**Отчет**

«Работа с данными с использованием Node-red»

Выполнили:

Студенты группы M23-DM-01

АНО ВО «Университет Иннополис»

Васильев Дмитрий

Коваль Виктория

Салун Марина

Содержание

[Введение 3](#_Toc151567205)

[Этап 1. Визуализация 4](#_Toc151567206)

[Этап 2. Сохранение данных 11](#_Toc151567207)

[Этап 3. Аналитика данных 11](#_Toc151567209)

[Этап 4. Уведомление 14](#_Toc151567210)

[Ответственные за работы 19](#_Toc151567210)

Введение

В современном мире большое значение приобретает эффективная работа с данными из различных источников.

Node-red представляет собой графический инструмент для программирования, который позволяет в удобной форме работать с данными, интегрировать различные устройства и создавать потоки информации.

Цель работы:

* Изучить систему для работы с данными через Node-red с целью упрощения обработки, визуализации и анализа информации из различных источников.

Задачи данной работы:

* Изучить принципы работы Node-red и его возможности для работы с данными;
* Создать виртуальную среду с установкой Node-red для проведения экспериментов и тестирования функционала;
* Интеграция с различными источниками данных: базы данных, датчики, устройства IoT;
* Разработать сценарии обработки и агрегации данных с использованием встроенных и пользовательских узлов;
* Визуализировать и представить данные с использованием различных типов интерфейсов: графики, диаграммы и текстовые данные;
* Сохранить полученные данные;
* Провести аналитику данных производительности и эффективность системы;
* Настроить уведомления, сообщающих об ошибках.

Этап 1. Визуализация

* 1. Развернули в облаке Node-Red

**Node-Red** (инструмент потокового программирования) – это инструмент визуального программирования для интернета вещей, позволяющий подключать друг к другу устройства, API и онлайн-сервисы.

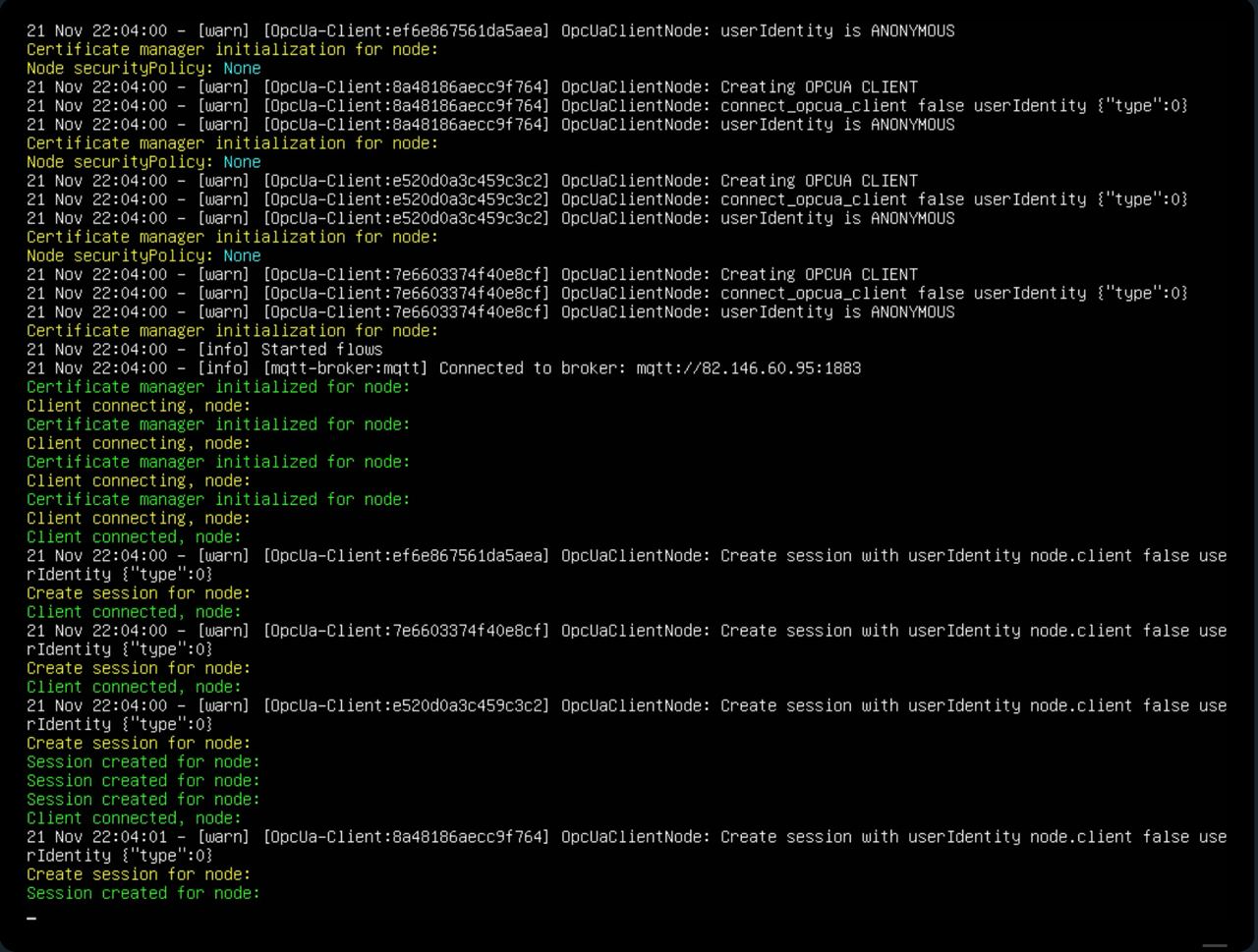


Рис. №1 Консоль. Стартовый запуск Node-red

* 1. Перешли к созданию web-страницы с тестовым OPC UA-сервером (несколько изменяющихся параметров).

**OPC UA (Open Platform Communications Unified Architecture)** – это стандартное протокольное средство для обмена информацией между различными устройствами. Создание web-страницы с тестовым OPC UA-сервером позволяет отображать параметры, полученные с устройств, на веб-интерфейсе. Это позволяет мониторить и контролировать процессы удаленно.

Создание web-страницы с тестовым OPC UA-сервером, изменяющимся параметрами, включает в себя несколько этапов:

* Настройка и подключение к тестовому серверу OPC UA-сервера;
* Создание web-страницы: разработка и создание web-страницы, которая будет отображать данные, полученные от OPC UA-сервера;
* Подключение к OPC UA-серверу: настройка web-страницы для подключения к OPC UA-серверу;
* Отображение изменяющихся параметров;

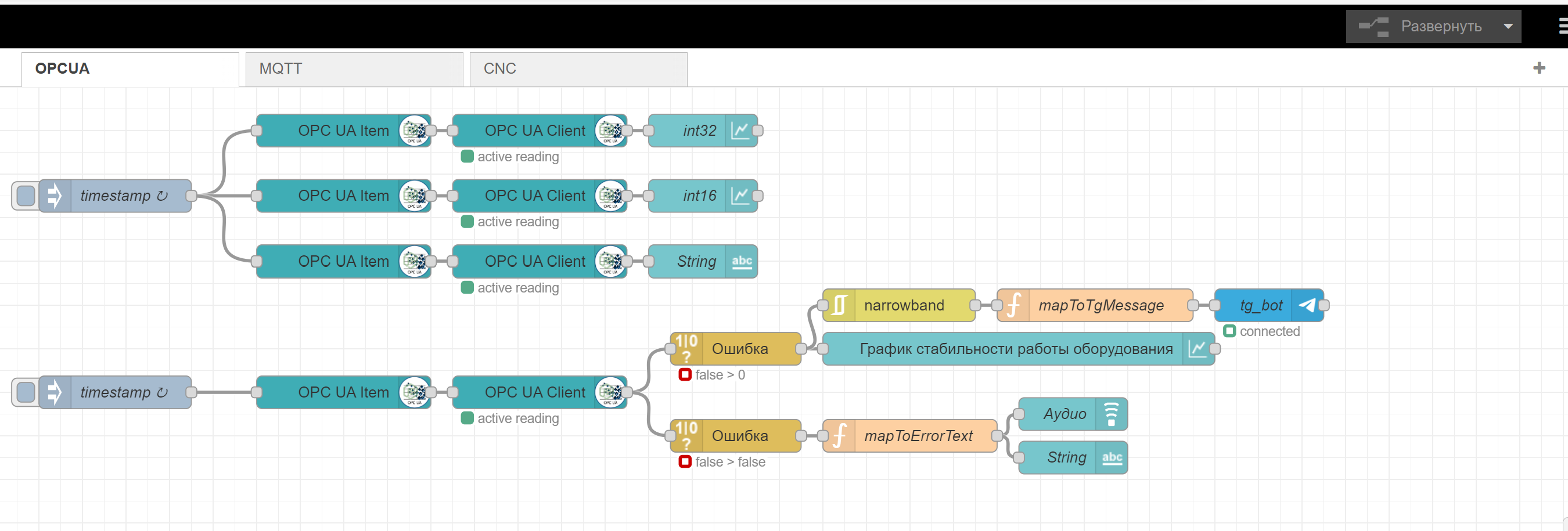


Рис. №2 Поток OPC UA сервера

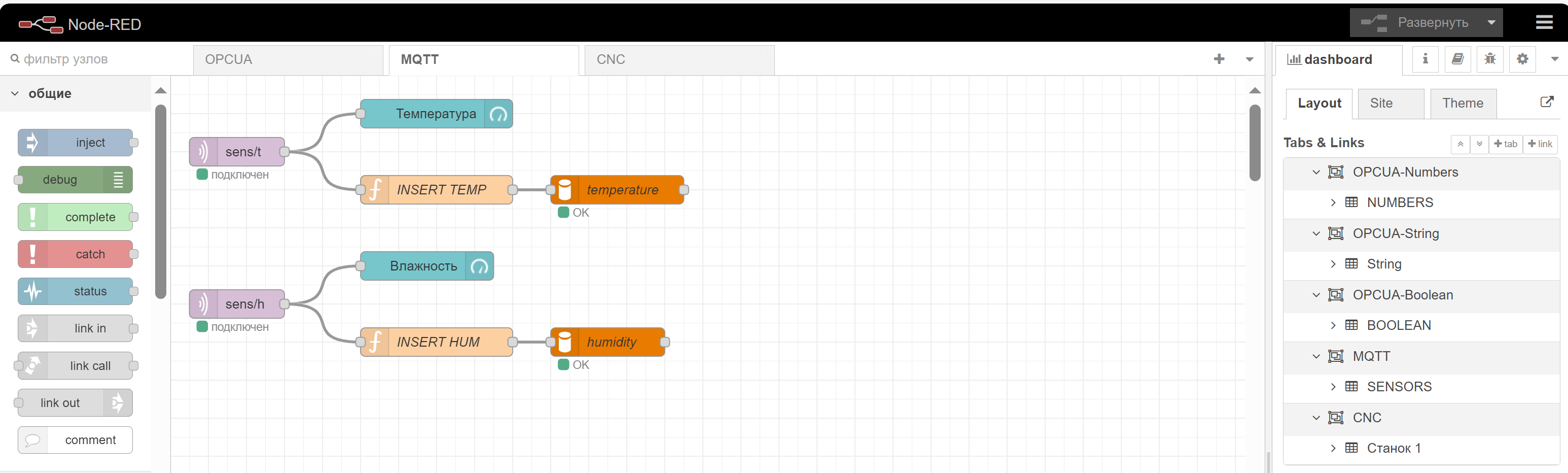


Рис. №3 Поток MQTT сервера

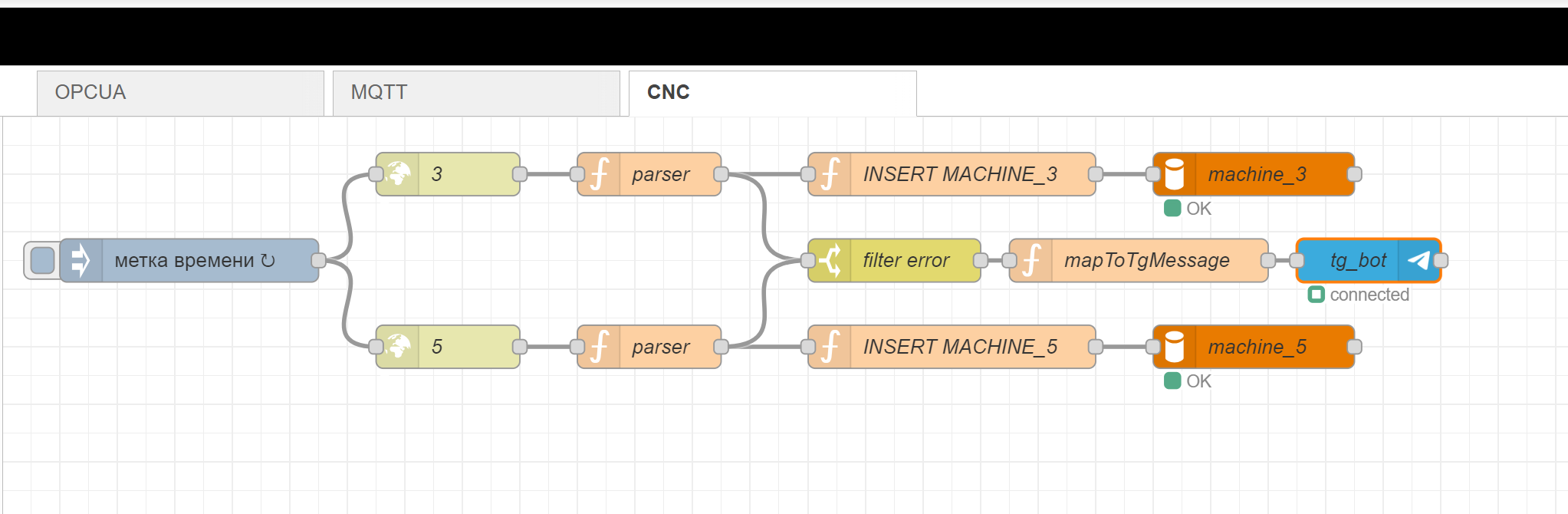


Рис. №4 Поток CNC сервера

С помощью Node-Red вывели графики для двух чисел и строки, так же с помощью bool-го значения сэмулировали ошибку, которую вывели текстом и в аудио формате, превратив сигнал ошибки в слово «ошибка».

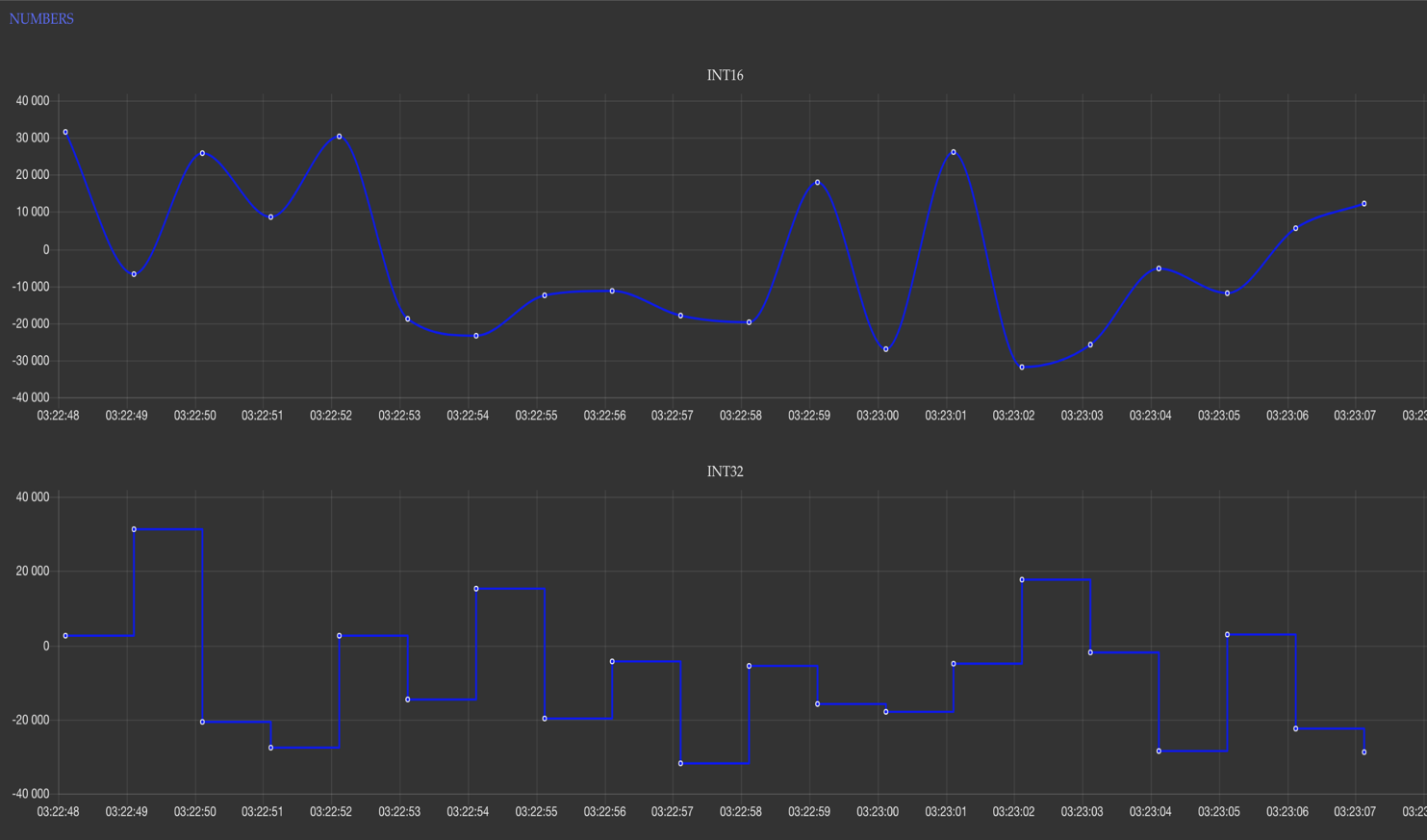


Рис. №5 Графики 16-битных и 32-битных чисел

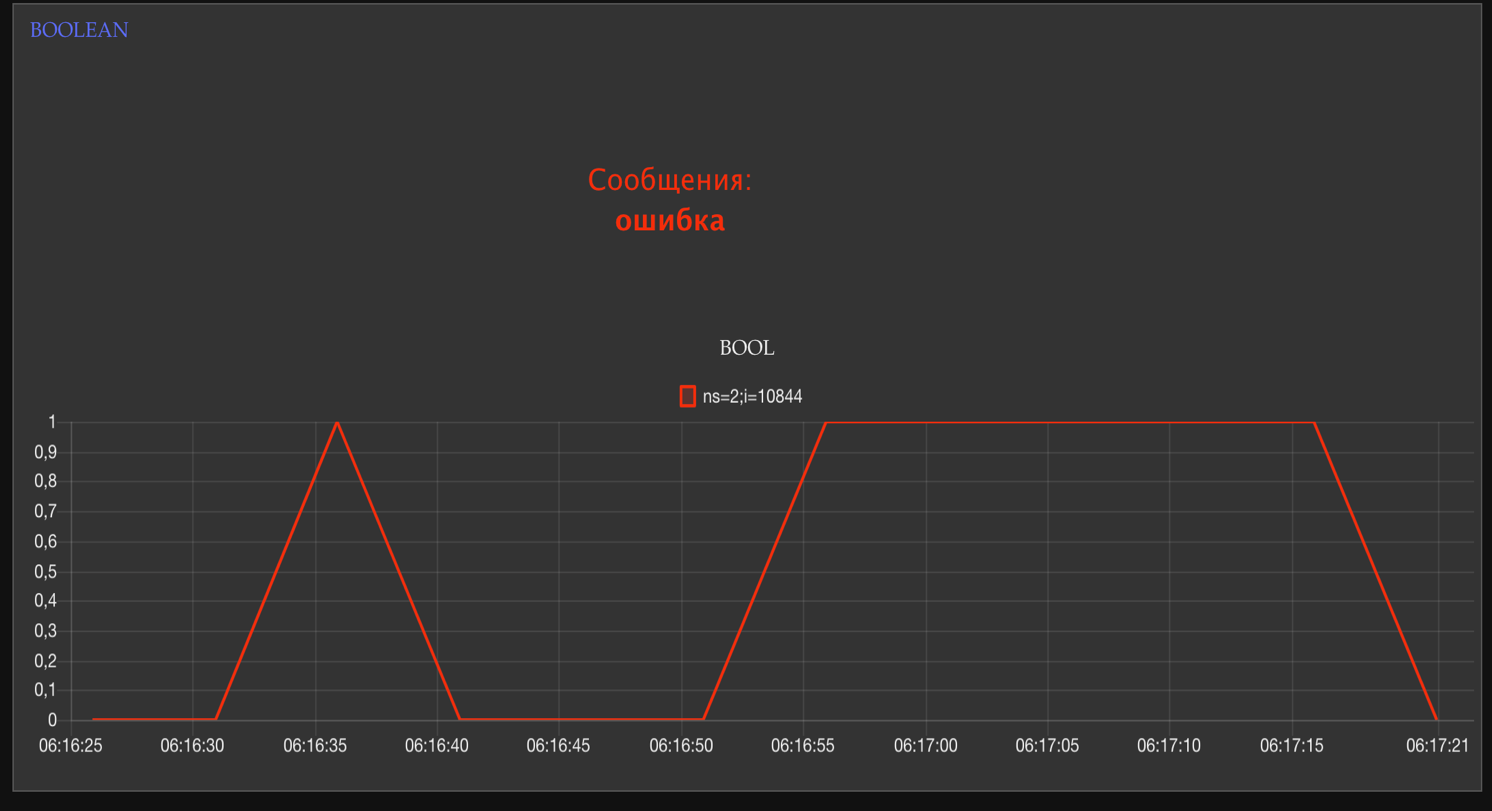


Рис. №6 График Bool-ых значений

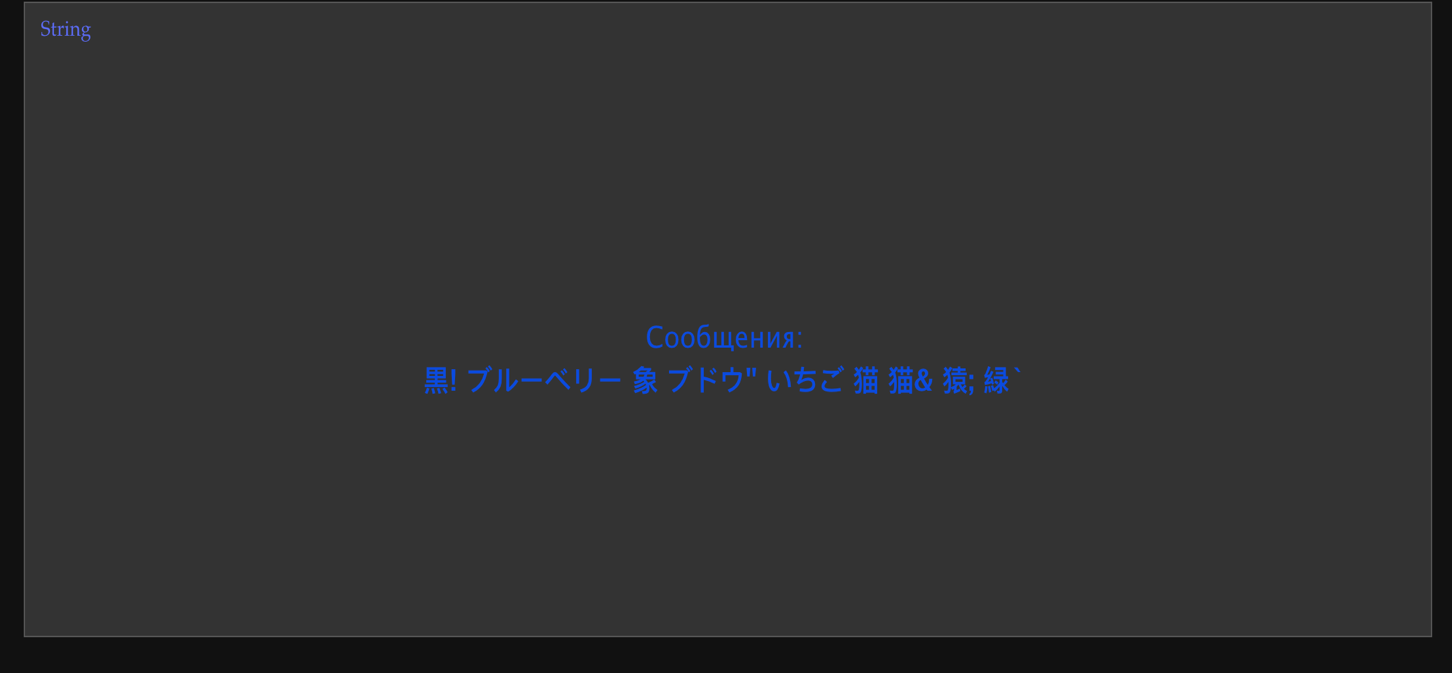


Рис. №7 График сообщений

В рамках дополнительного освоения возможностей Node-Red изучили вывод графиков в различной цветовой палитре.

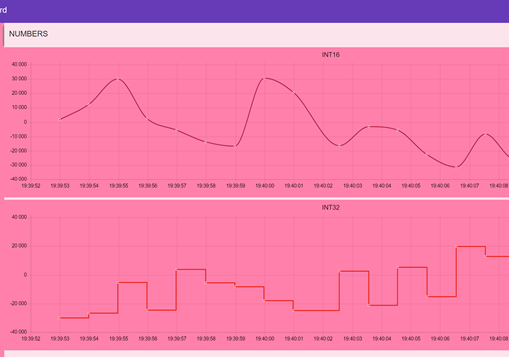


Рис. №8 Графики 16-битных и 32-битных чисел



Рис. №9 График сообщений

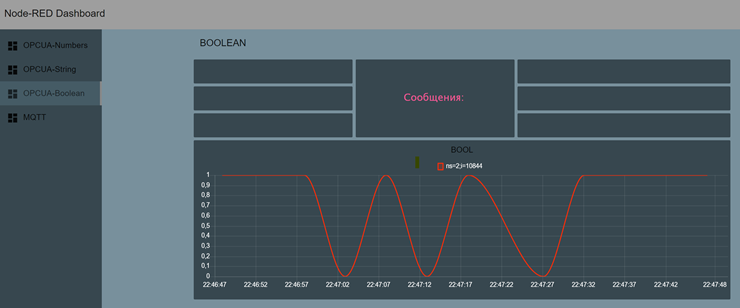


Рис. №10 График Bool-ых значений

* 1. Добавление данных с MQTT.

**MQTT (Message Queuing Telemetry Transport)** — это протокол, предназначенный для передачи сообщений между устройствами с низкой пропускной способностью и ограниченными ресурсами. Добавление данных с MQTT на web-страницу позволяет визуализировать данные в различных форматах, таких как графики, диаграммы и текст, что обеспечивает более наглядное представление информации.

1. Для добавления данных мы подключились к двум сенсорам: температуры и влажности.

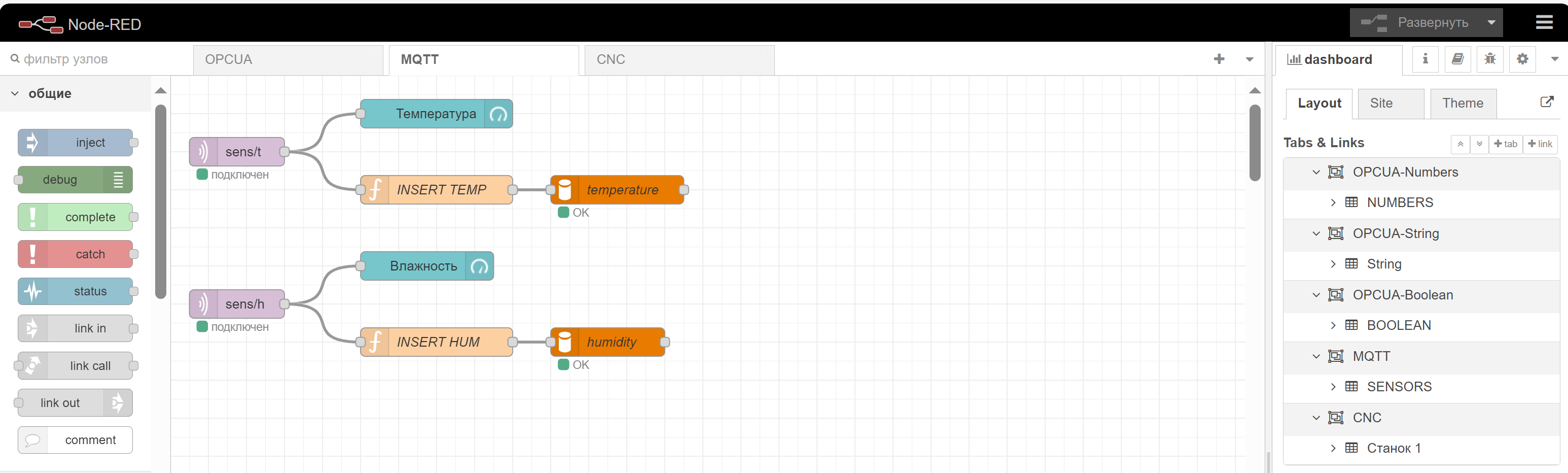


Рис. №11 Поток MQTT сервера

1. С помощью Node-Red нарисовали два графика температуры и влажности в виде диаграммы, а также настроили поток сохранения этих значений в базу данных.

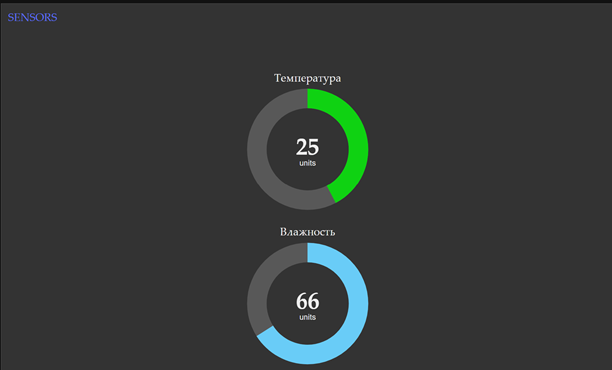
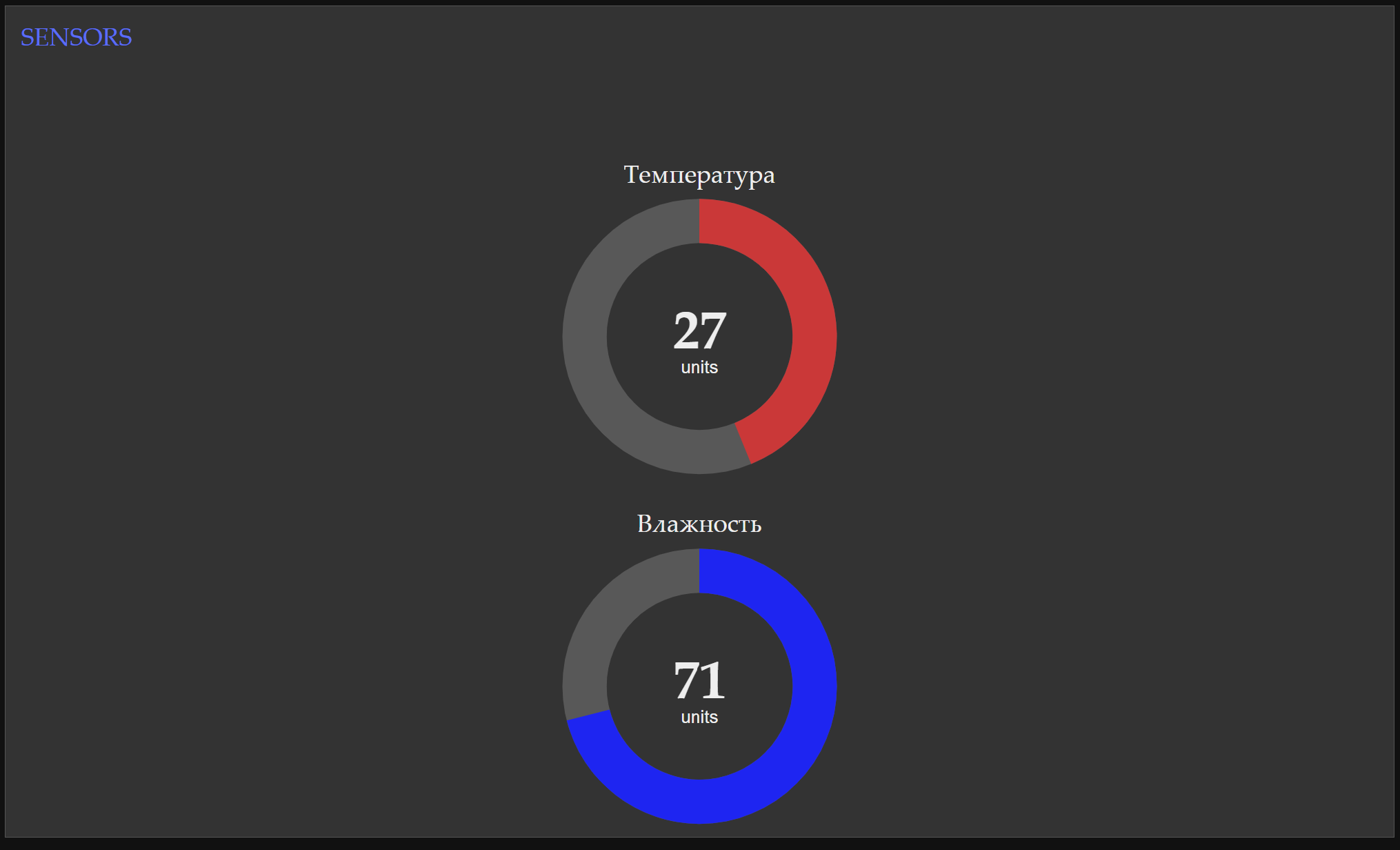
 

Рис. №12 Цветовые диаграммы изменения температуры и влажности

Цветовой индикатор помогает быстрее определить состояние датчиков.

Графическое изображение температуры и влажности настроили так, чтобы цвет диаграммы менялся в зависимости от показателей датчика.

|  |  |
| --- | --- |
| Температура:  Ниже 10° : **синий** цвет  От 10° до 25° : **зеленый** цвет  Выше 25° : **красный** цвет | Влажность:  0 – 40%: **красный** цвет  40-70%: **голубой** цвет  70-100%: **синий** цвет |

# Этап 2. Сохранение данных

# Настроили работу с MySQL. Создали таблицу в базе данных на phpMyAdmin для записи значений, которые получили, используя Node-Red (OPC UA и MQTT). Полученные данные записали в созданную таблицу. Полученную базу данных выгрузили в формате .csv для дальнейшей работы с данными в программе Phyton.

Этап 3. Аналитика данных

1. Собирали данные с протокола CNC, т.е. собрали данные с двух станков в виде json, визуализировали и так же сохранили в базу данных.

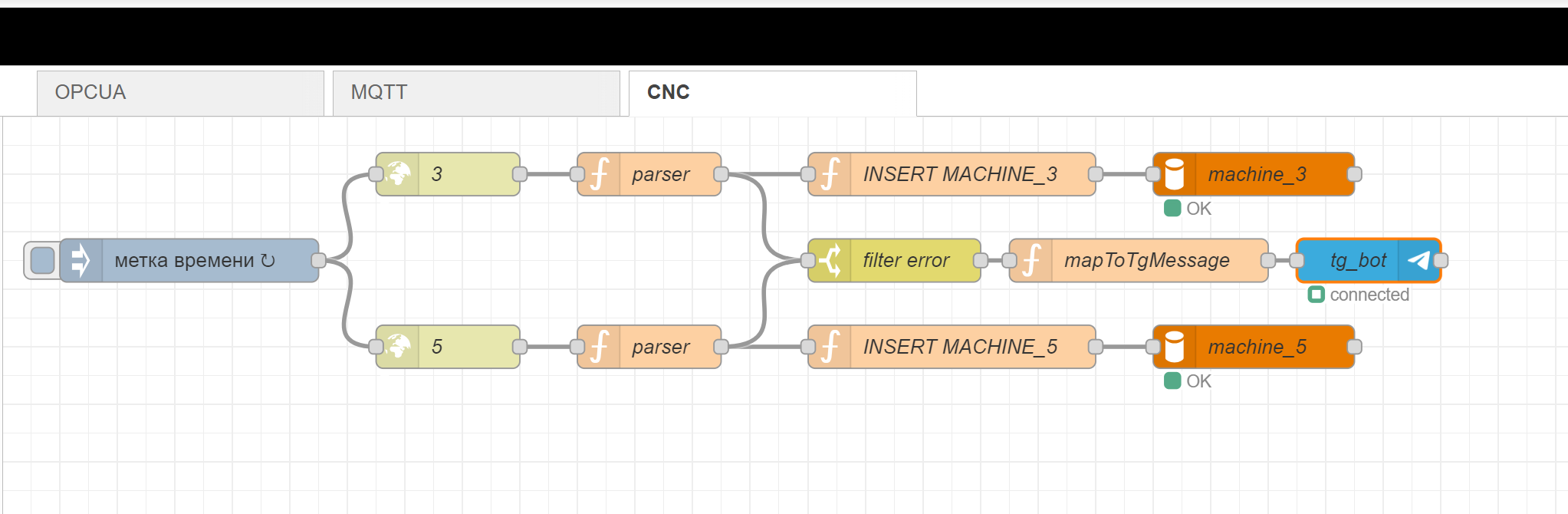


Рис. №13 Поток CNC сервера

Базу данных выгрузили в формате .csv для дальнейшей работы с данными в программе Phyton.

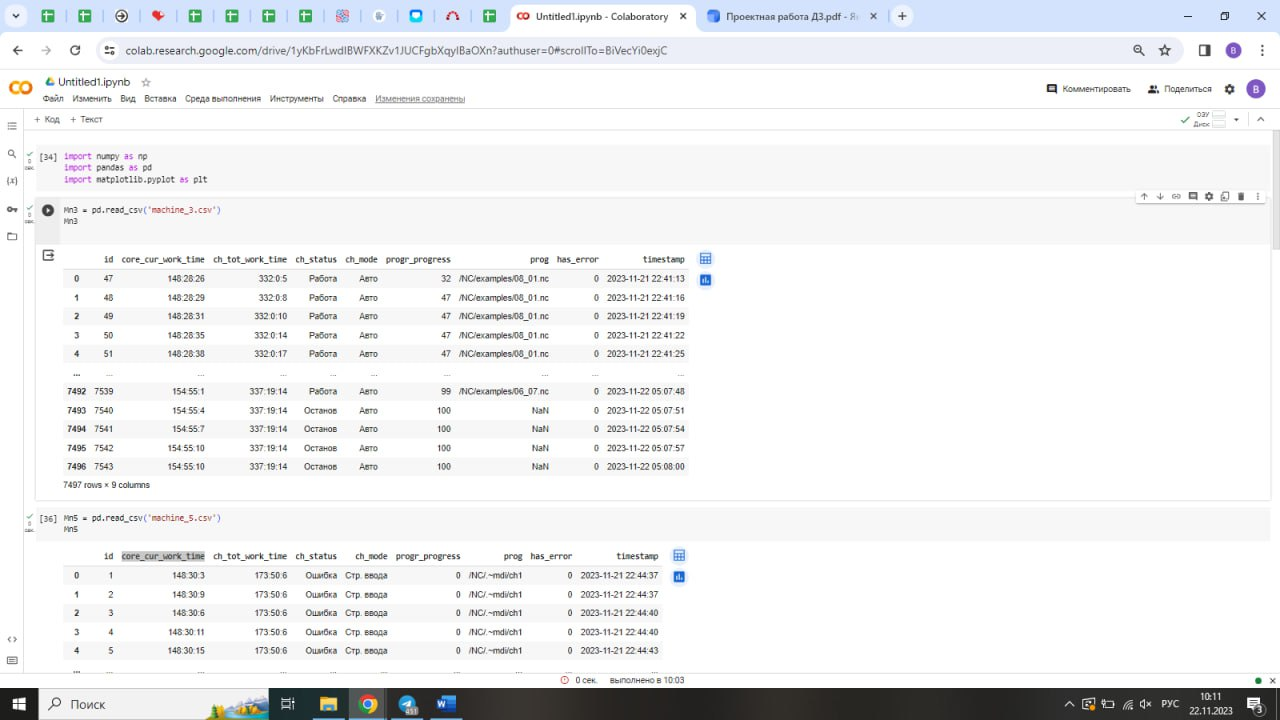


Рис. №14 Визуализация базы данных, собранных со станков №3 и №5

1. Нахождение программы, которая была запущена больше всего времени.

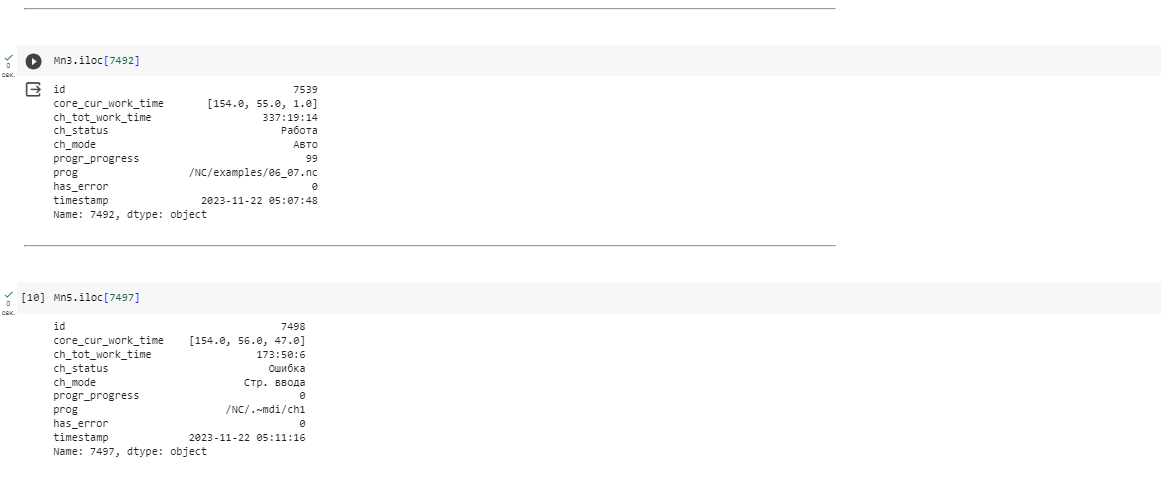


Рис. №15 Вывод «имени» программы, которая была запущена больше всего по времени

Исходя из данных, полученных от станка 3, можем сказать, что больше всего по времени была запущена программа: /NC/examples/06\_07.nc

Исходя из данных, полученных от станка 5, можем сказать, что больше всего по времени была запущена программа: /NC/.~mdi/ch1

1. Расчет, в каком режиме CNC находился больше всего времени.



Рис. №16 Вывод столбцов, используемых при подсчете уникальных комбинаций. Код для вывода данных на графике

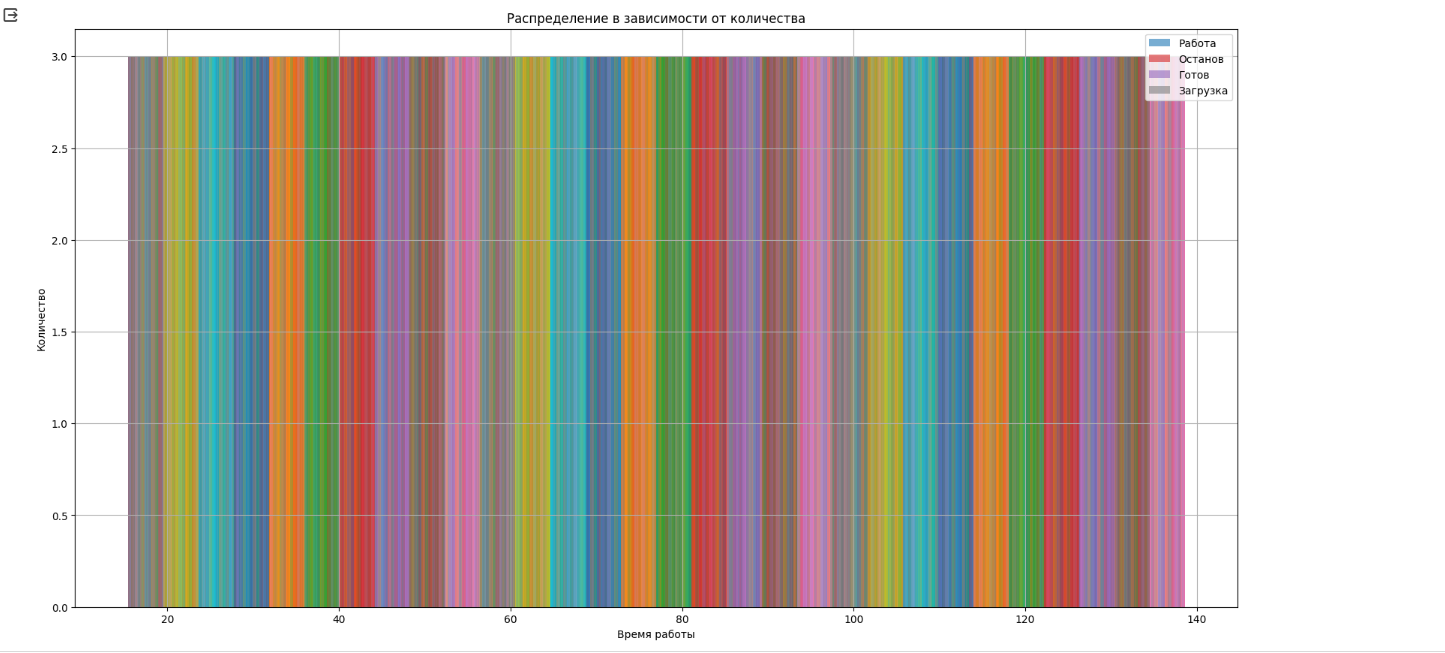


Рис. №17 Реализация кода из рис.№16

Исходя из графика, можно отметить, что больше всего CNC находился в режиме работы.

1. Расчет, как долго CNC работало (онлайн) и как долго оно было автономным.

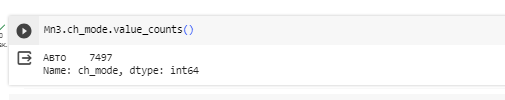


Рис. №18 Количество строк, используемых при подсчете уникальных комбинаций по запросу модуля работы.

Загрузка данных происходила онлайн, без нарушения автономности работы.

# Этап 4. Уведомление

В десктопной версии Телеграм в поисковой строке ввели @BotFather нажали кнопку запустить

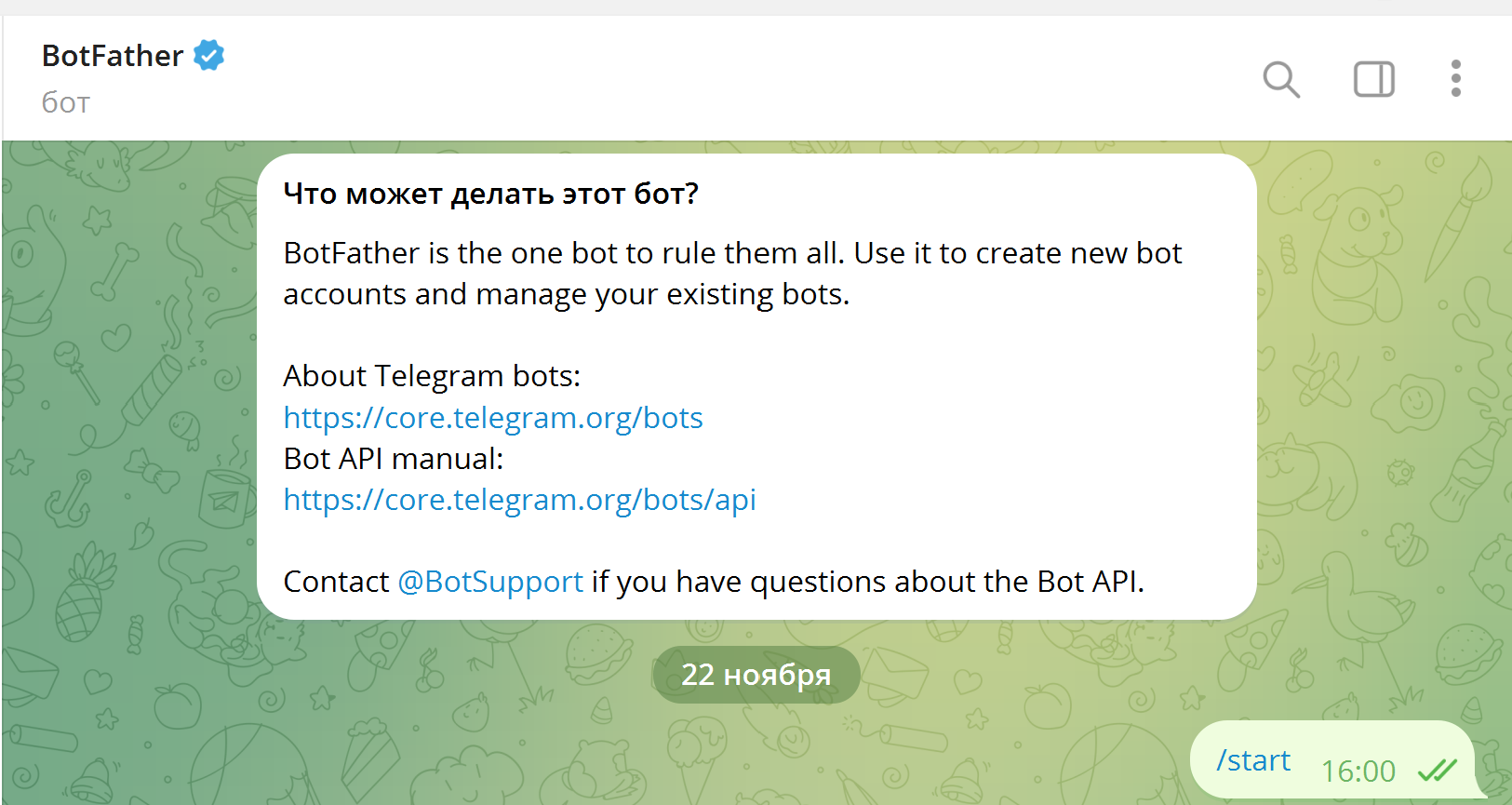


Рис. №19 Стартовая страница подключения в чат-боту @BotFather

Из предложенного списка выбрали /newbot, и ввели название нашего будущего бота inno\_prod\_data\_bot

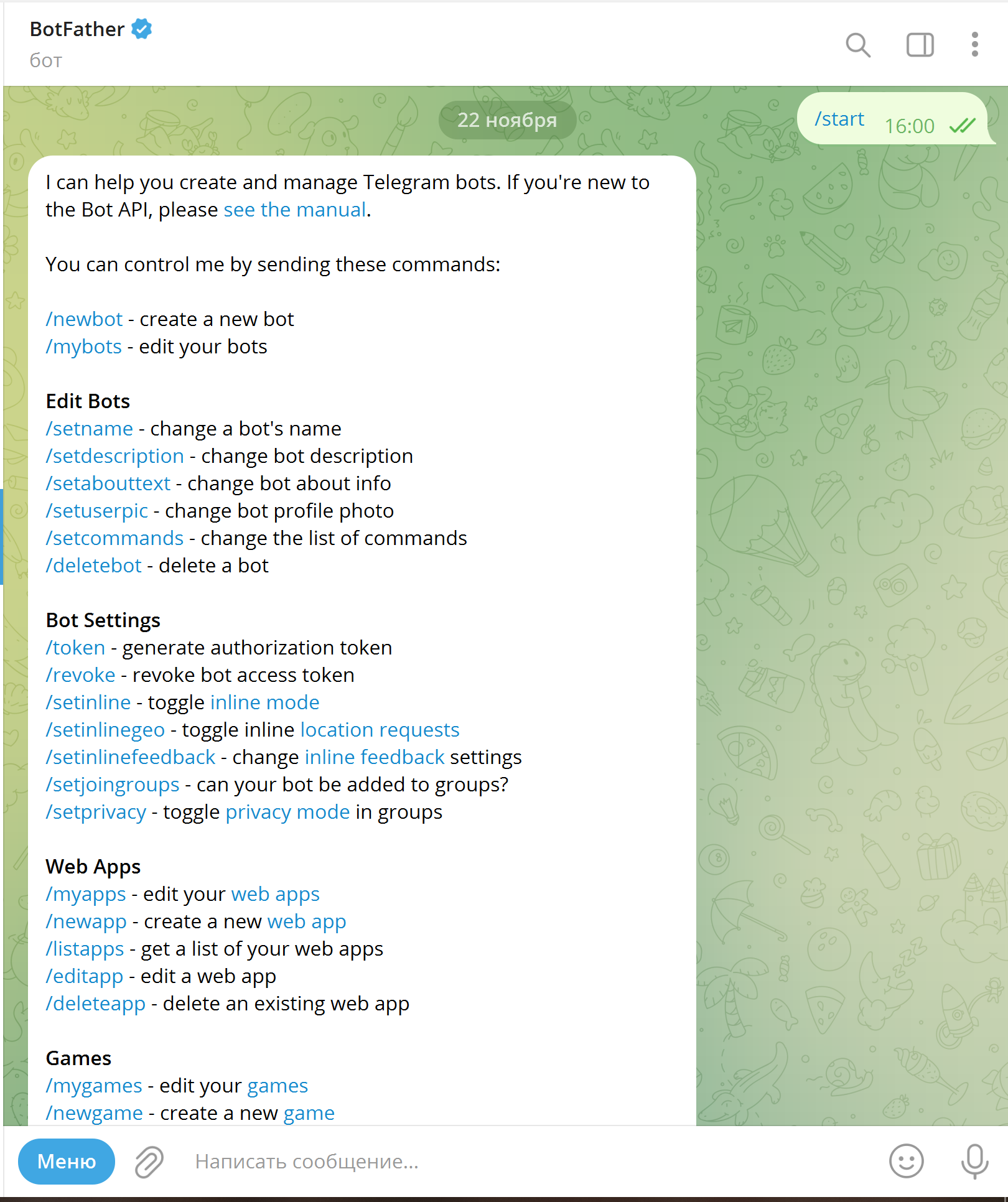
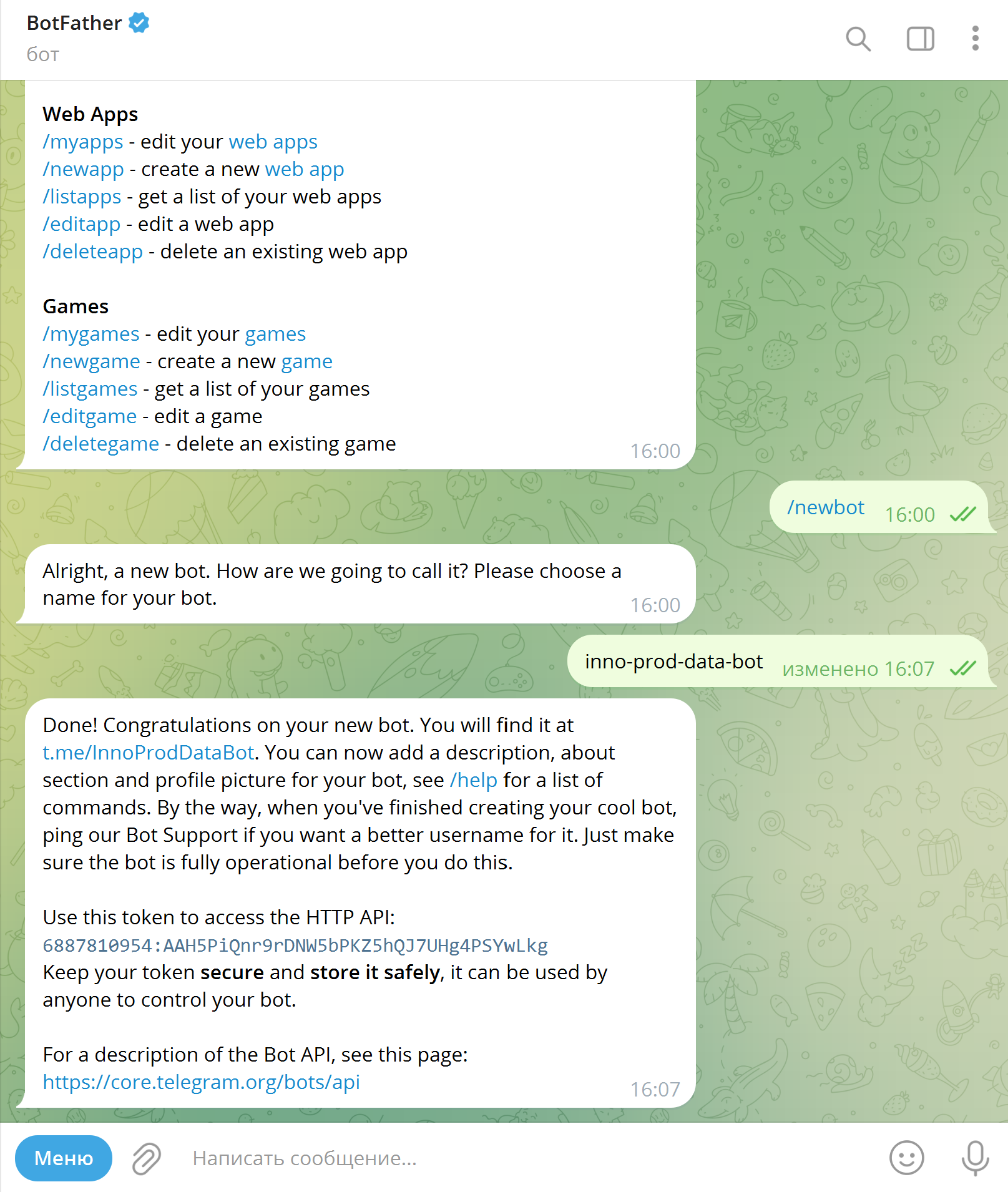
 

Рис. №20-21 Работа с чат-ботом @BotFather

Далее в Node-red. устанавливаем плагин node-red-contrib-telegrambot.

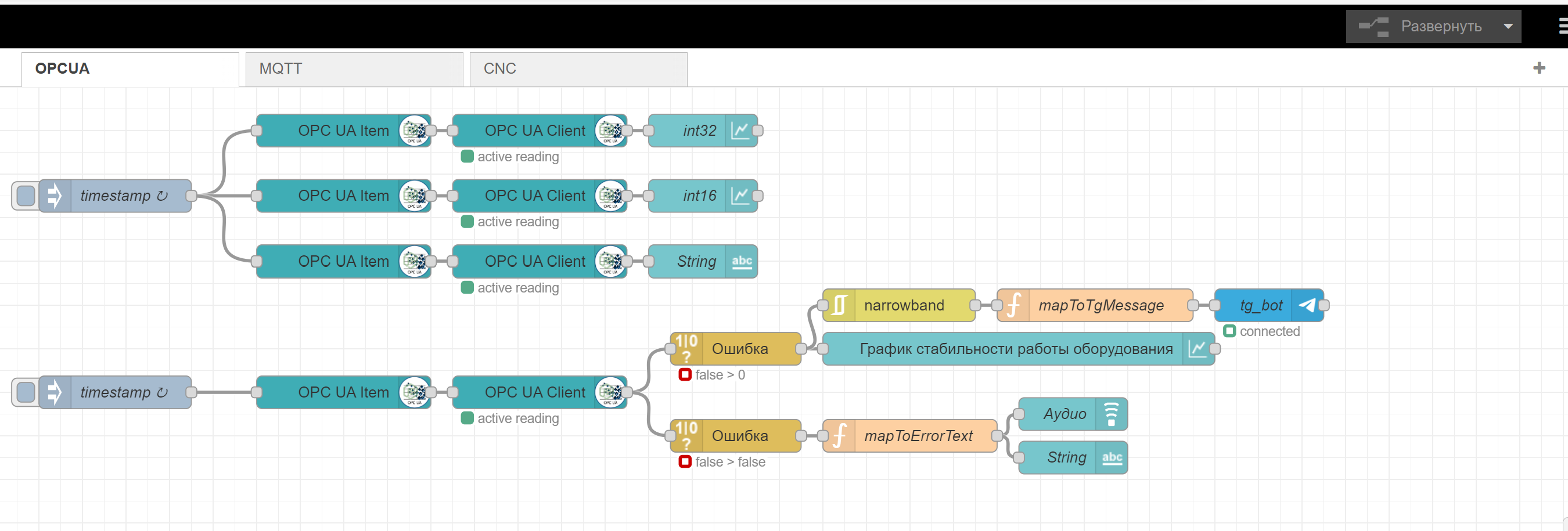


Рис. №22 Поток OPC UA сервера

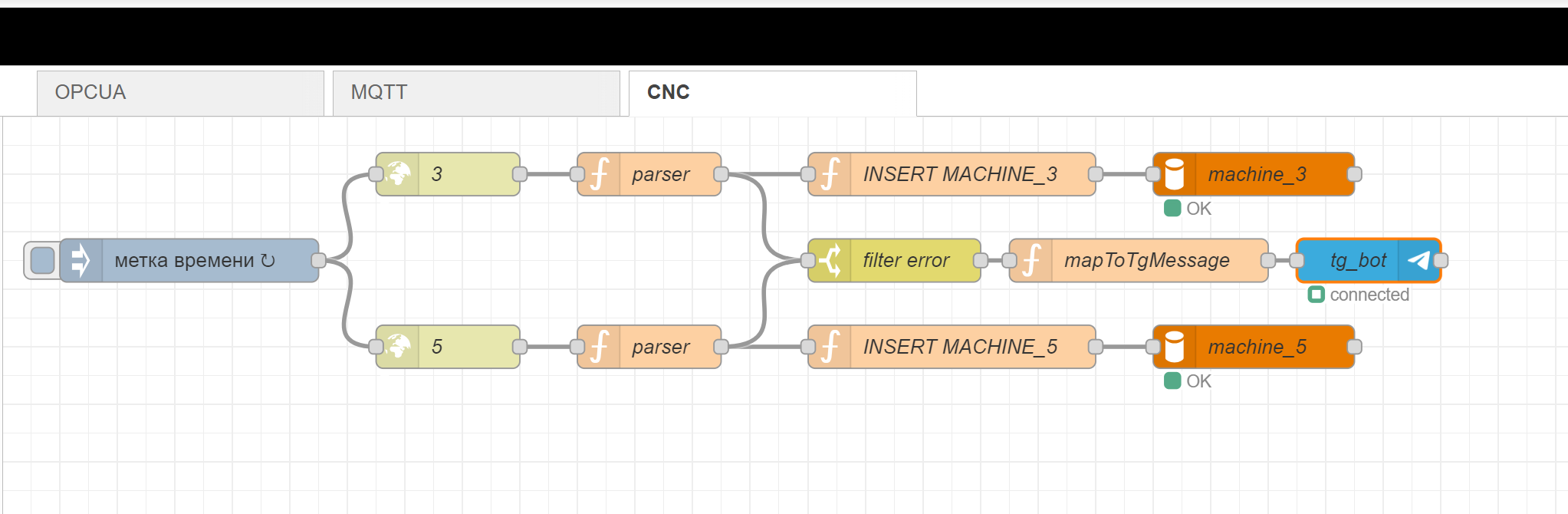


Рис. №23 Поток CNC сервера

В чат-боте Телеграм <https://t.me/myidbot> создаем id

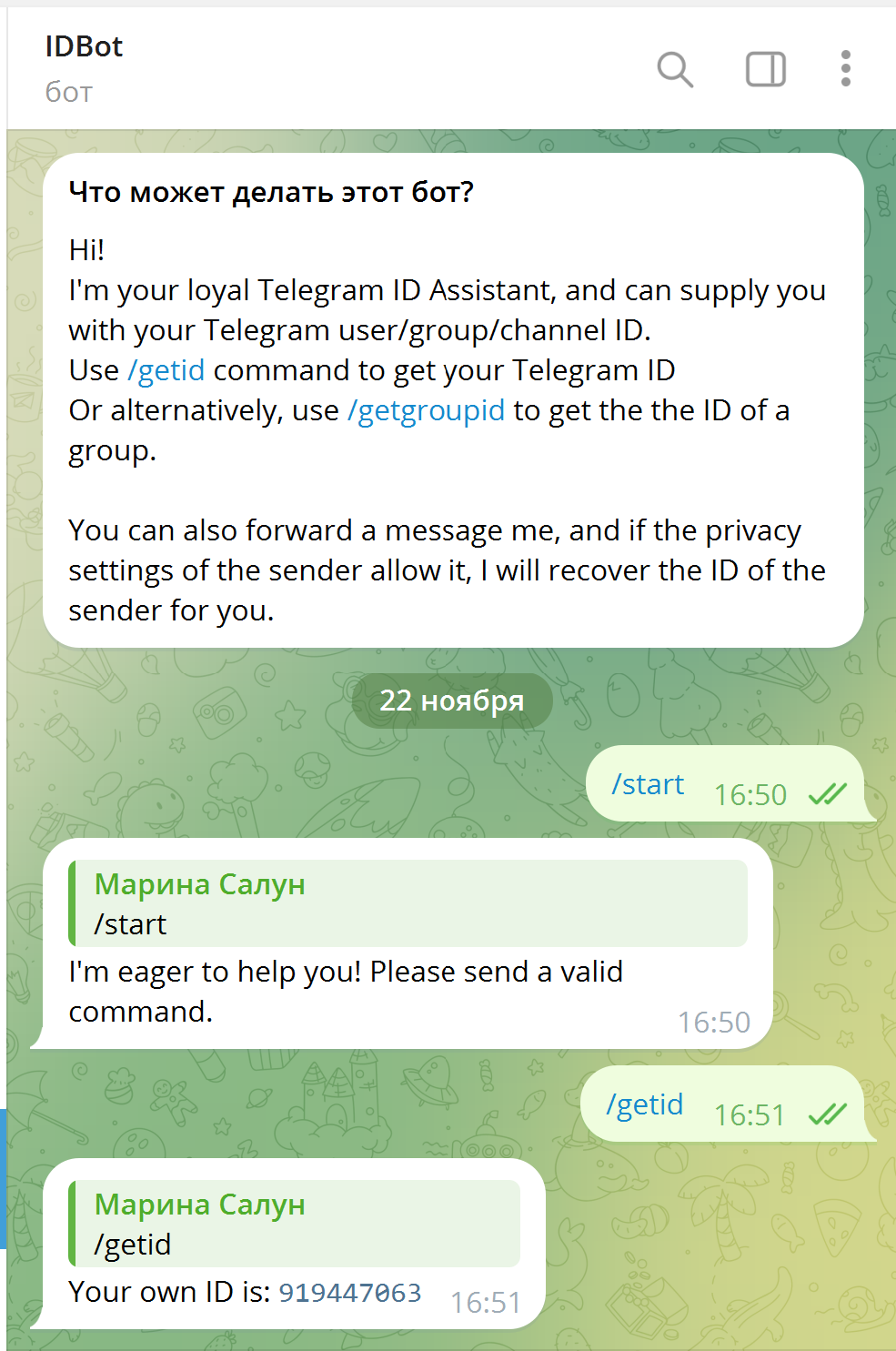


Рис. №24 Работа с чат-ботом IDBot, для получения ID

В ноде Sender заполнили имя нашего созданного бота, ввели id.

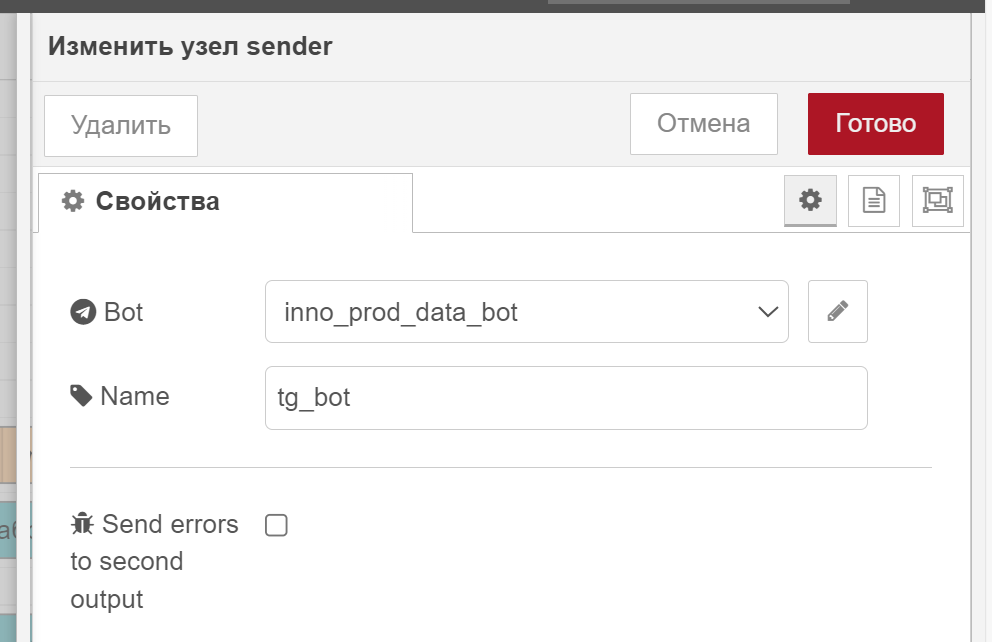


Рис. №25 Заполнение данных по нашему чат-боту, куда приходит сообщение

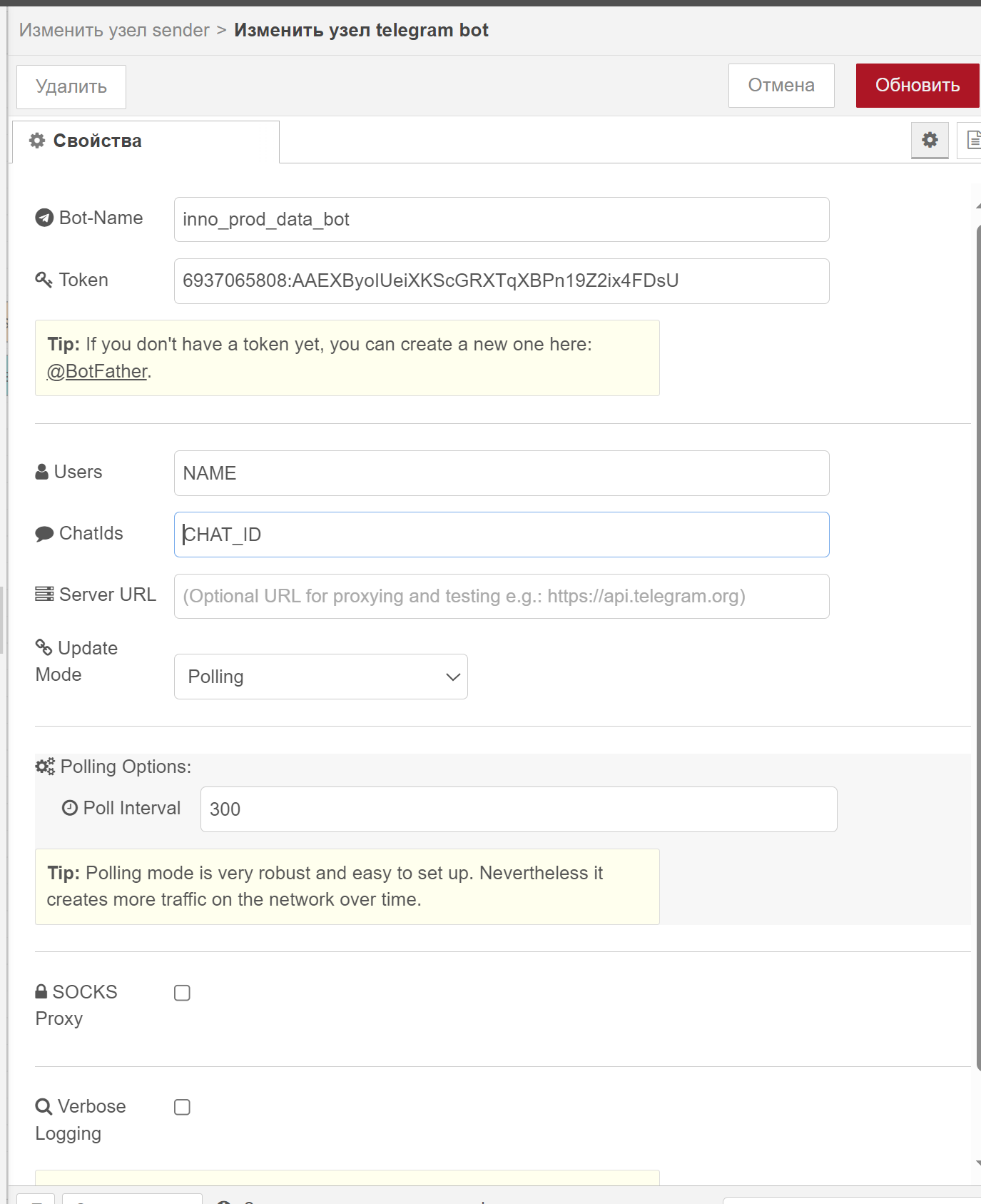


Рис. №26 Заполнение данных (имени и id) для получения смс в чат-бот

Теперь на адрес <https://t.me/inno_prod_data_bot> приходит уведомление об ошибке.

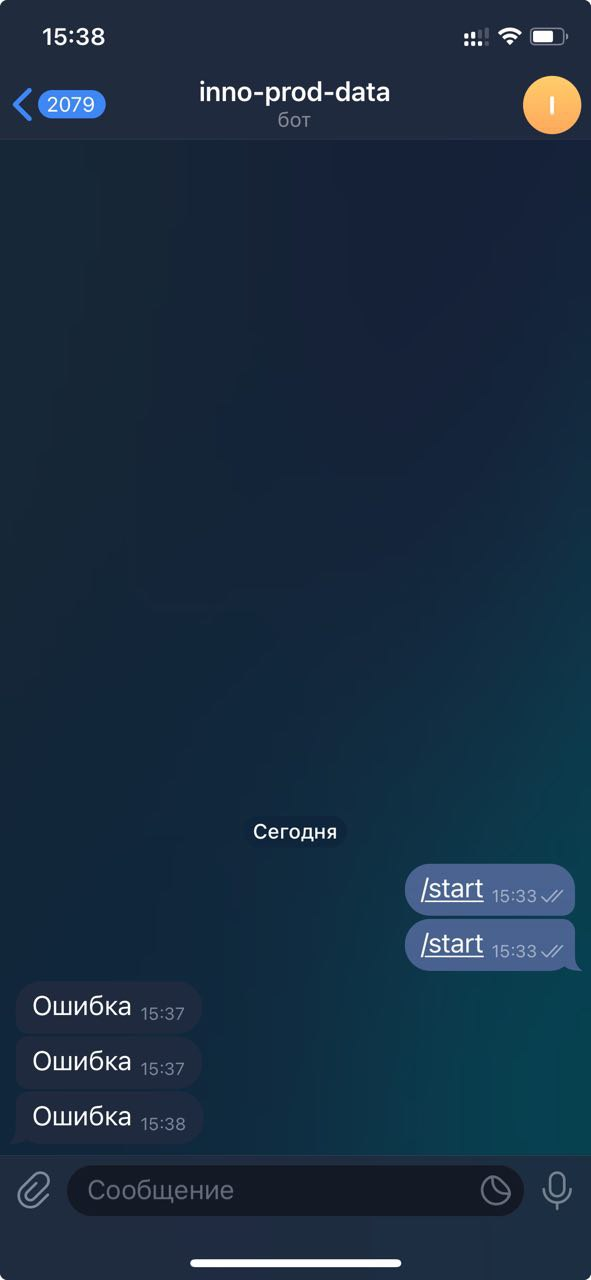
 

Рис. №27-28 Стартовая страница нашего чат-бота, пример получения сообщения об ошибки.

Ответственные за работы:

1. Васильев Дмитрий: Настройка работы Node-red, выгрузка базы данных, настройка чат-бота

2. Коваль Виктория: Анализ данных, заполнение отчета

3. Салун Марина: Визуализация данных (работа с графиками и диаграммами) в Node-red, настройка чат-бота, заполнение отчета