

Міністерство освіти і науки України
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри,
канд. техн. наук, доцент
_____ Я. М. Крайник
« __ » _____ 2022 р.

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА
**МЕРЕЖА ЗБОРУ ІНФОРМАЦІЇ ДЛЯ ПОСТАЧАЛЬНИКІВ
ЕНЕРГОРЕСУРСІВ В СИСТЕМІ SMART CITY**

Спеціальність «Комп'ютерна інженерія»
123 – КМР.1 – 605. 21510520

Студент _____ О. О. Стрельчук
підпис
« __ » _____ 2022 р.

Керівник канд. техн. наук, доцент _____ Л. В. Солобуто
підпис
« __ » _____ 2022 р.

Миколаїв 2022

ЗАВДАННЯ

на виконання бакалаврської роботи

НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!!!!!!

ЗАРЕЗЕРВОВАНА Сторінка 1

ця сторінка після друку буде замінена

ЗАВДАННЯ

на виконання бакалаврської роботи

НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!

ЗАРЕЗЕРВОВАНА Сторінка 2

ця сторінка після друку буде замінена

АНОТАЦІЯ

1 сторінка !!!!

НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!!!!!!

ЗАРЕЗЕРВОВАНА Сторінка 1

ця сторінка після друку буде замінена

ABSTRACT

1 сторінка !!!!

НЕ ВИДАЛЯТИ цю СТОРІНКУ з файлу !!!!!!!!!!!!!!!!!!!!!

ЗАРЕЗЕРВОВАНА Сторінка 2

ця сторінка після друку буде замінена

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП	9
РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ТА ПАТЕНТНОЇ ІНФОРМАЦІЇ	13
1.1 Ключові енергетичні дані.....	13
1.2 Розумне місто: що це таке і навіщо воно потрібне?.....	14
1.3 Розумні міста: компоненти та характеристики.....	16
1.4 Розумна енергія	22
1.5 Smart Grid.....	27
Висновки до розділу 1	28
РОЗДІЛ 2 ХАРАКТЕРИСТИКА ДОСЛІДЖУВАНОВОГО ОБ'ЄКТА	30
2.1 Огляд подібних технологій: Smart Grid.....	30
2.2 Використання подібних технологій у світі	33
2.6 Оптимізація ланцюга поставок енергоносіїв	37
2.7 Інформаційні технології в проектуванні та експлуатації енергетичних систем.....	39
2.8 Огляд технології LoRa.....	41
2.8.1 Сильні та слабкі сторони технології	42
2.8.2 Класи пристроїв LoRa	43
2.8.3 Фізичний рівень (PHY Layer)	44
2.8.4 Підтвердження отримання повідомлень	45
2.8.5 Адаптивна швидкість передачі (Adaptive Data Rate – ADR)...	46
2.8.6 Синхронізація приймача та передавача.....	47
2.8.7 Детектування CSS сигналу	49
Висновки до розділу 2	55
РОЗДІЛ 3 РОЗРОБКА МОДЕЛІ ТА СПОСОБИ ТЕСТУВАННЯ	56
3.1 Способи тестування програмного забезпечення	56
3.2 Обґрунтування обраних елементів	60

3.2.1 RYLR896.....	60
3.2.2. Arduino IDE.....	62
3.3 Процес розробки.....	64
3.3.1 Поєднання RYLR890/RYLR896 Модуль LoRa з Arduino.....	64
3.3.2 Вихідний код програми.....	69
3.3.3 Реалізація для умов Smart City	72
3.3.4 Програмування кількох передавачів Lora Arduino.....	74
3.3.5 Програмування приймача	74
Висновки до розділу 3	78
ВИСНОВКИ.....	79
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	81
ДОДАТОК А КОД ARDUINO ПЕРЕДАВАЧА	84
ДОДАТОК Б КОД ПРИЙМАЧА	87

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

GND	–	Ground
IDE	–	Integrated Development Environment (укр. – інтегроване середовище розробки)
NFC	–	Near Field Communication
PIN	–	Personal Identification Number
PWM	–	Pulse Width Modulation
RFID	–	Radio Frequency Identification
SRAM	–	Static Random Access Memory
USB	–	Universal Serial Bus
VCC	–	Voltage Common Collector
VIN	–	Input Voltage

ВСТУП

З підписанням Угоди про Асоціацію між Україною та Європейським Союзом наша країна взяла на себе зобов'язання щодо поглиблення інтеграції у багатьох сферах. Що стосується енергетики, це означає приєднання до Третього енергетичного пакету – зведення законів щодо лібералізації газового та електроенергетичного ринків. Головним аспектом цих законів є обмеження монополії постачальників енергоресурсів, які володіють виробничими потужностями та мережами доставки, та блокують доступ до них інших компаній. У рамках імплементації Третього енергопакета нашої країні ще необхідно виконати низку дій, спрямованих на реалізацію всіх вимог. Насамперед це завершення лібералізації ринку природного газу, який у серпні 2020 року був запущений для населення. По-друге – відкриття ринку електроенергії в сегменті В2С. Важливо також здійснити приєднання Об'єднаної енергетичної системи України до Європейської в рамках ENTSO-E та вирішити питання збалансованості енергосистеми з урахуванням швидкозростаючих потужностей відновлюваних джерел енергії.

Все це безпосередньо вплине на учасників енергетичного ринку України, зокрема на постачальників, сприяючи зростанню внутрішніх цін через пошук додаткових джерел фінансування для покриття «зеленого тарифу», а також призведе до зростання конкуренції і, одночасно, до укрупнення гравців на цих ринки.

Сьогодні в Україні повноцінно працює оптовий ринок електроенергії та його роздрібний сегмент В2С. Фактично за два роки функціонування ринку ми спостерігаємо якісні зміни: активно розвивається торгівля у рамках двосторонніх договорів, ринок стає більш цивілізованим, удосконалено регуляторну базу, практично відсутні можливості для маніпуляцій.

Учасники ринку бачать позитивні кроки в нормалізації продажу енергоресурсів, вироблених державними виробниками електроенергії, насамперед – ДП «НАЕК «Енергоатом» та ДП «Центренерго», результати

торгів яких на Українській енергетичній біржі раніше викликали багато питань у гравців ринку та правоохоронців. Об'єктивна ситуація така: держава, постачальники та споживачі готові до запуску роздрібного ринку електроенергії та в сегменті В2С, що становить 31% від усього споживання в Україні або близько 36 млрд кВт-год споживання за рік.

Постачальники навчилися формувати свої портфелі, прогнозувати споживання своїх клієнтів, мінімізувати небаланси. У фізосіб практично немає побоювань щодо отримання послуги з постачання електроенергії від незалежних приватних компаній, а не від постачальників універсальної послуги. Державі потрібно лише запустити фінансові механізми захисту малозабезпечених верств населення, тоді від відкриття роздрібного ринку електроенергії виграють усі. У тому числі й держава, оскільки ДП «НАЕК «Енергоатом» та ПрАТ «Укргідроенерго» було б звільнено від зобов'язань у рамках ПСО та змогли б вільно продавати свій ресурс на біржі за ринковими цінами, отримуючи при цьому надприбутки та поповнюючи бюджет податками. Постачальники будуть змушені еволюціонувати для того, щоб надавати клієнтам якісніші послуги [1]. Конкуренція неминуче призведе до того, що на ринку залишаться справді надійні компанії, орієнтовані на свого клієнта. Фізичні особи, у свою чергу, отримали б доступ до якісних різноманітних продуктів з пропозиціями щодо відстрочення платежів, закупівлі фіксованого обсягу на тривалий термін тощо.

Нинішнього, а особливо наступного року, на нас чекає наростаюча проблема із «зеленою генерацією», нарощування боргів перед її виробниками з боку держави та посилення балансування енергосистеми України. За оцінками НКРЕКУ, загальний обсяг виробництва електроенергії з відновлюваних джерел енергії у 2022 році, із введенням в експлуатацію нових СЕС та ВЕС, перевищить 15 млрд кВт-год (140% до 2020 року). Відповідно, очікується профіцит виробництва сонячної енергії в денні години та дефіцит балансуємих потужностей у нічний час. Балансування

енергосистеми України стане справжнім випробуванням як для «зелених», яких постійно обмежуватимуть у літній період, так і для всіх інших учасників ринку.

Також очікуються проблеми у тепловій генерації, яка працює на природному газі: цього сезону прогнозується висока ціна на ресурс, що призведе до збитковості їхньої діяльності з урахуванням собівартості виробництва електроенергії. Тому розробка портативного і недорогого у виробництві пристрою збору інформації являється **актуальною** задачею у реаліях систем постачання енергоресурсів.

Мета: покращення методів збору інформації для постачальників енергоресурсів.

Об'єкт: процес збору та передачі інформації щодо параметрів та стану систем енергопостачання.

Предмет: Система збору та передачі інформації щодо параметрів та стану систем енергопостачання.

Завдання:

- провести аналітичний огляд літератури та патентної інформації;
- порівняти вже існуючі аналоги і визначиться з оптимальним рішенням;
- обрати компоненти;
- розробити альтернативні шляхи моделювання пристрою;
- розробити моделі системи збору інформації для постачальників енергоресурсів;
- розробити програмний код;
- розробити питання безпеки життєдіяльності та охорони праці.

Практичне значення отриманих результатів даної роботи полягає у тому, що представлена модель «розумної енергосистеми» може значно розкрити потенціал подібних технологій в прагненні зробити життя людей комфортнішим. Але, окрім цього, технологія може мати великі перспективи

при впровадженні її в роботу державних організацій та приватних компаній, що часто мають необхідність аналізувати значну кількість параметрів у впровадженні та реалізації систем постачання енергоресурсів. Змодельована в ході даної роботи система може бути використана енергопостачальниками різних напрямів, для покращення роботи подібних систем.

Методи дослідження. При проведенні дослідження використовувались методи опису систем що використовуються енергопостачальниками для використання та передачі енергоресурсів. Також використовувались експериментальні методи, методи індукції, регресійного та спектрального аналізу, пояснення, абстрагування, сходження від абстрактного до конкретного, а також системно-структурний метод.

Гіпотеза. В ході проведення роботи було перевірено гіпотезу про те що збір інформації, з різних точок енергопостачальної системи є важливою умовою для покращення розвитку подібних систем.

Наукова новизна. На момент написання роботи не знайдено універсальних систем збору і передачі інформації для постачальників енергоресурсів, що і обумовлює наукову новизну даної роботи.

РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ТА ПАТЕНТНОЇ ІНФОРМАЦІЇ

1.1 Ключові енергетичні дані

Україна виробляє все викопне паливо (у 2018 році: 14,4 млн тонн нафтового еквіваленту [Мтн еквівалента] вугілля, 16,5 Мтне природного газу та 2,3 Мтне сирової нафти), але в кількостях, недостатніх для забезпечення загального попиту на енергію.

Тим не менш, майже 65% загальної потреби України в енергії покривається за рахунок внутрішнього виробництва. Така висока самодостатність пояснюється виробництвом ядерної енергії, оскільки Україна є сьомим у світі виробником (83 терават-години [ТВт-год] у 2019 році). Більше половини електроенергії в країні виробляється за допомогою атомної енергії, а Україна та Вірменія є єдиними країнами EU4Energy, які виробляють ядерну енергію [2].

Україна є найбільшим споживачем енергії серед країн, які фокусуються на EU4Energy. У 2018 році його первинне енергопостачання становило 93 млн тне, що відповідає приблизно 90% споживання Польщі.

Структура енергоресурсів в Україні відносно диверсифікована, і на жодне паливо не припадає більше 30% енергетичного балансу. У 2018 році частка вугілля (основного палива в країні) знизилася до 30%, за ним слідує природний газ (28%) і ядерна атомна електростанція (24%).

Україна залежить від імпорту приблизно на 83% споживання нафти, 33% природного газу та 50% вугілля. У 2018 році Україна імпортувала 8,5 млн т н.е (10,6 млрд куб. м) природного газу, 13,8 млн тне вугілля та 10,4 млн тне нафтопродуктів. Білорусь є основним постачальником нафтопродуктів в Україну.

У 2018 році загальне кінцеве споживання України (TFC; без сектору трансформації) становило 51,5 млн т н.е.

Промисловість є найбільшим кінцевим споживачем енергії (19,1 Мтн.е. у 2018 році). Житловий сектор займає друге місце (16,7 Мт.н.е.), причому домашні господарства є основними споживачами природного газу (8,7 Мт.н.е. у 2018 році). Частка вугілля в кінцевому споживанні дуже мала (12%), оскільки більша частина вугілля, що споживається в країні, використовується для виробництва електроенергії та тепла.

Енергоємність на ВВП за паритетом купівельної спроможності (ППС) дуже висока: з 0,25 тони нафтового еквіваленту (toe) на тисячу доларів США ППС у 2015 році, це друга за величиною серед країн після Туркменістану.

У 2018 році відновлювані джерела енергії становили лише 5% енергетичного балансу та 9% виробництва електроенергії (13,4 ТВт-год у 2019 році).

1.2 Розумне місто: що це таке і навіщо воно потрібне?

В останні кілька років спостерігається вибухове зростання інформаційно-комунікаційних технологій (ІКТ) за рахунок просування апаратних і програмних конструкцій. Використання ІКТ в містах в різних формах для різних міських заходів призвело до підвищення ефективності міських операцій, і ці міста були позначені багатьма термінами, такими як "кібервілля", "цифрове місто", "електронне місто", "гнучкість", "інформаційне місто", "провідне місто" та "розумне місто". Smart city є найбільшою абстракцією серед понять, що використовуються, оскільки воно охоплює інші поняття, що використовуються для міст. Розумне місто є концепцією, і до цих пір немає чіткого і послідовного визначення концепції серед наукових кіл і практиків. У спрощеному поясненні розумне місто – це місце, де традиційні мережі та послуги стають більш гнучкими, ефективними та стійкими з використанням інформаційних, цифрових та телекомунікаційних технологій, щоб покращити свою діяльність на благо мешканців. Іншими словами, в розумному місті цифрові технології перетворюються на кращі державні послуги для жителів, а також

використовуються для кращого керування ресурсами, впливаючи на навколишнє середовище менше. Одним з формальних визначень розумного міста є наступне: місто, що «з'єднує фізичну інфраструктуру, інформаційно-технологічну інфраструктуру, соціальну інфраструктуру та бізнес-інфраструктуру для використання колективного інтелекту міста». Іншим формальним і всеосяжним визначенням є наступне: «Розумне стале місто – це інноваційне місто, яке використовує інформаційно-комунікаційні технології (ІКТ) та інші засоби для підвищення якості життя, ефективності міських операцій і послуг, а також конкурентоспроможності, забезпечуючи при цьому задоволення потреб нинішнього і майбутніх поколінь щодо економічних, соціальних та екологічних аспектів».

Будь-яка комбінація різних розумних компонентів може зробити міста розумними. Місто не повинно мати всі компоненти, які позначені як розумні. Кількість розумних компонентів залежить від вартості і доступних технологій. Населення світу значно зросло за останні десятиліття, так само як і очікування рівня життя. Прогнозується, що близько 70% світового населення буде жити в міських районах до 2050 року.

В даний час міста споживають 75% світових ресурсів і енергії, що призводить до виробництва 80% парникових газів. Таким чином, в найближчі кілька десятиліть може бути серйозний негативний вплив на навколишнє середовище. Це робить концепцію розумних міст необхідністю. Створення розумних міст є природною стратегією для пом'якшення проблем, що виникають завдяки швидкій урбанізації та зростанню міського населення. Розумні міста, незважаючи на пов'язані з цим витрати, можуть зменшити споживання енергії, споживання води, викиди вуглецю, вимоги до транспортування та міські відходи [3].

Розумні міста по всьому світу досить різноманітні за своїми характеристиками, вимогами та компонентами. Загалом, стандарти, встановлені такими організаціями, як Міжнародна організація зі

стандартизації (ISO), забезпечують загальнозрозумілі специфікації для стимулювання зростання, забезпечуючи при цьому якість, ефективність та безпеку. Стандарти можуть відігравати важливу роль у розвитку та будівництві розумного міста [3]. Стандарти також можуть передбачати вимоги до моніторингу технічних і функціональних характеристик розумних міст. Стандарти також можуть допомогти вирішити проблему зміни клімату, вирішити питання безпеки та транспорту, забезпечуючи при цьому якість водних послуг. Стандарти враховують різні фактори, такі як ділова практика та управління ресурсами, допомагаючи контролювати роботу розумного міста і, таким чином, зменшити його вплив на навколишнє середовище. IEEE розробляє стандарти для розумних міст для своїх різних компонентів, включаючи розумні мережі, IoT, eHealth та інтелектуальні транспортні системи (ITS). Конкретним прикладом такого стандарту є ISO 37120, який визначає 100 показників ефективності міста, які включають 46 основних та 54 опорних показників. Деякі вибрані показники такі: економіка, освіта, енергетика та навколишнє середовище, які можуть бути використані міськими громадськими органами для порівняння їх сервісних показників, вивчення кращих практик з інших міст, а також порівняння свого міста з іншими містами.

1.3 Розумні міста: компоненти та характеристики

Компоненти розумних міст включають в себе наступне: розумна інфраструктура, розумні будівлі, розумний транспорт, розумна енергія, розумна охорона здоров'я, розумні технології, розумне управління, розумна освіта та розумні громадяни. Коротке обговорення цих компонентів буде представлено в наступних розділах. Різні розумні міста мають різні рівні цих розумних компонентів, залежно від їх фокусу.

Різні атрибути розумних міст включають стійкість, якість життя (QoL), урбанізацію та кмітливість. Стійкість розумного міста пов'язана з міською інфраструктурою та управлінням, енергетикою та зміною клімату,

забрудненням та відходами, а також соціальними проблемами, економікою та здоров'ям. Якість життя (QoL) можна виміряти з точки зору емоційного та фінансового благополуччя громадян. Аспекти урбанізації розумного міста включають в себе кілька аспектів і показників, таких як технології, інфраструктура, управління та економіка [5]. Розумність розумного міста концептуалізується як прагнення поліпшити економічні, соціальні та екологічні стандарти міста та його жителів. Різні часто цитовані аспекти розумності міста включають розумну економіку, розумних людей, розумне управління, розумну мобільність та розумне життя.

Є чотири основні теми для розумного міста, а саме: суспільство, економіка, навколишнє середовище та управління. Тема суспільства розумного міста означає, що місто для його жителів або громадян. Економічна тема розумного міста означає, що місто здатне процвітати при постійному зростанні робочих місць та економічному зростанні. Екологічна тема розумного міста вказує на те, що місто зможе підтримувати свою функцію і залишатися в експлуатації для нинішнього і майбутніх поколінь. Тема управління розумним містом передбачає, що місто є надійним у своїй здатності керувати політикою та об'єднувати інші елементи.

Інфраструктура розумного міста включає в себе фізичні, інформаційні та комунікаційні технології (ІКТ) і послуги. Фізична інфраструктура є реальним фізичним або структурним суб'єктом розумного міста, включаючи будівлі, дороги, залізничні колії, лінії електропередач та систему водопостачання. Фізична інфраструктура, як правило, є не розумним компонентом розумних міст. Інфраструктура ІКТ є основним розумним компонентом розумного міста, який склеює всі інші компоненти, по суті діючи як нервовий центр розумного міста. Сервісна інфраструктура базується на фізичній інфраструктурі і може мати деякі компоненти ІКТ. Приклади компонентів обслуговування включають систему швидкого

транзиту та розумні мережі. Кількість об'єктів міста, необхідних як функція населення міста, можна розрахувати наступним чином:

$$N_f = N_p \left(\frac{R_p}{Year} \right) \left(\frac{1 Year}{D Days} \right) \left(\frac{1 Hour}{N_c People} \right) \left(\frac{1 Day}{H Hours} \right)$$

де N_f – кількість об'єктів, N_p – населення міста в мільйонах, R_p – це ставка на людину, яка використовує в рік / тиждень, D – дні на рік, N_c – клієнти за годину, а H – години на день.

ІоТ може бути концептуалізований як налаштовувана динамічна глобальна мережа мереж. Є чотири основні компоненти ІоТ: (1) Річ, (2) Локальна мережа (LAN), (3) Інтернет і (4) Хмара. Річ – це датчик, вбудований обчислювальний пристрій або вбудована система, яка може передавати і отримувати інформацію через мережу для управління іншим пристроєм або взаємодії з користувачем. Прикладом речі є датчик температури, мікроконтролер або пристрій на основі мікропроцесора. З іншого боку, мікрохвильова піч, спринклер, будинок, пральна машина або будівля не підходять під визначення «річ» [6]. Тим не менш, ІоТ разом з цими фізичними суб'єктами, такими як будівлі, можуть створити кіберфізичну систему (CPS). "Річ" може виконувати наступне: (1) ідентифікація та зберігання інформації, (2) збирати інформацію, (3) розуміти команди, (4) передавати та отримувати повідомлення, (5) сенс і (6) приводити в дії. ІоТ може бути використаний для створення розумного транспорту, розумної охорони здоров'я та управління енергією в розумних містах.



Рисунок 1.1 – Ілюстрація роботи елементів «розумного міста»

Загалом, Big Data відноситься до набору великих і складних наборів даних, таких, які важко обробляти за допомогою звичайних інструментів управління базами даних або традиційних додатків обробки даних Інтернет речей (IoT), Великі дані та розумні міста тісно пов'язані між собою, оскільки один потребує двох інших. Міські дані, які позначені в просторі і часі і генеруються в розумних містах, можуть бути Big Data. Великі дані в розумних містах можуть генеруватися з великої колекції датчиків, баз даних, електронних листів, веб-сайтів та соціальних мереж. Підраховано, що поширення датчиків, соціальних мереж, веб-сторінок, графічних і відео додатків, а також мобільних пристроїв генерує більше 2,5 квінтильйонів байтів на день. Проблеми Big Data багатократні, включаючи візуалізацію, видобуток, аналіз, захоплення, зберігання, пошук та обмін. Big Data вимагає нових підходів до обробки, щоб забезпечити посилене прийняття рішень, виявлення розуміння та оптимізацію процесів. Складні механізми аналізу даних необхідні для пошуку та вилучення цінних моделей та знань з Великих даних IoT та розумних міст [7].

У класичному розумінні інфраструктура міста – це будь-який фізичний компонент міста, такий як дороги, будівлі та мости, які забезпечують роботу міста та його мешканців. Однак у контексті розумних міст усе фізичне, електричне та цифрове, що є основою розумного міста, можна розглядати як його інфраструктуру. Існує багато прикладів, і деякі з них: система швидкого транспорту, система управління відходами, дорожня мережа, залізнична мережа, система зв'язку, система світлофора, система вуличного освітлення, офісні приміщення, система водопостачання, система газопостачання, система електропостачання, пожежогасіння система, лікарняна система, мости, багатоквартирні будинки, готелі, цифрова бібліотека, правоохоронні органи, економічна система тощо. Бекендом розумної інфраструктури є інфраструктура ІКТ, яка створює фізичну інфраструктуру «розумний». Інфраструктура ІКТ є основоположною для побудови розумних міст і залежить від факторів, пов'язаних з її доступністю та продуктивністю. Інфраструктура ІКТ включає комунікаційну інфраструктуру, таку як волоконна оптика, мережі Wi-Fi, бездротові точки доступу, а також сервісно-орієнтовані інформаційні системи. Розумна інфраструктура є більш ефективною, безпечною та відмовостійкою порівняно з класичною інфраструктурою. Розумна інфраструктура може мати фізичну інфраструктуру, датчики, мікропрограмне забезпечення, програмне забезпечення та проміжне програмне забезпечення як загальні компоненти. «Проміжне програмне забезпечення», яке є специфічним типом програмного забезпечення, зазвичай відіграє вирішальну роль в автоматизації та швидкому реагуванні розумної інфраструктури. Проміжне програмне забезпечення накопичує дані та об'єднує їх у загальну платформу для аналітики та звітності. Проміжне програмне забезпечення в процесі може виконувати веб-відображення інформаційної панелі для візуального знімка інфраструктури. У разі високого споживання енергії, ненормальних витрат на технічне обслуговування та багатьох звичайних і ненормальних ситуацій

вимагається швидка увага обслуговуючого персоналу. Проміжне програмне забезпечення надає керівникам або оперативним працівникам численні відомості, включаючи управління викидами вуглецю та стійкість, а також загальну картину інфраструктури розумного міста, незалежно від того, скільки інфраструктур, будівель або географічних місць задіяно в цілому. Інформація про інтелектуальну інфраструктуру через проміжне програмне забезпечення та ІКТ доступна швидко і може бути доступна будь-де для операційного персоналу та керівництва для прийняття кращих рішень, які мають негайний вплив на роботу розумного міста. Конкретним прикладом розумної інфраструктури є розумна електромережа або, як її спрощено називають, розумна мережа. Розумна мережа складається з різних джерел енергії (відновлюваних або звичайних), розумних лічильників, механізмів оперативного контролю, механізмів балансування навантаження та відмовостійких механізмів для ефективною та надійною доставки електроенергії кінцевому споживачу від різних джерел енергії [8].

Розумні будівлі можна розглядати як частину розумної інфраструктури або як незалежні компоненти розумних міст. Розумна будівля може мати різне обладнання, програмне забезпечення, датчики та розумні пристрої для різних автоматизованих операцій, включаючи мережу передачі даних, передачу голосу по IP (VoIP), розповсюдження відео, відеоспостереження, контроль доступу, керування живленням та управління освітленням. Розумні будівлі відрізняються від зелених будівель. Зелені будівлі – це стійкі споруди з високою енергоефективністю, ефективністю використання води та внутрішнім контролем навколишнього середовища з метою зменшення вуглецевого сліду та забезпечення оптимальної енергетичної ефективності.

Розумні будівлі – це набагато більша концепція, ніж зелені будівлі. Розумні будівлі можуть легко підключатися до інших будівель, людей і технологій, глобального середовища та розумних електромереж. Розумні будівлі ефективно використовують знання, які є за межами їхніх стін і вікон.

Наприклад, розумну мережу можуть використовувати розумні будівлі. У цьому сценарії розумна будівля може легко адаптуватися до потреби в енергії, а також до мережі, щоб мати ефективне та недороге використання електроенергії. Розумні будівлі можуть використовувати динамічні тарифи на електроенергію, за яких будівля нараховується ближче до фактичної вартості виробництва електроенергії на момент її використання замість середньої вартості протягом тривалих періодів часу. Використання Інтернету речей (IoT) надає інтегровані рішення, які можуть обробляти та аналізувати великі обсяги даних, які максимізують експлуатаційну та енергоефективність розумних будівель. Переваги розумної будівлі включають наступне: прийняття рішень на основі даних для високої ефективності та недорогих операцій, більшого використання ресурсів, зниження структури капіталу та операційних витрат, визначення ризиків та управління ними, а також стійкість.

1.4 Розумна енергія

Енергія – це властивість об'єкта або системи, яка визначає його здатність виробляти роботу. Енергія може бути в різних формах, таких як потенційна енергія, кінетична енергія, хімічна енергія і теплова енергія. Джерела енергії також досить різноманітні, включаючи сонце, викопне паливо, газ, електроенергію та акумулятори. Енергія не може бути ні створена, ні знищена, але її можна трансформувати з однієї форми в іншу. За останні кілька років, окрім традиційних форм енергії, з нею пов'язують багато інших термінів, включаючи чисту енергію, зелену енергію, стійку енергію, відновлювані джерела енергії та розумну енергію. Побоювання, що джерела енергії, доступні для споживання людиною, будуть виснажені, є причиною цих нових термінів, пов'язаних з енергетикою. Терміни чиста енергія або зелена енергія говорять про те, що споживання енергії має дуже мінімальний негативний вплив на навколишнє середовище. Наприклад, сонячна енергія або енергія вітру є формами зелених джерел енергії. Стала

енергія та відновлювані джерела енергії – це джерела енергії, які неможливо спожити протягом кількох поколінь і які можна відновити швидше, ніж можна спожити. Однак між стійкою енергією та відновлюваною енергією можуть бути певні відмінності: стійкі джерела енергії – це ті, які не створені людьми, тоді як відновлювані джерела створені людьми. Одним із прикладів відновлюваної енергії є біогаз, який вимагає зростання, споживання та утилізації органічних матеріалів для його виробництва. Іншим спорідненим терміном є нульова енергетична система або будівлі з нульовим споживанням енергії, в яких споживана та вироблена енергія є однаковою кількістю, і, отже, чисте споживання в цих структурах можна вважати рівним нулю. Що таке розумна енергія? Розумна енергетика – це набагато ширше поняття, ніж будь-яка з перерахованих вище, наприклад, традиційна енергія чи чиста енергія тощо. Розумна – це концепція, яку можна розглядати як модель «Інтернету енергії». Ця модель заснована на одному або кількох принципах інтелектуального виробництва електроенергії, розумних електромереж, розумного зберігання та розумного споживання. По суті, будь-яка традиційна енергетика, чиста енергетика, зелена енергія, стійка енергетика та відновлювані джерела енергії разом з інформаційно-комунікаційними технологіями (ІКТ) створюють розумну енергію. Різні компоненти розумної енергії представлені на рис. 1.2. Ілюстрація розумної енергетичної системи представлена на рис. 1.3.

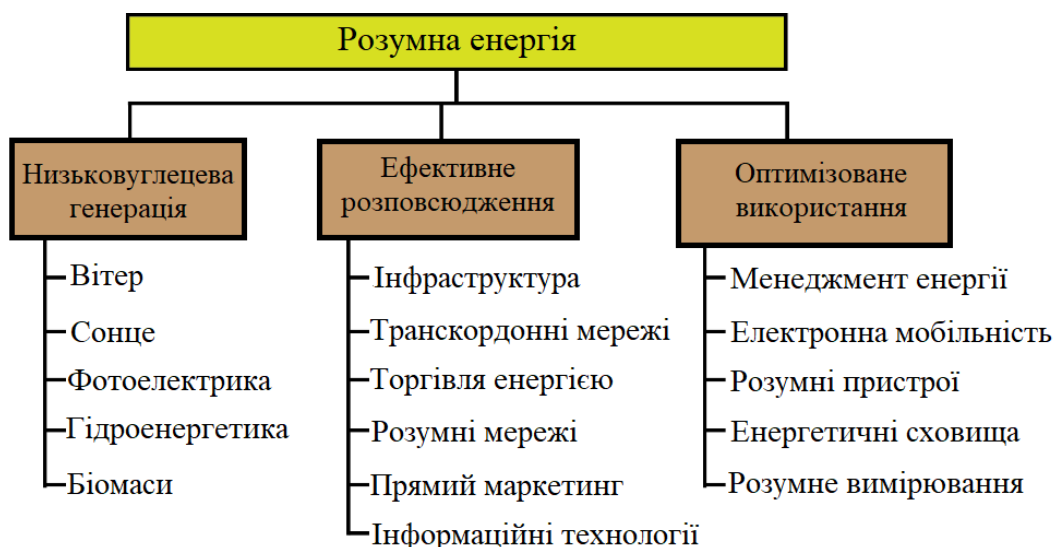


Рисунок 1.2 – Елементи системи «розумної енергії».

Розумна енергетична система складається з інтелектуальної інтеграції децентралізованих екологічних джерел енергії, ефективного розподілу та оптимізованого споживання електроенергії. Таким чином, розумна енергія складається з трьох незалежних будівельних блоків, які повинні бути зшиті разом і ефективно взаємодіяти один з одним, щоб утворити єдину розумну енергетичну систему.

Виробництво з низьким вмістом вуглецю, також відоме як зелена енергія, фотоелектрична, сонячна теплова, біогазова та вітрова енергія може бути важливою частиною розумної енергетичної системи.

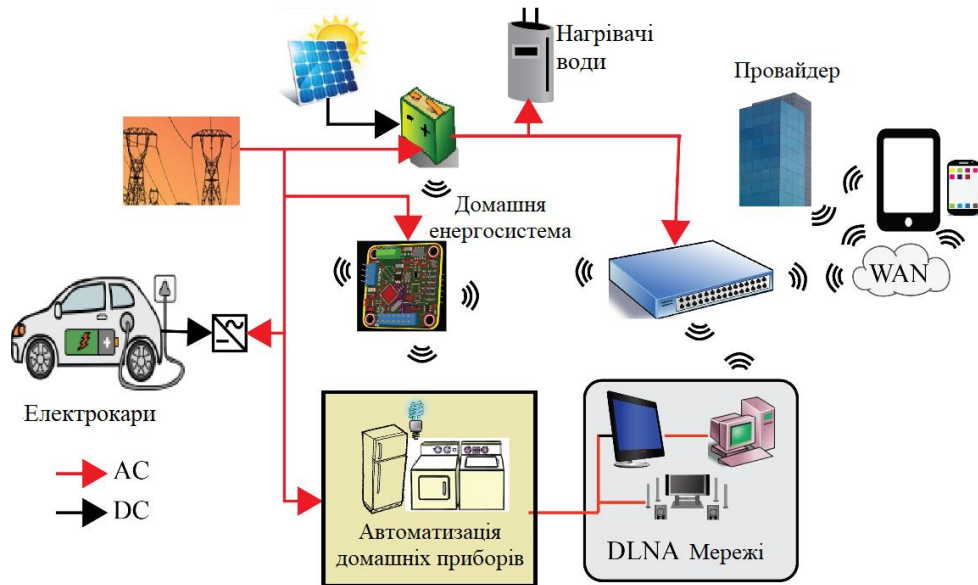


Рисунок 1.3 – ілюстрація системи «розумної енергії»

Ефективний розподіл в інтелектуальній енергетичній системі стає можливим завдяки використанню розумної інфраструктури, розумної мережі, розумних лічильників, а також належного рівня використання інформаційно-комунікаційних технологій (ІКТ). Ядром розумної енергетичної системи є інформаційна інфраструктура, яка відповідає за збір інформації про споживання енергії, а також обмін інформацією про тарифи постачальників. ІКТ можна використовувати для керування операціями з відповідним рівнем споживання енергії для розумних приладів, таких як посудомийні машини та водонагрівачі. ІКТ також корисні для транзакцій щодо електромобілів (PEV), а також опалення, вентиляції та кондиціонування повітря (HVAC). ІКТ можна ефективно використовувати для придбання енергії з різних джерел, таких як сонячні панелі, вітрові турбіни та інші можливі джерела енергії. Оптимізоване споживання системи є 3-м ключовим компонентом інтелектуальної енергетичної системи. Ефективне використання ефективного накопичення енергії, розумного вимірювання та ефективного управління енергією може стати ключем до оптимізації споживання енергії в розумній енергетичній системі.

Основою розумної енергетичної системи є розумна енергетична мережа або розумна мережа. У формальному визначенні розумна мережа ефективно інтегрує дії та поведінку всіх підключених користувачів, таких як споживачі, генератори та користувачі, які одночасно є споживачами і генераторами. Розумні мережі забезпечують ефективні, економічні та стійкі енергетичні системи з низьким рівнем втрат, більш високою якістю постачання, безпекою системи та користувачів, безпекою постачання та стійкістю системи. Розумна мережа дає змогу інтегрувати різноманітні доступні джерела енергії, від теплової енергії на основі викопного палива до зеленої фотоелектричної енергії та енергії вітру. Майбутні розумні мережі будуть набагато складнішими, ніж нинішнє покоління. Наприклад, може настати день, коли кожен користувач також виробляє сонячну енергію, енергію біопалива і навіть енергію вітру. Розумна мережа ефективно синхронізує цю енергію з різних джерел і забезпечує електроенергію з заданою напругою та частотою без будь-яких коливань. Використання ІКТ відіграє ключову роль у розумній електромережі для наступного: для підтримки управління споживанням енергії відповідно до попиту; для диспетчеризації виробництва електроенергії для сонячних панелей і вітрових турбін; для полегшення визначення місця розташування незалежні трансакційні послуги в точках продажу РЕV та покращення відносин зі споживачами. Розумний облік енергії є важливою складовою розумної мережі. Розумний лічильник реєструє споживання електроенергії через певні проміжки часу та передає цю інформацію комунальному підприємству для моніторингу та виставлення рахунків. Це забезпечує точне та надійне зчитування використання без участі людини зчитування або запису. Розумна батарея або інтелектуальна батарея, яка може бути виготовлена з літій-іонних або паливних елементів, може бути ефективною для зберігання енергії та ефективної доставки, а також мати більш тривалий термін служби.

1.5 Smart Grid

У багатьох країнах підприємства енергетичного сектора економіки переживають період реформування. Процеси злиття, поглинання і зміни структури управління, меж сфери діяльності та територіальної присутності змушують багато колишніх монополій шукати для себе нові моделі створення вартості. Неминуче змінюються завдання компаній та їх бізнес-процеси. Формуються ринки надання комунальних послуг. Впроваджуються ринкові механізми. Потрібні технологічні зміни, що відповідають сучасним потребам розвитку галузі. Хоча всі ці зміни відрізняються залежно від розташування та виду діяльності енергокомпаній, інновації неминуче призводять до перетворення всієї сфери комунальних послуг.

Відповідно до концепції Smart Grid у числі пріоритетних напрямів розвитку ІТ в енергетиці на найближчі роки можна виділити:

1. Широке впровадження на нових і модернізованих точках вимірювання інтелектуальних (smart) вимірювальних приладів – «розумних» лічильників з функцією дистанційного керування профілем навантаження вимірюваної лінії та вимірювальних перетворювачів зі стандартними комунікаційними інтерфейсами та протоколами (в тому числі бездротовими).

2. Установка на кожному великому об'єкті, приєднаному до електромережі (житловому районі, офісному центрі, фабриці тощо), удосконалених автоматизованих інформаційно-вимірювальних систем (АПС), що працюють у режимі реального часу. АПС повинні здійснювати моніторинг об'єктових процесів (наприклад, електро- або тепlopостачання, включаючи параметри якості енергії), виконувати прості алгоритми автоматичного регулювання та мати розвинені засоби інформаційного обміну із зовнішнім світом.

3. Створення широкої мережі інтегрованих комунікацій на базі різноманітних ліній зв'язку – ВОЛЗ, супутникових, GPRS, ВЧ-зв'язку по ЛЕП

та ін. Кожна АІС має бути підключена як мінімум по двох незалежних каналах зв'язку.

4. Використання в енергокомпаніях автоматизованих систем (АС) управління виробничою діяльністю. Оскільки всі енергопідприємства відносяться до виробництв із безперервним циклом, можна виділити чотири види таких систем:

- АС управління технічним обслуговуванням та ремонтами;
- АС роботи на ринках (комерційної диспетчеризації);
- АС обслуговування клієнтів;
- АС управління основним виробництвом – генерацією, передачею, розподілом, збутом (обліком споживання) або диспетчеризацією.

5. Створення інтегрованих інтерфейсів до АІС та АС управління виробничою діяльністю для автоматичного обміну даними з АС інших учасників ринку. При цьому мають бути визначені протоколи обміну та стандарти інформаційної безпеки для всіх категорій учасників ринку.

Ряд вендорів вже заявили про підтримку концепції Smart Grid та включення до своїх чергових релізів продуктів нового функціоналу. Деякі виводять на ринок рішення, побудовані відповідно до нової ідеології та демонструють велику гнучкість і функціональність у нових умовах.

Висновки до розділу 1

У зв'язку зі стрімким розвитком енергопостачальних систем України, в усіх напрямках цієї діяльності постала проблема мінімізувати втрати енергоресурсів а також створити ефективну систему контролю і постачання даних ресурсів, що враховувала би всі змінні параметри подібних систем, для надання найбільш ефективного і якісного сервісу. На сьогодні на сучасному ринку енергопостачання майже немає якоїсь універсальної системи збору інформації, яку можна було би інтегрувати в теперішніх реаліях розвитку енергопостачання, незалежно від типу енергетичних ресурсів та обраного напрямку в даній сфері. Саме тому розробка такої системи являється

актуальною. Це дозволить покращити методи збору інформації для постачальників енергоресурсів, що буде важливим кроком у розвитку енергосистем України а також дасть змогу більш точно і збалансовано розподіляти дані ресурси в залежності від отриманих параметрів. Така модель може значно розкрити потенціал подібних технологій в прагненні зробити життя людей комфортнішим. Але, окрім цього, технологія може мати великі перспективи при впровадженні її в роботу державних організацій та приватних компаній, що часто мають необхідність аналізувати значну кількість параметрів у впровадженні та реалізації систем постачання енергоресурсів. Це стосується не тільки постачання, і може мати вплив на розвиток систем реагування на збої та неполадки, а також дасть можливість використовувати отриману інформацію в методах прогнозування надзвичайних ситуацій пов'язаних з системою постачання енергоресурсів.

РОЗДІЛ 2

ХАРАКТЕРИСТИКА ДОСЛІДЖУВАНОВОГО ОБ'ЄКТА

2.1 Огляд подібних технологій: Smart Grid

Smart Grid ("інтелектуальні мережі електропостачання") – це модернізовані мережі електропостачання, які використовують інформаційні та комунікаційні мережі та технології для збору інформації про енерговиробництво та енергоспоживання, що дозволяє автоматично підвищувати ефективність, надійність, економічний зиск, а також стійкість виробництва та розподілу електроенергії.

Технологічні рішення Smart Grid можна розділити на п'ять ключових областей:

- вимірювальні прилади та пристрої, що включають, насамперед, smart-лічильники та smart-датчики;
- удосконалені методи управління;
- удосконалені технології та компоненти електричної мережі: гнучкі системи передачі змінного струму FACTS, надпровідні кабелі, напівпровідникова, силова електроніка, накопичувачі;
- інтегровані інтерфейси та методи підтримки прийняття рішень, технології управління попитом на енергію, розподілені системи моніторингу та контролю), розподілені системи поточного контролю за генерацією, автоматичні системи вимірювання протікаючих процесів, а також нові методи планування та проектування як розвитку, так і функціонування енергосистеми та її елементів;
- інтегровані засоби комунікації.

Смарт-лічильники (інтелектуальні лічильники) енергоресурсів – різновид удосконалених приладів обліку, забезпечених комунікаційними засобами для передачі накопиченої інформації за допомогою мережевих

технологій з метою моніторингу та здійснення розрахунків за комунальні послуги.

Термін "інтелектуальний лічильник", як правило, ставився до лічильників електроенергії, проте останнім часом також застосовується до засобів вимірювання інших споживаних ресурсів: природного газу, тепла та води.

АПС КОЕ – Автоматизована інформаційно-вимірювальна система комерційного обліку електроенергії.

АПС КОЕ є сукупністю апаратних та програмних засобів, що забезпечують дистанційний збір, зберігання та обробку даних про енергетичні потоки в електромережах. АПС КУЕ необхідна для автоматизації торгівлі електроенергією та також виконує технічні функції контролю за режимами роботи електрообладнання.

Ієрархічна система, що представляє собою технічний пристрій, що функціонально об'єднує сукупність вимірювально-інформаційних комплексів точок вимірювань, інформаційно-обчислювальних комплексів електроустановок, інформаційно-обчислювального комплексу та системи забезпечення єдиного часу, що виконує функції проведення вимірювань, збору, обробки та зберігання результатів вимірювань, інформації про стан об'єктів та засобів вимірювань, а також передачі отриманої інформації до інтегрованої автоматизованої системи управління комерційним обліком на оптовому ринку електроенергії в автоматизованому режимі.

Інформаційна мережа Smart Grid поєднує безліч технічних елементів та вузлів (рис. 2.1).

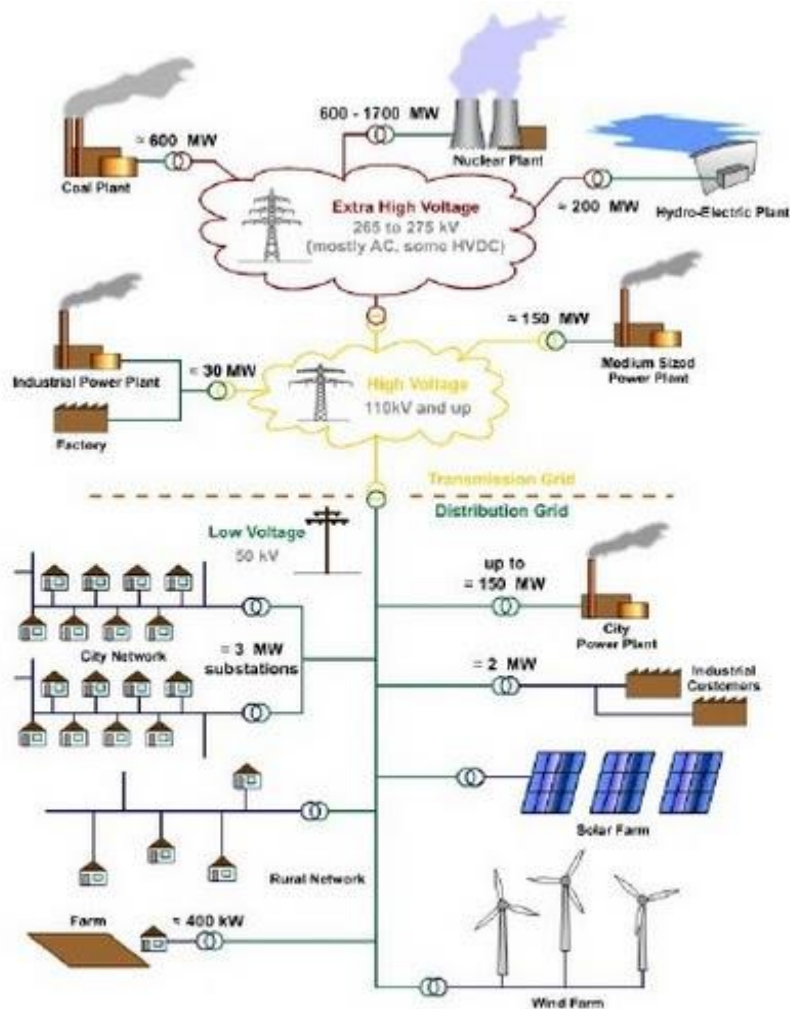


Рисунок 2.1 – Архітектура мережі Smart Grid

У домені споживачів електроенергії такими елементами є смарт-лічильники, електричні прилади, системи акумулювання енергії, електротранспорт та об'єкти розподіленої генерації. У домені передачі та розподілу енергії елементами інформаційної системи є блоки вимірювання фаз, контролери підстанцій, об'єкти розподіленої генерації, системи акумулювання енергії. В операційному домені елементами інформаційної системи є SCADA-системи. У свою чергу, кожен з перелічених вище доменів може складатися зі своїх підмереж, що робить загальну архітектуру мережі дуже складною. При цьому одним із ключових компонентів даної інформаційної системи є безпека зберігання та передачі даних.

Відповідно до загальноприйнятих підходів за кордоном, IP (Internet Protocol) – мережі є ключовим елементом інформаційних систем Smart Grid. До переваг використання IP-протоколів слід віднести широку поширеність даної технології, наявність значної кількості вже розроблених галузевих стандартів, значну кількість розроблених відповідних програмних продуктів.

Крім того, рішення, побудова на принципах IP, мають хорошу масштабованість, що дозволяє включати в інформаційну систему значну кількість елементів мережі (смарт-лічильників, домашніх приладів тощо).

Важливу роль формуванні перспектив світового ринку смарт-обліку грає уніфікація інтерфейсів передачі від первинних приладів обліку до концентраторів і зажадав від концентраторів - до системи обробки даних.

У країнах ЄС найбільшого поширення набули інтерфейси передачі за допомогою GSM/GPRS каналів, PLC- і радіоканалів. Нижче наведено ключові інтерфейси зв'язку, що використовуються у системах смарт-обліку енергоресурсів у країнах ЄС.

2.2 Використання подібних технологій у світі

Використовувані хмарні платформи для збору даних із підключених елементів SmartGrid та наскрізного оптимізаційного управління енергомережами можна класифікувати на два види:

інтеграційні, що використовуються переважно для збору даних та реалізації найбільш затребуваних завдань моніторингу, зокрема, автоматичного виявлення відключень споживачів та випадків крадіжки електроенергії, та аналітичні, які використовуються для оптимізаційного передиктивного керування енергомережами в режимі реального часу, у тому числі для керування програмами DemandResponse та об'єктами розподіленої генерації.

За оцінками J'son & Partners Consulting глобальний обсяг споживання сервісів обох видів хмарних IoT -платформ досяг у 2018 році \$860 млн., а темпи зростання споживання у період 2015-2018 років становили 85%

CAGR. За прогнозом J'son & Partners Consulting, світовий обсяг споживання видів платформ, що розглядаються, зросте в 2023 році до \$4,8 млрд, CAGR складе 41%. 75% всіх енергоресурсів на Землі споживається жителями міст, частка яких становить 53% від загального населення планети, підрахували експерти ООН . За оцінками Schneider Electric , до 2050 року потреби в енергоресурсах, за часткою городян вже в 86%, зросте на 50% [1] .

Почасти тому у світі кількість мегаполісів, що реалізують проекти розумних міст, інтелектуального обліку та розумних електромереж, постійно зростає. Експерти консалтингової компанії ABI Research навесні 2017 року спрогнозували, що до 2022 року зростання кількості розумних лічильників, які відслідковують споживання електроенергії, води та газу, виявиться дворазовим. Це відбудеться на тлі розвитку технологій енергоефективних мереж великого радіусу дії (LPWAN).

Обсяг прибутку мобільних операторів від підключення електромереж та розумних лічильників до 2026 року становитиме \$26 млрд, спрогнозували аналітики. Machine Research. Згідно з прогнозом, до 2022 року в Європі налічуватиметься 158 млн інтелектуальних приладів обліку, підключених до LPWA-мереж. Наразі близько 60 млн лічильників європейців не оснащені можливостями дистанційної передачі показань.

Згідно з підрахунками Єврокомісії, до 2020 року буде замінено 200 млн лічильників енергії та 45 млн лічильників газу. До зазначеного часу такі пристрої будуть встановлені у 72% споживачів електроенергії та 40% споживачів газу. І все ж на динаміку цього зростання насамперед впливатимуть ініціативи міських адміністрацій та урядів країн. Наприклад, у 2014 році уряд Великобританії завдання оснащення житла громадян розумними лічильниками визнав національним. За даними ABI Research, у Великій Британії рівень проникнення інтелектуальних приладів обліку на кінець 2016 року становив менше 10%. У 2017 році результати опитування

MoneySuperMarket показали: інтелектуальними приладами обліку користуються лише 16% жителів Об'єднаного Королівства.

Також, за даними ABI Research, уряд Індії планував реалізувати 15 «пілотів» у сфері розумного обліку, але сім із них навіть не розпочиналися. Агентство з регулювання енергетики ANEEL Бразилії, наприклад, відмовилося від планів з переведення жителів у найкоротші терміни та розумні лічильники.

Обсяг поставок розумних лічильників у Європі за підсумками року досягне 23 млн. Berg Insight прогнозує, що постачання розумних електричних лічильників у Європі зросте за підсумками 2017 року на 21%. Обсяг поставок складе 23,1 млн. пристроїв. Зростання продажів таких пристроїв обумовлено затребуваністю розумних приладів обліку у Франції, Італії та Великобританії та Іспанії [2]. Значна кількість пристроїв, розгорнутих у 2017 році, використовує зв'язок ліній електромережі (PLC). У 2018 році обсяг постачання розумних PLC-пристроїв піде на спад, проте зросте частка пристроїв, що працюють у мережах LTE-M/NB-IoT та RF.

«Франція та Іспанія стануть останніми великими країнами у Західній Європі, які впровадили інтелектуальні рішення з PLC. Починаючи з цього року, Італія переключатиметься на лічильники другого покоління, що працюють у гібридних мережах PLC/RF. Це дозволить підвищити надійність передачі даних», – зазначив Тобіас Руберг (Tobias Ryberg), старший аналітик Berg Insight. У Великобританії та Нідерландах зосереджено використання 2G/4G, а країни Скандинавії активно використовують RF.

Аналітики Berg Insight вважають, що на технології мобільного зв'язку, оптимізовані для IoT (LTE-M/NB-IoT), а також RF, протягом наступних 5-10 років припаде більшість підключень.

Перехід від 2G до 4G/5G зрештою забезпечить більш високу продуктивність за менших витрат. LTE -M, ймовірно, стане найбільш підходящою технологією для вимірювання споживання електрики. NB-IoT

задовольнятиме потреби в інтелектуальному вимірі газу та води. Основні глобальні тенденції розвитку технологій Smart Grid та смарт-обліку енергоресурсів

Аналіз передових світових практик, проведений J'son & Partners Consulting, дозволяє зробити висновок, що активний розвиток сучасних технологій обліку енергоресурсів дозволяє: вивести на принципово новий рівень якість збору та аналізу даних про енергоспоживання споживачів; підвищує ефективність оперативного управління енергетичними активами; сприяє активному залученню споживачів енергії до процесів регулювання власного енергоспоживання.

А також є важливим інструментом підвищення загальної енергоефективності економіки. Ринок технологій Smart Grid у світі знаходяться на етапі розвитку. При цьому розвиток і модернізація енергетичної інфраструктури, що триває безперервно, в кожному регіоні має свої особливості і підходи.

В останні роки до здійснення програм і проектів у напрямку Smart Grid, що охоплюють широкий спектр проблем і завдань, приступила переважна більшість індустріально розвинених держав, а також багато країн, що розвиваються. Найбільш масштабні програми та проекти в цьому напрямку розроблені та здійснюються в США, Канаді та країнах Євросоюзу, а також Китаї, Південній Кореї та Японії. Прийнято рішення щодо реалізації аналогічних програм і проектів у інших великих держав (Індія, Бразилія, Мексика). Зокрема, до 2020 р. Китай передбачає досягти рівня оснащення сучасними системами обліку енергоресурсів у 90-95%, США – 50-60%. У період після 2020 року 100% оснащення смарт-лічильниками планується в США, Китаї, Бразилії, Японії, більшості країн ЄС.

Наприклад, як одне з основних завдань енергетичної політики країн ЄС, визначених у "Директиві з електроенергії", є оснащення до 2020 року не менше 80% споживачів "інтелектуальними" вимірювальними системами.

Зобов'язання країн-учасників щодо визначення плану впровадження "розумних" лічильників створюють необхідний імпульс розгортання програм розвитку Smart-систем у Європейському Союзі.

Зокрема:

Франція: видання директиви щодо інтелектуальних лічильників у вересні 2010 р., яка доручає встановлення 95% інтелектуальних лічильників до 2016 року.

Німеччина: закон від січня 2010 р., який ставить умовою встановлення Smart-лічильників у нові будівлі, що реконструюються, або на вимогу споживача.

Великобританія: уряд доручив здійснити впровадження інтелектуальних лічильників між 2012 та 2020 рр.

Іспанія: Королівський декрет 1110/2007 та Наказ міністра 2860/2007 зобов'язує до 31 грудня 2018 р. замінити всі електромагнітні лічильники електроенергії на інтелектуальні.

2.6 Оптимізація ланцюга поставок енергоносіїв

Складним бар'єром для більшої інтеграції відновлюваних джерел енергії, таких як сонячна і вітер, є їх переривчастість. Швидкість сонячного опромінення та вітру змінюються щогодини, щодня, сезонно та географічно. Більше того, доступність сонячної та вітрової енергії часто асинхронна з потребами споживачів в енергії. Одним з потенційних рішень проблеми переривчастості є зберігання енергії під час періодів і в зонах надлишкового постачання. Згодом накопичена енергія може бути використана, коли відновлювані джерела енергії недоступні безпосередньо. DOE, IRENA та IEA визнали, що розробка економічно ефективних сховищ енергії є вирішальним кроком для більш широкого впровадження відновлюваних джерел енергії. Варіанти зберігання електричної енергії включають гідроакумулятори (PSH), накопичувачі енергії стисненого повітря (CAES), батареї та хімічні сполуки.

Незважаючи на те, що PSH і CAES є зрілими та вже розгорнутими великомасштабними технологіями, PSH і CAES географічно обмежені у своїх відповідних будівельних майданчиках. З іншого боку, ємність акумуляторів набагато менша. За нинішніх витрат масштабування батарей є надзвичайно дорогим, і вони більше підходять для розподілених додатків. Енергія може зберігатися в хімічних сполуках за допомогою електролізу води на основі відновлюваної енергії для отримання водню. Інші хімічні речовини, такі як аміак і метанол, також можна синтезувати з водню. Зберігання енергії в хімічних речовинах є привабливим, оскільки їхнє виробництво добре вивчене, його можна легко розширити до великих обсягів та отримати вигоду від ефекту масштабу. Крім того, хімічні речовини мають вищий енергетичний вміст, ніж батареї, і є географічно гнучкими з точки зору того, де їх можна виробляти та споживати. У цьому останньому відношенні хімічні речовини можуть діяти як енергоносії, зберігаючи та транспортуючи відновлювану енергію з регіонів надлишкової пропозиції до районів попиту. Енергоносії потім перетворюються назад в електроенергію на вимогу через паливні елементи або газові турбіни. Порівняно з іншими носіями даних, енергоносії мають ще кілька проміжних кроків, і це є існуючий бар'єр витрат, який необхідно подолати.

Інфраструктура, яка координує логістику перевізників, що зберігають і транспортують енергію, є складною енергетичною системою, в якій оптимальний дизайн не є самоочевидним. Серед інших рішень, це вимагає таких ключових міркувань: які відновлювані ресурси використовувати як сировину, які енергоносії виробляти, які типи виробничих потужностей будувати, де будувати, які засоби транспортувати перевізників, куди їх надсилати та що технології перетворення для використання. Для розробки економічно ефективної мережі ланцюга постачання енергоносіїв, яка максимізує потенціал носіїв і є конкурентоспроможною з PSH, CAES та акумуляторами, необхідні методи розробки енергетичних систем. Попередні

роботи розглядають лише окремі енергоносії в ланцюжку поставок, тоді як насправді оптимальний може включати комбінацію кількох. Тут усі варіанти несучих зібрані в надбудову мережі та моделюються за допомогою формулювання MILP. Бінарні змінні визначають розташування і тип виробничих потужностей і технології перетворення, тоді як безперервні змінні пов'язані з мережевими потоками та потужністю електроенергії. Обмеження рівності позначають баланси потоків мережі, тоді як нерівності визначають обмеження ресурсів і логічні обмеження. Загальна цільова функція моделі полягає в тому, щоб мінімізувати вирівняну вартість електроенергії (LCOE), що постачається в місцях попиту.

У попередніх розділах ми описали методології розробки енергетичних систем і показали пару прикладів, щоб підкреслити їх корисність при аналізі системи енергоносіїв. Оскільки джерела енергії та системи продовжують розвиватися, будуть розвиватися й методи розробки енергетичних систем та сфери застосування. За останні роки відбулися захоплюючі зміни в різних дисциплінах і галузях, і перед тим, як завершити цю статтю, ми хочемо торкнутися кількох напрямків, які, на нашу думку, будуть більш активно досліджуватися в найближчі роки інженерами енергетичних систем.

2.7 Інформаційні технології в проектуванні та експлуатації енергетичних систем

Виробничі підприємства збирають великі обсяги робочих даних завдяки вдосконаленим технологіям датчиків і моніторингу. Однак корисність даних обмежена без будь-яких надійних методів інтеграції, класифікації, візуалізації та аналізу даних. У своїй статті 2018 року Едгар і Пістікопулос повідомляють, що багато виробничих операцій у США багаті даними і мають низьку кількість знань. Вони вказують на те, що в той час як операції використовують складні технології моделювання та контролю, використання інструментів аналізу даних на етапі прийняття рішень все ще обмежене. Інтеграції виробничого інтелекту в режимі реального часу для

всієї виробничої операції наразі не існує. Концепція розумного виробництва (SM) визначається як використання правильних даних у правильній формі, правильних технологій і правильних операцій, де та коли це необхідно на всьому виробничому підприємстві. SM поєднує операційні технології з інформаційними технологіями для покращення виробничих платформ. Інтегровані підходи до моделювання, які поєднують датчики та моніторинг, аналіз даних.

У той час як штучний інтелект і машинне навчання були напрямками досліджень протягом багатьох десятиліть, лише нещодавно вони привернули більшу увагу через вибух даних в інформаційну епоху та зростання обчислювальної потужності. Останні помітні досягнення з IBM Watson і AlphaGo від Google навіть викликали інтерес широкої громадськості і дали обіцянки того, як «великі дані» можуть змінити те, як ми розуміємо та вивчаємо світ. Однак, як і у випадку з будь-якою новою технологічною розробкою, можливо, було б краще залишатися обережним оптимістом, щоб не перебільшувати плоди та не перебільшувати ризики, оскільки ще потрібно зробити багато прогресу, щоб штучний інтелект і машинне навчання дозріли, проникли та поширилися на більшого прийняття. Ми очікуємо, що з достатньою кількістю часу та спрямованими зусиллями досягнення в спільноті штучного інтелекту та машинного навчання перейдуть у розробку енергетичних систем і стануть такими ж широко використовуваними методологіями, як змішана цілочисельна оптимізація та МРС за останні кілька десятиліть. Такі галузі досліджень, як розробка каталізатора та відкриття ліків активно розробляють відкриті бази даних для моделювання побудови гібридної моделі на основі даних. Штучний інтелект і машинне навчання будуть особливо потрібні в таких областях, як експлуатація процесів і діагностика несправностей, де традиційні математичні підходи не підходять через відсутність першоосновної основи для розробки моделі.

Зараз існує більше ресурсів і доступного програмного забезпечення для вивчення та впровадження програм штучного інтелекту та машинного навчання порівняно з попередніми епохами. Зростаюча кількість таких інструментів дозволяє користувачам легко кодувати власне машинне навчання або моделі нейронних мереж, не докладаючи зусиль. У своєму цікавому погляді на статус штучного інтелекту в хімічній інженерії Венкатасубраманян зазначає, що існує ризик замінити добре навчене використання таких інструментів фактичним оволодінням знаннями зі штучного інтелекту та машинного навчання. Ми погоджуємося з його оцінкою, що майбутнім інженерам важливо мати належну освіту «ноу-чому», а не лише «ноу-хау». Ми вважаємо, що існує необхідність реформувати поточну інженерну програму та навчання для випускників, щоб включити більше матеріалів зі штучного інтелекту та машинного навчання, щоб краще підготувати людей до розвитку енергетичних систем майбутнього.

2.8 Огляд технології LoRa

Мережа LoRaWAN складається з наступних елементів: кінцевий пристрій, шлюзи, мережевий сервер та сервер додатків. **Кінцевий пристрій** призначений для здійснення керуючих або вимірювальних функцій. Містить набір необхідних датчиків та керуючих елементів.

Шлюз – пристрій, який приймає дані від кінцевих пристроїв за допомогою радіоканалу та передає їх у транзитну мережу. Як транзитна мережа можуть виступати мережа Ethernet, WiFi або мережі рухомого радіотелефонного зв'язку. Шлюз та кінцеві пристрої утворюють мережеву топологію типу зірка. Зазвичай цей пристрій містить багатоканальні приймачі для обробки сигналів в декількох каналах одночасно або навіть декількох сигналів в одному каналі. Відповідно, кілька таких пристроїв забезпечує зону радіопокриття мережі та прозору двонаправлену передачу даних між кінцевими пристроями та сервером.

Мережевий сервер – призначений управління мережею: завданням розкладу, адаптацією швидкості, зберіганням і обробкою прийнятих даних.

Сервер програм – може віддалено контролювати роботу кінцевих пристроїв та збирати необхідні дані з них.

Зрештою, LoRaWAN мережа має топологію зірка зі зірок, має кінцеві пристрої, які через шлюзи, що утворюють прозорі мости, спілкуються з центральним сервером мережі. За такого підходу зазвичай передбачається, що шлюзами і центральним сервером володіє оператор мережі, а кінцевими пристроями – абоненти. Абоненти мають можливість прозорої двонаправленої та захищеної передачі даних до кінцевих пристроїв.

2.8.1 Сильні та слабкі сторони технології

Переваги LoRaWAN:

- велика дальність передачі радіосигналу в порівнянні з іншими бездротовими технологіями, що використовуються для телеметрії, досягає 15 км.
- низьке енергоспоживання у кінцевих пристроїв завдяки мінімальним витратам енергії на передачу невеликого пакета даних;
- висока проникаюча здатність радіосигналу в міській забудові під час використання частот субгігагерцового діапазону.
- масштабованість мережі на великих територіях.
- відсутність необхідності отримання частотного дозволу та плати за радіочастотний спектр, внаслідок використання неліцензованих частот (ISM band).

Недоліки LoRaWAN:

- Відносно низька пропускна здатність, що варіюється в залежності від використовуваної технології передачі даних на фізичному рівні, становить від декількох сотень біт/с до декількох десятків кбіт/с.

– Затримка передачі даних від датчика до кінцевого додатка, пов'язана з часом передачі радіосигналу, може досягати від кількох секунд до кількох десятків секунд.

– Відсутність єдиного стандарту, який визначає фізичний шар та керування доступом до середовища для бездротових LPWAN-мереж.

– Ризики шуму спектру неліцензованого діапазону частот.

– Пропріетарна технологія модуляції LoRa, "закрита" патентом Semtech.

– Обмеження потужності сигналу.

Для успішного функціонування будь-якої системи обміну інформацією необхідна взаємна синхронізація приймача та передавача, що дозволяє визначити часові межі приймання-передачі цілого блоку даних (або кадру), так і одиничних символів.

Технологія LoRa використовує асинхронний режим приймання-передачі, у якому передавач може почати генерацію радіосигналу в будь-який момент часу. В цьому випадку потрібен механізм, що забезпечує синхронізацію приймача сигналом від передавача (аналог "старт-біта" протоколу RS232). Як такий механізм використовується преамбула, що передуює кожному сеансу зв'язку. Преамбула включає послідовність символів, що дозволяють приймачеві виявити активність передавача, визначити коефіцієнт розширення спектру (SF), що використовується передавачем, і виконати символну синхронізацію.

2.8.2 Класи пристроїв LoRa

Для вирішення різних завдань та застосувань в мережі LoRaWAN передбачено три класи кінцевих пристроїв:

- **Кінцеві пристрої класу А** (Bi-directional end-devices, Class A). Кінцеві пристрої "класу А" дозволяють організувати двонаправлений обмін. Причому зв'язок може ініціювати тільки кінцевий пристрій, після чого виділяються два часові вікна, протягом яких очікується

відповідь від мережі. Інтервал передачі планується кінцевим пристроєм на основі власних потреб у зв'язку з невеликими тимчасовими випадковими флуктуаціями (протокол типу ALOHA). Кінцеві пристрої «класу А» застосовуються в додатках, де передача даних від мережі можлива тільки як реакція у відповідь на отримання даних від кінцевого пристрою і потрібен максимальний час роботи від автономного джерела живлення.

- **Двонаправлені кінцеві пристрої «класу Б»** (Bi-directional end-devices, Class B) на додаток до функцій пристроїв «класу А» відкривають додаткові вікна прийому за розкладом. Щоб відкрити вікно прийому, кінцевий пристрій синхронізується за спеціальними сигналами від шлюзу (по маяках – Beacon). Це дозволяє мережі знати час, коли кінцевий пристрій готовий приймати дані.
- **Двонаправлені кінцеві пристрої "класу С"** з максимальним приймальним вікном (Bi-directional end-devices, Class C). Кінцеві пристрої класу С мають майже безперервно відкрите вікно прийому. Приймне вікно закривається лише на час передачі даних. Цей тип кінцевих пристроїв підходить для завдань, коли необхідно отримувати великі обсяги даних, і не потрібна тривала робота від автономного джерела живлення.

2.8.3 Фізичний рівень (PHY Layer)

Фізично забезпечується негарантована передача блоків даних між кінцевим пристроєм (End Node) і шлюзом LoRa (Gateway). На стороні передавального пристрою виконується:

- прийом блоку даних від MAC рівня (PHYPayload);
- формування фізичного заголовка пакета (PHDR + PHDR_CRC);
- кодування фізичного заголовка пакета (PHDR + PHDR_CRC) із фіксованою швидкістю 4/8;
- контрольної суми блоку корисних даних PHYPayload (CRC);

- кодування блоку корисних даних (PHYPayload + CRC) із встановленою швидкістю CR;
- передача радіоканалом преамбули;
- модуляція та передача по радіоканалу фізичного блоку даних.

На стороні приймального пристрою виконується:

- виявлення преамбули та визначення початку фізичного блоку даних;
- демодуляція сигналу;
- фізичного заголовка пакета (PHDR + PHDR_CRC) та перевірка його контрольної суми;
- декодування блоку корисних даних (PHYPayload + CRC) та перевірка його контрольної суми;
- підтвердження ухвалених даних (для відповідних типів повідомлень);
- Передача даних на MAC рівень.

На рисунках нижче наведено формати фізичних блоків даних низхідного (DL) та висхідного (UL) каналів:

2.8.4 Підтвердження отримання повідомлень

Технологія LoRa визначає два типи повідомлень – повідомлення, що потребують підтвердження отримання та повідомлення без підтвердження. Тип повідомлення – Confirmed (UL/DL) / Unconfirmed (UL/DL) визначається значенням поля MType (MessageType) заголовка MAC рівня.

Якщо відправником повідомлення, що вимагає підтвердження, є кінцевий пристрій (End Node), мережа підтверджує отримання такого повідомлення всередині вікон прийому, відкритих кінцевим пристроєм відразу після сеансу передачі. Якщо відправником повідомлення, що вимагає підтвердження, є мережа (LoRa gateway-шлюз), момент передачі підтвердження визначається кінцевим пристроєм (End Node).

Підтвердження може бути надіслано негайно (в т.ч. у складі порожнього повідомлення), що спрощує логіку функціонування End Node, або у складі чергового повідомлення, яке несе корисне навантаження, що

скорочує завантаження радіоканалу. У будь-якому випадку, підтверджується завжди лише останнє отримане повідомлення. Повідомлення, що є підтвердженням, характеризується встановленим бітом АСК заголовка рівня MAC. Повторну передачу підтверджень не передбачено. Необхідність повторної передачі непідтверджених повідомлень (або його видалення), а також моменти передачі і кількість повторів визначається логікою функціонування мережного сервера і кінцевого пристрою відповідно. При кожній повторній передачі можливе зниження швидкості потоку даних (data rate), що підвищує перешкодозахищеність. Також передбачена можливість провіжинінгу параметрів повторної передачі кінцеві пристрої з боку мережі.

У разі неотримання мережним сервером попередньо встановленого числа підтверджень від кінцевого пристрою, цей кінцевий пристрій може бути промаркований як недоступний (unreachable) аж до отримання будь-якого першого вхідного повідомлення.

2.8.5 Адаптивна швидкість передачі (Adaptive Data Rate – ADR)

У технології LoRa передбачені механізми адаптації швидкості передачі даних кінцевих пристроїв для того, щоб оптимізувати завантаження мережі та забезпечити кожному кінцевому пристрою можливість роботи на максимальних швидкостях, що забезпечують належну перешкодостійкість у тих радіоумовах, в яких даний пристрій знаходиться. Адаптацію швидкості передачі кінцевих пристроїв (End Node) виконує мережевий сервер за допомогою відповідних MAC команд. Рішення про вибір тієї чи іншої швидкості приймається виходячи з оцінки якості прийнятого від End Node сигналу.

Механізми адаптації швидкості доречно використовувати лише з пристроях, місце розташування яких і змінюється з часом (статичні устрою), т.к. для таких пристроїв та радіоумови загалом будуть достатньо стабільні від одного сеансу зв'язку до іншого. На мобільних пристроях, наприклад, встановлених на автомобілях, тваринах та ін. радіоумови між сеансами

зв'язку змінюються непередбачено. Отже, на таких пристроях доречно використовувати постійні (встановлені по дефолту) швидкості передачі. Статичні пристрої повинні ініціювати використання мережею режиму адаптації за допомогою установки ADR біта заголовка рівня MAC.

Якщо кінцевий пристрій використовує швидкість передачі даних вище встановленої дефолтної швидкості (відповідно до команди MAC рівня, отриманої від мережевого сервера), воно має періодично контролювати факт отримання мережею повідомлень (навіть при використанні режиму передачі без підтвердження) відповідно до наступної процедури:

- кінцеве пристрій (End Node) інкрементує лічильник ADR_ACK_CNT при кожному переданому у висхідному каналі повідомленні (UL-Msg) і скидає його при отриманні вхідного повідомлення по низхідному каналу (DL-Msg) у вікні прийому (receive window);

- при досягненні лічильником ADR_ACK_CNT порога ADR_ACK_LIMIT кінцевий пристрій (за допомогою установки біта ADRACKReq) запитує мережу направити йому будь-який DL-Msg, підтвердивши тим самим, що повідомлення цього кінцевого пристрою досягають мети; підтвердження має бути надіслане у вікні прийому одного з наступних UL-Msg (але не більше ніж задано порогом ADR_ACK_DELAY);

- за відсутності підтвердження кінцевий пристрій знижує швидкість передачі на один крок;

- подальше зниження швидкості передачі на один крок відбуватиметься після передачі кожних ADR_ACK_LIMIT UL-Msg до отримання підтвердження, або до досягнення попередньо встановленої дефолтної швидкості.

2.8.6 Синхронізація приймача та передавача

Для успішного функціонування будь-якої системи обміну інформацією необхідна взаємна синхронізація приймача та передавача, що дозволяє визначити часові межі приймання-передачі цілого блоку даних (або кадру),

так і одиничних символів. Технологія LoRa використовує асинхронний режим приймання-передачі, у якому передавач може почати генерацію радіосигналу в будь-який момент часу. В цьому випадку потрібен механізм, що забезпечує синхронізацію приймача сигналом від передавача (аналог "старт-біта" протоколу RS232). Як такий механізм використовується преамбула, що передує кожному сеансу зв'язку. Преамбула включає послідовність символів, що дозволяють приймачеві виявити активність передавача, визначити коефіцієнт розширення спектру (SF), що використовується передавачем, і виконати символну синхронізацію. Тривалість преамбули є величиною, що конфігурується, і повинна бути не менше, ніж $T_1 + 2 \cdot T_2$, де T_1 визначає максимальний час знаходження приймача у стані "сну" (Sleep), T_2 – визначає час пошуку приймачем преамбули (рис. 2.2).

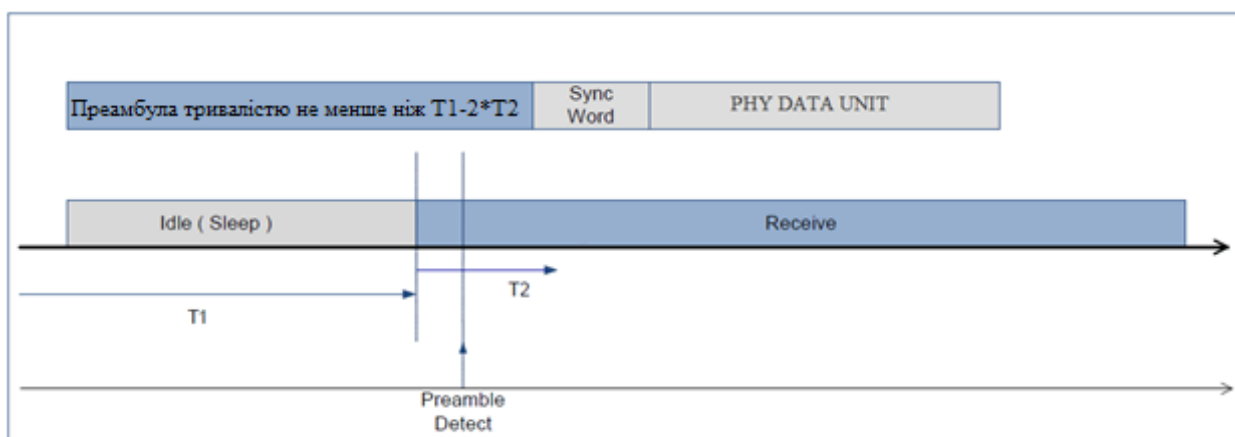


Рисунок 2.2 – Час пошуку приймачем преамбули

Після завершення преамбули слідує слово синхронізації (Sync Word) і блок даних фізичного рівня. Довжина слова синхронізації налаштовується від 1 до 8 байт. Специфікацією LoRa визначено ряд специфічних значень Sync Word – 0x34 для публічних мереж (public networks), 0x12 – для приватних мереж (private networks) та 0xC194C1 – для каналів із FSK модуляцією. На рис. 2.3 наведено загальну структуру кадру, що забезпечує передачу одного блоку даних.

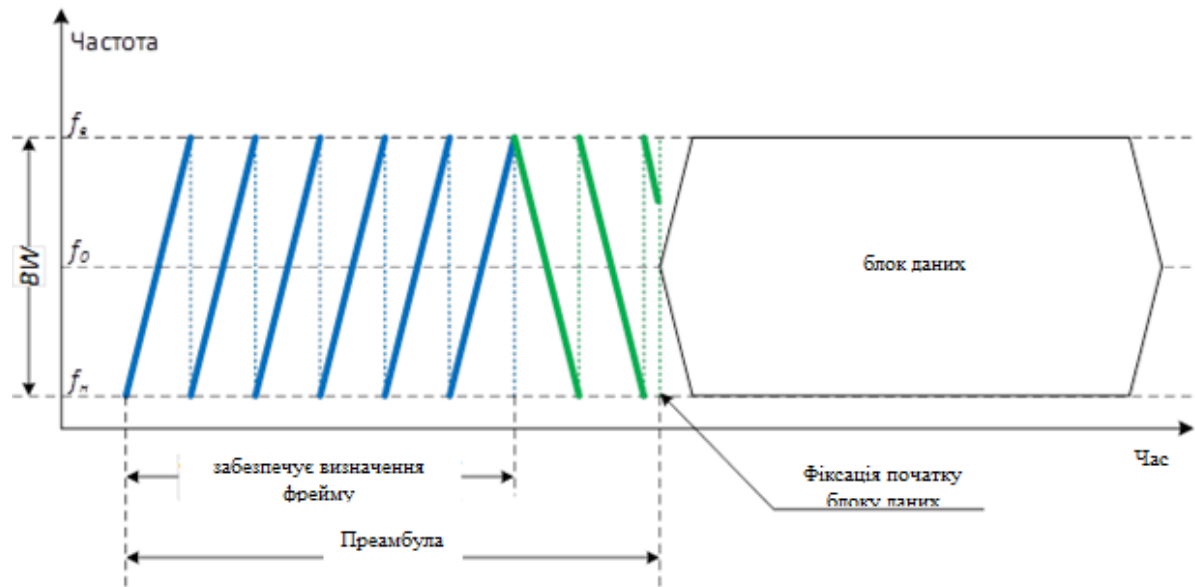


Рисунок 2.3 – Загальна структура кадру

2.8.7 Детектування CSS сигналу

Механізм функціонування детектора преамбули заснований на використанні узгодженого фільтра (СФ), імпульсна характеристика якого комплексно пов'язана з CSS радіосигналом в частотній області і має дзеркальне відображення його в часі:

$$h(t) = A_1 \cdot \cos\left(\omega_n \cdot (T_{sym} - t) - \frac{\mu}{2} \cdot (T_{sym} - t)^2\right), 0 \leq t < T_{sym}$$

Принцип передачі символів інформації блоку даних фізичного рівня (PHY DATA UNIT) за допомогою широкосмугового радіосигналу LoRa полягає в частотному зсуві $e^{j \cdot \Delta\omega \cdot k \cdot t}$ щодо опорного ЛЧМ радіосигналу $e^{j \cdot (\omega_n \cdot t + \mu \cdot t^2)}$, де $k=0,1,2,\dots,2^{SF}$ – інформаційний символ, розмірністю SF біт (рис. 2.4):

$$x(t) = \begin{cases} A_0 \cdot \cos\left(\omega_n \cdot t + \Delta\omega \cdot k \cdot t + \frac{\mu}{2} \cdot t^2\right), & 0 \leq t < T_0 \\ A_0 \cdot \cos\left(\omega_n \cdot t + \Delta\omega \cdot k \cdot t - BW \cdot t + \frac{\mu}{2} \cdot t^2\right), & T_0 \leq t < T_{sym} \end{cases}$$

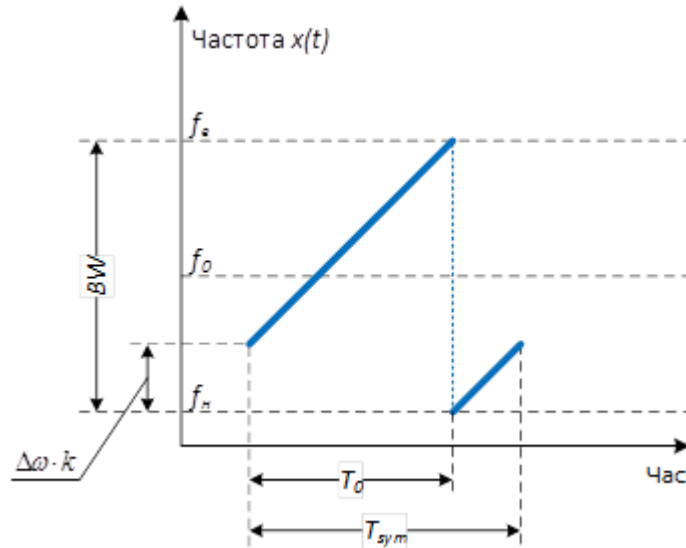


Рисунок 2.4 – Інформаційний символ, розмірністю SF біт

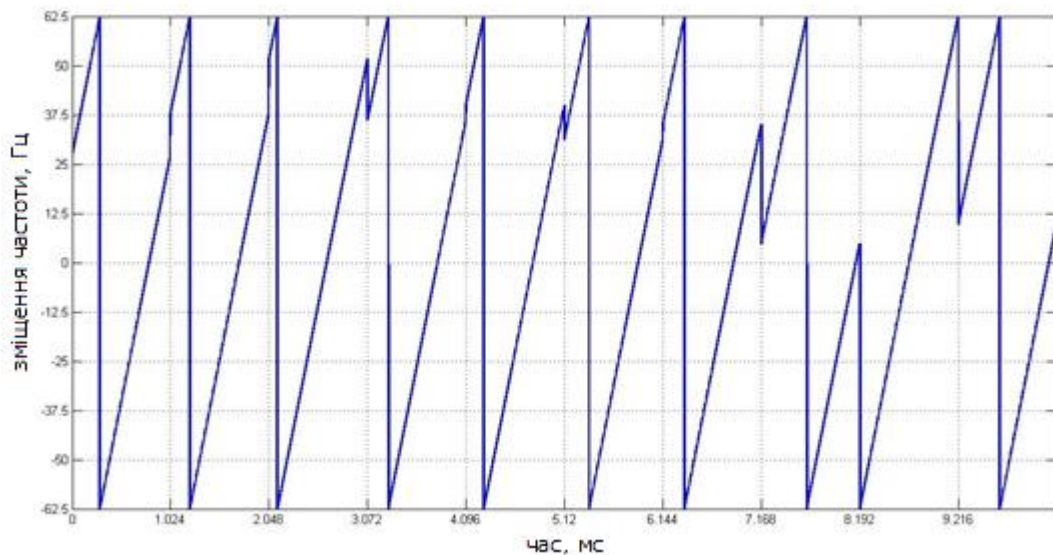


Рисунок 2.5 – Приклад залежності частоти радіосигналу від часу LoRa кадру

Тут:

$$c(t) = A_1 \cdot \cos\left(\omega_n \cdot t + \frac{\mu}{2} \cdot t^2\right), \quad 0 \leq t < T_{sym}$$

– Еталонний ЛЧМ сигнал

– Адитивний білий Гаусівський шум

Де – chirped сигнал:

$$y(t) = x(t) \cdot c(t) =$$

$$= \frac{A_0 \cdot A_1}{2} \cdot \begin{cases} \cos(\Delta\omega \cdot k \cdot t) + \cos(2 \cdot \omega_H \cdot t + \Delta\omega \cdot k \cdot t + \mu \cdot t^2), & 0 \leq t < T_0 \\ \cos(BW - \Delta\omega \cdot k \cdot t) + \cos(2 \cdot \omega_H \cdot t + \Delta\omega \cdot k \cdot t - BW \cdot t + \mu \cdot t^2), & T_0 \leq t < T_{sym} \end{cases}$$

$$n_y(t) = n(t) \cdot c(t).$$

$$n(t), \quad 0 \leq t < T_{sym}$$

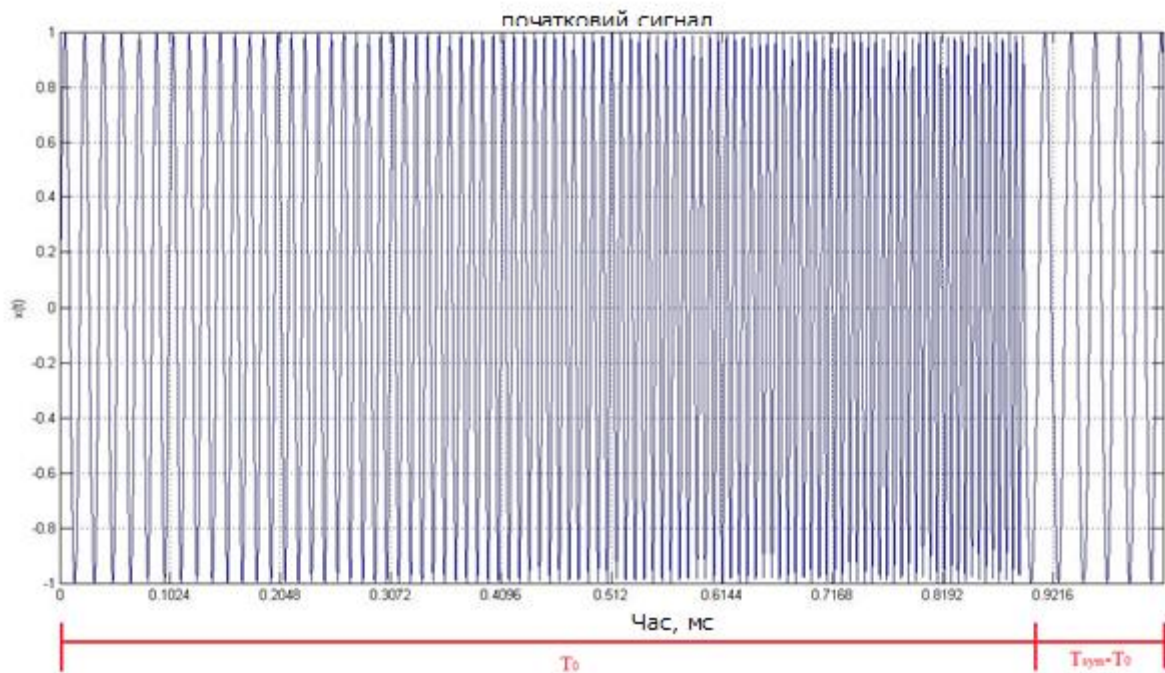


Рисунок 2.6 – початковий сигнал

Ключовою особливістю радіоінтерфейсу LoRa (як уже згадувалося вище) є його висока стійкість до перешкод. Рисунки нижче демонструють функціонування описаного детектора сигналу LoRa в умовах адитивного білого шуму гаусовського (відношення сигнал/шум SNR=0dB).

У технології LoRa передбачені механізми адаптації швидкості передачі даних кінцевих пристроїв для того, щоб оптимізувати завантаження мережі та забезпечити кожному кінцевому пристрою можливість роботи на максимальних швидкостях, що забезпечують належну перешкодостійкість у тих радіоумовах, в яких даний пристрій знаходиться.

Адаптацію швидкості передачі кінцевих пристроїв (End Node) виконує мережевий сервер за допомогою відповідних MAC команд.

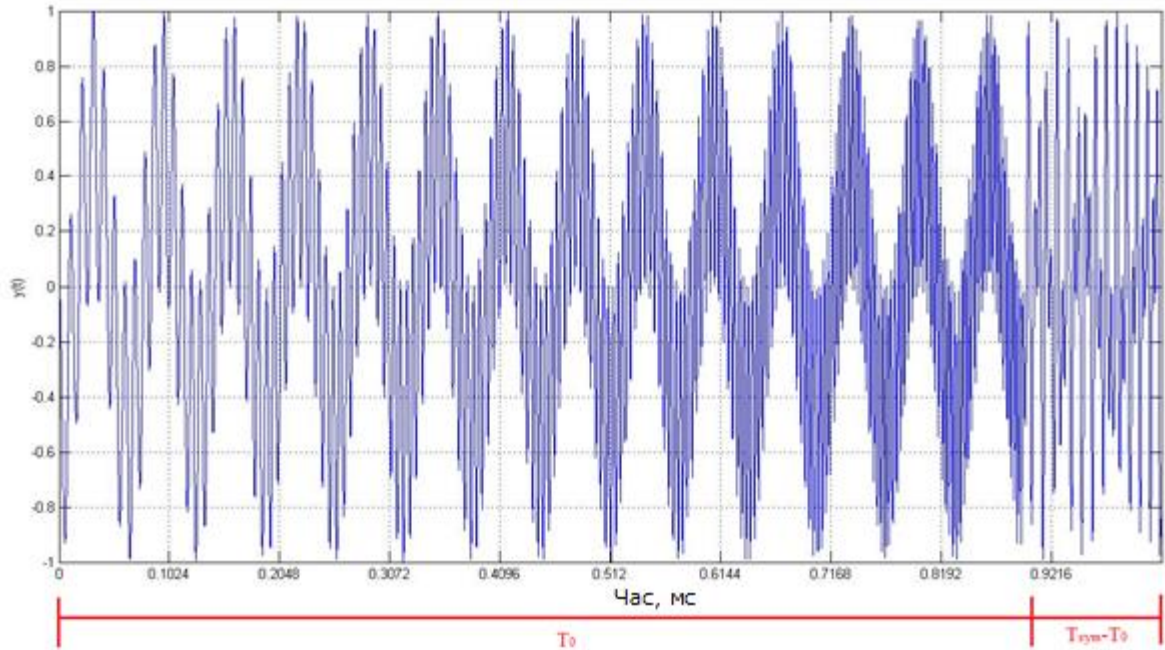


Рисунок 2.7 – де-chirped сигнал

Відкинувши у виразі для $y(t)$ другі доданки у фігурних дужках (як високочастотні складові):

$$y(t) = \frac{A_0 \cdot A_1}{2} \cdot \begin{cases} \cos(\Delta\omega \cdot k \cdot t), & 0 \leq t < T_0 \\ \cos([BW - \Delta\omega \cdot k] \cdot t), & T_0 \leq t < T_{sym} \end{cases}$$

на виході блоку перетворення Фур'є (FFT +) отримуємо наступний комплексний сигнал:

$$\begin{aligned} Y(\omega) &= \int_{-\infty}^{\infty} y(t) \cdot e^{-i\omega t} dt = \\ &= \frac{A_0 \cdot A_1}{4} \times e^{-j(\omega - \Delta\omega k) \frac{T_0}{2}} \cdot T_0 \cdot \frac{\sin[(\omega - \Delta\omega k) \cdot T_0/2]}{(\omega - \Delta\omega k) \cdot T_0/2} + \\ &+ \frac{A_0 \cdot A_1}{4} \times e^{-j(\omega - (BW - \Delta\omega k)) \frac{T_0 + T_{sym}}{2}} \cdot (T_{sym} - T_0) \cdot \frac{\sin[(\omega - (BW - \Delta\omega k)) \cdot (T_{sym} - T_0)/2]}{(\omega - (BW - \Delta\omega k)) \cdot (T_{sym} - T_0)/2} + \\ &+ \frac{A_0 \cdot A_1}{4} \times e^{-j(\omega + \Delta\omega k) \frac{T_0}{2}} \cdot T_0 \cdot \frac{\sin[(\omega + \Delta\omega k) \cdot T_0/2]}{(\omega + \Delta\omega k) \cdot T_0/2} + \\ &+ \frac{A_0 \cdot A_1}{4} \times e^{-j(\omega + (BW - \Delta\omega k)) \frac{T_0 + T_{sym}}{2}} \cdot (T_{sym} - T_0) \cdot \frac{\sin[(\omega + (BW - \Delta\omega k)) \cdot (T_{sym} - T_0)/2]}{(\omega + (BW - \Delta\omega k)) \cdot (T_{sym} - T_0)/2} \end{aligned}$$

Далі позбавляємося двох останніх доданків, що мають істотний вплив в області негативних частот і низьке в області позитивних:

$Y^+(\omega) = Y_1(\omega) + Y_2(\omega)$, где

$$Y_1(\omega) = \frac{A_0 \cdot A_1}{4} \times e^{-j(\omega - \Delta\omega k) \frac{T_0}{2}} \cdot T_0 \cdot \frac{\sin[(\omega - \Delta\omega k) \cdot T_0/2]}{(\omega - \Delta\omega k) \cdot T_0/2} +$$

$$Y_2(\omega) = \frac{A_0 \cdot A_1}{4} \times e^{-j(\omega - (BW - \Delta\omega k)) \frac{T_0 + T_{sym}}{2}} \cdot (T_{sym} - T_0) \cdot \frac{\sin[(\omega - (BW - \Delta\omega k)) \cdot (T_{sym} - T_0)/2]}{(\omega - (BW - \Delta\omega k)) \cdot (T_{sym} - T_0)/2}$$

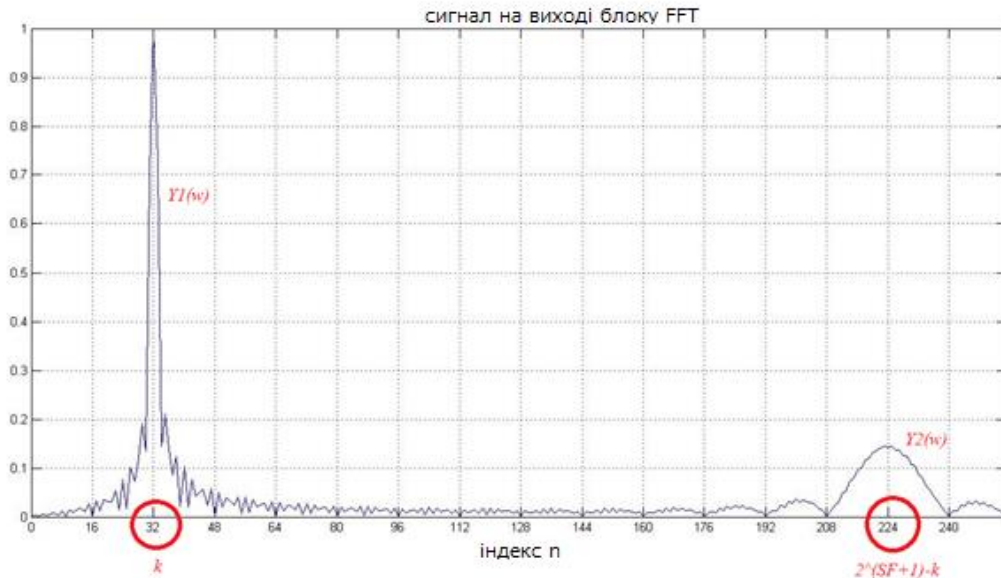


Рисунок 2.8 – Сигнал на виході блоку FFT

Щоб уникнути перекриття двох доданків $Y^+(\omega)$ при різних значеннях

k має виконуватися нерівність: $\Delta\omega < \frac{BW}{2 \cdot k}$.

$$\Delta\omega = \frac{BW}{2^{SF+1}}$$

$$T_0 = \frac{(2^{SF+1} - k)}{2^{SF+1}} \cdot T_{sym}$$

На наступному етапі обчислюється функція прийняття рішення $Z(\omega)$, що є сумою модулів функції $Y_1(\omega)$ та функції $Y_2(\omega)$, дзеркально

$\omega = BW$: відображеної щодо точки

$$Z(\omega) = Y_1(\omega) + Y_2(BW - \omega) \approx Y^+(\omega) + Y^+(BW - \omega)$$

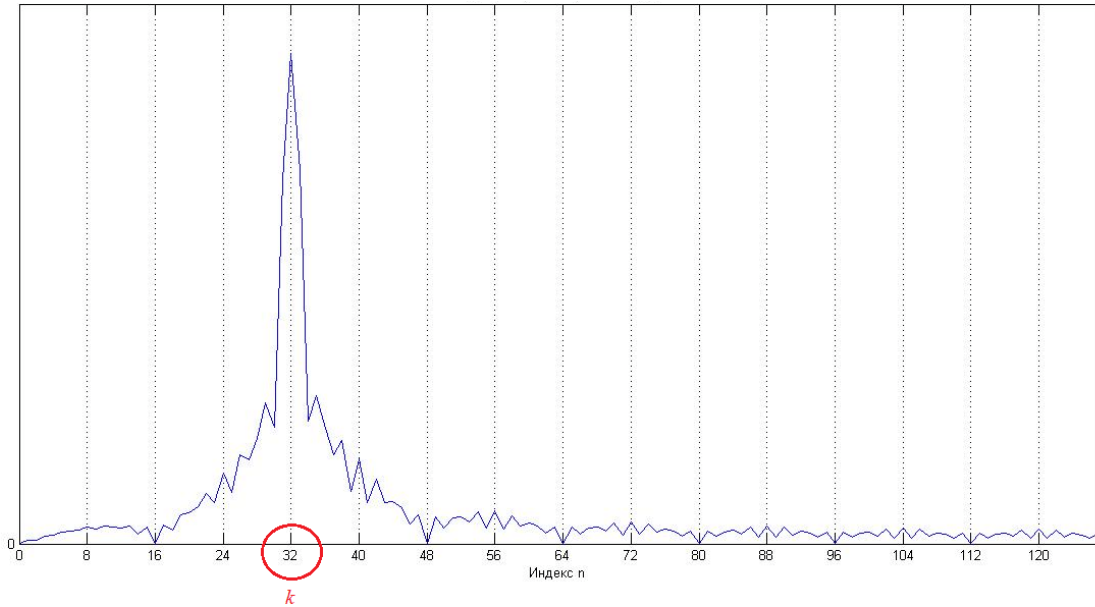


Рисунок 2.9 – Функція прийняття рішень $Z(n)$

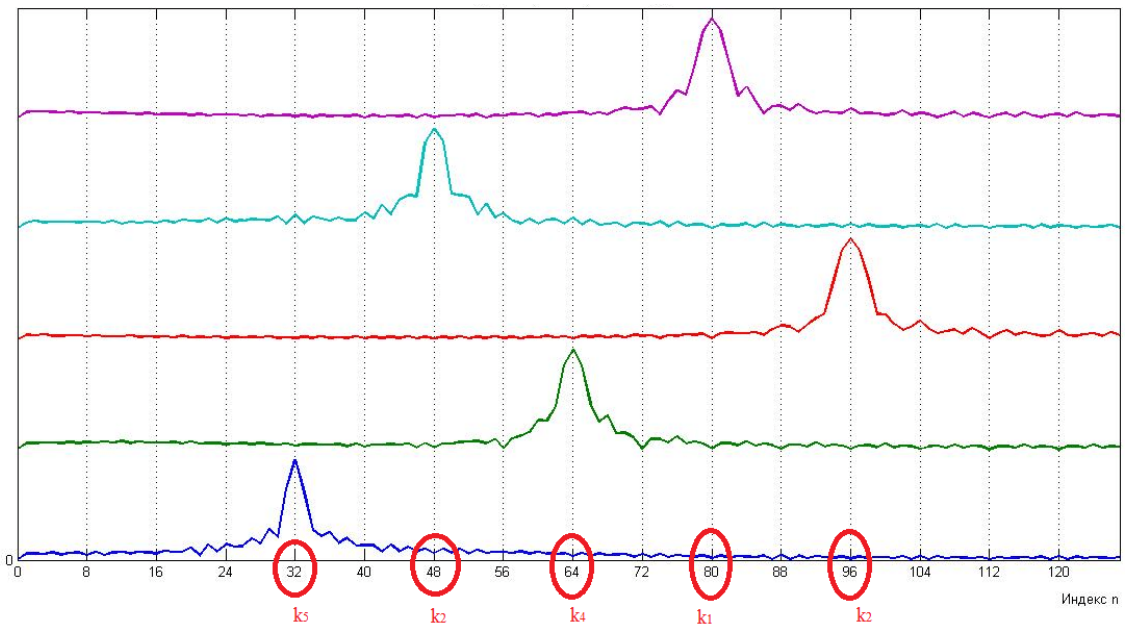


Рисунок 2.10 – Функція прийняття рішень $Z(n)$

Нарешті, визначаємо значення декодованого приймачем інформаційного символу k . Для цього знаходимо частоту ω , при якій функція прийняття рішення $Z(\omega)$ набуває максимального значення (ω_{max}):

$$Z(\omega = \omega_{max}) = \text{MAX}[Z(\omega)]$$

$$k = \frac{\omega_{max}}{\Delta\omega}$$

Ключовою особливістю радіоінтерфейсу LoRa (як згадувалося вище) є його висока завадостійкість. Рисунки нижче демонструють функціонування

описаного детектора сигналу LoRa в умовах адитивного білого шуму гаусовського (відношення сигнал/шум $SNR=0dB$).

Висновки до розділу 2

Проаналізувавши інформацію щодо існуючої технології Lora, можна зробити висновок, що дана технологія є доцільним рішенням для реалізації поставленої задачі в даній дипломній роботі. В цьому можна впевнитися, звернувши увагу на такі переваги технології:

- LoRa використовує частотний діапазон, дозволений для використання у Росії. Існують системи LoRa для діапазону 433 МГц, але в нашій країні більше прижилися частоти з діапазону 868. Там у нас є 1,5 МГц спектра, що не ліцензується. 864-865 МГц та 868,7-869,2. У першому інтервалі ми маємо обмеження за часом знаходження передавача в ефірі (не більше 0,1%) та за потужністю (не більше 25 мВт). У другому – лише за потужністю (ті самі 25 мВт). Також є застереження щодо використання поблизу аеропортів. Як уже згадувалося, ширина каналу LoRa – 125 кГц. LoRa розроблялася для роботи на потужності 25 мВт. І тут не порушуємо. Нехай вас не бентежить такий низький рівень – технологія може працювати нижче за рівень шуму.
- LoRa це відкритий стандарт. Чіпи для кінцевих пристроїв у вільному продажу, є вся документація, і вона відкрита будь-кому. Датчики та радіомодулі під цей стандарт лише в Росії роблять кілька компаній. Вона не «річ у собі», навіть якщо пропаде один із виробників, залишаться інші.
- LoRa має гарний радіус дії, вона може приймати інформацію від пристроїв у підвалі будинку або за кілометр від базової станції. Насправді може прийняти інформацію і від датчика в 4 кілометрах міських умов. Але тут страждає стабільність, оскільки починається втрата пакетів.

РОЗДІЛ 3 РОЗРОБКА МОДЕЛІ ТА СПОСОБИ ТЕСТУВАННЯ

3.1 Способи тестування програмного забезпечення

Тестування системи є дуже важливою частиною при розробці проектів подібного типу. Оскільки розроблюваний прилад буде працювати з особистими даними людей, та чіпами в яких вони записані, потрібно бути впевненим, що під час роботи приладу вірогідність виникнення непередбачуваних ситуацій, що можуть призвести до втрати цих даних, буде мінімальною. Розділяють наступні види тестування програмного забезпечення:

Приймальне тестування. Формальне тестування, що проводиться з метою визначення того, чи задовольняє система критеріям прийнятності та дозволяє клієнту визначити, приймати чи не приймати систему. Зазвичай це виконується замовником.

Тестування доступності: тип тестування, який визначає придатність продукту для людей з обмеженими можливостями (глухих, сліпих, розумово відсталих і т. д.). Процес оцінки проводиться особами з обмеженими можливостями.

Активне тестування: тип тестування, що складається з вступу тестових даних і аналізу результатів виконання. Зазвичай проводиться командою тестування.

Вікове тестування: тип тестування, який оцінює здатність системи виконувати в майбутньому. Процес оцінки проводиться групами тестування.

Спеціальне тестування: тестування виконується без планування і документування – тестувальник намагається «зламати» систему шляхом випадкового випробування її функціональних можливостей. Це виконується командою тестування.

Альфа-тестування: тип тестування програмного продукту або системи, що проводиться на сайті розробника. Зазвичай це виконується кінцевими користувачами.

Підтвердження тестування: тип тестування, що складається в перевірці, чи підтверджують умови вимоги продукту. Це виконується командою тестування.

Тестування API: Техніка тестування схожа на модульне тестування в тому сенсі, що вона націлена на рівень коду. Api Testing відрізняється від Unit Testing тим, що зазвичай це завдання QA, а не завдання розробника.

Тестування всіх пар: комбінаторний метод тестування, який перевіряє всі можливі дискретні комбінації вхідних параметрів. Це виконується командами тестування.

Автоматичне тестування. Техніка тестування, що використовує інструменти автоматичного тестування для управління налаштуванням середовища, виконанням тесту і звітування про результати. Він виконується комп'ютером і використовується всередині команд тестування.

Основне тестування шляху: Механізм тестування, який виводить логічну міру складності процедурного задуму і використовує його в якості керівництва для визначення базового набору шляхів виконання. Він використовується командами тестування при визначенні тестових випадків.

Бета-тестування: фінальне тестування перед випуском програми для комерційних цілей. Зазвичай це робиться кінцевими користувачами або іншими.

Тестування продуктивності: метод тестування, в якому використовуються типові набори програм і даних, призначені для оцінки продуктивності комп'ютерного обладнання та програмного забезпечення в заданій конфігурації. Це виконується командами тестування.

Тестування двійковій переносимості: метод, який перевіряє виконується додаток на переносимість між системними платформами і середовищами, зазвичай на відповідність специфікації ABI. Це виконується командами тестування.

Тестування чорного ящика: метод тестування програмного забезпечення, який перевіряє функціональність програми, не маючи спеціальних знань про код/внутрішній структурі програми. Тести засновані на вимогах і функціональності. Це виконується командами QA.

Тестування на основі коду: метод тестування, який використовує тестові середовища (такі як xUnit), які дозволяють виконувати модульні тести, щоб визначити, чи працюють різні розділи коду, як очікується, при різних обставинах. Це виконується командами розробників.

Тестування сумісності: методика тестування, яка перевіряє, наскільки добре програмне забезпечення працює в певному апаратному / програмному / операційному / мережевому оточенні.

Це виконується командами тестування.

Порівняльне тестування: методика тестування, яка порівнює сильні і слабкі сторони продукту з попередніми версіями або іншими подібними продуктами. Може виконуватися тестером, розробниками, менеджерами по продуктах або власниками продуктів.

Компонентне тестування: методика тестування, схожа на модульне тестування, але з більш високим рівнем інтеграції – тестування проводиться в контексті програми, а не просто шляхом безпосереднього тестування певного методу. Може бути виконаний командами тестування або розробки.

Тестування конфігурації: метод тестування, який визначає мінімальну і оптимальну конфігурацію апаратного і програмного забезпечення, а також ефект додавання або зміни ресурсів, таких як пам'ять, дисководи і процесор. Зазвичай це виконується інженерами Performance Testing.

Тестування покриття умов: тип тестування програмного забезпечення, при якому кожна умова виконується, роблячи його істинним і хибним, в кожному із способів хоча б один раз. Це зазвичай робиться командами тестування автоматизації.

Тестування на відповідність: тип тестування, який перевіряє, чи була система розроблена відповідно до стандартів, процедурами і керівними принципами. Зазвичай це виконується сторонніми компаніями, які пропонують бренд «Certified OGC Compliant».

Паралельне тестування: багато користувачів тестування, спрямоване на визначення ефектів доступу до одного і того ж коду додатка, модулю або записів бази даних. Зазвичай це роблять інженери.

Тестування відповідності: процес тестування відповідності реалізації специфікації, на якій вона заснована. Це зазвичай виконується командами тестування.

Тестування на основі контексту: техніка гнучкого тестування, яка підтримує безперервну і творчу оцінку можливостей тестування в світлі розкритою потенційної інформації та цінності цієї інформації для організації в певний момент. Це зазвичай виконується Agile командами тестування.

Тестування перетворення: Тестування програм або процедур, що використовуються для перетворення даних з існуючих систем для використання в системах заміни. Це зазвичай виконується командами QA.

Тестування покриття рішень: тип тестування програмного забезпечення, при якому кожна умова / рішення виконується шляхом установки його в значення true/false. Це, зазвичай, робиться командами автоматизації тестування.

Тестування залежностей: тип тестування, який перевіряє вимоги додатка до вже існуючого програмного забезпечення, початковим станом і конфігурації для підтримки належної функціональності. Це зазвичай виконується командами тестування.

Тестування предметної області: метод тестування білого ящика, який містить перевірки того, що програма приймає тільки допустимі дані. Зазвичай це робиться командами розробників програмного забезпечення, а іноді і командами автоматизації тестування.

Тестування обробки помилок: тип тестування програмного забезпечення, який визначає здатність системи правильно обробляти помилкові транзакції. Це зазвичай виконується командами тестування.

Наскрізне тестування. Подібно системному тестування, включає в себе тестування повною середовища додатків в ситуації, що імітує реальне використання, наприклад взаємодія з базою даних, використання мережевих комунікацій або взаємодія з іншим обладнанням, додатками або системами. у разі необхідності. Це виконується командами QA.

Тестування на витривалість: тип тестування, який перевіряє виток пам'яті або інші проблеми, які можуть виникнути при тривалому виконанні. Це зазвичай виконується виконавчими інженерами.

3.2 Обґрунтування обраних елементів

3.2.1 RYLR896

Модуль приймача-передавача RYLR890/RYLR896 оснащений модемом дальнього радіусу дії LoRa, який забезпечує зв'язок з наддальнім спектром поширення та високим інтерференційним імунітетом, мінімізуючи споживання струму. Раніше ми дізналися про LoRa Модуль SX1278 і його взаємозв'язання з Arduino через SPI піни. Модуль UART LoRa можна використовувати з будь-яким мікроконтролером, використовуючи всього 4 штифти.

Модуль Lora RYLR890/RYLR896 передає до діапазону 10-15 км. Він розроблений з використанням найкращої технології шумозаглушення. Модуль LoRa може бути легко з'їзниться з arduino board, ESP8266, ESP32 або STM32 Мікроконтролерами. Споживана потужність модуля дуже низька близько 43 мА при передачі при передачі 16,5 мА при прийомі. Перевівши модуль в сплячий режим, ще більше знижує потужність до 0,5. Для початку зробимо простий Arduino LoRa RYLR890 Передавач/приймач ланцюга та точку зв'язку, як контроль світлодіодної яскравості Бездротового за допомогою потенціометра. Далі відправимо дані датчика

бездротовим способом від передавача до приймача. Датчик, буде використано, – це датчик BME280, який може вимірювати температуру навколишнього середовища, вологість, тиск та висоту.

Нижче наведено список компонентів, необхідних для створення цього простого проекту на основі LoRa (табл. 3.1). Всі компоненти можна легко зібрати на дошці.

Таблиця 3.1 – список компонентів необхідних для реалізації пристрою

С.Н.	Ім'я	Опис	Кількість
1	Плата Ардуїно	Ардуїно Нано	2
2	Модуль LoRa	Reyah RYLR890/RYLR896	2
3	Потенціометри	10К	1
4	Світлодіоди	5мм led будь-якого кольору	1
5	BME280 Датчик	Барометричний датчик тиску BME280	1
6	Резистор	4,7К	1
7	Резистор	10К	1
8	З'єднувальні дроти	Перемички дроти	10
9	Дошка		1

Модуль LoRa RYLR890/RYLR896 від Reyah Technologies базується на мікроконтролері Semtech LoRa SX1276 та мікроконтролері STM32L151C8T6. Чіпи SX1276 працюють на частоті 868/915 МГц і сумісні з STM32L151C8T6 через SPI-штифти.

Модуль Reyah RYLR890/RYLR896 LoRa можна легко взаємодіяти з Arduino за допомогою пінів UART. Існує список команд AT, які можуть бути

використані для виконання будь-якого завдання, такого як надсилання даних або отримання даних або переведення пристрою в режим сну. Ви можете дізнатися більше про команди АТ з цієї технічної документації командування АТ: RYLR890/RYLR896. Особливості та технічні характеристики чіпу:

- Чіп Semtech SX1276.
- Відмінне блокування імунітету.
- Робоча напруга: 2,8 В – 3,3 В.
- Низьке споживання струму.
- Дальність зв'язку: 4,5 км Типовий & 15 км Максимум.
- Висока чутливість радіочастот: -148dBm.
- Легко керуйте командами АТ.
- 127 дБ динамічний діапазон RSSI.
- Невелика вбудована антена друкованої плати.
- Шифрування даних AES128.

3.2.2. Arduino IDE

Інтегроване середовище розробки Arduino – це багатоплатформна програма Java, що включає редактор коду, компілятор та модуль передачі мікропрограми на плату [11].

Середовище розробки засноване на мові програмування Processing і призначене для програмування початківцями, незнайомими з розробкою програмного забезпечення. Мова програмування схожа на мову провідки. Власне кажучи, це C++, доповнений деякими бібліотеками. Програми обробляються за допомогою препроцесора, а потім компілюються за допомогою AVR-GCC. Документація, прошивка та креслення Arduino розповсюджуються за ліцензією Creative Commons Attribution ShareAlike 2.5 і доступні на офіційному веб-сайті Arduino [12].

Вихідний код інтегрованого середовища розробки та бібліотек опублікований та доступний за ліцензією GPLv2. Плати Arduino та Arduino-сумісні спроектовані таким чином, щоб їх можна було розширити при

необхідності, додаючи нові компоненти в схему пристрою. Ці карти розширення підключені до Arduino за допомогою штифтових роз'ємів.

Інтегроване середовище розробки, ISR, — це програмна система, що використовується програмістами для розробки програмного забезпечення.

Зазвичай середовище розробки включає:

- текстовий редактор;
- упорядник або перекладач;
- засоби автоматизації збірки;
- налагоджувач.

Іноді він також включає інструменти для інтеграції з системами контролю версій та різноманітні інструменти для спрощення дизайну графічного інтерфейсу користувача. Багато сучасних середовищ розробки також включають браузер класів, інспектор об'єктів та схему ієрархії класів. Хоча існують постачальники послуг Інтернету, призначені для декількох мов програмування – наприклад, Eclipse, NetBeans, Embarcadero RAD Studio, Qt Creator або Microsoft Visual Studio, але зазвичай ISP призначений для однієї конкретної мови програмування - наприклад, Visual Basic, PureBasic, Delphi, Dev C ++.

Окремий випадок ISR, їх еволюційний розвиток – середовища візуального розвитку, які включають можливість візуального редагування інтерфейсу програми. Інтегровані середовища розробки були створені для максимізації продуктивності програміста завдяки тісно пов'язаним компонентам із простими користувальницькими інтерфейсами. Це дозволить розробнику робити менше дій для перемикання між різними режимами, на відміну від дискретних програм розробки.

Однак, оскільки IDE – це складний програмний пакет, лише після тривалого процесу навчання середовище розробки зможе якісно прискорити процес розробки програмного забезпечення. Як правило, IDE орієнтована на певну мову програмування, забезпечуючи набір функцій, що найбільш

відповідає парадигмам цієї мови програмування. Однак є деякі IDE, які підтримують кілька мов, такі як Eclipse, ActiveState Komodo, останні версії NetBeans, Microsoft Visual Studio, WinDev та Xcode.

IDE, як правило, єдина програма, в якій здійснювався весь розвиток. Зазвичай він містить безліч функцій для створення, модифікації, компіляції, розгортання та налагодження програмного забезпечення. Мета середовища розробки - абстрагувати конфігурацію, необхідну для поєднання утиліт командного рядка в єдиний модуль, що зменшить час вивчення мови та підвищить продуктивність розробника.

3.3 Процес розробки

3.3.1 Поєднання RYLR890/RYLR896 Модуль LoRa з Arduino

Тепер давайте дізнаємося, як ми можемо взаємодіяти з модулем RYLR890/RYLR896 LoRa з платою Arduino. У даному випадку зробимо передавач і приймач ланцюга і контролювати яскравість світлодіода за допомогою потенціометра бездротового зв'язку. Ланцюг передавача має потенціометр, підключений до A0 Pin Of Arduino для генерації змінної аналогової напруги при обертанні потенціометра.

Модуль LoRa працює на 3.3V, тому нам потрібна мережа розділювачів напруги для пінів UART, оскільки цифрові контакти Arduino можуть генерувати 5V. Таким чином, резистор 4.7K & 10K використовується для мережі роздільників напруги, яка перетворює логічну напругу 5V майже в 3,4 В. Схема збірки показана на рис. 3.1.

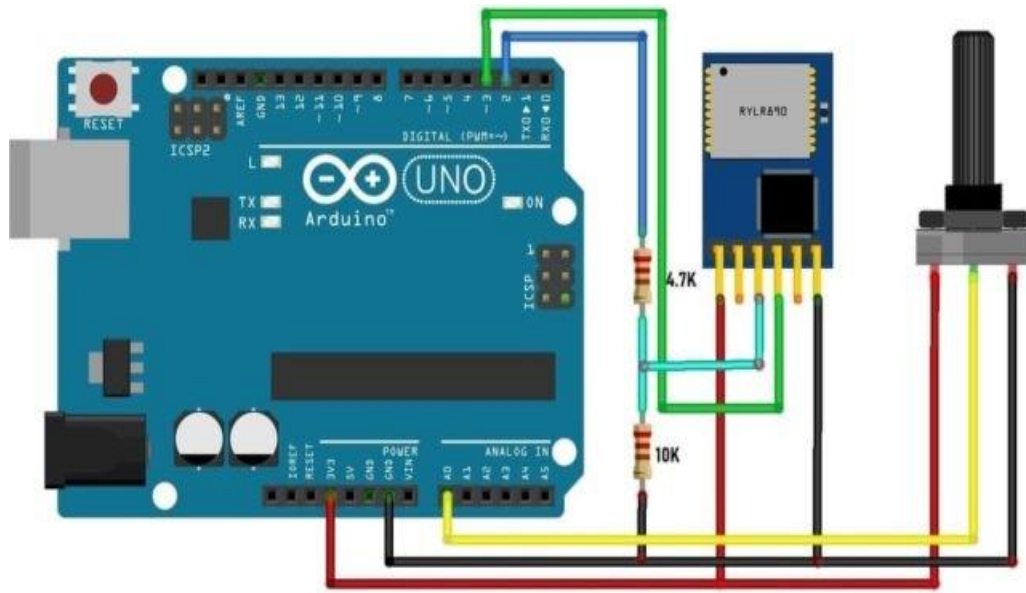


Рисунок 3.1 – Схема передавача

Схема приймача має світлодіод, підключений до D5 Pin Arduino, яскравість якого може регулюватися під час надсилання сигналу від передавача. Схема збірки приймача показана на зображенні 3.2

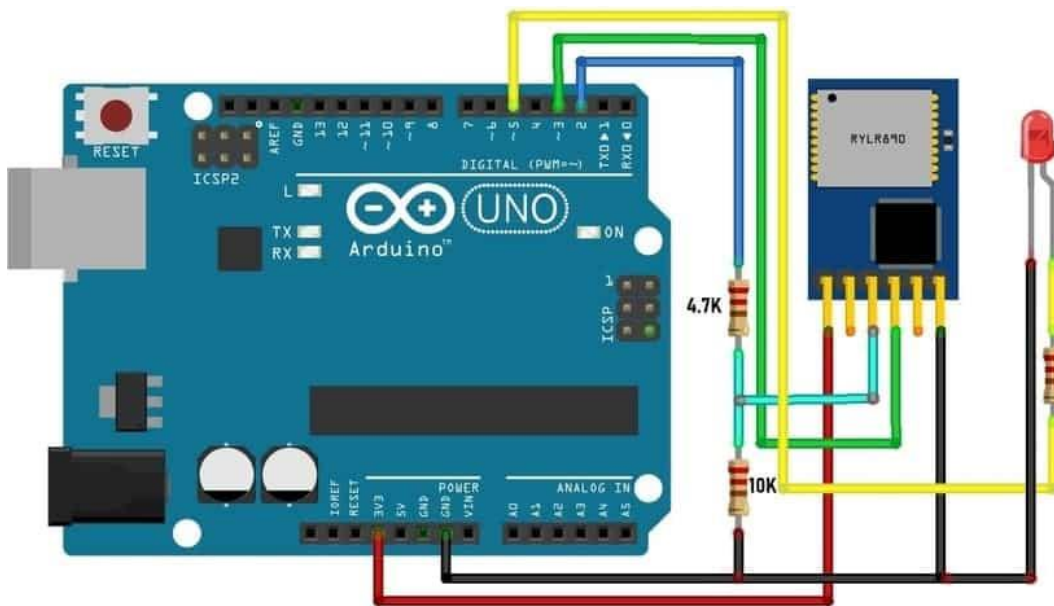


Рисунок 3.2 – Схема приймача

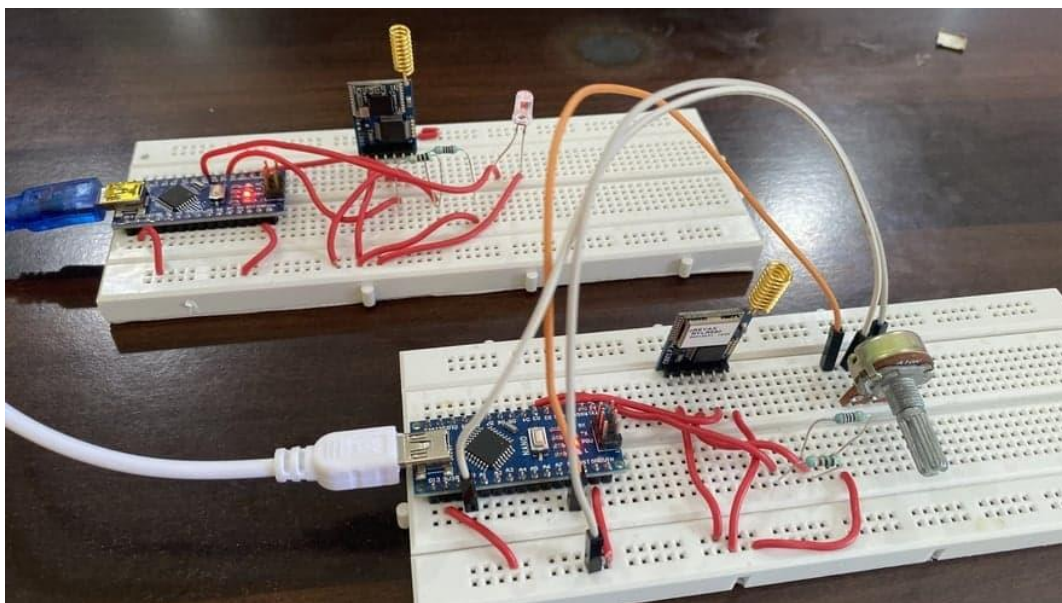


Рисунок 3.3 – Наочний приклад поєднання елементів

Після того, як код буде завантажений на обох мікроконтролерних дошках, можна приступати до тестування всієї схеми. З боку передавача потрібно повернути потенціометр, щоб світлодіодна яскравість приймача збільшилася або зменшилася.

Надішліть сенсорні дані бездротовим способом з LoRa RYLR890/RYLR896 & У цьому 2-му прикладі ми надішлемо дані барометричного датчика тиску BME280 бездротовим способом до приймача LoRa. Барометричний датчик тиску BME280 вимірює температуру, тиск, вологість і висоту.

Ланцюг передавача має датчик BME280, підключений до I2C-пінів Arduino. Модуль LoRa працює на 3.3V, тому нам потрібна мережа розділювачів напруги для пінів UART, оскільки цифрові контакти Arduino можуть генерувати 5V. Таким чином, резистор 4.7K & 10K використовується для мережі роздільників напруги, яка перетворює логічну напругу 5V майже в 3,4 В. Зберіть схему, як показано на зображенні.

Схема приймача не має нічого, крім модуля LoRa, підключеного до штифта UART Arduino

Код / програма для передавача та схеми приймача наведено нижче. Після складання всієї схеми відповідно до схеми можна завантажити код як на передавач, так і на схему приймача.

Код передавача:

```
#include <SoftwareSerial.h>
SoftwareSerial lora (2,3);
int pot = A0;
void setup ()
{
  Serial.begin (115200);
  lora.begin (115200);
  pinMode (pot, INPUT);
}
void loop ()
{
  int val = map (analogRead (pot), 0,1024,0,255);
  Serial.println (val);
  String potval = String (wave);
  String cmd = "AT + SEND = 0," + String (potval.length ()) + "," + String (wave) + "\r";
  //Serial.println ("AT+SEND=0,3,val ");
  lora.println (cmd);
  while (lora.available ()) {
    Serial.write (lora.read ());
  }
  Serial.println ();
  Serial.println (cmd);
  delay (50);
}
```

Код приймача:

```
#include <SoftwareSerial.h>
SoftwareSerial lora(2,3);
int LED = 5;
//String inString = "";
```

```
int val = 0;
void setup()
{
  Serial.begin(115200);
  lora.begin(115200);
  pinMode(LED, OUTPUT);
}
void loop()
{
  //char ch;
  String inString;
  while (lora.available())
  {
    if(lora.available()){
      inString += String(char(lora.read()));
    }
  }
  if(inString.length()>0)
  {
    //Serial.println(inString);
    String potval;
    potval= inString.substring(9,12);
    Serial.println(potval);
    analogWrite(LED,potval.toInt());
  }
  //delay(100);
}
```

Після завантаження коду можна відкрити послідовний монітор як для передавача, так і для частин приймача. Ланцюг відправника/передавача зчитає дані датчика BME280 і надішле їх бездротовим способом одержувачу. Приймач отримає дані і покаже їх на послідовному моніторі разом з сигналом RSSI.

3.3.2 Вихідний код програми

Зібравши всю схему відповідно до принципової схеми, ви можете завантажити код до ланцюгів як передавача, так і приймача.

```
#include <SoftwareSerial.h>
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
```

Підключаємо необхідні бібліотеки

```
SoftwareSerial lora(2,3);
#define SEALEVELPRESSURE_HPA (1013.25)
Adafruit_BME280 bme;
void setup()
{
  Serial.begin(115200);
  lora.begin(115200);
  if (!bme.begin(0x76))
  {
    Serial.println("Could not find a valid BME280 sensor, check wiring!");
    while (1);
  }
}
```

Дана частина коду перевіряє наявність сигналу від датчиків, і видає відповідне повідомлення в разі його відсутності.

```
}
void loop()
{
  float temperature = bme.readTemperature();
  float pressure = bme.readPressure() / 100.0F;
  float altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
  float humidity = bme.readHumidity();
```

Визначаємо необхідні змінні для параметрів які будемо передавати

```
Serial.print(F("Temperature = "));
Serial.print(temperature);
```

```
Serial.println(F("*C"));
Serial.print(F("Pressure = "));
Serial.print(pressure);
Serial.println(F("hPa"));
Serial.print(F("Approx. Altitude = "));
Serial.print(altitude);
Serial.println(F("m"));
Serial.print(F("Humidity = "));
Serial.print(humidity);
Serial.println(F("%"));
String temp = String(temperature);
String pres = String(pressure);
String alt = String(altitude);
String hum = String(humidity);
String values = String(temp)+" "+String(pres)+" "+String(alt)+" "+String(hum);
String cmd = "AT+SEND=0,"+String(values.length())+" "+values;
lora.println(cmd);
```

Друкуємо отримані дані на екран, і відправляємо їх на отримувач, далі знову переходимо в режим зчитування даних

```
while(lora.available())
{
  Serial.write(lora.read());
}
Serial.println();
Serial.println(cmd);
delay(5000);
}
```

Код отримувача:

```
#include <SoftwareSerial.h>
SoftwareSerial lora(2,3);
void setup()
{
  Serial.begin(115200);
  lora.begin(115200);
```

```

}
void loop()
{
String inString;
while (lora.available())
{
if(lora.available()){
inString += String(char(lora.read()));
}
}
if(inString.length(>0)
{
Serial.println(inString);
inString.remove(0);
}
}

```

Після завантаження коду можемо відкрити послідовний монітор для передавача та деталі ресивера. Схема відправника/передавача зчитає дані датчика BME280 та надішле через бездротовий зв'язок на приймач. Приймач отримає дані та відобразить їх на послідовному моніторі разом із сигналом RSSI. Результати роботи програми представлені на рис. 3.4

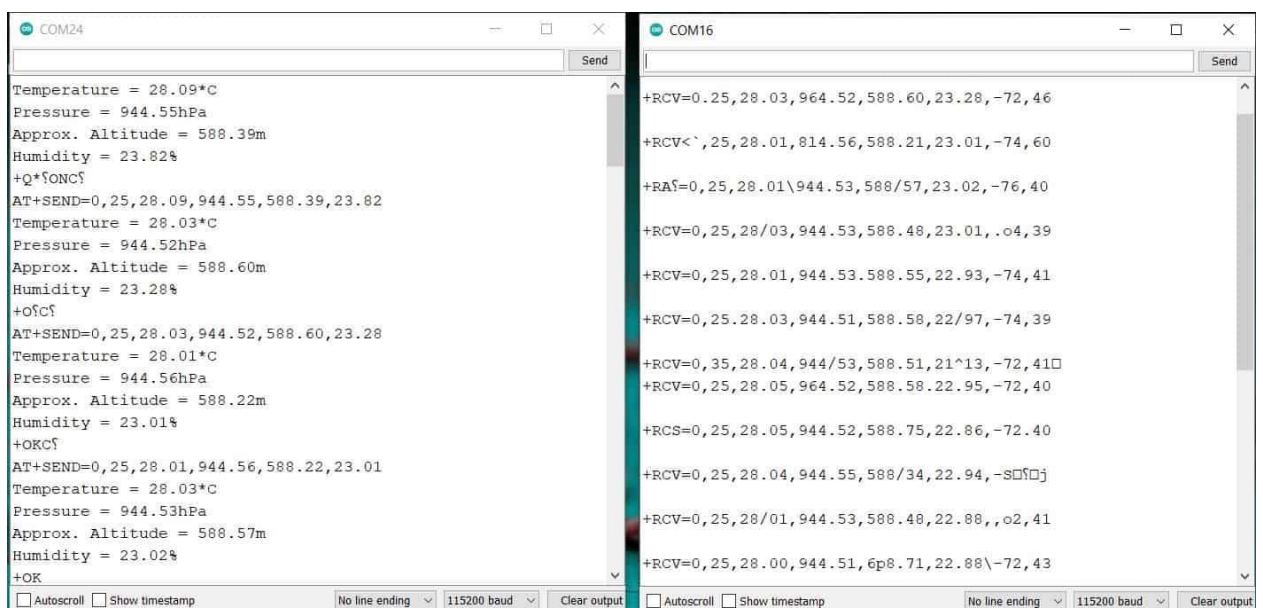


Рисунок 3.4 – Результати роботи програми

3.3.3 Реалізація для умов Smart City

Так як в умовах «розумного міста» ми матимемо безліч даних для аналізу, доцільно було би реалізувати подібний пристрій, але з можливістю приймати сигнал від декількох передавачів одночасно. Розглянемо простий приклад з трьома передавачами.



Рисунок 3.5 – Модулі приймача Lora RYLR896/RYLR890 на 15 км:

Це 15-ти кілометрові бездротові радіочастотні модулі LoRa трансивера 915 МГц від Reuax Technology. Особливістю цих модулів є те, що можна змінювати частоти цих модулів за допомогою AT-команд. Ці модулі призначені для зв'язку на великій відстані до 15 кілометрів .

Цей модуль трансивера має процесор, який зв'язується з Arduino через послідовний зв'язок. Таким чином, обмін даними між Rylr890 або Rylr896 і мікроконтролером контролюється цим процесором.

Як можна побачити на рис. 3.5, цей модуль трансивера має загалом 6 контактів; які чітко позначені як VDD, NRST «що є активним низьким рівнем контакту скидання». RXD, TXD, контакт номер 5 не використовується, тоді як контакт номер 6 є заземленням.

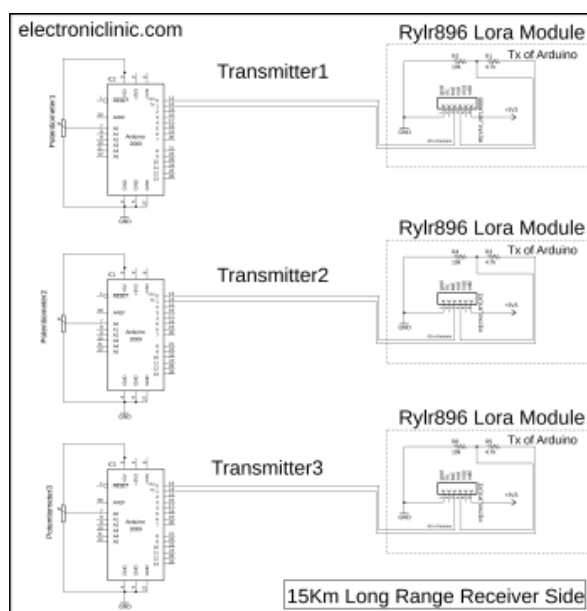


Рисунок 3.6 – Кілька передавачів на основі Lora і Arduino, принципова схема

На схемі (рис 3.6) можна побачити, що три модулі трансивера Lora сполучені з трьома платами Arduino. Модулі Lora Transceiver можуть працювати лише з рівнем напруги 3,3 В. Якщо ці модулі безпосередньо підключені до Arduino, існує велика ймовірність пошкодити модулі Lora.

Для безпечного використання модулів Lora є два варіанти: ви можете використовувати двонаправлений перетворювач рівня напруги або використовувати пару резисторів. В даному випадку використана пара резисторів 10k і 4,7k. Підключення всіх трьох передавачів абсолютно однакові.

З'єднання модуля Lora Transceiver з платою Arduino залишиться незмінним. Триканальний релейний модуль підключається до Arduino за допомогою цифрових контактів 8, 9 і 10. Можна використовувати готовий триканальний релейний модуль 12 В, або можна стежити за підключеннями і створити його самостійно.

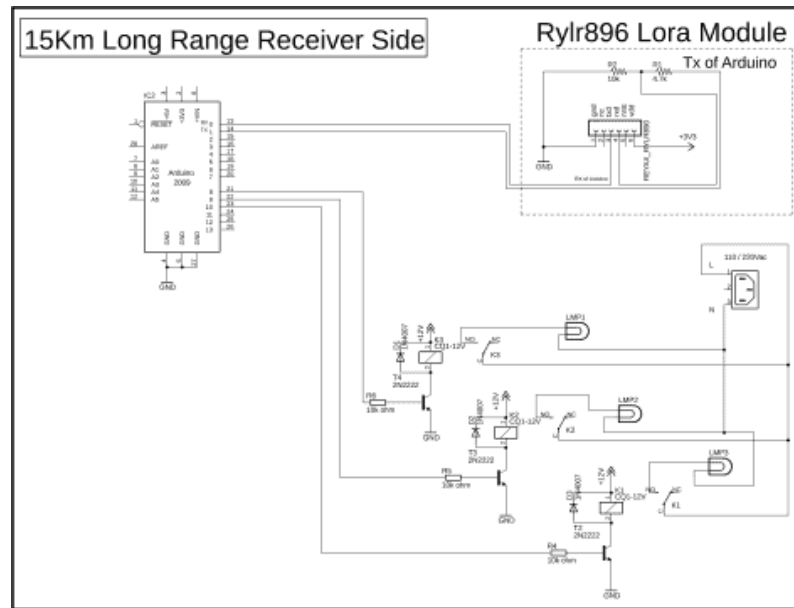


Рисунок 3.7 – Одиничний приймач на основі Lora і Arduino

3.3.4 Програмування кількох передавачів Lora Arduino

Так як в даному випадку використовуються три передавачі, треба буде 3 коди Arduino. В цілому можна використовувати той самий код на всіх трьох платах Arduino, але з різними заголовками. Код передавача наведений у додатку А.

У коді передавача, що наведено нижче, для кожного трансмітера замінюємо «S1» на «S2» та «S3» відповідно.

```
vrdata = analogRead(vresistor);  
sensorsdata = sensorsdata + "S1" + "%" + vrdata ;
```

Використовуючи S1, S2 і S3, приймач може точно знати, від якого передавача він отримує дані. Створюється повідомлення, яке складається із значення заголовка S1 або S2 або S3 і значення Sensor, розділеного за допомогою % як роздільника.

3.3.5 Програмування приймача

Код приймача наведено у додатку Б. В даній частині зупинимося на поясненні окремих частин даного коду.

Визначаємо три змінні Sensor1, Sensor2 і Sensor3. Ці змінні будуть використовуватися для зберігання значень потенціометрів, що надходять від передавачів.

```
int Sensor1;  
int Sensor2;  
int Sensor3;
```

Потім визначимо контакти для 3 реле, з'єднаних з контактами 8, 9 і 10.

```
int Relay1 = 8;  
int Relay2 = 9;  
int Relay3 = 10;  
void setup() {  
  Serial.begin(115200); // активуємо послідовний зв'язок, вибравши 115200 як швидкість  
  передачі даних.  
  Serial.print("AT\r\n");  
  delay(100);
```

Реле є вихідними пристроями

```
pinMode(Relay1, OUTPUT);  
pinMode(Relay2, OUTPUT);  
pinMode(Relay3, OUTPUT);  
void loop() {
```

Використовуємо умову if, щоб перевірити, чи плата Arduino отримала дані від Lora.

```
if ( Serial.available() > 0 ) {  
  garbage = Serial.readString(); // складається з +ERR=2 ERROR.
```

Читаємо рядок

```
myString = Serial.readString();
```

Друкуємо рядок для налагодження, щоб перевірити, чи ми отримали дані.

```
Serial.println(myString);  
//Serial.println("Garbage:");  
//Serial.println(garbage);  
String l = getValue(myString, ',', 0); // address  
String m = getValue(myString, ',', 1); // data length
```

```
String n = getValue(myString, ',', 2); // data
```

```
String o = getValue(myString, ',', 3); //RSSI
```

```
String p = getValue(myString, ',', 4); //SNR
```

Тепер розділимо рядок даних n, який складається із значень Sensors

У даному рядку значення датчиків розділяються знаком %.

```
String q = getValue(n, '%', 0); // Header
```

```
String r = getValue(n, '%', 1); // значення датчика від кількох передавачів
```

```
if ( ( q != "" ) && ( r != "" ) )
```

```
{
```

```
Serial.println("Sensor:");
```

```
Serial.println(q);
```

```
Serial.println("data:");
```

```
Serial.println(r);
```

Звідси починаємо перевірку, чи отримані значення датчика від Передавача1, Передавача2 або від Передавача3. Перевіряємо це за допомогою умов if. якщо q має рядок S1, це означає, що дані отримані від передавача1. Отже, значення, що зберігається в r, перетворюється в ціле число, а потім зберігається в Sensor1. Таким же чином робимо це для інших датчиків.

```
if ( q == "S1" )
```

```
{
```

```
Sensor1 = r.toInt();
```

```
}
```

```
if ( q == "S2" )
```

```
{
```

```
Sensor2 = r.toInt();
```

```
}
```

```
if ( q == "S3" )
```

```
{
```

```
Sensor3 = r.toInt();
```

```
}
```

```
myString = "";
```

```
}
```

```
Serial.println("Sensor1");  
Serial.println(Sensor1);  
Serial.println("Sensor2");  
Serial.println(Sensor2);  
Serial.println("Sensor3");  
Serial.println(Sensor3);  
if ( Sensor1 <= 200 )  
{  
digitalWrite(Relay1, HIGH);  
Serial.println("Relay1 High");  
}  
if ( Sensor1 > 200 )  
{  
digitalWrite(Relay1, LOW);  
Serial.println("Relay1 Low");  
}
```

Далі використовуємо ці умови для керування всіма трьома реле. У подальшому можна розширити кількість передавачів, для аналізу даних, керуючись уже розробленою схемою.

На рис. 3.8 наведено наочний приклад підключення компонентів. Коен передавач підключається схожим чином.

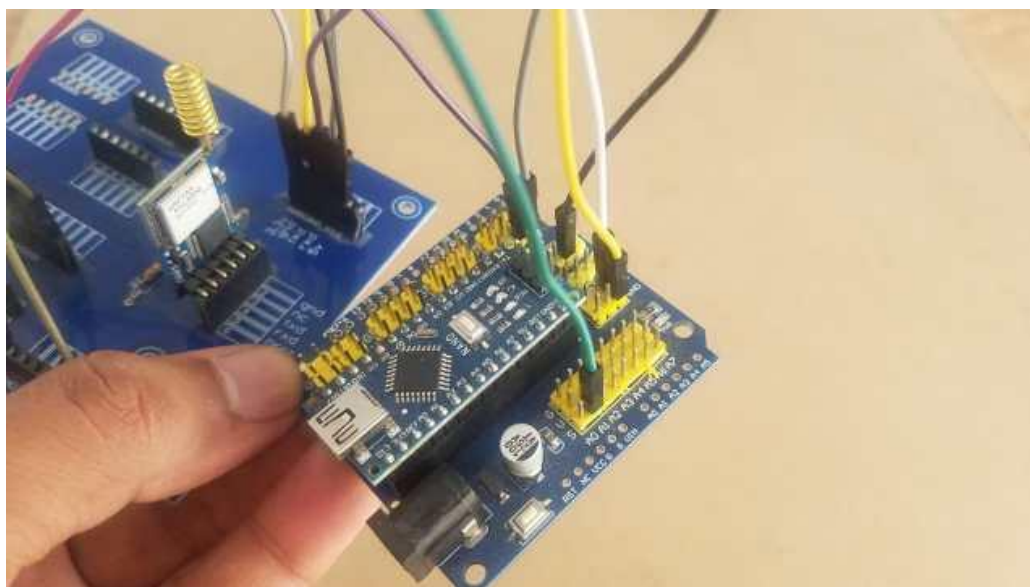


Рисунок 3.8 – Наочний приклад підключення компонентів

Висновки до розділу 3

У цьому розділі було обґрунтовано вибір компонентів, для створення пристрою бездротової передачі даних, описано процес розробки програмної та апаратної частини роботи, та визначено способи тестування програмного забезпечення які потрібно провести для впевненості у працездатності системи.

Описано важливі параметри обраного мікроконтроллера та програмного забезпечення, наведено принцип роботи даного пристрою та його принципову схему. Описано структуру коду, що наведено у додатках А та Б.

В результаті роботи над розділом 3, досягнуто наступних цілей дипломної роботи:

- обрано компоненти та програмне забезпечення для роботи;
- розроблено модель пристрою відправки та отримання даних
- написано програмне забезпечення для створеного пристрою.

Було реалізовано 2 варіанти вирішення питання: огляд простого пристрою передачі даних з використанням вибраних компонентів, а також розроблено модель, яка може бути використана в умовах «розумного міста», де передавачів буде набагато більше ніж приймачів даних. Таким чином можна зробити висновок що в даному розділі було досягнуто основних цілей роботи.

ВИСНОВКИ

При виконанні даної роботи була проаналізована можливість розробки пристрою для збору інформації для постачальників енергоресурсів у системах «розумного міста».

В результаті виконання дипломної роботи змодельовано пристрій який дозволяв би передавати дані на великі відстані, а також написано програмну частину яка дозволяє виводити отримані дані на екран комп'ютера. Також виявлено, що така система може бути дуже доцільною при використанні як в домашніх умовах так і на підприємствах, оскільки собівартість такого пристрою є нижчою ніж вартість існуючих на ринку представників даної технології, що пов'язано з використанням недорогих компонентів, які не дивлячись на дешевизну, добре виконують поставлену задачу. За результатами виконання роботи та аналізу літератури було вирішено що модернізація та впровадження таких систем є дуже важливим аспектом у створенні безпечних та комфортних умов існування людини.

З економічної точки зору це може бути ефективно якщо використовувати дешеві аналоги існуючих мікроконтролерів, замість вже існуючих дорогих компонентів, що представляються розповсюдженими на українському ринку компаніями.

Питання безпеки і охорони праці, а також здоров'я працівників теж є дуже важливими і актуальними для будь-якого підприємства в Україні. У розділі з охорони праці були виконані перевірочні розрахунки як природного так і штучного освітлення. В результаті розрахунку природного та штучного освітлення у приміщенні було з'ясовано, що для покращення освітленості необхідно створити систему штучного освітлення, яка буде складатись із чотирьох світильників, кожен з яких міститиме 4 лампи, що мають світловий потік у 1800 лм.

Безумовно, покращення умов праці персоналу потребують певних матеріальних витрат, але в розвинутих країнах доведено, що вони повністю

компенсуються за рахунок підвищення продуктивності праці та зменшення страхових виплат працівникам внаслідок порушень функціонального стану організму.

Одним із розглянутих питань – забезпечення безпеки персоналу в умовах надзвичайної ситуації пов'язаної з пожежею. Одним з основних засобів захисту є своєчасний і швидкий вивіз або вивід людей з небезпечної зони, тобто евакуація.

В результаті виконання магістерської роботи можна зробити висновок, що всі поставлені цілі та завдання було досягнуто, зокрема:

- проведено аналітичний огляд літератури та патентної інформації;
- порівняно вже існуючі аналоги для знаходження оптимального рішення;
- обрано компоненти пристрою для зчитування біометричних документів;
- розроблено альтернативні шляхи моделювання пристрою;
- розроблено модель пристрою зчитування даних;
- розроблено програмну частину;
- змодельовано пристрій з використанням обраних компонентів;
- розроблено питання безпеки життєдіяльності та охорони праці.

Розглянута технологія також має великі перспективи: збільшивши кількість передавачів, можна буде обробляти більшу кількість інформації, окрім того, систему можна модернізувати, додавши більше датчиків, і відповідно покращивши код, можна буде за той же час передавати більшу кількість даних. Це дозволить енергопостачальникам більш якісно контролювати свої системи, а також швидше аналізувати стан систем на предмет аварійних ситуацій і вчасно приймати попереджувальні заходи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- [1] В. У. О. Фінансування, “ПРИКЛАДНІ АСПЕКТИ ЗАХИСТУ ІНФОРМАЦІЇ В УМОВАХ ОБМЕЖЕНОГО ФІНАНСУВАННЯ,” *Кібербезпека*, vol. 1, no. 1, pp. 26–34, 2018.
- [2] ICAO, *Doc 9303 - Machine Readable Travel Documents - Part 01: Introduction*. 2015.
- [3] T. Matsumoto and H. Matsumoto, “Impact of artificial gummy fingers on fingerprint systems,” *Proc. ...*, vol. 4677, no. May, pp. 275–289, 2002, [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Tutomu+Matsumoto#2%5Cnhttp://cryptome.info/0001/gummy/gummy.htm>.
- [4] M. Drahanisky and D. Hejtmankova, “New experiments with optical liveness testing methods,” *J. Inf. Hiding Multimed. Signal Process.*, vol. 1, no. 4, pp. 301–309, 2010.
- [5] V. Tsurkan, “Method of Information Security Management Systems Functional Analysis,” *Cybersecurity Educ. Sci. Tech.*, vol. 4, no. 8, pp. 192–201, 2020, doi: 10.28925/2663-4023.2020.8.192201.
- [6] Г. В. Мигаль and В. П. Мигаль, “Когнітивні Та Ергономічні Аспекти Взаємодії Людини З Комп'Ютером,” *Radioelectron. Comput. Syst.*, vol. 1, no. 1, pp. 90–102, 2020, doi: 10.32620/reks.2020.1.09.
- [7] A. V. Sachenko, R. V. Konakova, and A. E. Belyaev, “Physical mechanisms providing formation of ohmic contacts metal– semiconductor,” *Semicond. Phys. Quantum Electron. Optoelectron.*, vol. 21, no. 1, pp. 5–40, 2018, doi: 10.15407/spqeo21.01.005.
- [8] B. Etlzinger and H. Wymeersch, “Synchronization and localization in wireless networks,” *Found. Trends Signal Process.*, vol. 12, no. 1, pp. 1–106, 2018, doi: 10.1561/20000000096.
- [9] ARduino, “Arduino Uno R3,” *Pololu Corporation*, 2018. <https://sites.google.com/site/mikhorkhxnthorllexr1/bxrd-mikhor->

-
- khxnthorllexr-arduino-uno-r3.
- [10] A. Tupasela, K. Snell, and H. Tarkkala, “The Nordic data imaginary,” *Big Data Soc.*, vol. 7, no. 1, 2020, doi: 10.1177/2053951720907107.
- [11] D. Malík and M. Dražanský, “Anatomy of biometric passports,” *J. Biomed. Biotechnol.*, vol. 2012, 2012, doi: 10.1155/2012/490362.
- [12] J. FINNESSY, John, “EP 2663106 B1,” 2017.
- [13] T. T. Mark Buer, “States Patent [US 10,552,604 B2],” 2018.
- [14] Julien Marie, “US 9813116 B2,” 2017.
- [15] A. Ali, A. H. S. Saad, and A. H. Ismael, “Data Hiding Technique Based on NFC-Enabled Smartphones,” *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 2400–2409, 2020, doi: 10.1016/j.procs.2020.04.260.
- [16] “News in brief,” *Biometric Technology Today*, vol. 2020, no. 8, p. 4, 2020.
- [17] K. Iqbal, M. O. Odetayo, and A. James, “Content-based image retrieval approach for biometric security using colour, texture and shape features controlled by fuzzy heuristics,” *J. Comput. Syst. Sci.*, vol. 78, no. 4, pp. 1258–1277, 2012, doi: 10.1016/j.jcss.2011.10.013.
- [18] P. Li and R. Zhang, “The evolution of biometrics,” *Proceedings - 2010 International Conference on Anti-Counterfeiting, Security and Identification, 2010 ASID*, 2010. .
- [19] H. J. La, “A conceptual framework for trajectory-based medical analytics with IoT contexts,” *J. Comput. Syst. Sci.*, vol. 82, no. 4, pp. 610–626, 2016, doi: 10.1016/j.jcss.2015.10.007.
- [20] “土の2次元切削理論 (第3報),” *JOURNAL of the JAPANESE SOCIETY of AGRICULTURAL MACHINERY*, vol. 38, no. Supplement. pp. 40–40, 1976, doi: 10.11357/jsam1937.38.Supplement_40.
- [21] S. Richardson, “Быстрый Старт,” 2006.
- [22] Z. Müftüoğlu and T. Yildirim, “Comparative Analysis of Crypto Systems Using Biometric Key,” *Procedia Comput. Sci.*, vol. 154, pp. 327–331, 2018, doi: 10.1016/j.procs.2019.06.047.

-
- [23] J. G. Herrero, “VULNERABILITIES AND ATTACK PROTECTION IN SECURITY SYSTEMS,” *Doctor*, no. November 2009, 2009.
- [24] Петин В. А., *Arduino u Raspberry Pi в проектах Intemet ofThings*. 2016.
- [25] D. Schiller, “EP12856426B2,” 2018.
- [26] D. Kumar and Y. Ryu, *A brief introduction of biometrics and fingerprint payment technology*, vol. 22. 2017.
- [27] М. Геддес, *25 крутых проектов с Arduino*. 2019.
- [28] Elechouse, *PN532 NFC RFID Module User Guide*. 2015.
- [29] P. S. Royston, “* EP002541791B1 *,” 2020.
- [30] І. А. Пількевич and І. В. Гуменюк, “Метод децентралізованого управління мережевими ресурсами інформаційно - комунікаційних мереж,” *Технічна інженерія*, vol. 2, no. 84, pp. 100–108, 2019.

ДОДАТОК А КОД ARDUINO ПЕРЕДАВАЧА

```
int vresistor = A0;
int vrdata = 0;
int data_length;
String sensorsdata;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(vresistor,INPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
  MultipleSensors();
  send_data(sensorsdata , data_length);
  delay(100);
  sensorsdata = "";
}
void MultipleSensors()
{
  vrdata = analogRead(vresistor);
  sensorsdata = sensorsdata + "S1" + "%" + vrdata ;
  //Serial.println("sensor value:");
  //Serial.println(vrdata);
  // find the length of data
  data_length = sensorsdata.length();
  //Serial.println("data length:");
  //Serial.println(data_length);
}
void send_data(String sensorvalue, int valuelength)
{
  String mymessage;
  mymessage = mymessage + "AT+SEND=0" + "," + valuelength + "," + sensorvalue +
  "\r";
```

```
Serial.println(mymessage);
  //Serial.println("AT+SEND=0,6,Hello!\r");
int vresistor = A0;
int vrdata = 0;
int data_length;
String sensorsdata;
void setup() {
  Serial.begin(115200);
  pinMode(vresistor,INPUT);
}
void loop() {
  // put your main code here, to run repeatedly:
  MultipleSensors();
  send_data(sensorsdata , data_length);
  delay(100);
  sensorsdata = "";
}
void MultipleSensors()
{
  vrdata = analogRead(vresistor);
  sensorsdata = sensorsdata + "S1" + "%" + vrdata ;
  //Serial.println("sensor value:");
  //Serial.println(vrdata);
  // find the length of data
  data_length = sensorsdata.length();
  //Serial.println("data length:");
  //Serial.println(data_length);
}
void send_data(String sensorvalue, int valuelength)
{
  String mymessage;
  mymessage = mymessage + "AT+SEND=0" + "," + valuelength + "," + sensorvalue +
"\r";
  Serial.println(mymessage);
```

```
//Serial.println("AT+SEND=0,6,Hello!\r");  
}
```

ДОДАТОК Б КОД ПРИЙМАЧА

```
String myString;
String garbage;
char data;
int Sensor1;
int Sensor2;
int Sensor3;
int Relay1 = 8;
int Relay2 = 9;
int Relay3 = 10;
void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    Serial.print("AT\r\n");
    delay(100);

    pinMode(Relay1, OUTPUT);
    pinMode(Relay2, OUTPUT);
    pinMode(Relay3, OUTPUT);

    // by default keep all relays low
    digitalWrite(Relay1, LOW);
    digitalWrite(Relay2, LOW);
    digitalWrite(Relay3, LOW);
}
void loop() {
    if ( Serial.available() > 0 )
    {
        garbage = Serial.readString(); // consists of the +ERR=2 ERROR.
        myString = Serial.readString();
        Serial.println(myString);
        //Serial.println("Garbage:");
    }
}
```

```
//Serial.println(garbage);
String l = getValue(myString, ',', 0); // address
String m = getValue(myString, ',', 1); // data length
String n = getValue(myString, ',', 2); // data
String o = getValue(myString, ',', 3); //RSSI
String p = getValue(myString, ',', 4); //SNR
// тепер розділяємо рядок даних n, який складається із значень Sensors
// у цьому рядку значення датчиків розділяються знаком %.
String q = getValue(n, '%', 0); // Header
String r = getValue(n, '%', 1); // значення датчика від кількох передавачів
if ( ( q != "" ) && ( r != "" ) )
{
  Serial.println("Sensor:");
  Serial.println(q);
  Serial.println("data:");
  Serial.println(r);
  if ( q == "S1" )
  {
    Sensor1 = r.toInt();
  }
  if ( q == "S2" )
  {
    Sensor2 = r.toInt();
  }
  if ( q == "S3" )
  {
    Sensor3 = r.toInt();
  }
  myString = "";
}
Serial.println("Sensor1");
Serial.println(Sensor1);
Serial.println("Sensor2");
Serial.println(Sensor2);
```



```
Serial.println("Sensor3");
Serial.println(Sensor3);
if ( Sensor1 <= 200 )
{
    digitalWrite(Relay1, HIGH);
    Serial.println("Relay1 High");
}
if ( Sensor1 > 200 )
{
    digitalWrite(Relay1, LOW);
    Serial.println("Relay1 Low");
}
if ( Sensor2 <= 200 )
{
    digitalWrite(Relay2, HIGH);
}
if ( Sensor2 > 200 )
{
    digitalWrite(Relay2, LOW);
}

if ( Sensor3 <= 200 )
{
    digitalWrite(Relay3, HIGH);
}
if ( Sensor3 > 200 )
{
    digitalWrite(Relay3, LOW);
}
}
}

String getValue(String data, char separator, int index)
{
    int found = 0;
```

```
int strIndex[] = { 0, -1 };
int maxIndex = data.length() - 1;

for (int i = 0; i <= maxIndex && found <= index; i++) {
    if (data.charAt(i) == separator || i == maxIndex) {
        found++;
        strIndex[0] = strIndex[1] + 1;
        strIndex[1] = (i == maxIndex) ? i+1 : i;
    }
}
return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

String myString;
String garbage;
char data;
int Sensor1;
int Sensor2;
int Sensor3;
int Relay1 = 8;
int Relay2 = 9;
int Relay3 = 10;
void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    Serial.print("AT\r\n");
    delay(100);
    pinMode(Relay1, OUTPUT);
    pinMode(Relay2, OUTPUT);
    pinMode(Relay3, OUTPUT);
    // by default keep all relays low
    digitalWrite(Relay1, LOW);
    digitalWrite(Relay2, LOW);
    digitalWrite(Relay3, LOW);
}
```

```
void loop() {
    // put your main code here, to run repeatedly:
    if ( Serial.available() > 0 )
    {
        garbage = Serial.readString(); // consists of the +ERR=2 ERROR.
        myString = Serial.readString();
        Serial.println(myString);
        //Serial.println("Garbage:");
        //Serial.println(garbage);
        String l = getValue(myString, ',', 0); // address
        String m = getValue(myString, ',', 1); // data length
        String n = getValue(myString, ',', 2); // data
        String o = getValue(myString, ',', 3); //RSSI
        String p = getValue(myString, ',', 4); //SNR
        // тут розділяється рядок даних n, який складається із значень Sensors
        // у цьому рядку значення розділяються знаком %.String q = getValue(n, '%', 0);
        // Header
        String r = getValue(n, '%', 1); // sensor values from multiple transmitters
        if ( ( q != "" ) && ( r != "" ) )
        {
            Serial.println("Sensor:");
            Serial.println(q);
            Serial.println("data:");
            Serial.println(r);
            if ( q == "S1" )
            {
                Sensor1 = r.toInt();
            }

            if ( q == "S2" )
            {
                Sensor2 = r.toInt();
            }
        }
    }
}
```

```
if ( q == "S3")
{
  Sensor3 = r.toInt();
}
myString = "";
}
Serial.println("Sensor1");
Serial.println(Sensor1);
Serial.println("Sensor2");
Serial.println(Sensor2);
Serial.println("Sensor3");
Serial.println(Sensor3);
if ( Sensor1 <= 200 )
{
  digitalWrite(Relay1, HIGH);
  Serial.println("Relay1 High");
}
if ( Sensor1 > 200 )
{
  digitalWrite(Relay1, LOW);
  Serial.println("Relay1 Low");
}
if ( Sensor2 <= 200 )
{
  digitalWrite(Relay2, HIGH);
}
if ( Sensor2 > 200 )
{
  digitalWrite(Relay2, LOW);
}
if ( Sensor3 <= 200 )
{
  digitalWrite(Relay3, HIGH);
}
```

```
if ( Sensor3 > 200 )
{
    digitalWrite(Relay3, LOW);
}
}
}

String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;
    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
```