Research Laboratory
of Design Automation
НИЛ автоматизации проектирования
Сделано в России

Тел.: (495) 26-66-700, info@reallab.ru, www.reallab.ru

Среда разработки CODESYS 3.5.22.20

Программирование шлюзов RealLab в среде разработки CODESYS 3.5

Программирование шлюзов RealLab в среде разработки CODESYS 3.5.22.20

Руководство по эксплуатации

Версия от 31 августа 2026 г.

Одной проблемой стало меньше!

Уважаемый покупатель!

Научно-исследовательская лаборатория автоматизации проектирования (НИЛ АП) благодарит Вас за покупку и просит сообщать нам свои пожелания по улучшению этого руководства или описанной в нем продукции. 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, www.reallab.ru

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

Пожалуйста, внимательно изучите настоящее руководство. Это позволит вам в кратчайший срок и наилучшим образом использовать приобретенное изделие.

Оглавление

1. Вводная часть

1.1. Введение

Настоящее руководство по настройке и программированию содержит необходимую информацию для начала работы и программирования в среде CODESYS V3.5. Процесс ознакомления со средой CODESYS V3.5 описан применительно к шлюзам RealLab серии NLS-Gateway, но часть информации в данном документе является универсальной и может быть использована для обучения работе с любыми шлюзами, программируемыми в этой среде.

Документ соответствует версии среды разработки CODESYS V3.5 SP22 Patch 2.

Основной документ, содержащий информацию о программировании ПЛК и работе с модулями ввода-вывода RealLab в CODESYS V3.5, находится по ссылке.

1.2. Общие сведения о CODESYS V3.5

CODESYS представляет собой комплекс программ для проектирования прикладного ПО, отладки в режиме эмуляции и загрузки программы в шлюз. Основными частями системы являются среда разработки программы и среда ее исполнения, которая находится в шлюзе.

В CODESYS входят графические и текстовые редакторы для всех пяти языков МЭК 61131–3. Этот комплекс полностью реализует требования стандарта и дополнительно вводит ряд оригинальных расширений, самым удобным из которых является объектно-ориентированное программирование.

В одном проекте может быть использовано несколько контроллеров разных производителей. Каждый из них может программироваться как независимое устройство или с учетом их взаимодействия в промышленной сети. Проект состоит из нескольких приложений, распределенных по нескольким контроллерам. В одном шлюзе может существовать несколько независимых приложений.

Программа, написанная на языках МЭК, компилируется системой CODESYS в машинный код, оптимизированный для заданной аппаратной платформы. Компилятор выдает диагностические сообщения как на этапе компиляции, так и на этапе ввода операторов языка.

При отсутствии реального контроллера отладку программы можно выполнять с помощью встроенного программного эмулятора.

Система имеет также встроенный многоканальный программный трассировщик (графический самописец) значений переменных. Он позволяет наглядно представить динамически изменяющиеся данные проекта. Данные аккумулируются в памяти шлюза и могут синхронизироваться с определенными событиями. Трассировщик полезен не только при отладке, но и при анализе нештатных ситуаций в процессе эксплуатации оборудования.

Для того, чтобы шлюз можно было программировать с помощью CODESYS, в контроллере должна быть установлена система исполнения. Установку системы выполняет фирма изготовитель контроллера. Изготовитель обеспечивает также поддержку всех модулей шлюза, поэтому конечный пользователь может сосредоточиться на разработке только прикладной программы.

Помимо средств программирования, CODESYS имеет встроенную систему визуализации, которая применяется для операторского управления, а также моделирования на этапе разработки. Визуализацию можно запустить на компьютере, графической панели ПЛК или встроенном в шлюз web-сервере.

Пользователь может самостоятельно расширять возможность CODESYS путем создания библиотек программных модулей. Например, он может реализовать поддержку нестандартных интерфейсов.

Для систем, связанных с безопасностью, CODESYS имеет библиотеку функциональных блоков PLCopen Safety, систему исполнения для оборудования с дублированием и специализированное расширение среды программирования. При внезапном отключении питания CODESYS автоматически сохраняет значения переменных в энергонезависимую память (FRAM).

1.3. Программируемые шлюзы RealLab в CODESYS 3.5

Ниже приведен список шлюзов RealLab, программируемых в среде CODESYS V3.5.22.20:

- NLS-Gateway-LoRa-2RS-CDS;
- NLS-Gateway-LoRa-2RS-PoE-CDS;
- NLS-Gateway-LoRa-1RS-1CAN-CDS;
- NLS-Gateway-LoRa-1RS-1CAN-PoE-CDS;
- NLS-Gateway-Zigbee-2RS-CDS;
- NLS-Gateway-Zigbee-2RS-PoE-CDS;

– NLS-Gateway-Zigbee-1RS-1CAN-CDS;

– NLS-Gateway-Zigbee-1RS-1CAN-PoE-CDS.

1.3.1. Основные характеристики шлюза

– **Процессор**: Broadcom BCM2711 (4 ядра ARM Cortex-A72, тактовая частота 1.5 ГГц).

– **Операционная система**: RealLab! Raspbian Linux arm64.

– **Оперативная память (ОЗУ)**: 2 ГБ.

– **Системная память**: 16 ГБ eMMC (встроенная флеш-память).

– **Интерфейсы**: UART, I²C, SPI, CAN, Ethernet, USB, LoRa , WiFi, Bluetooth.

– **Питание**: 10–30 В постоянного тока (диапазон может варьироваться в зависимости от модели).

– **Рабочая температура**: от –20 °C до +60 °C.

– **Крепление**: DIN-рейка.

1.3.2. Программное обеспечение и поддерживаемые протоколы

Все шлюзы серии NLS-Gateway работают под управлением ОС RealLab! Raspbian Linux arm64. Состав предустановленного программного обеспечения и набор поддерживаемых протоколов зависят от модификации шлюза.

1.3.2.1. Программное обеспечение и средства разработки

Шлюзы предоставляют следующие программные средства для разработки и настройки:

– **Веб-конфигуратор RealLab** – для начальной настройки шлюза и сетевых параметров.

– **Языки программирования:** C/C++ (системное и прикладное ПО), Python (скрипты и автоматизация), JavaScript (веб-интерфейсы и пользовательские приложения).

Поддерживаемые беспроводные протоколы

Выбор беспроводного протокола определяется установленным модулем:

– Модели с индексом **«LoRa»** (например, NLS-Gateway-LoRa-…) поддерживают LoRa / LoRaWAN (диапазон 864–870 МГц, до 8 каналов приёма).

– Модели с индексом **«Zigbee»** (например, NLS-Gateway-Zigbee-…) поддерживают Zigbee 3.0 (стек Z-Stack 3.x, диапазон 2405–2480 МГц).

– Все шлюзы также имеют встроенный **Wi-Fi** (IEEE 802.11 b/g/n/ac) и Bluetooth 5.0 BLE, которые работают независимо от основного беспроводного модуля.

1.3.2.2. Поддерживаемые промышленные протоколы

Набор промышленных протоколов зависит от наличия интерфейса CAN в конкретной модели:

Все модели поддерживают:

- Modbus RTU – через порты RS-485;
- Modbus TCP – через порт Ethernet;
- MQTT – для обмена данными с облачными и локальными IoT-платформами (встроенный MQTT-брокер).

Модели с интерфейсом CAN (в названии присутствует «1RS-1CAN») дополнительно поддерживают протокол CANOpen.

Модели без CAN (с маркировкой «2RS») не поддерживают CANOpen.

Таким образом, тип беспроводного протокола определяется индексом (LoRa/Zigbee), а поддержка CANOpen – наличием CAN-интерфейса. Все остальные протоколы (Modbus, MQTT) и базовые средства разработки являются общими для всей серии шлюзов.

1.3.3. Интерфейсы шлюзов

Шлюзы серии NLS-Gateway оснащены широким набором проводных и беспроводных интерфейсов, обеспечивающих гибкость подключения к различным промышленным сетям и устройствам. Состав интерфейсов может незначительно варьироваться в зависимости от модели (наличие CAN, количество портов RS-485, тип беспроводного модуля). Перечень основных интерфейсов представлен в табл. 1.

Табл. 1. Интерфейсы шлюзов серии NLS-Gateway

Интерфейс	Описание	Примечание
Ethernet	Один порт 10/100/1000Base-T (RJ-45)	Для подключения к локальной сети, настройки, обмена данными по

		Modbus TCP, MQTT и другим протоколам.
RS-485	От 1 до 2 портов	Для подключения устройств по протоколу Modbus RTU. Порты имеют гальваническую изоляцию и защиту от перенапряжения.
CAN	1 порт (в моделях с опцией 1RS-1CAN)	Для подключения устройств по протоколу CANopen.
USB 2.0	1 порт Type-A	Для подключения внешних устройств (флешнакопители, адаптеры, ключи защиты).
Micro-SD	Слот для карт памяти	Поддержка карт microSD объёмом до 128 Гбайт. Используется для расширения системной памяти, хранения резервных копий и журналов.
SMA	2 разъёма	Для подключения внешних антенн беспроводных модулей (LoRa/Zigbee и Wi-Fi).
LoRa / LoRaWAN	Встроенный модуль (для LoRa-моделей)	Диапазон частот 864–870 МГц, до 8 каналов приёма, мощность до +27 дБм.
Zigbee 3.0	Встроенный модуль (для Zigbee-моделей)	Диапазон частот 2405–2480 МГц, 16 каналов, мощность до +20 дБм.
Wi-Fi	Встроенный модуль IEEE 802.11 b/g/n/ac	Поддержка режимов клиента и точки доступа.
Bluetooth	Bluetooth 5.0 BLE	Для подключения периферийных устройств и настройки.

Часы реального времени (RTC)	Встроенные	С автономным питанием, для временных метки событий.

1.3.4. Сетевые интерфейсы и настройка связи: Настройка Ethernet и Wi-Fi (Клиент / Точка доступа)

Шлюзы серии NLS-Gateway имеют два независимых сетевых интерфейса, работающих одновременно:

- **Ethernet (eth0)** – проводной интерфейс для подключения к локальной сети.
- **Wi-Fi (wlan0)** – беспроводной интерфейс, который может работать в одном из двух режимов: клиент (подключение к существующей Wi-Fi сети) или точка доступа (создание собственной сети для локального доступа).

1.3.4.1. Настройка Ethernet

По умолчанию интерфейс Ethernet настроен на получение IP-адреса по протоколу DHCP. При подключении к сети с DHCP-сервером шлюз автоматически получает динамический IP-адрес.

Для настройки **статического IP-адреса** через веб-конфигуратор RealLab, подробное смотрите пункт 2.10 «**Настройка параметров сети**» в руководстве пользователя по работе с ПО шлюзов RealLab! серии NLS-Gateway, руководство скачать на странице шлюза на сайте компании RealLab.

1.3.4.2. Настройка Wi-Fi

Режим «Клиент» (Client)

По умолчанию шлюз настроен на стандартный режим Wi-Fi клиента. В этом режиме шлюз подключается к существующей Wi-Fi сети. Для настройки:

1. Подключитесь к веб-интерфейсу шлюза.
2. Перейдите в раздел «Сеть», вкладка «Wi-Fi»).
3. Нажмите кнопку «Параметры».
4. Укажите режим Wi-Fi «Клиент».
5. Нажмите «Применить».
6. Выполните сканирование Wi-Fi сетей.

7. Из появившегося списка нажмите на строку с именем той сети, к которой хотите подключиться. При необходимости введите пароль для подключения. Нажмите «Подключиться».
8. Настройте получение IP-адреса (DHCP или статический), если это необходимо.

Подробную информацию по настройке смотрите в руководстве пользователя по работе с ПО шлюзов RealLab! серии NLS-Gateway в пункте «Настройки IPv4 для Wi-Fi».

Режим «Точка доступа» (Access Point)

Режим «Точка доступа» позволяет подключаться к шлюзу напрямую без наличия внешней сети. Для настройки:

1. Для включения точки доступа перейдите в веб-конфигуратор RealLab – раздел «Сеть» – вкладка «Wi-Fi».
2. Нажмите кнопку «Параметры» и выберите режим «Точка доступа».
3. Введите имя сети, пароль, IP-адрес шлюза в созданной сети.
4. Нажмите кнопку «Применить».

Для изменения параметров точки доступа (SSID, пароль, канал) используется веб-конфигуратор шлюза. подробное смотрите пункт 2.10.2 «**Настройка Wi-Fi**» в руководстве пользователя по работе с ПО шлюзов RealLab! серии NLS-Gateway.

1.3.4.3. Одновременная работа интерфейсов

Интерфейсы Ethernet и Wi-Fi работают параллельно и не мешают друг другу. Это позволяет организовать резервирование каналов связи или разделить потоки данных (например, Ethernet – для промышленных протоколов, Wi-Fi – для удалённого мониторинга и настройки).

1.4. Актуальный состав лицензий

В поставку шлюза NLS-Gateway с ПО Codesys входят следующие лицензии:

– Лицензия на среду исполнения CoDeSys Control Runtime;

– Лицензия на пакет библиотек интернета вещей (CODESYS IIoT Libraries SL).

Пакет библиотек «CODESYS IIoT Libraries SL» включает в себя:

- **Web Client SL** – содержит функциональные блоки для связи с веб-сервером через HTTP или HTTPS;
- **MQTT Client SL** – позволяет отправлять сообщения с устройства с CODESYS брокеру MQTT, а также подписываться на топики;
- **Mail Service SL** – библиотека содержит функциональные блоки для отправки, получения и удаления писем с помощью протоколов SMTP и POP3. Связь с почтовым сервером может быть установлена как зашифрованной (TLS), так и незашифрованной;
- **SNMP Service SL** – поддерживаются версии V1, V2c, V3 протокола SNMP;
- **SNTP Service SL** – содержит функциональные блоки для простой реализации клиентских и серверных компонентов SNTP на устройстве с CODESYS (SNTP V3, SNTP V4);
- **AWS IoT Core Client SL** – библиотека предоставляет функциональные блоки для отправки и получения сообщений. Связь шифруется, и осуществляется с помощью протокола MQTT. В среде AES сообщения обычно передаются в формате JSON. Библиотека "JSON Utilities" может использоваться для обработки и создания JSON-файлов;
- **Azure IoT Hub Client SL** – библиотека предоставляет функциональные блоки для отправки и получения сообщений;
- **JSON Web Token SL** – библиотека содержит функциональный блок для создания JWT (JSON Web Token) на контроллере. Для этой цели поддерживаются алгоритмы HS256, HS384, HS512 и RS256;
- **Web Socket Client SL** – протокол WebSocket позволяет осуществлять двунаправленную связь между клиентом и веб-сокетным сервером через Интернет. Библиотека предоставляет функциональные блоки для связи через протокол WebSocket;
- **OpenWeather Client SL** – с помощью OpenWeather Client SL Library можно запрашивать погодные данные с API-сервера, предоставленного OpenWeather. Для использования нужен API-ключ OpenWeather;
- **CSV Utility SL** – предоставляет функциональные блоки для чтения и записи CSV-файлов;
- **INI File Utility SL** – библиотека для чтения и записи INI-файлов;
- **JSON Utilities SL** – библиотека для обработки JSON-файлов (чтение, изменение, создание, поиск по ключам);

– **XML Utility SL** – содержит функциональные блоки для чтения и записи xml-файлов или xml-строк на устройстве с CODESYS;

1.5. Преимущества интеграции среды исполнения CODESYS 3.5 на шлюзы серии NLS-Gateway

Использование CoDeSys наделяет NLS-Gateway функциями полноценного промышленного контроллера для работы с проводными и беспроводными сетями:

– **CODESYS 3.5** – мощная и гибкая среда разработки. Поддерживается программирование на языках стандарта МЭК 61131-3. Это позволяет разработчикам использовать наиболее подходящий язык для каждой задачи;

– Шлюзы серии NLS-Gateway построены на базе **процессора Broadcom BCM2711**. Производительности этого процессора достаточно для выполнения сложных алгоритмов управления и обработки данных в реальном времени;

– Глубокая интеграция с промышленными протоколами – **Modbus RTU/TCP, OPC UA, CANopen**;

– Поддержка работы с устройствами интернета вещей по **протоколу MQTT**. Это позволяет обрабатывать данные, получаемые от датчиков по протоколам LoRaWAN и Zigbee;

– **Встроенные интерфейсы**: RS-485, CAN, Ethernet, LoRa-концентратор / Zigbee-координатор;

2. Начало работы

Чтобы начать работу со средой разработки CODESYS, необходимо скачать и установить среду разработки CODESYS V3.5 SP22 Patch 2, а также установить таргет-файл RealLab с шаблонами модулей ввода и вывода и библиотеками для взаимодействия с ними. Последнюю версию таргет-файла можно загрузить с сайта RealLab. Установите загруженный пакет, создайте проект и подключитесь к шлюзу. На данном этапе все подготовлено для начала разработки в CODESYS 3.5. Все инструкции по установке и настройке CODESYS находятся в разделе 3 данного документа.

2.1. Установка среды разработки CODESYS 3.5

2.1.1. Системные требования

Для корректной работы среды разработки CODESYS 3.5 требуется:

– Операционная система Windows 7 или выше;
– 4 Гб оперативной памяти;
– 3 Гб свободного места на HDD;
– Процессор: Pentium V, Centrino > 3,0 GHz, Pentium M > 1,5GHz.

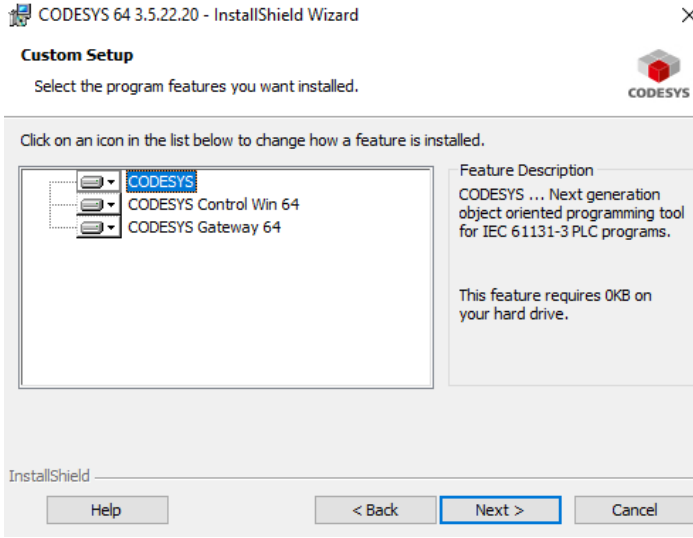


Рис. 2.1. Установка CODESYS 3.5

Для установки среды разработки CODESYS V3.5 SP22 Patch 2, скачайте установщик с нашего сайта и запустите его. В процессе установки CODESYS V3.5 предоставляется возможность выбора основных компонентов для установки (рис. 2.1). Рекомендуется установить все доступные компоненты.

2.2. Установка таргет-файла RealLab

Для упрощенной работы со шлюзом и модулями ввода и вывода RealLab! имеется таргет-файл с шаблонами, облегчающими процесс их добавления и настройки в среде CODESYS 3.5.

Все дополнительные компоненты устанавливаются отдельно. Скачайте с нашего сайта или с сайта CODESYS необходимые пакеты установки и запустите CODESYS.

В верхнем меню выберите «Инструменты – CODESYS Installer». Откроется окно со списком компонентов, установленных в CODESYS этой версии (рис. 2.2).

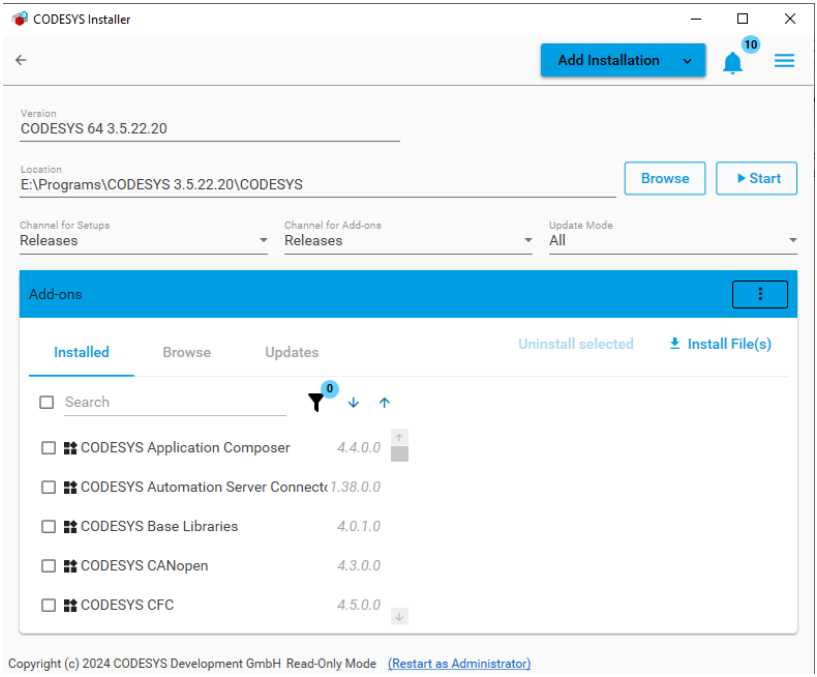


Рис. 2.2. CODESYS Installer

Нажмите «Install File(s)» и выберите ваш пакет в формате «.package» для установки. Закройте все окна среды разработки CODESYS, кроме CODESYS Installer. Следуйте инструкциям установщика. По окончании процесса откройте CODESYS и проверьте работоспособность установленных пакетов.

После установки пакета с шаблонами, таргет-файлами и библиотеками, при создании проекта нужно выбрать устройство «CODESYS Control for NLS-Gateway v4.20» из списка (см. рис. 2.3):

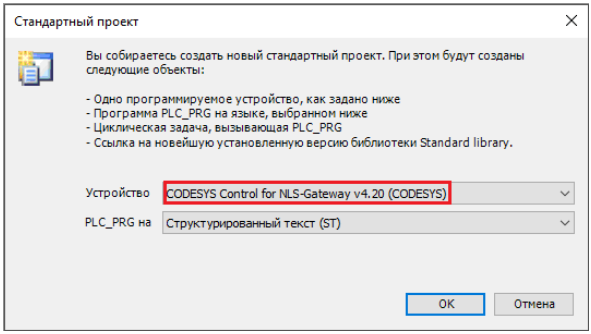


Рис. 2.3. Выбор шлюза из списка

Добавьте в проект Master-устройство для подключения модуля ввода-вывода соответствующего протокола – для этого кликните правой кнопкой мыши по «Device» в верхней части дерева устройств. В контекстном меню выберите «Добавить устройство» и выберите Master-устройство для добавления (см. рис. 2.4).

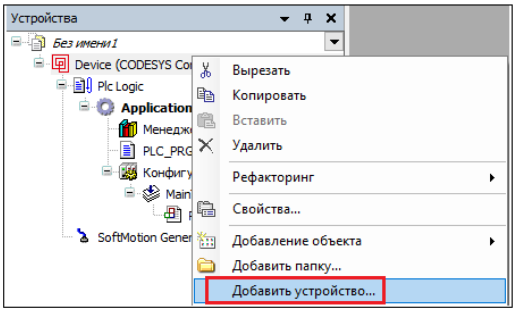


Рис. 2.4. Добавление Master-устройства

Далее вызовите контекстное меню Master-устройства, кликнув по нему правой кнопкой мыши и добавьте необходимые модули ввода-вывода.

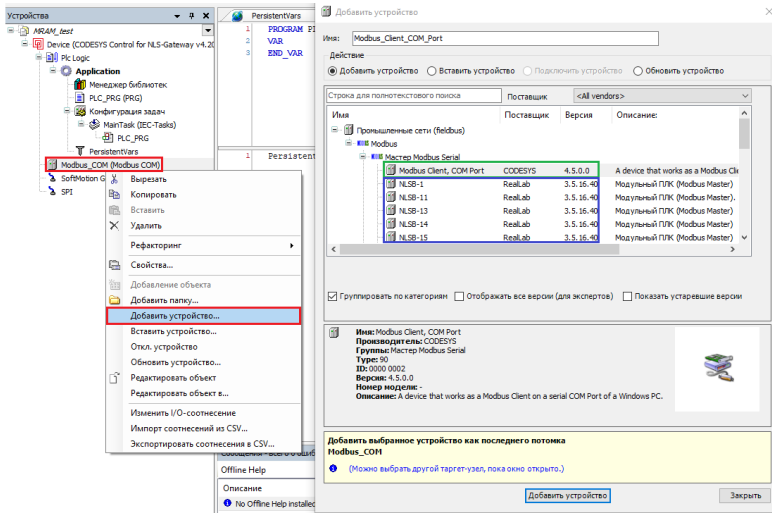


Рис. 2.5. Добавление модулей в проект

Для установки библиотек из верхнего меню CODESYS выберите «Инструменты – Репозиторий библиотек». Откроется окно репозитория библиотек (рис. 2.6). Чтобы установить необходимую библиотеку в формате «.library» или «.compiled-library» нажмите «Установить», выберите библиотеку, которую необходимо установить и следуйте инструкциям установщика. Для удаления/экспортирования библиотек используйте соответствующие кнопки.

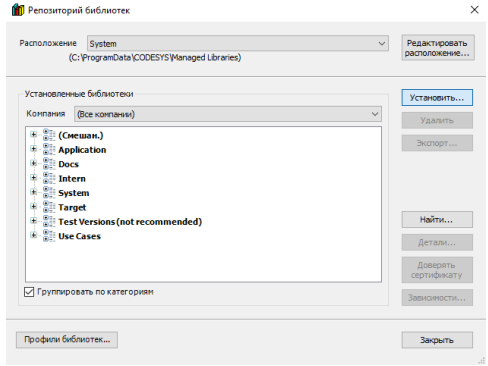


Рис. 2.6. Репозиторий библиотек.

Для установки таргет-файла какого-либо устройства, из верхнего меню выберите «Инструменты – Репозиторий устройств». Откроется окно репозитория устройств (рис. 2.7). Нажмите «Установить», выберите таргет-файл для установки и следуйте инструкциям установщика.

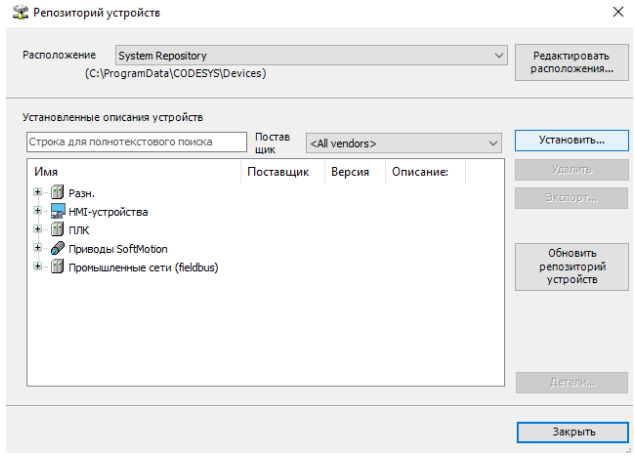


Рис. 2.7. Репозиторий устройств

2.3. Первый запуск CODESYS

2.3.1. Создание проекта

Запустите среду разработки CODESYS. Для создания нового проекта, кликните на «Новый проект…». Укажите расположение проекта и его имя, нажмите «ОК» (см. рис. 2.8).

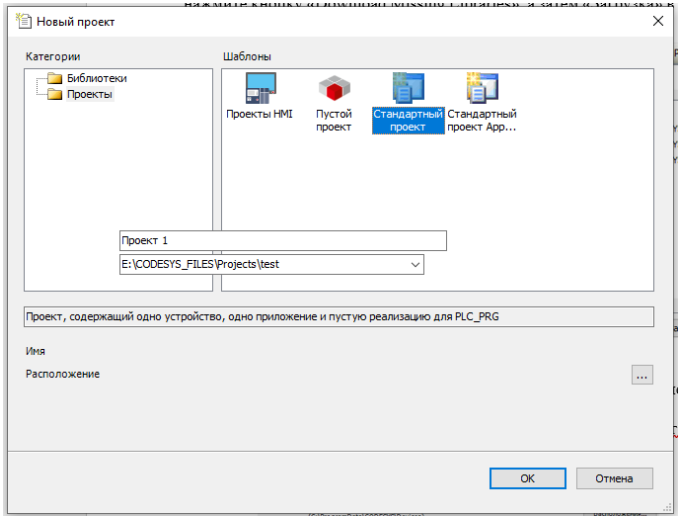


Рис. 2.8. Создание проекта

В следующем окне (рис. 2.9) выберите устройство и язык, на котором будет разрабатываться программа.

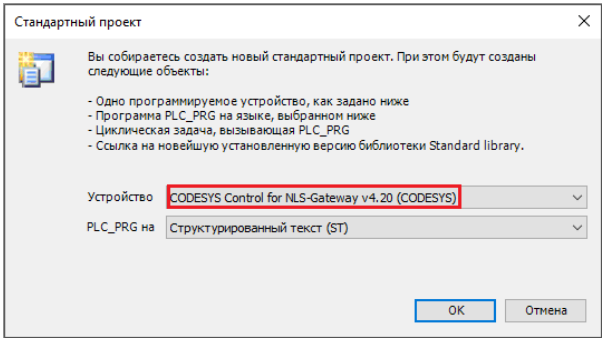


Рис. 2.9. Выбор устройства и языка программирования

2.3.2. Менеджер библиотек

Если в проекте сразу появились ошибки библиотек – перейдите в «Менеджер библиотек» (в дереве устройств кликните по нему дважды) и нажмите «Download Missing Libraries». В появившемся окне нажмите «Загрузка» и дождитесь окончания загрузки библиотек (рис. 2.10).

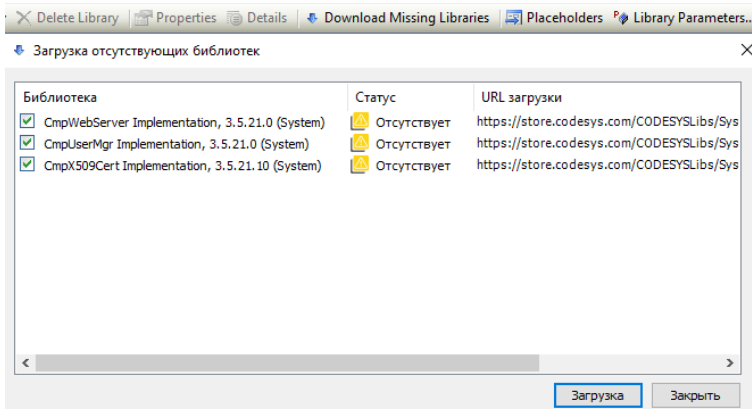


Рис. 2.10. Загрузка недостающих библиотек

2.3.3. Добавление устройств

Чтобы добавить устройство (например, модуль Modbus RTU), щелкните правой кнопкой мыши по названию устройства в дереве устройств – «Добавить устройство». Выберите из перечня необходимое устройство и добавьте его в проект (рис. 2.11).

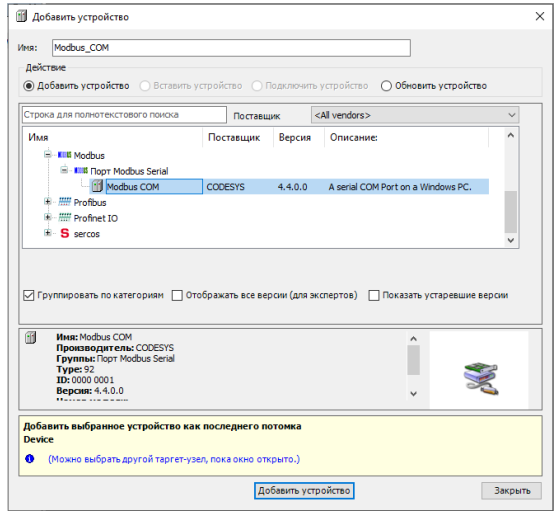


Рис. 2.11. Дважды кликните по устройству, чтобы добавить

Чтобы открыть параметры добавленного устройства, дважды кликните по нему в дереве устройств.

Кроме устройств можно добавлять различные объекты (например, менеджер визуализации, POU и т.д.). Для этого кликните правой кнопкой мыши по «Application» в дереве устройств и выберите «Добавление объекта».

2.3.4. Редактор кода

Код пишется в созданном по умолчанию файле «PLC_PRG». Кликните по нему дважды в дереве устройств, чтобы изменить его (рис. 2.12). Редактор кода состоит из двух частей. В верхней части переменные объявляются, а в нижней – пишется код, который циклически выполняется на шлюзе.

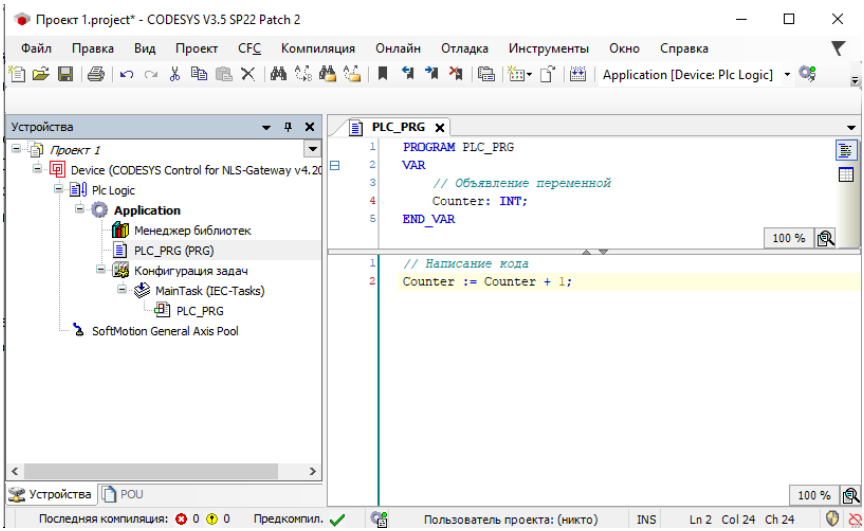


Рис. 2.12. Редактор кода

Чтобы создать новый файл для написания кода, кликните правой кнопкой по «Application – Добавление объекта» в дереве устройств и выберите POU.

2.3.5. Создание визуализации в CODESYS

Запустите среду разработки CODESYS. Создайте стандартный проект или откройте существующий и кликните правой кнопкой мыши по «Application» в дереве устройств. Выберите «Добавление объекта – Визуализация» (рис. 2.13).

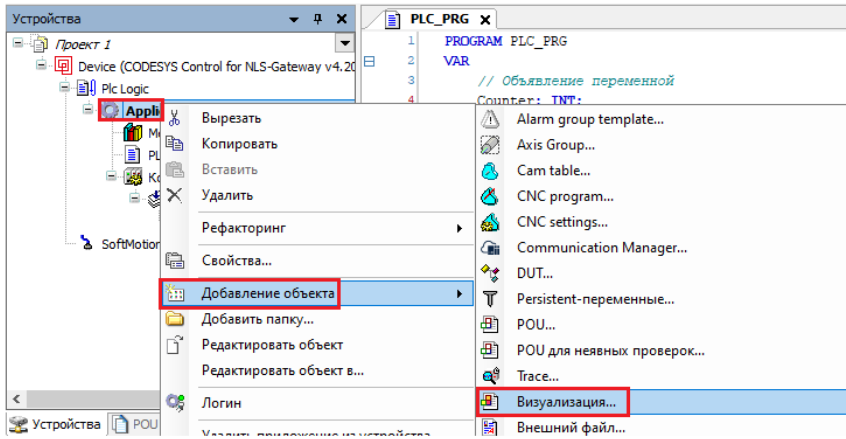


Рис. 2.13. Добавление визуализации в проект

Появится окно с просьбой ввести имя визуализации – по умолчанию в строку имени записывается «Visualization». В дереве устройств появится сама визуализация и менеджер визуализаций, где можно настроить параметры визуализации: стиль визуализации, использование переменных Unicode, параметры шрифта, объем памяти для визуализации и т. д. (рис. 2.14).

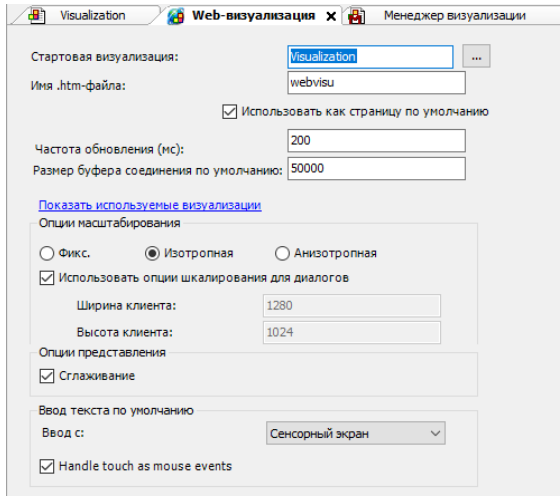


Рис. 2.14. Параметры Web-визуализации

В дереве устройств у элемента «Менеджер визуализации» есть подпункт «Web-визуализация». Там можно выбрать стартовую визуализацию, если их несколько, а также настроить масштабирование визуализации, частоту обновления и ее размер. Для перехода на страницу веб-визуализации введите в адресную строку браузера IP-адрес шлюза и порт 8080. Пример: «192.168.10.125:8000».

Элементы добавляются на экран визуализации перетаскиванием из панели инструментов визуализации, где можно выбрать категорию элементов (рис. 2.15).

Свойства элементов можно настраивать – нажмите левой кнопкой мыши на элемент, который нужно настроить.

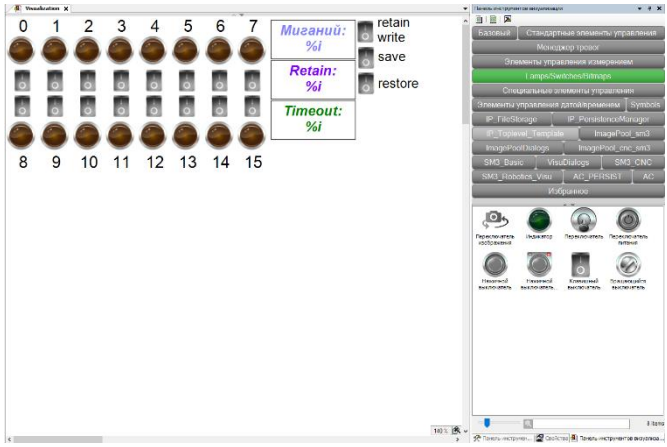


Рис. 2.15. Пример визуализации

Пример: на экране визуализации нажимаю на один из переключателей левой кнопкой мыши – с правой части экрана панель инструментов визуализации заменяется на вкладку свойств выбранного объекта. В зависимости от выбранного элемента настраиваются разные свойства. Для переключателя это размеры, переменная, которая меняет значение при его переключении и цвет (рис. 2.16).

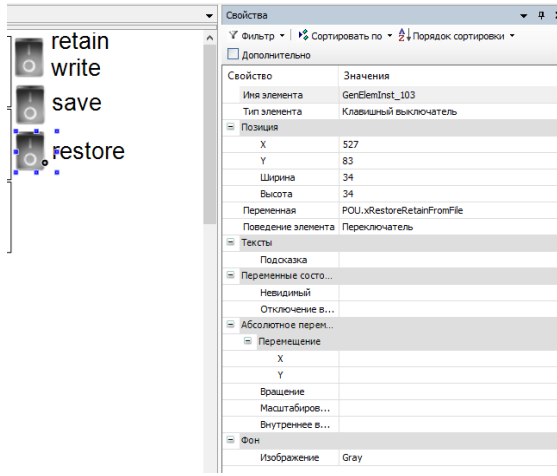


Рис. 2.16. Элемент-переключатель

2.4. Подключение к шлюзу в среде CODESYS

2.4.1. Настройка связи между шлюзом и ПК

По умолчанию сетевые настройки интерфейсов Ethernet шлюзов RealLab – динамические.

Варианты взаимного сетевого расположения контроллера и компьютера:

– Шлюз и компьютер находятся в одной локальной сети.

– Шлюз и компьютер находятся в разных локальных сетях, связанных с помощью соответствующих сетевых устройств (маршрутизаторов).

– Шлюз и компьютер соединены напрямую с помощью кабеля RJ45-RJ45.

В локальных сетях не должно возникать конфликта IP-адресов, т. е. разные устройства не должны обладать совпадающими адресами.

После настройки сетевых параметров контроллера и компьютера следует установить связь между ними в среде CODESYS.

Компонент Device (который определяется соответствующим таргет-файлом) должен соответствовать модели контроллера. В случае необходимости тип устройства можно изменить, выбрав в дереве проекта компонент Device

(всегда находится в верхней части дерева), и, нажав на него ПКМ, открыть окно «Обновить устройство»:

Дважды кликните на шлюз в дереве устройств – откроются «Установки соединения». Нажмите «Сканировать сеть» и дважды кликните по названию устройства, к которому нужно подключиться (см. рис. 2.17).

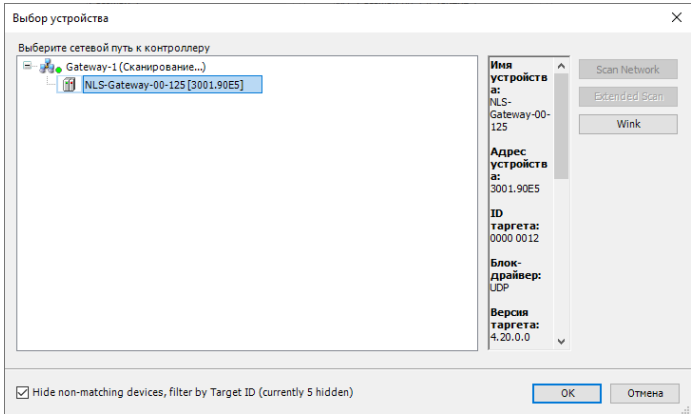


Рис. 2.17. Подключение к шлюзу

В случае успешной установки связи индикаторы шлюза и контроллера загорятся зеленым.

Для загрузки проекта на шлюз нажмите значок:  .

3. Основные функции программирования

3.1. Работа с протоколом MQTT с использованием шифрования

3.1.1. Создание проекта

MQTT (Message Queuing Telemetry Transport) — это легкий сетевой протокол для интернета вещей (IoT). Он работает по принципу «издатель – подписчик» поверх TCP/IP.

Общая информация:

- **Брокер** – основной сервер. Принимает все сообщения от издателей и передает их подписчикам;

- **Издатель** – устройство, которое собирает данные и отправляет их брокеру;
- **Подписчик** – устройство или программное обеспечение, которое получает данные от брокера;
- **Топик** – путь или имя канала связи, по которому подписчик получает сообщения от издателя;
- **Уровни надежности QoS** – 0 (отправка сообщения один раз, без подтверждения – быстро), 1 (гарантированная доставка сообщения без дедупликации – средне), 2 (гарантированная доставка сообщения с дедупликацией – медленно).

Данный пример демонстрирует использование шлюза NLS-Gateway в качестве MQTT-клиента для обмена данными по сети Ethernet. Скачать его можно с сайта RealLab.

Для использования шлюза RealLab в роли клиента MQTT необходимо установить в CODESYS пакет библиотек «IIoT Libraries SL» и добавить в проект библиотеку MQTT Client SL (см. рис. 3.1).

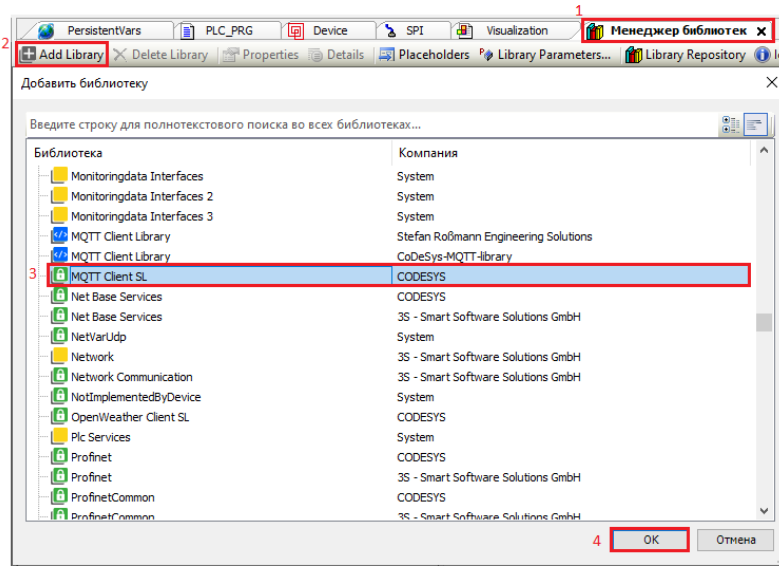


Рис. 3.1. Добавление библиотеки MQTT Client SL

Также необходимо добавить библиотеки Net Base Services, StringUtils, JSON Utilities SL (см. рис. 3.2). Для этого откройте «Менеджер библиотек» и нажмите «Добавить библиотеку» («Add Library»).

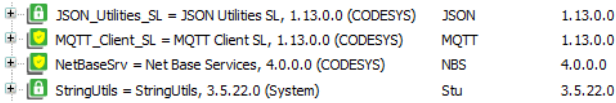


Рис. 3.2. Добавление библиотеки Net Base Services

Затем необходимо создать структуру и два функциональных блока:

- **ST_KeyValuePair** – структура, содержащая два элемента – ключ и значение;
- **FB_JSON2Array** – для разбора JSON-пакета, полученного по MQTT, на пары «ключ-значение». Пары будут храниться в массиве структур ST_KeyValuePair;
- **FB_ExtractSelectedKeys** – для получения значений конкретных ключей, переданных на вход данного ФБ.

Для создания структуры кликните правой кнопкой мыши по «Application» в дереве устройств. Выберите «Добавление объекта – DUT». Введите имя структуры ST_KeyValuePair и нажмите «Добавить» (см. рис. 3.3).

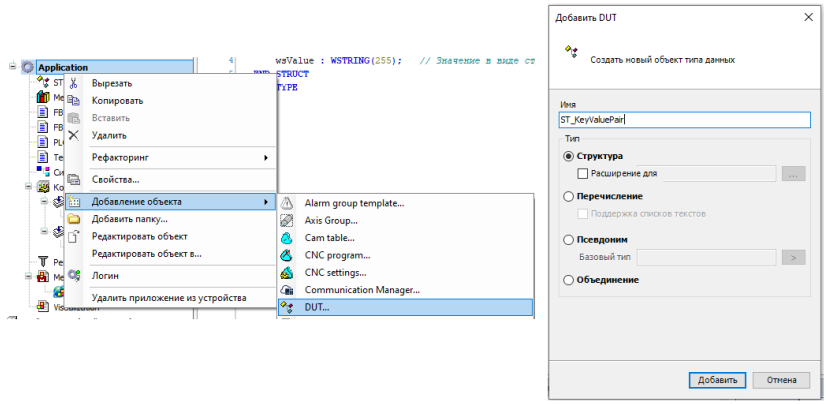


Рис. 3.3. Добавление структуры (DUT)

```
1  TYPE ST_KeyValuePair :
2  STRUCT
3      wsKey : WSTRING(255);
4      wsValue : WSTRING(255);
5  END_STRUCT
6  END_TYPE
```

Рис. 3.4. Код объявления переменных структуры

Также понадобится создать функцию «TerminateString» для усечения строки, содержащей данные в формате JSON, полученные по протоколу MQTT.

Для создания функции кликните правой кнопкой мыши по «Application» и добавьте «POU». Выберите тип «Функция» и задайте ее имя «TerminateString» (см. рис. 3.5).

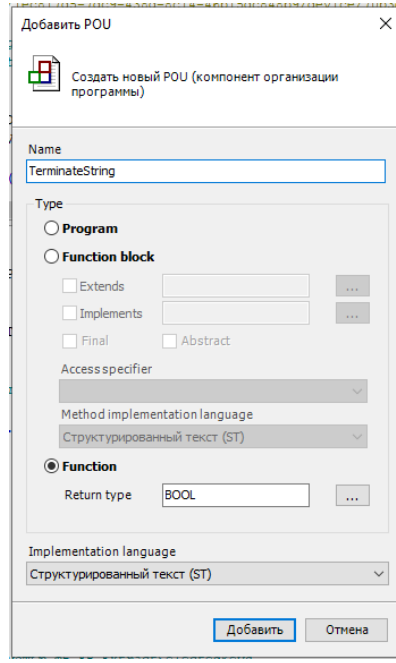


Рис. 3.5. Добавление функции

Код функции отображен на рис. 3.6.

```
// Terminates the STRING psIn at position udiLength
FUNCTION TerminateString : BOOL
VAR_INPUT
    psIn : POINTER TO BYTE; // Pointer to STRING
    udiLength : UDINT; // Length of psIn
END_VAR
VAR
END_VAR

psIn[udiLength] := 0;
TerminateString := TRUE;
```

Рис. 3.6. Код функции TerminateString

Для создания функционального блока FB_JSON2Array аналогично вызовите контекстное меню кликом по «Application» и добавьте POU. Введите наименование ФБ, выберите тип «Function block», язык - ST (см. рис. 3.7).

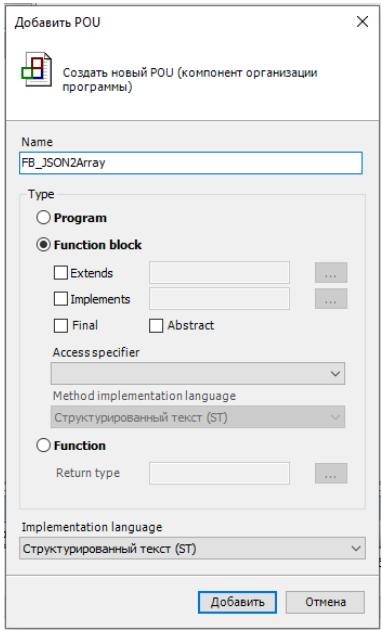


Рис. 3.7. Добавление функционального блока

Аналогично добавьте ФБ «FB_ExtractSelectedKeys»

В основной программе PLC_PRG необходимо объявить ФБ для получения данных по MQTT и написать код для их вызова.

Необходимые функциональные блоки:

- **MQTT.MQTTClient** – клиент MQTT. С помощью него будет установлено подключение к брокеру;
- **MQTT.MQTTPublish** – ФБ для публикации данных в топик MQTT-брокера;
- **MQTT.MQTTSubscribe** – ФБ для подписки на топик MQTT-брокера;
- **FB_JSON2Array** – ФБ для парсинга Unicode-строки, содержащей данные, структурированные в JSON;
- **FB_ExtractSelectedKeys** – ФБ для получения значений по конкретным ключам из JSON.

Код функциональных блоков и программы PLC_PRG можно посмотреть в проекте-примере передачи данных по протоколу MQTT для CODESYS на сайте RealLab.

Также была разработана визуализация для данного проекта-примера.

3.1.2. Подключение к брокеру MQTT без шифрования

Для подключения к брокеру MQTT введите IP-адрес брокера, порт. Также введите логин и пароль при необходимости. В параметрах подписки на топик.

После нажатия на кнопку «Подключить» должен загореться индикатор, соответствующий успешному подключению (см. рис. 3.8). Для отключения повторно нажмите на «Подключить». Подключение к брокеру будет разорвано и индикатор погаснет.

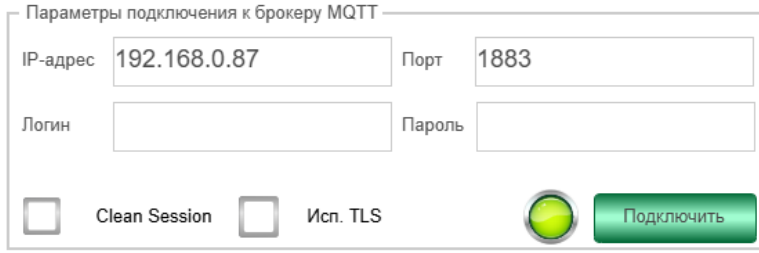


Рис. 3.8. Параметры подключения к брокеру MQTT

Для того, чтобы подписаться на топик, введите его в строку «Подписка на топик», а затем нажмите на кнопку справа от метки «Подписаться». Если подписка будет выполнена успешно, то загорится зеленый индикатор справа от кнопки подписки. В противном случае загорится индикатор «Ошибка» (см. рис. 3.9).

Когда клиент MQTT получит новые данные из топика, на который подписан, загорится индикатор «Обработано» (см. рис. 3.9) и в таблице (см. рис. 3.10) появятся пары ключ-значение, полученные после обработки данных, полученных в формате JSON.

Повторное нажатие на кнопку подписки отключает ее. Для подписки на другой топик отпишитесь от текущего топика, измените путь к топику, на который необходимо подписаться и нажмите кнопку подписки.

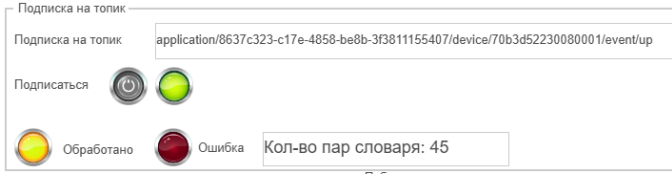


Рис. 3.9. Параметры подписки на топик

Разбор строки, содержащей данные в JSON-формате, на пары «ключ–значение» выполняет ФБ FB_JSON2Array. Результат обработки возвращается в виде массива jpArray в PLC_PRG. jpArray – массив структур «ST_KeyPairValue». Таблица на рис. 3.10 отображает ключи и значения, находящиеся в jpArray.

	Параметр	Значение
0	deduplicationId	f3a22b5e-b95a-4ec0-9a47-6ce43e74ef1c
1	time	2026-07-24T13:39:28.927805980+00:00
2	deviceInfo	Object
3	tenantId	7cb3a20d-e85c-418d-970e-10c17b1cca10
4	tenantName	ChirpStack
5	applicationId	8637c323-c17e-4858-be8b-3f3811155407
6	applicationName	tms_01
7	deviceProfileId	64af61d7-3e42-43b8-9fee-96f8e58c47f2
8	deviceProfileName	TR-xxx_01
9	deviceName	TR120091-L_0001
10	devEui	70b3d52230080001
11	deviceClassEnabled	CLASS_A
12	tags	Object
13	devAddr	008a36b6
14	adr	true
15	dr	0

Рис. 3.10. Таблица пар «ключ-значение», полученных из топика, на который подписан клиент

Также есть возможность получения отдельных пар «ключ-значение» с помощью ФБ FB_ExtractSelectedKeys. ФБ возвращает также массив структур ST_KeyValuePair. В PLC_PRG: fbExtractKeys – объявление ФБ, aKeysToSearch – массив ключей для поиска их значений, aResults – возвращенный массив значений (см. рис. 3.11). В примере на рис. 3.11 выборка из данных, полученных от термоштанги RealLab по MQTT.

Искомый параметр	Результат
humidity	58.6
temperatures	[23.3, 23.6, 23.8, 23.5, 23.5, 23.5, 23.3, 23.5, 23.2]
battery	3.993
deviceName	TR120091-L_0001

Рис. 3.11. Выборка значений по массиву ключей, переданных в ФБ FB_ExtractSelectedKeys

Для публикации сообщения в топик введите топик, в который будет публиковаться сообщение. Затем введите само сообщение и нажмите кнопку «Опубликовать». При зажатии кнопки «Опубликовать» загорится один из двух индикаторов справа от кнопки (см. рис. 3.12):

– зеленый – «успешная публикация»

– красный – «ошибка».

Рис. 3.12. Параметры публикации в топик

3.1.3. Подключение к брокеру MQTT с шифрованием TLS

Для подключения к брокеру MQTT введите IP-адрес брокера, порт. Также введите логин и пароль при необходимости. Если подключение к брокеру защищено шифрованием (TLS) установите галочку возле надписи «Исп. TLS». И нажмите кнопку «Подключить». Подключение должно установиться.

В данном случае происходит подключение без валидации сертификата.

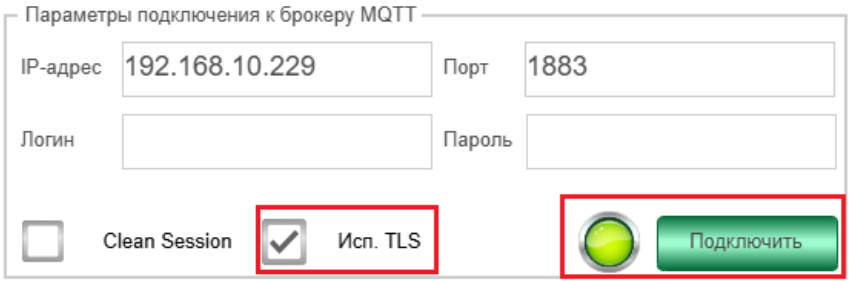


Рис. 3.13. Подключение к MQTT-брокеру с использованием TLS

Для подключения к брокеру с валидацией сертификата брокера используйте другой пример проекта (ОбщийПримерJSON_useTLS.projectarchive). Код другого примера проекта отличается в PLC_PRG. Более детально настроены параметры TLS (в проекте …_noTLS они закомментированы).

Код параметров TLS (см. рис. 3.14).

```
// ------------------------------------------------------------------------
// НАСТРОЙКА TLS / SSL ДЛЯ ОДНОСТОРОННЕГО TLS
// ------------------------------------------------------------------------
xUseTLSItf : BOOL := TRUE;

sCipherList : NBS.CIPHER_LIST := STRUCT(psList := ADR('HIGH'), udiSize := 4);

tlsContext : NBS.TLSContext := (
    sUseCaseName := '',
    ePurpose := NBS.PURPOSE.CLIENT_SIDE,
    sTLSVersion := '1.2',
    sCipherList := sCipherList,
    sHostname := sHostname,
    udiVerificationMode := 2
);
// Режим верификации сертификата (udiVerificationMode)
// 0 = Без проверки наличия сертификата (только шифрование без использования сертификата)
// 1 = Использование тестового / самоподписанного сертификата (требуется добавление сертификата в меню безопасности CODESYS)
// 2 = Проверка сертификата брокера через Trusted Certificates (требуется добавление сертификата в меню безопасности CODESYS)

itfTlsContext : NBS.ITLSContext := tlsContext;
itfNullTlsContext : NBS.ITLSContext := 0;
```

Рис. 3.14. Код настройки TLS

Табл. 2. Настройка уровня проверки сертификата

Значение	Описание
0	TLS, без проверки сертификата
1	TLS, проверка наличия самоподписанного сертификата
2	TLS, проверка доверенного сертификата

Внимание! В случае с уровнем проверки сертификата 1 или 2 требуется добавление сертификата в CODESYS.

Перейдите в веб-конфигуратор RealLab (ip-адрес-шлюза:8000). Откройте страницу «Сеть», вкладку «MQTT». В разделе «Шифрование трафика (TLS) установите параметр «Включить защищенное соединение в режим включено» и при отсутствии сгенерированных сертификатов нажмите кнопку «Перевыпуск сертификатов». Примените изменения кнопкой «применить настройки и перезапустить брокер» (см. рис. 3.15).

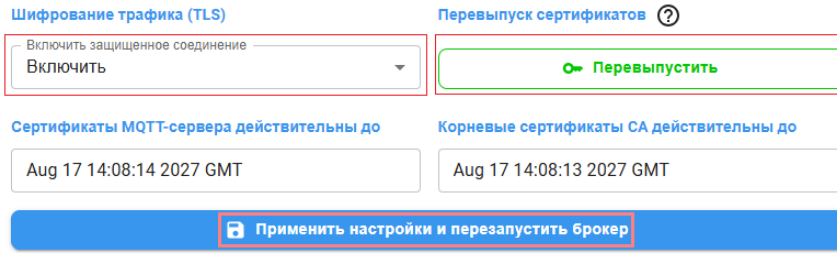


Рис. 3.15. Выпуск сертификатов на шлюзе

С помощью SFTP-клиента (через Bitvise SSH, например) скопируйте сертификат со шлюза (/etc/mosquitto/ca_certificates/ca.crt) на ПК с CODESYS.

Далее в верхнем меню CODESYS нажмите «Вид» - «Безопасность».

Внимание! Если этот пункт не отображается, установите «CODESYS Security Agent» и перезапустите CODESYS.

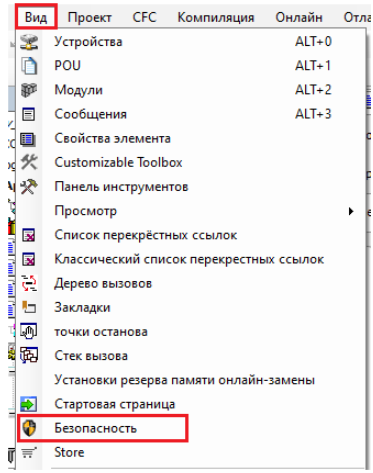


Рис. 3.16. Переход на страницу «Безопасность»

Импортируйте сертификат «ca.crt» со шлюза, добавьте его в Trusted Certificates в CODESYS (меню безопасности), выполните подключение к шлюзу (см. рис. 3.17 – рис. 3.18).

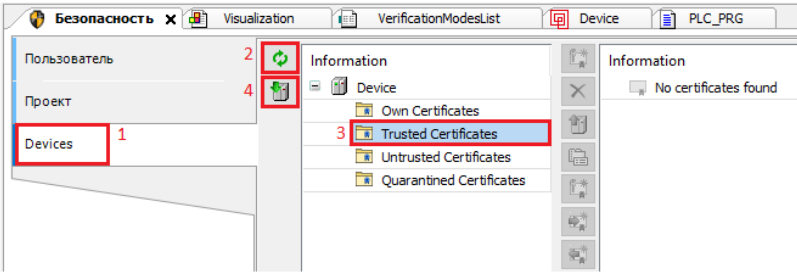


Рис. 3.17. Добавление сертификата ч. 1

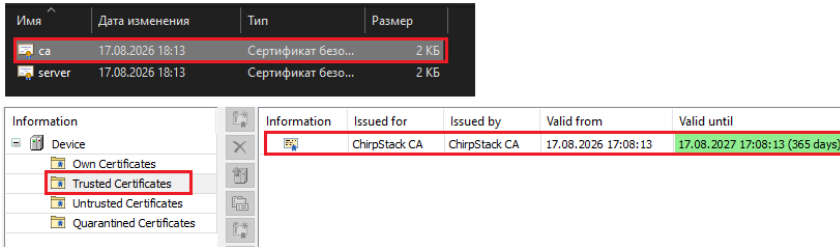


Рис. 3.18. Добавление сертификата ч. 2

Далее установите уровень проверки доверенного сертификата в проекте CODESYS (udiVerificationMode := 2 в параметрах tlsContext (см. рис. 3.14)).

Загрузите проект на шлюз, откройте визуализацию, введите IP-адрес, порт, логин и пароль при необходимости. Установите галочку «Исп. TLS» и нажмите подключиться.

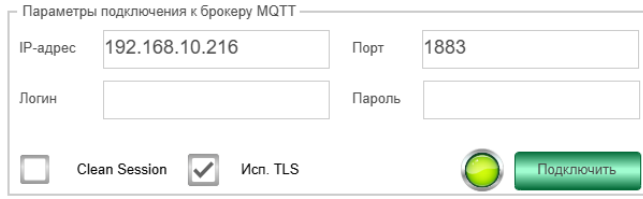


Рис. 3.19. Результат подключения

3.2. Настройка Modbus RTU

3.2.1. Шлюз NLS-Gateway в роли Modbus RTU Master

Откройте существующий проект или создайте новый. В дереве устройств вызовите контекстное меню кликом по «Device (CODESYS Control for NLS-Gateway) и добавьте устройство «Modbus COM». Укажите номер COM-порта и установите необходимые Вам параметры.

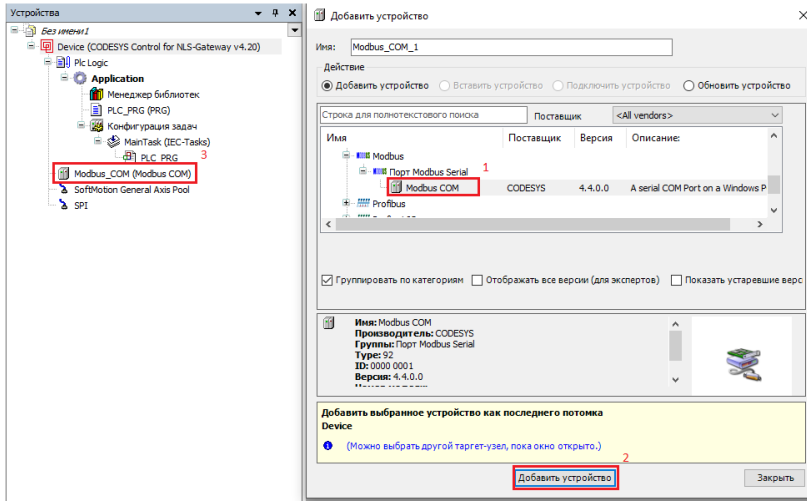


Рис. 3.20. Добавление Modbus COM

Во вкладке «Общее» Modbus COM необходимо указать номер COM-порта, используемого шлюзом, скорость передачи (по умолчанию 9600 бод), а также четность – NONE. Все остальные настройки без изменений (см. рис. 3.21).

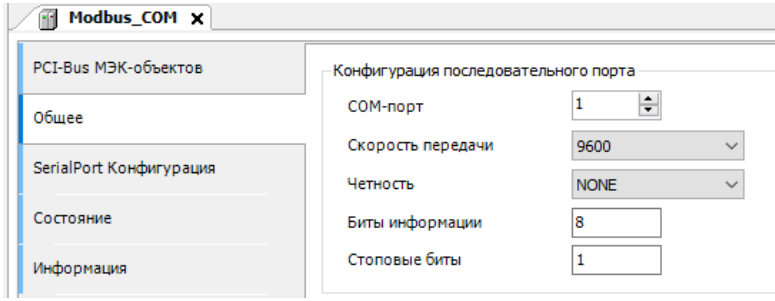


Рис. 3.21. Пример конфигурации Modbus COM

После Modbus COM следует добавить Modbus Client COM Port. Во вкладке «Общее» Modbus Client COM Port – установить галочку «Автоперезапуск соединения».

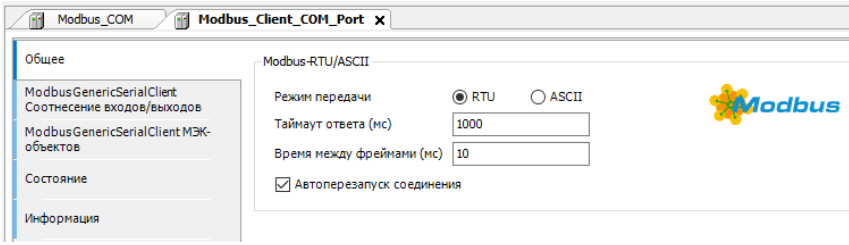


Рис. 3.22. Пример конфигурации Modbus Client COM Port

После Modbus Client COM Port следует добавить Modbus Server COM Port (см. рис. 3.23) либо добавить готовый шаблон модуля ввода-вывода. Во вкладке Общее Modbus Slave COM Port установить адрес Slave-устройства. Также можно указать индивидуальный Таймаут-ответа и ID устройства.

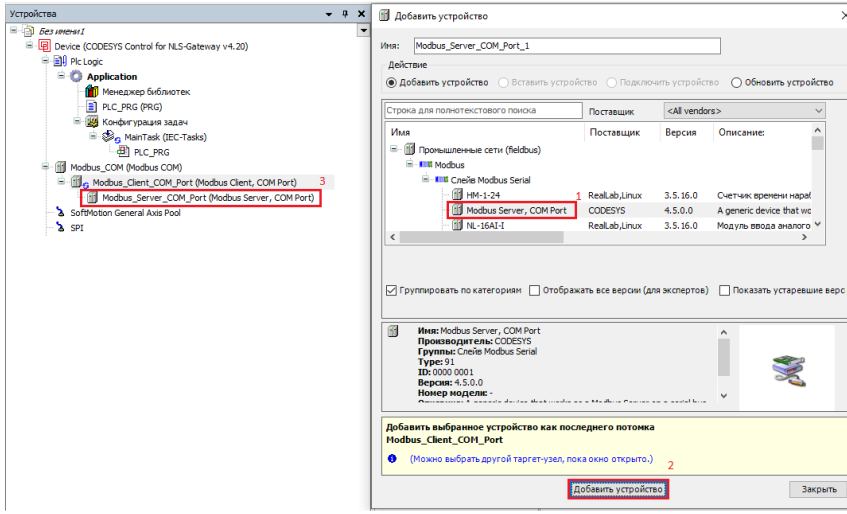


Рис. 3.23. Добавление Modbus Server COM Port

Во вкладке «Каналы Modbus Slave» Modbus Server COM Port создайте канал(ы) для опроса модуля ввода-вывода (см. рис. 3.24). Затем откройте ModbusGenericSerialClient «Соотнесение входов/выходов» и для необходимых каналов задайте с помощью Ассистента ввода переменные, которые должны использоваться в коде прикладной программы (см. рис. 3.25), а также установите параметр «Вкл. 1 (в задаче цикла шины, если не исп.)».

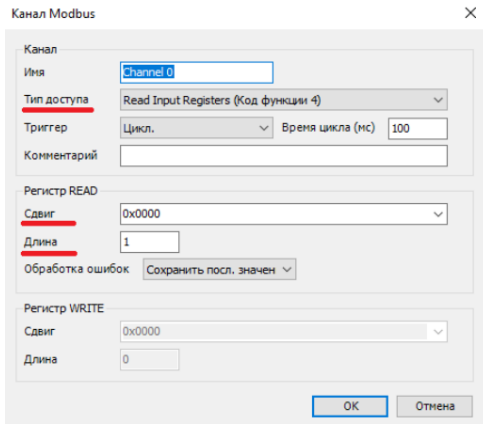


Рис. 3.24. Создание канала Modbus Server

```
1  PROGRAM PLC_PRG
2  VAR
3      wTestVar: WORD;
4  END_VAR
```

Рис. 3.25. Объявленная переменная в программе PLC_PRG

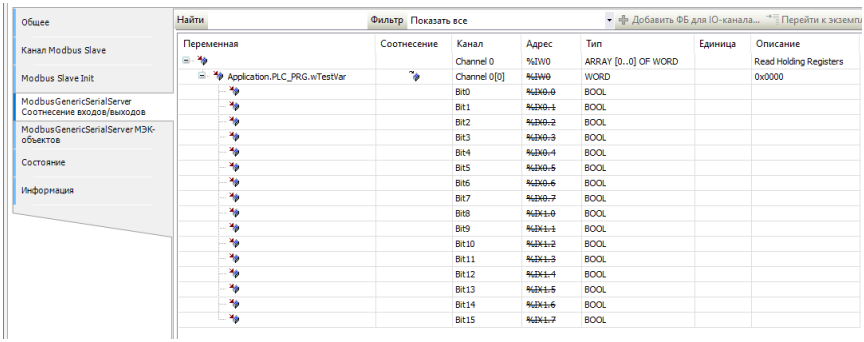


Рис. 3.26. Соотнесение входов/выходов.

В результате запуска шлюза в режиме Modbus RTU Master созданные компоненты в дереве устройств будут отображаться зеленой пиктограммой.

3.2.2. Шлюз NLS-Gateway в роли Modbus RTU Slave

Откройте существующий проект или создайте новый. В дереве устройств вызовите контекстное меню кликом по «Device (CODESYS Control for NLS-Gateway) и добавьте устройство «Modbus COM». Укажите номер COM-порта и установите необходимые Вам параметры.

После Modbus COM добавьте Modbus Serial Device. Во вкладке Modbus Serial Device (рис. 3.72) установить ID, который будет назначен данному COM-порту шлюза, а также количество Регистров хранения (Holding registers 2–500) и Входных регистров. Регистры хранения (Holding registers) – Тип доступа: чтение/запись. Входные регистры (Inputs registers) – Тип доступа: только чтение.

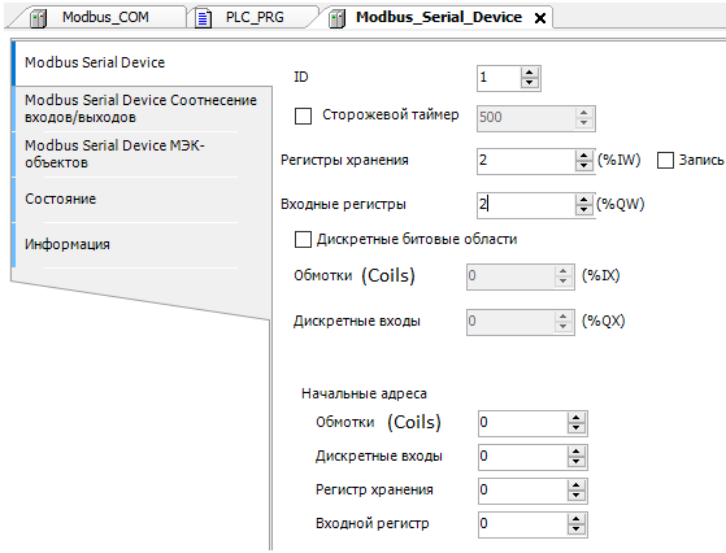


Рис. 3.27. Конфигурация Modbus Serial Device

В настройках на вкладке Modbus Serial Device «Соотнесение входов /выходов» для необходимых каналов задать с помощью Ассистента ввода переменные, которые должны использоваться в коде прикладной программы (см. рис. 3.73), а также установить параметр «Вкл. 1 (в задаче цикла шины, если не исп.)».

Переменная	Соотнесение	Канал
Application.PLC_PRG.wTestWrite		Регистры временного хранения
		Регистры временного хранения[0]
		Регистры временного хранения[1]
Application.PLC_PRG.wTestRead		Входные регистры
		Входные регистры[0]
		Входные регистры[1]

Рис. 3.28. Соотнесение входов/выходов

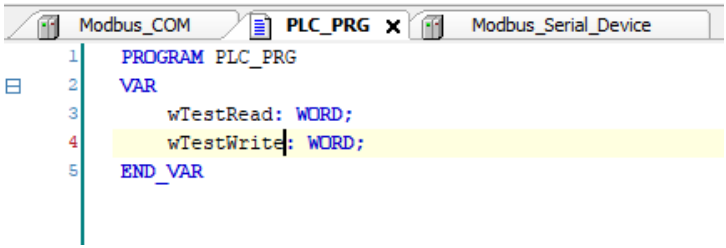


Рис. 3.29. Объявленные переменные в программе PLC_PRG

В результате запуска шлюза в режиме Modbus RTU Slave созданные компоненты в дереве устройств будут отображаться зеленой пиктограммой.

3.2.3. Работа с модулями ввода-вывода RealLab по протоколу Modbus RTU

Проекты с примерами работы с устройствами по протоколу Modbus RTU в среде разработки CODESYS 3.5 можно скачать на нашем сайте.

Подробная информация по работе с модулями ввода-вывода RealLab по протоколу Modbus RTU находится в руководстве по ссылке, п. 4.1.

3.3. Настройка Modbus TCP

3.3.1. Шлюз NLS-Gateway в роли Modbus TCP Master

Откройте существующий проект или создайте новый. В дереве устройств вызовите контекстное меню кликом по «Device (CODESYS Control for NLS-Gateway) и добавьте устройство «Ethernet» (см. рис. 3.30).

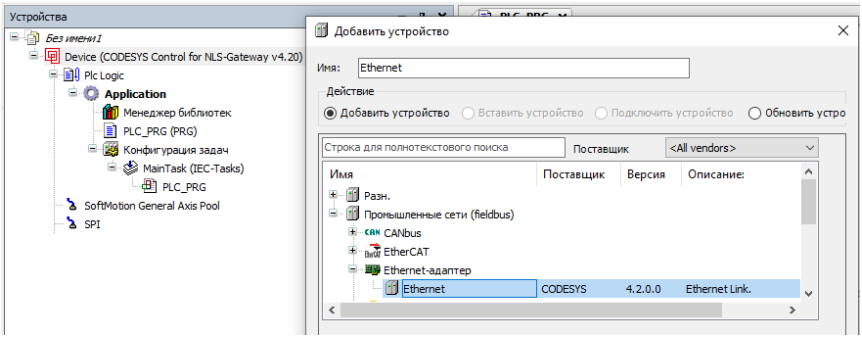


Рис. 3.30. Добавление устройства «Ethernet в проект»

Далее дважды кликните по «Device» в дереве устройств и установите соединение со шлюзом (кнопка «Сканировать сеть» – кликните дважды по строке со шлюзом, к которому необходимо подключиться)

Перейдите в конфигурацию Ethernet и выберите основной интерфейс сети (eth0 – проводное подключение, wlan0 – Wi-Fi) (см. рис. 3.31).

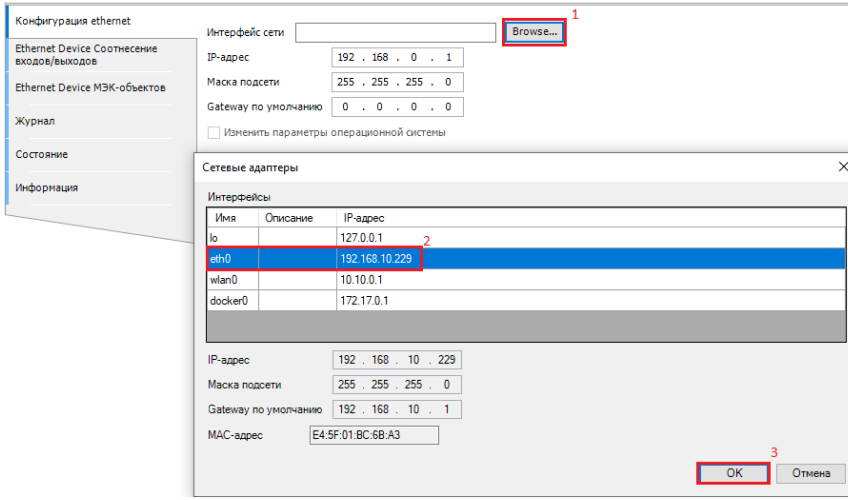


Рис. 3.31. Выбор сетевого интерфейса

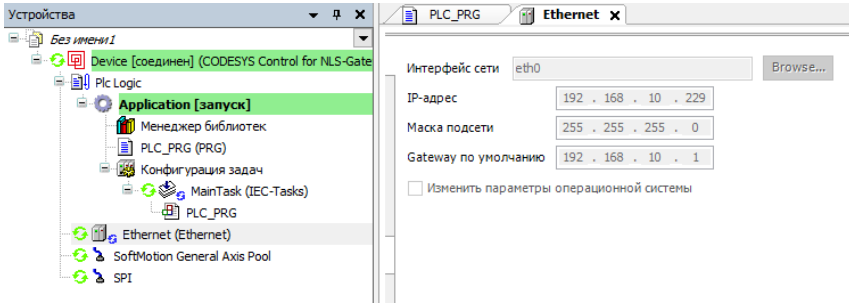


Рис. 3.32. Настройки Ethernet

После Ethernet следует добавить Modbus TCP Client. Во вкладке «Общее» Modbus TCP Master установить галочку «Автоподключение».

После Modbus TCP Master добавьте Modbus TCP Server. Во вкладке «Общее» Modbus TCP Slave установите адрес TCP Slave-устройства.

Во вкладке Канал Modbus Slave необходимо установить параметры опрашиваемого Slave-устройства (Тип доступа, Сдвиг регистра, Длина регистра).

В настройках каждого Modbus TCP Slave на вкладке «ModbusGenericSerialClient Соотнесение входов/выходов» (см. рис. 3.34) для необходимых каналов задать с помощью Ассистента ввода переменные, которые должны использоваться в коде прикладной программы (см. рис. 3.33), а также установить параметр «Вкл. 1 (в задаче цикла шины, если не исп.)».

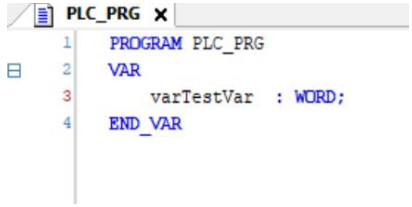


Рис. 3.33. Созданная переменной

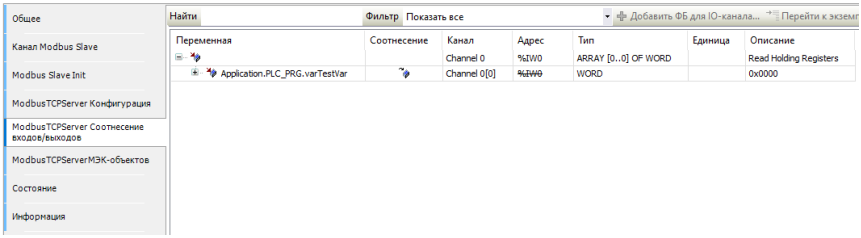


Рис. 3.34. Соотнесение входов/выходов

В результате запуска ПЛК в режиме Modbus TCP Master созданные компоненты в дереве устройств будут отображаться зеленой пиктограммой.

3.3.2. Шлюз NLS-Gateway в роли Modbus TCP Slave

Откройте существующий проект или создайте новый. В дереве устройств вызовите контекстное меню кликом по «Device (CODESYS Control for NLS-Gateway) и добавьте устройство «Ethernet» (см. рис. 3.35).

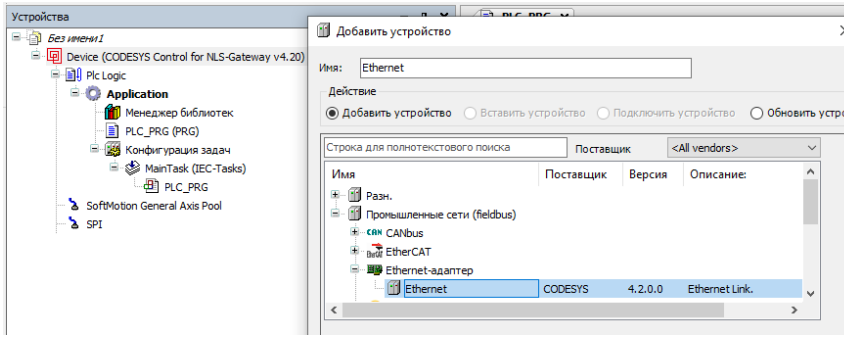


Рис. 3.35. Добавление устройства «Ethernet в проект»

Далее дважды кликните по «Device» в дереве устройств и установите соединение со шлюзом (кнопка «Сканировать сеть» – кликните дважды по строке со шлюзом, к которому необходимо подключиться)

Перейдите в конфигурацию Ethernet и выберите основной интерфейс сети (eth0 – проводное подключение, wlan0 – Wi-Fi) (см. рис. 3.36).

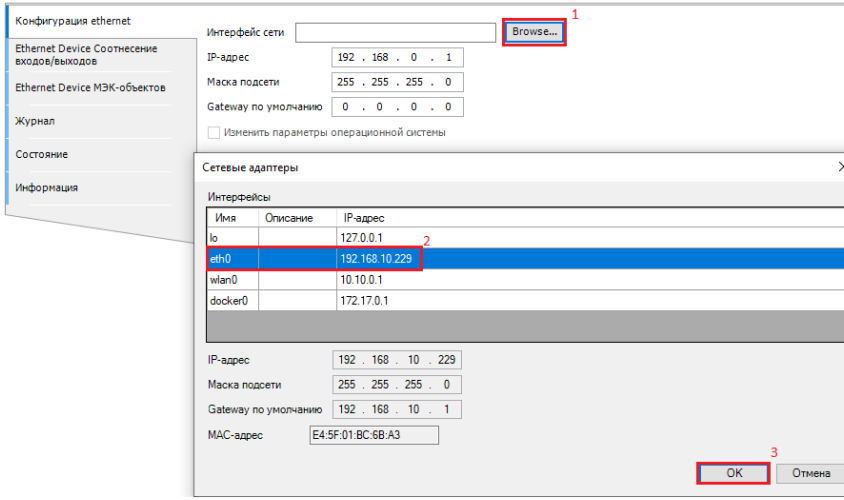


Рис. 3.36. Выбор сетевого интерфейса

После Ethernet следует добавить Modbus TCP Server Device. Во вкладке Modbus TCP Slave Device (см. рис. 3.37) установите количество регистров хранения (Holding registers 2-500) и Входных регистров (Inputs registers 2-

500). Регистры хранения (Holding registers) – Тип доступа: чтение/запись. Входные регистры (Inputs registers) – Тип доступа: только чтение.

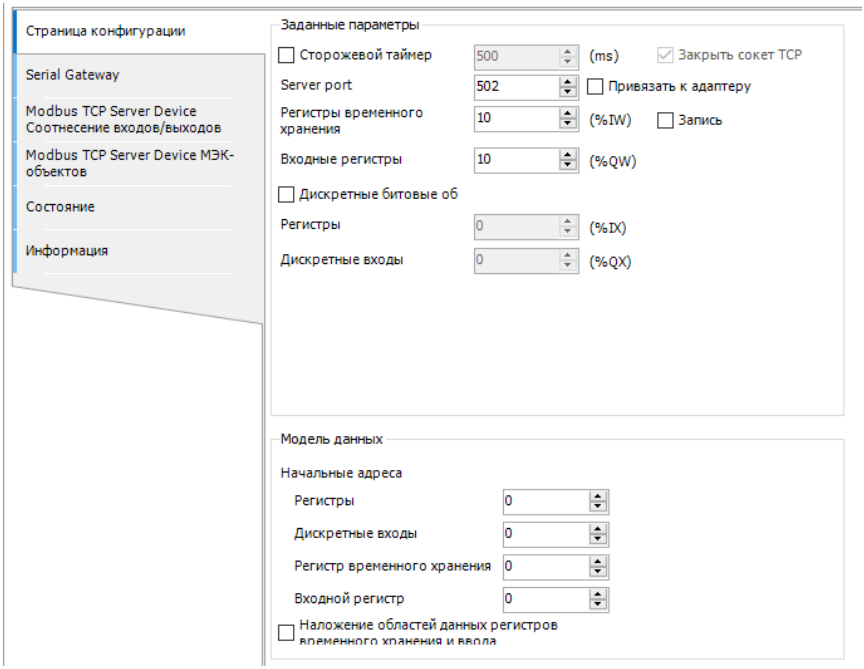


Рис. 3.37. Настройка регистров TCP Slave

В настройках на вкладке «Modbus TCP Server Device – Соотнесение входов/выходов» для необходимых каналов задать с помощью Ассистента ввода переменные, которые должны использоваться в коде прикладной программы (см. рис. 3.38), а также установить параметр «Вкл. 1 (в задаче цикла шины, если не исп.)».

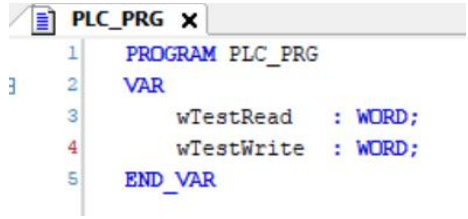


Рис. 3.38. Создание переменных

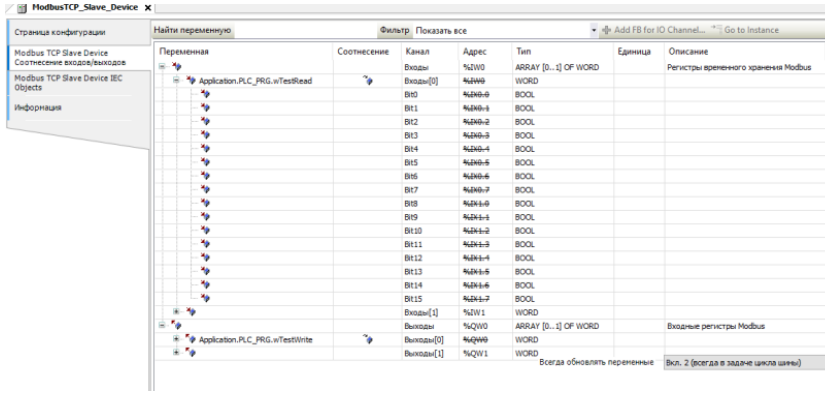


Рис. 3.39. Соотнесение переменных

В результате запуска шлюза в режиме Modbus TCP Server созданные компоненты в дереве устройств будут отображаться зеленой пиктограммой.

3.3.3. Работа с модулями ввода-вывода RealLab по протоколу Modbus TCP

Проекты с примерами работы с устройствами по протоколу Modbus TCP в среде разработки CODESYS 3.5 можно скачать на нашем сайте.

Подробная информация по работе с модулями ввода-вывода RealLab по протоколу Modbus TCP находится в руководстве по ссылке, п. 4.2.

3.4. Настройка CANopen

3.4.1. Подключение устройства CAN

Создайте новый проект. Подключитесь к шлюзу. В дереве устройств дважды кликните по «Device». В основном окне нажмите кнопку «Сканировать сеть» и выберите шлюз к которому хотите подключиться.

ПКМ (Правой кнопкой мыши) по "Device" в дереве устройств – Добавить устройство – CANbus - CANbus.

ПКМ по CANbus – Добавить устройство – CANopen – CANopenManager – CANopen_Manager.

ПКМ по CANopen_Manager – Добавить устройство – [выберите ваше устройство из списка].

Дважды кликните по CANbus в дереве устройств. Во вкладке CANbus «Общее» укажите номер CAN-порта и скорость передачи (Кбит/с) (см. рис. 3.40).

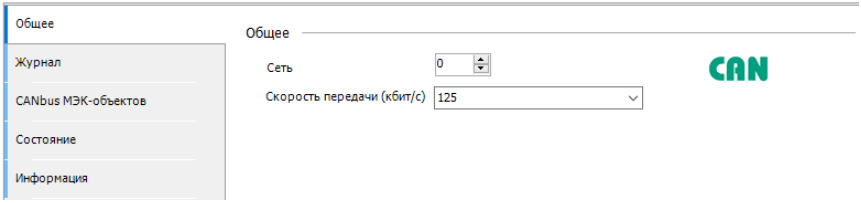


Рис. 3.40. Параметры CANbus

Двойной клик по CANopen_Manager – установите параметры как на рис. 3.41.

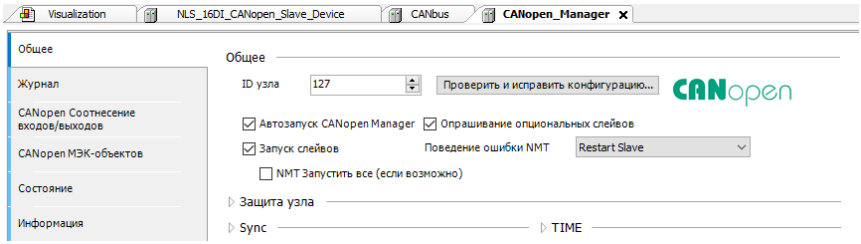


Рис. 3.41. Пример параметров CANopen_Manager для подключения

После CANopen Manager добавьте необходимое slave-устройство. Например, NLS-16DO-CAN.

В настройках slave-устройства (см. рис. 3.42) на вкладке «Общее» укажите ID узла. Поставьте галочку на «Экспертные установки», чтобы получить доступ к дополнительным параметрам. Снимите галочку «Сброс узла». Если обмен данными не происходит, установите галочку «Опц. устройство».

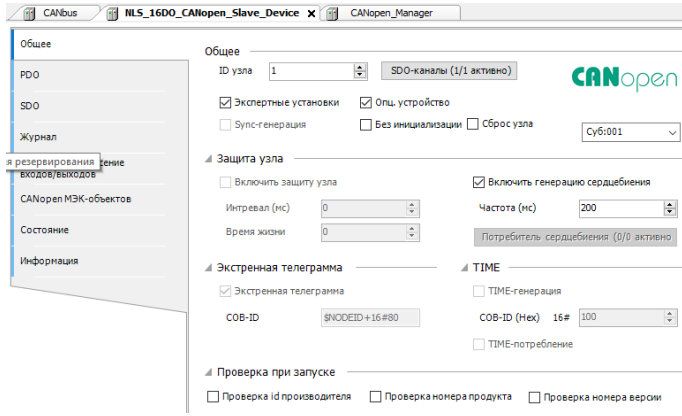


Рис. 3.42. Настройки устройства CAN

Если Вам необходимо, чтобы после запуска проекта, все подключенные CAN устройства запускались сразу, а не находились в состоянии конфигурации, установите галочки «Запуск слейвов» и «NMT Запустить все (если возможно)».

3.4.2. Настройка параметров PDO / SDO

Дважды кликните по имени вашего модуля в дереве устройств. Вам понадобятся вкладки: PDO, SDO, CANopen Соотнесение входов/выходов. PDO (объект данных процесса) и SDO (объект данных службы): CANopen определяет два основных метода связи: PDO и SDO.

- **PDO** используются для обмена данными между устройствами в режиме реального времени, обеспечивая быструю и эффективную передачу важной информации.
- **SDO**, с другой стороны, используются для настройки и доступа к параметрам и настройкам устройства, обеспечивая структурированный метод для удаленной настройки устройства.

3.4.2.1. PDO

Протокол PDO это протокол обмена данными между узлами сети. Длина данных 8 байт. Протокол PDO определяет формат данных в зависимости от настроек PDO. Протокол PDO соответствует CANopen application layer and communication profile CiA 301.

Важно! Кадры протокола PDO обрабатываются только в рабочем состоянии.

Протокол PDO включает в себя два вида кадров данных TPDOn (передаваемые) и RPDO (принимаемые). Каждый вид кадров включает в себя по четыре независимых потока. Каждый поток имеет параметры коммуникации и параметры сопоставления.

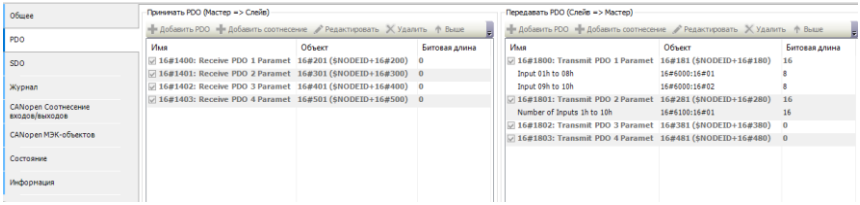


Рис. 3.43. Пример PDO модуля NLS-16DI-CAN

Чтобы добавить соотнесение в PDO, нажмите « + Добавить соотнесение» в верхней части интерфейса программы. Один объект RPDO/TPDO может иметь максимальную битовую длину, размером 64 бита.

Важно! В случае, когда используется асинхронный тип передачи TPDO, требуется установить параметр «Время события» не равным нулю в задействованных потоках TPDO, иначе обмен данными происходить не будет!.

Чтобы установить время события для потока TPDO, дважды кликните по его имени (например: «Transmit PDO n Parameter», где n – номер потока). Убедитесь, что тип передачи асинхронный и установите время события.

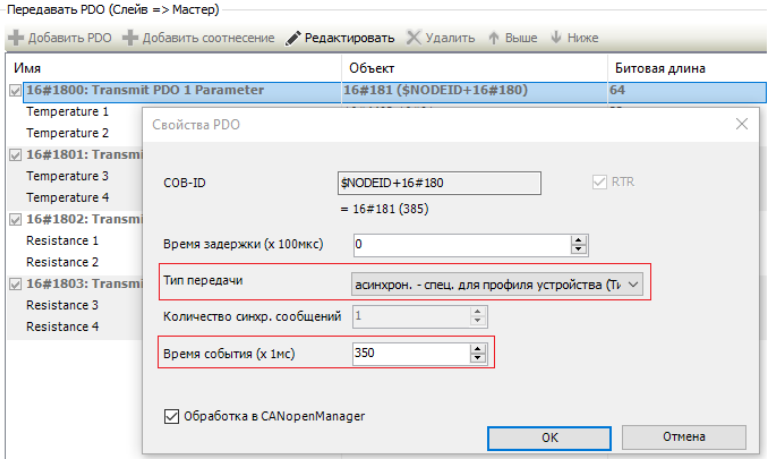


Рис. 3.44. Настройка времени события TPDO

3.4.2.2. SDO

Протокол SDO определяет команды доступа к объектам SDO. Протокол SDO соответствует CANopen application layer and communication profile CiA 301.

Важно! Кадры протокола SDO не обрабатываются, когда устройство находится в состоянии остановки.

Важно! Кадры протокола SDO с командой записи обрабатываются, только когда устройство находится в состоянии конфигурирования.

Обмен данными протокола SDO производится в режиме клиент – сервер. Устройство является сервером SDO, а ведущее устройство сети - клиентом SDO. Инициатором обмена выступает клиент SDO и на каждый запрос сервер генерирует ответ (если устройство не может предоставить какие-либо значимые данные, или если запрос сам по себе был ошибочным, то сервер SDO передаст информацию об ошибке).

Важно! Снимите флажок с пункта «Записать полную конфигурацию PDO», при установке флажка на пункт «Включить защиту узла» во вкладке «Общее» – «Защита узла». Иначе есть риск, что модуль будет перезапускаться сам по себе и перестанет отвечать на запросы ведущего устройства.

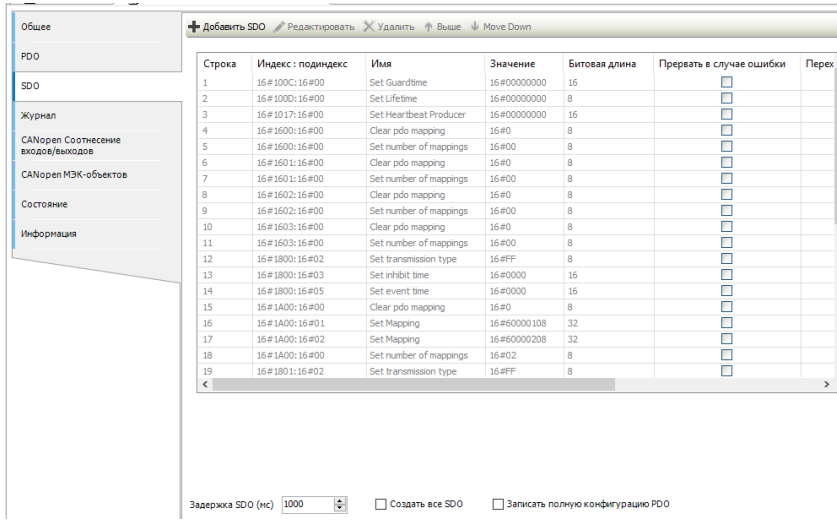


Рис. 3.45. Вкладка SDO

Внимание! Если в вашей программе (в визуализации) уже реализовано управление состояниями модуля и редактирование параметров SDO, снимите флажок с пункта «Создать все SDO» т.к. данная вкладка имеет больший приоритет выполнения, чем работа с SDO с помощью кода самой программы, и каждый раз когда устройство будет перезапускаться, параметры SDO будут взяты с таблицы данной вкладки, что очень неудобно в определенных случаях. Примеры управления модулем, его состоянием и параметрами SDO с помощью кода, включены в примеры использования модулей серии CAN: NLS-8AI, NLS-8TI, NLS-8AI-I, NLS-4RTD, NLS-4AO.

3.4.3. Работа с модулями ввода-вывода RealLab по протоколу CANopen

Проекты с примерами работы с устройствами по протоколу CANopen в среде разработки CODESYS 3.5 можно скачать на нашем сайте.

Подробная информация по работе с модулями ввода-вывода RealLab по протоколу CANopen находится в руководстве по ссылке, п. 4.3.

3.5. Работа с OPCUA-сервером

CODESYS OPC Server V3 – самый простой с точки зрения настройки OPC-сервер для организации обмена со шлюзом, так как он интегрирован в среду разработки и позволяет автоматически импортировать переменные проекта.

Создайте стандартный проект CODESYS. Добавьте в проект **Символьную конфигурацию** – Нажмите правой кнопкой мыши на «Application» в дереве устройств – «Добавить» – «Символьная конфигурация» (см. рис. 3.46).

При добавлении компонента пользователь может выбрать следующие настройки:

– Включить комментарии в XML – если установлена галочка, то в файл символьной конфигурации будут включены комментарии к переменным;

– Поддержка функций OPC UA – если установлена галочка, то в файл символьной конфигурации добавляется дополнительная информация, необходимая для поддержки функций OPC UA сервера.

– Размещение данных клиента – пользователь может выбрать структуру файла символьной конфигурации – совместимую со старыми версиями или оптимизированную. После добавления компонента Символьная конфигурация следует выполнить компиляцию проекта.

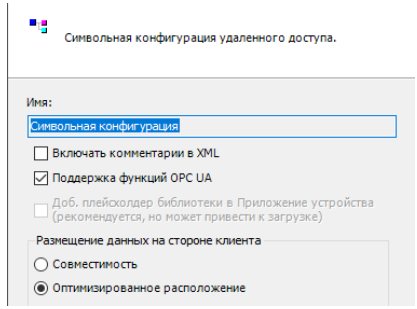


Рис. 3.46. Добавление символьной конфигурации в проект

Объявите в программе переменные.

```
1  PROGRAM PLC_PRG
2  VAR
3      xValue: BOOL;  // логическое значение
4      iValue: INT;   // целое число
5      rValue: REAL;   // число с плавающей точкой
6
7  END_VAR
```

Рис. 3.47. Переменные

Отметьте галочками переменные, которые будут считываться/изменяться OPC-сервером и указать для каждой из них права доступа (со стороны OPC-сервера). После этого настройка контроллера считается завершенной.

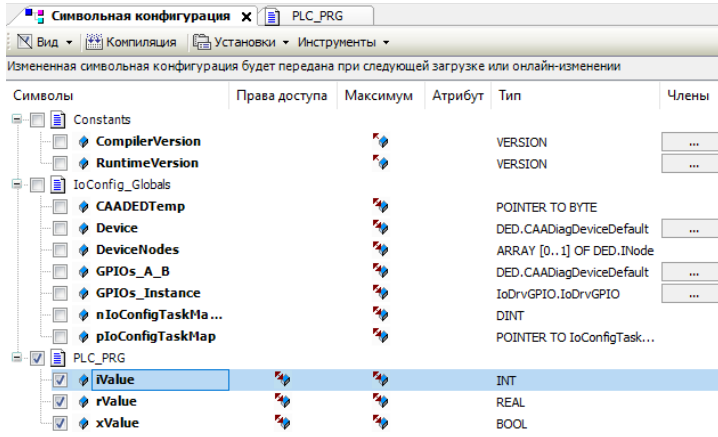


Рис. 3.48. Символьная конфигурация

3.5.1. Настройка OPC-сервера

Запустите приложение **OPC Configurator** из меню Пуск или папки CODESYS OPC Server V3, расположенной в директории установки CODESYS.

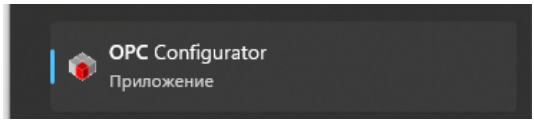


Рис. 3.49. OPC Configurator в меню Пуск

Нажмите на узел «Server» и в контекстном меню выберите «Append PLC». Нажмите на узел «Connection» – «Edit» для настройки IP-адреса шлюза и его порта (см. рис. 3.50).

Рис. 3.50. Настройка IP-адреса и порта шлюза

Далее сохраните настройки – нажмите «File – Save» в верхнем меню OPC Configurator.

3.5.2. Создание сертификата для сервера CODESYS OPC UA

Для шифрования данных и безопасного обмена ими с клиентом серверу необходим сертификат, который клиент должен классифицировать как доверенный при первом установлении соединения.

Требование: Задан активный путь к контроллеру.

1. Установите дополнение CODESYS Security Agent.

2. Щелкните Вид – Безопасность (Security Screen).
3. Выберите вкладку «Устройства (Devices)».
4. Выберите шлюз в левом представлении.

⇒ В правом представлении отображаются все службы контроллера, для которых требуется сертификат.

5. Выберите службу сервера OPC UA.
6. Создайте новый сертификат для устройства. Для этого щелкните значок.

⇒ Откроется диалоговое окно «Настройки сертификата».

7. Определите параметры сертификата и нажмите OK, чтобы закрыть диалоговое окно.

⇒ Сертификат создан на шлюзе (см. рис. 3.51).

8. Перезапустите систему исполнения.

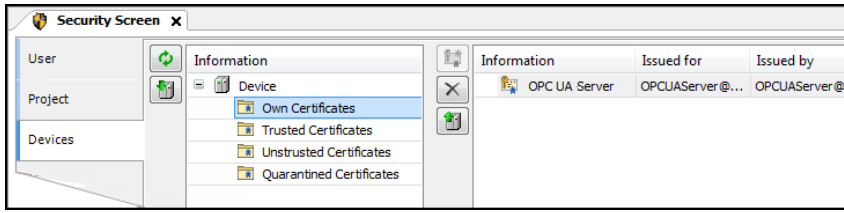


Рис. 3.51. Сертификат создан

3.5.3. Настройка зашифрованного соединения с клиентом UaExpert

Клиент OPC UA «UaExpert» представляет собой свободное доступное ПО, которое можно загрузить из Интернета. Этот клиент можно использовать для подключения к серверу CODESYS OPC UA. Следующее описание относится к этой программе. Другие клиенты OPC UA работают аналогичным образом.

1. Запустите программу UaExpert.
2. Щелкните Сервер «Добавить» и откроется диалоговое окно «Добавить сервер».
3. Выберите «Custom Discovery». Дважды кликните по опции «Double Click to Add Server». Введите параметры подключения (IP-адрес и порт) и нажмите «OK».

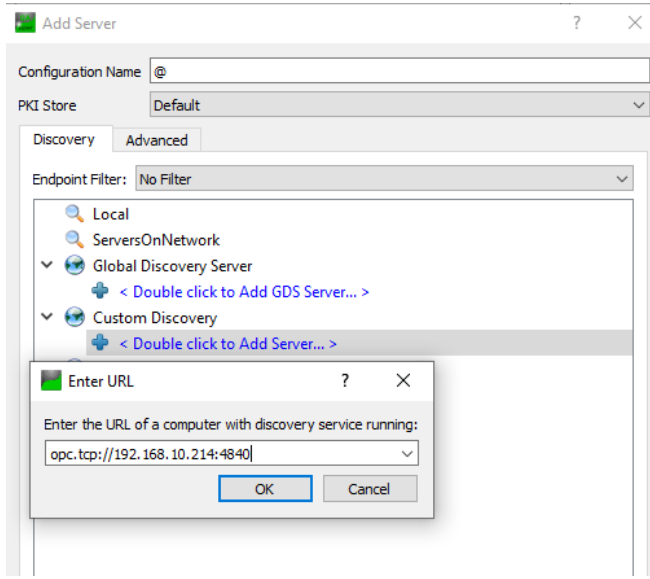


Рис. 3.52. Добавление сервера для подключения

4. Выберите тип подключения Basic256Sha256 — Sign & Encrypt (uatcp-uasc-uabinary) и нажмите «OK», чтобы закрыть диалоговое окно (см. рис. 3.53).

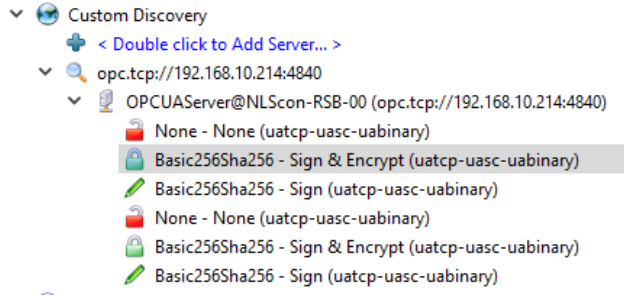


Рис. 3.53. Выбор типа подключения

5. Щелкните Сервер «Подключить» и откроется диалоговое окно проверки сертификата с сообщением об ошибке.

6. Выберите параметр «Временно принять сертификат сервера для этого сеанса» и нажмите «Продолжить».

7. В среде разработки CODESYS щелкните символ  и изображение обновится.

8. Выберите папку «Сертификаты на карантине» (см. рис. 3.54). Сертификат клиента UaExpert@… отображается справа.

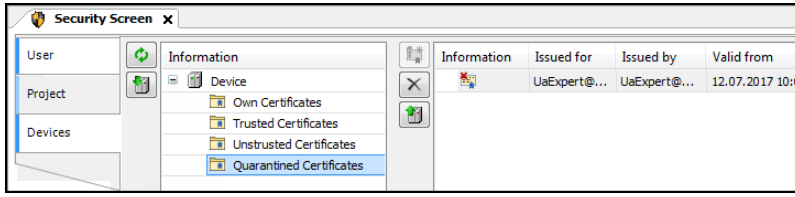


Рис. 3.54. Сертификаты на карантине

9. Перетащите сертификат в папку «Доверенные сертификаты». Теперь сертификат клиента классифицируется сервером как доверенный.

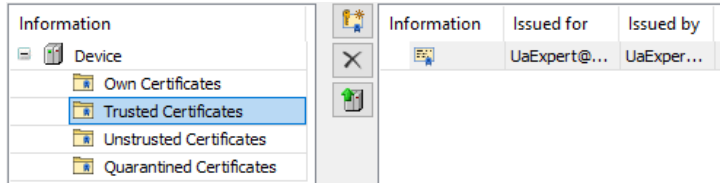


Рис. 3.55. Доверенные сертификаты

10. Щелкните Сервер «Подключить в клиенте UaExpert». Откроется диалоговое окно проверки сертификата с сообщением об ошибке.

11. Выберите параметр «Временно принять сертификат сервера для этого сеанса» и нажмите «Продолжить». Соединение установлено, и объекты отображаются в представлении «Адресное пространство».

3.5.4. Использование клиента OPC UA Server для изменения переменной

1. Разверните объект Objects ▸ DeviceSet ▸ CODESYS OPC UA ▸ Application ▸ Global Vars ▸ GVL в клиенте UaExpert представления Address Space. Переменные списка глобальных переменных видны (см. рис. 3.56).

2. Выберите переменные и перетащите их в представление доступа к данным. Отображаются переменные и их текущие значения.

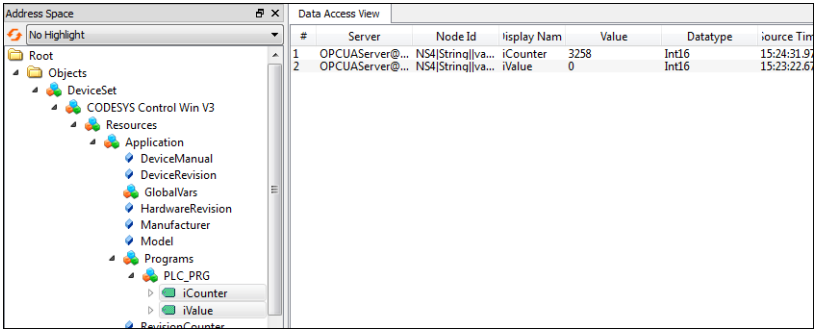


Рис. 3.56. Переменные в UaExpert

3. Измените значения переменных, дважды щелкнув поле «Значение».

3.6. Сбор данных по IIoT-протоколам

3.6.1. MQTT в CODESYS

Описание получения данных по MQTT с примером парсинга JSON-данных и подключением к MQTT-брокеру с использованием шифрования TLS находится в п. 3.1.

3.6.2. Получение данных с модулей NLS-xx-Ethernet с поддержкой MQTT

Модули дискретного и аналогового ввода-вывода серии NLS-Ethernet (например, NLS-16DI-Ethernet, NLS-16DO-Ethernet, NLS-8AI-Ethernet и другие) оснащены встроенным MQTT-клиентом. Это позволяет передавать телеметрию напрямую в брокер сообщений шлюза или в облачную платформу без постоянного циклического опроса (передача выполняется по таймеру).

Предварительное конфигурирование модуля

Для организации работы с модулем по протоколу MQTT необходимо выполнить предварительное конфигурирование.

Подключитесь к встроенному веб-интерфейсу модуля NLS-xx-Ethernet через браузер (по заводскому IP-адресу, например, 192.168.0.1).

В разделе «Сетевые настройки MQTT» задайте параметры подключения:

- Broker IP / Host: IP-адрес шлюза NLS-Gateway (или локальный IP брокера);
- Port: порт брокера (по умолчанию 1883);

- Цикл отправления данных.

Рис. 3.57. Меню веб-интерфейса «Настройки MQTT»

Просмотр и запись данных в MQTT Explorer (см. рис. 3.58).

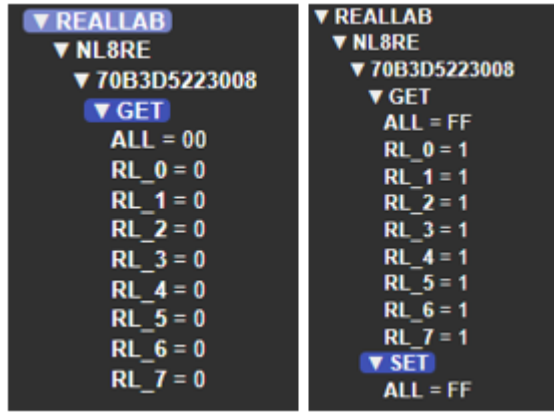


Рис. 3.58. Список релейных выходов NLS-8R-Ethernet

Примечание: Более подробная информация о конфигурировании и эксплуатации модулей приведена в эксплуатационной документации на странице конкретной модели на официальном сайте компании.

Приём данных в CODESYS 3.5

Для приёма данных в среде CODESYS 3.5 создайте проект и настройте его аналогично проекту из п. 3.1. Также можете скачать проект-пример получения данных по MQTT с сайта RealLab и использовать его как базу для дальнейшей работы.

Прием данных сводится к указанию корректного топика (переменная «wsSubscribeTopicFilter») для подписки и топика для публикации (переменная «wsPublishTopic») (см. рис. 3.59).

```
8       wsPublishTopic : WSTRING(1024) := "REALLAB/NL8RE/70B3D5223008/SET/ALL";
9       wsSubscribeTopicFilter : WSTRING(1024) := "REALLAB/NL8RE/70B3D5223008/GET/ALL";
```

Рис. 3.59. Пример конфигурации топиков для чтения и записи параметров NLS-8R-Ethernet по MQTT

Далее объявите переменные для чтения и записи значений. Т.к. считывается и записывается шестнадцатеричное значение релейных выходов, преобразованное в формат строки, необходимо преобразовать строку в число в формате BYTE, чтобы его можно было удобно разбить на биты и наоборот (см. рис. 3.60).

```
// Чтение
byRelayMask : BYTE;         // Маска в виде байта (0..255)
xRL : ARRAY[0..7] OF BOOL;  // Состояния реле RL_0 .. RL_7
sHexStr : STRING(10);
// Запись
xSetRL : ARRAY[0..7] OF BOOL;      // Переключатели реле (управление из визу-ии или кода)
xSetRL_old : ARRAY[0..7] OF BOOL; // Буфер для отслеживания изменений
bySetMask : BYTE;                  // Сформированный байт
xSendCmd : BOOL;                   // Флаг отправки
i : INT;
```

Рис. 3.60. Переменные для чтения и записи значений в NLS-8R-Ethernet по MQTT

Алгоритм чтения – если ФБ mqttSubscriber получил данные с топика, на который подписан, он добавляет к полученной строке «16#» и преобразует ее в байт с помощью функции STRING_TO_BYTE. Далее байт раскладывается на биты в массив xRL[0..7]. Соответственно, текущие значения выходов можно вывести в визуализацию, назначив на индикаторы значения xRL[0] – xRL[7] (см. рис. 3.61)

Рис. 3.61. Назначенная переменная xRL[0] на индикатор «Выход 1»

На переключатели под индикаторами назначены переменные xSetRL[0] – xSetRL[7]. Каждый цикл программа сверяет старые значения выходов xSetRL_old и новые xSetRL. Если они отличаются, то байт значений выходов конвертируется в HEX-строку и отправляется в топик …/SET/ALL для изменения значений релейных выходов модуля.

Загрузите проект на шлюз, подключитесь к MQTT-брокеру и подпишитесь на топик, в который приходят данные о состоянии выходов модуля NLS-xx-Ethernet. Попробуйте изменить состояние выходов кнопками под индикаторами. Если все настроено правильно – чтение и запись будет работать (см. рис. 3.62).

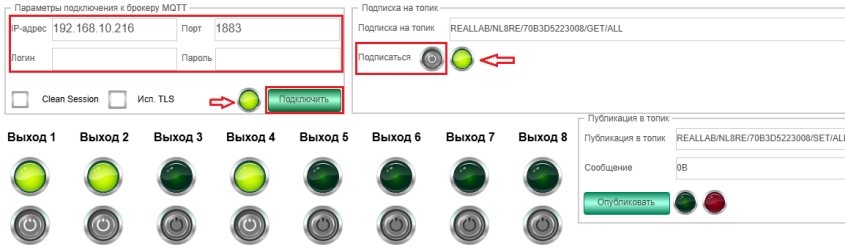


Рис. 3.62. Результат работы проекта (визуализация)

3.6.3. LoRaWAN

Ниже представлено пошаговое описание процесса передачи данных от LoRaWAN-датчика до шлюза с поддержкой CODESYS.

1. Датчик получает данные с подключенных к нему сенсоров (влажность, температура, атмосферное давление) и формирует пакет данных. Сформированный пакет отправляется на шлюз, к которому подключен датчик, по протоколу LoRaWAN.

2. Шлюз, оснащенный LoRa-концентратором, принимает пакет данных, полученный по радиоканалу от LoRaWAN-датчика. ПО Chirpstack (компонент MQTT Forwarder) кодирует полученный пакет в формат JSON либо Protobuf (в зависимости от настроек) и публикует его в специальном топике MQTT-брокера (параметры наименования топика, а также подключения к брокеру можно настроить через веб-конфигуратор RealLab).
3. В CODESYS создается проект, добавляется библиотека, предоставляющая функционал клиента MQTT. В программу добавляются функциональные блоки: MQTTClient, MQTTSubscribe, MQTTPublish. Настраиваются параметры подключения к брокеру (IP-адрес, порт, логин, пароль, параметры шифрования). Указывается топик подписки для MQTTSubscribe и топик публикации для MQTTPublish. Пишется код обработки полученных данных с помощью ФБ MQTTSubscribe. Возможно вывести данные на визуализацию, либо отправить их по протоколам Modbus (RTU/TCP), CANopen, MQTT, OPC UA и т.д.

На рис. 3.63 приведена обобщённая схема информационных потоков в системе термометрии с LoRa датчиками.

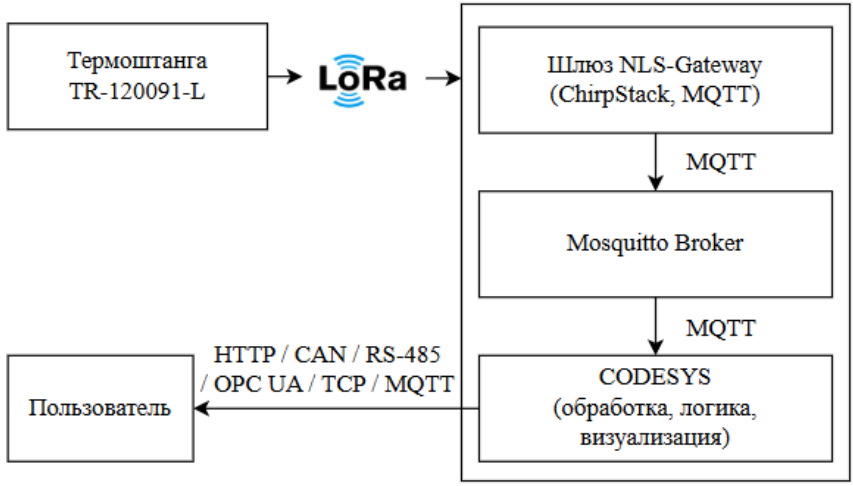


Рис. 3.63. Архитектура LoRa интеграции в шлюзе с поддержкой CODESYS

Приём и декодирование пакетов от LoRa-датчиков

Для подключения датчика и приёма его телеметрии в среде CODESYS необходимо выполнить следующие шаги:

1. **Настройка в ChirpStack**:

– В веб-интерфейсе сетевого сервера ChirpStack создать профиль устройства (Device Profile) и зарегистрировать приложение (Application).

– Добавить конечное устройство, указав идентификаторы и ключи аутентификации (DevEUI, AppKey / AppSKey, NwkSKey в зависимости от режима активации OTAA или ABP).

– В свойствах профиля устройства добавить JavaScript-код кодека (парсера) для автоматического декодирования (Payload Codec).

Подробное описание добавления устройств и настройки кодеков приведено в руководстве по работе с ПО шлюзов RealLab! серии NLS-Gateway на официальном сайте компании.

2. **Настройка приёма в CODESYS и обработка данных:**

Описание получения данных по MQTT с примером парсинга JSON-данных и подключением к MQTT-брокеру с использованием шифрования TLS находится в п. 3.1.

Пример получения данных с LoRaWAN-термоштанги можно скачать с сайта компании.

Возьмите за основу проект из п. 3.1.

Из визуализации удалите таблицы с отображением частей обработанного JSON-пакета с данными от LoRaWAN-датчика.

В PLC_PRG укажите параметры подключения к MQTT-брокеру и параметры подписки на топик (пример на рис. 3.64).

```
sHostname : STRING(255) := '192.168.0.93';
uiPort : UINT := 1883;
wsPublishTopic : WSTRING(1024) := "codesys/publication/tl";
wsSubscribeTopicFilter : WSTRING(1024) := "application/69a1894c-b278-4016-8e18-006626lfe4dc/device/70b3d52230080006/event/up";
sWillMessage : STRING(1024);
sPublishMessage : STRING(1024) := 'just_a_message';
sSubscribeMessage : STRING(4096);
```

Рис. 3.64. Параметры подключения к MQTT-брокеру и параметры подписки на топик

Укажите ключи, значения которых нужно получить, в массиве aKeysToSearch. Объявите триггеры для исключения повторной обработки массива температур (разделение строки со списком температур на отдельные значения), также объявите переменные для хранения температур после разделения (см. рис. 3.65).

```
fbExtractKeys : FB_ExtractSelectedKeys;
aKeysToSearch : ARRAY[0..9] OF WSTRING(255) := ["deviceName", "battery", "humidity", "temperatures", "gwTime", "rssi", "snr"];
aResults : ARRAY[0..9] OF WSTRING(255);

// Разделение температур на элементы массива
t_termo_s: STRING;
temp_termo_arr: ARRAY[1..9] OF STRING;
i: DINT := 0;

// ТРИГГЕРЫ ДЛЯ ИСКЛЮЧЕНИЯ ПОВТОРНОЙ ОБРАБОТКИ
rtMqttReceived : R_TRIG; // Триггер на приход MQTT сообщения
rtJsonDone     : R_TRIG; // Триггер на готовность JSON
```

Рис. 3.65. Переменные хранения температур, ключи для поиска значений температуры, влажности, имени устройства и т.д.

Как выглядит пакет целиком можно посмотреть через любой MQTT-клиент (например, MQTT Explorer, см. рис. 3.66).

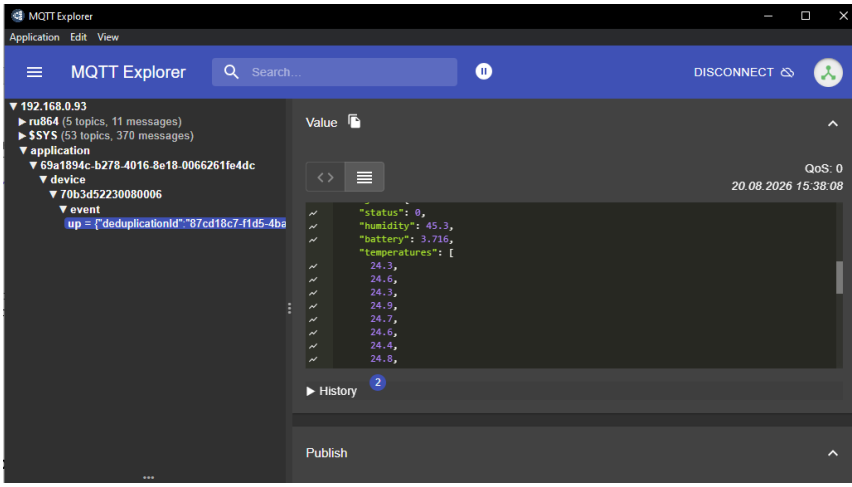


Рис. 3.66. Получение данных через MQTT Explorer

В цикле напишите код для запуска выборки ключей и разделения строки со списком температур термоштанги на составляющие (см. рис. 3.67).

```
45      // Парсинг температур (по триггеру)
46      t_termo_s := WSTRING_TO_STRING(aResults[3]);
47
48      IF FIND(t_termo_s, '[') > 0 THEN
49          t_termo_s := DELETE(t_termo_s, 1, FIND(t_termo_s, '['));
50      END_IF
51      IF FIND(t_termo_s, ']') > 0 THEN
52          t_termo_s := DELETE(t_termo_s, 1, FIND(t_termo_s, ']'));
53      END_IF
54
55      FOR i := 1 TO 9 DO
56          temp_termo_arr[i] := '';
57      END_FOR;
58
59      i := 1;
60      WHILE LEN(t_termo_s) > 0 AND i <= 9 DO
61          IF FIND(t_termo_s, ',') > 0 THEN
62              temp_termo_arr[i] := LEFT(t_termo_s, FIND(t_termo_s, ',') - 1);
63              t_termo_s := DELETE(t_termo_s, FIND(t_termo_s, ','), 1);
64          ELSE
65              temp_termo_arr[i] := t_termo_s;
66              t_termo_s := '';
67          END_IF;
68
69          i := i + 1;
70      END_WHILE;
71  END_IF
```

Рис. 3.67. Код разделения списка температур на составляющие

В визуализацию добавьте вывод полученных данных. Температуры хранятся в массиве temp_termo_arr. Остальные параметры (имя устройства, заряд, влажность, качество сигнала и пр.) хранятся в массиве aResults.

После загрузки проекта на шлюз откройте визуализацию, нажмите «подключить» в блоке подключения к брокеру и подпишитесь на топик. После успешного подключения и подписки на топик соответствующие индикаторы загорятся зеленым.

После того, как датчик пришлет данные на шлюз, они сразу же будут отображены в визуализации.

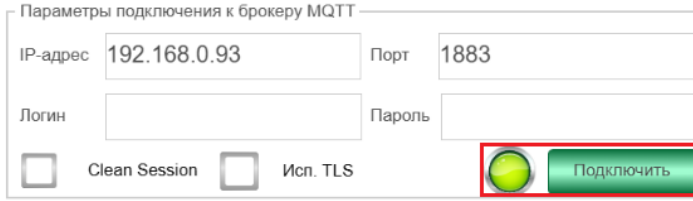


Рис. 3.68. Успешное подключение к брокеру

Рис. 3.69. Успешная подписка на топик и успешная обработка полученного пакета

Имя устройства	Заряд (В)	Влажность (%)	RSSI (Качество сигнала) дБм	SNR(Сигнал/Шум)	Время получения
TR120091-L_0006	3.716	47.4	-95	5.75	2026-08-20T12:48:07.825

Темп. 1	Темп. 2	Темп. 3	Темп. 4	Темп. 5	Темп. 6	Темп. 7	Темп. 8	Темп. 9
24.3	24.6	24.3	25.0	24.7	24.6	24.4	24.8	23.4

Рис. 3.70. Пример отображения полученных данных

3.7. Энергонезависимые переменные

Реманентные переменные могут сохранять свои значения в течение обычного цикла программы. Вы можете объявлять реманентные переменные как RETAIN-переменные или даже как перманентные переменные в приложении.

RETAIN-переменные объявляются с использованием ключевого слова RETAIN после ключевого слова типа переменной (VAR, VAR_GLOBAL и т.д.) в разделе объявления программного объекта. Они сохраняют свои значения после непредвиденного отключения (или онлайн-команды Сброс). При перезапуске программы система продолжает работу с сохраненными значениями. При этом CODESYS инициализирует ‘обычные’ переменные либо с заданными значениями, либо с начальными значениями по умолчанию.

Пример инициализации в POU:

```
VAR RETAIN
counter:INT:=1;
END_VAR
```

В GVL:

```
VAR_GLOBAL RETAIN
gvarRem : INT;
END_VAR
```

PERSISTENT-переменные при загрузке нового проекта сохраняют свои значения, если список PERSISTENT-переменных в новом проекте не отличается от старого.

PERSISTENT-переменные удобно использовать в тех случаях, когда заранее известно, что проект будет дорабатываться – например, он длительное время будет находиться в опытной эксплуатации, за время которой обслуживающий персонал будет формулировать пожелания и замечания. В течение этого времени операторы могут задавать уставки, параметры рецептов и т. д.

Для их добавления вызовите контекстное меню кликом правой кнопкой мыши по «Application» в дереве устройств. Выберите «Добавление объекта» и «Persistent-переменные» (см. рис. 3.71).

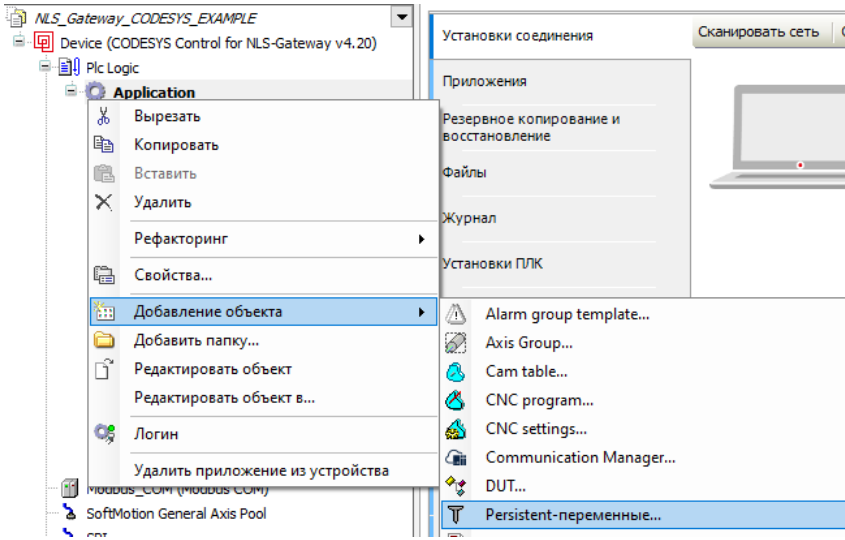


Рис. 3.71. Добавление списка перманентных переменных

Далее появляется вкладка, где вы можете перечислить все перманентные переменные.

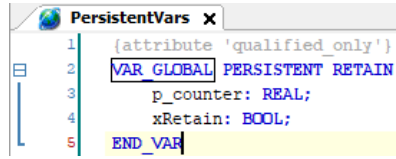


Рис. 3.72. Запись переменных

Если некоторые переменные объявлены в других структурах и POU, то необходимо указать все пути экземпляров, как показано на рис. 3.73 и рис. 3.74.

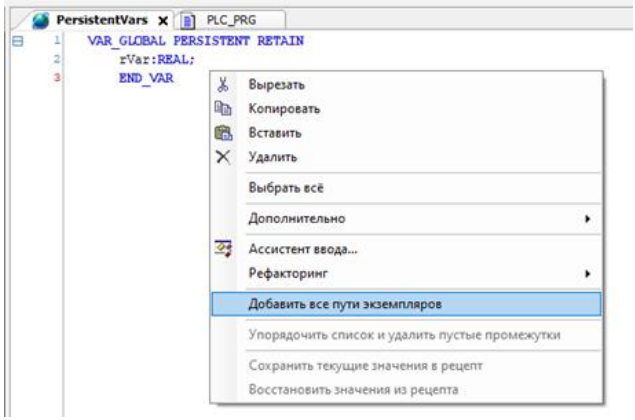


Рис. 3.73. Пример добавления путей экземпляров

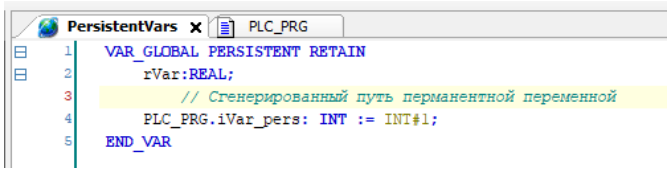


Рис. 3.74. Пример итогового результата

Примечание:

Начиная с CODESYS V3.3.0.1, объявление VAR_GLOBAL PERSISTENT имеет тот же действие, что и объявление VAR_GLOBAL PERSISTENT RETAIN или VAR_GLOBAL RETAIN PERSISTENT.

3.8. Логирование данных в среде CODESYS

Логирование и архивирование данных в среде **CODESYS** для шлюза – важная функция для мониторинга, диагностики и анализа работы автоматизированной системы.

3.8.1. Общие принципы логирования в CODESYS

Логирование в CODESYS может включать:

– Запись значений переменных (датчиков, состояний, ошибок и т. д.).

– Временные метки (timestamp).

– Сообщения о событиях (запуск, остановка, ошибка и т. д.).

– Архивирование на SD-карту, USB-накопитель или внешний сервер.

В процессе отладки и эксплуатации системы автоматизации важную роль играет логирование информации об ошибках и событиях. В шлюзах и ПЛК с системой исполнения CODESYS V3.5 для этих целей используется компонент PLC Logger (журнал ПЛК), основанный на системной библиотеке **CmpLog.** Стандартные компоненты CODESYS используют возможности библиотеки для записи в лог сообщений о своей работе. Кроме того, пользователь может использовать библиотеку для записи в лог собственных сообщений.

Библиотека **CmpLog** – это встроенная системная библиотека, предназначенная для логирования событий, ошибок и отладочной информации в процессе выполнения программы на шлюзе (ПЛК).

3.8.2. Функции библиотеки CmpLog

LogAdd – добавляет запись в логгер (для совместимости со старыми версиями библиотек);

LogAdd2 – добавляет запись в лог, позволяет указать ссылку на логгер (можно использовать стандартный – LogConstant.LOG_STD_LOGGER) и уникальный ID компонента;

LogCreate – создаёт новый логгер;

LogDelete – удаляет логгер;

LogOpen – открывает существующий логгер;

LogClose – закрывает логгер и обнуляет его дескриптор.

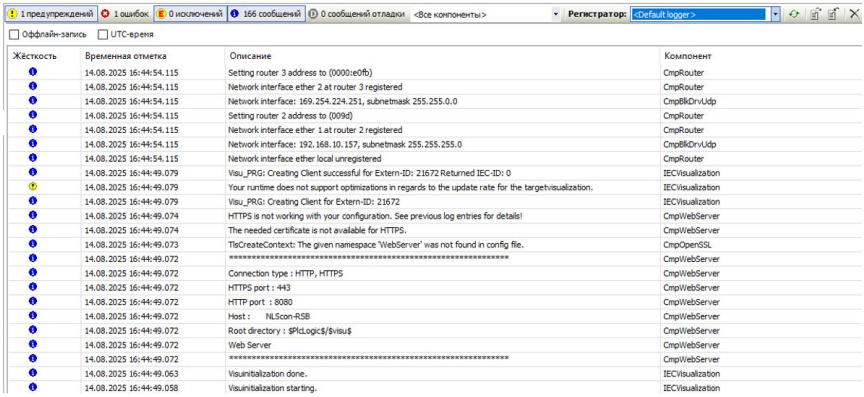
Информация об ошибках и событиях шлюза (ПЛК) отображается в CODESYS на вкладке Device – Журнал (см. рис. 3.75). Для того чтобы просмотреть журнал – необходимо подключиться к контроллеру. Для каждого события отображаются:

– **Жесткость** – класс события (Предупреждение /Ошибка /Исключение /Сообщение /Сообщение отладки);

– **Временная отметка** – метка времени события. По умолчанию отображается с учетом часового пояса шлюза, при установленной галочке UTC-время – метка времени отображается в UTC+0;

– **Описание** – текст сообщения;

– **Компонент** – название компонента, сформировавшего событие.



Жесткость	Временная отметка	Описание	Компонент
	14.08.2025 16:44:54.115	Setting router 3 address to (0000:e0fb)	CmpRouter
	14.08.2025 16:44:54.115	Network interface ether 2 at router 3 registered	CmpRouter
	14.08.2025 16:44:54.115	Network interface: 169.254.224.251, subnetmask 255.255.0.0	CmpBlkDrvUdp
	14.08.2025 16:44:54.115	Setting router 2 address to (009d)	CmpRouter
	14.08.2025 16:44:54.115	Network interface ether 1 at router 2 registered	CmpRouter
	14.08.2025 16:44:54.115	Network interface: 192.168.10.157, subnetmask 255.255.255.0	CmpBlkDrvUdp
	14.08.2025 16:44:54.115	Network interface ether local unregistered	CmpRouter
	14.08.2025 16:44:49.079	Visu_PRG: Creating Client successful for Extern-ID: 21672 Returned IEC-ID: 0	IECVisualization
	14.08.2025 16:44:49.079	Your runtime does not support optimizations in regards to the update rate for the target/visualization.	IECVisualization
	14.08.2025 16:44:49.079	Visu_PRG: Creating Client for Extern-ID: 21672	IECVisualization
	14.08.2025 16:44:49.074	HTTPS is not working with your configuration. See previous log entries for details!	CmpWebServer
	14.08.2025 16:44:49.074	The needed certificate is not available for HTTPS.	CmpWebServer
	14.08.2025 16:44:49.073	TlsCreateContext: The given namespace 'WebServer' was not found in config file.	CmpOpenSSL
	14.08.2025 16:44:49.072	**********	CmpWebServer
	14.08.2025 16:44:49.072	Connection type : HTTP, HTTPS	CmpWebServer
	14.08.2025 16:44:49.072	HTTPS port : 443	CmpWebServer
	14.08.2025 16:44:49.072	HTTP port : 8080	CmpWebServer
	14.08.2025 16:44:49.072	Host : NLScon-RSB	CmpWebServer
	14.08.2025 16:44:49.072	Root directory : $PlcLogic$/$visu$	CmpWebServer
	14.08.2025 16:44:49.072	Web Server	CmpWebServer
	14.08.2025 16:44:49.072	**********	CmpWebServer
	14.08.2025 16:44:49.063	Visuinitialization done.	IECVisualization
	14.08.2025 16:44:49.058	Visuinitialization starting.	IECVisualization

Рис. 3.75. Журнал контроллера (лог сообщений)

Пока актирован режим Auto scroll – при появлении нового события оно будет автоматически отображено в логгере (если режим отключен – то новые события отображаться не будут). Кнопки используются для экспорта/импорта лога в файл формата .xml. Кнопка очищает окно лога.

В случае установки галочки Оффлайн-запись – при отключении от контроллера окно лога будет автоматически очищено.

3.8.3. Настройки конфиг-файла

На шлюзах RealLab NLS-Gateway конфигурационные файлы CODESYS находятся в директории **/etc/codesyscontrol/**.

Настройки системного логгера хранятся в конфиг-файле **/etc/codesyscontrol/CODESYSControl.cfg**. Его можно отредактировать через веб-конфигуратор RealLab! Либо с помощью утилиты nano при подключении через SSH.

Откройте веб-конфигуратор RealLab (http://IP-адрес-шлюза:8000). Перейдите на страницу «Интеграции» и выберите вкладку «Codesys» (см. рис. 3.76).

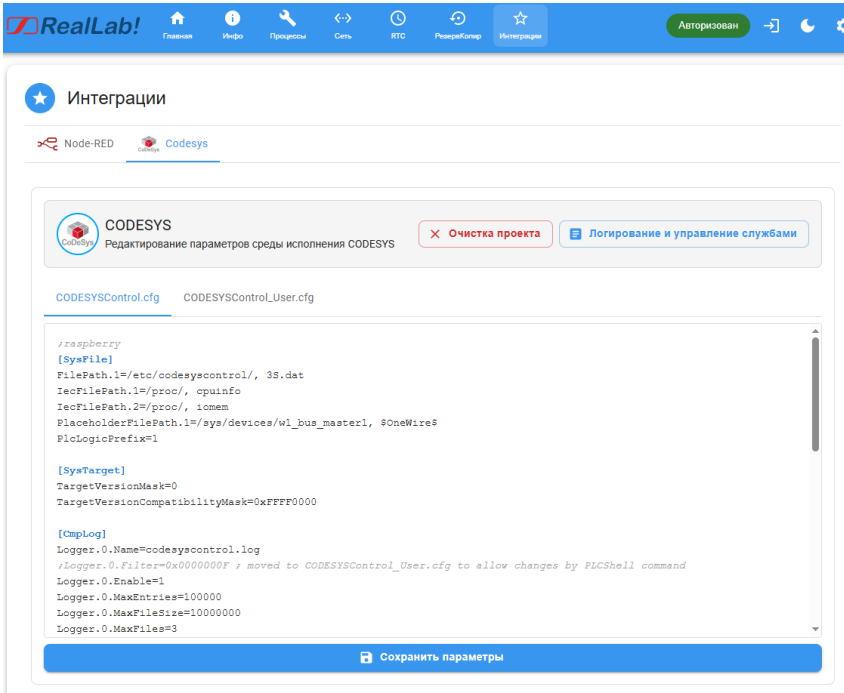


Рис. 3.76. Страница веб-конфигуратора «Интеграции» - вкладка CODESYS

В окне редактирования файла **CODESYSControl.cfg** найдите заголовок [CmpLog]. По умолчанию логгер уже настроен на хранение логов в **/var/opt/codesys/codesyscontrol.log** и их выводе в журнал служб (journalctl) (см. рис. 3.77).

```
[CmpLog]
Logger.0.Name=codesyscontrol.log
;Logger.0.Filter=0x0000000F ; moved to CODESYSControl_User.cfg to allow changes by PLCShell command
Logger.0.Enable=1
Logger.0.MaxEntries=100000
Logger.0.MaxFileSize=10000000
Logger.0.MaxFiles=3
Logger.0.Backend.0.ClassId=0x00000104 ;writes logger messages in a file
Logger.0.Backend.1.ClassId=0x0000010B ;writes to syslog
Logger.0.Backend.1.Flags=0x00000000
```

Рис. 3.77. Настройка логгера по умолчанию

Если закомментировано, то используются значения по умолчанию. Параметры логгера секция [CmpLog] имеют следующие настройки, согласно табл. 3.

Табл. 3. Параметры настроек логгера секция [CmpLog]

Настройка	Описание
секция [CmpLog]	Настройка логгера
Name	Имя логгера (если логи сохраняются в файлах – то названия файлов состоят из имени логгера и порядкового номера – например, StdLogger_0.csv и т.д.)
Enable	1 – логгер включен, 0 – логгер отключен
MaxEntries	Максимальное число записей в логгере. По достижению этого значения – самые старые записи начинают перезаписываться
MaxFileSize	Максимальный размер файла лога в байтах (используется в том случае, если бэкенд логгера – файлы). При достижении файлом этого размера – создается новый файл лога (см. MaxFiles)
MaxFiles	Максимальное число файлов логов (используется в том случае, если бэкенд логгера – файлы). Если создано максимальное число файлов и размер последнего достиг MaxFileSize – то начинается перезапись самого старого файла
Backend.<№>.ClassId	Тип бэкенда. Из внутренней документации известно, что значение 0x104 соответствует файлам, 0x135 – отправке на syslog-сервер, а 0x10B – выводу лога в последовательный порт (через компонент SysOut)
Filter	Фильтр классов сообщений, которые поддерживаются логгером. Значение фильтра представляет собой комбинацию флагов LogClass. По умолчанию используется значение 0xFFFFFFFF (LOG_ALL) – в этом режимы логгер поддерживает все классы событий

Для корректной работы необходимо добавить библиотеку **CmpLog** в секцию [ComponentManager] в настройках конфиг-файла **CODESYSControl_User.cfg** (см. рис. 3.78).

Рис. 3.78. Компонент CmpLog добавлен в ComponentManager

После внесения изменений в конфиг-файлы нажмите кнопку «Применить» в нижней части страницы. Изменения будут применены, среда исполнения CODESYS будет перезапущена.

Логи можно прочитать с помощью меню «Логирование и управление службами на той же странице, где настраиваются конфиг-файлы CODESYS (см. рис. 3.79).

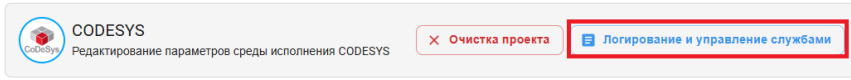


Рис. 3.79. Кнопка открытия меню для просмотра логов

В открывшемся диалоговом окне выберите количество строк логов, отображаемых с конца или снимите ограничение на вывод, сняв галочку слева от поля для ввода «Кол-во строк». Затем нажмите кнопку «Обновить» в верхней правой части окна. Если логи необходимо скачать, нажмите кнопку «Скачать» (см. рис. 3.80 и рис. 3.81).

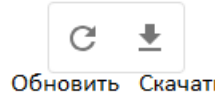


Рис. 3.80. Кнопки обновления и скачивания логов

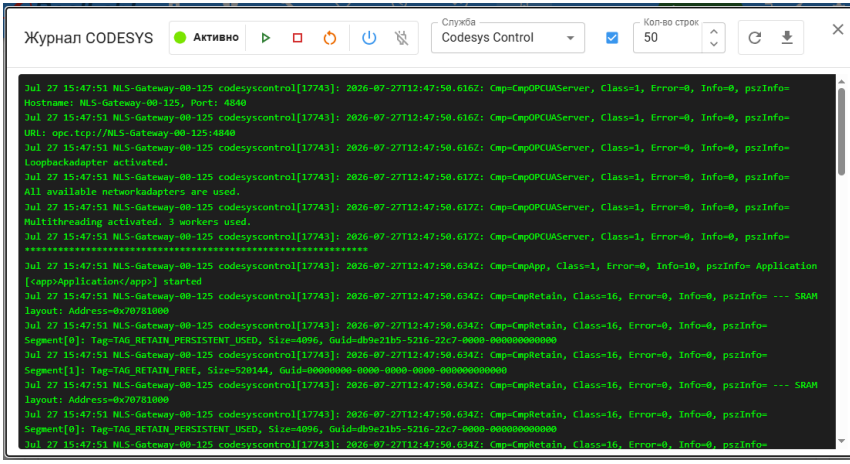


Рис. 3.81. Диалоговое окно «Журнал CODESYS»

Также можно просмотреть лог через терминал. Для этого подключитесь к шлюзу по SSH, откройте терминал и вызовите команду:

$sudo cat /var/opt/codesys/codesyscontrol.log

3.9. Взаимодействие с внешними устройствами (SD/USB)

3.9.1. Проверка и подготовка карты памяти для ее автоматического монтирования в ОС RealLab! Raspbian Linux arm64

Важно! После подключения SD-карты подключитесь к шлюзу по SSH (например, с помощью Bitvise SSH Client, логин: pi, пароль: 123) и выполните команду **lsblk**. Если напротив mmcblk2pX отображается путь монтирования в столбце MOUNTPOINTS (см. рис. 3.82), то SD-карта готова к работе.

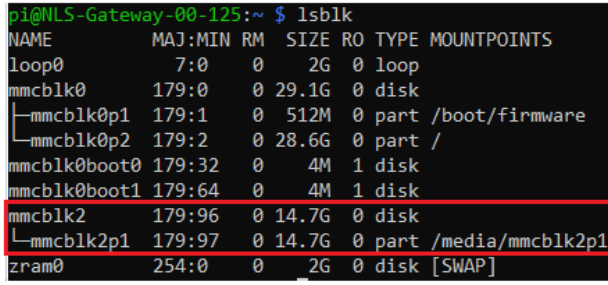


Рис. 3.82. Смонтированный раздел карты памяти (mmcblk2p1)

Если же отображается только mmcblk2 без разделов и точки монтирования (см. рис. 3.83), то необходимо отформатировать карту памяти в EXT3/4 или exFAT.

```
pi@NLS-Gateway-00-125:~ $ lsblk
NAME         MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0          7:0    0    2G  0 loop
mmcblk0      179:0    0 29.1G  0 disk
├─mmcblk0p1  179:1    0  512M  0 part /boot/firmware
└─mmcblk0p2  179:2    0 28.6G  0 part /
mmcblk0boot0 179:32   0    4M  1 disk
mmcblk0boot1 179:64   0    4M  1 disk
mmcblk2      179:96   0 14.7G  0 disk
zram0        254:0    0    2G  0 disk [SWAP]
```

Рис. 3.83. Карта памяти (mmcblk2) без разделов и точек монтирования

Пример форматирования в exFAT:

Выполните команду **sudo fdisk /dev/mmcblk2** для создания раздела на карте памяти (см. рис. 3.84). Введите «o» и нажмите Enter для создания таблицы разделов. Введите «n» для создания нового раздела, затем «p». Соглашайтесь с предлагаемыми значениями нажатием кнопки Enter. Введите «w» и нажмите Enter для применения изменений и выхода из утилиты (см. рис. 3.88).

Затем выполните команду **sudo mkfs.exfat /dev/mmcblk2p1** для форматирования созданного раздела в exFAT (см. рис. 3.85) либо **sudo mkfs.ext4 /dev/mmcblk2p1** для форматирования созданного раздела в ext4.

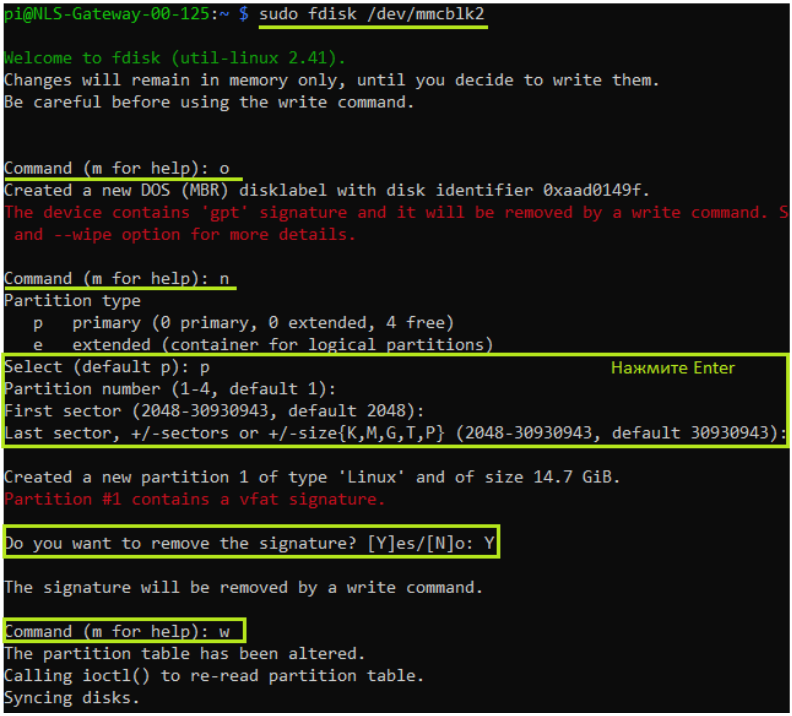


Рис. 3.84. Создание раздела на карте памяти

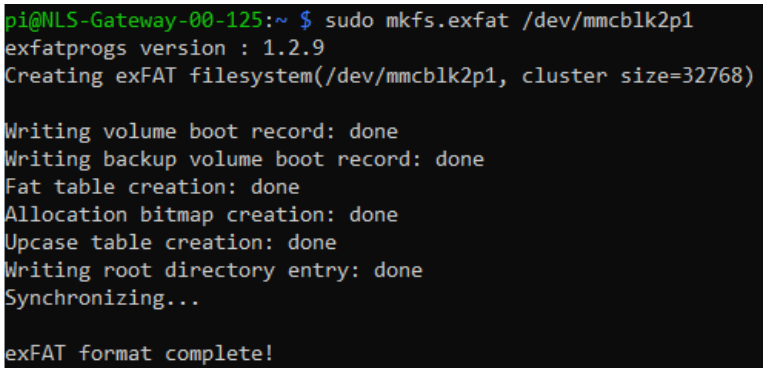


Рис. 3.85. Форматирование раздела карты памяти в exFAT

Извлеките карту памяти из шлюза и вставьте ее обратно. Выполните команду **lsblk**, раздел SD-карты должен получить точку монтирования (см. рис. 3.86).

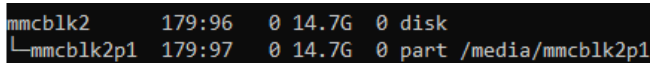


Рис. 3.86. Смонтированный раздел карты памяти (mmcblk2p1)

Выполните команду **sudo blkid**, найдите строку «/dev/mmcblk2p…». Свойство «TYPE» отображает текущую файловую систему SD-карты (exFAT) (см. рис. 3.87).

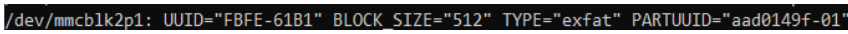


Рис. 3.87. Информация о подключенной карте памяти

Карта памяти готова к использованию!

Подключенная карта памяти будет всегда монтироваться по пути: **/mnt/sd/mmcblk2p***

USB-флешка будет монтироваться по пути: /**mnt/usb/sdXY** при отсутствии метки (label).

Если USB-флешка имеет метку (**label**), то она будет монтироваться по пути: /**mnt/usb/метка/**.

X – латинская буква, составляющая конечное название устройства (**sda/sdb/sdc** и т.д.). Первая подключенная флешка обычно определяется как **sda**, вторая – как **sdb**.

Y – номер раздела на флешке. Если раздел один, то флешка будет смонтирована как /**mnt/usb/sda1**, если разделов несколько, то монтирование будет выполнено соответствующе - /**mnt/usb/sda1**, /**mnt/usb/sda2** и т.д.

Если таблица разделов на накопителе отсутствует, то путь монтирования накопителя будет /**mnt/usb/sdX** (пример - /**mnt/usb/sda**).

Если подключено несколько флешек, то они будут монтироваться в /**mnt/usb/sdaY**, /**mnt/usb/sdbY**, /**mnt/usb/sdcY** и т.д.

После подключения накопителей их пути монтирования можно проверить с помощью команды **$ lsblk**.

ВАЖНО! Рекомендуется использовать накопители с файловой системой exFAT. Накопители с файловой системой NTFS не поддерживаются – они будут смонтированы с режимом доступа «Только для чтения».

Во время записи логов и пр. данных на внешнее устройство не рекомендуется отключать его от шлюза.

Например – если USB-флешка вставлена в шлюз, смонтирована в /dev/usb/sda1 и на нее в данный момент производится запись логов с CODESYS, то если в процессе записи ее отключить от шлюза, CODESYS продолжит пытаться записывать данные в ту же директорию и при подключении флешки к шлюзу она будет смонтирована как /dev/sdb1 (или как sdc1, если к шлюзу подключена еще одна флешка). В таком случае запись логов на флешку возобновится только после перезагрузки шлюза.

Чтобы этого избежать, необходимо заморозить или завершить процесс, который связан с записью данных на внешний накопитель, перед тем как снова подключить отключенный от шлюза накопитель. После подключения снова его запустить, убедившись, что устройство смонтировано правильно (в ту же директорию, что и раньше) (в случае с Codesys так делать не рекомендуется, т.к. для этого нужно завершить процесс codesyscontrol.bin, но это остановит работу всей исполнительной системы).

3.9.2. Сохранение данных проекта CODESYS на внешние накопители

Рекомендуется переносить проект на внешние накопители с форматом файловой системы ext4 или exFAT. Файловая система NTFS не поддерживается.

Для того чтобы расположить проект на внешнем устройстве необходимо отредактировать конфигурационный файл исполнительной системы Codesys на шлюзе. Ниже приведен пример для сохранения файлов проекта на sd-карту:

1. Сначала нужно проверить путь монтирования карты памяти командой **lsblk**:

```
pi@NLS-Gateway-00-125:~ $ lsblk
NAME         MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0          7:0    0    2G  0 loop
mmcblk0      179:0    0 29.1G  0 disk
├─mmcblk0p1  179:1    0  512M  0 part /boot/firmware
└─mmcblk0p2  179:2    0 28.6G  0 part /
mmcblk0boot0 179:32   0    4M  1 disk
mmcblk0boot1 179:64   0    4M  1 disk
mmcblk2      179:96   0 14.7G  0 disk
└─mmcblk2p1  179:97   0 14.7G  0 part /media/mmcblk2p1
zram0        254:0    0    2G  0 disk [SWAP]
```

Рис. 3.88. Карта памяти смонтирована в /mnt/sd/mmcblk2p1

2. Затем открыть файл конфигурации **CODESYSControl.cfg** исполнительной среды Codesys в веб-конфигураторе (Интеграции – Codesys).

3. Найдите блок [SysFile] и в конце блока допишите строку (см. рис. 3.89):

PlaceholderFilePath.2=/media/mmcblk2p1/cds_proj, $PlcLogic$

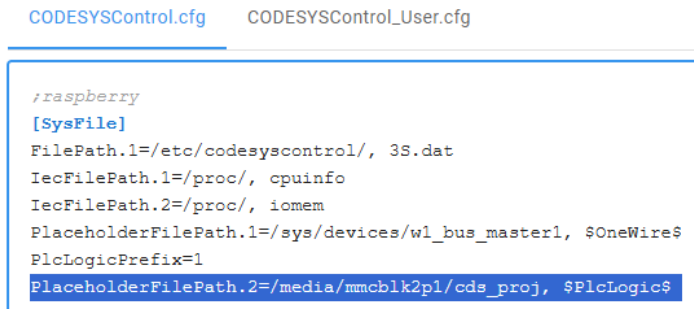


Рис. 3.89. Редактирование конфигурационного файла

4. Нажмите «Применить» и загрузите проект на шлюз.

В указанную папку будет сохранен загруженный на карту памяти шлюза проект.

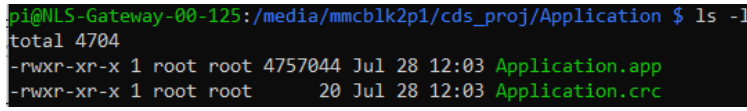


Рис. 3.90. Просмотр содержимого папки проекта CODESYS в назначенной директории

5. Чтобы снова сохранять проект на внутреннюю память достаточно закомментировать строку PlaceholderFilePath.2 в конфигурационном CODESYSControl.cfg (поставить точку с запятой в начале строки).

4. Диагностика и устранение неисправностей

В настоящем разделе описаны средства мониторинга состояния шлюза, способы просмотра системных журналов, а также приведены типовые не-

исправности и рекомендации по их устранению. Регулярный контроль загрузки ресурсов и анализ логов помогают своевременно выявлять отклонения в работе и предотвращать сбои.

4.1. Системный мониторинг ресурсов шлюза

Веб-конфигуратор RealLab! предоставляет наглядные инструменты для оценки текущей загрузки центрального процессора (CPU), использования оперативной памяти (RAM), температуры устройства и нагрузки от отдельных процессов (включая CODESYS).

Примечание. Подробное описание работы веб-конфигуратора RealLab! приведено в руководстве пользователя по работе с ПО NLS-Gateway. Это руководство доступно для скачивания на странице шлюза на сайте компании RealLab.

4.1.1. Просмотр информации о системе

Перейдите в веб-конфигуратор (http://IP-адрес-шлюза:8000/), откройте страницу «Инфо». На ней содержится информация о нагрузке на ядра процессора шлюза, количество занятой и свободной ОЗУ и ПЗУ, а также температура процессора.

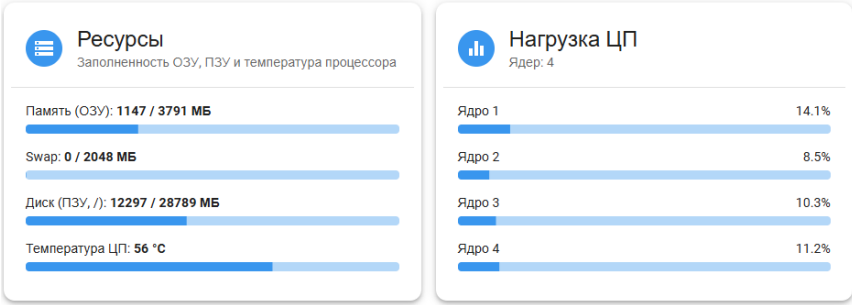


Рис. 4.1. Информация о нагрузке на шлюз

По умолчанию данные обновляются каждые 3 секунды (автообновление можно отключить).

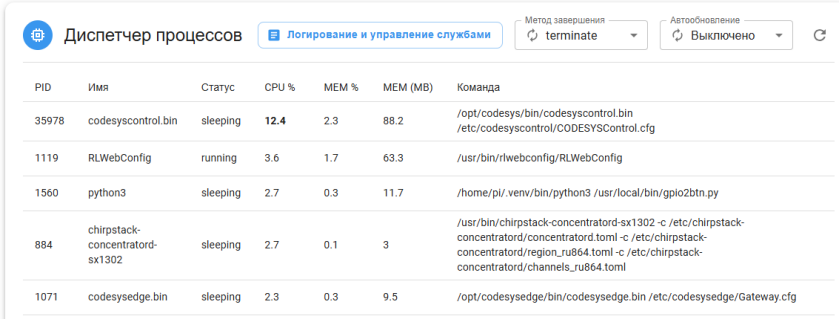
Эти показатели позволяют быстро оценить, не перегружен ли шлюз, и принять решение о перезапуске служб или оптимизации проектов (например, в CODESYS).

Также на странице «Инфо» содержатся общие сведения об операционной системе и сетевых интерфейсах:

- Наименование ОС:
- Версия ядра;
- Имя хоста;
- Идентификатор процессорного модуля;
- Время работы с последней перезагрузки;
- Состояние интерфейсов eth0, wlan0, bluetooth, can0 (при наличии).

4.1.2. Управление процессами

Страница «Процессы» выводит список всех запущенных процессов с их идентификаторами (PID), потреблением CPU и памяти. Процессы отсортированы по убыванию нагрузки на процессор, что облегчает поиск «тяжёлых» задач.



PID	Имя	Статус	CPU %	MEM %	MEM (MB)	Команда
35978	codesyscontrol.bin	sleeping	**12.4**	2.3	88.2	/opt/codesys/bin/codesyscontrol.bin /etc/codesyscontrol/CODESYSControl.cfg
1119	RLWebConfig	running	3.6	1.7	63.3	/usr/bin/rlwebconfig/RLWebConfig
1560	python3	sleeping	2.7	0.3	11.7	/home/pi/.venv/bin/python3 /usr/local/bin/gpio2btn.py
884	chirpstack-concentratord-sx1302	sleeping	2.7	0.1	3	/usr/bin/chirpstack-concentratord-sx1302 -c /etc/chirpstack-concentratord/concentratord.toml -c /etc/chirpstack-concentratord/region_ru864.toml -c /etc/chirpstack-concentratord/channels_ru864.toml
1071	codesysedge.bin	sleeping	2.3	0.3	9.5	/opt/codesysedge/bin/codesysedge.bin /etc/codesysedge/Gateway.cfg

Рис. 4.2. Диспетчер процессов в веб-конфигураторе на шлюзе

На этой странице доступны следующие действия:

– завершение процесса;

– включение автообновления списка с интервалом 1, 3, 5 или 10 секунд;

– просмотр логов и управление службами – для этого нажмите кнопку «Логирование и управление службами» в верхней части страницы и выберите нужную службу из списка.

Мониторинг процессов особенно полезен при диагностике зависаний или аномально высокой загрузки, вызванной, например, нештатной работой проекта CODESYS.

4.2. Просмотр системных журналов и диагностика ошибок

Большинство служб, установленных на шлюзе, ведут собственные журналы событий. Доступ к этим журналам организован единообразно через веб-конфигуратор RealLab!.

4.2.1. Общий подход к логированию

Для каждой службы в соответствующем разделе веб-конфигуратора предусмотрена кнопка «Логирование и управление службами» (или аналогичная). При её нажатии открывается диалоговое окно, в котором можно:

- выбрать конкретную службу из выпадающего списка;
- просмотреть логи (с возможностью ограничения количества строк от конца файла);
- скачать логи в файл;
- запустить, остановить, перезапустить службу;
- включить или отключить её автоматический запуск при загрузке шлюза.

Такой подход доступен для следующих служб:

- **ChirpStack** – логи концентратора, сетевого сервера и MQTT Forwarder;
- **MQTT-брокер Mosquitto** – логи брокера;
- **Zigbee2MQTT** – логи самого веб-интерфейса Z2M;
- **Node-RED** – журнал службы nodered.service;
- **CODESYS** – журнал службы CODESYS.

4.2.2. Рекомендации по анализу логов

При поиске неисправностей обращайте внимание на следующие маркеры:

- **Ошибки (ERROR)** – указывают на критические сбои, часто содержат код ошибки и описание причины;

- **Предупреждения (WARNING)** – могут сигнализировать о некорректных настройках или приближении к лимитам ресурсов;
- **Сообщения отладки (DEBUG)** – полезны для глубокого анализа, но их уровень должен быть включён в настройках соответствующей службы.

Если служба не запускается, сначала проверьте её статус через диалог управления службами, затем просмотрите логи – там обычно указывается конкретная причина отказа (например, неверный пароль в MQTT, отсутствие прав доступа к файлу, конфликт портов и т.д.).

Для удобства диагностики можно скачать логи и передать их технической поддержке RealLab!.

4.3. Типовые неисправности и методы их устранения

В табл. 4 приведены наиболее часто встречающиеся проблемы, их вероятные причины и рекомендуемые действия по восстановлению работоспособности.

Табл. 4. Типовые неисправности шлюзов NLS-Gateway и способы их устранения

Неисправность	Возможная причина	Действия по восстановлению
Служба ChirpStack / ChirpStack Gateway Bridge не запущена	Сбой после обновления, нехватка памяти, повреждение конфигурации	Через веб-конфигуратор RealLab! в разделе управления службами проверить статус и перезапустить службу. При необходимости просмотреть логи и исправить ошибки конфигурации.
Служба CODESYS не запущена или завершается с ошибкой	Ошибки в проекте, загруженном на шлюз; нехватка оперативной или постоянной памяти; нестабильное питание; сбой вызван подключенными устройствами;	Проверьте проект на наличие ошибок и предупреждений – неправильная работа с указателями может привести к аварийному завершению службы CODESYS. Также проверьте нагрузку на шлюз во время работы проекта. Используйте логи для выявления неисправностей.

Неисправность	Возможная причина	Действия по восстановлению
Нет связи с LoRaWAN-датчиками	Не совпадает Gateway ID в ChirpStack; неправильный частотный план; проблемы с радиомодулем	Проверить Gateway ID на вкладке LoRaWAN веб-конфигуратора и сравнить с идентификатором, добавленным в ChirpStack. Убедиться, что выбран правильный регион (частотный план) в параметрах концентратора. Проверить уровень сигнала и антенну.
Шлюз перезагружается/ зависает	Нестабильное питание, перегрев, ошибки файловой системы	Проверить температуру процессора (страница «Инфо»). Убедиться в исправности блока питания. Просмотреть системные логи через dmesg (по SSH) или через журналы служб. Выполнить перезагрузку.
Нет данных от Zigbee-устройств	Устройство не добавлено или потеряло связь; неправильно настроен MQTT-брокер; низкий уровень сигнала	Через веб-интерфейс Zigbee2MQTT убедиться, что устройство отображается в списке. Проверить качество связи (LQI) – при необходимости переместить шлюз или добавить маршрутизаторы. Проверить настройки подключения к MQTT.
Неверное время/дата в системе	Села батарейка RTC. Неверный часовой пояс	Проверить настройки часового пояса; Установить время вручную при необходимости через веб-интерфейс RealLab!
Конфликт IP-адресов	Два устройства в сети с одинаковым IP	Задать шлюзу статический IP через веб-интерфейс RealLab!

Лист регистрации изменений

Дата изменения	Описание изменения	Примечание