



ФОНД СОДЕЙСТВИЯ
ИННОВАЦИЯМ

Research Laboratory
of Design Automation
НИЛ автоматизации проектирования

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

Среда визуального программирования Node-RED

Программирование ПЛК RealLab в среде визуального программирования
Node-RED

Программирование ПЛК RealLab в среде визуального программирования NODE-RED

Руководство пользователя

© НИЛ АП, 2026

Версия 29 января 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.

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

Оглавление

1. Вводная часть	5
1.1. Общие сведения о среде Node-RED	5
1.2. Интеграция Node-RED с другими средами выполнения	6
1.2.1. Преимущества использования среды исполнения Node-RED с другими средами	7
1.3. Информация о ПЛК RealLab	8
1.3.1. Программируемые контроллеры ReaLab в Node-RED	8
1.3.2. Основные характеристики ПЛК	8
1.3.3. Интерфейсы ПЛК	9
1.4. Установка Node-RED на операционную систему RealLab! Raspbian Linux	10
1.4.1. Основные настройки при первом запуске	12
2. Начало работы	15
2.1. Первый запуск Node-RED	15
2.1.1. Доступ к редактору Node-RED. Интерфейс редактора потоков	15
2.1.2. Основные элементы интерфейса	16
2.2. Управление палитрой узлов	24
2.2.1. Установка пакетов в менеджере палитр	24
2.2.2. Просмотр списка установленных пакетов	28
2.3. Отладка в среде Node-RED	30
2.3.1. Установка и включение отладчика	30
2.4. Экспорт и импорт потоков	31
2.5. Управление выполнением потоков	34
2.5.1. Запуск и остановка потоков через среду разработки	34
2.5.2. Перезагрузка Runtime через среду разработки	36
3. Основной функционал Node-RED	37
3.1. Взаимодействие по Modbus RTU	37
3.1.1. Настройка Modbus RTU. Чтение данных с устройства	37

3.1.2. Настройка Modbus RTU. Запись данных в устройство	40
3.2. Взаимодействие по Modbus TCP.....	40
3.2.1. Настройка Modbus TCP. Чтение данных с устройства	40
3.2.2. Настройка Modbus TCP. Запись данных в устройство.....	42
3.3. Работа с протоколом MQTT	44
3.3.1. Подписка на топики (MQTT In)	44
3.3.2. Публикация сообщений (MQTT Out)	47
3.4. Работа с GPIO в Node-RED.....	48
3.5. Работа с OPC UA-клиентом.....	49
3.5.1. Информации о структуре тегов.....	50
3.5.2. Чтение одного тега	52
3.5.3. Чтение нескольких тегов	53
3.6. Работа с OPC UA-сервером	56
3.6.1. Настройка OPC-сервера	57
3.6.2. Настройка соединения с клиентом UaExpert	60
3.7. Визуализация данных с помощью Node-RED Dashboard	62
4. Возможности интеграции Node-RED	66
4.1. Интеграция с облачными платформами	66
4.2. Подключение к базам данных	67
4.3. Взаимодействие с REST API	67
4.4. Работа с мессенджерами	68
4.5. Поддержка IoT-протоколов	68
Лист регистрации изменений.....	69

1. Вводная часть

Настоящее руководство предназначено для пользователей, начинающих работу со средой визуального программирования Node-RED. В нём приведены пошаговые инструкции по установке, настройке и базовому использованию Node-RED в задачах промышленной автоматизации. Особое внимание уделено взаимодействию с программируемыми логическими контроллерами (ПЛК) семейства RealLab!, однако описанные принципы применимы и к другим ПЛК, работающим под управлением операционной системы *RealLab! Raspbian Linux*.

Руководство охватывает основные этапы работы: от установки необходимых компонентов и самого Node-RED до начальной конфигурации, управления потоками, подключения дополнительных пакетов и взаимодействия с популярными промышленными протоколами. Оно поможет пользователю быстро освоить базовые функции среды и приступить к разработке собственных автоматизированных решений.

ВАЖНО! Перед запуском любого пользовательского программного обеспечения на операционной системе ПЛК настоятельно рекомендуется провести тщательное тестирование, чтобы не нарушить основную функциональность контроллера. Мы не несём ответственности за последствия, вызванные использованием или запуском программного обеспечения, созданного пользователями самостоятельно.

1.1. Общие сведения о среде Node-RED

Node-RED – это платформа визуального программирования с открытым исходным кодом, разработанная для упрощения создания приложений интернета вещей (IoT) и систем автоматизации. Благодаря low-code подходу, Node-RED позволяет пользователям проектировать логику приложений путём графического соединения функциональных блоков, называемых узлами (nodes), без необходимости написания большого объёма программного кода.

Основные особенности Node-RED:

- Графический интерфейс – интуитивно понятная среда разработки, доступная через веб-браузер.
- Лёгкость интеграции – поддержка множества протоколов и сервисов (MQTT, HTTP, Modbus, OPC UA, Telegram, базы данных и др.).

- Модульная архитектура – возможность расширения функциональности за счёт установки дополнительных пакетов узлов из открытых репозитивов.
- Бесплатное использование и открытый исходный код – платформа распространяется по лицензии Apache 2.0 и активно развивается сообществом.

Node-RED состоит из двух ключевых компонентов:

1. **Редактор потоков (Flow Editor)** – веб-интерфейс, в котором пользователь создаёт и настраивает логику приложения, соединяя узлы в последовательности, называемые потоками (flows).
2. **Среда выполнения (Runtime)** – серверная часть, отвечающая за обработку и исполнение созданных потоков. В контексте данного руководства среда выполнения размещается непосредственно на промышленном контроллере RealLab! под управлением ОС RealLab! Raspbian Linux.

Типичные сценарии применения Node-RED:

- Автоматизация технологических процессов и промышленных систем;
- Сбор, обработка и визуализация данных с датчиков и ПЛК;
- Создание REST API и веб-хуков для интеграции с внешними системами;
- Управление устройствами умного дома или здания;
- Разработка человеко-машинных интерфейсов (HMI) и информационных панелей (Dashboard);
- Интеграция с мессенджерами (например, Telegram) для уведомлений и удалённого управления.

Благодаря своей гибкости и простоте освоения, Node-RED становится всё более популярным инструментом не только среди разработчиков, но и среди инженеров-автоматчиков, стремящихся быстро реализовать надёжные и масштабируемые решения для промышленной автоматизации.

1.2. Интеграция Node-RED с другими средами выполнения

При одновременном использовании сред выполнения Node-RED и CODESYS (MasterSCADA-4D, Полигон) на одном устройстве (например, ПЛК RealLab!) следует учитывать, что обе системы могут обращаться к одним и тем же аппаратным интерфейсам – таким как последовательные порты

(RS-485), GPIO или сетевые ресурсы. Если обе среды пытаются получить эксклюзивный доступ к одному и тому же физическому интерфейсу (например, Modbus RTU через `/dev/serial0`), это может привести к конфликту: одна из систем получит ошибку «порт занят» или данные будут повреждены из-за несинхронизированного доступа.

Рекомендуется:

- Назначать разные задачи каждой среде (например, CODESYS отвечает за управление, а Node-RED – за передачу данных по MQTT);
- Организовывать обмен данными между средами через внутренние механизмы (например, OPC UA, TCP-сокеты, файлы или разделяемую память), а не напрямую к одному и тому же драйверу устройства.

1.2.1. Преимущества использования среды исполнения Node-RED с другими средами

Использование Node-RED совместно с CODESYS (MasterSCADA-4D, Полигон) предоставляет значительные преимущества при построении современных систем автоматизации:

- **Гибкость интеграции** – Node-RED легко подключается к облачным платформам, мессенджерам (Telegram, WhatsApp), базам данных, REST API и IoT-протоколам (MQTT, HTTP), что выходит за рамки традиционных возможностей CODESYS.
- **Low-code подход** – визуальное программирование в Node-RED позволяет быстро создавать прототипы и реализовывать сложную логику обработки данных без глубокого знания языков программирования.
- **Разделение функций** – CODESYS остаётся ответственным за детерминированное, надёжное управление в реальном времени (ПИД-регуляторы, безопасность, логика оборудования), тогда как Node-RED берёт на себя задачи верхнего уровня: визуализацию, уведомления, архивацию, аналитику и удалённый доступ.
- **Быстрая адаптация к изменениям** – добавление нового сервиса (например, отправка SMS при аварии) в Node-RED реализуется в течение нескольких минут, тогда как в CODESYS (MasterSCADA-4D, Полигон) это потребовало бы перекомпиляции проекта и, возможно, изменения архитектуры.
- **Открытая экосистема** – Node-RED поддерживает обширную экосистему готовых узлов, разработанных сообществом разработчиков, что

значительно расширяет функциональность контроллера без необходимости писать код «с нуля».

Таким образом, сочетание детерминированной логики (CODESYS MasterSCADA-4D, Полигон) и гибкой интеграционной платформы Node-RED создаёт мощную гибридную архитектуру для современных промышленных и IoT-решений.

1.3. Информация о ПЛК RealLab

1.3.1. Программируемые контроллеры RealLab в Node-RED

Ниже приведён список контроллеров RealLab!, которые можно программировать в среде визуального программирования Node-RED:

- NLScon-RSB-(S, L-RS, L-CAN);
- NLScon-RSB-(XL-RS, XL-CAN);
- NLScon-RSB-(S, L-RS, L-CAN)-MS(V);
- NLScon-RSB-XL-CAN-RD
- NLScon-RSB-(S, XS)-PLG;
- NLScon-RSB-XS-PLG-RD;
- Панельные ПЛК NLcon-CED5-21;
- Панельные ПЛК NLcon-CED5-21-MSV;
- Панельные ПЛК NLcon-CED5-15-2CAN;

1.3.2. Основные характеристики ПЛК

- Процессор: Broadcom BCM2837B0 (4 ядра ARM Cortex-A53, тактовая частота 1.2 ГГц).
- Операционная система: RealLab! Raspbian Linux (включена в реестр российского программного обеспечения, № 29011 от 31.07.2025).
- Оперативная память (ОЗУ): 1 Гб.
- Системная память: 8 Гб eMMC (встроенная флеш-память).
- Интерфейсы: GPIO, UART, I²C, SPI, CAN, Ethernet, USB, HDMI.
- Питание: 10–30 В постоянного тока (диапазон может варьироваться в зависимости от модели).

- Рабочая температура: от -20°C до $+60^{\circ}\text{C}$.
- Крепление: DIN-рейка (для NLScon-RSB), панельное крепление (для NLcon-CED/LXD).
- Поддерживаемые промышленные протоколы:
 - Modbus RTU / TCP;
 - CANopen (опционально);
 - MQTT, OPC UA (через ПО).
- Особенности ПО:
 - Поддержка разработки на C++, Python, Qt5.9, PyQt;
 - Интеграция с системами SCADA;
 - Встроенное ПО включает конфигураторы для первоначальной настройки ПЛК и подключённых модулей ввода-вывода. Конфигураторы поддерживают настройку по следующим промышленным протоколам: Modbus TCP / RTU и CANopen.

1.3.3. Интерфейсы ПЛК

На ПЛК серии NLScon-RSB предусмотрены следующие интерфейсы:

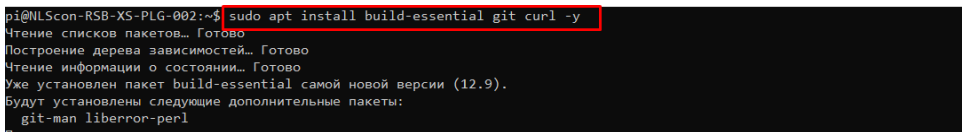
- 2 порта Ethernet — Eth0, Eth1;
- 2 интерфейса CAN — в операционной системе обозначаются как CAN0, CAN1;
- 2 порта RS485 — в операционной системе обозначаются как /dev/serial0, /dev/serial1;
- RTC (Real Time Clock) — модуль аппаратных часов реального времени;
- SD-карта — слот для карты памяти;
- Дискретные входы и выходы (Din/Dout):
 - Dout 1, 2 — подключены к GPIO 12, 13;
 - Din 1, 2 — подключены к GPIO 3, 2;
- Светодиоды индикации «Старт/Стоп» — подключены к GPIO 0, 1;
- Светодиоды индикации «Состояния» — подключены к GPIO 8, 9; доступны только в серии ПЛК NLScon-RSB-XL;

- Переключатель «Старт/Стоп» — подключены к GPIO 10, 11.
- На панельных ПЛК серии NLcon-CED и NLcon-LXD не предусмотрены дискретные входы и выходы, светодиоды индикации, а также переключатели «Старт/Стоп».

1.4. Установка Node-RED на операционную систему RealLab! Raspbian Linux

Для установки необходимых пакетов работы среды исполнения Node-RED выполните следующую команду:

```
sudo apt install build-essential git curl -y
```



```
pi@NLcon-RSB-XS-PLG-002:~$ sudo apt install build-essential git curl -y
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Уже установлен пакет build-essential самой новой версии (12.9).
Будут установлены следующие дополнительные пакеты:
  git-man liberror-perl
Получение пакетов...
```

Рис. 1.1. Установка необходимых пакетов для работы Node-RED

Далее установите Node.js, npm и Node-RED.

Выполните официальный установочный скрипт Node-RED, запустив в терминале следующую команду:

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

Данный скрипт выполнит следующие действия:

- Остановит текущий работающий Node-RED, чтобы избежать конфликтов при обновлении;
- Удалит предыдущие файлы и пакеты Node-RED и Node.js для установки более новой стабильной версии;
- Установит актуальную версию Node.js, необходимую для работы Node-RED;
- Очистит кэш менеджера пакетов npm, чтобы предотвратить ошибки при установке;
- Переместит глобально установленные ноды (например, через `npm install -g`) в локальную директорию проекта Node-RED (`~/node-red`), что рекомендуется для лучшей управляемости и совместимости;

- Пересоберет все установленные пользовательские ноды, чтобы обеспечить их совместимость с новой версией Node.js;
- Установит последнюю версию Node-RED через npm;
- Настроит Node-RED как системную службу;
- Добавит удобные команды: node-red-start, node-red-stop и др.;
- Обновит конфигурацию системного сервиса (systemd) для автоматического запуска Node-RED при загрузке системы.

```
Running Node-RED install for user pi at /home/pi on raspbian

This can take 20-30 minutes on the slower Pi versions - please wait.

Stop Node-RED
Remove old version of Node-RED
Remove old version of Node.js
Install Node.js
Clean npm cache
Install Node-RED core
Move global nodes to local
Npm rebuild existing nodes
Install extra Pi nodes
Add shortcut commands
Update systemd script

Any errors will be logged to /var/log/nodered-install.log
```

Рис. 1.2. Процесс установки или обновления Node-RED на ОС RealLab!
Raspbian Linux

Если установка завершится с ошибкой, то все ошибки и сообщения будут записаны в файл `/var/log/nodered-install.log`.

```
Running Node-RED install for user pi at /home/pi on raspbian

This can take 20-30 minutes on the slower Pi versions - please wait.

Stop Node-RED
Remove old version of Node-RED
Remove old version of Node.js
Install Node 20.19.1-1nodesource1 v20.19.1 Npm 10.8.2
Clean npm cache
Install Node-RED core
Move global nodes to local
Npm rebuild existing nodes
Install extra Pi nodes
Add shortcut commands
Update systemd script
```

Рис. 1.3. Процесс успешной установки

1.4.1. Основные настройки при первом запуске

После завершения установки Node-RED с помощью официального скрипта рекомендуется выполнить начальную конфигурацию. Это позволит обеспечить безопасность, удобство работы и управляемость проектов.

Настройка безопасности пользователя (User Security)

Система запросит включение базовой аутентификации.

Рекомендуется выбрать «**Yes**», чтобы защитить веб-редактор Node-RED от несанкционированного доступа. При этом будет предложено задать имя пользователя и пароль для входа в интерфейс редактора.

```
User Security
=====you keep up to date with the latest features and fixes.
? Do you want to setup user security? ... ck to the Node-RED team.
> Yeses not include any details of your flows or users.
  Nofull information on what information is collected and how it is used,
```

Рис. 1.4. Процесс настройки безопасности пользователя

Телеметрия

Node-RED предлагает отправлять анонимную статистику об использовании платформы разработчикам.

Выберите:

- **Yes, send my usage data** – если согласны участвовать в улучшении продукта;
- **No** – если не хотите передавать какие-либо данные.

Этот выбор не влияет на функциональность системы.

```
Projects
=====ee https://nodered.org/docs/telemetry
The Projects feature allows you to version control your flow using a local git repository.
> Yes, send my usage data
? Do you want to enable the Projects feature? ...
  Yes
> No
```

Рис. 1.5. Процесс выбора телеметрии

Включение функции "Проекты" (Projects)

Функция Projects интегрирует встроенный Git-репозиторий для управления версиями потоков (flows).

Рекомендуется выбрать «**Yes**», особенно при работе в команде или при необходимости резервного копирования логики. Это позволяет отслеживать изменения, откатываться к предыдущим версиям и организовывать совместную разработку.

Выбор рабочего процесса проекта (Project Workflow)

Если включена функция Projects, система предложит выбрать режим фиксации изменений:

- **manual** – пользователь вручную создаёт коммиты;
- **auto** – изменения автоматически сохраняются в Git после каждого развертывания (Deploy).

Рекомендуется выбрать auto, так как это упрощает работу и снижает риск потери изменений.

```
? What project workflow do you want to use? ...
  manual - you must manually commit changes
  auto - changes are automatically committed
```

Рис. 1.6. Процесс выбора рабочего процесса проекта

Выбор темы интерфейса (Theme)

На последнем этапе отображается список доступных тем оформления редактора. Используйте клавиши стрелок вверх/вниз для навигации и нажмите Enter для подтверждения выбора.

По умолчанию выбрана **default**. Для комфортной работы в условиях низкой освещённости рекомендуется использовать тёмные темы: dark, night-owl, noctis и др.

```
ory. ...
  default
  aurora
  cobalt2
  dark
  dracula
  espresso-libre
  github-dark
  github-dark-def
```

Рис. 1.7. Темы интерфейса (Theme) в Node-RED

Запуск и управление Node-RED

После завершения настройки Node-RED запустится автоматически.

Табл. 1. Основные команды в Node-RED

Действие	Команда
Запустить	node-red-start
Остановить	node-red-stop
Перезапустить	node-red-restart
Просмотреть логи	sudo node-red-log

Автозапуск при загрузке системы

Чтобы Node-RED запускался автоматически после перезагрузки ПЛК, выполните:

```
sudo systemctl enable nodered.service
```

```
sudo systemctl start nodered.service
```

Для отключения автозапуска:

```
sudo systemctl disable nodered.service
```

Логирование Node-RED0

Журнал событий (логи) Node-RED можно просматривать в реальном времени с помощью команды:

```
sudo node-red-log
```

```
pi@NLScon-RSB-XS-PLG-002:~$ sudo node-red-log
1 Dec 13:30:16 - [info] Linux 6.1.21-rt7-v7 arm LE
1 Dec 13:30:18 - [info] Загрузка узлов палитры
1 Dec 13:30:24 - [info] Файл настроек      : /home/pi/.node-red/settings.js
1 Dec 13:30:24 - [info] Хранилище контекста : 'default' [module=memory]
1 Dec 13:30:24 - [info] Каталог пользователя : /home/pi/.node-red
1 Dec 13:30:24 - [info] Projects directory: /home/pi/.node-red/projects
1 Dec 13:30:24 - [warn] Нет активного проекта : используется файл потоков по умолчанию
1 Dec 13:30:24 - [info] Файл потоков        : /home/pi/.node-red/flows.json
1 Dec 13:30:24 - [info] Создание нового файла flow
1 Dec 13:30:24 - [warn]
-----
Ваш файл учетных данных потока зашифрован с использованием
сгенерированного системой ключа.
Если сгенерированный системой ключ по какой-либо причине утерян, файл
с Вашими учетными данными не подлежит восстановлению. Вам придется
удалить его и повторно ввести свои учетные данные.
Вы должны установить свой собственный ключ, используя опцию
'credentialSecret' в Вашем файле настроек. После этого Node-RED
повторно зашифрует Ваш файл учетных данных, используя выбранный Вами
ключ, при следующем развертывании изменений.
-----
1 Dec 13:30:24 - [info] Сервер теперь работает на http://127.0.0.1:1880/
1 Dec 13:30:24 - [warn] Encrypted credentials not found
1 Dec 13:30:24 - [info] Запуск потоков
1 Dec 13:30:24 - [info] Запущены потоки
```

Рис. 1.8. Логирование в Node-RED

Доступ к редактору Node-RED

Когда Node-RED запущен, откройте его в браузере:

- На самом ПЛК: <http://localhost:1880/>;
- С другого компьютера в сети: http://<IP_адрес_Pi>:1880.

2. Начало работы

2.1. Первый запуск Node-RED

2.1.1. Доступ к редактору Node-RED. Интерфейс редактора потоков

Когда Node-RED запущен на ПЛК, откройте веб-браузер и перейдите по адресу: http://<IP-адрес_ПЛК>:1880.

Войдите, используя имя пользователя и пароль, если они были заданы ранее.

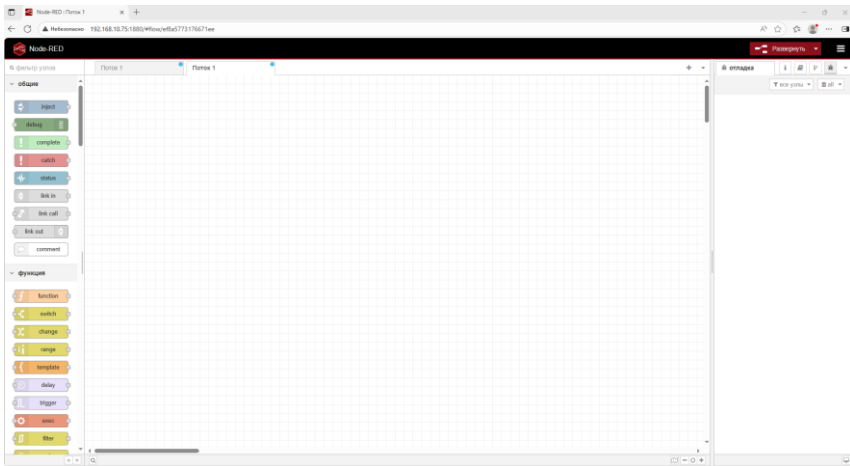


Рис. 2.1. Редактор Node-RED

Редактор Node-RED (рис. 2.1) предоставляет графический интерфейс, который позволяет пользователям легко создавать, настраивать и управлять потоками. Пользователи Node-RED визуально перетаскивают «узлы» из палитры, представляющие конечные точки, а затем соединяют их вместе в потоки для выполнения желаемой задачи.

2.1.2. Основные элементы интерфейса

Рассмотрим основные элементы интерфейса (см. рис. 2.2рис. 2.2):

- Узлы (Nodes);
- Поток (Flow);
- Развертывание (Deploy);
- Боковая панель.

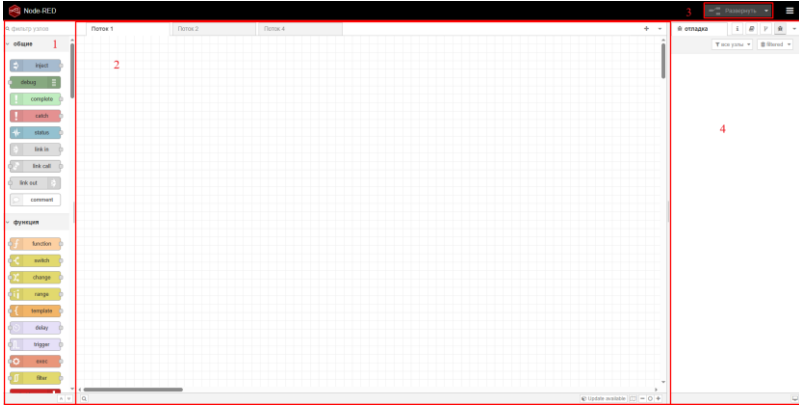


Рис. 2.2. Основные элементы интерфейса

1. Узлы (Nodes) – представляют собой боковую панель, содержащую все установленные и доступные для использования узлы.

Расположенная в верхней части палитры строка поиска позволяет быстро находить узлы по их названию.

Палитра разделена на несколько категорий (общие, функции и т.д.), каждая из которых содержит коллекции узлов. Когда устанавливается сторонний узел, он может создать новую категорию. Вы можете свернуть или развернуть категории, нажав на конкретную категорию.

В нижней части палитры вы найдете два значка со стрелками. Нажатие на первую иконку со стрелкой вверх сворачивает все категории, нажатие на значок со стрелкой вниз по умолчанию развернет все категории обратно.

При нажатии на кнопку посередине боковая панель палитры скрывается. Чтобы отобразить ее снова, нажмите на кнопку еще раз (см. рис. 2.3).

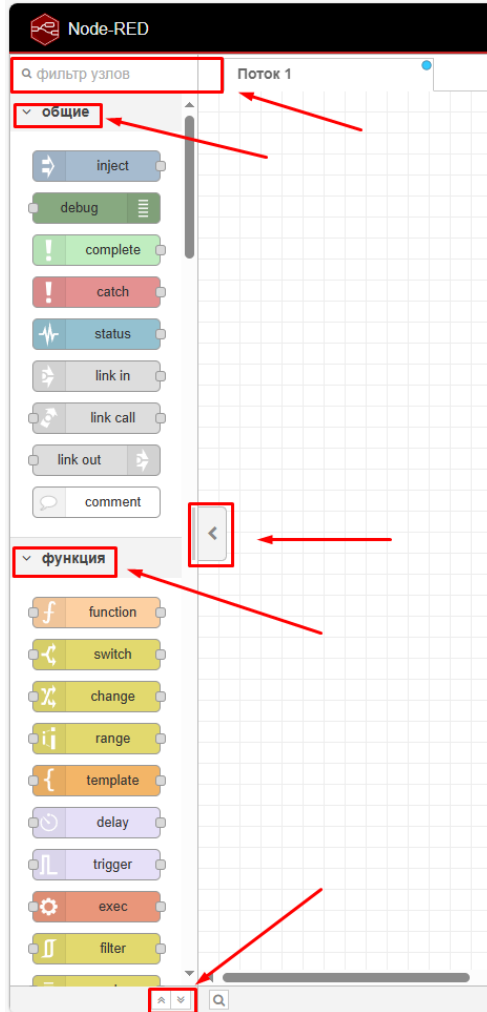


Рис. 2.3. Элементы управления узлами

2. Поток (Flow) – рабочая область, где потоки (группы узлов) развиваются путем перетаскивания узлов из палитры и связывания их вместе (см. рис. 2.4). При добавлении новой вкладки потока создается новое рабочее пространство.

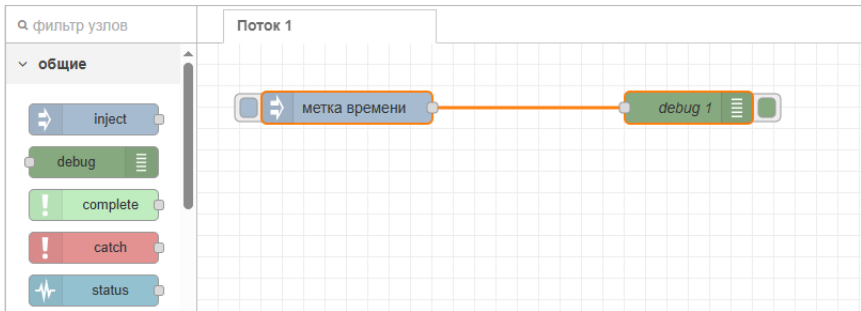


Рис. 2.4. Рабочая область (поток)

Чтобы создать родительский поток, нажмите на значок «+» в правом верхнем углу (см. рис. 2.5), в противном случае вы можете использовать опцию добавления потока в главном меню (см. рис. 2.6).

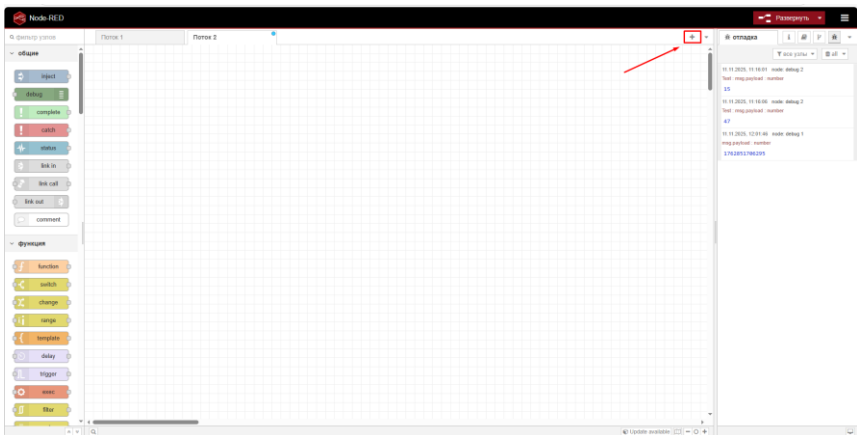


Рис. 2.5. Создание потока

2. Начало работы

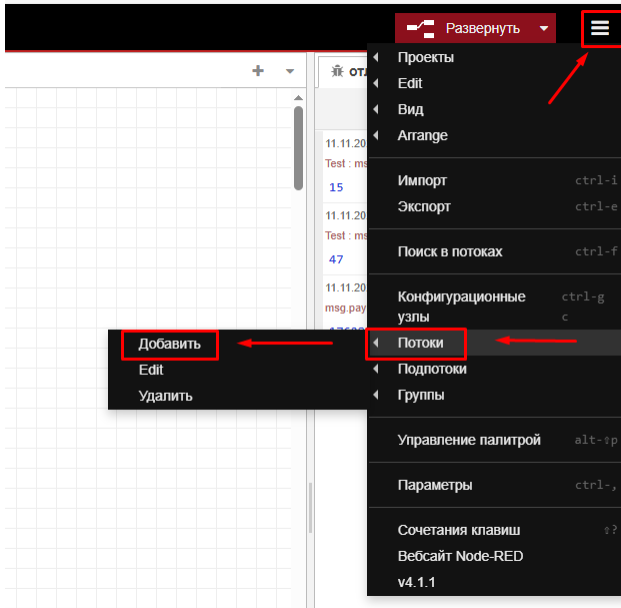


Рис. 2.6. Создание потока через главное меню

Чтобы включить или отключить поток, дважды щелкните вкладку потока. Нажмите внизу на кнопку «Отключить» или «Включить», если она уже отключена (см. рис. 2.7).

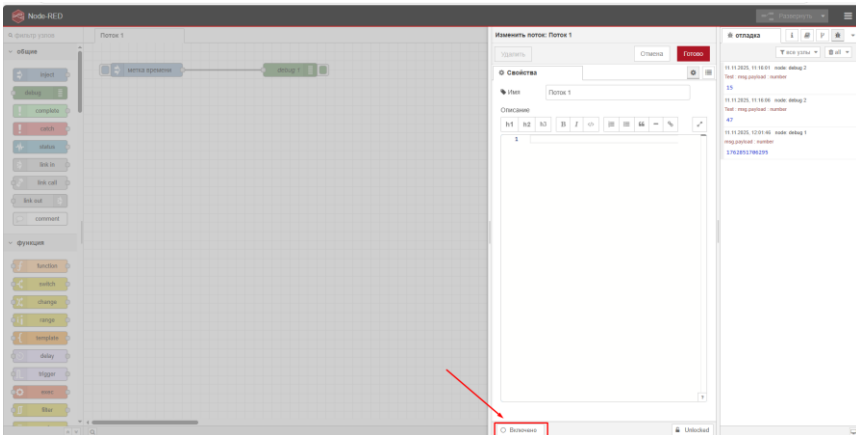


Рис. 2.7. Включение/отключение потока

Подпоток в Node-RED – это набор узлов, которые свернуты в отдельный узел в рабочей области. Чтобы создать подпоток используется опция добавления подпотока в главное меню (см. рис. 2.8). Созданный подпоток можно использовать как обычный узел в потоках.

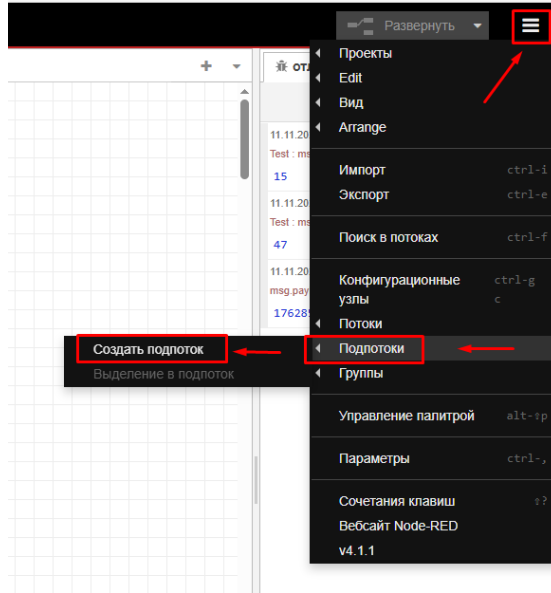


Рис. 2.8. Создание подпотока через главное меню

3. Развертывание (Deploy) – в верхней правой части расположена красная кнопка развертывания. При нажатии на нее выполняются все потоки внутри экземпляра. Красный цвет указывает на ожидающие изменения для развертывания, а серый – на отсутствие ожидающих изменений (см. рис. 2.9).

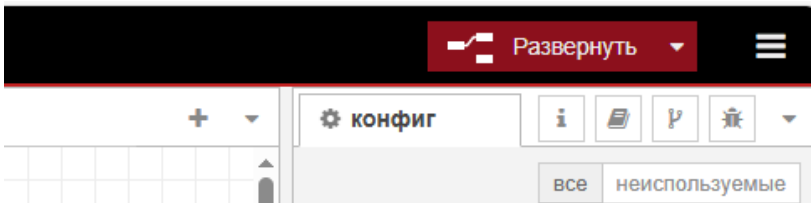


Рис. 2.9. Развертывание потоков

Чтобы развернуть только измененные потоки, или узлы, или остановить/перезапустить выполнение потока, нажмите значок развертывания на кнопке

развертывания. Выберите предпочтительный вариант и нажмите соответствующую кнопку развертывания (см. рис. 2.10).

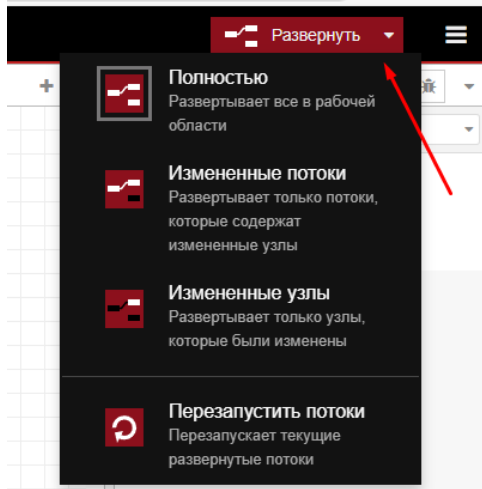


Рис. 2.10. Выбор варианта развертывания потоков

4. Боковая панель – предоставляет дополнительные контекстно-зависимые параметры и информацию в зависимости от выбранного узла или рабочей области. Инструменты, доступные на боковой панели, называются панелями (см. рис. 2.11).

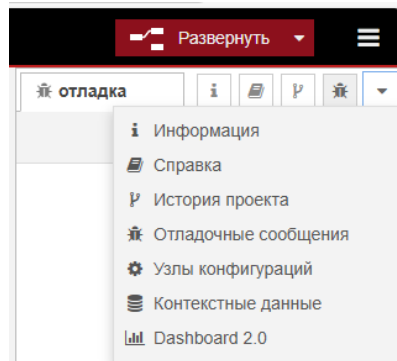


Рис. 2.11. Боковая панель

Боковая панель Node-RED по умолчанию включает в себя следующие панели:

Информация (Information) – показывает информацию обо всех потоках, присутствующих в них узлах, подпотоках, а также узлах глобальной конфигурации в древовидной структуре.

Свойства элемента – включает в себя (см. рис. 2.12):

- Справку, клик по ней приведет к переходу на панель помощи и покажет предоставленный автором файл readme для выбранного элемента;
- Копирование URL выбранного элемента, вы можете использовать эту функцию, когда хотите указать или обсудить конкретный элемент в рабочем пространстве с членом вашей команды;
- Позволяет найти этот элемент в рабочей области.



Рис. 2.12. Свойства элемента

Справка (Help) – представляет собой древовидную структуру и позволяет просматривать документацию.

Отладка (Debug) – отображает сообщения, выводимые узлом отладки, эта панель помогает легко отладить ваше приложение. Включает в себя опцию фильтра, позволяющую отображать информацию по:

- Всем узлам - при выборе этой опции будут отображаться сообщения, напечатанные всеми узлами;
- Выбранным узлам - позволяет выбрать конкретные узлы отладки, сообщения которых вы хотите видеть на панели отладки;
- Текущему потоку - будут печататься только сообщения, напечатанные узлами, которые присутствуют в текущем потоке.

ВАЖНО! Боковая панель "Отладка" может отображать только 100 самых последних сообщений. Если на боковой панели в данный момент

отображается отфильтрованный список сообщений, скрытые сообщения по-прежнему учитываются в лимите в 100 сообщений.

Контекстные данные – на этой панели отображаются контекстные переменные по их области. Каждая переменная включает в себя такие сведения, как дата и время, ее хранилище (в памяти или постоянное), а также имя и значение.

Узлы конфигурации (Config nodes) – значок шестеренки показывает информацию об узлах конфигурации. Узлы конфигурации не отображаются в основном рабочем пространстве, и это специальные узлы, которые содержат повторно используемые конфигурации, которые могут совместно использоваться несколькими узлами в потоке.

2.2. Управление палитрой узлов

2.2.1. Установка пакетов в менеджере палитр

Узлы Node-RED устанавливаются через Менеджер палитр – инструмент, позволяющий пользователям управлять узлами, доступными для использования в Node-RED. Менеджер палитр показывает все установленные узлы и позволяет пользователям обновлять и удалять их при необходимости. Например, для подключения устройств Modbus сначала потребуется установить необходимый пакет узлов. Для этого необходимо:

1. Откройте редактор FlowFuse Node-RED.
2. Нажмите на меню в правом верхнем углу и выберите «Управление палитрой» (см. рис. 2.13).

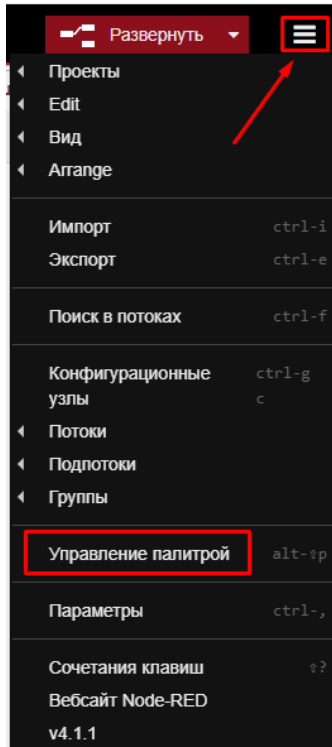


Рис. 2.13. Раздел управление палитрой в главном меню

3. Перейдите на вкладку «Установить» и выполните поиск по запросу (см. рис. 2.14):

node-red-contrib-modbus

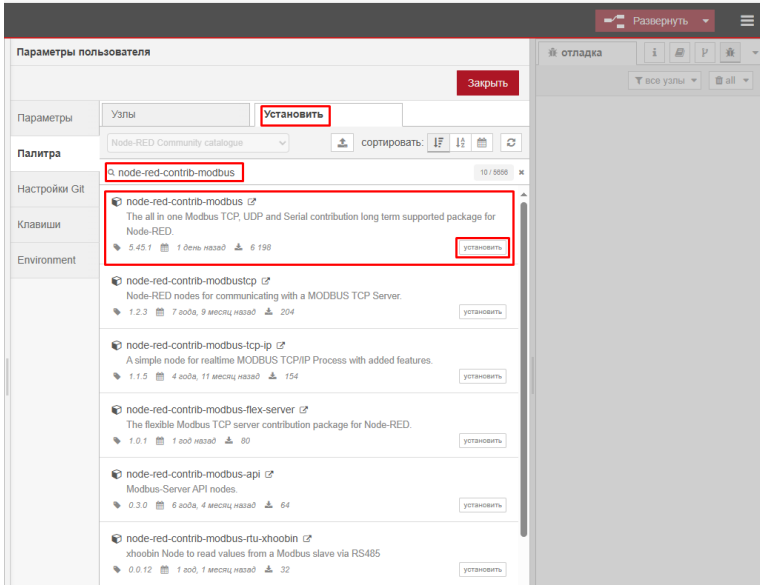


Рис. 2.14. Установка узлов для работы с Modbus

4. Нажмите на кнопку «Установить» во всплывающем окне (см. рис. 2.15).

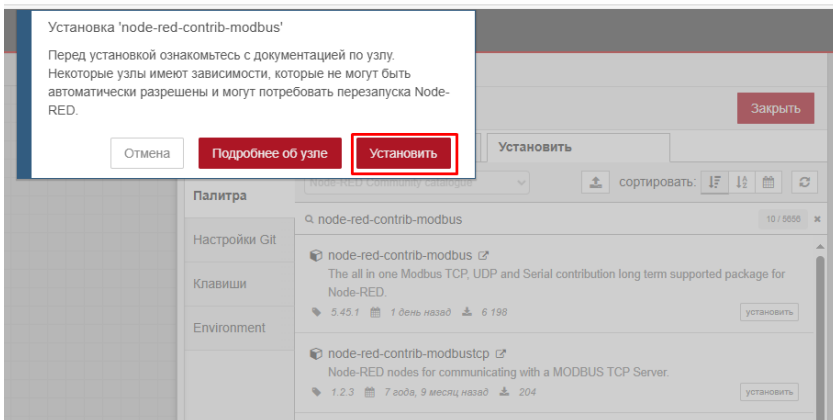


Рис. 2.15. Установка узлов для работы с Modbus

После установки в дереве отобразятся новые узлы (см. рис. 2.16).

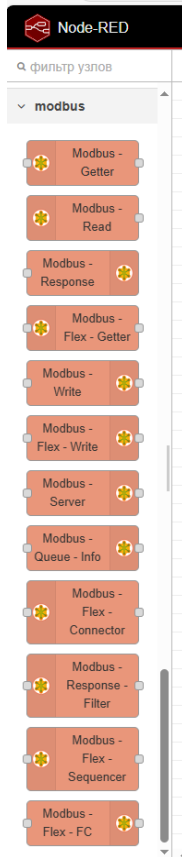


Рис. 2.16. Отображение установленных узлов для работы с Modbus в дереве

Аналогичным образом устанавливаются пакеты узлов, необходимые для работы проекта.

Табл. 2. Основные пакеты узлов

Пакет узлов	Описание
node-red-contrib-opcu	Для работы с OPC UA
@flowfuse/node-red-dashboard	Для создания дашборда для визуализации данных

2. Начало работы

node-red-debugger	Отладчик Flow Debugger
node-red-contrib-persist	Позволяет сохранять содержимое сообщений в невосприимчивом хранилище, которое можно восстановить после перезагрузки Node-RED
node-red-node-snmp	Для работы с протоколом SNMP

2.2.2. Просмотр списка установленных пакетов

Чтобы посмотреть список установленных пакетов необходимо ПКМ кликнуть на рабочей области, в открывающемся списке выбрать пункт **Show action list** (см. рис. 2.17).

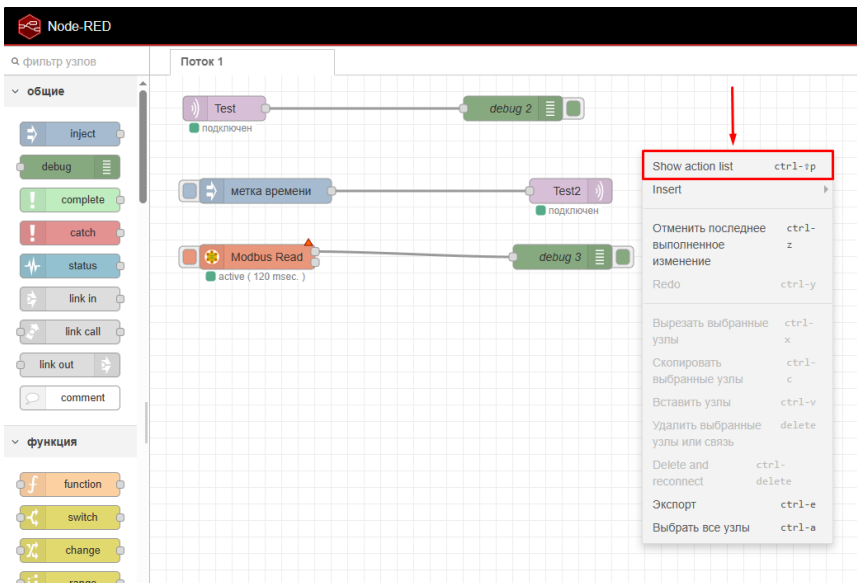


Рис. 2.17. Открытие списка Show action list

Далее в поисковой строке необходимо ввести **System Info** (см. рис. 2.18).

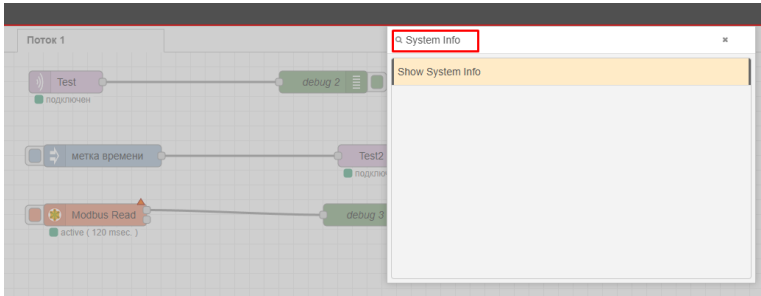


Рис. 2.18. Открытие списка Show action list

В открывшемся окне перейти на вкладку Visual. Найти «modules», в котором будет перечень всех установленных пакетов (см. рис. 2.19).

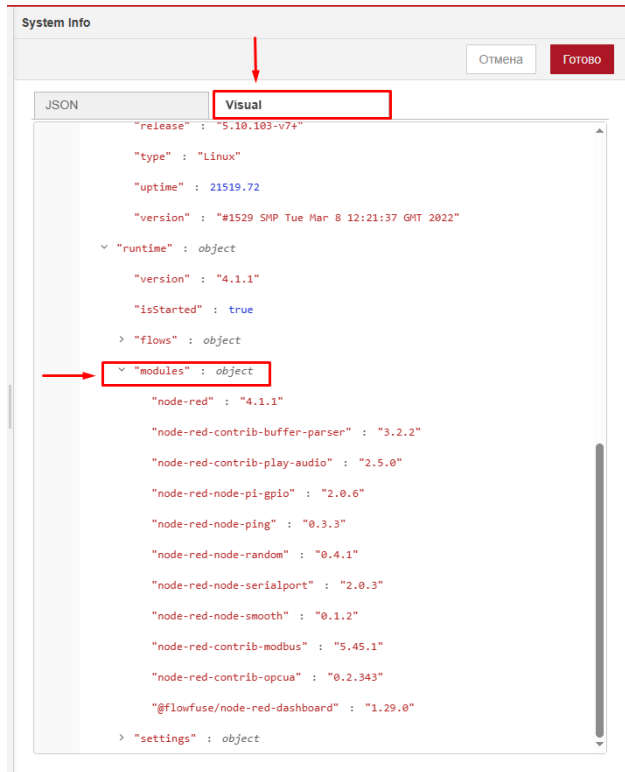


Рис. 2.19. Отображение перечня всех установленных пакетов

2.3. Отладка в среде Node-RED

2.3.1. Установка и включение отладчика

Чтобы установить отладчик Node-RED нажмите на значок меню в правом верхнем углу, выберите «Управление палитрой» и перейдите на вкладку «Установить». Найдите **node-red-debugger**.

Нажмите «Установить», чтобы добавить пакет (см. рис. 2.20).

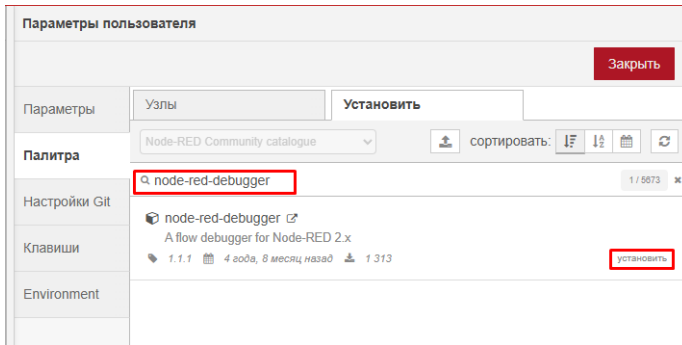


Рис. 2.20. Установка отладчика

После установки откройте вкладку «Отладка (debugger)» на боковой панели, щелкнув значок сворачиваемой стрелки на правой боковой панели и выберите «Отладчик потока» (Flow Debugger) (см. рис. 2.21).

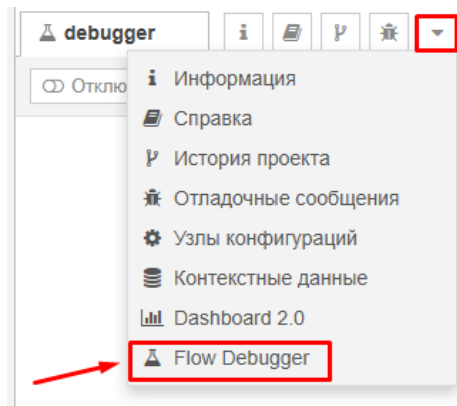


Рис. 2.21. Открытие «Отладчик потока»

На новой вкладке «Отладчик» переключите переключатель в верхнем левом углу боковой панели, чтобы включить отладчик. По умолчанию он отключен, поэтому включите его, прежде чем продолжить (см. рис. 2.22).

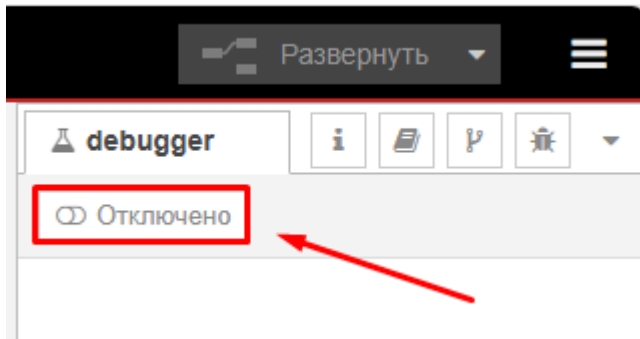


Рис. 2.22. Включение «Отладчик потока»

2.4. Экспорт и импорт потоков

Потоки в Node-RED можно экспортировать, импортировать и группировать (делиться потоками с другими пользователями или импортировать потоки, созданные другими). Группировка потоков позволяет организовывать связанные потоки для лучшего управления и совместного использования. Эта функция облегчает совместную работу и обмен проектами Node-RED.

Чтобы сохранить свои узлы, нажмите на меню в правом верхнем углу и выберите опцию «Экспорт» (см. рис. 2.23).

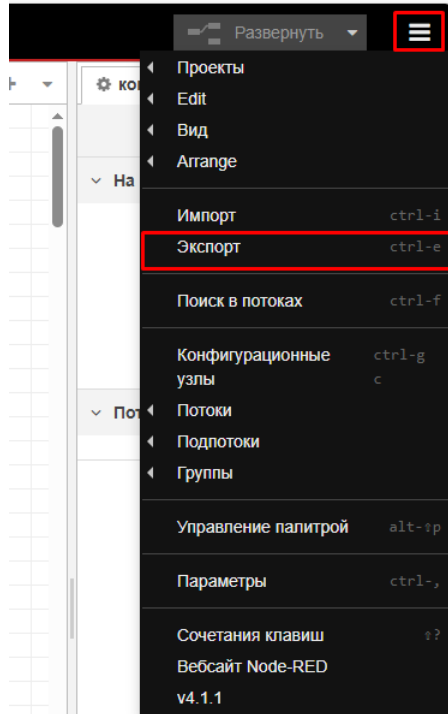


Рис. 2.23. Открытие «Экспорт узлов»

Открывается новое окно, где можно выбрать сохранить выбранные узлы, текущий поток или все потоки (см. рис. 2.24). Также можно загрузить узлы как файл JSON или скопировать JSON в буфер обмена.

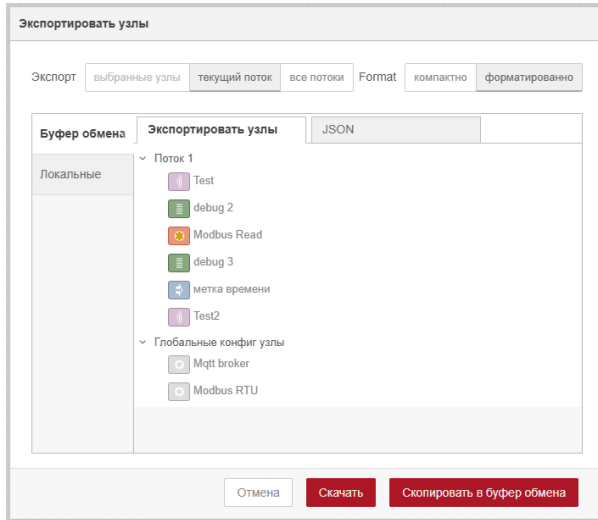


Рис. 2.24. Экспортировать узлы

В окне «Импорт узлов» (**Import nodes**) (см. рис. 2.25) вы можете загрузить файл JSON или вставить необработанный JSON (см. рис. 2.26).

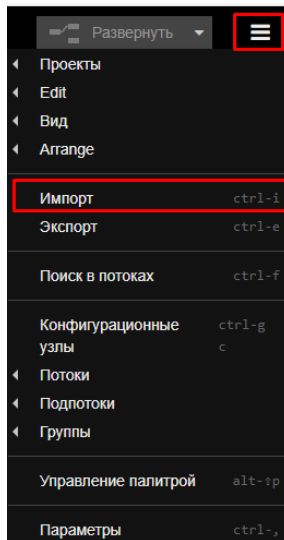


Рис. 2.25. Открытие «Импорт узлов»

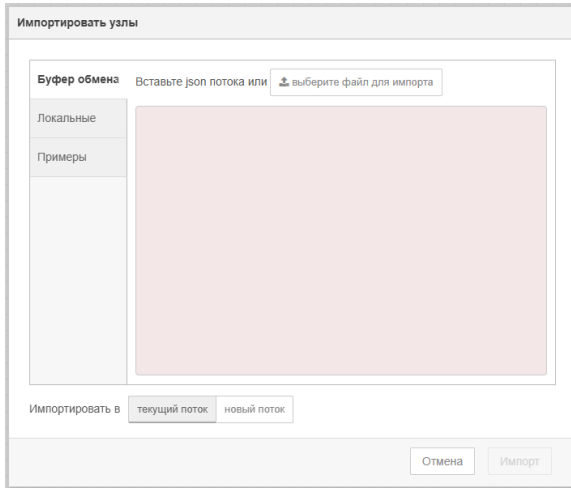


Рис. 2.26. Импортировать узлы

2.5. Управление выполнением потоков

2.5.1. Запуск и остановка потоков через среду разработки

Для добавления опции Start/Stop в выпадающем меню кнопки Deploy (Развернуть) необходимо изменить настройки **false** на **true** в разделе **runtimeState**.

Для этого выполните следующие действия:

1. Подключитесь к ПЛК либо напрямую, либо удалённо по протоколу SSH.
2. Откройте терминал и с помощью текстового редактора nano отредактируйте файл:

```
$sudo nano /home/pi/.node-red/settings.js
```
3. Найдите в файле параметр `runtimeState` и установите его значение в `true`.
4. Сохраните изменения (в nano — нажмите Ctrl+O, затем Enter) и выйдите из редактора (Ctrl+X).
5. Перезапустите среду выполнения Node-RED или перезагрузите ПЛК, чтобы изменения вступили в силу.

Чтобы убедиться, что настройка применена, выполните следующее:

1. Щёлкните правой кнопкой мыши на рабочей области **Node-RED**.
2. В открывшемся контекстном меню выберите пункт **Show action list**.
3. В поисковой строке введите **Show System Info** и откройте соответствующий элемент.
4. Перейдите на вкладку **Visual**, где в разделе **runtimeState** вы сможете увидеть текущее состояние (см. рис. 2.27).

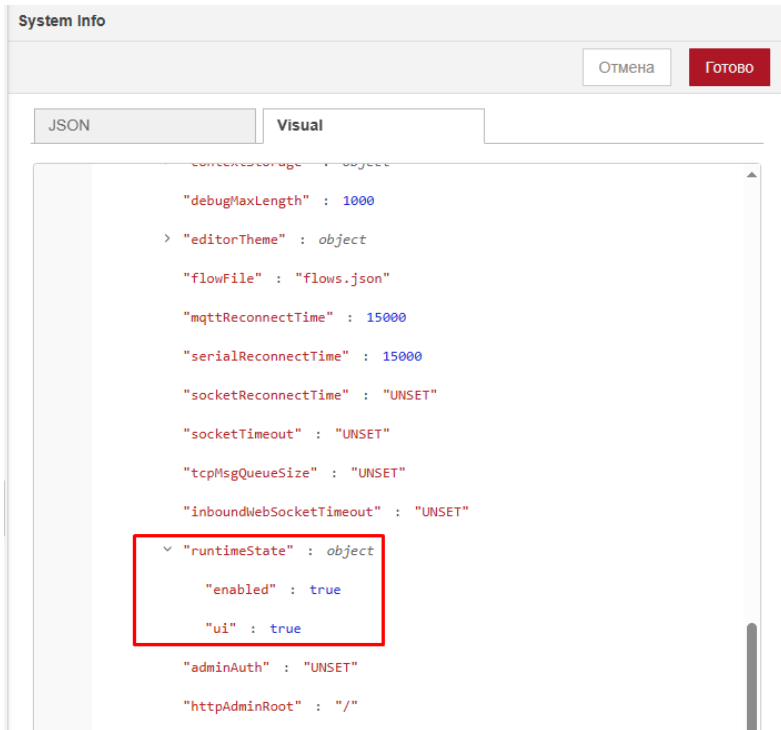


Рис. 2.27. Информации о настройках runtimeState

В результате станет доступна опции Start/Stop в выпадающем меню кнопки Deploy (Развернуть) (см. рис. 2.28).

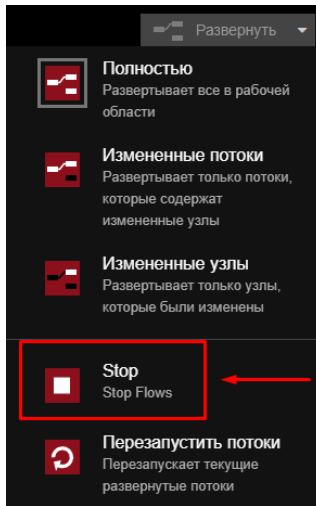


Рис. 2.28. Выбор варианта развертывания потоков

2.5.2. Перезагрузка Runtime через среду разработки

Перезагрузку runtime в Node-RED через среду разработки можно реализовать с помощью специальных пакетов узлов, которые позволяют запускать команды перезагрузки при определенных условиях (или по расписанию) и сохранять данные после перезагрузки Node-RED.

Пакет **node-red-contrib-rpi-shutdown** содержит узлы «Shutdown RPi» и «Restart RPi», которые запускают команду перезагрузки при любом условии или по расписанию (см. рис. 2.29).

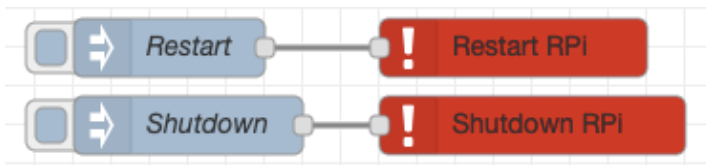


Рис. 2.29. Узлы для перезагрузки Node-RED

Перезапустить узел: если этот узел работает, Raspberry Pi будет перезагружен.

3. Основной функционал Node-RED

При разработке в Node-RED важное значение имеет эффективная связь между устройствами. Это взаимодействие облегчается различными протоколами, такими как MQTT, HTTP, CoAP и WebSockets.

Node-RED позволяет настраивать соединение по Modbus, MQTT, OPC UA, https.

3.1. Взаимодействие по Modbus RTU

3.1.1. Настройка Modbus RTU. Чтение данных с устройства

Modbus RTU является одним из наиболее широко используемых протоколов связи в промышленной автоматизации. Он позволяет считывать данные с датчиков, контролировать состояние оборудования и управлять устройствами с помощью простой архитектуры «ведущий-ведомый».

Добавьте узел Modbus – Read (чтение данных с ведомого устройства) на рабочую область. Подключите его к узлу debug (см. рис. 3.1).

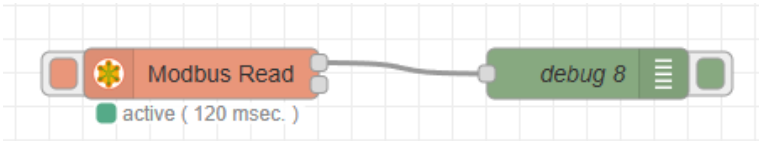


Рис. 3.1. Чтение данных с устройства по Modbus RTU

Дважды кликните мышкой по узлу Modbus – Read, чтобы открыть окно настроек (см. рис. 3.2). Введите Unit ID, выберите тип данных, которые вы хотите прочитать, укажите начальный адрес, установите количество значений для чтения и определите частоту опроса (как часто данные должны быть прочитаны).

Изменить узел Modbus-Read

Удалить Отмена Готово

Свойства

Settings Optionals

Имя Name

Тема Topic

Unit-Id 3

FC FC 4: Read Input Registers

Address 0

Quantity 16

Poll Rate 120 millisecond(s)

Delay to activate input

Allow oversized messages

Server Modbus RTU

Рис. 3.2. Окно настроек узла Modbus RTU – Read

Тип данных может принимать следующие значения:

FC 1: Read Coil Status – чтение DO.

FC 2: Read Input Status – чтение DI.

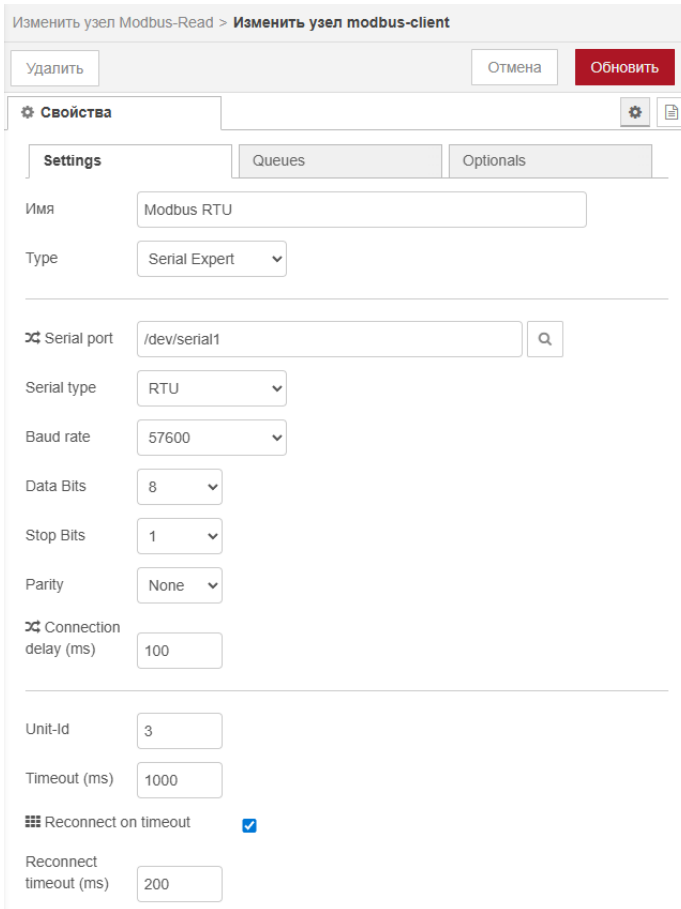
FC 3: Read Holding Registers – чтение АО.

FC 4: Read Input Registers – чтение AI.

1. Нажмите на значок + рядом с полем «Сервер», чтобы добавить сведения о подключении к Modbus (см. рис. 3.3);
2. В окне настройки сервера: задайте значение для параметра Type.;
3. В поле Serial port выберите порт, к которому подключено устройство Modbus в формате /dev/serial1;

3. Основной функционал Node-RED

4. Настройте параметры связи в соответствии с вашим устройством. Эти значения должны точно совпадать с конфигурацией устройства Modbus, в противном случае связь не будет установлена.



Изменить узел Modbus-Read > Изменить узел modbus-client

Удалить Отмена Обновить

Свойства

Settings Queues Optionals

Имя Modbus RTU

Type Serial Expert

Serial port /dev/serial1

Serial type RTU

Baud rate 57600

Data Bits 8

Stop Bits 1

Parity None

Connection delay (ms) 100

Unit-Id 3

Timeout (ms) 1000

Reconnect on timeout

Reconnect timeout (ms) 200

Рис. 3.3. Настройки сервера Modbus RTU – Read

5. Нажмите «Развернуть поток» (Deploy). Если все настроено правильно, вы начнете видеть данные с устройства Modbus в режиме реального времени на боковой панели отладки.

3.1.2. Настройка Modbus RTU. Запись данных в устройство

Modbus RTU позволяет записывать данные обратно на устройства.

1. Добавьте узел Modbus Write на рабочую область. Этот узел будет отправлять данные на ваше устройство Modbus.
2. В окне настроек введите: Unit ID устройства, выберите тип данных, которые вы хотите записать, укажите начальный адрес (см. рис. 3.4).
3. Нажмите на значок + рядом с полем «Сервер», чтобы добавить сведения о подключении к Modbus RTU.

The screenshot shows the configuration interface for a Modbus Write node. At the top, there are three buttons: 'Удалить' (Delete), 'Отмена' (Cancel), and 'Готово' (Done). Below this is a 'Свойства' (Properties) section with two tabs: 'Settings' and 'Optionals'. The 'Settings' tab is selected and contains the following fields:

- 'Name': A text input field.
- 'Unit-Id': A text input field containing the value '2'.
- 'FC': A dropdown menu showing 'FC 5: Force Single Coil'.
- 'Address': A text input field containing the value '1'.
- 'Delay to activate input': A checkbox that is currently unchecked.
- 'Server': A dropdown menu showing 'Modbus RTU', with edit and add icons to its right.

Рис. 3.4. Окно настроек узла Modbus RTU – Write

4. Подключите узел Inject к узлу Modbus Write для отправки значений. Нажмите «Развернуть поток» (Deploy).

3.2. Взаимодействие по Modbus TCP

3.2.1. Настройка Modbus TCP. Чтение данных с устройства

1. Добавьте узел Modbus – Read на рабочую область. Подключите его к узлу debug (см. рис. 3.5).

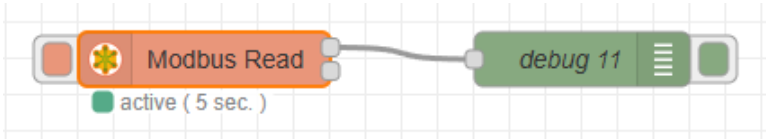


Рис. 3.5. Чтение данных с устройства по Modbus TCP

2. Дважды кликните мышкой по узлу Modbus – Read, чтобы открыть окно настроек (см. рис. 3.6).
3. Введите Unit ID, выберите тип данных, которые вы хотите прочитать, укажите начальный адрес, установите количество значений для чтения и определите частоту опроса (как часто данные должны быть прочитаны).

Рис. 3.6. Окно настроек узла Modbus TCP – Read

4. Нажмите на значок + рядом с полем «Сервер», чтобы добавить сведения о подключении к Modbus TCP.
5. В окне настройки сервера задайте: IP-адрес и номер порта (порт 502 - является портом по умолчанию для Modbus), тип TCP и идентификатор устройства (см. рис. 3.7).

3. Основной функционал Node-RED

Изменить узел Modbus-Read > Изменить узел modbus-client

Удалить Отмена **ОБНОВИТЬ**

⚙️ Свойства ⚙️ 📄

Settings Queues Optionals

Имя

Тип

Host

Port

TCP Type

Unit-Id

Timeout (ms)

🔌 Reconnect on timeout

Reconnect timeout (ms)

Рис. 3.7. Настройки сервера Modbus TCP – Read

6. Нажмите «Развернуть поток» (Deploy). Вы начнете видеть данные с устройства Modbus в режиме реального времени на боковой панели отладки (см. рис. 3.8).

```
20.11.2025, 16:41:12 node: debug 8
Test3 : msg.payload : array[8]
  ▶ [ false, true, false, false, false,
    true, false, false ]
```

Рис. 3.8. Вывод данных, считанных с устройства

3.2.2. Настройка Modbus TCP. Запись данных в устройство

1. Добавьте узел Modbus Write на рабочую область. Этот узел будет отправлять данные на ваше устройство Modbus.

3. Основной функционал Node-RED

2. Подключите узлы Inject к узлу Modbus Write для отправки значений true и false (см. рис. 3.9).

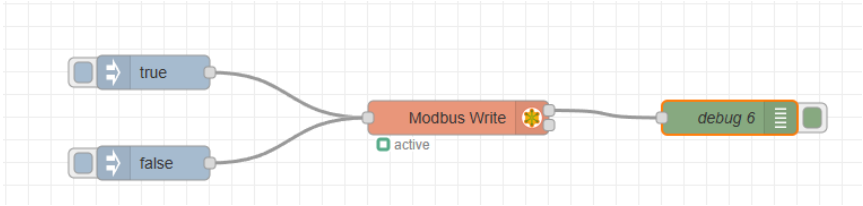


Рис. 3.9. Запись данных на устройство по Modbus TCP

3. Дважды кликните мышкой по узлу Modbus Write, чтобы открыть окно настроек (см. рис. 3.10).
4. Введите Unit ID, выберите тип данных, которые вы хотите записать, укажите начальный адрес.

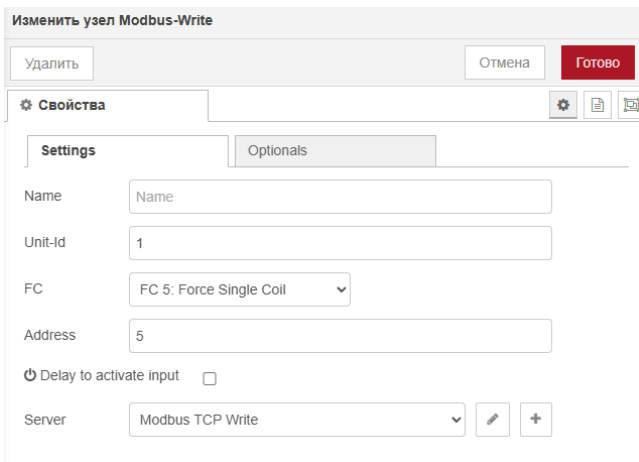
Тип данных может принимать следующие значения:

FC 5: Force Single Coil – запись одного DO.

FC 6: Preset Single Register – запись одного АО.

FC 15: Force Multiple Coils – запись нескольких DO.

FC 16: Force Multiple Registers – запись нескольких АО.

A screenshot of the configuration window for a 'Modbus Write' node in Node-RED. The window title is 'Изменить узел Modbus-Write'. At the top, there are buttons for 'Удалить', 'Отмена', and 'Готово'. Below is a 'Свойства' (Properties) section with a 'Settings' tab selected. The 'Settings' section contains the following fields:

- Name: A text input field with the value 'Name'.
- Unit-Id: A text input field with the value '1'.
- FC: A dropdown menu with 'FC 5: Force Single Coil' selected.
- Address: A text input field with the value '5'.
- Delay to activate input: A checkbox that is currently unchecked.
- Server: A dropdown menu with 'Modbus TCP Write' selected, and edit and add buttons.

Рис. 3.10. Окно настроек узла Modbus TCP – Write

3. Основной функционал Node-RED

5. Нажмите на значок + рядом с полем «Сервер», чтобы добавить сведения о подключении к Modbus TCP.
6. Настройте узлы Inject для отправки значений true и false (см. рис. 3.11).

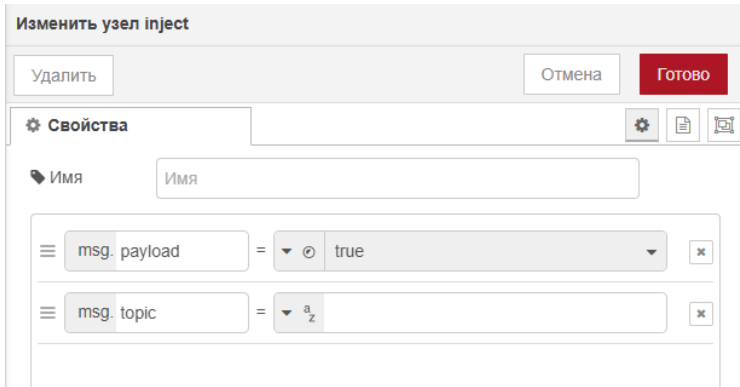


Рис. 3.11. Окно настроек узла Inject

7. Нажмите «Развернуть поток» (Deploy). Произойдет запись данных на устройство Modbus (см. рис. 3.12).

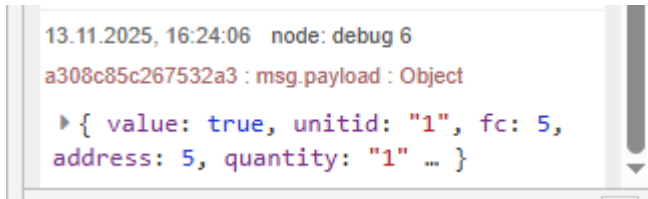


Рис. 3.12. Запись данных на устройство Modbus

3.3. Работа с протоколом MQTT

3.3.1. Подписка на топики (MQTT In)

1. Для приема сообщений из брокера используется узел MQTT In. Добавьте узел debug (для отображения получаемого узлом сообщения). Подключите его к узлу MQTT In (см. рис. 3.13).

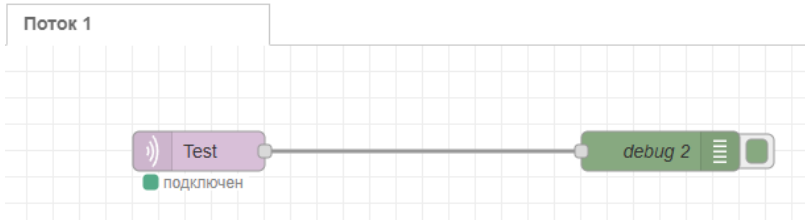


Рис. 3.13. Прием сообщений из брокера

2. Дважды кликните мышкой по узлу MQTT In, чтобы открыть окно настроек (см. рис. 3.14).
3. Задайте сервер, вид и тему подписки, значения параметров QoS и выхода (см. табл. 3).

The screenshot shows the configuration window for the 'mqtt in' node. The title is 'Изменить узел mqtt in'. At the top, there are buttons for 'Удалить' (Delete), 'Отмена' (Cancel), and 'Готово' (Done). Below the buttons is a section titled 'Свойства' (Properties) with a gear icon, a document icon, and a refresh icon. The properties are listed as follows:

- Сервер** (Server): A dropdown menu showing 'Mqtt broker' with edit and add icons.
- Action**: A dropdown menu showing 'Subscribe to single topic'.
- Тема** (Topic): A text input field containing 'Test'.
- QoS**: A dropdown menu showing '2'.
- Выход** (Output): A dropdown menu showing 'автоопределение (разобрать объект JSON, строка ил' (auto-detection (parse JSON object, string or...)).
- Имя** (Name): A text input field containing 'Имя'.

Рис. 3.14. Окно настроек узла MQTT In

Табл. 3. Свойства настроек узла MQTT In

Свойства	Описание
Сервер	Адрес брокера MQTT
Action (Вид подписки)	Подписаться на один топик. Динамическая подписка.
Topic (Тема)	Тема MQTT на которую необходимо подписаться.

3. Основной функционал Node-RED

QoS (Quality of Service) (Качество обслуживания)	Параметр QoS может принимать значения 0, 1 или 2. QoS 0 показывает, что сообщение должно быть доставлено не более одного раза (ноль или один раз). QoS 1 показывает, что сообщение должно быть доставлено не менее одного раза (один или более раз). QoS 2 показывает, что сообщение должно быть доставлено только один раз.
Выход	Может принимать значения: - автоопределение (разобрать объект JSON, строка или буфер); - автоопределение (строка или буфер); - буфер; - строка; - объект JSON; - строка в кодировке Base64.

4. Для узла **MQTT In** необходимо настроить соединение с брокером MQTT, которое настраивается нажатием на значок карандаша (см. рис. 3.15).

Стандартный порт MQTT-брокера для входящих TCP-соединений – 1883. При использовании защищенного SSL-подключения используется порт 8883.

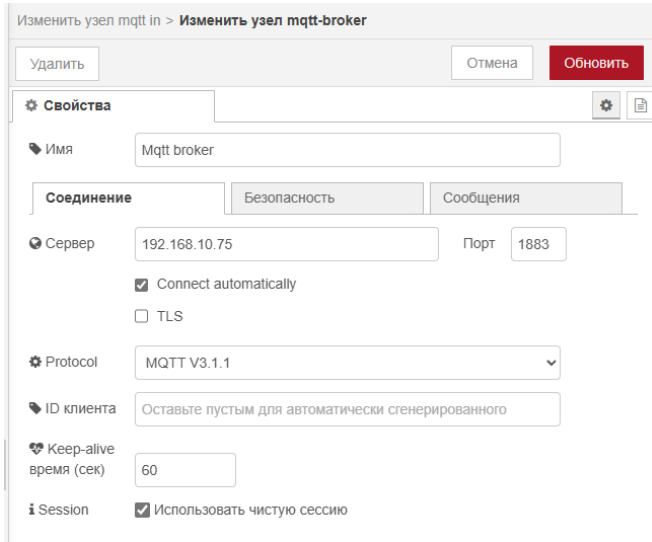


Рис. 3.15. Настройка соединения с брокером MQTT

6. Разверните поток «Развернуть» (Deploy). Узел MQTT In автоматически переподключается в случае потери соединения с брокером.

3.3.2. Публикация сообщений (MQTT Out)

Для публикаций сообщений в MQTT брокер используется узел MQTT Out.

1. Добавьте узел inject (для отправки текстового сообщения). Подключите его к узлу MQTT Out (см. рис. 3.16).



Рис. 3.16. Публикация сообщений в MQTT брокер

2. Для узла MQTT Out необходимо настроить соединение с брокером MQTT, задать тему опубликованного сообщения, указать параметр QoS, установить значение хранить – «да», если нужно, чтобы последнее сообщение сохранялось в брокере (см. рис. 3.17).

3. Нажмите «Развернуть поток» (Deploy).

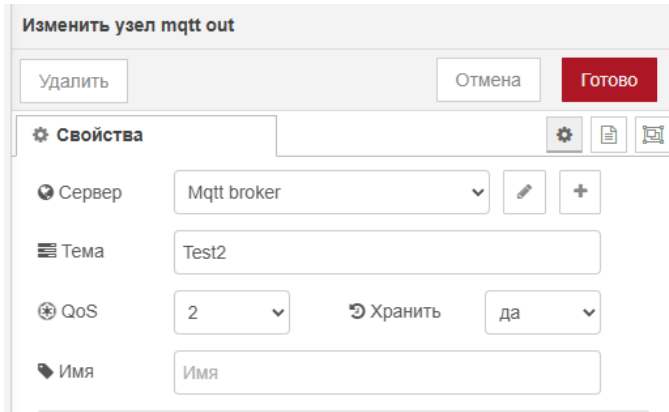


Рис. 3.17. Окно настроек узла MQTT Out

Примечание: ПЛК RealLab, работающие под управлением операционной системы RealLab! Raspbian Linux, могут выполнять роль MQTT-брокера. Подробную информацию см. в [руководстве по ссылке](#):

3.4. Работа с GPIO в Node-RED

Для работы с GPIO ПЛК RealLab! в Node-RED с использованием библиотеки Python RPi.GPIO, входящей в состав Raspbian, потребуется установить необходимый пакет узлов: **node-red-node-pi-gpio**

После установки появятся следующие узлы (см. рис. 3.18):



Рис. 3.18. Узлы для работы с GPIO

Пример работы с дискретными входами и выходами (см. рис. 3.19).

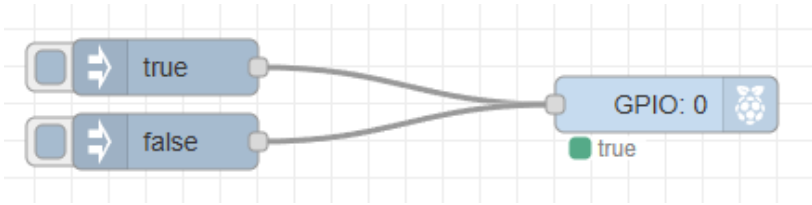


Рис. 3.19. Работа с узлом RPI – GPIO OUT

3.5. Работа с OPC UA-клиентом

Рассмотрим интеграцию OPC UA с использованием Node-RED. Это позволит считывать значения в реальном времени с ПЛК, датчиков и промышленных устройств.

Чтобы работать с OPC UA в Node-RED, сначала нужно установить необходимые узлы. Для установки пакета узлов необходимо в меню в правом верхнем углу выбрать «Управление палитрой», перейти на вкладку «Установить» и выполнить поиск по запросу:

node-red-contrib-opcua

В качестве OPC UA сервера будем использовать MasterOPC Universal Modbus Server (см. рис. 3.20).

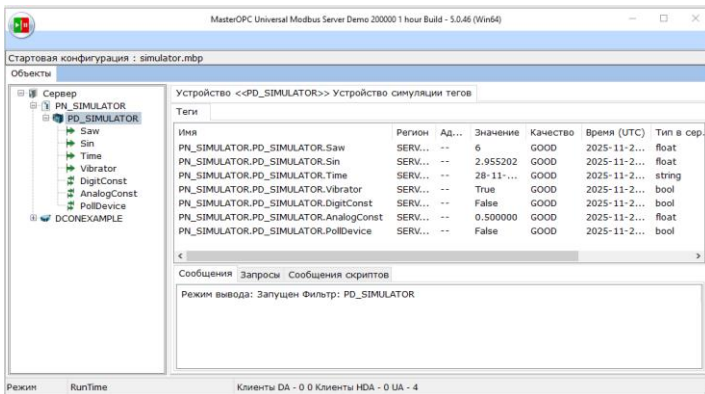


Рис. 3.20. MasterOPC Universal Modbus Server

3.5.1. Информация о структуре тегов

Если вы ещё не знаете идентификаторы узлов тегов, к которым хотите получить доступ, используйте узел OPC UA Browser для изучения структуры тегов.

1. Перетащите узлы: Inject, OPC UA Browser и Debug.
2. Подключите выход узла Inject ко входу узла браузера (OPC UA Browser), затем подключите выход браузера к узлу отладки (см. рис. 3.21).

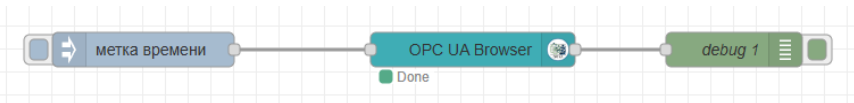


Рис. 3.21. Изучение структуры тегов при помощи узла OPC UA Browser

3. Дважды кликните мышкой по узлу OPC UA Browser, чтобы открыть окно настроек (см. рис. 3.22).

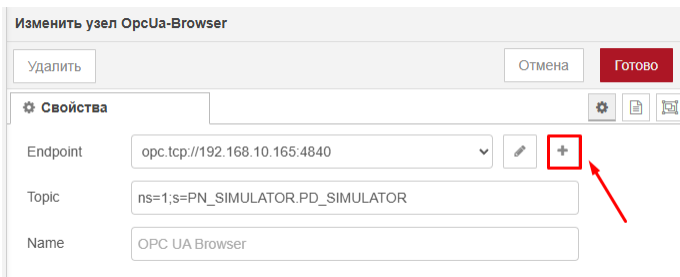


Рис. 3.22. Окно настроек узла OPC UA Browser

4. В узле Браузер нажмите на значок +, чтобы создать новую конфигурацию конечной точки. Введите адрес вашего сервера OPC UA, например: `opc.tcp://192.168.10.165:4840`.
5. Установите режим безопасности на None (можно добавить защиту позже) (см. рис. 3.23).

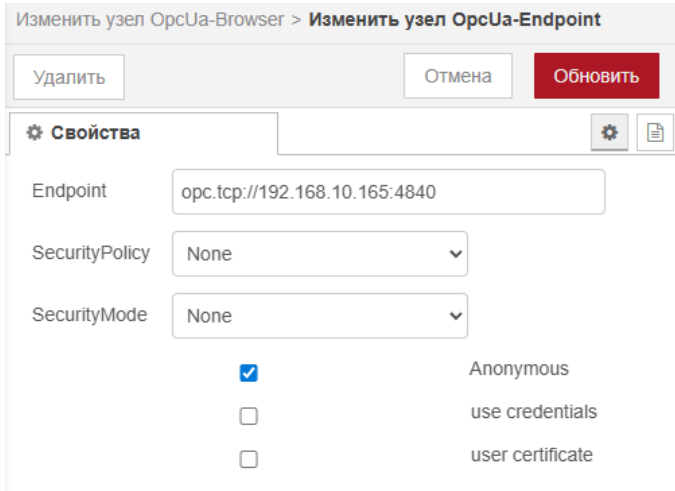


Рис. 3.23. Настройка конфигурации конечной точки

6. В узле Браузер установите тему на (корневую папку Objects) в формате: `ns=1;s=PN_SIMULATOR.PD_SIMULATOR`.
7. Настройте узел Inject для отправки временной метки. Нажмите «Развернуть поток» (Deploy) и кликните по узлу Inject.
8. На боковой панели отладки отобразится информация о структуре тегов (см. рис. 3.24). Теперь вы можете определить точные идентификаторы узлов для чтения или записи.

```
28.11.2025, 15:36:11 node: debug 1
msg.payload: array[7]
▼ array[7]
▼ 0: object
  ▼ item: object
    referenceTypeId: "ns=0;i=46"
    isForward: true
    nodeId:
      "ns=1;s=PN_SIMULATOR.PD_SIMULATOR
      .Saw"
    ▶ browseName: object
    ▶ displayName: object
    nodeClass: 2
    typeDefinition: "ns=0;i=63"
    value: 6
    dataType: "Float"
  ▶ 1: object
  ▶ 2: object
  ▶ 3: object
  ▶ 4: object
  ▶ 5: object
  ▶ 6: object
```

Рис. 3.24. Вывод данных о структуре тегов

Когда вы узнаете идентификаторы узлов, можно начинать читать данные с промышленного оборудования через сервер OPC UA.

3.5.2. Чтение одного тега

1. Перетащите узлы: Inject, OPC UA-Item, OPC UA Client и Debug на рабочую область (см. рис. 3.25).



Рис. 3.25. Чтение одного тега

2. В окне настроек узла OPC UA-Item задайте: идентификатор узла (Item) в формате ns=1;s=PN_SIMULATOR.PD_SIMULATOR.Saw и тип данных (см. рис. 3.26). Информация об идентификаторе узла отображается в поле nodeId (см. рис. 3.24).

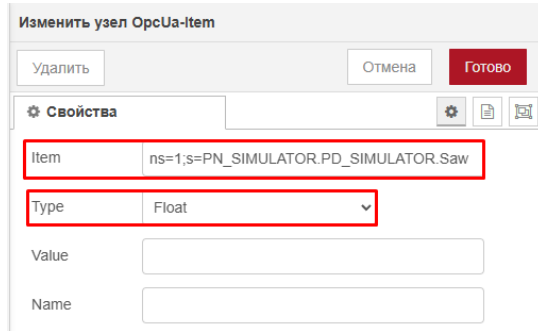


Рис. 3.26. Окно настроек узла OPC UA-Item

3. В окне настроек узла OPC UA Client задайте адрес вашего сервера OPC UA и установите действие на READ (см. рис. 3.27).

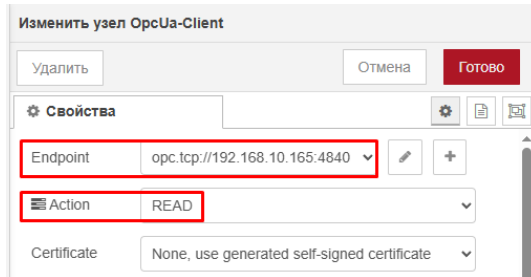


Рис. 3.27. Окно настроек узла OPC UA Client

4. Нажмите «Развернуть поток» (Deploy) и кликните по узлу Inject для запуска операции чтения тега. На боковой панели отладки отобразится значение тега (см. рис. 3.28).

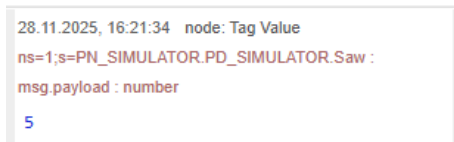


Рис. 3.28. Вывод информации о значении тега

3.5.3. Чтение нескольких тегов

1. Для реализации пакетного считывания данных (см. рис. 3.29) в окне настроек узла OPC UA Client установите действие на READ MULTIPLE (см. рис. 3.30).

3. Основной функционал Node-RED

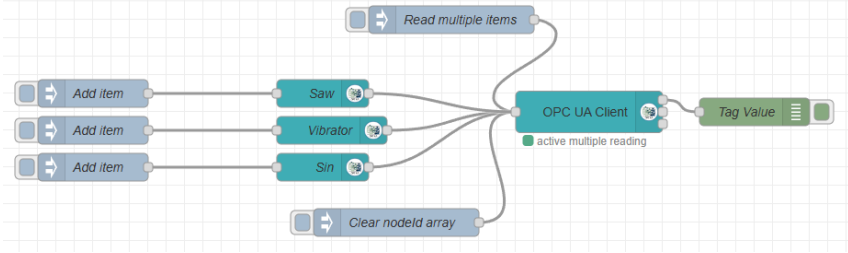


Рис. 3.29. Чтение нескольких тега

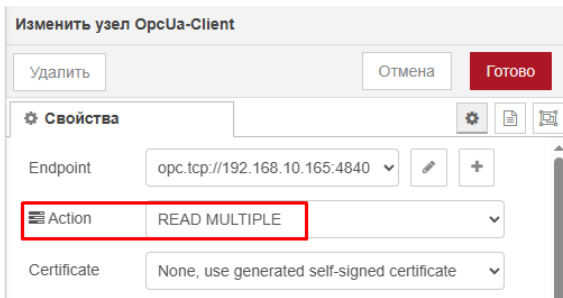


Рис. 3.30. Окно настроек узла OPC UA Client

2. Добавьте узлы Inject и OPC UA Item для каждого тега, который хотите прочитать. Настройте узлы Inject для отправки значений типа строка (см. рис. 3.31).

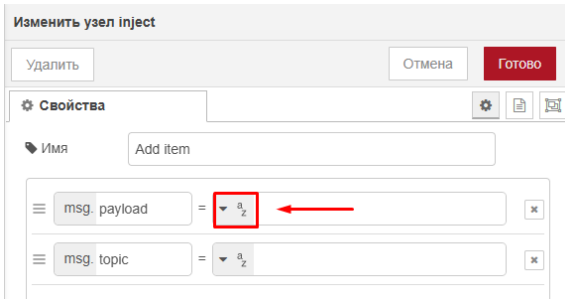


Рис. 3.31. Окно настроек узла Inject

3. На вход узла OPC UA Client добавьте узлы Inject для чтения нескольких тегов (см. рис. 3.32) и очистки массива идентификаторов узлов (см. рис. 3.33).

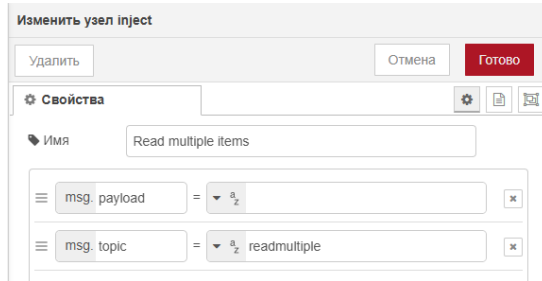


Рис. 3.32. Окно настроек узла Inject для чтения нескольких тегов

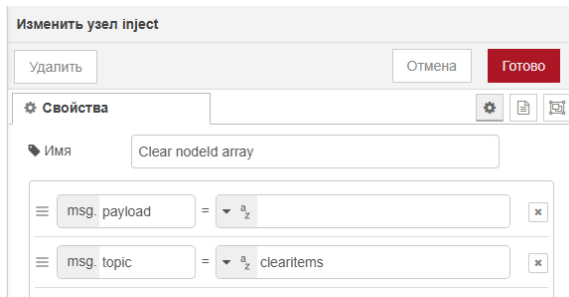


Рис. 3.33. Окно настроек узла Inject очистки массива идентификаторов узлов

4. Нажмите «Развернуть поток» (Deploy) и кликните по каждому узлу Inject. Кликните по узлу чтения нескольких тегов для запуска операции чтения тегов. На боковой панели отладки отобразятся значения тегов (см. рис. 3.34).

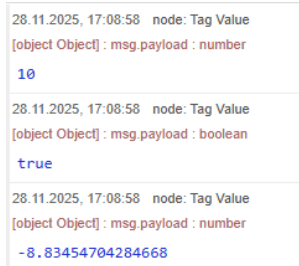


Рис. 3.34. Вывод информации о значениях тегов

3.6. Работа с OPC UA-сервером

Рассмотрим пример настройки OPC UA сервера (OpcUA-compact-server) с использованием Node-RED.

Добавьте узлы Inject, Function и Debug на рабочую область, для генерации 8 случайно сгенерированных Inputs и Outputs значений и сохранения их в виде потоковых переменных (см. рис. 3.35).

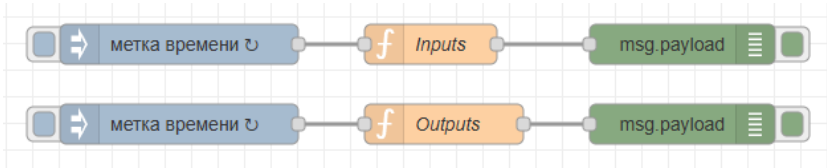


Рис. 3.35. Вывод информации о значениях тегов

В Node-RED функция `flow.set()` используется для хранения данных в контексте потока, позволяя сохранять информацию, доступную любому узлу внутри того же потока. Это особенно полезно, когда нужно обмениваться данными между разными частями вашего потока.

Значения потоковых переменных хранятся в памяти, щелкнув значок сворачиваемой стрелки на правой боковой панели и выбрав «Контекстные данные» (см. рис. 3.36) открывается список потоковых значений (см. рис. 3.37).

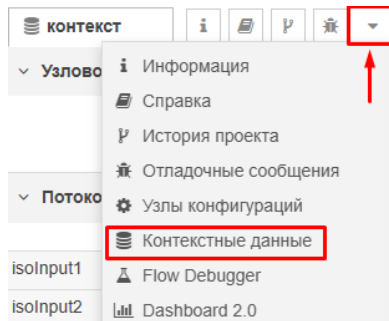


Рис. 3.36. Открытие контекстных данных

При нажатии на кнопку «обновить», список отображаемых Inputs и Outputs значений потоковых переменных меняется (см. Рис. 3.37).



The screenshot shows the Node-RED context menu with the 'Потоковый' (Stream) section expanded. A red box highlights the 'refresh' icon. Below the menu is a table of stream values.

Stream Name	Value
isoInput1	50.440475721280244
isoInput2	12.074009233835874
isoInput3	13.405493730445745
isoInput4	14.404319295072877
isoInput5	15.322563209972197
isoInput6	16.931189281604997
isoInput7	17.83208387063274
isoInput8	18.417939939359883
isoOutput1	10.901743279503556
isoOutput2	2.3395770108067087
isoOutput3	3.8334593785208533
isoOutput4	4.196468436145752
isoOutput5	5.286898098136687
isoOutput6	6.0667363899637605
isoOutput7	7.150803029065509
isoOutput8	8.326094674390896

Рис. 3.37. Вывод информации о потоковых значениях

3.6.1. Настройка OPC-сервера

1. Перетащите узел OpсUA-compact-server на рабочую область (см. рис. 3.38).



Рис. 3.38. Настройка узла OpсUA-compact-server

2. Дважды кликните по узлу OpсUA-compact-server, чтобы открыть настройки конфигурации. В панели конфигурации (Settings) введите номер порта (см. рис. 3.39).

3. Основной функционал Node-RED

Изменить узел орсуа-compact-server

Удалить Отмена Готово

Свойства

Settings Limits Security Users & Set Address Sp Discovery

Port 54845

Resource Path UA/NodeRED/Compact

Product URI NodeOPCUA-Server-(port)

Alternate Hostname

Init Timeout 1000

Shutdown Timeout 100

Close Timeout 200

Имя

Auditing

Show Activities

Show Errors

Рис. 3.39. Ввод номера порта на вкладке Settings

3. Переключитесь на вкладку «Discovery» в настройках конфигурации. Здесь вам нужно создать URL Endpoint для сервера OPC-UA (см. рис. 3.40).

Изменить узел орсуа-compact-server

Удалить Отмена Готово

Свойства

Settings Limits Security Users & Set Address Sp Discovery

Register Server Method HIDDEN

- HIDDEN: the server doesn't expose itself to the external world
- MDNS: the server publish itself to the mDNS Multicast network directly
- LDS: the server registers itself to the LDS or LDS-ME (Local Discovery Server)

Discovery

Endpoint Url opc.tcp://192.168.10.122:54845

MDNS capabilities NA,DA,...

Рис. 3.40. Ввод информации о URL

4. На вкладке «Security» по умолчанию активен анонимный доступ (для производственной системы позже можно установить настройки безопасности) (см. рис. 3.41).

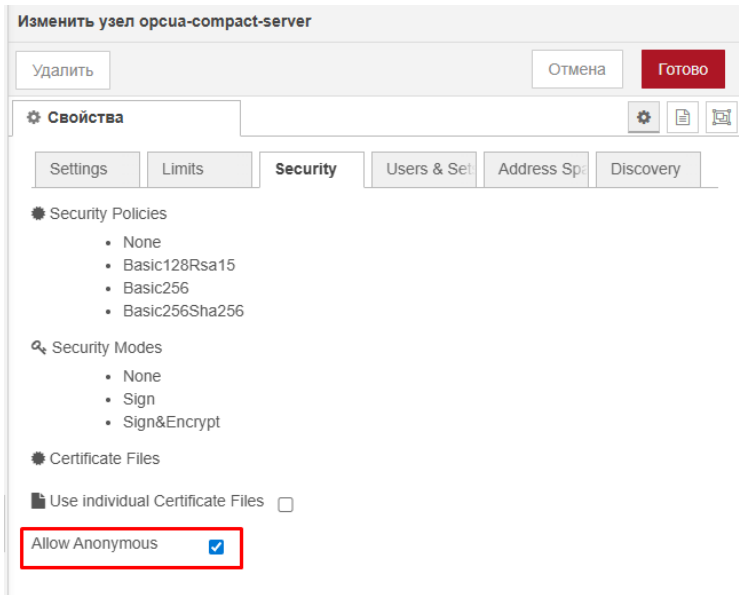


Рис. 3.41. Настройки безопасности

Вкладка «**Адресное пространство**» – это место, где строится информационная модель сервера OPC UA. Эта модель определяет структуру и данные, доступные клиентам OPC UA, используя классы и методы из библиотеки node-opcua SDK (см. рис. 3.42).

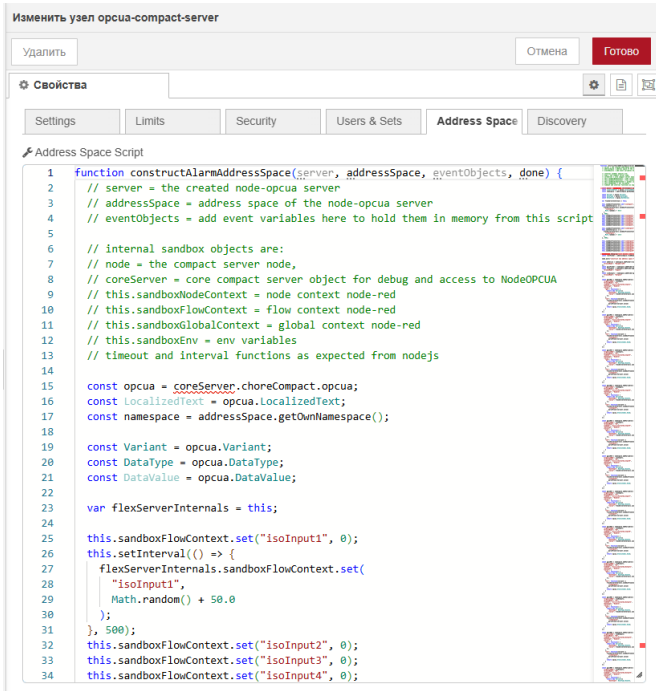


Рис. 3.42. Настройка адресного пространства

3.6.2. Настройка соединения с клиентом UaExpert

В качестве OPC UA клиента для подключения к серверу будем использовать «UaExpert» – свободное доступное ПО, которое можно загрузить из Интернета.

1. Создайте новое соединение: сервер «Добавить» – откроется диалоговое окно «Добавить сервер» (см. рис. 3.43).
2. Выберите вкладку «Advanced». Введите параметры подключения: URL вашего сервера OPC UA (например: `opc.tcp://192.168.10.122:54845`). Настройте все необходимые настройки безопасности или правила подключения в зависимости от конфигурации вашего сервера.

3. Основной функционал Node-RED

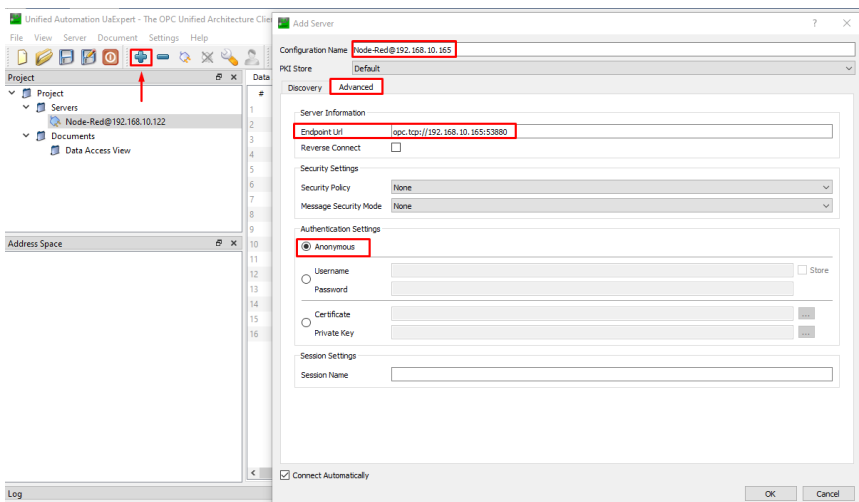


Рис. 3.43. Добавление сервера для подключения

- Щелкните Сервер «Подключить» и откроется диалоговое окно проверки сертификата с сообщением об ошибке.
- Выберите параметр «Временно принять сертификат сервера для этого сеанса» и нажмите «Продолжить». Соединение установлено, и объекты отображаются в представлении «Адресное пространство».
- Разверните объект Objects -> RaspberryPI -> GPIO -> Inputs -> Outputs.
- Выберите переменные из Inputs и Outputs и перетащите их в область представления доступа к данным. Можно визуально отслеживать как в режиме реального времени меняются значения переменных по мере работы системы (см. рис. 3.44).

3. Основной функционал Node-RED

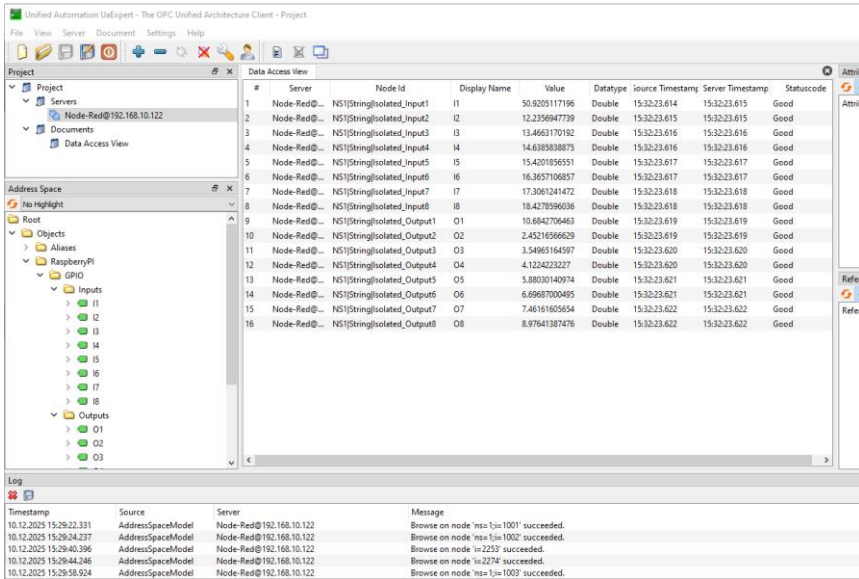


Рис. 3.44. Переменные в UaExpert

3.7. Визуализация данных с помощью Node-RED Dashboard

Node-RED Dashboard – это набор узлов и компонентов пользовательского интерфейса, которые позволяют создавать веб-панели в Node-RED. Он предоставляет виджеты, такие как кнопки, диаграммы, датчики и текстовые поля для отображения и взаимодействия с данными из ваших потоков в режиме реального времени.

Для установки пакета необходимо:

1. В меню в правом верхнем углу выбрать «Управление палитрой», перейти на вкладку «Установить» и выполнить поиск по запросу: *@flowfuse/node-red-dashboard*
2. После установки откройте вкладку на правой боковой панели, щелкнув значок сворачиваемой стрелки и выберите «Dashboard» (см. рис. 3.45).

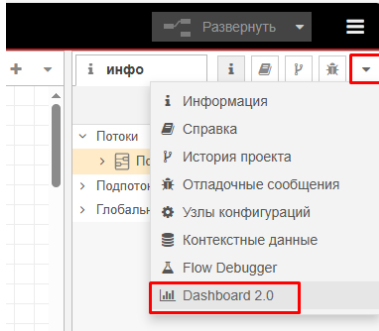


Рис. 3.45. Открытие «Dashboard»

Dashboard добавляет отдельную боковую панель в Node-RED для централизованного просмотра ваших страниц, групп и виджетов. Отсюда вы можете добавлять новые страницы и группы, менять существующие настройки и перестраивать контент по своему вкусу (см. рис. 3.46).

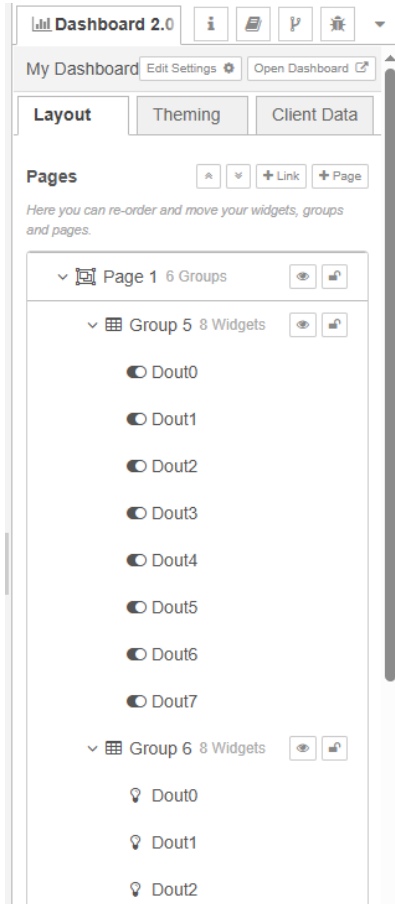


Рис. 3.46. Боковая панель «Dashboard»

3. После установки узлов перетащите узел из палитры в рабочую область, нажмите кнопку «Развернуть», а затем на кнопку «Open Dashboard» (см. рис. 3.47).

3. Основной функционал Node-RED

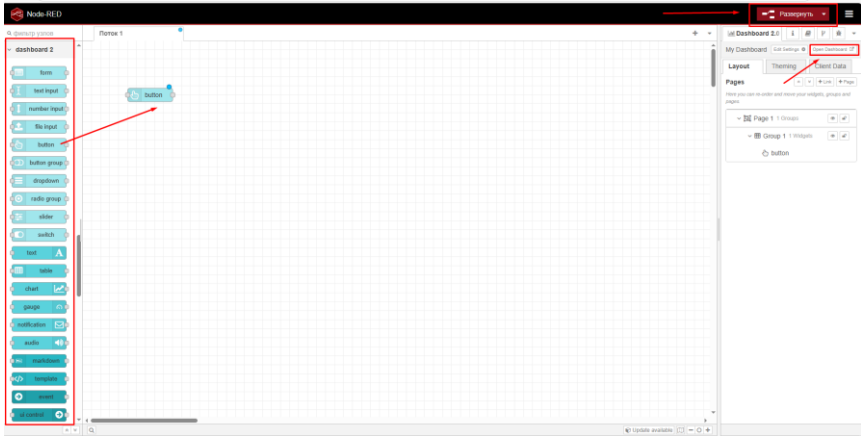


Рис. 3.47. Работа с узлами палитры «Dashboard»

Dashboard автоматически создаст новую страницу, в которой будет располагаться выбранный узел (см. рис. 3.48).

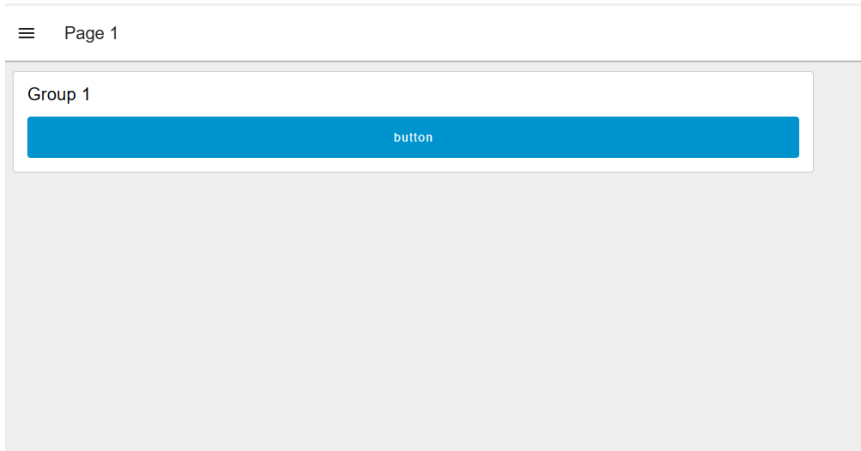


Рис. 3.48. Окно визуализации

4. Возможности интеграции Node-RED

4.1. Интеграция с облачными платформами

Node-RED предоставляет широкие возможности для интеграции с ведущими облачными платформами через специализированные ноды, доступные в менеджере палитр. Эта интеграция позволяет создавать гибридные решения, объединяющие локальные устройства и облачные сервисы.

Табл. 4. Поддерживаемые платформы и способы подключения

Платформа	Основные ноды	Ключевые возможности
AWS (Amazon Web Services)	node-red-contrib-aws, MQTT	IoT Core, Lambda, S3, DynamoDB, SNS, SQS
Microsoft Azure	node-red-contrib-azure-iot-hub, node-red-contrib-azure-blob-storage	IoT Hub, Event Hubs, Blob Storage, Cosmos DB, Functions
Google Cloud Platform (GCP)	node-red-contrib-google-cloud, node-red-contrib-firebase	Pub/Sub, Firebase Realtime Database, Firestore, Cloud Functions
IBM Cloud / Watson	Встроенные ноды Watson, node-red-contrib-ibm-cloud	AI сервисы (NLU, Speech-to-Text), IoT Platform, Cloudant
Alibaba Cloud	node-red-contrib-alicloud	IoT Platform, Tablestore, Function Compute
Oracle Cloud	node-red-contrib-oci	Oracle IoT, Autonomous Database, Functions

Порядок подключения:

1. Нажмите на меню в правом верхнем углу и выберите «Управление палитрой».
2. Перейдите на вкладку «Установить», найдите нужную ноду (например, node-red-contrib-azure).
3. Нажмите на кнопку «Установить» во всплывающем окне.

4. Настройте учетные данные (API-ключи, токены, endpoint'ы) в конфигурации ноды.

4.2. Подключение к базам данных

Node-RED предоставляет универсальный интерфейс для работы с различными системами управления базами данных (СУБД), что делает его идеальным инструментом для сбора, обработки и хранения данных IoT-устройств, бизнес-метрик и другой информации.

Табл. 5. Поддерживаемые базы данных и способы подключения

База данных	Нода	Установка
MySQL	node-red-node-mysql	Встроена по умолчанию
PostgreSQL	node-red-contrib-postgresql	Установка через менеджер
MongoDB	node-red-node-mongodb	Встроена
SQLite	node-red-node-sqlite	Установка через менеджер
InfluxDB	node-red-contrib-influxdb	Для IoT-данных
Redis	node-red-contrib-redis	Для кэширования

4.3. Взаимодействие с REST API

REST API (Representational State Transfer Application Programming Interface) – это стандартный подход для взаимодействия между системами через HTTP-протокол. Node-RED предоставляет мощные инструменты для работы с REST API как в роли клиента (отправка запросов), так и в роли сервера (создание собственного API).

Отправка запросов (HTTP Request)

1. Добавьте ноду HTTP Request;
2. Укажите:
 - Метод (GET/POST/PUT/DELETE).
 - URL (например, <https://api.weatherapi.com/v1/current.json?key=...&q=Moscow>)
 - Заголовки (если нужно: Content-Type: application/json).
 - Тело (для POST — JSON в msg.payload).

3. Результат приходит в `msg.payload`.

Создание своего API (HTTP In / HTTP Response)

1. Добавьте ноду HTTP In, настройте метод и маршрут (например, GET /api/status).
2. Обработайте входящий запрос (например, извлеките данные из базы).
3. Подключите ноду HTTP Response для формирования и отправки ответа клиенту.

4.4. Работа с мессенджерами

Платформа Node-RED обеспечивает эффективную интеграцию с популярными мессенджерами, что позволяет реализовывать следующие функциональные возможности:

- Разработка чат-ботов;
- Организация автоматизированных систем уведомлений;
- Создание полноценных диалоговых интерфейсов.

Основные направления интеграции включают:

- **Telegram** – поддержка создания ботов, отправки и приёма текстовых сообщений, а также обработки пользовательских команд;
- **WhatsApp** – взаимодействие через сторонние шлюзы, такие как Twilio, с возможностью отправки и получения сообщений в рамках соблюдения политик платформы.

4.5. Поддержка IoT-протоколов

Node-RED нативно поддерживает ключевые протоколы, используемые в экосистеме Интернета вещей:

- **MQTT** – основной протокол для обмена данными между IoT-устройствами и сервером. Поддерживается через встроенную ноду MQTT in/out.
- **HTTP/HTTPS** – применяется для устройств, не поддерживающих MQTT (например, ESP8266 с HTTP-клиентом). Node-RED может выступать как HTTP-сервер (через ноду HTTP In) или HTTP-клиент (через HTTP Request).

Лист регистрации изменений

Дата изменения	Описание изменения	Примечание