

Сборщик-компакт-0801

FrontControl Compact-0801

FCC-0801

Полное руководство пользователя



FCC-0801 — это суперкомпактный одноплатный компьютер на базе модуля Napi-C, работающий под управлением Linux (Armbian или NapiLinux). Хорошо подходит для промышленных и автоматизированных решений.

Первое включение модуля FCC-0801

Комплект поставки модуля сборщика FCC-0801 показан на рис. 1.1.



Рис. 1.1. Комплект поставки модуля FCC-0801

В состав комплекта входят:

- сборщик-компакт с ARMBian — 1 шт.;
- MicroSD карта 32 Гб с NapiLinux — 1 шт.;
- USB флешка с готовыми образами Armbian и NapiLinux — 1 шт.;
- консоль USB-UART — 1 шт.;
- кабель USB-A - USB type-c 1 шт.;

Для первоначальной настройки модуля вам также потребуется:

- локальная сеть, желательно с сервером DHCP;
- блок питания или преобразователь POE;
- компьютер с операционной системой Windows или Linux

В следующем разделе описан способ установки для тех случаев, когда сеть с сервером DHCP недоступна.

Лицевая панель модуля FCC-0801 показана на рис. 1-2.

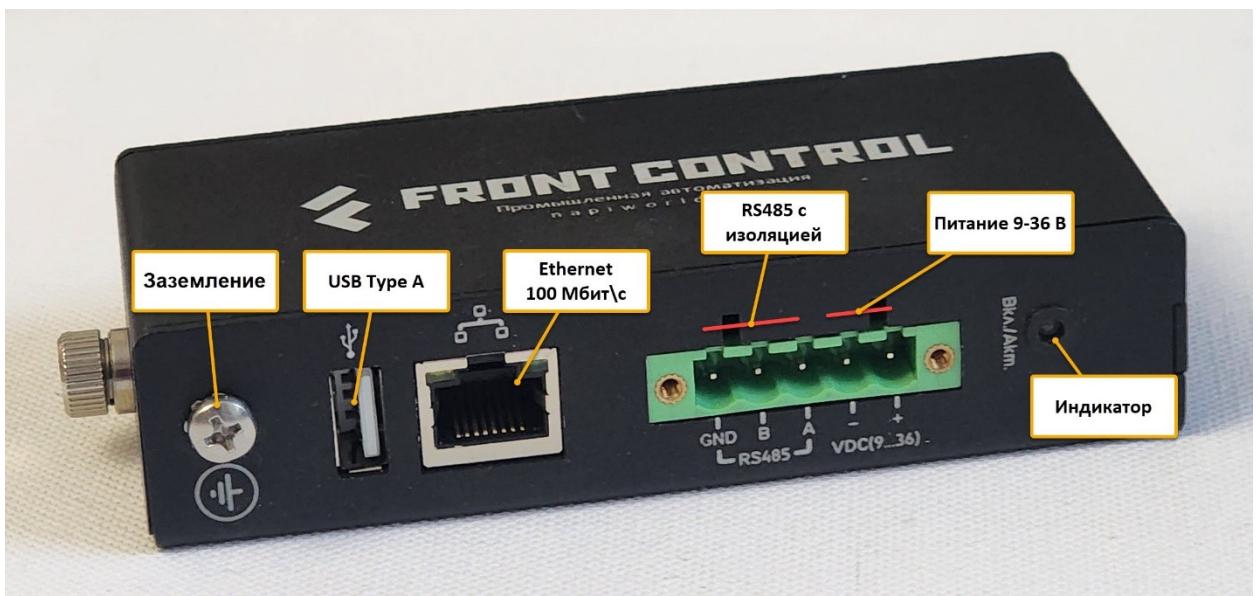


Рис. 1-2. Лицевая панель модуля FCC-0801

При первом включении устройства выполните следующие подключения:

1. подключите источник питания напряжением от 9 В до 36 В к разъему питания **VDC(9...36)**, соблюдая полярность, или подсоедините питание к преобразователю POE;
2. подключите устройство к локальной сети вашего офиса

Если все сделано правильно, после включения питания должен загореться зеленый индикатор **Вкл./Акт.**. Через некоторое время замигают светодиоды на разъеме локальной сети.

Настройка сетевого интерфейса

Необходимо определить текущий адрес IP модуля FCC-0801, и, при необходимости, прописать выделенный адрес IP.

Предусмотрено несколько способов определения адреса IP модуля.

Использование Интерфейс роутера

Подключитесь к Веб-интерфейсу роутера, узнав его адрес IP у вашего системного или сетевого администратора. Для сети 192.168.0.0/24 это может быть адрес такого вида: <http://192.168.0.1/>. Вам также будет нужен логин и пароль для подключения.

Вид Веб-интерфейса роутера зависит от его модели и версии. Вам нужно найти в этом интерфейсе раздел **LAN (Локальная сеть)** или аналогичный, где будет показан список адресов IP, выделенный через DHCP роутером. Для роутера ZXHN H118N этот список показан на рис. 2.1.

Выделенные адреса

MAC адрес	IP адрес	Оставшееся время аренды	Имя хоста	Порт
c2:4e:65:fc:25:05	192.168.0.15	77360	Tab-S9-FE-pol-zo	SSID1
7c:2f:80:2e:79:8f	192.168.0.10	46702	A510-IP	LAN3
04:42:1a:f0:eb:fc	192.168.0.14	80366	FROLOV-AMD	LAN3
7c:2f:80:2e:79:20	192.168.0.11	45647	A510-IP	LAN3
92:f9:61:33:bf:4c	192.168.0.12	67525	Galaxy-S22-Ultra	SSID1
1c:57:dc:5c:f3:dc	192.168.0.13	55359	Air-Dina	SSID1
18:af:61:3c:61:9b	192.168.0.16	77853	iPhone	SSID1
3a:5b:37:c0:ca:01	192.168.0.17	73232	iPhone	SSID1
02:07:23:87:2d:e5	192.168.0.18	84639	напи-rk3308b-s	LAN3

Рис. 2.1. Список полученных через DHCP адресов IP для роутера ZXHN H118N

Обратите внимание, что адрес **192.168.0.18** здесь был выделен хосту с именем **напи-rk3308b-s**. Это как раз и есть нужный нам адрес IP модуля FCC-0801, полученный от DHCP роутера.

На рис. 2.2 показана таблица адресов IP, выделенных через DHCP роутера Ruijie.

DHCP клиенты				
	№	Имя хоста	IP-адрес	MAC-адрес
	1	hp	192.168.110.76	f4:3b:d8:f7:1a:61
	2	напи-rk3308b-s	192.168.110.96	02:05:60:a2:7f:6f
	3	napi-armbian	192.168.110.47	02:3b:8f:1a:24:c6
	4	DESKTOP-7GQM322	192.168.110.99	68:7a:64:b1:72:2c

Рис. 2.2. Список полученных через DHCP адресов IP для роутера Ruijie

Здесь модулю FCC-0801 выделен адрес **192.168.110.96**.

Если адрес IP определен, попробуйте ввести в браузере адрес Веб-интерфейса конфигурирования NapiConfig вида <https://192.168.0.18/>, где вместо 192.168.0.18 нужно указать этот адрес модуля.

При открытии NapiConfig нужно разрешить использование самостоятельно подписанного сертификата (рис. 2.3).

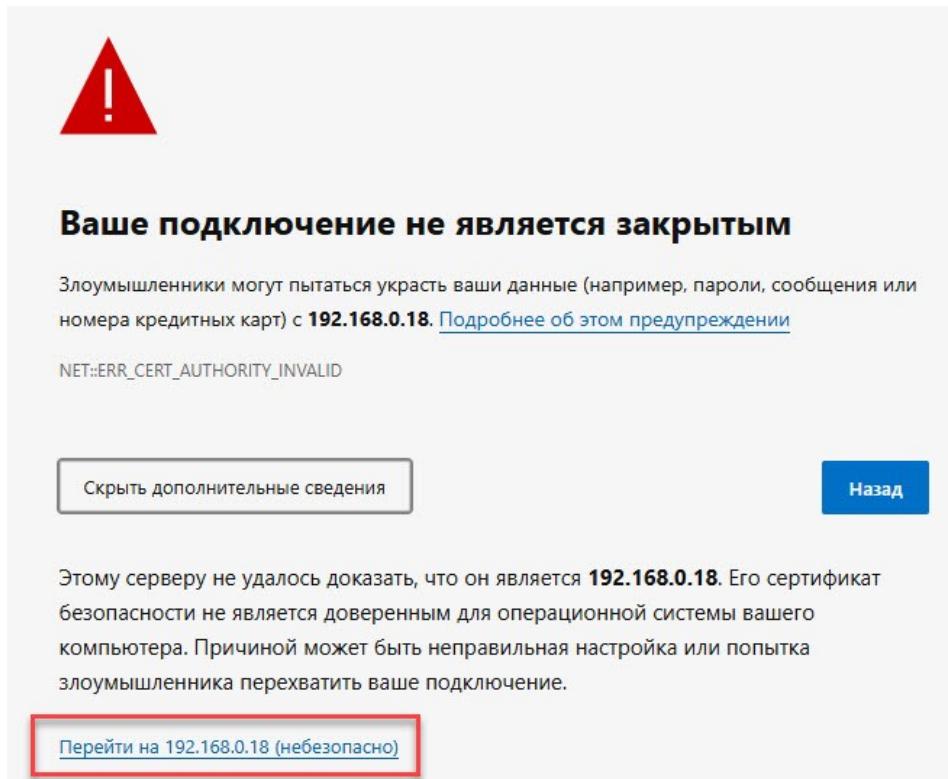


Рис. 2.3. Предупреждение об отсутствии доверенного сертификата

Чтобы использовать сертификат, установленный для NapiConfig по умолчанию, щелкните ссылку, выделенную на рис. 2.3 красной рамкой или аналогичную для вашего браузера.

После этого в браузере откроется окно Веб-интерфейса NapiConfig (рис. 2.4).

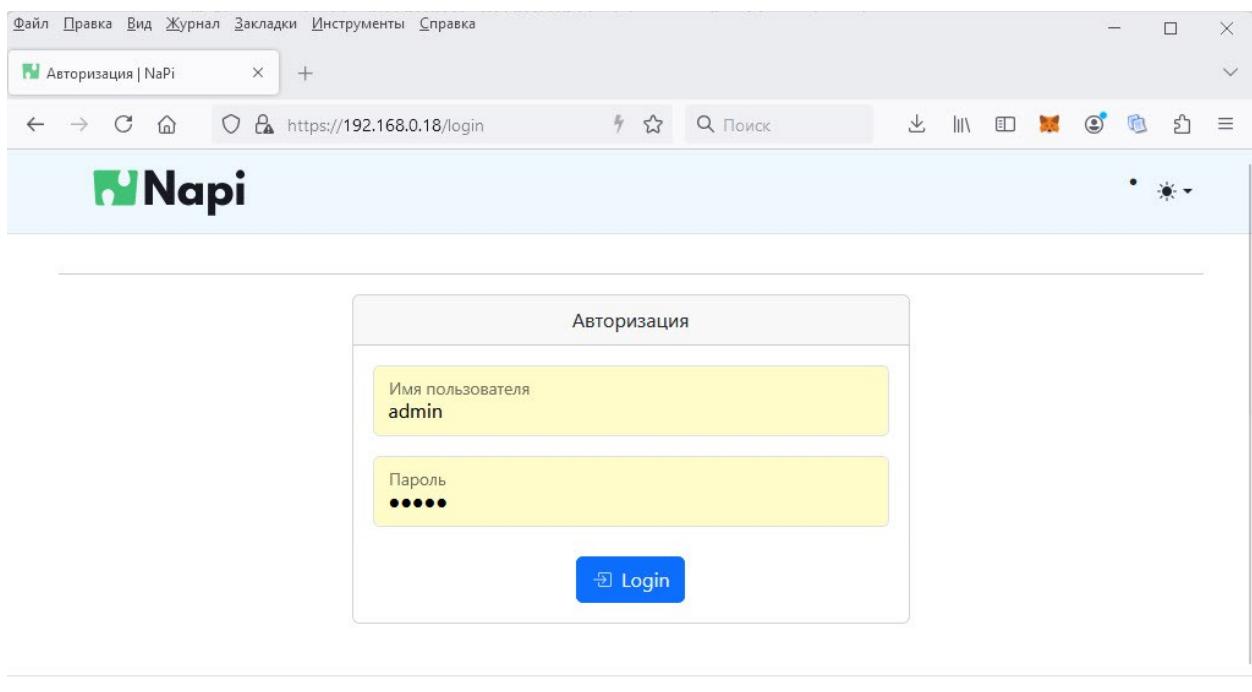


Рис. 2.4. Вход в Веб-сервис настроек модуля FCC-0801

Введите здесь логин admin и пароль admin. Теперь можно приступать к настройке модуля.

Ниже в разделе 2.5 рассказано, как установить собственный сертификат для вашего доменного имени.

Поиск через сканирование сети

Если нет доступа к Веб-интерфейсу роутера, можно найти адрес IP модуля FCC-0801 через сканирование сети, к которой он подключен. Здесь также предполагается, что в сети есть сервер DHCP, который выдал модулю адрес IP.

Для сканирования можно использовать бесплатную программу Angry IP Scanner, которую можно скачать на сайте <https://angryip.org/> для Windows, Mac OS и Linux. Также на сайте доступны ее исходные коды.

Если вы работаете в среде Windows, скачайте и запустите Windows Installer. После установки запустите программу (рис. 2.5).

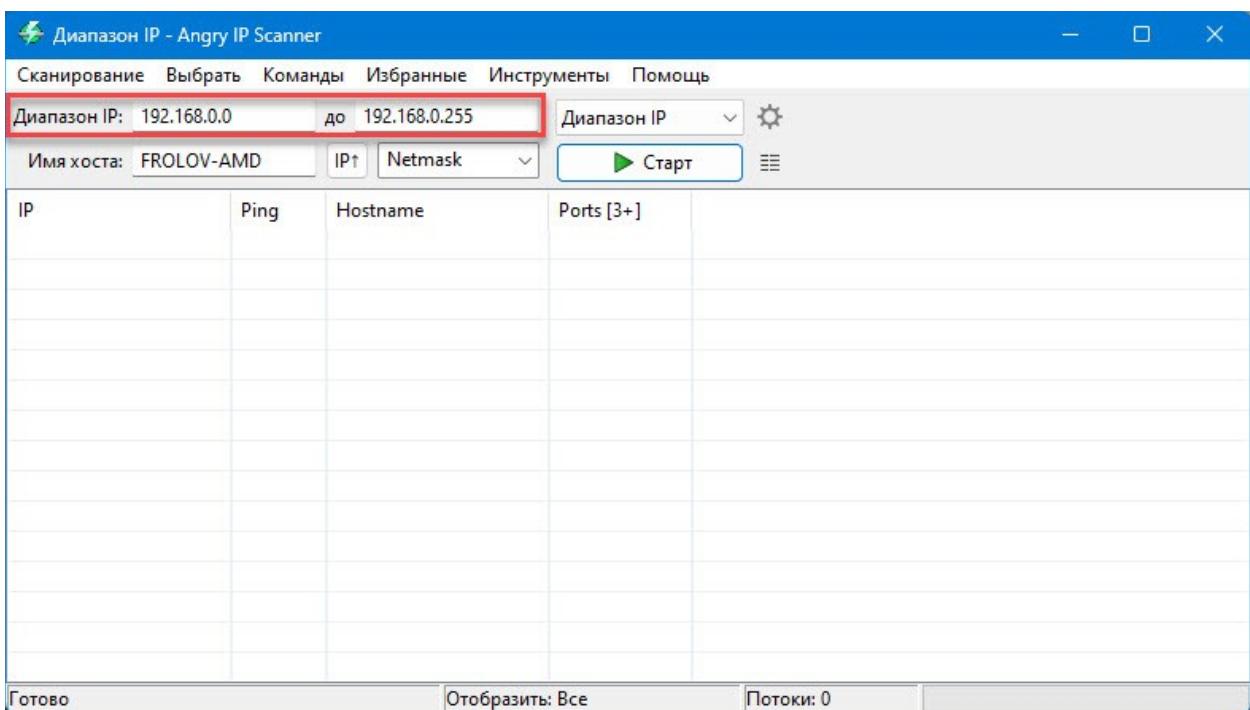


Рис. 2.5. Главное окно программы Angry IP Scanner

В полях **Диапазон IP** и **до** указаны начальный и конечный адреса IP текущей локальной сети, определенные автоматически (выделены красным прямоугольником на рис. 2.5). Проверьте эти адреса и при необходимости проконсультируйтесь с системным администратором.

Если администратор недоступен, откройте командную строку в Windows и введите команду **ipconfig** (рис. 2.6).

```
Administrator: Командная строка
C:\Windows\System32>ipconfig

Настройка протокола IP для Windows

DNS-суффикс подключения . . . . . :
Адаптер беспроводной локальной сети Подключение по локальной сети* 10:

Состояние среды. . . . . : Среда передачи недоступна.
DNS-суффикс подключения . . . . . :

Адаптер Ethernet Ethernet:

DNS-суффикс подключения . . . . . :
Локальный IPv6-адрес канала . . . . . : fe80::23cb:7715:4fa0:4c0f%4
IPv4-адрес . . . . . : 192.168.0.14
Маска подсети . . . . . : 255.255.255.0
Основной шлюз. . . . . : 192.168.0.1
```

Рис. 2.6. Определение настроек сети командой ipconfig

Как видите, здесь маска подсети 255.255.255.0, адрес шлюза 192.168.0.1. Таким образом, вы можете сканировать адреса подсети 192.168.0.0/24, оставив содержимое полей **Диапазон IP** и **до**, указанные на рис. 2.5.

Если вы работаете в ОС Linux, введите команду **ip a** (рис. 2.7).

```
frolov@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:48:07:a9 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.0.19/24 brd 192.168.0.255 scope global dynamic noprefixroute ens33
        valid_lft 68357sec preferred_lft 68357sec
    inet6 fe80::eed4:f514:ee2b:12b5/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Рис. 2.7. Определение настроек сети в ОС Linux

Здесь команда вывела на консоль адрес узла 192.168.0.19 в подсети 192.168.0.0/24, что соответствует диапазону адресов IP от 192.168.0.0 до 192.168.0.255.

Когда вы определили диапазон адресов IP для сканирования, задайте проверяемые номера портов. Мы будем искать адрес IP узла модуля FCC-0801 по открытым портам 22, 443 и 8082.

Выберите из меню **Инструменты** программы Angry IP Scanner строку **Предпочтения**, а затем откройте вкладку **Порты** (рис. 2.8).

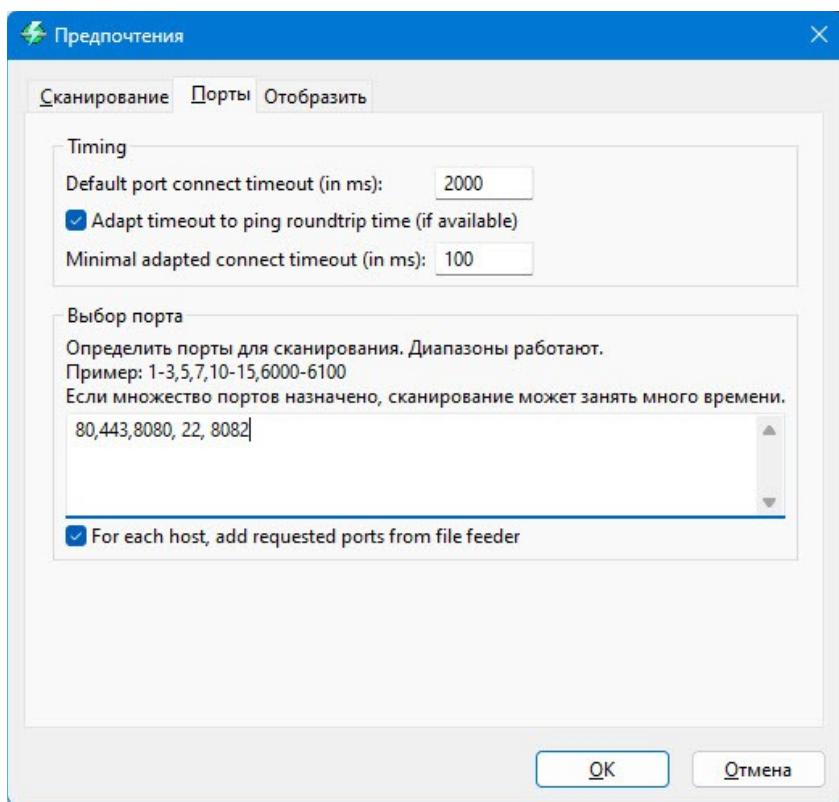


Рис. 2.8. Настройка портов сканирования

После добавления нужных портов щелкните кнопку **OK**. Далее для запуска сканирования воспользуйтесь кнопкой **Старт**.

Когда сканирование завершится, в окне программы обнаруженные узлы будут отмечены кружками зеленого и синего цвета (рис. 2.9).

Диапазон IP - Angry IP Scanner

Сканирование Выбрать Команды Избранные Инструменты Помощь				
Диапазон IP:	192.168.0.0	до	192.168.0.255	Диапазон IP
Имя хоста:	FROLOV-AMD	IP↑	Netmask	Старт
IP	Ping	Hostname	Ports [5+]	
192.168.0.1	3 ms	rt	80,443	
192.168.0.2	[n/a]	[n/s]	[n/s]	
192.168.0.3	[n/a]	[n/s]	[n/s]	
192.168.0.4	[n/a]	[n/s]	[n/s]	
192.168.0.5	[n/a]	[n/s]	[n/s]	
192.168.0.6	[n/a]	[n/s]	[n/s]	
192.168.0.7	[n/a]	[n/s]	[n/s]	
192.168.0.8	[n/a]	[n/s]	[n/s]	
192.168.0.9	[n/a]	[n/s]	[n/s]	
192.168.0.10	4 ms	[n/a]	80	
192.168.0.11	4 ms	[n/a]	80	
192.168.0.12	[n/a]	[n/s]	[n/s]	
192.168.0.13	70 ms	[n/a]	[n/a]	
192.168.0.14	0 ms	FROLOV-AMD	8080	
192.168.0.15	[n/a]	[n/s]	[n/s]	
192.168.0.16	[n/a]	[n/s]	[n/s]	
192.168.0.17	[n/a]	[n/s]	[n/s]	
192.168.0.18	4 ms	[n/a]	22,443,8082	
192.168.0.19	4 ms	[n/a]	[n/a]	

Готово Отобразить: Все Потоки: 0

Рис. 2.9. Сканирование завершено

Обратите внимание, что только у узла с адресом 192.168.0.18 открыты порты 22, 443 и 8082. Это и есть наш узел модуля FCC-0801.

К сожалению, программа не показала имя хоста этого узла, так как не все роутеры сохраняют имена хостов. На рис. 2.10 показаны результаты сканирования сети с другим роутером.

Диапазон IP - Angry IP Scanner

Сканирование Выбрать Команды Избранные Инструменты Помощь				
Диапазон IP:	192.168.110.0	до	192.168.110.255	Диапазон IP
Имя хоста:	DESKTOP-7GQM322	IP↑	Netmask	Старт
IP	Ping	Hostname	Ports [5+]	
192.168.110.99	0 ms	DESKTOP-7GQM322.lan	[n/a]	→ Desktop
192.168.110.96	4 ms	napi-rk3308b-s	22,443,8082	→ NapiLinux
192.168.110.1	5 ms	[n/a]	80,443	→ Router
192.168.110.47	75 ms	napi-armbian.lan	22	→ ArmBian
192.168.110.2	[n/a]	[n/s]	[n/s]	
192.168.110.3	[n/a]	[n/s]	[n/s]	
192.168.110.4	[n/a]	[n/s]	[n/s]	
192.168.110.5	[n/a]	[n/s]	[n/s]	
192.168.110.6	[n/a]	[n/s]	[n/s]	
192.168.110.7	[n/a]	[n/s]	[n/s]	
192.168.110.8	[n/a]	[n/s]	[n/s]	
192.168.110.9	[n/a]	[n/s]	[n/s]	
192.168.110.10	[n/a]	[n/s]	[n/s]	
192.168.110.11	[n/a]	[n/s]	[n/s]	
192.168.110.12	[n/a]	[n/s]	[n/s]	

Готово Отобразить: Все Потоки: 0

Рис. 2.10. Результаты сканирования сети с другим роутером

Как видите, здесь был обнаружен узел модуля FCC-0801 с адресом 192.168.110.96 и для него определилось имя хоста как napi-rk3308b-s.

Если вы работаете в ОС Linux, то для сканирования портов вам не потребуется программа Angry IP Scanner. Установите утилиту nmap и запустите ее следующим образом:

```
sudo apt install nmap  
sudo nmap -p 22,443,8082 192.168.0.0/24
```

Эта команда покажет узлы с открытыми портами 22, 443 и 8082 в сети 192.168.0.0/24. Фрагмент результата сканирования с обнаруженным узлом 192.168.0.18 показан на рис.2.11.

```
Nmap scan report for 192.168.0.18 (192.168.0.18)  
Host is up (0.00058s latency).  
  
PORT      STATE SERVICE  
22/tcp    open  ssh  
443/tcp   open  https  
8082/tcp  open  blackice-alerts  
MAC Address: 02:07:23:87:2D:E5 (Unknown)
```

Рис. 2.11. Обнаружен узел модуля FCC-0801

Теперь, когда адрес узла обнаружен, введите в адресной строке браузера URL вида <https://192.168.0.18/>, указав свой адрес IP модуля FCC-0801. Вы увидите сайт конфигурирования модуля NapiConfig, показанный на рис. 2.4.

Установка статического IP-адреса в FCC-0801

Соедините напрямую FCC-0801 с хостом под управлением Windows.

Убедитесь, что на Вашем хосте не установлен никакой статический ip-адрес (должно быть как на картинке 2.3.12).

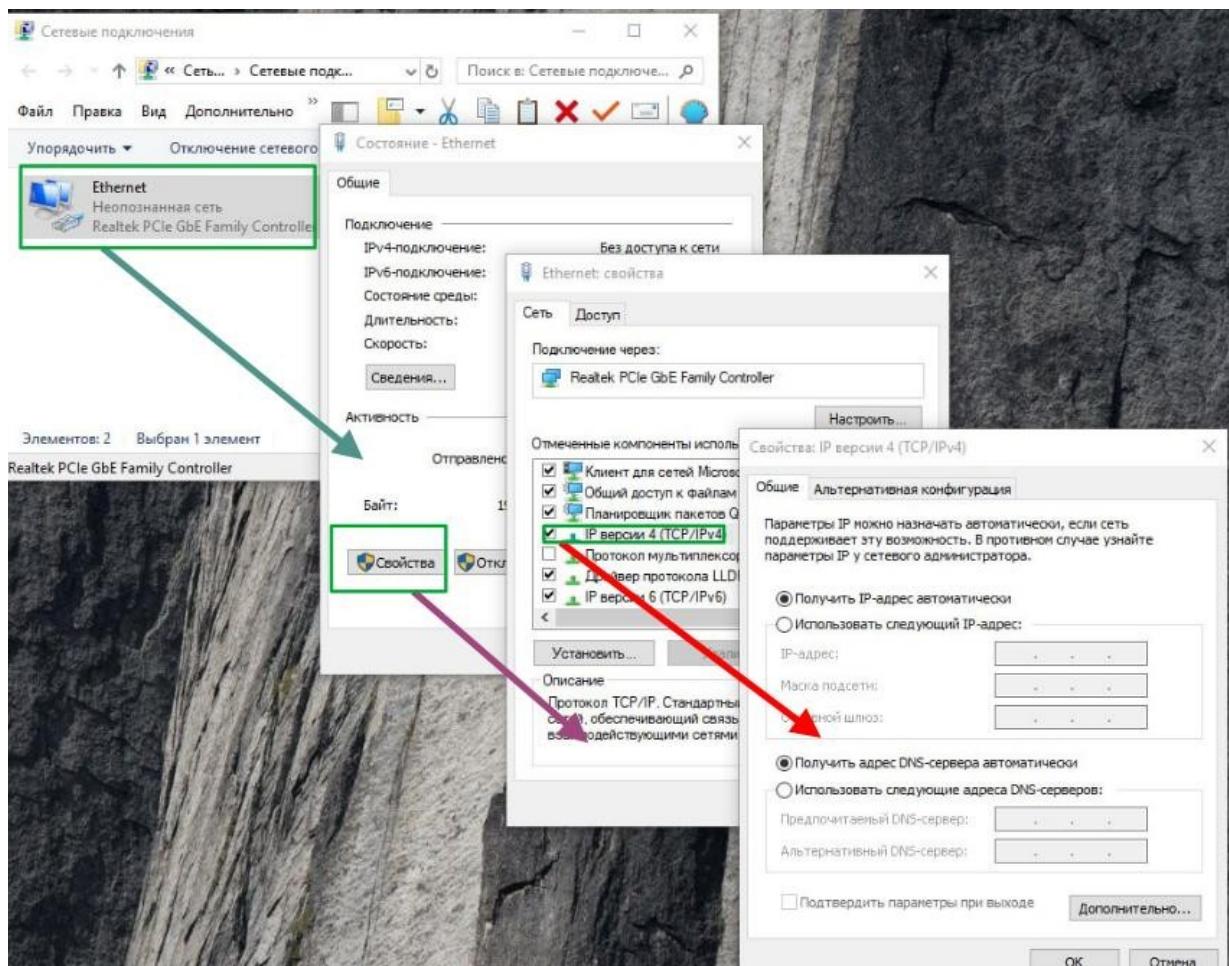


Рис. 2.3.12. На хосте не должен быть явным образом указан IP

Заходим на IP по умолчанию

Откройте Веб-браузер и зайдите на адрес: `<https://169.254.100.100>`
(префикс https:// в начале - обязательнен)

Рис. 2.3.13. Нажать дополнительно

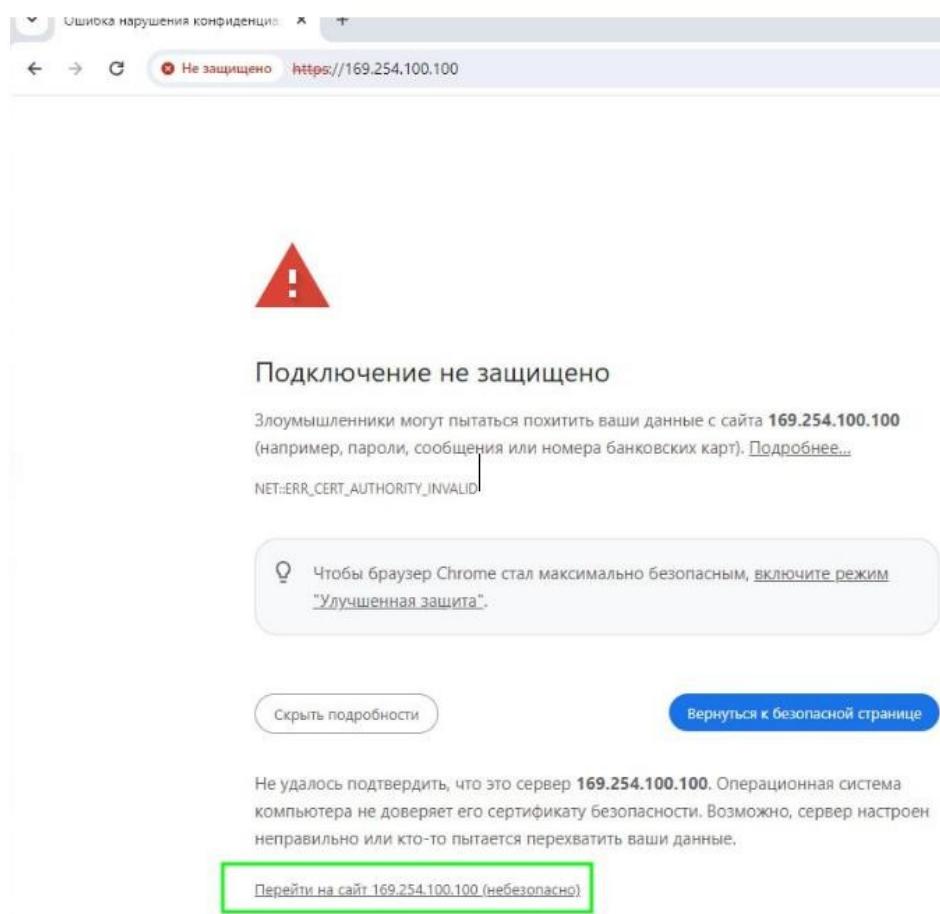


Рис. 2.3.14. Нажать «Перейти на сайт 169.254.100.100»

Войдите с логином\паролем по умолчанию или с тем, который Вы установили до того и выберите "Общие-Сеть" в верхнем меню.

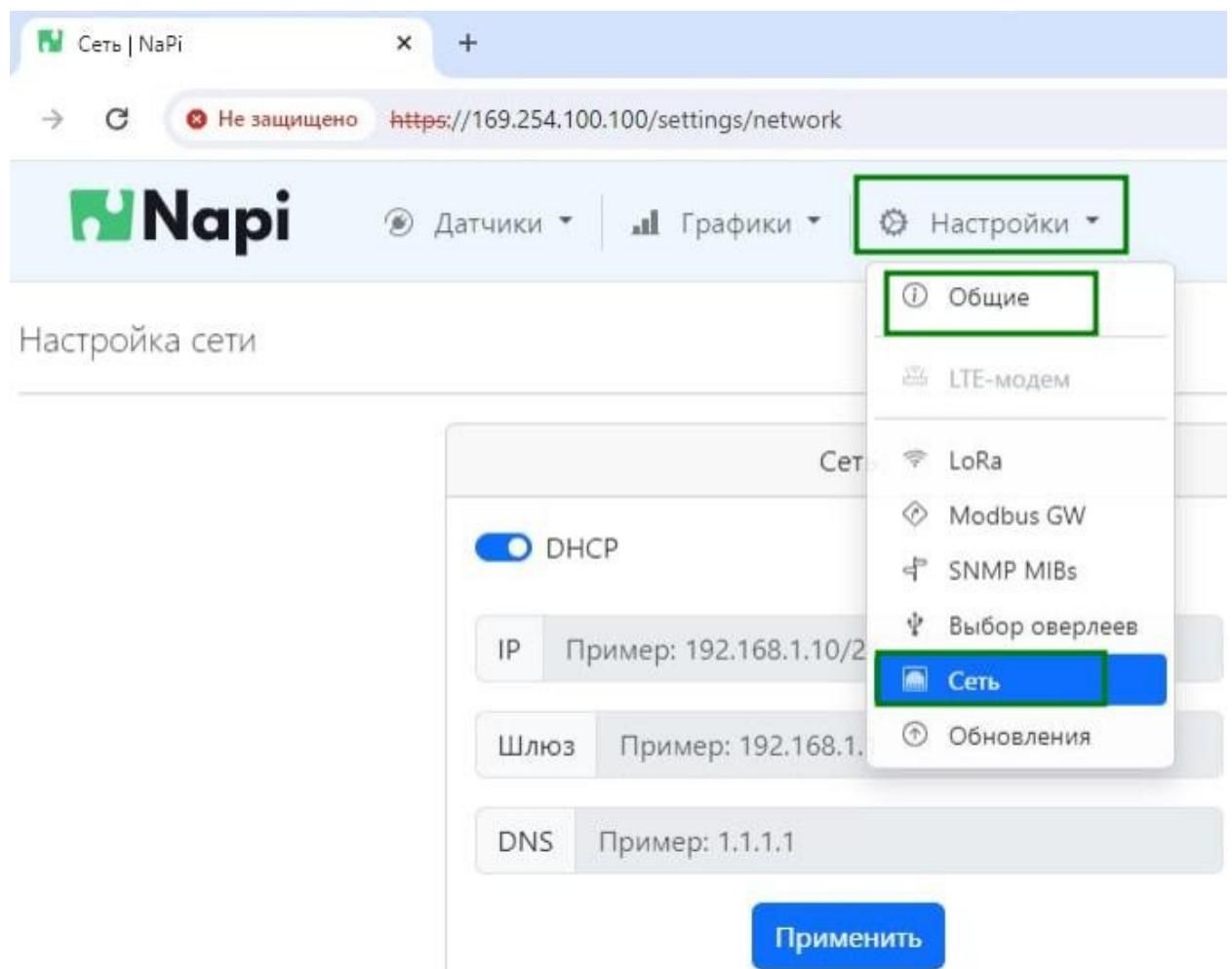


Рис. 2.3.15. Перейти в настройки Сети

Отключите флагок "DHCP" – у Вас появится возможность вводить параметры IP вручную.

Введите параметры

- IP/MASK
- Шлюз
- DNS

Нажмите "Применить"

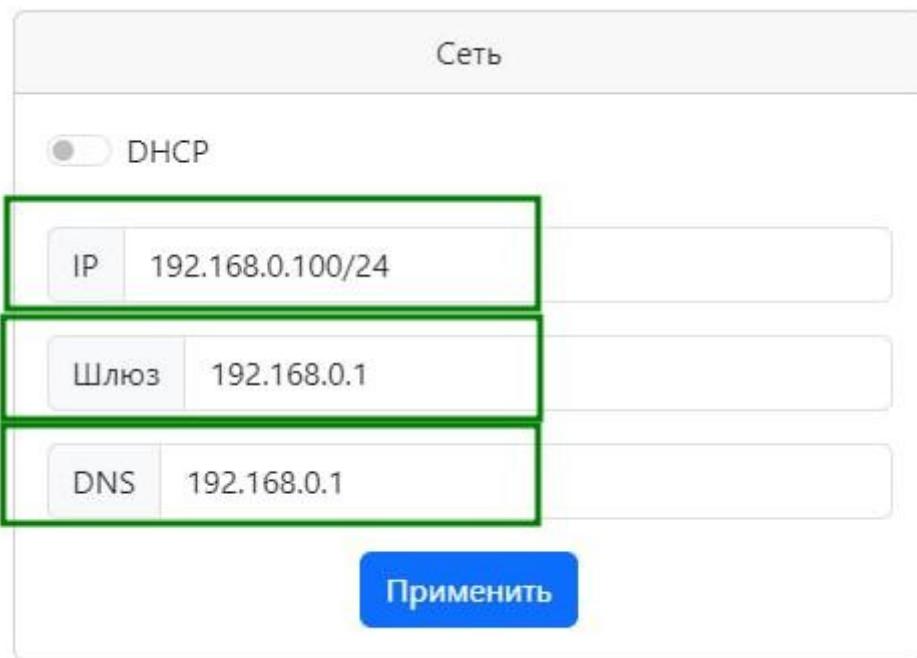


Рис. 2.3.16. Ввод статических IP-параметров

Зайдите на FCC-0801 уже по новому адресу.

Важно: адрес хоста тоже надо сменить, чтобы он оказался в той сети, которую Вы указали в статических параметрах модуля (в нашем примере можно указать 192.168.0.99).

Подключение FCC-0801 через консоль

Если на FCC-0801 ранее был установлен статический IP-адрес или что-то пошло не так в п 2.3 и 2.4, то можно настроить сетевые параметры, подключив к FCC-0801 консоль через UART.

Для подключения будет нужен кабель преобразователя UART-USB, который входит в комплект поставки FCC-0801.

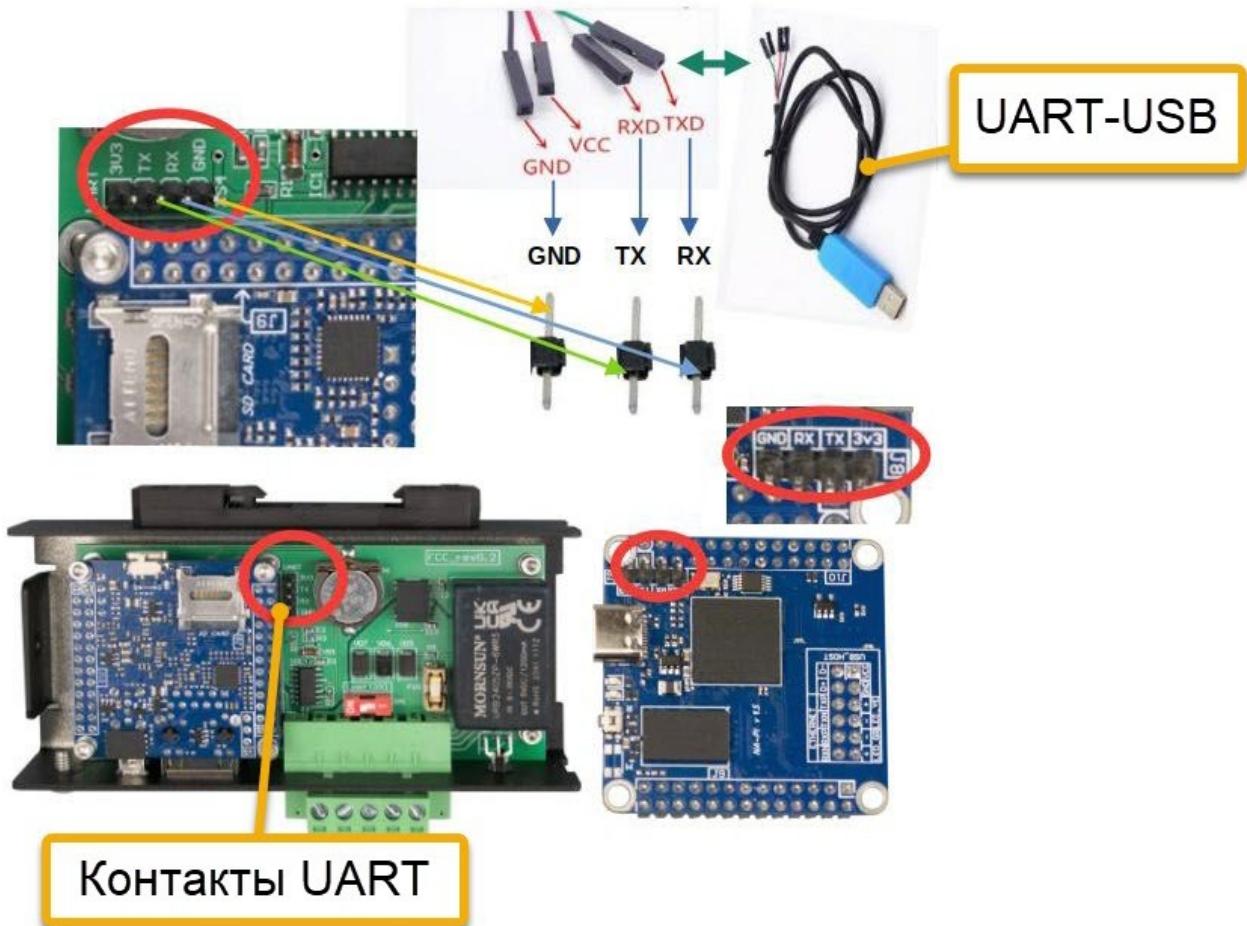


Рис. 2.17. Переходник UART-USB и контакты UART модуля FCC-0801

Если такого кабеля нет, можно использовать модуль преобразователя, например CH341A. Подключите его контакты следующим образом:

- контакт **GND** CH341A — к контакту **GND** модуля FCC-0801
- контакт **RX** CH341A — к контакту **TX** модуля FCC-0801
- контакт **TX** CH341A — к контакту **RX** модуля FCC-0801

Такое соединение показано на рис. 2.18.

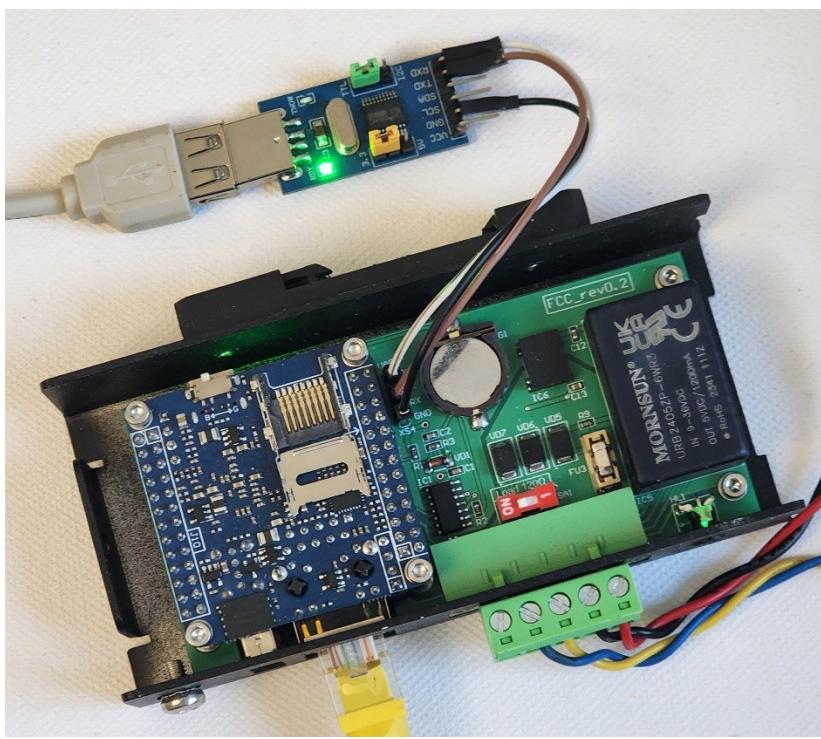


Рис. 2.18. Подключение модуля CH341A к модулю FCC-0801 через UART

▪ Работа с консолью в ОС Microsoft Windows

Если у вас установлена ОС Microsoft Windows, откройте диспетчер устройств, и найдите в нем раздел **Порты (COM и LPT)**. Этот раздел показан на рис. 2.14.

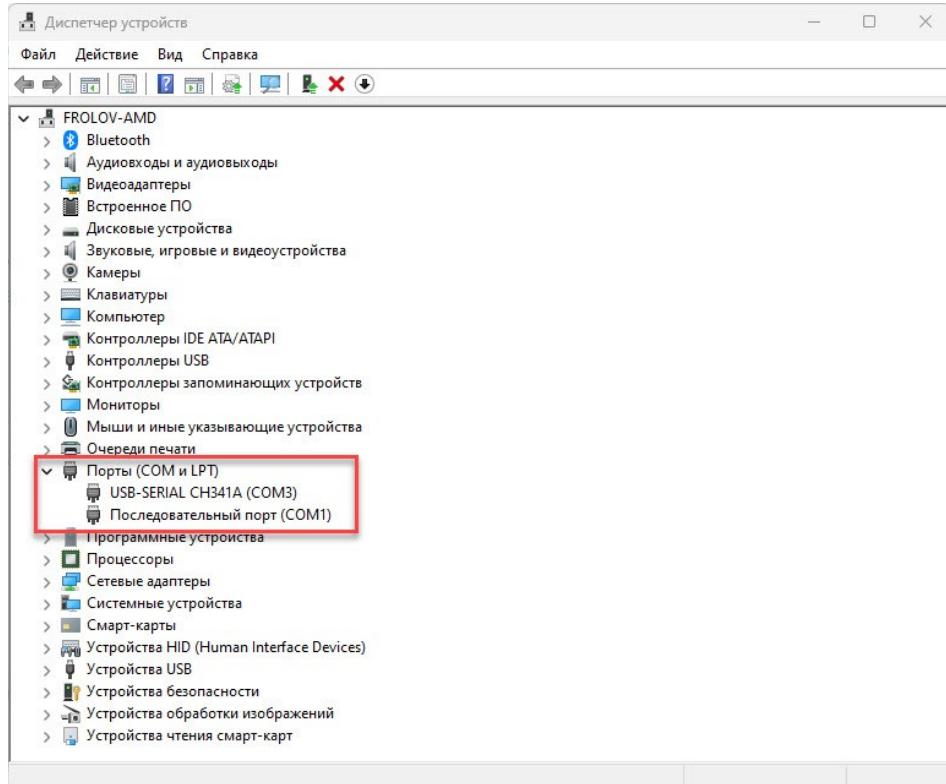


Рис. 2.14. Раздел Порты (COM и LPT) в диспетчере устройств

Запомните здесь номер порта COM, к которому подключен модуль CH314A или аналогичный. В нашем случае это COM3.

Далее запустите программу PuTTY, доступную для скачивания по адресу <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. Вы также можете найти эту программу на сайте <https://putty.org.ru/>.

Запустите PuTTY и в разделе **Connection** выделите строку **Serial** (рис. 2.15).

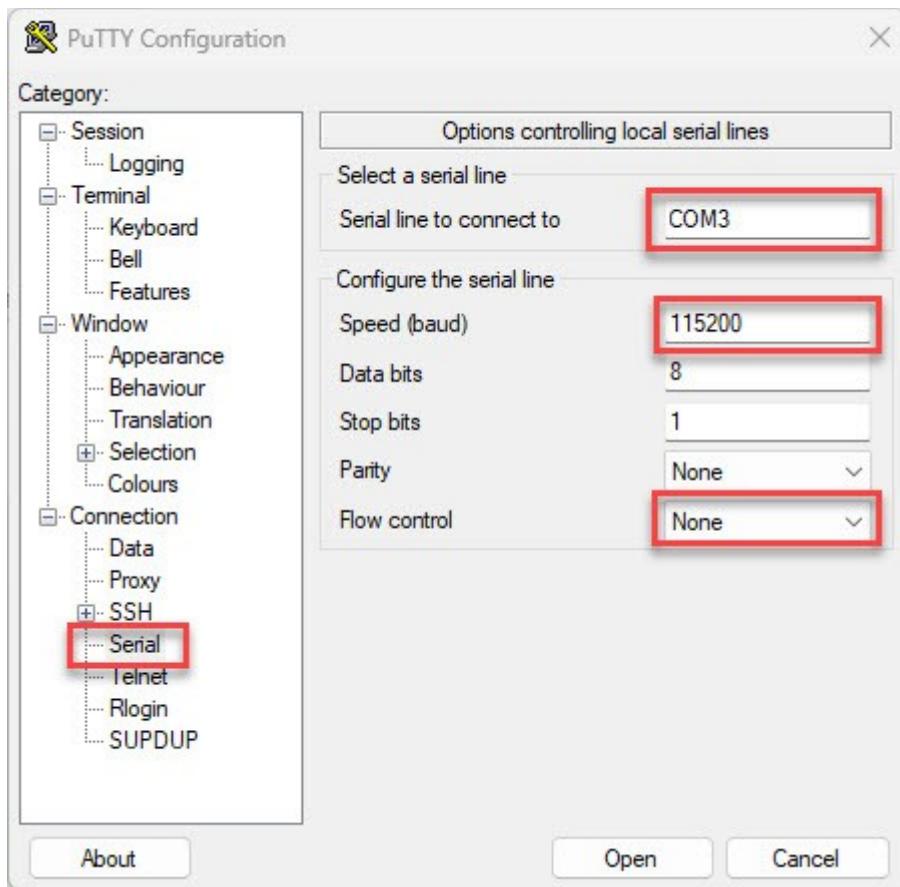


Рис. 2.15. Раздел Connection

Укажите в поле **Serial line to connect to** порт **COM3**, в поле **Speed (baud)** значение **115200**, а в меню **Flow control** выберите строку **None**.

Чтобы открыть соединение, выберите вкладку **Session** и выберите на ней тип соединения **Serial**. Далее щелкните кнопку **Open** (рис. 2.16).

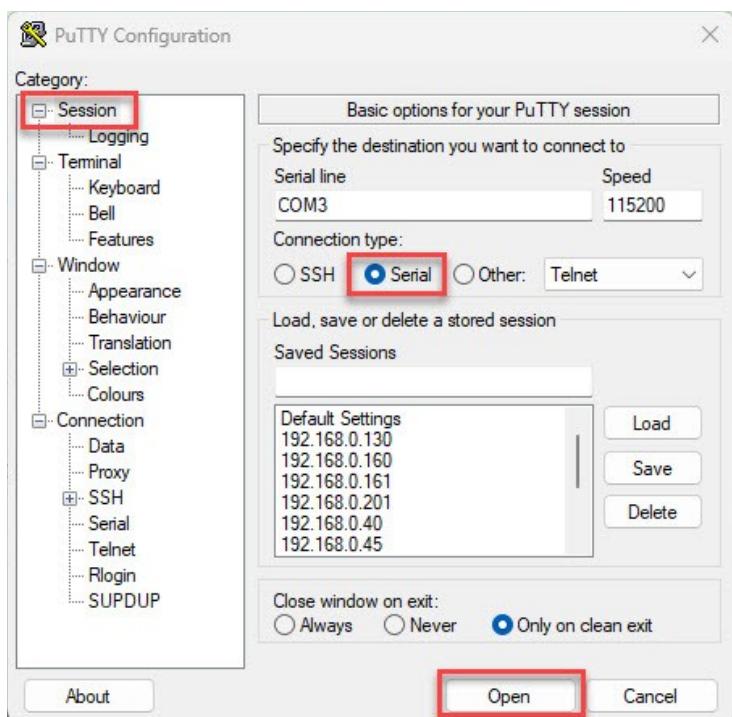


Рис. 2.16. Установка соединения

Откроется окно, в котором вам нужно нажать любую клавишу. Если все соединения и настройки были сделаны правильно, появится приглашение для ввода логина и пароля (рис. 2.17).

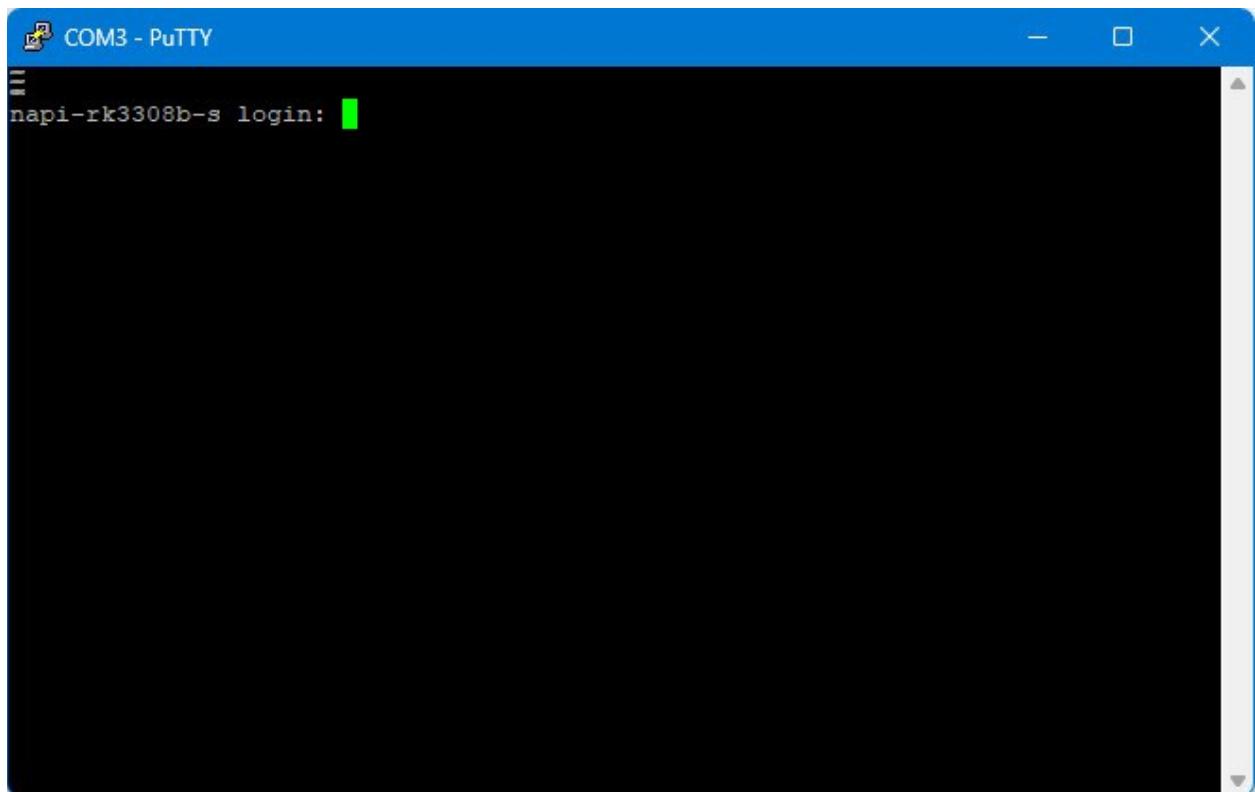


Рис. 2.17. Приглашение для ввода логина и пароля

Введите здесь логин root и пароль napilinux, установленные по умолчанию. В окне появится логотип NaPi и системное приглашение пользователя root (рис. 2.18).

```
COM3 - PuTTY
napi-rk3308b-s login: root
Password:

[ 346.048559] audit: type=1334 audit(1734091323.236:6): prog-id=9 op=LOAD
[ 346.057057] audit: type=1334 audit(1734091323.244:7): prog-id=10 op=LOAD
[ 347.327847] audit: type=1006 audit(1734091324.516:8): pid=512 uid=0 old-auid=
4294967295 auid=0 tty=(none) old-ses=4294967295 ses=1 res=1
[ 347.341750] audit: type=1300 audit(1734091324.516:8): arch=c00000b7 syscall=6
4 success=yes exit=1 a0=8 a1=7fe4f791b0 a2=1 a3=7f943c9460 items=0 ppid=1 pid=51
2 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=
1 comm="(systemd)" exe="/lib/systemd/systemd" key=(null)
[ 347.371029] audit: type=1327 audit(1734091324.516:8): proctitle="(systemd)"
[ 347.467371] audit: type=1334 audit(1734091324.656:9): prog-id=11 op=LOAD
[ 347.475076] audit: type=1300 audit(1734091324.656:9): arch=c00000b7 syscall=2
80 success=yes exit=8 a0=5 a1=7fd47caec8 a2=90 a3=0 items=0 ppid=1 pid=512 auid=
0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=1 comm=
"systemd" exe="/lib/systemd/systemd" key=(null)
[ 347.491752] systemd[512]: memfd_create() called without MFD_EXEC or MFD_NOEXE
C_SEAL set
[ 347.503209] audit: type=1327 audit(1734091324.656:9): proctitle="(systemd)"
[ 347.520281] audit: type=1334 audit(1734091324.656:10): prog-id=11 op=UNLOAD
[ 347.528303] audit: type=1300 audit(1734091324.656:10): arch=c00000b7 syscall=
57 success=yes exit=0 a0=8 a1=7f8ecf5460 a2=0 a3=7f8ecf5c20 items=0 ppid=1 pid=5
12 auid=0 uid=0 gid=0 euid=0 suid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=
1 comm="systemd" exe="/lib/systemd/systemd" key=(null)
root@napi-rk3308b-s:~#
```

Рис. 2.18. Системное приглашение пользователя root

Введите здесь команду ip a, и вы увидите текущий адрес IP, выделенный модулю FCC-0801 (рис. 2.19).

```

COM3 - PuTTY

[ 347.371029] audit: type=1327 audit(1734091324.516:8): proctitle="(systemd)"
[ 347.467371] audit: type=1334 audit(1734091324.656:9): prog-id=11 op=LOAD
[ 347.475076] audit: type=1300 audit(1734091324.656:9): arch=c00000b7 syscall=2
80 success=yes exit=8 a0=5 al=7fd47caec8 a2=90 a3=0 items=0 ppid=1 pid=512 auid=
0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=1 comm=
"systemd" exe="/lib/systemd/systemd" key=(null)
[ 347.491752] systemd[512]: memfd_create() called without MFD_EXEC or MFD_NOEXE
C_SEAL set
[ 347.503289] audit: type=1327 audit(1734091324.656:9): proctitle="(systemd)"
[ 347.520281] audit: type=1334 audit(1734091324.656:10): prog-id=11 op=UNLOAD
[ 347.528303] audit: type=1300 audit(1734091324.656:10): arch=c00000b7 syscall=1
57 success=yes exit=0 a0=8 a1=7f8ecf5460 a2=0 a3=7f8ecf5c20 items=0 ppid=1 pid=5
12 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses
=1 comm="systemd" exe="/lib/systemd/systemd" key=(null)
root@napi-rk3308b-s:~# [ 427.553215] rk_gmac-dwmac ff4e0000.ethernet end0: Link is Up - 100Mbps/Full - flow cont
rol rx/tx
[ 429.661349] IPv4: martian source 255.255.255.255 from 192.168.0.1, on dev end0
[ 429.669653] ll header: 00000000: 02 07 23 87 2d e5 78 96 82 74 6a 75 08 00
[ 429.701298] IPv4: martian source 255.255.255.255 from 192.168.0.1, on dev end0
[ 429.709480] ll header: 00000000: 02 07 23 87 2d e5 78 96 82 74 6a 75 08 00
[ 429.795102] kaudited_printk_skb: 7 callbacks suppressed
[ 429.795138] audit: type=1334 audit(1734091382.766:13): prog-id=13 op=LOAD
[ 429.795963] systemd-journald[229]: Time jumped backwards, rotating.
[ 429.801080] audit: type=1334 audit(1734091382.766:14): prog-id=14 op=LOAD
root@napi-rk3308b-s:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: end0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 02:07:23:87:2d:e5 brd ff:ff:ff:ff:ff:ff permaddr 82:f2:a0:4e:bb:b4
    inet 192.168.0.18/24 metric 10 brd 192.168.0.255 scope global dynamic end0
        valid_lft 86384sec preferred_lft 86384sec
root@napi-rk3308b-s:~#

```

Рис. 2.19. Просмотр адреса IP модуля FCC-0801

Работа с консолью в ОС Linux

В ОС Linux для работы с консолью можно использовать программу minicom.

Если ее нет в вашей ОС, установите minicom следующим образом:

```
# apt install minicom
```

Далее для подключения к модулю FCC-0801 в среде NapiLinux подготовьте файл конфигурации .minirc.napi-115200.minicom с таким содержимым:

```

pu rtscts          No
pu xonxoff         No
pu baudrate        115200

```

При запуске передавайте путь к этому файлу программе minicom в качестве параметра:

```
# minicom -D /dev/ttyS3 napi-115200.minicom
```

Устройство задается при помощи параметра -D.

Для Armbian используйте файл .minirc.napi-1M.minicom:

```

pu rtscts          No
pu xonxoff         No
pu baudrate        1500000

```

Команда запуска minicom для Armbian:

```
# minicom -D /dev/ttyS3 napi-1M.minicom
```

Назначение статического адреса IP для FCC-0801

Чтобы задать фиксированный адрес IP модулю FCC-0801 и другие сетевые параметры через консоль, нужно отредактировать файл /etc/systemd/network/20-wired.network. Ниже вы найдете его содержимое по умолчанию:

```
[Match]
Name=end0

[Network]
DHCP=ipv4
LinkLocalAddressing=ipv4
IPv6AcceptRA=no
IPv4LLStartAddress=169.254.100.100

[DHCP]
RouteMetric=10
ClientIdentifier=mac
SendHostname=yes
Hostname=napi-rk3308b-s
```

Для того чтобы настроить статический IP-адрес, шлюз и DNS-сервер, необходимо внести изменения в секцию **[Network]**.

Отключите DHCP и укажите необходимые параметры для статического IP-адреса, например:

```
[Match]
Name=end0

[Network]
DHCP=no
LinkLocalAddressing=no
IPv6AcceptRA=no
Address=192.168.0.50/24
Gateway=192.168.0.1
DNS=192.168.0.1

[DHCP]
RouteMetric=10
ClientIdentifier=mac
```

```
SendHostname=yes  
Hostname=napi-rk3308b-s
```

Здесь для отключения DHCP используется строка «DHCP=no». Параметр «LinkLocalAddressing=no» отключает использование локальных IPv4-адресов.

Статический адрес IP устанавливается строкой «Address=192.168.0.18/24». Что касается адреса шлюза и DNS-сервера, то они указываются в параметрах Gateway и DNS.

После внесения этих изменений необходимо перезапустить сетевые службы:

```
# systemctl restart systemd-networkd
```

Начальная настройка общих параметров FCC-0801

После того как вы определили адрес IP модуля FCC-0801, нужно выполнить настройку его параметров. Для этого подключитесь к Веб-интерфейсу NapiConfig модуля по адресу вида <https://192.168.0.18/>, где вместо 192.168.0.18 нужно указать адрес IP модуля FCC-0801.

При подключении укажите логин admin и пароль admin. В процессе настройки вам нужно будет изменить этот пароль.

На рис. 3.1 показано главное окно Веб-интерфейса NapiConfig.

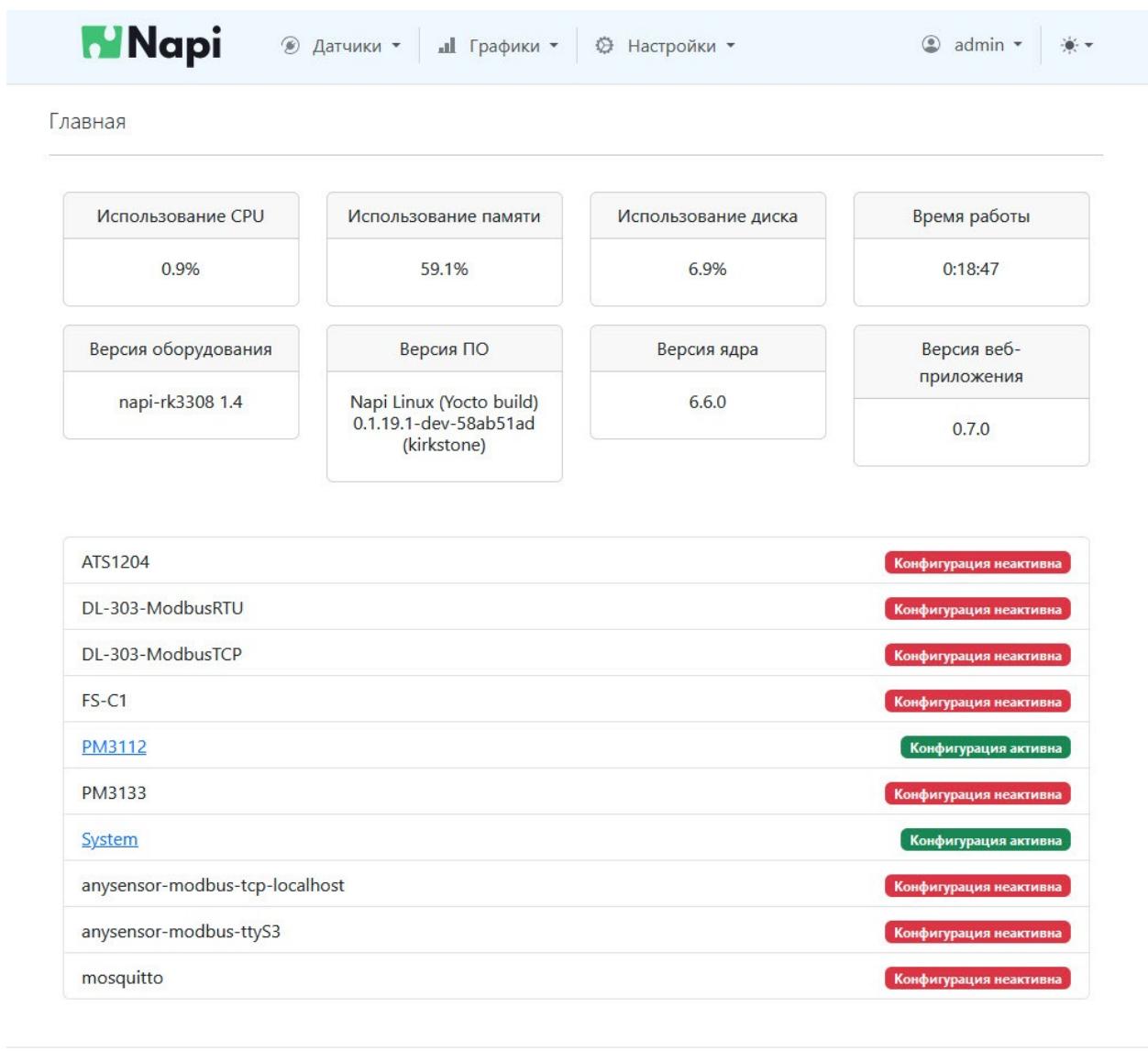


Рис. 3.1. Главное окно Веб-интерфейса настройки NapiConfig

Настройка общих параметров

Прежде всего нужно познакомиться с настройкой общих параметров. Для просмотра и редактирования общих параметров, а также для управления FCC-0801 выберите из меню **Настройки** Веб-интерфейса NapiConfig строку **Общие** (рис. 3.2).

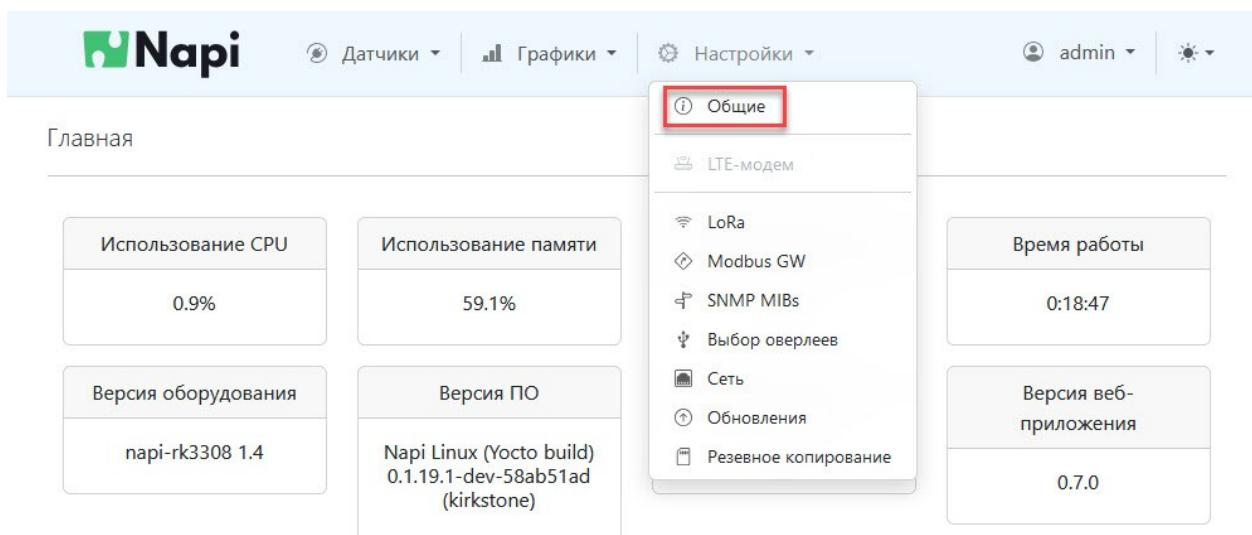


Рис. 3.2. Общие настройки модуля FCC-0801

Вы увидите страницу с блоками **Система**, **Дата и время**, а также **Системные службы**.

В блоке **Система** (рис. 3.3) показаны:

- загрузка процессора **CPU avg**;
- использование оперативной памяти (**RAM**, **Mem** и **Swap**);
- использование диска SD, EMMC, NAND (**Disk size**, **Used** и **Avail**);
- размер базы данных **Database size**, полученных от сенсоров (датчиков)

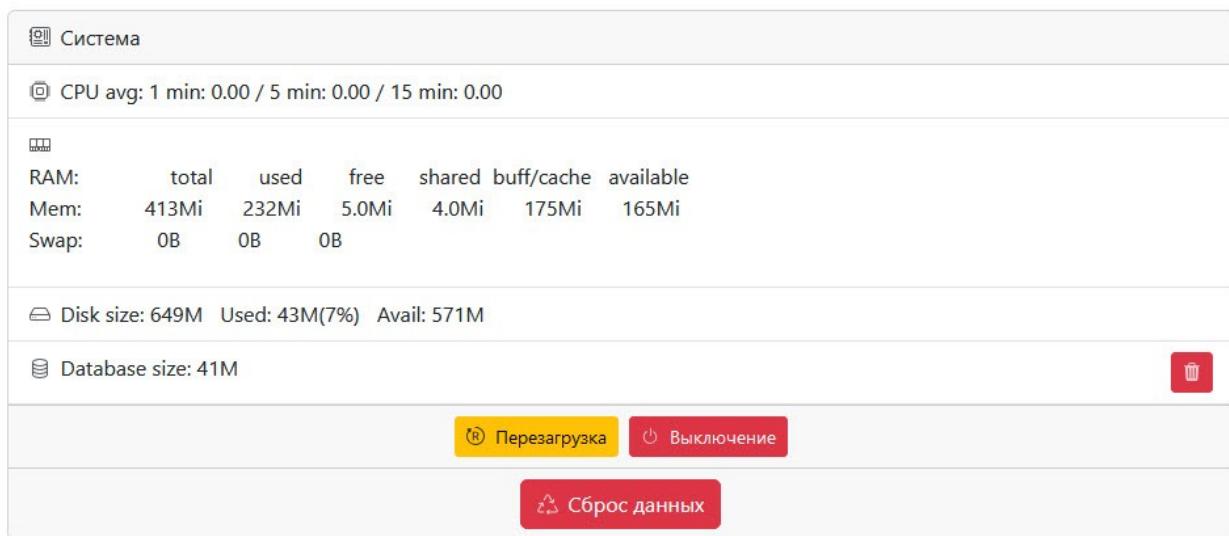


Рис. 3.3. Системные параметры и управление модулем

Кнопки **Перезагрузка** и **Выключение** предназначены для перезагрузки операционной системы (ОС) модуля, а также для его выключения, соответственно.

Справа от поля **Database size** расположена кнопка с изображением мусорной корзины. С ее помощью можно полностью удалить текущее содержимое базы данных модуля.

Кнопка **Сброс данных** предназначена для сброса настроек в состояние по умолчанию.

Далее в процессе начальной настройки проверьте содержимое блока **Дата и время** (рис. 3.4).



Рис. 3.4. Установка даты и времени

Проверьте, что в поле **Текущее время** отображается верная дата и время. Если это не так, щелкните кнопку с изображением часов. На экране появится окно с запросом подтверждения коррекции времени (рис. 3.5).

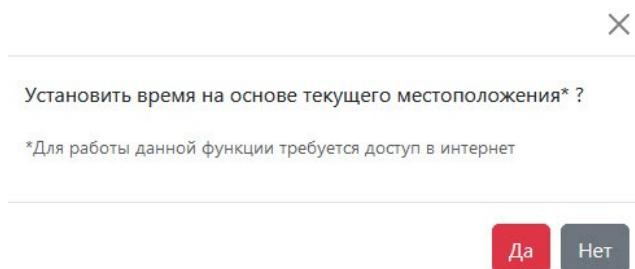


Рис. 3.5. Установка даты и времени

Если модуль подключен к сети с выходом в интернет, для коррекции даты и времени щелкните кнопку **Да**.

Если ваша сеть не подключена к интернету, используйте для просмотра и установки даты команду date из консоли:

```
# date  
Sat Dec 21 12:23:56 PM MSK 2024
```

Установить дату и время командой date можно следующим образом:

```
# date -s "2024-12-21 12:25:00"  
Sat Dec 21 12:25:00 PM MSK 2024
```

В поле **Время работы** можно узнать, сколько времени работала ОС модуля с момента предыдущей перезагрузки.

После проверки времени перейдите к блоку **Системные службы**, расположенному в самом низу окна настройки общих параметров (рис. 3.6).

Системные службы	Active	Enabled	Log
systemd-networkd	Active	Enabled	Log
systemd-resolved	Active	Enabled	Log
telegraf	Active	Enabled	Log
mosquitto	Active	Enabled	Log
grafana-server	Inactive	Disabled	Log
influxdb	Active	Enabled	Log
swupdate	Active	Enabled	Log
mbusd	Inactive	Disabled	Log
snmpd	Inactive	Disabled	Log

Рис. 3.6. Управление системными службами

Вы можете управлять следующими сервисами. Использовать или не использовать сервис зависит от решаемой задачи.

- systemd-networkd
- systemd-resolved
- telegraf
- mosquitto
- influxdb
- swupdate

Соответствующие этим службам кнопки должны быть зеленого цвета с надписями **Active** (запущены) и **Enabled** (будут запущены автоматически при перезагрузке ОС модуля).

Убедитесь, что у вас работают службы telegraf и influxdb. Это необходимо для работы с датчиками.

Если щелкнуть синюю кнопку **Log**, можно увидеть содержимое системного журнала. Такая информация может потребоваться сервисной службе для диагностики причин появления каких-либо проблем.

При дальнейшей настройке устройства вам, возможно, потребуется включать другие службы. Для этого воспользуйтесь кнопкой **Inactive** в соответствующей строке. Если служба должна запускаться автоматически при перезапуске ОС модуля FCC-0801 щелкните кнопку **Disabled**.

Настройка сетевых параметров

По умолчанию модуль FrontControl Compact пытается получить сетевые параметры от сервера DHCP, если он есть в сети. В частности, он получит динамический адрес IP, который со временем может измениться.

Лучше всего назначить модулю статический адрес IP, по которому он всегда будет доступен. Проще всего это сделать при помощи web-интерфейса NapiConfig, выбрав из меню **Настройки** строку **Сеть** (рис. 3.7).

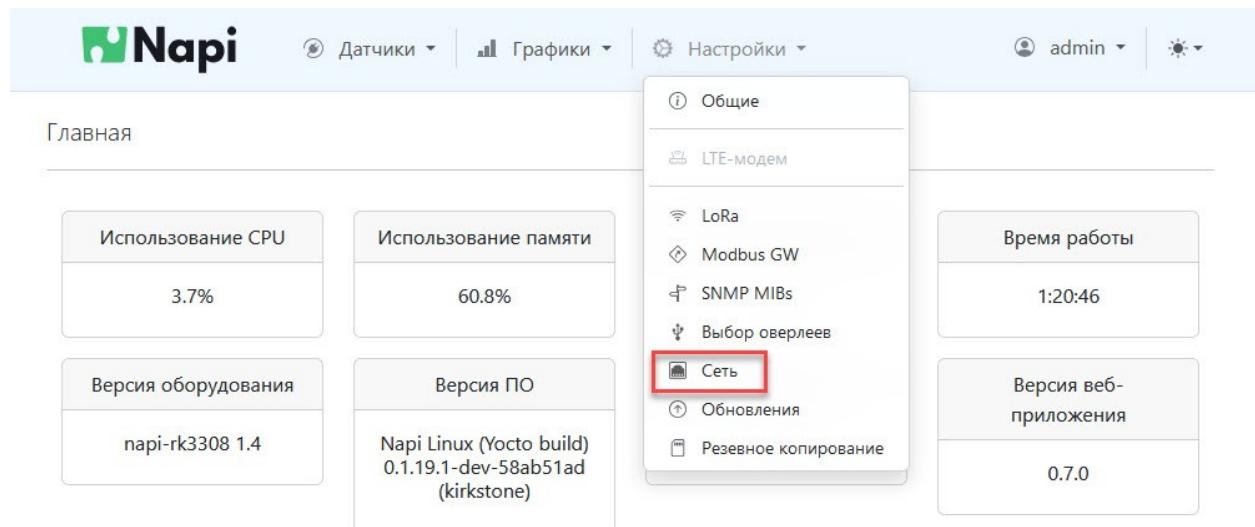


Рис. 3.7. Стока Сеть для настройки сетевых параметров

Страница настройки сетевых параметров показана на рис. 3.8.

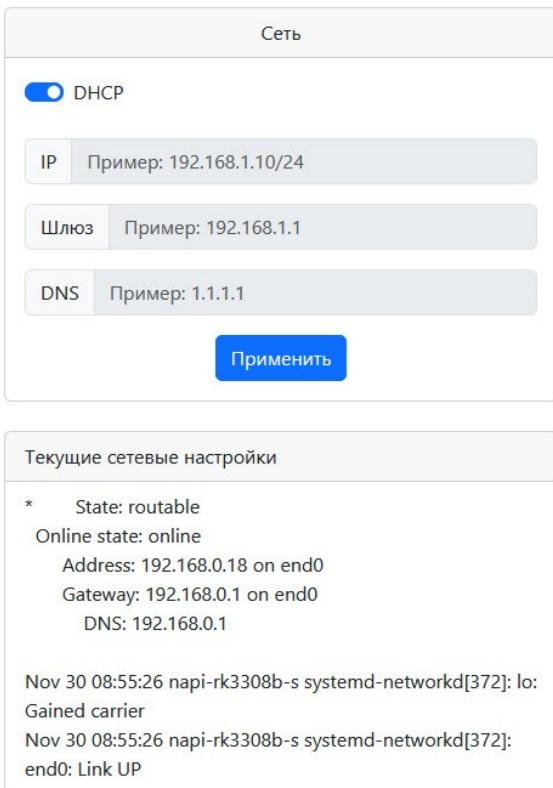


Рис. 3.8. Настройка сетевых параметров

Чтобы задать статический адрес IP, выключите переключатель **DHCP**, а затем задайте нужный адрес IP, адрес шлюза и адрес сервера DNS в полях **IP**, **Шлюз** и **DNS**, соответственно.

Для сохранения изменений воспользуйтесь кнопкой **Применить**.

В нижней части страницы настройки сетевых параметров отображаются текущие сетевые настройки.

Обращаем ваше внимание, что **сетевые параметры должны быть настроены без ошибок**, иначе доступ к устройству будет возможен только через консоль. Лучше всего перед изменением параметров получить консультацию у вашего системного или сетевого администратора.

Если при изменении сетевых параметров были допущены ошибки и доступ к модулю FrontControl Compact пропал, воспользуйтесь инструкцией по получению доступа через консоль из раздела **2.3 Определение адреса узла FrontControl Compact без DHCP**.

Активация датчика System

После того как вы настроили сетевые параметры, перейдите в web-интерфейс NapiConfig для проверки активации датчика System.

Выберите из меню **Датчики** строку **Управление**. На странице управления конфигурациями вы увидите список шаблонов для различных датчиков (рис. 3.9).

Загруженные шаблоны конфигурации:

	<input type="button" value="Написать шаблон"/>	<input type="button" value="Загрузить шаблон"/>
1. PM3112 : ICPDAS current sensor https://www.icpdas.com/web/product/download/energy_management/power_meter/document/manual/PM-3112_PM-3114_en.pdf	<input type="button" value="Активировать"/>	⋮
2. DL-303-ModbusRTU : ICPDAS DL-303 (CO\CO2\Temp\Humidity) Modbus RTU	<input type="button" value="Активировать"/>	⋮
3. DL-303-ModbusTCP : DL-303 (CO,CO2,temp,humidity) ModbusTCP	<input type="button" value="Активировать"/>	⋮
4. System : Сбор системных показателей этого устройства (НЕ ИЗМЕНЯТЬ!)	<input type="button" value="Отключить"/> <input type="button" value="Редактировать"/>	⋮
5. ATS1204 : Пример шаблона для сбора данных с устройства через SNMP	<input type="button" value="Активировать"/>	⋮
6. FS-C1 : Шаблон для встроенного датчика тока (Пример modbus RTU)	<input type="button" value="Активировать"/>	⋮
7. PM3133 : 3-фазный измеритель напряжения и тока (пример modbus TCP)	<input type="button" value="Активировать"/>	⋮
8. anysensor-modbus-ttyS3 : Any sensor with address=1. hold-register=0 (first) on /dev/ttyS3 (FCC device) with 9600\8n1 (sensor parameter)	<input type="button" value="Активировать"/>	⋮
9. anysensor-modbus-tcp-localhost : Any sensor with address=1. hold-register=0 (first) on localhost	<input type="button" value="Активировать"/>	⋮
10. mosquitto : Конфигурация брокера сообщений MQTT, на который будут отправляться метрики	<input type="button" value="Активировать"/>	⋮

Рис. 3.9. Список шаблонов для датчиков

Убедитесь, что у вас включен датчик **System**, предназначенный для сбора данных самого устройства FCC-0801. Не редактируйте его конфигурацию.

○ Просмотр графика нагрузки системы

После активации датчика **System** подождите пару минут и откройте в web-интерфейсе NapiConfig раздел просмотра графиков, выбрав из меню **Графики** строку **Системные**.

На странице системных графиков можно выбрать период просмотра от 1 минуты до 30 дней. В верхней части страницы находится график нагрузки процессора (рис. 3.10).

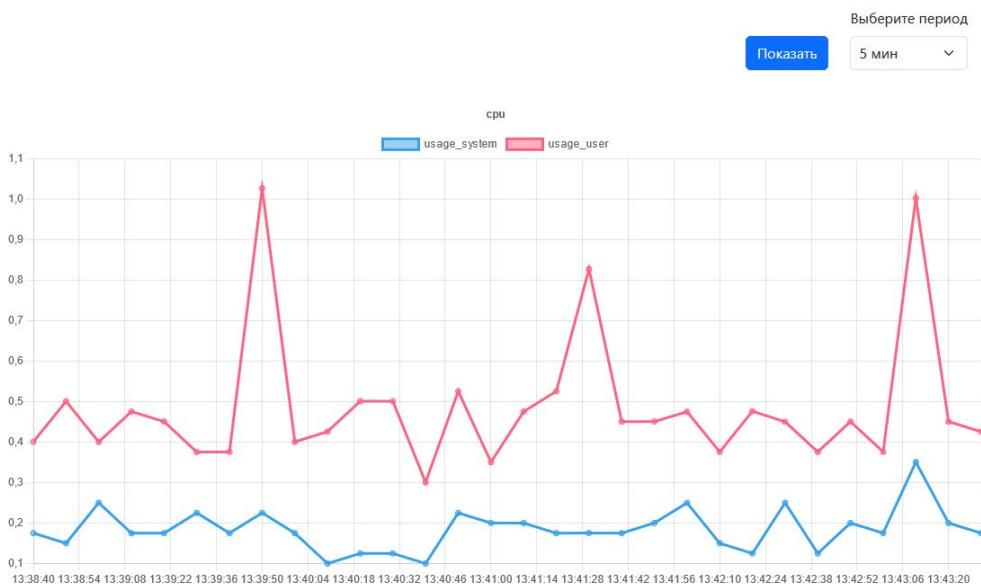


Рис. 3.10. График нагрузки процессора

Ниже вы увидите график использования дискового пространства (рис. 3.11), а также график использования памяти (рис. 3.12).

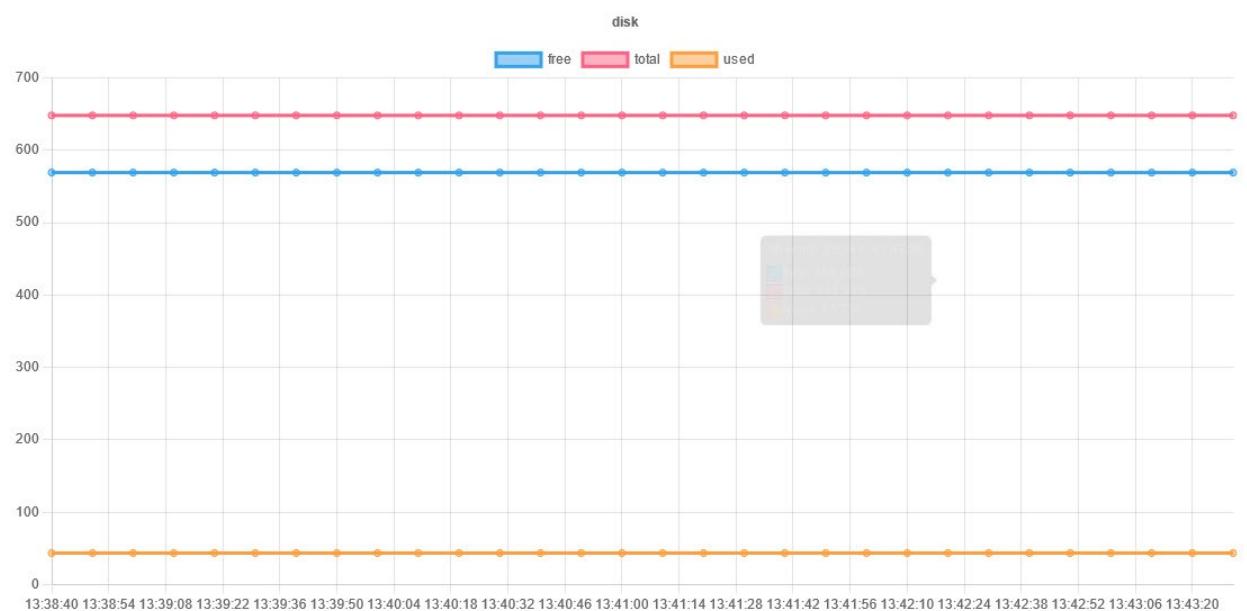


Рис. 3.11. График использования дискового пространства

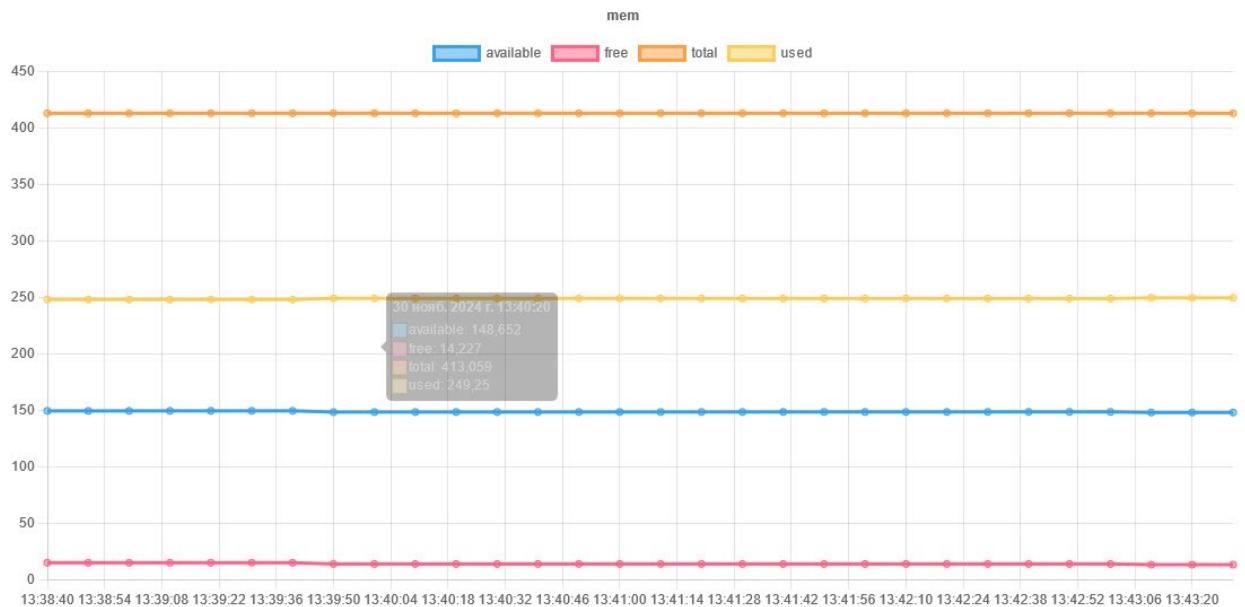


Рис. 3.12. График использования памяти

Просмотр графиков для выбранных датчиков

С помощью строки **Сенсоры** меню **Графики** можно просматривать графики для выбранных датчиков за определенный период времени (рис. 3.13).

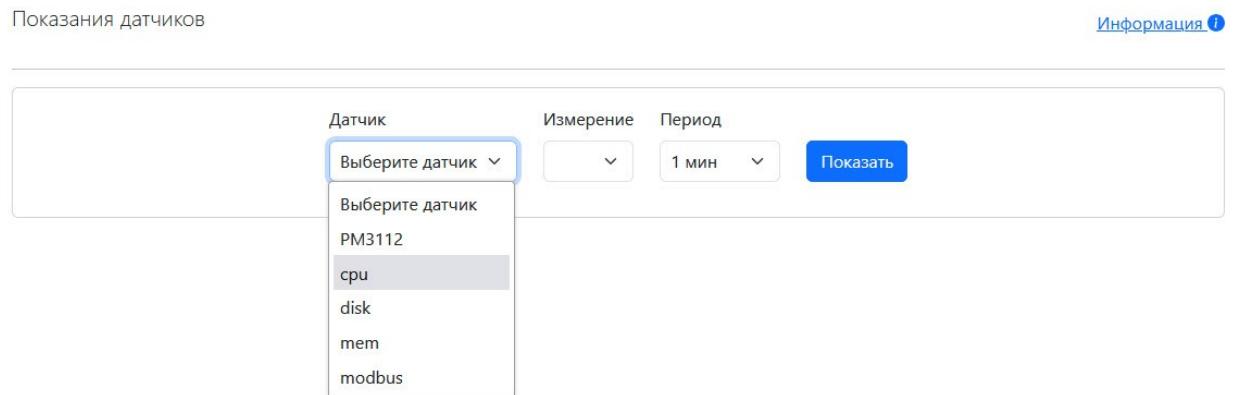


Рис. 3.13. Выбор датчика для просмотра графика

Для каждого датчика из списка **Датчик** можно выбрать нужное измерение, а также задать период времени для построения графика (рис. 3.14).

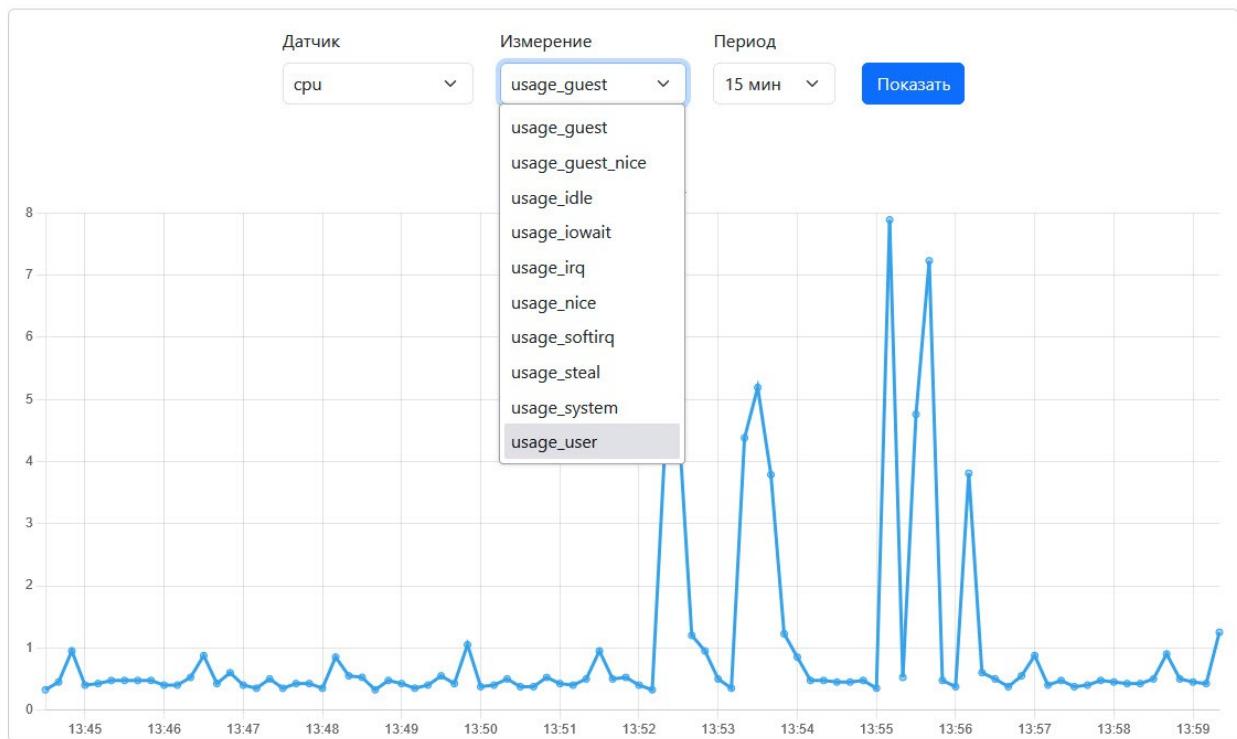


Рис. 14. Выбор измерения usage_user для датчика сри за 15 минут

По мере подключения различных датчиков будет доступно больше графиков.

Дополнительно можно подключить мониторинг с помощью Grafana. Хотя вы можете запустить службу grafana-server на узле FrontControl Compact, лучше использовать для этого отдельный сервер или виртуальную машину.

Инструкции по запуску Grafana вы найдете в разделе **5. Установка и запуск Grafana**.

Подключение и мониторинг датчиков

В этом разделе приведена общая информация об архитектуре и функциях модуля FrontControl Compact, а также руководство по настройке подключения к модулю различных датчиков.

Архитектура работы с датчиками модуля FCC-0801

Модуль позволяет получать информацию от сенсоров (датчиков) через RS485 по протоколу Modbus RTU, а также через сеть Ethernet по протоколу Modbus TCP. Полученная информация сохраняется в базе данных InfluxDB2 модуля FCC-0801 и может отображаться в виде графиков с помощью web-интерфейса NapiConfig, а также Grafana (рис. 4.1).

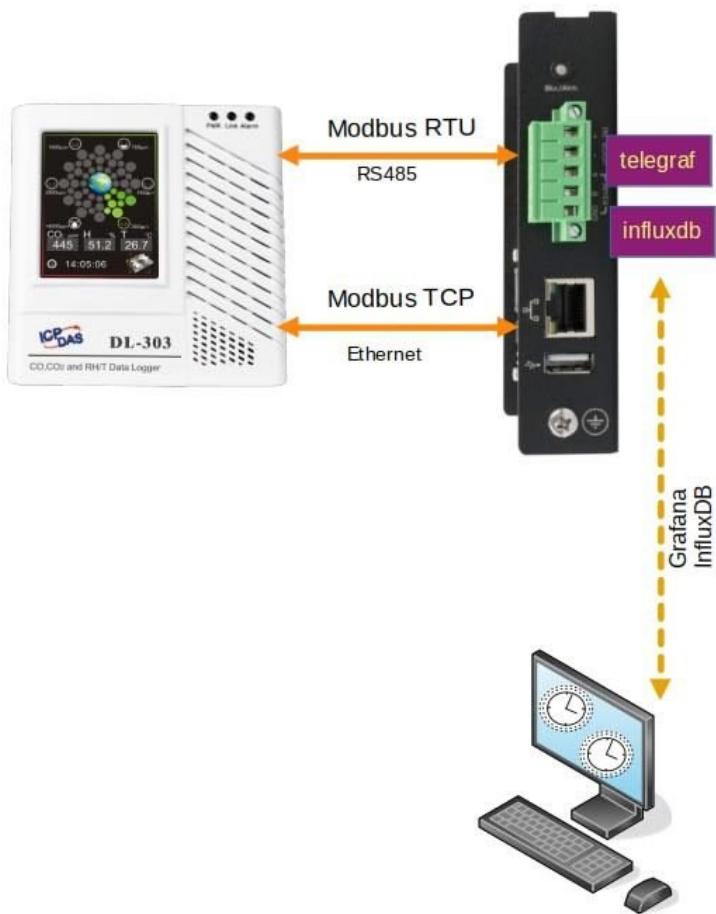


Рис. 4.1. Получение данных от датчиков и передача ее в InfluxDB2 и Grafana

На базе FCC-0801 можно сделать шлюз Modbus RTU-Modbus TCP. Он работает следующим образом:

- опрос от пользователей идет на модуль сборщика по IP этого модуля с помощью протокола Modbus TCP;
- сборщик преобразует запрос в Modbus RTU, опрашивает датчики по RS485 и возвращает ответы по Modbus TCP пользователю
-

Упрощенная архитектура ПО работы с датчиками модуля FCC-0801

Упрощенная архитектура ПО модуля FCC-0801 показана на рис. 4.2.

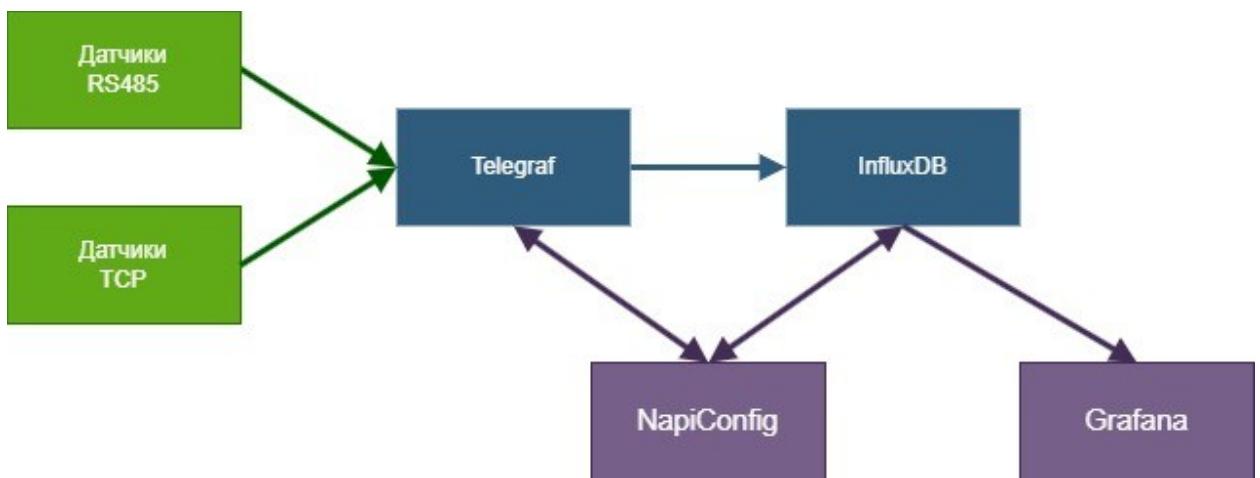


Рис. 4.2. Упрощенная архитектура ПО модуля FCC-0801

Данные, поступающие от сенсоров, подключенных к модулю FrontControl Compact через RS485 или Ethernet, собираются агентом Telegraf и сохраняются в базе данных InfluxDB.

Агент **Telegraf**, разработанный компанией InfluxData, специально предназначен для сбора данных из различных источников. Это могут быть не только датчики, но и облачные сервисы, системы мониторинга, хостинговые серверы, размещенные в data-центрах и различные приложения.

Для подключения Telegraf к источникам данных разработано свыше 200 входных плагинов. Собранные данные передаются в различные системы хранения и анализа также с помощью плагинов, которые называются выходными. В Telegraf имеется свыше 40 выходных плагинов.

В архитектуре, показанной на рис. 4.2, данные передаются из Telegraf в базу данных **InfluxDB** через готовый выходной плагин.

При необходимости вы можете настраивать существующие плагины, а также создавать собственные входные и выходные плагины, расширяя набор источников и потребителей данных.

Высокопроизводительная база данных временных рядов InfluxDB создана для хранения и анализа больших объемов данных, генерируемых в реальном времени. Она отлично подходит для работы с метриками, событиями и другими временными данными. В InfluxDB предусмотрены мощные инструменты для запросов и визуализации временных рядов.

Приложение **NapiConfig**, доступное через web-интерфейс, позволяет настраивать параметры работы системного ПО модуля FrontControl Compact, редактировать конфигурации входных плагинов, добавляя нужные датчики, а также просматривать графики изменения данных, полученных от этих датчиков. Это приложение исключает необходимость редактирования через командную строку Linux файлов

конфигурации ОС, программ Telegraf и InfluxDB, значительно упрощая работу с модулем FrontControl Compact.

Хотя графики изменения данных от датчиков можно просматривать в графическом виде средствами NapiConfig или InfluxDB, можно подключить внешние системы отображения и мониторинга, такие как Grafana.

Платформа **Grafana** предназначена для визуализации и мониторинга данных. Она позволяет создавать интерактивные и настраиваемые дашборды и применяется для мониторинга производительности систем, анализа данных и принятия решений на основе визуализированной информации.

В ходе настройки ПО FrontControl Compact можно активировать и другие службы (на рис. 2.4 не показаны):

- mosquitto
- swupdate
- mbusd
- snmpd

Легковесный брокер сообщений Mosquitto реализует протокол MQTT (Message Queuing Telemetry Transport). Он используется для передачи сообщений между устройствами в сетях с низкой пропускной способностью, таких как IoT (Интернет вещей). Mosquitto позволяет устройствам публиковать и подписываться на сообщения, обеспечивая надежную и эффективную передачу данных.

Служба SWUpdate должна быть уже активирована в процессе начальной настройки, чтобы работало обновление прошивки ПО FrontControl Compact.

Для того чтобы использовать модуль FrontControl Compact в качестве шлюза Modbus необходимо активировать службу mbusd.

И, наконец, служба snmpd (Simple Network Management Protocol Daemon) реализует SNMP, нужный для мониторинга и управления сетевыми устройствами (маршрутизаторы, коммутаторы, серверы).

Шаблоны плагинов Telegraf

Для сбора данных от датчиков нужно установить с помощью web-интерфейса NapiConfig необходимые плагины и при необходимости отредактировать из шаблоны.

В разделе **3.3. Активация датчика System** было рассказано как проверить активацию и при необходимости активировать датчик System. Этот датчик

предназначен для сбора данных самого устройства FCC-0801 и для него используется предустановленный плагин.

Чтобы открыть страницу управления плагинами датчиков (рис. 3.9), выберите из меню **Датчики** строку **Управление**. Сразу после начальной настройки модуля FrontControl Compact активирован только один датчик System.

Вы можете просмотреть любой шаблон, щелкните кнопку с изображением трех расположенных вертикально точек, а затем выберите из меню строку **Просмотреть** (рис. 4.3).

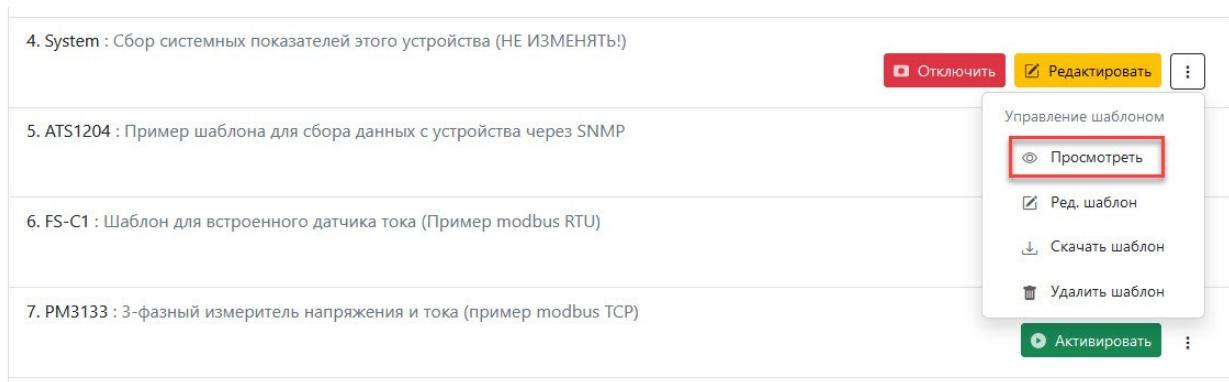


Рис. 4.3. Стока меню для просмотра шаблона датчика

Шаблон System, предназначенный для передачи данных узла модуля FCC-0801 в Telegraf, представлен ниже:

```
[[inputs.cpu]]  
  perccpu = false  
  totalcpu = true  
  collect_cpu_time = false  
  report_active = false  
  core_tags = false  
[[inputs.mem]]  
[[inputs.disk]]  
  mount_points = ["/"]
```

Этот шаблон задает конфигурацию агента Telegraf, собирающего метрики системы — данные о производительности и состоянии системы.

Заметим, что хотя вы можете отредактировать этот шаблон с помощью строки **Ред. шаблон**, делать это **не рекомендуется**, чтобы не нарушить сбор данных о загрузке процессора, память и диска узла модуля FrontControl Compact.

Строка `[[inputs.cpu]]` задает начало блока конфигурации для сбора метрик CPU. При этом наличие слова `inputs` говорит о том, что данный шаблон задает блок конфигурации плагина, предназначенный для сбора данных. Так обозначается конфигурация входного плагина.

Есть также выходные плагины, обозначаемые как outputs. Например, предустановленный плагин, конфигурация которого начинается со строки [[outputs.influxdb_v2]], предназначен для отправки собранных данных метрики в базу данных InfluxDB.

Возвращаясь ко входному плагину [[inputs.cpu]], перечислим определенные для него параметры:

- perccpu = false — метрики CPU не будут собираться для каждого отдельного ядра процессора;
- totalcpu = true — метрики CPU будут собирааться для всех ядер процессора в совокупности;
- collect_cputime = false — не будет собираться время, проведенное CPU в различных состояниях (например, пользовательское время, системное время);
- report_active = false — не будет собираться активное время CPU (время, когда CPU не был в состоянии простоя);
- core_tags = false — метки для каждого ядра процессора не будут добавлены к метрикам.

Секция [[inputs.mem]] указывает, что будут собираться метрики использования памяти. Так как здесь не указаны дополнительные параметры, будут использованы значения по умолчанию.

Аналогично, секция [[inputs.disk]] с параметром mount_points = ["/"] указывает, что будут собираться метрики диска только для корневой файловой системы.

Выходной плагин для передачи данных из Telegraf в базу данных InfluxDB [[outputs.influxdb_v2]] уже установлен и его не надо настраивать:

```
[ [outputs.influxdb_v2]
  ## The URLs of the InfluxDB cluster nodes.
  urls = ["http://localhost:8086"]
  ## Token for authentication.
  token = "*****"
  ## Organization is the name of the organization you wish to write to.
  organization = "nnz"
  ## Destination bucket to write into.
  bucket = "bucket1"
  ## HTTP User-Agent
  user agent = "telegraf"
```

Ниже мы привели краткое описание параметров плагина:

- urls — URL-адреса узлов кластера InfluxDB;
- token — токен для аутентификации;
- organization — название организации для записи данных;
- bucket — название контейнера для записи данных;

- `user_agent` — строка HTTP User-Agent

В нашем случае плагин `outputs.influxdb_v2` сконфигурирован для работы с одним узлом базы данных InfluxDB с таким адресом URL — <http://localhost:8086>.

Параметр `token` нужен для получения доступа к сервису базы данных.

С помощью параметра `organization` можно назначать различный доступ для разных организаций и отделов организаций.

Контейнеры InfluxDB представляют собой именованные хранилища для данных временных рядов. С их помощью можно объединять данные, политики хранения и индексацию для разных групп датчиков, журналов приложений и метрик производительности.

И, наконец, параметр `user_agent` используется для идентификации клиента, который отправляет запрос. В нашем случае он задан как `telegraf`.

Таким образом, теперь вы знаете, что Telegraf использует входные и выходные плагины для получения и отправки данных, соответственно. Чтобы подключить датчики и получать с них информацию, нужно добавить и настроить шаблоны соответствующих плагинов.

Как найти нужные плагины Telegraf

Создавая систему сбора данных, обычно вы уже представляете себе, какие в нее будут входить датчики. Зная названия этих датчиков, можно найти и настроить необходимые плагины.

В web-приложении NapiConfig уже есть несколько предустановленных входных плагинов Telegraf, которые, однако, нужно будет настроить.

Чтобы увидеть список таких плагинов, выберите в меню **Датчики** строку **Управление**. Появится страница со списком плагинов, показанная ранее на рис. 3.9.

Здесь вы найдете такие датчики:

- PM3112 — измеритель тока;
- DL-303-ModbusRTU — измеритель CO, CO2, температуры и влажности, подключенный через RS485 по протоколу ModbusRTU;
- DL-303-ModbusTCP — измеритель CO, CO2, температуры и влажности, подключенный через Ethernet по протоколу ModbusTCP;
- ATS1204 — устройство автоматического ввода резерва, работа через SNMP;
- FS-C1 — встроенный датчик тока, встроенный платы "Токо-сборщик"
- PM3133 — 3-фазный измеритель напряжения и тока;

- `anysensor-modbus-ttyS3` — пример шаблона для датчика, подключенного через RS485 по протоколу ModbusRTU;
- `anysensor-modbus-tcp-localhost` — пример шаблона для датчика, подключенного через Ethernet по протоколу ModbusTCP

Вы можете поискать шаблоны нужного вам датчика на сайте <https://github.com/lab240/telegraf-grafana-configs> в каталоге `conf-telegraf` (рис. 4.4).

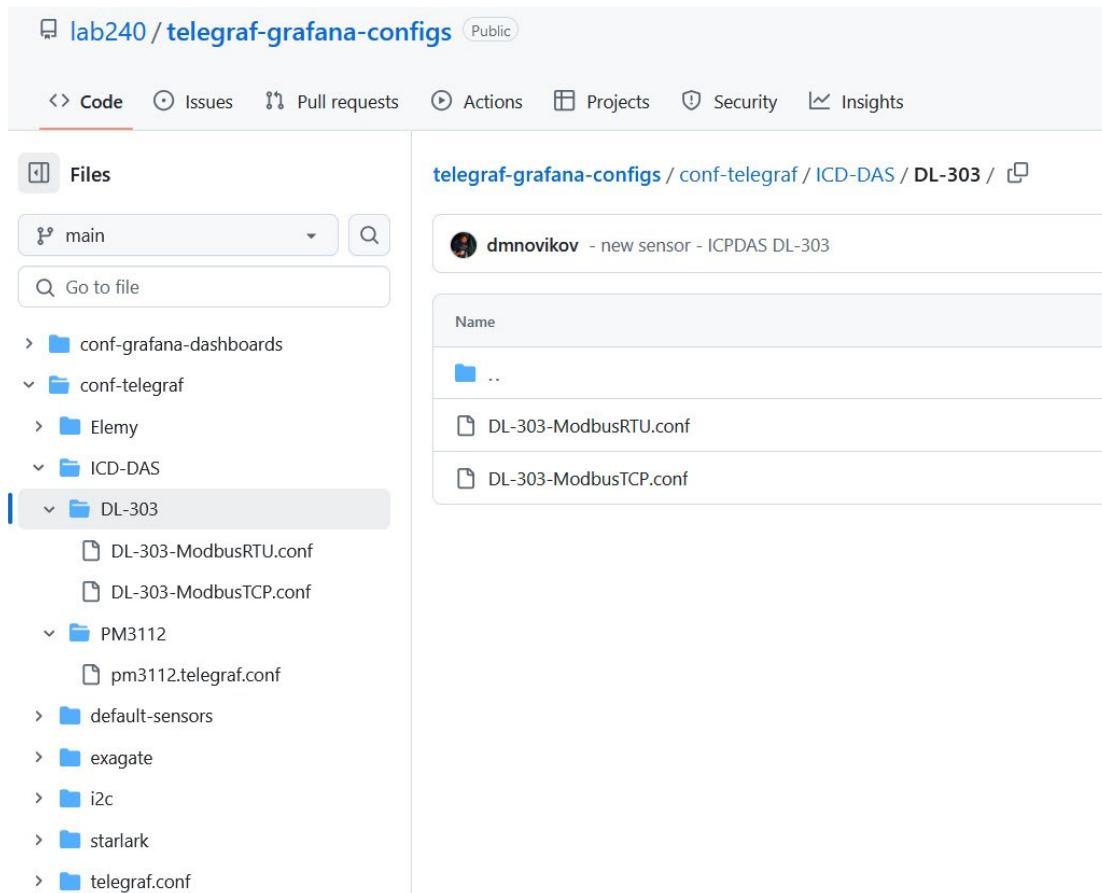


Рис. 4.4. Каталог с шаблонами датчиков

Кстати, на этом же сайте есть каталог файлов конфигураций панелей мониторинга (дашбордов) этих же датчиков для Grafana.

Очень большая коллекция шаблонов представлена на сайте <https://docs.influxdata.com/telegraf/v1/plugins/#input-plugins> (рис. 4.5).

The screenshot shows the InfluxData Documentation website for the Telegraf agent. The top navigation bar includes the InfluxData logo, a search bar, and a dropdown menu set to 'Telegraf'. The main content area has a title 'Plugin directory' in red. Below it, a text block describes Telegraf as a plugin-driven agent for collecting, processing, and writing metrics. It lists four categories of plugins: input, output, aggregator, and processor, noting that external plugins can integrate with the Telegraf Execd processor plugin. The page features two tables: one for 'Plugin type' (Input, Output, Aggregator, Processor, External) and another for 'Plugin category' (Applications, Build & Deploy, Cloud, Containers, Data Stores, IoT, Logging, Messaging, Networking, Servers, Systems, Web). A third table shows 'Operating system' (Linux, macOS, Windows) and 'Status' (New, Deprecated). At the bottom, there's a 'Jump to:' section with links to Input, Output, Aggregator, and Processor plugins.

Plugin type	Plugin category
<input type="checkbox"/> Input(253)	<input type="checkbox"/> Applications(32)
<input type="checkbox"/> Output(62)	<input type="checkbox"/> Build & Deploy(10)
<input type="checkbox"/> Aggregator(9)	<input type="checkbox"/> Cloud(32)
<input type="checkbox"/> Processor(33)	<input type="checkbox"/> Containers(11)
<input type="checkbox"/> External(20)	<input type="checkbox"/> Data Stores(40)
	<input type="checkbox"/> IoT(15)
	<input type="checkbox"/> Logging(15)
	<input type="checkbox"/> Messaging(26)
	<input type="checkbox"/> Networking(50)
	<input type="checkbox"/> Servers(28)
	<input type="checkbox"/> Systems(65)
	<input type="checkbox"/> Web(29)

Operating system	Status
<input type="checkbox"/> Linux(328)	<input type="checkbox"/> New(3)
<input type="checkbox"/> macOS(311)	<input type="checkbox"/> Deprecated(3)
<input type="checkbox"/> Windows(313)	

Jump to:

- [Input plugins](#)
- [Output plugins](#)
- [Aggregator plugins](#)
- [Processor plugins](#)

Рис. 4.5. Каталог плагинов на сайте InfluxData

С помощью формы поиска найдите нужный плагин, а затем щелкните кнопку **View**, расположенную напротив его названия (рис. 4.6). После этого вы попадете в соответствующий раздел Github, где найдете все необходимые инструкции.

Interrupts

 View

Plugin ID: `inputs.interrupts`

Telegraf 1.3.0+

The Interrupts input plugin gathers metrics about IRQs, including `interrupts` (from `/proc/interrupts`) and `soft_interrupts` (from `/proc/softirqs`).

IPMI Sensor

 View

Plugin ID: `inputs.ipmi_sensor`

Telegraf 0.12.0+

The IPMI Sensor input plugin queries the local machine or remote host sensor statistics using the `ipmitool` utility.

Ipset

 View

Plugin ID: `inputs.ipset`

Telegraf 1.6.0+

The Ipset input plugin gathers packets and bytes counters from Linux `ipset`. It uses the output of the command `ipset save`. Ipsets created without the `counters` option are ignored.

Рис. 4.6. Фрагмент списка плагинов

Еще один источник для поиска шаблонов плагинов находится по адресу <https://sensor.napilinux.ru>.

Шаблоны датчиков Modbus RTU

Когда вы используете готовый плагин для датчика, то, возможно, вам подойдет шаблон, уже встроенный в модуль FCC-0801, и его остается только отредактировать и активировать. Разумеется, предварительно нужно подключить датчик.

Откройте web-приложение NapiConfig, и выберите в меню **Датчики** строку **Управление**. Найдите нужный вам плагин, щелкните кнопку с тремя вертикальными точками и выберите строку **Просмотреть**. На рис. 4.7 мы показали это меню для датчика DL-303-ModbusRTU.

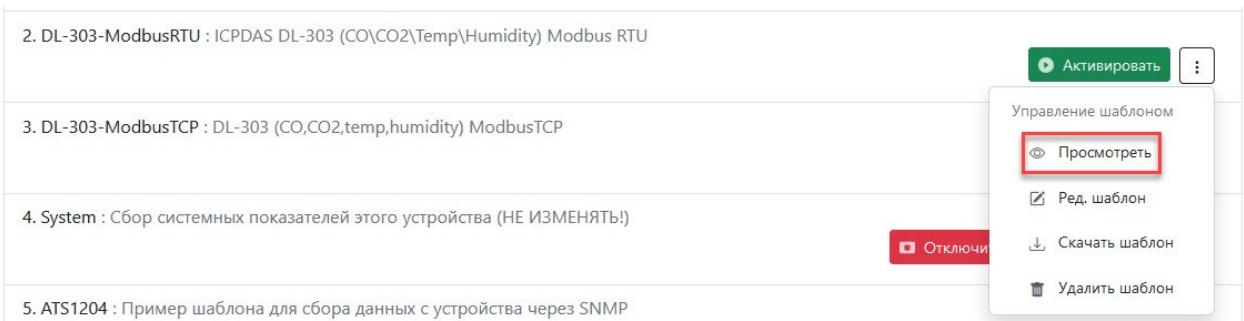


Рис. 4.7. Меню управления плагином для датчика DL-303-ModbusRTU

После выбора этой строки вы увидите текст шаблона соответствующего плагина:

```
## ICPDAS DL-303 (CO\CO2\Temp\Humidity) Modbus RTU
[[inputs.modbus]]
name = "DL-303-rtu"
slave_id = 1
timeout = "10s"
# busy_retries = 0
# busy_retries_wait = "100ms"
controller = "file:///dev/ttyS3"
baud_rate = 9600
data_bits = 8
parity = "N"
stop_bits = 1
transmission_mode = "RTU"
input_registers =
{
    measurement="DL-303-rtu", name = "CO", byte_order = "AB",
    data_type = "UINT16", scale=1.0, address = [0],
    {
        measurement="DL-303-rtu", name = "CO2", byte_order = "AB",
        data_type = "UINT16", scale=1.0, address = [1],
        {
            measurement="DL-303-rtu", name = "Humidity", byte_order = "AB",
            data_type = "UINT16", scale=1.0, address = [2],
            {
                measurement="DL-303-rtu", name = "Temp", byte_order = "AB",
                data_type = "UINT16", scale=1.0, address = [3],
                {
                    measurement="DL-303-rtu", name = "DewPoint", byte_order = "AB",
                    data_type = "UINT16", scale=1.0, address = [5]
                }
            }
        }
}
```

Строка [[inputs.modbus]] указывает, что далее идет шаблон входного плагина для протокола Modbus.

Параметр **name** задает имя датчика как DL-303-rtu.

Параметр **slave_id**, равный единице, задает идентификатор устройства на шине Modbus. Каждое устройство на шине Modbus должно иметь свой, уникальный идентификатор, или, как его еще называют, адрес Modbus.

При помощи параметра **timeout** задается время ожидания ответа от устройства. В зависимости от условий эксплуатации вам, возможно, придется изменять время, указанное в шаблоне по умолчанию.

Параметр **controller** задает порт, через который выполняется обмен данными с устройством. Здесь вам нужно оставить его значение как file:///dev/ttyS3, так как именно этот порт используется в модуле FrontControl Compact.

Параметры **baud_rate**, **data_bits**, **parity** и **stop_bits** также не требуют редактирования. Они задают параметры работы последовательного порта, такие как скорость, количество битов данных, наличие битов четности и стоп-битов.

Параметр **transmission_mode**, заданный как RTU, указывает, что для связи с устройством используется режим Modbus RTU. При этом используется бинарный режим передачи данных в протоколе Modbus с проверкой контрольной суммы.

Также вам может встретиться текстовый режим, заданный как ASCII.

Массив параметров **input_registers** определяет, какие регистры датчика будут считываться, как полученные данные будут интерпретироваться и как обрабатываться. В этом массиве каждый элемент соответствует одному регистру.

Параметры **address**, заданные здесь как [0], [1], [2], [3] и [5], задают адреса регистров ввода.

Поля **measurement** и **name** задают имена измерений и параметров, соответственно.

При помощи параметров **data_type** и **byte_order** указывается тип данных, а также порядок следования байт в слове данных. Значение "AB" означает, что первый байт (A) идет перед вторым байтом (B).

И, наконец, при помощи параметра **scale** можно задать масштабирование полученных данных. Если этот параметр указан как 1.0, данные передаются без масштабирования.

Если вы будете добавлять шаблон своего датчика или создавать собственный шаблон датчика, подключенного через Modbus RTU, воспользуйтесь заготовкой шаблона anysensor-modbus-ttyS3 (рис. 4.8).

8. anysensor-modbus-ttyS3 : Any sensor with address=1. hold-register=0 (first) on /dev/ttyS3 (FCC device) with 9600\8n1 (sensor parameter)

Активировать

⋮

Рис. 4.8. Заготовка шаблона anysensor-modbus-ttyS3

Текст этого шаблона приведен ниже:

```
## Any sensor with address=1. hold-register=0 (first) on /dev/ttyS3
(FCC device) with 9600\8n1 (sensor parameter)
# All Telegraf configs and Grafana dashboards:
https://github.com/lab240/telegraf-grafana-configs

[[inputs.modbus]]
#change name="YOUR_SENSOR" and measurement="YOUR_SENSOR"
name = "anysensor-modbus-rtu"
```

```
#change to your modbus address if your address is not equal 1
slave_id = 1

#change to your port settings, if you need
baud_rate = 9600
data_bits = 8
parity = "N"
stop_bits = 1

#change to your port (ttyS1 \ ttyS3 \ etc)
controller = "file:///dev/ttyS3"
transmission_mode = "RTU"

input_registers = [
    { measurement="anysensor-modbus-rtu", name = "First_Hold_Register",
byte_order = "AB", data_type = "UINT16", scale=1.0, address =
[0] },
]
```

Отредактируйте параметры этого шаблона в соответствии с описанием вашего датчика.

В разделе **8. Подключение и мониторинг датчиков Modbus RTU** вы найдете детальное описание подключения датчиков, таких как PZEM-016, XY-MD02 и ICPDAS DL-303 через RS485.

Редактирование и тестирование шаблона

Чтобы отредактировать шаблон, откройте web-приложение NapiConfig, и выберите в меню **Датчики** строку **Управление**. Найдите нужный вам плагин, щелкните кнопку с тремя вертикальными точками и выберите строку **Ред. шаблон**.

Откроется редактор шаблона (рис. 4.9).

Применить Назад

Тест конфигурации

```
1 ## Any sensor with address=1, hold-register=0 (first) on /dev/ttyS3 (FCC device) with 9600\8n1 (sensor parameter)
2
3 # All Telegraf configs and Grafana dashboards: https://github.com/lab240/telegraf-grafana-configs
4
5 [[inputs.modbus]]
6
7 #change name="YOUR_SENSOR" and measurement="YOUR_SENSOR"
8
9 name = "anysensor-modbus-rtu"
10
11 #change to your modbus address if your address is not equal 1
12 slave_id = 1
13
14 #change to your port settings, if you need
15 baud_rate = 9600
16 data_bits = 8
17 parity = "N"
18 stop_bits = 1
19
20 #change to your port (ttyS1 \ ttyUSB0 \ etc)
21 controller = "file:///dev/ttyS3"
22 transmission_mode = "RTU"
23
24 input_registers = [
25   { measurement="anysensor-modbus-rtu", name = "First_Hold_Register", byte_order = "AB", data_type = "UINT16", scale=1.0, address
26   = [0]},]
27
28
```

Применить Назад

Рис. 4.9. Редактор шаблона

После редактирования проверьте его формат при помощи кнопки **Тест конфигурации**. На рис. 4.10 показан результат успешного тестирования, когда в шаблоне нет ошибок.

Результат теста конфигурации ×

Data:

```
> XY_MD02-rtu,host=napi-rk3308b-s,name=XY_MD02,slave_id=1,type=input_register Humidity=26i,Temperature=22i
17351067960000000000
```

Info:

```
2024-12-25T06:06:36Z I! Loading config: /etc/telegraf/telegraf.conf
2024-12-25T06:06:36Z I! Loading config: /data/active/XY_MD02.conf
2024-12-25T06:06:36Z I! Starting Telegraf 1.28.5-2c3cd1c4 brought to you by InfluxData the makers of InfluxDB
2024-12-25T06:06:36Z I! Available plugins: 13 inputs, 9 aggregators, 29 processors, 24 parsers, 3 outputs, 5 secret-stores
2024-12-25T06:06:36Z I! Loaded inputs: modbus
2024-12-25T06:06:36Z I! Loaded aggregators:
2024-12-25T06:06:36Z I! Loaded processors:
2024-12-25T06:06:36Z I! Loaded secretstores:
2024-12-25T06:06:36Z W! Outputs are not used in testing mode!
2024-12-25T06:06:36Z I! Tags enabled: host=napi-rk3308b-s
2024-12-25T06:06:36Z I! [inputs.modbus] Got 0 request(s) touching 0 holding registers for 0 fields (slave 1)
2024-12-25T06:06:36Z I! [inputs.modbus] Got 1 request(s) touching 2 inputs registers for 2 fields (slave 1)
2024-12-25T06:06:36Z I! [inputs.modbus] Got 0 request(s) touching 0 discrete registers for 0 fields (slave 1)
2024-12-25T06:06:36Z I! [inputs.modbus] Got 0 request(s) touching 0 coil registers for 0 fields (slave 1)
```

Закрыть

Рис. 4.10. Результат успешного тестирования шаблона

Если в шаблоне имеются ошибки, в результатах тестирования появится сообщение об этом с указанием ошибки (рис. 4.11).

Результат теста конфигурации X

Data:

Info:

```
2024-12-25T06:09:17Z I! Loading config: /etc/telegraf/telegraf.conf
2024-12-25T06:09:17Z I! Loading config: /data/active/XY_MD02.conf
2024-12-25T06:09:17Z I! Starting Telegraf 1.28.5-2c3cd1c4 brought to you by InfluxData the makers of InfluxDB
2024-12-25T06:09:17Z I! Available plugins: 13 inputs, 9 aggregators, 29 processors, 24 parsers, 3 outputs, 5 secret-stores
2024-12-25T06:09:17Z I! Loaded inputs: modbus
2024-12-25T06:09:17Z I! Loaded aggregators:
2024-12-25T06:09:17Z I! Loaded processors:
2024-12-25T06:09:17Z I! Loaded secretstores:
2024-12-25T06:09:17Z W! Outputs are not used in testing mode!
2024-12-25T06:09:17Z I! Tags enabled: host=napi-rk3308b-s
2024-12-25T06:09:17Z I! [inputs.modbus] Got 0 request(s) touching 0 holding registers for 0 fields (slave 2)
2024-12-25T06:09:17Z I! [inputs.modbus] Got 1 request(s) touching 2 inputs registers for 2 fields (slave 2)
2024-12-25T06:09:17Z I! [inputs.modbus] Got 0 request(s) touching 0 discrete registers for 0 fields (slave 2)
2024-12-25T06:09:17Z I! [inputs.modbus] Got 0 request(s) touching 0 coil registers for 0 fields (slave 2)
2024-12-25T06:09:27Z E! [inputs.modbus] Error in plugin: slave 2: serial: timeout
2024-12-25T06:09:27Z E! [telegraf] Error running agent: input plugins recorded 1 errors
```

Закрыть

Рис. 4.11. В результате тестирования была обнаружена ошибка

В данном случае был задан неправильный адрес датчика на шине Modbus.

Если, например, задать неправильный адрес регистра, это будет отмечено в сообщении об ошибке:

```
2024-12-25T06:12:59Z E! [inputs.modbus] Error in plugin: slave 1:
modbus: exception '2' (illegal data address), function '132'
2024-12-25T06:12:59Z E! [telegraf] Error running agent: input plugins
recorded 1 errors
```

Если ошибок нет, сохраните шаблон кнопкой **Применить**. Далее добавьте шаблон в основную конфигурацию кнопкой **Объединить с оригиналом**.

Шаблоны датчиков ModbusTCP

Если датчик подключен к модулю сборщика FCC-0801 через Ethernet, для него нужен соответствующий шаблон.

Ниже приведен текст шаблона рассмотренного выше датчика DL-303, но подключенного с использованием протокола ModbusTCP:

```

## DL-303 (CO,CO2,temp,humidity) ModbusTCP
[[inputs.modbus]]

name = "DL-303"

## Slave ID - addresses a MODBUS device on the bus
## Range: 0 - 255 [0 = broadcast; 248 - 255 = reserved]
slave_id = 1

## Timeout for each request
timeout = "1s"

# TCP - connect via Modbus/TCP
controller = "tcp://10.10.10.120:502"

input_registers = [
    { measurement="DL-303", name = "CO", byte_order = "AB",
data_type = "UINT16", scale=1.0, address = [0]},
    { measurement="DL-303", name = "CO2", byte_order = "AB",
data_type = "UINT16", scale=1.0, address = [1]},
    { measurement="DL-303", name = "Humidity", byte_order = "AB",
data_type = "UINT16", scale=1.0, address = [2]},
    { measurement="DL-303", name = "Temp", byte_order = "AB",
data_type = "UINT16", scale=1.0, address = [3]},
    { measurement="DL-303", name = "DewPoint", byte_order = "AB",
data_type = "UINT16", scale=1.0, address = [5]},
]

```

Обратите внимание, что этот шаблон во многом аналогичен приведенному выше, за исключением параметра **controller**. Теперь этот параметр указывает на использование протокола TCP для подключения к устройству Modbus, расположенному по адресу IP 10.10.10.120 на порту 502.

Правильный адрес IP устройства и порт нужно взять из документации на ваш датчик. Также возможно вам нужно будет получить консультацию у системного или сетевого администратора, отвечающего за настройку сети.

Также обратите внимание на заготовку шаблона anysensor-modbus-tcp-localhost:

```

[[inputs.modbus]]
#change name="YOUR_SENSOR" and measurement="YOUR_SENSOR"
name = "anysensor-modbus-localhost"

#change to your modbus address if your address is not equal 1
slave_id = 1

## Timeout for each request
timeout = "1s"

# TCP - connect via Modbus/TCP
controller = "tcp://127.0.0.1:502"

```

```

input_registers = [
    { measurement="anysensor-modbus-localhost", name =
"First_Hold_Register", byte_order = "AB", data_type = "UINT16",
scale=1.0, address = [0] },
]

```

Вы должны здесь задать:

- имя вашего датчика name;
- уникальный идентификатор slave_id на шине Modbus;
- параметр timeout, соответствующий вашему датчику;
- адрес и порт устройства в параметре controller;
- массив input_registers с описаниями регистров вашего датчика

Подготовив шаблон, проверьте его, сохраните и активируйте.

Настройка шлюза Modbus RTU - Modbus TCP

Общие сведения

Как работает шлюз?

- Опрос от пользователей идет на **FCC-0801** по его адресу IP с помощью протокола Modbus TCP.
- FCC-0801** преобразует запрос в Modbus RTU, опрашивает датчики по RS-485 и возвращает ответы по Modbus TCP пользователю.

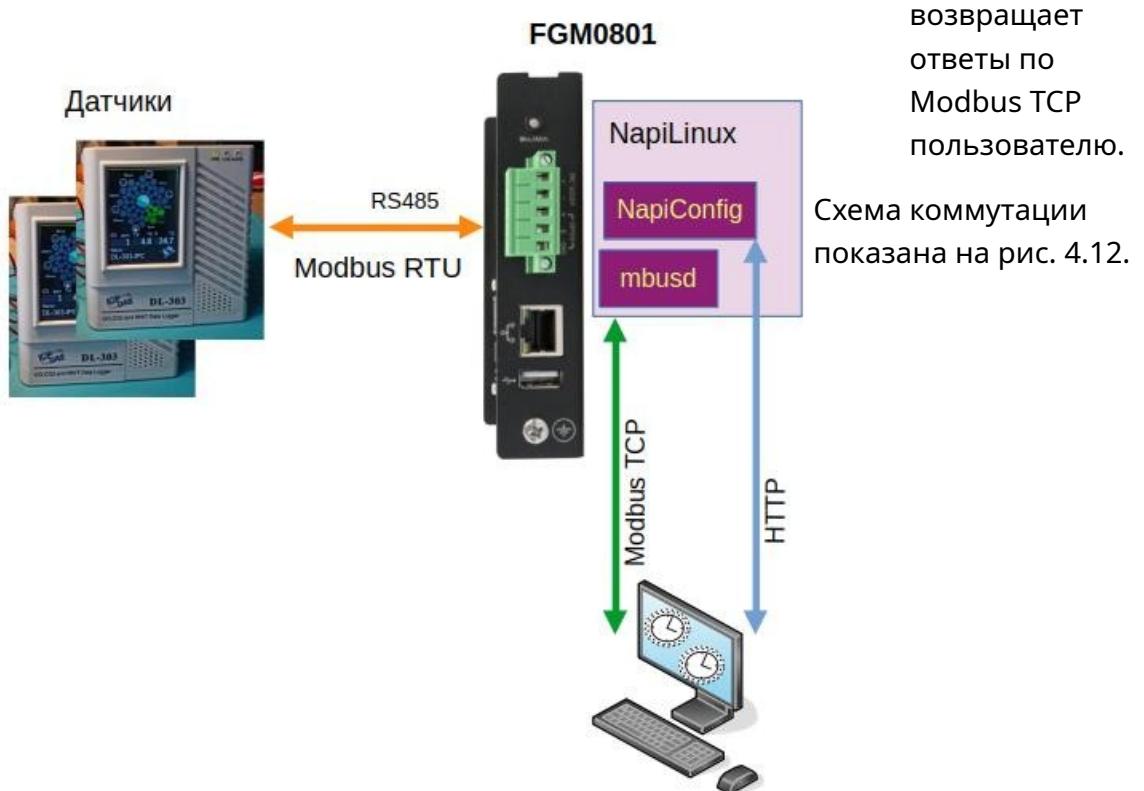


Рис. 4.12. Схема коммутации для шлюза между Modbus RTU и Modbus TCP

Согласно этой схеме:

- Датчики соединяются со **FCC-0801** через последовательный порт RS485.
- **FCC-0801** соединен с сетью, у него свой адрес IP

Активируйте службу mbusd

Для активации шлюза необходимо включить службу mbusd. Для этого откройте Веб-приложение NapiConfig, и выберите в меню **Настройки** строку **Общие**. Далее найдите в списке **Системные службы** строку **mbusd** и, если она не активна, щелкните кнопку **Inactive** красного цвета. Далее подтвердите включение службы, щелкнув во всплывающем окне кнопку **Подтвердить**.

Чтобы служба mbusd запускалась автоматически при загрузке **FCC-0801**, щелкните также кнопку **Disabled**, подтвердив автоматический запуск.

Если все сделано правильно, кнопки, расположенные справа от строки **mbusd**, примут вид, показанный на рис. 4.13.

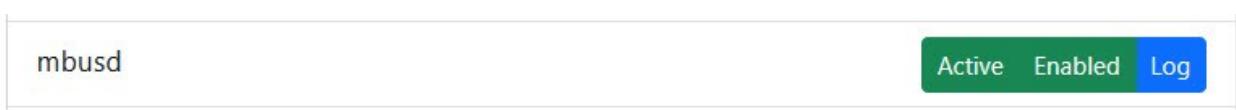


Рис. 4.13. Запущена служба mbusd

В списке **Системные Службы** сервис mbusd должен быть в статусе **Enabled, Active**. Если сервис не активирован, щелкните сначала кнопку **Disabled**, а затем **Inactive**. Статус сервиса должен измениться.

С помощью кнопки Log можно при необходимости просмотреть журнал службы mbusd. Если что-то пошло не так, в этом журнале могут быть ошибки, которые следует проанализировать.

Вы можете управлять сервисом mbusd через командную строку.

- Включить сервис

```
# systemctl enable mbusd
```

- Запустить сервис (при старте будет запускаться вручную)

```
# systemctl start mbusd
```

- После изменения конфигурации перезапустить сервис

```
# systemctl restart mbusd
```

- Остановить сервис

```
# systemctl stop mbusd
```

- Выключить сервис

```
# systemctl disable mbusd
```

⚠ Рекомендуем пользоваться Веб-интерфейсом, в нем нельзя "испортить" конфигурационный файл.

Настройка параметров шлюза mbusd

В **FCC-0801** использовано свободное программное обеспечение mbusd <https://github.com/3cky/mbusd>. В сборке NapiLinux этот пакет уже установлен.

Настройка mbusd производится через меню NapiConfig: **Общие — ModbusGW**

Здесь можно редактировать все параметры этого конфигурационного файла через Веб-интерфейс (рис. 4.14)

Уровень подробности логирования	<input type="text" value="2"/> ✓	0-9, default 2, 0 - errors only [/var/log/mbus.log]
Имя устройства последовательного порта	<input type="text" value="/dev/ttyS3"/> ✓	Имя устройства последовательного порта (например, /dev/ttyS3)
Скорость последовательного порта	<input type="text" value="115200"/> 1 ✓	Скорость последовательного порта (например, 115200)
Режим последовательного порта	<input type="text" value="8n1"/> 2 ✓	Режим последовательного порта (например, 8n1)
Тип управления направлением данных RS-485	<input type="text" value="addc"/> ✓	Тип управления направлением данных RS-485 (addc, rts_0, rts/rts_1, sysfs_0, sysfs_1)
Адрес TCP сервера	<input type="text" value="0.0.0.0"/> 3 ✓	для обращения (0.0.0.0 для всех интерфейсов или конкретный IP адрес сетевой карты)
Порт TCP сервера	<input type="text" value="502"/> 4 ✓	Порт TCP сервера (например, 502)
Макс. количество TCP соединений	<input type="text" value="32"/> 5 ✓	Максимальное количество одновременных TCP соединений (например, 32)
Время ожидания соединения в секундах	<input type="text" value="60"/> ✓	Время ожидания соединения в секундах (например, 60)
Максимальное количество попыток запроса	<input type="text" value="3"/> ✓	Максимальное количество попыток запроса (например, 3)
Пауза между запросами в миллисекундах	<input type="text" value="100"/> 6 ✓	Пауза между запросами в миллисекундах (например, 100)
Время ожидания ответа в миллисекундах	<input type="text" value="500"/> ✓	Время ожидания ответа в миллисекундах (например, 500)
Ответить на широковещательный запрос	<input type="checkbox"/>	Ответить на широковещательный запрос

Сохранить

Рис. 4.14. Настройка параметров шлюза между Modbus RTU и Modbus TCP

Также Вы можете править файл через файл /etc/mbusd.conf напрямую, но мы не рекомендуем это делать в стандартных конфигурациях.

Перечислим настройки.

Настройки последовательного порта

Основные настройки, которые необходимо сделать — это настроить параметры порта (на рис. 4.14 эти настройки обозначены цифрами 1 и 2). Скорость, четность, стоп-бит необходимо установить такими же, как в датчиках.

⚠ В 90% случаев настройки порта датчика — 9600\8n1 или 115200\8n1

Если датчиков несколько, то у них должны быть одинаковые параметры порта и разные modbus адреса! В случае одного датчика, modbus адрес знать необязательно.

По большому счету, остальные параметры можно оставить по умолчанию, если нет особых требований.

Сетевые настройки

⚠ В стандартных случаях менять сетевые настройки нет необходимости.

Настройка **Адрес TCP-сервера** (обозначена на рис. 4.14 цифрой 3) — это адрес IP, который "слушает" сервис. По умолчанию (0.0.0.0), сервис слушает все интерфейсы.

Порт TCP сервера (4) также можно сменить на нестандартный. Это может понадобиться, например, при запуске нескольких подобных сервисов.

Тюнинг

⚠ В стандартных случаях нет необходимости менять сетевые настройки.

Количество TCP-соединений (5) можно увеличить, если датчиков очень много и данные "застревают".

Можно также уменьшить паузу между запросами (6), если этого требуется.

Перезапуск службы после изменения параметров

Перезапуск службы после изменения параметров происходит автоматически.

Для ручного перезапуска службы (не рекомендуется) используйте команды

```
# systemctl restart mbusd
```

Затем проверьте статус сервиса следующей командой:

```
# systemctl status mbusd
```

Проверка работы шлюза: читаем датчик

Итак, датчик подсоединен к **FCC-0801** с верными параметрами по RS485. Попробуем прочитать его через шлюз. Для этого необходим компьютер с установленной утилитой modpoll (или mbpoll).

Допустим, IP адрес **FCC-0801**: 10.10.10.114.

Проверить доступен ли **FCC-0801** по сети можно командой ping:

```
$ ping 10.10.10.114
PING 10.10.10.114 (10.10.10.114) 56(84) bytes of data.
64 bytes from 10.10.10.114: icmp_seq=1 ttl=63 time=3.56 ms
64 bytes from 10.10.10.114: icmp_seq=2 ttl=63 time=3.60 ms
^C
--- 10.10.10.114 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 3.560/3.581/3.603/0.021 ms
```

Если ping закрыт файрволом, проверьте командой telnet порт, по которому отвечает mbusd:

```
$ telnet 10.10.10.114 502
Trying 10.10.10.114...
Connected to 10.10.10.114.
Escape character is '^]'.
```

Выполняем опрос датчика

На вашем компьютере (⚠ не на FCC-0801) выполняем команду опроса **FCC-0801** по Modbus TCP:

```
$ modpoll -m tcp -a 1 -r 1 -c 5 -t 3 10.10.10.114
modpoll 3.10 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright (c) 2002-2021 proconX Pty Ltd
Visit https://www.modbusdriver.com for Modbus libraries and tools.
```

```
Protocol configuration: MODBUS/TCP, FC4
Slave configuration....: address = 1, start reference = 1, count = 5
Communication.....: 10.10.10.114, port 502, t/o 1.00 s, poll rate
1000 ms
Data type.....: 16-bit register, input register table

-- Polling slave... (Ctrl-C to stop)
[1]: 0
[2]: 1102
[3]: 2797
[4]: 2717
[5]: 8090
-- Polling slave... (Ctrl-C to stop)
[1]: 0
[2]: 1102
[3]: 2798
[4]: 2716
[5]: 8088
-- Polling slave... (Ctrl-C to stop)
```

Если данные получены, поздравляем, шлюз работает !

Решение проблем

Если данные не приходят, необходимо проверить, а доступен ли датчик со **FCC-0801** по Modbus RTU. Для этого:

- Зайдите по ssh на **FCC-0801**
- Остановите сервис mbusd командой:

```
# systemctl stop mbusd
```

- Запустите команду опроса по modbus rtu с правильными параметрами

```
# modpoll -m rtu -b 115200 -p none -d 8 -a 1 -r 1 -c 5 -t 3
/dev/ttyS3
modpoll 3.10 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright (c) 2002-2021 proconX Pty Ltd
Visit https://www.modbusdriver.com for Modbus libraries and tools.

Protocol configuration: Modbus RTU, FC4
Slave configuration....: address = 1, start reference = 1, count = 5
Communication.....: /dev/ttyS3, 115200, 8, 1, none, t/o 1.00 s,
poll rate 1000 ms
Data type.....: 16-bit register, input register table

-- Polling slave... (Ctrl-C to stop)
[1]: 0
[2]: 1100
[3]: 2801
[4]: 2729
[5]: 8112
-- Polling slave... (Ctrl-C to stop)
[1]: 0
[2]: 1100
[3]: 2802
[4]: 2726
[5]: 8106
-- Polling slave... (Ctrl-C to stop)
```

Необходимо добиться, чтобы данные поступали !

Для этого проверяйте физическое соединение и совпадение параметров порта RS485 с датчиком.

- Когда регистры начнут читаться, снова запустите mbusd:

```
# systemctl start mbusd
```

- Убедитесь, что в конфигурации mbusd заданы верные параметры порта !

Установка и запуск Grafana

Если вы контролируете небольшое количество датчиков, можно использовать Grafana, установленный в модуле сборщика FCC-0801.

Также имеется возможность установить Grafana на отдельном компьютере, на сервере или на виртуальной машине.

Включение Grafana в сборщике FCC-0801

Чтобы активировать Grafana, установленный в модуле сборщика FCC-0801, выберите из меню **Настройки** web-интерфейса NapiConfig строку **Общие**, а затем перейдите к блоку **Системные службы**.

Щелкните там кнопки **Inactive** и модуле сборщика FCC-0801 в строке **grafana-server**, чтобы включить службу.

На рис. 5.1 показано, что служба уже активна.

Системные службы	Active	Enabled	Log
systemd-networkd	Active	Enabled	Log
systemd-resolved	Active	Enabled	Log
telegraf	Active	Enabled	Log
mosquitto	Active	Enabled	Log
grafana-server	Active	Enabled	Log
influxdb	Active	Enabled	Log
swupdate	Active	Enabled	Log
mbusd	Inactive	Disabled	Log
snmpd	Inactive	Disabled	Log

Рис. 5.1. Активация службы grafana-server

После активации откройте в браузере адрес вида <http://192.168.0.18:3000/>, где вместо 192.168.0.18 укажите адрес IP вашего модуля FCC-0801.

Обратите внимание, что для доступа по этому адресу нужно использовать протокол HTTP, а не HTTPS.

Если служба запустилась без ошибок, вы увидите в браузере главную страницу Grafana (рис. 5.2).

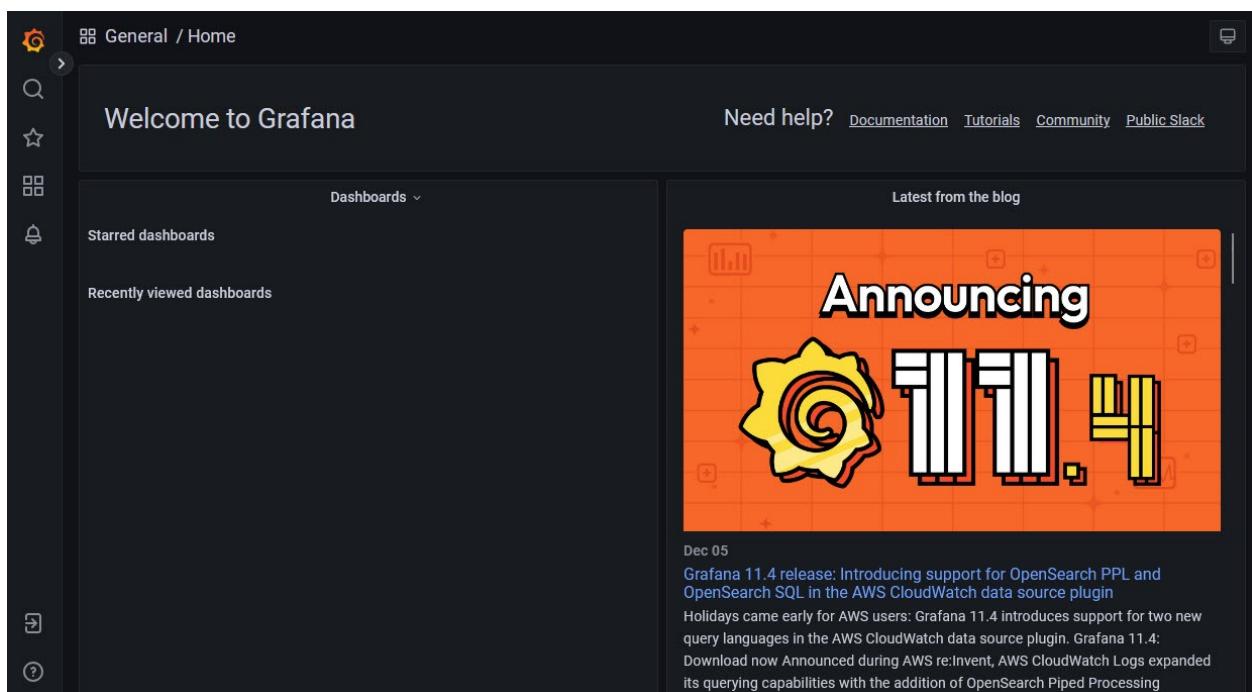


Рис. 5.2. Главная страница Grafana в сборщике FCC-0801

Для входа введите логин admin и пароль nnzadmin. Впоследствии пароль нужно будет изменить.

Установка Grafana из пакета в Ubuntu и Debian

Перед установкой Grafana на сервер (физический или виртуальный) Ubuntu обновите список пакетов и установите необходимые зависимости:

```
sudo apt-get update
```

Далее добавьте пользователя, скачайте и установите нужный пакет:

```
sudo apt-get install -y adduser libfontconfig1 musl
wget https://dl.grafana.com/oss/release/grafana\_11.4.0\_amd64.deb
sudo dpkg -i grafana_11.4.0_amd64.deb
```

Теперь вы можете запустить сервис Grafana и проверить его состояние:

```
sudo /bin/systemctl start grafana-server
sudo /bin/systemctl status grafana-server
```

Чтобы сервис Grafana запускался автоматически при перезагрузке ОС, введите такую команду:

```
sudo /bin/systemctl enable grafana-server
```

После запуска сервиса откройте в браузере адрес вида <http://192.168.0.20:3000/>, где вместо 192.168.0.20 укажите адрес IP узла, на который вы только что установили Grafana.

Для доступа по этому адресу нужно использовать протокол HTTP, а не HTTPS. Логин для входа admin, пароль — также admin. При первом входе вам будет предложено его заменить.

После входа вы увидите главное окно Grafana (рис. 5.3).

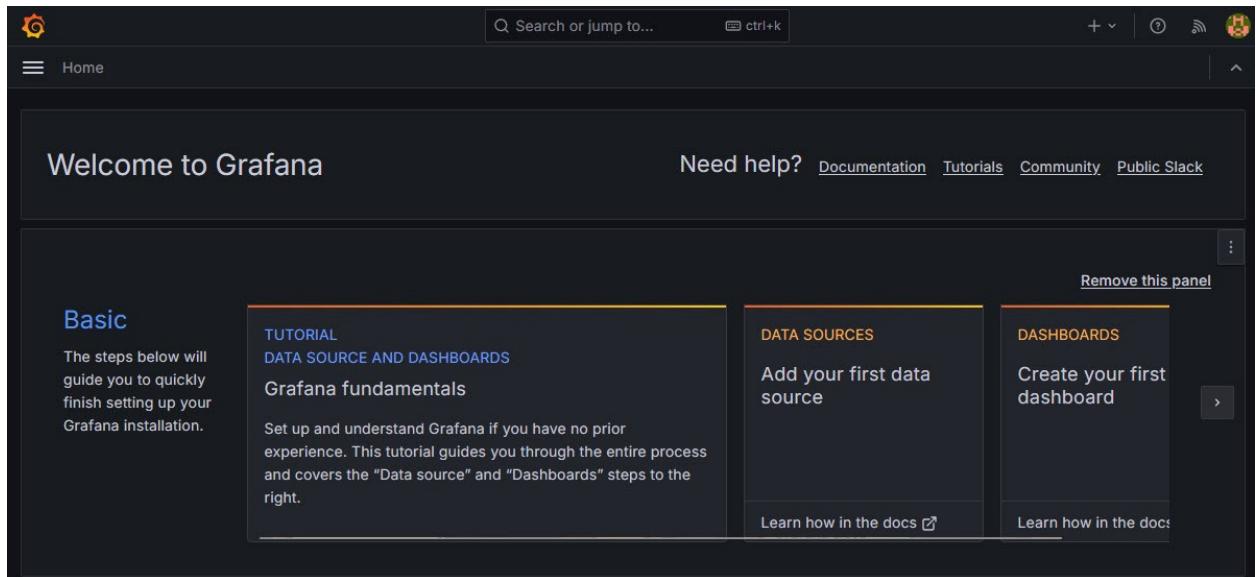


Рис. 5.3. Главное окно Grafana сразу после установки

Здесь нужно ввести логин admin и пароль admin.

Установка для процессоров ARM

Систему Grafana можно установить на сервер или микрокомпьютер с процессором ARM. Ниже представлена соответствующие команды установки:

```
sudo apt-get install -y adduser libfontconfig1 musl
wget https://dl.grafana.com/oss/release/grafana\_11.4.0\_arm64.deb
sudo dpkg -i grafana_11.4.0_arm64.deb
```

После установки запустите сервис и проверьте его состояние:

```
sudo /bin/systemctl start grafana-server
sudo /bin/systemctl status grafana-server
```

Если установка прошла без ошибок, то введите в браузере адрес вида <http://192.168.0.20:3000/>. Появится главное окно Grafana (рис. 5.3).

Запуск Grafana в Docker

Прежде всего, установите Docker, если он еще не установлен. Для Ubuntu выполните следующие подготовительные команды:

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \
"deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

Затем запустите саму установку:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-
buildx-plugin docker-compose-plugin
```

Проверьте, что все работает:

```
sudo docker run hello-world
```

После запуска этой команды на консоли должно появиться сообщение, содержащее строку «Hello from Docker!».

Для запуска Grafana в Docker проще всего воспользоваться файлом docker-compose.yml из репозитория <https://github.com/lab240/grafana-docker>:

```
services:

grafana:
  image: grafana/grafana
  container_name: lab_grafana
  restart: always
  volumes:
    - grafana-data:/var/lib/grafana
  ports:
    - "3000:3000"
  networks:
    - lab240-monitoring
  environment:
    # - GF_LOG_LEVEL=debug
    # - GF_LOG.CONSOLE_LEVEL=debug
    - GF_SECURITY_ADMIN_USER=${ADMIN_USER:-admin}
```

```
- GF_SECURITY_ADMIN_PASSWORD=${ADMIN_PASSWORD:-admin}
- GF_USERS_ALLOW_SIGN_UP=${ALLOW_SIGN_UP:-false}
- GF_AUTH_ANONYMOUS_ENABLED=${ANONYMOUS_ENABLED:-true}
networks:
  lab240-monitoring:
volumes:
  grafana-data:
```

Файл docker-compose.yml можно скачать по ссылке

<https://raw.githubusercontent.com/lab240/grafana-docker/refs/heads/main/docker-compose.yml>.

Чтобы запустить Grafana, выполните следующую команду из каталога, в котором находится файл docker-compose.yml:

```
sudo docker compose up -d
```

Через некоторое время Grafana запустится и вы сможете открыть главную страницу его web-интерфейса, как это было описано ранее.

Для остановки контейнера выдайте такую команду:

```
sudo docker compose down
```

Команда docker ps покажет список всех запущенных контейнеров вместе с их именами и идентификаторами:

```
sudo docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS
PORTS          NAMES
3b65c707014a   grafana/grafana "/run.sh"   18 seconds ago   Up 17
seconds        0.0.0.0:3000->3000/tcp, :::3000->3000/tcp   lab_grafana
```

Установка Grafana в Microsoft Windows

Для установки Grafana в Microsoft Windows нужно скачать соответствующую версию программы установки с [сайта Grafana](#) (рис. 5.4) и запустить ее.

The screenshot shows the Grafana Labs website with a black header bar containing links for Products, Open Source, Solutions, Learn, Docs, Company, Downloads, Contact us, and Sign in. Below the header is an orange navigation bar with the Grafana logo and links for Overview, Grafana project, Grafana Alerting, and Download (which is underlined). The main content area has a white background with a title 'Download Grafana' and a 'Nightly Builds' button. A callout box titled 'Grafana Cloud' encourages users to use Grafana Cloud for metrics, logs, and traces. Below the callout are dropdown menus for Version (set to 11.4.0) and Edition (set to OSS), followed by detailed information about the OSS edition. There are also links for Release Info and a 'What's New In Grafana 11.4.0'. At the bottom are download links for various platforms: Linux, Windows (highlighted in blue), Mac, Docker, and Linux on ARM64.

Download Grafana

Nightly Builds

Grafana Cloud

You can use Grafana Cloud to avoid installing, maintaining, and scaling your own instance of Grafana. [Create a free account to get started](#), which includes free forever access to 10k metrics, 50GB logs, 50GB traces, & more.

Version:

11.4.0

Edition:

OSS

The Enterprise Edition is the default and recommended edition. It includes all the features of the OSS Edition, can be used for free and can be upgraded to the full Enterprise feature set, including support for [Enterprise plugins](#).

License:

AGPLv3

Release Date:

December 5, 2024

Release Info:

[What's New In Grafana 11.4.0](#)



Windows Installer (64 Bit) SHA256: 5188632379e1930807fc25053b0e92f2cf1e758d8c307a0a91ac6ab4f43fac4

[Download the installer](#) (grafana-11.4.0.windows-amd64.msi) and run it.

Standalone Windows Binaries (64 Bit) SHA256: 4b9e7454dc2b65ca0c5cf938a7b0df1d50606e840a205c0d82fa37e09b0eb86e

[Download the zip file](#) (grafana-11.4.0.windows-amd64.zip) and follow the instructions in the installation guide below.

Read the [Windows installation guide](#).

Рис. 5.4. Выбор программы установки для Microsoft Windows

Для установки версии Open Source выберите в списке **Edition** строку **OSS**, а затем щелкните ссылку **Download the installer**.

После запуска установки вы, возможно, получите предупреждение от Microsoft Defender о предотвращении запуска неопознанного приложения (рис. 5.5).

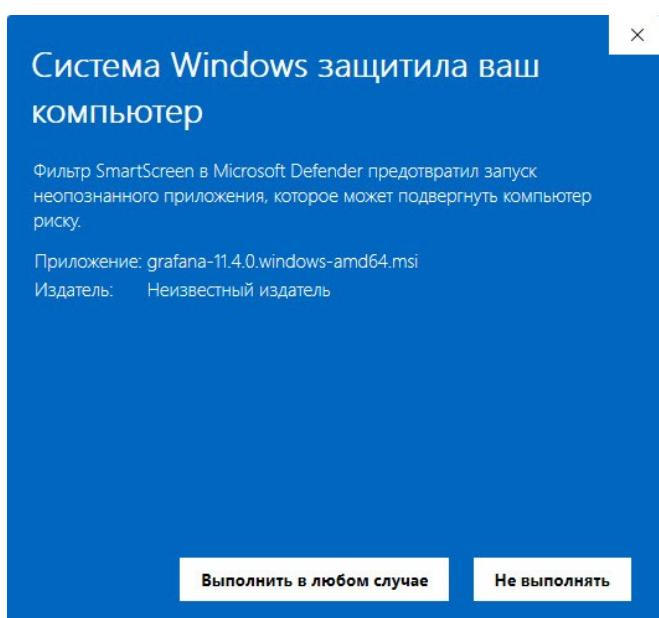


Рис. 5.5. Предупреждение о запуске неопознанного приложения

Для продолжения установки щелкните кнопку **Выполнить в любом случае**. После этого на экране появится первая панель программы установки (рис. 5.6).

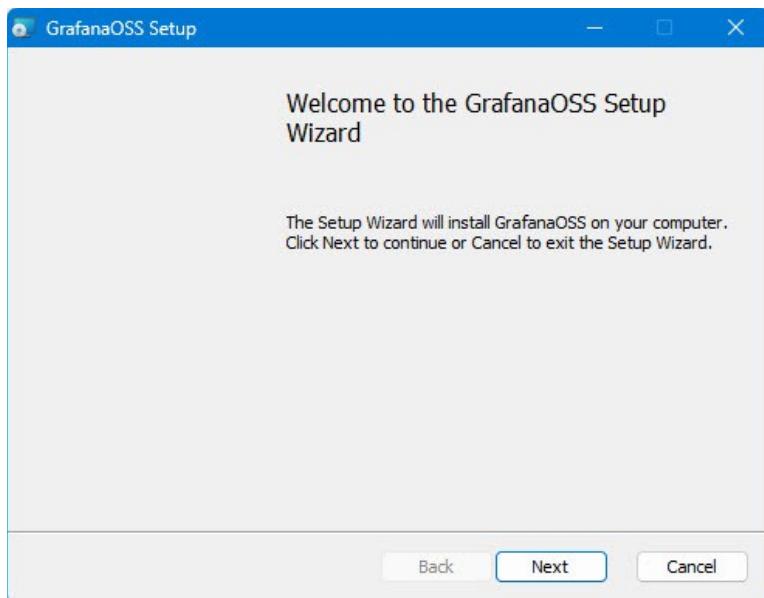


Рис. 5.6.Первая панель программы установки Grafana

Здесь нужно щелкнуть кнопку **Next** для перехода к просмотру лицензионного соглашения (рис. 5.7).

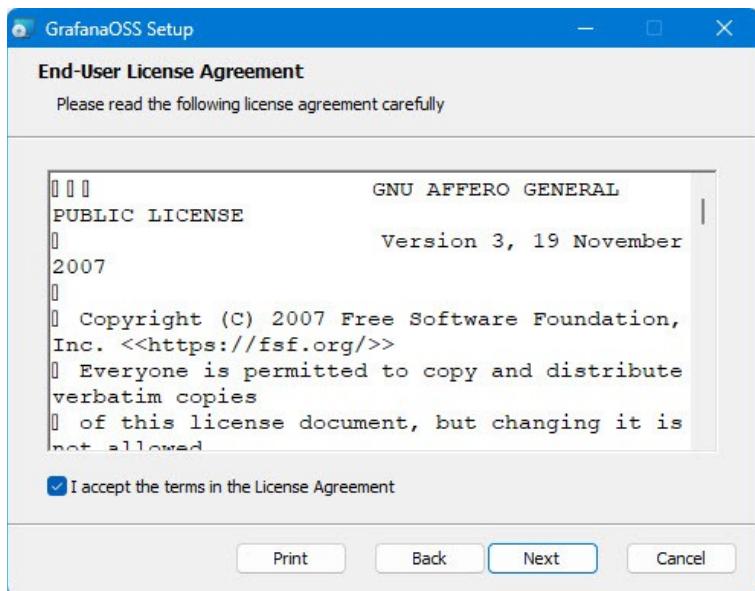


Рис. 5.7. Просмотр лицензионного соглашения

Установите флажок **I accept the terms in the License Agreement** и щелкните кнопку **Next**.

Следующий шаг — выбор параметров установки (рис. 5.8).

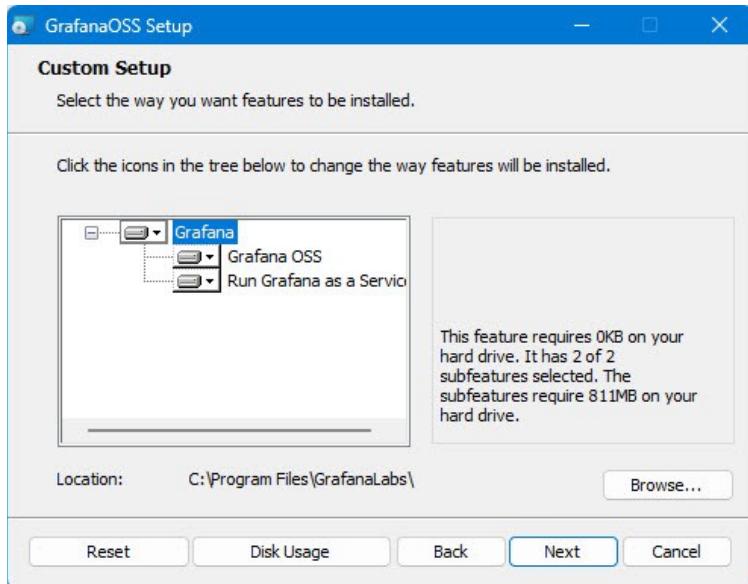


Рис. 5.8. Выбор параметров установки

Здесь все можно оставить по умолчанию, щелкнув снова кнопку **Next**. Вы увидите панель с кнопкой **Install**, которая запустит процесс установки (рис. 5.9).

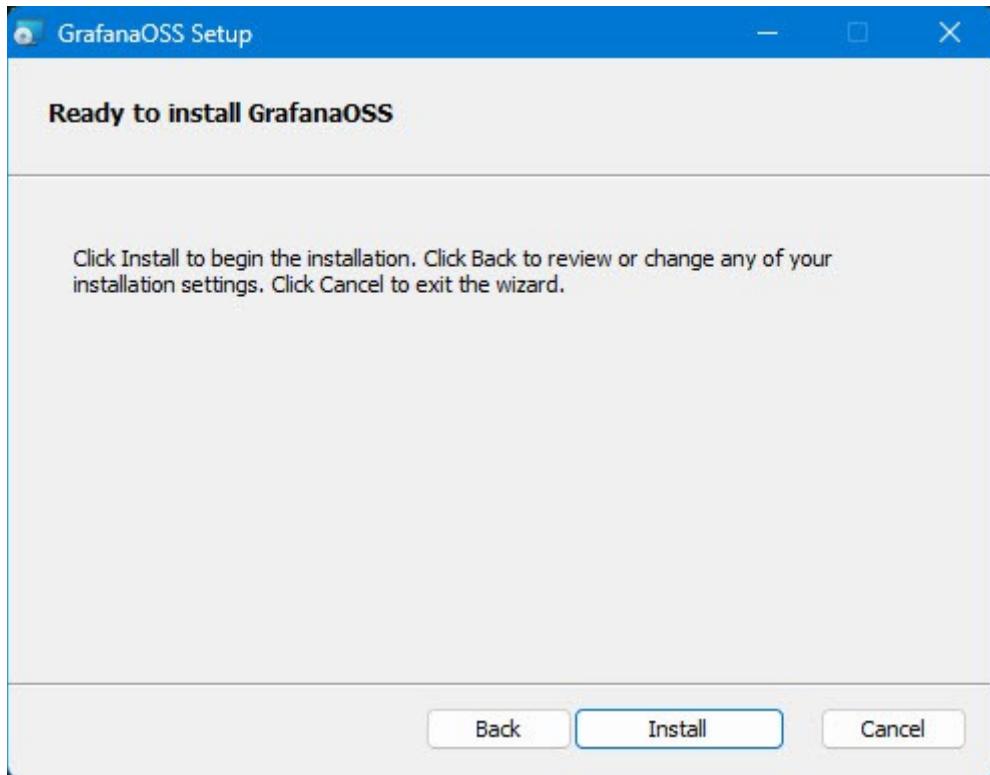


Рис. 5.9. Панель для запуска процесса установки

Здесь опять может появиться запрос подтверждения на установку, с которым вам нужно будет согласиться. Как только вы это сделаете, начнется процесс установки (рис. 5.10).

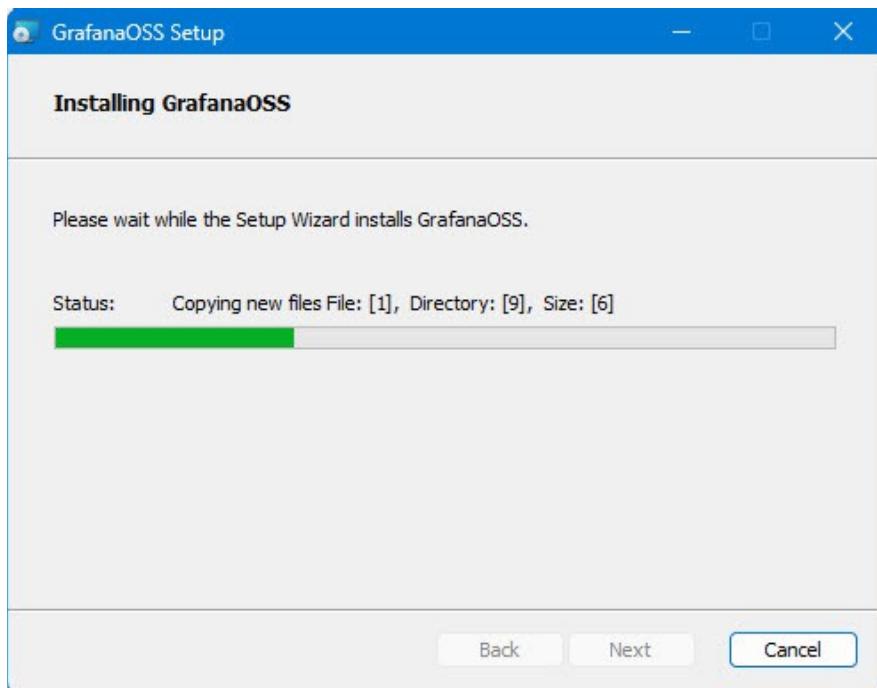


Рис. 5.10. Ход процесса установки

После завершения установки на экране появится последняя панель установщика, где нужно щелкнуть кнопку **Finish** (рис. 5.11).

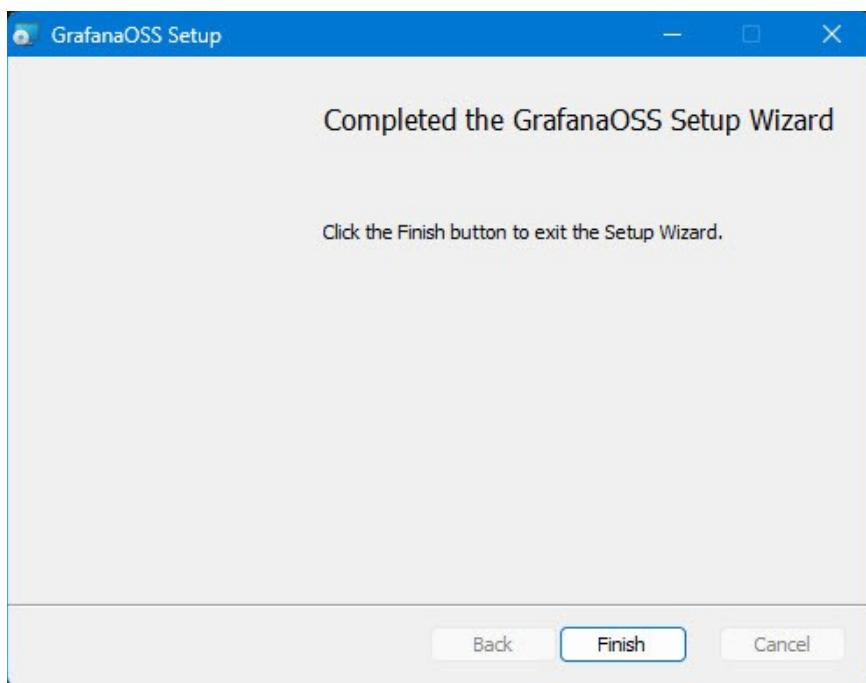


Рис. 5.11. Последняя панель установщика Grafana

После установки Grafana сразу будет запущена как служба, стартующая автоматически при загрузке Windows.

Если у вас на компьютере или сервере установлен брандмауэр или антивирус с функциями брандмауэра, необходимо разрешить входящие соединения TCP на порту 3000. На рис. 5.12 показан соответствующий запрос от антивируса Dr.Web, настроенного для работы в режиме обучения.

The screenshot shows the Dr.Web interface with the following details:

Брандмауэр
Обнаружена сетевая активность.

Отсутствует соответствующее сетевое правило для приложения.
Сетевое приложение запущено неизвестным процессом.

Имя приложения: grafana.exe
Путь: C:\Program Files\GrafanaLabs\grafana\bin\grafana.exe
Цифровая подпись: Не подписано
Адрес: tcp://:
Порт: 3000 (hbci, remoteware-cl)
Направление: Входящее

Кнопки: Создать правило..., Запретить однократно, Разрешить однократно

Рис. 5.12. Запрос на открытие порта 3000

Здесь вам нужно щелкнуть кнопку **Создать правило** и добавить два правила, показанных на рис. 5.13, разрешив приложениям Grafana сетевые подключения на порт 3000.

Брандмауэр

Новое правило для приложения.

Имя приложения: grafana.exe
Путь: C:\Program Files\GrafanaLabs\grafana\bin\grafana.exe
Цифровая подпись: Не подписано
Адрес: tcp://:
Порт: 3000 (hbci, remoteware-cl)
Направление: Входящее

Применить предустановленное правило

Разрешить приложению сетевые подключения на порт 3000 (hbci, ...)

Сетевое приложение запущено неизвестным процессом.

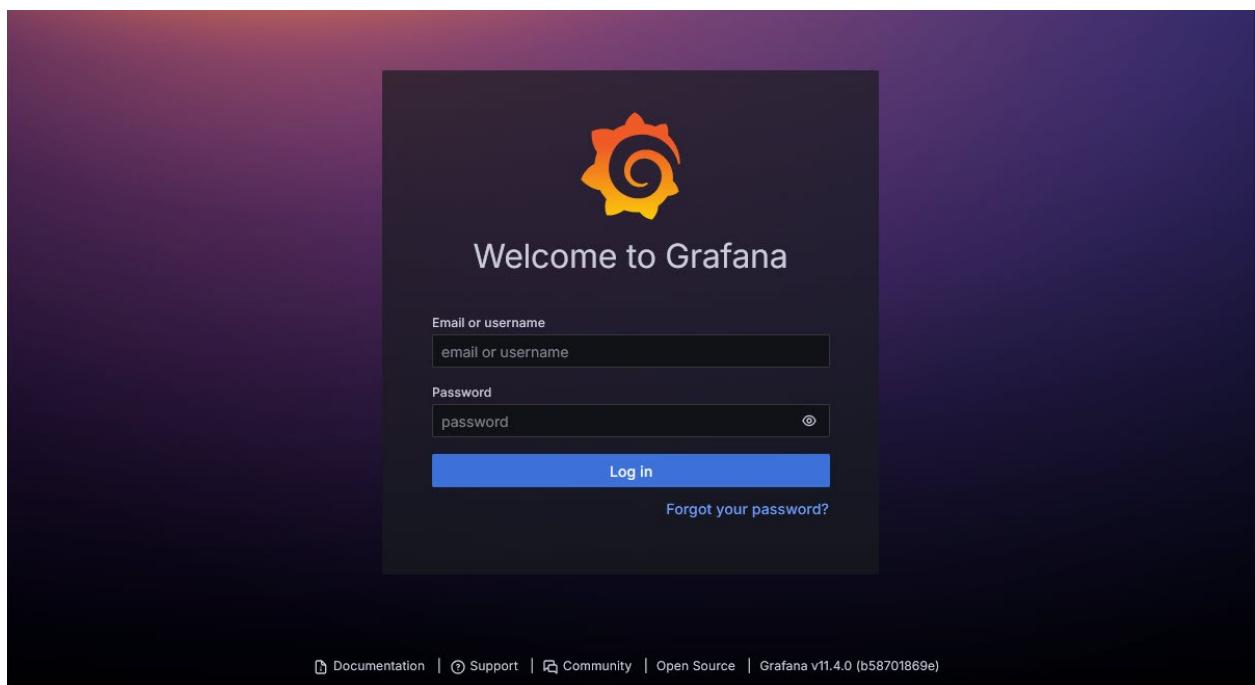
Приложение	Разр...	Забл...	Цифровая подпись	Путь
<input checked="" type="checkbox"/> grafana-server.exe	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/> Не подписано	C:\Program Files\Grafana...
<input checked="" type="checkbox"/> The non-sucking servi...	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="checkbox"/> Не подписано	C:\Program Files\Grafana...

OK

Отменить

Рис. 5.13. Правила для разрешения сетевых подключений на порт 3000

Когда вы это сделаете, откройте в браузере страницу по адресу <http://127.0.0.1:3000/login>. Появится страница с приглашением для входа в Grafana (рис. 5.16).

**Рис. 5.16. Страница с приглашением для входа в Grafana**

По умолчанию нужно указать логин admin и пароль admin. При первом входе вам будет рекомендовано изменить этот пароль.

Настройка Grafana

Перед тем как приступить к настройке Grafana, убедитесь, что соответствующий сервис запущен:

```
sudo /bin/systemctl status grafana-server
```

Добавление источника данных

Откройте главное меню и в разделе **Connections** выберите меню **Data sources** (рис. 6.1).

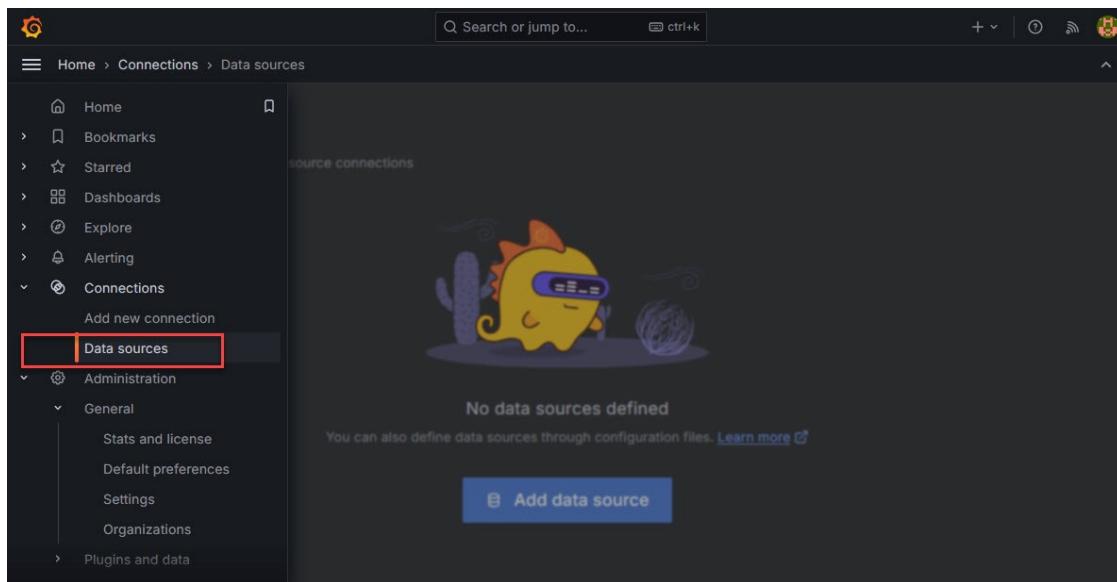


Рис. 6.1. Вход в раздел добавления источника данных

Щелкните здесь кнопку **Add data source** и выберите базу данных **InfluxDB** (рис. 6.2).

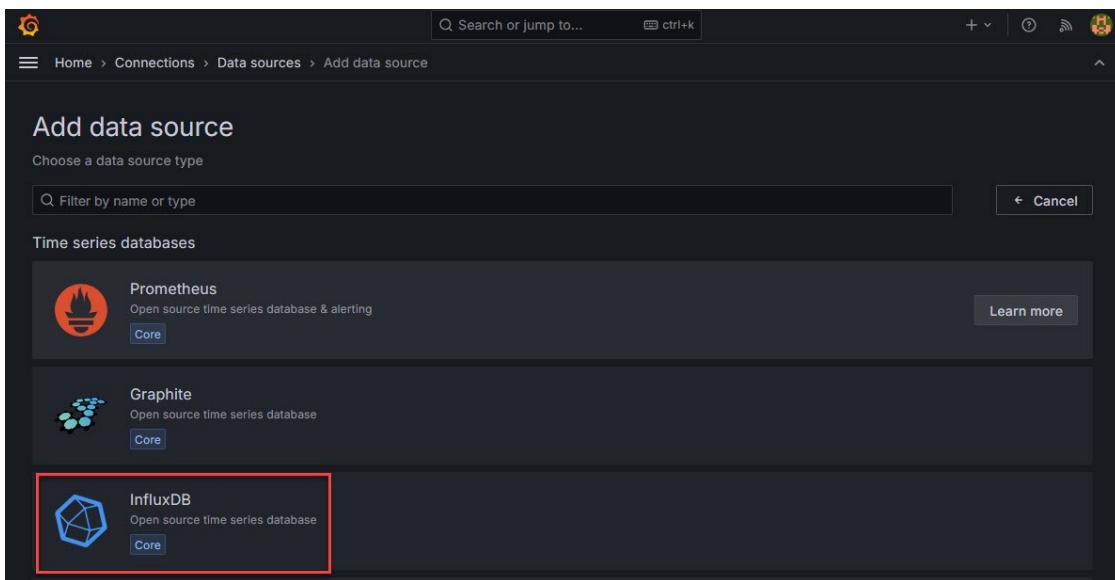


Рис. 6.2. Выбор базы данных InfluxDB

Появится страница настройки источника данных, верхняя часть которой показана на рис. 6.3.

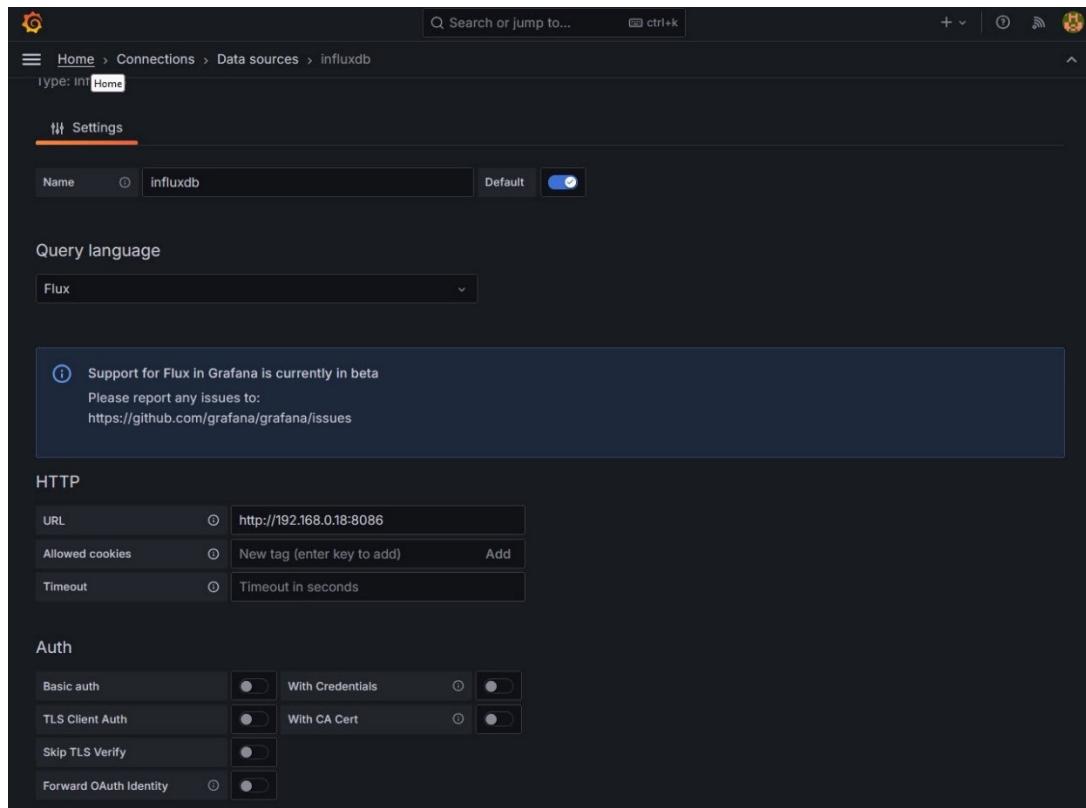


Рис. 6.3. Верхняя часть окна настройки источника данных

Здесь в списке **Query language** выберите строку **Flux**, а в поле **URL** введите адрес, по которому доступна база данных InfluxDB.

Если вы настраиваете Grafana для работы на модуле сборщика FCC-0801, то в поле **URL** адрес InfluxDB нужно указать как <http://127.0.0.1:8086>. Если Grafana настраивается для работы на другом узле, то вместо 127.0.0.1 укажите адрес IP, выделенный модулю FCC-0801.

Также отключите здесь режим **Basic auth** для того, чтобы использовать токены при авторизации в InfluxDB.

На рис. 6.4 показана нижняя часть окна настройки источника данных.

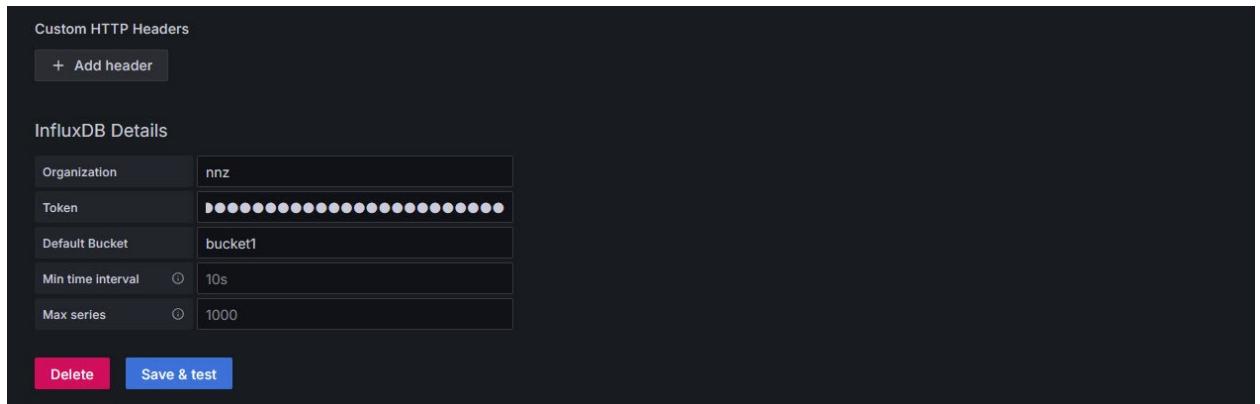


Рис. 6.4. Нижняя часть окна настройки источника данных

Здесь в поле **Organization** укажите строку «nzz», если будет подключение к узлу FCC-0801.

В поле **Token** необходимо ввести токен авторизации в InfluxDB, который можно узнать через командную строку модуля FCC-0801:

```
cat /etc/telegraf/telegraf.conf | grep token
```

Также вы можете запросить токен в службе сервисного обслуживания.

В поле **Default Bucket** следует ввести строку названия контейнера «bucket1».

Закончив заполнение перечисленных выше полей формы, щелкните кнопку **Save & test**. Если поля были заполнены верно, и Grafana смогла подключиться к источнику данных InfluxDB, в нижней части окна настройки появится сообщение об успехе (рис. 6.5) или сообщение об ошибке.

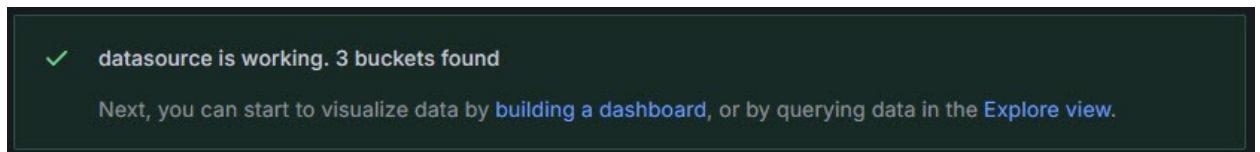


Рис. 6.5. Сообщение об успешном подключении к источнику данных

Подключение панелей Dashboard

Чтобы Grafana могла отображать данные измерений, полученные от датчиков, нужно создать панели Dashboard или подключить уже готовые.

Откройте главное меню Grafana и выберите строку **Dashboards** (рис. 6.6).

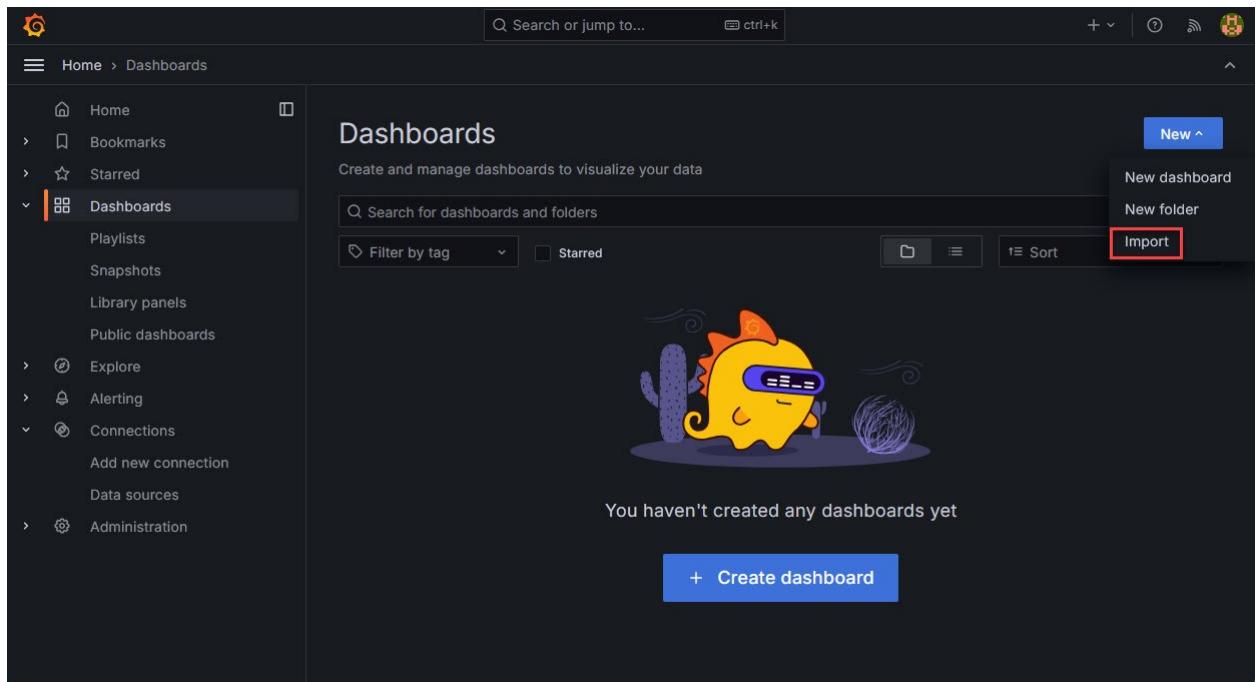


Рис. 6.6. Страница создания панелей Dashboard

Выберите здесь из меню **New** строку **Import** (выделена рамкой красного цвета на рис. 6.6).

Готовые шаблоны Grafana для датчиков вы можете найти в репозитории <https://github.com/lab240/telegraf-grafana-configs/tree/main/conf-grafana-dashboards>.

В качестве примера выберем шаблон датчика ICPDAS DL-303 (рис. 6.7).

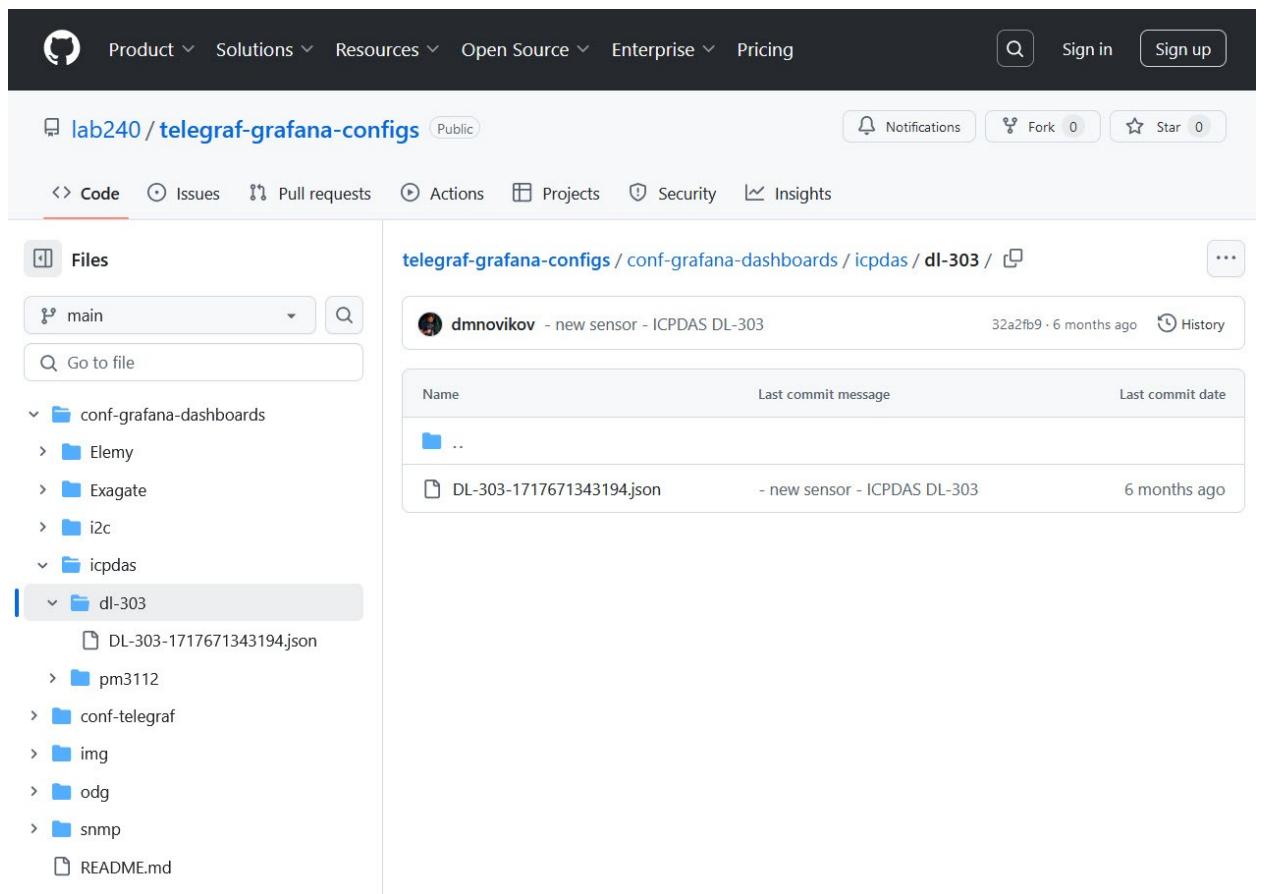


Рис. 6.7. Готовый шаблон Grafana для датчика ICPDAS DL-303

Вы можете скачать шаблон в виде файла, или загрузить его в Grafana по его адресу URL.

Щелкните имя файла шаблона, чтобы открыть его для просмотра (рис. 6.8).

The screenshot shows a GitHub repository page for 'lab240 / telegraf-grafana-configs'. The 'Code' tab is selected. On the left, there's a file tree under 'main' showing various configuration files like 'Elemy', 'Exagete', 'i2c', 'icpdas', and 'dl-303'. The file 'DL-303-1717671343194.json' is selected. The main pane displays the JSON content of this file:

```

1   {
2     "_inputs": [
3       {
4         "name": "DS_INFLUXDB-101010126",
5         "label": "InfluxDB-101010126",
6         "description": "",
7         "type": "datasource",
8         "pluginId": "influxdb",
9         "pluginName": "InfluxDB"
10      }
11    ],
12    "_elements": {},
13    "_requires": [
14      {
15        "type": "panel",
16        "id": "gauge",
17        "name": "Gauge",
18        "version": ""
19      },
20    ]
}

```

Рис. 6.8. Просмотр файла шаблона для датчика ICPDAS DL-303

Скачайте JSON-файл шаблона с помощью кнопки **Download raw file**. Загруженный файл загрузите в поле Upload dashboard JSON file (рис. 6.9).

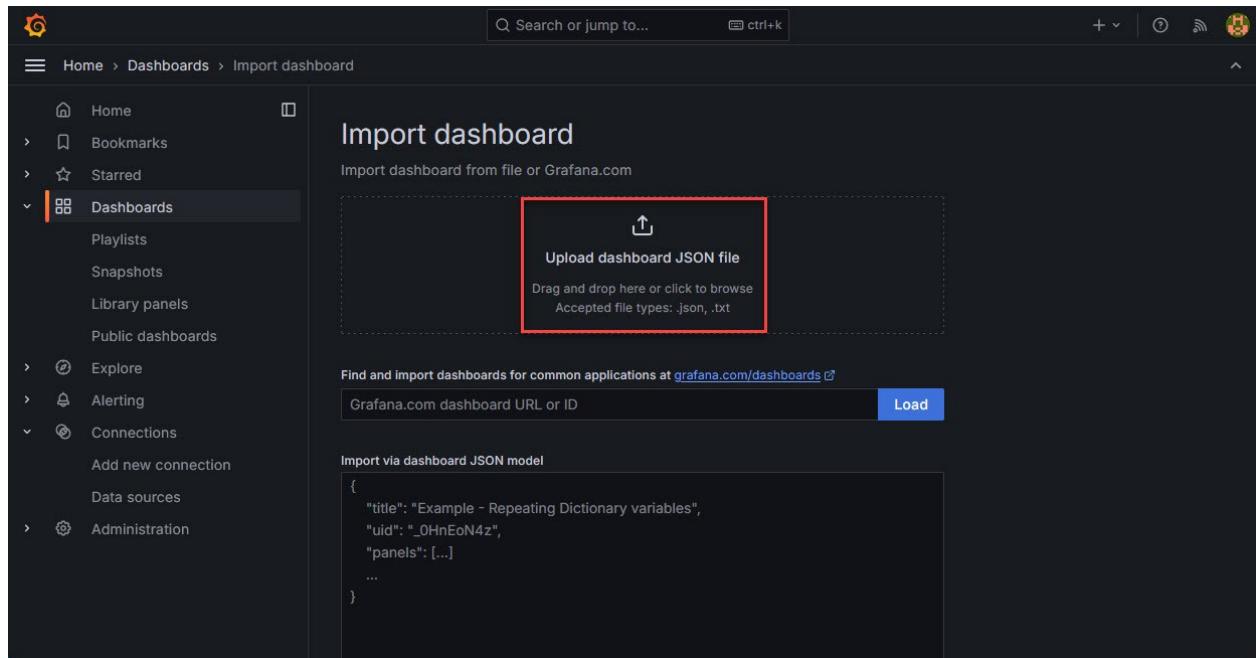


Рис. 6.9. Ввод адреса файла шаблона для импорта

После загрузки файла появится окно **Import dashboard** (рис. 6.10).

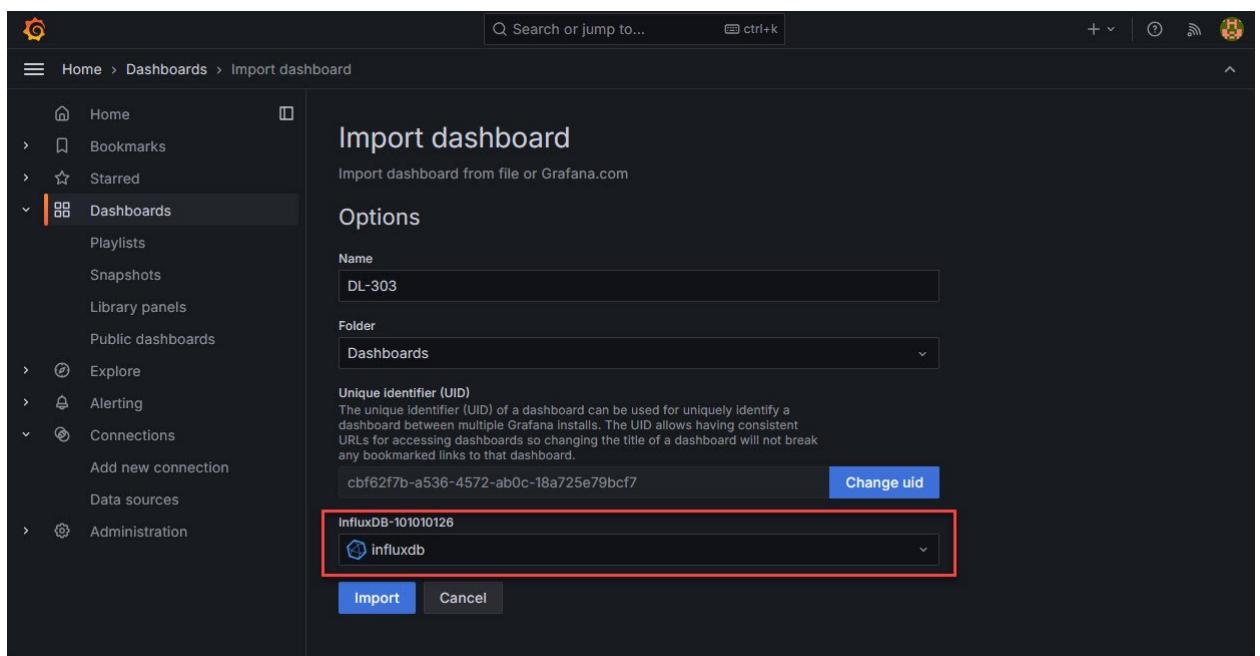


Рис. 6.10. Окно Import dashboard

Выберите источник данных influxdb, затем щелкните кнопку **Import**. Вы увидите импортированную панель, в которой будут отображаться данные измерений от датчика ICPDAS DL-303 (рис. 6.11).



Name	Last *	Min	Mean	Max
CO	0.00 ppm	0.00 ppm	0.00 ppm	0.00 ppm
CO2	558.00 ppm	552.00 ppm	557.45 ppm	566.00 ppm

Рис. 6.11. Панель с данными измерений от датчика ICPDAS DL-303

Вы можете просмотреть и отредактировать шаблон, находясь в этой панели. Например, откроем редактор для просмотра и редактирования CO.

Чтобы это сделать, щелкните кнопку с тремя точками в блоке CO, а затем выберите из контекстного меню строку **Edit** (рис. 6.12).

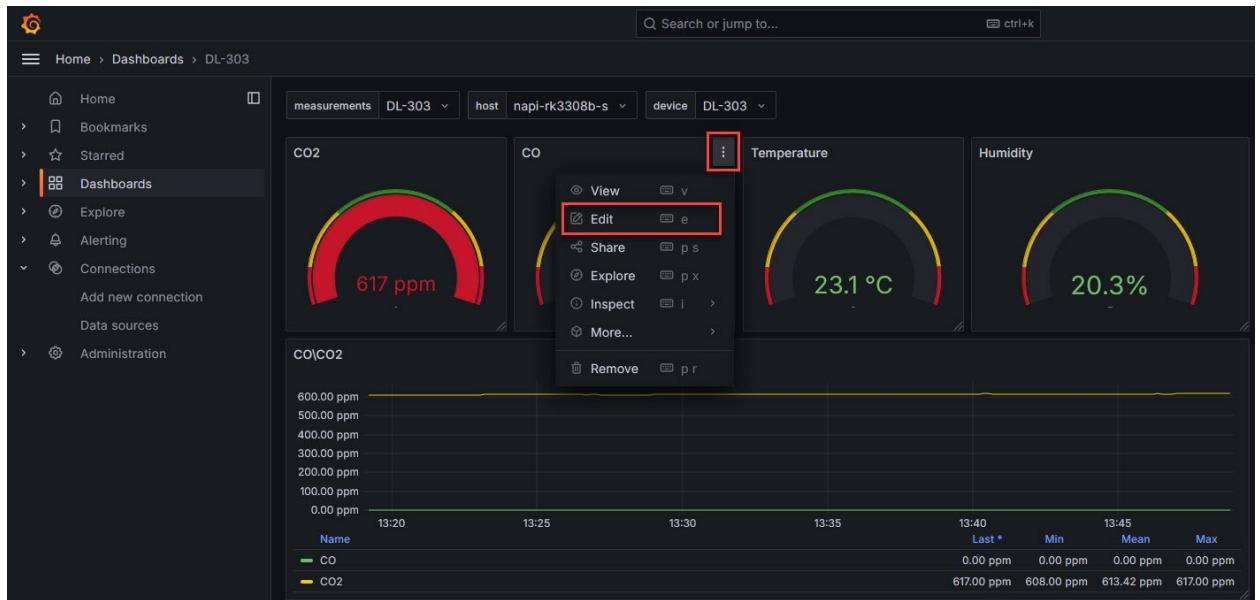


Рис. 6.12. Редактирование шаблона для блока CO

В нижней части окна находится запрос на языке Flux, соответствующий отображению панели CO в Dashboard (рис. 6.13).

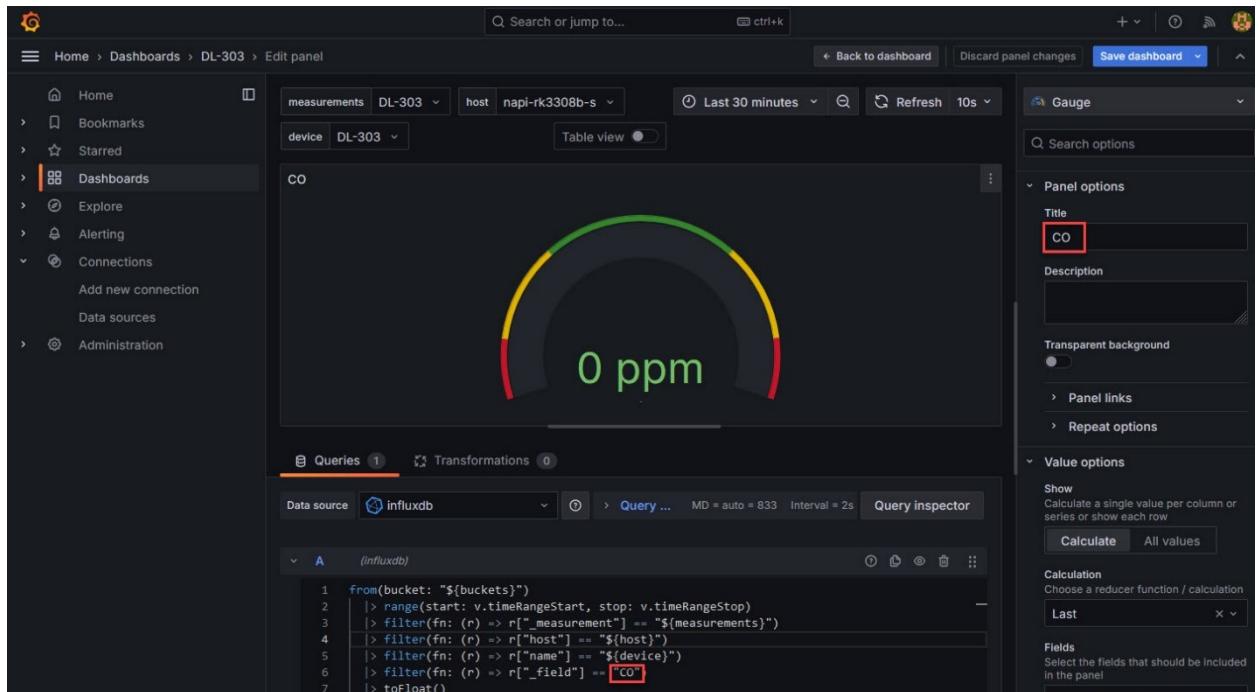


Рис. 6.13. Редактирование шаблона панели для CO

Здесь в блоке **Panel options** вы можете отредактировать параметры шаблона, например, заголовок.

В блоке **Standard options** можно изменить единицы измерения Unit, которые используются в панели для отображения данных (рис. 6.14).

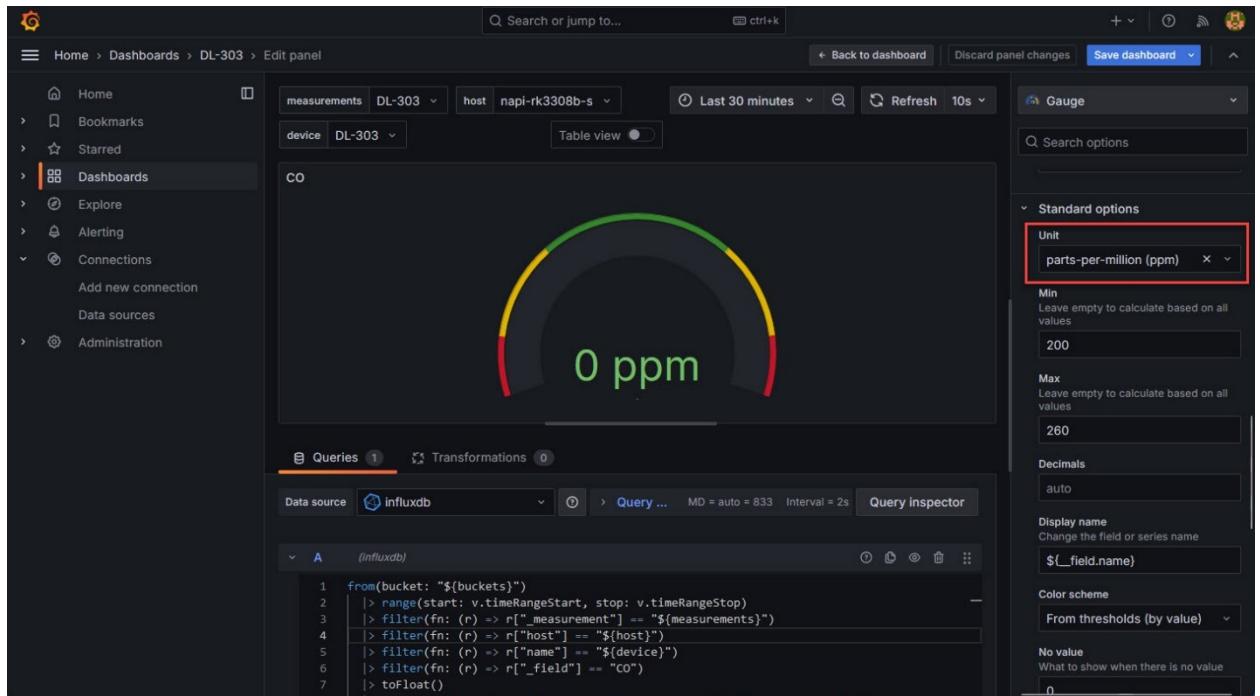


Рис. 6.14. Изменение единиц измерения для отображения данных

Подключение и мониторинг датчика Modbus RTU через RS485

В этом разделе мы расскажем о подключении датчиков Modbus RTU через интерфейс RS485. Последовательность действий такая:

- поиск описания датчика (даташита) с информацией о регистрах Modbus;
- подключение датчика к RS485;
- тестирование датчика утилитой mbpoll;
- поиск готового шаблона Telegraf или создание нового;
- подключение шаблона к Telegraf и тестирование получения от него данных

Поиск описания датчика

В большинстве случаев описание датчика поставляется вместе с самим датчиком. Если в вашем случае это не так, попробуйте найти его в интернете по названию.

В качестве примера приведем таблицу регистров датчика PZEM-014/016 AC communication module, доступное по адресу <https://smacont.ru/applying/015/PZEM-014-016-Datasheet-User-Manual.pdf> (рис. 7.1).

Register address	Description	Resolution
0x0000	Voltage value	1LSB correspond to 0.1V
0x0001	Current value low 16 bits	1LSB correspond to 0.001A
0x0002	Current value high 16 bits	
0x0003	Power value low 16 bits	
0x0004	Power value high 16 bits	1LSB correspond to 0.1W
0x0005	Energy value low 16 bits	
0x0006	Energy value high 16 bits	1LSB correspond to 1Wh
0x0007	Frequency value	1LSB correspond to 0.1Hz
0x0008	Power factor value	1LSB correspond to 0.01
0x0009	Alarm status	0xFFFF is alarm, 0x0000 is not alarm

Рис. 7.1. Описание регистров датчика PZEM-014/016 AC communication module

Здесь перечислены регистры Input Registers, содержащие результаты измерений.

Подключение датчика к RS485

Прежде всего, нужно найти на сборщике Front Control Compact и на датчике контакты интерфейса RS485. На сборщике эти контакты обозначены как **A** и **B** (рис. 7.2).

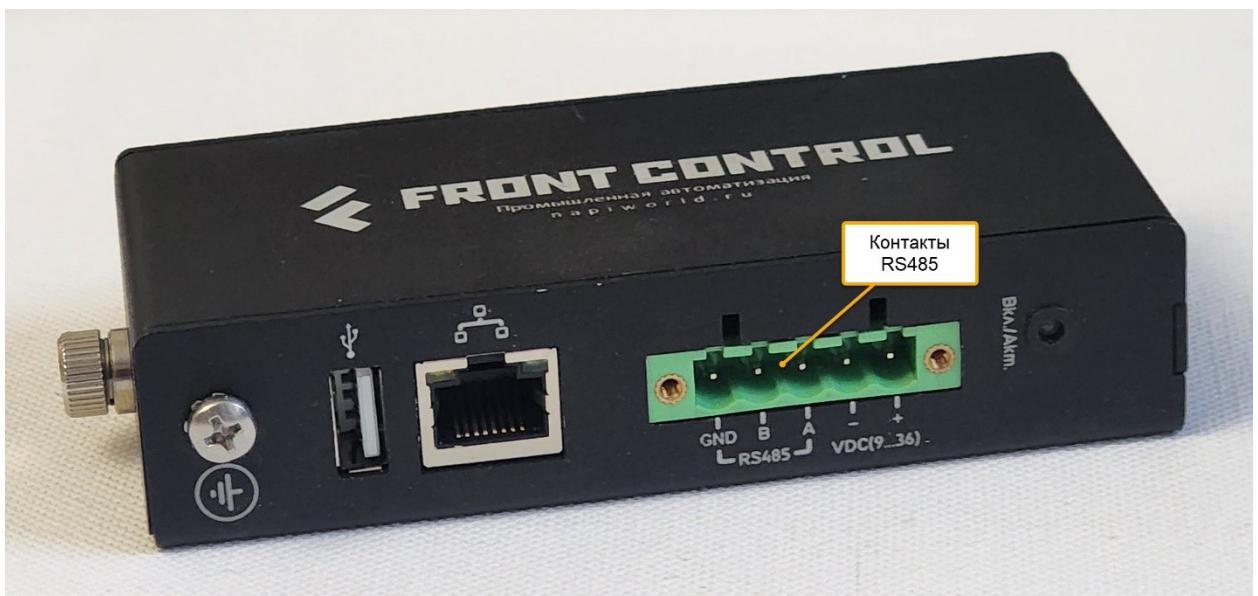


Рис. 7.2. Контакты RS485 сборщика Front Control Compact

Контакты RS485 на датчике могут обозначаться как А и В (рис. 8.3), а также как D+ и D- (рис. 7.4).

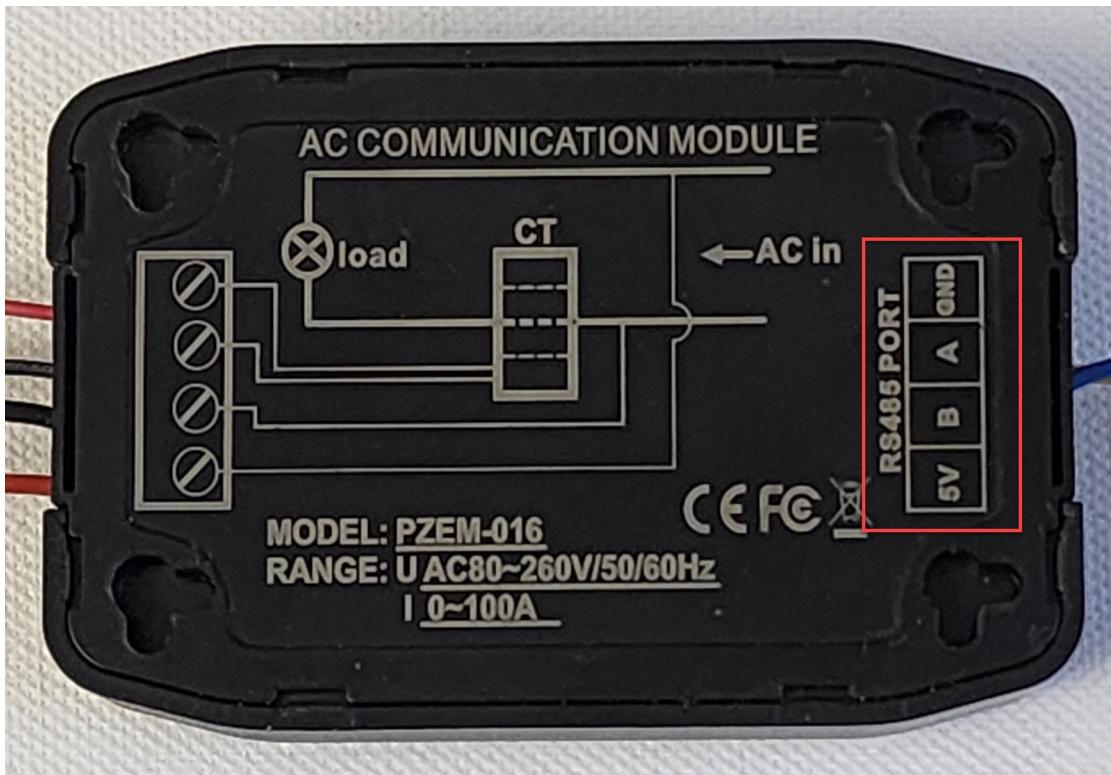


Рис. 7.3. Контакты RS485 датчика PZEM-016

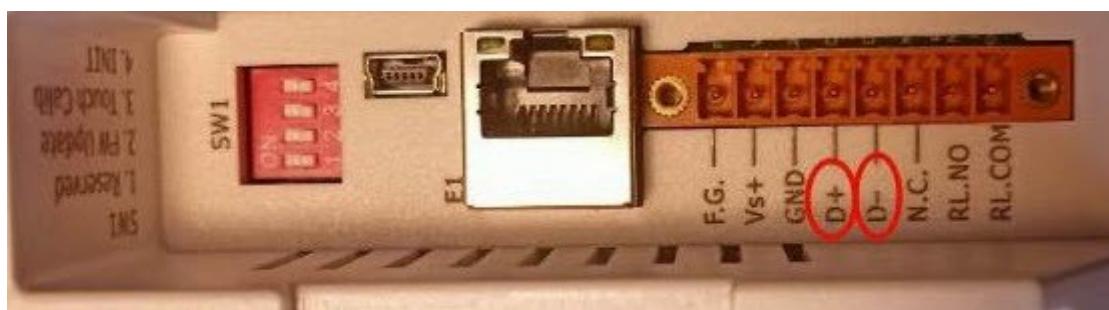


Рис. 7.4. Контакты RS485 датчика ICPDAS DL-303

Контакт А сборщика необходимо подключить к контакту А или D+ датчика, а контакт В — к контакту В или D- датчика:

- контакт А — контакт А или D+
- контакт В — контакт В или D-

Если датчик и сборщик подключены к различным источникам питания, а также при длинных линиях, рекомендуется соединять вместе контакты **GND** сборщика и датчика.

На рис. 7.5 показано подключение датчика XY-MD02 к сборщику Front Control Compact.

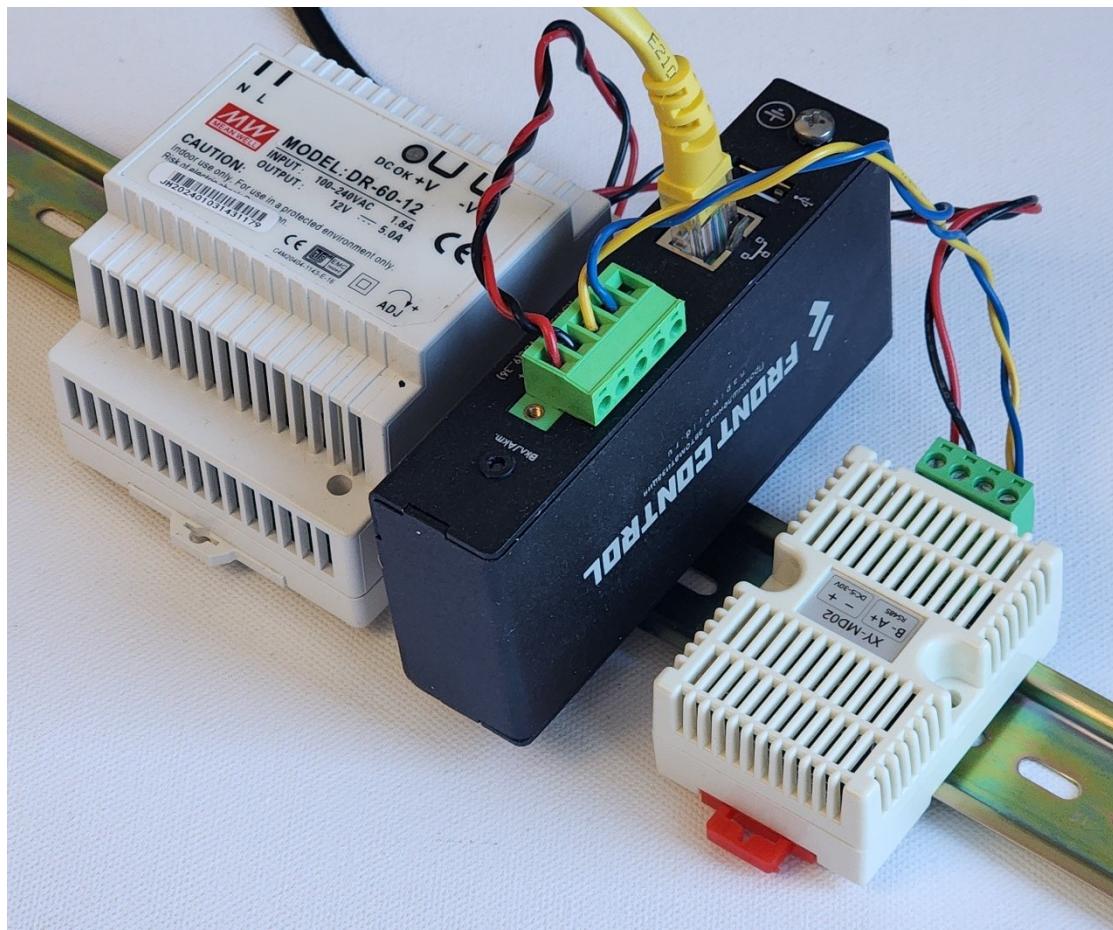


Рис. 7.5. Подключение датчика XY-MD02 к сборщику Front Control Compact

Тестирование датчика утилитой mbpoll

После подключения выполните тестирование датчика по протоколу Modbus RTU утилитой mbpoll.

Инструкция по установке mbpoll, а также описание ее параметров опубликованы на сайте <https://github.com/epsilononrt/mbpoll>.

Тестирование выполняется при подключении к сборщику Front Control Compact в консольном режиме через SSH. Вы можете использовать с этой целью программу PuTTY, MobaXterm или аналогичную.

Для подключения используйте определенный ранее адрес IP модуля Front Control Compact и пользователя root. По умолчанию для этого пользователя задан пароль napilinux (смените его при первой возможности).

Датчик PZEM-016

Датчик PZEM-016 AC communication module может измерять напряжение переменного тока, величину тока, мощность, частоту и потребляемую энергию.

Для тестирования PZEM-016 используйте такую команду:

```
# mbpoll -m rtu -a 1 -r 0 -c 10 -t 3 -b 9600 -P none -s 1 -d 8 -1 -0 -v /dev/ttyS3
debug enabled
Set device=/dev/ttyS3
mbpoll 1.0-0 - ModBus(R) Master Simulator
Copyright (c) 2015-2023 Pascal JEAN,
https://github.com/epsilononrt/mbpoll
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'mbpoll -w' for details.

Opening /dev/ttyS3 at 9600 bauds (N, 8, 1)
Set response timeout to 1 sec, 0 us
Protocol configuration: ModBus RTU
Slave configuration....: address = [1]
                        start reference = 0, count = 10
Communication.....: /dev/ttyS3,          9600-8N1
                    t/o 1.00 s, poll rate 1000 ms
Data type.....: 16-bit register, input register table
```

```
-- Polling slave 1...
[01][04][00][00][0A][70][0D]
Waiting for a confirmation...
<01><04><14><08><DE><00><BC><00><00><01><AA><00><00><00><B9><00><00><00><01><F4><00><64><00><00><60><31>
[0]:      2270
[1]:      188
[2]:       0
[3]:     426
[4]:       0
[5]:     185
[6]:       0
[7]:     500
[8]:     100
[9]:       0
```

Здесь используются такие параметры команды mbpoll:

- **-m rtu** — режим передачи данных Modbus RTU (Remote Terminal Unit);
- **-a 1** — адрес устройства Modbus равен 1;
- **-r 0** — чтение начинается с регистра 0;
- **-c 10** — будет прочитано содержимое десяти регистров;
- **-t 3** — чтение регистров Input Registers;
- **-b 9600** — скорость передачи данных 9600 бод;
- **-P none** — при передаче данных не используется контроль четности;
- **-s 1** — используется 1 стоп-бит;
- **-d 8** — используется 8 бит данных;
- **-1** — команда должна выполняться один раз (однократное чтение);
- **-0** — используется нумерация регистров, начиная с нуля;
- **-v** — включается режим отладки с выводом дополнительной информации о выполнении команды (отправленные и принятые данные в шестнадцатеричном коде);
- **/dev/ttyS3** — связь с устройством будет осуществляться через порт /dev/ttyS3

Обратите внимание, что параметр **-v** включает отладочный режим. В этом режиме на консоль выводятся бинарные данные команды, отправляемой в датчик, а также данные, которые датчик отправляет в сборщик:

```
[01][04][00][00][0A][70][0D]
Waiting for a confirmation...
<01><04><14><08><DE><00><BC><00><00><01><AA><00><00><00><B9><00><00><00><01><F4><00><64><00><00><60><31>
```

Их можно сверить с даташитом датчика, доступным на сайте
<https://smacont.ru/applying/015/PZEM-014-016-Datasheet-User-Manual.pdf>.

Мы получили следующие значения от команды mbpoll:

- Напряжение: 227.0 В
- Ток: 0.188 А
- Мощность: 42.6 Вт
- Энергия: 185 Вт*ч
- Частота: 50.0 Гц
- Коэффициент мощности: 1.00

Датчик XY-MD02

Для тестирования датчика XY-MD02 запустите утилиту mbpoll в отладочном режиме:

```
# mbpoll -m rtu -a 1 -r 2 -c 2 -t 3 -b 9600 -P none -s 1 -d 8 -l -v
/dev/ttyS3
debug enabled
Set device=/dev/ttyS3
mbpoll 1.0-0 - ModBus(R) Master Simulator
Copyright (c) 2015-2023 Pascal JEAN,
https://github.com/epsilononrt/mpoll
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'mbpoll -w' for details.

Opening /dev/ttyS3 at 9600 bauds (N, 8, 1)
Set response timeout to 1 sec, 0 us
Protocol configuration: ModBus RTU
Slave configuration...: address = [1]
                      start reference = 2, count = 2
Communication.....: /dev/ttyS3,          9600-8N1
                     t/o 1.00 s, poll rate 1000 ms
Data type.....: 16-bit register, input register table

-- Polling slave 1...
[01][04][00][01][00][02][20][0B]
Waiting for a confirmation...
<01><04><04><00><F3><00><FF><4B><F7>
[2]:    243
[3]:    255
```

Здесь для команды mbpoll заданы следующие параметры:

- **-m rtu** — режим передачи данных Modbus RTU (Remote Terminal Unit);
- **-a 1** — адрес устройства Modbus равен 1;
- **-r 2** — чтение начинается с регистра 2;
- **-c 2** — будет прочитано содержимое двух регистров;
- **-t 3** — чтение регистров Input Registers;
- **-b 9600** — скорость передачи данных 9600 бод;
- **-P none** — при передаче данных не используется контроль четности;

- `-s 1` — используется 1 стоп-бит;
- `-d 8` — используется 8 бит данных;
- `-1` — команда должна выполняться один раз (однократное чтение);
- `-v` — включается режим отладки с выводом дополнительной информации о выполнении команды (отправленные и принятые данные в шестнадцатеричном коде);
- `/dev/ttyS3` — связь с устройством будет осуществляться через порт `/dev/ttyS3`

Значения, полученные от датчика XY-MD02, можно интерпретировать следующим образом:

- Температура: 24.3 °C
- Влажность: 25.5 %

При тестировании вы можете сверить данные, полученные от команды `mbpoll`, с описанием из даташита. Даташит для датчика XY-MD02 можно найти на сайте <https://www.idbsmart.cz/wp-content/uploads/2024/05/xy-md02-manual.pdf>.

Датчик ICPDAS DL-303

По умолчанию адрес всех датчиков на шине Modbus равен 1. Некоторые датчики позволяют конфигурировать свой адрес на шине Modbus, а также параметры порта RS485. Для адреса возможны значения от 1 до 255.

Например, датчик ICPDAS DL-303 позволяет задавать адрес и параметры порта RS485 (рис. 7.6).

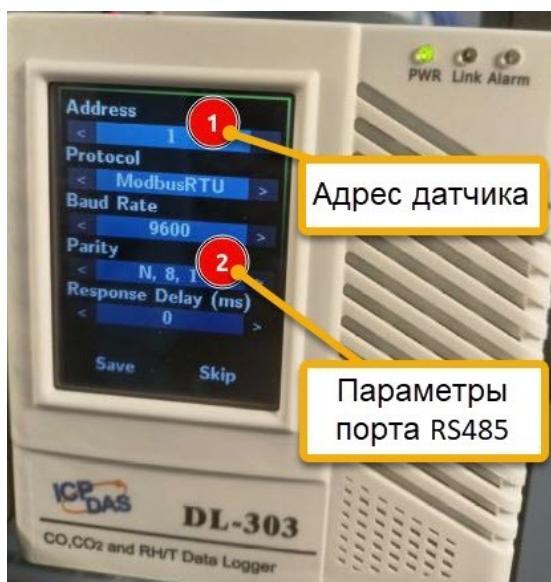


Рис. 7.6. Настройка адреса датчика ICPDAS DL-303 и параметров порта RS485

В датчике ICPDAS DL-303 для чтения доступны следующие регистры Input Registers:

- 1 — CO (ppm);
- 2 — CO2 (ppm);
- 3 — H (влажность в %);
- 4 — T (температура в C);
- 5 — T (температура в F);

Для тестирования используйте команду mbpoll вида:

```
# mbpoll -b 9600 -P none -d 8 -m rtu -a 1 -r 1 -c 4 -t 3 /dev/ttys3
```

Результат выполнения этой команды показан на рис. 7.7.

```
root@napi-armbian:~# mbpoll -b 9600 -P none -d 8 -m rtu -a 1 -r 1 -c 4 -t 3 /dev/ttys3
mbpoll 1.0-0 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright © 2015-2019 Pascal JEAN, https://github.com/epsilonrt/mpoll
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'mbpoll -w' for details.

Protocol configuration: Modbus RTU
Slave configuration...: address = [1]
                      start reference = 1, count = 4
Communication.....: /dev/ttys3,      9600-8N1
                     t/o 1.00 s, poll rate 1000 ms
Data type.....: 16-bit register, input register table

-- Polling slave 1... Ctrl-C to stop)
[1]:    1
[2]:   920
[3]: 2731
[4]: 2499
```

Рис. 7.7. Тестирование датчика ICPDAS DL-303

Для отладки вы также можете указать команде mbpoll параметр -v.

Ошибки при тестировании

В процессе тестирования утилитой mbpoll вы можете получать различные сообщения об ошибках. Как правило, ошибки связаны с неправильным указанием параметров порта связи, адреса датчика или номера регистра. Для коррекции рекомендуем обратиться к описанию (даташиту) датчика.

Рассмотрим типичные ошибки и ситуации, в которых они возникают.

- **Read input register failed: Connection timed out**

Ошибка Connection timed out может возникать, если команде mbpoll указан неправильный порт например:

```
# mbpoll -m rtu -a 1 -r 0 -c 10 -t 3 -b 9600 -P none -s 1 -d 8 -1 -0 -
v /dev/ttys1
```

Эта же ошибка может появиться, если указать неверный адрес датчика:

```
# mbpoll -m rtu -a 2 -r 0 -c 10 -t 3 -b 9600 -P none -s 1 -d 8 -1 -0 -v /dev/ttyS3
```

- **Read input register failed: Invalid CRC**

Ошибка Invalid CRC появится, если указать неправильные параметры порта, например:

```
# mbpoll -m rtu -a 1 -r 0 -c 10 -t 3 -b 10600 -P none -s 1 -d 8 -1 -0 -v /dev/ttyS3
```

- **Read input register failed: Illegal data address**

Ошибка Illegal Data Address появится, если указать неправильный адрес регистра датчика, неправильный тип регистра (Holding вместо Input) или неправильное количество регистров, которые нужно прочитать:

```
# mbpoll -m rtu -a 1 -r 7 -c 10 -t 3 -b 9600 -P none -s 1 -d 8 -1 -0 -v /dev/ttyS3
```

Создание шаблона Telegraf

Прежде чем создавать собственный шаблон для датчика, поищите готовый на сайте <https://github.com/lab240/telegraf-grafana-configs/tree/main/conf-telegraf>. Процедура создания и редактирования шаблона описана в разделе **Подключение и мониторинг датчиков**.

Ниже представлен шаблон датчика XY-MD02 для Telegraf:

```
## XY_MD02 Temperature\Humidity) Modbus RTU
[[inputs.modbus]]
  name = "XY_MD02" # Название устройства
  slave_id = 1
  timeout = "10s"

  # busy_retries = 0
  # busy_retries_wait = "100ms"

  controller = "/dev/ttyS3"
  baud_rate = 9600
  data_bits = 8
  parity = "N"
  stop_bits = 1
  transmission_mode = "RTU"

  # Список регистров для чтения
  input_registers = [
    { measurement="XY_MD02-rtu", name = "Temperature", byte_order =
      "AB", data_type = "UINT16", scale=1.0, address = [2] },
```

```
        { measurement="XY_MD02-rtu", name = "Humidity", byte_order =
"AB", data_type = "UINT16", scale=1.0, address = [3] },
]
```

Аналогично, для датчика PZEM-016 вы можете использовать такой шаблон Telegraf:

```
## PZEM-016 AC communication module
[[inputs.modbus]]
  name = "pzem016"
  controller = "file:///dev/ttyS3"
  baud_rate = 9600
  data_bits = 8
  parity = "N"
  stop_bits = 1
  transmission_mode = "RTU"

  slave_id = 1
  timeout = "5s"

  input_registers = [
    { measurement="pzem016-modbus-rtu", name = "Volt_a", byte_order =
"AB", data_type = "UINT16", scale=0.1, address = [0] },
    { measurement="pzem016-modbus-rtu", name = "Current", byte_order =
"AB", data_type = "UINT16", scale=0.001, address = [1] },
    { measurement="pzem016-modbus-rtu", name = "Power", byte_order =
"AB", data_type = "UINT16", scale=0.1, address = [3] },
    { measurement="pzem016-modbus-rtu", name = "Energy", byte_order =
"AB", data_type = "UINT16", scale=1.0, address = [5] },
    { measurement="pzem016-modbus-rtu", name = "Frequency",
byte_order = "AB", data_type = "UINT16", scale=0.1, address =
[7] },
    { measurement="pzem016-modbus-rtu", name = "Power factor",
byte_order = "AB", data_type = "UINT16", scale=1.0, address =
[8] },
    { measurement="pzem016-modbus-rtu", name = "Alarm status",
byte_order = "AB", data_type = "UINT16", scale=1.0, address =
[9] },
  ]
  # { measurement="pzem016-modbus-rtu", name = "Power", byte_order =
"AB", data_type = "UINT16", scale=1.0, address = [4] },
  # { measurement="pzem016-modbus-rtu", name = "Energy", byte_order =
"AB", data_type = "UINT16", scale=1.0, address = [6] },
  # { measurement="pzem016-modbus-rtu", name = "Frequency",
byte_order = "AB", data_type = "UINT16", scale=1.0, address =
[8] },
  # { measurement="pzem016-modbus-rtu", name = "Power factor",
byte_order = "AB", data_type = "UINT16", scale=1.0, address =
[12] },
```

Сохранив шаблон, проверьте его и щелкните кнопку **Объединить с оригиналом**.

Оглавление

Первое включение модуля FCC-0801.....	2
Настройка сетевого интерфейса.....	3
Использование Интерфейс роутера.....	3
Поиск через сканирование сети.....	6
Установка статического IP-адреса в FCC-0801.....	11
Подключение FCC-0801 через консоль.....	15
Работа с консолью в ОС Microsoft Windows.....	17
Работа с консолью в ОС Linux.....	21
Назначение статического адреса IP для FCC-0801.....	22
Начальная настройка общих параметров FCC-0801.....	23
Настройка общих параметров.....	24
Настройка сетевых параметров.....	28
Активация датчика System.....	30
Просмотр графика нагрузки системы.....	31
Просмотр графиков для выбранных датчиков.....	32
Подключение и мониторинг датчиков.....	33
Архитектура работы с датчиками модуля FCC-0801.....	33
Упрощенная архитектура ПО работы с датчиками модуля FCC-0801.....	34
Шаблоны плагинов Telegraf.....	36
Как найти нужные плагины Telegraf.....	39
Шаблоны датчиков Modbus RTU.....	42
Редактирование и тестирование шаблона.....	45
Шаблоны датчиков ModbusTCP.....	47
Настройка шлюза Modbus RTU - Modbus TCP.....	49
Общие сведения.....	49
Активируйте службу mbusd.....	50
Настройка параметров шлюза mbusd.....	51
Проверка работы шлюза: читаем датчик.....	53
Решение проблем.....	54

Установка и запуск Grafana.....	56
Включение Grafana в сборщике FCC-0801.....	56
Установка Grafana из пакета в Ubuntu и Debian.....	57
Установка для процессоров ARM.....	58
Запуск Grafana в Docker.....	59
Установка Grafana в Microsoft Windows.....	60
Настройка Grafana.....	67
Добавление источника данных.....	67
Подключение панелей Dashboard.....	70
Подключение и мониторинг датчика Modbus RTU через RS485.....	75
Поиск описания датчика.....	76
Подключение датчика к RS485.....	76
Тестирование датчика утилитой mbpoll.....	79
Датчик PZEM-016.....	79
Датчик XY-MD02.....	81
Датчик ICPDAS DL-303.....	82
Ошибки при тестировании.....	83
Создание шаблона Telegraf.....	84
Оглавление.....	86