

Міністерство освіти і науки України
ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»
Кафедра комп'ютерної інженерії та електроніки

Сачовський Андрій Мирославович
Andrii Sachovskyi

УДК 004:681.5

Спеціальність 123 «Комп'ютерна інженерія»
(шифр та назва спеціальності)

Кваліфікаційна робота
на здобуття освітньо-кваліфікаційного рівня бакалавр
(бакалавр, спеціаліст, магістр)

Розробка системи передавання метеоданих
через безпроводний протокол LoRa
Development of a meteorological data transmission
system via LoRa wireless protocol

Науковий керівник:
кандидат фіз.-мат. наук,
доцент Павлюк М.Ф.

Рецензенти:
Кандидат хім. наук,
проф. каф. фізики і хімії
твердого тіла
Горічок І.В.

Івано-Франківськ
2020

| Форм. | Зона | Поз. | Позначення | Найменування | К-ть. | Прим. |
|------------|-------------|-----------------|-------------------|-------------------------------|-------------|---------------|
| | | | | | | |
| | | | 123.УДК:004.681.5 | Блок-схема алгоритму програми | 1 | |
| | | | 123.УДК:004.681.5 | Принципова електрична схема | 1 | |
| | | | 123.УДК:004.681.5 | Пояснювальна записка | 58 | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | 123.УДК:004.681.5 | | |
| Зм. | Лист | № докум. | Підпис | Дата | | |
| Розроб. | | Сачуксый АМ | | | Лист | Листів |
| Пров. | | Павлюк М.Ф. | | | | 2 |
| Н.контр. | | | | | | |
| Утв. | | | | | | |
| | | | | Специфікація | | |

АНОТАЦІЯ

В бакалаврській дипломній роботі розроблено пристрій для збору метеоданих, та подальша передача отриманих даних за допомогою зв'язку LoRa. Даний пристрій призначений для збору метеоданих та передачі цих даних на центральний сервер. Основу системи складає радіомодуль LoRa Ra-02. Ініціалізацію, збір та підготовку даних до відправки виконує мікроконтролер ESP32 компанії Espresiff.

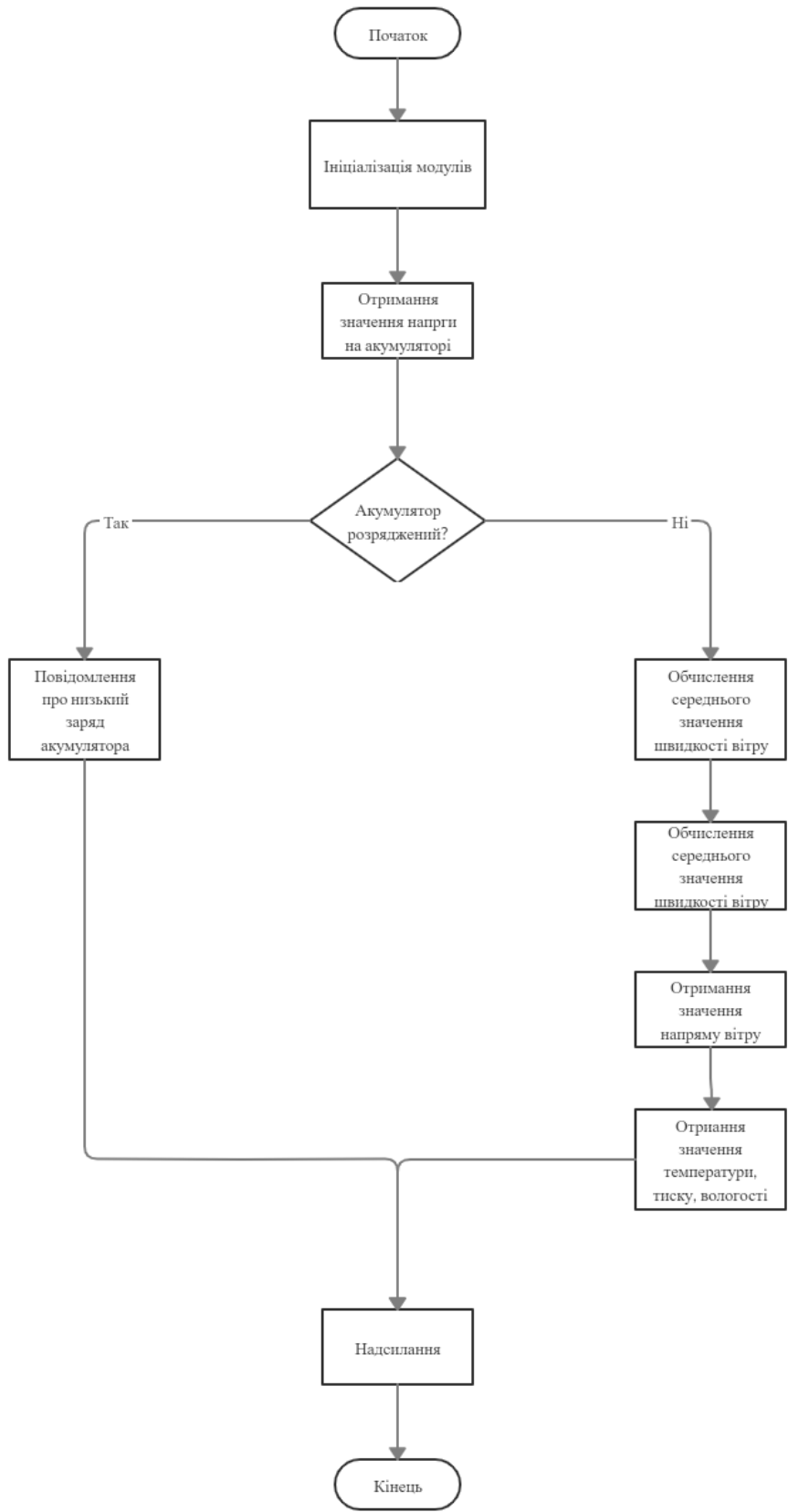
В роботі описується апаратна та програмна реалізація пристрою. Зроблено вибір та обґрунтування вибраних технічних засобів. Проведено економічну оцінку отриманого пристрою.

ABSTRACT

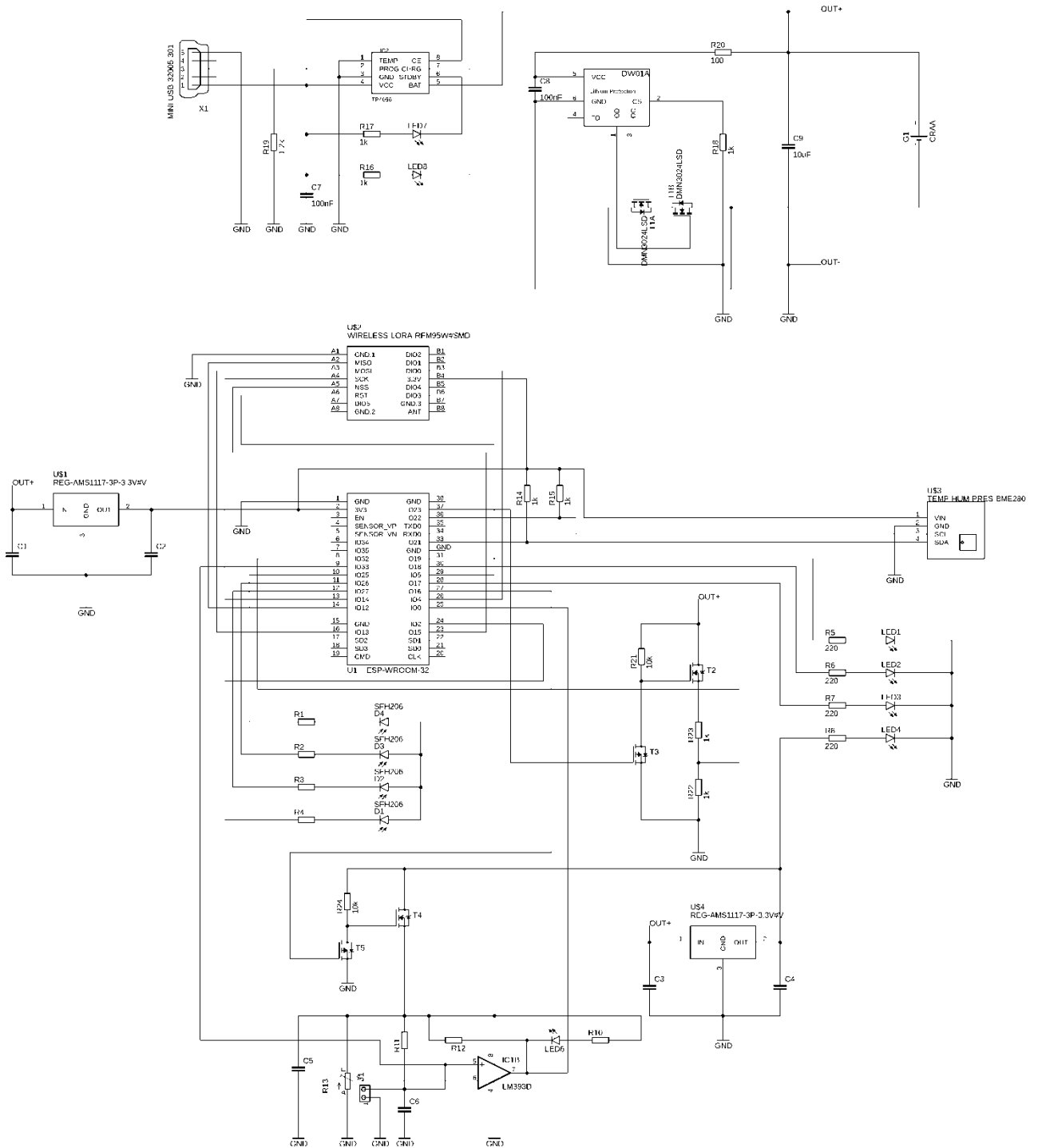
In graduation work has developed a device for collecting meteorological data and the subsequent transmission of the obtained data by means of LoRa communication. This device is intended for collecting weather data and transmitting this data to a central server. The basis of the system is the radio module LoRa Ra-02. The ESP32 microcontroller of Espresiff initializes, collects and prepares the data for dispatch.

The thesis describes the hardware and software implementation of the device. The selection and justification of the selected technical means. An economic evaluation of the device was carried out.

| | | | | | | | |
|-------------|-------------|------------------|---------------|-------------|--------------------------|-------------|----------------|
| | | | | | <i>123.УДК:004:681.5</i> | | |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | Анотація | | |
| Розробив | | Сачуєвський А.М. | | | | | |
| Перевірив | | Павлюк М.Ф. | | | | | |
| Н. Контр. | | . | | | | | |
| Затвердив | | | | | | | |
| | | | | | <i>Літ.</i> | <i>Арк.</i> | <i>Аркушіє</i> |
| | | | | | | 3 | 1 |



| | | | | | | | | | | |
|-------------|-------------|-----------------|---------------|-------------|--------------------------------------|--|--|-------------|-------------|----------------|
| | | | | | <i>123.УДК:004:681.5</i> | | | | | |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | Блок-схема алгоритму роботи програми | | | | | |
| Розробив | | Сачовський А.М. | | | | | | <i>Літ.</i> | <i>Арк.</i> | <i>Аркушів</i> |
| Перевірив | | Павлюк М.Ф. | | | | | | | 4 | 1 |
| Н. Контр. | | | | | | | | | | |
| Затвердив | | | | | | | | | | |



| | | | | |
|-----------|------|----------------|--------|------|
| Змн. | Арк. | № докум. | Підпис | Дата |
| Розробив | | Сяювський А.М. | | |
| Перевірив | | Павлюк М.Ф. | | |
| Н. Контр. | | | | |
| Затвердив | | | | |

123.УДК:004:681.5

Електрична принципова
схема

| | | |
|------|------|---------|
| Лит. | Арк. | Аркушів |
| | 5 | 1 |

Пояснювальна записка
до кваліфікаційної роботи

на тему:

**«Розробка системи передавання метеоданих через безпроводний
протокол LoRa»**

123.УДК:004:681.5

| Змн. | Арк. | № докум. | Підпис | Дата | | | | |
|-----------|------|-----------------|--------|------|----------------------|------|------|---------|
| Розробив | | Сачовський А.М. | | | Пояснювальна записка | Літ. | Арк. | Аркушів |
| Перевірив | | Павлюк М.Ф. | | | | | 6 | 58 |
| | | | | | | | | |
| Н. Контр. | | | | | | | | |
| Затвердив | | | | | | | | |

ПЕРЕЛІК ОСНОВНИХ СКОРОЧЕНЬ

LPWAN(Low Power Wide Area Network) – мережа широкого радіусу дії з низьким рівнем енергоспоживання

I2C(Inter-Integrated Circuit) – послідовна асиметрична шина для зв'язку з між пристроями

SPI(Serial Peripheral Interface) – повно дуплексний швидкісний синхронний трьохпровідний інтерфейс

AES(Advanced Encryption Standard) - стандарт розширеного шифрування, заснований на симетричному алгоритмі блочного шифрування

IoT (Internet of Things) - концепція мережі, що складається із взаємозв'язаних фізичних пристроїв

LGA(Land Grid Array) – тип корпусу мікросхем, що використовує матрицю контактних площадок, які розташовані на корпусі мікросхеми.

ТКО – температурний коефіцієнт опору, величина, що дорівнює відносній зміні електричного опору ділянки електричного кола або питомого опору речовини при зміні температури на один градус.

3GPP(3rd Generation Partnership Project) – партнерська асоціація груп телекомунікаційних компаній, головною метою створення якої є розробка і затвердження стандартів для мережевих технологій третього покоління(3G), стандартизація архітектури мереж та сервісів.

АЦП – аналогово-цифровий перетворювач.

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 7 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

ЗМІСТ

| | |
|---|----|
| ВСТУП..... | 10 |
| 1. АНАЛІЗ ОСНОВНИХ ЗАСОБІВ ПЕРЕДАЧІ ТА ЗБОРУ ДАНИХ..... | 11 |
| 1.1 Засоби передачі даних..... | 11 |
| 1.1.1 GSM/GPRS | 11 |
| 1.1.2 NB-IoT..... | 12 |
| 1.1.3 SIGFOX..... | 14 |
| 1.1.4 LoRa..... | 15 |
| 1.2 Методи вимірювання температури..... | 18 |
| 1.3 Методи вимірювання атмосферного тиску | 19 |
| 1.4 Методи вимірювання вологості повітря..... | 20 |
| 1.5 Методи вимірювання швидкості та визначення напрямку вітру | 23 |
| 2. ВИБІР ЕЛЕМЕНТНОЇ БАЗИ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ | 27 |
| 2.1 Вибір мови програмування | 27 |
| 2.2 Вибір мікроконтролера..... | 28 |
| 2.3 Вибір радіомодуля..... | 29 |
| 2.4 Вибір датчиків температури, тиску, вологості | 31 |
| 2.5 Вибір датчиків швидкості і напрямку вітру | 34 |
| 2.6 Вибір датчиків атмосферних опадів | 35 |
| 2.7 Модуль захисту та заряду акумулятора | 36 |
| 3. ПРОЕКТНИЙ РОЗДІЛ..... | 37 |
| 3.1. Підключення основних вузлів до мікроконтролера..... | 37 |
| 3.2. Розрахунок енергоспоживання пристрою | 39 |
| 3.3. Опис алгоритму роботи пристрою..... | 40 |
| 5. ЕКОНОМІЧНА ЧАСТИНА..... | 47 |
| ВИСНОВКИ | 49 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ | 50 |
| ДОДАТКИ..... | 51 |

ВСТУП

Спостереження за метеорологічними змінами здавна дозволяло людям визначати ті чи інші погодні умови. Як і колись так і сьогодні ці дані є досить важливими для людей. Для спостереження за погодою задіяна велика кількість техніки та людей, оскільки дана інформація допомагає запобігти серйозним наслідкам.

Все більше і більше пристроїв створюється для моніторингу певних систем. Однак постає проблема передачі інформації з різних датчиків для моніторингу або керування системою. Дальність передачі даних є важливим параметром для різного роду зв'язку. Існуючі засоби є ефективні якщо датчики розміщені не на великих дистанціях (Bluetooth, Wi-Fi).

Не менш важливим є енергоспоживання, що є вагомим аргументом для портативних пристроїв, які працюють від акумулятора. Зрозуміло що чим вища дальність зв'язку тим більше енергоспоживання передавача, що в свою чергу створює певні складнощі для збільшення автономної роботи пристрою.

Метою виконання даної роботи є створення пристрою для передачі метеорологічних даних (температура, тиск, вологість, напрям та швидкість вітру), з використанням зв'язку LoRa, вивчення основних методів вимірювання температури, тиску та вологості, визначення напрямку та швидкості вітру, а також подальша передача метеоданих на центральний сервер.

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 9 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

1. АНАЛІЗ ОСНОВНИХ ЗАСОБІВ ПЕРЕДАЧІ ТА ЗБОРУ ДАНИХ

Гострою проблемою на шляху між кінцевим пристроєм та центральним сервером є зв'язок. На сьогоднішній день є багато різних технологій та стандартів зв'язку, але далеко не всі підходять для мобільних пристроїв які живляться від акумулятора, чи розміщені на значній відстані від центрального шлюзу. Для таких пристроїв розроблені спеціальні мережі LPWAN. LPWAN – технологія що забезпечує передачу невеликих по об'єму даних, на значних відстанях, призначена для між машинного з'єднання та Інтернету речей. Дана мережа дає можливість отримувати дані з пристроїв: давачі, лічильники тощо. До цих мереж входять NB-IoT, SigFox, LoRaWAN.

1.1. Засоби передачі даних

1.1.1. Передача даних за допомогою GSM/GPRS

Дуже часто для передачі даних від давачів до центрального сервера використовують GSM/GPRS технології. За допомогою цієї технології можна відправляти дані у вигляді SMS, чи обмінюватись даними по GPRS. На даний момент існує багато модулів для автономних пристроїв – SIM900, SIM800L, A6, A7. Всі пристрої керуються AT командами. За допомогою цих команд можна задати номер кому відправляти текстові повідомлення з даними, чи задати адресу куди пересилати дані по GPRS. За допомогою даних модулів можна передавати дані на великі відстані. На сьогоднішній день оператори мобільного зв'язку розробили тарифи призначені для таких пристроїв.

Перевагою використання GSM/GPRS є:

- Значна відстань передачі даних
- Простота налаштування

Недоліками є високе споживання, необхідне покриття території мобільною мережею, висока ціна.

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 10 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

1.1.2.Передача даних за допомогою NB-IoT

NB-IoT – технологія зв’язку побудована на основі LTE, яка призначена для пристроїв, які передають невелику кількість інформації та живляться від акумулятора чи батарейки типу АА. Дана технологія стандартизована партнерською асоціацією 3GPP[1].

NB-IoT дозволяє операторам працювати з різними роду пристроями для моніторингу чи відслідковування. Сама технологія дозволяє операторам мобільного зв’язку розгортати мережу NB-IoT у GSM та LTE мережах. Для цього потрібне тільки спеціальне програмне забезпечення. Особливістю NB-IoT є те що він може розгортатись як і на існуючих мережах GSM, LTE так і окремою мережею.

Консорціум 3GPP пропонує 3 способи виділення частот для NB-IoT на існуючих мобільних мережах: автономний(Stand-Alone), внутрішньо смуговий(In-Band), на захисній смузі(Guard-Band). При використанні Stand-Alone методу, виділяється окремий частотний канал шириною 200кГц. Даний метод є найефективнішим для NB-IoT, але водночас і затратним. Для підтримки каналу шириною 200кГц потрібно додати захисні інтервали, тому в деяких випадках потрібно виділяти від 300 до 600кГц.

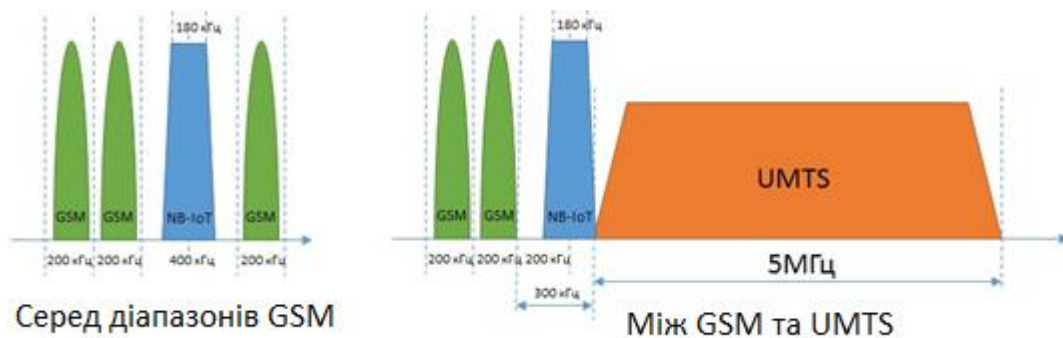


Рисунок 1.1. Розміщення каналу NB-IoT у режимі Stand-Alone

Як видно з рисунку для використання NB-IoT виокремлюють окремий канал у GSM спектрі, або розміщують канал між GSM та UMTS каналами. При використанні In-Band для NB-IoT виділяється канал всередині існуючої LTE мережі. Сигнал NB-IoT підсилюють на 6дБ. Завдяки цьому розпізнають дані для

NB-IoT. Даний варіант економить частотний ресурс оператора мобільного зв'язку, проте постає проблема взаємовпливу з мережею LTE.

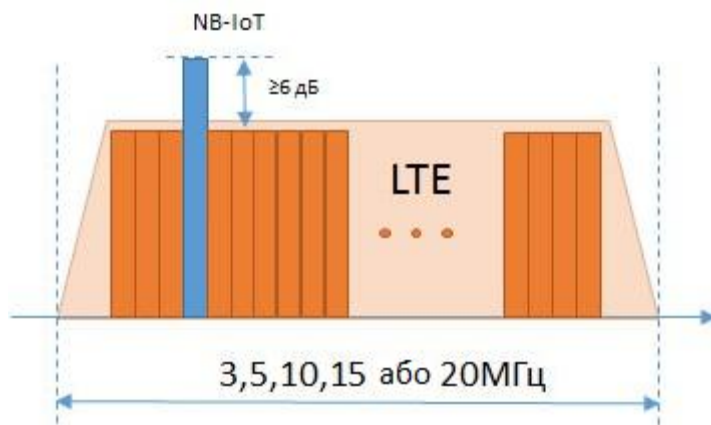


Рисунок 1.2. Розміщення каналу NB-IoT у режимі In-Band

При використанні Guard-Band канал NB-IoT виділяється в так званій захисній зоні. Наприклад, в смузі LTE 10МГц, по 500кГц захисного інтервалу. В даному режимі як і в режимі In-Band сигнал підсилений на 6-9дБ. Даний варіант дає можливість економити частотний простір і зменшити взаємовплив на мережу LTE.

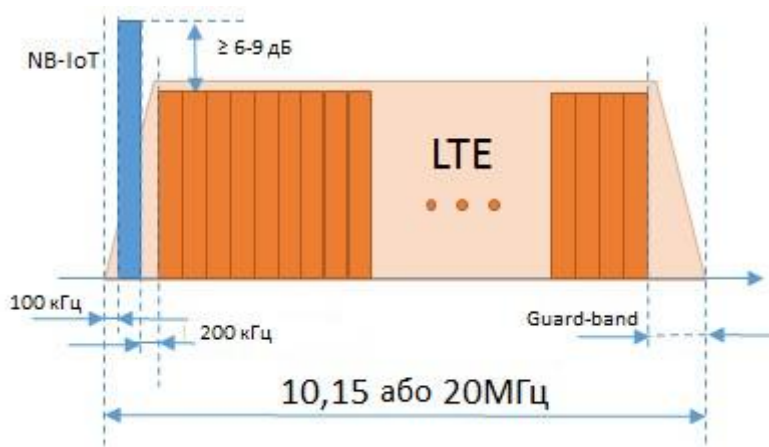


Рисунок 1.2. Розміщення каналу NB-IoT у режимі Guard-Band

Переваги NB-IoT:

- Низьке енергоспоживання кінцевих пристроїв
- Глобальне покриття
- Низька вартість послуг зв'язку[2]

Недоліками є:

- Низька швидкість передачі даних

| | | | | | | | |
|-----|------|----------|--------|------|--|-------------------|------|
| | | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | | 12 |
| Зм. | Арк. | № докум. | Підпис | Дата | | | |

- Пристрій прив'язаний до шлюзу
- Необхідність покриття мобільною мережею
- Висока ціна кінцевих пристроїв
- Висока ціна шлюзів

1.1.3. Передача даних за допомогою SigFox

Технологія SigFox дозволяє передавати дані на достатньо великі відстані, при цьому потужність передачі і енергоспоживання є достатньо низьким. Дана мережа добре підходить для автоматизованих пристроїв, які передають невелику кількість інформації. Сама мережа SigFox схоже, на звичайну мережу мобільного зв'язку, проте є більш енергоефективною. Мережа SigFox використовує вузьку полосу частот(Ultra Narrow Band, UNB) на основі радіо технології, для підключення пристроїв до глобальної мережі. Використання UNB дозволяє використовувати малопотужні передавачі. Дальність дії мережі SigFox зазвичай становить близько 30-50 км на відкритій місцевості і 3-10 км у місцях з високою щільністю забудов. На рисунку зображено топологію мережі технології SigFox.

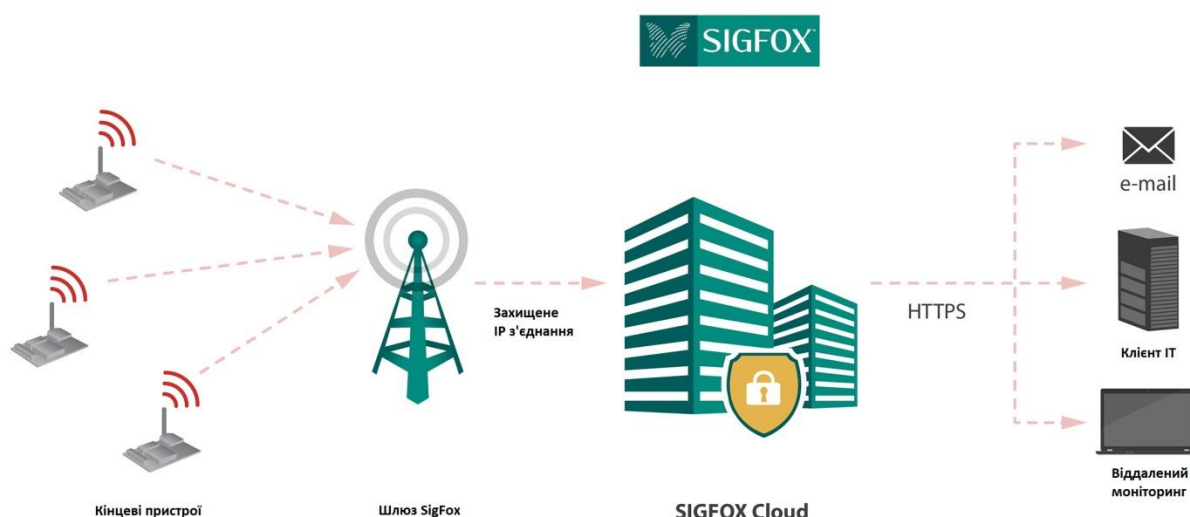


Рисунок 1.3. Архітектура мережі SigFox

Розмір повідомлення від кінцевого пристрою не повинен перевищувати 12 байт. Цього достатньо щоб передати дані від будь-яких давачів. Всі дані які передаються повинні бути зашифровані, тому технологія використовує AES

| | | | | | | | | | | |
|-----|------|----------|--------|------|--|--|--|--|--|------|
| | | | | | | | | | | Арк. |
| | | | | | | | | | | 13 |
| Зм. | Арк. | № докум. | Підпис | Дата | | | | | | |

шифрування з закритим ключем, який записаний в пристрої. Технологія SigFox націлена на низьку ціну пристрою, де потрібна широка зона покриття.

Перевагами технології SigFox:

- Велика зона покриття
- Висока проникна здатність
- Низьке енергоспоживання, що дозволяє довго працювати автономним пристроям від акумулятора.

Недоліками мережі є:

- Низька швидкість передачі даних
- Залежність від стільникової архітектури
- Обмежена поміхо стійкість
- Висока ціна
- Малий пакет для передачі даних
- Достатньо великий час передачі даних

1.1.4. Передача даних за допомогою LoRaWAN

LoRaWAN – мережевий протокол з низьким енергоспоживанням, та великою площею покриття, призначений для бездротового підключення датчиків(сенсорів), що працюють від акумулятора чи сонячних елементів, до Інтернету в регіональних, національних або глобальних мережах які орієнтовані на ключові вимоги Інтернету речей(IoT)[3].

Багато бездротових систем використовують FSK модуляцію як фізичний рівень оскільки вона є достатньо ефективною модуляцією для досягнення низькою потужності. LoRa базується на CSS модуляції, яка підтримує ті ж характеристики низького енергоспоживання, що і модуляція FSK, але значно збільшує дальність зв'язку.

Мережева архітектура LoRaWAN з'єднана у топологію типу зірка, в якій шлюзи(базові станції) передають повідомлення між кінцевими пристроями та центральним мережевим сервером. Шлюзи підключаються до мережевого

| | | | | | | | | | | |
|-----|------|----------|--------|------|--|--|--|--|--|------|
| | | | | | | | | | | Арк. |
| | | | | | | | | | | 14 |
| Зм. | Арк. | № докум. | Підпис | Дата | | | | | | |

123.УДК:004:681.5

сервера за допомогою стандартних IP-з'єднань і виконують роль прозорого моста, перетворюючи RF-пакети в IP-пакети і навпаки.

У мережі LoRaWAN виділяють такі основні частини:

Кінцевий вузол – пристрій який використовується для отримання даних з датчиків або здійснює керування.

Шлюз – пристрій який отримує дані з кінцевих вузлів за допомогою радіоканалу і передає їх в транзитну мережу. Транзитною мережею можуть бути Ethernet, або Wi-Fi чи інші телекомунікаційні канали зв'язку. [4]

Мережевий сервер виконує керування мережею, тобто задає розклад, виконує адаптацію швидкості, зберігає і опрацьовує отримані дані.

Мережеві додатки – програми які віддалено контролюють роботу кінцевих вузлів і отримують необхідні дані з них.

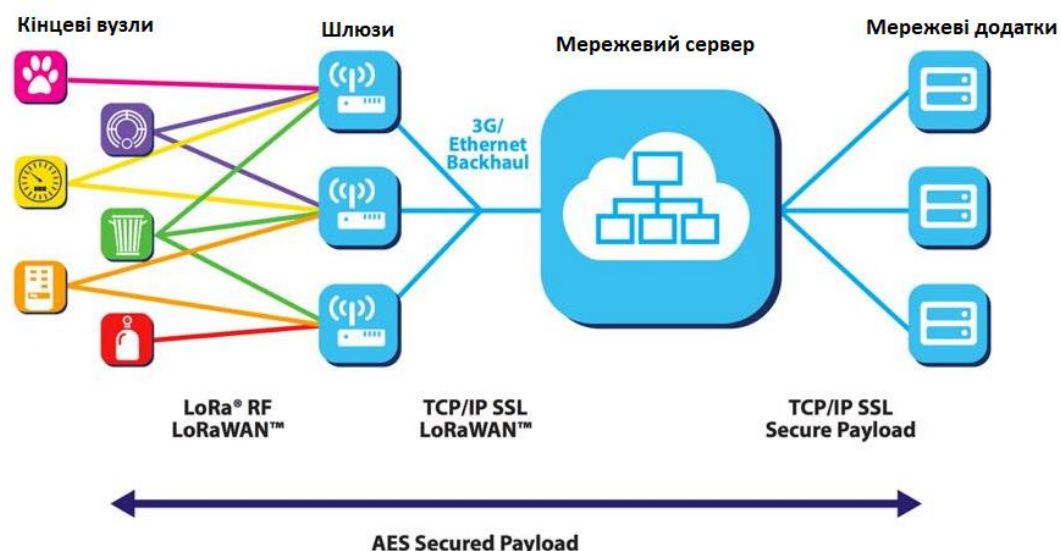


Рисунок 1.4. Архітектура мережі LoRaWAN

Важливими факторами LoRaWAN є:

- Мережева архітектура
- Дальність зв'язку
- Термін роботи пристрою від акумулятора
- Стійкість до перешкод
- Ємність мережі(максимальна кількість вузлів у мережі)
- Мережева безпека
- Одностороння чи двостороння передача даних

- Різноманітність програм що обслуговують мережу

LoRaWAN визначає протокол зв'язку та архітектуру системи для мережі, в той час як фізичний рівень LoRa дозволяє підтримувати дальність зв'язку. Протокол та архітектура мережі мають найбільший вплив при визначення терміну служби акумулятора вузла, ємність мережі, якість обслуговування, безпека і різноманітність програм що обслуговує система.

У мережі LoRaWAN вузли не пов'язані з певним шлюзом. Натомість дані, що передаються вузлом, зазвичай приймаються декількома шлюзами кожен шлюз буде пересилати отриманий пакет з кінцевого вузла у мережевий сервер.

Термін служби акумулятора

Вузли в мережі LoRaWAN асинхронні і спілкуються, коли вони отримали дані для надсилання, які керовані подіями чи заплановані. У синхронних мережа вузлам потрібно часто «прокидатися» для синхронізації з мережею. Ця синхронізація споживає багато електроенергії, що в свою чергу призводить до скорочення часу роботи від акумулятора.

Виділяють 3 види кінцевих пристроїв: Пристрої класу А, В та С.

Для двонаправлених пристроїв класу А характерна мінімальна споживана потужність при переобладнанні для відправки даних на сервер. Зазвичай пристрої класу А починають сеанс зв'язку. Пристрій відправляє дані на сервер, після чого він переходить в режим отримувача та очікує відповіді від сервера. Таким чином сервер може зв'язатися з пристроєм тільки після отримання пакету інформації.

На відміну від класу А пристрої класу Б використовують додатковий сеанс зв'язку, який пристрій починає по розкладу. Розклад пристрій узгоджує з шлюзом. Завдяки цьому додатковому сеансу сервер має змогу передати дані чи команду на пристрій у відомий для нього час.

Пристрої класу С зазвичай завжди очікують сигналу від сервера. При відповідному сигналу дані пристрої відправляють дані. Такі пристрої використовують тоді, коли потрібно отримувати велику кількість даних.

| | | | | | | | | | | |
|-----|------|----------|--------|------|--|--|--|--|--|------|
| | | | | | | | | | | Арк. |
| | | | | | | | | | | 16 |
| Зм. | Арк. | № докум. | Підпис | Дата | | | | | | |

LoRaWAN використовує не ліцензовані частоти, що значно здешевлює саму технологію, оскільки не потрібно отримувати дозвіл на використання частоти та оплачувати її використання.

Переваги:

- Велика площа покриття
- Низьке енергоспоживання кінцевих пристроїв
- Висока масштабованість мережі
- Низька ціна для розгортання мережі
- Використання вільних частот
- Кінцеві пристрої не прив'язані до шлюзу
- Адаптивна швидкість та вихідна потужність пристрою, що задається мережевим сервером

Недоліками:

- Затримка передачі інформації від кінцевого пристрою в залежності від дальності розміщення
- Ризики зашумлення спектру
- Обмежена потужність сигналу

1.2. Методи вимірювання температури

Вимірювання температури за допомогою термопара

Термопара — чутливий елемент термоелектричного перетворювача у вигляді двох ізольованих провідників із різнорідних матеріалів, з'єднаних на одному кінці, принцип дії якого ґрунтується на використанні термоелектричного ефекту для вимірювання температури[5]. Звичайна термопара це контакт двох різних спаяних металів(термоелектродів). У якості теромелектродів використовують як і чисті метали – платина, залізо, мідь так і сплави – алюмель, константан, копель, платинорідій, хромель. В основу дії термопари покладено ефект Зеєбека. Тобто при нагріванні контакту утворюється ЕРС який пропорційний різниці температур. Діапазон вимірювань становить -50-120°С

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 17 |

Вимірювання температури за допомогою терморезисторів

Терморезистор – це нелінійний резистор, виготовлений із напівпровідникового матеріалу, опір якого залежить від температури[6]. Особливістю терморезисторів є великий температурний коефіцієнт опору(ТКО), який є значно більший ніж у металів, і визначається за формулою:

$$\alpha = \frac{\Delta R}{\Delta T} * \frac{1}{R} * 100\% \quad (1.1)$$

Перевагами терморезисторів є простота використання, можливість працювати у складних кліматичних умовах, а також відзначаються стабільністю характеристик у часі. До основних параметрів терморезисторів відносять:

- Номінальний опір
- Температурний коефіцієнт опору
- Висока чутливість
- Інтервал робочих температур
- Максимально допустима потужність розсіювання
- Низька ціна
- Невеликий розмір

Усі терморезистори можна поділити на дві групи: терморезистори з негативним ТКО – термістори, терморезистори з позитивним ТКО – позистори.

Значення ТКО для різних термісторів при кімнатній температурі знаходиться в межах $\alpha = 0,8 * 10^{-2} - 6 * 10^{-2} (1/K)$ тобто 0,8-6% на градус.

Для позисторів значення α може досягати і десятків відсотків на один градус.

Термістори можна з легкістю використати для вимірювання температури навколишнього середовища.

1.3. Методи вимірювання атмосферного тиску

Для вимірювання атмосферного тиску існує 3 види давачів:

- Ємнісні
- Тензорезистивні
- П'єзоелектричні

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 18 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Вимірювання атмосферного тиску за допомогою ємнісних давачів тиску
У ємнісні давачі тиску зазвичай використовують кремнієві діафрагми. Під дією тиску відбувається зміщення діафрагми відносно опорної пластини, тим самим змінюючи ємність між ними. Дані давачі найефективніше працюють при невисоких тисках. Монолітний ємнісний давач тиску, який виготовлений з кремнієвих кристалів, володіють достатньо високою стабільністю робочих характеристик. Рух діафрагми може забезпечити 25% зміни ємності, що дає змогу проведення прямого оцифрування отриманої величини тиску.

Вимірювання атмосферного тиску за допомогою тензорезистивних давачі тиску. У даних давачах зазвичай використовується два компонента: пластинка відомої площі(діафрагма), а також детектор(тензорезистори), який створює сигнал пропорційний тиску.

Давач тиску у якій використовується кремнієва діафрагма складається з діафрагми та вбудованих в неї дифузійним методом тензорезистори. За рахунок того що монокристалічний кремній має достатньо хороші характеристики пружності, то в такому давачі відсутні інертність і гістерезис, навіть при високих тисках.

Максимальна вихідна напруга даних давачів є невеликою і становить кілька сотень мілівольт, тому на їх виході ставлять підсилювачі сигналів. Одним з недоліком таких давачів є те що тензорестори мають досить сильну температурну чутливість, тому при розробці пристроїв на їх основі потрібно створювати коло температурної залежності.

1.4. Методи вимірювання вологості

За принципом роботи давачі вологості можна поділити на такі типи:

- Ємнісні
- Резистивні
- Термісторні
- Оптичні
- Електронні

| | | | | | | |
|-----|------|----------|--------|------|--------------------------|------|
| | | | | | <i>123.УДК:004:681.5</i> | Арк. |
| | | | | | | 19 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Вимірювання вологості повітря за допомогою ємнісний давача вологості
Ємнісний давач являє собою конденсатор. Обкладки якого є електроди у вигляді пластин а діелектриком служить саме повітря між електродами. Так як вологість безпосередньо впливає діелектричну проникність повітря. Зміна вологості діелектрика призводить до зміни ємності повітряного конденсатора.[7]

Існує інший варіант ємнісного давача. В якості діелектрика використовують матеріал з діелектричною проникністю, яка змінюється в залежності від вологості.

Наступний ємнісний давач – це тонкоплівковий гігрометр, що складається з підкладки з двома електродами виготовлені у вигляді гребінки. В даному давачі ці гребінки являють собою обкладками конденсатора. Недоліком даного давача є його температурна чутливість, тому для компенсації температури в давач вводять термоелементи.

Перевагою ємнісних давачів є те що вони стійкі до конденсату, можуть працювати у широкому діапазоні вимірювань.

Вимірювання вологості повітря за допомогою резистивного давача вологості

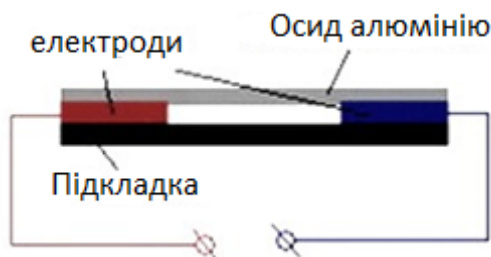


Рисунок 1.5. Будова резистивного давача вологості

Резистивні давачі складаються з 2 електродів. Зазвичай ці електроди нанесені на підкладку. На електроди нанесений шар струмопровідного матеріалу, в якості такого матеріалу може виступати оксид алюмінію або інші оксиди металів. Оксид алюмінію добре поглинає вологу з зовнішнього середовища, при цьому значно міняючи свій опір. Значення пройденого струму буде відображати значення рівня вологи. Перевагами таких давачів є:

- Низька вартість
- Низька споживана потужність

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 20 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

- Швидкий час відгуку
- Лінійність
- Низький гістерезис
- Широкий діапазон роботи
- Малий розмір

Вимірювання вологості повітря за допомогою термісторного давача вологості

Термісторний давач вологості є аналогом звичайного психрометра.

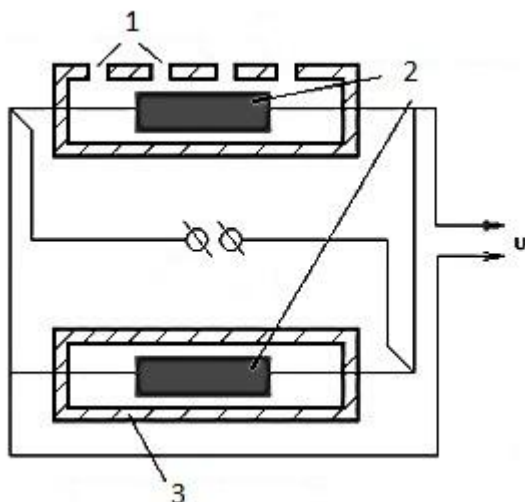


Рисунок 1.6. Термісторний давач вологості

де 1 – камера з отворами, 2 – термістори, 3 – герметична камера.

В даному давачі використовується два термістори(2). Один термістор поміщають у герметичну камеру(3) з сухим повітрям. Другий термістор знаходиться в камері з отворами(1), через які проходить вологе повітря. Принцип роботи такого давача вологості базується на тому, що при підвищенні вологості з поверхні термістора буде випаровуватись вода, цим самим охолоджуючи сам термістор. Чим вища вологість повітря тим більше буде охолоджуватись термістор. Використовуючи АЦП можна обчислити різницю напруг між двома термісторами. Це і буде значення вологості. Також можна використати мостову схему. На одну діагональ подається живлення, а з другою знімаються покази.

Вимірювання вологості повітря за допомогою оптичних давачів вологості

Даний вид давачів є найточнішим серед вище перерахованих.

Принцип роботи оптичних давачів вологості базується на фізичному явищі - точки роси. Коли система доходить до точки роси, то рідина і газ перебувають у рівновазі термодинаміки.

В реальних пристроях використовують дзеркало, температура якого, вища точки роси. Температуру дзеркала поступово знижують, до тих пір поки на дзеркалі не утвориться конденсат. Температурою переходу з газоподібного стану в рідкий і є точкою роси. Ця температура залежить від тиску і вологості навколишнього середовища. Якщо визначити температуру і тиск то можна легко визначити і вологість повітря. Даний метод є основним при вимірювання вологості оптичними давачами.

Для визначення конденсату на поверхні дзеркала використовують світлодіод та фоторезистор. У якості нагрівника чи охолоджувача дзеркала використовують модуль Пельте. На дзеркало встановлюють давачі температури. Під час утворення конденсату промінь світла від світлодіода буде розсіюватись, що призведе до зменшення струму у фоторезисторі. Отримавши дані з фоторезистора, потрібно підтримати температуру дзеркала, а давач температури визначить температуру. Знаючи температуру і тиск можна визначити рівень вологості.

1.5. Методи вимірювання напряму та швидкості вітру

Для вимірювання швидкості вітру використовують різного роду пристрої, які називають анемометрами. Кожен з них має свої переваги та недоліки. В залежності від способу вимірювання розрізняють:

- Обертальні анемометри(крильчаткові, чашкові)
- Теплові
- Ультразвукові

Вимірювання швидкості вітру обертальними анемометрами

Чашковий анемометри

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 22 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Найпростішим і найбільш розповсюджуваним є чашковий метод вимірювання швидкості вітру. Даний пристрій складається з 3 – 4 чашок, які розміщені на хрестовині. За формою чашки є напівсферичні.



Рисунок 1.7. Чашковий анемометр

Зазвичай для реєстрування оборотів осі використовують оптичний енкодер, диск якого закріплений на осі з чашками. За допомогою міток на диску можна підраховувати імпульси при оберті вісі, за певний проміжок часу. Також використовують датчик Холла і невеликий магніт для підрахунку кількості обертів.

Якщо на пристрій потрапить потік, то тиск повітря на внутрішню поверхню чашечки буде значно більший за тиск повітря на зовнішню поверхню, як результат виникає обертання чашечок.

Чашкові анемометри вимірюють швидкість потоку в площині, яка є перпендикулярною до осі обертання чашок. Можна отримати миттєве значення швидкості а також усереднене за певний проміжок часу. Зазвичай усереднене значення отримують на 10 або 3 хвилинний інтервал. Діапазон вимірювання такого пристрою становить від 1 до 50 м/с. Перевагою чашкових анемометрів є певна стійкість до турбулентних потоків. [8]

| | | | | | | | | | |
|-----|------|----------|--------|------|--|--|--|--|------|
| | | | | | | | | | Арк. |
| | | | | | | | | | 23 |
| Зм. | Арк. | № докум. | Підпис | Дата | | | | | |

Теплові анемометри

Теплові анемометри зазвичай використовують для вимірювання повільних потоків. Так як даний тип анемометрів споживає велику кількість енергії, він не використовується у мобільних пристроях, які живляться від акумулятора

Анемометр складається з пластини або нитки розжарення по якій тече струм. Принцип роботи такого анемометра базується на збільшенні тепловтрат нагрітої пластинки чи нитки. Під дією вітру пластинка охолоджується. В залежності від швидкості вітру потрібна різна енергія щоб підтримувати температуру пластинки. Метал який використовується у якості нитки розжарення повинен володіти позитивним коефіцієнтом температурного опору. До таких металів належать: ніхром, вольфрам, срібло, платина.

Перевагами таких анемометрів є:

- простота конструкції
- здатність вимірювати малі потоки
- висока точність вимірювання
- мала інертність

Недоліками даних анемометрів є:

- постійне енергоспоживання
- потребують постійного калібрування
- швидко зношуються нагрівальні елементи

Ультразвукові анемометри

Найпростіший ультразвуковий анемометр складається з 2 пар ультразвукових давачів, кожен з яких розташований під кутом 90 градусів один до одного.

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 24 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

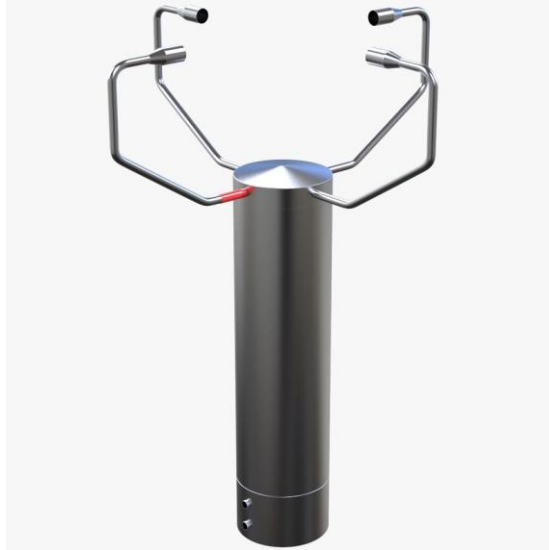


Рисунок 1.8. Ультразвуковий анемометр

Кожен датчик здатний випромінювати і приймати ультразвукові імпульси. Відстань між двома датчиками є строго визначена. Принцип роботи даного анемометра побудований на швидкості поширення звукових хвиль в залежності від напрямку вітру. Якщо вітру немає то час, який необхідний для проходження ультразвукового імпульсу з датчика N до датчика S буде рівний часу необхідний для проходження імпульсу з датчика S до датчика N. В іншому випадку звукова хвиля, напрям якої співпадає з напрямом вітру, пройде відстань між датчиками швидше ніж хвиля яка буде рухатися проти напрямку вітру. Наприклад, якщо вітер дме з N то час проходження хвилею від N до S буде більший порівняно з часом при проходженні хвилі від S до N. Отримані дані аналізують і обчислюють швидкість вітру. За допомогою ультразвукових анемометрів можна також визначити і сам напрям вітру.

Перевагами даних анемометрів є:

- відсутність обертових елементів
- точність вимірювання
- мала інертність
- стійкість до важких погодних умов

Недоліком є висока вартість та складність обслуговування анемометра.

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 25 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

2. ВИБІР ЕЛЕМЕНТНОЇ БАЗИ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ

Розробляється недорогий, мобільний пристрій для моніторингу за метеопказниками, з низьким енергоспоживанням, та передачею даних на центральний пристрій. При проектуванні пристрою було прийнято рішення використовувати мікроконтролер ESP та мову програмування Micropython.

2.1. Вибір мови програмування

Мова програмування Micropython є досить гнучкою у використанні та простою у вивченні.

MicroPython - це швидка та ефективна реалізація мови програмування Python3, яка включає в собі невелику кількість стандартних бібліотек Python та оптимізована для роботи на мікроконтролерах чи в обмежених середовищах.[9] MicroPython укомплектований передовими функціями такими як, цілі числа довільної точності, розуміння списків, генератори, обробка винятків тощо. Тим не менш, він досить компактний. Для його запуску необхідно лише 256 КБ кодового простору і 16 КБ оперативної пам'яті. Код не потрібно компілювати, тому не потрібно встановлювати додаткове програмне забезпечення.

Як і в звичайному Python3 так і в MicroPython підтримується інтерактивний режим. Це дає змогу виконувати різні команди разом із можливістю імпортування скриптів та їх запуск.

MicroPython підтримує велика кількість платформ та мікроконтролерів такі як:

- Pyboard
- ESP8266
- ESP32
- WiPy
- Raspberry Pi
- Micro:bit
- STM32f407VET6(ZET6)

На даний момент є велика кількість бібліотек необхідних для роботи з різного роду датчиками.

| | | | | | | | | | | |
|-----|------|----------|--------|------|--|--|--|--|--|------|
| | | | | | | | | | | Арк. |
| | | | | | | | | | | 26 |
| Зм. | Арк. | № докум. | Підпис | Дата | | | | | | |

2.2. Вибір мікроконтролера

Для реалізації пристрою було використано мікроконтролер ESP32. Даний мікроконтролер є достатньо популярний та потужний. ESP32 підтримує декілька мов програмування: C, Lua, MicroPython.



Рисунок 2.1. Зовнішній вигляд ESP32

ESP32 – це дешевий мікроконтролер з низьким енергоспоживанням[10]. ESP32-WROOM – модуль з мікроконтролером ESP32-D0WDQ6, та з мікросхемою Flash пам'яті на 4Мб з необхідними компонентами, які приховані під металеву кришку. Мікроконтролер ESP32-D0WDQ6 виконаний по технології SoC – система на кристалі.

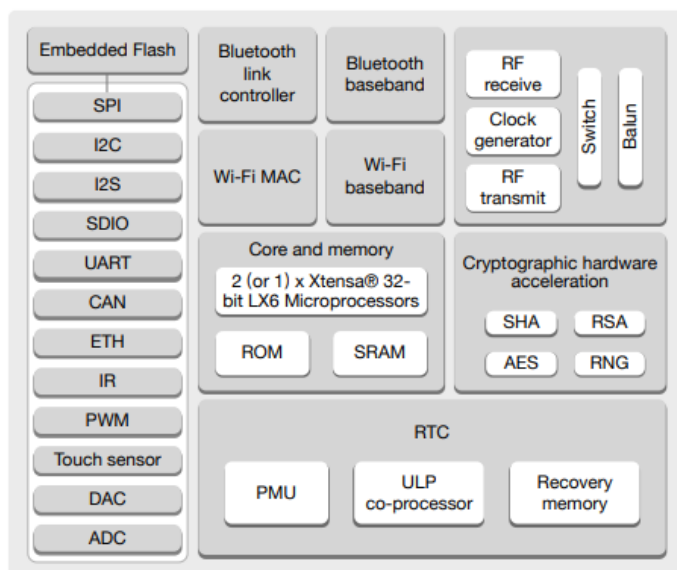


Рисунок 2.2. Функціональна діаграма ESP32

Як видно з функціональної діаграми ESP32 складається з 32-розрядного ядра Tensilica Xtensa LX6. В систему інтегрований радіочастотний тракт: балун, вбудовані антенні комутатори, радіочастотні компоненти, підсилювач потужності, фільтр. Мікроконтролер також включає в себе блок периферії в який входять: 3 UART, 2 I2C, 4 SPI, 2 I2S – для комунікації з іншими пристроями чи давачами. Також підтримується CAN 2.0 шина, 2 8-бітних ЦАП та 18 12-бітних АЦП, підтримка PWM на всіх портах. 10 портів можуть працювати в режимі сенсорних кнопок. Мікроконтролер також має вбудований давач температури та давач Холла.

ESP32 оснащений апаратним блоком шифрування. Даний блок дає можливість зашифрувати інформацію такими алгоритмами як AES, SHA, RSA, ECC а також використовувати генератор псевдовипадкових чисел.

Важливим питанням для портативний пристроїв що живляться від акумулятора є енергоспоживання. Максимальний струм споживання в режимі передачі інформації через Wi-Fi чи Bluetooth – 160-260мА. Якщо відключити Wi-Fi та Bluetooth споживання становитиме – 20 мА, а в режимі очікування – 10 мкА.

2.3. Вибір радіомодуля

Для передачі та прийому даних використовувався модуль LoRa Ra-02.



Рисунок 2.3. Зовнішній вигляд радіо модуля LoRa Ra-02

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 28 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

LoRa Ra-02 – це безпроводний передавач, який побудований на базі мікроконтролера SX1278 від компанії Semtech[11]. Під металевою кришкою розміщений сама мікросхема та вся необхідна обв'язка. Сама кришка захищає мікросхему від механічних пошкоджень та різних завад. Мікросхема Sx1278 використовує технологію LoRa, що здатна досягнути значної дальності ніж існуючі системи засновані на FSK чи OOK модуляції. Для максимальної зручності користувач налаштувати ширину спектру, коефіцієнтом розширення спектру, та швидкістю виправлення помилок. Використання широкого спектру, значно підвищує завадостійкість ніж при використанні вузького спектру.

Чутливість модуля становить приблизно -141дБ, а вихідна потужність сигналу 18дБ. Це дає змогу отримати радіозв'язок з дальністю до 10 км, на відкритій місцевості, при невеликому енергоспоживанні.

Передавач Ra-02 може працювати в діапазоні частот від 410 до 525 МГц. В Україні дані частоти є відкриті, тобто не потрібно отримувати ліцензії для їх використання. Для такого діапазону частот простіше підібрати чи виготовити антену. Для під'єднання зовнішньої антени модуль має роз'єм IPEX. Для початкової ініціалізації та обміну даними між мікроконтролером модуль LoRa Ra-02 використовує SPI. Ra-02 підтримує декілька режимів роботи.

Таблиця 2.1. Режими роботи радіомодуля LoRa Ra-02

| Режим роботи | Споживання | Одиниці вимірювання |
|-----------------|--|---------------------|
| Сон | 0,2 | мкА |
| Холостий хід | 1,5 | мкА |
| Очікування | 1,6 | мА |
| Прийому пакета | 10,8 | мА |
| Передача пакета | 120(20дБ) 87(17дБ) 29(13дБ) 20(2дБ) | мА |

В залежності від того наскільки далеко від шлюзу буде розміщуватись пристрій, його енергоспоживання буде різне. Чим ближче пристрій до шлюзу тим менша

вихідна потужність потрібна для передачі даних. Також енергоспоживання залежатиме від класу пристрою. Пристрій класу А буде менше споживати енергії ніж пристрій класу В чи С, оскільки він не додатково не прослуховує ефір, цим самим зменшує енерговитрати. Для зменшення енерговитрат пристрій вводять в режим сну. В такому режимі він буде “прокидатись” тільки для передачі чи прийому даних.

Невеликі розміри модуля 16x17 мм дають змогу створювати невеликі компактні пристрої.

2.4. Вибір датчика температури, тиску та вологості

BME280 це комбінований цифровий датчик вологості, тиску та температури. Датчик виконаний у LPA корпусі з металевою кришкою[12]. Розміри датчика становлять 2.5 x 2.5 x 0.93 мм. Його невеликі габарити і невелике енергоспоживання дозволяють використовувати даний датчик у пристроях з акумуляторами. Також невеликі розміри впливають і на інертність показників датчика, що дає змогу швидше реагувати на зміни показів.



Рисунок 2.4. Зовнішній вигляд датчика BME280

Для комунікації з пристроями BME280 можна використовувати як SPI так і I2C інтерфейси.

Основні характеристики датчика:

- Інтерфейс підключення I2C та SPI
- Максимальна швидкість інтерфейсу I2C до 3.4МГц
- Межі вимірювання температури: від -40 до +85°C
- Точність вимірювання температури: від 0,5 до 1°C

| | | | | | | | | | |
|-----|------|----------|--------|------|-------------------|--|--|--|------|
| | | | | | | | | | Арк. |
| | | | | | | | | | 30 |
| Зм. | Арк. | № докум. | Підпис | Дата | 123.УДК:004:681.5 | | | | |

- Межі вимірювання вологості: від 0 до 100%
- Точність вимірювання вологості: 3%
- Межі вимірювання тиску: від 300 до 1100гПа
- Точність вимірювання тиску: 1гПа
- Напруга: 1,71 – 3,6В

Режими роботи

ВМЕ280 має 3 режими роботи: режим сну, примусовий та звичайний режим

Режим сну вводиться за замовчуванням після підключення живлення. Вбудований генератор скидання (POR) після підключення живлення, скидає логічну частину та регістрові значення цим самим підготовлює давач до подальшої роботи. У режимі сну вимірювання не проводяться, а споживання електроенергії є мінімальним. Усі регістри доступні. Ідентифікатор та коефіцієнти компенсацій можна зчитати. Спеціальних обмежень на інтерфейси немає.

Примусовий режим

У примусовому режимі проводиться одноразове вимірювання. Коли вимірювання закінчено, давач повертається у режим сну, а результати вимірювання можна отримати з регістрів даних. Для наступного вимірювання потрібно знову обрати примусовий режим та виконати вимірювання. Використання примусового режиму рекомендується для пристроїв які вимагають невеликої кількості вимірювань впродовж короткого проміжку часу.

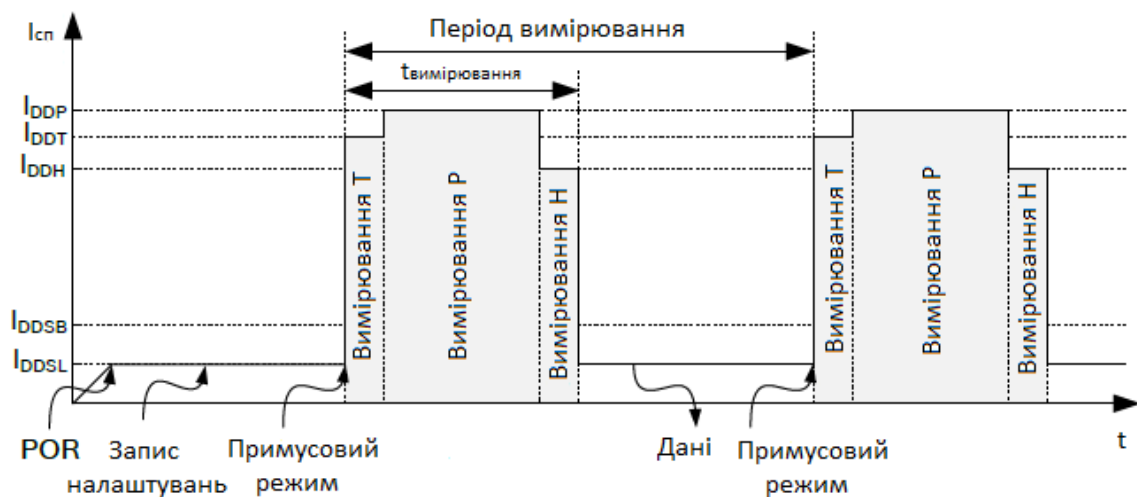


Рисунок 2.5. Часова діаграма роботи давача у примусовому режимі

Нормальний режим

Нормальний режим включає автоматизований вічний цикл між періодом вимірювання і періодом очікування. Вимірювання проводяться відповідно до обраних параметрів та фільтра. Час очікування задається як параметр. Після встановлення параметрів вимірювання і фільтра та ввімкнення нормального режиму, останні результати завжди можна отримати в реєстрах даних.

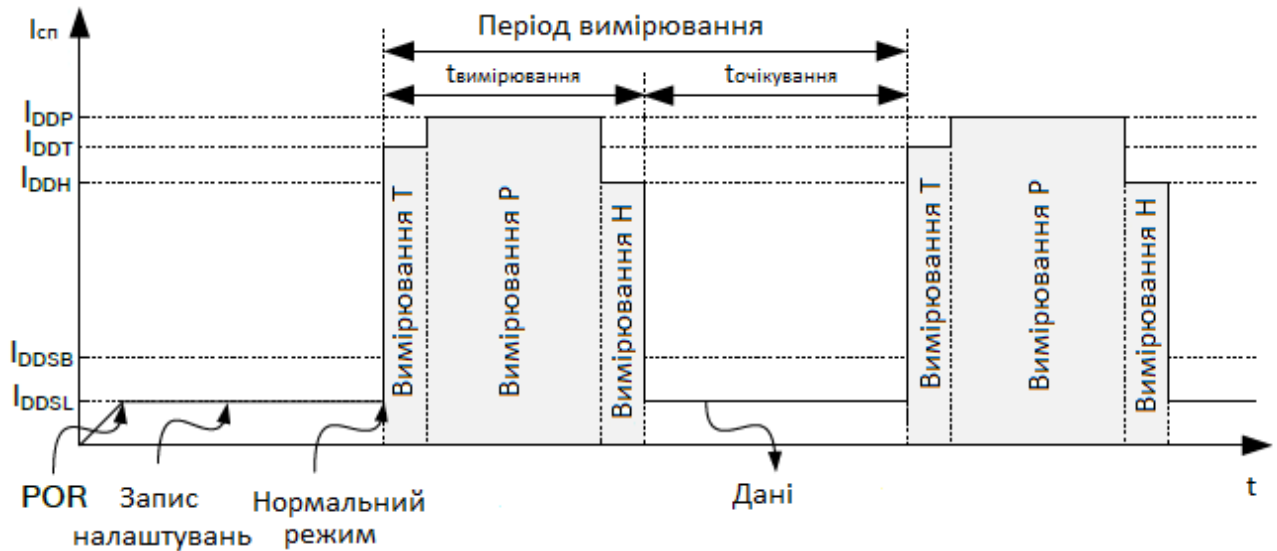


Рисунок 2.6. Часова діаграма роботи датчика у нормальному режимі

Як видно з часових діаграм під час вимірювання різних параметрів споживається різна кількість енергії

Таблиця 2.2. Енергоспоживання датчика ВМЕ280

| Параметр | Позначення | Споживання | Одиниці вимірювання |
|--------------------------------------|------------|------------|---------------------|
| Режим сну | I_{DDSL} | 0.1 | μA |
| Нормальний режим(під час очікування) | I_{DDSB} | 0.2 | μA |
| Вимірювання вологості | I_{DDH} | 340 | μA |
| Вимірювання тиску | I_{DDP} | 714 | μA |

| | | | |
|-------------------------|-----------|-----|---------|
| Вимірювання температури | I_{DDT} | 350 | μA |
|-------------------------|-----------|-----|---------|

Для моніторингу за погодою потрібна низька швидкість збору даних. Споживання енергії повинно бути мінімальним. Для цього потрібно запустити давач у примусовому режимі, при вимірюванні показників 1 раз в 1 хв. споживання буде рівне 0,16 мкА.

2.5. Вибір давача швидкості та напрямку вітру

Для визначення швидкості вітру у проєкті використовується чашковий анемометр. Дані анемометри є найпростіші у виробництві. Для реєстрування швидкості використовується оптичний опара та диск з щілиною. Вихід оптичного опари під'єднується до порту мікроконтролера. При спрацюванні мікроконтролеру потрібно відрахувати час до наступного імпульсу, і по заданій формулі обчислити швидкість вітру.

Для визначення напрямку вітру використовується флюгер. Для реєстрації напрямку як і для анемометра використовуються оптичні опари. Напрямок визначатиме абсолютний енкодер.

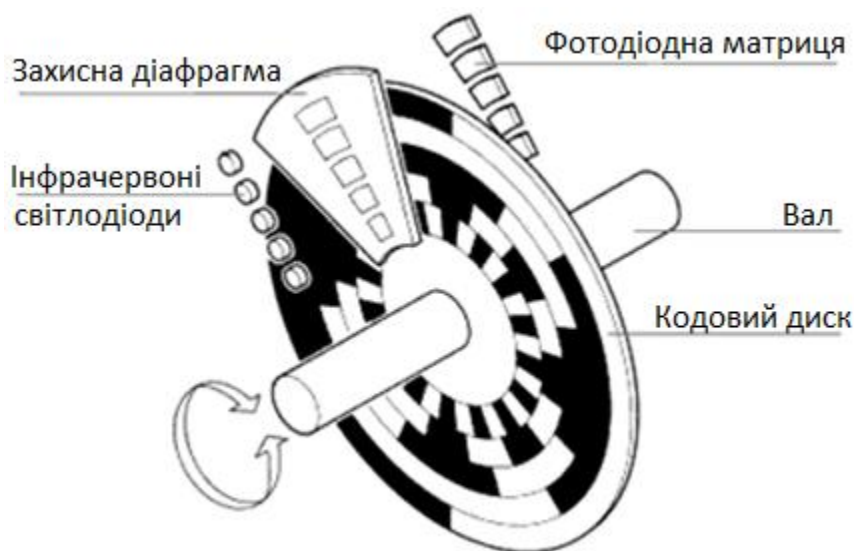


Рисунок 2.7. Абсолютний енкодер

Дані енкодери працюють не з двійковим кодом а з кодом Грея. Код Грея – одна із систем кодування інформації, в якій два послідовні коди відрізняються

значенням лише одного біта[13]. Кожний напрям можна позначити через двійковий код Грея. Для визначення напрямку вітру потрібно 3 бітовий абсолютний енкодер, оскільки для позначення сторін світу використовується 8 напрямів.

2.6. Вибір давача атмосферних опадів

Для визначення атмосферних опадів у проекті використовується давач дощу YL-83.

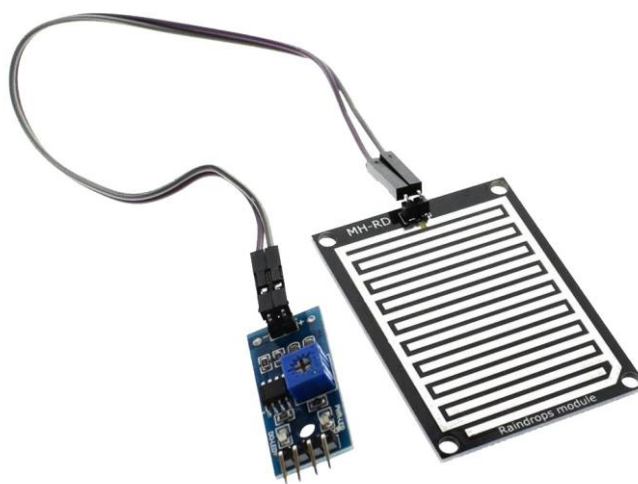


Рисунок 2.8. Давач атмосферних опадів YL-83

Сам давач складається з 2 частин: з чутливого до опадів елемента, та електронної частини. Електронна частина побудована на компараторі LM393D. Електронна частина перетворює аналоговий сигнал у цифровий. На платі розміщений потенціометр для встановлення порогового значення. Також на давач має і аналоговий вихід. Як правило опір чутливого елемента залежить від кількості крапель на його поверхні. Якщо елемент мокрий то опір зменшується, а вихідна напруга збільшується. Якщо ж елемент сухий, опір збільшується, а вихідна напруга зменшується. Даний модуль споживає 15мА.

2.7. Модуль заряду та захисту акумулятора

Для захисту акумулятора від пере розряду, перезаряду, короткого замикання використовується модуль TP4056. Даний модуль невеликий за розміром приблизно 27x17мм. Також модуль може заряджати акумулятори. Для цього на платі розміщений microUSB та 2 контактні площадки.

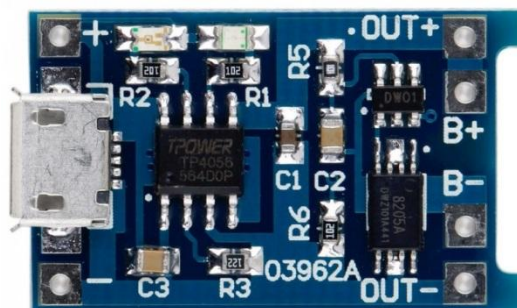


Рисунок 2.9. Модуль захисту акумулятора TP4056

Модуль відключає акумулятор, якщо напруга на акумуляторі становить 4.2В або 2.4В. Цим самим захищає акумулятор від перерозряду та перезаряду. В залежності від ємності акумулятора можна підібрати зарядний струм. Він задається резистором R3. Також модуль має світлову індикацію заряду. Для портативних пристроїв вхід даного модуля можна під'єднати до сонячних елементів. Таким чином можна заряджати акумулятор, цим самим збільшивши час автономної роботи.

| | | | | |
|-----|------|----------|--------|------|
| | | | | |
| Зм. | Арк. | № докум. | Підпис | Дата |

3. ПРОЕКТНИЙ РОЗДІЛ

Для виконання поставленого завдання створюється мобільний пристрій, який призначений для збору метеоданих і відправка їх за допомогою зв'язку LoRa. У даному розділі описується алгоритм роботи пристрою, розробка принципової схеми та друкованої плати, а також розрахунок параметрів деяких вузлів.

На рисунку 3.1 зображено загальну схему проектованого пристрою.

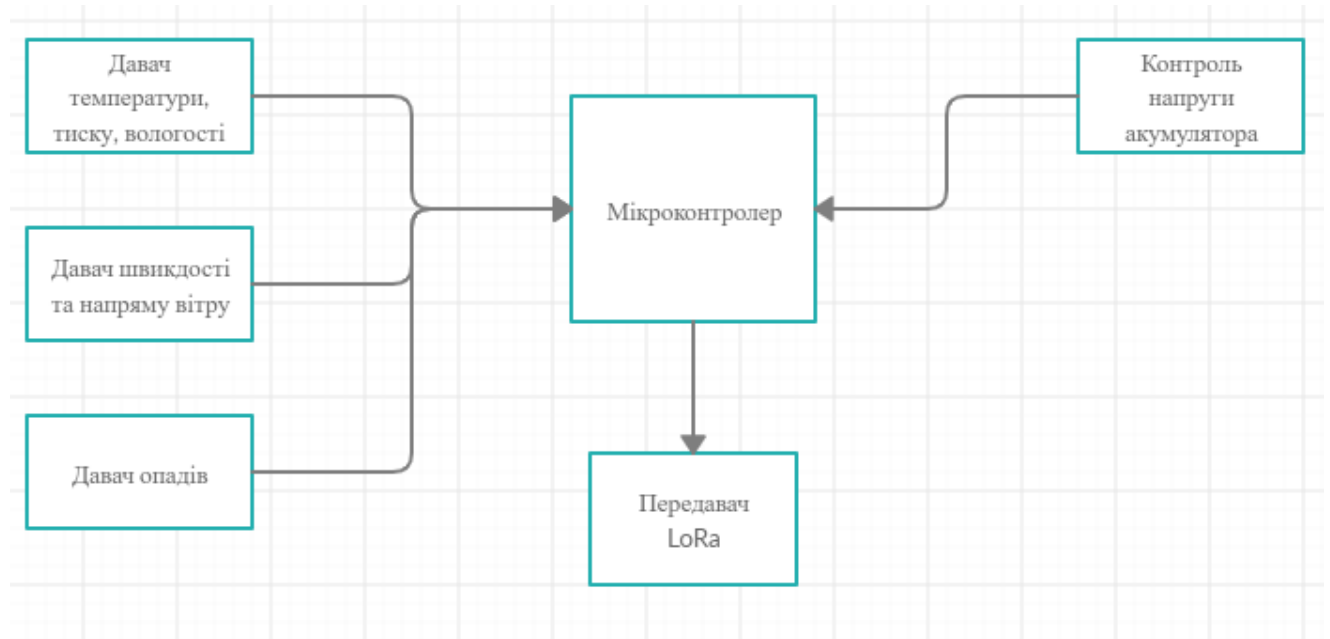


Рисунок 3.1. Загальна схема пристрою

Для отримання параметрів навколишнього середовища такі як температура, тиск та вологість використовується датчик ВМЕ280. Датчик є досить надійним та точним, з малим споживанням електроенергії. Датчик опадів та датчик швидкості та напрямку вітру є досить простими та надійними. Як і задумано отримані дані від датчиків передаються на приймач за допомогою модуля Ra-02. Після чого обробляються і виводяться користувачеві у зручний для нього формат.

3.1. Під'єднання основних вузлів до мікроконтролера

Схема електроживлення

Для нормальної роботи мікроконтролера потрібно джерело живлення на 3.3В. Діапазон вхідної напруги мікроконтролера по живленню становить 3.2-3.6В. Схема живлення мікроконтролера складається з стабілізатора AMS1117-3.3.

Для згладження пульсації стабілізатора використовується два конденсатора на вході та виході стабілізатора. На вхід стабілізатора можна подавати напругу до 18В. В пристрої буде використовуватись li-ion акумулятор. Максимальна напруга на акумуляторі становить 4.2В, що входить у діапазон допустимих напруг.

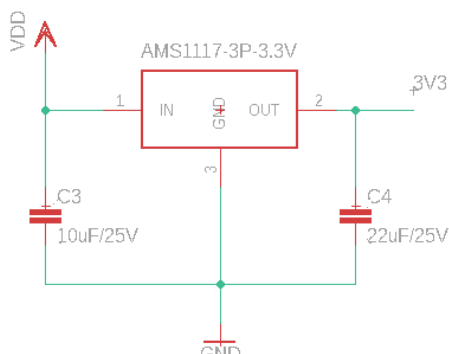


Рисунок 3.2. Перетворювач напруги AMS1117

Підключення датчиків температури тиску і вологості BME280. Для підключення датчиків до мікроконтролера використовуються апаратні I2C порти GPIO21 та GPIO22. Для стабільної роботи I2C між живленням та шиною підключають підтягуючі резистори.

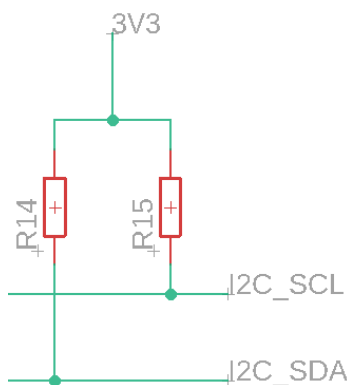


Рисунок 3.3. Схема підтягуючі резисторів

Номінали підтягуючі резисторів обчислюються за формулою[14]:

$$R_p = \frac{V_{cc} - 0.4}{3} = \frac{3.3 - 0.4}{0.003} = 966,667 \text{ Ом} \quad (3.1)$$

Згідно розрахунків для мікроконтролера ESP32 потрібно використовувати підтягуючі резистори на 1кОм.

Підключення датчиків напрямку та швидкості вітру. Для визначення напрямку вітру потрібно 3 порти ESP32 для ввімкнення інфрачервоних світло діодів та 3 порти

для отримання даних з фотодіодів. Для визначення швидкості вітру, необхідно 1 порт для отримання даних із фотодіода. Щоб не пошкодити порти мікроконтролера, на яких підключені світло діоди, потрібно використовувати обмежуючі резистори. Максимальний струм який може протікати через порт становить 20мА. За законом Ома:

$$R = \frac{U}{I} = \frac{3.3}{0.015} = 220 \text{ Ом} \quad (3.2)$$

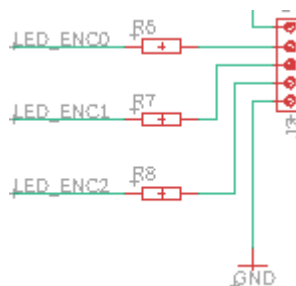


Рисунок 3.4. Схема обмежуючих резисторів

Згідно з розрахунків номінали обмежуючих резисторів повинні становити 220 Ом.

Модуль LoRa підключається до мікроконтролера ESP32 по інтерфейсу SPI. Інтерфейс SPI забезпечує високу швидкодію, та стійкий до поміх.

3.2. Розрахунок енергоспоживання пристрою

Так як пристрій буде живитись від акумулятора то споживання модулів потрібно звести до мінімуму.

Мікроконтролер ESP32 з вимкненим Wi-Fi та Bluetooth при низькій частоті споживає 3мА. Для зменшення енерговитрат доцільно використовувати режим сну. Тоді споживання зменшиться до 1мкА. Модуль LoRa Ra-02 в залежності від вихідної потужності для передачі сигналу може споживати від 20 до 120мА. Споживання давача ВМЕ 280 є достатньо малим і становить 1.5мА. Інфрачервоні світло діоди споживають приблизно 1мА. Тому їх можна вмикати тільки при визначення напрямку вітру. Світлодіод який визначає швидкість вітру повинен бути увімкненим постійно. Давай опадів споживає 15мА. Тому його як і давач

напрямку вітру доцільно вмикати при вимірювання. У таблиці показано споживання пристрою під час передачі даних на сервер.

Таблиця 3.1. Енергоспоживання пристрою під час збору та передачі даних

| Модуль | Споживана потужність | Одиниці вимірювання |
|-----------------------|----------------------|---------------------|
| ESP32 | 3 | мА |
| LoRa Ra-02 | 20 | мА |
| BME280 | 1.5 | мА |
| Давач швидкості вітру | 1 | мА |
| Давач напрямку вітру | 3 | мА |
| Давач опадів YL-83 | 15 | мА |
| Разом | 43.5 | мА |

У наступній таблиці показано енергоспоживання під час сну пристрою

Таблиця 3.2. Енергоспоживання пристрою у режимі сну

| Модуль | Споживана потужність | Одиниці вимірювання |
|-----------------------|----------------------|---------------------|
| ESP32 | 10 | мкА |
| LoRa Ra-02 | 1 | мкА |
| BME280 | 0.1 | мкА |
| Давач швидкості вітру | 1 | мА |
| Давач напрямку вітру | 0 | мкА |
| Давач опадів YL-83 | 0 | мкА |
| Разом | 1,011 | мА |

3.3 Опис алгоритму роботи програми

Алгоритм роботи програми зображено на аркуші 5. Алгоритм є не складним для розуміння. Спочатку відбувається ініціалізація модулів які під'єднанні до мікроконтролера. Під час ініціалізації визначаються порти I2C, SPI, призначаються порти для отримання сигналів з опт опар.

i2c = I2C(scl=Pin(22), sda=Pin(21))

```
pin_rain = Pin(4, Pin.IN, Pin.PULL_UP)
```

Відбувається ініціалізація АЦП для отримання значення напруги акумулятора та отримання значення з датчика дощу. Для модуля LoRa Ra-02 задається частота передачі, рівень потужності сигналу, ширина каналу.

```
sx127x.SX127x(name = 'SX127x',
```

```
parameters = {'frequency': 433E6, 'tx_power_level': 2, 'signal_bandwidth': 125E3, 'spreading_factor': 8, 'coding_rate': 5, 'preamble_length': 8, 'implicitHeader': False, 'sync_word': 0x12, 'enable_CRC': False}, onReceive = None)
```

Відповідно для модулів Ra-02 та BME280 задаються спеціальні числа, які передаються у вигляді команд.

Після того як відбулася ініціалізація перевіряється рівень заряду акумулятора. Якщо рівень заряду акумулятора низький то формується відповідне повідомлення. Надсилання повідомлення відбуватиметься доти, доки заряд акумулятора не буде критичний тоді весь пристрій ввійде у режим глибокого сну. Якщо рівень заряду акумулятора достатній для роботи, пристрій збирає всі дані. Дані про швидкість вітру збираються та усереднюються протягом 10хв. Також у пакет даних що передається входить і максимальне значення швидкості вітру. Наступним процесом буде збирання даних з датчика BME280. Отримані значення температури, тиску та вологості, та значення з датчика опадів. Всі дані складають у пакет даних, в яких надсилається номер станції, температура, тиск та вологість, значення опадів, середня швидкість вітру, максимальне значення швидкості вітру, та напрям вітру. Даний пакет передається за допомогою модуля LoRa Ra-02, після чого пристрій входить у режим сну.

Обчислення швидкості вітру

Для визначення швидкості вітру з точки зору програмної реалізації використовуються зовнішні переривання. Для цього необхідно призначити порт на якому буде відбуватися переривання. У MicroPython дана операція виглядатиме так:

```
imp = Pin(5, Pin.IN)
```

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------------|
| | | | | | 123.УДК:004:681.5 | Арк. 40 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

Після призначення ніжки на вхід, потрібно вказати мікроконтролеру що ця ніжка буде використовуватись для зовнішнього переривання:

```
imp.irq(trigger=Pin.IRQ_RISING, handler=handle_interrupt)
```

Даним рядком коду ми також визначаємо функцію, яка буде обробляти зовнішнє переривання.

```
def handle_interrupt(pin):  
    global start  
    global width_impuls  
    global state  
    global speed  
    if state == False:  
        start = time.ticks_ms()  
        print("start", start)  
        state = True  
    else:  
        width_impuls_2 = time.ticks_ms()  
        width_impuls = time.ticks_diff(width_impuls_2, start)  
        start = width_impuls_2  
        speed = (1/width_impuls)*1000
```

У функції переривання знаходяться 4 глобальні змінні start, width_impuls, state, speed. Змінна start призначення для початкового запам'ятовування часу, тобто ця змінна буде початком відліку. Змінна width_impuls використовується для зберігання часу між двома імпульсами. Змінна state – це прапор стану, якщо почалось вимірювання state = 1, якщо ж ні state = 0. Для фіксації закінчення імпульсу використовується змінна width_state_2. В ній записується значення кінця імпульсу який стане початком для іншого відліку часу. Width_impuls є різницею між початковим і кінцевим значенням та обчислюється вбудованою функцією ticks_diff().

Отримання даних про температуру тиск та вологість забезпечує бібліотека BME280. Для отримання значень потрібно виконати команди bme.values[0],

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 41 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

bme.values[1], bme.values[2]. Відповідно у bme.values[0] зберігається значення температури, bme.values[1] зберігається значення тиску, bme.value[3] зберігає значення вологості. Так як давач опадів працює в цифровому коді, то можна з легкістю зчитувати стан відповідного порта.

Для зчитування напрямку вітру використовується 3 оптопари та абсолютний енкодер. Кожному значенню коду відповідає сторона світу. Відповідність коду до сторони світу зображена у таблиці.

Таблиця 3.3. Позначення напрямку вітру

| Код | Сторона світу | Позначення у пакеті даних |
|-----|-----------------|---------------------------|
| 000 | Північ | 1 |
| 001 | Північний-Захід | 2 |
| 011 | Захід | 3 |
| 010 | Південий-Захід | 4 |
| 110 | Південь | 5 |
| 111 | Південний-Схід | 6 |
| 101 | Схід | 7 |
| 100 | Північний-Схід | 8 |

В результаті, усі отримані дані формуються у певному порядку: 1-номер станції, 2-температура, 3-тиск, 4-вологість, 5-швидкість вітру, 6-опаді. Наприклад: St12, 23.2, 983.4, 35.6, 2.2, 2, 0. Відповідно дані отримані з 12 метеостанції, температура 23.2, тиск 983.4 гПа, вологість 35.6%, швидкість вітру 2.2м/с, вітер північно-західний, опади відсутні. Даний запис зручний тим, що його можна легко опрацювати регулярними виразами.

Отримання значення напруги на акумуляторі. Для отримання значення з акумулятора використовується АЦП мікроконтролера. Згідно з документацією, на аналоговий вхід мікроконтролера повинна подаватись напруга не вище 3.6В. Напруга на зарядженому акумуляторі становить 4.2В. Така напруга може пошкодити АЦП мікроконтролера. Для вирішення цієї проблеми використовують дільник напруг. Завдяки дільнику напруг можна зменшити

рівень напруги, який буде придатний для вимірювання. Значення на виході дільника напруг обчислюється за формулою 3.3.

$$V_{out} = \frac{R_2}{R_1 + R_2} * V_{in} \quad (3.3)$$

де, V_{out} – вихідна напруга, V_{in} – вхідна напруга, R_1, R_2 – опори резисторів.

У схемі значення $R_1 = 100\text{кОм}$, $R_2 = 10\text{кОм}$. Напруга зарядженого акумулятора становить 4.2В, а розрядженого 3.2В. Критичне значення для акумулятора 2.9В.

Для зарядженого акумулятора значення на виході дільника напруг буде:

$$V_{зар} = \frac{10}{100 + 10} * 4.2 = \frac{10}{110} * 4.2 = 0.38\text{В}$$

Для розрядженого акумулятора значення буде:

$$V_{роз} = \frac{10}{100 + 10} * 3.2 = \frac{10}{110} * 3.2 = 0.3\text{В}$$

Для критичного значення рівня заряду акумулятора:

$$V_{роз} = \frac{10}{100 + 10} * 2.9 = \frac{10}{110} * 2.9 = 0.26\text{В}$$

Стандартна максимальна напруга на вході АЦП ESP32 становить 1В. Тобто мікроконтролер розбиває діапазон від 0 до 1В на 4096 частинок. Відповідно значення напруг буде:

Таблиця 3.4. Значення вхідної напруги після обробки

| Значення у вольтах, В | Цифровий код АЦП |
|-----------------------|------------------|
| 4.2 | 1556 |
| 3.2 | 1229 |
| 2.9 | 1065 |

Відповідно до таблиці якщо рівень напруги опуститься нижче 3.2В або значення з АЦП буде нижче 1229, пристрій повинен подати повідомити про низький заряд акумулятора. Для цього потрібно призначити відповідний порт АЦП. У MicroPython це робить наступний рядок:

```
bat_lev = ADC(Pin(32))
```

Після призначення порта, функція `battery_lev` перевіряє рівень заряду акумулятора, якщо він опуститься відповідного рівня, відповідне повідомлення буде надіслано центральному серверу.

```
def battery_level():  
    state = 0  
    bat_pin.on()  
    level = bat_lev.read()  
    bat_pin.off()  
    if level <= 1229:  
        state = 1  
        return state  
    else if level <= 1065:  
        state = 2  
        return state  
    else:  
        return state
```

Значення `state = 1` відповідає опущення рівня напруги до 3.2В, а `state = 2` до 2.9В.

Для того щоб дільник напруги не розряджав акумулятор використовується 2 польових транзистора, які будуть працювати як вимикач.

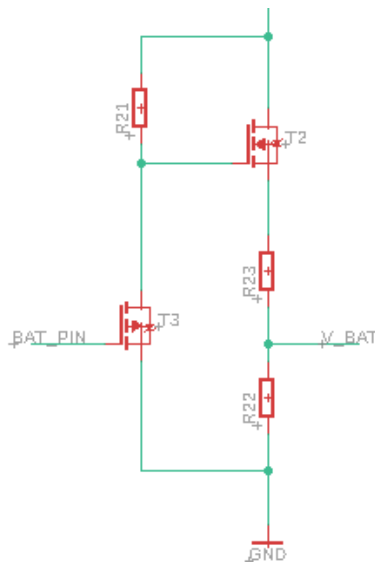


Рисунок 3.5. Схема підключення дільника напруги

Після того як мікроконтролер “прокинеться”, для отримання значення рівня заряду акумулятора, чи отримання значення з датчика опадів, потрібно на відповідні порти подати високий рівень. Тоді польові транзистори відкриються, після чого відбудеться отримання значень, і на відповідні порти подадуть низький рівень, для економії електроенергії. Рядок `bat_pin.on()` вмикає коло з дільником напруги. Після чого отримується значення напруги акумулятора `level = bat_lev.read()`. Зробивши виміри потрібно відключити коло з дільником напруги `bat_pin.off()`. Відповідно до значення напруги на центральний сервер передається відповідне повідомлення.
необхідну інформацію.

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 45 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

ЕКОНОМІНА ЧАСТИНА

Метою бакалаврської роботи є розробка портативної метеостанції з передачею даних на великих відстанях.

Даний пристрій, призначений для відслідковування за параметрами погоди, та передачею отриманих даних на центральний сервер, для подальшої обробки. Сфера застосування цього пристрою є досить різноманітна.

Такі метеостанції можуть використовуватись:

- Сільському господарстві
- Метеопрогнозі
- Аеропортах
- Домашнє застосування
- Лісництві
- Курорти, туристичні бази
- Екологічного контролю

Для конструювання пристрою було обрано мікроконтролер ESP32 компанії Espressif та модуль LoRa Ra-02 компанії Ai-Thinker. Вибір мікроконтролера зводиться до особистих бажань розробника оскільки порівняння мікроконтролера є недоцільним, так як існує безліч таких, які повністю відповідають поставленим вимогами та цілям даної кваліфікаційної роботи. В якості замінюваного аналога для порівняння обрано метеостанцію Vantage PRO2 plus від компанії Davis. Дана метеостанція здатна вимірювати такі параметри як температура, тиск, вологість, швидкість та напрям вітру. Безпроводна версія може передавати дані на відстань до 300м.

Точність вимірювання розробленого пристрою зводиться до точності давачів, які використовуються.

Таблиця 4.1. Порівнянні характеристики пристроїв

| Параметр | Vantage PRO2 plus | Розроблювана система |
|-----------------------|-------------------|----------------------|
| Температура(діапазон) | -40+65 | -40+85 |
| Температура(точність) | +/-0,5 | +/-0,5 |

| | | |
|---------------------------|-------------|-------------|
| Вологість(точність) | 3% | 3% |
| Тиск(діапазон) | 300-1100гПа | 300-1100гПа |
| Тиск(точність) | 1гПа | 1гПа |
| Дальність зв'язку | 300м | До 10км |
| Швидкість вітру(діапазон) | 1-50м/с | 1-50м/с |

Для порівняння пристроїв доцільно обчислити загальну вартість розроблюваного пристрою. Ціна пристрою зводиться до ціни комплектуючих. У таблиці 4.2 вказані ціни на комплектуючі.

Таблиця 4.2. Вартість комплектуючих виробів приладу

| Комплектуючі вироби | Кількість, шт. | Вартість за одиницю, грн | Сума, грн |
|----------------------|----------------|--------------------------|-----------|
| Мікроконтролер ESP32 | 1 | 125.7 | 125.7 |
| LoRa Ra-02 | 1 | 130.0 | 130.0 |
| Давач опадів YL-83 | 1 | 40.3 | 40.3 |
| ВМЕ280 | 1 | 91.0 | 91.0 |
| AMS1117-3.3 | 2 | 2.0 | 4.0 |
| Антенa | 1 | 37.0 | 37.0 |
| Акумулятор li-ion | 1 | 85.7 | 85.7 |
| | | | 513.7 |

Вартість комплектуючих для виготовлення пристрою становить 513.7 грн. Порівняно з метеостанцією Vantage PRO 2 plus, розроблений пристрій при тій же точності та діапазонах робіт є дешевшою у 25 раз.

ВИСНОВОК

В результаті виконання роботи розроблено пристрій для відслідковування за метеорологічними показниками, з передачею цих даних на центральний сервер. Даний пристрій є мобільний, з великим радіусом передачі даних, та з низьким енергоспоживанням. Пристрій побудований на мікроконтролері ESP32, потужності якого стає для обробки отриманої інформації, її упорядкуванням та підготовкою до відправлення. У якості радіо модуля використовується LoRa Ra-02, завдяки якому досягається велика дальність передачі. Розроблено алгоритм програми, який збирає, обробляє, та пересилає дані на центральний сервер. Проведено порівняння проектного рішення з пристроєм від фірми Dalas. В результаті при однакових діапазонах вимірювальних величин та при однакових допустимих похибках, розроблений пристрій виявився набагато дешевший.

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| | | | | | | 48 |
| Зм. | Арк. | № докум. | Підпис | Дата | | |

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. https://habr.com/ru/company/ru_mts/blog/430496/
2. <https://www.ericsson.com/en/blog/2019/10/what-is-NB-IoT>
3. <https://habr.com/ru/company/realtrac/blog/304312/>
4. <https://habr.com/ru/company/unwds/blog/396869/>
5. <https://uk.wikipedia.org/wiki/Термопара>
6. <https://uk.wikipedia.org/wiki/Терморезистор>
7. <http://elektrik.info/main/automation/1083-datchiki-vlazhnosti-kak-ustroeny-i-rabotayut.html>
8. <https://simvolt.ua/anemometr-prilad-dlya-viznachennya-shvidkost-ta-napryamku-ruhu-potoku.html>
9. <https://micropython.org/>
10. <http://micpic.ru/home/proekty-na-esp32/194-opisanie-mikrokontrollera-esp32.html>
11. <https://www.mini-tech.com.ua/lora-ra-02>
12. <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/>
13. https://ru.wikipedia.org/wiki/Код_Грея
14. <http://www.ti.com/lit/an/slva689/slva689.pdf>

| | | | | | | |
|------------|-------------|-----------------|---------------|-------------|--------------------------|-------------|
| | | | | | <i>123.УДК:004:681.5</i> | <i>Арк.</i> |
| <i>Зм.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 49 |

```
Controller.py
from time import sleep

class Controller:
    class Mock:
        pass

    ON_BOARD_LED_PIN_NO = None
    ON_BOARD_LED_HIGH_IS_ON = True
    GPIO_PINS = []

    PIN_ID_FOR_LORA_RESET = None

    PIN_ID_FOR_LORA_SS = None
    PIN_ID_SCK = None
    PIN_ID_MOSI = None
    PIN_ID_MISO = None

    PIN_ID_FOR_LORA_DIO0 = None
    PIN_ID_FOR_LORA_DIO1 = None
    PIN_ID_FOR_LORA_DIO2 = None
    PIN_ID_FOR_LORA_DIO3 = None
    PIN_ID_FOR_LORA_DIO4 = None
    PIN_ID_FOR_LORA_DIO5 = None

    def __init__(self,
                 pin_id_led = ON_BOARD_LED_PIN_NO,
                 on_board_led_high_is_on = ON_BOARD_LED_HIGH_IS_ON,
                 pin_id_reset = PIN_ID_FOR_LORA_RESET,
                 blink_on_start = (2, 0.5, 0.5)):

        self.pin_led = self.prepare_pin(pin_id_led)
        self.on_board_led_high_is_on = on_board_led_high_is_on
        self.pin_reset = self.prepare_pin(pin_id_reset)
        self.reset_pin(self.pin_reset)
        self.spi = self.prepare_spi(self.get_spi())
        self.transceivers = { }
        self.blink_led(*blink_on_start)

    def add_transceiver(self,
                       transceiver,
```

| | | | | | | | | | |
|-----|------|----------|--------|------|--|--|--|--|------|
| | | | | | | | | | Арк. |
| | | | | | | | | | 50 |
| Зм. | Арк. | № докум. | Підпис | Дата | | | | | |

```

pin_id_ss = PIN_ID_FOR_LORA_SS,
pin_id_RxDone = PIN_ID_FOR_LORA_DIO0,
pin_id_RxTimeout = PIN_ID_FOR_LORA_DIO1,
pin_id_ValidHeader = PIN_ID_FOR_LORA_DIO2,
pin_id_CadDone = PIN_ID_FOR_LORA_DIO3,
pin_id_CadDetected = PIN_ID_FOR_LORA_DIO4,
pin_id_PayloadCrcError = PIN_ID_FOR_LORA_DIO5):

transceiver.transfer = self.spi.transfer
transceiver.blink_led = self.blink_led

transceiver.pin_ss = self.prepare_pin(pin_id_ss)
transceiver.pin_RxDone = self.prepare_irq_pin(pin_id_RxDone)
transceiver.pin_RxTimeout = self.prepare_irq_pin(pin_id_RxTimeout)
transceiver.pin_ValidHeader = self.prepare_irq_pin(pin_id_ValidHeader)
transceiver.pin_CadDone = self.prepare_irq_pin(pin_id_CadDone)
transceiver.pin_CadDetected = self.prepare_irq_pin(pin_id_CadDetected)
transceiver.pin_PayloadCrcError = self.prepare_irq_pin(pin_id_PayloadCrcError)

transceiver.init()
self.transceivers[transceiver.name] = transceiver
return transceiver

def prepare_pin(self, pin_id, in_out = None):
    reason = ""
    # a pin should provide:
    # .pin_id
    # .low()
    # .high()
    # .value() # read input.
    # .irq() # (ESP8266/ESP32 only) ref to the irq function of real pin object.
    ""
    raise NotImplementedError(reason)

def prepare_irq_pin(self, pin_id):
    reason = ""
    # a irq_pin should provide:
    # .set_handler_for_irq_on_rising_edge() # to set trigger and handler.
    # .detach_irq()
    ""
    raise NotImplementedError(reason)

def get_spi(self):
    reason = ""

```

| | | | | | | | |
|-----|------|----------|--------|------|--|-------------------|------|
| | | | | | | 123.УДК:004:681.5 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | | 51 |

```

    # initialize SPI interface
    ""
    raise NotImplementedError(reason)

def prepare_spi(self, spi):
    reason = ""
    # a spi should provide:
    # .close()
    # .transfer(pin_ss, address, value = 0x00)
    ""
    raise NotImplementedError(reason)

def led_on(self, on = True):
    self.pin_led.high() if self.on_board_led_high_is_on == on else self.pin_led.low()

def blink_led(self, times = 1, on_seconds = 0.1, off_seconds = 0.1):
    for i in range(times):
        self.led_on(True)
        sleep(on_seconds)
        self.led_on(False)
        sleep(off_seconds)

def reset_pin(self, pin, duration_low = 0.05, duration_high = 0.05):
    pin.low()
    sleep(duration_low)
    pin.high()
    sleep(duration_high)

def __exit__(self):
    self.spi.close()

```

Controller_esp.py

```

from machine import Pin, SPI, reset
import controller

```

```

class Controller(controller.Controller):

```

```

    # LoRa config
    PIN_ID_FOR_LORA_RESET = 4

    PIN_ID_FOR_LORA_SS = 15
    PIN_ID_SCK = 14
    PIN_ID_MOSI = 13
    PIN_ID_MISO = 12

```

```
PIN_ID_FOR_LORA_DIO0 = 5
PIN_ID_FOR_LORA_DIO1 = None
PIN_ID_FOR_LORA_DIO2 = None
PIN_ID_FOR_LORA_DIO3 = None
PIN_ID_FOR_LORA_DIO4 = None
PIN_ID_FOR_LORA_DIO5 = None
```

```
# ESP config
```

```
ON_BOARD_LED_PIN_NO = 2
ON_BOARD_LED_HIGH_IS_ON = True
GPIO_PINS = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
             12, 13, 14, 15, 16, 17, 18, 19, 21, 22,
             23, 25, 26, 27, 32, 34, 35, 36, 37, 38, 39)
```

```
def __init__(self,
             pin_id_led = ON_BOARD_LED_PIN_NO,
             on_board_led_high_is_on = ON_BOARD_LED_HIGH_IS_ON,
             pin_id_reset = PIN_ID_FOR_LORA_RESET,
             blink_on_start = (2, 0.5, 0.5)):
```

```
    super().__init__(pin_id_led,
                    on_board_led_high_is_on,
                    pin_id_reset,
                    blink_on_start)
```

```
def prepare_pin(self, pin_id, in_out = Pin.OUT):
```

```
    if pin_id is not None:
        pin = Pin(pin_id, in_out)
        new_pin = Controller.Mock()
        new_pin.pin_id = pin_id
        new_pin.value = pin.value
```

```
    if in_out == Pin.OUT:
        new_pin.low = lambda: pin.value(0)
        new_pin.high = lambda: pin.value(1)
    else:
        new_pin.irq = pin.irq
```

```
    return new_pin
```

```
def prepare_irq_pin(self, pin_id):
```

```
    pin = self.prepare_pin(pin_id, Pin.IN)
    if pin:
```

```

pin.set_handler_for_irq_on_rising_edge = lambda handler: pin.irq(handler =
handler,
                                trigger = Pin.IRQ_RISING)
pin.detach_irq = lambda: pin.irq(handler = None, trigger = 0)
return pin

def get_spi(self):
    spi = None
    id = 1
    try:
        spi = SPI(id, baudrate = 1000000, polarity = 0, phase = 0, bits = 8, firstbit =
SPI.MSB,
                sck = Pin(self.PIN_ID_SCK, Pin.OUT, Pin.PULL_DOWN),
                mosi = Pin(self.PIN_ID_MOSI, Pin.OUT, Pin.PULL_UP),
                miso = Pin(self.PIN_ID_MISO, Pin.IN, Pin.PULL_UP))
    spi.init()

    except Exception as e:
        print(e)
        if spi:
            spi.deinit()
            spi = None
        reset()

    return spi

def prepare_spi(self, spi):
    if spi:
        new_spi = Controller.Mock()

def transfer(pin_ss, address, value = 0x00):
    response = bytearray(1)

    pin_ss.low()

    spi.write(bytes([address]))
    spi.write_readinto(bytes([value]), response)

    pin_ss.high()

    return response

new_spi.transfer = transfer

```

| | | | | | | | |
|-----|------|----------|--------|------|--|-------------------|------|
| | | | | | | 123.УДК:004:681.5 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | | 54 |

```
new_spi.close = spi.deinit
return new_spi
```

```
def __exit__(self):
    self.spi.close()
```

LoRaSender.py

```
from time import sleep
from machine import Pin, I2C, ADC
import bme280
import time
```

```
station = "1"
i2c = I2C(scl=Pin(22), sda=Pin(21))
bme = bme280.BME280(i2c=i2c)
pin_rain = Pin(4, Pin.IN, Pin.PULL_UP)
rain_pin = Pin(16, Pin.OUT)
bat_pin = Pin(23, Pin.OUT)
bat_lev = ADC(Pin(32))
start = 0
width_impuls = 0
width_impuls_2 = 0
speed = 0
state = False
bat_pin.off()
rain_pin.off()
```

```
def handle_interrupt(pin):
    global start
    global width_impuls
    global state
    global speed
    if state == False:
        start = time.ticks_ms()
        print("start", start)
        state = True
    else:
        width_impuls_2 = time.ticks_ms()
        width_impuls = time.ticks_diff(width_impuls_2, start)
        start = width_impuls_2
        speed = (1/width_impuls)*1000
```

```
impuls = Pin(4, Pin.OUT)
imp = Pin(5, Pin.IN)
```

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 55 |

```

imp.irq(trigger=Pin.IRQ_RISING, handler=handle_interrupt)
encod0 = Pin(25, Pin.IN)
encod1 = Pin(26, Pin.IN)
encod2 = Pin(27, Pin.IN)

w_speed = 0
m_speed = 0

def max_speed(speed):
    if speed > m_speed:
        m_speed = speed
    return m_speed

def wind_speed(speed):
    global w_speed
    w_speed = (w_speed + speed)/2
    return w_speed

def rain():
    rain_pin.on()
    a = 0
    if pin_rain.value() == '1':
        a = 1.0
    else:
        a = 0.0
    rain_pin.off()
    return a

def battery_level():
    state = 0
    bat_pin.on()
    level = bat_lev.read()
    bat_pin.off()
    if level <= 1229:
        state = 1
        return state
    else if level <= 1065:
        state = 2
        return state
    else:
        return state

def direction():
    if encod0.value() == 0 and encod1.value() == 0 and encod2.value() == 0:

```

| | | | | | | |
|------------|-------------|-----------------|---------------|-------------|--------------------------|-------------|
| | | | | | <i>123.УДК:004:681.5</i> | <i>Арк.</i> |
| <i>Зм.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | 56 |


```
    return "1.0"#N
if encod0.value()==0 and encod1.value()==0 and encod2.value()==1:
    return "2.0"#NW
if encod0.value()==0 and encod1.value()==1 and encod2.value()==1:
    return "3.0"#W
if encod0.value()==0 and encod1.value()==1 and encod2.value()==0:
    return "4.0"#SW
if encod0.value()==1 and encod1.value()==1 and encod2.value()==0:
    return "5.0"#S
if encod0.value()==1 and encod1.value()==1 and encod2.value()==1:
    return "6.0"#SE
if encod0.value()==1 and encod1.value()==0 and encod2.value()==1:
    return "7.0"#E
if encod0.value()==1 and encod1.value()==0 and encod2.value()==0:
    return "8.0"#NE

def send(lora):
    print("LoRa Sender")

    while True:
        payload =
'St:{0},T:{1},P:{2},H:{3},WS:{4},WD:{5},R:{6},A:{7}'.format(station,
bme.values[0],
                                bme.values[1], bme.values[2],
                                wind_speed(speed), direction(),
                                rain(), battery_level())

        time.sleep(600)
        print("Sending packet: \n{ }\n".format(payload))
        lora.println(payload)

Tx_driver.py
import sx127x
import time
from controller_esp import Controller
import LoRaSender

def main():

    controller = Controller()

    sx127x.SX127x(name = 'SX127x',
        parameters = {'frequency': 433E6, 'tx_power_level': 2, 'signal_bandwidth': 125E3,
'spreading_factor': 8, 'coding_rate': 5, 'preamble_length': 8, 'implicitHeader': False,
'sync_word': 0x12, 'enable_CRC': False}, onReceive = None)
```

| | | | | | | |
|------------|-------------|-----------------|---------------|-------------|--------------------------|------|
| | | | | | <i>123.УДК:004:681.5</i> | Арк. |
| | | | | | | 57 |
| <i>Зм.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> | | |

```

lora = controller.add_transceiver(sx127x.SX127x(name = 'LoRa'),
    pin_id_ss = Controller.PIN_ID_FOR_LORA_SS,
    pin_id_RxDone = Controller.PIN_ID_FOR_LORA_DIO0)
print("LoRa Sender")
LoRaSender.send(lora)
if __name__ == '__main__':
    main()

```

Інші файли необхідні для роботи пристрою розміщені на CD диску.

| | | | | | | |
|-----|------|----------|--------|------|-------------------|------|
| | | | | | 123.УДК:004:681.5 | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата | | 58 |