

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра комплексной информационной безопасности электронно-  
вычислительных систем (КИБЭВС)

ИНТЕРНЕТ ВЕЩЕЙ

Отчет о выполнении промежуточного аттестационного этапа группового  
проектного обучения (ГПО)  
Проект ГПО – КИБЭВС-1904

Ответственный исполнитель проекта:  
Студент гр. 738-1

\_\_\_\_\_ Н.Г. Останков  
(подпись)

«23» декабря 2021 г.

Проверил:

Руководитель проекта

Старший преподаватель каф. КИБЭВС

\_\_\_\_\_ О.В. Пехов  
(подпись) (оценка)

«23» декабря 2021 г.

Принял:

Ответственный за ГПО на кафедре


Доцент каф. БИС, к.т.н.

\_\_\_\_\_ И.А. Рахманенко  
(подпись) (оценка)

«23» декабря 2021 г.

Исполнители проекта ГПО КИБЭВС-1904:Студент гр. 738-1

\_\_\_\_\_ Н.Г. Останков

Студент гр. 739-1 \_\_\_\_\_ Ц.Б. ЦыриторовСтудент гр. 739-1 \_\_\_\_\_ Э.С. СемёновСтудент гр. 739-2

\_\_\_\_\_ А.А. Лобанов

**Министерство науки и высшего образования Российской Федерации**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)**

*Групповое проектное обучение*

---

УТВЕРЖДАЮ

Зав. кафедрой КИБЭВС  
Шелупанов Александр Александрович

«\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**на выполнение проекта № КИБЭВС-1904**

1. Основание для выполнения проекта: приказ № 3247ст от 26.06.2019.
2. Наименование проекта: Интернет вещей.
3. Цель проекта:
  - изучение технологий модуляции LoRa и сетевого протокола LoRaWAN;
  - создание IoT-сети, основанной на данных технологиях.
4. Основные задачи проекта на этапах реализации:
  - изучение технологий модуляции LoRa и сетевого протокола LoRaWAN;
  - изучение документации оборудования, которое будет использовано для реализации IoT-сети;
  - произвести настройку данного оборудования;
  - создание работоспособной IoT-сети.
5. Научная новизна проекта: Нет.

6. Планируемый срок реализации: Получение первых результатов ожидается к декабрю 2021г.
7. Целевая аудитория (потребители): IoT-разработчики.
8. Заинтересованные стороны: Компании, использующие IoT устройства.
9. Источники финансирования и материального обеспечения: Нет
10. Ожидаемый результат (полученный товар, услуга): Нет.
11. Руководитель проекта: Пехов О.В
12. Ответственный исполнитель проекта: Останков Н.Г.

Члены проектной группы:

Останков Никита Григорьевич 738-1;

Цыриторов Цырен Биликтуевич 739-1;

Семёнов Эдуард Саввич 739-1;

Лобанов Александр Алексеевич 739-2.

13. Место выполнения проекта: ул. Красноармейская, д. 146, 7 этаж, ауд. 707.

14. Календарный план выполнения проекта:

№ Этапа	Наименование этапа	Содержание работы	Сроки выполнения		Ожидаемые результаты этап
			Начало	Окончание	
1	Изучение технологий модуляции LoRa и сетевого протокола LoRaWAN	Сбор и систематизация информации, связанной с данной технологией	01.09.2021	10.09.2021	Систематизированная информация, связанная с данными технологиями
2	Изучение документации на Вега БС-2.2	Сбор и систематизация информации, связанной с данной базовой станцией, от технических характеристик до настройки	10.09.2021	20.09.2021	Систематизированная информация, связанная с данной базовой станцией, от технических характеристик до настройки
3	Изучение документации на STM32 NUCLEO F103RB	Сбор и систематизация информации, связанной с данным микроконтроллером, от технических характеристик до настройки	20.09.2021	30.09.2021	Систематизированная информация, связанная с данным микроконтроллером, от технических характеристик до настройки

		характеристик до настройки			
4	Изучение документации на RAK811	Сбор и систематизация информации, связанной с данным радио модулем, от технических характеристик до настройки	30.09.2021	10.10.2021	Систематизированная информация, связанная с данным радио модулем, от технических характеристик до настройки
5	Изучение документации на UMDK-THP	Сбор и систематизация информации, связанной с данным датчиком, от технических характеристик до настройки	10.10.2021	20.10.2021	Систематизированная информация, связанная с данным датчиком, от технических характеристик до настройки
6	Изучение протокола ПС	Сбор и систематизация информации, связанной с данным протоколом	20.10.2021	30.10.2021	Систематизированная информация, связанная с данным протоколом, от технических характеристик до настройки
7	Реализация IoT-сети на имеющемся оборудовании	Настройка и объединение оборудования в единую IoT-сеть	31.10.2021	15.12.2021	Работоспособная IoT-сеть
8	Отчётность	Написание отчёта по ГПО	15.12.2021	22.12.2021	Отчёт по проделанной работе

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

Руководитель проекта:

Старший преподаватель каф. КИБЭВС

(должность)

(подпись)

Пехов О.В.

(расшифровка)

Члены проектной группы:

(подпись)

Цыриторов

(подпись)

Вен

(подпись)

(подпись)

Останков Н.Г.

(расшифровка)

Цыриторов Ц.Б.

(расшифровка)

Семёнов Э.С.

(расшифровка)

Лобанов А.А.

(расшифровка)

## Реферат

Отчёт содержит 55 страниц, 37 рисунков, 1 таблица, 25 источника.

LORA, LORAWAN, LPWAN, IOT, STM32 NUCLEO, RAK, UMDK, UART, ПС, ООО «ВЕГА-АБСОЛЮТ», БАЗОВАЯ СТАНЦИЯ, МИКРОКОНТРОЛЛЕР, РАДИОМОДУЛЬ, USB АДАПТЕР, ДАТЧИКИ МИКРОКЛИМАТА.

Объекты исследования: Системы интернет вещей

Цели работы:

- изучение технологий модуляции LoRa и сетевого протокола LoRaWAN;
- изучение документации оборудования, которое будет использовано для реализации IoT-сети;
- произвести настройку данного оборудования;
- создание работоспособной IoT-сети.

Пояснительная записка к групповой проектной работе выполнена в текстовом редакторе Microsoft Word 2021.

Оформлено в соответствии с ОС ТУСУР 01 – 2013. [1]

## Содержание

1 Введение.....	8
2 Описание проекта.....	9
3 Теоретическая часть.....	10
3.1 LPWAN.....	10
3.2 LoRa.....	11
3.3 LoRaWAN .....	12
3.4 Цифровой протокол I2C .....	20
4 Практическая часть .....	22
4.1 Работа с базовой станцией Vega BC-2.2.....	22
4.2 Работа с IoT Vega Server и IoT Vega Admin Tool .....	25
4.3 Сетевое взаимодействие конечного устройства и сервера.....	34
4.4 Описание устройства для снятия микроклиматических параметров .....	36
5 Заключение .....	48
Список использованных источников .....	49
Приложение А (обязательное) Код прошивки микроконтроллера.....	53

## 1 Введение

Интернет вещей (Internet of Things, IoT) — это множество физических объектов, подключенных к интернету и обменивающихся данными. [2]

Концепция IoT может существенно улучшить многие сферы нашей жизни и помочь нам в создании более удобного, умного и безопасного мира.

Примеры Интернета вещей варьируются от носимых вещей, таких как умные часы, до умного дома, который умеет, например, контролировать и автоматически менять степень освещения и отопления. Также ярким примером служит так называемая концепция умного предприятия (Smart Factory), которое контролирует промышленное оборудование и ищет проблемные места, а затем перестраивается так, чтобы не допустить поломок.

Интернет вещей занимает важное место в процессе цифровой трансформации в компаниях. Прогнозируется, что к 2030 году количество подключенных к сети устройств вырастет примерно до 24 млрд с годовой выручкой до 1,5 трлн долларов. [3]

Целью проекта «Исследование и создание IoT-сети, основанной на технологиях модуляции LoRa и сетевого протокола LoRaWAN» является использование оборудования, способного взаимодействовать между собой при помощи данных технологий, для создания IoT-сети, чтобы в дальнейшем изучить способы атаки на данную сеть и методы противодействия им, а также произвести их реализацию.



## 2 Описание проекта

В ходе проекта «Исследование и создание IoT-сети, основанной на технологиях модуляции LoRa и сетевого протокола LoRaWAN» планируется изучить документацию оборудования, которое будет использовано для реализации IoT-сети:

- Базовая станция: Вега БС-2.2;
- Микроконтроллер: STM32 NUCLEO F103RB;
- Радио модуль: RAK811;
- USB-Адаптер: XBee USB-Adapter;
- Сенсор: Bosch BME280.

Произвести настройку данного оборудования. Создать работоспособную IoT-сеть.

Таким образом, в ходе выполнения проекта планируется представить работающую IoT-сеть, с помощью которой будет обрабатываться и передаваться микроклиматические параметры в учебных аудиториях.

### 3 Теоретическая часть

#### 3.1 LPWAN

LPWAN, Low-power Wide-area Network (энергоэффективная сеть дальнего радиуса действия) — семейство технологий беспроводной передачи небольших по объёму данных на дальние расстояния, разработанных для распределённых сетей телеметрии, межмашинного взаимодействия и Интернета вещей. [4]

Беспроводные технологии LPWAN обеспечивают среду передачи данных от различного конечного оборудования: датчиков, счётчиков и сенсоров. В основе передачи данных LPWAN систем лежит принцип увеличения энергетики, а значит и дальности связи при уменьшении скорости передачи.

Чем ниже битовая скорость передачи, тем больше энергии вкладывается в каждый бит и тем легче выделить его на фоне шумов на приёмной стороне. Таким образом, низкая скорость передачи данных позволяет добиться большей дальности распространения радиосигнала, и, как следствие, увеличения радиуса действия принимающей станции.

Полученные на сервере данные от конечных устройств используются для отображения, анализа, построения отчетов и принятия решений. Управление устройствами и обновление программного обеспечения происходит с использованием обратного канала связи.

Технологии LPWAN идеально подходят для соединения устройств, которым требуется передавать малые объёмы данных на большие расстояния, сохраняя при этом длительный срок службы батареи. Некоторые приложения Интернета вещей только передают незначительные объёмы информации. Низкое энергопотребление таких устройств позволяет решать задачи с минимальной стоимостью и отсутствием необходимости частой замены батареи.

### 3.2 LoRa

Технология модуляции LoRa (Long Range) представляет собой метод модуляции, который обеспечивает значительно бóльшую дальность связи (зону покрытия), чем другие конкурирующие с ним способы. Метод основывается на технологии модуляции с расширенным спектром и вариации линейной частотной модуляции (Chirp Spread Spectrum, CSS) с интегрированной прямой коррекцией ошибок (Forward Error Correction, FEC). [5]

Технология LoRa значительно повышает чувствительность приемника и, аналогично другим методам модуляции с расширенным спектром, использует всю ширину полосы пропускания канала для передачи сигнала, что делает его устойчивым к канальным шумам и нечувствительным к смещениям, вызванным неточностями в настройке частот при использовании недорогих опорных кварцевых резонаторов.

Технология LoRa позволяет осуществлять демодуляцию сигналов с уровнями на 19,5 дБ ниже уровня шумов, притом, что для правильной демодуляции большинству систем с частотной манипуляцией (Frequency Shift Keying, FSK) нужна мощность сигнала как минимум на 8-10 дБ выше уровня шума.

Модуляция LoRa определяет тот физический уровень (Physical Layer, PHY, иногда его называют слой), который может быть использован с различными протоколами и в различных вариантах сетевой архитектуры, таких как сетка (Mesh), звезда (Star), точка-к-точке (point-to-point) и т. п.

### 3.3 LoRaWAN

Общее описание LoRaWAN.

Типовая беспроводная сеть LoRaWAN представляет собой совокупность шлюзов (gateways), рассылающих сообщения между оконечными устройствами (end-devices) и центральным сервером (Network Server, NS), и характеризуется «звездной» топологией «star-of-stars». [6] (рис. 1)

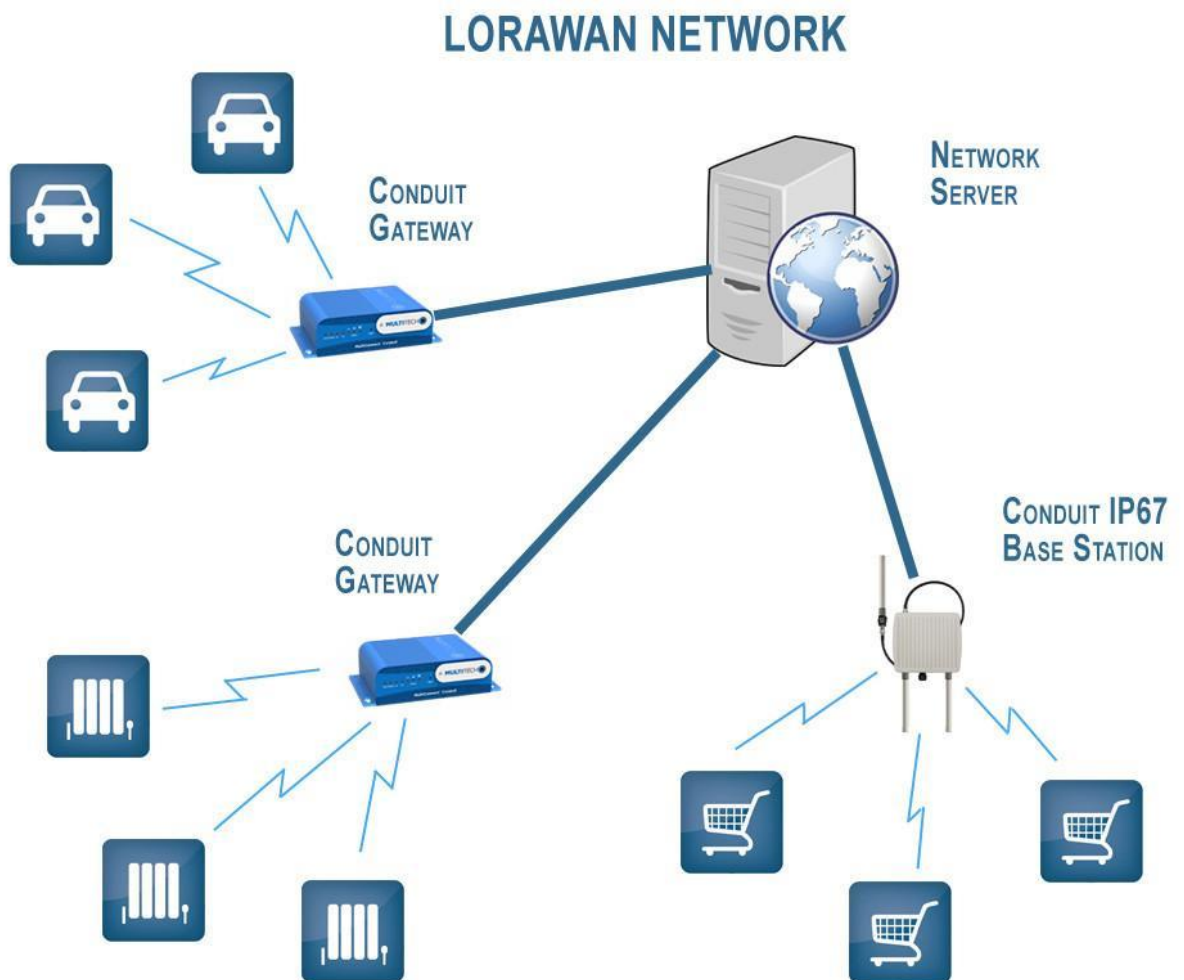


Рисунок 1 – Структура сети LoRaWAN

Связь между шлюзами и центральным сервером осуществляется через стандартные IP-соединения, а между шлюзами и оконечными устройствами — через беспроводные соединения, использующие широкополосную модуляцию LoRa. Модуляция LoRa была разработана компанией Semtech и предназначена

для низкоскоростной беспроводной передачи данных на расстояния до нескольких километров в безлицензионных диапазонах частот (Европа — 433 и 868 МГц).

Связь между шлюзами и оконечными устройствами является двусторонней, но предполагается, что основной объем данных передается от оконечных устройств к шлюзам. Технология LoRa обеспечивает скорость передачи в беспроводном канале от 0.3 до 50 кбит/с. Для разделения каналов используется как набор частотных каналов, так и скоростей передачи (data rates).

Основные преимущества беспроводных сетей LoRaWAN обусловлены использованием широкополосной модуляции LoRa и безлицензионных диапазонов частот. Сети LoRaWAN:

- совместимы с существующими сетями/технологиями беспроводной передачи данных;
- обладают высокой помехоустойчивостью;
- способны обслуживать десятки и сотни тысяч устройств;
- обеспечивают большую зону охвата и малое энергопотребление оконечных устройств.

Классы оконечных устройств LoRaWAN.

Спецификация определяет 3 класса оконечных устройств LoRaWAN: А, В и С, отличающиеся друг от друга режимами приема. Устройства данных классов являются двунаправленными. Класс А является базовым и должен поддерживаться всеми устройствами.

Класс А.

Устройства класса А после каждой передачи открывают два коротких временных окна на прием (обозначаются как RX1 и RX2).

Интервалы от конца передачи до открытия первого и второго временных окон могут конфигурироваться, но должны быть одинаковыми для всех устройств в данной сети (RECEIVE\_DELAY1, RECEIVE\_DELAY2). Для

европейского диапазона 868 МГц рекомендованное значение RECEIVE\_DELAY1 составляет 1 секунду. Значение RECEIVE\_DELAY2 должно равняться  $(\text{RECEIVE\_DELAY1} + 1)$  секунда. (рис. 2)

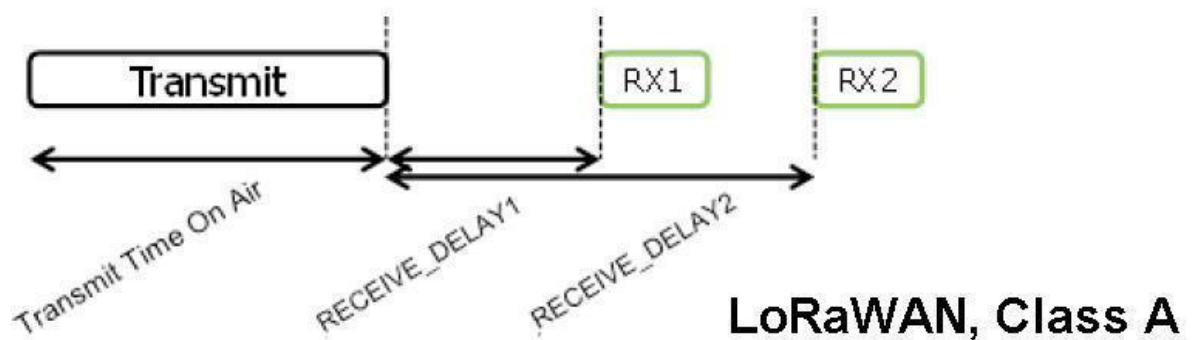


Рисунок 2 – Устройства класса А

Устройства класса А являются самыми низко потребляющими, но для передачи сообщения от сервера к конечному устройству необходимо дождаться следующего исходящего сообщения от этого устройства.

#### Класс В.

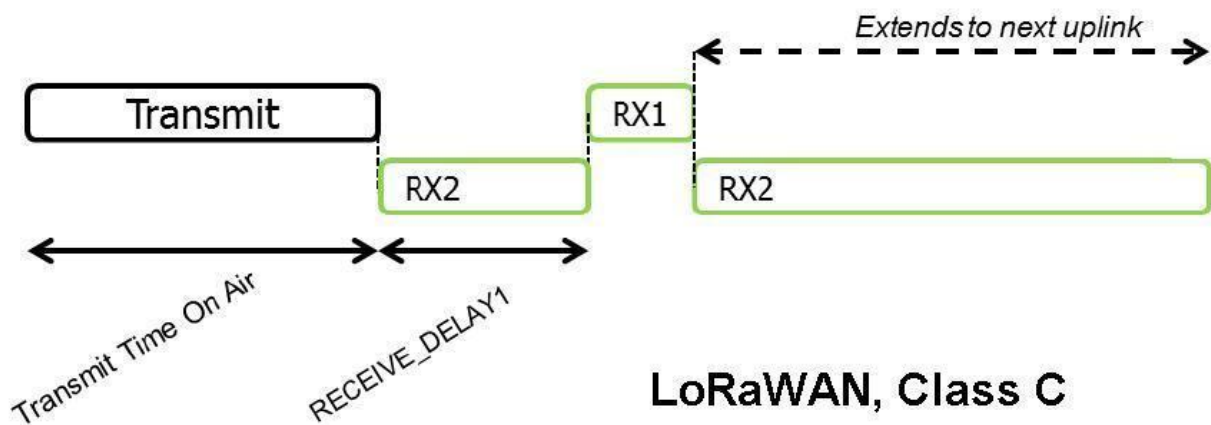
Вдобавок к окнам приема, определенным для устройств класса А, устройства класса В открывают дополнительные окна приема по расписанию. Для синхронизации времени открытия дополнительных окон приема шлюзы излучают маячки (beacons). Все шлюзы, входящие в состав одной сети, должны излучать маячки одновременно. Маячок содержит идентификатор сети и метку времени (UTC).

Использование класса В гарантирует, что при опросе конечных устройств задержка отклика не будет превышать определенную величину, определяемую периодом маячков.

#### Класс С.

Устройства класса С находятся в режиме приема практически всё время за исключением промежутков, когда они передают сообщения. За исключением

временного окна RX1 оконечное устройство использует параметры приема RX2. (рис. 3)



### LoRaWAN, Class C

Рисунок 3 – Устройства класса C

Класс C может применяться там, где не нужно изо всех сил экономить энергию (счетчики электрической энергии) или где необходимо опрашивать оконечные устройства в произвольные моменты времени.

Активация оконечных устройств.

Чтобы присоединиться к сети LoRaWAN, оконечное устройство должно пройти процедуру активации (End-Device Activation). Спецификация предусматривает два варианта активации устройств [7]:

- OTAA, Over-The-Air Activation (требуется пройти процедуру присоединения (join procedure), во время которой вырабатываются сессионные ключи шифрования и адрес DevAddr).
- ABP, Activation by Personalization (не требуется проходить процедуру присоединения, ключи шифрования и адрес DevAddr записываются в устройство заранее (персонализация устройства))

После активации устройство должно содержать следующие значения:

- End-device address (DevAddr) — локальный адрес устройства в данной сети [32 бита]. DevAddr состоит из двух полей: NwkID (идентификатор сети, биты 31...25) и NwkAddr (сетевой адрес, биты 24...0).

- Network session key (NwkSKey) — сетевой сессионный ключ [128 бит], используемый для расчета и проверки поля MIC (message integrity code) сообщений при обмене между конечным устройством и сетевым сервером (Network Server), а также шифрования сообщений MAC-уровня.
- Application session key (AppSKey) — сессионный ключ [128 бит], используемый для шифрования данных на уровне приложения (между конечным устройством и сервером приложения).
- Также устройство должно иметь идентификатор приложения [64 бита], который записывается заранее — Application identifier (AppEUI).

Активация «по воздуху» — Over-The-Air Activation.

При активации ОТАА конечное устройство должно проходить процедуру присоединения к сети каждый раз, когда сессионная информация (локальный адрес DevAddr, ключи NwkSKey, AppSKey) в устройстве отсутствует или неактуальна. Перед процедурой присоединения в устройство должны быть записаны:

- End-device identifier (DevEUI) — уникальный идентификатор, который присваивается устройству в процессе производства [64 бита].
- Application identifier (AppEUI).
- Application key (AppKey) — ключ [128 бит], который используется в процессе присоединения к сети для получения сессионных ключей NwkSKey и AppSKey.

Процедура присоединения всегда инициируется конечным устройством и состоит из двух сообщений, которыми обмениваются конечное устройство и сетевой сервер: join request (устройство → сервер) и join accept (сервер → устройство). (рис. 4)



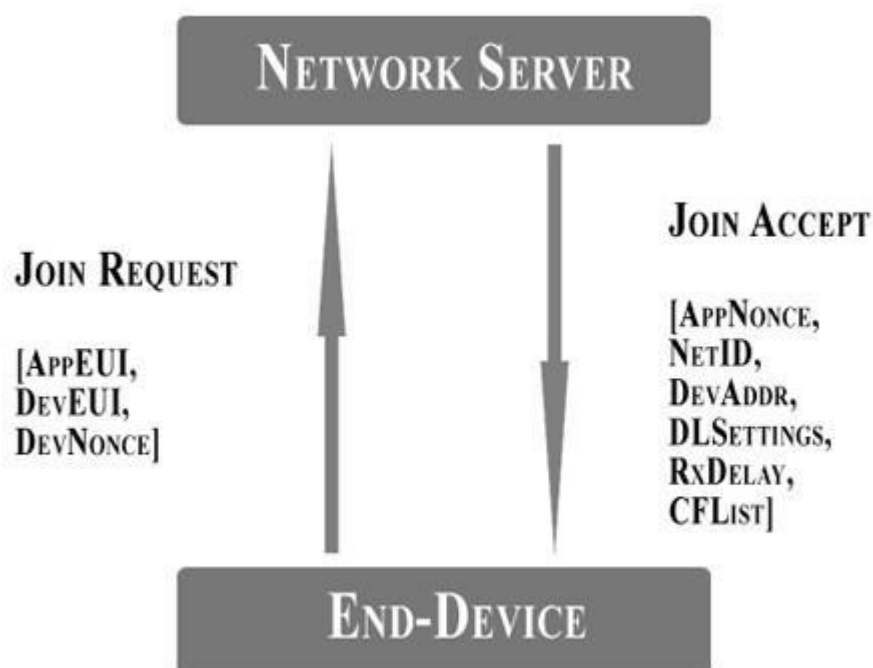


Рисунок 4 – Процедура присоединения при активации OTAA

AppNonce — случайное число, генерируемое сетевым сервером, [3 байта].

DevNonce — случайное число, генерируемое конечным устройством [2 байта].

Случайные числа DevNonce и AppNonce используются при вычислении сессионных ключей NwkSKey и AppSKey.

NetID — идентификатор сети [3 байта], 5 младших бит это NwkID.

DevAddr — адрес устройства в сети [4 байта] младшие 25 бит это NwkAddr.

DLSettings — параметры, используемые для нисходящих сообщений (сервер -> устройство) [1 байт]. В эти параметры входят: RFU — Reserved for Future Use на данный момент не используется [7-й бит], RX1DRoffset — разница между скоростями (data rates) на восходящей и нисходящей линиях, используемая для первого окна приема Rx1, по умолчанию равна 0 [биты с 6 по 4], RX2 Data rate — скорость, используемая для второго окна приема Rx2 [биты с 3 по 0].

RxDelay — задержка между окончанием передачи оконечного устройства и началом окна приема Rx1[1 байт].

CFList — список дополнительных частотных каналов, которые могут использоваться оконечным устройством [16 байт].

Для каждого оконечного устройства сетевой сервер запоминает значения DevNonce, которые использовались раньше, и игнорирует запросы, содержащие повторяющиеся значения DevNonce. В ответ на join request сервер посылает сообщение join accept.

Активация путем персонализации — Activation By Personalization.

«Активация путем персонализации» означает, что в устройство напрямую записываются значения DevAddr, NwkSKey и AppSKey (происходит персонализация устройства). Согласно спецификации, каждое устройство должно содержать уникальные значения сессионных ключей NwkSKey и AppSKey, чтобы компрометация этих значений, содержащихся в одном устройстве, не приводила к компрометации других устройств сети.

При таком способе активации оконечному устройству не нужно проходить процедуру присоединения к сети, сразу после включения устройство готово к передаче данных.

Spreading factor.

Коэффициент распределения контролирует частоту ЛЧМ-сигнала (англ. chirp) и, таким образом, контролирует скорость передачи данных. Модуляция LoRa имеет в общей сложности 6 коэффициентов распределения от SF7 до SF12. Они влияют на скорость передачи данных, время в эфире, срок службы батареи и чувствительность приемника. [8]

Более низкие коэффициенты распределения означают более быстрые ЛЧМ-сигналы и, следовательно, более высокую скорость передачи данных. При каждом увеличении коэффициента распределения частота ЛЧМ-сигнала уменьшается вдвое, и поэтому скорость передачи данных уменьшается вдвое.

Более низкие коэффициенты расширения сокращают диапазон передач LoRa, поскольку они уменьшают прирост при обработке и увеличивают скорость передачи данных. Изменение коэффициента распределения позволяет сети увеличивать или уменьшать скорость передачи данных для каждого конечного устройства за счет дальности (рис. 5).

Data Rate	Configuration (SF+BW)	Maximum application payload size (bytes)	Bit rate (bit/s)
0	LoRa: SF12 / 125 kHz	51	250
1	LoRa: SF11 / 125 kHz	51	440
2	LoRa: SF10 / 125 kHz	51	980
3	LoRa: SF9 / 125 kHz	115	1760
4	LoRa: SF8 / 125 kHz	242	3125
5	LoRa: SF7 / 125 kHz	242	5470
6	LoRa: SF7 / 250 kHz	242	11000

Рисунок 5 – Коэффициенты распределения

### 3.4 Цифровой протокол I2C

ПС — последовательная асимметричная шина для связи между интегральными схемами внутри электронных приборов. Использует две двунаправленные линии связи (SDA и SCL), применяется для соединения низкоскоростных периферийных компонентов с процессорами и микроконтроллерами.

I2C(I2C) — это достаточно широко распространённый сетевой последовательный интерфейс, придуманный фирмой Philips и завоевавший популярность относительно высокой скоростью передачи данных (обычно до 100 кбит/с, в современных микросхемах до 400 кбит/с), дешевизной и простотой реализации. Физически сеть представляет собой двухпроводную шину, линии которой называются DATA (SDA) и CLOCK (SCL) (необходим ещё и третий провод — земля, но интерфейс принято называть двухпроводным по количеству сигнальных проводов). Соответственно, по линии DATA передаются данные, линия CLOCK служит для тактирования (рис. 6). К шине может быть подключено до 128 абонентов, каждый со своим уникальным номером. В каждый момент времени информация передаётся только одним абонентом и только в одну сторону.

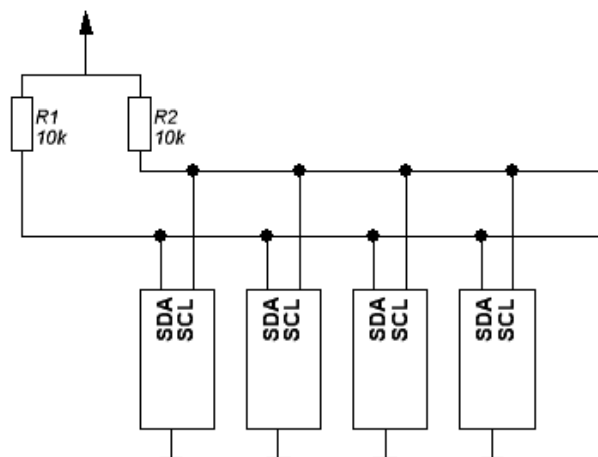


Рисунок 6 – Схема подключения устройств по протоколу I2C

Принцип подключения, следующий: есть ведущий (master) и ведомые (slave). Инициатором обмена всегда выступает ведущий, обмен между двумя ведомыми невозможен. Всего на одной двухпроводной шине может быть до 128 устройств.

Такты на линии SCL генерирует master. Линией SDA могут управлять как мастер, так и ведомый в зависимости от направления передачи. Единицей обмена информации является пакет, обрамленный уникальными условиями на шине, именуемыми стартовым и стоповым условиями. Мастер в начале каждого пакета передает один байт, где указывает адрес ведомого и направление передачи последующих данных. Данные передаются 8-битными словами. После каждого слова передается один бит подтверждения приема приемной стороной. На рисунке 7 представлена диаграмма по передаче одного слова с подтверждением (такт A). [9]

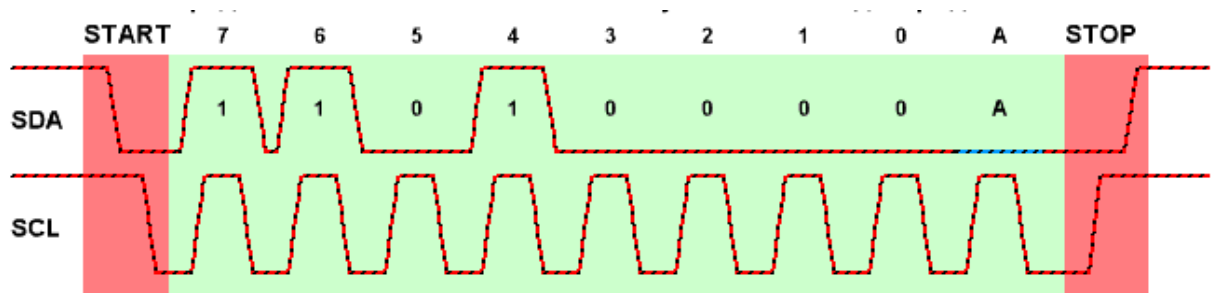


Рисунок 7 - Передача байта с битом подтверждения

#### 4 Практическая часть

##### 4.1 Работа с базовой станцией Вега БС-2.2

Базовая станция Вега БС-2.2, далее просто БС, от компании «ВЕГА АБСОЛЮТ» предназначена для разворачивания сети LoRaWAN на частотах диапазона 863-870 МГц (рис. 8). Полная информация о ней находится на сайте производителя. [10]



Рисунок 8 – Базовая станция Вега БС-2.2

Перед началом работы с данной БС были изучены её особенности и характеристики, а также руководства по эксплуатации и разворачиванию сети LoRaWAN. Полная информация о них находится на сайте производителя. [11]

Первым же делом была произведена инициализация БС на персональном компьютере, далее просто ПК, по протоколу SSH, используя терминальную программу PuTTY, вследствие чего было диагностировано правильное подключение станции к компьютеру, а также получена необходимая информация - IP-адрес БС, для её настройки через веб-интерфейс. (рис. 9-10).

```

root@am335x-evm:~/bs-dashboard/manager# ifconfig
eth0      Link encap:Ethernet  HWaddr 34:03:DE:7B:72:80
          inet addr:192.168.1.228  Bcast:192.168.1.255  Mask:255.255.254.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1478151 errors:0 dropped:614 overruns:0 frame:0
          TX packets:103187 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:109611064 (104.5 MiB)  TX bytes:23971656 (22.8 MiB)
          Interrupt:56

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@am335x-evm:~/bs-dashboard/manager# █

```

Рисунок 9 – Инициализация БС на ПК по протоколу SSH

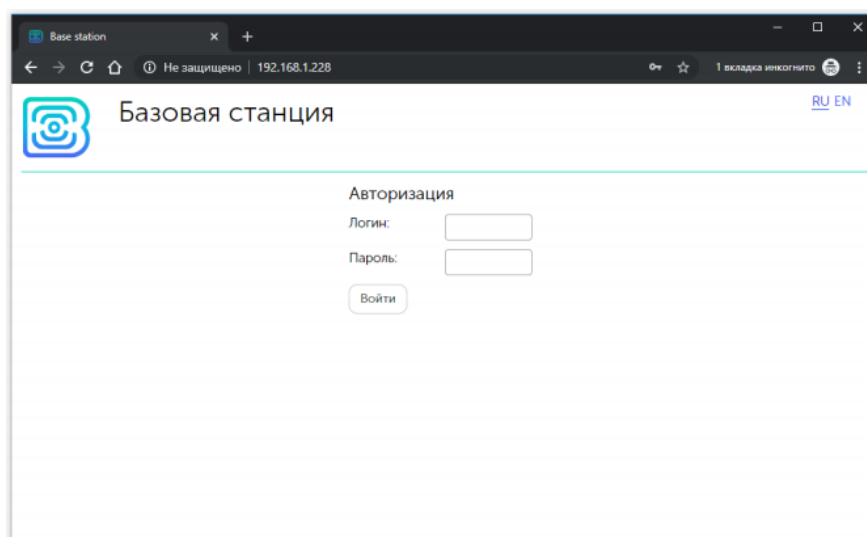
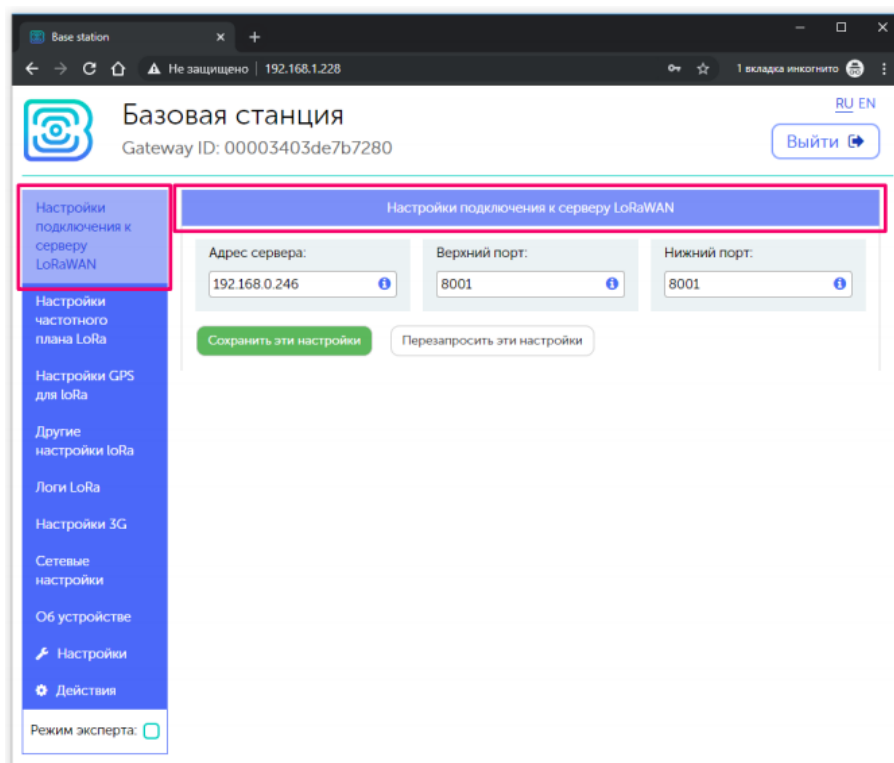


Рисунок 10 – Веб-интерфейс для настройки БС

Затем была выполнена настройка основных параметров БС через веб-интерфейс (рисунок 11), описанных ниже, все остальные же параметры остались по умолчанию:

- настройки подключения к серверу LoRaWAN - адрес будущего сервера, а также граница доступных портов для подключения внешних устройств.
- логи LoRa, здесь задается полное логирование всех событий.

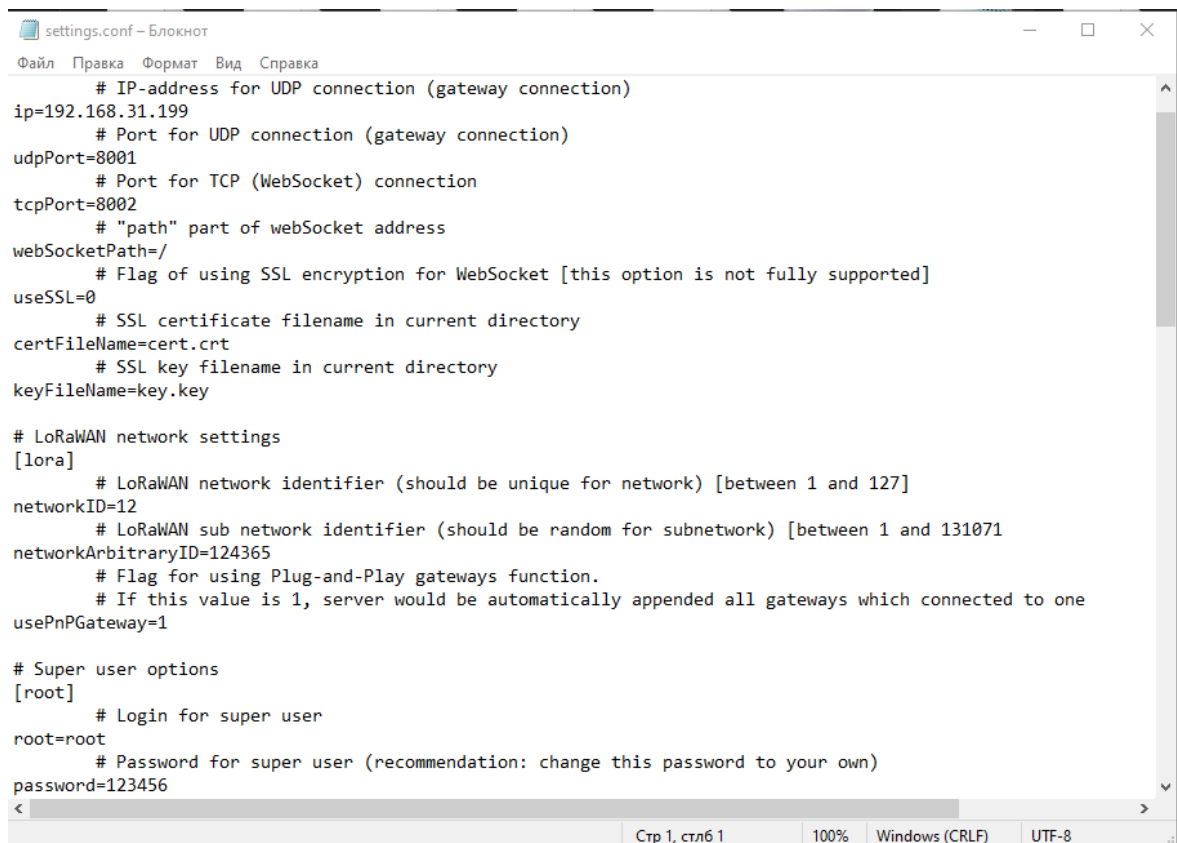




## 4.2 Работа с IoT Vega Server и IoT Vega Admin Tool

После был поставлен сетевой сервер IOT Vega Server - это инструмент для организации сетей стандарта LoRaWAN любого масштаба, который в данной работе предназначен для управления опорной сетью базовых станций, работающих под управлением ПО Packet forwarder от компании Semtech, приема данных с конечных устройств и передачи их внешним приложениям, а также передачи данных от внешних приложений на LoRaWAN® устройства. Полная информация о нём находится на сайте производителя. [12]

Перед тем, как выполнить запуск непосредственно самого сервера необходимо выполнить первичную настройку, а именно изменить несколько файл конфигурации “settings.conf”. (рис. 12)



```
settings.conf - Блокнот
Файл  Правка  Формат  Вид  Справка

# IP-address for UDP connection (gateway connection)
ip=192.168.31.199
# Port for UDP connection (gateway connection)
udpPort=8001
# Port for TCP (WebSocket) connection
tcpPort=8002
# "path" part of webSocket address
webSocketPath=/
# Flag of using SSL encryption for WebSocket [this option is not fully supported]
useSSL=0
# SSL certificate filename in current directory
certFileName=cert.crt
# SSL key filename in current directory
keyFileName=key.key

# LoRaWAN network settings
[lora]
# LoRaWAN network identifier (should be unique for network) [between 1 and 127]
networkID=12
# LoRaWAN sub network identifier (should be random for subnetwork) [between 1 and 131071]
networkArbitraryID=124365
# Flag for using Plug-and-Play gateways function.
# If this value is 1, server would be automatically appended all gateways which connected to one
usePnPGateway=1

# Super user options
[root]
# Login for super user
root=root
# Password for super user (recommendation: change this password to your own)
password=123456

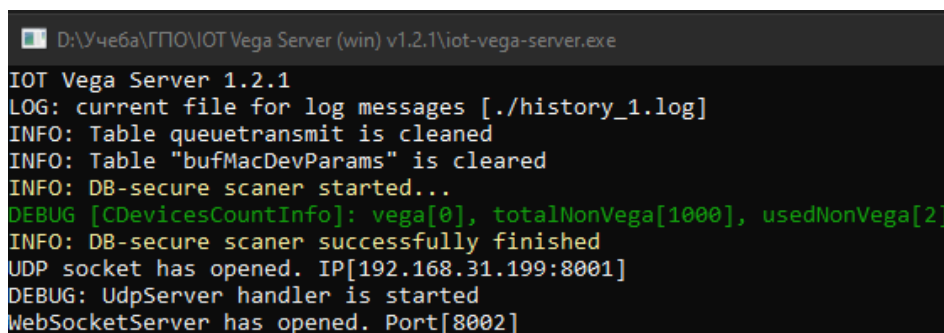
Стр 1, столб 1    100%    Windows (CRLF)    UTF-8
```

Рисунок 12 – Изменение файла конфигурации

В данном файле были изменены строки “ip” (отвечает за IP-адрес компьютера, на котором будет расположен сервер), а также строка “password”,

которая отвечает за пароль суперпользователя - он понадобится в дальнейшем при обращении к серверу через приложение IOT Vega AdminTool.

После первичной настройки был запущен непосредственно сам сервер. (рис. 13)



```

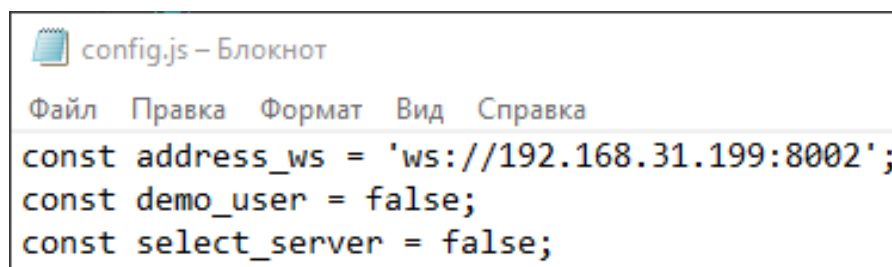
D:\Учеба\ГПО\IOT Vega Server (win) v1.2.1\iot-vega-server.exe
IOT Vega Server 1.2.1
LOG: current file for log messages [./history_1.log]
INFO: Table queuetransmit is cleaned
INFO: Table "bufMacDevParams" is cleared
INFO: DB-secure scanner started...
DEBUG [CDevicesCountInfo]: vega[0], totalNonVega[1000], usedNonVega[2]
INFO: DB-secure scanner successfully finished
UDP socket has opened. IP[192.168.31.199:8001]
DEBUG: UdpServer handler is started
WebSocketServer has opened. Port[8002]
  
```

Рисунок 13 – Запуск сервера

О том, что сервер работает корректно строки “UDP socket has opened” и “WebSocketServer has opened”.

Для удобного взаимодействия с сервером было использовано приложение IOT Vega Admin Tool. Оно открывает перед администратором сервера широкие возможности по управлению сетью LoRaWAN, например, с ним вы можете добавлять в сеть новые оконечные устройства, просматривать карту сети, контролировать базовые станции, а также управлять правами пользователей. Полная информация о данном приложении находится на сайте производителя. [13]

Перед тем, как начать работать с приложением был настроен файл конфигурации (рис. 14), а именно изменен постоянный IP-адрес сервера и номер порта в соответствии с ранее измененным файлом конфигурации.

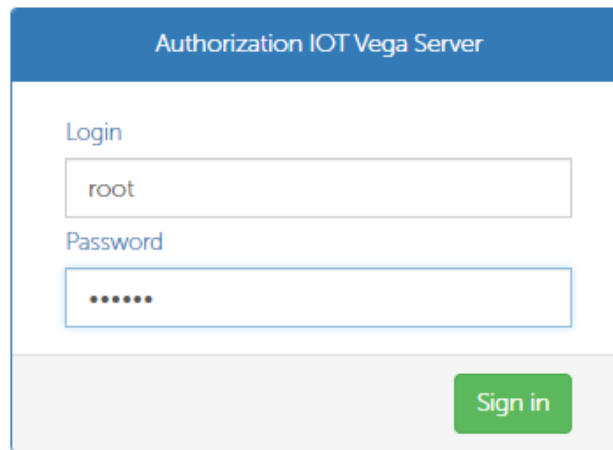


```

config.js – Блокнот
Файл  Правка  Формат  Вид  Справка
const address_ws = 'ws://192.168.31.199:8002';
const demo_user = false;
const select_server = false;
  
```

Рисунок 14 – Настройка файла конфигурации веб-приложения Admin Tool

Далее был произведен вход в веб-приложение, где используются логин и пароль суперпользователя, прописанные в файле конфигурации сервера. (рис. 15)



Authorization IOT Vega Server

Login

root

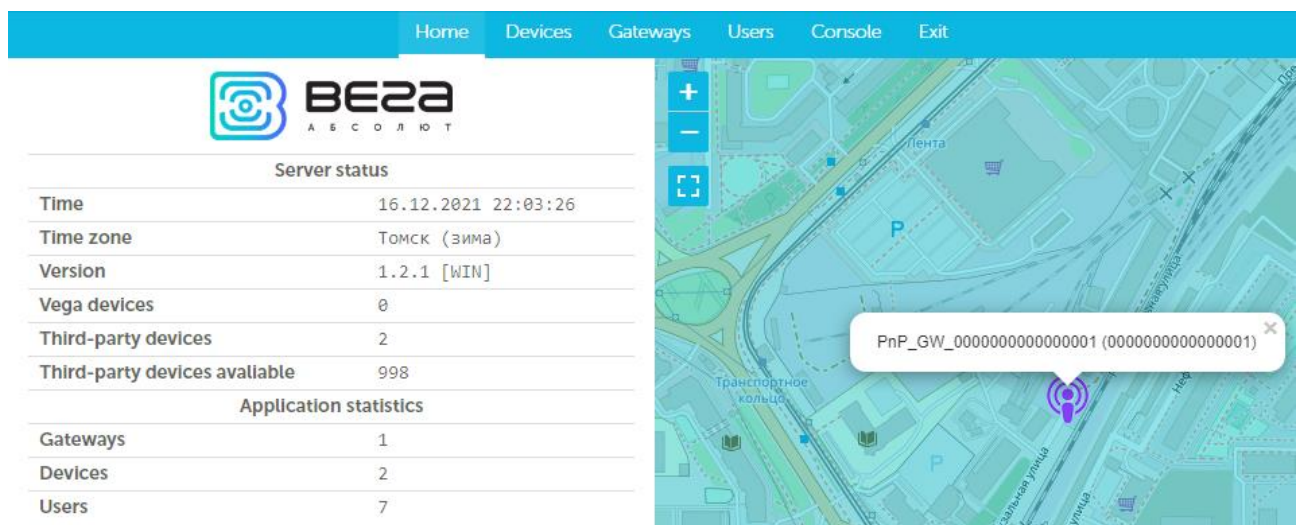
Password

.....

Sign in

Рисунок 15 – Окно авторизации в приложение IOT Vega Admin Tool

После авторизации в веб-приложении сразу же открывается вкладка “Home”, на которой можно увидеть информацию о времени, часовом поясе и версии сервера. Также на данной вкладке можно увидеть количество БС (gateways), количество подключенных оконечных устройств (Devices) и количество пользователей (users). В случае, если какое-либо устройство (БС или оконечное устройство) поддерживает GPS, то оно будет представлено на карте. (рис. 16)



Home Devices Gateways Users Console Exit

**BE2A**  
АБСОЛЮТ

**Server status**

Time	16.12.2021 22:03:26
Time zone	Томск (зима)
Version	1.2.1 [WIN]
Vega devices	0
Third-party devices	2
Third-party devices available	998

**Application statistics**

Gateways	1
Devices	2
Users	7

Map showing location of PnP\_GW\_0000000000000001 (0000000000000001)

Рисунок 16 – Вкладка “Home” веб-приложения IOT Vega Admin Tool

Для того, чтобы убедиться наверняка, что БС подключена к серверу был выполнен переход во вкладку “gateways”. (рис. 17)

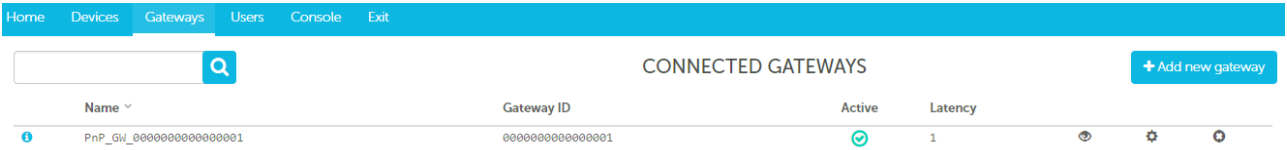


Рисунок 17 – Вкладка “Gateways” веб-приложения IOT Vega Admin Tool

Далее были созданы несколько пользователей, который понадобятся в дальнейшем, а именно: admin, humidity\_admin, pressure\_admin, temperature\_admin и user. (рис. 18-19)

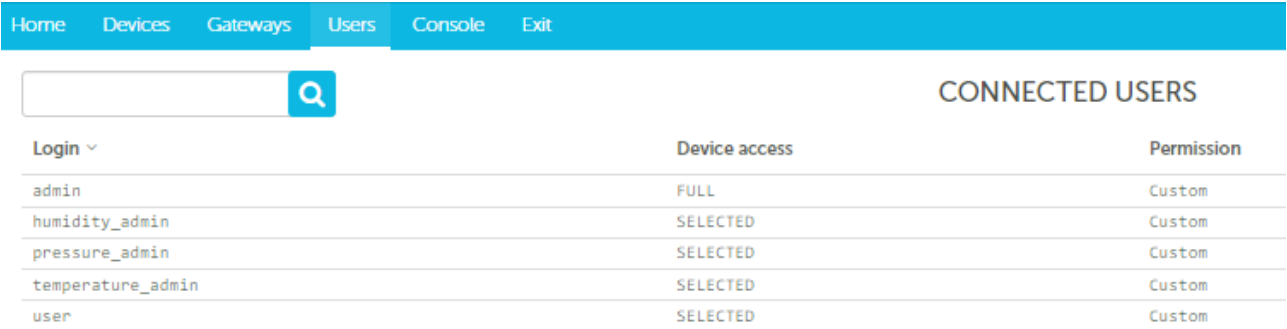


Рисунок 18 – Список пользователей сервера

User settings

Main settings

Login

admin

Permissions pattern

Custom

User privileges

Device access

All devices

☐ Send online data

☐ Console enable

Permissions commands api

Common

☒ View data from devices

☒ Send data to devices

☒ Send TX to devices

External application

☒ View devices by external app

☒ Add/edit devices by external app

☒ Delete devices by external app

Network management

☒ View users

☒ Create/edit users

☒ Delete users

☒ View gateways

☒ Create/edit gateways

☒ Delete gateways

☒ View registered devices

☒ Registered/edit devices

☒ Delete devices from network

☒ View network coverage map

☒ View server downlink queue

☐ Edit server downlink queue

☒ Getting server statistics

☐ Send email

☒ Select all

☒ Unselect all

Close

Save

Рисунок 19 – Настройка прав у пользователя “admin”

В настройках прав у пользователя “admin” (рис.19) имеется три группы прав: общие права (common), права для работы с внешним приложением (External application) и права для управления сетью (network management). Также

имеется раздел “User privileges”, отвечающий за то, к каким оконечным устройствам пользователь имеет право доступа.

Подобно тому, как были настроены права у пользователя “admin” на рисунке выше были настроены права у остальных пользователей. Некоторая информация за что отвечает, каждый из пользователей:

- admin - администратор сервера, у которого есть доступ на все устройства и все разрешения;
- humidity\_admin - администратор сервера, у которого будет доступ только к устройствам, которые измеряют влажность воздуха в аудитории и все разрешения в рамках данных устройств;
- pressure\_admin - администратор сервера, у которого будет доступ только к устройствам, которые измеряют давление в аудитории и все разрешения в рамках данных устройств;
- temperature\_admin - администратор сервера, у которого будет доступ только к устройствам, которые измеряют температуру в аудитории и все разрешения в рамках данных устройств;
- user - обычный пользователь - у которого будет доступ только к некоторым устройствам, в рамках которых он может лишь смотреть данные, которые отправляются с них (какая температура, давление или влажность в аудитории).

Требования к клиентскому приложению.

Для ещё более удобного взаимодействия было принято решение разработать клиентское приложение, которое будет отправлять запросы в базу данных (далее просто БД) сервера для того, чтобы получить данные с о температуре, влажности и давлении в той или иной аудитории.

Клиентское приложение должно отвечать следующим требованиям:

- синхронизация пользователей с сервером в режиме онлайн;
- возможность авторизации под различными пользователями путем ввода логина и пароля (те самые пользователи, которые были созданы ранее);

- синхронизация данных с БД сервера;
- возможность просматривать данные, отправленные с оконечных устройств на сервер;
- возможность отправить запрос на оконечное устройство для того, чтобы оно отправило актуальные данные;
- возможность просматривать состояние оконечных устройств (активно или неактивно);
- возможность подключать новые оконечные устройства и настраивать их.
- создание аналитики в виде различных графиков по полученным данным с сервера;
- отправка уведомлений на телефон, в виде SMS либо на почту в виде письма в случае, если какие-либо данные достигли критической (например, температура слишком низкая или наоборот слишком высокая).
- отправка уведомлений в случае, если какое-либо оконечное устройство уже более трех часов не отправляет данные.

Получение удаленного сервера и установка виртуальной машины.

В условиях пандемии в случае, если группу закрывают на карантин, то один из студентов “выбывается” из работы, т.к. необходимо работать с БС напрямую. Из-за этого было принято решение развернуть сервер БС на виртуальном частном сервере (далее VPS) с публичным IP-адресом.

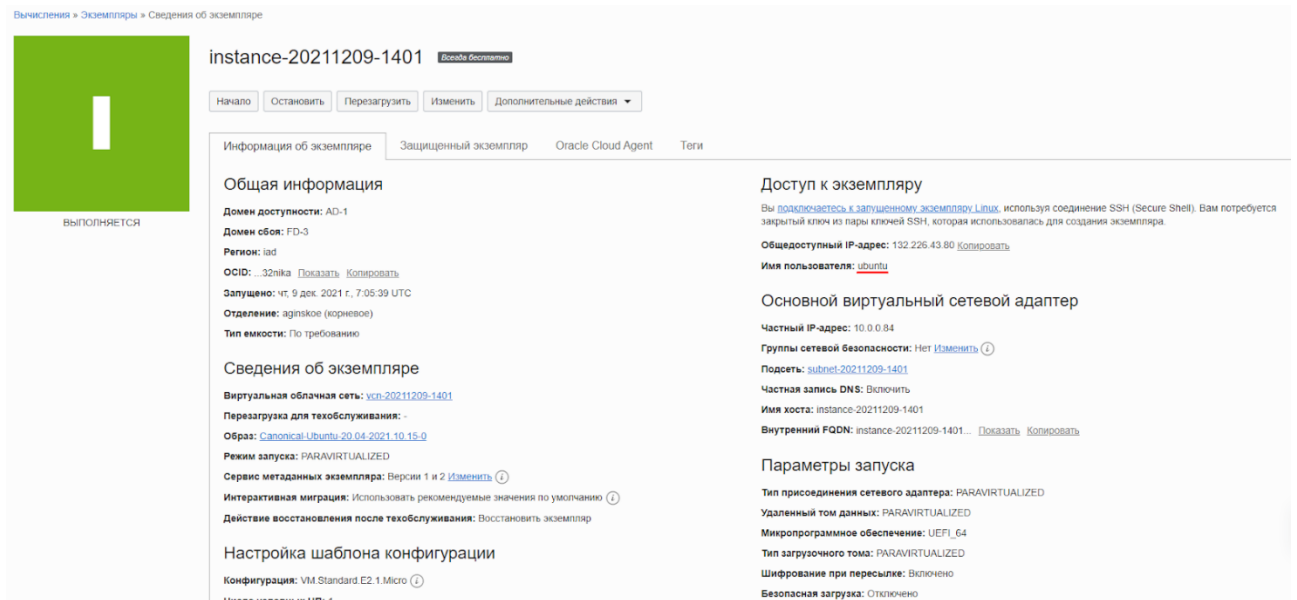
Для начала был выполнен анализ того, какие компании выдают бесплатные VPS-серверы с публичным IP-адресом. Оказалось, что таких компаний 3: Oracle, Google и Amazon. Технические характеристики, предоставляемых данными компаниями серверов. (табл. 1) [14]

Таблица 1 – Технические характеристики серверов

Компания	Oracle	Google	Amazon
Процессор	AMD EPYC 7551	Intel Xeon	Intel Xeon
Количество ядер	2	1	1
Частота процессора, МГц	2000	2200	2500
Количество оперативной памяти, Мб	1024	583	1024

По итогу был выбран сервер от компании Oracle. Для получения сервера была выполнена регистрация на их сайте. Далее на сервер была установлена виртуальная машина под операционной системой (далее ОС) “ubuntu. (рис. 20)

При этом во время установки виртуальной машины был сгенерирован и добавлен ключ SSH, который в дальнейшем был использован для авторизации через терминальную программу “PuTTY”. [15] [16]



Вычисления > Экземпляры > Сведения об экземпляре

instance-20211209-1401 Кнопка безопасности

Начало Остановить Перезагрузить Изменить Дополнительные действия

Информация об экземпляре Защищенный экземпляр Oracle Cloud Agent Тег

### Общая информация

Домен доступности: AD-1  
 Домен сбоя: FD-3  
 Регион: iad  
 OCID: :32nka [Показать](#) [Копировать](#)  
 Запущено: чт, 9 дек, 2021 г., 7:05:39 UTC  
 Отделение: адп/накоп (корневое)  
 Тип емкости: По требованию

### Сведения об экземпляре

Виртуальная облачная сеть: [vcn-20211209-1401](#)  
 Перезагрузка для техобслуживания: -  
 Образ: [Canonical.Ubuntu.20.04.2021.10.15.0](#)  
 Режим запуска: PARAVIRTUALIZED  
 Сервис метаданных экземпляра: Версия 1 и 2 [Изменить](#) ⓘ  
 Интерактивная миграция: Использовать рекомендуемые значения по умолчанию ⓘ  
 Действие восстановления после техобслуживания: Восстановить экземпляр

### Настройка шаблона конфигурации

Конфигурация: VM.Standard.E2.1.Micro ⓘ  
 Число условных CPU: 1

### Доступ к экземпляру

Вы [попытаетесь](#) [соединиться](#) с запущенным экземпляром Linux, используя соединение SSH (Secure Shell). Вам потребуется закрытый ключ из пары ключей SSH, которая использовалась для создания экземпляра.

Общедоступный IP-адрес: 132.226.43.80 [Копировать](#)  
 Имя пользователя: [ubuntu](#)

### Основной виртуальный сетевой адаптер

Частный IP-адрес: 10.0.0.84  
 Группы сетевой безопасности: Нет [Изменить](#) ⓘ  
 Подсеть: [subnet-20211209-1401](#)  
 Частная запись DNS: Включить  
 Имя хоста: instance-20211209-1401  
 Внутренний FQDN: instance-20211209-1401... [Показать](#) [Копировать](#)

### Параметры запуска

Тип присоединения сетевого адаптера: PARAVIRTUALIZED  
 Удаленный том данных: PARAVIRTUALIZED  
 Микропрограммное обеспечение: UEFI\_64  
 Тип загрузочного тома: PARAVIRTUALIZED  
 Шифрование при пересылке: Включено  
 Безопасная загрузка: Отключено

Рисунок 20 – Сведения об виртуальной машине

Дальнейшая настройка полученного сервера будет произведена в следующем семестре. В рамках настройки планируется установить на нем IOT



Vega Server и приложение для администрирования через веб-интерфейс Admin Tool.

### 4.3 Сетевое взаимодействие конечного устройства и сервера

Перед присоединением конечного устройства к сети, его необходимо настроить. Для решения поставленной задачи отлично подходят устройства класса А с методом активации ОТАА.

Настройка проводилась с помощью специальной утилиты от производителя устройств RAK - RAK Serial Port Tool. На рис. 21 представлено окно данной утилиты. Слева представлено окно вывода, а справа – список некоторых АТ-команд.

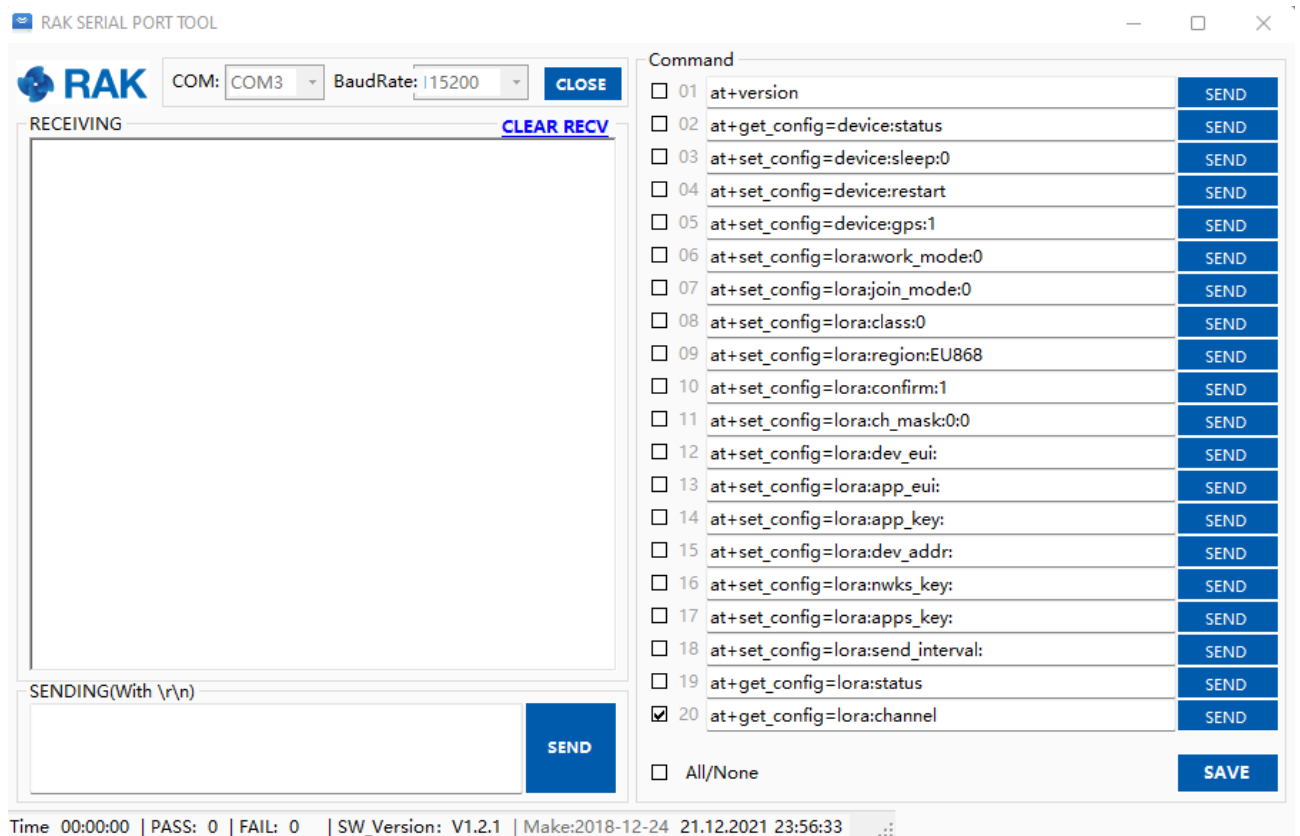
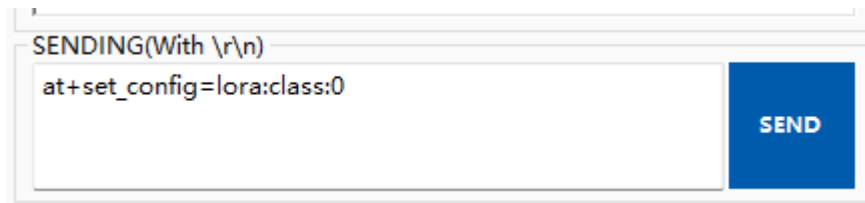


Рисунок 21 – RAK Serial Port Tool

Класс А был выбран, потому что:

- устройство питается от батареи;
- имеет самое низкое энергопотребление. [17]

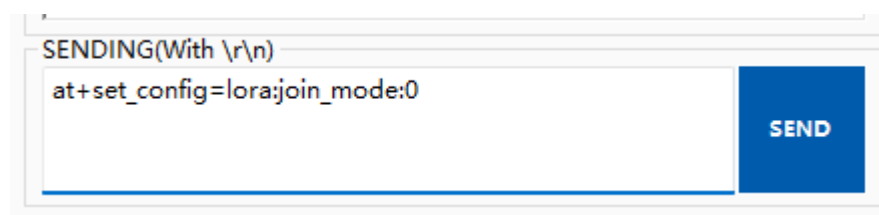
Данный параметр настраивается с помощью АТ-команды `at+set_config=lora:class:0`, где 0 устанавливает класс А (рис. 22).



```
SENDING(With \r\n)
at+set_config=lora:class:0
```

Рисунок 22 – Настройка класса устройства

Активация по воздуху (ОТАА) наиболее безопасный и рекомендуемый метод. Для его настройки была использована команда `at+set_config=lora:join_mode:0`, где 0 устанавливает ОТАА (рис. 23). [18]



```
SENDING(With \r\n)
at+set_config=lora:join_mode:0
```

Рисунок 23 – Настройка метода активации

Так же были изучены и настроены коэффициенты распределения (spreading factors). Скорость передачи данных должна быть максимально быстрой, чтобы свести к минимуму эфирное время. SF7BW125 хороший коэффициент для начала, так как он потребляет меньше всего энергии и эфирного времени. Если будет нужен больший диапазон, можно медленно увеличивать коэффициент, пока он не удовлетворит потребностям. [19]

Изменять Коэффициент распределения в устройствах LoRa можно с помощью at-команды `at+set_config=lora:dr:X`, где X - номер Data Rate от 0 до 5 (рис. 24).



```
SENDING(With \r\n)
at+set_config=lora:dr:5
```

Рисунок 24 – Настройка коэффициента распределения

Список AT-команд для RAK811 можно посмотреть в документации. [20]  
После настройки устройство было добавлено на сервер.

#### 4.4 Описание устройства для снятия микроклиматических параметров

Выбор и описание микроконтроллера.

Для данного проекта использовался модуль микроконтроллера STM32 — это наиболее популярное и широко используемое за счет низкой стоимости семейство микроконтроллеров, используемых и в реальной жизни, а не только в обучении. Программирование происходило на языке C.

Из данного семейства была выбрана отладочная плата NUCLEO-F103RB. На рисунке 25 представлено фото данного микроконтроллера.



Рисунок 25 – Отладочная плата NUCLEO-F103RB

Это стандартная учебная плата от ST на настоящий момент. Рекомендуется использовать именно её. В магазинах есть большой выбор таких плат с разными

характеристиками микроконтроллера, некоторые из них неудобны из-за малого количества памяти. Для реализации проекта памяти в данной модели достаточно.

Положительные стороны выбора данного микроконтроллера, следующие:

- Есть встроенный USB-UART (интерфейс для коммуникации платы и компьютера через консольный ввод/вывод).
- Легко доступна в продаже (как в плане цены, так и количества в продаже).
- Очень удобно перепрошивать: определяется как флэшка, поэтому на нее достаточно просто "бросить" файл, и это работает и в Linux, и в Windows.
- Совместима с Mbed без дополнительных настроек, изначально.

Минусы данной платы, следующие:

- Много различных версий с разными характеристиками (легко запутаться).
- Устаревший разъём MiniUSB (и одновременно преимущество – намного крепче использующегося во многих современных платах micro-USB).

Ниже приведены основные технические характеристики данной платы:

[21]

- Микроконтроллер STM32 в корпусе LQFP64.
- 1 пользовательский светодиод.
- Пользовательская кнопка и кнопка сброса.
- Кварцевый генератор частотой 32,768 кГц
- Разъемы платы: разъемы расширения ARDUINO Uno V3 и разъемы расширения morpho для полного доступа ко всем входам/выходам STM32.
- Гибкие варианты питания: ST-LINK, USB VBUS и возможность питания от внешних источников.
- Встроенный отладчик/программатор ST-LINK с возможностью повторного перечисления USB: накопитель, виртуальный COM-порт и порт отладки.

The diagram illustrates the pinout for the STM32NU60 microcontroller, showing connections for various pins and their functions. The pins are color-coded and labeled with their functions and corresponding pin numbers.

**Power and Reset Pins:**

- PC\_13: USER\_BUTTON
- IOREF
- NRST
- +3v3
- +5v
- GND
- VIN
- 5v - 9v

**Digital Pins:**

- PC\_13: USER\_BUTTON
- LED1
- PWM4/3
- PWM4/4
- PWM3/2
- PWM4/1
- PWM3/2
- PWM2/3
- PWM3/1
- PWM3/2
- PWM2/2
- PWM2/3
- PWM2/4

**Analog Pins:**

- A0
- A1
- A2
- A3
- A4
- A5
- Analog In
- Analog Out

**Peripheral Pins:**

- I2C1 SCL
- I2C1 SDA
- D15
- D14
- SP1 MISO
- SP1 MOSI
- I2C1 SCL
- D13
- D12
- D11
- D10
- D9
- D8
- D7
- D6
- D5
- D4
- D3
- D2
- D1
- D0
- Serial1 TX
- Serial1 RX
- Serial2 TX
- Serial2 RX
- Serial3 TX
- I2C2 SCL
- SP1 MISO
- SP1 MOSI
- SCK
- Analog In

The diagram also shows the connections for the microcontroller's internal components, including the STM32NU60 chip, the I2C1 and I2C2 controllers, the SPI1 controller, the Serial1 and Serial2 controllers, and the Analog In and Analog Out modules.

Почти все выходы можно использовать для своих целей (учитывая назначение пинов). К примеру, выводы D0 и D1 задействовать не рекомендуется, так как они уже заняты отладочной консолью. Учитывайте, что D0, D1 и так далее пронумерованы так только на самой плате (маленькие номера у каждого вывода). Выводы же с точки зрения микроконтроллера называются иначе - они сгруппированы в порты (PORT, или просто P), и в рамках каждого порта



пронумерованы (например, PA\_6, на плате этот вывод подписан как 12). То, что существует две разных нумерации, может показаться немного сложным, но если вы посмотрите на картинку ниже, то поймете их взаимное соответствие. Есть также третья нумерация – это абсолютная нумерация выводов микросхемы, то есть просто по номерам, без всяких букв. На данной схеме она не показана. [22]

#### Описание сенсора.

Согласно задаче проекта необходимо было выбрать датчики, которые специализируются на снятии микроклиматических параметров, а именно температуры, влажности и давления. По причине того, что в распоряжении уже были сенсоры от компании unwireddevices – UMDK-THP - датчик давления, влажности и температуры воздуха на базе высокоточного цифрового сенсора Bosch BME280 Схема датчика показана на рисунке 27. [23]

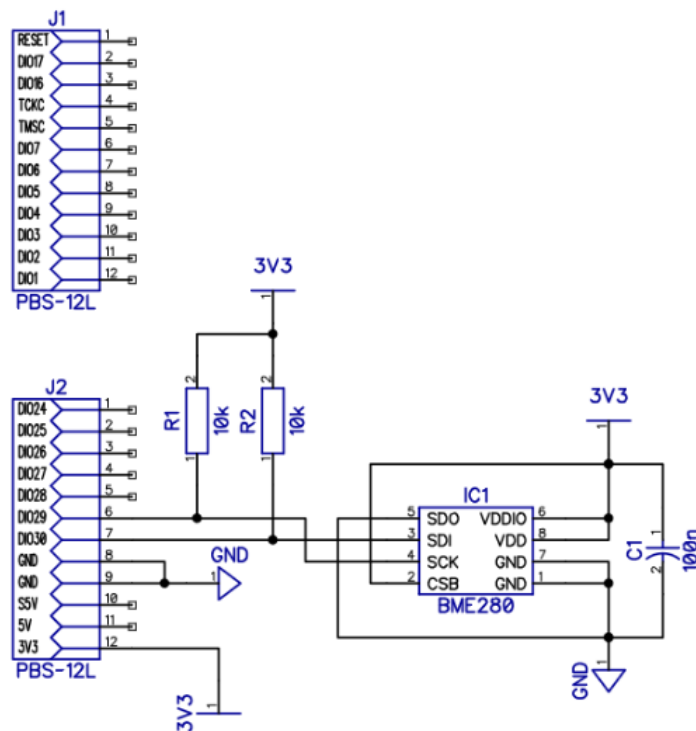


Рисунок 27 – Схема сенсора

В данном сенсоре используется шина I2C, соответственно, совместим со всеми I2C-модулями (это может быть полезно в случае, если устройство в будущем будет использовать другие датчики с таким интерфейсом). Как видно

на схеме, используются GPIO 29 и GPIO 30 (на схеме – DIO29 и DIO30), идущие в сам сенсор на входы SDI и SCK. Данные входы подключаются к выходам на плате – D14 и D15 (на рисунке 28 данные выходы выделены). [24]

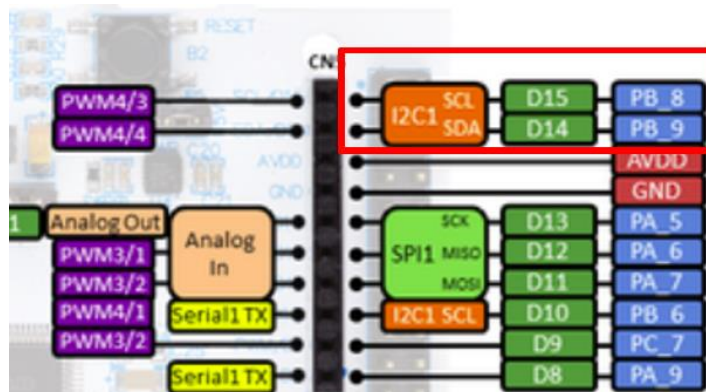


Рисунок 28 – Используемые выходы на микроконтроллере для инициализации датчика

Фотография самого сенсора представлена на рисунке 29.

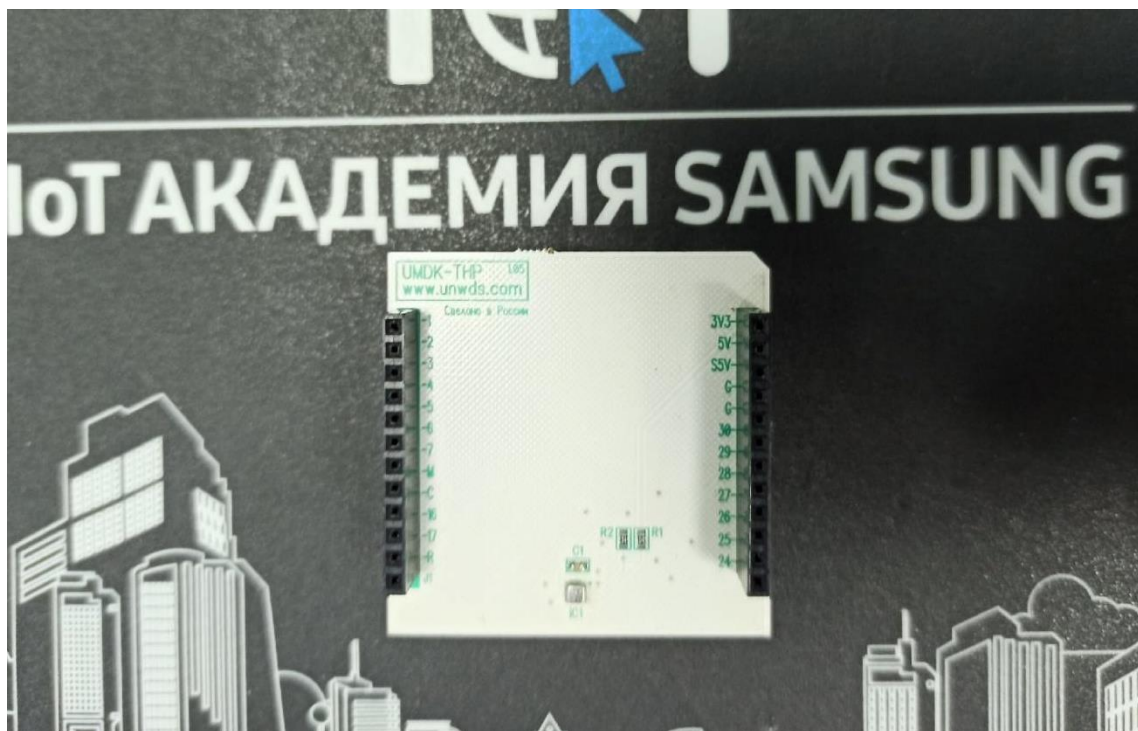


Рисунок 29 - Фотография сенсора UMDK-THP



Описание устройства-прототипа.

После нахождения всех необходимых устройств производился их сбор в одно цельно работающее устройство под командованием микроконтроллера STM32. На рисунке 30 представлена схема подключения.

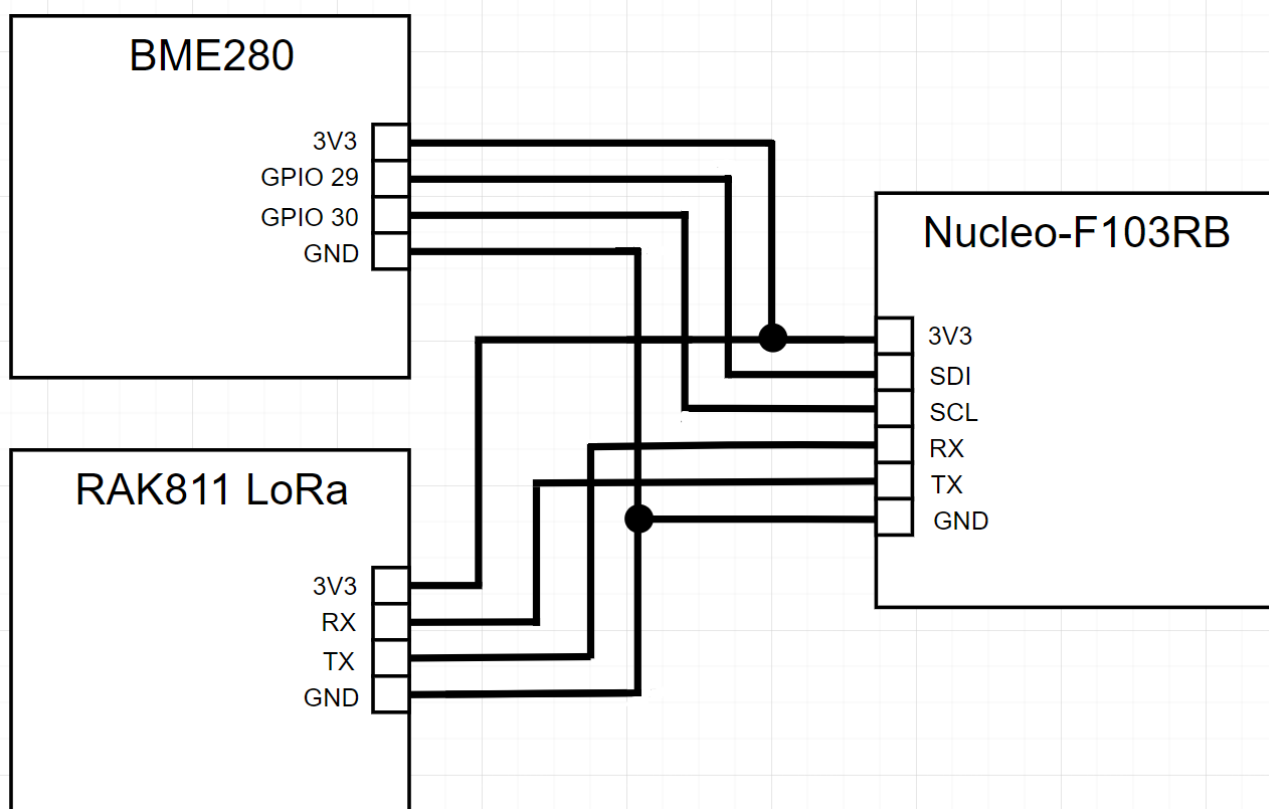


Рисунок 30 – Схема подключения

На рисунке 31 представлена фотография собранного прототипа.

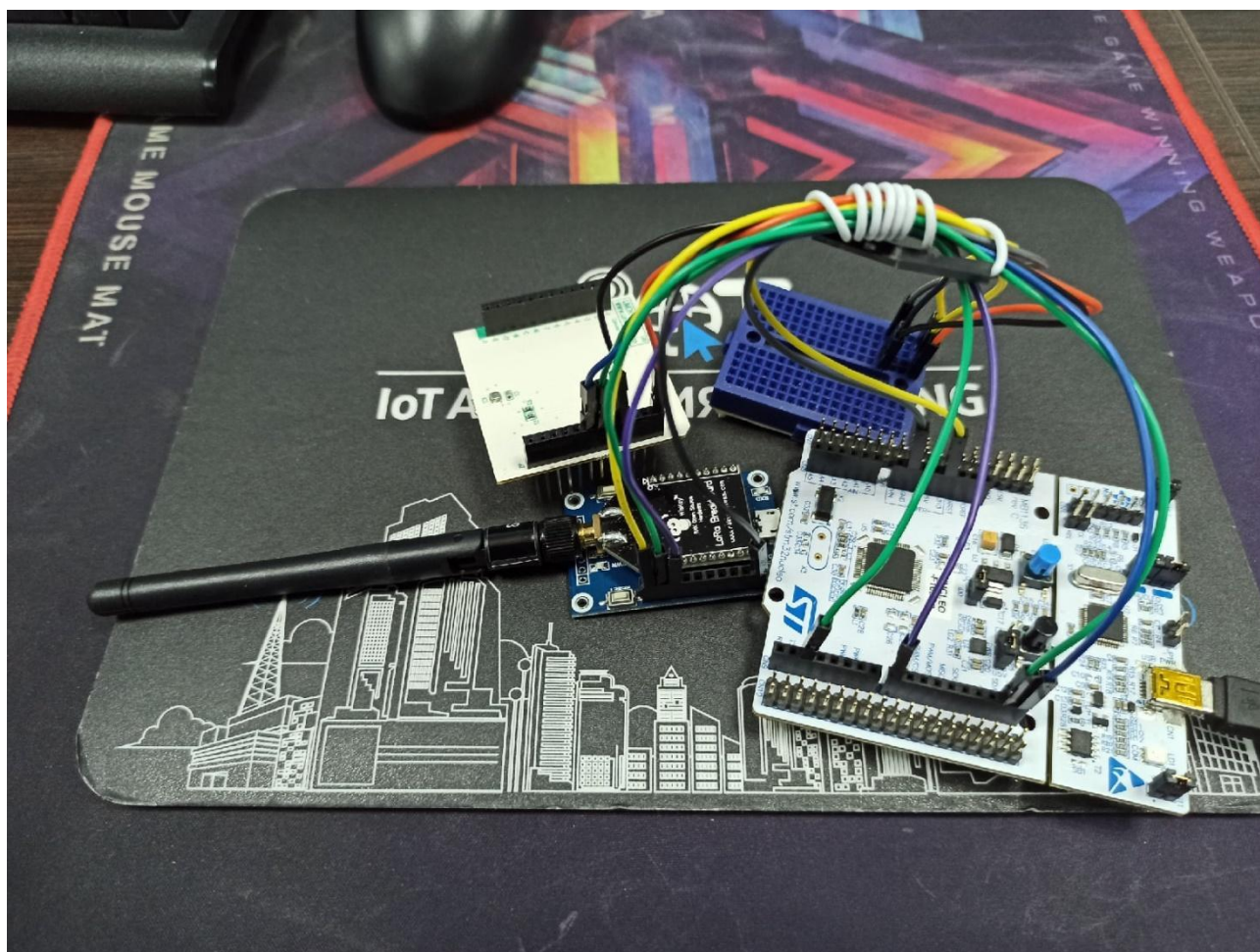


Рисунок 31 - Фотография прототипа устройства по снятию микроклиматических параметров

В данном прототипе присутствуют ранее описанные устройства: микроконтроллер STM32 Nucleo F103RB, сенсор BME280, конечное устройство RAK 811 LoRa, а также разветвитель. Последний необходим был, так как количество выходов питания и заземления ограничено, для всех устройств их не хватало при прямом подключении к микроконтроллеру.

Как было сказано, все устройства питаются от микроконтроллера, который сам подключен к источнику (в данном случае – через USB к компьютеру, от которого одновременно загружается необходимая прошивка). Питание осуществляется выводом на 3.3 В. Аналогично с выводом GND – заземлением. Выходы сенсора подключены следующим образом: питание и заземление – к выводам питания 3V3 и GND на плате микроконтроллера, выводы для снятия показаний с сенсоров подключены так: DIO29 (на рисунке 27) подключен к D15

на плате микроконтроллера (SCL, также подписан как D15 или PB\_8 на рисунке 28), DIO30 – к D14 (SDA, также D14 или PB\_9).

Конечное устройство RAK 811 LoRa подключено при помощи XBee USB Adapter – “переходник” между платой микроконтроллера и модулем. На рисунке 32 представлена фотография данного адаптера.



Рисунок 32 – XBee USB Adapter

Он необходим, так как форм-фактор RAK 811 отличается от остальных – шаг соединительных проводов используется другой. Подключается модуль LoRa следующим образом: питание и заземление – аналогично сенсору, при помощи разветвителя. Выводы RX (прием) и TX (отправка) подсоединены к плате микроконтроллера следующим образом: RX подключается к выводу D8 (PA\_9), TX – к D2 (PA\_10). Также заметно то, что эти выводы подключены “крест-накрест”. Такое подключение объясняется тем, что вывод информации от модуля должен приниматься со стороны платы, а за это будет отвечать выход приема информации, то есть противоположный по смыслу вывод.

На рисунке 33 показаны выводы на плате микросхемы.

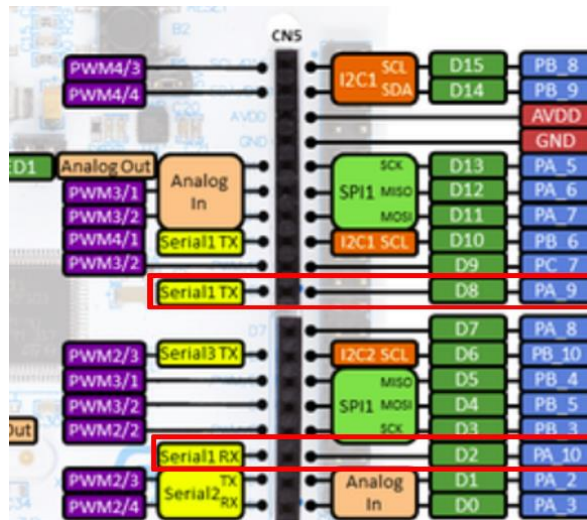


Рисунок 33 – Подключаемые выводы платы к модулю RAK 811

### Программирование микроконтроллера.

В качестве задачи по программированию микроконтроллера было создание структуры данных, содержащей в себе снятые сенсором микроклиматические параметры – температуру, давление и влажность. Также необходимо было предусмотреть ситуации, когда данные являются критическими и режимы сна всего устройства в целом.

Первым делом была изучена библиотека “BME280.h”, в которую были внесены некоторые изменения. Программирование платы производилось в Mbed Studio – среде разработки, которая специализирована на создании прошивки для микроконтроллеров Mbed-типа (к ним относится использующаяся в проекте STM32 Nucleo). В данной среде по умолчанию не используется тип данных float, а методы упомянутой библиотеки, возвращающие значения микроклиматических параметров (getTemperature(), getPressure(), getHumidity()), имеют тип возвращаемых параметров как раз-таки в float-типе. Поэтому было сделано следующее: тип данных методов был изменен на целочисленный, integer. Проблемы в том, что дробная часть будет отбрасываться, нет, так как округляющаяся часть, равная 0,5 для каждой из величин (в большую или меньшую сторону), не имеет значения.

Сама структура данных выглядит следующим образом: 1 байт (один символ) для несения информации о том, являются ли снятые параметры

нормальными (далее будет рассказано о нормативной части, где говорится о допустимых значениях микроклиматических показаний), далее 4 байта – данные о температуре (int, затем конвертируется в массив символов, соответственно варьируется от 1 до 3 символов, соответственно, от 1 до 3 байт), далее, аналогично температуре, давление (только уже 3 или 4 байта символов) и влажность (также конвертируется int-значение 4-байтное в массив символов от 1 до 3 символов).

На рисунке 34 представлена блок-схема прошивки.

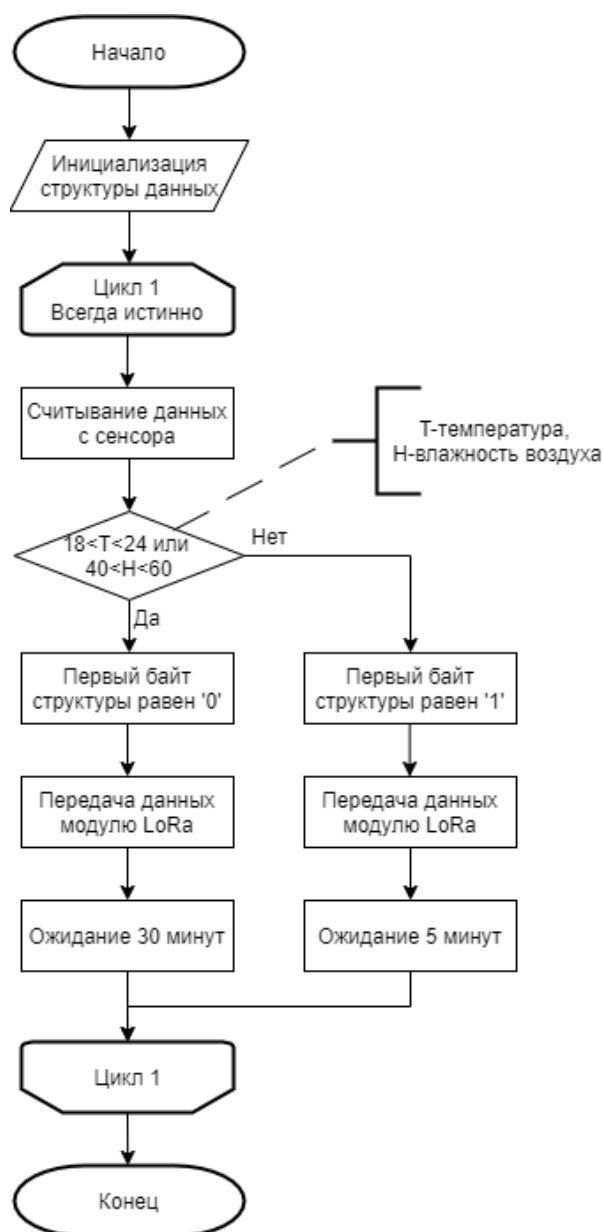


Рисунок 34 – Блок схема кода прошивки



На рисунке 35 показана инициализация структуры и то, каким образом происходит преобразование из `int` в символы.

```
struct data
{
    char Warning;
    char Temperature[MAX_DIGITS + sizeof(char)];
    char Pressure[MAX_DIGITS + sizeof(char)];
    char Humidity[MAX_DIGITS + sizeof(char)];
} data;

std::to_chars(data.Temperature, data.Temperature + MAX_DIGITS, sensor.getTemperature());
std::to_chars(data.Pressure, data.Pressure + MAX_DIGITS, sensor.getPressure());
std::to_chars(data.Humidity, data.Humidity + MAX_DIGITS, sensor.getHumidity());
```

Рисунок 35 – Созданная структура данных data

Особое внимание было обращено на нормативную составляющую, а именно допустимые величины параметров микроклимата в образовательных организациях, а именно: диапазон допустимой температуры воздуха в учебных кабинетах – 18-24°C; относительная влажность воздуха – 40-60%. В соответствии с этим в структуру было добавлено значение, которое показывало бы, соответствуют ли нормам снятые показания или нет. На рисунке 36 показана часть кода, показывающая, каким образом происходит сравнение с рекомендуемыми показаниями. [25]

```
if (sensor.getTemperature() < 18 || sensor.getTemperature() > 24 || sensor.getHumidity() < 40 || sensor.getHumidity() > 60)
{
    pc.attach(&pc_recv, Serial::RxIrq);
    dev.attach(&dev_recv, Serial::RxIrq);

    data.Warning = '1';
    wait(18000);
    sleep();
}
else
{
    pc.attach(&pc_recv, Serial::RxIrq);
    dev.attach(&dev_recv, Serial::RxIrq);

    data.Warning = '0';
    wait(10000);
    sleep();
}
```

Рисунок 36 – Проверка данных с сенсора

Как видно из рисунка, при выполнившемся условии, что параметры не соответствуют нормальным, поле из структуры данных `warning` принимает значение 1.

Также можно заметить то, как часто снимаются показания в зависимости от их качества. При соблюденных параметрах съем происходит один раз в 30 минут. Если идет отклонение от указанного диапазона, то съем – раз в 5 минут. Это можно видеть на рисунке выше в строке с вызовом функции wait().

В качестве проверки того, в каком виде передаются параметры, был открыт терминал и с помощью функции по выводу в него информации показания были выведены. На рисунке 37 показаны данные в терминале.

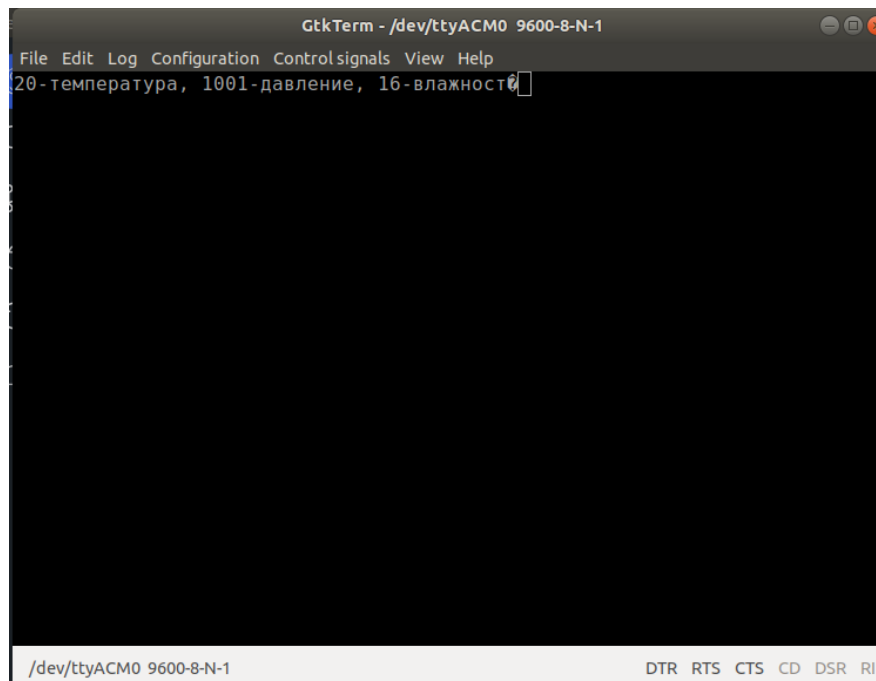


Рисунок 37 – Проверка снятия показаний и реализованной структуры

Весь код прошивки представлен в приложении А.

## 5 Заключение

В ходе выполнения проекта «Исследование и создание IoT-сети, основанной на технологиях модуляции LoRa и сетевого протокола LoRaWAN» были изучены документации по использованию оборудования и технологий, при помощи которых они работают и взаимодействуют между собой, была построена работоспособная IoT-сеть, с помощью которой будет обрабатываться и передаваться микроклиматические параметры в учебных аудиториях.

Были изучены документации по техническим характеристикам и эксплуатации базовой станции Вега БС-2.2. Произведена её установка и настройка в учебной аудитории.

Была выполнена установка и настройка IOT Vega Server. Было установлено приложение администрирования IOT Vega Admin Tool, в котором были созданы пользователи. Были разработаны требования к клиентскому приложению. Было выполнено получение удаленного сервера, на который была установлена операционная система.

Были выбраны класс конечных устройств, метод активации и коэффициент распределения. Было изучено сетевое взаимодействие устройств с сервером: способ отправки и получения данных, скорость передачи данных, типы сообщений, ограничения и безопасность.

На этапе работы с микроконтроллером и сенсором была написана программа-прошивка для отладочной платы, реализующая прием и структуризацию данных с датчика по снятию микроклиматических параметров. Также был собран прототип устройства, занимающийся сбором этих параметров и отправки их на сервер.

Пояснительная записка и инструкция пользователю устройства были написаны в соответствии с требованиями ОС ТУСУР 01-2013.



## Список использованных источников

1. Образовательный стандарт ВУЗа ОС ТУСУР 01-2013. [Электронный ресурс]. – Режим доступа: [https://storage.tusur.ru/files/40668/rules\\_tech\\_01-2013.pdf](https://storage.tusur.ru/files/40668/rules_tech_01-2013.pdf) (дата обращения 15.12.2021).
2. Что такое IoT и что о нем следует знать. [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/otus/blog/549550/> (дата обращения 16.12.2021).
3. Global IoT Market Will Grow to 24.1 Billion Devices in 2030, Generating \$1.5 Trillion Annual Revenue. [Электронный ресурс]. – Режим доступа: <https://www.prnewswire.com/news-releases/global-iot-market-will-grow-to-24-1-billion-devices-in-2030--generating-1-5-trillion-annual-revenue-301061873.html> (дата обращения: 17.12.2021).
4. Обзор технологий non-cellular LPWAN. [Электронный ресурс]. – Режим доступа: <https://itechinfo.ru/content/обзор-технологий-non-cellular-lpwan> (дата обращения 18.12.2021).
5. Технология LoRa в вопросах и ответах. [Электронный ресурс]. – Режим доступа: <https://wireless-e.ru/standarty/tehnologiya-lora-v-voprosah-i-otvetah/> (дата обращения 19.12.2021).
6. Спецификация LoRaWAN. Введение. Основные понятия и классы оконечных устройств. [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/316954/> (дата обращения 20.12.2021).
7. Спецификация LoRaWAN. Активация оконечных устройств. [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/317218/> (дата обращения 21.12.2021).
8. Spreading Factors. [Электронный ресурс]. – Режим доступа: <https://www.thethingsnetwork.org/docs/lorawan/spreading-factors/> (дата обращения 16.12.2021).

9. Подробное описание интерфейса I2C. [Электронный ресурс]. – Режим доступа: <https://radioham.ru/i2c/> (дата обращения 16.12.2021).
10. Вега БС-2.2 - базовая станция. [Электронный ресурс]. – Режим доступа: <https://iotvega.com/product/bs02-2> (дата обращения 09.09.2021).
11. Руководство по эксплуатации. [Электронный ресурс]. – Режим доступа: [https://iotvega.com/content/ru/bs/bs01/BEGA%20BC%20ПИ\\_rev%2024.pdf](https://iotvega.com/content/ru/bs/bs01/BEGA%20BC%20ПИ_rev%2024.pdf) (дата обращения 09.09.2021).
12. IoT Vega Server. [Электронный ресурс]. – Режим доступа: <https://iotvega.com/soft/server> (дата обращения 16.09.2021).
13. IoT Vega Admin Tool. [Электронный ресурс]. – Режим доступа: [https://iotvega.com/content/ru/soft/server/IOT%20Vega%20Admin%20Tool%20V1.1.6\\_ru.zip](https://iotvega.com/content/ru/soft/server/IOT%20Vega%20Admin%20Tool%20V1.1.6_ru.zip) (дата обращения 16.09.2021).
14. Как получить бесплатный VPS (VDS) сервер навсегда. [Электронный ресурс]. – Режим доступа: <https://seopulses.ru/kak-poluchit-besplatniy-vps-vds-server-navsegda/> (дата обращения 03.12.2021).
15. Бесплатный VPS (VDS) от Oracle навсегда: как получить. [Электронный ресурс]. – Режим доступа: <https://seopulses.ru/besplatniy-vps-vds-v-oracle/> (дата обращения 10.12.2021).
16. Установка Ubuntu 20.04 на бесплатный VPS сервер от Oracle. [Электронный ресурс]. – Режим доступа: [https://pikabu.ru/story/ustanovka\\_ubuntu\\_2004\\_na\\_besplatniy\\_vps\\_server\\_ot\\_oracle\\_prodlzhenie\\_posta\\_registriruem\\_always\\_free\\_vps\\_server\\_dlya\\_nuzhd\\_u\\_mnogo\\_doma\\_7972849](https://pikabu.ru/story/ustanovka_ubuntu_2004_na_besplatniy_vps_server_ot_oracle_prodlzhenie_posta_registriruem_always_free_vps_server_dlya_nuzhd_u_mnogo_doma_7972849) (дата обращения 10.12.2021).
17. Классы устройств. [Электронный ресурс]. – Режим доступа: <https://www.thethingsnetwork.org/docs/lorawan/classes/> (дата обращения 02.12.2021).

18. Активация устройств. [Электронный ресурс]. – Режим доступа: <https://www.thethingsnetwork.org/docs/lorawan/end-device-activation/> (дата обращения 02.12.2021).
19. Скорость передачи данных. [Электронный ресурс]. – Режим доступа: <https://www.thethingsnetwork.org/docs/lorawan/limitations/> (дата обращения 02.12.2021).
20. RAK811 Lora AT Command User Guide V1.4. [Электронный ресурс]. – Режим доступа: <https://www.compel.ru/item-pdf/7008b5e14cfb8d82cebabdb784d57018/pn/rak~rak811.pdf> (дата обращения 16.12.2021).
21. NUCLEO-F103RB - STM32 Nucleo-64 development board with STM32F103RB MCU. [Электронный ресурс]. – Режим доступа: [https://www.st.com/content/st\\_com/en/products/evaluation-tools/product-evaluation-tools/mcu-mpu-eval-tools/stm32-mcu-mpu-eval-tools/stm32-nucleo-boards/nucleo-f103rb.html](https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-mpu-eval-tools/stm32-mcu-mpu-eval-tools/stm32-nucleo-boards/nucleo-f103rb.html) (дата обращения 25.09.2021).
22. IT-academy Samsung. [Электронный ресурс]. – Режим доступа: <https://myitacademy.ru/edu/mod/book/view.php?id=520> (дата обращения 20.09.2021).
23. UnwiredDevices/Umdk-thp. [Электронный ресурс]. – Режим доступа: <https://github.com/unwireddevices/umdk-boards> (дата обращения 10.10.2021).
24. UMDK-THP. [Электронный ресурс]. – Режим доступа: <https://www.unwireddevices.com/docs/umdk/umdk-thp/> (дата обращения 12.10.2021).
25. СанПиН 2.4.3648-20 – О требованиях к микроклимату в образовательных учреждениях. [Электронный ресурс]. – Режим доступа: [https://docs.yandex.ru/docs/view?tm=1639898117&tld=ru&lang=ru&name=SP\\_2.4.3648-20.pdf&text=%D1%81%D0%BF%202.4.3648-20%20%D1%81%D0%BA%D0%B0%D1%87%D0%B0%D1%82%D1%8C&url=https%3A%2F%2Flyceum.urfu.ru%2Ffileadmin%2Fuser\\_upload%2Fdocs%2FSP\\_2.4.3648-20.pdf&lr=67&mime=pdf&110n=ru&sign=bd0b729475e6b8a283d3c0530ca182df&keyno=0&nos](https://docs.yandex.ru/docs/view?tm=1639898117&tld=ru&lang=ru&name=SP_2.4.3648-20.pdf&text=%D1%81%D0%BF%202.4.3648-20%20%D1%81%D0%BA%D0%B0%D1%87%D0%B0%D1%82%D1%8C&url=https%3A%2F%2Flyceum.urfu.ru%2Ffileadmin%2Fuser_upload%2Fdocs%2FSP_2.4.3648-20.pdf&lr=67&mime=pdf&110n=ru&sign=bd0b729475e6b8a283d3c0530ca182df&keyno=0&nos)

[w=1&serpParams=tm%3D1639898117%26tld%3Dru%26lang%3Dru%26name%3DSP\\_2.4.3648-20.pdf%26text%3D%25D1%2581%25D0%25BF%2B2.4.3648-20%2B%25D1%2581%25D0%25BA%25D0%25B0%25D1%2587%25D0%25B0%25D1%2582%25D1%258C%26url%3Dhttps%253A%2F%2Flyceum.urfu.ru%2Ffileadmin%2Fuser\\_upload%2Fdocs%2FSP\\_2.4.3648-20.pdf%26lr%3D67%26mime%3Dpdf%26l10n%3Dru%26sign%3Dbd0b729475e6b8a283d3c0530ca182df%26keyno%3D0%26nosw%3D1](https://flyceum.urfu.ru/fileadmin/user_upload/FSP_2.4.3648-20.pdf?text%3D%25D1%2581%25D0%25BF%2B2.4.3648-20%2B%25D1%2581%25D0%25BA%25D0%25B0%25D1%2587%25D0%25B0%25D1%2582%25D1%258C%26url%3Dhttps%253A%2F%2Flyceum.urfu.ru%2Ffileadmin%2Fuser_upload%2Fdocs%2FSP_2.4.3648-20.pdf%26lr%3D67%26mime%3Dpdf%26l10n%3Dru%26sign%3Dbd0b729475e6b8a283d3c0530ca182df%26keyno%3D0%26nosw%3D1) (дата обращения 05.12.2021).

## Приложение А

(обязательное)

## Код прошивки микроконтроллера

```
#include "mbed.h"
#include "BME280.h"
#include "serial_mac_api.h"
#include <string>
#include <iostream>
#include <cstdlib>
#include <charconv>

#define MAX_DIGITS 10

RawSerial pc(USBTX, USBRX);
RawSerial dev(D8, D2);
BME280 sensor(I2C_SDA, I2C_SCL);

void dev_recv()
{
    while(dev.readable()) {
        pc.putc(dev.getc());
    }
}

void pc_recv()
{
    while(pc.readable()) {
        dev.putc(pc.getc());
    }
}

int main()
{
```

```

struct data
{
    char Warning;
    char Temperature[MAX_DIGITS + sizeof(char)];
    char Pressure[MAX_DIGITS + sizeof(char)];
    char Humidity[MAX_DIGITS + sizeof(char)];
} data;

while(1) {
    std::to_chars(data.Temperature, data.Temperature + MAX_DIGITS,
sensor.getTemperature());

    std::to_chars(data.Pressure, data.Pressure + MAX_DIGITS, sensor.getPressure());
    std::to_chars(data.Humidity, data.Humidity + MAX_DIGITS, sensor.getHumidity());


    //pc.printf("%s\n", data.Temperature);
    //pc.printf("%s\n", data.Pressure);
    //pc.printf("%s\n", data.Humidity);
    //pc.printf("\n\n");
    //string comand = "at+send=lora:1:";
    pc.baud(115200);
    dev.baud(115200);
    //for(int i = 0; i < comand.length(); i++)
    //{
        // dev.putc(comand[i]);
    //}

    if (sensor.getTemperature() < 18 || sensor.getTemperature() > 24 || sensor.getHumidity() <
40 || sensor.getHumidity() > 60)
    {
        pc.attach(&pc_recv, Serial::RxIrq);
        dev.attach(&dev_recv, Serial::RxIrq);

        data.Warning = '1';
        wait(18000);
        sleep();
    }
}

```

```
    }  
    else  
    {  
        pc.attach(&pc_rcv, Serial::RxIrq);  
        dev.attach(&dev_rcv, Serial::RxIrq);  
  
        data.Warning = '0';  
        wait(108000);  
        sleep();  
    }  
}  
}
```