

**НАСТРОЙКА ОБМЕНА  
ДАНЫМИ ПО ПРОТОКОЛУ  
IEC 60870-5-101 / IEC 60870-5-104  
НА КОНТРОЛЛЕРАХ СЕРИИ  
REGUL RX00**

**Руководство пользователя**

**DPA-302.2**

**Версия ПО 1.7.1.0**

**Версия 2.8**

**Август 2023**

## История изменений руководства пользователя

Версия руководства пользователя	Описание изменения
2.1	<p>Добавлена история изменений руководства пользователя.</p> <p>Добавлены знаки с предупреждающей и поясняющей информацией.</p> <p>Добавлены новые разделы:</p> <ul style="list-style-type: none"> <li>– «Создание списка элементов данных, получаемых по IEC-104» (Slave 104);</li> <li>– «Обращение в службу технической поддержки».</li> </ul> <p><i>Раздел «Настройка списка IP-адресов Slave 104 Driver»:</i> дополнено описание возможности настраивать порты вместе с IP-адресами.</p> <p><i>Раздел «Настройка общих параметров устройства Slave 104 Driver»:</i> добавлено описание новых параметров и возможность логирования команды синхронизации.</p> <p><i>Раздел «Создание списка команд, получаемых по IEC-104»:</i> добавлено описание реализации возможности сохранять принятое время в переменной, не выполняя синхронизацию.</p> <p><i>Разделы (Master/Slave) «Привязка переменных программы к элементам данных и командам»:</i> добавлено описание реализации возможности чтение/запись привязанных переменных по адресу (прямой доступ к памяти ПЛК), без использования механизма I/O Mapping (соотнесение входов/выходов).</p> <p><i>Раздел (Slave) «Автоматическая генерация переменных и привязка к каналам ввода-вывода»:</i> добавлено описание реализации возможности использования вызова системной функции спорадической отправки из прикладной программы.</p> <p><i>Раздел (Master) «Привязка переменных программы к элементам данных и командам»:</i> добавлено описание реализации возможности установки резервных битов в метке времени.</p> <p>Дополнительно по тексту внесены небольшие изменения с уточняющей информацией</p>
2.2	<p><i>Раздел «Настройка списка IP-адресов Slave 104 Driver»:</i> дополнено описание о возможности задания диапазонов для основных и резервных адресов.</p> <p><i>Раздел «Создание переменных для работы с данными/командами, передаваемыми по IEC 104» (для Slave/Master 104 и Master 101):</i> дополнено описание о возможности получения статуса канала (расширена диагностика канала передачи данных).</p> <p><i>Раздел «Настройка общих параметров устройства Unbalanced Secondary 101 Driver»:</i> добавлено описание нового параметра, определяющего задержку ответа на запрос при работе по RS-485.</p> <p><i>Раздел «Настройка общих параметров устройства IEC 101 Outer Slave»:</i> добавлено описание нового параметра, определяющего количество повторений запросов при возникновении ошибок.</p> <p>Дополнительно по тексту внесены небольшие изменения с уточняющей информацией</p>
2.3	<p><i>Раздел «Настройка общих параметров устройства Slave 104 Driver»:</i> дополнено описание о возможности не копировать сообщения в режиме «Стоп», даже при включенном параметре <b>Использовать накопление очереди сообщений</b>. Журналирование события переполнения очередей спорадических данных.</p> <p>Дополнительно по тексту внесены небольшие изменения с уточняющей информацией</p>

Версия руководства пользователя	Описание изменения
2.4	<p><i>Раздел «Настройка общих параметров устройства Slave 104 Driver»:</i> дополнено описание о возможности удаления очереди сообщений из приложения.</p> <p><i>Раздел «Настройка общих параметров устройства Unbalanced Secondary 101 Driver»:</i> добавлено описание новых параметров (Игнорировать команду time sync; Считать, что время в команде time sync задано в UTC; Mc, которые надо добавлять, учитывая задержку команды TIME_SYNC).</p> <p><i>Раздел «Настройка общих параметров устройства IEC 101 Outer Slave»:</i> добавлено описание нового параметра Мин. интервал между PDU, мс</p>
2.5	<p>Обновление в связи с выпуском среды разработки Astra.IDE.</p> <p>Добавлены новые разделы:</p> <ul style="list-style-type: none"> <li>– «Настройка контроллера в качестве Slave 104 OS»;</li> <li>– «Настройка контроллера в качестве Master 104 OS».</li> </ul> <p><i>Раздел «Настройка списка IP-адресов Slave 104 Driver»:</i> настройка диапазона IP-адресов перенесена со вкладки <b>Slave IEC 104 Конфигурация</b> на вкладку редактора <b>Slave 104 Driver</b>.</p> <p><i>Раздел «Создание списка элементов данных, передаваемых по IEC-104/101» (Master/Slave):</i> добавлено описание о проверке заданного значения адреса.</p> <p><i>Раздел «Привязка переменных программы к элементам данных и командам» (IEC-104/101 и Master/Slave):</i> на этапе компиляции добавлена проверка типов привязанных переменных.</p> <p>Дополнительно по тексту внесены небольшие изменения с уточняющей информацией</p>
2.6	<p>Добавлен новый раздел:</p> <ul style="list-style-type: none"> <li>– «Реализация общего опроса станции».</li> </ul> <p><i>Раздел «Настройка контроллера в качестве Slave 104 OS. Журналирование работы драйвера»:</i> добавлено описание установки уровней логирования через свойство LogLevel для ФБ устройств. Добавлено описание свойства ФБ ActiveConnections.</p> <p><i>Раздел «Настройка списка IP-адресов Slave 104 Driver»:</i> добавлена важная информация под знаком «ВНИМАНИЕ!», описывающая ограничение «один клиент - одно соединение»</p>
2.7	<p>Добавлены новые разделы:</p> <ul style="list-style-type: none"> <li>– «Автоматическая генерация переменных и привязка к каналам ввода-вывода» (для Slave 104 OS);</li> <li>– «Настройка контроллера в качестве Slave 101 OS»;</li> <li>– «Настройка контроллера в качестве Master 101 OS».</li> </ul> <p><i>Подраздел «Особенности работы с функциональными блоками данных»:</i> добавлено описание возможности чтения/записи всех флагов метки времени CP56Time2a.</p> <p><i>Раздел «Настройка общих параметров устройства Slave 104 Driver»:</i> добавлено описание нового параметра (Приоритет GI данных перед спонтанными данными).</p> <p>Дополнительно по тексту внесены небольшие изменения с уточняющей информацией</p>

<b>Версия руководства пользователя</b>	<b>Описание изменения</b>
<b>2.8</b>	Добавлена информация про использование идентичных переменных в разных экземплярах драйверов (Master/Slave, 101/104 и Master/Slave, 101/104 OS)

## АННОТАЦИЯ

Настоящий документ содержит сведения о настройке на контроллерах серии Regul RX00 передачи данных по стандартам ГОСТ Р МЭК 60870-5-104-2004 и ГОСТ Р МЭК 60870-5-101-2006. Настройка осуществляется с помощью программного обеспечения Astra.IDE.

Данное руководство предназначено для эксплуатационного персонала и инженеров-проектировщиков АСУ ТП, которые должны:

- иметь, как минимум, среднее техническое образование;
- приступить к работе только после изучения данного руководства.


### Обновление информации в Руководстве

Производитель ООО «РегЛаб» оставляет за собой право изменять информацию в настоящем Руководстве и обязуется публиковать более новые версии с внесенными изменениями. Обновленная версия Руководства доступна для скачивания на официальном сайте Производителя: <https://reglab.ru/>.


Для своевременного отслеживания выхода новой версии Руководства рекомендуется оформить подписку на обновление документа. Для этого необходимо на сайте Производителя: <https://reglab.ru/> кликнуть на кнопку «Подписаться на обновления» и оставить свои контактные данные.

В руководстве присутствуют знаки с предупреждающей и поясняющей информацией. Каждый знак обозначает следующее:

### ПРЕДУПРЕЖДАЮЩИЕ ЗНАКИ

	<p><b>ВНИМАНИЕ!</b> Здесь следует обратить внимание на способы и приемы, которые необходимо в точности выполнять во избежание ошибок при эксплуатации или настройке.</p>
---	--

### ИНФОРМАЦИОННЫЕ ЗНАКИ

	<p><b>ИНФОРМАЦИЯ</b> Здесь следует обратить внимание на <u>важную</u> информацию</p>
---	--

## СОДЕРЖАНИЕ

<b>АННОТАЦИЯ .....</b>	<b>5</b>
<b>ВВЕДЕНИЕ .....</b>	<b>8</b>
Общие сведения.....	8
Начало работы .....	8
<b>НАСТРОЙКА IEC-104.....</b>	<b>11</b>
Настройка контроллера в качестве Slave 104.....	11
Добавление устройства Slave 104 Driver .....	11
Настройка списка IP-адресов Slave 104 Driver .....	12
Настройка общих параметров устройства Slave 104 Driver .....	15
Создание списка элементов данных, передаваемых по IEC-104 .....	21
Создание списка элементов данных, получаемых по IEC-104 .....	24
Создание списка команд, получаемых по IEC-104 .....	25
Создание переменных для работы с данными/командами, передаваемыми по IEC-104 .....	28
Привязка переменных программы к элементам данных и командам IEC-104.....	31
Автоматическая генерация переменных и привязка к каналам ввода-вывода.....	34
Настройка контроллера в качестве Slave 104 OS.....	37
Настройка общих параметров устройства Slave 104 OS .....	37
Поведение в режиме СТОП и буферизация данных .....	40
Журналирование работы драйвера .....	42
Создание списка элементов данных, передаваемых по IEC-104 .....	44
Создание списка команд, получаемых по IEC-104 .....	47
Создание переменных для работы с данными/командами, передаваемыми по IEC-104 .....	49
Особенности работы с функциональными блоками данных .....	53
Автоматическая генерация переменных и привязка к каналам ввода-вывода.....	55
Реализация спорадического метода передачи .....	56
Реализация общего опроса станции.....	58
Особенности работы с функциональными блоками команд.....	58
Настройка контроллера в качестве Master 104.....	60
Добавление устройства Master 104 Driver.....	60
Настройка общих параметров устройства IEC 104 Outer Slave .....	61
Создание списка элементов данных, передаваемых по IEC-104 .....	63
Создание списка команд, передаваемых по IEC-104 .....	67
Создание переменных для работы с данными/командами, передаваемыми по IEC-104 .....	69

Привязка переменных программы к элементам данных и командам .....	70
Автоматическая генерация переменных и привязка к каналам ввода-вывода.....	74
Возможность разрыва соединения со стороны IEC-программы.....	75
<b>Настройка контроллера в качестве Master 104 OS .....</b>	<b>77</b>
Настройка общих параметров устройства Outer Slave 104 .....	77
Создание списка элементов данных, передаваемых по IEC-104 .....	80
Создание списка команд, передаваемых по IEC-104 .....	83
Создание переменных для работы с данными/командами, передаваемыми по IEC-104 .....	85
Особенности работы с функциональными блоками данных и команд .....	89
Журналирование работы драйвера .....	91
<b>НАСТРОЙКА IEC-101.....</b>	<b>93</b>
Добавление порта .....	93
Настройка параметров порта .....	95
<b>Настройка контроллера в качества Slave 101 .....</b>	<b>97</b>
Добавление устройства Unbalanced Secondary 101 Driver.....	97
Настройка общих параметров устройства Unbalanced Secondary 101 Driver .....	98
<b>Настройка контроллера в качества Slave 101 OS.....</b>	<b>101</b>
Добавление устройства Драйвер Unbalanced Secondary 101 OS.....	101
Настройка общих параметров устройства Драйвер Unbalanced Secondary 101 OS .....	102
Журналирование работы драйвера .....	104
<b>Настройка контроллера в качества Master 101.....</b>	<b>107</b>
Добавление устройства Unbalanced Primary 101 Driver.....	107
Настройка общих параметров устройства IEC 101 Outer Slave .....	108
<b>Настройка контроллера в качества Master 101 OS .....</b>	<b>111</b>
Добавление устройства Драйвер Unbalanced Primary Master 101 OS.....	111
Настройка общих параметров устройства Драйвер Outer Slave 101 .....	112
Журналирование работы драйвера .....	114
<b>ЭКСПОРТ И ИМПОРТ КОНФИГУРАЦИИ IEC 60870-5 .....</b>	<b>116</b>
Экспорт/импорт списка элементов данных.....	116
Экспорт/импорт списка команд.....	118
<b>ТЕСТИРОВАНИЕ СИСТЕМЫ, ВЫЯВЛЕНИЕ НЕИСПРАВНОСТЕЙ И ИХ УСТРАНЕНИЕ.....</b>	<b>119</b>
Общие действия.....	119
Использование программы OPC-104 для тестирования .....	121
<b>ОБРАЩЕНИЕ В СЛУЖБУ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ.....</b>	<b>125</b>

## ВВЕДЕНИЕ

### Общие сведения

Контроллеры серии Regul RX00 подключаются к сетям передачи данных посредством интерфейса RS-232/485 и/или Ethernet. Каждый интерфейс поддерживает стандартный протокол обмена данными: Ethernet - ГОСТ Р МЭК 60870-5-104-2004 (далее по тексту IEC-104) и RS - 232/485 - ГОСТ Р МЭК 60870-5-101-2006 (далее по тексту IEC-101). Протокол IEC-101 поддерживает только небалансную передачу данных в локальной сети.

Стандарты подробно описаны в документе «Устройства и системы телемеханики. Часть 5. Протоколы передачи» (см. раздел 101 и 104). Протокол IEC-104 является модификацией протокола IEC-101, реализует и поддерживая набор функций протокола IEC - 101 (прикладной уровень), но с расширенными возможностями. Протокол IEC-104 ориентирован на передачу данных по TCP/IP (транспортный уровень). Оба протокола располагают схожими параметрами.

Программное обеспечение контроллера Regul позволяет сконфигурировать его как в качестве *Master*-устройства, так и в качестве *Slave*-устройства.

### Перечень рекомендуемых документов

Для получения информации по настройке других параметров контроллеров серии Regul RX00 в среде разработки Astra.IDE рекомендуется ознакомиться со следующими документами:

- Программное обеспечение Astra.IDE. Руководство пользователя;
- Regul R600. Системное руководство;
- Regul R500. Системное руководство;
- Regul R400. Системное руководство;
- Regul R200. Системное руководство.

### Начало работы

Установите на компьютер программное обеспечение **Astra.IDE**. Описание процесса установки программы, а также инструкции по работе с программой приведены в документе «Программное обеспечение Astra.IDE. Руководство пользователя». Программа установки и документация доступны на сайте <https://reglab.ru/>.

Запустите программу **Astra.IDE**. Создайте проект в соответствии с аппаратной конфигурацией контроллера с помощью Мастера конфигурации Regul (инструкции приведены в документе «Программное обеспечение Astra.IDE. Руководство пользователя»).



В случае, когда контроллер Regul RX00 выступает в качестве Slave, для настройки обмена данными нужно выполнить следующие действия согласно таблице 1.

Таблица 1 – Настройка контроллера в качестве Slave устройства

Slave 104 Driver	Slave 101 Driver
<p>Последовательность действий:</p> <ul style="list-style-type: none"> <li>– добавить к контроллеру устройство Slave 104 Driver (OS) – драйвер, который принимает и обрабатывает входящие соединения от IEC-104 клиентов;</li> <li>– указать внешние IP-адреса клиентов, от которых драйверу разрешается принимать входящие запросы, и соответствующие этим адресам TCP порты, открытые драйвером на прослушивание;</li> <li>– настроить общие параметры устройства Slave 104 Driver (OS), отвечающие за то, как будет происходить обмен данными;</li> <li>– создать список элементов данных и список команд, передаваемых по протоколу IEC-104;</li> <li>– создать в программном коде переменные, представляющие из себя функциональные блоки определенных типов, описанных в библиотеке PsIecCommon (PsIEC60870Bridge). После чего связать переменные программы с элементами данных и командами. Предусмотрена возможность автоматической генерации переменных и их привязка</li> </ul>	<p>Последовательность действий:</p> <ul style="list-style-type: none"> <li>– добавить к контроллеру последовательный порт – Regul Serial Port;</li> <li>– настроить параметры последовательного порта;</li> <li>– добавить устройство Unbalanced Secondary 101 Driver (OS) – драйвер, который принимает и обрабатывает входящие соединения от IEC-101 клиентов;</li> <li>– настроить общие параметры устройства Unbalanced Secondary 101 Driver (OS), отвечающие за то, как будет происходить обмен данными;</li> <li>– создать список элементов данных и список команд, передаваемых по протоколу IEC-101;</li> <li>– создать в программном коде переменные, представляющие из себя функциональные блоки определенных типов, описанных в библиотеке PsIecCommon (PsIEC60870Bridge). После чего связать переменные программы с элементами данных и командами. Предусмотрена возможность автоматической генерации переменных и их привязка</li> </ul>

В случае, когда контроллер Regul RX00 выступает в качестве Master, для настройки обмена данными нужно выполнить следующие действия согласно таблице 2

Таблица 2 – Настройка контроллера в качестве Master

Master 104 Driver	Master 101 Driver
<p>Последовательность действий:</p> <ul style="list-style-type: none"> <li>– добавить к контроллеру устройство Master 104 Driver (OS) – драйвер, который взаимодействует с сервером IEC-104. Далее к драйверу добавить IEC 104 Outer Slave (OS) – компонент взаимодействующий с внешним (outer) подчиненным (slave) устройством;</li> <li>– настроить общие параметры устройства IEC 104 Outer Slave (OS), отвечающие за то, как будет происходить обмен данными</li> </ul>	<p>Последовательность действий:</p> <ul style="list-style-type: none"> <li>– добавить к контроллеру последовательный порт – Regul Serial Port;</li> <li>– настроить параметры последовательного порта;</li> <li>– добавить устройство Unbalanced Primary 101 Driver (OS) – драйвер, который взаимодействует с сервером IEC-101; к нему добавить подчиненное устройство IEC 101 Outer Slave (OS);</li> </ul>

<b>Master 104 Driver</b>	<b>Master 101 Driver</b>
<p>по протоколу, включая IP-адреса (основной и резервный) и TCP порт удаленного сервера IEC-104;</p> <ul style="list-style-type: none"> <li>– создать список элементов данных и список команд, передаваемых по протоколу IEC-104;</li> <li>– создать в программном коде переменные, представляющие из себя функциональные блоки определенных типов, описанных в библиотеке PsIecCommon (PsIEC60870Bridge). После чего связать переменные программы с элементами данных и командами. Предусмотрена возможность автоматической генерации переменных и их привязка</li> </ul>	<ul style="list-style-type: none"> <li>– настроить общие параметры устройства IEC 101 Outer Slave (OS), отвечающие за то, как будет происходить обмен данными;</li> <li>– создать список элементов данных и список команд, передаваемых по протоколу IEC-101;</li> <li>– создать в программном коде переменные, представляющие из себя функциональные блоки определенных типов, описанных в библиотеке PsIecCommon (PsIEC60870Bridge). После чего связать переменные программы с элементами данных и командами. Предусмотрена возможность автоматической генерации переменных и их привязка</li> </ul>

## НАСТРОЙКА IEC-104



### ИНФОРМАЦИЯ

Начиная с версии СПО 1.7.0.0, доступна нативная версия драйвера с дополнением OS, реализованная на уровне операционной системы

## Настройка контроллера в качестве Slave 104

Контроллер выступает в роли ведомого, опрашиваемого устройства.

### Добавление устройства Slave 104 Driver

В окне дерева устройств поместите курсор на название контроллера, нажмите правую кнопку мыши. В появившемся контекстном меню выберите пункт **Добавить устройство...** Откроется окно **Добавить устройство**, в котором выберите *Regul* → *МЭК 60870* → *TCP 60870-104-Slave* → *Slave 104 Driver*. Нажмите кнопку **Добавить устройство** или дважды щелкните левой кнопкой мыши. Выбранное устройство появится в проекте в дереве устройств (Рисунок 1).

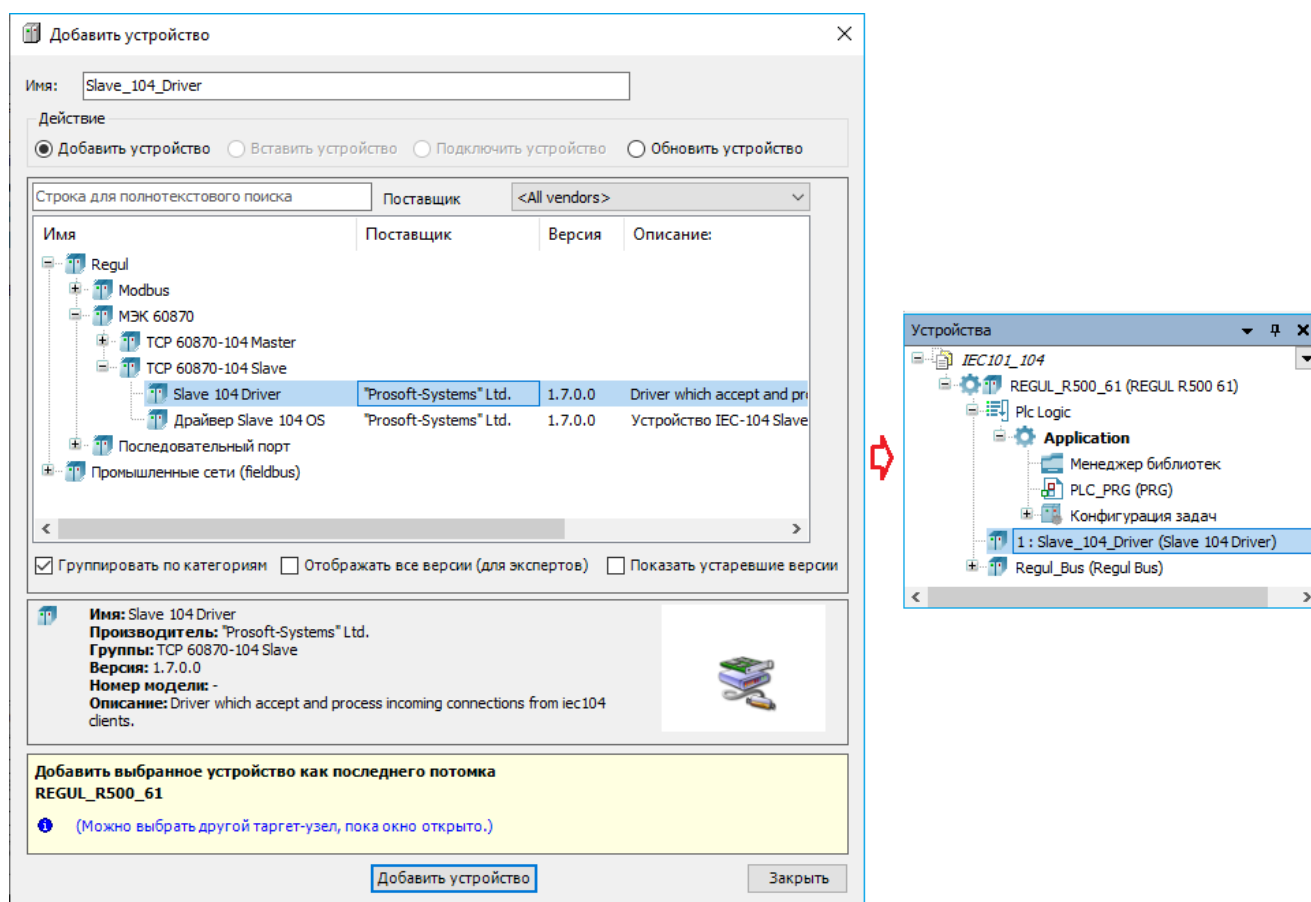


Рисунок 1 – Добавление в конфигурацию контроллера устройство Slave 104 Driver

Двойным щелчком по названию устройства **Slave 104 Driver** откройте вкладку параметров. По умолчанию открывается первая внутренняя вкладка **Редактор устройства IEC 104 Slave** (Рисунок 2).

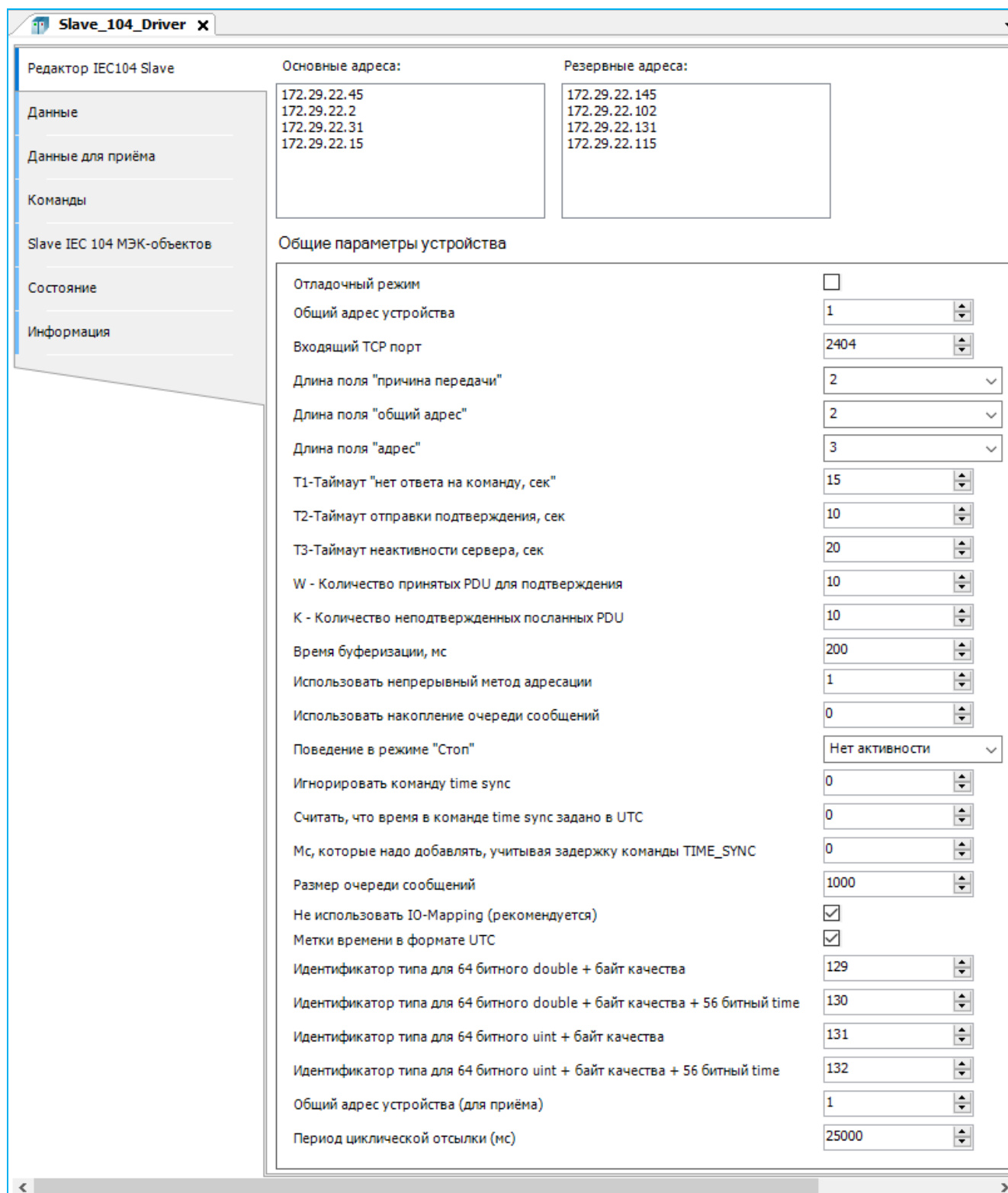


Рисунок 2 – Редактор устройства IEC104 Slave

### Настройка списка IP-адресов Slave 104 Driver

В редакторе устройства **Slave 104 Driver** в верхней части вкладки находятся два блока: **Основные адреса** и **Резервные адреса**. Здесь указывается список IP-адресов, от которых разрешается принимать входящие запросы, иначе говоря, список разрешенных «мастеров» IEC-104.



### ИНФОРМАЦИЯ

Каждое соединение поддерживает отдельный буфер очереди данных. Для нормальной работы рекомендуем указывать не более 15 каналов основной/резервный для соединения

Каждый мастер может вести опрос по основному или резервному каналу (в приоритете основной канал, мастер переключается на обмен с резервным, только если нет связи с основным). Добавить идентичные IP-адреса в основной и резервный список не получится.

Для добавления основного или резервного IP-адреса поместите курсор в блок адресов, нажмите правую кнопку мыши. В появившемся контекстном меню выберите пункт **Добавить новый IP**. Появится дополнительное окно, в котором нужно ввести IP-адрес (Рисунок 3).

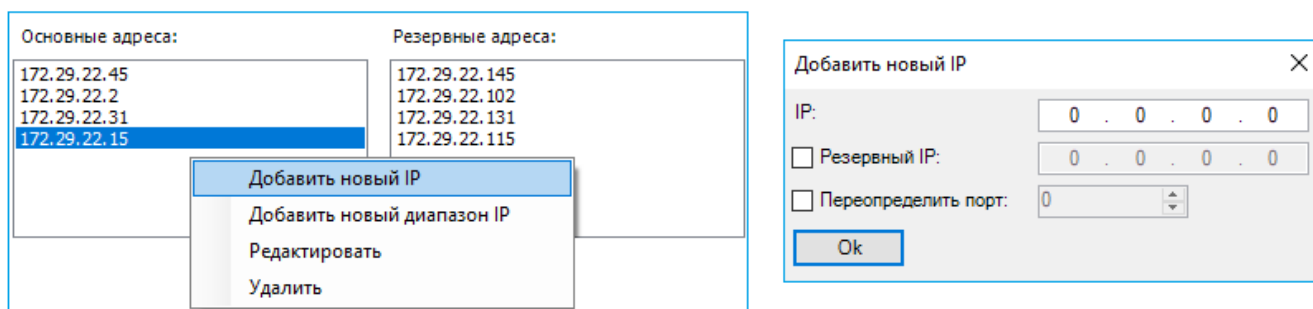


Рисунок 3 – Добавление нового IP-адреса

Также, кроме IP-адреса, можно задать номер порта соединения, для этого необходимо установить флажок в поле параметра **Переопределить порт** и указать необходимый номер в поле (Рисунок 4). Можно указывать несколько комбинаций IP-адресов с номерами портов соединения. Существует ограничение на количество прослушиваемых портов, не более 10.

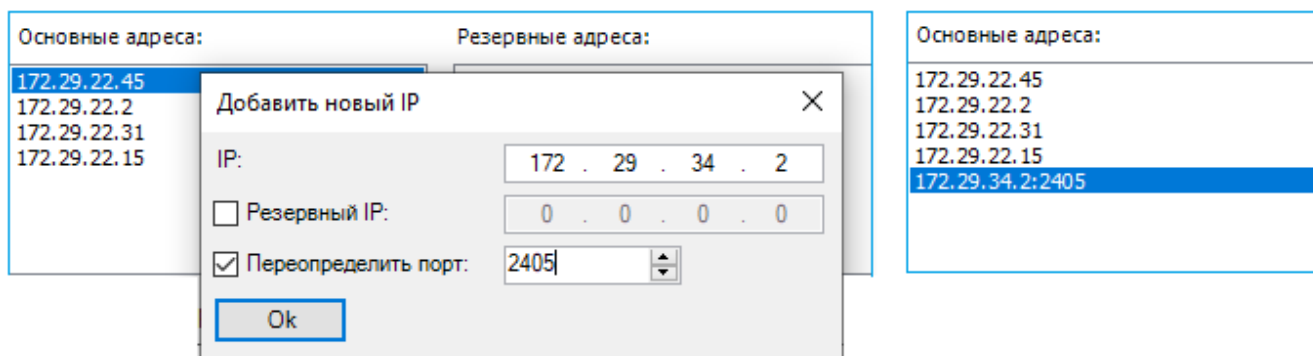


Рисунок 4 - Присвоение номера порта для IP-адреса




### ИНФОРМАЦИЯ

Число резервных адресов не может быть больше, чем основных

Если опрос мастером ведется только по одному каналу, то его IP адрес следует поместить в конец списка **Основные адреса**, резервные адреса для него не указывать.

Для редактирования адреса выберите его в списке, вызовите контекстное меню, пункт **Редактировать**. Появится дополнительное окно, в котором можно изменить IP-адрес. Для удаления адреса выберите его в списке, вызовите контекстное меню, пункт **Удалить**.

	<p><b>ВНИМАНИЕ!</b></p> <p>Программа не запрашивает подтверждение на удаление</p>
---	---

Есть возможность задавать диапазон адресов на вкладке редактора устройства. Для добавления диапазонов IP-адресов поместите курсор в блок адресов, нажмите правую кнопку мыши. В появившемся контекстном меню выберите пункт **Добавить новый диапазон IP** (Рисунок 5).

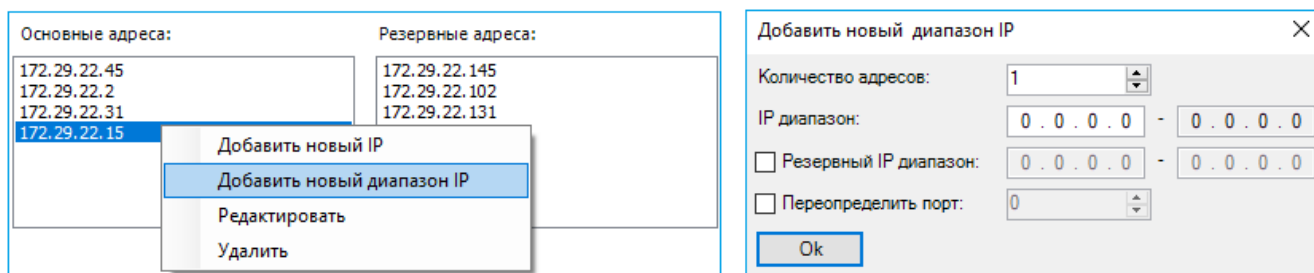


Рисунок 5 – Добавление нового диапазона IP-адресов

При задании диапазона IP-адресов необходимо учесть следующее:

- первым задается начальный адрес диапазона, затем, через дефис, конечный адрес;
- в поле параметра **Количество адресов** задается одинаковое количество IP-адресов в основном и резервном диапазоне (ограничение – до 255);
- можно задать номер порта соединения, для этого необходимо установить флажок в поле параметра **Переопределить порт** и указать необходимый номер в поле (задавать не обязательно). В случае отсутствия заданного значения порта, принимается значение по умолчанию;
- первые три байта IP-адреса начала диапазона должны быть идентичны первым трем байтам конца диапазона.

Пример:

‘174.29.35.1-174.29.35.5:2405’, первые три байта IP-адреса – «174.29.35».

При некорректном вводе, в журнале появится сообщение следующего вида:

«Addr. range 172.29.34.1-172.29.35.5 first 3 bytes differ».

Диапазоны могут быть заданы для основных и резервных адресов. Например, указаны следующие адреса:

Основные адреса	Резервные адреса
172.29.34.1	172.29.34.3
174.29.35.1-174.29.35.5	174.29.35.11-174.29.35.15
175.30.40.1-175.30.40.5*	

\* - Диапазон 175.30.40.1-175.30.40.5 задает адреса основных каналов, не имеющих парных резервных. Такие диапазоны следуют последними в списке адресов

Пары [основной ↔ резервный] в этом случае будут следующие:

Соответствующие пары	
172.29.34.1	172.29.34.3
174.29.35.1	174.29.35.11
174.29.35.2	174.29.35.12
174.29.35.3	174.29.35.13
174.29.35.4	174.29.35.14
174.29.35.5	174.29.35.15

При формировании пар диапазонов основной ↔ резервный необходимо учесть следующее:

- множество адресов в основном диапазоне не должно пересекаться со множеством адресов в резервном диапазоне. При некорректном вводе (например, основной: '172.29.34.1-172.29.34.9 и резервный: '172.29.34.8-172.29.34.16'), основные соединения будут приниматься, а резервные нет.



**ВНИМАНИЕ!**

В связи с тем, что единственным идентификатором клиента (мастера) является его IP-адрес, и входящие соединения с одного IP-адреса (АРМа) невозможно соотнести с конкретной системой опроса, используя только средства протокола IEC-60870, то со стороны драйвера Slave-104 для каждой резервной пары соединений допускается поддержание установления только одного подключения со стороны, как основного, так и резервного адресов. Если с одного и того же IP-адреса происходит повторная попытка подключения, то оно становится активным, а предыдущее подключение драйвером Slave-104 принудительно закрывается

**Настройка общих параметров устройства Slave 104 Driver**

В редакторе устройства **Slave 104 Driver** в блоке **Общие параметры устройства** доступны для настройки параметры, представленные в таблице 3.

Таблица 3 – Общие параметры устройства

Параметр	Описание	Значение по умолчанию
<b>Отладочный режим</b>	При установке флажка <input checked="" type="checkbox"/> в этом поле будут записаны в журнал работы контроллера все детали обмена по IEC-104, а именно передаваемые и принимаемые пакеты, комментарии к возникающим ошибкам и т.п.	
<b>Общий адрес устройства</b>	Содержит общий адрес (COMMON ADDRESS, ASDU ADDRESS) устройства, все отдаваемые данные и выполняемые команды имеют общую часть, равную значению этого параметра (см. спецификацию IEC-104)	1
<b>Общий адрес устройства (для приема)</b>	Содержит общий адрес (COMMON ADDRESS, ASDU ADDRESS) устройства, все получаемые данные имеют общую часть, равную значению этого параметра (см. спецификацию IEC-104)	1
<b>Входящий TCP порт</b>	Номер порта TCP, который открыт данным slave для приема входящих соединений	2404
<b>Длина поля «причина передачи»</b>	Содержит длину поля «причина передачи» (COT, Cause Of Transmission). Допустимые значения параметра: 1 или 2	2
<b>Длина поля «общий адрес», Длина поля «адрес»</b>	Характеристики PDU (Protocol Data Unit – протокольная единица, пакет), передаваемых по IEC-104, должны быть одинаковы для пары master slave. Допустимые значения параметра «общий адрес»: 1 или 2. Допустимые значения параметра «адрес»: 2 или 3	«общий адрес» - 2, «адрес» - 3
<b>T1-Таймаут «нет ответа на команду», сек</b>	Время, в течение которого ожидается ответ на команду или подтверждение доставки PDU. Превышение таймаута приводит к разрыву соединения	15
<b>T2-Таймаут отправки подтверждения, сек</b>	При получении PDU запускается таймер на T2 секунд, подтверждение о приеме (пакет S-PDU) отсылается либо по истечению этого таймера, либо по факту приема некоторого предельного количества PDU, указанного в параметре W. T2 должен быть меньше T1	10
<b>T3-Таймаут неактивности сервера, сек</b>	Периодически как master, так и slave могут отсылать специальные пакеты для проверки связи. Если в течение интервала, заданного в параметре T3, нет никакого обмена, то slave может закрыть соединение	20
<b>W-Количество принятых PDU для подтверждения</b>	Количество принятых PDU, которое необходимо подтвердить. Рекомендуем выбирать значение W, не превышающее 2/3 от значения K (Master). Параметры должны быть согласованы на Master и Slave. Например, при K=10 на Master, W=8 на Slave	10
<b>K-Количество неподтвержденных посланных PDU</b>	Количество неподтвержденных отправленных PDU, после которого драйвер прекращает отправлять PDU, дожидаясь подтверждения отправленных PDU.  Следует отметить, что в случае, когда требуется отправка	10



Параметр	Описание	Значение по умолчанию
	<p>большого количества данных, значение параметра К следует устанавливать большим. Например, если требуется отправка 20000 отсчетов в секунду, значение К следует увеличивать до 150. Иначе драйвер будет приостанавливать отставку данных, ожидая подтверждение клиента, размеры очередей при этом будут нарастать.</p> <p>Для оптимальной работы драйвера рекомендуется ограничить отставку на уровне 25000 отсчетов в секунду. Отсчет – изменение одного элемента данных. Если введено 500 элементов данных и все они изменяются раз в 10 мс, то это равносильно <math>500 \cdot 1000 / 10 = 50000</math> отсчетов в секунду</p>	
<b>Время буферизации, мс</b>	Период краткосрочного ожидания пополнения очереди на отставку. Используется для накопления в одном PDU нескольких данных одного типа, для оптимизации	200
<b>Период циклической отсылки (мс)</b>	При описании элемента данных IEC-104 можно включить его в циклическую рассылку (установка флажка в поле <b>Участвует в циклической рассылке</b> ). Для всех таких элементов их значения, даже при отсутствии изменений, будут отсылаться с указанным здесь периодом (в миллисекундах)	25000
<b>Период циклической отсылки, сек</b>	При описании элемента данных IEC-104 можно включить его в циклическую рассылку (установка флажка в поле <b>Участвует в циклической рассылке</b> ). Для всех таких элементов их значения, даже при отсутствии изменений, будут отсылаться с указанным здесь периодом (в секундах). Данный параметр оставлен в целях обратной совместимости. Если значение параметра <b>Период циклической отсылки (мс)</b> отлично от нуля, то данный параметр игнорируется. Начиная с версии СПО 1.6.5 параметр скрыт	25
<b>Использовать непрерывный метод адресации</b>	Экономный режим IEC-104, в отсылаемых пакетах указывается адрес только первого элемента, считаем адреса остальных последовательно нарастающими. 0 – не используется, 1 – использовать этот метод	1
<b>Использовать* накопление очереди сообщений</b>	Накапливать данные при отсутствии связи с мастером. 0 – не используется (сбрасывается очередь данных после восстановления связи). Любое число, отличное от 0, активирует использование накопления сообщений с ограничением, которое задается параметром <b>Размер очереди сообщений</b>	0
<b>Поведение в режиме «Стоп»**</b>	<p>Определяет поведение компонента при остановке программы.</p> <p><i>Нет активности</i> - означает, что устройство Slave104 прекращает все соединения и не воспринимает новые;</p> <p><i>Нормальная работа</i> - означает продолжение работы в обычном режиме</p>	Нет активности

Параметр	Описание	Значение по умолчанию
<b>Игнорировать команду time sync</b>	Если значение не равно 0, то при получении команды синхронизации времени (C_CS_NA_1), синхронизации времени не происходит. Если этот параметр равен 0, то команда синхронизации времени выполняется	0
<b>Считать, что время в команде time sync задано в UTC</b>	Если в этом поле задано значение 1, то, при синхронизации времени, время в составе команды будет интерпретировано, как UTC. Если задано значение 0, то, это время будет интерпретировано, как локальное время	0
<b>Миллисекунды, которые надо добавлять, учитывая задержку команды TIME_SYNC</b>	Корректировка команды синхронизации времени	0
<b>Размер очереди сообщений</b>	Определяет размер буфера сообщений для пересылки в нормальном режиме, а также при отсутствии связи. Следует избегать установки размера очереди сообщений более 50000 сообщений, поскольку в этом случае возникает риск исчерпания резерва оперативной памяти. Вместе с тем, размер очереди должен быть не меньше планируемого количества элементов данных. Это связано с тем, что при внешнем подключении IEC-104 Slave может получить команду общего опроса, что приведет к одновременному заполнению очереди последними значениями всех элементов данных. Если размер очереди меньше рекомендуемого, то часть данных не будет отправлена	1000
<b>Не использовать IO-Mapping (рекомендуется)</b>	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что чтение/запись привязанных переменных происходит по адресу (прямой доступ к памяти ПЛК), без использования механизма I/O Mapping (соотнесение входов/выходов)	
<b>Метки времени в формате UTC</b>	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что значение метки времени соответствует UTC, иначе - локального времени	
<b>Идентификатор типа для 64 битного double + байт качества</b>	Задание ID для пользовательского типа LREAL(double), без метки времени	129
<b>Идентификатор типа для 64 битного double + байт качества + 56 битный time</b>	Задание ID для пользовательского типа LREAL(double), с меткой времени	130
<b>Идентификатор типа для 64 битного uint + байт качества</b>	Задание ID для пользовательского типа ULINT(uint), без метки времени	131
<b>Идентификатор типа для 64 битного uint</b>	Задание ID для пользовательского типа ULINT(uint), с	132

Параметр	Описание	Значение по умолчанию
+ байт качества + 56 битный time	меткой времени	
<p><b>Примечания</b></p> <p>*- предусмотрена возможность отключения накопления очереди сообщений, когда ведомый контроллер находится в режиме «Стоп». То есть, если задан параметр <b>Использовать накопление очереди сообщений</b> (любое число, отличное от 0) и ведомый контроллер остановлен, то, при возобновлении работы и подключении к нему, устаревшие данные не будут поступать. Ведущий контроллер, при потерях связи, будет накапливать очередь сообщений и при подключении поступят все данные.</p> <p>1) Если требуется отключить накопление сообщений в режиме «Стоп», то пользователь может использовать следующий оператор:</p> <pre>Slave_104_Driver.m_outer.set_not_accum_in_stop(not_accum_in_stop);</pre> <p>После вызова этого оператора в Slave_104_Driver не будут копиться отчеты в режиме «Стоп».</p> <p>2) Если потребуется очистить очередь сообщений из приложения, то пользователь может использовать следующий метод:</p> <pre>Slave_104_Driver.purge_queue();</pre> <p>После вызова этого метода в Slave_104_Driver будет удалено содержимое буфера сообщений.</p> <p>** - предусмотрена возможность самостоятельно активировать «поведение в режиме «Стоп»» в программном коде. Для активации режима требуется в программе присвоить значение TRUE свойству ActivateStopBehavior необходимого устройства:</p> <pre>«IEC104Slave_device_name».ActivateStopBehavior := TRUE;</pre> <p>После этого slave-устройство перейдет в Стоп-режим работы</p>		

Предусмотрено журналирование событий синхронизации времени при получении команды синхронизации (time sync). Клиент посылает команду синхронизации и на основной вкладке параметров устройства во вкладке **Журнал**, в зависимости от выбранного значения параметра **«Игнорировать команду time sync»**, должны появиться сообщения следующего вида: «Set system clock to ...», «Set system clock succeeded» или «Set system clock failed» (Рисунок 6).

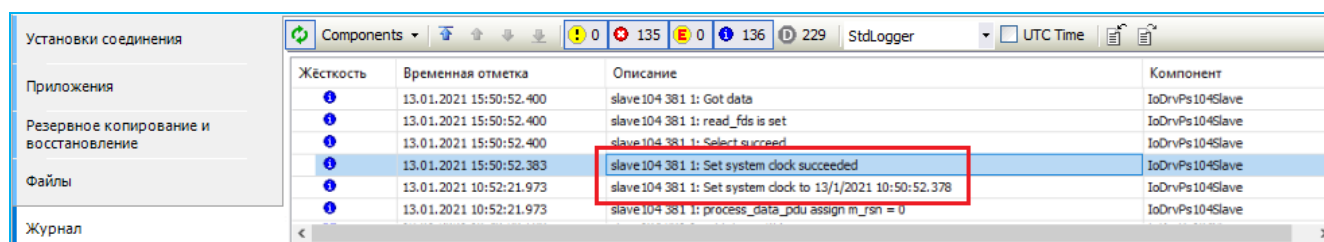


Рисунок 6 - Журналирование команды синхронизации времени

Также предусмотрено журналирование события переполнения очередей спорадических данных. Возможно два случая переполнения очереди:

- переполняется очередь изменений данных (одна очередь для всех соединений) и должно появиться сообщение следующего вида:  
«Slave\_104\_Driver change queue overflow»
- переполняется очередь спонтанных сообщений конкретной пары соединений основное-резервное и должно появиться сообщение следующего вида:

«Slave\_104\_Driver 192.168.0.10:2404 queue overflow».

На вкладке **Slave IEC 104 Конфигурация**, можно задать значение параметра **Приоритет GI данных перед спонтанными данными** (Рисунок 7).

Параметр	Тип	Значение	Значение по умолчанию	Единица
Считать, что время в команде time sync задано в UTC	DWORD	0	0	
мс, которые надо добавлять, учитывая задержку команды TIME_SYNC	DWORD	0	0	
Идентификатор типа для 64 битного double + байт качества	BYTE	129	129	
Идентификатор типа для 64 битного double + байт качества + 56 битный time	BYTE	130	130	
Идентификатор типа для 64 битного uint + байт качества	BYTE	131	131	
Идентификатор типа для 64 битного uint + байт качества + 56 битный time	BYTE	132	132	
Метки времени в формате UTC	BOOL	True	True	
Общий адрес устройства (для приёма)	WORD	1	1	
Период циклической отсылки (мс)	DWORD	25000	25000	
Starting ParameterId of new data descriptions (input)	DWORD	84000	84000	
<b>Приоритет GI данных перед спонтанными данными</b>	<b>BOOL</b>	<b>True</b>	<b>True</b>	
Размер очереди сообщений	WORD	1000	1000	

Рисунок 7 - Вкладка Slave IEC 101 Конфигурация

Двойной щелчок левой кнопкой мыши по ячейке в столбце **Значение** в строке **Приоритет GI данных перед спонтанными данными** позволяет задать необходимое значение.

Если значение параметра TRUE (значение по умолчанию), то данные ответа на общий опрос станции отсылаются сразу после получения запроса от клиента.

Например:

Исходное состояние:

- последовательность спонтанных данных на момент GI запроса: **1, 2, 3, 4, 5, 6, 7;**
- в момент получения запроса GI очередь: **4, 5, 6, 7;**
- текущее состояние данных: **7.**

В ответе на GI отсылается текущее значение: **7**, а далее досылаются данные из очереди: **4, 5, 6, 7**. За время этих отсылок возникли новые спонтанные данные: **8, 9, 10**.

В итоге отсылка данных:

**1 sp, 2 sp, 3 sp, 7 GI, 4 sp, 5 sp, 6 sp, 7 sp, 8 sp, 9 sp, 10 sp**

где sp - спонтанные данные, GI - данные ответа на запрос GI.

Если значение параметра FALSE, то данные ответа на общий опрос станции отсылаются после спонтанных данных находящихся в очереди, в момент получения запроса GI. Спонтанные данные, возникающие после получения запроса GI, отсылаются после ответа на GI.

В итоге отсылка данных:

**1 sp, 2 sp, 3 sp, 4 sp, 5 sp, 6 sp, 7 sp, 7 GI, 8 sp, 9 sp, 10 sp**



### ИНФОРМАЦИЯ

Если вкладка **Slave IEC 104 Конфигурация** отсутствует, то необходимо перейти в меню **Инструменты** ⇒ **Опции** ⇒ найти пункт **Редактор устройств** и установить флажок напротив поля **Показывать общие окна конфигурации устройств**

### Создание списка элементов данных, передаваемых по IEC-104

В редакторе устройства **Slave 104 Driver** перейдите на внутреннюю вкладку **Данные** (Рисунок 8).

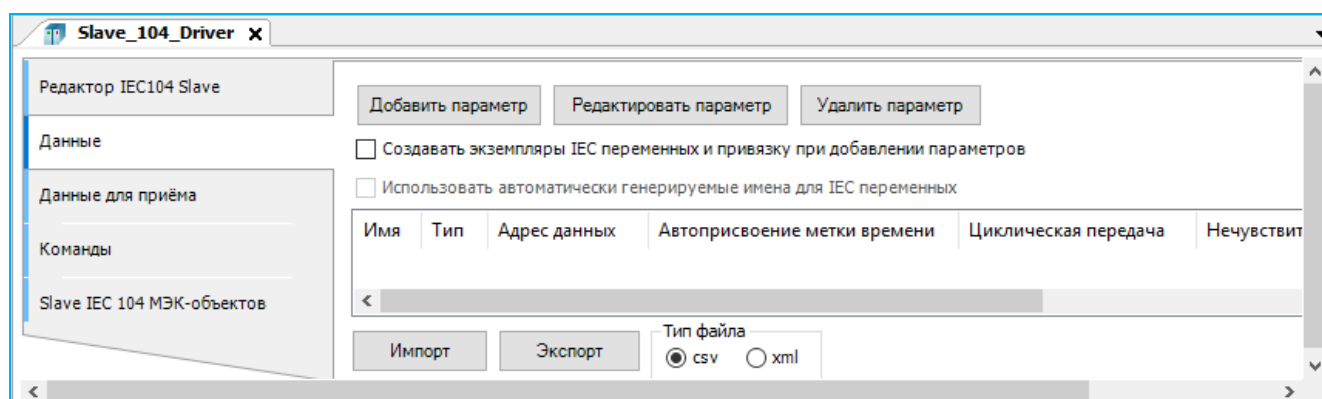


Рисунок 8 – Редактор Slave 104 Driver. Вкладка «Данные»

Элементы данных описываются как параметры. Для добавления, изменения и удаления элементов данных используются соответственно кнопки **Добавить параметр**, **Редактировать параметр**, **Удалить параметр**. Перейти к редактированию параметра также можно двойным щелчком левой кнопкой мыши по нужной строке. Окна ввода/редактирования элемента данных выглядят следующим образом (Рисунок 9).

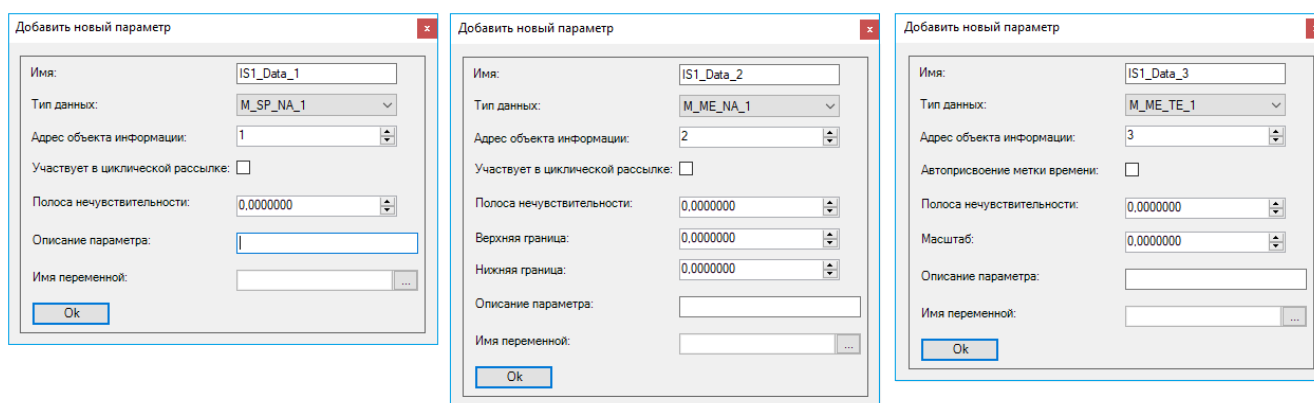




Рисунок 9 – Окна добавления нового параметра (элемента данных)

Для элемента данных укажите значения в полях. Параметры элементов представлены в таблице 4.

Таблица 4 – Параметры элементов данных IEC-104

Параметр	Описание
<b>Имя</b>	Наименование элемента данных
<b>Тип данных</b>	Ассоциированный с этим элементом данных тип IEC-104. Справочные типы данных IEC-104 приведены в таблице 5. Часто используемые типы данных: <ul style="list-style-type: none"> <li>o M_SP_TB_1 – для дискретов с меткой времени;</li> <li>o M_ME_TF_1 – для float-значений с меткой времени;</li> <li>o M_IT_TB_1 – для целочисленных значений</li> </ul>
<b>Адрес объекта информации</b>	Уникальный адрес элемента данных. Значение адреса не должно превышать максимально допустимого значения, которое обуславливается размером адресного поля PDU. Размер адресного поля PDU определяется параметром <b>Длина поля «адрес»</b> . Например, если задано значение длины поля адреса равное 2, то максимально допустимое значение адреса не может превышать 65535 (для значения 3 - не может превышать 16777215). При некорректном вводе, в логе появится сообщение о переполнении адреса, например: «Address overflow: 100001 max: 65535»
<b>Участвует в циклической рассылке</b>	Установка флажка в этом поле активирует участие элемента данных в циклической рассылке. Для всех таких элементов их значения даже при отсутствии изменений будут отсылааться с периодом (в сек.), указанным на вкладке общих параметров
<b>Полоса нечувствительности (deadband)</b>	Устанавливаемая пользователем минимальная разница между последним отосланным значением и текущим измеренным (рассчитанным) в программе контроллера, превышение которой вызывает спонтанную отсылку
<b>Верхняя граница, Нижняя граница</b>	Параметры, задающие диапазон для нормализованных типов (M_ME_NA_1, M_ME_TA_1, M_ME_ND_1, M_ME_TD_1)
<b>Автоприсвоение времени (для типов с меткой времени)</b>	При установленном в этом поле флажке не требуется явно задавать временную метку для отсылаемого значения элемента данных. Временная метка будет <i>автоматически</i> задаваться в локальном времени и присваиваться <b>m_timestamp</b> . При отсутствии флажка, пользователь <i>вручную</i> в программе присваивает <b>m_timestamp</b> (с возможностью смещения часового пояса для локального времени). Если не присвоить <b>m_timestamp</b> , то будет передаваться нулевая временная метка. Значение метки времени соответствует UTC или локальному времени в зависимости от параметра « <b>Метки времени в формате UTC</b> »
<b>Масштаб</b>	Для масштабируемых параметров, при выборе типа M_ME_TE_1

Параметр	Описание
Описание параметра	Опционально, текстовое описание элемента данных
Имя переменной*	<p>Указывается название переменной IEC (программы контроллера), в которой хранятся передаваемые данные.</p> <p>Для заполнения поля <b>Имя переменной</b> нажмите в этом поле кнопку , открывающую окно <b>Ассистент ввода</b>. Найдите нужную переменную. Если установлен флажок в поле <b>Структурированный вид</b>, то раскрывайте списки с помощью кнопки . Если флажок снят и переменные представлены одним большим списком, для удобства поиска воспользуйтесь фильтром</p>
<p><b>Примечание:</b> -* - параметр присутствует в окне ввода/редактирования элемента данных, когда привязка переменных происходит по адресу (прямой доступ к памяти ПЛК), без использования механизма I/O Mapping (соотнесение входов/выходов), при условии, что не снят флажок с параметра «<b>Не использовать IO-Mapping</b>»</p>	

Реализованы следующие идентификаторы типа, представленные в таблице 5

Таблица 5 – Перечень типов данных IEC-104

Тип данных	ID типа данных	Описание
M_SP_NA_1	1	Одноэлементная информация
M_DP_NA_1	2	Двухэлементная информация
M_ST_NA_1	5	Информация о положении отпаяк
M_BO_NA_1	7	Строка из 32 бит
M_ME_NA_1	9	Измеряемая величина, нормализованное значение
M_ME_NB_1	11	Измеряемая величина, масштабируемое значение
M_ME_NC_1	13	Измеряемая величина, формат с плавающей запятой
M_IT_NA_1	15	Интегральная сумма
M_ME_ND_1	21	Измеряемая величина, нормализованное значение без описателя качества
M_SP_TB_1	30	Одноэлементная информация с 56-битной меткой времени
M_DP_TB_1	31	Двухэлементная информация с 56-битной меткой времени
M_ST_TB_1	32	Информация о положении отпаяк с 56-битной меткой времени
M_BO_TB_1	33	Строка из 32-х бит с 56-битной меткой времени
M_ME_TD_1	34	Измеряемая величина нормализованное значение с 56 битной меткой времени
M_ME_TE_1	35	Измеряемая величина масштабированное значение с 56 битной меткой времени
M_ME_TF_1	36	Измеряемая величина с плавающей запятой с 56 битной меткой времени
M_IT_TB_1	37	Интегральная сумма с 56-битной меткой времени

Тип данных	ID типа данных	Описание
M_EP_TD_1	38	Информация о работе релейной защиты с 56-битной меткой времени
M_LR_NA	129*	Информация пользовательского типа LREAL из 64-х бит + байт качества, без метки времени
M_LR_TC	130*	Информация пользовательского типа LREAL из 64-х бит + байт качества с 56-битной меткой времени
M_ULINT_NA	131*	Информация пользовательского типа ULINT из 64-х бит + байт качества, без метки времени
M_ULINT_TC	132*	Информация пользовательского типа ULINT из 64-х бит + байт качества с 56-битной меткой времени

**Примечание** - \* - ID задается пользователем

Заполненная вкладка **Данные** показана на рисунке 10.

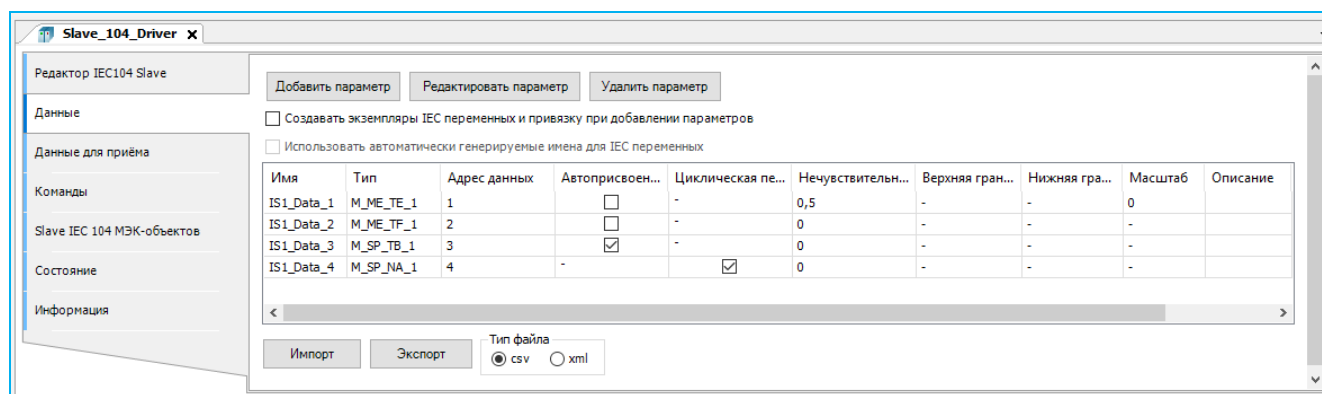


Рисунок 10 – Пример списка элементов данных

### Создание списка элементов данных, получаемых по IEC-104

Наряду с отправкой изменившихся данных по протоколу IEC 60870-104, объект Slave\_104\_Driver также может и принимать данные от другой стороны обмена и, в соответствии с принятой протокольной информацией, изменять значения переменных приложения.

В то время как на вкладке **Данные** пользователь может конфигурировать передаваемые по протоколу данные, то на вкладке **Данные для приема** можно сконфигурировать данные, которые получаются по протоколу. Вид и назначение полей на вкладках **Данные** и **Данные для приема** идентичны (Рисунок 11).



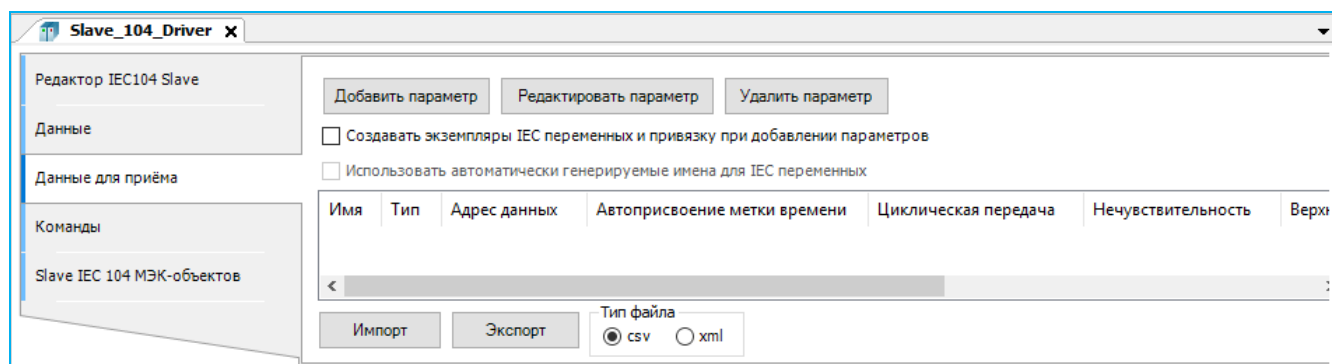


Рисунок 11 – Редактор Slave 104 Driver. Вкладка «Данные для приема»

И там и там задается соотношение между параметрами протокола (адрес, тип) и переменной приложения. Разница только в том, что переменные приложения, указанные на вкладке **Данные для приема**, будут изменять свое значение, метку времени, качество после получения входящих PDU с причиной передачи «Спонтанная», «Циклическая», «Общий опрос».

Адреса информационных объектов «Данных» и «Данные для приема» могут пересекаться. В рамках протокола разница определяется параметром «Адрес ASDU» (иногда его называют «Общий адрес»). Иными словами, существует 2 независимых множества адресов информационных объектов (IO address) для отдаваемых и получаемых данных. Каждое множество имеет свой уникальный общий адрес.

Например:

Общий адрес для отдаваемых данных = 1. Отдаются данные с адресами [1, 2, 3];

Общий адрес для получаемых данных = 2. Получаются данные с адресами [2, 3, 4].

В случае, когда множество получаемых данных не пустое, Slave\_104\_Driver формирует команду общего опроса в ответ на команду START\_DATA после установки входящего соединения. Устройство типа Slave\_104\_Driver не посылает никаких команд за исключением команды общего опроса в вышеописанном случае.

### Создание списка команд, получаемых по IEC-104

Кроме запроса данных с контроллера в IEC-104 реализованы команды, используемые для установки значения какой-либо переменной, либо выполнения по сигналу каких-либо действий.

В редакторе устройства Slave 104 Driver перейдите на внутреннюю вкладку **Команды**. Общий вид редактора команд аналогичен редактору элементов данных. Команды на данной вкладке описываются как параметры. Для добавления, изменения и удаления команд используются соответственно кнопки **Добавить параметр**, **Редактировать параметр**, **Удалить параметр**. Перейти к редактированию команды также можно двойным щелчком левой кнопкой мыши по нужной строке.

Окно ввода/редактирования команд выглядит следующим образом (Рисунок 12).

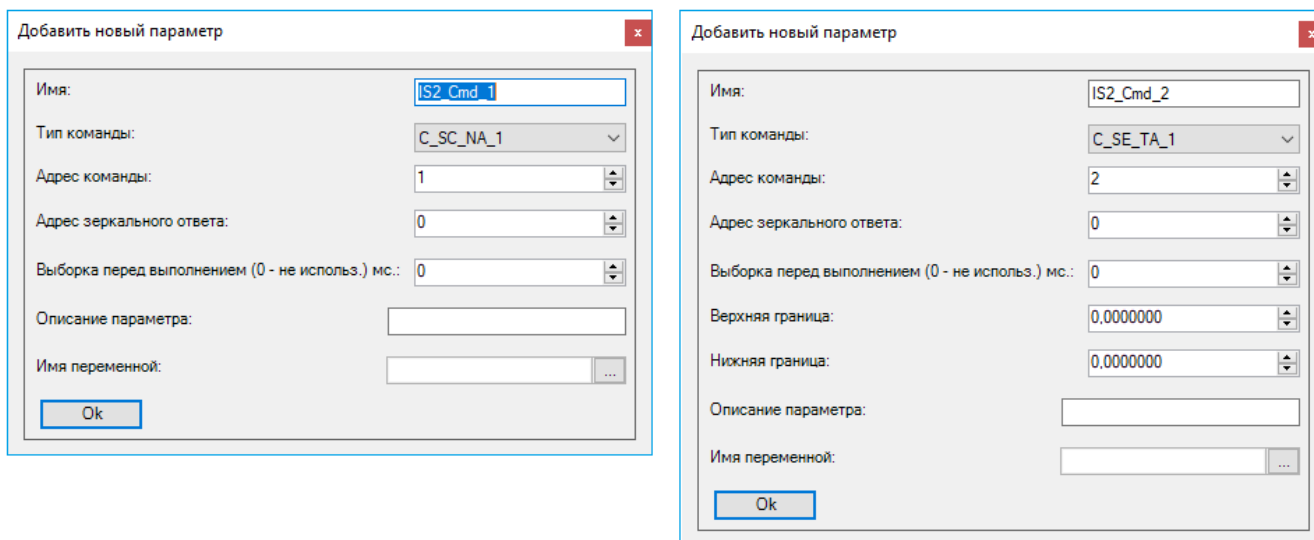


Рисунок 12 – Окно добавления новой команды

Для команд укажите значения в полях. Параметры команд представлены в таблице 6.

Таблица 6 – Параметры команд IEC-104

Параметр	Описание
<b>Имя</b>	Наименование команды
<b>Тип данных</b>	Ассоциированный с этой командой тип IEC-104. Справочные типы данных IEC-104 приведены в таблице 7
<b>Адрес объекта информации</b>	<p>Уникальный адрес команды. Значение адреса не должно превышать максимально допустимого значения, которое обуславливается размером адресного поля PDU.</p> <p>Размер адресного поля PDU определяется параметром <b>Длина поля «адрес»</b>.</p> <p>Например, если задано значение длины поля адреса равное 2, то максимально допустимое значение адреса не может превышать 65535 (для значения 3 - не может превышать 16777215).</p> <p>При некорректном вводе, в логе появится сообщение о переполнении адреса команд, например: «Command addr overflow: 100002 max:65535»</p>
<b>Адрес зеркального ответа</b>	Адрес элемента данных, в который будет помещено значение, переданное командой (0 – не используется)
<b>Время блокировки (0 – не использ.)</b>	Команды могут быть простыми и с «выборкой перед выполнением» (Select Before Operate). При ненулевом значении это время, в течение которого команда остается в состоянии «выбрана» для выполнения (после отправки запроса select). Успешное выполнение команды (execute) возможно до истечения данного таймаута
<b>Масштаб</b>	Для команд, задающих масштабируемые значения
<b>Верхняя граница, Нижняя граница</b>	Пара параметров, определяющих диапазон значений команды для нормализованных команд (C_SE_NA_1 и C_SE_TA_1)



Параметр	Описание
Описание параметра	Опционально, текстовое описание команды
Имя переменной*	Указывается название переменной IEC (программы контроллера), в которой хранятся передаваемые команды. Для заполнения поля <b>Имя переменной</b> нажмите в этом поле кнопку  , открывающую окно <b>Ассистент ввода</b> . Найдите нужную переменную. Если установлен флажок в поле <b>Структурированный вид</b> , то раскрывайте списки с помощью кнопки  . Если флажок снят и переменные представлены одним большим списком, для удобства поиска воспользуйтесь фильтром
<b>Примечание:</b> *- параметр присутствует в окне ввода/редактирования команды, когда привязка переменных происходит по адресу (прямой доступ к памяти ПЛК), без использования механизма I/O Mapping (соотнесение входов/выходов), при условии, что не снят флажок с параметра « <b>Не использовать IO-Mapping</b> »	

Таблица 7 – Перечень типов данных IEC-104

Тип данных	ID типа данных	Описание
C_SC_NA_1	45	Бинарная команда
C_DC_NA_1	46	Команда с тремя состояниями
C_RC_NA_1	47	Регулирующая пошаговая команда
C_SE_NA_1	48	Команда уставки (нормализованное значение)
C_SE_NB_1	49	Команда уставки (масштабируемое значение)
C_SE_NC_1	50	Команда уставки (значение типа float)
C_BO_NA_1	51	Команда установки побитового регистра
C_SC_TA_1	58	Бинарная команда с 56-битной меткой времени
C_DC_TA_1	59	Команда с тремя состояниями с 56-битной меткой времени
C_RC_TA_1	60	Регулирующая пошаговая команда с 56-битной меткой времени
C_SE_TA_1	61	Команда уставки (нормализованное значение) с 56-битной меткой времени
C_SE_TB_1	62	Команда уставки (масштабируемое значение) с 56-битной меткой времени
C_SE_TC_1	63	Команда уставки (значение типа float) с 56-битной меткой времени
C_BO_TA_1	64	Команда установки побитового регистра с 56-битной меткой времени
C_CS_NA_1	103	Команда синхронизации времени

Возможность сохранять принятое время в переменной, не выполняя синхронизацию

Если у объекта Slave\_104\_Driver установлен указатель на интерфейс i\_clock\_handler с помощью вызова Slave\_104\_Driver.m\_outer.set\_clock\_sync\_handler(i\_clock\_handler), то при получении команды C\_CS\_NA\_1 сервер iec-60870-104(101) не будет пытаться сам синхронизировать время, вместо этого будет вызвана пользовательская функция on\_clock\_sync

(timestamp\_type). В этой функции интерфейса i\_clock\_sync\_handler пользователь имеет возможность выполнить действия, продиктованные логикой работы контроллера.

Пример использования:

```
FUNCTION_BLOCK clocksync_handler_fb IMPLEMENTS IEC_LIB.i_clock_sync_handler

METHOD on_clock_sync // пользовательский метод обработки команды C_CS_NA_1
VAR_INPUT
t : IEC_LIB.timestamp_type;
END_VAR
GVL_1.g_ts := t;

PROGRAM PLC_PRG
VAR
first_cycle : BOOL := TRUE;
timesync_handler : clocksync_handler_fb;
END_VAR
IF ( first_cycle )
THEN
first_cycle := FALSE;
Unbalanced_Secondary_101_Driver_1.m_outer_master.set_clock_sync_handler(timesync_h
andler); // для 101-го
Slave_104_Driver.m_outer.set_clock_sync_handler(timesync_handler); // для 104-го
END_IF
```

### Создание переменных для работы с данными/командами, передаваемыми по IEC-104

Элементы данных и команды, описанные в редакторе IEC-104, в терминологии среды разработки являются **каналами ввода-вывода**. Для реализации обмена данными необходимо создать эти каналы (см. предыдущие разделы), создать в программном коде переменные специальных типов, после чего связать переменные с каналами ввода-вывода.

Требуемые типы переменных (функциональные блоки) описаны в библиотеке PsIecCommon. Эта библиотека, а также использующая ее библиотека PsIoDrvIec104Slave автоматически подключаются при добавлении устройства Slave 104 Driver.

Для преобразования временных меток m\_timestamp в systimedate и обратно, в библиотеке PsIecCommon присутствуют – функция systimedate\_to\_timestamp и функциональный блок timestamp\_to\_systimedate.

Для данных используются следующие функциональные блоки: bo\_tb\_fb, ep\_td\_fb, it\_tb\_fb, me\_tf\_fb, sp\_tb\_fb, me\_td\_fb. В наименовании блока фактически указан тип данных IEC-104, например, блоку bo\_tb\_fb соответствует тип M\_BO\_TB\_1, а me\_tf\_fb соответствует M\_ME\_TF\_1. Количество функциональных блоков меньше количества типов данных IEC-104, каждый функциональный блок может использоваться сразу для нескольких типов.

Таблица 8 – Соответствие функциональных блоков и типов данных IEC-104

Функц. блок	Тип данных IEC-104, указываемый для канала ввода-вывода	Комментарий
bo_tb_fb	M_BO_NA_1, M_BO_TA_1, M_BO_TB_1	Для 4-байтных целых без знака (32-битные bitstring)
	M_DP_NA_1, M_DP_TA_1, M_DP_TB_1	Допустимые значения 0, 1, 2
	M_ST_NA_1, M_ST_TA_1, M_ST_TB_1	Допустимые значения 0 ... 255
ep_td_fb	M_EP_TD_1	Для событий защиты оборудования
it_tb_fb	M_IT_NA_1, M_IT_TA_1, M_IT_TB_1	Для 4-байтных целых со знаком
me_tf_fb	M_ME_NC_1, M_ME_TC_1, M_ME_TF_1	Для 4-байтных чисел с плавающей точкой
	M_ME_NA_1, M_ME_TA_1, M_ME_TD_1	Для нормализованных чисел с плавающей точкой, описание канала должно содержать hi/low
	M_ME_NB_1, M_ME_TB_1, M_ME_TE_1	Для масштабируемых величин (scaled values), описание канала должно содержать scale
sp_tb_fb	M_SP_NA_1, M_SP_TA_1, M_SP_TB_1	Для булевых дискретов
me_td_fb	M_ME_NA_1, M_ME_TA_1, M_ME_TD_1	Для нормализованных чисел с плавающей точкой, описание канала должно содержать hi/low

Для создания переменных откройте редактор ПЛК-программы. Например, в редакторе ST для программы PLC\_PRG создание переменных выглядит следующим образом:

```
PROGRAM PLC_PRG
VAR
IS1_Data_1:me_tf_fb;
IS1_Data_2:me_tf_fb;
IS1_Data_3:sp_tb_fb;
IS1_Data_4:sp_tb_fb;
END_VAR
```

Ниже представлен пример объявления переменных:

```
active_connections : STRING;           //список соединений
linked : BOOL;                          //статус соединения

iec1_real          : PsIoDrvIec104Slave.me_tf_fb; //float с меткой времени
iec3_di            : PsIoDrvIec104Slave.sp_tb_fb; //дискрет с меткой времени
iec5_int           : PsIoDrvIec104Slave.it_tb_fb; //целочисленное с меткой времени

ts                 : PsIoDrvIec104Slave.timestamp_type;
//переменная типа «временная отметка для IEC-104»
cmd1               : PsIoDrvIec104Slave.common_command_type; //команда;
```

Предусмотрена возможность программно получить список подключенных к slave мастеров (соединений):

```
active_connections := Slave_104_Driver.m_active_connections;
```

Переменная *active\_connections* показывает список IP-адресов в виде строки, где адреса разделены пробелами: '172.29.23.152 172.29.23.10'.

Также предусмотрена возможность получать статус соединения (true, если соединение установлено, иначе – false):

```
linked := Slave_104_Driver.Linked();
```

Свойства (properties) функциональных блоков для передачи данных следующие:

- **value** – данному свойству присваивается значение, которое будет передаваться по IEC-104. Используемый тип значения (имеется в виду простой тип – int, real, bool...) будет зависеть от того, какой тип (по IEC-104) имеет канал ввода-вывода, связанный с данной переменной. Свойство *value* принимает значение длиной не более 4 байт;
- **quality** – байт качества значения. Значение по умолчанию – 0 (качество *good*). Допустимы также 128 (80 hex) – качество *invalid*, 64 (40 hex) – качество *substituted*. Все допустимые значения указаны в описании протокола IEC-60870-5-104(101);
- **timestamp** – временная отметка, назначаемая при изменении состояния. Представлена типом данных *timestamp\_type*. Может генерироваться автоматически, если указано в описании связанного канала ввода-вывода, либо задаваться вручную. В случае автоматической генерации *timestamp* назначается при изменении значения *value*.

Ниже приведен пример генерации значения *timestamp\_type* из текущей даты/времени. Реализовано в виде метода для программы, требуется подключение библиотеки SysTimeRtc.

#### Объявление

```
METHOD PUBLIC get_current_timestamp : PsIoDrvIec104Slave.timestamp_type
VAR_INPUT
END_VAR
VAR
```

```
    stTime          : SYSTIME;
    sdtDateHiRes    : SYSTIMEDATE;
```

```
END_VAR
```

#### Реализация

```
IF (SysTimeRtcHighResGet(stTime)=0) AND
(systimertc.SysTimeRtcConvertHighResToDate(stTime, sdtDateHiRes)=0)
THEN
get_current_timestamp.m_year := sdtDateHiRes.wYear;
get_current_timestamp.m_month := sdtDateHiRes.wMonth;
get_current_timestamp.m_day := sdtDateHiRes.wDay;
get_current_timestamp.m_hour := sdtDateHiRes.wHour;
get_current_timestamp.m_minute := sdtDateHiRes.wMinute;
get_current_timestamp.m_second := sdtDateHiRes.wSecond;

get_current_timestamp.m_millisecond := sdtDateHiRes.wMilliseconds;
```

END\_IF

Для команд предусмотрен один тип функционального блока: **common\_command\_type**. Его основные компоненты следующие:

- **m\_value** – передается содержание всего поля Information Object PDU команды. Имеет тип **common\_union** т.е. объединение размером 4 байта. Данные 4 байта могут быть интерпретированы как BYTE, WORD, DWORD, INT, DINT, REAL в зависимости от типа команды;
- **m\_timestamp** – в случае, когда команда имеет тип, включающий в себя метку времени, данное поле содержит метку времени, полученную в составе команды. Для команд без метки времени данное поле не используется.

Ниже приведен пример использования переменных в программном коде.

```
active_connections := PsIoDrvIec104Slave.m_active_connections;
//список IP адресов активных соединений
iec1_real.value := 3.14; //задаем действительное значение
iec3_di.value:= 1; //задаем дискретное значение
iec3_di.timestamp := get_current_timestamp();
//дополнительно генерируем timestamp
iec5_int.value:=515; //целочисленное значение
cmd_value:= cmd1.m_value; //пришедшая команда
```

### Привязка переменных программы к элементам данных и командам IEC-104

Начиная с версии СПО 1.5.7.0, чтение/запись привязанных переменных происходит по адресу (прямой доступ к памяти ПЛК), без использования механизма I/O Mapping (соотнесение входов/выходов), при условии, что не снят флажок с параметра «Не использовать IO-Mapping» (параметр активирован).

Если IEC устройство содержит параметры старого формата, для которых применяется механизм I/O Mapping (соотнесение входов/выходов), то их можно обновить до нового формата с использованием прямой адресации переменных с помощью кнопки *Обновить параметры* (Рисунок 13). Кнопка появляется на вкладке **Данные** или **Команды** только в том случае, если активирован параметр «Не использовать IO-Mapping» и обнаружены параметры старого формата. Такие параметры расположены в другом диапазоне ID.

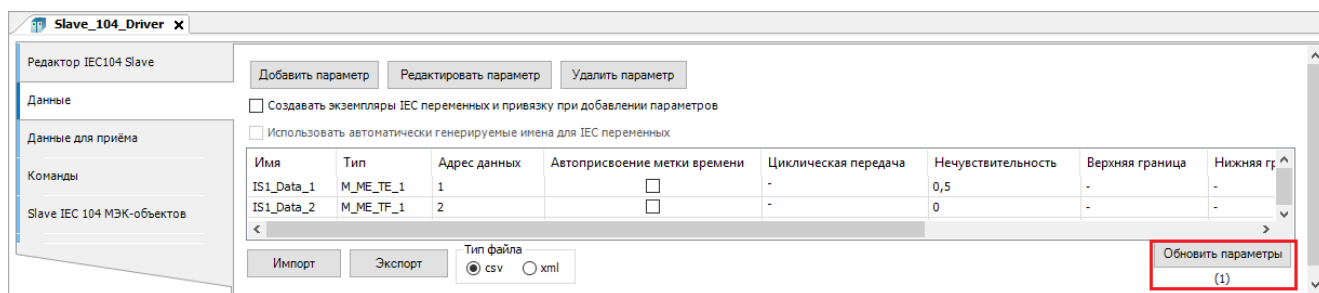


Рисунок 13 - Обновление параметров до нового формата

При активированном параметре «**Не использовать IO-Mapping**» в редакторе устройства **Slave 104 Driver** будет отсутствовать внутренняя вкладка **Соотнесение входов/выходов**.



**ВНИМАНИЕ!**

Рекомендуется не деактивировать параметр «**Не использовать IO-Mapping**»

Если деактивировать параметр (снять флажок), необходимо будет закрыть основную вкладку, а затем заново открыть и при добавлении данных/команд автоматически появится вкладка **Соотнесение входов/выходов** (Рисунок 14).

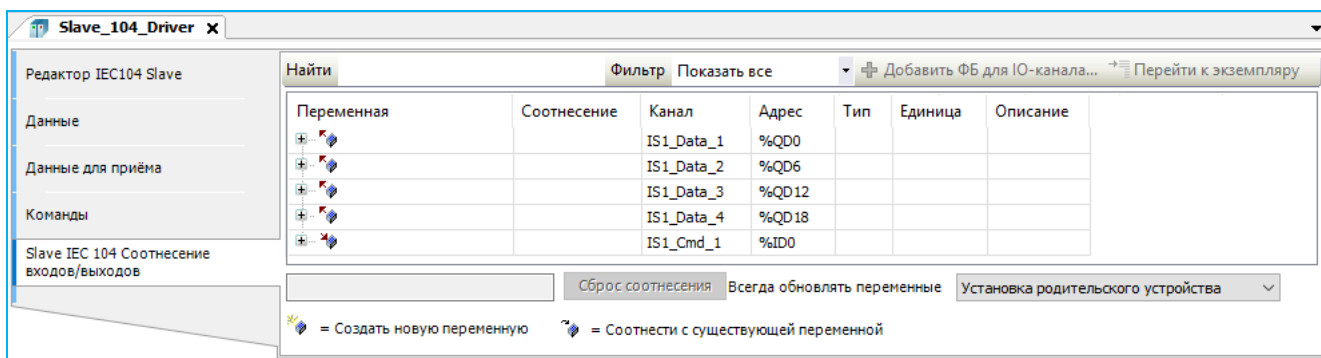


Рисунок 14 – Slave 104 Driver Соотнесение входов/выходов

На этой вкладке представлен список каналов ввода-вывода (в терминологии среды разработки), ассоциированных с устройством. Когда пользователь создает элемент данных или команду в соответствующем редакторе (вкладка **Данные**, вкладка **Команды**), этот параметр автоматически появляется здесь в виде канала вывода (для данных) или канала ввода (для команд). Наименование добавленного параметра указано в колонке **Канал**.

Для того, чтобы значения, приходящие/отправляемые по каналу, были доступны в программе контроллера, нужно привязать канал к переменной программы. Дважды щелкните левой кнопкой мыши в строке нужного канала. Появится курсор (можно вручную ввести имя переменной, семантика имен описана ниже) и кнопка **...**, открывающая окно **Ассистент ввода** (Рисунок 15).

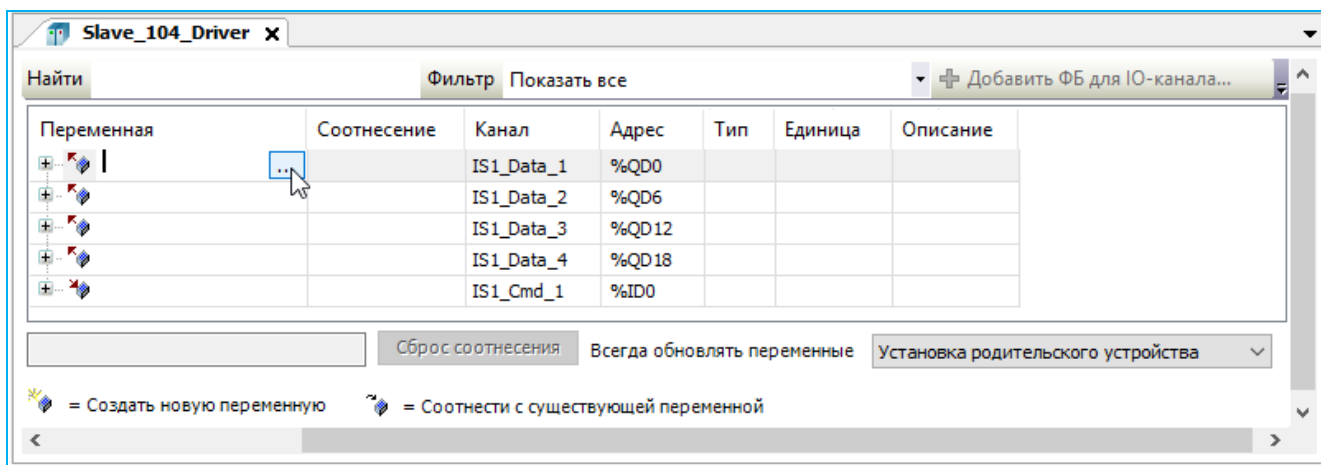



Рисунок 15 – Ручной ввод переменной или вызов ассистента ввода



В окне **Ассистент ввода** (Рисунок 16) найдите нужную переменную. Если установлен флажок в поле **Структурированный вид**, то раскрывайте списки с помощью кнопки . Если флажок снят и переменные представлены одним большим списком, для удобства поиска воспользуйтесь фильтром.

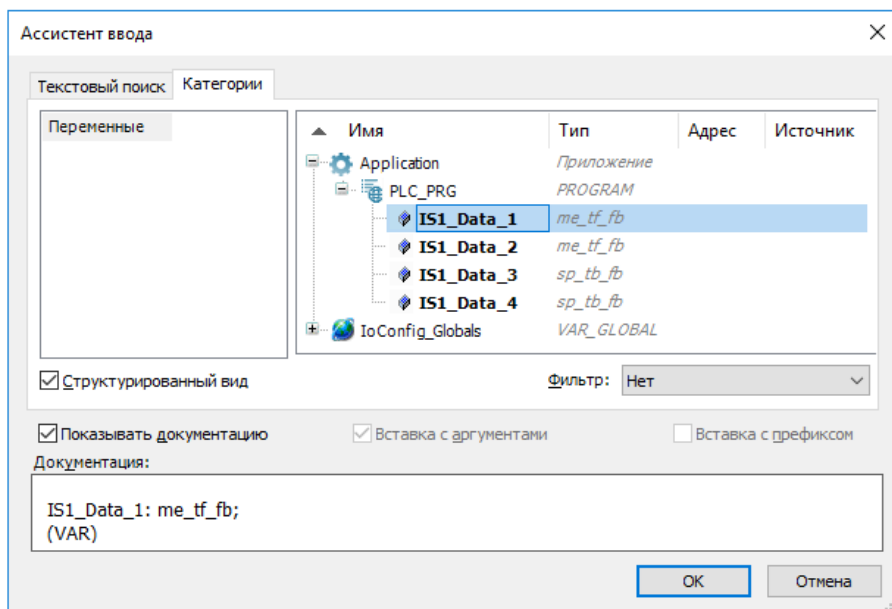


Рисунок 16 – Диалоговое окно «Ассистент ввода»

После выбора переменной нажмите кнопку **OK**, закроется окно **Ассистент ввода**, а переменная появится на вкладке **Соотнесение входов/выходов** (Рисунок 17).

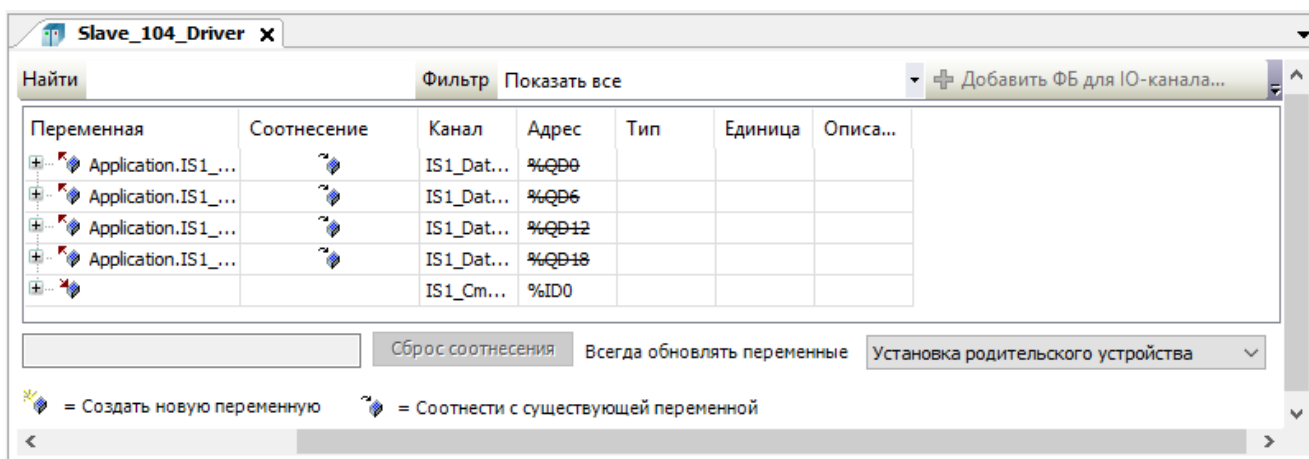


Рисунок 17 – Переменные привязаны к каналам вывода



### ИНФОРМАЦИЯ

Допускается использование одних и тех же экземпляров переменных (и данных, и команд) для привязки к разным экземплярам драйверов (Master/Slave, 101/104), при условии, что настройки привязанной переменной (параметры элементов данных/команд) для всех экземпляров драйверов идентичны.

На этапе компиляции проекта выполняется проверка совпадения настроек, в случае несовпадения, в окно сообщений выводится сообщение об ошибке, с указанием имени переменной и имен соответствующих экземпляров драйверов.

Запрещено привязывать переменную данных одновременно во вкладке **Данные** и **Данные для приема**

При ручном вводе семантика имен следующая (угловые скобки при вводе НЕ используются):  
*<Имя приложения>. <Имя программного юнита>. <Имя переменной>*

Как ранее указано, для привязки к каналу IEC-104 используются только переменные, представляющие из себя функциональные блоки определенных типов, описанных в библиотеке PsIecCommon.



### ИНФОРМАЦИЯ

Тип данных IEC-104 (например, M\_BO\_NA\_1) задается при описании элемента данных или команды. После «привязки» переменной приложения к соответствующему каналу, переменная (типа bo\_tb\_fb) будет передаваться по протоколу IEC-60870-5-104(101) с использованием типа M\_BO\_NA\_1

Все функциональные блоки, используемые для описания переменных IEC-104, имеют поле метки времени. Но в случаях, когда тип данных IEC-104, указанный в описании элемента данных или команды, не имеет метки времени (например, M\_BO\_NA\_1), поле **Метка времени** функционального блока просто не используется. Также в зависимости от типа данных IEC-104 поле **Метка времени** может сериализоваться в 56-битный или в 24-битный timestamp.

### Автоматическая генерация переменных и привязка к каналам ввода-вывода

При добавлении параметров на вкладках **Данные** и **Команды** можно активировать режим, при котором для каждого канала автоматически создаются переменные ПЛК-программы и формируется привязка переменной к каналу. Для этого при создании нового элемента данных или команды установите флажок в поле **Создавать экземпляры IEC переменных и привязку при добавлении параметров** (Рисунок 18).

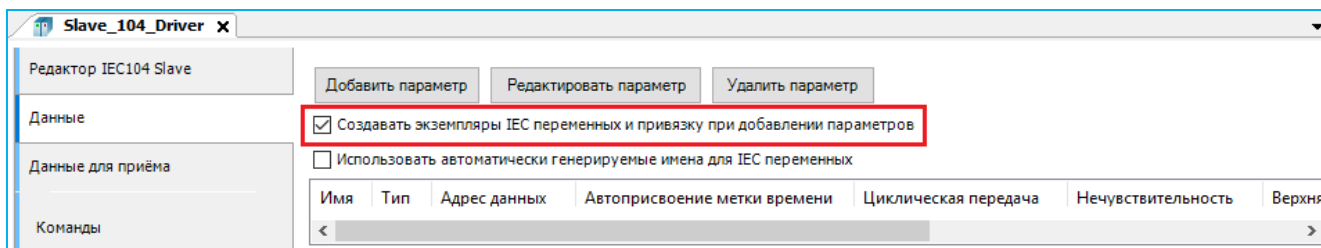


Рисунок 18 – Установка флажка в поле «Создавать экземпляры IEC переменных и привязку при добавлении параметров»

В этом режиме при добавлении канала для устройства Slave 104 Driver создается собственный список глобальных переменных с именем вида I104\_GVL\_X, где X – условный порядковый номер устройства, начиная с единицы (Рисунок 19). В этом списке будет создана переменная соответствующего типа.

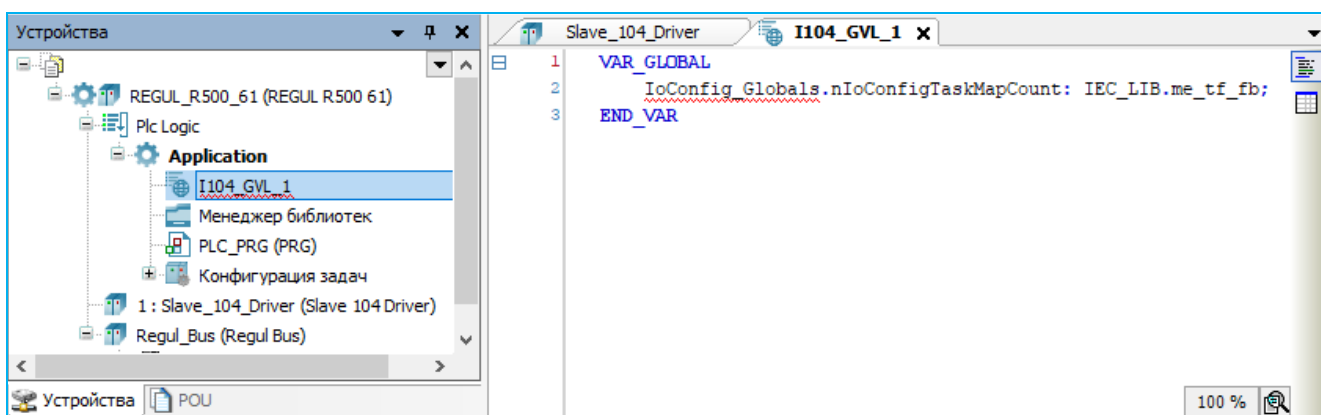


Рисунок 19 – Пример описания переменных в списке I104\_GVL\_1

При наличии вкладки **Соотнесение входов/выходов** новая переменная будет автоматически привязана к новому каналу (Рисунок 20). Наименование переменной будет совпадать с именем канала (пробелы заменяются на подчеркивания).

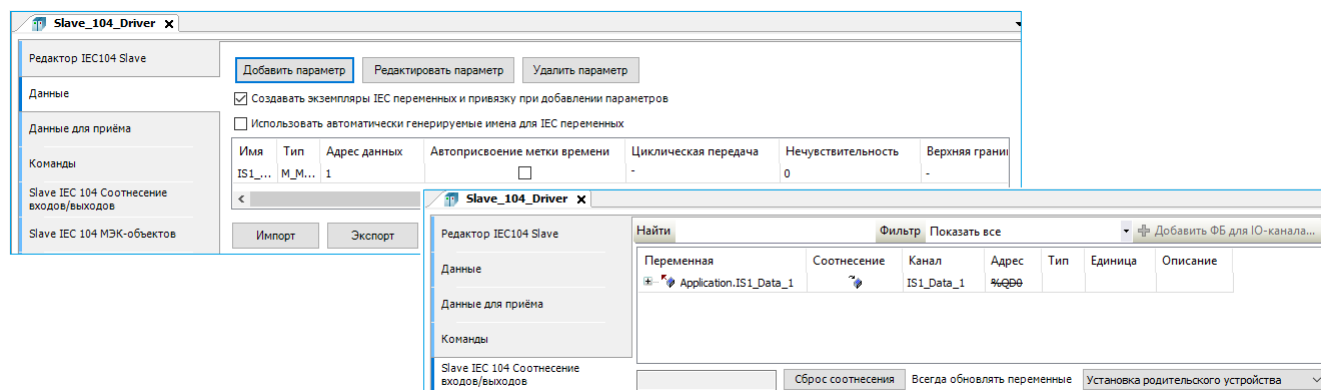


Рисунок 20 – Автоматическое создание переменной и автоматическая привязка ее к каналу

При установленном флажке в поле **Использовать автоматически генерируемые имена для IEC переменных** имена переменных будут создаваться на основе шаблона: *iec<случайное число>\_data\_fb*. При отсутствии флажка в этом поле имена генерируемых переменных имеют вид: *IS<X>\_Data\_<N>* (для элементов данных), *IS<X>\_Cmd\_<N>* (для команд), где X – номер

Slave-устройства, N – нумератор переменных. Генерируемые по умолчанию имена каналов данных и команд совпадают с именами связываемых переменных.



**ВНИМАНИЕ!**

Использование вызова системной функции спорадической отправки из прикладной программы не является стандартизированной опцией протокола. Настоятельно рекомендуем применять только в тех системах, где это является прямым требованием проекта

Спонтанные уведомления инициируются в момент, когда объекту данных присваивается значение, отличное от предыдущего.

Например:

```
sp_na : IEC_LIB.sp_tb_fb;  
sp_na.value := NOT sp_na.value;
```

Так же можно инициировать отсылку спонтанного уведомления, если значение объекта данных не изменилось. Осуществить это можно двумя способами:

- 1) `Slave_104_Driver.send_spontaneous(sp_na)` – только один драйвер отправляет эти данные;
- 2) `sp_na.send_spontaneous ()` – все драйверы, которые в конфигурации имеют `sp_na`, отправят `sp_na`.

Функция спорадической передачи `send_spontaneous()` срабатывает всегда, независимо от установленного значения параметра полосы нечувствительности (`deadband`). Данные будут отправлены с причиной передачи «спонтанная», даже если значение не вышло за границы полосы нечувствительности. В остальных случаях (когда переменная меняет свое значение), если установлен `deadband` - он применяется всегда.

## Настройка контроллера в качестве Slave 104 OS

Добавление устройства и настройка ip-адресов производится идентично тому, как описано в разделе «Настройка контроллера в качестве Slave 104».

### Настройка общих параметров устройства Slave 104 OS

В редакторе устройства **Slave 104 OS** (Рисунок 21) в блоке **Общие параметры устройства** доступны для настройки параметры, представленные в таблице 9.

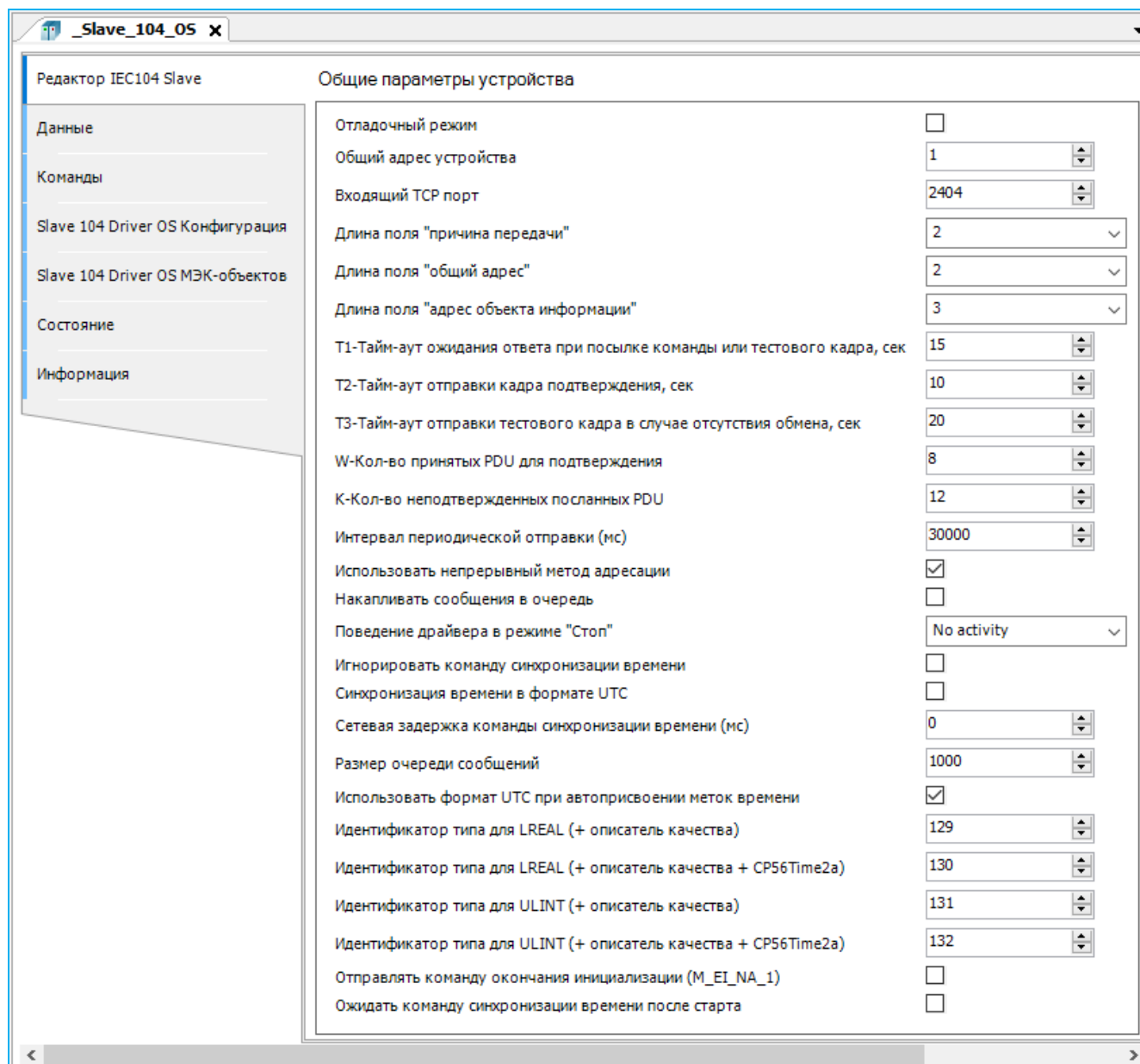


Рисунок 21 – Редактор Slave 104 OS

Таблица 9 – Общие параметры устройства

Параметр	Описание	Значение по умолчанию
<b>Отладочный режим</b>	При установке флажка <input checked="" type="checkbox"/> в этом поле в журнал работы контроллера будут выводиться сообщения об ошибках, предупреждениях и ключевых моментах работы драйвера.	
<b>Общий адрес устройства</b>	Содержит общий адрес (COMMON ADDRESS, ASDU ADDRESS) устройства, все отдаваемые данные и выполняемые команды имеют общую часть, равную значению этого параметра (см. спецификацию IEC-104)	1
<b>Входящий TCP порт</b>	Номер TCP порта для приема входящих соединений от мастера IEC-104.	2404
<b>Длина поля «причина передачи»</b>	Содержит длину поля «причина передачи» (COT, Cause Of Transmission). Допустимые значения параметра: 1 или 2	2
<b>Длина поля «общий адрес»</b>	Содержит длину поля «общий адрес» (CA, Common Address). Допустимые значения параметра: 1 или 2.	2
<b>Длина поля «адрес объекта информации»</b>	Содержит длину поля «адрес объекта информации» (IOA, Information Object Address). Допустимые значения параметра: 2 или 3	3
<b>T1-Тайм-аут ожидания ответа при посылке команды или тестового кадра, сек</b>	Время, в течение которого ожидается ответ на команду или подтверждение доставки PDU. Превышение таймаута приводит к разрыву соединения	15
<b>T2-Тайм-аут отправки кадра подтверждения, сек</b>	При получении PDU запускается таймер на T2 секунд, подтверждение о приеме (пакет S-PDU) отсылается либо по истечению этого таймера, либо по факту приема некоторого предельного количества PDU, указанного в параметре W. T2 должен быть меньше T1	10
<b>T3-Тайм-аут отправки тестового кадра в случае отсутствия обмена, сек</b>	Периодически как master, так и slave могут отсылать специальные пакеты для проверки связи.	20
<b>W-Количество принятых PDU для подтверждения</b>	Количество принятых PDU, которое необходимо подтвердить. Рекомендуем выбирать значение W, не превышающее 2/3 от значения K (Master). Параметры должны быть согласованы на Master и Slave. Например, при K=10 на Master, W=8 на Slave	8
<b>K-Количество неподтвержденных посланных PDU</b>	Количество неподтвержденных отправленных PDU, после которого драйвер прекращает отправлять новые I-пакеты, дожидаясь подтверждения отправленных ранее.  Следует отметить, что в случае, когда требуется отправка большого количества данных, значение параметра K следует устанавливать большим. Например, если требуется отправка	12

Параметр	Описание	Значение по умолчанию
	20000 отсчетов в секунду, значение К нужно устанавливать от 1000 и больше. Иначе драйвер будет приостанавливать отправку данных, ожидая подтверждение клиента, размеры очередей при этом будут нарастать.	
<b>Интервал периодической отправки (мс)</b>	При описании элемента данных IEC-104 можно включить его в периодическую рассылку (установка флажка в поле <b>Участвует в циклической рассылке</b> ). Для всех таких элементов их значения, даже при отсутствии изменений, будут отсылааться с указанным здесь периодом (в миллисекундах)	30000
<b>Использовать непрерывный метод адресации</b>	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что активирован «экономный» режим IEC-104 - в отсылаемых пакетах указывается адрес только первого элемента	
<b>Накапливать сообщения в очередь</b>	При установке флажка <input checked="" type="checkbox"/> в этом поле будет производиться накопление данных при отсутствии связи с мастером. Глубина накопления задается параметром <b>Размер очереди сообщений</b> (см. таблицу 10)	
<b>Поведение драйвера в режиме «Стоп» *</b>	Определяет поведение компонента при остановке программы (см. таблицу 10). <i>Нет активности (No activity)</i> - означает, что устройство Slave закрывает все соединения и не принимает новые <i>Нормальная работа (Normal work)</i> - означает продолжение работы в обычном режим	No activity
<b>Игнорировать команду синхронизации времени</b>	При установке флажка <input checked="" type="checkbox"/> в этом поле будет запрещена обработка команды синхронизации времени (C_CS_NA_1)	
<b>Синхронизация времени в формате UTC</b>	Если будет установлен флажок <input checked="" type="checkbox"/> в этом поле, то, при синхронизации временная метка в составе команды будет интерпретирована, как UTC. Если флажок снят, то, это время будет интерпретировано, как локальное время	
<b>Сетевая задержка команды синхронизации времени (мс)</b>	Корректировка команды синхронизации времени – добавляется к временной метке команды	0
<b>Размер очереди сообщений</b>	Определяет размер буфера сообщений для пересылки в нормальном режиме, а также при отсутствии связи. Следует избегать установки размера очереди сообщений более 50000 сообщений, поскольку в этом случае возникает риск исчерпания резерва оперативной памяти. Вместе с тем, размер очереди должен быть не меньше планируемого количества элементов данных. Это связано с тем, что при внешнем подключении IEC-104 Slave может получить команду общего опроса, что приведет к единовременному заполнению очереди последними	1000

Параметр	Описание	Значение по умолчанию
	значениями всех элементов данных. Если размер очереди меньше рекомендуемого, то часть данных не будет отправлена – более старые данные будут перезаписаны	
<b>Использовать формат UTC при автоприсвоении меток времени</b>	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что значение метки времени при отправке временных типов данных будет выставлено в формате UTC, иначе - локального времени	
<b>Идентификатор типа для LREAL (+описатель качества)</b>	Задание ID для пользовательского типа LREAL(double), без метки времени	129
<b>Идентификатор типа для LREAL (+описатель качества +CP56Time2a)</b>	Задание ID для пользовательского типа LREAL(double), с меткой времени	130
<b>Идентификатор типа для ULINT (+описатель качества)</b>	Задание ID для пользовательского типа ULINT(uint64), без метки времени	131
<b>Идентификатор типа для ULINT (+описатель качества +CP56Time2a)</b>	Задание ID для пользовательского типа ULINT(uint64), с меткой времени	132
<b>Отправлять команду окончания инициализации (M_EI_NA_1)</b>	При установке флажка <input checked="" type="checkbox"/> в этом поле, после загрузки и инициализации, slave отправляет указанную команду по активному соединению для каждой резервной пары подключений	
<b>Ожидать команду синхронизации времени после старта</b>	Если будет установлен флажок <input checked="" type="checkbox"/> в этом поле, то, до получения команды синхронизации все данные с меткой времени содержат установленный в этой метке бит недействительности (IV)	
<p><b>Примечания</b></p> <p>*- предусмотрена возможность самостоятельно активировать «поведение в режиме «Стоп» в программном коде. Для активации режима требуется в программе присвоить значение TRUE свойству ActivateStopBehaviour необходимого устройства:</p> <pre>«IEC104SlaveOS_device_name».ActivateStopBehaviour := TRUE;</pre> <p>После этого slave-устройство перейдет в STOP-режим работы</p>		

### Поведение в режиме СТОП и буферизация данных

Настройки параметров «Поведение драйвера в режиме «Стоп»» и «Накапливать сообщения в очередь» для устройства Slave определяют логику передачи и накопления данных в зависимости от состояния МЭК-приложения (старт/стоп) или вызова в коде приложения



свойства `ActivateStopBehaviour` (программная эмуляция старт/стоп) для конкретного драйвера. Логика включения накопления очереди сообщений представлена в таблице 10.

Таблица 10 – Логика включения накопления очереди сообщений

Значение параметра		Активация режима Стоп (в ПЛК, в IDE) или в коде через вызов <code>ActivateStopBehaviour</code>	Наличие подключения мастера	Итог
Поведение драйвера в режиме «Стоп»	Накапливать сообщения в очередь			
No activity	Не важно	Включена	Не важно	Драйвер не работает, данные не копятя и не отправляются
No activity	Включена	Выключена	Есть	Данные отправляются мастеру
No activity	Включена	Выключена	Нет	Данные копятя в очереди
No activity	Выключена	Выключена	Есть	Данные отправляются мастеру
No activity	Выключена	Выключена	Нет	Данные <b>НЕ</b> копятя в очереди
Normal work	Включена	Не важно	Есть	Данные отправляются мастеру
Normal work	Включена	Не важно	Нет	Данные копятя в очереди
Normal work	Выключена	Не важно	Есть	Данные отправляются мастеру
Normal work	Выключена	Не важно	Нет	Данные <b>НЕ</b> копятя в очереди

Для принудительной очистки очереди спонтанных данных драйвера Slave 104 OS пользователю предоставляется метод `purge_queue()`, который необходимо вызвать для экземпляра драйвера:

```
Slave_104_OS.purge_queue();
```

Это приведет к очистке текущих очередей спонтанных данных для всех подключений (резервированных пар подключений).

## Журналирование работы драйвера

Для просмотра журнала работы драйвера перейдите на основную страницу параметров устройства во вкладку **Журнал**, куда могут выводиться следующие сообщения драйвера Slave 104 OS:

- подключение/отключение клиента (мастера) по основному/резервному IP-адресам, включая команды активации (STARTDT\_ACT) и деактивации (STOPDT\_ACT) канала (Рисунок 22);

Жёсткость	Временная отметка	Описание	Компонент
!	24.03.2022 16:02:10.564	Slave104[s1]: connection [reserve] 192.168.34.111:45253 is closed	PsIEC60870Slave104_OS
!	24.03.2022 16:02:09.952	Slave104[s1]: connection [reserve] 192.168.34.111:45253 is deactivated	PsIEC60870Slave104_OS
!	24.03.2022 16:01:37.696	Slave104[s1]: connection [reserve] 192.168.34.111:45253 is activated	PsIEC60870Slave104_OS
!	24.03.2022 16:01:21.076	Slave104[s1]: connection [reserve] 192.168.34.111:45253 is opened	PsIEC60870Slave104_OS
!	24.03.2022 15:54:48.248	Slave104[s1]: connection [base] 172.29.34.6:44923 is closed	PsIEC60870Slave104_OS
!	24.03.2022 15:54:47.468	Slave104[s1]: connection [base] 172.29.34.6:44923 is deactivated	PsIEC60870Slave104_OS
!	24.03.2022 15:54:43.172	Slave104[s1]: connection [base] 172.29.34.6:44923 is activated	PsIEC60870Slave104_OS
!	24.03.2022 15:54:42.248	Slave104[s1]: connection [base] 172.29.34.6:44923 is opened	PsIEC60870Slave104_OS
!	24.03.2022 15:54:37.593	Slave104[s1]: started listening on port 2404	PsIEC60870Slave104_OS

Рисунок 22 – Сообщения о работе драйвера Slave 104 OS

- обработка запуска и остановки приложения через переключатель Run/Stop на модуле ЦП, либо через меню среды разработки «Отладка⇒Старт/Стоп» или с помощью свойства «ActivateStopBehaviour» экземпляра драйвера в коде приложения (Рисунок 23);

Жёсткость	Временная отметка	Описание	Компонент
!	24.03.2022 16:13:15.510	Slave104[s1]: started listening on port 2404	PsIEC60870Slave104_OS
!	24.03.2022 16:13:15.510	Slave104[s1]: STOP mode is OFF	PsIEC60870Slave104_OS
!	24.03.2022 16:13:15.487	s1 : BusCycle : APP is STARTED	PsIEC60870Slave104_OS
!	24.03.2022 16:13:15.485	Start application done	PsLed
!	24.03.2022 16:13:15.485	Switch RUN/STOP changed to RUN	PsLed
!	24.03.2022 16:12:29.029	Slave104[s1]: stopped listening on port 2404	PsIEC60870Slave104_OS
!	24.03.2022 16:12:29.005	Slave104[s1]: STOP mode is ON, STOPBEHAVIOUR = NO_ACTIVITY	PsIEC60870Slave104_OS
!	24.03.2022 16:12:28.992	Stop application done(RUNSTOP_SWITCH)	PsLed
!	24.03.2022 16:12:28.980	s1 : BusCycle : APP is STOPED	PsIEC60870Slave104_OS
!	24.03.2022 16:12:28.979	Switch RUN/STOP changed to STOP	PsLed

Рисунок 23 – Сообщения о запуске и остановке приложения

- переполнение очереди изменений данных (одна очередь для всех соединений) при генерации в приложении большого объема данных (Рисунок 24);

Жёсткость	Временная отметка	Описание	Компонент
!	24.03.2022 17:02:27.320	Slave_104_OS : on_data_change(): 140000 spont values are discarded, internal cache is FULL!	PsIEC60870Slave104_OS
!	24.03.2022 17:02:26.816	Slave_104_OS : on_data_change(): 130000 spont values are discarded, internal cache is FULL!	PsIEC60870Slave104_OS
!	24.03.2022 17:02:26.311	Slave_104_OS : on_data_change(): 120000 spont values are discarded, internal cache is FULL!	PsIEC60870Slave104_OS
!	24.03.2022 17:02:25.807	Slave_104_OS : on_data_change(): 110000 spont values are discarded, internal cache is FULL!	PsIEC60870Slave104_OS
!	24.03.2022 17:02:25.303	Slave_104_OS : on_data_change(): 100000 spont values are discarded, internal cache is FULL!	PsIEC60870Slave104_OS

Рисунок 24 – Сообщения о переполнении очереди изменений данных

- переполнение очереди спонтанных данных, глубина которой, определяется параметром «Размер очереди сообщений» (на уровне системного драйвера), при этом, более старые данные в очереди перезаписываются новыми (Рисунок 25);

Жёсткость	Временная отметка	Описание	Компонент
⚠	24.03.2022 17:28:38.095	Slave104[Slave_104_OS]: 6000 oldest values are discarded, internal cache is FULL!	PsIEC60870Slave104_OS
⚠	24.03.2022 17:28:38.045	Slave104[Slave_104_OS]: 5000 oldest values are discarded, internal cache is FULL!	PsIEC60870Slave104_OS
⚠	24.03.2022 17:28:38.032	Slave104[Slave_104_OS]: 4000 oldest values are discarded, internal cache is FULL!	PsIEC60870Slave104_OS
⚠	24.03.2022 17:28:27.949	Slave104[Slave_104_OS]: 3000 oldest values are discarded, internal cache is FULL!	PsIEC60870Slave104_OS
⚠	24.03.2022 17:28:27.937	Slave104[Slave_104_OS]: 2000 oldest values are discarded, internal cache is FULL!	PsIEC60870Slave104_OS
⚠	24.03.2022 17:28:17.864	Slave104[Slave_104_OS]: 1000 oldest values are discarded, internal cache is FULL!	PsIEC60870Slave104_OS
ℹ	24.03.2022 17:27:58.052	Slave104[Slave_104_OS]: started listening on port 2404	PsIEC60870Slave104_OS

Рисунок 25 - Сообщения о переполнении очереди спонтанных данных

– получение команды синхронизации времени от мастера (Рисунок 26).

Жёсткость	Временная отметка	Описание	Компонент
ℹ	13.04.2022 09:59:57.523	Slave104[s1]: time sync: UTC '2022-04-13 04:59:57.507' is set OK, local now is '2022-04-13 09:59:57.507000'	PsIEC60870Slave104_OS
ℹ	13.04.2022 09:59:57.523	Slave104[s1]: time sync: 2022-04-13 04:59:57.507(UTC) format received	PsIEC60870Slave104_OS
ℹ	13.04.2022 09:59:57.523	Slave104[s1]: connection [base]172.29.34.6:39044 is activated	PsIEC60870Slave104_OS
ℹ	13.04.2022 09:59:57.523	Slave104[s1]: connection [base]172.29.34.6:39044 is opened	PsIEC60870Slave104_OS
ℹ	13.04.2022 09:59:29.590	Slave104[s1]: started listening on port 2404	PsIEC60870Slave104_OS

Рисунок 26 – Сообщения о получении команды синхронизации времени

Дополнительную информацию о работе драйвера можно получить, просмотрев лог-файлы. Для загрузки файлов, перейдите на страницу параметров устройства во вкладку **Файлы**. В области **Исполнение** нажмите кнопку (**Обновить**). В окне отобразится дерево файлов, имеющих на контроллере. Перейдите в папку **logs**→**logger**→**user** (Рисунок 27). Имена лог-файлов имеют вид: «*iec60870driver\_\*.log*». Скопируйте лог-файлы с контроллера для просмотра на ПК (кнопкой , из **Исполнение** в **Хост**).

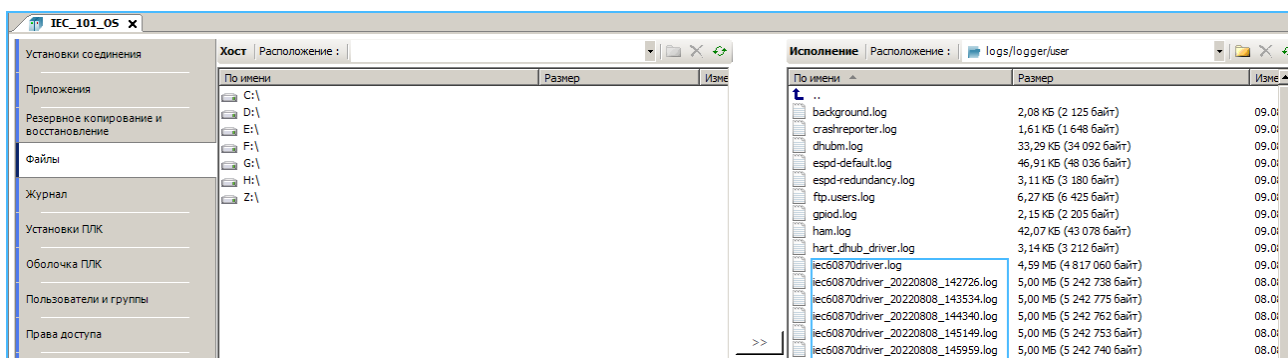


Рисунок 27 - Константы

Можно задать детальность трассировки в файлах с помощью свойства **LogLevel** экземпляра драйвера в процессе работы приложения:

```
Slave104.LogLevel:= PsIEC60870Bridge.LOG_LEVEL_PDU_MAX;
```

Константы, включающие определенный уровень логирования, описаны в библиотеке **PsIEC60870Bridge** (Рисунок 28). Константы можно объединить с помощью двоичного оператора ИЛИ (OR). Уровень, соответствующий **LOG\_LEVEL\_BASE**, включен всегда.

Имя	Тип	Наследовано от	Адрес	Начальн.	Комментарий
LOG_LEVEL_BASE	DWORD			16#0	base log level
LOG_LEVEL_PDU_SHORT	DWORD			16#1	short PDU trace log level
LOG_LEVEL_PDU_MAX	DWORD			16#2	full PDU trace log level
LOG_LEVEL_SPONT_DATA	DWORD			16#4	spontaneous data event log level
LOG_LEVEL_SPONT_VALUE	DWORD			16#8	spontaneous data values log level
LOG_LEVEL_CMD_DATA	DWORD			16#10	command data event log level
LOG_LEVEL_CMD_VALUE	DWORD			16#20	command data values log level
LOG_LEVEL_IG_DATA	DWORD			16#40	interrogation data event log level
LOG_LEVEL_IG_VALUE	DWORD			16#80	interrogation data values log level
LOG_LEVEL_PERIODIC_DATA	DWORD			16#100	periodic data event log level
LOG_LEVEL_PERIODIC_VALUE	DWORD			16#200	periodic data values log level
LOG_LEVEL_READ_DATA	DWORD			16#400	read request data event log level
LOG_LEVEL_READ_VALUE	DWORD			16#800	read request data values log level
LOG_LEVEL_ALL	DWORD			16#FFF	all log levels

Рисунок 28 – Константы

Предусмотрена возможность программно получить список подключенных к **Slave 104 OS** мастеров (активных соединений) с помощью свойства ФБ драйвера *ActiveConnections*:

```
active_connections      : STRING(255);
active_connections := Slave_104_OS_Driver.ActiveConnections;
```

Переменная *active\_connections* при этом покажет список IP-адресов в виде строки, где адреса мастеров (с локальными портами ПЛК, на которых драйвер принимает входящие подключения) разделены пробелами: '172.29.23.152:2404 192.168.34.110:2405'.

### Создание списка элементов данных, передаваемых по IEC-104

Для добавления элементов данных в редакторе устройства **Драйвер Slave 104 OS** перейдите на внутреннюю вкладку **Данные** (Рисунок 29).

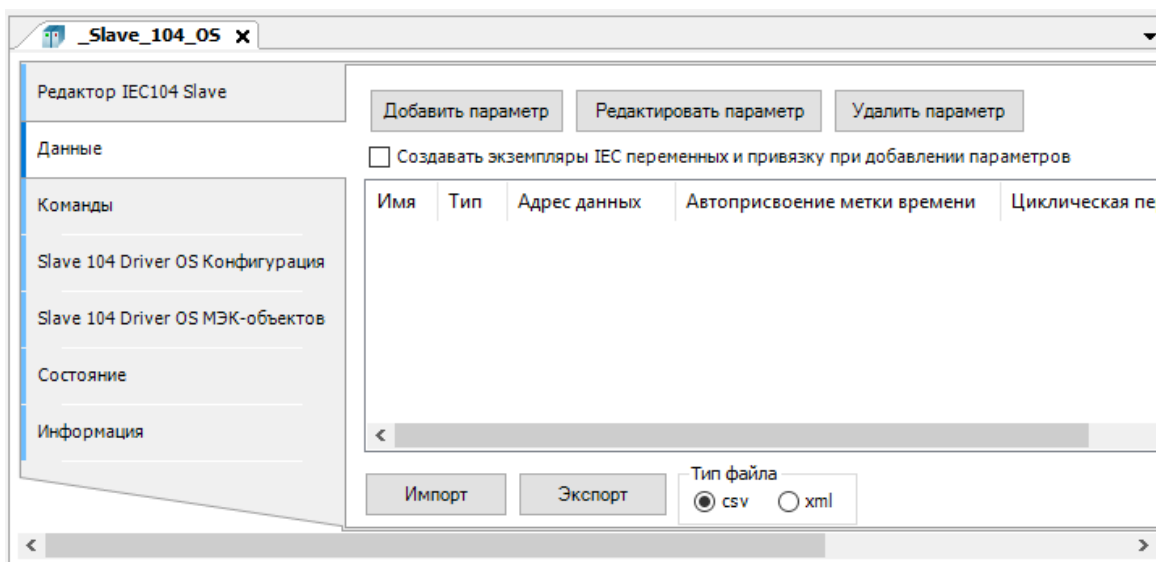


Рисунок 29 – Редактор Драйвер Slave 104 OS. Вкладка «Данные»

Для добавления, изменения и удаления элементов данных используются соответственно кнопки *Добавить параметр*, *Редактировать параметр*, *Удалить параметр*. Перейти к редактированию параметра также можно двойным щелчком левой кнопкой мыши по нужной

строке с описанием добавленного элемента. Окна ввода/редактирования элемента данных выглядят следующим образом (Рисунок 30).

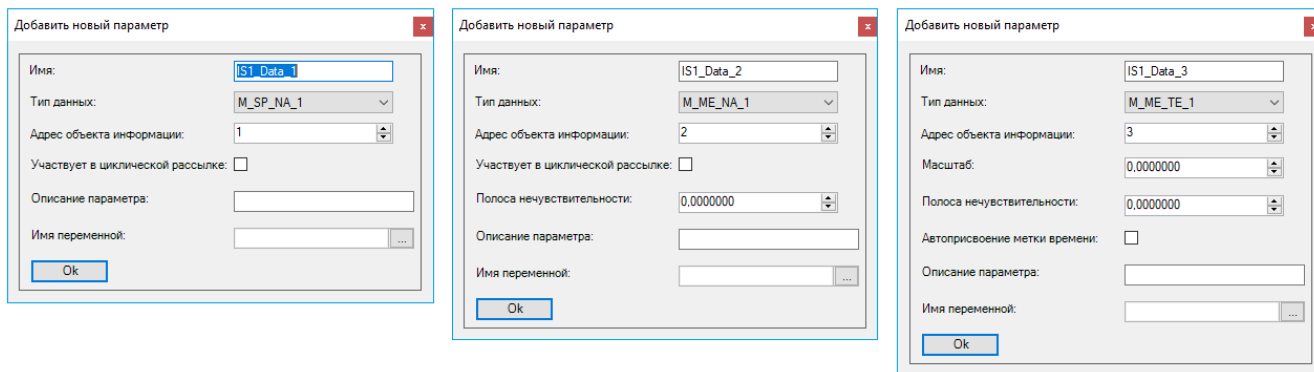




Рисунок 30 – Окна добавления нового параметра (элемента данных)

Для элемента данных укажите значения в полях. Параметры элементов представлены в таблице 11

Таблица 11 – Параметры элементов данных IEC-104 (OS)

Параметр	Описание
<b>Имя</b>	Наименование элемента данных
<b>Тип данных</b>	Ассоциированный с этим элементом данных тип IEC-104. Справочные типы данных IEC-104 приведены в таблице 12. Часто используемые типы данных: <ul style="list-style-type: none"> <li>o M_SP_TB_1 – для дискретов с меткой времени;</li> <li>o M_ME_TF_1 – для float-значений с меткой времени;</li> <li>o M_IT_TB_1 – для целочисленных значений</li> </ul>
<b>Адрес объекта информации</b>	Уникальный адрес элемента данных. Значение адреса не должно превышать максимально допустимого значения, которое обуславливается размером адресного поля PDU. Размер адресного поля PDU определяется параметром <b>Длина поля «адрес»</b> . Например, если задано значение длины поля адреса равное 2, то максимально допустимое значение адреса не может превышать 65535 (для значения 3 - не может превышать 16777215)
<b>Участует в циклической рассылке</b>	Установка флажка в этом поле активирует участие элемента данных в циклической рассылке. Для всех таких элементов их значения даже при отсутствии изменений будут отсылаяться с периодом, указанным на вкладке общих параметров. Доступно для типов: M_ME_NA_1, M_ME_NB_1, M_ME_NC_1, M_ME_ND_1 и M_SP_NA_1 (вне стандарта)
<b>Полоса нечувствительности (deadband)</b>	Устанавливаемая пользователем минимальная разница между последним отправленным значением и текущим измеренным (рассчитанным) значением в программе контроллера, превышение которой вызывает спонтанную отсылку
<b>Автоприсвоение метки</b>	При установленном в этом поле флажке временная метка элемента



Параметр	Описание
времени (для типов с меткой времени)	данных будет <i>автоматически</i> задаваться и присваиваться полю <b>timestamp</b> перед отсылкой; формат - UTC или локальное время, зависит от параметра «Использовать формат UTC при автоприсвоении меток времени»
Масштаб	Для масштабируемых параметров, при выборе типа M_ME_NB_1 и M_ME_TE_1
Описание параметра	Опционально, текстовое описание элемента данных
Имя переменной	Указывается название переменной IEC (программы контроллера), в которой хранятся передаваемые данные. Для заполнения поля <b>Имя переменной</b> нажмите в этом поле кнопку  , открывающую окно <b>Ассистент ввода</b> . Найдите нужную переменную. Если установлен флажок в поле <b>Структурированный вид</b> , то раскрывайте списки с помощью кнопки  . Если флажок снят и переменные представлены одним большим списком, для удобства поиска воспользуйтесь фильтром

Реализованы следующие идентификаторы типа, представленные в таблице 12.

Таблица 12 – Перечень типов данных IEC-104

Тип данных	ID типа данных	Описание
M_SP_NA_1	1	Одноэлементная информация
M_DP_NA_1	2	Двухэлементная информация
M_ST_NA_1	5	Информация о положении отпаяк
M_BO_NA_1	7	Строка из 32 бит
M_ME_NA_1	9	Измеряемая величина, нормализованное значение
M_ME_NB_1	11	Измеряемая величина, масштабируемое значение
M_ME_NC_1	13	Измеряемая величина, формат с плавающей запятой
M_IT_NA_1	15	Интегральная сумма
M_ME_ND_1	21	Измеряемая величина, нормализованное значение без описателя качества
M_SP_TB_1	30	Одноэлементная информация с 56-битной меткой времени
M_DP_TB_1	31	Двухэлементная информация с 56-битной меткой времени
M_ST_TB_1	32	Информация о положении отпаяк с 56-битной меткой времени
M_BO_TB_1	33	Строка из 32-х бит с 56-битной меткой времени
M_ME_TD_1	34	Измеряемая величина нормализованное значение с 56 битной меткой времени
M_ME_TE_1	35	Измеряемая величина масштабированное значение с 56 битной меткой времени

Тип данных	ID типа данных	Описание
M_ME_TF_1	36	Измеряемая величина с плавающей запятой с 56 битной меткой времени
M_IT_TB_1	37	Интегральная сумма с 56-битной меткой времени
M_EP_TD_1	38	Информация о работе релейной защиты с 56-битной меткой времени
M_LREAL_NA	129*	Информация пользовательского типа LREAL из 64-х бит + байт качества, без метки времени
M_LREAL_TC	130*	Информация пользовательского типа LREAL из 64-х бит + байт качества с 56-битной меткой времени
M_ULINT_NA	131*	Информация пользовательского типа ULINT из 64-х бит + байт качества, без метки времени
M_ULINT_TC	132*	Информация пользовательского типа ULINT из 64-х бит + байт качества с 56-битной меткой времени
<b>Примечание</b> - * - ID задается пользователем		

### Создание списка команд, получаемых по IEC-104

Кроме запроса данных с контроллера в IEC-104 реализованы команды, используемые для установки значения какой-либо переменной, либо выполнения по сигналу каких-либо действий.

В редакторе устройства Slave 104 Driver перейдите на внутреннюю вкладку **Команды**. Общий вид редактора команд аналогичен редактору элементов данных. Команды на данной вкладке описываются как параметры. Для добавления, изменения и удаления команд используются соответственно кнопки *Добавить параметр*, *Редактировать параметр*, *Удалить параметр*. Перейти к редактированию команды также можно двойным щелчком левой кнопкой мыши по нужной строке.

Окно ввода/редактирования команд выглядит следующим образом (Рисунок 31).

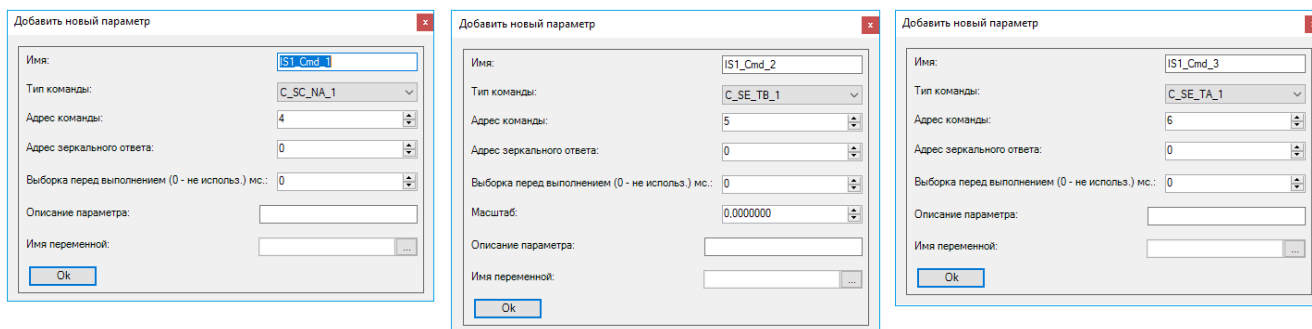


Рисунок 31 – Окно добавления новой команды

Для команд укажите значения в полях. Параметры команд представлены в таблице 13.

Таблица 13 – Параметры команд IEC-104 (OS)



Параметр	Описание
<b>Имя</b>	Наименование команды
<b>Тип данных</b>	Ассоциированный с этой командой тип IEC-104. Справочные типы данных IEC-104 приведены в таблице 14
<b>Адрес объекта информации</b>	<p>Уникальный адрес команды. Значение адреса не должно превышать максимально допустимого значения, которое обуславливается размером адресного поля PDU.</p> <p>Размер адресного поля PDU определяется параметром <b>Длина поля «адрес»</b>.</p> <p>Например, если задано значение длины поля адреса равное 2, то максимально допустимое значение адреса не может превышать 65535 (для значения 3 - не может превышать 16777215)</p>
<b>Адрес зеркального ответа</b>	Адрес элемента данных, в который будет помещено значение, переданное командой (0 – не используется)
<b>Время блокировки (0 – не использ.)</b>	Команды могут быть простыми и с «выборкой перед выполнением» (Select Before Operate). При ненулевом значении это время, в течение которого команда остается в состоянии «выбрана» для выполнения (после отправки запроса select). Успешное выполнение команды (execute) возможно до истечения данного таймаута
<b>Масштаб</b>	Для команд, задающих масштабируемые значения C_SE_NB_1 и C_SE_TB_1
<b>Описание параметра</b>	Опционально, текстовое описание команды
<b>Имя переменной</b>	<p>Указывается название переменной IEC (программы контроллера), в которой хранятся передаваемые команды.</p> <p>Для заполнения поля <b>Имя переменной</b> нажмите в этом поле кнопку , открывающую окно <b>Ассистент ввода</b>. Найдите нужную переменную. Если установлен флажок в поле <b>Структурированный вид</b>, то раскрывайте списки с помощью кнопки . Если флажок снят и переменные представлены одним большим списком, для удобства поиска воспользуйтесь фильтром</p>

Таблица 14 – Перечень типов команд IEC-104 (OS)

Тип данных	ID типа данных	Описание
C_SC_NA_1	45	Бинарная команда
C_DC_NA_1	46	Команда с тремя состояниями
C_RC_NA_1	47	Регулирующая пошаговая команда
C_SE_NA_1	48	Команда уставки (нормализованное значение)
C_SE_NB_1	49	Команда уставки (масштабируемое значение)
C_SE_NC_1	50	Команда уставки (значение типа float)



Тип данных	ID типа данных	Описание
C_BO_NA_1	51	Команда установки побитового регистра
C_SC_TA_1	58	Бинарная команда с 56-битной меткой времени
C_DC_TA_1	59	Команда с тремя состояниями с 56-битной меткой времени
C_RC_TA_1	60	Регулирующая пошаговая команда с 56-битной меткой времени
C_SE_TA_1	61	Команда уставки (нормализованное значение) с 56-битной меткой времени
C_SE_TB_1	62	Команда уставки (масштабируемое значение) с 56-битной меткой времени
C_SE_TC_1	63	Команда уставки (значение типа float) с 56-битной меткой времени
C_BO_TA_1	64	Команда установки побитового регистра с 56-битной меткой времени

### Создание переменных для работы с данными/командами, передаваемыми по IEC-104

Элементы данных и команды, описанные в редакторе IEC-104, в терминологии среды разработки являются **каналами ввода-вывода**. Для реализации обмена данными необходимо создать эти каналы (см. предыдущие разделы), создать в программном коде переменные специальных типов, после чего связать переменные с каналами ввода-вывода.

Требуемые типы переменных (функциональные блоки) описаны в общей библиотеке **PsIEC60870Bridge**. Эта библиотека, а также использующая ее библиотека **PsIoDrvIec104Slave\_OS**, автоматически подключаются при добавлении устройства Драйвер Slave 104 OS.

В таблице 15 приведено описание поддерживаемых типов данных IEC-104 и соответствующих им функциональных блоков с указанием главы стандарта IEC 60870-5-101 для этого типа и перечислением полей функционального блока для работы с его значением, качеством и меткой времени.

Таблица 15 – Соответствие функциональных блоков и типов данных/команд IEC-104

№	Функц.блок	Тип IEC-104	Применение	Поля функционального блока
1	m_sp_na_1_fb	M_SP_NA_1 (IEC 60870-5-101, 7.2.6.1)	Дискрет: 0, 1	Значение (BYTE): SPI, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB
2	m_sp_tb_1_fb	M_SP_TB_1 (IEC 60870-5-101, 7.3.1.22)	Дискрет: 0, 1 с временной меткой CP56Time2a	Наследует от m_sp_na_1_fb Время: timestamp (psCP56Time2a_t)

№	Функц.блок	Тип IEC-104	Применение	Поля функционального блока
3	m_dp_na_1_fb	M_DP_NA_1 (IEC 60870-5-101, 7.3.1.3)	2-ух-битовое поле: 0-3	Значение (BYTE): DPI, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB
4	m_dp_tb_1_fb	M_DP_TB_1 (IEC 60870-5-101, 7.3.1.23)	2-ух-битовое поле: 0-3 с временной меткой CP56Time2a	Наследует от m_dp_na_1_fb Время: timestamp (psCP56Time2a_t)
5	m_st_na_1_fb	M_ST_NA_1 (IEC 60870-5-101, 7.3.1.5)	7-битовое значение (-63 ... +64) с битом переходного состояния (0,1)	Значение (BYTE): VTI, value, TRANS Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
6	m_st_tb_1_fb	M_ST_TB_1 (IEC 60870-5-101, 7.3.1.24)	7-битовое значение (-63 ... +64) с битом переходного состояния (0,1) с временной меткой CP56Time2a	Наследует от m_st_na_1_fb Время: timestamp (psCP56Time2a_t)
7	m_bo_na_1_fb	M_BO_NA_1 (IEC 60870-5-101, 7.3.1.7)	4-байтное целое без знака (32-битный BITSTRING)	Значение (DWORD): BSI, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
8	m_bo_tb_1_fb	M_BO_TB_1 (IEC 60870-5-101, 7.3.1.25)	4-байтное целое без знака (32-битный BITSTRING) с временной меткой CP56Time2a	Наследует от m_bo_na_1_fb Время: timestamp (psCP56Time2a_t)
9	m_me_na_1_fb	M_ME_NA_1 (IEC 60870-5-101, 7.3.1.9)	Нормализованное число с плавающей точкой: -1 ... +1-2 <sup>-15</sup>	Значение (REAL): NVA, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
10	m_me_td_1_fb	M_ME_TD_1 (IEC 60870-5-101, 7.3.1.26)	Нормализованное число с плавающей точкой: -1 ... +1-2 <sup>-15</sup> с временной меткой CP56Time2a	Наследует от m_me_na_1_fb Время: timestamp (psCP56Time2a_t)
11	m_me_nb_1_fb	M_ME_NB_1 (IEC 60870-5-101, 7.3.1.11)	Масштабированное значение: -2 <sup>15</sup> ... +2 <sup>15</sup> -1. В описании канала задается коэффициент масштаба	Значение (REAL): SVA, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV

№	Функц.блок	Тип IEC-104	Применение	Поля функционального блока
12	m_me_te_1_fb	M_ME_TE_1 (IEC 60870-5-101, 7.3.1.27)	Масштабированное значение: -2 <sup>15</sup> ... +2 <sup>15</sup> -1 с временной меткой CP56Time2a. В описании канала задается коэффициент масштаба	Наследует от m_me_nb_1_fb Время: timestamp (psCP56Time2a_t)
13	m_me_nc_1_fb	M_ME_NC_1 (IEC 60870-5-101, 7.3.1.13)	4-байтное число с плавающей запятой (короткий формат)	Значение (REAL): R32, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
14	m_me_tf_1_fb	M_ME_TF_1 (IEC 60870-5-101, 7.3.1.28)	4-байтное число с плавающей запятой (короткий формат) с временной меткой CP56Time2a	Наследует от m_me_nc_1_fb Время: timestamp (psCP56Time2a_t)
15	m_it_na_1_fb	M_IT_NA_1 (IEC 60870-5-101, 7.3.1.15)	4-байтное целое со знаком	Значение (DINT): COUNTER, value Качество (BYTE): quality, SN_SQ, SN_CY, SN_CA, SN_IV, SN
16	m_it_tb_1_fb	M_IT_TB_1 (IEC 60870-5-101, 7.3.1.29)	4-байтное целое со знаком с временной меткой CP56Time2a	Наследует от m_it_na_1_fb Время: timestamp (psCP56Time2a_t)
17	m_me_nd_1_fb	M_ME_ND_1 (IEC 60870-5-101, 7.3.1.21)	Нормализованное число с плавающей точкой: -1 .. +1-2 <sup>15</sup> без описателя качества	Значение (REAL): NVA, value
18	m_ep_td_1_fb	M_EP_TD_1 (IEC 60870-5-101, 7.3.1.30)	События устройств защиты с временной меткой CP56Time2a	Значение (BYTE): SEP, ES, value Интервал (UINT): RELAY Качество (BYTE): quality, QD_EI, QD_BL, QD_IV, QD_NT, QD_SB Время: timestamp (psCP56Time2a_t)
19	m_lreal_na_1_fb	M_LREAL_NA_1 Custom type	8-байтное число с плавающей запятой	Значение (LREAL): R64, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
20	m_lreal_tc_1_fb	M_LREAL_TC_1 Custom type	8-байтное число с плавающей запятой с временной меткой CP56Time2a	Наследует от m_lreal_na_1_fb Время: timestamp (psCP56Time2a_t)

№	Функц.блок	Тип IEC-104	Применение	Поля функционального блока
21	m_ulint_na_1_fb	M_ULINT_NA_1 Custom type	8-байтное целое беззнаковое	Значение (ULINT): COUNTER, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
22	m_ulint_tc_1_fb	M_ULINT_TC_1 Custom type	8-байтное целое беззнаковое с временной меткой CP56Time2a	Наследует от m_ulint_na_1_fb Время: timestamp (psCP56Time2a_t)
23	c_sc_na_1_fb	C_SC_NA_1 (IEC 60870-5-101, 7.3.2.1)	Однопозиционная команда, дискреты: 0,1	Значение (BYTE): SCO, SCS, value Описатель (BYTE): QOC, QU Действие (BYTE): SE
24	c_sc_ta_1_fb	C_SC_TA_1 (IEC 60870-5-101, 7.3.2.1)	Однопозиционная команда с временной меткой CP56Time2a, дискреты: 0,1	Наследует от c_sc_na_1_fb Время: timestamp (psCP56Time2a_t)
25	c_dc_na_1_fb	C_DC_NA_1 (IEC 60870-5-101, 7.3.2.2)	Двухпозиционная команда с 2-битовым полем	Значение (BYTE): DCO, DCS, value Описатель (BYTE): QOC, QU Действие (BYTE): SE
26	c_dc_ta_1_fb	C_DC_TA_1 (IEC 60870-5-101, 7.3.2.2)	Двухпозиционная команда с временной меткой CP56Time2a, с 2-битовым полем	Наследует от c_dc_na_1_fb Время: timestamp (psCP56Time2a_t)
27	c_rc_na_1_fb	C_RC_NA_1 (IEC 60870-5-101, 7.3.2.3)	Команда пошагового регулирования	Значение (BYTE): RCO, RCS, value Описатель (BYTE): QOC, QU Действие (BYTE): SE
28	c_rc_ta_1_fb	C_RC_TA_1 (IEC 60870-5-101, 7.3.2.3)	Команда пошагового регулирования с временной меткой CP56Time2a	Наследует от c_rc_na_1_fb Время: timestamp (psCP56Time2a_t)
29	c_se_na_1_fb	C_SE_NA_1 (IEC 60870-5-101, 7.3.2.4)	Команда уставки, нормализованное значение	Значение (REAL): NVA, value Описатель (BYTE): QOS, QL Действие (BYTE): SE
30	c_se_ta_1_fb	C_SE_TA_1 (IEC 60870-5-101, 7.3.2.4)	Команда уставки, нормализованное значение с временной меткой CP56Time2a	Наследует от c_se_na_1_fb Время: timestamp (psCP56Time2a_t)

№	Функц.блок	Тип IEC-104	Применение	Поля функционального блока
31	c_se_nb_1_fb	C_SE_NB_1 (IEC 60870-5-101, 7.3.2.5)	Команда уставки, масштабированное значение	Значение (INT): SVA, (REAL) value Описатель (BYTE): QOS, QL Действие (BYTE): SE
32	c_se_tb_1_fb	C_SE_TB_1 (IEC 60870-5-101, 7.3.2.5)	Команда уставки, масштабированное значение с временной меткой CP56Time2a	Наследует от c_se_nb_1_fb Время: timestamp (psCP56Time2a_t)
33	c_se_nc_1_fb	C_SE_NC_1 (IEC 60870-5-101, 7.3.2.6)	Команда уставки, короткий формат с плавающей запятой	Значение (REAL): R32, value Описатель (BYTE): QOS, QL Действие (BYTE): SE
34	c_se_tc_1_fb	C_SE_TC_1 (IEC 60870-5-101, 7.3.2.6)	Команда уставки, короткий формат с плавающей запятой с временной меткой CP56Time2a	Наследует от c_se_nc_1_fb Время: timestamp (psCP56Time2a_t)
35	c_bo_na_1_fb	C_BO_NA_1 (IEC 60870-5-101, 7.3.2.7)	Строка из 32 бит	Значение (DWORD): BSI, value
36	c_bo_ta_1_fb	C_BO_TA_1 (IEC 60870-5-101, 7.3.2.7)	Строка из 32 бит с временной меткой CP56Time2a	Наследует от c_bo_na_1_fb Время: timestamp (psCP56Time2a_t)

### Особенности работы с функциональными блоками данных

Для всех типов поле **value** является унифицированным синонимом поля фактического значения конкретного типа, например, **R32** для M\_ME\_NC\_1 или **SPI** для M\_SP\_NA\_1, и взаимозаменяемо с ним в коде приложения.

Однако, для некоторых типов, **value** является частью общей структуры поля данных. Например, для M\_ST\_NA\_1, **value** представляет младшие 7 бит поля **VTI**, а **TRANS** – старший восьмой бит. Для M\_EP\_TD\_1, **value** является синонимом **ES** – поля состояния события, представляющего два младших бита поля **SEP**.

Для типов с плавающей запятой (M\_ME\_NA\_1, M\_ME\_NB\_1, M\_ME\_NC\_1, M\_ME\_ND\_1 и их вариантов с меткой времени), **value** работает на входе/выходе со значением в формате REAL, при этом нормализация и масштабирование значения производятся только при сериализации типа в процессе отправки данных мастеру.

Для работы с отдельными (битовыми) полями качества общего описателя качества (QD, Quality Descriptor) элемента данных пользователю предоставляются следующие свойства функционального блока:

- QD\_BL – наличие (1) / отсутствие (0) блокировки;
- QD\_SB – наличие (1) / отсутствие (0) замещения;
- QD\_NT – неактуальное (1) / актуальное (0) значение;
- QD\_IV – недействительное (1) / действительное (0) значение;
- QD\_OV – есть (1) / нет (0) переполнения;
- QD\_EI – недействительное (1) / действительное (0) значение интервала времени.

Эти поля имеют тип BYTE и на входе принимают «0» / «не 0» (преобразуется в «1»), на выходе отдают «0» / «1».

Также, с каждым типом, пользователь может использовать поле описателя качества с унифицированным именем quality, содержащим логическое объединение вышеуказанных битов качества со смещениями, соответствующими описанию этого конкретного типа в стандарте. Это поле имеет тип BYTE. При его изменении, соответственно, обновляются и все отдельные битовые поля описателя качества, и наоборот.

Для типов с временной меткой используется структура данных psCP56Time2a\_t, предоставляющая доступ ко всем полям метки времени CP56Time2a (IEC 60870-5-101, 7.2.6.18):

```

TYPE psCP56Time2a_t :
STRUCT
    year          : BYTE; // Year - <0 .. 99>
    month         : BYTE; // Month - <1 .. 12>
    day           : BYTE; // Day - <1 .. 31>
    hour          : BYTE; // Hour - <0 .. 23>
    minute        : BYTE; // Minute - <0 .. 59>
    second        : BYTE; // Second - <0 .. 59>
    milliseconds : WORD; // Milliseconds - <0 .. 999>
    invalid       : BYTE; // Invalid - <0>:=valid, <1>:=invalid
    summer        : BYTE; // Summer time - <0>:=standard time, <1>:=summer time
    day_of_week   : BYTE; // Day of week - <1 .. 7>, Monday=1, Sunday=7
    res1_23       : BIT; // RES1;
    res2_30       : BIT; // RES2;
    res2_31       : BIT; // RES2;
    res3_45       : BIT; // RES3;
    res3_46       : BIT; // RES3;
    res3_47       : BIT; // RES3;
    res3_48       : BIT; // RES3;
    res4_56       : BIT; // RES4;
END_STRUCT
END_TYPE
    
```

Также, в общей библиотеке PsIEC60870Bridge доступна функция получения текущего времени *get\_psCP56Time2a\_now(xUTC : BOOL)* в виде структуры psCP56Time2a\_t в локальном или UTC формате. При этом все поля, не относящиеся к времени, зануляются.

Для функциональных блоков типов с меткой времени предусмотрен метод `set_timestamp_flags(...)`, позволяющий задавать все, не относящиеся к времени, флаги структуры. Метод вызывается для экземпляра ФБ соответствующего типа, например:

```
data_me_tf_1 : PsIEC60870Bridge.m_me_tf_1_fb;  
...  
data_me_tf_1.set_timestamp_flags(1,0,0,0,0,0,0,0,0,0);
```

Если в описании канала установлена галочка «Автоприсвоение метки времени», то, при отправке элемента данных, временная метка будет заполняться автоматически с форматом согласно настройке «Использовать формат UTC при автоприсвоении меток времени». Автоприсвоение метки времени имеет приоритет над ее ручным изменением через поле `timestamp`. При этом, флаги, ранее выставленные с помощью метода `set_timestamp_flags(...)`, не меняются, обновляются только поля даты и времени.

### Автоматическая генерация переменных и привязка к каналам ввода-вывода

При добавлении параметров на вкладках **Данные** и **Команды** можно активировать режим, при котором для каждого канала автоматически создаются переменные ПЛК-программы и формируется привязка переменной к каналу. Для этого при создании нового элемента данных или команды установите флажок в поле **Создавать экземпляры IEC переменных и привязку при добавлении параметров** (Рисунок 32).

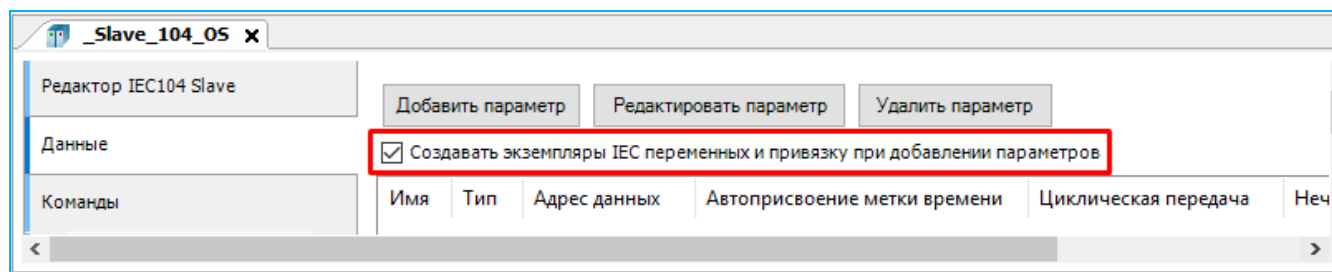


Рисунок 32 – Установка флажка в поле «Создавать экземпляры IEC переменных и привязку при добавлении параметров»

В этом режиме при добавлении канала для устройства Драйвер Slave 104 OS создается собственный список глобальных переменных с именем вида `I104_GVL_X`, где X – условный сквозной порядковый номер устройства, начиная с единицы. В этом списке будет создана переменная соответствующего типа.



## ИНФОРМАЦИЯ

Допускается использование одних и тех же экземпляров переменных (и данных, и команд) для привязки к разным экземплярам драйверов (Master/Slave, 101/104 OS), при условии, что настройки привязанной переменной (параметры элементов данных/команд) для всех экземпляров драйверов идентичны.

На этапе компиляции проекта выполняется проверка совпадения настроек, в случае несовпадения, в окно сообщений выводится сообщение об ошибке, с указанием имени переменной и имен соответствующих экземпляров драйверов

## Реализация спорадического метода передачи

Согласно стандарту, под изменением состояния информационного объекта понимается не только изменение его текущего значения, но и изменение любого признака его описателя качества. Изменение у информационного объекта только «метки времени» не считается изменением его состояния.

По умолчанию, при изменении значения (или одного из его полей) или качества (или любого его признака) информационного объекта, происходит добавление снимка его текущего состояния в очередь отправки спонтанных данных с причиной передачи <3>. Это может приводить к нежелательному поведению, когда, например, в коде пользователя присутствуют несколько следующих друг за другом изменений полей данных одного экземпляра объекта:

```
data_me_tf_1          : PsIEC60870Bridge.m_me_tf_1_fb;
...
data_me_tf_1.value := 123.45;
data_me_tf_1.QD_SB := 1;
data_me_tf_1.timestamp := PsIEC60870Bridge.get_psCP56Time2a_now(TRUE);
```

Вторая и третья строки вызовут отправки двух состояний объекта, каждое с одним измененным полем. Изменение временной метки будет зафиксировано, отправки данных не последует.

Для поддержания консистентности данных на уровне экземпляра функционального блока информационного объекта предлагается использовать один из двух режимов работы драйвера **Slave 104 OS**.

### Режим автоматической отправки данных (включен по умолчанию)

В этом режиме происходит автоматическая отправка данных с причиной передачи <3>, при изменении значения (или одного из его полей) или качества (или любого его признака) информационного объекта. При этом, используется следующий механизм блокировки отправки спонтанных данных - блок кода, в котором производятся изменения различных полей объекта данных, заключается в вызовы соответствующих методов:

```
data_me_tf_1          : PsIEC60870Bridge.m_me_tf_1_fb;
...
data_me_tf_1.lock_send();           // блокировка отправки
data_me_tf_1.value := 123.45;
```



```
data_me_tf_1.QD_SB := 1;
data_me_tf_1.QD_IV := 0;
data_me_tf_1.timestamp := PsIEC60870Bridge.get_psCP56Time2a_now(TRUE);
data_me_tf_1.unlock_send(); // отправка измененных данных
```

Метод **unlock\_send()** вызывает автоматическую отправку данных, в случае фактического изменения значения и/или качества объекта, с учетом заданного диапазона полосы нечувствительности для этого информационного объекта.

При необходимости, включение этого режима для каждого отдельного экземпляра драйвера **Slave 104 OS** производится через свойство **SendMode**:

```
Slave_104_OS.SendMode := PsIEC60870Bridge.e_sm_auto;
```

### Режим ручной отправки данных

В ручном режиме не происходит автоматической отправки данных, при изменении значения или качества информационного объекта. Для помещения состояния экземпляра функционального блока в очередь отправки спонтанных данных необходимо в коде вызвать для него метод **send()**:

```
data_me_tf_1.value := 123.45;
data_me_tf_1.QD_SB := 1;
data_me_tf_1.QD_IV := 0;
data_me_tf_1.timestamp := PsIEC60870Bridge.get_psCP56Time2a_now(TRUE);
data_me_tf_1.send(); // отправка измененных данных
```

При необходимости, включение режима для каждого отдельного экземпляра драйвера **Slave 104 OS** производится также через свойство **SendMode**:

```
Slave_104_OS.SendMode := PsIEC60870Bridge.e_sm_manual;
```

Дополнительно, в ручном режиме работы предусмотрена возможность отправки набора всех измененных на текущий момент времени данных. Для этого, для экземпляра драйвера **Slave 104 OS**, в коде, необходимо вызвать метод **send()**:

```
data_me_tf_1.value := 123.45;
data_me_tf_1.QD_SB := 1;
data_me_tf_01.value := 123;
data_me_tf_02.quality := 0x80;
data_me_tf_03.timestamp := PsIEC60870Bridge.get_psCP56Time2a_now(TRUE);
...
data_me_tf_KK.value := 543.21;
...
data_me_tf_NN.timestamp := PsIEC60870Bridge.get_psCP56Time2a_now(TRUE);

Slave_104_OS.send();
```



### ИНФОРМАЦИЯ

В обоих режимах отправки пользователю предоставляется возможность безусловной отправки данных без учета их изменения и настройки полосы нечувствительности deadband, т.е. отправки текущего состояния экземпляра функционального блока, вызвав для него в коде метод **send\_now()**:

```
data_me_tf_1.send_now();
```



### ИНФОРМАЦИЯ

Методы lock\_send(), unlock\_send(), send() и send\_now() ФБ данных доступны для работы только при привязке переменных этих типов к экземплярам драйверов Slave 104 OS и Slave 101 OS

## Реализация общего опроса станции

В драйвере реализовано три режима отправки данных в ответ на команду общего опроса станции (значение указателя опроса QOI = 20). Выбор режима задается с помощью свойства ФБ драйвера *InterrogationProfile*, при присвоении ему одного из значений перечисления *interrogation\_profile\_type\_enum*, описанного в библиотеке **PsIEC60870Bridge**:

- 1) Slave104.InterrogationProfile := PsIEC60870Bridge.standard;

Отправка данных согласно стандарту, включая следующие типы: M\_SP\_NA\_1, M\_DP\_NA\_1, M\_ST\_NA\_1, M\_BO\_NA\_1, M\_ME\_NA\_1, M\_ME\_NB\_1, M\_ME\_NC\_1, M\_PS\_NA\_1, M\_ME\_ND\_1;

- 2) Slave104.InterrogationProfile := PsIEC60870Bridge.custom\_ps;

Отправка данных включает все типы, описанные в п.1 и M\_IT\_NA\_1.

Переменные с типами с временной меткой высылаются с соответствующим типом без временной метки, то есть M\_SP\_TA\_1 ⇒ M\_SP\_NA\_1 и т.д. Этот режим включен по умолчанию;

- 3) Slave104.InterrogationProfile := PsIEC60870Bridge.custom\_tn;

Отправка данных включает все типы, описанные в п.1 с причиной передачи COT = 20. Переменные с типами с временной меткой высылаются с причиной передачи COT= 3.

## Особенности работы с функциональными блоками команд

Для типов C\_BO\_xA\_1 и C\_SE\_xC\_1 поле **value** является синонимом поля фактического значения конкретного типа, например, **BSI** для C\_BO\_xA\_1 и **R32** для C\_SE\_xC\_1, и взаимозаменяемо с ним в коде приложения.

Для типов **C\_SC\_xA\_1**, **C\_DC\_xA\_1** и **C\_RC\_xA\_1** **value** является частью общей структуры поля данных. Например, для **C\_SC\_xA\_1** **value** представляет младший (**SCS**) бит поля **SCO**.

Функциональный блок типа **C\_SE\_xA\_1** принимает и отдает значения типа **REAL** в диапазоне от -1 до  $+1 \cdot 2^{-15}$  через свои поля **NVA** и **value**. Перевод в нормализованное число с фиксированной запятой со знаком ( $F_{16}$ , IEC 60870-5-101, 7.2.6.6, Тип 4.1) производится только при сериализации типа, в процессе обмена данными с мастером.

Для типа **C\_SE\_xB\_1** поле **value** принимает и отдает значения типа **REAL**. Само значение хранится во внутреннем представлении в виде масштабированного 2-х байтного целого со знаком ( $I_{16}$ , IEC 60870-5-101, 7.2.6.7, Тип 2.1) и доступно для записи/чтения через поле **SVA** (**INT**). Масштабирование производится путем деления полученного значения типа **REAL** на коэффициент масштабирования (положение десятичной запятой), который задается в описании канала, а также доступен для изменения через поле функционального блока **scale**.

При работе с полями значения и описателя функциональных блоков команд следует учитывать то, что составные поля используются «как есть» - согласно описанию данного типа, в терминах битовых полей в стандарте, а к отдельным полям применяется смещение значения вправо.

Например, для **C\_SC\_xA\_1** (**SCO** := CP8{**SCS**, **RES**, **QOC**}, IEC 60870-5-101, 7.2.6.15) составные поля типа **SCO** и описателя **QOC** обрабатываются, соответственно, как байт и битовое поле этого байта по маске с 3 по 8 бит. Отдельные поля значения **SCS** и описателя команды **QU** и **SE** сдвигаются (при необходимости) вправо, до первого значащего бита этого поля. Например, поле **SCS** представляет 1-ый бит в байте **SCO**, поэтому, он используется без сдвига, принимая на входе значения «0»/ «не 0» (преобразуется в «1»), и отдавая на выходе «0» / «1». Битовое поле **QU** с 3 по 7 бит сдвигается на два бита вправо, принимая (отдавая) значения из диапазона [0,31] (IEC 60870-5-101, 7.2.6.26). Битовое поле **SE**, представляющее 8-ой бит поля **SCO** сдвигается вправо на семь позиций, принимая на входе значения «0»/ «не 0» (преобразуется в «1»), и отдавая на выходе «0» / «1».

Для типа **C\_SE\_xA\_1** (CP24{**NVA**, **QOS**}, IEC 60870-5-101, 7.2.6.6 и 7.2.6.39) составное поле описателя команды **QOS** обрабатывается как байт. Его отдельные битовые поля **QL** и **SE** сдвигаются (при необходимости) вправо, до первого значащего бита этого поля. Битовое поле **QL** с 1 по 7 используется без сдвига, принимая (отдавая) значения из диапазона [0,127] (IEC 60870-5-101, 7.2.6.39). Битовое поле **SE**, представляющее 8-ой бит поля **QOS** сдвигается вправо на семь позиций, принимая на входе значения «0»/ «не 0» (преобразуется в «1»), и отдавая на выходе «0» / «1».

## Настройка контроллера в качестве Master 104

Контроллер выступает в роли ведущего, опрашивающего устройства. При настройке режима необходимо задать параметры slave-устройства, которое будет опрашиваться контроллером. Кроме того, требуется описать набор данных, который будет запрашиваться.

### Добавление устройства Master 104 Driver

В окне дерева устройств поместите курсор на название контроллера, нажмите правую кнопку мыши. В появившемся контекстном меню выберите пункт **Добавить устройство...** Откроется окно **Добавить устройство**, в котором выберите: *Regul* → *МЭК 60870* → *TCP 60870-104-Master* → *Master 104 Driver*. Нажмите кнопку **Добавить устройство** или дважды щелкните левой кнопкой мыши. Выбранное устройство появится в проекте в дереве устройств.

Далее к устройству **Master 104 Driver** нужно подключить одно или несколько slave-устройств (outer slaves), которые будут опрашиваться контроллером: *Regul* → *МЭК 60870* → *TCP 60870-104-Master* → *IEC 104 Outer Slave* (Рисунок 33).

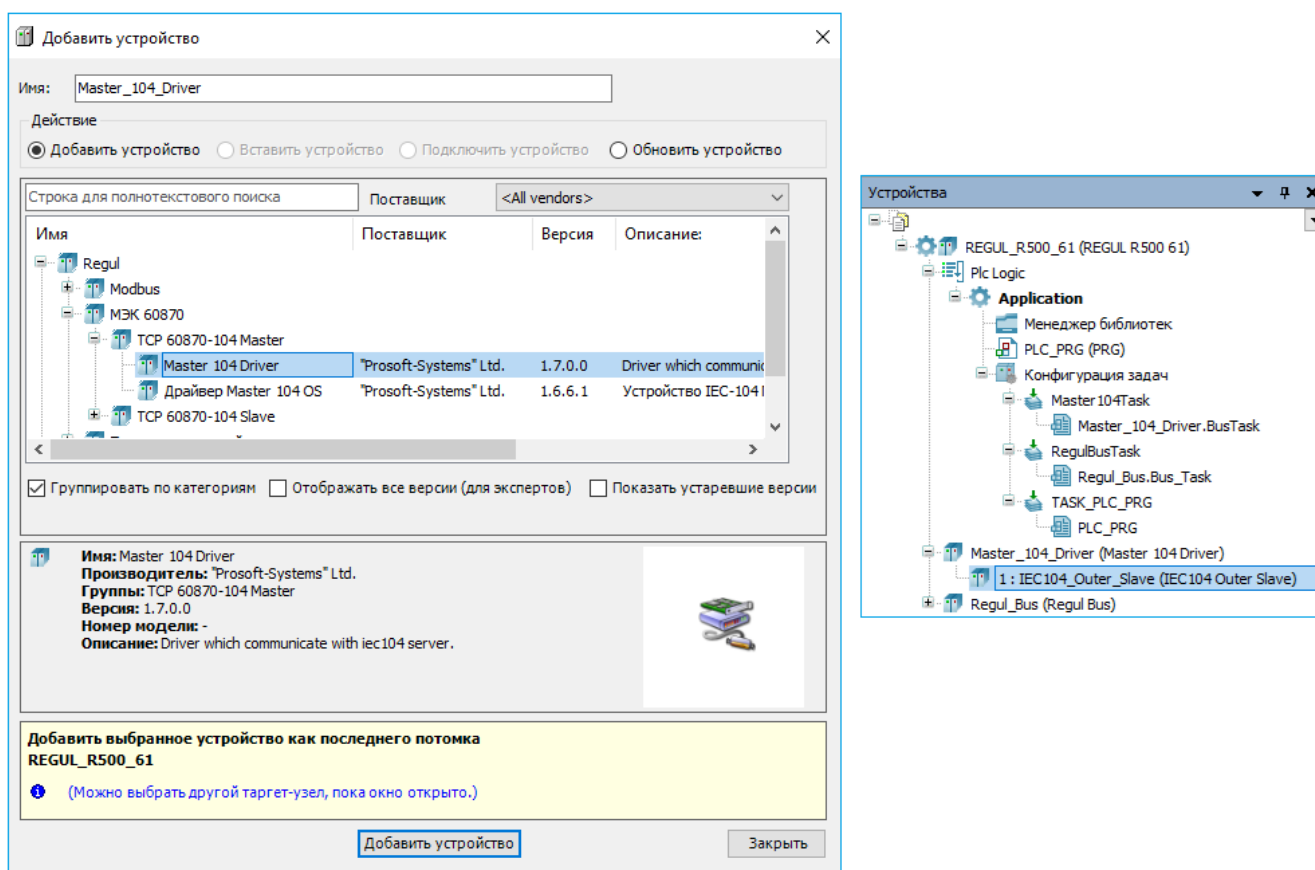


Рисунок 33 – Добавление в конфигурацию контроллера устройства IEC 104 Outer Slave

Двойным щелчком по названию устройства **IEC 104 Outer Slave** откройте вкладку параметров. По умолчанию открывается первая внутренняя вкладка **Редактор IEC 104 Outer Slave** (Рисунок 34).

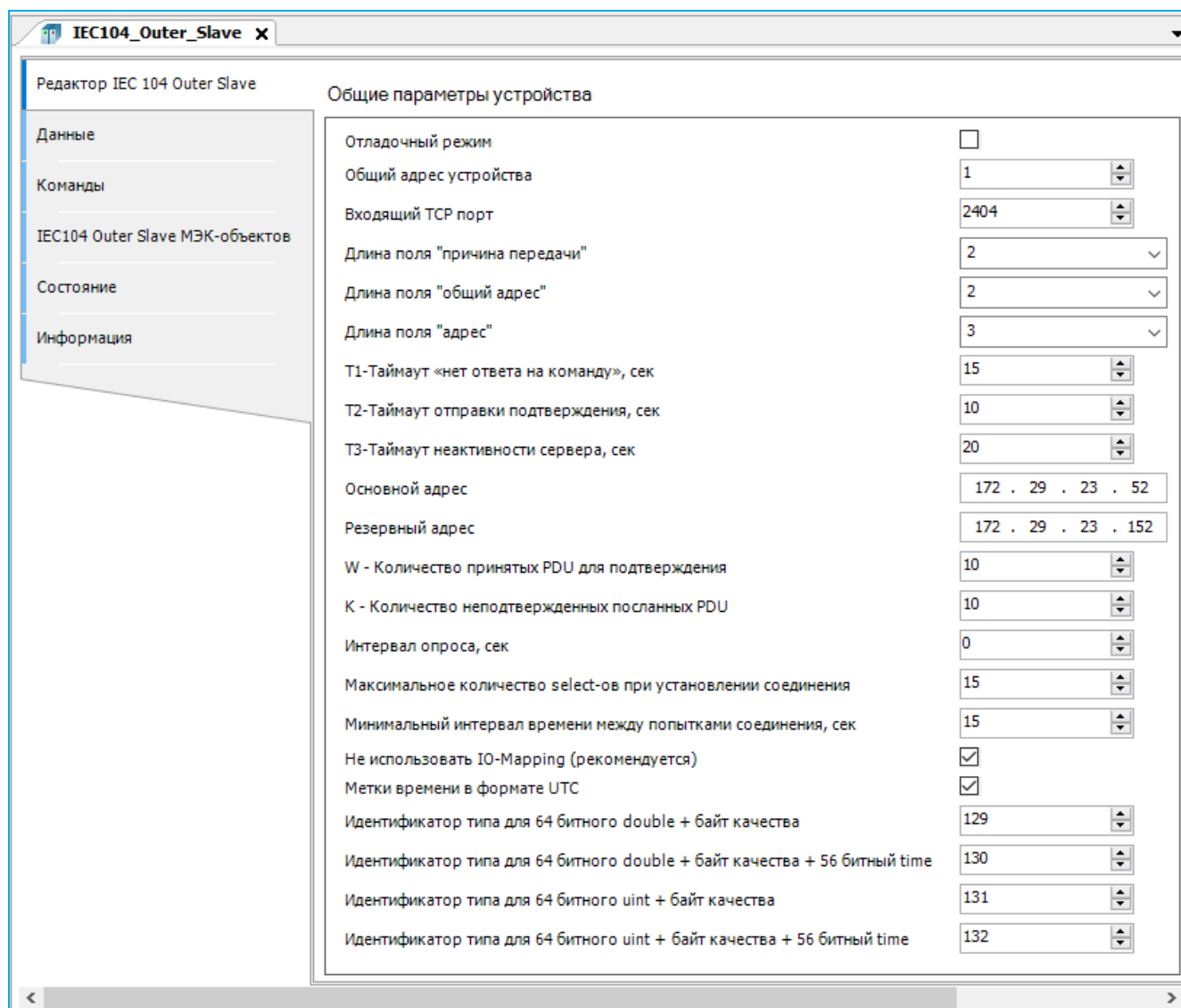


Рисунок 34 – Редактор устройства IEC 104 Outer Slave

### Настройка общих параметров устройства IEC 104 Outer Slave

В редакторе устройства **IEC 104 Outer Slave** в блоке **Общие параметры устройства** доступны для настройки параметры, представленные в таблице 16.

Таблица 16 – Общие параметры устройства

Параметр	Описание	Значение по умолчанию
<b>Отладочный режим</b>	При установке флажка <input checked="" type="checkbox"/> в этом поле будут записаны в журнал работы контроллера все детали обмена по IEC-104, а именно передаваемые и принимаемые пакеты, комментарии к возникающим ошибкам и т.п.	
<b>Общий адрес устройства</b>	Содержит общий адрес (COMMON ADDRESS, ASDU ADDRESS) устройства, все данные и команды имеют общую часть, равную значению этого параметра (см. протокол IEC-104)	1

Параметр	Описание	Значение по умолчанию
<b>Входящий TCP порт</b>	Номер порта TCP, который открыт данным slave для приема входящих соединений	2404
<b>Длина поля «причина передачи»</b>	Содержит длину поля «причина передачи» (COT, Cause Of Transmission). Допустимые значения параметра: 1 или 2	2
<b>Длина поля «общий адрес», Длина поля «адрес»</b>	Характеристики PDU (Protocol Data Unit – протокольная единица, пакет), передаваемых по IEC-104, должны быть одинаковы для пары master slave. Допустимые значения параметра «общий адрес»: 1 или 2. Допустимые значения параметра «адрес»: 2 или 3	«общий адрес» – 2, «адрес» – 3
<b>T1-Таймаут «нет ответа на команду», сек</b>	Время, в течение которого ожидается ответ на команду или подтверждение доставки PDU. Превышение таймаута приводит к разрыву соединения	15
<b>T2-Таймаут отправки подтверждения, сек</b>	При получении PDU запускается таймер на T2 секунд, подтверждение о приеме (пакет S-PDU) отсылается либо по истечению этого таймера, либо по факту приема некоторого предельного количества PDU, указанного в параметре W. T2 должен быть меньше T1	10
<b>T3-Таймаут неактивности сервера, сек</b>	Периодически как master, так и slave могут отсылать специальные пакеты для проверки связи. Если в течение интервала, заданного в параметре T3, не было никакой активности со стороны клиента, то посылаются тестовое PDU и включается таймер T1. Если в течении интервала T1 не было активности со стороны другой стороны обмена, соединение разрывается	20
<b>Основной адрес</b>	IP-адрес основного соединения	
<b>Резервный адрес</b>	IP-адрес резервного соединения	
<b>W-Количество принятых PDU для подтверждения</b>	Количество принятых PDU, которое необходимо подтверждать. Рекомендуем выбирать значение W, не превышающее 2/3 от значения (Slave). Параметры должны быть согласованы на Master и Slave. Например, при K=10 на Slave, W=8 на Master.	10
<b>K-Количество неподтвержденных посланных PDU</b>	Количество неподтвержденных отправленных PDU, после которого драйвер прекращает отправлять PDU, дожидаясь подтверждения отправленных PDU.  Следует отметить, что в случае, когда требуется отправка большого количества данных, значение параметра K следует устанавливать большим. Например, если требуется отправка 20000 отсчетов в секунду, значение K нужно устанавливать до 150. Иначе драйвер будет приостанавливать отправку данных, ожидая подтверждения клиента, размеры очередей при этом будут нарастать.  Для оптимальной работы драйвера рекомендуется ограничить отправку на уровне 25000 отсчетов в секунду.	10

Параметр	Описание	Значение по умолчанию
	Отсчет – изменение одного элемента данных. Если введено 500 элементов данных и все они изменяются раз в 10 мс, то это равносильно $500 \cdot 1000 / 10 = 50000$ отсчетов в секунду	
<b>Интервал опроса, сек</b>	Период отсылки команды общего опроса. Значение 0 – отсылка команды общего опроса не производится	0
<b>Максимальное количество select-ов при установлении соединения</b>	Максимальное количество запросов типа «Выборка перед выполнением» (Select before operate)	15
<b>Минимальный интервал времени между попытками соединения, сек (T0)</b>	Промежуток времени в секундах между попытками установления соединения с сервером	15
<b>Не использовать IO-Mapping (рекомендуется)</b>	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что чтение/запись привязанных переменных происходит по адресу (прямой доступ к памяти ПЛК), без использования механизма I/O Mapping (соотнесение входов/выходов)	
<b>Метки времени в формате UTC</b>	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что значение метки времени соответствует UTC, иначе - локального времени	
<b>Идентификатор типа для 64 битного double + байт качества</b>	Задание ID для пользовательского типа LREAL(double), без метки времени	129
<b>Идентификатор типа для 64 битного double + байт качества + 56 битный time</b>	Задание ID для пользовательского типа LREAL(double), с меткой времени	130
<b>Идентификатор типа для 64 битного uint + байт качества</b>	Задание ID для пользовательского типа ULINT(uint), без метки времени	131
<b>Идентификатор типа для 64 битного uint + байт качества + 56 битный time</b>	Задание ID для пользовательского типа ULINT(uint), с меткой времени	132

### Создание списка элементов данных, передаваемых по IEC-104

Список элементов данных имеет ограничение в 10000 элементов для каждого устройства Slave 104 Outer Slave.

В редакторе устройства **IEC 104 Outer Slave** перейдите на внутреннюю вкладку **Данные** (Рисунок 35).

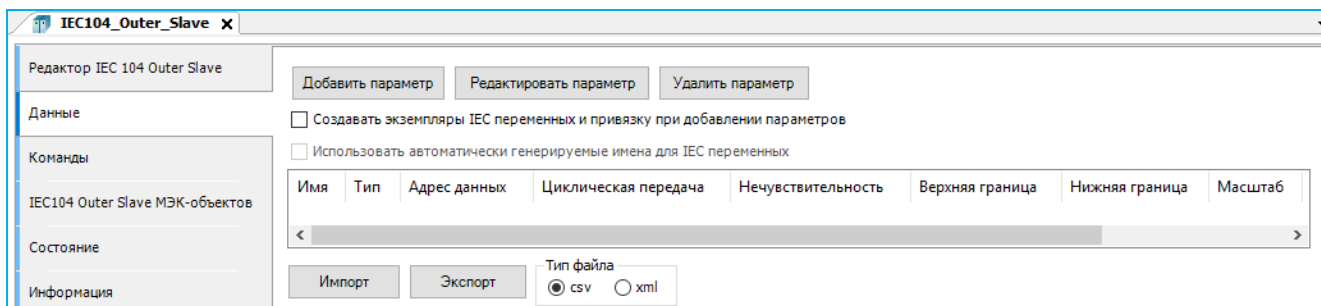


Рисунок 35 – Редактор IEC 104 Outer Slave. Вкладка «Данные»

Элементы данных описываются как параметры. Для добавления, изменения и удаления элементов данных используются соответственно кнопки *Добавить параметр*, *Редактировать параметр*, *Удалить параметр*. Перейти к редактированию параметра также можно двойным щелчком левой кнопкой мыши по нужной строке. Окно ввода/редактирования элемента данных выглядит следующим образом (Рисунок 36).

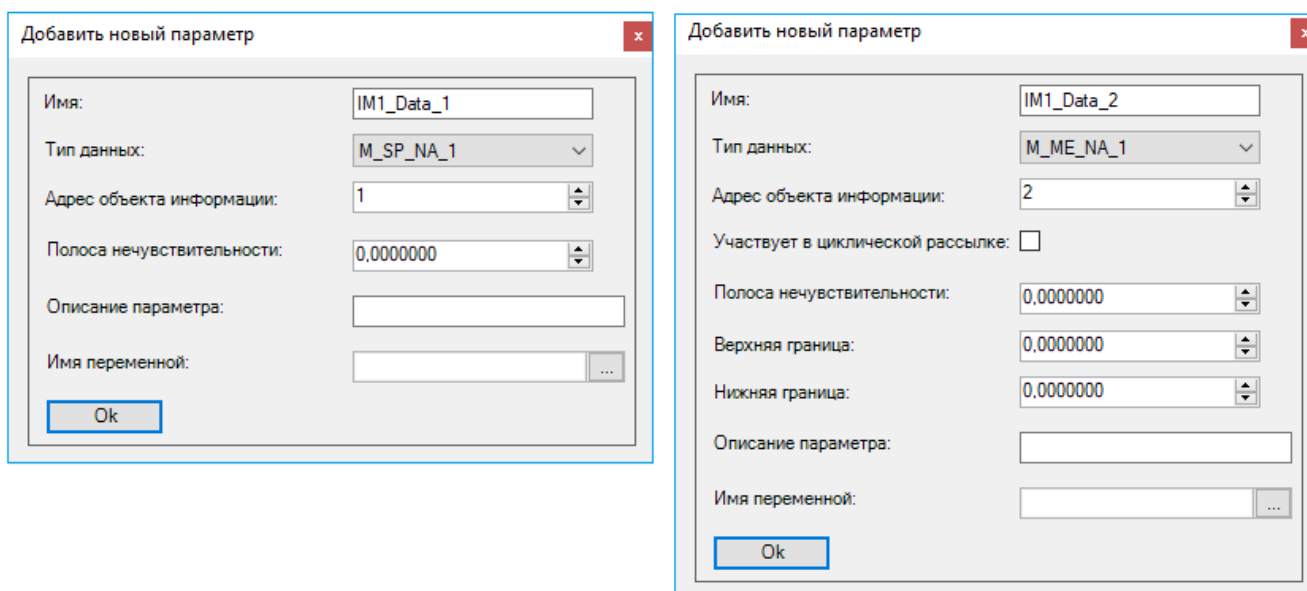


Рисунок 36 – Окно добавления нового параметра (элемента данных)

Для элемента данных укажите значения в полях. Параметры элементов представлены в таблице 17.

Таблица 17 – Параметры элементов данных IEC-104

Параметр	Описание
<b>Имя</b>	Наименование элемента данных
<b>Тип данных</b>	Ассоциированный с этим элементом данных тип IEC-104. Справочные типы данных IEC-104 приведены в таблице 18. Часто используемые типы данных: <ul style="list-style-type: none"> <li>o M_SP_TB_1 – для дискретов с меткой времени;</li> <li>o M_ME_TF_1 – для float-значений с меткой времени;</li> <li>o M_IT_TB_1 – для целочисленных значений</li> </ul>





Параметр	Описание
<b>Адрес объекта информации</b>	<p>Уникальный адрес элемента данных. Значение адреса не должно превышать максимально допустимого значения, которое обуславливается размером адресного поля PDU.</p> <p>Размер адресного поля PDU определяется параметром <b>Длина поля «адрес»</b>.</p> <p>Например, если задано значение длины поля адреса равное 2, то максимально допустимое значение адреса не может превышать 65535 (для значения 3 - не может превышать 16777215).</p> <p>При некорректном вводе, в логе появится сообщение о переполнении адреса, например: «Address overflow: 100001 max: 65535»</p>
<b>Участствует в циклической рассылке</b>	<p>Установка флажка в этом поле активирует участие элемента данных в циклической рассылке. Для всех таких элементов их значения даже при отсутствии изменений будут отсылаться с периодом (в сек.), указанным на вкладке общих параметров</p>
<b>Полоса нечувствительности (deadband)</b>	<p>Устанавливаемая пользователем минимальная разница между последним отосланным значением и текущим измеренным (рассчитанным) в программе контроллера, превышение которой вызывает спонтанную отсылку</p>
<b>Верхняя граница, Нижняя граница</b>	<p>Параметры, задающие диапазон для нормализованных типов (M_ME_NA_1, M_ME_TA_1, M_ME_ND_1, M_ME_TD_1)</p>
<b>Масштаб</b>	<p>Для масштабируемых параметров, при выборе типа M_ME_NB_1, M_ME_TE_1</p>
<b>Описание параметра</b>	<p>Опционально, текстовое описание элемента данных</p>
<b>Имя переменной*</b>	<p>Указывается название переменной IEC (программы контроллера), в которой хранятся передаваемые данные.</p> <p>Для заполнения поля <b>Имя переменной</b> нажмите в этом поле кнопку , открывающую окно <b>Ассистент ввода</b>. Найдите нужную переменную. Если установлен флажок в поле <b>Структурированный вид</b>, то раскрывайте списки с помощью кнопки . Если флажок снят и переменные представлены одним большим списком, для удобства поиска воспользуйтесь фильтром</p>
<p><b>Примечание:</b> -* - параметр присутствует в окне ввода/редактирования элемента данных, когда привязка переменных происходит по адресу (прямой доступ к памяти ПЛК), без использования механизма I/O Mapping (соотнесение входов/выходов), при условии, что не снят флажок с параметра «<b>Не использовать I/O-Mapping</b>»</p>	

Таблица 18 – Перечень типов данных IEC-104

Тип данных	ID типа данных	Описание
M_SP_NA_1	1	Одноэлементная информация
M_DP_NA_1	2	Двухэлементная информация
M_ST_NA_1	5	Информация о положении отпак

Тип данных	ID типа данных	Описание
M_BO_NA_1	7	Строка из 32 бит
M_ME_NA_1	9	Измеряемая величина, нормализованное значение
M_ME_NB_1	11	Измеряемая величина, масштабируемое значение
M_ME_NC_1	13	Измеряемая величина, формат с плавающей запятой
M_IT_NA_1	15	Интегральная сумма
M_ME_ND_1	21	Измеряемая величина, нормализованное значение без описателя качества
M_SP_TB_1	30	Одноэлементная информация с 56-битной меткой времени
M_DP_TB_1	31	Двухэлементная информация с 56-битной меткой времени
M_ST_TB_1	32	Информация о положении отпаяк с 56-битной меткой времени
M_BO_TB_1	33	Строка из 32-х бит с 56-битной меткой времени
M_ME_TD_1	34	Измеряемая величина нормализованное значение с 56 битной меткой времени
M_ME_TE_1	35	Измеряемая величина масштабированное значение с 56 битной меткой времени
M_ME_TF_1	36	Измеряемая величина с плавающей запятой с 56 битной меткой времени
M_IT_TB_1	37	Интегральная сумма с 56-битной меткой времени
M_EP_TD_1	38	Информация о работе релейной защиты с 56-битной меткой времени
M_LR_NA	129*	Информация пользовательского типа LREAL из 64-х бит + байт качества, без метки времени
M_LR_TC	130*	Информация пользовательского типа LREAL из 64-х бит + байт качества с 56-битной меткой времени
M_ULINT_NA	131*	Информация пользовательского типа ULINT из 64-х бит + байт качества, без метки времени
M_ULINT_TC	132*	Информация пользовательского типа ULINT из 64-х бит + байт качества с 56-битной меткой времени
Примечание - * - ID задается пользователем		

Заполненная вкладка **Данные** показана на рисунке 37.

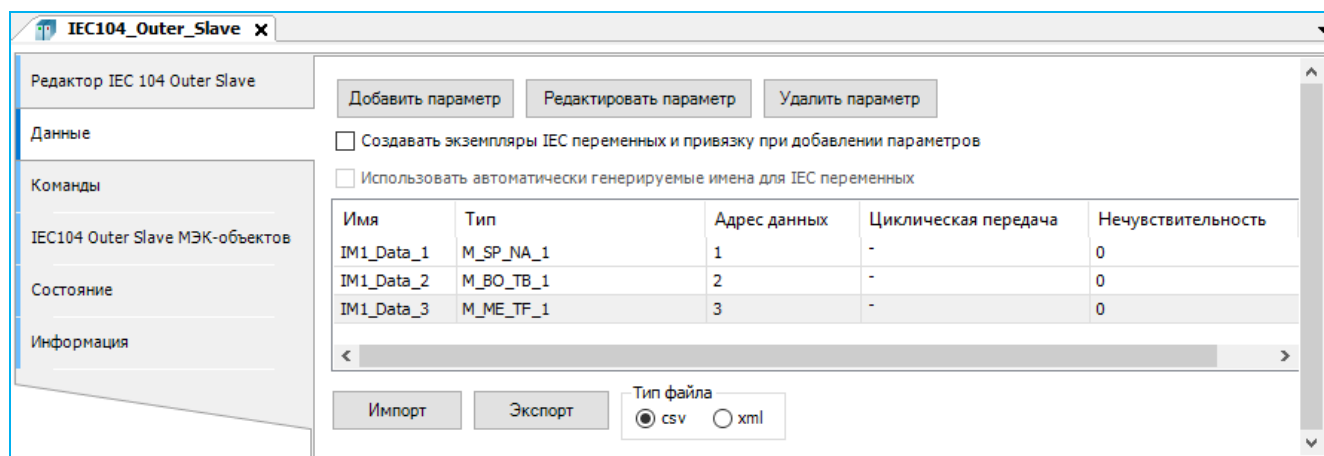


Рисунок 37 – Пример списка элементов данных

### Создание списка команд, передаваемых по IEC-104

Кроме передачи данных по IEC-104 реализованы команды, используемые для установки значения какой-либо переменной, либо выполнения по сигналу каких-либо действий.

Список команд имеет ограничение в 10000 элементов для каждого устройства Slave 104 Outer Slave.

В редакторе устройства **IEC 104 Outer Slave** перейдите на внутреннюю вкладку **Команды**. Общий вид редактора команд аналогичен редактору элементов данных. Команды на данной вкладке описываются как параметры. Для добавления, изменения и удаления команд используются соответственно кнопки *Добавить параметр*, *Редактировать параметр*, *Удалить параметр*. Перейти к редактированию команды также можно двойным щелчком левой кнопкой мыши по нужной строке.

Окно ввода/редактирования команд выглядит следующим образом (Рисунок 38).

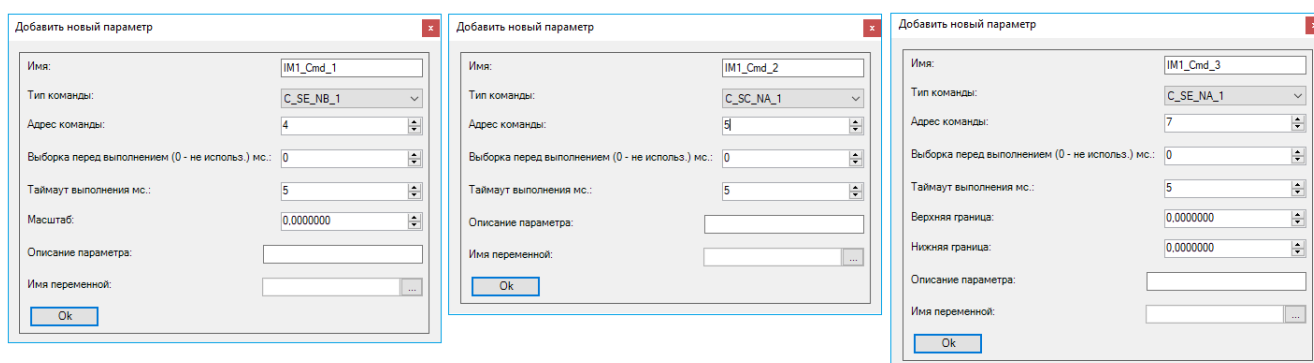


Рисунок 38 – Окно добавления новой команды

Для команды укажите значения в полях. Параметры команд представлены в таблице 19

Таблица 19 – Параметры команд IEC-104

Параметр	Описание
Имя	Наименование команды



Параметр	Описание
<b>Тип команды</b>	Ассоциированный с этой командой тип IEC-104. Справочные типы данных IEC-104 приведены в таблице 20
<b>Адрес команды</b>	<p>Уникальный адрес команды. Значение адреса не должно превышать максимально допустимого значения, которое обуславливается размером адресного поля PDU.</p> <p>Размер адресного поля PDU определяется параметром <b>Длина поля «адрес»</b>.</p> <p>Например, если задана длина поля адреса - 2, то максимально допустимое значение адреса не может превышать 65535.</p> <p>При некорректном вводе, в логе появится сообщение о переполнении адреса команд, например:</p> <p>«Command addr overflow: 100002 max:65535»</p>
<b>Выборка перед выполнением (0 – не использ.) мс.</b>	Команды могут быть простыми и с «выборкой перед выполнением» (Select Before Operate). При ненулевом значении это время, в течение которого команда остается в состоянии «выбрана» для выполнения (после отправки запроса select). Успешное выполнение команды (execute) возможно до истечения данного таймаута
<b>Таймаут выполнения мс.</b>	<p>Определяет время ожидания выполнения команды со стороны управляющего узла. Задается в секундах. Если в течении данного интервала времени не был получен ответ (позитивный или негативный), то управляющий узел полагает, что произошла нештатная ситуация и помечает команду как ошибочную.</p> <p>Параметр exes_timeout актуален только для управляющего узла (Master 104 Driver)</p>
<b>Верхняя граница, Нижняя граница</b>	Пара параметров, определяющих диапазон значений команды для нормализованных команд (C_SE_NA_1 и C_SE_TA_1)
<b>Масштаб</b>	Для команд, задающих масштабируемые значения C_SE_NB_1 и C_SE_TB_1
<b>Описание параметра</b>	Опционально, текстовое описание команды
<b>Имя переменной*</b>	<p>Указывается название переменной IEC (программы контроллера), в которой хранятся передаваемые команды.</p> <p>Для заполнения поля <b>Имя переменной</b> нажмите в этом поле кнопку , открывающую окно <b>Ассистент ввода</b>. Найдите нужную переменную. Если установлен флажок в поле <b>Структурированный вид</b>, то раскрывайте списки с помощью кнопки . Если флажок снят и переменные представлены одним большим списком, для удобства поиска воспользуйтесь фильтром</p>
<p><b>Примечание:</b> *- параметр присутствует в окне ввода/редактирования команды, когда привязка переменных происходит по адресу (прямой доступ к памяти ПЛК), без использования механизма I/O Mapping (соотнесение входов/выходов), при условии, что не снят флажок с параметра «Не использовать IO-Mapping»</p>	

Таблица 20 – Перечень типов данных IEC-104

Тип данных	ID типа данных	Описание
C_SC_NA_1	45	Бинарная команда
C_SE_NA_1	48	Команда уставки (нормализованное значение)
C_SE_NB_1	49	Команда уставки (масштабируемое значение)
C_SE_NC_1	50	Команда уставки (значение типа float)
C_BO_NA_1	51	Команда установки побитового регистра
C_SC_TA_1	58	Бинарная команда с 56-битной меткой времени
C_SE_TC_1	63	Команда уставки (значение типа float) с 56-битной меткой времени
C_BO_TA_1	64	Команда установки побитового регистра с 56-битной меткой времени

### Создание переменных для работы с данными/командами, передаваемыми по IEC-104

Элементы данных и команды, описанные в редакторе IEC-104, в терминологии среды разработки являются **каналами ввода-вывода**. Для реализации обмена данными необходимо создать эти каналы (см. предыдущие разделы), создать в программном коде переменные специальных типов, после чего связать переменные с каналами ввода-вывода.

Требуемые типы переменных (функциональные блоки) описаны в библиотеке PsIECCommon. Эта библиотека, а также использующая ее библиотека PsIoDrvIec104Master автоматически подключаются при добавлении устройства Master 104 Driver.

Для данных используются следующие функциональные блоки: bo\_tb\_fb, ep\_td\_fb, it\_tb\_fb, me\_tf\_fb, sp\_tb\_fb, me\_td\_fb. В наименовании функционального блока фактически указан тип данных IEC-104, например, блоку bo\_tb\_fb соответствует тип M\_BO\_TB\_1, а me\_tf\_fb соответствует M\_ME\_TF\_1. Количество функциональных блоков меньше количества типов данных IEC-104, каждый функциональный блок может использоваться сразу для нескольких типов.

При создании списка элементов данных IEC-104 Outer Slave используются те же самые функциональные блоки, которые описаны в разделе «Настройка контроллера в качестве Slave. Создание переменных для работы с данными/командами, передаваемыми по IEC-104» (таблица 3). Описание этих функциональных блоков и примеры создания переменных также приведены в вышеуказанном разделе.

Предусмотрена возможность программно управлять и получать информацию об уставленном соединении основной/резервный с помощью следующих методов:

- IEC104\_Outer\_Slave.base\_connected() – true, если основное соединение установлено;

- IEC104\_Outer\_Slave.reserv\_connection() – true, если резервное соединение установлено;
- IEC104\_Outer\_Slave.close\_all\_connections() – закрыть основное и резервное соединения;
- IEC104\_Outer\_Slave.close\_base\_connection() – закрыть основное соединение;
- IEC104\_Outer\_Slave.close\_reserv\_connection() – закрыть резервное соединение.

Также предусмотрена возможность получать статус соединения (true, если соединение установлено, иначе – false):

```
linked : BOOL
-----
linked := IEC104_Outer_Slave.Link();
-----
```

Для команд используются следующие функциональные блоки: active\_sc\_fb, active\_se\_nc\_fb, active\_bo\_fb.

Основные свойства функциональных блоков, описывающих команды, следующие:

- **value** – данному свойству присваивается значение, которое будет передаваться командой. Используемый тип значения (имеется в виду простой тип – int, real, bool...) будет зависеть от типа функционального блока. Свойство *value* принимает значение длиной не более 4 байт;
- **m\_timestamp** – в случае, когда команда имеет тип, включающий в себя метку времени, данное поле содержит метку времени, передаваемую в составе команды. Для команд без метки времени данное поле не используется.

### Привязка переменных программы к элементам данных и командам

Начиная с версии СПО 1.5.7.0, чтение/запись привязанных переменных происходит по адресу (прямой доступ к памяти ПЛК), без использования механизма I/O Mapping (соотнесение входов/выходов), при условии, что не снят флажок с параметра «**Не использовать IO-Mapping**» (параметр активирован).

Если IEC устройство содержит параметры старого формата, для которых применяется механизм I/O Mapping (соотнесение входов/выходов), то их можно обновить до нового формата с использованием прямой адресации переменных с помощью кнопки *Обновить параметры* (Рисунок 39). Кнопка появляется на вкладке **Данные** или **Команды** только в том случае, если активирован параметр «**Не использовать IO-Mapping**» и обнаружены параметры старого формата. Такие параметры расположены в другом диапазоне ID.

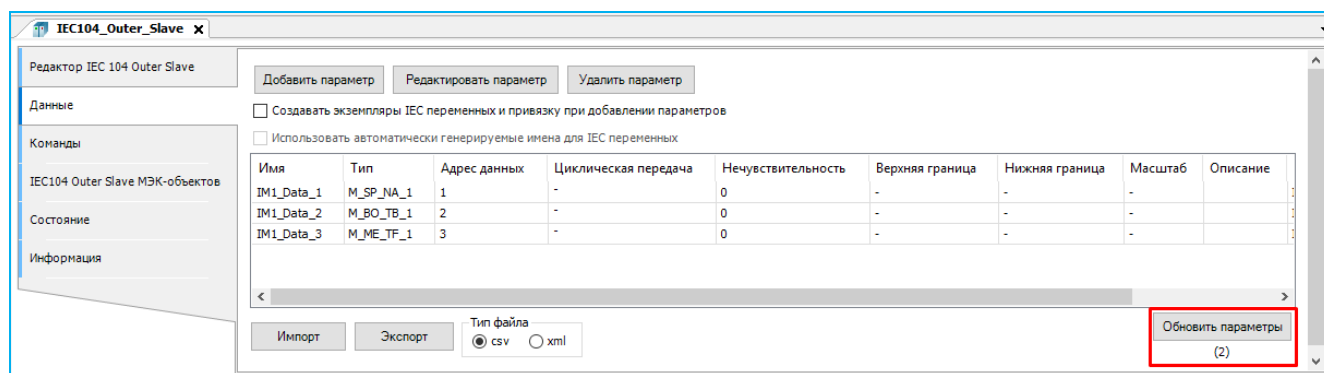


Рисунок 39 – Обновление параметров до нового формата

При активированном параметре «**Не использовать IO-Mapping**» в редакторе устройства **IEC 104 Outer Slave** будет отсутствовать внутренняя вкладка **Соотнесение входов/выходов**.



**ВНИМАНИЕ!**

Рекомендуется не деактивировать параметр «**Не использовать IO-Mapping**»

Если деактивировать параметр (снять флажок), необходимо будет закрыть основную вкладку устройства **IEC 104 Outer Slave**, а затем заново открыть и при добавлении данных/команд автоматически появится вкладка **Соотнесение входов/выходов** (Рисунок 40).

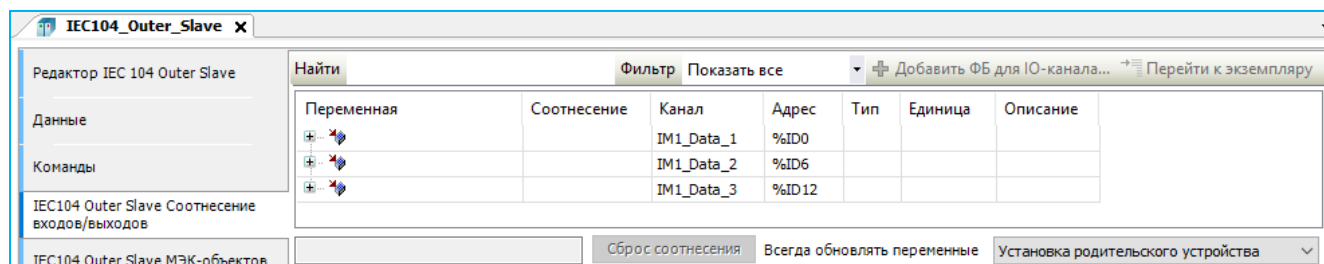


Рисунок 40 – IEC 104 Outer Slave Соотнесение входов/выходов

На этой вкладке представлен список каналов ввода-вывода (в терминологии среды разработки), ассоциированных с устройством. Когда пользователь создает элемент данных или команду в соответствующем редакторе (вкладка **Данные**, вкладка **Команды**), этот параметр автоматически появляется здесь в виде канала вывода (для данных) или канала ввода (для команд). Наименование добавленного параметра указано в колонке **Канал**.

Для того, чтобы значения, приходящие/отправляемые по каналу, были доступны в программе контроллера, нужно привязать канал к переменной программы. Дважды щелкните левой кнопкой мыши в строке нужного канала. Появится курсор (можно вручную ввести имя переменной, семантика имен описана ниже) и кнопка **...**, открывающая окно **Ассистент ввода** (Рисунок 41).

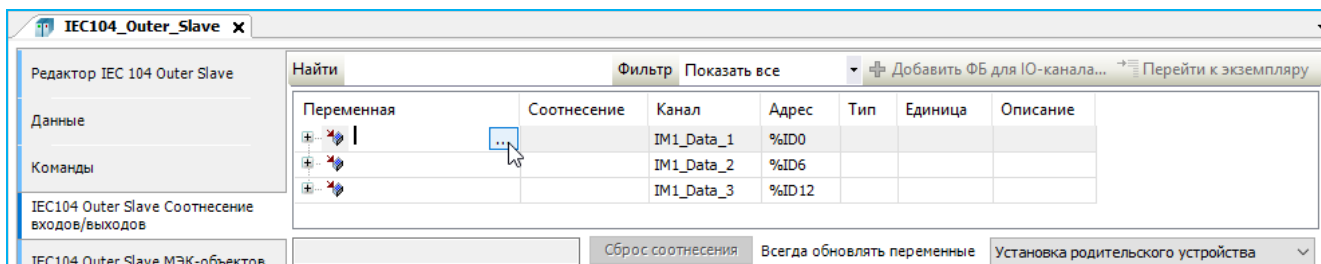


Рисунок 41 – Ручной ввод переменной или вызов ассистента ввода

В окне **Ассистент ввода** (Рисунок 42) найдите нужную переменную. Если установлен флажок в поле **Структурированный вид**, то раскрывайте списки с помощью кнопки **+**. Если флажок снят и переменные представлены одним большим списком, для удобства поиска воспользуйтесь фильтром.

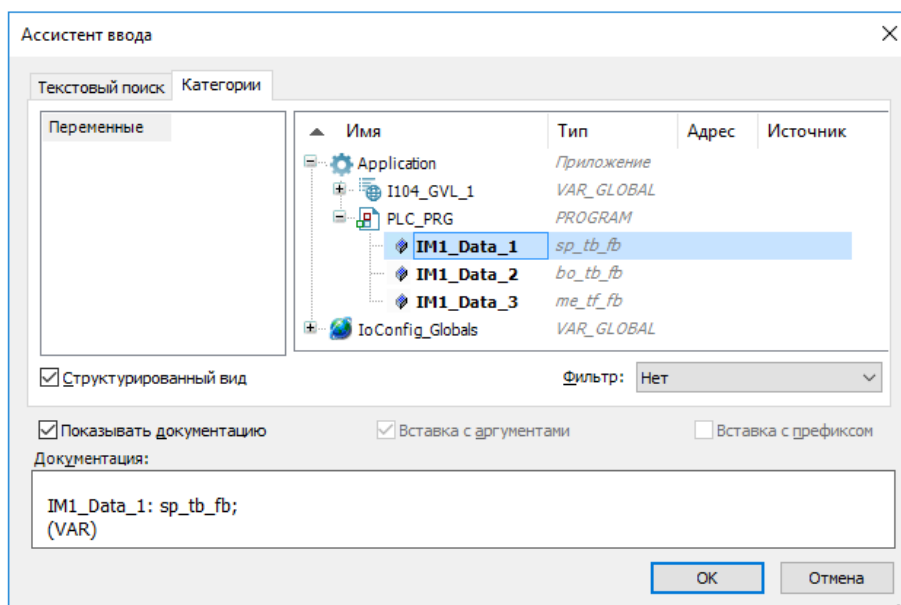


Рисунок 42 – Диалоговое окно «Ассистент ввода»

После выбора переменной нажмите кнопку **OK**, закроется окно **Ассистент ввода**, а переменная появится на вкладке **Соотнесение входов/выходов** (Рисунок 43).

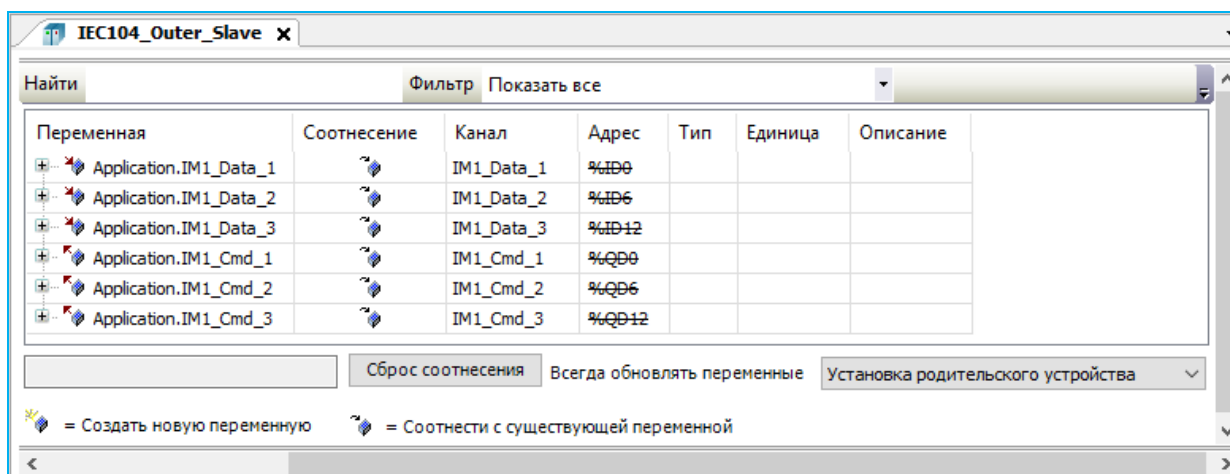


Рисунок 43 – Переменные привязаны к каналам вывода





### ИНФОРМАЦИЯ

Допускается использование одних и тех же экземпляров переменных (и данных, и команд) для привязки к разным экземплярам драйверов (Master/Slave, 101/104), при условии, что настройки привязанной переменной (параметры элементов данных/команд) для всех экземпляров драйверов идентичны.

На этапе компиляции проекта выполняется проверка совпадения настроек, в случае несовпадения, в окно сообщений выводится сообщение об ошибке, с указанием имени переменной и имен соответствующих экземпляров драйверов.

Запрещено привязывать переменную данных одновременно во вкладке **Данные** и **Данные для приема**

При ручном вводе семантика имен следующая (угловые скобки при вводе НЕ используются):  
*<Имя приложения>.<Имя программного юнита>.<Имя переменной>*

Как ранее указано, для привязки к каналу IEC-104 используются только переменные, представляющие из себя функциональные блоки определенных типов, описанных в библиотеке PsIecCommon.



### ИНФОРМАЦИЯ

Тип данных IEC-104 (например, M\_BO\_NA\_1) задается при описании элемента данных или команды. После «привязки» переменной приложения к соответствующему каналу, переменная (типа bo\_tb\_fb) будет передаваться по протоколу IEC-60870-5-104(101) с использованием типа M\_BO\_NA\_1

На этапе компиляции производится проверка типов привязанных переменных. При несоответствии типов, в окне сообщений появится информация об ошибке (Рисунок 44).

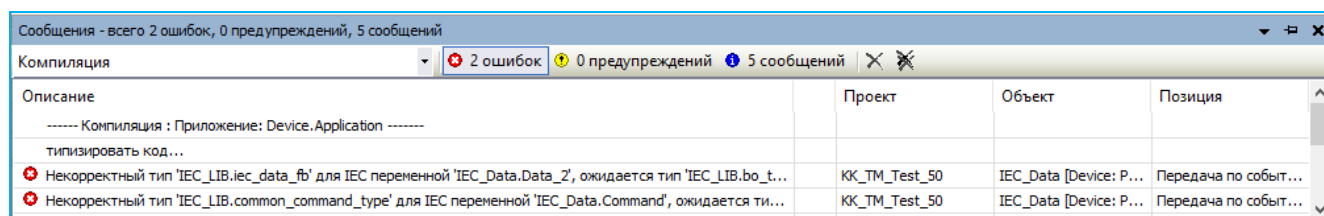


Рисунок 44 – Пример. Сообщение об ошибке (Master 104)

Все функциональные блоки, используемые для описания переменных IEC-104, имеют поле метки времени. Но в случаях, когда тип данных IEC-104, указанный в описании элемента данных или команды, не имеет метки времени (например, M\_BO\_NA\_1), поле **Метка времени** функционального блока просто не используется. Также в зависимости от типа данных IEC-104 поле **Метка времени** может сериализоваться в 56-битный или в 24-битный timestamp.

#### Установка резервных бит в метке времени

Имеется возможность передачи резервных битов в timestamp (IEC 60870-5-4 п 6.8 COMPOUND INFORMATION ELEMENTS CP56Time2a). Это осуществляется путем установки старших битов в компоненты переменной типа IEC\_LIB.timestamp\_type.

Octet 3 биты IV, Res1 можно установить с помощью  
 ts.m\_minute := ts.m\_minute OR 16#C000; // установлены 2 бита

Octet 4 биты SU, RES2 можно установить с помощью  
 ts.m\_hour := ts.m\_hour OR 16#E000; // установлены 3 бита

Octet 5 биты Day of week можно установить с помощью  
 ts.m\_day := ts.m\_day OR 16#E000; // установлены 3 бита

Octet 6 биты RES3 можно установить с помощью  
 ts.m\_month := ts.m\_month OR 16#F000; // установлены 4 бита

Octet 7 бит RES1 можно установить с помощью  
 ts.m\_year := ts.m\_year OR 16#8000; // установлен 1 бит.

### Автоматическая генерация переменных и привязка к каналам ввода-вывода

При добавлении параметров на вкладках **Данные** и **Команды** можно активировать режим, при котором для каждого канала автоматически создаются переменные ПЛК-программы и формируется привязка переменной к каналу. Для этого при создании нового элемента данных или команды установите флажок в поле **Создавать экземпляры IEC переменных и привязку при добавлении параметров** (Рисунок 45).

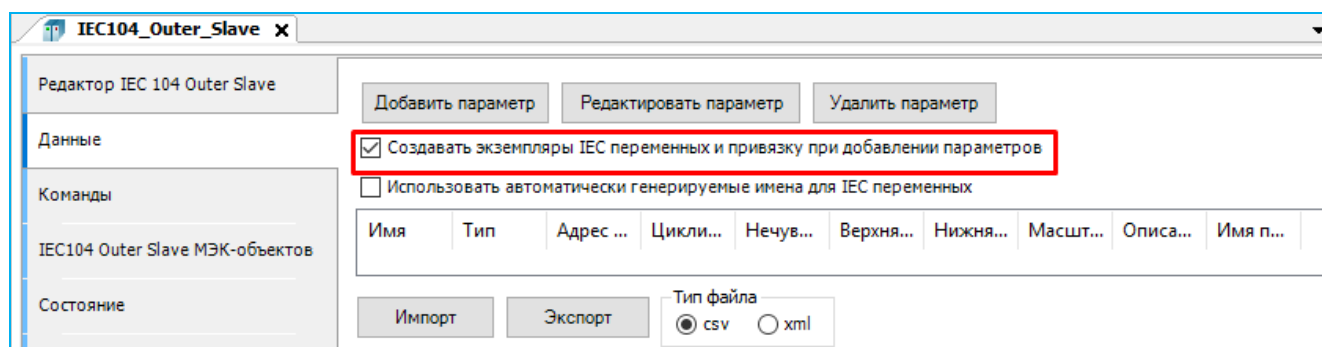


Рисунок 45 – Установка флажка в поле «Создавать экземпляры IEC переменных и привязку при добавлении параметров»

В этом режиме при добавлении канала для устройства **IEC 104 Outer Slave** создается собственный список глобальных переменных с именем вида I104\_GVL\_X, где X – условный порядковый номер устройства, начиная с единицы (Рисунок 46). В этом списке будет создана переменная соответствующего типа.

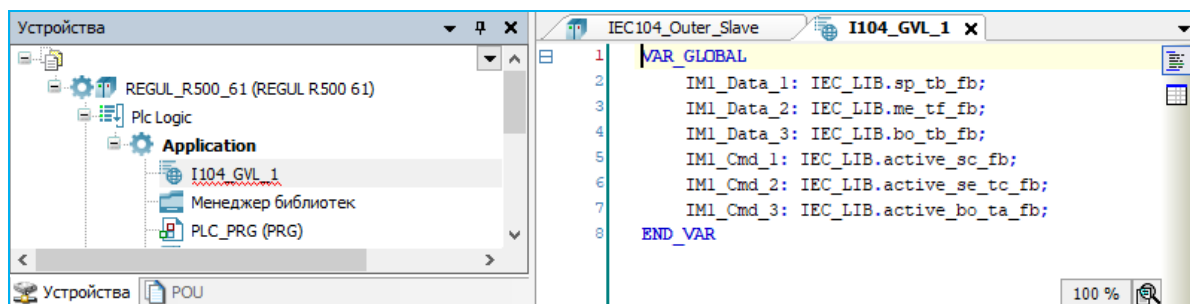


Рисунок 46 – Пример описания переменных в списке I104\_GVL\_1

При наличии вкладки **Соотнесение входов/выходов** новая переменная будет автоматически привязана к новому каналу (Рисунок 47). Наименование переменной будет совпадать с именем канала (пробелы заменяются на подчеркивания).

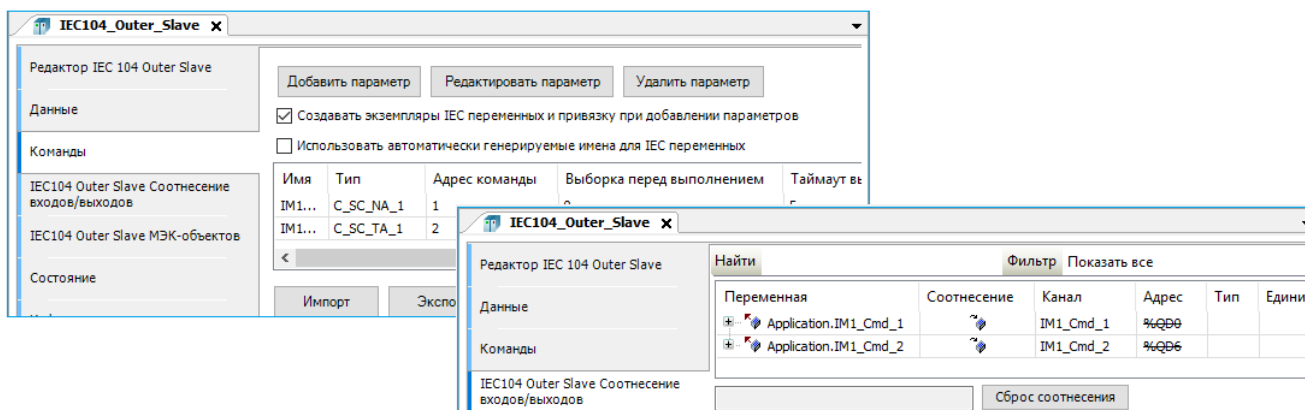


Рисунок 47 – Автоматическое создание переменной и автоматическая привязка ее к каналу

При установленном флажке в поле **Использовать автоматически генерируемые имена для IEC переменных** имена переменных будут создаваться на основе шаблона: *iec<случайное число>\_data\_fb*. При отсутствии флажка в этом поле имена генерируемых переменных имеют вид: *IS<X>\_Data\_<N>* (для элементов данных), *IS<X>\_Cmd\_<N>* (для команд), где X – номер Slave-устройства, N – нумератор переменных. Генерируемые по умолчанию имена каналов данных и команд совпадают с именами связываемых переменных.

### Возможность разрыва соединения со стороны IEC-программы

Для контроллера в качестве Master добавлена функциональность, позволяющая IEC-приложению управлять разрывом соединения. Когда программа определяет, что обмен данными по протоколу IEC-104 осуществляется некорректно, выполняется команда, которая разрывает основное соединение и происходит переключение на резервный канал. Попытка соединения по основному каналу состоится через некоторый промежуток времени (*reconnect\_interval*). Если она будет успешной, то резервное соединение закроется и обмен продолжится по основному каналу.

The screenshot displays a PLC programming environment with two main sections: a variable declaration table at the top and a ladder logic program below it.

Expression	Type	Value	Prepared value
connect	BOOL	TRUE	
velocity	DWORD	0	
var1	REAL	0	
addr_exposed	BOOL	FALSE	
dw	DWORD	0	
str	STRING	*	

```

1 me_tf_1;
2 sp_na_1;
3 it_tb_1;
4 c_sc;
5
6 IF ( do_disconnect FALSE )
7 THEN
8   do_disconnect FALSE := FALSE;
9   IEC104_Outer_Slave.m_base_connection.deferred_disconnect();
10 END_IF
11
12 b TRUE := IEC104_Outer_Slave.base_connected();
13 re FALSE := IEC104_Outer_Slave.m_reserv_connection.connected();
14
15 IF ( do_command FALSE )
16 THEN
17   c_sc.value := command_value 0;
18   do_command FALSE := FALSE;
19 END_IF
    
```

The Watch window below the code shows the following data:

Expression	Application	Type	Value	Prepared value	Execution point
IEC104_Outer_Slave.m_base_connection.m_state	REGUL_R500_51_61.Application_use_master104	CONNECTION_104_STATE_E	ready_for_data		Cyclic Monitoring
IEC104_Outer_Slave.m_reserv_connection.m_state	REGUL_R500_51_61.Application_use_master104	CONNECTION_104_STATE_E	disconnected		Cyclic Monitoring

Рисунок 48 – Разрыв соединения со стороны программы

## Настройка контроллера в качестве Master 104 OS

Добавление устройства производится идентично тому, как описано в разделе «Настройка контроллера в качестве Master 104» (Драйвер Master 104 OS).

### Настройка общих параметров устройства Outer Slave 104

В редакторе устройства **Outer Slave 104** (Рисунок 49) в блоке **Общие параметры устройства** доступны для настройки параметры, представленные в таблице 21.

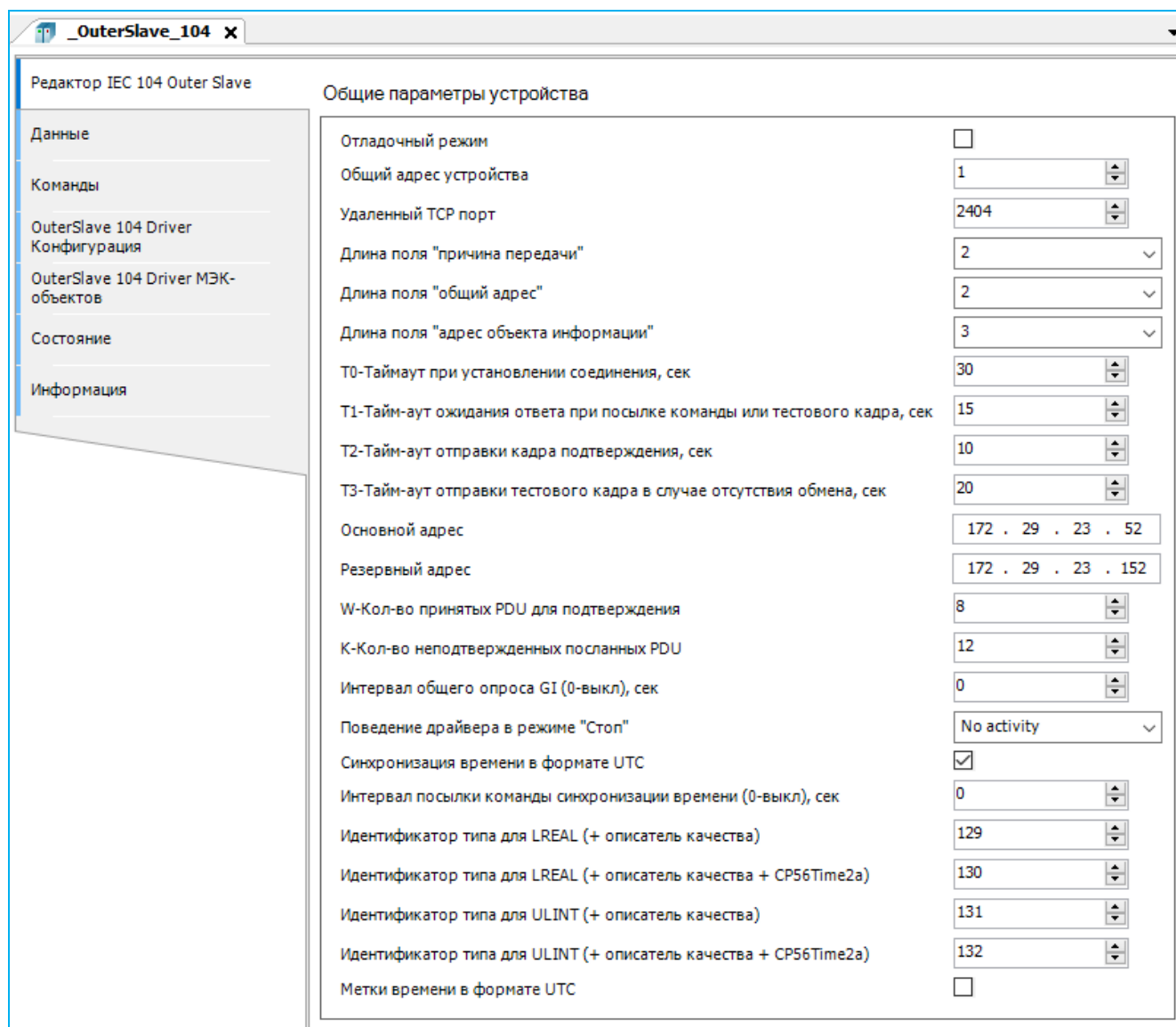


Рисунок 49 – Редактор устройства Outer Slave 104 (OS)

Таблица 21 – Общие параметры устройства

Параметр	Описание	Значение по умолчанию
<b>Отладочный режим</b>	При установке флажка <input checked="" type="checkbox"/> в этом поле в журнал работы контроллера будут выводиться сообщения об ошибках, предупреждениях и ключевых моментах работы драйвера.	

Параметр	Описание	Значение по умолчанию
<b>Общий адрес устройства</b>	Содержит общий адрес (COMMON ADDRESS, ASDU ADDRESS) устройства, все отдаваемые данные и выполняемые команды имеют общую часть, равную значению этого параметра (см. спецификацию IEC-104)	1
<b>Удаленный TCP порт</b>	Номер TCP порта слэйва IEC-104 для подключения мастера	2404
<b>Длина поля «причина передачи»</b>	Содержит длину поля «причина передачи» (COT, Cause Of Transmission). Допустимые значения параметра: 1 или 2	2
<b>Длина поля «общий адрес»</b>	Содержит длину поля «общий адрес» (CA, Common Address). Допустимые значения параметра: 1 или 2	2
<b>Длина поля «адрес объекта информации»</b>	Содержит длину поля «адрес объекта информации» (IOA, Information Object Address). Допустимые значения параметра: 2 или 3	3
<b>T0-Тайм-аут при установлении соединения, сек</b>	Время, в течение которого мастер пытается установить соединение	30
<b>T1-Тайм-аут ожидания ответа при посылке команды или тестового кадра, сек</b>	Время, в течение которого ожидается ответ на команду или подтверждение доставки PDU. Превышение таймаута приводит к разрыву соединения	15
<b>T2-Тайм-аут отправки кадра подтверждения, сек</b>	При получении PDU запускается таймер на T2 секунд, подтверждение о приеме (пакет S-PDU) отсылается либо по истечению этого таймера, либо по факту приема некоторого предельного количества PDU, указанного в параметре W. T2 должен быть меньше T1	10
<b>T3-Тайм-аут отправки тестового кадра в случае отсутствия обмена, сек</b>	Периодически как master, так и slave могут отсылать специальные пакеты для проверки связи.	20
<b>Основной адрес</b>	IP-адрес основного соединения. При обрыве связи переключение на резервное соединение произойдет не раньше, чем через T1 при наличии активного обмена и через T3 при отсутствии обмена.	
<b>Резервный адрес</b>	IP-адрес резервного соединения. При обрыве связи переключение на резервное соединение произойдет не раньше, чем через T1 при наличии активного обмена и через T3 при отсутствии обмена.	
<b>W-Количество принятых PDU для подтверждения</b>	Количество принятых PDU, которое необходимо подтвердить. Рекомендуем выбирать значение W, не превышающее 2/3 от значения K (Master). Параметры должны быть согласованы на Master и Slave. Например, при K=10 на Master, W=8 на Slave	8

Параметр	Описание	Значение по умолчанию
<b>К-Количество неподтвержденных посланных PDU</b>	Количество неподтвержденных отправленных PDU, после которого драйвер прекращает отправлять новые I-пакеты, дожидаясь подтверждения отправленных ранее.  Следует отметить, что в случае, когда требуется отправка большого количества данных, значение параметра К следует устанавливать большим. Например, если требуется отправка 20000 отсчетов в секунду, значение К нужно устанавливать от 1000 и больше. Иначе драйвер будет приостанавливать отправки данных, ожидая подтверждения клиента, размеры очередей при этом будут нарастать	12
<b>Интервал общего опроса GI (0-выкл), сек</b>	Период отсылки команды общего опроса. Значение 0 – отсылка команды общего опроса не производится	0
<b>Поведение драйвера в режиме «Стоп» *</b>	Определяет поведение компонента при остановке программы.  <i>Нет активности (No activity)</i> - означает, что устройство Master закрывает все соединения  <i>Нормальная работа(Normal work)</i> - означает продолжение работы в обычном режиме	No activity
<b>Синхронизация времени в формате UTC</b>	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что при синхронизации, временная метка в составе команды будет в формате UTC, иначе - локального времени	
<b>Интервал посылки команды синхронизации времени (0-выкл), сек</b>	Мастер может посылать команду синхронизации времени с заданным периодом	0
<b>Идентификатор типа для LREAL (+описатель качества)</b>	Задание ID для пользовательского типа LREAL(double), без метки времени	129
<b>Идентификатор типа для LREAL (+описатель качества +CP56Time2a)</b>	Задание ID для пользовательского типа LREAL(double), с меткой времени	130
<b>Идентификатор типа для ULINT (+описатель качества)</b>	Задание ID для пользовательского типа ULINT(uint64), без метки времени	131
<b>Идентификатор типа для ULINT (+описатель качества +CP56Time2a)</b>	Задание ID для пользовательского типа ULINT(uint64), с меткой времени	132
<b>Метки времени в</b>	Если пользователь не задал метку времени перед отправкой	

Параметр	Описание	Значение по умолчанию
<b>формат UTC</b>	команды, то она автоматически формируется с заданным форматом	
<p><b>Примечания</b></p> <p>*- предусмотрена возможность самостоятельно активировать «поведение в режиме «Стоп» в программном коде. Для активации режима требуется в программе присвоить значение TRUE свойству ActivateStopBehaviour необходимого устройства:</p> <p>«IEC104MasterOS_device_name».ActivateStopBehaviour := TRUE;</p> <p>После этого master-устройство перейдет в STOP-режим работы</p>		

### Создание списка элементов данных, передаваемых по IEC-104

В редакторе устройства **Драйвер OuterSlave 104** перейдите на внутреннюю вкладку **Данные** (Рисунок 50).

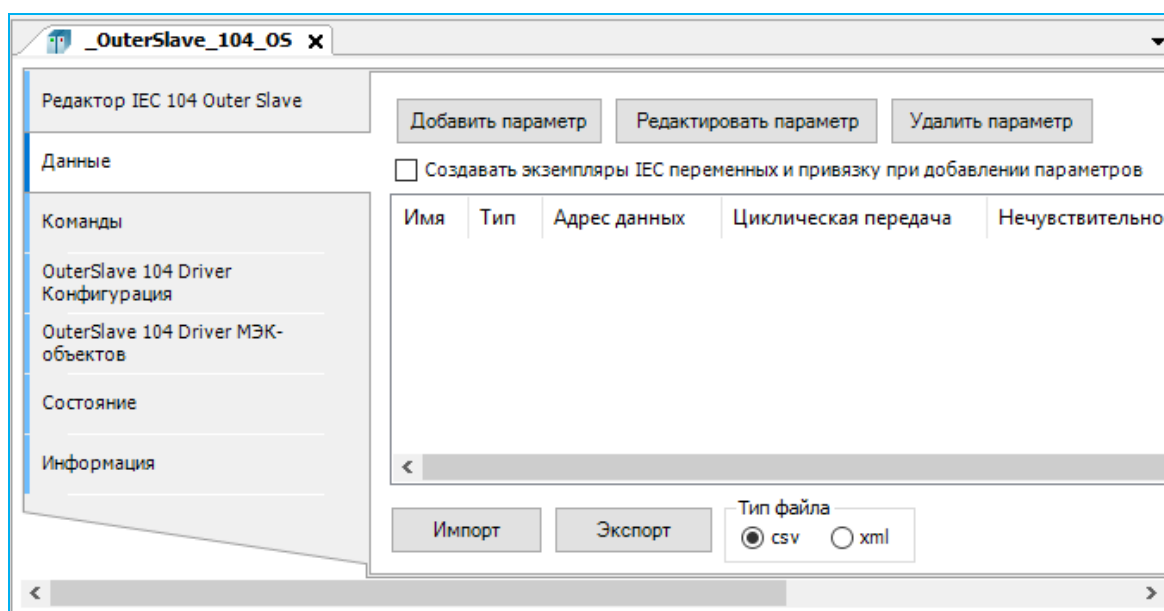


Рисунок 50 – Редактор драйвера Outer Slave 104 (OS). Вкладка «Данные»

Элементы данных описываются как параметры. Для добавления, изменения и удаления элементов данных используются соответственно кнопки **Добавить параметр**, **Редактировать параметр**, **Удалить параметр**. Перейти к редактированию параметра также можно двойным щелчком левой кнопкой мыши по нужной строке. Окно ввода/редактирования элемента данных выглядит следующим образом (Рисунок 51).



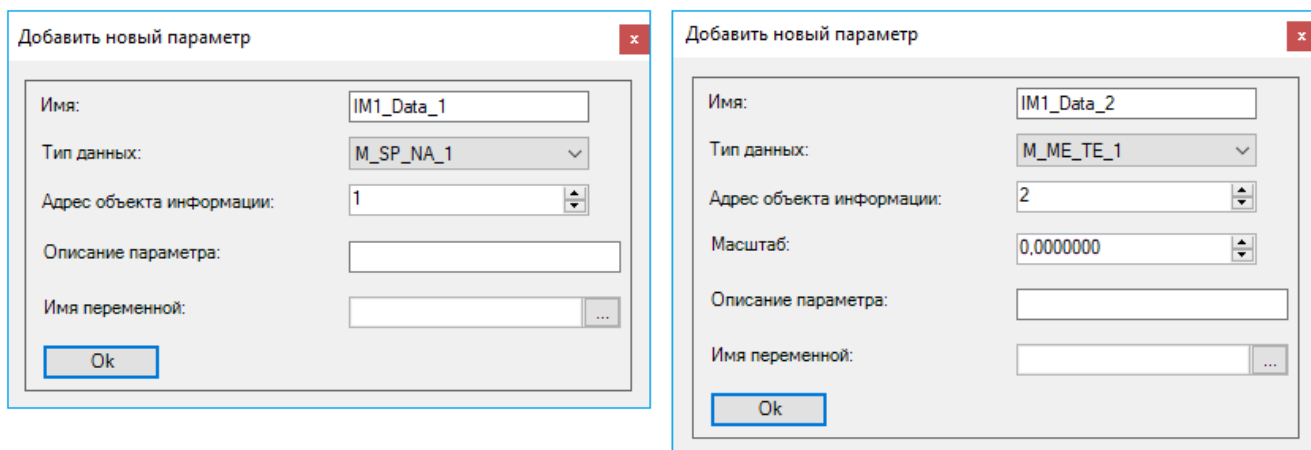




Рисунок 51 – Окно добавления нового параметра (элемента данных)

Для элемента данных укажите значения в полях. Параметры элементов представлены в таблице 22.

Таблица 22 – Параметры элементов данных IEC-104

Параметр	Описание
<b>Имя</b>	Наименование элемента данных
<b>Тип данных</b>	Ассоциированный с этим элементом данных тип IEC-104. Справочные типы данных IEC-104 приведены в таблице 23. Часто используемые типы данных: M_SP_TV_1 – для дискретов с меткой времени; M_ME_TF_1 – для float-значений с меткой времени; M_IT_TV_1 – для целочисленных значений
<b>Адрес объекта информации</b>	Уникальный адрес элемента данных. Уникальный адрес элемента данных. Значение адреса не должно превышать максимально допустимого значения, которое обуславливается размером адресного поля PDU. Размер адресного поля PDU определяется параметром <b>Длина поля «адрес»</b> . Например, если задано значение длины поля адреса равное 2, то максимально допустимое значение адреса не может превышать 65535 (для значения 3 - не может превышать 16777215)
<b>Масштаб</b>	Для масштабируемых параметров, при выборе типа M_ME_NB_1, M_ME_TE_1
<b>Описание параметра</b>	Опционально, текстовое описание элемента данных
<b>Имя переменной</b>	Указывается название переменной IEC (программы контроллера), в которой хранятся передаваемые данные. Для заполнения поля <b>Имя переменной</b> нажмите в этом поле кнопку  , открывающую окно <b>Ассистент ввода</b> . Найдите нужную переменную. Если установлен флажок в поле <b>Структурированный вид</b> , то раскрывайте списки с помощью кнопки  . Если флажок снят и переменные представлены одним большим списком, для удобства

Параметр	Описание
	поиска воспользуйтесь фильтром

Таблица 23 – Перечень типов данных IEC-104

Тип данных	ID типа данных	Описание
M_SP_NA_1	1	Одноэлементная информация
M_DP_NA_1	2	Двухэлементная информация
M_ST_NA_1	5	Информация о положении отпак
M_BO_NA_1	7	Строка из 32 бит
M_ME_NA_1	9	Измеряемая величина, нормализованное значение
M_ME_NB_1	11	Измеряемая величина, масштабируемое значение
M_ME_NC_1	13	Измеряемая величина, формат с плавающей запятой
M_IT_NA_1	15	Интегральная сумма
M_ME_ND_1	21	Измеряемая величина, нормализованное значение без описателя качества
M_SP_TB_1	30	Одноэлементная информация с 56-битной меткой времени
M_DP_TB_1	31	Двухэлементная информация с 56-битной меткой времени
M_ST_TB_1	32	Информация о положении отпак с 56-битной меткой времени
M_BO_TB_1	33	Строка из 32-х бит с 56-битной меткой времени
M_ME_TD_1	34	Измеряемая величина нормализованное значение с 56 битной меткой времени
M_ME_TE_1	35	Измеряемая величина масштабированное значение с 56 битной меткой времени
M_ME_TF_1	36	Измеряемая величина с плавающей запятой с 56 битной меткой времени
M_IT_TB_1	37	Интегральная сумма с 56-битной меткой времени
M_EP_TD_1	38	Информация о работе релейной защиты с 56-битной меткой времени
M_LREAL_NA	129*	Информация пользовательского типа LREAL из 64-х бит + байт качества, без метки времени
M_LREAL_TC	130*	Информация пользовательского типа LREAL из 64-х бит + байт качества с 56-битной меткой времени
M_ULINT_NA	131*	Информация пользовательского типа ULINT из 64-х бит + байт качества, без метки времени
M_ULINT_TC	132*	Информация пользовательского типа ULINT из 64-х бит + байт качества с 56-битной меткой времени
<b>Примечание</b> - * - ID задается пользователем		

## Создание списка команд, передаваемых по IEC-104

Кроме передачи данных по IEC-104 реализованы команды, используемые для установки значения какой-либо переменной, либо выполнения по сигналу каких-либо действий.

В редакторе устройства **Драйвер OuterSlave 104** перейдите на внутреннюю вкладку **Команды**. Общий вид редактора команд аналогичен редактору элементов данных. Команды на данной вкладке описываются как параметры. Для добавления, изменения и удаления команд используются соответственно кнопки *Добавить параметр*, *Редактировать параметр*, *Удалить параметр*. Перейти к редактированию команды также можно двойным щелчком левой кнопкой мыши по нужной строке.

Окно ввода/редактирования команд выглядит следующим образом (Рисунок 52).

Рисунок 52 – Окно редактирования команды

Для команды укажите значения в полях. Параметры команд представлены в таблице 24.

Таблица 24 – Параметры команд IEC-104 (OS)

Параметр	Описание
<b>Имя</b>	Наименование команды
<b>Тип данных</b>	Ассоциированный с этой командой тип IEC-104. Справочные типы данных IEC-104 приведены в таблице 25
<b>Адрес команды</b>	Уникальный адрес команды. Значение адреса не должно превышать максимально допустимого значения, которое обуславливается размером адресного поля PDU. Размер адресного поля PDU определяется параметром <b>Длина поля «адрес»</b> . Например, если задана длина поля адреса - 2, то максимально допустимое значение адреса не может превышать 65535
<b>Выборка перед выполнением (0 – не</b>	Команды могут быть простыми и с «выборкой перед выполнением» (Select Before Operate). При ненулевом значении это время, в течение



Параметр	Описание
использ.) мс	которого команда остается в состоянии «выбрана» для выполнения (после отправки запроса select). Успешное выполнение команды (execute) возможно до истечения данного таймаута
Таймаут выполнения мс	Определяет время ожидания выполнения команды со стороны управляющего узла. Задается в миллисекундах. Если в течении данного интервала времени не был получен ответ (позитивный или негативный), то управляющий узел полагает, что произошла ошибка
Масштаб	Для команд, задающих масштабируемые значения C_SE_NB_1 и C_SE_TB_1
Описание параметра	Опционально, текстовое описание команды
Имя переменной	Указывается название переменной IEC (программы контроллера), в которой хранятся передаваемые команды.  Для заполнения поля <b>Имя переменной</b> нажмите в этом поле кнопку  , открывающую окно <b>Ассистент ввода</b> . Найдите нужную переменную. Если установлен флажок в поле <b>Структурированный вид</b> , то раскрывайте списки с помощью кнопки  . Если флажок снят и переменные представлены одним большим списком, для удобства поиска воспользуйтесь фильтром

Таблица 25 – Перечень типов команд IEC-104

Тип данных	ID типа данных	Описание
C_SC_NA_1	45	Бинарная команда
C_DC_NA_1	46	Команда с тремя состояниями
C_RC_NA_1	47	Регулирующая пошаговая команда
C_SE_NA_1	48	Команда уставки (нормализованное значение)
C_SE_NB_1	49	Команда уставки (масштабируемое значение)
C_SE_NC_1	50	Команда уставки (значение типа float)
C_BO_NA_1	51	Команда установки побитового регистра
C_SC_TA_1	58	Бинарная команда с 56-битной меткой времени
C_DC_TA_1	59	Команда с тремя состояниями с 56-битной меткой времени
C_RC_TA_1	60	Регулирующая пошаговая команда с 56-битной меткой времени
C_SE_TA_1	61	Команда уставки (нормализованное значение) с 56-битной меткой времени
C_SE_TB_1	62	Команда уставки (масштабируемое значение) с 56-битной меткой времени
C_SE_TC_1	63	Команда уставки (значение типа float) с 56-битной меткой времени

C_BO_TA_1	64	Команда установки побитового регистра с 56-битной меткой времени
-----------	----	--

### Создание переменных для работы с данными/командами, передаваемыми по IEC-104

Элементы данных и команды, описанные в редакторе IEC-104, в терминологии среды разработки являются **каналами ввода-вывода**. Для реализации обмена данными необходимо создать эти каналы (см. предыдущие разделы), создать в программном коде переменные специальных типов, после чего связать переменные с каналами ввода-вывода.

Требуемые типы переменных (функциональные блоки) описаны в общей библиотеке **PsIEC60870Bridge**. Эта библиотека, а также использующая ее библиотека **PsIoDrvIec104Master\_OS**, автоматически подключаются при добавлении устройства Драйвер Master 104 OS / Драйвер OuterSlave 104.

В таблице 26 дано описание поддерживаемых типов данных IEC-104 и соответствующих им функциональных блоков с указанием главы стандарта IEC 60870-5-101 для этого типа и перечислением полей функционального блока для работы с его значением, качеством и меткой времени.

Таблица 26 – Соответствие функциональных блоков и типов данных/команд IEC-104

№	Функц.блок	Тип IEC-104	Применение	Поля функционального блока
1	m_sp_na_1_fb	M_SP_NA_1 (IEC 60870-5-101, 7.2.6.1)	Дискрет: 0, 1	Значение (BYTE): SPI, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB
2	m_sp_tb_1_fb	M_SP_TB_1 (IEC 60870-5-101, 7.3.1.22)	Дискрет: 0, 1 с временной меткой CP56Time2a	Наследует от m_sp_na_1_fb Время: timestamp (psCP56Time2a_t)
3	m_dp_na_1_fb	M_DP_NA_1 (IEC 60870-5-101, 7.3.1.3)	2-ух-битовое поле: 0-3	Значение (BYTE): DPI, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB
4	m_dp_tb_1_fb	M_DP_TB_1 (IEC 60870-5-101, 7.3.1.23)	2-ух-битовое поле: 0-3 с временной меткой CP56Time2a	Наследует от m_dp_na_1_fb Время: timestamp (psCP56Time2a_t)
5	m_st_na_1_fb	M_ST_NA_1 (IEC 60870-5-101, 7.3.1.5)	7-битовое значение (-63 .. +64) с битом переходного состояния (0,1)	Значение (BYTE): VTI, value, TRANS Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV

№	Функц.блок	Тип IEC-104	Применение	Поля функционального блока
6	m_st_tb_1_fb	M_ST_TB_1 (IEC 60870-5-101, 7.3.1.24)	7-битовое значение (-63 - +64) с битом переходного состояния (0,1) с временной меткой CP56Time2a	Наследует от m_st_na_1_fb Время: timestamp (psCP56Time2a_t)
7	m_bo_na_1_fb	M_BO_NA_1 (IEC 60870-5-101, 7.3.1.7)	4-байтное целое без знака (32-битный BITSTRING)	Значение (DWORD): BSI, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
8	m_bo_tb_1_fb	M_BO_TB_1 (IEC 60870-5-101, 7.3.1.25)	4-байтное целое без знака (32-битный BITSTRING) с временной меткой CP56Time2a	Наследует от m_bo_na_1_fb Время: timestamp (psCP56Time2a_t)
9	m_me_na_1_fb	M_ME_NA_1 (IEC 60870-5-101, 7.3.1.9)	Нормализованное число с плавающей точкой: $-1 \dots +1 \cdot 2^{-15}$	Значение (REAL): NVA, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
10	m_me_td_1_fb	M_ME_TD_1 (IEC 60870-5-101, 7.3.1.26)	Нормализованное число с плавающей точкой: $-1 \dots +1 \cdot 2^{-15}$ с временной меткой CP56Time2a	Наследует от m_me_na_1_fb Время: timestamp (psCP56Time2a_t)
11	m_me_nb_1_fb	M_ME_NB_1 (IEC 60870-5-101, 7.3.1.11)	Масштабированное значение: $-2^{15} \dots +2^{15}-1$ . В описании канала задается коэффициент масштаба	Значение (REAL): SVA, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
12	m_me_te_1_fb	M_ME_TE_1 (IEC 60870-5-101, 7.3.1.27)	Масштабированное значение: $-2^{15} \dots +2^{15}-1$ с временной меткой CP56Time2a. В описании канала задается коэффициент масштаба	Наследует от m_me_nb_1_fb Время: timestamp (psCP56Time2a_t)
13	m_me_nc_1_fb	M_ME_NC_1 (IEC 60870-5-101, 7.3.1.13)	4-байтное число с плавающей запятой (короткий формат)	Значение (REAL): R32, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
14	m_me_tf_1_fb	M_ME_TF_1 (IEC 60870-5-101, 7.3.1.28)	4-байтное число с плавающей запятой (короткий формат) с временной меткой CP56Time2a	Наследует от m_me_nc_1_fb Время: timestamp (psCP56Time2a_t)

№	Функц.блок	Тип IEC-104	Применение	Поля функционального блока
15	m_it_na_1_fb	M_IT_NA_1 (IEC 60870-5-101, 7.3.1.15)	4-байтное целое со знаком	Значение (DINT): COUNTER, value Качество (BYTE): quality, SN_SQ, SN_CU, SN_CA, SN_IV, SN
16	m_it_tb_1_fb	M_IT_TB_1 (IEC 60870-5-101, 7.3.1.29)	4-байтное целое со знаком с временной меткой CP56Time2a	Наследует от m_it_na_1_fb Время: timestamp (psCP56Time2a_t)
17	m_me_nd_1_fb	M_ME_ND_1 (IEC 60870-5-101, 7.3.1.21)	Нормализованное число с плавающей точкой: -1 .. +1-2 <sup>-15</sup> без описателя качества	Значение (REAL): NVA, value
18	m_ep_td_1_fb	M_EP_TD_1 (IEC 60870-5-101, 7.3.1.30)	События устройств защиты с временной меткой CP56Time2a	Значение (BYTE): SEP, ES, value Интервал (UINT): RELAY Качество (BYTE): quality, QD_EI, QD_BL, QD_IV, QD_NT, QD_SB Время: timestamp (psCP56Time2a_t)
19	m_lreal_na_1_fb	M_LREAL_NA_1 Custom type	8-байтное число с плавающей запятой	Значение (LREAL): R64, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
20	m_lreal_tc_1_fb	M_LREAL_TC_1 Custom type	8-байтное число с плавающей запятой с временной меткой CP56Time2a	Наследует от m_lreal_na_1_fb Время: timestamp (psCP56Time2a_t)
21	m_ulint_na_1_fb	M_ULINT_NA_1 Custom type	8-байтное целое беззнаковое	Значение (ULINT): COUNTER, value Качество (BYTE): quality, QD_BL, QD_IV, QD_NT, QD_SB, QD_OV
22	m_ulint_tc_1_fb	M_ULINT_TC_1 Custom type	8-байтное целое беззнаковое с временной меткой CP56Time2a	Наследует от m_ulint_na_1_fb Время: timestamp (psCP56Time2a_t)
23	c_sc_na_1_fb	C_SC_NA_1 (IEC 60870-5-101, 7.3.2.1)	Однопозиционная команда, дискреты: 0,1	Значение (BYTE): SCO, SCS, value Описатель (BYTE): QOC, QU Действие (BYTE): SE
24	c_sc_ta_1_fb	C_SC_TA_1 (IEC 60870-5-101, 7.3.2.1)	Однопозиционная команда с временной меткой CP56Time2a, дискреты: 0,1	Наследует от c_sc_na_1_fb Время: timestamp (psCP56Time2a_t)

№	Функц.блок	Тип IEC-104	Применение	Поля функционального блока
25	c_dc_na_1_fb	C_DC_NA_1 (IEC 60870-5-101, 7.3.2.2)	Двухпозиционная команда с 2-битовым полем	Значение (BYTE): DCO, DCS, value Описатель (BYTE): QOC, QU Действие (BYTE): SE
26	c_dc_ta_1_fb	C_DC_TA_1 (IEC 60870-5-101, 7.3.2.2)	Двухпозиционная команда с временной меткой CP56Time2a, с 2-битовым полем	Наследует от c_dc_na_1_fb Время: timestamp (psCP56Time2a_t)
27	c_rc_na_1_fb	C_RC_NA_1 (IEC 60870-5-101, 7.3.2.3)	Команда пошагового регулирования	Значение (BYTE): RCO, RCS, value Описатель (BYTE): QOC, QU Действие (BYTE): SE
28	c_rc_ta_1_fb	C_RC_TA_1 (IEC 60870-5-101, 7.3.2.3)	Команда пошагового регулирования с временной меткой CP56Time2a	Наследует от c_rc_na_1_fb Время: timestamp (psCP56Time2a_t)
29	c_se_na_1_fb	C_SE_NA_1 (IEC 60870-5-101, 7.3.2.4)	Команда уставки, нормализованное значение	Значение (REAL): NVA, value Описатель (BYTE): QOS, QL Действие (BYTE): SE
30	c_se_ta_1_fb	C_SE_TA_1 (IEC 60870-5-101, 7.3.2.4)	Команда уставки, нормализованное значение с временной меткой CP56Time2a	Наследует от c_se_na_1_fb Время: timestamp (psCP56Time2a_t)
31	c_se_nb_1_fb	C_SE_NB_1 (IEC 60870-5-101, 7.3.2.5)	Команда уставки, масштабированное значение	Значение (INT): SVA, (REAL) value Описатель (BYTE): QOS, QL Действие (BYTE): SE
32	c_se_tb_1_fb	C_SE_TB_1 (IEC 60870-5-101, 7.3.2.5)	Команда уставки, масштабированное значение с временной меткой CP56Time2a	Наследует от c_se_nb_1_fb Время: timestamp (psCP56Time2a_t)
33	c_se_nc_1_fb	C_SE_NC_1 (IEC 60870-5-101, 7.3.2.6)	Команда уставки, короткий формат с плавающей запятой	Значение (REAL): R32, value Описатель (BYTE): QOS, QL Действие (BYTE): SE
34	c_se_tc_1_fb	C_SE_TC_1 (IEC 60870-5-101, 7.3.2.6)	Команда уставки, короткий формат с плавающей запятой с временной меткой CP56Time2a	Наследует от c_se_nc_1_fb Время: timestamp (psCP56Time2a_t)



№	Функц.блок	Тип IEC-104	Применение	Поля функционального блока
35	c_bo_na_1_fb	C_BO_NA_1 (IEC 60870-5-101, 7.3.2.7)	Строка из 32 бит	Значение (DWORD): BSI, value
36	c_bo_ta_1_fb	C_BO_TA_1 (IEC 60870-5-101, 7.3.2.7)	Строка из 32 бит с временной меткой CP56Time2a	Наследует от c_bo_na_1_fb Время: timestamp (psCP56Time2a_t)



### ИНФОРМАЦИЯ

Допускается использование одних и тех же экземпляров переменных (и данных, и команд) для привязки к разным экземплярам драйверов (Master/Slave, 101/104 OS), при условии, что настройки привязанной переменной (параметры элементов данных/команд) для всех экземпляров драйверов идентичны.

На этапе компиляции проекта выполняется проверка совпадения настроек, в случае несовпадения, в окно сообщений выводится сообщение об ошибке, с указанием имени переменной и имен соответствующих экземпляров драйверов

### Особенности работы с функциональными блоками данных и команд

Описание функциональных блоков и особенности работы с их экземплярами в плане чтения/записи полей значения, качества и временной метки совпадают с разделом «Настройка контроллера в качестве Slave 104 OS».

Особенностью исполнения команд мастера, для которых задано ненулевое значение настройки «Выборка перед выполнением», т.е. SBO-команд, является автоматическое выполнение обоих шагов – выборки (Select) и выполнения (Execute), без участия пользователя. При этом, после отправки команды выбора (SELECT.req) драйвер отслеживает задержку получения подтверждения выбора от слэйва (SELECT.con). Если время ожидания превышает заданную для данной команды величину в настройке «Выборка перед выполнением», то выполнение команды отменяется. Иначе, драйвер отправляет команду исполнения (EXCO.req) и ожидает получение от слэйва подтверждения исполнения (EXCO.con) в течение периода времени, заданного в настройке «Таймаут выполнения». На всех этапах обмена пакетами между мастером и слэйвом выполняется проверка идентичности пересылаемого значения и метки времени (если имеется).

Аналогично функциональным блокам данных драйвера Slave 104 OS, для поддержания консистентности на уровне экземпляра функционального блока команды доступны два режима работы при отправке команд драйвера OuterSlave104.

### Режим автоматической отправки команд (включен по умолчанию)

В этом режиме происходит автоматическая отправка команды при каждом присвоении значения (или одного из его полей) или описателя команды (или любого его признака)

информационного объекта. При этом, используется следующий механизм блокировки отправки команды - блок кода, в котором производятся изменения различных полей объекта команды, заключается в вызовы соответствующих методов:

```
cmd_c_se_ta_1          : PsIEC60870Bridge.c_se_ta_1_fb;
...
cmd_c_se_ta_1.lock_send();           // блокировка отправки
cmd_c_se_ta_1.value := 123.45;
cmd_c_se_ta_1.QL := 1;
cmd_c_se_ta_1.SE := 1;
cmd_c_se_ta_1.timestamp := PsIEC60870Bridge.get_psCP56Time2a_now(TRUE);
cmd_c_se_ta_1.unlock_send();        // отправка команды
```

Метод *unlock\_send()* вызывает автоматическую отправки команды. Если метка времени не устанавливалась вручную, то для соответствующих типов она будет автоматически обновлена с сохранением флагов метки, не относящихся к времени (Invalid, Summer, Reserve bits).

При необходимости, включение этого режима для каждого отдельного экземпляра драйвера **OuterSlave104** производится через свойство *SendMode*:

```
OuterSlave_104.SendMode := PsIEC60870Bridge.e_sm_auto;
```

### Режим ручной отправки команды

В ручном режиме не происходит автоматической отправки команды, при изменении ее значения или описателя. Для помещения состояния экземпляра функционального блока в очередь отправки команд необходимо в коде вызвать для него метод *send()*:

```
cmd_c_se_ta_1.value := 123.45;
cmd_c_se_ta_1.QL := 1;
cmd_c_se_ta_1.SE := 1;
cmd_c_se_ta_1.timestamp := PsIEC60870Bridge.get_psCP56Time2a_now(TRUE);
cmd_c_se_ta_1.send();           // отправка команды
```

При необходимости, включение режима для каждого отдельного экземпляра драйвера **Outer Slave 104** производится также через свойство *SendMode*:

```
OuterSlave_104.SendMode := PsIEC60870Bridge.e_sm_manual;
```

Дополнительно, в ручном режиме работы, предусмотрена возможность отправки набора всех измененных на текущий момент времени команд. Для этого для экземпляра драйвера **Outer Slave 104** в коде необходимо вызвать метод *send()*:

```
cmd_c_se_ta_01.value := 123.45;
cmd_c_se_ta_01.QL := 1;
cmd_c_se_ta_01.SE := 123;
cmd_c_se_ta_02.QOS := 0x80;
cmd_c_se_ta_03.timestamp := PsIEC60870Bridge.get_psCP56Time2a_now(TRUE);
...
cmd_c_se_ta_KK.value := 543.21;
...
cmd_c_se_ta_NN.timestamp := PsIEC60870Bridge.get_psCP56Time2a_now(TRUE);

OuterSlave_104.send();
```



### ИНФОРМАЦИЯ

Методы lock\_send(), unlock\_send(), и send() ФБ команд доступны для работы только при привязке переменных этих типов к экземплярам драйверов Outer Slave 104 OS и Outer Slave 101 OS

## Журналирование работы драйвера

Для просмотра журнала работы драйвера перейдите на основную страницу параметров устройства во вкладку Журнал, куда могут выводиться следующие сообщения драйвера Master 104 OS:

- подключение/отключение к серверу (слэйву) по основному/резервному IP-адресам (Рисунок 53).

Жесткость	Временная отметка	Описание	Компонент
!	10.08.2022 17:03:02.118	Master 104[Master 104]: disconnected from 172.29.34.134:2404[base]	PsiIEC60870Master 104_OS
!	10.08.2022 17:03:02.094	Master 104[Master 104]: STOP mode is ON, STOPBEHAVIOUR = NO_ACTIVITY	PsiIEC60870Master 104_OS
!	10.08.2022 17:03:02.071	Master 104 : BusCycle : APP is STOPPED	PsiIEC60870Master 104_OS
!	10.08.2022 17:02:50.384	Master 104[Master 104]: connected to 172.29.34.134:2404[base]	PsiIEC60870Master 104_OS
!	10.08.2022 17:02:49.616	Master 104[Master 104]: STOP mode is OFF	PsiIEC60870Master 104_OS
!	10.08.2022 17:02:49.596	Master 104 : BusCycle : APP is STARTED	PsiIEC60870Master 104_OS
!	10.08.2022 17:02:04.287	Master 104[Master 104]: stopped (STOP mode is ON, STOPBEHAVIOUR = NO_ACTIVITY)	PsiIEC60870Master 104_OS
!	10.08.2022 17:02:04.287	Master 104[Master 104]: '14' commands	PsiIEC60870Master 104_OS
!	10.08.2022 17:02:04.287	Master 104[Master 104]: '22' data params	PsiIEC60870Master 104_OS

Рисунок 53 - Сообщения о работе драйвера Master 104 OS

Дополнительную информацию о работе драйвера можно получить, просмотрев лог-файлы. Для загрузки файлов, перейдите на страницу параметров устройства во вкладку **Файлы**. В области **Исполнение** нажмите кнопку (**Обновить**). В окне отобразится дерево файлов, имеющихся на контроллере. Перейдите в папку **logs**→**logger**→**user** (Рисунок 54). Имена лог-файлов имеют вид: «*iec60870driver\_\*.log*». Скопируйте лог-файлы с контроллера для просмотра на ПК (кнопкой , из **Исполнение** в **Хост**).

Имя	Размер	Изменено
background.log	2,08 KB (2 125 байт)	09.01
crashreporter.log	1,61 KB (1 648 байт)	09.01
dnhub.log	33,29 KB (34 092 байт)	09.01
espd-default.log	46,91 KB (48 036 байт)	09.01
espd-redundancy.log	3,11 KB (3 180 байт)	09.01
ftp_users.log	6,27 KB (6 425 байт)	09.01
gnod.log	2,15 KB (2 205 байт)	09.01
ham.log	42,07 KB (43 078 байт)	09.01
hart_dnhub_driver.log	3,14 KB (3 212 байт)	09.01
iec60870driver.log	4,59 MB (4 817 060 байт)	09.01
iec60870driver_20220808_142726.log	5,00 MB (5 242 738 байт)	08.01
iec60870driver_20220808_143534.log	5,00 MB (5 242 775 байт)	08.01
iec60870driver_20220808_144340.log	5,00 MB (5 242 762 байт)	08.01
iec60870driver_20220808_145149.log	5,00 MB (5 242 753 байт)	08.01
iec60870driver_20220808_145959.log	5,00 MB (5 242 740 байт)	08.01

Рисунок 54 - Дополнительные лог-файлы

Можно задать детальность трассировки в файлах с помощью свойства **LogLevel** экземпляра драйвера в процессе работы приложения, например:

```
Master101.LogLevel:= PsiIEC60870Bridge.LOG_LEVEL_PDU_MAX;
```

Константы, включающие определенный уровень логирования, описаны в библиотеке PsIEC60870Bridge. (Рисунок 55). Константы можно объединить с помощью двоичного оператора ИЛИ (OR). Уровень, соответствующий LOG\_LEVEL\_BASE, включен всегда.

Имя	Тип	Наследовано от	Адрес	Начальн.	Комментарий
LOG_LEVEL_BASE	DWORD			16#0	base log level
LOG_LEVEL_PDU_SHORT	DWORD			16#1	short PDU trace log level
LOG_LEVEL_PDU_MAX	DWORD			16#2	full PDU trace log level
LOG_LEVEL_SPONT_DATA	DWORD			16#4	spontaneous data event log level
LOG_LEVEL_SPONT_VALUE	DWORD			16#8	spontaneous data values log level
LOG_LEVEL_CMD_DATA	DWORD			16#10	command data event log level
LOG_LEVEL_CMD_VALUE	DWORD			16#20	command data values log level
LOG_LEVEL_IG_DATA	DWORD			16#40	interrogation data event log level
LOG_LEVEL_IG_VALUE	DWORD			16#80	interrogation data values log level
LOG_LEVEL_PERIODIC_DATA	DWORD			16#100	periodic data event log level
LOG_LEVEL_PERIODIC_VALUE	DWORD			16#200	periodic data values log level
LOG_LEVEL_READ_DATA	DWORD			16#400	read request data event log level
LOG_LEVEL_READ_VALUE	DWORD			16#800	read request data values log level
LOG_LEVEL_ALL	DWORD			16#FFF	all log levels

Рисунок 55 – Константы

## НАСТРОЙКА IEC-101



### ИНФОРМАЦИЯ

Можно использовать одни и те же переменные в разных экземплярах драйверов **Slave 101/104** и **Slave 101/104 OS**, при условии, что в драйверах осуществлены идентичные настройки для переменных. На этапе компиляции будет производиться проверка переменных и, в случае выявления несоответствия, выдавать ошибку

### Добавление последовательного порта в конфигурацию контроллера

Перед настройкой IEC-101 в конфигурацию контроллера необходимо добавить последовательный порт.

#### Добавление порта

Для обмена данными будет использоваться один из портов: COM1 (интерфейс RS-232) или COM2 (интерфейс RS-485) на модуле центрального процессора, либо один из портов на модуле коммуникационного процессора (интерфейс RS-485).

#### Добавление последовательного порта модуля центрального процессора

Если для обмена данными будет использоваться порт модуля центрального процессора, то в конфигурацию необходимо добавить к контроллеру устройство **Regul Serial Port**. Для этого в окне дерева устройств поместите курсор на головное устройство (Device), нажмите правую кнопку мыши. В появившемся контекстном меню выберите пункт **Добавить устройство...** Откроется окно **Добавить устройство**, в котором выберите *Regul* → *Последовательный порт* → *Regul Serial Port*. Нажмите кнопку **Добавить устройство** или дважды щелкните левой кнопкой мыши. Выбранное устройство появится в проекте в дереве устройств (Рисунок 56).

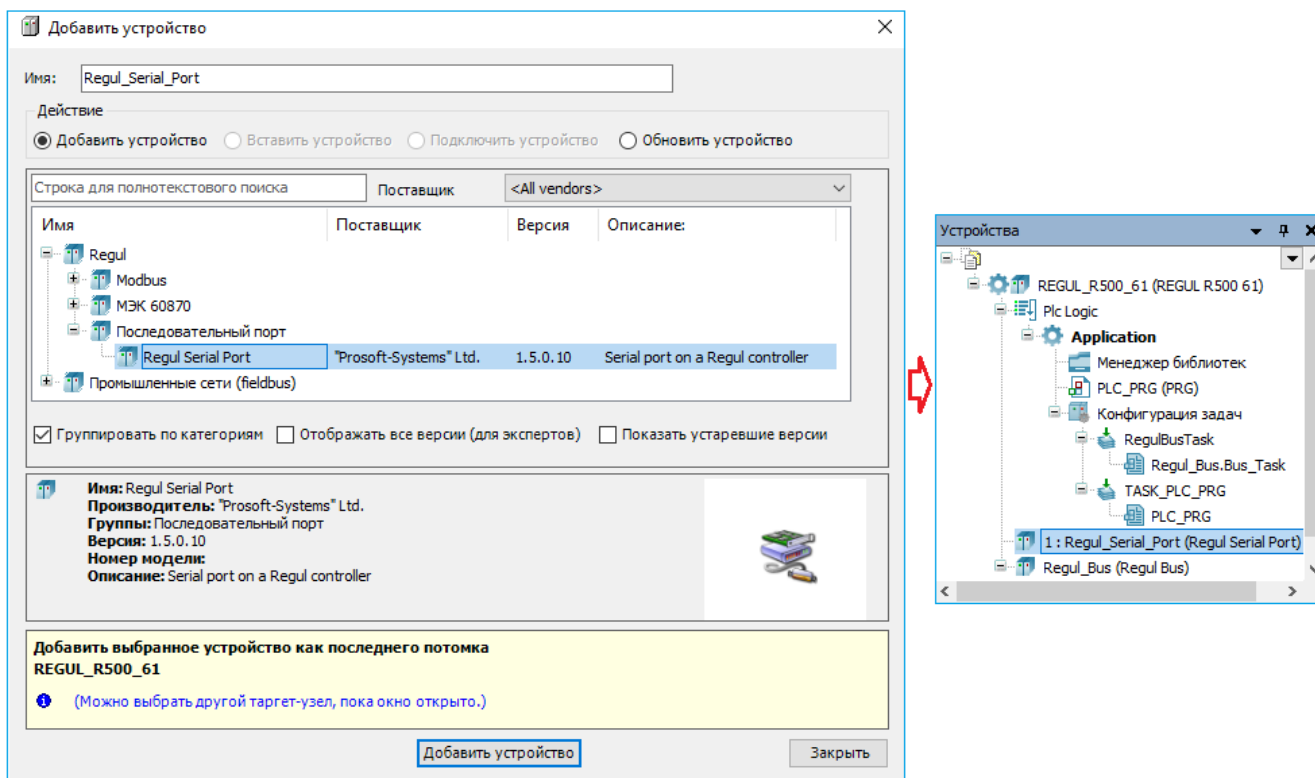


Рисунок 56 – Добавление последовательного порта Regul Serial Port

Двойным щелчком по названию **Regul Serial Port** откроется вкладка параметров (Рисунок 57).

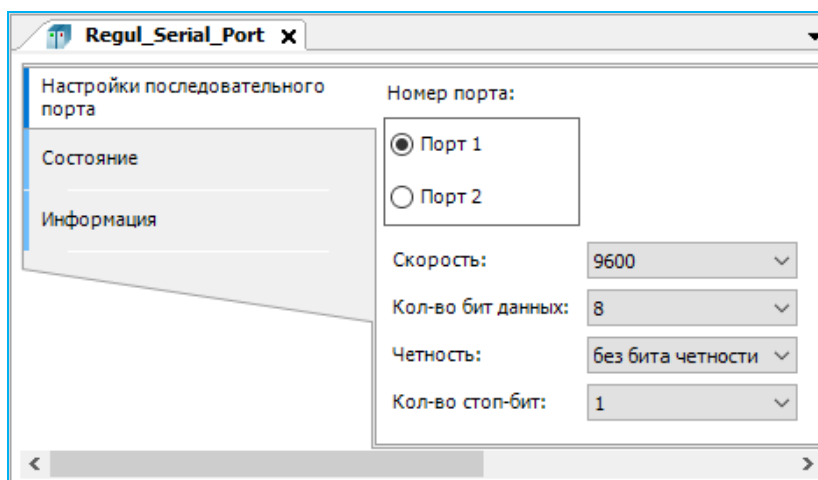


Рисунок 57 – Настройки последовательного порта Regul Serial Port

### Добавление последовательного порта коммуникационного модуля

Если для обмена данными будет использоваться порт модуля коммуникационного процессора, то в конфигурацию необходимо добавить к модулю коммуникационного процессора устройство **Extended Regul Serial Port** (*Regul* → *Последовательный порт* → *Extended Regul Serial Port*). Выбранное устройство появится в проекте в дереве устройств (Рисунок 58).

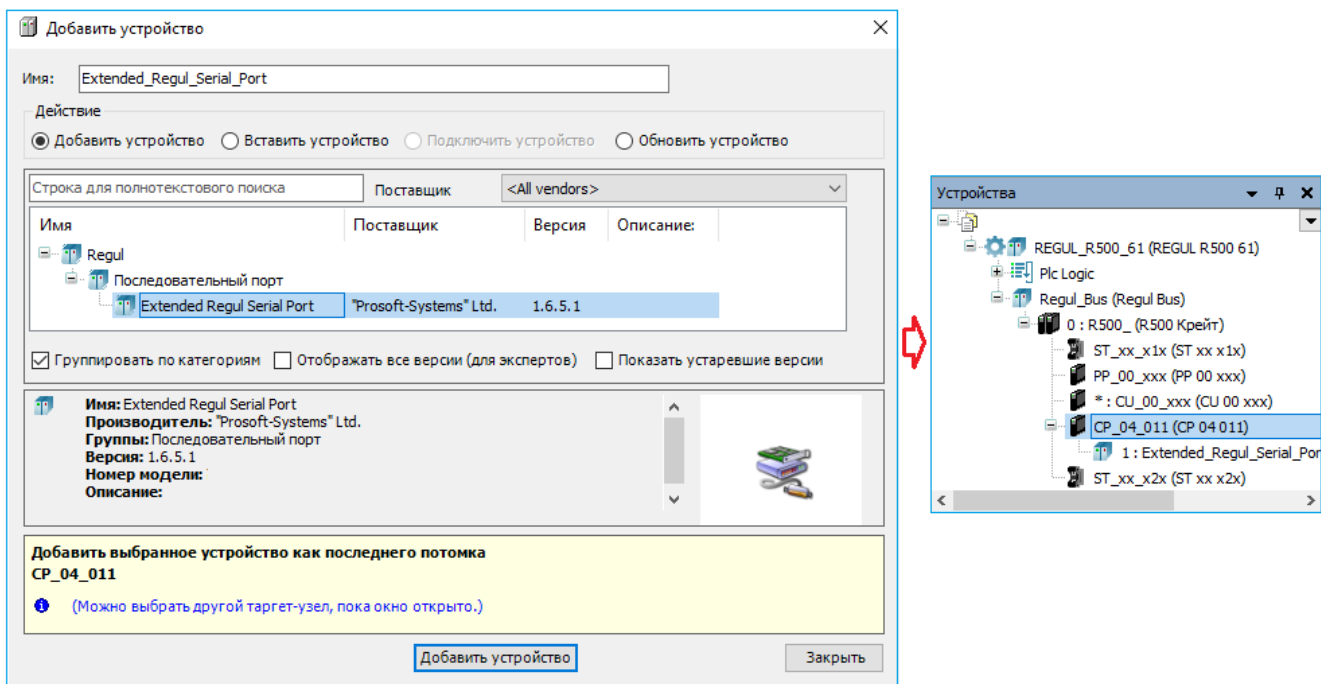


Рисунок 58 – Добавление последовательного порта Extended Regul Serial Port

Двойным щелчком по названию **Extended Regul Serial Port** откроется вкладка параметров (Рисунок 59).

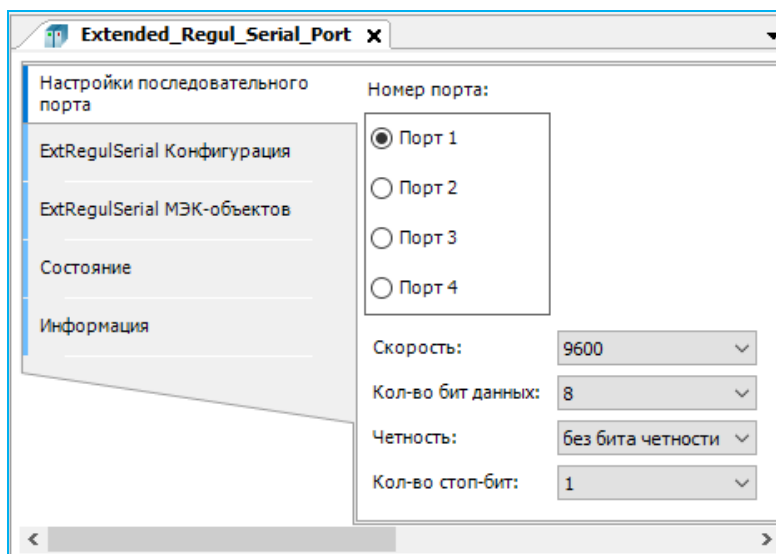


Рисунок 59 – Настройки последовательного порта Extended Serial Port

### Настройка параметров порта

Откройте главную вкладку параметров порта и перейдите на внутреннюю вкладку **Настройки последовательного порта** (Рисунок 58, 59). В поле **Номер порта** выберите из списка порт модуля, к которому подключено внешнее устройство и установите переключатель в соответствующей строке. Далее, выбирая значение в раскрывающемся списке, задайте следующие параметры:

- **Скорость** – значения скорости передачи данных: 1200, 2400, 4800, 9600, 19200, 38400, 115200;
- **Кол-во бит данных** – значения: 5, 6, 7, 8;
- **Четность** – значения: без бита четности, проверка на нечетность, проверка на четность;
- **Кол-во стоп-бит** – 1 или 2.



## Настройка контроллера в качества Slave 101

### Добавление устройства Unbalanced Secondary 101 Driver

Добавьте устройство **Unbalanced Secondary 101 Driver** к последовательному порту **Regul Serial Port** или **Extended Regul Serial Port** (*Regul → МЭК 60870 → Serial 60870-101 Slave → Unbalanced Secondary 101 Driver*). Выбранное устройство появится в проекте в дереве устройств (Рисунок 60).

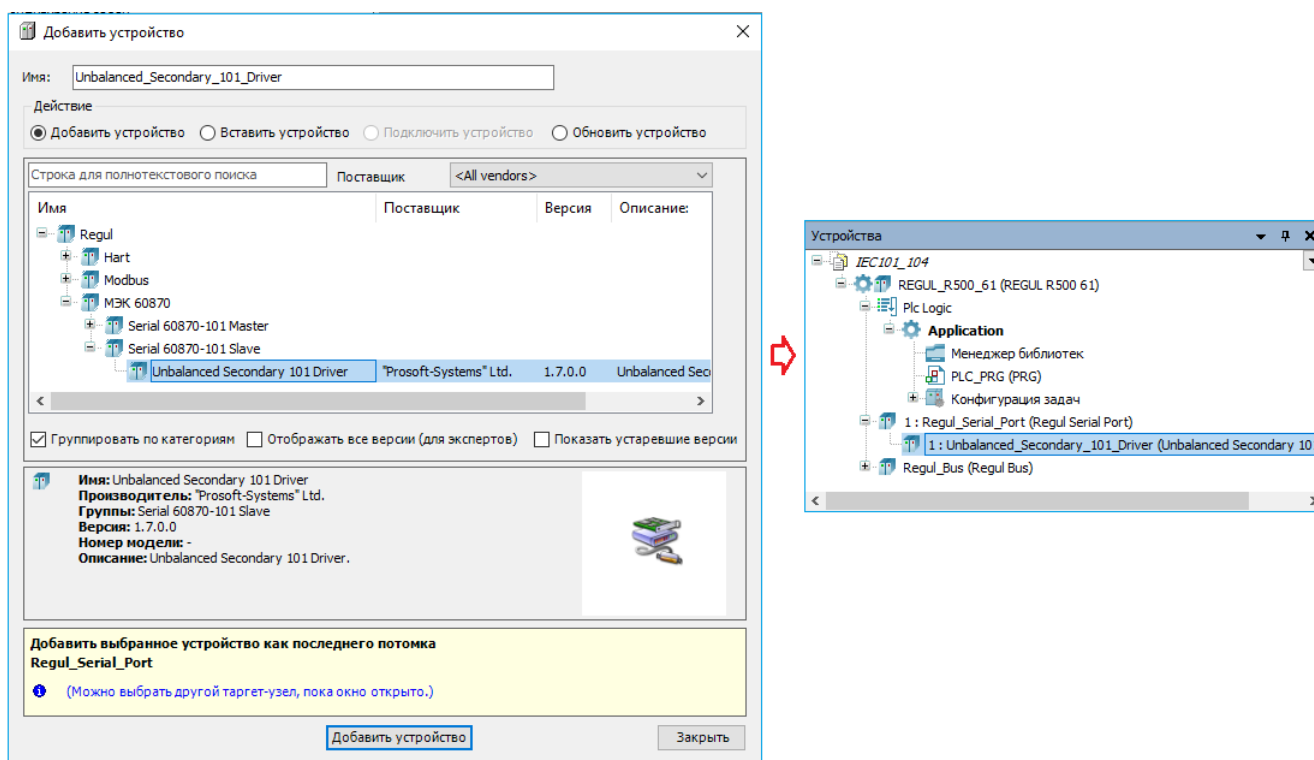


Рисунок 60 – Добавление в конфигурацию контроллера устройство Unbalanced Secondary 101 Driver

Двойным щелчком по названию устройства **Unbalanced Secondary 101 Driver** откройте вкладку параметров. По умолчанию открывается первая внутренняя вкладка **Редактор IEC 101 Unbalanced Secondary** (Рисунок 61).

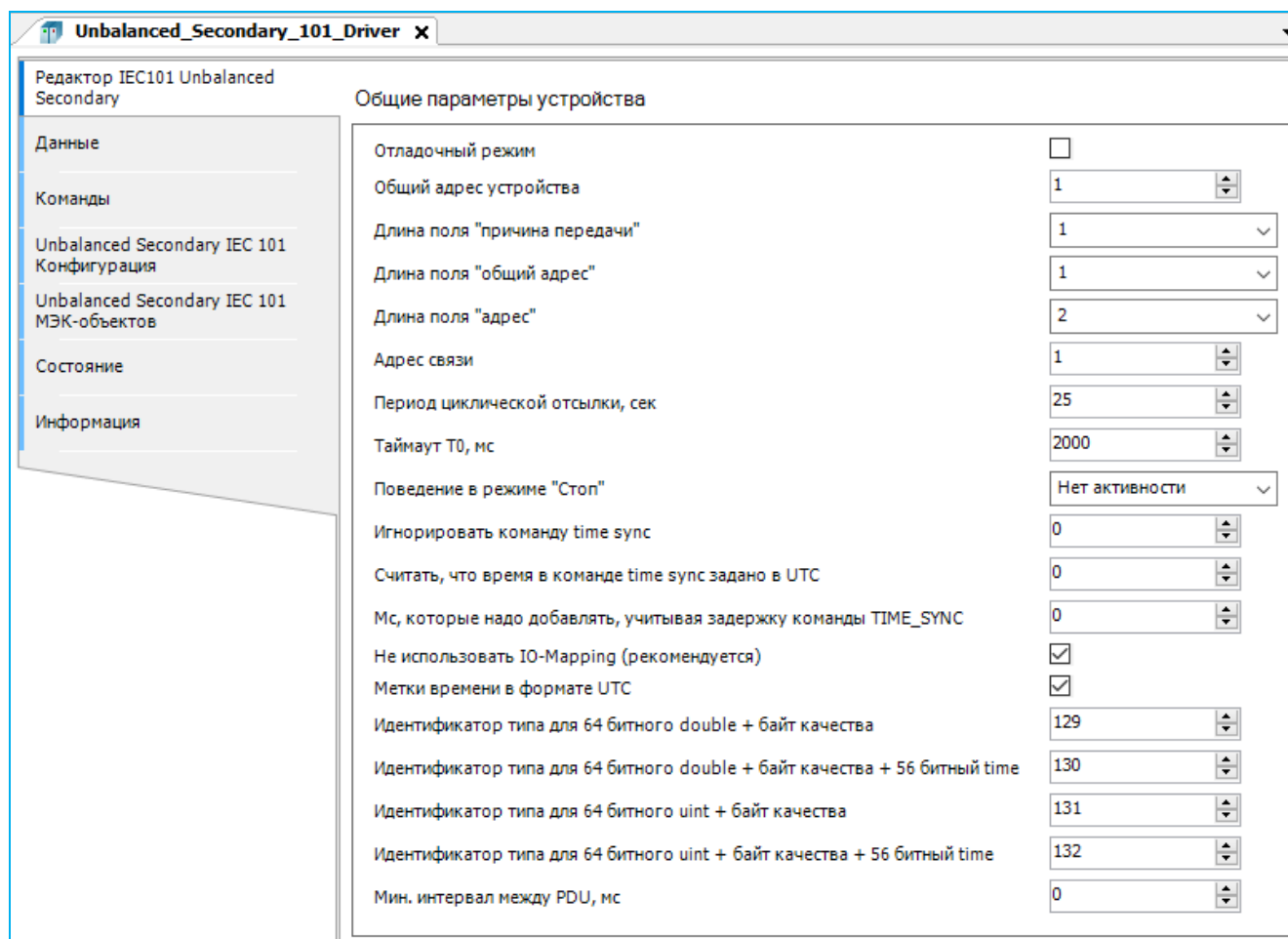


Рисунок 61 – Редактор устройства IEC101 Unbalanced Secondary

### Настройка общих параметров устройства Unbalanced Secondary 101 Driver

В редакторе устройства (Рисунок 61) в блоке **Общие параметры устройства** доступны для настройки параметры, представленные в таблице 27.

Таблица 27 – Общие параметры устройства

Параметр	Описание	Значение по умолчанию
<b>Отладочный режим</b>	При установке флажка <input checked="" type="checkbox"/> в поле, все детали обмена по IEC-101 будут записаны в журнал работы контроллера, а именно передаваемые и принимаемые пакеты, комментарии к возникающим ошибкам и т.п.	
<b>Общий адрес устройства</b>	Содержит общий адрес (COMMON ADDRESS, ASDU ADDRESS) устройства, все данные и команды имеют общую часть, равную значению этого параметра (см. спецификацию IEC-101)	1
<b>Длина поля «причина передачи»</b>	Содержит длину поля «причина передачи» (COT, Cause Of Transmission). Допустимые значения параметра: 1 или 2	1

Параметр	Описание	Значение по умолчанию
Длина поля «общий адрес», Длина поля «адрес»	Характеристики PDU (Protocol Data Unit – протокольная единица, пакет), передаваемых по IEC-101, должны быть одинаковы для пары master-slave. Допустимые значения параметра «общий адрес»: 1 или 2. Допустимые значения параметра «адрес»: 2 или 3	«общий адрес» - 1, «адрес» - 2
Адрес связи (Link Address)	Уникальный адрес для процедур обмена данными канального уровня	1
Период циклической отсылки, сек	При описании элемента данных IEC-101 можно включить его в циклическую рассылку (установка флажка в поле <b>Участвует в циклической рассылке</b> ). Для всех таких элементов их значения даже при отсутствии изменений будут отсылаться с указанным здесь периодом (в секундах)	25
Таймаут T0, мс	Интервал времени в миллисекундах. Определяет максимальную задержку ответа на запрос. Если задержка превышает данный интервал, то соединение считается потерянным	2000
Поведение в режиме «Стоп»*	Определяет поведение компонента при остановке программы. <i>Нет активности</i> - означает, что устройство Unbalanced Secondary 101 Driver прекращает все соединения и не воспринимает новые; <i>Нормальная работа</i> - означает продолжение работы в обычном режиме	Нет активности
Игнорировать команду time sync	Если значение не равно 0, то при получении команды синхронизации времени (C_CS_NA_1), синхронизации времени не происходит. Если этот параметр равен 0, то команда синхронизации времени выполняется	0
Считать, что время в команде time sync задано в UTC	Если в этом поле задано значение 1, то, при синхронизации времени, время в составе команды будет интерпретировано, как UTC. Если задано значение 0, то, это время будет интерпретировано, как локальное время	0
мс, которые надо добавлять, учитывая задержку команды TIME_SYNC	Корректировка команды синхронизации времени	0
Не использовать IO-Mapping (рекомендуется)	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что чтение/запись привязанных переменных происходит по адресу (прямой доступ к памяти ПЛК), без использования механизма I/O Mapping (соотнесение входов/выходов)	

Параметр	Описание	Значение по умолчанию
<b>Метки времени в формате UTC</b>	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что значение метки времени соответствует UTC, иначе - локального времени	
<b>Идентификатор типа для 64 битного double + байт качества</b>	Задание ID для пользовательского типа LREAL(double), без метки времени	129
<b>Идентификатор типа для 64 битного double + байт качества + 56 битный time</b>	Задание ID для пользовательского типа LREAL(double), с меткой времени	130
<b>Идентификатор типа для 64 битного uint + байт качества</b>	Задание ID для пользовательского типа ULINT(uint), без метки времени	131
<b>Идентификатор типа для 64 битного uint + байт качества + 56 битный time</b>	Задание ID для пользовательского типа ULINT(uint), с меткой времени	132
<b>Мин. интервал между PDU, мс</b>	Определяет минимальный интервал времени между PDU, в случае, когда используется RS-485 (рекомендуемое значение для RS-485: 5...20)	0
<p><b>Примечание</b> -*- предусмотрена возможность самостоятельно активировать «поведение в режиме «Стоп» в программном коде. Для активации режима требуется в программе присвоить значение TRUE свойству ActivateStopBehavior необходимого устройства:</p> <p>«IEC101USSlave_device_name».ActivateStopBehavior := TRUE;</p> <p>После этого slave-устройство перейдет в Стоп-режим работы</p>		



### ИНФОРМАЦИЯ

Для обзора всех параметров можно перейти на вкладку **Unbalanced Secondary IEC 101 Конфигурация**, если вкладка отсутствует, то необходимо перейти в меню **Инструменты** ⇒ **Опции** ⇒ найти пункт **Редактор устройств** и установить флажок напротив поля **Показывать общие окна конфигурации устройств**

Все последующие настройки областей данных Slave 101 и приемы работы с ними идентичны тем, что описаны в разделе **Настройка контроллера в качестве Slave 104**, начиная с подраздела [Создание списка элементов данных, передаваемых по IEC-104](#).

## Настройка контроллера в качества Slave 101 OS

### Добавление устройства Драйвер Unbalanced Secondary 101 OS

Добавьте устройство **Драйвер Unbalanced Secondary 101 OS** к последовательному порту **Regul Serial Port** или **Extended Regul Serial Port** (*Regul* → *МЭК 60870* → *Serial 60870-101 Slave* → *Драйвер Unbalanced Secondary 101 OS*). Выбранное устройство появится в проекте в дереве устройств (Рисунок 62).

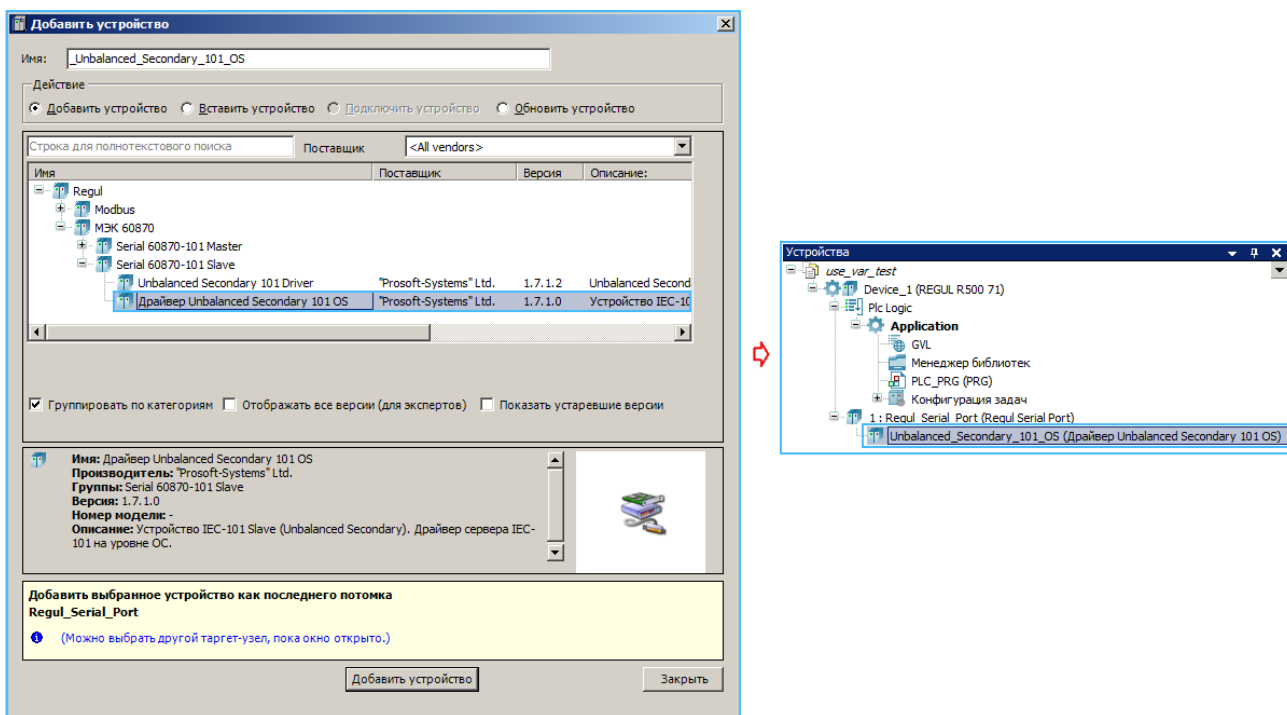


Рисунок 62- Добавление в конфигурацию контроллера устройство Драйвер Unbalanced Secondary 101 OS

Двойным щелчком по названию устройства **Драйвер Unbalanced Secondary 101 OS** откройте вкладку параметров. По умолчанию открывается первая внутренняя вкладка **Редактор IEC 101 Unbalanced Secondary** (Рисунок 63).

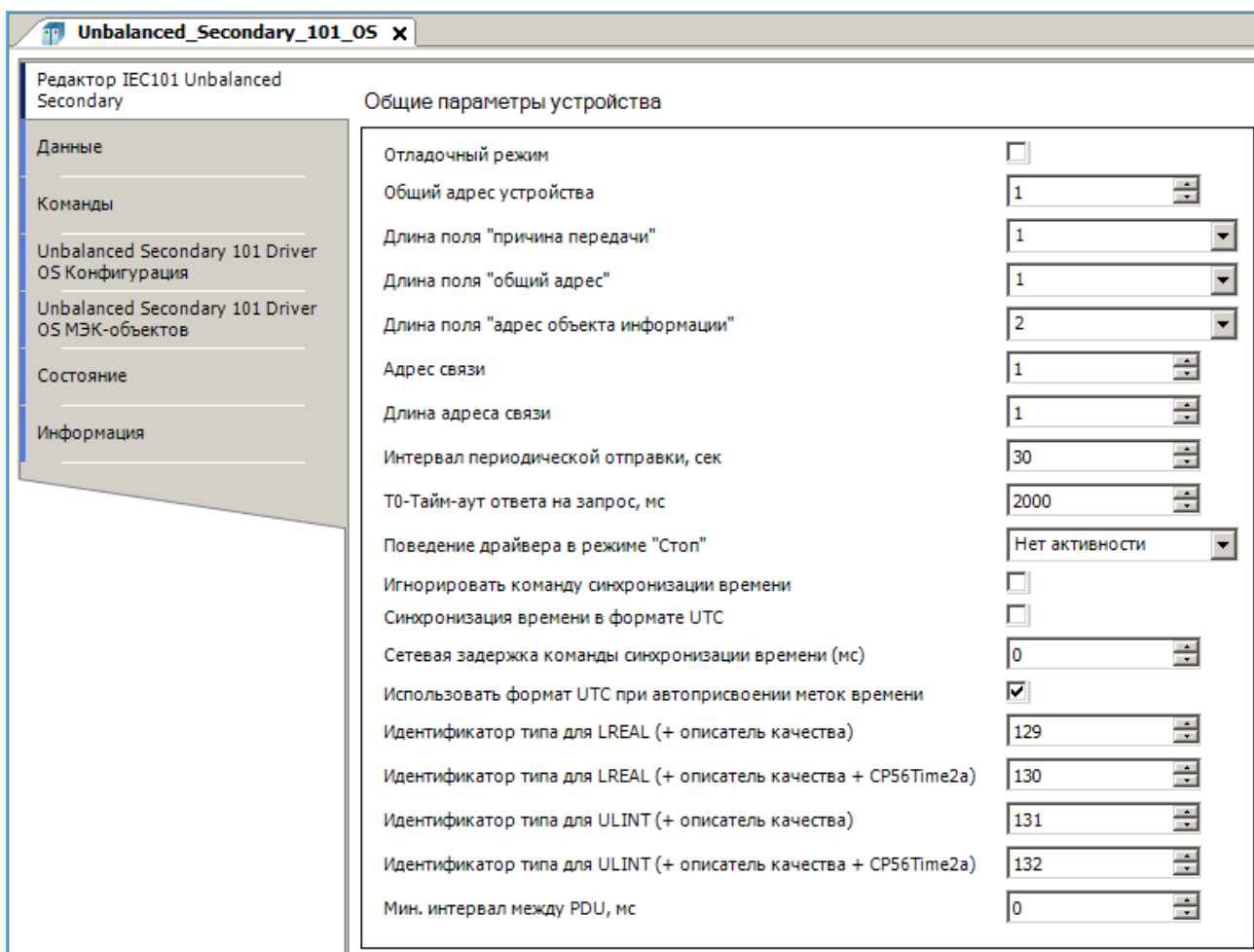


Рисунок 63 - Редактор устройства IEC101 Unbalanced Secondary

### Настройка общих параметров устройства Драйвер Unbalanced Secondary 101 OS



В редакторе устройства (Рисунок 63) в блоке Общие параметры устройства доступны для настройки параметры, представленные в таблице 28.

Таблица 28 – Общие параметры устройства

Параметр	Описание	Значение по умолчанию
<b>Отладочный режим</b>	При установке флажка <input checked="" type="checkbox"/> в этом поле в журнал работы контроллера будут выводиться сообщения об ошибках, предупреждениях и ключевых моментах работы драйвера.	
<b>Общий адрес устройства</b>	Содержит общий адрес (COMMON ADDRESS, ASDU ADDRESS) устройства, все данные и команды имеют общую часть, равную значению этого параметра (см. спецификацию IEC-101)	1
<b>Длина поля «причина передачи»</b>	Содержит длину поля «причина передачи» (COT, Cause Of Transmission). Допустимые значения параметра: 1 или 2	1
<b>Длина поля «общий адрес»</b>	Содержит длину поля «общий адрес» (CA, Common Address). Допустимые значения параметра: 1 или 2.	1

Параметр	Описание	Значение по умолчанию
Длина поля «адрес объекта информации»	Содержит длину поля «адрес объекта информации» (IOA, Information Object Address). Допустимые значения параметра: 2 или 3	2
Адрес связи	Уникальный адрес для процедур обмена данными канального уровня	1
Длина адреса связи	Длина поля уникального адреса, размер: от 1 до 65535 (1 или 2 байта)	1
Интервал периодической отправки, сек	При описании элемента данных IEC-101 можно включить его в периодическую рассылку (установка флажка в поле <b>Участвует в циклической рассылке</b> ). Для всех таких элементов их значения, даже при отсутствии изменений, будут отсылааться с указанным здесь периодом	30
T0-Таймаут ответа на запрос, мс	Интервал времени в миллисекундах. Определяет максимальную задержку ответа на запрос. Если задержка превышает данный интервал, то соединение считается потерянным	2000
Поведение драйвера в режиме «Стоп»*	Определяет поведение компонента при остановке программы. <i>Нет активности</i> - означает, что устройство Драйвер Unbalanced Secondary 101 прекращает обмен и закрывает последовательный порт. <i>Нормальная работа</i> - означает продолжение работы в обычном режиме.	Нет активности
Игнорировать команду синхронизации времени	При установке флажка <input checked="" type="checkbox"/> в этом поле будет запрещена обработка команды синхронизации времени (C_CS_NA_1)	0
Синхронизация времени в формате UTC	Если будет установлен флажок <input checked="" type="checkbox"/> в этом поле, то, при синхронизации временная метка в составе команды будет интерпретирована, как UTC. Если флажок снят, то, это время будет интерпретировано, как локальное время	0
Сетевая задержка команды синхронизации времени (мс)	Корректировка команды синхронизации времени – добавляется к временной метке команды	0
Использовать формат UTC при автоприсвоении меток времени	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что значение метки времени при отправке временных типов данных будет выставлено в формате UTC, иначе - локального времени	0
Идентификатор типа для LREAL (+описатель качества)	Задание ID для пользовательского типа LREAL(double), без метки времени	129

Параметр	Описание	Значение по умолчанию
<b>Идентификатор типа для LREAL (+описатель качества +CP56Time2a)</b>	Задание ID для пользовательского типа LREAL(double), с меткой времени	130
<b>Идентификатор типа для ULINT (+описатель качества)</b>	Задание ID для пользовательского типа ULINT(uint64), без метки времени	131
<b>Идентификатор типа для ULINT (+описатель качества +CP56Time2a)</b>	Задание ID для пользовательского типа ULINT(uint64), с меткой времени	132
<b>Мин. интервал между PDU, мс</b>	Определяет минимальный интервал времени между PDU, в случае, когда используется RS-485 (рекомендуемое значение для RS-485: 5...20)	0
<p><b>Примечание</b> -*- предусмотрена возможность самостоятельно активировать «поведение в режиме «Стоп» в программном коде. Для активации режима требуется в программе присвоить значение TRUE свойству ActivateStopBehavior необходимого устройства:                      «IEC101USSlave_device_name».ActivateStopBehavior := TRUE;                      После этого slave-устройство перейдет в Стоп-режим работы</p>		

	<p><b>ИНФОРМАЦИЯ</b></p> <p>Для обзора всех параметров можно перейти на вкладку <b>Unbalanced Secondary 101 Драйвер OS Конфигурация</b>, если вкладка отсутствует, то необходимо перейти в меню <b>Инструменты</b> ⇒ <b>Опции</b> ⇒ найти пункт  <b>Редактор устройств</b> и установить флажок напротив поля <b>Показывать общие окна конфигурации устройств</b></p>
---	---

### Журналирование работы драйвера

Для просмотра журнала работы драйвера перейдите на основную страницу параметров устройства во вкладку **Журнал**, куда могут выводиться следующие сообщения драйвера Slave 101 OS:

- инициализация, запуск и остановка работы драйвера (Рисунок 64);



Жёсткость	Временная отметка	Описание	Компонент
!	09.08.2022 11:59:58.662	Slave101[s101_1]: serial '#10' is closed	PsiIEC60870UsSlave101_OS
!	09.08.2022 11:59:58.662	Slave101[s101_1]: stopped	PsiIEC60870UsSlave101_OS
!	09.08.2022 11:59:58.650	Slave101[s101_1]: STOP mode is ON, STOPBEHAVIOUR = NO_ACTIVITY	PsiIEC60870UsSlave101_OS
!	09.08.2022 11:59:58.638	s101_1 : BusCycle : APP is STOPED	PsiIEC60870UsSlave101_OS
!	09.08.2022 11:59:54.461	Slave101[s101_1]: started	PsiIEC60870UsSlave101_OS
!	09.08.2022 11:59:54.461	Slave101[s101_1]: serial '#10 9600 8N1' is created and opened OK	PsiIEC60870UsSlave101_OS
!	09.08.2022 11:59:54.461	Slave101[s101_1]: STOP mode is OFF	PsiIEC60870UsSlave101_OS
!	09.08.2022 11:59:54.439	s101_1 : BusCycle : APP is STARTED	PsiIEC60870UsSlave101_OS
!	09.08.2022 11:59:40.253	Slave101[s101_1]: stopped (STOP mode is ON, STOPBEHAVIOUR = NO_ACTIVITY)	PsiIEC60870UsSlave101_OS
!	09.08.2022 11:59:40.253	Slave101[s101_1]: '14' commands	PsiIEC60870UsSlave101_OS
!	09.08.2022 11:59:40.241	Slave101[s101_1]: '22' data params, cyclic '0', interrogation '21'	PsiIEC60870UsSlave101_OS

Рисунок 64 - Трассировка событий драйвера в журнале устройства

– получение команды синхронизации времени от мастера (Рисунок 65).

Жёсткость	Временная отметка	Описание	Компонент
!	26.10.2022 15:23:04.183	Slave101[s101]: time sync: new time is set OK, local now is '2022-10-26 15:23:04.174000'	PsiIEC60870UsSlave101_OS
!	26.10.2022 15:23:04.183	Slave101[s101]: time sync: 2022-10-26 15:23:04.174(Local) format received	PsiIEC60870UsSlave101_OS
!	26.10.2022 15:23:03.995	Slave101[s101]: started	PsiIEC60870UsSlave101_OS
!	26.10.2022 15:23:03.995	Slave101[s101]: serial '#10 9600 8N1' is created and opened OK	PsiIEC60870UsSlave101_OS
!	26.10.2022 15:23:03.995	Master101[m101]: started	PsiIEC60870UpMaster101_OS

Рисунок 65 - Трассировка событий драйвера в журнале устройства

Дополнительную информацию о работе драйвера можно получить, просмотрев лог-файлы. Для загрузки файлов, перейдите на страницу параметров устройства во вкладку **Файлы**. В области **Исполнение** нажмите кнопку (**Обновить**). В окне отобразится дерево файлов, имеющих на контроллере. Перейдите в папку **logs**→**logger**→**user** (Рисунок 66). Имена лог-файлов имеют вид: *<iec60870driver\_\*.log>*. Скопируйте лог-файлы с контроллера для просмотра на ПК (кнопкой , из **Исполнение** в **Хост**).

Имя	Размер	Изменено
background.log	2,08 КБ (2 125 байт)	09.01.2022
crashreporter.log	1,61 КБ (1 648 байт)	09.01.2022
dhubm.log	33,29 КБ (34 092 байт)	09.01.2022
espd-default.log	46,91 КБ (48 036 байт)	09.01.2022
espd-redundancy.log	3,11 КБ (3 180 байт)	09.01.2022
ftp.users.log	6,27 КБ (6 425 байт)	09.01.2022
gmod.log	2,15 КБ (2 205 байт)	09.01.2022
ham.log	42,07 КБ (43 078 байт)	09.01.2022
hart_dhub_driver.log	3,14 КБ (3 212 байт)	09.01.2022
iec60870driver.log	4,59 МБ (4 817 060 байт)	09.01.2022
iec60870driver_20220808_142726.log	5,00 МБ (5 242 738 байт)	08.01.2022
iec60870driver_20220808_143534.log	5,00 МБ (5 242 775 байт)	08.01.2022
iec60870driver_20220808_144340.log	5,00 МБ (5 242 762 байт)	08.01.2022
iec60870driver_20220808_145149.log	5,00 МБ (5 242 753 байт)	08.01.2022
iec60870driver_20220808_145959.log	5,00 МБ (5 242 740 байт)	08.01.2022

Рисунок 66 - Дополнительные лог-файлы

Можно задать детальность трассировки в файлах с помощью свойства **LogLevel** экземпляра драйвера в процессе работы приложения:

```
Slave101.LogLevel:= PsiIEC60870Bridge.LOG_LEVEL_PDU_MAX;
```

Константы, включающие определенный уровень логирования, описаны в библиотеке **PsiIEC60870Bridge** (Рисунок 67). Константы можно объединить с помощью двоичного оператора ИЛИ (OR). Уровень, соответствующий **LOG\_LEVEL\_BASE**, включен всегда.

Имя	Тип	Наследовано от	Адрес	Начальн.	Комментарий
LOG_LEVEL_BASE	DWORD			16#0	base log level
LOG_LEVEL_PDU_SHORT	DWORD			16#1	short PDU trace log level
LOG_LEVEL_PDU_MAX	DWORD			16#2	full PDU trace log level
LOG_LEVEL_SPONT_DATA	DWORD			16#4	spontaneous data event log level
LOG_LEVEL_SPONT_VALUE	DWORD			16#8	spontaneous data values log level
LOG_LEVEL_CMD_DATA	DWORD			16#10	command data event log level
LOG_LEVEL_CMD_VALUE	DWORD			16#20	command data values log level
LOG_LEVEL_IG_DATA	DWORD			16#40	interrogation data event log level
LOG_LEVEL_IG_VALUE	DWORD			16#80	interrogation data values log level
LOG_LEVEL_PERIODIC_DATA	DWORD			16#100	periodic data event log level
LOG_LEVEL_PERIODIC_VALUE	DWORD			16#200	periodic data values log level
LOG_LEVEL_READ_DATA	DWORD			16#400	read request data event log level
LOG_LEVEL_READ_VALUE	DWORD			16#800	read request data values log level
LOG_LEVEL_ALL	DWORD			16#FFF	all log levels

Рисунок 67 - Константы

Все последующие настройки параметров и команд Slave 101 OS и приемы работы с ними идентичны тем, что описаны в разделе **Настройка контроллера в качестве Slave 104 OS**, начиная с подраздела [Создание списка элементов данных, передаваемых по IEC-104](#).

## Настройка контроллера в качества Master 101

### Добавление устройства Unbalanced Primary 101 Driver

Добавьте устройство **Unbalanced Primary 101 Driver** к последовательному порту **Regul Serial Port** или **Extended Regul Serial Port** (*Regul → МЭК 60870 → Serial 60870-101 Master → Unbalanced Primary 101 Driver*). Нажмите кнопку **Добавить устройство** или дважды щелкните левой кнопкой мыши. Выбранное устройство появится в проекте в дереве устройств.

Далее к устройству **Unbalanced Primary 101 Driver** необходимо подключить одно или несколько slave-устройств (outer slaves), которые будут опрашиваться контроллером: *Regul → МЭК 60870 → Serial 60870-101 Master → IEC 101 UP Outer Slave* (Рисунок 68).

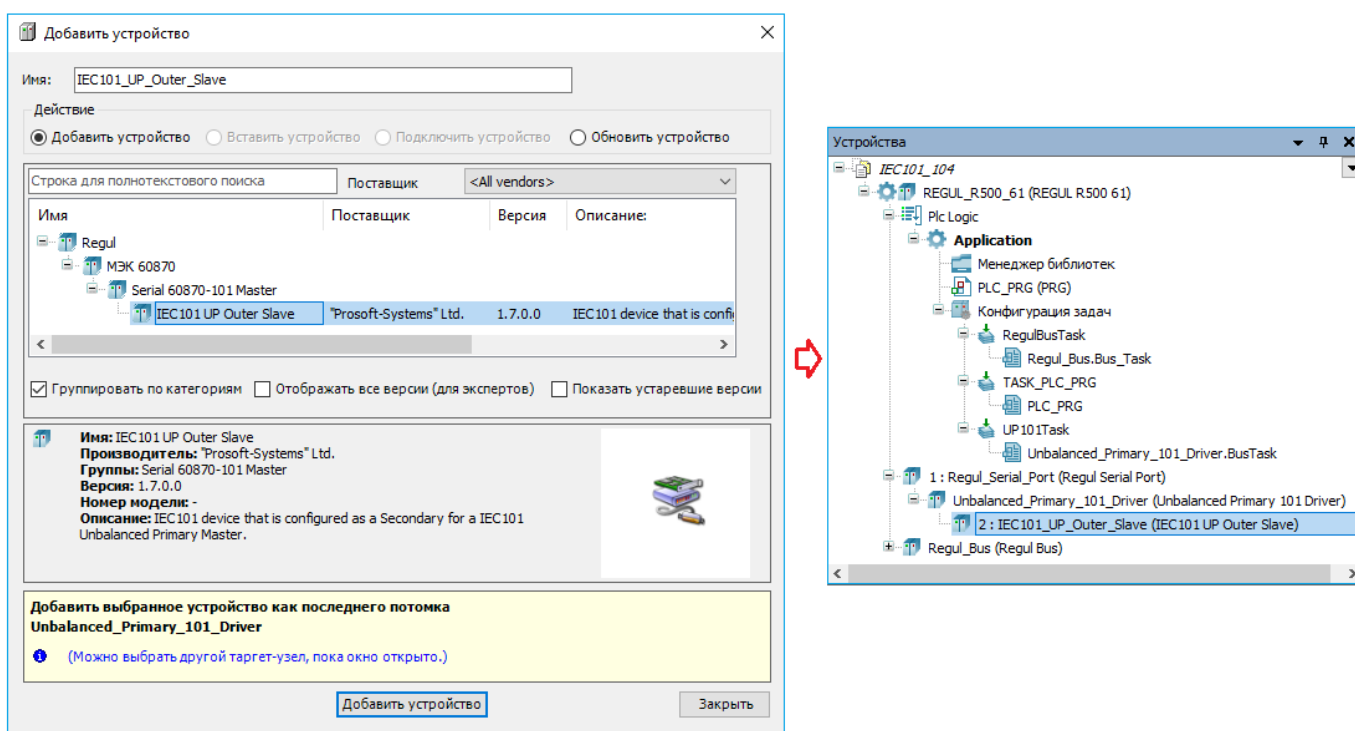


Рисунок 68 – Добавление в конфигурацию контроллера устройство IEC 101 UP Outer Slave

Двойным щелчком по названию устройства **IEC 101 UP Outer Slave** откройте вкладку параметров. По умолчанию открывается первая внутренняя вкладка **Редактор IEC 101 Outer Slave** (Рисунок 69).

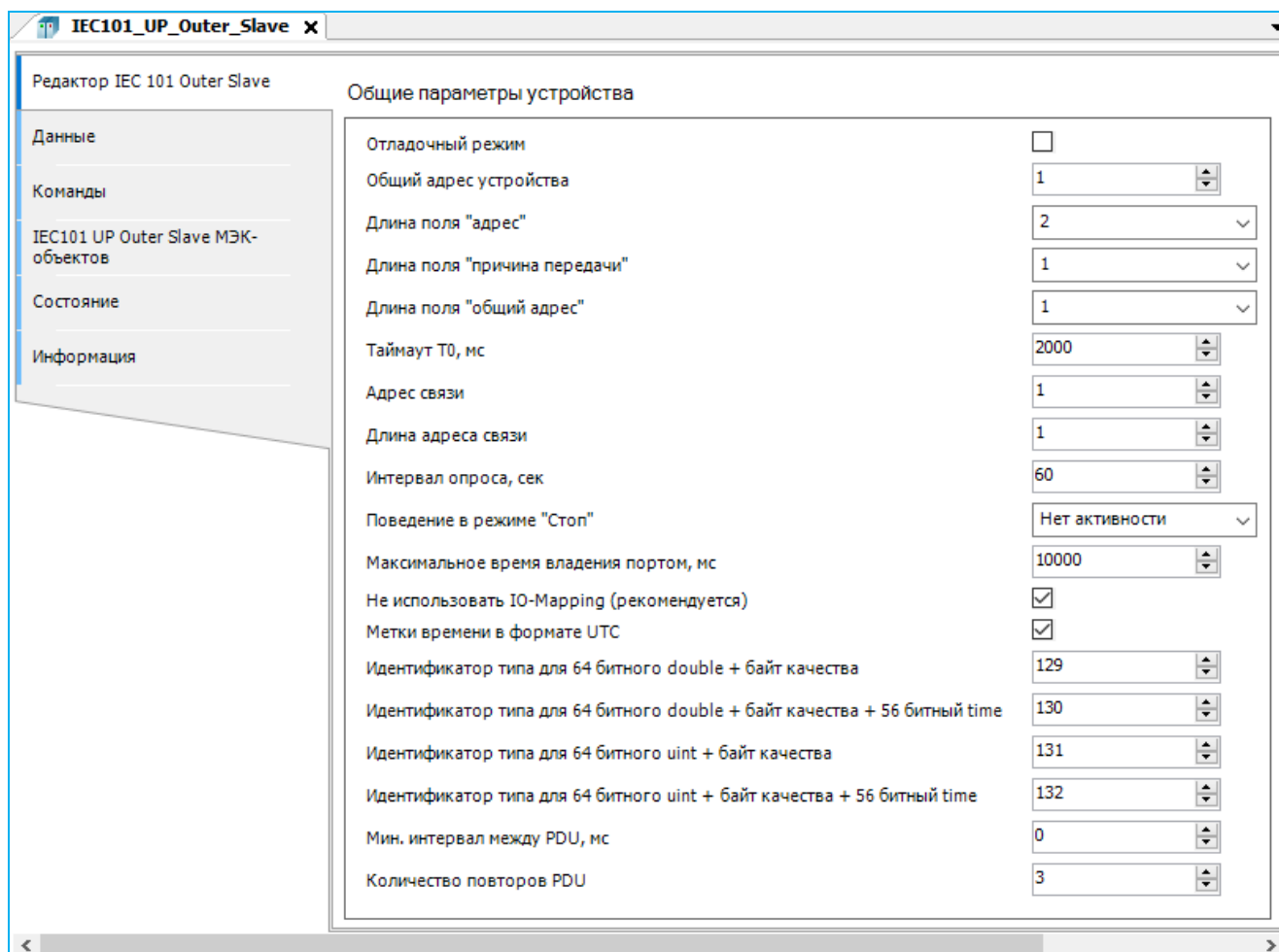


Рисунок 69 – Редактор устройства IEC 101 Outer Slave

### Настройка общих параметров устройства IEC 101 Outer Slave

В редакторе устройства (Рисунок 69) в блоке **Общие параметры устройства** доступны для настройки параметры, представленные в таблице 29.

Таблица 29 – Общие параметры устройства

Параметр	Описание	Значение по умолчанию
<b>Отладочный режим</b>	При установке флажка <input checked="" type="checkbox"/> в поле, все детали обмена по IEC-101 будут записаны в журнал работы контроллера, а именно передаваемые и принимаемые пакеты, комментарии к возникающим ошибкам и т.п.	
<b>Общий адрес устройства</b>	Содержит общий адрес (COMMON ADDRESS, ASDU ADDRESS) устройства, все данные и команды имеют общую часть, равную значению этого параметра (см. спецификацию IEC-101)	1
<b>Длина поля «адрес», Длина поля «общий адрес»</b>	Характеристики PDU (Protocol Data Unit – протокольная единица, пакет), передаваемых по IEC-101, должны быть одинаковы для пары master-slave. Допустимые значения параметра «адрес»: 2 или 3. Допустимые значения параметра «общий адрес»: 1 или 2	«адрес» – 2, «общий адрес» – 1

Параметр	Описание	Значение по умолчанию
Длина поля «причина передачи»	Содержит длину поля «причина передачи» (COT, Cause Of Transmission). Допустимые значения параметра: 1 или 2	1
Таймаут T0, мс	Интервал времени в миллисекундах. Определяет максимальную задержку ответа на запрос. Если задержка превышает данный интервал, то соединение считается потерянным	2000
Адрес связи	Уникальный адрес одного из подчиненных вторичных устройств, которое опрашивается первичным	1
Длина адреса связи	Длина поля уникального адреса, размер: от 1 до 65535 (1 или 2 байта)	1
Интервал опроса, сек	Период отсылки команды общего опроса. Значение 0 – отсылка команды общего опроса не производится	60
Поведение в режиме «Стоп»*	Определяет поведение компонента при остановке программы. <i>Нет активности</i> - означает, что Unbalanced Primary 101 Driver прекращает взаимодействовать с соответствующим IEC 101 UP Outer Slave; <i>Нормальная работа</i> - означает продолжение работы в обычном режиме	Нет активности
Максимальное время владения портом, мс	Максимальный интервал времени обмена данными с одним устройством, в миллисекундах	10000
Не использовать IO-Mapping (рекомендуется)	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что чтение/запись привязанных переменных происходит по адресу (прямой доступ к памяти ПЛК), без использования механизма I/O Mapping (соотнесение входов/выходов)	
Метки времени в формате UTC	Флажок <input checked="" type="checkbox"/> , установленный в поле, означает, что значение метки времени соответствует UTC, иначе - локального времени	
Идентификатор типа для 64 битного double + байт качества	Задание ID для пользовательского типа LREAL(double), без метки времени	129
Идентификатор типа для 64 битного double + байт качества + 56 битный time	Задание ID для пользовательского типа LREAL(double), с меткой времени	130
Идентификатор типа для 64 битного uint + байт качества	Задание ID для пользовательского типа ULINT(uint), без метки времени	131

Параметр	Описание	Значение по умолчанию
<b>Идентификатор типа для 64 битного uint + байт качества + 56 битный time</b>	Задание ID для пользовательского типа ULINT(uint), с меткой времени	132
<b>Мин. интервал между PDU, мс</b>	Определяет минимальный интервал времени между PDU, в случае, когда используется RS-485 (рекомендуемое значение для RS-485: 5...20)	0
<b>Количество повторов PDU</b>	Определяет количество повторов отправки запроса, в случае, когда ответ от вторичного устройства отсутствует, либо содержит ошибку	3



### ИНФОРМАЦИЯ

Для обзора всех параметров можно перейти на вкладку **IEC101 UP Outer Slave Конфигурация**, если вкладка отсутствует, то необходимо перейти в меню **Инструменты** ⇒ **Опции** ⇒ найти пункт **Редактор устройств** и установить флажок напротив поля **Показывать общие окна конфигурации устройств**

Все последующие настройки областей данных Master 101 и приемы работы с ними идентичны тем, что описаны в разделе **Настройка контроллера в качестве Master 104**, начиная с подраздела [Создание списка элементов данных, передаваемых по IEC-104](#).

Для получения статуса соединения Master IEC 101 (true, если соединение установлено, иначе – false) необходимо прописать, например:

```
IEC101_UP_Outer_Slave
IEC101_UP_Outer_Slave_1
-----
linked : BOOL;
linked2 : BOOL;
-----
linked := IEC101_UP_Outer_Slave.Linked();
linked2 := IEC101_UP_Outer_Slave_1.Linked();
```

## Настройка контроллера в качества Master 101 OS

### Добавление устройства Драйвер Unbalanced Primary Master 101 OS

Добавьте устройство **Драйвер Unbalanced Primary Master 101 OS** к последовательному порту **Regul Serial Port** или **Extended Regul Serial Port** (*Regul* → *МЭК 60870* → *Serial 60870-101 Master* → *Драйвер Unbalanced Primary Master 101 OS*). Нажмите кнопку **Добавить устройство** или дважды щелкните левой кнопкой мыши. Выбранное устройство появится в проекте в дереве устройств.

Далее к устройству **Драйвер Unbalanced Primary Master 101 OS** необходимо подключить одно или несколько slave-устройств (outer slaves), которые будут опрашиваться контроллером: *Regul* → *МЭК 60870* → *Serial 60870-101 Master* → *Драйвер OuterSlave 101* (Рисунок 70).

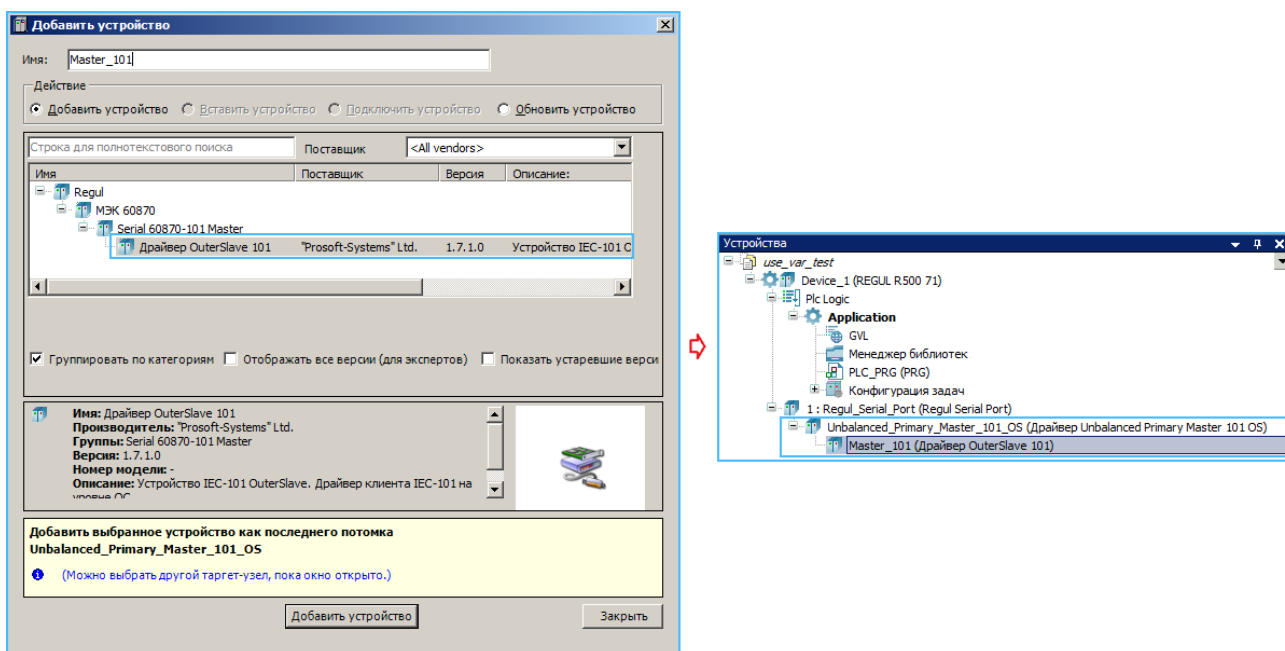


Рисунок 70 - Добавление в конфигурацию контроллера устройство Драйвер OuterSlave 101

Двойным щелчком по названию устройства **Драйвер Outer Slave 101** откройте вкладку параметров. По умолчанию открывается первая внутренняя вкладка **Редактор IEC 101 Outer Slave** (Рисунок 71).

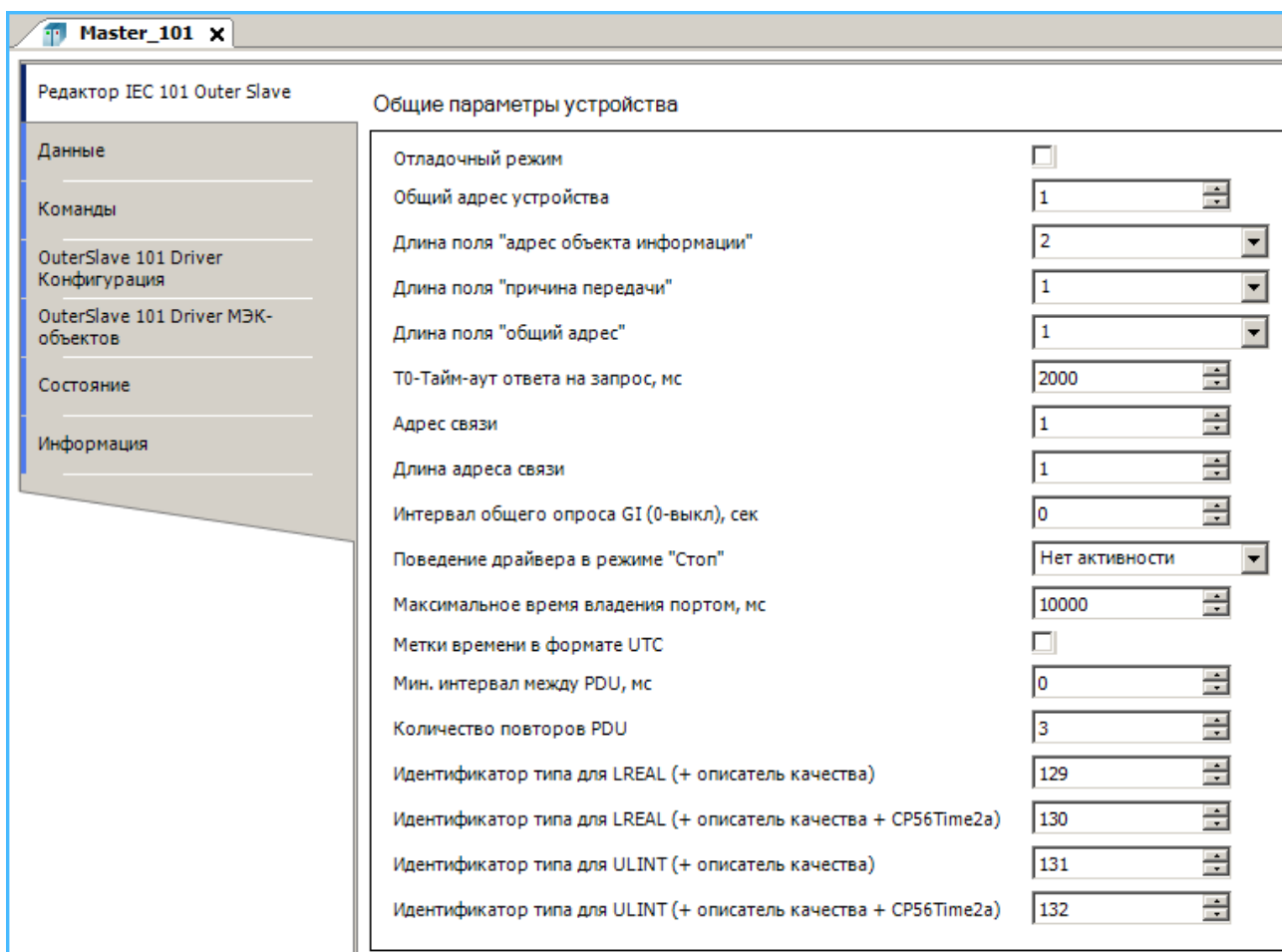


Рисунок 71 - Редактор устройства Драйвер OuterSlave 101

### Настройка общих параметров устройства Драйвер Outer Slave 101

В редакторе устройства (Рисунок 71) в блоке **Общие параметры устройства** доступны для настройки параметры, представленные в таблице 30.

Таблица 30 – Общие параметры устройства

Параметр	Описание	Значение по умолчанию
<b>Отладочный режим</b>	При установке флажка <input checked="" type="checkbox"/> в этом поле в журнал работы контроллера будут выводиться сообщения об ошибках, предупреждениях и ключевых моментах работы драйвера.	
<b>Общий адрес устройства</b>	Содержит общий адрес (COMMON ADDRESS, ASDU ADDRESS) устройства, все отдаваемые данные и выполняемые команды имеют общую часть, равную значению этого параметра (см. спецификацию IEC-104)	1
<b>Длина поля «адрес объекта информации»</b>	Содержит длину поля «адрес объекта информации» (IOA, Information Object Address). Допустимые значения параметра: 2 или 3	2
<b>Длина поля «причина передачи»</b>	Содержит длину поля «причина передачи» (COT, Cause Of Transmission). Допустимые значения параметра: 1 или 2	1



Параметр	Описание	Значение по умолчанию
Длина поля «общий адрес»	Содержит длину поля «общий адрес» (CA, Common Address). Допустимые значения параметра: 1 или 2	1
T0-Таймаут ответа на запрос, мс	Интервал времени в миллисекундах. Определяет максимальную задержку ответа на запрос. Если задержка превышает данный интервал, то соединение считается потерянным	2000
Адрес связи	Уникальный адрес для процедур обмена данными канального уровня	1
Длина адреса связи	Длина поля уникального адреса, размер: от 1 до 65535 (1 или 2 байта)	1
Интервал общего опроса GI (0-выкл), сек	Период отсылки команды общего опроса. Значение 0 – отсылка команды общего опроса не производится	0
Поведение драйвера в режиме «Стоп» *	Определяет поведение компонента при остановке программы. Нет активности (No activity) - означает, что устройство Master останавливает обмен Нормальная работа(Normal work) - означает продолжение работы в обычном режиме	Нет активности
Максимальное время владения портом, мс	Максимальный интервал времени обмена данными с одним устройством, в миллисекундах	10000
Метки времени в формат UTC	Если пользователь не задал метку времени перед отправкой команды, то она автоматически формируется с заданным форматом	
Мин. интервал между PDU, мс	Определяет минимальный интервал времени между PDU, в случае, когда используется RS-485 (рекомендуемое значение для RS-485: 5...20)	0
Количество повторов PDU	Определяет количество повторов отправки запроса, в случае, когда ответ от вторичного устройства отсутствует, либо содержит ошибку	3
Идентификатор типа для LREAL (+описатель качества)	Задание ID для пользовательского типа LREAL(double), без метки времени	129
Идентификатор типа для LREAL (+описатель качества +CP56Time2a)	Задание ID для пользовательского типа LREAL(double), с меткой времени	130
Идентификатор типа для ULINT (+описатель качества)	Задание ID для пользовательского типа ULINT(uint64), без метки времени	131

Параметр	Описание	Значение по умолчанию
<b>Идентификатор типа для ULINT (+описатель качества +CP56Time2a)</b>	Задание ID для пользовательского типа ULINT(uint64), с меткой времени	132



### ИНФОРМАЦИЯ

Для обзора всех параметров можно перейти на вкладку **OuterSlave 101 Конфигурация**, если вкладка отсутствует, то необходимо перейти в меню **Инструменты** ⇒ **Опции** ⇒ найти пункт **Редактор устройств** и установить флажок напротив поля **Показывать общие окна конфигурации устройств**

### Журналирование работы драйвера

Для просмотра журнала работы драйвера перейдите на основную страницу параметров устройства во вкладку **Журнал**, куда могут выводиться следующие сообщения драйвера OuterSlave 101 OS:

- инициализация, запуск и остановка работы драйвера (Рисунок 72).

Жёсткость	Временная отметка	Описание	Компонент
	10.08.2022 16:20:51.265	Master101[m_101]: stopped	PsIEC60870UpMaster101_OS
	10.08.2022 16:20:51.265	Master101[m_101]: STOP mode is ON, STOPBEHAVIOUR = NO_ACTIVITY	PsIEC60870UpMaster101_OS
	10.08.2022 16:20:51.240	m_101 : BusCycle : APP is STOPEd	PsIEC60870UpMaster101_OS
	10.08.2022 16:20:47.495	Master101[m_101]: started	PsIEC60870UpMaster101_OS
	10.08.2022 16:20:47.495	Master101[m_101]: STOP mode is OFF	PsIEC60870UpMaster101_OS
	10.08.2022 16:20:47.469	m_101 : BusCycle : APP is STARTED	PsIEC60870UpMaster101_OS
	10.08.2022 16:20:30.344	Master101[m_101]: stopped (STOP mode is ON, STOPBEHAVIOUR = NO_ACTIVITY)	PsIEC60870UpMaster101_OS
	10.08.2022 16:20:30.344	Master101[m_101]: '14' commands	PsIEC60870UpMaster101_OS
	10.08.2022 16:20:30.344	Master101[m_101]: '34' data params	PsIEC60870UpMaster101_OS
	10.08.2022 16:20:30.320	Master101[m_101]: serial '#2 9600 8N1' is opened	PsIEC60870UpMaster101_OS

Рисунок 72 - Трассировка событий драйвера в журнале устройства

Дополнительную информацию о работе драйвера можно получить, просмотрев лог-файлы. Для загрузки файлов, перейдите на страницу параметров устройства во вкладку **Файлы**. В области **Исполнение** нажмите кнопку (**Обновить**). В окне отобразится дерево файлов, имеющихся на контроллере. Перейдите в папку **logs**→**logger**→**user** (Рисунок 73) Имена лог-файлов имеют вид: «*iec60870driver\_\*.log*». Скопируйте лог-файлы с контроллера для просмотра на ПК (кнопкой , из **Исполнение** в **Хост**).

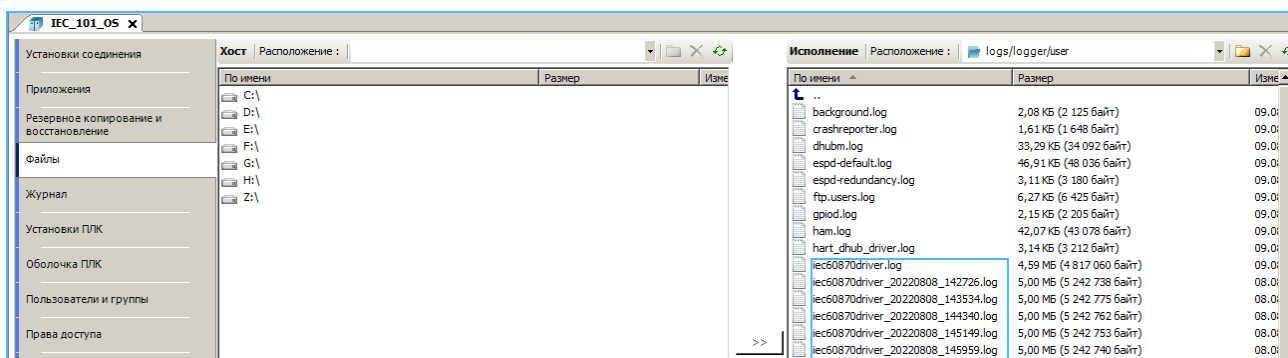


Рисунок 73 - Дополнительные лог-файлы

Можно задать детальность трассировки в файлах с помощью свойства **LogLevel** экземпляра драйвера в процессе работы приложения:

```
Master101.LogLevel:= PsIEC60870Bridge.LOG_LEVEL_PDU_MAX;
```

Константы, включающие определенный уровень логирования, описаны в библиотеке **PsIEC60870Bridge** (Рисунок 74). Константы можно объединить с помощью двоичного оператора ИЛИ (OR). Уровень, соответствующий LOG\_LEVEL\_BASE, включен всегда.

Имя	Тип	Наследовано от	Адрес	Начальн.	Комментарий
LOG_LEVEL_BASE	DWORD			16#0	base log level
LOG_LEVEL_PDU_SHORT	DWORD			16#1	short PDU trace log level
LOG_LEVEL_PDU_MAX	DWORD			16#2	full PDU trace log level
LOG_LEVEL_SPONT_DATA	DWORD			16#4	spontaneous data event log level
LOG_LEVEL_SPONT_VALUE	DWORD			16#8	spontaneous data values log level
LOG_LEVEL_CMD_DATA	DWORD			16#10	command data event log level
LOG_LEVEL_CMD_VALUE	DWORD			16#20	command data values log level
LOG_LEVEL_IG_DATA	DWORD			16#40	interrogation data event log level
LOG_LEVEL_IG_VALUE	DWORD			16#80	interrogation data values log level
LOG_LEVEL_PERIODIC_DATA	DWORD			16#100	periodic data event log level
LOG_LEVEL_PERIODIC_VALUE	DWORD			16#200	periodic data values log level
LOG_LEVEL_READ_DATA	DWORD			16#400	read request data event log level
LOG_LEVEL_READ_VALUE	DWORD			16#800	read request data values log level
LOG_LEVEL_ALL	DWORD			16#FFF	all log levels

Рисунок 74 - Константы

Все последующие настройки параметров и команд Master 101 OS и приемы работы с ними идентичны тем, что описаны в разделе Настройка контроллера в качестве Master 104 OS, начиная с подраздела [Создание списка элементов данных, передаваемых по IEC-104](#).

## ЭКСПОРТ И ИМПОРТ КОНФИГУРАЦИИ IEC 60870-5

Для упрощения создания списка каналов и их привязки к переменным программы предусмотрена возможность импорта/экспорта конфигурации IEC-104/ IEC-101.

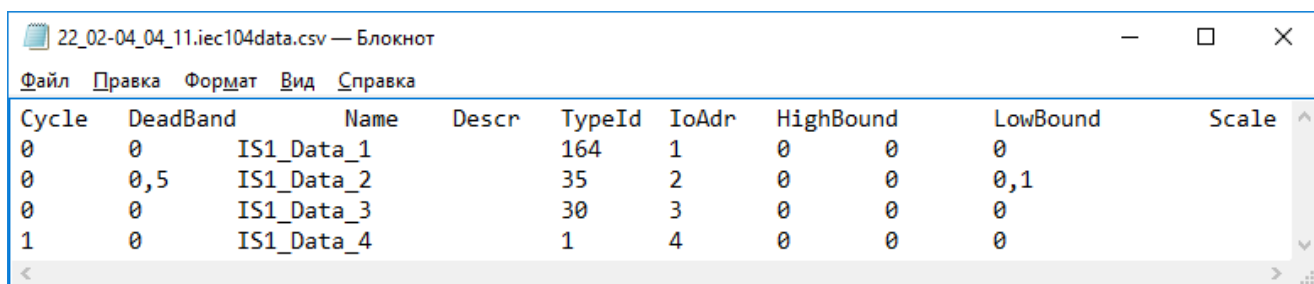
Конфигурацию контроллера в качестве Slave, экспортированную в файл, можно импортировать и использовать при настройке контроллера в качестве Master. При этом требуется следить за уникальностью имен сгенерированных переменных. В противном случае, т.е. когда имена переменных не уникальны, происходит следующее:

- при настройке контроллера в качестве Slave пользователь экспортирует список IEC-104/ IEC-101- переменных,
- затем в этом проекте пользователь настраивает контроллер в качестве Master и использует список переменных для импорта,
- в результате процедуры импорта ранее созданные переменные с теми же именами будут стерты, и заменены новыми, характерными для мастера.

### Экспорт/импорт списка элементов данных

В редакторе устройства на вкладке **Данные** создайте список элементов данных.

В нижней части окна в блоке **Тип файла** поставьте переключатель на значение  **CSV**. Нажмите кнопку **Экспорт**. Откроется окно **Save iec 104/101 settings**. Определите папку, в которой будет храниться файл конфигурации IEC-104(101), задайте ему имя, нажмите кнопку **Сохранить**. Информация будет сохранена в файл формата csv (текстовый формат), где все поля отделяются символом табуляции (ТАВ), первая строка обязательно представляет имена столбцов, все столбцы обязательны. Если в описании элемента данных для какого-либо столбца ничего не указано, то получается два символа ТАВ, идущие подряд.



Cycle	DeadBand	Name	Descr	TypeId	IoAdr	HighBound	LowBound	Scale
0	0	IS1_Data_1		164	1	0	0	0
0	0,5	IS1_Data_2		35	2	0	0	0,1
0	0	IS1_Data_3		30	3	0	0	0
1	0	IS1_Data_4		1	4	0	0	0

Рисунок 75 – Структура файла конфигурации для экспорта/импорта данных на примере IEC-104

В файле конфигурации присутствуют следующие поля:

- **Cycle** – участвует ли в циклической рассылке (значение 0 – нет, 1 – да);
- **DeadBand** – полоса нечувствительности (если применима к данному типу, значение 0 – не используется);

- **Name** – наименование канала (и соответствующей переменной, в режиме автоматической генерации);
- **Descr** – описание канала (недопустим символ табуляции в описании, пустое значение допустимо);
- **TypeId** – идентификатор типа согласно протоколу IEC-60870-5-104(101). Для типов с меткой времени можно указать автоматическую генерацию временных меток, прибавив к идентификатору типа 128 (80 hex);
- **IoAdr** – уникальный адрес элемента согласно протоколу IEC-60870-5-104(101);
- **HighBound** – верхняя граница для диапазонных типов данных (0 – не используется);
- **LowBound** – нижняя граница для диапазонных типов данных (0 – не используется);
- **Scale** – масштаб, для масштабируемых величин (0 – не используется).

Для импорта конфигурации из файла в проект выберите тип файла, далее нажмите кнопку **Импорт**. Откроется окно **Import iec 104/101 settings**. Выберите на компьютере нужный файл, нажмите кнопку **Открыть**. На вкладке **Данные** появится новый список элементов данных.

Файл с описанием элементов данных не обязательно должен быть создан в проекте и экспортирован. Пользователь может самостоятельно создать файл формата Tab delimited с помощью электронных таблиц, например, Microsoft Excel. В электронной таблице необходимо создать и заполнить поля в соответствии со структурой файла (описана выше). Далее сохранить файл с расширением csv.

	A	B	C	D	E	F	G	H	I
1	Cycle	DeadBand	Name	Descr	TypeId	IoAdr	HighBound	LowBound	Scale
2	0	0	IS1_Data_6		35	1	0	0	0
3	0	0,6	IS1_Data_7		30	2	0	0	0
4	1	0	IS1_Data_8		1	3	0	0	0
5	1	0	IS1_Data_9		164	4	0	0	0

Рисунок 76 – Создание в электронной таблице списка переменных на примере IEC-104 для последующего импорта в проект

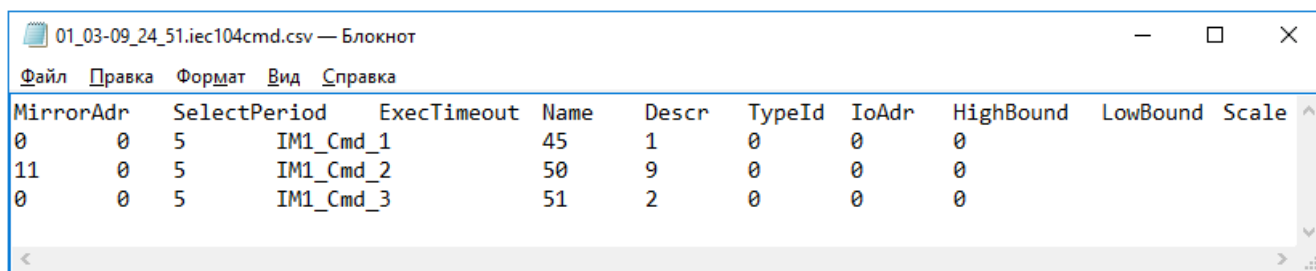
При экспорте конфигурации может быть выбран формат xml (в блоке **Тип файла** поставьте переключатель на значение  xml). Файл будет представлять собой правильный XML-документ, его структура логически будет такая же, как у файла Tab delimited, с учетом специфики xml-форматирования.

Импорт файла типа xml выполняется аналогично импорту файла типа csv.

## Экспорт/импорт списка команд

Сохранение списка команд в файл (как с расширением csv, так и с расширением xml), создание файлов конфигурации другими инструментами (вне проекта), импорт файлов выполняется так же, как и аналогичные действия со списком элементов данных. На рисунке 77 показана структура файла конфигурации для экспорта/импорта команд.

Максимальное количество команд для каждого устройства составляет 2000 элементов.



MirrorAdr	SelectPeriod	ExecTimeout	Name	Descr	TypeId	IoAdr	HighBound	LowBound	Scale
0	5	45	IM1_Cmd_1	1	0	0	0	0	
11	5	50	IM1_Cmd_2	9	0	0	0	0	
0	5	51	IM1_Cmd_3	2	0	0	0	0	

Рисунок 77 – Структура файла конфигурации для экспорта/импорта команд на примере IEC-104

В файле конфигурации присутствуют следующие поля:

- **MirrorAdr** – адрес элемента данных, в который будет помещено значение, переданное командой (0 – не используется);
- **SelectPeriod** – время блокировки для команд с функциональностью «Выборка перед выполнением» (0 – не используется);
- **ExecTimeout** – таймаут выполнения;
- **Name** – наименование канала (и соответствующей переменной, в режиме автоматической генерации);
- **Descr** – описание канала (недопустим символ табуляции в описании, пустое значение допустимо);
- **TypeId** – идентификатор типа согласно протоколу IEC-60870-5-104(101). Для типов с меткой времени можно указать автоматическую генерацию временных меток, прибавив к идентификатору типа 128 (80 hex);
- **IoAdr** – уникальный адрес элемента согласно протоколу IEC-60870-5-104(101);
- **HighBound** – верхняя граница для диапазонных типов данных (0 – не используется);
- **LowBound** – нижняя граница для диапазонных типов данных (0 – не используется);
- **Scale** – масштаб, для масштабируемых величин (0 – не используется).

## ТЕСТИРОВАНИЕ СИСТЕМЫ, ВЫЯВЛЕНИЕ НЕИСПРАВНОСТЕЙ И ИХ УСТРАНЕНИЕ

### Общие действия

После настройки IEC-104 Slave (на контроллерах серии Regul RX00) нередко возникает ситуация, когда в силу каких-либо причин IEC-104 Master, представленный SCADA-системой или иным сторонним приложением, не может подключиться к контроллеру по протоколу IEC-104. Основные вероятные причины (в порядке проверки):

- контроллер недоступен из подсети с клиентом IEC-104 Master;
- отсутствует возможность связаться с контроллером по порту, указанному в настройках IEC-104 Slave;
- неправильные настройки списка основных и резервных мастер-адресов IEC-104 Slave;
- ошибки в работе подключающегося клиента IEC-104 Master, несоответствие протоколу.

Доступность контроллера требуется проверить, если подсеть, в которой находится клиент IEC-104 Master, не совпадает с подсетью, к которой принадлежит компьютер (с установленным Astra.IDE), используемый для настройки контроллера. Для проверки используйте команду **ping** командного интерпретатора Windows, либо аналогичную команду в shell-терминале Linux-подобных операционных систем. Выполняется команда на компьютере с клиентом IEC-104 Master, укажите IP-адрес контроллера серии Regul RX00.

Пример успешного выполнения ping (командный интерпретатор Windows):

```
C:\>ping 172.29.22.240
Обмен пакетами с 172.29.22.240 по 32 байт:
Ответ от 172.29.22.151: число байт=32 время<1мс TTL=64
...
```

Пример – ping не выполнен:

```
C:\>ping 172.29.22.153
Обмен пакетами с 172.29.22.153 по 32 байт:
Превышен интервал ожидания для запроса.
...
```

Если ping до контроллера не выполняется, то следует провести диагностику сетевых подключений и настроек шлюзов (gateways) и т.п. В общем случае это выходит за рамки данного руководства. Возможно также, что на тестируемом компьютере запущен брандмауэр, в настройках которого заблокированы исходящие ICMP-пакеты. Следует изменить настройки брандмауэра, либо временно его отключить.

При наличии связи с контроллером проверьте возможность TCP-соединения с портом, указанным в настройках IEC-104 Slave, по умолчанию порт 2404. В качестве тестовой программы можно использовать Telnet-клиент, которая входит в поставку ОС семейства Windows, как опциональный компонент (можно установить через панель «Администрирование»).

Пример попытки подключения telnet (командный интерпретатор Windows) к IP 172.29.22.153, порт 2404:

```
C:\>telnet 172.29.22.153 2404
```

При сбое подключения появится следующее сообщение:

```
Подключение к 172.29.22.153...
Не удалось открыть подключение к этому узлу, на порт 2404: Сбой подключения
```

В такой ситуации следует подключиться к контроллеру (выполнить Login) и проверить состояние устройства **Slave 104 Driver** в дереве объектов. Значок «красный треугольник» показывает сбой в работе – в этом случае, скорее всего, потребуется обратиться в техподдержку, приложив журнал контроллера и файл проекта. Однако более вероятна ситуация, когда устройство **Slave 104 Driver** запущено и работает, но отсутствуют внешние подключения, и устройство помечено серым треугольником.

Если не удастся подключиться к требуемому порту контроллера, то следует проверить и изменить при необходимости настройки брандмауэра (правила для исходящих соединений) на тестируемом компьютере с клиентом IEC-104 Master. Если этот компьютер подключается к контроллеру из другой сети, то может потребоваться настройка маршрутизаторов.

При успешном подключении клиент Telnet покажет пустое окно терминала, без сообщений об ошибках. В программе Astra.IDE в журнале контроллера можно будет увидеть следующую запись от компонента IoDrvPs104Slave (Рисунок 78).

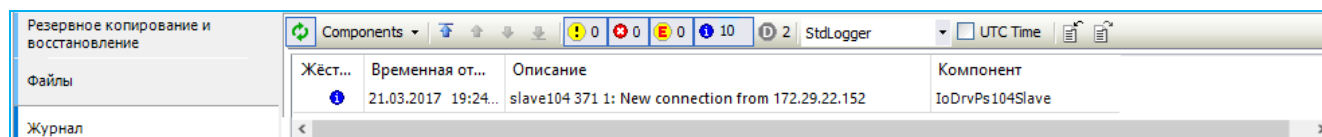


Рисунок 78 – Сообщение от компонента IoDrvPs104Slave

В окне дерева устройств состояние устройства **Slave 104 Driver** будет показано зеленым кружком, что означает, что устройство в работе и появились внешние подключения. Если, несмотря на это, клиент IEC-104 Master не получает данных, следует проверить наличие IP-адреса клиента в списке основных или резервных IP-адресов в настройках Slave 104 Driver и, при отсутствии, добавить. Если все настройки верны, но данных нет, то в рамках «стандартной» проверки предлагается использовать ПО «Программный шлюз-конвертор OPC IEC 60870-5-104 (МЭК-104)» (далее OPC-104) производства компании «РегЛаб». Демо-версия OPC-104 доступна по запросу. Программа принимает данные по IEC-104/101 и отдает по OPC DA, значения запрашиваемых параметров доступны для просмотра через интерфейс программы.



## Использование программы OPC-104 для тестирования

Предполагается, что в настройках Slave 104 Driver на контроллере задано несколько элементов данных и команд, как показано на рисунках (Рисунок 79, Рисунок 80).

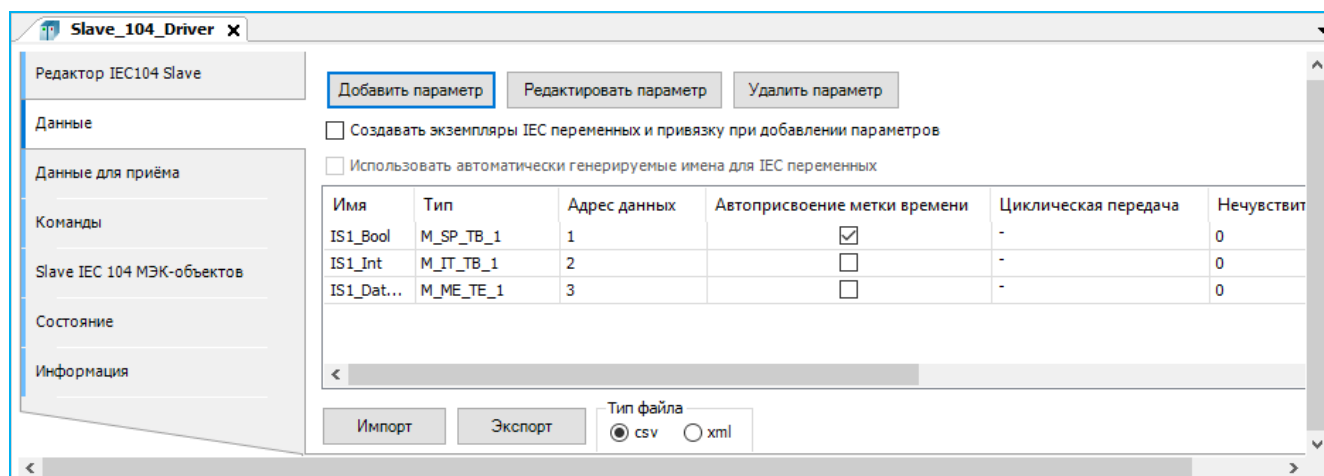


Рисунок 79 – Пример списка элементов данных

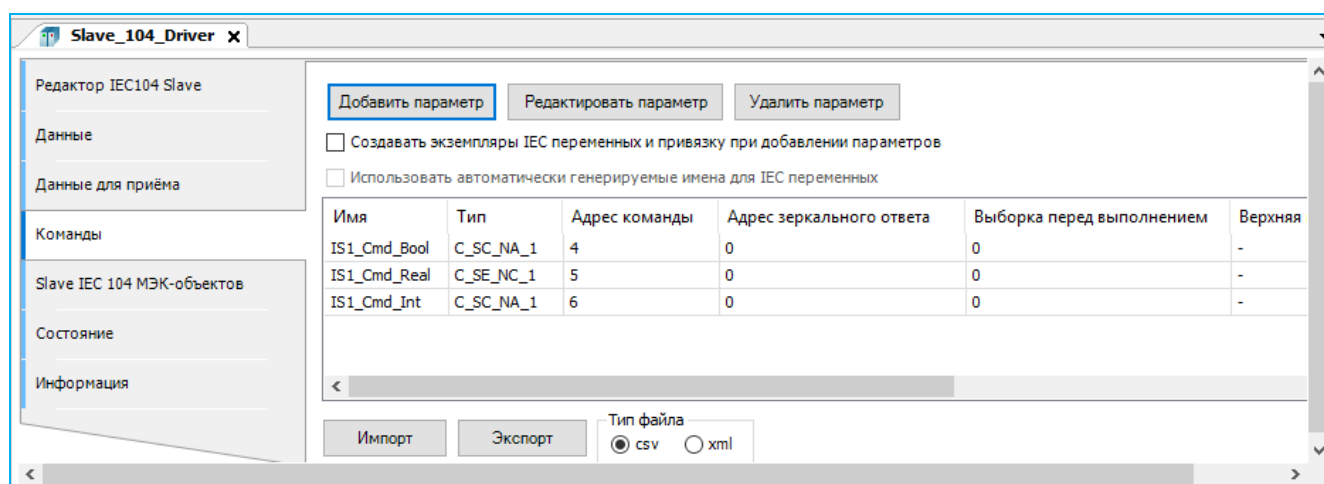


Рисунок 80 – Пример списка команд

Для обработки данных и команд в контроллере используется следующий программный код:

### Объявление

```
PROGRAM Test_IEC104
VAR
    //переменные для имитации данных МЭК-104
    bool_value : BOOL;
    int_value: DINT;
    real_value: REAL;
    //переменные для отображения принятых команд
    bool_cmd: BOOL;
    int_cmd: DINT;
    real_cmd: REAL;
    //для генерации дискретного значения
    timer_of_bool_value:TON:=(PT:=T#2S);
END_VAR
```

### Реализация

```
//генерация данных
```

```

timer_of_bool_value(IN:=TRUE);
IF (timer_of_bool_value.Q) THEN
    bool_value:= NOT bool_value;
    timer_of_bool_value(IN:=FALSE);
END_IF
int_value:=int_value+1;
real_value:=SIN (2*3.141592653*TIME_TO_REAL(TIME())/5000);
//добавление данных в очередь отправки IEC-104 Slave
IS1_Boolean.value:= BOOL_TO_WORD(bool_value);
IS1_Real.value:= real_value;
IS1_Int.value:= int_value;
//получение значений из команд, принятых IEC-104 Slave
bool_cmd := WORD_TO_BOOL(IS1_Cmd_Boolean.m_value.m_word);
int_cmd := IS1_Cmd_Int.m_value.m_dint;
real_cmd := IS1_Cmd_Real.m_value.m_float;
    
```

Для проверки связи с контроллером по протоколу IEC-104 установите на тестируемом компьютере ПО OPC-104, запустите программу (Рисунок 81).

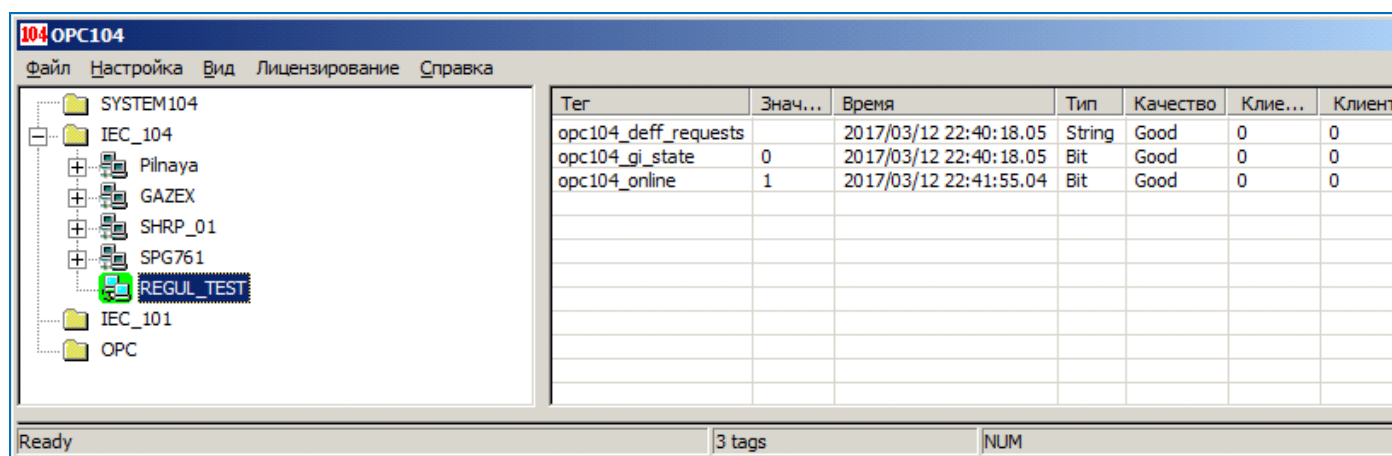


Рисунок 81 – Программный шлюз-конвертор OPC IEC 60870-5-104 (МЭК-104)

В дереве объектов выберите элемент IEC\_104, нажмите правую кнопку мыши и в появившемся контекстном меню выберите пункт **Добавить узел**. Укажите условное имя узла, IP-адрес контроллера, порт; установите флажок в поле **Active connection**, затем нажмите кнопку **ОК** (Рисунок 82). Предполагается, что остальные настройки узла соответствуют тем, которые заданы на контроллере.

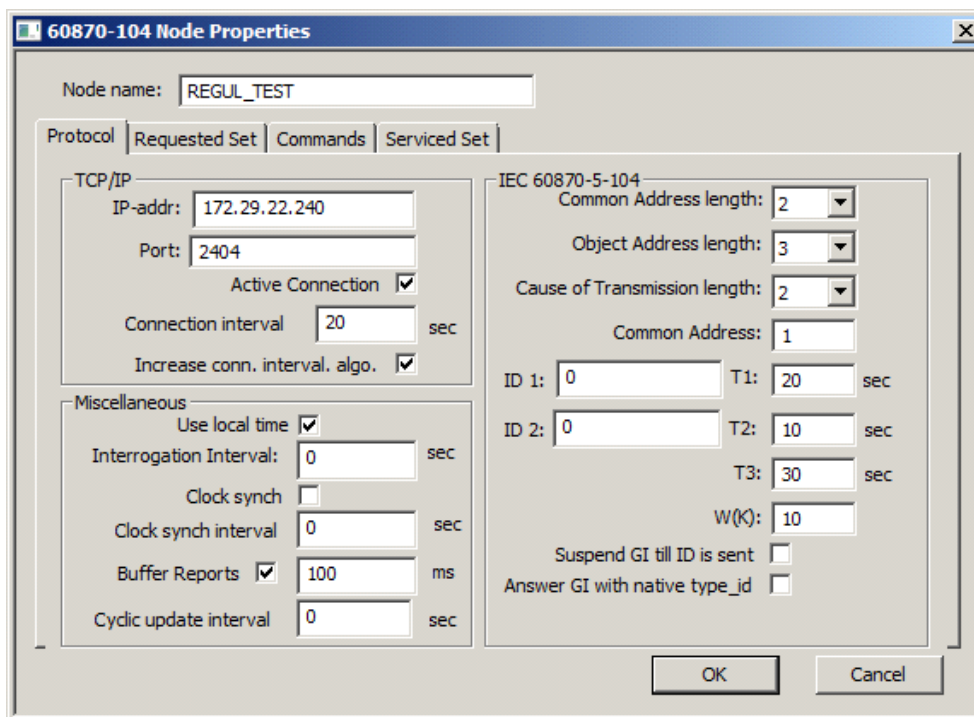


Рисунок 82 – Установка параметров узла

Далее опишите элементы данных и команды, соответствующие тем, что были созданы в контроллере. С помощью контекстного меню откройте окно **Node Properties**, перейдите на вкладку **Requested Set**. Опишите 3 элемента данных с адресами с 1 по 3 (Рисунок 83).

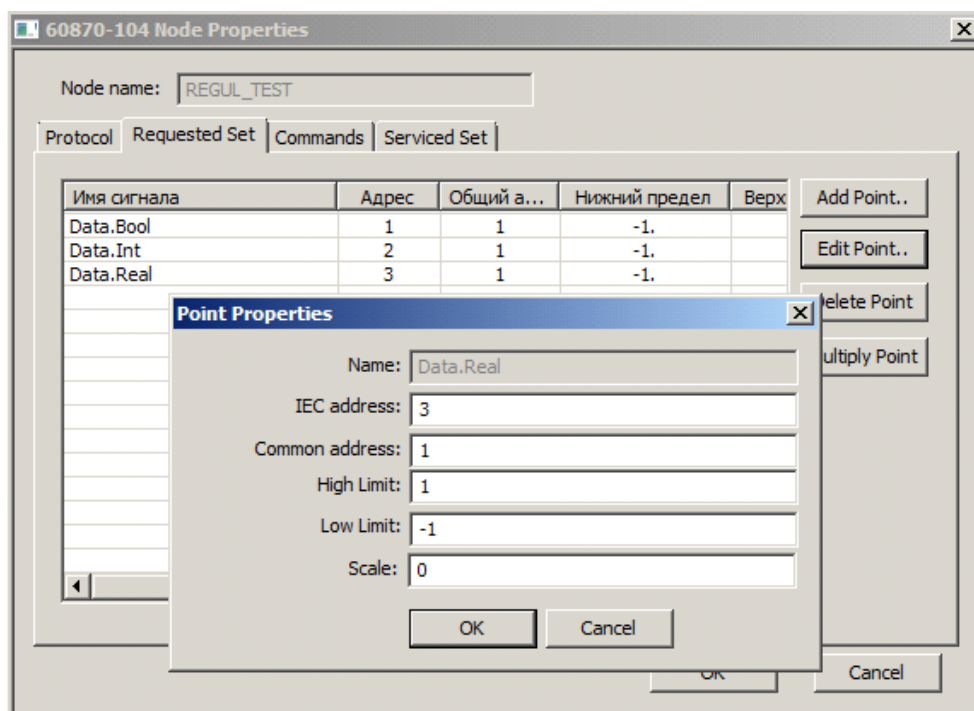


Рисунок 83 – Параметры элемента данных

Перейдите на вкладку **Commands**, где опишите соответствующие команды – для каждой команды нужно указать тип и адрес (Рисунок 84). Типы и адреса соответствуют командам, описанным в контроллере.

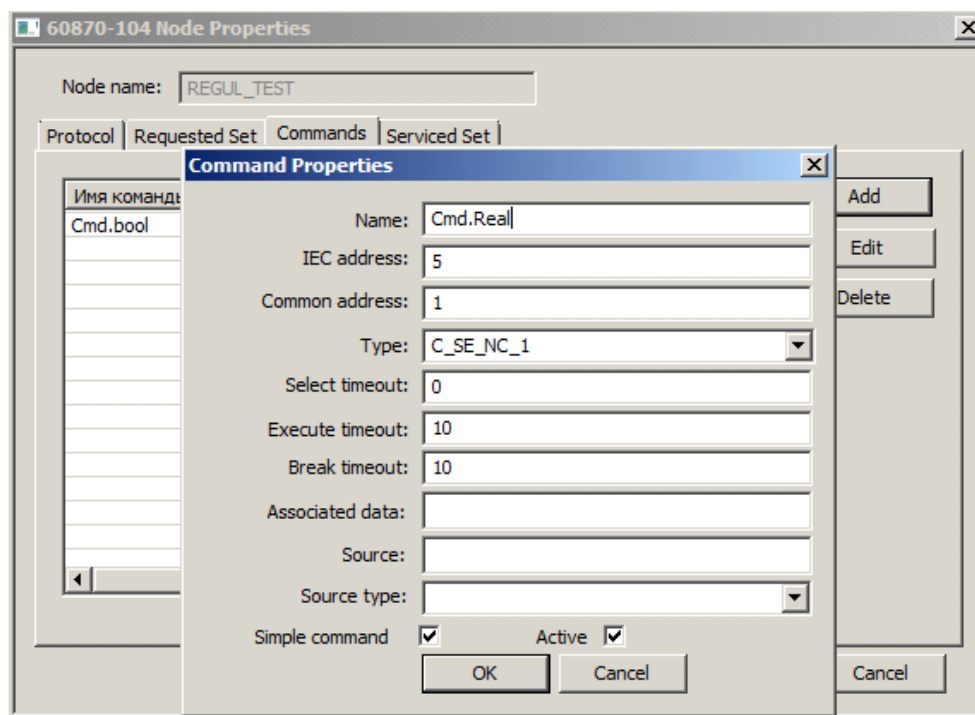


Рисунок 84 – Параметры команды

После окончания настроек нажмите кнопку **OK**.

Для запуска тестирования выберите в контекстном меню узла пункт **Открыть**. Пиктограмма узла окрасится зеленым цветом. При раскрытии списка тегов узла становится видно, что с контроллера приходят данные.

Для тестирования команды укажите на нее в списке, нажмите клавишу **Enter**. Появится форма для ввода нового значения, где укажите значение, далее нажмите клавишу **Enter**. Значение должно быть записано на контроллер.

Если, несмотря на правильные настройки, вы не смогли добиться связи с контроллером, то следует обратиться в техподдержку.

## ОБРАЩЕНИЕ В СЛУЖБУ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ

Для обращения в техническую поддержку Пользователю необходимо сформировать запрос на сайте технической поддержки: <https://support.prosoftsystems.ru>, либо отправить письмо по электронной почте: support@prosoftsystems.ru. В первом случае требуется предварительная регистрация.

Обращение обязательно должно содержать следующие сведения:

- подробное описание сложившейся ситуации;
- наименование объекта и его месторасположение;
- наименование системы автоматизации;
- модель ПЛК;
- серийный номер ПЛК;
- версия пакета обновления для среды разработки Astra.IDE;
- версия СПО-контроллера;
- архив с лог-файлами (см. документ «Astra.IDE User Guide DPA 302. Раздел «Журнал событий»);
- архив с лог-файлами, включающими в себя период времени, когда произошел отказ;
- дата и время возникновения отказа. А также периодичность и устойчивость повторения подобных отказов в случае, если такая информация имеется.

Желательно прислать проект для Astra.IDE, так как это может значительно упростить и ускорить процесс поиска причины отказа.

Для того, чтобы узнать, как получить необходимую информацию (сведений о версии Astra.IDE, версии СПО и так далее), ознакомьтесь с содержимым документа «Astra.IDE User Guide DPA 302».