



Техническо училище “Електронни системи”
към ТУ-София

Документация

**Тема: Система за записване на атмосферните
условия**

Проектант:
Алекс Огнянов №1

Научен ръководител:
Росен Витанов

2023 г.

УВОД

В нашето забързано ежедневие е важно да внимаваме за качеството на околната среда. Прекарване на много време в среда с лошо качество на въздуха може да доведе до сериозни заболявания. За тази цел помагат станциите, които измерват различни елементи от въздуха, който ни заобикаля. Едни от основните проблеми на тези станции са цената и предоставените функционалности. Някои станции нямат безжична връзка със сървъри или цената им е прекалено висока.

All-in-one-weather е иновативна система за записване и обработване на информация за околната среда. Целта ѝ е да реши всички проблеми на комерсиалните системи. Тя се състои от 2 устройства: сървър и клиент.

Сървърът се намира на място, което има Wi-Fi връзка и използва LoRa комуникация с клиента. Той отговаря за връзката с онлайн сървъра и качването на записаната информация.

Клиентът съдържа в себе си всички измервателни устройства и модули, нужни за измерване на основните характеристики на околната среда. Тъй като използва LoRa комуникация и слънчеви панели за захранване, може да бъде поставен както навън, така и вътре. При нестабилна LoRa връзка, информацията бива записвана на MicroSD карта и когато отново е налична стабилна връзка, записаните данни биват изпратени. Конкурентната му цена и онлайн интеграцията прави системата достъпна за организации, които имат нужда да използват много на брой измервателни устройства.

Благодарение на онлайн интеграцията си, устройството може да бъде използвано заедно с домакински смарт устройства, за да бъдат регулирани условията у дома автоматично, което би помогнало на хора с домашни любимци, които пътуват и не могат да вземат домашния си любimeц с тях.

I. Съществуващи метеорологични станции и съпоставка с All-in-one-weather

1. Atmos 41

Atmos 41 метеорологичната станция е разработена от METER. В себе си включва 12 сензора за измерване на метеорологични характеристики. Станцията може да бъде лесно свързана със ZL6 записвач на данни чрез кабел, който записва данните, за да може данните да бъдат интегрирани в приложения и програми, които допълнително да извършват операции върху тях. Atmos 41 притежава следните функционалности като всеки клиент може да избира сам кои функционалности да бъдат включени:

- Измерване на температура и относителна влажност
- Измерване на барометрично налягане и налягане на парата
- Измерване на скорост и посока на ветрове или пориви като има способността да разпознава вятър от 0,01 m/s
- Измерване на слънчева радиация
- Измерване на количество дъжд, чрез плувиометър с филтър против запушване
- Измерване на мълнии и разстояние на мълниите
- Засичане на необичайно движение на станцията, чрез акселерометър

Цената на станцията може да варира от 500 до 1000 долара, в зависимост от включените сензори и функционалности.



фиг 1.1.1 Atmos 41 метеорологична станция

2. Weathercom

Weathercom метеорологичната станция е разработена от Oizom. Oizom предоставят облачни услуги за лесен достъп до данните. Може да бъде захранвана както от соларни панели, така и от електрическата мрежа. Има много начини за връзка с облачните услуги, като например:

- GSM
- LTE
- NB-IoT
- Wi-Fi
- Ethernet
- SigFox
- LoRa
- ModBus

Устройството предоставя много функционалности, които могат да бъдат избириани от клиента, като например:

- Измерване на количество дъжд
- Измерване на количество светлина, достигната станцията и UV лъчение
- Измерване на посока и скорост на ветрове и пориви
- Измерване на видимост при мъгла
- Измерване на състояние на пътна настилка
- Измерване на влажност на почвата
- Измерване на температура, влажност и налягане
- Измерване на ниво на вода при наводнение



Фиг 1.2.1 Weathercom метеорологична станция

1) Weathercom Lite

Версия на Weathercom станцията, предназначена за общо ползване. Включва следните функционалности:

- Измерване на скорост на вятъра
- Измерване на посока на вятъра
- Измерване на количество дъжд
- Измерване на количество светлина и UV лъчение
- Измерване на температура, влажност и налягане

Цената на станцията е €3,129.

2) Weathercom Smart

Версия на Weathercom станцията, предназначена главно за отглеждане на земеделски култури. Включва следните функционалности:

- Измерване на количество дъжд
- Измерване на количество светлина, достигнала станцията и UV лъчение
- Измерване на посока и скорост на ветрове и пориви
- Измерване на температура, влажност и налягане
- Измерване на влажност на почвата

Цената на станцията е €3,380.

3) Weathercom Pro

Версия на Weathercom станцията, предназначена главно за следене на състояние на пътната настилка. Включва следните функционалности:

- Измерване на количество дъжд
- Измерване на количество светлина, достигнала станцията и UV лъчение
- Измерване на посока и скорост на ветрове и пориви

- Измерване на температура, влажност и налягане
- Измерване на видимост при мъгла
- Измерване на състояние на пътна настилка

Цената на станцията е €11,672.

3. Съпоставка между All-in-one-weather и пазарните конкуренти

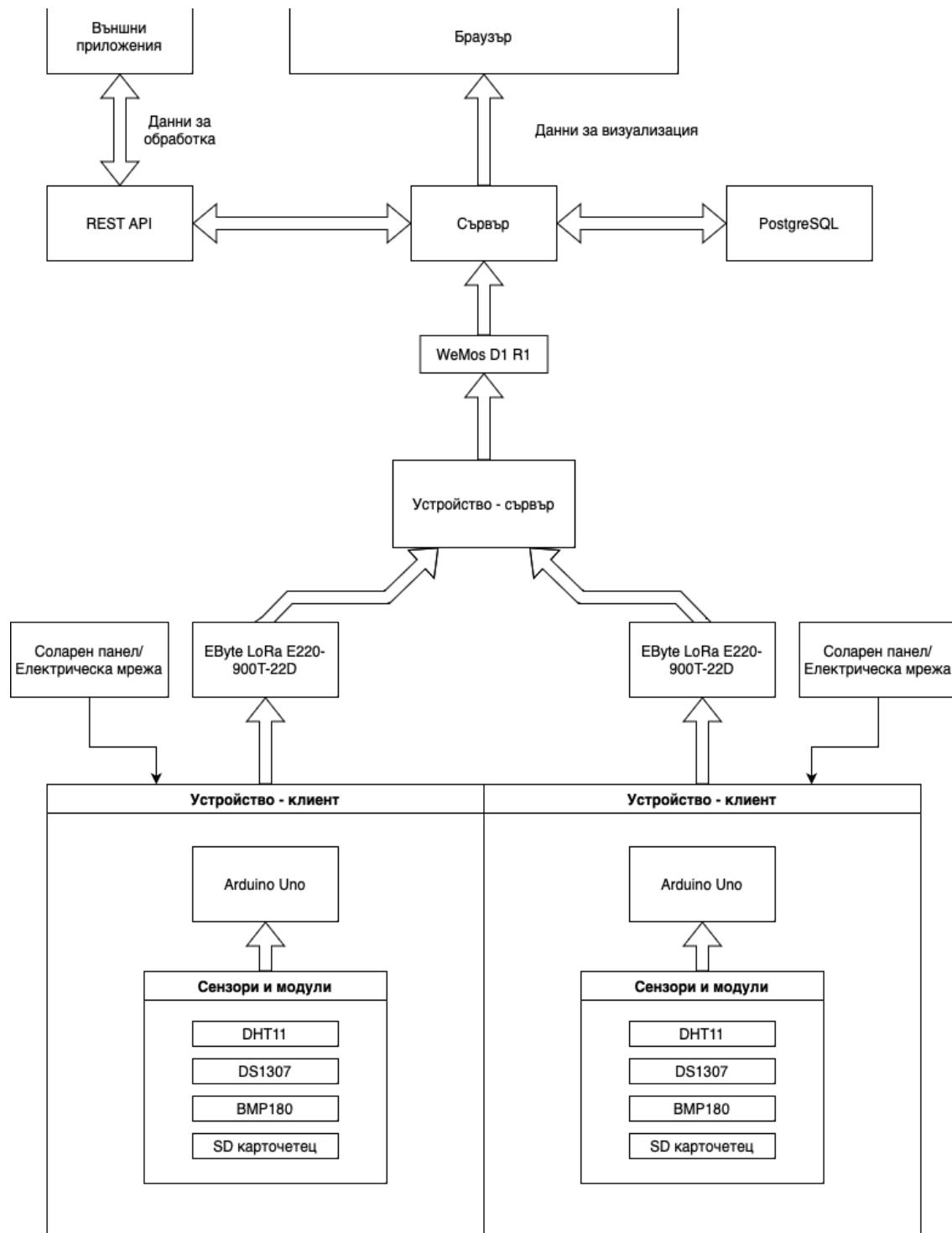
Всяка от изборените станции има своите предимства и недостатъци. Главният проблем в тях е цената и свързаността. Weathercom станциите предоставят голяма точност и много методи за лесна работа, но цената им е прекалено висока. Един от главните проблеми на Atmos 41 станцията, е че единственият начин за записване на данните е посредством кабел, което прави поставянето му на отдалечени места почти невъзможно. Друг проблем на Atmos 41 станцията е, че трябва да бъде закупен допълнителен записвач на данни и периодично да бъде добавяна памет, което допълнително осъществява проекта. За разлика от горепосочените станции, All-in-one-weather станцията създава LoRa връзка между клиента и сървъра, използва слънчев панел за захранване, но има и възможността да бъде захранвана от електрическата мрежа. Тя може да измерва температура, налягане и влажност на въздуха и надморска височина. При нужда могат да бъдат добавени модули за измерване на количество дъжд, UV лъчение и скорост и посока на ветрове и пориви. При нестабилна връзка, данните биват записвани на MicroSD карта, за да могат да бъдат изпратени след възстановяване на LoRa връзката. Освен това цената на All-in-one-weather станцията е значително по-ниска, равнявайки се на около общо 200 лева за сървъра и клиента с всички горепосочени

допълнителни функционалности (цената не включва хостинг на уебсайт и база данни).

Станция	Цена	Точност	Функционалности
Atmos 41	898 - 1,797 лв.	Висока	Скорост и посока на вятъра, температура на въздуха, относителна влажност, барометрично налягане, налягане на парите, UV лъчение, валежи и мълнии
Weathercom Lite	6118 лв.	Средна	Скорост и посока на вятъра, дъжд, светлина, UV лъчение, температура, влажност, налягане
Weathercom Smart	6608 лв.	Средна	Скорост и посока на вятъра, валежи, светлина, UV лъчение, температура, влажност, налягане, влажност на почвата
Weathercom Pro	22822 лв.	Висока	Скорост и посока на вятъра, валежи, светлина, UV лъчение, температура, влажност, налягане, видимост на пътя, състояние на пътната настилка
All-in-one-weather	200 лв.	Средна	Температура, налягане и влажност на въздуха, надморска височина

II. Реализация

1. Блок схема



Фиг. 2.1.1 Блок схема на модули и връзки за комуникация

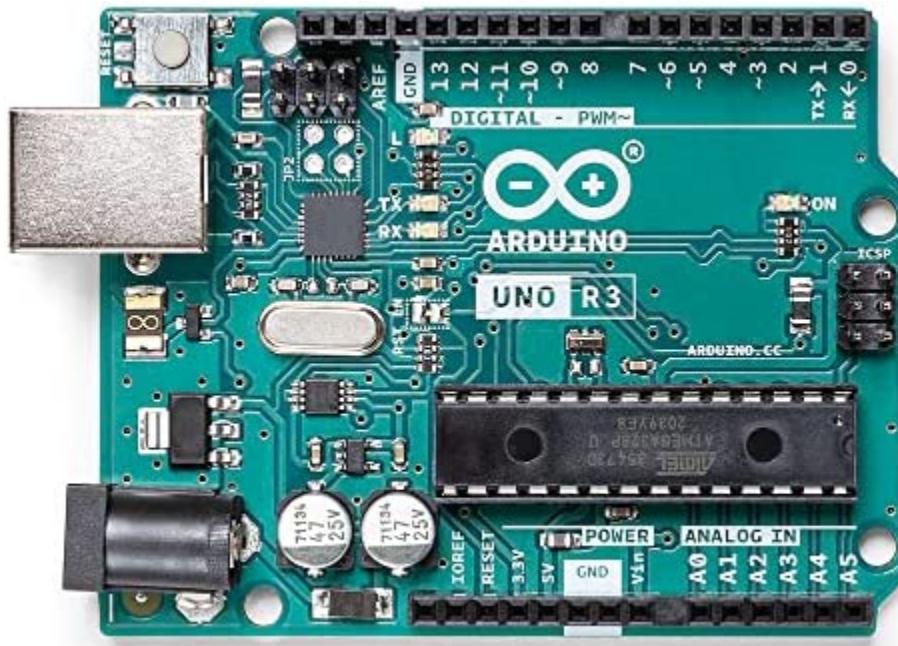
2.Начин на работа

All-in-one-weather станцията е решение на всички текущи проблеми с метеорологичните станции като ценова достъпност и точност на измерващите сензори. Тя използва сензори за температура, налягане и влажност на въздуха, надморска височина. Устройствата работят на база на итерации. Една итерация започва, когато Arduino Uno в устройството-клиент извлича данните от сензорите, обработва ги и изпраща към устройството-сървър посредством LoRa връзка. Когато устройството-сървър получи информацията, тя бива допълнително обработена и посредством Wi-Fi мрежата бива изпращана на NestJS сървър, който допълнително обработва данните и ги съхранява в PostgreSQL база от данни. При нужда данните могат да бъдат изпращани и обработвани от външни апликации посредством заявки към сървъра. Копие на данните се записват локално на MicroSD, за да няма загуба на данни при нестабилна Wi-Fi връзка или смущения в сигнала. Итерацията е завършена. Итерациите се случват на всеки 60 минути.

III. Електрическа схема и компоненти

1. Компоненти на устройството-клиент

- **Arduino Uno**

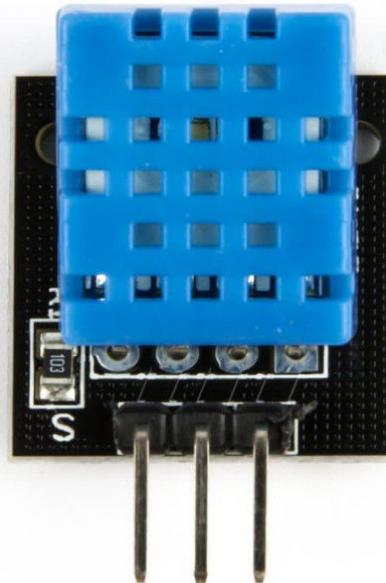


Фиг. 3.1.1 Arduino Uno микроконтролер

1. Характеристика

- 1.1. Работна честота:** 16 MHz.
- 1.2. Работно напрежение:** 5V.
- 1.3. Захранващо напрежение:** 7-12V(през VIN и буксата за захранване), 5V (USB).
- 1.4. Брой шифтове:**
 - 1.4.1. Цифрови:** 14 (6 поддържат ШИМ).
 - 1.4.2. Аналогови:** 6.
- 1.5. Памет:**
 - 1.5.1. Flash:** 32 KB.
 - 1.5.2. EEPROM:** 1 KB.

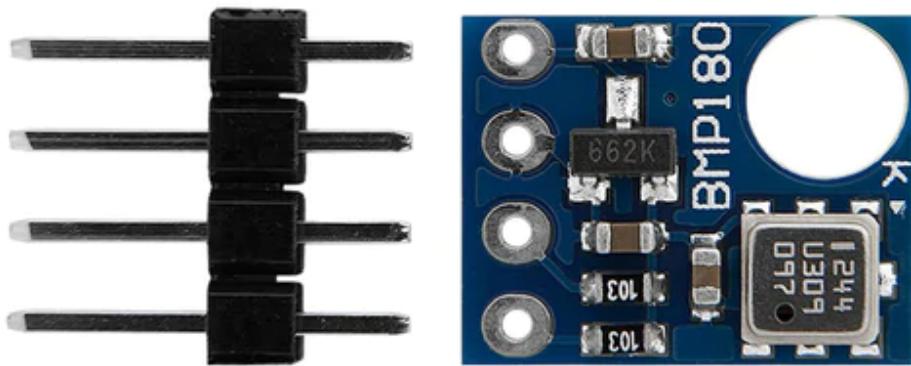
- **DHT11 Breakout**



Фиг 3.1.2 DHT11 сензор

1. Характеристика
 - 1.1. Използва се за измервания на **температура и влажност**.
 - 1.2. **Работно напрежение:** 3-5 V.
 - 1.3. **Максимална консумация:** 2.5 mA.
 - 1.4. **Влажностен обхват:** 20 - 80 %.
 - 1.5. **Температурен обхват:** 0 - 50 °C.
 - 1.6. **Работна честота:** 1 Hz.
 - 1.7. **Интерфейс:** цифров.

- **BMP180 (GY-68)**

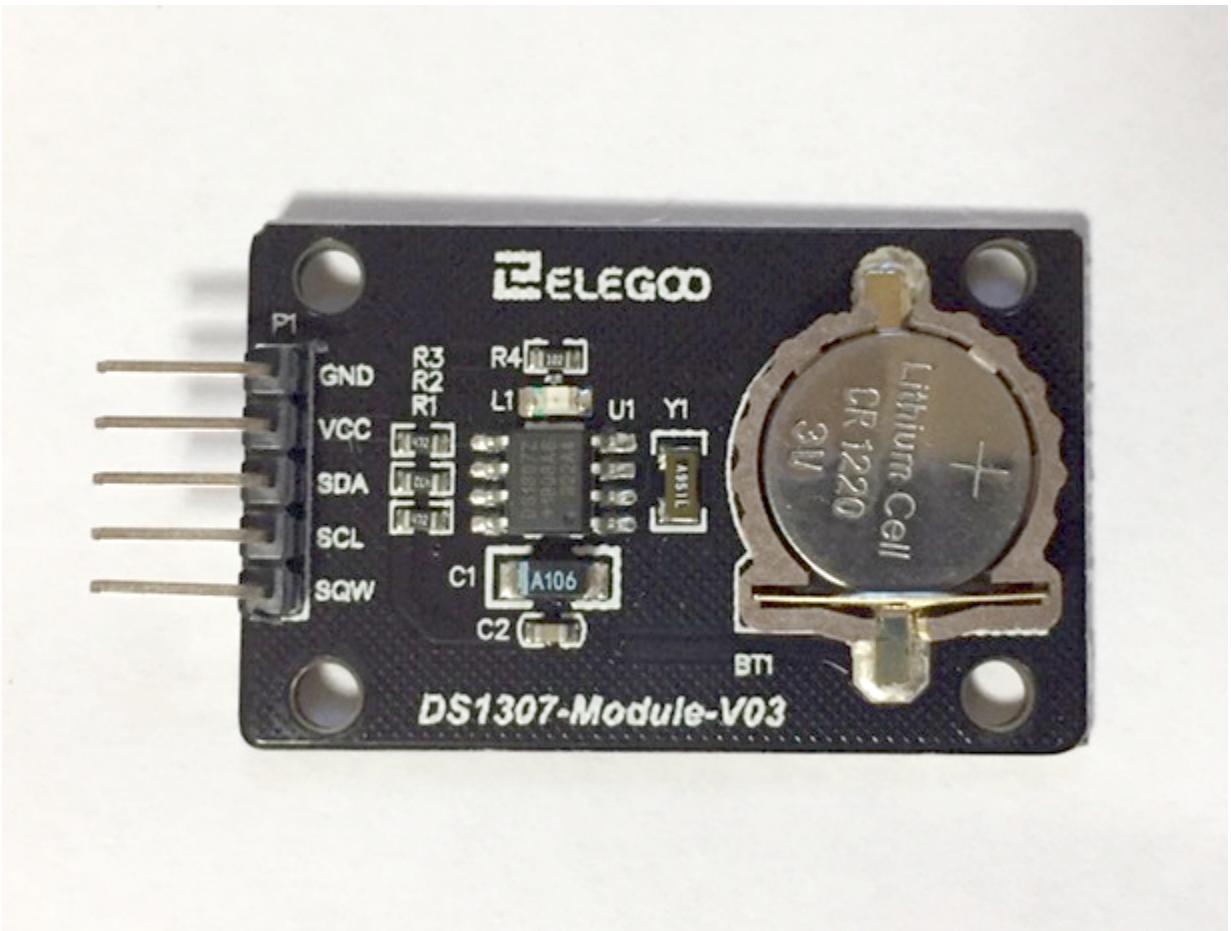


Фиг 3.1.3 BMP180 сензор

1. Характеристика

- 1.1. Използва се за измерване на **налягане**.
- 1.2. **Работно напрежение:** 3.3V.
- 1.3. **Консумация:** 65 - 100 mA.
- 1.4. **Обхват на налягане:** 300 - 1100 hPa.
- 1.5. **Температурен обхват:** -40°C - 85°C.
- 1.6. **Резолюция на налягането:** 0.02 - 0.06hPa.
- 1.7. **Интерфейс:** I2C.

- DS1307 (RTC)

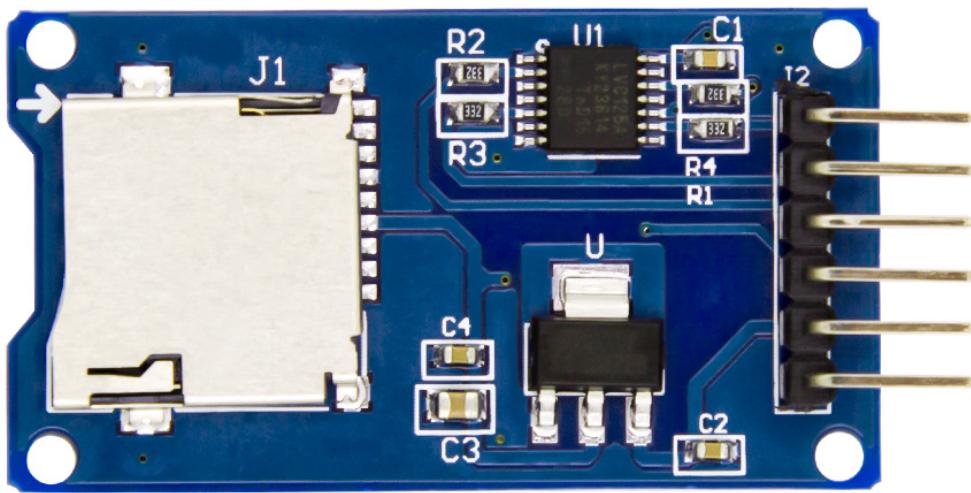


Фиг 3.1.4 DS1307 модул (RTC)

1. Характеристика

- 1.1. Използва се като **часовник**.
- 1.2. **Работно напрежение:** 5 V.
- 1.3. **Консумация:**
 - 1.3.1. при захранване само от батерия: <500nA.
 - 1.3.2. при предаване на информация: <100mA.
- 1.4. **Интерфейс:** I2C.

- MicroSD Card Reader

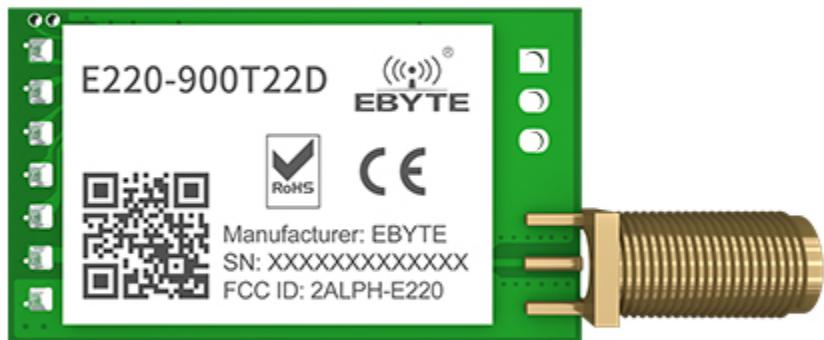


Фиг 3.1.5 Модул за работа с MicroSD карта

1. Характеристика

- 1.1. Използва се за **записване на информация**, в случай, че LoRa връзката е нестабилна
- 1.2. **Работно напрежение:** 5 V.
- 1.3. **Консумация:** 5 - 40 mA.
- 1.4. **Интерфейс:** I2C.

- EByte LoRa E220-900T-22D



Фиг. 3.1.6 EByte LoRa E220-900T-22D

1. Характеристика

- 1.1. Използва се за **безжична връзка** между устройствата.
- 1.2. **Работно напрежение:** 3.3 - 5V
- 1.3. **Работна температура:** -30C до +85C.
- 1.4. **Честотен диапазон:** 850.125 ~ 930.125MHz.
- 1.5. **Програмируема скорост:** до 62 kbps.
- 1.6. **Изходна мощност:** + 22 dBm.
- 1.7. **Интерфейс:** UART.

2. Компоненти на устройството-сървър

- Arduino Uno

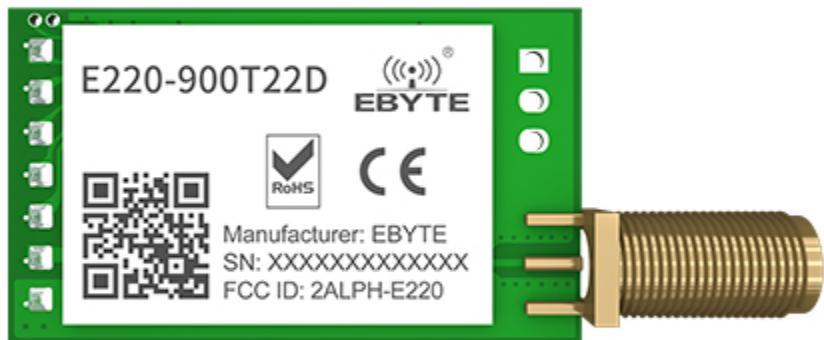


Фиг. 3.2.1 Arduino Uno микроконтролер

1. Характеристика

- 1.1. Работна честота: 16 MHz.
- 1.2. Работно напрежение: 5V.
- 1.3. Захранващо напрежение: 7-12V(през VIN и буксата за захранване), 5V (USB).
- 1.4. Брой шифтове:
 - 1.4.1. Цифрови: 14 (6 поддържат ШИМ).
 - 1.4.2. Аналогови: 6.
- 1.5. Памет:
 - 1.5.1. Flash: 32 KB.
 - 1.5.2. EEPROM: 1 KB.

- EByte LoRa E220-900T-22D

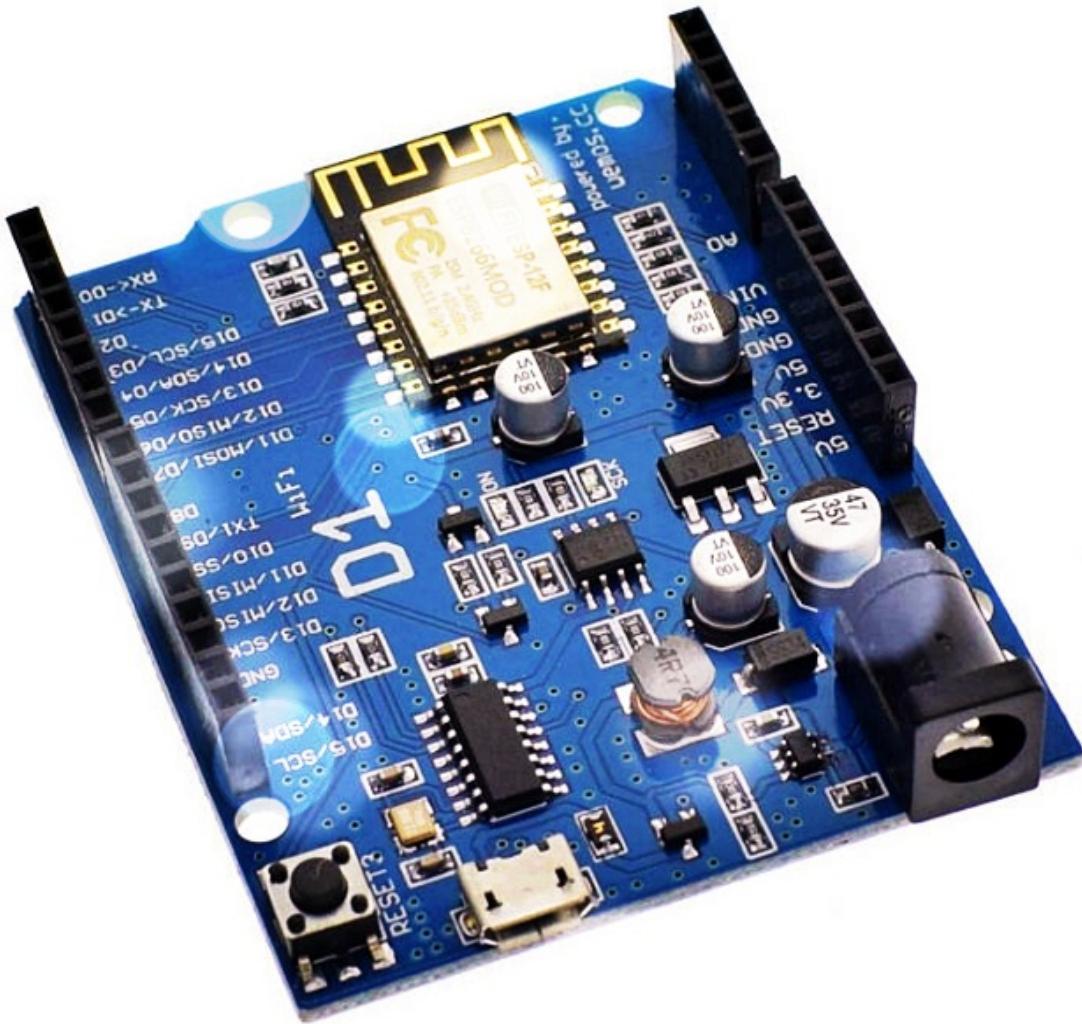


Фиг. 3.2.2 EByte LoRa E220-900T-22D

1. Характеристика

- 1.1. Използва се за **безжична връзка** между устройствата.
- 1.2. **Работно напрежение:** 3.3 - 5V
- 1.3. **Работна температура:** -30C до +85C.
- 1.4. **Честотен диапазон:** 850.125 ~ 930.125MHz.
- 1.5. **Програмируема скорост:** до 62 kbps.
- 1.6. **Изходна мощност:** + 22 dBm.
- 1.7. **Интерфейс:** UART.

- Wemos D1 R1



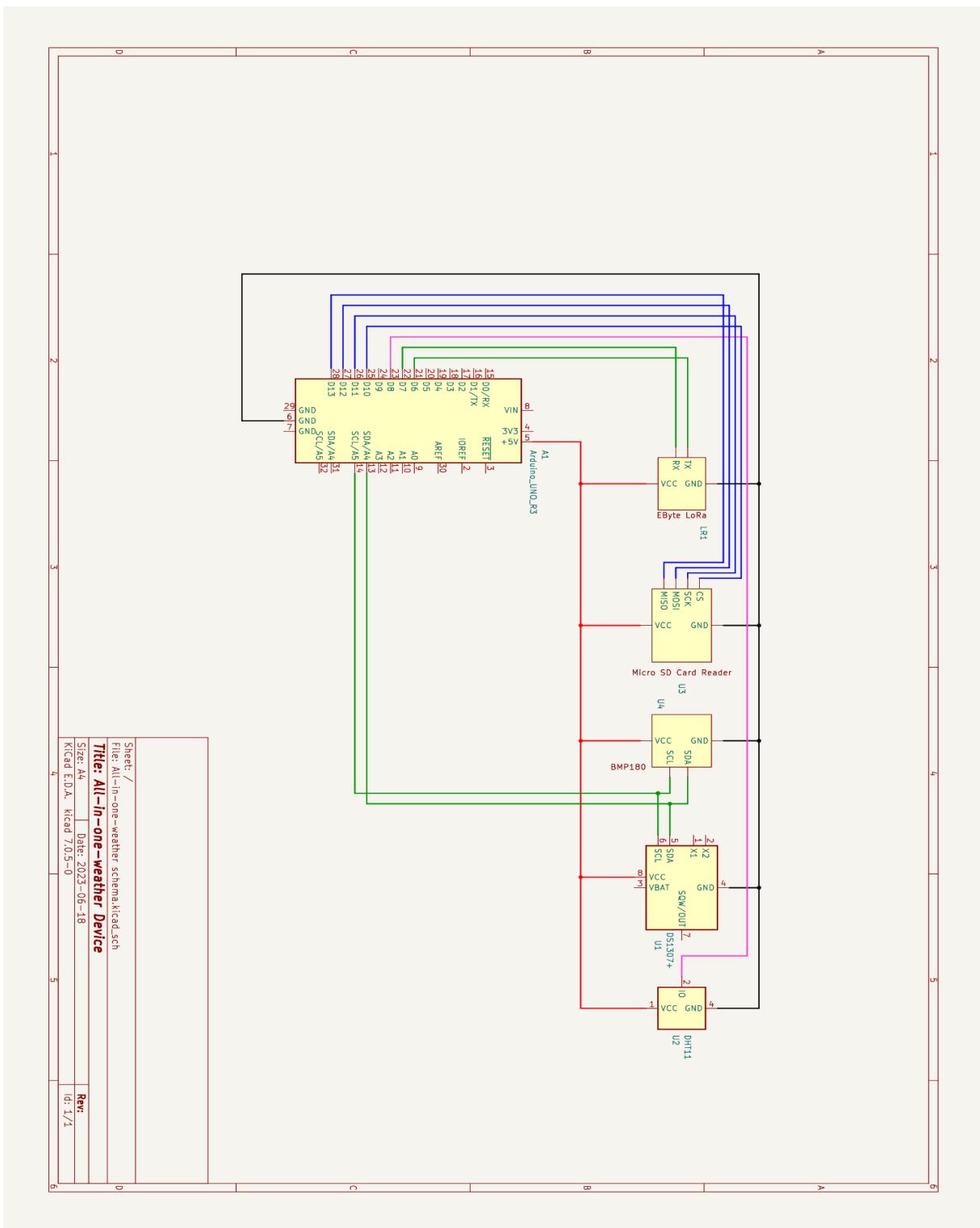
Фиг 3.2.3. Wemos D1 R1

1. Характеристика

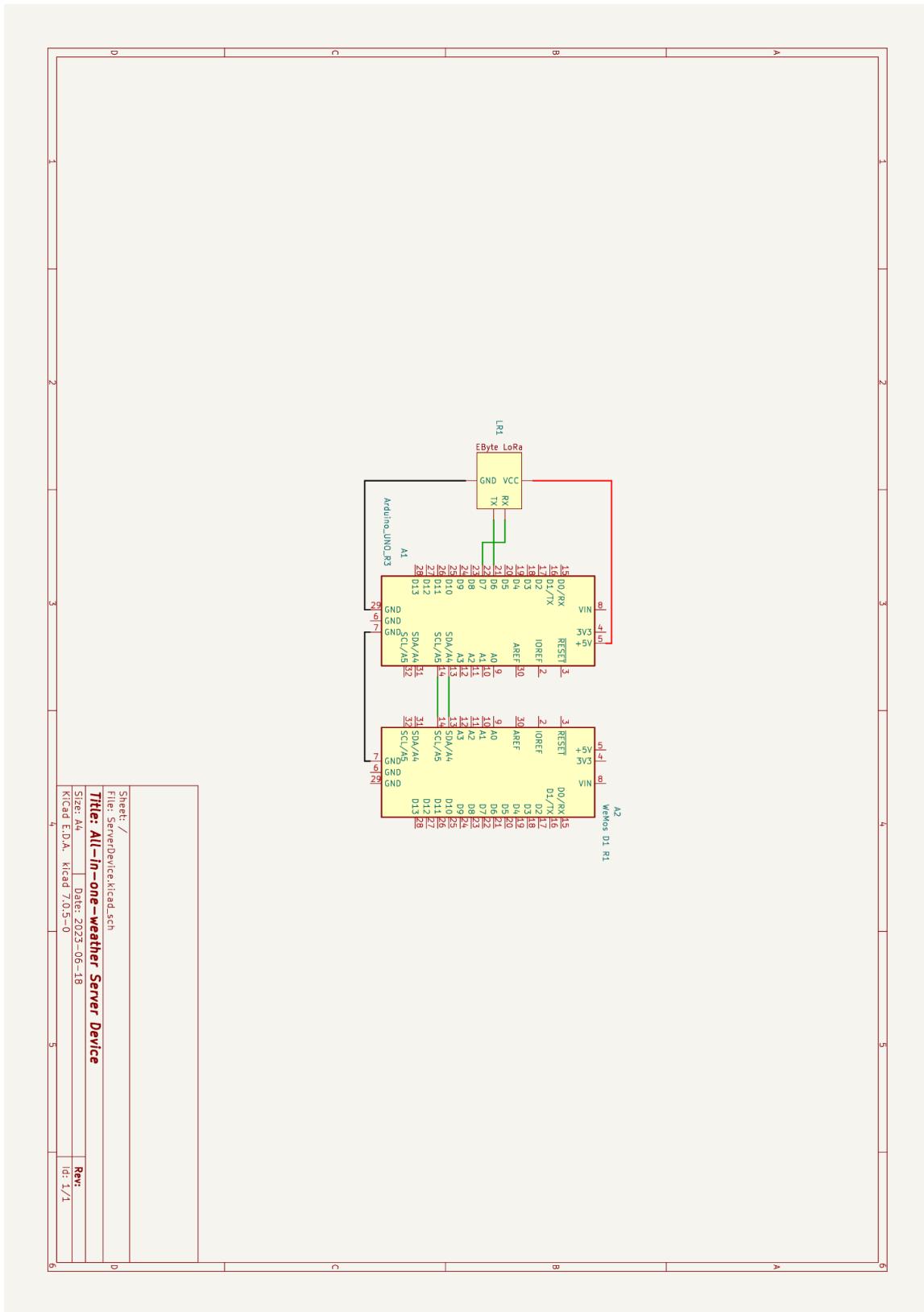
- 1.1. Използва се за **връзка** между онлайн сървъра и устройството-сървър.
- 1.2. **Работно напрежение:** 5 V.
- 1.3. **Wi-Fi честота:** 2.4 GHz.
- 1.4. **Интерфейс:** I2C.

3. Електрическа схема

• Устройство-клиент

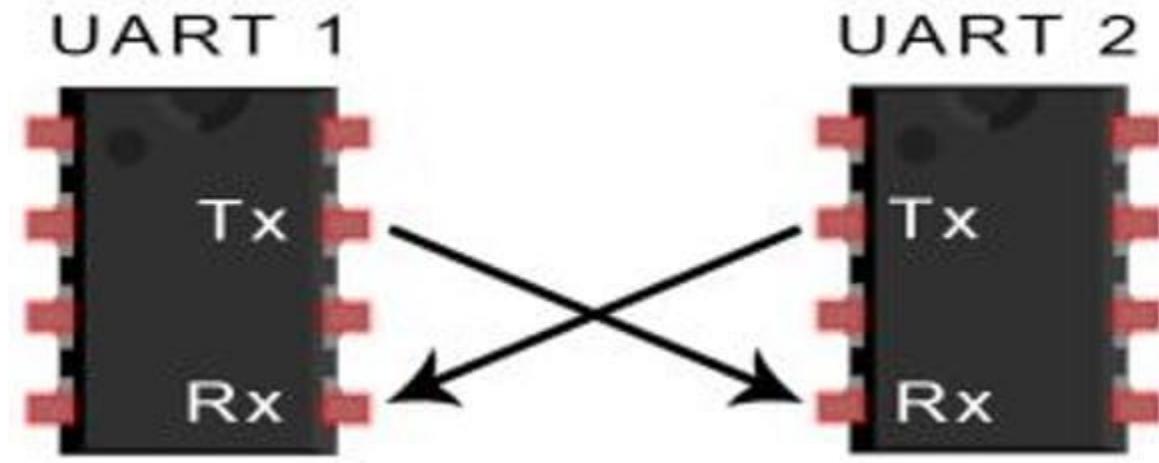


• Устройство-сървър



IV. Комуникации

1. UART (Universal Asynchronous Receiver-Transmitter)



Фиг 4.1. Схема на UART комуникация

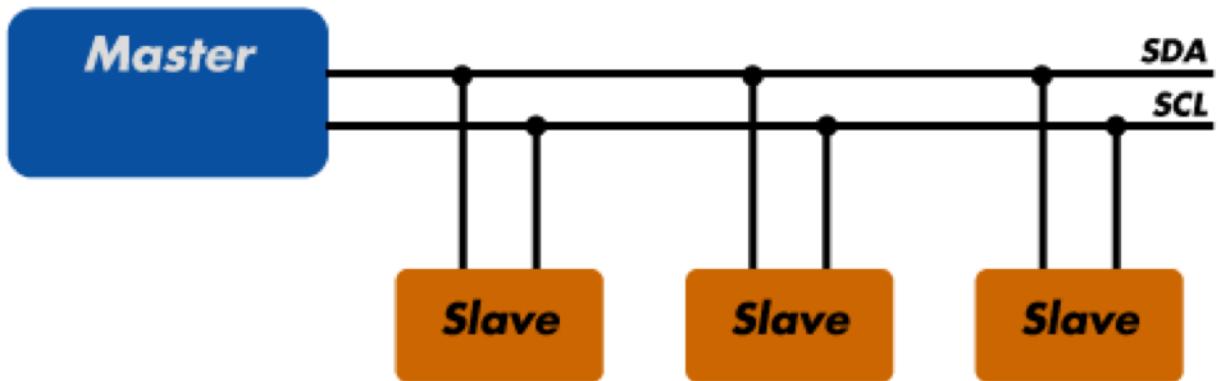
Използва се за серийна комуникация между 2 устройства. Използват се щифтовете **RX** (Receive) и **TX** (Transmit), които се свързват на кръст (*Фиг 4.1.*).

1. Характеристики:

- 1.1. Няма нужда от задаване на честота (clock).
- 1.2. 3 режима на работа:
 - 1.2.1. **Simplex** - предава се информация само в една посока.
 - 1.2.2. **Half-duplex** - предава се информация в двете посоки, но не едновременно.
 - 1.2.3. **Full-duplex** - предава се информация в двете посоки едновременно.

UART комуникацията използва **baud rate** вместо зададена честота. За получаването на използваема информация и двете устройства трябва да работят на един и същ **baud rate**.

2. I2C (Inter-Integrated Circuit)



Фиг 4.2. Схема на I2C комуникация

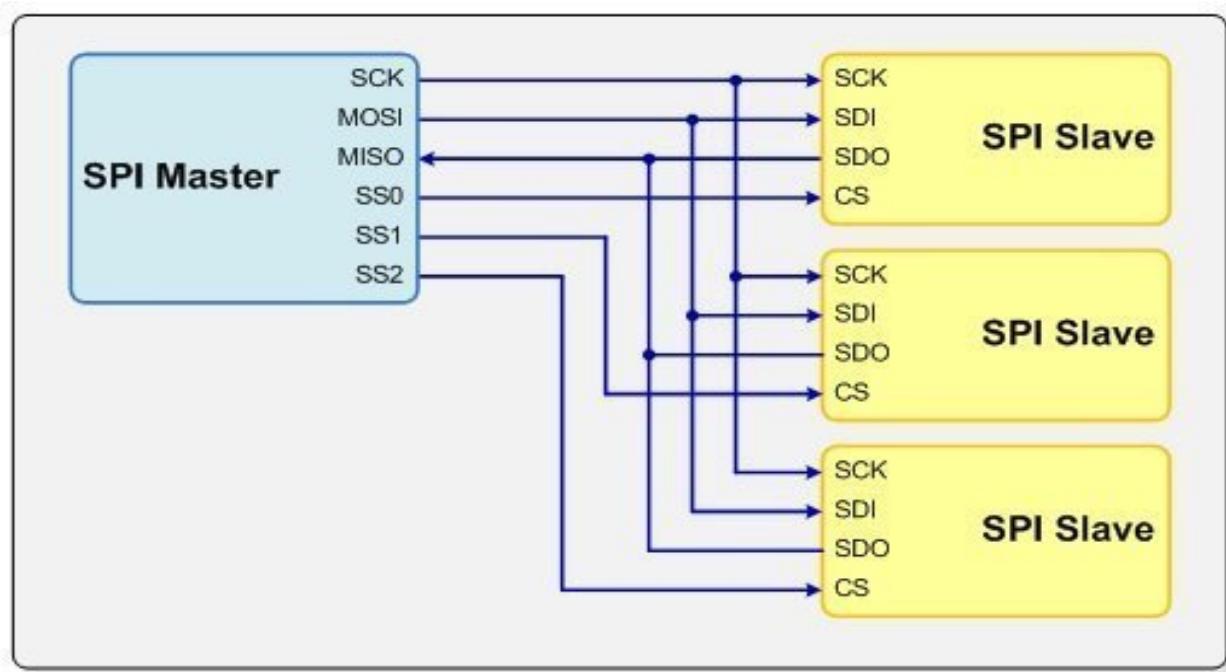
Използва се за свързване на 2 или повече устройства, които поддържат I2C комуникационния протокол. Използват се обозначенията **master** и **slave**. Освен това се използват **SCL** (Serial Clock) и **SDA** (Serial Data) щифтове (Фиг 4.2.).

1. Характеристики:

- 1.1. Позволява свързването на повече от 2 устройства.
- 1.2. Схемата става по-сложна с всеки нов master/slave.
- 1.3. При високи скорости дължината на проводниците има лимит.

Комуникацията се осъществява чрез **half-duplex**, тоест информацията може да протича само в посока от **slave** към **master**.

3. SPI (Serial Peripheral Interface)



Фиг 4.3. Схема на SPI комуникацията

SPI използва **MISO** (Master In Slave Out), **MOSI** (Master Out Slave In), **SCK** (Clock) и **SS** (Slave Select) щифтове (Фиг 4.3.).

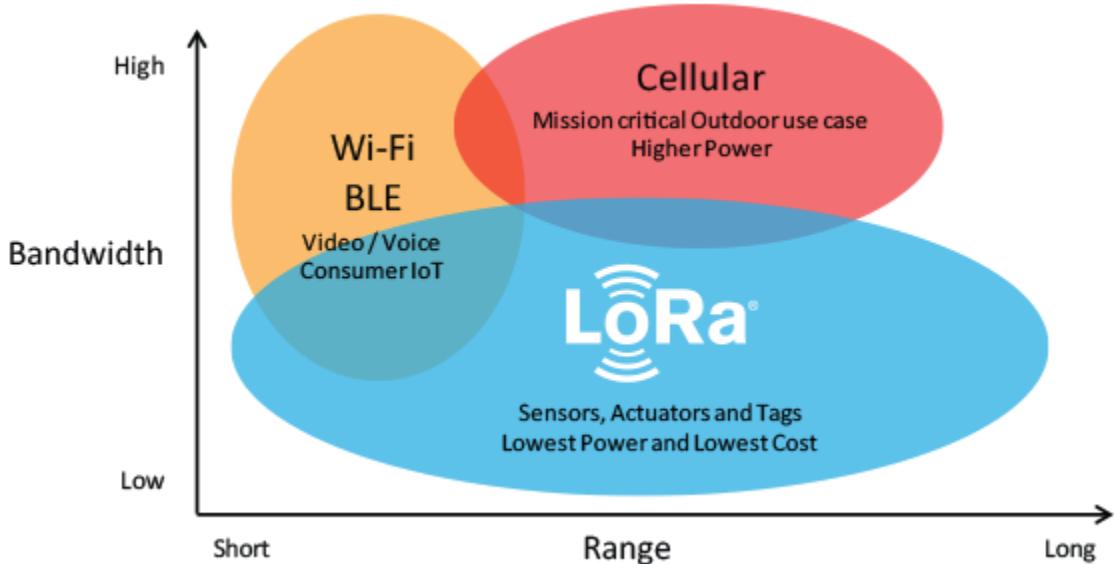
Комуникацията се осъществява асинхронно. Комуникацията се осъществява във **Full-duplex** режим, затова едновременно може да се приема и предава информация.

Предимства:

1. Позволява свързването на неограничен брой устройства.
2. Изискват се голямо количество проводници дори и за един slave
3. Всеки slave изиска собствен SS щифт, което допълнително увеличава количеството проводници на схемата

Може да има само един master

4. LoRa (Long Range)



Фиг. 4.4. Схема на честотите на различните комуникации

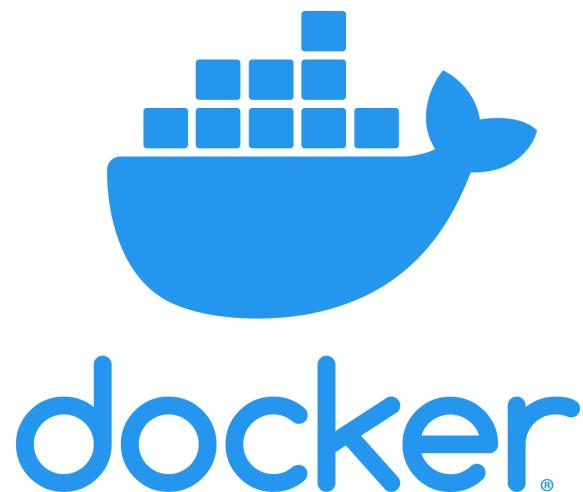
LoRa използва безлицензни суб гигахерцови радиочестотни ленти **EU868 (863-870/873 MHz)** в Европа; **AU915/AS923-1 (915-928 MHz)** в Южна Америка; **US915 (902-928 MHz)** в Северна Америка; **IN865 (865-867 MHz)** в Индия; и **AS923 (915-928 MHz)** в Азия. LoRa позволява предавания на големи разстояния с ниска консумация на енергия. Технологията покрива физическия слой, докато други технологии и протоколи, като LoRaWAN (Long Range Wide Area Network), покриват горните слоеве. Тя може да постигне скорост на предаване на данни между 0,3 kbit/s и 27 kbit/s, в зависимост от коефициента на разпространение.

1. Предимства
 - 1.1. Дълъг обхват
 - 1.2. Ниска консумация на енергия
 - 1.3. Икономически изгодна

IV. Сървър

1. Използвани технологии

1.1. Docker



Фиг. 4.1. Docker

Docker е платформа с отворен код, която позволява на разработчиците да автоматизират инсталациейта и управлението на приложения в леки, изолирани контейнери. Тези контейнери осигуряват последователна и възпроизвеждаема среда за работа на приложенията, независимо от основната операционна система. Docker използва технологията за контейнеризация, за да пакетира приложенията и техните зависимости в самостоятелни единици, което позволява лесна преносимост и мащабируемост. Контейнерите, създадени с Docker, са изолирани един от друг, което гарантира, че приложенията работят последователно в различни среди. Docker също така предоставя солиден набор от инструменти и функции за изграждане, разпространение и управление на контейнеризирани приложения, което го прави популярен избор за разработчици и системни администратори.

1.2. PostgreSQL



Фиг. 4.2. PostgreSQL

PostgreSQL е система за управление на релационни бази данни с отворен код, известна със своята надеждност и гъвкавост. Тя следва модела клиент-сървър, като позволява на множество клиенти да се свързват към един сървър. Заявките се обработват от процес на сървъра, който ги анализира и оптимизира за ефективно изпълнение. PostgreSQL поддържа управление на транзакциите, като осигурява последователност на данните и контрол на паралелността. Той предлага широк набор от типове данни, разширени възможности за заявки и разширяемост чрез системата си за разширения, което го прави подходящ за различни приложения - от малки до корпоративни системи. Инстанция на PostgreSQL бива поддържана от Docker контейнер.

1.3. Node.js



Фиг. 4.3. Node.js

Node.js е мощна среда за изпълнение на JavaScript от страна на сървър с отворен код. Тя позволява на разработчиците да изпълняват JavaScript код извън уеб браузъра, като им дава възможност да създават и изпълняват приложения. Node.js използва неблокиращ модел на входно-изходни операции, управляем от събития, което го прави изключително ефективен и способен да обработва голям брой заявки едновременно. Той се предлага с богат набор от вградени модули и обширна екосистема от външни библиотеки, което го прави подходящ за широк спектър от приложения. Като цяло Node.js направи революция в уеб разработката, като предостави мащабируема и ефективна платформа за изграждане на бързи и отзивчиви мрежови приложения.

1.4. Express



Фиг. 4.4. Express

Express е минималистична, гъвкава и бърза платформа за уеб приложения за Node.js. Тя предоставя набор от надеждни функции и инструменти за изграждане на уеб приложения и API. Express опростява процеса на обработка на HTTP заявки, маршрутизиране, управление на междинни програми и генериране на отговори. Благодарение на олекотения си дизайн и модулния подход, разработчиците могат лесно да персонализират и разширяват Express, за да отговори на техните специфични нужди. Той има жизнена екосистема с широк набор от наличен междинен софтуер и приставки, което го прави популярен избор за изграждане на уеб приложения в Node.js. Като цяло Express предлага рационализиран и ефективен начин за създаване на сървърни приложения в JavaScript.

1.5. NestJS



Фиг. 4.5. NestJS

NestJS е мощна и мащабируема рамка на Node.js за изграждане на приложения от страна на сървъра. Тя е изградена на базата на Express и осигурява модулен и структуриран подход към разработването на приложения. NestJS поддържа TypeScript, което позволява на разработчиците да пишат силно типизиран код и да се възползват от неговите статично типизиране и обектно-ориентирани функции. Той следва архитектурния модел на Angular, което го прави познат на разработчиците, които вече работят с Angular приложения. NestJS предлага богата екосистема от модули и декоратори, което позволява на разработчиците ефективно да изграждат сложни приложения с лекота. Като цяло NestJS съчетава надеждността на Node.js със структурата и поддръжката на Angular, което го прави отличен избор за изграждане на мащабируеми и поддържани сървърни приложения.

1.6. Prisma



Фиг. 4.6. Prisma

Prisma е инструмент за управление на релации между обекти(ORM), който осигурява абстрактен слой за бази данни за изграждане на приложения, базирани на данни. Когато се използва с NestJS, Prisma опростява операциите с бази данни и подобрява производителността на разработчиците. Той предлага типово сигурен API за работа с бази данни, което позволява на разработчиците да пишат заявки и мутации към бази данни с помощта на TypeScript. Prisma генерира клиентска библиотека въз основа на схемата на базата данни, като осигурява автоматично попълване, проверка на типа и валидиране по време на компилация. Със своя мощен конструктор на заявки и възможности за миграция на схеми Prisma помага при управлението на промените в схемите на базите данни и при създаването на версии. Чрез интегрирането на Prisma с NestJS разработчиците могат лесно да се справят с взаимодействията с базата данни и да се съредоточат повече върху изграждането на бизнес логиката.

1.7. Handlebars



Фиг. 4.7. Handlebars

Handlebars.js е популярен механизъм за шаблониране за JavaScript, който позволява на разработчиците да създават динамични шаблони за HTML. Той предоставя прост и интуитивен синтаксис за вграждане на променливи, условия и цикли в HTML маркер. С помощта на Handlebars.js разработчиците могат да създават шаблони за многократна употреба и да ги визуализират с данни от различни източници, например JSON обекти или данни от страна на сървъра. Той следва подход без логика, като отделя слоя за представяне от логиката на приложението. Handlebars.js поддържа потребителски помощници и части, което позволява на разработчиците да разширяват функционалността му и да създават модулни шаблони.

1.8. ChartJS



Chart.js

Фиг. 4.8. Chart.js

Chart.js е популярна библиотека на JavaScript, която се използва за създаване на интерактивни и адаптивни диаграми и графики в уеб страници. Тя предоставя прост и интуитивен API, който позволява на разработчиците лесно да генерираят различни видове диаграми, като например стълбови, линейни, кръгови и други. Chart.js е изграден на базата на елемента HTML5 canvas, което позволява изобразяването на диаграми в среда на браузъра. Тя поддържа адаптивен дизайн, което позволява графиките автоматично да се адаптират и мащабират в зависимост от наличното пространство. С богат набор от опции за конфигуриране и обширна документация Chart.js е мощен инструмент за визуализация на данни в уеб приложения.

1.9. Swagger



Фиг. 4.9. Swagger

Swagger, известен също като OpenAPI Specification (OAS), е рамка и набор от инструменти, които позволяват проектирането, документирането и внедряването на RESTful API. Тя предоставя стандартизиран начин за описание на структурата и поведението на приложните програмни интерфейси, като се използва формат JSON или YAML. Swagger позволява на разработчиците да описват адреси на различни ендпоинти, формати на заявка/ответ, параметри, методи за автентикация и др. Той генерира интерактивна документация, която служи като справка за разработчиците и опростява потреблението на API. Освен това Swagger поддържа генериране на код, което позволява на разработчиците автоматично да генерират сървърни модули и клиентски SDK въз основа на дефиницията на API. Като цяло Swagger насърчава откриваемостта на API, сътрудничеството и оперативната съвместимост, като предоставя обща и еднаква спецификация за проектиране и документиране на API. Swagger документацията е достъпна на /api ендпоинта.

Използвана литература

1. [Atmos 41](#)
2. [Weathercom](#)
3. [Arduino Uno](#)
4. [Wemos D1 R1](#)
5. [DHT11](#)
6. [BMP180](#)
7. [DS1307](#)
8. [MicroSD Card Reader](#)
9. [ESP8266](#)
10. [Docker](#)
11. [PostgreSQL](#)
12. [Node.js](#)
13. [Express](#)
14. [NestJS](#)
15. [Prisma](#)
16. [Handlebars](#)
17. [ChartJS](#)
18. [Swagger](#)

Създадени символи (KiCAD)

1. BMP180
2. DS1307
3. MicroSD Card Reader

Съдържание

Увод	2
I. Съществуващи метеорологични станции и съпоставка с All-in-one-weather	3
1. Atmos 41	3
2. Weathercom	4
1) Weathercom Lite	6
2) Weathercom Smart	6
3) Weathercom Pro	6
3. Съпоставка между All-in-one-weather и пазарните конкуренти	7
II. Реализация	9
1. Блок схема	9
2. Начин на работа	10
III. Електрическа схема и компоненти	11
1. Компоненти на устройството-клиент	11
• Arduino Uno	11
• DHT11 Breakout	12
• BMP180 (GY-68)	13
• DS1307 (RTC)	14
• MicroSD Card Reader	15
• EByte LoRa E220-900T-22D	16
2. Компоненти на устройството-сървър	17
• Arduino Uno	17
• EByte LoRa E220-900T-22D	18
• Wemos D1 R1	19
3. Електрическа схема	20
• Устройство-клиент	20
• Устройство-сървър	21

IV. Комуникации	22
1. UART (Universal Asynchronous Receiver-Transmitter)	22
2. I2C (Inter-Integrated Circuit)	23
3. SPI (Serial Peripheral Interface)	24
4. LoRa (Long Range)	25
IV. Сървър	26
1. Използвани технологии	26
1.1. Docker	26
1.2. PostgreSQL	27
1.3. Node.js	28
1.4. Express	29
1.5. NestJS	30
1.6. Prisma	31
1.7. Handlebars	32
1.8. ChartJS	33
1.9. Swagger	34
Използвана литература	35
Създадени символи (KiCAD)	35
Съдържание	36