

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально–науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Розроблення проєкту програмно–апаратного комплексу
автоматизації процесу злиття посилок на логістичних терміналах»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2021
Стаднік Максим Ігорович
Керівник: Одарущенко Олена
Борисівна
Рецензент: Муравльов Володимир
Вячеславович

Полтава – 2025 року

ВСТУП

Сучасні тенденції розвитку логістики вимагають від підприємств підвищення ефективності, точності та швидкості обробки посилок у всіх ланках логістичного ланцюга. Одним із ключових напрямів оптимізації є впровадження програмно–апаратних систем автоматизації, які дозволяють зменшити частку ручної праці, знизити кількість помилок та забезпечити цілісність обробки даних у режимі реального часу.

Актуальність теми – автоматизація процесу злиття посилок є критично важливою з точки зору забезпечення коректного сортування, мінімізації логістичних витрат, формування відправлень відповідно до заданих маршрутів, а також підготовки їх до автоматизованого зчитування штрих кодів. Попри наявність окремих рішень у цій галузі, більшість із них не враховують особливостей національної логістичної інфраструктури, не забезпечують гнучкого масштабування системи відповідно до потреб підприємства, а також мають низку інших обмежень. Зокрема, переважна частина таких рішень розробляється за кордоном, відрізняється надзвичайно високою вартістю через обмежену кількість варіантів конфігурації, складністю технічного обслуговування, а також труднощами у внесенні змін до системи. Додатковим ускладненням є наявність технічної підтримки виключно іноземною мовою. Саме тому розробка ефективного програмно–апаратного комплексу автоматизації процесу злиття посилок є актуальним і науково–практично значущим завданням.

Мета роботи – розроблення ескізного проєкту програмно–апаратного комплексу автоматизації процесу злиття посилок на логістичних терміналах.

Для досягнення поставленої мети визначено такі завдання:

- провести аналіз сучасних програмно–апаратних платформ, що використовуються в промисловій автоматизації;
- розробити структурну та функціональну схеми комплексу автоматизації;

– створити ескізний проєкт програмно–апаратного комплексу автоматизації процесу злиття посилок.

Предмет дослідження кваліфікаційної роботи – є методи, технічні засоби та принципи побудови програмно–апаратного комплексу для автоматизації процесу злиття посилок.

Об'єкт дослідження кваліфікаційної роботи – є процес автоматизації технологічних операцій на логістичних терміналах.

Методи дослідження: системний аналіз (розділи 1, 2), ітераційне розроблення (розділи 2, 3).

Інформаційна база, технічна документація виробників апаратних засобів, наукові праці, статті в профільних виданнях, а також результати досліджень у сфері логістики та автоматизації.

Практична значущість: роботи полягає у можливості застосування розробленого ескізного проєкту як основи для створення реального програмно–апаратного комплексу, який може бути впроваджений у діяльність логістичних компаній з метою автоматизації процесу злиття посилок.

Робота складається зі вступу, трьох розділів, висновків та списку літератури. Обсяг роботи становить 62 сторінки, 3 таблиці, 34 рисунка, 8 додатків.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНИХ ПРОГРАМНО–АПАРАТНИХ ПЛАТФОРМ ПРОМИСЛОВОЇ АВТОМАТИЗАЦІЇ

1.1 Сучасні підходи до автоматизації логістичних процесів

Сучасна логістика зараз активно змінюється завдяки впровадженню цифрових технологій та автоматизованих систем керування. Основним напрямком розвитку є автоматизація повторюваних та рутинних операцій з метою підвищення швидкості, точності й ефективності. У сучасних умовах розвитку електронної комерції та зростання обсягів вантажоперевезень ефективність логістичних процесів набуває критичного значення.

Одним із головних інструментів сучасної автоматизації є програмовані логічні контролери (PLC). Програмований логічний контролер (англ. programmable logic controller) – електронний пристрій, який використовується для автоматизації технологічних процесів таких як, керування конвеєрною лінією, насосами тощо [1]. Ці пристрої, що дозволяють реалізовувати керування обладнанням у реальному часі, забезпечують виконання логіки обробки вантажів, управління привідною технікою, контролю за станом системи і передачі даних.

Програмований контролер значною мірою усунув необхідність у складному жорсткому з'єднанні, притаманному релейним системам керування. Додатковими перевагами є швидка реакція, простота програмування та встановлення, висока швидкість керування, сумісність з мережами, зручність діагностики й тестування, а також висока надійність [2].

У поєднанні з PLC активно використовуються:

- фотодатчики, датчики наявності – для виявлення посилок і їх характеристик;
- енкодери – для контролю переміщення по конвеєрах;
- сканери штрих кодів – для ідентифікації вантажів;

- частотні перетворювачі, електродвигуни – для керування конвеєрами і транспортувальними механізмами;
- панелі оператора (HMI) – для візуалізації процесів, керування параметрами, відображення стану системи.

Контролери Schneider Electric, серії Modicon, ідеально підходять для таких завдань завдяки підтримці різноманітних інтерфейсів (Modbus, Ethernet, CANopen), гнучкому програмуванню, широкому вибору модулів розширення та сумісності з HMI-панелями.

Modbus (RTU/TCP) – це один із найпопулярніших протоколів зв'язку, розроблений для обміну даними між пристроями в промислових автоматизованих системах [3]. Цей протокол допомагає пристроям, наприклад датчикам і контролерам обмінюватись даними, простий в налаштуванні. Modbus RTU (Serial) – працює через послідовний інтерфейс (RS-485, RS-232) має низьку швидкість в порівнянні з іншими протоколами, реалізованими на контролері (рис. 1.1).

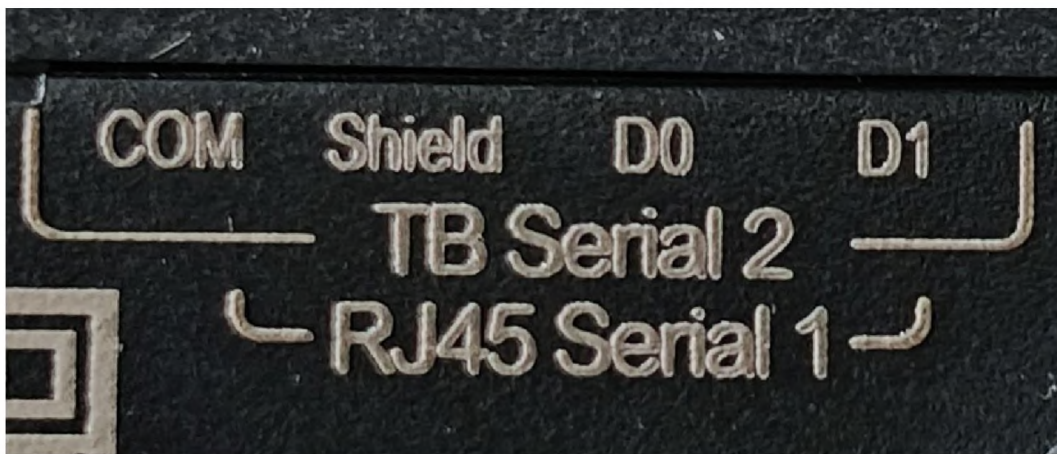


Рисунок 1.1 – Приклад реалізації клемного блоку Serial на контролері M241

Modbus TCP – реалізується через Ethernet має високу швидкість а також простий та універсальний (рис. 1.2). Ethernet/IP – найпопулярніший протокол кабельних комп'ютерних мереж, що працює на фізичному та каналному рівні мережевої моделі OSI [4].



Рисунок 1.2 – Зовнішній вигляд Ethernet порту на контролері M241

Високо швидкісний протокол для інтеграції пристроїв у промислових мережах. Підтримує детермінований обмін даних. Досить швидке й стабільне передавання великих обсягів даних. Зручно для зв'язку між кількома PLC та HMI.

CANopen – протокол зв'язку на основі CAN. Стандарт CANopen є корисним, оскільки він забезпечує готову взаємодію між пристроями (вузлами), наприклад, промислове обладнання. Крім того, він надає стандартні методи конфігурації пристроїв (рис. 1.3) [5].



Рисунок 1.3 – Зовнішній вигляд CANopen порту на контролері M251

Цей протокол використовується для створення компактних, та надійних локальних мереж в промисловості, дає змогу з'єднати в одну систему багато простих пристроїв, підходить для розподілених систем.

Окрім підтримки різних протоколів, контролери Schneider Electric відповідають міжнародному стандарту IEC 61131-3 [6] який дозволяє програмувати їх на кількох мовах:

- LD (Ladder Diagram) – найпоширеніша мова для автоматизації.
- FBD (Function Block Diagram) – зручна для графічного створення алгоритмів.
- ST (Structured Text) – текстова мова, подібна до Pascal або C, підходить для складної логіки, обробки масивів, циклів.
- SFC (Sequential Function Chart) – для побудови послідовних процесів, наприклад, логіки злиття.

Стандарт IEC 61131 складається з кількох частин, що охоплюють різні аспекти проектування та реалізації програмованих логічних контролерів:

- IEC 61131-1 – загальні положення: описує терміни, структуру PLC, їх функціональні характеристики [7].
- IEC 61131-2 – вимоги до апаратного забезпечення: визначає електричні, механічні та екологічні характеристики PLC, а також методи тестування [8].
- IEC 61131-3 – мови програмування: ключова частина, що описує 7 мов програмування (LD, FBD, ST, IL, SFC, CFC, Fuzzy), типи даних і програмну структуру.
- IEC 61131-4 – керівництво користувача: рекомендації з використання стандарту в практичних задачах автоматизації [9].
- IEC 61131-5 – комунікація: моделі обміну даними між PLC та іншими пристроями [10].
- IEC 61131-6 – функціональна безпека: вимоги до PLC у критичних системах згідно з рівнями SIL [11].
- IEC 61131-7 – нечітке керування: підтримка fuzzy logic для нечітких систем автоматизації [12].
- IEC 61131-8 – рекомендації реалізації: методичні вказівки для розробників середовищ програмування [13].

– IEC 61131–9 – IO–Link: інтерфейс для підключення дискретних датчиків та виконавчих механізмів [14].

– IEC 61131–10 – XML–формат: стандарт на обмін проектами між різними програмними середовищами на основі XML [15].

З упровадженням міжнародного стандарту IEC 1131 (пізніше перейменованого на IEC 61131) було створено основу для уніфікованого програмування PLC, що використовує сучасні концепції програмної інженерії [16].

Після загального ознайомлення з темою зазвичай зосереджуються на програмуванні контролера конкретного виробника. Однак значно корисніше вивчати PLC за допомогою універсальної мови, яка надає ґрунтовне уявлення про принципи програмування, що можуть бути адаптовані до будь–якого контролера [17].

Відповідно можливо обрати найзручніший підхід або поєднати кілька мов, що полегшить реалізацію поставленої задачі таких як обробка сигналів від сенсорів, енкoderів і реалізація логіки злиття.

1.2 Програмно–апаратний комплекс у сучасній логістиці

Сучасна логістика потребує точності, швидкості і гнучкої адаптації до змін обсягів і структури вантажопотоку. У таких умовах програмно апаратні комплекси (ПАК) відіграють ключову роль забезпечуючи автоматичне керування логістичних процесів.

Програмно апаратний комплекс – це інтегрована система, яка поєднує в собі програмне забезпечення і апаратне забезпечення [18]. До програмного входить:

- алгоритми логічного керування (наприклад, визначення, які посилки в який момент потрібно зливати);
- обробка сигналів від датчиків, енкoderів;

- інтерфейси для взаємодії з оператором (через НМІ–панель).

До апаратного входить:

- програмований логічний контролер – керують усім обладнанням;
- датчики, енкодери– визначають фізичні параметри посилок (розмір, розміщення і рух);
- частотні перетворювачі – приймають команди від контролера і керують моторами згідно заданих в них налаштуваннях;
- панелі оператора (НМІ) – дають змогу персоналу контролювати та налаштовувати систему;
- привідна техніка – двигуни, конвеєри все, що забезпечує фізичне переміщення посилок.

ПАК має низку переваг над людською обробкою. Підвищення швидкості обробки вантажів, він працює в режимі реального часу. Всі дії виконуються за 20 мс – Виявлення посилки на лінії з точністю до 1 см, прийняття рішення, команда до частотного перетворювача і це одночасно обробка з 9 напрямків. Це значно скорочує затримки та дозволяє обробляти великі кількості вантажу. Людина, що викладає вантаж з фури не вирішує які посилки зливати – це виконує алгоритм, закладений у ПАК. Зменшення людського чинника підвищує точність і прогнозованість роботи. ПАК можна перепрограмувати або розширити, якщо змінюються логістичні вимоги [19]. Система може себе само діагностувати і здійснювати оперативне втручання в разі збоїв.

Таким чином програмно–апаратний комплекс є центральним елементом автоматизованого логістичного терміналу, що забезпечує інтелектуальне керування вантажопотоком.

1.3 Огляд обладнання Schneider Electric для задач автоматизації

Компанія Schneider Electric є одним із провідних світових виробників рішень для промислової автоматизації. У сфері автоматизації процесів вона

пропонує ефективні, надійні та масштабовані рішення на базі контролерів, панелей оператора та програмного забезпечення. Особливе місце в лінійці займають контролери серії Modicon, які часто використовуються у складі програмно–апаратних комплексів (ПАК) для управління різними задачами.

Modicon M241 – це компактний, високопродуктивний контролер, розроблений спеціально для керування автоматизованими машинами та розподіленими системами.

Основні характеристики:

- час виконання для 1К інструкції – 0,3 мс для ситуаційної і періодичної задачі, 0,7 мс для інших інструкцій;

- структура завдань – 3 циклічні задачі + 1 вільна задача, або 4 циклічні задачі, або 8 ситуаційних задач;

- пам'ять – до 256 Мб Flash + 64 МБ RAM;

- входи/виходи – до 40 вбудованих дискретних і аналогових, з можливістю розширення через додаткові модулі. Кожен модуль має власний мікропроцесор і виконує попередню обробку сигналів, що забезпечує високу швидкість обробки й надійність системи. Взаємодія між модулями відбувається через системну шину [20]. Система введення/виведення забезпечує інтерфейс між центральним процесором і польовим обладнанням [21];

- час реакції – ≤ 2 мкс увімкнення вимкнення швидких входів і виходів, ≤ 50 мкс увімкнення і вимкнення клеми входів, ≤ 34 мкс увімкнення виходів, ≤ 250 мкс вимкнення виходів;

- комунікація – Ethernet–порт з підтримкою Modbus TCP, 2 порти Serial (RS232/RS485) підтримка Modbus RTU, USB–порт для програмування та діагностики [22].

Отже, основні переваги висока швидкодія, підтримка багатьох протоколів – легко інтегрується з іншими пристроями, гнучка архітектура з можливістю доповнення модулями (рис. 1.4).



Рисунок 1.4 – Зовнішній вигляд контролера M241

Панель Harmony GTO HMIGTO5310 – сучасна серія HMI-панелей від Schneider Electric, призначена для візуалізації процесів автоматизації, налаштування та моніторингу стану обладнання.

Технічні характеристики:

- розмір дисплея: 10";
- Touchscreen (резистивний), кольоровий TFT-дисплей;
- підтримка Ethernet, USB (рис. 1.5);
- підтримка COM-портів (рис. 1.6);
- можливість запису даних, перегляду логів, виведення аварій [23].



Рисунок 1.5 – Розміщення роз'ємів USB та ETHERNET на HMIGTO5310

Панелі Harmony GTO створені з урахуванням повної сумісності з PLC лінійки Modicon M221, M241, M251 та іншими контролерами Schneider Electric. Вони підтримують пряме з'єднання через стандартні промислові протоколи

(Modbus TCP, Modbus RTU, Ethernet/IP), що забезпечує надійний та стабільний обмін даними без необхідності в додаткових адаптерах або шлюзах. Інформаційна ємність, швидкодія та надійність є основними характеристиками засобів відображення інформації. Вони безпосередньо впливають на ефективність взаємодії людини з автоматизованою системою керування [24].

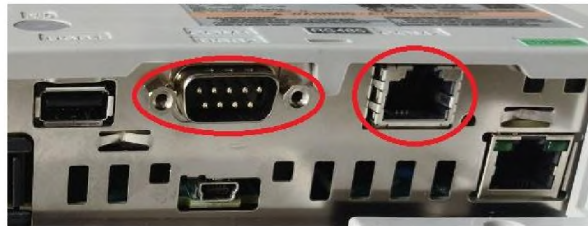


Рисунок 1.6 – Зовнішній вигляд роз'ємів COM на HMIGTO5310

Конфігурування зв'язку відбувається швидко, завдяки готовим драйверам у Vijeo Designer. Панель може автоматично підключатись до змінних, створених у середовищі програмування PLC (EcoStructure Machine Expert), без дублювання імен, що знижує ризик помилок.

Панель є досить зручним способом відображення стану конвеєрів, сенсорів, поточних рішень логіки злиття. Вона забезпечує наочне графічне подання всіх елементів автоматизованої системи, зокрема:

- зображення стану стрічкових транспортерів;
- сигнальні індикатори стану сенсорів (виявлення посылки, помилка датчика);
- виведення поточного стану логіки злиття – наприклад, які посылки в черзі на об'єднання, які вже з'єднані, які відбраковані.

Це дозволяє оператору контролювати систему в режимі реального часу, без потреби звертатися до контролера напряму або через SCADA, і надає можливість змінювати параметри логіки злиття (наприклад, змінити затримку між злиттями або дозволити/заборонити об'єднання з певних напрямків), також можливо запускати або зупиняти окремі елементи системи і переглядати повідомлення

про помилки. Це значно спрощує роботу персоналу, особливо у випадках технічного обслуговування або аварійних ситуацій.

При реалізації програмно–апаратного комплексу, велике значення має не тільки вибір апаратної частини (PLC, HMI, датчики), але й середовище розробки програмного забезпечення. Саме через ПЗ визначається логіка керування процесами, обробка сигналів від сенсорів, управління приводами, а також взаємодія оператора з системою. Schneider Electric надає своє інтегроване середовище розробки для програмування контролерів і HMI панельок і створення візуальних ефектів.

SoMachine – це інтегроване середовище розробки (IDE) від Schneider Electric, орієнтоване на побудову автоматизованих машин і програмно–апаратних комплексів на основі PLC серії Modicon (рис. 1.7).



Рисунок 1.7 – Зовнішній вигляд іконки SoMachine

Програма включає в себе все, що необхідно для програмування контролера при запуску програми одразу доступно декілька варіантів відкрити недавно закриті проекти, перевірити які пристрої підключено, створити новий проект або відкрити проект вручну (рис. 1.8).

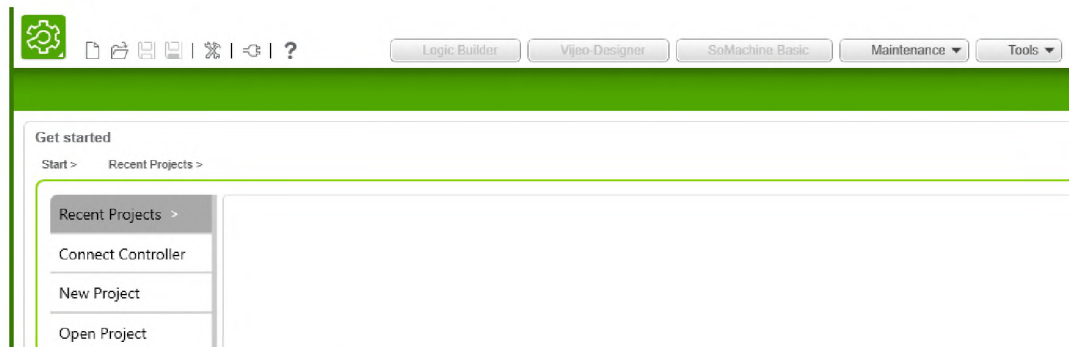


Рисунок 1.8 – Зовнішній вигляд стартового вікна SoMachine V4.3

На вкладці Recent projects можна обрати проект над яким недавно працювали переглянути його назву, час останніх змін, також директорію де збережено проект, час і день створення, заголовок, якщо заповнювався, автора, компанію, якщо вказувалась, і версію проєкту (рис. 1.9).

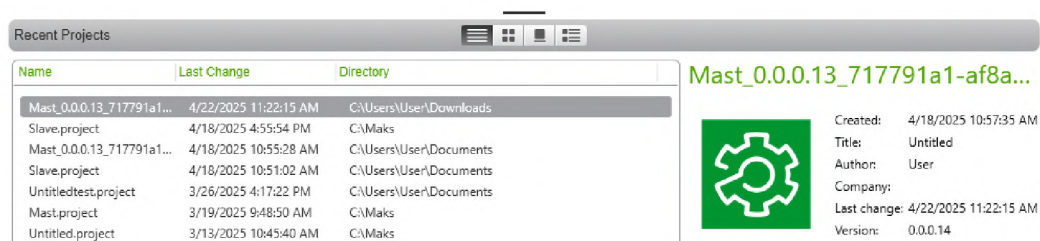


Рисунок 1.9 – Зовнішній вигляд вікна Recent projects SoMachine V4.3

На вкладці Connect Controller можна побачити яким саме способом контролер підключений до пристрою де запущена програма, тип контролера, назва проєкту, що зашитий в контролер, IP контролера (якщо підключено через Ethernet), які картриджі встановлені, час з останнього запуску, автор, внутрішня версія прошивки і Node name пристрою (рис. 1.10).

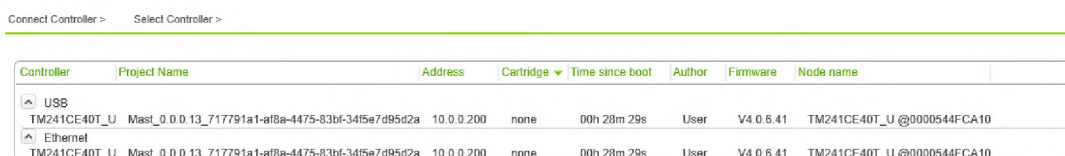


Рисунок 1.10 – Зовнішній вигляд вікна Connect Controller SoMachine V4.3

На вкладці New Project можна обрати 4 варіанти (рис. 1.11).

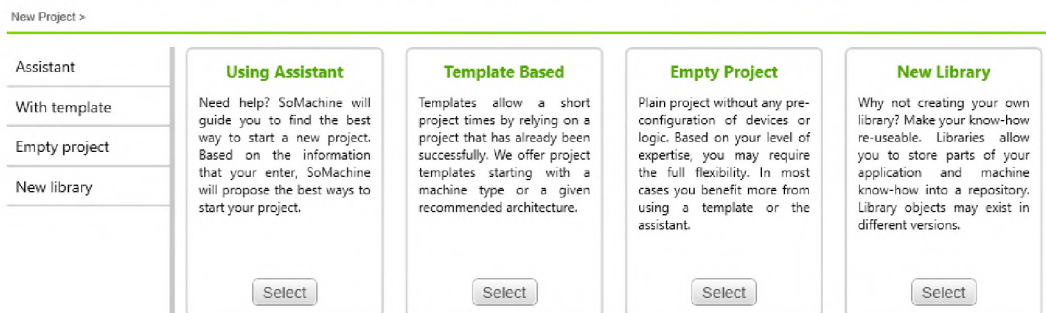


Рисунок 1.11 – Зовнішній вигляд вікна New project SoMachine V4.3

Using Assistant – створення нового проекту з допомогою асистента,
Template Based – створення нового проекту на основі уже існуючого проекту,
Empty Project – створення нового пустого проекту без попередніх налаштувань,
New Library – створити нову власну бібліотеку.

Обравши Open Project відкривається Провідник де можна обирати файли типу *.project; *.library, *.projectarchive для відкриття.

Після створення проекту автоматично відкривається редактор проекту (рис. 1.12). Зліва розміщено дерево девайсів Devices tree – це вікно для конфігурації пристроїв проекту. В ньому можливо додавати контролери, НМІ панелі, додавати до контролера модулі входів виходів також тут налаштовується Ethernet порт і Serial порти. Справа розміщено вікно Devices and Modules усі пристрої та модулі у цьому вікні додаються з бібліотек, які входять до складу SoMachine. З списку можна обрати контролери, модулі входів виходів, мережеві пристрої (інтерфейси), НМІ–панелі, пристрої контролю моторів (частотні перетворювачі), сенсори, безпекові контролери.

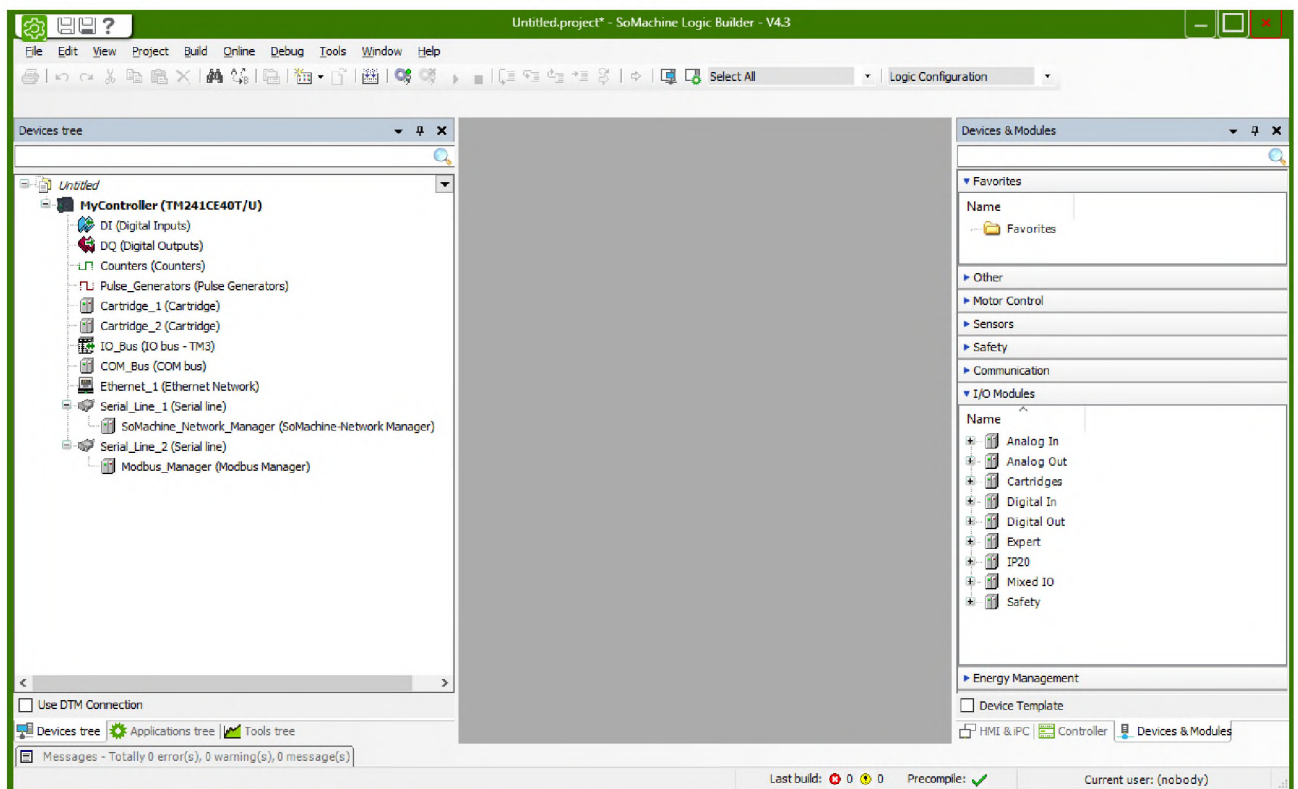


Рисунок 1.12 – Перспектива Logic Configuration в SoMachine V4.3

Вікно Devices tree можна перемкнути на Applications tree (рис. 1.13) або на Tools tree (рис. 1.14). В Applications tree вікні кофігуруються і створюються POU. Створюються глобальні змінні, налаштовуються чи зберігаються змінні в енергонезалежній пам'яті. Також створюються і налаштовуються Task, а саме, коли який виконується. В кожен з taskів додаються відповідні POU, що будуть виконуватись за певних умов. В Tools tree знаходиться менеджер бібліотек також конфігуратор символів, що використовується для налаштування доступу з панелі до змінних контролера. Також присутня вкладка Global де знаходиться загальна інформація про проект, загальні налаштування і в випадку додавання HMI панельки тут можна знайти і HMI програму при її обранні відкривається Vijeo Designer.

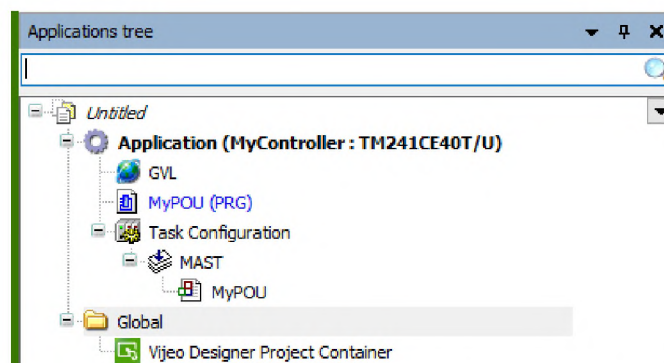


Рисунок 1.13 – Зовнішній вигляд Applications tree в SoMachine V4.3

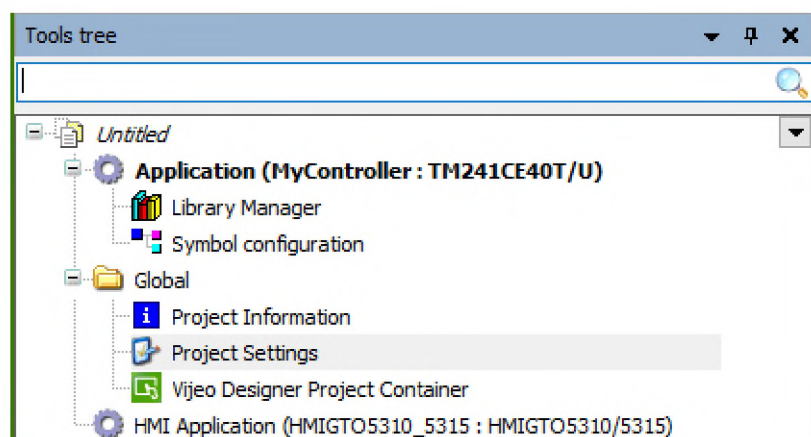


Рисунок 1.14 – Зовнішній вигляд Tools tree в SoMachine V4.3

Vijeo Designer – це середовище розробки HMI інтерфейсів використовується для створення їх на панелях серії Harmony GTO, GTU, STO/STU які можуть зчитувати і обмінюватись даними з PLC і відображати потрібні данні. Створення панелей досить просте просто перетягуються елементи інтерфейсу – кнопки, лампочки, поля введення. В програмі також налаштовується комунікація з контролером. Також можливо налаштувати створення логів, можливе створення попередження про аварії або вивід нестандартних ситуацій.

У результаті огляду обладнання Schneider Electric, призначеного для реалізації задач автоматизації, можна зробити висновок, що компанія пропонує комплексні та інноваційні рішення для створення ефективних автоматизованих систем. Її продукція охоплює широкий спектр пристроїв – від контролерів і сенсорних панелей до комунікаційних модулів, що забезпечують високу продуктивність, надійність і гнучкість. Все вище описане в розділі дозволяє успішно застосовувати обладнання в різних сферах промисловості та підвищувати ефективність технологічних процесів.

РОЗДІЛ 2

РОЗРАХУНОК ПАРАМЕТРІВ СИСТЕМИ АВТОМАТИЗАЦІЇ ПРОЦЕСУ ЗЛИТТЯ ПОСИЛОК НА ЛОГІСТИЧНОМУ ТЕРМІНАЛІ

2.1 Характеристика логістичного терміналу

В роботі досліджені функції та характеристики логістичного терміналу, що виконує функції приймання, сортування вантажу в одній з передових логістичних компаній. Термінал має 50 рамп для завантаження і розвантаження вантажних автомобілів термінал має високу інтенсивність роботи. Проте попри сучасну логістичну архітектуру будівлі, більшість операцій тут здійснюється вручну, що створює ризики для стабільності й ефективності функціонування.

Обробка вантажу відбувається так, одночасно проводиться розвантаження максимум з двох вантажних автомобілів, на дві окремі конвеєрні лінії. Обидві ці лінії керуються вручну: далі оператори формують один спільний потік шляхом поєднання двох входів. Цей етап вимагає фізичної координації та злагодженості дій персоналу. Після об'єднання, потік переходить на центральну сортувальну лінію, яка має велику довжину і оснащена множинними боковими відгалуженнями (непривідними). Біля кожного відгалуження знаходиться оператор зі сканером штрих кодів. У момент проходження посилки працівник здійснює сканування, після чого сканер сигналізує, чи повинна посилка залишатися на центральному конвеєрі чи бути забрана на дану лінію відведення. Весь процес базується на мментальній реакції людини, і вимагає високої концентрації та фізичної присутності на всіх ділянках.

Отже, система працює на межі можливостей: максимальна продуктивність досягає 2500 посилок/годину, що створює серйозне навантаження на працівників. Людський фактор є ключовим обмеженням: несвоєчасна реакція на сигнал сканера або перевантаження в один конкретний порт змушує зупинити всю лінію, що призводить до затримок у логістичному ланцюжку. І термінал може опрацьовувати лише 2 фури на вивантаження. Також періодично один з

потоків, що йде з фури призупиняють і потрібно цими 2 потоками вручну керувати, щоб фури розвантажувались рівномірно. Враховуючи вищезазначені обмеження, виникає об'єктивна необхідність у автоматизації логістичних процесів.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Автоматизація процесу злиття потоків:

– реалізувати алгоритм черговості та синхронізації подачі посилок з 8 вхідних конвеєрів на центральну лінію;

– забезпечити відсутність простоїв та перетинів потоків під час об'єднання;

– впровадити контроль дотримання технологічної відстані між посилками для забезпечення подальшого сканування автоматичними системами ідентифікації.

2. Контроль роботи центрального конвеєра:

– передбачити можливість регулювання швидкості центральної стрічки у залежності від навантаження та логістичних умов;

– встановити датчики контролю накопичення вантажу (затори, надмірне скупчення), з інтеграцією їх показників у систему візуалізації (НМІ);

– реалізувати візуальне відображення місця виникнення затору на панелі оператора у реальному часі.

3. Продуктивність системи:

Забезпечити пропускну здатність не менше 7000 посилок на годину, виходячи з середньої довжини посилки близько 60 см, що вимагає синхронної роботи всіх компонентів системи з точним позиціонуванням.

4. Інтерфейс оператора (НМІ) надає оператору можливість:

– динамічно змінювати параметри системи, зокрема відстань між посилками;

– перемикати окремі вхідні потоки в ручний режим – з можливістю вимкнення для проведення технічного обслуговування;

– автоматично вимикати аварійний потік при виявленні несправностей – без порушення роботи інших ділянок системи;

– контролювати загальний стан системи, переглядати статуси датчиків, зони активності та аварій.

5. Енергозбереження та адаптивність:

– передбачити функцію «засинання» (енергозберігаючого режиму) для окремих потоків або всієї лінії при зниженні навантаження або у випадку простою;

– забезпечити автоматичне «пробудження» системи при поновленні активності хоча б одного потоку.

6. Рівномірне завантаження з машин:

Розробити механізм послідовного або чергового розвантаження з двох автомобілів, щоб уникати пікових навантажень і перевантаження системи.

Таким чином, задача передбачає розробку комплексного програмно-апаратного рішення, що базується на сучасному PLC (наприклад, Modicon M241), інтегрованому із сенсорами, привідною технікою та операторським інтерфейсом (HMI). Проект має забезпечити високу надійність, масштабованість, а також можливість адаптації до майбутніх змін або розширення логістичного терміналу. Також важливим аспектом є забезпечення безперервного моніторингу та дистанційного керування системою з метою підвищення ефективності експлуатації та обслуговування.

2.2 Обрані технічні рішення

Контролер Modicon M241 було обрано, як основний обчислювальний вузол автоматизованої системи з наступних причин:

– широкі комунікаційні можливості: наявність вбудованих інтерфейсів Ethernet, USB, CANopen і Serial (RS-232/485) дозволяє легко інтегруватися з

периферійними пристроями, датчиками, приводами, НМІ, а також із зовнішніми системами управління за потреби;

- висока швидкодія: контролер має потужний процесор і здатний обробляти велику кількість дискретних сигналів у реальному часі, що критично важливо при швидкісній подачі вантажу;

- безкоштовне середовище програмування SoMachine (нині EcoStructure Machine Expert Basic) – дозволяє реалізовувати алгоритми в стандарті ІЕС 61131-3 (LD, ST, FBD тощо), використовуючи функціональні блоки, а також засоби онлайн-моніторингу та діагностики;

- гнучкість та масштабованість – можливість підключення модулів вводу/виводу серії ТМЗ дозволяє адаптувати конфігурацію контролера до будь-якої топології системи до 2400 розподілених цифрових та / або аналогових входів / виходів.

Як інтерфейс для взаємодії оператора з системою було обрано панель НМІ серії Harmony GTO, яка забезпечує:

- повну апаратну та програмну сумісність з контролерами Modicon;
- швидку розробку інтерфейсів у середовищі Vijeo Designer, з використанням шаблонів, графіків, повідомлень про помилки, логів подій;
- відображення стану системи в реальному часі: статус датчиків, швидкість конвеєра, відстань між посилками, візуалізація заторів;
- гнучке керування параметрами: оператор може змінювати робочі режими, налаштовувати відстань між посилками, запускати/зупиняти окремі потоки;
- підтримка протоколів Modbus TCP та Ethernet/IP, що забезпечує стабільну комунікацію з PLC без використання додаткового обладнання.

Також для забезпечення точного контролю положення вантажу на конвеєрних лініях та точного процесу злиття, на окремих ділянках було впроваджено енкодери, які генерують 1 імпульс на кожен метр переміщення стрічки. Це дозволяє системі з високою точністю визначати положення посилки в просторі, синхронізуючи подачу з інших потоків.

В якості основних засобів виявлення об'єктів було обрано оптичні датчики, які забезпечують надійне розпізнавання посилок незалежно від їхнього кольору та форми, а також мають швидкий час реакції, що є критичним при великих швидкостях переміщення.

Для реалізації функції зміни швидкості конвеєрної стрічки та адаптації до змін у навантаженні було застосовано частотний перетворювач, який підтримує керування через протокол Modbus TCP та аналогові входи для зміни швидкості в режимі реального часу. Крім того, пристрій оснащено конфігурованими виходами, які дозволяють передавати сигнали про стан або аварії до PLC або НМІ, що підвищує загальну інформативність та діагностичність системи.

2.3 Розрахунок параметрів автоматизованої лінії

Щоб отримання продуктивності в 7000 вантажів на годину при середніх габаритах довжини вантажу в 60 см і для гарантованого збросу потрібна відстань між посилками в 30 см. Для початку розрахуємо загальну довжину посилок яку повинні пройти 7000 посилок з урахуванням інтервалу.

Довжина, яку займає одна посилка з інтервалом, дорівнює сумі довжини вантажу і інтервалу $L_{\text{одного місця}} = 0,6 + 0,3 = 0,9$ м.

Загальна довжина 7000 посилок на годину: $L_{\text{загальна}} = 7000 \times 0,9 = 6300$ м/год.

1 година = 3600 секунд, тому $V = \frac{6300}{3600} \approx 1,75$ м/с.

Отже, щоб забезпечити продуктивність 7000 посилок/год, при середньому розмірі посилки 60 см і інтервалі 30 см, швидкість лінії має становити не менше 1,75 м/с. Для стабільного результату швидкість буде встановлена на 2 м/с.

З'єднання потоків в 1 відбувається під кутом в 30 градусів. Потрібно розрахувати швидкість, з якою будуть крутитись конвеєри, що стоять під кутом до центральної лінії, щоб проекція їх швидкості на центральну лінію відповідала 2 м/с.

$$\text{Швидкість бокового конвеєра: } V = \frac{2}{\cos(30^\circ)} = 2/\frac{\sqrt{3}}{2} \approx 2,31\text{м/с.}$$

Конвеєри будуть розміщуватись за такою схемою (рис. 2.1).

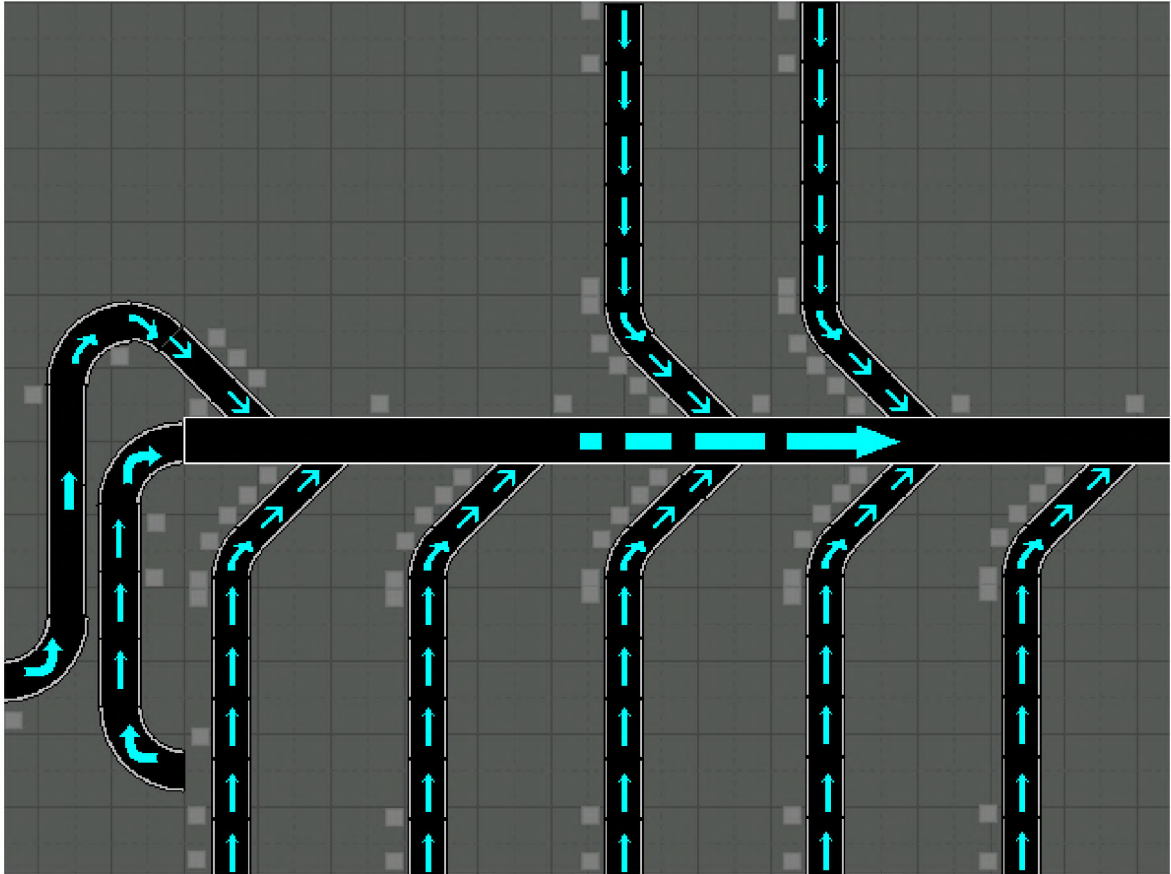


Рисунок 2.1 – Схема розміщення конвеєрів

Така конфігурація розміщення конвеєрів, з підведенням бічних потоків під кутом до центральної транспортної лінії, зумовлена як вимогами технічного завдання замовника, так і конструктивними особливостями приміщення логістичного терміналу. Також на конвеєрах одного потоку швидкість відрізняється для забезпечення розриву між посилками ще на етапі їх підвозу до розгонного конвеєра і забезпечення уникнення ситуації (за умови викладки на перший конвеєр з розривом мінімум в 2 см), коли 2 об'єкти виїжджають одночасно на центральну лінію з одного потоку.

Кожна з стрілок відповідає реальному конвеєру. Отже, на більшості вхідних потоків, що йдуть з рамп по 8 конвеєрів (рис. 2.2).



Рисунок 2.2 – Нумерація конвеєрів на вхідних потоках

Конвеєр під номером 1 при роботі потоку завжди ввімкнений, це зроблено для того щоб, в будь якого випадку, вантаж вливався в центральний потік з вірною швидкістю.

Другий конвеєр, є розгонним конвеєром на ньому посилка вимірюється, може зупинятись очікуючи свою чергу, для виміру розміру на ньому встановлений енкодер. Ці конвеєри мають 2 швидкісні режими, один це 2,3 м/с, інший 1,15 м/с, для можливості влиття посилки в центральну лінію. Ці 2 режими при швидкості 2 м/с дають більшу варіативність для влиття вантажу, наприклад, коли при швидкості 2,3 м/с для нього немає місця, бо вже їде щось, швидкість скидається і перевіряється чи буде місце для вкидання в таких умовах, якщо і в цьому випадку не буде місця то розгонний зупиниться і вантаж почне шукати собі місце в потоці стоячи на ньому.

Третій конвеєр це поворотний конвеєр, у випадку, коли на розгонному вже чекає вантаж на місце, вантаж, що приїде і зупиниться на поворотному це ще одне місце очікування.

Четвертий конвеєр це конвеєр де попередньо вимірюється розмір вантажу, він теж може виступати, як окреме місце для його зупинки на ньому також встановлений енкодер.

Конвеєри 5,6,7 керуються однаково, так наче це один конвеєр. Використано три конвеєри для забезпечення додаткового розриву за рахунок різниці швидкостей на кожному з цих конвеєрів.

І восьмий конвеєр це конвеєр, на який викладають вантаж це може бути, як звичайний конвеєр так і телескопічний, що може видвигатись всередину машини з вантажем.

На конвеєрі під номером 1 відсутні датчики. В випадку аварійної зупинки і подальшого старту, система буде запускати ці конвеєри з урахуванням можливості наявності на них вантажу середнього розміру. В інших випадках вантаж тут не зупинятиметься.

На 2 конвеєрі, що є розгонним встановлено 3 датчики по 1 на початку і на кінці і один перпендикулярно до центральної лінії. Ці датчики використовуються для виміру вантажу, для розуміння, що вантаж потрапив на / покинув розгонний конвеєр. Перпендикулярний датчик потрібен для контролю наявності посилки на конвеєрі також він, як і інші використовується при калібруванні лінії.

На третьому конвеєрі є 1 датчик, що використовується для відслідковування наявності на ньому вантажу.

На четвертому стоять 2 датчики які використовуються для виміру вантажу. Також наявність двох датчиків потрібно для точної зупинки вантажу на краю конвеєра і запобігає закидуванню другого об'єкта на поворотний.

На конвеєрах 5,6,7 датчики відсутні.

На 8 конвеєрі на кінці знаходиться 1 датчик який використовується для розриву між посилками. Якщо час з моменту останнього спрацювання менший ніж налаштовано то конвеєр зупиниться і поїде це гарантує розрив між вантажем який далі буде лише збільшуватись за рахунок різниці швидкостей. Також на цьому конвеєрі є датчик який використовується для роботи конвеєра до моменту

поки вантаж не перетне датчик в випадку, якщо лінія заповнена і всі наступні конвеєри вже зупинені. Схема розміщення датчиків (рис. 2.3).

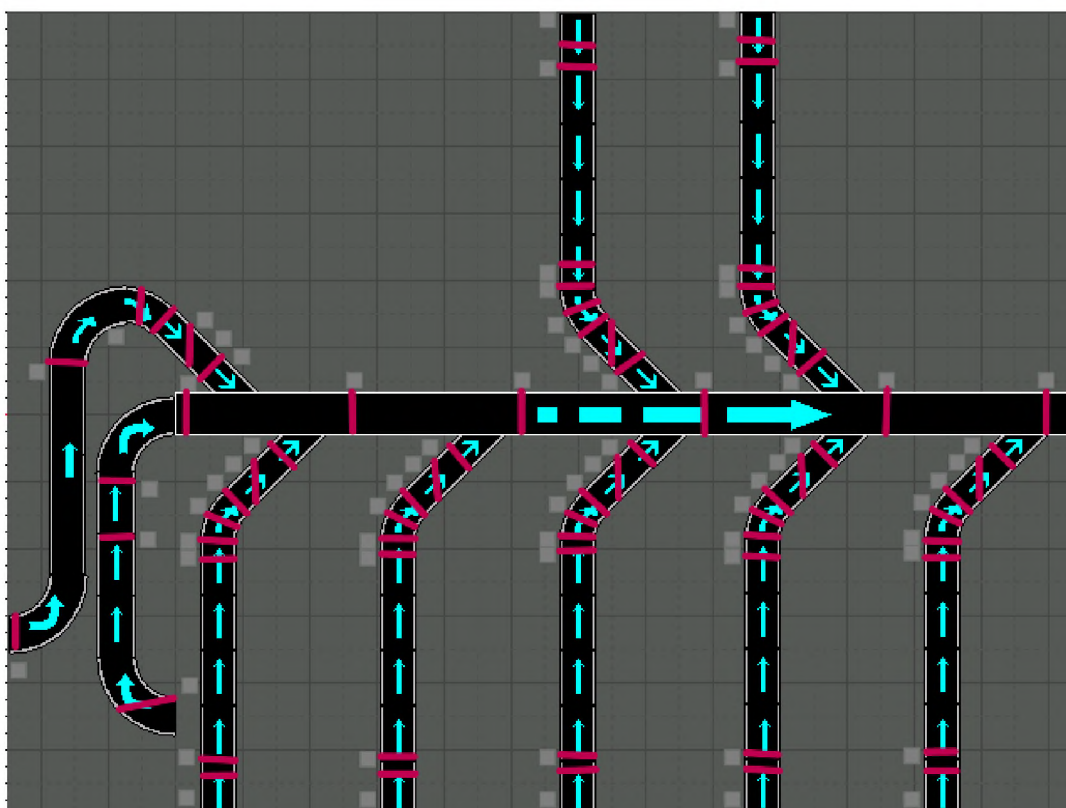


Рисунок 2.3 – Схема розміщення датчиків

Така схема розміщення датчиків забезпечує достатньо інформації для забезпечення коректної роботи логіки злиття. Деякі датчики також використовуються для відстеження заторів на лінії, а також частина датчиків допомагає забезпечити розрив між посилками на етапі їх під'їзду до центральної лінії злиття. Енкодер з датчиками на центральній лінії забезпечують точне відстеження посилки.

Для підключення і регулювання такої кількості датчиків і енкодерів і керуванням конвеєрами використано такі модулі входів/виходів (рис. 2.4). Використано 24 входа на самому контролері + 3 модуля входів на 32 входа + 16 виходів на контролері + 2 модуля на 32 вихода і 3 модуля аналогових виходів. Оскільки до контролера можна напряму під'єднати лише 7 модулів було встановлено модулі розширення приймач і передавач які надають можливість до

встановити ще 7 модулів входів виходів. Модульні входи/виходи поділяються на секції, у які можна підключати окремі модулі. Це значно розширює можливості та гнучкість системи. Користувач може обирати з—поміж модулів, доступних у виробника, і комбінувати їх довільним чином [25].

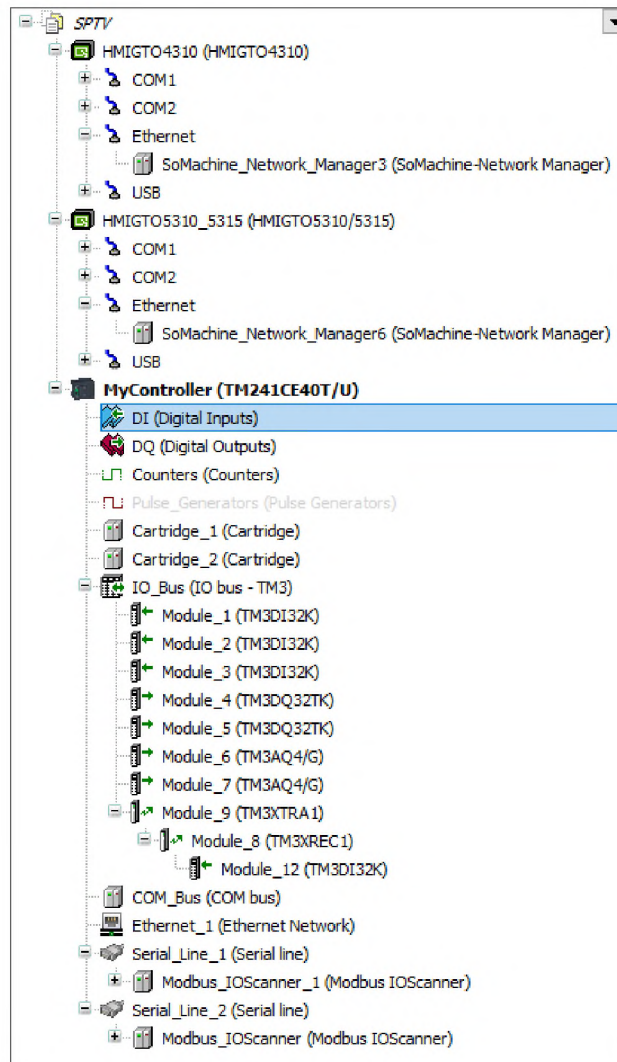


Рисунок 2.4 – Конфігурація входів і виходів

На швидкі входи контролера, яких 8 заведені сигнали з енкодерів, що розміщені на розгонних конвеєрах (конвеєр під номером 2), а також енкодер, що знаходиться на центральній лінії оскільки швидкості цих конвеєрів найбільші. На наступні 8 входів контролера заведено інші 8 енкодерів які знаходяться на четвертому конвеєрі, що використовуються для попереднього виміру вантажу для них не потрібні швидкі входи оскільки швидкість цих конвеєрів всього 1 м/с,

а енкодер має 1 імпульс на 100 см тому контролер встигне опрацювати сигнали і звичайними входами . На контролер теж заведені всі 8 датчиків, що знаходяться на центральній лінії. 16 виходів контролера задіяні на керування 3 конвеєрами, що розміщені на центральній лінії, і всіма конвеєрами з 2 потоків. На кожен потік іде 6 сигналів 1 для команди пуск на 8 конвеєр 2 на команду стоп для цього ж 8 конвеєру 3 сигнал йде сигнал для запуску конвеєрів 7,6,5 ці конвеєри запускаються одним сигналом причина чому їх саме 3 в тому, що швидкості на цих конвеєрах по руху вантажу збільшується відповідно розрив між вантажем додатково збільшується, 4 сигнал на потік для керування 4 конвеєром, 5 для керування 3 поворотним конвеєром, 6 для керування 2 конвеєром (розгонним).

На перший модуль, що є модулем входів заведено 32 сигнали з датчиків 2 датчика затора на центральній стрічці, а також датчики з 3 вхідних потоків повністю, а саме 1, 2, 3 і 4 частково, на 2 модуль входів заведено датчики з 4,5,6 і 7 потоків. На 3 модуль входів заведені датчики 8 потоку, а також сигнали готовності частотних перетворювачів 8 потоків і центральної лінії, також сюди заведено перемикач робота/наладка, сигнал з доп, контактів пускача, що відповідає за наявність силової частини, а також сигнал роботи сортувальної лінії, що стоїть далі.

На модулі виходів йдуть сигнали керування конвеєрами 3,4,5,6,7,8 потоків, а також вихідні сигнали аварія лінії злиття, сигнал, що лінія заснула, і сигнал готовності до роботи, також сигнали для керування світлосигнальною колоною.

Далі йдуть модулі аналогових виходів до цих модулів під'єднані центральний конвеєр і розгонні на всіх вхідних потоках це створено для регулювання частоти на частотних перетворювачах і відповідно швидкості відповідних конвеєрів.

Для реалізації логіки роботи функції зміни швидкості частотні перетворювачі всіх перших конвеєрів і третіх (поворотних) під'єднані до контролера по протоколу Modbus через дві Serial_Line (рис. 2.5).

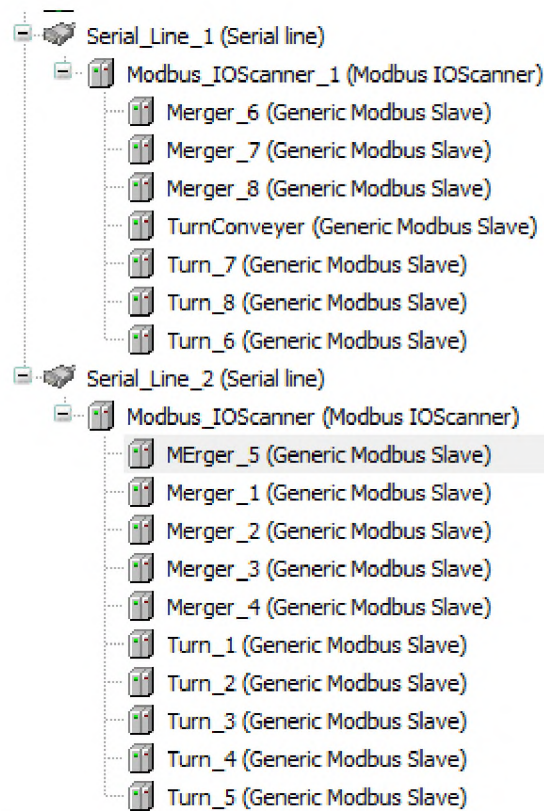


Рисунок 2.5 – Конфігурація Serial_Line

До Serial_Line_1 підключені частотні перетворювачі 6, 7 та 8 потоків надсилання команд відбувається через Modbus_IOScanner (рис. 2.6).

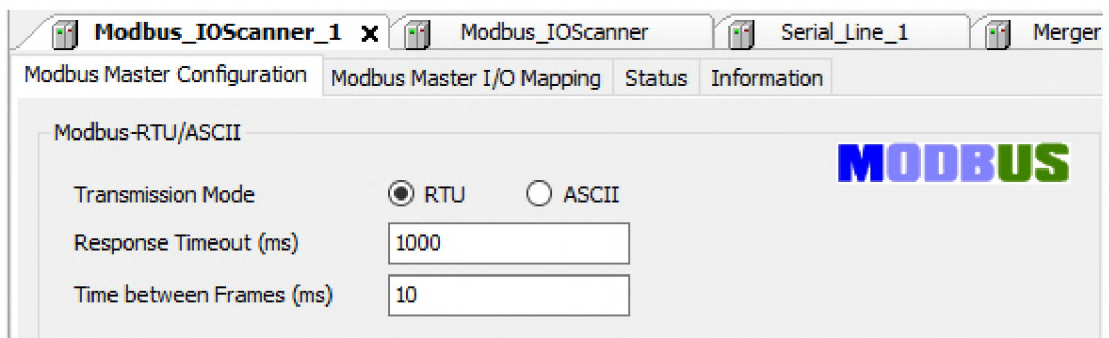


Рисунок 2.6 – Конфігурація Modbus_IOScanner

Ці налаштування аналогічні і для іншого Modbus_IOScanner. Далі вже безпосередньо ідуть налаштування кожного частотного перетворювача, як Slave девайса. Наприклад, це загальні налаштування для зв'язку з частотним перетворювачем з адресою 6, час для відповіді встановлений на 1 с (рис. 2.7).

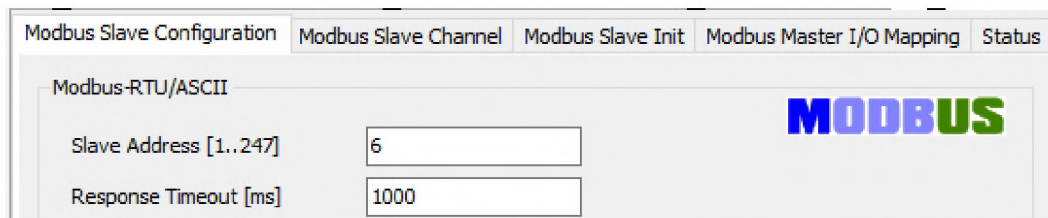


Рисунок 2.7 – Зовнішній вигляд Modbus Slave Configuration

Ось так виглядає конфігурація безпосередньо для запису в відповідний регістр частотного перетворювача значення частоти (рис. 2.8).

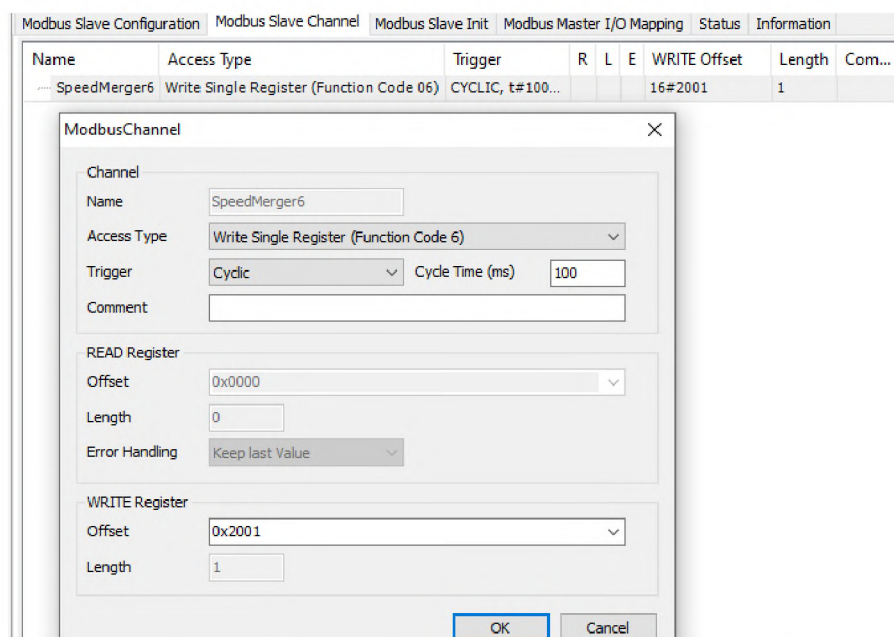


Рисунок 2.8 – Зовнішній вигляд налаштування каналу для Modbus Slave Configuration

Контролер періодично надсилає на частотний перетворювач команду для запису значення частоти. Значення, що записується дорівнює відповідній змінній в програмі і присвоюється ось так (рис. 2.9).

Завдяки такій конфігурації вдається досягти поставлено задачі для регулювання швидкості без встановлення додаткових модулів аналогових виходів, а саме 5 штук оскільки по протоколу передаються частоти для 17 частотних перетворювачів, а до одного модуля аналогових виходів можна підключити лише 4 пристрої.

Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
Application.SPEED_MERGER_6		SpeedMerger6	%QW14	ARRAY [0..0] OF WORD			Write Single Register
		SpeedMerger6[0]	%QW14	WORD			WRITE 16#2001 (=08193)
		Bit0	%QB28.0	BOOL	FALSE		
		Bit1	%QB28.1	BOOL	FALSE		
		Bit2	%QB28.2	BOOL	FALSE		
		Bit3	%QB28.3	BOOL	FALSE		
		Bit4	%QB28.4	BOOL	FALSE		
		Bit5	%QB28.5	BOOL	FALSE		
		Bit6	%QB28.6	BOOL	FALSE		
		Bit7	%QB28.7	BOOL	FALSE		
		Bit8	%QB29.0	BOOL	FALSE		
		Bit9	%QB29.1	BOOL	FALSE		
		Bit10	%QB29.2	BOOL	FALSE		
		Bit11	%QB29.3	BOOL	FALSE		
		Bit12	%QB29.4	BOOL	FALSE		
		Bit13	%QB29.5	BOOL	FALSE		
		Bit14	%QB29.6	BOOL	FALSE		
		Bit15	%QB29.7	BOOL	FALSE		

Рисунок 2.9 – Зовнішній вигляд I/O Mapping для Modbus Slave Configuration

Розгонні конвеєра підключені через аналогові виходи по причині того, що вони мають 2 швидкісні режими роботи в будь якому швидкісному режимі роботи центральної стрічки. І плюсом підключення аналоговим виходом є швидкість передачі і вона стабільна. А в випадку, якщо частотний перетворювач керується по Modbus є певна різниця в залежності в який момент циклу передачі змінилась частота, а також у випадку виникнення аварії на певному потоці і вимкнення частотного перетворювача то цикл автоматично збільшується на 1000 мс оскільки буде очікуватись відповідь від частотного перетворювача, що вимкнений. Такі затримки не допустимі для розгонного оскільки за 1 с при швидкості в 2 м/с вантаж проїде 2 м відповідно ніякої точності злиття. Для першого конвеєра і поворотного це не критично оскільки процес зміни швидкості на них відбувається поступово приблизно на протязі 30 с на цей час злиття припиняється. Фактично частота змінюється 1 раз на досить довгий проміжок часу і немає потреби в її постійній зміні, як для розгонного. І навіть в випадку вимкненого частотного перетворювача по причині, наприклад ТО, що відбувається на лінії то при ввімкненні за рахунок циклічності він отримає частоту і подасть сигнал готовності до роботи і вже тоді центральний контролер почне опрацьовувати і керувати даним потоком.

РОЗДІЛ 3

ЕСКІЗНИЙ ПРОЄКТ ПРОГРАМНО–АПАРАТНОГО КОМПЛЕКСУ АВТОМАТИЗАЦІЇ ПРОЦЕСУ ЗЛИТТЯ ПОСИЛОК НА ЛОГІСТИЧНИХ ТЕРМІНАЛАХ

3.1 Реалізація логіки

У Task Configuration було створено нове завдання з найвищим пріоритетом. Тригером для його виконання є зовнішній імпульс, а саме – сигнал з енкодера, який розміщено на центральній лінії. Цей сигнал заведений на швидкий вхід контролера (IO) (рис. 3.1).

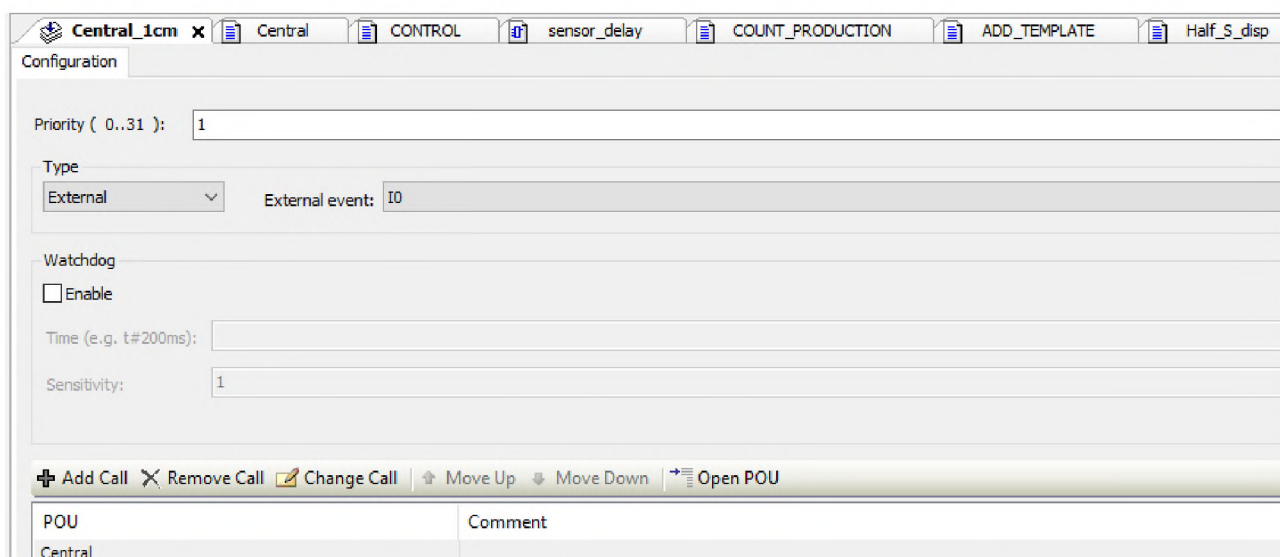


Рисунок 3.1 – Конфігурація Task в SoMachine V4.3

Таким чином, при кожному переміщенні вантажу на 1 см (один імпульс енкодера), буде виконуватись POU (Program Organization Unit), закріплений за цим завданням.

Створено додаткове циклічне завдання з низьким пріоритетом, для виконання допоміжних операцій, завдання буде циклічно виконуватись з інтервалом у 500 мс (рис. 3.2). Цей інтервал означає, що завдання буде запускатись двічі на секунду, що є цілком достатнім для операцій, які не

вимагають високої частоти опитування або обробки. Пріоритетність цього завдання встановлена найнижчою серед доступних рівнів. Створення такого Task є розвантаження основного циклу програми за рахунок винесення в нього дій, які не критичні по часу, виконуються рідко або не вимагають високої частоти оновлення. Завдяки низькому пріоритету, це завдання не впливає на виконання більш важливих та частих задач, і виконується лише тоді, коли система має достатньо вільних ресурсів.

POU	Comment
Half_5_disp	

Рисунок 3.2 – Конфігурація Task в SoMachine V4.3

Крім допоміжного Task, створено основне циклічне завдання, яке відповідальне за виконання основної логіки програми (рис. 3.3). Пріоритетність цього Task встановлена на середньому рівні серед доступних це забезпечує стабільну та достатньо швидку роботу головного циклу, одночасно не блокує інші важливі завдання, такі як ті, що працюють на імпульсах з енкодера. Інтервал виконання головного Task – 20 мс, тобто він запускається 50 разів на секунду. Це забезпечує достатню реактивність для управління процесами на лінії.

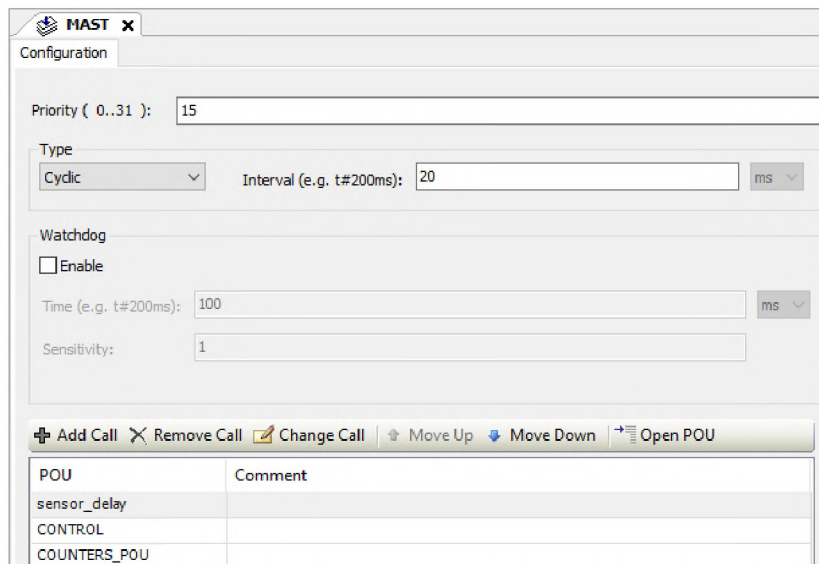


Рисунок 3.3 – Конфігурація Task в SoMachine V4.3

COUNTERS POU (додаток А) реалізовано мовою FBD (Function Block Diagram). У цій частині програми використовуються стандартні функціональні блоки SEC_HSC.HSCSimple_M241 для роботи з високошвидкісними лічильниками, які фіксують імпульси з енкoderів на входах ПЛК. Входи блоку, Enable активує підрахунок, Sync – обнуляє лічильник, CurrentValue – відображає поточне значення. Усього задіяно 15 лічильників (максимум 16 підтримується апаратно, але один вхід використано під інше завдання). Блоки HscSimple_1–HscSimple_8 обробляють сигнали з енкoderів на розгінних конвеєрах, а HscSimple_9–HscSimple_15 – з основних конвеєрів перед поворотним, що дає змогу точніше визначати розміри вантажу. Поточні значення лічильників зберігаються у масивах COUNTER_LINE[] і COUNTER_PREV[].

У головному циклі (Mast) виконується POU sensor_delay (рис. 3.4), реалізований у вигляді функціональних блоків. Його призначення – відображення стану датчиків на НМІ з затримкою, щоб уникнути хибних короткочасних спрацювань.

Для цього використовується таймер затримки вимкнення TOF, а також логічні елементи AND та OR. Поведінка датчиків залежить від режиму DELAY_MODE:

- Якщо DELAY_MODE активний, спрацювання сенсора затримується на час DELAY_INDICATE, і вихід SENSOR_ARRAY_D[1,1] утримується до завершення таймера TOF.
- Якщо DELAY_MODE вимкнений, сигнал з датчика SENSOR[1,1] одразу подається на вихід.

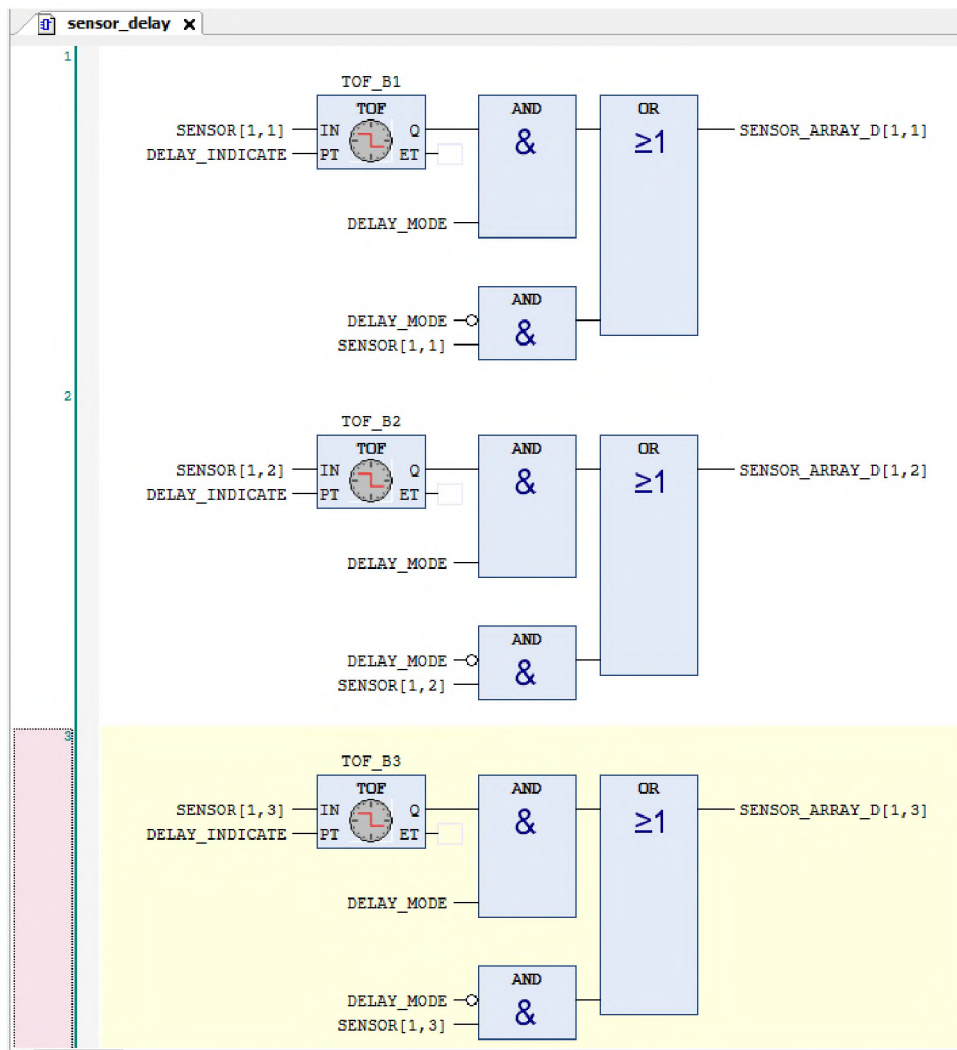


Рисунок 3.4– Код sensor_delay в SoMachine V4.3

І остання ROU в основному циклі MAST це CONTROL. Повний код (додаток Б). Цю ROU реалізовано мовою програмування Structured Text (ST), є центральною частиною логіки автоматизованої лінії. У ньому реалізовані всі основні процеси керування – від обробки сигналів сенсорів до підрахунку продуктивності та логіки сортування вантажів.

1. Скидання лічильників

На початку виконання відбувається обнулення значень лічильників, отриманих із блоку COUNTERS_POU. Це забезпечує коректну роботу системи при старті або перезапуску.

2. Обробка станів сенсорів

Викликається функція READ_COMPARE_INPUT(), яка виконує аналіз змін сигналів на всіх сенсорах системи. Обробка побудована на двовимірному масиві (8 потоків × 12 сенсорів), що дозволяє масштабовано відстежувати стан кожного датчика.

Основні завдання функції:

- виявлення зміни стану сенсора (з 0 на 1 або навпаки);
- фіксація переднього (RISING) та заднього (FALLING) фронтів сигналів;
- встановлення відповідних логічних прапорців (SENSOR_RISE_FLAG, SENSOR_FALL_FLAG);
- контроль за тим, щоб обробка фронту відбулася лише один раз під час зміни стану (через *_PERMITTED).

Це забезпечує фільтрацію шумів та точну фіксацію подій.

3. Підрахунок продуктивності

Для контролю ефективності лінії реалізовано підрахунок кількості об'єктів, що проходять повз сенсор SENSOR_FALL[0,8] (останній датчик на центральній стрічці). Результати виводяться у вигляді:

- кількості за останні 10 секунд;
- кількості за хвилину.

4. Скидання статистики помилок

Передбачено обнулення статистики щодо помилок влиття вантажів на стрічку. Статистика помилок ведеться для різних режимів:

- після зупинки на розгінному конвеєрі;
- без зупинки на розгонному, роботі у потоці.

5. Режим налаштування та калібрування

Програма перевіряє, чи активовано режим налаштування. Якщо так, виконуються дії, пов'язані з функціями ADJUSTMENT або ADJUST_INFEED_FUN:

- калібрування потоків;
- визначення положення сенсорів на центральній стрічці;
- збереження та відновлення параметрів для різних режимів (в залежності від швидкості);
- доступ з НМІ для ручного керування виходами.

Це спрощує запуск і обслуговування системи.

6. Масив позицій об'єктів

Для управління розміщенням вантажів використовується масив типу Byte[3300]:

- 3000 байтів відповідають 30 метрам центральної лінії (1 байт = 1 см);
- 300 байтів – для об'єктів, що повертаються після повторного сортування.

Цей масив дозволяє точно моделювати положення кожного вантажу на стрічці, уникати конфліктів при подачі нових об'єктів та забезпечувати рівномірне завантаження потоків.

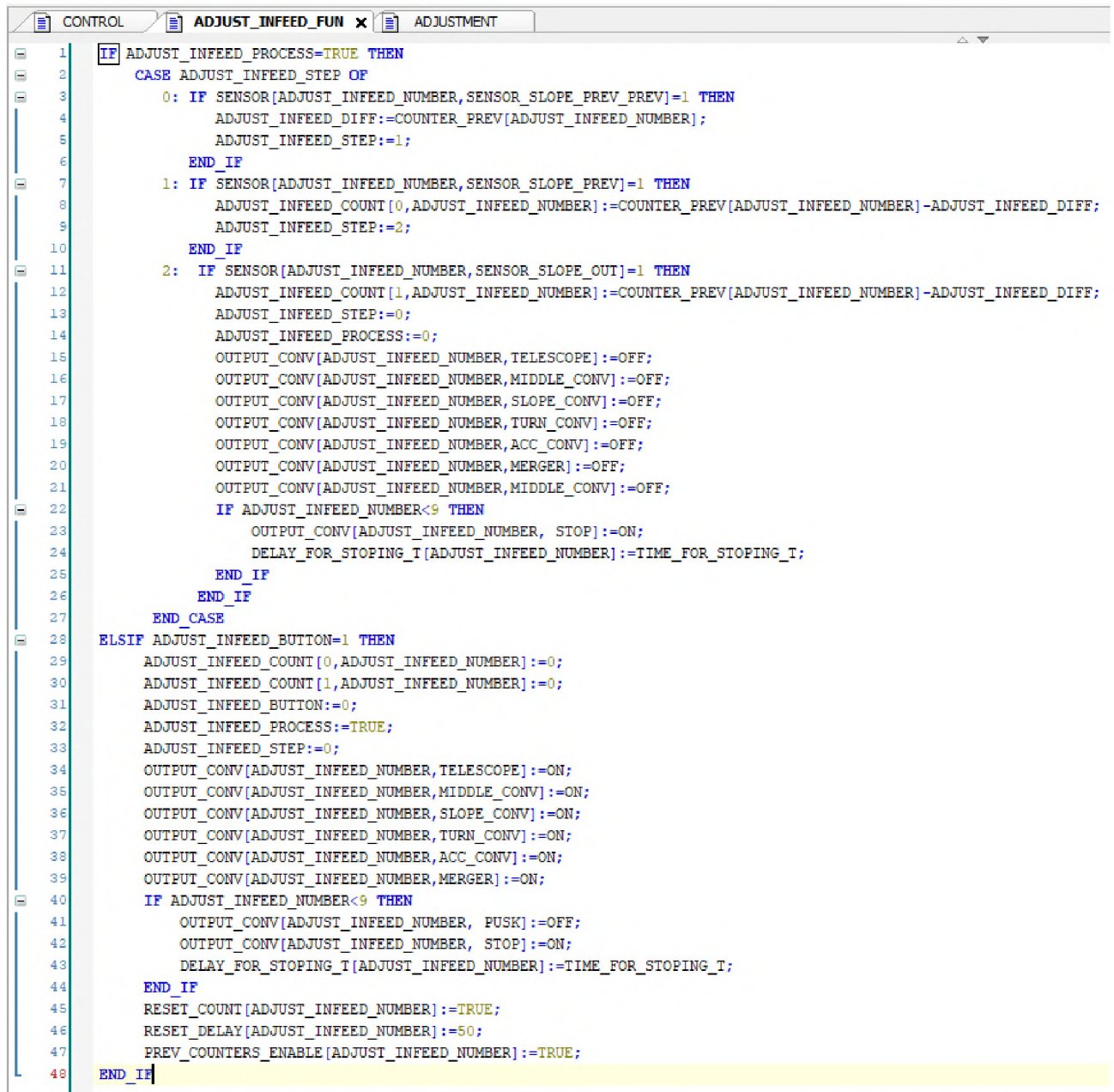
Функція ADJUST_INFEED_FUN (рис. 3.5), що викликається в POU CONTROL в відповідає за покрокову процедуру калібрування однієї з вхідних ліній (потоків) транспортувальної системи. Її логіка ділиться на дві частини.

Перша частина – покроковий процес калібрування (якщо ADJUST_INFEED_PROCESS = TRUE):

- Крок 0: очікується перше спрацювання сенсора, зчитується значення лічильника, яке фіксується як базовий розмір об'єкта (вимірювання проводиться на 4-му конвеєрі).

- Крок 1: очікується друге спрацювання сенсора, після чого обчислюється перша відстань – час проходження об'єкта до перпендикулярного датчика на розгонному конвеєрі.

– Крок 2: очікується третє спрацювання (задній фронт сигналу на центральній лінії). Зберігається друга відстань – кількість імпульсів до виходу на центральну стрічку. Після цього процес калібрування завершується, усі конвеєри зупиняються.



```
1  IF ADJUST_INFEED_PROCESS=TRUE THEN
2  CASE ADJUST_INFEED_STEP OF
3    0: IF SENSOR[ADJUST_INFEED_NUMBER,SENSOR_SLOPE_PREV_PREV]=1 THEN
4      ADJUST_INFEED_DIFF:=COUNTER_PREV[ADJUST_INFEED_NUMBER];
5      ADJUST_INFEED_STEP:=1;
6    END_IF
7    1: IF SENSOR[ADJUST_INFEED_NUMBER,SENSOR_SLOPE_PREV]=1 THEN
8      ADJUST_INFEED_COUNT[0,ADJUST_INFEED_NUMBER]:=COUNTER_PREV[ADJUST_INFEED_NUMBER]-ADJUST_INFEED_DIFF;
9      ADJUST_INFEED_STEP:=2;
10   END_IF
11   2: IF SENSOR[ADJUST_INFEED_NUMBER,SENSOR_SLOPE_OUT]=1 THEN
12     ADJUST_INFEED_COUNT[1,ADJUST_INFEED_NUMBER]:=COUNTER_PREV[ADJUST_INFEED_NUMBER]-ADJUST_INFEED_DIFF;
13     ADJUST_INFEED_STEP:=0;
14     ADJUST_INFEED_PROCESS:=0;
15     OUTPUT_CONV[ADJUST_INFEED_NUMBER,TELESCOPE]:=OFF;
16     OUTPUT_CONV[ADJUST_INFEED_NUMBER,MIDDLE_CONV]:=OFF;
17     OUTPUT_CONV[ADJUST_INFEED_NUMBER,SLOPE_CONV]:=OFF;
18     OUTPUT_CONV[ADJUST_INFEED_NUMBER,TURN_CONV]:=OFF;
19     OUTPUT_CONV[ADJUST_INFEED_NUMBER,ACC_CONV]:=OFF;
20     OUTPUT_CONV[ADJUST_INFEED_NUMBER,MERGER]:=OFF;
21     OUTPUT_CONV[ADJUST_INFEED_NUMBER,MIDDLE_CONV]:=OFF;
22     IF ADJUST_INFEED_NUMBER<9 THEN
23       OUTPUT_CONV[ADJUST_INFEED_NUMBER,STOP]:=ON;
24       DELAY_FOR_STOPPING_T[ADJUST_INFEED_NUMBER]:=TIME_FOR_STOPPING_T;
25     END_IF
26   END_IF
27 END_CASE
28 ELSIF ADJUST_INFEED_BUTTON=1 THEN
29   ADJUST_INFEED_COUNT[0,ADJUST_INFEED_NUMBER]:=0;
30   ADJUST_INFEED_COUNT[1,ADJUST_INFEED_NUMBER]:=0;
31   ADJUST_INFEED_BUTTON:=0;
32   ADJUST_INFEED_PROCESS:=TRUE;
33   ADJUST_INFEED_STEP:=0;
34   OUTPUT_CONV[ADJUST_INFEED_NUMBER,TELESCOPE]:=ON;
35   OUTPUT_CONV[ADJUST_INFEED_NUMBER,MIDDLE_CONV]:=ON;
36   OUTPUT_CONV[ADJUST_INFEED_NUMBER,SLOPE_CONV]:=ON;
37   OUTPUT_CONV[ADJUST_INFEED_NUMBER,TURN_CONV]:=ON;
38   OUTPUT_CONV[ADJUST_INFEED_NUMBER,ACC_CONV]:=ON;
39   OUTPUT_CONV[ADJUST_INFEED_NUMBER,MERGER]:=ON;
40   IF ADJUST_INFEED_NUMBER<9 THEN
41     OUTPUT_CONV[ADJUST_INFEED_NUMBER,PUSK]:=OFF;
42     OUTPUT_CONV[ADJUST_INFEED_NUMBER,STOP]:=ON;
43     DELAY_FOR_STOPPING_T[ADJUST_INFEED_NUMBER]:=TIME_FOR_STOPPING_T;
44   END_IF
45   RESET_COUNT[ADJUST_INFEED_NUMBER]:=TRUE;
46   RESET_DELAY[ADJUST_INFEED_NUMBER]:=50;
47   PREV_COUNTERS_ENABLE[ADJUST_INFEED_NUMBER]:=TRUE;
48 END_IF
```

Рисунок 3.5 – Код ADJUST_INFEED_FUN в SoMachine V4.3

Друга частина – ініціалізація калібрування (при натисканні кнопки ADJUST_INFEED_BUTTON):

– вмикаються необхідні конвеєри;

- скидаються лічильники;
- встановлюється затримка (наприклад, 50 циклів) для стабілізації;
- активується дозвіл на зчитування попередніх значень;
- потік калібрується кілька разів (мінімум тричі в кожному з трьох режимів – загалом дев'ять разів);
- система розраховує середнє значення для кожного режиму.

Функція ADJUSTMENT, що викликається в POU, реалізує логіку автоматичного калібрування вхідних потоків у трьох режимах. Вона працює в двох основних станах: у режимі очікування (ADJUST_IN_PROCESS = FALSE) система обробляє натискання кнопок – запуску, скидання та збереження; у режимі активного калібрування (ADJUST_IN_PROCESS = TRUE) – виконується фактичний процес налаштування у кілька кроків для кожної окремої лінії (ADJUST_LINE) згідно з поточним етапом (ADJUST_STEP).

При натисканні кнопки запуску калібрування (ADJUST_BUTTON = TRUE) система ініціалізується: обнуляються кроки, попередні лічильники, масиви вимірювань, а також статистика по поточній лінії. Основна логіка далі виконується через оператор CASE за номером лінії. Для центральної лінії (0) алгоритм найпростіший – запускаються два конвеєри, виконується затримка для виходу на швидкість, вмикається сигналізація, і процес завершується при досягненні дев'ятого кроку або при скиданні. Для ліній 1–8 логіка складніша: поетапно вмикаються центральна та вхідна стрічки, очікується набір швидкості, далі запускаються всі конвеєри відповідного потоку, встановлюється швидкість розгону, після чого всі конвеєри вимикаються, записуються вимірювання, рахується середнє значення, і оновлюється статистика.

Усі вимірювання зберігаються в змінних: ADJUST_COUNT для кожної спроби, ADJUST_SUM для підсумку, ADJUST_AVERAGE для середнього. Лічильник спроб зменшується при натисканні кнопки скидання (ADJUST_RESET_BUTTON), останній результат видаляється з суми, а середнє оновлюється. Кнопка збереження (ADJUST_SAVE_BUTTON) фіксує середні значення. Для центральної лінії ці значення записуються як еталонні позиції

сенсорів (SENSOR_PLACE), для інших ліній – обчислюються відстані (LINE_GAP), коефіцієнти перерахунку (LINE_KOEF) та параметри зупинки потоку (LINE_STOP_GAP).

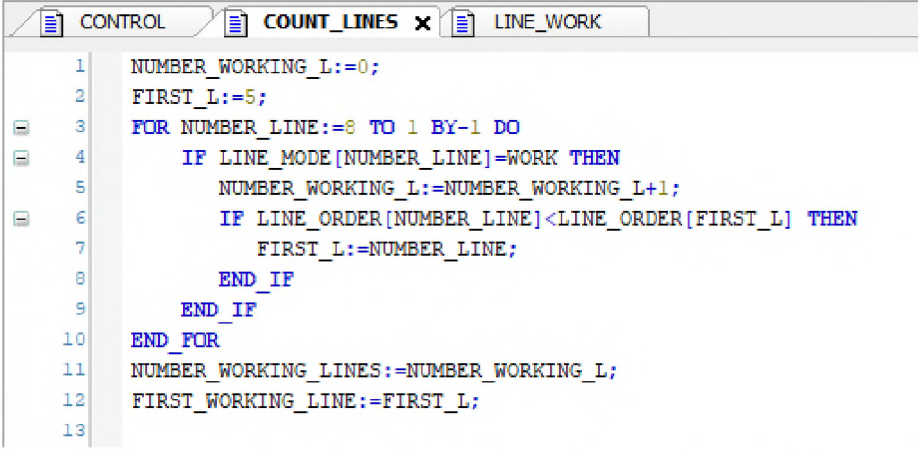
Таким чином, функція ADJUSTMENT дозволяє точно й автоматично налаштувати роботу системи транспортування для кожної лінії, з урахуванням її особливостей і обраного режиму.

Другий режим в POU CONTROL, це звичайний робочий режим. Режими в CASE MODE_WORK OF:

- ALARM – аварійне завершення при ALARM_JAM сенсорах.
- STOP – чекає готовності, запускає виходи по сенсорам.
- READY_FOR – перевірка на готовність, переходить у SWITCHING_ON.

WORK, SLEEP – робочі стани. SWITCHING_ON – стартова пауза перед запуском. Всі переходи умовні, залежать від сигналів CONTACTOR (готовність силової частини), READY_INVT (готовність частотного перетворювача центральної лінії), WORK_SWITCH (чи не переведено в ручний режим лінію), SENSOR (чи не спрацьовані датчики заторів на лінії), SORT_SYSTEM_WORK (чи є головний сигнал роботи всієї сортувальної системи).

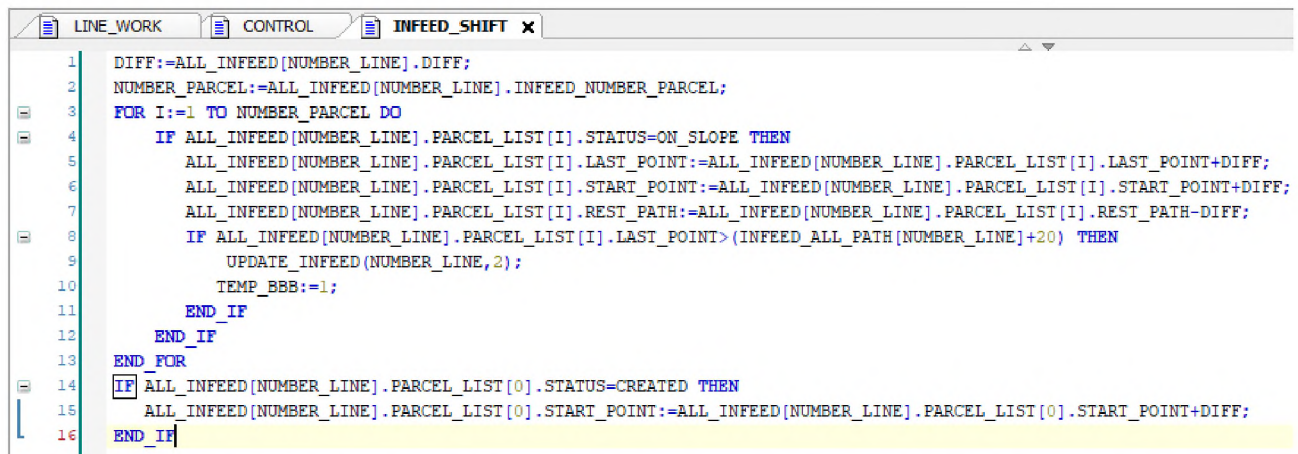
В режимах WORK, SLEEP викликаються функції COUNT_LINES(). Функція COUNT_LINES() виконує простий аналіз підраховує скільки потоків працює і який з потоків працює першим (рис. 3.6).



```
1  NUMBER_WORKING_L:=0;
2  FIRST_L:=5;
3  FOR NUMBER_LINE:=8 TO 1 BY-1 DO
4      IF LINE_MODE[NUMBER_LINE]=WORK THEN
5          NUMBER_WORKING_L:=NUMBER_WORKING_L+1;
6          IF LINE_ORDER[NUMBER_LINE]<LINE_ORDER[FIRST_L] THEN
7              FIRST_L:=NUMBER_LINE;
8          END_IF
9      END_IF
10 END_FOR
11 NUMBER_WORKING_LINES:=NUMBER_WORKING_L;
12 FIRST_WORKING_LINE:=FIRST_L;
13
```

Рисунок 3.6 – Код функції COUNT_LINES в SoMachine V4.3

Далі йде виклик функції INFEED_SHIFT (рис. 3.7).



```
1 DIFF:=ALL_INFEED[NUMBER_LINE].DIFF;  
2 NUMBER_PARCEL:=ALL_INFEED[NUMBER_LINE].INFEED_NUMBER_PARCEL;  
3 FOR I:=1 TO NUMBER_PARCEL DO  
4     IF ALL_INFEED[NUMBER_LINE].PARCEL_LIST[I].STATUS=ON_SLOPE THEN  
5         ALL_INFEED[NUMBER_LINE].PARCEL_LIST[I].LAST_POINT:=ALL_INFEED[NUMBER_LINE].PARCEL_LIST[I].LAST_POINT+DIFF;  
6         ALL_INFEED[NUMBER_LINE].PARCEL_LIST[I].START_POINT:=ALL_INFEED[NUMBER_LINE].PARCEL_LIST[I].START_POINT+DIFF;  
7         ALL_INFEED[NUMBER_LINE].PARCEL_LIST[I].REST_PATH:=ALL_INFEED[NUMBER_LINE].PARCEL_LIST[I].REST_PATH-DIFF;  
8         IF ALL_INFEED[NUMBER_LINE].PARCEL_LIST[I].LAST_POINT>(INFEED_ALL_PATH[NUMBER_LINE]+20) THEN  
9             UPDATE_INFEED(NUMBER_LINE,2);  
10            TEMP_BBB:=1;  
11        END_IF  
12    END_IF  
13 END_FOR  
14 IF ALL_INFEED[NUMBER_LINE].PARCEL_LIST[0].STATUS=CREATED THEN  
15     ALL_INFEED[NUMBER_LINE].PARCEL_LIST[0].START_POINT:=ALL_INFEED[NUMBER_LINE].PARCEL_LIST[0].START_POINT+DIFF;  
16 END_IF
```

Рисунок 3.7 – Код функції INFEED_SHIFT в SoMachine V4.3

Ця функція відповідає за керування рухом посилок на вхідному потоці транспортної системи. На початку роботи оголошуються дві ключові змінні: DIFF, яка фіксує зміну координати посилки за один цикл (тобто, як далеко вона просунулась по стрічці), та NUMBER_PARCEL – кількість посилок, що наразі знаходяться на конкретній вхідній лінії NUMBER_LINE.

Далі функція переходить до циклу, який проходить по кожній із цих посилок. Якщо посилка перебуває на одному з активних конвеєрів (№4–7), то оновлюються її параметри: координати (START_POINT і LAST_POINT просуваються вперед), а також REST_PATH – залишкова відстань до точки зняття – зменшується. Якщо ж посилка повністю проїхала весь маршрут і її LAST_POINT вийшла за допустимі межі довжини (з запасом у 20 одиниць), викликається процедура UPDATE_INFEED – для подальшого опрацювання.

Процедура UPDATE_INFEED базується на змінній EVENT, яка виступає як перемикач, що визначає, яку саме дію слід виконати. Ця логіка дозволяє точно відслідковувати розміщення й переміщення кожної окремої посилки, обробляти появу та зникнення вантажу з-під сенсора, і своєчасно очищати або оновлювати інформацію про рух потоку, що забезпечує стабільну й точну роботу вхідної секції транспортної системи.

Присутня функція функція WAKE_SLEEP вона змінює режим роботи, як для вхідних потоків, так і для центральної лінії. Вона приймає два параметри: NUMBER_LINE (номер лінії: 1–8 або 0 для центральної) та COMMAND (WAKE, SLEEP, STOP, PUSK, TRAIN, JUMP_OFF_TRAIN). Залежно від команди, змінюються режими та активуються або вимикаються відповідні частини системи.

- WAKE переводить лінію в режим WORK, запускає вихідний конвеєр №1, ініціалізує режими, таймери та лічильники. Для 6-ї лінії задається збільшена затримка сну ($\times 3$).

- SLEEP вимикає всі конвеєри потоку та переводить їх у режим сну.

- STOP зупиняє всі конвеєри, вимикає PUSK, вмикає STOP і встановлює таймер.

Функція ERASE_COLOR(NUMBER_LINE, 3) (додаток В) викликається для візуального очищення кольору, який позначає стан вантажу на певній ділянці потоку. Це дозволяє розрізнити, з якого потоку йде вантаж, і в якому він зараз стані: створення місця, підготовка до розміщення на центральній лінії, або проходження датчика на центральній лінії.

Функція визначає номер кольору з масиву PALLETE і далі працює залежно від значення MODE:

- MODE = 0 – очищається ділянка розгінного конвеєра, скидаються змінні, пов'язані з місцем на ньому.

- MODE = 1 – очищення для поворотного конвеєра та скидання відповідних змінних.

- MODE = 2 – очищення ділянки на конвеєрі №4, згідно з довжиною першої посилки в черзі, також обнуляються відповідні змінні.

- MODE = 3 – повне очищення всіх трьох попередніх зон.

У кінці, у межах від START_POINT до LAST_POINT, усі елементи з заданим кольором замінюються на 0, що візуально означає очищення з екрана.

Повертаємось до WAKE_SLEEP наступна команда в цій функції це PUSK запускає лінію в режимі WORK, вмикає перший конвеєр і в випадку, якщо

сенсори на розгонному не активні то ініціюється процес влиття вантажу в центральну лінію з розмірами, що встановлені в параметрі UNKNOWN_PARCEL, а поворотному (конвеєр номер 3) присвоюється режим очікування, в іншому випадку режим роботи розгонного TAKE, а поворотного GIVE інші конвеєра отримують режим WAIT також затримки скидаються на початкові значення.

Наступна команда в функції WAKE_SLEEP це TRAIN це спеціальний режим роботи, всі конвеєра переводяться в режим GIVE також скидається режим Stop і вмикається PUSK.

Наступна команда це JUMP_OFF_TRAIN це для виходу з режиму TRAIN і переведення роботи лінії назад в звичайний режим WORK. І в залежності від датчиків конвеєри переключаються.

Далі йде логіка команд для центральної лінії обробляє PUSK, WAKE, SLEEP і STOP: PUSK і WAKE вмикають конвеєр, змінюють LINE_MODE на WORK і скидають таймер сну. SLEEP вимикає конвеєр і переводить лінію в режим SLEEP, STOP зупиняє конвеєр і встановлює LINE_MODE у STOP.

У функції LINE_WORK обробляються кнопки вимкнення потоку і перевіряється готовність частотних перетворювачів. При вимкненні потоку оператором або відсутності готовності відповідний вхідний потік переходить у режим STOP. Переходи в STOP для кожного конвеєра контролюються таймерами, а запуск (пуск) відбувається при розблокуванні з панелі та готовності частотників.

Далі йде нормальна робота: при спрацюванні нижнього сенсора на восьми конвеєрах через функцію WAKE_SLEEP пробуджується весь вхідний потік і центральна лінія (якщо вона спала). Перевіряються розблокування потоку і готовність частотного перетворювача.

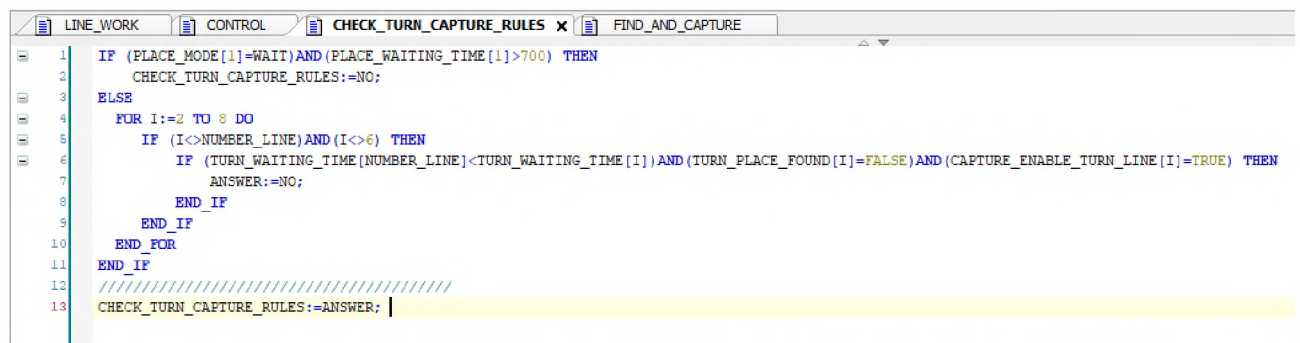
Для поворотних конвеєрів (номер 3) існує три режими:

– TAKE – очікування посилки; при виявленні вантажу на сенсорі повороту оновлюються параметри довжини, перевіряється готовність розгінного (PLACE_MODE = TAKE) і переходить у GIVE.

– GIVE – видача посилки; якщо датчики не спрацювали, повертається в TAKE.

– WAIT – очікування, поки з'явиться місце для видачі або шукається нове. Обмеження – 3000 тактів, лічильник інкрементується. Якщо місце звільнено або закінчилася затримка видачі, переходить у GIVE, таймер оновлюється. Інакше перевіряється дозвіл на створення місць (CREATE_TEMPLATE), загальний та локальний дозвіл на захоплення, і стан пошуку місця (TURN_PLACE_FOUND).

Далі викликається перевірка правил (CHECK_TURN_CAPTURE_RULES) – чи дозволено захопити це місце (рис. 3. 8). Якщо перевірка пройдена – викликається FIND_AND_CAPTURE(NUMBER_LINE, 1) (додаток Г).



```
1 IF (PLACE_MODE[1]=WAIT)AND (PLACE_WAITING_TIME[1]>700) THEN
2   CHECK_TURN_CAPTURE_RULES:=NO;
3 ELSE
4   FOR I:=2 TO 8 DO
5     IF (I<>NUMBER_LINE)AND (I<>6) THEN
6       IF (TURN_WAITING_TIME[NUMBER_LINE]<TURN_WAITING_TIME[I])AND (TURN_PLACE_FOUND[I]=FALSE)AND (CAPTURE_ENABLE_TURN_LINE[I]=TRUE) THEN
7         ANSWER:=NO;
8       END_IF
9     END_IF
10  END_FOR
11 END_IF
12 //////////////////////////////////////
13 CHECK_TURN_CAPTURE_RULES:=ANSWER; |
```

Рисунок 3.8 – Код функції CHECK_TURN_CAPTURE_RULES

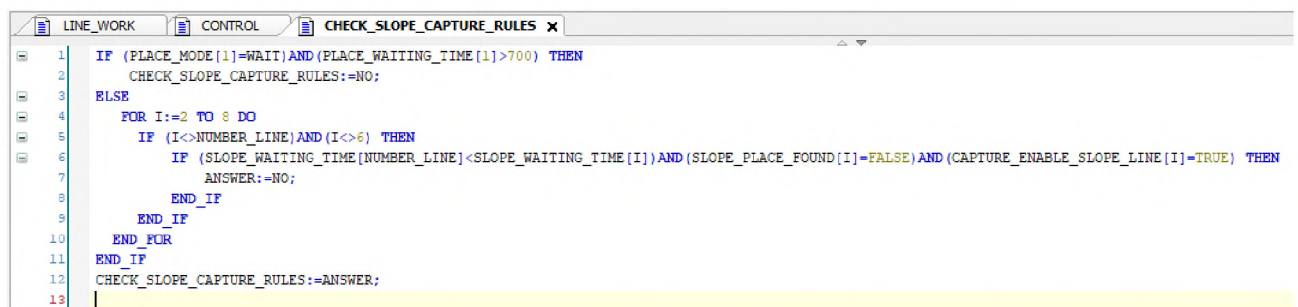
Функція CHECK_TURN_CAPTURE_RULES перевіряє, чи дозволено лінії (NUMBER_LINE) захопити місце для передачі посилки на поворотному конвеєрі. Якщо перше місце видачі в режимі WAIT і чекає понад 700 тактів, захоплення забороняється – це запобігає захопленню місця, коли вантаж на розгонному конвеєрі вже довго чекає на центральній стрічці. Також функція порівнює час очікування поточної лінії з іншими (крім 6-ї); якщо інша лінія чекає довше, не знайшла місце та має дозвіл на захоплення, то поточній лінії забороняється захоплювати місце. Це забезпечує справедливий розподіл черги між лініями.

Функція FIND_AND_CAPTURE (див. додаток Г). шукає вільне місце для вантажу на маршруті залежно від режиму MODE:

- MODE = 0 – розгінний конвеєр (2-й);
- MODE = 1 – поворотний конвеєр (3-й);
- MODE = 2 – 4-й конвеєр.

Для кожного режиму визначаються кінцева точка перевірки (TOTAL_COUNT) і довжина місця (PLACE_LENGTH) з урахуванням відступів і довжини посилки. Далі перевіряється, чи є на маршруті послідовність вільних точок у масиві PATH[], які або порожні (0), або відповідають лінії (PALLETE). Якщо знаходиться достатньо довге вільне місце (EMPTY_POINT_COUNT ≥ TOTAL_LENGTH), визначаються координати START_POINT і LAST_POINT з буфером DISTANCE, і ця ділянка маркується як зайнята відповідним кольором. Результати записуються в змінні залежно від режиму.

Також викликається функція CHECK_SLOPE_CAPTURE_RULES (рис. 3.9).



```
1 IF (PLACE_MODE[1]=WAIT) AND (PLACE_WAITING_TIME[1]>700) THEN
2   CHECK_SLOPE_CAPTURE_RULES:=NO;
3 ELSE
4   FOR I:=2 TO 8 DO
5     IF (I<>NUMBER_LINE) AND (I<>6) THEN
6       IF (SLOPE_WAITING_TIME[NUMBER_LINE]<SLOPE_WAITING_TIME[I]) AND (SLOPE_PLACE_FOUND[I]=FALSE) AND (CAPTURE_ENABLE_SLOPE_LINE[I]=TRUE) THEN
7         ANSWER:=NO;
8       END_IF
9     END_IF
10  END_FOR
11 END_IF
12 CHECK_SLOPE_CAPTURE_RULES:=ANSWER;
13
```

Рисунок 3.9 – Код функції CHECK_SLOPE_CAPTURE_RULES

Логіка схожа на перевірку попереднього конвеєра. Якщо вантаж, що перший у черзі і знаходиться на розгонному, довше 700 одиниць часу в режимі WAIT, захоплення місця на центральній лінії іншими конвеєрами забороняється (CHECK_SLOPE_CAPTURE_RULES := NO).

Далі перевіряються лінії з 2 по 8 (крім поточної NUMBER_LINE і виняткової лінії 6). Якщо інша лінія I чекає довше за поточну, ще не знайшла місце і має дозвіл на захоплення, то перевага надається їй, і поточній лінії

відмовляється в захопленні (ANSWER := NO). Це гарантує справедливий розподіл місць – їх отримує лінія, яка довше чекає.

Після виконання функції, які були розглянуті вище повертаємось до виконання логіки POU CONTROL. Де далі виконується логіка підрахунку активних ліній. Лінія вважається активною, якщо вона в режимі WORK або TRAIN, якщо в лінії затримка до сну ще не вичерпана, і це не стосується 6 лінії, бо вона має буферну логіку. Також оновлюється масив в якому зберігається чи активна лінія зараз чи ні.

Далі в POU CONTROL йде фрагмент перевірки, коли системі дозволено запуснитись, Основна система сортування вимкнена (SORT_SYSTEM_WORK=FALSE) і немає зв'язку (WORK_CONNECTION=0), вимкнено контактор або вимикач роботи, або є аварія затору на будь-якому датчику від ALARM_JAM[0] до ALARM_JAM[6]. У цьому випадку викликається функція ALL_OFF() (рис. 3.10), що вимикає все обладнання. Логіка цієї функції спочатку вимикає головний вихідний конвеєр (центральный конвеєр). Далі циклом FOR логіка проходиться по всіх 8 вхідних потоках і вимикає всі типи конвеєрів для кожної лінії. Також всі MODE для різних конвеєрів переводяться в режим STOP. Як результат всі конвеєра кожної лінії більше не активні і система не буде приймати або передавати товари. І для ліній 1–4 активується окремий вихід сигнал STOP з певною затримкою, сигнал певний час утримується для зупинки телескопічних конвеєрів.

Після виклику функції ALL_OFF() система визначає причину та встановлює відповідний режим, якщо затор то встановлюється режим ALARM, якщо вимкнений контактор або вимикач – режим STOP, якщо нічого критичного, але запуск ще не дозволений – переводиться в режим READY_FOR.

```
CONTROL ALL_OFF x
1 OUTPUT_CONV[0,0]:=OFF;
2 FOR NUMBER_LINE:=1 TO 8 DO
3     OUTPUT_CONV[NUMBER_LINE,TELESCOPE]:=OFF;
4     OUTPUT_CONV[NUMBER_LINE,MIDDLE_CONV]:=OFF;
5     OUTPUT_CONV[NUMBER_LINE,SLOPE_CONV]:=OFF;
6     OUTPUT_CONV[NUMBER_LINE,TURN_CONV]:=OFF;
7     OUTPUT_CONV[NUMBER_LINE,ACC_CONV]:=OFF;
8     OUTPUT_CONV[NUMBER_LINE,MERGER]:=OFF;
9     LINE_MODE[NUMBER_LINE]:=STOP;
10    PLACE_MODE[NUMBER_LINE]:=STOP;
11    TURN_MODE[NUMBER_LINE]:=STOP;
12    SLOPE_MODE[NUMBER_LINE]:=STOP;
13    MIDDLE_MODE[NUMBER_LINE]:=STOP;
14    TELESCOPE_MODE[NUMBER_LINE]:=STOP;
15    IF NUMBER_LINE<5 THEN
16        OUTPUT_CONV[NUMBER_LINE, STOP]:=ON;
17        DELAY_FOR_STOPING_T[NUMBER_LINE]:=TIME_FOR_STOPING_T;
18    END_IF
19 END_FOR
```

Рисунок 3.10 – Код функції ALL_OFF

Якщо умови дозволяють запуск, декрементується таймер перед стартом (START_DELAY). Коли він досягає нуля, режим переходить у WORK – основний робочий стан. Логіка перевіряє готовність кожної лінії: якщо вона не вимкнена через НМІ і всі частотні перетворювачі готові, лінія вмикається, інакше конвеєри переходять у режим STOP.

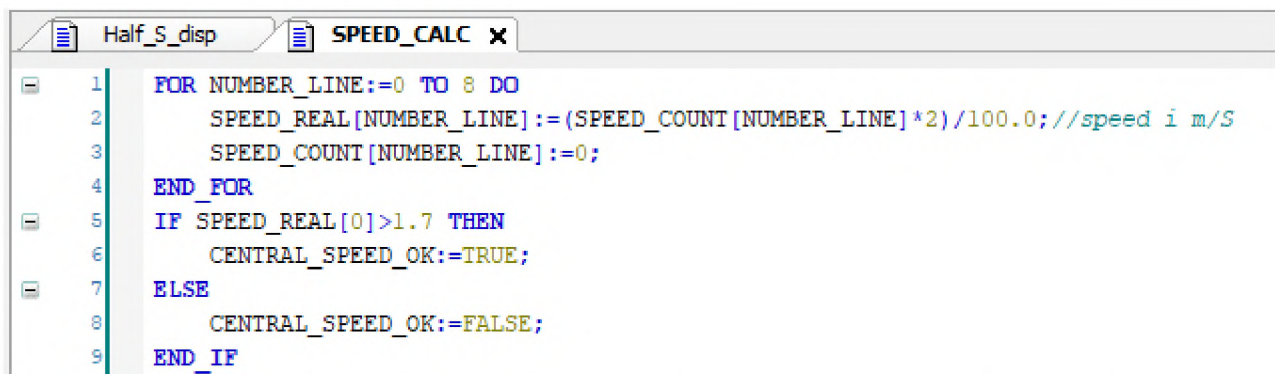
У CONTROL циклічно зменшуються всі таймери затримки, якщо вони більші за нуль. Далі увімкнення допоміжних конвеєрів прив'язане до стану центрального: якщо центральний увімкнений – допоміжні також вмикаються, інакше – вимикаються. Обробка головного вимикача зупиняє систему через виклик ALL_OFF(), скидає аварії ALARM_JAM і дозволяє підрахунок на всіх лініях (ENABLE_COUNTER).

У POU CONTROL реалізована логіка обробки заторів.

Також це POU керує світлосигнальною колоною: зелений світло – лінія працює, жовтий – готовність, моргання жовтого – режим TRAIN або SWITCHING_ON, червоний – зупинка, моргання зеленого – режим SLEEP, моргання червоного – аварія.

В кінці POU CONTROL скидаються всі лічильники вантажу на вхідних потоках і обробляється кнопка скидання статистики продуктивності лінії.

На цьому всі POU в Task MAST розглянуто, переходимо до TASK Half_S . В цьому Task викликається лише одна POU Half_S_disp (додаток Д). На початку коду викликається функція SPEED_CALC() (рис. 3.11).



```
1 FOR NUMBER_LINE:=0 TO 8 DO
2     SPEED_REAL[NUMBER_LINE]:=(SPEED_COUNT[NUMBER_LINE]*2)/100.0;//speed i m/S
3     SPEED_COUNT[NUMBER_LINE]:=0;
4 END_FOR
5 IF SPEED_REAL[0]>1.7 THEN
6     CENTRAL_SPEED_OK:=TRUE;
7 ELSE
8     CENTRAL_SPEED_OK:=FALSE;
9 END_IF
```

Рисунок 3.11 – Код функції SPEED_CALC() в SoMachine V4.3

Цикл FOR для перевірки швидкості ліній вхідних потоків розрахунок множиться на 2 і ділиться на 100, бо для отримання м/с, бо лічильник рахує в сантиметрах, а функція викликається раз в пів секунди. Далі йде перевірка швидкості на центральній лінії.

Далі в POU Half_S_disp йде фрагмент коду для керування світлосигнальної колони для реалізації мигання певним кольором.

Потім іде код, що реалізує логіку оновлення швидкостей, якщо режим роботи звичайний то оновлюються швидкості різних ділянок і центральної лінії і поворотного конвеєрі, конвеєрів під номером 1 і всі значення беруться з таблиці в залежності який профіль швидкості обрано. В режимі налаштування лінії, швидкості на поворотних збільшуються в два рази. Далі в цьому режимі нічого не змінюється лише додатково розраховується фактична швидкість по вхідних потоках.

Переходимо до TASK Central_1cm в цьому завданні викликається лише POU Central (додаток Е).

У програмі POU Central на початку збільшується лічильник для підрахунку швидкості центральної стрічки. Потім залежно від режиму центральної лінії виконується відповідна логіка. В режимі ADJUST_MODE викликається функція ADJUST(), яка відповідає за налаштування центральної лінії та восьми вхідних потоків. Для центральної лінії очікують спрацювання першого датчика і збільшують крок налаштування. Якщо ADJUST_STEP між 3 і 5, при спрацюванні датчика з відповідним індексом крок збільшується далі.

Для потоків 1–8 спочатку зупиняються конвеєри за заднім фронтом вантажу, потім перевіряється режим налаштування. У режимах 0 і 1 на 3-му кроці при спрацюванні датчика запускається лічильник і встановлюється швидкість розгонного конвеєра, кроки збільшуються. На 4-му кроці при спрацюванні фінального датчика лічильник зупиняється, а кроки доводяться до 6. У режимі 2 на 3-му кроці розгонний конвеєр зупиняється, встановлюється затримка для повної зупинки, запускається лічильник і збільшується крок. На 4-му кроці після закінчення затримки конвеєр вмикається, крок знову збільшується. На 5-му кроці, при спрацюванні фінального датчика, лічильник зупиняється і кроки доводяться до 6. Логіка ADJUST() завершується.

У режимі USUAL_MODE викликаються функції SHIFT_PATH() та PLACE_LINE_WORK(). SHIFT_PATH() управляє зсувом головного масиву PATH на 3300 елементів – він копіює кожен елемент із попереднього, працюючи у циклі в зворотному напрямку. Якщо увімкнено ENABLE_COPY_ARRAY, масив розділяється на дві частини через обмеження НМІ: перші 2000 елементів копіюються у PATH_COPY_1, а решта 1300 – у PATH_COPY_2.

Далі перевіряється активність сенсорів центральної стрічки – якщо датчик активний або імітований, за наявності режиму пошуку помилок викликається відповідна процедура, інакше визначаються межі по двох датчиках на стрічці: перед та після виїзду вантажу. Після обробки елемент масиву, що відповідає датчику, позначається як вільний (білий). Є також обробка кнопки, яка заповнює простір вантажем за командою оператора.

Функція PLACE_LINE_WORK() відповідає за пошук вільного місця для вантажу. Спочатку інкрементуються лічильники часу очікування для кожної лінії – чим довше вантаж чекає, тим більший його пріоритет для резервування місця. Для кожної лінії залежно від її режиму роботи, у випадку режиму WAIT, перевіряється наявність вільного місця. Якщо місце є, лінія переходить у режим GIVE, скидає час очікування, додає вантаж у потік, а раніше зарезервоване місце звільняється і стає доступним іншим лініям. Якщо ж вільного місця немає – лінія продовжує чекати

3.2 Зовнішній вигляд НМІ панелей

Відповідно до технічного завдання, для взаємодії оператора з системою передбачено використання людино–машинних інтерфейсів (НМІ) (додаток Ж). Характеристики та опис панелей наведено нижче.

Головний екран системи (рис. Ж.1) відображає ключову інформацію про її стан. Надписи «ЗАТОР!!!» з'являються лише при виникненні відповідних несправностей на датчиках, дозволяючи оператору швидко локалізувати проблему.

У правому верхньому куті розташовані важливі вхідні та вихідні сигнали:

1. Вхідні сигнали:
 - CONTACTOR: сигнал від додаткових контактів силового контактора, що індидує наявність живлення;
 - READY_CENTRAL: готовність частотного перетворювача центральної стрічки;
 - CENTRALSPEED_OK: відповідність швидкості центральної стрічки.
 - READY_OVERFLOW: готовність лінії для вантажу, що йде на повторне проходження;
 - СОРТИРОВКА: наявність вхідного сигналу, що свідчить про запуск процесу сортування.

2. Вихідні сигнали:

- ALARM OUT: з'являється при заторі на лінії або переключенні лінії в ручний режим;
- SLEEP OUT: відображає стан «SLEEP» системи;
- Ready OUT: індикація готовності системи до роботи.

Під цими індикаторами відображається назва останньої помилки, яка спричинила зупинку системи.

Біля кожного вхідного потоку розташована кнопка з його номером. При відсутності готовності потоку кнопка блимає червоним. Якщо оператор натискає на неї, потік зупиняється, а логіка ігнорує його датчики до повторного натискання. Центральна стрічка відображає розраховану системою швидкість. По ходу лінії розташовані кнопки для імітації спрацювання датчиків.

Зліва вгорі знаходиться кнопка для імітації сигналу «СОРТИРОВКА», що дозволяє запускати систему для техобслуговування без зовнішнього сигналу. Праворуч розташована кнопка, що вмикає режим виводу сигналів сенсорів із затримкою вимкнення.

Кожна стрілочка на панелі відповідає реальному сигналу роботи конвеєра: її видимість свідчить про подачу сигналу. Невеликі сірі квадратики відображають спрацювання сенсорів, змінюючи колір на зелений.

Зліва вгорі знаходиться кнопка «KONSORT» натиснувши на неї можна потрапити на панель сервісного меню (рис. Ж.2).

Під час роботи лінії кнопки «Входи/виходи» і «Налаштування» не доступні для натискання.

Кнопка «Входи/виходи» створена для переходу на панель де можна моніторити входа і вручну запускати вихода контролера.

Кнопка «Параметри» активує перехід на панель параметрів всієї системи.

Кнопка «Налаштування» переводить на панель, що використовується для автоматичного калібрування або налаштування системи.

Кнопка «Конфігурація» переводить в заводське меню панелі де можна налаштувати яскравість, контрастність, можна провести діагностику панелі від

перевірки на биті пікселі до перевірки відклику на натискання, також там знаходиться конфігурація IP.

Кнопка «Головний екран» поверне до початкового екрану з якого ми почали розгляд панелей.

Кнопка «Статистика» переведе на панель де знаходиться статистика скільки посилок і в якому режимі було злито з кожного потоку окремо.

Кнопка «RESET HMI» перезапустить панель.

На панелі «Входи/Виходи» (рис. Ж.3) ви можете переглядати стан усіх входів контролера. Після натискання кнопки «DEBUG» з'явиться можливість змінювати стани виходів контролера.

Виходи згруповані в стовпчики від «L1» до «L8». Натискання кнопки біля кожного з цих написів активує всі виходи, що належать до відповідного потоку. Нижче розташовані окремі кнопки для кожного виходу контролера. Потоки 1, 2, 3, 4 мають на один вихід більше, оскільки останній конвеєр є телескопічним і керується двома сигналами (старт і стоп), що вимагає двох виходів. Інші конвеєри працюють за принципом «якщо є сигнал, конвеєр працює». Під кожним потоком знаходяться два поля: Перше відображає поточне значення аналогового сигналу, що подається на частотний перетворювач. Друге дозволяє вручну змінювати це значення.

Кнопка «Центр» відповідає за вихід, що керує центральною лінією. Праворуч від неї відображається швидкість центральної стрічки, розрахована системою. Кнопка «EXIT» повертає до панелі сервісного меню.

Далі розглянемо панель параметрів всієї системи (рис. Ж.4).

Всі параметри на цій панелі можна змінювати в межах, що задані для кожного з них в налаштуваннях кожного поля вводу і робота всієї лінії залежить від значень, що вказані в цих налаштуваннях, частина з них відповідає за вимкнення або увімкнення певних функцій, частина відповідає за затримки для кожної з дій або час який буде відраховувати система для спрацювання тієї чи іншої функції.

Також є кнопка «EXIT», що поверне в сервісне меню і кнопка, що переведе на панель для редагування параметрів, що система розрахувала під час власного калібрування (рис. Ж.5).

На панелі налаштувань користувач отримує повний контроль над параметрами потоків, що були ретельно відкалібровані. Це дозволяє не лише переглядати актуальні налаштування, але й оперативно їх змінювати відповідно до потреб або бля оптимізації процесу.

Особливе значення має функція ENABLE_MIDDLE. На данній панелі реалізована можливість централізовано вмикати або вимикати її для всіх потоків. Важливо зазначити, що потік 6 є винятком з цього правила, оскільки для нього за проханням замовника була розроблена інша, індивідуальна логіка роботи. Кнопка над «EXIT» переводить на панель налаштувань швидкості центральної стрічки де можна обрати з якою швидкістю буде працювати система (рис. Ж.6).

Виділений елемент потрібен для вводу значення на який швидкісний режим переводити лінію справа від нього знаходяться всі збережені аналогові значення, що відповідають швидкісним режимам стовпчик NOW виводить значення, що подається зараз на частотний перетворювач. Нижній рядок створений для можливості ввести корективи попередньо всі ці значення отримуються і зберігаються системою автоматично під час калібрування. Присутня кнопка збереження змін. «EXIT» поверне до панелі сервісне меню.

Панель налаштувань (рис. Ж.7). На цій панелі відбувається повне керування процесом самостійного налаштування системи. Від визначення розміщення датчиків на центральній стрічці відносно елементів масиву центральної лінії. В залежності від потоку, що буде налаштовуватись кількість об'єктів на панелі буде змінюватись. Є також є зелена лампочка яка відображає чи запущений процес налаштування чи ні.

Справа біля надпису «ЛІНІЯ» знаходиться поле для вводу яка лінія буде налаштовуватись, якщо 0 то центральна, якщо інші значення то потік номер якого був вказаний. Справа від надпису «ADJUSTING_MODE» є поле в якому при налаштуванні вхідного потоку потрібно ввести в якому моді провести

налаштування. Правіше від лампочки виводиться значення який крок налаштування зараз виконує лінія. І 4 в ряду елемент це поле для вводу значення для якого швидкісного режиму проводяться налаштування. В всіх інших полях значення будуть з'являтися по мірі проходження налаштування в різних модах.

Кнопка «СТАРТ» ініціює процес налаштування. Кнопка «СКИНУТИ» стирає останні отримані значення, кнопка «ЗБЕРЕГТИ» перезаписує існуючі значення параметрів для лінії на нові. Кнопка «Exit» повертає в сервісне меню.

Панель відображення вантажу (рис. Ж.8) активується прозорою кнопкою у правому куті головного екрану. На цій панелі використовуються шість кольорів для візуалізації вантажу. Лінії 1 та 6 не мають режиму створення місця, тому для них колір відсутній. Інші лінії використовують два кольори. Прозорий колір - відображає виділене місце в масиві для відповідної лінії. Насичений колір - показує фактично влитий вантаж, який пройшов перший датчик на центральній лінії. Після проходження наступного датчика колір вантажу змінюється на білий.

«EXIT»: Повертає на головний екран.

3.3 Техніко–економічне обґрунтування

У межах даного розділу проаналізовано основні статті витрат, що включають витрати на розробку програмного забезпечення та вартість необхідного обладнання. Всі розрахунки та фінансові показники систематизовано та представлено у наступних таблицях (табл. 3.1, 3.2):

Таблиця 3.1 – Витрати на розробку ПЗ

Найменування	Сума, долари США	Сума, грн
Оплата праці розробників	10000	414386
Оплата праці тестувальників	3800	151466
Оплата праці інженерів з пусконаладження	5000	207193
Навчання персоналу	1750	72518
Разом	20550	845563

Таблиця 3.2 – Вартість обладнання

Найменування	Ціна 1 шт. в грн	Кількість шт.
Програмований логічний контролер М241 40 вх/вих транз 2RS485 1ETH	21681,36	1
Модуль ТМ4 4 порти ETHERNET комутатор	6125,25	1
Модуль ТМ3 32 вх =24В	7647,55	3
Модуль ТМ3 32 вих транз	7693,08	2
Модуль ТМ3 4 аналог вих 12 біт	7875,20	2
Модуль ТМ3 передавач	3218,60	1
Модуль ТМ3 приймач	3218,60	1
Модуль ТМ3 16 вх =24В	5462,60	1
Модуль ТМ3 16 вих 0,5А транз	5508,13	1
Сенсор. пан. 10.4" VGA-TFT НМІGТO5310	58252,11	1
Разом	157545,86	

PLC (програмований логічний контролер) та всі його модулі та комплектуючі можна і доцільно відносити до основних засобів (ОЗ). Розрахунок амортизації лінійним методом провадиться за формулою:

$$A = \frac{C \times H}{100 \%}, \quad (3.1)$$

де А – сума амортизаційних відрахувань за рік, грн;

Н – норма амортизаційних відрахувань, %;

С – середньорічна вартість основних засобів;

Відповідно: $196480,07 \times 20 \% : 100 \% = 31509,17$ грн/рік.

або приблизно: $39296,014 / 41,44 \approx 760,36$ долари США.

Загальний кошторис охоплює основні статті витрат (табл. 3.3).

Таблиця 3.3 – Кошторис витрат

Найменування економічних елементів витрат	Сума, долари США	Сума, грн
Витрати на розробку ПЗ	20550	845563
Амортизація основних засобів	760,36	31509,17
Інші витрати (комунальні послуги)	470	19476,15
Разом	21 780,36	896548,32

До розрахунків було прийнято середній за червень 2025 року курс долара по відношенню до гривні: 1 дол. США = 41,44 грн.

ВИСНОВКИ

Дана кваліфікаційна праця присвячена вивченню проблематики розробки інтегрованої системи, що поєднує програмні та апаратні засоби, з метою автоматизації операцій об'єднання відправлень на логістичних центрах. Результатом проведеного аналізу став комплекс теоретичних висновків та практичних напрацювань, які свідчать про обґрунтованість та результативність застосування аналогічних комплексів у логістичній галузі.

Було здійснено огляд актуальних програмно–апаратних рішень, що застосовуються для автоматизації промислових процесів. Виявлено, що для впровадження автоматизованих систем у логістичному секторі оптимальним є застосування апаратури модульного типу, зокрема промислових логічних контролерів. Такі технологічні підходи гарантують адаптивність конфігурації, значний рівень експлуатаційної стійкості, можливість розширення функціоналу та інтеграцію з наявними інфраструктурними об'єктами логістики.

Сформульовано концептуальні засади програмно–апаратного рішення, що має на меті автоматизувати процедуру консолідації посилок. Запропонований підхід бере до уваги особливості виконання логістичних завдань та спрямований на оптимізацію часових витрат при обробці вантажних потоків, мінімізацію ймовірності помилок через людський чинник, а також на підвищення прецизійності операцій сортування та подальшого відвантаження.

Підготовлено попередній проєкт програмно–апаратного забезпечення, який охоплює втілення основних функціональних можливостей, а саме злиття вантажу в 1 один потік з чітким збереженням відстані між вантажем для подальшого процесу сканування і автоматичного процесу сортування. Передбачена також адаптивність архітектури системи до масштабів конкретного логістичного об'єкта.

Прикладна цінність дослідження визначається створенням технічно виваженої концептуальної моделі, що може слугувати фундаментом для побудови діючого автоматизованого рішення. При розробці акцент робився на

застосуванні загальнодоступних складових, що забезпечує економічну привабливість системи та спрощує її технічну підтримку в майбутньому.

Досягнуті в ході роботи результати є перспективними для імплементації у виробничі процеси транспортно–логістичних підприємств, складських комплексів, центрів сортування вантажів та розподільчих вузлів.

Підсумовуючи, варто констатувати повне досягнення мети, поставленої у кваліфікаційній праці, та успішне виконання всіх визначених завдань. Сформований ескізний проект інтегрованого програмно–апаратного рішення гармонує з актуальними стандартами, що висуваються до систем автоматизації логістичних процесів, і демонструє значні перспективи для його практичного застосування.