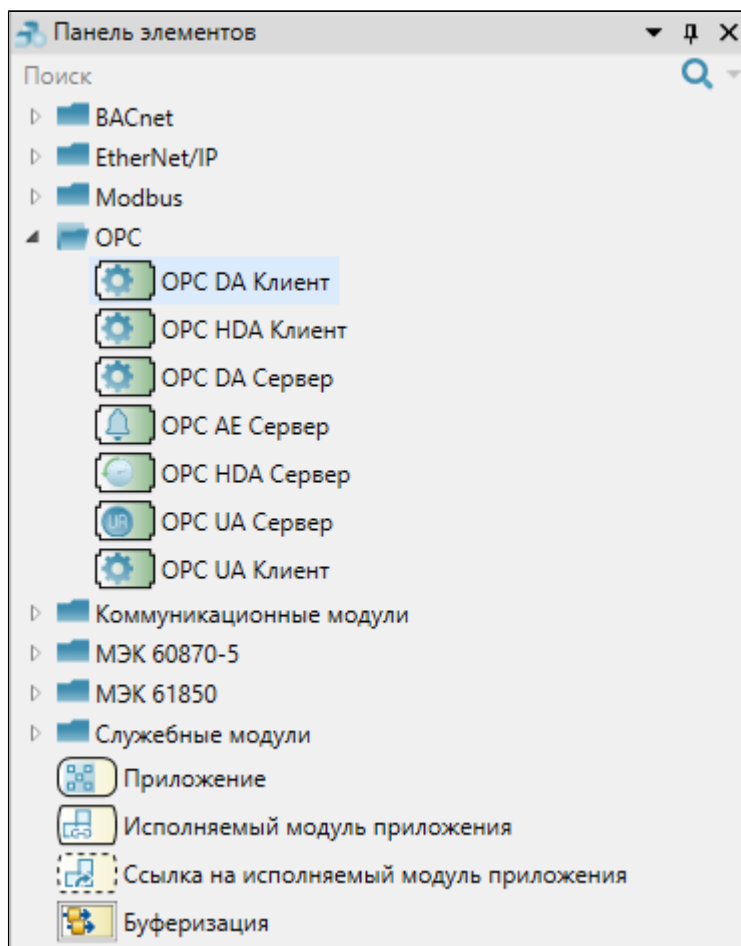
- › Значение (Value) — принятые значения сигналов от серверов записываются в сигнал;
- › Качество (Quality);
- › Метка времени (Timestamp) — метка времени сигнала устанавливается равной метке времени сервера, с которого пришло значение сигнала.

Принцип работы модуля с серверами OPC DA, которые находятся в резервной паре:

- › данные принимаются только от сервера, находящегося в режиме РАБОТА;
- › в случае потери связи с основным сервером модуль переводит режим работы резервного сервера в РАБОТА и начинает принимать данные от него.

# Настройка модуля

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.



Свойства	
ОpcDaClient OPC DA Клиент	
⤴ Параметры модуля	
Настройки по умолчанию	
Активность	Да
Отображаемое имя	
⤴ Параметры журналирования	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
⤴ Общие	
Имя	ОpcDaClient

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Настройки по умолчанию	Выбор файла с готовой конфигурацией
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>➤ Да – модуль запущен;</li> <li>➤ Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

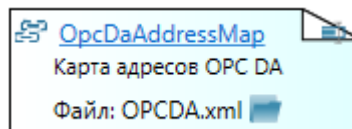
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.




## 1.1.2.4.2.2. Карта адресов

Для настройки сигналов модуля OPC DA Клиент используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов OPC DA в исполняемое приложение;



2. Откройте редактор карты адресов.

	Сигнал	Тип	Привязка	Тег в сервере	Номер бита	Позиция в мас	Категория дан
	▼	▼	▼	▼	▼	▼	▼
	uint4	uint4	непосредственн	1	2	3 4	
	int4	int4	непосредственн	1	2	3 4	

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>› Непосредственно;</li><li>› Только чтение;</li><li>› Не привязан.</li></ul>
Тег в сервере	Тег в сервере — адрес сигнала или свойства в адресном пространстве OPC DA сервера, к которому подключается модуль. Чтобы настроить сигнал на получение значения свойства сигнала со стороннего DA сервера, в параметре укажите через точку имя свойства сигнала
Номер бита	Для входящих сигналов типа bool можно указать Бит, чтобы при приеме целочисленного значения в сигнал записывалось значение указанного бита. Разрешенные значения [0, 63]

Позиция в массиве	Для входящих сигналов любого типа можно указать параметр Позиция, чтобы при приеме массива значений данного типа в сигнал записывалось значение из указанной позиции в массиве. Значения в массиве считаются с нуля
Категория данных	Отнесение сигнала к категориям данных

## 1.1.2.4.2.3. Качество сигналов

Установка модулем значения качества обслуживаемых сигналов происходит по алгоритму определения значения качества.

При старте модуля OPC DA Client модуль устанавливает сигналам одно из следующих значений качества:

- › (4) OPC\_QUALITY\_CONFIG\_ERROR – ошибка конфигурации сигнала;
- › (8) OPC\_QUALITY\_NOT\_CONNECTED – сигнал принят на обслуживание, но соединение с сервером еще не установлено;
- › (24) OPC\_QUALITY\_COMM\_FAILURE – соединение с сервером установить не удалось.

При установке соединения с OPC DA сервером модуль устанавливает сигналам одно из следующих значений качества:

- › (64) OPC\_QUALITY\_UNCERTAIN – связь с сервером установлена, но данные еще не получены;
- › (4) OPC\_QUALITY\_CONFIG\_ERROR – не удалось подписаться на сигнал.

При получении данных с OPC DA сервера модуль устанавливает сигналам одно из следующих значений качества:

- › (96) CONVERT\_ERROR – полученное модулем значение не удалось преобразовать к типу сигнала в Astra.Server.

Значение качества, полученное с OPC DA сервера:

- › (192) OPC\_QUALITY\_GOOD – значение получено с хорошим качеством;
- › (216) OPC\_QUALITY\_LOCAL\_OVERRIDE – качество хорошее, значение сигнала могло быть изменено клиентом;
- › При других значениях параметра Quality качество сигнала определяется как OPC\_QUALITY\_UNKNOWN. При числовых значениях меньше 192 оно определяется как плохое. При значениях больше 192 качество определяется как хорошее.

При потере связи с OPC DA сервером устанавливается значение качества:

› (24) OPC\_QUALITY\_COMM\_FAILURE – связь с сервером, от которого должны быть получены данные, отсутствует.

При останове модуля всем сигналам устанавливается значение качества:

› (28) OPC\_QUALITY\_OUT\_OF\_SERVICE – сигнал не обслуживается.

Для сигналов, у которых не установлено свойство Quality, если они не используются какими-либо прочими модулями, устанавливается качество:

› (28) OPC\_QUALITY\_OUT\_OF\_SERVICE – сигнал не используется;

› (0) OPC\_QUALITY\_BAD – не выполнена подписка на сигнал.

## 1.1.2.4.2.4. Диагностика работы модуля

### Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

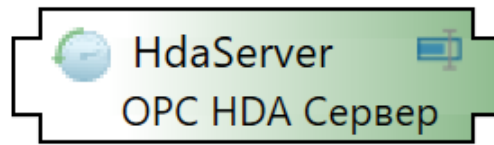
Для анализа ошибок, возникающих в процессе работы модуля, воспользуйтесь сервисным приложением Просмотрщик лога кадров.

№	Дата	Время	Описание	Группа
17	04.07.2023	12:30:11:005	Принят параметр конфигурации 'UseAsyncWrite' со значением 'False'	Group_1
18	04.07.2023	12:30:11:005	Принят параметр конфигурации 'ReadFromDevice' со значением 'False'	Group_1
19	04.07.2023	12:30:11:005	Принят параметр конфигурации 'ForcedReadOnConnect' со значением 'False'	Group_1
20	04.07.2023	12:30:11:005	Принят параметр конфигурации 'AllowAdrDubForInSignals' со значением 'False'	Group_1
21	04.07.2023	12:30:11:005	Принят параметр конфигурации 'LocalTimestampForInSignals' со значением 'False'	Group_1
22	04.07.2023	12:30:11:005	Принят параметр конфигурации 'ReAddItems' со значением 'True'	Group_1
23	04.07.2023	12:30:11:005	Принят параметр конфигурации 'ReAddItemsInterval' со значением '60'	Group_1
24	04.07.2023	12:30:11:005	Принят параметр конфигурации 'Algorithm' со значением '0'	Group_1
25	04.07.2023	12:30:11:005	Принят параметр конфигурации 'ActiveServerSelectionStrategy' со значением '0'	Group_1
26	04.07.2023	12:30:11:005	<-- CGroup::InitializeParameters	Group_1
27	04.07.2023	12:30:11:005	Принят параметр конфигурации 'Server_1.CompName' со значением 'localhost'	Group_1

Принят параметр конфигурации 'Server\_1.CompName' со значением 'localhost'

0	1	2	3	4	5	6
---	---	---	---	---	---	---

## 1.1.2.4.3. OPC HDA Сервер



Модуль OPC HDA Server предназначен для предоставления истории событий и значений сигналов клиентам по спецификации OPC HDA.

Модуль выполняет следующие функции:

- › предоставляет историю значений сигналов по спецификации OPC HDA;
- › предоставляет историю событий по внутреннему расширению спецификации OPC HDA.



Историю событий от OPC HDA Server могут получать только клиентские приложения Astra.HMI.Alarms, Astra.HMI.

Порядок предоставления истории событий/значений:

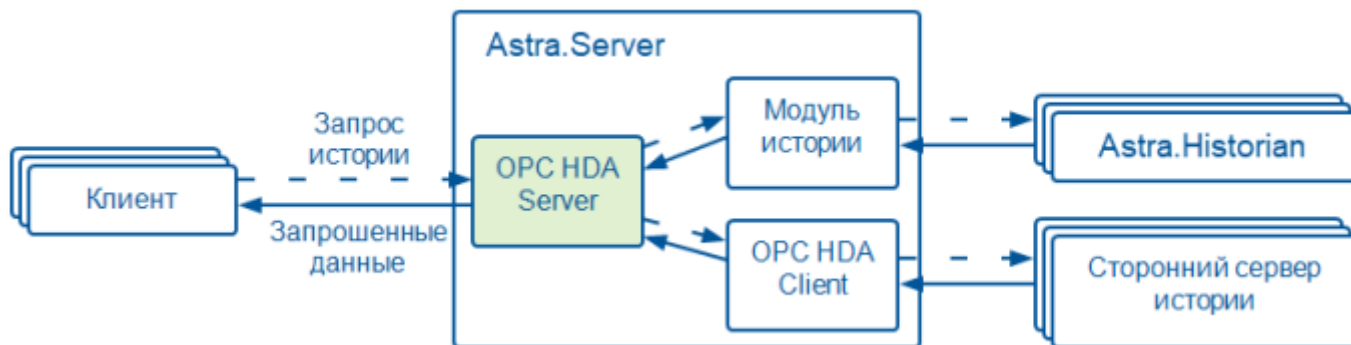
1. Клиент запрашивает у модуля OPC HDA Server историю событий/значений.
2. Запрос передаётся серверу истории, в котором хранятся нужные данные:
  - › серверу истории Astra.Historian запрос передаёт модуль истории;
  - › стороннему серверу истории запрос передаёт модуль OPC HDA Client.



История событий хранится и запрашивается только из Astra.Historian.

3. Полученные от сервера истории данные OPC HDA Server передаёт запросившему их клиенту.

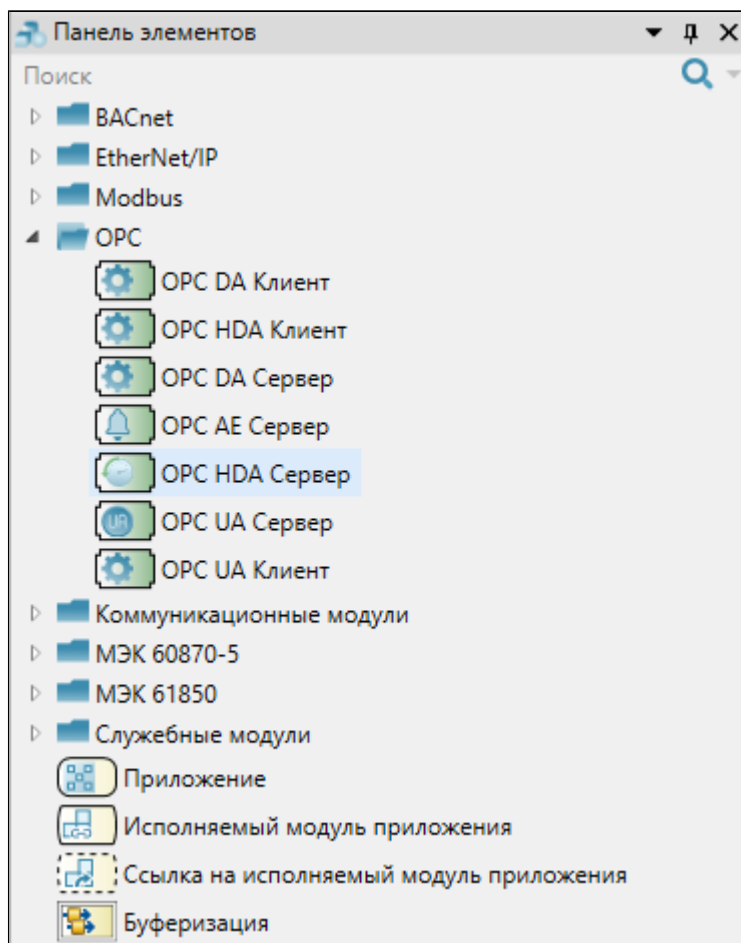




В режиме РЕЗЕРВ модуль выполняет те же функции, что и в режиме РАБОТА.

# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

Свойства	
HdaServer OPC HDA Сервер	
Общие	
Идентификатор сервера	Astra.HDAServer
Имя	HdaServer
Параметры модуля	
Максимальное количество записей в одном запросе	100000
Активность	Да
Отображаемое имя	
Параметры журналирования	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения

## Общие

Параметр	Описание
Имя	Идентификатор модуля в конфигурации Astra.Server, значение сервисного сигнала «Id»
Идентификатор сервера	Идентификатор, используемый для подключения к серверу

## Параметры модуля

Параметр	Описание
Максимальное количество записей в одном запросе	Значение по умолчанию: 100000
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>› Да – модуль запущен;</li> <li>› Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом Active.Set

Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## Настройка предоставления истории

Параметры, используемые для подключения клиентов по OPC HDA, указаны в файле конфигурации:



<папка Astra.Server>\Server\Astra.Server.xml.

```

Astra.Server.xml – Блокнот
Файл  Правка  Формат  Вид  Справка
<?xml version="1.0" encoding="windows-1251"?>
<configuration>
  <install ServiceName="Astra.Server" ExeName="Astra.Server.exe">
    <ComServers>
      <OPCDA ProgID="Astra.OPCDA Server" CLSID="{28A2AD9C-C45E-4C6b-A0C3-6E363F99CA72}" />
      <OPCAE ProgID="Astra.OPCAE Server" CLSID="{0CAEA48A-D7E6-44A4-85FD-C27836727D07}" />
      <HDA ProgID="Astra.HDA Server" CLSID="{8002740A-886F-1488-2280-42058D6D5CA8}" />
    </ComServers>
  </install>
  <Storage Filename="AstraServer.cfg" />
  <Connection Port="4572" />
  <Backup Path="..\Backups" Time="00:00" StorageDepth="14" />
  <Log Path="..\Logs" />
  <Instance ID="FF9C3A30-0C6F-43EB-AAAF-5FDE00846D94" />
  <Culture LangID="ru-RU" />
</configuration>

```

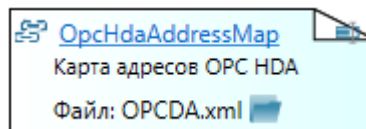
Стр 1, столб 46      100%      Windows (CRLF)      UTF-8

Значения параметров задаются при установке сервера. Чтобы изменить их, укажите в файле новые значения и перезагрузите Astra.Server.




## 1.1.2.4.3.2. Карта адресов

Для настройки сигналов модуля OPC HDA Сервер используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов OPC HDA в исполняемое приложение;



2. Откройте редактор карты адресов.

	Сигнал	Тип	Привязка	Идентификатор узла
	▼	▼	▼	▼
	uint4	uint4	непосредственн	1
	int4	int4	непосредственн	2

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>› Непосредственно;</li><li>› Только чтение;</li><li>› Не привязан.</li></ul>
Идентификатор узла	Тег в сервере — адрес сигнала или свойства в адресном пространстве OPC HDA сервера, к которому подключается модуль. Чтобы настроить сигнал на получение значения свойства сигнала со стороннего DA сервера, в параметре укажите через точку имя свойства сигнала

# Диагностика работы модуля

## Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

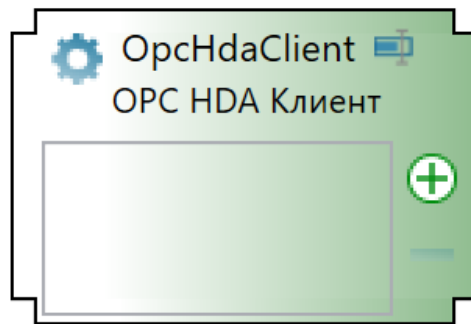
Для просмотра журнала работы модуля воспользуйтесь сервисным приложением Просмотрщик лога кадров.

№	Дата	Время	Описание
7	03.07.2023	21:19:48:072	Получена команда завершения работы.
8	03.07.2023	21:19:48:369	Журнал закрыт
9	03.07.2023	21:20:01:790	Журнал открыт для записи
10	03.07.2023	21:27:22:850	Получена команда завершения работы.
11	03.07.2023	21:27:22:850	Получена команда завершения работы.
12	03.07.2023	21:27:23:225	Журнал закрыт
13	03.07.2023	21:28:34:631	Журнал открыт для записи
14	03.07.2023	21:28:35:819	Не удалось зарегистрировать COM сервер: 0x80004015 (Класс настроен н
15	03.07.2023	21:28:36:491	Получена команда завершения работы.
16	03.07.2023	21:28:36:787	Журнал закрыт
17	03.07.2023	21:29:21:865	Журнал открыт для записи

Журнал закрыт

0 1 2 3 4 5 6 7 8 9

## 1.1.2.4.4. OPC HDA Клиент



Модуль OPC HDA Client предназначен для запроса истории значений из сторонних серверов истории (не Astra.Historian) по OPC HDA.

Принцип работы:

- Для каждого параметра, значения которого хранятся в стороннем сервере истории, создаётся сигнал в Astra.Server и ставится на обслуживание модулю OPC HDA Client.
- Когда клиент запрашивает у Astra.Server историю значений сигнала, поставленного на обслуживание модулю OPC HDA Client, модуль запрашивает историю значений сигнала у сервера истории, в котором она хранится. Данные, полученные от сервера истории, передаются запросившему их клиенту.



История значений предоставляется клиентам по протоколам OPC HDA и OPC UA.



### Работа с дублирующими серверами истории



Для работы с дублирующими серверами истории (серверами, в которых хранятся одинаковые данные), они объединяются в группы.

Каждая группа включает в себя серверы истории, дублирующие друг друга, или один сервер, если у него нет дублирующих серверов истории. Если в группе более одного сервера истории, запрос будет передан первому серверу истории, с которым будет установлено соединение.

## **Работа в резерве**

В режиме РЕЗЕРВ модуль OPC HDA Client выполняет те же функции, что и в режиме РАБОТА.

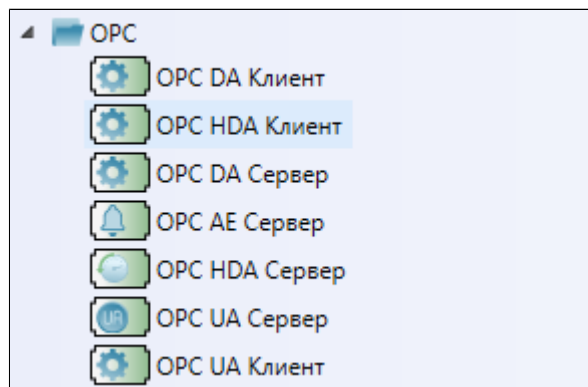
## **Работа в составе AccessPoint**

При работе в составе Astra.AccessPoint модуль OPC HDA Client выполняет те же функции, что и при работе в составе Astra.Server. Однако в Astra.AccessPoint не нужно настраивать получение исторических данных для сигналов, подключаемых в составе источников данных: настройки сигналов и групп серверов загружаются из источников данных при подключении к ним.

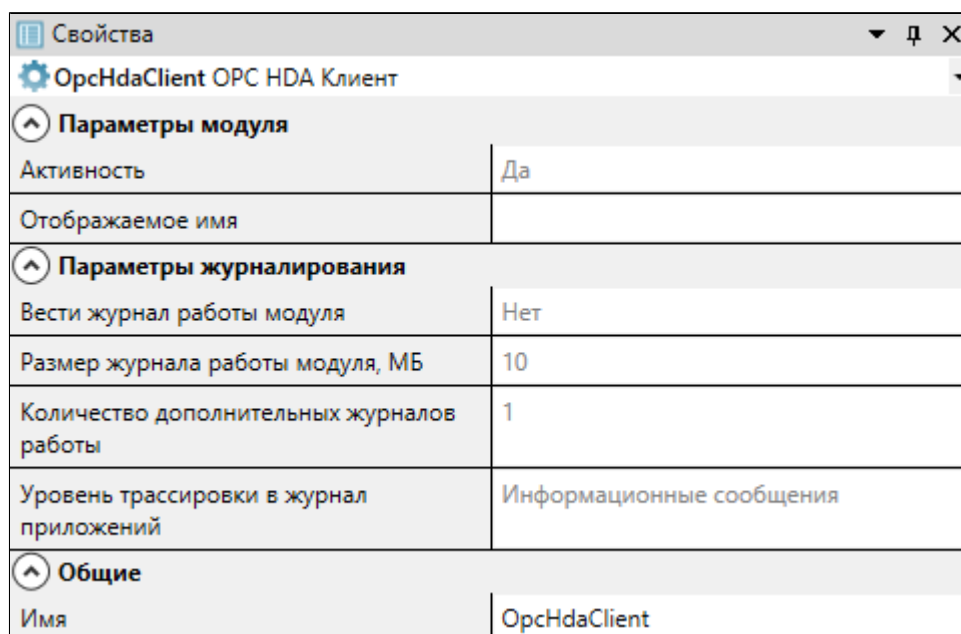
Информация о группах серверов передаётся по TCP между модулем TCP Server в составе Astra.Server и модулем HUB в составе Astra.AccessPoint.

## Добавление и общие настройки

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.



### Общие

Параметр	Описание
Имя	Имя модуля

### Параметры модуля

Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>➤ Да – модуль запущен;</li> <li>➤ Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

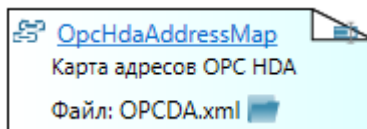
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.




## 1.1.2.4.4.2. Карта адресов

Для настройки сигналов модуля OPC HDA Клиент используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов OPC HDA в исполняемое приложение;



2. Откройте редактор карты адресов.

	Сигнал	Тип	Привязка	Идентификатор узла
	▼	▼	▼	▼
	uint4	uint4	непосредственн	1
	int4	int4	непосредственн	2

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>➤ Непосредственно;</li><li>➤ Только чтение;</li><li>➤ Не привязан.</li></ul>
Идентификатор узла	Тег в сервере — адрес сигнала или свойства в адресном пространстве OPC HDA сервера, к которому подключается модуль. Чтобы настроить сигнал на получение значения свойства сигнала со стороннего DA сервера, в параметре укажите через точку имя свойства сигнала

# Диагностика работы модуля

## Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

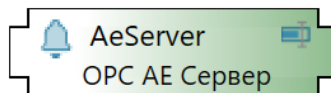
Для просмотра журнала работы модуля воспользуйтесь сервисным приложением Просмотрщик лога кадров.

№	Дата	Время	Описание
1	03.07.2023	18:20:24:014	Журнал открыт для записи
2	03.07.2023	21:19:25:259	Получена команда завершения работы.
3	03.07.2023	21:19:25:259	Получена команда завершения работы.
4	03.07.2023	21:19:31:400	Журнал закрыт
5	03.07.2023	21:19:46:228	Журнал открыт для записи
6	03.07.2023	21:19:47:415	Не удалось зарегистрировать COM сервер: 0x80004015 (Класс настроен на использование идентификатора безопасности, отличного от используемого вызывающей стороной)
7	03.07.2023	21:19:48:072	Получена команда завершения работы.
8	03.07.2023	21:19:48:369	Журнал закрыт
9	03.07.2023	21:20:01:790	Журнал открыт для записи
10	03.07.2023	21:27:22:850	Получена команда завершения работы.
11	03.07.2023	21:27:22:850	Получена команда завершения работы.

Не удалось зарегистрировать COM сервер:  
0x80004015 (Класс настроен на использование  
идентификатора безопасности, отличного от  
используемого вызывающей стороной)

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

## 1.1.2.4.5. OPC AE Сервер

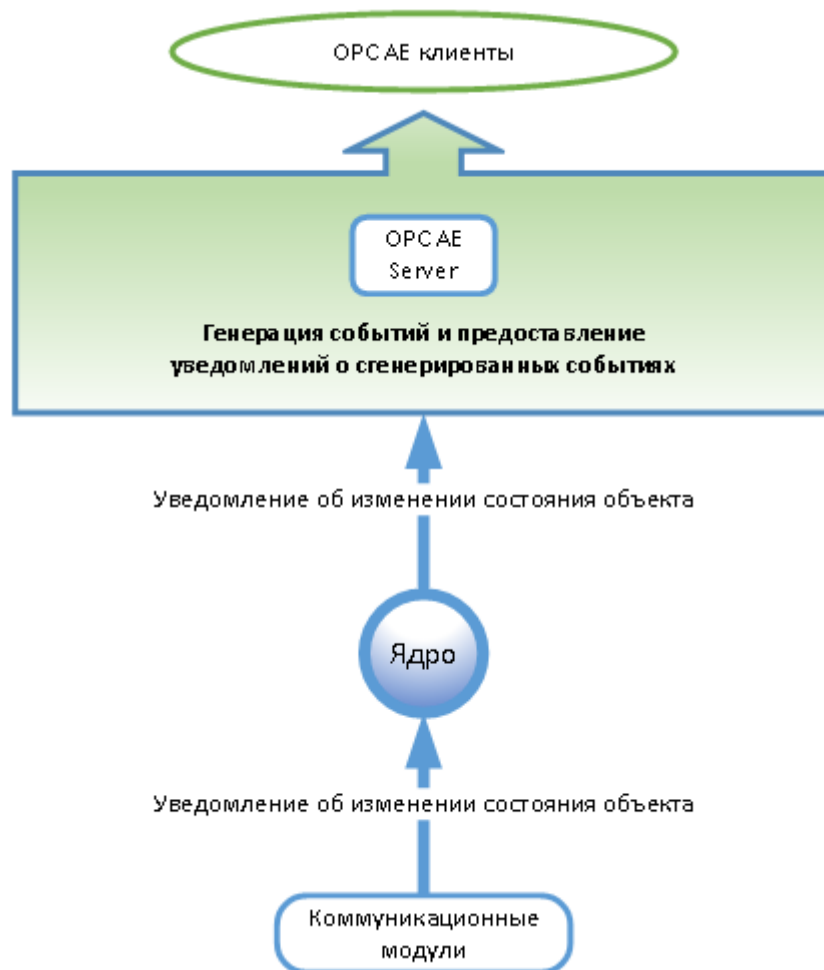


Основные функции:

- генерация уведомлений о событиях по заданным условиям и передача сгенерированных уведомлений подключенным клиентам;
- прием информации о квитировании уведомлений о событиях от одного из клиентов и передача этой информации другим клиентам.

Работая в составе Astra.Server, модуль реализует серверную часть спецификации OPC AE. В составе Astra.Server может присутствовать только один модуль OPC AE Server.

Ядро получает от коммуникационного модуля уведомление об изменении состояния объекта и передает полученное уведомление модулю OPC AE Server, если у модуля есть подписка на пришедшее уведомление. Модуль OPC AE Server генерирует событие по пришедшему уведомлению, которое удовлетворяет условию генерации события. Генерация производится по типам условий. Источниками данных для генерации события являются сигналы Astra.Server. Клиент получает уведомление о сгенерированном событии от модуля OPC AE Server, в соответствии со спецификацией OPC AE.



Модуль OPC AE Server посылает уведомления клиенту, если:

- наступило новое событие, на которое подписан клиент;
- поступил запрос обновления списка активных условий от клиента.

Уведомления о событиях придут клиенту с пометкой, что уведомления обновленные, а не оригинальные.

При старте модуля происходит инициализация модуля. Инициализация включает в себя:

- чтение конфигурационных данных модуля;
- подписка у ядра на получение уведомлений об изменениях сигналов;
- проверка корректности настройки адресов для сигналов и постановка их на обслуживание.

После инициализации модуль ожидает подключения клиента.

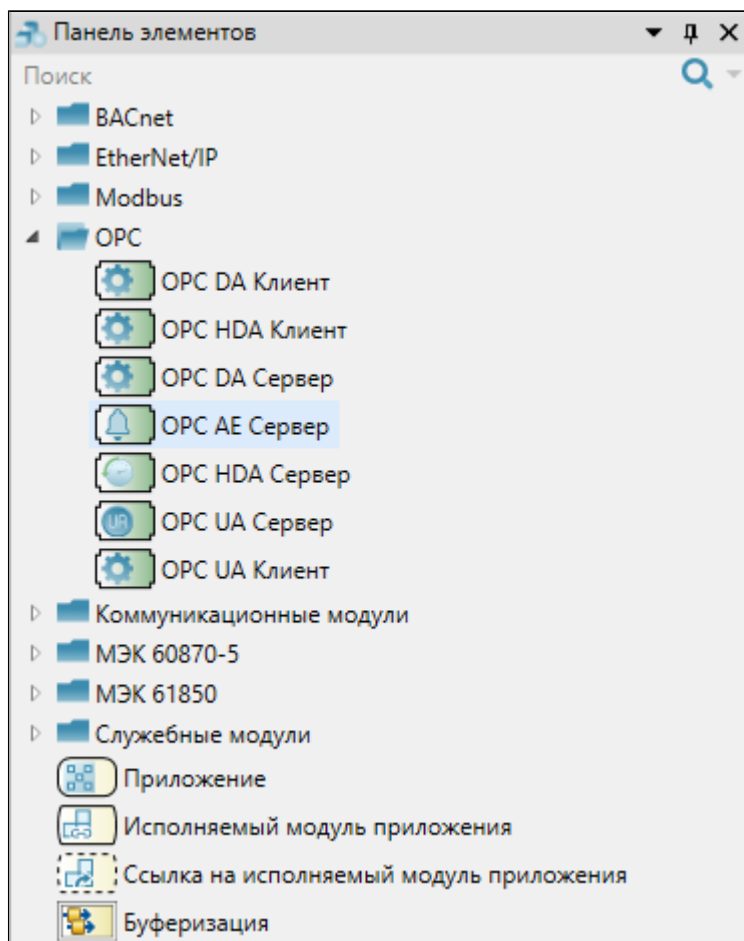
Модуль позволяет клиенту делить уведомления по категориям (Discrete, Enumeration, Dynamic, Level, VSLevel, Deviation), но области источника не

различает. Другими словами, модуль не позволяет клиенту различать дерево сигналов источника.



# Настройка модуля

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

Свойства	
AeServer OPC AE Сервер	
<b>Общие</b>	
Идентификатор сервера	Astra.OPCAEServer
Имя	AeServer
<b>Параметры модуля</b>	
Правило обработки подавленных событий	На стороне сервера
Отсылать квитирование по подавленным источникам	Нет
Использовать расширения OPC AE	Нет
Уведомления о блокировании/подавлении	Отключены
Максимальный размер очереди уведомлений	100000
Передавать события по OPC DA протоколу	Нет
Тип источника для сервиса агрегации событий	Оперативные события
Шаблон сообщения события	
Активность	Да
Отображаемое имя	
<b>Таблица оперативных событий</b>	
Сохранять события в оперативной таблице событий	Нет
Папка для хранения оперативных событий	
Ограничение по давности цепей событий, час	0
Ограничение по количеству цепей событий	0
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения

## Общие

Параметр	Описание
----------	----------

Имя	Имя модуля
Идентификатор сервера	Используется для подключения к серверу

## Параметры модуля

Параметр	Описание
Правило обработки подавленных событий	Значения: <ul style="list-style-type: none"> <li>&gt; На стороне сервера</li> <li>&gt; На стороне клиента</li> </ul>
Отсылать квитирование по подавленным источникам	Значения: <ul style="list-style-type: none"> <li>&gt; Да</li> <li>&gt; Нет</li> </ul>
Использовать расширения OPC AE	Значения: <ul style="list-style-type: none"> <li>&gt; Да</li> <li>&gt; Нет</li> </ul>
Уведомления о блокировании/подавлении	Значения: <ul style="list-style-type: none"> <li>&gt; Отключены</li> <li>&gt; По каждому условию</li> <li>&gt; По источнику/зоне</li> </ul>
Максимальный размер очереди уведомлений	Количество уведомлений, стоящих в очереди на генерацию уведомлений о событии для клиентов. При достижении максимального количества уведомлений приостанавливается работа модуля. Модуль продолжает работу, если очередь освободилась хотя бы на одно уведомление. Значение параметра находится в диапазоне от 10 000 до 10 000 000. По умолчанию параметр принимает значение 10 000
Передавать события по OPC DA протоколу	Предоставление возможности доступа к событиям и квитирования событий по OPC DA протоколу. Доступ предоставляется через массив динамических сигналов, с

	<p>которыми можно работать через любой OPC DA клиент. Если параметр имеет значение Да - возможности предоставляются, если Нет - не предоставляются. Чтобы массив сигналов появился, перезапустите Astra.Server</p>
Тип источника для сервиса агрегации событий	<p>Активность модуля при запуске/перезапуске Astra.Server:</p> <ul style="list-style-type: none"> <li>➤ Оперативные события;</li> <li>➤ OAT</li> </ul>
Шаблон сообщения события	Выбор файла с шаблоном сообщений событий
Активность	<p>Активность модуля при запуске/перезапуске Astra.Server:</p> <ul style="list-style-type: none"> <li>➤ Да – модуль запущен;</li> <li>➤ Нет – модуль остановлен.</li> </ul> <p>Управляется служебным сигналом Active.Set</p>
Отображаемое имя	Отображаемое имя модуля

## Таблица оперативных событий

Параметр	Описание
Сохранять события в оперативной таблице событий	<p>Значения:</p> <ul style="list-style-type: none"> <li>➤ Да</li> <li>➤ Нет</li> </ul>
Строка подключения ODBC	Строка подключения ODBC к базе данных с таблицей оперативных событий
Тип оперативной таблицы событий	<p>Значения:</p> <ul style="list-style-type: none"> <li>➤ PostgreSQL OAT</li> <li>➤ Native OAT</li> </ul>

Ограничение по давности цепей событий, час	Предел давности деактивированной цепи событий, после которого оно будет автоматически квитировано (0 трактуется как отсутствие ограничения )
Ограничение по количеству цепей событий	Предел по количеству цепей событий, превышение которого приведет к автоматическому квитированию неактивных цепей (0 трактуется как отсутствие ограничения)

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

# Функциональные возможности

## Режимы работы модуля

Модуль OPC AE Server может работать в двух режимах: РАБОТА или РЕЗЕРВ. Режим работы модуля в сервере устанавливает ядро. Режим работы модуля напрямую зависит от режима работы сервера. Режим работы сервера может измениться после резервных переходов.

В режиме РАБОТА модуль выполняет функции:

- › получает уведомления об изменении состояния объекта;
- › генерирует события;
- › передает уведомления о событии клиенту;
- › передает клиенту уведомление о квитировании.

Отличие режима РЕЗЕРВ от режима РАБОТА заключается в том, что модуль не передает уведомления о событиях клиенту, т.к. не происходит генерация событий.

При переходе из режима РЕЗЕРВ в режим РАБОТА модуль отправляет клиенту уведомления о событиях, пришедшие со времени работы модуля в режиме РАБОТА.

## Постановка сигнала на обслуживание

Для постановки сигнала на обслуживание модулем OPC AE Server при запуске модуля выполняются следующие действия:

- › проверка правильности задания адреса сигнала;
- › проверка разрешений алармов для типа объекта;
- › проверка наличия подусловий и разрешений подусловий.

## Генерация события

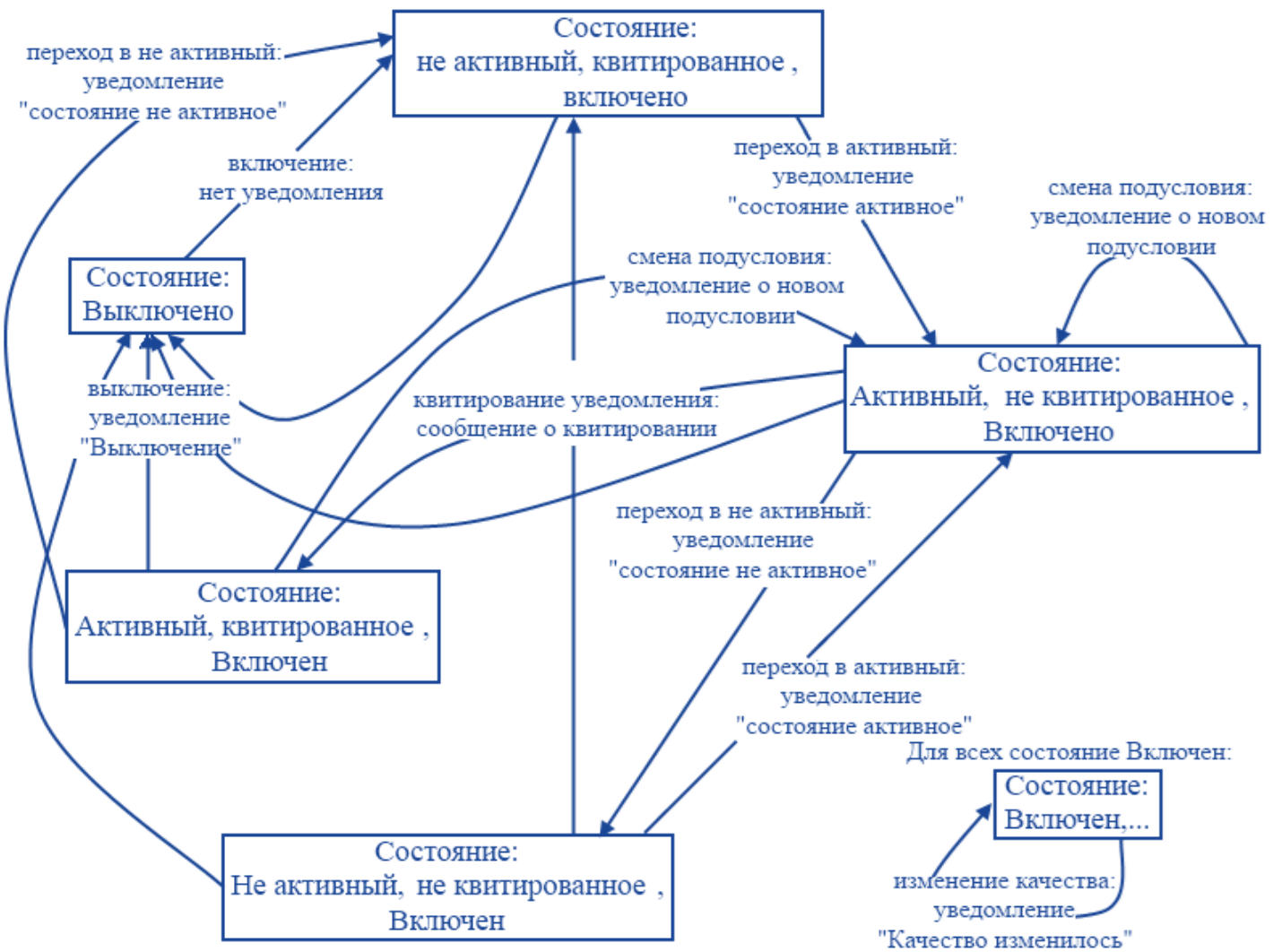
На рисунке ниже представлен алгоритм перехода между состояниями условий в виде графа. Первоначальным состоянием условия при запуске модуля

является состояние: неактивное условие, квитируванное, включено для создания уведомлений.

Переход условия в состояние активное происходит при получении модулем уведомления о событии. При этом происходит проверка выполнения, какого либо из подусловий. Определив, какое из подусловий выполняется, модуль генерирует событие и посылает уведомление о событии клиенту.

Переход условия в состояние включено происходит при изменении в файле конфигурации значения атрибута Enabled для тега EventCondition.

Переход условия в состояние квитируванное происходит при подтверждении прочтения уведомления о событии клиентом.



Для модуля OPC AE Server источниками данных являются сигналы Astra.Server. С каждым источником данных может быть связано несколько подусловий. При выполнении любого подусловия OPC AE клиент получает уведомление о событии.

Модуль OPC AE Server реализует генерацию событий по следующим типам условий:

- › перечисления;
- › дискретные;
- › динамические;
- › события по уровню;
- › события по произвольному количеству уровней;
- › предельное отклонение.

## Перечисления

События по перечислению генерируются в соответствии с идентификатором события, получаемым от стороннего источника данных (под сторонним источником подразумеваются любые источники данных сервера, которые реализованы вне модуля событий – любые коммуникационные модули, модуль вычислений и т.п.). Идентификатор содержит код подусловия, которое активируется и по которому формируется событие. Применяются при работе с численными типами объектов: uint1, int1, uint2, int2, uint4, int4, uint8, int8, float, double.

## Дискретные

События по дискретным параметрам содержат два независимых подусловия: On и Off. Для параметра они могут быть настроены, а могут отсутствовать. Подусловия активируются при соответствующем изменении значения параметра. Применяются при работе с типами объектов bool.

## Динамические



События по динамическим условиям, созданным пользователем. Подусловия активируются при соответствующем изменении значения параметра. Возможность настраивать события для динамических сигналов сервера. Например, для сигналов, характеризующих состояние резервирования. Применяются при работе с типами объектов string.

Модуль OPC AE Server имеет возможность указать источники данных для генерации событий со стороны OPC DA клиента. Такими источниками выступают строковые сигналы типа string. В настройках модуля следует связать сигнал с типом условия, по которому будут генерироваться события.

Чтобы сгенерировать динамические алармы в тип данных string (строковый параметр) следует внести запись в значение XML конструкции вида:

## **События по уровню**

События по уровням, созданным пользователем. Подусловия активируются при соответствующем изменении значения параметра.

Чтобы указать пределы, используются типы подусловий - Normal, Hi, Lo, HiHi, LoLo, HiHin, LoLon (n в диапазоне от 2 до 5):

- Normal - подусловие, которое выполняется, если не активно ни одно из других подусловий;
- Hi, HiHi и HiHin - подусловия, которые выполняются, если значение параметра больше или равно указанного значения;
- Lo, LoLo и LoLon - подусловия, которые выполняются, если значение параметра меньше или равно указанного значения.

## **События по произвольному количеству уровней**

События по уровням, созданным пользователем. Пользователь может задать произвольное количество подусловий. Подусловия активируются при соответствующем изменении значения параметра.

Чтобы указать пределы, используются типы подусловий - Above, Below и Normal:

- › **Above** - подусловие, которое выполняется, если значение параметра больше или равно указанного значения;
- › **Below** - подусловие, которое выполняется, если значение параметра меньше или равно указанного значения;
- › **Normal** - подусловие, которое выполняется, если не активно ни одно из подусловий типов Above, Below.

Каждое такое подусловие должно иметь уникальное имя в рамках условия. Имя подусловия задается произвольно пользователем в атрибуте Name элемента Subcondition. Ниже показаны примеры уникальных имен подусловий:

- › **UpperEmergency** - уровень выше аварийного;
- › **UpperLimit** - уровень выше предельного;
- › **UpperNormative** - уровень выше нормативного;
- › **Normal** - уровень в норме;
- › **LowerLimit** - уровень ниже предельного;
- › **LowerEmergency** - уровень ниже аварийного.

Применяются при работе с численными типами объектов: uint1, int1, uint2, int2, uint4, int4, uint8, int8, float, double.

## Предельное отклонение

События с предельным отклонением содержат два независимых подусловия: Up и Down. Подусловия активируются при соответствующем изменении значения параметра. Применяются при работе с численными типами объектов: uint1, int1, uint2, int2, uint4, int4, uint8, int8, float, double.

## Атрибуты уведомлений

Модуль OPC AE Server предоставляет клиентам возможность просматривать уведомления с дополнительной информацией. Дополнительная информация представляет собой атрибуты сообщения.

Отправленные модулем уведомления для клиента включают следующие атрибуты:

- › Source – источник события – полное имя тега, от которого произошло событие;
- › Time – время, когда сгенерировано уведомление о событии;
- › EventCategory – категория события – коды категорий событий:
  - › 0 – Discrete;
  - › 1 – Enumeration;
  - › 2 – Dynamic;
  - › 3 – Level;
  - › 3 – VSLevel;
  - › 4 – Deviation.
- › Type – тип события – существуют следующие типы событий:
  - › Discrete;
  - › Enumeration;
  - › Dynamic;
  - › Level;
  - › VSLevel;
  - › Deviation.
- › Severity – приоритет – важность данного события;
- › Message – сообщение – текст, который описывает событие.

Отправленные модулем уведомления для клиента помимо стандартных атрибутов имеют следующие дополнительные атрибуты, которые отображаются в OPC AE клиенте:

- › ConditionName – имя условия генерации события;
- › SubConditionName – имя подусловия генерации события;
- › ChangeMask – маска изменений – признак показывает, какие условия изменились, чтобы вызвать сервер для отправки уведомления о событии;
- › NewState – индикатор нового состояния условия показывает новые значения свойств условий: разрешений, активности, квитирования;
- › ConditionQuality – качество условия показывает качество основных элементов данных, на которых это условие основывается;

- › AckRequired – требование квитирования – флаг, определяющий требуется ли квитирование от клиента уведомления о событии;
- › ActiveTime – время перехода состояния события в активное (время, когда произошло событие);
- › Cookie – текстовая информация о событии, хранящаяся в сервере. Информация является недоступным для клиента;
- › ActorID – идентификатор OPC клиента, который квитирует событие.

## Рассылка уведомлений о квитировании

Модуль получает уведомление о квитировании события от клиента и отправляет его остальным клиентам, подписанным на это событие.



Уведомления о квитировании события содержат следующую информацию:

- › имя пользователя;
- › дата и время квитирования события;
- › примечания пользователя.

## Блокирование и подавление

Модуль OPC AE Server предоставляет клиентам возможности блокирования и подавления источников или объектов.

При блокировании источника или объекта модуль OPC AE Server:

- › прекращает генерацию событий данного источника или объекта;
- › прекращает сохранение в историю событий данного источника или объекта;

- › прекращает уведомление клиентов о событиях данного источника или объекта.

Блокирование источников или объектов выполняется бессрочно, пока блокировка не будет отменена клиентом.

При подавлении источника или объекта модуль OPC AE Server:

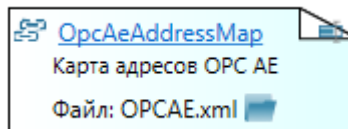
- › продолжает генерацию событий данного источника или объекта;
- › продолжает сохранение в историю сгенерированных событий данного источника или объекта;
- › прекращает уведомление клиентов о событиях данного источника или объекта.

Подавление источника или объекта возможно на любой период времени или бессрочно, пока подавление не будет отменено клиентом.




## 1.1.2.4.5.3. Карта адресов

Для настройки сигналов модуля OPC AE Сервео используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов OPC AE в исполняемое приложение;



2. Откройте редактор карты адресов.

	Сигнал	Тип	Привязка	Идентификатор узла
	▼	▼	▼	▼
	int4	int4	непосредстве	
	uint4	uint4	непосредстве	

## Параметры карты адресов

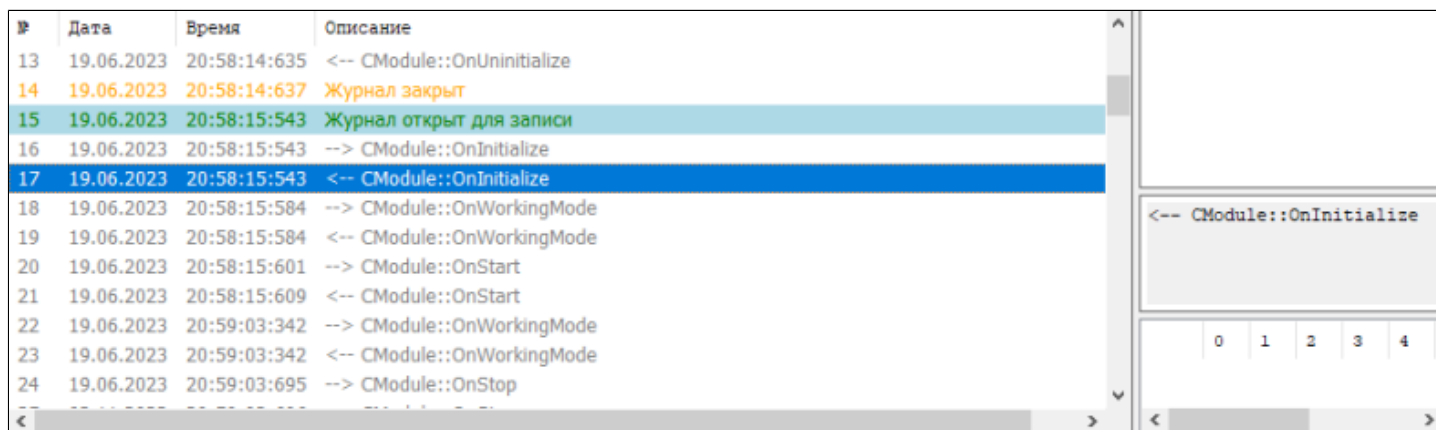
Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>› Непосредственно;</li><li>› Только чтение;</li><li>› Не привязан.</li></ul>
Идентификатор узла	Идентификатор узла в OPC EA сервере.

# Диагностика работы модуля

## Журнал работы модуля

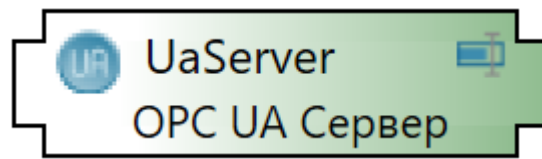
Для анализа ошибок, возникающих в процессе работы модуля, воспользуйтесь сервисным приложением Просмотрщик лога кадров.

№	Дата	Время	Описание
13	19.06.2023	20:58:14:635	<-- CModule::OnUninitialize
14	19.06.2023	20:58:14:637	Журнал закрыт
15	19.06.2023	20:58:15:543	Журнал открыт для записи
16	19.06.2023	20:58:15:543	--> CModule::OnInitialize
17	19.06.2023	20:58:15:543	<-- CModule::OnInitialize
18	19.06.2023	20:58:15:584	--> CModule::OnWorkingMode
19	19.06.2023	20:58:15:584	<-- CModule::OnWorkingMode
20	19.06.2023	20:58:15:601	--> CModule::OnStart
21	19.06.2023	20:58:15:609	<-- CModule::OnStart
22	19.06.2023	20:59:03:342	--> CModule::OnWorkingMode
23	19.06.2023	20:59:03:342	<-- CModule::OnWorkingMode
24	19.06.2023	20:59:03:695	--> CModule::OnStop



The screenshot shows a log viewer application with a table of log entries and a detailed view of a selected entry. The table has columns for ID, Date, Time, and Description. The selected entry (ID 17) is highlighted in blue. The detailed view on the right shows the selected entry's description: <-- CModule::OnInitialize. Below the detailed view is a navigation bar with buttons labeled 0, 1, 2, 3, and 4.

## 1.1.2.4.6. OPC UA Сервер



Модуль OPC UA (Unified Architecture) – коммуникационный модуль, предназначенный для обмена данными между Astra.Server и сторонними клиентами по спецификации OPC UA.

Функции модуля OPC UA:

- › предоставление оперативных значений и событий клиентам;
- › предоставление исторических значений и событий клиентам;
- › запись значений в Astra.Server клиентами.

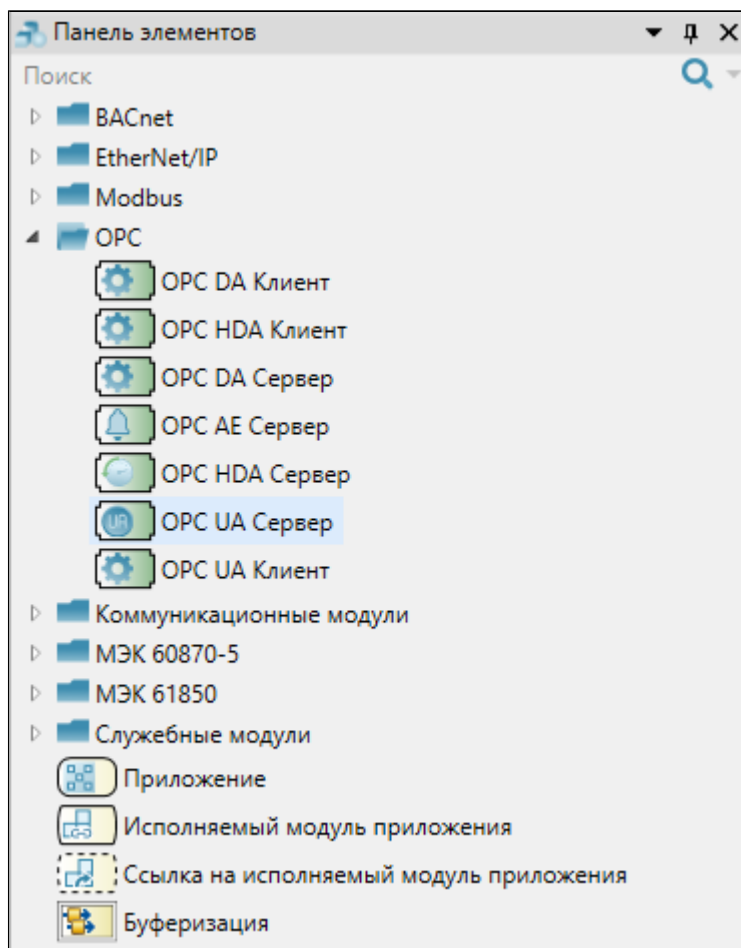
Преимущества:

- › кроссплатформенность;
- › отказ от закрытого стандарта COM/DCOM в пользу TCP;
- › простота настроек удаленного подключения;
- › безопасность передаваемых данных благодаря использованию шифрования и аутентификации;
- › оперативные данные, исторические данные, события передаются через один сервер по единому интерфейсу.



## 1.1.2.4.6.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

Свойства	
UaServer OPC UA Сервер	
<b>Общие</b>	
Имя точки подключения	AstraOpcUaServer
Запретить клиентам изменение сигналов	Нет
Переводить в состояние SUSPENDED в резерве	Нет
Имя	UaServer
<b>Сетевые параметры</b>	
Порт OPC TCP протокола	62544
Порт HTTP протокола	8080
Использовать HTTP протокол	Нет
Режим безопасности сетевого канала	None
Discovery сервер	opc.tcp://localhost:4840
<b>Параметры аутентификации</b>	
Анонимный доступ	Да
Учетной записью Windows	Нет
Пользовательским сертификатом	Нет
<b>Параметры модуля</b>	
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения


## Общие

Параметр	Описание
Имя точки подключения	Используется при подключении через HTTP и HTTPS
Запретить клиентам изменения сигналов	<p>Определяет возможность клиентов изменять значения сигналов сервера:</p> <ul style="list-style-type: none"> <li>› Да - изменение значений сигналов запрещено;</li> <li>› Нет - клиенты могут изменять значения сигналов сервера.</li> </ul>

Переводить в состояние SUSPEND в резерве	<p>Определяет возможность перевода UA сервера в состояние SUSPENDED при переходе Astra.Server в состояние РЕЗЕРВ:</p> <ul style="list-style-type: none"> <li>➤ Да - сигнал Server.ServerStatus.State отображает состояние сервера:</li> <li>➤ SUSPENDED - Astra.Server находится в состоянии РЕЗЕРВ;</li> <li>➤ RUNNING - Astra.Server находится в состоянии РАБОТА;</li> <li>➤ Нет - при любом состоянии Astra.Server сигнал Server.ServerStatus.State принимает значение RUNNING. Значение по умолчанию</li> </ul>
Имя	Название модуля, которое отображается в дереве модулей и Редакторе адреса в Конфигураторе, а также в тегах служебных сигналов

## Сетевые параметры

Параметр	Описание
Порт OPC TCP протокола	Порт для обмена данными по протоколу TCP
Порт HTTP протокола	Порт для обмена данными по протоколу HTTP/HTTPS
Использовать HTTP протокол	<p>Обмен данными по протоколу HTTP/HTTPS:</p> <ul style="list-style-type: none"> <li>➤ Нет - обмен данными по протоколу HTTP/HTTPS запрещен;</li> <li>➤ HTTP - обмен данными по протоколу HTTP</li> <li>➤ HTTPS - обмен данными по протоколу HTTPS</li> </ul>
Режим безопасности сетевого канала	<p>Определяет защищенность канала передачи данных при установлении соединения между клиентом и сервером:</p> <ul style="list-style-type: none"> <li>➤ None - режим не обеспечивает дополнительных средств безопасности, данные передаются без шифрования;</li> </ul>

	<p>›Sign - для установки безопасного соединения требуется обмен сертификатами между сервером и клиентом;</p> <p>›SignAndEncrypt - для установки безопасного соединения требуется обмен сертификатами между сервером и приложением-клиентом. Поток данных между клиентом и сервером будет передаваться в зашифрованном виде</p>
Discovery сервер	<p>Адрес Local Discovery Server (LDS), по которому доступен UA сервер. Значение по умолчанию opc.tcp://localhost:4840</p> <div style="border: 1px solid #add8e6; padding: 10px; margin-top: 10px;"> <p> Local Discovery Server (LDS) - сервер, выполняющий роль точки доступа к множеству UA-серверов компьютера. С помощью LDS клиенты могут получить доступ к Astra.Server по спецификации OPC UA.</p> </div>

## Параметры аутентификации

Параметр	Описание
Анонимный доступ	<p>Определяет доступ пользователей без прохождения процедуры аутентификации:</p> <ul style="list-style-type: none"> <li>›Разрешить - к серверу могут подключаться любые пользователи.</li> <li>›Запретить - доступ анонимных пользователей запрещен.</li> </ul> <p>Чтобы к серверу могли подключаться любые пользователи без прохождения процедуры аутентификации, установите параметру значение Разрешить.</p>
Учетной записью Windows	<p>Определяет доступ пользователей по системной учетной записи:</p>

	<ul style="list-style-type: none"> <li>› Разрешить - к серверу могут подключаться пользователи, предоставившие логин и пароль локальной или доменной учетной записи.</li> <li>› Запретить - доступ пользователей по учетной записи запрещен.</li> </ul> <p>Чтобы к серверу могли подключаться только пользователи, предоставившие логин и пароль учетной записи Windows, установите параметру значение Разрешить .</p>
Пользовательским сертификатом	<p>Определяет доступ пользователей с помощью сертификата:</p> <ul style="list-style-type: none"> <li>› Разрешить - для аутентификации требуется предоставить пользовательский сертификат и приватный ключ.</li> <li>› Запретить - доступ пользователей с помощью сертификата запрещен.</li> </ul>

## Параметры модуля

Параметр	Описание
Активность	<p>Активность модуля при запуске/перезапуске Astra.Server:</p> <ul style="list-style-type: none"> <li>› Да – модуль запущен;</li> <li>› Нет – модуль остановлен.</li> </ul>
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

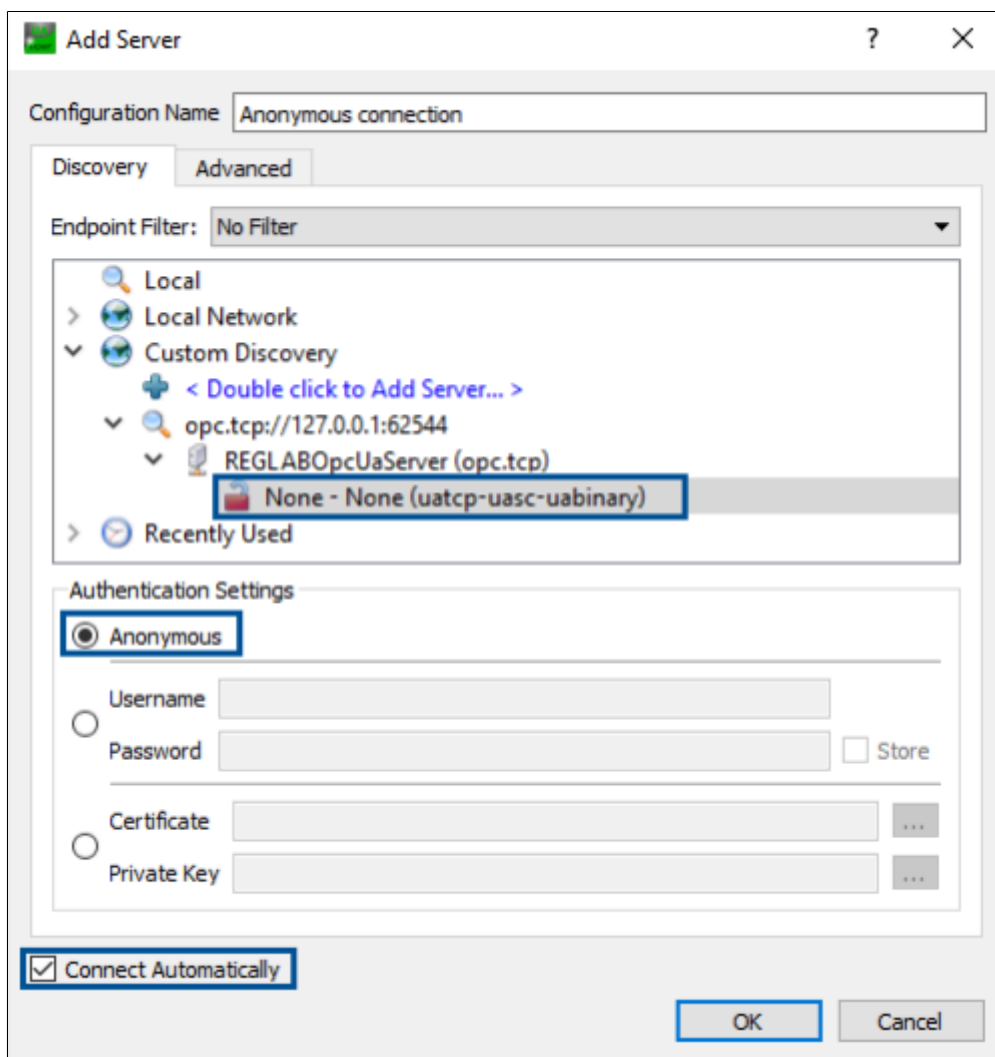
## 1.1.2.4.6.2. Способы аутентификации

Модуль OPC UA допускает несколько способов аутентификации пользователей.

### Анонимный доступ

Чтобы к серверу могли подключаться любые пользователи без прохождения процедуры аутентификации, установите параметру "Доступ анонимных пользователей" значение "Разрешить".

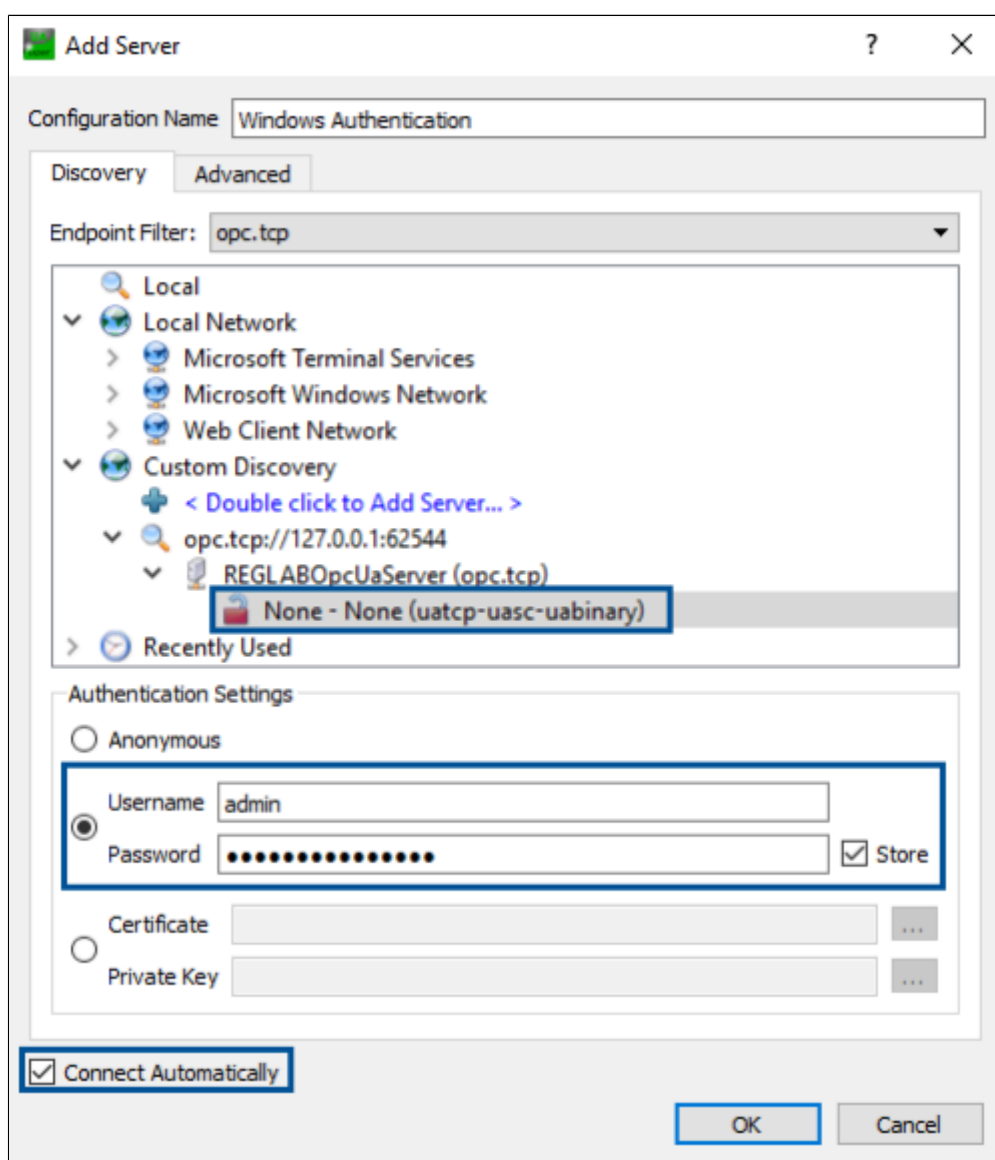
В клиенте UA Expert выберите сервер и тип аутентификации "Anonymous". Чтобы подключение к серверу началось сразу после нажатия кнопки "OK", установите флаг "Connect Automatically".



### Windows-аутентификация

Чтобы к серверу могли подключаться только пользователи, предоставившие логин и пароль учетной записи Windows, установите параметру Аутентификация учетной записью Windows значение Разрешить.

Чтобы подключиться к серверу, в клиенте UA Expert выберите сервер и введите Username и Password в соответствующие поля. Чтобы подключение к серверу началось сразу после нажатия кнопки ОК, установите флаг Connect Automatically.

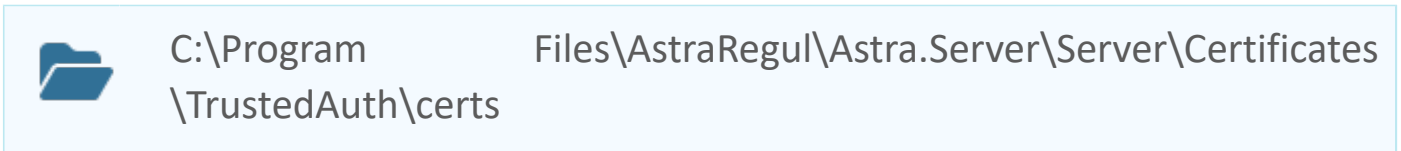


## Аутентификация сертификатом и приватным ключом

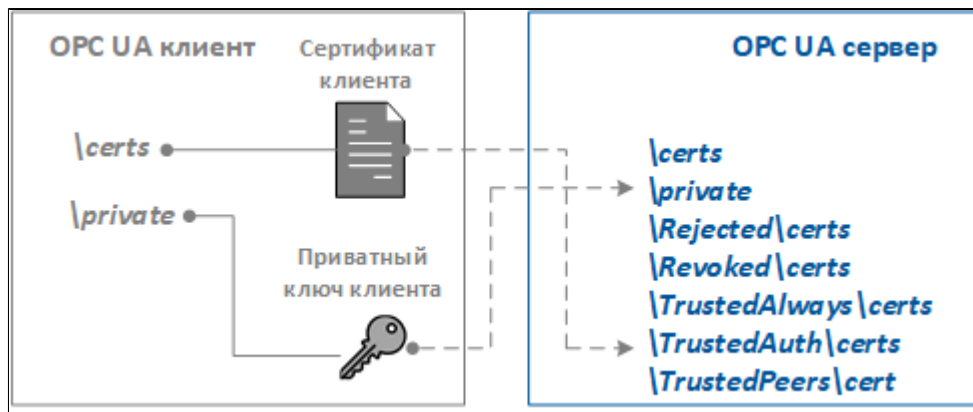
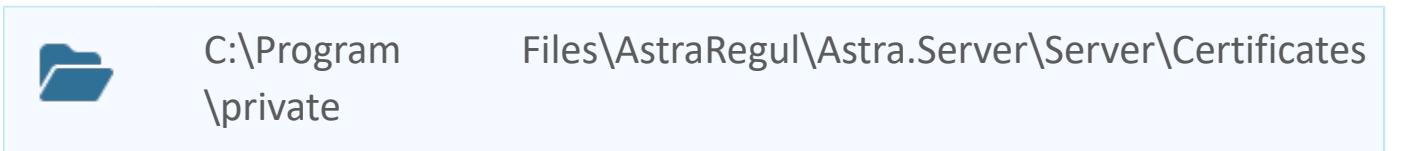
Чтобы аутентификация выполнялась с помощью сертификата пользователя и приватного ключа, установите параметру Аутентификация пользователя сертификатом значение Разрешить, а параметру Режим безопасности сетевого канала установите значение SignAndEncrypt.

Разместите сертификат и приватный ключ клиента на стороне сервера:

› сертификат - файл DER в папке:



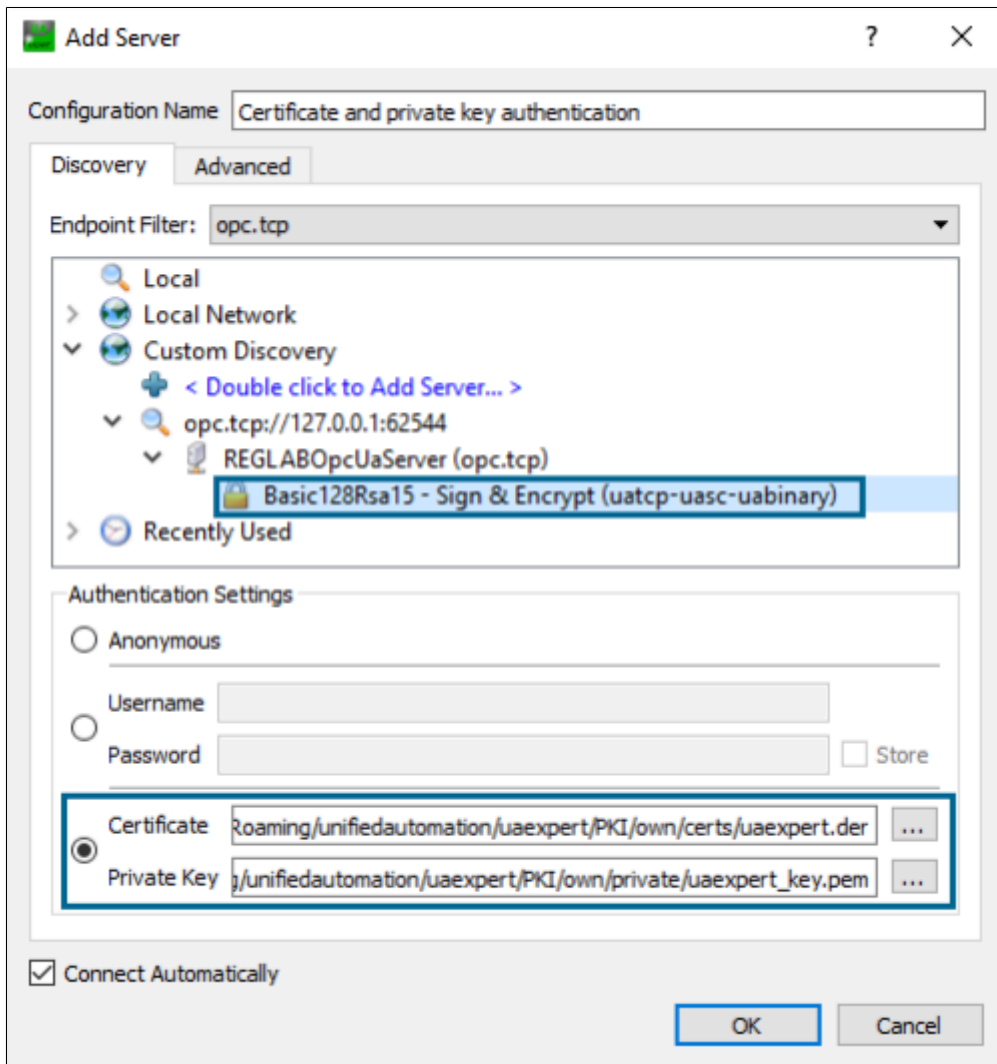
› приватный ключ - файл PEM в папке:



Сертификат и приватный ключ клиента UA Expert хранятся в папках:  
%appdata%\unifiedautomation\uaexpert\PKI\own\certs  
%appdata%\unifiedautomation\uaexpert\PKI\own\private

Чтобы подключиться к серверу с помощью сертификата и приватного ключа, в клиенте UA Expert в поле Certificate укажите путь к файлу сертификата (файл DER), в поле Private Key укажите путь к файлу приватного ключа (файл PEM).





## 1.1.2.4.6.2.1. Работа с клиентскими сертификатами

Для установки безопасного соединения ([режимы безопасности](#) Sign и SignAndEncrypt) клиент и сервер обмениваются сертификатами. Для хранения сертификатов на сервере используется папка:

Для ОС Windows:



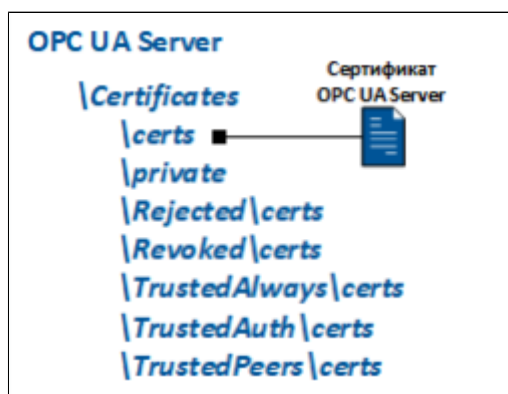
C:\Program Files\AstraRegul\Astra.Server\Server\Certificates

Для ОС Linux:



/opt/AstraRegul/Astra.Server/Certificates

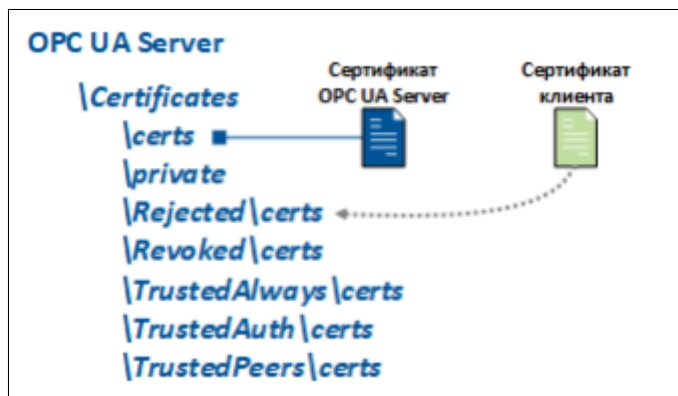
Сертификат сервера хранится в папке **certs**.



Сертификаты клиентов при обращении к серверу попадают в папку неподобренных сертификатов



C:\Program Files\AstraRegul\Astra.Server\Server\Certificates\Rejected\cert



Клиенты с неподобренными сертификатами **не имеют права** подключаться к серверу.

## Одобрение сертификата

Чтобы одобрить сертификат и разрешить обмен данными с клиентом, переместите сертификат клиента из папки неподобренных сертификатов:



C:\Program Files\AstraRegul\Astra.Server\Server\Certificates\Rejected\cert

В папку одобренных сертификатов:



C:\Program Files\AstraRegul\Astra.Server\Server\Certificates\TrustedPeers\cert

## Одобрение истекшего сертификата

Чтобы одобрить сертификат с истекшим сроком действия и разрешить обмен данными с клиентом, переместите сертификат клиента из папки одобренных сертификатов:



C:\Program Files\AstraRegul\Astra.Server\Server\Certificates\TrustedPeers\cert

В папку просроченных сертификатов:



C:\Program  
\CertificatesTrustedAlways\cert

Files\AstraRegul\Astra.Server\Server

При предоставлении клиентом нового сертификата удалите старый сертификат из папки



C:\Program  
\CertificatesTrustedAlways\cert

Files\AstraRegul\Astra.Server\Server

## Временный запрет подключения

Чтобы временно запретить клиенту подключаться к серверу, переместите его сертификат из папки одобренных сертификатов:



C:\Program  
\TrustedPeers\cert

Files\AstraRegul\Astra.Server\Server\Certificates

или папки просроченных сертификатов:



C:\Program  
\CertificatesTrustedAlways\cert

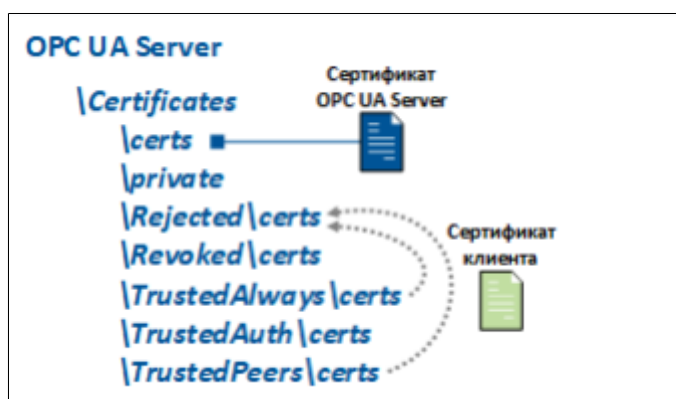
Files\AstraRegul\Astra.Server\Server

в папку неподобренных сертификатов:



C:\Program  
\cert

Files\AstraRegul\Astra.Server\Server\Certificates\Rejected

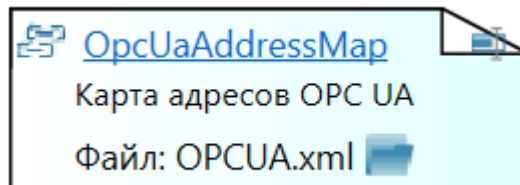


Для снятия запрета переместите сертификат обратно.

## 1.1.2.4.6.3. Карта адресов

Для настройки сигналов модуля OPC UA Сервер используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:


1. Добавьте карту адресов OPC UA в исполняемое приложение;




2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	Адресное пространство	Тип идентифи	Идентификатор узла	Позиция в мас	Отправлять ка	Отправлять ме	Категория дан
string	string	не привязан							
string1	string	не привязан							
bool	bool	не привязан							
uint2	uint2	не привязан							

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>› Непосредственно;</li><li>› Только чтение;</li><li>› Не привязан.</li></ul>
Адресное пространство	Адресное пространство группы серверов. <div style="border: 1px solid #add8e6; padding: 5px; margin: 5px 0;"> Если для группы серверов не настроена таблица адресного пространства, то URL-путь подключения к OPC UA серверу следует вводить вручную</div> Для контроллеров REGUL используется адресное пространство: urn:ProsoftSystems:regul_ua_server:iec_data
Идентификатор узла	Идентификатор узла в OPC UA сервере:

	 <p>Идентификатор узла записывается строго в том виде, в котором хранится в OPC UA сервере. Например, тег сигнала или численный идентификатор</p>
Тип идентификатора	<p>Тип идентификатора, указанного в параметре Идентификатор узла:</p> <ul style="list-style-type: none"> <li>› Строковый. Пример идентификатора: Items.Level;</li> <li>› Числовой. Пример идентификатора: 318;</li> <li>› Глобальный. Пример идентификатора: 1c35cb2b-e5d0-495a-93c9-da095c652230;</li> <li>› Скрытый. Пример идентификатора: d3571a</li> </ul>
Протокольный тип	<p>Направление передачи:</p> <ul style="list-style-type: none"> <li>› Входящий - сигнал используется для приёма данных с OPC UA сервера;</li> <li>› Исходящий – сигнал используется для записи данных в OPC UA сервер;</li> <li>› Корень поддерева – приём значений поддерева сигналов OPC UA сервера.</li> </ul>
Отправлять качество	Флаг устанавливается, если требуется записывать качество сигнала в OPC UA сервер
Отправлять метку времени	Флаг устанавливается, если требуется записывать метку времени сигнала в OPC UA сервер
Позиция в массиве	Номер позиции, значение которой будет записано в сигнал при получении от OPC сервера массива значений
Категория данных	Отнесение сигнала к категориям данных

## 1.1.2.4.6.4. Диагностика работы

### Журнал работы

Модуль OPC UA ведёт журнал работы, в который записывается информация о работе модуля и обмене данными с клиентами.



Чтобы модуль вёл журнал работы, в общих параметрах модуля установите параметру Вести журнал работы модуля значение Да или установите сервисному сигналу модуля `FrameLogEnable.Set` значение `true`.

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:



`C:\Program Files\AstraRegul\Astra.Server\Logs;`

» в Linux системах в директории:



`/opt/AstraRegul/Astra.Server/Logs.`

Для просмотра журнала работы модуля используется сервисное приложение Просмотрщик лога кадров.

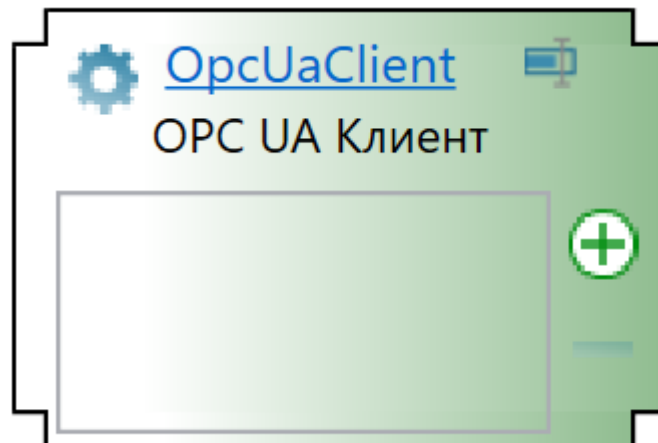
№	Дата	Время	Описание
4	27.04.2023	11:58:30:484	Добавлен основной поток планировщика задач (ThreadId: 12c4, TotalThreadCount: 2).
5	27.04.2023	11:58:30:485	Добавлен основной поток планировщика задач (ThreadId: 47c8, TotalThreadCount: 3).
6	27.04.2023	11:58:30:485	Добавлен основной поток планировщика задач (ThreadId: 2fd8, TotalThreadCount: 4).
7	27.04.2023	11:58:30:485	Планировщик задач запущен (количество потоков по приоритету задач: высокий - 1, средний - 1, низкий - 2; восп
8	27.04.2023	11:58:35:877	(opc.tcp://DEV70-24-RUB:62544) Установлено соединение по каналу передачи данных (ChannelId: 1).
9	27.04.2023	11:58:35:879	(opc.tcp://DEV70-24-RUB:62544) Установлено соединение по каналу передачи данных (ChannelId: 2).
10	27.04.2023	11:58:35:884	(opc.tcp://DEV70-24-RUB:62544#2, opc.tcp://172.16.145.219:61856) Не удалось удалить сессию. Сессия не привязан.
11	27.04.2023	11:58:35:885	(opc.tcp://DEV70-24-RUB:62544) Закрыт канал передачи данных (ChannelId: 2).
12	27.04.2023	11:58:35:888	(opc.tcp://DEV70-24-RUB:62544) Установлено соединение по каналу передачи данных (ChannelId: 3).
13	27.04.2023	11:58:35:897	(opc.tcp://DEV70-24-RUB:62544#3, opc.tcp://172.16.145.219:61857, SessionId: ns=4;g=cdd2c664-5337-6bc7-3232-7ef.
14	27.04.2023	11:58:35:898	(opc.tcp://DEV70-24-RUB:62544#3, opc.tcp://172.16.145.219:61857, SessionId: ns=4;g=cdd2c664-5337-6bc7-3232-7ef.
15	27.04.2023	11:58:35:898	(opc.tcp://DEV70-24-RUB:62544#3, opc.tcp://172.16.145.219:61857, SessionId: ns=4;g=cdd2c664-5337-6bc7-3232-7ef.
16	27.04.2023	11:58:35:903	(opc.tcp://DEV70-24-RUB:62544#3, opc.tcp://172.16.145.219:61857, SessionId: ns=4;g=cdd2c664-5337-6bc7-3232-7ef.
17	27.04.2023	11:58:35:905	(opc.tcp://DEV70-24-RUB:62544#3, opc.tcp://172.16.145.219:61857, SessionId: ns=4;g=cdd2c664-5337-6bc7-3232-7ef.
18	27.04.2023	11:58:35:906	(opc.tcp://DEV70-24-RUB:62544#3, opc.tcp://172.16.145.219:61857, SessionId: ns=4;g=cdd2c664-5337-6bc7-3232-7ef.
19	27.04.2023	11:58:35:909	(opc.tcp://DEV70-24-RUB:62544#3, opc.tcp://172.16.145.219:61857, SessionId: ns=4;g=cdd2c664-5337-6bc7-3232-7ef.
20	27.04.2023	11:58:40:483	(opc.tcp://DEV70-24-RUB:62544) Закрыт канал передачи данных (ChannelId: 1).

(opc.tcp://DEV70-24-RUB:62544#3,  
opc.tcp://172.16.145.219:61857,  
SessionId: ns=4;g=cdd2c664-5337-6bc7-  
3232-7ef3e7de5911) Сессия создана  
(AuthTokenId: b=p2yум0  
+e10Fcy2xuLBJEF/r6Jvfn0y7TMIbFeb5cSKdY

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---



## 1.1.2.4.7. OPC UA Клиент



Модуль OPC UA Клиент – коммуникационный модуль, предназначенный для обмена данными между Astra.Server и сторонними серверами по спецификации OPC UA.

Функции модуля OPC UA Клиент:

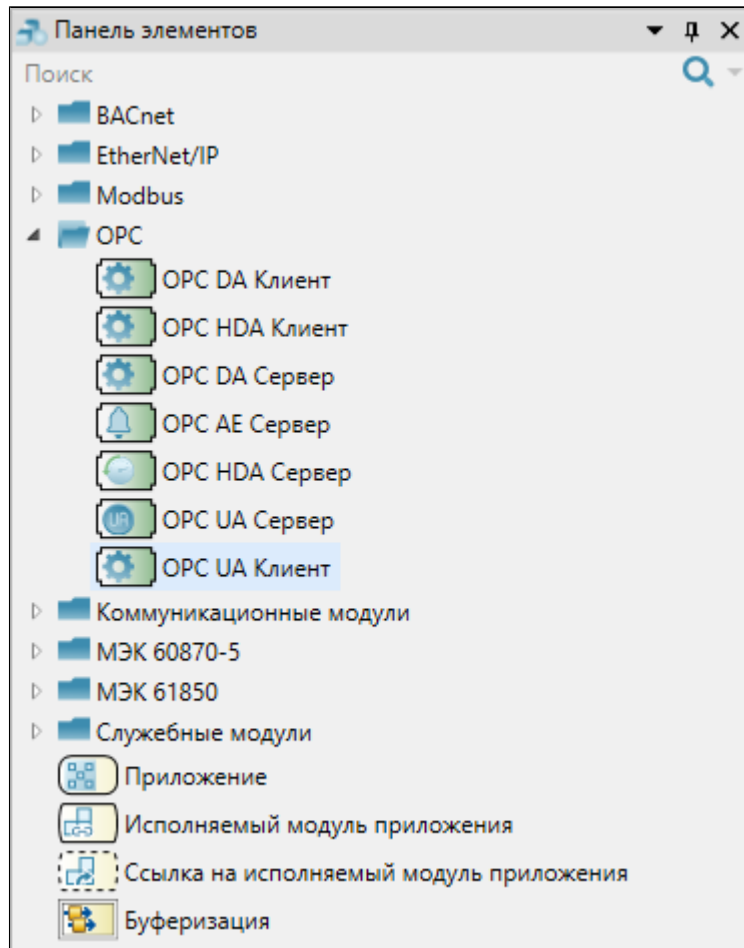
- › приём и запись значений сигналов OPC UA сервер;
- › приём значений поддерева сигналов OPC UA сервера;
- › приём и запись значений элементов массива OPC UA сервера.

Обмен данными по спецификации OPC UA (Unified Architecture) в сравнении со спецификацией OPC DA (Data Access) имеет следующие преимущества:

- › полностью кроссплатформенный стандарт;
- › отказ от закрытого стандарта COM/DCOM в пользу TCP;
- › простота настроек удаленного подключения;
- › безопасность передаваемых данных благодаря использованию шифрования и аутентификации;
- › оперативные данные, исторические данные, события передаются через один сервер по единому интерфейсу.

## 1.1.2.4.7.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Одновременно в составе конфигурации Astra.Server может функционировать несколько экземпляров модуля OPC UA Client.

Выделите модуль, чтобы настроить его свойства.

Свойства	
OpcUaClient OPC UA Клиент	
Общие	
Имя приложения	REGLABOpcUaClient
Имя	OpcUaClient
Параметры модуля	
Активность	Да
Отображаемое имя	
Параметры журналирования	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения

## Общие

Параметр	Описание
Имя	Имя модуля
Имя приложения	Значение по умолчанию: REGLABOpcUaClient

## Параметры модуля

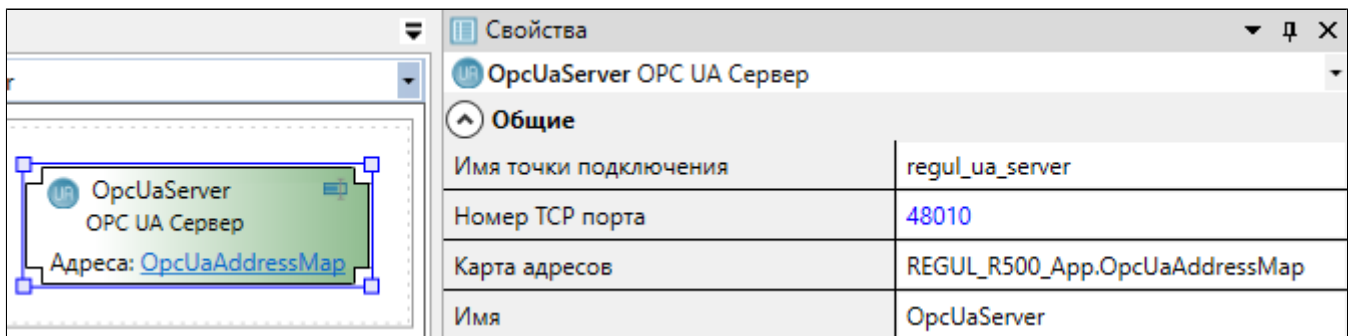
Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен;</li> <li>&gt; Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

# Настройка сервера

Для настройки сервера укажите параметры подключения.



Параметр	Описание
Имя сервера	Имя сервера
Имя точки подключения	Зависит от типа UA сервера, с которым будет устанавливаться соединение
Номер TCP порта	TCP порт для обмена данными с OPC UA сервером
Карта адресов	Карта адресов OPC UA сервера

Объединение вышеуказанных параметров дает URL-путь подключения к серверу.

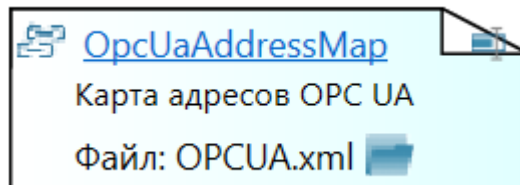


URL-путь подключения к серверу:  
opc.tcp://localhost:48010/regul\_ua\_server

## 1.1.2.4.7.3. Карта адресов

Для настройки сигналов модуля OPC UA Клиент используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:


1. Добавьте карту адресов OPC UA в исполняемое приложение;




2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	Адресное пространство	Тип идентифи	Идентификатор узла	Позиция в мас	Отправлять ка	Отправлять ме	Категория дан
string	string	не привязан							
string1	string	не привязан							
bool	bool	не привязан							
uint2	uint2	не привязан							

## Параметры карты адресов

Параметр	Значение
Привязка	Привязка адреса к сигналу: <ul style="list-style-type: none"><li>› Непосредственно;</li><li>› Только чтение;</li><li>› Не привязан.</li></ul>
Адресное пространство	Адресное пространство группы серверов. <div style="border: 1px solid #add8e6; padding: 5px;"> Если для группы серверов не настроена таблица адресного пространства, то URL-путь подключения к OPC UA серверу следует вводить вручную</div> Для контроллеров REGUL используется адресное пространство: urn:ProsoftSystems:regul_ua_server:iec_data
Идентификатор узла	Идентификатор узла в OPC UA сервере:

	 <p>Идентификатор узла записывается строго в том виде, в котором хранится в OPC UA сервере. Например, тег сигнала или численный идентификатор</p>
Тип идентификатора	<p>Тип идентификатора, указанного в параметре Идентификатор узла:</p> <ul style="list-style-type: none"> <li>› Строковый. Пример идентификатора: Items.Level;</li> <li>› Числовой. Пример идентификатора: 318;</li> <li>› Глобальный. Пример идентификатора: 1c35cb2b-e5d0-495a-93c9-da095c652230;</li> <li>› Скрытый. Пример идентификатора: d3571a</li> </ul>
Протокольный тип	<p>Направление передачи:</p> <ul style="list-style-type: none"> <li>› Входящий - сигнал используется для приёма данных с OPC UA сервера;</li> <li>› Исходящий – сигнал используется для записи данных в OPC UA сервер;</li> <li>› Корень поддерева – приём значений поддерева сигналов OPC UA сервера.</li> </ul>
Отправлять качество	Флаг устанавливается, если требуется записывать качество сигнала в OPC UA сервер
Отправлять метку времени	Флаг устанавливается, если требуется записывать метку времени сигнала в OPC UA сервер
Позиция в массиве	Номер позиции, значение которой будет записано в сигнал при получении от OPC сервера массива значений
Категория данных	Отнесение сигнала к категориям данных

# Диагностика работы

## Служебные сигналы

Модуль OPC UA Client динамически создаёт служебные сигналы контроля состояния серверов, а также стандартные для модулей Astra.Server служебные сигналы для контроля и управления основными параметрами модуля.

Полный тег служебных сигналов контроля серверов:

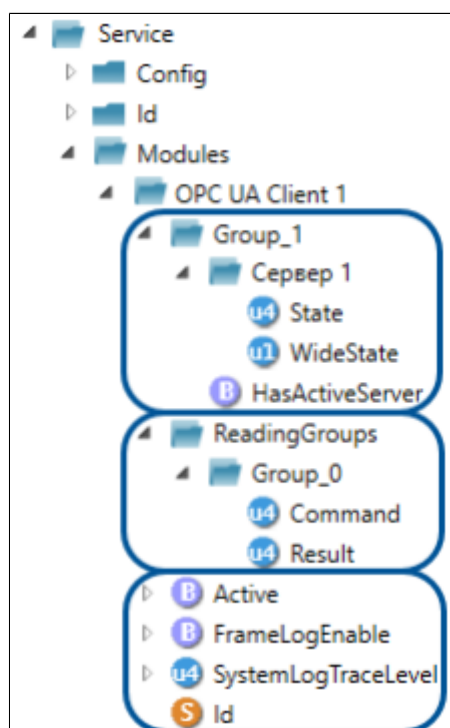


Service.Modules.<Идентификатор модуля>.<Название группы серверов>.<Имя сигнала>

Полный тег стандартных служебных сигналов имеет вид:



Service.Modules.<Идентификатор модуля>.<Имя сигнала>



## Служебные сигналы контроля серверов

Сигнал	Тип	Описание сигнала
State	Uint4	Состояние сервера: <ul style="list-style-type: none"><li>› 0 - не подключен;</li><li>› 1 - подключается;</li><li>› 2 - подключен, в резерве;</li><li>› 3 - подключен, в работе;</li><li>› 4 - отключается.</li></ul>
WideState	Uint1	Активность сервера: <ul style="list-style-type: none"><li>› 2 - не подключен;</li><li>› 4 - подключен, в резерве;</li><li>› 8 - подключен, в работе, неактивный (выставляется, если сигнал State = 3, но в качестве активного выбран другой сервер из резервной пары);</li><li>› 9 - подключен, в работе, активный (выставляется если State = 3 и сервер выбран в качестве активного сервера резервной пары).</li></ul>
HasActiveServer	Bool	Наличие активного сервера в группе: <ul style="list-style-type: none"><li>› True - есть активный сервер;</li><li>› False - нет активного сервера.</li></ul>

## Служебные сигналы опроса групп чтения

Сигнал	Тип	Описание сигнала
Command	Uint4	Команда опроса группы чтения: <ul style="list-style-type: none"><li>› «1» - получить значения сигналов группы.</li></ul> После завершения опроса группы сигналу устанавливается значение «0»
Result	Uint4	Результат выполнения опроса группы. Значение увеличивается на единицу после завершения опроса



## Стандартные служебные сигналы

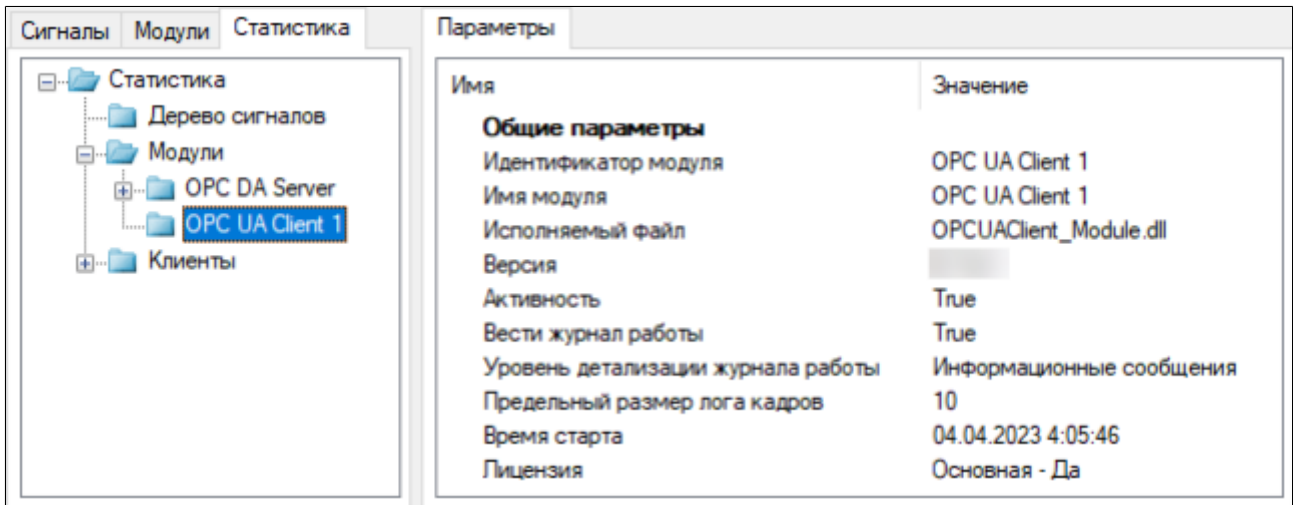
Сигнал	Тип	Описание сигнала
Active	Bool	Активность модуля: <ul style="list-style-type: none"><li>› True – запущен;</li><li>› False – остановлен.</li></ul> Соответствует значению параметра Активность. Управляется служебным сигналом Active.Set
FrameLogEnable	Bool	Ведение Журнала работы: <ul style="list-style-type: none"><li>› True – ведётся;</li><li>› False – не ведётся.</li></ul> Соответствует значению параметра Вести журнал работы модуля. Управляется служебным сигналом FrameLogEnable.Set
SystemLogTraceLevel	Uint4	Уровень детализации Журнала работы: <ul style="list-style-type: none"><li>› 1 – Предупреждения и аварийные сообщения;</li><li>› 2 – Информационные сообщения;</li><li>› 3 – Отладочные сообщения.</li></ul> Соответствует значению параметра Уровень трассировки в журнал приложений. Управляется служебным сигналом SystemLogTraceLevel.Set
Id	String	Идентификатор модуля в конфигурации Astra.Server

## Параметры статистики

Статистика работы модуля OPC UA Client отображается на вкладке Статистика сервисного приложения Конфигуратор, а также в сервисном приложении Статистика.

Для просмотра параметров статистики модуля подключитесь к Astra.Server приложением статистики и выберите в дереве модуль OPC UA Client.

Параметры статистики модуля:



## Общие параметры

Параметр	Описание
Идентификатор модуля	Идентификатор модуля в конфигурации Astra.Server
Имя модуля	Название модуля
Исполняемый файл	Имя исполняемого файла модуля в каталоге установки Astra.Server
Версия	Версия модуля OPC UA Client
Активность	Активность модуля
Вести журнал работы модуля	Ведение записи сообщений о работе модуля в журнал работы
Уровень детализации журнала работы	Типы сообщений, которые фиксируются в журнал приложений
Предельный размер лога кадров	Размер файла в мегабайтах для записи журнала работы модуля
Время старта	Время запуска модуля
Лицензия	Текущее состояние лицензирования модуля

# Журнал работы

Модуль OPC UA Client ведёт журнал работы, в который записывается информация о работе модуля и обмене данными с OPC UA сервером.



Чтобы модуль вёл журнал работы, в общих параметрах модуля установите параметру Вести журнал работы модуля значение Да или установите сервисному сигналу модуля FrameLogEnable.Set значение true.

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

» в Linux системах в директории:



/opt/AstraRegul//Astra.Server/Logs.

Для просмотра журнала работы модуля используется сервисное приложение Просмотрщик лога кадров.

№	Дата	Время	Описание	Группа	Сервер	№	Ter	Значение	Метка времени	Качество
873	04.04.2023	04:05:46:703	Пересмотренные сервером 'Сервер 2' параметры подписки:...	Group_1	Сервер 2	1	Input.Item1	0	04.04.2023 04:05:39:972	BAD_Wai
874	04.04.2023	04:05:46:704	Пересмотренные сервером 'Сервер 1' параметры подписки:...	Group_1	Сервер 1	2	Input.Item2	0	04.04.2023 04:05:39:972	BAD_Wai
875	04.04.2023	04:05:46:704	Group_1 Сервер 2: соединение установлено	Group_1	Сервер 2	3	Tree.Level	0	04.04.2023 04:05:39:972	BAD_Wai
876	04.04.2023	04:05:46:704	Group_1 Сервер 1: соединение установлено	Group_1	Сервер 1	4	Tree.Pressure	0	04.04.2023 04:05:39:972	BAD_Wai
877	04.04.2023	04:05:46:704	Состояние сервера 'Сервер 2' изменено на 'Подключен в ра...	Group_1	Сервер 2					
878	04.04.2023	04:05:46:704	Состояние сервера 'Сервер 1' изменено на 'Подключен в ра...	Group_1	Сервер 1					
879	04.04.2023	04:05:46:704	Состояние сервера 'Сервер 2' изменено на 'Подключен в ра...	Group_1	Сервер 2					
880	04.04.2023	04:05:46:704	В группе 'Group_1' выбран активный сервер Сервер 2	Group_1	Сервер 2					
881	04.04.2023	04:05:47:699	Получено уведомление от сервера. Количество значений = 4	Group_1	Сервер 2					
882	04.04.2023	04:05:47:699	Значения прочитаны, количество: 4	Group_1	Сервер 2					
883	04.04.2023	04:06:43:699	Получено уведомление от сервера. Количество значений = 2	Group_1	Сервер 2					
884	04.04.2023	04:06:43:699	Значения прочитаны, количество: 2	Group_1	Сервер 2					
885	04.04.2023	04:06:50:699	Получено уведомление от сервера. Количество значений = 2	Group_1	Сервер 2					
886	04.04.2023	04:06:50:699	Значения прочитаны, количество: 2	Group_1	Сервер 2					
887	04.04.2023	04:06:58:368	Асинхронная запись в сервер. Количество значений = 1	Group_1	Сервер 2					

№	Ter	Значение	Метка времени	Качество
1	Input.Item1	0	04.04.2023 04:05:39:972	BAD_Wai
2	Input.Item2	0	04.04.2023 04:05:39:972	BAD_Wai
3	Tree.Level	0	04.04.2023 04:05:39:972	BAD_Wai
4	Tree.Pressure	0	04.04.2023 04:05:39:972	BAD_Wai

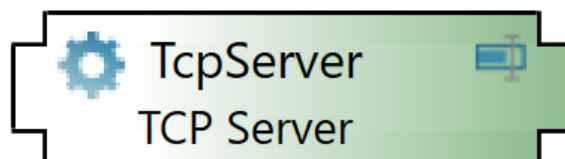
Значения прочитаны, количество: 4

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

## 1.1.2.5. Коммуникационные модули

Модуль	Описание
<a href="#">TCP Server</a>	Серверный TCP интерфейс для работы с данными
<a href="#">Модуль NetDiag</a>	Модуль диагностики связи с сетевыми устройствами в сетях TCP/IP
<a href="#">Модуль NetDiag2</a>	Модуль диагностики связи с сетевыми устройствами в сетях TCP/IP
<a href="#">Менеджер SNMP</a>	Коммуникационный модуль для опроса агентов по протоколу SNMP
<a href="#">HUB Модуль</a>	Концентратор данных
<a href="#">SQL Connentor</a>	Модуль взаимодействия с источниками данных с помощью SQL
<a href="#">Клиент FINS</a>	Обмен данными с программируемым контроллером компании Omron по транспортному протоколу UDP
<a href="#">Клиент ПОРТАЛ</a>	Модуль, импортирующий значения из системы ПОРТАЛ
<a href="#">Siemens S7 Client</a>	Обмен данными с программируемым контроллером Siemens S7 PLC по протоколу S7
<a href="#">Syslog Сервер</a>	Модуль сбора сообщений Syslog
<a href="#">Syslog Клиент</a>	Модуль отправки сообщений Syslog
<a href="#">ТЭМ-104</a>	Сбор данных из систем теплоснабжения

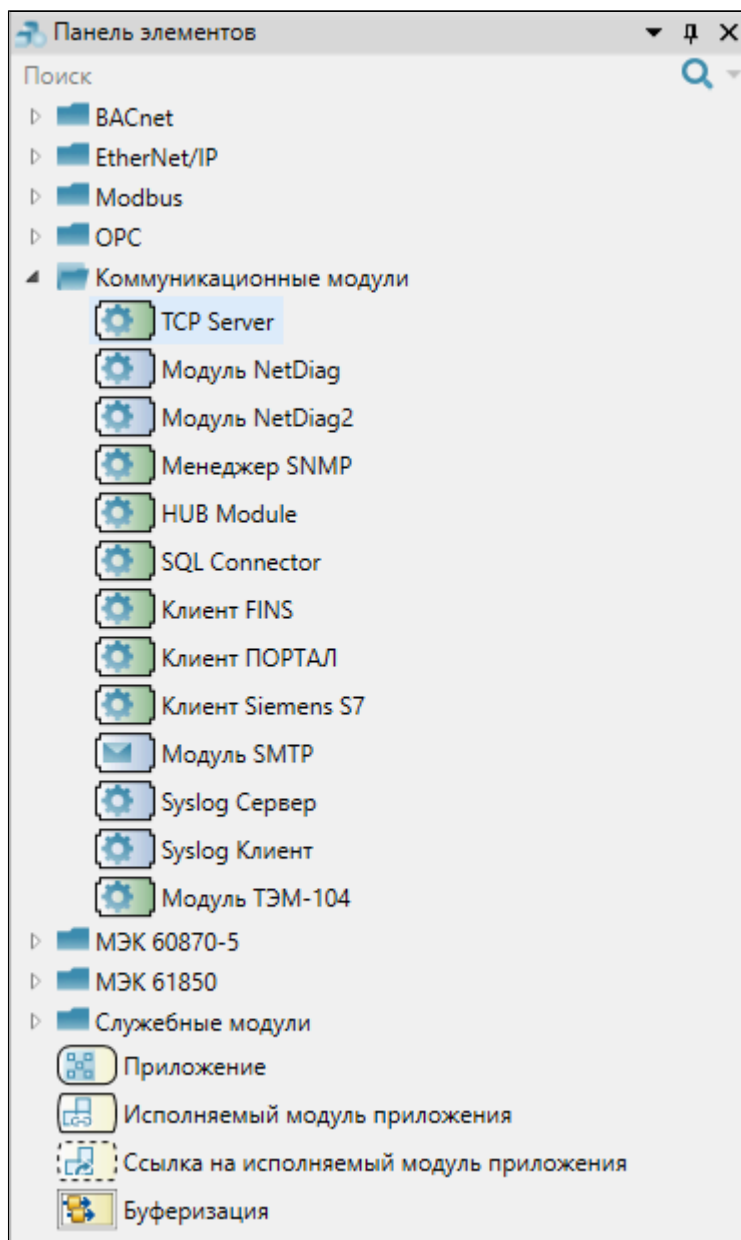
## 1.1.2.5.1. TCP Server



Модуль TCP Server Module предоставляет клиентам данные Astra.Server по протоколу TCP. Потребителем данных может быть Astra.AccessPoint или модуль HUB в составе другого Astra.Server.

# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

Свойства	
TcpServer TCP Server	
<b>Параметры модуля</b>	
Номер TCP порта	4388
Разрешить изменения для анонимных клиентов	Да
Очередь данных	500000
Очередь сообщений	100000
Таймаут очистки застоявшихся читателей, мин.	1
Активность	Да
Отображаемое имя	
<b>Настройки доступа к истории</b>	
Номер TCP порта	4950
Время удержания сессии	10
Ограничение ожидающих сессий	64
<b>Настройки файлового интерфейса</b>	
Период генерации файлов, мс	5000
Максимальное количество изменений в файле	10000
Период записи всех данных, сек.	60
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	TcpServer

Для обмена данными с Astra.Server по протоколу TCP, настройте параметр модуля Номер TCP порта. Данный порт будет использоваться для подключения клиентов к Astra.Server по TCP-протоколу. В качестве клиентов может выступать Astra.AccessPoint или модуль HUB в составе другого Astra.Server.



Данные между Astra.AccessPoint и Astra.Server передаются в зашифрованном виде. Для шифрования данных используются собственные алгоритмы ООО "РЕГЛАБ".

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Номер TCP порта	Номер TCP порта для подключения к Astra.Server по TCP-протоколу
Разрешить изменения для анонимных клиентов	Значения: <ul style="list-style-type: none"><li>› Да</li><li>› Нет</li></ul>
Очередь данных	Максимальный размер очередей данных
Очередь сообщений	Максимальный размер очередей сообщений
Таймаут очистки застоявшихся читателей	Время с момента последней операции, после которого читатель будет принудительно освобожден.
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"><li>› Да – модуль запущен;</li><li>› Нет – модуль остановлен.</li></ul>
Отображаемое имя	Отображаемое имя модуля

В очереди данных накапливаются изменения значений сигналов или сигнальных свойств для последующей отправки уведомлений клиентам. Параметр Максимальный размер очередей данных позволяет установить лимит на количество элементов в этой очереди.



В очереди сообщений накапливаются сообщения о событиях для отправки уведомлений клиентам. Параметр Максимальный размер очереди сообщений позволяет установить лимит на количество элементов в этой очереди.



При полном заполнении очереди поступающие в неё данные начнут отбрасываться, а в журнал работы модуля начнут поступать сообщения о переполнении очереди.

## Настройки доступа к истории

Параметр	Описание
Номер TCP порта	Номер TCP порта для доступа к истории
Время удержания сессии	Время удержания ожидающей подключения сессии, секунд
Ограничение ожидающих сессий	Количество возможных одновременно удерживаемых сессий в режиме ожидания подключения

## Настройки файлового интерфейса

Параметр	Описание
Период генерации файлов, мс	Период, не чаще которого будут записываться файлы с данными
Максимальное количество изменений в файле	Значение по умолчанию: 10000
Период записи всех данных, сек	Период принудительной записи всех данных в файл вне зависимости от того, изменились ли их значения

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

# Диагностика работы модуля

## Журнал работы модуля

Для более детального анализа работы модуля воспользуйтесь сервисным приложением Просмотрщик лога кадров.

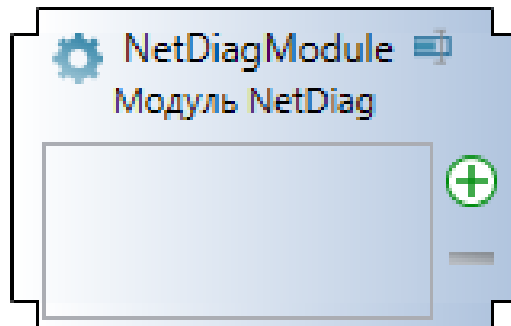
№	Дата	Время	Описание
1	29.06.2023	20:40:03:005	Журнал открыт для записи
2	29.06.2023	20:40:03:091	Подключился клиент (#2 127_0_0_1:60139)
3	29.06.2023	20:40:03:195	Подключился клиент (#3 127_0_0_1:60141)
4	29.06.2023	20:40:03:659	Подключился клиент (#4 127_0_0_1:60144)
5	29.06.2023	20:40:03:998	Подключился клиент (#5 127_0_0_1:60145)
6	29.06.2023	20:40:04:853	#2 127_0_0_1:60139 SetMsgListner2: position - Данные для.Привязка 1
7	29.06.2023	20:40:04:853	#2 127_0_0_1:60139 SetMsgListner2: Failed - Узла Данные для.Привязка...
8	29.06.2023	20:40:04:915	#3 127_0_0_1:60141 SetMsgListner2: position - Данные для.Привязка 1
9	29.06.2023	20:40:04:915	#3 127_0_0_1:60141 SetMsgListner2: Failed - Узла Данные для.Привязка...

#2 127\_0\_0\_1:60139 SetMsgListner2:  
position - Данные для.Привязка 1

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

< >

## 1.1.2.5.2. Модуль NetDiag



Модуль NetDiag предназначен для диагностики связи с сетевыми устройствами в сетях TCP/IP.

Модуль NetDiag записывает информацию в динамические сигналы сервера. Динамические сигналы создаются при запуске сервера и не конфигурируются.

Модуль выполняет следующие функции:

- › проверка возможности доставки IP-пакетов до сетевого устройства (Ping);
- › определение маршрута следования IP-пакетов до сетевого устройства (TraceRoute).

### Сигналы модуля

Сигналы модуля используются для:

- › записи информации о запросах, отправленных сетевым устройствам по каждому каналу;
- › управления запросами, отправляемыми сетевым устройствам по каждому каналу.

Сигналы модуля NetDiag динамические: их создаёт модуль при старте сервера в папке, указанной в параметрах модуля. Структура папки:

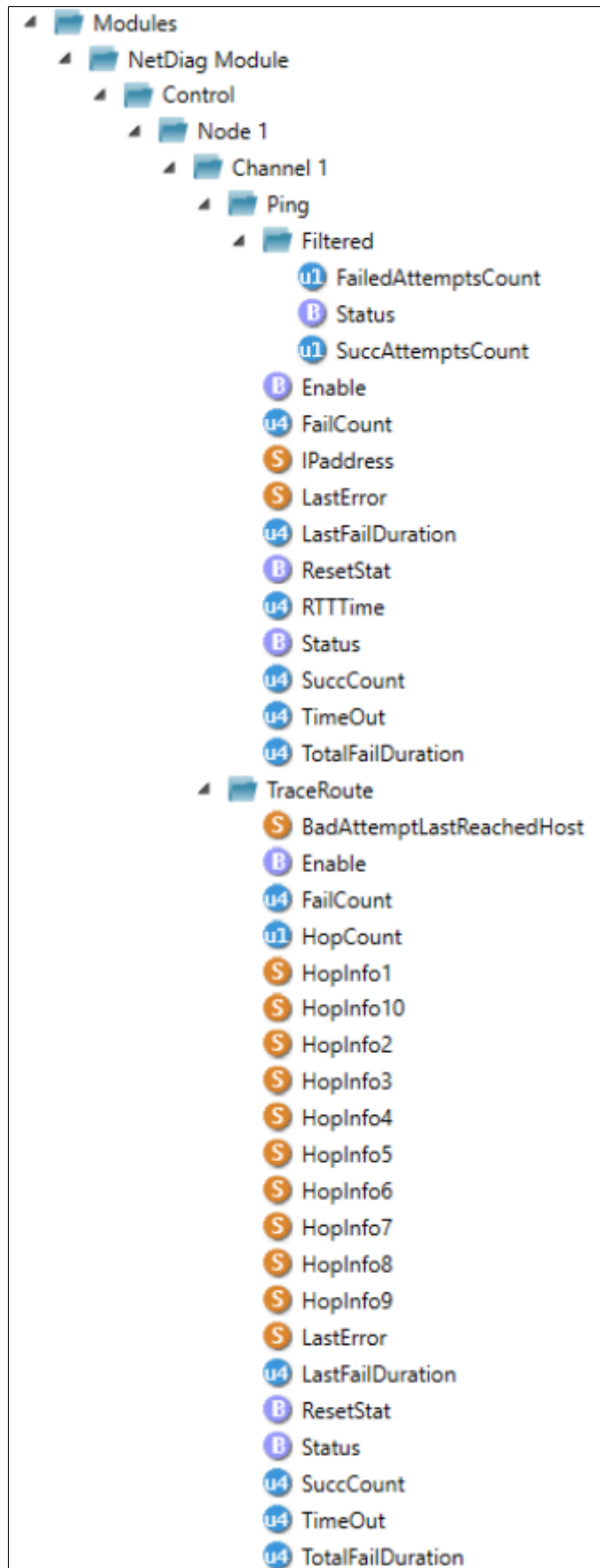
- › в папке для каждого сетевого устройства создаётся папка;
- › имя папки – псевдоним сетевого устройства;
- › в папке сетевого устройства для каждого канала создаётся папка; имя папки – псевдоним канала;

› в папке канала создаются две папки:

› «Ping»;

› «TraceRoute».

› в папках «Ping» и «TraceRoute» создаются сигналы модуля; имя сигнала – название функции, которую он выполняет (описание функций приведено ниже).



## Принципы работы

Модуль периодически отправляет сетевым устройствам запросы Ping и TraceRoute.

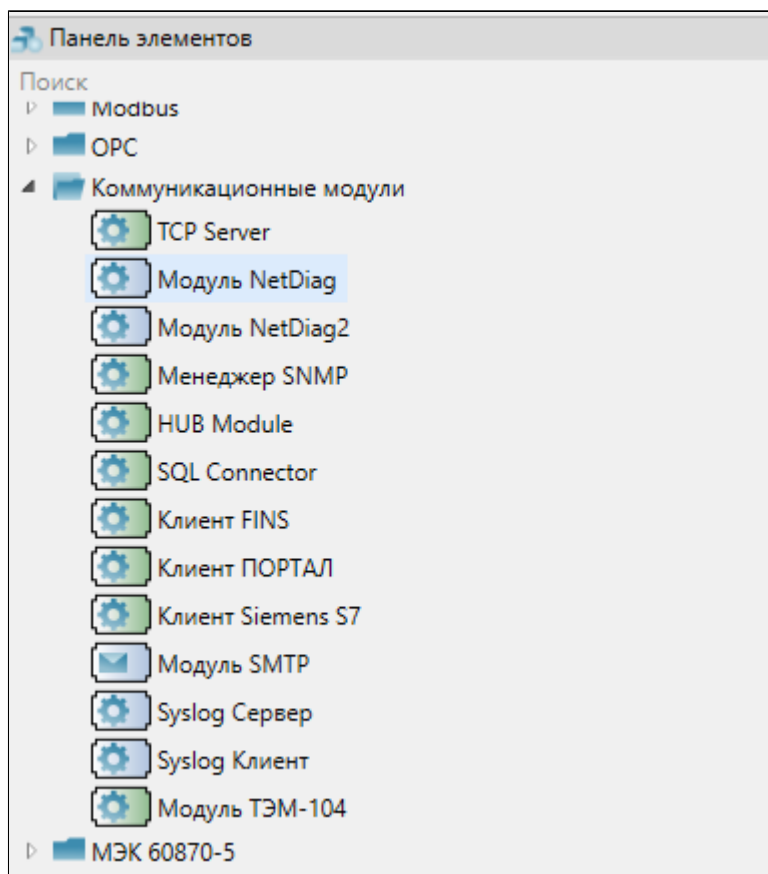
Запросы передаются согласно протоколу ICMPv4, период отправления запросов и список сетевых устройств указываются при конфигурировании модуля. Результаты запросов модуль записывает в сигналы сервера.

## **Работа в резерве**

В режиме РЕЗЕРВ модули выполняют те же функции, что и в режиме РАБОТА.

## 1.1.2.5.2.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигулятора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.



NetDiagModule Модуль NetDiag	
<b>Параметры модуля</b>	
Имя папки сигналов модуля	Service.Modules.NetDiag Module.Control
Период диагностики Ping, мс	5000
Период диагностики TraceRoute, мс	10000
Настройки по умолчанию	
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	NetDiagModule

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Имя папки сигналов модуля	Папка с сигналами для модуля
Период диагностики Ping, мс	Промежуток времени между запросами для проверки наличия соединения с сетевым устройством в мс
Период диагностики TraceRoute, мс	Промежуток времени между запросами для определения маршрута следования данных до сетевого устройства в мс

Настройки умолчанию	по	Выбор файла с готовой конфигурацией
Активность		Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен;</li> <li>&gt; Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом Active.Set
Отображаемое имя		Отображаемое имя модуля

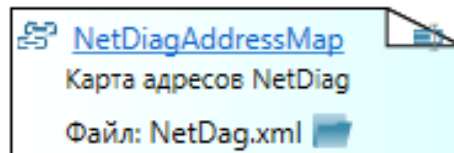
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.5.2.2. Карта адресов

Для настройки сигналов модуля NetDiag используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов NetDiag в исполняемое приложение;



2. Откройте редактор карты адресов.

	Сигнал	Тип	Привязка	Узел сети	Адаптер	Тип запроса	Функция сигнала	Номер
	▼	▼	▼	▼	▼	▼	▼	▼
	bool	bool	непосредстве	host	EthernetAdapte	Ping	Enable	
	uint1	uint1	непосредстве	host	EthernetAdapte	Ping	Filtered.SuccAtt	

## Параметры карты адресов

Параметр	Описание
Узел сети	Сетевое устройство. Выбирается из списка устройств, добавленных модулю
Адаптер	Сетевой адаптер. Выбирается из списка адаптеров выбранного сетевого устройства
Тип запроса	Тип запроса: <ul style="list-style-type: none"><li>› Ping</li><li>› TraceRoute</li></ul>
Функция сигнала	Функция сигнала. Список доступных функций зависит от типа сигнала и выбранного режима
Номер	Номер промежуточного узла в диапазоне от 1 до 255. Указывается только если выбран режим TraceRoute и функция HopInfo

## 1.1.2.5.2.3. Диагностика работы модуля

### Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:

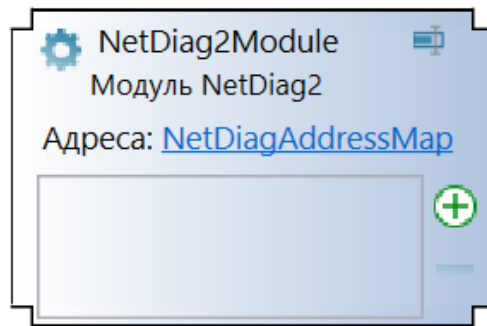


C:\Program Files\AstraRegul\Astra.Server\Logs;

Для просмотра журнала работы модуля воспользуйтесь сервисным приложением [Просмотрщик лога кадров](#).

№	Дата	Время	Описание
0	06.03.2017	11:07:18:779	Журнал открыт для записи
1	06.03.2017	11:07:22:825	Node 2 '-', Невозможно получить адрес для имени '-'
2	06.03.2017	11:07:22:825	Node 2 '-', Невозможно получить адрес для имени '-'

## 1.1.2.5.3. Модуль NetDiag2



Модуль NetDiag2 предназначен для диагностики связи с сетевыми устройствами в сетях TCP/IP.

Модуль NetDiag2 записывает информацию в статические сигналы сервера. Статические сигналы создаются и конфигурируются пользователем.

Модуль NetDiag2 следует использовать, если нужно конфигурировать сигналы, используемые модулем: разместить сигналы в разных ветвях дерева сигналов и/или конфигурировать свойства сигналов.

Модуль выполняет следующие функции:

- › проверка возможности доставки IP-пакетов до сетевого устройства (Ping);
- › определение маршрута следования IP-пакетов до сетевого устройства (TraceRoute).

### Сигналы модуля

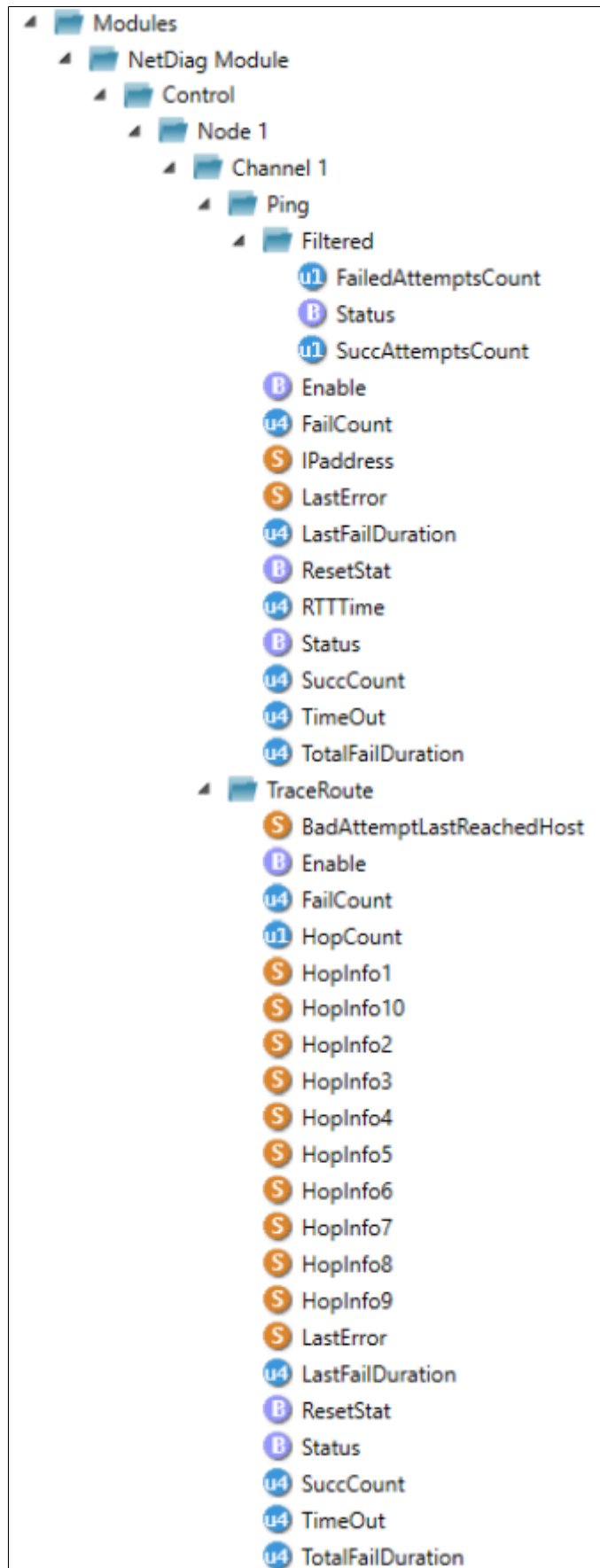
Сигналы модуля используются для:

- › записи информации о запросах, отправленных сетевым устройствам по каждому каналу;
- › управления запросами, отправляемыми сетевым устройствам по каждому каналу.

Сигналы модуля NetDiag динамические: их создаёт модуль при старте сервера в папке, указанной в параметрах модуля. Структура папки:

- › в папке для каждого сетевого устройства создаётся папка;

- › имя папки – псевдоним сетевого устройства;
- › в папке сетевого устройства для каждого канала создаётся папка; имя папки – псевдоним канала;
- › в папке канала создаются две папки:
  - › «Ping»;
  - › «TraceRoute».
- › в папках «Ping» и «TraceRoute» создаются сигналы модуля; имя сигнала – название функции, которую он выполняет (описание функций приведено ниже).



## Принципы работы

Модуль периодически отправляет сетевым устройствам запросы Ping и TraceRoute.

Запросы передаются согласно протоколу ICMPv4, период отправления запросов и список сетевых устройств указываются при конфигурировании модуля. Результаты запросов модуль записывает в сигналы сервера.

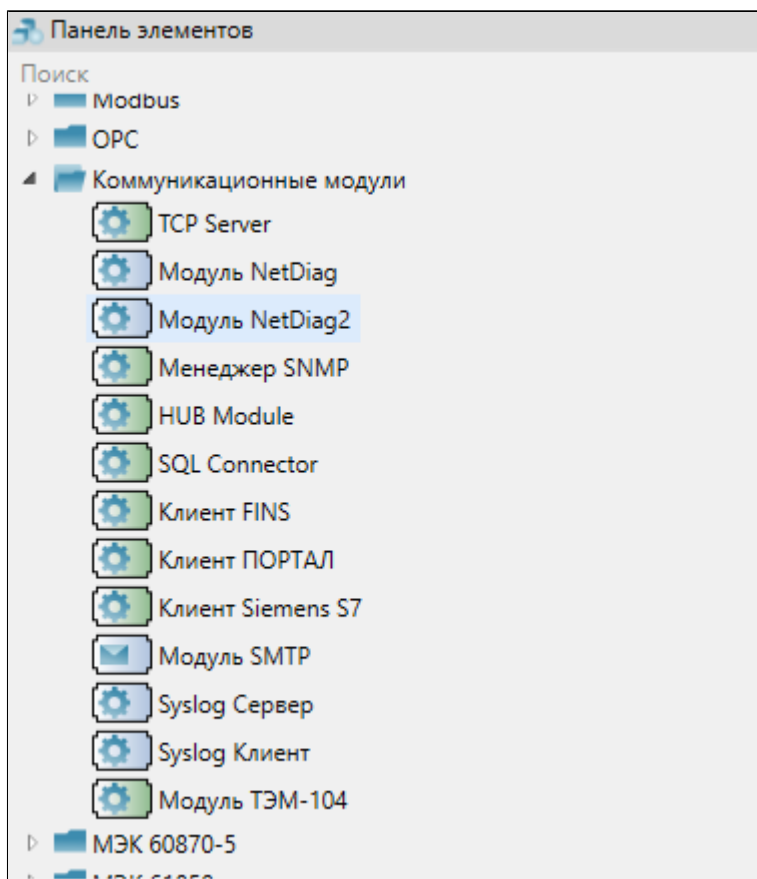
## **Работа в резерве**

В режиме РЕЗЕРВ модули выполняют те же функции, что и в режиме РАБОТА.



## 1.1.2.5.3.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигулятора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

NetDiag2Module Модуль NetDiag2	
<b>Общие</b>	
Карта адресов	SNMP.STATION_SnmpApp.NetDiagAddressMap
Имя	NetDiag2Module
<b>Параметры модуля</b>	
Период диагностики Ping, мс	5000
Период диагностики TraceRoute, мс	10000
Настройки по умолчанию	
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения

## Общие

Параметр	Описание
Имя	Имя модуля
Карта адресов	Карта адресов NetDiag, которая будет использована для данного адаптера

## Параметры модуля

Параметр	Описание
Период диагностики Ping, мс	Промежуток времени между запросами для проверки наличия соединения с сетевым устройством в мс
Период диагностики TraceRoute, мс	Промежуток времени между запросами для определения маршрута следования данных до сетевого устройства в мс

Настройки по умолчанию	Выбор файла с готовой конфигурацией
Активность	Активность модуля при запуске/перезапуске Astra.Server: > Да – модуль запущен; > Нет – модуль остановлен. Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

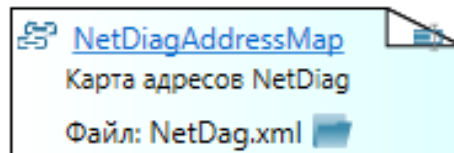
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.




## 1.1.2.5.3.2. Карта адресов

Для настройки сигналов модуля NetDiag используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов NetDiag в исполняемое приложение;



2. Откройте редактор карты адресов.

	Сигнал	Тип	Привязка	Узел сети	Адаптер	Тип запроса	Функция сигнала	Номер
	▼	▼	▼	▼	▼	▼	▼	▼
	bool	bool	непосредстве	host	EthernetAdapte	Ping	Enable	
	uint1	uint1	непосредстве	host	EthernetAdapte	Ping	Filtered.SuccAtt	

## Параметры карты адресов

Параметр	Описание
Узел сети	Сетевое устройство. Выбирается из списка устройств, добавленных модулю
Адаптер	Сетевой адаптер. Выбирается из списка адаптеров выбранного сетевого устройства
Тип запроса	Тип запроса: <ul style="list-style-type: none"><li>› Ping</li><li>› TraceRoute</li></ul>
Функция сигнала	Функция сигнала. Список доступных функций зависит от типа сигнала и выбранного режима
Номер	Номер промежуточного узла в диапазоне от 1 до 255. Указывается только если выбран режим TraceRoute и функция HopInfo

## 1.1.2.5.3.3. Диагностика работы модуля

### Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:

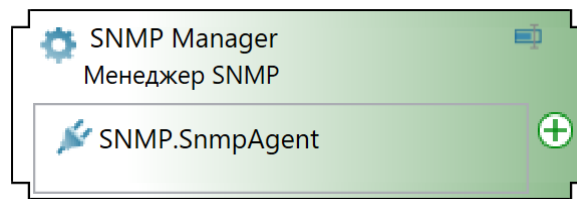


C:\Program Files\AstraRegul\Astra.Server\Logs;

Для просмотра журнала работы модуля воспользуйтесь сервисным приложением [Просмотрщик лога кадров](#).

№	Дата	Время	Описание
0	06.03.2017	11:07:18:779	Журнал открыт для записи
1	06.03.2017	11:07:22:825	Node 2 '-', Невозможно получить адрес для имени '-'
2	06.03.2017	11:07:22:825	Node 2 '-', Невозможно получить адрес для имени '-'

## 1.1.2.5.4. Менеджер SNMP



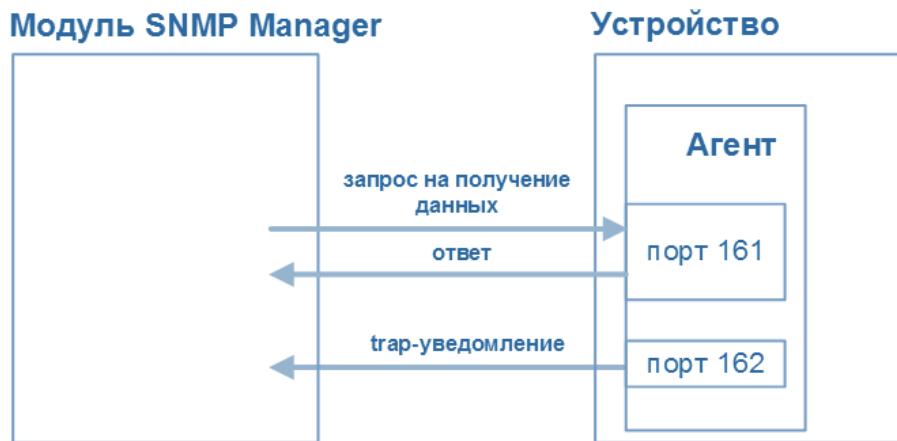
Модуль SNMP Manager предназначен для взаимодействия с устройствами в сети по протоколам SNMPv1, SNMPv2c и SNMPv3. Основными функциями модуля SNMP Manager являются удаленный мониторинг и контроль функционирования сетевых устройств (коммутатор, маршрутизатор, принтер и т.д.). Контроль и мониторинг устройств выполняется путем передачи агентом модулю SNMP Manager различных характеристик сетевого устройства (версия ОС, физический адрес, IP-адрес, объем свободной оперативной памяти и т.д.).

Модуль работает с устройствами, которые используют протокол TCP/IP. Схема передачи данных с одним сетевым устройством показана на рисунке ниже.

Модуль поддерживает одновременное соединение с несколькими агентами по нескольким каналам связи. Один агент получает данные только с одного сетевого устройства. Если связь потеряна по одному из каналов, выполняется переключение соединения на работающие резервные каналы. Канал, используемый модулем для передачи данных, называется активным.

Модуль SNMP Manager выполняет следующее:

- циклически отправляет агентам запросы;
- получает trap уведомления от агентов об изменении значений параметров.



Значение и качество сигнала, которые получены модулем во время опроса агентов, сравниваются со значением и качеством сигнала в ядре Astra.Server. Если значение сигнала отличается от последнего значения в ядре, происходит перезапись значений сигнала.



Инициализирующие значения сигналов, которые обслуживаются модулем SNMP Manager, во время старта Astra.Server равны EMPTY.

История работы модуля сохраняется в журнале работы модуля. Ведение журнала работы модуля настраивается в общих параметрах модуля.

## Безопасность

В SNMPv3 поддерживается модель безопасности USM, которая основана на аутентификации имени пользователя. Модель безопасности USM (User-Based Security Model) использует концепцию авторизованного сервера (Authoritative Engine). Во время любой передачи сообщения одна или две сущности (агента), передатчик или приемник, рассматриваются в качестве авторизованного SNMP-сервера. Таким образом, получатель и отправитель сообщений являются авторизованными.

Для каждого сетевого устройства пароль преобразуется в некоторый уникальный ключ. Это обеспечивает дополнительную безопасность. Для шифрования пароля используется алгоритм MD5 или алгоритм SHA. Для шифрования ключа используется алгоритм шифрования AES или DES.

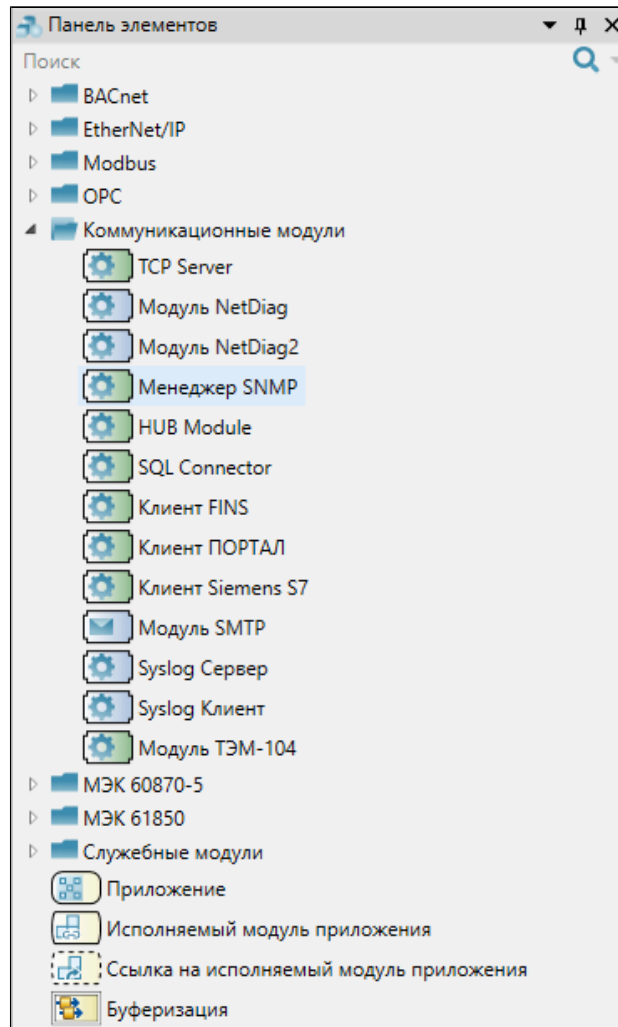
В SNMPv3 предусмотрено три уровня безопасности:

- NoAuthNoPriv – пароли передаются в открытом виде, конфиденциальность данных отсутствует;
- AuthNoPriv – аутентификация без конфиденциальности;
- AuthPriv – аутентификация и шифрование, максимальный уровень защищенности.



## 1.1.2.5.4.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



После добавления модуля настройте его свойства.

Свойства	
SNMP Manager Менеджер SNMP	
^ Параметры модуля	
Настройки по умолчанию	
Активность	Да
Отображаемое имя	
^ Параметры журналирования	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
^ Общие	
Имя	SNMP Manager

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Настройки по умолчанию	Выбор файла с готовой конфигурацией
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен;</li> <li>&gt; Нет – модуль остановлен.</li> </ul>
Отображаемое имя	Отображаемое имя модуля

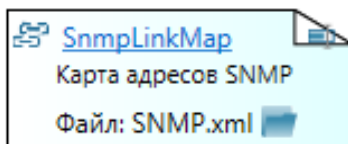
## Параметры журналирования

Параметры журналирования настраиваются одинаково для всех модулей.

## 1.1.2.5.4.2. Карта адресов

Для настройки сигналов модуля Менеджер SNMP используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов SNMP в исполняемое приложение;



2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	OID элемента	Индекс элемента массива	Композитный тип объекта	Способ получения данных
bool	bool	непосредственно				
uint1	uint1	непосредственно				

## Параметры карты адресов

Поле	Значение
Привязка	Тип привязки параметра: <ul style="list-style-type: none"><li>➤ непосредственно;</li><li>➤ только чтение;</li><li>➤ не привязан.</li></ul>
OID элемента	<ul style="list-style-type: none"><li>➤ OID параметра, значение которого записывается в сигнал. OID параметров опрашиваемого устройства содержится в документации на устройство. Получение значения OID конкретного параметра возможно с помощью специальных утилит (MIB Browsers);</li><li>➤ значение Any - указывается при необходимости получать только значения типа OctetString.</li></ul>
Индекс элемента массива	Индекс элемента массива

Композитный тип объекта	<p>Содержание параметра, значение которого записывается в сигнал. Параметр используется для корректного отображения значения сигнала при получении: IP-адреса устройства; MAC адреса устройства. Принимает значения:</p> <ul style="list-style-type: none"> <li>➤ IP-адрес;</li> <li>➤ Физический адрес.</li> <li>➤ Дата и время.</li> </ul>
Способ получения данных	<p>Способ получения данных от агента. Параметр принимает значения:</p> <ul style="list-style-type: none"> <li>➤ Только через опрос – запись значения в сигнал производится только через опрос;</li> <li>➤ Только через уведомления– запись значения в сигнал производится только через trap уведомления;</li> <li>➤ Через опрос и уведомления– запись значения в сигнал производится через опрос и trap уведомления (значение по умолчанию).</li> </ul>

## 1.1.2.5.4.3. Преобразование типов данных

Преобразование стандартных типов данных протокола SNMP в типы данных сигналов Astra.Server производится согласно таблице ниже.

Тип данных SNMP	Описание	Тип данных в Astra.Server
TimeTicks (0x43)	32-битное беззнаковое целое	uint4
Counter (0x41)	32-битное беззнаковое целое	uint4
Counter64	64-битное беззнаковое целое	uint8
Counter32	32-битное беззнаковое целое	uint4
Gauge (0x42)	32-битное беззнаковое целое	uint4
OctetString	массив байт (ANSI строка)	string
Integer	32-битное целое со знаком	int4
IpAddress	строка символов	string

Чтобы корректно преобразовать данные IP-адрес (тип IpAddress) и физический адрес (тип OctetString), используйте дополнительный параметр ComposedType в строке адреса сигнала. Физический адрес преобразуется в число 16-ричной системы счисления.

### Качество сигналов

Качества сигналов, которые выставлены коммуникационным модулем SNMP Manager, приведены в таблице ниже.

Числовое значение качества	Идентификатор качества	Расшифровка
0	BAD	Несоответствие типов SNMP сигнала и сигнала Astra.Server
4	CONFIG_ERROR	Сигнал неправильно сконфигурирован
8	NOT_CONNECTED	Сигнал принят на обслуживание, но устройство еще не инициализировано (отсутствует связь с агентом)
28	OUT_OF_SERVICE	Модуль не запущен
64	UNCERTAIN	Связь установлена, но значения еще не пришли
192	GOOD	Значение сигнала достоверно

# Диагностика работы модуля

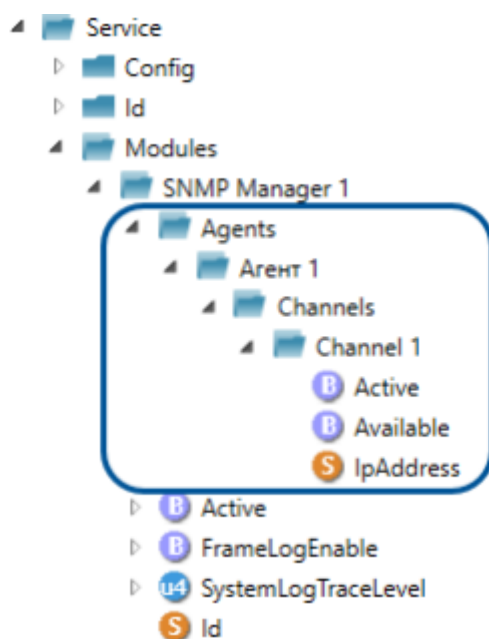
## Служебные сигналы

Модуль SNMP Manager динамически создаёт служебные сигналы контроля состояния каналов связи, а также стандартные для модулей Astra.Server служебные сигналы контроля и управления основными параметрами модуля.

Полный тег сигналов контроля состояния каналов связи имеет вид:




Service.Modules.<Имя модуля>.Agents.<Имя агента>.Channels.Channel N.<Имя сигнала>



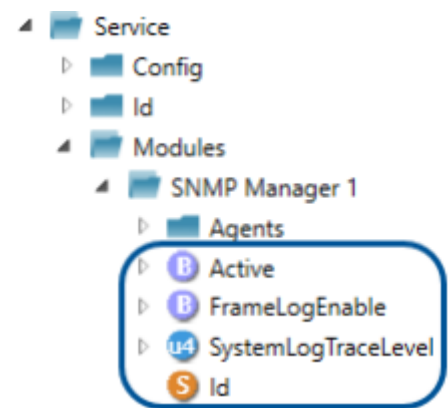
Сигнал	Тип	Описание сигнала
Active	Bool	Признак активности канала: > «True» – по каналу ведётся обмен данными с агентом; > «False» – обмен данными по каналу не ведётся. Если для обмена данными с агентом используется несколько каналов связи, то активным одновременно может быть только один из каналов
Available	Bool	Признак доступности канала:

		<ul style="list-style-type: none"> <li>› True – агент отвечает на тестовые запросы проверки наличия связи</li> <li>› False – агент не отвечает на тестовые запросы проверки наличия связи</li> </ul>
IpAddress	String	IP адрес канала

Полный тег стандартных служебных сигналов имеет вид:



Service.Modules.<Имя модуля>.<Имя сигнала>



Сигнал	Тип	Описание сигнала
Active	Bool	Активность модуля: <ul style="list-style-type: none"> <li>› True – запущен;</li> <li>› False – остановлен.</li> </ul> Соответствует значению параметра Активность. Управляется служебным сигналом Active.Set
FrameLogEnable	Bool	Ведение журнала работы: <ul style="list-style-type: none"> <li>› True – ведётся;</li> <li>› False – не ведётся.</li> </ul> Соответствует значению параметра Вести журнал работы модуля. Управляется служебным сигналом FrameLogEnable.Set
SystemLogTraceLevel	Uint4	Уровень детализации журнала работы: <ul style="list-style-type: none"> <li>› 1 – Предупреждения и аварийные сообщения;</li> <li>› 2 – Информационные сообщения;</li> </ul>

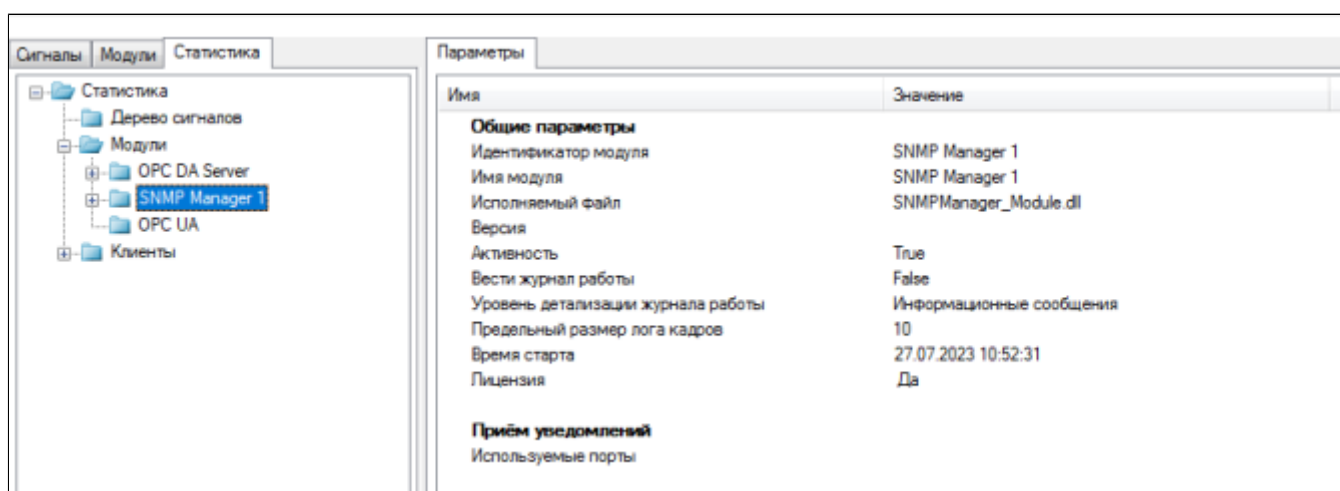


		<p>» 3 – Отладочные сообщения.</p> <p>Соответствует значению параметра Уровень трассировки в журнал приложений. Управляется служебным сигналом SystemLogTraceLevel.Set</p>
Id	String	Идентификатор модуля в конфигурации Astra.Server

## Статистические данные модуля

Подробную информацию о работе модуля SNMP Manager можно просмотреть на закладке Статистика сервисного приложения Конфигуратор, либо с помощью сервисного приложения Статистика.

Чтобы просмотреть параметры статистики модуля, подключитесь к Astra.Server и выберите в дереве объектов модуль SNMP Manager N.



Общая статистическая информация модуля представлена в группе Общие параметры.

Статистическая информация модуля о количестве прослушиваемых портов представлена в группе Приём уведомлений.

## Прием уведомлений

Параметр	Описание
----------	----------

Используемые порты	Количество портов, которые прослушивает модуль, для получения trap уведомлений
--------------------	--

## Статистические данные агентов

Подробную информацию о работе агента модуля SNMP Manager можно просмотреть на закладке Статистика сервисного приложения Конфигуратор, либо с помощью программы Статистика.

Чтобы просмотреть параметры статистики агента, подключитесь к Astra.Server и выберите в дереве объектов агента модуля SNMP Manager N.

Имя	Значение
<b>Основные параметры</b>	
Адрес	127.0.0.1
Порт для опроса	161
Порт для уведомлений	162
Агент включен	Да
Интервал опроса	1
Состояние опроса	Соединение установлено
<b>Сигналы на обслуживании</b>	
Всего	0
На опрос	0
На получение уведомлений	0
<b>Изменения значений сигналов</b>	
Получено опросом	0
Получено уведомлениями	0

Каждый агент, с которым поддерживается соединение, предоставляет следующие статистические данные:

## Основные параметры

Параметр	Описание
Адрес	Текущий IP адрес сетевого устройства, с которым поддерживается соединение
Порт для опроса	Порт, открываемый агентом для соединения с модулем
Порт для уведомлений	Порт для отправки trap уведомлений модулю

Состояние агента	Состояние связи с агентом
Интервал опроса	Максимальное время ожидания ответа агента на отправленный запрос
Состояние опроса	Состояние, показывающее проводится опрос агента или нет

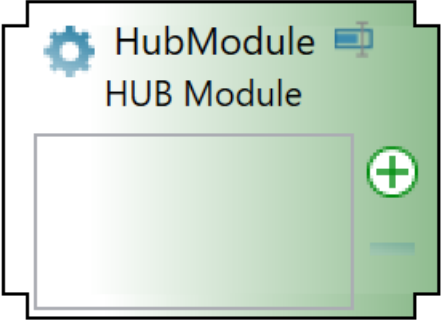
## Сигналы на обслуживании

Параметр	Описание
Всего	Суммарное количество обслуживаемых сигналов с разными типами данных
На опрос	Суммарное количество сигналов, настроенных на получение данных через опрос
На получение уведомлений	Суммарное количество сигналов, настроенных на получение данных через trap уведомления

## Изменения значений сигналов

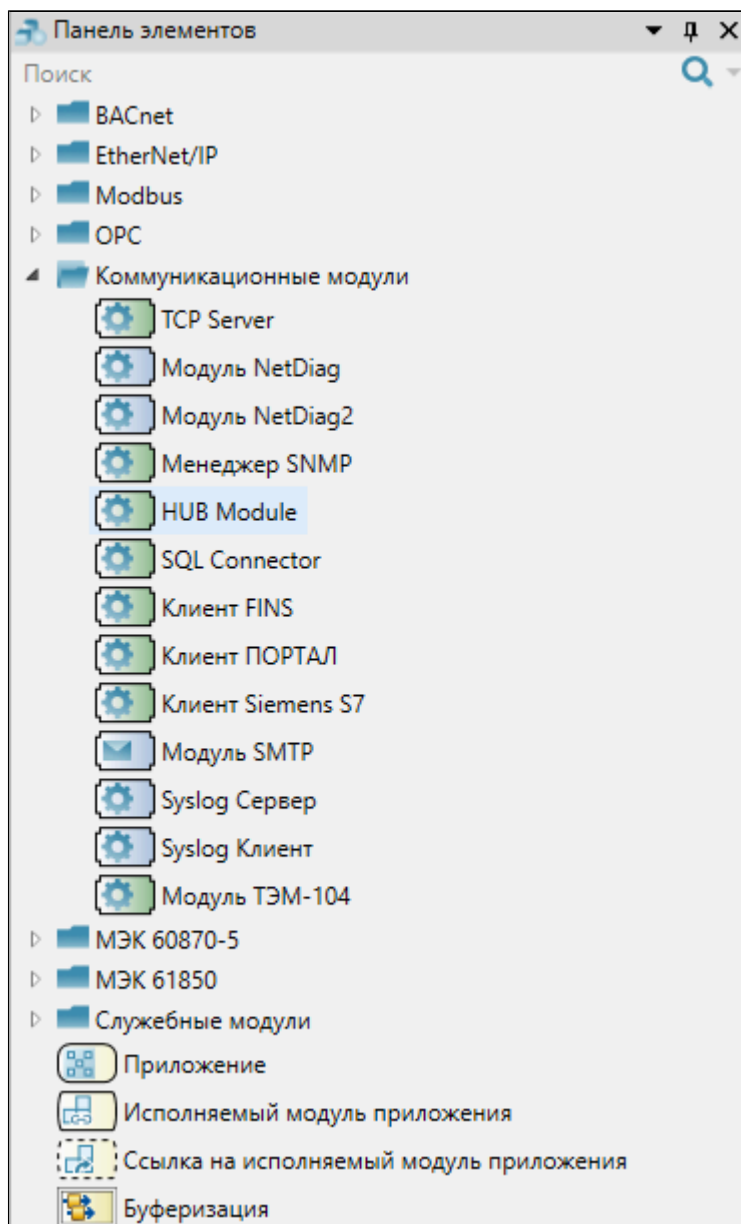
Параметр	Описание
Получено опросом	Количество перезаписей значений сигнала, полученных путем опроса
Получено уведомлениями	Количество перезаписей значений сигнала, полученных путем trap уведомлений

# 1.1.2.5.5. HUB Модуль



# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

Свойства	
HubModule HUB Module	
⌵ Параметры модуля	
Разрешено изменять значения сигналов	Да
Настройки по умолчанию	
Активность	Да
Отображаемое имя	
⌵ Параметры журналирования	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
⌵ Общие	
Имя	HubModule

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

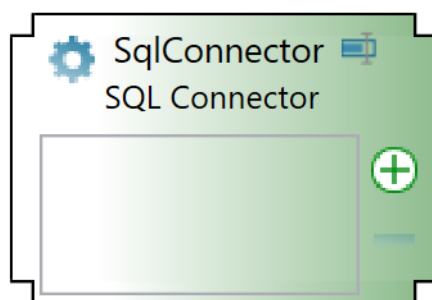
Параметр	Описание
Разрешено изменять значения сигналов	Значения: <ul style="list-style-type: none"> <li>&gt; Да</li> <li>&gt; Нет</li> </ul>
Настройки по умолчанию	Путь к готовой конфигурации модуля
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен</li> <li>&gt; Нет – модуль остановлен</li> </ul>

Отображаемое имя	Отображаемое имя модуля
------------------	-------------------------

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.5.6. SQL Connector



Модуль SQL Connector работает в составе Astra.Server. Модуль предназначен для взаимодействия с реляционными БД посредством запросов языка SQL. Данные, полученные из БД, записываются в сигналы дерева Astra.Server в соответствии с логикой разбора результатов запроса.

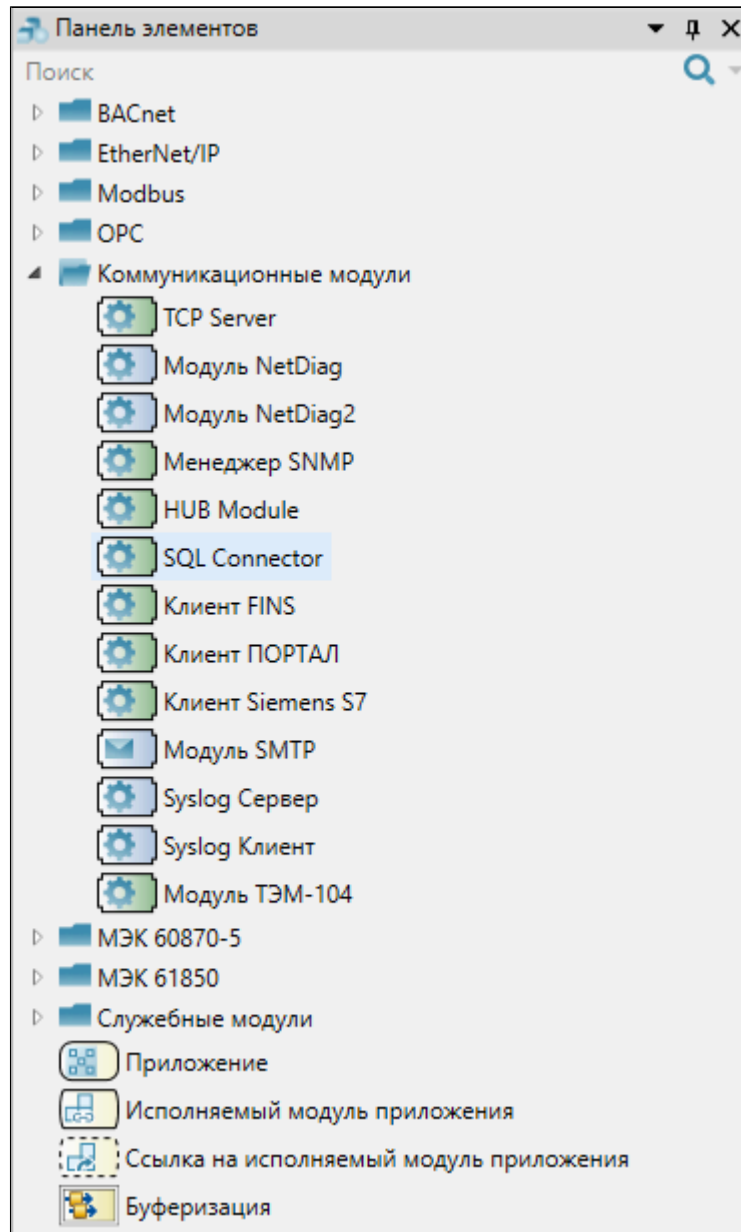


Поддерживается работа с СУБД PostgreSQL версий 9.5, 9.6 и 11.

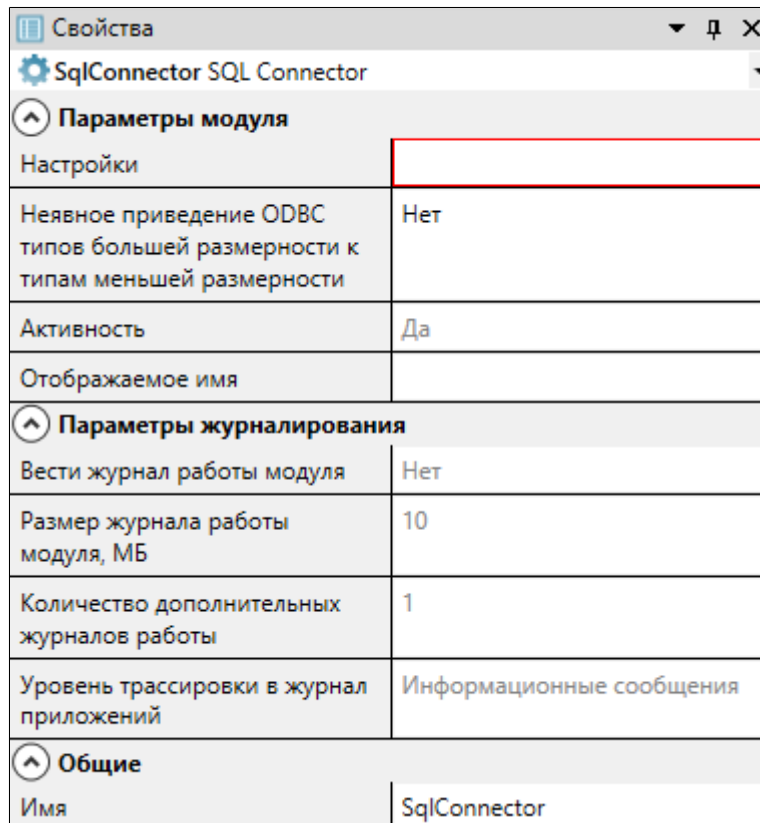


## 1.1.2.5.6.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.



## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Настройки по умолчанию	Выбор файла с готовой конфигурацией
Неявное приведение ODBC типов большей размерности к типам меньшей размерности	Режим, при котором значения большей размерности записываются в сигналы сервера меньшей размерности. Активируется только если параметру Тип источника установлено значение «ODBC»

Активность	Активность модуля при запуске/перезапуске Astra.Server: > Да – модуль запущен > Нет – модуль остановлен
Отображаемое имя	Отображаемое имя модуля

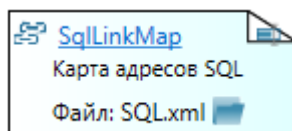
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.5.6.2. Карта адресов

Для настройки сигналов модуля SQL Conector используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов SQL в исполняемое приложение;



2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	ID Запроса	Тип	Value	Quality	Timestamp	KeyFieldValue	ParamID	ParamValuePropID
bool	bool	непосредственн	1	QRead	1	1	1	1	1	1
uint1	uint1	непосредственн	1	QRead	1	1	1	1	1	1

## Параметры карты адресов

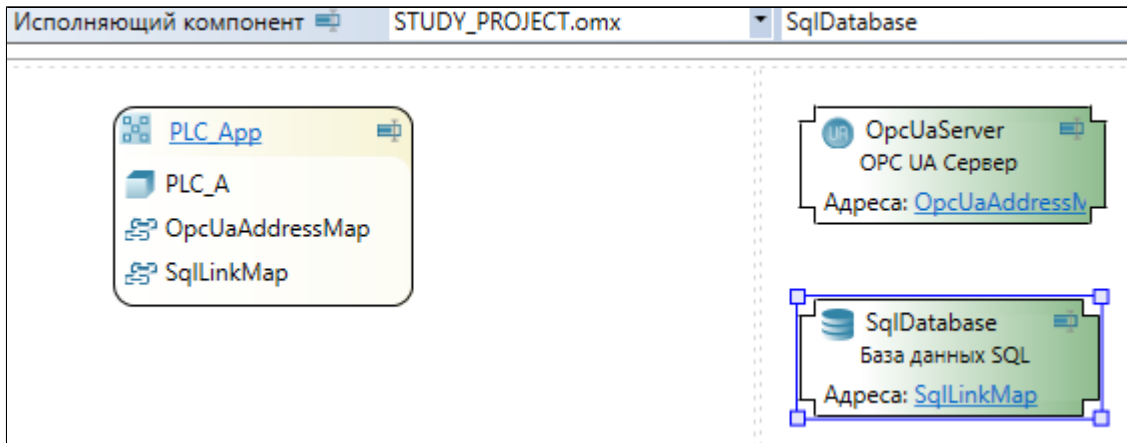
Поле	Значение
Привязка	Тип привязки параметра: <ul style="list-style-type: none"><li>› непосредственно;</li><li>› только чтение;</li><li>› не привязан.</li></ul>
ID запросов	Идентификатор запроса, который будет отправлен в источник или в текст которого будет вставлено значение параметра
Тип	Тип сигнала: <ul style="list-style-type: none"><li>› QRead - для непосредственной перекладки данных из результирующей выборки</li><li>› QParam - для динамического формирования SQL запроса</li></ul>
Value	Имя столбца результирующей выборки, из которого будет браться значение для сигнала

Quality	Имя столбца результирующей выборки, из которого будет выставляться качество сигнала. Если не указано, то выставляется качество GOOD
Timestamp	Имя столбца результирующей выборки, из которого будет выставляться метка времени для сигнала. Если не указано, то выставляется метка времени NOW
KeyFieldValue	Значение ключевого поля. Если указано, то к результирующей выборке применяется дополнительный фильтр, оставляющий только те строки, где ключевое поле принимает указанное значение
ParamID	Порядковый номер неизвестного параметра в тексте запроса
ParamValueopID	Номер свойства сигнала, значение которого будет перекладываться в текст запроса. Если не указано, то используется свойство 2 (Value)

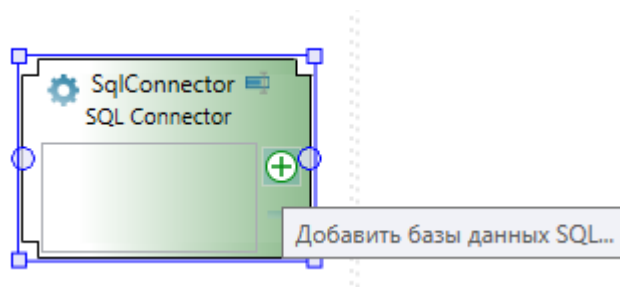
## 1.1.2.5.6.3. Добавление базы данных

Для того, чтобы добавить базу данных для обмена данными, необходимо:

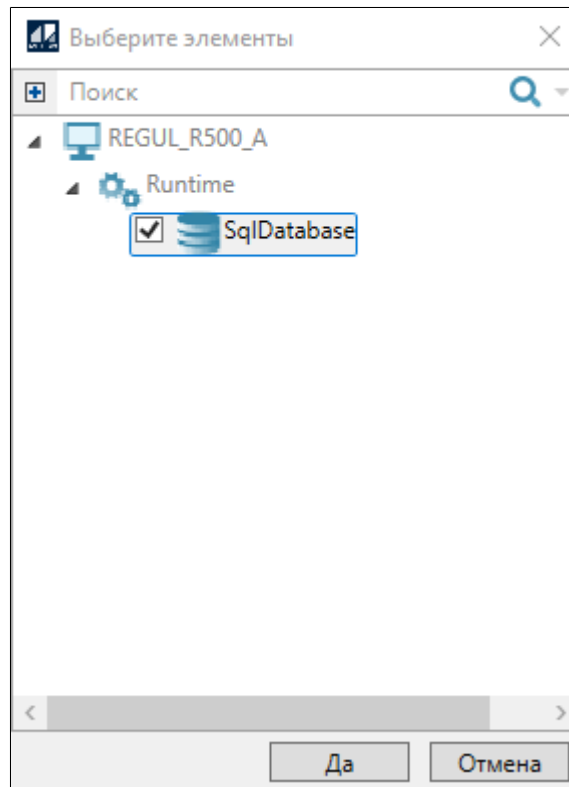
1. В исполняемом приложении добавьте модуль "База данных SQL"



2. В модуле SQL Connector нажмите на кнопку "Добавить базы данных SQL..."



3. В открывшемся окне выберите необходимые базы данных для подключения



4. Заполните параметры взаимодействия с базой данных, которые описаны в таблице ниже:

Параметр	Описание
База данных SQL	Идентификатор источника
Поставщик данных	Интерфейс доступа к данным: <ul style="list-style-type: none"> <li>&gt;OLE DB</li> <li>&gt;ODBC</li> </ul>
Строка подключения	Параметры подключения к источнику. Значение по умолчанию: Provider=SQLOLEDB; DataSource=ServerName; Initial Catalog=DatabaseName; UserID=UserID;
Пароль	Пароль пользователя из строки подключения

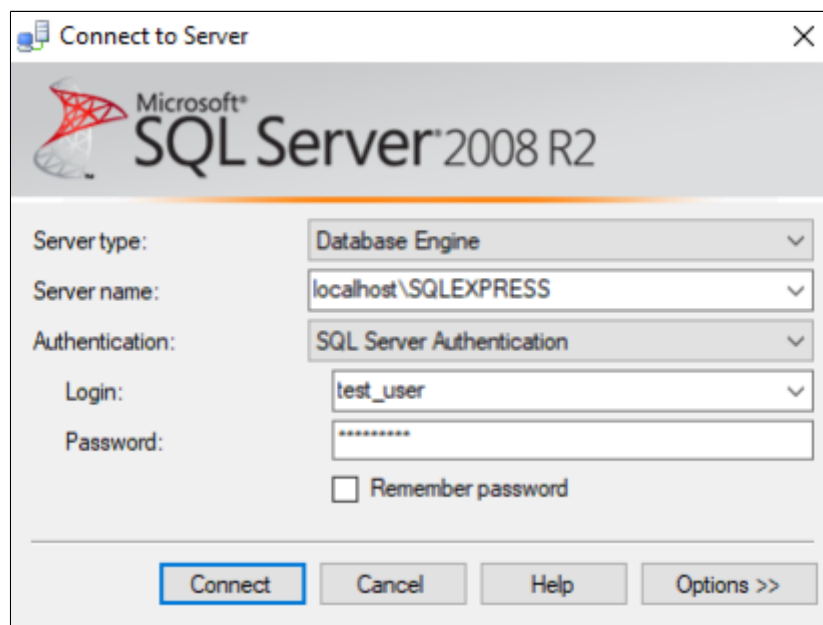
### Параметры строки подключения

Параметр	Описание
----------	----------


Provider	Имя провайдера для работы с источником. Определение значения параметра описано ниже
Data Source	Имя источника
Initial Catalog	Имя базы данных в составе источника
User ID	Логин пользователя

## Соединение с SQL Server 2008 R2

Имеется локальный экземпляр сервера SQLEXPRESS, в его составе есть база mydb. Подключение проходит по логину test\_user и паролю qwerty123.



Строка подключения к подобному источнику с помощью модуля SQL Connector показана ниже:

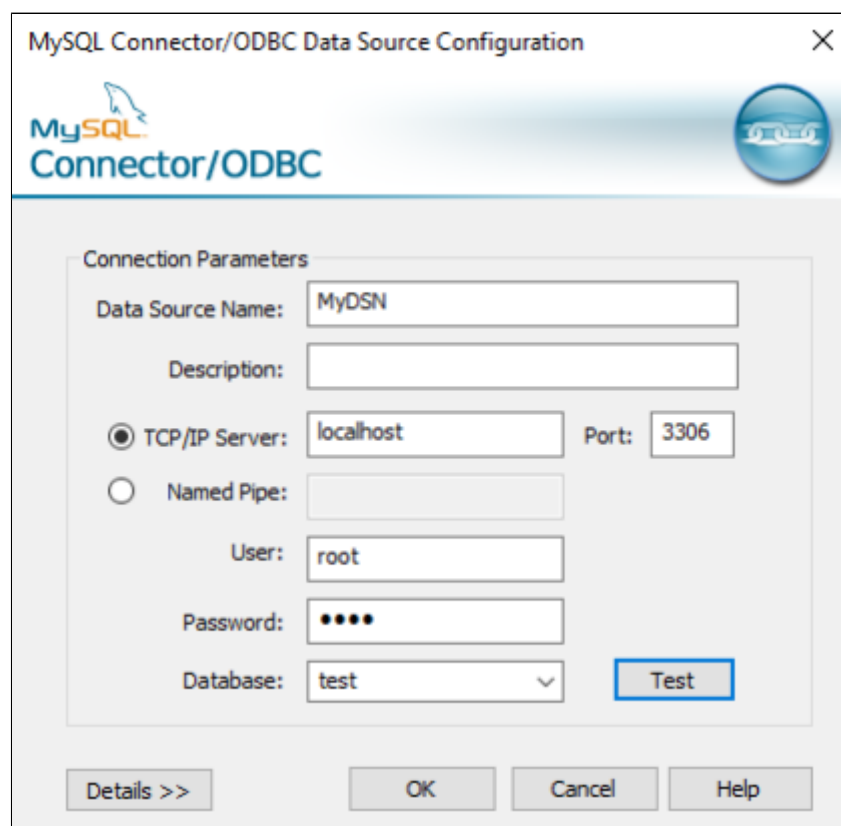
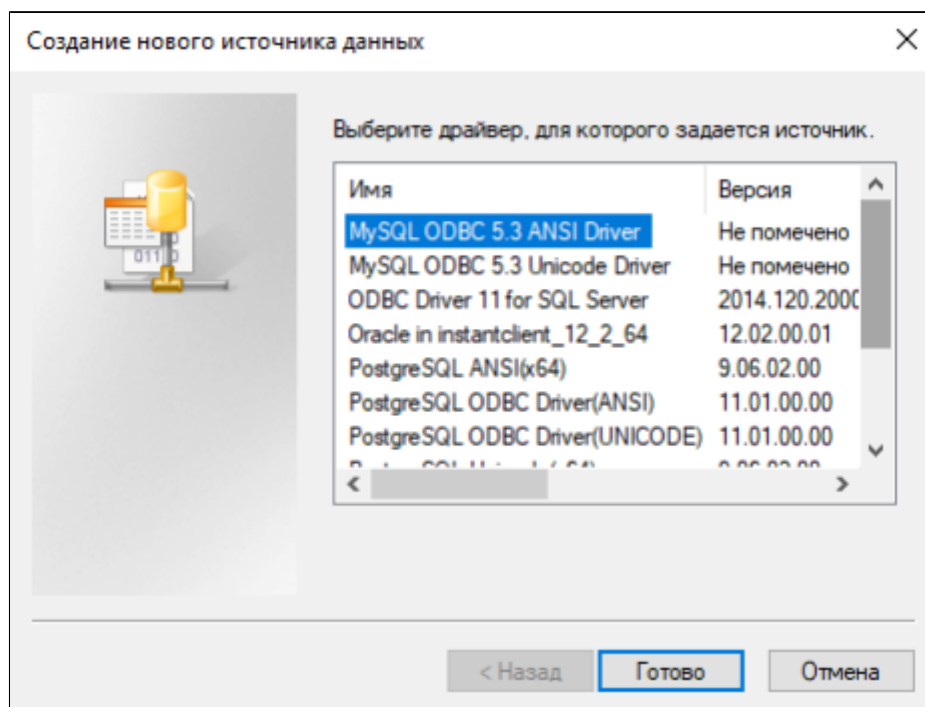
 Provider=SQLOLEDB; Data Source=localhost\SQLEXPRESS; Initial Catalog=mydb; User ID=test\_user; Password=qwerty123;

## Соединение с MySQL

Есть локальная база MySQL, подключение к которой сконфигурировано через инструмент Администратор источников данных ODBC (Панель управления →



Администрирование → Источники данных ODBC), как показано на рисунках ниже.



Строка подключения к подобному источнику с помощью модуля SQL Connector показана ниже.



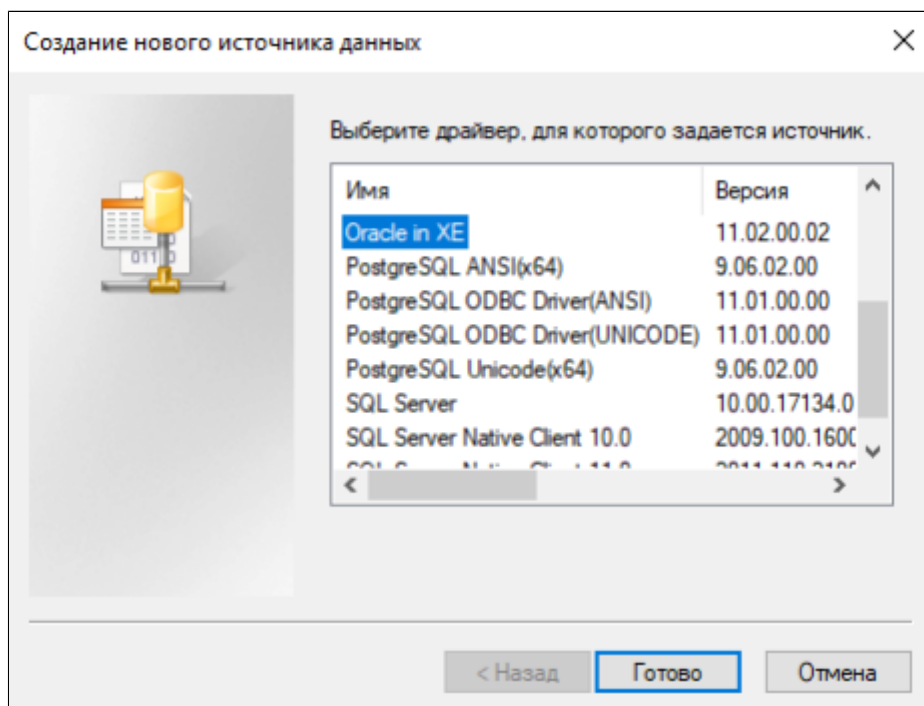
Provider=MSDASQL.1; Data Source=MyDSN;



Установка и настройка драйвера ODBC для MySQL в ОС Linux приведена в приложении.

## Соединение с Oracle XE

Есть локальная база Oracle XE со стандартным пользователем system (пароль qwerty123). Чтобы протестировать подключение к базе сконфигурируйте подключение через инструмент Администратор источников данных ODBC (Панель управления → Администрирование → Источники данных ODBC), как показано на рисунках ниже.



Строка подключения к подобному источнику с помощью модуля SQL Connector показана ниже.



Provider=OraOLEDB.Oracle.1; Data Source=XE; User ID=system;  
Password=qwerty123;

## Определение значения параметра Provider для строки подключения

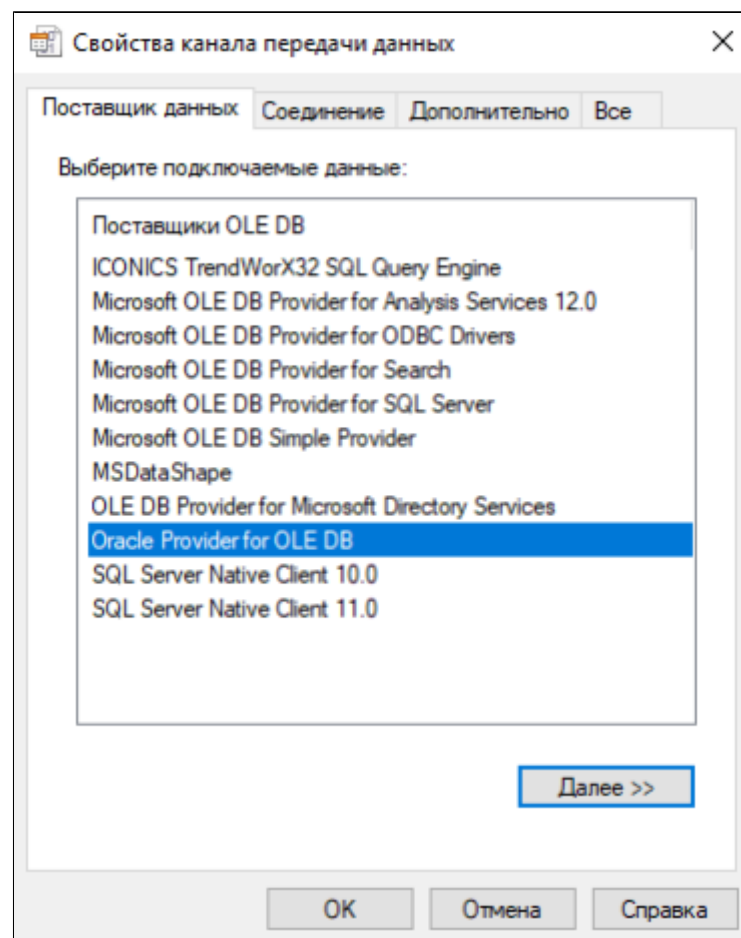
Чтобы определить значение параметра Provider для строки подключения SQL Connector выполните следующие действия:

1. Создайте новый текстовый файл и переименуйте его в MyUdl.udl.

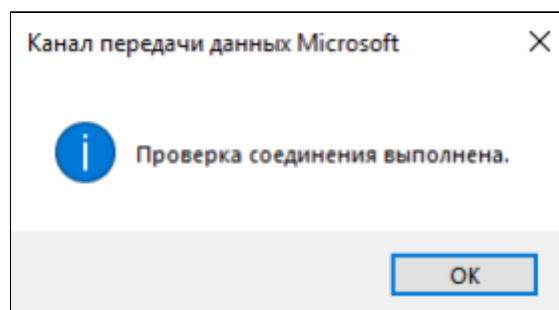
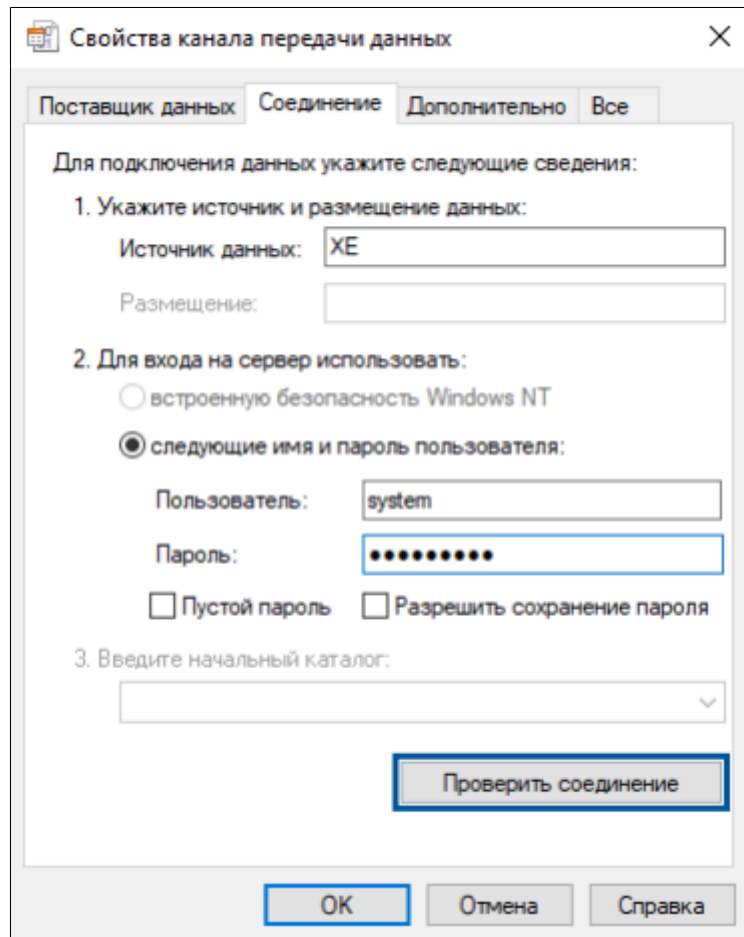


2. Запустите файл MyUdl.udl.

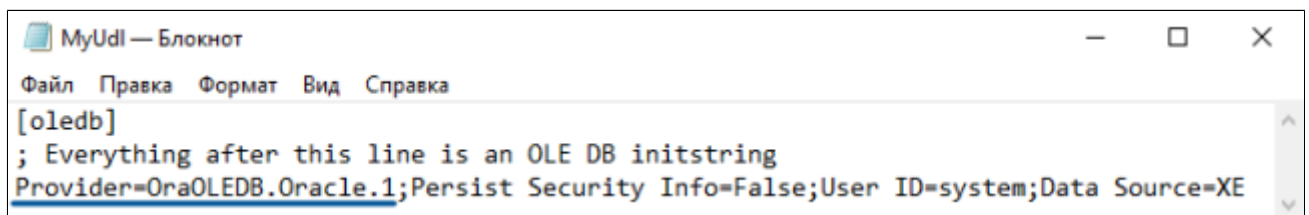
3. На вкладке Поставщик данных выберите нужного провайдера.



4. На вкладке Соединение укажите Источник данных, Пользователя и Пароль. Затем протестируйте соединение.



5. Если соединение прошло успешно, то закройте файл MyUdl.udl, переименуйте его в файл MyUdl.txt и откройте в текстовом редакторе.



В сформированной строке будет присутствовать значение параметра Provider, которое можно применять в строке подключения для модуля SQL Connector.

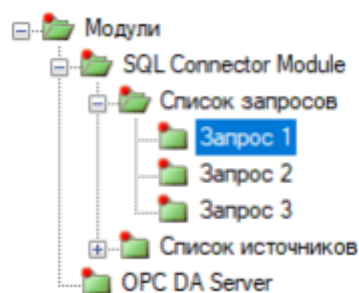


Provider=OraOLEDB.Oracle.1; User ID=system; Data Source=XE;

При возникновении трудностей с подключением к источникам данных обратитесь к журналу работы модуля, который можно просмотреть через сервисное приложение Просмотрщик лога кадров.

## Настройка запросов к источникам данных

Модуль SQL Connector посылает SQL-запросы к источнику данных. Запросы добавляются в ветку модуля Список запросов.



Каждый запрос имеет уникальный Идентификатор, основные параметры и значения по умолчанию.

## Основные параметры

Параметр	Описание
Текст запроса	Инструкция типа SELECT для выборки данных из источника, инструкции типа UPDATE, INSERT для модификации данных в источнике или инструкции вызова хранимых процедур (EXECUTE или CALL)
Имя ключевого поля	Имя столбца таблицы, по которому будет организована дополнительная фильтрация выбранных данных

Период исполнения запроса	Интервал в миллисекундах, с которым будет выполняться SQL-запрос. Если указан «0», то запрос будет выполняться только по отдельной команде
Возможность результатов запроса быть пустым	Если установлено значение Да, то сигналы не будут менять качества даже при пустых результатах запроса
Часовой пояс	Параметр указывает, в каком часовом поясе UTC хранятся метки времени в базе данных. Все метки времени, отправляемые в базу данных (запросами типа UPDATE, INSERT), будут скорректированы к указанному часовому поясу. Все метки времени, полученные из базы данных (запросами типа SELECT), будут скорректированы к UTC

## Значения по умолчанию

Параметр	Описание
Качество при неполном значении	Значение качества, устанавливаемое в случае если результатом запроса является пустое значение (NULL). По умолчанию устанавливается значение качества 64 (Uncertain)
Значения по умолчанию	Значения столбцов, устанавливаемые в случае если результатом запроса является пустое значение (NULL)



Если параметр Значения по умолчанию не задан и при чтении значений столбца встречается пустое значение (NULL), последующие значения столбца считываться не будут. В этом случае запрос выдаёт ошибку, которая отображается в журнале приложений.



Текст запроса может строиться динамически. Для этого он должен иметь неизвестные параметры, отмеченные знаками "?".

# Диагностика работы модуля

## Журнал работы модуля

Для анализа ошибок, возникающих в процессе работы модуля, воспользуйтесь сервисным приложением Просмотрщик лога кадров.

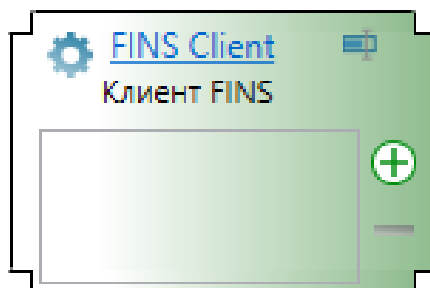
№	Дата	Время	Описание
1	26.06.2023	00:15:37:857	Журнал открыт для записи
2	26.06.2023	00:15:37:868	Некорректная строка адреса для сигнала "Новый сигнал 1". В к...
3	26.06.2023	00:15:37:868	Нет сигналов для обслуживания модулем
4	26.06.2023	00:15:37:868	Журнал закрыт
5	26.06.2023	00:16:00:266	Журнал открыт для записи
6	26.06.2023	00:16:00:266	Некорректная строка адреса для сигнала "Новый сигнал 1". В к...
7	26.06.2023	00:16:00:266	Нет сигналов для обслуживания модулем
8	26.06.2023	00:16:00:266	Журнал закрыт
9	26.06.2023	00:17:43:790	Журнал открыт для записи
10	26.06.2023	00:17:43:790	Некорректная строка адреса для сигнала "Новый сигнал 1". Сиг...

Некорректная строка адреса для сигнала "Новый сигнал 1". Сигнал не принимается на обслуживание, запрос сконфигурирован без параметров.

0 1 2 3 4 5 6 7 8 9

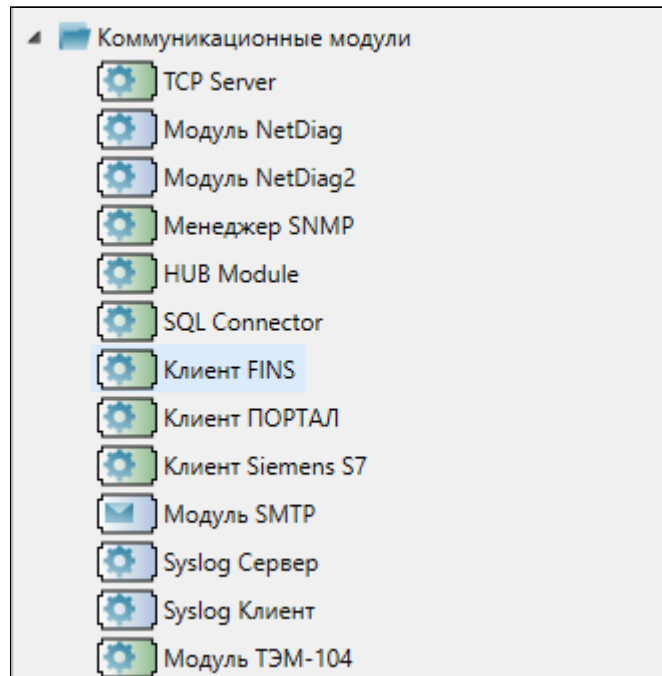


## 1.1.2.5.7. Клиент FINS



# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора [Astra.AStudio](#).



После добавления модуля настройте его свойства.

В свойствах модуля имеются параметры модуля, параметры журналирования и общие параметры.

⚙️ FINS Client Клиент FINS	
⬆️ <b>Общие</b>	
Адрес сети	0
Адрес узла	1
Имя	FINS Client
⬆️ <b>Параметры модуля</b>	
Активность	Да
Отображаемое имя	
⬆️ <b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения

## Общие

Параметр	Описание
Адрес сети	Адрес сети
Адрес узла	Адрес узла
Имя	Имя модуля

## Параметры модуля

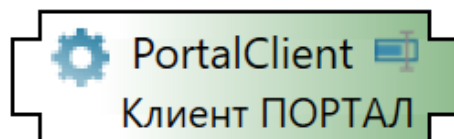
Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен</li> <li>&gt; Нет – модуль остановлен</li> </ul>
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

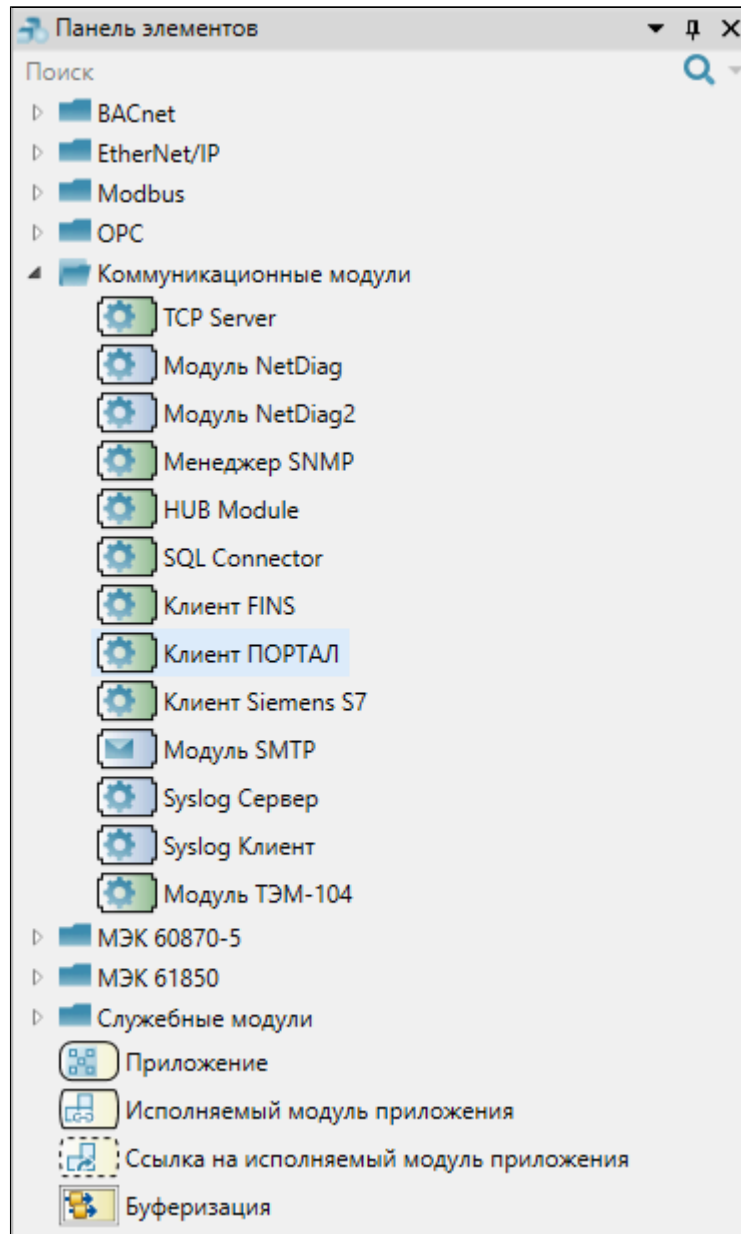
## 1.1.2.5.8. Клиент ПОРТАЛ

Модуль, импортирующий значения из системы ПОРТАЛ.



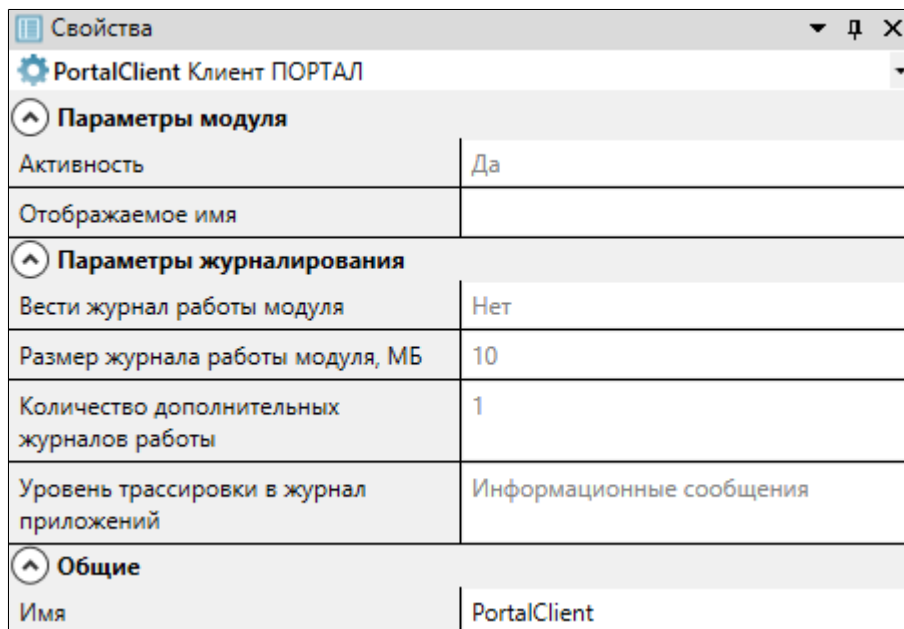
# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора [Astra.AStudio](#).



После добавления модуля настройте его свойства.

В свойствах модуля имеются параметры модуля, параметры журналирования и общие параметры.



## Общие

Параметр	Описание
Имя	Имя модуля

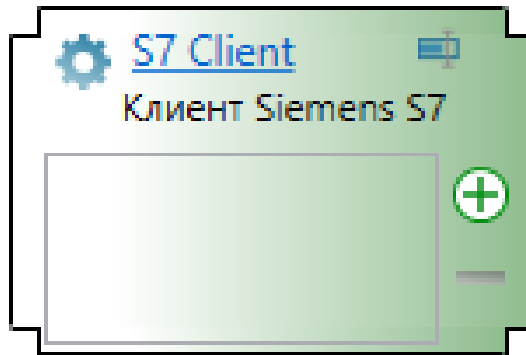
## Параметры модуля

Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>› Да – модуль запущен</li> <li>› Нет – модуль остановлен</li> </ul>
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.5.9. Siemens S7 Client



Модуль Siemens S7 Client – коммуникационный модуль, предназначенный для обмена данными между Astra.Server и программируемым контроллером Siemens S7 PLC (ПЛК) по протоколу S7.

Функции модуля Siemens S7 Client:

- › сбор данных - получение значений переменных с ПЛК и сохранение полученных значений в сигналы Astra.Server;
- › подача команд управления - отправка значений сигналов Astra.Server в ПЛК.

### Информация о ПЛК

ПЛК имеет один или два канала связи (основной и резервный), по которым устанавливается подключение. Для подключения к ПЛК используются параметры:

- › IP-адрес;
- › номер стойки (Rack number);
- › номер слота (Slot number).

Переменные в ПЛК располагаются в областях данных (Areas):

- › Область DB (Data Blocks). Областей DB в ПЛК множество. Каждая из областей имеет свой номер - адрес области данных (AreaAddress).
- › Область MB (Merkers);
- › Область EB (Process Inputs);
- › Область AB (Process Outputs);

- › Область ТМ (Timers);
- › Область СТ (Counters).

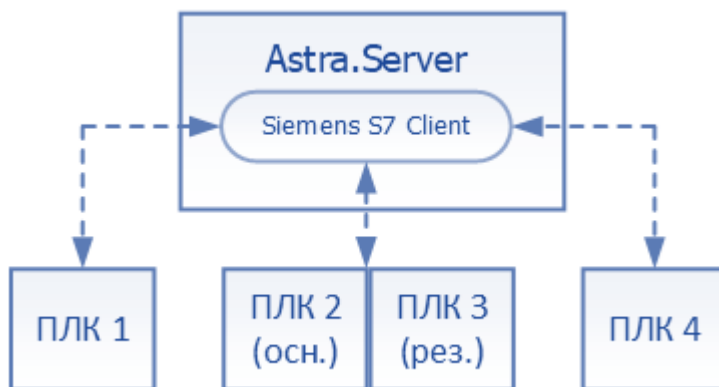
Каждая область данных содержит множество участков памяти размером 1 байт. Каждый участок памяти имеет свой адрес - смещение в области данных ПЛК (ValueAddress).

Для чтения или записи значения переменной ПЛК необходима информация о переменной:

- › область данных, в которой располагается переменная;
- › адрес области данных (для переменных Области DB);
- › смещение относительно начала области данных;
- › тип переменной.

## Обмен данными с ПЛК

Обмен данными между модулем Siemens S7 Client и ПЛК основан на архитектуре Клиент - Сервер. Модуль Siemens S7 Client является клиентом, а ПЛК - сервером. Модуль Siemens S7 Client может обмениваться данными одновременно с несколькими ПЛК, в том числе и резервируемыми ПЛК.



Обмен данными выполняется по сети Ethernet (протокол TCP) в режиме запрос-ответ. Инициатором запроса является модуль Siemens S7 Client. Самостоятельно ПЛК данные не передаёт.



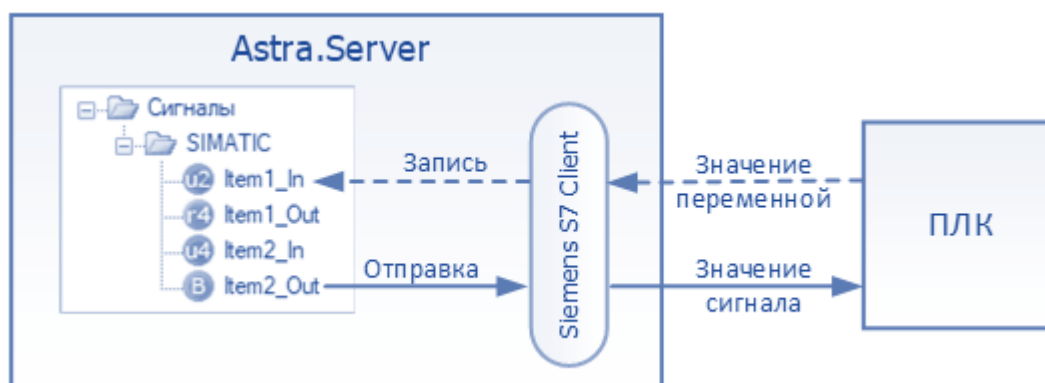
Запросы и ответы представляют собой кадры данных. Для модуля Siemens S7 Client запрос, отправляемый в ПЛК, является исходящим кадром, а ответ, полученный от ПЛК – входящим кадром.

Модуль Siemens S7 Client при обмене данными с ПЛК:

- получает значения переменных ПЛК и записывает полученные данные в сигналы Astra.Server;
- отправляет значения сигналов Astra.Server в ПЛК.

При этом сигналы Astra.Server должны иметь типы, соответствующие типам переменных ПЛК. Соответствие типов данных ПЛК и типов данных Astra.Server приведено в приложении.

Сигналы, в которые записываются полученные значения переменных ПЛК, называются входящими. Сигналы, значения которых отправляются в переменные ПЛК, называются исходящими.



## Получение данных от ПЛК

Модуль Siemens S7 Client получает значения переменных путём опроса ПЛК. Опрос может выполняться:

- периодически через заданный промежуток времени;
- по команде путем изменения значения служебного сигнала.

Способ выполнения опроса задаётся в настройках категории данных, к которой относится входящий сигнал.



Категория данных - это группа сигналов, значения которых обновляются с одинаковой частотой. Категории данных позволяют запрашивать значения сигналов с частотой их обновления периодически или только по запросу.

Порядок получения значений переменных ПЛК:

1. Модуль Siemens S7 Client формирует запрос на чтение, содержащий данные запрашиваемых переменных: область данных, адрес области данных (для Области DB), смещение в области данных ПЛК, количество байт в зависимости от типа соответствующего сигнала в Astra.Server.
2. Сформированный запрос на чтение отправляется в ПЛК в виде исходящего кадра.
3. Проверяется ответ, полученный от ПЛК: если при обмене данными произошла ошибка, то в журнал приложений и журнал работы модуля запишутся сообщения с расшифровкой текста ошибки. Если ошибок нет, то ответ обрабатывается далее.
4. Из полученного входящего кадра извлекается значение переменной.
5. В сигнал Astra.Server записывается полученное значение переменной, а значение качества и метки времени модуль Siemens S7 Client устанавливает самостоятельно.

## Отправка данных в ПЛК

Порядок отправки значений сигналов в ПЛК:

1. При изменении значения исходящего сигнала Astra.Server, модуль Siemens S7 Client формирует запрос на запись значения в ПЛК, содержащий данные записываемой переменной - область данных, адрес области данных (для Области DB), смещение в области данных ПЛК, а также значение сигнала, преобразованное в массив байт. Качество и метка времени не передаются.

2. Сформированный запрос на запись отправляется в ПЛК в виде исходящего кадра.
3. Проверяется ответ, полученный от ПЛК: если при обмене данными произошла ошибка, то в журнал приложений и журнал работы модуля запишутся сообщения с расшифровкой текста ошибки. Если ошибок нет - реакции модуля не последует.

## Качество и метка времени

ПЛК предоставляет и принимает только значение переменной. Качество и метка времени при обмене данными не передаются.

Модуль Siemens S7 Client самостоятельно устанавливает метку времени и качество полученным от ПЛК значениям. В метку времени устанавливается время приёма данных, а качество - GOOD (192).

Во время отсутствия связи с источниками данных входящим сигналам устанавливается качество QUALITY\_COMM\_FAILURE (24).

## Выбор ПЛК при обмене данными с резервируемыми ПЛК

При обмене данными с резервируемыми ПЛК модуль Siemens S7 Client выбирает ПЛК для установки соединения в следующем порядке:

1. По умолчанию для обмена данными используется основной ПЛК и его основной канал связи.
2. Если по основному каналу не удалось установить соединение, то происходит переключение на резервный канал (при его наличии) основного ПЛК.
3. Если в процессе обмена данными удалось восстановить соединение по основному каналу связи, то обмен данными будет продолжен по основному каналу.

4. Если ни по одному из каналов основного ПЛК не удалось установить соединение, то происходит переключение на резервный ПЛК. Выбор канала связи резервного ПЛК выполняется аналогично выбору канала связи основного ПЛК.

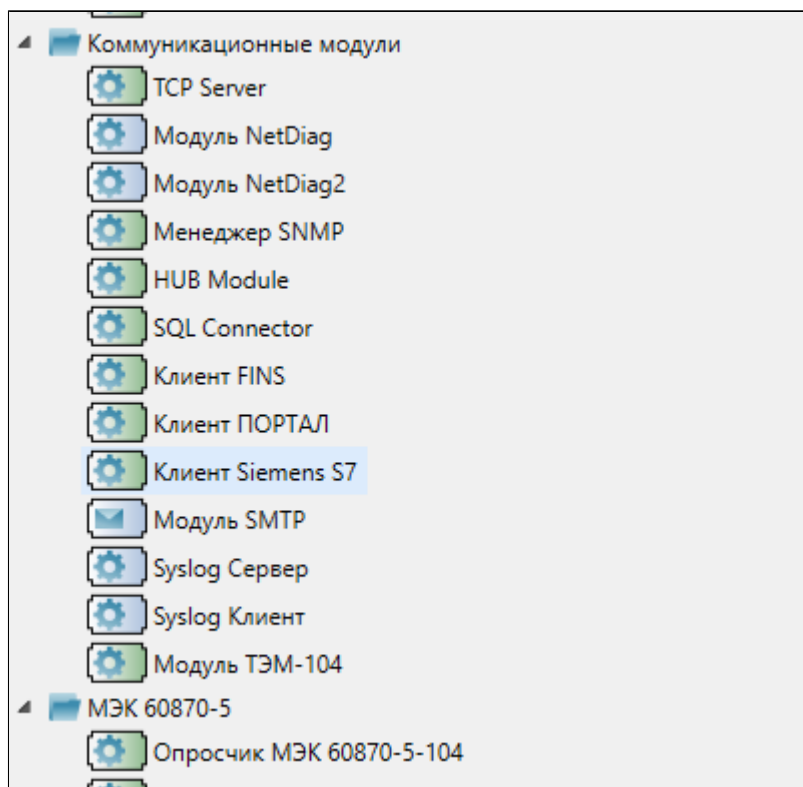
5. Если в процессе обмена данными удалось восстановить соединение с основным ПЛК по одному из его каналов связи, то обмен данными будет продолжен с основным ПЛК.

## **Работа модуля в резерве**

В режиме РЕЗЕРВ модуль Siemens S7 Client не отправляет данные в ПЛК, но может вести опрос источников, поддерживать соединение с ПЛК не опрашивая его, либо разрывать соединение с ПЛК. Работа модуля в РЕЗЕРВЕ настраивается в параметрах модуля.

## 1.1.2.5.9.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

S7 Client Клиент Siemens S7	
<b>Параметры модуля</b>	
Режим работы в резерве	Поддерживать соединение
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	S7 Client

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Режим работы в резерве	<p>Опрос источников при переходе модуля в РЕЗЕРВ:</p> <ul style="list-style-type: none"> <li>➤ Поддерживать соединения – модуль прекращает опрос ПЛК, при этом соединения с ПЛК не закрываются;</li> <li>➤ Вести опрос – модуль продолжает опрос ПЛК;</li> <li>➤ Закрывать соединения – модуль закрывает соединения с ПЛК.</li> </ul> <p>Значение по умолчанию Поддерживать соединения.</p>
Активность	<p>Активность модуля при запуске/перезапуске Astra.Server:</p> <ul style="list-style-type: none"> <li>➤ Да – модуль запущен</li> </ul>

	» Нет – модуль остановлен
Отображаемое имя	Отображаемое имя модуля

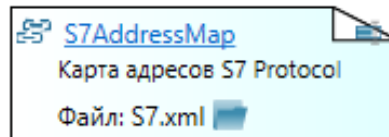
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.5.9.2. Карта адресов

Для настройки сигналов модуля Siemens S7 используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов S7 Protocol в исполняемое приложение;



2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	Область памяти	Номер DB	Смещение	Номер бита	Тип строки	Размер строки	Категория данных
bool	bool	непосредственн	DB (Data Blocks)		0				
uint1	uint1	непосредственн	DB (Data Blocks)		0				

## Параметры карты адресов

Поле	Значение
Привязка	Тип привязки параметра: <ul style="list-style-type: none"><li>› непосредственно;</li><li>› только чтение;</li><li>› не привязан.</li></ul>
Области памяти	Переменные в ПЛК располагаются в областях данных (Areas): <ul style="list-style-type: none"><li>› Область DB (Data Blocks).</li><li>› Область MB (Merkers);</li><li>› Область EB (Process Inputs);</li><li>› Область AB (Process Outputs);</li><li>› Область TM (Timers);</li><li>› Область CT (Counters).</li></ul>
Номер DB	Адрес области данных, в которой располагается переменная. Значение в диапазоне от 1 до 65535.



Смещение	Адрес переменной в области данных. Значение в диапазоне от 1 до 65535.
Номер бита	Номер бита в байте. Только для сигналов типа Bool. Значения в диапазоне от 0 до 7
Тип строки	Тип строки: <ul style="list-style-type: none"> <li>› Обычная строка</li> <li>› Широкая строка</li> </ul> Только для сигналов типа String
Размер строки	Размер строки. Значения в диапазоне: <ul style="list-style-type: none"> <li>› от 1 до 255 - для обычной строки;</li> <li>› от 1 до 122 - для широкой строки.</li> </ul>
Категория данных	Имя категории данных, к которой относится сигнал (только для входящего сигнала). Имя категории данных в карте адресов прописывается вручную.

# 1.1.2.5.9.3. Диагностика работы

## Служебные сигналы

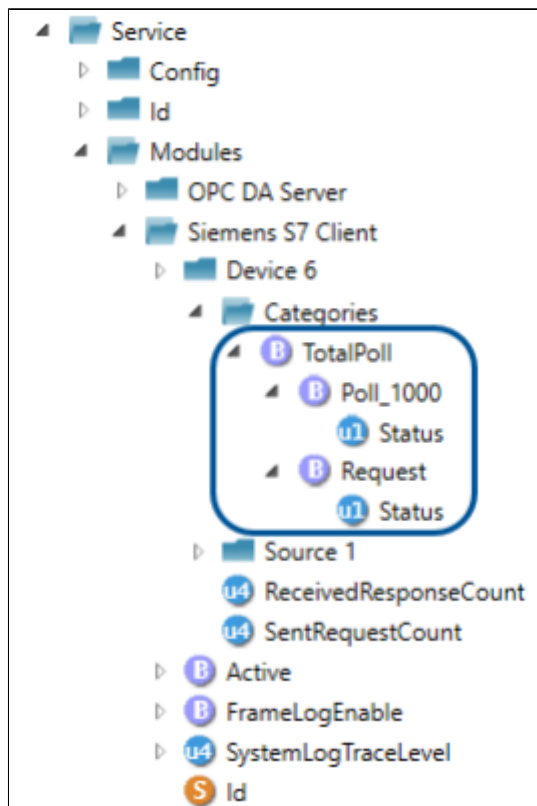
Модуль Siemens S7 Client динамически создаёт служебные сигналы:

- › опроса категорий данных по команде;
- › контроля состояния источника;
- › контроля операций;
- › контроля и управления основными параметрами модуля.

## Опрос категорий данных по команде

Полный тег служебных сигналов опроса категорий данных по команде:

```
Service.Modules.<Имя модуля>.Device <Идентификатор устройства>.Categories.<Имя сигнала>
```



Сигнал	Тип	Описание сигнала
TotalPoll	Bool	Команда опроса всех категорий данных:

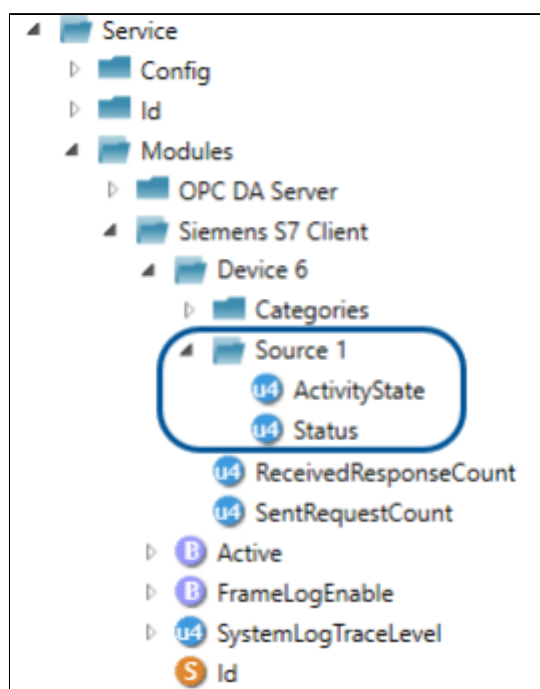
		<ul style="list-style-type: none"> <li>True - опросить все категории данных</li> </ul>
Имя категории	Bool	Команда опроса категории данных: <ul style="list-style-type: none"> <li>True - опросить категорию данных</li> </ul>
Status	Uint4	Статус опроса категории данных: <ul style="list-style-type: none"> <li>0 - опрос категории не ведётся</li> <li>1 - категория данных опрашивается</li> </ul>

## Контроль состояния источника

Полный тег служебных сигналов контроля состояния источника:



Service.Modules.<Имя модуля>.Device <Идентификатор устройства>.Source.<Имя сигнала>



Сигнал	Тип	Описание сигнала
ActivityState	Uint4	Состояние источника: <ul style="list-style-type: none"> <li>0 – не активный - обмен данными с источником не ведётся</li> <li>1 – активный - ведётся обмен данными с источником</li> </ul>

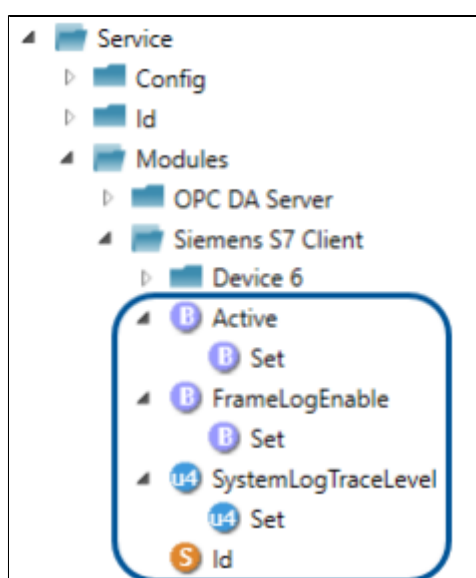
Status	Uint4	Статус ПЛК: <ul style="list-style-type: none"> <li>&gt;4 – остановлен</li> <li>&gt;8 – в работе</li> <li>&gt;9 – в работе в составе резервной пары ПЛК</li> </ul>
--------	-------	--

## Контроль и управление основными параметрами модуля

Полный тег стандартных служебных сигналов контроля и управления основными параметрами модуля имеет вид



Service.Modules.<Имя модуля>.<Имя сигнала>



Стандартные служебные сигналы:

Сигнал	Тип	Описание сигнала
Active	Bool	Активность модуля: <ul style="list-style-type: none"> <li>&gt; True – запущен</li> <li>&gt; False – остановлен</li> </ul> Соответствует значению параметра Активность. Управляется служебным сигналом «Active.Set»
FrameLogEnable	Bool	Ведение журнала работы:

		<ul style="list-style-type: none"> <li>› True – ведётся</li> <li>› False – не ведётся</li> </ul> <p>Соответствует значению параметра Вести журнал работы модуля. Управляется служебным сигналом «FrameLogEnable.Set»</p>
SystemLogTraceLevel	Uint4	<p>Уровень детализации журнала работы:</p> <ul style="list-style-type: none"> <li>› 1 – предупреждения и аварийные сообщения</li> <li>› 2 – информационные сообщения</li> <li>› 3 – отладочные сообщения</li> </ul> <p>Соответствует значению параметра Уровень трассировки в журнал приложений. Управляется служебным сигналом «SystemLogTraceLevel.Set»</p>
Id	String	Идентификатор модуля в конфигурации Astra.Server

## Журнал работы

Модуль Siemens S7 Client ведёт журнал работы, в который записывается информация о работе модуля и обмене данными с контроллером.



Чтобы модуль вёл журнал работы, в общих параметрах модуля установите параметру Вести журнал работы модуля значение Да или установите сервисному сигналу модуля FrameLogEnable.Set значение true.

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

› в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

› в Linux системах в директории:



/opt/AstraRegul//Astra.Server/Logs.

Для просмотра журнала работы модуля используется сервисное приложение Просмотрщик лога кадров.

№	Дата	Время	Описание	Номер устройства	Источник	Канал	№	ID устройства	Область данных	Адрес
1..	18.01.2023	18:46:42:701	Исходящий кадр: запрос на чтение данных, количе...	6	1 (Основной)	1 (Основной)	1	6	DB67	8
1..	18.01.2023	18:46:42:765	Исходящий кадр: запрос на чтение данных, количе...	6	1 (Основной)	1 (Основной)	2	6	DB67	10
1..	18.01.2023	18:46:42:769	Исходящий кадр: запрос на чтение данных, количе...	6	1 (Основной)	1 (Основной)	3	6	DB67	266
1..	18.01.2023	18:46:42:789	Входящий кадр: ответ на запрос чтения данных, к...	6	1 (Основной)	1 (Основной)	4	6	DB67	267
1..	18.01.2023	18:46:42:789	Входящий кадр: ответ на запрос чтения данных, к...	6	1 (Основной)	1 (Основной)	5	6	DB67	268
1..	18.01.2023	18:46:42:870	Исходящий кадр: запрос на чтение данных, количе...	6	1 (Основной)	1 (Основной)				
1..	18.01.2023	18:46:42:879	Входящий кадр: ответ на запрос чтения данных, к...	6	1 (Основной)	1 (Основной)				
1..	18.01.2023	18:46:42:971	Входящий кадр: ответ на запрос чтения данных, к...	6	1 (Основной)	1 (Основной)				
1..	18.01.2023	18:46:43:701	Исходящий кадр: запрос на чтение данных, количе...	6	1 (Основной)	1 (Основной)				
1..	18.01.2023	18:46:43:766	Исходящий кадр: запрос на чтение данных, количе...	6	1 (Основной)	1 (Основной)				
1..	18.01.2023	18:46:43:770	Исходящий кадр: запрос на чтение данных, количе...	6	1 (Основной)	1 (Основной)				
1..	18.01.2023	18:46:43:790	Входящий кадр: ответ на запрос чтения данных, к...	6	1 (Основной)	1 (Основной)				
1..	18.01.2023	18:46:43:790	Входящий кадр: ответ на запрос чтения данных, к...	6	1 (Основной)	1 (Основной)				
1..	18.01.2023	18:46:43:790	Входящий кадр: ответ на запрос чтения данных, к...	6	1 (Основной)	1 (Основной)				
1..	18.01.2023	18:46:43:871	Исходящий кадр: запрос на чтение данных, количе...	6	1 (Основной)	1 (Основной)				

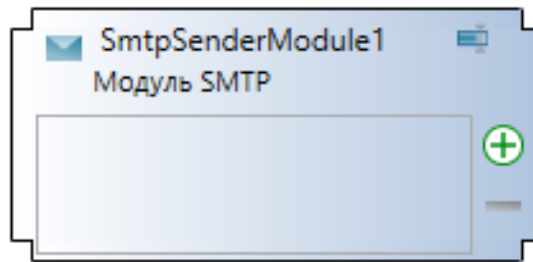
№	ID устройства	Область данных	Адрес
1	6	DB67	8
2	6	DB67	10
3	6	DB67	266
4	6	DB67	267
5	6	DB67	268

Входящий кадр: ответ на запрос чтения данных, количество значений: 7

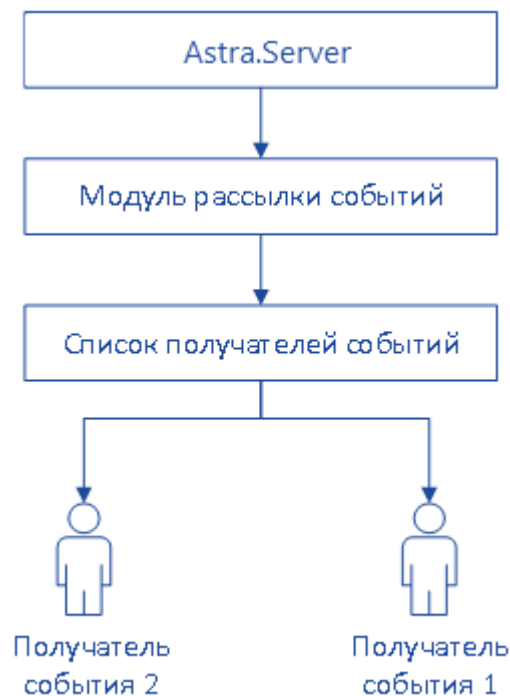
0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

Каждая запись журнала имеет порядковый номер, дату, время и описание. Записи входящих и исходящих кадров модуля дополнительно содержат номер устройства, источник и канал.

## 1.1.2.5.10. Модуль SMTP



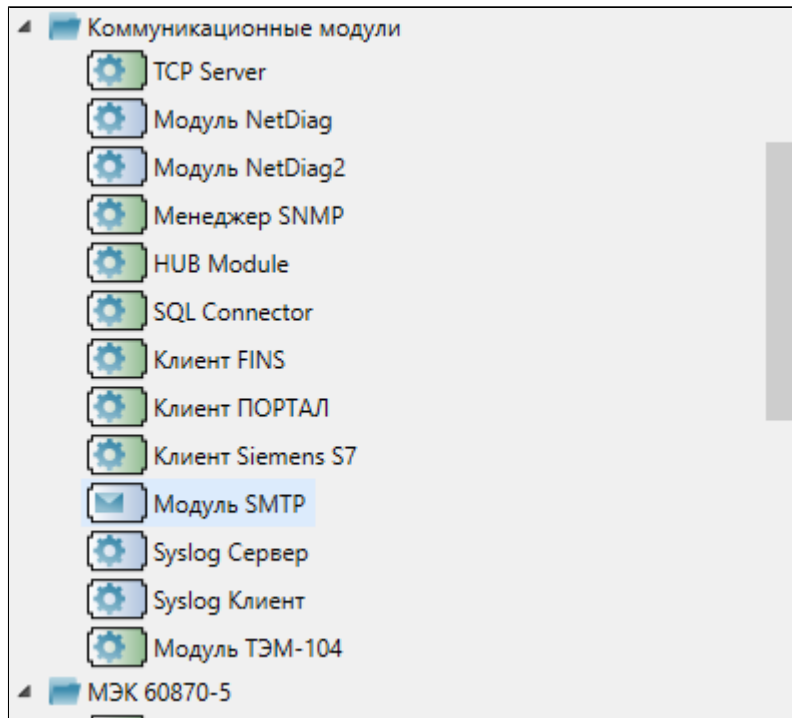
Модуль рассылки событий применяется для рассылки уведомлений по электронной почте.



Astra.Server передает события Модулю рассылки событий. Модуль отправляет события списку получателей событий. В зависимости от важности события, события распределяются по получателям событий и рассылаются на электронный адрес получателей.

# Настройка модуля

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигурирования Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.



У пользователя на SMTP сервере должно быть разрешение на рассылку сообщений. Если на SMTP сервере стоит настройка защиты от небезопасного входа, рассылки событий не будет.




SmtpSenderModule Модуль SMTP	
Общие	
Карта адресов	
Имя	SmtpSenderModule
Параметры модуля	
Имя хоста почтового сервера	smtp.gmail.com
Тип подключения	STARTTLS
Порт почтового сервера	587
Почтовый адрес отправителя	astraregul
Имя учетной записи почты	astraregul
Пароль учетной записи почты	.....
Активность	Да
Отображаемое имя	
Параметры журналирования	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Имя хоста почтового сервера	Имя или адрес SMTP-сервера
Тип подключения	Протокол шифрования, используемый при обмене данных с почтовым сервером: <ul style="list-style-type: none"> <li>&gt; STARTTLS</li> <li>&gt; SSL</li> </ul>

Порт почтового сервера	Порт SMTP-сервера
Почтовый адрес отправителя	Адрес, который будет отображен в уведомлении
Имя учетной записи почты	Имя авторизации на почтовом сервере
Пароль учетной записи почты	<p>Пароль учетной записи отправителя. Хранится и передается в зашифрованном виде.</p> <div style="border: 1px solid #add8e6; padding: 5px;">  Пароль хранится в файле конфигурации в зашифрованном виде </div>
Активность	<p>Активность модуля при запуске/перезапуске Astra.Server:</p> <ul style="list-style-type: none"> <li>› Да – модуль запущен;</li> <li>› Нет – модуль остановлен.</li> </ul> <p>Управляется служебным сигналом Active.Set</p>
Отображаемое имя	Отображаемое имя модуля

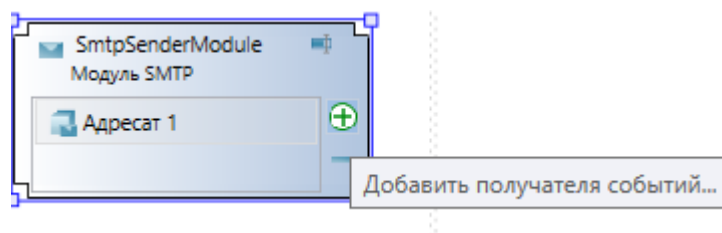
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

# Настройка списка получателей событий

Получатели событий – те, на чьи адреса электронной почты будут отправляться события заданной важности. В список получателей событий можно добавить произвольное количество получателей и каждому настроить свою важность событий.

Для настройки списка получателей событий выберите Модуль рассылки событий и добавьте нужное количество получателей событий.



## Параметры получателя событий

Получатель сообщений	
<b>Общие</b>	
Имя	Адресат 1
Адрес получателя	astraregul
Отправлять события АЕ	Да
Минимальная важность	1
Максимальная важность	1000
<b>Настройки сообщений</b>	
Минимальный период отправки сообщений, мс	100
Максимальный размер сообщения	1000
Максимальное число отправляемых изменений значения сигнала	1
<b>Настройки очереди сообщений</b>	
Размер очереди сообщений	100
Процент стираемых сообщений	10
Сообщение при превышении очереди	Очередь переполнена, {N} сообщений удалены.

Параметр	Описание
----------	----------

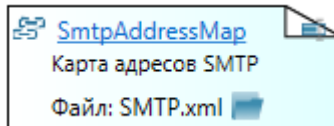
Имя	Имя, под которым будет отображаться получатель в списке получателей событий
Адрес получателя	Электронный адрес, на который будут отправляться события
Минимальная важность	Нижняя грань важности событий. Начиная с этой важности, будут отправляться события
Максимальная важность	Верхняя грань важности событий. События будут отправляться до данной важности

Чтобы приходили полностью все события на адрес получателя, укажите важность событий с 1 до 1000. Для получения событий только одной определенной важности, например 667, укажите в нижнем и верхнем порогах важности событий 667.

## 1.1.2.5.10.3. Карта адресов

Для настройки сигналов модуля SMTP используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов SMTP в исполняемое приложение;



2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	Получатели сообщения
string	string	непосредственн	
string1	string	непосредственн	

## Параметры карты адресов

Поле	Значение
Привязка	Тип привязки параметра: <ul style="list-style-type: none"><li>› непосредственно;</li><li>› только чтение;</li><li>› не привязан.</li></ul>
Получатели сообщения	Электронный адрес, на который будут отправляться события

# Диагностика работы модуля

## Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

› в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

› в Linux системах в директории:

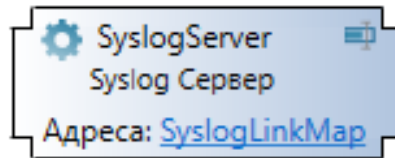


/opt/AstraRegul/Astra.Server/Logs.

Для просмотра журнала работы модуля воспользуйтесь сервисным приложением Просмотрщик лога кадров.

№	Дата	Время	Описание
0	15.03.2021	14:23:39:872	Журнал открыт для записи
1	15.03.2021	14:26:49:715	Отправляется 1 событий. (размер очереди 1)
2	15.03.2021	14:26:49:716	Resolving host smtp.yandex.ru:465
3	15.03.2021	14:26:49:813	Connecting to host smtp.yandex.ru:465
4	15.03.2021	14:26:49:892	Initializing SSL connection
5	15.03.2021	14:26:50:387	Wait server HELLO
6	15.03.2021	14:26:50:639	220 smtp1h.mail.yandex.net ESMTP (Want to use Yandex.Mail for...
7	15.03.2021	14:26:50:639	<-- Ответ SMTP:
8	15.03.2021	14:26:50:639	220 smtp1h.mail.yandex.net ESMTP (Want to use Yand...
9	15.03.2021	14:26:50:639	--> Запрос SMTP:
10	15.03.2021	14:26:50:639	EHLO smtp.yandex.ru
11	15.03.2021	14:26:50:693	250-smtp1h.mail.yandex.net250-8BITMIME250-PIPELINING250-SI...

## 1.1.2.5.11. Syslog Сервер



Модуль Syslog Server предназначен для диагностики сетевого оборудования путём сбора диагностических сообщений по протоколу Syslog.

Модуль выполняет следующие функции:

- › сбор Syslog сообщений от сетевых устройств;
- › запись полученных сообщений в сигналы.

Поддерживаемые форматы сообщений:

- › [RFC 5424](#)
- › [RFC 3164](#)

### Протокол Syslog

Протокол Syslog – способ передачи информации о событиях, происходящих в устройстве. Информация передаётся в виде текстовых сообщений.

Сообщение содержит:

- › текст события;
- › информацию о событии: где и когда оно произошло, важность и пр.



Формат сообщения зависит от спецификации, согласно которой формируется сообщение.

Сообщения передаются по схеме клиент-сервер: при возникновении события устройство формирует сообщение и отправляет его серверу Syslog. Сервер Syslog принимает и обрабатывает сообщение. Правила обработки протоколом не регламентируются и зависят от реализации сервера.

Сообщения передаются в открытом виде (по транспортному протоколу UDP) или с использованием шифрования (по транспортному протоколу TLS).

## Принципы работы

Модуль Syslog Server принимает Syslog сообщения от различных сетевых устройств и записывает полученные сообщения в подготовленные для них сигналы. Каждое сообщение записывается только в один сигнал сервера.

В сервере можно настроить генерацию OPC AE событий при записи Syslog сообщений в сигналы сервера. Настройка генерации событий описана в приложении.

Сообщения неизвестного формата могут быть записаны в сигнал только в исходном виде.

Сообщения известного формата могут быть записаны в сигнал в исходном виде либо в преобразованном для генерации динамических событий модулем OPC AE Server: это регулируется настройкой Генерировать события модуля Syslog Server.

Если включена настройка Генерировать события, то можно распределять сообщения по разным сигналам в соответствии с настроенными фильтрами в свойстве адреса сигналов.

Если сообщения записываются в сигнал в исходном виде, можно разбирать их с помощью модуля логики, формировать строку для генерации динамического события и записывать её в сигнал, который генерирует события.

VQT значение, записываемое в сигнал:

› Value (значение) – строка



```
<Subcondition Type="Dynamic" Message="MSG" />
```



MSG – значение поля MSG в полученном сообщении.



Формат значения предназначен для автоматической генерации OPC AE события при его записи в сигнал

- › Quality (качество) – 192 (хорошее);
- › Timestamp (метка времени) – значение поля TIMESTAMP в полученном сообщении.

## Работа в резерве

При работе в состоянии РЕЗЕРВ модуль не принимает сообщения от сетевых устройств.

## Работа по TLS/UDP

При работе с транспортным протоколом TLS сетевое устройство и модуль устанавливают между собой безопасное соединение. Для этого они обмениваются сертификатами. Модуль принимает сообщения только от тех сетевых устройств, сертификат которых есть в папке доверенных сертификатов.

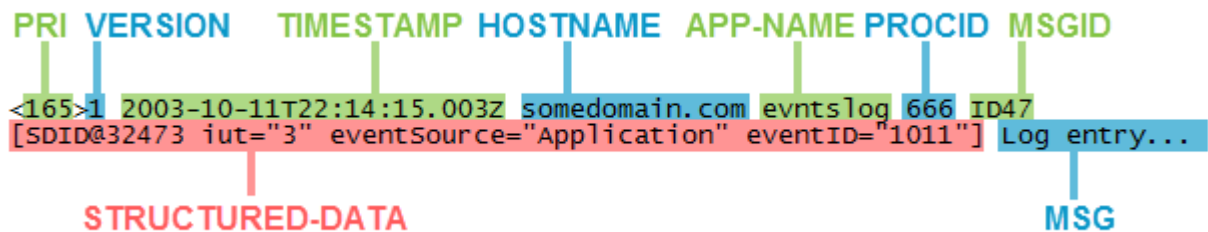
При работе с транспортным протоколом UDP модуль принимает сообщения от любых сетевых устройств.

## RFC 5424


Syslog сообщение в формате RFC 5424 имеет следующую структуру:

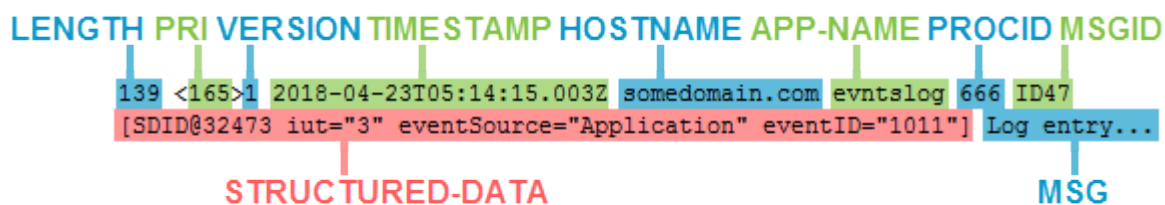


```
<PRI>VERSION TIMESTAMP HOSTNAME APPNAME PROCID MSGID  
[STRUCTURED-DATA] MSG
```



При передаче Syslog сообщений по протоколу TLS в начале сообщения указывается его размер:

 LENGTH <PRI>VERSION TIMESTAMP HOSTNAME APPNAME PROCID MSGID [STRUCTURED-DATA] MSG




Согласно спецификации RFC 5424 в сообщении могут быть не указаны значения следующих полей:


- › HOSTNAME (Идентификатор хоста)
- › APPNAME (Идентификатор приложения)
- › PROCID (Идентификатор процесса)
- › MSGID (Тип сообщения)

Вместо отсутствующего значения в поле должен быть указан символ -.

## RFC 3164

 Syslog сообщения в формате RFC 3164 передаются только по транспортному протоколу UDP.

Syslog сообщение в формате RFC 3164 имеет следующую структуру:

 <PRI> TIMESTAMP HOSTNAME APPNAME [PROCID]:MSG

```

PRI   TIMESTAMP   HOSTNAME APPNAME   PROCID   MSG
|     |           |         |         |         |
<34> Oct 11 22:14:15 mymachine su[10]: Log entry...

```

Согласно спецификации RFC 3164 в сообщении могут быть не указаны значения любых полей. В этом случае значения соответствующих полей будут отсутствовать в структуре сообщения (допускается передача пустого сообщения, не содержащего ни одного символа):

- если в сообщении не указано значение **TIMESTAMP**, то в качестве значения поля используется момент получения сообщения;
- если в сообщении не указано значение **PRI**, то в качестве значения поля **PRI** используется значение 41 (Facility – 5, Severity – 1);
- при отсутствии значений прочих полей, в качестве значения этих полей используется пустая строка.

Формату RFC 3164 соответствуют любые сообщения, которые могут быть получены: если при попытке разобрать отдельные поля в полученном сообщении произойдет ошибка, то вся строка сообщения будет использоваться в качестве значения поля **MSG**.

## Поля сообщения Syslog

Поле	Описание						
LENGTH	Количество символов в сообщении. Указывается только в сообщениях, передаваемых по протоколу TLS						
PRI	<p>Приоритет. Содержит два значения: Facility (субъект, сформировавший сообщение) и Severity (уровень важности сообщения). Severity принимает значения от 0 до 7, Facility - от 0 до 23. Значение PRI вычисляется как <math>PRI = Facility * 8 + Severity</math>.</p> <p>Facility (Тип оборудования): Severity (Важность сообщения):</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 50%;">0 – Ядро операционной системы</td> <td style="width: 50%;">0 – Авария, система неработоспособна</td> </tr> <tr> <td>1 – ПО пользователя</td> <td>1 – Тревога, система требует немедленного вмешательства</td> </tr> <tr> <td>2 – Почтовая система</td> <td></td> </tr> </table>	0 – Ядро операционной системы	0 – Авария, система неработоспособна	1 – ПО пользователя	1 – Тревога, система требует немедленного вмешательства	2 – Почтовая система	
0 – Ядро операционной системы	0 – Авария, система неработоспособна						
1 – ПО пользователя	1 – Тревога, система требует немедленного вмешательства						
2 – Почтовая система							

	<p>3 – Системные службы  4 – Сообщения безопасности/авторизации  5 – Собственные сообщения syslogd  6 – Подсистема печати  7 – Подсистема новостных групп  8 – Подсистема UUCP  9 – Службы времени  10 – Сообщения безопасности/авторизации  11 – Служба FTP  12 – Подсистема NTP  13 – Сообщения аудита  14 – Аварийные сообщения  15 – Службы времени 2  16-23 – локальное происхождение 0-7</p>	<p>2 – Состояние системы критическое  3 – Сообщения об ошибках  4 – Предупреждения о возможных проблемах  5 – Сообщения о нормальных, но важных событиях  6 – Информационные сообщения  7 – Отладочные сообщения</p>
VERSION	Версия протокола	
TIMESTAMP	Метка времени сообщения	
HOSTNAME	IP-адрес или сетевое имя устройства, сгенерировавшего сообщение	
APP-NAME	Идентификатор устройства или приложения, сгенерировавшего сообщение	
PROCID	Позволяет регистрировать перерывы в потоке данных и обычно используется в качестве идентификатора процесса	
MSGID	Идентификатор типа сообщения	
STRUCTURED-DATA	Структурированные данные. В работе модуля Syslog Server не используются	
MSG	Текст сообщения	

Поля Syslog сообщений используются следующим образом:

- › поля MSG и TIMESTAMP используются при формировании VQT значения, записываемого в сигнал.
- › поле HOSTNAME используется при поиске сигнала, предназначенного для записи сообщений от устройства, указанного в этом поле.
- › поля PRI, APP-NAME, PROCID, MSGID используются при поиске сигнала, имеющего соответствующие параметры фильтрации.

## Генерация событий

Генерацию событий в сервере выполняет модуль OPC AE Server. События генерируются при записи Syslog сообщений в настроенные сигналы.

Для полученного Syslog сообщения в сервере не будет сгенерировано событие в следующих случаях:

- › если в сервере нет подходящего сигнала для записи сообщения
- › если для сигнала, в который сообщение было записано, не настроена генерация событий

Для каждого сигнала, в который записываются Syslog сообщения, необходимо:

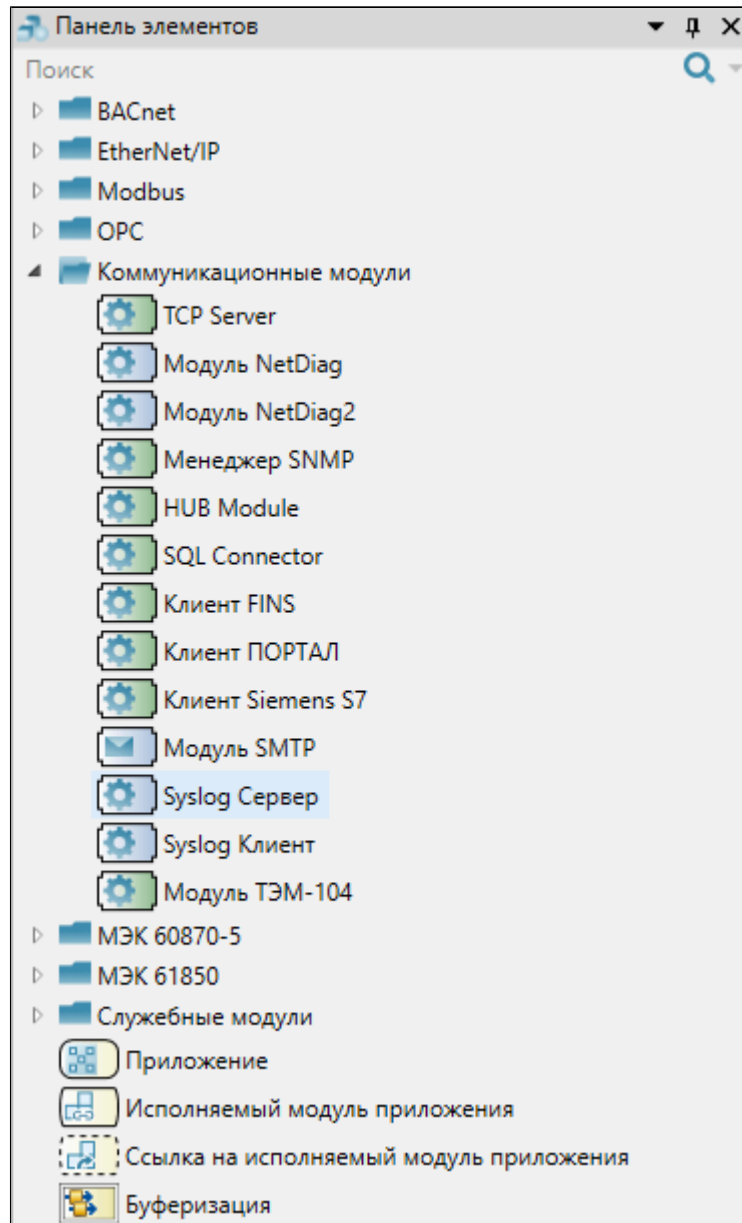
- › настроить в свойствах родительской папки условие генерации события. Тип условия – Dynamic.
- › поставить сигнал на обслуживание модулю OPC AE Server с параметром Conditions=(CONDITION\_NAME), где CONDITION\_NAME – это название условия, указанное в свойствах папки.



Данный список содержит общее описание необходимых действий по настройке генерации событий. Подробное описание выполняемых настроек приведено в документации на модуль OPC AE Server.

## 1.1.2.5.11.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

Свойства	
SyslogServer Syslog Сервер	
Общие	
Карта адресов	IOS_App.SyslogLinkMap
Генерировать события	Да
Транспортный протокол	TLS
Номер TCP порта	6514
Путь к сертификату сервера	C:\Windows\cert
Путь к приватному ключу сервера	C:\Windows\keys
Имя	SyslogServer
Параметры модуля	
Активность	Да
Отображаемое имя	
Параметры журналирования	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения

## Общие

Параметр	Описание
Карта адресов	Карта адресов Syslog, которая будет использована для данного адаптера
Генерировать события	Значения: <ul style="list-style-type: none"> <li>➤ Да</li> <li>➤ Нет</li> </ul> "Да" - syslog сообщение будет записано в преобразованном виде: <Subcondition Type="Dynamic" Message="Message Text" /> "Нет" - syslog сообщение будет записано в исходном виде.
Транспортный протокол	Тип транспортного протокола, используемого для передачи Syslog сообщений: <ul style="list-style-type: none"> <li>➤ TLS</li> <li>➤ UDP</li> </ul>

Номер TCP порта	Порт приема данных
Путь к сертификату сервера	Абсолютный путь к сертификату сервера или имя сертификата сервера. Используется только при работе по протоколу TLS
Путь к приватному ключу сервера	Абсолютный путь к приватному ключу сервера или только имя приватного ключа сервера. Используется только при работе по протоколу TLS
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>➤ Да – модуль запущен;</li> <li>➤ Нет – модуль остановлен.</li> </ul>
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.



Для получения Syslog сообщений по обоим транспортным протоколам, добавьте в состав Astra.Server два модуля Syslog Server. В этом случае описанная настройка (в том числе настройка сигналов) выполняется для каждого добавленного модуля независимо.



## 1.1.2.5.11.2. Передача данных

### Передача данных по TLS

При работе с транспортным протоколом TLS сетевое устройство и сервер устанавливают между собой безопасное соединение. Для этого сетевое устройство и сервер обмениваются сертификатами.

Сервер принимает Syslog сообщения по протоколу TLS только от доверенных сетевых устройств. Устройство является доверенным, если его сертификат есть в папке доверенных сертификатов



C:\ProgramFiles\AstraRegul\Astra.Server\Certificates\TrustedPeers\certs

Чтобы настроить приём сообщений по протоколу TLS, в параметрах модуля:

1. Укажите тип транспортного модуля TLS.
2. Укажите порт приёма данных.

По умолчанию порт для протокола TLS – 6514.

3. Укажите абсолютный путь к сертификату сервера или только имя сертификата сервера, расположенного в папке



C:\ProgramFiles\AstraRegul\Astra.Server\Certificates\TrustedPeers\certs

4. Укажите абсолютный путь к приватному ключу сервера или только имя приватного ключа сервера, расположенного в папке



C:\ProgramFiles\AstraRegul\Astra.Server\Certificates\private.



Сертификат сервера и приватный ключ сервера создаются в процессе установки Astra.Server.

Чтобы добавить доверенное устройство, добавьте его сертификат в папку доверенных устройств.

Если сертификат устройства не будет найден в папке доверенных сертификатов при попытке установить соединение с сервером, то сертификат устройства, полученный при попытке соединения, будет сохранён в папке отклонённых сертификатов.



C:\ProgramFiles\AstraRegul\Astra.Server\Certificates\Rejected\certs

При каждой попытке соединения в папке отклонённых сертификатов будет создана отдельная копия полученного сертификата.

Чтобы сделать устройство доверенным, переместите любую копию сертификата этого устройства в папку доверенных сертификатов.

## Передача данных по UDP

При работе по транспортному протоколу UDP сетевому устройству и серверу не требуется устанавливать между собой безопасное соединение. Сервер принимает все Syslog сообщения, передаваемые по протоколу UDP.

Чтобы настроить приём сообщений по протоколу UDP, в параметрах модуля:

1. Укажите тип транспортного протокола – UDP.
2. Укажите порт приема данных.

По умолчанию порт для протокола UDP – 514.

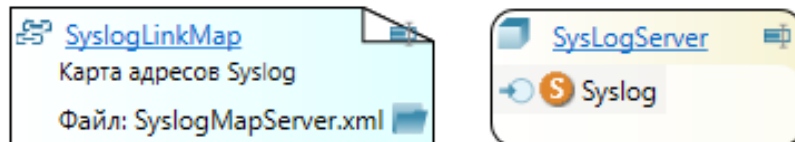


Параметры "Путь к сертификату сервера" и "Путь к приватному ключу сервера" указывать не нужно.

## 1.1.2.5.11.3. Карта адресов

Для настройки сигналов модуля Syslog Сервер используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов Syslog в исполняемое приложение;



2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	Получатель сообщения	Формат	Адрес отправителя	Тип оборудования	Важность	Приложение	Процесс	Тип сообщения
SysLogServer.Syslog	string	непосредственно								



Модуль Syslog Сервер принимает на обслуживание только строковые сигналы.

## Параметры карты адресов

Параметр	Значение
Привязка	Тип привязки параметра: <ul style="list-style-type: none"><li>&gt; непосредственно</li><li>&gt; только чтение</li><li>&gt; не привязан</li></ul>
Адрес отправителя	IP-адрес или сетевое имя устройства, сгенерировавшего сообщение. Используется как параметр фильтрации
Тип оборудования	Используется как параметр фильтрации по типу оборудования: <ul style="list-style-type: none"><li>&gt; Ядро операционной системы</li><li>&gt; ПО пользователя</li><li>&gt; Почтовая система</li><li>&gt; Системные службы</li></ul>


	<ul style="list-style-type: none"> <li>› Сообщения безопасности/авторизации</li> <li>› Собственные сообщения syslog</li> <li>› Подсистема печати</li> <li>› Подсистема новостных групп</li> <li>› Подсистема UUCP</li> <li>› Службы времени</li> <li>› Сообщения безопасности/авторизации</li> <li>› Служба FTP</li> <li>› Подсистема NTP</li> <li>› Сообщения аудита</li> <li>› Аварийные сообщения</li> <li>› Службы времени 2</li> <li>› локальное происхождение 0-7</li> </ul>
Важность сообщения	<p>Используется как параметр фильтрации по важности:</p> <ul style="list-style-type: none"> <li>› Авария, система неработоспособна</li> <li>› Тревога, система требует немедленного вмешательства</li> <li>› Состояние системы критическое</li> <li>› Сообщения об ошибках</li> <li>› Предупреждения о возможных проблемах</li> <li>› Сообщения о нормальных, но важных событиях</li> <li>› Информационные сообщения</li> <li>› Отладочные сообщения</li> </ul>
Приложение	Идентификатор устройства или приложения, сгенерировавшего сообщение
Процесс	Позволяет регистрировать перерывы в потоке данных и обычно используется в качестве идентификатора процесса
Тип сообщения	Идентификатор типа сообщения

# 1.1.2.5.11.4. Диагностика работы модуля


## Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.aplog по умолчанию:

› в ОС Windows в папке:

 C:\Program Files\AstraRegul\Astra.Server\Logs;

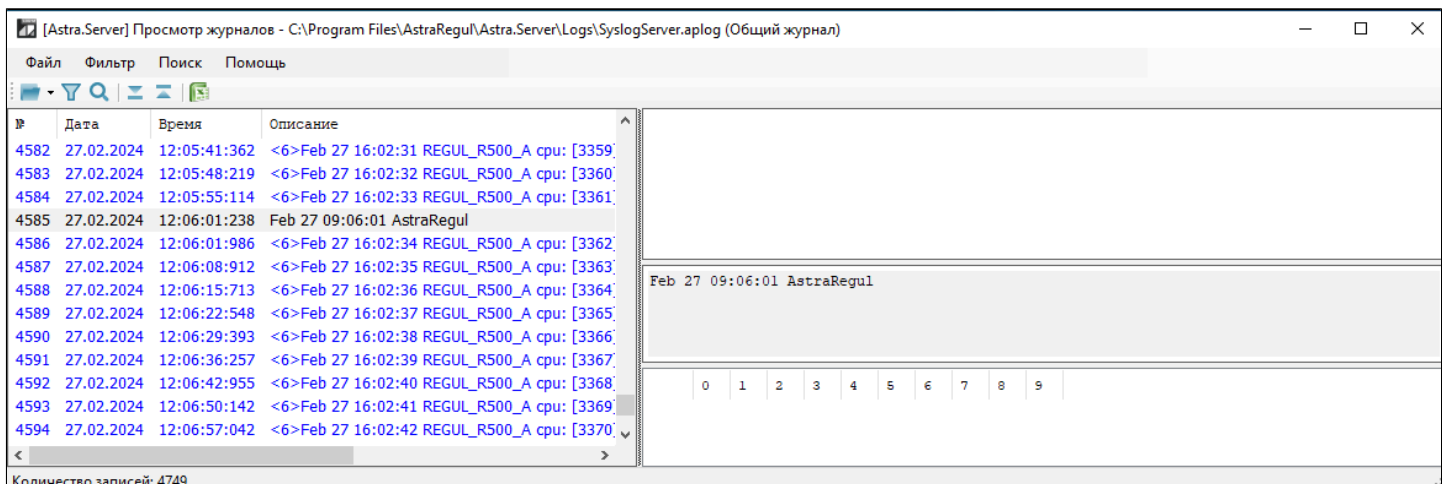
› в Linux системах в директории:

 /opt/AstraRegul/Astra.Server/Logs.

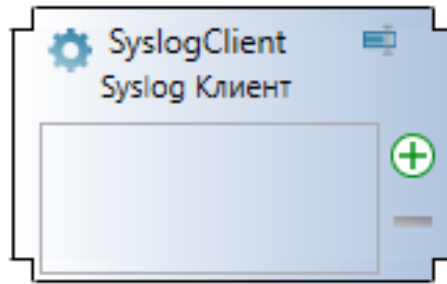
Для просмотра журнала работы модуля воспользуйтесь сервисным приложением Просмотрщик лога кадров.

В журнал работы модуля записываются:

- › все полученные Syslog сообщения;
- › уведомления об отсутствии в сервере сигнала, подходящего для записи полученного сообщения;
- › при работе по протоколу TLS – уведомления о неверном формате полученного сообщения.



## 1.1.2.5.12. Syslog Клиент



Модуль Syslog Client предназначен для отправки Syslog сообщений на указанные Syslog сервера.

Модуль выполняет следующие функции:

- › Формирование Syslog сообщений;
- › Отправка Syslog сообщений получателям.

### Протокол Syslog

Протокол Syslog – способ передачи информации о событиях, происходящих в устройстве. Информация передаётся в виде текстовых сообщений.

Сообщение содержит:

- › текст события;
- › информацию о событии: где и когда оно произошло, важность и пр.



Формат сообщения зависит от спецификации, согласно которой формируется сообщение.

Сообщения передаются по схеме клиент-сервер: при возникновении события устройство формирует сообщение и отправляет его серверу Syslog. Сервер Syslog принимает и обрабатывает сообщение. Правила обработки протоколом не регламентируются и зависят от реализации сервера.

Сообщения передаются в открытом виде (по транспортному протоколу UDP) или с использованием шифрования (по транспортному протоколу TLS).

## Принципы работы

При изменении значения сигнала, модуль генерирует Syslog сообщение и помещает его в очередь получателя сообщений. Далее через заданный период отправки, сообщение достаётся из очереди и отправляется получателю по заданному транспортному протоколу. В случае когда в сигнал будет записана пустая строка, сообщение генерироваться и отправляться не будет. В резервном режиме модуль не генерирует и не отправляет сообщения.

## Принципы работы очереди сообщений

У каждого получателя своя очередь сообщений. Она работает по принципу FIFO (первым пришёл — первым вышел), то есть первыми будут переданы те сообщения, которые были помещены в очередь первыми. Для ограничения потребления памяти очередью, пользователь может задать в настройках получателя максимальный размер очереди и тогда количество сообщений в очереди не будет превышать заданное значение. При обнаружении переполнения в очереди сообщений из ее начала удаляется N сообщений (количество удаляемых сообщений зависит от параметра "Процент стираемых сообщений"). Если процент стираемых сообщений равен 0, то поступившее сообщение добавляется в конец очереди, при этом удаляется сообщение из начала очереди.

## Работа по TLS/UDP

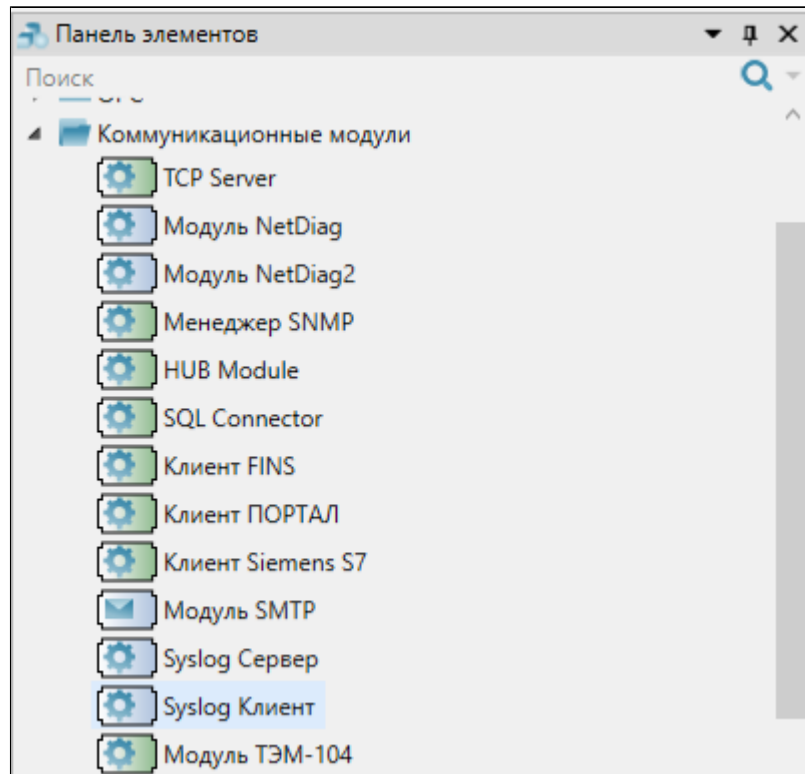
При работе с транспортным протоколом TLS сетевое устройство и модуль устанавливают между собой безопасное соединение. Для этого они обмениваются сертификатами. Модуль принимает сообщения только от тех сетевых устройств, сертификат которых есть в папке доверенных сертификатов.

При работе с транспортным протоколом UDP модуль принимает сообщения от любых сетевых устройств.



## 1.1.2.5.12.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуриатора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

SyslogClient Syslog Клиент	
Общие	
Карта адресов	Runtime.Application.SyslogLinkMap
Имя	SyslogClient
Параметры модуля	
Активность	Да
Отображаемое имя	
Параметры журналирования	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения

## Общие

Параметр	Описание
Карта адресов	Карта адресов Syslog, которая будет использована для данного адаптера
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"><li>› Да – модуль запущен;</li><li>› Нет – модуль остановлен.</li></ul>
Отображаемое имя	Отображаемое имя модуля

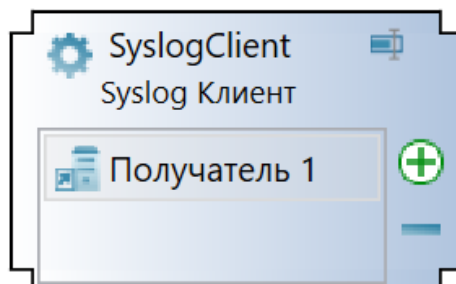
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## Настройка получателей сообщений

В модуле Syslog Клиент необходимо настроить получателей сообщений. Количество получателей не ограничено. Для добавления получателей нажмите

кнопку .



Получатель сообщений	
Общие	
Имя	Получатель 1
Транспортный протокол	UDP
IP адрес	127.0.0.1
Порт	514
Формат сообщения	RFC 3164
Размер сообщения	1024
Период отправки сообщений, мс	100
Размер очереди сообщений	100
Процент стираемых сообщений	10

## Основные параметры модуля

Параметр	Описание
Имя	Имя получателя
Транспортный протокол	Задаёт протокол для передачи Syslog сообщения. На данном этапе доступен только UDP
IP адрес	Задаёт ip адрес Syslog сервера, значение по умолчанию 127.0.0.1
Порт	Задаёт порт Syslog сервера, значение по умолчанию 514

## Настройки сообщений

Параметр	Описание
Формат сообщений	Задаёт формат сообщения. На данном этапе доступен только RFC 3164. Значение по умолчанию RFC 3164
Размер сообщения	Задаёт максимальное количество символов в сообщении. При указании 0 не контролируется. Если количество символов в сообщении превышает данное значение, то сообщение будет обрезано до максимально возможного количества символов. Значение по умолчанию 1024

Период отправки сообщений	Задаёт частоту отправки сообщений. При указании 0 не контролируется. Значение по умолчанию 100
---------------------------	--

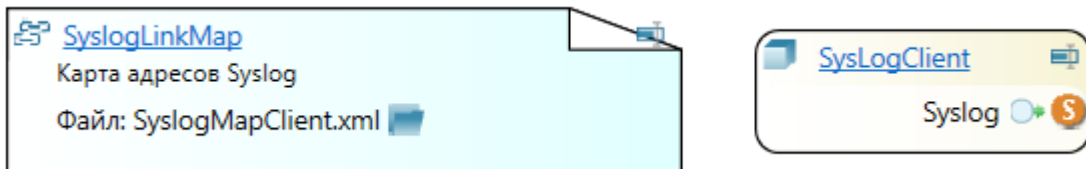
## Настройки очереди сообщений

Параметр	Описание
Размер очереди сообщений	Задаёт максимальный размер очереди сообщений. При указании 0 не контролируется. Значение по умолчанию 100
Процент стираемых сообщений	задаёт процент сообщений, которые будут удалены из очереди при её переполнении. При указании 0 удаление не выполняется. При указании 100 отчищается вся очередь. Значение по умолчанию 10

## 1.1.2.5.12.2. Карта адресов

Для настройки сигналов модуля Syslog Клиент используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов Syslog в исполняемое приложение:



2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	Получатель сообщения	Формат	Адрес отправителя	Тип оборудо	Важность	Приложение	Процесс	Тип сообщени
SysLogClient.Syslog	string	непосредственно	Получатель 1	RFC	%NodeNameAuto%			%AppNameAuto%	%ProcIdAuto%	



Модуль Syslog Клиент принимает на обслуживание только строковые сигналы.

## Параметры карты адресов

Параметр	Значение
Привязка	Тип привязки параметра: <ul style="list-style-type: none"><li>› непосредственно;</li><li>› только чтение;</li><li>› не привязан.</li></ul>
Получатель сообщения	Задаёт имя получателя, которому будут отправляться сообщения. Если не указать получателя, то сообщения будут отправлять всем получателям из конфигурации модуля
Формат	Формат: <ul style="list-style-type: none"><li>› RFC (модуль конструирует сообщение по формату RFC, используя параметры свойства 5000 и значение сигнала)</li><li>› Manual (модуль считает, что значение сигнала - готовое Syslog сообщение, и отправляет его как есть)</li></ul>

Адрес отправителя	<p>Возможные значения поля:</p> <ul style="list-style-type: none"> <li>› %NodeNameAuto% - происходит подстановка имени устройства</li> <li>› %FQDNAuto% - происходит подстановка полного доменного имени устройства</li> <li>› Любое другое значение интерпретируется как значение, которое нужно использовать в данном поле в сообщении Syslog</li> <li>› Если значение не указано поле пропускается</li> </ul>
Тип оборудования	<p>Используется как параметр фильтрации по типу оборудования:</p> <ul style="list-style-type: none"> <li>› Ядро операционной системы</li> <li>› ПО пользователя</li> <li>› Почтовая система</li> <li>› Системные службы</li> <li>› Сообщения безопасности/авторизации</li> <li>› Собственные сообщения syslog</li> <li>› Подсистема печати</li> <li>› Подсистема новостных групп</li> <li>› Подсистема UUCP</li> <li>› Службы времени</li> <li>› Сообщения безопасности/авторизации</li> <li>› Служба FTP</li> <li>› Подсистема NTP</li> <li>› Сообщения аудита</li> <li>› Аварийные сообщения</li> <li>› Службы времени 2</li> <li>› локальное происхождение 0-7</li> </ul>
Важность сообщения	<p>Используется как параметр фильтрации по важности:</p> <ul style="list-style-type: none"> <li>› Авария, система неработоспособна</li> <li>› Тревога, система требует немедленного вмешательства</li> <li>› Состояние системы критическое</li> <li>› Сообщения об ошибках</li> <li>› Предупреждения о возможных проблемах</li> </ul>


	<ul style="list-style-type: none"> <li>› Сообщения о нормальных, но важных событиях</li> <li>› Информационные сообщения</li> <li>› Отладочные сообщения</li> </ul>
Приложение	<p>Идентификатор устройства или приложения, сгенерировавшего сообщение.</p> <p>Возможные значения поля:</p> <ul style="list-style-type: none"> <li>› %AppNameAuto% - имя процесса (имя процесса сервера)</li> <li>› Любое другое значение интерпретируется как значение, которое нужно использовать в данном поле в сообщении Syslog</li> <li>› Если значение не указано поле пропускается</li> </ul>
Процесс	<p>Позволяет регистрировать перерывы в потоке данных и обычно используется в качестве идентификатора процесса</p> <p>Возможные значения поля:</p> <ul style="list-style-type: none"> <li>› %ProclIdAuto% - идентификатор процесса (идентификатор процесса сервера)</li> <li>› Любое другое значение интерпретируется как значение, которое нужно использовать в данном поле в сообщении Syslog</li> <li>› Если значение не указано поле пропускается</li> </ul>
Тип сообщения	Идентификатор типа сообщения

# 1.1.2.5.12.3. Диагностика работы модуля


## Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.aplog по умолчанию:

› в ОС Windows в папке:

 C:\Program Files\AstraRegul\Astra.Server\Logs;

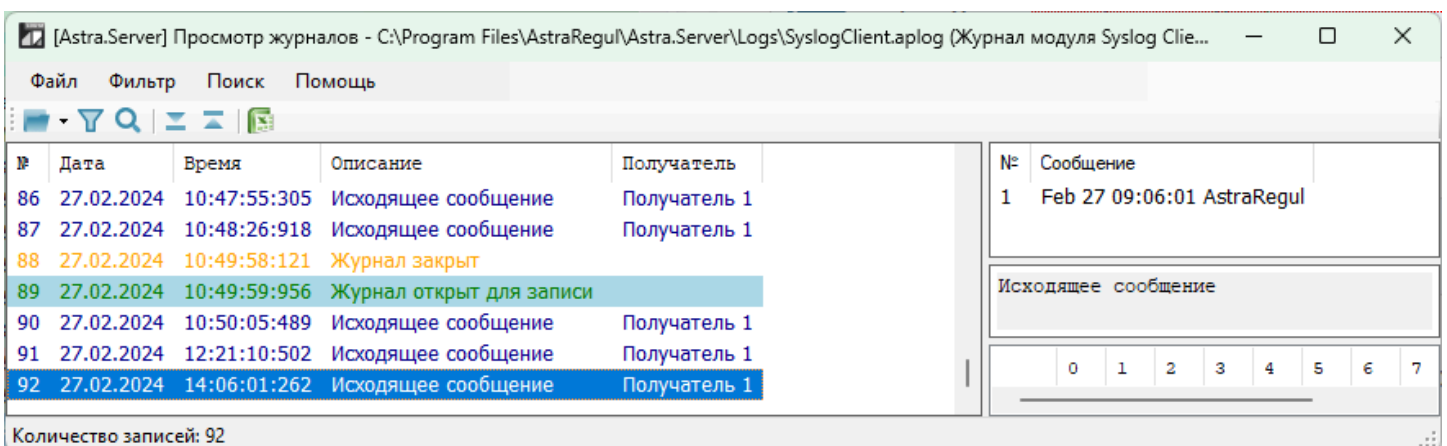
› в Linux системах в директории:

 /opt/AstraRegul/Astra.Server/Logs.

Для просмотра журнала работы модуля воспользуйтесь сервисным приложением Просмотрщик лога кадров.

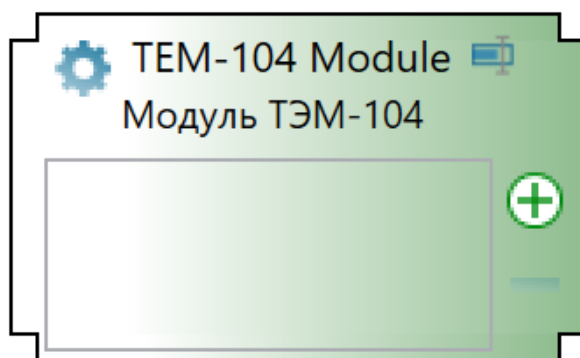
В журнал работы модуля записываются:

- › Ошибки и предупреждения, возникающие при работе модуля;
- › Все отправленные Syslog сообщения;
- › Случаи превышения максимального размера сообщения;
- › Случаи превышения максимального размера очереди (с указанием количества удалённых сообщений).





## 1.1.2.5.13. ТЭМ-104



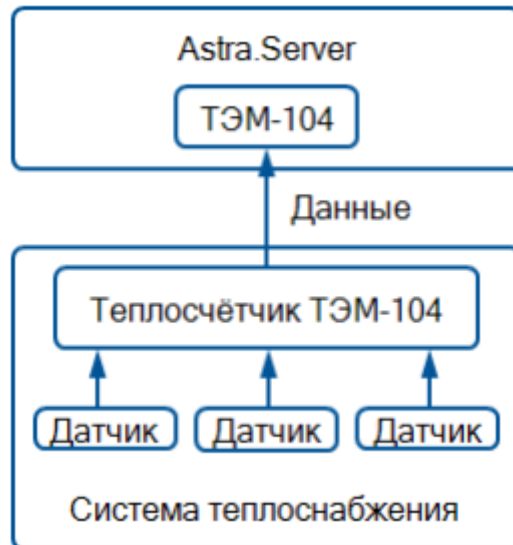
Модуль ТЭМ-104 предназначен для сбора данных из систем теплоснабжения.

Система теплоснабжения – система передачи и потребления тепла или теплоносителя (обычно – вода).

Система включает в себя:

- Потребители тепла/теплоносителя – батареи, краны подачи горячей воды и пр.
- Подающий/обратный трубопровод – трубы, по которым теплоноситель движется к/от потребителям тепла.
- Датчики – измеряют параметры теплоносителя в подающем/обратном трубопроводе.
- Теплосчётчик ТЭМ-104 – электронно-вычислительное устройство, которое:
  - Получает данные от датчиков.
  - Вычисляет параметры системы: потреблённое количества тепла, расход теплоносителя и прочие.
  - Записывает параметры системы во внутреннюю память.

Модуль ТЭМ-104 собирает данные с теплосчётчиков ТЭМ-104 и записывает их в сигналы сервера.



Сбор данных выполняется периодически в течение всего времени работы. Каждый цикл модуль собирает с устройства все данные, которые необходимо обновить (данные, для которых есть сигналы).

Сбор данных выполняется по схеме Master-Slave: модуль посылает устройству запрос, устройство посылает модулю ответ, содержащий запрошенные данные. Каждый цикл модуль посылает устройству столько запросов, сколько необходимо, чтобы получить все данные, которые нужно обновить.

Если модуль не получает ответ, он отправляет запрос повторно. По истечении количества попыток, связь с устройством считается потерянной. С устройствами, с которыми нет связи, модуль периодически пытается установить связь.

Запросы передаются через COM-порт. COM-порт позволяет одновременно выполнять только один запрос: при отправке нескольких запросов (одному устройству или нескольким) модуль посылает каждый запрос только после получения ответа на предыдущий.

## Работа в резерве

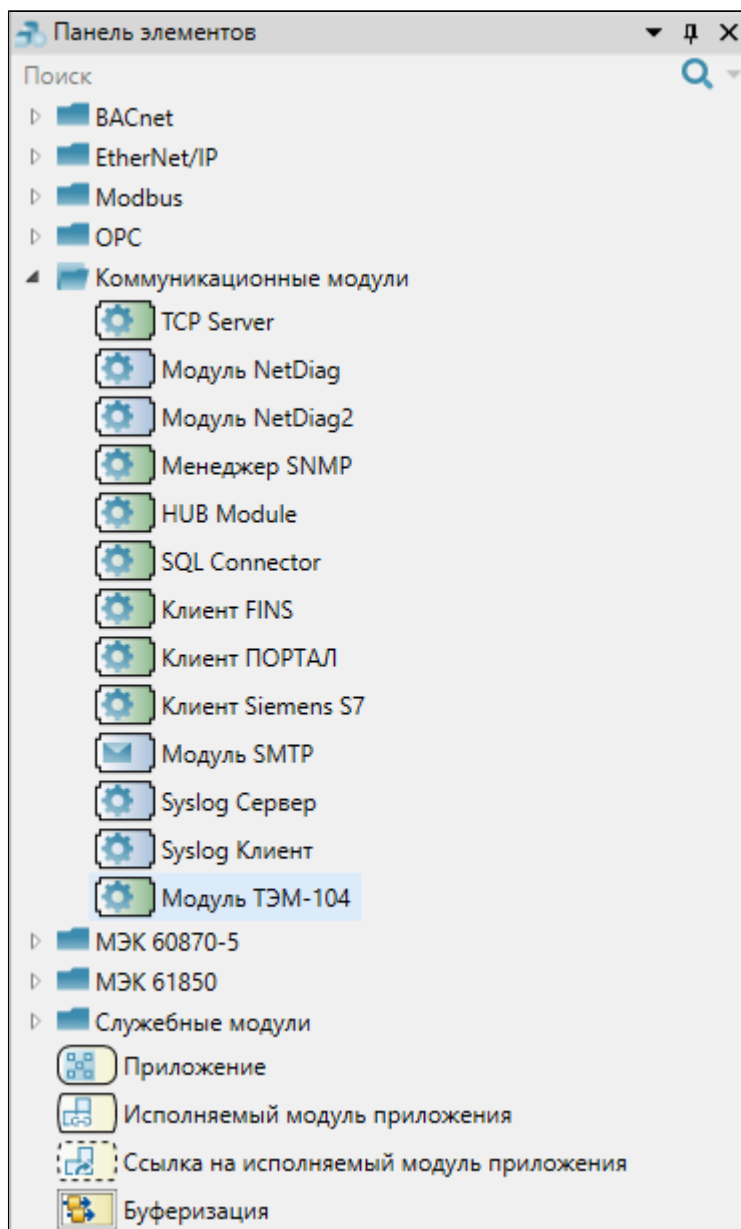
В режиме РЕЗЕРВ модуль не собирает данные.

## Работа в Astra.AccessPoint

Модуль недоступен в составе Astra.AccessPoint.

# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

Свойства	
<b>ТЕМ-104 Module</b> Модуль ТЭМ-104	
<b>Параметры модуля</b>	
Таймаут выполнения запроса, мс.	500
Количество попыток выполнения запроса	5
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	ТЕМ-104 Module

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Таймаут выполнения запроса, мс	Значение по умолчанию: 500
Количество попыток выполнения запроса	Значение по умолчанию: 5
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен</li> <li>&gt; Нет – модуль остановлен</li> </ul>

Отображаемое имя	Отображаемое имя модуля
------------------	-------------------------

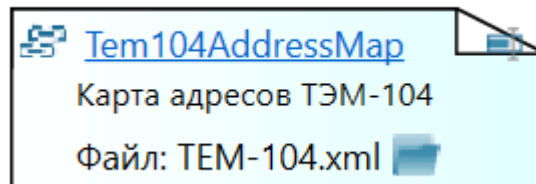
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.






## 1.1.2.5.13.2. Карта адресов

Для настройки сигналов модуля ТЭМ-104 используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов ТЭМ-104 в исполняемое приложение;



2. Откройте редактор карты адресов.

	Сигнал	Тип	Привязка	Источник данных	Параметр	Система	Канал
	▼	▼	▼	▼	▼	▼	▼
	string	string	не привязан				
	string1	string	не привязан				
	bool	bool	не привязан				
	uint2	uint2	не привязан				

## Параметры карты адресов

Параметр	Значение
Привязка	Тип привязки параметра: <ul style="list-style-type: none"><li>› непосредственно</li><li>› только чтение</li><li>› не привязан</li></ul>
Источник данных	Тип данных: <ul style="list-style-type: none"><li>› Оперативный – оперативные данные</li><li>› Исторический за последний час – статистика за час</li><li>› Исторический за последние сутки – статистика за сутки</li><li>› Исторический за отчетный период – статистика за отчетный период</li></ul>

	<p>› Исторические данные запрашиваются за последний сохранённый в устройстве час/сутки/отчётный период</p>
Параметр	Запрашиваемый параметр. Список доступных параметров зависит от типа сигнала и выбранного источника данных
Система	Номер системы. Указывается, если в устройстве каждой системе соответствует своё значение параметра, в противном случае параметр недоступен. Возможные значения: от 1 до 4
Канал	Номер канала связи с устройством. Указывается, если в устройстве каждому каналу связи с устройством соответствует своё значение параметра, в противном случае параметр недоступен. Возможные значения: от 1 до 4



## 1.1.2.5.13.3. Теплосчетчик

### Системы теплоснабжения

Системы – системы теплоснабжения, к которым подключён теплосчётчик и для которых регистрирует данные. Теплосчётчик может иметь от 1 до 4 систем.

### Типы систем

Система	Описание
Расходомер V	Измерение и регистрация объёмного расхода
Расходомер М	Измерение и регистрация массового расхода
Магистраль	Трубопровод системы теплоснабжения
Подача	Закрытая система теплоснабжения с преобразователем расхода на подающем трубопроводе
Обратка	Закрытая система теплоснабжения с преобразователем расхода на обратном трубопроводе
Тупиковая ГВС	ГВС без циркуляции
Подпитка НСО	Подпитка независимой системы теплоснабжения
Подпитка источника	Подпитка источника тепла
Подача + Р	Закрытая система теплоснабжения с контрольным преобразователем расхода на обратном трубопроводе
Открытая	Система теплоснабжения с отбором теплоносителя и приравненные к ним: промышленные потребители ЦТП, потребители с тепловой нагрузкой более 2Гкал/ч
ГВС с рециркуляцией	Циркуляционная система ГВС
Источник	Источник тепла (котельная)

### Данные

В теплосчётчике ТЭМ-104 данные хранятся в нескольких типах памяти:

- › Оперативные данные
  - › Память таймера 128 байт
  - › Память таймера 2К байт
  - › Оперативная память
- › Исторические данные
  - › Память Flash

## Память таймера 128 байт

В памяти таймера 128 байт хранится текущее время теплосчётчика.

Параметр	Тип сигнала	Описание
rtc	uint8	Время теплосчётчика. В устройстве хранится в виде отдельных полей со значениями года, месяца и т.д.. При запросе модуль получает значения полей, преобразует в формат серверной метки времени и записывает в сигнал

## Память таймера 2К байт

В памяти таймера 2К байт хранятся:

- › информация о теплосчётчике

Параметр	Тип сигнала	Описание
systems	uint1	Число систем
type_g	uint1	Тип датчиков расхода: <ul style="list-style-type: none"><li>› 0 – частотные</li><li>› 1 – импульсные</li></ul>
type_q	uint1	Тип датчиков энергии: <ul style="list-style-type: none"><li>› 0 – Вт*ч</li><li>› 1 – Кал</li></ul>

		› 2 – Дж
type_t	uint1	Тип температур в статистике: 0 – среднеарифметические 1 – средневзвешенные
net_num	uint4	Номер прибора в сети
number	uint4	Заводской номер прибора
diam	uint2	Диаметр условного прохода по каналам, мм
g_max	float	Максимальное значение расхода по системам, т/ч
g_pcmt_max	uint1	Установленное значение G <sub>уmax</sub> , т/ч
g_pcmt_min	uint1	Установленное значение G <sub>уmin</sub> , т/ч
f_max	float	Максимальная частота, Гц
weight	float	Вес импульса, л/имп
next_hour	uint4	Адрес следующей часовой записи
next_day	uint4	Адрес следующей суточной записи
next_month	uint4	Адрес следующей записи на отчётную дату

› интеграторы по системам (SysInt)

Параметр	Тип сигнала	Описание
tek_dat	uint8	Время и дата записи
prev_dat	uint8	Время и дата предыдущей записи
IntV	double	Интегратор объёма по каналам, м <sup>3</sup> . В устройстве целая и дробная часть хранятся в отдельных полях, при запросе модуль получает значения обоих полей и записывает их в сигнал в виде общего значения
IntM	double	Интегратор массы по каналам, т. В устройстве целая и дробная часть хранятся в отдельных полях, при запросе модуль получает значения обоих полей и записывает их в сигнал в виде общего значения
IntQ	double	Интегратор энергии по каналам, МВт.

		В устройстве целая и дробная часть хранятся в отдельных полях, при запросе модуль получает значения обоих полей и записывает их в сигнал в виде общего значения
TRab	uint4	Время работы прибора при поданном питании, сек
TNar	uint4	Время работы систем без ошибок, сек
TMin	uint4	Расход меньше минимального, сек
TMax	uint4	Расход больше максимального, сек
Tdt	uint4	Разность температур меньше минимальной, сек
Ttn	uint4	Техническая неисправность, сек
tekerr	uint1	Ошибки по системам
teherr	uint2	Ошибки по системам
t	uint2	Температура по системам, °С/100
p	uint1	Давление по системам, МПа/100
rshv	float	Интеграторы объёмного расхода по каналам, м3/ч

➤ конфигурация систем (SysCon)

Параметр	Тип сигнала	Описание
SysType	uint1	Тип системы
Gprog	uint1	Расход по каналам: <ul style="list-style-type: none"> <li>➤ 0 – измеряемый</li> <li>➤ 1..100 – программируемый в % от Gmax</li> </ul>
Gchan	uint1	Используемые каналы расхода
Tprog	uint1	Температура по каналам: <ul style="list-style-type: none"> <li>➤ 0 – измеряемая</li> <li>➤ 1..150 – программируемая в °С</li> </ul>
Tchan	uint1	Используемые каналы температуры
Pprog	uint1	Давление по каналам: <ul style="list-style-type: none"> <li>➤ 0 – измеряемое</li> <li>➤ 1..16 - программируемое в МПа/10</li> </ul>

Pchan	uint1	Используемые каналы давления
-------	-------	------------------------------

## Оперативная память

В оперативной памяти хранятся текущие параметры по системам – четыре структуры SysPar.

Параметр	Тип сигнала	Описание
current_tmp	float	Текущие значения температуры по каналам, °С
current_prs	float	Текущие значения давления по каналам, МПа
current_ro	float	Текущие значения плотности теплоносителя
current_hent	float	Текущие значения энтальпии
current_rshv	float	Текущие значения объёмного расхода, м3/ч
current_rshm	float	Текущие значения массового расхода, т/ч
current_pwr	float	Текущие значения энергии, МВт
current_tekerr	uint1	Ошибки
current_teherr	uint2	Ошибки

## Память Flash

В памяти Flash хранится архив статистики:

- › Часовые записи (1536 записей – 2 месяца);
- › Суточные записи (368 записей – 1 год);
- › Месячные записи (144 записей – 12 лет) .

Каждая запись представляет собой структуру SysInt.

Параметр	Тип сигнала	Описание
tek_dat	uint8	Время и дата записи
prev_dat	uint8	Время и дата предыдущей записи

IntV	double	Интегратор объёма по каналам, м3. В устройстве целая и дробная часть хранятся в отдельных полях, при запросе модуль получает значения обоих полей и записывает их в сигнал в виде общего значения
IntM	double	Интегратор массы по каналам, т. В устройстве целая и дробная часть хранятся в отдельных полях, при запросе модуль получает значения обоих полей и записывает их в сигнал в виде общего значения
IntQ	double	Интегратор энергии по каналам, МВт. В устройстве целая и дробная часть хранятся в отдельных полях, при запросе модуль получает значения обоих полей и записывает их в сигнал в виде общего значения
TRab	uint4	Время работы прибора при поданном питании, сек
TNar	uint4	Время работы систем без ошибок, сек
TMin	uint4	Расход меньше минимального, сек
TMax	uint4	Расход больше максимального, сек
Tdt	uint4	Разность температур меньше минимальной, сек
Ttn	uint4	Техническая неисправность, сек
tekerr	uint1	Ошибки по системам
teherr	uint2	Ошибки по системам
t	uint2	Температура по системам, °С/100
p	uint1	Давление по системам, МПа/100
rshv	float	Интеграторы объёмного расхода по каналам, м3/ч



При запросе исторических данных, модуль запрашивает данные только за последний час/сутки/месяц из Flash памяти устройства.

## 1.1.2.5.13.4. Модуль: Запросы

Ниже приведены запросы, которые модуль посылает устройствам, и ответы устройств.

Запросы:

- › Установление связи
- › Чтение памяти таймера 128 байт
- › Чтение памяти таймера 2К байт
- › Чтение оперативной памяти
- › Чтение памяти Flash



Информация в данном разделе предназначена для анализа запросов/ответов в журнале модуля. Для конфигурирования эта информация не требуется.

### Установление связи

#### Запрос

Байт	Обозначение	Описание
0	SIG	Признак начала
1	ADDR	Сетевой адрес устройства
2	!ADDR	Инверсное значение сетевого адреса
3	CGRP	Группа команд
4	CMD	Идентификация устройства
5	LEN	Число байт посылаемых данных (0)
6	CS	Контрольная сумма (дополнение до нуля)

#### Ответ

Байт	Обозначение	Описание
0	SIG	Признак начала

1	ADDR	Сетевой адрес теплосчётчика, которому адресуется пакет
2	!ADDR	Инверсное значение сетевого адреса
3	CGRP	Группа команд
4	CMD	Идентификация устройства
5	LEN	Число байт посылаемых данных
6	DATA	T
7	DATA	E
8	DATA	M
9	DATA	-
A	DATA	1
B	DATA	0
C	DATA	4
D	CS	Контрольная сумма (дополнение до нуля)

## Чтение памяти таймера 128 байт

### Запрос

Байт	Обозначение	Описание
0	SIG	Признак начала
1	ADDR	Сетевой адрес устройства
2	!ADDR	Инверсное значение сетевого адреса
3	CGRP	Группа команд
4	CMD	Чтение памяти таймера 128
5	LEN	Число байт посылаемых данных (2)
6	TADDR	Начальный адрес в памяти таймера 128
7	TLEN	Длина считываемого блока данных (1-64 байт)
8	CS	Контрольная сумма (дополнение до нуля)



## Ответ

Байт	Обозначение	Описание
0	SIG	Признак начала
1	ADDR	Сетевой адрес устройства
2	!ADDR	Инверсное значение сетевого адреса
3	CGRP	Группа команд
4	CMD	Чтение памяти таймера 128
5	LEN	Число байт посылаемых данных (равно полю TLEN в запросе)
6-...	DATA	Данные
LEN+5	CS	Контрольная сумма (дополнение до нуля)

## Чтение памяти таймера 2К байт

### Запрос

Байт	Обозначение	Описание
0	SIG	Признак начала
1	ADDR	Сетевой адрес устройства
2	!ADDR	Инверсное значение сетевого адреса
3	CGRP	Группа команд
4	CMD	Чтение памяти таймера 2К
5	LEN	Число байт посылаемых данных (3)
6	TADRH	Начальный адрес в памяти таймера 2К (старший байт)
7	TADRL	Начальный адрес в памяти таймера 2К (младший байт)
8	TLEN	Длина считываемого блока данных (1-64 байт)
9	CS	Контрольная сумма (дополнение до нуля)

## Ответ

Байт	Обозначение	Описание
0	SIG	Признак начала
1	ADDR	Сетевой адрес устройства
2	!ADDR	Инверсное значение сетевого адреса
3	CGRP	Группа команд
4	CMD	Чтение памяти таймера 2К
5	LEN	Число байт посылаемых данных (равно полю TLEN в запросе)
6-...	DATA	Данные
LEN+5	CS	Контрольная сумма (дополнение до нуля)

## Чтение оперативной памяти

### Запрос

Байт	Обозначение	Описание
0	SIG	Признак начала
1	ADDR	Сетевой адрес устройства
2	!ADDR	Инверсное значение сетевого адреса
3	CGRP	Группа команд
4	CMD	Чтение оперативной памяти
5	LEN	Число байт посылаемых данных (3)
6	TADRH	Начальный адрес в оперативной памяти (старший байт)
7	TADRL	Начальный адрес в оперативной памяти (младший байт)
8	TLEN	Длина считываемого блока данных (1-64 байт)
9	CS	Контрольная сумма (дополнение до нуля)

## Ответ

Байт	Обозначение	Описание
0	SIG	Признак начала
1	ADDR	Сетевой адрес устройства
2	!ADDR	Инверсное значение сетевого адреса
3	CGRP	Группа команд
4	CMD	Чтение оперативной памяти
5	LEN	Число байт посылаемых данных (равно полю TLEN в запросе)
6-...	DATA	Данные
LEN+5	CS	Контрольная сумма (дополнение до нуля)

## Чтение памяти Flash

### Запрос

Байт	Обозначение	Описание
0	SIG	Признак начала
1	ADDR	Сетевой адрес устройства
2	!ADDR	Инверсное значение сетевого адреса
3	CGRP	Группа команд
4	CMD	Чтение оперативной памяти
5	LEN	Число байт посылаемых данных (5)
6	TLEN	Длина считываемого блока данных (1-64 байт)
7	FADR3	Начальный адрес в памяти Flash (старший байт)
8	FADR2	...
9	FADR1	...
A	FADR0	Начальный адрес в памяти Flash (младший байт)
B	CS	Контрольная сумма (дополнение до нуля)

## Ответ

Байт	Обозначение	Описание
0	SIG	Признак начала
1	ADDR	Сетевой адрес устройства
2	!ADDR	Инверсное значение сетевого адреса
3	CGRP	Группа команд
4	CMD	Чтение памяти Flash
5	LEN	Число байт посылаемых данных (равно полю TLEN в запросе)
6-...	DATA	Данные
LEN+5	CS	Контрольная сумма (дополнение до нуля)


# 1.1.2.5.13.5. Диагностика работы модуля

## Журнал работы модуля


Чтобы модуль записывал журнал, в общих параметрах модуля установите флаг Вести журнал работы модуля.

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:

 C:\Program Files\AstraRegul\Astra.Server\Logs;

» в Linux системах в директории:

 /opt/AstraRegul/Astra.Server/Logs.

Чтобы просмотреть журнал, откройте его в сервисном приложении Просмотрщик лога кадров.

№	Дата	Время	Описание	Станция	Т.	Адрес	Hex	Знаковое	Беззнаковое
26980	04.07.2023	17:14:39:375	Входящий кадр Функция 3 (Read Holding Registers)	1	3..	0	0	0	0
26981	04.07.2023	17:14:39:375	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				
26982	04.07.2023	17:14:39:386	Входящий кадр Функция 3 (Read Holding Registers)	1	3..				
26983	04.07.2023	17:14:39:386	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				
26984	04.07.2023	17:14:39:397	Входящий кадр Функция 3 (Read Holding Registers)	1	3..				
26985	04.07.2023	17:14:39:397	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				
26986	04.07.2023	17:14:39:428	Входящий кадр Функция 3 (Read Holding Registers)	1	3..				
26987	04.07.2023	17:14:39:428	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				
26988	04.07.2023	17:14:39:460	Входящий кадр Функция 3 (Read Holding Registers)	1	3..				
26989	04.07.2023	17:14:39:460	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				
26990	04.07.2023	17:14:39:491	Входящий кадр Функция 3 (Read Holding Registers)	1	3..				
26991	04.07.2023	17:14:39:491	Исходящий кадр Функция 3 (Read Holding Registers)	1	3..				

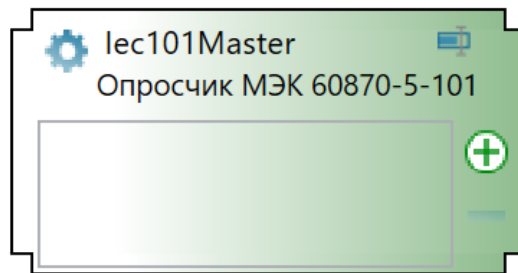
  

Исходящий кадр функция 3 (Read Holding Registers)
0000 93 76 00 00 00 05 01 03 02
0001 00

## 1.1.2.6. МЭК-60870-5

Модуль	Описание
<a href="#">Опросчик МЭК 60870-5-101</a>	Коммуникационный модуль для опроса станций по протоколу ГОСТ Р МЭК 60870-5-101
<a href="#">Станция МЭК 60870-5-101</a>	Коммуникационный модуль для передачи данных по протоколу ГОСТ Р МЭК 60870-5-101
<a href="#">Опросчик МЭК 60870-5-104</a>	Коммуникационный модуль для опроса станций по протоколу ГОСТ Р МЭК 60870-5-104
<a href="#">Станция МЭК 60870-5-104</a>	Коммуникационный модуль для передачи данных по протоколу ГОСТ Р МЭК 60870-5-104

## 1.1.2.6.1. Опросчик МЭК 60870-5-101



Опросчик МЭК 60870-5-101 – коммуникационный модуль, предназначенный для обмена данными между Astra.Server и подчиненными станциями по протоколу ГОСТ Р МЭК 870-5-101.



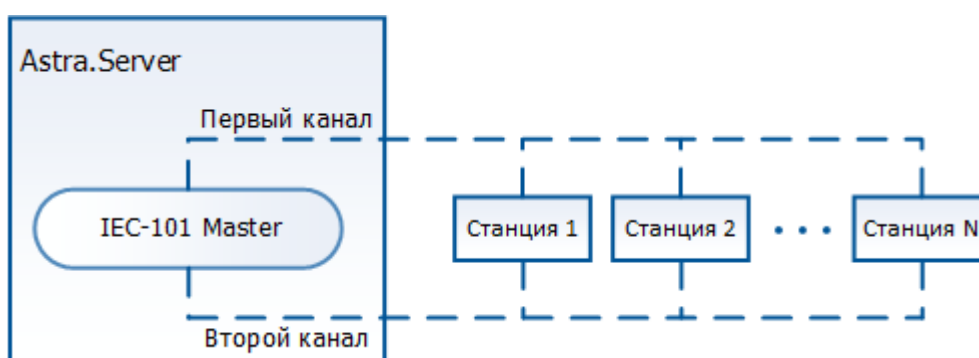
Обмен данными по протоколу ГОСТ Р МЭК 870-5-101 выполняется через последовательный интерфейс RS-232/422/485, поэтому для работы модуля Опросчик МЭК 60870-5-101 требуется наличие COM-портов на компьютере с установленным Astra.Server.

Функции модуля Опросчик МЭК 60870-5-101:

- › сбор данных подчиненных станций;
- › управление подчиненным станциям.

### Обмен данными

Опросчик МЭК 60870-5-101 может выполнять обмен данными с подчиненными станциями по одному или двум каналам связи в зависимости от количества COM-портов на компьютере и настроек модуля.



Обмен данными по каждому из каналов ведётся независимо от другого. Подчиненные станции по одному каналу опрашиваются по очереди.

Обмен данными с подчиненной станцией единовременно может выполняться только по одному каналу, который выбран основным для станции. Если для станции используется резервирование канала, то при потере связи по основному каналу обмен данными продолжается по второму каналу.

## **Сбор данных**

Опросчик МЭК 60870-5-101 выполняет сбор данных подчиненных станций следующими способами:

1. Периодический опрос - модуль циклически отправляет подчиненным станциям запросы и получает значения изменившихся сигналов. Период опроса для каждой станции задаётся индивидуально.
2. Спорадическая передача данных - модуль получает от подчиненной станции изменившиеся значения сигналов не дожидаясь очередного опроса.
3. По команде общего опроса - модуль получает значения всех сигналов подчиненных станций. Команда общего опроса подаётся циклически. Периодичность подачи команды общего опроса задаётся в настройках модуля.

## **Управление подчиненными станциями**

Опросчик МЭК 60870-5-101 управляет подчиненными станциями, посылая команды управления и команды уставок. Команды имеют более высокий приоритет, чем опрос станций, поэтому при подаче команды модуль прерывает опрос и передаёт команду подчиненной станции.

Для команд управления и уставок возможна настройка сигналов доставки. Значение сигнала доставки определяет состояние отправленной команды.

Отправка команд и формирование значений сигналов доставки выполняется по следующему алгоритму:



1. Сформированная команда управления или установки проверяется на соответствие требованиям:

- › резервная пара серверов активна и Astra.Server находится в режиме РАБОТА;
- › опросчик МЭК 60870-5-101 находится в режиме Опросчик;
- › значение, передаваемое командой, имеет хорошее качество.

Если все требования соблюдены, то команда помещается в очередь исходящих данных станции, а соответствующему сигналу доставки устанавливается значение 1.

Если соблюдены не все требования, команда не помещается в очередь исходящих данных станции, а соответствующему сигналу доставки устанавливается значение -4.

2. Перед отправкой очередь исходящих данных проверяется на наличие устаревших команд. Устаревшие команды удаляются из очереди, а сигналу доставки устанавливается значение -3.

3. Отправка команды выполняется, если:

- › соединение с подчиненной станции установлено по одному из каналов модуля;
- › канал является активным;

Если команда успешно отправлена, то она удаляется из очереди исходящих данных, а сигналу доставки устанавливается значение 2.

4. После отправки команды ожидается подтверждение доставки. Если доставка сигнала подтверждена, сигналу доставки устанавливается значение 3. Если доставка не подтверждена, то сигналу доставки устанавливается значение -2.

Возможные значениями сигнала доставки:

<b>Значение</b>	<b>Состояние команды</b>
1	Команда успешно помещена в очередь на отправку
2	Команда в ожидании подтверждения доставки

3	Доставка команды подтверждена
-1	Команду не удалось отправить
-2	Подтверждение доставки команды не получено
-3	Команда устарела и была изъята из очереди исходящих данных
-4	Команда не помещена в очередь исходящих данных, т.к. предварительные проверки не были пройдены

## Синхронизация времени

Чтобы данные, получаемые опросчиком МЭК 60870-5-101, имели одинаковую метку времени на стороне подчиненной станции и на стороне модуля, необходима синхронизация времени. Для этого опросчик МЭК 60870-5-101 подаёт команду синхронизации времени, периодичность которой задаётся в параметрах модуля.



Для синхронизации времени на стороне подчиненной станции должна поддерживаться корректировка времени по команде опросчика.

## Режимы работы модуля

Опросчик МЭК 60870-5-101 поддерживает два режима работы - Опросчик и Подслушка.

В режиме **Опросчик** модуль:

- формирует и отправляет запросы подчиненной станции;
- обрабатывает ответы, полученные от подчиненной станции;
- отправляет подчиненной станции команды управления.

В режиме **Подслушка** модуль обрабатывает ответы подчиненной станции на запросы стороннего опросчика. Отправка команд управления подчиненной станции невозможна.

## Работа модуля в резерве

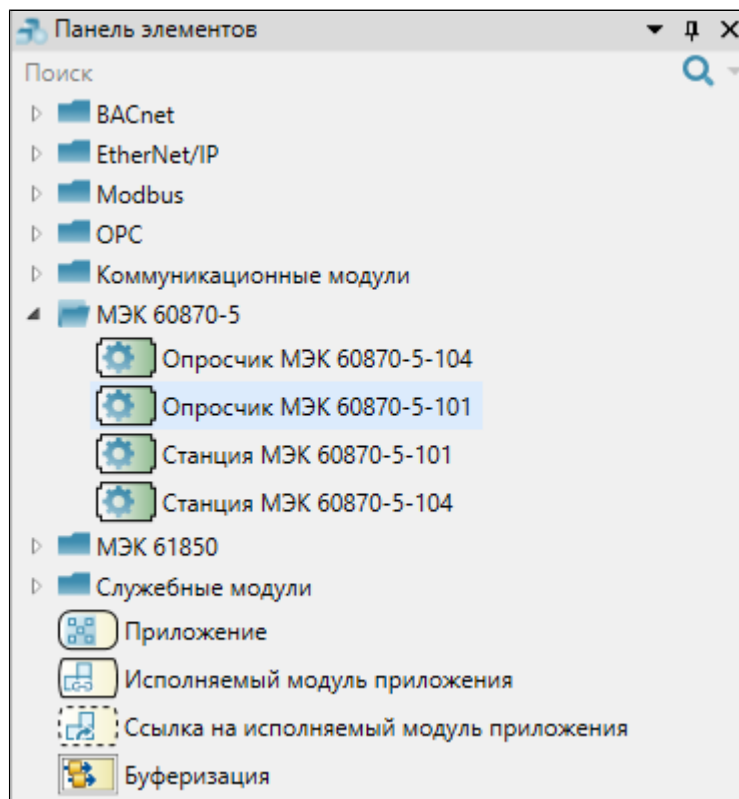
В режиме РЕЗЕРВ модуль опросчик МЭК 60870-5-101 не ведёт опрос подчиненных станций и не отправляет команды управления.

## 1.1.2.6.1.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Одновременно в составе конфигурации Astra.Server может функционировать несколько экземпляров опросчика МЭК 60870-5-101.



Выделите модуль, чтобы настроить его свойства.

Свойства	
lec101Master Опросчик МЭК 60870-5-101	
<b>Параметры модуля</b>	
Режим работы модуля	Опросчик
Таймаут потери связи, сек.	20
Интервал опроса, мин	30
Интервал синхронизации станций, мин	30
Опрос при смене канала	Нет
Настройки по умолчанию	
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	lec101Master



Для параметров модуля допустимо использование значений по умолчанию, однако рекомендуется устанавливать параметру "Активность" значение "Да", чтобы модуль запускался при запуске/перезапуске сервера.

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Режим работы модуля	Выбор режима работы модуля: <ul style="list-style-type: none"> <li>➤ Опросчик - модуль выполняет опрос подчиненных станций, обрабатывает полученные ответы. Возможна подача команд управления;</li> <li>➤ Подслушка - модуль не ведёт опрос подчиненных станций, а только обрабатывает ответы, полученные на запросы стороннего опросчика. Подача команд управления невозможна</li> </ul>
Таймаут связи, сек	Период после потери связи с подчиненной станцией, через который связь считается потерянной. Значение по умолчанию 0 секунд
Интервал опроса, мин	Период, через которой подчиненным станциям подаётся команда общего опроса. Значение по умолчанию 30 минут. При значении 0 команда общего опроса не подаётся
Интервал синхронизации станций, мин	Период, через которой подчиненным станциям подаётся команда синхронизации времени. Значение по умолчанию 30 минут. При значении 0 команда синхронизации времени не подаётся
Опрос при смене канала	Подача команды общего опроса подчиненным станциям при смене канала: <ul style="list-style-type: none"> <li>➤ Да - подавать команду общего опроса</li> <li>➤ Нет - не подавать команду общего опроса</li> </ul>
Настройки по умолчанию	Выбор файла с готовой конфигурацией модуля
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>➤ Да – модуль запущен;</li> <li>➤ Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.6.1.2. Настройка COM порта

Для подключения к подчиненной станции настройте параметры COM-порта в группе Параметры Первого канала. Если требуется вести опрос станций по двум каналам или использовать резервирование основного канала, то в группе Параметры Второго канала установите параметру Использовать второй канал значение Да и настройте параметры второго COM-порта.



Значения параметров COM-портов модуля Опросчик МЭК 60870-5-101 должны совпадать со значениями параметров COM-портов подчиненной станции.

Параметр	Описание
Имя COM-порта	Системное имя COM-порта, через который работает модуль. В зависимости от используемой операционной системы имя COM-порта имеет вид: <ul style="list-style-type: none"><li>› В ОС Windows - \\.\COM&lt;n&gt;;</li><li>› В ОС Linux - /dev/ttyS&lt;n&gt;, /dev/ttyUSB&lt;n&gt; или другие;</li></ul> где <n> - номер COM-порта
Скорость (бит/с)	Скорость обмена данными между модулем и подчиненной станцией. Выбирается из выпадающего списка
Чётность	Выбор способа контроля чётности для обнаружения ошибок при обмене данными: <ul style="list-style-type: none"><li>› Нет – контроль чётности не ведется, проверки ошибок нет;</li><li>› Чет – проверка на чётность;</li><li>› Нечет – проверка на нечётность</li></ul>
Стоповые биты	Количество стоповых битов, необходимых для правильного распознавания конца байта. Выбирается из выпадающего списка – 1 или 2 стоповых бита
Управление потоком	Выбор режима управления потоком передаваемых данных: <ul style="list-style-type: none"><li>› Нет – управление потоком выключено;</li></ul>

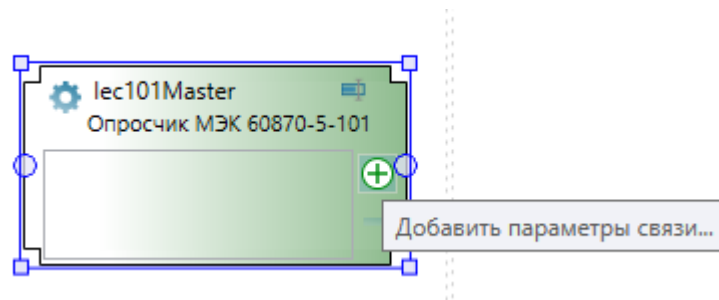


- Аппаратное – аппаратная синхронизация, передатчик останавливает передачу данных, если приемник не готов к приему. Обеспечивает быструю реакцию на состояние приемника;
- Хон/Хофф – программная синхронизация, приемник посылает сигнал о неготовности передатчику, после чего передатчик останавливает посылку данных. Необходим двунаправленный канал связи

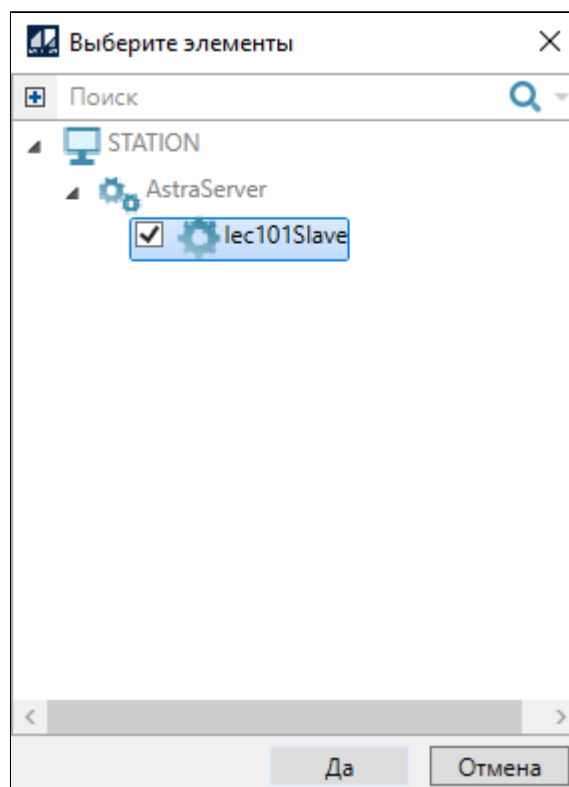
## 1.1.2.6.1.3. Настройка станции

Чтобы добавить одну или несколько подчиненных станций:

1. В модуле Опросчик МЭК 60870-5-101 нажмите на кнопку "Добавить параметры связи...".



2. В открывшемся окне выберите необходимые станции для опроса.



3. Заполните параметры станции, описание которых представлено ниже.

Свойства	
Параметры связи со станцией МЭК 60870-5	
<b>Параметры работы по основному каналу</b>	
Период опроса, сек.	0
Таймаут на получение ответа, мс	500
Количество повторов запроса данных	2
Количество повторов ТУ	2
Максимальный размер кадра	64
Максимальное количество запросов подряд	3
<b>Параметры работы по резервному каналу</b>	
Период опроса, сек.	0
Таймаут на получение ответа, мс	500
Количество повторов запроса данных	2
Количество повторов ТУ	2
Максимальный размер кадра	64
Максимальное количество запросов подряд	3
<b>Параметры станции</b>	
Выполнять процедуру тестирования	Да
<b>Общие</b>	
Подчиненная станция	Iec101Slave

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры станции

Параметр	Описание
----------	----------

Выполнять процедуру тестирования	Проверка связи с устройством с помощью команды C_ST_NA_1
----------------------------------	--

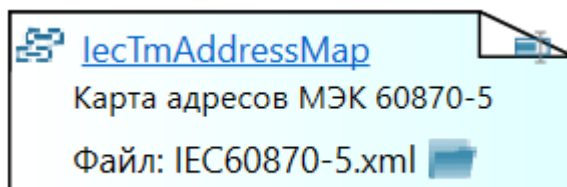
## Параметры работы по основному/резервному каналу

Параметр	Описание
Период опроса, сек	Период, через который проводится опрос подчиненной станции. Если с момента последнего опроса период не истёк, то в текущем цикле опроса станция не опрашивается
Таймаут на получение ответа, мс	Период ожидания ответа на запрос. Если по истечении периода ответ не получен, то связь со станцией считается потерянной
Количество повторов запроса данных	Количество попыток повторного запроса данных, если данные подчиненной станции не были получены после первого запроса
Количество повторов ТУ	Количество попыток повторной подачи команды управления, если команда не была подана с первой попытки
Максимальный размер кадра	Максимальный размер кадра данных в байтах, передаваемый подчиненной станции
Максимальное количество запросов подряд	Максимальное количество запросов на получение данных, передаваемых станции за один цикл опроса

## 1.1.2.6.1.4. Карта адресов

Для настройки сигналов модуля Опросчик МЭК 60870-5-101 используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов МЭК 60870-5 в исполняемое приложение;



2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	Адрес	Протокольный тип	Интервал нечувствительности	Положение десятичной запятой	Номер бита	Категория данных
string	string	непосредстве	0	U-CTRL				
string1	string	непосредстве	0	U-MON				
bool	bool	непосредстве	0 45:	C_SC_NA_1				
uint2	uint2	непосредстве	0 33:	M_BO_TB_1				

## Параметры карты адресов

Параметр	Значение
Привязка	Тип привязки параметра: <ul style="list-style-type: none"><li>› непосредственно;</li><li>› только чтение;</li><li>› не привязан.</li></ul>
Адрес	Адрес элемента информации. Размер адреса элемента информации 3 байта. Значения в диапазоне от 0 до 16777215
Протокольный тип	Идентификатор типа. Определяет структуру, тип и формат элемента информации. Модуль поддерживает <a href="#">стандартные</a> и <a href="#">частные</a> типы
Интервал нечувствительности	Величина, при превышении которой значение сигнала с подчиненной станции отправляется ведущей станции

Номер бита	Номер бита в байте. Значения в диапазоне от 0 до 7. Устанавливается для протокольных типов TS и TC
Положение десятичной запятой	Положение десятичной запятой. Соответствует числу, равному степени числа 10, и применяется в качестве множителя к полученному значению сигнала. Значения в диапазоне от -41 до +33.

## 1.1.2.6.1.5. Диагностика работы модуля

### Журнал работы модуля

Модуль Опросчик МЭК 60870-5-101 ведёт журнал работы, в который записывается информация о работе модуля и обмене данными с подчиненной станцией.



Чтобы модуль вёл журнал работы, в общих параметрах модуля установите параметру Вести журнал работы модуля значение Да или установите сервисному сигналу модуля `FrameLogEnable.Set` значение `true`.

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

› в ОС Windows в папке:



`C:\Program Files\AstraRegul\Astra.Server\Logs;`

› в Linux системах в директории:



`/opt/AstraRegul/Astra.Server/Logs.`

Для просмотра журнала работы модуля используется сервисное приложение Просмотрщик лога кадров.

№	Дата	Время	Описание	Станция	У.	Л.	П.	№	Адрес	Значение	Метка времени	Каче
24105	01.07.2021	20:53:06:706	Попытка отправки управляющих данных для станции '1'					1	1	20		ОПС
24106	01.07.2021	20:53:07:396	Исходящий кадр <45> C_SC_NA_1 (отправка/подтверждение)	1			6..					
24107	01.07.2021	20:53:07:396	Входящий кадр Положительная квитанция	1			-					
24108	01.07.2021	20:53:07:396	Совершена отправка управляющих данных для станции '1' по о...									
24109	01.07.2021	20:53:07:417	Исходящий кадр Запрос о состоянии канала связи	2			-					
24110	01.07.2021	20:53:08:120	Исходящий кадр Запрос о состоянии канала связи	2			-					
24111	01.07.2021	20:53:08:820	Исходящий кадр Запрос о состоянии канала связи	2			-					
24112	01.07.2021	20:53:09:531	Попытка получить данные от станции '1'									
24113	01.07.2021	20:53:09:626	Исходящий кадр Запрос данных класса 1	1			-					
24114	01.07.2021	20:53:09:626	Входящий кадр <5> M_ST_NA_1	1			3..					
24115	01.07.2021	20:53:09:626	Исходящий кадр Запрос данных класса 1	1			-					
24116	01.07.2021	20:53:09:626	Входящий кадр Отрицательная квитанция. Запрошенные данн...	1			-					
24117	01.07.2021	20:53:09:626	Исходящий кадр Запрос данных класса 2	1			-					
24118	01.07.2021	20:53:09:626	Входящий кадр Отрицательная квитанция. Запрошенные данн...	1			-					
24119	01.07.2021	20:53:09:626	Данные от станции '1' получены по основному каналу									
24120	01.07.2021	20:53:09:627	Исходящий кадр Запрос о состоянии канала связи	2			-					

Входящий кадр <5> M\_ST\_NA\_1

	0	1	2	3	4	5	6	7	8	9
0000	68	0D	0D	68	08	01	05	01	03	00
0001	01	00	01	00	00	14	00	28	16	

Количество записей: 18067

Каждая запись журнала имеет порядковый номер, дату, время, описание. Записи исходящих и входящих кадров модуля дополнительно содержат номер подчиненной станции и причину передачи.

Побайтовое представление и данные кадров модуля отображаются в соответствующих полях окна сервисного приложения Просмотрщик лога кадров.

№	Адрес	Значение	Метка врем...	Качество	Ном...	Тип
1	1	20		ОПС_GOOD (0x...		M_ST_NA_1

Входящий кадр <5> M\_ST\_NA\_1

	0	1	2	3	4	5	6	7	8	9
0000	68	0D	0D	68	08	01	05	01	03	00
0001	01	00	01	00	00	14	00	28	16	

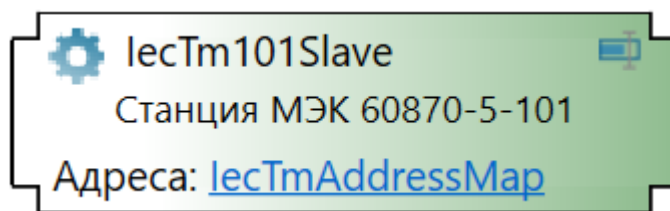
Индикация байт в поле побайтового представления кадра данных переменной длины приведена в таблице:

Байт	Цвет	Описание
0	Голубой	Стартовый байт – 0x68
1	Голубой	Длина блока - число байт пользовательских данных, включая поле управления и адресное поле
2	Голубой	Длина блока - число байт пользовательских данных, включая поле управления и адресное поле



3	Голубой	Стартовый байт – 0x68
4	Светло-голубой	Байт управления
5	Светло-голубой	Адрес - 0x01
от 6 до N	Зеленый	Блок данных прикладного уровня (ASDU)
N+1	Зеленый	Байт качества
N+2	Коричневый	Контрольная сумма
N+3	Розовый	Конечный байт - 0x16

## 1.1.2.6.2. Станция МЭК 60870-5-101



Модуль Станция МЭК 60870-5-101 – коммуникационный модуль, предназначенный для обмена данными между Astra.Server и опросчиком по протоколу ГОСТ Р МЭК 870-5-101. Модуль реализует одну или несколько подчиненных станций.



Обмен данными по протоколу ГОСТ Р МЭК 870-5-101 выполняется через последовательный интерфейс RS-232/422/485, поэтому для работы модуля Станция МЭК 60870-5-101 требуется наличие COM-портов на компьютере с установленным Astra.Server.

Функции модуля Станция МЭК 60870-5-101:

- › предоставление данных опросчику;
- › приём команд опросчика.

### Обмен данными

Модуль Станция МЭК 60870-5-101 может выполнять обмен данными с опросчиком по одному или двум каналам связи в зависимости от количества COM-портов на компьютере и настроек модуля.



Обмен данными по каждому из каналов ведётся независимо от другого. Единоновременно обмен данными может выполняться только по одному каналу, который выбран основным для станции. Если для станции используется резервирование канала, то при потере связи по основному каналу обмен данными продолжается по резервному каналу.

## Предоставление данных

Модуль Станция МЭК 60870-5-101 предоставляет опросчику изменившиеся значения сигналов самостоятельно и по запросу опросчика.

## Приём команд опросчика

Модуль Станция МЭК 60870-5-101 принимает от опросчика команды управления и команды уставок. Команды имеют более высокий приоритет, чем предоставление данных, поэтому при получении команды модуль прерывает передачу данных и принимает команду опросчика.

## Синхронизация времени

Чтобы данные, предоставляемые модулем Станция МЭК 60870-5-101, имели одинаковую метку времени с опросчиком, необходима синхронизация времени. Для этого опросчик подаёт команду синхронизации времени

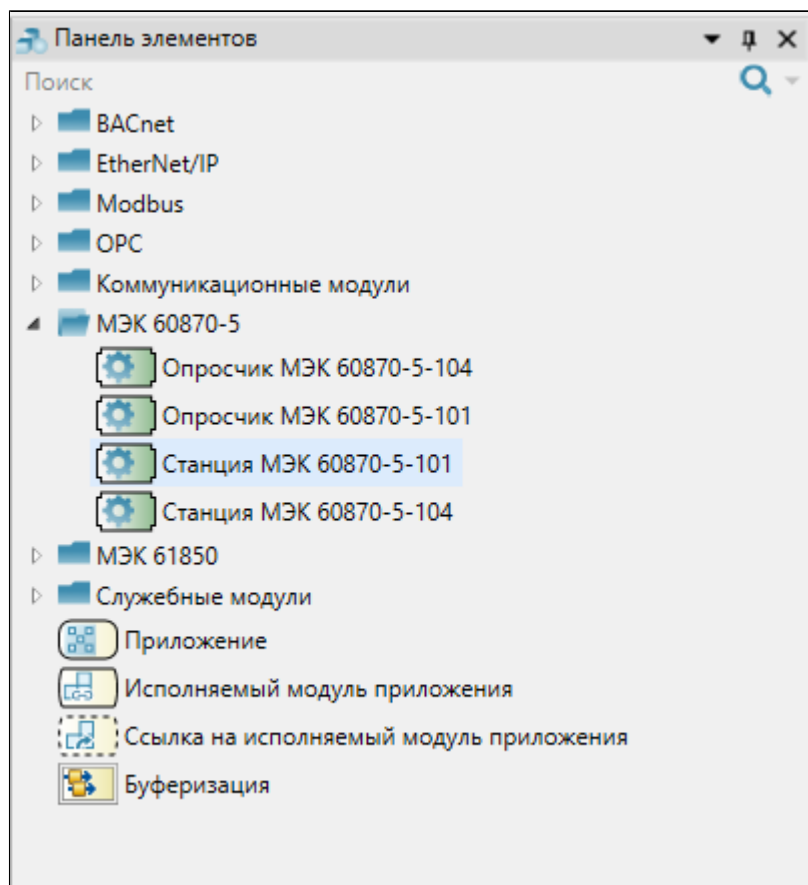
с некоторой периодичностью, а модуль Станция МЭК 60870-5-101 синхронизирует время со временем опросчика.

## **Работа модуля в резерве**

Работа модуля в РЕЗЕРВЕ настраивается в параметрах модуля. В режиме РЕЗЕРВ модуль Станция МЭК 60870-5-101 в зависимости от настройки может работать в полнофункциональном режиме аналогично состоянию в РАБОТЕ, либо не устанавливать соединение с опросчиком, не предоставлять данные и не принимать команды управления.

# Настройка модуля

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

lec101Slave Станция МЭК 60870-5-101	
<b>Параметры модуля</b>	
Интервал сохранения соединения (первый канал)	10000
Интервал сохранения соединения (второй канал)	10000
Настройки по умолчанию	
Работать в РЕЗЕРВЕ	Нет
Корректировать время по команде опросчика	Нет
Сбрасывать ТУ	Нет
Задержка сброса ТУ, мс	False
Менять качество входных сигналов	Да
Активность	Да
Отображаемое имя	
<b>Общие</b>	
Карта адресов	
Имя	lec101Slave
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения



Для параметров модуля допустимо использование значений по умолчанию, однако рекомендуется устанавливать параметру "Активность" значение "Да", чтобы модуль запускался при запуске/перезапуске сервера.

## Общие

Параметр	Описание
Имя	Имя модуля
Карта адресов	Карта адресов МЭК 60870-5, которая будет использована для данного адаптера

## Параметры модуля

Параметр	Описание
Интервал сохранения соединения (первый канал)	Интервал сохранения соединения на первом канале
Интервал сохранения соединения (второй канал)	Интервал сохранения соединения на втором канале
Настройки по умолчанию	Выбор файла с готовой конфигурацией модуля
Работать в РЕЗЕРВЕ	Если данный флаг сброшен, то соединение с мастером в резервном режиме не устанавливается Значения: <ul style="list-style-type: none"> <li>› Да</li> <li>› Нет</li> </ul>
Корректировать время по команде опросчика	Значения: <ul style="list-style-type: none"> <li>› Да</li> <li>› Нет</li> </ul>
Сбрасывать ТУ	Значения: <ul style="list-style-type: none"> <li>› Да</li> <li>› Нет</li> </ul>
Задержка сброса ТУ, мс	Значение по умолчанию: False
Менять качество входных сигналов	Менять качество входных сигналов при изменении статуса соединения Значения: <ul style="list-style-type: none"> <li>› Да</li> <li>› Нет</li> </ul>
Активность	Активность модуля при запуске/перезапуске Astra.Server:

	<ul style="list-style-type: none"><li>› Да – модуль запущен;</li><li>› Нет – модуль остановлен.</li></ul> Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

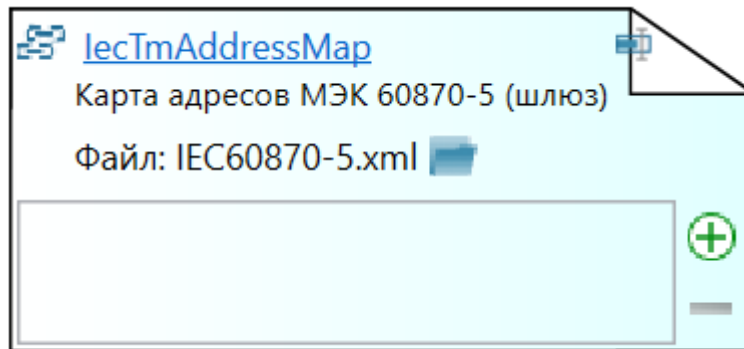
[Параметры журналирования](#) настраиваются одинаково для всех модулей.



## 1.1.2.6.2.2. Карта адресов

Для настройки сигналов модуля Станция МЭК 60870-5-101 используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов МЭК 60870-5 (шлюз) в исполняемое приложение;



2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	Станция	Адрес	Протокольный тип	Интервал нечувствительности	Положение десятичной запятой	Номер бита
string	string	непосредств	0	0	U-CTRL			
string1	string	непосредств	0	0	U-MON			
bool	bool	непосредств	0	0	45: C_SC_NA_1			
uint2	uint2	непосредств	0	0	33: M_BO_TB_1			

## Параметры карты адресов

Параметр	Значение
Привязка	Тип привязки параметра: <ul style="list-style-type: none"><li>&gt; непосредственно;</li><li>&gt; только чтение;</li><li>&gt; не привязан;</li><li>&gt; унаследован.</li></ul>
Станция	Номер станции, к которой относится сигнал
Адрес	Адрес элемента информации. Размер адреса элемента информации 3 байта. Значения в диапазоне от 0 до 16777215

Протокольный тип	Идентификатор типа. Определяет структуру, тип и формат элемента информации. Модуль поддерживает <a href="#">стандартные</a> и <a href="#">частные</a> типы
Интервал нечувствительности	Величина, при превышении которой значение сигнала с подчиненной станции отправляется ведущей станции
Положение десятичной запятой	Соответствует числу, равному степени числа "10", и применяется в качестве множителя к полученному значению сигнала. Значения в диапазоне от "-41" до "+33"
Номер бита	Номер бита в байте. Значения в диапазоне от "0" до "7". Устанавливается для протокольных типов TS, TC, TCR.

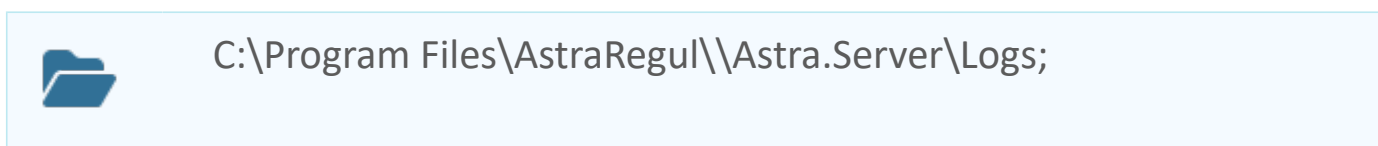
# Диагностика работы модуля

## Журнал работы

Каждый модуль ведет журнал работы. В него сохраняется вся информация о работе модуля и об обмене данными с подчиненными станциями. Журнал работы модуля предназначен для контроля работы модуля в режиме реального времени, а также просмотра прошедших событий модуля.

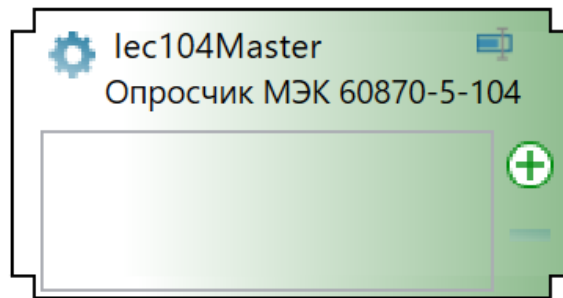
Для просмотра журнала работы модуля используется сервисное приложение Просмотрщик лога кадров.

Чтобы просмотреть журнал работы модуля, выберите файл IEC Slave.aplog в директории:



№	Дата	Время	Описание	Станция	Удаленный адрес	Лок.	№	Адрес	Значение	Метка вре...	Качество	Но...	Тип					
166526	09.11.2021	05:48:43:799	Исходящий кадр <15> M_IT_NA_1 (S:10172;R:1709)	1	172.16.150.18:53040	172.	1	294			OPC_BAD (0xFF)		STR_Status					
166527	09.11.2021	05:48:43:799	Исходящий кадр <162> Упакованные данные (S:10173;R:1709)	1	172.16.150.18:53040	172.	Исходящий кадр <190> STR_Status (S:10176;R:1709)											
166528	09.11.2021	05:48:43:799	Исходящий кадр <158> STR (S:10174;R:1709)	1	172.16.150.18:53040	172.												
166529	09.11.2021	05:48:43:799	Исходящий кадр <162> Упакованные данные (S:10175;R:1709)	1	172.16.150.18:53040	172.												
166530	09.11.2021	05:48:43:799	Исходящий кадр <190> STR_Status (S:10176;R:1709)	1	172.16.150.18:53040	172.												
166531	09.11.2021	05:48:43:845	Входящий кадр S-кадр (R:10173)	1	172.16.150.18:53040	172.												
166532	09.11.2021	05:48:43:845	Исходящий кадр <162> Упакованные данные (S:10177;R:1709)	1	172.16.150.18:53040	172.												
166533	09.11.2021	05:48:43:845	Исходящий кадр <222> STR Time (S:10178;R:1709)	1	172.16.150.18:53040	172.												
							0000	68	0F	80	4F	5A	0D	BE	01	14	00	
							0001	01	00	2E	01	00	00	80				

### 1.1.2.6.3. Опросчик МЭК 60870-5-104



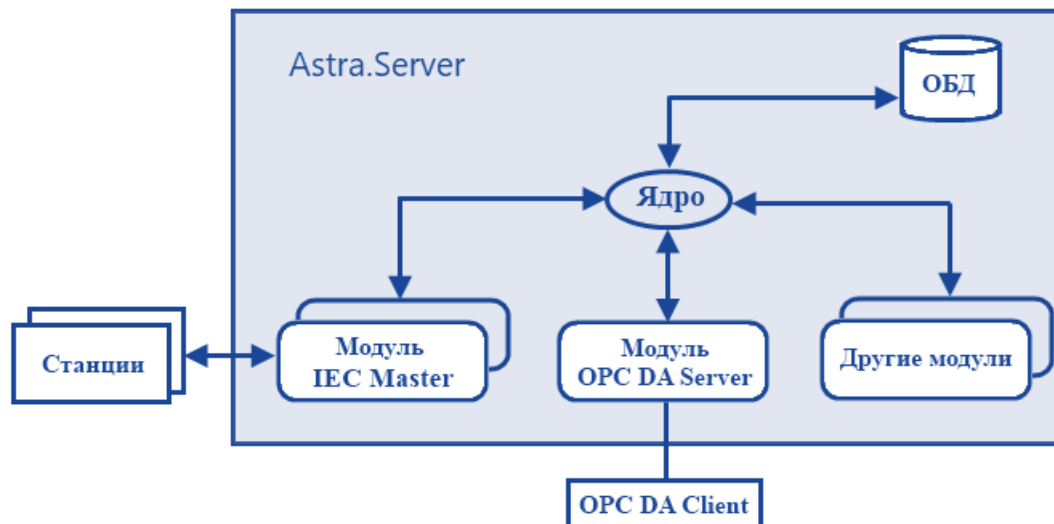
Опросчик МЭК 60870-5-104 работает в составе Astra.Server, который является частью системы для сбора, анализа и регулирования параметров технологического процесса.

Опросчик МЭК 60870-5-104 реализует опросчик в соответствии с ГОСТ Р МЭК 870-5-104-2004 и расширением стандартного диапазона типов IEC\_CT1.

Основными функциями опросчика МЭК 60870-5-104 являются:

- › опрос подчиненных станций;
- › управление подчиненными станциями (отправка команд ТУ и ТР).

Опросчик МЭК 60870-5-104 является событийным модулем. Работая в составе Astra.Server опросчик МЭК 60870-5-104, получает данные со станции при изменении значения параметра, а также опрашивает подчиненные станции, циклически отправляя подчиненным станциям запросы. Полученные данные через интерфейсы ядра сохраняются в оперативную базу данных (ОБД). Ядро Astra.Server при получении изменившихся сигналов уведомляет об изменениях значений сигналов другие модули, которые обслуживают эти сигналы, в том числе и модуль OPC DA Server. Далее модуль OPC DA Server передает данные клиентам, подписанным на изменившиеся сигналы.



Модуль работает в двух режимах: РАБОТА и РЕЗЕРВ.

Опросчик МЭК 60870-5-104 производит параллельный опрос активного источника в составе станции по нескольким каналам связи. При параллельном опросе источника (когда количество каналов больше 1) применяются типы данных с меткой времени.

Для упорядочивания событий по времени и для получения времени наступления событий на подчиненных станциях имеется возможность синхронизации времени.

Для быстрого выявления неполадок связи поддерживается тестирование соединения согласно ГОСТ Р МЭК 870-5-104-2004.

Для расширения функциональных возможностей кроме поддержки стандартных типов данных МЭК, поддерживаются пользовательские типы данных.

Имеется возможность подключения 65534 подчиненных станций. Адрес 65535 является широковещательным, применяется при синхронизации времени и общем опросе.

История работы модуля сохраняется в журнале работы модуля. Изменение флага записи в журнал модуля может производиться динамически, т.е. без перезапуска Astra.Server.

## Режимы работы модуля

Модуль Опросчик МЭК 60870-5-104 может работать в двух режимах: РАБОТА или РЕЗЕРВ. Режим функционирования модуля в Astra.Server устанавливает ядро. В процессе функционирования режим может измениться.

Режим РЕЗЕРВ предназначен для повышения надежности системы управления. Повышение надежности происходит благодаря тому, что при выходе из строя Astra.Server, содержащего активный модуль, происходит быстрое переключение на другой Astra.Server, и при этом минимизируются потери данных со станции. При работе двух модулей одновременно переключения нет. В режиме РЕЗЕРВ модуль имеет 2 варианта работы, вариант работы в режиме РЕЗЕРВ настраивается в параметрах конфигурации:

- › проводить опрос, при этом отсутствует выдача команд ТУ и команд ТР. В этом состоянии минимальное время для перехода в режим РАБОТА;
- › закрыть соединение.

## Опрос подчиненных станций

При запуске модуля Опросчик МЭК 60870-5-104 происходит его инициализация. Процедура инициализации работы модуля требуется для установки модуля в правильное рабочее состояние до того, как начнется опрос и передача данных. Процесс инициализации включает в себя:

- › чтение конфигурационных данных модуля;
- › подписка у ядра на получение уведомлений об изменениях сигналов;
- › проверка корректности настройки адресов для сигналов уведомлений;
- › постановка сигналов уведомлений на обслуживание;
- › построение списка очереди опроса.

Так же во время инициализации модуля устанавливается соединение с подчиненными станциями и их источниками. При установленном соединении модуль отправляет станциям сообщение о своей готовности принимать данные.

Конфигурация модуля включает в себя список станций, номера которых соответствуют номеру КП. Каждая станция может содержать от 1 до 4

источников данных. Каждый источник данных может содержать от 1 до 4 каналов связи.

Опрос активного источника в составе станции ведется по всем каналам параллельно.

Общий опрос станции проводится периодически, период задается в настройках конфигурации.

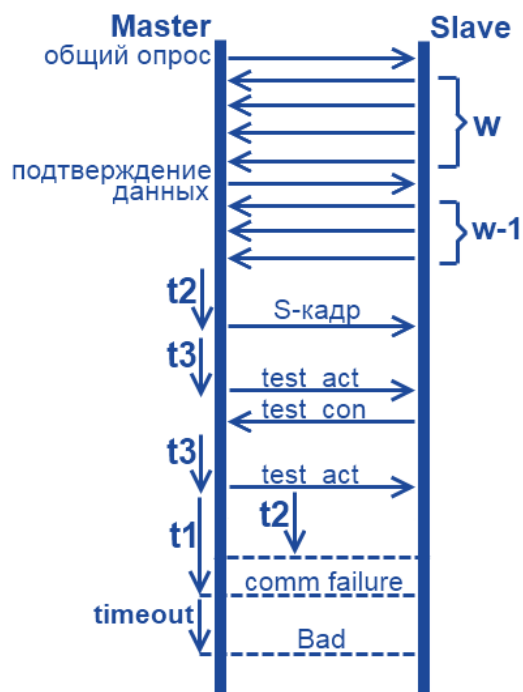
Модуль Опросчик МЭК 60870-5-104 является событийным и получает данные со станции при изменении значения параметра, а также в зависимости от настроек подчиненной станции получает данные при изменении качества сигнала и метки времени.

Для получения данных с подчиненной станции, необходимо создать сигналы подписки на эти данные в конфигураторе Astra.Server.

Если адрес сигнала с подчиненной станции не обслуживается модулем, то пришедшие данные игнорируются. Если у подчиненной станции нет подписки на обслуживание команды отправленной модулем, то команда игнорируется.

Если полученные данные с подчиненной станции не успевают обрабатываться и очередь данных достигает максимального размера (параметр конфигурации), то опрос станции не останавливается, но происходит запись в журнал работы модуля об этом событии. Данные, которые не входят в очередь – игнорируются.

В ответ на команду общего опроса модулю приходят кадры данных.



При достижении количества кадров для подтверждения  $w$  модуль отправляет уведомление подтверждения данных, в котором находится количество полученных кадров. Если по каналу не проходит, каких-либо сообщений за интервал тестирования  $t3$ , то отправляется команда тестирования связи. После отправки команды модуль ожидает сообщение подтверждения тестирования промежуток времени  $t2$ .

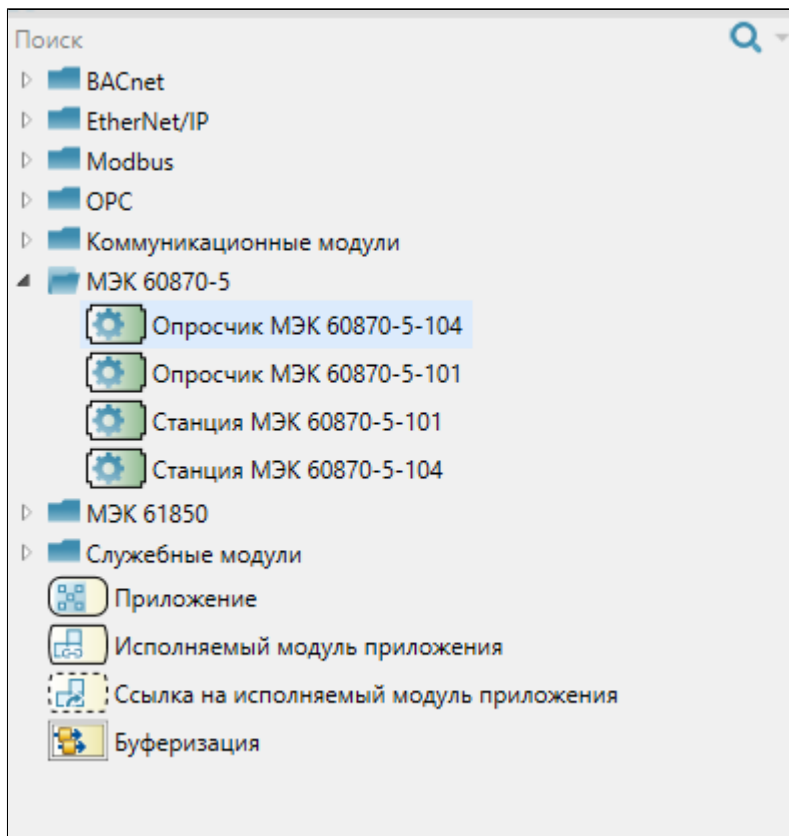
Параметр	Описание
$w$	Количество входящих кадров для подтверждения.
$t1$	Интервал отправки данных — таймаут при посылке данных. Если нет подтверждения на отправленные данные в течение этого времени, то сигналу выставляется качество COMM_FAILURE, обозначающее состояние потери связи и начинается отчет Таймаута потери связи (параметр конфигурации). В течение Таймаута потери связи модуль пытается восстановить соединение с подчиненной станцией. Если связь не восстановлена, то всем сигналам модуля выставляется плохое качество. При этом должно выполняться $t2 < t1$ , так как сначала должно прийти подтверждение получения данных.



t2	Таймаут подтверждения данных — промежуток времени, после которого модуль должен выслать сообщение подтверждения пришедших данных. При получении данных от станции модуль увеличивает счетчик пришедших пакетов данных, когда истекает время t2, модуль высылает число пакетов, полученных за это время, для проверки верности передаваемых данных.
t3	Интервал тестирования — период отправки команд тестирования. Если в течение этого промежутка времени по каналу связи не проходит, каких-либо сообщений, то отправляется команда тестирования связи. После отправки команды модуль ожидает сообщение подтверждения тестирования промежуток времени t1.

## 1.1.2.6.3.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

lec104Master Опросчик МЭК 60870-5-104	
<b>Параметры модуля</b>	
Режим работы модуля	
Работать в РЕЗЕРВЕ	Да
Настройки по умолчанию	
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	lec104Master

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Режим работы модуля	Выбор режима работы модуля: <ul style="list-style-type: none"> <li>➤ Полная функциональность — режим без ограничений (опрос и выдача управляющих воздействий);</li> <li>➤ Только опрос — модуль не способен выдавать управляющие воздействия, а производит исключительно опрос подчиненных станций.</li> </ul>
Работать в РЕЗЕРВЕ	Определяет поведение модуля, когда сервер находится в состоянии РЕЗЕРВ: <ul style="list-style-type: none"> <li>➤ Да — модуль получает данные, но не выдает управляющих воздействий, независимо от состояния активности резервной пары;</li> </ul>

	<ul style="list-style-type: none"> <li>› Нет — модуль не работает, когда сервер находится в состоянии РЕЗЕРВ.</li> </ul>
Настройки по умолчанию	Выбор файла с готовой конфигурацией модуля
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>› Да – модуль запущен;</li> <li>› Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

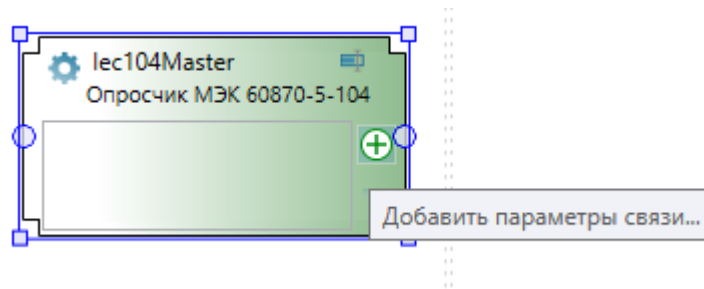
## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

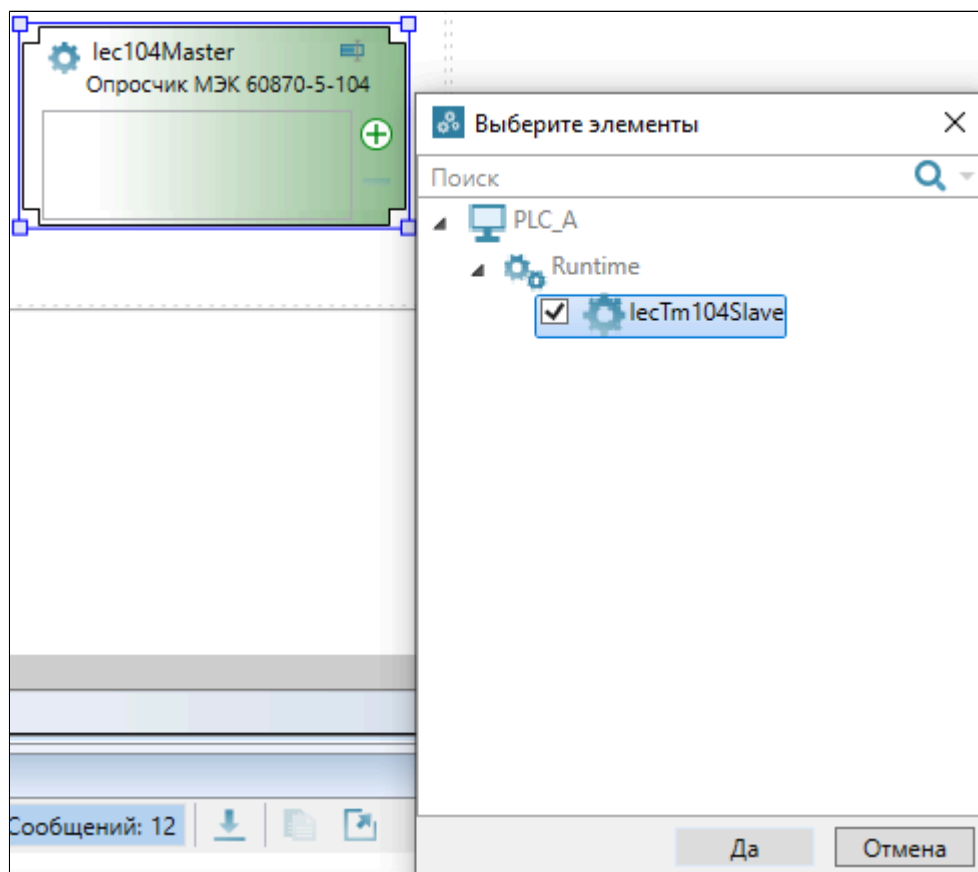
## 1.1.2.6.3.2. Добавление станций

Чтобы добавить одну или несколько подчиненных станций необходимо выполнить следующие действия:

1. В модуле Опросчик МЭК 60870-5-104 нажмите на кнопку "Добавить параметры связи..."



2. В открывшемся окне выберите необходимые станции для опроса



3. Заполните параметры станции, описание которых представлено ниже.

Свойства	
Параметры связи со станцией МЭК 60870-5-104	
<b>Параметры станции</b>	
Интервал опроса, мин	30
Интервал синхронизации станций, мин	30
Задержка между попытками подключения, сек.	0
Таймаут потери связи, сек.	0
Отправлять ТУ (C_SC_NA_1) со значениями false	Да
Максимальный размер очереди приема данных	10000
Максимальный размер очереди данных на отправку	1000
Отправлять все исходящие сигналы при активации источника	Нет
Сдвиг метки времени	0
Алгоритм валидации входящих значений	Верхнее ограничение метки времени
Использовать предварительный выбор	Нет
Время, отведенное на отправку команды в секундах	15
Количество повторов команд управления при их не доставке	0
Отправлять команды во все источники	Нет
Опрашивать все источники	Нет
Каналы	
<b>Верхнее ограничение метки времени</b>	
Предельное отклонение метки времени сигнала, сек	5
Игнорировать инверсии метки времени при записи в ядро	Нет
<b>Принадлежность доверительному интервалу</b>	
Негативное отклонение метки времени, сек	60
Допустимое отклонение метки времени, сек	60
Максимальное допустимое отклонение метки времени, сек	52560000
<b>Получение файлов</b>	
Включить получение файлов	Нет
Путь к папке для полученных файлов	C:\IEC-870-5\ReceivedFiles
Интервал обновления информации о директории, сек	60
Имя подкаталога	0
Адрес подкаталога	0
<b>Общие</b>	
Подчиненная станция	lec104Slave

## Параметры станции

Параметр	Описание
Интервал опроса, мин.	Период циклического опроса подчиненной станции (по умолчанию – 30 минут). При значении 0 команда общего опроса не подаётся.
Интервал синхронизации станций, мин.	Период циклической синхронизации времени подчиненной станции с временем сервера (по умолчанию – 30 минут). При значении 0 команда синхронизации времени не подаётся.
Таймаут потери связи, сек.	Промежуток времени, после которого связь со станцией считается потерянной (по умолчанию – 0 мс).
Отправлять значениями false (C_SC_NA_1) со значениями true	Если флаг установлен, то допускается отправка данного типа команд управления со значением false. Если флаг снят, данный тип команд отправляется только со значением true.
Максимальный размер очереди приема данных	Размер очереди значений для записи в сервер, при достижении которого заносится соответствующая запись в журнал, но опрос не останавливается (значение по умолчанию 10000).
Максимальный размер очереди данных на отправку	Размер очереди управляющих воздействий для станции, при достижении которого заносится соответствующая запись в журнал, но подача управляющих воздействий не останавливается (значение по умолчанию 1000).
Отправлять все исходящие сигналы при активации источника	Если флаг установлен, то после активации источника в очередь на отправку добавятся текущие значения всех исходящих сигналов, имеющих приемлемое качество и значение. Количество изменений, отправляемых при активации источника, отражено в параметре статистики исходящей нагрузки для станции.

Сдвиг метки времени	Временной сдвиг в часах, применяемый для корректировки метки времени, если время на сервере и станции отличается.
Алгоритм валидации входящих значений	Выбор алгоритма обработки входящих значений сигналов в зависимости от метки времени: <ul style="list-style-type: none"> <li>➤ Проверять верхнее ограничение метки времени — если метка времени сигнала превышает величину предельного отклонения, то сигнал считается недостоверным с качеством <code>QUALITY_DEVICE_FAILURE</code> (12). Алгоритм используется по умолчанию;</li> <li>➤ Проверять принадлежность метки времени доверительному интервалу — если метка времени сигнала попадает в определённый диапазон относительно текущей метки времени сервера и метки времени последнего сохраненного значения, то сигналу устанавливаются качество и метка времени, соответствующие данному диапазону.</li> </ul>
Опрашивать все источники	Если флаг установлен, модуль запрашивает данные у станции одновременно по всем каналам. При этом в сигналы сохраняются лишь данные от активного источника.
Время, отведенное на отправку команды в секундах	Время, в течение которого предпринимаются попытки отправки управляющих воздействий станции.
Количество повторов команд управления при их не доставке	Параметр определяет число попыток отправки управляющей команды.
Использовать предварительный выбор	Если флаг установлен, перед отправкой управляющего воздействия на станцию подаётся запрос, который определяет допустима ли данная команда для станции.
Отправлять команды во все источники	Если флаг установлен, модуль отправляет команду одновременно по всем каналам.



## Верхнее ограничение метки времени

Параметр	Описание
Предельное отклонение метки времени сигнала, сек.	Величина предельного отклонения метки времени сигнала. Если значение параметра равно 0, то контроль отклонения не ведется.
Игнорировать инверсии метки времени при записи в ядро	Если флаг установлен, проверка на предельное отклонение метки времени не проводится, установка в ядро производится только при изменении качества или значения. Функция работает только если в источнике содержится 1 канал связи.

## Принадлежность доверительному интервалу

Параметр	Описание
Негативное отклонение метки времени, с	Минимальное негативное отклонение метки времени полученного значения относительно метки времени сохраненного значения, при котором значение считается недостоверным. Значение по умолчанию 60 секунд;
Допустимое отклонение метки времени, с	Максимальное позитивное отклонение метки времени полученного значения относительно метки времени сервера, при котором значение считается достоверным. Значение по умолчанию 60 секунд;
Максимальное допустимое отклонение метки времени, с	Максимальное позитивное отклонение метки времени полученного значения относительно метки времени сервера, при котором значение не будет учитываться. Значение по умолчанию 52560000 минут (100 лет);

## Получение файлов

Параметр	Описание
----------	----------

Включить получение файлов	Значения: <ul style="list-style-type: none"> <li>› Да</li> <li>› Нет</li> </ul>
Путь к папке для полученных файлов	Путь до папки, где будут сохраняться полученные файлы
Интервал обновления информации директории, сек	Интервал обновления, по умолчанию 60 секунд
Имя подкаталога	Идентификатор подкаталога
Адрес подкаталога	Адрес подкаталога

## 1.1.2.6.3.3. Функциональное содержание

### Качество сигналов

Качества сигналов, выставляемых коммуникационным модулем приведены в таблице ниже.

Числовое значение качества	Идентификатор качества	Расшифровка
4	CONFIG_ERROR	сигнал неправильно сконфигурирован
8	NOT_CONNECTED	сигнал принят на обслуживание, но устройство еще не инициализировано
12	DEVICE_FAILURE	ошибка инициализации устройства
24	COMM_FAILURE	КП нет на связи
28	OUT_OF_SERVICE	модуль не запущен
64	UNCERTAIN	связь установлена, но значения еще не пришли
192	GOOD	значение сигнала достоверно
216	LOCAL_OVERRIDE	значение достоверно, введено вручную

Для обеспечения дополнительной информации о качестве сигнала модуль Опросчик МЭК 60870-5-104 поддерживает дополнительные статусы для определенных протокольных типов сигналов. Статус состоит из шести определенных битов (флагов) качества, которые могут устанавливаться независимо друг от друга:

Флаг	Описание
OV	Флаг переполнения. Значение величины объекта информации лежит вне заранее определенного диапазона значений: <ul style="list-style-type: none"><li>&gt;0 — нет переполнения;</li><li>&gt;1 — переполнение.</li></ul>

BL	<p>Флаг блокировки. Значение величины объекта информации заблокировано для передачи, оно остается в состоянии, в котором было до блокировки. Блокировка и деблокировка инициируются местными блокирующим устройством или автоматически на основании местной величины:</p> <ul style="list-style-type: none"> <li>›0 — нет блокировки;</li> <li>›1 — блокировка.</li> </ul>
SB	<p>Флаг замещения. Значение величины объекта информации поступает на вход от оператора или от автоматического источника:</p> <ul style="list-style-type: none"> <li>›0 — нет замещения;</li> <li>›1 — произведено замещение.</li> </ul>
NT	<p>Флаг актуальности. Значение величины актуально, если большинство опросов было успешным. Оно неактуально, если оно не обновлялось в течение заданного промежутка времени или было недоступно:</p> <ul style="list-style-type: none"> <li>›0 — актуальное значение;</li> <li>›1 — неактуальное значение.</li> </ul>
IV	<p>Флаг действительности. Значение величины действительно, если правильно получено. Данный флаг используется для указания месту назначения, что значение величины может быть неправильным и не должно использоваться:</p> <ul style="list-style-type: none"> <li>›0 — действительное значение;</li> <li>›1 — недействительное значение.</li> </ul>
СУ	<p>Флаг переноса:</p> <ul style="list-style-type: none"> <li>›0 — за соответствующий период интегрирования не произошло переполнения счетчика;</li> <li>›1 — за соответствующий период интегрирования произошло переполнения счетчика.</li> </ul>

При взведенных флагах статуса сигналу выставляется определенное ОРС качество. Преобразование статуса в качество для сигналов стандартного диапазона протокольных типов с номерами идентификаторов 1, 3, 30, 31 приведено в таблице ниже.

№ бита	Взведенный флаг	ОРС качество
--------	-----------------	--------------

1	BL	GOOD_Spec3 (204)
2	SB	GOOD_Spec2 (200)
3	—	GOOD (192)
4	NT	UNCERTAIN (64)
5	IV	BAD (0)

Преобразование статуса в качество для сигналов стандартного диапазона протокольных типов с номерами идентификаторов 5, 7, 9, 11, 13, 20, 32, 33, 34, 35, 36 приведено в таблице ниже.

№ бита	Взведенный флаг	ОПС качество
1	BL	GOOD_Spec3 (204)
2	SB	GOOD_Spec2 (200)
3	—	GOOD (192)
4	NT	UNCERTAIN Engineering Units Exceeded (84)
5	NT	UNCERTAIN (64)
6	IV	BAD (0)

Если в регистре статуса присутствует несколько взведенных флагов, то результирующее ОПС качество будет наихудшим среди качеств, соответствующих каждому из флагов.

## Параллельный опрос источника

Модуль Опросчик МЭК 60870-5-104 имеет возможность обмена данными с активным источником станции по нескольким каналам связи в параллельном режиме. Максимальное количество каналов связи Master-Slave равно 4. Количество каналов настраивается в параметрах конфигурации.

Модуль Опросчик МЭК 60870-5-104 производит опрос активного источника в параллельном режиме (производится опрос по всем активным каналам источника). Активным источником является, тот в составе которого имеется как минимум один канал с работающим протоколом МЭК. При старте модуля Опросчик МЭК 60870-5-104 активным станет тот источник, который первым установит TCP соединение хотя бы по одному каналу связи.

В процессе работы модуля может произойти смена активного источника по следующим причинам:

- ошибка по протоколу МЭК;
- ошибка TCP соединения;
- принудительная смена активного источника командой пользователя;
- захват активности источником с более высоким приоритетом.

Автоматическая смена активного источника происходит при появлении потенциально активного источника с более высоким приоритетом или при разрыве связи по всем каналам с текущим активным источником. Потенциально активным источником считается тот, у которого есть TCP соединение, как минимум по одному каналу связи. Появление на связи потенциально активного источника сопровождается проверкой его приоритета. Если приоритет потенциально активного источника выше приоритета активного источника, то происходит смена активного источника на новый источник. Если при появлении потенциально активного источника у станции ещё не было активного источника, то потенциально активный источник становится активным автоматически.

Деактивация активного источника сопровождается посылкой команды STOPDT по всем каналам источника. При деактивации источника TCP соединение не рвется и продолжается обмен кадрами жизнеохраны (test\_act, test\_con).

Активация источника сопровождается параллельной рассылкой по всем его каналам (с установленным TCP соединением) команд: STARTDT, синхронизация времени и общий опрос.

Параллельный опрос не допускает использование сигналов без метки времени. При параллельном опросе источника по нескольким каналам модуль получает несколько значений сигнала от источника, но в ядро устанавливается значение сигнала с меткой времени позднее текущей (в данный момент установленной в ядре). Таким образом повторное получение данных игнорируется.

Управляющие воздействия подаются параллельно по всем каналам источника, которые имеют активированный МЭК. По первому подтверждению доставки кадра с управляющей информацией, формируется сигнал доставки со значением равным 3. Остальные подтверждения, не генерируют изменения значения сигнала доставки.

## **Алгоритмы валидации входящих значений**

Модуль Опросчик МЭК 60870-5-104 поддерживает два алгоритма валидации входящих значений с меткой времени:

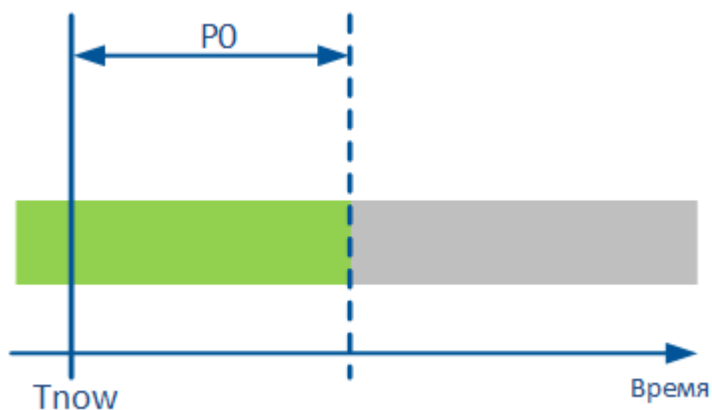
- › Проверка верхнего ограничения метки времени;
- › Проверка принадлежности метки времени доверительному интервалу.

В результате обработки сигналу устанавливаются качество и метка времени с учетом параметров выбранного алгоритма.

Алгоритм и параметры обработки задаются при конфигурировании модуля в настройках станции.

### **Проверка верхнего ограничения метки времени**

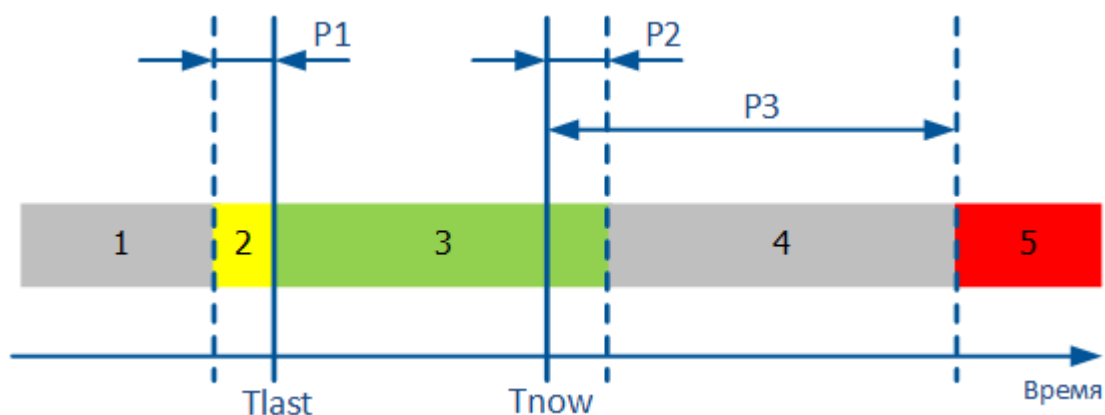
Если метка времени сигнала превышает величину предельного отклонения (PO) относительно текущего времени сервера  $T_{now}$ , то сигнал считается недостоверным с качеством `QUALITY_DEVICE_FAILURE` (12).



Величина  $P0$  задаётся в параметре настройки станции (A1) Предельное отклонение метки времени сигнала, сек. Если значение параметра равно 0, то контроль отклонения метки времени не ведётся.

## Проверка принадлежности метки времени доверительному интервалу

Если метка времени сигнала  $T_s$  попадает в определённый интервал относительно текущей метки времени сервера  $T_{now}$  и метки времени последнего сохраненного значения  $T_{last}$ , то сигналу устанавливаются качество и метка времени, соответствующие данному интервалу.



Отклонение	Описание
P1	Минимальное негативное отклонение метки времени полученного значения относительно метки времени последнего сохраненного значения $T_{last}$ , при котором значение считается



	недостовверным. Задаётся в параметре настройки станции (A2) Негативное отклонение метки времени, с. Значение по умолчанию 60 секунд.
P2	Максимальное позитивное отклонение метки времени полученного значения относительно метки времени сервера Tnow, при котором значение считается достоверным. Задаётся в параметре настройки станции (A2) Допустимое отклонение метки времени, с. Значение по умолчанию 60 секунд.
P3	Максимальное позитивное отклонение метки времени полученного значения относительно метки времени сервера Tnow, при котором значение не будет учитываться. Задаётся в параметре настройки станции (A2) Максимальное допустимое отклонение метки времени, с. Значение по умолчанию 52560000 минут (100 лет).

В таблице приведены значения качества и метки времени, устанавливаемые сигналу при попадании в соответствующий интервал.

Интервал	Условие интервала	Качество	Метка времени
1	$T_s \leq T_{last} - P1$	OPC_QUALITY_BAD (0)	$T_{last} + 1 \text{ мс}$
2	$T_{last} - P1 < T_s \leq T_{last}$	OPC_QUALITY_GOOD (192)	$T_{last} + 100 \text{ нс}$
3	$T_{last} < T_s \leq T_{now} + P2$	OPC_QUALITY_GOOD (192)	$T_s$
4	$T_{now} + P2 < T_s \leq T_{now} + P3$	OPC_QUALITY_BAD (0)	$T_{now}$
5	$T_{now} + P3 < T_s$	Значение отбрасывается	

## Отправка управляющих и регулирующих воздействий

Под управляющими и регулируемыми воздействиями подразумеваются команды TU и TP. Отправка команд выполняется, когда пара резервируемых серверов активна и Astra.Server находится в режиме РАБОТА.

Чтобы сразу после активации источника в очередь на отправку добавились текущие значения всех исходящих сигналов, используйте [параметр](#) Отправлять все исходящие сигналы при активации источника.

Для команд ТУ/ТР модуль позволяет настраивать сигналы доставки. Значения сигналов доставки дает понимание в каком состоянии находится отправленная команда.

Алгоритм отправки команд ТУ/ТР и формирования значения сигнала доставки:

1. Каждая станция обладает собственной независимой [исходящей очередью команд](#).
2. Добавление команд ТУ/ТР в очередь происходит после успешного прохождения проверок:
  - управляющие данные должны быть хорошего качества;
  - сервер должен находиться в состоянии, подразумевающим управление.

Признаком успешного помещения команды ТУ/ТР в очередь является изменение значения соответствующего сигнала доставки на 1. Если данные не прошли проверку, то значение сигнала доставки меняется на -4.

3. Проверка первого элемента очереди на наличие устаревших данных ([параметр](#) Время отведенное на отправку команды в секундах). Устаревшие команды ТУ/ТР изымаются из очереди, а их сигналам доставки устанавливается значение -3.

4. Отправка команд ТУ/ТР:

- команды отправляются в активный источник;
- отправка команды в активный источник осуществляется [параллельно по всем каналам](#);
- некоторые каналы могут отказывать в передаче (нет соединения, не активен протокол МЭК, канал находится в состоянии ожидания);
- отправка считается успешной, если хотя бы один канал отправил кадр с командой;
- отправленные команды ТУ/ТР изымаются из очереди;

➤ после каждой успешной попытки отправки команды увеличивается специальный счетчик и создается структура данных для ожидания подтверждения;

➤ значение сигнала доставки меняется на 2, если была успешная попытка отправки команды.

5. Если доставка не подтверждена, то проверяется не закончились ли попытки отправки ([параметр](#) Количество повторов команд управления при их недоставке):

➤ если попытки отправки подошли к концу, то сигналу доставки присваивается значение -2;

➤ если ещё имеются попытки отправки, то команды снова возвращаются в очередь для новой попытки отправки.

6. Если доставка подтверждена, то значение сигнала доставки меняется на 3. Подтверждение отправляется по любому из каналов, которые были задействованы при отправке команды.

Ниже представлена таблица с возможными значениями сигнала доставки.

Состояние	Значение
Команды успешно помещены в очередь на отправку	1
В ожидании подтверждения доставки	2
Доставка подтверждена	3
Не удалось отправить	-1
Подтверждение доставки не получено	-2
Команды устарели и были изъяты из очереди на отправку	-3
Предварительные проверки не были пройдены и команды ТУ/ТР не были помещены в очередь на отправку	-4

## Синхронизация времени

Чтобы иметь правильную хронологическую последовательность данных, которые передаются модулю, время подчиненной станции должно

быть синхронизировано с временем модуля. В начале работы модуля синхронизация происходит после инициализации системы.

Команда синхронизации времени содержит информацию о дате и времени. Информация должна быть скорректирована на станции. Величина корректировки времени определяется как сумма задержки передачи и произведения длины кадра синхронизации на скорость передачи, согласно ГОСТ Р МЭК 60870-5-104. Время исчисляется в десяти миллисекундных тиках с начала года.

Синхронизация времени между подчиненной станцией и модулем Опросчик МЭК 60870-5-104 выполняется в случаях если:

1. В настройках модуля Опросчик МЭК 60870-5-104 установлен параметр Интервал синхронизации станции, мин.
2. Подчиненной станция имеет возможность корректировать время или в настройках модуля IEC Slave установлен флаг Корректировать время по команде опросчика.

## Служебные сигналы модуля

Ниже приведен список служебных сигналов модуля Опросчик МЭК 60870-5-104. Их изменение влечет за собой изменение логики работы модуля. Значения нижеперечисленных сигналов можно задать с помощью OPC-клиента.

Полный тег сервисных сигналов имеет вид:



```
Service.Modules.IEC-104 Master.Stations.StationN.<Имя сигнала>
```

Смена активного источника:

- › ChangeActiveSource.SourceNumber — номер источника, который нужно сделать активным.
- › ChangeActiveSource.Command — команда для смены активного источника.

- › `ChangeActiveSource.BoolResult` — результат команды смены источника в булевом формате.
- › `ChangeActiveSource.StringResult` — результат смены источника в строковом формате.
- › `Sources.Source1.Priority` — задать приоритет источника. Может быть числом от 0 до 255.

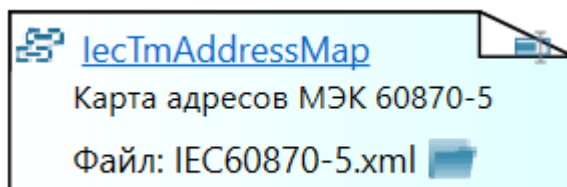
Общий опрос источника:

- › `CommonPoll.Command` — подать команду общего опроса в активный источник.
- › `CommonPoll.BoolResult` — результат подачи команды на общий опрос в активный источник. Изменение приоритета источника модуль воспринимает динамически.

## 1.1.2.6.3.4. Карта адресов

Для настройки сигналов модуля Опросчик МЭК 60870-5-104 используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов МЭК 60870-5 в исполняемое приложение;



2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	Адрес	Протокольный тип	Интервал нечувствительности	Положение десятичной запятой	Номер бита	Категория данных
string	string	непосредстве	0	U-CTRL				
string1	string	непосредстве	0	U-MON				
bool	bool	непосредстве	0	45: C_SC_NA_1				
uint2	uint2	непосредстве	0	33: M_BO_TB_1				

## Параметры карты адресов

Параметр	Значение
Привязка	Тип привязки параметра: <ul style="list-style-type: none"><li>› непосредственно;</li><li>› только чтение;</li><li>› не привязан.</li></ul>
Адрес	Адрес элемента информации. Размер адреса элемента информации 3 байта. Значения в диапазоне от 0 до 16777215
Протокольный тип	Идентификатор типа. Определяет структуру, тип и формат элемента информации. Модуль поддерживает <a href="#">стандартные</a> и <a href="#">частные</a> типы
Интервал нечувствительности	Величина, при превышении которой значение сигнала с подчиненной станции отправляется ведущей станции

Номер бита	Номер бита в байте. Значения в диапазоне от 0 до 7. Устанавливается для протокольных типов TS и TC
Положение десятичной запятой	Положение десятичной запятой. Соответствует числу, равному степени числа 10, и применяется в качестве множителя к полученному значению сигнала. Значения в диапазоне от -41 до +33.


# 1.1.2.6.3.5. Диагностика работы модуля

## Журнал работы


Каждый модуль ведет журнал работы. В него сохраняется вся информация о работе модуля и об обмене данными с подчиненными станциями. Журнал работы модуля предназначен для контроля работы модуля в режиме реального времени, а также просмотра прошедших событий модуля.

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:

 C:\Program Files\AstraRegul\Astra.Server\Logs;

» в Linux системах в директории:

 /opt/AstraRegul/Astra.Server/Logs.

Для просмотра журнала работы модуля используется сервисное приложение Просмотрщик лога кадров.

№	Дата	Время	Описание	Станция	Удаленный адрес	Локальный адрес	№	Адрес	Значение	Метка времени
12985	09.11.2021	14:33:33:553	Входящий кадр <15> M_IT_NA_1 (S:13600;R:2787)	1	172.16.150.25:2404	172.16.150.17:627	1	361	0	
12986	09.11.2021	14:33:33:553	Исходящий кадр S-кадр (R:13601)	1	172.16.150.25:2404	172.16.150.17:627				
12987	09.11.2021	14:33:33:553	Входящий кадр <162> Упакованные данные (S:13601;R:2787)	1	172.16.150.25:2404	172.16.150.17:627				
12988	09.11.2021	14:33:33:553	Входящий кадр <158> STR (S:13602;R:2787)	1	172.16.150.25:2404	172.16.150.17:627				
12989	09.11.2021	14:33:33:553	Входящий кадр <162> Упакованные данные (S:13603;R:2787)	1	172.16.150.25:2404	172.16.150.17:627				
12990	09.11.2021	14:33:33:553	Входящий кадр <190> STR_Status (S:13604;R:2787)	1	172.16.150.25:2404	172.16.150.17:627				
12991	09.11.2021	14:33:33:553	Входящий кадр <162> Упакованные данные (S:13605;R:2787)	1	172.16.150.25:2404	172.16.150.17:627				
12992	09.11.2021	14:33:33:646	Входящий кадр <222> STR_Time (S:13606;R:2787)	1	172.16.150.25:2404	172.16.150.17:627				
12993	09.11.2021	14:33:33:646	Входящий кадр <162> Упакованные данные (S:13607;R:2787)	1	172.16.150.25:2404	172.16.150.17:627				
12994	09.11.2021	14:33:33:646	Входящий кадр <230> M_IT_ND_1 (S:13608;R:2787)	1	172.16.150.25:2404	172.16.150.17:627				
12995	09.11.2021	14:33:33:646	Исходящий кадр S-кадр (R:13609)	1	172.16.150.25:2404	172.16.150.17:627				
12996	09.11.2021	14:33:33:693	Входящий кадр <230> M_IT_ND_1 (S:13609;R:2787)	1	172.16.150.25:2404	172.16.150.17:627				
12997	09.11.2021	14:33:33:693	Входящий кадр <231> M_IT_TD_1 (S:13610;R:2787)	1	172.16.150.25:2404	172.16.150.17:627				

0	1	2	3	4	5	6	7	8	9
0000	68	1E	50	EA	CE	18	EE	01	14 00
0001	01	00	E9	01	00	00	00	00	00
0002	00	00	00	80					



## 1.1.2.6.4. Станция МЭК 60870-5-104

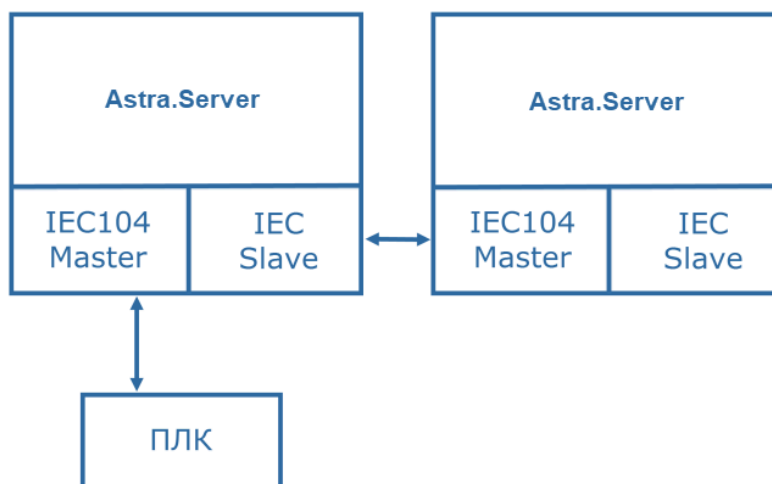


Станция МЭК 60870-5-104 реализует подчиненную станцию в соответствии с ГОСТ Р МЭК 870-5-104-2004 и расширением стандартного диапазона типов IEC\_CT1.

Основными функциями станции МЭК 60870-5-104 являются:

- передача данных опросчику;
- прием команд от управляющей станции.

Работая в составе Astra.Server, станция МЭК 60870-5-104 реализует подчиненную станцию: получает данные со станций при изменении значения параметра и передает полученные данные Мастеру (опросчику), а также выполняет команды, пришедшие от активного Мастера (опросчика). Полученные данные через интерфейсы ядра сохраняются в оперативную базу данных (ОБД). Ядро сервера при получении изменившихся сигналов уведомляет о них другие модули, которые обслуживают эти сигналы, в том числе и модуль OPC DA Server. Он в свою очередь передает данные клиентам, подписанным на изменившиеся сигналы.



Модуль поддерживает работу с любыми опросчиками, которые соответствуют ГОСТ Р МЭК 870-5-104-2004.

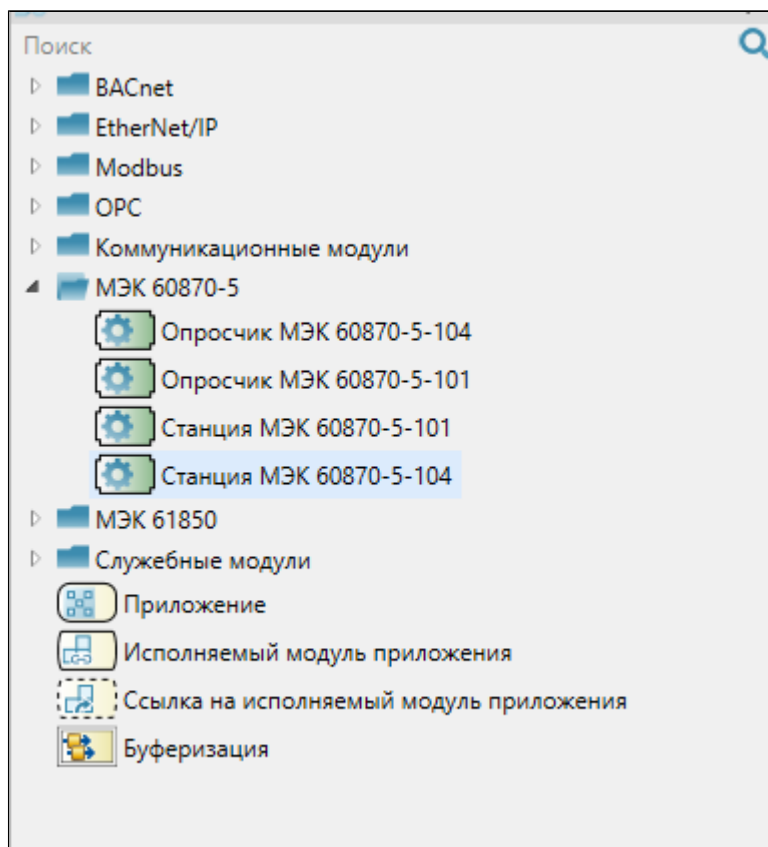
Модуль имеет возможность реализовывать более одной подчиненной станции.

Модуль отправляет данные при изменении значения параметра и качества сигнала.

Модуль работает в двух режимах: РАБОТА и РЕЗЕРВ.

# Настройка модуля

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

Свойства	
Iec104Slave1 Станция МЭК 60870-5-104	
<b>Параметры модуля</b>	
Учитывать качество при общем опросе	Нет
Смена типа при передаче по команде общий опрос	Да
Максимальный размер буфера	4096
Базовый порт станций	2404
Настройки	
Настройки по умолчанию	
Работать в РЕЗЕРВЕ	Нет
Корректировать время по команде опросчика	Нет
Сбрасывать ТУ	Нет
Задержка сброса ТУ, мс	False
Менять качество входных сигналов	Да
Активность	Да
Отображаемое имя	
<b>Параметры взаимодействия</b>	
Отправлять подтверждение исполнения (активации)	Нет
Отправлять завершение активации в ответ на команду	Нет
Отправлять подтверждение общего опроса	Нет
Отправлять завершение общего опроса	Нет
<b>Общие</b>	
Карта адресов	
Имя	Iec104Slave1
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения



Для параметров модуля допустимо использование значений по умолчанию, однако рекомендуется устанавливать параметру

"Активность" значение "Да", чтобы модуль запускался при запуске/перезапуске сервера.

## Общие

Параметр	Описание
Имя	Имя модуля
Карта адресов	Карта адресов МЭК 60870-5, которая будет использована для данного адаптера

## Параметры модуля

Параметр	Описание
Учитывать качество при общем опросе	Данные для отправки в ответ на общий опрос отбирать с учетом качества Значения: > Да > Нет
Смена типа при передаче по команде общий опрос	Значения: > Да > Нет
Базовый порт станций	Если данный флаг сброшен, то соединение с мастером в резервном режиме не устанавливается Значения: > Да > Нет
Настройки	Значения: > Да > Нет

Настройки по умолчанию	Значения: <ul style="list-style-type: none"> <li>› Да</li> <li>› Нет</li> </ul>
Работать В РЕЗЕРВЕ	Если данный флаг сброшен, то соединение с мастером в резервном режиме не устанавливается Значения: <ul style="list-style-type: none"> <li>› Да</li> <li>› Нет</li> </ul>
Корректировать время по команде опросчика	Значения: <ul style="list-style-type: none"> <li>› Да</li> <li>› Нет</li> </ul>
Сбрасывать ТУ	Значения: <ul style="list-style-type: none"> <li>› Да</li> <li>› Нет</li> </ul>
Задержка сброса ТУ, мс	Значение по умолчанию: False
Менять качество входных сигналов	Менять качество входных сигналов при изменении статуса соединения Значения: <ul style="list-style-type: none"> <li>› Да</li> <li>› Нет</li> </ul>
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>› Да – модуль запущен;</li> <li>› Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

# Функциональное содержание

## Режимы работы модуля

Модуль Станция МЭК 60870-5-104 работает в двух режимах: РАБОТА или РЕЗЕРВ. Режим работы модуля в Astra.Server устанавливает ядро. В процессе работы режим может измениться.

В режиме РЕЗЕРВ модуль работает в соответствии с флагом конфигурации Работать в РЕЗЕРВЕ:

- › Да — модуль работает в полнофункциональном режиме, независимо от состояния активности резервной пары;
- › Нет — модуль в режиме РЕЗЕРВ не отвечает на попытку соединения Мастером.

## Работа с опросчиком

При запуске модуля Станция МЭК 60870-5-104 происходит его инициализация. Процедура инициализации работы модуля требуется для установки модуля в правильное рабочее состояние до того, как начнется передача данных при опросе. Процесс инициализации включает в себя:

- › чтение конфигурационных данных модуля;
- › подписка у ядра на получение уведомлений об изменениях сигналов;
- › проверка корректности настройки адресов сигналов;
- › постановка сигналов на обслуживание.

Так же во время инициализации модуля устанавливается соединение с опросчиками. При установленном соединении модуль отправляет Мастеру и станциям сообщение о своей готовности принимать и отправлять данные.

Для того чтобы модуль принимал управляющие воздействия от Мастера, необходима подписка на сигнал управления.

## Качество сигналов

Для обеспечения дополнительной информации о качестве сигнала модуль Станция МЭК 60870-5-104 поддерживает дополнительные статусы для определенных протокольных типов сигналов. Статус состоит из шести определенных битов (флагов) качества, которые могут устанавливаться независимо друг от друга.

При взведенных флагах статуса сигналу выставляется определенное OPC качество. Преобразование статуса в качество для сигналов [стандартного диапазона протокольных типов](#) с номерами идентификаторов 1, 3, 30, 31 приведено в таблице ниже.

№ бита	Взведенный флаг	OPC качество
1	BL	GOOD_Spec3 (204)
2	SB	GOOD_Spec2 (200)
3	—	Любое GOOD (192) кроме п. 1 и 2
4	NT	Любое качество UNCERTAIN
5	IV	Любое качество BAD

Преобразование статуса в качество для сигналов стандартного диапазона протокольных типов с номерами идентификаторов 5, 7, 9, 11, 13, 20, 32, 33, 34, 35, 36 приведено в таблице ниже.

№ бита	Взведенный флаг	OPC качество
1	CY	GOOD_Spec1 (196)
2	—	Любое GOOD (192) кроме п. 1
3	IV	Любое качество BAD или UNCERTAIN

Если в регистре статуса присутствует несколько взведенных флагов, то результирующее OPC качество будет наихудшим среди качеств, соответствующих каждому из флагов.

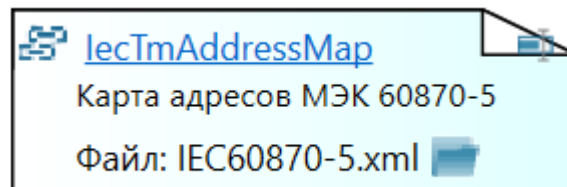




## 1.1.2.6.4.3. Карта адресов

Для настройки сигналов модуля Станция МЭК 60870-5-104 используется приложение Astra.AStudio. Для добавления сигналов необходимо выполнить следующие действия:

1. Добавьте карту адресов МЭК 60870-5 в исполняемое приложение;



2. Откройте редактор карты адресов.

Сигнал	Тип	Привязка	Адрес	Протокольный тип	Интервал нечувствительности	Положение десятичной запятой	Номер бита	Категория данных
string	string	непосредстве	0	U-CTRL				
string1	string	непосредстве	0	U-MON				
bool	bool	непосредстве	0	45: C_SC_NA_1				
uint2	uint2	непосредстве	0	33: M_BO_TB_1				

## Параметры карты адресов

Параметр	Значение
Привязка	Тип привязки параметра: <ul style="list-style-type: none"><li>› непосредственно;</li><li>› только чтение;</li><li>› не привязан.</li></ul>
Адрес	Адрес элемента информации. Размер адреса элемента информации 3 байта. Значения в диапазоне от 0 до 16777215
Протокольный тип	Идентификатор типа. Определяет структуру, тип и формат элемента информации. Модуль поддерживает <a href="#">стандартные</a> и <a href="#">частные</a> типы
Интервал нечувствительности	Величина, при превышении которой значение сигнала с подчиненной станции отправляется ведущей станции

Номер бита	Номер бита в байте. Значения в диапазоне от 0 до 7. Устанавливается для протокольных типов TS и TC
Положение десятичной запятой	Положение десятичной запятой. Соответствует числу, равному степени числа "10", и применяется в качестве множителя к полученному значению сигнала. Значения в диапазоне от "-41" до "+33".

# Диагностика работы модуля

## Журнал работы

Каждый модуль ведет журнал работы. В него сохраняется вся информация о работе модуля и об обмене данными с подчиненными станциями. Журнал работы модуля предназначен для контроля работы модуля в режиме реального времени, а также просмотра прошедших событий модуля.

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

» в Linux системах в директории:



/opt/AstraRegul/Astra.Server/Logs.

Для просмотра журнала работы модуля используется сервисное приложение Просмотрщик лога кадров.

№	Дата	Время	Описание	Станция	Удаленный адрес	Локал	№	Адрес	Значение	Метка вре...	Качество	Но...	Тип				
166526	09.11.2021	05:48:43:799	Исходящий кадр <15> M_IT_NA_1 (S:10172;R:1709)	1	172.16.150.18:53040	172.	1	294			OPC_BAD (0xFF)		STR_Status				
166527	09.11.2021	05:48:43:799	Исходящий кадр <162> Упакованные данные (S:10173;R:1709)	1	172.16.150.18:53040	172.	Исходящий кадр <190> STR_Status (S:10176;R:1709)										
166528	09.11.2021	05:48:43:799	Исходящий кадр <158> STR (S:10174;R:1709)	1	172.16.150.18:53040	172.											
166529	09.11.2021	05:48:43:799	Исходящий кадр <162> Упакованные данные (S:10175;R:1709)	1	172.16.150.18:53040	172.											
166530	09.11.2021	05:48:43:799	Исходящий кадр <190> STR_Status (S:10176;R:1709)	1	172.16.150.18:53040	172.											
166531	09.11.2021	05:48:43:845	Входящий кадр S-кадр (R:10173)	1	172.16.150.18:53040	172.											
166532	09.11.2021	05:48:43:845	Исходящий кадр <162> Упакованные данные (S:10177;R:1709)	1	172.16.150.18:53040	172.	0000	68	0F	80	4F	5A	0D	BE	01	14	00
166533	09.11.2021	05:48:43:845	Исходящий кадр <222> STR Time (S:10178;R:1709)	1	172.16.150.18:53040	172.	0001	01	00	26	01	00	00	80			

# МЭК стандартный диапазон типов

## Информация о процессе в направлении контроля (Slave → Master)

ID	Символьный ID	Тип сервера	Диапазон значений	Описание
1	M_SP_NA_1	bool	[true; false]	Одноэлементная информация
3	M_DP_NA_1	uint1	[0; 3]	Двухэлементная информация
5	M_ST_NA_1	int1	[-64; 63]	7 бит значащей информации
7	M_BO_NA_1	uint4	[0; 4294967295]	Строка из 32 бит
9	M_ME_NA_1	float	[-1; 0,999969]	Значение измеряемой величины, нормализованное значение
11	M_ME_NB_1	float	[-327680; 327669,999999]	Значение измеряемой величины, масштабированное значение
13	M_ME_NC_1	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Значение измеряемой величины, короткий формат с плавающей запятой
15	M_IT_NA_1	int4	[-2147483648; 2147483647]	Интегральные суммы
21	M_ME_ND_1	float	[-1; 0,999969]	Значение измеряемой величины, нормализованное значение без описателя качества

30	M_SP_TB_1	bool	[true; false]	Одноэлементная информация с меткой времени CP56Время2а
31	M_DP_TB_1	uint1	[0; 3]	Двухэлементная информация с меткой времени CP56Время2а
32	M_ST_TB_1	int1	[-64; 63]	7 бит значащей информации с меткой времени CP56Время2а
33	M_BO_TB_1	uint4	[0; 4294967295]	Строка из 32 бит с меткой времени CP56Время2а
34	M_ME_TD_1	float	[-1; 0,999969]	Значение измеряемой величины, нормализованное значение с меткой времени CP56Время2а
35	M_ME_TE_1	float	[-327680; 327669,999999]	Значение измеряемой величины, масштабированное значение с меткой времени CP56Время2а
36	M_ME_TF_1	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Значение измеряемой величины, короткий формат с плавающей запятой с меткой времени CP56Время2а
37	M_IT_TB_1	int4	[-2147483648; 2147483647]	Интегральная сумма с меткой времени CP56Время2а
38	M_EP_TD_1	uint1	[0; 3]	Действие устройств защиты с меткой времени CP56Время2а

## Информация о процессе в направлении управления (Master → Slave)

ID	Символьный ID	Тип сервера	Диапазон значений	Описание
45	C_SC_NA_1	bool	[true; false]	Однопозиционная команда
46	C_DC_NA_1	uint1	[0; 3]	Двухпозиционная команда
47	C_RC_NA_1	uint1	[0; 3]	Команда пошагового регулирования
48	C_SE_NA_1	float	[-1; 0,999969]	Команда уставки, нормализованное значение
49	C_SE_NB_1	float	[-327680; 327669,999999]	Команда уставки, масштабированное значение
50	C_SE_NC_1	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Команда уставки, короткий формат с плавающей запятой
51	C_BO_NA_1	uint4	[0; 4294967295]	Строка из 32 бит
58	C_SC_TA_1	bool	[true; false]	Однопозиционная команда с меткой времени CP56Время2а
59	C_DC_TA_1	uint1	[0; 3]	Двухпозиционная команда с меткой времени CP56Время2а
60	C_RC_TA_1	uint1	[0; 3]	Команда пошагового регулирования с меткой времени CP56Время2а
61	C_SE_TA_1	float	[-1; 0,999969]	Команда уставки, нормализованное значение с меткой времени CP56Время2а

62	C_SE_TB_1	float	[-327680; 327669,999999]	Команда уставки, масштабированное значение с меткой времени CP56Время2а
63	C_SE_TC_1	float	[ $\pm 1.5 \times 10^{-45}$ ; $\pm 3.4 \times 10^{38}$ ]. Точность 6-9 цифр	Команда уставки, короткий формат с плавающей запятой с меткой времени CP56Время2а
64	C_BO_TA_1	uint4	[0; 4294967295]	Строка из 32 бит с меткой времени CP56Время2а



# МЭК частный диапазон типов

## Информация о процессе в направлении контроля (Slave → Master)

ID	Символьный ID	Тип сервера	Диапазон значений	Описание
137	U-MON	Любой	Соответствующий типу	Унифицированный мониторинг
144	TS	bool	[true; false]	Состояние контролируемого объекта без метки времени. Размер 1 байт. Каждый бит байта отвечает за независимый объект информации. Адрес для всех объектов один, но дополнительно в конфигурации задаётся номер бита для каждого объекта
146	TM1	int1 или uint1	[-128, 127] или [0, 255]	Значение измеряемой величины, целое число размером 1 байт без метки времени
147	TM2	int2 или uint2	[-32768, 32767] или [0, 65535]	Значение измеряемой величины, целое число размером 2

				байта без метки времени
148	TMF4	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Значение измеряемой величины, вещественное число размером 4 байта без метки времени
149	TMC	int4 или uint4	[-2147483648 , 2147483647] или [0 , 4294967295]	Интегральные суммы, целое число размером 4 байта без метки времени
153	TR1R	int1 или uint1	[-128, 127] или [0, 255]	Ответ на команду уставки размером 1 байт (TR1). Формат соответствует типу 6
154	TR2R	int2 или uint2	[-32768 , 32767] или [0 , 65535]	Ответ на команду уставки размером 2 байта (TR2). Формат соответствует типу 7
155	TCR	bool	[true; false]	Ответ на команду управления размером 1 байт (TC). Формат соответствует типу 1
156	TRF4R	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Ответ на команду уставки размером 4 байта (TRF4)
158	STR	string		Текстовая строка
159	UF	string		Неформатные данные. Массив из 32 байт произвольной информации

176	TS_Status	bool	[true; false]	Состояние контролируемого объекта с качеством без метки времени. Размер 1 байт. Каждый бит байта отвечает за независимый объект информации. Адрес для всех объектов один, но дополнительно в конфигурации задаётся номер бита для каждого объекта
178	TM1_Status	int1 или uint1	[-128, 127] или [0, 255]	Значение измеряемой величины, целое число размером 1 байт с качеством без метки времени
179	TM2_Status	int2 или uint2	[-32768, 32767] или [0, 65535]	Значение измеряемой величины, целое число размером 2 байта с качеством без метки времени
180	TMF4_Status	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Значение измеряемой величины, вещественное число размером 4 байта с качеством без метки времени

181	TMC_Status	int4 или uint4	[-2147483648 , 2147483647] или [0 , 4294967295]	, Интегральные суммы, целое число размером 4 байта с качеством без метки времени
185	TR1R_Status	int1 или uint1	[-128, 127] или [0 , 255]	Ответ на команду уставки размером 1 байт с качеством (TR1_Status)
186	TR2R_Status	int2 или uint2	[-32768 , 32767] или [0 , 65535]	Ответ на команду уставки размером 2 байт с качеством (TR2_Status)
187	TCR_Status	bool	[true; false]	Ответ на команду управления размером 1 байт с качеством (TC_Status)
188	TRF4R_Status	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Ответ на команду уставки размером 4 байта с качеством (TRF4_Status)
190	STR_Status	string		Текстовая строка с качеством
191	UF_Status	string		Неформатные данные с качеством
208	TS_Time	bool	[true; false]	Состояние контролируемого объекта с меткой времени. Размер 1 байт. Каждый бит байта отвечает за независимый объект информации. Адрес

				для всех объектов один, но дополнительно в конфигурации задаётся номер бита для каждого объекта
210	TM1_Time	int1 или uint1	[-128, 127] или [0, 255]	Значение измеряемой величины, целое число размером 1 байт с меткой времени
211	TM2_Time	int2 или uint2	[-32768, 32767] или [0, 65535]	Значение измеряемой величины, целое число размером 2 байта с меткой времени
212	TMF4_Time	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Значение измеряемой величины, вещественное число размером 4 байта с меткой времени
213	TMC_Time	int4 или uint4	[-2147483648, 2147483647] или [0, 4294967295]	Интегральные суммы, целое число размером 4 байта с меткой времени
217	TR1R_Time	int1 или uint1	[-128, 127] или [0, 255]	Ответ на команду уставки размером 1 байт с меткой времени (TR1_Time)
218	TR2R_Time	int2 или uint2	[-32768, 32767] или [0, 65535]	Ответ на команду уставки размером 2

				байта с меткой времени (TR2_Time)
219	TCR_Time	bool	[true; false]	Ответ на команду управления размером 1 байт с меткой времени (TC_Time)
220	TRF4R_Time	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Ответ на команду установки размером 4 байта с меткой времени (TRF4_Time)
222	STR_Time	string		Текстовая строка с меткой времени
223	UF_Time	string		Неформатные данные с меткой времени
230	M_IT_ND_1	double	$[\pm 5.0 \times 10^{-324}; \pm 1.7 \times 10^{308}]$ . Точность 15-17 цифр	Значение измеряемой величины, вещественное число двойной точности без метки времени
231	M_IT_TD_1	double	$[\pm 5.0 \times 10^{-324}; \pm 1.7 \times 10^{308}]$ . Точность 15-17 цифр	Значение измеряемой величины, вещественное число двойной точности с меткой времени
232	M_ME_NO_1	int8	$[-9.2 \times 10^{18}; 9.2 \times 10^{18}]$	Значение измеряемой величины, целое число размером 8 байт со знаком без метки времени

233	M_ME_TO_1	int8	$[-9.2 \times 10^{18}; 9.2 \times 10^{18}]$	Значение измеряемой величины, целое число размером 8 байт со знаком с меткой времени
234	M_ME_NX_1	uint8	$[0; 18.4 \times 10^{18}]$	Значение измеряемой величины, целое число размером 8 байт без знака без метки времени
235	M_ME_TX_1	uint8	$[0; 18.4 \times 10^{18}]$	Значение измеряемой величины, целое число размером 8 байт без знака с меткой времени
240	TS_Time_Status	bool	[true; false]	Состояние контролируемого объекта с меткой времени и качеством. Размер 1 байт. Каждый бит байта отвечает за независимый объект информации. Адрес для всех объектов один, но дополнительно в конфигурации задаётся номер бита для каждого объекта

242	TM1_Time_Status	int1 или uint1	[-128, 127] или [0, 255]	Значение измеряемой величины, целое число размером 1 байт с меткой времени и качеством
243	TM2_Time_Status	int2 или uint2	[-32768, 32767] или [0, 65535]	Значение измеряемой величины, целое число размером 2 байта с меткой времени и качеством
244	TMF4_Time_Status	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Значение измеряемой величины, вещественное число размером 4 байта с меткой времени и качеством
245	TMC_Time_Status	int4 или uint4	[-2147483648, 2147483647] или [0, 4294967295]	Интегральные суммы, целое число размером 4 байта с меткой времени и качеством
249	TR1R_Time_Status	int1 или uint1	[-128, 127] или [0, 255]	Ответ на команду уставки размером 1 байт с качеством и меткой времени (TR1_Time_Status)
250	TR2R_Time_Status	int2 или uint2	[-32768, 32767] или [0, 65535]	Ответ на команду уставки размером 2 байт с качеством и меткой времени (TR2_Time_Status)



251	TCR_Time_Status	bool	[true; false]	Ответ на команду управления размером 1 байт с качеством и меткой времени (TC_Time_Status)
252	TRF4R_Time_Status	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Ответ на команду уставки размером 4 байта с качеством и меткой времени (TRF4_Time_Status)
254	STR_Time_Status	string		Текстовая строка с меткой времени и качеством
255	UF_Time_Status	string		Неформатные данные с меткой времени и качеством

## Информация о процессе в направлении управления (Master → Slave)

ID	Символьный ID	Тип сервера	Диапазон значений	Описание
136	U-CTRL	Любой	Соответствующий типу	Унифицированное управление
145	TC	bool	[true; false]	Команда управления. Размер 1 байт. Каждый бит байта отвечает за независимый объект информации.

				Адрес для всех объектов один, но дополнительно в конфигурации задаётся номер бита для каждого объекта
150	TR1	int1 или uint1	[-128, 127] или [0, 255]	Команда уставки, целое число размером 1 байт без метки времени
151	TR2	int2 или uint2	[-32768, 32767] или [0, 65535]	Команда уставки, целое число размером 2 байта без метки времени
152	TRF4	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Команда уставки, вещественное число размером 4 байта без метки времени
158	STR-COMMAND	string		Текстовая строка-команда
159	UF-COMMAND	string		Неформатные данные - команда. Массив из 32 байт произвольной информации
177	TC_Status	bool	[true; false]	Команда управления размером 1 байт с качеством. Каждый бит байта отвечает за

				независимый объект информации. Адрес для всех объектов один, но дополнительно в конфигурации задаётся номер бита для каждого объекта
182	TR1_Status	int1 или uint1	[-128, 127] или [0, 255]	Команда уставки, целое число размером 1 байт с качеством
183	TR2_Status	int2 или uint2	[-32768, 32767] или [0, 65535]	Команда уставки, целое число размером 2 байта с качеством
184	TRF4_Status	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Команда уставки, вещественное число размером 4 байта с качеством
190	STR-COMMAND_Status	string		Текстовая строка - команда с качеством
191	UF-COMMAND_Status	string		Неформатные данные - команда с качеством
209	TC_Time	bool	[true; false]	Команда управления размером 1 байт с меткой времени. Каждый бит байта отвечает за

				независимый объект информации. Адрес для всех объектов один, но дополнительно в конфигурации задаётся номер бита для каждого объекта
214	TR1_Time	int1 или uint1	[-128, 127] или [0, 255]	Команда уставки, целое число размером 1 байт с меткой времени
215	TR2_Time	int2 или uint2	[-32768, 32767] или [0, 65535]	Команда уставки, целое число размером 2 байта с меткой времени
216	TRF4_Time	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Команда уставки, вещественное число размером 4 байта с меткой времени
222	STR-COMMAND_Time	string		Текстовая строка - команда с меткой времени
223	UF-COMMAND_Time	string		Неформатные данные - команда с меткой времени
241	TC_Time_Status	bool	[true; false]	Команда управления размером 1 байт с меткой времени и качеством.

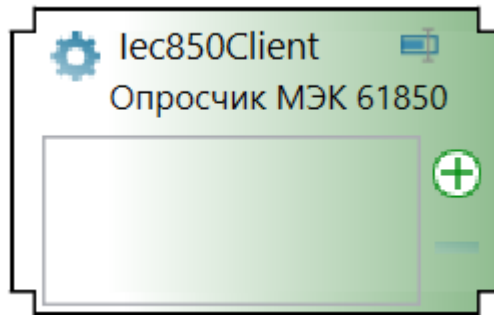
				Каждый бит байта отвечает за независимый объект информации. Адрес для всех объектов один, но дополнительно в конфигурации задаётся номер бита для каждого объекта
246	TR1_Time_Status	int1 или uint1	[-128, 127] или [0, 255]	Команда уставки, целое число размером 1 байт с меткой времени и качеством
247	TR2_Time_Status	int2 или uint2	[-32768, 32767] или [0, 65535]	Команда уставки, целое число размером 2 байта с меткой времени и качеством
248	TRF4_Time_Status	float	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Команда уставки, целое число размером 4 байта с меткой времени и качеством
254	STR-COMMAND_Time_Status	string		Текстовая строка - команда с меткой времени и качеством
255	UF-COMMAND_Time_Status	string		Неформатные данные с меткой

					времени качеством	и
--	--	--	--	--	----------------------	---

## 1.1.2.7. МЭК-61850

Модуль	Описание
<a href="#">Опросчик</a> <a href="#">МЭК 61850</a>	Коммуникационный модуль для опроса станций по протоколу ГОСТ Р МЭК 61850

## 1.1.2.7.1. Опросчик МЭК 61850



Опросчик МЭК 61850 — коммуникационный модуль Astra.Server, предназначенный для обмена данными между сервером и интеллектуальными электронными устройствами (IED) по стандарту МЭК 61850.

Функции опросчика МЭК 61850:

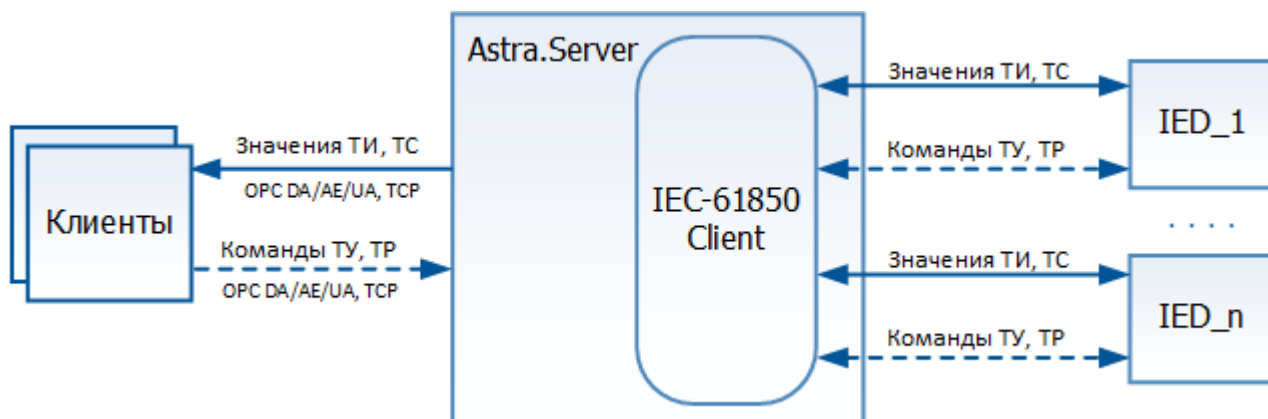
- › сбор данных с устройств;
- › подача команд управления;
- › получение файлов осциллограмм.

Опросчик МЭК 61850 собирает данные с устройств и записывает в Astra.Server. Сервер предоставляет полученные данные клиентам по различным протоколам и спецификациям.

Выдача управляющих воздействий происходит в обратной последовательности: клиенты передают команды ТУ и ТР в Astra.Server по различным протоколам и спецификациям, сервер передаёт полученные команды опросчику МЭК 61850, а модуль передаёт их соответствующим устройствам. После отправки команды модулем устройство опрашивает ответный сигнал с результатом подачи команды управления.

Взаимодействие модуля одновременно с несколькими устройствами производится параллельно.





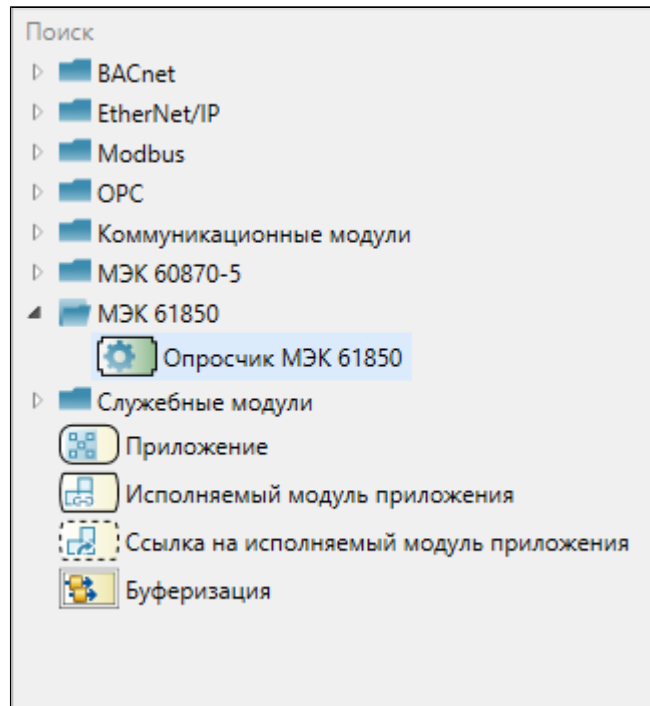
## Работа в резерве

В зависимости от настроек модуль в режиме РЕЗЕРВ:


- устанавливает соединения с устройствами, но не проводит опрос и получение данных;
- закрывает соединения с устройствами.

# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора [Astra.AStudio](#).



Выделите модуль, чтобы настроить его свойства.

 Iec850Client Опросчик МЭК 61850	
<b>↑ Параметры модуля</b>	
Настройки по умолчанию	
Активность	Да
Отображаемое имя	
<b>↑ Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>↑ Общие</b>	
Имя	Iec850Client

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Настройки по умолчанию	Выбор файла с готовой конфигурацией
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен;</li> <li>&gt; Нет – модуль остановлен.</li> </ul> Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.7.1.2. Обмен данными с устройством

### Получение данных

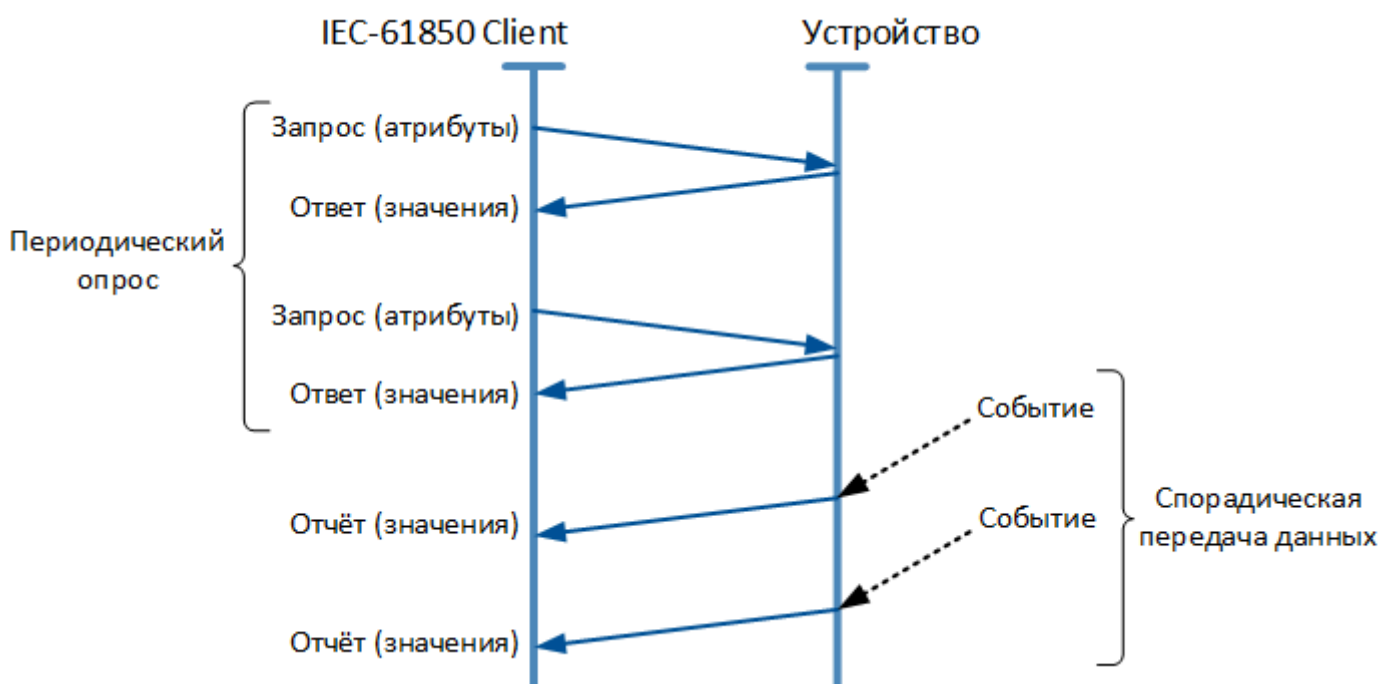
Модуль Опросчик МЭК 61850 может получать данные от устройств следующими способами:

- › [периодический опрос \(поллинг\)](#);
- › [спорадическая передача данных \(отчёты\)](#).

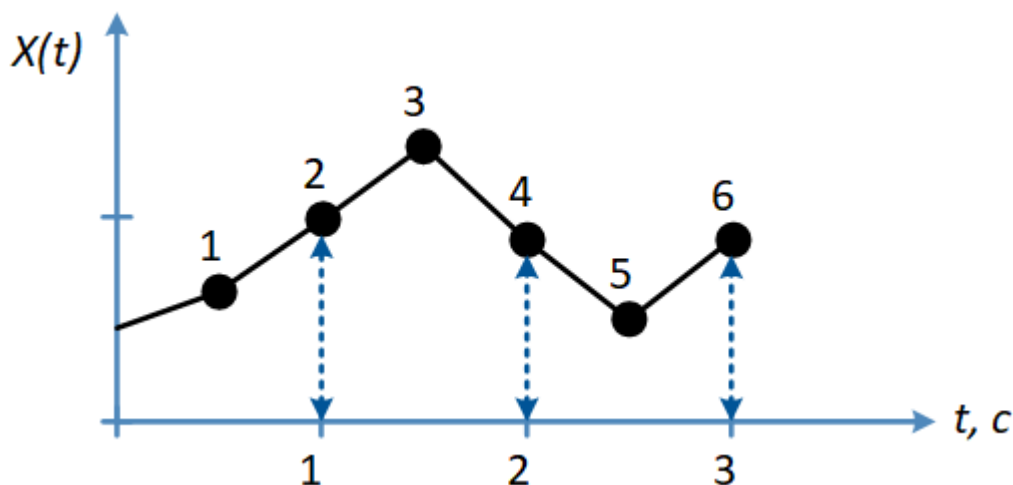
При периодическом опросе инициатором получения данных выступает модуль Опросчик МЭК 61850, который посылает устройству запросы со списком атрибутов и получает ответы, содержащие значения атрибутов.

При спорадической передаче данных по наступлению определенного события устройство самостоятельно формирует отчёт и передаёт его модулю не ожидая очередного опроса.

На рисунке ниже представлен механизм обмена данными между модулем и устройством.



Получение данных отчётами имеет преимущество в случае, если необходимо получать все значения контролируемого параметра, который быстро изменяется во времени. При получении данных периодическим опросом часть значений может быть потеряна. Например, на рисунке приведен график быстро изменяющегося во времени контролируемого параметра  $X(t)$ , значения которого модуль Опросчик МЭК 61850 получает от устройства:



- при периодическом опросе устройства с периодом опроса 1 секунда модуль получит значения параметра, соответствующие точкам 2, 4 и 6;
- при получении данных отчётами модуль получит все значения параметра: точки 1...6.

Способ получения данных отчётами позволяет отказаться от регулярного опроса устройств, что сокращает нагрузку на информационную сеть, процессор устройства и серверного компьютера, на котором функционирует Astra.Server. Также обеспечивается быстрая доставка уведомлений о возникающих событиях на стороне устройства.



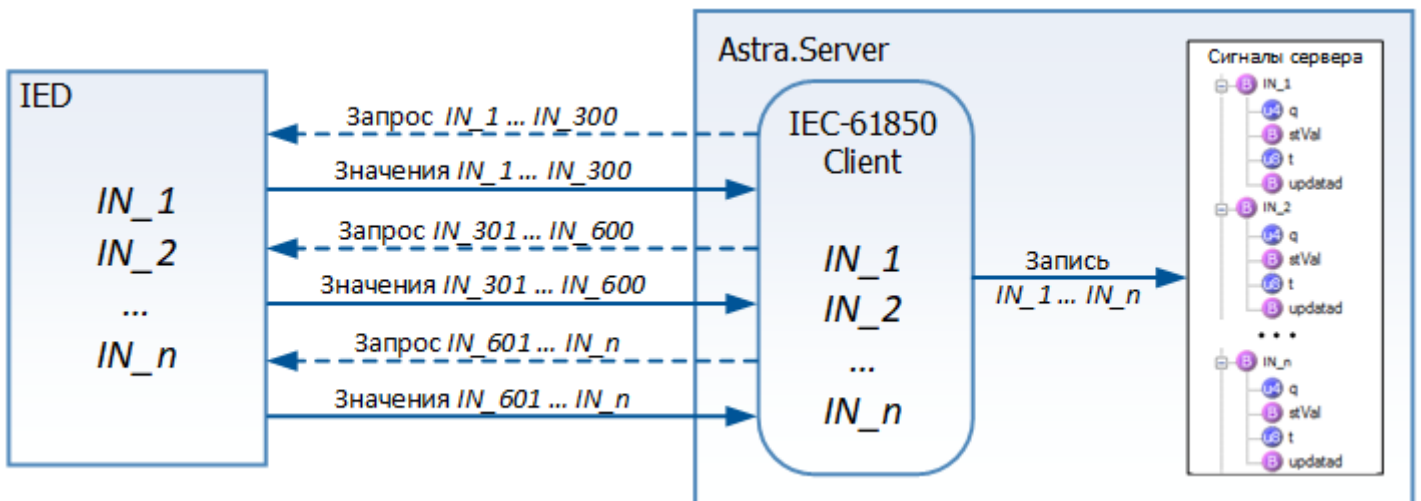
Получение данных через отчёты возможно только от устройств, поддерживающих функцию предоставления данных отчётами.

Способ получения данных модулем Опросчик МЭК 61850 от конкретного устройства определяет пользователь в процессе настройки обмена данными с устройством.

## Периодический опрос

При [периодическом опросе](#) получение данных модулем Опросчик МЭК 61850 от устройства осуществляется в следующем порядке:

1. Модуль отправляет устройству запрос, содержащий список атрибутов, значения которых необходимо получить. Максимальное число атрибутов в одном запросе — 300.
2. От устройства модуль получает значения атрибутов и отправляет следующий запрос. Период ожидания ответа от устройства — 1 секунда. Если за данный промежуток времени ответ от устройства не получен, запрашиваемым атрибутам устанавливается качество 24 (COMM\_FAILURE) и модуль переходит к следующему запросу.
3. Получив значения всех необходимых атрибутов от устройства, модуль записывает их в соответствующие сигналы Astra.Server.
4. После записи в сервер всех значений атрибутов объекта данных для каждого объекта формируется [сигнал окончания обновления данных](#) по объекту.



## Спорадическая передача данных

### Виды отчётов

Модуль Опросчик МЭК 61850 может получать данные от устройств в виде буферизованных (BRCB) и небуферизованных (URCB) отчётов.

При наличии связи между устройством и модулем передача данных осуществляется незамедлительно по факту возникновения определенных событий на стороне устройства, как при использовании буферизованных, так и при использовании небуферизованных отчетов.

Основным отличием буферизованного отчёта от небуферизованного является то, что при генерации буферизованного отчёта информация будет получена модулем даже если на момент готовности выдачи отчёта связь между модулем и устройством будет отсутствовать. Вся формируемая информация накапливается в памяти устройства и при восстановлении связи передаётся модулю. Ограничением является объем памяти для хранения отчётов в устройстве.

## **Блоки управления отчётами**

Передача отчётов от устройства модулю управляется блоками управления отчётами (RCB). Каждому RCB сопоставляется свой набор данных (DataSet), параметры которого определяются в устройстве в настройках RCB:

- › DataSet — имя набора данных;
- › RptID — полное имя отчёта.

## **Наборы данных**

Наборы данных содержат перечни объектов и атрибутов, значения которых передаются модулю. Существует два вида наборов данных:

- › набор данных устройства;
- › динамический набор данных.

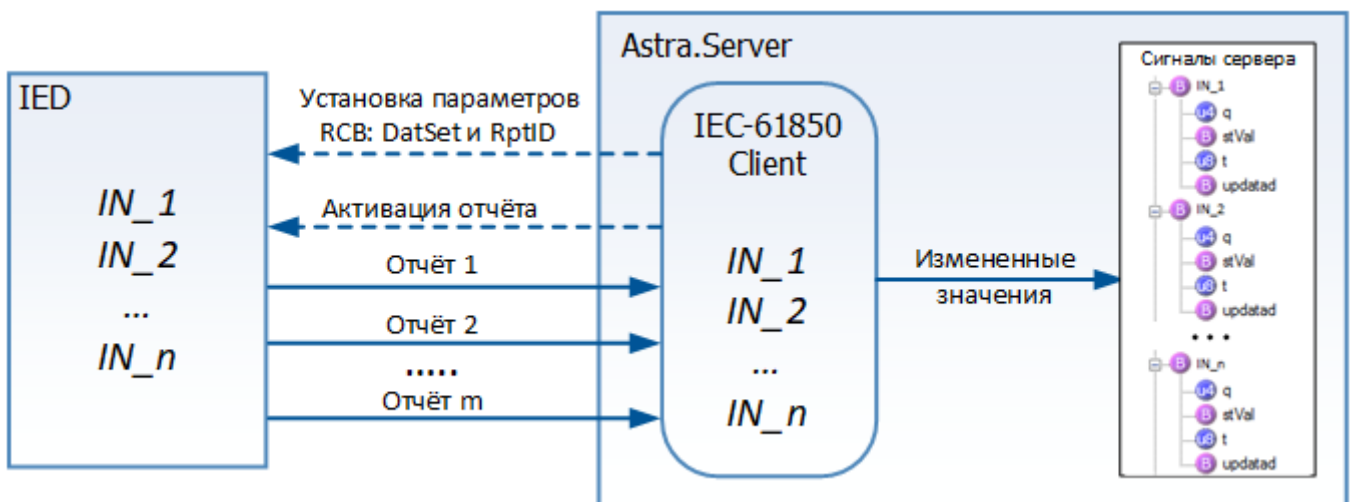
Состав набора данных устройства определен в конфигурации устройства и содержит полный перечень контролируемых объектов и атрибутов. Набор данных устройства не изменяется модулем в процессе работы.

Состав динамического набора данных определяет пользователь, формируя в устройстве соответствующий набор с перечнем объектов и атрибутов. Динамический набор данных создаётся на период работы модуля.

## Порядок получения данных

Получение данных модулем Опросчик МЭК 61850 от устройств в виде отчётов осуществляется в следующем порядке:

1. Модуль устанавливает в устройство значения параметров RCB:
  - для динамического набора данных параметры DataSet и RptID;
  - для набора данных устройства параметр RptID.
2. Производится активация отчёта, при этом, в зависимости от вида используемого набора данных, выполняются следующие действия:
  - 2.1. Для динамического набора данных:
    - 2.1.1. Существующий динамический набор данных удаляется из модуля;
    - 2.1.2. Создаётся новый динамический набор данных, указанный в настройках модуля.
  - 2.2. Для набора данных устройства:
    - 2.2.1. Модуль получает информацию о составе набора данных;
    - 2.2.2. Ассоциирует сигналы сервера с атрибутами набора данных.
3. Активизируется используемый RCB с указанным набором данных.
4. Модуль получает сформированные отчёты от устройства.
5. Каждый полученный отчёт разбирается модулем и измененные значения атрибутов записываются в сигналы сервера.





Отчёты принимаются модулем до тех пор, пока активен канал связи. При потере связи модуль переключается на другой доступный канал и заново выполняет процедуры установки параметров RCB и активации отчёта, после чего производит получение отчётов.

При резервном переходе:

- модуль сервера, находившегося в режиме РАБОТА, производит деактивацию RCB;
- модуль сервера, находившегося в режиме РЕЗЕРВ, производит активацию RCB.

Если активация RCB не произошла, модуль производит повторные попытки активации RCB с периодичностью 1 секунда.

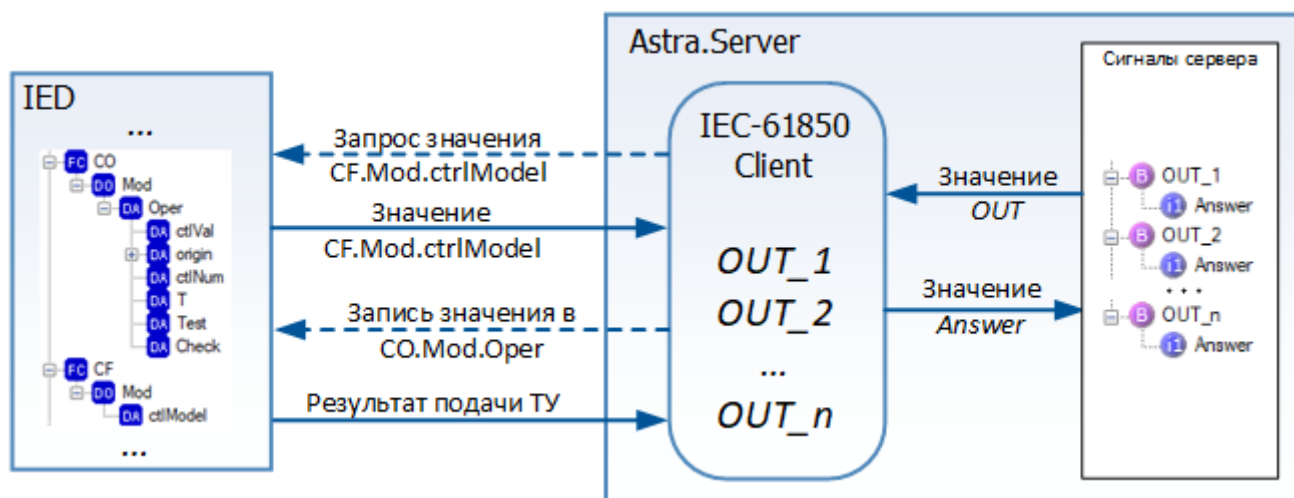
Если в настройках устройства установлен флаг Сбрасывать владение RCB путем переподключения, то при переходе из состояния РАБОТА в состояние РЕЗЕРВ модуль переподключается к устройству.

## Подача команд управления

Подача команд управления модулем Опросчик МЭК 61850 осуществляется в следующем порядке:

1. При изменении значения сигнала управления в Astra.Server модуль считывает из конфигурации устройства значение атрибута CF.Mod.ctrlModel, которое соответствует используемому режиму управления объектом данных. Могут использоваться следующие режимы управления:
  - прямое управление с нормальной безопасностью;
  - управление с предварительным выбором (SBO) с нормальным уровнем безопасности;
  - прямое управление с улучшенным уровнем безопасности;
  - управление с предварительным выбором (SBO) с улучшенным уровнем безопасности.
2. Считанное значение записывается в устройство в структуру объекта данных CO.Oper.

3. Устройство управляет ответный сигнал с результатом подачи команды управления, значение которого записывается в сервер.



## Получение файлов осциллограмм

Получение файлов осциллограмм модулем Опросчик МЭК 61850 осуществляется в следующем порядке:

1. С указанной в настройках периодичностью модуль получает список файлов, содержащихся в устройстве.
2. Модуль проверяет содержимое папки на диске, в которую принимаются файлы от устройства:
  - если в устройстве содержится файл, которого нет в папке, данный файл загружается с устройства;
  - если в устройстве и в папке содержится файл с одинаковым именем и размером, данный файл не загружается с устройства.

Папка, в которую модуль принимает файлы от устройства, указывается в настройках файлового обмена и создаётся модулем на диске.

Если в устройстве файлы располагаются в отдельных папках, необходимо вручную создать соответствующие папки в каталоге загрузки на диске.



Если в устройстве файлы располагаются в отдельных папках, необходимо вручную создать соответствующие папки в каталоге загрузки на диске.

## 1.1.2.7.1.3. Набор данных устройства и сигналы сервера

### Набор данных устройства

Каждое устройство имеет свой набор данных, состоящий из объектов и атрибутов. Полное имя атрибута данных в устройстве в соответствии с МЭК 61850-8-1 имеет вид:



Логическое устройство/логический узел\$Функциональная связь  
\$Объект данных\$Атрибут данных

Функциональные связи:

- › ST — информация о состоянии;
- › MX — измеряемые величины (аналоговые значения);
- › CO — управление;
- › SP — уставка;
- › DC — описание;
- › CF — конфигурация.

Пример полного имени атрибута данных устройства:



IED1LD6/ledsGGIO1\$ST\$Ind19\$stVal

Набор данных устройств описан в SCL-файлах следующих форматов:

- › icd — описание возможностей устройства;
- › cid — описание конфигурации устройства;
- › scd — описание конфигурации подстанции.

### Сигналы Astra.Server

Сигналы Astra.Server, с которыми взаимодействует модуль Опросчик МЭК 61850, по функциональному назначению делятся на три группы:

- входящие — сигналы, значения которых модуль получает от устройства;
- исходящие — сигналы, значения которых при изменении в сервере передаются от модуля в устройство;
- внутренние — сигналы, значения которых формируются на основании проведенных модулем действий с входящими или исходящими сигналами.

Кроме того, при запуске модуль создает динамические сигналы для мониторинга наличия соединений по каналам и управления активным каналом, сигналы мониторинга файлового обмена и отчетов, а также стандартные для всех модулей Astra.Server динамические сигналы, позволяющие контролировать работу модуля.

## Связь сигналов сервера с атрибутами данных устройства

Для обмена данными между модулем и устройством необходимо связать сигналы Astra.Server с соответствующими атрибутами данных устройства.

Поле строки адреса	Описание
Station	Название устройства в конфигурации модуля Опросчик МЭК 61850 сервера
Address	Полное имя атрибута данных устройства
ProtocolType	Протокольный тип, значение которого выбирается в зависимости от функционального назначения сигнала и типа атрибута данных в устройстве

### Входящие сигналы

Для входящих сигналов значение поля ProtocolType имеет формат IN\_\*, где \* — строковое представление типа атрибута данных устройства в соответствии с МЭК 61850-6 (п.п. 9.5.4.2), например:

- › IN\_BOOLEAN;
- › IN\_Quality.

Для входящих сигналов, значения которых формируются на основе обработки атрибутов объекта данных, значение поля ProtocolType имеет формат:

Протокольный тип	Описание
IN_*_BSI	<p>Классы общих данных SPS, DPS, INS (информация о состоянии), где * — строковое представление типа атрибута данных.</p> <p>Атрибуты, на основе обработки которых формируется значение сигнала:</p> <ul style="list-style-type: none"> <li>› stVal (тип BOOLEAN) — состояние (true/false);</li> <li>› t (тип TimeStamp) — метка времени;</li> <li>› q (тип Quality) — качество.</li> </ul>
IN_BOOL_ACT	<p>Класс ACT (сведения об активации защиты).</p> <p>Атрибуты, на основе обработки которых формируется значение сигнала:</p> <ul style="list-style-type: none"> <li>› general (тип BOOLEAN) — логическое ИЛИ состояния фаз (true/false);</li> <li>› t (тип TimeStamp) — метка времени;</li> <li>› q (тип Quality) — качество.</li> </ul>
IN_FLOAT32_CMV	<p>Класс CMW (комплексные измеряемые значения).</p> <p>Атрибуты, на основе обработки которых формируется значение сигнала:</p> <ul style="list-style-type: none"> <li>› \$cVal\$mag\$f (тип FLOAT32) — значение аналогового сигнала;</li> <li>› t (тип TimeStamp) — метка времени;</li> <li>› q (тип Quality) — качество.</li> </ul>

Поддерживаемые типы атрибутов данных, соответствующие типам сигналов Astra.Server:

Тип атрибута данных	Тип сигнала в Astra.Server											
	bool	int1	int2	int4	int8	uint1	uint2	uint4	uint8	float	double	string
BOOLEAN	+											
INT8		+	+	+	+							
INT16			+	+	+							
INT24				+	+							
INT32				+	+							
INT8U			+	+	+	+	+	+	+			
INT16U				+	+		+	+	+			
INT24U					+			+	+			
INT32U					+			+	+			
FLOAT32										+	+	
FLOAT64											+	
Enum				+	+							
Quality					+			+	+			
Timestamp									+			
VisString32												+
VisString255												+
Unicode255												+
Dbpos			+	+	+	+	+	+	+			
EnumStr												+

Для типа EnumStr (string) значение поля ProtocolType имеет формат IN\_EnumStr\_\*, где \* — строковое представление типа перечисления,

описанного в SCL-файле (перечисления SCL согласно МЭК 61850-7-3 и МЭК 61850-7-4 приведены в приложении В ГОСТ Р МЭК 61850-6).

Если для значения, полученного от устройства, невозможно подобрать соответствующее строковое значение при конвертации в EnumStr, то значение сигнала принимается равное целому значению. Подробная информация отображается в журнале приложений.

## Исходящие сигналы

Для исходящих сигналов значение поля Address должно быть сопоставлено с атрибутом Oper в устройстве для функциональных связей типов CO (управление) и SP (уставка).

Значение поля ProtocolType имеет формат OUT\_\*, где \* — строковое представление управляющего типа CDC в соответствии с МЭК 61850, например:

- OUT\_SPC;
- OUT\_INC.

Поддерживаемые управляющие типы CDC, соответствующие типам сигналов Astra.Server:

Тип CDC	Тип сигнала в Astra.Server											
	bool	int1	int2	int4	int8	uint1	uint2	uint4	uint8	float	double	string
SPC	+											
DPC	+											
INC				+								
ISC		+										

## Внутренние сигналы



Для внутренних сигналов значение поля ProtocolType имеет формат INTERNAL\_\*, где \* — тип внутреннего сигнала:

- › UpdDO — сигнал окончания обновления данных по объекту;
- › CtrlRes — результат подачи команды управления.

Соответствие внутренних сигналов типам сигналов Astra.Server:

Тип	Тип сигнала в Astra.Server											
	bool	int1	int2	int4	int8	uint1	uint2	uint4	uint8	float	double	string
UpdDO	+											
CtrlRes		+	+	+	+							

## Сигнал окончания обновления данных по объекту

Данный тип сигнала предназначен для написания формул, формирующих итоговое значение на основании значений атрибутов данных, принадлежащих одному объекту. Сигналу устанавливается значение true, если все значения атрибутов объекта данных записаны в сигналы сервера. При считывании значений сигналов из сервера в формулах присутствуют значения, полученные в один момент времени.

## Результат подачи команды управления

Сигнал формируется на основании операций, проведенных с управляющим значением.

Принимаемые значения:

Значение	Состояние
1	Данные приняты к исполнению, поставлены в исходящую очередь
3	Команда исполнена

-1	Данные не приняты к исполнению (плохое качество, неверный режим работы модуля)
-2	Нет связи с устройством
-3	Ошибка получения режима подачи управления (не прочитан соответствующий атрибут CF.Mod.ctrlModel)
-4	Ошибка получения спецификации команды управления (не прочитана структура CO.Mod.Oper)
-5	Неподдерживаемый модулем режим управления
-6	Протокольная ошибка подачи управления
-7	Ошибка создания значения управления
-8	Отрицательный предварительный выбор значения

## Качество сигналов

Значения качества сигналов, выставляемых модулем Опросчик МЭК 61850, приведены в таблице:

Значение качества	Идентификатор качества	
4	CONFIG_ERROR	Сигнал неправильно ск
8	NOT_CONNECTED	Сигнал принят на обслу инициализировано
12	DEVICE_FAILURE	Ошибка инициализаци
24	COMM_FAILURE	КП нет на связи
28	OUT_OF_SERVICE	Модуль не запущен
64	UNCERTAIN	Связь установлена, но з
96	CONVERT_ERROR	Полученное от источни типу сигнала сервера
192	GOOD	Значение сигнала дост

Для сигналов типа Timestamp, байт качества метки времени добавляется к качеству сигнала в старший байт младшего слова. Например, результирующее качество сигнала метки времени имеет значение 0x000001C0, это означает, что качество метки времени 0x01, а качество сигнала 0xC0 — GOOD.

## 1.1.2.7.1.4. Типы данных

Модуль Опросчик МЭК 61850 поддерживает следующие типы атрибутов данных устройства согласно IEC-61850-7-2:

Тип сигнала в устройстве	Диапазон	Описание	Тип сигнала в сервере
BOOLEAN	0 – false, 1 – true	Логический тип	bool
INT8	от -128 до 127	Целое знаковое 1 байт	int1
INT16	от -32 768 до 32 767	Целое знаковое 2 байта	int2
INT24	от -8 388 608 до 8 388 607	Целое знаковое 3 байта	int4
INT32	от -2 147 483 648 до 2 147 483 647	Целое знаковое 4 байта	int4
INT8U	от 0 до 255	Целое беззнаковое 1 байт	uint1
INT16U	от 0 до 65 535	Целое беззнаковое 2 байта	uint2
INT24U	от 0 до 16 777 215	Целое беззнаковое 3 байта	uint4
INT32U	от 0 до 4 294 967 295	Целое беззнаковое 4 байта	uint4
FLOAT32	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр	Вещественное 4 байта	float
FLOAT64	$[\pm 5.0 \times 10^{-324}; \pm 1.7 \times 10^{308}]$ . Точность 15-17 цифр	Вещественное 8 байт	double
OCTET STRING		Строковый тип	string

VISIBLE STRING		Строковый тип	string
UNICODE STRING		Строковый тип	string

# Диагностика работы модуля

## Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

» в Linux системах в директории:



/opt/AstraRegul/Astra.Server/Logs.

Для просмотра журнала работы модуля используется сервисное приложение Просмотрщик лога кадров.

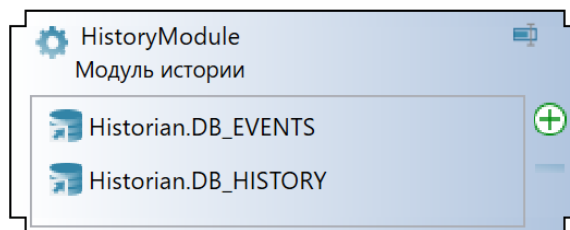
№	Дата	Время	Описание
14624	18.06.2021	17:53:42:283	ENIP_P1.Канал 2: Невозможно установить соединение '20.3.10.123:102'.
14625	18.06.2021	17:53:42:285	KP4.Канал 2: Невозможно установить соединение '20.3.10.116:102'.
14626	18.06.2021	17:53:42:285	PA2.Канал 2: Невозможно установить соединение '20.3.10.106:102'.
14627	18.06.2021	17:53:42:285	TS2.Канал 2: Невозможно установить соединение '20.3.10.113:102'.
14628	18.06.2021	17:53:42:286	VV1.Канал 1: Невозможно установить соединение '20.3.9.104:102'.
14629	18.06.2021	17:53:42:286	A02.Канал 1: Невозможно установить соединение '20.3.9.157:102'.
14630	18.06.2021	17:53:42:286	TN1.Канал 2: Невозможно установить соединение '20.3.10.109:102'.
14631	18.06.2021	17:53:42:286	TO2.Канал 2: Невозможно установить соединение '20.3.10.114:102'.
14632	18.06.2021	17:53:42:286	KP3.Канал 1: Невозможно установить соединение '20.3.9.101:102'.
14633	18.06.2021	17:53:42:287	P1.Канал 1: Невозможно установить соединение '20.3.9.98:102'.
14634	18.06.2021	17:53:42:287	ENIP_TS2.Канал 2: Невозможно установить соединение '20.3.10.136:102'.

Количество записей: 15249

## 1.1.2.8. Служебные модули

Модуль	Описание
<a href="#">Модуль истории</a>	Модуль ведения истории изменений сигналов
<a href="#">Модуль SnapShot</a>	Модуль для генерации и восстановления файлов-срезов (XML)
<a href="#">Модуль ESU TU</a>	Модуль ESU TU (Элементы сервера)
<a href="#">Модуль Data Buffer</a>	Модуль разбора буфера данных
<a href="#">Модуль Write-VQT</a>	Модуль изменения основных свойств сигналов
<a href="#">Клиент безопасности</a>	Модуль Клиент безопасности (Элементы сервера)
<a href="#">Контроллер кластера безопасности</a>	Модуль реализации контроллера кластера безопасности
<a href="#">Модуль вычислений</a>	Модуль для проведения расчетов
<a href="#">Модуль резервирования</a>	Модуль реализации резервирования
<a href="#">Модуль Process Monitor</a>	Модуль мониторинга состояния процессов

## 1.1.2.8.1. Модуль истории



Модуль истории предназначен для сохранения данных в сервер истории.

### Функции:

- › сохранение значений в базы данных (Astra.Historian, MS SQL Server, PostgreSQL).
- › сохранение событий в базы данных Astra.Historian;
- › запрос сохранённых данных.

### Сохранение данных

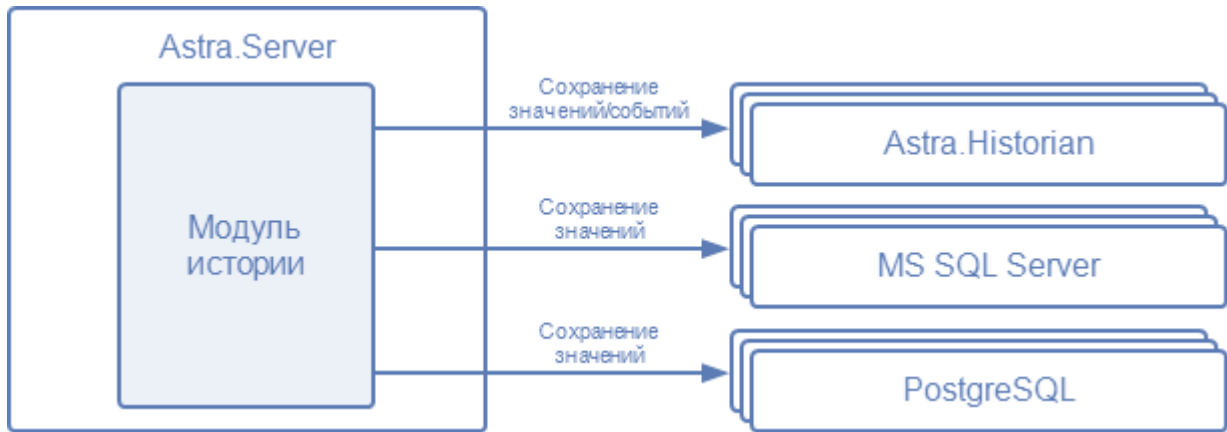
Модуль истории сохраняет:

- › Значения сигналов, для которых настроено сохранение в историю – сохраняются в момент изменения значения;
- › Все генерируемые события – сохраняются сразу после генерации.

Для резервирования истории данные сохраняются параллельно во все базы данных, настроенные в модуле.

При отсутствии связи с базой данных несохранённые данные копятся в очереди данных на стороне Astra.Server до завершения транзакции записи.





## Запрос сохранённых данных

Запрос сохранённых данных выполняется в следующем порядке:

1. Клиент запрашивает историю событий/значений у модуля, предоставляющего исторические данные:

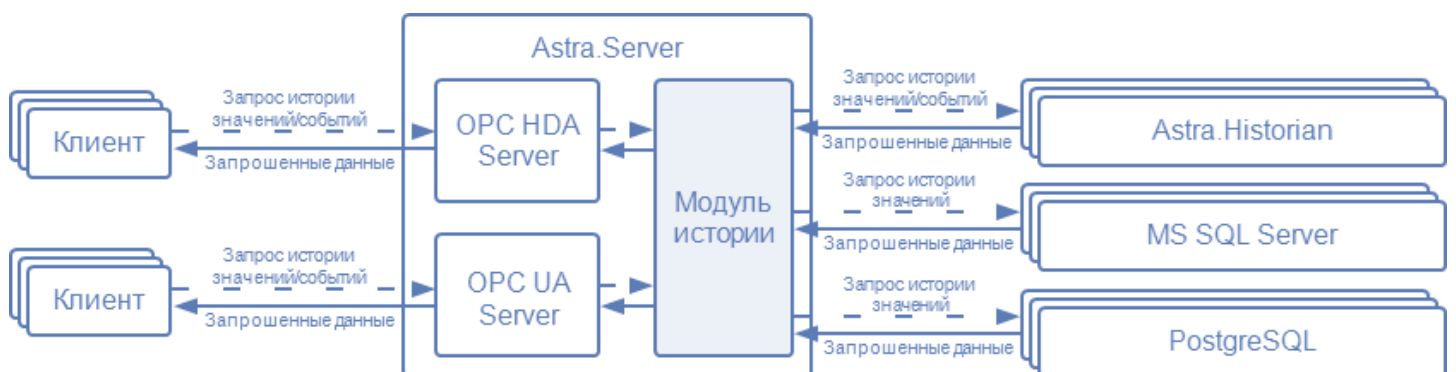
- OPC HDA Server
- OPC UA Server

2. Запрос передаётся модулю истории.

3. Модуль истории запрашивает данные из той базы данных, с которой быстрее всего установлено соединение.

При потере связи с текущей базой данных, модуль автоматически переключается на базу данных, с которой есть связь.

4. Полученные данные передаются запросившему их клиенту.



## Работа в резерве

В режиме РЕЗЕРВ модуль:

- › не сохраняет данные;
- › может запрашивать сохранённые данные.

Возможны исключительные ситуации, при которых значение сигнала обновится в резервном сервере, но не обновится в основном сервере резервной пары и не будет сохранено в историю. Чтобы значение было сохранено в историю, резервный сервер для каждого настроенного сигнала хранит последнее значение, которое должно быть сохранено в историю: это значение будет сохранено в историю при переходе сервера в режим РАБОТА. Данная функция включается в параметрах модуля.

Ситуации, при которых значение сигнала не обновляется в основном сервере резервной пары:

- › основной сервер не имеет связи с контроллером, посылающим значения сигнала;
- › сигнал был добавлен в конфигурацию резервного сервера – основной сервер не будет получать значения этого сигнала до обновления конфигурации.

## Работа в Astra.AccessPoint

В составе Astra.AccessPoint модуль истории только запрашивает исторические данные, сохранённые источниками данных. Сохранение данных в Astra.AccessPoint недоступно.

Запрос исторических данных в Astra.AccessPoint осуществляется так же, как в Astra.Server. Параметры подключения к базам данных модуль истории получает от модулей истории в составе источников данных при подключении к ним.

## 1.1.2.8.1.1. Настройка

Настройка модуля включает в себя следующие шаги:

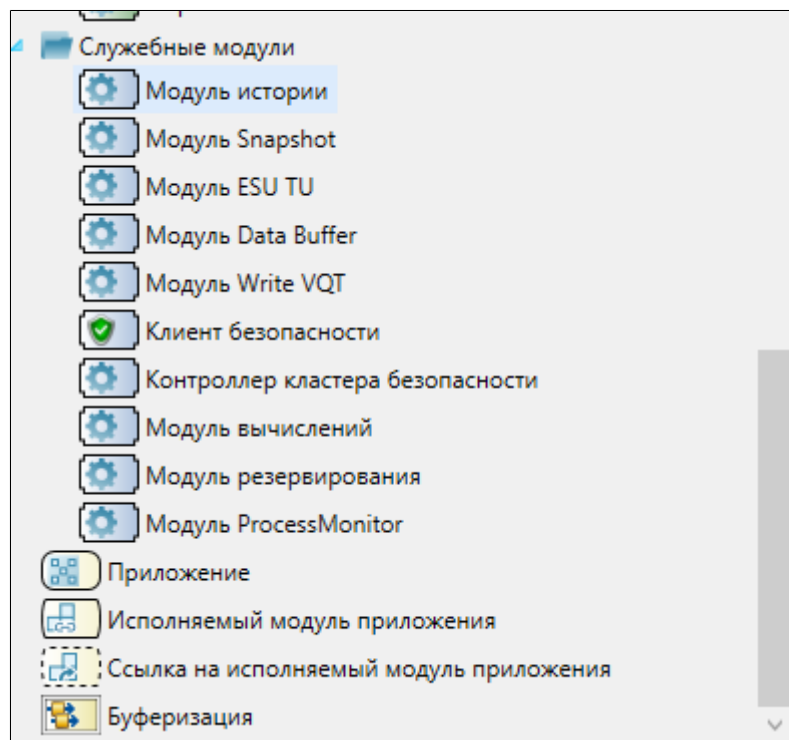
1. Конфигурирование параметров модуля.
2. Конфигурирование параметров подключения к базам данных:
  - БД Astra.Historian
  - БД MS SQL Server
  - БД PostgreSQL
3. Конфигурирование сигналов



В Astra.AccessPoint нужно только добавить модуль и активировать его.

Параметры подключения к базам данных модуль получит от модулей истории в составе источников данных.

Для добавления модуля в состав конфигурации сервера воспользуйтесь панелью элементов в программе AStudio.



Выделите модуль, чтобы настроить его свойства.

HistoryModule Модуль истории	
<b>Параметры модуля</b>	
Размер кэша для данных	500000
Количество значений в одной транзакции	5000
Таймаут выполнения транзакции, мс	1000
Записывать значения при активации	Нет
Сохранять дополнительную метку времени сервера	Да
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	HistoryModule

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Размер кэша для данных	Размер очереди данных, измеряется в количестве значений. Значение параметра должно находиться в диапазоне от 10000 до 100 000 000
Количество значений в одной транзакции	Ограничение на количество значений в одной транзакции, передаваемых серверам MS SQL Server. При достижении указанного количества выполнится транзакция записи.

	Значение параметра должно находиться в диапазоне от 1000 до 10000
Тайм-аут наполнения транзакции, мс.	Максимальное время на накопление данных для выполнения транзакции записи в серверы MS SQL Server. По истечении указанного времени с предыдущей транзакции, выполнится транзакции записи. Значение параметра должно находиться в диапазоне от 500 до 5000
Записывать значения при активации	Если выбрано Да, то при работе сервера в режиме РЕЗЕРВ для каждого настроенного сигнала будет храниться последнее неотфильтрованное значение; эти значения будут записаны в историю при переходе сервера в режим РАБОТА
Активность	Активность модуля: <ul style="list-style-type: none"> <li>› Да – модуль запущен</li> <li>› Нет – модуль остановлен</li> </ul>
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

# Диагностика работы модуля

## Служебные сигналы

Модуль истории динамически создаёт служебные сигналы:

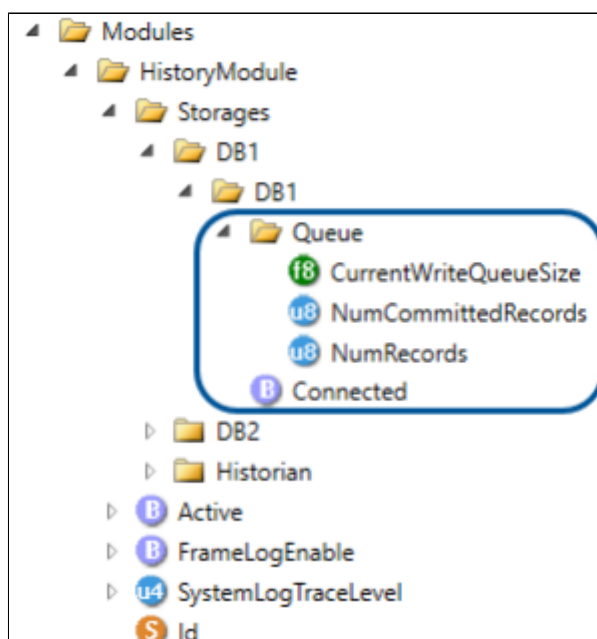
- › контроля связи с Astra.Historian (при подключении по TCP);
- › блокировки регионов хранилища Astra.Historian от удаления (при подключении по TCP);
- › контроля связи с БД Astra.Historian (при подключении по OPC);
- › контроля и управления основными параметрами модуля.

## Контроль связи с Astra.Historian

При подключении к БД Astra.Historian по TCP модуль истории создаёт служебные сигналы контроля связи с Astra.Historian. Полный тег сигналов контроля связи с Astra.Historian имеет вид:



Service.Modules.<Имя модуля>.Storages.<Имя хранилища>.<Имя БД>.<Имя сигнала>



Сигнал	Тип	Описание сигнала
--------	-----	------------------


Connected	Bool	Состояние сессии записи в БД Astra.Historian: <ul style="list-style-type: none"> <li>&gt; «True» – есть открытая сессия</li> <li>&gt; «False» – нет открытой сессии</li> </ul>
CurrentWriteQueueSize	Float	Текущий размер данных, ожидающих сохранения на диск, кБ
NumRecords	UInt8	Число записей в очереди
NumCommittedRecords	UInt8	Число записей в очереди, сохраненных на диск

## Блокировка регионов хранилища Astra.Historian от удаления

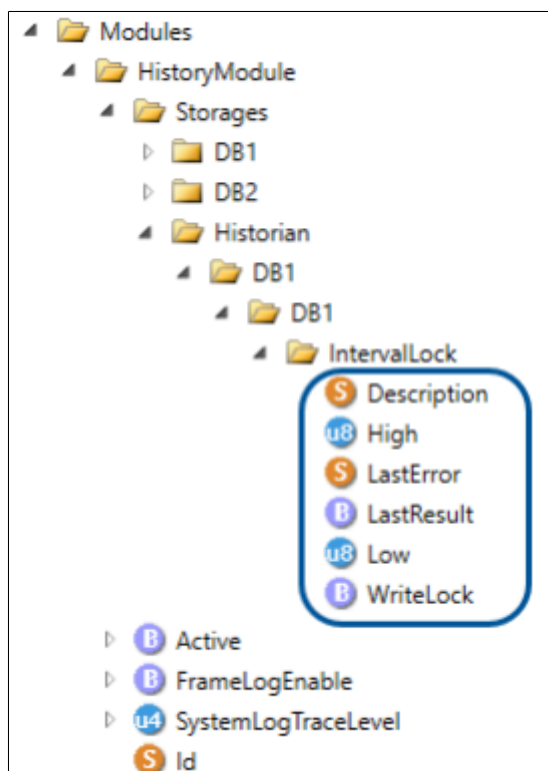
При подключении к БД Astra.Historian по TCP модуль истории позволяет сохранять блокировки временных интервалов на стороне БД Astra.Historian:

1. Модуль истории создаёт для каждой БД служебные сигналы, с помощью которых задаются свойства блокировки и подаётся команда записи.
2. По команде записи блокировки модуль истории считывает свойства блокировки из служебных сигналов и сохраняет запись о блокировке в файловую очередь.
3. Запись о блокировке из файловой очереди сохраняется в БД Astra.Historian.

Полный тег служебных сигналов имеет вид:



```
Service.Modules.<Имя хранилища>.<Имя БД>.<Имя сигнала>
                                модуля>.Storages.Historian.<Имя
```



Сигнал	Тип	Описание сигнала
Description	String	Описание создаваемой блокировки
High	UInt8	Верхняя граница блокируемого временного интервала в формате FILETIME по времени UTC
LastError	String	Описание ошибки последней операции записи, если она была неуспешной
LastResult	Bool	Результат выполнения последней команды записи блокировки: <ul style="list-style-type: none"> <li>› «True» – запись блокировки выполнена успешно</li> <li>› «False» – запись блокировки не выполнена</li> </ul>
Low	UInt8	Нижняя граница блокируемого временного интервала в формате FILETIME по времени UTC
WriteLock	Bool	Команда: <ul style="list-style-type: none"> <li>› «True» – запись блокировки</li> </ul>

Для подачи команды записи блокировки:

1. Укажите значения свойств блокировки, задав значения служебных сигналов «Description», «Low», «High».
2. Сигналу «WriteLock» задайте значение «True».



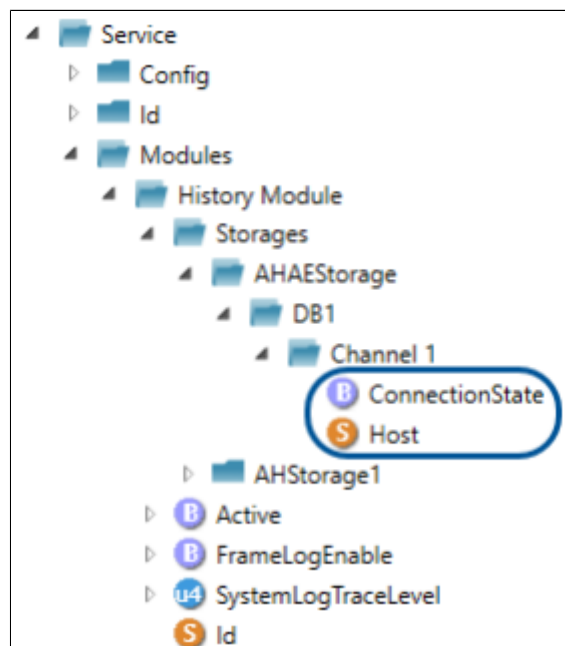
## Контроль связи с БД Astra.Historian

При подключении к БД Astra.Historian по OPC модуль истории создаёт служебные сигналы контроля связи с БД.

Полный тег сигналов контроля связи с БД имеет вид:



Service.Modules.<Имя модуля>.Storages.<Имя хранилища>.<Имя БД>.Channel <Номер канала>.<Имя сигнала>



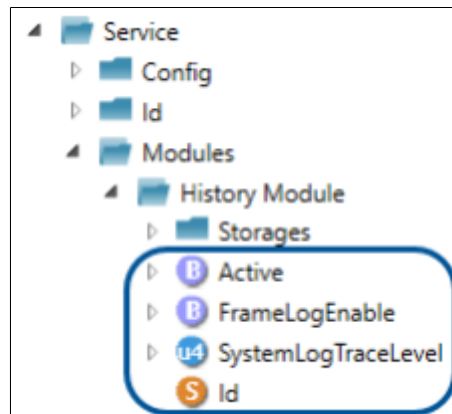
Сигнал	Тип	Описание сигнала
ConnectionState	Bool	Наличие связи по каналу: > «True» – связь по каналу установлена > «False» – связь по каналу отсутствует
Host	String	Имя хоста БД

## Контроль и управление основными параметрами модуля

Полный тег стандартных служебных сигналов контроля и управления основными параметрами модуля имеет вид:



Service.Modules.<Имя модуля>.<Имя сигнала>



Сигнал	Тип	Описание сигнала
Active	Bool	Активность модуля: > «True» – запущен > «False» – остановлен Соответствует значению параметра Активность. Управляется служебным сигналом «Active.Set»
FrameLogEnable	Bool	Ведение журнала работы: > «True» – ведётся > «False» – не ведётся Соответствует значению параметра Вести журнал работы модуля. Управляется служебным сигналом «FrameLogEnable.Set»
SystemLogTraceLevel	Uint4	Уровень детализации журнала работы: > «1» – Предупреждения и аварийные сообщения > «2» – Информационные сообщения > «3» – Отладочные сообщения Соответствует значению параметра Уровень трассировки в журнал приложений. Управляется служебным сигналом «SystemLogTraceLevel.Set»
Id	String	Идентификатор модуля в конфигурации Astra.Server

# Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.aplog по умолчанию:

» в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

» в Linux системах в директории:



/opt/AstraRegul/Astra.Server/Logs.

Для просмотра журнала работы модуля воспользуйтесь сервисным приложением Просмотрщик лога кадров.

[Astra.Server] Просмотр журналов - C:\Program Files\ProSyst\Astra.Server\Logs\History Module1.aplog (Общий журнал)

Файл Фильтр Поиск Помощь

№	Дата	Время	Описание
95	27.07.2021	22:39:10:934	Сигнал AccessPoint.HUB.String.deadband: допуск по значению н...
96	27.07.2021	22:39:10:934	Установлено соединение с БД для чтения истории: 'localhost\A...
97	27.07.2021	22:39:10:934	Сигнал AccessPoint.ТочкаДоступа.ХАБ.СтроковаяПеременная.d...
98	27.07.2021	22:39:10:934	Сигнал History.String.deadband: допуск по значению не исполь...
99	27.07.2021	22:39:13:137	Журнал закрыт
100	27.07.2021	22:39:33:309	Журнал открыт для записи
101	27.07.2021	22:39:33:324	Соединение с БД установлено: localhost\Astra.Historian.Server...
102	27.07.2021	22:39:33:340	Настройки HistoryModule f1a86500-76ba-4f60-8cd7-d507e7884a...
103	27.07.2021	22:39:33:340	Соединение с БД установлено: 127.0.0.1\Astra.Historian.Server...
104	27.07.2021	22:39:33:340	Сигнал AccessPoint.HUB.String.deadband: допуск по значению н...
105	27.07.2021	22:39:33:340	Установлено соединение с БД для чтения истории: 'localhost\A...

Количество записей: 207

Сигнал  
AccessPoint.ТочкаДоступа.ХАБ.Строко  
ваяПеременная.deadband: допуск по  
значению не используется для  
сигналов этого типа. Указанный

0 1 2 3 4 5 6 7

# Устранение неполадок

В этом разделе описаны проблемы, которые могут возникнуть во время работы модуля, представлены возможные причины их возникновения и приведены инструкции по их устранению.

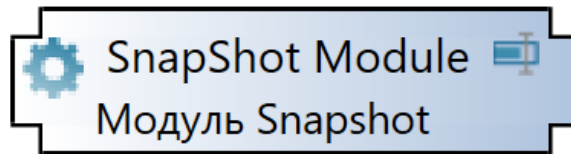
Проблема	Возможные причины и описание решения
Значения сигналов не сохраняются в историю всех БД	<p>Сообщения в журнале:</p> <ul style="list-style-type: none"><li>➤ Модуль 'History Module' перешел в состояние &lt;статус&gt;</li><li>➤ Модуль 'History Module' остановлен</li></ul> <p>Причина:</p> <ul style="list-style-type: none"><li>➤ Модуль истории имеет статус отличный от: В работе Активный либо модуль остановлен.</li></ul> <p>Решение:</p> <ul style="list-style-type: none"><li>➤ Перевести Astra.Server из режима РЕЗЕРВ в режим РАБОТА и установить значение Да параметра Активность модуля истории.</li></ul>
	<p>Сообщения в журнале:</p> <ul style="list-style-type: none"><li>➤ Не удалось установить привилегию для фиксации страниц памяти. Проверьте наличие соответствующего разрешения в локальной политике</li><li>➤ Модуль 'History Module' (HistoryModule.dll) не инициализирован</li></ul> <p>Причина:</p> <ul style="list-style-type: none"><li>➤ В локальных политиках безопасности пользователю, от имени которого запущен Astra.Server, не разрешена блокировка страниц памяти.</li></ul> <p>Решение:</p> <ul style="list-style-type: none"><li>➤ Разрешить пользователю блокировку страниц памяти</li></ul>

<p>Значения сигналов не сохраняются в историю при работе с Astra.Historian</p>	<p>Сообщения в журнале:</p> <ul style="list-style-type: none"> <li>➤ Провалилась попытка установления связи с БД: IP \\&lt;имя экземпляра сервера истории&gt;:&lt;имя БД&gt; – Интерфейс не поддерживается</li> <li>➤ Провалилась попытка установления связи с БД: IP \\&lt;имя экземпляра сервера истории&gt;:&lt;имя БД&gt; – Не найден указанный модуль</li> </ul> <p>Причина:</p> <ul style="list-style-type: none"> <li>➤ Недоступна или отсутствует библиотека astra.historian.server.comapi.dll в Astra.Server и/или Astra.Historian.</li> </ul> <p>Библиотека расположена в &lt;папка Astra.Server&gt; \Common и в &lt;папке Astra.Historian&gt;</p> <p>Решение:</p> <ul style="list-style-type: none"> <li>➤ Переустановить Astra.Server и Astra.Historian</li> </ul>
	<p>Сообщения в журнале:</p> <ul style="list-style-type: none"> <li>➤ Провалилась попытка установления связи с БД: IP \\&lt;имя экземпляра сервера истории&gt;:&lt;имя БД&gt; – Неопознанная ошибка</li> </ul> <p>Причина:</p> <ul style="list-style-type: none"> <li>➤ Идентификационные данные БД и сервера истории в конфигурации модуля истории не совпадают с соответствующими данными в конфигурационном файле сервера истории.</li> </ul> <p>Решение проблемы:</p> <ul style="list-style-type: none"> <li>➤ Сверить конфигурационные данные</li> </ul>
	<p>Сообщения в журнале:</p> <ul style="list-style-type: none"> <li>➤ Провалилась попытка установления связи с БД: IP \\&lt;имя экземпляра сервера истории&gt;:&lt;имя БД&gt; - Отказано в доступе</li> </ul> <p>Причина:</p> <ul style="list-style-type: none"> <li>➤ На компьютере с Astra.Server не настроен протокол DCOM.</li> </ul> <p>Решение проблемы:</p>

	<ul style="list-style-type: none"> <li>› Настроить DCOM для всех COM-объектов, имеющихся на компьютере с Astra.Server.</li> </ul>
<p>Значения сигналов не сохраняются в историю при работе с MS SQL Server</p>	<p>Сообщения в журнале:</p> <ul style="list-style-type: none"> <li>› Хранилище &lt;имя хранилища&gt; Не удалось установить соединение с хранилищем: SQL Server не существует, или доступ запрещен</li> </ul> <p>Причина:</p> <ul style="list-style-type: none"> <li>› Отсутствует связь с сервером (с хранилищем).</li> </ul> <p>Решение:</p> <ul style="list-style-type: none"> <li>› Проверить активность, возможность подключения и правильность настроечных параметров MS SQL Server</li> </ul>
	<p>Сообщения в журнале:</p> <ul style="list-style-type: none"> <li>› Хранилище &lt;имя хранилища (имя компьютера \имя экземпляра сервера)&gt; Не удалось установить соединение с хранилищем: Ошибка входа пользователя &lt;имя пользователя&gt;</li> </ul> <p>Причина:</p> <ul style="list-style-type: none"> <li>› Идентификационные данные пользователя в конфигурации модуля не совпадают с настройками экземпляра MS SQL Server</li> </ul> <p>Решение:</p> <ul style="list-style-type: none"> <li>› Сверить конфигурационные данные.</li> </ul>
	<p>Сообщения в журнале:</p> <ul style="list-style-type: none"> <li>› Хранилище &lt;имя хранилища (имя компьютера \имя экземпляра сервера)&gt; Не удалось установить соединение с хранилищем: Не удается открыть базу данных &lt;имя БД&gt;, запрашиваемую именем входа. Не удалось выполнить вход</li> </ul> <p>Причина:</p> <ul style="list-style-type: none"> <li>› Невозможно получить доступ к БД</li> </ul> <p>Решение:</p> <ul style="list-style-type: none"> <li>› Проверить наличие запрашиваемой БД и права доступа к ней.</li> </ul>

<p>После резервного перехода в историю не записано ни одного значения сигнала</p>	<p>Возможная причина:</p> <ul style="list-style-type: none"><li>➤ Последнее неотфильтрованное значение пришло во время работы в резерве, значения сигнала, пришедшие после резервного перехода, были отфильтрованы.</li></ul> <p>Решение:</p> <ul style="list-style-type: none"><li>➤ Выбрать значение Да параметра модуля Записывать значения при активации</li></ul>
---	--

## 1.1.2.8.2. Модуль SnapShot



;

### Функции

- › сохранение текущих значений сигналов в файл-срезы XML-формата;
- › восстановление данных из файлов-срезов;
- › просмотр массива сгенерированных файлов-срезов.

Запуск операций по сохранению значений сигналов в файл-срез и восстановлению данных из файла-среза производится с помощью управляющих сигналов модуля.

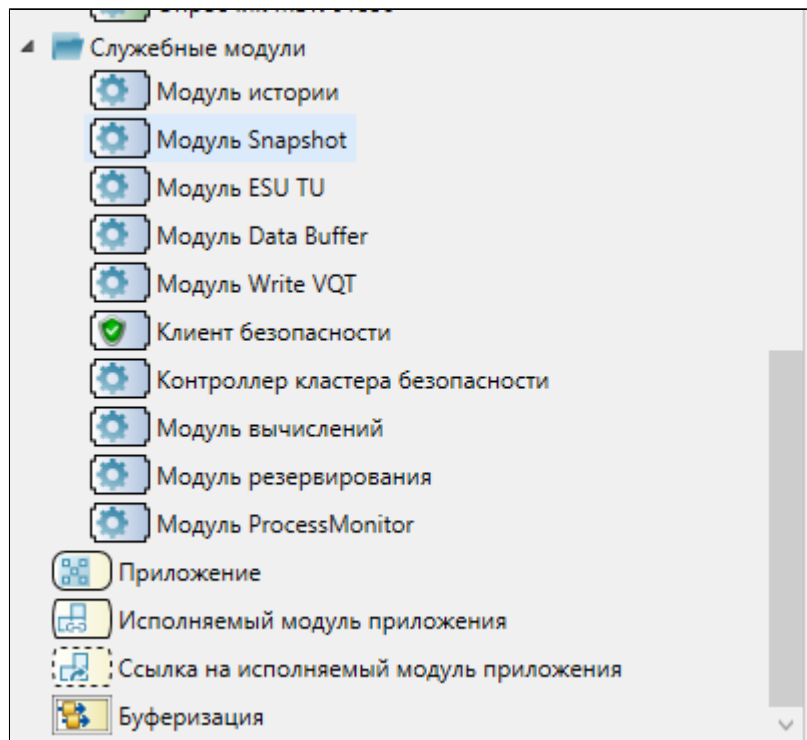


Просмотреть XML-файл, сформированный при генерации файла-среза, можно любым просмотрщиком текстовых файлов.



## 1.1.2.8.2.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

Свойства	
SnapShot Module Модуль Snapshot	
^ Параметры модуля	
Каталог файлов-срезов	C:\FSModule\
Каталог с шаблонами файлов-срезов	C:\FSModule\Templates\
Генерировать файл-срезы по абсолютному пути	Нет
Активность	Да
Отображаемое имя	
^ Параметры журналирования	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
^ Общие	
Имя	SnapShot Module

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Каталог файлов-срезов	Путь до папки, в которую генерируются файлы-срезы. По умолчанию: C:\FSModule\
Каталог с шаблонами файлов-срезов	путь до папки, в которой расположены шаблоны файлов-срезов. По умолчанию: C:\FSModule\Templates\
Генерировать файл-срезы по абсолютному пути	Если параметр активен, то в сигнале OutFileName можно указывать абсолютный путь для генерации файла-среза. Если параметр не активен, то генерация файла-среза производится в каталог, указанный в параметре Каталог

	файлов-срезом, а в сигнале OutFileName указывается только имя генерируемого файла.
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>› Да – модуль запущен;</li> <li>› Нет – модуль остановлен.</li> </ul>
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.8.2.2. Диагностика работы модуля

### Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

› в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

› в Linux системах в директории:



/opt/AstraRegul//Astra.Server/Logs.

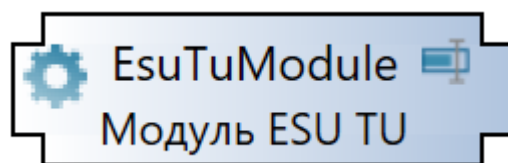
Для анализа ошибок, возникающих в процессе работы модуля, воспользуйтесь сервисным приложением Просмотрщик лога кадров.

№	Дата	Время	Описание
5	19.07.2021	17:00:35:181	Команда на генерацию файла-среза исполнена
6	19.07.2021	17:01:15:273	Команда смены выборки поставлена в очередь исполнения
7	19.07.2021	17:01:15:283	Начинаем исполнение команды по смене выборки
8	19.07.2021	17:01:15:308	Команда по смене выборки исполнена
9	19.07.2021	17:01:57:822	Команда на восстановление файл-среза поставлена в очередь исполнения
10	19.07.2021	17:01:57:831	Восстанавливаем данные из файла: C:\FSModule\vsto_tu1_lin 19-07-2017 17-00-35.xml
11	19.07.2021	17:01:57:831	Начинаем исполнение команды восстановления данных из файл-среза
12	19.07.2021	17:01:57:837	Команда на восстановление данных из файл-среза исполнена
13	19.07.2021	17:02:14:275	Журнал закрыт
14	19.07.2021	17:02:16:191	Журнал открыт для записи
15	19.07.2021	17:02:27:083	Команда смены выборки поставлена в очередь исполнения
16	19.07.2021	17:02:27:088	Начинаем исполнение команды по смене выборки
17	19.07.2021	17:02:27:145	Команда по смене выборки исполнена
18	19.07.2021	17:02:51:159	Команда на восстановление файл-среза поставлена в очередь исполнения
19	19.07.2021	17:02:51:167	Восстанавливаем данные из файла: C:\FSModule\vsto_tu1_lin 19-07-2017 17-00-35.xml
20	19.07.2021	17:02:51:167	Начинаем исполнение команды восстановления данных из файл-среза
21	19.07.2021	17:02:51:173	Команда на восстановление данных из файл-среза исполнена
22	19.07.2021	17:03:11:241	Журнал закрыт

### Возможные проблемы и пути их решения

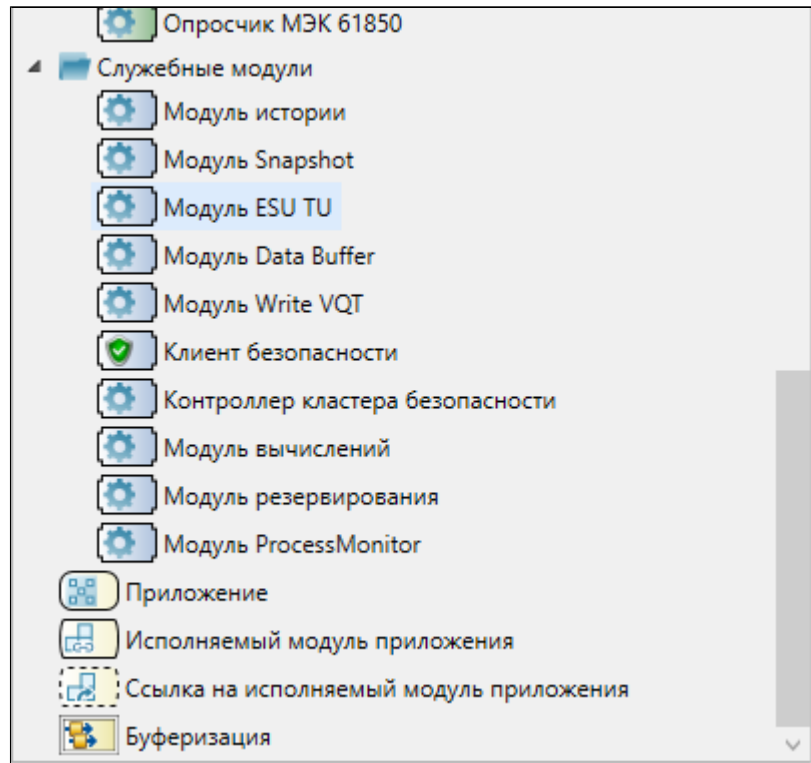
<b>Ситуация</b>	<b>Причины возникновения</b>	<b>Решения</b>
<p>Два сервера (один основной, другой резервный) установлены на двух компьютерах. На одном компьютере с установленным основным сервером генерируется файл-срез, а на другом компьютере с резервным сервером некорректно восстанавливаются значения вещественных сигналов</p>	<p>На компьютерах различаются региональные настройки (параметр: разделитель целой и дробной части)</p>	<p>Необходимо на компьютерах установить одинаковые региональные настройки</p>

### 1.1.2.8.3. Модуль ESU TU



# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуриатора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

EsuTuModule Модуль ESU TU	
<b>Параметры модуля</b>	
Настройки	
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	EsuTuModule

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

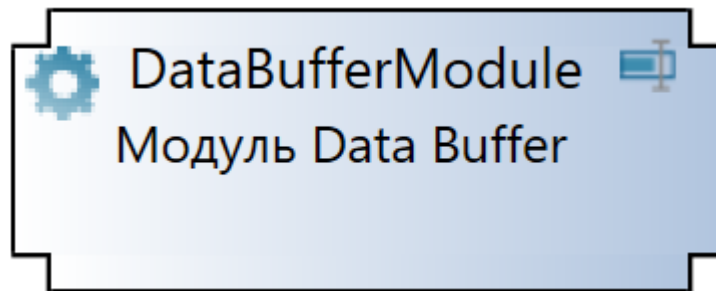
Параметр	Описание
Настройки	Путь к готовой конфигурация модуля
Активность	Активность модуля при запуске/перезапуске Astra.Server: > Да – модуль запущен; > Нет – модуль остановлен.
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

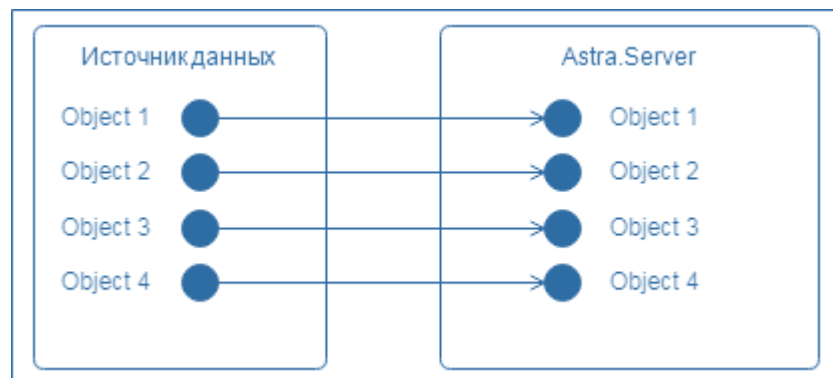
[Параметры журналирования](#) настраиваются одинаково для всех модулей.



## 1.1.2.8.4. Модуль Data Buffer

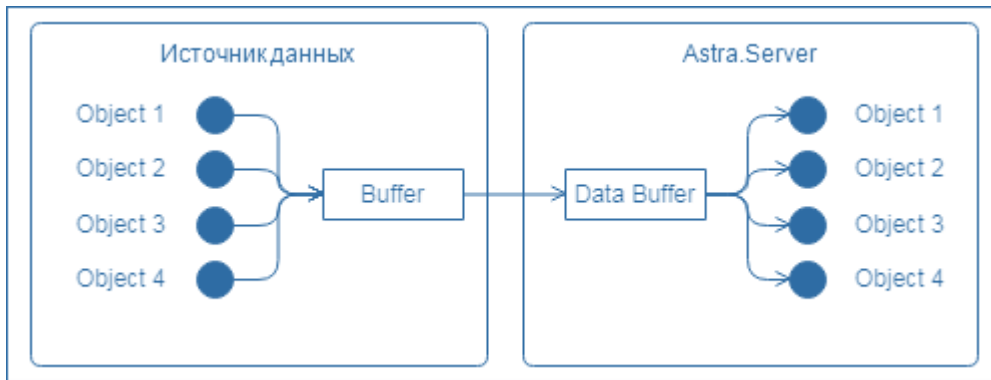


Сервер ввода/вывода `Astra.Server` получает от источников данные об объектах. Под объектами здесь понимаются отдельные параметры или структуры, содержащие набор параметров. Для этого в `Astra.Server` создаётся описание соответствующих объектов: каждому параметру объекта соответствует отдельный сигнал, для которого настраивается получение значений данного параметра объекта от соответствующего источника.



Однако, если в источнике выполняется буферизация отправляемых данных, то при получении данных из буфера источника необходимо сначала определить объект, к которому относятся полученные данные, после чего положить полученные данные в сигналы объекта-получателя.

Для решения данной задачи предназначен модуль `Data Buffer`.



Модуль Data Buffer выполняет следующие функции:

- определяет объект-получатель данных в Astra.Server
- перекладывает полученные данные в сигналы объекта-получателя

## Принцип работы

Буфер данных представляет из себя последовательность строк. Строка буфера данных – это отдельная запись в буфере данных источника о некотором событии.

Строка буфера данных состоит из полей:

- в одном из полей хранится код события и код объекта, в котором произошло событие. Данное поле будем

называть идентифицирующим;

В зависимости от реализации источника, код события может состоять из различного количества бит или отсутствовать вовсе. В зависимости от устройства идентифицирующего поля в источнике, настраиваются маски кода объекта и кода события в настройках модуля Data Buffer (см. подробнее).

- в остальных полях хранятся параметры события.

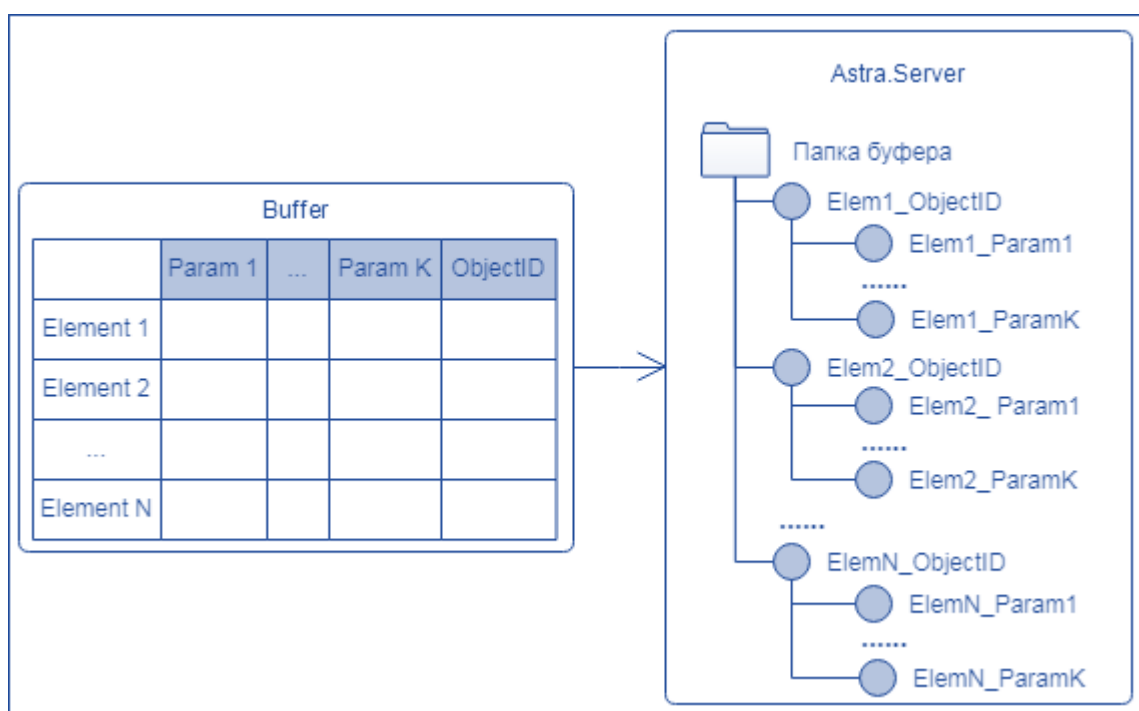


Примером параметра события является значение, вызвавшее событие.

Список полей в строке буфера одинаков для всех строк буфера данных и определяется устройством источника.

В Astra.Server получение данных из буфера источника может выполнять любой коммуникационный модуль. Для того, чтобы получение данных из буфера источника и разбор полученных данных выполнялись независимо, в Astra.Server полученные данные записываются в специальную папку буфера:

- каждое поле строки буфера записывается в подготовленный для него сигнал в папке буфера;
- каждой строке буфера (их количество определяется настройками источника) в папке буфера соответствует свой набор сигналов для хранения полей этой строки.



При записи строки буфера в сигналы значение идентифицирующего поля записывается последним: изменение этого сигнала означает, что все поля строки буфера записаны в соответствующие им дочерние сигналы и можно начинать разбор данных записанных в сигналы.

Разбор начинается при изменении значения идентифицирующего сигнала и состоит из следующих этапов:

- получение кода объекта из идентифицирующего сигнала с помощью маски кода объекта;
- перекладка значений сигналов из папки буфера в сигналы объекта, имеющего соответствующий код.

## Разбор идентифицирующего сигнала

Ниже показано, как модуль выделяет из идентифицирующего сигнала код объекта и код события.

В идентифицирующем сигнале (сигнал типа uint4 в папке буфера) содержится значение (к примеру 555). Модуль Data Buffer работает с этим значением в шестнадцатеричной системе счисления.

Значение в буфере <sup>(16)</sup>	00	00	02	2B
-----------------------------------	----	----	----	----

Модуль производит разбор сигнала в соответствии с принципом наложения масок на значение сигнала по правилам логической операции "И":

- › Маска значения – выделяет из сигнала часть, содержащую код события;
- › Маска кода объекта – выделяет из сигнала часть, содержащую код объекта.

Маски устанавливаются пользователем в настройках модуля.

## Выделение кода объекта

Значение в буфере <sup>(16)</sup>	00	00	02	2B
<b>Логическое «И»</b>				
Маска кода объекта	0F	FF	FF	00
=				
Код объекта <sup>(16)</sup>	00	00	02	00

## Выделение кода события

Значение в буфере <sup>(16)</sup>	00	00	02	2B
<b>Логическое «И»</b>				
Маска значения <sup>(16)</sup>	00	00	00	FF
=				
Код события <sup>(16)</sup>	00	00	00	2B

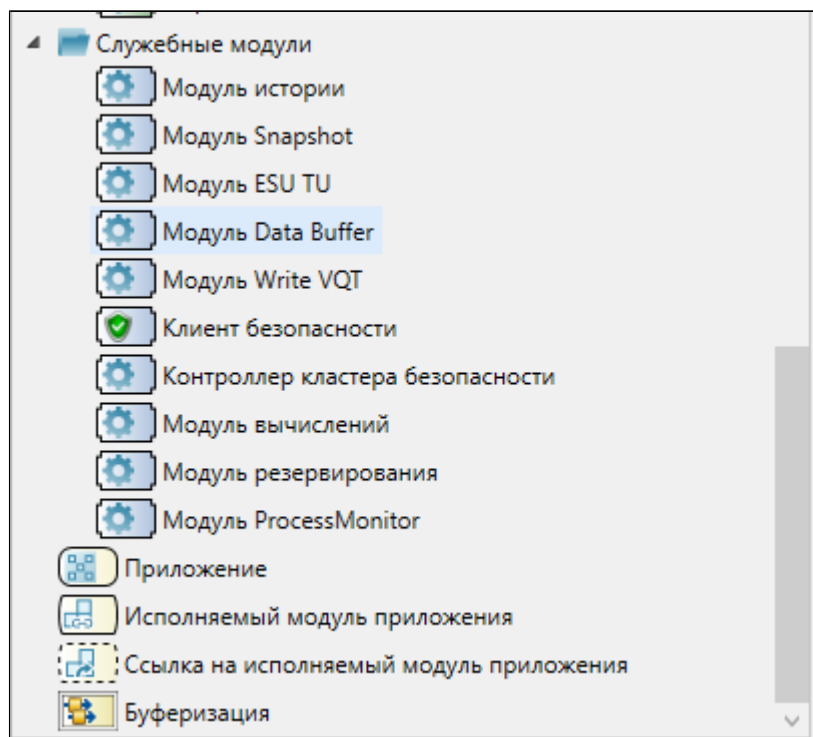
Результат разбора:

- › Код объекта – 2
- › Код события – 2B (в десятичной системе счисления – 43)

После разбора буфера модуль запишет код события 43 в сигнал-получатель в объекте 2.

# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигулятора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

DataBufferModule Модуль Data Buffer	
<b>Параметры модуля</b>	
Маска значения	0x000000FF
Маска кода объекта	0x0FFFFFF00
Папка буфера	NewFolder
Режим изменения метки времени при хорошем качестве	Нет
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	DataBufferModule

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Маска значения	Используется для выделения из идентифицирующего сигнала кода события
Маска кода объекта	Используется для выделения из идентифицирующего сигнала кода объекта
Папка буфера	Папка, в дереве сигналов, используемая для получения данных из буфера источника
Режим изменения метки времени	<ul style="list-style-type: none"> <li>› True – модуль будет устанавливать значения целевых сигналов только на основании изменений метки времени при хорошем качестве сигнала-источника</li> </ul>

при хорошем качестве	<ul style="list-style-type: none"> <li>› False – модуль будет устанавливать значения целевых сигналов при любых изменениях сигнала-источника</li> </ul>
Активность	Активность модуля: <ul style="list-style-type: none"> <li>› Да – модуль запущен</li> <li>› Нет – модуль остановлен</li> </ul>
Отображаемое имя	Имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.



## 1.1.2.8.4.2. Диагностика работы модуля

### Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.aplog по умолчанию:

› в ОС Windows в папке:



C:\Program Files\AstraRegul\Astra.Server\Logs;

› в Linux системах в директории:



/opt/AstraRegul/Astra.Server/Logs.

Журнал работы модуля записывается в файл <путь к каталогу сервера>/Logs/<имя модуля>.aplog.

Для просмотра журнала работы модуля воспользуйтесь сервисным приложением Просмотрщик лога кадров.

[Astra.Server] Просмотр журналов - C:\Program Files\ProSyst\Astra.Server\Logs\Data Buffer.aplog (Общий журн... — □ ×

Файл Фильтр Поиск Помощь

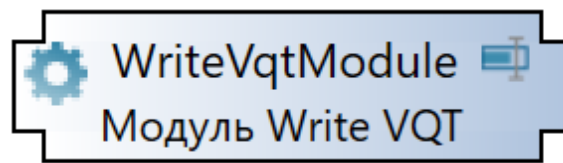
№	Дата	Время	Описание
36	28.07.2021	12:44:11:769	Журнал закрыт
37	28.07.2021	12:44:27:129	Журнал открыт для записи
38	28.07.2021	12:44:29:801	Журнал закрыт
39	28.07.2021	12:44:43:832	Журнал открыт для записи
40	28.07.2021	12:44:46:363	Журнал закрыт
41	28.07.2021	12:45:00:582	Журнал открыт для записи
42	28.07.2021	12:45:07:488	Журнал закрыт
43	28.07.2021	13:15:52:040	Журнал открыт для записи
44	28.07.2021	13:16:12:102	Журнал закрыт
45	28.07.2021	13:16:15:196	Журнал открыт для записи
46	28.07.2021	13:34:05:300	Журнал закрыт

Журнал закрыт

0 1 2 3 4 5 6 7

Количество записей: 46

## 1.1.2.8.5. Модуль Write-VQT



Согласно спецификации OPC, изменение сигнала подразумевает комплексное воздействие на его свойства: значение (Value, свойство 2), качество (Quality, свойство 3) и метку времени (Timestamp, свойство 4).

Модуль Write VQT позволяет изменять каждое из перечисленных свойств независимо друг от друга.

### Принцип работы

Чтобы изменить значение, качество или метку времени родительского сигнала, установите нужные значения дочерним сигналам `vqt_value`, `vqt_quality`, `vqt_timestamp`, `vqt_timestamp64`, а затем задайте маску через сигнал `vqt_changemask`. Если указаны оба формата метки времени - `vqt_timestamp64` и `vqt_timestamp`, то приоритет отдается метке времени из сигнала `vqt_timestamp64`.

В таблице ниже перечислены соответствия между значениями маски и изменяемыми свойствами родительского сигнала.

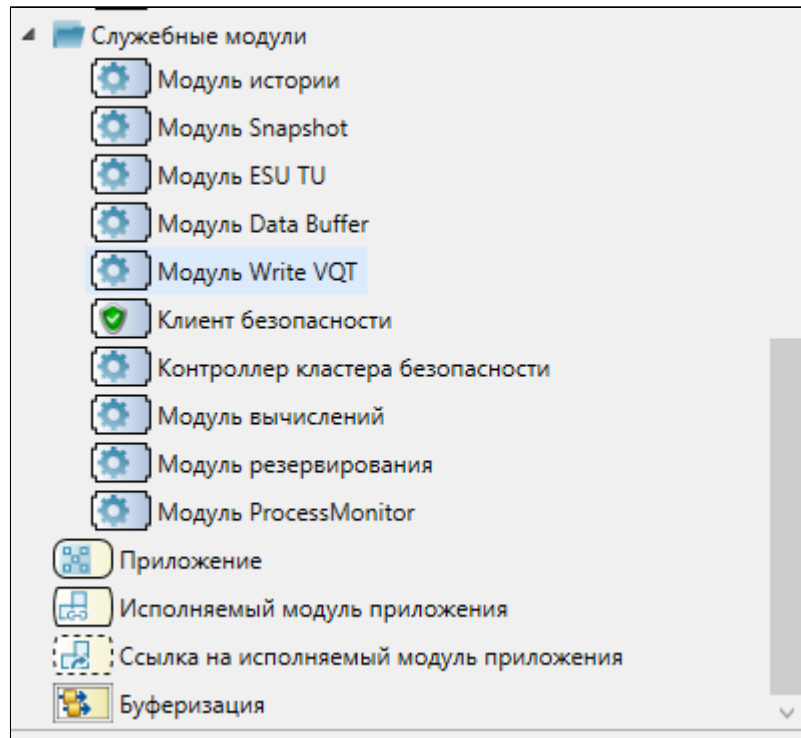
Значение маски <code>vqt_changemask</code>	Изменяемые свойства родительского сигнала		
	Value (2)	Quality (3)	Timestamp (4)
1	+		
2		+	
3	+	+	
4			+
5	+		+

6		+	+
7	+	+	+
32*			+
33*	+		+
34*		+	+
35*	+	+	+

\*Если используются маски 32 и более, то свойству сигнала Timestamp присваивается текущее системное время.

## 1.1.2.8.5.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигурирования Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

WriteVqtModule Модуль Write VQT	
<b>Параметры модуля</b>	
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	WriteVqtModule

### Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен;</li> <li>&gt; Нет – модуль остановлен.</li> </ul>
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.8.5.2. Диагностика работы модуля

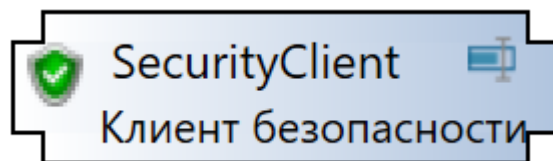
### Журнал работы модуля

Журнал работы модуля записывается в файл <путь к каталогу сервера>/Logs/<имя модуля>.arlog.

Для просмотра журнала работы модуля воспользуйтесь сервисным приложением Просмотрщик лога кадров.

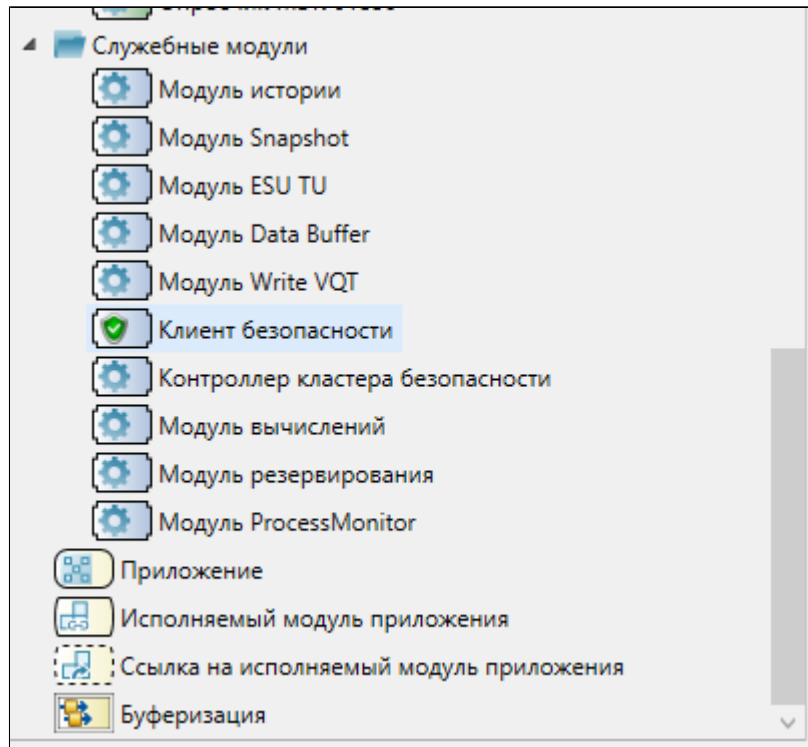
№	Дата	Время	Описание
0	03.08.2021	16:44:57:814	Журнал открыт для записи
1	04.08.2021	09:18:10:134	Сигнал TM.D, маска 0x04: для изменения указанных свойств значение сигнала должно быть определено.
2	04.08.2021	09:18:34:263	Сигнал TM.D, маска 0x01: значение сигнала не изменено, так как значение служебного сигнала 'vqt_value' недостоверно.
3	04.08.2021	09:18:38:176	Сигнал TM.D, маска 0x02: значение сигнала не изменено, так как значение служебного сигнала 'vqt_quality' недостоверно.
4	04.08.2021	09:18:40:921	Сигнал TM.D, маска 0x03: значение сигнала не изменено, так как значение служебного сигнала 'vqt_quality' недостоверно.
5	04.08.2021	09:18:45:098	Сигнал TM.D, маска 0x05: значение сигнала не изменено, так как значение служебного сигнала 'vqt_value' недостоверно.
6	04.08.2021	09:18:49:218	Сигнал TM.D, маска 0x06: значение сигнала не изменено, так как значение служебного сигнала 'vqt_quality' недостоверно.
7	04.08.2021	09:18:52:259	Сигнал TM.D, маска 0x07: значение сигнала не изменено, так как значение служебного сигнала 'vqt_quality' недостоверно.

## 1.1.2.8.6. Клиент безопасности

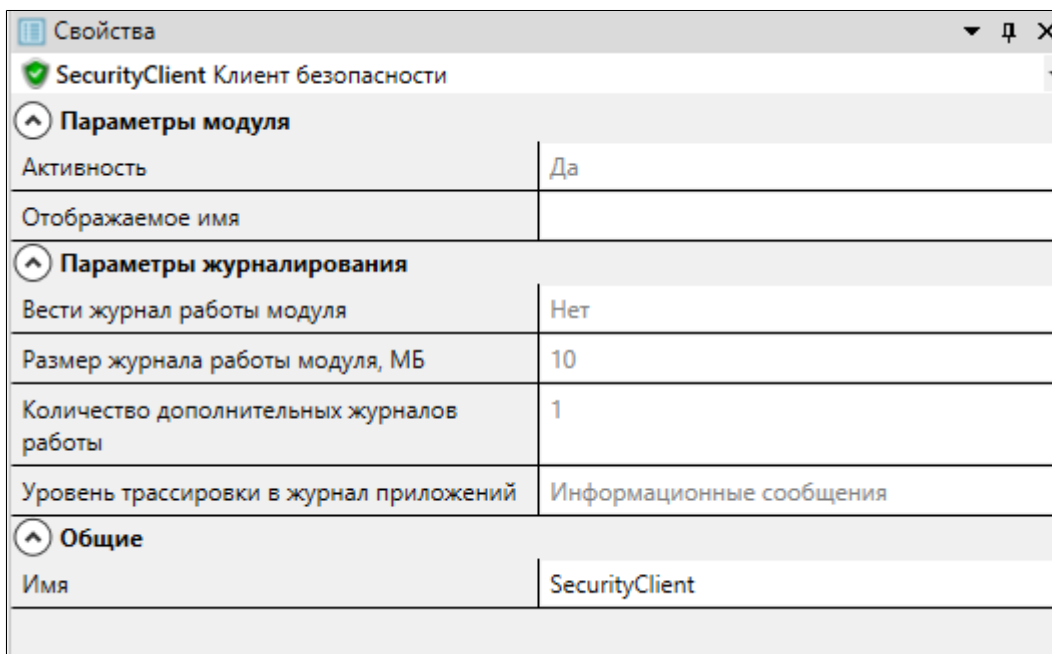


# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора [Astra.AStudio](#).



Выделите модуль, чтобы настроить его свойства.



## Общие



Параметр	Описание
Имя	Имя модуля

## Параметры модуля

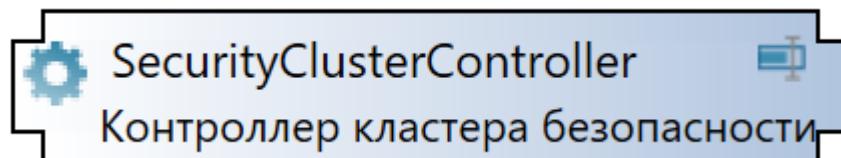
Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"> <li>&gt; Да – модуль запущен;</li> <li>&gt; Нет – модуль остановлен.</li> </ul>
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

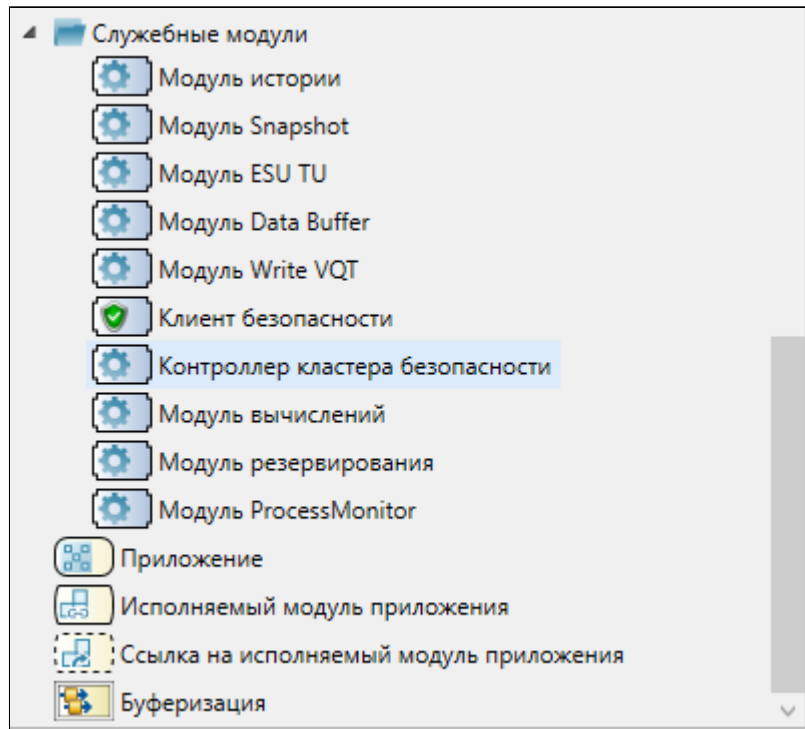
## 1.1.2.8.7. Контроллер кластера безопасности

Модуль реализации контроллера кластера безопасности.

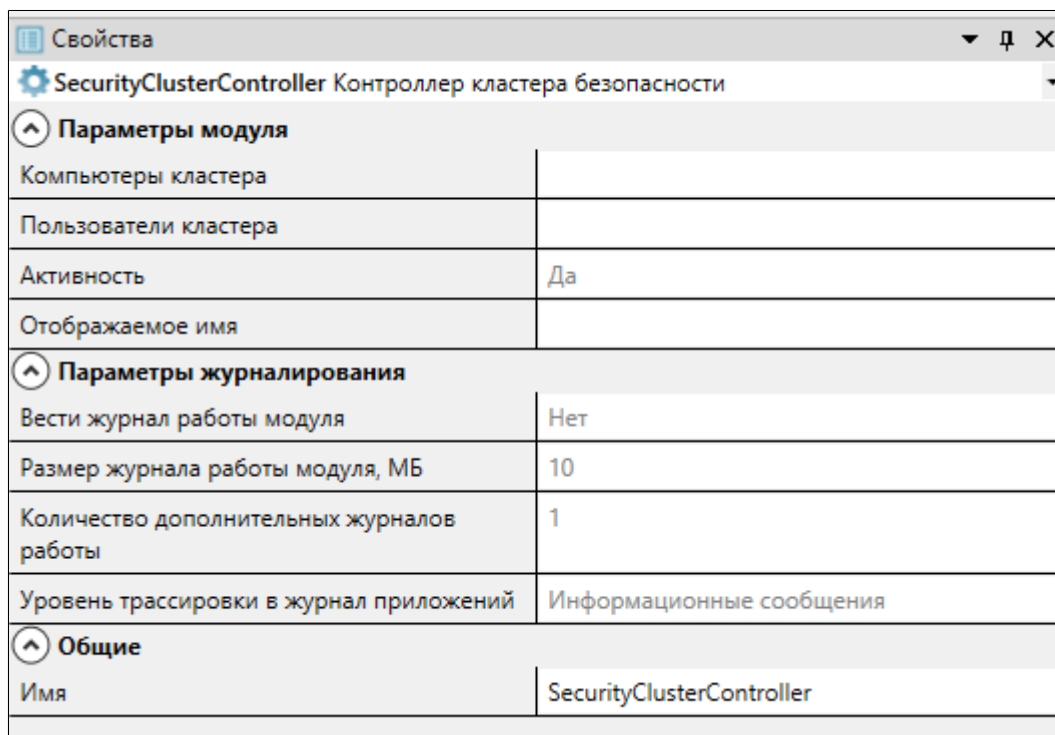


# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигурирования Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.



## Общие

Параметр	Описание
Имя	Имя модуля

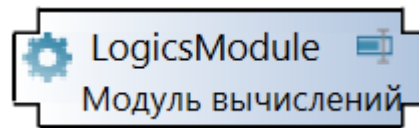
## Параметры модуля

Параметр	Описание
Компьютеры кластера	Компьютеры кластера безопасности
Пользователи кластера	Пользователи кластера безопасности
Активность	Активность модуля при запуске/перезапуске Astra.Server: > Да – модуль запущен; > Нет – модуль остановлен.
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

## 1.1.2.8.8. Модуль вычислений



Основное назначение модуля – логико-вычислительная обработка собранных оперативных данных. Возможности модуля определяются возможностями языка Astra.Om.

### Примечания к работе флагов обратной совместимости



Если Регистрозависимость исходного кода отключена (режим обратной совместимости):

- › Нельзя обратиться к пространству имён Variant.
- › Для обращения к пространству имён String используйте имя Str.



Не рекомендуется включать флаг Игнорировать правила приведения типов для деклараций, так как это может привести к потере точности данных.



Если в формуле есть операция, в которой операнды имеют недопустимый тип, и включён флаг Разрешить неявные преобразования из примитивных типов в вариант, то операнды будут приведены к типу variant, а результат выполнения операции будет VT\_EMPTY (неопределённое значение).

Например, выражение `100 && 200` будет успешно скомпилировано. Если флаг выключен, произойдёт ошибка компиляции (правильное поведение).



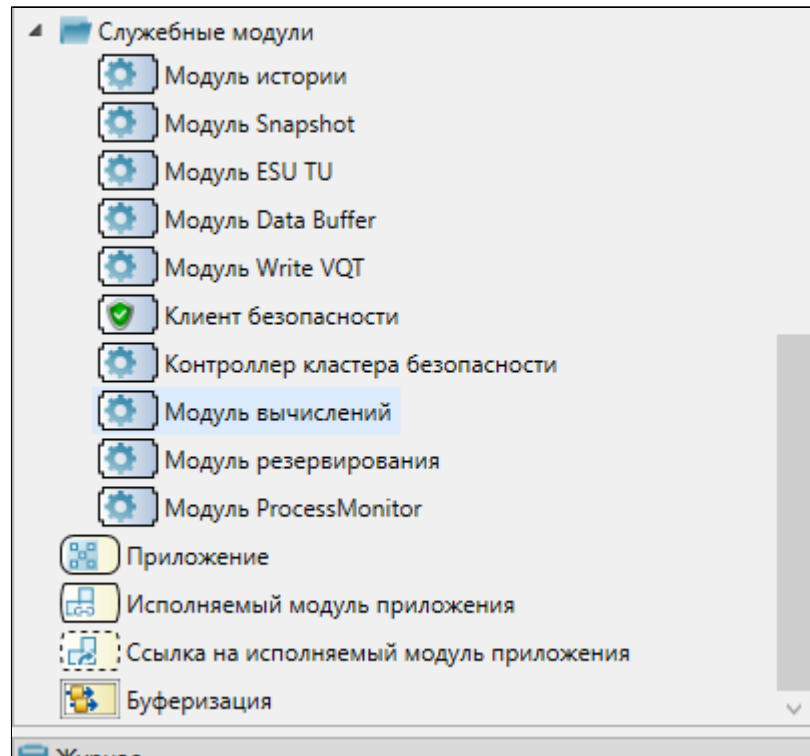
Если выполняется логическая операция над целыми числами и включены флаги Неявное приведение целочисленного значения к булевому и Разрешить неявные преобразования из примитивных типов в вариант, то в операции операнды будут приведены к

логическому типу (а не к типу variant) и операция вернёт логическое значение true или false.

Например, в выражении `100 && 200` значения 100 и 200 будут преобразованы в true и результатом вычисления значения выражения будет true.

# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигурирования [Astra.AStudio](#).



Выделите модуль, чтобы настроить его свойства.

LogicsModule Модуль вычислений	
<b>Параметры модуля</b>	
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	LogicsModule

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

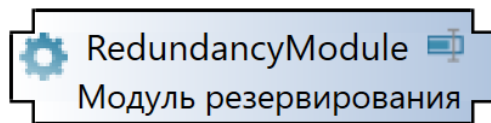
Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: > Да – модуль запущен; > Нет – модуль остановлен. Управляется служебным сигналом Active.Set
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.



## 1.1.2.8.9. Модуль резервирования



Задача модуля резервирования — повышение надежности системы сбора технологических данных за счет автоматического контроля и управления серверами резервной пары серверов Astra.Server.



Если серверы резервной пары защищены от несанкционированного управления, то пароли доступа к каждому из серверов должны быть одинаковыми.

Функции:

- › синхронизация значений сигналов между серверами резервной пары;
- › поддержка резервной пары серверов в устойчивом состоянии.

### Состояния резервной пары

Резервная пара серверов находится в устойчивом состоянии, если один из серверов находится в режиме РАБОТА (основной сервер), а второй — в режиме РЕЗЕРВ (резервный сервер).

Возможные состояния резервной пары приведены в таблице ниже.

Состояние резервной пары	Сервер 1	Сервер 2
Остановлена	Остановлен	Остановлен
Работа (устойчивое состояние)	Работа	Резерв
	Резерв	Работа
Работа без пары	Работа	Остановлен

	Остановлен	Работа
Ошибка	Работа	Работа

Изменения состояний серверов резервной пары отражаются в журнале работы модуля. Текущее состояние серверов резервной пары отражено в значениях сервисных сигналов.

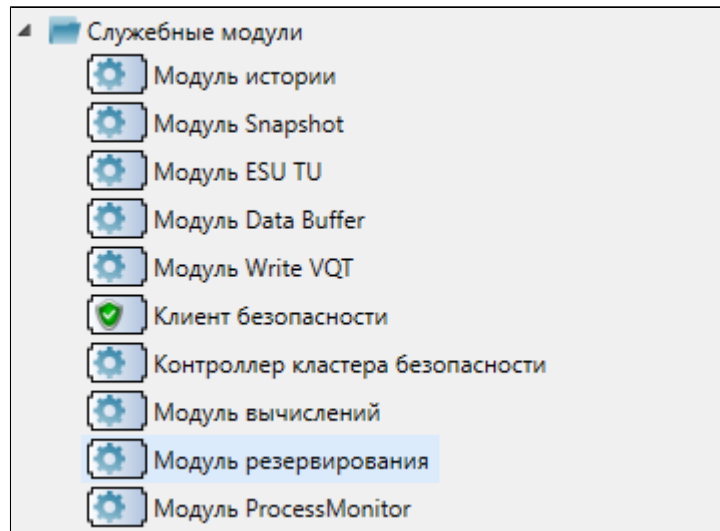
## **Остановка основного сервера в резервной паре**

Порядок перехода основного сервера из режима РАБОТА в режим ОСТАНОВЛЕН:

1. Основной сервер переходит из режима РАБОТА в режим РЕЗЕРВ.
2. Резервный сервер, обнаружив, что основной сервер перешел в режим РЕЗЕРВ, сразу же переходит в режим РАБОТА.
3. После перехода резервного сервера в режим РАБОТА, текущий сервер останавливает свою работу.

## 1.1.2.8.9.1. Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигуратора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

⚙️ RedundancyModule Модуль резервирования	
⤴️ <b>Параметры модуля</b>	
Активность	Да
Отображаемое имя	
⤴️ <b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
⤴️ <b>Общие</b>	
Имя	RedundancyModule

### Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"><li>› Да – модуль запущен;</li><li>› Нет – модуль остановлен.</li></ul>
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

[Параметры журналирования](#) настраиваются одинаково для всех модулей.

# 1.1.2.8.9.2. Диагностика работы модуля

## Журнал работы модуля

Журнал работы модуля сохраняется в файл <имя модуля>.arlog по умолчанию:

» в ОС Windows в папке:

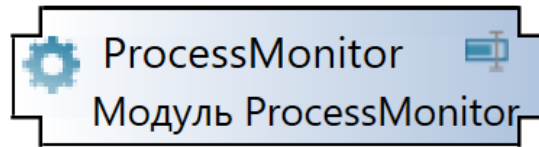


C:\Program Files\AstraRegul\Astra.Server\Logs;

Для просмотра журнала работы модуля воспользуйтесь сервисным приложением Просмотрщик лога кадров.

ID	Дата	Время	Описание
26	17.02.2021	13:11:20:775	Изменился режим работы парного сервера.
27	17.02.2021	13:11:26:347	Изменился режим работы парного сервера.
28	17.02.2021	13:11:26:487	Установлено соединение (для передачи данных) с парным сервером по резервному каналу ("127.0.0.1 - 127.0.0.1").
29	17.02.2021	13:11:42:139	Закрыто соединение (для передачи данных) с парным сервером по резервному каналу ("127.0.0.1 - 127.0.0.1").
30	17.02.2021	13:11:42:169	Изменился режим работы парного сервера.
31	17.02.2021	13:11:54:798	Изменился режим работы парного сервера.
32	17.02.2021	13:11:54:888	Установлено соединение (для передачи данных) с парным сервером по резервному каналу ("127.0.0.1 - 127.0.0.1").
33	17.02.2021	13:11:57:677	Закрыто соединение (для передачи данных) с парным сервером по резервному каналу ("127.0.0.1 - 127.0.0.1").
34	17.02.2021	13:11:58:017	Изменился режим работы парного сервера.
35	17.02.2021	13:12:03:372	Изменился режим работы парного сервера.
36	17.02.2021	13:12:03:512	Установлено соединение (для передачи данных) с парным сервером по резервному каналу ("127.0.0.1 - 127.0.0.1").
37	17.02.2021	13:12:05:949	Закрыто соединение (для передачи данных) с парным сервером по резервному каналу ("127.0.0.1 - 127.0.0.1").
38	17.02.2021	13:12:06:598	Изменился режим работы парного сервера.
39	17.02.2021	13:13:33:707	Изменился режим работы парного сервера.
40	17.02.2021	13:13:33:797	Установлено соединение (для передачи данных) с парным сервером по резервному каналу ("127.0.0.1 - 127.0.0.1").
41	17.02.2021	13:16:15:503	Изменился режим работы парного сервера.
42	17.02.2021	13:16:15:503	Закрыто соединение (для передачи данных) с парным сервером по резервному каналу ("127.0.0.1 - 127.0.0.1").
43	17.02.2021	13:17:53:151	Потеряно соединение (для опроса состояния) с парным сервером по резервному каналу "127.0.0.1 - 127.0.0.1": Сервер RPC недоступен.
44	17.02.2021	13:17:53:171	Потеряно соединение (для опроса состояния) с парным сервером по основному каналу "127.0.0.1 - 127.0.0.1": Сервер RPC недоступен.
45	17.02.2021	13:17:53:171	Отсутствуют соединения с парным сервером по всем каналам.
46	17.02.2021	13:17:53:391	Установлено соединение (для опроса состояния) с парным сервером по основному каналу ("127.0.0.1 - 127.0.0.1").
47	17.02.2021	13:17:53:391	Обнаружен конфликт состояний серверов: работают оба сервера. Конфликт не удалось разрешить ни по одному из критериев.
48	17.02.2021	13:17:53:421	Установлено соединение (для опроса состояния) с парным сервером по резервному каналу ("127.0.0.1 - 127.0.0.1").
49	17.02.2021	13:17:57:540	Журнал закрыт

## 1.1.2.8.10. Модуль ProcessMonitor



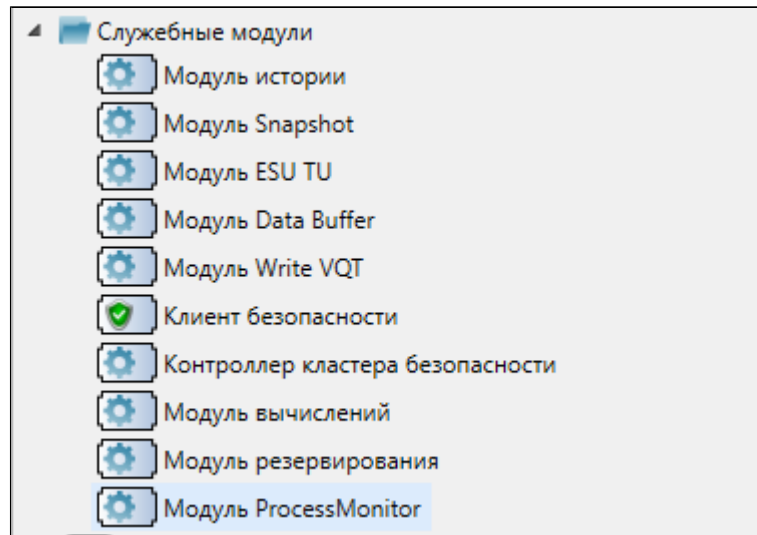
Модуль ProcessMonitor предназначен для отслеживания процессов Windows. Модуль позволяет отслеживать процессы приложений, которые запускаются только в единственном экземпляре.

При запуске Astra.Server модуль динамически создаёт сигналы, в которых отображается информация об отслеживаемом процессе:

- › общее использование процессора на всех ядрах;
- › состояние выполнения процесса;
- › время выполнения процесса.

# Настройка

Для добавления в состав конфигурации, перетащите модуль в рабочую область из панели элементов конфигулятора Astra.AStudio.



Выделите модуль, чтобы настроить его свойства.

ProcessMonitor Модуль ProcessMonitor	
<b>Параметры модуля</b>	
Активность	Да
Отображаемое имя	
<b>Параметры журналирования</b>	
Вести журнал работы модуля	Нет
Размер журнала работы модуля, МБ	10
Количество дополнительных журналов работы	1
Уровень трассировки в журнал приложений	Информационные сообщения
<b>Общие</b>	
Имя	ProcessMonitor

## Общие

Параметр	Описание
Имя	Имя модуля

## Параметры модуля

Параметр	Описание
Активность	Активность модуля при запуске/перезапуске Astra.Server: <ul style="list-style-type: none"><li>› Да – модуль запущен;</li><li>› Нет – модуль остановлен.</li></ul>
Отображаемое имя	Отображаемое имя модуля

## Параметры журналирования

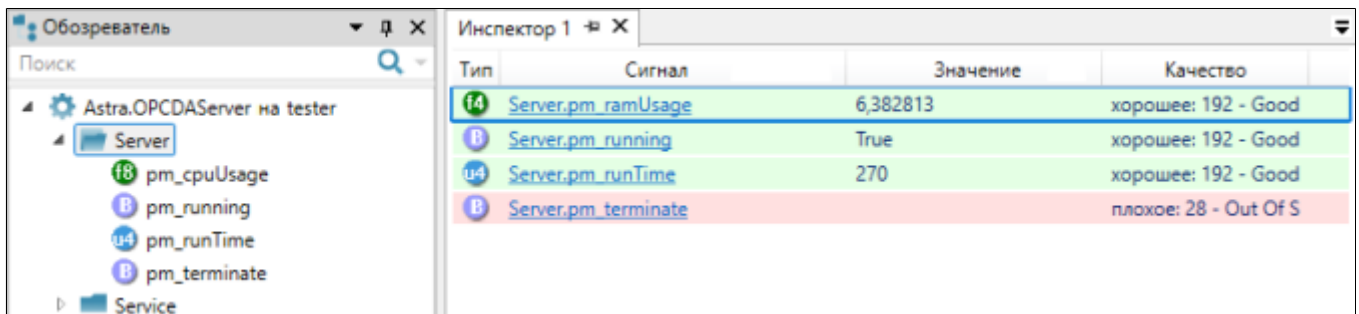
[Параметры журналирования](#) настраиваются одинаково для всех модулей.



## 1.1.2.8.10.2. Диагностика работы модуля

Для просмотра динамических сигналов модуля подключитесь к Astra.Server утилитой OpсExplorer.

Модуль ProcessMonitor создаёт для папки или сигнала указанного процесса дочерние сигналы мониторинга.



The screenshot shows the OpсExplorer interface. On the left, a tree view displays the module structure: Astra.OPCDAServer на tester > Server > pm\_cpuUsage, pm\_running, pm\_runTime, pm\_terminate, and Service. On the right, the 'Инспектор 1' (Inspector 1) window shows a table of signals:

Тип	Сигнал	Значение	Качество
f4	Server.pm_ramUsage	6,382813	хорошее: 192 - Good
B	Server.pm_running	True	хорошее: 192 - Good
u4	Server.pm_runTime	270	хорошее: 192 - Good
B	Server.pm_terminate		плохое: 28 - Out Of S



Установка сигналу «pm\_terminate» значения «True» завершает отслеживаемый процесс, однако данный способ завершения процесса крайне нежелателен.

Сигнал	Тип	Описание сигнала
pm_cpuUsage	float	Общее использование процессора на всех ядрах в процентах
pm_running	bool	Состояние выполнения процесса: > True – процесс выполняется. > False – процесс завершен.
pm_runTime	uint4	Время выполнения процесса в секундах
pm_terminate	bool	Команда: > True – завершить процесс (не рекомендуется).

## 1.1.3. Адресное пространство

Сигналы Astra.Server используются для передачи значений контролируемых параметров пользователю и выдачи управляющих воздействий пользователя технологическим объектам.

Просмотр адресного пространства Astra.Server возможен при подключении клиентов по спецификации OPC UA. Полное имя сигнала формируется из имен узлов дерева сигналов, последовательно включающих данный сигнал. Точка в полном имени сигнала является разделителем уровней и не может входить в имя сигнала.

## 1.1.3.1. Типы сигналов

**Статические** – сохраняются в файл конфигурации AstraServer.cfg. Инициализирующие значения статистических сигналов задаются и корректируются с помощью конфигуратора Astra.AStudio. Заданные значения устанавливаются сигналам и свойствам при старте сервера. В случае изменения значений сигналов и свойств с помощью OPC-клиентов, внесенные изменения действуют только до перезапуска сервера. OPC-клиенты могут подписываться только на именованные свойства сервера.

**Динамические** – создаются ядром сервера или модулями сервера. Динамические сигналы не сохраняются в конфигурации сервера. Служебные сигналы, созданные модулями и ядром сервера, позволяют наблюдать за работой сервера или конкретного модуля. Просмотр и изменение сигналов и свойств возможен только с помощью OPC-клиентов. Изменения значений динамических сигналов и свойств действуют только до перезагрузки сервера. После перезапуска сервера сигнал или свойство принимает исходное значение.

## 1.1.3.2. Типы данных

Тип данных определяет множество допустимых значений, которые может принимать параметр, принадлежащий этому типу.

Astra.Server поддерживает следующие типы данных:

Тип	Описание	Допустимые значения
 Int1	Знаковое целое 1 байт	[-128; 127]
 UInt1	Беззнаковое целое 1 байт	[0; 255]
 Int2	Знаковое целое 2 байта	[-32 768; 32 767]
 UInt2	Беззнаковое целое 2 байта	[0; 65 535]
 Int4	Знаковое целое 4 байта	[-2 147 483 648; 2 147 483 647]
 UInt4	Беззнаковое целое 4 байта	[0; 4 294 967 295]
 Int8	Знаковое целое 8 байт	[-9 223 372 036 854 775 808; 9 223 372 036 854 775 807]
 UInt8	Беззнаковое целое 8 байт	[0; 18 446 744 073 709 551 615]
 Float	Значение с плавающей запятой 4 байта	$[\pm 1.5 \times 10^{-45}; \pm 3.4 \times 10^{38}]$ . Точность 6-9 цифр
 Double	Значение с плавающей запятой 8 байт	$[\pm 5.0 \times 10^{-324}; \pm 1.7 \times 10^{308}]$ . Точность 15-17 цифр
 Bool	Логическое значение	true, false
 String	Текстовая строка в кодировке UTF16	до 2 миллиардов знаков, каждый знак занимает 16 бит (2 байта)

### 1.1.3.3. Качество сигналов

Достоверность значения технологического параметра определяется исходя из качества сигнала Quality.

Инициализирующие значения качества сигналов настраиваются в конфигурации Astra.Server. В случае если качество не было задано на этапе конфигурирования сигналов, при старте сервера ядро устанавливает всем сигналам качество OUT\_OF\_SERVICE. Указанное значение качества говорит о том, что сигнал не поставлен на обслуживание ни одним модулем. Далее сигналы передаются на обслуживание модулям и качество выставляется обслуживающими модулями. Качество сигнала может изменяться при пересчете значения сигнала.

Качество сигналов можно посмотреть, подключившись к Astra.Server с помощью OPC-клиента.

Значение	Описание
GOOD (192)	Значение получено от источника данных с хорошим качеством
CALCULATED (200)	Значение получено в результате вычислений на основе данных с хорошим качеством
LOCAL_OVERRIDE (216)	Значение изменено пользователем вручную
BAD (0)	Данные получены с плохим качеством
CONFIG_ERROR (4)	Неверно заданы параметры сигнала (например, адрес). Расширенную информацию можно получить из журнала работы модуля.
NOT_CONNECTED (8)	Модуль, обслуживающий сигнал, не подключен к источнику данных (соединение не установлено). Например, последовательный порт, через который поступает значение сигнала, не открыт.

DEVICE_FAILURE (12)	Модулю, обслуживающему сигнал, не удалось подключиться к источнику данных. Например, не удалось открыть последовательный порт, через который получается значение сигнала.
SENSOR_FAILURE (16)	Источник данных, с которого должно быть получено значение сигнала, неисправен.
LAST_KNOWN (20)	Связь с источником данных, поставляющим значение сигнала, потеряна. Отображается последнее известное значение параметра
COMM_FAILURE (24)	Не удалось установить связь с источником данных, от которого должно быть получено значение сигнала.
OUT_OF_SERVICE (28)	Сигнал не обслуживается ни одним активным (запущенным) модулем. Например: адрес сигнала содержит неверный тип протокола.
UNCERTAIN (64)	Связь с источником данных установлена, но данные еще не получены
EGU_EXCEEDED (84)	Возвращаемое значение выходит за границы, определенные для этого параметра. Например, значение сигнала вышло за рамки при пересчете.
SUB_NORMAL (88)	Значение сигнала вычислено на основе нескольких сигналов и не все сигналы имеют достоверное качество GOOD.

## 1.2. Astra.AccessPoint

**Astra.AccessPoint** – единая точка доступа, объединяющая данные с различных источников (серверов).

Устанавливается на АРМ оператора (тонкий клиент).

Источники данных, с которыми будет работать Astra.AccessPoint, добавляются в модуль HubModule.

Из сигналов источников будет строиться объединенное адресное пространство Astra.AccessPoint.

Клиенты подключаются к Astra.AccessPoint по спецификациям OPC DA, OPC UA и OPC AE как к единой точке доступа.

Функции:

- › объединение адресных пространств нескольких источников данных;
- › поддержание связи с источниками данных;
- › передача значений и событий в виде TCP/IP-трафика в условиях различных сетевых топологий;
- › поддержка сбора данных по файловому интерфейсу;
- › доступ к данным Astra.AccessPoint по спецификациям OPC DA, OPC AE, OPC UA.



Спецификации OPC DA и OPC AE базируются на COM/DCOM и используются только в ОС Windows.

Astra.AccessPoint может выступать в качестве источника данных (режим каскадирования).



Данные между Astra.AccessPoint и Astra.Server передаются в зашифрованном виде.

Для шифрования данных используются алгоритмы ООО "РЕГЛАБ".

---

## Сервисные приложения

В состав дистрибутива Astra.AccessPoint входят сервисные приложения:

- › [Конфигуратор](#)
- › [Статистика](#)
- › [Просмотрщик лога кадров](#)

Каталог по умолчанию:



C:\Program Files\AstraRegul\Astra.AccessPoint\Service.

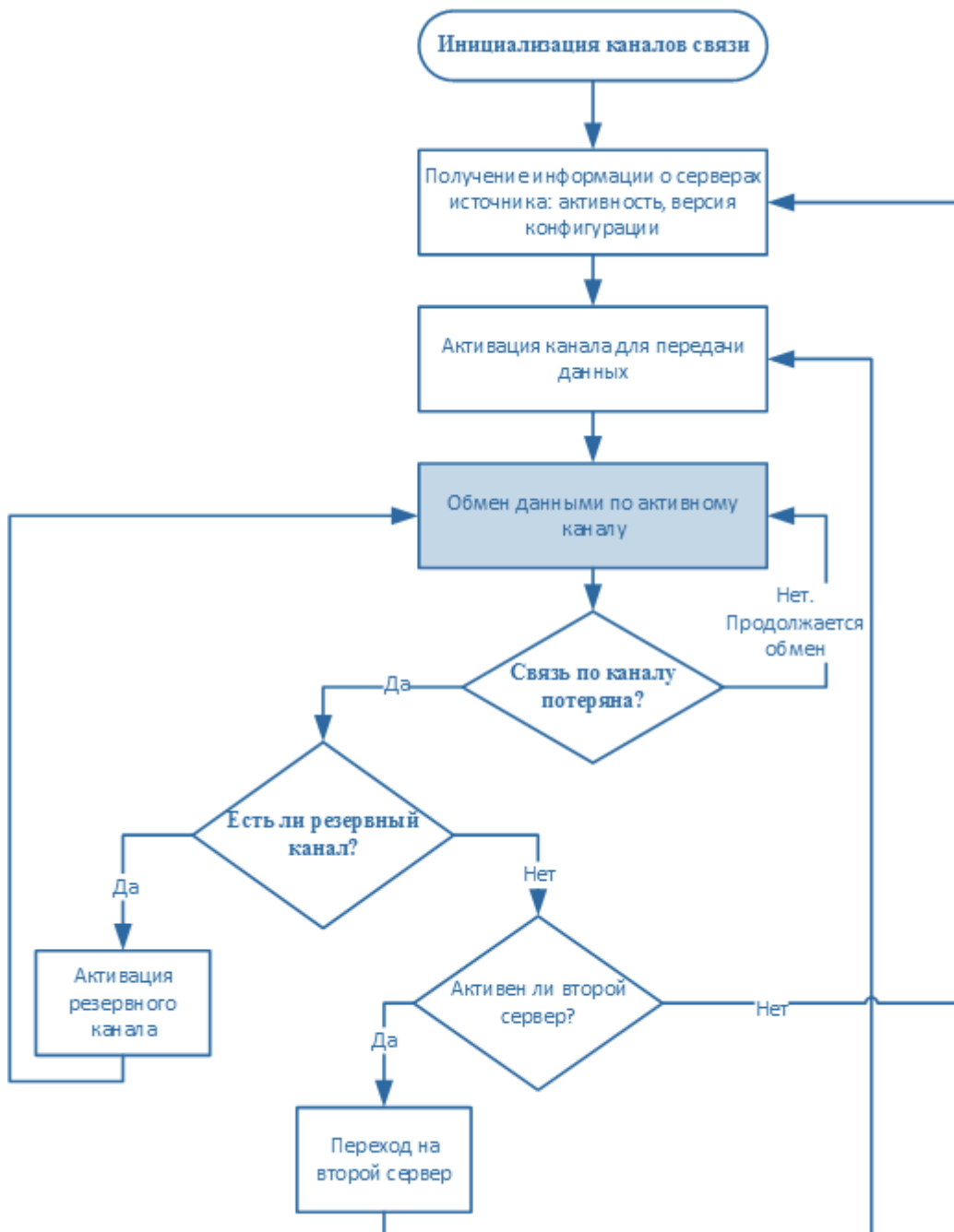


## Работа с резервируемой парой серверов

Astra.AccessPoint автоматически проводит инициализацию всех указанных каналов связи, а затем определяет активный сервер в составе источника данных.

В случае разрыва соединения Astra.AccessPoint пытается восстановить связь через резервный канал, если такой имеется.

В случае отсутствия резервных каналов в составе активного сервера, Astra.AccessPoint переключается на работу с резервным сервером.



## Модули

Модуль	Поддержка в ОС	
	Windows	Linux
HUB Module	+	+
TCP Server Module	+	+
OPC AE Server	+	+
OPC UA	+	+
History Module	+	+
OPC DA Server	+	
OPC HDA Server	+	
OPC HDA Client	+	

## 1.2.1. Настройка

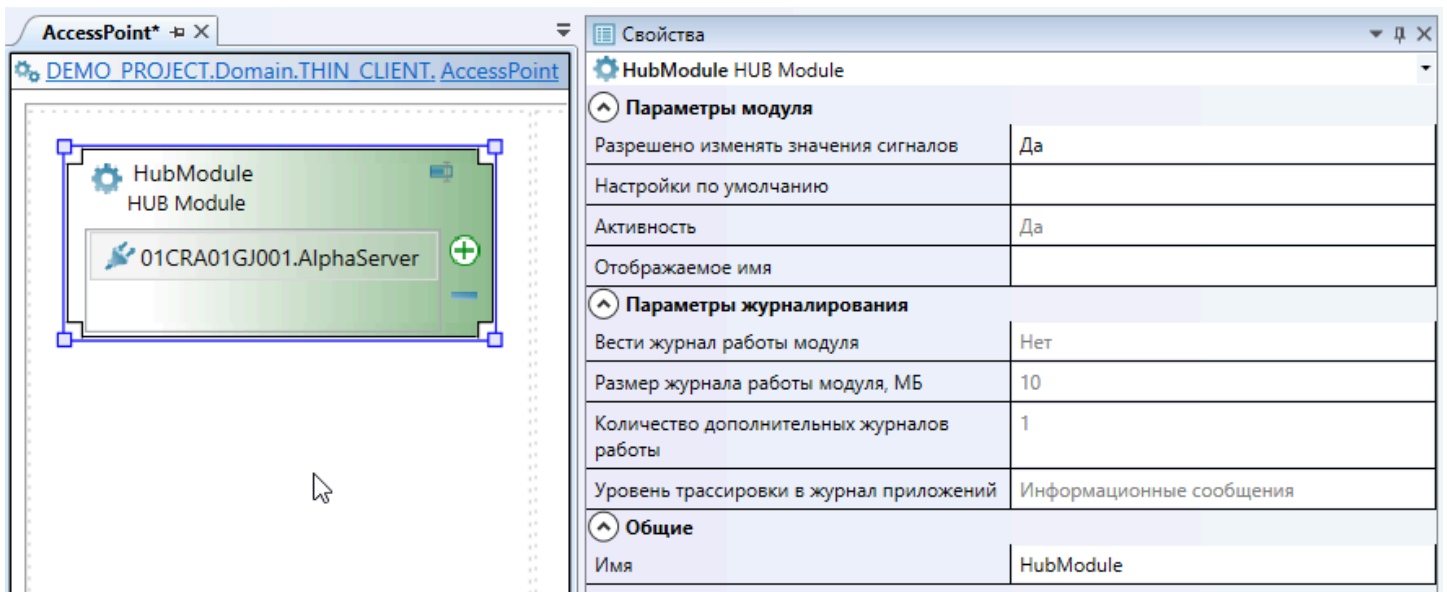
Базовая настройка APM оператора и службы Astra.AccessPoint происходит в среде разработки Astra.IDE.



Должен быть установлен плагин AstraRegul.

Для более точной настройки Astra.AccessPoint используется конфигуратор Astra.Astudio.

## 1.2.1.1. Настройки HubModule



### Общие параметры

Параметр	Описание
Имя	Название модуля. По умолчанию: HUBModule. Не рекомендуется изменять.

### Параметры модуля

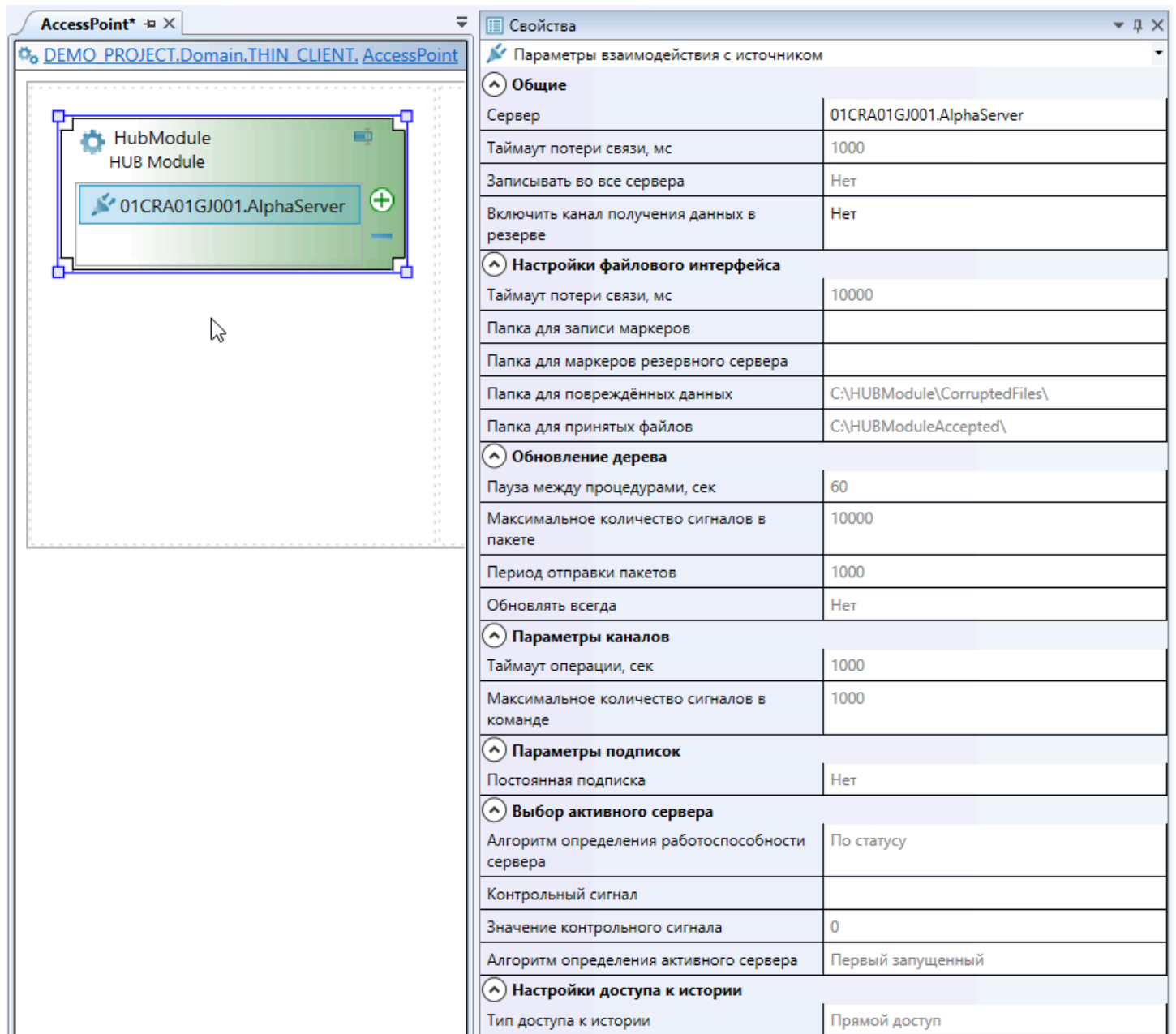
Параметр	Описание
Разрешено изменять значения сигналов	<ul style="list-style-type: none"><li>› Да – разрешено.</li><li>› Нет – запрещено.</li></ul>
Настройки по умолчанию	Идентификатор модуля в Astra.Server.
Активность	Активность модуля: <ul style="list-style-type: none"><li>› Да – модуль запущен.</li><li>› Нет – модуль остановлен.</li></ul>
Отображаемое имя	По умолчанию не заполнено.

## Параметры журналирования

Параметр	Описание
Вести журнал работы модуля	<ul style="list-style-type: none"><li>› Да – сведения о работе модуля сохраняются в журнал.</li><li>› Нет – журнал работы модуля не ведётся.</li></ul>
Размер журнала работы модуля, МБ	При достижении максимального размера создается новый файл, копия старого файла хранится на рабочем диске.
Количество дополнительных журналов работы	Количество файлов заполненных журналов работы модуля. Возможные значения: от 1 до 255.
Уровень трассировки в журнал приложений	Типы сообщений, которые фиксируются в журнал приложений: <ul style="list-style-type: none"><li>› Предупреждения и аварийные сообщения.</li><li>› Информационные сообщения.</li><li>› Отладочные сообщения.</li></ul> Вышестоящий уровень сообщений входит в состав нижестоящего.

## 1.2.1.2. Настройка источника данных

Для настройки источника данных выберите необходимый источник из списка.



The screenshot shows the configuration interface for an AccessPoint. On the left, a HubModule is visible with a data source '01CRA01GJ001.AlphaServer' selected. On the right, the 'Свойства' (Properties) window is open, displaying various configuration parameters for the data source.

Параметры взаимодействия с источником	
<b>Общие</b>	
Сервер	01CRA01GJ001.AlphaServer
Таймаут потери связи, мс	1000
Записывать во все сервера	Нет
Включить канал получения данных в резерве	Нет
<b>Настройки файлового интерфейса</b>	
Таймаут потери связи, мс	10000
Папка для записи маркеров	
Папка для маркеров резервного сервера	
Папка для повреждённых данных	C:\HUBModule\CorruptedFiles\
Папка для принятых файлов	C:\HUBModuleAccepted\
<b>Обновление дерева</b>	
Пауза между процедурами, сек	60
Максимальное количество сигналов в пакете	10000
Период отправки пакетов	1000
Обновлять всегда	Нет
<b>Параметры каналов</b>	
Таймаут операции, сек	1000
Максимальное количество сигналов в команде	1000
<b>Параметры подписок</b>	
Постоянная подписка	Нет
<b>Выбор активного сервера</b>	
Алгоритм определения работоспособности сервера	По статусу
Контрольный сигнал	
Значение контрольного сигнала	0
Алгоритм определения активного сервера	Первый запущенный
<b>Настройки доступа к истории</b>	
Тип доступа к истории	Прямой доступ



Если источник данных защищен от несанкционированного обмена данными с Astra.AccessPoint, то для подключения к источнику потребуется ввод пароля. Укажите пароль доступа к источнику во всех узлах конфигурирования каналов источника, в настройках HUB Module.

## Общие

Параметр	Описание
Сервер	Идентификатор источника данных.
Таймаут потери связи, мсек	Период времени, после превышения которого связь с источником данных считается потерянной. По умолчанию равен 1000 мсек. Отсчет таймаута начинается с момента, когда все каналы источника данных перестают отвечать на запросы Astra.AccessPoint
Записывать во все сервера	Параметр активирует запись значений сигналов в оба сервера источника данных (в активный и в резервный). ➤ <b>Нет</b> - запись производится только в активный сервер источника. ➤ <b>Да</b> - может быть полезно, когда не возможна синхронизация значений сигналов между активным и резервным серверами источника.
Включить канал получения данных в резерве	Возможные значения: ➤ <b>Нет</b> ➤ <b>Да</b>

## Настройка файлового интерфейса

Параметр	Описание
Таймаут потери связи, мсек	Значение по умолчанию: 10000.

Папка для записи маркеров	Папка, в которую будут записываться файлы-маркеры резервного сервера.
Папка для маркеров резервного сервера	Папка, из которой будут считываться файлы-маркеры резервного сервера.
Папка для поврежденных данных	В эту папку будут перемещаться поврежденные данные.
Папка для принятых данных	В эту папку будут перемещаться данные, сгенерированные ТСП-сервером.

## Обновление дерева

Группа параметров Обновление дерева отвечает за то, с какой интенсивностью Astra.AccessPoint будет опрашивать источники данных. Обновления от источников приходят в Astra.AccessPoint в виде группы пакетов, внутри которых содержатся сигналы. Ниже описаны параметры, настройка которых позволяет найти нужное сочетание скорости и нагрузки на источник данных в ходе выполнения запросов.

Параметр	Описание
Обновлять всегда	Если параметр активен, то происходит постоянное циклическое обновление дерева источника с определенными паузами. Если параметр неактивен, то конфигурация обновится единожды - при подключении Astra.AccessPoint к источнику данных. Активность этого параметра имеет смысл только при использовании нескольких Astra.AccessPoint в режиме каскадирования.
Пауза между процедурами, сек	Период простоя, перед следующим обновлением конфигурации источника. Малые паузы между



	обновлениями повышают актуальность дерева сигналов источника, но увеличивает нагрузку на источник данных за счет более частых запросов. По умолчанию – 60 секунд.
Максимальное количество сигналов в пакете	Параметр определяет вместимость каждого пакета с сигналами дерева. Большие пакеты сигналов ускорят процесс обновления дерева, но повысят нагрузку на сервер-источник. По умолчанию 10000 сигналов в пакете.
Период отправки пакетов	Время, отведенное источнику, за которое он должен отправить один пакет с сигналами. Малое значение параметра ускорит процесс обмена данными, но увеличит нагрузку на сервер за счет более интенсивной отправки пакетов. Параметр измеряется в миллисекундах. По умолчанию 1000 мсек.

## Параметры каналов

Каждый сервер источника данных может содержать несколько каналов связи. В целях резервирования и повышения надежности каналы работают по разным физическим линиям передачи данных и имеют разные сетевые адреса. IP-адреса каналов и их порты задаются на стороне источника данных.

Параметр	Описание
Таймаут операции, мсек	Период времени в миллисекундах после которого связь с каналом считается потерянной. Отсчет таймаута начинается с момента, когда источник перестает отвечать на запросы Astra.AccessPoint по данному каналу связи. По умолчанию: 1000 мсек.
Максимальное количество сигналов в команде	Параметр ограничивает максимальное количество сигналов, на которые можно подписаться в рамках одной команды.

## Параметры подписок

Параметр	Описание
Постоянная подписка	<p>Определяет момент подписки на сигналы из ветви адресного пространства источника, указанной в параметре Папка источник:</p> <ul style="list-style-type: none"> <li>➤ Да - подписка на сигналы происходит при запуске модуля HUB;</li> <li>➤ Нет - подписка на сигналы происходит только когда к Astra.AccessPoint подключается клиент.</li> </ul>

## Выбор активного сервера

Параметр	Описание
Алгоритм определения работоспособности сервера	<p>Способ проверки работоспособности серверов в составе источника:</p> <ul style="list-style-type: none"> <li>➤ <b>По статусу</b> - по ответу на запрос к серверу специальным методом GetStatus. Значение dwServerState = OPC_STATUS_RUNNING означает, что сервер находится в работе.</li> <li>➤ <b>По качеству</b> - по качеству контрольного сигнала. Если качество контрольного сигнала &lt; 192, то связь с сервером потеряна. Если качество сигнала равно 192, то сервер работоспособен.</li> <li>➤ <b>По значению</b> - по значению контрольного сигнала. Указывается значение сигнала, при котором считается, что сервер работоспособен. При других значениях сигнала связь с источником потеряна.</li> </ul>
Контрольный сигнал	<p>Параметр предназначен для указания тега сигнала, который будет считаться контрольным для определения работоспособности сервера.</p>

Значение контрольного сигнала	Параметр предназначен для указания значения контрольного сигнала, при котором сервер считается работоспособным.
Алгоритм определения активного сервера	<ul style="list-style-type: none"> <li>➤ <b>Любой работоспособный.</b> При потере связи модуль пытается установить соединение с любым работоспособным сервером в составе источника.</li> <li>➤ <b>Первый по порядку.</b> Модуль работает с первым сервером в приоритетном порядке. При потере связи с первым сервером, модуль установит соединение с любым работоспособным сервером в составе источника, но не прекратит попыток восстановить связь с первым сервером. Как только связь с первым сервером будет восстановлена, модуль возобновит с ним работу. Чтобы определить порядок серверов, воспользуйтесь стрелками в окне Серверы.</li> <li>➤ <b>Первый запущенный.</b> Модуль выбирает для работы тот сервер в составе источника, время запуска которого было раньше.</li> </ul>

## Параметры доступа к истории

Параметр	Описание
Тип доступа к истории	Тип доступа к истории: <ul style="list-style-type: none"> <li>➤ Прямой доступ.</li> <li>➤ Удаленный доступ.</li> </ul>

## 1.2.2. Получение данных

[Получение данных по OPC DA](#)

[Получение данных по OPC AE](#)

[Получение данных по OPC UA](#)

## 1.2.2.1. Получение данных по OPC DA



Спецификация OPC DA базируются на COM/DCOM и используются только в ОС Windows.

После того, как Astra.AccessPoint был сконфигурирован, подключитесь к нему любым OPC DA клиентом.



ProgID для подключения: Astra.OPCDAserver.AccessPoint.

## 1.2.2.2. Получение данных по OPC AE



Спецификация OPC AE базируются на COM/DCOM и используются только в ОС Windows.

Astra.AccessPoint может выступать в качестве единой точки приема событий технологического процесса, генерируемых на подключенных источниках данных. Для этого добавьте в состав конфигурации Astra.AccessPoint модуль OPC AE Server и активируйте его.

После этого вы сможете отслеживать события всех источников данных, подключившись к Astra.AccessPoint любым OPC AE-клиентом.



ProgID для подключения: Astra.OPCAEServer.AccessPoint.

## 1.2.2.3. Получение данных по OPC UA

Чтобы получить доступ к оперативным или историческим данным Astra.AccessPoint по спецификации OPC UA, добавьте в конфигурацию модуль OPC UA Server и произведите его настройку.

После настройки модуля OPC UA возможно подключение к Astra.AccessPoint любым OPC UA клиентом, например UA Expert.



Порт подключения по умолчанию: 62544.

[Получение оперативных данных](#)

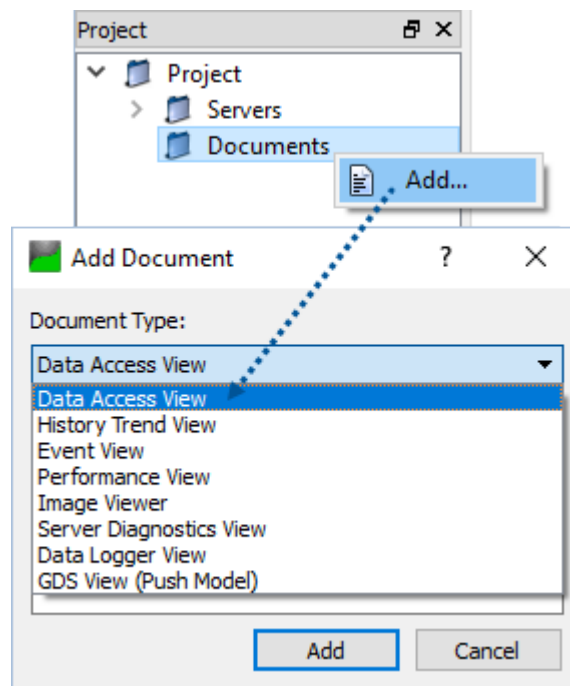
[Получение исторических данных](#)

## 1.2.2.3.1. Получение оперативных данных

### Получение оперативных значений сигналов

Чтобы получать оперативные значения сигналов, выполните следующие действия:

1. Создайте в дереве проекта элемент типа Documents - Data Access View.



2. Перетащите сигналы, значения которых вы хотите получать, из адресного пространства сервера на вкладку Data Access View. Значения добавленных сигналов будут обновляться автоматически.



Unified Automation UaExpert - The OPC Unified Architecture Client - NewProject\*

File View Server Document Settings Help

Project Data Access View

#	Node Id	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Statuscode
1	NS1 String Items.Level	Level	15	SByte	14:11:30.014	14:11:30.014	GoodLocalOverride
2	NS1 String Items.Pressure	Pressure	220	Byte	14:11:34.165	14:11:34.165	GoodLocalOverride

Address Space

No Highlight

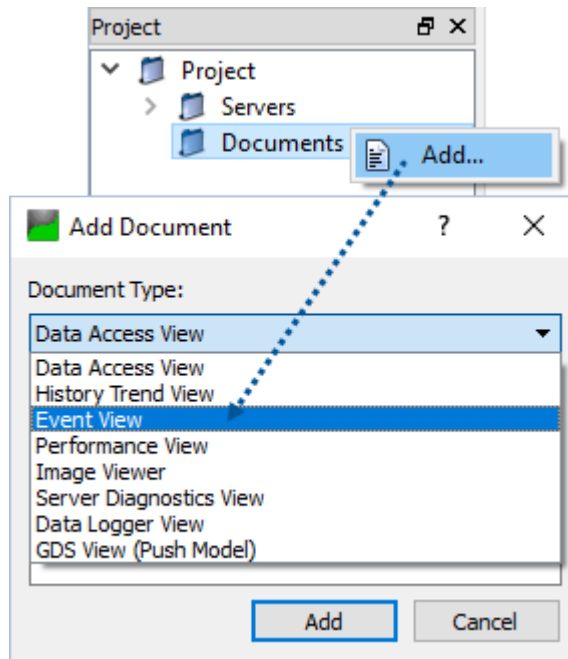
Root

- Objects
  - Items
    - Level
    - Pressure
  - Server
  - Service
  - Types
  - Views

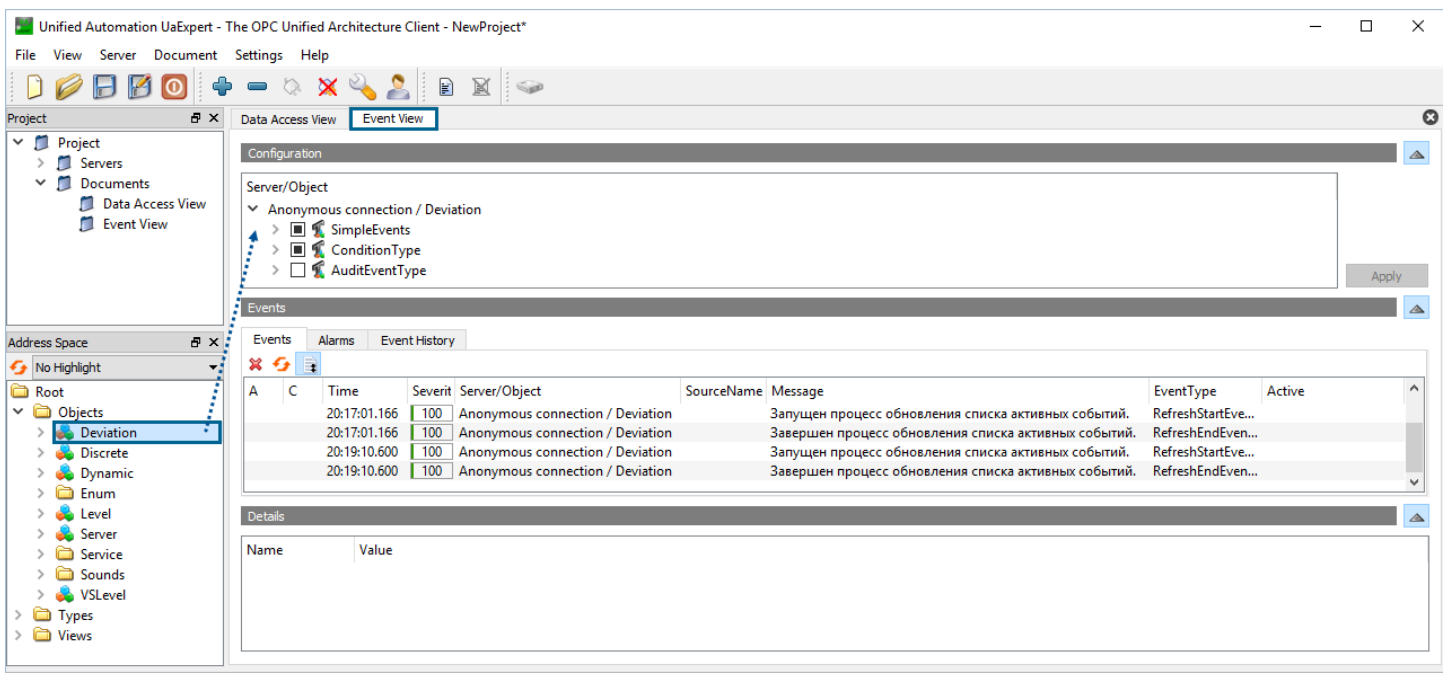
# Получение оперативных событий

Чтобы получать оперативные события, выполните следующие действия:

1. Создайте в дереве проекта элемент типа Documents - Event View.



2. Перетащите из адресного пространства сервера на вкладку Event View сигналы, события по которым вы хотите получать.



Чтобы получать все события, происходящие на сервере, перетащите элемент Server из адресного пространства сервера на вкладку Event View.

The screenshot shows the Unified Automation UaExpert software interface. The main window is titled "Unified Automation UaExpert - The OPC Unified Architecture Client - NewProject\*". The interface includes a menu bar (File, View, Server, Document, Settings, Help), a toolbar, and several panels:

- Project Panel:** Shows a tree view with "Project", "Servers", "Documents", "Data Access View", and "Event View".
- Address Space Panel:** Shows a tree view with "Root", "Objects", "Deviation", "Discrete", "Dynamic", "Enum", "Level", "Server", "Service", "Sounds", "VSLLevel", "Types", and "Views". The "Server" folder is highlighted.
- Configuration Panel:** Shows "Server/Object" configuration with a tree view containing "Anonymous connection / Server", "SimpleEvents", "ConditionType", and "AuditEventType".
- Events Panel:** Shows a table of events with columns: A, C, Time, Severit, Server/Object, SourceName, Message, EventType, and Active. The table contains several rows of event data.
- Details Panel:** Shows the details of the selected event, including "Name", "Value", "EventId", "EventType", "NamespaceIndex", "IdentifierType", and "Identifier".

The Events table data is as follows:

A	C	Time	Severit	Server/Object	SourceName	Message	EventType	Active
		20:17:01.166	100	Anonymous connection / Deviation		Завершен процесс обновления списка активных событий.	RefreshEndEven...	
		20:19:10.600	100	Anonymous connection / Deviation		Запущен процесс обновления списка активных событий.	RefreshStartEve...	
		20:19:10.600	100	Anonymous connection / Deviation		Завершен процесс обновления списка активных событий.	RefreshEndEven...	
		9:06:56.855	100	Anonymous connection / Deviation		Запущен процесс обновления списка активных событий.	RefreshStartEve...	
		9:06:56.858	100	Anonymous connection / Deviation		Завершен процесс обновления списка активных событий.	RefreshEndEven...	

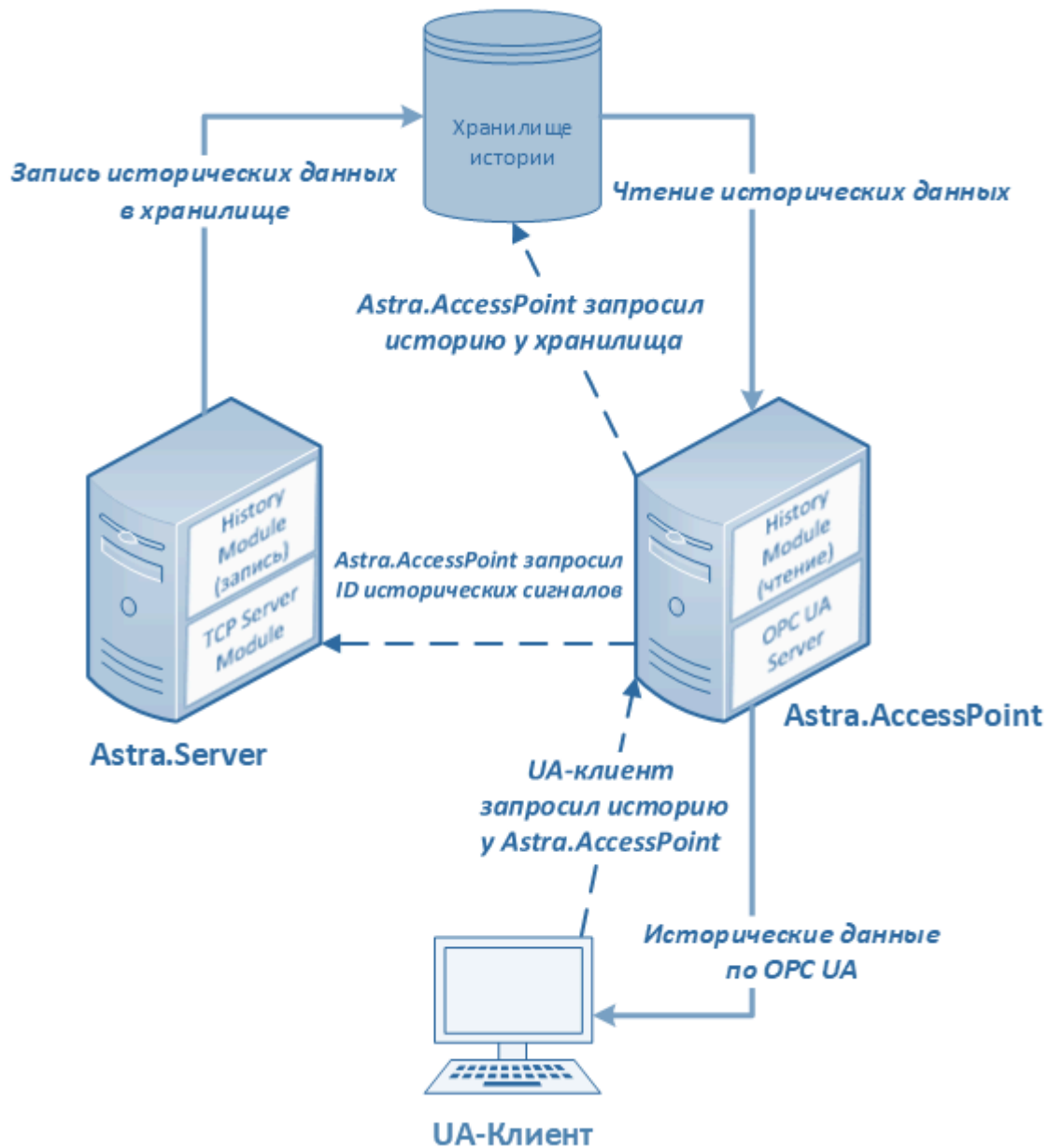
## 1.2.2.3.2. Получение исторических данных

Для предоставления исторических данных по спецификации OPC UA Astra.AccessPoint должен включать модули:

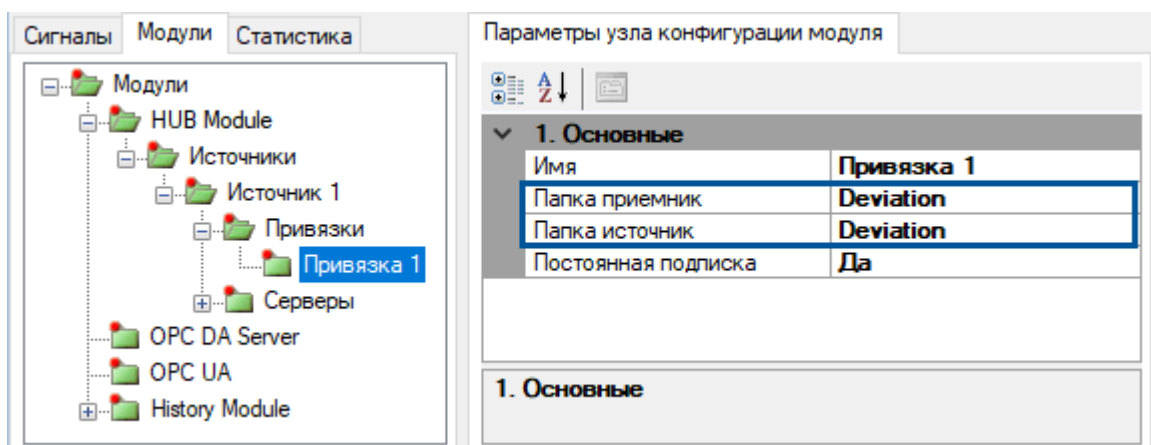
- › модуль истории (History Module) - выполняет выборку истории из хранилища по определенным id сигналов;
- › модуль OPC UA Server - предоставляет исторические данные клиентам по спецификации OPC UA.



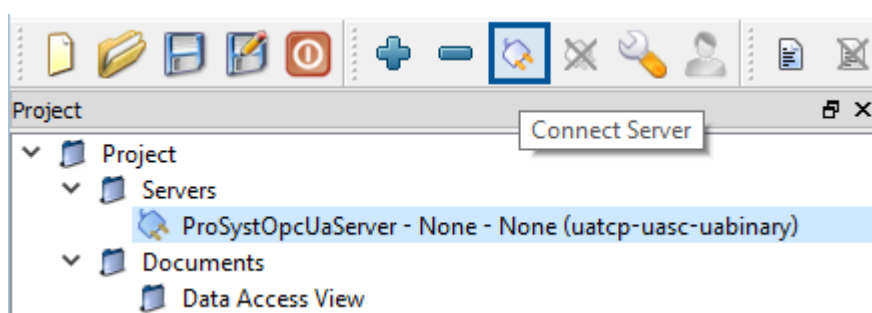
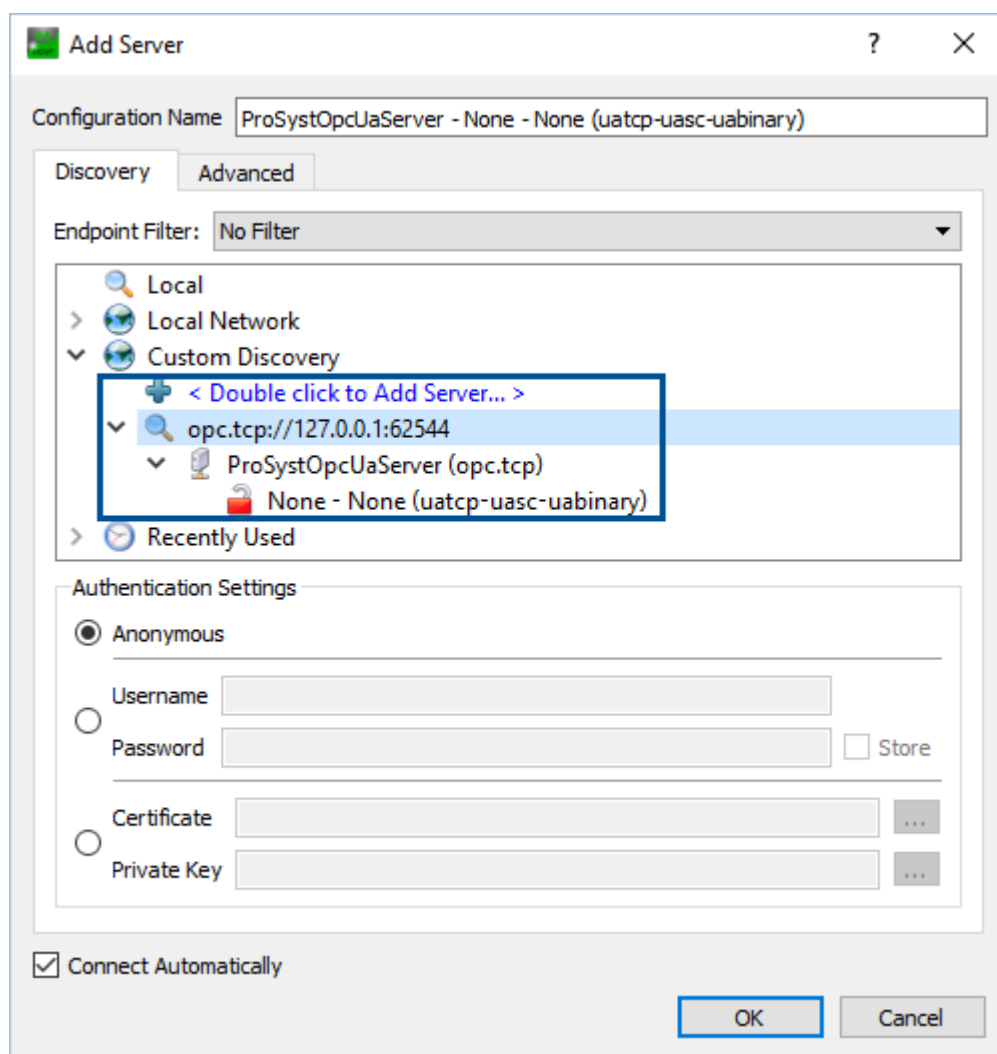
Конфигурация источника должна включать модуль TCP Server Module - для предоставления ID сигналов, по которым требуется выборка истории.



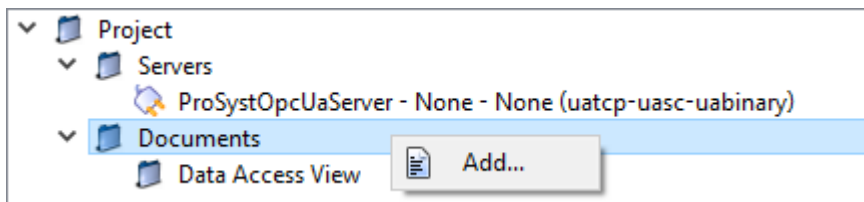
В узле Источники модуля HUB Module необходимо создать источник данных. В узле источника данных настройте параметры привязки Папка источник и Папка приемник. Привязка должна загружать в Astra.AccessPoint все узлы источника, по которым имеются исторические данные.



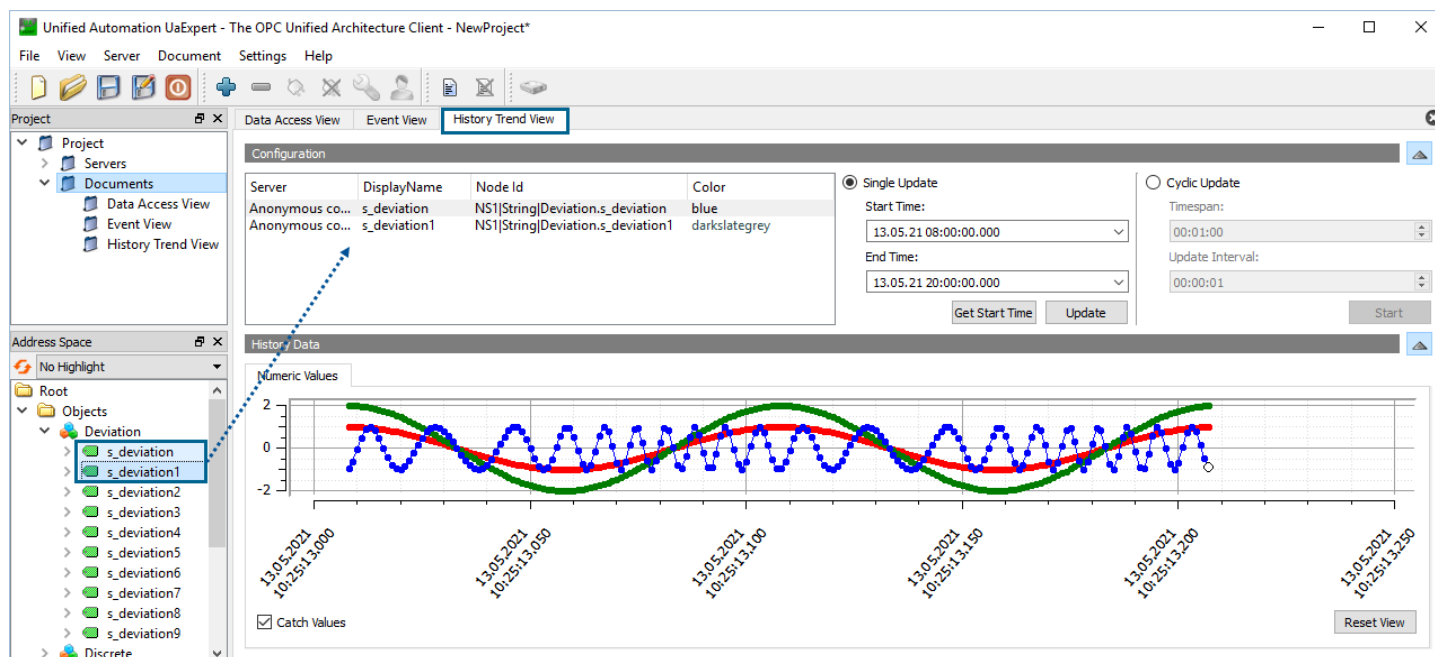
После произведенных настроек подключитесь к Astra.AccessPoint любым OPC UA клиентом (например, UA Expert) и запросите историю изменений значений сигналов.



Для работы с историческими данными создайте в дереве проекта элемент типа Documents - History Trend View.



Для работы с историческими данными перетаскивайте необходимые сигналы из дерева Astra.AccessPoint на вкладку History Trend View. В области Single Update задайте начальное и конечное время для построения графика истории и нажмите Update.



Чтобы просмотреть информацию по истории изменений добавленного сигнала в табличной форме, перейдите на соседнюю вкладку с именем добавленного сигнала.

Unified Automation UaExpert - The OPC Unified Architecture Client - NewProject\*

File View Server Document Settings Help

Project: Project > Servers > Documents > Data Access View > Event View > History Trend View

Address Space: No Highlight > Root > Objects > Deviation > s\_deviation > 5000 > 6000 > 8001 > s\_deviation1 > s\_deviation2 > s\_deviation3 > s\_deviation4 > s\_deviation5 > s\_deviation6 > s\_deviation7

**Configuration**

Server	DisplayName	NodeId	Color
Anonymous co...	s_deviation	NS1 String Deviation.s_deviation	blue
Anonymous co...	s_deviation1	NS1 String Deviation.s_deviation1	red

Single Update  
 Start Time: 13.05.21 08:00:00.000  
 End Time: 13.05.21 20:00:00.000  
 [Get Start Time] [Update]

Cyclic Update  
 Timespan: 00:01:00  
 Update Interval: 00:00:01  
 [Start]

**History Data**

Numeric Values: ■ NS1|String|Deviation.s\_deviation ■ NS1|String|Deviation.s\_deviation1

Source Timestamp	Server Timestamp	Value	Status
2021-05-13 17:07:28.213	2021-05-13 17:07:28.213	2	GoodLocalOverride (0x00960000)
2021-05-13 17:07:21.773	2021-05-13 17:07:21.773	0	GoodLocalOverride (0x00960000)
2021-05-13 17:07:17.235	2021-05-13 17:07:17.235	4	GoodLocalOverride (0x00960000)
2021-05-13 17:05:56.880	2021-05-13 17:05:56.880	2	GoodLocalOverride (0x00960000)
2021-05-13 17:05:52.215	2021-05-13 17:05:52.215	5	GoodLocalOverride (0x00960000)
2021-05-13 17:05:46.320	2021-05-13 17:05:46.320	3	GoodLocalOverride (0x00960000)



## 1.2.2.4. Квитирование событий

Когда пользователь квитирует событие в подключённом к Astra.AccessPoint клиенте, Astra.AccessPoint передаёт информацию о квитировании события в источник данных, в котором оно произошло.



Если источником данных является пара серверов, работающих в режиме горячего резервирования, то информация о квитировании передаётся в оба сервера.

## 1.2.3. Диагностика работы

Для диагностики работы Astra.AccessPoint и его модулей воспользуйтесь сервисными приложениями Статистика и сервисным приложением Просмотрщик лога кадров.



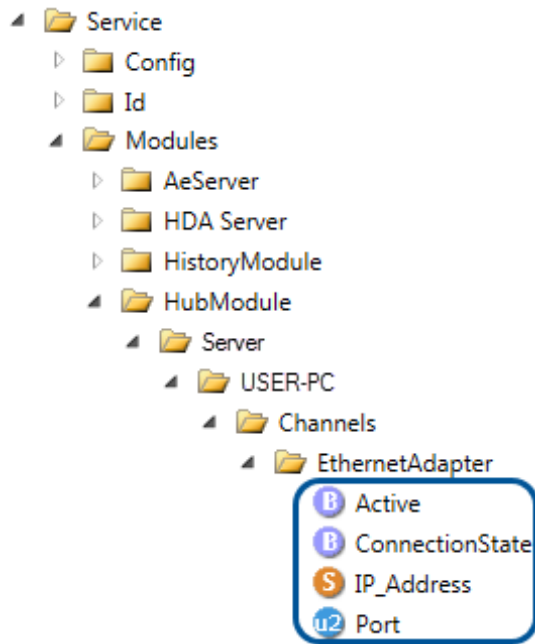
Порт подключения: 4976.

Для мониторинга состояния связи с серверами источника по разным каналам в Astra.AccessPoint используются сервисные сигналы в папке:




Service.Modules.HUB Module.Source N.Server N.Channels.ChannelN

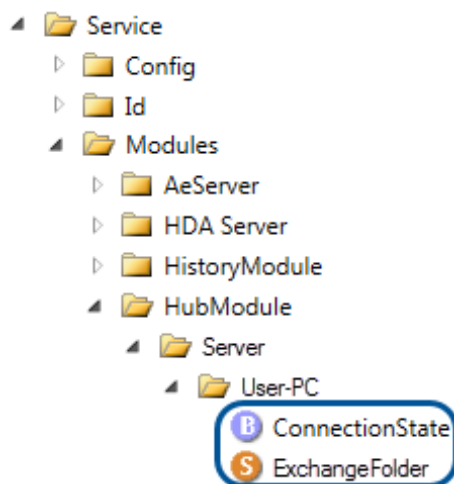
Тег сигнала	Тип	Описание
Active	bool	Активность канала
ConnectionState	bool	Состояние связи по каналу
IP_Address	string	IP-адрес канала
Port	uint2	Порт канала



Для мониторинга состояния файлового интерфейса используются сервисные сигналы в папке:

 Service.Modules.HUB Module.Sources.Источник N

Тег сигнала	Тип	Описание
ConnectionState	bool	Состояние связи с источником
ExchangeFolder	string	Путь к папке обмена файлами



## 1.3. Astra.Imitator

**Astra.Imitator** – программный компонент, предназначенный для просмотра истории технологического процесса посредством мнемосхем.

Astra.Imitator загружает исторические данные процесса за указанный период времени и проигрывает их в виде потока оперативных данных, что позволяет пользователю просмотреть историю хода технологического процесса на мнемосхемах.



Astra.Server получает данные от нижестоящих систем и записывает значения в Astra.Historian. Astra.Imitator считывает исторические данные из Astra.Historian и воспроизводит историю технологического процесса за определенный промежуток времени, передавая исторические данные клиенту в виде потока оперативных данных.

## 1.3.1. Настройка конфигурационных файлов

- › [Astra.Historian](#)
- › [Astra.Domain](#)

## 1.3.1.1. Astra.Historian

Для воспроизведения истории с помощью Astra.Imitator необходимо настроить сохранение данных Astra.Server в отдельную базу Astra.Historian, из которой Astra.Imitator будет получать данные для воспроизведения истории. Такая база данных называется имитационной.

Чтобы создать имитационную базу данных в Astra.Historian:

1. Откройте файл конфигурации Astra.Historian.Server.xml.
2. В элемент Bases добавьте дочерний элемент DB\_IMIT.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <Astra.Historian.Server StatPort="3388" DefaultPrimaryDir="c:\Historian\Databases" MaxPoolMemorySize="512" EnableDCOM="0">
3   <tcp-server default-port=4949 idle-sessions-count=1 idle-sessions-timeout=15>
4     <server-endpoint host="0.0.0.0" />
5   </tcp-server>
6   <Bases>
7     <Base Alias="DB_EVENTS" PreferredCommonCacheLimit="128" ActiveStorageDepth="3" StorageDepth="365" VolumeLimit="2000"/>
8     <Base Alias="DB_HISTORY" PreferredCommonCacheLimit="128" ActiveStorageDepth="3" StorageDepth="365" VolumeLimit="2000"/>
9     <Base Alias="DB_IMIT" PreferredCommonCacheLimit="128" ActiveStorageDepth="3" StorageDepth="365" VolumeLimit="2000"/>
10  </Bases>
11 </Astra.Historian.Server>
12
13
```

3. Сохраните изменения в файле Astra.Historian.Server.xml.
4. Перезапустите службу Astra.Historian.Server.

## 1.3.1.2. Astra.Domain

### Astra.Domain

Чтобы применить конфигурацию на Astra.Imitator необходимо добавить его в конфигурацию Astra.Domain.

Для добавления Astra.Imitator в конфигурацию Astra.Domain выполните следующие действия:

1. Перейдите в папку по следующему пути:



C:\Program Files\AstraRegul\Astra.Domain

2. Откройте файл astra.domain.agent.xml при помощи редактора Notepad++.

3. В элемент InstalledComponents добавьте дочерний элемент Astra.Server со следующими атрибутами:

- › Name – InstalledImitator;
- › ServiceName – имя службы Astra.Imitator;
- › DefaultActivation – 1.

```
*C:\Program Files\AstraRegul\Astra.Domain\astra.domain.agent.xml - Notepad++ [Administrator]
Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?

astra.domain.agent.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <Astra.Domain.Agent Name="NDA">
3   <EntryPointNetAgent Name="AstraRegul" Address="127.0.0.1" Port="1010" />
4   <InstalledComponents>
5     <Astra.Server Name="InstalledServer" ServiceName="Astra.Server" DefaultActivation="1" />
6     <Astra.Server Name="InstalledAccessPoint" ServiceName="Astra.AccessPoint" DefaultActivation="1" />
7     <Astra.Server Name="InstalledImitator" ServiceName="Astra.Imitator" DefaultActivation="1" />
8   </InstalledComponents>
```

4. В элемент Server добавьте добавьте компонент с атрибутами:

- › InstalledName – значение атрибута Name элемента InstalledComponents, добавленного на предыдущем шаге;
- › Name – имя Astra.Imitator, используемое в Astra.Astudio.
- › StorageLimitSize - 0;
- › StorageLimitNum - 0.

```
15   <ConfigurationCache Path="c:\DomainStorage\cache" />
16 </Domain>
17 <Server>
18   <Components StoragePath="c:\DomainStorage\cache\server">
19     <Component InstalledName="InstalledServer" Name="AstraServer" StorageLimitSize="0" StorageLimitNum="0" />
20     <Component InstalledName="InstalledAccessPoint" Name="AccessPoint" StorageLimitSize="0" StorageLimitNum="0" />
21     <Component InstalledName="InstalledImitator" Name="Imitator" StorageLimitSize="0" StorageLimitNum="0" />
22   </Components>
23 </Server>
24 <Options LoggerLevel="2" />
25 </Astra.Domain.Agent>
```

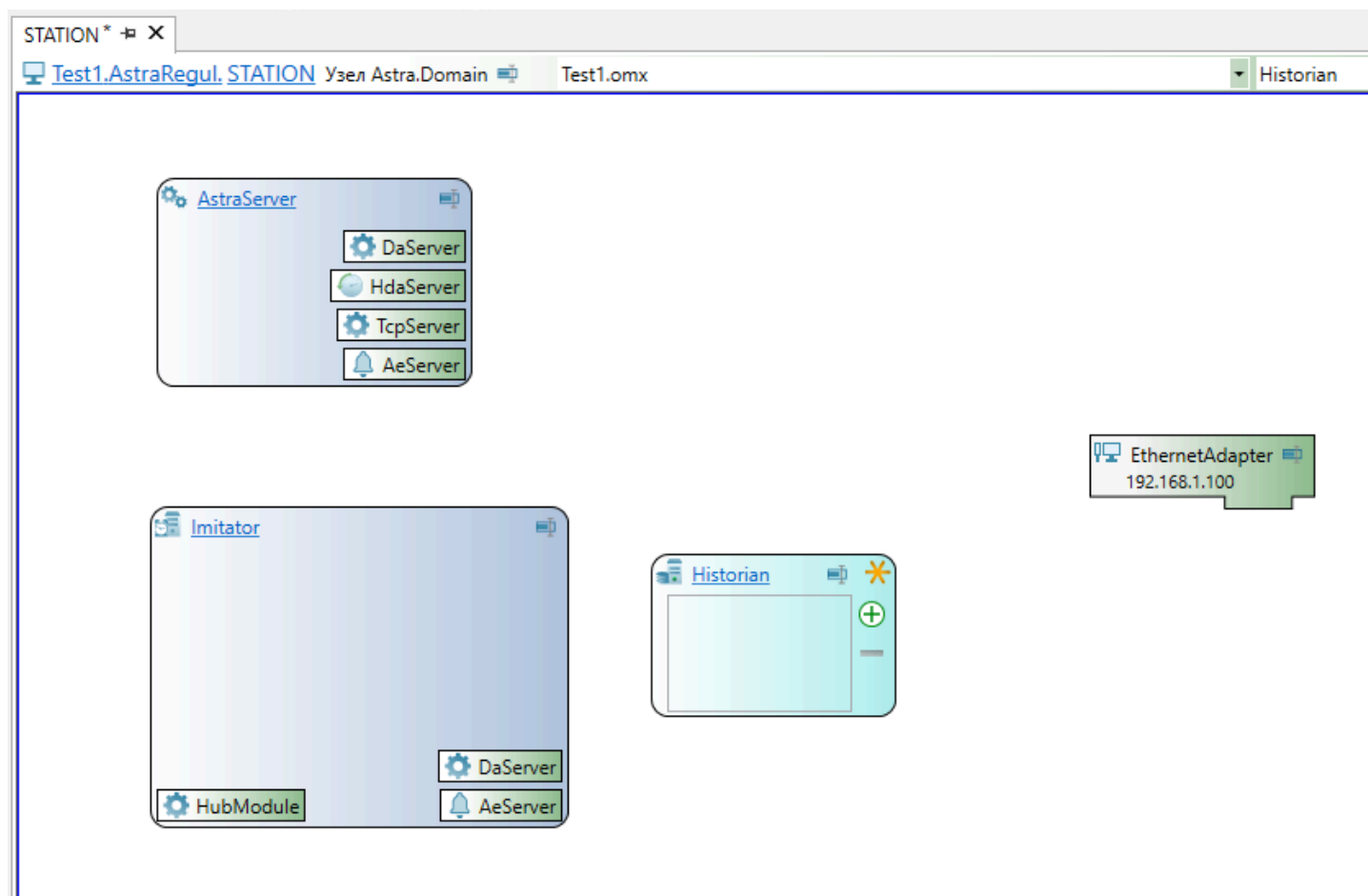
5. Сохраните файл astra.domain.agent.xml и перезапустите службы Astra.Domain.Agent и Astra.Net.Agent.



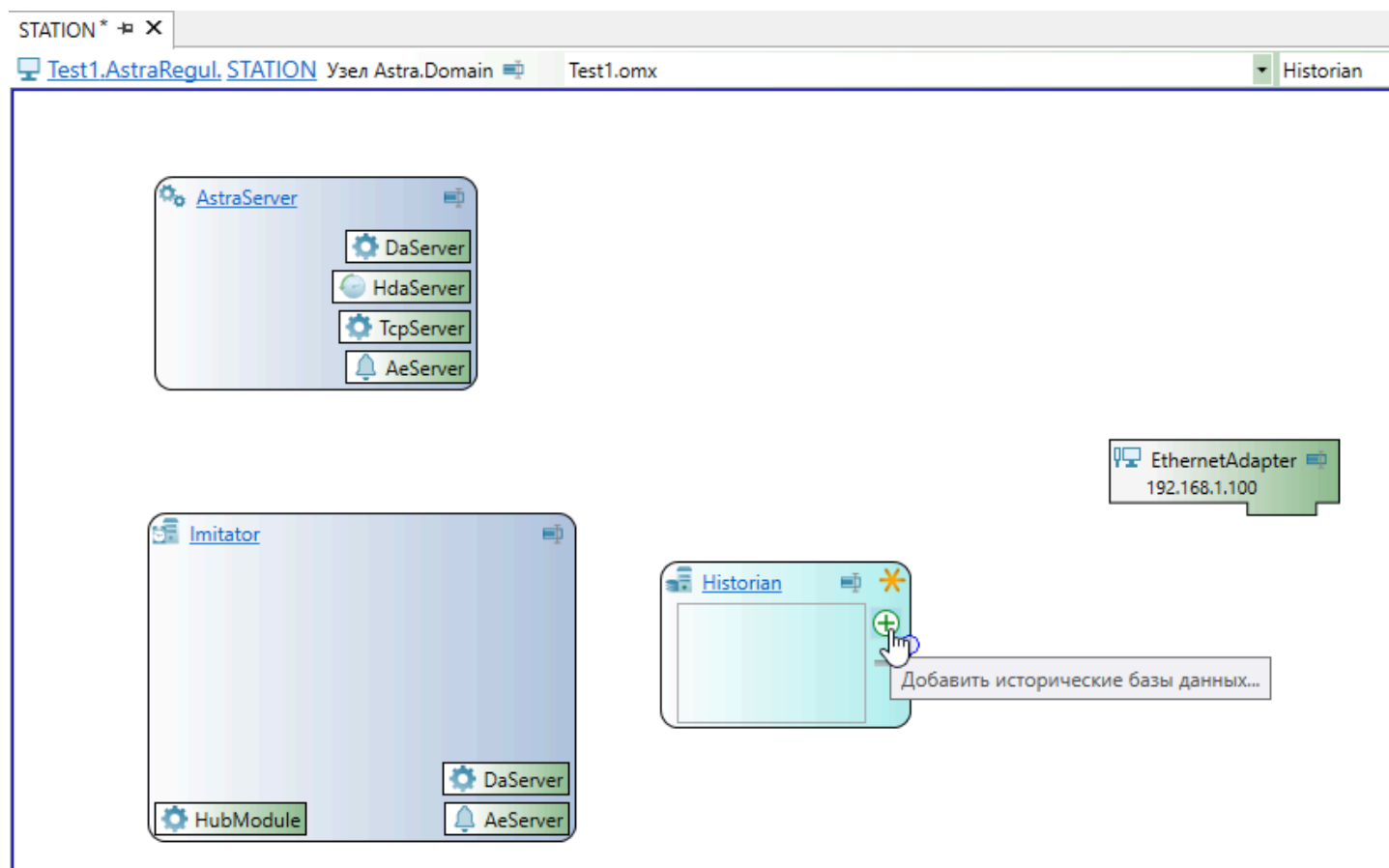
## 1.3.2. Добавление Astra.Imitator в проект

Чтобы добавить Astra.Imitator в проект Astra.AStudio необходимо выполнить следующие действия:

1. Перейдите в узел Инженерной станции и добавьте элемент Astra.Imitator из панели элементов

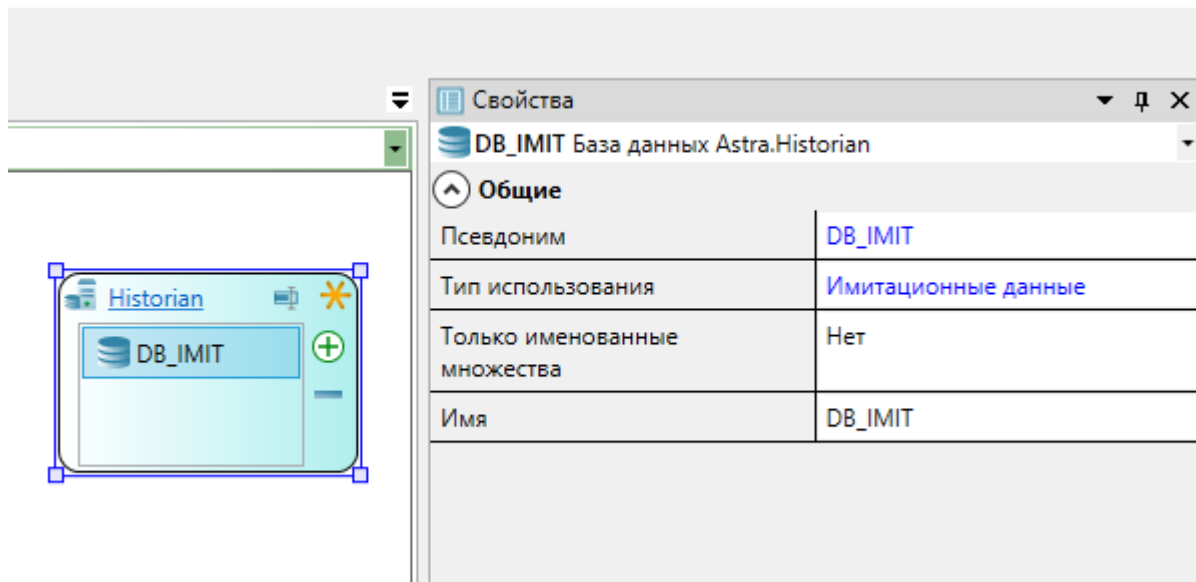


2. Добавьте новую базу данных Astra.Historian нажав на кнопку "Добавить исторические базы данных".

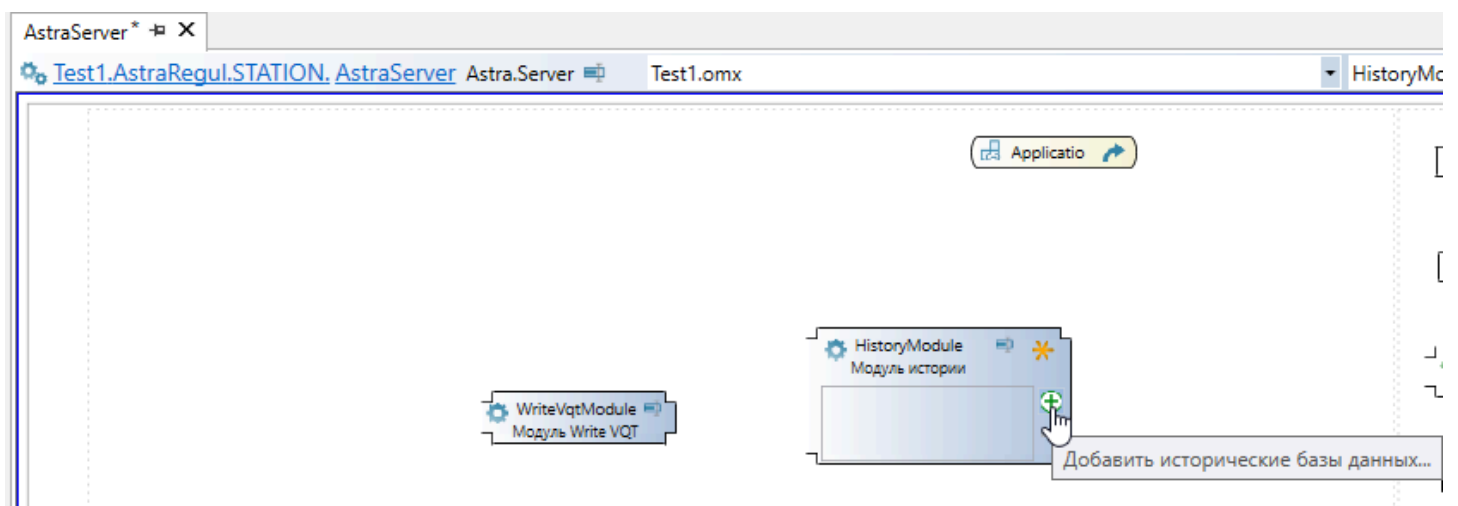


3. Выделите добавленную базу данных и укажите следующие свойства:

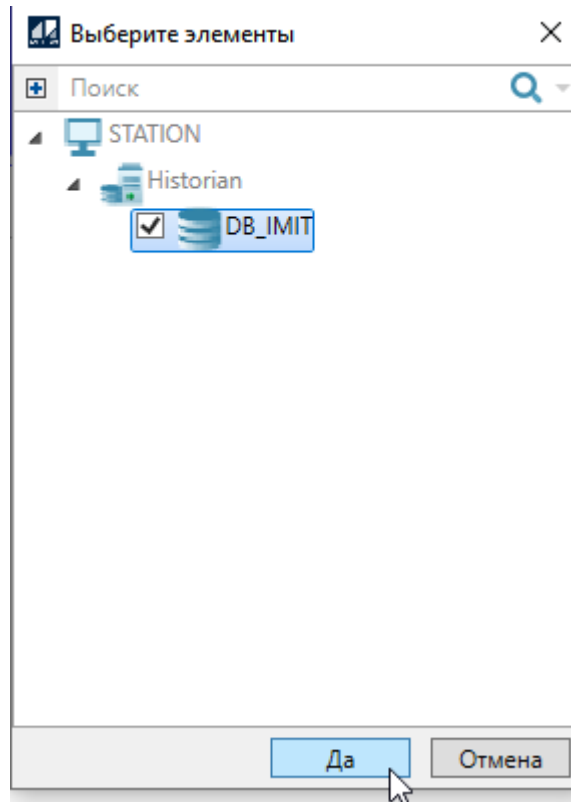
- Псевдоним - имя имитационной базы данных, указанной в настройках Astra.Historian - DB\_IMIT».
- Тип использования - «Имитационные данные».
- Имя - DB\_IMIT».



4. Перейдите в элемент Astra.Server и в модуле "Модуль истории" нажмите на кнопку добавления базы данных в которую будет записываться история.

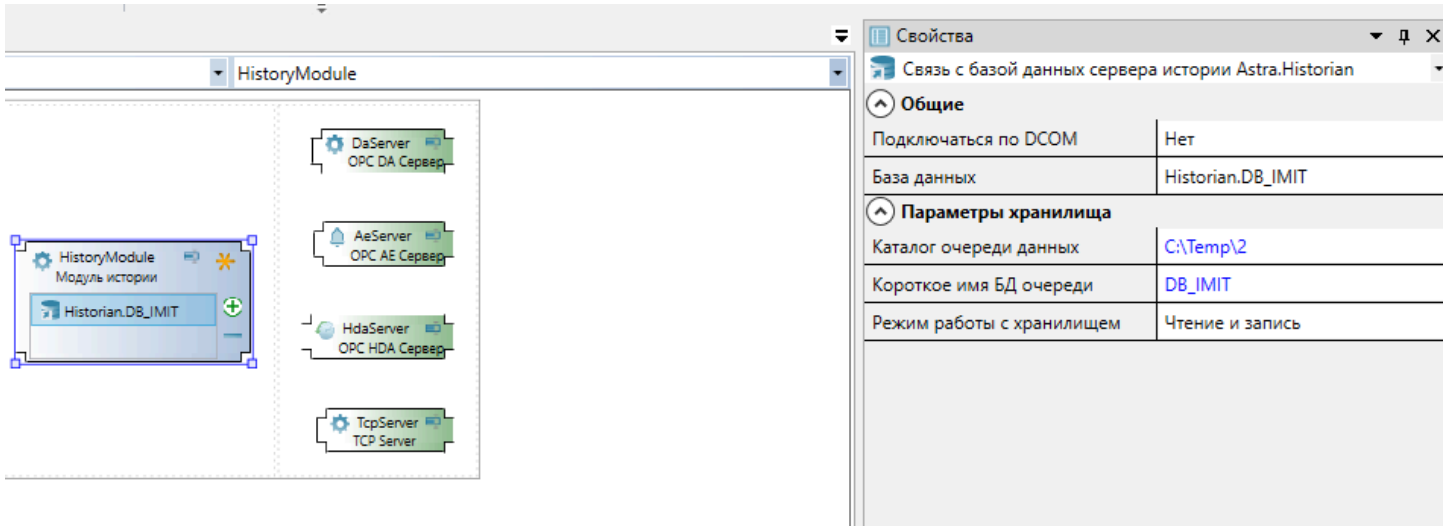


5. В открывшемся окне выберите базу данных DB\_IMIT и нажмите кнопку "Да".

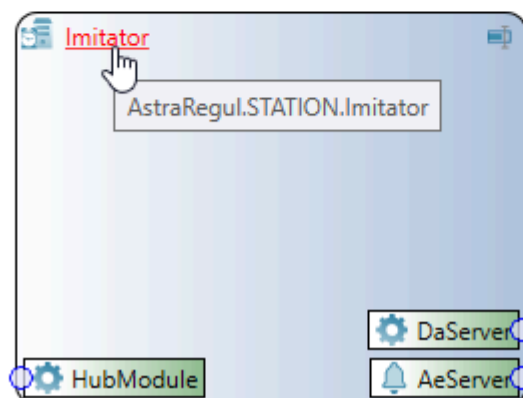
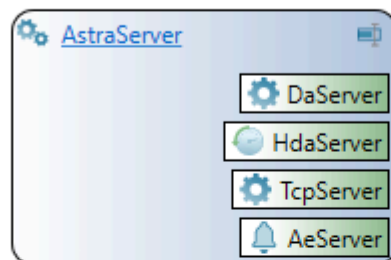
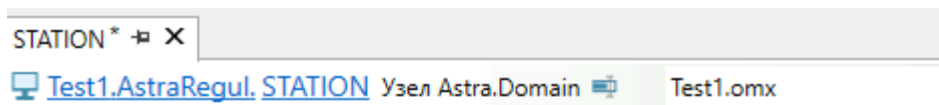



6. Выделите добавленную базу данных и укажите следующие свойства:

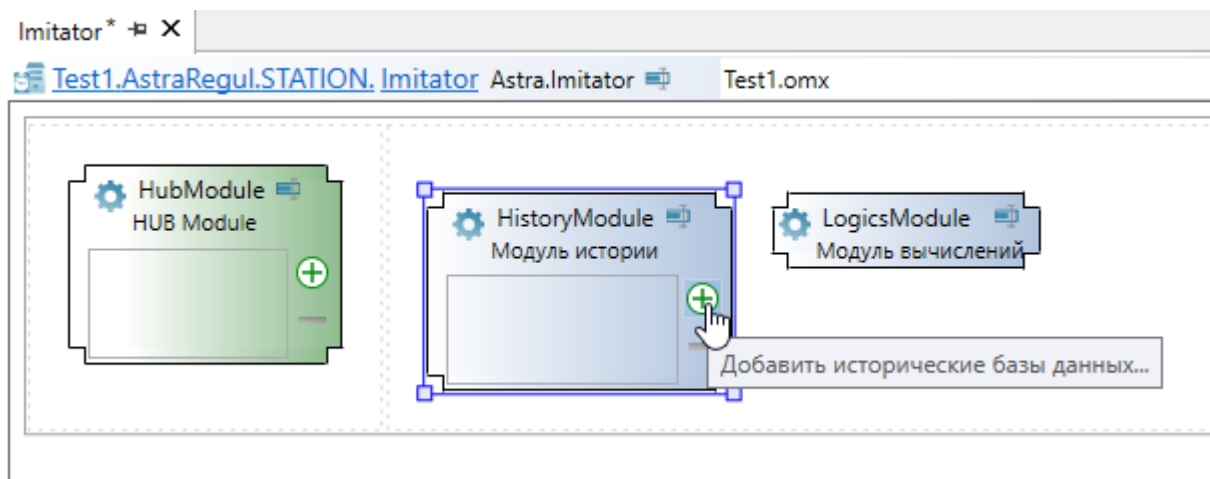
- › Короткое имя БД очереди - имя имитационной базы данных, в которую будет записываться история - DB\_IMIT»;
- › Режим работы с хранилищем - «Чтение и запись».



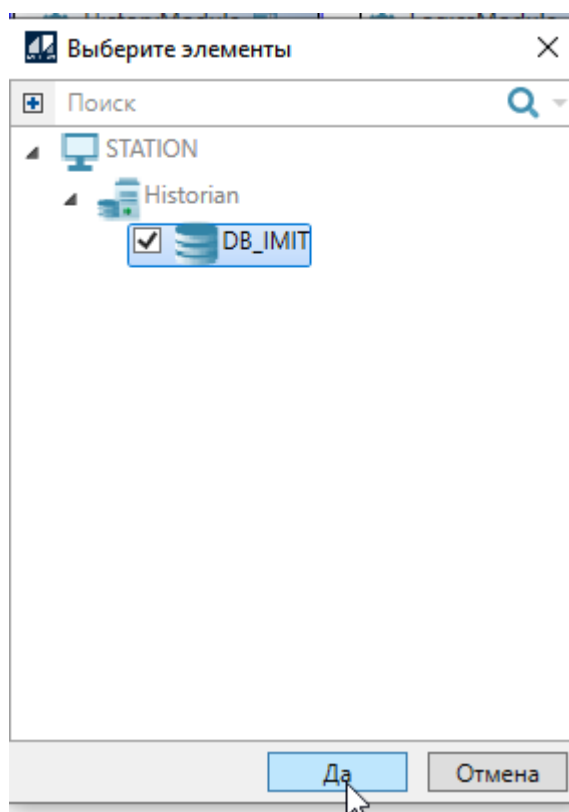
7. Вернитесь в узел Инженерной станции и перейдите в элемент Astra.Imitator.



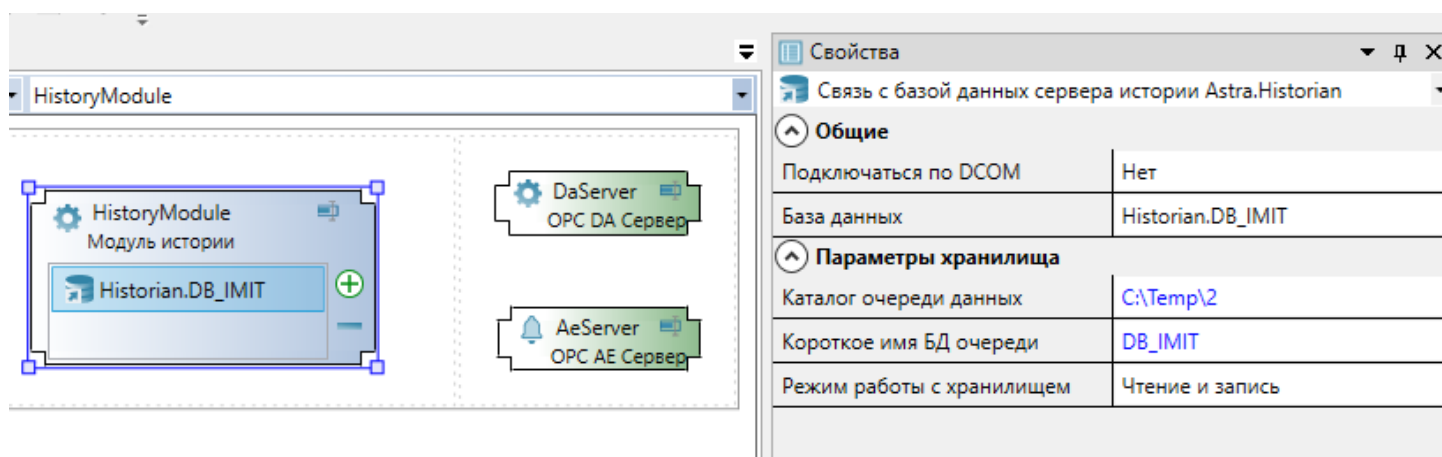
8. На элементе "Модуль истории" нажмите кнопку  для добавления базы данных, из которой будет загружаться исторические данные для дальнейшего воспроизведения.



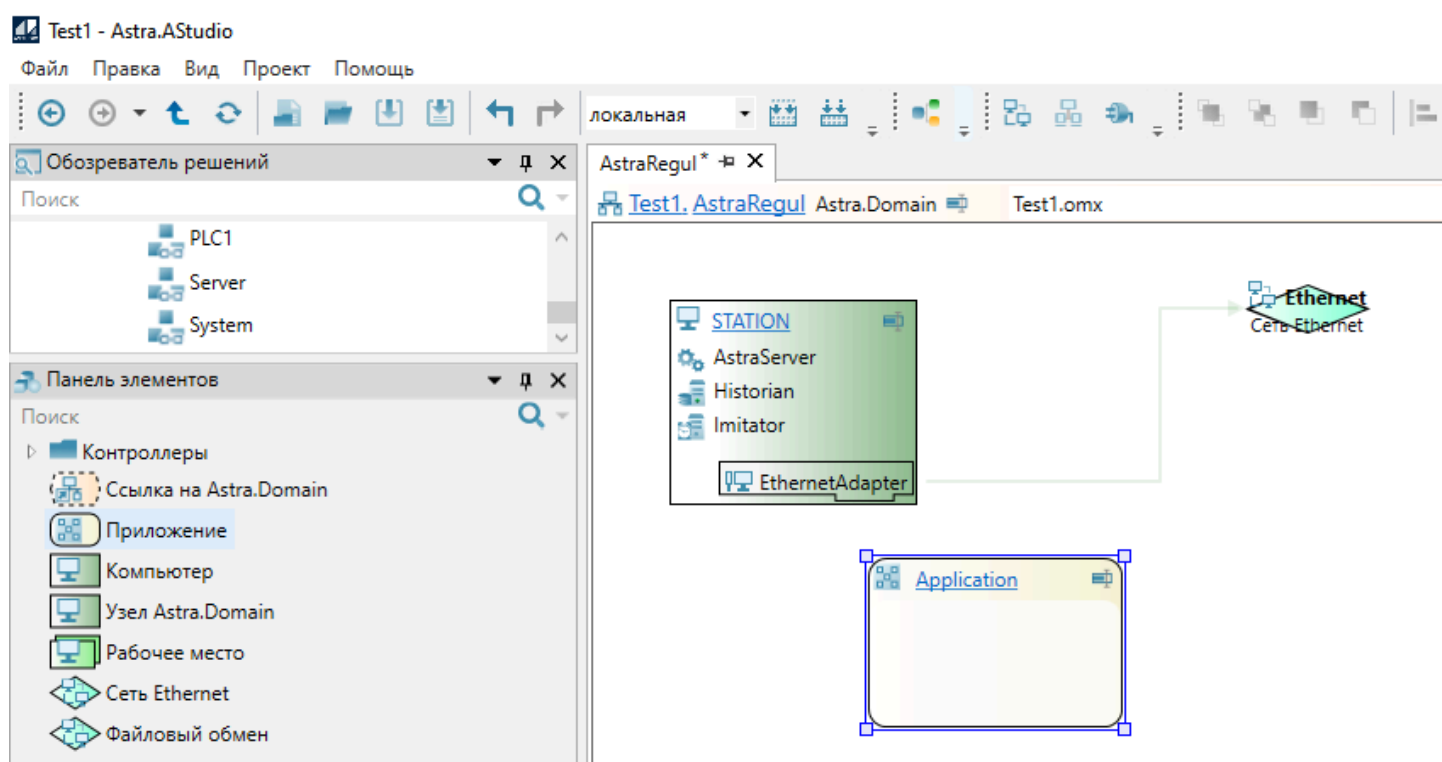
9. В открывшемся окне выберите базу данных DB\_IMIT и нажмите кнопку "Да".



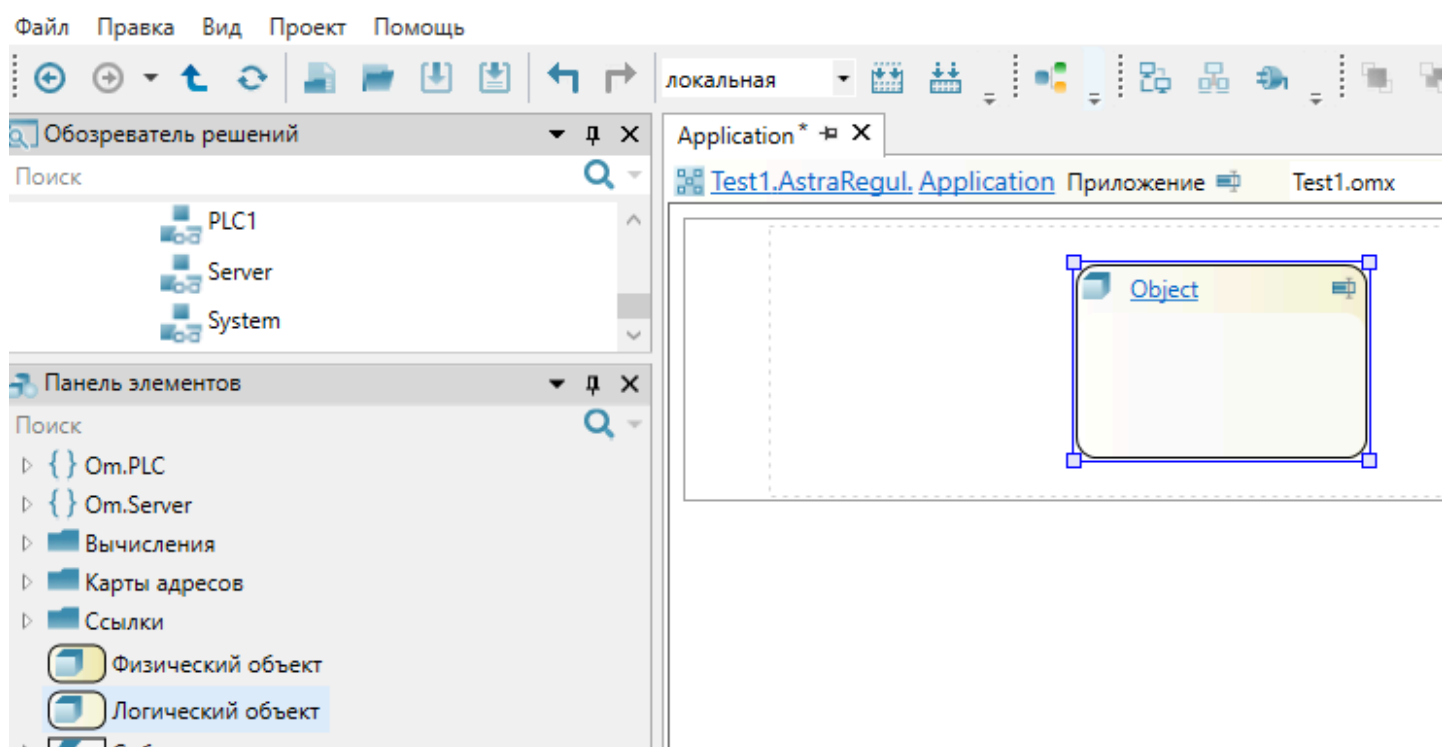
10. Для добавленной базы данных выполните те же настройки, что и для базы данных модуля истории Astra.Server.



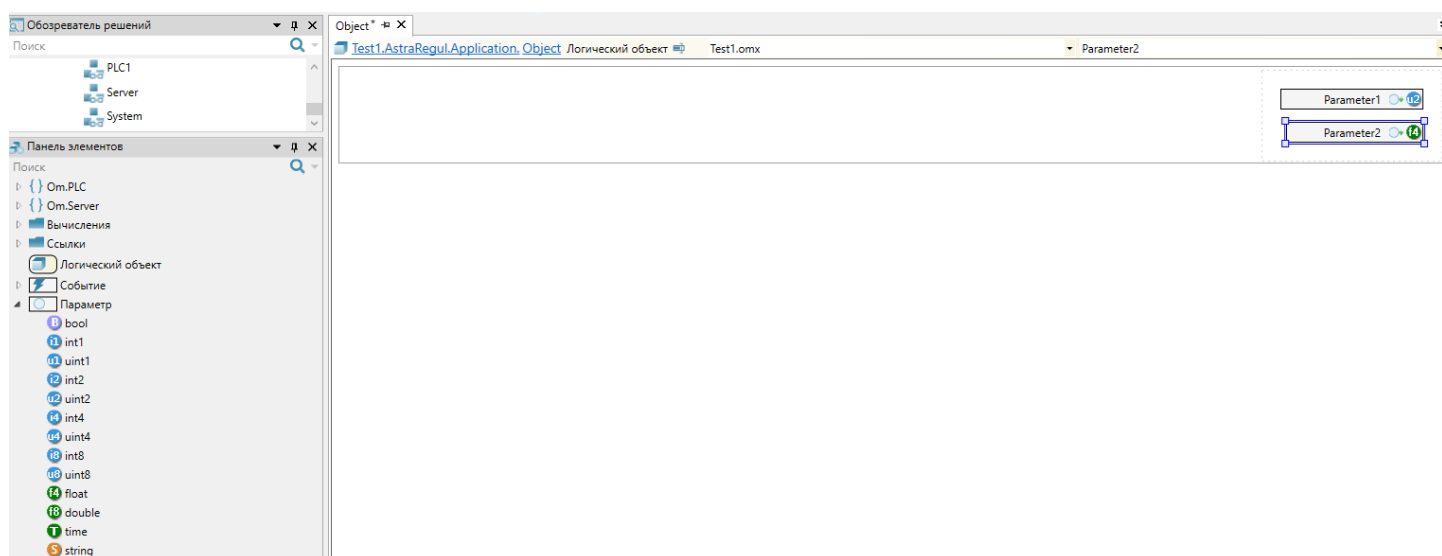
11. Перейдите в в узел AstraRegul и из панели элементов добавьте элемент Приложение.



12. Перейдите в Приложение Application и из панели элементов добавьте Логический объект для описания параметров, по которым будет воспроизводиться история.

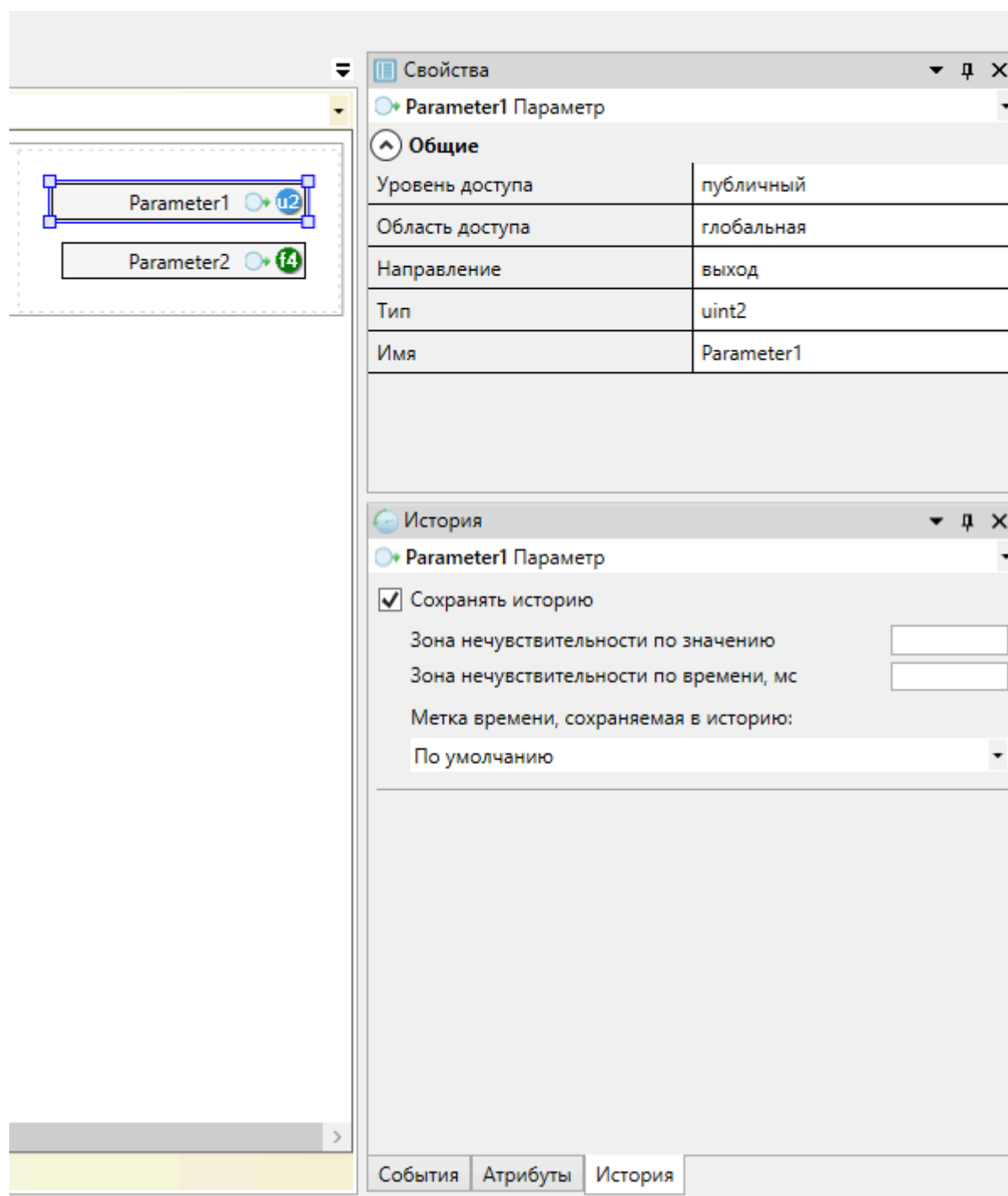


13. Перейдите в логический объект «Object» и добавьте параметры, по которым требуется сохранять историю в Astra.Historian, а затем воспроизводить с помощью Astra.Imitator.

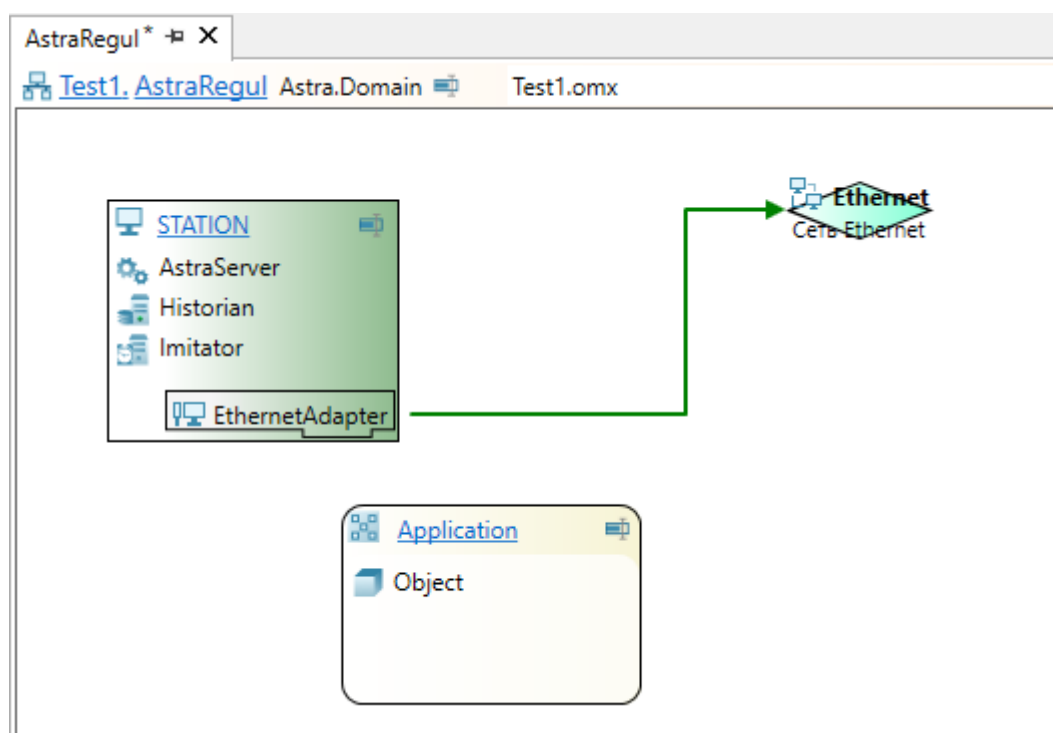




14. Для добавленных параметров необходимо настроить сохранение истории. Для этого выделите параметр левым кликом мыши, перейдите на вкладку "История" и установите флаг "Сохранять историю".

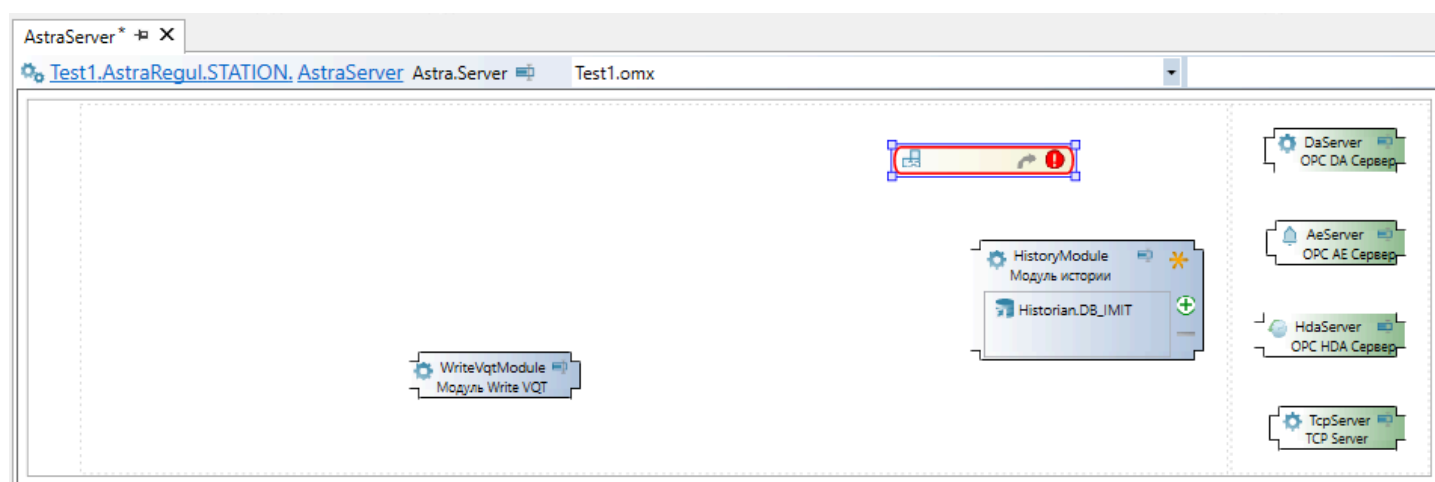


15. Так как наборы параметров конфигурации Astra.Server и Astra.Imitator должны совпадать, приложение, описывающее набор параметров размещено в узле AstraRegul.

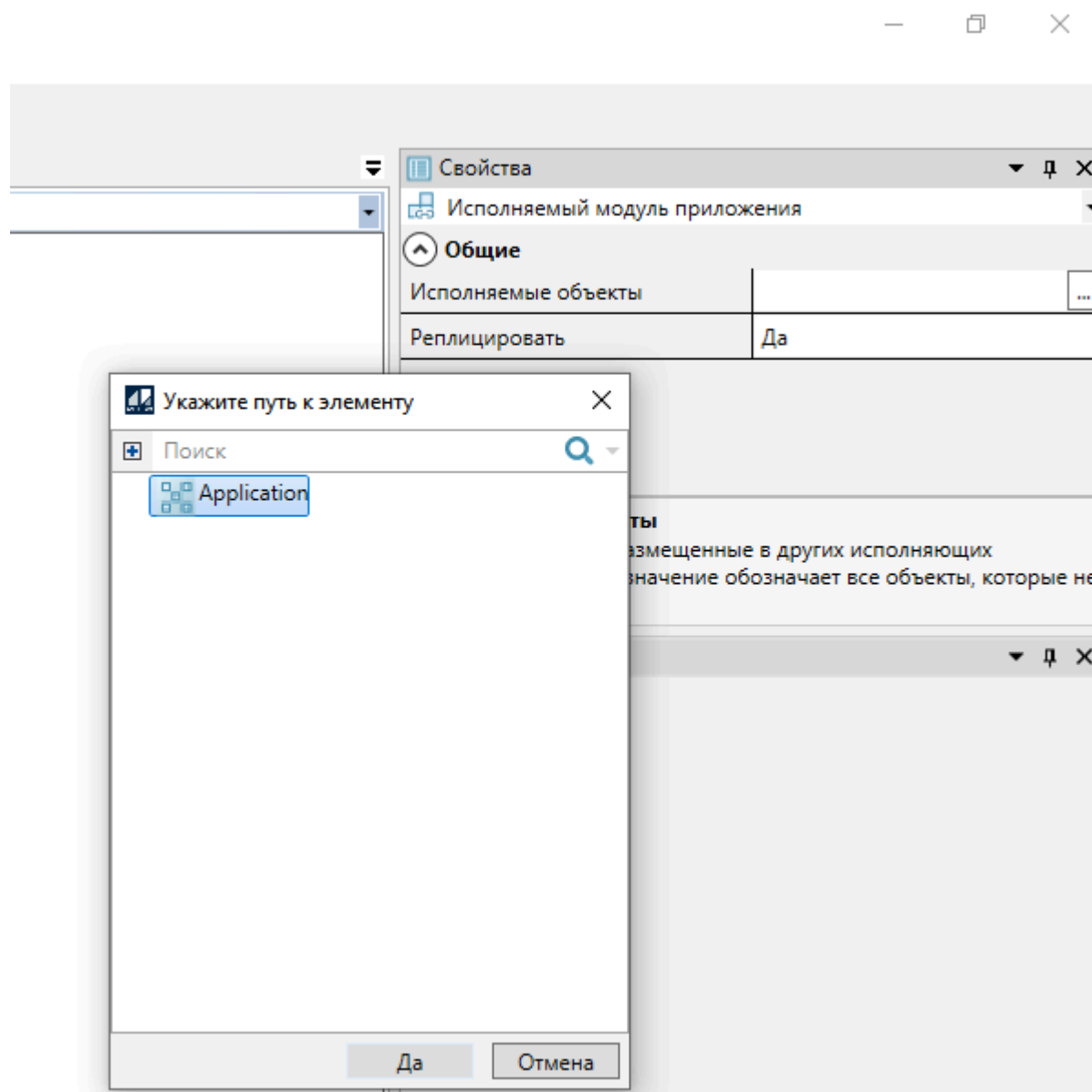


Теперь необходимо сослаться на Приложение Application в Astra.Server и Astra.Imitator.

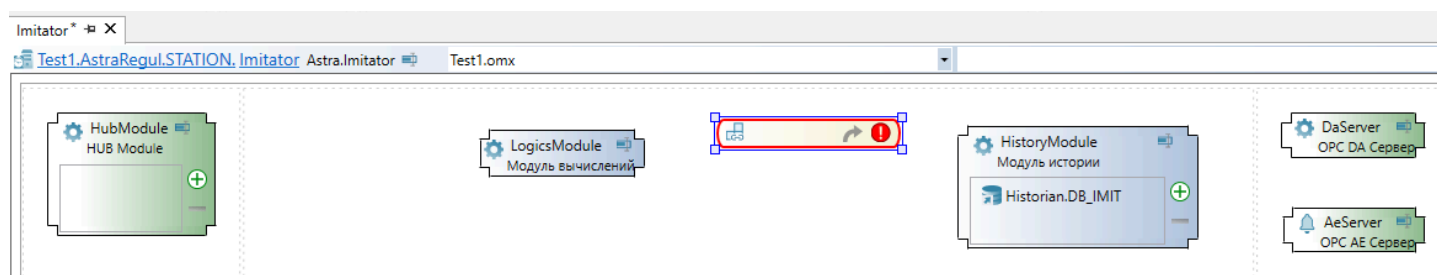
16. Перейдите в узел Инженерной станции, а затем в Astra.Server. Добавьте в Astra.Server из панели элементов Исполняемый модуль приложения.



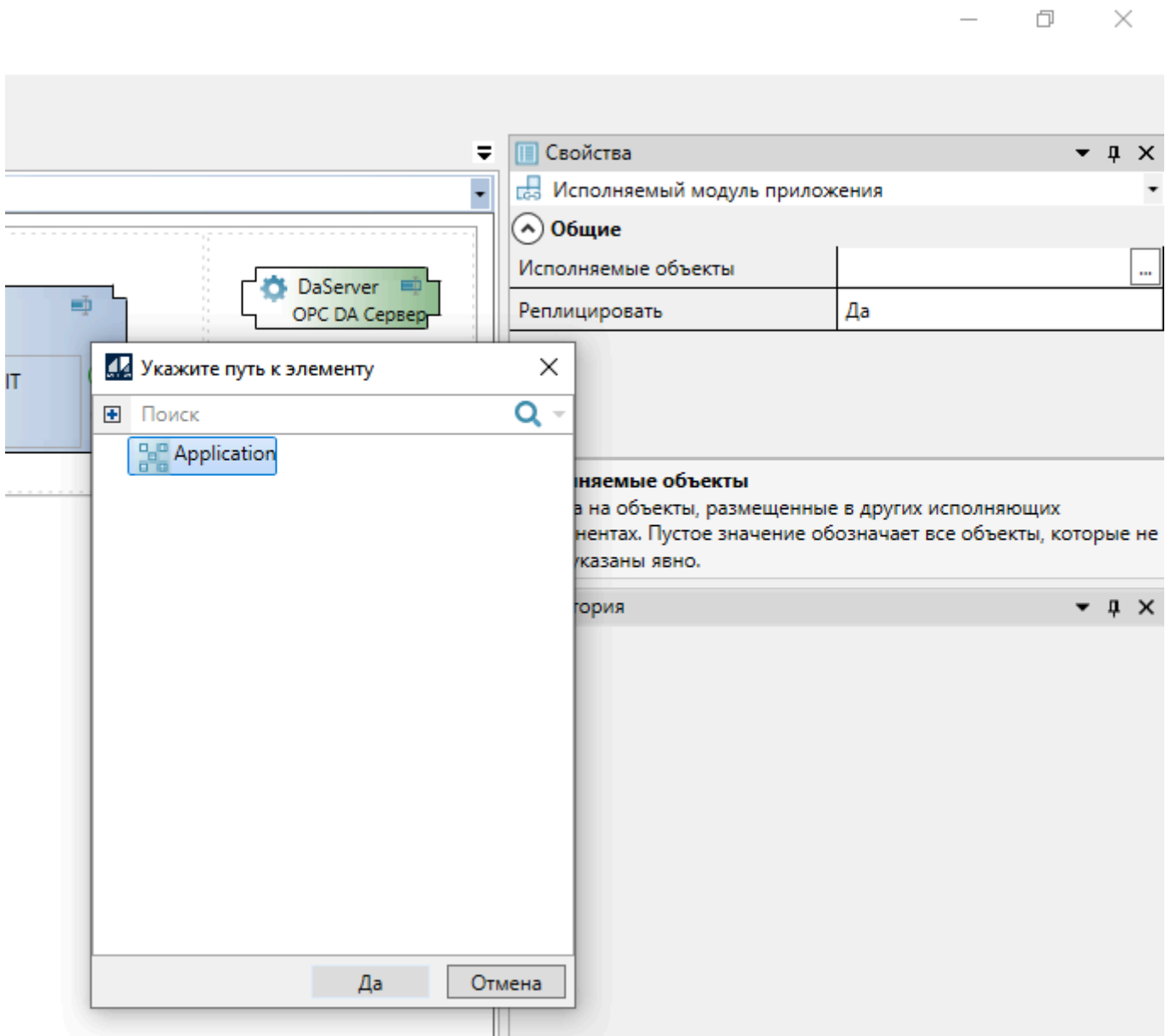
17. В свойстве Исполняемые объекты добавленного элемента укажите добавленное ранее Приложение и нажмите кнопку "Да".



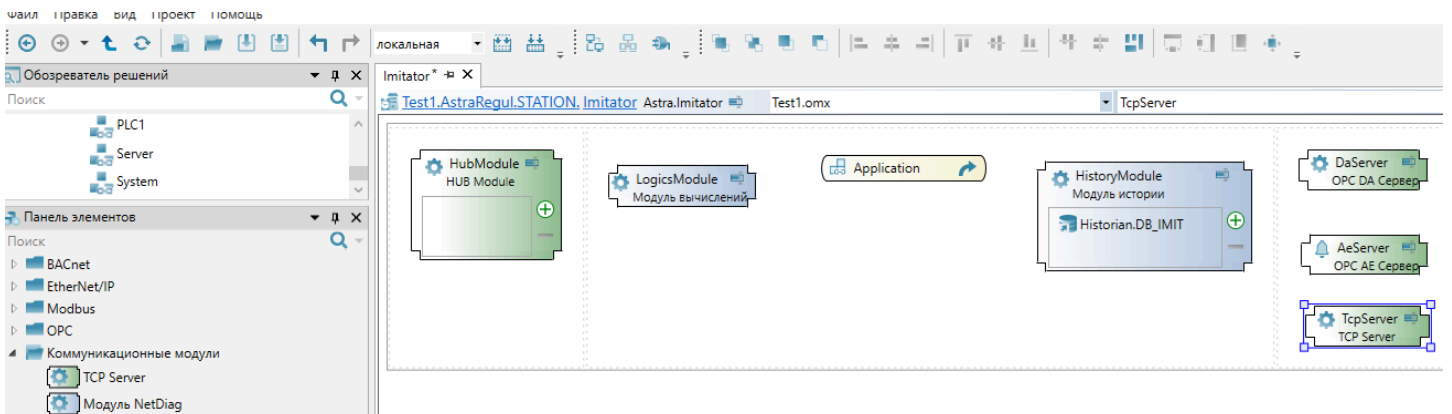
18. Перейдите в Astra.Imitator, добавьте из панели элементов Исполняемый модуль приложения.



19. В свойстве Исполняемые объекты добавленного элемента укажите описанное ранее Приложение и нажмите кнопку "Да".



20. Добавьте в Astra.Imitator из панели элементов модуль TCP Server.

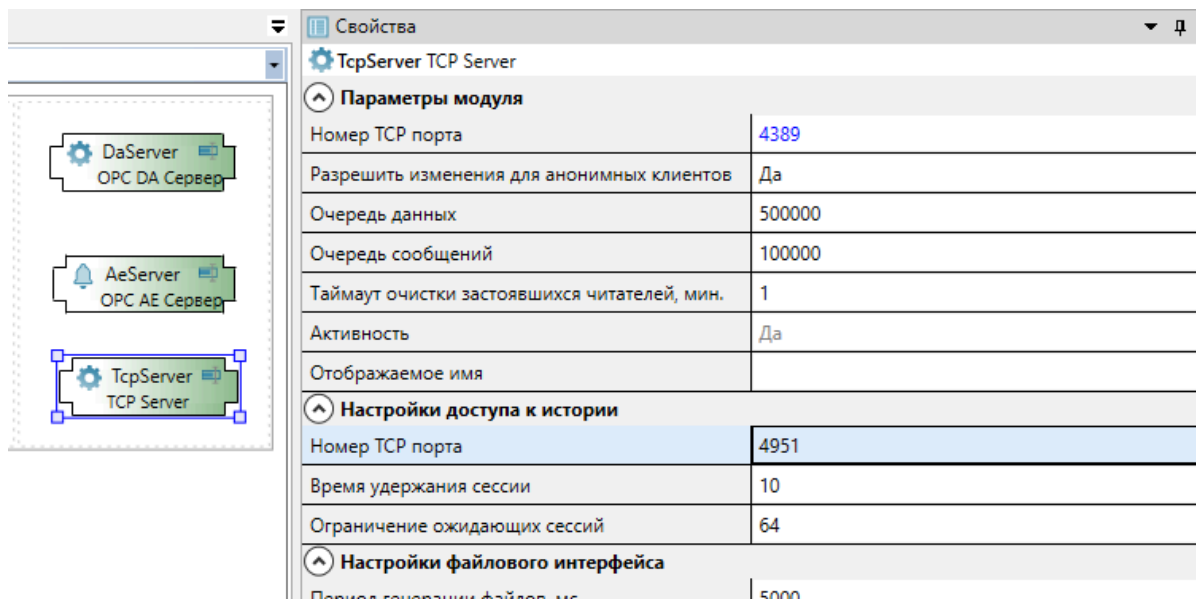




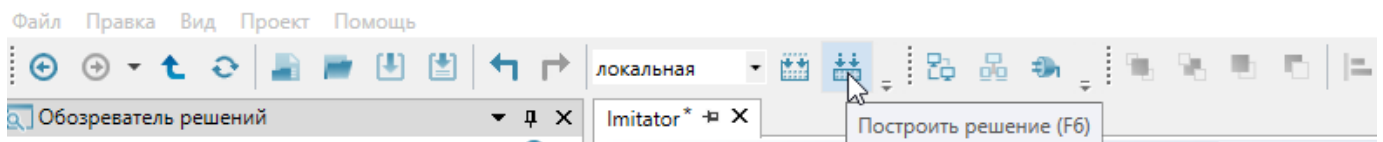
Коммуникационный модуль TCP Server необходимо добавить для того, чтобы передавать данные на HMI.

21. Укажите следующие свойства у добавленного модуля TCP Server:

- Номер TCP порта - должен быть отличным от других серверов, например 4389.
- Номер TCP порта для доступа к истории - должен быть отличным от других серверов, например 4951.



22. Постройте решение нажав кнопку "Построить решение" (F6) на панели инструментов.



23. Перейдите в Мастер развертывания и примените конфигурации к Astra.Server и Astra.Imitator.

Test1 [icon] X

Мастер развертывания

<input type="checkbox"/>	Исполняющий компонент	Построенная версия	Активная версия	Стабильная версия
<input checked="" type="checkbox"/>	AstraRegul			
<input checked="" type="checkbox"/>	STATION			
<input type="checkbox"/>	AstraServer	1.0.0.0.d4	1.0.0.0.d4 от 08.12.2023 12:10:21	
<input type="checkbox"/>	Imitator	1.0.0.0.d4	1.0.0.0.d4 от 08.12.2023 12:10:21	

## 1.3.3. Воспроизведение истории

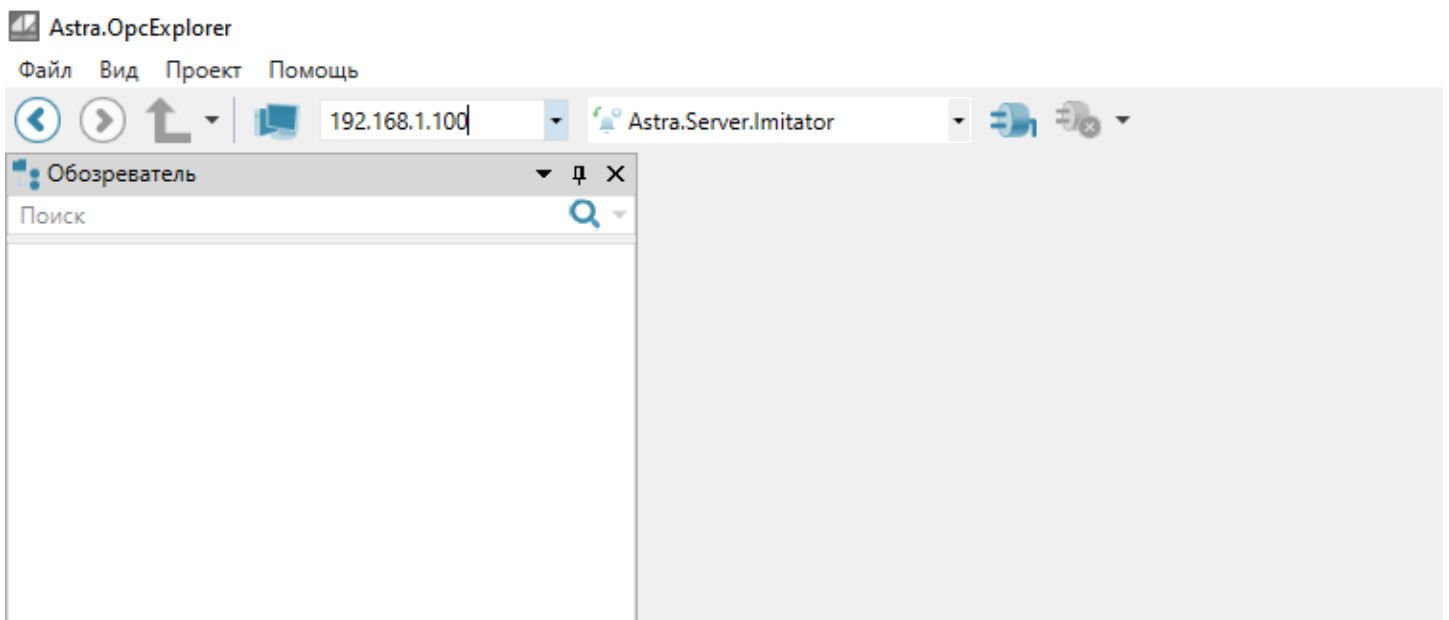
- › [Подключение к серверу](#)
- › [Сервисные сигналы](#)
- › [Просмотр истории](#)

## 1.3.3.1. Подключение к серверу

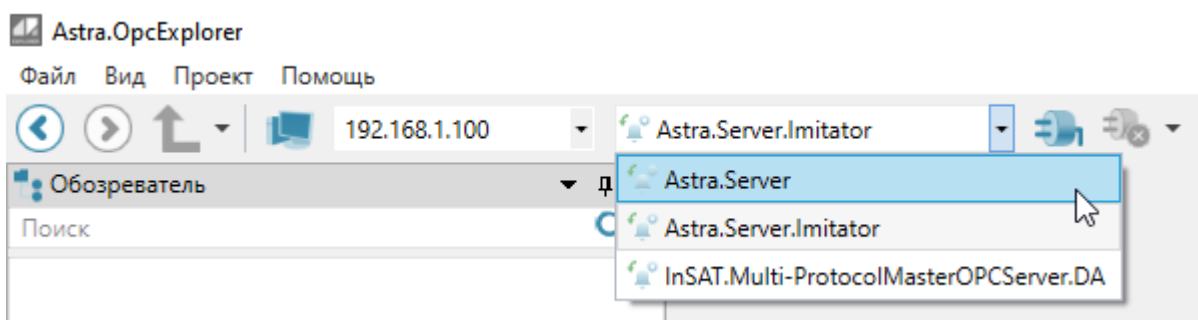
При старте Astra.Imitator создаёт сервисные сигналы для управления и контроля процесса проигрывания.

Чтобы изменять значения динамических сигналов, необходимо выполнить подключение к Astra.Imitator утилитой OpсExplorer выполнив следующие действия:

1. В открывшейся утилите OpсExplorer в поле источника введите IP адрес, который был указан в Astra.Astudio и нажмите клавишу Enter.

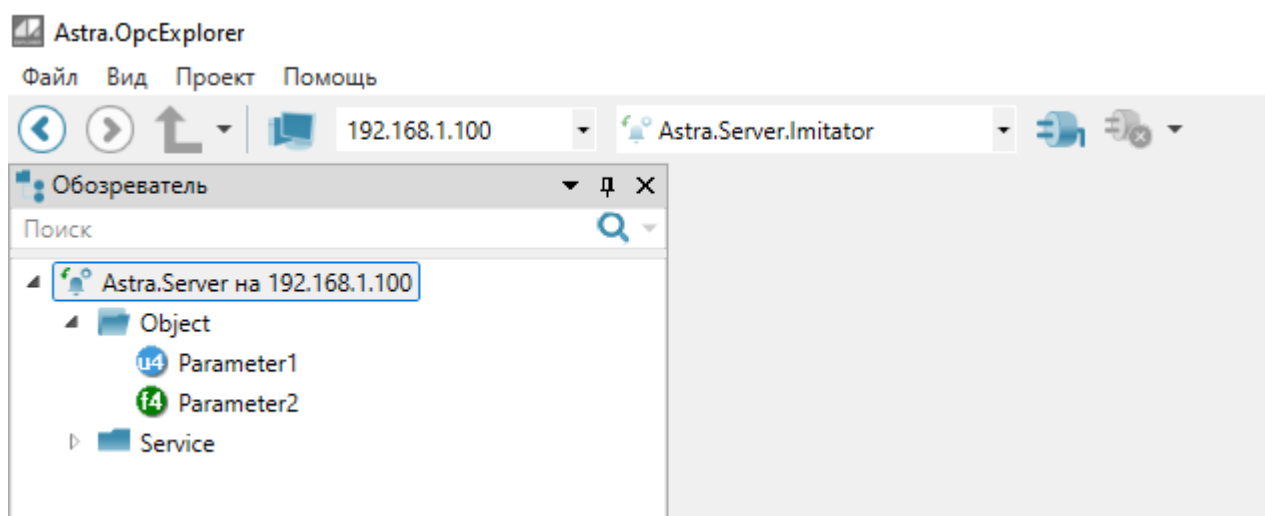


2. В поле источники появилось 2 сервера для подключения: Astra.Server и Astra.Server.Imitator.

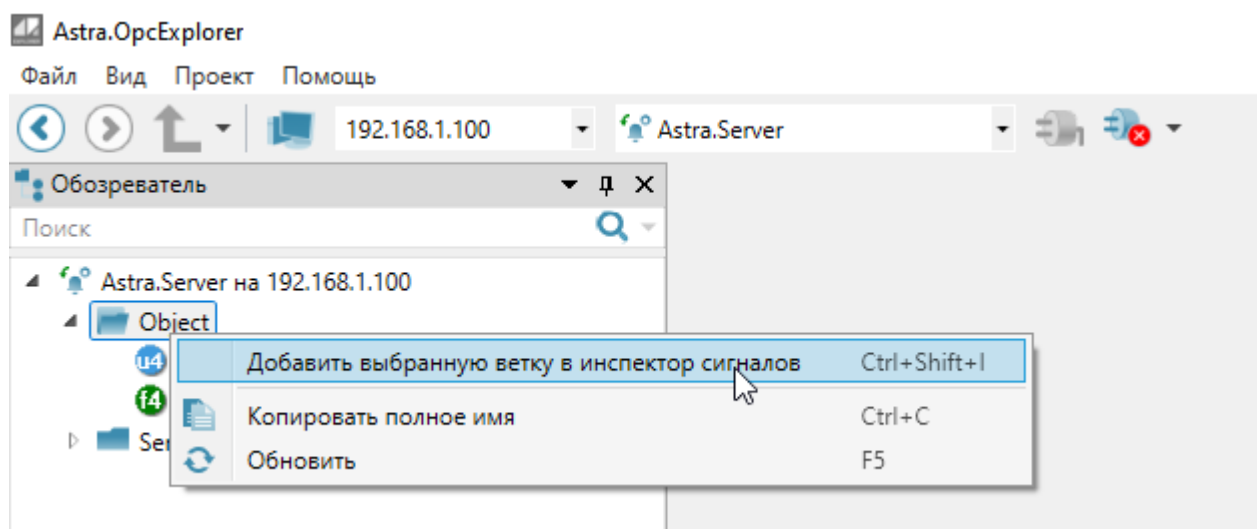




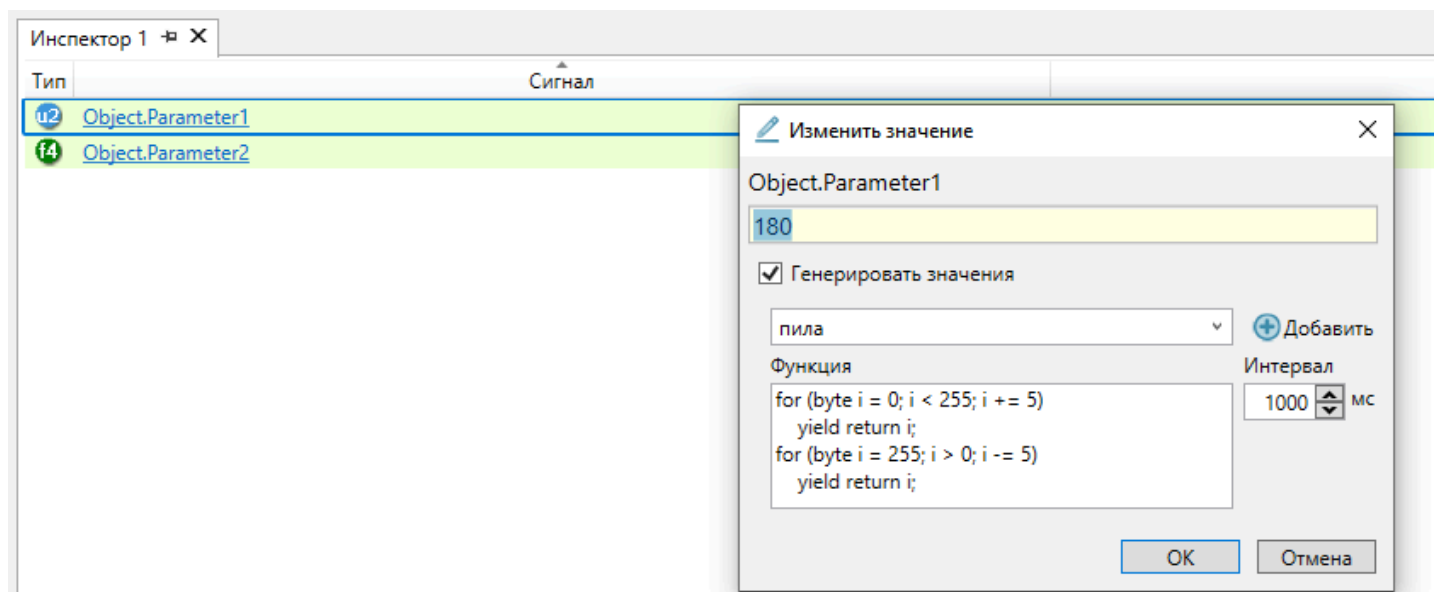
3. Выполните подключение к Astra.Server и в обозревателе раскройте папку "Object".



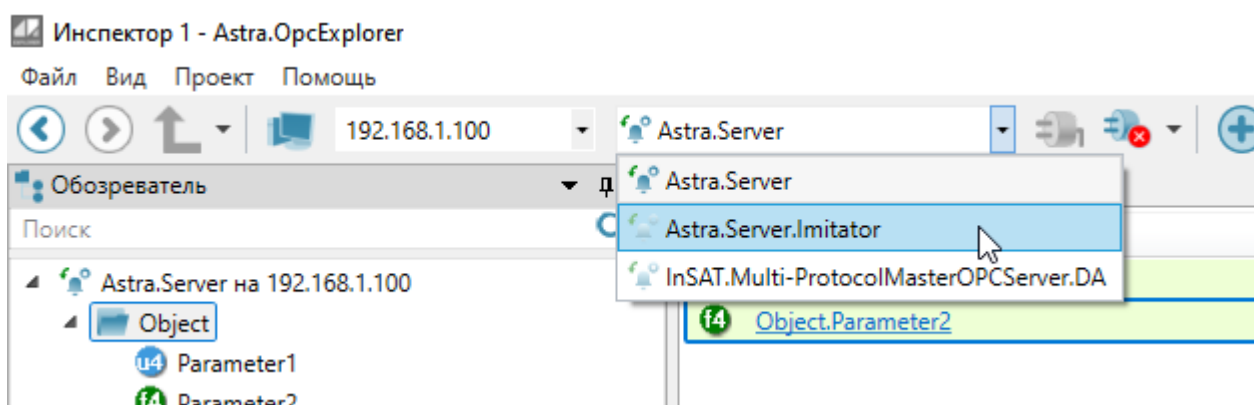
4. Кликните правой кнопкой мыши по папке Object и в контекстном меню выберите команду "Добавить выбранную ветку в инспектор сигналов".



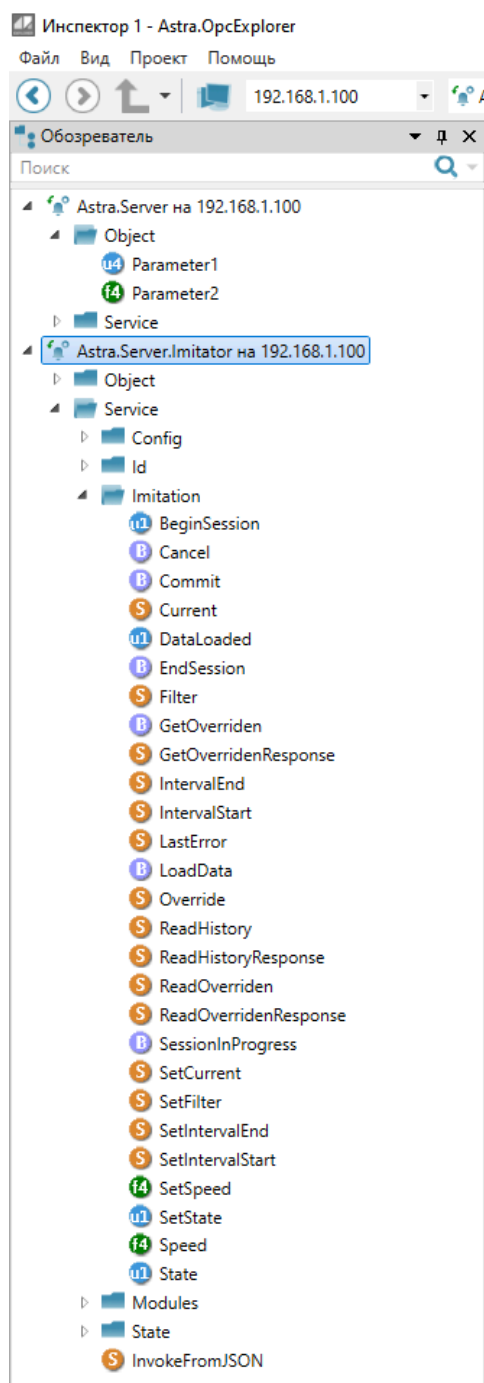
5. Добавьте произвольную генерацию значений для Parameter1 и Parameter2.



6. Из списка источников выберите Astra.Server.Imitator и выполните подключение.



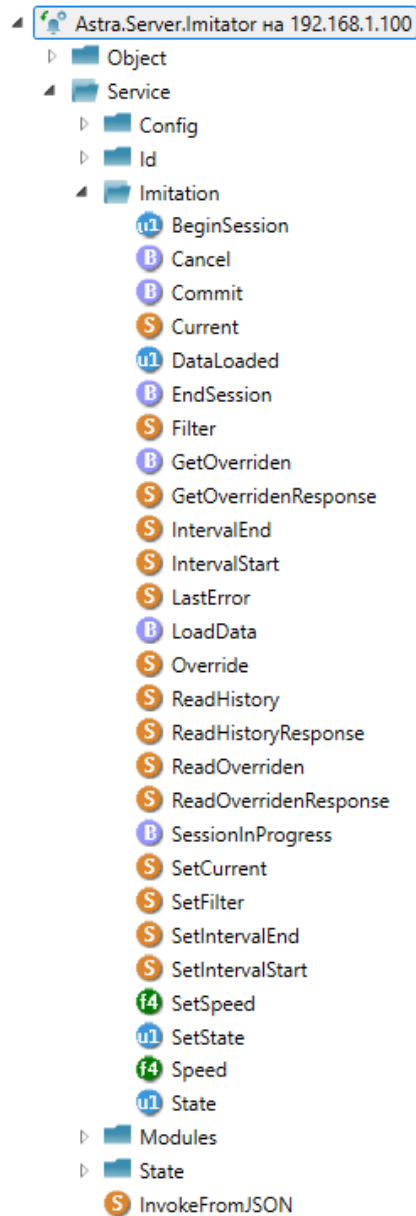
7. Astra.Server.Imitator будет добавлен в обозреватель, раскройте папку Service и далее перейдите в папку Imitation.



Управление воспроизведением истории Astra.Imitator осуществляется при помощи сервисных сигналов.

## 1.3.3.2. Сервисные сигналы

Сервисные сигналы сгруппированы в узле Service в папке Imitation.



### Управление сессией

Тег сигнала	Тип	Описание и использование
BeginSession	UInt1	Начать новую сессию воспроизведения истории: ‣ 0 – Режим плеера истории
EndSession	Bool	Завершить сессию воспроизведения истории: ‣ true – закрыть сессию



## Управление загрузкой и проигрыванием

Тег сигнала	Тип	Описание и использование
IntervalStart	String	Метка времени начала интервала проигрывания. Задается в формате <code>datetime_json</code> . Сбрасывается после установки сигналов <code>EndSession</code> и <code>BeginSession</code> . Значение по умолчанию: "" – левая граница отсутствует
SetIntervalStart	String	Задать метку времени начала интервала проигрывания. Задается в формате <code>datetime_json</code>
IntervalEnd	String	Метка времени конца интервала проигрывания. Задается в формате <code>datetime_json</code> . Сбрасывается после установки сигналов <code>EndSession</code> и <code>BeginSession</code> . Значение по умолчанию: "" – правая граница отсутствует
SetIntervalEnd	String	Задать метку времени конца интервала проигрывания. Задается в формате <code>datetime_json</code>
Filter	String	Фильтр по тегам. Сбрасывается после установки сигналов <code>EndSession</code> и <code>BeginSession</code> . Значение по умолчанию: "" – все теги
SetFilter	String	Установить фильтр по тегам
DataLoaded	Uint1	Процент готовности данных к проигрыванию
LastError	String	Последняя ошибка

## Параметры имитации

Тег сигнала	Тип	Описание и использование
LoadData	Bool	Загрузить данные для проигрывания: <ul style="list-style-type: none"> <li>› true – загрузить данные</li> </ul> Загрузка возможна только в рамках открытой сессии
SetSpeed	Float	Установить скорость проигрывания. Делитель времени ожидания между точками имитации. Сбрасывается после установки сигналов EndSession и BeginSession
SetCurrent	String	Установить текущее положение процесса проигрывания. Перемещает текущую метку проигрывания на заданную. Задаётся в формате datetime_json. Сбрасывается после установки сигналов EndSession и BeginSession
SetState	Uint1	Управление состоянием проигрывания: <ul style="list-style-type: none"> <li>› 0 – пауза;</li> <li>› 1 – проигрывание</li> </ul> Проигрывание возможно после полной загрузки данных
Cancel	Bool	Отменить результаты имитации: <ul style="list-style-type: none"> <li>› true – отменить.</li> </ul>
Commit	Bool	Применить результаты имитации: <ul style="list-style-type: none"> <li>› true – применить.</li> </ul>
GetOverriden	Bool	Запросить список переопределяемых тегов <ul style="list-style-type: none"> <li>› true – запросить.</li> </ul>
GetOverridenResponse	String	Список переопределяемых тегов
Override	String	Задать перезаписываемые сигналы
ReadHistory	String	Запросить список исторических значений
ReadHistoryResponse	String	Список исторических значений

ReadOverriden	String	Запросить список переопределяемых значений
ReadOverridenResponse	String	Список переопределяемых значений
SetFilter	String	Установить фильтр по тегам

## Контроль текущего состояния сессии проигрывания

Тег сигнала	Тип	Описание и использование
DataLoaded	Uint1	Процент загруженных данных от 0 до 100. Сбрасывается после установки сигналов EndSession и BeginSession. Значение по умолчанию: 0
Current	String	Текущее положение процесса проигрывания. Отображается в формате datetime_json. Сбрасывается после установки сигналов EndSession и BeginSession. Значение по умолчанию: ""
Speed	Float	Скорость проигрывания. Делитель времени ожидания между точками имитации. Сбрасывается после установки сигналов EndSession и BeginSession. Значение по умолчанию: 1
State	Uint1	Статус проигрывания: <ul style="list-style-type: none"> <li>&gt; 0 – проигрывание остановлено или завершено;</li> <li>&gt; 1 – проигрывание</li> </ul> Сбрасывается после установки сигналов EndSession и BeginSession. Значение по умолчанию: 0
SessionInProgress	Bool	Флаг наличия активной сессии. Устанавливается после установки сигнала BeginSession, сбрасывается после установки сигнала EndSession.



	Значение по умолчанию: false
--	------------------------------

## Формат datetime\_json

Метка времени задаётся и отображается по времени UTC в сервисных сигналах в формате datetime\_json и имеет вид:



```
{"y":<uint>,"mo":<uint>,"d":<uint>,"h":<uint>,"m":<uint>,"s":<uint>,"ms":<uint>
```

где:

- > y – год;
- > mo – месяц;
- > d – день;
- > h – часы;
- > m – минуты;
- > s – секунды;
- > ms – миллисекунды.



Заполнение всех полей необязательно, при этом для пропущенных полей метки времени применяется текущая дата и время 00:00:00.000.



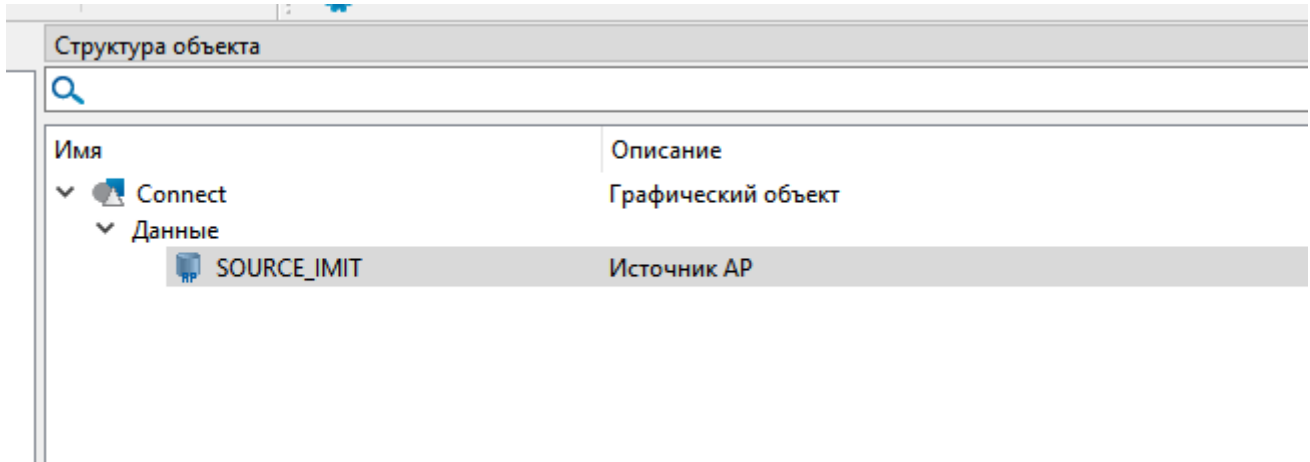
Метка времени 08.12.2023 10:12:15.2:

```
{"y":2023,"mo":12,"d":8,"h":5,"m":12,"s":15,"ms":2}
```

### 1.3.3.3. Просмотр истории

Для просмотра истории будет использоваться встраиваемый компонент Astra.HMI.Trends. Чтобы настроить просмотр истории необходимо выполнить следующие действия:

1. В проекте Astra.HMI добавьте новый источник AP и задайте ему произвольное название.



2. У добавленного источника заполните следующие свойства:

- › Порт - должен быть таким же как и в Astra.Astudio.
- › Порт истории - должен быть таким же как и в Astra.Astudio.
- › Хост - адрес источника данных.
- › Активность подключения - true.

Структура объекта

Имя | Описание

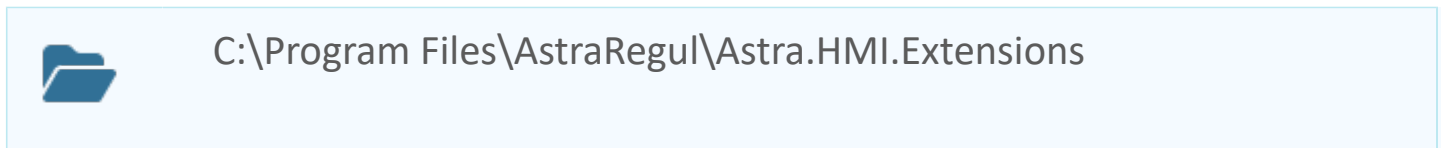
- Connect | Графический объект
  - Данные
    - SOURCE\_IMIT | Источник AP

Редактор свойств

Свойство | Значение

Отображаемое имя	SOURCE_IMIT
Кардинальное число	1
Родительский источник	<не определено>
Источник учетных данных	<не определено>
Таймаут операций с БД	<не определено>
Таймаут установления соединения с ...	<не определено>
Максимальный размер пула буферов	<не определено>
Общее соединение	<не определено>
Хост	127.0.0.1
Порт	4389
Порт истории	4951
Путь	
Активность	true
Период переподписки	1000
Имя клиента	HMI
Идентификатор клиента	HMI

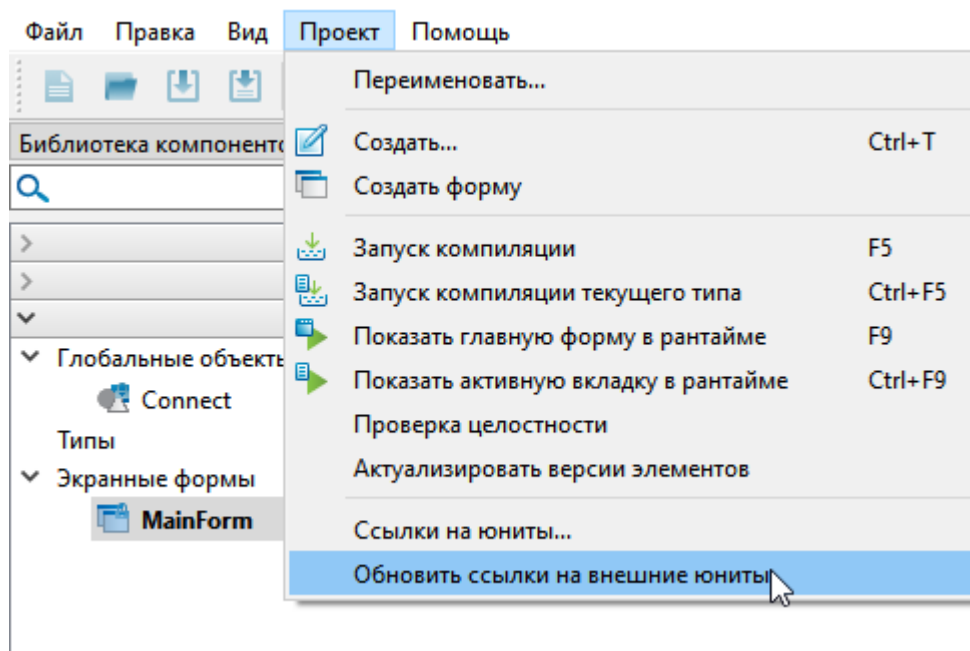
3. Добавьте внешний модуль Astra.HMI.Trends, переместив его из следующей папки:



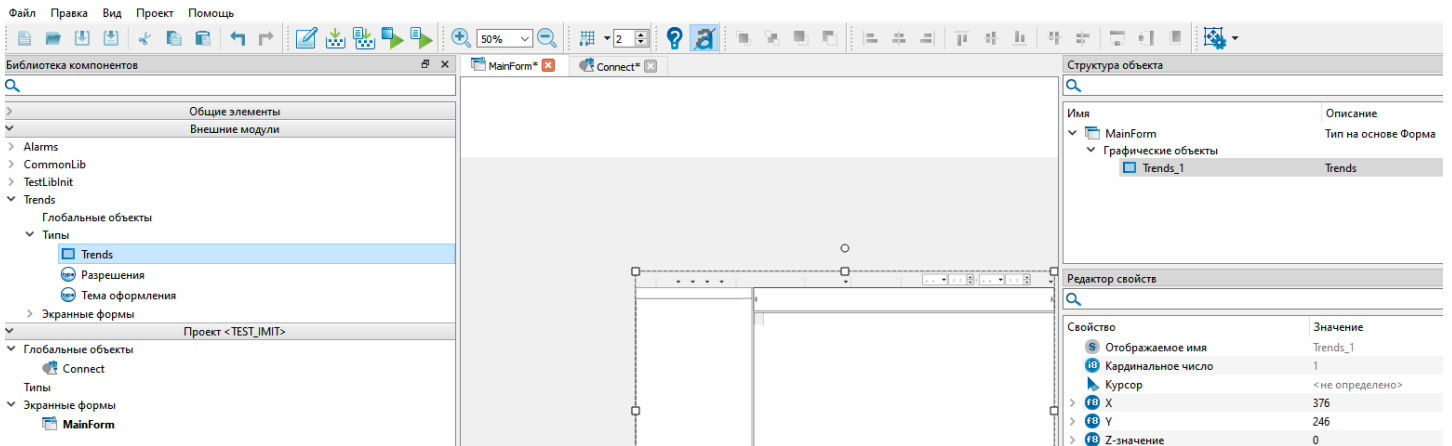
В папку externals проекта.

Имя	Дата изменения	Тип	Размер
externals	08.12.2023 14:13	Папка с файлами	
objects	08.12.2023 14:13	Папка с файлами	
output	08.12.2023 11:10	Папка с файлами	
resources	16.11.2023 9:55	Папка с файлами	
TEST_IMIT.hmi	07.12.2023 11:46	Astra.HMI project	1 КБ

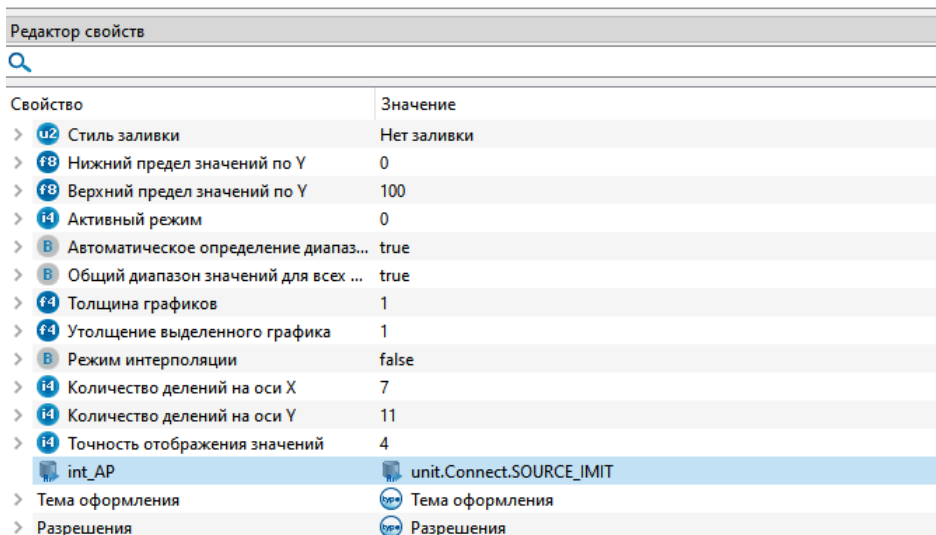
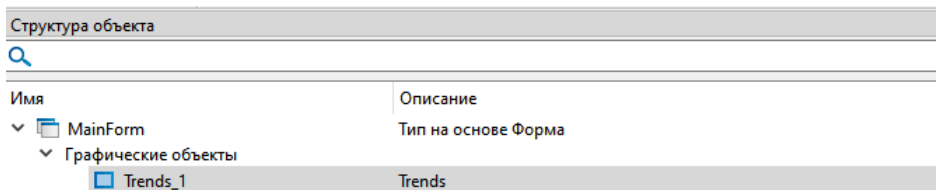
4. Чтобы добавленный модуль отобразился в проекте необходимо воспользоваться командой "Обновить ссылки на внешние юниты" из вкладки Проект.



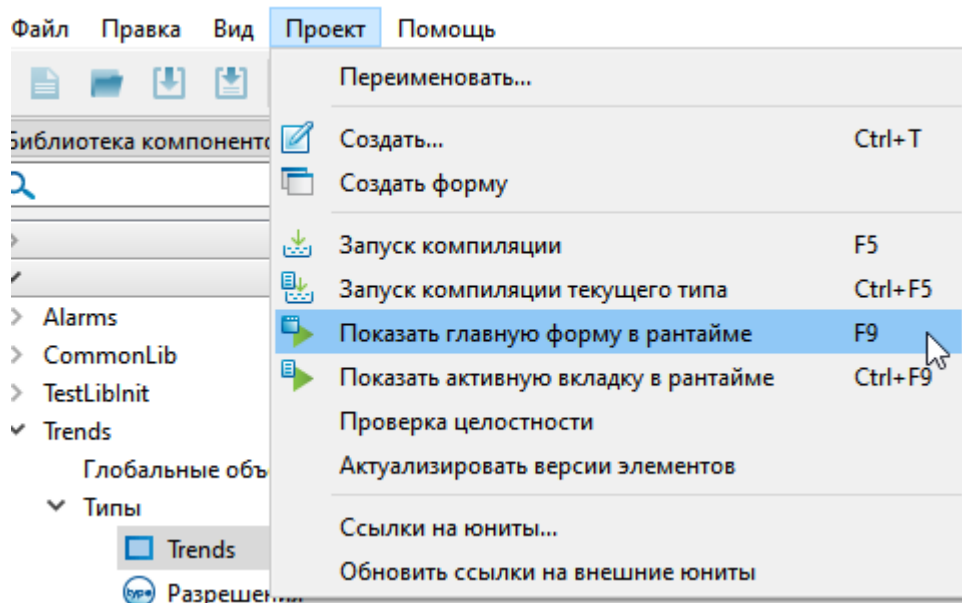
5. Добавьте из вкладки "Внешние модули" экземпляр типа Trends на мнемосхему.



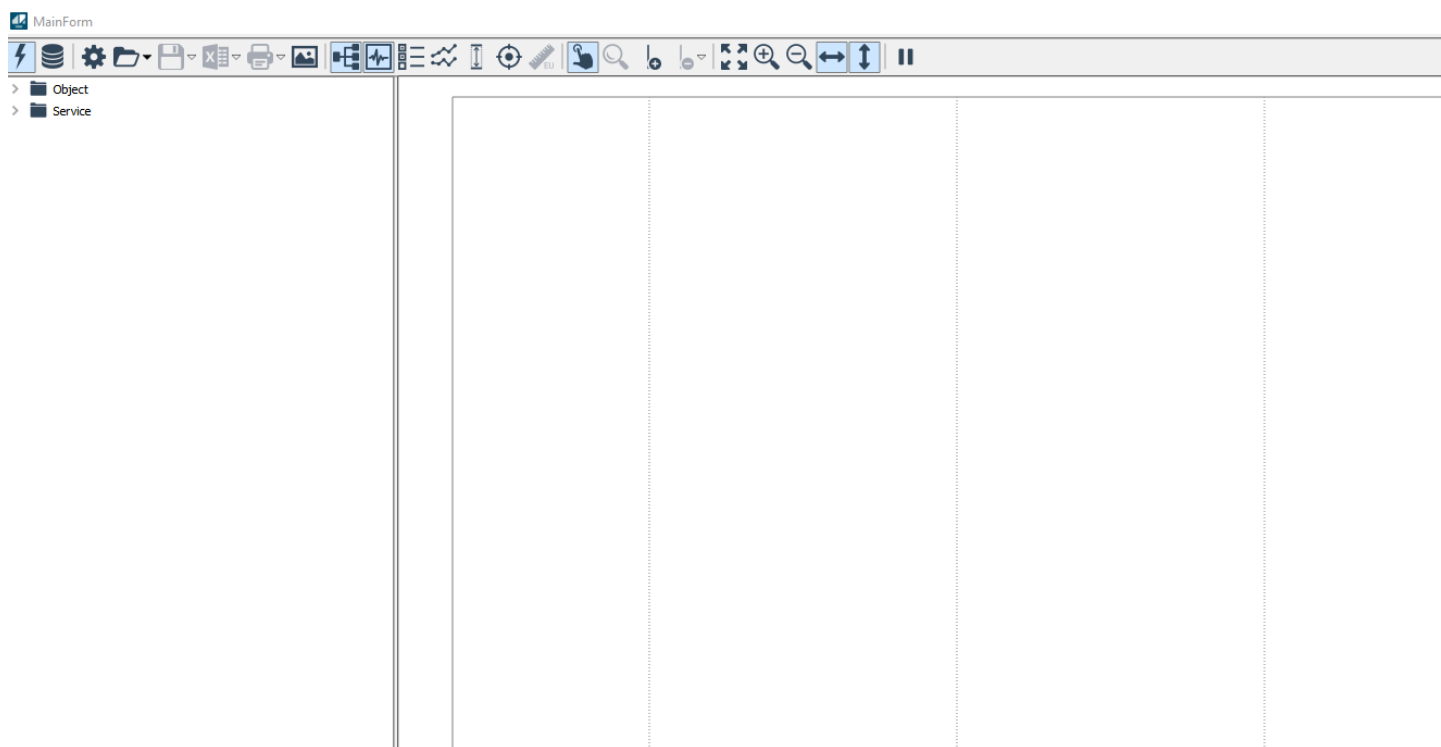
6. У экземпляра типа Trends в редакторе свойств в свойстве int\_AP укажите ссылку на добавленный источник в глобальном объекте.



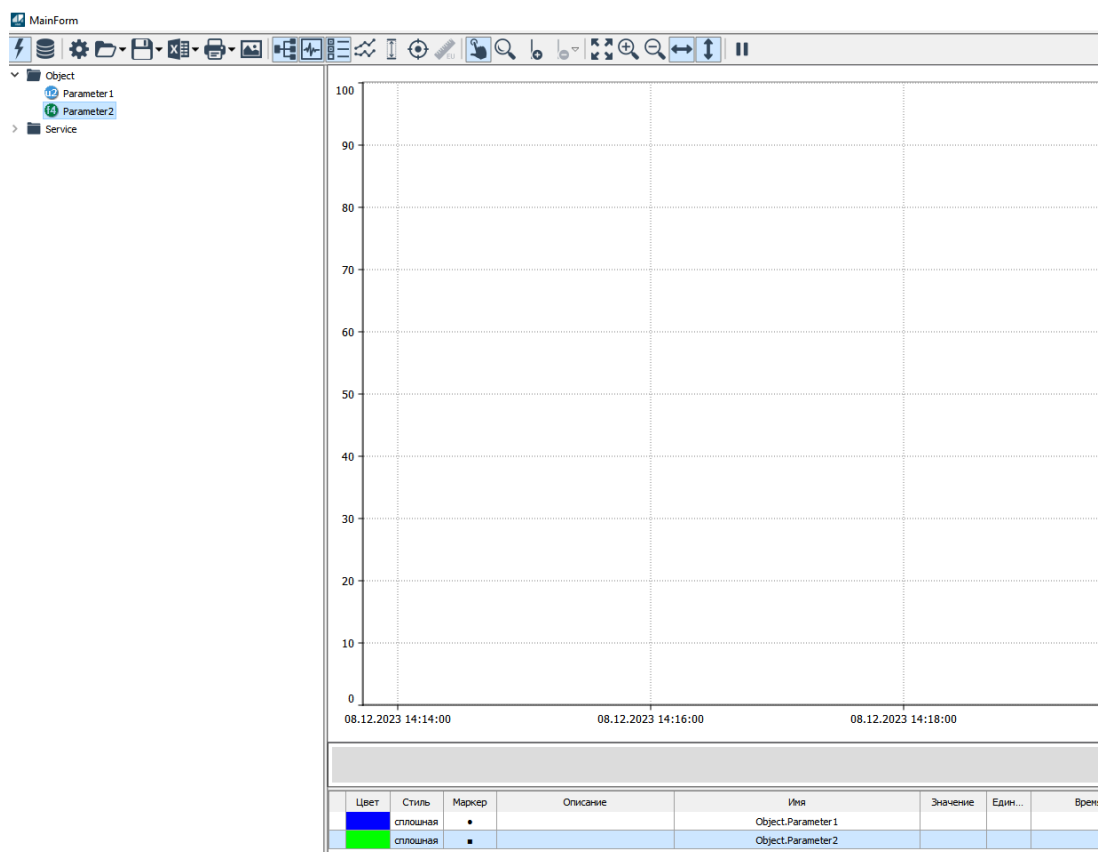
7. Запустите проект в режим исполнения клавишей F9 или воспользуйтесь командами из вкладки "Проект".



8. После запуска проекта в режиме исполнения откроется форма с экземпляром типа Trends.

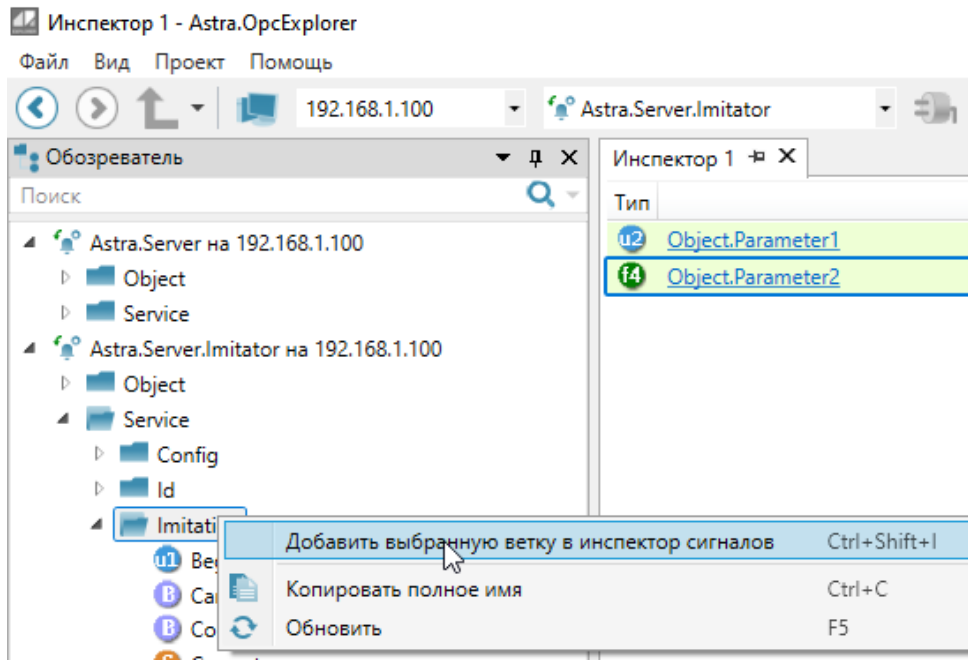


9. В дереве сигналов раскройте папку "Object" и добавьте на поле графиков 2 сигнала Parametr1 и Parametr2 путем перетаскивания сигналов на трендовое поле.

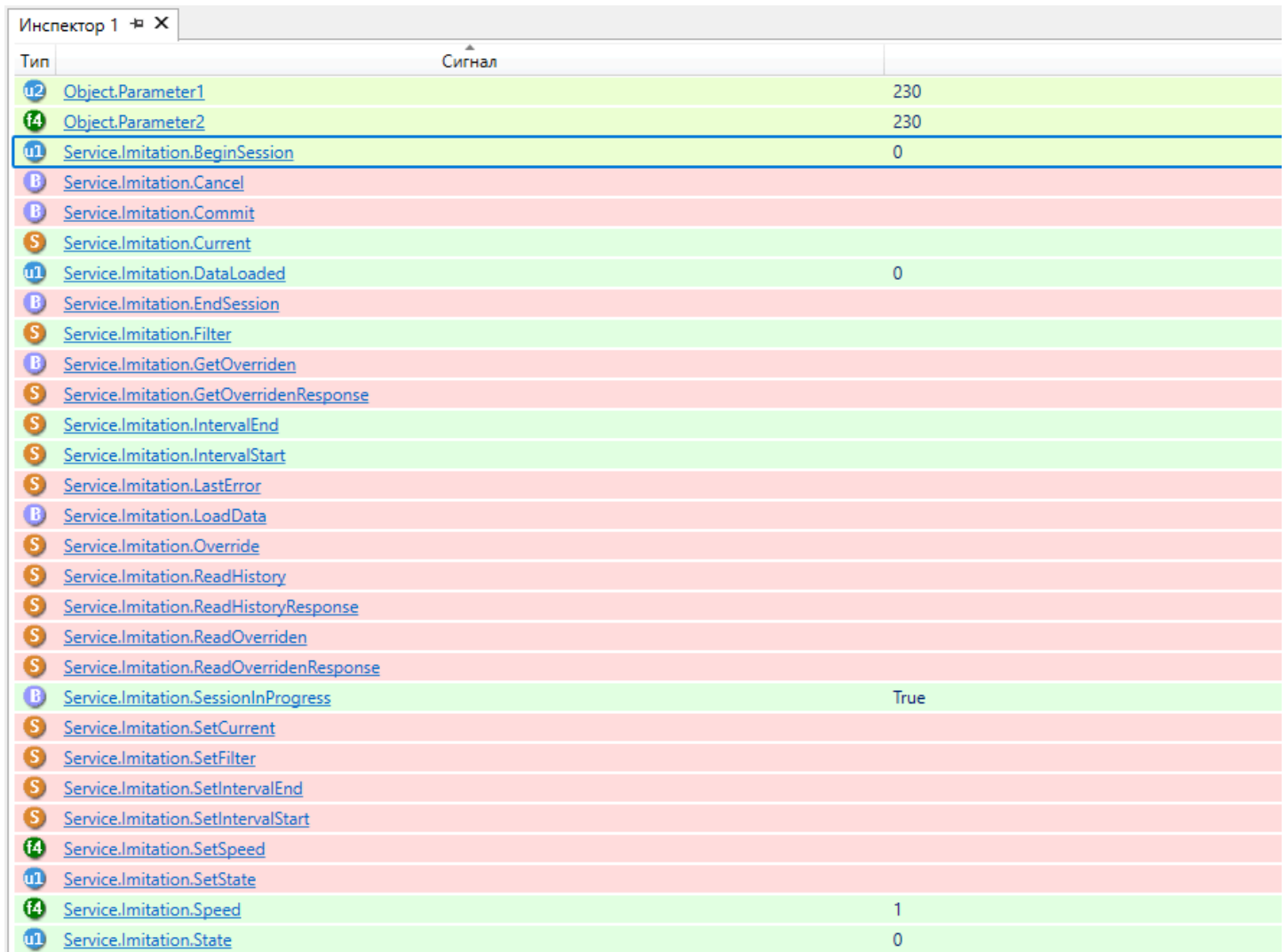


10. Откройте приложение OrcExplorer и добавьте сигналы Astra.Imitator из папки "Imitation" в инспектор сигналов кликнув правой кнопкой мыши по папке.





11. Откройте новую сессию воспроизведения истории, установив значение сигнала «BeginSession» = «0».



Тип	Сигнал	Значение
u2	Object.Parameter1	230
f4	Object.Parameter2	230
u1	Service.Imitation.BeginSession	0
B	Service.Imitation.Cancel	
B	Service.Imitation.Commit	
S	Service.Imitation.Current	
u1	Service.Imitation.DataLoaded	0
B	Service.Imitation.EndSession	
S	Service.Imitation.Filter	
B	Service.Imitation.GetOverriden	
S	Service.Imitation.GetOverridenResponse	
S	Service.Imitation.IntervalEnd	
S	Service.Imitation.IntervalStart	
S	Service.Imitation.LastError	
B	Service.Imitation.LoadData	
S	Service.Imitation.Override	
S	Service.Imitation.ReadHistory	
S	Service.Imitation.ReadHistoryResponse	
S	Service.Imitation.ReadOverriden	
S	Service.Imitation.ReadOverridenResponse	
B	Service.Imitation.SessionInProgress	True
S	Service.Imitation.SetCurrent	
S	Service.Imitation.SetFilter	
S	Service.Imitation.SetIntervalEnd	
S	Service.Imitation.SetIntervalStart	
f4	Service.Imitation.SetSpeed	
u1	Service.Imitation.SetState	
f4	Service.Imitation.Speed	1
u1	Service.Imitation.State	0

Новая сессия открывается только если нет активной сессии – флаг наличия активной сессии «SessionInProgress» = «false». Если «SessionInProgress» = «true», то необходимо завершить текущую сессию воспроизведения истории, установив значение сигнала «EndSession» = «true».

12. Задайте временной интервал в формате `datetime_json`, за который требуется воспроизвести историю, а также скорость проигрывания, установив значения сигналов:

- › «SetIntervalStart» = {"y":2023,"mo":12,"d":8,"h":6,"m":12,"s":15,"ms":2};
- › «SetIntervalEnd» = {"y":2023,"mo":12,"d":8,"h":7,"m":12,"s":15,"ms":2};
- › «SetSpeed» = 1.

Данные сигналы устанавливают значения сигналов «IntervalStart», «IntervalEnd», «Speed» соответственно.

Инспектор 1 ▾ X		
Тип	Сигнал	
u2	Object.Parameter1	115
f4	Object.Parameter2	115
u1	Service.Imitation.BeginSession	0
B	Service.Imitation.Cancel	
B	Service.Imitation.Commit	
S	Service.Imitation.Current	
u1	Service.Imitation.DataLoaded	0
B	Service.Imitation.EndSession	
S	Service.Imitation.Filter	
B	Service.Imitation.GetOverriden	
S	Service.Imitation.GetOverridenResponse	
S	Service.Imitation.IntervalEnd	{ "y": 2023, "mo": 12, "d": 8, "h": 7, "m": 12, "s": 15, "ms": 2 }
S	Service.Imitation.IntervalStart	{ "y": 2023, "mo": 12, "d": 8, "h": 6, "m": 12, "s": 15, "ms": 2 }
S	Service.Imitation.LastError	
B	Service.Imitation.LoadData	
S	Service.Imitation.Override	
S	Service.Imitation.ReadHistory	
S	Service.Imitation.ReadHistoryResponse	
S	Service.Imitation.ReadOverriden	
S	Service.Imitation.ReadOverridenResponse	
B	Service.Imitation.SessionInProgress	True
S	Service.Imitation.SetCurrent	
S	Service.Imitation.SetFilter	
S	Service.Imitation.SetIntervalEnd	{ "y": 2023, "mo": 12, "d": 8, "h": 7, "m": 12, "s": 15, "ms": 2 }
S	Service.Imitation.SetIntervalStart	{ "y": 2023, "mo": 12, "d": 8, "h": 6, "m": 12, "s": 15, "ms": 2 }
f4	Service.Imitation.SetSpeed	1
u1	Service.Imitation.SetState	
f4	Service.Imitation.Speed	1
u1	Service.Imitation.State	0

13. Загрузите данные для воспроизведения истории, установив значение сигнала «LoadData» = «true».

Процент загруженных данных отображается в значении сигнала «DataLoaded».

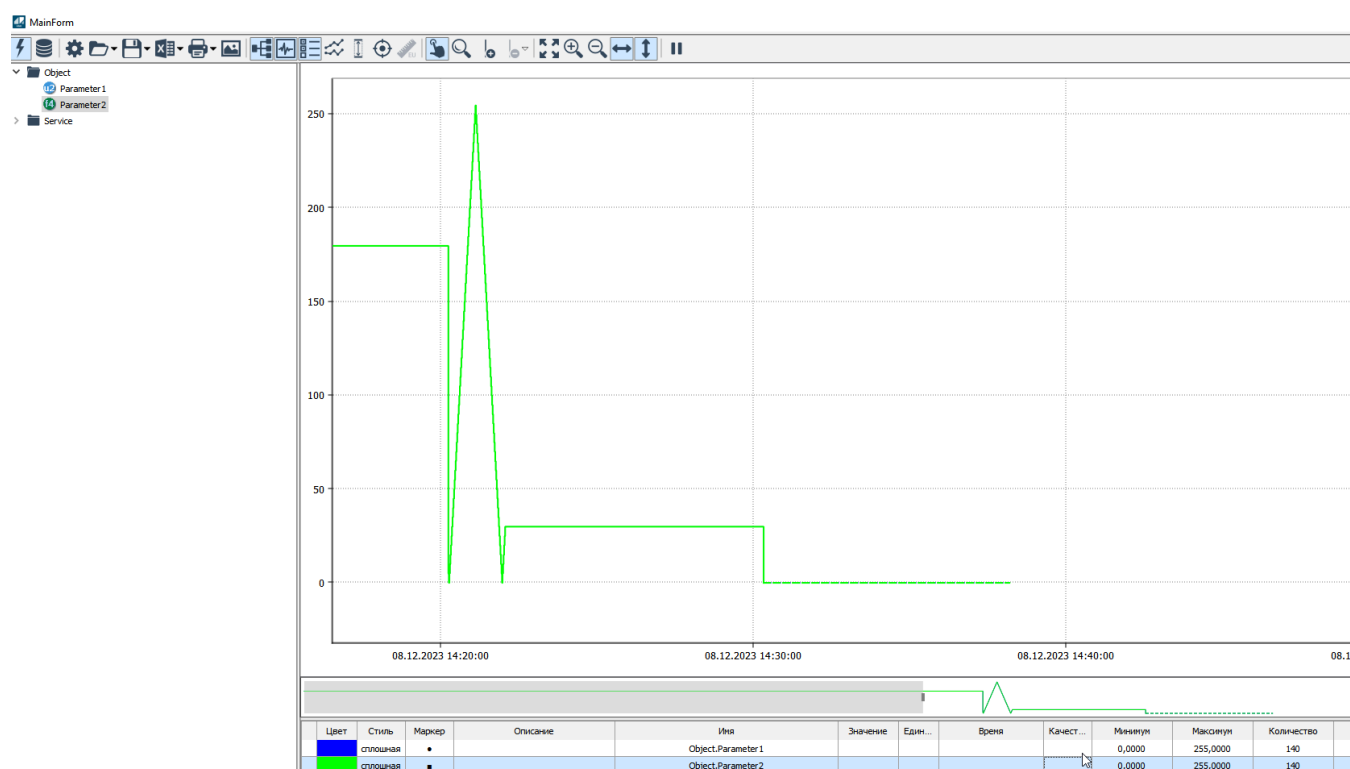
Инспектор 1 ✕

Тип	Сигнал	Значение
u2	Object.Parameter1	215
f4	Object.Parameter2	215
u1	Service.Imitation.BeginSession	0
B	Service.Imitation.Cancel	
B	Service.Imitation.Commit	
S	Service.Imitation.Current	
u1	Service.Imitation.DataLoaded	100
B	Service.Imitation.EndSession	
S	Service.Imitation.Filter	
B	Service.Imitation.GetOverriden	
S	Service.Imitation.GetOverridenResponse	
S	Service.Imitation.IntervalEnd	{ "y": 2023, "mo": 12, "d": 8, "h": 7, "m": 12, "s": 15, "ms": 2 }
S	Service.Imitation.IntervalStart	{ "y": 2023, "mo": 12, "d": 8, "h": 6, "m": 12, "s": 15, "ms": 2 }
S	Service.Imitation.LastError	
B	Service.Imitation.LoadData	True
S	Service.Imitation.Override	
S	Service.Imitation.ReadHistory	
S	Service.Imitation.ReadHistoryResponse	
S	Service.Imitation.ReadOverriden	
S	Service.Imitation.ReadOverridenResponse	
B	Service.Imitation.SessionInProgress	True
S	Service.Imitation.SetCurrent	
S	Service.Imitation.SetFilter	
S	Service.Imitation.SetIntervalEnd	{ "y": 2023, "mo": 12, "d": 8, "h": 7, "m": 12, "s": 15, "ms": 2 }
S	Service.Imitation.SetIntervalStart	{ "y": 2023, "mo": 12, "d": 8, "h": 6, "m": 12, "s": 15, "ms": 2 }
f4	Service.Imitation.SetSpeed	1
u1	Service.Imitation.SetState	
f4	Service.Imitation.Speed	1
u1	Service.Imitation.State	0

14. После полной загрузки данных («DataLoaded» == «100») запустите воспроизведение истории, установив значение сигнала «SetState» = «1».

Инспектор 1		
Тип	Сигнал	
u2	Object.Parameter1	40
f4	Object.Parameter2	40
u1	Service.Imitation.BeginSession	0
B	Service.Imitation.Cancel	
B	Service.Imitation.Commit	
S	Service.Imitation.Current	{ "y": 2023, "mo": 12, "d": 8, "h": 7, "m": 10, "s": 32, "ms": 80 }
u1	Service.Imitation.DataLoaded	100
B	Service.Imitation.EndSession	
S	Service.Imitation.Filter	
B	Service.Imitation.GetOverriden	
S	Service.Imitation.GetOverridenResponse	
S	Service.Imitation.IntervalEnd	{ "y": 2023, "mo": 12, "d": 8, "h": 7, "m": 12, "s": 15, "ms": 2 }
S	Service.Imitation.IntervalStart	{ "y": 2023, "mo": 12, "d": 8, "h": 6, "m": 12, "s": 15, "ms": 2 }
S	Service.Imitation.LastError	
B	Service.Imitation.LoadData	True
S	Service.Imitation.Override	
S	Service.Imitation.ReadHistory	
S	Service.Imitation.ReadHistoryResponse	
S	Service.Imitation.ReadOverriden	
S	Service.Imitation.ReadOverridenResponse	
B	Service.Imitation.SessionInProgress	True
S	Service.Imitation.SetCurrent	
S	Service.Imitation.SetFilter	
S	Service.Imitation.SetIntervalEnd	{ "y": 2023, "mo": 12, "d": 8, "h": 7, "m": 12, "s": 15, "ms": 2 }
S	Service.Imitation.SetIntervalStart	{ "y": 2023, "mo": 12, "d": 8, "h": 6, "m": 12, "s": 15, "ms": 2 }
f4	Service.Imitation.SetSpeed	1
u1	Service.Imitation.SetState	1
f4	Service.Imitation.Speed	1
u1	Service.Imitation.State	1

15. Перейдите на открытую форму в режиме исполнения с запущенным экземпляром типа Trends. Astra.Imitator начнет воспроизведение истории сигналов за выбранный интервал в сигнала SetIntrevalEnd и SetIntervalStart.



В процессе воспроизведения истории («State» == «1») возможно:

- Изменение скорости воспроизведения в сигнале «SetSpeed».
- Изменение текущего положения воспроизведения в сигнале «SetCurrent».
- Приостановка воспроизведения (пауза) - «SetState» = «0».
- Завершение сессии воспроизведения - «EndSession» = «true».

При достижении конца воспроизводимого интервала или приостановке воспроизведения («State» == «0») возможно:

- Изменение скорости воспроизведения истории в сигнале «SetSpeed».
- Изменение текущего положения воспроизведения в сигнале «SetCurrent».
- Возобновление воспроизведения - «SetState» = «1».
- Завершение сессии воспроизведения - «EndSession» = «true».

Чтобы завершить сессию воспроизведения истории, установите значение сигнала «EndSession» = «true».

При этом сбрасываются значения сигналов «IntervalEnd», «IntervalStart», «Current», а также устанавливаются следующие значения сигналов:

- > «State» = «0»;
- > «Speed» = «1»;
- > «SessionInProgress» = «false».

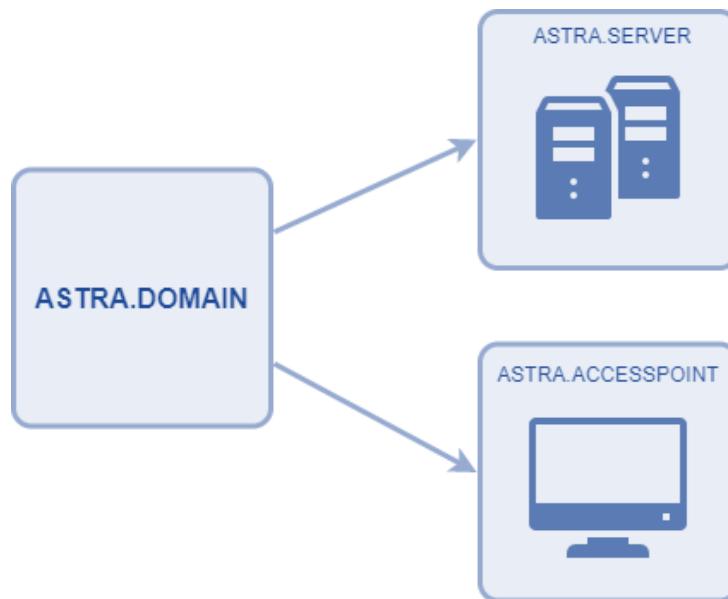
## 1.4. Astra.Domain

**Astra.Domain** - программный компонент, объединяющий отдельные исполняющие компоненты в домен ПТК AstraRegul. Служит для развертывания конфигурации на все компьютеры системы (серверы, АРМ) из Astra.AStudio.

### Функции

- › передача конфигураций по сети Astra.Net (служба Astra.Net Agent).
- › применение конфигурации на экземпляры Astra.Server и Astra.AccessPoint (служба Astra.Domain.Agent).

Исполняющие компоненты объединяются в домен, а компьютеры, на которых они установлены – в сеть Astra.Net.



Astra.Domain устанавливается на всех компьютерах, где есть исполняющие компоненты Astra.Server и Astra.AccessPoint.

После установки службы необходимо [сконфигурировать](#).



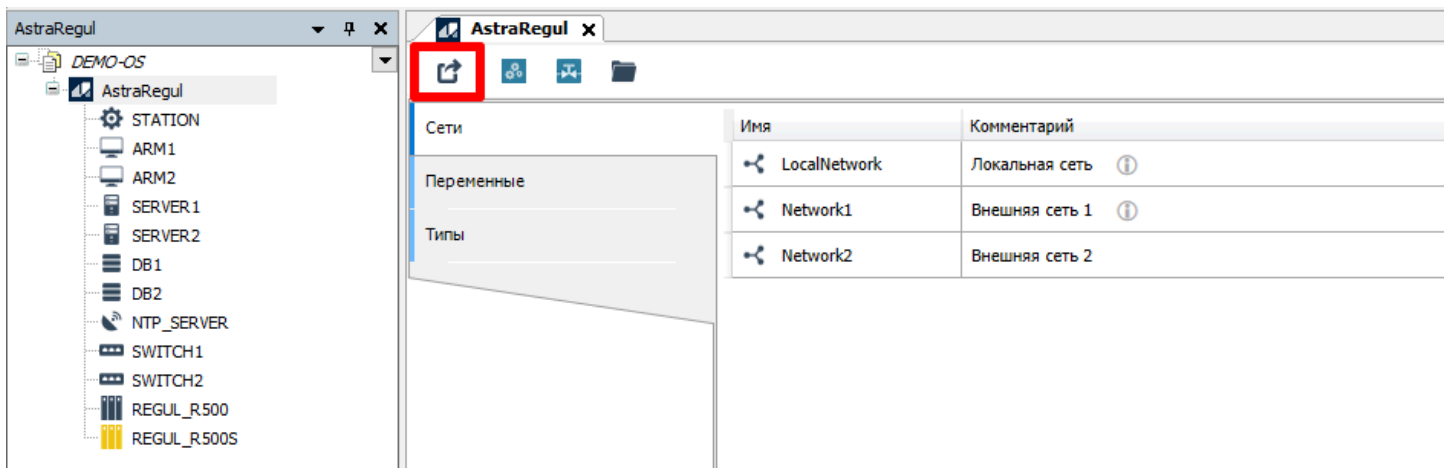
## 1.4.1. Конфигурирование

Чтобы настроить Astra.Domain, необходимо сконфигурировать его службы Astra.Net.Agent и Astra.Domain.Agent.

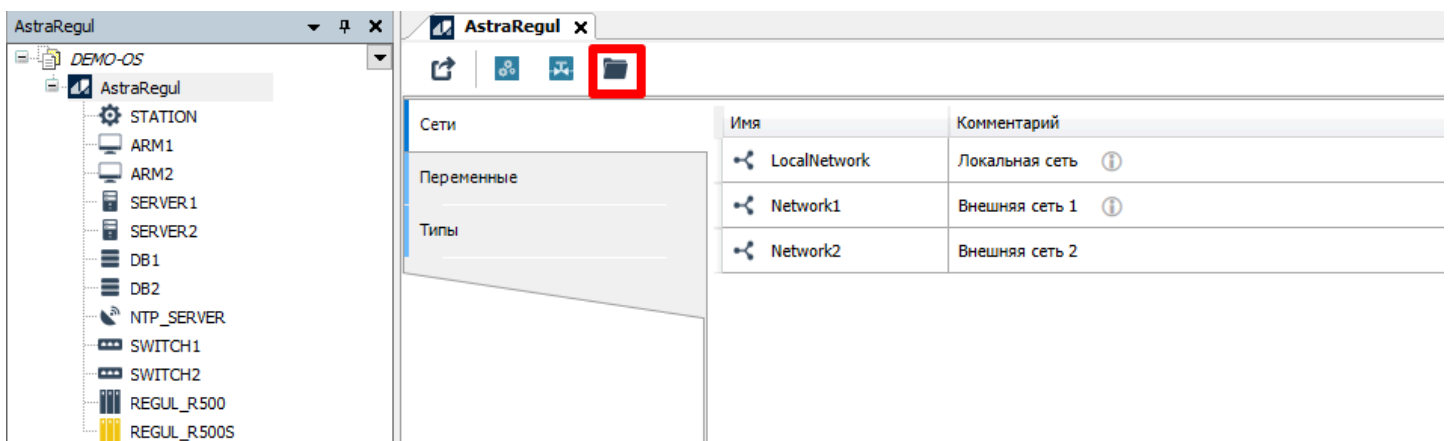
Конфигурационные файлы создаются автоматически при нажатии кнопки "Экспорт" в среде разработки Astra.IDE.



В Astra.IDE должен быть установлен плагин AstraRegul.



Чтобы открыть каталог с созданными конфигурационными файлами, нажмите кнопку "Открыть папку".



Сгенерированные файлы необходимо подложить на соответствующие компьютеры.



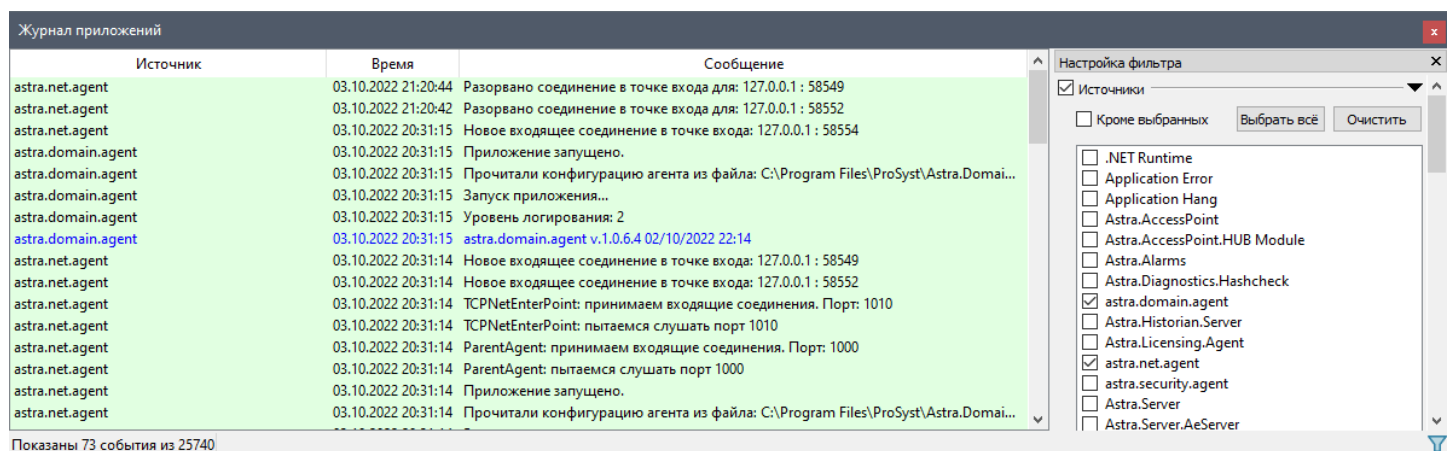
Чтобы применить изменения, перезапустите службы Astra.Net.Agent и Astra.Domain.Agent.

## 1.4.2. Диагностика работы

Диагностика работы Astra.Net.Agent и Astra.Domain.Agent осуществляется через системный журнал.

### Windows

Чтобы посмотреть системный журнал Windows, воспользуйтесь приложением EventLogViewer, входящим в набор инструментов Astra.Tools.



### Linux

Чтобы посмотреть системный журнал Linux, выполните команду (в зависимости от интересующей службы):

▶ `sudo journalctl -u Astra.Net -xe`

▶ `sudo journalctl -u Astra.Domain -xe`