

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

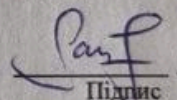
Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

на тему «Система збору даних сенсорів на основі багаторівневої системи залишкових класів»

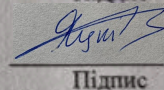
КвРКІ.180149.20.01.34 ПЗ

Виконав: студент 2 курсу, група КІ2м-20-1


Підпис

Федоришин О.М.
Ініціали, прізвище

Керівник доктор техн. наук, професор
Науковий ступінь, вчене звання


Підпис

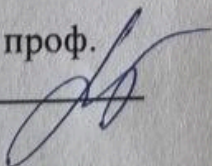
Яцків В.В.
Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КІС, д.т.н., проф.

Т.О. Говорущенко

19 05 2022 р.



ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Г.О.Говорущенко

“ 01 ” 09 2021 р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Федоришину Олександрю Михайловичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система збору даних сенсорів на основі багаторівневої системи залишкових класів

Керівник проекту (роботи) Яцків В.В., д.т.н., професор

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 06.01.2022 р. № 1

2. Строк подання студентом проекту (роботи) на кафедру 03.05.2022 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз області використання, характеристик та топологій сучасних БСМ

Моделі та методи кодування та передачі даних сенсорів на основі системи залишкових класів

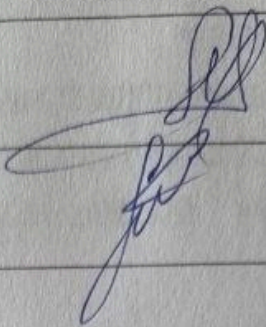
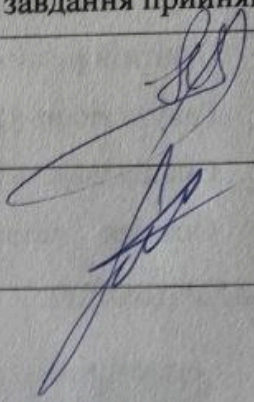
Запропонований метод кодування та передачі даних сенсорів

Програмна реалізація БСМ та системи збору та обробки даних із застосуванням розробленого методу кодування

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Презентаційні матеріали (слайди)

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

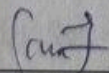
7. Дата видачі завдання « 06 » 09 2021р.


КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики ДРМ з керівником	05.09.2021	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	05.10.2021	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	05.11.2021	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	05.12.2021	виконано
5	Робота над науковою статтею	05.01.2022	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2022	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	05.04.2022	виконано
8	Оформлення пояснювальної записки згідно вимог	15.04.2022	виконано
9	Попередній захист ДРМ	18.04.2022	виконано
10	Захист ДРМ на засіданні ЕК	До 10.05.2022	

Студент

Керівник проекту (роботи)


Підпис


Підпис

О.М. Федоришин
Ініціали, прізвище

В.В. Яцків
Ініціали, прізвище

РЕФЕРАТ

Тема дипломної роботи: Система збору даних сенсорів на основі багаторівневої системи залишкових класів.

Автор роботи: Федоришин Олександр Михайлович.

Керівник роботи: Яцків Василь Васильович.

Пояснювальна записка: 79 с., 36 рис., 3 табл., 3 дод., 50 джерел.

БСМ, сенсор, система залишкових класів, кодування, декодування

Об'єктом дослідження є процеси кодування у системах збору даних сенсорів.

Предметом дослідження є методи та засоби кодування та передачі даних у сенсорних мережах на основі системи залишкових класів.

Метою роботи розробка способу кодування та передачі даних сенсорів, який дозволить зменшити обсяг службової інформації в протоколах передачі даних безпроводних сенсорних мереж.

Для розв'язання поставлених задач використовувалися методи кодування даних за допомогою багаторівневої системи залишкових класів.

Наукова новизна отриманих результатів:

– отримав подальший розвиток метод кодування даних сенсорів, який забезпечує ефективне кодування даних сенсорів різної розрядності за рахунок використання відповідних модулів системи залишкових класів;

– удосконалено метод передачі даних у бездротових сенсорних мереж, що дозволило досягти зменшення обсягу службових даних у протоколах передачі даних.

На основі проведених досліджень розроблений метод кодування та передачі даних сенсорів що дозволяє закодувати показники сенсорів у єдине значення. Що дозволяє скоротити обсяг службових даних; розроблена архітектура і компоненти віртуальної бездротової сенсорної мережі а також системи збору та обробки даних для застосування розробленого методу кодування.

Практична значимість отриманих результатів полягає у підвищенні ефективності використання пропускну здатності безпроводних сенсорних мереж.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	6
ВСТУП.....	7
1 ДОСЛІДЖЕННЯ СЕНСОРНИХ СИСТЕМ ТА МЕТОДІВ ЗБОРУ ДАНИХ	3
СЕНСОРІВ	9
1.1 Визначення БСМ.....	9
1.2 Области застосування БСМ.....	9
1.3 Класифікація БСМ.....	13
1.4 Характеристики БСМ	14
1.5 Топології мереж	15
1.6 Сенсорний вузол	18
1.7 Висновки.....	23
2 НЕДВІЙКОВІ СИСТЕМИ ЧИСЛЕНЬ ТА КОДУВАННЯ В СИСТЕМІ	24
ЗАЛИШКОВИХ КЛАСІВ	24
2.1 Теорія недвійкових систем числення	24
2.2 Система залишкових класів.....	25
2.2.1 Вибір модулів.....	27
2.2.2 Робота з від’ємними числами.....	28
2.2.3 Базові математичні операції	30
2.2.4 Перетворення	31
2.3 Области застосування системи залишкових класів	32
2.4 Обробка цифрових сигналів	34
2.4.1 Цифрові фільтри	35
2.4.2 FIR фільтри.....	35

2.5	Висновки.....	38
3	АЛГОРИТМИ ТА ТЕХНОЛОГІЇ КОДУВАННЯ В СИСТЕМІ ЗАЛИШКОВИХ КЛАСІВ.....	40
3.1	Багаторівневий метод кодування даних в СЗК.....	40
3.2	Виявлення помилок	45
3.3	Експериментальні дослідження	47
3.4	Запропонований метод кодування даних сенсорів на основі багаторівневої системи залишкових класів.....	49
3.5	Висновки.....	51
4	РЕАЛІЗАЦІЯ СИСТЕМИ КОДУВАННЯ, ПЕРЕДАЧІ ТА ОБРОБКИ ЗАКОДОВАНИХ ДАНИХ СЕНСОРІВ НА ОСНОВІ БАГАТОРІВНЕВОЇ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ.....	52
4.1	Використання інструменту програмування Node-RED для реалізації системи кодування та передачі даних.....	52
4.1.1	Симуляція генерації даних сенсорів в модельованій системі	59
4.1.2	Шифрування даних сенсорів в модельованій системі	61
4.1.3	Передача даних до системи збору даних в модельованій системі.....	63
4.2	Реалізація Web API для системи збору та обробки даних за допомогою ASP.NET Core.....	64
4.2.1	Опис проекту Core	65
4.2.2	Опис проекту WebApp.....	66
4.3	Реалізація візуальної частини системи збору та обробки даних за допомогою Angular.....	68
4.3.1	Компонент nav-menu.....	69
4.3.2	Компонент losses-table.....	70
4.4	Перевірка коректності роботи системи	70

4.4.1 Перевірка коректності роботи БСМ.....	71
4.4.2 Перевірка коректності роботи системи збору та обробки даних.....	71
4.5 Висновки	74
ВИСНОВКИ.....	75
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	76
ДОДАТОК А КОД РОЗРОБЕНОЇ БСМ.....	81
ДОДАТОК Б ТЕЗИ ДОПОВДІ КОНФЕРЕНЦІЇ АПКН.....	94
ДОДАТОК В ПРЕЗЕНТАЦІЯ.....	97

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЕОМ – електронно-обчислювальна машина

БСМ – безпроводна сенсорна мережа

СЧ – система числення

ПЗ – програмне забезпечення

СЗК – система залишкових класів

DTO – data transfer object

API – application programming interface

ВСТУП

Безпроводні технології передачі даних широко використовуються в різних галузях економіки, в тому числі в системах збору інформації про стан фізичного середовища, в задачах технічного та екологічного моніторингу та інших.

Однак, передача повідомлень малого обсягу (8 - 24 біт), які формують сенсори, приводить до неефективного використання пропускної здатності безпроводних сенсорних мереж (БСМ). Кожний пакет, який передається в БСМ, крім даних містить службову інформацію необхідну для надійної доставки і обробки пакетів. В [1, 2] показано, що відсоток службових даних у деяких протоколах становить від 4 до 80 % та залежить від довжини поля даних. Отже, розробка методів кодування направлених на зменшення обсягу службових даних є актуальною науковою задачею.

Метою роботи є розробка способу кодування та передачі даних сенсорів, який дозволить зменшити обсяг службової інформації в протоколах передачі даних безпроводних сенсорних мереж.

Поставлена мета досягається розв'язанням таких основних задач:

- здійснити аналіз БСМ, області їх застосування та характеристик;
- провести аналіз існуючих методів кодування даних сенсорів;
- розробити метод кодування на основі прямого та оберненого перетворення система залишкових класів;
- метод повинен забезпечувати кодування та декодування даних.
- створення віртуальної бездротової сенсорної мережі та системи збору та обробки даних для застосування запропонованого методу кодування.

Об'єктом дослідження є процеси кодування у системах збору даних сенсорів.

Предметом дослідження є методи та засоби кодування та передачі даних у сенсорних мережах на основі системи залишкових класів.

Наукова новизна отриманих результатів:

1. Отримав подальший розвиток метод кодування даних сенсорів, який забезпечує ефективне кодування даних сенсорів різної розрядності за рахунок використання відповідних модулів системи залишкових класів.

2. Удосконалено метод передачі даних у бездротових сенсорних мережах, що дозволило досягти зменшення обсягу службових даних у протоколах передачі даних.

Практична цінність отриманих результатів полягає у можливості застосування для підвищення ефективності використання пропускнуої здатності у бездротових сенсорних мережах.

В результаті виконаного наукового дослідження розроблено метод кодування та передачі даних сенсорів, що дозволяє скоротити обсяг службової інформації в протоколах передачі даних БСМ.

1 ДОСЛІДЖЕННЯ СЕНСОРНИХ СИСТЕМ ТА МЕТОДІВ ЗБОРУ ДАНИХ З СЕНСОРІВ

1.1 Визначення БСМ

Бездротова сенсорна мережа (БСМ) – розподілена мережа, яка здатна до самоорганізації та складається з певної кількості сенсорів (датчиків), головної станції, маршрутизаторів та інших пристроїв, які взаємодіють через радіосигнал. Такі мережі можуть покривати область від кількох метрів до десятків кілометрів за допомогою ретрансляторів, які передають інформацію між елементами мережі.

БСМ призначені для збору інформації різних середовищ та подальшої передачі цієї інформації до пунктів обробки інформації. Такі системи здатні вимірювати різні показники середовищ, такі як температура, звук, рівень забруднення, швидкість вітру, вологість тощо [3].

1.2 Области застосування БСМ

Моніторинг території. Один із найпоширеніших видів застосування БСМ. Сенсорна мережа розгортається на конкретній території, де потрібно відстежувати певні явища. Наприклад, в військовій сфері БСМ використовуються для виявлення вторгнення противника. Ще одним прикладом використання БСМ є геозагородження нафтопроводів.

Моніторинг охорони здоров'я. Сюди входять декілька типів сенсорних мереж для медичних застосувань: імплантовані, переносні та вбудовані в навколишнє середовище. Імплантовані медичні пристрої це ті, які вносяться всередину людського тіла. Переносні пристрої використовуються на поверхні тіла користувача або в безпосередній близькості до користувача. Системи, які вбудовані в навколишнє середовище використовують датчики, які розміщені в цьому ж середовищі. Можливі застосування включають відслідковування позиції тіла, місця знаходження, загальний моніторинг пацієнта в лікарні або вдома. Пристрої, вбудовані в навколишнє середовище, відстежують фізичний стан людини для

безперервної діагностики стану здоров'я, використовуючи дані мереж камер, датчиків підлоги та інших подібних пристроїв [4-6]. Системи, які встановлені на тіло людини, можуть збирати інформацію про фізичну форму і витрати енергії.

Моніторинг ареалів проживання тварин. Бездротові сенсорні мережі використовувалися для моніторингу різних видів і місць проживання, починаючи з розгортання острова Great Duck, включаючи бабаків, очеретяних жаб в Австралії та зебр в Кенії.

Зондування навколишнього середовища/Землі. Існує багато застосувань у моніторингу параметрів навколишнього середовища, приклади яких наведено нижче. Вони поділяються на додаткові категорії за проблемами, що пов'язані з суворими умовами та зниженим джерелом живлення [7]:

1. Моніторинг якості повітря [8]. За результатами експериментів, що вплив забрудненого повітря на людей може сильно відрізнятися. Для дослідницьких цілей були розгорнуті бездротові сенсорні мережі для моніторингу концентрації небезпечних газів для громадян (наприклад, у Лондоні [9]). Однак датчики для газів і твердих частинок можуть виходити з ладу через високу мінливість показників сенсорів мережі, перехресну чутливість та (концептуальний) дрейф. Більше того, якість даних наразі недостатня для прийняття надійних рішень, оскільки калібрування на місцях розгортання БСМ призводить до недостовірних результатів вимірювань, і може знадобитися повторне калібрування [10-12].

2. Виявлення лісових пожеж. БСМ може бути встановлена в лісі, щоб фіксувати початок пожеж. Вузли можуть бути оснащені датчиками для вимірювання температури, вологості та газів, які утворюються в результаті горіння дерев та рослин. Своєчасне виявлення пожежі має вирішальне значення для успішної дії пожежників; завдяки БСМ, пожежна команда має інформацію про час виникнення пожежі, її причина, та напрямок поширення.

3. Виявлення ґрунтових зсувів. Система виявлення зсувів використовує БСМ для виявлення незначних рухів ґрунту та змін різних параметрів, які вказують про те, що зсув відбувається або можливий найближчим часом. За допомогою зібраних даних можливо дізнатися про місце зсуву та уникнути жертв.

4. Моніторинг якості води. Моніторинг якості води включає аналіз властивостей води в дамбах, річках, озерах і океанах [13], а також запасів підземних вод. Використання багатьох бездротових розподілених датчиків дозволяє створити більш точну карту стану води і дозволяє розгортати станції моніторингу у важкодоступних місцях без необхідності отримання даних вручну.

5. Запобігання стихійним лихам. Бездротові сенсорні мережі можуть бути ефективними для запобігання руйнівних наслідків стихійних лих, наприклад повені. Є приклади успішного розгортання бездротових вузлів біля річок, де необхідно відстежувати зміни рівня води в режимі реального часу.

Промисловий моніторинг. Приклади застосування БСМ в промисловості:.

1. Моніторинг стану механізмів [14]. Існують БСМ, які розроблені для технічного обслуговування на основі стану механізмів, оскільки вони пропонують значну економію коштів і надають нові функції. Бездротові датчики можна розташовувати в місцях, важкодоступних або неможливих за допомогою дротової системи, таких як обертові механізми та неприкріплені транспортні засоби.

2. Реєстрація даних [15]. БСМ також використовуються для збору даних в рамках збору інформації відносно екологічному стану навколишнього середовища. Це може бути як контроль температури в холодильнику так і моніторинг рівня води в переливних баках на атомних електростанціях. Потім статистичну інформацію можна використовувати, щоб показати, ніскільки успішно працюють системи. Перевага БСМ перед звичайними реєстраторами полягає в тому, що можливий "живий" канал даних.

3. Моніторинг води/стічних вод. Моніторинг якості та рівня води включає різноманітність заходів, таких як перевірка якості підземних або поверхневих вод і забезпечення водної інфраструктури країни на користь як людей, так і тварин. Його можна використовувати для захисту від втрат води.

4. Моніторинг структурного стану. БСМ можна використовувати для моніторингу стану цивільної інфраструктури та пов'язаних із ними геофізичних процесів в реальному часі протягом тривалого періоду шляхом реєстрації даних, використовуючи відповідні інтерфейси датчиків.

5. Виробництво вина. БСМ використовуються для моніторингу виробництва вина, як у полі, так і в погребі [16].

Виявлення загрози. Wide Area Tracking System (WATS) – це прототип мережі для виявлення наземного ядерного пристрою, такого як ядерні ракети. WATS розробляється в Ліверморській національній лабораторії Лоуренса. WATS буде складатися з бездротових гамма- та нейтронних датчиків, з'єднаних бездротовою мережею зв'язку. Дані, які збираються датчиками, піддаються «злиттю даних», що перетворює інформацію в форми, що легко інтерпретуються; злиття даних є найважливішим аспектом системи. Процес злиття даних відбувається в сенсорній мережі, а не на централізованому комп'ютері, і виконується за спеціально розробленим алгоритмом на основі байєсівської статистики. WATS не буде використовувати централізований комп'ютер для аналізу, оскільки дослідники виявили, що такі фактори, як затримка та доступна пропускна здатність, як правило, створюють деякі проблеми. Дані, які обробляються в польових умовах самою мережею (шляхом передачі невеликих обсягів даних між сусідніми датчиками), швидше й роблять мережу більш масштабованою. Важливим фактором у розвитку WATS є простота розгортання, оскільки більше датчиків покращує швидкість виявлення та зменшує помилкові тривоги. Датчики WATS можуть бути розгорнуті на постійних позиціях або вмонтовані в транспортні засоби для мобільного захисту певних місць. Однією з перешкод для впровадження WATS є розмір, вага, потреби в енергії та вартість наявних на даний момент бездротових датчиків.

Моніторинг інцидентів. Існують дослідження, які показують, що використання датчиків для моніторингу інцидентів значно покращує реакцію пожежників і поліції на надзвичайну ситуацію. Для раннього виявлення інцидентів можна використовувати акустичні датчики, щоб виявити сплеск міського шуму через можливу аварію, або використовувати термодатчики для виявлення можливої пожежі [17-19].

1.3 Класифікація БСМ

Класифікація БСМ ускладнена широким переліком задач, які дані системи виконують. Тому для класифікації запропоновано застосовувати модель класифікації на основі трьох факторів:

1. За вимогами до швидкості передачі показників апаратної частини датчиків та сенсорів, БСМ поділяються на наступні категорії:

- 1) миттєвої передачі: показники передаються одразу після моменту їх фіксації, де затримка є мінімальною;
- 2) із низькою латентністю: показники передаються на протязі кількох секунд;
- 3) із високою латентністю: показники передаються із значною затримкою.

2. За типом організації живлення системи, БСМ поділяються на наступні категорії:

- 1) стаціонарні: живлення вузлів БСМ, незалежно від навантаження, відбувається з використанням екстерних мереж живлення або об'єктів живлення з високою ємністю;
- 2) напівстаціонарні: живлення вузлів БСМ високої навантаженості відбувається з використанням екстерних мереж живлення або об'єктів живлення з високою ємністю; кінцеві вузли БСМ забезпечені автономним живленням;
- 3) автономні: рядові вузли та ретранслятори забезпечені індивідуальними обмеженими джерелами живлення.

3. За розрахунком часу життя системи, БСМ поділяються на наступні категорії:

- 1) короткострокові: мережі, яка мають короткий час функціонування: від кількох годин до кількох днів;
- 2) середньострокові: мережі, які мають час функціонування до кількох місяців;
- 3) довгострокові: мережі, які мають час функціонування до кількох років.

Довгострокові системи з низькою латентністю, а також системи довгострокового функціонування стаціонарного призначення миттєвої реакції є найпоширенішими типами БСМ. Найпопулярнішими БСМ є системи першого класу. Для побудови яких доцільно використовувати найпростіші топології мереж (кластерне дерево або зірка), алгоритми маршрутизації більш високої складності та часової синхронізації а також індивідуальні методи відокремлення каналів для передачі даних.

1.4 Характеристики БСМ

Основні характеристики БСМ включають:

1. Енергоефективність: обмеження енергоспоживання вузлів які використовують батареї або збирають енергію [20, 21].

2. Стійкість: здатність справлятися зі збоями вузлів.

3. Мобільність: оптимальність фізичних характеристик вузлів, що дає змогу переміщати вузли.

4. Неоднорідність вузлів: можливість мережі містити вузли різного типу та забезпечувати зв'язок між ними.

5. Однорідність вузлів: висока ефективність мережі, яка складається з однотипних вузлів.

6. Масштабування: можливість розширювати мережу шляхом додання нових вузлів та елементів.

7. Пристосування: здатність мережі функціонувати у суворих умовах навколишнього середовища.

8. Простота експлуатації.

9. Міжрівнева оптимізація: забезпечення ефективної роботи у багаторівневих системах [22-24].

Перехресна багаторівневність є важливим об'єктом досліджень в бездротовій системі комунікації, так як традиційний шаровий підхід має наступні суттєві проблеми:

1. Відсутність можливості ділитися інформацією одного рівня системи з іншими, в результаті чого, різні рівні можуть мати різну або неповну інформацію. Також, традиційний підхід не може гарантувати оптимальної роботи мережі в цілому.

2. Підхід не дає можливості системі адаптуватися відповідно до змін навколишнього середовища.

3. Через колізії під час роботи багатьох користувачів одночасно, конфлікти доступу, згасання сигналу та зміни середовища в бездротових сенсорних мережах, традиційний підхід для дротових мереж не може застосовуватись до бездротових мереж.

Перехресна багаторівневність може застосовуватись для забезпечення оптимальної модуляції щоб підвищити продуктивність передачі даних, енергоефективність, якість роботи. Сенсорні вузли можна представити як міні-комп'ютери, які є дуже примітивними з точки зору компонентів, з яких вони складаються. Зазвичай вони складаються з процесора обмеженої потужності та невеликим об'ємом пам'яті, сенсорів, пристроїв комунікації та джерела живлення, яке, у більшості випадків, являє собою батарею. Іншими можливими компонентами є модулі збору енергії (такі як сонячні батареї), додаткові чіпи та порти комунікації (такі як RS-232 та USB).

Головна станція БСМ це один або декілька компонентів БСМ, які є більш потужнішими, мають сильніші джерела живлення а також більш складні ресурси комунікації. Вони діють як шлюз між вузлами мережі і кінцевим користувачем та передають інформацію з БСМ на сервер обробки. До інших важливих елементів БСМ входять маршрутизатори, які керують таблицями маршрутизації.

1.5 Топології мереж

Головна станція БСМ це один або декілька компонентів БСМ, які є більш потужнішими, мають сильніші джерела живлення а також більш складні ресурси комунікації. Вони діють як шлюз між вузлами мережі і кінцевим користувачем та

передають інформацію з БСМ на сервер обробки. До інших важливих елементів БСМ входять маршрутизатори, які керують таблицями маршрутизації.

Топологією мережі називають представлення фізичного розміщення вузлів мережі та ліній зв'язку між ними. В першу чергу це поняття застосовується по відношенню до локальних мереж, простеження структури яких, на відміну від глобальних мереж, не є складним.

Топологія мережі складається з фізичної та логічної структури. Фізична структура призначена для відображення фізичних з'єднань вузлів. Логічна структура представляється логічною організацією взаємодії вузлів. Таким чином, фізична і логічна структури взаємодоповнюються, дозволяючи отримати широке уявлення про структуру конкретної мережі.

Вибір топології залежить від наступних характеристик мережі:

1. Склад оснащення.
2. Характеристики оснащення.
3. Розширюваність мережі.
4. Спосіб управління мережею.

Деякі топології можуть мати ряд необхідних вимог, такі як тип кабелю для з'єднань вузлів мережі, тип його прокладки і т.д.

З безлічі способів з'єднань можна виділити наступні топології, які є базовими:

1. Зірка: мережа має явно виділений вузол, до якого підключаються інші вузли і через який здійснюється обмін інформацією.

2. Кільце: вузли підключені до кабелю, який замикається в коло. Сигнал в таких мережах передається по колу. Через відсутність вільного кінця, процес додання нового вузла є складним.

3. Шина: мережа складається з вузлів які підключені шляхом відгалуження від основного кабелю. Таким чином, кожний вузол може взаємодіяти з будь-яким іншим вузлом мережі без необхідності передавати інформацію через проміжні вузли.

Базові тоології мереж представлені на рисунку 1.1.

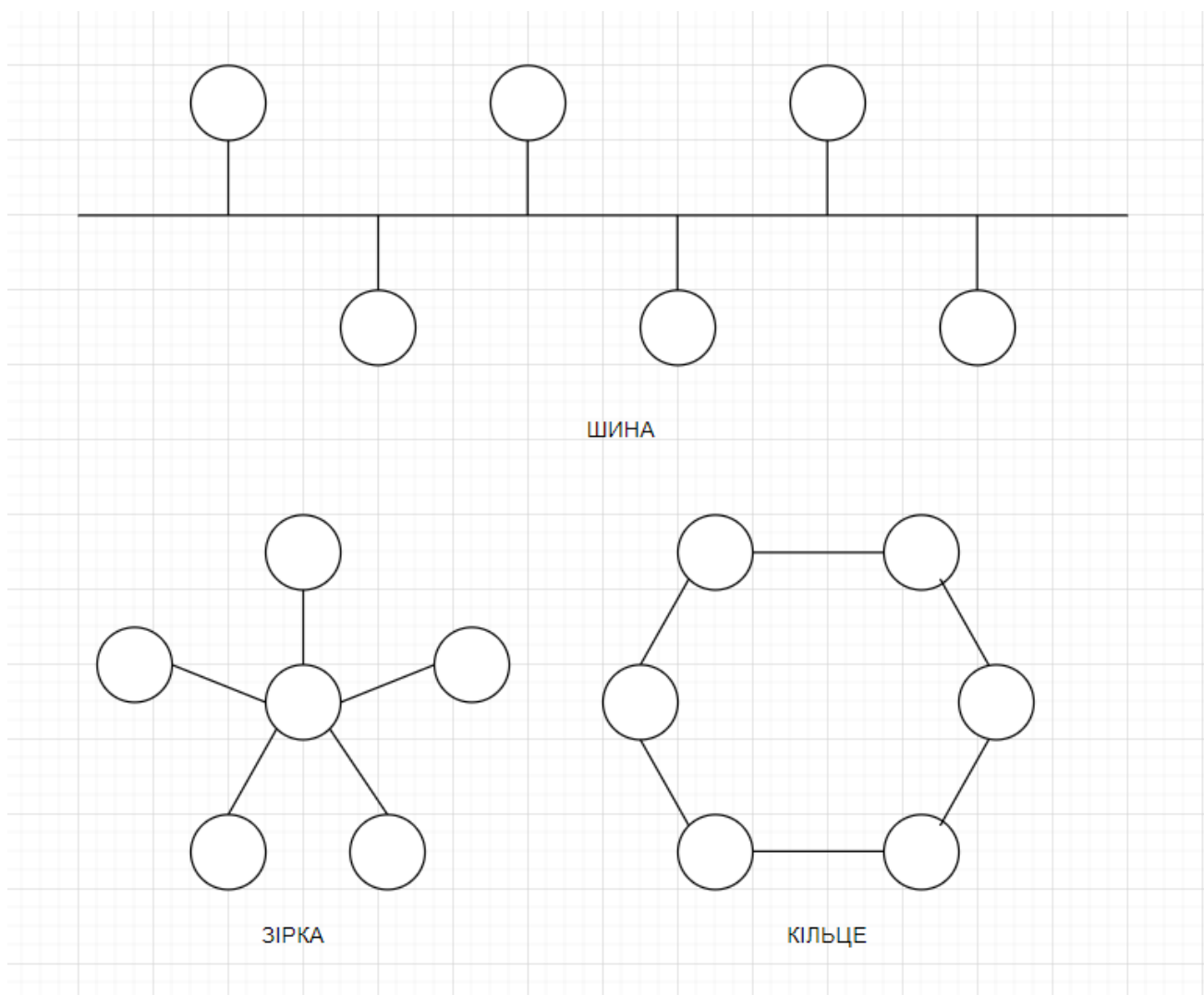


Рисунок 1.1 – Схематичне розміщення вузлів для базових топологій

Існують змішані або гібридні топології, які базуються на основних. Деякі з них мають власну назву. До таких топологій відносяться наступні:

1. Подвійне кільце.
2. Соти.
3. Решітка.
4. Дерево.
5. Fat tree.
6. Сніжинка.
7. Повнозв'язна.

Змішані топології мереж відображені на рисунку 1.2

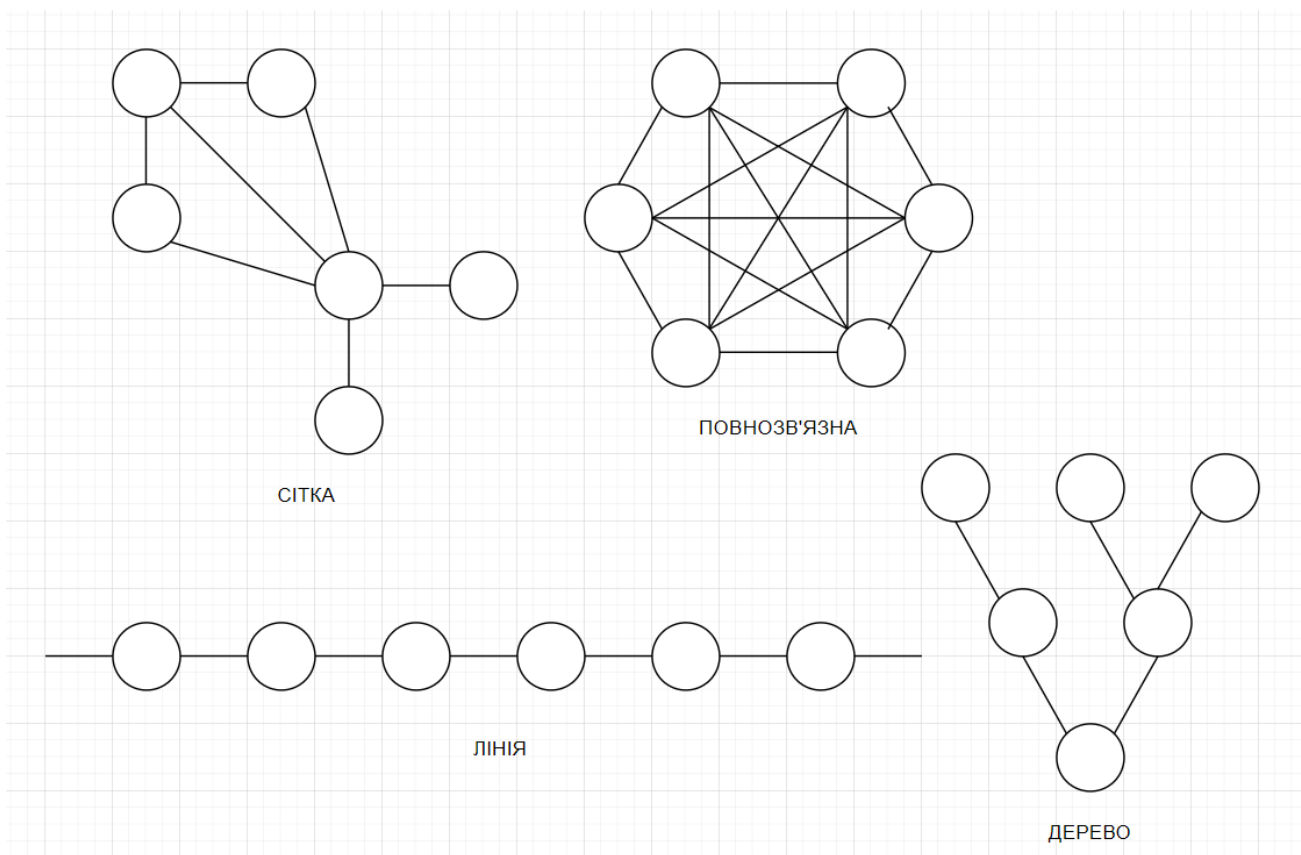


Рисунок 1.2 – Схематичне представлення топологій «Сітка», «Повнозв'язна», «Лінія» та «Дерево»

1.6 Сенсорний вузол

Сенсорний вузол, також відомий як *mote*, - це вузол у сенсорній мережі, який здатний виконувати певну обробку, збирати сенсорну інформацію та комунікувати з іншими вузлами в мережі. Сенсорний вузол являється вузлом сенсорної мережі, але не кожний вузол сенсорної мережі являється сенсорним вузлом.

Не дивлячись на існування бездротових сенсорних вузлів на протязі десятиліть і використовувалися для таких різноманітних застосувань, як вимірювання землетрусів до війни, сучасна розробка невеликих сенсорних вузлів розпочалась проектами *Smartdust* та *NASA Sensor Web*. Одна з головних цілей проекту *Smartdust* полягала у створенні автономного зондування та зв'язку у межах кубічного міліметра простору. Незважаючи на раннє закінчення проекту, він призвів до багатьох подальших дослідницьких проектів. Вони включають великі

дослідницькі центри в Berkeley NEST і CENS. Залучені до цих проектів дослідники, придумали термін *mote* для позначення сенсорного вузла. Еквівалентним терміном у проекті NASA Sensor Webs для фізичного сенсорного вузла є *rod*, хоча вузол датчика в Sensor Web може бути інша система Sensor Web. Фізичні сенсорні вузли змогли збільшити свої можливості в поєднанні із законом Мура. Мікросхема містить більш складні та енергоефективніші мікроконтролери. Таким чином, для того ж самого вузла можна додати більше силіконових можливостей. Сьогодні моти зосереджені на забезпеченні найбільшої бездротової мережі, що складає десятки кілометрів, найменшого споживання енергії (кілька мкА) та найпростішого процесу розробки для користувача.

Поширена структура сенсорного вузла відображена на рисунку 1.3.

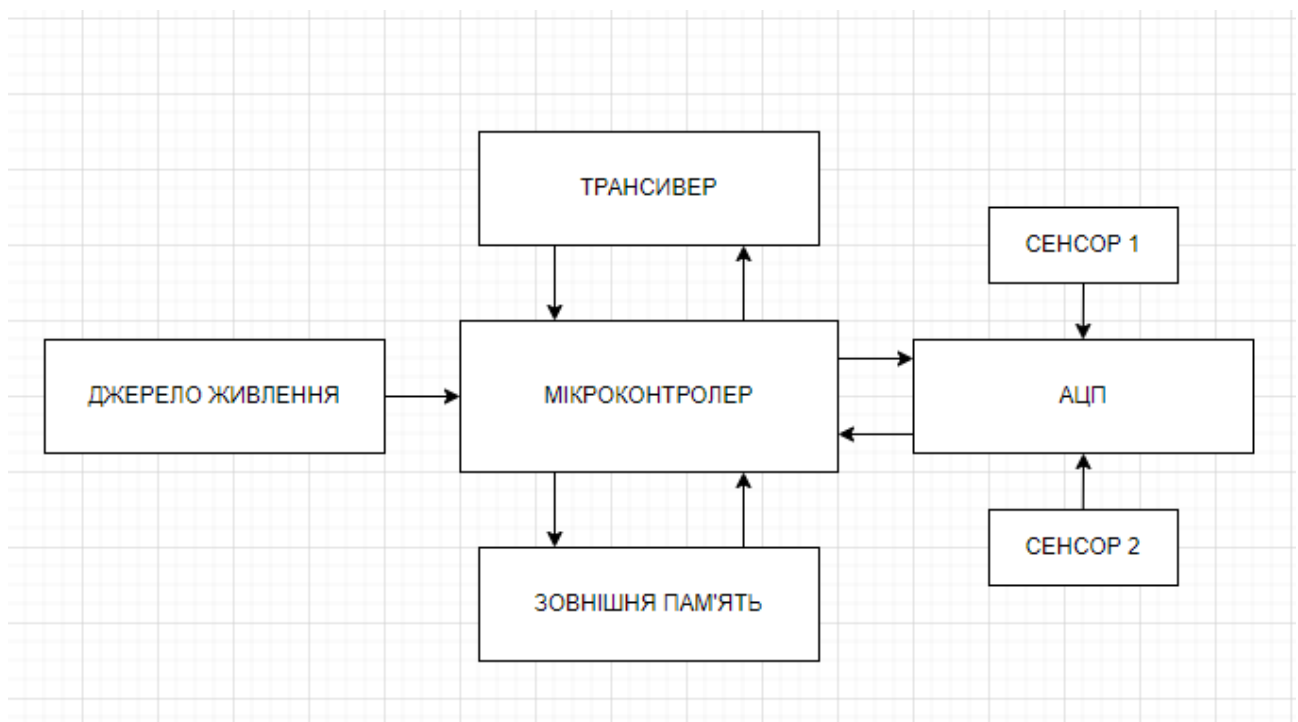


Рисунок 1.3 – Типова архітектура сенсорного вузла

Основні компоненти сенсорного вузла:

1. Контролер. Контролер виконує задачі, обробку даних та контролює функціонування інших компонентів у сенсорному вузлі. Хоча найпоширенішим контролером є мікроконтролер, іншими альтернативами, які можна

використовувати в якості контролера, є: настільний мікропроцесор загального призначення, процесори цифрових сигналів, FPGA та ASIC. Мікроконтролер часто використовується в багатьох вбудованих системах, таких як сенсорні вузли, через його низьку вартість, гнучкість підключення до інших пристроїв, простоту програмування та низьке енергоспоживання. Мікропроцесор загального призначення, як правило, споживає більше енергії, ніж мікроконтролер, тому він значно рідше використовується у сенсорних вузлах. Процесори цифрових сигналів можуть використовуватись для широкосмугового бездротового зв'язку, але в бездротовий зв'язок в БСМ часто скромний, тобто простіша, легша для обробки модуляція, а задачі обробки сигналу фактичного сприйняття даних менш складні. Тому переваги процесорів цифрових сигналів зазвичай не мають великого значення для БСМ. FPGA можна перепрограмувати та переналаштувати відповідно до вимог, але це займає багато часу та ресурсів.

2. Трансивер. Сенсорні вузли часто використовують діапазон ISM, характеристиками якого є радіо, розподіл спектру та загальна доступність. Можливий вибір середовищ бездротової передачі між радіочастотним, оптичним та інфрачервоним зв'язками. Для підтримки оптичного (лазерного) зв'язку необхідно менше енергії у порівнянні з іншими видами зв'язку, але цей тип зв'язку потребує прямої видимості та є чутливим до умов навколишнього середовища. Для інфрачервоного зв'язку, як і для лазерів, не потрібна антена, але він має обмежені можливості мовлення. Радіочастотний зв'язок є популярним у БСМ. БСМ, як правило, використовують неліцензовані частоти зв'язку: 173, 433, 868 і 915 МГц; і 2,4 ГГц. Функціональні можливості передавача і приймача об'єднані в один пристрій, відомий як приймач або трансивер. Часто, трансивери не забезпечені унікальним ідентифікатором. Операційними станами трансивера є передача, прийом, очікування та стан сну. Сучасні трансивери мають вбудовані кінцеві автомати, які виконують деякі операції автоматично. Більшість трансиверів мають однаковий рівень енергоспоживання у режимі сну і режиму прийому/передачі сигналу [25]. Тому, коли трансивер не здійснює прийом або передачу сигналу, для

збереження енергії краще його вимкнути. При перемиканні з режиму сну в режим передачі, витрачається значна кількість енергії.

3. Зовнішня пам'ять. З точки зору енергоспоживання, найбільш релевантними типами пам'яті є вбудована пам'ять мікроконтролера та флеш-пам'ять. Позачіпова оперативна пам'ять використовується рідко, якщо взагалі використовується. Флеш-пам'ять використовується через їх низьку вартість і задовільний об'єм для зберігання. Вимоги до пам'яті є сильно залежними від конкретних застосувань. За призначенням, пам'ять можна розділити на дві категорії: пам'ять користувача, яка використовується для зберігання даних, пов'язаних із програмою або особистих даних користувача, і системна пам'ять програми, яка використовується для програмування пристрою. Пам'ять програм також містить ідентифікаційні дані пристрою, якщо такі існують.

4. Джерела живлення. Бездротовий сенсорний вузол є популярним рішенням, коли важко або неможливо підключити мережеве живлення до сенсорного вузла. Однак, оскільки вузол бездротового датчика часто розміщується у важкодоступному місці, регулярна заміна батареї може бути дорогою і незручною. Важливим аспектом розробки бездротового сенсорного вузла є забезпечення постійного наявності достатньої енергії для живлення системи. Сенсорний вузол споживає енергію для виявлення, обміну інформацією та обробки даних. Для передачі даних потрібно більше енергії, ніж будь-який інший процес. Вартість енергії на передачу 1 Кб на відстань 100 метрів приблизно така ж, як і для виконання 3 мільйонів інструкцій процесором. Джерелами живлення є або батареї або конденсатори [26]. Батареї, як акумуляторні, так і безперезарядні, є основним джерелом живлення для сенсорних вузлів. Вони також класифікуються відповідно до електрохімічного матеріалу, який використовується для електродів, таких як NiCd (нікель-кадмій), NiZn (нікель-цинк), NiMH (нікель-металгідрид) і літій-іонний. Сучасні датчики здатні відновлювати свою енергію від сонячних джерел, радіочастот, перепадів температур або вібрації. Використовуються два основні принципи енергозбереження: динамічне керування живленням (ДКЖ) і динамічне масштабування напруги (ДШН). ДКЖ заощаджує електроенергію, вимикаючи

частини сенсорного вузла, які в даний момент не використовуються або не активні. Схема ДШН змінює рівні потужності всередині сенсорного вузла залежно від недетермінованого робочого навантаження. Змінюючи напругу разом із частотою, можна отримати суттєве зменшення рівня споживання електроенергії.

5. Сенсори (датчики). Сенсори використовуються вузлами бездротових датчиків для збору даних із навколишнього середовища, в якому вони розміщені. Це апаратні пристрої, які реагують на зміну фізичного стану, як температура або тиск. Датчики вимірюють фізичні дані параметра, що є об'єктом моніторингу моніторингу, і мають конкретні характеристики, такі як точність, чутливість тощо. Безперервний аналоговий сигнал, вироблений датчиками, оцифровується аналого-цифровим перетворювачем і надсилається на контролери для подальшої обробки. Деякі датчики містять необхідну електроніку для перетворення вихідних сигналів у показники, які можна отримати за допомогою цифрового каналу (наприклад, I2C, SPI), а багато перетворюються в одиниці вимірювання, наприклад, °C. Більшість сенсорних вузлів мають невеликі розміри, споживають мало енергії, працюють із високими об'ємами даних, є автономними та працюють без нагляду оператора, а також адаптуються до навколишнього середовища. Оскільки бездротові сенсорні вузли, як правило, є дуже малими електронними пристроями, вони можуть бути оснащені лише обмеженим джерелом живлення менше 0,5-2 ампер-години та 1,2-3,7 вольт. Датчики поділяються на три категорії:

- 1) пасивні, всенаправлені датчики;
- 2) пасивні вузькопроменеві датчики;
- 3) активні датчики.

Пасивні датчики сприймають дані, фактично не маніпулюючи навколишнім середовищем шляхом активного зондування. Вони автономні; тобто енергія потрібна лише для посилення їх аналогового сигналу. Активні датчики досліджують навколишнє середовище в реальному часі, наприклад, гідролокатор або радарний датчик, і вони потребують постійної енергії від джерела живлення. Вузькопроменеві датчики спрямовані на моніторинг області в конкретному напрямку, подібно до камери. Всенаправлені датчики не мають здійснюють

моніторинг всієї області, а не тільки конкретну частину, на відміну від вузькопроменевих. Більшість теоретичних робіт з БСМ передбачає використання пасивних всеспрямованих датчиків. Кожний вузол датчика має певну зону покриття, на якій він може надійно та точно здійснювати моніторинг. До найбільш енергопотребуючих процесів, які виконує датчик, можна віднести:

- 1) вибірка сигналу та перетворення фізичних сигналів в електричні;
- 2) формування сигналу;
- 3) аналого-цифрове перетворення.

Просторова щільність сенсорних вузлів у полі може досягати 20 вузлів на кубічний метр.

1.7 Висновки

У розділі наведено визначення бездротової сенсорної мережі. У результаті аналізу області застосування бездротових сенсорних мереж встановлено, що БСМ має широку область застосування у багатьох сферах діяльності: виробництво, охорона здоров'я, попередження руйнівних наслідків природних явищ а також, моніторинг.

Визначені основні класи та характеристики БСМ залежно від швидкості передачі показників сенсорів, способу організації живлення та часом життя системи.

Представлено основні топології та архітектури мереж які застосовуються в БСМ.

Наведено опис архітектури сенсорного вузла, наведено основні частини сенсорних вузлів та описано функції цих частин.

2 НЕДВІЙКОВІ СИСТЕМИ ЧИСЛЕНЬ ТА КОДУВАННЯ В СИСТЕМІ ЗАЛИШКОВИХ КЛАСІВ

2.1 Теорія недвійкових систем числення

Недвійкові системи числення (НСЧ), або недвійкові комп'ютерні арифметики (НКА), були розроблені для підвищення виконання математичних операцій різними електронно-обчислювальними машинами. Характеристиками обчислювальних машин, побудованих на базі таких систем числень, відповідно, є вдосконаленість швидкодії та надійності у порівнянні з машинами, що базуються на двійковій системі числення.

До НСЧ відносяться такі системи числення, як ієрархічні, непозиційну систему залишкових класів та інші.

В основі ієрархічних систем числень покладений принцип об'єднання непозиційних та позиційних систем числення, що дає змогу використовувати переваги даних систем числень та мінізувати недоліки, які існують при реалізації тієї чи іншої системи в чистому вигляді.

Кроки та принцип створення ієрархічних систем:

- 1) вибір деякої зовнішньої СЧ A з алфавітом α ;
- 2) кодування чисел цієї системи у вигляді слів окремої внутрішньої СЧ B з алфавітом β ;

Яскравим прикладом ієрархічної СЧ є двійково-десятькова, задачею якої є представлення десятикової СЧ у деяких ЕОМ.

Принципом системи залишкових класів є представлення кодованих чисел залишками від ділення на обрану систему основи, які являються простими числами.

Такі математичні операції, як додавання, віднімання та множення чисел в СЗК відбувається індивідуально по кожній основі, без необхідності переносів між розрядами.

Більш складних алгоритмів в СЗК потребують операції порівняння, розподілу та інші. Це основним недоліком, через який СЗК має вузьке застосування. Незважаючи на недоліки, СЗК це єдина система числення в сучасних

обчислювальних машинах, яка здатна працювати з великими та надвеликими числами з можливістю давати результати операцій в режимі реального часу. Тому, дослідники світу спрямовують зусилля на удосконалення СЗК шляхом підвищення швидкодії при здійсненні деяких операцій, які потребують складних і повільних алгоритмів.

2.2 Система залишкових класів

Система числення залишкових класів базуються на відношенні конгруентності [27, 28], що має наступне визначення: два цілі числа a і b називаються конгруентними за модулем m , якщо m точно ділить різницю a і b . У математичному вигляді це можна описати наступним чином: $a \equiv b \pmod{m}$. Приклади, $20 \equiv 2 \pmod{6}$, $20 \equiv 2 \pmod{3}$, $20 \equiv 2 \pmod{3}$ і $20 \equiv 4 \pmod{8}$.

Якщо q і r - частка і залишок відповідно поділу цілого числа a на m , тобто $a = q \cdot m + r$ - тоді, за визначенням, маємо $a \equiv r \pmod{m}$. Число називається залишком a відносно до m , далі це буде позначатися як $r = |a|_m$. Набір найменших значень при діленні на m , $\{0, 1, 2, \dots, m - 1\}$, вважається залишками та називається множиною найменших позитивних залишків за модулем m . Якщо інші залишки не зазначені, можна припустити що інших значень залишок приймати не може.

Нехай у нас є множина $\{m_1, m_2, \dots, m_N\}$, з N додатніх і попарних відносно простих модулів. Нехай M - добуток модулів. Потім кожне число $X < M$ має унікальне представлення в системі числення залишку, яка є множиною залишків $\{|X|_{m_i} : 1 \leq i \leq N\}$. Часткове підтвердження цього полягає в наступному. Припустимо, що X_1 і X_2 - два різні числа з однаковими наборами залишків. Тоді $|X_1|_{m_i} = |X_2|_{m_i}$, і тому $|X_1 - X_2|_{m_i} = 0$. Отже $X_1 - X_2$ - найменше спільне кратне (НСД) m_i . Але якщо m_i є відносно простим числом, то їх НСД дорівнює M , тому $X_1 - X_2$ є кратне M . Отже, не може бути, що $X_1 < M$ і $X_2 < M$. Отже, встановити $\{|X|_{m_i} : 1 \leq i \leq N\}$ є унікальним і може бути взятим як подання X . Записавши таке подання у вигляді $\{x_1, x_2, \dots, x_n\}$, де $x_i = |X|_{m_i}$, можна вказати на зв'язок між X і його залишковим записом $X \sim = \{x_1, x_2, \dots, x_n\}$. Число M називається динамічний

діапазон СЗК, оскільки можливих чисел представлено M . Для чисел без знака цей діапазон дорівнює $[0, M - 1]$.

Представлення системи в якій модулі не є відносно простими, не будуть унікальними: два чи більше числа будуть представлені однаково. Наприклад, залишки цілих чисел від нуля до п'ятнадцяти відносно модулів два, три і п'ять (які є відносно простими) наведено в лівій половині таблиці 2.1. І залишки цих самих чисел щодо модулів два, чотири та шість (які не є відносно простими) наведено в правій половині тієї ж таблиці. Слід звернути увагу, що в першій половині таблиці повторів в послідовності залишків немає, тоді як повтори є в другій половині таблиці.

Таблиця 2.1 – Залишки цілих чисел від 0 до 15 відносно взаємно простих модулів та модулів, які не є взаємно простими

N	Відносно прості модулі			Не відносно прості модулі		
	m1 = 2	m2 = 3	m3 = 5	m1 = 2	m2 = 4	m3 = 6
0	0	0	0	0	0	0
1	1	1	1	1	1	1
2	0	2	2	0	2	2
3	1	0	3	1	3	3
4	0	1	4	0	0	4
5	1	2	0	1	1	5
6	0	0	1	0	2	0
7	1	1	2	1	3	1
8	0	2	3	0	0	2
9	1	0	4	1	1	3
10	0	1	0	0	2	4
11	1	2	1	1	3	5
12	0	0	2	0	0	0
13	1	1	3	1	1	1
14	0	2	4	0	2	2
15	1	0	0	1	3	3

2.2.1 Вибір модулів

Найкращими модулями, ймовірно, є прості числа, принаймні з чисто математичної точки зору. Особливо корисною властивістю таких модулів є «корінь». Якщо модуль m простий, то існує принаймні один примітивний корінь (або степінь), $p \leq m - 1$, такий, що множина $\{ |p^i|_m : i = 0, 1, 2, \dots, m - 2 \}$ є множиною всіх ненульових залишків відносно m . Наприклад, якщо $m = 7$, то ми можемо взяти $p = 3$, оскільки $\{ |3^0|_7 = 1, |3^1|_7 = 3, |3^2|_7 = 2, |3^3|_7 = 6, |3^4|_7 = 4, |3^5|_7 = 5 \} = \{1, 2, 3, 4, 5, 6\}$; 5 також є примітивним коренем. Очевидно, що для таких модулів множення та піднесення залишків до степеня можна проводити в рамках простих операцій над взятими простими коренями цих самих залишків, подібно до використання логарифмів та антилогарифмів у звичайному множенні.

Для комп'ютерних додатків важливо мати набори модулів, які полегшують як ефективне представлення, так і баланс, де останнє означає, що різниця між модулями має бути якомога меншою. Візьмемо, наприклад, вибір 13 і 17 для модулів, які є взаємно простими числами; найменше спільне кратне становить 221. При простому двійковому кодуванні чотири біти та п'ять біт відповідно будуть потрібні для представлення відповідних залишків. У першому випадку ефективність представлення дорівнює $13/16$, а в другому – $17/32$. Якщо замість цього ми виберемо 13 і 16, то ефективність репрезентації буде покращена — до $16/16$ у другому випадку — але ціною зменшення спільного кратного (до 208). З іншого боку, з кращою збалансованою парою, 15 і 16, ми мали б і кращу ефективність, і більший діапазон: $15/16$ і $16/16$ для першого і 240 для останнього.

Також корисно мати модулі, які спрощують реалізацію математичних дій. Це неодмінно означає, що математичні операції на залишках не повинні занадто сильно відхилятися від математичних операцій на початкових числах. Поширеним вибором простого модуля, який не ускладнює математичні операції і має непогану ефективність представлення, є $m_i = 2^i - 1$. Не всі пари чисел виду $2^i - 1$ є відносно простими, але можна побачити, що $2^j - 1$ і $2^k - 1$ є відносно простими тоді і тільки тоді, коли j і k є відносно простими. Багато наборів модулів базуються на цих

варіантах, але є й інші варіанти; наприклад, множини модулів виду $\{2n - 1, 2n, 2n + 1\}$ є одними з найпопулярніших у використанні.

Отже, загалом слід враховувати чотири фактори при виборі модулів. По-перше, вибрані модулі повинні забезпечувати достатній діапазон, водночас гарантуючи, що представлення найменшого спільного кратного є унікальним. По-друге, як зазначено вище, ефективність бінарних операцій; у цьому відношенні також важливий баланс між різними модулями в заданому наборі модулів. Третє полягає в тому, що в ідеалі реалізації арифметичних одиниць для найменшого спільного кратного для залишків мають бути певною мірою сумісні з такими одиницями для початкових чисел, особливо враховуючи «спадщину», яка існує для останньої. І четверте - це розмір окремих модулів: хоча деякі арифметичні операції СЗК не вимагають перенесення між цифрами, що є однією з основних переваг СЗК, це відбувається лише між цифрами. Оскільки цифра в кінцевому підсумку представлена у двійковому форматі, між бітами будуть перенесення, і тому важливо переконатися, що цифри (і модулі) не є занадто великими. Цифри з низькою точністю також дозволяють реалізувати ефективне виконання арифметичних операцій для пошуку в таблиці. Але, з іншого боку, якщо модулі малі, то для забезпечення достатнього динамічного діапазону може знадобитися їх велика кількість. Звичайно, в кінцевому підсумку вибір, який буде зроблено, а також чи є СЗК ефективною чи ні, залежить від конкретних додатків і технологій.

2.2.2 Робота з від'ємними числами

Деякі програми потребують можливості представляти як негативні, так і додатні числа за допомогою залишків. Як і у звичайних системах числення, для такого представлення в СЗК може використовуватися будь-який метод з доповнення до основи, доповнення до зниженого кореня або позначення знака та величини. Переваги та недоліки вибору одного чи іншого методу подібні до додатніх значень. На відміну від додатніх значень, однак, визначення залишку для числа з урахуванням знаку є набагато важчим процесом. Це має місце навіть із

записом знака для значень, оскільки визначити знак результату математичної операції, наприклад додавання чи віднімання, непросто, навіть якщо знаки операндів відомі.

Розширення СЗК для позначення знаку числа передбачає використання однієї знакової цифри або додавання до кожного залишку у поданні додаткового біта або цифри для знака. Для позначень доповнення діапазон представлених чисел зазвичай розбивається на дві приблизно рівні частини, так що приблизно половина чисел є додатніми, а решта від'ємними. Таким чином, якщо використані модулі m_1, m_2, \dots, m_N , то в наборі чисел $M = \prod_{i=1}^N m_i$ кожне представлене число X , задовольняє одне з двох співвідношень, блок 2.1 якщо M непарне або блок 2.2 якщо M парне.

$$-(M - 1) / 2 \leq X \leq (M - 1) / 2. \quad (2.1)$$

$$-M / 2 \leq X \leq M / 2 - 1. \quad (2.2)$$

Тоді, для запису доповнення, якщо $\{x_1, x_2, \dots, x_N\}$ є представленням X , де $x_i = |X|_{m_i}$, тоді представленням $-X$ є $\{x_1, x_2, \dots, x_N\}$, де x_i – доповнення до m_i , тобто доповнення до основи, або доповнення до $(m_i - 1)$, тобто доповнення до зменшеного кореня від x_i . Наприклад, із набором модулів $\{2, 3, 5, 7\}$ подання сімнадцяти є $\{1, 2, 2, 3\}$, а представлення його заперечення у доповненні є $\{1, 1, 3, 4\}$, від $\{2 - 1, 3 - 2, 5 - 2, 7 - 3\}$. Виправленням для доповнення (заперечення) кожної цифри залишку є $|m_i - x_i|_{m_i} = -x_i$.

Якщо взяти $[0, M - 1]$ як номінальний діапазон СЗК, тоді буде видно, що в останньому прикладі $\{1, 1, 3, 4\}$ також є представленням 193. Тобто, розбиваючи діапазон $[0, M - 1]$, за додатні числа береться піддіапазон $[0, M/2 - 1]$, якщо M парне, або $[0, (M - 1)/2]$, якщо M непарне, і, відповідно, за від'ємні числа береться діапазон $[M/2, M - 1]$ або $[(M + 1) / 2, M - 1]$. Це справедливо, оскільки $|M - X|_M = -X$ (в останньому прикладі $210 - 17 = 193$).

Якщо знак не вказано, зазвичай вважається, що потрібно працювати тільки з додатними числами.

2.2.3 Базові математичні операції

Стандартні математичні операції додавання, віднімання та множення легко реалізуються за допомогою позначення залишків, залежно від вибору модулів, але ділення набагато складніше. Останнє не дивно, у світлі наведеного вище твердження про труднощі визначення знака та порівняння величини.

Додавання залишків здійснюється шляхом окремого додавання відповідних цифр щодо модуля їх положення. Тобто виконання з позиції однієї цифри не поширюється на позицію наступної цифри.

Як приклад, із набором модулів $\{ 2, 3, 5, 7 \}$ представлення сімнадцяти є $\{ 1, 2, 2, 3 \}$, дев'ятнадцяти - $\{ 1, 1, 4, 5 \}$, і додавання двох залишків дає результат $\{ 0, 0, 1, 1 \}$, що є представленням для числа тридцять шість в цій системі.

Віднімання може бути здійснене шляхом заперечення (у будь-якому обраному позначенні), мінусом або додаванням від'ємного числа. Це просто для чисел у записах доповнення або доповнення зі зменшеним коренем. Для чисел, представлених зі знаком та величиною залишку, необхідна невелика модифікація алгоритму для звичайних знаків і величини: знаковий розряд розгортається до всіх позицій у представленні залишку, а потім додавання відбувається, як у випадку для чисел без знаків, але зі звичайним алгоритмом знака та величини.

Множення також можна виконати простим перемноженням відповідних пар чисел-залишків відносно модуля; тобто множити цифри та ігнорувати чи коригувати відповідну частину результату. Якщо дані модулі m_1, m_2, \dots, m_N , $X \sim \{ x_1, x_2, \dots, x_N \}$ і $Y \sim \{ y_1, y_2, \dots, y_N \}$, тобто $x_i = |X|_{m_i}$ і $y_i = |Y|_{m_i}$, то ми можемо представити множення $X \times Y = Z$ як $X \times Y \sim \{ x_1, x_2, \dots, x_N \} \times \{ y_1, y_2, \dots, y_N \} = \{ z_1, z_2, \dots, z_N \}$ де $z_i = |x_i \times y_i|_{m_i} \sim Z$.

Як приклад, із набором модулів $\{ 2, 3, 5, 7 \}$ представлення сімнадцяти є $\{ 1, 2, 2, 3 \}$, дев'ятнадцяти - $\{ 1, 1, 4, 5 \}$, а їх добуток, триста двадцять три, це $\{ 1, 2, 3, 1 \}$. Як і при додаванні, отримання модуля по відношенню до m_i може бути реалізовано без поділу, і досить ефективно, в разі якщо m_i має відповідну форму.

Основне ділення з фіксованою точкою, по суті, складається з послідовності віднімань, порівнянь величини та виділення часткових цифр. Але порівняння в СЗК є складною операцією, оскільки СЗК не є позиційною. Наприклад, той факт, що з набором модулів $\{ 2, 3, 5, 7 \}$ число сімнадцять представлено як $\{ 1, 2, 2, 3 \}$, дев'ятнадцять представлено як $\{ 1, 1, 4, 5 \}$, але це далеко не очевидно. Тому очевидно, що поділ буде складною операцією. Одним із способів легкого виконання операції ділення є перетворення операндів у звичайний вигляд, використовуючи звичайну процедуру ділення, а потім перетворювати результат назад у представлення залишку. Однак перетворення займають багато часу, і, якщо можливо, слід використовувати прямий алгоритм. Таким чином, суть хорошого алгоритму поділу в СЗК буде відносно ефективним методом виконання порівнянь величин. У певному сенсі всі алгоритми вимагають, по суті, перетворення з СЗК, і, у порівнянні зі звичайними алгоритмами поділу, всі вони є досить неефективними.

2.2.4 Перетворення

Найпростіший спосіб перетворення з звичайного представлення числа до залишкового це процес, що відомий як пряме перетворення - це поділити на кожен із заданих модулів, а потім зібрати залишки. Однак це, ймовірно, буде затратною операцією, якщо число представлено в довільному корені і модулі довільні. Якщо, з іншого боку, число представлено в основі-2 (або корені, який є степенем двійки), а модулі мають відповідну форму (наприклад, 2^n-1), то існують процедури, які можна реалізувати з високою ефективністю. Перетворення від залишкового вигляду до звичайного, це процес, відомий як зворотне перетворення, є складнішим (концептуально, але не обов'язково в реалізації) і до цих пір був однією з основних перешкод на шляху до використання СЗК. Один із способів, яким це можна зробити - призначити ваги цифрам в залишковому записі, а потім створити з цього «звичайне» (тобто позиційне, зважене) змішане подання. Це змішане подання можна потім перетворити на будь-яку звичайну форму представлення. На практиці використання процедури прямого перетворення для останнього можна уникнути,

виконавши математичні операції перетворення залишків для результату. Інший підхід передбачає використання китайської теореми залишку, яка є основою багатьох алгоритмів перетворення від залишкового представлення до звичайного.

2.3 Области застосування системи залишкових класів

Наводиться приклад, який демонструє типове застосування систем числення залишків: операція множення та додавання послідовності скалярів. Це операція, яка дуже часто зустрічається в програмах обробки цифрових сигналів, одна з областей, для яких СЗК ефективно застосовується.

Наприклад, є множина модулів $\{m_i\} = \{2, 3, 5, 7\}$. Динамічний діапазон цього модуля 210. Припустимо, ми хочемо оцінити суму добутків $7 \times 3 + 16 \times 5 + 47 \times 2$. Множини залишків є:

- 1) $|2|_{m_i} = \{0, 2, 2, 2\}$;
- 2) $|3|_{m_i} = \{1, 0, 3, 3\}$;
- 3) $|5|_{m_i} = \{1, 2, 0, 5\}$;
- 4) $|7|_{m_i} = \{1, 1, 2, 0\}$;
- 5) $|16|_{m_i} = \{0, 1, 1, 2\}$;
- 6) $|47|_{m_i} = \{1, 2, 2, 5\}$.

Спочатку обчислюються добутки шляхом множення відповідних залишків:

- 1) $|7 \times 3|_{m_i} = \{|1 \times 1|_2, |1 \times 0|_3, |2 \times 3|_5, |0 \times 3|_7\} = \{1, 0, 1, 0\}$
- 2) $|16 \times 5|_{m_i} = \{|0 \times 1|_2, |1 \times 2|_3, |1 \times 0|_5, |2 \times 5|_7\} = \{0, 2, 0, 3\}$
- 3) $|47 \times 2|_{m_i} = \{|1 \times 0|_2, |2 \times 2|_3, |2 \times 2|_5, |5 \times 2|_7\} = \{0, 1, 4, 3\}$

Тепер, після обчислення добутків, суму добутків можна представити, додавши відповідні залишки:

$$|7 \times 3 + 16 \times 5 + 47 \times 2|_{m_i} = \{|1 + 0 + 0|_2, |0 + 2 + 1|_3, |1 + 0 + 4|_5, |0 + 3 + 3|_7\} = \{1, 0, 0, 6\}$$

СЗК також корисні для виявлення та виправлення помилок. Це можливо, враховуючи незалежність цифр у представленні залишкового числа: помилка в одній цифрі не пошкоджує інші цифри. Загалом, використання надлишкових

модулів, тобто додаткових модулів, які не відіграють ролі у визначенні динамічного діапазону, полегшує як виявлення помилок, так і виправлення. Але навіть без надлишкових модулів можлива стійкість до збою в системі, оскільки обчислення можуть продовжуватися після ізоляції несправних позицій цифр, за умови, що менший динамічний діапазон прийнятний. Нарешті, СЗК може допомогти прискорити математичні операції наж комплексними числами: наприклад, множення двох комплексних чисел вимагає чотирьох операцій множення і двох додавання, якщо виконувати традиційно, і два множення та два додавання з правильним типом СЗК. Загальна структура процесора СЗК показана на рисунку 2.1.

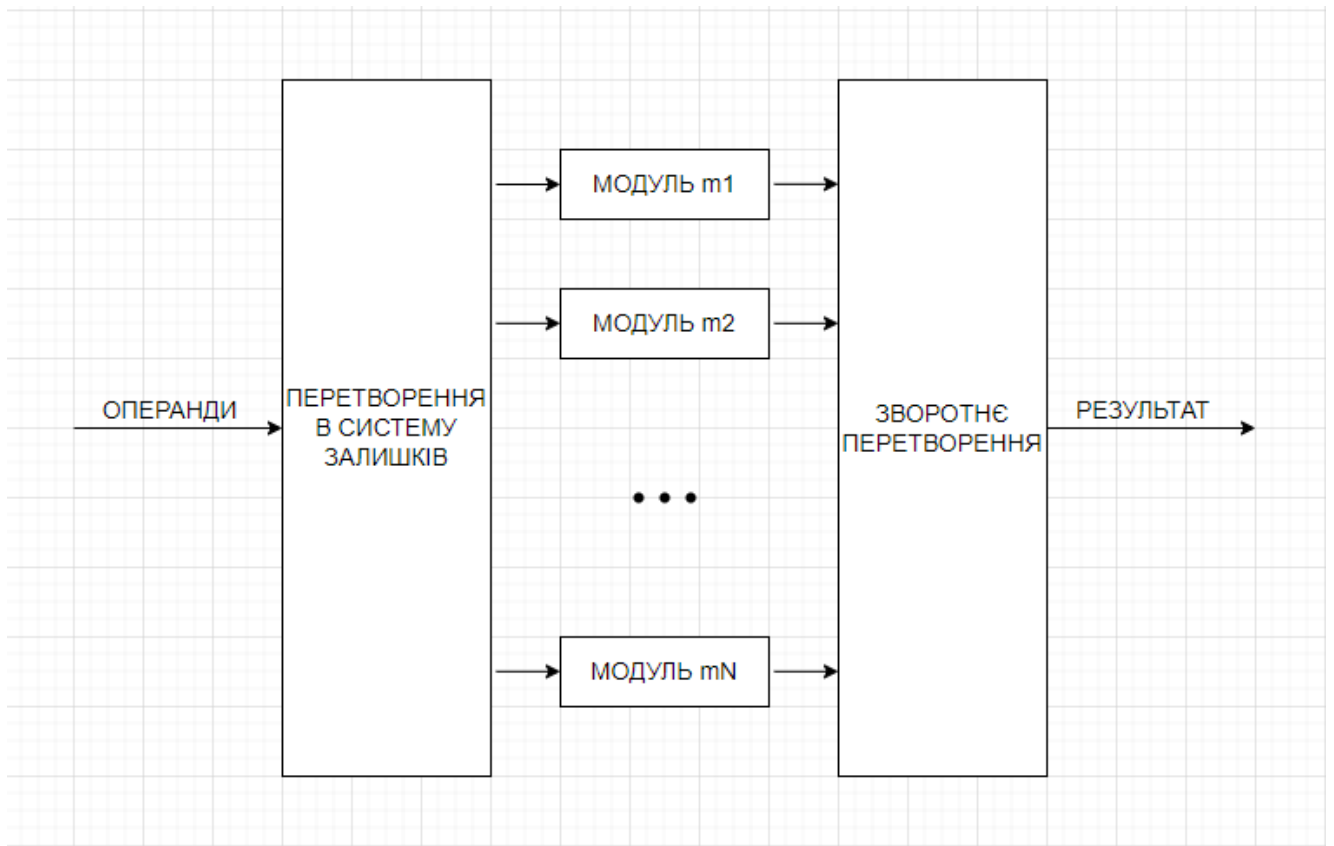


Рисунок 2.1 – Загальна структура процесора СЗК

2.4 Обробка цифрових сигналів

Більшість сигналів реального світу є аналоговими, і обробка таких сигналів для отримання корисної інформації досить утомлива. Більше того, це ускладнюється майже повною відсутністю аналогових комп'ютерів. Отже, цифрова обробка сигналів є основним засобом для вилучення інформації. У обробці цифрових сигналів (ОЦС) вхідний сигнал перетворюється в послідовність чисел, що є результатом дискретизації та квантування аналогових сигналів. У процесі вилучення корисної інформації сигнали, як правило, перетворюються з одного типу представлення в інший тип, в якому певні характеристики сигналу стають очевидними. Основна перевага, яку пропонує ОЦС, полягає в гнучкості, пов'язаної з генерацією та модифікацією алгоритмів, що використовуються для цих перетворень. Крім того, цифрова обробка сигналів пропонує інші переваги, такі як легкість програмування, стабільність, повторюваність і легка реалізація систем. Під системами маються на увазі схеми, які реалізують прикладні алгоритми в інженерних галузях, таких як зв'язок, обробка зображень, шифрування, стиснення даних для передачі і так далі. Цифрова обробка також дозволяє легко виправляти помилки, які виникають під час передачі сигналу або під час зберігання даних.

Аналіз сигналів і прикладні алгоритми в ОЦС потребують багато обчислень. Попит виникає через те, що найбільш часто використовуваною математичною операцією майже в усіх алгоритмах ОЦС є сума добутків, яка складається з послідовностей множення та додавання. Множення може бути затратною операцією з точки зору швидкодії та задіяності мікросхеми, і тому може становити певні складності у реалізації алгоритмів ОЦС. Операція суми добутків має бути дуже швидкою, особливо це стосується програм, які вимагають обробки в режимі реального часу, де особливе значення має обсяг обробки, який можна виконати до надходження нових даних. Оскільки множення займає більше часу, ніж додавання, якщо множення можна прискорити, то загальний час обробки можна суттєво зменшити. Множення великих чисел займає багато часу або дорого коштує в апаратній частині, або й те й інше; тому перетворення даних у систему залишків

може ефективно скоротити час і витрати на множення: при правильному виборі множників модулів множники можуть бути представлені в системі залишків, що, у свою чергу, призводить до менших і швидших множників.

2.4.1 Цифрові фільтри

Цифрові фільтри особливо важливі в ОЦС, оскільки їх можна використовувати для широкого спектру застосувань: зменшення шуму, розщеплення смуги, обмеження смуги, інтерполяція, децимація, формування імпульсів, приглушення відлуння, вирівнювання тощо. Перевагою цифрових фільтрів, порівняно з аналоговими, є їх повторюваність. Це пояснюється тим, що аналогові фільтри використовують такі компоненти, як резистори та конденсатори, які мають певний допуск. З іншого боку, цифрові фільтри в основному покладаються на операцію суми добутків, і для одного і того ж набору операндів це дає однаковий результат щоразу, коли виконується ця операція. Крім того, систему ОЦС можна легко адаптувати до змін середовища шляхом простої зміни параметрів системи, що не є характеристикою аналогових систем. Крім того, системи ОЦС можна перепрограмувати, і тому, наприклад, їх характеристики від фільтра нижніх частот до фільтра високих частот або смугового фільтра можна змінити без будь-яких фізичних змін.

У ОЦС зазвичай реалізуються два основних типи фільтрів: фільтр з скінченною імпульсивною характеристикою (finite impulse response або FIR) і фільтр з нескінченною імпульсивною характеристикою (infinite impulse response або IIR).

2.4.2 FIR фільтри

Вхідний сигнал цифрового фільтра, як правило, є послідовністю чисел, отриманої в результаті вибірки аналогового сигналу. FIR фільтр може бути представлений блок-схемою типу, показаного на рисунку 2.2. Блок z^{-1} представляє

одиничну затримку. Якщо вхідним є $x(n)$, то вихід елемента затримки дорівнює $x(n - 1)$, що є значенням $x(n)$ за один період часу до цього, або, простіше кажучи, попередній вхід. Аналогічно, $x(n - 2)$ просто означає значення вхідних даних за два періоди вибірки до цього моменту. Фільтр, показаний на рисунку 2.3, може бути представлений різницевим рівнянням (блок 2.3).

$$y(n) = a_0x(n) + a_1x(n - 1) + a_2x(n - 2). \quad (2.3)$$

Кількість розгалужень у фільтрі називається кранами фільтра. Фільтр можна описати в термінах його імпульсної характеристики, яка є серією зважених імпульсів. За допомогою імпульсної характеристики легко обчислити вихід фільтра, просто помноживши імпульсну характеристику та вхідний ланцюг вибірових імпульсів, присутніх у конкретний час. Ідею елементів затримки можна використовувати щоб переписати різницеве рівняння і таким чином отримати передатну функцію. Оскільки z^{-1} означає елемент затримки, в результаті отримується наступне рівняння:

$$y(n) = (a_0 + a_1z^{-1} + a_2z^{-2}) \times x(n). \quad (2.4)$$

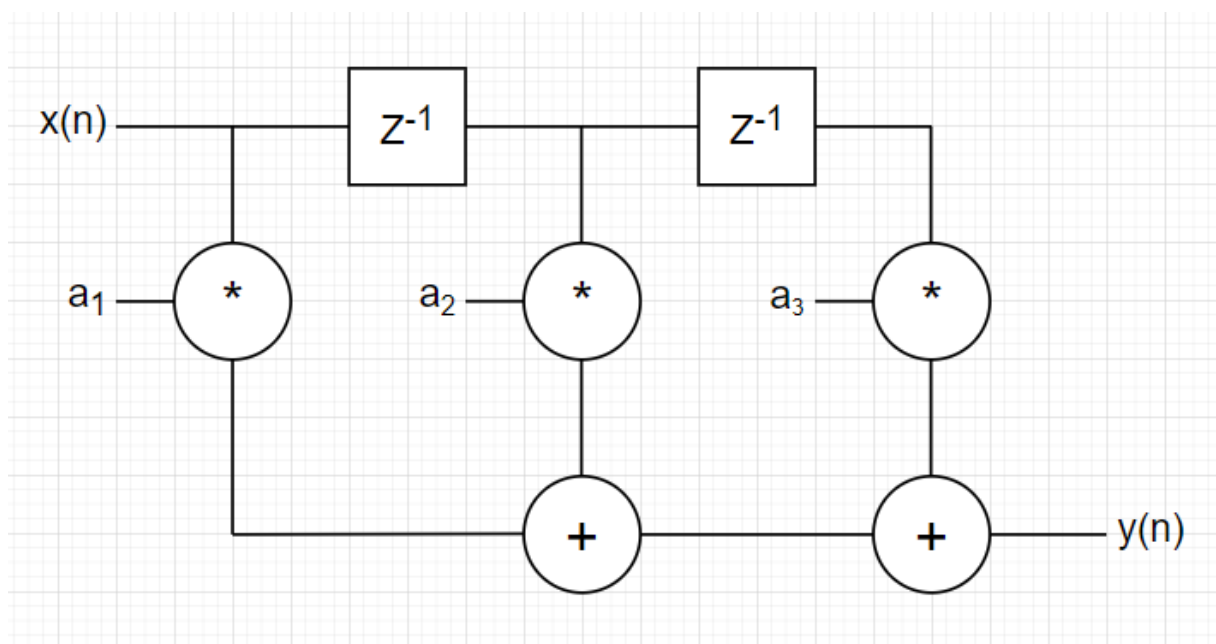


Рисунок 2.2 – FIR фільтр з трьома натискачами

Імпульсна характеристика фільтра є:

$$h(n) = y(n) / x(n) = a_0 + a_1z^{-1} + a_2z^{-2}. \quad (2.5)$$

У випадку з FIR фільтром імпульсна характеристика має лише чисельний поліном, корені якого називають нулями фільтра. FIR фільтр за своєю суттю стабільний через відсутність полінома знаменника в його імпульсній характеристиці.

Щоб краще зрозуміти реалізацію, наведено фільтр другого порядку, який розглянутий раніше на рисунку 2.2. В цього фільтра $m = 2$, а вихід у будь-який момент часу визначається рівнянням 2.6.

$$y(n) = a_0x(n) + a_1x(n - 1) + a_2x(n - 2). \quad (2.6)$$

Для фільтра на рисунку 2.4 з трьома натискачами потрібні три операції множення та дві операції додавання. Як відомо, множення може бути затратною операцією з точки зору часу виконання, необхідної потужності та ресурсів, якщо вона реалізована з використанням схеми надвеликого рівня інтеграції (very-large-scale integration або VLSI). Тим не менш, якщо можливо знайти спосіб розкласти великі числа на менші, виконати множення цих менших чисел паралельно і якимось чином об'єднати частковий результат, щоб отримати той самий результат, що і при множенні двох вихідних чисел, тоді є можливим досягнення суттєвої економії на всіх зазначених вище вимогах. Таку схему відносно легко реалізувати в системі залишків: операнди представлені їхніми залишками і перемножуються разом у паралельних каналах для отримання залишків добутку. Тоді зворотне перетворення дає умовний еквівалент продукту. Використовуючи СЗК для реалізації FIR фільтра, вигідно вибрати набір модулів, який складається з великої кількості малих чисел. Тоді залишки, а отже, і ширину множників можна зберегти невеликими значеннями. Розглянуто реалізацію СЗК загального N-tap FIR фільтра (рисунок 2.3).

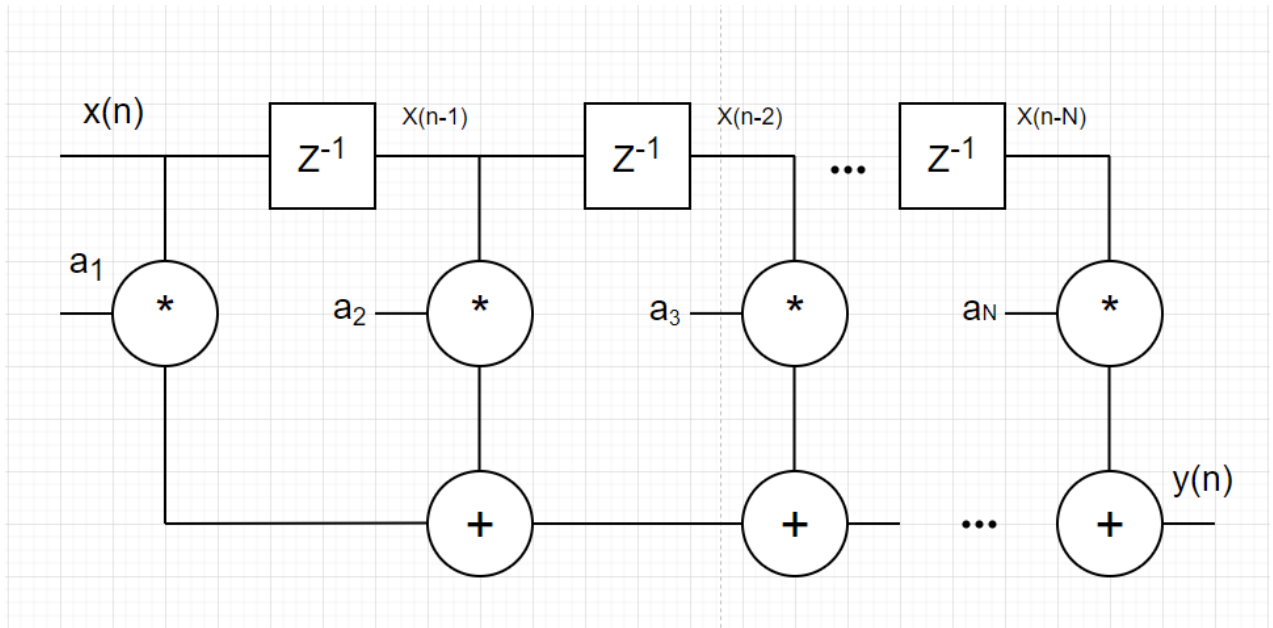


Рисунок 2.3 – FIR фільтр з N натискачами

Загальну архітектуру реалізації на основі СЗК показано на рисунку 2.2. Набір модулів складається з L малих відносно простих чисел m_1, m_2, \dots, m_L , добуток яких достатній для отримання потрібного динамічного діапазону. Реалізація FIR фільтра включає на своєму вході перетворювач двійкового представлення в залишковий, щоб перетворити вхідні дані в еквівалентні залишки. Фільтрація в основному виконується в центральному блоці. Оскільки в наборі залишків є L залишків, L підфільтрів використовуються для обробки відповідних залишків з входу. Суматори та множники можуть бути будь-якого з типів, залежно від того, які характеристики необхідні. Для кращої ефективності субфільтри можуть бути реалізовані як бітово-послідовні блоки.

2.5 Висновки

В розділі описано теорію системи залишкових класів та основні відмінності позиційних систем числення від непозиційних. Описані основні принципи роботи та складові алгоритмів системи залишкових класів. Також, описано процес вибору та застосування модулів, особливості та складності роботи з від'ємними числами, здійснення основним математичних операцій в СЗК.

Визначено переваги та недоліки застосування СЗК для здійснення різних операцій.

Описано застосування системи залишкових класів в процесах обробки цифрових сигналів, реалізації цифрових фільтрів.

Також, розглянути простий приклад де описано процесор СЗК, а саме процес перетворення в ЗСК та процес зворотнього перетворення результату в позиційну систему числення.

3 АЛГОРИТМИ ТА ТЕХНОЛОГІЇ КОДУВАННЯ В СИСТЕМІ ЗАЛИШКОВИХ КЛАСІВ

3.1 Багаторівневий метод кодування даних в СЗК

Особливість даного підходу полягає в представленні даних сенсорів в системі залишкових класів (блок 3.1) та їх перетворення в позиційну систему числення (блок 3.2). На приймальній стороні (на сервері) відбувається зворотне перетворення даних з позиційної системи числення (блок 3.4) в СЗК (блок 3.5) (рисунок 3.1).



Рисунок 3.1 – Алгоритм запропонованого методу кодування

В СЗК будь-яке ціле додатне число представляється в виді набору найменших додатних залишків від ділення цього числа A на фіксовані цілі додатні взаємно прості числа p_1, p_2, \dots, p_n , які називають модулями. Позначимо α_i найменший додатній залишок від ділення A на p_i .

$$\alpha_i = A - \left[\frac{A}{p_i} \right] \cdot p_i, \quad (3.1)$$

де $[\bullet]$ – заокруглення до меншого цілого.

Тоді число A в СЗК запишеться у вигляді:

$$A = \{\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_n\}.$$

Суть багаторівневого кодування полягає в наступному. Для кожного сенсора вибирається взаємно простий модуль згідно умови

$$p_i > a_{i \max}, \quad (3.2)$$

де $a_{i \max}$ – максимальне значення сенсорів.

Дані, що формують сенсори об'єднуються в пакети за формулою (3.3):

$$A_{ij} = \left| \sum_{i=1}^n a_{ij} \cdot B_{ij} \right|_{P_{ij}}, \quad (3.3)$$

де B_{ij} – ортогональні базиси, $P_{ij} = \prod_{i=1}^n p_i$, p_i – взаємно прості числа, n – кількість модулів, $i = \overline{1, n}$, j – номер рівня.

$$B_{ij} = \frac{P_{ij}}{p_i} \cdot \delta_i \equiv 1 \pmod{p_i}, \quad (3.4)$$

де $0 < \delta_i < p_i$ – вага ортогонального елемента.

В блоках A_{12} , A_{i2} , A_{n2} відбувається об'єднання даних (a_1, a_2, \dots, a_n) сенсорів верхнього рівня на основі формули (3.3) (рисунок 3.2 та рисунок 3.3). В блоці A_{13} відбувається об'єднання даних, які поступили з попереднього рівня (блоки A_{12} , A_{i2} , A_{n2}) і т.д. В результаті багаторівневого кодування отримаємо повідомлення в позиційній системі числення в якому представлені виміряні значення сенсорів.

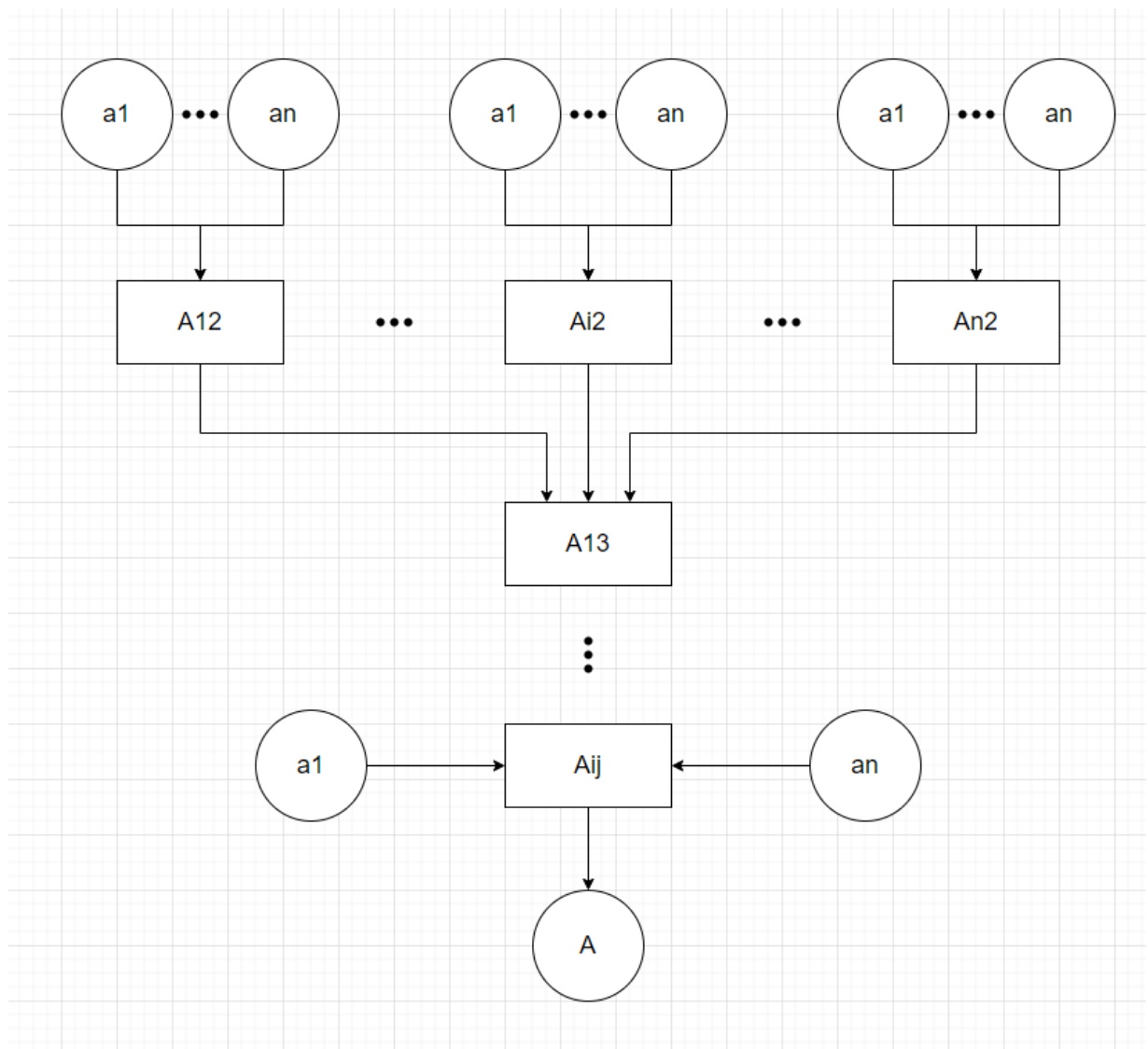


Рисунок 3.2 – Структурна схема багаторівневого кодування даних сенсорів

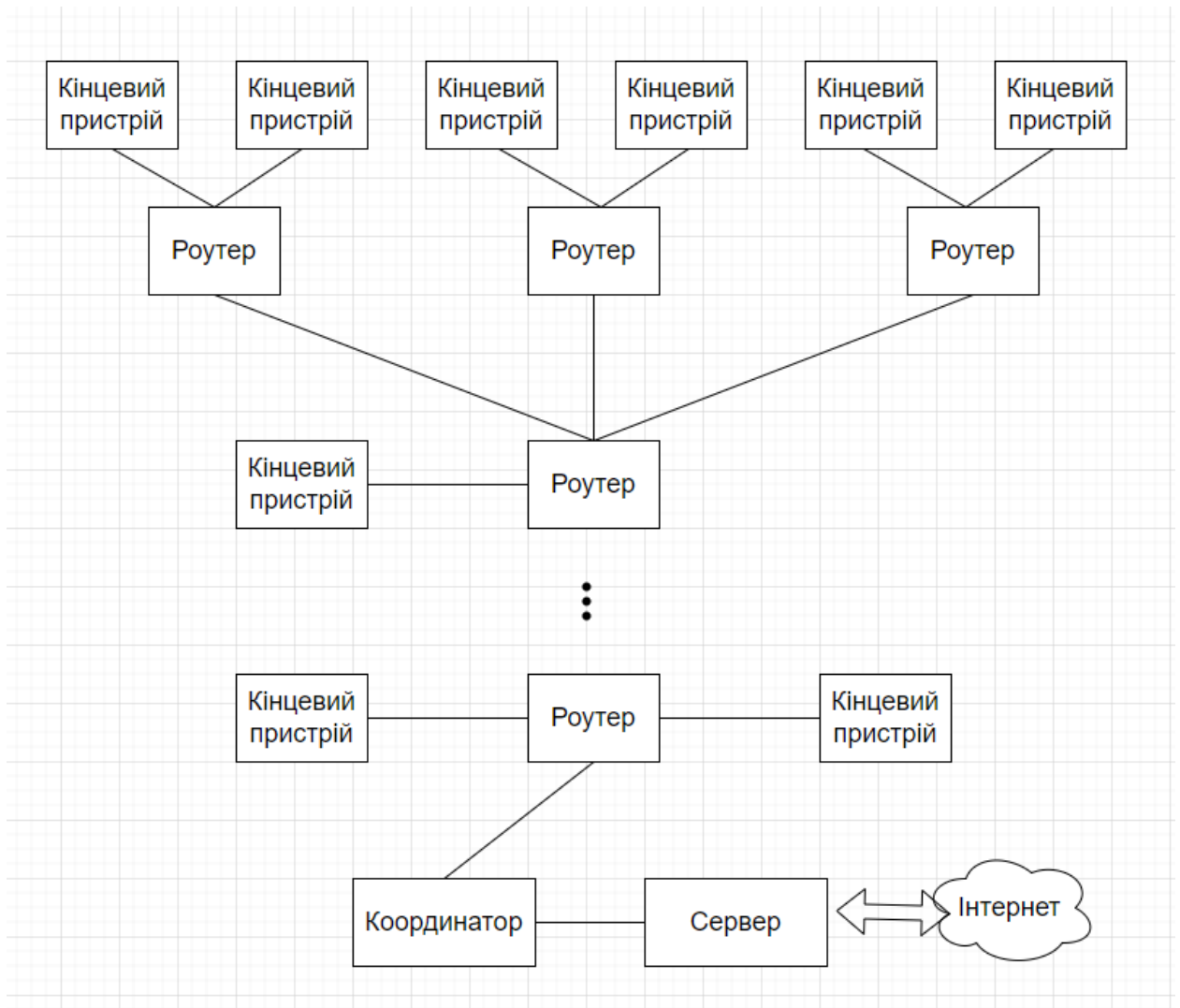


Рисунок 3.2 – Функціональна схема багаторівневого кодування даних сенсорів

Декодування даних відбувається за допомогою операції послідовного отримання залишку по відповідних модулях:

$$A_{ij-1} = A_{ij} \pmod{P_{ij-1}}; \quad (3.5)$$

$$A_{ij-2} = A_{ij-1} \pmod{P_{ij-2}};$$

...

$$A_{i1} = A_{i2} \pmod{P_{i1}}.$$

Розглянемо приклад об'єднання даних з трьох сенсорів A_1 , A_2 , A_3 (рисунок 3.4). Нехай сенсори формують повідомлення в діапазоні: $A_1 \in \{0, 6\}$,

$A_2 \in \{0, 10\}$, $A_3 \in \{0, 12\}$ тоді згідно умови (3.2) вибираємо взаємопрості модулі: $p_1 = 7$, $p_2 = 11$, $p_3 = 13$. Виберемо значення сенсорів $a_1 = 3$, $a_2 = 5$, $a_3 = 9$. Діапазон представлення чисел $P = 7 \cdot 11 \cdot 13 = 1001$. За формулою (3.4) обчислимо базисні числа $B_1 = 715$, $B_2 = 364$, $B_3 = 924$. В результаті розрахунку за формулою (3.3) у вузлі A_{12} отримаємо значення $a_{12} = 296$. Значення a_{12} передається по каналу зв'язку на сервер.

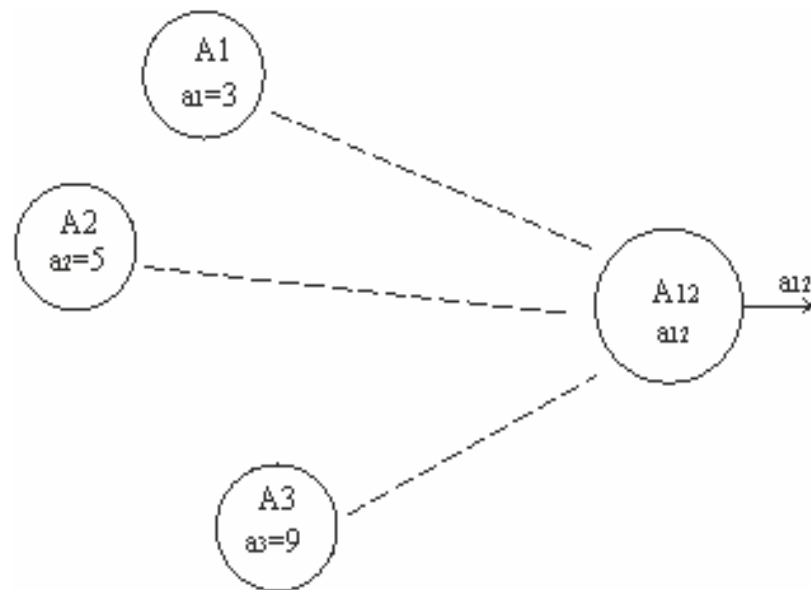


Рисунок 3.4 – Об'єднання даних на основі СЗК

Такий підхід дозволяє об'єднувати дані різної розрядності. При цьому непотрібно додавати адрес сенсора, його роль виконує унікальне значення модуля системи залишкових класів, який присвоюється кожному сенсору.

Зауважимо, що збільшення розміру кадру підвищує пропускну здатність мережі, тільки при надійній роботі мережі (дані не спотворюються і не втрачаються). В іншому випадку збільшення розміру пакету може привести до зменшення пропускну здатності, так як мережі необхідно буде повторно передавати втрачені пакети.

3.2 Виявлення помилок

Безпроводні мережі, за рахунок впливу електромагнітних завад, характеризуються достатньо високою імовірністю бітової помилки (Bit Error Rate, BER) порівняно з кабельними мережами. Для БСМ типовим значенням BER є $10^{-4} \div 10^{-6}$.

В стандарті IEEE 802.15.4 для виявлення помилок використовуються коди CRC (Cyclic Redundancy Check), поліном $G_{16}(x) = x^{16} + x^{12} + x^5 + 1$, які забезпечують виявлення однократних помилок з імовірністю 100 %, а іншу кількість помилок з імовірністю $P = (1 - 2^{-n})$, де n – кількість розрядів контрольної суми.

В БСМ при виявленні помилок відбувається повторна передача пакету даних, що приводить до зменшення корисної пропускну здатності та збільшення енергозатрат.

Підвищити корисну пропускну здатність безпроводної мережі можна застосувавши коректуючі коди. Використання коректуючих кодів зменшує кількість повторних передач пакетів, які спотворені внаслідок дії завад. В роботі розроблений алгоритм виявлення та виправлення помилок на основі коректуючих кодів СЗК з двома контрольними основами.

Розглянемо приклад використання розширеної системи модулів. Для забезпечення можливості виявлення та виправлення помилок в даних, які надходять з блоків A_1 , A_2 , A_3 , розширюємо діапазон їх представлення за допомогою модулів p_4 , p_5 в блоці A_{12} (рисунок 3.5). При цьому отримуємо

$$a_4 = a_{12} \pmod{p_4};$$

$$a_5 = a_{12} \pmod{p_5}.$$

Після цього, значення a_1, \dots, a_5 об'єднуємо за формулою (3.3).

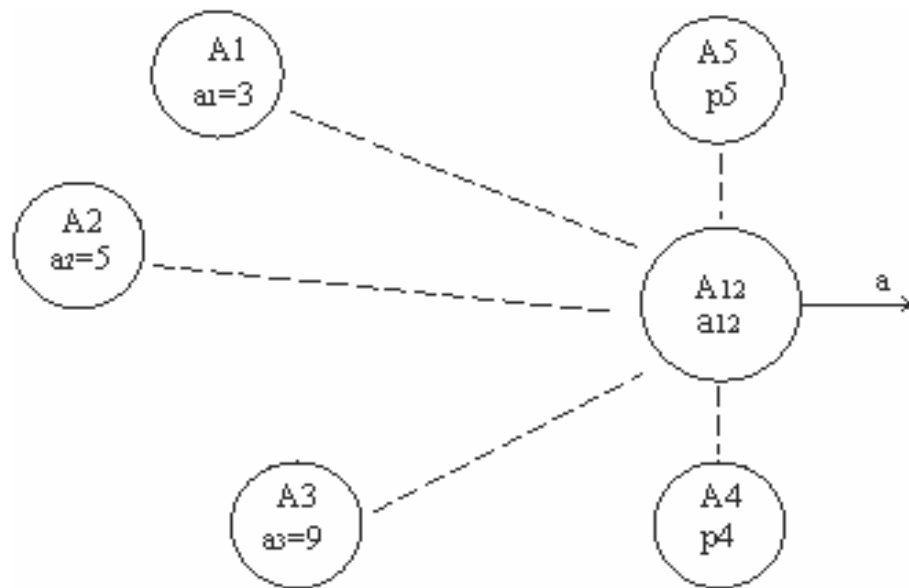


Рисунок 3.5 – Об’єднання даних з додатковими коректуючими модулями

На приймальній стороні (сервері) здійснюється перевірка отриманих даних на наявність помилок, виправлення помилок за алгоритмом. Для декодування даних декодер повинен знати модулі, які використовувалися при кодуванні. Тому поле даних протоколу крім закодованих даних (Data) містить номери модулів $N_{M1} \dots N_{Mn}$ (рисунок 3.6). Номер модуля також ідентифікує сенсор з якого поступають дані.

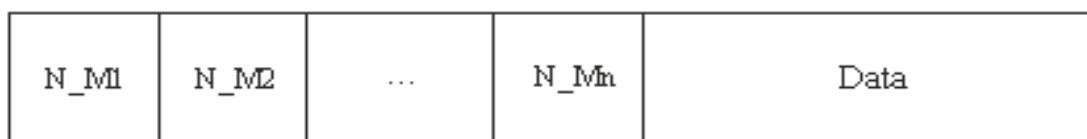


Рисунок 3.6 – Структура поля даних протоколу передачі

В результаті послідовного ділення прийнятих даних на взаємно прості модулі СЗК отримаємо вихідні дані сенсорів.

Як відомо, система числення в залишкових класах відкриває можливість використання єдиного завадостійкого коду для боротьби з помилками, що виникають при передачі інформації по каналах зв’язку і при її обробці в цифрових інформаційних системах. Універсальність кодів СЗК пояснюється не тільки їх

високими коректуючими можливостями, арифметичністю і можливістю боротьби з пакетами помилок, але і їх адаптивною зміною коректуючих властивостей без зміни способу кодування.

3.3 Експериментальні дослідження

Розрахуємо корисну пропускну здатність при використанні коректуючих кодів СЗК з двома контрольними модулями. Для кадрів максимальної довжини ($104 - 2 = 102$ байт, контрольні модулі займають 2 байти) корисна пропускну здатність $C_p = m_2 \cdot (d_2 - 2) = 200,7$ Кбіт/с.

Отже, при використанні коректуючих кодів системи залишкових класів з двома контрольними модулями, які забезпечують виправлення помилок по будь-якому модулю системи, корисна пропускну здатність становить пibližно 200 кбіт/с.

Розрахуємо корисну пропускну здатність при різній імовірності помилки:

$$C_p = \frac{m \cdot L}{1 + P_b \cdot L},$$

де m – кількість кадрів в секунду; L – довжина поля даних, P_b – імовірність помилки.

На рисунку 3.7 приведені результати розрахунку корисної пропускну здатності при різній імовірності бітової помилки.

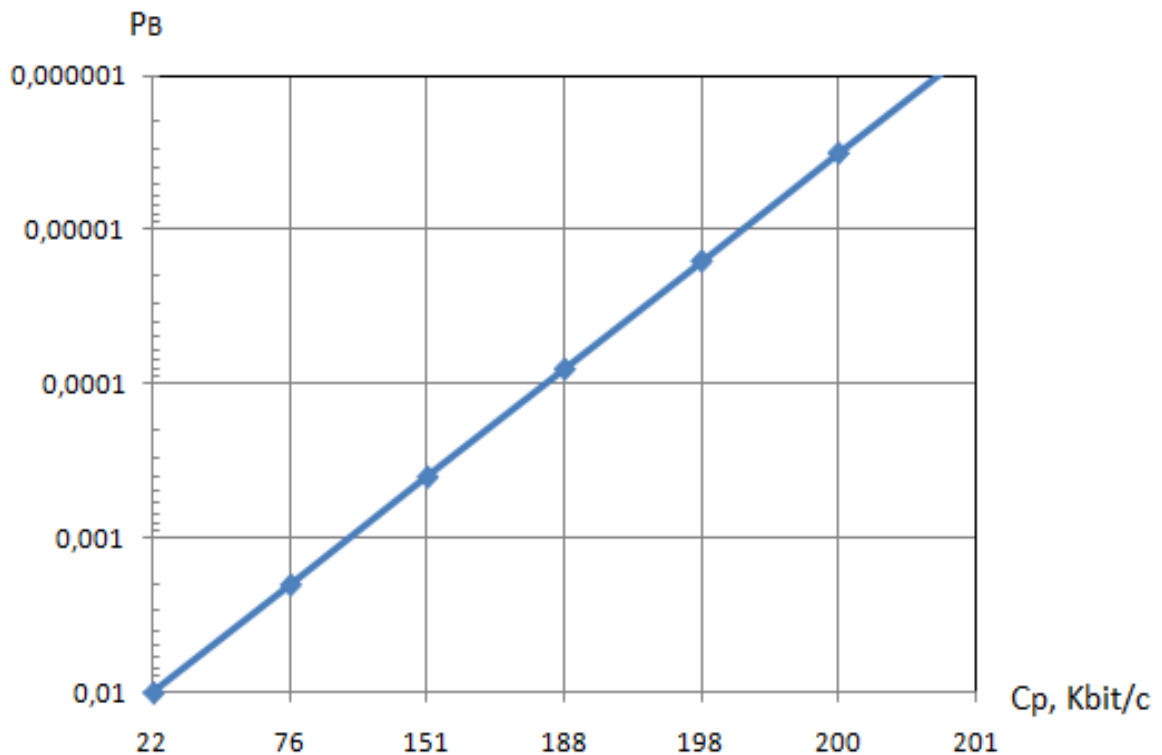


Рисунок 3.7 – Залежність корисної пропускної здатності від імовірності появи помилки для максимального розміру блоку даних (104 байти)

Отже, при імовірності помилки в каналі більше 10^{-5} є доцільним використання коректуючих кодів СЗК (рисунок 2.6). Корисна пропускна здатність, зменшується в два рази при імовірності помилки 10^{-3} .

Таким чином запропонований метод кодування даних сенсорів дозволяє зменшити обсяг службових даних, і тим самим суттєво підвищити корисну пропускну здатність протоколів передачі даних БСМ. Крім того, кодування даних на основі перетворення системи залишкових класів забезпечує додаткову можливість виявлення та виправлення помилок.

Запропоновано багаторівневий метод кодування даних сенсорів, який підвищує корисну пропускну здатність протоколів передачі даних WSN в 3 – 5 разів за рахунок об'єднання даних сенсорів різної розрядності та зменшує кількість передач.

Проведені дослідження корисної пропускної здатності в залежності від імовірності бітової помилки, при використанні коректуючих кодів СЗК з двома

контрольними модулями, показали що застосування коректуючих кодів є доцільним при імовірності бітової помилки більше 10^{-5} .

3.4 Запропонований метод кодування даних сенсорів на основі багаторівневої системи залишкових класів

Суть пропонованого способу кодування полягає у наступному [29, 30]: вибираємо систему взаємно простих чисел (модулів) значення яких має бути більшим за максимальне значення даних сенсора. Відповідно, дані сенсорів x_1 - x_8 будемо вважати залишками по вибраній системі модулів. Наступним кроком здійснюємо перетворення залишків x_1 - x_8 у позиційну (десяткову або двійкову) систему числення. Код для проведення відповідних обчислень на мові Python має вигляд:

$$\begin{aligned} N1 &= x_2 + p_2 * ((k_0 * (x_1 - x_2)) \% p_1) \\ N2 &= x_4 + p_4 * ((k_1 * (x_3 - x_4)) \% p_3) \\ X01 &= N2 + p_{34} * ((k_2 * (N1 - N2)) \% (p_{12})) \\ N3 &= x_6 + p_6 * ((k_3 * (x_5 - x_6)) \% p_5) \\ N4 &= x_8 + p_8 * ((k_4 * (x_7 - x_8)) \% p_7) \\ X02 &= N4 + p_{78} * ((k_5 * (N3 - N4)) \% p_{56}) \\ X &= X02 + p_{5_8} * ((k_6 * (X01 - X02)) \% p_{1_4}), \end{aligned}$$

де $p_{12} = p_1 * p_2$; $p_{34} = p_3 * p_4$; $p_{56} = p_5 * p_6$; $p_{78} = p_7 * p_8$; $p_{1_4} = p_1 * p_2 * p_3 * p_4$; $p_{5_8} = p_5 * p_6 * p_7 * p_8$; $k_0 * p_2 \% p_1 == 1$; $k_1 * p_4 \% p_3 == 1$; $k_2 * p_{34} \% p_{12} == 1$; $k_3 * p_6 \% p_5 == 1$; $k_4 * p_8 \% p_7 == 1$; $k_5 * p_{78} \% p_{56} == 1$; $k_6 * p_{5_8} \% p_{1_4} == 1$.

Структурна схема перетворення даних з СЗК у десяткову на основі нової китайської теореми про залишки приведена на рисунку 3.7 [29].

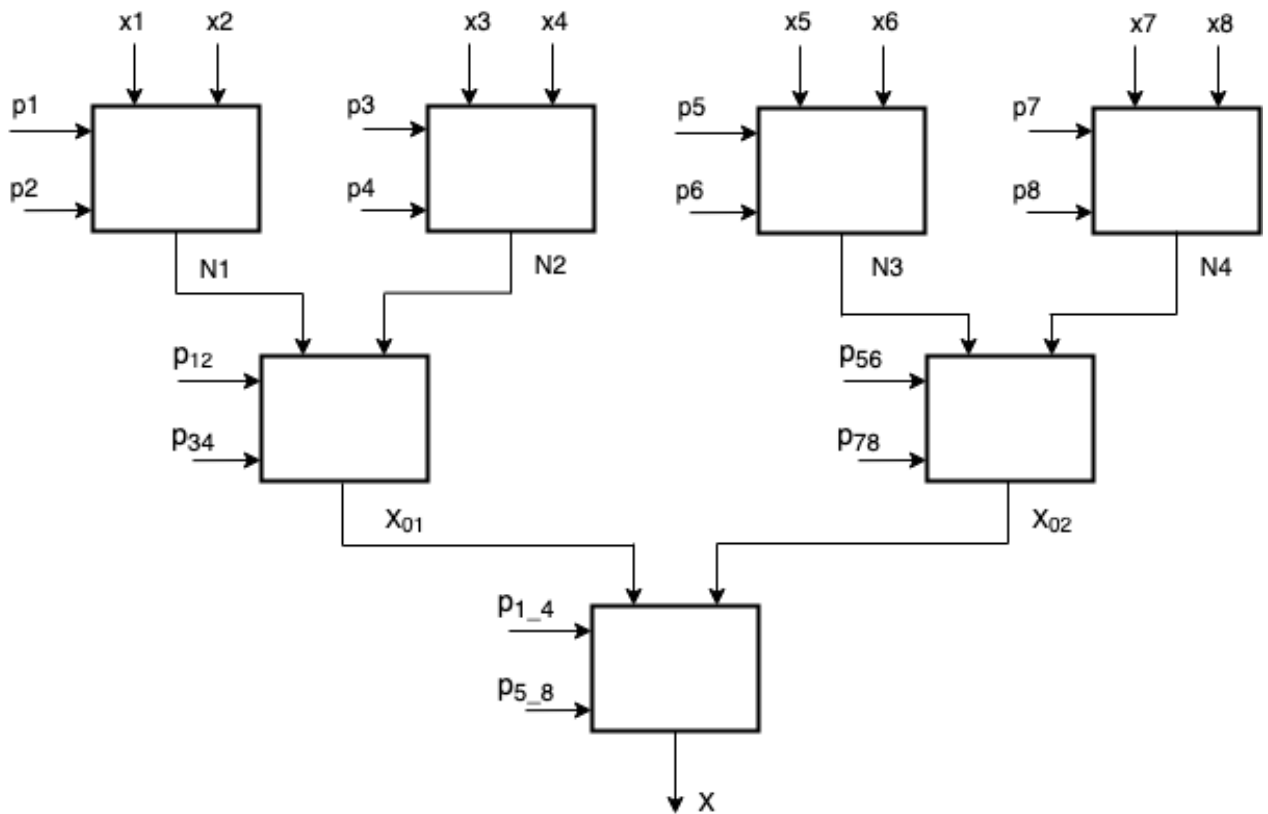


Рисунок 3.7 – Структурна схема кодування даних сенсорів

В результаті обчислення отримуємо число у десятковій або двійковій системі числення, яке відображає значення всіх сенсорів. В даному випадку ідентифікатором сенсора виступає модуль СЗК.

Для кодування даних восьми сенсорів (розрядність даних -10 біт) в якості модулів СЗК вибираємо наступні прості числа: $p_1=1031$; $p_2=1033$; $p_3=1039$; $p_4=1049$; $p_5=1051$; $p_6=1061$; $p_7=1063$; $p_8=1069$.

Відновлення даних відбувається за формулою:

$$x_i = X(\text{mod } p_i). \quad (7)$$

Розглянемо приклад кодування. Нехай сенсори генерують значення: $x_1=488$; $x_2=956$; $x_3=969$; $x_4=28$; $x_5=62$; $x_6=964$; $x_7=363$; $x_8=825$. Коефіцієнти k_i для вказаних модулів: $k_0=516$; $k_1=104$; $k_2=213123$; $k_3=946$; $k_4=886$; $k_5=502052$; $k_6=1123305632393$.

За формулою (6) знаходимо значення $X=1032780252932$.

Використовуючи формулу (7) знаходимо значення сенсорів:
 $x_1=1032780252932 \bmod p_1=516$, $x_2=1032780252932 \bmod p_2=956$ і т.д.

Виконавши декодування результатом будуть значення які відповідають даним сенсорів, що підтверджує коректну роботу схеми кодування.

3.5 Висновки

Наведено метод багаторівневого кодування даних сенсорів який базується на системі залишкових класів та надає можливість об'єднати дані різних сенсорів у одне значення для підвищення ефективності використання ресурсів передачі даних мережі.

Наведено структурну та функціональну схеми багаторівневого кодування даних сенсорів.

Здійснено експериментальні дослідження з наведенням прикладу кодування даних сенсорів, результатами якого є теоретичне підтвердження збільшення корисної пропускну здатності протоколів передачі даних БСМ шляхом зменшення передачі необхідних службових даних за рахунок зменшення кількості передач.

4 РЕАЛІЗАЦІЯ СИСТЕМИ КОДУВАННЯ, ПЕРЕДАЧІ ТА ОБРОБКИ ЗАКОДОВАНИХ ДАНИХ СЕНСОРІВ НА ОСНОВІ БАГАТОРІВНЕВОЇ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ

4.1 Використання інструменту програмування Node-RED для реалізації системи кодування та передачі даних

З метою реалізації системи кодування та передачі даних сенсорів використано Node-RED для створення симуляції системи яка складається з чотирьох сенсорів, отримання їх показників з подальшим кодуванням на основі багаторівневої системи залишкових класів та передача до системи збору даних.

Node-RED – це інструмент візуального програмування, що розроблений компанією IBM. Інструмент призначений для поєднання різноманітних веб-сервісів та API. Node-RED надає можливість роботи з редактором потоків на основі веб-браузера, який можна використовувати для створення функцій JavaScript. Елементи додатків можна зберігати або використовувати для повторного використання. Середовище виконання побудовано на Node.js. Потоки, створені в Node-RED, зберігаються за допомогою JSON [31].

Схема модельованої системи в Node-RED показана на рисунку 4.1.

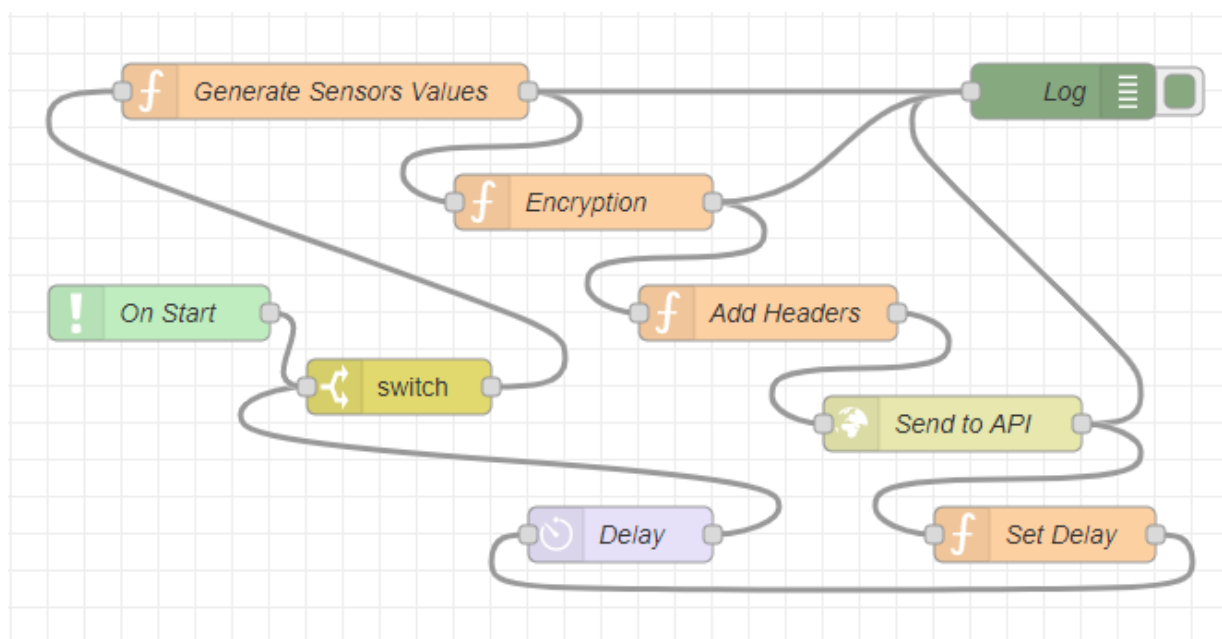


Рисунок 4.1 – Схема модельованої системи в Node-RED

Об'єкт On Start являється початковим елементом системи. Його функцією є запуск процесів генерації, кодування та відправки даних. Даний об'єкт це елемент типу complete [32] в Node-RED. Налаштування елемента показано на рисунку 4.2.

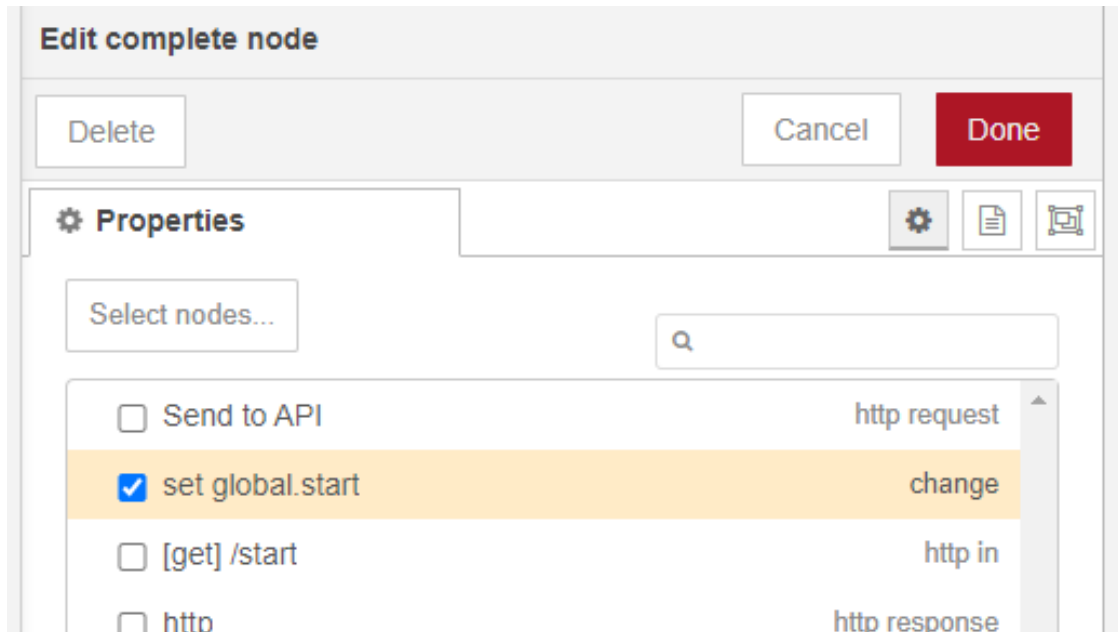


Рисунок 4.2 – Налаштування об'єкта On Start в Node-RED

On Start спрацьовує після встановлення глобальної логічної змінної start. Змінна start встановлюється за допомогою комбінації об'єктів http in [33] та change [34] (рисунок 4.3).

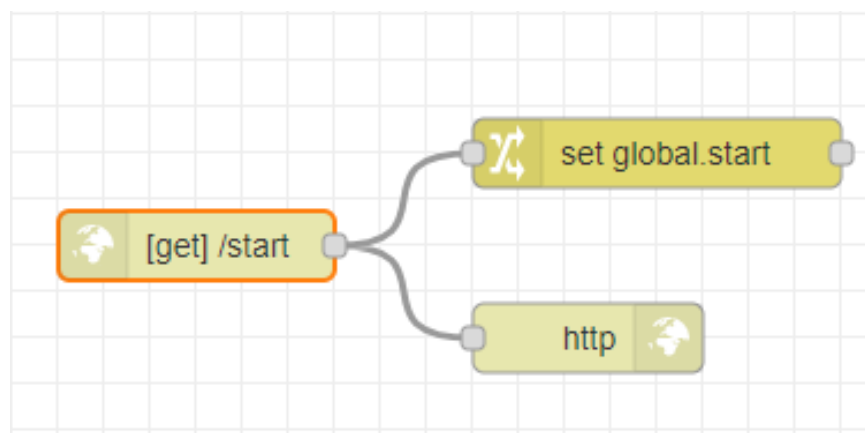


Рисунок 4.3 – Об'єкти http in, change та http response для можливості зовнішнього запуску системи за допомогою http запиту

Перший об'єкт представляє собою http endpoint по url `http://localhost:1880/start`. Налаштування об'єкта «http in» показано на рисунку 4.4.

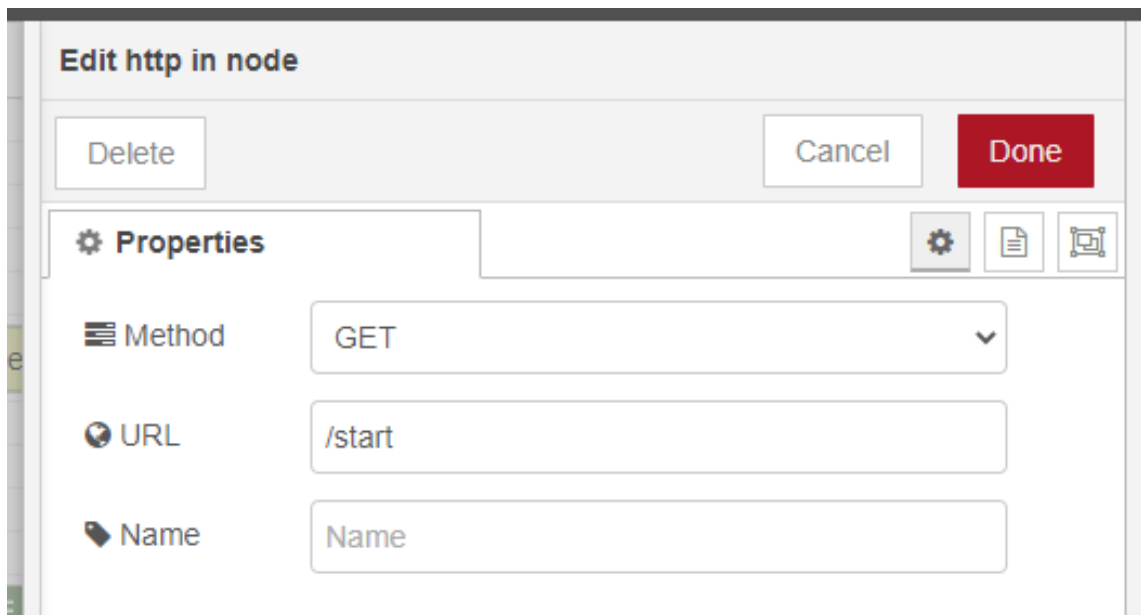


Рисунок 4.4 – Налаштування об'єкта http in в Node-RED

При зверення на url адресі `/start` глобальна логічна змінна `start` приймає значення `true`. Це реалізовано за допомогою другого об'єкта – об'єкта `change`, налаштування якого показано на рисунку 4.5.

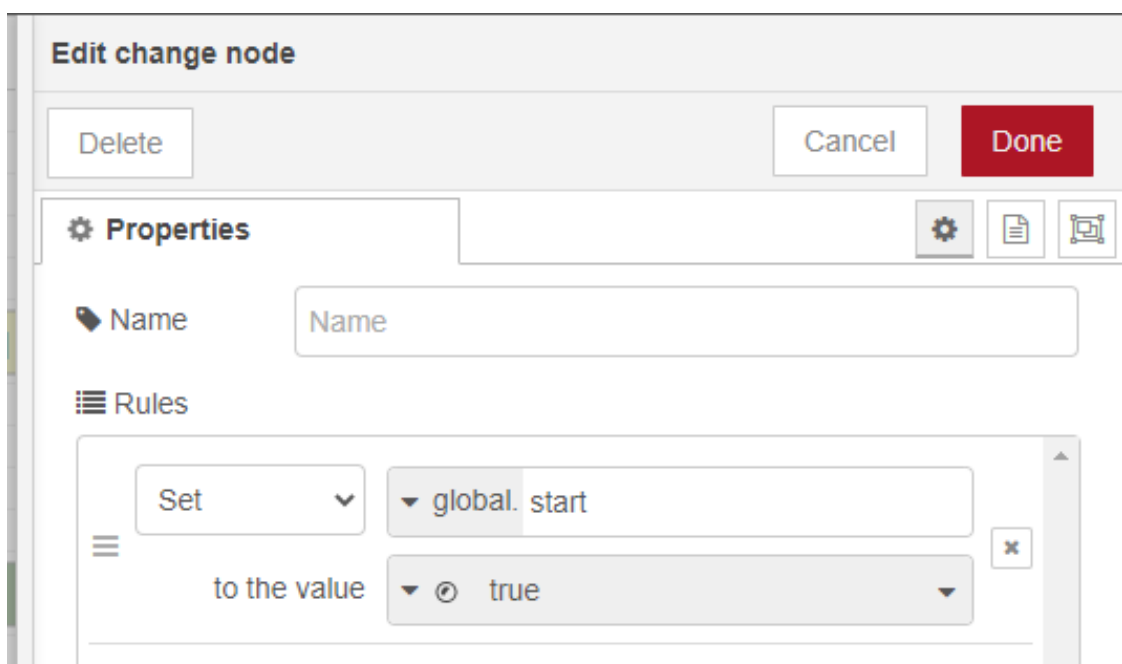


Рисунок 4.5 – Налаштування об'єкта change в Node-RED

Зміна значення глобальної змінної `start` відбувається за допомогою аналогічної комбнації об'єктів `http in` та `change`, відмінністю є тільки те, що `http` реагує на GET запит на `http` адресу `/stop`, а об'єкт `change` встановлює глобальній змінній `start` значення `false`. Це необхідно для припинення роботи системи.

Наступним об'єктом по порядку виклику на рисунку 4.1 є об'єкт `switch` [35]. Функцією цього об'єкта є перевірка значення глобальної логічної змінної `start`. Об'єкт `switch` передасть наступному об'єкту в разі, коли змінна має позитивне значення. При негативному значенні системи припинить свою роботу. Налаштування об'єкта `switch` показані на рисунку 4.6.

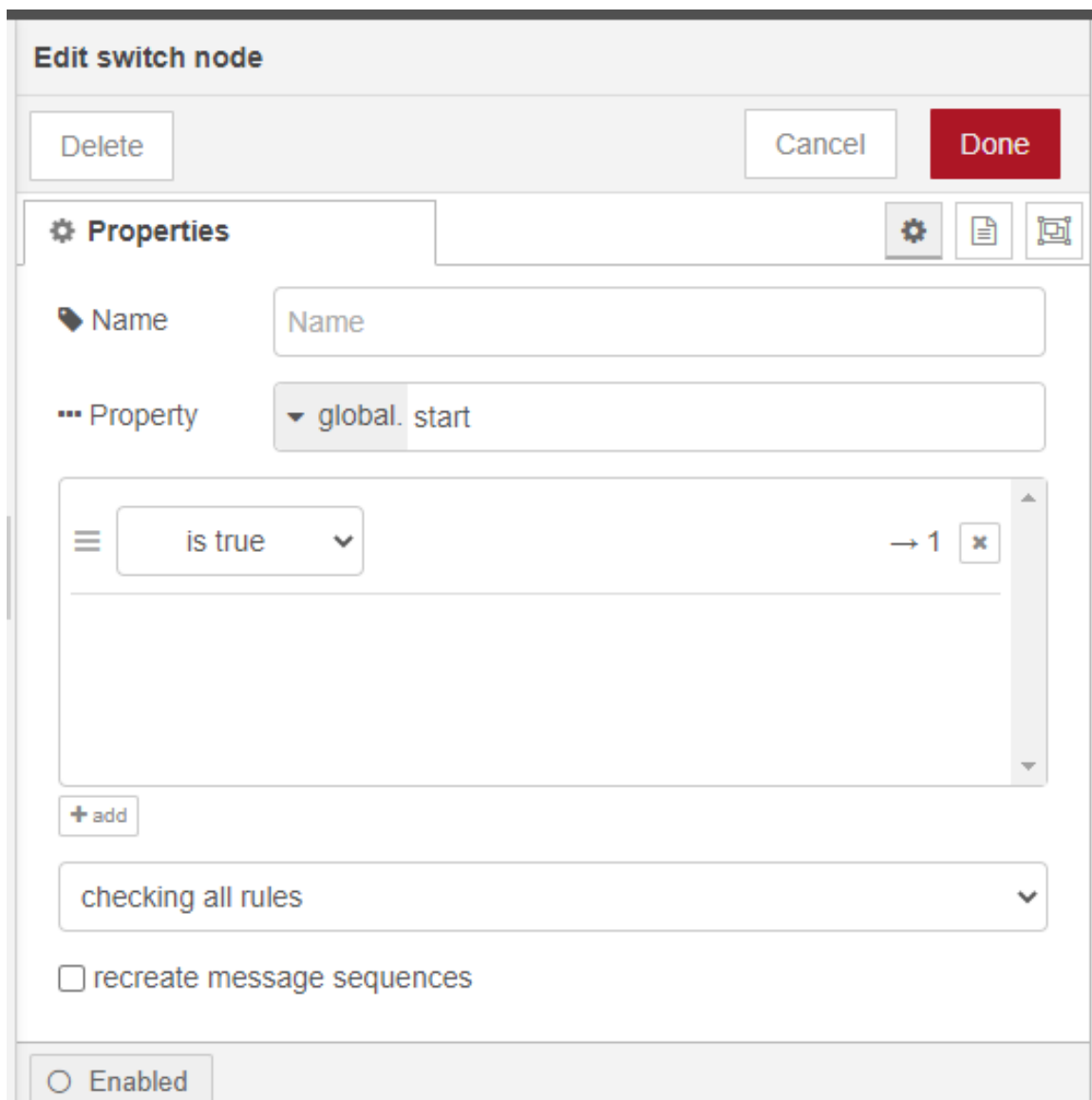


Рисунок 4.6 – Налаштування об'єкта `switch` в Node-RED

Об'єкт `Generate Sensors Values` призначений для генерації даних сенсорів для передачі для шифрування. Деталі об'єкта описані в розділі 4.1.1.

Об'єкт `Encryption` призначений для шифрування даних сенсорів, отриманих на виході з елемента `Generate Sensors Values`. Деталі об'єкта описані в розділі 4.1.2. Після кодування даних формується JSON об'єкт, який містить поле `value`, в якому зберігається закодоване значення.

Об'єкт `Add Headers` призначений для створення HTTP заголовків, необхідних для здійснення HTTP запиту до системи збору даних. Налаштування даного об'єкта в Node-RED показано на рисунку 4.7.

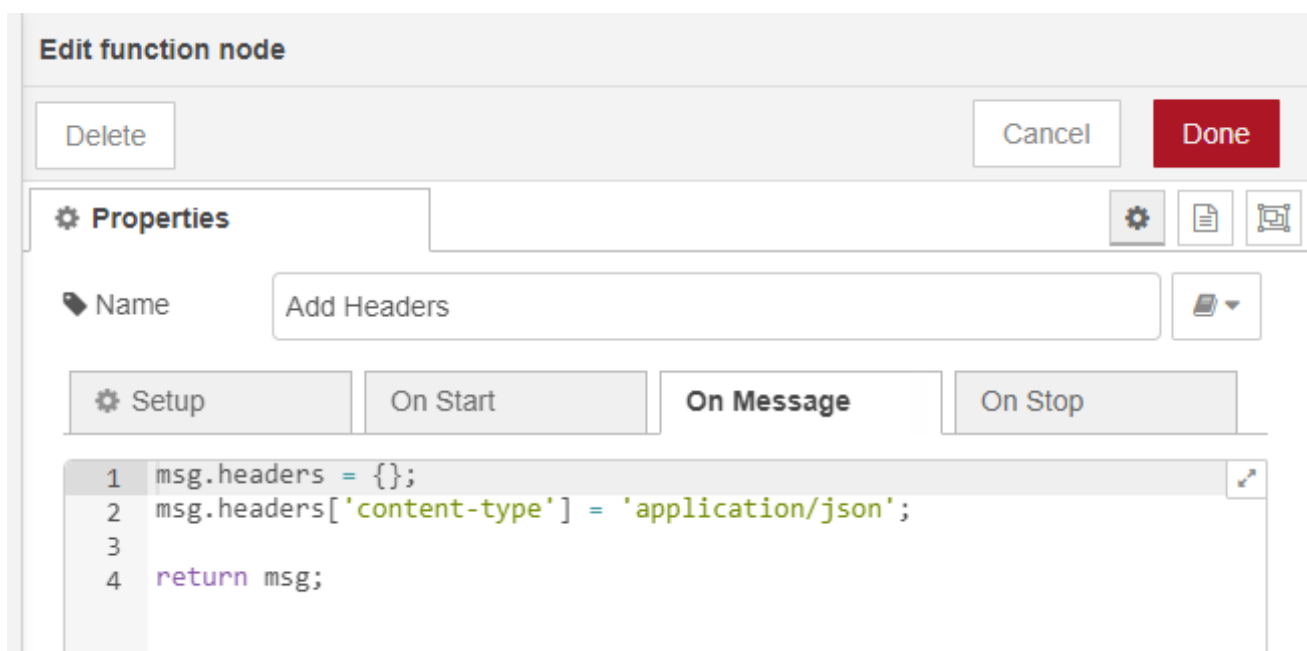


Рисунок 4.7 – Налаштування об'єкта `Add Headers` в Node-RED

Об'єкт `Send to API` здійснює HTTP POST запит для відправки закодованих значень сенсорів на API. Деталі об'єкта описані в розділі 4.1.3.

Об'єкт `Set Delay` використовується для отримання значення глобальної змінної `delay`, яка використовується для наступного об'єкта в ланцюгу викликів – об'єкта `Delay`. Налаштування об'єкта `Set Delay` показано на рисунку 4.8.

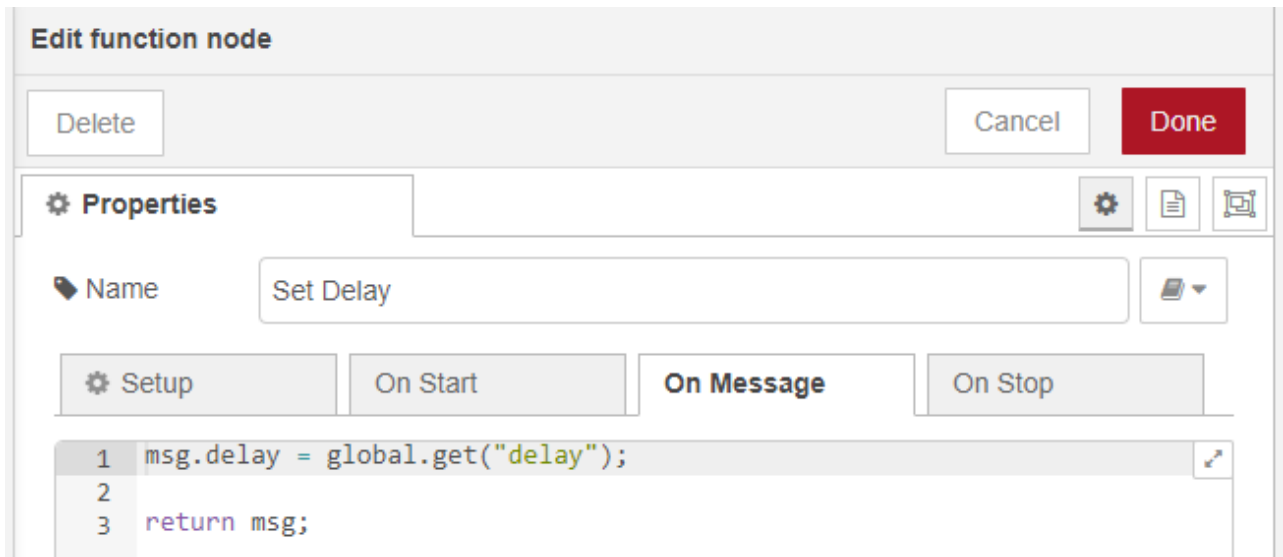


Рисунок 4.8 – Налаштування об'єкта Set Delay в Node-RED

Глобальна змінна `delay` встановлюється за допомогою комбінації об'єктів `http in` та `change` (рисунок 4.9).

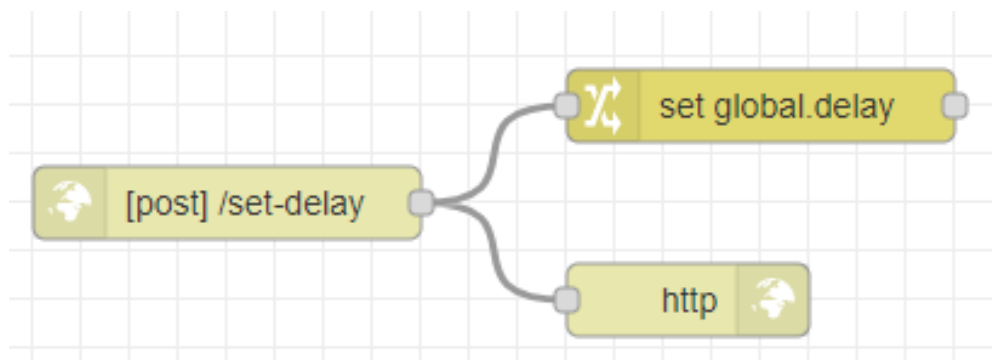


Рисунок 4.9 – Об'єкти `http in`, `change` та `http response`

Перший об'єкт на рисунку 4.9 є `http endpoint` що реагує на `POST` запити по `url` адресі `/set-delay` (рисунок 4.10).

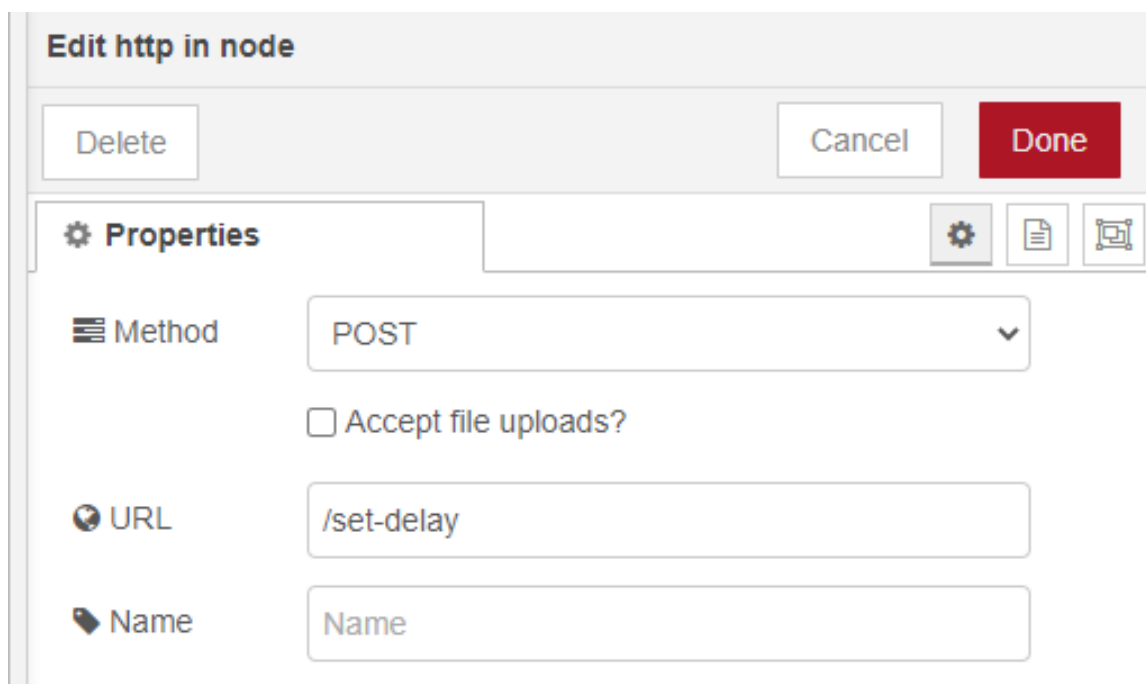


Рисунок 4.10 – Налаштування об'єкта http in в Node-RED

Endpoint приймає числове значення delay, яке записується в глобальну змінну delay за допомогою об'єкта change, налаштування якого показані на рисунку 4.11. Таким чином, забезпечується можливість зовнішньої зміни інтервалу генерації та надсилання закодованих даних сенсорів до центру обробки даних.

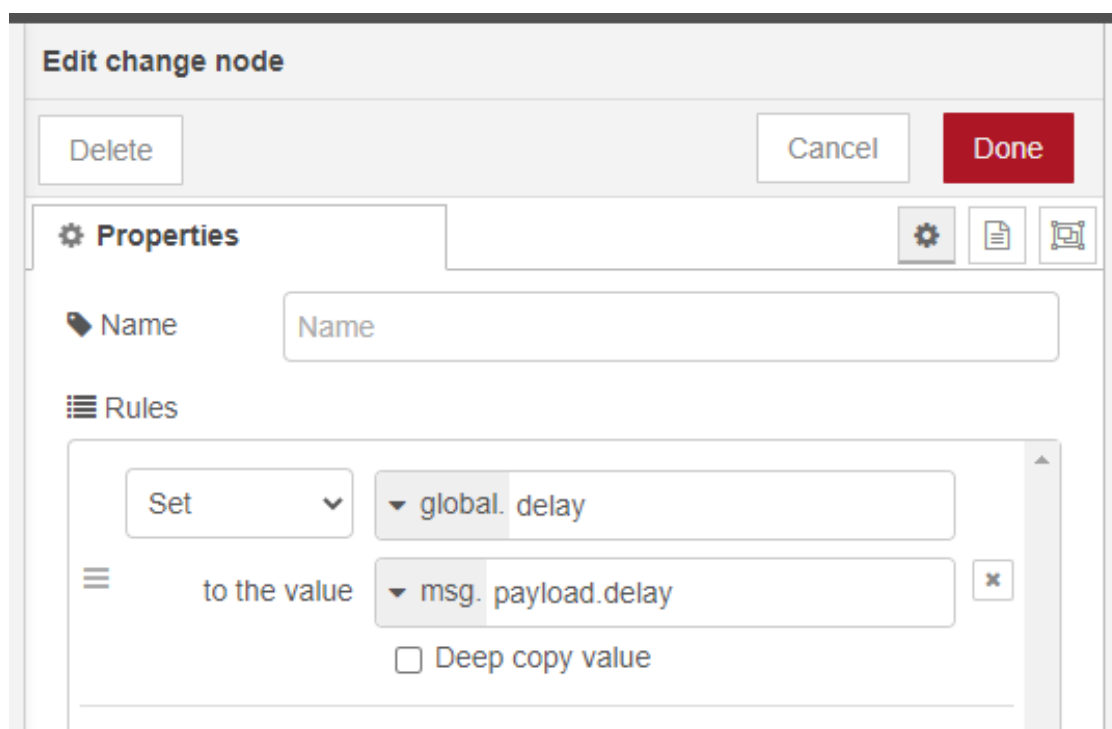


Рисунок 4.11 – Налаштування об'єкта change в Node-RED

Наступним об'єктом після об'єкта Set Delay на рисунку 4.1 є б'єкт Delay, який призначений для встановлення інтервалу виконання процесів генерації, кодування та передачі даних сенсорів з інтервалом, значення якого об'єкт отримує на вході. У разі, коли при вході об'єкт не отримує значення, встановлюється інтервал, значення якого дорівнює 5 секунд. Налаштування об'єкта показано на рисунку 4.12.

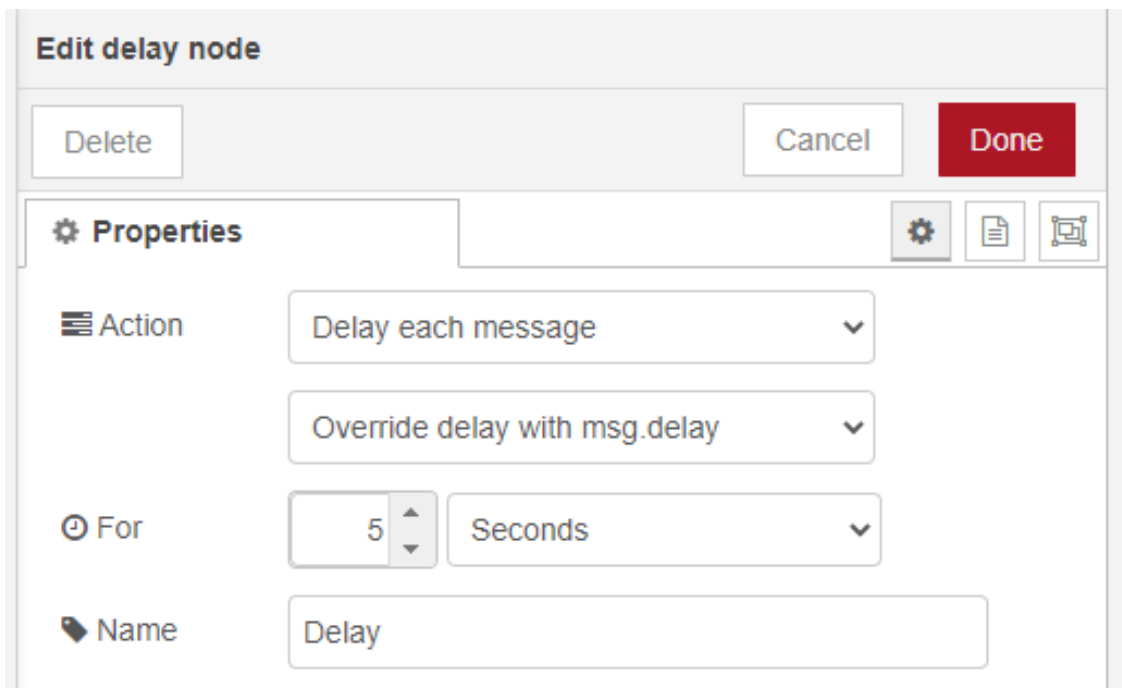


Рисунок 4.12 – Налаштування об'єкта Delay в Node-RED.

4.1.1 Симуляція генерації даних сенсорів в модельованій системі

У модельованій системі кожний з чотирьох сенсорів може згенерувати показник в межах від 1 до 1000.

У модельованій системі генерація даних сенсорів представлена об'єктом function. Функція отримує кількість сенсорів, за замовчуванням це 4, та для кожного сенсора генерує значення від 1 до 1000. З генерованих значень формується вихідний масив. Блоксхема коду генерації представлена на рисунку 4.13.

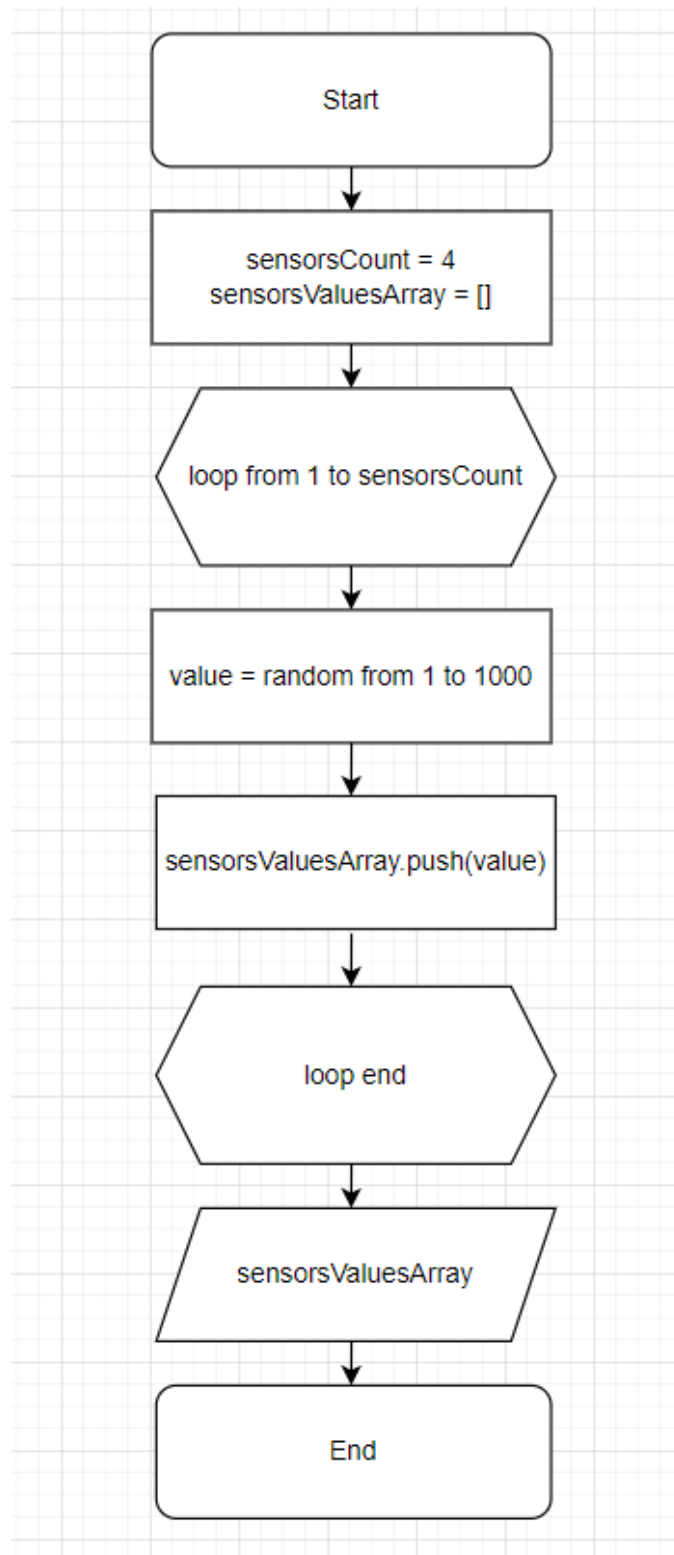


Рисунок 4.13 – Блок-схема коду генерації даних сенсорів

Для виконання генерації даних сенсорів використовується об'єкт `function`, який має назву `Generate Sensors Values`. Налаштування об'єкта `Generate Sensors Values` для генерації даних сенсорів в Node-RED показані на рисунку 4.14.

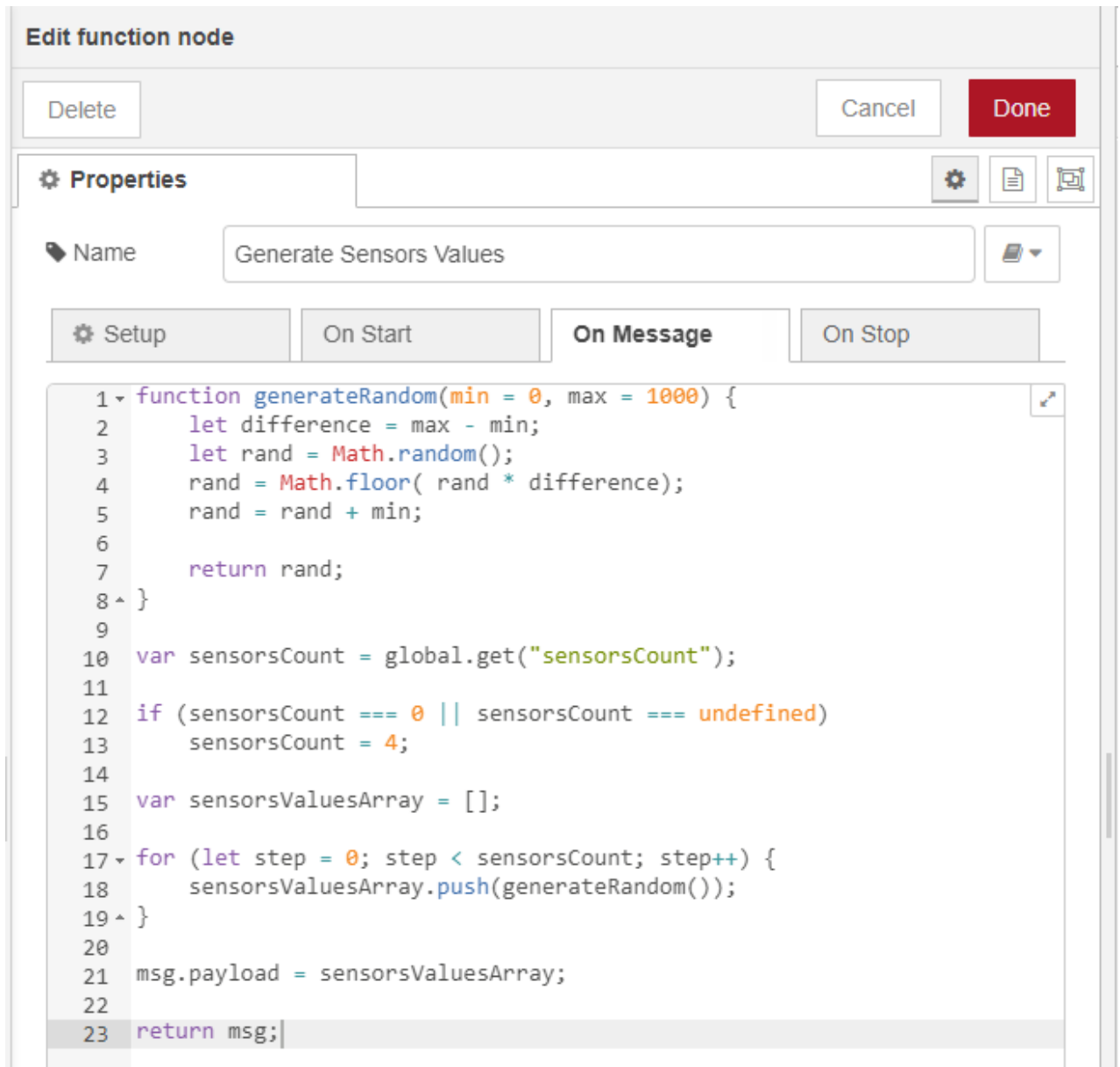


Рисунок 4.14 – Налаштування об’єкта Generate Sensors Values в Node-RED для генерації даних сенсорів

4.1.2 Шифрування даних сенсорів в модельованій системі

Шифрування здійснюється за формулами які описані в розділі 3. Для шифрування в Node-RED використано об’єкт function, який приймає масив генерованих значень сенсорів, набір коефіцієнтів k_0 , k_1 та k_2 , вибрані модулі p_1 , p_2 , p_3 та p_4 . Вихідними даними функції є єдине значення яке представляє показники чотирьох сенсорів. Блоксхема коду процесу кодування представлена на рисунку 4.15.

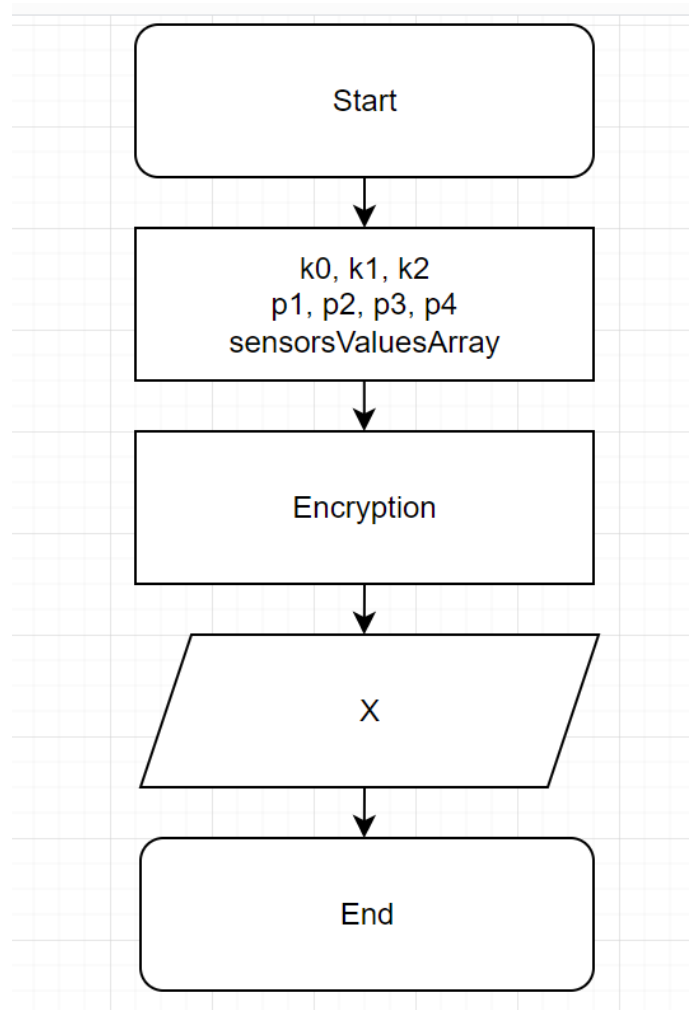


Рисунок 4.15 – Блок-схема коду шифрування даних сенсорів

Для виконання шифрування використовується об'єкт `function`, який має назву `Encryption`. Налаштування об'єкта `function` для шифрування даних сенсорів в `Node-RED` показані на рисунку 4.16.

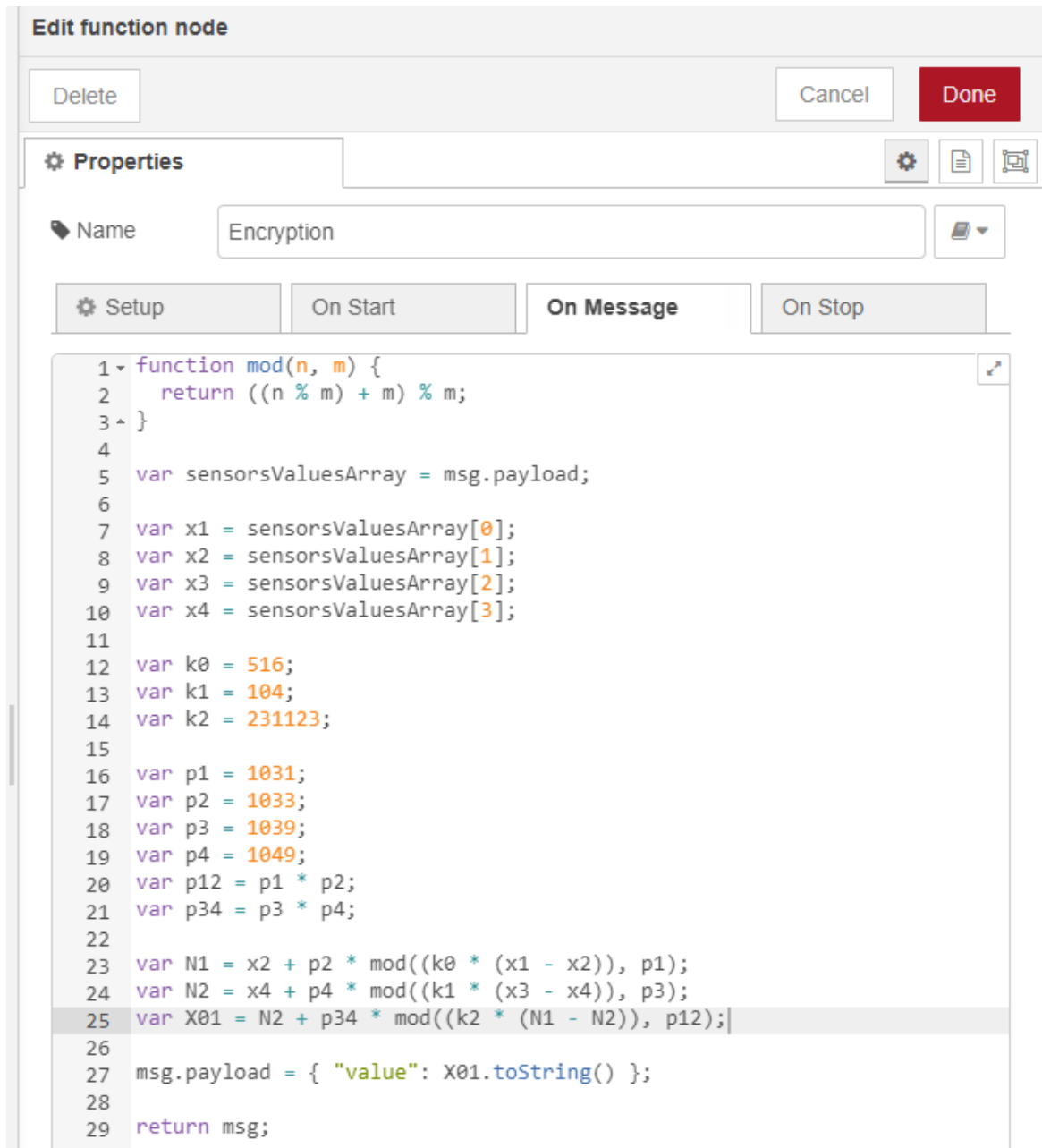


Рисунок 4.16 – Налаштування об’єкта Encryption в Node-RED для шифрування даних сенсорів

4.1.3 Передача даних до системи збору даних в модельованій системі

Для передачі закодованих даних до системи збору, використано об’єкт `http-request`. Даний об’єкт призначений для здійснення HTTP запитів. В даному випадку здійснюється POST запит на Web API, описану в розділі 4.2. В якості даних, які передаються, виступає закодоване значення показників сенсорів. Налаштування об’єкта `http-request` показано на рисунку 4.17.

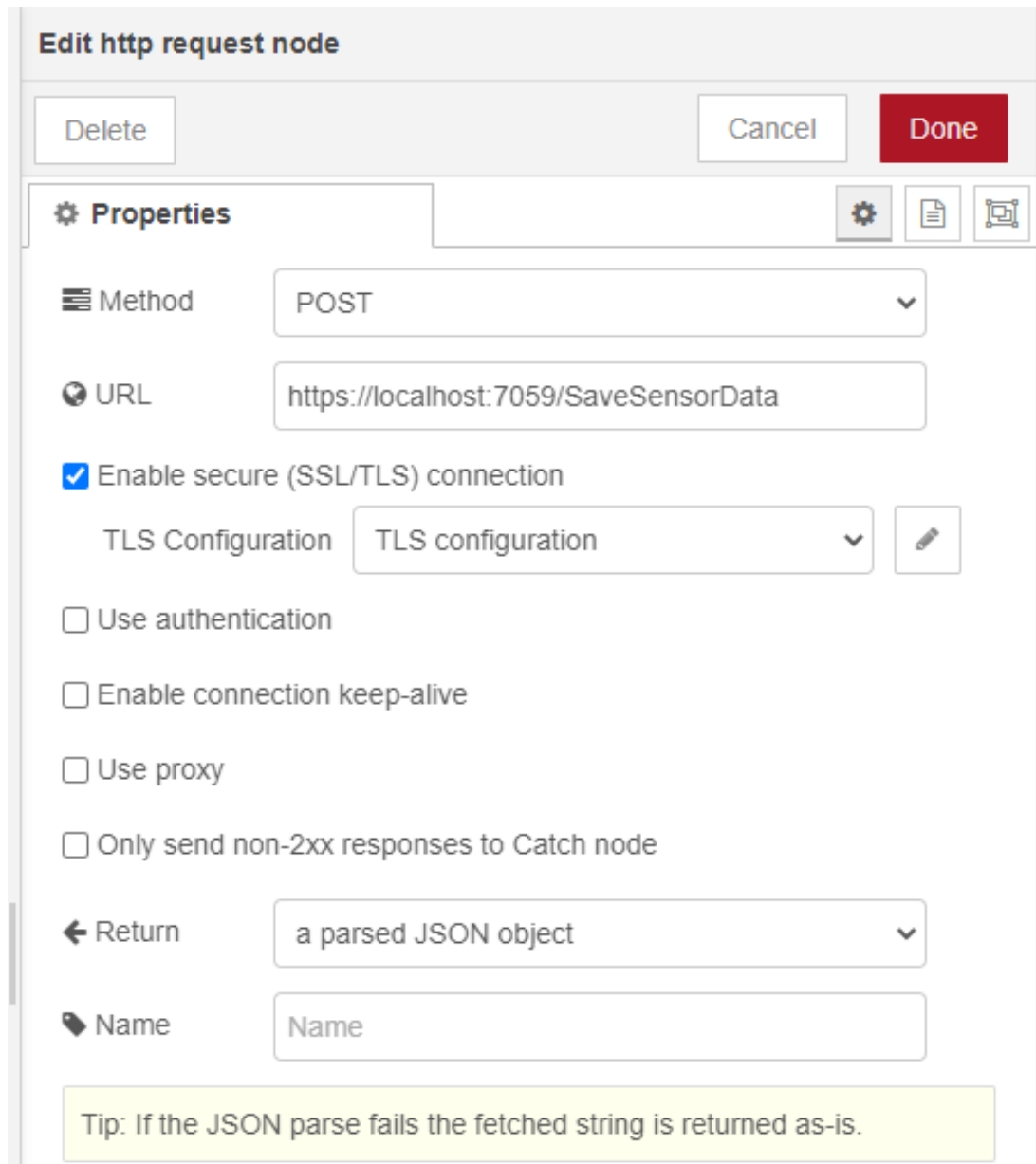


Рисунок 4.17 – Налаштування об’єкту http-request в Node-RED для передачі даних сенсорів до системи збору даних

4.2 Реалізація Web API для системи збору та обробки даних за допомогою ASP.NET Core

ASP.NET Core [36] – це веб-фреймворк з відкритим вихідним кодом і наступник ASP.NET, розроблений компанією Microsoft. Це модульний фреймворк що працює як на базі .NET Framework на Windows, так і на базі .NET.

В рамках реалізації системи збору та обробки даних, було створено два проекти .csproj Core та WebApp.

4.2.1 Опис проекту Core

Проект Core це бібліотека класів C# (Class Library) на базі .NET фреймворку версії 6.0. Проект містить необхідну бізнес-логіку, а саме засоби для збереження, отримання даних та декодування зашифрованих даних сенсорів отриманих з системи яка побудована з використанням Node-RED (розділ 4.1).

Проект має наступну структуру:

1. Abstraction. Містить інтерфейси;
2. Models. Містить моделі (класи) бізнес-об'єктів предметної області;
3. Services. Містить реалізації інтерфейсів.

Діаграма класів, які використовуються в проекті Core показана на рисунку 4.18:



Рисунок 4.18 – Діаграма класів проекту Core

Повна структура проекту Core показана на рисунку 4.19:

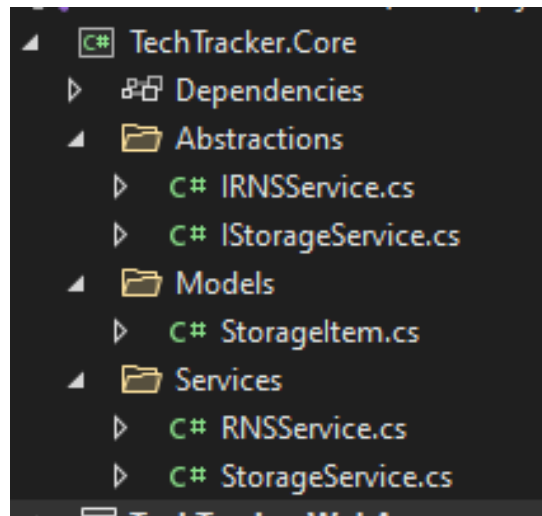


Рисунок 4.19 – Структура проекту Core в Visual Studio

Клас `StorageItem` містить поля `EncodedValue` та `DecodedValues`. Клас представляє сутність декодованих значень. Поле `EncodedValue` містить закодоване значення тоді як поле `DecodedValues` містить набір декодованих значень.

Клас `RNSService` імплементує інтерфейс `IRNSService` і реалізує метод `Decode`. Метод `Decode` здійснює декодування закодованих даних сенсорів за рахунок знаходження модулів закодованого значення за значеннями `p1`, `p2`, `p3` і `p4`, які є закритими членами класу `RNSService`, та повертає об'єкт класу `StorageItem`. Знаходження модуля здійснюється через використання закритого метода `Mod`.

Клас `StorageService` імплементує інтерфейс `IStorageService` і реалізує методи `GetItems` та `AddItem`. Клас призначений для збереження історії декодованих значень у вигляді набору об'єктів класу `StorageItem`.

4.2.2 Опис проекту WebApp

Проект `WebApp` це проект типу `Console Application` на базі `.NET` фреймворку версії 6.0. Проект представляє собою `Web API` [37], що розгортається на хості `IIS Express`. Містить налаштування для розгортання веб-додатку на основі `Angular` (розділ 4.3). Призначення проекту це здійснення комунікації між симуляцією

бездротової сенсорної мережі на основі Node-RED (розділ 4.1) та візуальною частиною а також здійснення обробки даних сенсорів.

Проект має наступну структуру:

1. Controllers. Містить ендпойнти.
2. Mappers. Містить мапери для перетворення бізнес-об'єктів в об'єкти DTO і навпаки.
3. Models. Містить DTO-об'єкти.
4. ClientApp. Містить веб додаток Angular SPA.

Повна структура проекту WebApp показана на рисунку 4.20:

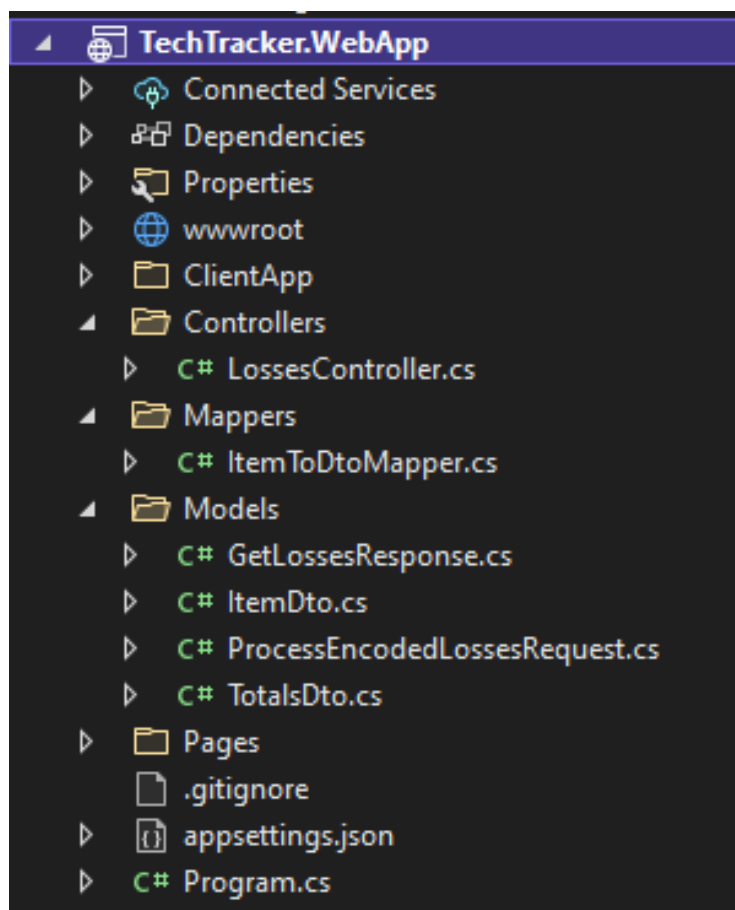


Рисунок 4.20 – Структура проекту WebApp в Visual Studio

Клас LossesController містить наступні ендпойнти:

1. GetLosses. Викликається при здійсненні GET запиту по url адресі /losses/list. Повертає список об'єктів StorageItem за допомогою IStorageService;
2. ProcessEncodedLosses. Викликається при здійсненні POST запиту по url

адресі `/losses/process`. Як параметр приймає об'єкт типу `ProcessEncodedLossesRequest`, який містить закодоване значення сенсорів. Здійснює декодування даних за допомогою `IRNSService` та збереження декодованого значення за допомогою `IstorageService`.

3. `StartSimulation`. Викликається при здійсненні GET запиту по url адресі `/losses/start`. Здійснює запит до системи симуляції на Node-RED який запускає роботу БСМ.

4. `StopSimulation`. Викликається при здійсненні GET запиту по url адресі `/losses/stop`. Здійснює запит до системи симуляції на Node-RED який призупиняє роботу БСМ.

Клас `ProcessEncodedLossesRequest` представляє собою модель, яку надсилає система симуляції БСМ. Містить поле `EncodedValue` яке зберігає закодоване значення сенсорів.

Клас `ItemDto` представляє DTO класу `StorageItem`. Призначений для комунікації проекту з клієнт-додатком.

Клас `GetLossesResponse` представляє собою об'єкт який повертається у відповідь на виклик ендпоинту `LossesController.GetLosses`.

4.3 Реалізація візуальної частини системи збору та обробки даних за допомогою Angular

Angular – це безкоштовна платформа веб-додатків з відкритим вихідним кодом на основі TypeScript, яку очолює команда Angular у Google.

В ході реалізації системи збору та обробки даних було створено проект `ClientApp` з використанням фреймворку Angular.

Проект містить наступні основні компоненти:

1. `nav-menu`: представляє собою навігаційне меню;
2. `losses-table`: містить таблицю яка відображає 10 останніх передані показники сенсорів;
3. `losses-tile`: представляє собою елемент таблиці `losses-table`.

Повна структура проекту показана на рисунку 4.21.

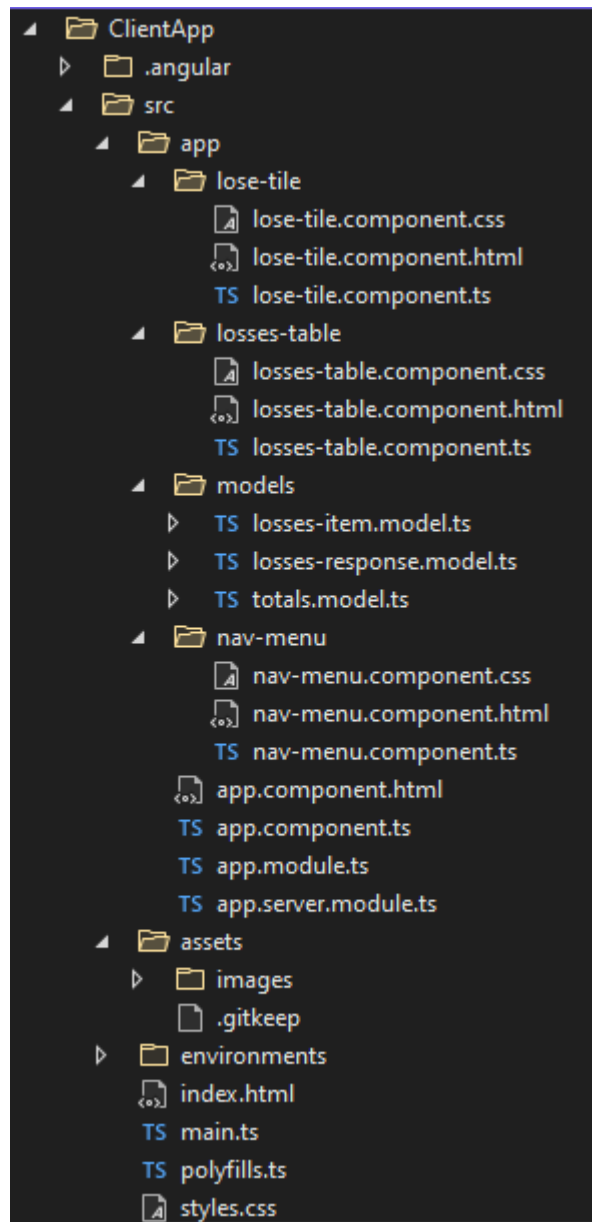


Рисунок 4.21 – Повна структура проекту ClientApp

4.3.1 Компонент nav-menu

Компонент nav-menu містить розмітку, стилі та функціонал навігаційної панелі. Містить дві кнопки для запуску і припинення роботи симуляції БСМ.

При натисканні кнопки запуску здійснюється GET запит до Web API системи збору та обробки даних (розділ 4.2) на url адресу /losses/start, яка в свою чергу здійснює аналогічний запит до Node-RED і запускає процес генерації та передачі

даних сенсорів.

Також, при натисканні кнопки запуску, здійснюються запити до системи збору та обробки даних на url адресу /losses/list для отримання даних з певним інтервалом. Таким чином забезпечено візуальне представлення останніх показників сенсорів у вигляді таблиці.

При натисканні кнопки припинення здійснюється GET запит до Web API системи збору та обробки даних на url адресу /losses/stop, яка в своє чергу здійснює GET запит до Node-RED і припиняє процес генерації та пережачі даних з встановленим інтервалом, тим самим припиняючи надходження нових даних до клієнтської частини системи.

4.3.2 Компонент losses-table

Компонент losses-table містить розмітку, стилі та функціонал таблиці яка відображає останні отримані розкодовані дані сенсорів.

Актуальність даних забезпечується шляхом підписки на події компоненту nav-menu. Після отримання даних з системи збору та обробки даних, вміст таблиці оновлюється відповідно до отриманих даних

4.4 Перевірка коректності роботи системи

Для перевірки коректності роботи системи згенеровано та закодовано 10 наборів значень. Контроль здійснено в наступних частинах системи:

1. Симуляція БСМ (Node-RED): перевірка генерації та кодування значень сенсорів а також відправка даних до системи збору та обробки даних;
2. Система збору та обробки даних (ASP.NET Core): перевірка декодування значень та передача даних до клієнт-додатку;
3. Клієнт-додаток (Angular): перевірка відображення представлення декодованих значень.

4.4.1 Перевірка коректності роботи БСМ

Перевірка генерації та кодування значень у моделюванні БСМ у Node-RED здійснено за допомогою логування даних у консоль редактора Node-RED. У таблиці 4.1 показано результати роботи БСМ. Система повинна генерувати та кодувати 4 цілочисельних значення.

Таблиця 4.1 – 10 наборів згенерованих та закодованих значень сенсорів БСМ

№	Набір згенерованих значень сенсорів	Закодовані значення сенсорів
1	47, 14, 4, 17	1112116736584
2	37, 13, 14, 21	579677120987
3	27, 37, 2, 6	600836980414
4	43, 17, 19, 20	894470230226
5	24, 2, 13, 49	737269638276
6	21, 20, 33, 46	97561964963
7	38, 12, 49, 23	685769388164
8	42, 46, 33, 22	58808437994
9	20, 12, 42, 21	331000632360
10	25, 38, 40, 39	1149045320351

Дані, внесені в таблицю 4.1, свідчать про коректність роботи системи БСМ в Node-RED.

4.4.2 Перевірка коректності роботи системи збору та обробки даних

Перевірено коректність прийому та декодування даних БСМ. Система повинна приймати закодоване значення сенсорів та здійснювати розкодування, в результаті чого мають бути отримані набори з 4 значень, які представляють значення кожного з 4 сенсорів БСМ. Отримані дані та результати декодування показано на таблиці 4.2.

Таблиця 4.2 – 10 наборів закодованих та декодованих значень сенсорів

№	Закодовані значення сенсорів	Набір декодованих значень сенсорів
1	1112116736584	47, 14, 4, 17
2	579677120987	37, 13, 14, 21
3	600836980414	27, 37, 2, 6
4	894470230226	43, 17, 19, 20
5	737269638276	24, 2, 13, 49
6	97561964963	21, 20, 33, 46
7	685769388164	38, 12, 49, 23
8	58808437994	42, 46, 33, 22
9	331000632360	20, 12, 42, 21
10	1149045320351	25, 38, 40, 39

Дані, внесені в таблицю 4.2, у порівнянні з даними, внесеними в таблицю 4.1 свідчать про коректність роботи системи збору та обробки даних.

4.4.3 Перевірка коректності роботи клієн-додатку

Клієнт-додаток повинен відображати дані, отримані з системи збору та обробки даних. Закодовані та декодовані значення сенсорів відображено на рисунку 4.22.

1112116736584	47	14	4	15
579677120987	37	13	14	21
600836980414	27	37	2	6
894470230226	43	17	19	20
737269638276	24	2	13	49
97561964963	21	20	33	46
685769388164	38	12	49	23
58808437994	42	46	33	22
331000632360	20	12	42	21
1149045320351	25	38	40	39

Рисунок 4.22 – Значення сенсорів, відображені в клієнт-додатку

Отримавши результати роботи кожного з 3 етапів системи, складено порівняльну таблицю 4.3, в якій представлено дані БСМ та дані, відображені у клієнт-додатку.

Таблиця 4.3 – Порівняльна таблиця переданих даних БСМ та віжображених даних клієнт-додатку

№	БСМ		Клієнт-додаток	
	Згенеровані значення	Закодовані значення	Закодовані значення	Декодовані значення
1	47, 14, 4, 17	1112116736584	1112116736584	47, 14, 4, 17
2	37, 13, 14, 21	579677120987	579677120987	37, 13, 14, 21
3	27, 37, 2, 6	600836980414	600836980414	27, 37, 2, 6
4	43, 17, 19, 20	894470230226	894470230226	43, 17, 19, 20
5	24, 2, 13, 49	737269638276	737269638276	24, 2, 13, 49
6	21, 20, 33, 46	97561964963	97561964963	21, 20, 33, 46
7	38, 12, 49, 23	685769388164	685769388164	38, 12, 49, 23
8	42, 46, 33, 22	58808437994	58808437994	42, 46, 33, 22

Продовження таблиці 4.3 - Порівняльна таблиця переданих даних БСМ та візображених даних клієнт-додатку

№	БСМ		Клієнт-додаток	
	Згенеровані значення	Закодовані значення	Закодовані значення	Декодовані значення
9	20, 12, 42, 21	331000632360	331000632360	20, 12, 42, 21
10	25, 38, 40, 39	1149045320351	1149045320351	25, 38, 40, 39

Дані, представлені в таблиці 4.3, свідчать про коректність роботи системи в цілому, а саме генерація, кодування, передача та декодування даних на 3 етапах системи.

4.5 Висновки

Створено систему, яка симулює роботу БСМ та систему збору та обробки даних з цієї БСМ.

Система складається з трьох частин:

1. Симуляція БСМ – підсистема, реалізована за допомогою Node-RED, яка відповідає за генерацію, кодування та передачі даних до системи збору та обробки;
2. Система збору та обробки даних – підсистема, реалізована за допомогою ASP.NET, містить Web API, відповідає за декодування та зберігання даних сенсорів;
3. Клієнтський додаток – візуальна частина, реалізована за допомогою Angular, відповідає за візуальне представлення даних сенсорів, отриманих з системи збору та обробки даних.

Комунікація трьох частин системи здійснюється через HTTP запити, де Web API системи збору та обробки даних виступає проміжним елементом в комунікації клієнт-додатку з Node-RED.

Здійснено контроль коректності роботи системи в цілому.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено метод кодування даних сенсорів а також віртуальну БСМ із застосуванням розробленого методу кодування за допомогою Node-RED та систему збору та обробки даних за допомогою ASP.NET Core і Angular.

У першому розділі здійснено аналіз бездротових сенсорних мереж, що включає область використання, класифікація, характеристики та топології БСМ, також здійснено огляд поширених архітектур БСМ та стандартну будову сенсорного вузла.

У другому розділі проведено огляд недвійкових систем числення, особливо систему залишкових класів. Досліджено характеристики системи залишкових класів, особливості здійснення базових математичних операцій та переваги і недоліки у порівнянні з іншими системами числення. Здійснено аналіз області застосування системи залишкових класів.

У третьому розділі оглянуто багторівневий метод кодування на базі системи залишкових класів. Представлено запропонованих метод кодуванням та проведено дослідження ефективності запропонованого методу.

У четвертому розділі здійснено імплементацію віртуальної системи БСМ за допомогою Node-RED та системи збору та обробки даних за допомогою ASP.NET та Angular. Для кодування, передачі та декодування даних сенсорів застосовано розроблений метод кодування. Здійснено контроль коректності роботи системи шляхом звірення даних до кодування та передачі та даних, отриманих після декодування у системі збору та обробки даних.

Набув подальшого розвитку метод кодування та передачі даних у бездротових сенсорних мережах.

Впровадження результатів роботи дозволили підвищити ефективність використання пропускну здатності мережі БСМ шляхом скорочення обсягу службової інформації, що передається при передачі показників сенсорів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Yatskiv V., Su Jun, Yatskiv N., Sachenko A., Osolinskiy O. Multilevel Method of data coding in WSN. *Proceeding of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications: IDAACS '2011*, 2011. – P. 863 – 866.
2. Yatskiv V., Yatskiv N., Sachenko A., Yatskiv S., Tsavolyk T. Adaptive Data Transmission Protocol for Wireless Sensor Networks Based on Residue Number System Correcting Codes. *Proceedings of the 4th IEEE International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS* (Lviv, Ukraine, 20-21 September, 2018), Lviv. 2018. P.131-136.
3. Ullo, S. L., & Sinha, G. R. Advances in smart environment monitoring systems using IoT and sensors. *Sensors*. 2020, p. 3113.
4. Peiris, V. Highly integrated wireless sensing for body area network applications. *SPIE Newsroom*. 2013, 10(2.1201312), p.00512.
5. O'Donovan, T., O'Donoghue, J., Sreenan, C., Sammon, D., O'Reilly, P., & O'Connor, K. A. A context aware wireless body area network (BAN). *2009 3rd International Conference on Pervasive Computing Technologies for Healthcare*. 2009, pp. 1-8.
6. Bilal, M., & Kang, S. G. An authentication protocol for future sensor networks. *Sensors*, 17(5). 2017, p. 979.
7. Hart, J. K., & Martinez, K. (2006). Environmental sensor networks: A revolution in the earth system science?. *Earth-Science Reviews*, 78(3-4). 2006, pp. 177-191.
8. Apte, J. S., Messier, K. P., Gani, S., Brauer, M., Kirchstetter, T. W., Lunden, M. M., ... & Hamburg, S. P. High-resolution air pollution mapping with Google street view cars: exploiting big data. *Environmental science & technology*, 51(12). 2017, pp. 6999-7008.

9. Ma, Q. *New insights into understanding urban traffic emissions using novel mobile air quality measurements in the Breathe London pilot study* (Doctoral dissertation, University of Cambridge). 2021.
10. Feinberg, S., Williams, R., Hagler, G. S., Rickard, J., Brown, R., Garver, D., ... & Garvey, S. Long-term evaluation of air sensor technology under ambient conditions in Denver, Colorado. *Atmospheric measurement techniques*, 11(8). 2018, pp. 4605-4615.
11. Balzano, L., & Nowak, R. Blind calibration of networks of sensors: Theory and algorithms. *Networked Sensing Information and Control*. 2018, pp. 9-37.
12. Saukh, O., Hasenfratz, D., & Thiele, L. Reducing multi-hop calibration errors in large-scale mobile sensor networks. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. 2015, pp. 274-285.
13. Sreeja, S. P., & Bharathi, M. A Survey on Security and Privacy Issues in Wireless Sensor Networks for Healthcare application.
14. Tiwari, A., Ballal, P., & Lewis, F. L. Energy-efficient wireless sensor network design and implementation for condition-based maintenance. *ACM Transactions on Sensor Networks (TOSN)*, 3(1). 2017, p.1.
15. Saleem, K., Faisal, N., & Al-Muhtadi, J. Empirical studies of bio-inspired self-organized secure autonomous routing protocol. *IEEE Sensors Journal*, 14(7). 2014, pp. 2232-2239.
16. Anastasi, G., Farruggia, O., Re, G. L., & Ortolani, M. Monitoring high-quality wine production using wireless sensor networks. In *2009 42nd Hawaii International Conference on System Sciences*. 2014, pp. 1-7.
17. Igartua, M. A., Mendoza, F. A., Redondo, R. P. D., Vicente, M. I. M., Forné, J., Campo, C. & Rico-Novella, F. J. INRISCO: INcident monitoRING in smart communities. *IEEE access*, 8. 2020, pp. 72435-72460.
18. Pastor-Aparicio, A., Segura-Garcia, J., Lopez-Ballester, J., Felici-Castell, S., Garcia-Pineda, M., & Pérez-Solano, J. J. Psychoacoustic annoyance implementation with wireless acoustic sensor networks for monitoring in smart cities. *IEEE Internet of Things Journal*, 7. 2019, pp. 128-136.

19. Lloret, J., Garcia, M., Bri, D., & Sendra, S. A wireless sensor network deployment for rural and forest fire detection and verification. *sensors*, 9. 2009, pp. 8722-8747.
20. Fahmida, S., Modekurthy, V. P., Rahman, M., Saifullah, A., & Brocanelli, M. Long-lived LoRa: Prolonging the lifetime of a LoRa network. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*. 2020, pp. 1-12.
21. Bowen, C. R., Topolov, V. Y., & Kim, H. A. Piezoelectric mechanical energy harvesters and related materials. In *Modern Piezoelectric Energy-Harvesting Material*. 2016, pp. 113-138.
22. Saleem, K., Fisal, N., Hafizah, S., Kamilah, S., Rashid, R., & Baguda, Y. Cross layer based biological inspired self-organized routing protocol for wireless sensor network. In *TENCON 2009-2009 IEEE Region 10 Conference*. 2009, pp. 1-6.
23. Miao, G., Zander, J., Sung, K. W., & Slimane, S. B. *Fundamentals of mobile data networks*. Cambridge University Press. 2016.
24. Aghdam, S. M., Khansari, M., Rabiee, H. R., & Salehi, M. WCCP: A congestion control protocol for wireless multimedia communication in sensor networks. *Ad Hoc Networks*, 13. 2014, pp. 516-534.
25. Y. Xu, J. Heidemann, and D. Estrin, Geography-informed energy conservation for ad hoc routing. *Proc. Mobicom*. 2001, pp. 70–84.
26. Amit Sinha and Anantha Chandrakasan. Dynamic Power Management in Wireless Sensor Networks. *IEEE Design & Test of Computers*, Vol. 18, No. 2. 2014.
27. Mohan, P. A. Springer Science & Business Media. *Residue number systems: algorithms and architectures*. 2016.
28. Omondi, A. R., & Premkumar, A. B. *Residue number systems: theory and implementation* (Vol. 2). World Scientific. 2017.
29. Omondi A. Residue Number System: Theory and Implementation A.Omond, B.Premkumar. Imperial College Press, 2007. – Vol. 2. – 296 p.
30. Wang, Yuke. Residue-to-binary converters based on new Chinese remainder theorems. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 2000, 47.3: 197-205.

31. Node-RED, 2022, URL: <https://nodered.org/> (дата звернення 20.02. 2022).
32. Knowing when a node is done, 2019, URL: <https://nodered.org/blog/2019/09/20/node-done> (дата звернення 27.02.2022).
33. Create an HTTP Endpoint, 2022, URL: <https://cookbook.nodered.org/http/create-an-http-endpoint> (дата звернення 27.02.2022).
34. The Core Nodes, 2022, URL: <https://nodered.org/docs/user-guide/nodes#change> (дата звернення 27.02.22).
35. The Core Nodes, 2022, URL: <https://nodered.org/docs/user-guide/nodes#switch> (дата звернення 27.02.22).
36. Freeman, A. *Pro Asp. net core MVC*. Apress. 2016.
37. Mulloy, B. *Web API design*. 2017.
38. Shahzad, M. K., Nguyen, D. T., Zalyubovskiy, V., & Choo, H. LNDIR: A lightweight non-increasing delivery-latency interval-based routing for duty-cycled sensor networks. *International Journal of Distributed Sensor Networks*, 14(4). 2018, p. 1550147718767605.
39. Simon, D., Cifuentes, C., Cleal, D., Daniels, J., & White, D. Java™ on the bare metal of wireless sensor devices: the squawk Java virtual machine. *In Proceedings of the 2nd international conference on Virtual execution environments*. 2016, pp. 78-88.
40. Silva, D., Ghanem, M., & Guo, Y. WikiSensing: an online collaborative approach for sensor data management. *Sensors*, 12(10). 2012, pp. 13295-13332.
41. Niazi, M. A., & Hussain, A. A novel agent-based simulation framework for sensing in complex adaptive environments. *IEEE Sensors Journal*, 11(2). 2014, pp. 404-412.
42. Bulusu, N., Heidemann, J., & Estrin, D. GPS-less low-cost outdoor localization for very small devices. *IEEE personal communications*, 7(5). 2000, pp. 28-34.
43. Mitchell, R., & Chen, R. A survey of intrusion detection in wireless network applications. *Computer Communications*, 42. 2014, pp. 1-23.
44. Li, W., Wang, Z., Wei, G., Ma, L., Hu, J., & Ding, D. A survey on multisensor fusion and consensus filtering for sensor networks. *Discrete Dynamics in Nature and Society*, 2015. 2015.

45. Bosman, H. H., Iacca, G., Tejada, A., Wörtche, H. J., & Liotta, A. Ensembles of incremental learners to detect anomalies. *Ad hoc sensor networks. ad hoc networks*, 35. 2015, pp. 14-36.
46. Bosman, H. H., Liotta, A., Iacca, G., & Wörtche, H. J. Anomaly detection in sensor systems using lightweight machine learning. *IEEE International Conference on Systems, Man, and Cybernetics*. 2013, pp. 7-13.
47. Bosman, H. H., Liotta, A., Iacca, G., & Wörtche, H. J. Online extreme learning on fixed-point sensor networks. *IEEE 13th International Conference on Data Mining Workshops*. 2013, pp. 319-326.
48. Bosman, H. H., Iacca, G., Wörtche, H. J., & Liotta, A. Online fusion of incremental learning for wireless sensor networks. *IEEE International Conference on Data Mining Workshop*. 2014, pp. 525-532.
49. Iacca, G. Distributed optimization in wireless sensor networks: an island-model framework. *Soft Computing*, 17(12). 2018, pp. 2257-2277.
50. Bosman, H. H., Iacca, G., Tejada, A., Wörtche, H. J., & Liotta, A. Spatial anomaly detection in sensor networks using neighborhood information. *Information Fusion*, 33. 2017, pp. 41-56.

ДОДАТОК А (обов'язковий) КОД РОЗРОБЛЕНОЇ БСМ

Код створеної БСМ за допомогою Node-RED в форматі JSON (для імпорту в Node-RED)

```
[
  {
    "id": "7ec279858bd668ff",
    "type": "tab",
    "label": "Flow 2",
    "disabled": false,
    "info": "function getRandomInt(min, max) {\n  return\n  Math.floor(Math.random() * (max - min + 1)) + min;\n}\n\ncontext.loop\n= context.loop || \"stop\";\ncontext.loops = context.loops ||\n0;\n\nswitch (msg.payload) {\n\tcase \"stop\":\n\t\tcontext.loops =\n\t\tcontext.loops + 1;\n\t\tmsg.payload = \"stopped\";\n\t\tcontext.loop\n\t\t= \"stop\";\n\t\treturn [msg,null];\n\tcase \"toggle\":\n\t\tif\n\t\t(context.loop == \"start\") {\n\t\t\tmsg.payload =\n\t\t\t\"stopped\";\n\t\t\tcontext.loop = \"stop\";\n\t\t\treturn\n\t\t\t[msg,null];\n\t\t\t} else {\n\t\t\t\tmsg.payload =\n\t\t\t\t\"started\";\n\t\t\t\tcontext.loop = \"loop\";\n\t\t\t\tcontext.loops =\n\t\t\t\t1;\n\t\t\t\treturn [msg,msg];\n\t\t\t}\n\t\tcase\n\t\t\"start\":\n\t\t\tmsg.payload = 20;\n\t\t\tcontext.loop =\n\t\t\t\"loop\";\n\t\t\tcontext.loops = 1;\n\t\t\treturn\n\t\t\t[msg,msg];\n\t\t\tdefault:\n\t\t\t\tif (context.loop == \"loop\")\n\t\t\t\t{\n\t\t\t\t\tcontext.loops = context.loops + 1;\n\t\t\t\t\tif (msg.payload >\n\t\t\t\t\t40) {\n\t\t\t\t\t\tmsg.payload = msg.payload + getRandomInt(-4,\n\t\t\t\t\t\t0)\n\t\t\t\t\t\t}\n\t\t\t\t\telse if (msg.payload < -20) {\n\t\t\t\t\t\tmsg.payload = msg.payload + getRandomInt(-4, 0)\n\t\t\t\t\t\t}\n\t\t\t\t\telse\n\t\t\t\t\t{\n\t\t\t\t\t\tmsg.payload = msg.payload + getRandomInt(-3,\n\t\t\t\t\t\t3);\n\t\t\t\t\t}\n\t\t\t\t\treturn [msg,msg];\n\t\t\t\t} else {\n\t\t\t\t\treturn\n\t\t\t\t\t[null,null];\n\t\t\t\t}\n\t\t}\n}",
    "env": []
  },
  {
    "id": "aa56b4e7258d3003",
    "type": "http request",
    "z": "7ec279858bd668ff",

```

```
"name": "Send to API",
"method": "POST",
"ret": "obj",
"paytoqs": "body",
"url": "https://localhost:44497/losses/process",
"tls": "28b72b04b2add527",
"persist": false,
"proxy": "",
"authType": "",
"senderr": false,
"x": 810,
"y": 420,
"wires": [
  [
    "e7145d99bc0a3327"
  ]
]
},
{
  "id": "ae64668202fdbe0b",
  "type": "change",
  "z": "7ec279858bd668ff",
  "name": "",
  "rules": [
    {
      "t": "set",
      "p": "start",
      "pt": "global",
      "to": "true",
      "tot": "bool"
    }
  ]
}
```

```
],
  "action": "",
  "property": "",
  "from": "",
  "to": "",
  "reg": false,
  "x": 360,
  "y": 80,
  "wires": [
    []
  ]
},
{
  "id": "c4cc1f7e87cf1f15",
  "type": "http in",
  "z": "7ec279858bd668ff",
  "name": "",
  "url": "/start",
  "method": "get",
  "upload": false,
  "swaggerDoc": "",
  "x": 160,
  "y": 120,
  "wires": [
    [
      "1313f7f91ff20a0c",
      "ae64668202fdbe0b"
    ]
  ]
},
{
```

```

    "id": "1313f7f91ff20a0c",
    "type": "http response",
    "z": "7ec279858bd668ff",
    "name": "",
    "statusCode": "",
    "headers": {},
    "x": 330,
    "y": 160,
    "wires": []
  },
  {
    "id": "ff1a5ec1933b0eb0",
    "type": "function",
    "z": "7ec279858bd668ff",
    "name": "Encryption",
    "func": "function mod(n, m) {\n  return ((n % m) + m) %\n  m;\n}\n\nvar sensorsValuesArray = msg.payload;\n\nvar x1 =\n  sensorsValuesArray[0];\n\nvar x2 = sensorsValuesArray[1];\n\nvar x3 =\n  sensorsValuesArray[2];\n\nvar x4 = sensorsValuesArray[3];\n\n\nvar k0 =\n  516;\n\nvar k1 =\n  104;\n\nvar k2 =\n  231123;\n\n\nvar p1 =\n  1031;\n\nvar p2 =\n  1033;\n\nvar p3 =\n  1039;\n\nvar p4 =\n  1049;\n\nvar p12 =\n  p1 * p2;\n\nvar p34 =\n  p3 * p4;\n\n\nvar N1 =\n  x2 + p2 * mod((k0 * (x1 - x2)), p1);\n\nvar N2 =\n  x4 + p4 * mod((k1 * (x3 - x4)), p3);\n\n\nvar X01 =\n  N2 + p34 * mod((k2 * (N1 - N2)), p12);\n\n\nmsg.payload = { \"encodedValue\":\n  X01.toString() };\n\nreturn msg;";
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 610,
    "y": 300,
    "wires": [
      [

```

```

        "0dd0f93942770830",
        "bde4bfe4e9254e63"
    ]
]
},
{
    "id": "0dd0f93942770830",
    "type": "function",
    "z": "7ec279858bd668ff",
    "name": "Add Headers",
    "func": "msg.headers = {}; \nmsg.headers['content-type'] =
'application/json'; \n\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 710,
    "y": 360,
    "wires": [
        [
            "aa56b4e7258d3003"
        ]
    ]
},
{
    "id": "5d7e3a34c371a7f1",
    "type": "function",
    "z": "7ec279858bd668ff",
    "name": "Generate Sensors Values",

```

```

    "func": "function generateRandom(min = 1, max = 50) {\n
let difference = max - min;\n    let rand = Math.random();\n    rand
= Math.floor( rand * difference);\n    rand = rand + min;\n\n
return rand;\n}\n\nvar sensorsCount =
global.get(\"sensorsCount\");\n\nif (sensorsCount === 0 ||
sensorsCount === undefined)\n    sensorsCount = 4;\n\nvar
sensorsValuesArray = [];\n\nfor (let step = 0; step < sensorsCount;
step++) {\n
sensorsValuesArray.push(generateRandom());\n}\n\nmsg.payload =
sensorsValuesArray;\n\nreturn msg;\",

    "outputs": 1,

    "noerr": 0,

    "initialize": "",

    "finalize": "",

    "libs": [],

    "x": 470,

    "y": 240,

    "wires": [

        [

            "ff1a5ec1933b0eb0",

            "bde4bfe4e9254e63"

        ]

    ]

},

{

    "id": "bde4bfe4e9254e63",

    "type": "debug",

    "z": "7ec279858bd668ff",

    "name": "Log",

    "active": true,

    "tosidebar": true,

    "console": false,

    "tostatus": false,

    "complete": "payload",

```

```
"targetType": "msg",
"statusVal": "",
"statusType": "auto",
"x": 870,
"y": 240,
"wires": []
},
{
  "id": "57d5a6cf92bc394e",
  "type": "delay",
  "z": "7ec279858bd668ff",
  "name": "Delay",
  "pauseType": "delayv",
  "timeout": "5",
  "timeoutUnits": "seconds",
  "rate": "1",
  "nbRateUnits": "1",
  "rateUnits": "second",
  "randomFirst": "1",
  "randomLast": "5",
  "randomUnits": "seconds",
  "drop": false,
  "allowrate": false,
  "outputs": 1,
  "x": 630,
  "y": 480,
  "wires": [
    [
      "431214e91237667c"
    ]
  ]
}
```

```

},
{
  "id": "e7145d99bc0a3327",
  "type": "function",
  "z": "7ec279858bd668ff",
  "name": "Set Delay",
  "func": "msg.delay = global.get(\"delay\");\n\nreturn msg;",
  "outputs": 1,
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 860,
  "y": 480,
  "wires": [
    [
      "57d5a6cf92bc394e"
    ]
  ]
},
{
  "id": "3f4814c217775a94",
  "type": "change",
  "z": "7ec279858bd668ff",
  "name": "",
  "rules": [
    {
      "t": "set",
      "p": "delay",
      "pt": "global",
      "to": "payload.delay",

```

```
        "tot": "msg"
      }
    ],
    "action": "",
    "property": "",
    "from": "",
    "to": "",
    "reg": false,
    "x": 1120,
    "y": 80,
    "wires": [
      []
    ]
  },
  {
    "id": "c06bb5157611a3ba",
    "type": "http in",
    "z": "7ec279858bd668ff",
    "name": "",
    "url": "/set-delay",
    "method": "post",
    "upload": false,
    "swaggerDoc": "",
    "x": 900,
    "y": 120,
    "wires": [
      [
        "175a772e050abdfc",
        "3f4814c217775a94"
      ]
    ]
  }
]
```

```
},
{
  "id": "175a772e050abdfc",
  "type": "http response",
  "z": "7ec279858bd668ff",
  "name": "",
  "statusCode": "",
  "headers": {},
  "x": 1090,
  "y": 160,
  "wires": []
},
{
  "id": "e7b90cc88560fcee",
  "type": "complete",
  "z": "7ec279858bd668ff",
  "name": "On Start",
  "scope": [
    "ae64668202fdbe0b"
  ],
  "uncaught": false,
  "x": 380,
  "y": 360,
  "wires": [
    [
      "431214e91237667c"
    ]
  ]
},
{
  "id": "89b6ebb8dcdbc440",
```

```
"type": "change",
"z": "7ec279858bd668ff",
"name": "",
"rules": [
  {
    "t": "set",
    "p": "start",
    "pt": "global",
    "to": "false",
    "tot": "bool"
  }
],
"action": "",
"property": "",
"from": "",
"to": "",
"reg": false,
"x": 720,
"y": 80,
"wires": [
  []
]
},
{
  "id": "29a3b1b36a75ddfc",
  "type": "http in",
  "z": "7ec279858bd668ff",
  "name": "",
  "url": "/stop",
  "method": "get",
  "upload": false,
```

```

"swaggerDoc": "",
"x": 520,
"y": 120,
"wires": [
  [
    "ea7bb4b56d673dc8",
    "89b6ebb8dcdbc440"
  ]
]
},
{
  "id": "ea7bb4b56d673dc8",
  "type": "http response",
  "z": "7ec279858bd668ff",
  "name": "",
  "statusCode": "",
  "headers": {},
  "x": 690,
  "y": 160,
  "wires": []
},
{
  "id": "431214e91237667c",
  "type": "switch",
  "z": "7ec279858bd668ff",
  "name": "",
  "property": "start",
  "propertyType": "global",
  "rules": [
    {
      "t": "true"
    }
  ]
}

```

```
    }  
  ],  
  "checkall": "true",  
  "repair": false,  
  "outputs": 1,  
  "x": 510,  
  "y": 400,  
  "wires": [  
    [  
      "5d7e3a34c371a7f1"  
    ]  
  ]  
},  
{  
  "id": "28b72b04b2add527",  
  "type": "tls-config",  
  "name": "",  
  "cert": "",  
  "key": "",  
  "ca": "",  
  "certname": "",  
  "keyname": "",  
  "caname": "",  
  "servername": "https://localhost",  
  "verifyservercert": false,  
  "alpnprotocol": ""  
}  
]
```

ДОДАТОК Б

ТЕЗИ ДОПОВІДІ КОНФЕРЕНЦІЇ АПКН-2021

Актуальні проблеми комп'ютерних наук

УДК 004.4

Федоришин О. М., Яцків В. В.

*Хмельницький національний університет***СПОСІБ КОДУВАННЯ ДАНИХ СЕНСОРІВ НА ОСНОВІ СИСТЕМИ
ЗАЛИШКОВИХ КЛАСІВ**

Запропоновано спосіб кодування даних сенсорів на основі перетворення системи залишкових класів. Розроблено структурну схему кодування даних розрядністю 10 біт для восьми сенсорів. Наведено приклад кодування та декодування даних для заданої системи модулів.

The sensor data encoding approach based on the Residue Number System is proposed. The block diagram of 10-bit data coding for eight sensors has been developed. An example of data encoding and decoding for the defined modulo is shown.

Безпроводні технології передачі даних широко використовуються в різних галузях економіки, в тому числі в системах збору інформації про стан фізичного середовища, в задачах технічного та екологічного моніторингу та інших.

Однак, передача повідомлень малого обсягу (8 -24 біт), які формують сенсори, приводить до неефективного використання пропускну здатності безпроводних сенсорних мереж (БСМ). Кожний пакет, який передається в БСМ, крім даних містить службову інформацію необхідну для надійної доставки і обробки пакетів. В [1, 2] показано, що відсоток службових даних у деяких протоколах становить від 4 до 80 % та залежить від довжини поля даних. Отже, розробка методів кодування направлених на зменшення обсягу службових даних є актуальною науковою задачею.

Метою роботи є розробка способу кодування та передачі даних сенсорів, який дозволить зменшити обсяг службової інформації в протоколах передачі даних безпроводних сенсорних мереж.

В роботі запропоновано спосіб кодування на основі використання прямого та оберненого перетворення системи залишкових класів [3, 4].

Суть пропонованого способу кодування полягає у наступному: вибираємо систему взаємно простих чисел (модулів) значення яких має бути більшим за максимальне значення даних сенсора. Відповідно, дані сенсорів x_1 - x_8 будемо вважати залишками по вибраній системі модулів. Наступним кроком здійснюємо перетворення залишків x_1 - x_8 у позиційну (десяткову або двійкову) систему числення. Код для проведення відповідних обчислень на мові Python має вигляд:

$$\begin{aligned}
 N1 &= x2 + p2 * ((k0 * (x1 - x2)) \% p1) \\
 N2 &= x4 + p4 * ((k1 * (x3 - x4)) \% p3) \\
 X01 &= N2 + p34 * ((k2 * (N1 - N2)) \% (p12)) \\
 N3 &= x6 + p6 * ((k3 * (x5 - x6)) \% p5) \\
 N4 &= x8 + p8 * ((k4 * (x7 - x8)) \% p7) \\
 X02 &= N4 + p78 * ((k5 * (N3 - N4)) \% p56) \\
 X &= X02 + p5_8 * ((k6 * (X01 - X02)) \% p1_4), \tag{1}
 \end{aligned}$$

де $p12 = p1 * p2$; $p34 = p3 * p4$; $p56 = p5 * p6$; $p78 = p7 * p8$; $p1_4 = p1 * p2 * p3 * p4$; $p5_8 = p5 * p6 * p7 * p8$; $k0 * p2 \% p1 = 1$; $k1 * p4 \% p3 = 1$; $k2 * p34 \% p12 = 1$; $k3 * p6 \% p5 = 1$; $k4 * p8 \% p7 = 1$; $k5 * p78 \% p56 = 1$; $k6 * p5_8 \% p1_4 = 1$.

Структурна схема перетворення даних з СЗК у десяткову на основі нової китайської теореми про залишки приведена на рисунку 1 [3].

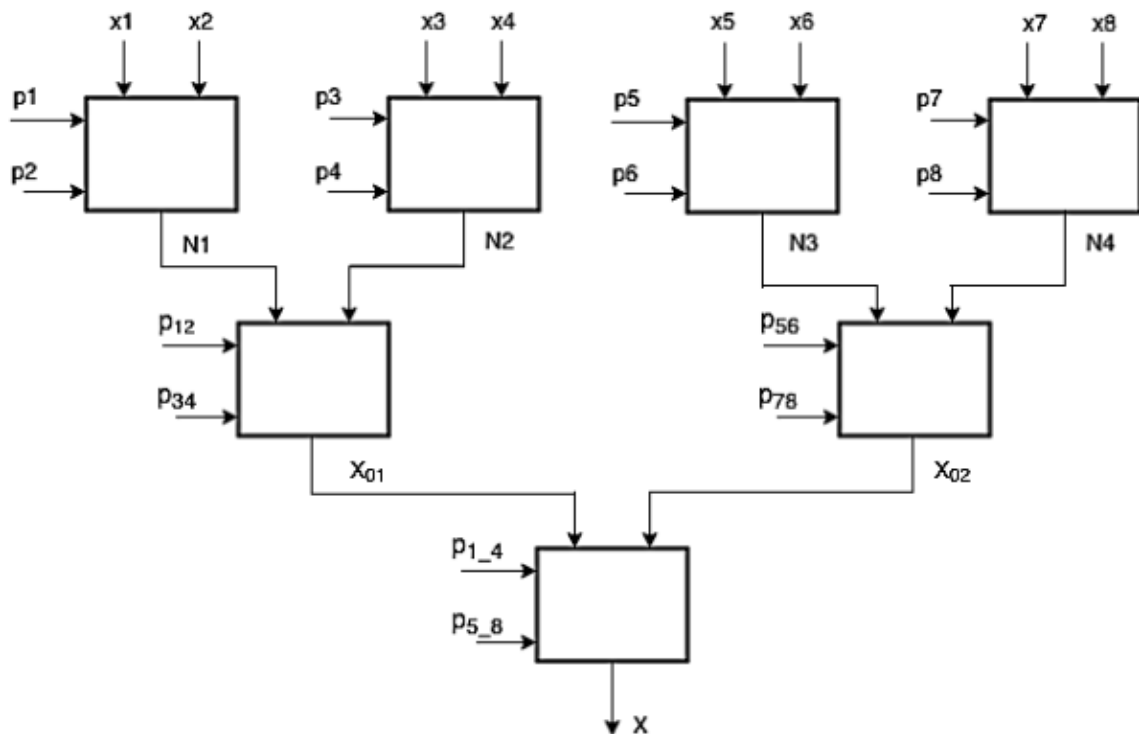


Рисунок 1 – Структурна схема кодування даних сенсорів

В результаті обчислення отримаємо число у десятковій або двійковій системі числення, яке відображає значення всіх сенсорів. В даному випадку ідентифікатором сенсора виступає модуль СЗК.

Для кодування даних восьми сенсорів (розрядність даних -10 біт) в якості модулів СЗК вибираємо наступні прості числа: $p1=1031$; $p2=1033$; $p3=1039$; $p4=1049$; $p5=1051$; $p6=1061$; $p7=1063$; $p8=1069$.

Відновлення даних відбувається за формулою:

$$x_i = X(\text{mod } p_i). \quad (2)$$

Розглянемо приклад кодування. Нехай сенсори генерують значення: $x_1=488$; $x_2=956$; $x_3=969$; $x_4=28$; $x_5=62$; $x_6=964$; $x_7=363$; $x_8=825$. Коефіцієнти k_i для вказаних модулів: $k_0=516$; $k_1=104$; $k_2=213123$; $k_3=946$; $k_4=886$; $k_5=502052$; $k_6=1123305632393$.

За формулою (1) знаходимо значення $X=1032780252932$.

Використовуючи формулу (2) знаходимо значення сенсорів: $x_1=1032780252932 \text{ mod } p_1=516$, $x_2=1032780252932 \text{ mod } p_2=956$ і т.д.

Виконавши декодування ми отримали значення які відповідають даним сенсорів, що підтверджує коректну роботу схеми кодування.

Перевагою запропонованого способу кодування є те, що передаються тільки дані сенсорів, тобто немає необхідності передавати ідентифікатори сенсорів. Подальші дослідження спрямовані на імплементацію та дослідження даного методу кодування у безпроводних сенсорних мережах.

Перелік посилань

1. Yatskiv V., Su Jun, Yatskiv N., Sachenko A., Osolinskiy O. Multilevel Method of data coding in WSN. // Proceeding of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications: IDAACS '2011, 2011. – P. 863 – 866
2. Yatskiv V., Yatskiv N., Sachenko A., Yatskiv S., Tsavolyk T. Adaptive Data Transmission Protocol for Wireless Sensor Networks Based on Residue Number System Correcting Codes. Proceedings of the 4th IEEE International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS (Lviv, Ukraine, 20-21 September, 2018), Lviv. 2018. P.131-136
3. Omondi A. Residue Number System: Theory and Implementation / A.Omond, B.Premkumar. Imperial College Press, 2007. – Vol. 2. – 296 p.
4. Wang, Yuke. Residue-to-binary converters based on new Chinese remainder theorems. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 2000, 47.3: 197-205

ДОДАТОК В

ПРЕЗЕНТАЦІЯ

СЛАЙД 1

Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерної інженерії та системного програмування

**СИСТЕМА ЗБОРУ ДАНИХ СЕНСОРІВ
НА ОСНОВІ БАГАТОРІВНЕВОЇ
СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ**

АВТОР РОБОТИ:
ФЕДОРИШИН О.М.

КЕРІВНИК РОБОТИ:
ЯЦКІВ В.В.

СЛАЙД 2

**МЕТА: ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ СИСТЕМИ ЗБОРУ
ДАНИХ СЕНСОРІВ НА ОСНОВІ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ**

**ОБ'ЄКТ ДОСЛІДЖЕННЯ: ПРОЦЕСИ КОДУВАННЯ У
СИСТЕМАХ ЗБОРУ ДАНИХ СЕНСОРІВ**

**ПРЕДМЕТ ДОСЛІДЖЕННЯ: МЕТОДИ ТА ЗАСОБИ
КОДУВАННЯ ТА ПЕРЕДАЧІ ДАНИХ У СЕНСОРНИХ МЕРЕЖАХ
НА ОСНОВІ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ**

Публікація: Федоришин О.М., Яцків В.В. Спосіб кодування даних сенсорів на основі системи залишкових класів. XIII ВСЕУКРАЇНСЬКА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ. АКТУАЛЬНІ ПРОБЛЕМИ КОМП'ЮТЕРНИХ НАУК АПКН–2021.

СЛАЙД 3

ВСТУП

Безпроводні технології передачі даних широко використовуються в різних галузях економіки, в тому числі в системах збору інформації про стан фізичного середовища, в задачах технічного та екологічного моніторингу та інших.

СЛАЙД 4

ОПИС ПРОБЛЕМИ

Неефективність використання пропускної здатності безпроводних сенсорних мереж у зв'язку з великою часткою службових даних при передачі повідомлень сенсорів, яка складає від 4 до 80% всіх даних

СЛАЙД 5

СУТЬ ПРОПОНОВАНОГО МЕТОДУ КОДУВАННЯ

- Вибір системи взаємно простих чисел значення яких більше за максимальне значення даних сенсора
- Дані сенсора - залишки по вибраній системі чисел
- Перетворення залишків у позиційну систему числення
- Отримання числа у позиційній системі числення яке відображає значення всіх сенсорів
- Відновлення даних

СЛАЙД 6

ПРИКЛАД КОДУВАННЯ ТА ДЕКОДУВАННЯ ДАНИХ ЧОТИРЬОХ СЕНСОРІВ

Подання даних в системі залишкових класів

Розглянемо систему залишкових класів з модулями

$$p_1, p_2, \dots, p_i, \dots, p_n$$

в якій число X представляється набором залишків по відповідних модулях

$$X = (x_1, x_2, \dots, x_i, \dots, x_n)$$

де $x_i = X \pmod{p_i}$

Діапазон перетворення $P = \prod_{i=1}^n p_i$

Модулі p_i вибираються з умови: $p_1 < p_2 < \dots < p_i \dots < p_n$

Наприклад, Нехай модулі: $p_1=3$; $p_2=5$; $p_3=7$; $X=10$

Тоді: $x_1=10 \pmod{3}=1$; $x_2=10 \pmod{5}=0$; $x_3=10 \pmod{7}=3$;

$$10 = (1, 0, 3)$$

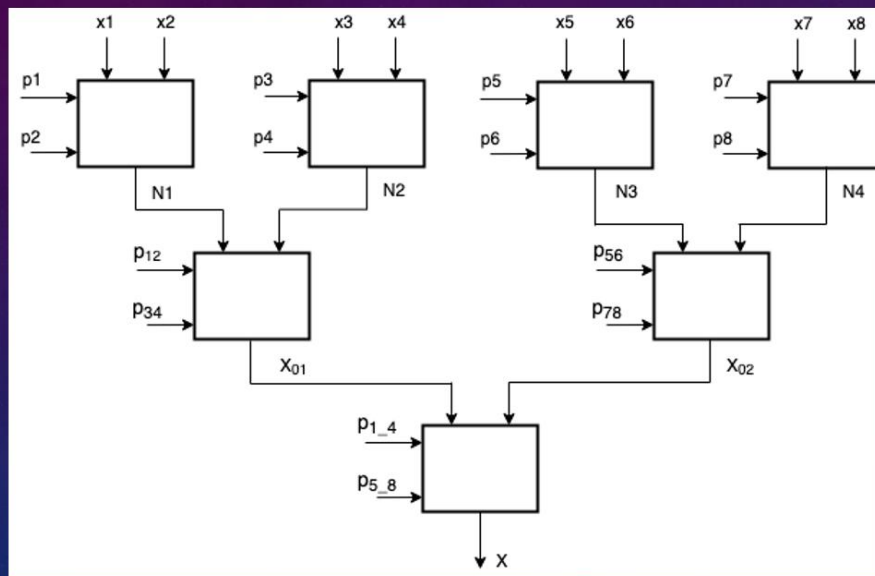
СЛАЙД 7

ЗАЛИШКИ ЧИСЕЛ ВІД 0 ДО 15 ЗА ВЗАЄМНО ПРОСТИМИ МОДУЛЯМИ

N	Відносно прості модулі		
	m1 = 2	m2 = 3	m3 = 5
0	0	0	0
1	1	1	1
2	0	2	2
3	1	0	3
4	0	1	4
5	1	2	0
6	0	0	1
7	1	1	2
8	0	2	3
9	1	0	4
10	0	1	0
11	1	2	1
12	0	0	2
13	1	1	3
14	0	2	4
15	1	0	0

СЛАЙД 8

СТРУКТУРНА СХЕМА КОДУВАННЯ ДАНИХ СЕНСОРІВ



СЛАЙД 9

ПРИКЛАД КОДУВАННЯ ТА ДЕКОДУВАННЯ ДАНИХ ЧОТИРЬОХ СЕНСОРІВ

Формули кодування даних сенсорів

$$N1 = x2 + p2 * ((k0 * (x1 - x2)) \% p1)$$

$$N2 = x4 + p4 * ((k1 * (x3 - x4)) \% p3)$$

$$X = N2 + p34 * ((k2 * (N1 - N2)) \% (p12))$$

де $p12 = p1 * p2$; $p34 = p3 * p4$; $k0 * p2 \% p1 = 1$; $k1 * p4 \% p3 = 1$; $k2 * p34 \% p12 = 1$;

СЛАЙД 10

ПРИКЛАД КОДУВАННЯ ТА ДЕКОДУВАННЯ ДАНИХ ЧОТИРЬОХ СЕНСОРІВ

Приклад кодування

Обрані модулі: $p1 = 1031$, $p2 = 1033$, $p3 = 1039$, $p4 = 1049$

Дані сенсорів: $x1 = 488$, $x2 = 956$, $x3 = 969$, $x4 = 28$

Закодоване значення: $X = 1032780252932$

Для отримання значення конкретного сенсора потрібно отримати залишок по відповідному модулю від закодованого значення: $x1 = X \bmod p1 = 488$ і т.д.

СЛАЙД 11

ПЕРЕВАГИ ПРОПОНОВАНОГО МЕТОДУ КОДУВАННЯ

- Передача виключно даних сенсорів
- Усунення необхідності передачі службової інформації
- Підвищення ефективності використання пропускнуої здатності безпроводних сенсорних мереж

СЛАЙД 12

РЕАЛІЗАЦІЯ СИСТЕМИ

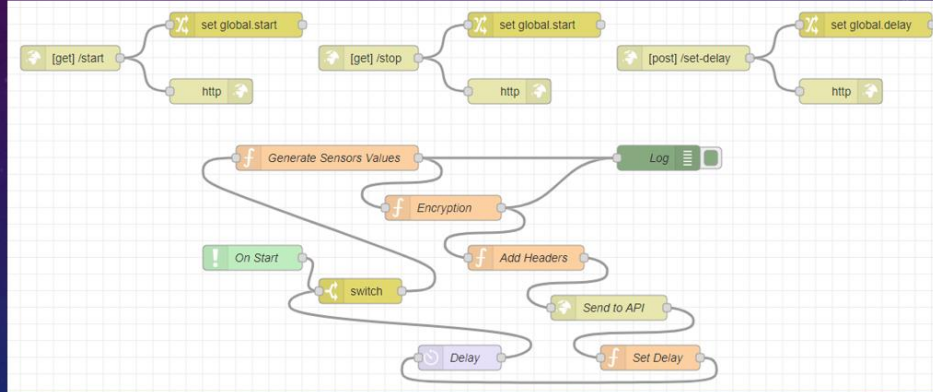
Система складається з трьох частин:

- Симуляція БСМ, створена за допомогою Node-RED
- Система збору та обробки даних - Web API на базі ASP.NET Core
- Клієнт-додаток для візуалізації даних, побудований за допомогою Angular

СЛАЙД 13

РЕАЛІЗАЦІЯ СИСТЕМИ

БСМ представлена наступним набором нодів



СЛАЙД 14

РЕАЛІЗАЦІЯ СИСТЕМИ

Візуальна частина має наступний вигляд

Value	Танки	БМ	Рашисти	БПЛА
148717155907	43	30	2	13
214254921538	1	31	7	15
67526742054	49	3	26	44
169695437656	13	17	30	24
406017135744	39	10	23	1
212530141042	25	8	32	5
994656922155	13	29	23	5
1077765469272	42	39	37	23

225 (+42)

167 (+39)

180 (+37)

130 (+23)

СЛАЙД 15

ВИСНОВОК

- Розроблено метод кодування та передачі даних сенсорів з підвищеною ефективністю пропускної здатності БСМ
- Створено систему з трьох частин, в якій застосовано розроблений метод кодування

СЛАЙД 16

ДЯКУЮ ЗА УВАГУ!

Ім'я користувача:
Кафедра КІ

ID перевірки:
1011239873

Дата перевірки:
18.05.2022 20:30:07 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
18.05.2022 20:30:23 EEST

ID користувача:
100005591

Назва документа: Федоришин Система збору даних сенсорів на основі багаторівневої системи залишкових ...

Кількість сторінок: 75 Кількість слів: 12104 Кількість символів: 93172 Розмір файлу: 1.95 MB ID файлу: 1011131030

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

4.47%

Схожість

Найбільша схожість: 1.25% з Інтернет-джерелом (https://uk.wikiaro.ru/wiki/Sensor_node)

3.7% Джерела з Інтернету

169

Сторінка 77

0.84% Джерела з Бібліотеки

48

Сторінка 78

0.2% Цитат

Цитати

1

Сторінка 79

Не знайдено жодних посилань

0%

Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

7

Підозріле форматування

18
сторінок

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 7%

ID: 103621 Название: Система збору даних сенсорів на основі багаторівневої системи залишкових класів Добавлено в БД: 2022-05-18 Авторы: Федоришин О.М. Руководители: Яцків В.В. Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	73846	645	468 (1%)	10 (2%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система збору даних сенсорів на основі багаторівневої системи залишкових класів

Автор: Федоришин Олександр Михайлович

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Василь Васильович Яцків

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 4.47% і адресується до 218 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІСч



В. В. Яцків

О. С. Савенко

Т. О. Говорущенко

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Федоринин Олександр Михайлович

Тема: Системи збору даних сенсорів на основі Бататорівської системи
застосування машин

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

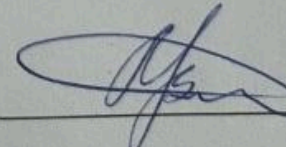
Кількість листів креслень 0 Кількість сторінок записки 19

1. Короткий зміст роботи та прийнятих рішень: у роботі запропоновано метод пошуку та передачі даних сенсорів
2. Висновок про відповідність роботи дипломному завданню: дипломна робота відповідає вимогам завдання
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: в результаті узагальнення методу пошуку було розроблено програму запису даних - віртуальну БСМ
4. Позитивні сторони роботи: в результаті виконання наукового дослідження було розроблено метод пошуку даних сенсорів. При аналізі існуючих методів передачі встановлено, що БСМ переважно використовує програмний запис даних. Розроблено метод передачі даних з використанням програмного запису.
5. Негативні сторони роботи: в світі не здійснено аналіз методів оптимізації процесу пошуку
6. Оцінка графічного оформлення та пояснювальної записки роботи: матеріал є структурованим у кожній формі та відображає повільніше виконання
7. Відгук про роботу в цілому: робота виконана на досить високому рівні
8. Інші зауваження: -
9. Оцінка дипломної роботи: робота заслуговує оцінки "добре"

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Мартинович Валерій Володимирович,
з.т.н., проф., зав. каф. АКІТ

"18" 05 2022 р.

 (підпис)