

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

**Кваліфікаційна робота
на здобуття ступеня магістра**

за спеціальністю 122 Комп'ютерні науки
на тему:

**РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ ТА
МОНІТОРИНГУ ВИРОБНИЧИХ ПРОЦЕСІВ ЕЛЕВАТОРНОГО
КОМПЛЕКСУ**

Виконав студент 2-го курсу магістратури
Жук Ілля Ігорович

(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Омельчук Людмила Леонідівна

(підпис)

Засвідчую, що в цій роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент

(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри теорії та технології
програмування

« ____ » _____ 2021 р.,

протокол № ____
Завідувач кафедри

М. С. Нікітченко

(підпис)

РЕФЕРАТ

Обсяг роботи 107 сторінок, 41 ілюстрація, 18 таблиць, 23 джерела посилання.

GPS-ТРЕКЕР, МОНИТОРИНГ ПОЛОЖЕННЯ, ІНТЕРФЕЙС ПРОГРАМНОГО ПРОДУКТУ, ІНФОРМАТИЗАЦІЯ ЛОГІСТИЧНИХ ПРОЦЕСІВ, ДОСЛІДЖЕННЯ, ЕЛЕВАТОР, ІНТЕРФЕЙС ПРОГРАМНОГО ПРОДУКТУ, АВТОМАТИЗАЦІЯ ВИРОБНИЧИХ ПРОЦЕСІВ ЕЛЕВАТОРНОГО КОМПЛЕКСУ.

Об'єктом роботи є рухоме майно підприємства. Предметом роботи є система автоматизованого збору даних та моніторингу об'єкта розробки, що включає: проектування підсистеми апаратного устаткування, проектування системи контролю, проектування програмного пакету.

Метою роботи є розробка та реалізація апаратно-програмного комплексу моніторингу положення об'єкту у просторі з параметризацією пересувань та автоматизацією процесів елеваторного комплексу з реалізацією пакету SCADA.

Методи розроблення. програмування контролерів типу SoC (System on Chip), побудова компонентних апаратних систем. Програмування контролерів типу ПЛК, побудова системи ЛМВ. Інструменти розроблення: пропрієтарне, інтегроване середовище розробки програмного забезпечення систем автоматизації технологічних процесів TIA Portal 15, а саме, компоненти Simatic Step 7 Basic(Professional), Simatic WinCC 7.4, середовище розробки програмного забезпечення Visual Studio Code, засіб прошивання пам'яті ESPRESSIF Flash Download Tool.

Результати роботи. Проведено дослідження поточного стану елеваторних комплексів та апаратних комплексів модулів зв'язку, розроблено програмний продукт для опрацювання GPS-повідомлень з реалізацією

стаціонарного та мобільного центру моніторингу. спроектовано технічне завдання та розроблено програмний продукт для автоматизації.

Підставою для виконання роботи є проведення дослідження з модернізації, моніторингу та автоматизації апаратного устаткування логістичних підприємств.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	6
ВСТУП	7
РОЗДІЛ 1 ОСНОВИ ПРОЕКТУВАННЯ МОНІТОРИНГОВИХ СИСТЕМ	12
1.1 Дослідження об'єкту розробки	12
1.2 Дослідження об'єкту автоматизації	14
1.2.1 Електромеханічне устаткування елеваторного комплексу	15
1.2.2 Компонентне дослідження	16
1.2.3 Формування вимог до АС	21
1.3 Дослідження об'єкту моніторингу	24
1.3.1 Устаткування пересувних об'єктів	28
1.3.2 Компонентна структура суміжних систем	31
1.3.3 Формування вимог до моніторингової системи	32
1.4 Концепція управління виробничим процесом	35
1.4.1 Локальні зони контролю	36
1.4.2 Устаткування локальних зон контролю	37
1.4.3 Система інтегрованого моніторингу	43
РОЗДІЛ 2 РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ	45
2.1 Розробка проектних рішень АС	45
2.1.1 Побудова моделі системи	46
2.1.2 Технологічна схема	47
2.1.3 Розробка підсистеми введення-виведення	49
2.1.4 Розробка підсистеми контролю та обчислень	51
2.1.5 Розробка алгоритмічного представлення	55
2.1.6 Розробка підсистеми диспетчеризації	59
2.1.7 Розробка інтерфейсу оператора	60
РОЗДІЛ 3 Реалізація моделі системи	63
3.1 Програмне забезпечення	63

3.2 Технологічне представлення системи	66
3.3 Компонентне представлення системи	72
3.4 Інструментарій розробки пристрою моніторингу	74
3.5 Побудова системи	80
3.6 Реалізація фонового моніторингового процесу	82
3.7 Реалізація обміну даними з модулем GPS-приймача	82
ВИСНОВКИ	86
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	88
ДОДАТОК А. Вибір сфери дослідження	91
ДОДАТОК Б. Конфігурування мобільного застосунку	93
ДОДАТОК В. Інтеграція Vlynk API у проект	97
ДОДАТОК Г. Реалізація динамічного підключення	102
ДОДАТОК Д. Система адаптивних сповіщень	106

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

SCADA	- Supervisory Control and Data Acquisition, диспетчерське управління і збір даних.
ПЛК	- програмований логічний контролер.
ЛМВ	- людино-машинна взаємодія
TIA	- Totally integrated automation, повністю інтегрована автоматизація
IAS	- Integrated automation system, інтегрована система автоматизації
АСУ	- автоматизована система управління
ВВП	- Внутрішній валовий продукт
COT	- Світова організація торгівлі
АПК	- агропромисловий комплекс
ISO	- International Organization for Standardization, Міжнародна організація зі стандартизації
ТП	- технологічний процес
АРМ	- автоматизоване робоче місце
ПК	- персональний комп'ютер
UPS	- Uninterruptible Power Supply, джерело безперебійного живлення
ІІМ	- інформаційно-логічна модель
КС	- Комп'ютерна система
HMI	- Human-machine interface, людино-машинний інтерфейс
NFC	- Near field communication, комунікація ближнього поля
TMS	- Transportation Management System

ВСТУП

Оцінка сучасного стану об'єкта розробки. Технологічний процес роботи елеватора полягає в механічному керуванні агрегатом групою операторів. Основним недоліком застарілого підходу є людський фактор, час на реакцію при аварійній ситуації, складність визначення станів системи та її механічних елементів. В поточних умовах роботи елеватора може скластися ситуація, в якій робочий персонал не зможе своєчасно відреагувати на нештатну подію. Своєчасне запобігання аварійних станів системи є одним з головних факторів оптимальної роботи агрегатів.

Дослідження характеристик сільськогосподарської продукції, що знаходяться на зберіганні є трудомістким процесом, який полягає в фізичних замірах параметрів, передачі даних до лабораторії, яка обчислює оптимальні умови зберігання продукту і надає рекомендації щодо корегування станів системи. Механічний спосіб отримання інформації є одним з найменш ефективних підходів у виробництві. Скорочення витрат по часу є запорукою підвищення ефективності праці, що веде за собою реструктуризацію штату компанії. Працівники, обов'язки яких буде автоматизовано в рамках модернізації, після перекваліфікації отримають нові робочі місця.

У рамках розвитку інтернету речей та інтегрованих систем автоматизації проводяться роботи зі створення мереж пов'язаних пристроїв. Наразі ступінь залучення мережевих технологій знаходиться на рівні створення незалежних об'єктів готових до глобального індустріального інтегрування. У дипломній роботі мною було проведено розробку відкритої системи керування агрегатами елеватору. Дана робота є експериментальним дослідженням можливостей використання технологій глобальної навігації для відстежування рухомих транспортних засобів у реальному часі. Транспортна логістика представляє невід'ємну частину бізнес процесу елеваторних підприємств. Дослідження можливостей оптимізації та автоматизації охоронних систем рухомого складу представляється перспективною ланкою глобальної інтеграції бізнес процесів та систем моніторингу і керування.

Актуальність роботи та підстави для її виконання. При розробці даного проекту, було визначено необхідність проектування комплексу заходів з усунення проблем пов'язаних з технологічним процесом, людським фактором і аварійними станами системи.

Основним підходом до організації сучасного технологічного процесу є інформатизація, яка надає змогу використовувати такі переваги: наявність єдиного інформаційного простору, автоматична обробка інформації, організація продуктивної діяльності системи, реалізація інформаційних відношень. Використання інформатизації в даному проекті полягає в централізації збору та обробки даних обраного підприємства.

Наявність єдиного інформаційного простору визначає можливість автоматизації системи, оскільки повна інформація про поточний стан системи надає змогу приймати відповідні сценарні рішення. Управління станами системи реалізовано за допомогою комп'ютеризації, що дало змогу централізувати засоби керування, автоматизувати технологічний процес та модернізувати підприємство.

Вимірювальні пристрої, використані в рамках проекту, отримують інформацію в реальному часі та знаходяться безпосередньо в робочій зоні, що позитивно впливає на швидкість прийняття рішень щодо роботи агрегатів та забезпечує автоматичний цілодобовий контроль над всіма компонентами структури. Аварійний контроль більше не потребує безпосередньої участі персоналу, а швидкість реагування системи на критичні збої механізмів відповідає системам реального часу.

На вірне прийняття рішення оператора впливає інформаційна повнота, коректність та своєчасність представлення інформації, яку він отримує з програмного продукту. В даному проекті було запроваджено програмне забезпечення, яке дає можливість керувати та спостерігати за всіма параметрами системи одному оператору, що значно спростило роботу всього підприємства.

При проведенні досліджень з можливості розробки системи моніторингу, було визначено необхідність проектування апаратно–програмного комплексу зі створення методів усунення проблем пов'язаних з технологічним процесом, людським фактором і моніторингу працездатності систем об'єктів відстеження.

Наявність єдиного інформаційного простору визначає можливість автоматизації системи, оскільки повна інформація про поточний стан системи в цілому та окремих її складових надає змогу приймати відповідні сценарні рішення. Управління станами системи реалізовано за допомогою комп'ютеризації, що надасть змогу централізувати засоби керування, автоматизувати технологічний процес та модернізувати підприємство.

Вимірювальні пристрої, використані в рамках проекту, отримують інформацію в реальному часі та знаходяться безпосередньо у об'єкті відстежування, що забезпечує високу швидкість збору даних та автоматичний цілодобовий контроль над всіма компонентами структури.

Мета й завдання роботи. Метою кваліфікаційної роботи є створення програмного засобу для спрощення користування агрегатами та мінімізації витрат часу на обробку та зберігання зернових культур. Для досягнення цієї мети поставлено такі завдання:

- Дослідити можливість впровадження автоматизації в аграрну сферу
- Дослідити актуальність застосування інтегрованих систем збору інформації та автоматизованого управління в аграрній сфері;
- Розробити технічне завдання до продукту;
- Розробити інтерфейс програмного продукту;
- Запрограмувати контролери на виконання визначених алгоритмів, що автоматизують або пришвидшують роботу системи;
- Дослідити можливість локального обміну даними з системами глобального позиціонування;
- Дослідити можливість моніторингу положення об'єкту у просторі.

Об'єкт, методи й засоби розроблення. Об'єктом розроблення програмного засобу «IAS Management» є систематизація роботи заготівельного зерносховища (елеватор) шляхом впровадження інтегрованої системи збору інформації та автоматизованого управління.

Під час розробки програмного продукту використана ітераційна модель, згідно якої життєвий цикл проекту складається з наступних етапів:

- Визначення вимог до продукту;
- Специфікація вимог;
- Проектування;
- Реалізація;
- Тестування та налагодження;
- Експлуатація та підтримка.

А головним принципом є вільний перехід між стадіями розробки продукту у зворотному напрямку. Основними факторами вибору цієї моделі було:

- Зниження можливості нанесення серйозних фінансових збитків на ранніх стадіях проекту, що веде до мінімізації витрат на їх усунення;
- організація ефективного зворотного зв'язку проектної команди зі замовниками і створення продукту, реально задовольняючого його вимогам.

В якості інструменту створення програмного засобу було обрано Siemens TIA Portal 15 – інтегроване середовище розробки програмного забезпечення систем автоматизації технологічних процесів від рівня приводів і контролерів до рівня людино–машинного інтерфейсу.

Можливі сфери застосування. Програмний продукт «IAS Management» може застосовуватися в аграрних підприємствах різного типу з подальшою доробкою за потребою клієнта.

РОЗДІЛ 1

ОСНОВИ ПРОЕКТУВАННЯ МОНІТОРИНГОВИХ СИСТЕМ

В даному розділі досліджено особливості протікання виробничих, логістичних, біологічних та бізнес процесів виробництва зернової продукції в аграрній сфері. Описано вимоги до створення автоматизованих моніторингових систем управління за міжнародними стандартами. Детально представлені компоненти об'єкта автоматизації, наведено базову інформацію про суб'єкти виробничого процесу.

1.1 Дослідження об'єкта розробки

Для дослідження об'єкта розробки проведено розгляд окремих структурних елементів високорівневої моделі системи, наведених в розділах 1.2, 1.3, 1.4.

Дослідження етапів. Основні етапи бізнес–процесу з вироблення зернової продукції [6] наведено у таблиці 1.1.

Таблиця 1.1 Етапи бізнес–процесу з вироблення зернової продукції

№	Назва етапу	Опис
1	Виробництво	Комплекс заходів з рекультивації, селекції та подальшого вирощування культур
2	Розподіл	Груповий розподіл зернових культур при зборі врожаю та транспортування для подальшої обробки зерна
3	Переробка	Зберігання, сушіння, первина обробка зерна. Після проходження цього етапу отримується готовий до збуту продукт
4	Збут	Реалізація готового продукту на ринку сільськогосподарської продукції

Вищезазначені етапи супроводжуються бухгалтерським обліком, проте розгляд цього процесу не є актуальним у рамках даної роботи.

Рівень виробництва. Автоматизація збору врожаю не проводиться, адже потребує роботизації складних зернозбиральних та транспортувальних машин й зумовлює придбання додаткового спеціального обладнання і засобів локального контролю.

Рівень розподілу. Розподіл зернових культур (Далі ЗК) є простим лінійним процесом, що полягає у роздільному опрацюванні отриманої сировини. Різні культури доставляються на об'єкти зберігання окремими транспортними засобами, після чого настає етап переробки (№3 згідно таблиці 1.1).

Рівень переробки. Даний етап реалізує систему заходів наведену у таблиці 1.2.

Рівень збуту. Етап збуту передбачає:

- Збутова логістика;
- Вибудовування взаємин із зовнішньою збутовою мережею;
- Налагодження відносин з продавцями та потенціальними покупцями;
- Виконання зобов'язань постачальника в частині поставки продукції споживачеві.

Крім процесів які піддаються інформатизації, цей етап сильно залежить від ринкових умов, які включають в себе маркетинг, основи підприємницької діяльності, знання економіки та ринкових відносин.

Створити групу вузькоспеціалізованих систем обліку можливо, але ринок переповнений готовими рішеннями. Забезпечити інтеграцію нового продукту зі старими рішеннями навряд чи змінить існуючий порядок обліку, а інтегрування всіх етапів в єдину систему є неможливим, оскільки збут готової продукції сильно залежить від маркетингових рішень та фінансових стратегій.

Аргументація. В даній роботі проводиться заходи з модернізації рівня переробки, оскільки саме на цьому рівні зосереджені виробничі потужності, які мають високий модернізаційний потенціал. Класичні механізми з параметричним, часто лінійним управлінням, наявність кінцевого набору станів, які є необхідним умовою при створенні об'єктної моделі.

1.2 Дослідження об'єкта автоматизації

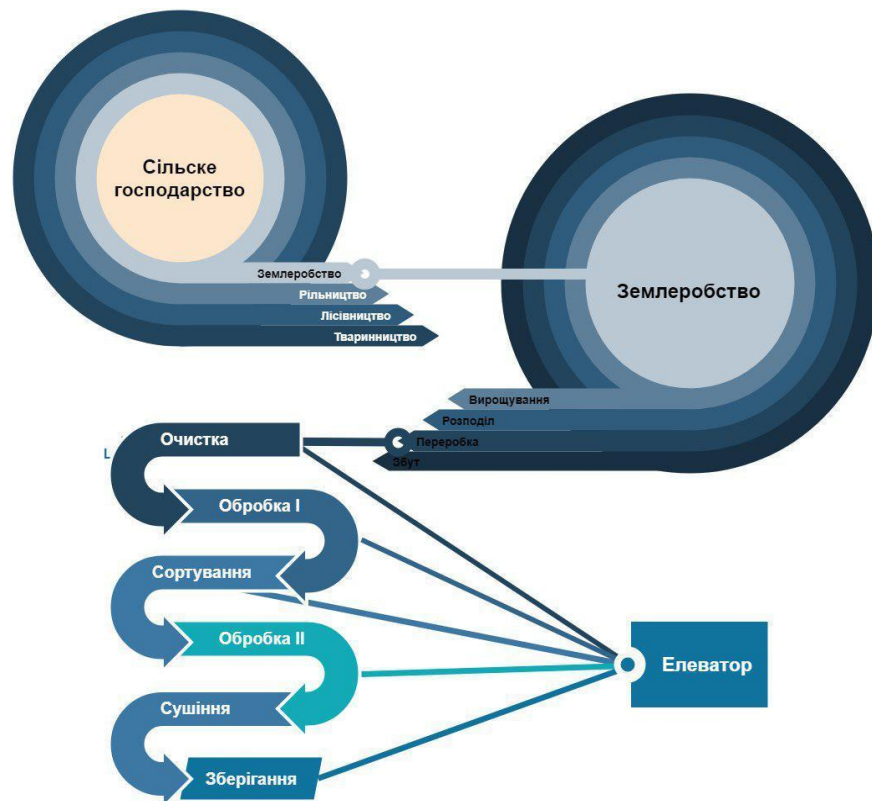


Рисунок 1.1 Функціонал елеватора в бізнес–процесі

Елеватор – це комплекс систем зі зберігання, сушіння та первинної обробки зернових культур, що складається з механічних агрегатів та ручної системи управління. Рівень типової автоматизації полягає в тому, що персонал має можливість дистанційно керувати обладнанням, при даному підході відсутня можливість здійснювати контроль параметрів роботи системи, аварійна безпека об'єкту залежить від швидкості реакції персоналу та рівня обізнаності про реальний стан механізмів. Системи додаткового незалежного контролю відсутні, що знижує рівень інформаційної повноти.

Післяаварійні експертизи ускладнені відсутністю культури документування ТП.

На актуальному етапі розвитку технічних засобів контролю та управління елеваторними системами активно використовується комп'ютеризація та інформатизація ТП. Такий комплекс зумовлює створення інформаційної моделі системи, що надає змогу використовувати принципи автоматизації та ефективізації виробництва. Ряд провідних світових компаній пропонують апаратні й програмні рішення з реалізації наведених принципів.

1.2.1 Електромеханічне устаткування елеваторного комплексу

Слід зауважити, що виконавчими механізмами для процесів 1–5 включно (згідно таблиці 1.3) є агрегати, що керуються на рівні сигналів. Таким чином, вони є складними виконавчими механізмами, управління якими здійснюється системами введення–виведення.

Таблиця 1.3 Заходи з переробки зернової продукції

№	Назва процесу	Опис
1	Очистка сировини	Видалення забруднень та домішок з загальної маси ЗК
2	Гідротермічна обробка зерна	Підготовка зерна до фракціонування та механічної обробки, зміна його фізичних показників
3	Фракціонування та сортування	Виділення та виокремлення проміжних продуктів переробки
4	Механічна обробка	Приведення сировини до відповідної форми та контроль вимог якості.
5	Сушка сировини	Завершальний етап переробки, створення умов для довгострокового зберігання та подовження строку придатності продукції
6	Транспортування продукції	Фізичне переміщення продукції по локальній або регіональній зоні

1.2.2 Компонентне дослідження

Компонентне представлення елеватора наведене в таблиці 1.4.

Таблиця 1.4 Компонентне представлення системи

№	Назва елемента	Тип	Опис
1	Силос	Елемент	Сховище для зберігання та підтримки оптимальної кондиції зерна
2	Транспортер	Механізм	Тяговий і вантажний орган конвеєра

3	Норія	Механізм	Орган конвеєра, призначений для вертикального пересування зерна по системі
4	Перекидний клапан	Механізм	Елемент системи для здійснення регулювання завантажувально–навантажувальних робіт зернопродуктів з тари за допомогою перенаправлення потоків
5	Засувка рейкова автоматична	Механізм	Встановлюється в місцях вивантаження накопичувальних ємностей, бункерів, складів, зерносховищ силосного типу.
6	Сепаратор	Агрегат	Агрегат, який використовується для очищення та калібрування зерна.
7	Сушарка	Агрегат	Агрегат, який використовується для сушіння зернової продукції
8	Самоплив	Елемент	Елемент системи для руху зерна під дією сили власної ваги

Висновки з автоматизації елементів елеватора представлено у таблиці 1.5

Таблиця 1.5 Можливість автоматизації компонентів

Компонент сис–ми	Можливість автоматизації	Аварійний контроль	Засоби автоматизації
1	+	–	Датчики
2	+	–	Контролери та датчики

3	+	+	Контролери та датчики
4	+	+	Контролери та датчики
5	+	+	Контролери та датчики
6	+	–	Контролери та датчики
7	–	–	Контролери та датчики
8	–	–	Не потребує

Описане в таблиці 1.4 обладнання чітко типізується за ролями в системі. Доцільно провести його групування для спрощення подальшого добору контролерів та визначення рівня навантаження на підсистеми збору даних та керування. Групування дозволить використовувати вузькоспеціалізовані контролери, які не потребують налаштування портів, залежно від режиму роботи прикладного обладнання. При використанні такої моделі кожен пристрій отримує власний адрес, за яким контролер може до нього звертатися та здійснювати обмін даними. Діаграма підключень наведена на рисунку 1.2, опис представлених груп наведено у таблиці 1.6.

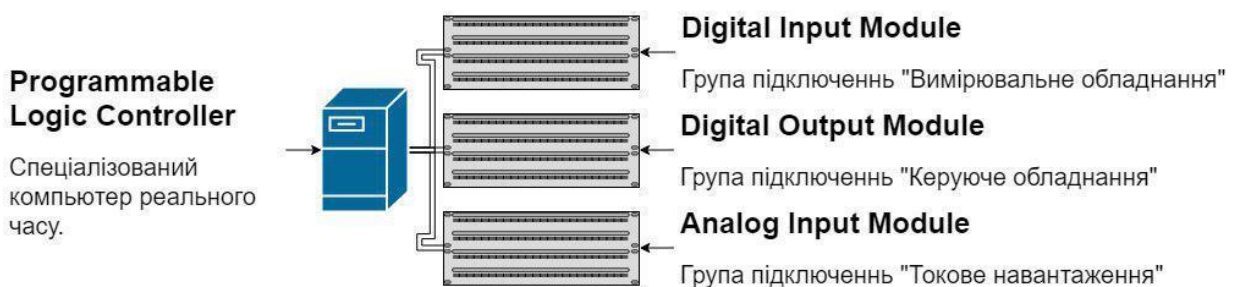


Рисунок 1.2 Інтерфейси підключення компонентів до контролера

Таблиця 1.6 Представлення груп керування

Тип	Група	Опис	Абревіатура
Digital output module	Керуюче обладнання	Патч–панель для адресації та управління станами елементів системи	DQ
Digital input module	Вимірювальне обладнання	Патч–панель для адресації та управління вимірювальними пристроями системи	DI
Analog input module	Токове навантаження	Патч–панель для вимірювання токового навантаження на елементах системи	AI

Кожен елемент елеватора може належати до жодної, однієї або декількох груп керування, що зумовлено конструктивними особливостями механізмів. Належність елементу елеватора до певної групи контролю представлена у таблиці 1.7.

Таблиця 1.7 Належність компонентів системи до груп керування.

№	Назва елемента	DQ	DI	AI
1	Силос	+	–	–
2	Транспортер	+	+	+
3	Норія	+	+	+
4	Перекидний клапан	+	+	+
5	Засувка рейкова	+	+	+
6	Сепаратор	+	+	+
7	Сушарка	+	+	+
8	Самоплив	–	–	–

Параметри контролю та керування компонентів типу «Елемент» (згідно таблиці 1.4) наведено у таблиці 1.8.

Таблиця 1.8 Параметри контролю та керування типом «Елемент»

Назва елемента	Параметри контролю	Параметри керування
Силос	Датчик температури	–
	Датчик наповненості	–
Самоплив	–	–

Механізми в системі елеватора є аварійно небезпечними компонентами, тому потребують контролю станів та аварійного захисту. Параметри контролю та керування компонентів типу «Механізм» (згідно таблиці 1.4) наведено у таблиці 1.9.

Таблиця 1.9 Параметри контролю та керування типом «Механізм»

Назва елемента	Параметри контролю	Параметри керування
Транспортер	Контроль захисту механізму	Включення механізму
	Контроль роботи механізму	
	Контроль датчика підпору	
	Контроль датчика швидкості	
Норія	Контроль захисту механізму,	Включення механізму
	Контроль роботи механізму	
	Контроль датчика підпору	
	Контроль датчика швидкості	

Перекидний клапан	Контроль захисту механізму,	Керування положеннями (вліво та вправо)
	Контроль відкриття механізму	
	Контроль закриття механізму	
Засувка рейкова	Контроль захисту механізму,	Керування положеннями (відкрити і закрити)

Агрегати є найскладнішими компонентами системи, що потребують моніторингу стану, автоматичного аварійного контролю та штатного віддаленого керування. Параметри контролю та керування компонентів типу «Агрегат» (згідно таблиці 1.4) наведено у таблиці 1.10.

Таблиця 1.10 Параметри контролю та керування типом «Агрегат»

Назва елементу	Параметри контролю	Параметри керування
Сепаратор	Контроль захисту механізму	Включення механізму
	Контроль роботи механізму	
Сушарка	Контроль захисту механізму	Включення механізму

1.2.3 Формування вимог до АС

Система управління повинна бути побудована за класичною трирівневою структурою АСУ ТП, яка передбачає виокремлене представлення ланок клієнтських застосунків (рівень 1), обробляючих серверів (рівень 2) та підсистеми зберігання даних (рівень 3) [7].

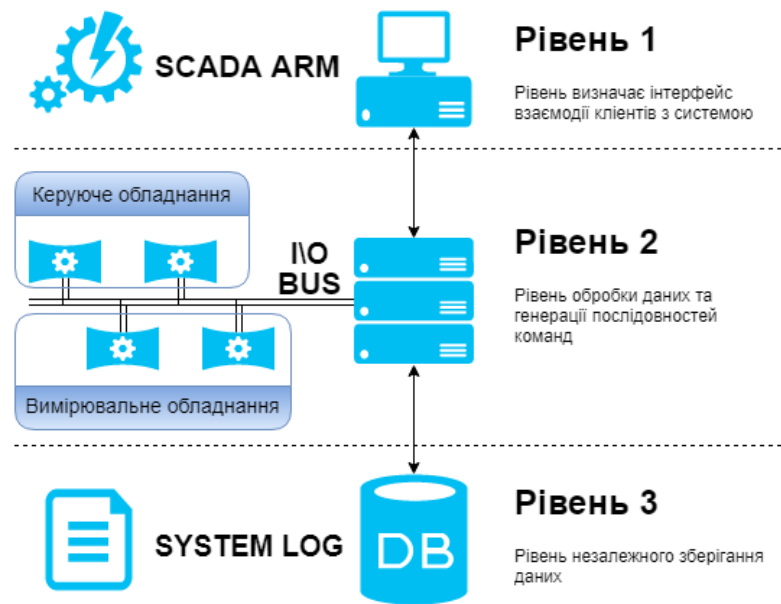


Рисунок 1.3 Трирівнева структура побудови системи

Рівень 1. Система диспетчерського контролю і управління, АРМ оператора. АРМ оператора представляє собою людино–машинний інтерфейс SCADA системи WinCC на ПК, що дозволяє здійснювати контроль та управління технологічним процесом та обладнанням, вести облік параметрів роботи обладнання, своєчасно виявляти відхилення від нормальної роботи і попереджувати виникнення аварійних ситуацій.

Архітектура умов розгортання SCADA–систем наведена на рисунку 1.3.

Діаграма включає в себе:

- Рівні та їх компоненти;
- Групи компонентів;
- Елементи сполучення.

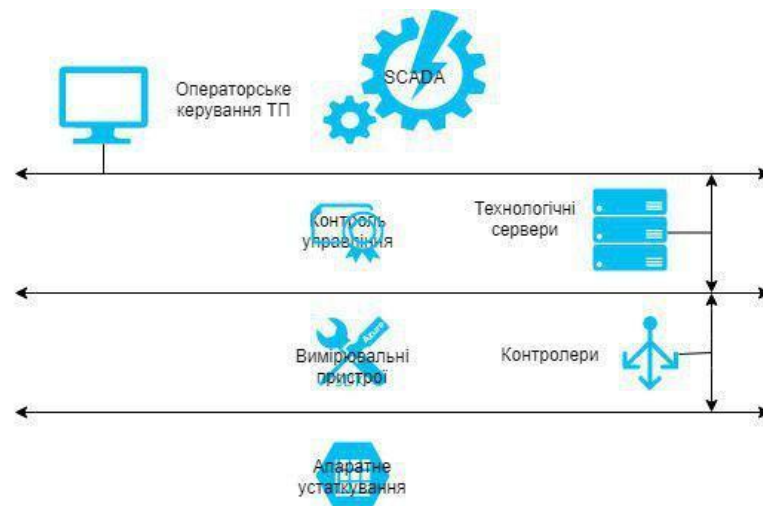


Рисунок 1.4 Представлення SCADA–системи

Рівень 2. На даному рівні система АСУ ТП повинна здійснювати:

- Контроль роботи електродвигунів за допомогою розподільчої апаратури (перетворювачі частоти, електронні пристрої пуску, автоматичні вимикачі);
- Контроль параметрів технологічного процесу за допомогою збору та обробки даних від контрольно–вимірювальних приладів;

Забезпечувати:

- Обробку отриманих даних з формуванням необхідних запобіжних сигналів оператору;
- Безпечну зупинку технологічних ділянок в разі виникнення технологічного захисного блокування за сигналом або сукупності логічно пов'язаних сигналів.

Рівень повинен бути представлений центральним процесором Siemens серії S7–1500 з системою децентралізованої периферії – інтерфейсні модулі Siemens ET200SP з модулями вводу–виводу сигналів. Зібрана від модулів ET200SP інформація повинна бути передана для подальшої обробки у центральний керуючий модуль обчислень серії S7–1500 за допомогою промислового протоколу передачі даних Industrial Ethernet.

Місця розташування на ситуаційному плані і кількість інтерфейсних модулів ET200SP обрати виходячи з технологічної необхідності (за результатами розрахунків кількості і типів оброблюваних сигналів).

АСУ ТП повинна мати змогу зовнішнього внесення змін в будь-який розділ програмного забезпечення без зупинки технологічного процесу.

АСУ ТП повинна реалізувати функціонал безперервного контролю працездатності власних підсистем, забезпечувати сигналізацію аварійних та предаварійних станів. При відмові Системи в цілому, виконавчі механізми повинні перейти у положення, визначене комплексом станів, що мають назву «Аварійна зупинка» і будуть визначатися їх конструктивними особливостями.

Енергобезпеку системи повинен забезпечувати комплекс UPS. При вичерпанні ємності якого технологічний процес повинен бути зупинений, а система переведа у стан «Аварійна зупинка».

Конструкція АСУ ТП повинна забезпечувати кінцевому користувачу взаємозамінність однотипних складових частин.

Рівень 3. На даному рівні система АСУ ТП повинна забезпечити збір даних та параметрів технологічного процесу через опитування відповідних контрольно-вимірювальних приладів та датчиків системи розподілення, здійснювати керуючий вплив на виконавчі механізми (електродвигуни та сервоприводи).

Пристроями впливу на виконавчі механізми виступатимуть: перетворювачі частоти, пристрої плавного пуску, електронні пристрої пуску (твердотільний реверсивний пускач, твердотільні реле), магнітні пускачі, реле.

1.3 Дослідження об'єкту моніторингу

Прийняте припущення. В даній роботі будуть розглядатися питання, пов'язані з прийманням і відвантаженням зерна, доставленого на елеватор або вивозиться з елеватора, тільки за допомогою автотранспорту. Під виразом «Автоматизація внутрішньої логістики елеватора», я маю на увазі

автоматизацію наступних процесів, які мають місце на кожному елеваторі в варіанті доставки зерна на елеватор автотранспортом:

- Ідентифікація автотранспорту, введення даних в систему обліку;
- Взяття проб зерна для визначення його якості;
- Зважування автотранспорту – бруто;
- Вивантаження зерна з автотранспорту;
- Зважування автотранспорту – нетто;
- Ідентифікація автотранспорту, введення даних в систему обліку.

При відвантаженні зерна з елеватора на автотранспорт всі вище перераховані операції повторюються в зворотному порядку.

Проведемо аналіз питання автоматизації зовнішньої логістики елеватора. Під виразом «Автоматизація зовнішньої логістики елеватора», мається на увазі автоматизацію наступних процесів, які мають місце на елеваторах, в варіанті доставки зерна на елеватор автотранспортом:

1. Оптимізація кількості автотранспорту, що доставив зерно на елеватор протягом кожної години протягом добового графіка роботи елеватора. Іншими словами, управління чергою на вивантаження зерна на елеваторі протягом доби;
2. Управління процесом поставки зерна на елеватор по зерновим культурам та їх якості;
3. При розташуванні елеватора в житловій зоні керування транспортними потоками протягом доби, а також управління маршрутами руху автотранспорту в житловій зоні;
4. Оптимізація необхідної майданчики відстою автотранспорту перед елеватором, перед вивантаженням зерна на елеватор;
5. При розташуванні елеватора в житловій зоні, і необхідності пункту –(ів) накопичення автотранспорту з зерном за межами населеного пункту:

- оптимізація необхідної площі пункту (–ів) накопичення автотранспорту з зерном; перенесення первинної обробки автотранспорту з зерном (взяття проби зерна, аналізи зерна) на пункт (и) накопичення автотранспорту і автоматизований обмін інформацією лабораторних пунктів поза елеватора з лабораторією на елеваторі.

У рамках розвитку інтернету речей та інтегрованих систем автоматизації проводяться роботи зі створення мереж пов'язаних пристроїв. Наразі ступінь залучення мережевих технологій знаходиться на рівні створення незалежних об'єктів готових до глобального індустріального інтегрування. Однією з активних галузей індустріального моніторингу є дослідження можливостей використання технологій глобальної навігації для відстежування рухомих транспортних засобів у реальному часі. Транспортна логістика представляє невід'ємну частину бізнес процесу елеваторних підприємств. Дослідження можливостей оптимізації та автоматизації охоронних систем рухомого складу представляється перспективною ланкою глобальної інтеграції бізнес процесів та систем моніторингу і керування.

TMS. Згідно визначення Gartner, TMS використовується для планування та оцінки процесу вантажоперевезень, автоматизації оцінки показників ефективності операції, вибору оптимального маршруту і перевізника, а також управління рахунками і платежами.

Ключові процеси управління перевезеннями: Планування, транспортування, процесуальна комунікація та параметричні розрахунки.

Планування. Система керування транспортування обирає найбільш ефективну схему перевезення, враховуючи параметри: вартості перевезення, кількість залученого рухомого складу, параметри та показники на всій часовій протяжності процесу перевезення.

Транспортування. Моніторинг процесу доставки вантажу у цілодобовому режимі. Автоматизація роботи диспетчера та контролю

переміщення рухомого складу із залученням сторонніх сервісів глобальної навігації. Встановлення додаткового обладнання надає змогу локальній системі збору даних періодично опрацьовувати стани системи, що надає змогу отримувати фіксовані у часі набори показників на всій протяжності маршруту транспортування. Найбільш ефективним підходом є збереження даних з системи локального моніторингу у постійній пам'яті моніторингового пристрою, з виокремленою процедурою представлення даних на наступні за рівнем абстракції рівні.

Типовий набір для викладеного у роботі об'єкта дослідження параметрів, що рекомендовані до моніторингу, представлений у переліках глобального та локального моніторингу нижче.

До глобального моніторингу рекомендується віднести наступні параметри для одиниці рухомого складу:

- Об'ємно–вагова наповненість;
- Маршрут слідування;
- Відхилення у часі проходження схеми маршруту;
- Відхилення у траєкторії проходження схеми маршруту.

Процесуальна комунікація. Процедура транспортування реалізується із залученням варіативної множини вантажно–пересувних засобів, організація єдиної суб'єктної системи комунікації представляє одне із завдань при модернізації логістичного процесу.

Параметричні розрахунки. TMS–рішення також включає представлення про автоматизацію розрахункових процесів, пов'язаних з перевезенням виробничої сировини або продукції. Серед впливових показників: оптимальні втрати вантажу, облік запасів, добові норми витрат палива, оптимальна кількість зупинок. Бізнес–логіка TMS аналізує дані, що надходять на основі ключових показників ефективності.

Комплекс програмних рішень також може включати модулі автоматизації моніторингу та управління побічних об'єктів:

- Система управління складом (WMS);
- Система планування ресурсів підприємства (ERP);
- Система управління активами підприємства (EAM);
- Система супроводу виробництва (MES);
- Система управління ланцюгами поставок (SCM).

Рада з питань захисту природних ресурсів (NRDC) підрахувала, що до 40% виробленого продовольства, втрачає кондиційний стан до моменту отримання кінцевим споживачем. Експерти NRDC вважають, що значний обсяг втрати якості сировини викликаний проблемами в ланцюзі поставок. Фахівцям часто бракує актуальних даних для ефективного прийняття рішень. Таким чином, приблизно одна третина сировини та продуктів харчування псується передчасно, і це призводить до збільшення витрат для фермерів, вантажовідправників, роздрібних торговців і кінцевих споживачів.

1.3.1 Устаткування пересувних об'єктів

Під час проходження поточного етапу інформатизації зберігається проблема інтеграції даних з локальних та мобільних систем у єдину інформаційну модель.

Прикладом такого рішення є TMS – система управління перевезеннями. Не менш важливим для виробничих процесів є можливість зберігати та накопичувати дані про параметри кондиції товару або сировини а також поточне положення товару, що транспортується.

Сучасні Системи управління транспортом надають детальну інформацію про:

- Історію відвантаження;
- Прогноз попиту;
- Варіанти та умови перевізника;
- Транзитний вантаж.

Усі ці деталі можна передавати в режимі реального часу, а автоматичні сповіщення дозволяють вантажовідправникам виявляти основні причини проблем та вживати заходів для їх компенсації. Завдяки цьому Системи управління транспортом стали важливими елементами ланцюгів поставок, які з розвитком стають менш вразливі до більшої кількості збоїв і більш адаптивні до затримок. Крім того, деяким галузям доводиться долати тиск попиту піку сезону, і у випадку зриву будь-якому бізнесу потрібно якомога швидше відновити свою транспортну мережу до нормальної роботи. Ось чому програмне забезпечення для управління ланцюгами поставок – це потужний інструмент, здатний надавати точні деталі в потрібний момент, що дозволяє приймати складні композитні рішення щодо профілю ризику бізнесу та періоду, що розглядається.

Завдяки представленому цифровому рішенню можна:

- інтегрувати та автоматизувати основні потоки даних;
- виключити ручне введення даних;
- уникати людських помилок;
- зменшити кількість маніпуляцій;
- швидше передавати інформацію;
- економити час та скоротити пов'язані витрати.

Програмне забезпечення для супроводу і організації бізнес-процесу виробничих комплексів повинно поєднувати автоматизовану інтеграцію даних, візуалізацію та аналітику, дозволяючи планування попиту та координацію підрядників. Це також зробить можливим відстежування більшості показників індивідуальної і суб'єктивної ефективності та надавати клієнтам краще розуміння можливих збоїв у ланцюгу поставок. Апаратно-процесуальна архітектура наведена на рисунку 1.5.

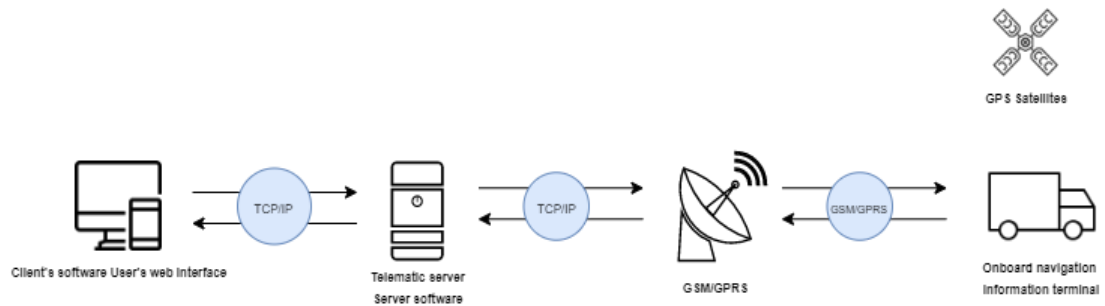


Рисунок 1.5 Архітектура моніторингового процесу відстежування

1.3.2 Компонентна структура суміжних систем

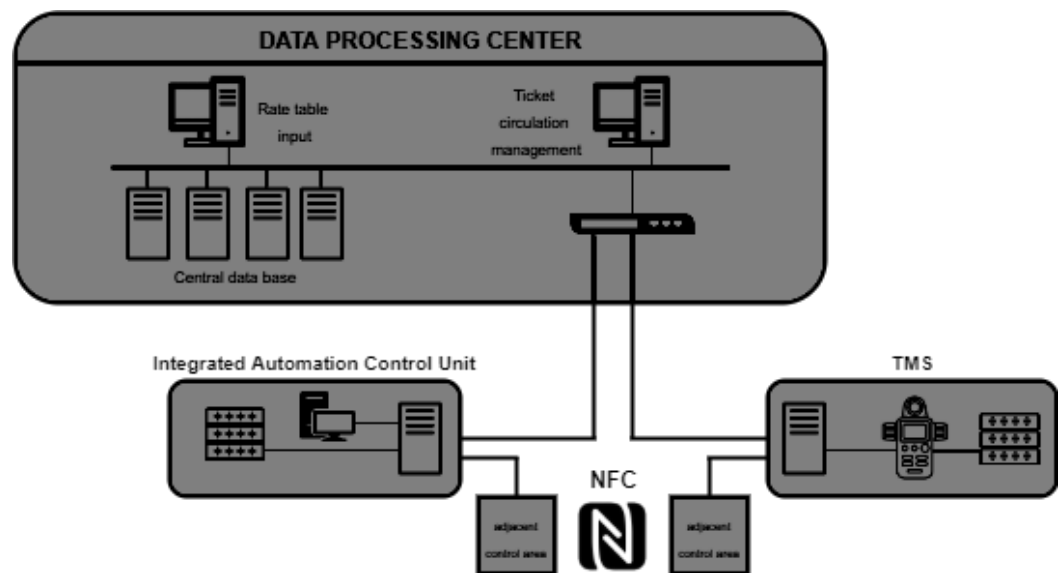


Рисунок 1.6 Даталогічна архітектура суміжних систем

Основи архітектури системи супроводу повного циклу виробництва та постачання включають IAC та TMS–системи, а також всі їх складові підсистеми наведені на рисунку 1.6. Ці інформаційні системи можуть бути як стандартними програмними пакетами з налаштування параметрів або адаптування програмних програм сконфігуровані відповідно до конкретних потреб компанії. Основні системи в ланцюгу поставок забезпечують специфічні функції для типових груп користувачів:

1. ERP, системи планування ресурсів підприємства: закупівля, управління матеріалами та продажами;

2. WMS, системи управління складом: зберігання даних про продуктово–економічні транзакції, управління ресурсами і компонування замовлень;
3. TMS, системи управління транспортом: бронювання транспорту, планування та моніторинг.

Послідовне управління даними за допомогою ERP, TMS та WMS систем представляють композитну систему менеджменту високорівневою бізнес–логікою елеваторного та суміжних комплексів. Кожна з систем має свою базу даних, що забезпечує повну або часткову інформатизацію процесу виробництва.

Інтегроване програмне забезпечення в наведених системах забезпечує автоматичну роботу підсистем для реалізації функцій з динамічного перетворення та компонування збережених даних протягом кожної ітерації бізнес–процесу. Користувачі можуть отримати доступ до інформації, яку вони потребують під час виконання робочих обов'язків через інтерфейс користувача. ERP, WMS та TMS зосереджені на координації процеси всередині організації, компонентна логіка взаємодії наведена на рисунку 1.7. Їх бази даних зберігають внутрішні дані, програми забезпечують високорівневу логіку для внутрішньої координації через спеціалізовані інтерфейси, що відповідають за представлення даних у стандартному для галузі та регіону форматі.

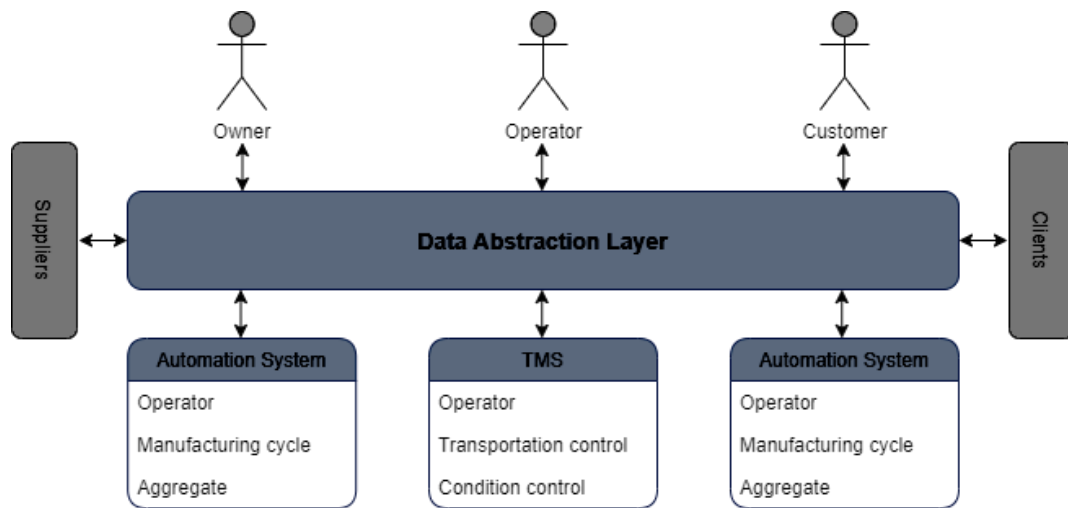


Рисунок 1.7 Компонентна логіка суміжних систем

Програмні компоненти передбачають забезпечення для управління повним циклом виробництва у організаційній мережі, основного функціоналу IAS, ERP, WMS та TMS (включаючи інтерфейси EDI) недостатньо. Додатково необхідно організувати системи управління розподіленими процесами, які найчастіше існують на глобальному, міжорганізаційному рівні. Тому в ланцюгу поставок архітектура управління має бути представлена додатковими до перелічених програмними компонентами.

1.3.3 Формування вимог до моніторингової системи

Програмні компоненти в архітектурі називають ланцюговими двигунами виробничого процесу. Головною метою організації такої структури системи є забезпечення можливості модифікації даних на віддалених робочих станціях з різних організацій. Разом компоненти програмного забезпечення складають системний рівень (SCE), присвячений управлінню міжорганізаційними процесами. Програмні компоненти в архітектурі можна класифікувати на три категорії, диференційовані до рівня управління ланцюгами поставок підтримується двигунами:

1. Сервіс комунікації функціонально забезпечує базовий зв'язок між системами (та користувачами) у ланцюзі постачання;

2. Інформаційні сервіси функціонально забезпечують стандартну інформація про системи (та користувачів) у ланцюзі постачання;
3. Сервіс керування функціонально забезпечує вдосконалене управління всіма системами у ланцюзі постачання.

На комунікаційному рівні, сервіс опрацювання повідомлень, підтримує обмін повідомлень між системами, що використовують або SMTP (електронна пошта), FTP (передача файлів) та HTTP (веб-протокол). На кожному підключеному до системи комп'ютері встановлений сервіс обміну повідомленнями. Потім локальні системи можуть обмінюватися повідомленнями через локальні сервіси, які упаковують, відправляють, передають, отримують та розпаковують повідомлення.

На інформаційному рівні – сервіси можуть подавати дані про запаси від різних локальних систем. Зовнішні користувачі можуть задати їх інформаційний запит через веб-браузер. Сервіси отримують запит з локальних систем та інтегрують інформацію в базу даних. Даний огляд може відображатися зовнішньому користувачеві або може бути імпортований в іншу систему в ланцюг поставок.

Рівень управління включає сервіси для вдосконалення управління ланцюгом поставок. Вони мають логічні правила для повністю автоматичного прийняття рішень і напівавтоматичної підтримки прийняття рішень. Нижче буде описано розробку та застосування сервісу контролю товару в архітектурі програмних компонентів для управління ланцюгами поставок.

Згальна даталогічна модель системи наведена на рисунку 1.8.

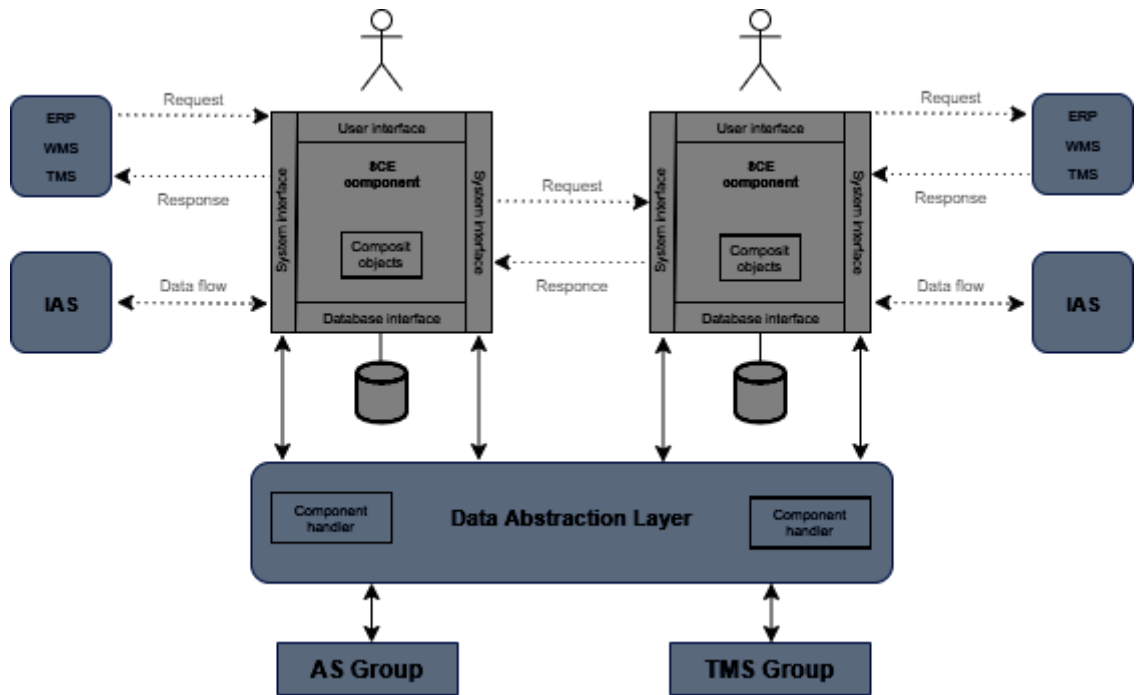


Рисунок 1.8 Даталогічна модель системи моніторингу

Сервіс управління продукцією. Механізм управління продукцією включає в себе інтегроване управління запасами в мережевих організаціях. Ці програмні компоненти разом з існуючими системами в ланцюгу поставок представляють архітектуру управління продукцією. Сервіс управління продукцією (ІМЕ) – надзвичайно елементарна система, управління рівнем запасів лише однієї точки запасу SKU. Програмні компоненти слабо пов'язані з один одним в мережах, щоб підтримувати інтеграл управління запасами в динамічних ланцюгах поставок. Локальні системи управління (ERP, TMS або WMS) можуть використовуватися ІМЕ для отримання та встановлення інформації про запаси продукції. ІМЕ можна розподілити між кількома локаціями та організаціями в ланцюгу поставок.

За допомогою системних змінних та системних рівнянь в мережі ІМЕ, інтеграл управління запасами може підтримуватися відповідно до одного з цих алгоритмів. Для підтримки мережевих організацій деякі додаткові системні змінні та рівняння включені до ІМЕ. Програмні компоненти надають гнучкість конфігурації, гнучкість синхронізації та гнучкість алгоритму. Гнучкість конфігурації – це здатність динамічно поєднувати та

роз'єднувати ІМЕ мережі. Ці властивості дозволяють надати автономність мережевих організацій з урахуванням місцезнаходження, часу та типу управління.

Класи об'єктів в ІМЕ оснащені атрибутами та операціями, які дозволяють їм виконувати задачі відповідно своїх цілей. Об'єкти працюють разом, надсилаючи повідомлення, що представляють запит на послугу та відповідь на цей запит. Клієнти та постачальники в межах середовища знову можуть бути оснащені з ІМЕ, але вони можуть також використовувати іншу інформаційну систему. Якщо замовник або постачальник також використовує ІМЕ, базові системні асоціації конфігуруються з точки зору замовника або постачальника.

Мережевий сервіс управління запасами можна реалізувати за допомогою застосування ІМЕ для кожного окремого пункту запасу SKU. Програмні компоненти утворюють системний рівень для управління ланцюгами поставок, абстрагований з існуючих систем місцевого управління.

1.4 Концепція управління виробничим процесом

Існує багато методів управління виробничими процесами, які розроблені безпосередньо для промислового виробництва. Такі стандарти, як ISO 9000 вимагають періодичного проведення аудиту компаній на відповідність вимогам.

Стандарти ISO 9000 включають систему вимоги, вимоги управління, вимоги до ресурсів, реалізація вимог та виправлення вимог. Сертифікація на цей стандарт також входить до сфери управління якості.

Основними концепціями управління виробництвом, описаними в літературі і застосовуються в даний час в практиці роботи виробничих підприємств, є Lean Production, Six Sigma, кайдзен, QRM – Quick Response Manufacturing, TOC – Theory of Constraints, Agile Manufacturing. Найбільш популярна в даний час концепція Lean Production. Lean Production – це підхід,

спрямований на постійне вдосконалення виробничого процесу і усунення дій, які не створюють цінності продукту (з точки зору споживача) і, відповідно, є «втратами».

У Lean Production виділяється сім категорій прихованих втрат:

1. Перевиробництва (виготовлення продукції не в точній відповідності з тим, що потрібно клієнту);
2. Через помилки, що допускаються при виробництві продукції (брак, дефекти), і необхідності переробки виробів;
3. При зайвому переміщенні (непотрібні переміщення персоналу та обладнання);
4. При транспортуванні;
5. Від зайвих запасів (запаси сировини, матеріалів, готової продукції);
6. Від зайвої обробки виробу в процесі виробництва (включення в технологічний процес виготовлення);
7. Втрати часу на очікування (люди, обладнання, продукція очікують початку виконання дій).

Lean Production базується на кайдзен – філософії «постійного, безперервного вдосконалення». Поступовість і безперервність – важливі умови будь-яких перетворень. Такий підхід дозволить спростити навчання нового персоналу і, найголовніше, буде сприяти виявленню непотрібних дій при виконанні тієї чи іншої операції. Таким чином, основними принципами Lean Production є: робота в команді, комунікація, ефективне використання ресурсів і скорочення витрат, а також постійне вдосконалення.

1.4.1 Локальні зони контролю

Проектування та розміщення зон контролю локалізованого типу реалізує інформатизацію життєвого циклу виробничої сировини та кінцевого продукту при проходженні циклу виробництва. З моменту надходження довільної сировини до будь-якого об'єкта системи, розпочинається

автоматичний аналіз та підготовка інформаційних пакетів для подальшого представлення операторам локальних систем.

Процес супроводу продукції є ізольованим процесом, при чому на деяких етапах оператори системи позбавлені інструментів впливу на кондиційні параметри і не можуть впливати на біологічні процеси, які протікають у виробничих і транспортних локалях. Незважаючи на це, статистичні детерміновані у часі дані про вже здійснений вплив довколишніх чинників на об'єм продукції залишаються важливим фактором для прийняття рішень, щодо подальшого впливу на продукцію з боку систем контролю біологічних параметрів.

Лабораторний аналіз продукції потребує загального уявлення про вплив сторонніх та внутрішніх чинників на якість продукта. Первинний аналіз отриманої сировини здійснюється лабораторно на обмеженій вибірці із загального об'єму. У випадку, коли існує задокументована інформація про фактори зовнішнього впливу, ступінь якості аналізу, а також його параметричні характеристики можуть зрости. Така модель дозволяє зменшити супутні витрати, час на опрацювання вибірки та зменшити вплив погрешностей дослідження на всіх етапах життєвого циклу продукції.

Пропонується організувати підсистему транзакції даних, що поєднуватиме локальні зони контролю у протяжний, неперервний процес моніторингу. Для реалізації такого підходу необхідно розмістити комплекси та набори первинних вимірювальних перетворювачів.

1.4.2 Устаткування локальних зон контролю

Промислова система управління (ICS) – це збірний термін, що використовується для опису різних типів систем управління та супутніх приладів, які включають пристрої, системи, мережі та засоби управління, що використовуються для роботи та автоматизації промислових процесів. Залежно від галузі, кожна ICS функціонує по-різному і побудована для ефективного електронного управління завданнями. Сьогодні пристрої та

протоколи, що використовуються в ICS, використовуються майже у всіх промислових секторах та критично важливих інфраструктурах, таких як виробнича, транспортна, енергетична та водоочисна промисловість.

Змінні операційної технології (ОТ) включають апаратні та програмні системи, які відстежують і контролюють фізичні пристрої на місцях. Завдання ОТ різняться залежно від галузі. Датчики, які відстежують температурні показники, є наглядним прикладом пристроїв операційних технологій.

Зближення ІТ та ОТ забезпечує підприємствам більшу інтеграцію та видимість ланцюга постачання, що включає їх важливі активи, логістику, плани та операційні процеси. Архітектура типової системи наведена на рисунку 1.9.

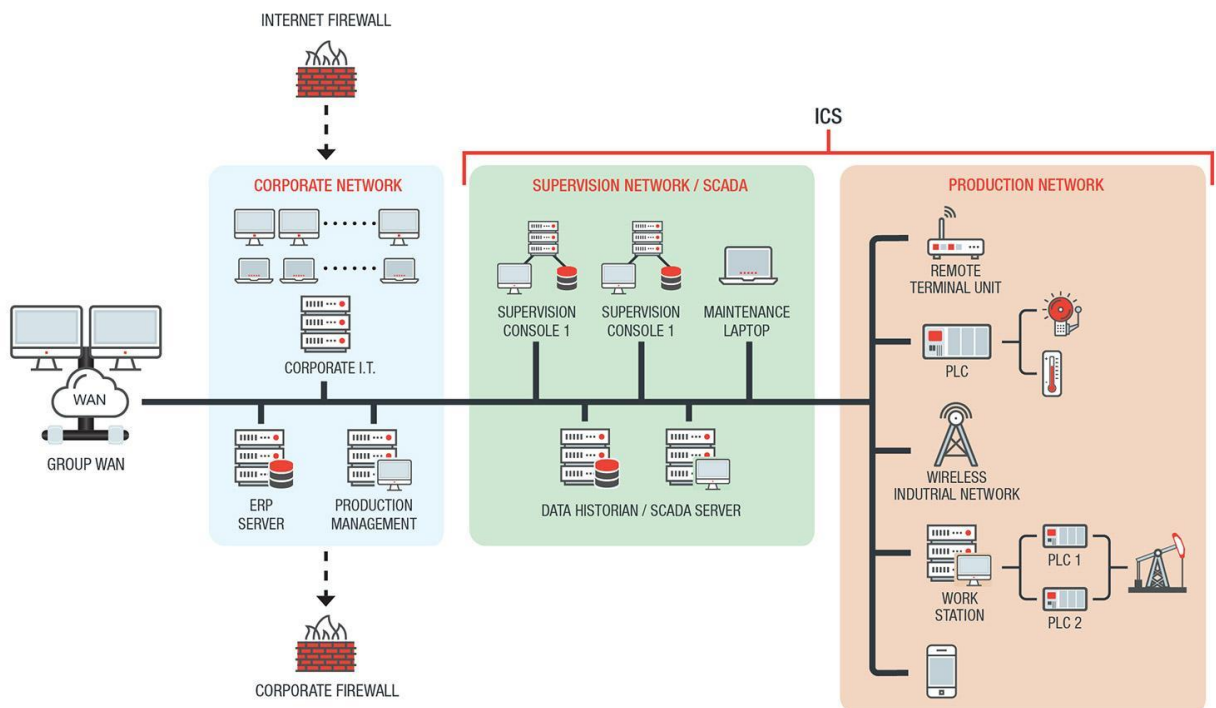


Рисунок 1.9 Типова архітектура ICS системи

Програмований логічний контролер (PLC). Це тип обладнання, що використовується як в системах DCS, так і в системах SCADA як компонент управління загальною системою. Він також забезпечує локальне управління

процесами, що запускаються за допомогою пристроїв управління зворотним зв'язком, таких як датчики та виконавчі механізми.

У SCADA PLC надає ту саму функціональність, що й віддалені модулі терміналів (RTU). У DCS PLC використовуються локальні контролери в рамках наглядової схеми управління. ПЛК також реалізовані як основні компоненти в менших конфігураціях системи управління.

Віддалений термінал. RTU (remote terminal unit, віддалений термінал) – це польовий пристрій, керований мікропроцесором, який приймає команди і надсилає інформацію назад до MTU (Maximum Transmission Unit).

Контур управління. Кожен контур управління складається з апаратних засобів, таких як PLC та виконавчі механізми. Контур управління інтерпретує сигнали від датчиків, регулюючих клапанів, вимикачів, двигунів та інших подібних пристроїв. Потім змінні, виміряні цими датчиками, передаються контролеру для виконання завдання та / або завершення процесу.

Людино–машинний інтерфейс (НМІ). Додаток графічного інтерфейсу користувача (GUI), що дозволяє взаємодіяти між оператором та обладнанням контролера. Він також може відображати інформацію про стан та історичні дані, зібрані пристроями в середовищі ICS.

Дистанційна діагностика та обслуговування. Термін використовується для позначення процедур з виявлення, запобігання та відновлення після операцій або збоїв. Сервер управління розміщує програмне забезпечення для нагляду DCS або PLC і взаємодіє з пристроями управління нижчого рівня.

Сервер SCADA або головний термінал (MTU) це пристрій, який видає команди RTU на місцях. Функціональне навантаження MTU наведено на рисунку 1.10.

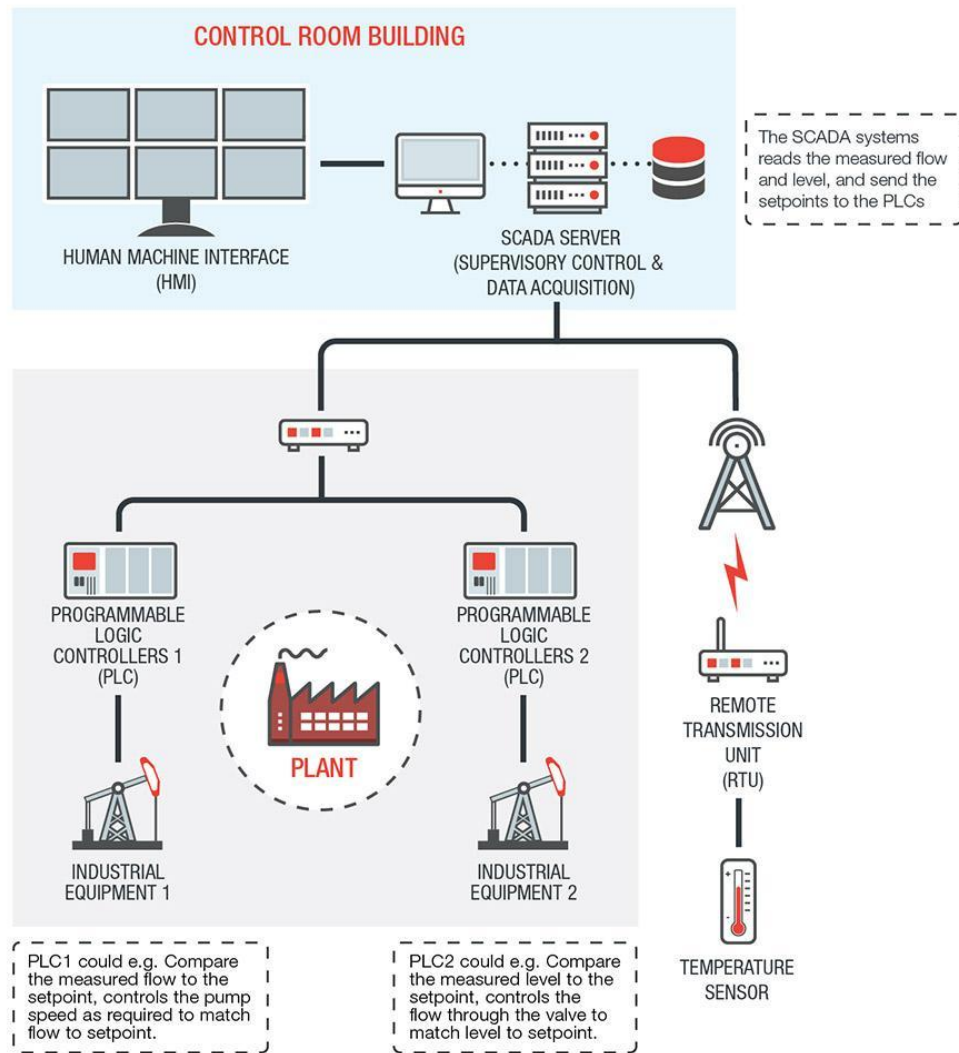


Рисунок 1.10 Топологія зв'язків MTU SCADA

Пристрої та модулі управління. В системах ICS передають інформацію через протоколи зв'язку. Існує кілька протоколів зв'язку, що використовуються через різні середовища ICS. Більшість із цих протоколів розроблено для конкретних цілей, таких як автоматизація процесів, автоматизація будівель, автоматизація енергосистем та багато іншого. Ці протоколи також були розроблені для забезпечення взаємодії між різними виробниками. Однак існують деякі протоколи, які працюють лише в тому випадку, якщо протоколи та обладнання надходять від одного виробника. Використаний протокол в даній системі це PROFIBUS.

Система контролю температури. Складається з програмно-апаратної частини, що включає в себе датчики температури і програмної логіки

управління обладнанням. Відсутність контролю за температурою вузлів і механізмів обладнання, наявність пилу може призвести до пожежі на елеваторі, а відповідно до істотних збитків власника. Типові режими роботи СКТ наведні на рисунку 1.11.



Рисунок 1.11 Система контролю температури

Система контролю пилу. Складається з програмно–апаратної частини, що включає в себе безпосередно датчики детекції пилу в потоці та програмної логіки управління фільтром в залежності від запиленості.

Система інтелектуального технічного обслуговування. Веде годинник напрацювання устаткування і співставляє їх з календарем ТО після чого автоматично призначає дату обслуговування вузлів. Також веде архів усіх дій з обладнанням і не дає можливості обслуговуючому персоналу пропустити ТО, для чого зв'язується з Головним Інженером, у разі необхідності. Типові режими роботи систем ІТО наведні на рисунку 1.11.

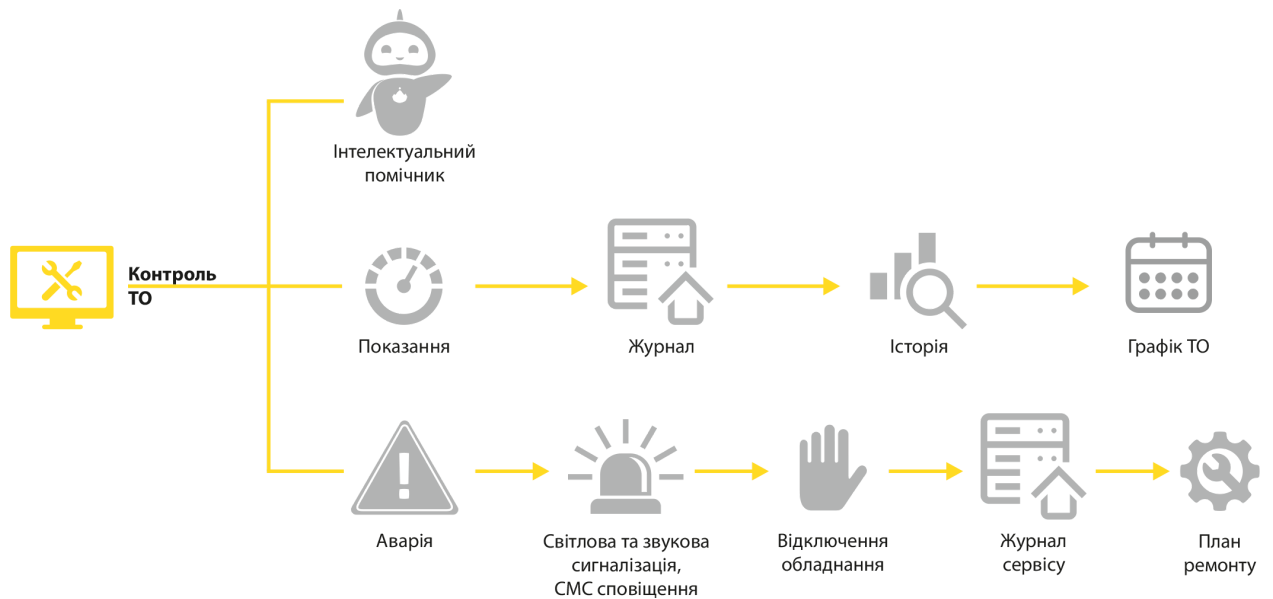


Рисунок 1.12 Система ІТО

Обов'язки з організації та ведення черги накладаються на конкретного співробітника. Це може бути реєстратор транспорту, який вже є на терміналі і зараз випусує паперові талони, охоронець на в'їзді або інший менеджер, уповноважений керівництвом.

Створюється електронна черга з порядковим номером і орієнтовним часом заїзду автомобіля.

Менеджер по роботі з чергою створює необхідний план заїздів, виходячи з вимог зернового терміналу: часу обробки однієї машини, кращою культурою, яку потрібно прийняти. Вся попередня робота по внесенню первинних даних в систему обліку елеватора, вже зроблена під час запису машини в чергу або при реєстрації у менеджера черги на майданчику відстою. Час обслуговування кожної машини безпосередньо на елеваторі скорочується, що призводить до збільшення загальної кількості машин, оброблених за добу.

Основні завдання, які повинні вирішувати модуль організації та ведення електронної черги.

1. Організація і управління електронною чергою в'їзду на зерновий термінал;

2. Введення і перевірка попередніх даних, необхідних для обслуговування транспорту безпосередньо на елеваторі. Набір конкретних даних, які можна ввести в АСУ залежить від вимог конкретного елеватора. Він може бути від мінімальної інформації про машину, водія і поклажодавця до введення всіх даних з ТТН, привезеної водієм;
3. Контроль виїзду машин з території елеватора. У ньому на виїзді машини повинно відзначатися час. Цілком можливо, що робитиметься це може і автоматично за допомогою камер, що зчитують номер;
4. Формування запиту керівництва в разі будь-яких позаштатних ситуацій і отримання від нього узгодженої відповіді;
5. Формування інформації про поточний стан черги і планованих на розвантаження культур в розрізі найменувань і тоннажу для менеджерів елеватора;
6. Збір і накопичення інформації в електронному вигляді про реальний потік транспорту для подальшого аналізу.

Робота з електронною чергою зовсім не скасовує вільного приїзду машини з зерном, черговість обслуговування якої буде вирішувати менеджерами терміналу вже на місці.

1.4.3 Система інтегрованого моніторингу

Зростаючі вимоги до надійності процесів, реагування на вимоги клієнтів та досягнення стійкого виробничого середовища вимагає застосування більш досконалого та комплексного моніторингу системи. На практиці інтегровані системи моніторингу найчастіше застосовуються лише в повністю автоматизованих системах. Такі системи мають дуже обмежену функціональність, орієнтовану на практичну реалізацію примітивного функціоналу та загальне керування набором кінцевих станів робочого агрегату чи комплексу агрегатів. У більшості складних виробничих систем

інформаційний супровід обмежується лише звітністю про хід виробництва без можливості моніторингу станів процесів та окремих агрегатів. В даний час більшість досліджень з вдосконалення моніторингу зосереджені на окремих рішеннях, які в результаті утворюють одну складну систему. Швидкий розвиток галузі інформаційних систем та спеціальні рішення, такі як хмарні обчислення, відкривають можливості для практичної реалізації моніторингової інтеграції в розподілені Інтернет–системи. Інтегровані вдосконалені системи моніторингу повинні підтримувати управління виробництвом та ланцюгами поставок, а також оптимізацію виробничих процесів. При наявності у виробничому процесі дискретних етапів, що можуть чинити вплив на загальний стан продукту або об'єкта виробництва, інтеграція функцій контролю в одну систему є проблемною галуззю. Дані, отримані датчиками та спеціалізованими системами, повинні бути вчасно оброблені спеціальними адаптивними алгоритмами. Вдосконалена обробка вхідних даних з набору підсистем необхідна для отримання інформації про стан суперсистеми. Застосування технології CPS для моніторингу обробних процесів, верстатів та виготовлених деталей дозволить ефективно отримувати інформацію. Для ефективного використання інформації з майбутніх систем моніторингу знадобляться високодосконалі IT–системи: гнучкі, ефективні, відкриті для реконфігурації та здатні працювати з великим обсягом даних у наборі систем виробничого середовища.

При проведенні досліджень з можливості розробки системи моніторингу, було визначено необхідність проектування апаратно–програмного комплексу зі створення методів усунення проблем пов'язаних з технологічним процесом, людським фактором і моніторингу працездатності систем об'єктів відстеження.

РОЗДІЛ 2

РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ

2.1 Розробка проектних рішень АС

В даному розділі було реалізовано та описано проектні рішення з розробки АС.

Загальну модель системи з урахуванням рівнів взаємодії, виокремлення функціональних зон та представлення компонентів системи наведено через представлення мережі інформаційного простору АС та зображено на рисунку 2.1. Детальний розбір даної моделі описано в розділі 2.3.

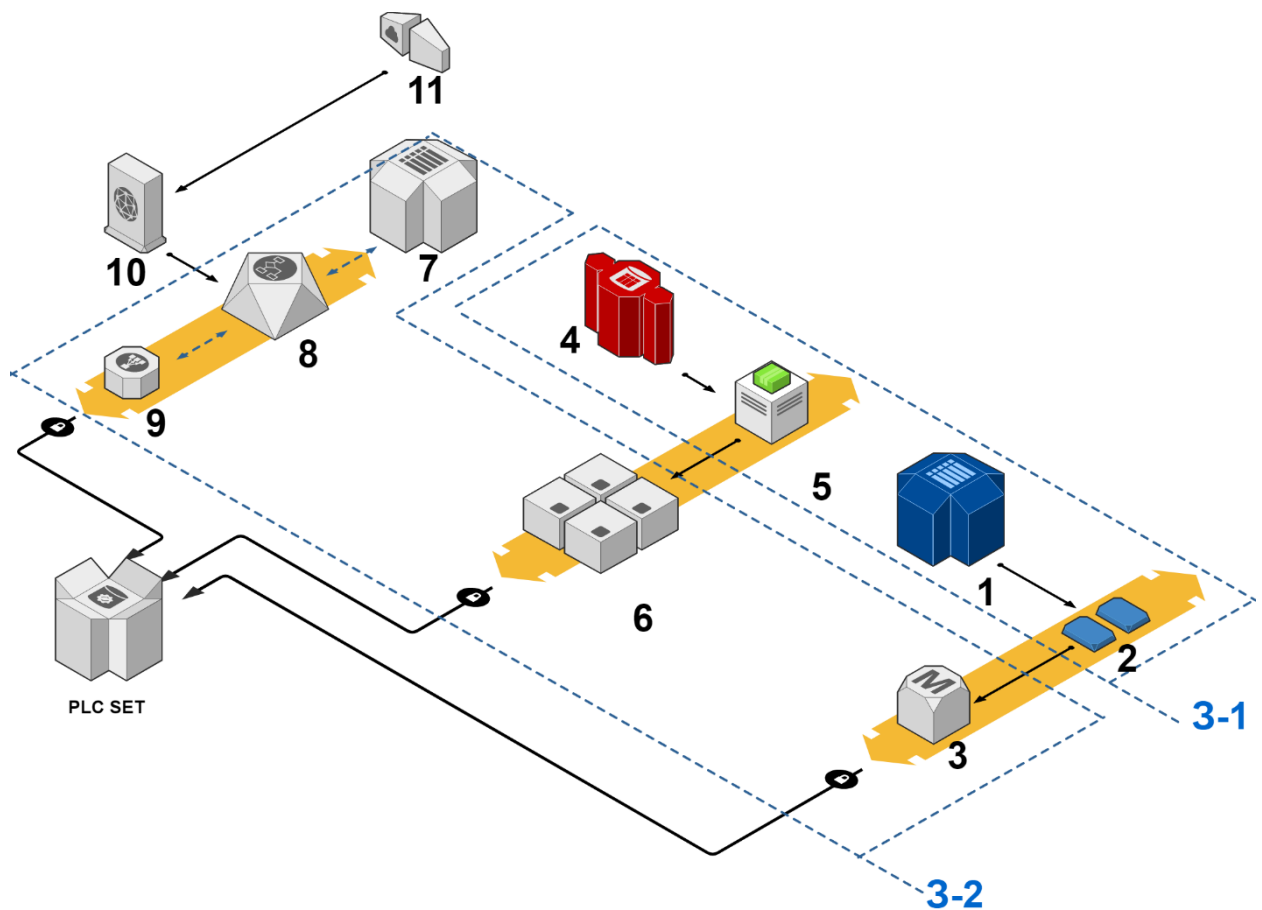


Рисунок 2.1 Мережа інформаційного простору «IAS Management»

2.1.1 Побудова моделі системи

Інфологічне проектування являє собою побудову інформаційно–логічної моделі (ІЛМ) предметної області, в якій виключена надмірність даних і відображені інформаційні особливості об'єкту управління.

При автоматизації об'єкт підлягає інформатизації, в результаті якої буде побудована ІЛМ. Елементи системи мають скінчену множину станів, а його сутності можна зв'язати логічно, що є необхідною і достатньою умовою для побудови моделі скінченого автомату. Наявність даної моделі визначає інформаційний простір та можливість його доповнення, що є суттєвою перевагою у відношенні до стандартних систем лінійної логіки і забезпечує модульність моделі. Такий підхід спрощує підтримку та полегшує розробку для інженерів на наступних ітераціях. На ознайомлення нового фахівця з системою буде витрачено менше часу.

Чим більше параметрів комп'ютер може контролювати в автоматичному режимі, тим вище ступінь захищеності системи в цілому.

ІЛМ включає набір правил заданих користувачем, що можуть бути доповнені у ході розвитку системи.

Система матиме змогу виявляти та запобігати розвитку аварійно–небезпечних процесів. А на глобальних рівнях контролю КС зможе автоматично формувати сигнали, залежно від загального статусу об'єкту, що забезпечить можливість створення інтерфейсу НМІ та диспетчеризувати керування системою.

Модель системи включає параметри, наведені в таблиці 2.1.

Таблиця 2.1 ІЛМ–набори систем

Елемент ІЛМ	Набір	Опис
Механізм	DQ(Відповідно до таблиці 1.6)	Датчики

	DI(Відповідно до таблиці 1.6)		Контролери
Змінна	Bool	–	Булева змінна
	Integer	XX–int	Цілочисельна змінна
	Timer	T–XX	Часова структура
	Counter	–	Лічильник
	Word	–	VCD–Слово
Теговий простір	XX–YY		Кодова прив'язка Об'єкт–Операція

2.1.2 Технологічна схема

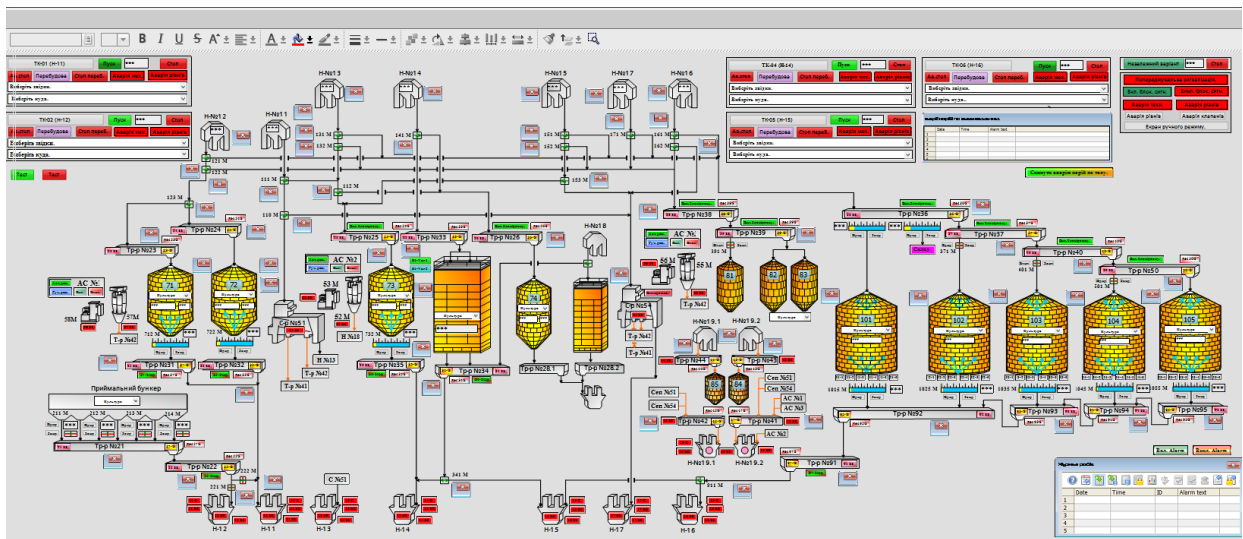


Рисунок 2.2 Представлення технологічної схеми

Технологічна схема – це графічне модельне уявлення технологічного процесу у вигляді послідовних виробничих функцій, технологічних і транспортних операцій, спрямованих на отримання товарної продукції, а також позначення логічних зв'язків між ними [8].

Технологічна схема, яка була реалізована в розділі 2.2 схематично ознайомлює зі всіма компонентами технологічного процесу системи.

Структура приведена у технологічній схемі реалізує достатню кількість заданих маршрутів для роботи системи оминаючи певні агрегати, якщо кондиція зернової продукції відповідає нормам зберігання. Це було створено для того, щоб організувати в системі потрібну кількість оперативних зон для збереження ресурсу механізмів, обходу етапів первинної чи вторинної обробки та організації оптимальної циркуляції зернової продукції по елеватору [9].

Надає представлення про зв'язок елементів, приналежність компонентів до технологічних каналів або маршрутів. Спрощує подальшу модернізацію об'єкта та є важливою частиною технічної документації всієї системи.

Дана технологічна схема не дає представлення про внутрішню організацію механізмів, не визначає точні параметри з'єднань та не зберігає масштаб.

Креслення одного з елементів системи(норія) наведено на схемі частково, оскільки повне його перенесення креслення у масштабі буде перекривати інші компоненти та технологічні канали системи.

Представлення компонентів технологічної схеми наведено в таблиці 2.2 і включає: номер згідно таблиці кодування 1.4, вимоги до кодування, задіяна кількість у системі та прив'язка компонентів.

Таблиця 2.2 Кількісне представлення елементів системи та вимоги до їх кодування

№	Компонент	Кодування компоненту	Кількість у системі	Прив'язка до компоненту
1	Силос	301/302/303	3	–

2	Транспортер	20/21/22/31/32/41	6	Послідовна прив'язка елементів
3	Норія	11/13/15/17	4	–
4	Перекидний клапан	111/211/151/411	4	Прив'язка до кодування механізму згідно рисунку XX
5	Засувка рейкова	222/221/223/224	4	Прив'язка до кодування механізму згідно рисунку XX
6	Сепаратор	–	1	–
7	Сушарка	–	1	–
8	Самоплив	–	32	–

2.1.3 Розробка підсистеми введення–виведення

Схему кодування вихідних сигналів з датчиків елементів системи та їх опис наведено в таблиці 2.2. Код кожного сигналу складається з ідентифікатора обладнання та єдиного коду функції. Це означає, що номер кодування відповідає певній функції контролера чи датчика. Однотипні сигнали з різних пристроїв будуть відрізнятися тільки ідентифікатором обладнання.

Таблиця 2.3 Кодування вихідних сигналів елементів системи

Елементи системи	Кодування	Опис
Силос	–	–
Транспортер	XX–15	Контроль захисту мех–ма
	XX–31	Контроль роботи мех–ма
	XX–51	Контроль датчика підпору

	XX-61	Контроль датчика швидкості
Норія	XX-15	Контроль захисту мех-ма
	XX-31	Контроль роботи мех-ма
	XX-51	Контроль датчика підпору
	XX-61	Контроль датчика швидкості
Перекидний клапан	XX-31	Контроль захисту мех-ма
	XX-(41, 43)	Контроль положення мех-ма(ліво, право)
Засувка рейкова автоматична	XX-31	Контроль захисту мех-ма
	XX-(41, 43)	Контроль положення мех-ма(відкрито, закрито)
Сепаратор	XX-15	Контроль захисту мех-ма
	XX-31	Контроль роботи мех-ма
Сушарка	XX-15	Контроль захисту мех-ма
	XX-31	Контроль роботи мех-ма

Впровадження CiR. Аббревіатура CiR розшифровується як "Configuration in RUN" (англ. "Конфігурація під час виконання"). Цей термін використовується для позначення способу, що застосовується для зміни конфігурації системи під час її штатної роботи.

Деякі установки не повинні виключатися під час виробничого процесу. Це може бути пов'язано, наприклад, зі складністю автоматизованого процесу або з високою вартістю пускових та передпускових робіт. Необхідність переконфігурування технологічної установки з видаленням або заміною обладнання не повинно потребувати зміни режиму роботи системи або повної чи часткової заміни компонента. Відключення або підключення електронного обладнання до комп'ютеризованої системи під час її роботи без виключення живлення і зупинки системи, заміни або перепідключення блоку в цілому носить назву

“Hot Swap”. Така зміна конфігурації апаратного забезпечення, з використанням технології CiR, дозволяє виконати певні зміни в конфігурації установки у час, коли технологічний процес знаходиться в активній фазі, а система в режимі виконання (Режим виконання в літературі може позначатися як RUN–режим, runtime–режим або RT–режим). “Hot Swap” зміна конфігурації обладнання перериває процес на досить короткий проміжок часу. При цьому за замовчуванням в якості верхнього граничного значення такого періоду часу встановлюється значення, припустиме для конкретного технологічного процесу. Під час короткого переривання, вихідні значення сигналів управління обладнанням зберігають свої останні значення.

Можливість зовнішнього внесення змін в будь–який розділ програмного та апаратно–технічного забезпечення без зупинки технологічного процесу забезпечується за рахунок Технології CiR, яка в рамках даного проекту дозволяє:

- Додавати нові чи видаляти існуючі станції розподіленого вводу–виводу і прилади рівня 3–1(Згідно діаграми 2.1), що виконують функції ведених пристроїв;
- Скасовувати введені конфігурації;
- Виконувати перенастроювання модулів станції ET 200M.

2.1.4 Розробка підсистеми контролю та обчислень

Обслуговування вихідних та генерація вхідних (керуючих) сигналів здійснюється сімейством контролерів реального часу Siemens S7, моделі S7–1500, що повністю задовольняють потребам проекту. Оброблені дані з підсистеми введення–виведення чинять вплив на інформаційний простір системи у реальному часі. Повна інформація про поточний стан системи отримується підсистемами обчислень та аварійного контролю. Підсистеми реалізовані на ПЛК проводять обчислення по заданим правилам ґрунтуючись на множині станів системи. Збір даних здійснюється за допомогою

розробленою строго типізованої системи введення–виведення. Обмін сигналами відбувається у єдиному просторі кодування наведеного в таблицях 2.3 та 2.4. Наявна інформація конвертується у зрозумілий для оператора вигляд та може бути передана до централізованого пункту керування системою згідно рисунку 1.4.

Реалізація обміну сигналів. Повинна здійснюватися за рахунок універсальних інтерфейсних модулів та загальної мережі PROFINET, що являє собою відкритий промисловий стандарт для автоматизації від асоціації PROFINET INTERNATIONAL(PI). Інтерфейсним модулем було обрано ET 200MP, який є універсальною модульною системою введення–виведення, що є стандартним та підключається безпосередньо до контролера S7–1500. SIMATIC ET 200MP вирізняється найкоротшим часом циклів внутрішньої шини, що значно зменшує час потрібний системі на опрацювання послідовності сигналів. При роботі з великими обсягами даних система, побудована на модулях ET 200MP може бути масштабована горизонтально, що реалізується за допомогою паралельного підключення модулів. Також інтерфейсний модуль підключає станцію SIMATIC ET 200MP до мережі PROFINET і підтримує обмін даними між контролером PROFINET IO і модулями станції, за протоколом TCP/IP.

Міжрівнева конвертація керуючих сигналів. Кожний рівень має власну виконавчу мову, що забезпечує ефективне використання обчислювальних потужностей, але створює проблему при необхідності обміну інформації всередині системи. Часто перехід від одного пристрою до іншого супроводжується процесом конвертації сигналів. На нижчих примітивних рівнях виникає потреба захисту керуючого пристрою від електромагнітних перешкод та шумів обладнання. Загальна структура обміну сигналів в системі «IAS Management» розподілена на два рівні за складністю сигналів.

Перший рівень представлений зоною “1” (З–1) згідно рисунку 2.1 реалізує простий обмін даними між компонентами 1, 2, 3 та 4, 5, 6.

Другий та третій рівень представлений зоною “2” (З–2) згідно рисунку 2.1 реалізує обмін даними за протоколом TCP/IP та конвертує дані для більш примітивних рівнів. Не менш важливою функцією даного рівня є захист локального інформаційного простору від втручання ззовні.

Захист. Не дивлячись на те, що З–1 працює у власному командному просторі та слабо піддається атакам, в З–2 зберігається системна інформація, яка підлягає захисту, тому не можна нехтувати стандартними системами протидії.

Система поділена таким чином, що ПЛК не відноситься до жодної з зон, тому що він виконує роль супервайзера системи та є проміжним шаром між примітивними елементами системи та системою диспетчерського керування. Для взаємодії між виконавчими механізмами та ПЛК реалізовано модульні контролери через які відбувається зв'язок між ними. Функціональна роль підсистеми збору даних (ПЗД) полягає у забезпеченні безпечного змiну станів системи. Перед внесенням змін в інформаційний простір системи виконавча команда з ПЛК проходить перевірку ПЗД. У разі відповідності команди заданим правилам ПЗД проводить транзакцію та повідомляє ПЛК про новий стан системи.

Опис компонентів рисунку 2.1 представлено в таблиці 2.5. Нумерація згідно рисунку 2.1.

Таблиця 2.5 Опис компонентів ІЛМ

Рівні	№	Компоненти	Опис
З–1	1	Виконавчий механізм	Електрообладнання
	2	Вимірювальний елемент	Датчики
	3	Модульні частини контролера	Проміжна ланка між ПЛК та механізмами

	4	Система збору даних	Управління інформацією
	5	Система керування	Диспетчер прикладних служб системи
	6	Серверна частина	Відповідає за обмін сигналами між ПЛК та системою збору даних
3–2	7	Локальна система зберігання даних	Система авторизації та ведення протоколу роботи системи
	8	Диспетчеризація керування	Пункт керування системою
	9	Диспетчеризація збору	Підготовка та представлення інформації

За існуючої архітектури системи має обмежений кінцевий набір станів (всього чотири). Керувати зміною станів системи може сім категорій сигналів різного рівня. Діаграма станів системи та керуючих сигналів приведено на рисунку 2.3.

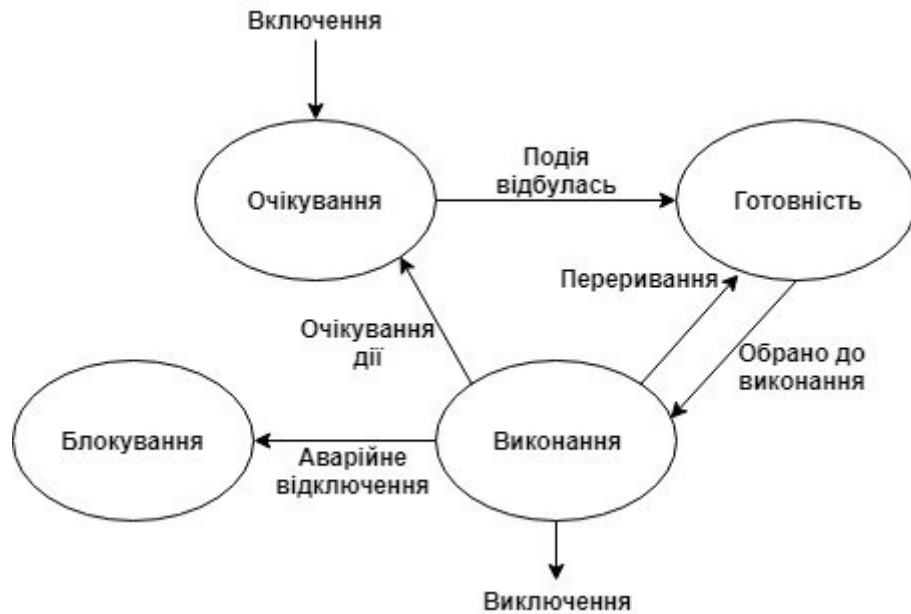


Рисунок 2.3 Зображення всіх станів системи

2.1.5 Розробка алгоритмічного представлення

В даному розділі описано алгоритмічні представлення процесів, що утворюють технологічний процес переробки зерна. Наведено приклади розробки та організації принципів схем, які реалізують логіку системи правил.

Представлення маршрутів системи. В цілях економії фінансових та апаратних ресурсів системи було вирішено розробити систему маршрутів.

Маршрут №1 Автоприйм – Оперативний силос було розроблено для оперативного прийому готової зернової продукції на зберігання в системі.



Рисунок 2.4 Алгоритмічне представлення маршруту №1

Маршрут №2 Автоприйм – Сепаратор – Оперативний силос було розроблено для прийому зернової продукції, що потребує первинної обробки, на зберігання в системі.



Рисунок 2.5 Алгоритмічне представлення маршруту №2

Маршрут №3 Оперативний силос – Сушарка було розроблено для термічної обробки зернової продукції, яка зберігається в системі.



Рисунок 2.6 Алгоритмічне представлення маршруту №3

Маршрут №4 Сушарка – Сушарка було розроблено для забезпечення термічної циркуляції зернової продукції.

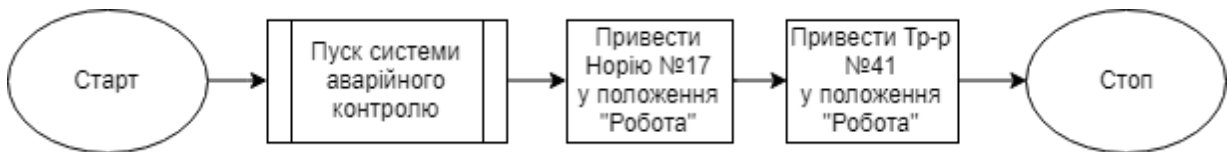


Рисунок 2.7 Алгоритмічне представлення маршруту №4

Маршрут №5 Сушарка – Силос – Сушарка було розроблено для забезпечення можливості току зернової продукції, яка знаходиться на зберіганні в оперативній зоні.

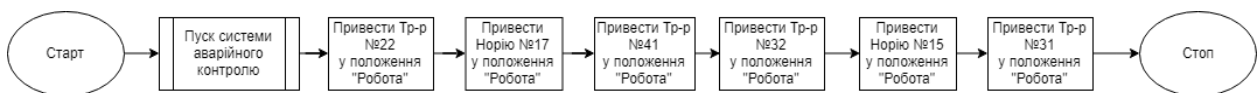


Рисунок 2.8 Алгоритмічне представлення маршруту №5

Маршрут №6 Сушарка – Зберігання було розроблено для оперативної термічної обробки перед вивантаженням зернової продукції або переведенням її на довгострокове зберігання.



Рисунок 2.9 Алгоритмічне представлення маршруту №6

Маршрут №7 Автоприйм – Силос було розроблено для довгострокового зберігання готової зернової продукції в системі.

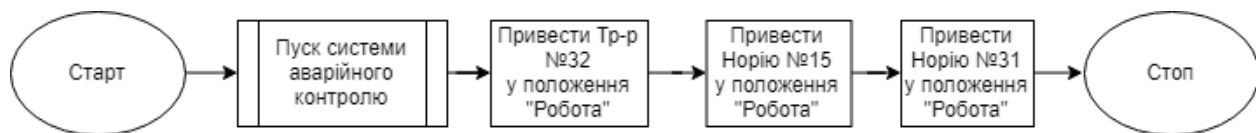


Рисунок 2.10 Алгоритмічне представлення маршруту №7

Представлення логічно–принципових схем. Елементи системи правил розроблені у вигляді принципової схеми, яка описує процес обробки вхідних сигналів. Оскільки апаратне устаткування технологічної зони 1 (відповідно рисунку 2.1) оперує на рівні введення–виведення, будь–яке правило може бути сформовано за допомогою логічних операцій. Загальна кількість правил та операцій нараховує 107 блоків. Детальний розгляд кожного з них в рамках роботи не представляє інтересу. Для загального ознайомлення з процесом розробки блоків системи обчислень, детально розібрано принципову схему наведено на рисунку 2.9.

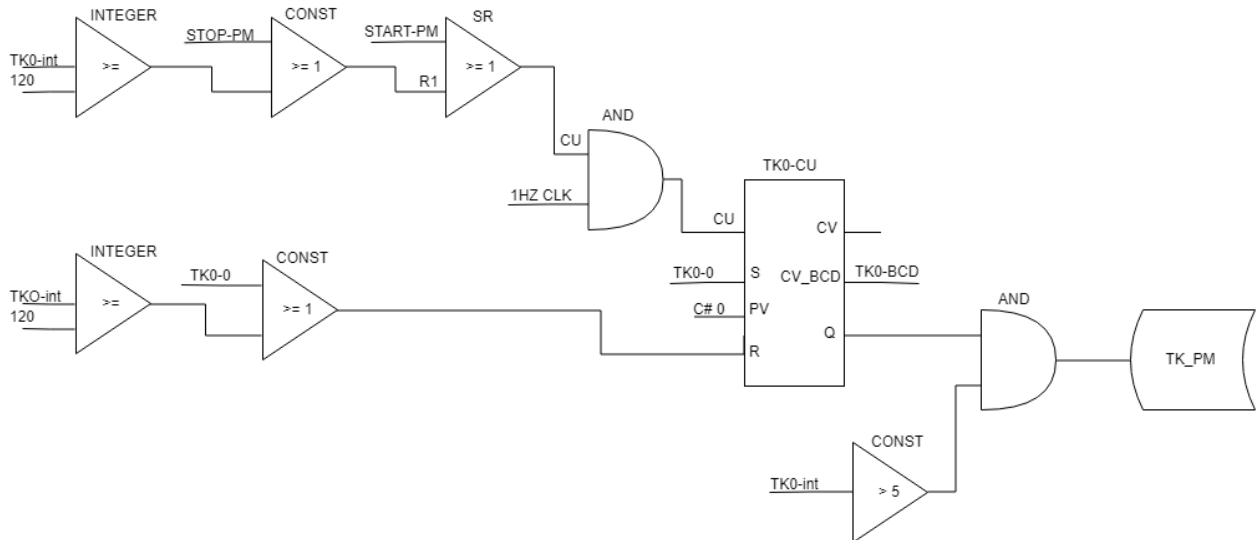


Рисунок 2.11 Принципова схема запуску аварійної сигналізації

Включення попередньої сигналізації сповіщає робочий штат про запуск системи елеватора та готовність компонентів системи. Дана система зумовлює використання двох таймерів. П'ятнадцятисекундна затримка зумовлена звуковим сигналюванням запуску системи. Керування системою неможливе до закінчення сигналізації. Після чого система стане сприйнятлива до змін на встановлений регламентом проміжок часу.

Реалізація логіки даної системи представлено на рисунку 2.9. За допомогою компараторів (позначені трикутниками), кон'юнкторів (позначені півколами) та блоку сортування (позначений прямокутником) була побудована принципова схема, в результаті проходження сигналів якою обчислювальний елемент може керувати системою попередньої сигналізації. Наявність у схемі кон'юнкторів суворо обмежує функціонал системи у випадках порушень регламентів часу.

Для виконання міжрівневої конвертації згідно пункту 2.3 необхідно передбачити перетворювач, який згідно обраного кодеку виконуватиме конвертацію наданих даних. Передбаченні логічні виводи такого блоку наведено на Рисунку 2.10.

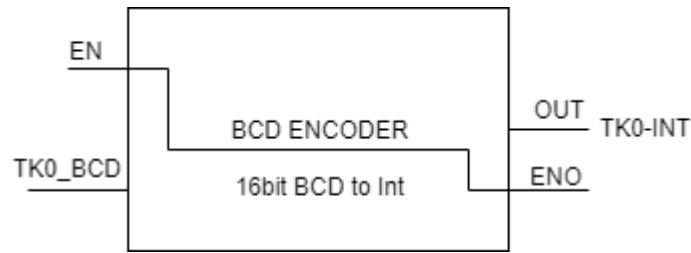


Рисунок 2.12 Принципова схема підключення перетворювача типу «слово—число»

Блок передбачає отримання даних для конвертації, поточного кодеку, та цільового кодеку.

2.1.6 Розробка підсистеми диспетчеризації

Система диспетчеризації є комплексом програмних і апаратних засобів, який дозволяє здійснювати віддалене управління інженерними системами одного або декількох об'єктів.

Автоматизована система диспетчерського управління (АСДУ) необхідна для контролю інженерного обладнання (№1, 3 згідно рисунку 2.1). Як правило, диспетчеризація включається в систему управління багатофункціональними об'єктами зі складною інженерною інфраструктурою якою є об'єкт даної розробки.

Проектування систем диспетчеризації полягає в вирішенні задачі візуалізації інформації про функціонування інженерних систем і надання оператору можливості прямого управління обладнанням з диспетчерського пункту (№8 згідно рисунку 2.1). Дані про стан інженерного обладнання надходять від контролерів локальної автоматики (№2 згідно рисунку 2.1) і передаються на сервер (№6 згідно рисунку 2.1). Оброблені технологічні дані з необхідною аналітичною інформацією надходять на сервер диспетчеризації (№9 згідно рисунку 2.1) і виводяться на екранах комп'ютерів на робочих місцях операторів в наочному динамічному графічному вигляді.

Дані, отримані і оброблені системою диспетчеризації, формуються в повідомлення різного виду, які архівуються в локальному довготривалому сховищі(№ 7 згідно рисунку 2.1). На основі цієї інформації, доступної в будь-який час, формуються звіти.

Система диспетчеризації надає ключові переваги при управлінні об'єктом:

- постійний централізований контроль роботи інженерних систем;
- оперативне реагування в аварійних ситуаціях;
- зменшення впливу людського фактора;
- оптимізація системи звітності.

Розроблювальна система забезпечує керування агрегатами одним оператором, який за рахунок інтерфейсу на комп'ютері буде ознайомлений зі всіма станами системи. АРМ оператора представляє собою людино-машинний інтерфейс SCADA системи WinCC на ПК, що дозволяє здійснювати контроль, управління технологічним процесом та обладнанням, своєчасно виявляти відхилення від нормальної роботи і попереджувати

виникнення аварійних ситуацій. На екрані АРМ оператора відображається візуалізація технологічного процесу, маршрутів руху продукту. Система облікових записів буде запобігати можливості управління агрегатами несанкціонованою людиною чи оператором, кваліфікаційний рівень якого, не дозволяє вести управління системою.

2.1.7 Розробка інтерфейсу оператора

На екрані АРМ оператора відображається візуалізація технологічного процесу, маршрутів руху продукту. Інтерфейс зроблений за стандартами зручності використання, придатності використання та ергономічності (ISO 25010).

При проектуванні системи були використані основні принципи UX/UI розробки:

- Принцип «Keep it short and simple» (від англ. Зберегти коротким і простим). Інтерфейс повинен бути простим і зрозумілим, завдання повинні вирішуватися мінімальним числом дій;
- Звичні елементи управління. У будь-якому сучасному інтерфейсі є багато елементів управління, буде краще використовувати звичні елементи і візуальні образи;
- Принцип «трьох кліків». Для виконання будь-якої базової операції оператору не повинно знадобитися більше трьох кліків. Будь-яка важлива інформація повинна бути доступна не більше ніж в три кліка;
- Однорідність. Елементи управління повинні групуватися на робочому екрані.

Для трактування станів агрегатів та системи в цілому використати візуалізацію та анімацію. Всі важливі компоненти системи повинні бути підписані згідно таблиці 1.4. Написи виконані в одному розмірі (16х, напівжирний) та чітко виділяються з фонового зображення.

Технологічні канали повинні бути зображені лініями з індикатором напрямку току зерна. Канали, які схематично перетинають один одного але не мають технологічного перетину, повинні помічатися особливим знаком, а у всіх інших випадках індикацію вважати надмірною і не використовувати.

Силосна група сховищ повинні бути зображені з використанням відносного масштабу. Залежно від ступеня наповненості об'єкт повинен змінювати колір.

Група перекидних клапанів повинні бути зображені з використанням одного масштабу. Індикація положення механізму залежить від обраного напрямку току зерна.

Група рейкових засувки повинні бути зображені з використанням одного масштабу. Індикація положення механізму залежить від обраної можливості току зерна.

Група транспортерів не мають вимог масштабування. Індикація механізму залежить від режиму його роботи.

Група норій повинні бути зображені з використанням одного масштабу. Як зазначено в пункті 2.5, для запобігання перекриття інших елементів системи, прийняти рішення про використання спрощеної моделі. Індикація механізму залежить від режиму його роботи.

Сушарка не має вимог масштабування. Індикація механізму залежить від режиму його роботи та від наповненості агрегата.

Сепаратор не має вимог масштабування. Індикація механізму залежить від режиму його роботи.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ МОДЕЛІ СИСТЕМИ

В даному розділі описано використане, при розробці даного проекту, програмне забезпечення. Наведено структуру реалізованих модулів, їх функціональне представлення та інструменти розробки. Визначено заходи з тестування та впровадження.

3.1 Програмне забезпечення

Працюючи на базі модельного ряду Siemens єдино можливим засобом розроблення прикладних функціональних модулів є інтегрована середа розробки TIA Portal. Являючись пропріетарним програмним забезпеченням корпоративного типу, вільний доступ до якого обмежено, для подальшої розробки системи було прийнято рішення про сумісну співпрацю з ТОВ «ЮГОВ–Проект». На базі якого мені було надане робоче місце з встановленою корпоративною ліцензією ПЗ 15.1 версії. Подальша розробка проводилась на базі зазначеного робочого місця.

Представлення компонентів, що утворюють інструментарій розробки наведено у таблиці 3.1 [10].

Таблиця 3.1 Опис використаного програмного забезпечення

№	Назва	Версія	Опис
1	Simatic Step 7	5.6 SP1	Програмне забезпечення призначене для розробки систем на базі програмованих логічних контролерів Simatic
2	Simatic WinCC	7.5	Програмне забезпечення призначене для створення людино–машинного інтерфейсу
3	Simatic PLCSIM	15.1	Система симуляції ПЛК, яка забезпечує ефективну підтримку при розробці та

			тестуванні програм і наступного фактичного застосування
--	--	--	---

В рамках угоди №11–45 від 03.10.2020 «Розробка проекту модернізації об'єкту “ЕВ–2”» між компанією та ФОП «Жук Ілля Ігорович» було прийняте рішення про проведення дослідження об'єкту « ЕВ–2» на предмет можливості продовження експлуатації та автоматизації технологічного процесу в ході якого заступник директора з виробництва Іванюта В.П сформував комплекс заходів з модернізації об'єкту.

Моєю сферою діяльності було проектування комплексу підсистем:

- Введення–виведення
- Контролю та обчислень
- Диспетчеризації

Комплекс заходів з розробки продукту наведено у таблиці 3.2

Таблиця 3.2 Етапи розробки проекту

Етап	По ча то к	За ве рш ен ня	Опис	Виконавець	Головний інженер ТОВ«ПУЕ ІК»
Постановка задачі	04. 10. 20 20	04. 10. 20 20	Визначення призначення системи	Жук І.І.	Сидоренко В.М.
Деталізація предметної області	05. 10.	08. 10.	Визначення базових вимог до системи	Жук І.І.	Сидоренко В.М.

	20 20	20 20			
Проектування ескізу ТЗ	10. 10. 20 20	15. 10. 20 20	Визначення функціоналу системи	Жук І.І.	Сидоренко В.М.
Проектування ескізу ТЗ	20. 10. 20 20	23. 10. 20 20	Визначення функціоналу «IAS Management»	Жук І.І.	Сидоренко В.М.
Узгодження функціоналу	26. 10. 20 20	26. 10. 20 20		Майстер з експлуатації Лісовий О.С.	Сидоренко В.М.
Внесення правок в документацію	26. 10. 20 20	30. 10. 20 20	Корегування ТЗ	Жук І.І..	Сидоренко В.М.
Інфологічне проективання та реалізація «IAS Management»	05. 11. 20 20	14. 11. 20 20	Розробка Структури «IAS Management»	Жук І.І.	Сидоренко В.М.
	16. 11. 20 20	06. 12. 20 20	Специфікація структурних елементів «IAS Management»	Жук І.І.	Сидоренко В.М.

	15. 12. 20 20	02. 01. 20 21	Деталізація сутностей	Жук І.І.	Сидоренко В.М.
	02. 01. 20 21	14. 01. 20 21	Логічна реструктуризація	Жук І.І.	Сидоренко В.М.
	29. 01. 20 12 0	— — — —	Надання продукту на розгляд комісії	Жук І.І.	Сидоренко В.М.

3.2 Технологічне представлення системи

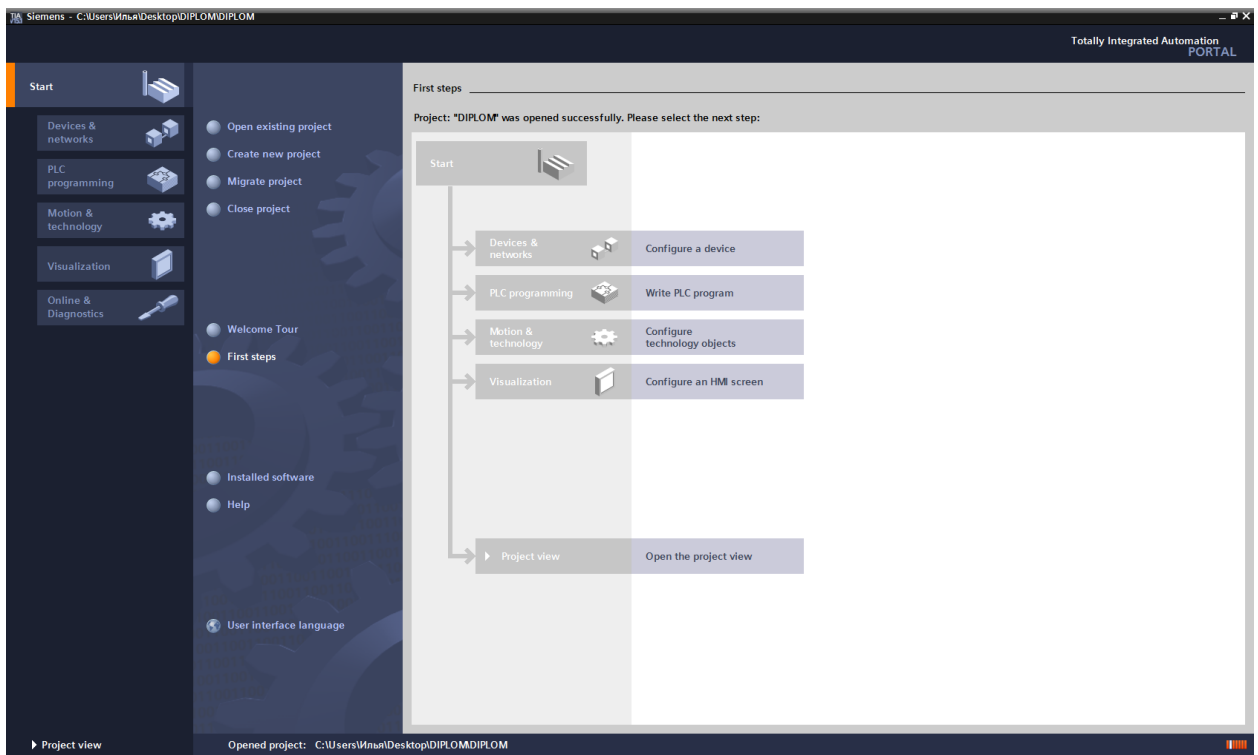


Рисунок 3.1 Інструментарій реалізації прикладної системи

Ознайомлення з інструментами розробки проектів автоматизації наведено в рисунку 3.1. Для початку розробки необхідно створити загальний проект майбутньої системи без деталізації структури та визначення компонентів. Користувачеві надаються рекомендації з створення інфраструктури проекту. Згідно функціоналу інтегрованого середовища розробки послідовність визначення інформаційного простору системи наступна:

1. Налаштування пристроїв та мереж обміну даними;
2. Програмування супервізорів підмереж;
3. Налаштування технологічних компонентів;
4. Візуалізація процесів виробництва;
5. Генерація проектного представлення;

Згідно покрокового визначення системного функціонала, розпочнемо налаштування пристроїв та профільних мереж обміну даними. Основними сутностями предмета розробки є системний супервізор та засоби керування системою. Для спрощення подальшої розробки та забезпечення зручності підтримки системи створимо об'єкти з назвами "PLC_1" та "PC station" (відповідно). Наявність нумератора у назві логічного контролера забезпечить можливість подальшого горизонтального масштабування системи зі збереженням контекстної назви обладнання. Такий підхід надає змогу додавати нові компоненти у систему та її документацію у етапній послідовності, що значено спрощує використання посилань у документації та внутрішнім контингентом спеціалістів. АРМ оператора, що включає засоби представлення інформації та інструменти керування системою несе єдину назву і не потребує додаткових специфікаторів, адже незалежно від прав доступа та фізичного знаходження системи кінцеві користувачі отримують єдину повну версію. Подальше керування правами відбувається локально і

залежить від централізованого сховища даних. Створені об'єкти наведено на Рисунку 3.2.

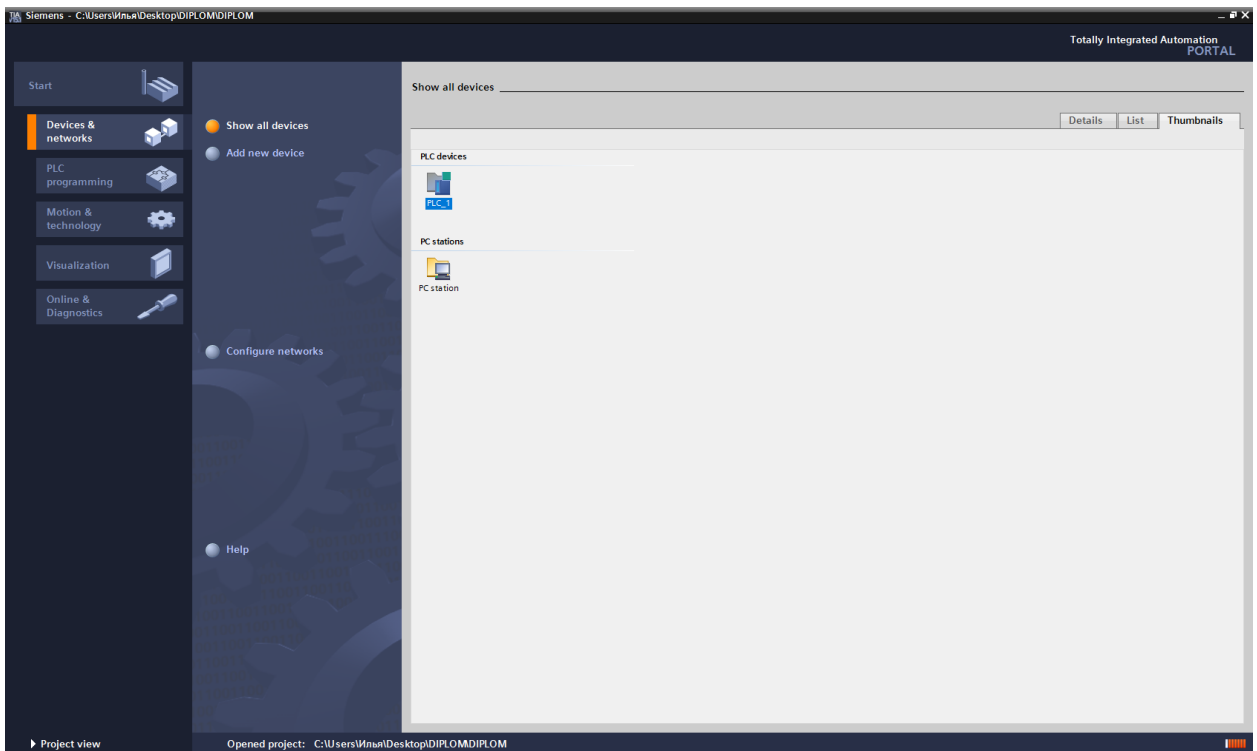


Рисунок 3.2 Етап створення мережі блоків програмування та задання структури ПЛК

Згенерувавши об'єкти система пропонує створити керуючу програму та набір необхідних функцій. Виокремлення таких модулів забезпечує можливість їх повторного використання, та дозволяє організувати систему посилянь, які під час компіляції отримують динамічний адресний простір, за яким менеджер посилянь отримає доступ до вмісту модуля. Такий підхід забезпечує виключення строгої специфікації адрес модулів, що дозволяє компілятору отримати останню версії розробленого модуля незалежно від роботи програм, що супроводжують процес вивантаження, фактично унеможливується помилка поєднання модулів та небезпечні операції з пам'яттю. Розмір блоків коду та даних обраховується перед вивантаженням проекту [11]. Наявні блоки даних та організаційних правил наведені на Рисунку 3.3.

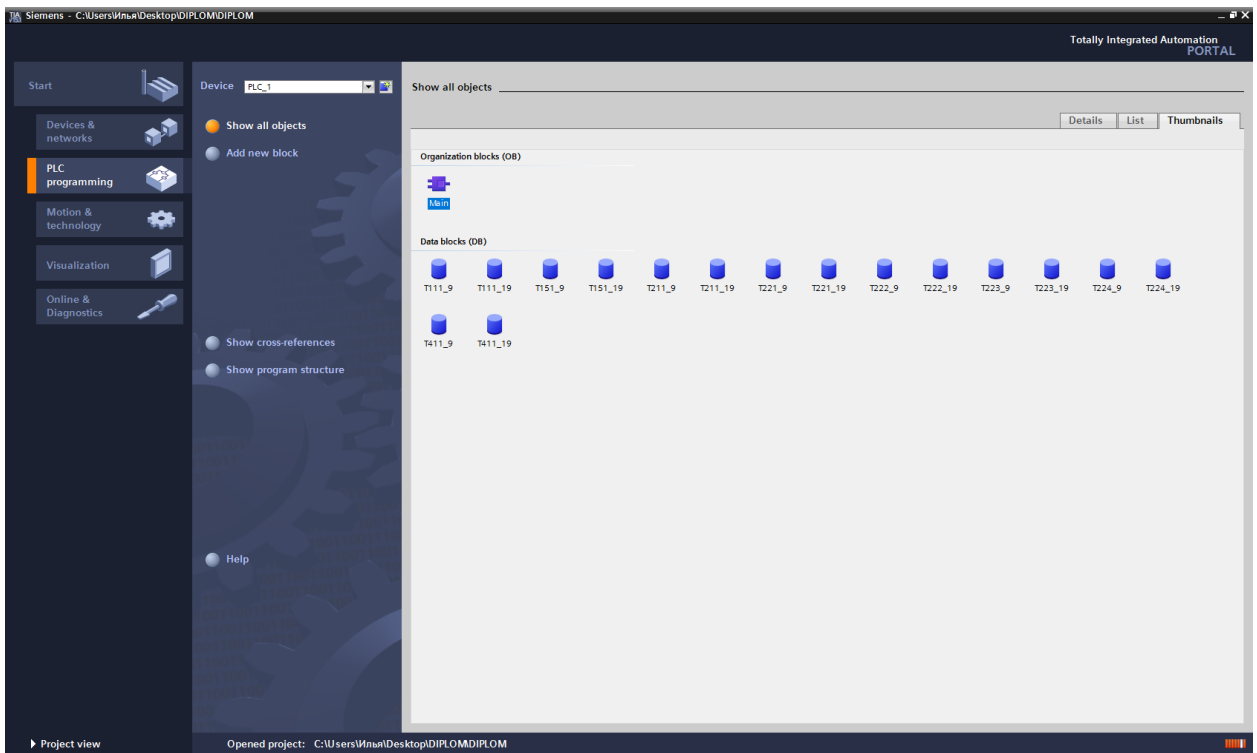


Рисунок 3.3 Етап програмування ПЛК

Впровадження систем інтегрованого контролю (далі СК) та безпеки виробництва не застосовується у зв'язку з використанням власних технологічних рішень товариством замовника. Проте даний інструмент надає змогу інтегрувати засоби контролю прав доступу, використовувати сумісне обладнання нагляду за персоналом та технологічним процесом. Багаторівнева автентифікація персоналу може бути втілена з використанням технологій NFC, що в поєднанні з класичними засобами підтвердження особи зменшить ризик несанкціонованого доступу до апаратного та програмного устаткування. Можливе розгортання систем з використанням колективного підтвердження операції, за яких для зміни стану систему потрібна група операторів, що має певний набір кваліфікаційних якостей.

Контроль позиціонування рухомих об'єктів з подальшим автоматизованим контролем входить до переліку інтегрованих засобів, проте потребує застосування сертифікованого апаратного устаткування. Засоби створення СК наведено на Рисунку 3.4.

кінцевих споживачів. Інструменти налаштування зазначених параметрів системи наведено на Рисунку 3.5.

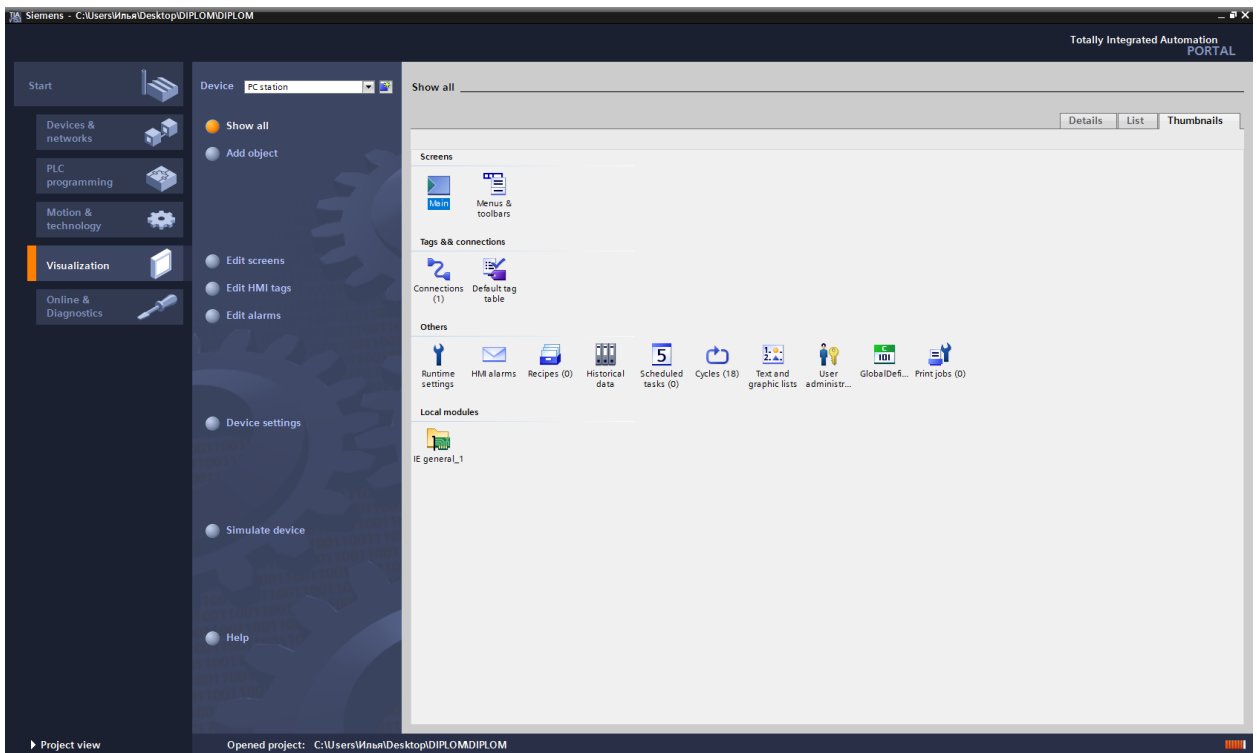


Рисунок 3.5 Етап інформатизації системи

Підтримка, контроль та модифікація системи виконуються у розділі «Діагностування та мережеві засоби керування». Предексплуатаційна діагностика системи, контроль параметрів обладнання та створення резервних конфігураційних записів реалізовано окремим модулем, що зумовлено скрізним застосуванням інструментарію на всіх етапах розвитку проекту.

Засоби віддаленого керування (далі ЗВК) дозволяють централізувати технічну підтримку, що значно спрощує процес позаштатної підтримки системи. Система з налаштованим ЗВК не потребує включення до контингенту спеціалістів інженерів підтримки та надає змогу звернення до сторонніх компаній підтримки. Зменшення витрат з експлуатації прикладної системи забезпечується скороченням штату елеватора та часових витрат на аварійне обслуговування системи.

Необхідно проводити тестування системи керування обладнанням у режимі симуляції перед внесенням змін у робочий порядок роботи елеватора. ТІА надає достатній та вичерпний інструмент для проведення таких заходів. Оскільки система пов'язана з механічним устаткуванням, вартість усунення недоліків ПЗ може виходити за рамки фінансування процесу розробки та підтримки. Інструменти здійснення мережевого моніторингу та керування наведено на Рисунку 3.6.

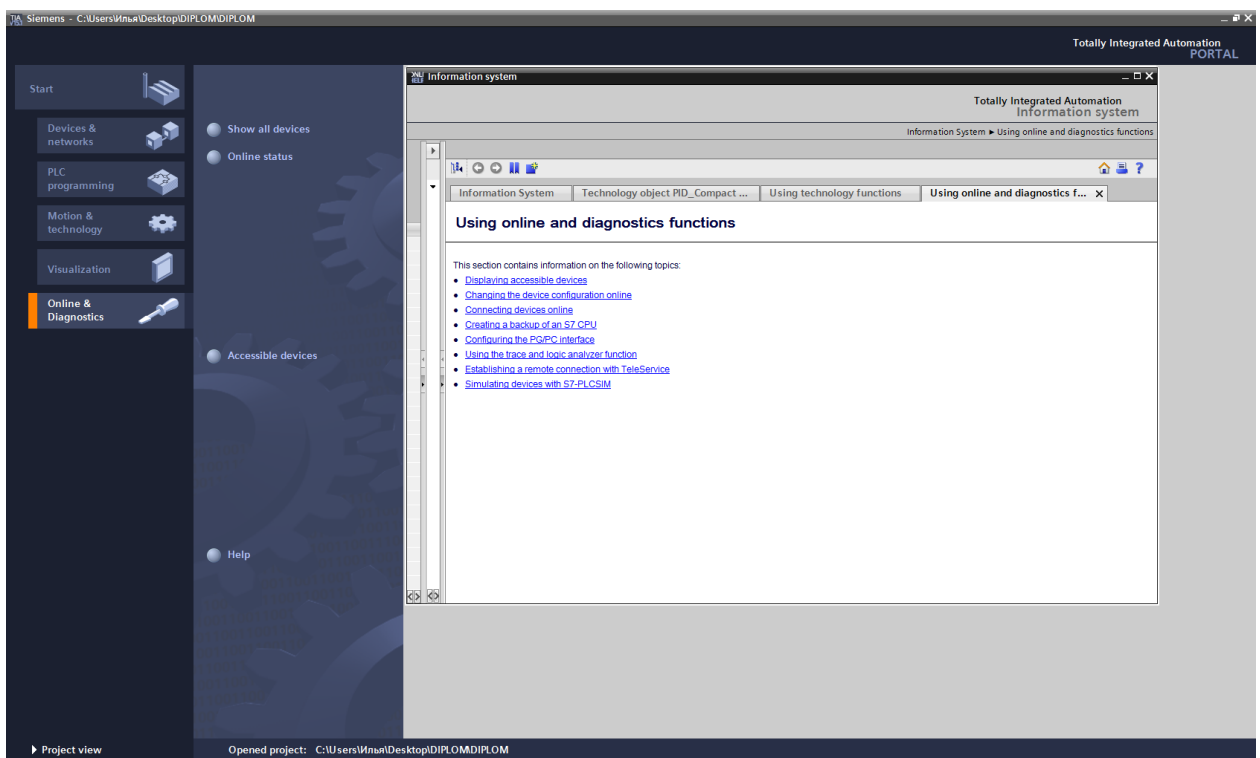


Рисунок 3.6 Інструменти моніторингу та мережевого керування системою

3.3 Компонентне представлення системи

Створивши технологічне представлення системи, користувач набуває можливості конкретизувати модель системи, зменшення рівня абстракції реалізовано модульно, до компіляції проекту система не потребує абсолютної прив'язки. Повідомлення про необхідність оголошення відсутніх елементів не заважають роботі користувача та не блокують подальші дії. Система передбачає автоматичну шаблону генерацію невизначених елементів.

Деталізація компонентів системи проводиться у наступному порядку:

1. ПЛК та модулі;
2. АРМ оператора;
3. Програмування блоків ПЛК;
4. Визначення функціональних блоків ПЛК;
5. Реалізація технологічної схеми об'єкту;
6. Створення тегового простору;
7. Налаштування мережевого моніторингу і тестування.

ПЛК та модулі. У першу чергу визначається набір використовуваного апаратного устаткування системи, оскільки дана модель не набуватиме змін у процесі подальшої деталізації. В рамках моделі визначається розташування та належність модулів контролера у технологічних шафах. Інтерфейси підключення та шини живлення також детермінують на цьому етапі. У роботі наведено реалізований склад технологічних шаф, що включають ПЛК та його модулі (Додаток Ж).

АРМ оператора. Необхідні для функціонування системи компоненти автоматизованого робочого місця оператора включають апаратні та програмні вимоги. Для коректного з'єднання з основною системою до складу АРМ повинно входити:

1. Персональний комп'ютер;
2. Інтерфейс підключення до виробничої мережі;
3. Програмний додаток WinCC RT Proof версії 7.5.

У роботі наведено реалізований склад АРМ оператора системи (Додаток И)

Визначення функціональних блоків ПЛК. Для програмування складних логічних послідовностей використано інструмент створення блоків даних. Визначивши вхідні та вихідні сигнали та встановивши правило їх взаємодії блок зберігається у інтегрованому середовищі розробки для

подальшого використання. У разі необхідності використати ідентичний блок, він може бути викликаний з пам'яті та автоматично вбудований у програмовану послідовність.

Програмування блоків ПЛК. Функціонування всієї системи залежить від програмованих блоків ПЛК. Програмуються усі елементи системи незалежно від їх прямого призначення [12]. Варіант блоку програмування згідно рисунку 2.11 наведено в роботі (Додаток К).

Реалізація технологічної схеми об'єкту. Представлення технологічної схеми з подальшою кодовою прив'язкою об'єктів до тегового простору надає змогу організувати інфографічне представлення системи агрегатів. Інтерфейс диспетчерського керування виконується за технологічною схемою. У разі потреби існуюча схема має бути доповнена з метою підвищення інформативності. Представлення технологічної схеми реалізовано в даному проекті (Додаток М).

Створення тегового простору. Для створення інформаційного простору складових компонентів системи створено теговий простір. Оголошення тегів устаткування та його компонентів аргументується необхідністю подальшої ідентифікації обладнання та організації автоматичного програмного керування. Теговий простір проекту включає 395 тегів, що повністю забезпечують інформатизацію системи. Приклад наведено у роботі (Додаток Н).

Налаштування мережевого моніторингу і тестування. Для проведення тестування програмних модулів та верифікації відповідності логічних схем було налаштовано симулятор системи S7-PLCSIM, який дозволяє редагувати значення змінних під час роботи симуляції. Приклад роботи запрограмованого симулятора наведено у роботі (Додаток П).

3.4 Інструментарий розробки пристрою моніторингу

На момент розробки платформу esp32 підтримують середовища програмування Arduino IDE та VScode зі встановленим плагіном «Arduino».

VSCode несумнівно має більший функціонал для створення та налагодження програмних додатків, проте плагін, забезпечуючий підтримку платформ та існуюче розходження версій драйверів пристроїв ускладнюють розробку проектів на базі VSCode. Вихідні файли, після створення проекту у VS являють собою звичайні .cpp файли, які не утворюють інтегрований проект, що ускладнює поширення проекту та його швидке розгортання третіми особами.

Не дивлячись на суттєву примітивність середовища Arduino IDE, розробка та підтримка проекту проводиться у ньому, адже такий підхід значно спростить розповсюдження програмного продукту серед користувачів та ентузіастів IoT. Адже після завершення проекту програмне забезпечення розповсюджуватиметься стандартним для Arduino IDE файлом з розширенням .ino та .zip архівованими бібліотеками, які встановлюються у середовище за допомогою вбудованого інструменту «Підключити бібліотеку».

Інсталяція ПЗ Arduino IDE (Integrated Development Environment, англ. Інтегроване Середовище Розробки). Назва констатує наявність необхідного комплексу засобів для повного супроводу будь-якого проекту на платформі Arduino. Середовище позбавлено функції автоматичного доповнення команд та інших зручностей. Максимальна простота забезпечує привабливість для новачків і звужує мінімальний набір знань для початку роботи у Arduino IDE.

Для цього необхідно виконати наступні дії:

1. Завантаження програмного забезпечення з офіційного веб-сайту Arduino;
2. Інсталяція середовища і перший запуск;
3. Налаштування відображення номерів рядків.

Для цього, необхідно перейти у меню (згори) за шляхом "Файл" → "Налаштування" і поставити галку біля відповідного рядка "Відобразити номери рядків". Зображення інтерфейсу налаштувань наведено на рис. 3.7.

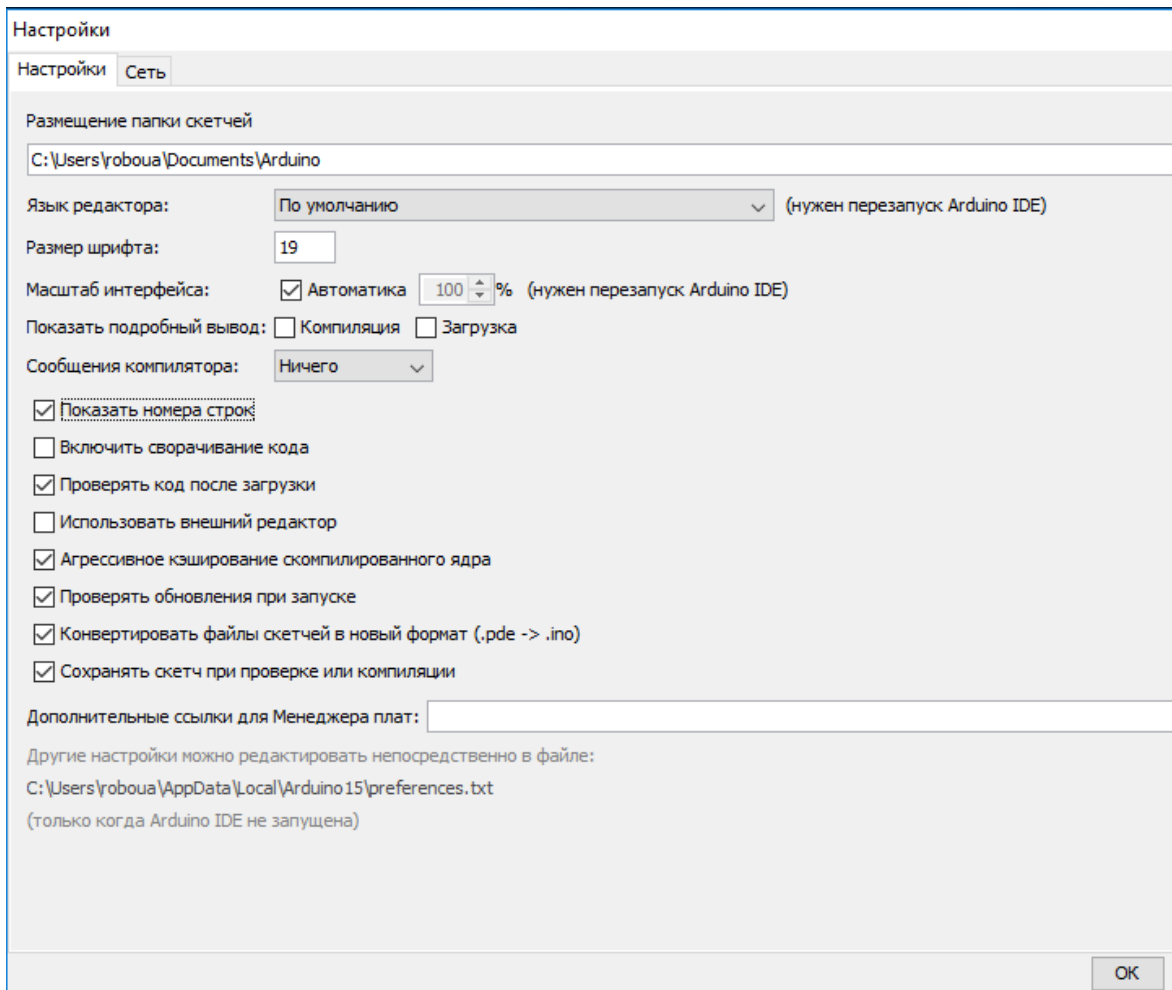


Рисунок 3.7 Інтерфейс налаштування середовища

Підключення плати до IDE. При приєднанні плати Arduino до комп'ютера, плата завантажує останній скетч або знаходиться у режимі очікування. Живлення подається USB-портом (англ. Universal Serial Bus – Універсальна послідовна шина) на плату, проте IDE не встановлює з'єднання з платою. Для цього використаємо меню у верхній частині вікна. Шляхом переходу "Інструменти"→"Порт" Користувач обирає порт з'єднання COM (англ. Communication Port – Порт обміну даними). Даний тип з'єднання передбачає послідовну передачу даних, що означає обмін бітовою інформацією. Інформація передається біт за бітом (на відміну від паралельного порту). У разі звичайної швидкодії групи – проведіть належні роз'яснення зараз, в протилежному випадку відкладіть їх до кінця уроку.

Зазвичай середовище відображає всього один варіант у вкладці "Порт" (у випадку, коли підключено всього один модуль Arduino). Після встановлення з'єднання середовище готове до обміну даними в обох напрямках.

Середовище здатне працювати з усім сімейством плат Arduino, проте особливості перекладу програм написаних користувачем на мову виконуючого пристрою вимагають вибору користувачем типу плати. Подивіться на наявну плату, на ній буде зазначено модель. Оберіть цю назву у розділі меню "Інструменти"→"Плата". Для роботи з платформою esp32 необхідно виконати налаштування, що включають:

1. Встановлення драйверу USB–UART.
2. Встановлення інструментарію розробника у середовище Arduino IDE.

Після встановлення USB–UART драйверу у диспетчері пристроїв з'явиться розпізнаний пристрій виробництва Silicon Labs, що означатиме коректне та вдале завершення підготовки конвертора до роботи (див. рис. 3.9).

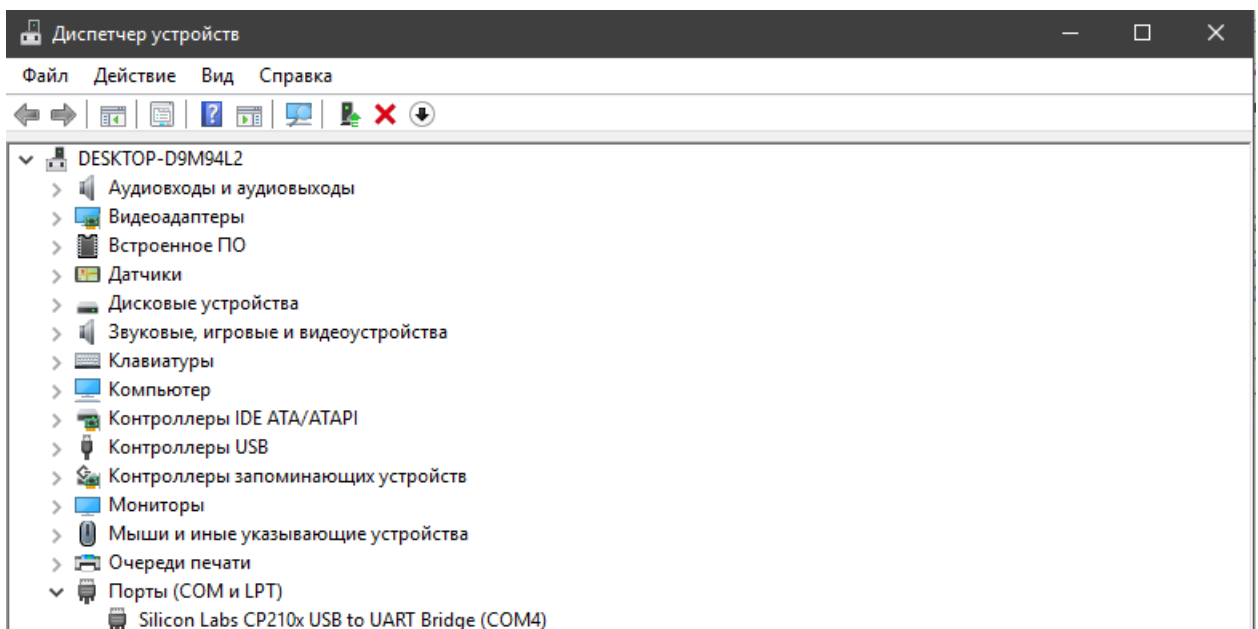


Рисунок 3.8 Розпізнаний пристрій у Диспетчері пристроїв

Надалі необхідно встановити інструментарій розробки для Arduino IDE на базі ESP32–Devboard. Розробники платформи підготували перелік правил і

інструментів для роботи з пристроєм, а розробники середовища надали можливість користувачу їх задіяти.

У меню середовища оберіть «Файл→Налаштування», перед вами з’явиться вікно налаштувань середовища, наведене на рис. 3.10. У параметр «Додаткові посилання для менеджера плат» впишіть посилання на офіційний набір засобів програмування та обміну даними з платформою ESP32–Devkit.

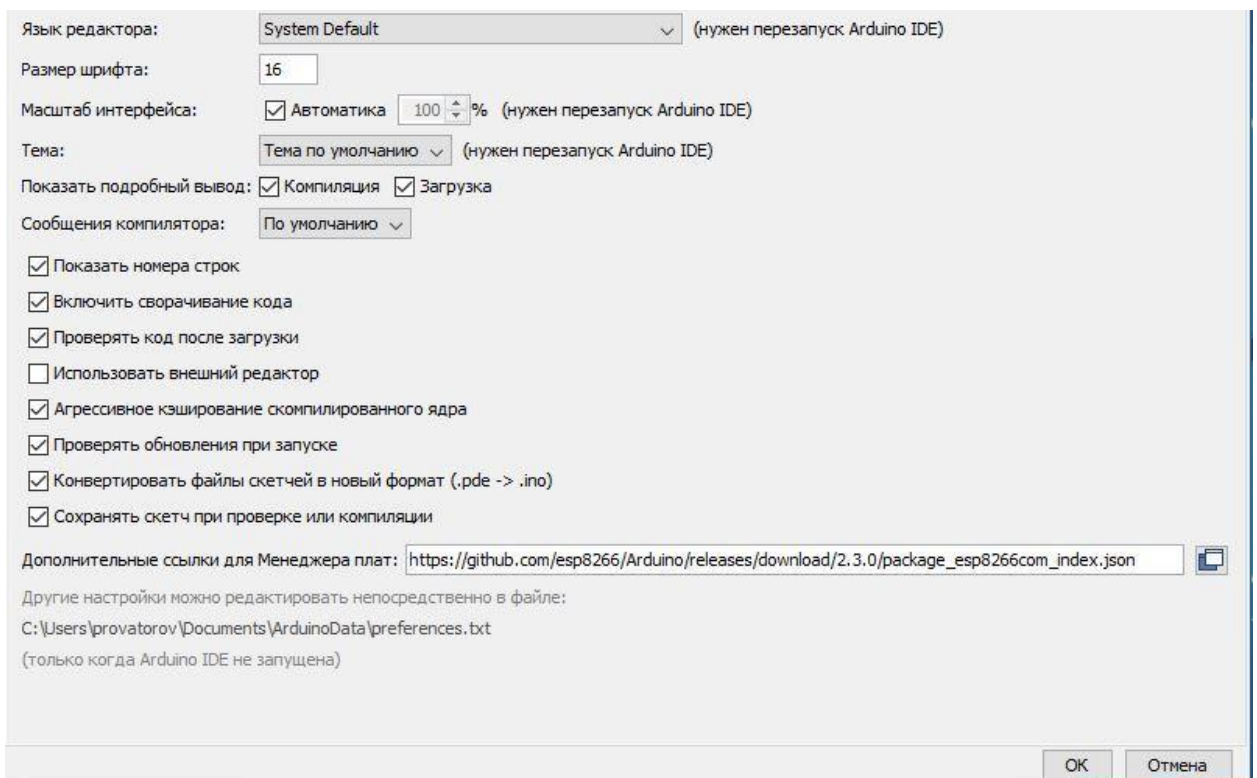


Рисунок 3.9 Вікно налаштувань середовища

Збережіть зміни і перезавантажте середовище, після чого перейдіть за шляхом «Інструменти–Плата–Менеджер Плат» у меню середовища. Відкриється вікно менеджеру плат, наведеного на рис. 3.11. Відразу по відкриттю розпочнеться завантаження переліку плат за посиланнями, доданими у конфігураційний файл.

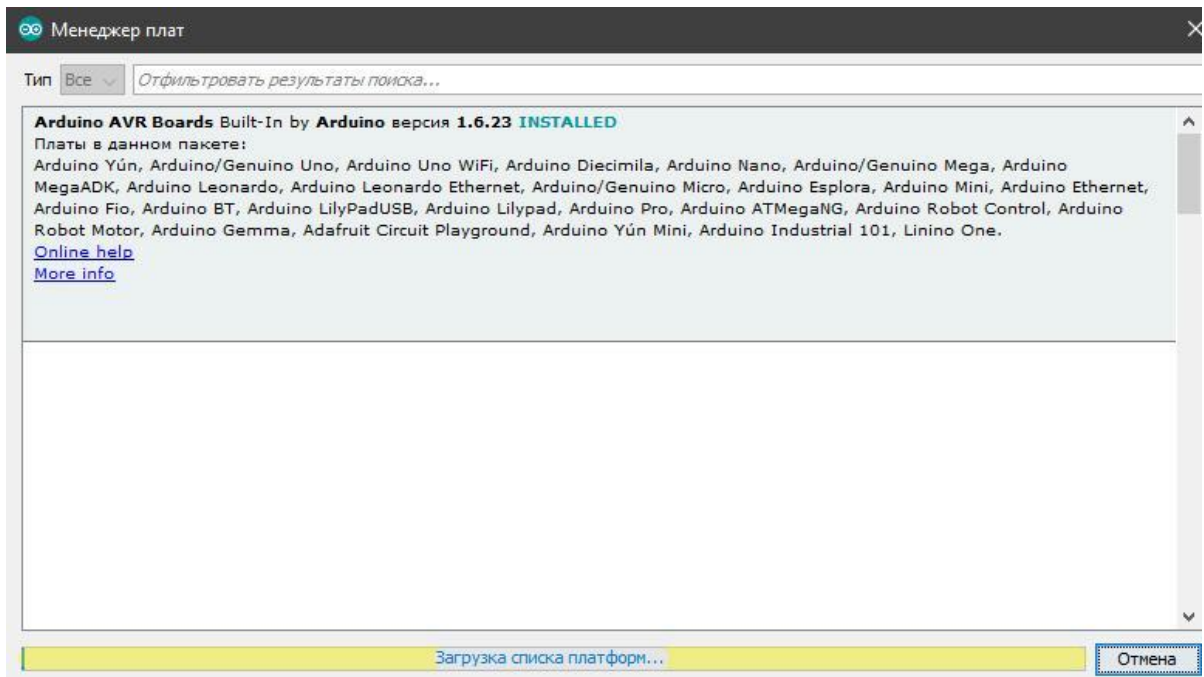


Рисунок 3.10 Завантаження переліку платформ у менеджері плат

У формі пошуку назва мікросхеми – esp32. По завершенню пошуку, буде обрано варіант «esp32 by ESP32 Community» за останньою доступною версією.

3.5 Побудова системи

В рамках роботи проводиться дослідження та розробка можливостей побудови інтегрованих систем автоматизації із залученням систем глобального позиціонування. Для демонстрації можливостей використання бездротових сенсорних мереж у сфері промислового охоронного обладнання вирішено побудувати систему, що надасть кінцевому користувачу можливість відстежування поточного положення рухомого об'єкту, на який встановлена система. Продукт також включатиме автоматизовану захисну систему керування силовими мережами та агрегатами. Всього передбачено до 6 керованих об'єктів, обмін даними з якими проходитиме на рівні сигналів керування або більш складних протоколів: SPI, I2C, UART.

Робоча напруга обчислювальної системи потребує наявності джерела живлення постійного струму з напругою 5 В. Допустиме відхилення вхідної напруги складає до 10 %. Пристрій забезпечуватиме живленням пристрій GPS-модуля з постійним рівнем напруги 3.3 В.

Для керування силовими мережами або агрегатами система включає перетворювач логічних рівнів, на який можливо встановити до 6 однаканальних пристроїв типу реле, або до 3х пристроїв з двоканальним трактом керування. В рамках роботи використовується механічне двоканальне реле з максимальним допустимим струмом 10 А. Схема електричного кола розроблена для реалізації продукту наведена на рис. 3.12.

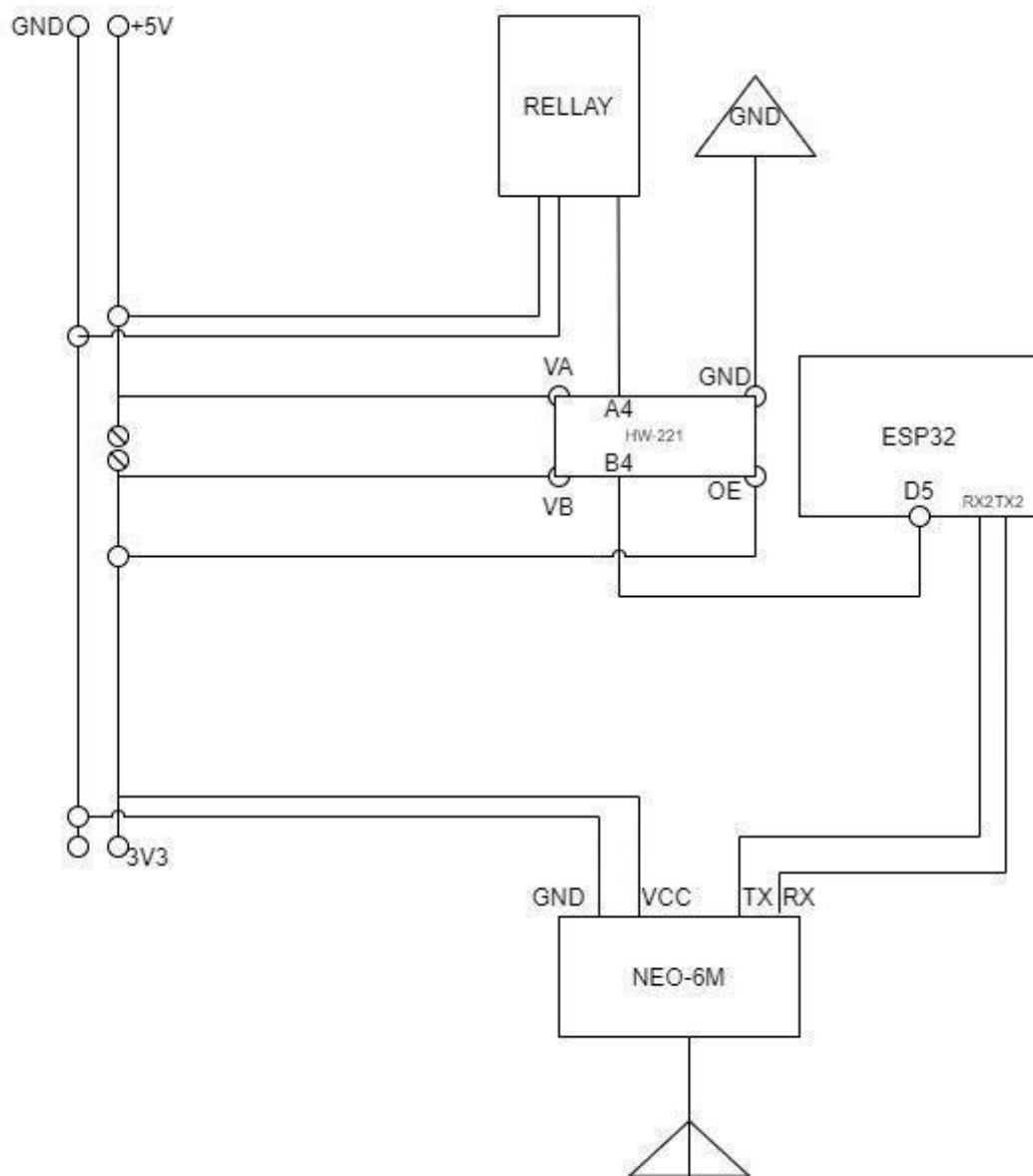


Рисунок 3.11 Принципова схема електричного кола розробленого GPS-трекеру

Складові компоненти електричного кола наведеного на рис. 3.12:

1. Мікроконтролер ESP32 – Обчислення та керування обладнанням;
2. GPS – модуль NEO-6M – Обмін даними з всесвітньою системою позиціонування GPS;
3. Двоканальне реле – керування системою енергоживлення об'єкта;
4. Перетворювач логічних рівнів HW-221 – Розмежування пристроїв з різною робочою напругою.

Обрані протоколи обміну даними між пристроями:

- ESP32 – веб–сторінка: Asynchronous TCP;
- ESP32 – пам'ять Flash: Serial Peripheral Interface Flash File System (SPIFFS);
- ESP32 – реле: взаємодія на рівні сигналів;
- ESP32 – датчик температури: обмін даними за протоколом Inter–Integrated Circuit (I2C);
- ESP32 – Blynk: обмін даними за допомогою HTTP запитів.

3.6 Реалізація фонового моніторингового процесу

Для коректної роботи системи збору даних прийнято рішення реалізувати спеціальний програмний застосунок, сумісний з більшістю сімейств інтегрованих систем на базі мікропроцесорів, для інтеграції функціоналу збору даних.

Застосунок працює у фоновому режимі, а не під прямим контролем користувача, чекаючи активації внаслідок виникнення певної події чи стану.

Для забезпечення швидкодії прикладного застосунку, усі дані зберігаються у бінарних пошукових структурах даних, опрацювання даних відбувається в окремих потоках, а прикладних програмний інтерфейс має віділений потік. У випадку неперебачуваного збою у роботі системи при її перезавантаженні у рамках проектної аварії – застосунок відновлює останній збережений набір конфігурацій з файлу резервного копіювання, збереженого у енергонезалежному сховищі пристрою. Даний тип застосунку використовується у всіх стаціонарних моніторингових пристроях, пунктах збору даних та суміжних зонах контролю.

3.7 Реалізація обміну даними з модулем GPS–приймача

Інтерфейс NEO–6M реалізує протокол UART, виробник надає повну інформацію про передачу даних у посібнику з використання, викладеному у вільний доступ у середовищі інтернет. Інформація про поточні параметри

роботи модулю зберігаються у EEPROM пам'яті модулю, тож першочерговою задачею є реалізація програмного модулю обміну даними з пристроєм GPS. Для зручності подальшого використання вирішено розробити бібліотеку, що реалізуватиме інкапсульований обмін даними з модулем, та публічний інтерфейс структури даних, що міститиме параметри поточного розташування модулю.

Для коректної та спрощеної роботи з даними вирішено розробити та реалізувати наступні структури даних:

- RawDegrees – відповідатиме за обробку значень градусних мір, отриманих від супутників. Зважаючи на специфіку протоколу обміну даними між супутником та модулем від'ємність значень має зберігатися окремим бітом;
- ModuleLocation – Відповідатиме за валідацію значень отриманих від супутника та зберігатиме їх у форматі градусних вимірів широти та довготи відповідно. Також вирішено зберігати дані про останнє поновлення даних, що полегшить визначення актуальності даних на носії;
- NeoGPSDate – відповідатиме за зберігання та валідацію даних отриманих від супутника, що стосуються поточної дати;
- NeoGPSTime – відповідатиме за зберігання та валідацію даних отриманих від супутника, що стосуються поточного часу;
- NeoGPSSpeed – відповідатиме за зберігання та обробку даних що стосуються швидкості модуля у просторі. Для зручності використання структура реалізуватиме методи конвертації швидкості у кілометри на годину. Оскільки супутник надає дані про швидкість у морських вузлах на годину;
- NeoGPSCourse – відповідатиме за отримання та опрацювання даних від супутника про поточний курс модуля;
- NeoGPSAltitude – відповідатиме за отримання та опрацювання даних від супутника про поточну відстань від модуля до рівня моря;

- Додаткові структури для типів даних: Цілочисельний, Десятковий.

Для зручності та безпечності використання, вирішено створити клас, що реалізуватиме інтерфейс обміну даними між користувачем та ядром обробки даних. Клас `ParserNeo_6M` забезпечуватиме зручне користування модулем користувачеві, залишаючи базові інструменти контролю поточною валідацією та обміну даними захищеними від втручання користувацьких програм–реалізацій.

Пропоноване API бібліотеки:

Публічні змінні:

- `Location` – дані типу `ModuleLocation`, що зберігають інформацію про поточне положення модуля;
- `Date` – дані типу `NeoGPSDate`, що зберігають інформацію про поточну дату у пам'яті модуля;
- `Time` – дані типу `NeoGPSTime`, що зберігають інформацію про поточний час модулю;
- `Speed` – дані типу `NeoGPSSpeed`, що зберігають інформацію про поточну швидкість модуля;
- `Course` – дані типу `NeoGPSCourse`, що зберігають інформацію про поточний курс модуля;
- `Altitude` – дані типу `NeoGPSAltitude`, що зберігають інформацію про поточну висоту модулю над рівнем моря;
- `Satellites` – дані, що зберігають інформацію про поточну кількість видимих супутників GPS.

Також доступними методами для користувача є:

- `distanceBetween(double lat1, double long1, double lat2, double long2)` – розрахунок відстані між точками за координатами;
- `static double courseTo(double lat1, double long1, double lat2, double long2)` – розрахунок курсу між точками за координатами;

- `static void parseDegrees(const char *term, RawDegrees °)` – розшифрувати GPS повідомлення і конвертувати їх у градуси;
- `uint32_t charsProcessed()` – отримати кількість опрацьованих символів з GPS – термів;
- `uint32_t failedChecksum()` – кількість нерозпізнаних повідомлень за сеанс;
- `uint32_t passedChecksum()` — кількість вдало розшифрованих повідомлень за сеанс.

ВИСНОВКИ

Після проведення досліджень, з огляду на технологічні особливості, об'єктом розробки обрано інтегровану систему моніторингу та керування промисловим устаткуванням для типових елеваторних комплексів. Було визначено актуальність проведення подальших робіт з розробки апаратно–програмного комплексу для системи моніторингу положення об'єктів. Для використання підходу децентралізованої системи збору та оброблення даних було обрано концепцію IoT.

Темою роботи обрано автоматизацію повного циклу технологічного процесу з використанням інтегрованих систем: збору даних, аварійного контролю, керування виконуючими елементами.

На підставі проведених досліджень, було розпочато проектування системи «IAS Management» з елементами системного контролю та віддаленого керування. При проектуванні апаратного комплексу реалізовано інтегровану систему збору даних локального типу. Такий підхід надає можливість подальшої інтеграції модульних підсистем керування та збору даних. В рамках розробки демонстраційно показано потенціал модульності даного проекту на прикладі компоненту керування енергосистеми.

Для реалізації моніторингу рухомого складу було розроблено програмний продукт для опрацювання GPS–повідомлень з реалізацією стаціонарного та мобільного центру моніторингу. Універсальність інтерфейсних рішень дозволяє проводити швидку модульну модернізацію згідно потреб клієнта. Така система відповідає методам інформатизації та автоматизації, які є основним підходом до організації сучасного технологічного процесу.

Переваги проекту: наявність єдиного інформаційного простору, автоматична обробка інформації, організація продуктивної діяльності системи, реалізація інформаційних відношень. Вимірювальні пристрої, використані в рамках проекту, отримують інформацію в реальному часі та

знаходяться безпосередньо у об'єкті відстежування, що забезпечує високу швидкість збору даних та автоматичний цілодобовий контроль над усіма компонентами структури. Розроблене рішення було представлено на розгляд комісії північно–української експортно–імпортової компанії, в результаті чого було укладено угоду з ТОВ «БІОЕНЕРДЖІ-ЛАЙФ» № 11–45 «про розробку проекту модернізації об'єкту “ЕВ–2”» від 03.10.2020р. Для реалізації положень угоди, я розпочав співпрацю з ТОВ «ЮГОВ Проект» від 05.11.2020 р., у рамках якої мені було надано робоче місце з проприетарною ліцензією програмного забезпечення Siemens TIA Portal версії 15.1. Система реалізована згідно вимог викладених у розділі 1.4. Календарний план заходів з реалізації системи наведено у розділі 3.1.

Згідно висновку «про впровадження модернізаційного проектного рішення “IAS Management”», експлуатаційні витрати підприємства з експлуатації системи «ЕВ–2» (Елеватор) зменшаться на 14% (чотирнадцять відсотків), починаючи з четвертого кварталу поточного року. Витрати з супроводу та реконструкції об'єкту «ЕВ–2» знижаться на 4% (чотири відсотки), починаючи з четвертого кварталу поточного року.

За результатами дослідження комісія вважає можливим впровадження проектних рішень згідно угоди від 03.10.2020 № 11–45, а саме:

- 11–45.1 («Впровадження підсистеми введення–виведення»)
- 11–45.2 («Впровадження підсистеми контролю та обчислень»)
- 11–45.3 («Впровадження підсистеми диспетчеризації»)

Подальший розвиток системи передбачає розкриття потенціалу горизонтального масштабування, закладеного під час проектування системи. Значна частина обладнання, з впровадженням технології CIR, може бути замінена або перепрограмована без зупинки технологічного процесу. Система може бути доповнена новими елементами у будь–який час, що у поєднанні з можливістю масштабування створює довготривалу перспективу експлуатації системи, навіть після планових реконструкцій об'єкту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Статистичний щорічник «Сільське господарство України» за 2011 рік / Держкомстат України, – К.– 2012. – 391с.
2. Зінчук Т.О. Європейська інтеграція: проблеми адаптації аграрного сектора економіки України: моногр. / Т.О. Зінчук.– Житомир: ДВНЗ “Держ. агрокол. ун–т”, 2008. – 384 с.
3. Аграрний сектор України на шляху до євроінтеграції: моногр. / М. Бетлій, О. Бородіна, С. Бородін, А. Гайдуцький, Т.Бородіної. – Ужгород: ІВА, 2006. – 496 с.
4. Видання під редакцією міжнародного фонду «Відродження» Аналіз впливу асоціації з Європейським Союзом на агропромисловий сектор Чернігівщини та Миколаївщини [Електронний ресурс]: Режим доступу до посібника:
https://issuu.com/irf_ua/docs/eu-2014-6
5. Гаврилюк Л.А. Трудові ресурси і соціально–трудові процеси в економічній системі ринку аграрної праці // Збірник наукових праць Уманського державного аграрного університету, 2009, № 63 ч.2, 149 с.
6. Посібник під редакцією ДержКомСтату України [Електронний ресурс]: Режим доступу до посібника:
https://ukrstat.org/uk/druk/publicat/Arhiv_u/07/
7. Проектування інформаційних систем. Посібник. Під ред. В.С.Пономаренко, – К., Вид. центр "Академія", 2002
8. Справочник по монтажу оборудования элеваторов, зерноперерабатывающих и комбикормовых заводов. Под редакцией М. А. Тартаковского М. «Колос» 1983
9. Юдаев Н. В. Элеваторы, склады, зерносушилки. – ГИОРД, 2008. –128 с.
10. Петров И. В. Програмируемые контроллеры. Стандартные языки и приемы прикладного проектирования / Под ред. проф. В. П. Дьяконова. – М.: СОЛОН–Пресс, 2004. – 256 с. ISBN 5–98003–079–4

11. Бергер Г. Автоматизация посредством STEP 7 с использованием STL и SCL и программируемых контроллеров SIMATIC S7–1500. Siemens AG, Нюрнберг, 2005.
12. Программирование с помощью STEP 7. Руководство 6ES7810–4CA07–8BW1. Siemens AG, Нюрнберг, 2004.
13. Слюсар В.И. Цифровые антенные решетки. Решения задач GPS. //Электроника: наука, технология, бизнес. – 2009. – № 1: [Электронный ресурс]: Режим доступа до посібника: http://www.electronics.ru/files/article_pdf/0/article_163_187.pdf
14. HowStuffWorks: How GPS Receivers Work. 2006–05–14. [Электронный ресурс]: Режим доступа до посібника: <https://electronics.howstuffworks.com/gadgets/travel/gps1.htm>
15. Satellite message format. Gpsinformation.net [Электронный ресурс]: Режим доступа до посібника: <http://gpsinformation.net/gpssignal.htm>
16. Peter H. Dana. GPS Week Number Rollover Issues [Электронный ресурс]: Режим доступа до посібника: <https://web.archive.org/web/20130225182002/http://www.colorado.edu/geography/gcraft/notes/gps/gpseow.htm>
17. Interface Specification IS–GPS–200, Revision D: Navstar GPS Space Segment/Navigation User Interfaces (PDF). Navstar GPS Joint Program Office [Электронный ресурс]: Режим доступа до посібника: <https://web.archive.org/web/20120908003700/http://www.losangeles.af.mil/shared/media/document/AFD–070803–059.pdf>
18. NEO–6 u–blox 6 GPS Modules Data Sheet. [Электронный ресурс]: Режим доступа до посібника: [https://www.u-blox.com/sites/default/files/products/documents/NEO–6_DataSheet_\(GPS.G6–HW–09005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO–6_DataSheet_(GPS.G6–HW–09005).pdf)
19. ESP32 Data Sheet. [Электронный ресурс]: Режим доступа до посібника: <https://www.alldatasheet.com/view.jsp?Searchword=Esp32%20datasheet&g>

- [clid=Cj0KCQjwn7j2BRDrARIsAHJkxmx6C1mWop_1PPvb7QswWqiGxe
CCEM2nbcCmBVyrpvITw7Owe_UoiAaAvTNEALw_wcB](https://www.arduino.cc/en/Main/software)
20. ARDUINO IDE. Official software. [\[Електроний ресурс\]: Режим доступу до посібника: https://www.arduino.cc/en/Main/software](#).
 21. USB–UART software drivers. [\[Електроний ресурс\]: Режим доступу до посібника: https://www.silabs.com/products/development-tools/software/usb-to-uart-b
ridge-vcp-drivers](#)
 22. ESP32–Devkit. Espressif Software. [\[Електроний ресурс\]: Режим доступу до посібника: https://dl.espressif.com/dl/package_esp32_index.json](#)
 23. Blynk library. [\[Електроний ресурс\]: Режим доступу до посібника https://github.com/blynkkk/blynk-library](#).

ДОДАТОК А. Вибір сфери дослідження

Було проаналізовано склад Внутрішнього Валового Продукту (далі ВВП) країни по галузям. За статистикою його основними складовими є послуги, промисловість, сільське господарство. Дані про галузеві долі згідно статистики Державної служби статистики України приведені нижче у таблиці 1.1.

Таблиця Склад ВВП України за галузями

Галузь	млн грн	%
Чисті податки на продукцію	298 086	15.06
Оптова та роздрібна торгівля	288 096	14.55
Переробна промисловість	239 066	12.08
Сільське господарство	236 003	11.92

Було досліджено експорт України по галузям. Дані з Міністерства Економічного Розвитку та Торгівлі України приведені нижче на Рисунку 1.1.

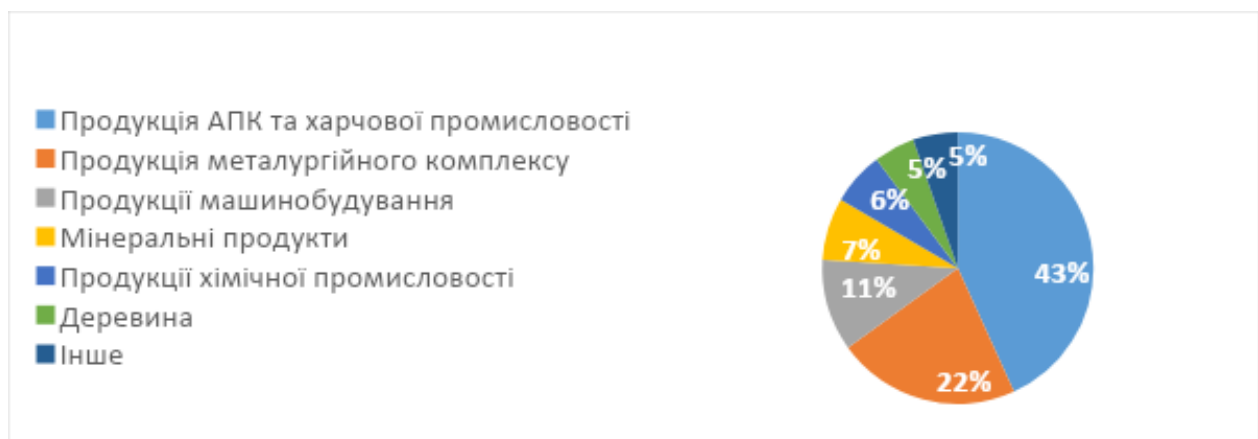


Рисунок 1.1 Структура експорту товарів з України. Міністерство економічного розвитку та торгівлі України

Посилаючись на дані приведені міністерством економічного розвитку та торгівлі України та державною службою статистики України можна дійти висновку, що аграрна сфера є однією з основних галузей ВВП та експорту України. За даними статистичного збірника «Сільське господарство

України» [1] матеріально–технічна база сільськогосподарських підприємств є морально та фізично застарілою, про що свідчать показники рівня зношення устаткування в 70–80%. Застарілі або слабо розвинені підприємства переробки аграрної продукції призводять до скорочення періоду і можливості її реалізації[2].

В рамках вступу України до Світової організації торгівлі постає питання про відповідність вітчизняних технологічних процесів європейським стандартам[3]. Автоматичні системи збору інформації значно зменшують ризики пов'язані з зовнішнім втручанням персоналу та спрощують порядок контролю станів системи та якості продукції й мають відповідати правилам ISO 22000:2007. Під час триваючої асоціації з СОТ стандарти контролю якості продукції повинні бути сформовані відповідно до норм Міжнародної організації зі стандартизації (ISO 24333–2017). Модернізація потребує значних грошових витрат, а оскільки 70% всіх сільськогосподарських угідь знаходяться під контролем приватних підприємців(за даними Статистичного збірника "Сільське господарство України" від 2016 року) [1] необхідно забезпечити інвестиційну привабливість. Нові системи відповідні до рівня розвитку міжнародного агропромислового комплексу (далі АПК) роблять український ринок сільськогосподарської продукції зрозумілим та привабливим для міжнародних інвесторів [4]. Розвитку вітчизняного АПК сприятиме досвід зарубіжних спеціалістів та змога перекваліфікації українського штату [5].

ДОДАТОК Б. Конфігурування мобільного застосунку

API (абр. англ. Application Programming Interface – Прикладний програмний інтерфейс) – Набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення. Спрощено – це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення та програмних бібліотек.

Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для малювання вікна чи іконки на екрані. Програмісти використовують переваги API у функціональності, таким чином їм не доводиться розробляти з початку компоненти програми. API є абстрактним поняттям – програмне забезпечення, що пропонує деякий API, часто називають реалізацією (англ. implementation) даного API. У багатьох випадках API є частиною набору розробки програмного забезпечення, водночас, набір розробки може включати як API, так і інші інструменти/апаратне забезпечення, отже ці два терміни не є взаємозамінювані.

Високорівневі API часто програють у гнучкості. Виконання деяких функцій нижчого рівня стає набагато складнішим, або навіть неможливим. Це означає, що незалежно від кількості функціоналу, що доступний користувачу через API – все одно багато можливостей для розробки і вдосконалення втрачається. Все, що не передбачене видавцем API назавжди залишиться недоступним, навіть якщо програма чи система може мати більший набір функціоналу.

Існує два основних варіанти впровадження прикладного програмного інтерфейсу:

- Захист інформації про програмний інтерфейс від широкого загалу. Наприклад, компанія *Sony* дозволила розробляти програмний

інтерфейс для *PlayStation 2* лише ліцензованим розробникам. Це дозволило *Sony* контролювати, хто розробляв ігри для *PlayStation 2*. Такий варіант надає компаніям переважаючий контроль якості за випуском продукції, і також надає можливості для додаткового доходу від ліцензування;

- Надання можливості використання ППІ для сторонніх розробників. Наприклад, компанія *Microsoft* зробила програмний інтерфейс до *Microsoft Windows* загально доступним, а компанія *Apple*, у свою чергу, впроваджує прикладні програмні інтерфейси *Carbon* та *Cocoa*, для того, щоб дозволити писати програмне забезпечення під свої платформи.

Blynk є додаток для мобільного пристрою, що дозволяє в формі графічного конструювання створювати повнофункціональні додатки для зв'язку їх з пристроями на платформі *Arduino* і сумісними з нею. Для початку роботи достатньо встановити додаток *Blynk* на ваш мобільний або планшет. У кожного елемента є можливість налаштування параметрів, але основа криється в прив'язці елементів управління або відображення до контактів фізичного пристрою. Контакти, можуть бути як справжніми (фізичними) так і віртуальними.

На стороні пристрою досить тільки підключити бібліотеку, налаштувати зв'язок між програмою і реєстраційним кодом додатку у застосунку *Blynk*, помістити в цикл обробник подій і призначити алгоритми обробників.

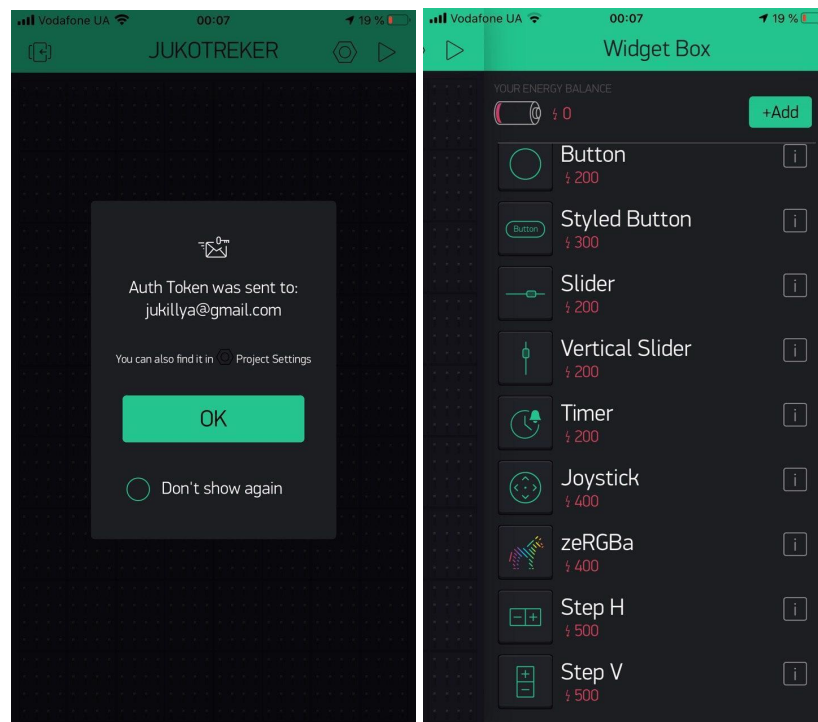
Для завантаження додатку перейдіть до магазину вашої операційної системи (*Play Store* для *Android* і *App Store* для *Apple*). У рядок пошуку введіть назву застосунку *Blynk*. Після завершення пошуку оберіть та інсталюйте застосунок.

Кожен віджет *Blynk* потребує енергії для роботи. Енергія являє собою внутрішню валюту, коли ви видаляєте віджет, енергія повертається. Ви

завжди може не придбати більше енергії у магазині. В рамках роботи не передбачається закупівля додаткової енергії, тому демонстрація різних можливостей продукту буде наведена окремо.

Для створення нового проекту необхідно вказати ім'я проекту у полі введення «Project Name» (англ. Назва проекту), обрати модель пристрою з яким буде здійснюватися комунікація і обрати колірну тему для проекту (світлу (англ. Light) або темну(англ. Dark)). Натисніть кнопку «Створити» (англ. Create). На вказану персональну пошту буде надіслано спеціальний ідентифікаційний код, який ми згодом використаємо при програмуванні ESP32. Натисніть кнопку «ОК».

Також унікальний ідентифікатор проекту завжди доступний у налаштуваннях проекту.



Вікно порожнього проекту

Після створення проекту, додаток перейде на сторінку конструювання додатку. Інтерфейс застосунку наведено на рисунку. На формі присутні елементи керування:

- для початку роботи проекту необхідно натиснути кнопку «Запуск» (позначена трикутником);
- для переходу до налаштувань проекту необхідно натиснути кнопку «Налаштування» (позначена іконкою шестерні);
- для відкриття оглядача віджетів необхідно натиснути кнопку «Додати» (позначений іконкою +).

Відкрийте оглядач віджетів натиснувши іконку + у верхній лівій частині інтерфейсу застосунку. Оберіть елемент button, після чого у просторі проекту з'явиться обраний елемент.

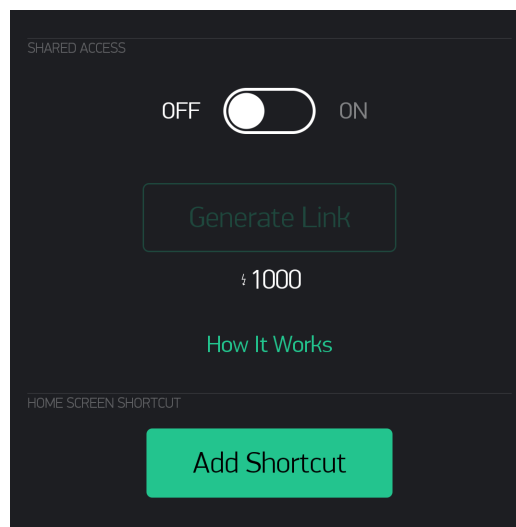
Натисніть і утримуйте елемент для переміщення, натисніть і відпустіть для переходу до параметрів.

Відкрийте параметри кнопки і встановіть параметру PIN значення V1 (або будь-яке інше, з переліку V). Дійсно, додаток може керувати і фізичними командами (за номером GPIO), проте використання віртуальних контактів надає значно більший перелік можливостей.

Також слід перемістити перемикач Push–Switch у режим перемикача, оскільки такий параметр дозволить кнопці відображати поточний стан пристрою, яким він керує.

ДОДАТОК В. Інтеграція Blynk API у проект

Засобом підключення мобільного застосунку *Blynk* до апаратної платформи виступає згенерований при створенні проекту у додатку ключ код. Унікальний ідентифікатор гарантує встановлення зв'язку між сторонами. На рисунку зображено інтерфейс Blynk.



Інтерфейс налаштувань проекту у додатку *Blynk*

У скетчі Arduino створіть змінну, яка зберігатиме ідентифікатор:

```
char auth[] = "16xO4sQik49hqR_cvpieqZTe2dY661-f";
```

Для ініціалізації обміну даними між віджетами застосунку і програмою проводиться встановлення з'єднання за допомогою функції *Blynk.begin()*:

```
Blynk.begin(auth);
```

Слід зазначити, що функція підтримує можливість підключення до мережі WiFi та сервісів Blynk за допомогою команди:

```
Blynk.begin(auth, ssid, pass);
```

Підтримка з'єднання забезпечується функцією:

```
Blynk.run();
```

Для обміну даними між віджетами та програмою на контролері створюється набір обробників подій, прив'язка яких здійснюється автоматично бібліотечними структурами. Для отримання значень з застосунку *Blynk* створюється група методів *BLYNK_WRITE*:

```
BLYNK_WRITE(V1){  
    ledState = !ledState;  
}
```

У дужках вказується назва віртуального контакту, до якого прив'язаний віджет.

Для запису даних у додаток *Blynk* використовується:

```
Blynk.virtualWrite(V2, value);
```

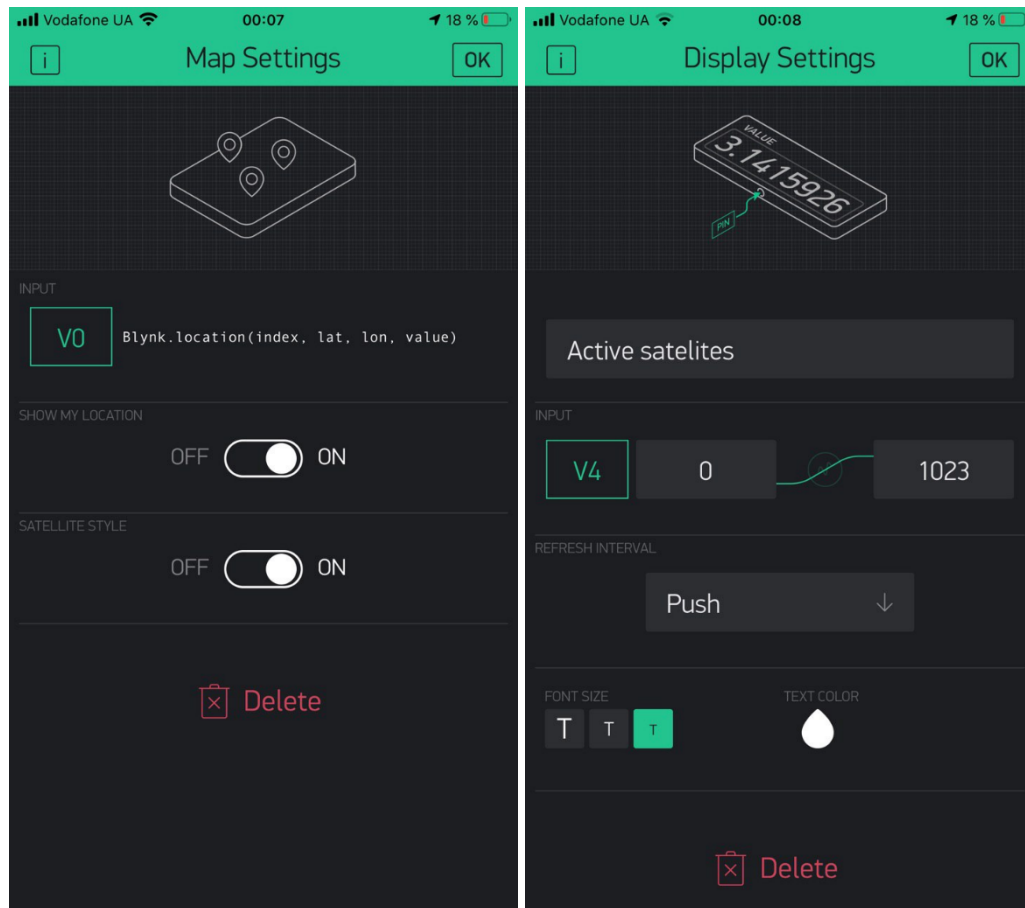
Де у дужках вказується номер віртуального контакту та значення, що потрібно передати віджета. При цьому у додатку необхідно обрати спосіб отримання значення як PUSH.

Далі проведемо конфігурацію віджетів для роботи з розробленим ПЗ.

Всього знадобиться 8 віджетів:

- 4 покажчика значень «Value Display»;
- 1 карта;
- 1 віджет повідомлень;
- 1 кнопка покажчик стану двигуна;
- 1 віджет Bluetooth для підтвердження власника.

Скріншоти конфігурацій віджетів наведені нижче на рисунку.



Налаштування віджету «Карта» та покажчика активних супутників.

Відповідна конфігурація переводить обидва віджета у режим поновлення командою від мікроконтролеру. Форматування даних реалізується на стороні обчислювального пристрою та має наступну параметризацію для віджету карта:

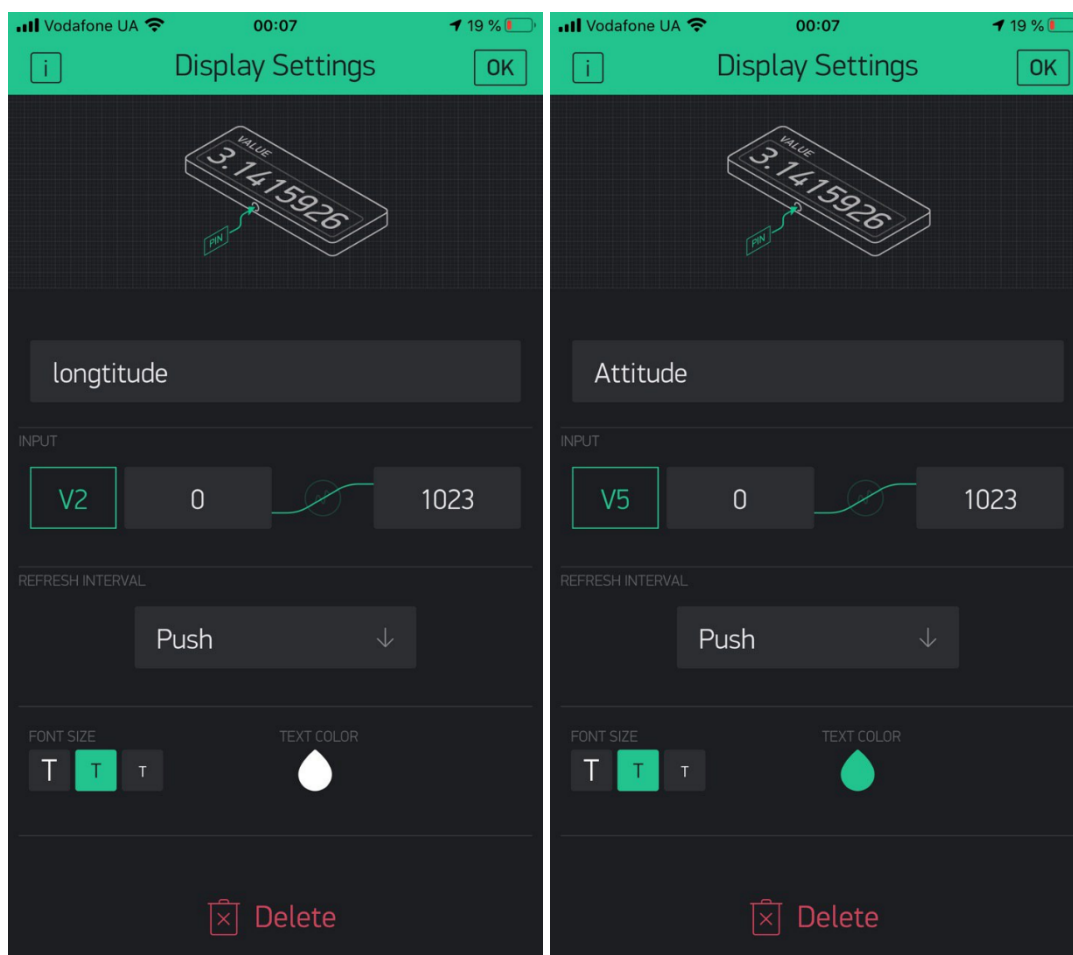
```
myMap.location(1, latestPosition[0], latestPosition[1], "Tracker Position");
```

Перший параметр визначає можливість перезапису положення мітки, наступні два відповідають за значення широти та довготи, тоді як останній передає бажане ім'я мітки на карті. Приведення типів не використовується адже бажано використовувати максимальну точність при відображенні координат.

Для віджету покажчика кількості активних супутників:

```
Blynk.virtualWrite(V4, (int)sats);
```

Для передачі цілочисельного значення використаємо приведення типів. Оскільки кількість супутників не може мати значень після коми – можемо просто відкинути дрібне значення без округлення. На рисунку зображено налаштування інтерфейсів Blynk.



Налаштування віджетів – показчиків поточної довготи та широти.

Обидва віджета переведемо в режим поновлення значень з боку контролера. Проте для даного типа показчиків необхідно параметрично задати точність показань, оскільки максимальна роздільна здатність віджета більша ніж у придбаного модуля NEO–6M.

Для конвертації використаємо метод приведення типів до рядкових значень з округленням:

```
Blynk.virtualWrite(V1, String(latestPosition[0], 4));
```

Точність у чотири позиції після коми для показань довготи та широти.

```
Blynk.virtualWrite(V5, String(latestPosition[2], 1));
```

Точність у одну позицію після коми для висоти над рівнем моря.

ДОДАТОК Г. Реалізація динамічного підключення

Для динамічного поновлення елементів сторінок без перезавантаження документа використаємо запити до сервера, які реалізовані за допомогою інтерфейсу *XMLHttpRequest()*. Інтерфейс *XMLHttpRequest()* є програмним, а отже він визначає правила взаємодії між компонентами програми або програм.

Надалі мова піде про взаємодію між інтегрованими у браузер інструментами і сервером.

Для початку роботи з *XMLHttpRequest()* виконується наступний код:

```
var xhttp = new XMLHttpRequest();
```

Після ініціалізації, буде автоматично створений обробник подій, який отримає назву *onreadystatechange*, проте доступ до нього можливий лише з визначеного і ініціалізованого *XMLHttpRequest()*. Обробник викликає зв'язану функцію кожний раз, коли властивість *readyState* запиту змінюється. Значення станів і їх опис наведено у таблиці

Таблиця. Значення станів XMLHttpRequest та їх опис

#	Стан	Опис
1	UNSENT	Об'єкт створено, функція <code>open()</code> не викликалаь
2	OPENED	функцію <code>open()</code> виконано
3	HEADERS_RECEIVED	Функцію <code>send</code> виконано, заголовки <code><head></code> отримано
4	LOADING	Триває процес завантаження
5	DONE	Операцію завершено

Створимо функцію, яка викликатиметься кожного разу, коли змінюється стан запиту:

```
xhttp.onreadystatechange = function() {
};
```

Створивши набір команд всередині функції ми отримаємо їх автоматичне виконання по зміні стану запиту. Оскільки в рамках завдання необхідно поновлювати значення без перезавантаження сторінки, надалі нам необхідно накласти де-які обмеження на виконання функції, оскільки взаємодія з вмістом документа можлива лише при успішному виконанні попереднього запиту, і коректній роботі сервера:

```
if (this.readyState == 4 && this.status == 200) {
}
```

Так як ми звертались до представника *XMLHttpRequest* за допомогою його екземпляру *xhttp* – *xhttp.onreadystatechange*. Такий саме принцип стоїть і за використанням ідентифікатора *this*, він завжди містить посилання на об'єкт який викликав функцію. В даному випадку це є об'єкт *xhttp*. Звернутись до нього можна і напряду, проте використання *this* гарантуватиме належне посилання, адже в даній програмі обробників і екземплярів *XMLHttpRequest* буде декілька.

Тепер напишемо приклад коду, який поновлюватиме значення широти, довготи або будь-якого іншого значення. Нам необхідно отримати нове значення параметра та вставити його в документ.

Отримати посилання на елемент за його назвою можна використавши функцію `getElementById("Назва")`. Викликати функцію необхідно по відношенню до сторінки, яка носить назву *document* у браузері.

```
document.getElementById("latitude")
```

Поверне вказівник на елемент *latitude* на веб-сторінці.

```
document.getElementById("latitude").innerHTML = this.responseText;
```

Така конструкція отримає вказівник на елемент з назвою *latitude* після чого використає його у якості об'єкту по відношенню до якого буде виконано виклик *.innerHTML* який дозволить замінити текстовий вміст тегу. Джерелом нового тексту слугуватиме відповідь від сервера.

Для формування запита на сервер необхідно виконати код, схожий на вміст обробника запитів на стороні сервера.

```
xhttp.open("GET", "/latitude", true);
```

Такий код створить запит, або відредагує існуючий. Параметр *Get* (англ. Отримати) визначає тип запиту, */latitude* є параметром запиту, а флаг *true* визначає чи є запит асинхронним (*true*), або блокуючим (*false*). Для відправки підготованого запиту скористайтесь функцією

```
xhttp.send();
```

Останнім кроком стає визначення інтервалу поновлення. Для цього скористаємось інструментом *setInterval*, який приймає два аргумента – функція яка має бути викликана по вичерпанню заданого строку і власне значення інтервалу. Такий таймер викликатиме функцію раз на 10 секунд:

```
setInterval(function (), 10000 );
```

Для спрощення, JS дозволяє визначити метод прямо в середині параметрів функції *setInterval*. Для розміщення коду JS всередині HTML оточимо його тегом `<script></script>`. Загальний код наведено нижче.

```
setInterval(function () {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("longitude").innerHTML = this.responseText;
```

```

document.getElementById("RTB").innerText += this.responseText;
lon = this.responseText;
}

document.getElementById("NavigateButton").onclick = function() {
this.href = "http://maps.google.com/maps?&z=15&mrt=yp&t=k&q="+ lati + "+" + lon;
console.log(this.href);
};
};
xhttp.open("GET", "/humidity", true);
xhttp.send();
}, 10000 );

```

В результаті отримання визначеного набору параметрів, виконується команда у фігурних лапках, що відправляє клієнту текст, замість звичайного документа html. Значення *latitude* зберігає на платформі ESP32 значення широти та довготи перетворюється з цілочисельного типу даних до строкового. За допомогою команди *String(latitude).c_str()*.

Зауважу, що використання сервісу Google Maps без додаткової платної підписки можливе лише через офіційний ресурс карт. Спеціальний фрагмент скрипту компонує посилання з відповідними дійсним координатам та відкриває посилання у новій вкладці браузера:

```

document.getElementById("NavigateButton").onclick = function() {
this.href = "http://maps.google.com/maps?&z=15&mrt=yp&t=k&q="+ lati + "+" + lon; };

```

ДОДАТОК Д. Система адаптивних сповіщень

Push повідомлення – це невеликі спливаючі вікна на екрані вашого пристрою. Вони можуть з'являтися на екрані будь-якого пристрою, де є область сповіщень, або є можливість виведення на екран даних прийнятих з мережі Інтернет.

Ви напевно приймаєте оповіщення на свою електронну пошту: заходите в поштовий клієнт і переглядаєте листи. В даному випадку ця технологія є технологією *pull*.

У випадку ж з *push*–повідомленнями, ресурс відправляє нові дані клієнту самостійно. При цьому ви відразу отримуєте найсвіжіші дані, адже в даній технології немає певного періоду перевірки даних, вони приходять в режимі онлайн. Використовуючи *Push повідомлення* ви не обов'язково повинні отримувати повідомлення. Через *push*–технологію можна наприклад синхронізувати дані якщо вони оновилися.

Повідомлення бувають:

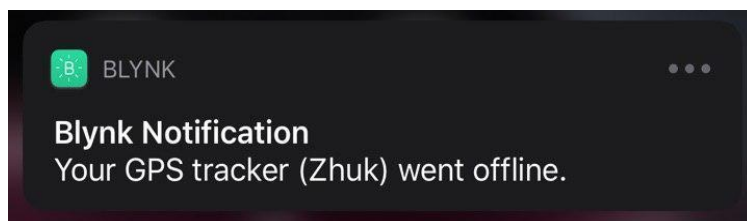
- Браузерні – через *push api* або через додаток;
- Мобільні – через мобільний додаток;
- На РС через додаток, незалежно від браузера.

Мобільні *push* повідомлення – це оповіщення, які приходять на мобільний телефон від різних додатків. Для системи з розширеним охоронним функціоналом мобільні сповіщення відіграють важливу роль, адже час реакції на несанкціоноване втручання є важливим параметром будь-якої системи контролю станів. Усі сценарії для повідомлень задаються програмно на стороні контролеру, та відправляються клієнтам по виконанні певні умови. Від так система може надсилати сповіщення по:

- Зміні стану об'єкту контролю;
- Прибутті рухомого об'єкта до визначених координат або їх околу;
- Перевищення порогової швидкості об'єкта;

- Виходу системи з мережі;
- Появи системи у мережі.

Приклад таких повідомлень наведено на рисунку.



Приклад сповіщення Blynk