

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Прикладних інформаційних технологій та електроінженерії

(повна назва факультету)

Комп'ютерно-інтегрованих технологій

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розроблення автоматизованої системи моніторингу та управління
процесом дозрівання сирів в кліматичній камері.

Виконав(ла): студент(ка) 4 курсу, групи КТс-41

спеціальності 174 – Автоматизація, комп'ютерно-

інтегровані технології та робототехніка

(шифр і назва спеціальності)

Фіськович В.В.

(підпис)

(прізвище та ініціали)

Керівник

(підпис)

Королук Р.І.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Чихіра І.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Голотенко О.С

(прізвище та ініціали)

Рецензент

(підпис)

Сисак І.М.

(прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)

Кафедра Комп'ютерно-інтегрованих технологій
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

« » (підпис) « » (прізвище та ініціали)
2026 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
(шифр і назва спеціальності)

студенту Фіськович Василь Володимирович
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення автоматизованої системи моніторингу та управління процесом дозрівання сирів в кліматичній камері.

Керівник роботи старший викладач кафедри КТ Корольок Ростислав Ігорович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «14» квітня 2026 року № № 4/9-185

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи технологічні параметри дозрівання сиру (температура, вологість, вага), характеристики кліматичної камери та перелік обраних засобів автоматизації.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

1. Аналітична частина

2. Спеціальна частина

3. Проектна частина

4. Безпека життєдіяльності, основи охорони праці

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Графічний матеріал становить 20 слайдів формату А4

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С. к.т.н. заступник зав. кафедри МТ		

7. Дата видачі завдання 20.04.2026

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
	Оформлення розділу Вступ	15.04.26	
	Оформлення розділу Аналітична частина	28.04.26	
	Оформлення розділу Спеціальна частина	04.05.26	
	Оформлення розділу Проєктна частина	24.05.26	
	Оформлення розділу безпека життєдіяльності, основи охорони праці	07.06.26	
	Підготовка до захисту	16.06.26	

Студент

(підпис)

Фіськович В.В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Королук Р.І.

(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота складається з пояснювальної записки та графічної частини.

Об'єм графічної частини кваліфікаційної роботи становить 20 слайдів. Об'єм пояснювальної записки складає 83 друкованих сторінок формату А4 (210×297), об'єм додатків – 26 друкованих сторінок формату А4.

Метою даної кваліфікаційної роботи була розробка прототипу системи автоматизованого моніторингу передбачає створення системи розумних сенсорів, які в режимі реального часу контролюють мікроклімат. При виявленні найменших відхилень від норми система негайно генерує сповіщення на смартфон. Це гарантує оперативну реакцію персоналу, повністю нівелює ризик критичних втрат та зберігає високу якість продукту.

Вирішено актуальне науково-практичне завдання — розроблено прототип системи автоматизованого моніторингу та керування кліматичною камерою дозрівання сирів для малих крафтових підприємств на базі IoT-технологій та концепції периферійних обчислень (Edge Computing).

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1.АНАЛІТИЧНА ЧАСТИНА	9
1.1.Технологічні особливості процесу дозрівання твердого сиру	9
1.2. Огляд існуючих систем моніторингу та автоматизації у контексті малого виробництва	12
1.3. Концепція IoT (Internet of Things) у контексті малого харчового виробництва.....	16
1.4.Порівняльний аналіз технологій передачі даних (Wi-Fi, Bluetooth, LoRaWAN, Zigbee) для виробничих приміщень.....	18
2.СПЕЦІАЛЬНА ЧАСТИНА	24
2.1. Вибір обладнання.....	24
2.1.1. Сімейство одноплатних комп'ютерів Rasbery Pi	24
2.1.2. Аналіз вагових датчиків	27
2.1.3. Аналіз датчиків температури та вологості	28
2.1.4. Аналіз нагрівачів, осушувачів та зволожувачів	32
2.2.Структурна схема підключення	37
2.3 Вибір платформи для зберігання даних	41
3.ПРОЄКТНА ЧАСТИНА	44
3.1. Розробка алгоритму роботи камери дозрівання сирів	44
3.2. Розробка електричної схеми камери дозрівання	44
3.3 Розробка програмного забезпечення мікроконтролера	47

3.4. Тестування та аналіз даних	49
4.БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	51
4.1. Актуальність безпеки життєдіяльності людини	51
4.2.Санітарно-гігієнічні вимоги до умов праці	56
4.3.Пожежна профілактика на робочому місці	53
Висновки	55
Перелік посилань.....	57
Додатки.....	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АЦП - (аналого-цифровий перетворювач) - це електронний пристрій, який перетворює неперервний аналоговий сигнал у цифровий код, який може обробляти комп'ютер чи мікроконтролер.

GPIO - (General Purpose Input/Output - ввід/вивід загального призначення) - це універсальні цифрові контакти (піни) на платах мікрокомп'ютерах (як-от Raspberry Pi), які використовуються для зв'язку з різними електронними компонентами.

MQTT - (Message Queue Telemetry Transport) - це легковажний мережевий протокол для обміну повідомленнями, який працює поверх TCP/IP.

Афінаж сиру - це процес контрольованого дозрівання сиру, який відповідає за смак, колір, запах, текстуру та зовнішній вигляд.

Регулювання - це процес управління, упорядкування або контролю певних суспільних, економічних чи технічних процесів.

Н_{поточ} - Поточний рівень вологості в кліматичній камері

Н_{задана} - Заданий рівень вологості в кліматичній камері

Т_{поточ} - Поточний рівень температури в кліматичній камері

Т_{задана} - Заданий рівень температури в кліматичній камері

ВСТУП

Актуальність теми: Процес дозрівання сиру це один із складних етапів, від якого залежить смак, текстура та готовність продукту. Навіть незначні відхилення температури та вологості може призвести появи небажаної мікрофлори та псування цілої партії дороговартісної продукції. Звичні системи автоматизації камер дозрівання нерідко обмежуються локальним підтриманням температури та вологості.

Звичні системи автоматизації камер дозрівання мають вагомі вади: відсутність віддаленого доступу до даних, складність масштабування та неможливість аналізу статистики. Впровадження технологій Internet of Things (IoT) дозволяє перетворити камеру у «розумний» об'єкт. Це забезпечує безперервний моніторинг 24/7, мінімізує вплив «людського фактору» та дозволяє запобігти збиткам ще на ранніх стадіях відхилення режиму.

Основна складність - підтримання точних параметрів вологості (90-95%) та температури (10-12°C). Традиційні методи ігнорують нерівномірний розподіл повітря. Впровадження IoT-системи створює розподілену мережу датчиків, яка формує повну «мапу мікроклімату», забезпечуючи однакові умови для кожної головки сиру на кожній полиці. Об'єкт дослідження: Моніторинг процесу дозрівання сиру в кліматичній камері.

Предмет дослідження: Методи та засоби моніторингу за такими показниками: температура, вологість, вага.

Мета роботи: Розробка прототипу системи автоматизованого моніторингу передбачає створення системи розумних сенсорів, які в режимі реального часу контролюють мікроклімат. При виявленні найменших відхилень від норми система негайно генерує сповіщення на смартфон. Це гарантує оперативну реакцію персоналу, повністю нівелює ризик критичних втрат та зберігає високу якість продукту.

1 АНАЛІТИЧНА ЧАСТИНА

1.1 Технологічні особливості процесу дозрівання твердого сиру

Історичні аспекти та сучасна ревіталізація крафтового сироваріння. Сироваріння є одним із найдавніших біотехнологічних процесів у людській історії, перші згадки про який датуються близько 4000-2300 рр. до н.е. Еволюція технології пройшла шлях від первісного випадкового зсідання молока під впливом природних ферментів у кочових племен до цілеспрямованого виведення нових сортів у середньовічних європейських монастирях, де започаткували використання специфічних видів культивованої плісняви, копчення та спецій [1].

В Україні традиції сироваріння сягають кількох століть назад, коли молоко у сільських господарствах перетворювали на поживні продукти тривалого зберігання. За часів Київської Русі селяни виготовляли прості розсільні сири на основі натуральних ферментів і солі (типу сучасної бринзи). Згодом ці традиції трансформувалися в унікальні технології Карпатського регіону, які за своїми характеристиками (текстурою та ароматом) подібні до швейцарських аналогів [2].

У сучасних умовах в Україні спостерігається активний розвиток малих фермерських підприємств та крафтових сироварень. На відміну від масового індустріального виробництва, малі господарства орієнтуються на створення нестандартних, вишуканих та ексклюзивних продуктів, поєднуючи ручну працю («hand-made») з автентичними рецептурами (додавання чебрецю, часнику, горіхів тощо). Проте дрібносерійне виробництво є дуже чутливим до стабільності якісних показників, оскільки будь-яке відхилення параметрів середовища може призвести до браку всієї партії продукту [2].

Фізико-хімічні основи підготовки сировини в малому виробництві. Технологічні властивості готового продукту безпосередньо залежать від вхідної сировини - молока коров'ячого, козячого, овечого або їхніх комбінацій. Для крафтових виробників характерне активне експериментування з сумішами різних

видів молока з метою досягнення ідеального балансу смаку та кремової текстури. Окрім молочної основи, у виробничому циклі малих сироварень застосовують таку групу обов'язкових компонентів:

- сичужний фермент (тваринного походження або його замінники): виступає головним каталізатором коагуляції казеїну та перетворення молока на густу масу;
- бактеріальні закваски (молочнокислі бактерії): забезпечують процес спрямованого бродіння, визначають рівень кислотності (рН) та формують специфічний профіль (ніжний або гострий);
- хлорид натрію (сіль): використовується для покращення смакових якостей, консервації продукту та стимуляції утворення захисної скоринки;
- органолептичні добавки: трави, спеції, горіхи чи культури шляхетної блакитної плісняви, які надають кожному крафтовому сорту неповторної індивідуальності [1].

Етапи технологічного процесу виробництва на міні-сироварнях. Процес отримання твердого сиру в умовах малих підприємств складається з таких послідовних етапів:

1) пастеризація та нормалізація сировини: вхідне молоко піддається термічній обробці для деактивації небажаної мікрофлори. В умовах крафтового виробництва на цьому етапі особливу увагу приділяють контролю рівня рН за допомогою автоматизованих дозаторів для запобігання технологічним дефектам;

2) коагуляція (згортання молока): у підготовлену сировину вносять ферменти й закваски, що призводить до формування щільного згустку;

3) постановка та обробка сирного зерна: отриманий згусток механічно подрібнюють для інтенсифікації відділення вільної сироватки;

4) формування та пресування: сирну масу відокремлюють від сироватки та розподіляють за формами. Для отримання твердої структури застосовують тривале пресування, яке забезпечує видалення залишків міжзернової рідини та ущільнення головки;

5) посолка: сформовані блоки занурюють у соляні розчини для зупинки активного бродіння та первинного формування скоринки;

6) дозрівання (афінаж): фінальна стадія, що триває від кількох місяців до року.

На малих підприємствах цей етап часто намагаються організувати в природних умовах (наприклад, у печерах чи спеціальних підвалах Карпат), де природне середовище додає глибини смаку. Проте відсутність технічного контролю в таких приміщеннях створює значні ризики.

Специфіка впливу мікроклімату на афінаж та економічна чутливість малих підприємств. Етап дозрівання є найбільш критичним вікном у всьому виробничому циклі. Для більшості класичних твердих сирів технологічний регламент вимагає підтримання стабільної температури в межах від +11 до +13°C (допустимий діапазон від 8 до 12°C) та відносної вологості повітря 80 - 90% (в окремих випадках від 90 до 95%).

Для власника малого чи крафтового сироварного бізнесу система моніторингу параметрів дозрівання є головним запобіжником проти критичних фінансових збитків. Якщо великі промислові заводи можуть нівелювати втрату кількох партій за рахунок великих обсягів, то для мікропідприємства брак однієї камери дозрівання означає тотальні збитки. У таблиці 1.1 наведено наслідки та вплив технологічних відхилень під час дозрівання сиру для крафтових сироварень

Таблиця 1.1 - Наслідки та вплив технологічних відхилень під час дозрівання сиру для крафтових сироварень

Технологічне відхилення	Фізико-хімічний наслідок для продукту	Вплив на бізнес-показник крафтової сироварні
Надмірна вологість / застій повітря	Активне виділення сироватки, розвиток патогенних мікроорганізмів, дикої плісняви,	Повне псування дороговартісної партії продукції, штрафи

	процесів гниття або поява сирного кліща (сальмонела, лістеріоз).	контролюючих органів, судові позови споживачів.
Недостатня вологість / жорсткий нагрів	Інтенсивне випаровування вологи з поверхні головки, утворення тріщин та ефект «закалу» (передчасне висихання скоринки, що блокує ферментацію всередині).	Втрата масової частки готової продукції (надмірна усушка), погіршення товарного вигляду, рекламациї від дистриб'юторів.

Традиційні камери дозрівання на малих підприємствах часто обмежуються локальними побутовими приладами, які ігнорують нерівномірний розподіл повітряних потоків на різних полицях. Для крафтового сектору критично важливо мати розподілену мережу датчиків, яка формує повну «мапу мікроклімату».

Впровадження доступних систем автоматизації на базі концепції Інтернету речей (IoT) дозволяє малим господарствам забезпечити безперервний моніторинг 24/7, мінімізувати вплив «людського фактору» (ручних перевірок) та запобігти збиткам ще на ранніх стадіях відхилення режиму, гарантуючи стабільну високу якість кожної головки сиру.

1.2 Огляд існуючих систем моніторингу та автоматизації у контексті малого виробництва

Забезпечення інваріантності параметрів мікроклімату в камерах афінажу є визначальним фактором у формуванні стабільних органолептичних, текстурних та біохімічних характеристик твердих і напівтвердих сирів. Традиційний комплекс контролю якості на підприємствах зазвичай диференціюється на кілька рівнів: від суб'єктивного візуально-органолептичного моніторингу до регламентованих

фізико-хімічних, токсикологічних та мікробіологічних лабораторних досліджень. Проте, якщо великі індустріальні об'єкти мають можливість нівелювати фінансові втрати від локального браку за рахунок значних обсягів виробництва, то для суб'єктів малого та крафтового підприємництва девіація температурно-вологісного режиму навіть в одній камері може призвести до критичних збитків і втрати всієї партії дороговартісної продукції.

З огляду на це, автоматизація контролю параметрів середовища (стабілізація температури в межах 8 - 12 °C та відносної вологості 80 - 95%) є першочерговим завданням для мінімізації впливу «людського фактора» та превенції розвитку патогенної мікрофлори чи небажаних видів плісняви. У сучасній практиці автоматизації технологічних процесів харчових виробництв виділяють три основні концептуальні підходи до побудови систем моніторингу, кожен з яких має суттєві обмеження при впровадженні у секторі малого бізнесу.

Промислові програмовані логічні контролери (ПЛК). До цієї групи належать серійні кліматичні рішення від провідних брендів (Carel серії MPXPRO, Siemens Climatix, Eliwell, ОВЕН). Вони є галузевим стандартом для великих промислових холодильних комплексів та складів.

- переваги: висока апаратна надійність, стійкість до агресивного та вологого середовища (захист за стандартами IP65/IP67), наявність сертифікатів відповідності для харчових виробництв.

- недоліки: капітальні витрати на закупівлю, Проектування, програмування та сервісне обслуговування таких систем є надмірними для крафтових сироварень. Крім того, промислові ПЛК мають закриту архітектуру, що унеможливорює або значно здорожує інтеграцію нестандартних вимірювальних каналів - зокрема, прецизійного тензометричного моніторингу динаміки усушки сиру - без розгортання складних і коштовних промислових мереж (Modbus/Profibus).

Побутові та напівпромислові комерційні IoT-екосистеми. Застосування готових модулів автоматизації для «розумного будинку» (на базі платформ TuYa Smart, Sonoff, Smart Life) приваблює малих виробників мінімальною вартістю та простотою розгортання графічних інтерфейсів користувача.

- переваги: відсутність потреби у глибоких знаннях програмування, низький фінансовий поріг входження, наявність інтегрованих бездротових інтерфейсів (Wi-Fi/Zigbee).

- недоліки: низька деградаційна стійкість та малий життєвий цикл сенсорів в умовах постійної екстремальної вологості (до 95%). Відсутня можливість гнучкого налаштування локальних алгоритмів регулювання (наприклад, PID-регулювання розподілених гнучких нагрівачів). Найбільш критичним для виробництва є повна залежність від наявності зовнішнього інтернет-зв'язку із хмарними серверами розробника платформи, що суперечить вимогам до автономності локальних обчислювальних вузлів (Edge Computing).

Спеціалізовані академічні та DIY-розробки на базі мікроконтролерів

У сучасній науково-технічній літературі активно досліджується концепція ретрофітінгу (модернізації та дооснащення) традиційних камер дозрівання за допомогою відкритих архітектур Інтернету речей (IoT), мікроконтролерів (ESP32, Arduino) та одноплатних комп'ютерів типу Raspberry Pi. Головний фокус проекту - відстеження та контроль втрати маси сиру (cheese mass loss) під час дозрівання [3].

- переваги: максимальна гнучкість проектування, використання відкритих схемотехнічних рішень та програмного коду з низьким рівнем інвестицій порівняно з пропрієтарними системами. Досвід впровадження подібних систем на крафтових підприємствах доводить можливість зменшення флуктуацій температури в камерах дозрівання на 90%, що забезпечує високу повторюваність результатів та стабільну якість продукції[5]. Платформи дозволяють легко поєднувати кліматичні та тензометричні сенсори, а також організовувати прямі адаптивні канали сповіщення користувачів.

- недоліки: більшість описаних у літературі систем розглядаються відірвано від комплексної специфіки технології афінажу. Вони концентруються виключно на макропараметрах повітряного середовища (температура, вологість), ігноруючи математичний аналіз добового градієнта втрати маси продукту (швидкості усушки) в реальному часі, що є головним технологічним індикатором правильного проходження процесу дозрівання сирної головки та захисту від утворення «закалу».

Для проведення порівняльного аналізу та обґрунтування вибору вектору проектування у таблиці 1.2 систематизовано характеристики існуючих рішень та запропонованої системи автоматичного керування (САК).

Аналітичний огляд існуючих технічних рішень свідчить про наявність вираженого технологічного та економічного бар'єра для суб'єктів малого бізнесу: промислові системи є фінансово недоступними та функціонально закритими, тоді як бюджетні побутові пристрої не забезпечують належної надійності та комплексного аналізу даних. Це обґрунтовує доцільність розробки спеціалізованого енергоефективного прототипу САК на базі мікроконтролера Raspberry Pi Pico, який дозволить крафтовим сироварням без суттєвих капітальних витрат реалізувати концепцію Edge Computing, сумістивши прецизійний моніторинг макроклімату із безперервним відстеженням маси продукту.

Таблиця 1.2 - Порівняльний аналіз систем моніторингу кліматичних камер у контексті крафтового виробництва

Критерій порівняння	Промислові ПЛК (Carel / Siemens)	Комерційні IoT-екосистеми (Tuya / Sonoff)	Запропонована САК на базі Raspberry Pi Pico
---------------------	----------------------------------	---	---

Комплексний кліматичний контроль (T, Rh)	Висока точність, проте висока вартість інфраструктури	Середня точність, обмежений ресурс сенсорів у вологому середовищі	Висока точність (BME280/DHT22), адаптація до умов вологості
Моніторинг динаміки маси (усушки)	Вимагає додаткових коштовних модулів спряження	Функціонал повністю відсутній	Інтегрований (тензодатчик + АЦП NH711 з аналізом градієнта ваги)
Рівень капітальних інвестицій	Критично високий для малого бізнесу	Мінімальний	Низький (доступна та відкрита елементна база)
Архітектурна автономність (Edge Computing)	Повна (локальне виконання алгоритмів у контролері)	Відсутня (високий ризик відмови при збоях інтернет-зв'язку)	Повна (локальна буферизація у флеш-пам'ять при збоях зв'язку)
Канал віддаленого сповіщення оператора	Складні SCADA-системи або дорогі SMS-шлюзи	Стандартні системні push-повідомлення додатка	Гнучкий інтерактивний без проміжних серверів

1.3 Концепція IoT (Internet of Things) у контексті малого харчового виробництва

Впровадження концепції Інтернету речей (IoT) у секторі малого та крафтового харчового виробництва має свою виражену специфіку, що відрізняє його від масштабних індустріальних проєктів. Для мікропідприємств та сімейних ферм головними критеріями автоматизації є низька вартість рішень, висока автономність

та простота обслуговування. У таких умовах класична трирівнева модель IoT (датчик - хмара - користувач) зазнає трансформації у бік периферійних обчислень (Edge Computing).

Використання мікроконтролерів класу Raspberry Pi Pico як центральних вузлів локальної IoT-мережі дозволяє реалізувати ключові переваги цієї концепції без залучення дорогих серверних потужностей:

1. Автономність та живучість системи (Edge AI/Computing): На відміну від побутових IoT-екосистем, які повністю залежать від стабільності інтернет-зв'язку із зовнішніми серверами, архітектура на базі локального мікроконтролера виконує алгоритми двопозиційного чи PID-регулювання безпосередньо на об'єкті. У разі аварії чи збою мережі на фермі, камера дозрівання сиру продовжує функціонувати в штатному режимі, накопичуючи логи у внутрішню пам'ять.

2. Прецизійний локальний моніторинг: Для малих господарств критично важливо контролювати не просто усереднені параметри приміщення, а мікроклімат на рівні кожної полиці. Концепція IoT дозволяє розгорнути розподілену мережу цифрових датчиків, що збирають інформацію про температуру, вологість та динаміку втрати маси продукту (усушку) в реальному часі.

3. Гнучкість та інтеграція з мобільними сервісами: Замість розгортання складних промислових інтерфейсів (SCADA-систем), концепція малого IoT орієнтована на пряму взаємодію з оператором через популярні API-інтерфейси месенджерів. Це забезпечує миттєве push-оповіщення про критичні відхилення параметрів середовища, що мінімізує вплив людського фактора.

Таким чином, для крафтового сироваріння IoT є не просто інструментом віддаленого перегляду графіків, а надійною та доступною технологією забезпечення стандартів безпеки (НАССР), яка гарантує стабільну якість продукції за мінімальних капітальних інвестицій.

1.4 Порівняльний аналіз технологій передачі даних (Wi-Fi, Bluetooth, LoRaWAN, Zigbee) для виробничих приміщень.

У сучасному світі інформаційно-комунікаційних технологій, що динамічно розвиваються, передача даних стає ключовим елементом у процесі створення та розповсюдження інформації, забезпечуючи це ефективно й безпечно.

Передача даних - це процес обміну цифровою інформацією між двома або більше пристроями через певне середовище передачі. В основі цього процесу лежить передача бітів (найменших одиниць двійкової інформації) за допомогою електричних, електромагнітних сигналів або світлових хвиль. У контексті бездротових технологій це означає передачу інформації на відстань за допомогою радіохвиль, інфрачервоного, оптичного або лазерного випромінювання

Для того щоб передача даних відбулася успішно, необхідні кілька базових компонентів:

1. Середовище передачі: Це канал, по якому переміщуються дані. Воно може бути фізичним (керованим), наприклад, мідні чи коаксіальні кабелі та волоконна оптика, або бездротовим (некерованим), де сигнал передається через повітря чи порожній простір у вигляді радіохвиль.

2. Пристрої відправлення та отримання: Це обладнання, що ініціює та приймає сигнал (наприклад, комп'ютери, смартфони, маршрутизатори, модеми, мережеві комутатори, а також датчики).

3. Комунікаційні протоколи та стандарти: Це суворі набори правил і технічних процедур, які визначають формат та спосіб обміну даними між пристроями. Прикладами протоколів є TCP/IP (для Інтернету) або HTTP (для веб-сайтів), а стандартів - Wi-Fi, Ethernet та Bluetooth.

Wi-Fi (WLAN): Це високошвидкісний стандарт для побудови локальних мереж, який працює у частотних діапазонах 2,4 ГГц та 5 ГГц Залежно від стандарту

(наприклад, 802.11n), швидкість передачі може сягати 300 Мбіт/с на відстані 20-50 метрів у приміщеннях. Найчастіше застосовується для доступу до Інтернету, передачі великих файлів та відеопотоків.

Bluetooth (WPAN): Технологія для створення персональних бездротових мереж на невеликих відстанях (від 10 до 100 метрів), що працює в неліцензійному діапазоні 2,4 ГГц. Має нижчу швидкість (до 3 Мбіт/с) і зазвичай використовується для заміни дротових з'єднань між обладнанням, передачі невеликих файлів або аудіо. Щоб уникати перешкод від Wi-Fi, сучасний Bluetooth використовує технологію адаптивної зміни частоти (AFH)

Zigbee: Технологія, побудована на базі стандарту IEEE 802.15.4, яка спеціально створена для пристроїв із надзвичайно низьким енергоспоживанням - датчики можуть працювати від однієї батареї від кількох місяців до років. Швидкість передачі даних низька (до 250 Кбіт/с), проте Zigbee дозволяє будувати складні та надійні мережі (наприклад, топологію mesh/«кожен з кожним»), де кожен вузол може передавати інформацію далі. Це робить його ідеальним для промислового моніторингу та керування, автоматизації будівель, систем безпеки та мереж датчиків.

LoRaWAN: Енергоефективна технологія дальнього радіусу дії, яка активно застосовується для дистанційного обліку ресурсів (води, електроенергії, газу, тепла). Батарейні лічильники на базі LoRaWAN здатні стабільно працювати у складних умовах, наприклад, у холодних неопалюваних будівлях або на віддалених об'єктах. Завдяки збору даних через LoRaWAN на підприємствах також можна використовувати машинне навчання для прогнозування витоків ресурсів та контролю піків їх споживання.

У таблиці 1.3, таблиці 1.4, таблиці 1.5 наведено порівняння характеристик технологій передачі даних для виробничих приміщень, які були описані вище.

Таблиця 1.3 - Порівняння частотного діапазону та пропускної здатності технологій передачі даних для виробничих приміщень

Технологія	Стандарт IEEE	Частотний діапазон (ГГц)	Пропускна здатність / Швидкість
Wi-Fi	802.11n	2.4, 5.0	До 300-480 Мбіт/с
Zigbee	802.15.4	2.4, 0.868, 0.915	20-250 Кбіт/с
LoRaWAN	Proprietary / LoRa Alliance	0.868 (Субгігагерцовий діапазон)	290 біт/с - 50 Кбіт/с
Bluetooth	802.15.1 / 802.15.3 (EDR)	2.4	До 3 Мбіт/с (версія 2.0+EDR)

Таблиця 1.4 - Порівняння дальності та енергоспоживання технологій передачі даних для виробничих приміщень

Технологія	Дальність дії в приміщенні (м)	Енергоспоживання / Час роботи
Wi-Fi	10-50	Високе (12-120 год від батареї)
Zigbee	10-100	Низьке (2400-24000 год від батареї)
LoRaWAN	2000-5000 в місті; до 15000 поза містом	Дуже низьке (роки роботи від батареї)
Bluetooth	10-100	Середнє (24-240 год від батареї)

Таблиця 1.5 - Порівняння топології мережі та сфер застосування технологій передачі даних для виробничих приміщень

Технологія	Топологія мережі	Сфери застосування на виробництві
Wi-Fi	Точка-точка, WLAN	Побудова локальних мереж, передача великих даних, відео, мультимедіа
Zigbee	Зірка, Mesh, кластерне дерево, кожен з кожним	Промисловий моніторинг і керування, мережі датчиків, автоматизація
LoRaWAN	Зірка (через шлюзи)	Дистанційний облік ресурсів, моніторинг датчиків, промисловий інтернет речей
Bluetooth	Точка-точка, точка-багато точок, пікомережі (piconet), scatternet	Заміна дротового з'єднання, передача аудіо, файли невеликого об'єму

Вибір бездротової технології для виробничих приміщень сироварні безпосередньо впливає на вартість розгортання інфраструктури та подальше обслуговування системи:

1. Wi-Fi: Має високу швидкість, але обмежений радіус (20-50 м) і високе енергоспоживання. Вимагає значних інвестицій у побудову стабільної мережі в умовах залізобетонних камери сироварні.

2. Bluetooth: Економічно вигідний для локальних рішень (контроль окремого об'єкта на відстані до 100 м), але не підходить для масштабних підприємств через обмежену топологію.

3. Zigbee: Оптимальне рішення для внутрішньої автоматизації підприємства. Завдяки mesh-топології (кожен датчик передає сигнал далі) покриває великі виробничі площі без купівлі дорогого потужного обладнання. Низьке

енергоспоживання дозволяє датчикам працювати роками на одній батареї, що критично зменшує сервісні витрати.

4. LoRaWAN: Найкращий вибір для територіально розподіленого бізнесу (наприклад, якщо камери дозрівання, ферми та склади розташовані на відстані кількох кілометрів). Технологія дозволяє збирати дані у складних, неопалюваних умовах і є фундаментом для впровадження предиктивної аналітики щодо витрат ресурсів.

У першому розділі кваліфікаційної роботи проведено комплексний науково-технічний та аналітичний огляд об'єкта автоматизації, досліджено існуючі підходи до побудови кліматичних систем та обґрунтовано концепцію проектування САК. За результатами проведеного аналізу сформульовано такі основні висновки:

- Досліджено технологічні особливості процесу дозрівання (афінажу) твердого сиру та встановлено, що жорстке підтримання температури в межах +11...+13 °C та відносної вологості повітря 80-95 % є визначальними факторами формування якості готового продукту. Виявлено, що для малих крафтових підприємств левітація цих параметрів або їх нерівномірний розподіл на полицях камери спричиняють критичні фінансові збитки через ризик псування всієї партії (розвиток патогенів, поява дикої плісняви або утворення дефекту «закалу»). Обґрунтовано, що головним технологічним індикатором правильного перебігу процесу є добовий градієнт втрати маси продукту (швидкість усушки), що вимагає впровадження безперервного тензометричного моніторингу.

- Проведено критичний аналіз існуючих засобів автоматизації. Доведено, що промислові ПЛК (Carel, Siemens, ОВЕН) мають надмірну для малого бізнесу вартість та закриту архітектуру, яка унеможлиблює бюджетну інтеграцію прецизійних тензодатчиків. Водночас комерційні побутові IoT-екосистеми (TuYa, Sonoff) характеризуються низькою деградаційною стійкістю сенсорів в умовах екстремальної вологості та повною залежністю від зовнішніх хмарних серверів, що загрожує зупинкою контуру регулювання при збоях інтернет-зв'язку.

- Обґрунтовано доцільність застосування концепції Інтернету речей (IoT) на принципах периферійних обчислень (Edge Computing) для автоматизації малих харчових виробництв. Доведено, що перенесення обчислювального навантаження безпосередньо на об'єкт за допомогою мікроконтролерів дозволяє реалізувати автономні та живучі локальні контури керування, мінімізувати вплив «людського фактора», забезпечити відповідність стандартам безпеки НАССР та уникнути капітальних витрат на оренду серверних потужностей.

- На основі порівняльного аналізу бездротових технологій передачі даних (Wi-Fi, Bluetooth, LoRaWAN, Zigbee) в умовах залізобетонних приміщень сироварні визначено, що оптимальним рішенням для локальної кліматичної камери є інтерфейс Wi-Fi. Він забезпечує достатню пропускну здатність для трансляції телеметрії та дозволяє організувати пряму інтеграцію з мобільними сервісами оператора без використання проміжних серверів.

- Сформовано технічні вимоги та визначено апаратний вектор подальшого проектування системи. Для реалізації енергоефективного прототипу САК обґрунтовано вибір мікроконтролера Raspberry Pi Pico W, високоточного цифрового датчика мікроклімату DHT22, тензOMETричного модуля АЦП НХ711, релейного модуля Waveshare Pico-Relay-V для гальванічної розв'язки силових контурів (компресор, ІЧ-нагрівач, ультразвуковий зволожувач, елементи Пельтьє).

2 СПЕЦІАЛЬНА ЧАСТИНА

2.1 Вибір обладнання

Успішна реалізація автоматизованої системи моніторингу дозрівання сиру значною мірою залежить від ретельного й оптимального вибору апаратного забезпечення. Основними критеріями, на які орієнтується підприємець під час вибору, є наступні:

1. Капітальні витрати: оцінка вартості датчиків і мікроконтролерів на етапі їх закупівлі.
2. Операційні витрати: врахування енергоефективності, простоти обслуговування та можливості заміни компонентів без необхідності залучення дорогих вузькопрофільних фахівців.
3. Надійність в умовах агресивного середовища: враховуючи високу вологість камер дозрівання сиру (до 95%), апаратура повинна бути стійкою до корозії та конденсату.

Для створення прототипу та подальшого впровадження комерційної системи автоматизації прийнято рішення використовувати архітектуру, що включає центральний обчислювальний вузол на базі одноплатного комп'ютера, тензометричну систему для контролю ваги (з метою оцінки рівня усушки продукту), а також прецизійні датчики мікроклімату.

2.1.1 Сімейство одноплатних комп'ютерів Rasbery Pi

Для автоматизації збору даних і управління кліматичними системами замість традиційних промислових контролерів (ПЛК), що відзначаються високою вартістю та складністю у програмуванні, доцільніше застосовувати одноплатні комп'ютери серії Raspberry Pi.

З бізнесової точки зору, Raspberry Pi виконує роль пристрою периферійних обчислень (Edge Computing), що дає змогу обробляти дані безпосередньо на об'єкті.

Це забезпечує автономну роботу, наприклад, камери дозрівання, навіть за тимчасової відсутності доступу до інтернету. Завдяки великій спільноті користувачів, підтримці Linux-операційних систем і наявності GPIO-контактів, система має можливість швидко масштабуватися, підключати велику кількість датчиків та легко інтегрувати різне обладнання з хмарними платформами. Серед різної продукції компанії Raspberry Pi Foundation для цих задач справиться Raspberry Pi Pico.

Raspberry Pi Pico -компактний пристрій, що ефективно поєднує можливості потужних одноплатних комп'ютерів та класичних мікроконтролерів. Завдяки доступній ціні та високій надійності, це ідеальне рішення для створення мережі датчиків у крафтових і промислових сироварнях.

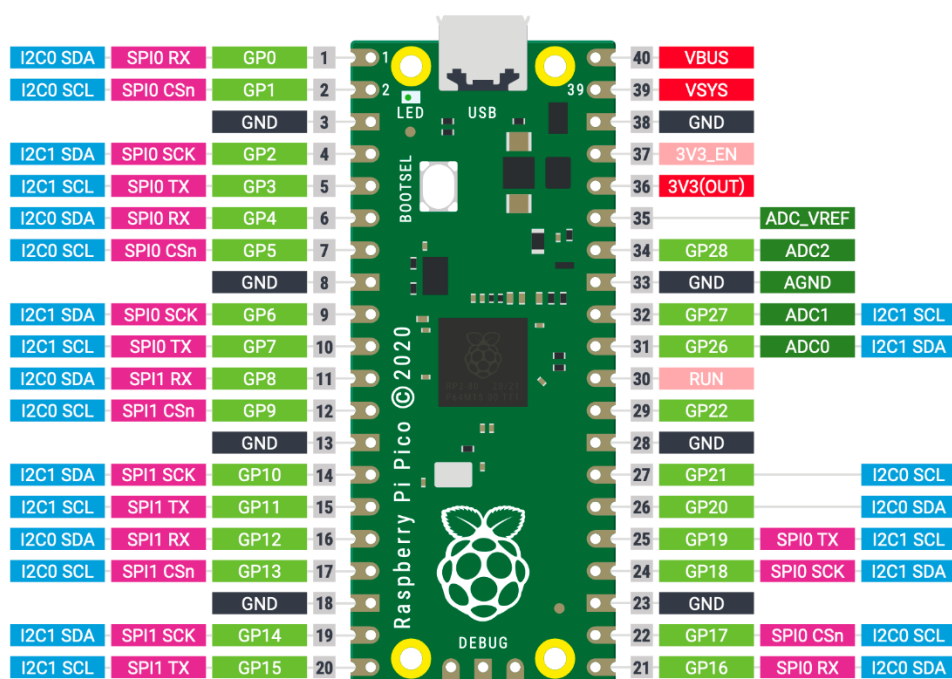


Рисунок 2.1 - Raspberry Pi Pico

Основні технічні характеристики Raspberry Pi Pico:

1. Обчислювальні можливості: Пристрій базується на мікросхемі RP2040 із двоядерним процесором Arm Cortex-M0+, який працює на частоті до 133 МГц. Цього достатньо для одночасної обробки даних із декількох сенсорів, таких як датчики ваги, температури та вологості, у режимі реального часу.

2. Пам'ять: Об'єм оперативної пам'яті становить 264 КБ SRAM, а вбудована флеш-пам'ять дорівнює 2 МБ (розширюється до 16 МБ). Це дозволяє зберігати локальні журнали даних навіть у разі перебоїв зі зв'язком.

3. Периферійні можливості: Плата оснащена 26 багатофункціональними GPIO-пінами і підтримує інтерфейси UART, I2C та SPI. Такий набір дозволяє підключати цілі групи сенсорів і виконавчих пристроїв, таких як реле або вентилятори, до одного контролера.

4. Компактність: Розмір плати -лише 51x21 мм, що робить її зручною для прихованого монтажу в герметичних захисних корпусах. Це дозволяє застосовувати плату навіть у приміщеннях із до 95% вологості (наприклад, у камерах дозрівання сиру). Наявність USB-порту спрощує як живлення пристрою, так і його перепрошивку або налагодження.

Переваги для бізнесу при використанні Raspberry Pi Pico:

1. Зменшення капітальних витрат: Вартість плати значно нижча за повноцінні моделі Raspberry Pi 4 або 5, що забезпечує економію бюджету при оснащенні обладнанням окремих камер або полиць.

2. Швидкість розробки: Raspberry Pi Pico можна програмувати за допомогою MicroPython або C/C++, використовуючи середовища командного рядка чи Visual Studio. Ця гнучкість спрощує процес створення системи як для новачків, так і для досвідчених розробників, скорочуючи час виходу продукту на ринок.

3. Надійність і стабільність: Завдяки високій якості збірки та відсутності операційної системи, яка могла б вплинути на роботу контролера, плата забезпечує безперебійний моніторинг в режимі 24/7. Це мінімізує ризик втрати дорогих ресурсів через технічні неполадки.

4. Такий підхід демонструє ефективність пристрою для автоматизації виробничого процесу, забезпечуючи оптимальне співвідношення вартості та продуктивності.

2.1.2 Аналіз вагових датчиків

Тензометричний датчик, або тензодатчик (назва походить від латинського слова "tensus", що означає "напружений"), є пристроєм, який перетворює деформацію в зручний для аналізу сигнал, зазвичай електричний, переважно у вигляді напруги. Існує багато методів вимірювання деформацій, серед яких тензорезистивний, п'єзоелектричний, оптико-поляризаційний, п'єзорезистивний, волоконно-оптичний, а також просте зчитування показань із механічного тензодатчика за допомогою шкали. З усіх електронних варіантів найбільшу популярність отримали тензорезистивні датчики, які широко застосовуються і саме такий тип використовується у приладі, що розглядається.

Тензорезистивний датчик зазвичай складається зі спеціально спроектованої пружної конструкції, на якій закріплений тензорезистор разом із іншими допоміжними елементами. Після калібрування можна визначити зміну опору тензорезистора і, відповідно, ступінь деформації. Ця величина безпосередньо залежить від сили, яка прикладається до конструкції.

Контроль ваги сирної головки під час її дозрівання відіграє ключову роль у забезпеченні економічної ефективності сироварні. Природні втрати вологи (усушка) безпосередньо впливають на обсяг готової продукції, а також на фінансові показники підприємства. У той же час надмірна або занадто швидка усушка може сигналізувати про порушення технологічного процесу, що призводить до появи дефектів.

Для автоматизації цього контролю рекомендується застосування тензометричних датчиків (Load Cells).

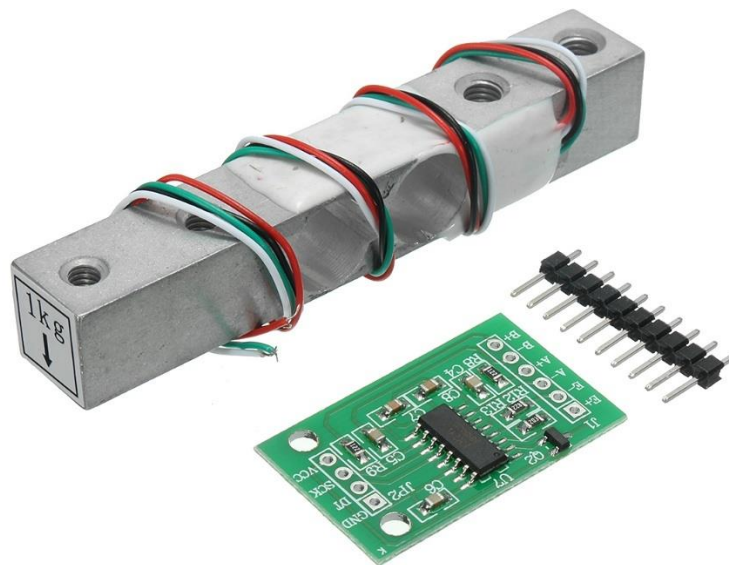


Рисунок 2.2 - Тензометричний датчик та HX711

У рамках даного аналізу було детально розглянуто різні типи цих датчиків (балочні, S-подібні, платформні) та електронні підсилювачі сигналу, включаючи популярний модуль HX711. Вибір найкращих компонентів здійснено на основі таких критеріїв, як тривала стабільність показників при постійній масі сиру, стійкість до підвищеної вологості та доступність за ціною, що дозволяє встановити датчики для кожної полиці чи стелажа в камерах дозрівання.

2.1.3 Аналіз датчиків температури та вологості

Для реалізації завдань моніторингу мікрокліматичних параметрів в межах сучасних вбудованих систем найбільш широко застосовуються цифрові датчики для вимірювання температури та вологості. На ринку серед найбільш поширених рішень, які поєднують доступність, простоту інтеграції та достатню для багатьох завдань точність, виділяються сенсорні модулі серії DHT (DHT11, DHT22) та високоточний сенсор BME280. Кожен із зазначених пристроїв характеризується

власними архітектурними особливостями, протоколами передавання даних та технічними характеристиками, які визначають їх доцільність використання в залежності від специфіки конкретного проєкту.

Короткий опис характеристик зазначених сенсорів:

1. DHT11: Це базовий та економічно вигідний датчик, який доцільно використовувати в освітніх проєктах і системах, де не висуваються високі вимоги до точності чи діапазону вимірювань (наприклад, для примітивних побутових метеостанцій). Для обміну даними з мікроконтролером використовує унікальний однопровідний цифровий протокол.

2. DHT22 (також відомий під назвою AM2302): Покращена версія датчика DHT11, що забезпечує суттєво більший діапазон вимірювань і вищу точність. Проте цей прилад характеризується більшими фізичними розмірами та потребує більшого часу для відновлення між двома зчитуваннями даних (мінімальний інтервал - 2 секунди). Аналогічно до попередньої моделі, DHT22 використовує однопровідний інтерфейс для передавання інформації.

3. BME280: Це високоточний сенсор, розроблений компанією Bosch, який здатен вимірювати не лише температуру і вологість, але й атмосферний тиск. Його основними перевагами є висока точність, низьке енергоспоживання та підтримка стандартних промислових інтерфейсів I2C та SPI. Дані характеристики роблять BME280 ідеальним рішенням для професійних метеостанцій та портативних пристроїв із підвищеними вимогами до функціональності.

У табл. 2.1 порівняльна характеристика датчиків вологості

Таблиця 2.1 - Характеристики датчиків DHT11, DHT22 та BME280

Характеристика	DHT11	DHT22 (AM2302)	BME280
Діапазон вимірювання температури	0°C ... +50°C	-40°C ... +80°C	-40°C ... +85°C

Точність (температура)	$\pm 2.0^{\circ}\text{C}$	$\pm 0.5^{\circ}\text{C}$	$\pm 1.0^{\circ}\text{C}$
Діапазон вимірювання вологості	20% ... 80%	0% ... 100%	0% ... 100%
Точність (вологість)	$\pm 5\%$	$\pm 2\%$	$\pm 3\%$
Додаткові вимірювання	Немає	Немає	Атмосферний тиск (300...1100 hPa)
Інтерфейс зв'язку	Однопровідний (власний)	Однопровідний (власний)	I2C, SPI
Робоча напруга	3.3V - 5.5V	3.3V - 5.5V	1.71V - 3.6V

Для забезпечення безперервного моніторингу параметрів мікроклімату в камері дозрівання сиру було проведено порівняльний аналіз поширених на ринку цифрових сенсорів: DHT11, DHT22 (аналог AM2302) та прецизійного BME280. Попри вищу абсолютну точність та інтегрований барометр у датчику BME280, для практичної реалізації проєктованої системи автоматизації було обрано сенсор DHT22.



Рисунок 2.3 - Датчик температури та вологості DHT22

Дане рішення є технічно та економічно обґрунтованим у контексті орієнтації проєкту на малі та крафтові господарства з огляду на такі чинники:

- відповідність технологічному діапазону вологості: Технологія афінажу твердих та напівтвердих сирів вимагає тривалого підтримання відносної вологості повітря на критично високому рівні - від 85% до 95%. Датчик DHT22 підтримує вимірювання вологості в повному діапазоні від 0% до 100% із допустимою похибкою $\pm 2\%$ (у цільовому діапазоні), тоді як дешевший сенсор DHT11 обмежений верхньою межею у 80%, що робить його використання у сироварінні неможливим.

- економічна доцільність та висока ремонтпридатність: Тривала експлуатація за умов вологості понад 90% неминуче призводить до поступової деградації та хімічного насичення чутливого елемента будь-якого ємнісного сенсора, що вимагає його періодичної заміни. Використання дорогих промислових датчиків суттєво підвищує капітальні витрати малого підприємства. DHT22 поєднує низьку ринкову вартість із достатньою точністю, що дозволяє оперативно й бюджетно замінити сенсор у разі виходу з ладу, не збільшуючи загальну вартість володіння системою;

- стійкість до довжини ліній зв'язку (Завадостійкість): Сенсор DHT22 використовує для обміну даними власний однопровідний цифровий протокол (Single-Bus), який задіює лише один GPIO-вивід мікроконтролера Raspberry Pi Pico. На відміну від інтерфейсу I^2C (який використовується в BME280 і є вкрай чутливим до ємності довгих дротів), протокол DHT22 дозволяє виносити датчик на відстань до 20 метрів від головного контролера без використання додаткових підсилювачів шини. Це критично важливо для розміщення сенсора безпосередньо на віддалених полицях із сиром;

- конструктивне виконання: Датчик постачається у перфорованому пластиковому корпусі, який захищає вимірювальну матрицю від випадкових механічних пошкоджень та великих крапель конденсату під час санітарної

обробки камери, що вигідно відрізняє його від безкорпусного SMD-виконання датчика BME280;

- простота програмної інтеграції: Сенсор видає вже повністю відкалібрований 16-бітний цифровий сигнал. Це виключає потребу в проведенні складних математичних обчислень чи апроксимацій на стороні Raspberry Pi Pico, заощаджуючи обчислювальні ресурси мікроконтролера для виконання алгоритмів керування контурами зворотного зв'язку.

2.1.4 Аналіз нагрівачів, осушувачів та зволожувачів

Для ефективного управління процесом дозрівання сиру автоматизована система мікроклімату повинна забезпечувати безперервний, точний і взаємопов'язаний контроль температури та відносної вологості повітря. Процес афінажу або ж дозрівання сиру становить одну з найважливіших стадій технологічного циклу, оскільки саме на цьому етапі формуються основні органолептичні характеристики, текстура й смакові властивості продукту. З позиції підприємницької діяльності, будь-яке відхилення від заданих технологічних параметрів (для більшості напівтвердих і твердих сирів оптимальна температура становить $+11...+13^{\circ}\text{C}$, а відносна вологість- $80...90\%$) може спричинити суттєві фінансові втрати. Економічні збитки можуть виникати через такі чинники: по-перше, появу дефектів у продукції, як-от небажана пліснява або бактеріальне забруднення, яке часто є наслідком надмірної вологості чи застою повітря. Ці проблеми унеможливають реалізацію партії продукту. По-друге, через недостатній рівень вологості може спостерігатися надмірне випаровування вологи з поверхні сирних голів, що своєю чергою призводить до утворення тріщин і зменшення масової частки готової продукції, що підлягає реалізації. Таким чином, це є прямими економічними втратами. У зв'язку з цим вибір технічних засобів регулювання мікроклімату, таких як обігрівачі, осушувачі та зволожувачі повітря, має базуватися на компромісі між капітальними витратами, експлуатаційними

витратами та забезпеченням технологічної надійності й безпеки виробничого процесу.

Підтримання температурного плато в камері афінажу вимагає м'якого нагріву без створення зон локального перегріву. Розглянемо основні типи виконавчих елементів:

1. Трубчасті електронагрівачі (ТЕНи) з примусовою конвекцією. Мають високу теплову потужність та низьку початкову вартість. Проте висока температура поверхні ТЕНа призводить до локального «випалювання» кисню та інтенсивного осушення повітря. У сироварінні це створює ефект «закалу» (передчасне висихання скоринки сиру, що блокує дозрівання всередині).

2. Гнучкі нагрівальні елементи (вуглецеві кабелі, силіконові мати). Забезпечують низькотемпературний, але розподілений по великій площі нагрів. Вони мають мінімальну теплову інерційність, що дозволяє інтегрувати їх у системи з PID-регулюванням для досягнення точності контролю до $\pm 0.1^{\circ}\text{C}$.

3. Інфрачервоні плівкові нагрівачі. Ефективно передають тепло безпосередньо поверхням (полицям для сиру), але створюють нерівномірний температурний градієнт у повітряному просторі камери, що ускладнює загальну автоматизацію.

Для малих та середніх крафтових сироварень оптимальним рішенням є гнучкі розподілені нагрівачі.

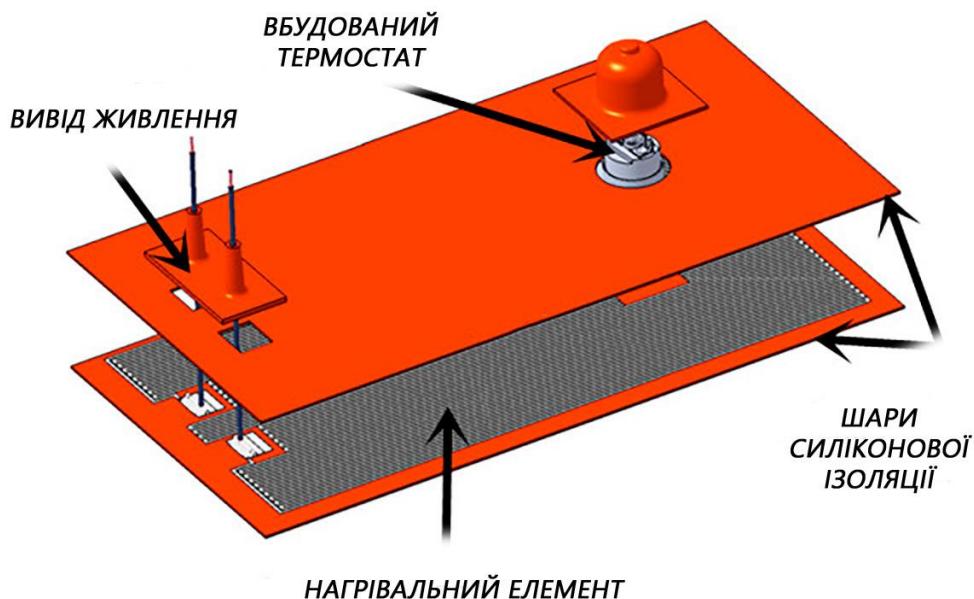


Рисунок 2.4 - Гнучкий розподілений нагрівач

Їх використання мінімізує деградацію вологості, знижує навантаження на систему зволоження і зменшує експлуатаційні витрати за рахунок тривалого терміну служби та низької аварійності.

Під час визрівання сир активно виділяє вологу у навколишнє середовище, особливо на початкових етапах процесу. Без належного осушення мікроклімат швидко набуває вологості близько 100%, що створює ризик гниття продукту та сприяє розвитку патогенних мікроорганізмів. Для комерційного регулювання вологості використовуються кілька основних технологій:

1. Конденсаційні (компресорні) осушувачі повітря. Їх принцип роботи ґрунтується на охолодженні повітря нижче точки роси, внаслідок чого волога конденсується і відводиться. Ці осушувачі є найбільш енергоефективними (кількість видаленої води на кВт-год) у промислових масштабах. Проте їх ефективна робота забезпечується лише за температури вище +15 °С. У стандартних для сироварні умовах з температурою +12 °С на випарнику може утворюватися лід, що вимагає запуску циклів автоматичного розморожування і знижує коефіцієнт корисної дії системи.

2. Термоелектричні осушувачі (елементи Пельтьє). Вони характеризуються компактністю, безшумною роботою, відсутністю рухомих частин і екологічністю (оскільки не використовують хладагенти). Пристрої легко інтегруються з мікроконтролерами завдяки можливості управління через транзисторні ключі (ШИМ). Однак їхній низький коефіцієнт корисної дії та невисока продуктивність (до 0.5-1 літра води на добу) обмежують сферу застосування невеликими демонстраційними чи лабораторними камерами об'ємом до 1 кубічного метра.

3. Адсорбційні осушувачі. Ці пристрої використовують для осушення спеціальний ротор, що вбирає вологу, і є ідеальним вибором для умов із низькими температурами (0...+10 °C). Вони забезпечують глибоку дегідратацію, але мають два недоліки: значну вартість обладнання і високе енергоспоживання під час регенерації адсорбенту.

Ретельний вибір відповідної технології осушення є важливим фактором для підтримки оптимальних умов у камерах визрівання та забезпечення якості сироварів. У табл. 2.2 матриця вибору осушувача для камери афінажу сиру.

Таблиця 2.2 - Матриця вибору осушувача для камери афінажу сиру

Тип обладнання	Технологічна точність	Рівень інвестицій	Операційні ризики
Компресорний	Середня	Середній	Обмерзання випарника
Елемент Пельтьє	Висока (для мікро)	Низький	Низька потужність
Адсорбційний	Дуже висока	Високий	Високі операційні витрати

Для мікропідприємств із камерами об'ємом до 1,5 м³ доцільно обирати спарені блоки елементів Пельтьє, оскільки вони відзначаються простотою заміни та обслуговування. Для більших промислових обсягів оптимальним варіантом є адаптовані компресорні системи, які оснащені функцією «No Frost» або низькотемпературним контуром, що забезпечує ефективність і зручність у роботі.

У разі зниження рівня вологості в камері нижче критичного порогу відбувається інтенсивна втрата ваги сиру, а його скоринка набуває надмірної товщини та жорсткості, що є технологічним дефектом. Водночас система зволоження повинна відповідати санітарно-гігієнічним нормам, зокрема стандартам НАССР (Hazard Analysis and Critical Control Points). Розглянемо основні технології зволоження, що застосовуються в умовах сироваріння:

1. Ультразвукові зволожувачі - ці пристрої функціонують завдяки п'єзоелектричним мембранам, які перетворюють воду в холодний туман із розміром частинок 1-5 мікрон. Серед їхніх ключових переваг варто відзначити високу чутливість до команд мікроконтролера, низьке споживання електроенергії та відсутність впливу на температуру всередині камери. Проте ультразвукові зволожувачі вимагають використання демінералізованої (осмотичної) води. Використання звичайної води спричиняє утворення сольового нальоту на поверхні сирних голівок та обладнання, що псує товарний вигляд готової продукції.

2. Парогенератори (ізотермічне зволоження) - ця технологія базується на нагріванні води до кипіння, що забезпечує стерильність отриманої пари, оскільки в процесі кипіння знищуються бактерії. Головним недоліком такої системи є високе енергоспоживання, а також підвищення температури всередині камери, що змушує холодильне обладнання працювати інтенсивніше, тим самим збільшуючи операційні витрати підприємства.

3. Випарні (традиційні) зволожувачі - у цьому випадку волога випаровується через обдування спеціального вологого картриджа. Перевагою цього методу є відсутність ризику утворення сольового нальоту. Проте випарні

зволожувачі характеризуються значною інерційністю та можуть стати джерелом бактеріального забруднення через накопичення мікроорганізмів і спор дикої плісняви на фільтрах. Це є неприйнятним у процесах виробництва сиру.

У крафтовому сироварінні стандартом де-факто стали ультразвукові системи зволоження. Хоча інтеграція цих модулів вимагає наявності ефективної системи зворотного осмосу для очищення води, вони забезпечують високу точність підтримання вологості у межах 85-90% при мінімальному споживанні енергії, що робить їх оптимальним вибором для забезпечення стабільної якості продукту.

2.2 Структурна схема підключення

На основі якого у підпункті 2.1 техніко-економічного аналізу та обґрунтування вибору апаратних компонентів розроблено структурну схему системи автоматичного керування (САК) кліматичною камерою для дозрівання сиру. Структурна схема визначає архітектурну побудову системи, встановлює контури регулювання технологічних параметрів, а також відображає інформаційні та силові зв'язки між функціональними блоками. Відповідно до визначених умов проектування, до складу структурної схеми входить п'ять взаємопов'язаних підсистем, апаратний склад яких узгоджено із критеріями надійності, енергоефективності та капітальних витрат.

Підсистема обробки даних та керування. Обчислювальним центром системи є мікроконтролерний модуль Raspberry Pi Pico (або його мережева модифікація Pico W), який виконує функцію центрального контролера периферійних обчислень. На цьому рівні здійснюється циклічне опитування датчикової підсистеми, первинна фільтрація сигналів від завад, порівняння поточних значень мікроклімату із заданими технологічними уставками, розрахунок керуючих впливів та координація роботи мережевих протоколів.

Підсистема первинних вимірювальних перетворювачів (датчиків). Призначена для збору інформації про стан об'єкта автоматизації та включає такі елементи:

1. Цифровий датчик температури та відносної вологості повітря (згідно з аналізом у п.п. 2.1.3, доцільно використовувати датчик високої точності, стійкий до умов підвищеної вологості до 95%). Сенсор транслює цифрові дані про стан повітряного середовища безпосередньо в обчислювальний блок.

2. Тензометричний модуль контролю маси, що складається з балочного тензодатчика (Load Cell), інтегрованого під полицю з продуктом, та 24-розрядного аналогово-цифрового перетворювача HX711. Цей блок забезпечує прецизійне вимірювання ваги сирної головки для безперервного моніторингу динаміки усушки.

Комутаційна підсистема. Для забезпечення безпечної роботи низьковольтної логіки мікроконтролера (3.3 В) та безпосереднього керування силовими виконавчими механізмами застосовано модуль розширення Waveshare Pico-Relay-B. Даний пристрій виконує функцію гальванічної розв'язки (оптичної та електромагнітної ізоляції), захищаючи виходи Raspberry Pi Pico від комутаційних сплесків напруги в колах живлення виконавчих пристроїв.

Підсистема виконавчих механізмів. Регулювання мікроклімату в обмеженому об'ємі камери здійснюється за допомогою чотирьох виконавчих елементів, підключених до відповідних каналів модуля Waveshare Pico-Relay-B:

1. Побутовий холодильник, що використовується для зниження температури та основою кліматичної камери.

2. Нагрівальний контур: гнучкий інфрачервоний плівковий нагрівач із геометричними розмірами $430 \times 303 \times 1$ мм та максимальною потужністю 30 Вт. Вибір цього елемента обумовлений його низькою термічною інерційністю та здатністю забезпечувати м'який розподілений нагрів без деградації вологості повітря.

3. Контур зволоження: ультразвуковий зволожувач повітря на базі п'єзоелектричного елемента, що генерує дрібнодисперсний холодний туман, мінімізуючи вплив на температурний режим камери.

4. Контур осушення: термоелектричний осушувач на основі елементів Пельтьє, вибраний для умов мікрокамери як найбільш компактне та просте в обслуговуванні рішення для видалення надлишкової вологи.

Підсистема бездротового зв'язку та віддаленого моніторингу. Використовуючи вбудований інтерфейс Wi-Fi, мікроконтролер інтегрується в локальну бездротову мережу. Інформаційна взаємодія з оператором (технологом) здійснюється за допомогою захищеного протоколу передачі даних через прикладний програмний інтерфейс.

Робота розробленої структури автоматизації базується на замкненому контурі зворотного зв'язку і реалізує двопозиційний закон регулювання з фіксованим гістерезисом (ΔT та ΔH). Динаміка взаємодії елементів схеми описується наступною послідовністю етапів:

1. Збір та обробка телеметрії: Мікроконтролер Raspberry Pi Pico з інтервалом τ_1 зчитує дані з датчика температури/вологості DHT22 та тензомодуля НХ711. Зафіксовані параметри поточного стану $[T_{\text{поточ}}, H_{\text{поточ}}, W_{\text{поточ}}]$ записуються в оперативну пам'ять пристрою.

2. Управління температурним плато: Якщо $T_{\text{поточ}}$ перевищує верхню допустиму межу ($T_{\text{задана}} + \Delta T$), мікроконтролер видає сигнал на Waveshare Pico-Relay-V для замикання реле компресора холодильника. У випадку падіння температури нижче межі ($T_{\text{задана}} - \Delta T$), реле компресора розмикається, і активується реле інфрачервоного нагрівального коврика потужністю 30 Вт.

3. Управління вологісним режимом: Якщо вологість повітря падає нижче норми ($H_{\text{поточ}} < H_{\text{задана}} - \Delta H$), через релейний модуль вмикається

п'єзоелектричний зволожувач. Якщо внаслідок активного виділення сироватки вологість перевищує норму ($H_{поточ} > H_{задана} + \Delta H$), реле активує термоелектричний осушувач Пельтьє.

4. Контроль та реєстрація маси: Дані про вагу продукту ($W_{поточ}$) обробляються для математичного розрахунку добового градієнта втрати маси. Це дозволяє оперативно виявити аномально швидку усушку, що свідчить про порушення процесу кіркоутворення.

5. Режим моніторингу: Зі встановленою періодичністю τ_2 сформований пакет технологічних даних надсилається через Wi-Fi модуль на MQTT сервери, де користувач отримує структурований звіт у вікні діалогу з ботом.

6. Режим оповіщення: У випадку виникнення аварійних ситуацій (вихід параметрів мікроклімату за критичні межі через відмову виконавчих механізмів або аномальна зміна ваги продукту), САК миттєво перериває стандартний цикл роботи та надсилає пріоритетне push-повідомлення користувачеві для оперативного втручання в технологічний процес.

Таким чином, представлена структура системи забезпечує високу точність стабілізації мікроклімату, мінімізує операційні ризики втрати продукції та повністю відповідає сучасній концепції Інтернету речей (IoT) в агропромисловому секторі.

2.3 Вибір платформи для зберігання даних

Ефективне функціонування системи автоматичного керування кліматичною камерою потребує не лише оперативного регулювання параметрів мікроклімату, але й організації надійного сховища для накопичення ретроспективних даних. Збір історичних даних про температуру, відносну вологість повітря та, особливо, динаміку зміни маси сиру є критично важливим для технологічного аналізу процесу афінажу. Наявність збереженої інформації дозволяє технологу оцінювати правильність перебігу усушки, будувати графіки залежностей та вдосконалювати

рецептуру. З огляду на використання мікроконтролера Raspberry Pi Pico з обмеженим об'ємом оперативної пам'яті (264 КБ SRAM) та апаратних ресурсів, вибір платформи для зберігання даних має базуватися на компромісі між архітектурною складністю, надійністю та операційними витратами.

Для реалізації інформаційного сховища в межах даного проєкту було розглянуто та проаналізовано три найбільш перспективні архітектурні підходи.

Хмарні спеціалізовані IoT-платформи часових рядів (наприклад, ThingSpeak). Даний клас платформ орієнтований на роботу з потоковими даними від датчиків, які надходять через фіксовані інтервали часу. Перевагами такого рішення є наявність готового інструментарію для графічної візуалізації, простота інтеграції з мікроконтролерами за допомогою лаконічних HTTP REST API або MQTT запитів, а також відсутність необхідності адміністрування власної серверної інфраструктури. Це повністю відповідає бізнес-критерію мінімізації капітальних витрат. Проте безкоштовні тарифні плани таких платформ часто мають обмеження на частоту оновлення даних (наприклад, не частіше ніж один раз на 15 секунд) та загальний об'єм зберігання, що, втім, є цілком прийнятним для повільних технологічних процесів дозрівання сиру.

Хмарні NoSQL бази даних реального часу (наприклад, Google Firebase Realtime Database). Цей підхід забезпечує високу гнучкість структури даних, які зберігаються у форматі JSON, та дозволяє реалізувати миттєву синхронізацію між пристроями. Firebase є масштабованим і надійним рішенням, яке легко інтегрується з мобільними та вебдодатками. Основним недоліком для даного проєкту є високе обчислювальне навантаження на процесор RP2040 мікроконтролера Raspberry Pi Pico. Робота з хмарами такого рівня вимагає складних процедур шифрування та автентифікації (SSL/TLS), а також використання об'ємних програмних бібліотек для розбору JSON, що може призвести до дефіциту оперативної пам'яті контролера та зниження стабільності роботи системи в режимі 24/7.

Реляційні бази даних на базі локального або орендованого сервера (наприклад, MySQL / PostgreSQL). Організація власної бази даних через проміжний вебсервер (скрипти на PHP/Python) надає розробнику абсолютний контроль над архітектурою сховища, безпекою та об'ємом інформації без будь-яких зовнішніх лімітів. Дані передаються від мікроконтролера простими POST-запитами, що мінімізує навантаження на периферійний пристрій. Головним стримуючим фактором для крафтових підприємств є зростання операційних витрат, адже розгортання та підтримка власного сервера вимагає постійної оплати хостингу, резервного копіювання та залучення ІТ-спеціалістів для обслуговування.

Обґрунтування прийнятого рішення

З урахуванням критеріїв, сформульованих у підпункті 2.1 (низькі капітальні витрати, висока енергоефективність, простота обслуговування без залучення вузькопрофільних фахівців), для розроблюваної системи автоматизації прийнято рішення використовувати гібридну схему зберігання даних.

Основним хмарним сховищем обрано спеціалізовану IoT-платформу часових рядів (ThingSpeak). Вона дозволяє без додаткових витрат акумулювати дані мікроклімату й ваги, автоматично будувати тренди усушки продукту та надає зручний доступ до графіків через веб-інтерфейс з будь-якої точки світу. Передача даних здійснюється за допомогою легковагого протоколу HTTP, який не перевантажує обчислювальне ядро Raspberry Pi Pico і легко реалізується мовою MicroPython.

Для забезпечення живучості та автономності системи (реалізація принципу Edge Computing) хмарне рішення доповнюється двома локальними контурами:

1. Короткострокове локальне буферизаційне сховище: у випадку тимчасової відсутності Інтернет-з'єднання мікроконтролер переходить у режим накопичення даних у внутрішню флеш-пам'ять (пам'ять логів). Після відновлення зв'язку накопичений буфер автоматично вивантажується в хмару, що запобігає втраті цінної технологічної інформації.

2. Оперативне інформування та ефемерне зберігання: паралельно з відправкою даних на IoT-платформу, система транслює поточні показники та аварійні сповіщення у створений MQTT сервер. Він виступає додатковим швидким сховищем, зручним для оперативного перегляду останніх подій безпосередньо зі смартфона користувача.

3 ПРОЄКТНА ЧАСТИНА

3.1 Розробка алгоритму роботи камери дозрівання сирів

Для забезпечення оптимальних умов технологічного процесу дозрівання сирної продукції необхідно підтримувати чітко задані параметри температури та відносної вологості повітря. З цією метою було розроблено алгоритм роботи керуючого пристрою, що базується на циклічному опитуванні сенсорів, порівнянні отриманих даних із уставками та формуванні керуючих сигналів на виконавчі механізми (охолоджувач, нагрівач, зволожувач, вентиляція).

Особливістю алгоритму є його дворівнева структура: локальне керування (виконання критично важливих завдань підтримання мікроклімату) та мережева взаємодія (передача телеметрії та отримання команд через протокол MQTT).

3.2 Розробка електричної схеми камери дозрівання

Для організації ефективного керування технологічним процесом дозрівання сирів, чіткого розмежування функціональних обов'язків апаратних модулів та визначення напрямків потоків інформації було розроблено структурну схему системи. Структурна схема є основою для подальшого проєктування електричної принципової схеми та розробки програмного забезпечення мікроконтролера.

Головним завданням розробленої структури є забезпечення безперервного моніторингу параметрів мікроклімату (температури та відносної вологості повітря) всередині камери, локального автоматичного регулювання цих параметрів та двостороннього обміну даними із зовнішніми сервісами за допомогою бездротового каналу зв'язку.

Розроблена структурна схема системи автоматизованого керування камерою дозрівання сирів представлена на рисунку 3.1.

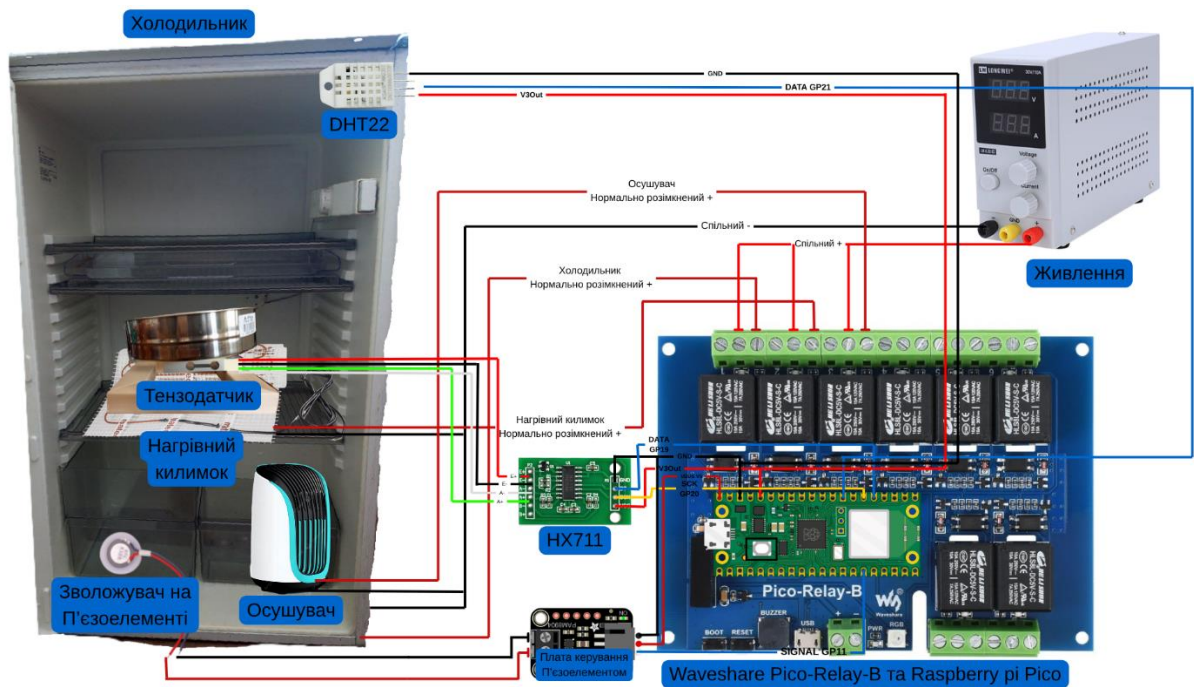


Рисунок 3.1 - Структурна схема системи автоматизованого керування камерою дозрівання сирів

Згідно з розробленою архітектурою (рис. 3.1), систему автоматизації можна розділити на кілька основних взаємопов'язаних підсистем та блоків:

1. Центральний керуючий модуль (Мікроконтролер). Ядром системи виступає мікроконтролер Raspberry Pi Pico W, побудований на базі високоефективного двоядерного процесора RP2040. Вибір цього контролера зумовлений наявністю інтегрованого бездротового модуля CYW43439, що дозволяє реалізувати концепцію Edge Computing (периферійних обчислень). Контролер виконує локальний алгоритм опитування датчиків, здійснює математичну обробку сигналів, керує виконавчими механізмами та підтримує стек мережових протоколів для зв'язку з хмарною інфраструктурою.

2. Сенсорна підсистема (Блок датчиків) Призначена для отримання первинної інформації про стан середовища в камері дозрівання. До її складу входять цифрові датчики температури та відносної вологості повітря. Датчики здійснюють вимірювання фізичних величин, перетворюють їх у цифровий код та передають на

мікроконтролер через послідовні інтерфейси зв'язку. Це мінімізує вплив аналогових завад на точність вимірювання.

3. Блок комутації та виконавчих механізмів Забезпечує фізичний вплив на об'єкт керування для підтримання заданого технологічного регламенту. Оскільки виконавчі пристрої працюють від мережі змінного струму напругою 220 В, а мікроконтролер оперує логічними рівнями 3.3 В, до складу блока входить модуль реле з оптичною гальванічною розв'язкою. Мікроконтролер через силові ключі керує такими виконавчими елементами:

- охолоджувач (компресор або вентилятор охолодження) - для зниження температури;
- нагрівач (ТЕН) - для підвищення температури у разі переохолодження;
- зволожувач повітря - для підтримки високого рівня вологості (80-95%), необхідного для правильного формування сирної скоринки;
- система вентиляції - для забезпечення циркуляції повітря та видалення надлишків вуглекислого газу й аміаку, що виділяються під час дозрівання.

4. Блок живлення Призначений для перетворення нестабільної напруги промислової мережі 220 В у стабілізовану постійну напругу 5 В, яка необхідна для функціонування мікроконтролера, плати реле та сенсорних модулів.

5. Мережева інфраструктура та Хмарні сервіси (Зовнішній контур) Зв'язок локальної системи із зовнішнім світом відбувається через бездротову точку доступу Wi-Fi. Інформаційні потоки розподіляються за трьома основними напрямками:

- NTP-сервер (pool.ntp.org): використовується контролером під час ініціалізації для отримання точного часу UTC, що необхідно для часового маркування технологічних логів;
- MQTT-брокер: виступає в ролі центрального сервера автоматизації, через який пристрій публікує телеметрію та приймає командні пакети (вибір режимів, зміна уставок).

Особливістю розробленої структури є її стійкість до відмов: у разі втрати зв'язку із Wi-Fi маршрутизатором або MQTT-брокером, центральний контролер переходить у режим автономного функціонування (локальний контур регулювання продовжує працювати), паралельно активуючи вбудовану точку доступу (Config Server) для локального бездротового налаштування параметрів мережі оператором.

3.3 Розробка програмного забезпечення мікроконтролера

Програмне забезпечення керуючого контролера камери дозрівання сирів реалізовано мовою MicroPython для платформи Raspberry Pi Pico W. Архітектура ПЗ побудована за модульним принципом: кожен модуль відповідає за окрему функціональну підсистему — мережеве підключення, обмін повідомленнями через MQTT, управління режимами, виконання задач та зважування продукту.

`main.py` є головним скриптом ініціалізації та зв'язку керуючого контролера камери, лістинг якого наведено в Додатку А. Під час запуску він виконує ланцюжок ініціалізації: зупиняє залишкові задачі від попереднього сеансу, зчитує або генерує унікальний ключ пристрою, підключається до мережі Wi-Fi, синхронізує системний час через NTP та встановлює з'єднання з MQTT-брокером. Після успішного підключення контролер надсилає подію `waiting_for_mode` і переходить у безперервний цикл очікування команд. У разі відсутності збережених мережевих налаштувань запускається вбудований HTTP-сервер конфігурації.

Модуль `mqtt_client.py` реалізує всю логіку взаємодії з MQTT-брокером HiveMQ Cloud через захищене з'єднання TLS/SSL на порту 8883, лістинг якого наведено в Додатку Г. Модуль забезпечує підключення до персонального топіка пристрою `devices/{device_key}/modes`, обробку вхідних повідомлень із новими режимами роботи, публікацію подій та дій у відповідні топіки, а також перевірку зважування між послідовними діями — якщо сир не було знято з ваги між двома

виконаннями дії, система генерує попередження. Для підвищення надійності всі операції обгорнуто в блоки обробки виключень.

Модуль `wifi.py` відповідає за підключення до Wi-Fi та надання інтерфейсу первинного налаштування, лістинг якого наведено в Додатку Е. При наявності збережених облікових даних у файлі `wifi.json` здійснюється спроба підключення протягом 10 секунд. За відсутності або невдалому підключенні мікроконтролер активує режим точки доступу з назвою `PicoSetup` і запускає мінімальний HTTP-сервер на порту 80. Веб-інтерфейс сервера відображає список доступних мереж, отриманий скануванням ефіру, поле введення пароля та унікальний ключ пристрою для прив'язки до хмарного сервісу.

Модуль `device.py` реалізує генерацію та збереження унікального ідентифікатора пристрою у файлі `device.json`. Ключ формується як рядок із чотирьох символів (великі літери та цифри) на основі апаратного генератора випадкових чисел `urandom`. Функція `get_device_key()` повертає збережений ключ або автоматично створює новий, якщо файл відсутній або пошкоджений. Цей ключ використовується у всіх MQTT-топіках для маршрутизації повідомлень між конкретним пристроєм і хмарним сервісом. Лістинг модуля наведено в рамках Додатку Б.

Модуль `modes.py` забезпечує збереження та вибір поточного режиму роботи камери. Режим — це JSON-об'єкт із полями `name` та `tasks`, де `tasks` є масивом задач із параметрами `action`, `interval` та `count`. Функція `choose_mode()` повертає перший режим із файлу `modes.json`, що дозволяє зберігати режим між перезавантаженнями. Лістинг модуля наведено в рамках Додатку В.

Модуль `tasks.py` реалізує планувальник задач на основі апаратних таймерів `machine.Timer`, лістинг якого наведено в рамках Додатку Г. Кожна задача зі списку `tasks` поточного режиму запускається як окремий переривально-керований таймер із заданим інтервалом у секундах. Підтримується обмежена кількість виконань через параметр `count`: після досягнення ліміту таймер автоматично деактивується.

Функція `stop_tasks()` зупиняє всі активні таймери та очищає список, що гарантує коректне перемикання між режимами.

Модуль `weight_monitor.py` реалізує моніторинг ваги за допомогою тензодатчика через АЦП `NX711`, лістинг якого наведено в Додатку Д. Ініціалізація сенсора виконується з налаштуванням підсилення 128 та очікуванням стабілізації. Нульова точка зберігається у файлі `zero_point.txt` і завантажується при повторному запуску, що дозволяє уникнути необхідності калібрування після кожного перезавантаження. Функція `read_weight()` усереднює три послідовних виміри для зменшення шуму. Моніторинг здійснюється таймером із періодом 2 секунди: якщо показання опускаються нижче 0.1 кг, фіксується факт знімання сиру.

3.4 Тестування та аналіз даних

Тестування програмного забезпечення мікроконтролера проводилося на рівні модульних тестів окремих функцій, що дозволило виявити та усунути потенційні помилки ще до розгортання на реальному обладнанні. Оскільки `MicroPython` не підтримує стандартну бібліотеку `unittest` у повному обсязі, для тестування використовувався підхід на основі імітаційних об'єктів (`mock-об'єктів`) та простих функцій-перевірок, що виконуються на стандартному `Python 3` у середовищі розробника.

Перевірялися ключові функції модулів: генерація унікального ключа пристрою, вибір режиму роботи, обрахунок ваги продукту та логіка виявлення факту знімання сиру. Нижче наведено приклад набору тестів для функцій модуля `device.py` та `weight_monitor.py`.

Тест генерації ключа пристрою перевіряє, що функція `generate_device_key()` завжди повертає рядок довжиною 4 символи, що складається виключно з великих літер латинського алфавіту та цифр. Лістинг файлу `test_device.py` якого наведено в Додатку Є.

Тест логіки визначення ваги перевіряє коректність обчислення маси продукту за відомими значеннями АЦП та нульовою точкою, а також граничну умову

виявлення знімання сиру (поріг 0.1 кг) Лістинг файлу `test_weight_monitor.py` якого наведено в Додатку Ж. Всі наведені тести виконуються незалежно від апаратного забезпечення та дозволяють автоматично верифікувати коректність бізнес-логіки при внесенні змін до коду. Результати тестування підтвердили коректну роботу ключових алгоритмів: генерація ключів завжди повертає валідні чотирисимвольні ідентифікатори, розрахунок ваги відповідає очікуваним значенням з точністю до 0.01 кг, а логіка виявлення знімання сиру правильно реагує на перехід показань нижче граничного значення 0.1 кг.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Актуальність безпеки життєдіяльності людини

Запровадження автоматизованих систем моніторингу та управління (АСМіУ) у харчовій промисловості, зокрема на етапах дозрівання сирів, орієнтоване не лише на вдосконалення технологічних процесів, але й на зменшення впливу людського фактора на виробничі ризики. Однак використання сучасного комп'ютеризованого обладнання і кліматичних камер створює комплекс специфічних небезпечних та шкідливих виробничих факторів (НШВФ), які впливають на персонал: інженерів-програмістів, операторів АСМіУ та технологів.

Актуальність забезпечення безпеки життєдіяльності та охорони праці в цьому контексті обумовлена такими ризиками:

- електротравматизм: елементи автоматики, датчики, контролери, виконавчі механізми (зокрема компресори, вентилятори, ТЕНи) отримують живлення від промислової електромережі, що створює потенційну можливість уражень електрострумом;
- специфічні умови мікроклімату. Обслуговування кліматичних камер вимагає регулярного контакту з ділянками підвищеної вологості (85-95%) та низької температури (10-15 °С);
- психофізіологічне навантаження. Робота оператора диспетчерського пункту передбачає тривале зорове навантаження, обмежену рухову активність (гіподинамію) і значний нервово-емоційний стрес під час прийняття рішень у нестандартних ситуаціях.

Впровадження превентивних заходів у сфері охорони праці дозволяє гарантувати безперебійну роботу системи «людина - машина - середовище», зводячи до мінімуму ризики виробничого травматизму та професійних захворювань.

4.2. Санітарно-гігієнічні вимоги до умов праці

Організація робочого місця оператора АСМіУ та обслуговування кліматичної камери мають відповідати чинним нормативно-правовим актам України (зокрема, ДСанПіН 3.3.2.007-98 та ДСТУ EN ISO 9241).

Мікроклімат робочої зони

Для приміщення, де розташоване автоматизоване робоче місце (АРМ) оператора, встановлюються оптимальні мікрокліматичні параметри (відповідно до категорій робіт за важкістю - легка Іа):

- Температура повітря: 22-24°C (у теплий період), 21-23°C (у холодний період).
- Відносна вологість повітря: 40-60%.
- Швидкість руху повітря не більше 0,1 м/с.

Важливо: Безпосередньо всередині кліматичної камери для дозрівання сирів вологість сягає 90%. Тому електронне обладнання (датчики вологості/температури, блоки комутації) повинно мати клас захисту не нижче IP65/IP67 для запобігання короткому замиканню та корозії, а персонал під час завантаження/вивантаження продукції має використовувати відповідний спецодяг.

Освітлення

Освітленість робочого місця оператора ПК є критичним фактором. Робочий стіл розміщують так, щоб природне світло падало збоку (переважно ліворуч).

- штучне освітлення виконується за комбінованою системою (загальне + місцеве);
- нормована освітленість на поверхні стола в зоні розміщення документів становить не менше 300-500 лк;
- для обмеження блікості екранів застосовуються антивідблискові покриття та сонцезахисні жалюзі на вікнах.

Ергономіка та неіонізуюче випромінювання

Сучасні РК-монітори персональних комп'ютерів відповідають суворим стандартам (наприклад, TCO), що мінімізує рівень електромагнітного та рентгенівського випромінювання. Проте, для зменшення статичного напруження опорно-рухового апарату конструкція робочого крісла повинна забезпечувати підтримку попереку, регулювання висоти та кута нахилу спинки. Відстань від очей оператора до екрана монітора має становити 600-700 мм.

4.3. Пожежна профілактика на робочому місці

Проектowana система управління містить значну кількість електрообладнання (шафи керування, силові кабелі, сервери, блоки живлення), що класифікує приміщення АРМ та зони розміщення автоматики як пожежонебезпечні. Основними причинами пожеж у таких локаціях є електричні перевантаження, великі перехідні опори, короткі замикання або іскріння.

Пожежна безпека забезпечується комплексом організаційних та інженерно-технічних заходів згідно з вимогами Кодексу цивільного захисту України та НАПБ Б.03.002-2007.

Технічні заходи пожежної профілактики:

- захисна автоматика: Електричні схеми живлення АСМіУ обов'язково оснащуються автоматичними вимикачами, пристроями захисного вимкнення (ПЗВ) та плавкими запобіжниками для миттєвого знеструмлення мережі у разі виникнення аварійних режимів роботи;
- матеріали: Ізоляція кабельних ліній та внутрішній монтаж шаф керування виконуються з матеріалів, що не підтримують горіння (наприклад, з низьким димовиділенням типу НГ-LS);
- системи сигналізації: Приміщення з комп'ютерною технікою та безпосередньо виробничі цехи обладнуються автоматичною пожежною сигналізацією з оптико-електронними димовими сповіщувачами.

Первинні засоби пожежогасіння

Гасіння пожеж в електроустановках під напругою (клас пожежі Е) водою або піною суворо заборонено через високу електропровідність гасильних речовин.

Таблиця 4.1 -Вибір типу вогнегасників залежно від призначення та локації

Локація	Тип вогнегасника	Призначення
Приміщення АРМ оператора	Вуглекислотні (ВВ-2, ВВ-5)	Ефективно охолоджують зону горіння, не пошкоджують електронні плати та не залишають бруду.
Виробнича зона (шафи керування)	Порошкові (ВП-5, ВП-9)	Використовуються для локалізації займань силового кабельного господарства.

Організаційні заходи

- Розроблення та розміщення на видних місцях планів евакуації персоналу.
- Проведення регулярних інструктажів з пожежної безпеки (вступний, первинний, повторний).
- Призначення осіб, відповідальних за пожежний стан приміщень.
- Облаштування спеціальних місць для куріння за межами виробничих та адміністративних корпусів.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне науково-практичне завдання — розроблено прототип системи автоматизованого моніторингу та керування кліматичною камерою дозрівання сирів для малих крафтових підприємств на базі IoT-технологій та концепції периферійних обчислень (Edge Computing). За результатами виконання роботи сформульовано такі основні висновки:

- встановлено, що процес дозрівання (афінажу) твердого сиру є найбільш критичним і фінансово ризикованим етапом виробничого циклу. Жорстке підтримання температури в межах $+11...+13\text{ }^{\circ}\text{C}$ та відносної вологості повітря 80-95 % є визначальними факторами формування якості готового продукту. Доведено, що для малих крафтових підприємств девіація цих параметрів або їх нерівномірний розподіл на полицях камери спричиняє критичні фінансові збитки. Обґрунтовано, що головним технологічним індикатором правильного перебігу дозрівання є добовий градієнт втрати маси продукту — швидкість усушки;

- проведено критичний аналіз існуючих засобів автоматизації трьох класів: промислових ПЛК (Carel, Siemens, ОВЕН), комерційних IoT-екосистем (TuYa, Sonoff) та DIY-розробок на мікроконтролерах. Доведено, що жоден із наявних класів рішень не задовольняє повному переліку вимог малого крафтового виробництва: промислові системи є фінансово недоступними та мають закриту архітектуру, а побутові IoT-модулі характеризуються низькою деградаційною стійкістю в умовах екстремальної вологості та повною залежністю від зовнішніх хмарних серверів;

- обґрунтовано доцільність та реалізовано концепцію Edge Computing на базі мікроконтролера Raspberry Pi Pico W. Такий підхід дозволяє виконувати критичні алгоритми регулювання безпосередньо на об'єкті, незалежно від стабільності інтернет-з'єднання, забезпечити локальну буферизацію даних у флеш-пам'ять та інтегрувати систему з мобільними сервісами оператора без проміжних серверів;

- на підставі порівняльного аналізу бездротових технологій передачі даних (Wi-Fi, Bluetooth, LoRaWAN, Zigbee) визначено оптимальний інтерфейс для даного

застосування — Wi-Fi. Він забезпечує достатню пропускну здатність для трансляції телеметрії, прямий зв'язок із мобільним застосунком оператора та необхідне покриття в межах залізобетонної камери сироварні;

- розроблено та спроектовано апаратну частину системи: структурну схему із п'ятьма основними підсистемами (центральный керуючий модуль, сенсорна підсистема, блок комутації та виконавчих механізмів, блок живлення, мережева інфраструктура), а також електричну принципову схему на основі мікроконтролера Raspberry Pi Pico W, датчиків DHT22, тензодатчика з АЦП HX711 та релейного модуля Waveshare Pico-Relay-B із гальванічною розв'язкою;

- реалізовано модульне програмне забезпечення мовою MicroPython із дворівневою архітектурою (локальний та мережевий контури), що включає модулі керування Wi-Fi, MQTT-клієнт із TLS/SSL-захистом, планувальник задач на апаратних таймерах, систему тензометричного моніторингу ваги та механізм автономної роботи при втраті зв'язку. Усі модулі верифіковано модульними тестами; точність обчислення маси продукту складає не гірше 0,01 кг;

- розроблено розділ із безпеки життєдіяльності та охорони праці: проаналізовано специфічні небезпечні та шкідливі виробничі фактори (електротравматизм, підвищена вологість, психофізіологічне навантаження), визначено санітарно-гігієнічні вимоги до умов праці оператора (мікроклімат, освітленість, ергономіка) та сформульовано комплекс організаційно-технічних заходів пожежної профілактики;

- практична цінність результатів роботи полягає в тому, що розроблений прототип системи дозволяє малим крафтовим сироварням без суттєвих капітальних витрат реалізувати безперервний моніторинг мікроклімату 24/7, відстежувати динаміку усушки сирної головки в реальному часі, отримувати миттєві сповіщення про відхилення параметрів та мінімізувати вплив людського фактора, що в сукупності гарантує стабільну якість продукції та захист від критичних фінансових збитків.

ПЕРЕЛІК ПОСИЛАНЬ

1. Бездротові технології передачі даних WI-FI, BLUETOOTH та ZIGBEE. URL: <https://ela.kpi.ua/server/api/core/bitstreams/ab30c487-9147-47aa-8561-47b2064c3ce4/content>.
2. Виробники сиру в Україні: глибокий огляд галузі у 2025 році. URL: <https://intime.ua/vyrobnyky-syru-v-ukrayini-glybokyj-oglyad-galuzi-u-2025-roczy>
3. Santos J. M., Garcia J. M., Dias J. Energy Sustainability in the Ripening of Traditional Cheese: Renewable Energy Sources and Internet of Things Based Energy Monitoring. *Dairy*. 2025. Vol. 6, Iss. 6. URL: <https://www.mdpi.com/2624-862X/6/6/63>
4. New ripening room monitoring technology for improving the efficiency and sustainability of cheese ripening processes (SMART-RIPE) : Project Fact Sheet FP7. *CORDIS: European Commission*. Project ID 613827. URL: <https://cordis.europa.eu/project/id/613827>
5. IoT Sensors in Dairy Production: Real-Time Monitoring of pH, Temperature, and Humidity for Optimal Cheese and Yogurt Quality. *DairyCraftPro*. 2025. URL: <https://dairycraftpro.com/iot-sensors-in-dairy-production-real-time-monitoring-of-ph-temperature-and-humidity-for-optimal-cheese-and-yogurt-quality/>
6. Розгляд процесу створення сиру. URL: <https://intime.ua/yak-roblyat-syr-detalnyj-rozglyad-proczesu-vid-moloka-do-stolu/>
7. Tailor G. et al. IoT-Based Food Quality Monitoring System. *IET Wireless Sensor Systems*. 2025. URL: <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/wss2.70008>
8. Applying Computational Fluid Dynamics in the Development of Smart Ripening Rooms for Traditional Cheeses. *Foods*. 2021. Vol. 10, Iss. 8. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8394216/>
9. Use of Internet of Things for Monitoring, Automation, and Performance in Industrial Production of Cheese. *Research, Society and Development*. 2025. Vol. 14, No. 6. DOI: 10.33448/rsd-v14i6.49054. URL: <https://rsdjournal.org/rsd/article/view/49054>

10. Doropoulou D., Angelopoulos V., Amanatidis P., Lagkas T., Karampatzakis D. Teaching Embedded Systems and IoT at the University using MicroPython and Raspberry Pi Pico. 27th Pan-Hellenic Conference on Progress in Computing and Informatics (PCI 2023). ACM, 2023. DOI: 10.1145/3635059.3635079. URL: <https://dl.acm.org/doi/10.1145/3635059.3635079>
11. Loker D. Embedded Systems using the Raspberry Pi Pico. 2022 ASEE Annual Conference & Exposition. 2022. URL: <https://peer.asee.org/embedded-systems-using-the-raspberry-pi-pico.pdf>
12. Building an open-source, open-hardware platform for IoT education using Raspberry Pi Pico W. Proceedings of the 14th International Conference on the Internet of Things. ACM, 2024. DOI: 10.1145/3703790.3703842. URL: <https://dl.acm.org/doi/full/10.1145/3703790.3703842>
13. Raspberry Pi Pico W: Getting Started with MQTT (MicroPython). Random Nerd Tutorials. 2024. URL: <https://randomnerdtutorials.com/raspberry-pi-pico-w-mqtt-micropython/>
14. Vakaliuk T. A., Andreiev O. V., Dubyna O. F. et al. Use of wireless technologies in IoT projects. Journal of Edge Computing. 2024. Vol. 3, No. 2. URL: <https://acnsci.org/journal/index.php/jec/article/view/747>
15. DHT22 Temperature-Humidity Sensor. Waveshare Electronics. URL: https://www.waveshare.com/wiki/DHT22_Temperature-Humidity_Sensor
16. Overview: DHT11, DHT22 and AM2302 Sensors. Adafruit Learning System. URL: <https://learn.adafruit.com/dht/overview>
17. HX711 24-Bit Analog-to-Digital Converter for Weigh Scales. Avia Semiconductor Co., Ltd. Datasheet. URL: <https://kobastar.com/en/hx711/>

ДОДАТКИ

Додаток А. Лістинг файлу “main.py”

```
import ntptime, time
from source import wifi, modes, tasks, mqtt_client, device
from source.tasks import stop_tasks

stop_tasks()

DEVICE_KEY = device.get_device_key() # 📁 автоматично завантажить або
створить ключ

def sync_time():
    print("🕒 Syncing NTP...")
    try:
        ntptime.host = "pool.ntp.org"
        ntptime.settime()
        print("✅ Time synced")
    except Exception as e:
        print("⚠️ NTP failed:", e)
    tm = time.localtime()
    print("📅 Current UTC:", "{:04d}-{:02d}-{:02d}"
{:02d}:{:02d}:{:02d}".format(*tm[0:6]))

def main():
    print(f"🔑 Device key: {DEVICE_KEY}") # Для логів

    if wifi.connect_wifi():
        sync_time()

    # 🌀 Підключення до MQTT
    if not mqtt_client.connect():
        print("⚠️ Continue without MQTT")
```

```

else:
    # 📩 Надсилаємо подію що Pico чекає режиму
    mqtt_client.publish_event("waiting_for_mode", {"device_key":
DEVICE_KEY})
    print("⌚ Очікую режим з MQTT...")

    # 🔄 Основний цикл - слухаємо MQTT
    while True:
        mqtt_client.check_messages()
        time.sleep(1)
    else:
        print("⚠ No WiFi, starting config server")
        wifi.start_config_server()

if __name__ == "__main__":
    main()
#

```

Додаток Б. Лістинг файлу “device.py”

```
import ujson, urandom

DEVICE_FILE = "device.json"

def load_device_key():
    try:
        with open(DEVICE_FILE, "r") as f:
            return ujson.load(f)["key"]
    except:
        return None

def save_device_key(key):
    with open(DEVICE_FILE, "w") as f:
        ujson.dump({"key": key}, f)

def generate_device_key():
    chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
    key = "".join(chars[urandom.getrandbits(8) % len(chars)] for _ in
range(4))
    save_device_key(key)
    return key

def get_device_key():
    return load_device_key() or generate_device_key()
```

Додаток В. Лістинг файлу “modes.py”

```
import ujson

MODES_FILE = "modes.json"

def load_modes():
    try:
        with open(MODES_FILE, "r") as f:
            return ujson.load(f)
    except:
        print("⚠ No modes found")
        return []

def choose_mode():
    """Повертає перший режим з файлу (якщо є)"""
    modes = load_modes()
    if not modes:
        return None
    return modes[0]
```

Додаток Г. Лістинг файлу “mqtt_client.py”

```
import ssl
import json
import ujson
from umqtt.simple import MQTTClient
from . import modes, tasks
from . device import get_device_key
from . import weight
from . import weight_monitor

# 🔑 MQTT параметри
MQTT_BROKER = "766f8c18abdb42f884289d068330e583.s1.eu.hivemq.cloud"
MQTT_PORT = 8883
MQTT_CLIENT_ID = "PICO"
MQTT_USER = "Cheez"
MQTT_PASSWORD = "RaspberryTeam2025!"

DEVICE_FILE = "device.json"

# SSL контекст
context = ssl.SSLContext(ssl.PROTOCOL_TLS_CLIENT)
context.verify_mode = ssl.CERT_NONE

mqtt_client = None
device_key = None # кешуємо, щоб не читати з файла кожен раз

def on_message(topic, msg):
    """Обробка вхідних повідомлень"""
    print(f"📧 MQTT message on {topic}: {msg}")
```

```

try:
    mode_data = ujson.loads(msg)

    if "name" not in mode_data or "tasks" not in mode_data:
        print("⚠ Invalid mode format!")
        return

    # Зберігаємо у modes.json
    with open(modes.MODES_FILE, "w") as f:
        ujson.dump([mode_data], f)

    print("✅ New mode saved:", mode_data["name"])

    # Зупиняємо старі задачі
    tasks.stop_tasks()

    # Запускаємо новий режим
    new_mode = modes.choose_mode()
    if new_mode:
        tasks.start_tasks(new_mode["tasks"], publish_action)
        print("🚀 New mode started:", new_mode["name"])

        # Надсилаємо підтвердження у MQTT
        publish_event("mode_started", {"name": new_mode["name"]})

except Exception as e:
    print("❌ Error handling message:", e)

def connect():
    global mqtt_client, device_key

```

try:

```
#         device_key = load_device_key()
device_key = get_device_key()
if not device_key:
    print("✗ No device key, cannot connect MQTT")
    return False

mqtt_client = MQTTClient(
    client_id=MQTT_CLIENT_ID,
    server=MQTT_BROKER,
    port=MQTT_PORT,
    user=MQTT_USER,
    password=MQTT_PASSWORD,
    ssl=context,
)
mqtt_client.set_callback(on_message)

mqtt_client.connect()

# 📁 Персональний топик для режимів
sub_topic = f"devices/{device_key}/modes"
mqtt_client.subscribe(sub_topic.encode())
print(f"✅ Connected & subscribed to {sub_topic}")
weight.init_hx()

return True
except Exception as e:
    print("✗ MQTT connection error:", e)
    return False
```

```

def disconnect():
    global mqtt_client
    try:
        if mqtt_client:
            mqtt_client.disconnect()
            print("🔌 MQTT disconnected")
    except Exception as e:
        print("⚠️ Error on disconnect:", e)


def publish_action(action):
    global mqtt_client, device_key
    if not mqtt_client or not device_key:
        return
    try:
        topic = f"devices/{device_key}/actions"
        msg = json.dumps({"action": action, "device_key": device_key})
        mqtt_client.publish(topic.encode(), msg)
        print(f"📡 Sent to MQTT [{topic}]: {msg}")

    # 🔄 Ініціалізуємо вагу, якщо ще не ініціалізовано
    if weight_monitor.hx is None:
        weight_monitor.init_hx711()
        weight_monitor.get_zero_point()

    # Якщо попередня вага не падала – сир не знімали
    if not weight_monitor.weight_dropped:
        # mqtt_client.publish_event("warning", {"message": "Cheese was
        not flipped last time!"})

```

```
        mqtt_publish_event("warning", {"message": "Cheese was not  
flipped last time!"})
```

```
    # Після відправки дії – скидаємо стан і запускаємо новий моніторинг  
    weight_monitor.start_monitor()
```

```
except Exception as e:  
    print("✗ Publish error:", e)
```

```
def mqtt_publish_event(event_name, data):  
    """Відправка подій у MQTT, як обгортка для publish_event"""  
    global mqtt_client, device_key  
    if not mqtt_client or not device_key:  
        return  
    try:  
        topic = f"devices/{device_key}/events"  
        payload = {"device_key": device_key, "event": event_name}  
        if data:  
            payload.update(data)  
        msg = json.dumps(payload)  
        mqtt_client.publish(topic.encode(), msg)  
        print(f"📡 Event sent [{topic}]: {msg}")  
    except Exception as e:  
        print(f"✗ Event publish error: {e}")
```

```
def publish_event(event, data=None):  
    global mqtt_client, device_key
```

```

if not mqtt_client or not device_key:
    return
payload = {"event": event, "device_key": device_key}
if data:
    payload.update(data)

try:
    topic = f"devices/{device_key}/events"
    mqtt_client.publish(topic.encode(), json.dumps(payload))
    print(f"📡 Event sent [{topic}]: {payload}")
except Exception as e:
    print("❌ Event publish error:", e)

```

```

def check_messages():
    global mqtt_client
    try:
        if mqtt_client:
            mqtt_client.check_msg()
    except Exception as e:
        print("⚠️ Error checking messages:", e)

```

```
#
```

Додаток Г. Лістинг файлу “tasks.py”

```
import machine

timers = []

def start_tasks(tasks_config, action_callback):
    """Запускає задачі по режиму"""
    global timers
    stop_tasks() # зупиняємо попередні
    timers = []

    for task in tasks_config:
        interval = task.get("interval", 60)
        action = task.get("action", "unknown")
        count = task.get("count", 0) # скільки разів виконати (0 =
        безкінечно)

        executed = {"n": 0} # локальний лічильник

        def handler(timer, a=action, c=count, state=executed, t=None):
            # Виконати дію
            action_callback(a)
            state["n"] += 1

            # Якщо count > 0 і вже виконали потрібну кількість – зупиняємо
            таймер

            if c > 0 and state["n"] >= c:
                try:
                    timer.deinit()
                    print(f"■ Task '{a}' finished after {state['n']}
times")
                except:
```

```
pass
```

```
t = machine.Timer(-1)
t.init(
    period=interval * 1000,
    mode=machine.Timer.PERIODIC,
    callback=lambda x, h=handler: h(x, t=None)
)
timers.append(t)
```

```
print(f"✅ Started {len(timers)} tasks")
```

```
def stop_tasks():
    """Зупиняє всі задачі"""
    global timers
    for t in timers:
        try:
            t.deinit()
        except:
            pass
    timers = []
    print("🛑 All tasks stopped")
```

Додаток Д. Лістинг файлу “weight_monitor.py”

```
import time
from machine import Pin, Timer
from hx711 import hx711

CALIBRATION_FACTOR = 435000
HX_DATA_PIN = 13
HX_CLK_PIN = 12

hx = None
zero_point = 0.0
check_timer = None

# Стан між діями
weight_dropped = True    # чи була вага ~0 між діями
last_weight = 0.0

def init_hx711():
    global hx
    try:
        hx = hx711(Pin(HX_CLK_PIN), Pin(HX_DATA_PIN))
        hx.set_power(hx711.power.pwr_up)
        hx.set_gain(hx711.gain.gain_128)
        hx711.wait_settle(hx711.rate.rate_10)
        print("✅ HX711 initialized")
    except Exception as e:
        print(f"❌ HX711 init error: {e}")
        hx = None

def save_zero_point(value):
    try:
```

```
    with open("zero_point.txt", "w") as f:
        f.write(str(value))
    print("💾 Zero point saved to file")
except Exception as e:
    print("⚠ Save zero error:", e)
```

```
def load_zero_point():
    try:
        with open("zero_point.txt", "r") as f:
            value = float(f.read().strip())
            print("📂 Loaded zero point from file")
            return value
    except Exception:
        return None
```

```
def get_zero_point():
    """Отримує нульову точку з файлу або створює нову, якщо файл
    відсутній"""
```

```
    global zero_point
```

```
    if hx is None:
        print("⚠ HX711 not ready")
        return 0.0
```

```
    # спроба зчитати з файлу
    saved = load_zero_point()
    if saved is not None:
        zero_point = saved
        return saved
```

```
    # якщо файлу нема – робимо новий замір
```

```

try:
    print("✍ Calculating new zero point...")
    vals = [hx.get_value() for _ in range(10)]
    zero_point = sum(vals) / len(vals)
    save_zero_point(zero_point)
    return zero_point
except Exception as e:
    print("✖ Zero calc error:", e)
    return 0.0

```

```

def read_weight():
    if hx is None:
        return 0.0
    try:
        values = [hx.get_value() for _ in range(3)] # усереднення 3
        вибірок
        avg = sum(values) / len(values)
        adjusted = avg - zero_point
        return round(adjusted / CALIBRATION_FACTOR, 2)
    except Exception as e:
        print("⚠ Weight read error:", e)
        return 0.0

```

```

def start_monitor():
    """Постійно перевіряє вагу раз на 2 сек"""
    global check_timer, weight_dropped
    stop_monitor()
    weight_dropped = False
    print("🔔 Weight monitor started")

    def handler(timer):

```

```
global weight_dropped, last_weight
w = read_weight()
last_weight = w
print(f"👁️ Weight: {w} kg")
```

```
# Якщо вага менше 0.1 кг – сир знято
if w < 0.1:
    if not weight_dropped:
        print("✅ Weight dropped → cheese removed detected")
        weight_dropped = True
```

```
check_timer = Timer(-1)
check_timer.init(period=2000, mode=Timer.PERIODIC, callback=handler)
```

```
def stop_monitor():
    global check_timer
    if check_timer:
        try:
            check_timer.deinit()
        except:
            pass
    check_timer = None
```

Додаток Е. Лістинг файлу “wifi.py”

```
import network, socket, time, machine, ujson
from .device import get_device_key

WIFI_FILE = "wifi.json"

def load_wifi():
    try:
        with open(WIFI_FILE, "r") as f:
            return ujson.load(f)
    except:
        return None

def save_wifi(ssid, password):
    with open(WIFI_FILE, "w") as f:
        ujson.dump({"ssid": ssid, "password": password}, f)

def url_decode(s):
    return s.replace("%21", "!").replace("+", " ")

def connect_wifi():
    creds = load_wifi()
    sta = network.WLAN(network.STA_IF)
    ap = network.WLAN(network.AP_IF)

    if creds:
        print("🔑 Trying saved WiFi:", creds["ssid"])
        sta.active(True)
        sta.connect(creds["ssid"], creds["password"])
        for _ in range(20):
            if sta.isconnected():
                print("✅ Connected:", sta.ifconfig())
```

```
        ap.active(False)
        return True
    time.sleep(0.5)
    print("✕ Failed to connect")
```

```
ap.active(True)
ap.config(essid="PicoSetup", password="12345678")
print("🌐 AP started:", ap.ifconfig())
return False
```

```
def start_config_server():
    ap = network.WLAN(network.AP_IF)
    addr = socket.getaddrinfo("0.0.0.0", 80)[0][-1]
    s = socket.socket()
    s.bind(addr)
    s.listen(1)
    device_key = get_device_key()
    print("🌐 Connect to WiFi AP 'PicoSetup' and go to http://192.168.4.1")

    while True:
        # Скануємо доступні мережі при кожному відкритті сторінки
        sta = network.WLAN(network.STA_IF)
        sta.active(True)
        nets = sta.scan()
        ssid_options = "".join(
            f'<option value="{net[0].decode()}">{net[0].decode()} ({net[3]}
dBm)</option>'
            for net in nets
        )

        cl, addr = s.accept()
```

```

req = cl.recv(1024).decode()
if "ssid=" in req:
    try:
        params = req.split("ssid=")[1].split(" ")[0]
        ssid, password = params.split("&pwd=")
        ssid, password = url_decode(ssid), url_decode(password)
        save_wifi(ssid, password)
        cl.send("HTTP/1.0 200 OK\r\n\r\nSaved! Reboot Pico.")
        cl.close()
        time.sleep(2)
        machine.reset()
    except:
        cl.send("HTTP/1.0 200 OK\r\n\r\nError parsing")
        cl.close()
else:
    page = f"""HTTP/1.0 200 OK\r\nContent-type: text/html\r\n\r\n
<html>
<head>
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<style>
    body {{
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        font-family: Arial, sans-serif;
    }}
    .container {{
        text-align: center;
        border: 2px solid #333;
        padding: 20px;

```

```

        border-radius: 10px;
        box-shadow: 2px 2px 10px rgba(0,0,0,0.3);
    }}
    input, select {{
        padding: 6px;
        margin: 8px;
    }}
    input[type=submit] {{
        background: #4CAF50;
        color: white;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        padding: 8px 16px;
    }}
    input[type=submit]:hover {{
        background: #45a049;
    }}
</style>
<script>
    function togglePassword() {{
        var pwd = document.getElementById('pwd');
        pwd.type = (pwd.type === 'password') ? 'text' :
'password';
    }}
</script>
</head>
<body>
    <div class="container">
        <h2>Setup WiFi</h2>
        <form>
            SSID: <select name="ssid">{ssid_options}</select><br>

```

```
        Password: <input id="pwd" name="pwd" type="password"><br>
        <input type="checkbox" onclick="togglePassword()"> Show
password<br><br>
        <input type="submit" value="Save">
    </form>
    <p style="color:red"><b>Device Key: {device_key}</b></p>
</div>
</body>
</html>""
cl.send(page)
cl.close()
```

Додаток Є. Лістинг файлу “test_device.py”

```
# tests/test_device.py
import sys, os, json, tempfile, pytest

# Мок для urandom (стандартний Python 3)
class MockUrandom:
    def getrandbits(self, n):
        return 42 # фіксоване значення для детермінованого тесту

sys.modules['urandom'] = MockUrandom()

# Вставляємо тимчасову директорію для device.json
CHARS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'

def generate_device_key_impl(urandom_mod):
    return ''.join(
        CHARS[urandom_mod.getrandbits(8) % len(CHARS)]
        for _ in range(4)
    )

def test_key_length():
    key = generate_device_key_impl(MockUrandom())
    assert len(key) == 4, f'Key length should be 4, got {len(key)}'

def test_key_chars():
    key = generate_device_key_impl(MockUrandom())
    for ch in key:
        assert ch in CHARS, f'Invalid character: {ch}'

def test_key_deterministic():
    key1 = generate_device_key_impl(MockUrandom())
    key2 = generate_device_key_impl(MockUrandom())
```

```
assert key1 == key2, 'Same seed should produce same key'
```

```
if __name__ == '__main__':  
    test_key_length()  
    test_key_chars()  
    test_key_deterministic()  
    print('✅ All device tests passed')
```

Додаток Ж. Лістинг файлу “test_weight_monitor.py”

```
# tests/test_weight_monitor.py
CALIBRATION_FACTOR = 435000

def read_weight_impl(raw_values, zero_point):
    avg = sum(raw_values) / len(raw_values)
    adjusted = avg - zero_point
    return round(adjusted / CALIBRATION_FACTOR, 2)

def test_weight_calculation():
    zero = 100000
    # Очікувана вага: (535000 - 100000) / 435000 = 1.0 кг
    raw = [535000, 535000, 535000]
    w = read_weight_impl(raw, zero)
    assert abs(w - 1.0) < 0.01, f'Expected ~1.0 kg, got {w}'

def test_weight_zero():
    zero = 100000
    raw = [100000, 100000, 100000]
    w = read_weight_impl(raw, zero)
    assert w == 0.0, f'Expected 0.0 kg, got {w}'

def test_cheese_removed_detection():
    weights = [1.2, 0.8, 0.05] # остання < 0.1 → сир знято
    weight_dropped = False
    for w in weights:
        if w < 0.1 and not weight_dropped:
            weight_dropped = True
    assert weight_dropped, 'Should detect cheese removal'

def test_cheese_not_removed():
    weights = [1.2, 1.1, 1.0] # вага стабільна
```

```
weight_dropped = False
for w in weights:
    if w < 0.1:
        weight_dropped = True
assert not weight_dropped, 'Should not flag removal for stable weight'

if __name__ == '__main__':
    test_weight_calculation()
    test_weight_zero()
    test_cheese_removed_detection()
    test_cheese_not_removed()
    print('✅ All weight monitor tests passed')
```