



Среда разработки MasterSCADA 4D

Работа с ПЛК и модулями RealLab в среде разработки MasterSCADA 4D:
создание, настройка и запуск проектов

Программирование ПЛК RealLab в среде разработки MasterSCADA 4D

Руководство пользователя

© НИЛ АП, 2024

Версия от 15 июля 2024 г.

Одной проблемой стало меньше!

Уважаемый покупатель!

Научно-исследовательская лаборатория автоматизации проектирования (НИЛ АП) благодарит Вас за использование нашей продукции и просит сообщать нам свои пожелания по улучшению программного обеспечения. Ваши пожелания можно направлять по почтовому или электронному адресу, а также сообщать по телефону:

НИЛ АП, пер. Биржевой спуск, 8, Таганрог, 347900,

Тел. (495) 26-66-700,

e-mail: info@reallab.ru, <https://www.reallab.ru>.

Вы можете также получить консультации по применению нашей продукции, воспользовавшись указанными выше координатами.

Авторские права на программное обеспечение, модуль и настоящее руководство принадлежат НИЛ АП.

Оглавление

1. Вводная часть.....	6
1.1. Введение	6
1.2. Начало работы.....	7
1.3. Программируемые контроллеры RealLab в MasterSCADA 4D	7
2. Установка и настройка IDE MasterSCADA 4D	8
2.1. Системные требования.....	8
2.2. Установка MasterSCADA 4D	9
2.3. Работа над проектом.....	10
2.3.1. Создание проекта.....	10
2.3.2. Подключение библиотеки RealLab	12
Добавление устройств и протоколов в проект.....	14
2.3.3. Работа с модулями в проекте.....	15
2.3.4. Работа с объектами в проекте.....	21
2.3.5. Формирование задач узлов	27
2.3.6. Компиляция проекта и загрузка конфигураций в узлы.....	28
2.3.7. Приоритет выполнения задачи	30
2.4. Подключение к ПЛК	31
2.5. Настройка статического IP-адреса ПЛК.....	31
2.5.1. Настройка статического IP-адреса ПЛК серии NLScon-A40.	32
2.5.2. Настройка статического IP-адреса ПЛК серии NLScon-RSB	33
2.6. Работа с Retain-переменными.....	34
2.7. Работа светодиодов и переключателя Run-Stop на ПЛК.....	36
2.8. Работа с визуализацией	37
2.8.1. Добавление окна	37
2.8.2. Добавление объектов в окно визуализации	38

2.8.3. Организация обмена данными между параметрами и визуализацией	39
2.8.4. Формат отображаемых данных	41
2.8.5. Вызов программ по кнопке из окна визуализации	42
2.8.6. Добавление картинки в визуализацию	42
2.8.7. Добавление экранной клавиатуры в визуализацию	44
2.9. OPC UA Server	46
2.10. Диагностика контроллера	50
2.10.1. Стандартная диагностика	50
2.10.2. Диагностика по TCP/IP	51
2.10.3. Специальные ФБ.....	54
3. Работа с модулями ввода-вывода	55
3.1. Рекомендации для работы с ПЛК и модулями серии NL, NLS	55
3.2. Протокол Modbus RTU.....	56
3.2.1. Пример использования NLS-16DI.....	57
3.2.2. Пример использования NLS-16DO	58
3.2.3. Пример использования NLS-8R	59
3.2.4. Пример использования NLS-4AO	60
3.2.5. Пример использования NLS-8TI.....	61
3.2.6. Пример использования NLS-8AI-I.....	62
3.2.7. Пример использования NLS-4RTD	63
3.2.8. Пример использования NLS-4C	64
3.3. Протокол Modbus TCP	66
3.3.1. Пример использования NLS-16DI-Ethernet.....	67
3.3.2. Пример использования NLS-16DO-Ethernet	68
3.3.3. Пример использования NLS-8R-Ethernet	69
3.3.4. Пример использования NLS-4RTD-Ethernet	70
3.3.5. Пример использования NLS-8AI-Ethernet.....	71

3.3.6. Пример использования NLS-16AI-I-Ethernet	72
3.3.7. Пример использования NLS-8TI-Ethernet	74
3.3.8. Пример использования NLS-4AO-Ethernet	75

1. Вводная часть

1.1. Введение

MasterSCADA 4D – система для создания АСУТП, MES, решения задач учета и диспетчеризации объектов промышленности, ЖКХ и автоматизации зданий.

Единая среда разработки SCADA системы

Разработка всех элементов проекта MasterSCADA ведется в единой инструментальной среде. Это и разработка аппаратной архитектуры системы, и проектирование алгоритмов обработки, описание логической структуры и базы параметров, настройка первичной обработки, разработка системы визуализации, отчетов и журналов, и прочих компонентов системы.

Такой подход, когда все инструменты находятся под рукой, а проект представляет единое информационное пространство, позволяет легко решать следующие задачи:

- Решить проблемы программной стыковки различных устройств системы управления;
- Перераспределять сигналы или алгоритмы их обработки по отдельным устройствам;
- Создавать распределенные по устройствам алгоритмы контроля и управления;
- Иметь доступ с любого рабочего места к любой информации, имеющейся в системе.

Двухслойная структура проекта

В MasterSCADA проект разделен на два слоя.

Слой описания аппаратной структуры проекта. Здесь описываются рабочие станции, контроллеры и модули ввода-вывода, входящие в проект. Задаются их свойства и настраиваются связи между ними;

Слой описания логики проекта. Здесь описывается логическая структура проекта: какие технологические объекты автоматизируются в данном проекте, их свойства, параметры, алгоритмы управления и мнемосхемы. Кроме того, здесь разрабатываются отчеты, настраиваются журналы и параметры информационной безопасности системы.

Такой подход дает возможность:

- Разрабатывать эти структуры параллельно;
- Независимо работать специалистам различных профилей;
- Решить проблему перехода от одной технической структуры системы к другой (например, реализовать тот же проект на базе других контроллеров).

Объектный подход

Объект в MasterSCADA – это основная единица разрабатываемой системы, соответствующая реальному технологическому объекту (цеху, участку, аппарату, насосу, задвижке, датчику и т.п.), управляемому разрабатываемой с помощью MasterSCADA системой. С другой стороны, это и традиционный с точки зрения программирования объект, обладающий стандартными для программных объектов качествами.

1.2. Начало работы

Чтобы начать работу с MasterSCADA 4D, скачайте и установите среду разработки. Запустите MasterSCADA 4D, создайте проект, скачайте и импортируйте библиотеку с шаблонами модулей и ПЛК RealLab. Добавьте в проект необходимые ПЛК, модули, объекты и подключите нужные Вам библиотеки. Создайте программу на одном из языков МЭК или на языке C#. Соединитесь с ПЛК. После этого все будет готово для создания полноценного проекта.

В состав MasterSCADA 4D включен ряд демо-проектов. Доступ к ним можно получить со стартовой страницы редактора проектов либо с вкладки **Учебные** диалога открытия проектов. Также на нашем сайте размещены примеры работы с модулями RealLab в среде MasterSCADA 4D.

1.3. Программируемые контроллеры RealLab в MasterSCADA 4D

Ниже приведен список контроллеров RealLab, поддерживающих программирование в среде разработке MasterSCADA 4D (версия не ниже 1.3.5).

- NLScon-RSB-S
- NLScon-A40-S

2. Установка и настройка IDE MasterSCADA 4D

2.1. Системные требования

Минимальные системные требования:

- Операционная система (ОС) – Windows 10 SP1 x64 с поддержкой русского языка;
- Процессор - Intel® Core™ i3, 2.3 ГГц;
- ОЗУ – 8 Гб;
- Дисплей – 1280x1024;
- Жесткий диск – 10 Гб свободного пространства;
- Клавиатура, мышь.

Рекомендуемые системные требования:

- Windows 10 x64 или старше;
- Процессор – современный многоядерный, не ниже Intel® Core™ i5, 3.4 ГГц;
- ОЗУ – не менее 16 Гб;
- Дискретная видеокарта, с актуальными драйверами (не старше 5 лет);
- Дисплей – 1920x1080;
- Жесткий диск – SSD – 100 Гб свободного пространства;
- Клавиатура, мышь.

ВАЖНО! При соблюдении минимальных системных требований среда разработки MasterSCADA 4D может быть установлена, но при разработке проектов может наблюдаться торможение при переключении между вкладками, при работе с деревьями и панелями редактора. Время выполнения длительных операций, например, конвертации проектов, может быть значительным.

2.2. Установка MasterSCADA 4D

Скачайте установщик последней версии MasterSCADA 4D с нашего сайта, запустите его и следуйте подсказкам установщика MasterSCADA.

При запуске установщика откроется начальный диалог мастера установки.

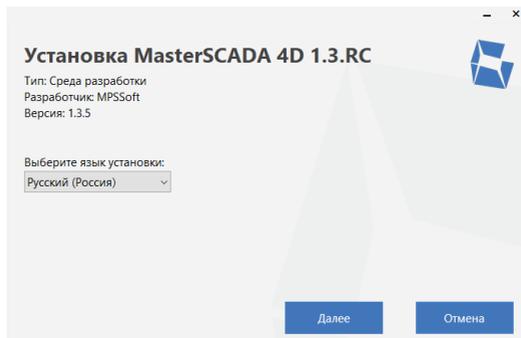


Рис. 2.1. Мастер по установке MasterSCADA 4D

В диалоге выбирается локализация продукта (поддерживаются русский и английский языки). По команде **Далее** мастера открывается диалог лицензионного соглашения. Прочитав текст, отметьте пункт “Я принимаю условия лицензионного соглашения” и нажмите кнопку “**Далее**”. После этого откроется диалог с выбором каталога установки. Выберите каталог для установки программы и нажмите “**Далее**”.

ВАЖНО! При выборе пути установки следует избегать папок, названия которых состоят из кириллических символов.

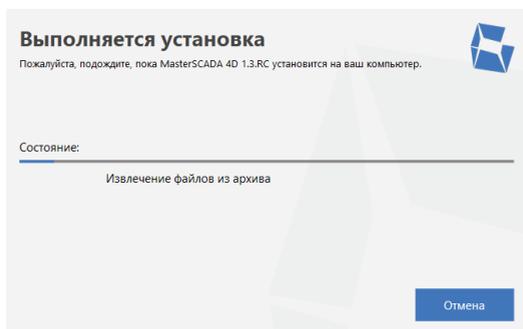


Рис. 2.2. Процесс установки MasterSCADA 4D

После того, как программа будет установлена, ее можно сразу же запустить и начать работу со средой разработки.

2.3. Работа над проектом

Ниже описан процесс создания проекта в MasterSCADA 4D и его настройки для взаимодействия с модулями RealLab. Для упрощения процесса добавления и настройки ПЛК и модулей, была создана библиотека, которую достаточно импортировать в среду разработки. Библиотека RealLab содержит элементы, необходимые для конфигурирования обмена данными с оборудованием RealLab. Вы можете загрузить её с [нашего сайта](#).

2.3.1. Создание проекта

Запустите среду разработки MasterSCADA 4D. Чтобы открыть готовый проект или демо-проект, нажмите “Открыть проект”. Нажатие на кнопку “С чего начать?” вызовет дополнительные разделы “Быстрый старт”, “Справочный раздел”, “Демонстрационные проекты”.

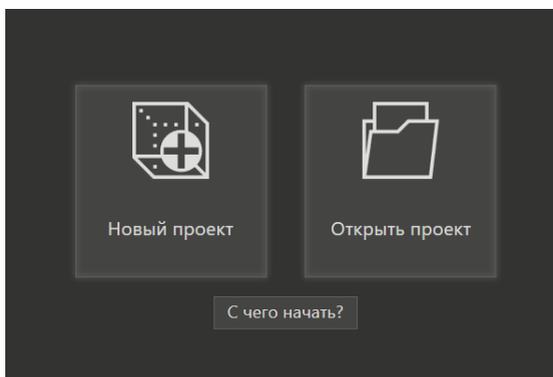


Рис. 2.3. Главный экран MasterSCADA 4D

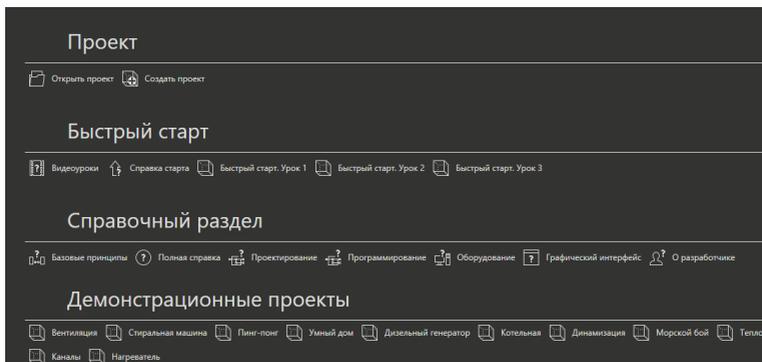


Рис. 2.4. Вкладка "С чего начать?"

Чтобы создать проект нажмите на кнопку “Новый проект” – появится диалог создания. Введите название проекта, выберите хранилище (Firebird или PostgreSQL), тип сервера (встроенный или внешний) и укажите путь к хранилищу. Далее нажмите “Создать проект”.

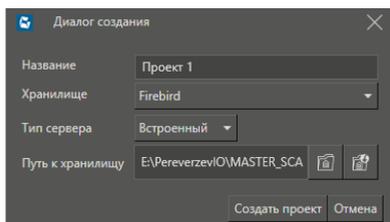


Рис. 2.5. Диалог создания

После создания проекта откроется интерфейс работы с ним. Слева расположено дерево системы (1), справа – свойства выбранного элемента/модуля/параметра (2), основную часть экрана (3) занимает окно стартовой страницы, которое в дальнейшем заменяется на окно визуализации/окно отладки/окно каналов устройства и т.д., вверху располагается, снизу – палитра/легенда (5).

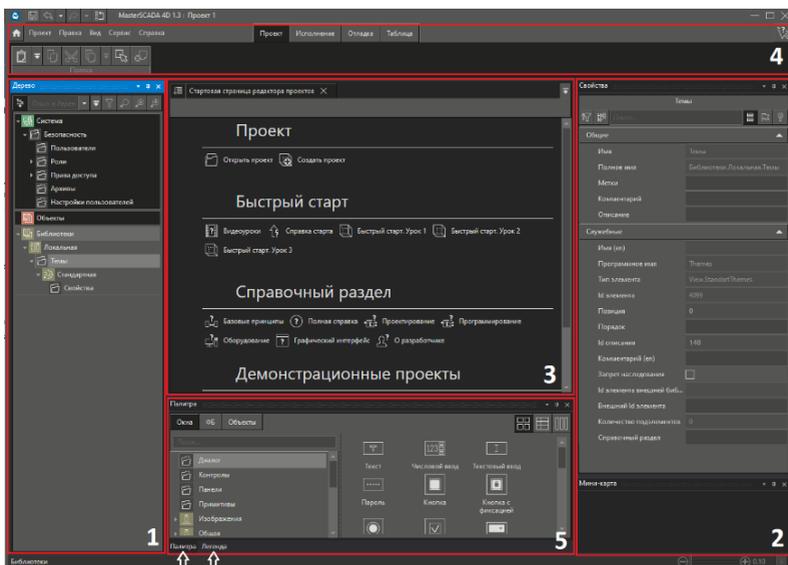


Рис. 2.6. Интерфейс программы при настройке проекта

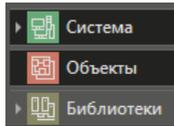


Рис. 2.7. Элементы дерева

В дереве системы задается физическая конфигурация Вашей системы – указываются ваши ПЛК, модули, прочие устройства, их каналы связи и т.д.

В дереве объектов создается логическая структура проекта. По умолчанию, оно не содержит дочерних элементов.

Дерево библиотек содержит библиотеки, как поставляемые со средой разработки, так и созданные пользователем (локальные).

2.3.2. Подключение библиотеки RealLab

Для того, чтобы импортировать библиотеку в MasterSCADA 4D, откройте любой проект в MasterSCADA 4D. В верхнем меню выберите “Проект” – “Импортировать” – “Библиотеку”.

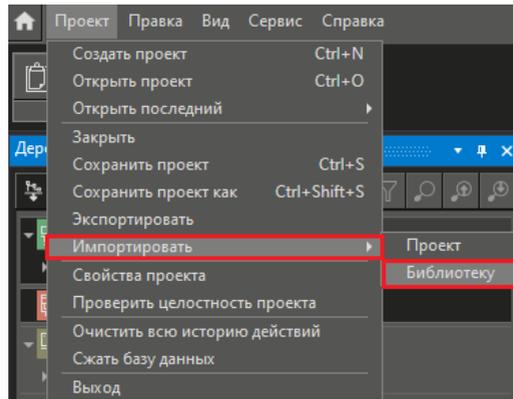


Рис. 2.8. Меню импорта библиотек

В появившемся окне укажите путь к архиву библиотеки, которую собираетесь импортировать. Нажмите на название библиотеки, а затем на кнопку “Импортировать библиотеку”.

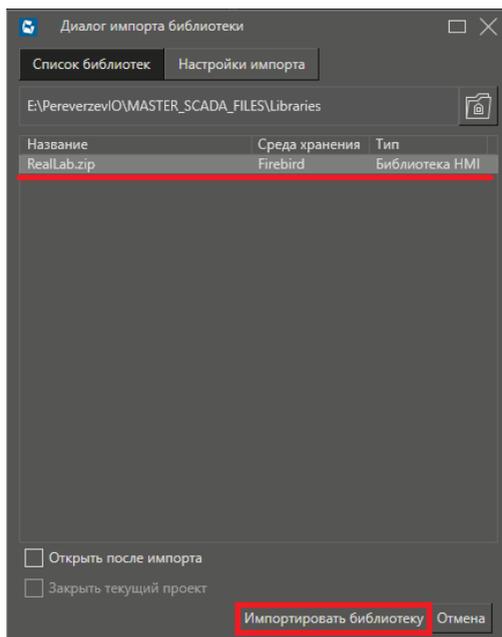


Рис. 2.9. Выбор библиотеки для импорта

После того, как импорт завершится, вы можете подключить библиотеку к Вашему проекту. Для этого откройте проект, к которому нужно подключить библиотеку. В дереве библиотек нажмите правой кнопкой мыши на **“Библиотеки”** – **“Подключить библиотеку”** – **“RealLab”** – **RealLab**. После подключения вы сможете воспользоваться шаблонами модулей Modbus TCP и Modbus RTU и шаблоном ПЛК RealLab.

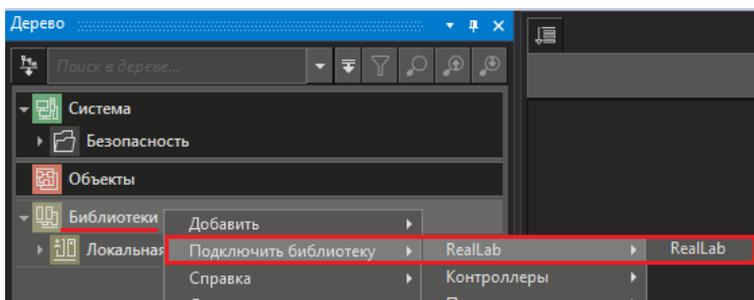


Рис. 2.10. Подключение библиотеки к проекту

Добавление устройств и протоколов в проект

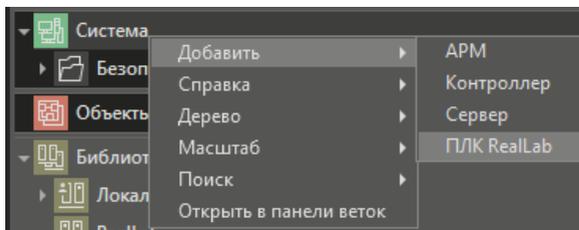


Рис. 2.11. Добавление ПЛК из библиотеки

Затем добавьте протокол – правой кнопкой мыши нажмите в дереве устройств на **“Протоколы”** – **“Добавить”** – **“Modbus RTU/TCP”**.

В свойствах добавленного протокола укажите порт подключения модулей, скорость передачи данных, четность, стоп-бит, разрядность и таймаут опроса модулей в случае с Modbus RTU. В случае с Modbus TCP можно таймаут опроса модулей (необязательно). Также можно выбрать способ записи параметров – по изменению, периодически и т.д.

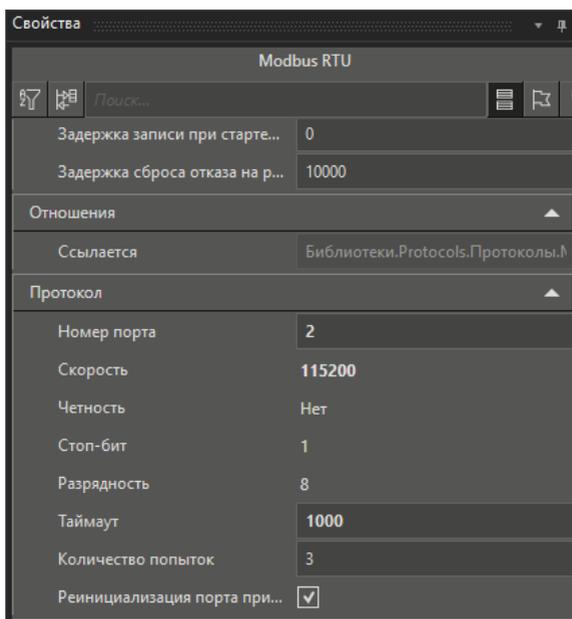


Рис. 2.12. Пример свойств Modbus RTU

В свойствах обоих протоколов можно установить разрешение доступа к устройствам и их регистрам через OPC UA Server.

Теперь можно добавлять модули.

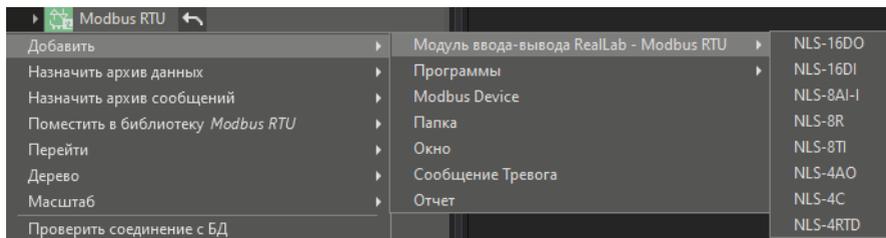


Рис. 2.13. Добавление модулей Modbus RTU

2.3.3. Работа с модулями в проекте

После добавления нужного модуля укажите в его свойствах его ID, если это модуль Modbus RTU. Для модуля Modbus TCP достаточно указать IP-адрес. В свойствах Modbus RTU/TCP модулей также можно выбрать использование функций записи 05 (Single Coil) и 06 (Single Register). Установка галочки напротив названия функции означает, что все регистры модуля, подходящие под область работы данной функции, будут записываться раздельно с помощью данной функции.

Т.е. Если установить галочку, например, на функцию 06 и попытаться записать два регистра из области HOLDING_REGISTERS одновременно, появится ошибка, т.к. более одного регистра можно записать только функциями 15 и 16 для Coils и Holding Registers соответственно.

ПЛК RealLab 1 Modbus RTU 1 : Modbus Tag ID 82727, при выключенной групповой записи нельзя писать типы длиннее одного регистра

Рис.2.14. Ошибка записи нескольких регистров одновременно

ВАЖНО! Не все одиночные регистры можно записать функцией 16 (функцией записи нескольких регистров одновременно).

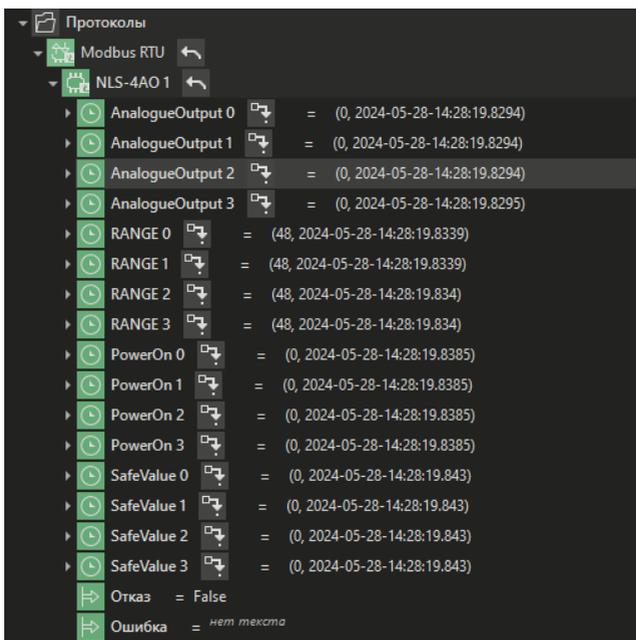


Рис. 2.15. Пример работы модуля NLS-4AO

Ниже находится более подробная информация по работе с модулями, каналами и т.д.

Как работать с каналами модулей? Кликните правой кнопкой мыши по названию устройства – Добавить – Канал. Добавляйте каналы в соответствии с возможностями подключаемых устройств.

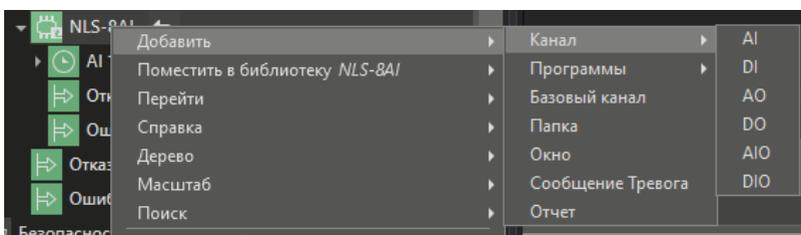


Рис. 2.16. Добавление канала

В каждом созданном канале есть три параметра:

- Value - читаемое или записываемое значение;

- SourceTime - метка времени, формируемая MasterSCADA 4D при получении и отправке данных;
- StatusCode - признак качества, начальное значение которого установлено в BadWaitingForInitialData для того, чтобы до получения значения по связи не выполнялась запись начального значения в архив, а также выдача начального значения на выходные каналы. Если в дальнейшем опрос произойдет успешно, то значение параметра поменяется на Good. Если при первом опросе возникнет ошибка, то параметр установится в BadNoCommunication. Если при опросах сначала будет установлено Good, а потом возникнет ошибка, то параметр примет значение BadOutOfService (при этом в Value останется последнее полученное значение).

Если нужно создать много каналов одного типа, можно продублировать уже созданный канал сколько необходимо раз. Перед дублированием можно настроить дублируемый канал, чтобы не повторять лишние действия на каждом из продублированных каналов.

Дважды кликните на устройство, которому принадлежит настраиваемый канал. Откроется новое окно с параметрами канала устройства. Дважды кликните по каналу, который нужно настроить и произведите настройку, после чего нажмите ОК > Применить.

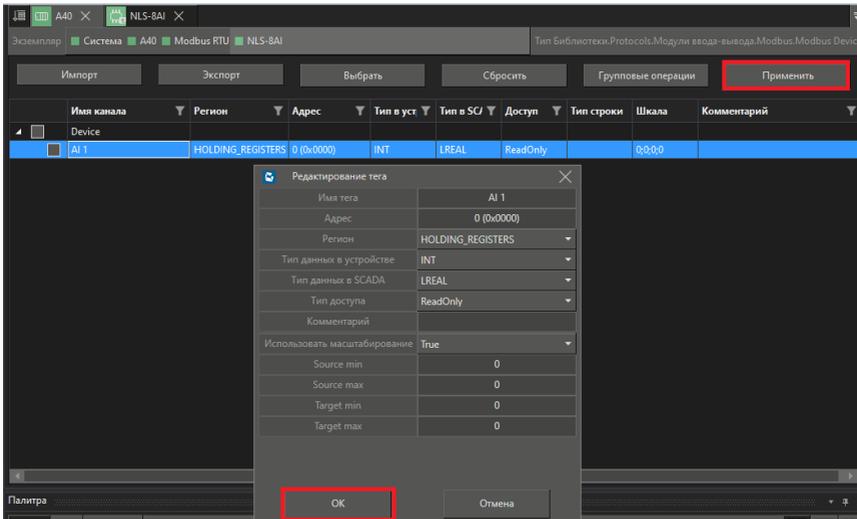


Рис. 2.17. Настройка канала

Для того, чтобы работать с каналом, который имеет **Выход** напрямую (не через объекты):

В дереве устройств откройте каналы настраиваемого устройства и выберите канал для настройки, раскройте его и выберите параметр **Выход**.

В панели свойств выберите – “Начальное значение” – “StatusCode” . Установите значение **Good**, иначе запись происходить не будет. В примере приведена настройка канала DO (дискретный выход). При записи в выход канала значений через объект, установите такие же настройки **StatusCode** для выхода тега, из которого значение передается в канал. В случае обмена через объект настраивать **StatusCode** в самом канале не нужно.

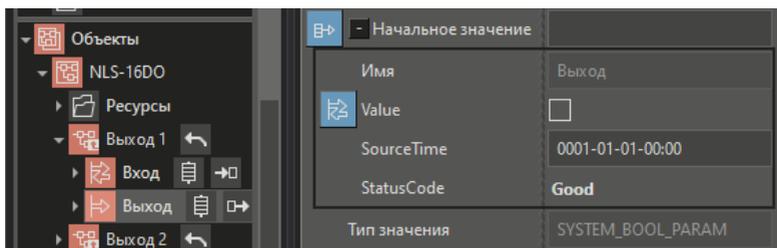


Рис. 2.18. Настройка Выхода тега

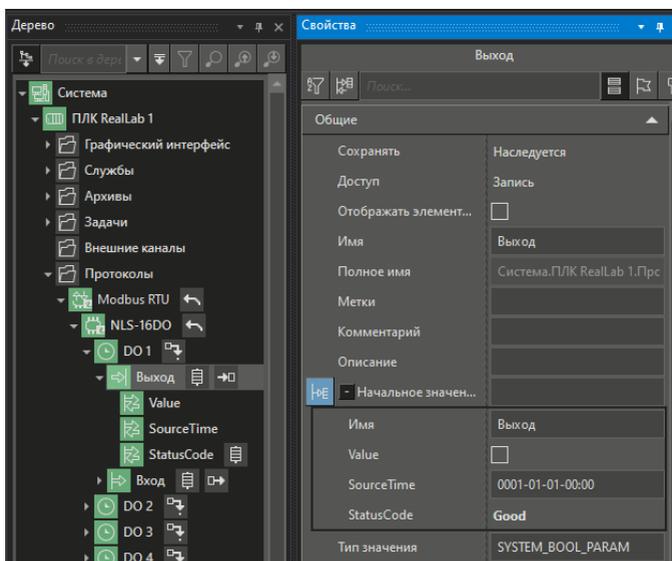


Рис. 2.19. Настройка Выхода канала

Нажмите правой кнопкой мыши на уже созданный канал и введите количество раз для дублирования канала, затем нажмите “Дублировать ...”.

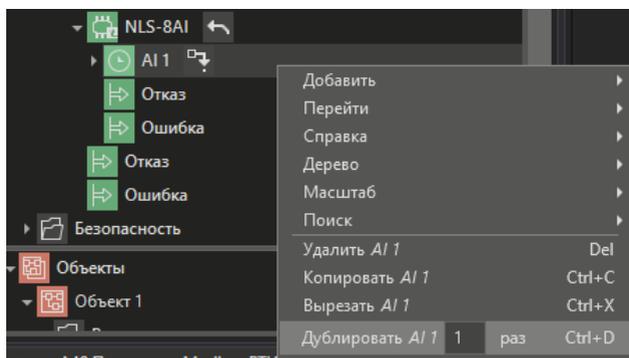


Рис. 2.20. Дублирование каналов

Измените настройки каждого из продублированных каналов, т.к. их адрес совпадает с адресом канала, на основе которого были продублированы остальные. Чтобы открыть таблицу каналов модуля, дважды кликните по его названию.

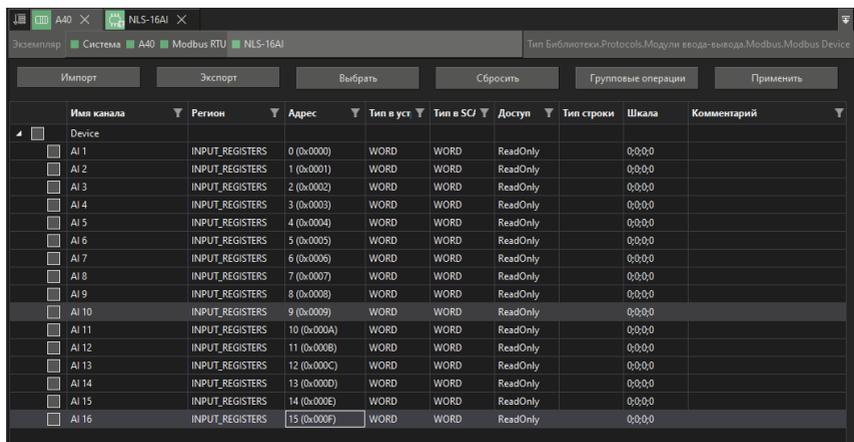


Рис. 2.21. Результат дублирования

ВАЖНО! Если Вам необходимо считывать по 2 WORD-регистра и объединять их в REAL (число с плавающей запятой), выполните следующие шаги:

- Должна быть отключена функция 0x6 Write Single Register в свойствах модуля;

- Во время добавления канала, который будет читать 2 регистра сразу, настройте его следующим образом: тип данных в устройстве: **REAL**, тип данных в SCADA: **REAL**. Значение “тип данных в устройстве” определяет тип считываемых данных. REAL = 4 байта, значит канал будет считывать 4 байта начиная с адреса 32 (см. рис. ниже). Т.к. регистр в Modbus = 2 байта, то при такой настройке канал сразу будет считывать 32 и 33 регистры, и выдавать значение в формате REAL.

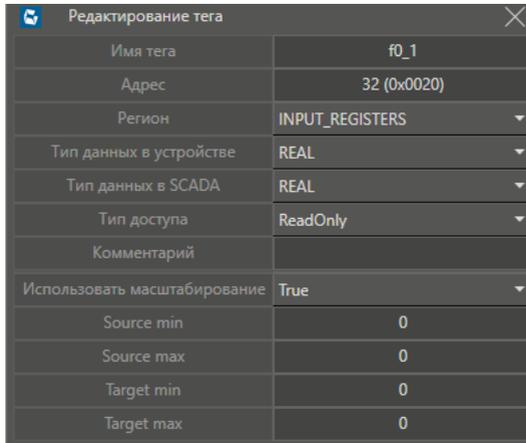


Рис. 2.22. Канал будет читать 2 регистра

- Чтобы число выводилось в правильном формате с плавающей запятой при работе с модулями RealLab кликните левой кнопкой мыши по названию модуля в дереве системы и в свойствах выбранного устройства выберите чередование байт, как на картинке ниже:

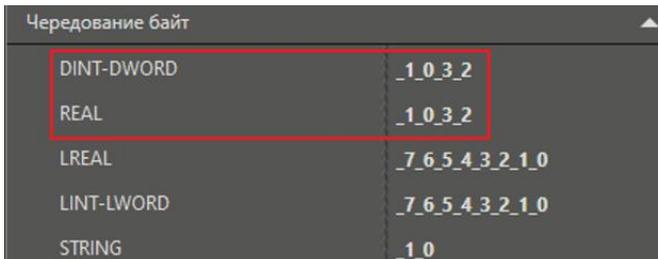


Рис. 2.23. Чередование байт

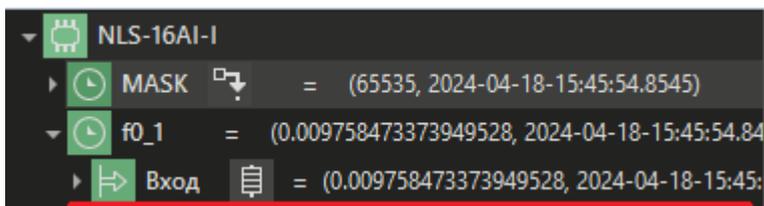


Рис. 2.24. Результат в формате числа с плавающей запятой

Чтобы ограничить кол-во отображаемых символов после запятой в числе с плавающей запятой откройте свойства канала или тега, в который передается значение с канала. Выберите в пункте “Шкала” параметр шкалы и в поле “Формат” впишите значение Fx, где x – кол-во знаков после запятой.

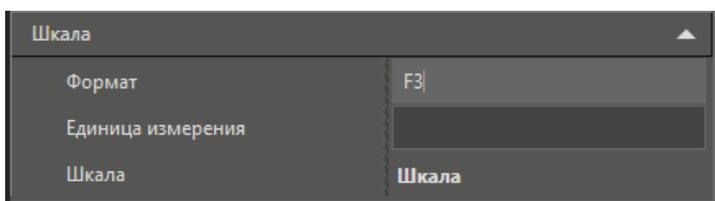


Рис. 2.25. Формат данных

Работа с физической частью проекта завершена.

2.3.4. Работа с объектами в проекте

Для начала необходимо вызвать контекстное меню дерева объектов, нажав правую кнопку мыши, и выполнить пункт Добавить - Объект.

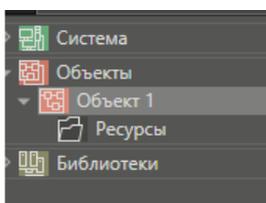


Рис. 2.26. Добавлен объект 1

Помимо параметров, в объект можно добавлять программы для обработки данных, графические окна, теги и другие элементы.

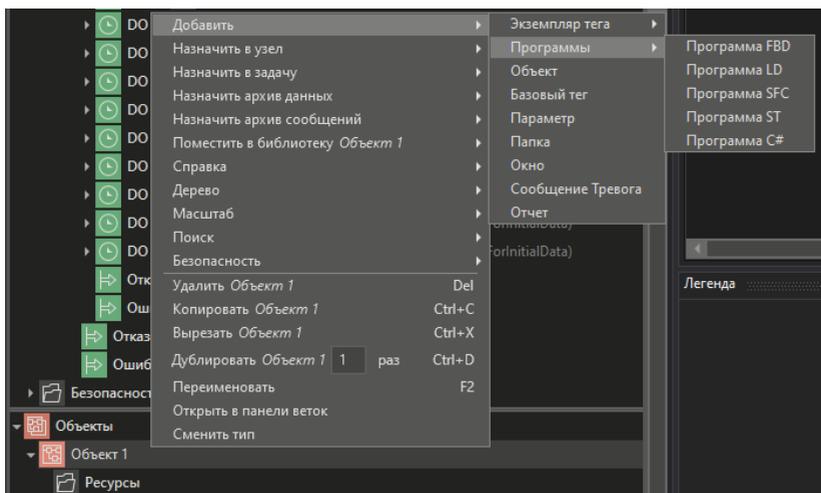


Рис. 2.27. Добавление программы в объект

Для проекта управления NLS-16DO по протоколу Modbus RTU были выполнены действия, описанные выше для NLS-8AI. Каналы NLS-16DO настроены как дискретные выходы. StatusCode у всех выходов тегов имеет начальное значение Good.

Имя канала	Регион	Адрес	Тип в уст	Тип в SCJ	Доступ
Device					
DO 1	COILS	0 (0x0000)	BOOL	BOOL	WriteOnly
DO 2	COILS	1 (0x0001)	BOOL	BOOL	WriteOnly
DO 3	COILS	2 (0x0002)	BOOL	BOOL	WriteOnly
DO 4	COILS	3 (0x0003)	BOOL	BOOL	WriteOnly
DO 5	COILS	4 (0x0004)	BOOL	BOOL	WriteOnly
DO 6	COILS	5 (0x0005)	BOOL	BOOL	WriteOnly
DO 7	COILS	6 (0x0006)	BOOL	BOOL	WriteOnly
DO 8	COILS	7 (0x0007)	BOOL	BOOL	WriteOnly
DO 9	COILS	8 (0x0008)	BOOL	BOOL	WriteOnly
DO 10	COILS	9 (0x0009)	BOOL	BOOL	WriteOnly
DO 11	COILS	10 (0x000A)	BOOL	BOOL	WriteOnly
DO 12	COILS	11 (0x000B)	BOOL	BOOL	WriteOnly
DO 13	COILS	12 (0x000C)	BOOL	BOOL	WriteOnly
DO 14	COILS	13 (0x000D)	BOOL	BOOL	WriteOnly
DO 15	COILS	14 (0x000E)	BOOL	BOOL	WriteOnly
DO 16	COILS	15 (0x000F)	BOOL	BOOL	WriteOnly

Рис. 2.28. Каналы NLS-16DO

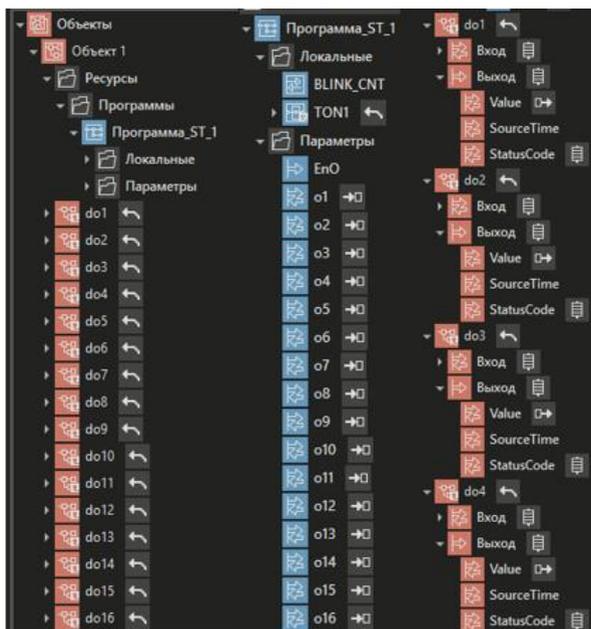


Рис. 2.29. Объекты проекта

Объект 1 содержит программу на языке ST и 16 логических тегов (BOOL) с названием do*, где * - номер выхода.

Программа_ST_1 содержит 17 параметров – 16 из них передают значения на логические теги, а теги передают значения на выходы.

Чтобы сформировать связь, достаточно перетащить переменную на значение Value входа/выхода тега. Чтобы передать значение тега в канал для записи на устройство, перетащите значение Value выхода тега на значение Value выхода канала устройства (также можно вместо этого перетащить **Выход** тега на **Выход** канала). Появится значок, подтверждающий связь между объектами.

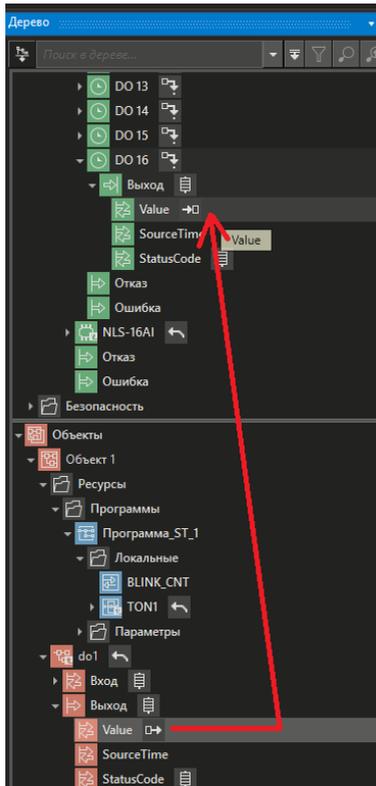


Рис. 2.30. Передача значение из тега на выход DO

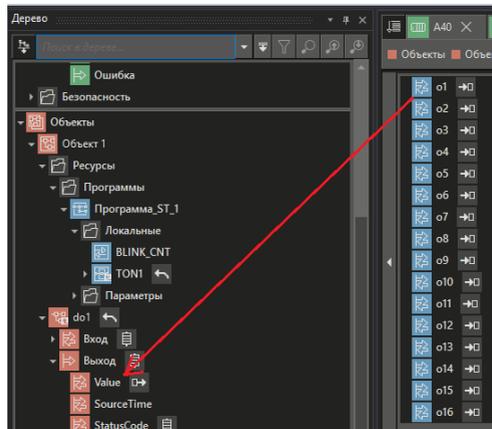


Рис. 2.31. Установление связи – передача значения o1 на Value

Локальные переменные объявляются в заголовке ST, который доступен из “Легенды”. Легенду можно открыть с помощью верхнего меню – “Вид” – “Легенда”.

Тип доступа параметров также устанавливается в Легенде.

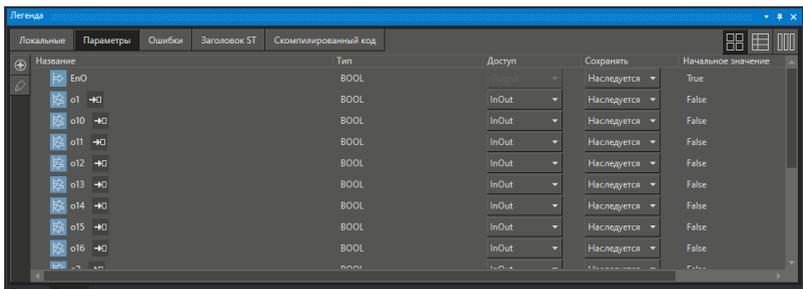


Рис. 2.32. Настройка параметров в легенде

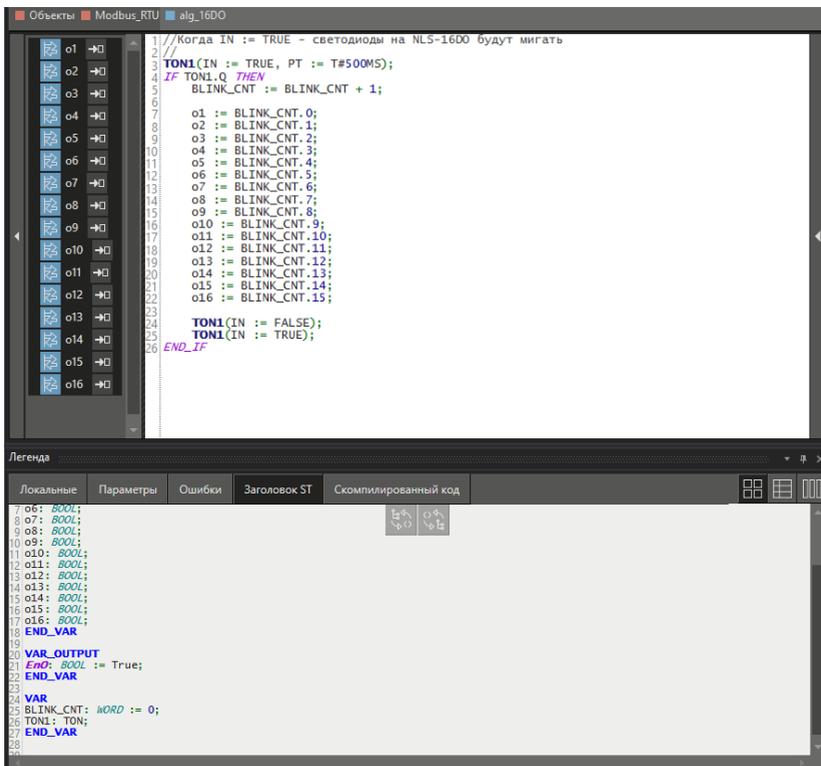


Рис. 2.33. Редактор ST-кода, заголовок ST в легенде

После того как все связи установлены, программа написана и проект готов к запуску, в верхней части программы кликните на “Исполнение” и нажмите на значок запуска проекта с загрузкой конфигурации устройства.

Если компиляция пройдет без ошибок, проект запустится.



Рис. 2.34. Запуск проекта

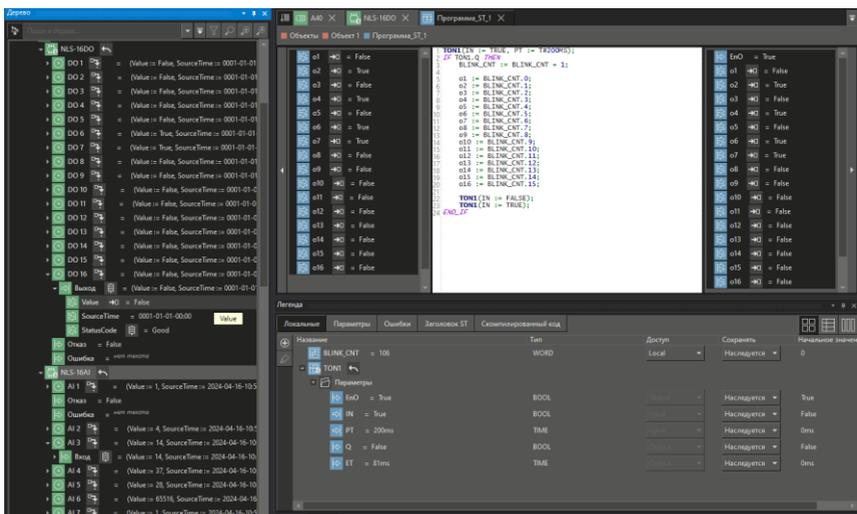


Рис. 2.35. Результат работы проекта

Сам проект опрашивает добавленные в него модули, позволяет записывать в них измененные параметры и меняет значение дискретных выходов NLS-16DO в соответствии с алгоритмом – двухбайтовое (WORD) число BLINK_CNT каждые 500 мс инкрементируется на единицу. Каждый бит BLINK_CNT записывается в соответствующий дискретный выход NLS-16DO.

Доработав проект и дополнив его еще несколькими модулями, получаем следующий результат:



Рис. 2.36. Выбор окна

The screenshot displays a complex configuration interface for a Modbus RTU device. It is organized into several panels:

- NLS-8R (Relay outputs):** A 7x2 grid of red circles representing relay outputs, numbered 0 to 14.
- NLS-16DI (Discrete inputs):** A 7x2 grid of red circles representing discrete inputs, numbered 0 to 15.
- NLS-16DO (Discrete outputs):** A 7x2 grid of green circles representing discrete outputs, numbered 0 to 15.
- NLS-4C (Control):** A table of control parameters including status signals, filter delays, and filter counts.
- NLS-8TI (Analog inputs):** A table listing 8 analog input channels with their addresses, scaling factors, and units.
- NLS-8AI (Analog inputs):** A table listing 8 analog input channels with their addresses, scaling factors, and units.
- NLS-4AO (Analog outputs):** A table listing 4 analog output channels with their addresses, scaling factors, and units.
- Right Panel (NLS-4RTD):** A table for RTD sensor configuration, including channel numbers, addresses, and scaling factors.

Рис. 2.37. Окно Modbus RTU

2.3.5. Формирование задач узлов

По умолчанию, все объекты исполняются в Основной задаче узла. Период обработки программы зависит от настроек задачи.

В случае если необходимо, чтобы часть объектов или программ исполнялись независимо от основной задачи, то создаются дополнительные задачи в узле в соответствующей группе. Затем тот или

иной элемент проекта при помощи контекстного меню назначается в ту или иную задачу.

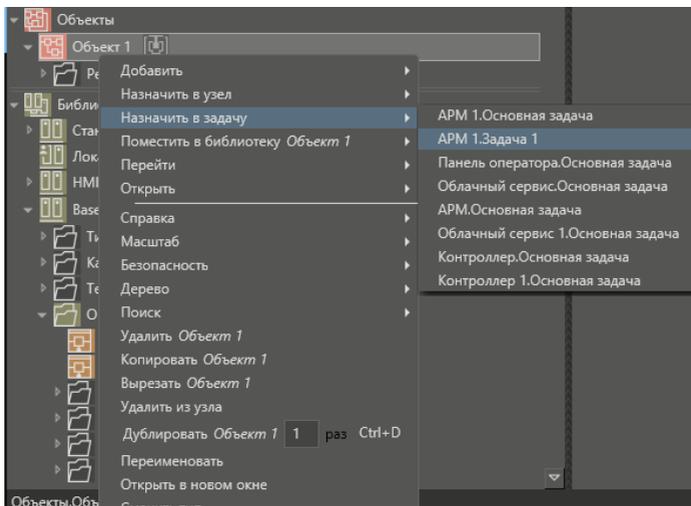


Рис. 2.38. Назначение объекта в задачу

2.3.6. Компиляция проекта и загрузка конфигураций в узлы

Разработанный проект должен быть скомпилирован. Для этого необходимо нажать кнопку Проверить панели инструментов вкладки Исполнение.

Для проверки конфигурации отдельного узла следует выполнить соответствующую команду контекстного меню в дереве системы узла. При компиляции проекта производится также проверка и компиляция программ.

В случае успешной компиляции проекта кнопка останется зеленой, и на экране отобразится соответствующее сообщение:

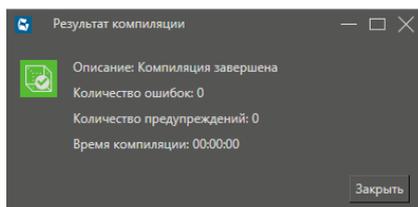


Рис. 2.39. Компиляция без ошибок и предупреждений

Если в результате компиляции не будет ошибок, но будут предупреждения, то кнопка изменит свой цвет на желтый, а на экране отобразится соответствующее предупреждение. Информация об ошибках выведется в окно панели Компиляция вкладки Исполнение:

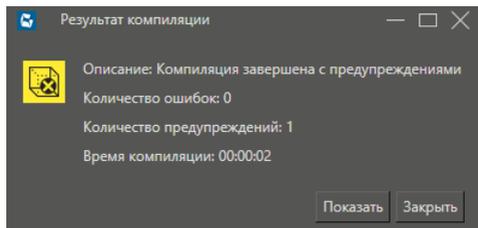


Рис. 2.40. Компиляция без ошибок, но с предупреждениями

При наличии в проекте ошибок кнопка изменит свой цвет на красный, а на экране отобразится соответствующее предупреждение. Информация об ошибках выведется в окно панели Компиляция вкладки Исполнение.

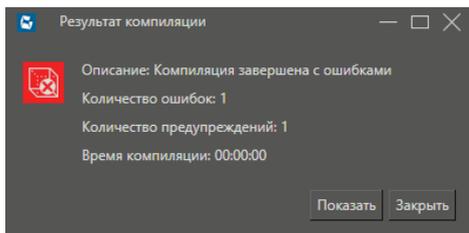


Рис. 2.41. Компиляция с ошибками

Действия, которые выполняют кнопки на скриншоте ниже:

1. Подключение к выбранному узлу и загрузка текущей конфигурации.
2. Подключение к выбранному узлу без загрузки текущей конфигурации.
3. Запуск выбранного проекта в режиме эмуляции (автономная отладка).
4. Загрузка конфигурации проекта в выбранный узел.
5. Компиляция проекта и проверка на наличие ошибок.
6. Отображение системного окна сообщений проекта.



Рис. 2.42. Кнопки во вкладке “Исполнение”

2.3.7. Приоритет выполнения задачи

В свойствах любой задачи, службы, протокола присутствует параметр Приоритет. Он задает приоритет протокола. Если работа по протоколу с повышенным приоритетом не укладывается в свой период опроса, то в случае одноядерного процессора в устройстве другие задачи исполняться не будут, а в случае наличия многоядерного процессора они будут исполняться в реже. В большинстве проектов эту настройку можно оставить без изменения. Использование ее оправдано только когда необходимо гарантировано обеспечить цикл вычисления наиболее важных частей проекта. Приоритет более 100 считается повышенным.

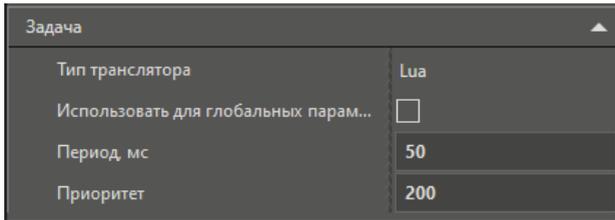


Рис. 2.43. Установка приоритета основной задачи

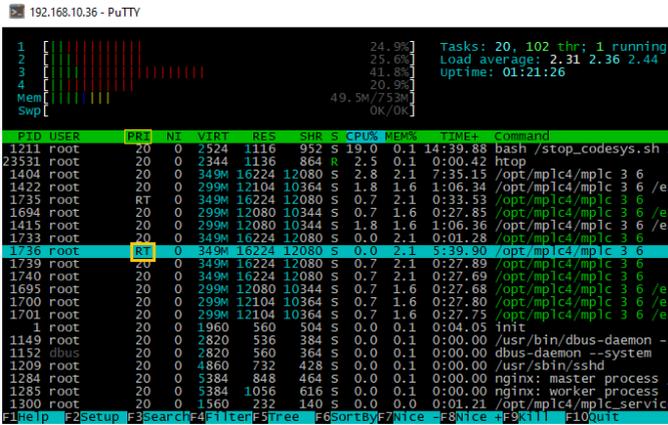


Рис. 2.44. Результат изменения приоритета после загрузки проекта в память ПЛИК NLSccon-A40

2.4. Подключение к ПЛК

После того, как Вы создали проект, необходимо подключиться к ПЛК для его загрузки. Чтобы это сделать, необходимо знать IP-адрес ПЛК. Если IP-адрес ПЛК неизвестен, воспользуйтесь сканером IP-адресов устройств, на которых установлена среда исполнения MasterPLC. Сканер можно скачать с [нашего сайта](#). Настройка статического IP-адреса ПЛК описана в п. 2.5.

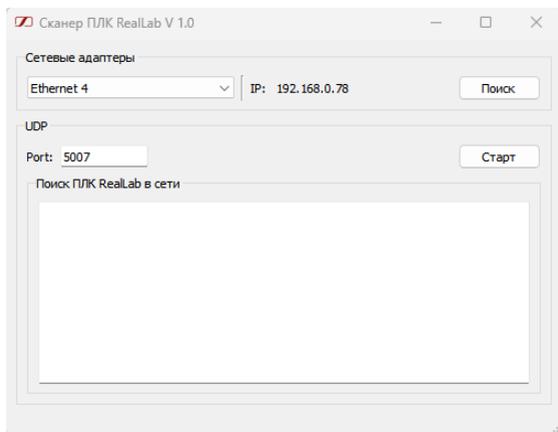


Рис. 2.45. Интерфейс сканера

После добавления ПЛК Вам необходимо указать его IP-адрес. Это можно сделать в свойствах ПЛК – “Связь”:

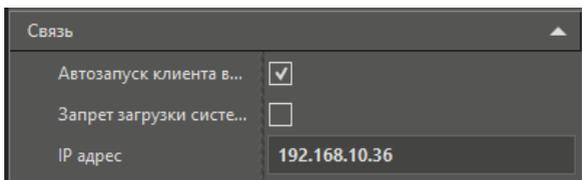


Рис. 2.46. Указание IP-адреса ПЛК

2.5. Настройка статического IP-адреса ПЛК

По умолчанию ПЛК настроены на получение динамического IP-адреса по DHCP. Для корректной работы MasterSCADA 4D рекомендуется настроить на ПЛК статический IP-адрес.

2.5.1. Настройка статического IP-адреса ПЛК серии NLScon-A40

Подключитесь к ПЛК через debug-порт или SSH.

Для настройки статического IP адреса интерфейсов eth0, eth1. необходимо отредактировать файл «interfaces» в директории “/etc/network/”. Откройте файл с помощью терминального текстового редактора командой:

```
sudo nano /etc/network/interfaces
```

Приведите файл к виду, как на скриншоте ниже:

```
GNU nano 2.7.1 File: /etc/network/interfaces

auto lo
iface lo inet loopback

#[eth0]-----

#allow-hotplug eth0
auto eth0

#iface eth0 inet dhcp

iface eth0 inet static
    address 192.168.0.104
    netmask 255.255.255.0
    gateway 192.168.0.254
    dns-nameservers 192.168.0.7 8.8.8.8

#[eth1]-----
allow-hotplug eth1
auto eth1
#iface eth1 inet dhcp

iface eth1 inet static
    address 192.168.10.36
    netmask 255.255.255.0
    gateway 192.168.10.254
    dns-nameservers 192.168.0.7 8.8.8.8
```

Рис. 2.47. Настройка IP-адреса ПЛК серии NLScon-A40

Если строка “iface eth0 inet dhcp” раскомментирована, это означает, что ПЛК получит IP-адрес от хоста по протоколу DHCP после подключения кабеля Ethernet к порту eth0 ПЛК.

Если строка “iface eth0 inet dhcp” закомментирована и раскомментирована строка “iface eth0 inet static”, а также указаны следующие параметры, то будет установлен статический IP-адрес для интерфейса eth0:

```
address 192.168.0.104 (IP-адрес)
netmask 255.255.255.0 (Маска)
```

gateway 192.168.0.254 (Шлюз)
dns-nameservers 192.168.0.7 8.8.8.8 (DNS-серверы)

Сохраните изменения сочетанием клавиш CTRL + O, ENTER, CTRL + X. И перезагрузите ПЛК, командой `sudo reboot`.

Проверить результат можно через команду "`ifconfig`". Интерфейс `eth1` настраивается аналогично интерфейсу `eth0`.

Более подробно настройка статического IP-адреса описана в руководстве эксплуатации настраиваемого ПЛК.

2.5.2. Настройка статического IP-адреса ПЛК серии NLScon-RSB

Подключитесь к ПЛК через VNC или SSH. Для настройки статического IP-адреса интерфейсов `eth0` и `eth1` будет использоваться встроенный в ПЛК консольный конфигуратор – `rl-config`.

Вызовите конфигуратор командой “`sudo rl-config`” из терминала ПЛК.

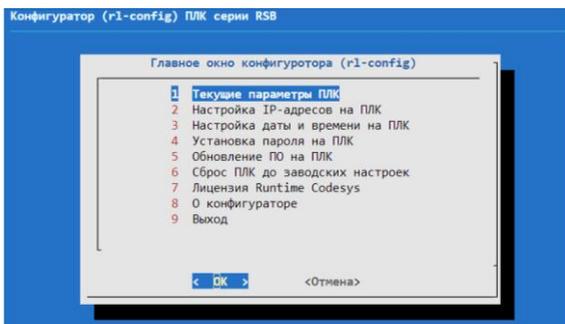


Рис. 2.48. Конфигуратор `rl-config`

Выберите пункт 2 - “Настройка IP-адресов на ПЛК”. Из списка выберите интерфейс для настройки (`eth0` или `eth1`). Введите значения для IP-адреса, маски, DNS-серверов.

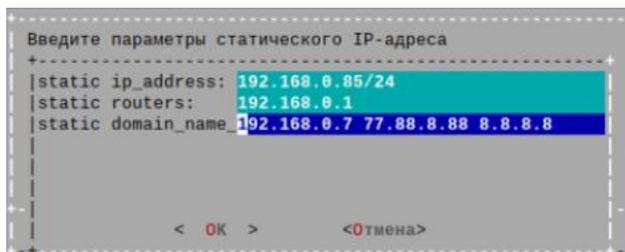


Рис. 2.49. Настройка статического IP-адреса

Сохраните результат изменения, нажав на «Ок» и подтвердив данные изменения. Проверить результат изменения IP адреса можно в конфигураторе в разделе “Текущие параметры ПЛК”.

Более подробно настройка статического IP-адреса описана в руководстве эксплуатации настраиваемого ПЛК.

2.6. Работа с Retain-переменными

Чтобы работать с Retain-переменными должна быть активна и настроена служба “Сохранение состояния”. В MasterSCADA 4D, в дереве системы откройте папку “Службы” и нажмите на “Сохранение состояния”. В свойствах данной службы (в самом верху) **не должна** стоять галочка “Отключить исполнение”.

Сохраняются те значения параметров, в настройках которых установлено свойство “Сохранять” = Да. Данные будут сохраняться в файле session.bin. Расположение файла зависит от настроек службы – его можно указать в “Свойствах” – “Протокол” – “место хранения данных”. Если поле пустое, то файл сохранится в следующую директорию исполнительной системы: /opt/mplc4/

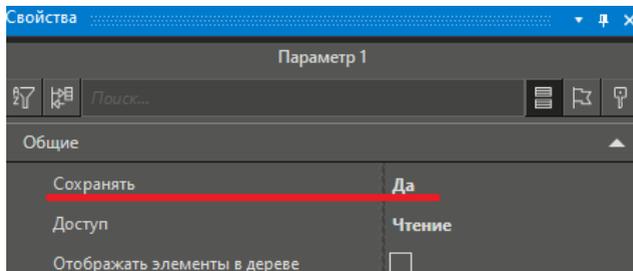


Рис. 2.50. Сохранение Retain-переменной

Поддерживается сохранение значений параметров различных типов данных, в том числе динамических массивов (массивов переменной длины), также длинных строк (более 256 символов), при условии, что параметр имеет тип STRING.

ВАЖНО! *Свойство Сохранять=Да нельзя устанавливать для параметров программ, у которых свойство Доступ=InOut.*

Если при перезагрузке ПЛК происходит обнуление, то нужно проверить файл горячего рестарта. Как правило, данные для горячего рестарта хранятся в файлах session.bin, session2.bin. Запись в них идет поочередно.

При старте сравниваются времена обновления файлов, вначале происходит попытка загрузки из более нового. Если контрольная сумма не совпадает, идет попытка загрузить из другого.

Основное место расположения файлов зависит от настроек проекта, а также от типа контроллера, на котором работает исполнительная система. Исключения составляют контроллеры с энергонезависимой памятью, у которых данные хранятся непосредственно в этой памяти. Часто возникает задача произвести импорт/экспорт данных для горячего рестарта.

Для импорта файла данных для горячего рестарта необходимо скопировать файл `session.bin` в рабочую папку исполнительной системы `/opt/mplc4/import`. При очередной перезагрузке необходимые данные считываются из этого файла, после чего он будет удален, а сами данные сохранятся в основном месте хранения.

Для резервирования (backup'a) файла данных для горячего рестарта можно скопировать файл `session.bin` в папку `/opt/mplc4/backup`. При перезагрузке данные для горячего рестарта, как обычно, будут считываться из основного места хранения, но, в случае возникновения ошибки чтения, смогут быть получены из этого файла

Для удаления файла данных для горячего рестарта необходимо запустить `mplc` с опцией `/delhr`.

ВАЖНО!

- 1. Если на момент запуска проекта, `Retain` параметр отсутствует в файле `session.bin` (новый созданный параметр), то атрибут «Начальное значение» применяется, в противном случае значение загружается с файла `session.bin`.*
- 2. Если в проекте есть параметры, которые изменяются один раз при создании проекта, то настраивайте их как константы «Сохранять-Константа».*
- 3. Если у параметра `Retain` есть входящая связь с параметром, не сохраняющим своё значение, то при перезагрузке ПЛК в параметр `Retain` будет записано начальное значение не сохраняемого параметра.*
- 4. При отключении питания ПЛК отключается с небольшой задержкой, чтобы `retain`-переменные успели сохраниться в энергонезависимую память.*

2.7. Работа светодиодов и переключателя Run-Stop на ПЛК

- Переключатель Run-Stop, останавливает/запускает МЭК программу MasterSCADA 4D;
- При запуске МЭК программы светодиод горит зеленым цветом, а при остановке – красным.
- При запуске ПЛК МЭК-программа проверяет состояние переключателя. Если переключатель в положении Стоп, то МЭК-программа останавливается, а если в переключатель в положении Старт, то запускается.
- При отключении основного питания ПЛК, ПЛК продолжает работать с аккумулятором X секунд (по умолчанию 5 секунд) и отправляет команду для безопасного завершения работы Runtime MasterPLC и сохранения retain-переменных и затем отправляет команду выключения ПЛК.

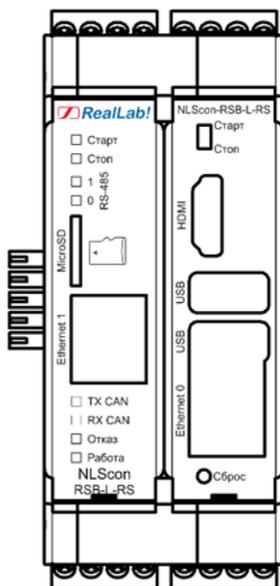


Рис. 2.51. Вид на лицевую панель ПЛК NLScon-RSB-L-RS

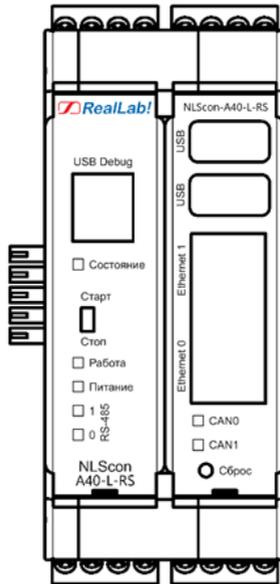


Рис. 2.52. Вид на лицевую панель ПЛК NLScon-A40-L-RS

2.8. Работа с визуализацией

2.8.1. Добавление окна

Для создания визуализации в дереве объектов кликните по объекту и добавьте окно. У объекта сразу появится окно, которое откроется после создания визуализации.

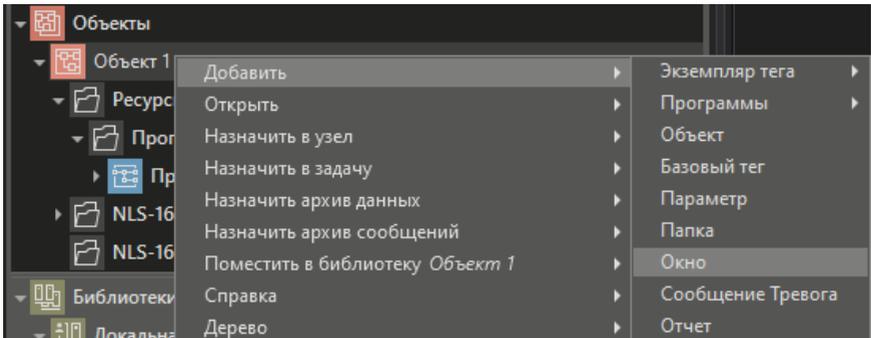


Рис. 2.53. Создано окно визуализации

Чтобы сделать созданное окно стартовым, кликните по названию окна – назначить стартовым окном – шаблон экрана.



Рис. 2.54. Назначение стартового окна

2.8.2. Добавление объектов в окно визуализации

Добавить объекты (контролы, примитивы и т.д.) в окно визуализации можно с помощью палитры.

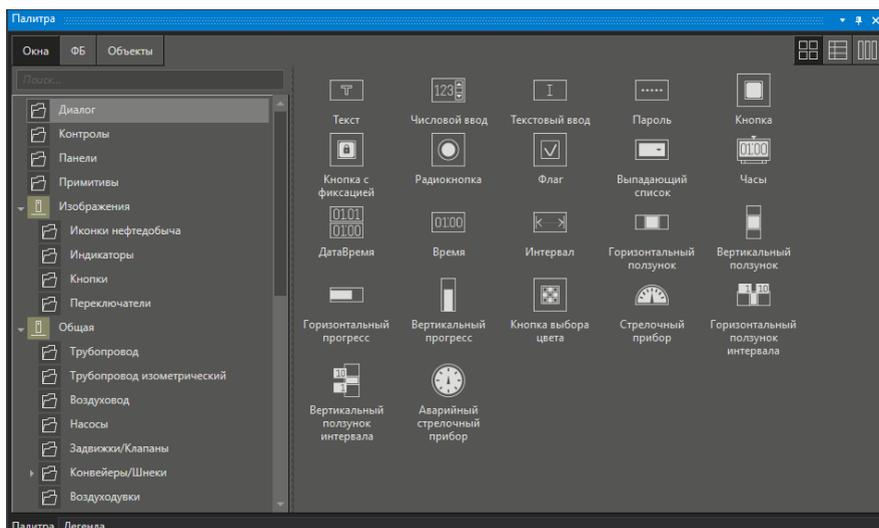


Рис. 2.55. Палитра

Если Палитра не отображается – нажмите в верхнем меню “Вид” – “Палитра”. Палитра включает в себя множество элементов, которые можно использовать для визуализации. Сначала добавьте в окно визуализации элемент “Кнопка с фиксацией” и примитив “Овал”.

2.8.3. Организация обмена данными между параметрами и визуализацией

Далее создайте параметры в дереве объектов. В эти параметры будут передаваться значения с каналов устройств, для передачи их на визуализацию, и в обратном порядке – с визуализации на устройство.

При создании параметра можно выбрать тип данных, начальное значение и права обращения к параметру (чтение/запись).

Далее перетащите значение (Value) канала на название параметра, с которым хотите установить связь.

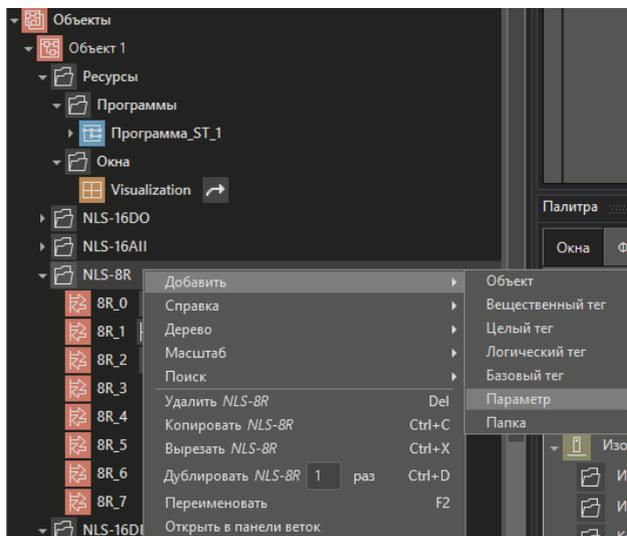


Рис. 2.56. Добавление параметра

Для того, чтобы связать управление параметром с кнопкой фиксации, перетащите параметр на ту кнопку, с помощью которой хотите управлять состоянием параметра (в случае с логическим типом параметра). Теперь при нажатии на кнопку, состояние связанного с ней параметра будет меняться на противоположное. Повторное нажатие на кнопку возвращает параметр в исходное состояние.

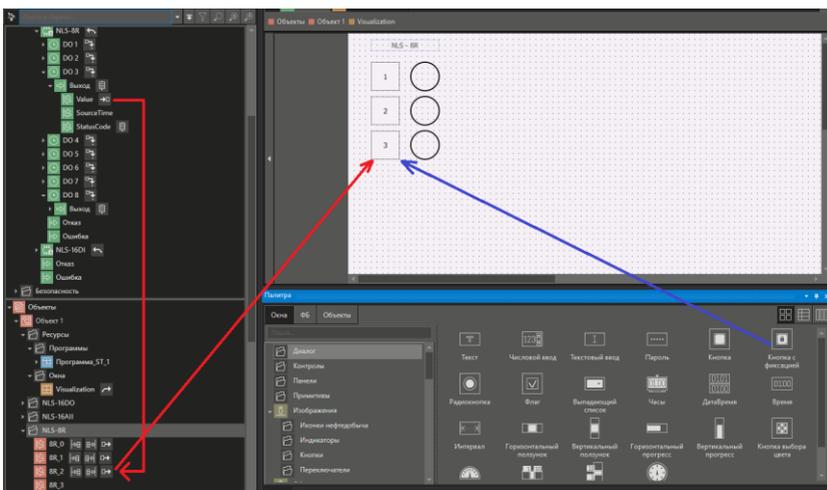


Рис. 2.57. Установление связей между каналом, параметром, визуализацией

Чтобы овалы, которые добавлены в визуализацию, меняли цвет в зависимости от состояния параметра, необходимо перетащить параметр объекта на свойство овала “Заливка фона”.

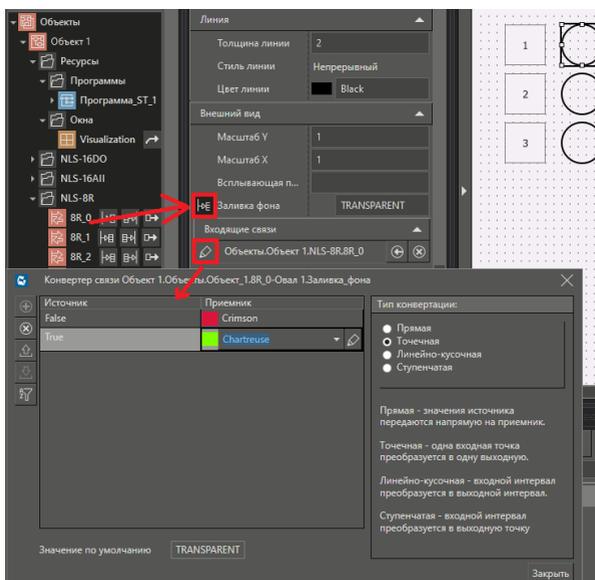


Рис. 2.58. Назначение цветов овала на состояния параметра 8R_0.

Далее нажмите на кнопку редактирования входящей связи – откроется “Конвертер связи”. Выберите тип конвертации “точечная” и добавьте два значения – True и False. Значению True присвойте зеленый цвет, а значению False – красный. Закройте окно конвертера связи.



Рис. 2.59. Результат работы с цветными овалами (индикаторами)

2.8.4. Формат отображаемых данных

Чтобы добавить отображение значений аналоговых входов в текстовом формате (число с плавающей запятой) перетащите “Вход” канала на параметр объекта, в который будет передаваться значение входа. Этот же параметр перетащите на визуализацию, он появится в виде элемента с текстом. Свойства текста и формат отображения можно задать в свойствах текста. Формат значений для чисел с плавающей запятой – Fx, где x – количество знаков после запятой.

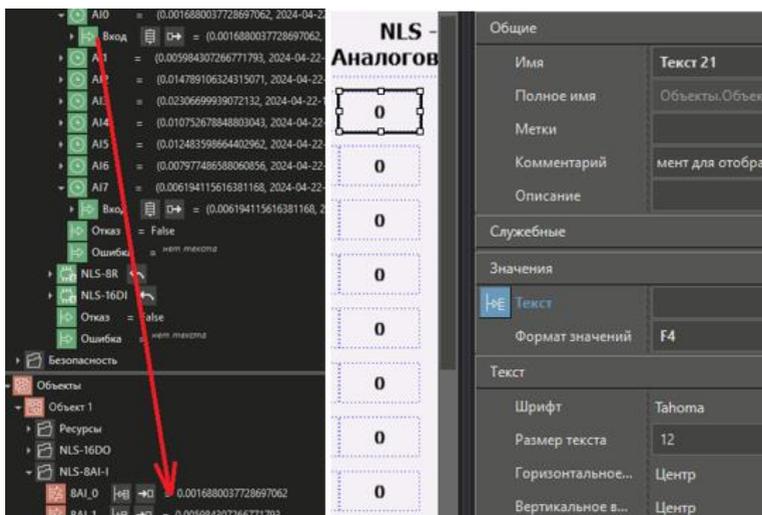


Рис. 2.60. Свойства текста

2.8.5. Вызов программ по кнопке из окна визуализации

Есть возможность сделать вызов программ по кнопке в визуализации.

Для этого просто перетащите из дерева объектов программу, которую нужно вызвать, в окно визуализации. Автоматически появится кнопка вызова программы с ее именем. Чтобы программа не исполнялась постоянно, а вызывалась только по кнопке, в свойствах этой программы установите параметр “Способ исполнения” – “По вызову”.

Также в свойствах элементов визуализации можно настроить размер, название, цвет, свойства текста, связь с переменными (например, на кнопку можно назначить логическую переменную, и переменная будет менять состояние в зависимости от нажатия на кнопку). Присутствует возможность настройки действий по наведению мыши, по клику мыши и т.д.

2.8.6. Добавление картинки в визуализацию

В редакторе визуализации есть возможность добавить любую картинку. Для этого в **Палитре** воспользуйтесь поисковой строкой или выберите папку “Примитивы” и перетащите в визуализацию примитив “Картинка”.

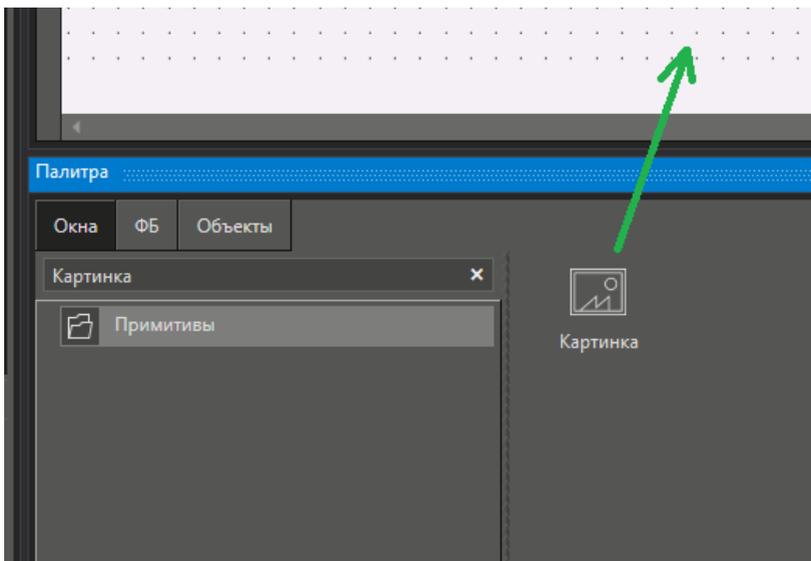


Рис. 2.61. Добавление картинки

В свойствах Картинки, в пункте **“Изображение”** нажмите на «...».

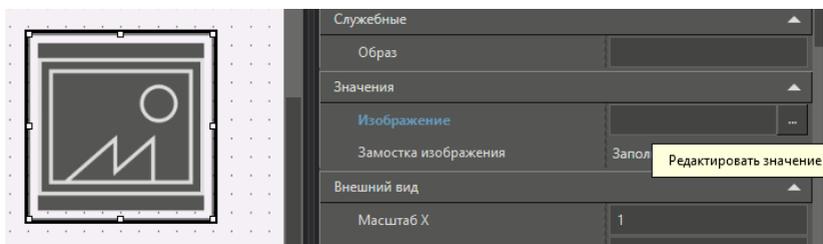


Рис. 2.62. Выбор изображения

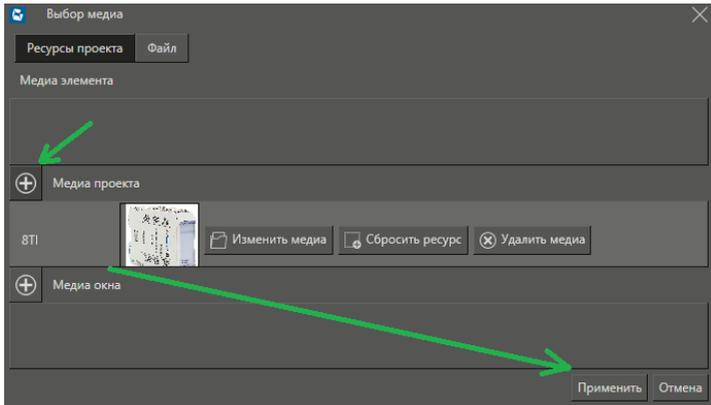


Рис. 2.63. Продолжение выбора изображения

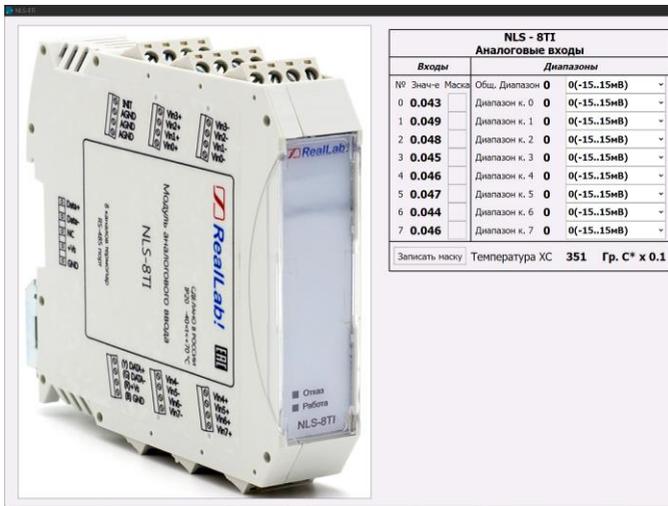


Рис. 2.64. Результат

2.8.7. Добавление экранной клавиатуры в визуализацию

Экранную клавиатуру удобно использовать, если визуализация отображается на устройстве с сенсорным экраном, что позволяет вводить данные в окно визуализации без подключения физической клавиатуры.

В шаблон ПЛК RealLab, который содержится в библиотеке RealLab, уже встроена экранная клавиатура. Работает она следующим образом:

После того, как Вы создадите окно визуализации и назначите его стартовым окном, добавьте элемент “текстовый ввод” из Палитры в окно визуализации.

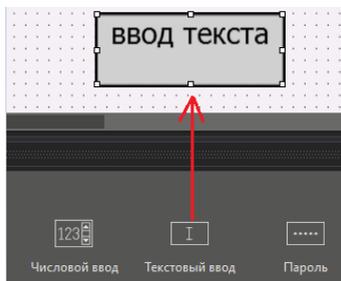


Рис. 2.65. Добавление текстового ввода из палитры

Подключитесь к ПЛК с загрузкой конфигурации. Откройте клиент визуализации. При клике курсором на поле ввода данных, справа от него появляется значок, при нажатии на который по центру экрана появляется экранная клавиатура.

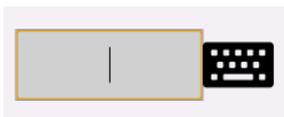


Рис. 2.66. Значок экранной клавиатуры



Рис. 2.67. Экранная клавиатура

Для закрытия клавиатуры без применения значения нажмите клавишу “Esc”. Для применения значения нажмите клавишу “< Enter”. Клавиша переключения раскладки находится в нижнем левом углу клавиатуры и содержит сокращенное название выбранного языка.

Если возникла ошибка и вместо визуализации отображается надпись “Загрузка...” долгое время выполните следующие действия:

- Найдите на накопителе вашего ПК или скачайте снова библиотеку RealLab для MasterSCADA 4D. Распакуйте ее в любую папку. В папке с распакованной библиотекой найдите папку “Res”. В ней хранятся файлы, отвечающие за работу клавиатуры.

The screenshot shows a Windows File Explorer window with the address bar set to 'e:\Pereverzev\IO\MASTER_SCADA_FILES\Libraries\RealLab*.*'. The main pane displays a table of files and folders:

Name	Ext	Size	Date
[..]		<DIR>	24.07.2024 09:21
[Res]		<DIR>	24.07.2024 09:21
info	xml	623	24.07.2024 09:21
RealLab	fdb	73 932 800	24.07.2024 09:21

Рис. 2. 68. Папка Res

- Скопируйте или переместите папку “Res” по пути C:\Users\Public\MasterSCADA4D1.3\Libraries\RealLab\
- Снова загрузите проект с виртуальной клавиатурой на ПЛК и проверьте ее работоспособность.

The screenshot shows a Windows File Explorer window with the address bar set to 'c:\Users\Public\Documents\MasterSCADA4D1.3.Beta\Libraries\RealLab*.*'. The main pane displays a table of files and folders:

Name	Ext	Size	Date
[..]		<DIR>	24.07.2024 11:16
[Queries]		<DIR>	08.07.2024 13:27
[Res]		<DIR>	24.07.2024 11:16
info	xml	623	08.07.2024 13:56
RealLab	fdb	73 932 800	08.07.2024 13:57

Рис. 2. 69. Пример того, где должна находиться папка Res с ресурсами для работы экранной клавиатуры

2.9. OPC UA Server

При работе в роли сервера MasterSCADA 4D поддерживает режимы чтения и подписки. Поддерживаются простые типы данных, а также возможна работа с одномерными массивами, состоящими как из элементов простых типов данных, так и из предопределенных типов вида SYSTEM_*_PARAM. Также поддерживаются одномерные массивы структур.

Для того, чтобы MasterSCADA 4D выступала в роли OPC UA Server’a, создайте новый проект или откройте существующий. Добавьте в проект ПЛК при его отсутствии, откройте **свойства** ПЛК, вкладку “Связь”. Установите галочку на параметр “Доступ по OPC UA” и укажите **порт для**

OPC UA. Загрузите конфигурацию в память ПЛК. По окончании загрузки конфигурации узла среда подключается к исполнительной системе.

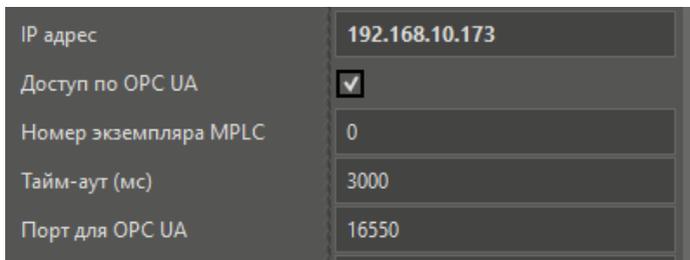


Рис. 2.70. Свойства ПЛК, Связь

Также у каждого из параметров, каналов, объектов, устройств, протоколов в **Свойствах** есть пункт **“Разрешения”**. Там можно настроить отдельно для каждого параметра доступ по OPC UA.

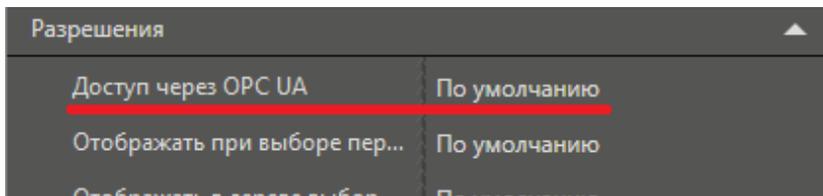


Рис. 2.71. Свойства канала, Разрешения

Отключите среду разработки от исполнительной среды (нажмите кнопку **“Стоп”** во вкладке **“Исполнение”**).

Запустите клиент OPC UA (в примере используется UAExpert) и нажмите значок **“+”** в верхнем меню для добавления сервера.

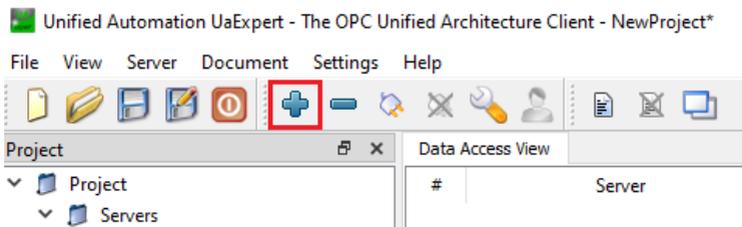


Рис. 2.72. Добавление сервера в UAExpert

На вкладке **“Advanced”** открывшегося диалога настройте подключение к исполнительной системе MasterSCADA 4D как к серверу OPC UA.

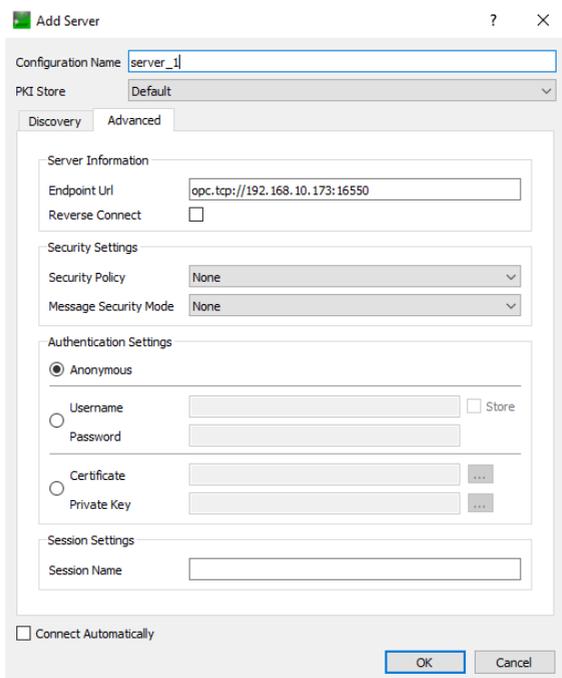


Рис. 2.73. Параметры добавляемого сервера

Далее в дереве проекта нажмите ПКМ на название добавленного сервера, выберите **“Connect”**.

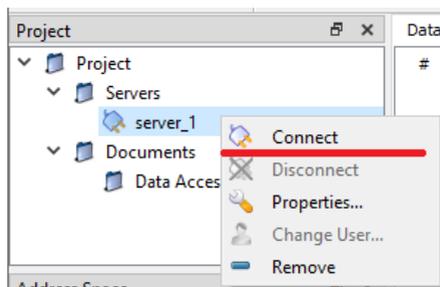


Рис. 2.74. Подключение к серверу OPC UA

Появится ошибка **“URL not found”**. Нажмите **ОК**. Затем откроется диалог с сообщением о том, что сертификат OPC UA сервера не принадлежит к числу доверенных. Нажмите **“Trust Server Certificate”** в нижней части окна и установите галочку на **“Accept the server certificate temporarily for this session”**. Далее нажмите **“Continue”**.

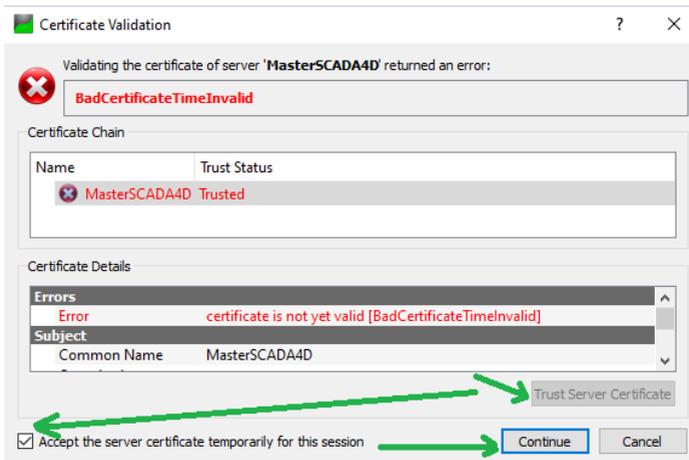


Рис. 2.75. – Подтверждение подключения

Элементы проекта будут отображаться в дереве клиента в зависимости от настройки элементов Доступ через OPC UA. Способ отображения структур в OPC UA клиенте определяется свойством узла **“Отображение структур”** в OPC UA сервере.

При работе по протоколу OPC UA Slave создается, если его не было ранее, сертификат в рабочей папке исполнительной системы CertificateStore.

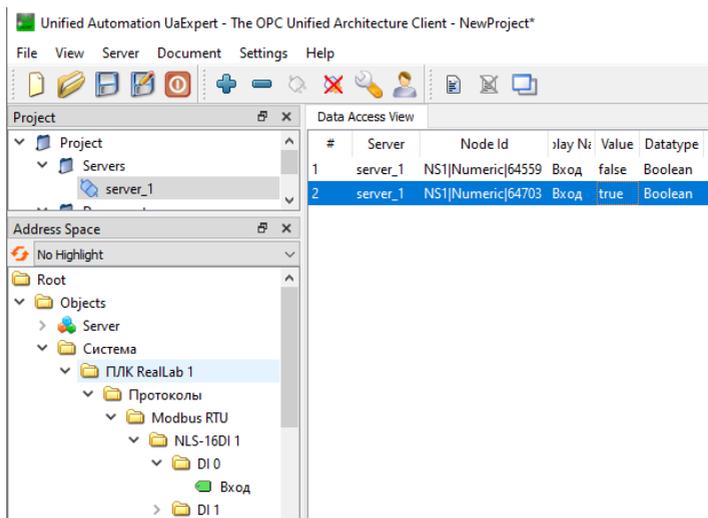


Рис. 2.76. Чтение параметров через OPC UA Client (UaExpert)

Порядок передачи данных при подписке

В клиенте, получающим данные от исполнительной системы MasterSCADA 4D, можно установить следующие настройки согласно стандарту OPC UA:

Publishing Interval - период, с которым сервер отправляет клиенту пакеты с данными.

Queue Size - настройка, задаваемая у тега, указывает максимальное количество значений по этому тегу, передаваемых в одной подписке.

SamplingInterval - это частота, с которой вы хотите, чтобы сервер выполнял выборку/опрос/мониторинг элемента. Элементы подписки могут поступать из разных источников, и каждый из них может отбираться по своему усмотрению.

Если установлено значение -1 или 0, то клиенту отправляется каждое изменение параметра (не более QueueSize в каждом пакете). Если больше 0, то клиенту отправляются значения, если его метка отличается от предыдущего, отправленного больше на **SamplingInterval**.

Например, в клиенте установлено: PublishingInterval = 1сек., QueueSize = 10, SamplingInterval = -1, а значение меняется раз в 100мс, то сервер должен отправлять клиенту по 10 значений, но не чаще 1 раз в секунду.

Поддержка параметра у тега подписки MonitoringMode=Sampling. Установка данного параметра означает, что элемент только отбирается, а изменения значений помещаются в очередь, которая есть у каждого отслеживаемого элемента, но об изменениях фактически не сообщается клиенту.

2.10. Диагностика контроллера

Существует три метода получения диагностической информации работы среды исполнения:

- Стандартная диагностика
- Диагностика по TCP/IP
- Специальные ФБ

2.10.1. Стандартная диагностика

Для получения диагностической информации о состоянии контроллера можно использовать “Панель Узла”. Панель можно открыть, нажав дважды

левой кнопкой мыши на узел в дереве системы, либо выбрав в контекстном меню узла пункт “Открыть узел”.

После подключения среды разработки к работающей среде исполнения во вкладке “Управление Панели Узла” появится возможность получать информацию о состоянии контроллера и о загруженном в него программном обеспечении. Во вкладке “Статистика Панели узла” будут отображаться статистики всех задач исполнительных системы в виде таблицы.

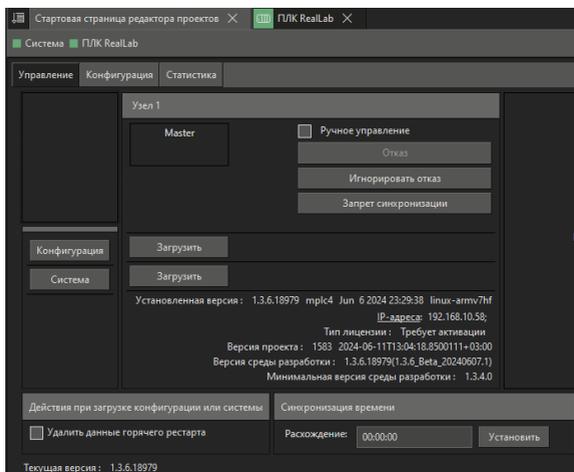


Рис. 2.77. Информация о контроллере

Задача	Период задачи (мс)	Факт. период зап. задачи (мс)			Факт. время исп. задачи (мс)			
		Сред.	Мин.	Макс.	Послед. цикл	Сред.	Мин.	Макс.
Modbus RTU	100	138.85	99	3104	96	133.71	94	3099
Основная задача	100	100.16	100	101	3	2.66	1	9

Рис. 2.78. Вкладка со статистикой

2.10.2. Диагностика по TCP/IP

Работающая среда исполнения может выдавать диагностическую информацию по TCP/IP через порт 31550. Для чтения этой информации можно воспользоваться любым программным эмулятором терминала, например NuregTerminal, PuTTY и др.

Рассмотрим запись лога с отладочной информацией при помощи программы PuTTY. После запуска программы откроется ее основная вкладка. Установите следующие параметры: (порт 31550, тип соединения: Raw):

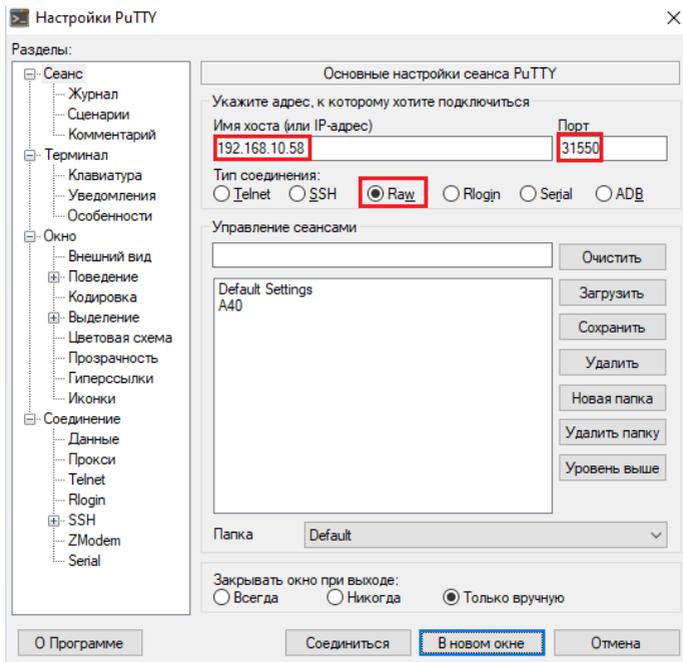


Рис. 2.79. Установка параметров подключения

Далее перейдите на вкладку “Журнал”. Установите путь сохранения лога и журналирование сессии = “Весь вывод”.

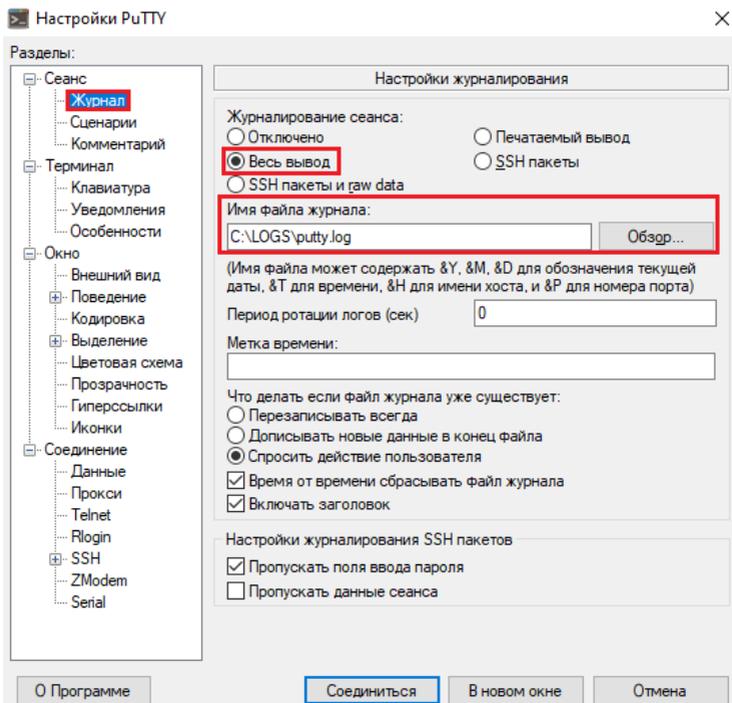


Рис. 2.80. Параметры журналирования

После того, как все параметры настроены, нажмите **“Соединиться”**. В появившемся окне будет отображаться отладочная информация.

Для получения расширенной диагностической информации требуется нажать клавишу клавиатуры ‘o’ (в английской раскладке), а затем Enter.

Один раз в 10 секунд будут выводиться сообщения со статистикой всех задач. Информация о работе системы архивирования будет выдаваться один раз в минуту.

```
192.168.10.58 - PuTTY
4477902: |-1538358192| [TRA] (COM2) Rx: [9] 06 03 04 00 32 00 00
2D 3C 4477902: |-1538358192| [TRA] (COM2) Tx: [8] 06 03 04 10 00 02 C5 49
4477907: |-1538358192| [TRA] (COM2) Rx: [9] 06 03 04 93 E0 00 04 A1 82
907: |-1538358192| [INF] (COM2) <Modbus> Read done
4477907: |-1538358192| [INF] (COM2) <Modbus> Address 6 ID 65410. Read start 2024-06-11T13:38:37.20
4016 4477908: |-1538358192| [TRA] (COM2) Tx: [8] 06 03 00 20 00 04 44 74
77913: |-1538358192| [TRA] (COM2) Rx: [13] 06 03 08 00 01 00 01 00 01 00 0
1 32 A3 4477913: |-1538358192| [TRA] (COM2) Tx: [8] 06 03 00 30 00 04 45 B
1 4477917: |-1538358192| [TRA] (COM2) Rx: [13] 06 03 08 00 96 00 96 00 9A
00 00 11 98 4477918: |-1538358192| [TRA] (COM2) Tx: [8] 06 03 00 40 00 04
44 6A 4477922: |-1538358192| [TRA] (COM2) Rx: [13] 06 03 08 00 01 00 01 00
01 00 01 32 A3 4477922: |-1538358192| [TRA] (COM2) Tx: [8] 06 03 00 50 00
04 45 AF 4477927: |-1538358192| [TRA] (COM2) Rx: [13] 06 03 08 00 01 00 0
1 00 01 00 01 32 A3 4477927: |-1538358192| [TRA] (COM2) Tx: [8] 06 03 00 6
0 00 01 85 A3 4477931: |-1538358192| [TRA] (COM2) Rx: [7] 06 03 02 00 08 4
C 43 4477932: |-1538358192| [TRA] (COM2) Tx: [8] 06 03 00 70 00 01 84 66
4477932: |-1538358192| [TRA] (COM2) Rx: [7] 06 03 01 00 50 50 06 02
```

Рис. 2.81. Окно с расширенной диагностической информацией

2.10.3. Специальные ФБ

Для получения диагностической информации встроенными средствами MasterSCADA 4D можно использовать специальные функциональные блоки библиотеки BaseObjects:

- **GetSystemInfo** - получение информации об установленной версии исполнительной системы и загруженном проекте
- **RedundancyControl** - управление резервированием.
- **FileIntegrityControl** - управление контролем целостности ПО и проекта.
- **GetOpcUaServerInfo** - получения информации и статистики по работе OPC UA сервера.
- **GetDataArchiveStatistic** - получения информации и статистики по работе архива данных.
- **GetEventsArchiveStatistic** - получения информации и статистики по работе архива данных.
- **GetTaskStatistics** - получение статистики работы задач.

Подробное описание функциональных блоков приведено в основной справочной информации среды разработки MasterSCADA 4D.

3. Работа с модулями ввода-вывода

3.1. Рекомендации для работы с ПЛК и модулями серии NL, NLS

1. Рекомендуется устанавливать (по возможности) скорость обмена по шине 115200 бод, а также использовать контрольные суммы.
2. Модули аналогового ввода (NLS-8AI, NLS-4RTD и др.) способны производить аналого-цифровое преобразование со скоростью около 10 раз в секунду. Это значит, что частота опроса не должна превышать 10 Гц.
3. Датчик NL-1S111 способен работать только на скорости обмена 9600 бод. Частота опроса не более 1 Гц. Рекомендуется подключать эти датчики в отдельную сеть, чтобы они не мешали работе более скоростных модулей ввода-вывода.
4. Модули ввода-вывода серии NL имеют собственный сторожевой таймер, который настраивается при конфигурировании модуля. Принцип действия этого таймера основан на передаче модулям по шине данных специальной широкополосной команды «00 06 0A 02 00 01 [CRC16]». Приняв такую команду модуль сбрасывает свой сторожевой таймер. Если таймер сработал до прихода команды, модуль переходит в защищенное состояние, которое также настраивается при конфигурировании модуля.
5. Для ускорения опроса модулей ввода-вывода (и для ускорения работы ПЛК соответственно) рекомендуется разделить эти модули на «скоростные» (те, которые быстро отвечают, как правило – дискретные) и «обычные» (те, которым на обработку команды и ответ требуется 100 мс и более, как правило – аналоговые). «Скоростные» модули рекомендуется подключить к одной шине данных, «обычные» – к другой. В конфигурации задач также рекомендуется разделить опрос «скоростных» и «обычных» модулей по задачам. Т.е. одна задача запускает программный модуль, который использует данные каналов «скоростных» модулей ввода-вывода, другая задача запускает программный модуль, опрашивающий «обычные» модули. Данные между программными модулями можно передавать с помощью глобальных переменных. Разделение алгоритма работы ПЛК по задачам позволит организовать одновременную работу обеих шин данных, а также установить периоды повторения задач, свойственные модулям на шине, с которыми задача взаимодействует. Разделение модулей по шинам данных на «скоростные» и «обычные» позволит работать модулям на своей скорости и не ожидать ответов более медленных модулей. Такое замедление становится особо заметным, когда «обычный» модуль вдруг пропускает запрос (запрос при этом приходится повторять), а «скоростной»

модуль вынужден ожидать (довольно длительный таймаут), когда ПЛК обратится к нему.

3.2. Протокол Modbus RTU

Проекты с примерами работы с устройствами в среде разработки MasterSCADA 4D можно скачать на нашем сайте reallab.ru.

Modbus RTU – это стандартный протокол обмена по MODBUS по последовательному интерфейсу. При обмене узел играет роль ведущего (Master).

Для того, чтобы работать с одним или несколькими модулями RealLab создайте проект, подключите библиотеку RealLab и добавьте необходимые устройства в проект (см. п. 2.3.3). Настройте добавленные в проект протоколы, модули и ПЛК – установите правильные IP-адреса, для протокола выберите скорость обмена данными и порт.

Также после добавления шаблона устройства в проект имеется возможность добавления и удаления каналов устройства и изменения их параметров.

ВАЖНО! Если в наборе модулей проекта, работающих по протоколу Modbus RTU, преобладают модули, которые работают на запись и скорость записи вам важнее скорости чтения, включите у протокола ModbusRTU настройку "Ограничивать время чтения периодом задачи". Эта настройка включает приоритет записи над чтением и работает по следующему алгоритму:

- 1) Вход в Execute протокола.
- 2) Проход по всем модулям протокола и поиск выходов на запись - записываем их в устройство.
- 3) Начинается опрос устройств.
- 4) Прочитав устройство, проверяется не истекло ли время задачи.
- 5) Если истекло - запоминается номер модуля, на котором опрос остановился. Записываются все считанные за текущий цикл задачи значения в теги.
- 6) Выход из задачи.

При следующем входе в Execute опять проводится запись значений (если таковые есть) и читается следующий по списку модуль.

ВАЖНО! В некоторых модулях для записи нулевого значения необходимо записать сначала любое другое, а затем нулевое. Это обусловлено тем, что при использовании способа записи "По изменению" изначально все

значения выходов в тегах равны нулю и при повторной записи нуля значение не записывается т.к. оно не изменилось.

Для удобства управления модулем также можно добавить визуализацию (см. п. 2.4).

3.2.1. Пример использования NLS-16DI

Пример использования модуля содержит 16 каналов дискретных входов (DI) и окно визуализации с индикаторами.

Шаблонный проект NLS-16DI содержит следующие каналы:

Имя канала	Регион	Адрес	Тип в усг	Тип в SCADA	Доступ
DI 0	DISCRETE_INPUTS	0 (0x0000)	BOOL	BOOL	ReadOnly
DI 1	DISCRETE_INPUTS	1 (0x0001)	BOOL	BOOL	ReadOnly
DI 2	DISCRETE_INPUTS	2 (0x0002)	BOOL	BOOL	ReadOnly
DI 3	DISCRETE_INPUTS	3 (0x0003)	BOOL	BOOL	ReadOnly
DI 4	DISCRETE_INPUTS	4 (0x0004)	BOOL	BOOL	ReadOnly
DI 5	DISCRETE_INPUTS	5 (0x0005)	BOOL	BOOL	ReadOnly
DI 6	DISCRETE_INPUTS	6 (0x0006)	BOOL	BOOL	ReadOnly
DI 7	DISCRETE_INPUTS	7 (0x0007)	BOOL	BOOL	ReadOnly
DI 8	DISCRETE_INPUTS	8 (0x0008)	BOOL	BOOL	ReadOnly
DI 9	DISCRETE_INPUTS	9 (0x0009)	BOOL	BOOL	ReadOnly
DI 10	DISCRETE_INPUTS	10 (0x000A)	BOOL	BOOL	ReadOnly
DI 11	DISCRETE_INPUTS	11 (0x000B)	BOOL	BOOL	ReadOnly
DI 12	DISCRETE_INPUTS	12 (0x000C)	BOOL	BOOL	ReadOnly
DI 13	DISCRETE_INPUTS	13 (0x000D)	BOOL	BOOL	ReadOnly
DI 14	DISCRETE_INPUTS	14 (0x000E)	BOOL	BOOL	ReadOnly
DI 15	DISCRETE_INPUTS	15 (0x000F)	BOOL	BOOL	ReadOnly

Рис.3.1. Каналы NLS-16DI

На вход Din 12 подано напряжение 12 Вольт.



Рис.3.2. Визуализация модуля в проекте

3.2.2. Пример использования NLS-16DO

Пример использования модуля содержит 16 каналов дискретных выходов (DO) и окно визуализации с индикаторами и кнопками.

Используемый способ записи в проекте-примере – “Периодический” (можно использовать способ записи “По изменению”).

Шаблонный проект NLS-16DO содержит следующие каналы:

Имя канала	Регион	Адрес	Тип в уст	Тип в SCI	Доступ
DO 1	COILS	0 (0x0000)	BOOL	BOOL	ReadWrite
DO 2	COILS	1 (0x0001)	BOOL	BOOL	ReadWrite
DO 3	COILS	2 (0x0002)	BOOL	BOOL	ReadWrite
DO 4	COILS	3 (0x0003)	BOOL	BOOL	ReadWrite
DO 5	COILS	4 (0x0004)	BOOL	BOOL	ReadWrite
DO 6	COILS	5 (0x0005)	BOOL	BOOL	ReadWrite
DO 7	COILS	6 (0x0006)	BOOL	BOOL	ReadWrite
DO 8	COILS	7 (0x0007)	BOOL	BOOL	ReadWrite
DO 9	COILS	8 (0x0008)	BOOL	BOOL	ReadWrite
DO 10	COILS	9 (0x0009)	BOOL	BOOL	ReadWrite
DO 11	COILS	10 (0x000A)	BOOL	BOOL	ReadWrite
DO 12	COILS	11 (0x000B)	BOOL	BOOL	ReadWrite
DO 13	COILS	12 (0x000C)	BOOL	BOOL	ReadWrite
DO 14	COILS	13 (0x000D)	BOOL	BOOL	ReadWrite
DO 15	COILS	14 (0x000E)	BOOL	BOOL	ReadWrite
DO 16	COILS	15 (0x000F)	BOOL	BOOL	ReadWrite

Рис.3.3. Каналы NLS-16DO

На визуализации управления модулем NLS-16DO были нажаты кнопки, подающие на дискретные выходы 0, 1, 5, 9, 12 значение ‘1’.



Рис.3.4. Визуализация модуля в проекте

Индикаторы на визуализации имеют такие же состояния, как физические индикаторы на модуле, дискретные выходы 0, 1, 5, 9, 12 находятся в состоянии 'true'.

3.2.3. Пример использования NLS-8R

Пример использования модуля содержит 8 каналов релейных выходов и окно визуализации с индикаторами и кнопками.

Используемый способ записи в проекте-примере – “Периодический” (можно использовать способ записи “По изменению”).

Шаблонный проект NLS-8R содержит следующие каналы:

	Имя канала	Регион	Адрес	Тип в уст.	Тип в SCA	Доступ
▾	Device					
<input type="checkbox"/>	DO 1	COILS	0 (0x0000)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 2	COILS	1 (0x0001)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 3	COILS	2 (0x0002)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 4	COILS	3 (0x0003)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 5	COILS	4 (0x0004)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 6	COILS	5 (0x0005)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 7	COILS	6 (0x0006)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 8	COILS	7 (0x0007)	BOOL	BOOL	ReadWrite

Рис.3.5. Каналы NLS-8R

На визуализации управления модулем NLS-8R были нажаты кнопки, подающие на релейные выходы 1 и 7 значение '1'. Индикаторы на визуализации имеют такие же состояния, как физические индикаторы на модуле, выходы 1 и 7 находятся в состоянии 'true'.



Рис.3.6. Визуализация модуля NLS-8R

3.2.4. Пример использования NLS-4AO

Пример использования модуля содержит 4 канала аналоговых выходов, по одному каналу диапазонов, PowerOn-значений и SafeValue-значений на один канал аналогового выхода. Также проект содержит окно визуализации со значениями выходов, выпадающими списками для выбора диапазонов и полями для ввода новых значений.

Используемый способ записи в проекте-примере – “**По изменению**”. Способ записи “**Периодический**” использовать рекомендуется только с триггерами (F_TRIG, R_TRIG) для изменения значений диапазонов каналов.

Шаблонный проект NLS-4AO содержит следующие каналы:

Имя канала	Регион	Адрес	Тип в уст	Тип в SCJ	Доступ
Device					
AnalogueOutput 0	HOLDING_REGISTERS	0 (0x0000)	WORD	WORD	ReadWrite
AnalogueOutput 1	HOLDING_REGISTERS	1 (0x0001)	WORD	WORD	ReadWrite
AnalogueOutput 2	HOLDING_REGISTERS	2 (0x0002)	WORD	WORD	ReadWrite
AnalogueOutput 3	HOLDING_REGISTERS	3 (0x0003)	WORD	WORD	ReadWrite
RANGE 0	HOLDING_REGISTERS	16 (0x0010)	WORD	WORD	ReadWrite
RANGE 1	HOLDING_REGISTERS	17 (0x0011)	WORD	WORD	ReadWrite
RANGE 2	HOLDING_REGISTERS	18 (0x0012)	WORD	WORD	ReadWrite
RANGE 3	HOLDING_REGISTERS	19 (0x0013)	WORD	WORD	ReadWrite
PowerOn 0	HOLDING_REGISTERS	768 (0x0300)	WORD	WORD	ReadWrite
PowerOn 1	HOLDING_REGISTERS	769 (0x0301)	WORD	WORD	ReadWrite
PowerOn 2	HOLDING_REGISTERS	770 (0x0302)	WORD	WORD	ReadWrite
PowerOn 3	HOLDING_REGISTERS	771 (0x0303)	WORD	WORD	ReadWrite
SafeValue 0	HOLDING_REGISTERS	784 (0x0310)	WORD	WORD	ReadWrite
SafeValue 1	HOLDING_REGISTERS	785 (0x0311)	WORD	WORD	ReadWrite
SafeValue 2	HOLDING_REGISTERS	786 (0x0312)	WORD	WORD	ReadWrite
SafeValue 3	HOLDING_REGISTERS	787 (0x0313)	WORD	WORD	ReadWrite

Рис.3.7. Каналы NLS-4AO

Визуализация управления модулем позволяет изменять значения аналоговых выходов, их диапазоны, PowerOn и SafeValue-значения.

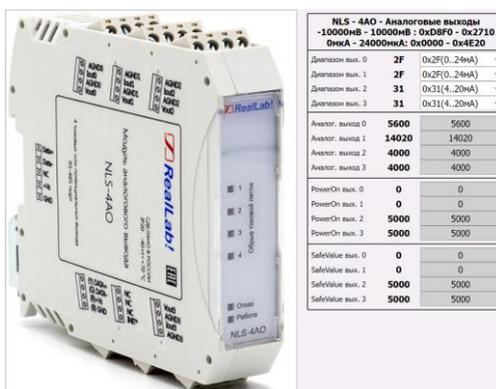


Рис.3.8. Визуализация модуля в проекте

3.2.5. Пример использования NLS-8TI

Пример использования модуля содержит 8 каналов аналоговых входов, по одному каналу диапазонов на один канал аналогового входа, канал чтения/записи маски входов, канал чтения температуры холодного спая. Также проект содержит окно визуализации со значениями входов, выпадающими списками для выбора диапазонов и программу для записи маски входных каналов.

Используемый способ записи в проекте-примере – “По изменению”. Способ записи “Периодический” использовать рекомендуется только с триггерами (F_TRIG, R_TRIG) для изменения значений диапазонов каналов.

Шаблонный проект NLS-8TI содержит следующие каналы:

	Имя канала	Регион	Адрес	Тип в уст	Тип в SCADA	Доступ
<input type="checkbox"/>	Device					
<input type="checkbox"/>	AI 0	INPUT_REGISTERS	32 (0x0020)	REAL	REAL	ReadOnly
<input type="checkbox"/>	AI 1	INPUT_REGISTERS	34 (0x0022)	REAL	REAL	ReadOnly
<input type="checkbox"/>	AI 2	INPUT_REGISTERS	36 (0x0024)	REAL	REAL	ReadOnly
<input type="checkbox"/>	AI 3	INPUT_REGISTERS	38 (0x0026)	REAL	REAL	ReadOnly
<input type="checkbox"/>	AI 4	INPUT_REGISTERS	40 (0x0028)	REAL	REAL	ReadOnly
<input type="checkbox"/>	AI 5	INPUT_REGISTERS	42 (0x002A)	REAL	REAL	ReadOnly
<input type="checkbox"/>	AI 6	INPUT_REGISTERS	44 (0x002C)	REAL	REAL	ReadOnly
<input type="checkbox"/>	AI 7	INPUT_REGISTERS	46 (0x002E)	REAL	REAL	ReadOnly
<input type="checkbox"/>	MASK	HOLDING_REGISTERS	1536 (0x0600)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE	HOLDING_REGISTERS	514 (0x0202)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ColdJunctionTemp	INPUT_REGISTERS	16 (0x0010)	WORD	WORD	ReadOnly
<input type="checkbox"/>	RANGE_0	HOLDING_REGISTERS	1792 (0x0700)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE_1	HOLDING_REGISTERS	1793 (0x0701)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE_2	HOLDING_REGISTERS	1794 (0x0702)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE_3	HOLDING_REGISTERS	1795 (0x0703)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE_4	HOLDING_REGISTERS	1796 (0x0704)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE_5	HOLDING_REGISTERS	1797 (0x0705)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE_6	HOLDING_REGISTERS	1798 (0x0706)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE_7	HOLDING_REGISTERS	1799 (0x0707)	WORD	WORD	ReadWrite

Рис.3.9. Каналы модуля NLS-8TI

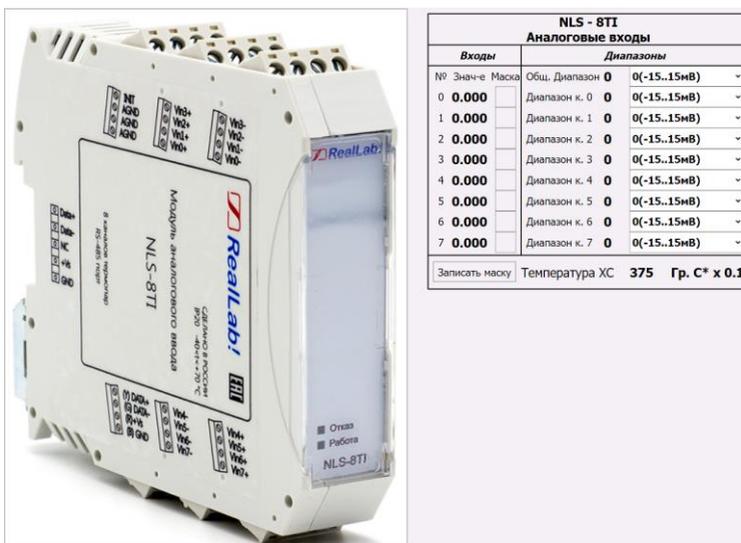


Рис.3.10. Визуализация модуля в проекте

3.2.6. Пример использования NLS-8AI-I

Пример использования модуля содержит 8 каналов аналоговых входов, канал чтения/записи маски входов. Также проект содержит окно визуализации со значениями входов и программу для записи маски входных каналов.

Используемый способ записи в проекте-примере – “По изменению”.

Шаблонный проект NLS-8AI-I содержит следующие каналы:

	Имя канала	Регион	Адрес	Тип в уст	Тип в SCU	Доступ
<input type="checkbox"/>	Device					
<input type="checkbox"/>	MASK	HOLDING_REGISTERS	1536 (0x0600)	WORD	WORD	ReadWrite
<input type="checkbox"/>	A10	INPUT_REGISTERS	32 (0x0020)	REAL	REAL	ReadOnly
<input type="checkbox"/>	A11	INPUT_REGISTERS	34 (0x0022)	REAL	REAL	ReadOnly
<input type="checkbox"/>	A12	INPUT_REGISTERS	36 (0x0024)	REAL	REAL	ReadOnly
<input type="checkbox"/>	A13	INPUT_REGISTERS	38 (0x0026)	REAL	REAL	ReadOnly
<input type="checkbox"/>	A14	INPUT_REGISTERS	40 (0x0028)	REAL	REAL	ReadOnly
<input type="checkbox"/>	A15	INPUT_REGISTERS	42 (0x002A)	REAL	REAL	ReadOnly
<input type="checkbox"/>	A16	INPUT_REGISTERS	44 (0x002C)	REAL	REAL	ReadOnly
<input type="checkbox"/>	A17	INPUT_REGISTERS	46 (0x002E)	REAL	REAL	ReadOnly

Рис.3.11. Каналы модуля NLS-8AI-I



Рис.3.12. Визуализация модуля в проекте

3.2.7. Пример использования NLS-4RTD

Пример использования модуля содержит 4 канала аналоговых входов, канал общих диапазонов, канал чтения/записи маски входов, канал чтения/записи формата читаемых данных (температура/сопротивление). Также проект содержит окно визуализации со значениями входов, выпадающими списками для выбора диапазона и формата данных, программу для записи маски входных каналов.

Используемый способ записи в проекте-примере – “По изменению”. Способ записи “Периодический” использовать рекомендуется только с триггерами (F_TRIG, R_TRIG) для изменения значений диапазонов каналов.

Шаблонный проект NLS-4RTD содержит следующие каналы:

	Имя канала	Регион	Адрес	Тип в уст	Тип в SCADA	Доступ
▲	Device					
<input type="checkbox"/>	ДанныеКанала 0	INPUT_REGISTERS	0 (0x0000)	WORD	WORD	ReadOnly
<input type="checkbox"/>	ДанныеКанала 1	INPUT_REGISTERS	1 (0x0001)	WORD	WORD	ReadOnly
<input type="checkbox"/>	ДанныеКанала 2	INPUT_REGISTERS	2 (0x0002)	WORD	WORD	ReadOnly
<input type="checkbox"/>	ДанныеКанала 3	INPUT_REGISTERS	3 (0x0003)	WORD	WORD	ReadOnly
<input type="checkbox"/>	ДиапазонОбщий	HOLDING_REGISTERS	514 (0x0202)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ФорматДанных	HOLDING_REGISTERS	515 (0x0203)	WORD	WORD	ReadWrite
<input type="checkbox"/>	MASK	HOLDING_REGISTERS	1536 (0x0600)	WORD	WORD	ReadWrite

Рис.3.13. Каналы модуля NLS-4RTD

	Имя канала	Регион	Адрес	Тип в уст	Тип в SC/	Доступ
	Device					
<input type="checkbox"/>	СчетныйВход 0	HOLDING_REGISTERS	0 (0x0000)	DWORD	DWORD	ReadOnly
<input type="checkbox"/>	СчетныйВход 1	HOLDING_REGISTERS	2 (0x0002)	DWORD	DWORD	ReadOnly
<input type="checkbox"/>	СчетныйВход 2	HOLDING_REGISTERS	4 (0x0004)	DWORD	DWORD	ReadOnly
<input type="checkbox"/>	СчетныйВход 3	HOLDING_REGISTERS	6 (0x0006)	DWORD	DWORD	ReadOnly
<input type="checkbox"/>	МакЭнСчет 0	HOLDING_REGISTERS	16 (0x0010)	DWORD	DWORD	ReadWrite
<input type="checkbox"/>	МакЭнСчет 1	HOLDING_REGISTERS	18 (0x0012)	DWORD	DWORD	ReadWrite
<input type="checkbox"/>	МакЭнСчет 2	HOLDING_REGISTERS	20 (0x0014)	DWORD	DWORD	ReadWrite
<input type="checkbox"/>	МакЭнСчет 3	HOLDING_REGISTERS	22 (0x0016)	DWORD	DWORD	ReadWrite
<input type="checkbox"/>	МинЗначСчет_0	HOLDING_REGISTERS	150 (0x0096)	DWORD	DWORD	ReadWrite
<input type="checkbox"/>	МинЗначСчет_1	HOLDING_REGISTERS	152 (0x0098)	DWORD	DWORD	ReadWrite
<input type="checkbox"/>	МинЗначСчет_2	HOLDING_REGISTERS	154 (0x009A)	DWORD	DWORD	ReadWrite
<input type="checkbox"/>	МинЗначСчет_3	HOLDING_REGISTERS	156 (0x009C)	DWORD	DWORD	ReadWrite
<input type="checkbox"/>	НижнПорогСигнСчет	HOLDING_REGISTERS	1024 (0x0400)	DWORD	DWORD	ReadWrite
<input type="checkbox"/>	ВерхПорогСигнСчет	HOLDING_REGISTERS	1040 (0x0410)	DWORD	DWORD	ReadWrite

Рис.3.16. Каналы NLS-4С, часть FLOAT

	Имя канала	Регион	Адрес	Тип в уст	Тип в SC/	Доступ
	Device					
<input type="checkbox"/>	ДлитФильтЛог0_0	HOLDING_REGISTERS	32 (0x0020)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ДлитФильтЛог0_1	HOLDING_REGISTERS	33 (0x0021)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ДлитФильтЛог0_2	HOLDING_REGISTERS	34 (0x0022)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ДлитФильтЛог0_3	HOLDING_REGISTERS	35 (0x0023)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ДлитФильтЛог1_0	HOLDING_REGISTERS	48 (0x0030)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ДлитФильтЛог1_1	HOLDING_REGISTERS	49 (0x0031)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ДлитФильтЛог1_2	HOLDING_REGISTERS	50 (0x0032)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ДлитФильтЛог1_3	HOLDING_REGISTERS	51 (0x0033)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусФильт_0	HOLDING_REGISTERS	64 (0x0040)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусФильт_1	HOLDING_REGISTERS	65 (0x0041)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусФильт_2	HOLDING_REGISTERS	66 (0x0042)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусФильт_3	HOLDING_REGISTERS	67 (0x0043)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусGate_0	HOLDING_REGISTERS	80 (0x0050)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусGate_1	HOLDING_REGISTERS	81 (0x0051)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусGate_2	HOLDING_REGISTERS	82 (0x0052)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусGate_3	HOLDING_REGISTERS	83 (0x0053)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ЧтениеУрЛог0	HOLDING_REGISTERS	96 (0x0060)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ЧтениеУрЛог1	HOLDING_REGISTERS	112 (0x0070)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СостДискрВых	HOLDING_REGISTERS	256 (0x0100)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон	HOLDING_REGISTERS	514 (0x0202)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусСигнализации	HOLDING_REGISTERS	516 (0x0204)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СчетчикОтветовНаКом	HOLDING_REGISTERS	521 (0x0209)	WORD	WORD	ReadWrite
<input type="checkbox"/>	НомерКаналаСигнализац	HOLDING_REGISTERS	532 (0x0214)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусСчет 0	HOLDING_REGISTERS	600 (0x0258)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусСчет 1	HOLDING_REGISTERS	601 (0x0259)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусСчет 2	HOLDING_REGISTERS	602 (0x025A)	WORD	WORD	ReadWrite
<input type="checkbox"/>	СтатусСчет 3	HOLDING_REGISTERS	603 (0x025B)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ВерхПорогСигнЧаст	HOLDING_REGISTERS	1072 (0x0430)	WORD	WORD	ReadWrite
<input type="checkbox"/>	НижнПорогСигнЧаст	HOLDING_REGISTERS	1056 (0x0420)	WORD	WORD	ReadWrite

Рис.3.17. Каналы NLS-4С, часть WORD

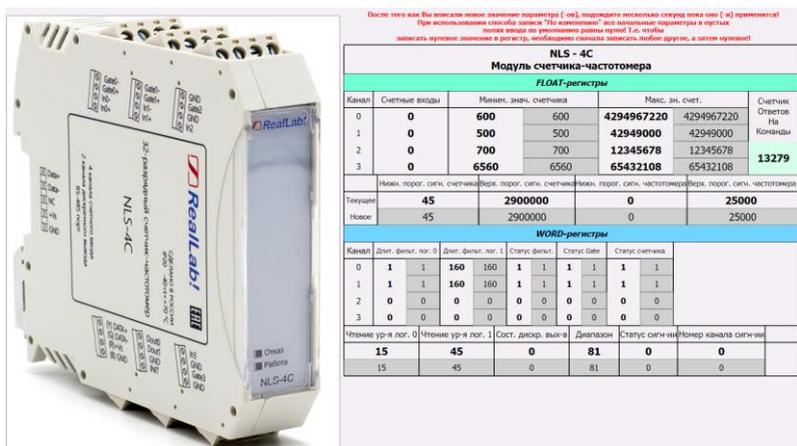


Рис.3.18. Визуализация модуля в проекте

3.3. Протокол Modbus TCP

Проекты с примерами работы с устройствами в среде разработки MasterSCADA 4D можно скачать на нашем сайте reallab.ru.

Modbus TCP – это протокол обмена по MODBUS по сети по протоколу TCP/IP. При обмене узел играет роль ведущего (Master).

Для того, чтобы работать с одним или несколькими модулями RealLab создайте проект, подключите библиотеку RealLab и добавьте необходимые устройства в проект (см. п. 2.3.3). Настройте добавленные в проект протоколы, модули и ПЛК – установите IP-адреса модулей, способ записи данных в свойствах протокола и модулей.

Также после добавления шаблона устройства в проект имеется возможность добавления и удаления каналов устройства и изменения их параметров.

ВАЖНО! В некоторых модулях для записи нулевого значения необходимо записать сначала любое другое, а затем нулевое. Это обусловлено тем, что при использовании способа записи "По изменению" изначально все значения выходов в тегах равны нулю и при повторной записи нуля значение не записывается т.к. оно не изменилось.

Для удобства управления модулем также можно добавить визуализацию (см. п. 2.4).

3.3.1. Пример использования NLS-16DI-Ethernet

Пример использования модуля содержит 16 каналов дискретных входов (DI) и окно визуализации с индикаторами.

При подаче напряжения на дискретный вход, индикатор задействованного входа будет светиться зеленым на визуализации.

Шаблонный проект NLS-16DI-Ethernet содержит следующие каналы:

Имя канала	Регион	Адрес	Тип в уст	Тип в SC	Доступ
Device					
DI 0	DISCRETE_INPUTS	0 (0x0000)	BOOL	BOOL	ReadOnly
DI 1	DISCRETE_INPUTS	1 (0x0001)	BOOL	BOOL	ReadOnly
DI 2	DISCRETE_INPUTS	2 (0x0002)	BOOL	BOOL	ReadOnly
DI 3	DISCRETE_INPUTS	3 (0x0003)	BOOL	BOOL	ReadOnly
DI 4	DISCRETE_INPUTS	4 (0x0004)	BOOL	BOOL	ReadOnly
DI 5	DISCRETE_INPUTS	5 (0x0005)	BOOL	BOOL	ReadOnly
DI 6	DISCRETE_INPUTS	6 (0x0006)	BOOL	BOOL	ReadOnly
DI 7	DISCRETE_INPUTS	7 (0x0007)	BOOL	BOOL	ReadOnly
DI 8	DISCRETE_INPUTS	8 (0x0008)	BOOL	BOOL	ReadOnly
DI 9	DISCRETE_INPUTS	9 (0x0009)	BOOL	BOOL	ReadOnly
DI 10	DISCRETE_INPUTS	10 (0x000A)	BOOL	BOOL	ReadOnly
DI 11	DISCRETE_INPUTS	11 (0x000B)	BOOL	BOOL	ReadOnly
DI 12	DISCRETE_INPUTS	12 (0x000C)	BOOL	BOOL	ReadOnly
DI 13	DISCRETE_INPUTS	13 (0x000D)	BOOL	BOOL	ReadOnly
DI 14	DISCRETE_INPUTS	14 (0x000E)	BOOL	BOOL	ReadOnly
DI 15	DISCRETE_INPUTS	15 (0x000F)	BOOL	BOOL	ReadOnly

Рис.3.19. Каналы модуля NLS-16DI-Ethernet



Рис.3.20. Визуализация модуля в проекте

3.3.2. Пример использования NLS-16DO-Ethernet

Пример использования модуля содержит 16 каналов дискретных выходов (DO) и окно визуализации с индикаторами и кнопками.

Используемый способ записи в проекте-примере – “Периодический” (можно использовать способ записи “По изменению”).

Шаблонный проект NLS-16DO-Ethernet содержит следующие каналы:

Имя канала	Регион	Адрес	Тип в уст	Тип в SCJ	Доступ
Device					
<input type="checkbox"/> DO 1	COILS	0 (0x0000)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 2	COILS	1 (0x0001)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 3	COILS	2 (0x0002)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 4	COILS	3 (0x0003)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 5	COILS	4 (0x0004)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 6	COILS	5 (0x0005)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 7	COILS	6 (0x0006)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 8	COILS	7 (0x0007)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 9	COILS	8 (0x0008)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 10	COILS	9 (0x0009)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 11	COILS	10 (0x000A)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 12	COILS	11 (0x000B)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 13	COILS	12 (0x000C)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 14	COILS	13 (0x000D)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 15	COILS	14 (0x000E)	BOOL	BOOL	ReadWrite
<input type="checkbox"/> DO 16	COILS	15 (0x000F)	BOOL	BOOL	ReadWrite

Рис.3.21. Каналы модуля NLS-16DO-Ethernet



Рис.3.22. Визуализация модуля в проекте

Для того, чтобы изменить состояние дискретного выхода, на визуализации нажмите на кнопку с номером выхода рядом с индикаторами на передней

панели. При успешном изменении состояния выхода изменится и цвет индикатора.

3.3.3. Пример использования NLS-8R-Ethernet

Пример использования модуля содержит 8 каналов релейных выходов и окно визуализации с индикаторами и кнопками. Используемый способ записи в проекте-примере – “Периодический” (можно использовать способ записи “По изменению”).

Шаблонный проект NLS-8R-Ethernet содержит следующие каналы:

	Имя канала	Регион	Адрес	Тип в уст	Тип в SC	Доступ
	Device					
<input type="checkbox"/>	DO 1	COILS	0 (0x0000)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 2	COILS	1 (0x0001)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 3	COILS	2 (0x0002)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 4	COILS	3 (0x0003)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 5	COILS	4 (0x0004)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 6	COILS	5 (0x0005)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 7	COILS	6 (0x0006)	BOOL	BOOL	ReadWrite
<input type="checkbox"/>	DO 8	COILS	7 (0x0007)	BOOL	BOOL	ReadWrite

Рис.3.23. Каналы модуля NLS-8R-Ethernet

Для того, чтобы изменить состояние релейного выхода, на визуализации нажмите на кнопку с номером выхода рядом с индикаторами на передней панели. При успешном изменении состояния выхода изменится и цвет индикатора.



Рис.3.24. Визуализация модуля в проекте

3.3.4. Пример использования NLS-4RTD-Ethernet

Пример использования модуля содержит по 4 канала аналоговых входов на измерение температуры и сопротивления, по одному каналу настройки диапазона на каждый вход, канал чтения/записи маски входов, по одному каналу настройки схемы подключения на каждый вход.

Также проект содержит окно визуализации со значениями входов, выпадающими списками для выбора диапазона и т.д.

Используемый способ записи в проекте-примере – “По изменению”. Способ записи “Периодический” использовать рекомендуется только с триггерами (F_TRIG, R_TRIG) для изменения значений диапазонов каналов.

Шаблонный проект NLS-4RTD-Ethernet содержит следующие каналы:

Имя канала	Регион	Адрес	Тип в усг	Тип в SCJ	Доступ
<input type="checkbox"/> Device					
<input type="checkbox"/> Вход 0 (Сопр)	INPUT_REGISTERS	0 (0x0000)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 1 (Сопр)	INPUT_REGISTERS	2 (0x0002)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 2 (Сопр)	INPUT_REGISTERS	4 (0x0004)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 3 (Сопр)	INPUT_REGISTERS	6 (0x0006)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 0 (Темп)	INPUT_REGISTERS	64 (0x0040)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 1 (Темп)	INPUT_REGISTERS	66 (0x0042)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 2 (Темп)	INPUT_REGISTERS	68 (0x0044)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 3 (Темп)	INPUT_REGISTERS	70 (0x0046)	REAL	REAL	ReadOnly
<input type="checkbox"/> MASK	HOLDING_REGISTERS	9312 (0x2460)	WORD	WORD	ReadOnly
<input type="checkbox"/> Диапазон 0	HOLDING_REGISTERS	9216 (0x2400)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 1	HOLDING_REGISTERS	9217 (0x2401)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 2	HOLDING_REGISTERS	9218 (0x2402)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 3	HOLDING_REGISTERS	9219 (0x2403)	WORD	WORD	ReadWrite
<input type="checkbox"/> Схема подключения 0	HOLDING_REGISTERS	9442 (0x24E2)	WORD	WORD	ReadWrite
<input type="checkbox"/> Схема подключения 1	HOLDING_REGISTERS	9445 (0x24E5)	WORD	WORD	ReadWrite
<input type="checkbox"/> Схема подключения 2	HOLDING_REGISTERS	9448 (0x24E8)	WORD	WORD	ReadWrite
<input type="checkbox"/> Схема подключения 3	HOLDING_REGISTERS	9451 (0x24EB)	WORD	WORD	ReadWrite

Рис.3.25. Каналы модуля NLS-4RTD-Ethernet

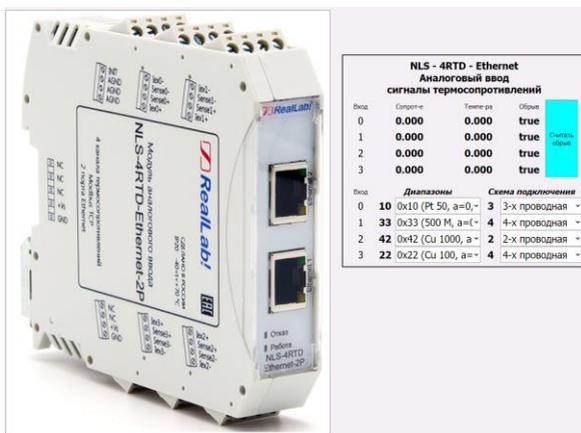


Рис.3.26. Визуализация модуля в проекте

3.3.5. Пример использования NLS-8AI-Ethernet

Пример использования модуля содержит 8 каналов аналоговых входов (16 в дифф. режиме), каналы чтения/записи диапазонов и т. д. Также проект содержит окно визуализации для управления модулем и считывания показаний.

Используемый способ записи в проекте-примере – **“По изменению”** (**“Периодический”** способ записи рекомендуется использовать только с триггерами (F_TRIG, R_TRIG)).

Шаблонный проект NLS-8AI-Ethernet содержит следующие каналы:

Имя канала	Регион	Адрес	Тип в уст.	Тип в SCJ	Доступ
<input type="checkbox"/> Device					
<input type="checkbox"/> Вход 0	INPUT_REGISTERS	0 (0x0000)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 1	INPUT_REGISTERS	2 (0x0002)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 2	INPUT_REGISTERS	4 (0x0004)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 3	INPUT_REGISTERS	6 (0x0006)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 4	INPUT_REGISTERS	8 (0x0008)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 5	INPUT_REGISTERS	10 (0x000A)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 6	INPUT_REGISTERS	12 (0x000C)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 7	INPUT_REGISTERS	14 (0x000E)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 8	INPUT_REGISTERS	16 (0x0010)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 9	INPUT_REGISTERS	18 (0x0012)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 10	INPUT_REGISTERS	20 (0x0014)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 11	INPUT_REGISTERS	22 (0x0016)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 12	INPUT_REGISTERS	24 (0x0018)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 13	INPUT_REGISTERS	26 (0x001A)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 14	INPUT_REGISTERS	28 (0x001C)	REAL	REAL	ReadOnly
<input type="checkbox"/> Вход 15	INPUT_REGISTERS	30 (0x001E)	REAL	REAL	ReadOnly

Рис.3.27. Каналы модуля NLS-8AI-Ethernet, ч. 1

<input type="checkbox"/> Диапазон 0	HOLDING_REGISTERS	9216 (0x2400)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 1	HOLDING_REGISTERS	9217 (0x2401)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 2	HOLDING_REGISTERS	9218 (0x2402)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 3	HOLDING_REGISTERS	9219 (0x2403)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 4	HOLDING_REGISTERS	9220 (0x2404)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 5	HOLDING_REGISTERS	9221 (0x2405)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 6	HOLDING_REGISTERS	9222 (0x2406)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 7	HOLDING_REGISTERS	9223 (0x2407)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 8	HOLDING_REGISTERS	9224 (0x2408)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 9	HOLDING_REGISTERS	9225 (0x2409)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 10	HOLDING_REGISTERS	9226 (0x240A)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 11	HOLDING_REGISTERS	9227 (0x240B)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 12	HOLDING_REGISTERS	9228 (0x240C)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 13	HOLDING_REGISTERS	9229 (0x240D)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 14	HOLDING_REGISTERS	9230 (0x240E)	WORD	WORD	ReadWrite
<input type="checkbox"/> Диапазон 15	HOLDING_REGISTERS	9231 (0x240F)	WORD	WORD	ReadWrite
<input type="checkbox"/> РежимРаботы	HOLDING_REGISTERS	9296 (0x2450)	WORD	WORD	ReadWrite

Рис.3.28. Каналы модуля NLS-8AI-Ethernet, ч. 2

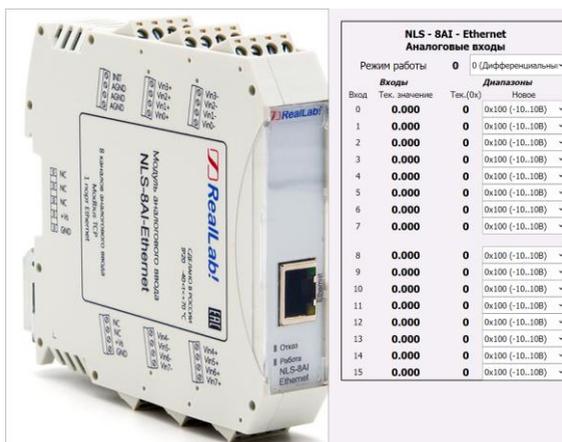


Рис.3.29. Визуализация модуля в проекте

3.3.6. Пример использования NLS-16AI-I-Ethernet

Пример использования модуля содержит 16 каналов аналоговых входов и каналы для настройки остальных параметров модуля.

Также проект включает в себя визуализацию для управления модулем.

Используемый способ записи в проекте-примере – **“По изменению”**.

Способ записи **“Периодический”** использовать рекомендуется только с триггерами (F_TRIG, R_TRIG) для изменения значений диапазонов каналов.

Шаблонный проект NLS-16AI-I-Ethernet содержит следующие каналы:

<input type="checkbox"/>	Вход 0	INPUT_REGISTERS	0 (0x0000)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 1	INPUT_REGISTERS	2 (0x0002)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 2	INPUT_REGISTERS	4 (0x0004)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 3	INPUT_REGISTERS	6 (0x0006)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 4	INPUT_REGISTERS	8 (0x0008)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 5	INPUT_REGISTERS	10 (0x000A)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 6	INPUT_REGISTERS	12 (0x000C)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 7	INPUT_REGISTERS	14 (0x000E)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 8	INPUT_REGISTERS	16 (0x0010)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 9	INPUT_REGISTERS	18 (0x0012)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 10	INPUT_REGISTERS	20 (0x0014)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 11	INPUT_REGISTERS	22 (0x0016)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 12	INPUT_REGISTERS	24 (0x0018)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 13	INPUT_REGISTERS	26 (0x001A)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 14	INPUT_REGISTERS	28 (0x001C)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Вход 15	INPUT_REGISTERS	30 (0x001E)	REAL	REAL	ReadOnly
<input type="checkbox"/>	Диапазон 0	HOLDING_REGISTERS	9216 (0x2400)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 1	HOLDING_REGISTERS	9217 (0x2401)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 2	HOLDING_REGISTERS	9218 (0x2402)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 3	HOLDING_REGISTERS	9219 (0x2403)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 4	HOLDING_REGISTERS	9220 (0x2404)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 5	HOLDING_REGISTERS	9221 (0x2405)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 6	HOLDING_REGISTERS	9222 (0x2406)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 7	HOLDING_REGISTERS	9223 (0x2407)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 8	HOLDING_REGISTERS	9224 (0x2408)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 9	HOLDING_REGISTERS	9225 (0x2409)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 10	HOLDING_REGISTERS	9226 (0x240A)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 11	HOLDING_REGISTERS	9227 (0x240B)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 12	HOLDING_REGISTERS	9228 (0x240C)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 13	HOLDING_REGISTERS	9229 (0x240D)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 14	HOLDING_REGISTERS	9230 (0x240E)	WORD	WORD	ReadWrite
<input type="checkbox"/>	Диапазон 15	HOLDING_REGISTERS	9231 (0x240F)	WORD	WORD	ReadWrite

Рис.3.30. Каналы модуля NLS-16AI-I-Ethernet



NLS - 16AI - I - Ethernet		
Аналоговые входы		
Входы		Диапазоны
Вход	Точность эн-е (мВ)	Точ(0x) Назов
0	0.000	0 0x120 (0.25 mA) ▾
1	0.000	0 0x120 (0.25 mA) ▾
2	0.000	0 0x120 (0.25 mA) ▾
3	0.000	0 0x120 (0.25 mA) ▾
4	0.000	0 0x120 (0.25 mA) ▾
5	0.000	0 0x120 (0.25 mA) ▾
6	0.000	0 0x120 (0.25 mA) ▾
7	0.000	0 0x120 (0.25 mA) ▾
8	0.000	0 0x120 (0.25 mA) ▾
9	0.000	0 0x120 (0.25 mA) ▾
10	0.000	0 0x120 (0.25 mA) ▾
11	0.000	0 0x120 (0.25 mA) ▾
12	0.000	0 0x120 (0.25 mA) ▾
13	0.000	0 0x120 (0.25 mA) ▾
14	0.000	0 0x120 (0.25 mA) ▾
15	0.000	0 0x120 (0.25 mA) ▾

Рис.3.31. Визуализация модуля в проекте

3.3.7. Пример использования NLS-8TI-Ethernet

Пример использования модуля содержит по 8 каналов аналоговых входов на измерение напряжения и температуры с аналоговых входов модуля, а также каналы для настройки остальных параметров модуля. Также проект содержит визуализацию для удобного управления модулем.

Используемый способ записи в проекте-примере – “По изменению”. Способ записи “Периодический” использовать рекомендуется только с триггерами (F_TRIG, R_TRIG) для изменения значений диапазонов каналов.

Шаблонный проект NLS-8TI-Ethernet содержит следующие каналы:

	Имя канала	Регион	Адрес	Тип в уст.	Тип в SC/	Доступ
	Device					
<input type="checkbox"/>	InVoltage 0	INPUT_REGISTERS	0 (0x0000)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InVoltage 1	INPUT_REGISTERS	2 (0x0002)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InVoltage 2	INPUT_REGISTERS	4 (0x0004)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InVoltage 3	INPUT_REGISTERS	6 (0x0006)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InVoltage 4	INPUT_REGISTERS	8 (0x0008)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InVoltage 5	INPUT_REGISTERS	10 (0x000A)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InVoltage 6	INPUT_REGISTERS	12 (0x000C)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InVoltage 7	INPUT_REGISTERS	14 (0x000E)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InTemperature 0	INPUT_REGISTERS	64 (0x0040)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InTemperature 1	INPUT_REGISTERS	66 (0x0042)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InTemperature 2	INPUT_REGISTERS	68 (0x0044)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InTemperature 3	INPUT_REGISTERS	70 (0x0046)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InTemperature 4	INPUT_REGISTERS	72 (0x0048)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InTemperature 5	INPUT_REGISTERS	74 (0x004A)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InTemperature 6	INPUT_REGISTERS	76 (0x004C)	REAL	REAL	ReadOnly
<input type="checkbox"/>	InTemperature 7	INPUT_REGISTERS	78 (0x004E)	REAL	REAL	ReadOnly
<input type="checkbox"/>	ColdJunctionTemp	INPUT_REGISTERS	128 (0x0080)	REAL	REAL	ReadOnly
<input type="checkbox"/>	MASK	HOLDING_REGISTERS	9312 (0x2460)	WORD	WORD	ReadOnly

Рис.3.32. Каналы модуля NLS-8TI-Ethernet, ч. 1

<input type="checkbox"/>	RANGE 0	HOLDING_REGISTERS	9216 (0x2400)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE 1	HOLDING_REGISTERS	9217 (0x2401)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE 2	HOLDING_REGISTERS	9218 (0x2402)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE 3	HOLDING_REGISTERS	9219 (0x2403)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE 4	HOLDING_REGISTERS	9220 (0x2404)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE 5	HOLDING_REGISTERS	9221 (0x2405)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE 6	HOLDING_REGISTERS	9222 (0x2406)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE 7	HOLDING_REGISTERS	9223 (0x2407)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ColdJunctionTemp_0/1	HOLDING_REGISTERS	9328 (0x2470)	WORD	WORD	ReadWrite
<input type="checkbox"/>	ТемператураXC_0_1	HOLDING_REGISTERS	9328 (0x2470)	WORD	WORD	ReadOnly

Рис.3.33. Каналы модуля NLS-8TI-Ethernet, ч. 2

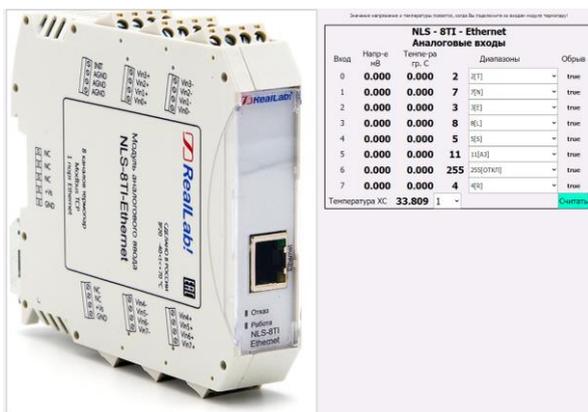


Рис.3.34. Визуализация модуля в проекте

3.3.8. Пример использования NLS-4AO-Ethernet

Пример использования модуля содержит 4 канала аналоговых выходов и множество каналов для настройки параметров значений. Также проект содержит окно визуализации для управления параметрами модуля и отслеживания получаемых показаний. Используемый способ записи в проекте-примере – “По изменению”. Способ записи “Периодический” использовать рекомендуется только с триггерами (F_TRIG, R_TRIG) для изменения значений диапазонов каналов.

Шаблонный проект NLS-4AO-Ethernet содержит следующие каналы:

Device						
<input type="checkbox"/>	Выход 0	HOLDING_REGISTERS	9472 (0x2500)	REAL	REAL	ReadWrite
<input type="checkbox"/>	Выход 1	HOLDING_REGISTERS	9474 (0x2502)	REAL	REAL	ReadWrite
<input type="checkbox"/>	Выход 2	HOLDING_REGISTERS	9476 (0x2504)	REAL	REAL	ReadWrite
<input type="checkbox"/>	Выход 3	HOLDING_REGISTERS	9478 (0x2506)	REAL	REAL	ReadWrite
<input type="checkbox"/>	PowerOn 0	HOLDING_REGISTERS	8304 (0x2070)	REAL	REAL	ReadWrite
<input type="checkbox"/>	PowerOn 1	HOLDING_REGISTERS	8306 (0x2072)	REAL	REAL	ReadWrite
<input type="checkbox"/>	PowerOn 2	HOLDING_REGISTERS	8308 (0x2074)	REAL	REAL	ReadWrite
<input type="checkbox"/>	PowerOn 3	HOLDING_REGISTERS	8310 (0x2076)	REAL	REAL	ReadWrite
<input type="checkbox"/>	SafeValue 0	HOLDING_REGISTERS	8336 (0x2090)	REAL	REAL	ReadWrite
<input type="checkbox"/>	SafeValue 1	HOLDING_REGISTERS	8338 (0x2092)	REAL	REAL	ReadWrite
<input type="checkbox"/>	SafeValue 2	HOLDING_REGISTERS	8340 (0x2094)	REAL	REAL	ReadWrite
<input type="checkbox"/>	SafeValue 3	HOLDING_REGISTERS	8342 (0x2096)	REAL	REAL	ReadWrite
<input type="checkbox"/>	RANGE 0	HOLDING_REGISTERS	9505 (0x2521)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE 1	HOLDING_REGISTERS	9506 (0x2522)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE 2	HOLDING_REGISTERS	9507 (0x2523)	WORD	WORD	ReadWrite
<input type="checkbox"/>	RANGE 3	HOLDING_REGISTERS	9508 (0x2524)	WORD	WORD	ReadWrite
<input type="checkbox"/>	SPEED 0	HOLDING_REGISTERS	9521 (0x2531)	WORD	WORD	ReadWrite
<input type="checkbox"/>	SPEED 1	HOLDING_REGISTERS	9522 (0x2532)	WORD	WORD	ReadWrite
<input type="checkbox"/>	SPEED 2	HOLDING_REGISTERS	9523 (0x2533)	WORD	WORD	ReadWrite
<input type="checkbox"/>	SPEED 3	HOLDING_REGISTERS	9524 (0x2534)	WORD	WORD	ReadWrite

Рис.3.35. Каналы модуля NLS-4AO-Ethernet

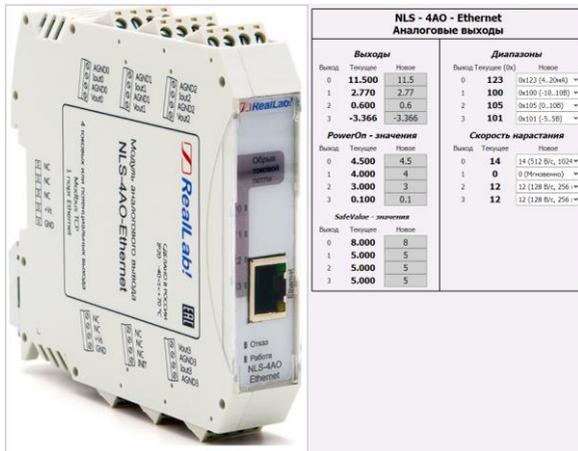


Рис.3.36. – Визуализация модуля в проекте