

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Навчально-науковий
інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)
Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

здобувача Осетрової Ангеліни Олексіївни
(ПІВ)

академічної групи 123-21-1
(шифр)

спеціальності 123 Комп'ютерна інженерія
(код і назва спеціальності)

за освітньо-професійною програмою 123 Комп'ютерна інженерія
(офіційна назва)

на тему “ІоТ-система термометрії елеватора з детальним опрацюванням під-
системи збирання даних температурного поля силосу”
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц.Ткаченко С.М.			
загального розділу	доц.Ткаченко С.М.			
спеціального розділу	доц.Ткаченко С.М.			
Рецензент				
Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро
2025

ЗАТВЕРДЖЕНО:
завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії
(повна назва)

Гнатушенко В.В.
(підпис) (прізвище, ініціали)

"25" січня 2025 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавр

здобувача Осетрової А.О. академічної групи 123-21-1
(прізвище та ініціали) (шифр)

спеціальності 123 «Комп'ютерна інженерія»

за освітньо-професійною програмою 123 «Комп'ютерна інженерія»
(офіційна назва)

на тему "IoT-система термометрії елеватора з детальним опрацюванням під-системи збирання даних температурного поля силосу"

затверджену наказом ректора НТУ «Дніпровська політехніка» від 05.05.2025 № 336-с

Розділ	Зміст	Термін виконання
Стан питання і постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел показати актуальність завдання, сформулювати мету та задачі виконання кваліфікаційної роботи	10.02.2025
Технічні вимоги до об'єкту вивчення	Сформулювати найменування й призначення IoT-системи, висунути технічні вимоги до неї	15.03.2025
Розробка апаратної частини	Виконати технічне проєктування апаратної частини IoT-комп'ютерної системи з необхідними інженерними розрахунками	20.04.2025
Програмування системи	Обґрунтувати технічні характеристики програми й розробити програму збору даних від цифрових датчиків температури	31.05.2025

Завдання видано _____
(підпис керівника)

доц. Ткаченко С.М.
(прізвище, ініціали)

Дата видачі 25.01.2025

Дата подання до екзаменаційної комісії _____ 10.06.2025

Прийнято до виконання _____

Осетрова А.О.

РЕФЕРАТ

Пояснювальна записка: 60 с., 19 рис., 3 табл., 1 дод., 29 джерел.

ТЕРМОМЕТРИЯ, ТЕРМОПІДВІС, РАДІОКАНАЛ, ЦИФРОВИЙ ДАТЧИК, СИЛОС, ТЕМПЕРАТУРНЕ ПОЛЕ.

Об'єкт професійної діяльності – підсистема збирання даних температурного поля силосу.

Мета роботи – розробити технічно-програмне рішення блоку збирання даних температурного поля силосу на основі обладнання Arduino, придатне для використання у складі IoT-системи термометрії зернових силосів елеватора.

Здійснено огляд існуючих технологій і побудову елеватора, який використовує металеві силоси. Проаналізовано існуючі технології вимірювання температури зерна під час зберігання, існуючі аналоги систем термометрії, які використовують термопідвіси з цифровими датчиками. Сформульовано технічні вимоги до підсистеми збирання даних температурного поля силосу, яка має виконувати наступні функції:

- опитування датчиків температури;
- зберігання адрес і підключень датчиків, а також термопідвісів;
- зберігання температурного поля датчиків силосу;
- опційне опитування адрес датчиків підключених термопідвісів;
- виконання алгоритму керування Підсистемою у складі IoT-системи;
- підтримка прикладного протоколу обміну з автоматизованим робочим місцем;
- підтримка каналного й фізичного рівнів мережі ModBus RTU;
- підтримка радіоканалу мережі ModBus RTU.

Розроблено технічне й програмне рішення, яке задовольняє поставленим вимогам. Результати роботи можуть бути використані для розробки автоматизованого робочого місця оператора термометрії зерна у складі IoT-системи термометрії.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	7
1 Стан питання і постановка задачі.....	8
1.1 Значення систем контролю температури для зберігання зерна...	8
1.2 Характеристика, структура, особливості, умови роботи об'єкту впровадження	9
1.2.1 Явище самозігрівання зерна на елеваторах	9
1.2.2 Огляд розташування технологічних споруд металевого елеватора	10
1.3 Огляд існуючих методів, технологій, аналогів систем термометрії	12
1.3.1 Методи вимірювання температури у силосах елеватора ...	12
1.3.2 Види первинних перетворювачів	14
1.3.3 Системи збору даних з температури від цифрових термопідвісів	17
1.4 Обґрунтування напрямку вирішення задачі розробки системи термометрії	23
1.5 Мета і задачі і роботи	24
2 Розробка блоку збирання даних температурного поля силосу.....	26
2.1 Технічні вимоги до об'єкту професійної діяльності	26
2.1.1 Найменування і призначення об'єкту професійної діяльності	26
2.1.2 Вимоги до структури і функціонування об'єкту професійної діяльності	26
2.1.3 Вимоги до показників призначення	27
2.1.4 Додаткові вимоги	27

2.2 Розробка апаратної частини	27
2.2.1 Розробка функціональної схеми автоматизації	27
2.1.2 Аналіз входів і виходів Підсистеми	28
2.1.3 Розрахунок потреб у електронній пам'яті для баз даних Підсистеми	30
2.1.4 Вибір елементної бази для Підсистеми	31
2.1.5 Розробка принципової схеми Підсистеми	36
3 Розробка програмного забезпечення системи	38
3.1 Призначення і сфера застосування програмного забезпечення ...	38
3.2 Обґрунтування технічних характеристик програми	38
3.2.1 Функції, виконувані програмою	38
3.2.2 Вимоги до функцій, виконуваних програмою	39
3.2.3 Обґрунтування вхідних і вихідних даних програми	40
3.2.4 Обґрунтування алгоритмів і методів, застосованих для реалізації програмного забезпечення	41
3.2.5 Технічні та програмні засоби, застосовані для реалізації програми	47
3.3 Опис розробленої програми	49
3.3.1 Загальна інформація	49
3.3.2 Функціональне призначення	48
3.3.3 Опис логічної структури програми	49
3.4 Використані технічні засоби	52
3.5 Виклик та завантаження	52
3.6 Вхідні й вихідні дані	52
Висновки	56
Перелік джерел посилання	57
Додаток А. Текст програми підсистеми збирання даних температурного поля силосу	61

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

GMSK	–	англ. Gaussian Minimum Shift Keying, Гауссова мінімальна маніпуляція зсувом;
IIoT	–	англ. Industrial Internet of Things, промисловий інтернет речей;
OPC	–	англ. Open Platform Communications, комунікації даних на основі відкритих платформ;
SCADA	–	англ. supervisory control and data acquisition, диспетчерське управління і збір даних;
SMA антена	–	англ. SubMiniature version A антена;
SPI	–	англ. Serial Peripheral Interface, послідовний периферійний інтерфейс;
USB	–	англ. Universal Serial Bus, універсальна послідовна шина;
АРМ	–	автоматизоване робоче місце;
АРМ ОТЗ	–	АРМ оператора термометрії зерна;
БД	–	база даних;
КТЗ	–	комплекс технічних засобів;
ПЗ	–	програмне забезпечення;
СУБД	–	система управління базою даних.

ВСТУП

Існуючі способи й технології тимчасового й тривалого зберігання зерна й зернопродуктів потребують здійснення систематичного контролю температури зерна під час його зберігання, щоб запобігти псуванню. Тому засоби контролю температури зерна є обов'язковими. Необхідна постійна присутність засобів вимірювання у зернових масах у силосах, що зумовлює актуальність розробки й побудови профільних систем термометрії.

Мета представленої роботи кваліфікаційної роботи – розробити технічно-програмне рішення блоку збирання даних температурного поля силосу на основі обладнання Arduino, придатне для використання у складі IoT-системи термометрії зернових силосів елеватора.

Обрана тема кваліфікаційної роботи актуальна для зернових елеваторів, побудованих на основі сучасних металевих силосів, які можуть бути територіально віддалені від робочого місця оператора термометрії, яке буде розташоване у такому випадку у лабораторії якості зерна. Результати роботи можуть бути також використані для термометрії складів підлогового зберігання й залізобетонних силосних корпусів тривалого зберігання зерна.

Вихідними даними для представленої роботи є матеріали передатестаційної практики, присвячені огляду існуючих технологій зберігання зерна та існуючих аналогів систем термометрії, а також матеріали XVII Міжнародна конференція «Проблеми використання інформаційних технологій в освіті, науці та промисловості». 24 листопада 2022 року м. Дніпро.

1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАДАЧІ

1.1 Значення систем контролю температури для зберігання зерна

Питання термометрії у сфері зберігання і переробки зерна виникає під час тимчасового й тривалого зберігання зернових[1], зернопродуктів чи шроту. Для зберігання застосовуються споруди[2]:

- криті склади, де зерно насипано певним шаром на горизонтальну або похилу підлогу;
- елеватори різної місткості;
- силосні башти і металеві бункери;
- бетонні склади;
- інші комори і будівлі, забезпечуюся нормативні умови для зберігання зернових культур.

Вказані способи потребують здійснення систематичного контролю температури зерна під час зберігання, щоб запобігти його псуванню через явища самозігрівання й злежування [3]. Альтернативна технологія, зберігання зерна у герметичних полімерних рукавах[4], контролю температури зерна не передбачає, але вона використовується лише як тимчасовий захід, який потребує додаткових попередніх операцій з переробки зерна, що не завжди можливо на призначеному місці зберігання, а також вимагає використання значних підготовлених площ, які охороняються.

Таким чином, для згаданих вище споруд, обов'язковими є засоби контролю температури зерна. З урахуванням того, що самозігрівання зерна може відбуватись тривалий час і його джерела можуть знаходитись глибоко у зернових масах[3], необхідна постійна присутність засобів вимірювання у таких масах. Ця обставина приводить до актуальності розробки й побудови не лише засобів, але й систем термометрії, зокрема, для силосів елеватору.

1.2 Характеристика, структура, особливості, умови роботи об'єкту впровадження

1.2.1 Явище самозігрівання зерна на елеваторах

Розглянемо явище самозігрівання на елеваторі з силосами з плоским та конічним дном. Самозігрівання пов'язане з випаровуванням вологи, яка виділяється під час природніх біологічних процесів у живому зерні. Розрізняють такі типи самозігрівання: гніздове, пошарове (пластове) і суцільне[3].

Гніздове самозігрівання може відбуватися в будь-якій частині зернової маси з причин[3]:

- зволоження частини зернової маси через протікання або недостатню гідроізоляцію стін сховища;
- розміщення в сховищі зерна з різною вологістю та температурою;
- утворення в зерновій масі ділянок із підвищеним вмістом домішок та пилу через розміщення разом різнорідного за вмістом домішок зерна;
- накопичення шкідників в окремій частині насипу та активізація їх життєдіяльності.

Пошарове (пластове) самозігрівання виникає в насипу у вигляді горизонтального або вертикального шару за зберігання у складах і силосах. Залежно від того, на якій ділянці насипу утворюється такий шар, розрізняють самозігрівання: верхове; низове; вертикально-пластове. Пошарове самозігрівання виникає недалеко від поверхні насипу або в шарах, які близько розташовані від підлоги і стін сховища[3]. Явище пов'язане з конденсатом, який утворюється внаслідок перепаду температур різних у шарах зерна через сезонні перепади температури навколишнього середовища. Так, низове зігрівання зерна, показане на рисунку 1.1, виникає восени під час завантаження вологого зерна, верхове – пізньої осені й ранньої весни під час завантаження теплого зерна, вертикальне – можливе у будь-яку пору року через різницю температур між зерном під стінкою силосу і в глибині маси. Вертикальне самозігрівання більш притамане металевим силосам, ніж бетонним.

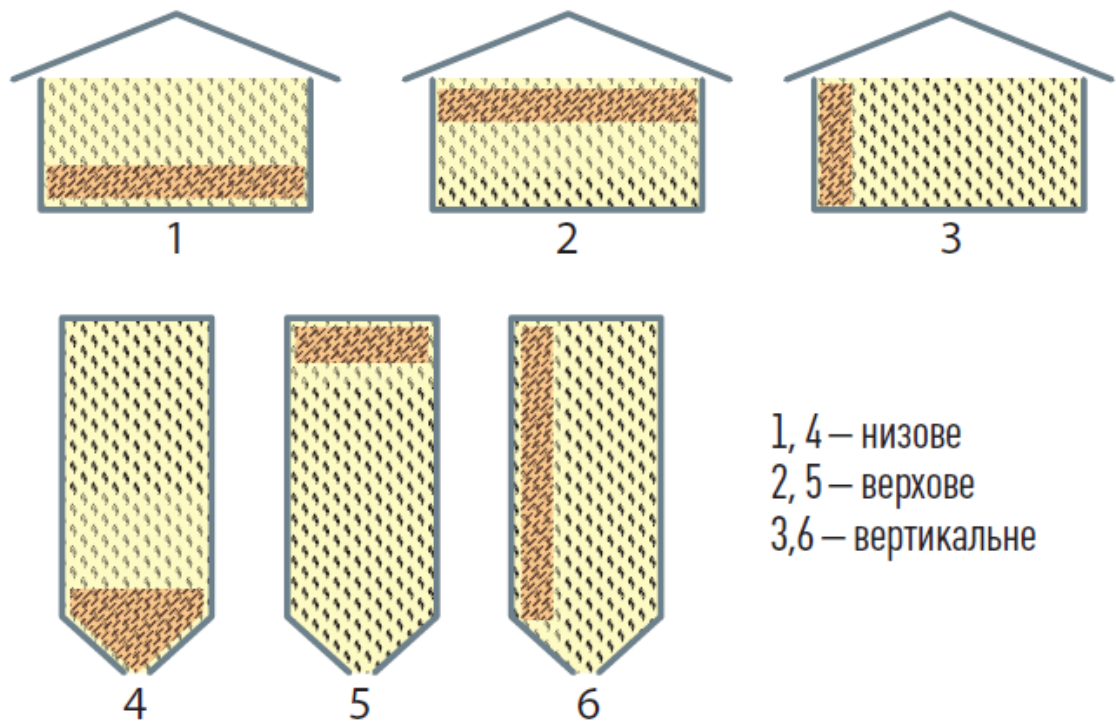


Рисунок 1.1 – Види пошарового самозігрівання зерна у силосах

Надалі, в якості об'єкту впровадження, розглянемо елеватор, споруджений на основі металевих силосів, оскільки саме такий тип є найбільш масовим у світі України[5,6,7].

1.2.2 Огляд розташування технологічних споруд металевого елеватора

В якості ілюстрації, розглянемо фрагмент деякого елеватору з металевими силосами, представлений на рисунку 1.2. Його було взято з сайту elevatorist.com[7].

На рисунку видно залізницю з тепловозами, які стоять поряд з платформою для відбору лабораторних проб (це видно по наявності щупів-відбірників на платформі). Таким чином на об'єкті впровадження є лабораторія якості й ділянка залізничного вивантаження із завальною ямою. На задньому плані є 6 силосів зберігання з плоским дном, з'єднаних надсилосною галереєю зверху. Надсилосна галерея містить транспортер та інше обладнан-

ня завантаження, а також лінію освітлення. Під силосами є підсилосна галерея з відповідним обладнанням розвантаження.



Рисунок 1.2 – Зовнішній вигляд ділянки елеватора[7] із силосами зберігання. Ілюстрація

За силосами зберігання на рисунку 1.2 розташована відкрита силосна башта з норіями, поряд з баштою – сушарка зерна модульного типу, бункер завантаження автомобілів, бункер сміття. Наявність бункера сміття свідчить про наявність, як мінімум, однієї зерноочисної машини на елеваторі. На ілюстрації не видно жодного силосу з конічним дном, але, оскільки такі силоси використовуються як оперативні, наприклад, для сушарки чи непідготовленого зерна, то вони є. Заслужують уваги дві повітряні транспортні галереї, що відходять вправо й уліво від силосної башти. Це значить, що є ще споруди і обладнання, недоступні для огляду. Зокрема, галерея, що відходить вправо за межі ілюстрації показує, що за залізничними коліями знаходиться ще група силосів тривалого зберігання. Де знаходиться лабораторія якості, а

також де знаходиться робоче місце для контролю температури зерна з рисунку 1.2 з'ясувати не вдається. Але зрозуміло, що за будь якого розташування лабораторії й робочого місця, відстані до місць вимірювання значні, у межах кількох сотень метрів, що є приводом розглянути необхідність застосування засобів дистанційної передачі даних. Розглянемо технологію та існуючі аналоги систем вимірювання температури у силосах елеватора.

1.3 Огляд існуючих методів, технологій, аналогів систем термометрії

1.3.1 Методи вимірювання температури у силосах елеватора

Згідно ДСТУ ISO 4112:2014 Зернові та бобові. Вимірювання температури зерна[8], точки вимірювання повинні бути розташовані на відстані 3-4 м одна від одної по горизонталі й близько 2...3 м по вертикалі. На практиці, для забезпечення необхідної точності, по вертикалі є рекомендація вимірювати температуру зерна частіше, наприклад, через кожний метр по вертикалі[9], що у даному випадку пояснюється точністю методу. Щодо відстаней між точками вимірювання температури по горизонталі, то тут у виробників загальний підхід досить приблизний. Для подальшого аналізу скористуємось схемою розташування підвісок з датчиками температури, показаною на рисунку 1.3[10]:

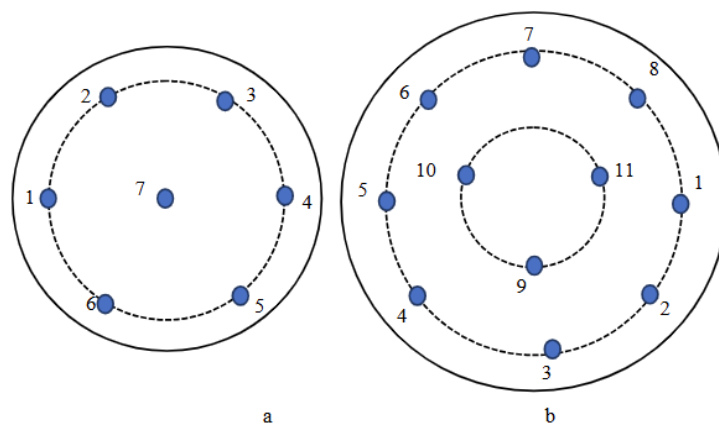


Рисунок 1.3 – Схема розташування термопідвісок у силосах діаметром 19,9 м (a) та 22,2 м (b), представлена у роботі [10]

З іншого боку, у роботі [11], обґрунтовано теоретичну кількість точок вимірювання у одному тернопідвісі, яка залежить від висоти й конструкції силоса і може досягати 35 із розрахунком, що точки вимірювання розташовані на відстанях 1,3...2 м одна від одної. Об'єднавши дані з робіт [10] і [11], можна стверджувати, що з використанням силосу з плоским дном діаметру 22,2 м кількість точок вимірювання не перевищить 385 на один силос з 11 тернопідвісами. Іншими словами, під час вимірювання температур лаборант повинен отримати не більше 385 записів від одного силоса. Але є й більш складні випадки. Наприклад, у селі Червонозаводське Полтавської області у силосах використовується на 24 тернопідвіси по 17 датчиків[11], а це вже 408 точок на силос. Для подальших розрахунків приймемо, що силос з плоским дном діаметром близько 22 метрів може мати до 410 точок вимірювання. На рисунку 1.1 силосів 7, що потребує, відповідно, від лаборанта отримання і обробки даних від 2 870, що представляє собою складну задачу навіть без урахування територіальної розподіленості об'єкту. Таким чином, виникає задача побудови системи термометрії, яка має виконувати дистанційний збір, обробку й збереження технологічних даних.

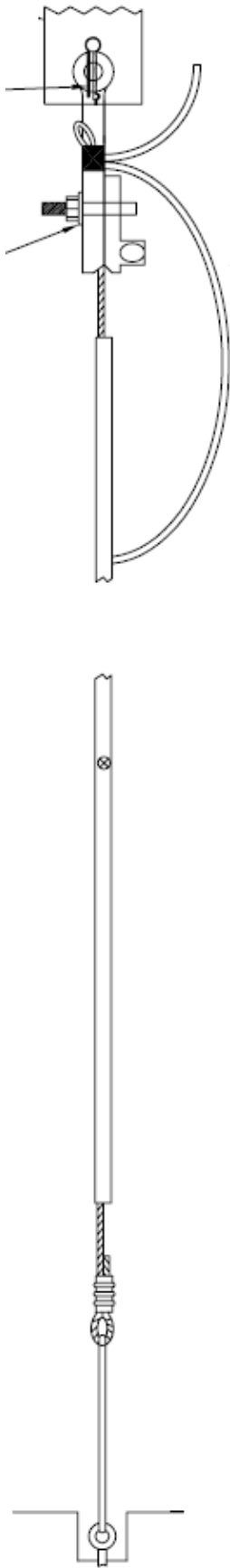
Стосовно періодичності вимірювання температурного поля силосів з зерном, то вона залежить від вологості культури, що зберігається [9]. Ця залежність показана у таблиці 1.1. З таблиці Очевидно, що мова не йде про отримання даних у реальному режимі часу, просто даних з температурного поля дуже багато і їх потрібно упорядковувати по номеру чи дентифікатору силоса та по просторовим координатам всередині силосу так, щоб потім було можливо відбудувати просторову модель розподілу температур і розвиток теплових процесів у часі. Оскільки задача вимірювання температурного поля зерна у силосі не є новою, необхідно розглянути вже існуючі її рішення.

Таблиця 1.1 – Періодичність вимірювання температури зерна, що зберігається

Перші 3 місяці			
Зерно	Періодичність виміру		
Сухе та середньої сухості	Раз на два дні		
Сире та вологе	Кожен день		
Після 3-х місяців			
Зерно	Температура		
	>10°C	від 0 до 10°C	<0°C
Сухе та середньої сухості	2 рази на 10 днів	1 раз на 10 днів	
Вологе	1 раз на 2 дні	2 рази на 10 днів	1 раз на 10 днів
Сире	Кожен день	1 раз на 2 дні	2 рази на 10 днів

1.3.2 Види первинних перетворювачів

Вимірювання температур всередині зернового силоса, а часто і сховища підлогового зберігання здійснюється за допомогою термошупів, термоштанг чи термopідвісів. Термошупи використовуються для вимірювання температури зерна вручну[9] у випадках, коли зерна зберігається небагато. У межах поставленої задачі вони далі не розглядаються. Щодо термоштанг – це жорсткі конструкції, виконані на основі труби. Така технологія певний час мала місце для залізобетонних силосів, оскільки всередині силоса не передбачалось виконувати штатні роботи. Термоштанги були більш дешевим рішенням у порівнянні з гнучкими металевими термopідвісами з аналоговими



датчиками, тому вони були бажані на складах насіння соняшника чи шроту через нездатність накопичувати статичний заряд, який міг би дати іскру. Поява іскри у сховищі шроту чи соняшника небезпечна, бо може призвести до вибуху метану, який виділяє соняшник та його відходи. На сьогодні жорсткі термоштанги використовуються в основному для вимірювання температури в резервуарах з паливом на сховищах відповідного профілю. Розглянемо технологію термopідвісів. Термopідвіс – це гнучка конструкція, яка підвішується зверху. У випадку металевих силосів на рисунку 1.1 – спускається через технологічні отвори в даху силоса. Завдяки гнучкості їх можна скручувати, транспортувати, монтувати й демонтувати, обслуговувати набагато простіше ніж термоштанги. На рисунку 1.4 показано термopідвіс, який спускається крізь дах силоса й кріпиться знизу до спеціального кільця струною[12]. Струна необхідна, щоб термopідвіс не відхилився від потрібної зони контролю температури під час завантаження чи вивантаження зерна. Струна перерізається вручну кожний раз, перед очисткою силоса з плоским дном, щоб дати змогу пройти під термopідвісами очисному шнеку. Сам термopідвіс виконується у вигляді кабель-тросу з оболонкою з ПВХ, у якій протягується лінія з датчиками.

Рисунок 1.4 – Термopідвіс у силосі

Іскробезпечний варіант більш дорогий – це металевий трос, у який вплетена лінія з датчиками. В якості датчиків у термopідвісі використовують термометри опору, термopари або цифрові датчики температури. Термометри опору на сьогодні вважаються застарілою технологією, хоча іноді використовуються.

Їх недоліки: вони не вміщуються в типові кабель-троси і вимагають спеціального металевого конструктиву, що здорожчує та додає ваги термopідвісу; У процесі експлуатації під дією ваги зерна термометри опору деформуються, що вимагає переналагодження й перекалібрування. Термopари випускають в основному виробники Канади й США[12]. На ринку України вони вважаються морально застарілими, оскільки також потребують калібрування і отримання інформації від такої підвіски досить складне, хоча проблема може бути вирішена штатними промисловими засобами[13]. Найбільш масова технологія вимірювання в Україні – використання термopідвісів на цифрових датчиках DS18B20[14] від Dallas Semiconductor.

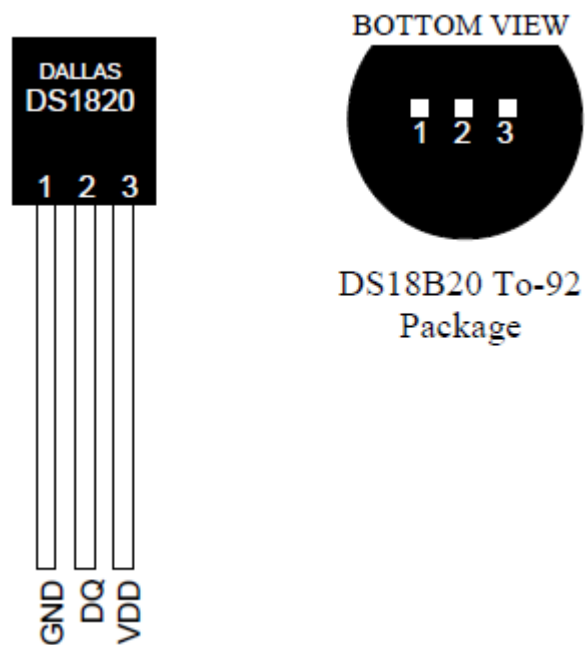


Рисунок 1.4 – Зовнішній вигляд датчика DS18B20[14]

Переваги: не потрібно калібрувати, точність вимірювань з часом не змінюється, мініатюрність, цифровий зв'язок через шину 1-Wire, що зменшує кількість провідників у кабель-каналі й спрощує ремонт.

Таким чином, для вирішення задачі збору даних з термометрії зерна в елеваторі силосного типу обираємо термопідвіс на цифрових датчиках. Розглянемо способи отримання даних з обраними термопідвісами й передачі їх у лабораторію на АРМ термометрії.

1.3.3 Системи збору даних з температури від цифрових термопідвісів

Якщо відразу відмежуватись від розгляду переносних вимірювальних щупів, від переносних приладів вимірювання, які тимчасово підключаються до термопідвісок, то ринок України достатньо насичений системами термометрії, що використовують цифрові термопідвіси [15–19]. Вони мають схожу архітектуру й принципи роботи, але є свої відмінності. Розглянемо термометрію зерна від компанії Технотест, рисунок 1.5 [16]:



Рисунок 1.5 – Архітектура системи термометрії від компанії Технотест[16]

Згідно рисунку 1.5, кожний термopідвіс підключається до власного контролера, котрі мають дротовий зв'язок зі збірковим контролером. Збірковий контролер, у свою чергу підключається до АРМ оператора термометрії зерна, далі АРМ ОТЗ. Кожний термopідвіс має власний контролер, розміщений у голівці термopідвісу. Недоліки підходу очевидні:

- голівки будуть підвішені під дахом силосу. Це збільшує механічне навантаження на кріплення підвісу, особливо в умовах великого силосу з широкою конусністю, де не лише термopідвіс на великих радіусах схеми розташування, але і його голівка буде засипана масою зерна;

- обслуговування контролера у голівці термopідвісу можливе лише із залученням фахівців з промислового альпінізму. Початковий монтаж контролера й термopідвісу або на етапі будівництва силосу, або з використанням промислового альпінізму;

- постає питання організації живлення контролера у голівці. Це можливо від окремого джерела постійного струму, і тоді живлення може надходити в тому ж кабелі, що й інтерфейс зв'язку, або від силової мережі надсилосної галереї. У такому випадку, скоріш за все, необхідні додаткові кабельні лінії до голівок термopідвісів;

- наявність центрального збіркового контролера призводить до додаткових ліній зв'язку, що здорожує проєкт;

- виробник використовує власні технічні рішення і при цьому не надає ніякої інформації про сумісність технічних рішень зі стандартними промисловими засобами, інтерфейсами, протоколами.

Переваги:

- є можливість використання бездротових засобів зв'язку. Могло б усунути необхідність у додаткових лініях зв'язку, але це опція, яка потребує уточнення у документації: які саме засоби можуть бути використані;

- є засоби дистанційного вмикання вентиляторів охолодження. Але рішення про вмикання цих вентиляторів приймається технологом на основі даних про температуру й вологість зерна, про температуру й вологість зовніш-

нього повітря, про точку температуру роси, яка може настати у бункері. Оскільки інформації про відповідні технічні й програмні засоби не надається, можна вважати, що дистанційне вмикання вентиляторів – не більше ніж опція ручного вмикання вентиляторів.

Схоже рішення запропоновано компанією IGT[17]:



Рисунок 1.6 – Архітектура системи термометрії від компанії IGT [16]

З рисунка 1.5. видно, що тут виробник не зміг відійти від ідеї використання голівок термopідвісок, хоча їх габарити більш компактні. Це можливе тому, що загальний контролер збору даних від усіх термopідвісів одного силосу розміщується у надсилосній галереї[17]. Звісно, таких контролерів може бути декілька. Також з'явилися додаткові засоби контролю вологості повітря над силосом та засоби контролю параметрів зовнішньої атмосфери, що може полегшити прийняття рішення про можливість і необхідність ввімкнення вентилятора охолодження. Додатково з АРМ ОТЗ з'явився хмарний сервіс. Ця опція надає дистанційний доступ до технологічних параметрів відповідальним особам і, можливо, клієнтам елеватора.

Недоліки рішення:

- голівка термопідвісу у силосному корпусі, що виробник намагається нівелювати, пропонуючи монтаж додаткових технологічних люків у даху силоса;

- вимірювання вологості над зерном само по собі не дозволяє у повній мірі оцінити наслідки закачування зовнішнього повітря знизу, оскільки датчик вимірює вологість повітря, що до цього або не було, або вже залишило зернову масу. Що насправді зараз відбувається у зерновій масі, він не показує;

- виробник використовує власні технічні рішення і не надає інформації про сумісність зі стандартними промисловими засобами;

- немає бездротових засобів зв'язку;

- немає достовірних даних, що задача автоматизації підтримки прийняття рішень з вентилування силосу вирішена, хоча технічні засоби наявні.

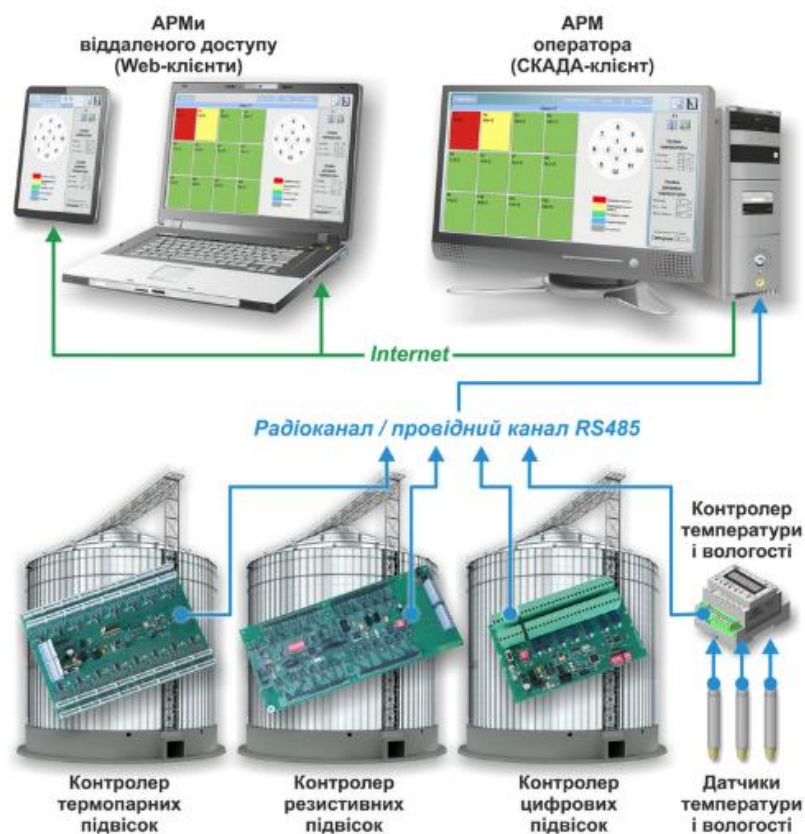


Рисунок 1.7 – Архітектура системи термометрії від компанії IGT [17]

Досить потужне рішення пропонує компанія Інновинпром [18]. Воно показане на рисунку 1.7. Тут видно, що задача термометрії вирішуються також і для термопар, і для термометрів опору. Технології, хоч і застарілі, але все ще підтримуються більшістю вітчизняних виробників. Варто зазначити, що у цьому рішенні надається доступ до АРМ ОТЗ і web-клієнтам, але без використання хмарних технологій. Також є окремий модуль вимірювання температури й вологості зовнішнього повітря, але відразу варто звернути увагу, що досить крупна компанія зі значним досвідом не заявляє про автоматизацію підтримки прийняття рішень з вентилявання й не намагається вимірювати вологу повітря над зерном.

Цифрові термопідвіси виробника [17] не містять голівок. Цей конструктив призводить до зменшення механічного навантаження на термопідвіс, а також дозволяє монтувати їх і обслуговувати через герметизовані технологічні отвори на даху елеватора діаметром 16 мм. Звичайно, за такого підходу контролери термопідвісів будуть розміщені в надсилосній галереї, де доступ до них більш легкий. Ще одна перевага системи – можливість використовувати стандартний провідний інтерфейс RS-485 або радіоканал.

Недоліки:

- виробник використовує контролери власного виробництва і не надає інформації про протокол передачі даних від контролерів до АРМ. Хоча можна припустити, що це може бути ModBus. Принаймні, технічних перешкод до його реалізації немає.

Асоціація підприємств «Ельдорадо» [15] має не менший досвід у розробці й впровадженні систем термометрії, ніж Інновинпром. Цифрові термопідвіски від «Ельдорадо» також не використовують голівку і мають ті ж самі переваги через вдалу конструкцію. Додатково, термопідвіси виробника [15] мають засоби зняття статичного заряду, який виникає під час завантажувально-розвантажувальних процесів і може призвести до виходу з ладу обладнання або вибухів зернового пилу чи метану. Також є спеціальні термопідвіси для іскробезпечних середовищ, виконані у вигляді металевих тросів.

Цей виробник прямо заявляє про використання промислової мережі ModBus RTU з можливістю передачі даних через радіоканал.

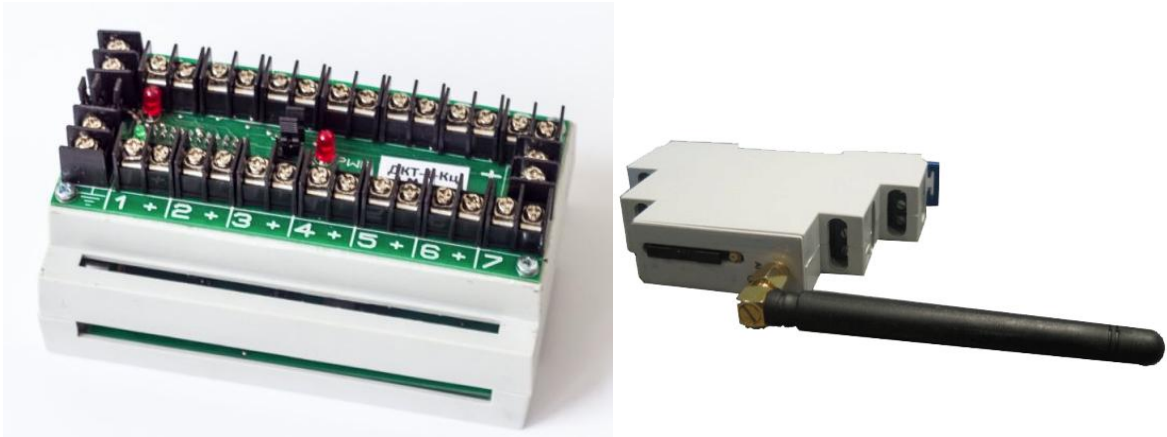


Рисунок 1.8 – Контролер збору даних від цифрових термопідвісок та модуль бездротової передачі даних від асоціації «Ельдорадо» [15]

Основний недолік рішень [15-19] – використання власних технічних рішень. Це непоганий і вигідний виробнику підхід в умовах стабільного ринку й невисокої конкуренції, тим більше, що крупні виробники засобів промислової автоматизації не поспішають зайняти нішу систем термометрії зерна. Але в умовах війни з росією ринок не є стабільним і питання виробництва відповідних технічних засобів може бути складним. З іншого боку, розвиток елементної бази дешевих контролерів, придатних, також і для застосування у промисловій сфері надає можливості розробки систем термометрії на більш поширеній та стандартизованій елементній базі [11]. Робота [11] показує можливості елементної бази Arduino для створення систем термометрії і надає ряд кількісних характеристик, які зумовлюють вимоги до технічного рішення задачі. Це дозволяє визначитися з напрямом напрямку вирішення задачі.

1.4 Обґрунтування напрямку вирішення задачі розробки системи термометрії

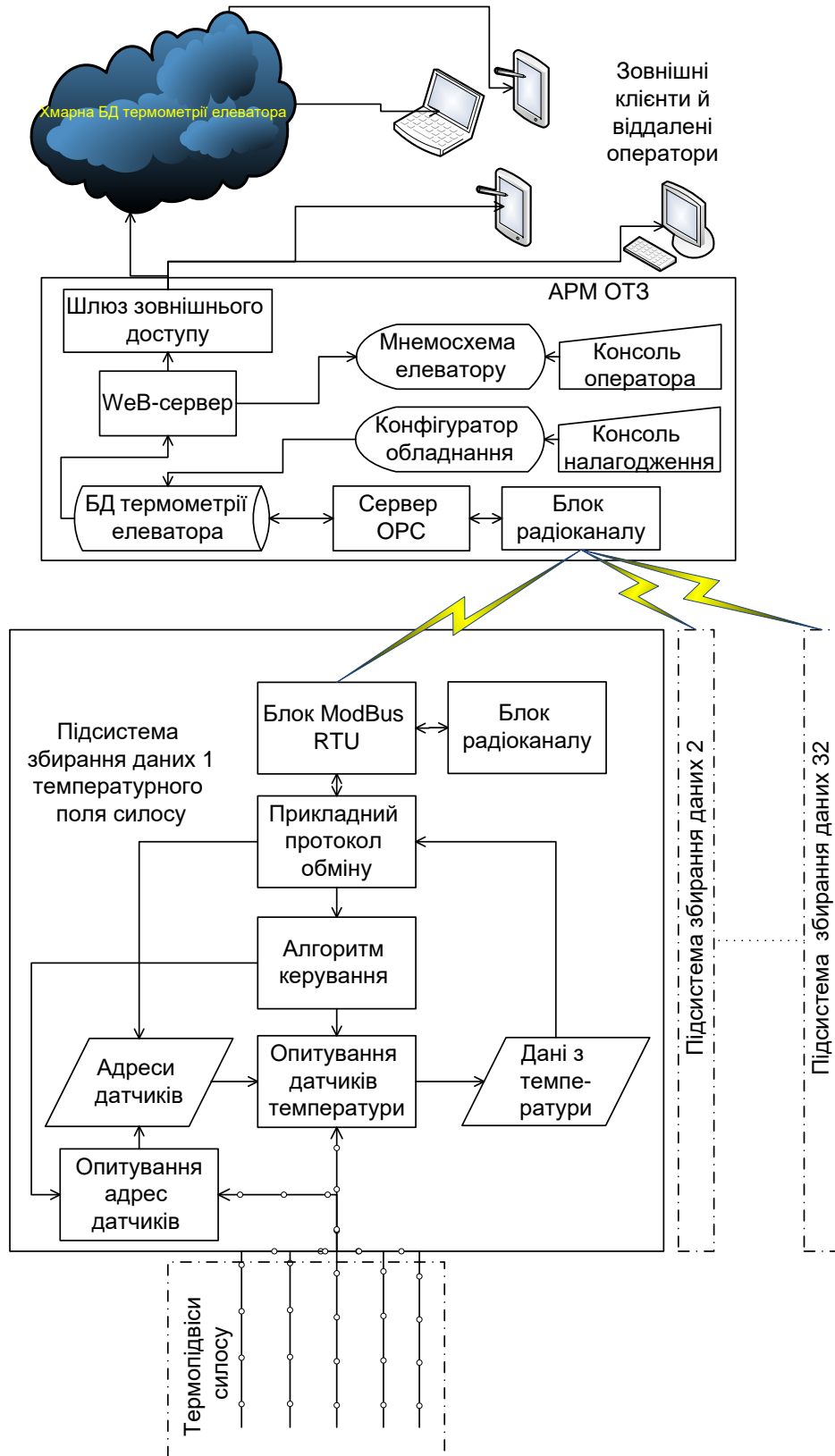


Рисунок 1.9 – IoT-система термометрії елеватора. Схема функціональної структури

Система термометрії зерна на елеваторі може бути побудована як ІоТ-система, що дозволить застосовувати хмарні, туманні та web-технології. Реалізація структурних елементів і обладнання системи можливе із використанням готової поширеної елементної бази Arduino, як показано у роботі [11].

В цілому, пропонується функціональна структура системи термометрії, представлена на рисунку 1.9.

Наразі варіанти вибору шляхів реалізації АРМ ОТЗ можуть бути різні: використання ОРС-серверів окремо чи у складі SCADA-систем, використання окремих баз даних зі своєю СУБД чи БД SCADA, web-технологій чи хмарних. Всі ці шляхи обмежуються задачею детальної розробки блоків збирання даних температурного поля силосу. Ідея роботи полягає у тому, що збір даних з температурного поля силосів зернового елеватора, які надходять від цифрових термопідвісів можлива шляхом застосування технічно-програмних засобів на основі контролерів Arduino. Таким чином, кваліфікаційна робота надалі присвячується вирішенню задачі у напрямку детальної розробки підсистеми збирання даних температурного поля силосу, далі Підсистеми.

1.5 Мета і задачі і роботи

Оскільки кваліфікаційна робота присвячується вирішенню задачі у напрямку детальної розробки Підсистеми, мета роботи – розробити технічно-програмне рішення блоку збирання даних температурного поля силосу на основі обладнання Arduino, придатне для використання у складі ІоТ-системи термометрії зернових силосів елеватора. Для досягнення поставленої мети необхідно виконати наступні задачі:

- на основі виконаного аналізу об'єкту впровадження, методів вимірювання, технологій і існуючих систем термометрії сформулювати технічні вимоги до Підсистеми як базової складової ІоТ-системи термометрії.

- провести аналіз вхідних і вихідних сигналів блоку Підсистеми та обґрунтувати обладнання, що має бути використане технічного рішення блоку.

Провести інженерний розрахунок джерела живлення блоку;

- розробити принципову схему Підсистеми;

- провести аналіз вхідних і вихідних даних, сформулювати алгоритми роботи ПЗ Підсистеми;

- на основі розроблених алгоритмів створити програмне рішення, яке повинне використовуватись із технічною реалізацією Підсистеми.

2 РОЗРОБКА БЛОКУ ЗБИРАННЯ ДАНИХ ТЕМПЕРАТУРНОГО ПОЛЯ СИЛОСУ

2.1 Технічні вимоги до об'єкту професійної діяльності

2.1.1 Найменування і призначення об'єкту професійної діяльності

Підсистема збирання даних температурного поля силосу, далі Підсистема, призначена для роботи у складі ІоТ-системи термометрії елеватора у галузі переробки й зберігання зерна.

2.1.2 Вимоги до структури і функціонування об'єкту професійної діяльності

Підсистема включає наступні функціональні складові:

- опитування датчиків температури. Апаратно-програмний блок для отримання даних від датчиків цифрових термопідвісів;
- адреси датчиків. БД адрес і підключень датчиків і термопідвісів;
- дані з температури. БД температурного поля датчиків Підсистеми;
- опитування адрес датчиків. Опційний програмний блок отримання адрес датчиків підключених термопідвісів під час обслуговування, якщо такі не отримані від АРМ ОТЗ;
- алгоритм керування. Основний алгоритм керування Підсистемою;
- прикладний протокол обміну. Програмний блок керування потоками даних з адресами датчиків та їх температурами;
- блок ModBus RTU. Апаратно-програмний блок підтримки каналного рівня мережі ModBus RTU;
- блок радіоканалу. Апаратний блок підтримки фізичного рівня мережі ModBus RTU.

Підсистема має наступні взаємозв'язки:

- працює у складі ІоТ-система термометрії елеватора;
- здійснює обмін даними з термопідвісами силосу;
- здійснює обмін з даними з АРМ ОТЗ.

Підсистема повинна працювати без збоїв в автоматичному режимі не менше 2 годин на добу.

2.1.3 Вимоги до показників призначення

Кількість термopідвісів, підключених до однієї Підсистеми: до 12.

Кількість датчиків на термopідвіс: до 35.

Довжина термopідвісу разом з комутаційним кабелем: до 100 м.

Лінія зв'язку термopідвісу: 1-wire із виділеним живленням.

Кількість підсистем, одночасно підключених до АРМ ОТЗ: до 32.

Відстань між АРМ ОТЗ та Підсистемами до 1000 м.

2.1.4 Додаткові вимоги

Підсистема повинна працювати в умовах під відкритим небом.

2.2 Розробка апаратної частини

2.2.1 Розробка функціональної схеми автоматизації

Підсистеми термометрії розташовуються у надсилосній галереї по одній на силос із плоским днищем, якщо кількість термopідвісів у цьому силосі не перевищує 12 (це типовий варіант). Якщо передбачається встановлення більше, ніж 12 термopідвісок на силос, то може бути використано 2 і більше підсистем на один силос. І навпаки, якщо дозволяє відстань, може бути використана одна підсистема на групу силосів, наприклад, оперативних з конусним днищем. Підсистеми термометрії у складі ПоТ-система термометрії елеватора взаємодіють через радіоканал з АРМ ОТЗ.

Функціональна схема автоматизації ПоТ-системи термометрії елеватора буде виглядати, як показано на рисунку 2.1.

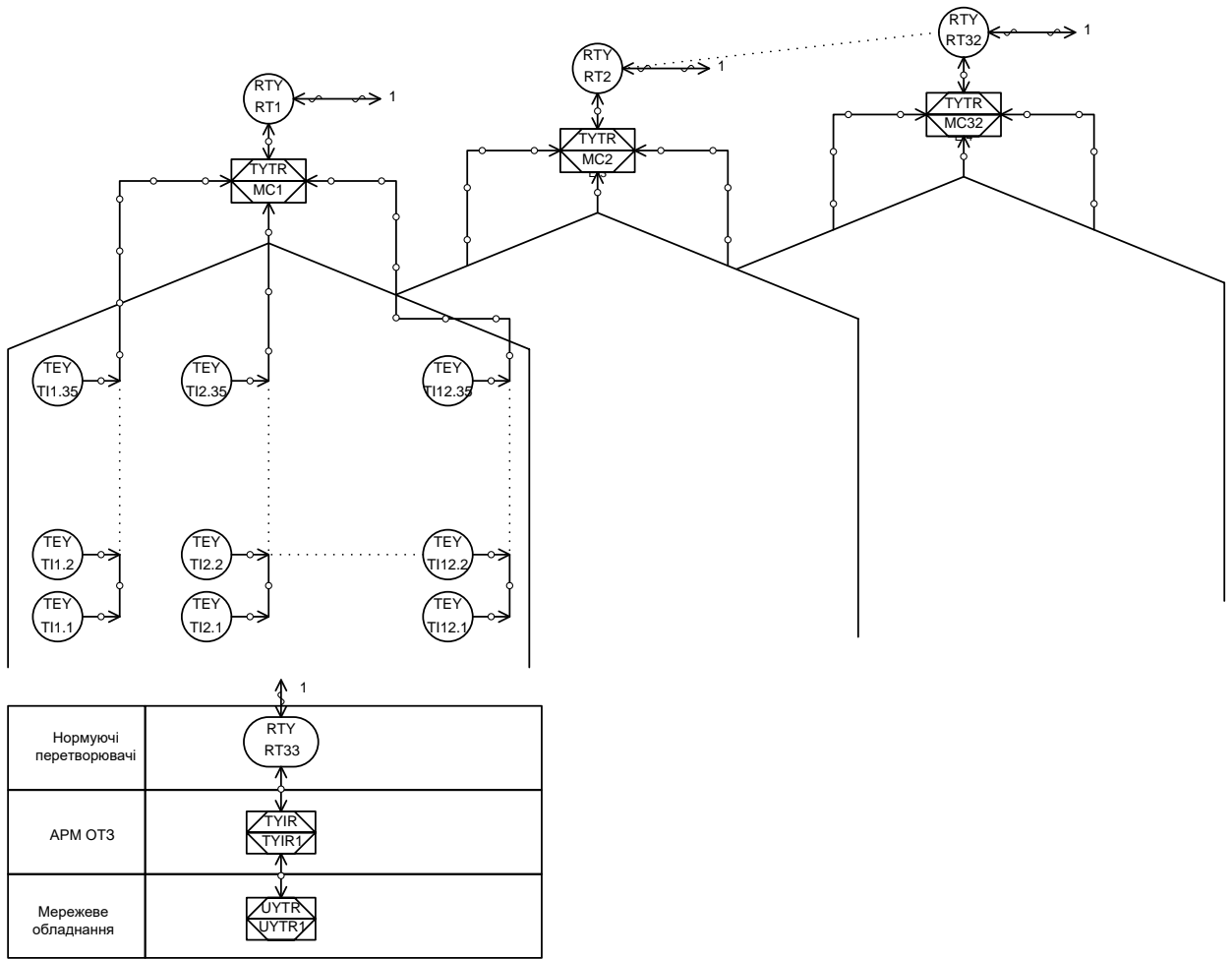


Рисунок 2.1 – IoT-система термометрії елеватора. Функціональна схема автоматизації

Представлена функціональна схема атоматизації, технічні вимоги та аналіз процесу отримання даних дозволяють перейти до аналізу входів і виходів Підсистеми.

2.2.2 Аналіз входів і виходів Підсистеми

Для аналізу входів і виходів проведемо класифікацію сигналів за допомогою записів в таблиці 2.1.

Таблиця 2.1 – Перелік вхідних і вихідних сигналів. Підсистема збирання даних температурного поля силосу

№ п/п.	Найменування інформації	Іденти- фікатор	Напр. вх./вих.	Функція	Вид	Джерело/ Отримувач	Форма представлення (разрядність, точність)		Період вв./вив.
							Зовнішня	Внутрішня	
Лінія опитування термopідвісу (12 виводів)									
1.	Лінія обміну да- ними	TIY1– TIY12	Вхід	Отри- мання даних	Цифровий	Датчики тер- мopідвісу	Інтерфейс і протокол 1- wire	35 двохбай- тових слів	≤ 24 годин
Передавач поля температур силосу									
2.	Радіоканал пере- дачі даних	RT1	Вихід	Передача даних	Цифровий	АРМ ОТЗ	Радіоканал ModBus RTU	35x12=420 двохбайто- вих слів	≤ 24 годин
Приймач поля температур силосу									
3.	Радіоканал прийма- ння даних	RT33	Вхід	Отри- мання даних	Цифровий	Передавачі поля темпера- тур силосу	Радіоканал ModBus RTU	35x12=420 двохбайто- вих слів	≤ 24 годин

Отримані описи вхідних і вихідних сигналів є основою для створення математичного забезпечення Підсистеми і для вибору елементної бази системи керування.

2.2.3 Розрахунок потреб у електронній пам'яті для баз даних Підсистеми

Розглянемо найпростіші структури зберігання даних і адрес в пам'яті контролера Підсистеми. Це можуть бути двовимірні масиви у вигляді матриць 12x35 по кількості, відповідно, підвісок і датчиків у підвісці. Таким чином, якщо за складний індекс запису приймати координати елемента масиву, то отримаємо БД адрес на 420 записів і, відповідно, БД температурного поля на 420 записів. Адреса датчика ds18b20 складається з 8 байт[11]. Таким чином, Для збереження таблиці адрес потрібно виділити у контролері підсистеми

$$420 * 8 = 3\ 360 \text{ байт}$$

енергонезалежної пам'яті даних.

Значення температури зберігається у регістрі розмірністю 2 байти[11]. Тому для зберігання поля температур Підсистеми необхідно виділити в оперативній пам'яті об'єм:

$$420 * 2 = 840 \text{ байт.}$$

Варто також зазначити, що для дистанційного прописування адрес датчиків знадобиться буфер оперативної пам'яті.

Якщо прийняти, що адреси датчиків записуються по ModBus RTU, маємо обмеження даних кадра передачі 512 байт[11]. За таких умов нехай запис адрес датчиків йде по кожній підвісці окремо. Тоді, виходячи з кількості датчиків 35 на підвіс та довжини адреси 8 маємо довжину буферу приймання адрес датчиків:

$$35 * 8 + 2 = 282 \text{ байти,}$$

де 2 байти знадобляться для вказання номера підвіски.

Для передачі температурного поля буферизація не знадобиться, можна всю таблицю передати чотирма фрагментами по

$$840/4=210 \text{ байт.}$$

Таким чином, для обробки порівняно великих об'ємів даних знадобиться

$$840+282=1122 \text{ байт}$$

енергозалежної пам'яті даних.

З урахуванням оперативних потреб на обчислення, округлимо це обмеження до 2 кБ.

Значення 2 кБ енергозалежної і 3 360 Б енергонезалежної пам'яті даних приймаємо як обмежуючу умову під час вибору контролера Підсистеми.

2.2.4 Вибір елементної бази для Підсистеми

Якщо врахувати обмеження на обсяги енергонезалежної та енергозалежної пам'яті висунуті в п. 2.2.3, то контролерами, придатними для реалізації підсистеми, можна вважати Arduino MEGA та Arduino Due [11]. Але за ціною краще обирати Arduino MEGA. Розглянемо технічні характеристики Arduino MEGA [20]:

- мікроконтролер ATmega2560;
- спосіб підключення до USB – ATmega16U2-MU USB контролер;
- робоча напруга 5В;
- вхідна напруга (гранична) 7-11V;
- цифрові входи/виходи – 54 (15 з яких забезпечують вихід PWM/ШИМ);
- аналогові входи – 16;
- постійний струм контактів введення/виведення 40 мА;
- постійний струм на 3,3В Pin – 50 мА;
- флеш-пам'ять 256 кб, з яких 8 кБ використовуються завантажувачем;
- 8 кБ SRAM;
- EEPROM 4 кБ;
- тактова частота 16 МГц.

Зовнішній вигляд Arduino MEGA R3:

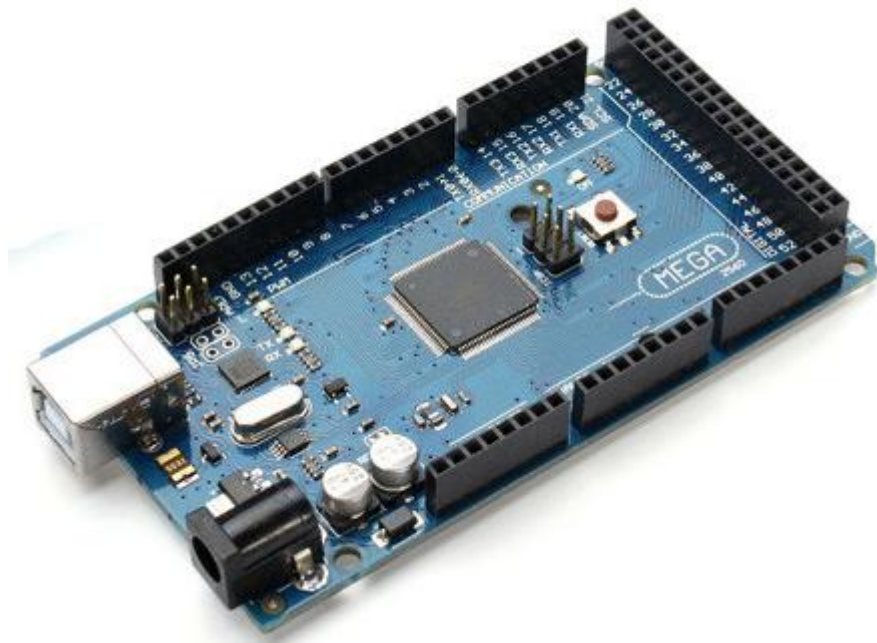


Рисунок 2.2 – Зовнішній вигляд контролера Arduino MEGA R3 [20]

Варто зазначити наявність 54 цифрових входів і виходів у обраній моделі. Цього достатньо для підключення і термopідвісів, і модуля радіоканалу. Для підключення ліній 1-wire слід передбачити 12 підтягуючих резисторів номіналом 4,7 кОм потужністю до 0,5 Вт згідно схеми підключення ds12b20[26].

Оберемо модуль радіоканалу із розрахунку необхідності передачі на відстань до 1000 м з доступних комплектів Arduino. На сьогодні на потрібну відстань передає модуль NRF24L01 з зовнішньою SMA антеною [21]:



Рисунок 2.2 – Зовнішній вигляд радіомодуля NRF24L01 з зовнішньою SMA антеною для Arduino [21]

Технічні характеристики радіомодуля NRF24L01 [21]:

- потужність передачі більше +20 дцБм, 50 Ом, підтримує семиканальний прийом, швидкість передачі до 2 Мбіт/сек;
- робоча напруга 2.7–3.6В, але керуючі контакти толерантні до 5 В (тобто живимо до 3.6, але керуючі без проблем підключаємо до контактів Arduino);
- мультичастотні точки: 125 частотних точок;
- підтримує пересилання втрачених пакетів;
- робоча частота: 2400–2524 МГц;
- модуляція: GMSK;
- чутливість приймача: -95 дцБм;
- максимальний струм при передачі: 115 мА;
- максимальний струм прийому: 45 мА;
- робочий діапазон температур: індустриальний стандарт -45...+85 град;
- посилення PA: 20 дцБ;
- посилення LNA: 10 дцБ.

З боку АРМ ОТЗ для зв'язку з контролерами Підсистем можливо застосувати SDR приймач ADALM-PLUTO [22].

Розрахуємо джерело живлення елементної бази Підстеми. Для цього наведемо дані зі споживання елементів у таблиці 2.2.

Таблиця 2.2 – Розрахунок потреб у живленні Підстеми

№ п/п	Найменування	Споживання на од.	Кількість	Групове споживання
1.	Контролера Arduino MEGA R3	9 В	1	-
2.	Внутрішнє джерело 1 Arduino MEGA R3	5 В	1	1 А
3.	Внутрішнє джерело 2 Arduino MEGA R3	3,3 В	1	150 мА
4.	Радіомодуль NRF24L01	115 мА, 3,3 В	1	115 мА
5.	Інтерфейс радіомодуля	40 мА, 5 В	5	200 мА
6.	Датчик ds18b20	До 4 мА, 3.3-5 В	1*	4 мА
7.	Термопідвіс у стані спокою	До 35 мА, 3.3-5 В	1**	35 мА
8.	Разом споживання, 3,3 В			239 мА

Примітка: * оскільки вимоги реального часу не критичні, передбачається індивідуальне опитування [11] датчиків термопідвісів

Виходячи з розрахунків, наведених у таблиці 2.2, приймемо блок живлення для контролера на 9В[23]:



Рисунок 2.3 – Блок живлення для контролера Arduino [23]

Технічні характеристики блоку живлення [23]:

- вх. напруга 90...264 В;
- частота 47...63 Hz;
- вих. напруга 9 В;
- потужність 18 Вт;
- вих. сила струму 2 А;
- вихідний роз'єм 2.1 x 5.5 мм;
- полярність вих. роз'єму "+" в центрі;
- експлуатаційна температура 0...-40 °С;
- захист КЗ, перевантаження, перенапруга.

Оскільки Підсистема має експлуатуватись під відкритим небом, варто передбачити нагрівач електричний для шафи. Пропонується наступна модель, яка споживає потужність 15 Вт з живленням від мережі 230 В [24]:



Рисунок 2.4 – Нагрівач NTL151-15W для електричної шафи [24]

До нагрівача додається термостат [25]:



Рисунок 2.5 – Термостат NTL10A-F [24]

Обрана елементна база дозволяє перейти до реалізації принципової схеми Підсистеми.

2.2.5 Розробка принципової схеми Підсистеми

Виходячи з таблиці 2.1 і обраної елементної бази пропонується принципове рішення, показане на рисунку 2.5:

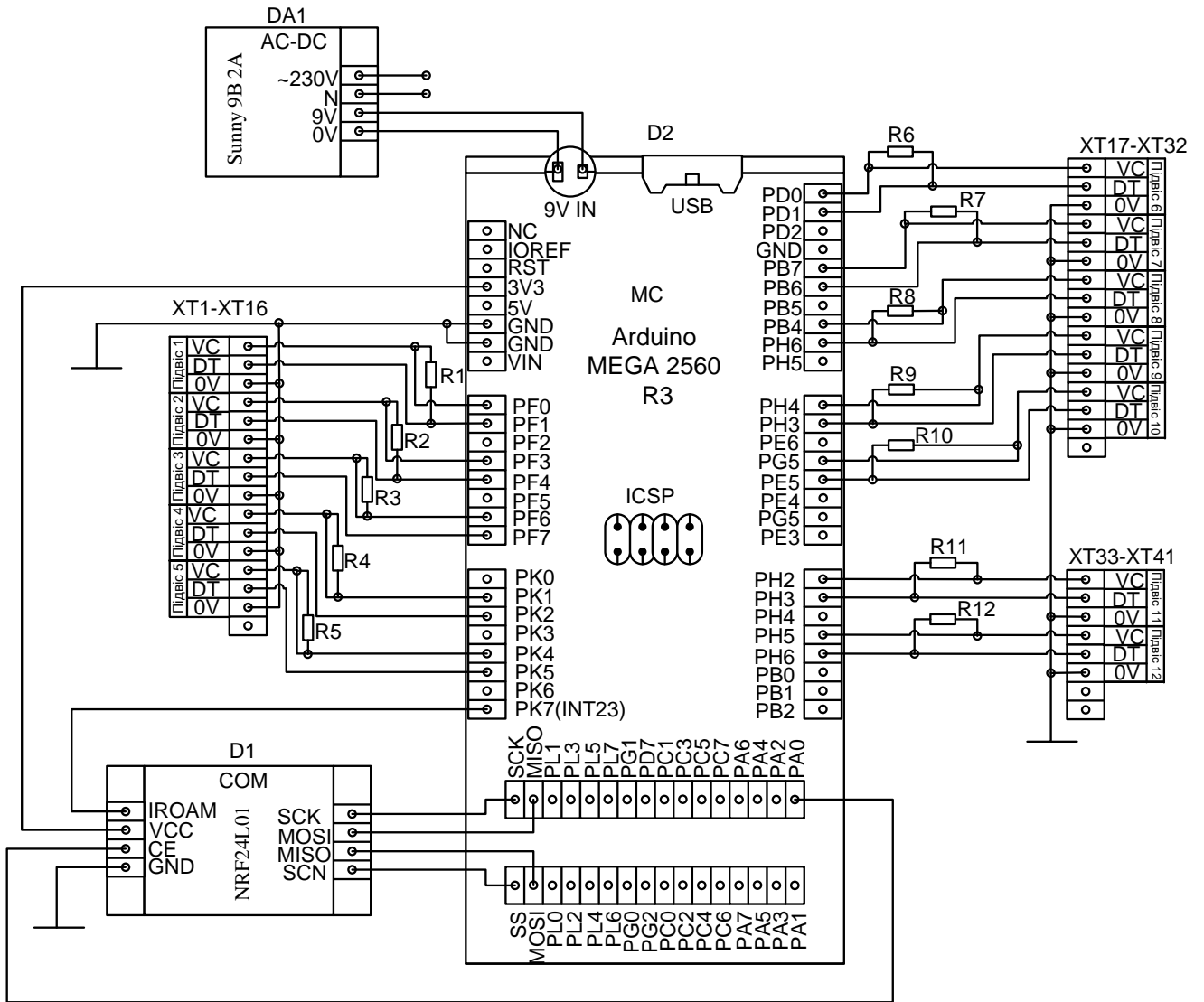


Рисунок 2.5 – Принципова схема Підсистеми

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Призначення і сфера застосування програмного забезпечення

Програмне забезпечення призначене для роботи у складі програмно-технічного забезпечення підсистеми збирання даних температурного поля силосу ПоТ-системи термометрії елеватора.

Програмне забезпечення призначене для опитування цифрових датчиків у складі термопідвісів, які вимірюють температуру у металевих силосах елеватора і для подальшої передачі цих даних у БД, розміщену на АРМ оператора термометрії зерна. Програмне забезпечення і сама підсистеми збирання даних температурного поля силосу ПоТ-системи термометрії елеватора призначені для застосування на елеваторах, побудованих на основі металевих конструкцій у сфері зберігання й переробки зерна.

3.2 Обґрунтування технічних характеристик програми

3.2.1 Функції, виконувані програмою

Програма Підсистеми, реалізована на збірковому контролері силоса елеватора, повинна виконувати наступні функції:

- опитування 12 термопідвісів по 35 датчиків кожен через виділені для кожного термопідвісу сегменти мережі 1-Wire;
- керувати споживанням струму збірковим контролером шляхом відключення датчиків термопідвісів, які не опитуються у даний момент;
- зберігати базу даних адрес датчиків у термопідвісах в енергонезалежній пам'яті;
- переносити базу даних адрес датчиків у термопідвісах з енергонезалежної в енергозалежну пам'ять після подачі живлення на збірковий контролер;
- зберігати базу даних температурного поля силоса, отриманого у результаті опитування цифрових датчиків;

- редагування привілейованим оператором і зберігання бази протоколу вимірювань;
- за командою від АРМ ОТЗ здійснювати опитування адрес датчиків термopідвісів з сортуванням їх у правильній послідовності зі збереженням в енергозалежній пам'яті;
- за командою від АРМ ОТЗ отримувати адреси датчиків термopідвісів від АРМ ОТЗ зі збереженням в енергозалежній пам'яті;
- за командою від АРМ ОТЗ переносити адреси датчиків термopідвісів від енергозалежної до енергонезалежної пам'яті;
- за командою від АРМ ОТЗ пересилати дані температурного поля силоса на АРМ ОТЗ;
- здійснювати обробку команд від АРМ ОТЗ й підтримувати основний алгоритм керування Підсистеми;
- формувати й обробляти прикладний протокол обміну з АРМ ОТЗ;
- формувати повідомлення веденого у протоколі ModBus RTU;
- обробляти запити ведучого у протоколі ModBus RTU;
- здійснювати внутрішній обмін даними між контролером Arduino MEGA та модулем радіозв'язку через інтерфейс SPI.

3.2.2 Вимоги до функцій, виконуваних програмою

Параметри інтерфейсу SPI: 9600 біт/с, 8 біт.

Час затримки реакції на запит від АРМ ОТЗ через радіоканал ModBus RTU не більше 500 мс.

Час отримання даних з температури по одній точці через мережу 1-wire не більше 2 с.

Режим доступу до цифрових датчиків термopідвісів – індивідуальне опитування.

Технологія підтримки мережі 1-wire – бітбенінг [11].

Обмін даними через SPI виконується у фоновому режимі.

Ініціалізація збіркового контролера Підсистеми виконується після подачі живлення на КТЗ.

Затримка оновлення даних на АРМ ОТЗ не повинна перевищувати 3 хвилин за умови роботи всієї системи не менше 1 години.

Періодичність повного оновлення даних на контролері не більше 20 хвилин.

Процес отримання даних температурного поля виконується постійно після ініціалізації збіркового контролера, якщо від АРМ ОТЗ не прийшов запит на виконання інших функцій.

3.2.3 Обґрунтування вхідних і вихідних даних програми

Вхідні дані для ПЗ Підсистеми:

- дані від датчиків температури;
- адреси датчиків від енергонезалежної пам'яті контролера;
- адреси датчиків від АРМ ОТЗ;
- запити від АРМ ОТЗ на тримання адрес датчиків від термopідвісів;
- запити від АРМ ОТЗ на отримання даних температурного поля;
- запити від АРМ ОТЗ на занесення адрес датчиків в від енергонезалежну пам'яті контролера\$
- запити ведучого за протоколом ModBus RTU.

Вихідні дані від ПЗ Підсистеми:

- сигнали підключення живлення термopідвісів;
- сигнали запуску вимірювання датчиків;
- сигнали опитування датчиків температури;
- сигнали опитування адрес датчиків;
- адреси датчиків енергозалежної пам'яті контролера;
- відповіді на запити від АРМ ОТЗ з даними температурного поля;
- відповіді на запити від АРМ ОТЗ з адресами датчиків;
- відповіді веденого за протоколом ModBus RTU.

3.2.4 Обґрунтування алгоритмів і методів, застосованих для реалізації програмного забезпечення

Виходячи зі схеми функціональної структури та п. 3.2.3, загальний алгоритм ПЗ Підсистеми виглядає, як показано на рисунку 3.1:

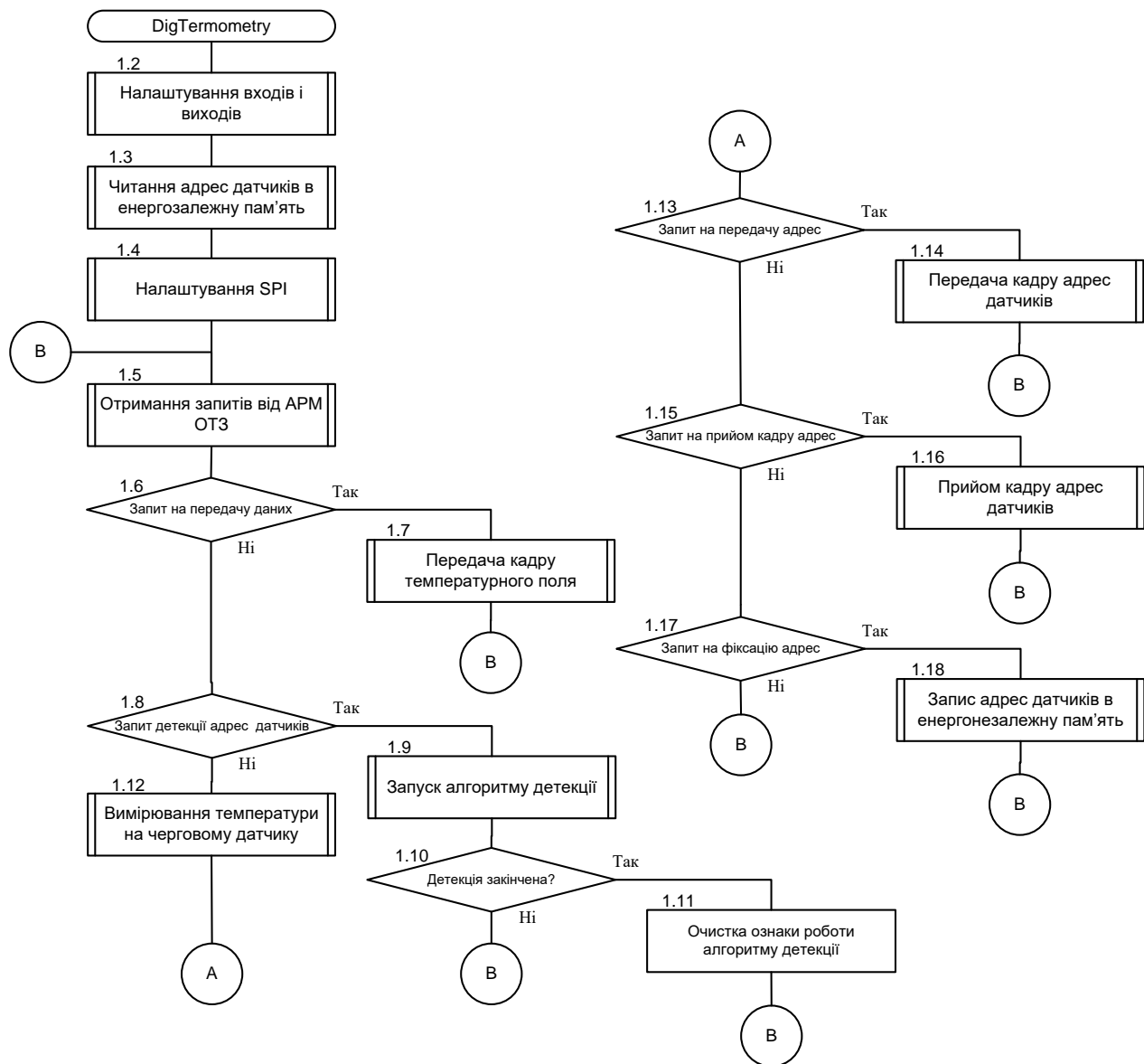


Рисунок 3.1 – Загальна схема алгоритму ПЗ Підсистеми

Загальна схема алгоритму на рисунку 3.1 має бути розкрита більш докладно. Розглянемо процес ініціалізації ПЗ, представлений блоками 1.2–1.4. Його можна представити алгоритмом на рисунку 3.2:

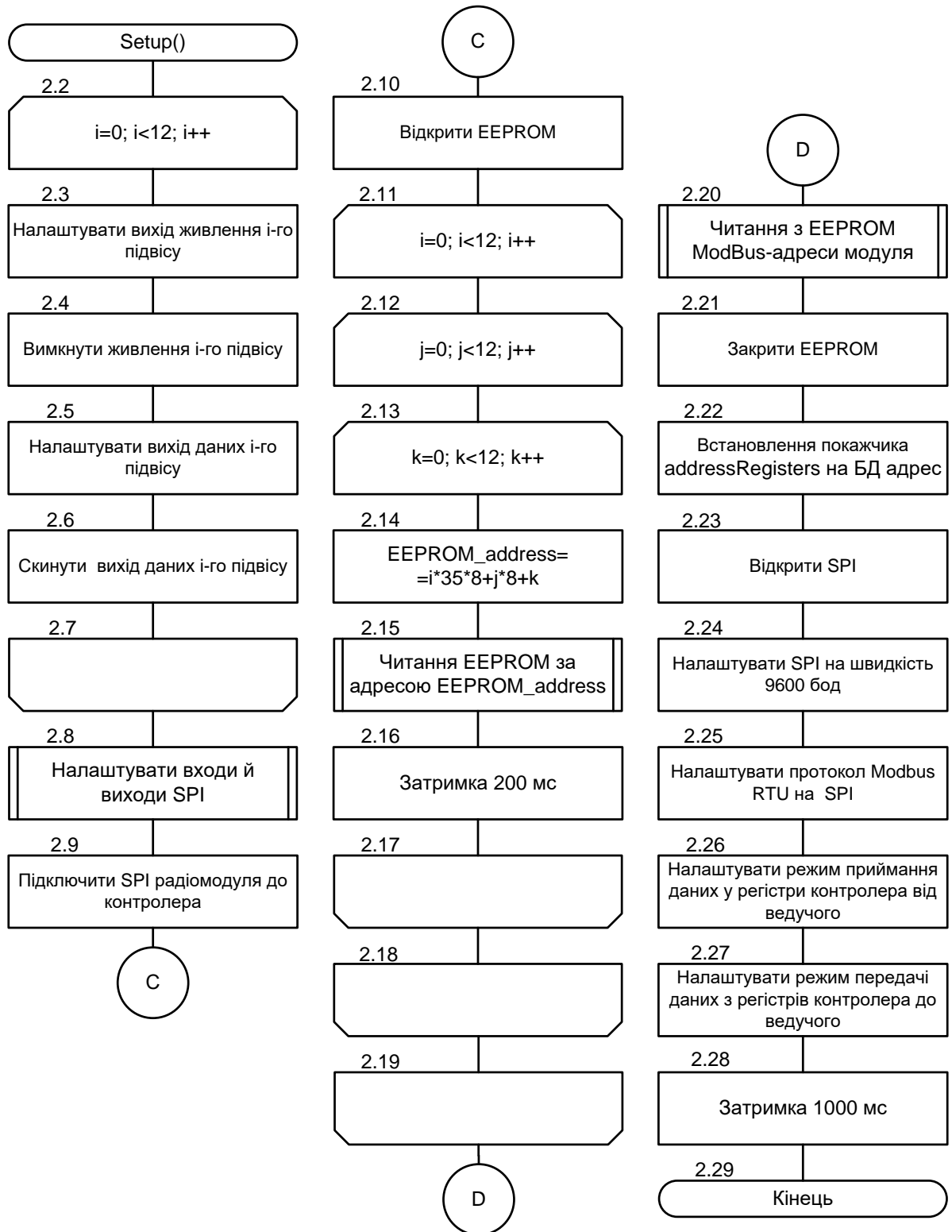


Рисунок 3.2 – Алгоритм ініціалізації ПЗ Підсистеми

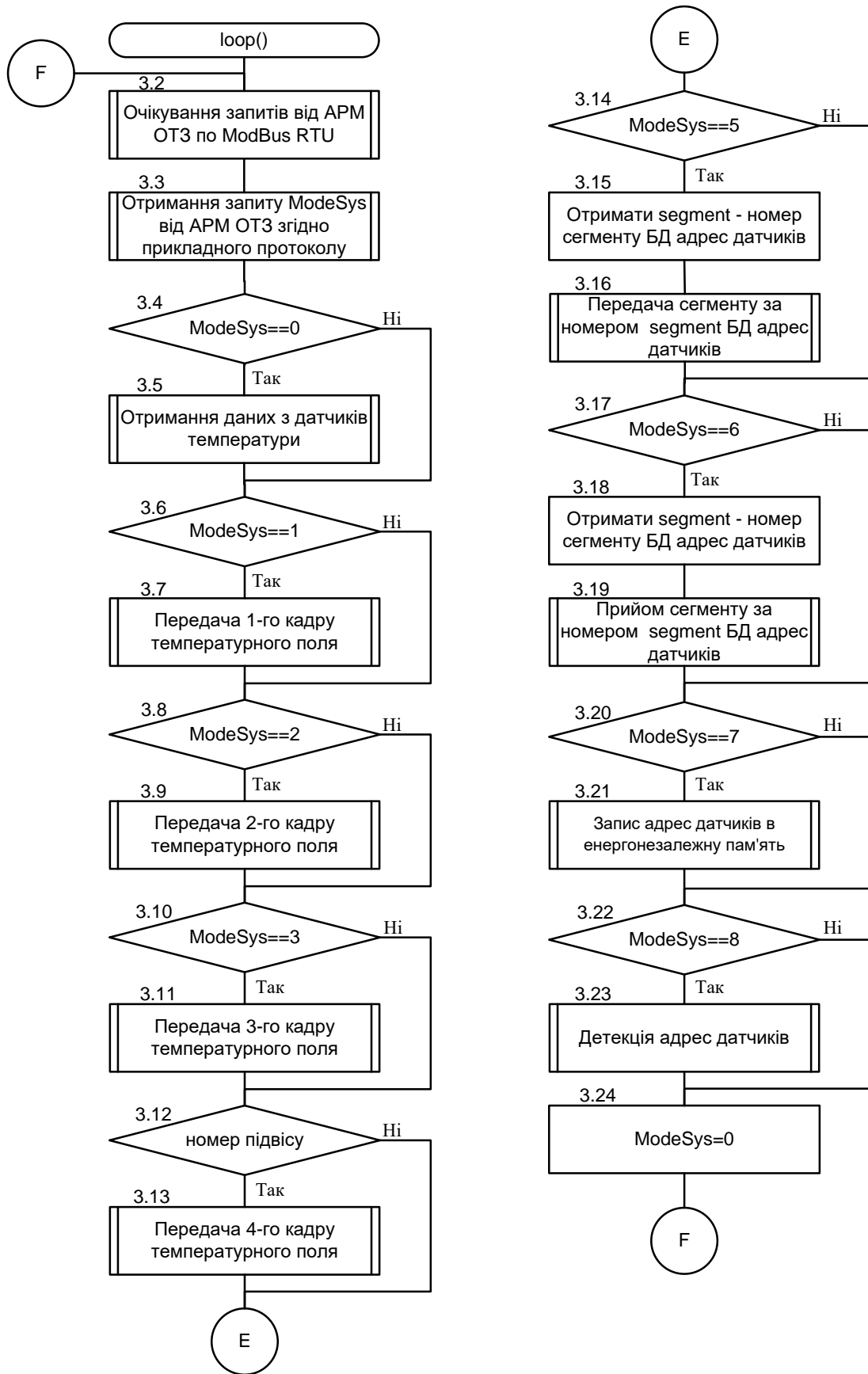


Рисунок 3.3 – Основний алгоритм ПЗ Підсистеми

З урахуванням того, що контролер Підсистеми з боку мережі ModBus RTU є веденим, основний алгоритм контролера, представлений блоками 1.5–1.18, реалізує в собі обробку прикладного протоколу обміну та алгоритм керування Підсистемою у відповідності зі схемою функціональної структури на рисунку 1.9. Основний алгоритм представлено на рисунку 3.3. З нього також можна отримати інформацію про прикладний протокол керування Підсистемою з боку АРМ ОЗТ. Він базується на подачі команд, які будуть представлені значеннями змінної ModeSys:

0 – отримання температур від датчиків у термopідвісах, виконується без отримання жодних команд від АРМ ОЗТ;

1 – передача першого кадру температурного поля силосу, 105 двохбайтових слів;

2 – передача другого кадру температурного поля силосу, 105 двохбайтових слів;

3 – передача третього кадру температурного поля силосу, 105 двохбайтових слів;

4 – передача четвертого кадру температурного поля силосу, 105 двохбайтових слів;

5 – передача сегментів БД адрес датчиків, номер сегменту вказано у додатковій змінній, розмір сегменту становить 70 двохбайтових слів;

6 – приймання сегментів БД адрес датчиків, номер сегменту вказано у додатковій змінній;

7 – запис адрес датчиків з енергозалежної до енергонезалежної пам'яті;

8 – опитування термopідвісів на предмет встановлення адрес датчиків.

Запис адрес датчиків в енергонезалежну пам'ять може здійснюватися за алгоритмом, представленим на рисунку 34.4:

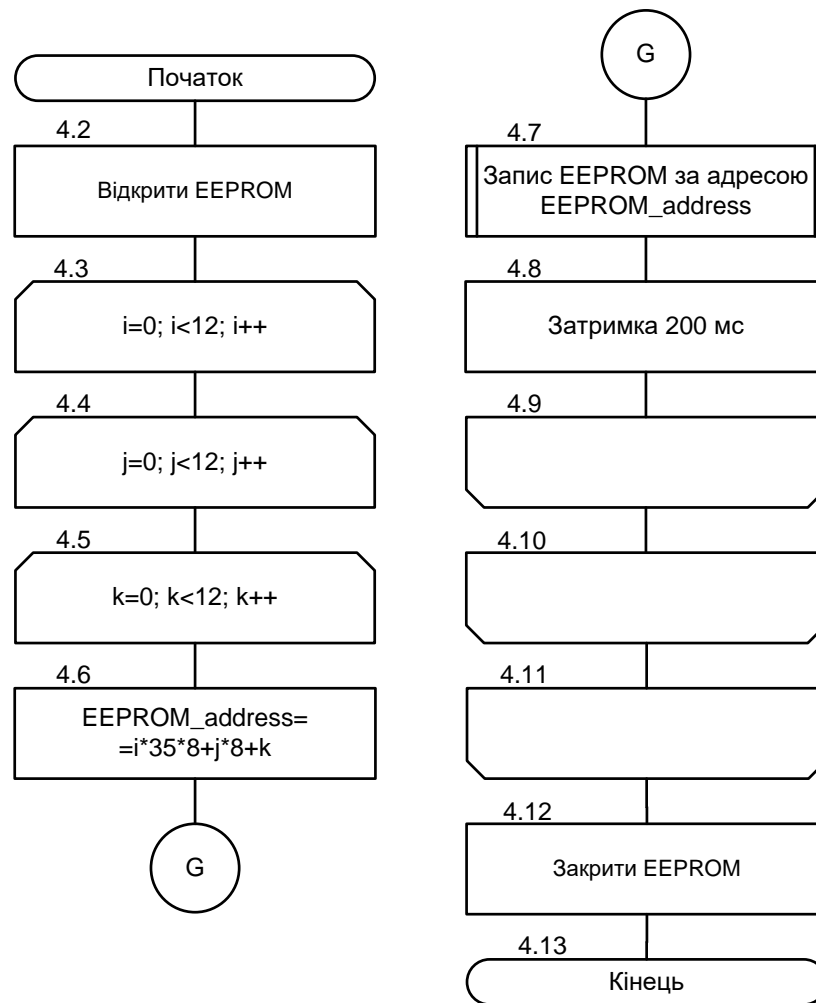


Рисунок 3.4 – Запис адрес датчиків в енергонезалежну пам'ять

Отримання температур від датчиків у термопідвісах є основною функцією ПЗ Підсистеми. Її алгоритм представлено на рисунку 3.5:

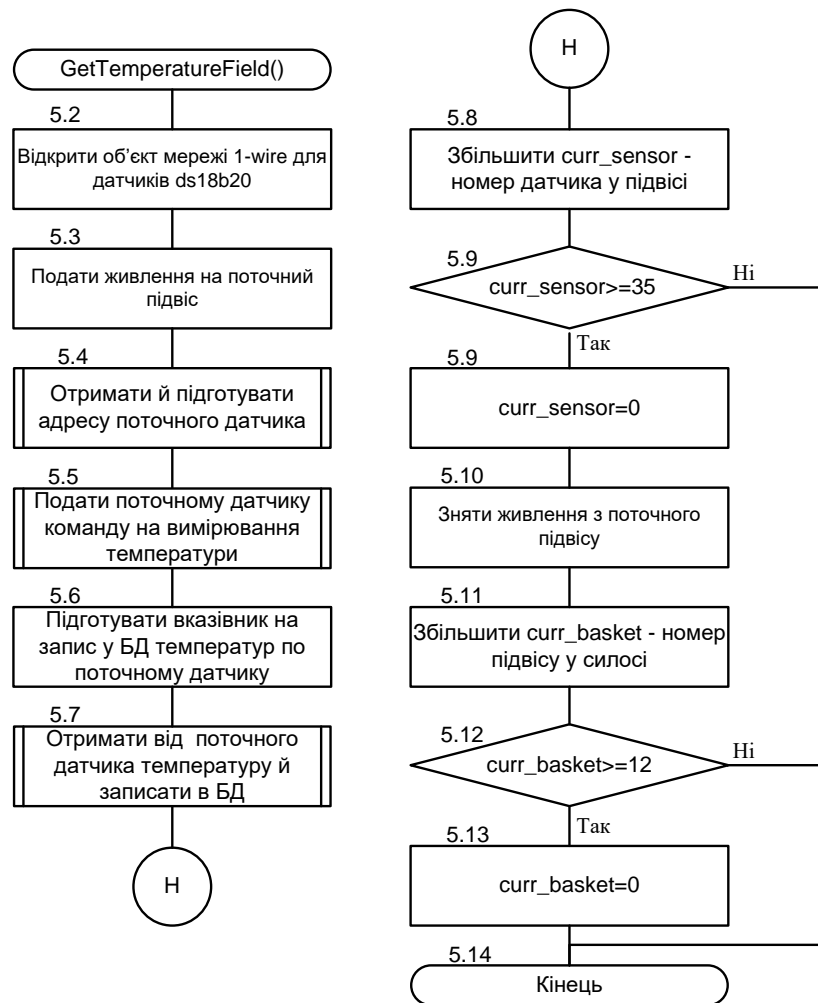


Рисунок 3.5 – Отримання температур від датчиків у термопідвісах

Опитування термопідвісів на предмет встановлення адрес датчиків є опційною функцією Підсистеми. Її алгоритм представлено на рисунку 3.6.

Представлені на рисунках 3.1–3.6 алгоритми після належного визначення програмних і технічних засобів може бути використана для написання ПЗ Підсистеми.

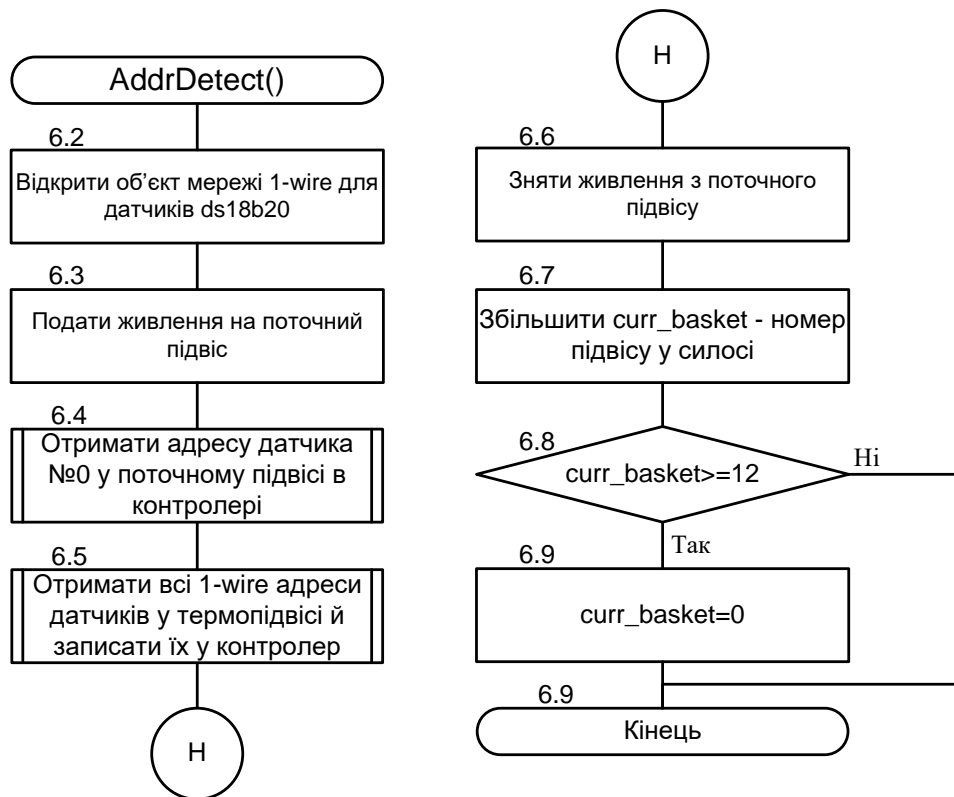


Рисунок 3.5 – Опитування термпідвісів на предмет встановлення 1-wire адрес датчиків

3.2.5 Технічні та програмні засоби, застосовані для реалізації програми

Технічні та програмні засоби, застосовані для розробки програмного забезпечення:

- персональний комп'ютер з вільним інтерфейсом USB;
- операційна система не нижче Windows 10;
- кабель з роз'ємами USB A/USB B;
- середовище Arduino IDE 2.3.4;
- драйвер зв'язку з Arduino Mega 2560;
- бібліотека Arduino.h для роботи з апаратними компонентами Arduino[27];
- бібліотека EEPROM.h для використання об'єкту EEPROM[27];
- бібліотека SPI.h для використання об'єкту керування SPI [28];

– бібліотека ModbusRtu.h для використання об'єкту керування протоколом Modbus RTU[29];

– бібліотеки OneWire.h та DallasTemperature.h для використання об'єкту керування датчиками ds18b20[26].

Мова програмування – C++ для Arduino.

3.3 Опис розробленої програми

3.3.1 Загальна інформація

Текст програми розміщено у проєкті DigTermometry.ino для Arduino IDE у вигляді окремого файлу.

Для завантаження програми у контролер Підсистеми потрібне середовище Arduino IDE 3.2.5. Завантажена програма розміщується в енергонезалежній пам'яті програм контролера Arduino Mega 2560 у вигляді бінарного виконаного коду. Для виконання програми використовуються вбудована в Arduino Mega 2560 операційна система. Запуск програми на виконання відбувається під час ввімкнення живлення.

Мова реалізації програми – C++ для Arduino.

3.3.2 Функціональне призначення

Програма здійснює керування вимірюванням і отримує дані з температур від 420 датчиків ds12b20, розподілених по 35 у 12 термopідвісів.

Програма керує 12 сегментима мережі 1-wire в якості ПЗ ведучого пристрою.

Програма керує споживанням електроенергії датчиками температури у підвісах.

Програма зберігає дані з температур у внутрішній БД у вигляді температурного поля силосу та надає їх АРМ ОТЗ.

Програма організує енергонезалежне зберігання адреси контролера й адрес датчиків ds12b20.

Програма здійснює виявлення адрес датчиків ds12b20 межах термopідвісів і надає їх АРМ ОТЗ для корекції їх просторового розміщення.

Програма обробляє запити від АРМ ОТЗ на надання даних з температурного поля силосу, надання адрес датчиків ds12b20, на отримання всіх адрес цих датчиків, і на занесення їх в енергонезалежну пам'ять.

Програма готує й оброблює повідомлення для АРМ ОТЗ, надаючи доступ до контролера Підсистеми як до якості веденого за протоколом ModBus RTU.

Програма керує радіомодулем Підсистеми.

Програма не встановлює послідовність розміщення датчиків ds12b20 в окремому термopідвісі.

Програма не веде протокол процесу.

Програма не керує обігрівом шафи КТЗ Підсистеми.

3.3.3 Опис логічної структури програми

Логічна структура програми відповідає схемі функціональної структури, розробленої у розділі 1 пояснювальної записки.

Програма розміщена в енергонезалежній пам'яті програм контролера Arduino Mega 2560 у вигляді виконуваного коду, також окремо у файлі DigTermometry.ino у вигляді початкового коду.

Файл DigTermometry.ino містить:

- область директив компілятора;
- область глобальних змінних;
- функцію ініціалізації setup();
- функцію readHoldingRegisters();
- функцію writeMultipleRegisters();
- функцію AddrDetect();
- функцію GetTemperatureField;
- функцію loop().

Область директив компілятора містить об'яви підключення використуваних бібліотек: `Arduino.h`, `EEPROM.h`, `SPI.h`, `ModbusRtu.h`, `OneWire.h`, `DallasTemperature.h`.

Область глобальних змінних містить: об'яви й ініціалізацію змінних, що використовуються прикладним протоколом обміну Підсистеми з АРМ ОТЗ; описання розпіновки виходів контролера для підключення живлення й ліній даних для обміну з термopідвісами; розпіновку виходів для підключення ліній обміну даними й керування радіомодулем; масиви даних для зберігання адрес датчиків, даних температури, ModBus-адреси Підсистеми, інших службових даних для обміну з АРМ ОТЗ; об'яви й ініціалізацію об'єктів, які виконують обслуговування датчиків `ds18b20`, керування зв'язком по мережі 1-wire, обслуговування запитів по протоколу Modbus RTU.

Функція `setup()` містить виклики макросів налаштування входів і виходів, а також встановлення початкових значень виходів, які використовуються в обміні даними по мережі SPI, для керування живленнями термopідвісами, для обміну даними й керування радіомодулем. Виконується читання з енергонезалежної до енергозалежної пам'яті даних адрес датчиків `ds18b20` та ModBus-адреси Підсистеми. Налаштовується покажчик для взаємодії з базою даних адрес датчиків в енергозалежній пам'яті. Відкриваються канал зв'язку по мережі SPI з радіомодулем та ініціюються поля-канали обміну для читання й запису за допомогою об'єкту керування протоколом ModBus RTU.

Функція `readHoldingRegisters()` готує посилку-відповідь Підсистеми на запит від АРМ ОТЗ згідно протоколу ModBus RTU, заповнюючи даними поле даних.

Функція `writeMultipleRegisters()` обробляє запит від АРМ ОТЗ на запис групи регістрів контролера Підсистеми, копіюючи вміст поля даних запиту у призначений для цього буфер даних.

Функція `AddrDetect()`, відкриває канал зв'язку 1-wire і здійснює по чергове сканування всіх термopідвісів на предмет встановлення адрес 1-wire всіх наявних датчиків на кожному з 12 термopідвісів.

Функція `GetTemperatureField` відкриває канал зв'язку 1-wire і підключає живлення поточного опитуваного термopідвісу. Функція отримує з енергозалежної пам'яті адресу 1-wire поточного у термopідвісі датчика, надсилає йому індивідуальну команду на вимірювання температури, отримує дані з температурою. Потім номер поточного датчика збільшується і очікується наступний виклик функції `GetTemperatureField`. У випадку, коли поточний датчик вже був останнім у термopідвісі і збільшувати його немає сенсу, розпочинаються дії по зміні номера поточного термopідвісу. Для цього спочатку знімається живлення з поточного термopідвісу. Потім номер поточного датчика змінюється на нульовий (фактично, це номер першого датчика у наступному термopідвісі). Далі збільшується номер поточного підвісу й відбувається вихід з функції. Наступний виклик `GetTemperatureField` буде вже з датчиком номер 0 з іншим термopідвісом. У випадку, коли поточний термopідвіс вже був на номером 11, замість збільшення буде присвоєння 0 змінній, що зберігає номер поточного термopідвісу. Таким чином забезпечується циклічність отримання значень температурного поля силосу.

Функція `loop()` є основним програмним циклом контролера Підсистеми. Перший раз вона викликається після того, як відпрацювала функція `setup()`. Далі операційна система контролера викликає функцію `loop()` циклічно й безперервно після закінчення відпрацювання попереднього виклику функції `loop()`. Функція викликає метод об'єкту-веденого ModBus RTU, який здійснює очікування і обробку ModBus-кадру із запитом ведучого. Після отримання запиту ведучого відбувається виклик функції `writeMultipleRegisters()` з подальшим розпізнаванням слова протоколу прикладного рівня. В залежності від результатів розпізнавання, у функції `loop()` виконується наступне:

- виклик функції `GetTemperatureField()` для отримання температури від поточного датчика температури. Ця дія виконується за замовчуванням, навіть без отримання слова протоколу прикладного рівня від АРМ ОТЗ;

- виклик функції `readHoldingRegisters()` для передачі першого, другого, третього чи четвертого кадру з даними по температурному полю силоса;

- виклик функції `readHoldingRegisters()` для передачі на АРМ ОТЗ одного з шести (від 0 до 5) кадрів, що містять адреси 1-wire датчиків температури;
- виклик функції `writeMultipleRegisters()` для отримання від АРМ ОТЗ одного з шести (від 0 до 5) кадрів, що містять адреси 1-wire датчиків температури;
- запис до енергонезалежної пам'яті даних з енергонезалежної адрес датчиків ds18b20 та ModBus-адреси Підсистеми;
- виклик функції `AddrDetect()` для отримання адрес 1-wire датчиків температури поточного термопідвісу.

3.4 Використані технічні засоби

Для виконання програми необхідні: контролер Arduino Mega R3 2560 – 1 шт.; радіомодуль NRF24L01 – 1 шт.; SMA-антена – 1 шт.; блок живлення 9В 2А – 1 шт.

3.5 Виклик та завантаження

Відповідні модулі програми завантажуються до енергонезалежної пам'яті контролера Arduino Mega 2560, по інтерфейсу USB A/USB B, де знаходяться весь час експлуатації Підсистеми. Початковий проект знаходиться на окремому енергонезалежному носії у вигляді файлу `DigTermometry.ino`.

Виклик програми на виконання відбувається після ввімкнення живлення обладнання Підсистеми.

Вхідна точка в програми – функція `setup()`.

3.6 Вхідні й вихідні дані

Вхідні та вихідні дані наведено у таблиці 3.1:

Таблиця 3.1 – Вхідні й вихідні дані Підсистеми

№ п/п	Змінна	Тип	Напрямок	Призначення
1.	ModeSys	int	Вхід	Режим роботи обладнання (слово протоколу прикладного рівня)
2.	dsVCC[PF0, PF3, PF6, PK1, PK4, PD0, PB7, PB3, PG5]	const int	Вхід	Розпіновка ліній живлення термопідвісів
3.	dsData[PF1, PF4, PF7, PK2, PK5, PD1, PB6, PE6, PE5]	const int	Вхід/Вихід	Розпіновка ліній обміну даними термопідвісів
4.	oneWire(dsData[0])	OneWire	Вхід/Вихід	Екземпляр класу обміну даними з 1-wire
5.	sensors(&oneWire)	DallasTemperature	Вхід/Вихід	Екземпляр класу обслуговування ds18b20
6.	deviceAddresses[35]	DeviceAddress	Вхід	Оперативний масив для збереження адрес датчиків
7.	curr_sensor	int	Вихід	поточний датчик

Продовження таблиці 3.1

№ п/п	Змінна	Тип	Напрямок	Призначення
8.	curr_basket	int	Вихід	поточний підвіс
9.	CE=22	const int	Вихід	Керування радіомодулем. Режим роботи, RX/TX
10.	SS_PIN	const int	Вихід	Керування радіомодулем. Вибір модуля
11.	SCK	const int	Вихід	Керування радіомодулем. Синхроімпульси
12.	MOSI	const int	Вихід	Керування радіомодулем. Вихід даних
13.	MISO	const int	Вхід	Керування радіомодулем. Вхід даних
14.	IROAM	const int	Вхід	Керування радіомодулем. Запит на переривання
15.	AdrDB[12, 35, 8]	Масив uint8_t	Вхід/Вихід	Адреси датчиків в енергозалежній пам'яті

Завершення таблиці 3.1

№ п/п	Змінна	Тип	Напрямок	Призначення
16.	NUM_REGISTERS	const int	Вихід	Кількість регістрів даних з температурами для одного кадру ModBus
17.	dataRegisters[420]	Масив uint16_t	Вхід/Вихід	Регістри даних з температурами
18.	addressRegisters	uint16_t *	Вхід/Вихід	Вказівник на сегмент області регістрів адрес датчиків для передачі по ModBus
19.	commandRegister[2]	Масив uint16_t	Вхід	Регістр прикладного протоколу обміну з АРМ ОТЗ
20.	slaveAddress	uint8_t	Вихід	Регістр з адресою веденого Modbus
21.	slave(slaveAddress, 0, 0)	Modbus	Вхід /Вихід	Екземпляр класу веденого Modbus

ВИСНОВКИ

У роботі розроблюються детальні технічне й програмне рішення для Підсистеми збирання даних температурного поля силосу у складі IoT-системи термометрії елеватора на основі схеми функціональної структури та сформульованих технічних вимог. У результаті отримано технічне рішення, представлене функціональною схемою автоматизації та принциповою схемою Підсистеми. Технічне рішення побудоване на елементній базі для контролерів ардуіно, з урахуванням необхідності роботи в умовах під відкритим небом. В якості програмного рішення для Підсистеми розроблено алгоритм і програму, яка здійснює опитування термопідвісів силосу елеватора, побудованих на основі цифрових датчиків від Dallas Semiconductor, а також обмін даними з автоматизованим робочим місцем оператора термометрії зерна по протоколу ModBus RTU через радіоканал.

Таким чином, можна стверджувати, що запропоновані програмні й технічні рішення у повній мірі відповідає технічним вимогам, поставленим до Підсистеми збирання даних у складі IoT-системи термометрії елеватора.

Ступінь вирішення поставлених задач достатня для досягнення мети кваліфікаційної роботи – розробити технічно-програмне рішення блоку збирання даних температурного поля силосу на основі обладнання Arduino, придатне для використання у складі IoT-системи термометрії зернових силосів елеватора.

Практичне значення результатів роботи для зернового елеватора, побудованого на основі металевих силосів, полягає у забезпеченні процесу термометрії зерна на основі готових стандартних технічних засобів з використанням стандартного протоколу обміну ModBus RTU та безпроводного каналу передачі даних в межах 1000 метрів.

Подальший розвиток системи передбачено напрямку розробки технічного й програмного рішення для автоматизованого робочого місця оператора.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Міністерство аграрної політики України. Наказ № 228 від 15.06.2004. Про затвердження Технічного регламенту зернового складу <https://zakon.rada.gov.ua/laws/show/z0835-04#Text>.
2. Зберігання зернових культур. Компанія Trotec. <https://trotec.com.ua/uk/blog/zberigannya-zernovyh-kultur/>.
3. Як запобігти самозігріванню та злежуванню зерна. Головне управління Держпродспоживслужби в Черкаській області. <https://www.cherk-consumer.gov.ua/novyny/3483-yak-zapobihyty-samozihrivanniu-ta-zlezhuvanniu-zerna>.
4. Зберігання зерна у полімерних рукавах: особливості технології. Головне управління Держпродспоживслужби в Черкаській області. <https://www.cherk-consumer.gov.ua/hromadianam/upravlinnia-fitosanitarnoi-bezpeky/novyny-upravlinnia-fitosanitarnoi-bezpeky/3669-zberihannia-zerna-u-polimernykh-rukavakh-osoblyvosti-tekhnologii>.
5. Силос для зерна RIELA. Компанія RIELA. https://riela.com.ua/sylosy-dlya-zerna-s-ploskym-dnyshhem/?gad_source=1&gclid=CjwKCAiAiOa9BhBqEiwABCdG8714FJupn3KMzklHpnj2BdVHbdtZxRj8qZG0y08ZTXyKP93DPV5kMB0CTzsQAvD_BwE.
6. Елеватори. Компанія Арсенал. <https://mysilo.com.ua/pro-kompaniyu/elevatory.html>.
7. Олена Гайдук. Які бувають елеватори за конструкцією та підходами до діяльності – класифікація елеваторів. <https://elevatorist.com/spetsproekt/224-slovník-terminiv-yaki-buvayut-elevatori-za-konstruktsiyeyu-ta-pidhodami-dodiyalnosti--klasifikatsiya-elevatoriv>.
8. ДСТУ ISO 4112:2014 Зернові та бобові. Вимірювання температури зерна. Затверджено наказом Мінекономрозвитку України № 1429 від 2 грудня 2014 р. Чинний від 2015–05–01.

9. Контроль температури зерна ручними термощупами та автоматично. Компанія Технотест. <https://technotest.com.ua/kontrol-temperature-zerna-ruchnymi-termoschupami-uk.html>
10. O. Sokolovskaya, L. Valevskaya. Study of grain thermometry during storage. Зернові продукти і комбікорми, Vol.23, I.1(89)/2023 <http://journals.ontu.edu.ua/index.php/gpmf/article/view/2681/2844>.
11. С.М. Ткаченко. Обґрунтування технічного рішення для систем цифрової термометрії зерна на основі контролера Arduino. XVII Міжнародна конференція «Проблеми використання інформаційних технологій в освіті, науці та промисловості». 24 листопада 2022 року м. Дніпро. Збірник наукових праць № 7. Дніпро: НТУ «ДП» 2023. с.52-61.
12. PC Based Temperature Cable Installation Instructions. Safe-Grain Inc. Mason, Ohio 04.07.07 12p.
13. С.М. Ткаченко Модернізація систем термометрії зерна в силосах елеватора з підвісками на основі термопар. Збірник наукових праць Національного гірничого університету. - 2017. - № 50. - С. 276-288. - Режим доступу: http://nbuv.gov.ua/UJRN/znpngu_2017_50_34.
14. DS18B20. Programmable Resolution 1-Wire® Digital Thermometer. Datasheet. Dallas Semiconductor Inc. <https://datasheet4u.com/datasheet-pdf/DallasSemiconductor/DS18B20/pdf.php?id=1011203>.
15. Обладнання для елеваторів зберігання зерна та зернопродуктів на базі сучасних систем термометрії з використанням цифрових датчиків температури фірми Dallas Semiconductor у складі термопідвісок. Асоціація науково-виробничих підприємств «Ельдорадо». <http://www.dnvpeldorado.com/index.php/ua/seeds-elevators>.
16. Термометрія елеватора із підключенням до ПК – Агролог TMS5000. Компанія Технотест Груп. <https://technotest.com.ua/dlya-zernovoy-promyshlennosti-uk/sistemy-termometrii-agrolog-uk/sistema-kontrolya-temperature-s-podklyucheniem-k-pk-agrolog-tms5000.html>.

17. Система термоконтролю. Компанія ІТГ.
<https://itg.bg/uk/products/sistema/>.
18. Системи термометрії зернових елеваторів. Компанія Інновинпром.
<https://innovinnprom.com/galuzevi-rishennya/systemy-termometriyi-zernovyh-elevatoriv>.
19. Системи контролю температури на основі термопідвісок. Компанія Temix.
<https://www.temix.com.ua/product-category/temperature-control-systems/temperature-control-based-on-temperature-probe/>.
20. Arduino Mega 2560 Rev3. Компанія ARDUINO.UA.
<https://arduino.ua/prod243-arduino-mega-2560-rev3?srsId=AfmBOorzWDWDtJcegTK8xzqhEQvZNSXIVdBdPtFCY3ADgW-yGRLHHVi2>.
21. Бездротовий модуль NRF24L01+PA+LNA з зовнішньою SMA антеною. Компанія ARDUINO.UA.
<https://arduino.ua/prod640-besprovodnoi-modyl-nrf24l01palna-s-vneshnei-sma-antennai>.
22. ADALM-PLUTO (Active Learning Module - PlutoSDR) SDR приймач та передавач з відкритим вихідним кодом. Компанія ARDUINO.UA.
<https://arduino.ua/prod6887-adalm-pluto-sdr>.
23. Блок живлення Sunny 9В 2А 2.1x5.5мм. Компанія ARDUINO.UA.
https://arduino.ua/prod788-blok-pitaniya-9v-2a?srsId=AfmBOoqujSvMLtkZruyK_q9C0H0rFyVoX8-ZhY6cpsVhRTJWAmLKIfW1.
24. Нагрівач NTL151-15W для електричної шафи. Мале підприємство "Ремікс".
https://remix.in.ua/ua/p961080084-nagrevatel-ntl151-15w.html?source=merchant_center&gad_source=1&gclid=CjwKCAiA5pq-BhBuEiwAvkzVZT93jXmln81coa7Xnn3hrWefqkPO7cHdsZKP6usSmf4ex59dLPe2wRoCKOcQAvD_BwE.
25. Термостат NTL10A-F. Мале підприємство "Ремікс".
https://remix.in.ua/p959265838-termostat-ntl10a.html?source=merchant_center&gad_source=1&gclid=CjwKCAiA5pq-

[BhBuEiwAvkzVZU5Bu4OUE8T0t2PzN3qP_5GFDpuB6dt6j_0ML7Rm0NxervCeDMquaBoCENcQAvD_BwE.](#)

26. Підключення датчика температури DS18B20 до Arduino та ESP.
Сайт Qazf. <https://qazf.com.ua/ds18b20-arduino/>.

27. EEPROM Arduino. Сайт Qazf. <https://qazf.com.ua/eeprom-arduino/>.

28. SPI. Сайт ARDUINO.
<https://reference.arduino.cc/reference/en/language/functions/communication/spi/>.

29. Arduino Modbus Slave. Kri Bielinski.
<https://github.com/yaacov/ArduinoModbusSlave/blob/master/README.md>.

ДОДАТОК А

Текст програми підсистеми збирання даних температурного поля силосу

**Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
“ДНІПРОВСЬКА ПОЛІТЕХНІКА”**

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
ПІДСИСТЕМА ЗБИРАННЯ ДАНИХ ТЕМПЕРАТУРНОГО ПОЛЯ
СИЛОСУ**

Текст програми

804.02070743.25018-01 12 01

Листів 10

АНОТАЦІЯ

Дана програма містить в собі початковий програмний код підсистеми збирання даних температурного поля силосу, призначеної для роботи у складі IoT-системи термометрії елеватора у галузі переробки й зберігання зерна.

Програма призначена для отримання даних з температурою контрольних точок зернової маси, яка зберігається у металевому силосі елеватора з 12 термопідвісами та надсилання їх по радіоканалу згідно протоколу ModBus на автоматизоване робоче місце оператора термометрії зерна.

Програма написана мовою C++ для Arduino, відлагоджена із застосуванням середовища Arduino IDE і призначена для застосування на контролерах Arduino Mega R3.

ЗМІСТ

	С.
1. Файл DigThermometry.ino	4

```

#include <Arduino.h>
#include <EEPROM.h>
#include <SPI.h>
#include <ModbusRtu.h>
#include <OneWire.h>
#include <DallasTemperature.h>

//Режим роботи обладнання (слово протоколу прикладного рівня)
int ModeSys = -1;

//Керування 1-Wire
const int dsVCC[PF0, PF3, PF6, PK1, PK4, PD0, PB7, PB3, PG5]; //
Розпіновка ліній живлення термopідвісів
const int dsData[PF1, PF4, PF7, PK2, PK5, PD1, PB6, PE6, PE5]; //
Розпіновка ліній обміну даними термopідвісів
OneWire oneWire(dsData[0]); // Конструктор класу обміну даними з 1-
wire
DallasTemperature sensors(&oneWire); // Конструктор класу
обслуговування ds18b20
DeviceAddress deviceAddresses[35]; // оперативний масив для
збереження адрес датчиків
int curr_sensor=0;// поточний датчик
int curr_basket=0;// поточний підвіс

//Керування радіомодулем
const int CE=22;//режим роботи, RX/TX
const int SS_PIN=53;//вибір модуля
const int SCK=52;//синхроімпульси
const int MOSI=50;//вихід даних
const int MISO=51;//вхід даних

```

```

const int IROAM=82;//запит на переривання

uint8_t AdrDB[12, 35, 8];//адреси датчиків в енергозалежній пам'яті

// Налаштування регістрів даних з температурами для ModBus
const int NUM_REGISTERS = 125;
uint16_t dataRegisters[420];
uint16_t * addressRegisters;
uint16_t commandRegister[2];

// Налаштування Modbus та SPI
uint8_t slaveAddress = 61; // Початкова адреса веденого
Modbus slave(slaveAddress, 0, 0); // Адреса веденого = 61, та
використання Serial port

void setup() {

    //Налаштування 1-wire
    for(int i=0; i<12; i++)
    {
        pinMode(dsVCC[i],OUTPUT);
        digitalWrite(dsVCC[i], LOW);
        pinMode(dsData[i],OUTPUT);
        digitalWrite(dsData[i], LOW);
    }

    pinMode(CE, OUTPUT);
    pinMode(SS_PIN,OUTPUT);
    pinMode(SCK, INPUT);
    pinMode(MOSI,OUTPUT);

```

```

pinMode(MISO, INPUT);
pinMode( IROAM, INPUT);
digitalWrite(CE, HIGH);

// Читання адрес датчиків з енергонезалежної пам'яті
EEPROM.begin();
for(i=0; i<12; i++)
  for(int j=0; j<35; j++)
    for(int k=0; k<8; k++)
      {
        int EEPROM_address=i*35*8+j*8+k;
        AdrDB[i,j,k] = EEPROM.read(EEPROM_address);
        delay(200);
      }
  slaveAddress = EEPROM.read(3800); //отримання дійсної адреси
веденого
EEPROM.end();

addressRegisters = &AdrDB[0, 0, 0]; //встановлення покажчика на базу
даних адрес

//Ініціалізація порта керування радіомодулем
SPI.begin();
Serial.begin(9600);

// Ініціалізація Modbus з поточною адресою веденого
slave.begin(9600);
  slave.cbVector[CB_WRITE_MULTIPLE_REGISTERS] =
writeMultipleRegisters;

```

```

        slave.cbVector[CB_READ_HOLDING_REGISTERS]
readHoldingRegisters;
        delay(1000);

    }

uint8_t readHoldingRegisters(uint8_t fc, uint16_t address, uint16_t
numRegs) {
    // Читання регістрів та повернення значень
    for (int i = 0; i < numRegs; i++) {
        if ((address + i) < NUM_REGISTERS) {
            slave.R[i] = dataRegisters[address + i];
        } else {
            slave.R[i] = 0; // Повернення 0, якщо адреса виходить за межі
        }
    }
    return Modbus::EX_SUCCESS;
}

uint8_t writeMultipleRegisters(uint8_t fc, uint16_t address, uint16_t
numRegs) {
    // Запис групи регістрів
    for (int i = 0; i < numRegs; i++) {
        if ((address + i) < NUM_REGISTERS) {
            dataRegisters[address + i] = slave.R[i];
        } else {
            return Modbus::EX_ILLEGAL_DATA_ADDRESS; // Повернення
помилки, якщо адреса виходить за межі
        }
    }
}

```

```

return Modbus::EX_SUCCESS;
}

void AddrDetect()
{
    // Ініціалізація датчиків температури
    sensors.begin();

    //Подати живлення на поточний підвіс
    digitalWrite(dsVCC[curr_basket], HIGH);

    //Отримати адресу датчика №0 у поточному підвісі в контролері
    uint8_t * pAddr = &deviceAddresses[0];
    for(int i=0; i<8; i++)
    {
        pAddr[i] = AddrDB[curr_basket, 0, i];
    }

    //Отримати адресу по 1-wire
    for (i = 0; i < sensors.getDeviceCount(); i++)
        sensors.getAddress(deviceAddresses[i], i)

        curr_sensor=0;//Скинути лічильник датчиків у підвісі
        digitalWrite(dsVCC[curr_basket], LOW);//Зняти живлення з підвісу
        curr_basket++;//Збільшити лічильник підвісів
        if(curr_basket>=12)//Якщо лічильник підвісів більше 12 -
            curr_basket=0;//Скинути лічильник підвісів
    }
}

```

```

void GetTemperatureField()
{
    // Ініціалізація датчиків температури
    sensors.begin();

    //Подати живлення на поточний підвіс
    digitalWrite(dsVCC[curr_basket], HIGH);

    //Отримати адресу поточного датчика
    uint8_t * pAddr = &deviceAddresses[0];
    for(int i=0; i<8; i++)
    {
        pAddr[i] = AddrDB[curr_basket, curr_sensor, i];
    }

    // Вимірювання температури одним датчиком
    sensors.requestTemperaturesByAddress(deviceAddresses);
    int pDataRegister = 12*curr_basket+35*curr_sensor;
    dataRegisters[pDataRegister] = int (sensors.getTempC(deviceAddresses));

    curr_sensor++;//Збільшити номер датчика у підвісі
    if(curr_sensor>= 35)//Якщо номер датчика у підвісі більше 35 -
    {
        curr_sensor=0;//Скинути номер датчика у підвісі
        digitalWrite(dsVCC[curr_basket], LOW);//Зняти живлення з підвісу
        curr_basket++;//Збільшити номер підвісу
        if(curr_basket>=12)//Якщо лномер підвісу більше 12 -
            curr_basket=0;//Скинути номер підвісу
    }
}

```

```
}  
  
void loop() {  
  
    slave.poll(); // Очікування запитів від АРМ ОТЗ  
  
    writeMultipleRegisters(0x10, commandRegister[0], 2);  
    ModeSys = commandRegister[0];  
  
    //Вимірювання температур через 1-wire  
    if(ModeSys == 0)  
        GetTemperatureField();  
  
    //Передача температурного поля на АРМ ОТЗ  
    if(ModeSys == 1)  
        readHoldingRegisters(0x3, &dataRegisters[0], 105);  
    if(ModeSys == 2)  
        readHoldingRegisters(0x3, &dataRegisters[105], 105);  
    if(ModeSys == 3)  
        readHoldingRegisters(0x3, &dataRegisters[210], 105);  
    if(ModeSys == 4)  
        readHoldingRegisters(0x3, &dataRegisters[315], 105);  
  
    //Передача адрес датчиків  
    if(ModeSys == 5)  
    {  
        int segment = commandRegister[1];  
        readHoldingRegisters(0x10, &addressRegisters[70*segment], 70);  
    }  
}
```

```
//Отримання адрес датчиків
if(ModeSys == 6)
{
    segment = commandRegister[1];
    writeMultipleRegisters(0x10, &addressRegisters[70*segment], 70);
}

//Запис адрес датчиків в енергонезалежну пам'ять
if(ModeSys == 7)
{
    EEPROM.begin();
    for(int i=0; i<12; i++)
        for(int j=0; j<35; j++)
            for(int k=0; k<8; k++)
            {
                int EEPROM_address=i*35*8+j*8+k;
                EEPROM.update(EEPROM_address, AdrDB[i,j,k]);
                delay(200);
            }
    EEPROM.end();
}

//Детекція адрес датчиків
if(ModeSys == 8)
    AddrDetect();

ModeSys = 0;

}
```