

Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра автоматизації, комп'ютерних наук і технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеню вищої освіти – магістр
за освітньо-професійною програмою
«Кіберфізичні системи в промисловості, бізнесі та транспорті»

зі спеціальності

174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

тема роботи:

«Автоматизоване керування сортуванням деталей по кольору з використанням пристроїв пневмоавтоматики»

Виконав ст. гр. АКІТР-23-1м.	_____	Максименко О. В.
Керівник	_____	Савицький О. І.
Нормоконтроль	_____	Маринич І. А.
Завідувач кафедри	_____	Рубан С. А.

Кривий Ріг – 2024

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: інформаційних технологій

Кафедра: автоматизації, комп'ютерних наук і технологій

Ступінь вищої освіти: Магістр

Спеціальність: 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

ЗАТВЕРДЖУЮ

Зав. кафедри: к.т.н. Рубан С.А.

« 5 » липня 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра

студентові групи АКІТР-23-1м. Максименко Олексію Володимировичу

1. Тема кваліфікаційної роботи: «Автоматизоване керування сортуванням деталей по кольору з використанням пристроїв пневмоавтоматики»

затверджено наказом по університету № 595с від 04.07.2024 р.

2. Термін здачі кваліфікаційної роботи: 01.12.2024 р.

3. Склад кваліфікаційної роботи: Пояснювальна записка обсягом 87с., додатки, презентація у Microsoft PowerPoint (16 слайдів) в електронному та друкованому вигляді

4. Консультанти кваліфікаційної роботи:

Розділ 1-3

доц. Савицький О. І.

Нормоконтроль

доц. Маринич І. А.

5. Календарний план:

№	Етапи роботи	Термін виконання
1	<i>Вступ</i>	<i>10.07.24</i>
2	<i>Розділ 1</i>	<i>15.07.24</i>
3	<i>Розділ 2</i>	<i>18.08.24</i>
4	<i>Розділ 3</i>	<i>19.09.24</i>
5	<i>Висновки</i>	<i>15.10.24</i>
6	<i>Оформлення кваліфікаційної роботи</i>	<i>20.11.24</i>
7	<i>Підготовка презентації та графічного матеріалу</i>	<i>28.11.24</i>
8	<i>Підготовка доповіді до захисту</i>	<i>01.12.24</i>

6. Дата видачі завдання: 28.06.2024р.

Керівник _____ /Савицький О. І./

7. Запевнення: Я, Максименко Олексій Володимирович, запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомлений.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Здобувач _____ / Максименко О. В./

АНОТАЦІЯ

Максименко О.В. «Автоматизоване керування сортуванням деталей по кольору з використанням пристроїв пневмоавтоматики».

Кваліфікаційна робота на здобуття ступеню вищої освіти магістр за освітньо-професійною програмою «Кіберфізичні системи в промисловості, бізнесі та транспорті» зі спеціальності 174 – Автоматизація, комп'ютерно – інтегровані технології та робототехніка– Криворізький національний університет, Кривий Ріг, 2024.

Об'єктом розроблення виступає автоматизована система, призначена для сортування об'єктів на основі їхніх кольорових параметрів. Ця система забезпечує інтеграцію сенсорних компонентів, що виконують зчитування кольору деталей, і подальшу обробку отриманих даних за допомогою контролера. Контролер, у свою чергу, управляє сортувальним механізмом, що працює з використанням пневматичного обладнання.

Дослідження зосереджене на вдосконаленні технічних засобів для точного розпізнавання кольору та ефективного сортування у виробництві.

У першому розділі розглянуто принципи роботи систем сортування за кольором, методи їх реалізації та аналіз існуючих рішень. Було обґрунтовано можливість модернізації лабораторного стенду та вибір апаратного забезпечення.

У другому розділі описано технічні аспекти роботи датчика кольору, кінематику маніпулятора й реалізацію стенду сортування.

У третьому розділі наведено програмну реалізацію системи, яка забезпечує обробку даних і керування процесом сортування. Було реалізовано алгоритм програми та зроблено блок-схему алгоритму. Було проведено тестування системи, яка показала хороші результати точності та ефективності.

Ключові слова:

СОРТУВАННЯ ДЕТАЛЕЙ, АВТОМАТИЗАЦІЯ, РОЗПІЗНАВАННЯ КОЛЬОРУ, КОЛІРНІ СЕНСОРИ, АЛГОРИТМИ СОРТУВАННЯ, ПНЕВМАТИЧНІ СИСТЕМИ, ARDUINO, NODE-RED, ІНДУСТРІЯ 4.0, SIMATIC S7-1200.

ANNOTATION

Maksymenko O.V. "Automated control of the sorting of parts by color using pneumatic automation devices."

Graduation master's work for obtaining an educational degree «Master» for the educational and professional program « Cyber-physical systems in industry, business and transport » in specialty 174 – «Automation, computer-integrated technologies, and robotics». – Kryvyi Rih National University, Kryvyi Rih, 2024.

The object of development is an automated system designed for sorting objects based on their color parameters. This system provides the integration of sensor components that read the color of the parts, and further processing of the received data using the controller. The controller, in turn, controls the sorting mechanism, which works with the use of pneumatic equipment.

The research is focused on improving technical means for accurate color recognition and efficient sorting in production.

The first chapter deals with the principles of color sorting systems, methods of their implementation and analysis of existing solutions. The possibility of modernizing the laboratory stand and the choice of hardware were substantiated.

The second chapter describes the technical aspects of the color sensor, the kinematics of the manipulator, and the implementation of the sorting stand.

The third section presents the software implementation of the system, which provides data processing and management of the sorting process. The algorithm of the program was implemented and the block diagram of the algorithm was made. The system was tested and showed good accuracy and efficiency results.

Keywords:

PARTS SORTING, AUTOMATION, COLOR RECOGNITION, COLOR SENSORS, SORTING ALGORITHMS, PNEUMATIC SYSTEMS, ARDUINO, NODE-RED, INDUSTRY 4.0, SIMATIC S7-1200.

ЗМІСТ

<i>ВСТУП.....</i>	6
<i>РОЗДІЛ 1 АНАЛІЗ СИСТЕМ СОРТУВАННЯ ДЕТАЛЕЙ ЗА КОЛЬОРОМ ТА ОГЛЯД СУЧАСНИХ РІШЕНЬ У ГАЛУЗІ АВТОМАТИЗАЦІЇ.....</i>	8
1.1 Аналіз технологічного процесу та огляд існуючих рішень інтегрованих АСС за кольоровими характеристиками.....	8
1.1.1 Аналіз особливостей технологічного процесу сортування за кольором	8
1.1.2 Класифікація систем сортування об'єктів за кольоровими характеристиками	12
1.2 Аналіз можливостей застосування технологій Industry 4.0 для інтелектуальності процесів сортування деталей.....	18
1.3 Огляд сучасних рішень у сфері автоматизованого сортування об'єктів за кольором	21
1.4 Обґрунтування впровадження удосконалень у систему сортування.....	26
<i>РОЗДІЛ 2 АНАЛІЗ ТА ВИБІР АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ СОРТУВАННЯ ЗА КОЛЬОРОМ</i>	28
2.1 Структурний аналіз об'єкта керування лабораторного стенду «маніпулятор-конвеєр».....	28
2.2 Математичне забезпечення маніпулятора лабораторного стенду	40
2.3 Математичне моделювання в середовищі Matlab.....	43
2.3 Вибір пристроїв та датчиків для розробки системи сортування	45
2.3.1 Вибір елементів визначення кольору для системи	45
2.3.2 Вибір елементів сортування об'єктів на стрічковому конвеєрі	49
2.4 Математичне моделювання розпізнавання кольорів та керування пневмосоплами	51
2.5 Обґрунтування вибору програмного забезпечення для реалізації системи сортування.....	56
<i>РОЗДІЛ 3 ПРОГРАМНО-ТЕХНІЧНА РЕАЛІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ ПРОЦЕСОМ СОРТУВАННЯ.....</i>	58

	4
3.1 Проектування схем автоматизації системи сортування	58
3.2 Алгоритм роботи системи сортування по кольору	62
3.3 Програмна реалізація проекту	64
3.3.1 Калібровка датчику та реалізація ідентифікації кольору	64
3.3.2 Налаштування зв'язку між Arduino і ПЛК	71
3.3.3 Модифікація програми стенду системою сортування за кольором	75
<i>ВИСНОВКИ</i>	<i>79</i>
<i>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ</i>	<i>81</i>

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

Скорочення:

АСУТП – автоматизована система управління технологічним процесом

ІЗ – інформаційне забезпечення

ЛМІ – людино машинний інтерфейс

ПЛК (або PLC) – програмований логічний контролер або РС-сумісний контролер

ТЗ – технічне завдання

СІК- система інтеграції контролерів

АПС – автоматизований процес сортування

ОС – об'єкт сортування

ЛЗК – логіка зчитування кольору

RGB – система передачі кольору (Red, Green, Blue)

ПО – програмне забезпечення

ДК – датчик кольору

АСС – автоматизована система сортування

КФС – кіберфізичні системи

ВСТУП

Актуальність теми магістерської роботи пояснюється швидким прогресом технологій автоматизації, що охоплює всі сфери сучасної промисловості. Зокрема, автоматичне сортування об'єктів за кольоровими параметрами відіграє ключову роль у виробничих галузях, де від точності та оперативності цього процесу залежить загальна продуктивність підприємств.

Традиційні методи сортування, через свою обмежену гнучкість і значну потребу в людських ресурсах, не відповідають сучасним вимогам, що призводить до збільшення витрат і зниження ефективності. Упровадження новітніх сенсорів і автоматизованих керуючих пристроїв забезпечує високоточне розпізнавання кольорів і зменшує залежність від людського фактору, що й визначило вибір теми даного дослідження.

У магістерській роботі досліджується проектування автоматизованої системи для сортування деталей за кольоровими характеристиками із використанням сенсорних технологій та пневматичних механізмів. Такий підхід є перспективним, оскільки дозволяє досягти балансу між продуктивністю, вартістю обладнання та універсальністю в адаптації до різних виробничих завдань.

Крім того, однією з переваг впровадження таких систем є можливість їх інтеграції у вже існуючі виробничі лінії. У цьому контексті важливим стає дослідження сумісності обладнання, алгоритмів взаємодії та розробка гнучких рішень для управління процесом сортування.

Хоча сьогодні доступні різноманітні автоматизовані рішення, сортування за кольорами залишається актуальним напрямом досліджень, який потребує подальшої розробки. Існуючі наукові праці часто зосереджені на загальних питаннях автоматизації або роботі сенсорів, але доволі рідко висвітлюють специфіку сортувальних процесів, особливо в частині інтеграції кольорових сенсорів із пневматичними системами. Серед ключових напрямків варто відзначити використання оптичних і кольорових датчиків, а також програмно-апаратних комплексів для управління сортувальними механізмами.

Мета роботи полягає у створенні системи автоматизованого сортування деталей за кольорами, що базується на сучасних сенсорах та пневматичних пристроях, для підвищення точності та продуктивності виробничих процесів.

Об'єкт дослідження – автоматизована система сортування, яка використовує сенсори для розпізнавання кольорів і пневматичні механізми для переміщення об'єктів.

Предмет дослідження – технічні рішення та алгоритми для розпізнавання кольорів і керування сортувальними пристроями на основі даних від сенсорів і контролерів у реальних умовах виробництва.

Для досягнення поставленої мети визначено такі завдання:

1. Дослідити сучасні технології автоматизованого сортування за кольором та оцінити їх ефективність.
2. Розробити алгоритм обробки даних від сенсорів кольору для прийняття рішень про сортування.
3. Спроекувати систему керування сортувальними процесами із застосуванням пневматичних виконавчих пристроїв.
4. Реалізувати і протестувати розроблену систему на експериментальному стенді.

Ця робота спрямована на створення оптимального рішення для автоматизованого сортування об'єктів за кольоровими характеристиками, що відповідає сучасним технологічним і виробничим вимогам.

РОЗДІЛ 1

АНАЛІЗ СИСТЕМ СОРТУВАННЯ ДЕТАЛЕЙ ЗА КОЛЬБОРОМ ТА ОГЛЯД СУЧАСНИХ РІШЕНЬ У ГАЛУЗІ АВТОМАТИЗАЦІЇ

1.1 Аналіз технологічного процесу та огляд існуючих рішень інтегрованих АСС за кольоровими характеристиками

1.1.1 Аналіз особливостей технологічного процесу сортування за кольором

Технологічний процес сортування деталей за їх кольоровими характеристиками є невід'ємною складовою сучасних виробничих ліній. Його завданням є класифікація об'єктів за кольором для забезпечення їх коректного розподілу, подальшої обробки, зберігання чи транспортування. Цей процес є критично важливим для галузей, де продуктивність, якість та точність виконання операцій мають вирішальне значення. До таких галузей належать харчова промисловість, електроніка, текстильна і автомобільна сфери, де кольорова сортувальна операція визначає ефективність усього виробничого циклу.

Ключовим аспектом успішного впровадження технологічного процесу сортування є автоматизація, яка дозволяє досягти високої швидкості обробки, усунути залежність від людського фактору та забезпечити стабільність роботи системи. Їх активно застосовують на етапах складання для ідентифікації виробів за кольоровими характеристиками та їх подальшої класифікації. Основними завданнями використання таких систем є забезпечення перевірки якості готової продукції, оптимізація процесів сортування та пакування, контроль стану зберігання товарів, включно з визначенням їхнього терміну придатності, а також спостереження за обсягом і характером виробничих відходів. Автоматизовані системи сортування базуються на інтеграції сучасних сенсорів, високопродуктивних контролерів і виконавчих механізмів. Ця інтеграція забезпечує швидку обробку великого обсягу інформації, що надходить у режимі

реального часу, і точне виконання команд із мінімальними затримками. Крім того, завдяки високій точності сенсорів, вони дозволяють автоматизованим лініям швидко реагувати на виявлені невідповідності, що суттєво підвищує ефективність виробництва. Їх інтеграція сприяє зменшенню залежності від людського фактору та знижує ймовірність помилок, забезпечуючи стабільно високу якість продукції навіть за великих обсягів виробництва.

Процес сортування деталей [1] за кольором складається з кількох ключових етапів, які забезпечують ефективну роботу системи (рис. 1.1).

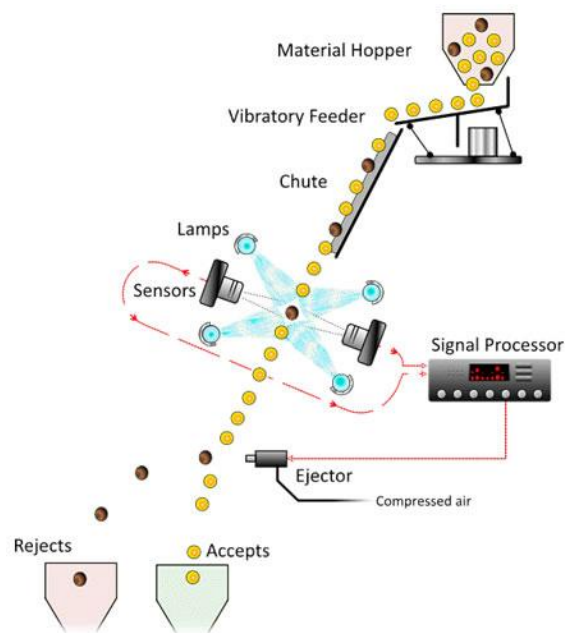


Рисунок 1.1 – Схематична система сортування за кольором

На першому етапі деталі подаються на сортувальну лінію. Для цього зазвичай використовується конвеєрна система або спеціальні подавальні механізми, які гарантують рівномірний потік об'єктів і правильне їхнє розташування. Подача деталей повинна бути організована так, щоб між ними не виникало зіткнень чи зупинок, оскільки це може призвести до збою роботи сенсорів, зниження швидкості обробки або навіть до механічних пошкоджень об'єктів.

Наступним етапом є зчитування кольорових характеристик об'єктів. Для цього використовуються спеціалізовані сенсори, які аналізують поверхню деталей та визначають їх кольорові параметри. Найбільш поширеними є RGB-сенсори, які вимірюють інтенсивність трьох базових кольорів: червоного, зеленого та синього.

Ці сенсори мають високу швидкодію, що дозволяє зчитувати параметри об'єкта навіть під час його руху по конвеєру. У більш складних випадках, наприклад, для сортування об'єктів із неоднорідними кольоровими характеристиками, використовуються спектральні сенсори або кольорові камери (рис. 1.2). Вони здатні забезпечувати вищу точність завдяки аналізу спектрального складу відбитого світла та порівнянню отриманих даних з еталонними значеннями. Завдяки цьому можна точно визначити, до якої категорії належить кожен об'єкт.



Рисунок 1.2 – Система сортування продуктів кольоровими камерами

Дані, отримані від сенсорів, передаються до контролера, який здійснює обробку інформації та приймає рішення про подальшу класифікацію об'єкта. Обробка даних є складним процесом, що враховує задані параметри сортування, такі як точність, швидкість і допустимі відхилення у характеристиках кольору. Контролери працюють у режимі реального часу, що дозволяє забезпечити швидку реакцію системи навіть у складних умовах. Алгоритми, які використовуються контролерами, зазвичай базуються на порівнянні отриманих параметрів із заздалегідь заданими характеристиками. Наприклад, якщо параметри кольору об'єкта відповідають червоному кольору, контролер направляє сигнал до виконавчого механізму, який перенаправляє цей об'єкт у відповідний контейнер.

Переміщення об'єктів [2] здійснюється за допомогою виконавчих механізмів, які можуть бути механічними, електричними або пневматичними. У виробничих умовах найбільшого поширення набули пневматичні системи, які забезпечують швидке, точне і безшумне сортування об'єктів. Виконавчі механізми повинні працювати в строгій синхронізації з контролером і сенсорами, щоб уникнути помилок під час перенаправлення об'єктів. Після класифікації об'єкти розподіляються у відповідні контейнери чи зони накопичення, що завершує технологічний процес. Важливо, щоб об'єкти розподілялися рівномірно та без пошкоджень, що досягається завдяки точній роботі всієї системи.

Технологічний процес сортування за кольором [3] має свої особливості та виклики. Серед ключових особливостей можна виділити високу швидкість обробки, необхідну для забезпечення безперервної роботи виробничої лінії, та високу точність визначення кольору. У цьому контексті особливу увагу слід приділити інтеграції різних елементів системи, що забезпечує її ефективну роботу. Наприклад, сенсори повинні бути правильно налаштовані для роботи в умовах виробництва, таких як змінне освітлення чи наявність сторонніх відблисків. Виконавчі механізми повинні бути достатньо міцними, щоб витримувати тривалі навантаження без зниження продуктивності.

Попри високу ефективність автоматизованих систем сортування, їхній розвиток супроводжується рядом викликів. Зокрема, необхідність синхронізації між компонентами системи вимагає використання складних алгоритмів управління. Крім того, висока інтенсивність роботи призводить до зношування виконавчих механізмів, таких як пневматичні приводи, що потребує регулярного обслуговування. Ще одним викликом є забезпечення високої точності зчитування кольору в умовах змінного освітлення, що вимагає використання додаткових оптичних фільтрів або алгоритмів компенсації.

Прикладом організації сортування деталей за кольором може бути система, яка використовує пневматичні приводи. Деталі рухаються конвеєрною стрічкою, проходять зону сканування, де сенсори кольору визначають їхні характеристики. Контролер обробляє дані та активує відповідні пневматичні приводи, які

перенаправляють деталі у відповідні контейнери. Такий підхід забезпечує не тільки високу точність і швидкість, але й адаптивність до змін у виробничих умовах.

Отже, технологічний процес сортування деталей за кольором є комплексною процедурою, яка включає подачу об'єктів, зчитування кольору, обробку даних, класифікацію та переміщення об'єктів у відповідні зони. Сучасні автоматизовані системи дозволяють виконувати ці операції з високою точністю та швидкістю, що відповідає вимогам сучасної промисловості. Завдяки інтеграції сучасних сенсорів, контролерів і виконавчих механізмів процес сортування стає не тільки більш ефективним, але й універсальним для застосування в різних галузях.

1.1.2 Класифікація систем сортування об'єктів за кольоровими характеристиками

Системи сортування об'єктів за кольоровими характеристиками є важливою складовою сучасних виробничих процесів, що дозволяють автоматизувати класифікацію продукції для її подальшої обробки, пакування або транспортування. Вибір типу сортувальної системи [4] залежить від особливостей об'єктів, вимог до точності й швидкості роботи, а також від технологічних і економічних можливостей підприємства. Для оптимального впровадження цих систем важливо розуміти їх класифікацію, яка ґрунтується на різних критеріях: рівень автоматизації, методи виявлення кольору та тип виконавчих механізмів.

Залежно від ступеня автоматизації, сортувальні системи поділяють на ручні, напівавтоматизовані та автоматизовані [5]:

- **Ручні системи** сортування передбачають безпосередню участь людини у всіх етапах процесу. Працівник візуально визначає колір об'єктів і класифікує їх вручну. Цей метод є малоефективним, адже залежить від фізичної витривалості оператора та має високий ризик помилок.
- **Напівавтоматизовані системи** комбінують ручну працю з автоматизованими елементами, наприклад, з використанням сенсорів для визначення кольору. У таких системах класифікація об'єктів здійснюється

автоматично, але переміщення чи сортування виконує працівник. Напів автоматизовані системи дозволяють підвищити точність, але обмежують швидкість процесу.

- **Автоматизовані системи** сортування є найбільш прогресивним варіантом, адже весь процес — від виявлення кольору до розподілу об'єктів у відповідні зони — виконується автоматично. Ці системи мають високу продуктивність, мінімізують людський фактор і забезпечують стабільну точність навіть за великих обсягів виробництва.

Вибір методу аналізу [6] кольору є критичним елементом у проектуванні систем сортування, адже він визначає точність, швидкість і функціональність процесу. Сучасні системи сортування за кольоровими характеристиками використовують кілька основних підходів для зчитування кольорових параметрів об'єктів. Основними з них є використання RGB-датчиків, спектральних аналізаторів, кольорових камер і фотодіодів. Нижче наведено класифікацію за цими способами:

1. **RGB-метод.** Найпоширеніший метод, заснований на вимірюванні інтенсивності трьох основних кольорів: червоного (Red), зеленого (Green) і синього (Blue). Датчики аналізують рівні інтенсивності цих кольорів, обчислюють їхні співвідношення та визначають колір об'єкта. Дані сенсори мають високу швидкість зчитування, доволі прості в інтеграції в автоматизовані системи виробництв, але мають високу чутливість до змін освітлення, що може впливати на точність сканування кольору об'єкту. Також, у даних датчиків є обмеження у розпізнаванні складних кольорових відтінків, але це компенсується їх ціною та простотою у використанні в нескладних системах сортування. Основними сферами застосування можна виділити сортування пластикових виробів, виробництво пакувальних матеріалів, іграшкова промисловість, текстильне виробництво та харчова промисловість (наприклад, сортування цукерок або етикеток).
2. **Спектральний аналіз.** Цей метод базується на аналізі спектральних характеристик відбитого світла. Сенсори розпізнають колір шляхом порівняння спектральної відбивної здатності об'єкта з еталонними

значеннями. Аналізатори даного типу мають високу точність, особливо для складних кольорів або неоднорідних поверхонь, та можливість роботи в умовах змінного рівня освітлення. Серед недоліків можна виділити вищу вартість та складність у налаштуванні порівняно з датчиками RGB-методом. В основному використовуються для сортування фруктів та овочів за ступенем стиглості, контролем якості фарбування покриття в автомобільній промисловості і виробництві електроніки. В деяких країнах цей метод застосовують в виробництві медичних пристроїв. Спектральний аналіз дозволяє в стерильних умовах сортувати вироби за кольором для різних серій або партій (наприклад, елементи для одноразових шприців чи катетерів).

3. **Метод кольорових камер.** Використовуються високоточні камери, які сканують об'єкти та передають зображення до системи обробки. Алгоритми комп'ютерного зору аналізують кольорові параметри, форму та текстуру об'єктів. Вони використовуються у високотехнологічних галузях, таких як виробництво електроніки чи фармацевтики. З переваг можна виділити багатофункціональність та можливість аналізу об'єктів складної форми. Недоліками можна назвати високу вартість та великим навантаженням на обчислююче обладнання через великі обсяги даних від камер.
4. **Поляризаційний метод.** Базується на аналізі поляризаційних властивостей світла, що відбивається від об'єкта. Метод дозволяє розпізнавати кольори з різними характеристиками текстур і матеріалів. Переваги: застосовується для специфічних матеріалів, які важко розпізнати іншими методами. Недоліки: обмежене використання в загальних виробничих умовах.
5. **ІЧ-спектроскопія (інфрачервоний аналіз)** Використовується для аналізу кольорів, які неможливо визначити в видимому спектрі. Метод застосовується для матеріалів із прозорою або напівпрозорою текстурою. Перевагою є додаткова можливість аналізу матеріалів. Недоліки складаються зі складності і специфічності обладнання.

Одним із важливих критеріїв класифікації є рівень інтеграції у виробничу лінію. [7] Наприклад, окремі модулі сортування функціонують автономно й зазвичай

застосовуються для попередньої класифікації або як частина підготовчих етапів перед пакуванням. Водночас інтегровані системи є невіддільною частиною автоматизованої лінії й взаємодіють із іншими елементами, такими як системи контролю якості, маркування або складання. Такий підхід забезпечує високу продуктивність і точність роботи.

Спосіб передачі об'єктів до зони сортування також впливає на класифікацію систем. Лінійні системи, де об'єкти рухаються по прямолінійній траєкторії за допомогою конвеєрної стрічки, відрізняються простотою конструкції та високою швидкістю обробки. Натомість кругові системи базуються на використанні об'єктів, які рухаються по замкненій траєкторії, що дозволяє ефективно використовувати простір виробництва. Безперервні системи забезпечують обробку в постійному потоці, що мінімізує затримки, тоді як дискретні системи, навпаки, працюють із фіксованими точками сканування, забезпечуючи вищу точність для кожного об'єкта.

Для переміщення та сортування об'єктів у сучасних системах використовуються різні типи виконавчих механізмів, які відіграють ключову роль у забезпеченні точності, швидкості й ефективності процесу. Вибір типу механізму залежить від характеристик об'єктів, вимог до сортування та специфіки виробничих умов.

Механічні системи (рис. 1.3) є одними з найбільш простих і економічних рішень.

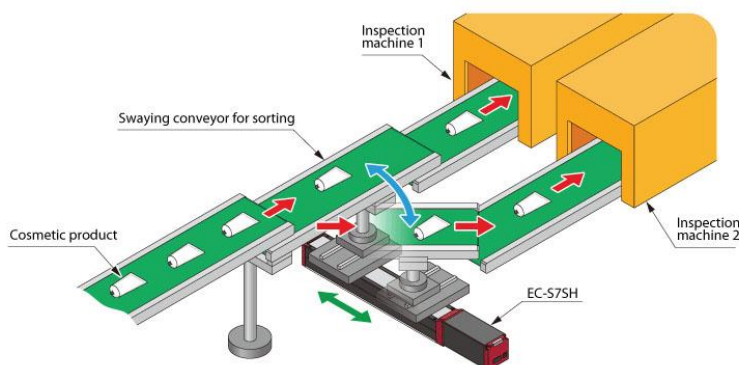


Рисунок 1.3 – Механічне обладнання для гойдання конвеєра

Вони включають такі елементи, як штовхачі, поворотні платформи або конвеєрні направляючі, які переміщують об'єкти до відповідних зон сортування.

Їх конструкція є відносно простою, що знижує витрати на виробництво та обслуговування. Однак механічні механізми мають обмеження щодо швидкості та гнучкості, а також потребують частого технічного обслуговування через фізичний контакт із деталями. Такі системи найбільше підходять для сортування однорідних і міцних об'єктів, які не пошкоджуються при контакті.

Пневматичні системи (рис. 1.4) базуються на використанні стисненого повітря для перенаправлення об'єктів до потрібних зон. Їх головна перевага полягає у високій швидкості роботи та відсутності прямого контакту з поверхнею об'єкта, що є важливим для крихких або легких деталей, таких як скляні вироби, харчові продукти чи пластикові компоненти. Пневматичні виконавчі механізми забезпечують високу точність і мінімізують ризик пошкоджень, але водночас вимагають наявності компресорів і систем подачі повітря, що може ускладнювати інсталяцію та збільшувати енергоспоживання. [8]

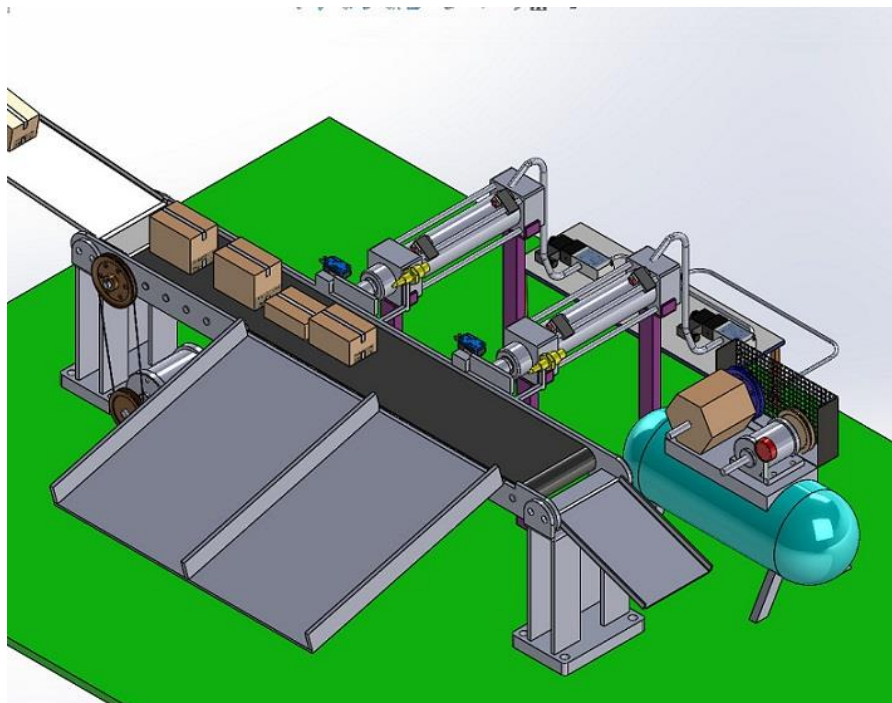


Рисунок 1.4 – Макет системи пневматичного сортування

Роботизовані маніпулятори [9] (рис. 1.5) є найбільш гнучким і високотехнологічним рішенням для сортувальних завдань. Вони забезпечують

точне позиціонування й здатні виконувати складні операції, зокрема сортування об'єктів з нерегулярними формами, неоднорідними текстурами або високою варіативністю кольору. Завдяки використанню програмного забезпечення роботизовані системи можуть швидко адаптуватися до змін у виробничих вимогах.

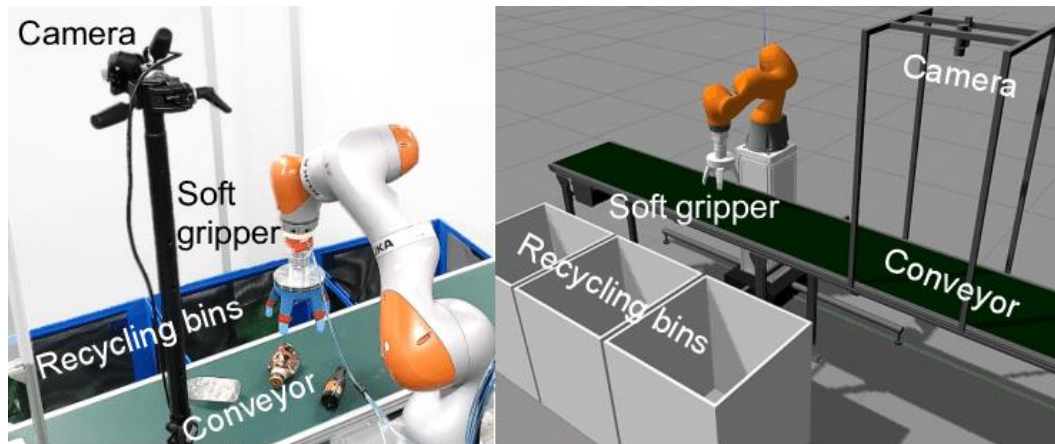


Рисунок 1.5 – Автоматизована система сортування відходів по кольору з використанням камери та маніпулятору

Однак такі системи вимагають значних капіталовкладень і кваліфікованого персоналу для обслуговування. Їх найчастіше використовують у високотехнологічних галузях, таких як автомобільна промисловість або виробництво електроніки.

Кожен із типів виконавчих механізмів має свої переваги й обмеження, які слід враховувати під час вибору рішення для конкретного виробництва. Прості механічні системи є доступними за ціною, але їх функціональність обмежена. Пневматичні механізми забезпечують швидкість і дбайливе переміщення об'єктів, але їхня експлуатація потребує наявності додаткових інженерних мереж. Роботизовані системи, хоча й забезпечують високу гнучкість і точність, можуть бути надмірно дорогими для малих і середніх підприємств.

У сучасному світі підприємства намагаються знайти баланс між точністю, продуктивністю та витратами. Наприклад, комбіновані рішення, що використовують RGB-сенсори разом із пневматичними штовхачами, дозволяють зберігати прийнятну вартість при забезпеченні достатнього рівня точності.

1.2 Аналіз можливостей застосування технологій Industry 4.0 для інтелектуальності процесів сортування деталей

Industry 4.0 - це концепція сучасного промислового виробництва, що базується на інтеграції кіберфізичних систем, Інтернету речей (IoT), штучного інтелекту та великих даних (Big Data). Основна мета цієї парадигми — створення інтелектуальних, взаємопов'язаних і самонавчальних виробничих систем, здатних адаптуватися до швидких змін умов і вимог ринку. У межах Industry 4.0 всі компоненти виробничих процесів об'єднані в єдину цифрову мережу, що дозволяє автоматизувати складні процеси, оптимізувати ресурси та підвищувати продуктивність. Особливу увагу приділено аналізу даних у реальному часі, що дає змогу прогнозувати можливі збої, підвищувати точність і забезпечувати якість кінцевої продукції. Такий підхід формує основу для інтеграції інноваційних рішень, зокрема в системах сортування, що є невіддільною частиною сучасних виробничих ліній.

Впровадження технологій Industry 4.0 [10] (рис. 1.6) в автоматизовані процеси сортування деталей за кольором на конвеєрі є важливим кроком у підвищенні ефективності сучасного виробництва. Завдяки використанню кіберфізичних систем (CPS), Інтернету речей (IoT), програмних рішень на базі Node-RED та інтеграції високотехнологічних контролерів створюється можливість досягти нового рівня точності, швидкості й адаптивності виробничих систем.

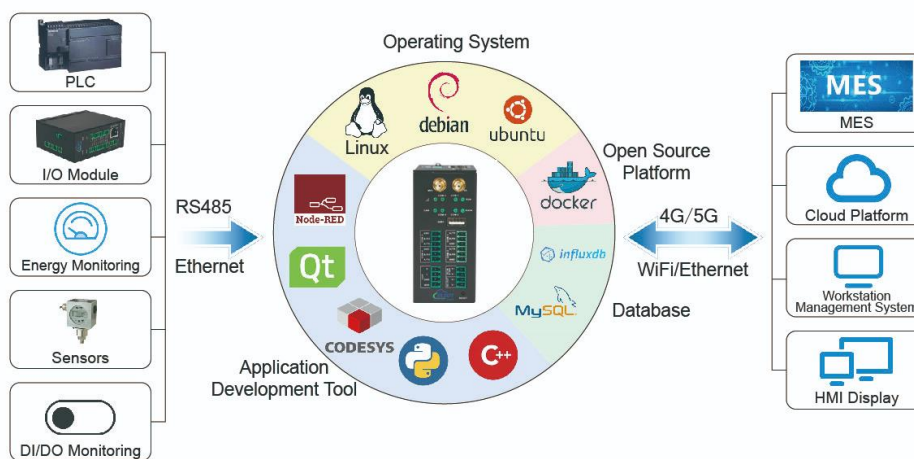


Рисунок 1.6 – Технології Industry 4.0 у виробництві

У контексті кваліфікаційної роботи, присвяченої розробці системи автоматичного сортування об'єктів за кольором з використанням пневматичного віддуву, було розглянуто можливості застосування технологій Industry 4.0 для інтелектуалізації цього процесу:

– *Використання кіберфізичних систем у процесі сортування деталей.* КФС є основою технологій Industry 4.0, адже вони забезпечують взаємодію між фізичними елементами (такими як сенсори, виконавчі механізми, конвеєри) та цифровими компонентами (системами аналізу даних, штучним інтелектом, програмним забезпеченням). У сортуванні деталей за кольором КФС інтегрують усі складові виробничого процесу в єдину інтелектуальну екосистему. [11]

Сенсори кольору, встановлені на конвеєрній стрічці, зчитують параметри поверхні деталей, такі як інтенсивність кольорів або спектральні характеристики. Ці дані передаються в реальному часі до центрального ПЛК, який аналізує їх, порівнює з еталонними значеннями та формує команди для виконавчих механізмів, наприклад, пневматичних приводів або роботизованих маніпуляторів. КФС дозволяють не тільки автоматизувати процес сортування, але й забезпечують його самонавчання: система може адаптувати параметри сенсорів до зміни освітлення, умов роботи чи характеристик деталей.

Ключова перевага кіберфізичних систем полягає у здатності виконувати самодіагностику. Наприклад, якщо один із компонентів, таких як сенсор або привід, виходить із ладу, система може автоматично виявити проблему, повідомити про неї оператору та, за можливості, переналаштувати інші компоненти для мінімізації впливу несправності.

– *Інтеграція Інтернету речей (IoT) у сортувальні системи.* Інтернет речей є важливим інструментом для забезпечення взаємодії між різними компонентами сортувальних систем. Підключення сенсорів, камер, контролерів та виконавчих механізмів до IoT-мережі дозволяє здійснювати обмін даними між усіма елементами системи в режимі реального часу. Це забезпечує ефективний моніторинг та управління всіма процесами на конвеєрі. [12]

Наприклад, сенсори кольору, підключені до IoT, передають дані про кожну деталь на хмарний сервер, де вони обробляються спеціалізованим програмним забезпеченням. Це дозволяє не лише сортувати деталі за кольором, але й здійснювати їх повне трасування, що особливо важливо для забезпечення якості та дотримання стандартів виробництва.

IoT також відкриває можливості для дистанційного управління сортувальними системами. Оператор або інженер може через мобільний додаток чи веб-інтерфейс отримувати актуальну інформацію про стан системи, діагностувати можливі проблеми або навіть змінювати параметри сортування без необхідності фізичної присутності на виробничій лінії.

– *Використання Node-RED для управління потоками даних.* Node-RED — це потужний інструмент для організації інтеграції даних у системах, побудованих на принципах Industry 4.0. [13] Ця платформа забезпечує візуальне програмування та дозволяє створювати складні потоки обробки інформації між сенсорами, контролерами та виконавчими механізмами.

У системах сортування деталей за кольором Node-RED виконує роль інтеграційного середовища, яке пов'язує всі компоненти системи в єдину інформаційну мережу. Наприклад, платформа може отримувати дані від сенсорів, виконувати їх попередню обробку, передавати результати на контролер і формувати команди для виконавчих механізмів. Крім того, Node-RED дозволяє створювати зручні візуальні інтерфейси, через які оператор може відслідковувати стан системи, переглядати статистику роботи та налаштовувати параметри.

– *Роль контролерів у реалізації сортувальних систем.* Програмовані логічні контролери (ПЛК) є центральним елементом сучасних систем сортування за кольором. Вони відповідають за обробку даних, що надходять від сенсорів, формування команд для виконавчих механізмів та забезпечення синхронізації між усіма компонентами системи.

Сучасні ПЛК підтримують різноманітні протоколи зв'язку, такі як MQTT, OPC-UA, Ethernet/IP, що дозволяє їм взаємодіяти з іншими пристроями та програмними компонентами в рамках IoT-інфраструктури. Завдяки високій

обчислювальній потужності контролери здатні працювати у режимі реального часу, що є критично важливим для забезпечення точного сортування на високошвидкісних конвеєрах.

Крім того, ПЛК можуть інтегруватися з системами штучного інтелекту [14], які аналізують накопичені дані та допомагають оптимізувати роботу сортувальної системи. Наприклад, контролер може прогнозувати потребу в технічному обслуговуванні або адаптувати параметри сортування залежно від змін у характеристиках об'єктів.

Інтеграція технологій Industry 4.0 у сортувальні системи забезпечує низку переваг:

- *Підвищення точності та швидкості.* Автоматизовані системи працюють із мінімальною затримкою, що дозволяє обробляти чисельні обсяги продукції.
- *Гнучкість та адаптивність.* Штучний інтелект і алгоритми самонавчання дозволяють системам пристосовуватися до нових умов роботи.
- *Ефективний моніторинг.* IoT забезпечує постійний контроль стану параметрів системи, а Node-RED дозволяє візуалізувати всі дані технологічного процесу в реальному часі.
- *Зменшення витрат.* Оптимізація роботи системи знижує витрати на енергію та обслуговування.

Інтеграція IoT, кіберфізичних систем, Node-RED та сучасних ПЛК забезпечує високу точність, швидкість і гнучкість сортувального процесу. Завдяки таким рішенням виробництва можуть досягати значних покращень у продуктивності, якості та ефективності роботи.

1.3 Огляд сучасних рішень у сфері автоматизованого сортування об'єктів за кольором

У даному пункті проаналізуємо існуючі рішення, які використовують різні підходи до розпізнавання кольорів, такі як RGB-датчики, спектральний аналіз і кольорові камери, а також їх інтеграцію з виконавчими механізмами, включаючи пневматичні приводи, роботизовані маніпулятори та механічні штовхачі.

Мета огляду — визначити сильні сторони та обмеження цих технологій, що дозволить обрати та обґрунтувати вибір методу розпізнавання кольорів деталей та засобу подальшого сортування для нашої розробки.

Нижче було наведено деякі з найбільш відомих систем сортування за кольоровими характеристиками, що застосовуються у виробництві, а також ключові патенти, що описують технології, які лежать в основі цих систем.

У першій статті [15] описано нову систему для розпізнавання та сортування промислових виробів за кольоровими характеристиками. Після натискання оператором кнопки запуску активуються дві стрічки конвеєрів, і система починає функціонувати. Коли об'єкт розміщується на стрічці, автоматично запускається процес сортування. Датчики кольору розташовані в такій послідовності: синій, зелений і сірий.

У разі розпізнавання датчиком заданого кольору система управління активує відповідний штовхач, який перенаправляє об'єкт у визначений жолоб. Якщо жоден із трьох датчиків не ідентифікує колір, об'єкт потрапляє до четвертого жолоба, призначеного для об'єктів, які не піддаються сортуванню. Ця процедура повторюється для кожного наступного об'єкта, що надходить на конвеєрну стрічку.

Як показано на рисунку 1.7, система сортування складається з двох конвеєрних стрічок різної довжини: перша має довжину 4 м, а друга — 2 м. Для розподілу об'єктів передбачено чотири жолоби, три штовхачі та три датчики для аналізу кольорових характеристик об'єктів. Крім того, система обладнана трьома колонками для кріплення датчиків.

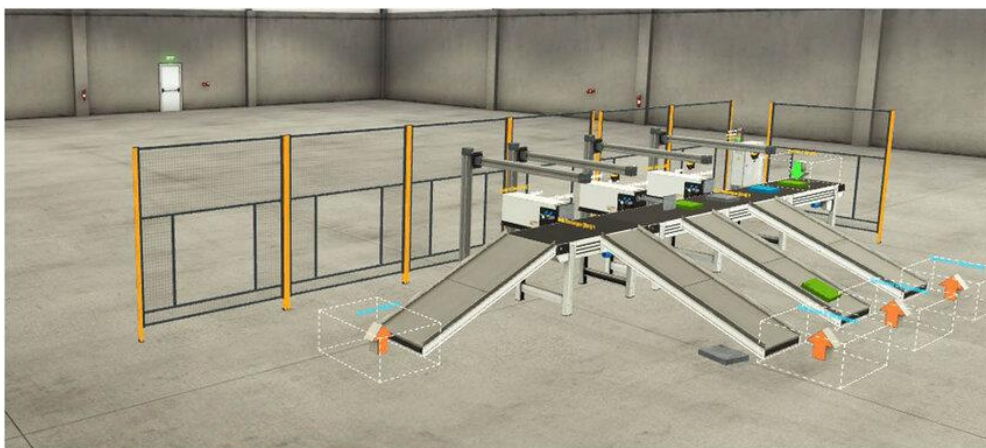


Рисунок 1.7 – 3D-модель системи в FACTORY I/O

Управління виконується за допомогою програмованого логічного контролера (ПЛК) Siemens CPU 1511-1 PN, а програмування та моделювання реалізовано у середовищі TIA PORTAL та FACTORY I/O. Стан роботи системи контролюється через інтерфейси SCADA і HMI, що показано на рисунку 1.8.

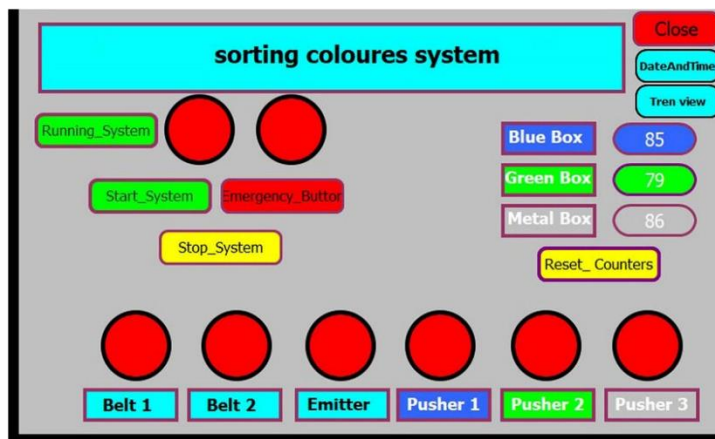


Рисунок 1.8 – Інтерфейс SCADA системи сортування

Додатково система оснащена кнопками скидання лічильників, зупинки та аварійного відключення. Вони забезпечують безпечне управління системою, дозволяючи вирішувати нештатні ситуації. За результатами роботи система відсортувала 250 об'єктів за 20 хвилин, включаючи 85 синіх, 89 зелених і 86 сірих (металевих) об'єктів.

Друга стаття описує розробку системи керування роботами лабораторного стенду, яка спрямована на контроль процесів автоматизації сортування деталей за кольоровими характеристиками. Система передбачає управління рухами затискних пристроїв роботів, які переміщують об'єкти з вхідної ємності до відповідних зон сортування залежно від кольору. Усі операції можуть виконуватися в ручному або автоматичному режимах. У ручному режимі рух кожної ланки роботів керується серводвигунами через потенціометри, що дозволяє оператору змінювати положення ланок у просторі. Автоматичний режим реалізується за допомогою програми, що завантажується через персональний комп'ютер і виконується за участі мінікомп'ютера Raspberry Pi та контролерів Arduino UNO. [16]

Raspberry Pi отримує команди через Wi-Fi, обробляє їх і передає сигнали управління до Arduino. Контролери використовують широтно-імпульсну

модуляцію (ШИМ) для керування 6 серводвигунами кожного робота. Роботи обмінюються інформацією через UART-протокол, що забезпечує синхронізовану роботу. Для розпізнавання кольору об'єктів застосовується датчик кольору, який визначає характеристики кожного об'єкта, після чого об'єкт спрямовується до відповідної ємності: для червоних, зелених або синіх об'єктів.

У перспективі планується інтеграція відеокамери для аналізу статично неупорядкованих об'єктів, що знаходяться в робочій зоні одного з роботів. Це вимагатиме створення складнішого алгоритмічного забезпечення для визначення параметрів розташування об'єктів і розширення функціональних можливостей системи. Функціональна схема системи керування промисловими роботами лабораторного стенда для сортування об'єктів за кольоровими характеристиками зображена на рисунку 1.9.

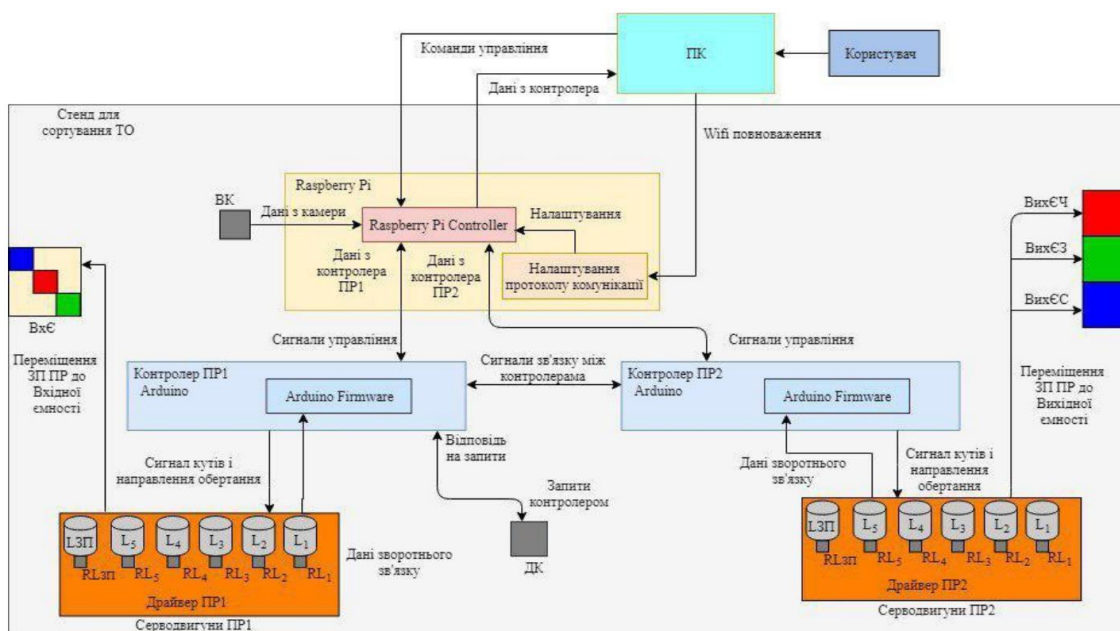


Рисунок 1.9 - Функціональна схема автоматизованої системи керування роботами стенду задля сортування об'єктів за кольором

В третій статті описано розроблення робочої моделі сортування круглих деталей на конвеєрних стрічках за допомогою ПЛК та пневмосистеми. Це дослідження відзначається поєднанням різних технологічних систем для розробки електропневматичного робота. У роботі створена інтелектуальна мехатронна конвеєрна система, яка автоматизує процес транспортування та розміщення

круглих об'єктів, що мають застосування в виробничих і пакувальних процесах. Всі керуючі операції реалізуються за допомогою програмованого електричного реле, яке називається міні-програмованим логічним контролером (PLC), а також датчика кольору, ручки захвату та електронних перемикачів. Для управління роботом, що відповідає за розміщення об'єктів, використовується електропневматична система. Пристрій складається з чотирьох основних компонентів: резервуара для деталей, першої конвеєрної стрічки, маніпулятора для переміщення деталей, другої конвеєрної стрічки та розподільчих рейок. Окрім цього, система включає додаткові елементи, такі як сенсори, компресори, повітряні резервуари, ПЛК та з'єднувальні елементи. [17] Усі ці компоненти зібрані в єдину систему, яка зображена на рисунку 1.10.

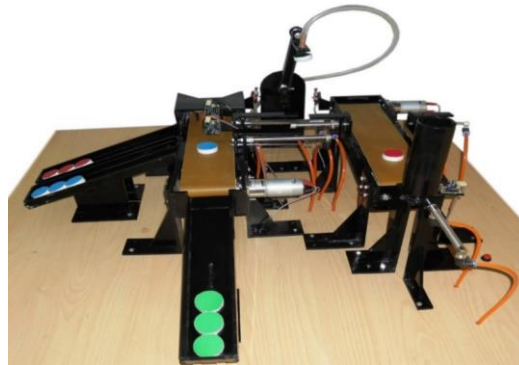


Рисунок 1.10 – Модель системи сортування, побудована на пневмоелементах

Система працює з трьома кольорами: червоним, синім і зеленим. На початку зразки поміщаються у резервуар, звідки пневматичний підйомник подає їх на перший конвеєр. Об'єкт рухається конвеєрною стрічкою, і два механізми направляють його в потрібну позицію на першому етапі. Після цього зразок приймає точні координати для подальшого переміщення. На кінці першої стрічки вакуумний маніпулятор на основі пневматичної конструкції переміщує об'єкт на другу стрічку. Після цього сенсор кольору визначає хроматичні характеристики деталі та дає команду на відповідний пневматичний механізм, який перенаправляє деталь на одну з рейок для сортування. Зелені зразки проходять другим конвеєром до кінцевої точки, де відділяються від синіх і червоних зразків, оскільки для них не передбачено додаткового розподілу.

Система показала високу надійність у розпізнаванні кольорів без затримок, а вакуумний механізм у маніпуляторі працює з високою точністю, не допускаючи помилок під час відбору деталей. Швидкість конвеєрних стрічок може бути змінена залежно від виробничих потреб, а система залишається стабільною навіть при високій швидкості.

1.4 Обґрунтування впровадження удосконалень у систему сортування

Аналіз існуючих систем сортування показав, що для лабораторних експериментів найефективнішим рішенням є використання RGB-датчика кольору та пневматичної системи. З огляду на простоту форми та однорідність кольору деталей, RGB-датчик забезпечить високу точність і швидкість сортування. Пневматична система, в свою чергу, оптимальна для переміщення об'єктів, так як вона поєднує високу швидкість і ефективність при мінімальних енергетичних витратах, що є критично важливим для тестувань у лабораторних умовах.

Отже, інтеграція RGB-датчика для розпізнавання кольору та пневматичної системи для сортування дозволяє створити високоінтелектуальну, точну та надійну систему, що підходить для проведення досліджень і тестувань у лабораторних умовах. З урахуванням цього, було ухвалено рішення про модернізацію наявної системи, використовуючи модель «маніпулятор-конвеєр» на лабораторному стенді Siemens в ауд. 367 Криворізького національного університету.

Модернізацією наявної системи стенду буде являтися додавання на початкову платформу об'єкта датчику кольору для створення автоматизованої системи сортування деталей за їх кольоровими характеристиками. У якості методу сортування буде обрано пневматичну систему, яка в залежності від кольору завдяки повітрю під високим тиском відправляти деталь на потрібний конвеєр. Також буде написано програмне забезпечення датчику для точного виявлення кольору деталі. Дана інформація повинна буде оброблятися проміжним пристроєм і відправлятися на логічний контролер лабораторного стенду. Після цього, на основі отриманих даних ПЛК буде керувати роботою пневмосистеми.

Інтеграція цієї системи дозволить не лише покращити ефективність і точність сортування, але й забезпечить високий рівень автоматизації та зменшить потребу в ручному втручанні, що зробить виробничий процес більш надійним і продуктивним.

Висновки до розділу:

Було проаналізовано особливості процесу сортування деталей за їх кольоровими характеристиками на виробництві, класифікацію систем по різних типах. Було розглянуто можливість інтеграції технологій Industry 4.0 у процес сортування. Аналіз існуючих систем автоматизованого сортування дозволив вивчити різноманітні конструкції та варіанти їх застосування в лабораторних умовах. Були оцінені їх переваги та недоліки, що стало підставою для ухвалення рішення про модернізацію лабораторного стенду Siemens у Криворізькому національному університеті. Ця модернізація не лише підвищить ефективність досліджень, але й створить нові можливості для проведення лабораторних робіт для студентів університету.

РОЗДІЛ 2

АНАЛІЗ ТА ВИБІР АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ СОРТУВАННЯ ЗА КОЛЬОРОМ

2.1 Структурний аналіз об'єкта керування лабораторного станду «маніпулятор-конвеєр»

Було здійснено детальне вивчення апаратної складової та технічної документації, що стосується експлуатації та програмування лабораторного станду №3, розташованого в аудиторії №367. Стенд оснащений сучасним контролером Simatic S7-1200 1215C (рис. 2.1), блоками живлення, комунікаційним модулем, частотним перетворювачем, який регулює швидкість руху конвеєра, електродвигунами конвеєра та маніпулятора, енкодером конвеєра, оптичними датчиками, маніпулятором із поворотною вежею та магнітним захватом, модулем зчитування RFID-міток, а також, конвеєрною стрічкою, платформою для розміщення вантажу та панеллю оператора НМІ. За оригінальним проектом роботи станду, scara-робот, в залежності від наявності мітки на деталі, виконує захоплення та підняття вантажу з платформи, переміщує його на конвеєрну стрічку, яка транспортує об'єкт у кінцеву точку.

Ключовим елементом лабораторного станду є **програмований логічний контролер Siemens Simatic S7-1200** із процесорним модулем 1215C DC/DC/DC, який забезпечує управління всіма компонентами системи через завантажений програмний код. Контролер характеризується високою функціональністю та надійністю, має ступінь захисту IP20 та монтується на стандартну DIN-рейку шириною 35 мм. Робочий температурний діапазон пристрою становить від 0 до 60°C, що дозволяє використовувати його в різних умовах експлуатації. [18]

Simatic S7-1200 підтримує підключення від 2 до 51 аналогового і від 14 до

284 цифрових каналів, що забезпечує значну гнучкість у конфігурації системи. Додатково, до контролера можна підключати комунікаційні модулі, сигнальні плати або модулі вводу-виводу для роботи з дискретними та аналоговими сигналами. Інтегрований Ethernet-інтерфейс дозволяє здійснювати зв'язок з іншими пристроями, такими як HMI панелі, сервера, SCADA системи, а також виконувати діагностику або взаємодію з іншими елементами автоматизації.

У складі стенду контролер працює разом із блоком живлення PM 1207 на 24 В та комутатором CSM 1277, який забезпечує обмін даними між елементами системи. Для під'єднання зовнішніх сенсорів та виконавчих механізмів до цифрових і аналогових входів-виходів використовуються зручні контакти з гвинтовими затискачами. Роз'єми на бокових поверхнях контролера дають змогу послідовно підключати до 8 додаткових модулів, розширюючи функціональні можливості системи.

Така модульна структура дозволяє адаптувати систему до конкретних завдань лабораторного стенду. Використання Simatic S7-1200 як центрального елемента забезпечує високу точність і швидкість обробки сигналів, необхідних для координації роботи маніпулятора, конвеєра, датчиків і виконавчих пристроїв. Цей контролер також сприяє інтеграції системи в складніші автоматизовані комплекси, що робить його універсальним рішенням для навчальних та дослідницьких цілей.



Рисунок 2.1 – Siemens Simatic S7-1200 CPU 1215C DC/DC/DC

Живлення контролера і всіх інших елементів підтримується завдяки **блоку живлення** від тієї ж компанії Siemens, **PM 1207 6EP1 332-1SH71** (рис. 2.2). [19]



Рисунок 2.2 – Блок живлення PM 1207 для лабораторного стенду

Блок живлення Siemens PM 1207 (6EP1 332-1SH71) є надійним джерелом постійного струму, спеціально розробленим для забезпечення електроживлення програмованих логічних контролерів Siemens Simatic S7-1200 та інших компонентів автоматизації. Пристрій характеризується високою надійністю, стабільністю та енергоефективністю, що робить його незамінним у складних умовах експлуатації. Вхідна напруга блоку варіюється від 85 до 264 В змінного струму або від 110 до 300 В постійного струму, що забезпечує універсальність у підключенні до різних джерел живлення. Вихідна напруга становить 24 В постійного струму при максимальному струмі 2,5 А, що дозволяє жити не лише контролер, але й периферійні пристрої, такі як датчики, актуатори або комунікаційні модулі.

Вбудована функція корекції коефіцієнта потужності (PFC) підвищує ефективність використання енергії та зменшує навантаження на електромережу. Пристрій має інтегровані захисні механізми від перевантаження, короткого замикання та перенапруги, що забезпечує безпечну роботу обладнання та тривалий термін служби. Його робочий температурний діапазон від -20°C до $+70^{\circ}\text{C}$ дозволяє використовувати блок живлення у широкому спектрі кліматичних умов. Компактні розміри та можливість монтажу на стандартну DIN-рейку 35 мм роблять установку швидкою та зручною, а наявність світлодіодних індикаторів дозволяє легко контролювати стан пристрою.

Для забезпечення зв'язку з окремими компонентами стенду до контролера підключено **4-канальний некерований комутатор CSM 1277** (рис. 2.3). Цей

пристрій виконує функцію мережевого концентратора, що дозволяє об'єднати кілька елементів системи в єдину Ethernet-мережу. Завдяки своїй компактності та простоті в експлуатації, комутатор ідеально підходить для лабораторних стендів, де необхідно забезпечити надійну взаємодію між НМІ панеллю, контролером, периферійними пристроями або іншими модулями автоматизації.



Рисунок 2.3 – Промисловий комутатор CSM 1277

Комутатор CSM 1277 має чотири порти Ethernet зі швидкістю передачі даних 10/100 Мбіт/с, що дозволяє підключати декілька пристроїв одночасно. Завдяки некерованій архітектурі, він не вимагає попередньої конфігурації, що значно спрощує процес установки та запуску. Компактні розміри пристрою дозволяють зручно розміщувати його у вузьких монтажних шафах або безпосередньо на стандартній DIN-рейці, що є важливим для обмеженого простору лабораторних або промислових систем.

Пристрій оснащений світлодіодними індикаторами стану, які відображають активність кожного порту та загальний стан мережі, що полегшує моніторинг і діагностику під час експлуатації. Комутатор має робочий температурний діапазон від $-40\text{ }^{\circ}\text{C}$ до $+70\text{ }^{\circ}\text{C}$, що дозволяє використовувати його не лише в лабораторних умовах, але й у промислових середовищах з підвищеними вимогами до надійності та стабільності роботи обладнання.

Для управління кроковим двигуном, який забезпечує рух конвеєрної стрічки, застосовується перетворювач частоти **Sinamics G120** із силовим модулем **PM240-2** (рис. 2.4). Цей пристрій відіграє ключову роль у регулюванні швидкості

та плавності роботи конвеєра, забезпечуючи високоточне налаштування параметрів приводу відповідно до заданих режимів.

Перетворювач Sinamics G120 розроблений для широкого спектра застосувань і підтримує гнучке налаштування завдяки модульній структурі. Силовий модуль PM240-2 забезпечує енергоефективну передачу електроенергії до двигуна, має вбудований захист від перенапруги, перевантаження та короткого замикання, що значно підвищує надійність роботи системи.

Перетворювач підтримує широкий діапазон напруги живлення, що дозволяє його використання в різних електричних мережах. Його модуль керування включає передові алгоритми керування двигуном, які забезпечують високу точність роботи, навіть у складних режимах навантаження. Крім того, Sinamics G120 має вбудовану систему енергозбереження, яка автоматично адаптує споживання енергії до поточних вимог системи. [20]



Рисунок 2.4 – Інтелектуальна панель оператора IOP-2 перетворювача Sinamics G120

У пристрої передбачено запуск майстра налаштування, який дозволяє зручно та послідовно конфігурувати основні параметри системи. Серед доступних для налаштування параметрів виділяються:

- Структура керування (Control structure) – вибір схеми управління системою;
- Задане значення та джерела команд (Defaults of the setpoint source and the command source) – визначення джерел введення команд і встановлення бажаних значень;

- Тип двигуна (Motor type) – вибір моделі двигуна;
- Технічні характеристики двигуна (Motor data) – введення основних параметрів, таких як потужність, напруга, частота тощо;
- Функції приводу (Drive functions) – активація та налаштування додаткових функцій для роботи приводу;
- Ключові параметри (Important parameters) – конфігурування критичних параметрів, що впливають на роботу системи;
- Розрахунок характеристик двигуна (Calculation of the motor data) – автоматичний підрахунок параметрів на основі введених даних;
- Резюме (Summary) – підсумкове відображення всіх введених і розрахованих параметрів.

Для забезпечення руху ланок маніпулятора застосовуються **три драйвери ТВ6560 V2** (рис. 2.5) у поєднанні з **трьома кроковими двигунами моделі 28BJ48-12-300-01** (рис. 2.6). Така конфігурація дозволяє реалізувати точне і плавне керування положенням маніпулятора, забезпечуючи його функціональність у складі автоматизованої системи.



Рисунок 2.5 – Драйвер двигуна крокового ТВ6560 V2

Драйвери ТВ6560 V2 є пристроями керування, спеціально розробленими для забезпечення плавного та точного переміщення крокових двигунів. Вони підтримують мікрокроковий режим роботи, що дозволяє значно підвищити роздільну здатність і точність керування двигунами. Ці драйвери здатні працювати з напругою живлення від 10 до 35 В і струмом до 3 А на фазу, що робить їх сумісними з широким спектром двигунів. ТВ6560 V2 оснащений вбудованими захистами від перегріву, перевантаження та короткого замикання, що забезпечує стабільну роботу навіть у складних умовах.

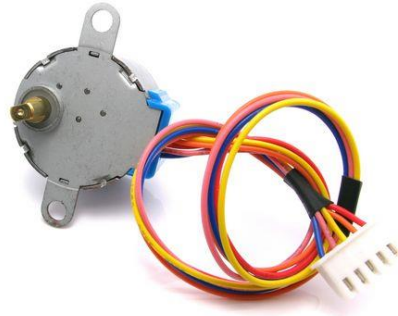


Рисунок 2.6 – Кроковий двигун маніпулятора 28BJ48-12-300-01

Крокові двигуни 28BJ48-12-300-01 характеризуються високою точністю та стабільністю роботи. Їх конструкція дозволяє реалізувати точне позиціонування без використання складних зворотних зв'язків, що значно спрощує систему керування. Ці двигуни мають високу моментну характеристику, що забезпечує достатню потужність для переміщення ланок маніпулятора навіть при роботі з навантаженням. Вони можуть працювати з низькою швидкістю обертання без втрати точності, що особливо важливо для задач, які потребують делікатного переміщення об'єктів.

Для контролю наявності вантажу на початковій та кінцевій ділянках конвеєрної стрічки встановлено два **оптичних датчики E3F-DS30C4** (рис. 2.7). У цьому випадку застосовуються оптичні датчики бар'єрного типу, які складаються з випромінювача і світловідбивача. Принцип роботи базується на аналізі переривання світлового потоку: коли об'єкт потрапляє в зону дії датчика, він блокує цей потік, що фіксується як подія наявності вантажу. Такий підхід забезпечує високу точність і надійність роботи, оскільки інфрачервоні промені менш чутливі до впливу навколишнього освітлення, пилу та інших перешкод.

Оптичний датчик E3F-DS30C4 має широкий діапазон робочих відстаней, що дозволяє адаптувати його до різних умов експлуатації. У даній конфігурації датчики встановлені так, щоб забезпечити контроль за вантажами на початковій і кінцевій ділянках конвеєра. Це дозволяє автоматизувати процеси моніторингу переміщення об'єктів і своєчасно передавати сигнали до контролера про наявність або відсутність вантажу.



Рисунок 2.7 – Кроковий двигун маніпулятора 28BJ48-12-300-01

Для моніторингу позиції конвеєра на його валу встановлений оптичний **інкрементальний енкодер моделі 6FX2001-4DB00** (рис. 2.8). Цей пристрій використовує оптичний принцип роботи для вимірювання обертів та кутового переміщення валу, що дозволяє точно визначати позицію конвеєра в реальному часі. Інкрементальний енкодер здійснює перетворення механічних рухів у електричні сигнали, які потім аналізуються для контролю точності переміщення.



Рисунок 2.8 – Енкодер конвеєру 6FX2001-4DB00

Енкодер 6FX2001-4DB00 працює за принципом інкрементального вимірювання, де положення валу визначається шляхом підрахунку імпульсів, що генеруються у відповідь на обертання. Оптична схема цього пристрою включає джерело світла, обертовий диск із кодovаними прорізами та фотоприймач. Під час обертання валу світловий потік переривається диском, створюючи послідовність імпульсів, кількість яких пропорційна кутовому переміщенню. Додатковий канал, розташований під кутом 90° до основного, забезпечує визначення напрямку обертання.

Підключення енкодера до контролера здійснюється через високошвидкісні цифрові входи, що дозволяє оперативно аналізувати сигнали та коригувати роботу привідної системи у разі відхилень. У лабораторному стенді енкодер використовується для моніторингу руху конвеєра, що дає змогу точно визначати його положення під час виконання операцій із сортування або транспортування вантажів. Це забезпечує синхронізацію роботи конвеєра з іншими компонентами системи, такими як маніпулятор чи оптичні датчики.

На рисунку 2.9 можна побачити повну систему «маніпулятор-конвеєр» лабораторного стенду № 3.

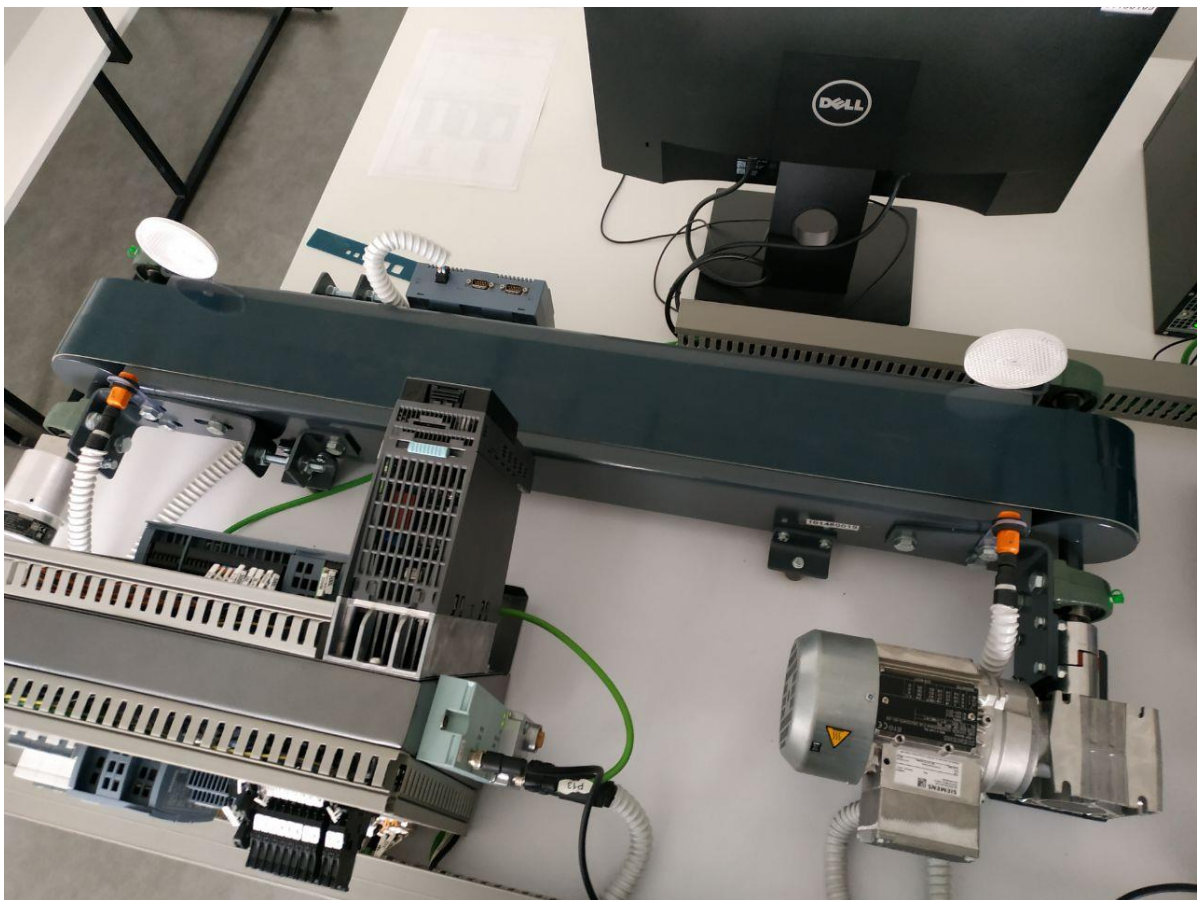


Рисунок 2.9 – Загальне зображення стенду разом із усіма його компонентами

За допомогою програмного забезпечення для проектування Eplan Education 2023 , було виконано креслення розмірностей параметрів лабораторного стенду. На рисунку 2.10 зображено маніпулятор, вид збоку. Даний маніпулятор має модульну конструкцію, яка складається з: циліндричної башти, плеча, стріли та підйомнику з магнітом.

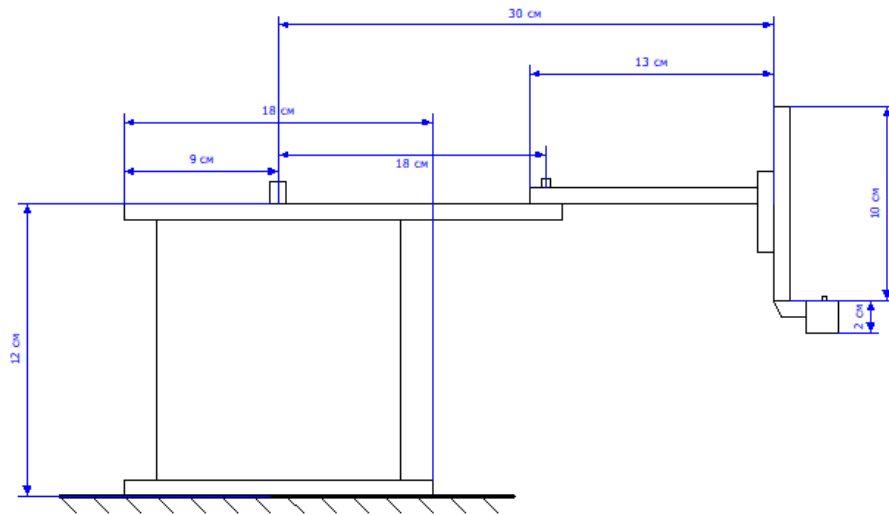


Рисунок 2.10 – Маніпулятор. Вид збоку

На рисунку 8 зображено маніпулятор, вид зверху. Дана схема зображує зону дії кожної частини маніпулятора. Плече маніпулятора має зону дії в 270° , а стріла – в 300° .

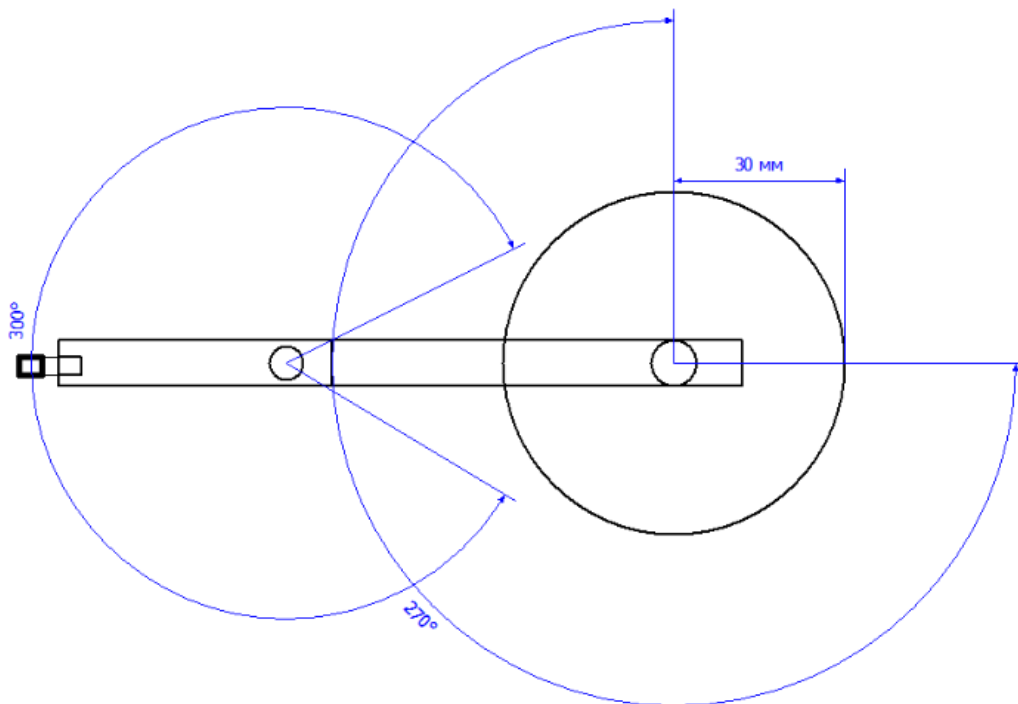


Рисунок 2.11 – Маніпулятор. Вид зверху

На рисунку 2.12 зображено підйомник маніпулятора з магнітом, вид збоку. Дана схема зображує максимальну та мінімальну висоту підняття підйомнику.

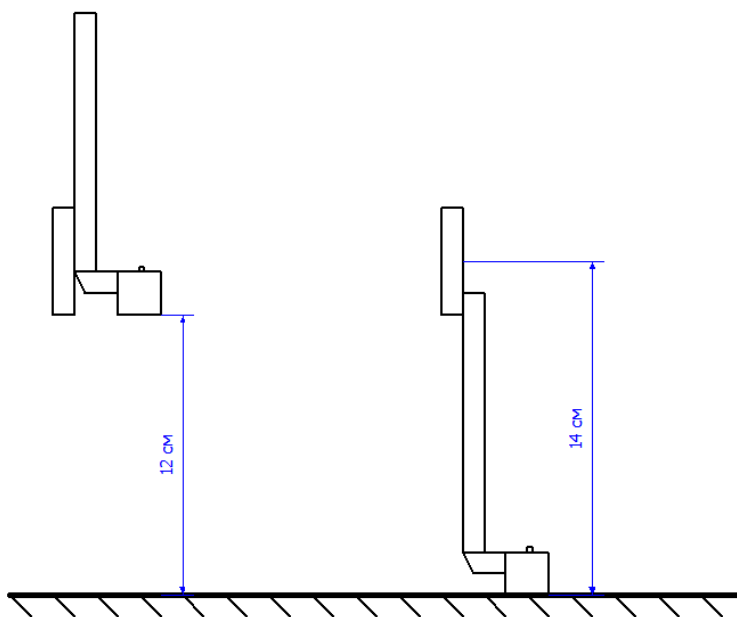


Рисунок 2.12 – Підйомник. Два стани

На рисунку 2.13 зображено конвеєрну стрічку, яка оснащена двома оптичними датчиками. Оптичні датчики контролюють вмикання та вимикання конвеєру. Перший датчик вмикає конвеєр, коли маніпулятор ставить деталь напроти нього на конвеєр. І так само, другий датчик вимикає конвеєр, коли він захватує наявність деталі перед ним.

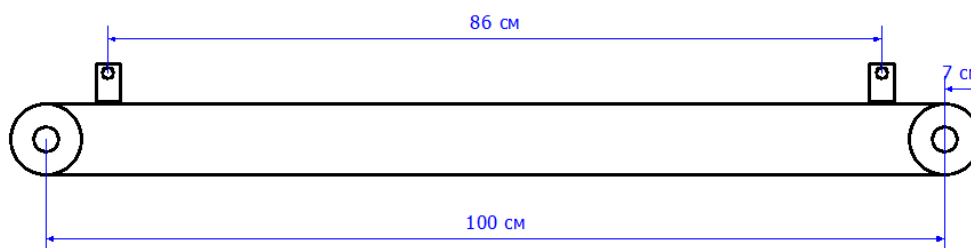


Рисунок 2.13 – Конвеєр. Вид збоку

Управління роботою стану здійснюється через НМІ панель, що дозволяє вибір між ручним та автоматичним режимами керування. Інтерфейс панелі оператора, призначений для управління станом, представлений на рисунку (рис. 2.14).

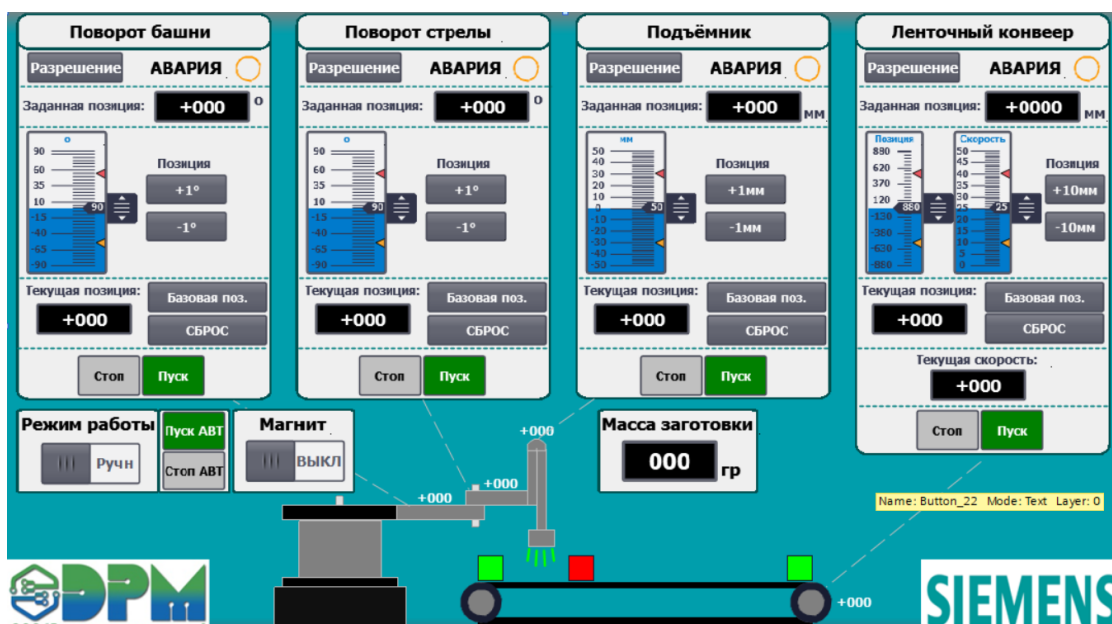


Рисунок 2.14 – Интерфейс НМІ панелі оператора

Для здійснення управління стандом у **ручному режимі** необхідно спочатку активувати кожен рухомий елемент, надаючи йому дозвіл на виконання операцій шляхом натискання кнопки «Разрешение». Після цього можна переходити до налаштування параметрів позицій кожного компонента за допомогою повзунків, або, для точнішого введення, скористатися кнопками «Позиция +1» і «Позиция -1». Після введення всіх необхідних значень для позицій потрібно натискати кнопку «Пуск», після чого маніпулятор або конвеєр переміщуються до заданої позиції. У разі потреби призупинити виконання задачі, слід натискати кнопку «Стоп». Для активації захоплення металевого вантажу використовується магніт, що включається за допомогою перемикача «Магнит ВКЛ/ВЫКЛ». У ручному режимі можливе визначення будь-якої траєкторії руху маніпулятора, з урахуванням фізичних обмежень кутів повороту, зазначених на кресленні (рис. 3).

Також у ручному режимі здійснюється налаштування базових позицій, що є важливим для забезпечення коректної роботи станду в автоматичному режимі. Це виконується шляхом ручного встановлення елементів станду на візуальні мітки позиції 0, після чого потрібно натискати кнопки «Базовая поз.».

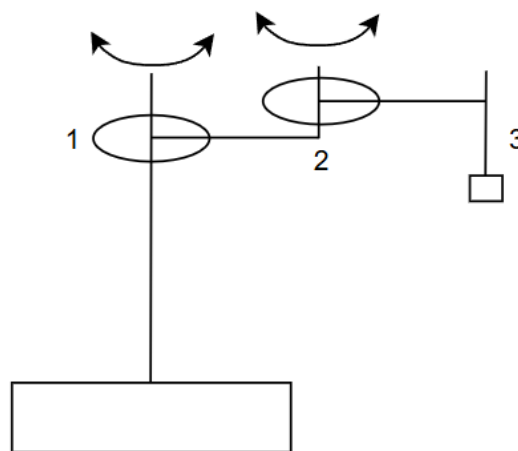
Для активації **автоматичного режиму** керування необхідно перевести перемикач «Режим работы» з положення «РУЧН» в стан «АВТО». Після цього

стенд почне виконувати програму, завантажену в головний контролер S7-1200 1215C. Програма автоматично налаштує необхідні дозволи, визначить параметри позицій та розпочне переміщення елементів стенду. В автоматичному режимі маніпулятор може виконувати різні траєкторії руху залежно від поточної програми, забезпечуючи автоматизацію всіх необхідних операцій.

2.2 Математичне забезпечення маніпулятора лабораторного стенду

Для початку розробимо математичну модель маніпулятора стенду, який має три ступені свободи. На рисунку 2.15 представлено спрощену кінематичну схему цього маніпулятора, що дозволяє чітко візуалізувати його механічну структуру та основні елементи.

Така модель слугує основою для аналізу кінематики маніпулятора, визначення його робочих характеристик і точності позиціонування.



1 – башта маніпулятора; 2 – стріла маніпулятора; 3 – підйомник маніпулятора

Рисунок 2.15 – Кінематична схема SCARA-роботу

На основі представленої кінематичної схеми буде виконано розрахунки змінних параметрів маніпулятора шляхом розв'язання оберненої задачі кінематики з використанням матричного методу. Для цього задаються координати кінцевого положення робочого органу:

$$X = 90, Y = -135, Z = 60.$$

Кінематична схема містить кінематичні пари, які включають як обертальні, так і поступальні компоненти. Це дозволяє відобразити характер руху ланок маніпулятора та забезпечити аналіз їх взаємодії для визначення необхідних параметрів керування.

Таблиця 2.1 – Параметри scara-роботу [20]

Номер кінематичної пари	Значення параметрів scara-роботу			
	Θ	S	d	α
1	Θ_1	S_1	d_1	0
2	Θ_2	S_2	$-d_2$	0

Для розв'язання прямої задачі кінематики необхідно сформулювати матриці A_1 та A_2 , які базуються на розширеній матриці A_i і характеризують зв'язок між $i-1$ та i -та системами координат.

Далі проводиться обчислення розширених матриць переходу для аналізованої кінематичної структури, використовуючи параметри, наведені у відповідній таблиці (табл. 2.2). Це дозволяє визначити взаємозв'язок між елементами системи та їх положення в просторі.

$$A_1 = \begin{bmatrix} \cos \Theta_1 & -\sin \Theta_1 * \cos 0 & \sin \Theta_1 * \sin 0 & S_1 * \cos \Theta_1 \\ \sin \Theta_1 & \cos \Theta_1 * \sin 0 & -\cos \Theta_1 * \sin 0 & S_1 * \sin \Theta_1 \\ 0 & \sin 0 & \cos 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$= \begin{bmatrix} \cos \Theta_1 & -\sin \Theta_1 & 0 & S_1 \cos \Theta_1 \\ \sin \Theta_1 & \cos \Theta_1 & 0 & S_1 \sin \Theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} \cos \Theta_2 & -\sin \Theta_2 * \cos 0 & \sin \Theta_2 * \sin 0 & S_1 * \cos \Theta_2 \\ \sin \Theta_2 & \cos \Theta_2 * \sin 0 & -\cos \Theta_2 * \sin 0 & S_1 * \sin \Theta_2 \\ 0 & \sin 0 & \cos 0 & -d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$$= \begin{bmatrix} \cos \Theta_2 & -\sin \Theta_2 & 0 & S_2 \cos \Theta_2 \\ \sin \Theta_2 & \cos \Theta_2 & 0 & S_2 \sin \Theta_2 \\ 0 & 0 & 1 & -d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Для розв'язання прямої задачі кінематики виконується послідовне множення матриць (2.1) і (2.2). У результаті цього обчислення, знаючи загальні координати, стає можливим визначити компоненти матриці T_2 . Це дозволяє точно встановити просторове положення захватного механізму маніпулятора.

$$\begin{aligned}
 T_2 &= \begin{bmatrix} \cos \Theta_1 & -\sin \Theta_1 & 0 & S_1 \cos \Theta_1 \\ \sin \Theta_1 & \cos \Theta_1 & 0 & S_1 \sin \Theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \\
 &\begin{bmatrix} \cos \Theta_2 & -\sin \Theta_2 & 0 & S_2 \cos \Theta_2 \\ \sin \Theta_2 & \cos \Theta_2 & 0 & S_2 \sin \Theta_2 \\ 0 & 0 & 1 & -d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3) \\
 &= \begin{bmatrix} \cos(\Theta_1 + \Theta_2) & -\sin(\Theta_1 + \Theta_2) & 0 & \cos(\Theta_1 + \Theta_2)S_2 + S_1 \cos \Theta_1 \\ \sin(\Theta_1 + \Theta_2) & \cos(\Theta_1 + \Theta_2) & 0 & \sin(\Theta_1 + \Theta_2)S_2 + S_1 \sin \Theta_1 \\ 0 & 0 & 1 & -d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

У підсумку координати положення захватного механізму визначаються за допомогою формул (2.4–2.6).

$$x_2^0 = \cos(\Theta_1 + \Theta_2)S_2 + S_1 \cos \Theta_1 \quad (2.4)$$

$$y_2^0 = \sin(\Theta_1 + \Theta_2)S_2 + S_1 \sin \Theta_1 \quad (2.5)$$

$$z_2^0 = -d_2 \quad (2.6)$$

Для розв'язання оберненої задачі кінематики необхідно прирівняти перші три елементи четвертого стовпця матриці результату (2.3) до заданих координат захоплювального механізму. У результаті цього отримаємо наступну систему рівнянь:

$$\begin{cases} x_2^0 = \cos(\Theta_1 + \Theta_2)S_2 + S_1 \cos \Theta_1 \\ y_2^0 = \sin(\Theta_1 + \Theta_2)S_2 + S_1 \sin \Theta_1 \\ z_2^0 = -d_2 \end{cases} \quad (2.7)$$

$$\begin{cases} 90 = \cos(\Theta_1 + \Theta_2)S_2 + S_1 \cos \Theta_1 \\ -135 = \sin(\Theta_1 + \Theta_2)S_2 + S_1 \sin \Theta_1 \\ 60 = -d_2 \end{cases} \quad (2.8)$$

$$\begin{cases} 90 = (\cos(\Theta_1 + \Theta_2) + \cos \Theta_1) * (S_2 + S_1) \\ -135 = \sin(\Theta_1 + \Theta_2)S_2 + S_1 \sin \Theta_1 \\ 60 = -d_2 \end{cases} \quad (2.9)$$

Використовуючи теорему Піфагора, обчислимо значення S_1 та S_2 :

$$\begin{aligned} (S_1 + S_2) &= \pm\sqrt{90^2 + (-135)^2} \\ &= \pm\sqrt{8100 + 18\,225} = \pm 162.3 \end{aligned} \quad (2.10)$$

$$90 = \cos(\Theta_1 + \Theta_2) + \cos \Theta_1 * 162.3 \quad (2.11)$$

$$\cos(\Theta_1 + \Theta_2) + \cos \Theta_1 = \frac{90}{162.3} = 0.5545 \quad (2.12)$$

$$\cos(\Theta_1 + \Theta_2) + \cos \Theta_1 = \arccos 0.3328 = 1.23 \text{ рад} = 70^\circ 33'$$

2.3 Математичне моделювання в середовищі Matlab

У середовищі програмування Matlab [21, 22] проведемо моделювання прямої та оберненої кінематичної задачі для маніпулятора. Попередньо визначимо довжини його ланок (рис. 2.16).

```
>> syms L_1 L_2 theta_1 theta_2 XE YE
>> L1=190;
>> L2=140;
```

Рисунок 2.16 – Визначення довжини ланок

Відобразимо координати X та Y у вигляді кутів Θ_1 та Θ_2 (рис. 2.17).

```
XE_RHS =
L_2*cos(theta_1 + theta_2) + L_1*cos(theta_1)
>> YE_RHS = L_1*sin(theta_1) + L_2*sin(theta_1+theta_2)
YE_RHS =
L_2*sin(theta_1 + theta_2) + L_1*sin(theta_1)
```

Рисунок 2.17 - X та Y координати як Θ_1 та Θ_2

Виконаємо трансформацію символічних виразів у функціональний формат (рис. 2.18).

```
>> XE_MLF = matlabFunction(XE_RHS, 'Vars', [L_1 L_2 theta_1 theta_2]);
>> YE_MLF = matlabFunction(YE_RHS, 'Vars', [L_1 L_2 theta_1 theta_2]);
```

Рисунок 2.18 – Трансформація виразів у функціональні залежності

Виконаємо обчислення прямої кінематики. Встановимо значення кутів робочої зони ланок маніпулятора та здійснимо їх конвертацію з градусної міри в радіанну (рис. 2.19).

```
>> t1_degs_row = linspace(-180,90,100);
>> t2_degs_row = linspace(-150,150,100);
>> [tt1_degs,tt2_degs] = meshgrid(t1_degs_row,t2_degs_row);
>> tt1_rads = deg2rad(tt1_degs);
>> tt2_rads = deg2rad(tt2_degs);
```

Рисунок 2.19 – Задавання початкових значень кутів та їх переведення в радіанну міру

За допомогою вбудованої функції Matlab обчислимо координати X та Y (рис. 2.20).

```
>> X_mat = XE_MLF(L1,L2,tt1_rads,tt2_rads);
>> Y_mat = YE_MLF(L1,L2,tt1_rads,tt2_rads);
```

Рисунок 2.20 - Обчислення координат X і Y

Для визначення зворотної кінематики скористаємося рівняннями прямої кінематики: сформулюємо систему рівнянь та розв'яжемо її відносно змінних Θ_1 та Θ_2 (рис. 2.21).

```
>> XE_EQ = XE == XE_RHS;
>> YE_EQ = YE == YE_RHS;
>> S = solve([XE_EQ YE_EQ], [theta_1 theta_2])

S =

struct with fields:

    theta_1: [2×1 sym]
    theta_2: [2×1 sym]
```

Рисунок 2.21 - Розв'язання рівнянь для визначення Θ_1 та Θ_2

Одержані результати аналізу зворотної кінематики для Θ_1 виведемо з відповідної структури (рис. 2.22).

```
>> simplify(S.theta_1)

ans =

2*atan((2*L_1*YE + (- L_1^4 + 2*L_1^2*L_2^2 + 2*L_1^2*XE^2 + 2*L_1^2*YE^2 - L_2^4 + 2*L_2^2*XE^2
2*atan((2*L_1*YE - (- L_1^4 + 2*L_1^2*L_2^2 + 2*L_1^2*XE^2 + 2*L_1^2*YE^2 - L_2^4 + 2*L_2^2*XE^2
+ 2*L_2^2*YE^2 - XE^4 - 2*XE^2*YE^2 - YE^4)^(1/2))/(L_1^2 + 2*L_1*XE - L_2^2 + XE^2 + YE^2))
+ 2*L_2^2*YE^2 - XE^4 - 2*XE^2*YE^2 - YE^4)^(1/2))/(L_1^2 + 2*L_1*XE - L_2^2 + XE^2 + YE^2))
```

Рисунок 2.22 - Отримання результату для Θ_1

Аналогічно виконаємо виведення рішень для Θ_2 (рис. 2.23).

```
>> simplify(S.theta_2)

ans =

-2*atan((- L_1^2 + 2*L_1*L_2 - L_2^2 + XE^2 + YE^2)
2*atan((- L_1^2 + 2*L_1*L_2 - L_2^2 + XE^2 + YE^2)

*(L_1^2 + 2*L_1*L_2 + L_2^2 - XE^2 - YE^2))^(1/2)
*(L_1^2 + 2*L_1*L_2 + L_2^2 - XE^2 - YE^2))^(1/2)

/(- L_1^2 + 2*L_1*L_2 - L_2^2 + XE^2 + YE^2))
/(- L_1^2 + 2*L_1*L_2 - L_2^2 + XE^2 + YE^2))
```

Рисунок 2.23 - Визначення результату для Θ_2

2.3 Вибір пристроїв та датчиків для розробки системи сортування

2.3.1 Вибір елементів визначення кольору для системи

Оскільки в рамках дослідження передбачається використання датчика кольору, необхідно забезпечити ефективну передачу даних з цього датчика до контролера стану. Для цього важливу роль відіграє використання перехідного пристрою, який виконує функцію з'єднання та адаптації сигналів між датчиком та основною системою управління.

Одним з таких рішень є застосування мікроконтролера **Arduino UNO R3** [11], який здатен зчитувати дані з датчика кольору, обробляти їх і передавати на основний контролер стану. Arduino UNO R3 була обрана для цієї системи завдяки

своїй доступності, простоті програмування та великій підтримці з боку спільноти. Вона ідеально підходить для лабораторних досліджень завдяки наявності всіх необхідних портів для підключення сенсорів, зокрема датчиків кольору. Мікроконтролер ATmega328 забезпечує необхідну обчислювальну потужність для обробки даних, що дозволяє ефективно використовувати систему в освітніх цілях.

Arduino UNO R3 (CH340) (рис. 2.24) на основі мікроконтролера ATmega328 є універсальною платформою для розробки електронних проектів і інтеграції сенсорних систем. Ця плата є економічно ефективним варіантом контролера, який завдяки використанню драйвера CH340 забезпечує надійний USB-зв'язок для програмування та обміну даними з ПК. [23]



Рисунок 2.24 – Зовнішній вигляд Arduino UNO R3 (CH340)

Мікроконтролер ATmega328, який є ядром плати, працює на тактовій частоті 16 МГц, має 32 КБ флеш-пам'яті для збереження прошивки, 2 КБ оперативної пам'яті SRAM та 1 КБ EEPROM для збереження даних. Це дозволяє ефективно обробляти дані з підключених сенсорів та виконувати алгоритми керування.

Живлення плати здійснюється через USB-кабель або зовнішній блок живлення з напругою 7–12 В, при цьому вбудований стабілізатор забезпечує роботу плати при різних входних напругах. Максимальний струм для вихідних цифрових пінів становить 40 мА, що дозволяє напряму підключати невеликі пристрої, такі як світлодіоди чи серводвигуни.

Для інтеграції з датчиком кольору Arduino UNO R3 пропонує широкий спектр інтерфейсів вводу/виводу, включаючи аналогові входи (6 каналів), цифрові входи/виходи (14 пінів, з яких 6 підтримують ШІМ-сигнали) та послідовні інтерфейси I2C і SPI. Такі можливості забезпечують гнучкість у зчитуванні параметрів датчика та передачі даних для подальшої обробки.

Кольорові датчики забезпечують надійні рішення для вирішення складних задач автоматизації та знаходять широке застосування в багатьох галузях промисловості, таких як виробництво продуктів харчування та напоїв, автомобілебудування та інших виробничих процесах. Їх функціональні можливості включають виявлення матеріалів, ідентифікацію кольорових маркерів на деталях, перевірку етапів виробничих процесів та інші важливі операції.

У промислових умовах, де вимагається висока точність, часто використовуються високоякісні кольорові датчики. Однак для лабораторних умов в університетах, де не завжди потрібна максимальна точність або складні налаштування, популярним варіантом є економніші альтернативи, наприклад, датчик кольору TCS230.

Датчик TCS230 [23] (також відомий як TCS3200) (рис. 2.25) отримав визнання завдяки своїй доступності, простоті використання та хорошій продуктивності в умовах лабораторних експериментів.

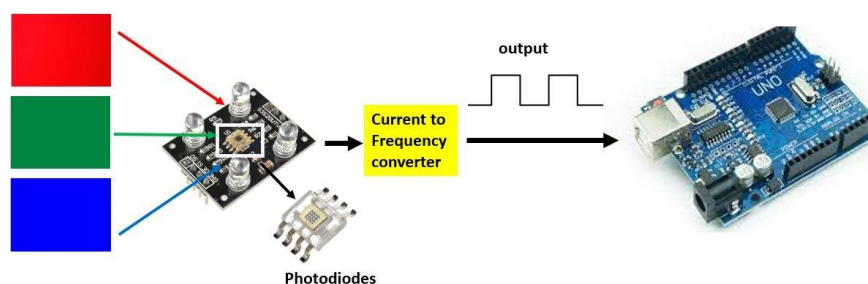


Рисунок 2.25 – Принцип дії датчика

Цей датчик, незважаючи на свою порівняно низьку вартість, здатен ефективно виконувати задачі виявлення кольору і кольорової ідентифікації, що робить його ідеальним для використання в навчальних проектах. Датчик працює на основі світлового аналізу кольору. Він випромінює світловий потік через вбудовані

світлодіоди, який потрапляє на поверхню об'єкта, а потім відбивається назад. Повернене світло поглинається датчиком, і його характеристики змінюються залежно від кольору об'єкта.

Мікроконтролер Arduino, з'єднаний із датчиком, обробляє отримані дані, які містять інформацію про інтенсивність відбитого світла у різних спектральних діапазонах. На основі цієї інформації Arduino визначає колір об'єкта з високою точністю.

Модуль кольорового датчика TCS230 має 8 основних контактів, серед яких S0, S1, S2, S3, вихідний контакт, контакт дозволу, заземлення (GND) та живлення (VCC).

Висновки S0 та S1 служать для встановлення масштабування вихідної частоти. Це дозволяє налаштувати частоту на одне з трьох попередньо визначених значень: 2%, 20% або 100%. Завдяки цій функції датчик можна легко інтегрувати з різними мікроконтролерами та іншими електронними пристроями. У таблиці 2.2 наведені комбінації станів S0 і S1 для отримання відповідних вихідних частот.

Таблиця 2.2 – Масштабування датчику

S0	S1	Максимальна вихідна частота	Відсоткова шкала
НИЗЬКИЙ	НИЗЬКИЙ	Вимкнення живлення	Вимкнення живлення
НИЗЬКИЙ	ВИСОКА	12 кГц	2%
ВИСОКА	НИЗЬКИЙ	120 кГц	20%
ВИСОКА	ВИСОКА	600 кГц	100%

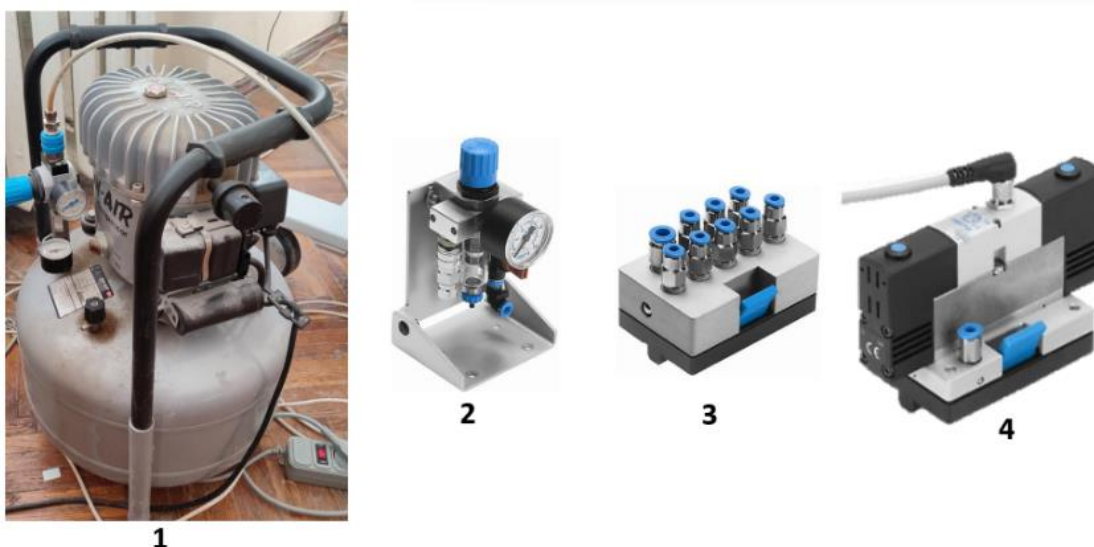
Контакти S2 і S3 відповідають за вибір колірної фільтра масиву фотодіодів. Наприклад, щоб визначити зелений колір, можна активувати 16 фотодіодів із червоним фільтром, встановивши обидва контакти в стан HIGH, як зазначено в таблиці. Аналогічно, залежно від обраних комбінацій S2 та S3, можна використовувати різні групи фотодіодів, що забезпечує гнучкість у роботі з різними кольорами. Всі можливі комбінації станів представлені в таблиці 2.3.

Таблиця 2.3 – Вибір кольору масиву

S2	S3	Тип фотодіода
НИЗЬКИЙ	НИЗЬКИЙ	Червоний
НИЗЬКИЙ	ВИСОКА	Синій
ВИСОКА	НИЗЬКИЙ	Без фільтра
ВИСОКА	ВИСОКА	Зелений

2.3.2 Вибір елементів сортування об'єктів на стрічковому конвеєрі

Для реалізації системи сортування пневмосоплами було використано частину обладнання електропневматичного лабораторного станду FESTO Didactic [24] з аудиторії 200а. Необхідні пневматичні елементи зображено на рисунку 2.26.



1 – масляний компресор; 2 – пусковий клапан; 3 - колектор 4 – х3/2 розподільник соленоїдний

Рисунок 2.26 – Використані елементи пневмоавтоматики станду

У якості джерела повітря під тиском виступає **масляний компресор JUN-AIR 6-25** (рис. 2.26-1). Цей пристрій забезпечує подачу повітря під тиском у резервуари системи, що слугують джерелом живлення для всіх пневматичних компонентів. Максимальний тиск, який можна досягти за допомогою даного обладнання, становить 16 атмосфер.

Пусковий клапан (рис. 2.26-2), оснащений регулювальним клапаном фільтра та манометром, є пристроєм для керування подачею або перекриттям

повітря під тиском у пневматичну систему. Керування здійснюється вручну за допомогою спеціального клапана. Інтегрований манометр забезпечує можливість контролю рівня тиску повітря, що подається в систему.

Колектор (рис. 2.26-3) є суто пневматичним елементом, призначеним для розподілу повітря у пневматичній системі. Його функція полягає в забезпеченні рівномірного розподілу потоку шляхом збільшення кількості точок подачі повітря. Вхідним отвором служить отвір із найбільшим діаметром серед усіх наявних.

3/2 соленоїдний розподільник (рис. 2.26-4), нормально закритий, слугує для регулювання потоку повітря до пневмоциліндрів шляхом перемикання внутрішнього клапана. Пристрій має один вхід і два виходи, обладнані клапанами для контролю. Електропідключення здійснюється через чотириконтактний провід: два контакти забезпечують живлення (червоний і синій), а два інших керують ввімкненням клапанів за нумерацією. Завдяки пружинному механізму, при відсутності сигналу клапани автоматично повертаються у вихідне положення, перекриваючи повітряний потік.

Необхідні електричні елементи зображено на рисунку 2.27.

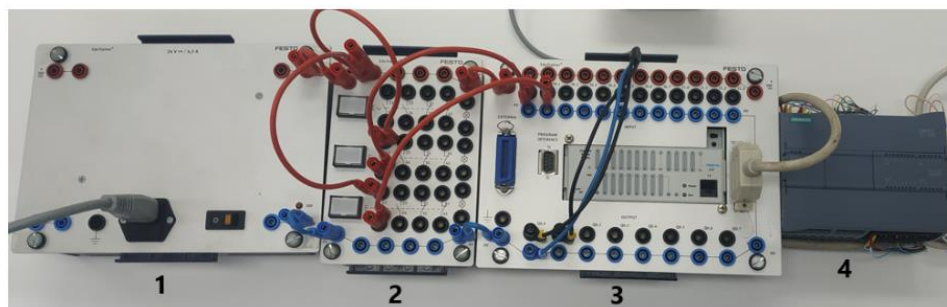


Рисунок 2.27 – Використані електричні елементи стенду

Блок живлення (2.27-1), що постачається разом зі стендом, є виробом компанії FESTO. Він отримує змінну напругу 220V і ефективно перетворює її в стабільну постійну напругу 24V, яка повністю забезпечує енергопостачання для пневматичної системи та контролера.

Блок введення електричних сигналів (рис. 2.27-2) представляє собою пристрій, що включає електричні входи та виходи, має три кнопки і відповідні набори контактів для керування ними. Загальна кількість контактів в цьому блоці

складає 12. Під час натискання на кнопку подається сигнал, що взаємодіє з відповідним блоком контактів, що забезпечує їх замикання або розмикання в залежності від типу дії.

Для забезпечення нормальної роботи блок потребує живлення напругою 24V, яку він пропускає через свої контакти при активації і передає на інші елементи системи.

ПЛК FECFC34 (рис. 2.27-3) – це компактний та потужний програмований логічний контролер (ПЛК), який можна легко адаптувати для промислових застосувань і швидко встановити. Контролер оснащений промисловим дизайном, має 12 цифрових входів і 8 цифрових виходів, забезпечуючи всі необхідні функції для автоматизації завдань, що вимагають лише кількох входів і виходів. Статус цих входів і виходів відображається за допомогою світлодіодних індикаторів.

Враховуючи несправність вбудованого контролера FECFC34 від компанії FESTO, для заміни в систему стенду було інтегровано інший контролер: **SIMATIC S7-1200 (CPU-1215C DC/DC/DC)** (рис. 2.27-4) від SIEMENS. Це значно розширило функціональні можливості електро-пневматичного стенду. Така інтеграція стороннього контролера не тільки відновила можливість програмування та управління роботою стенду через інтелектуальні реле, а й дозволила збільшити кількість доступних мов програмування, оскільки новий контролер підтримує чотири мови програмування.

Ще однією перевагою цього рішення є те, що новий контролер працює на одній і тій же напрузі 24V, яку постачає фірмовий блок живлення. Таким чином, система з новим контролером може житися від вбудованого блоку через комунікаційний кабель DB25, без необхідності додаткових допоміжних джерел живлення.

2.4 Математичне моделювання розпізнавання кольорів та керування пневмосоплами

Математичне забезпечення для датчика кольору TCS230 базується на аналізі

частотного сигналу, який генерується датчиком залежно від кольору об'єкта. Основні етапи включають перетворення світлового сигналу в частотний, вибірколірного фільтра, визначення частоти для кожного кольорового каналу (червоний, зелений, синій) та обчислення відносних інтенсивностей кольорів. [25]

1. Перетворення світла в частоту:

TCS230 використовує масив фотодіодів, які перетворюють інтенсивність світла на частоту. Частота f вихідного сигналу залежить від інтенсивності світла, що відбивається від об'єкта (2.13):

$$f = k * I \quad (2.13)$$

де f - вихідна частота, k - коефіцієнт пропорційності, який залежить від конфігурації масштабування (S_0 , S_1), I - інтенсивність світла, яке потрапляє на фотодіоди.

2. Масштабування вихідної частоти:

Вихідну частоту можна масштабувати до 2%, 20% або 100% залежно від станів S_0 і S_1 (2.14). Наприклад:

$$f_{scaled} = s * f \quad (2.14)$$

де s - коефіцієнт масштабування (0.02, 0.2 або 1).

3. Визначення інтенсивності кольору

Для кожного колірної каналу вимірюється (2.15) частота f_{color} , що пропорційна інтенсивності кольору:

$$I_{color} = \frac{f_{color}}{k * s} \quad (2.15)$$

де I_{color} - інтенсивність кольору (червоний, зелений, синій).

4. Розрахунок відносних інтенсивностей кольорів

Відносна інтенсивність кожного кольору може бути визначена як (2.16-18):

$$R = \frac{I_{red}}{I_{red} + I_{green} + I_{blue}} \quad (2.16)$$

$$G = \frac{I_{green}}{I_{red} + I_{green} + I_{blue}} \quad (2.17)$$

$$B = \frac{I_{blue}}{I_{red} + I_{green} + I_{blue}} \quad (2.18)$$

де R, G, B - нормалізовані інтенсивності червоного, зеленого та синього кольорів.

5. Ідентифікація кольору

Отримані значення R,G,B порівнюються з еталонними значеннями для визначення кольору об'єкта. Це можна записати як (2.19):

$$Color = \arg \min_c \sqrt{(R - R_c)^2 + (G - G_c)^2 + (B - B_c)^2} \quad (2.19)$$

де R_c, G_c, B_c - еталонні значення для кольору c , Color - визначений колір.

Пневматичне сопло є ключовим компонентом системи, що забезпечує точний вплив повітряного потоку на об'єкт для його переміщення з основного конвеєра на інший. Математичне забезпечення включає розрахунок динаміки потоку, необхідного тиску, витрат повітря, а також моделювання траєкторії деталі.

1. Формування повітряного потоку

Рух повітряного потоку можна описати за рівняннями Бернуллі та законами аеродинаміки. Загальний тиск повітряного потоку P_t визначається як сума статичного P_s та динамічного P_d тисків (2.20): [26]

$$P_t = P_s + \frac{1}{2} \rho v^2 \quad (2.20)$$

де ρ - густина повітря ($\text{кг}/\text{м}^3$), v - швидкість повітряного потоку ($\text{м}/\text{с}$).

2. Швидкість повітряного потоку

Швидкість повітряного потоку v через сопло залежить від робочого тиску P і діаметра сопла D . Швидкість повітряного потоку через сопло залежить від різниці тисків у системі та атмосферного тиску, а також від щільності повітря (2.21):

$$v = \sqrt{\frac{2(P - P_a)}{\rho}} \quad (2.21)$$

P - тиск у системі (Па), P_a - атмосферний тиск (Па).

3. Витрата повітря

Тепер, витрату повітря Q , яка визначається через швидкість потоку, можна виразити через площу поперечного перерізу (2.22):

$$Q = A * v \quad (2.22)$$

де A - площа перерізу сопла ($A = \frac{\pi * D^2}{4}$), v - швидкість потоку (м/с).

Для того, щоб розрахувати тиск, який необхідно подати через пневмосопло для того, щоб здути деталь вагою 200 г, необхідно врахувати кілька важливих факторів.

1. Визначення сили, необхідної для переміщення деталі

Сила, яка потрібна для переміщення деталі, залежить від її ваги. Вага деталі $m = 200 \text{ г} = 0.2 \text{ кг}$. Сила тяжіння визначається за формулою 2.23.

$$F = m * g \quad (2.23)$$

де m - маса деталі (кг), g - прискорення вільного падіння (приблизно 9.81 м/с^2).

Отже, сила буде:

$$F = m * g = 0.2 * 9.81 = 1.962 \text{ Н} \quad (2.23)$$

2. Визначення тиску, необхідного для переміщення деталі через сопло

Для того, щоб деталь переміщувалася, повітря, що виходить із сопла, повинно створювати таку силу, яка здатна зрушити деталь. Сила, що створюється

поток повітря через сопло, визначається за допомогою рівняння для повітряного потоку та тиску (2.24):

$$F = P * A \quad (2.24)$$

де F — сила, що діє на деталь (Н), P — тиск, що подається через сопло (Па), A — площа вихідного перерізу сопла (м^2).

Площа перерізу сопла визначається за формулою для кола (2.25):

$$A = \frac{\pi * D^2}{4} \quad (2.25)$$

де D — діаметр сопла (м).

Звідси можна виразити тиск, який необхідно подати для здування деталі (2.26):

$$P = \frac{F}{A} = \frac{4F}{\pi * D^2} \quad (2.26)$$

3. Розрахунок тиску

Для того, щоб розрахувати необхідний тиск, потрібно знати діаметр сопла D . Припустимо, що діаметр сопла $D = 0.0025$ м. Тоді площа перерізу буде:

$$A = \frac{\pi * 0.0025^2}{4} = 3.1416 * \frac{0.00000625}{4} = 0.0000049088 \text{ м}^2 \quad (2.25)$$

Тепер можемо обчислити необхідний тиск:

$$P = \frac{F}{A} = \frac{4F}{\pi * D^2} = \frac{4 * 1.962}{\pi * 0.0025^2} = \frac{7.848}{0.000019635} = 399,694 \text{ Па} \quad (2.26)$$

Переведемо тиск у бари (2.27):

$$P_{\text{бар}} = \frac{P}{100000} = \frac{399,694}{100000} = 3.997 \text{ бар} \quad (2.27)$$

2.5 Обґрунтування вибору програмного забезпечення для реалізації системи сортування

Для розробки системи автоматичного сортування об'єктів за кольором було обрано комплексне програмне забезпечення, що забезпечує гнучкість, надійність та простоту інтеграції. Програмування датчика кольору, підключеного до плати Arduino, реалізовано мовою **Python**. Python було обрано через його простий синтаксис, доступ до широкого спектра бібліотек для роботи з датчиками та зручність розробки алгоритмів обробки даних.

Node-RED використовується для організації потоків даних між Arduino, контролерами та іншими компонентами системи. Цей інструмент забезпечує візуальне середовище для швидкої інтеграції компонентів, що значно спрощує реалізацію системи.

Для моніторингу даних і налагодження роботи системи використовується **UAExpert** як клієнт для перегляду інформації, переданої через протокол OPC-UA. **OPC Server** забезпечує стандартизовану передачу даних між різними пристроями, створюючи універсальну платформу для взаємодії обладнання. [27]

Для управління пневмосоплами та реалізації логіки системи сортування було обрано середовище **TIA Portal**. Воно дозволяє програмувати контролери SIMATIC S7-1200, забезпечуючи гнучкість у створенні програм керування, діагностику системи та інтеграцію з іншими компонентами.

Важливим елементом системи є використання **інтерфейсу S7 PUT/GET** для прямого обміну даними між контролерами та іншими пристроями. Цей підхід забезпечує високошвидкісну передачу інформації, що є критично важливим для оперативного керування процесом сортування.

Поєднання цих інструментів забезпечує надійне функціонування системи, можливість її масштабування та інтеграцію з новими технологічними рішеннями у майбутньому.

Висновки до розділу:

Було проведено аналіз апаратного та програмного забезпечення для розробки автоматизованої системи сортування за кольором. Виконано структурний аналіз лабораторного стенду «маніпулятор-конвеєр» та розроблено математичну модель його роботи, що оптимізує рух маніпулятора. Проведено моделювання системи в середовищі Matlab, що дозволило оцінити ефективність алгоритмів. Обґрунтовано вибір обладнання, включаючи датчики кольору та механізми сортування, з урахуванням точності, швидкодії та вартості. Розроблено математичну модель для розпізнавання кольорів і керування пневмосоплами. Обґрунтовано вибір програмного середовища, що забезпечує інтеграцію компонентів і налаштування алгоритмів. Цей розділ закладає основу для створення ефективної системи сортування.

РОЗДІЛ 3

ПРОГРАМНО-ТЕХНІЧНА РЕАЛІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ ПРОЦЕСОМ СОРТУВАННЯ

3.1 Проектування схем автоматизації системи сортування

За допомогою AutoCad було реалізовано схематичну документацію до проекту. Структура проекту складається з: макетної схеми вузлу аналізу кольору та структурної схеми усієї системи. [28]

Було розроблено **структурну схему роботи системи** (рис. 3.1), яка графічно відображає ключові компоненти, що використовуються у функціонуванні системи. Стенд поділений на шість основних частин: модулі управління, аналіз кольору, сортування, маніпулятор, RFID зчитування і конвеєр.

До складу модулів управління входить базовий елемент – програмований логічний контролер (ПЛК), який забезпечує координацію роботи всіх компонентів системи. Контролер виконує функцію як джерела керуючих сигналів для інших елементів, так і обробника зворотного зв'язку, що надходить від датчиків, зокрема енкодера, датчику RFID та датчику кольора.

Для взаємодії з деякими компонентами використовуються додаткові проміжні пристрої. Наприклад, панель оператора підключена до ПЛК через комунікаційний модуль, а кроковий двигун конвеєра підключено через зовнішній частотний перетворювач. Водночас керування кроковими двигунами маніпулятора здійснюється безпосередньо через драйвери двигунів, які отримують сигнали від внутрішнього частотного перетворювача контролера.

Вузол розпізнавання кольору складається з датчику кольору, який аналізує характеристики кольору кожної деталі і передає дані до мікроконтролера Arduino. Плата Arduino обробляє отримані дані та через USB-з'єднання передає результати на персональний комп'ютер.

На ПК встановлено програмне забезпечення Node-RED, яке збирає дані від Arduino, обробляє їх і формує команди для подальшого управління системою.

Отримана з ПК інформація передається до основного контролера, який відповідає за управління виконавчими механізмами, такими як маніпулятор і пневматична система. Маніпулятор використовується для переміщення деталей на конвеєр, після цього конвеєр переміщає об'єкт. У випадку, якщо деталь потребує сортування, контролер передає команди до допоміжного контролера.

Допоміжний контролер координує роботу пневматичної системи через інтерфейс S7 PUT/GET. Ця система включає пневмосопла, яке створює потік стисненого повітря для відхилення деталей на інший конвеєр або у відповідний контейнер.

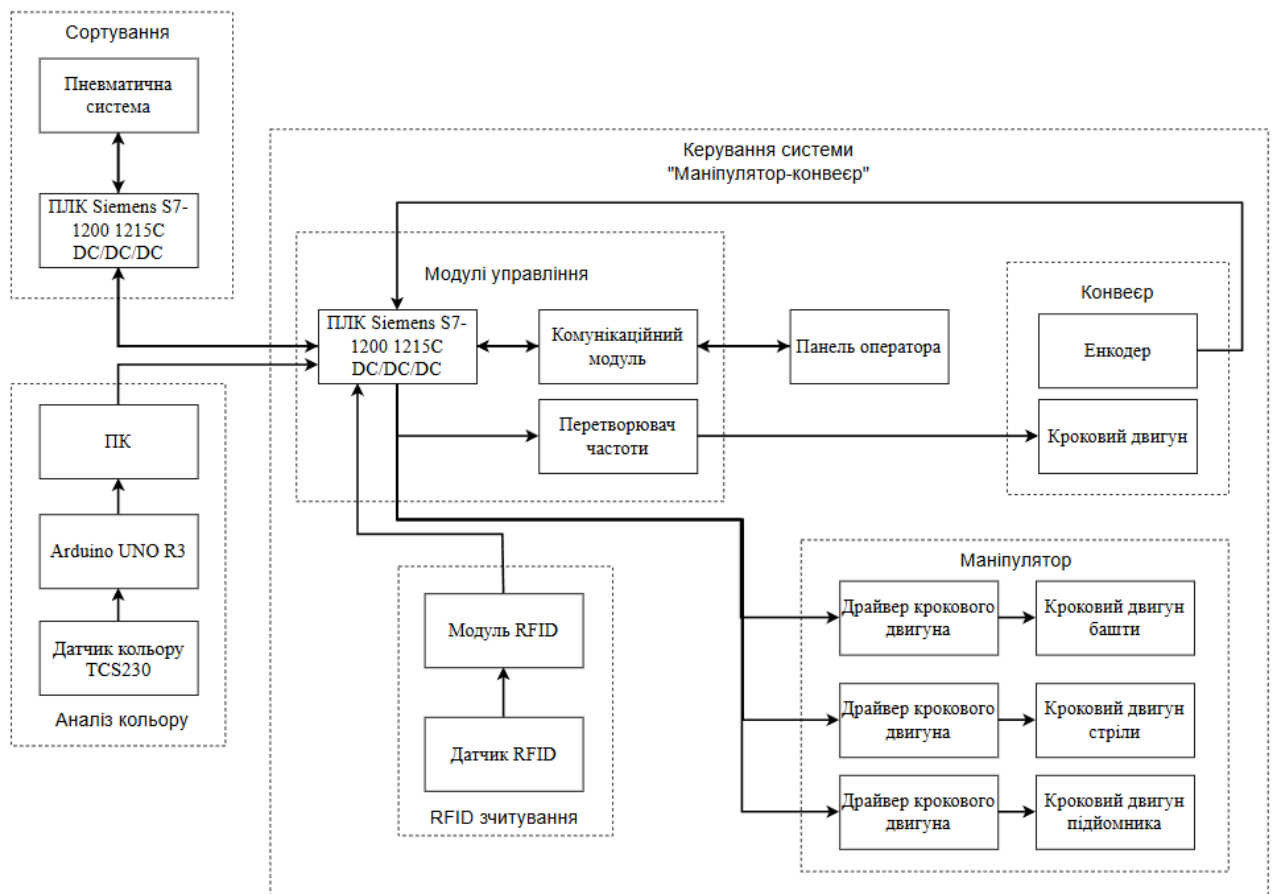


Рисунок 3.1 - Структурна схема системи сортування

Макетна схема підключення датчика кольору до мікроконтролера Arduino (рис. 3.2). Така схема дозволяє реалізувати основний функціонал: зчитування даних

з датчика кольору, їх обробку та передачу відповідних сигналів іншим компонентам системи.

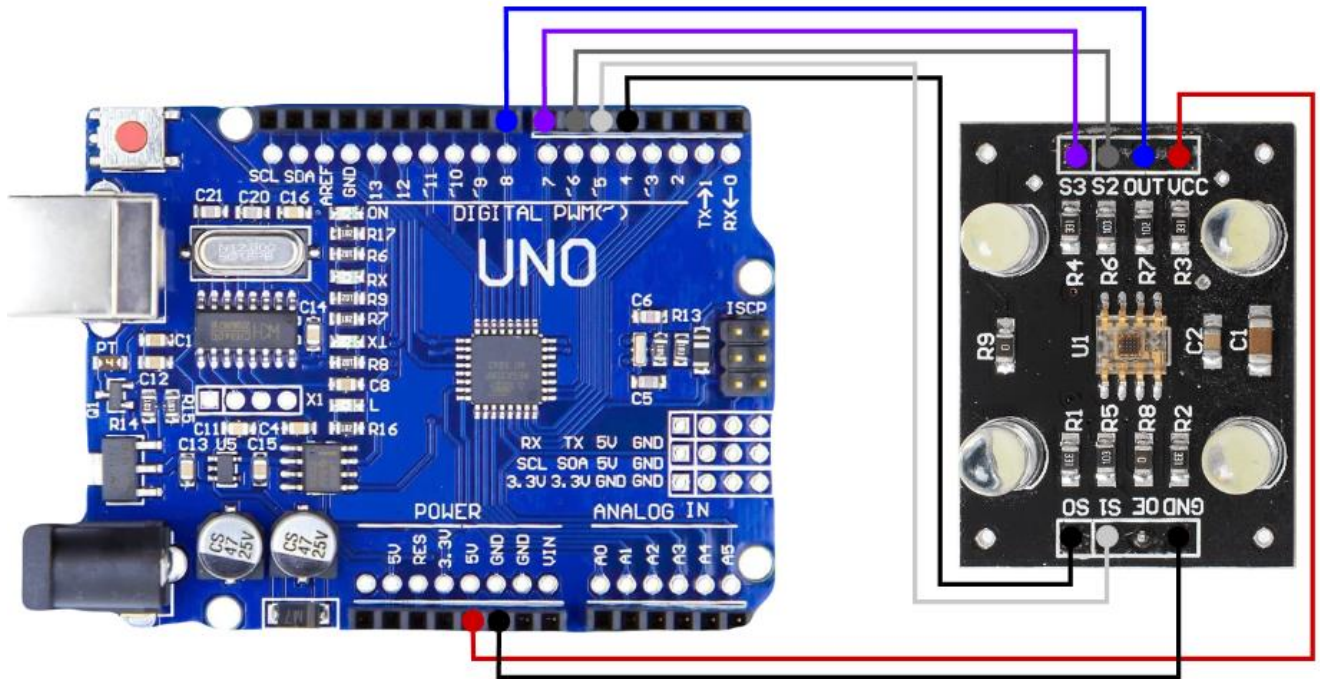


Рисунок 3.2 – Макетна схема системи аналізу модулю

Як було розглянуто раніше, модуль TCS230 має вісім терміналів, проте для підключення до плати Arduino буде використано лише сім з них. Контакт Output Enable (OE) залишиться непідключеним. У нашій схемі OE залишиться непідключеним, тому що за замовчуванням (у стані "плаваючого" контакту) датчик працює в активному режимі. Це спрощує підключення, оскільки немає необхідності додатково керувати цим контактом, а всі сигнали з датчика передаються без перешкод.

Датчик кольору працює в діапазоні напруги від 2,7 В до 5,5 В, тому термінал VCC під'єднаємо до 5 В, що спільний із платою Arduino. Терміни S0, S1, S2, S3 і OUT підключатимуться до цифрових входів плати Arduino. У нашому випадку використовуються GPIO4, GPIO5, GPIO6, GPIO7 і GPIO8 відповідно. Додатково, термінал GND датчика з'єднується із загальним контактом GND на платі Arduino.

Макетна схема підключення другого контролеру до пневмосистеми зображена на рис. 3.3. Дане підключення реалізовано через кабель комунікаційний кабель DB25, без застосування додаткових пристроїв живлення. [29]

На рисунку 3.3 зображено принцип комунікації контактів кабеля db25 з контактами контролера S7-1200 CPU 1215C DC/DC/DC від Siemens. Номер на рисунку відповідає номеру контакту на контролері (таб. 3.1).

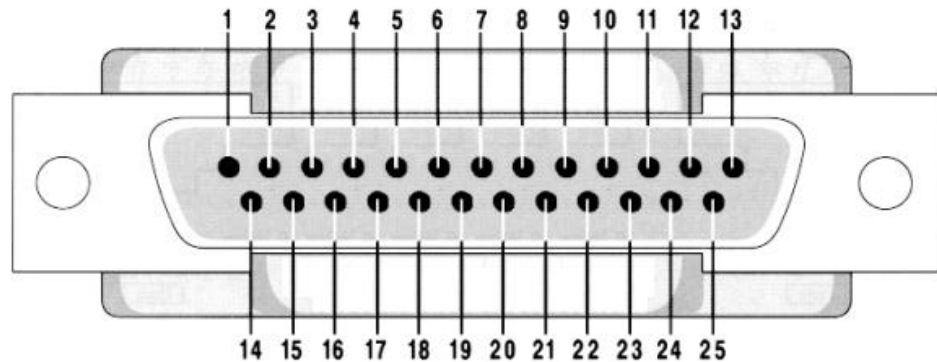


Рисунок 3.3 – Піни кабеля db25

Таблиця 3.1 – Відповідність контактів

Номер піна	1	2	3	4	5	6	7	8
Назва контакту контролера	DI.a %I0.0	DI.a %I0.1	DI.a %I0.2	DI.a %I0.3	DI.a %I0.4	DI.a %I0.5	DI.a %I0.6	DI.a %I0.7
Номер піна	9	10	11	12	13	14	15	16
Назва контакту контролера	DI.b %I0.0	DI.b %I0.1	DI.b %I0.2	DI.b %I0.3	DI.b %I0.4	DI.b %I0.5	L+ ↓	M ↓
Номер піна	17	18	19	20	21	22	23	24
Назва контакту контролера	Заземлення	4L+	4M	DQ.a %Q0. 0	DQ.a %Q0. 1	DQ.a %Q0. 2	DQ.a %Q0. 3	DQ.a %Q0. 4
Номер піна	25							
Назва контакту контролера	DQ.b %Q0. 4							

3.2 Алгоритм роботи системи сортування по кольору

Спочатку користувач активує програму на панелі оператора, перемикаючи повзунок у стан «авт.» (автоматичний режим). Після цього натискається кнопка «Пуск АВТ», яка запускає програму. Система переходить в режим очікування, чекаючи, поки деталь буде поставлена на конвеєр. Якщо деталь вже знаходиться на конвеєрі до натискання кнопки «Пуск АВТ», система автоматично розпочинає транспортування деталі.

Датчик кольору, який підключений до плати Arduino, постійно сканує деталі, що рухаються по конвеєру, і передає отриману інформацію в Arduino. Arduino обробляє ці дані та визначає колір деталі, яку він отримує. Залежно від кольору деталі (червоний, зелений, синій тощо), Arduino визначає тип деталі і передає отриману інформацію через COM-порт у комп'ютер, де працює Node-RED.

Node-RED, отримавши дані через COM-порт, обробляє їх у відповідному потоці (flow). На основі значення кольору, Node-RED формує потрібну інформацію для подальшої передачі контролеру. UaExpert, у свою чергу, є OPC клієнтом, що підключається до OPC сервера. UaExpert використовується для моніторингу і запису значень кольору, які надходять від Node-RED. Цей процес дозволяє адресувати тег в OPC сервері, де зберігається значення кольору, яке передається контролеру.

Контролер отримує через OPC сервер значення кольору, яке було передано через UaExpert. Крім того, контролер має інформацію про поточну позицію деталі на конвеєрі, що дозволяє йому прийняти рішення про подальші дії. Якщо, наприклад, деталь червона і її позиція на конвеєрі знаходиться в певному місці, контролер передає сигнал на другий контролер для активації пневмосопла.

Другий контролер отримує сигнал від основного контролера і передає команду на пневматичну систему. Пневматична система активує відповідне пневмосопло, яке віддуває деталь у вказану корзину, залежно від її кольору, наприклад, червону деталь направляє в корзину для червоних деталей. Після того,

як деталь буде відправлена у потрібну корзину, система знову переходить у режим очікування наступної деталі.

Таким чином, вся система працює у автоматичному режимі, забезпечуючи точне сортування деталей по кольору за допомогою комунікації між датчиком кольору, Arduino, Node-RED, UaExpert та контролерами. UaExpert в цій системі виконує роль OPC клієнта, що забезпечує налаштування та моніторинг передачі даних між Node-RED і контролером, дозволяючи записувати значення кольору у відповідний тег контролера. Це дозволяє забезпечити точність і ефективність автоматизованого процесу сортування деталей.

На рисунку 3.4 зображено блок-схему алгоритму роботи системи сортування.

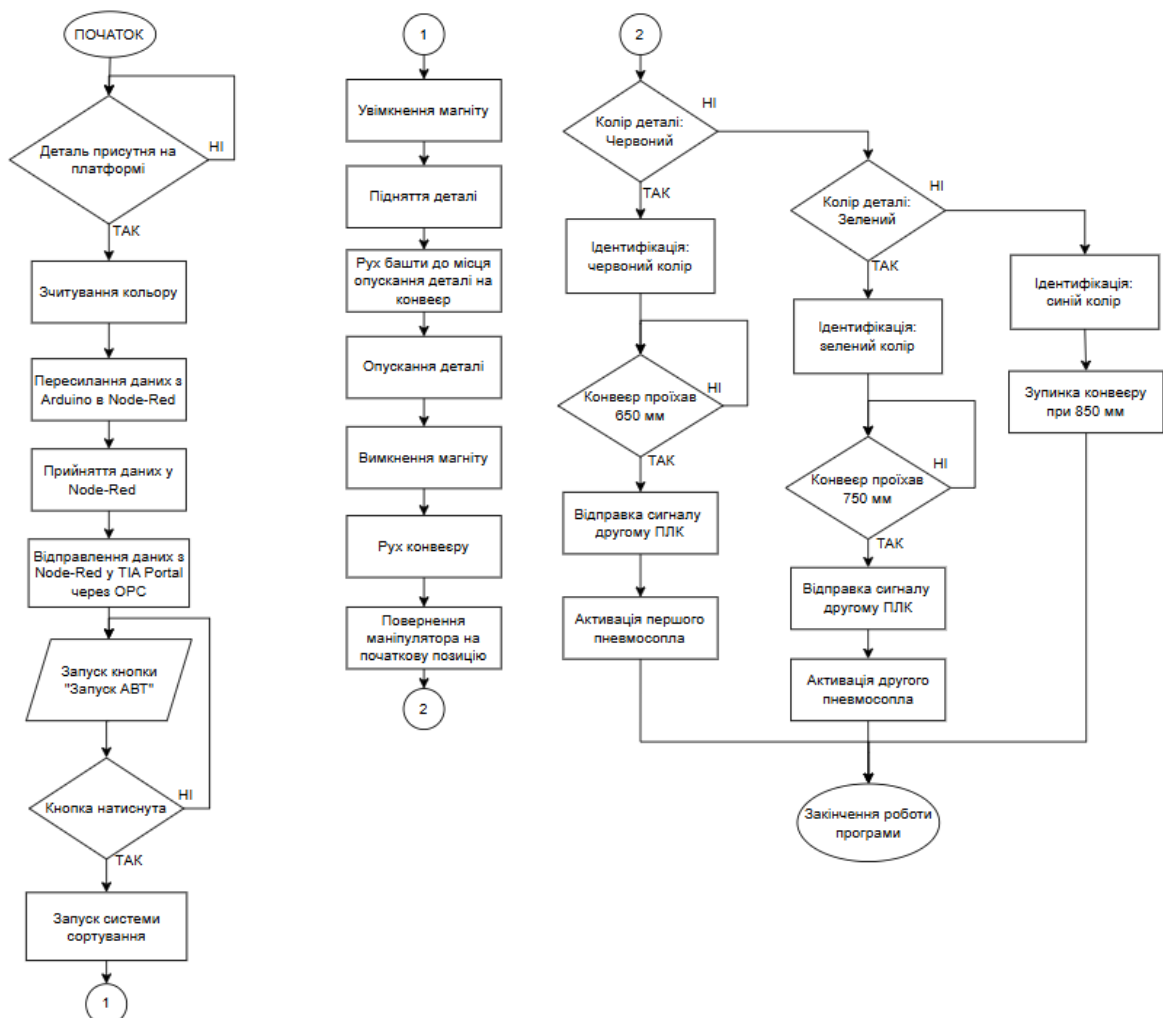


Рисунок 3.4 – Блок-схема алгоритму роботи програми

Окрім цього, було реалізовано ручний режим роботи системи. Ручний режим роботи сортувальної системи забезпечується інтегрованою пневматичною

частиною стенда, яка дозволяє оператору виконувати сортування деталей у разі виходу автоматичного режиму з ладу. Управління здійснюється за допомогою двох кнопок, кожна з яких відповідає за керування одним із двох пневмосопел (рис. 3.5).

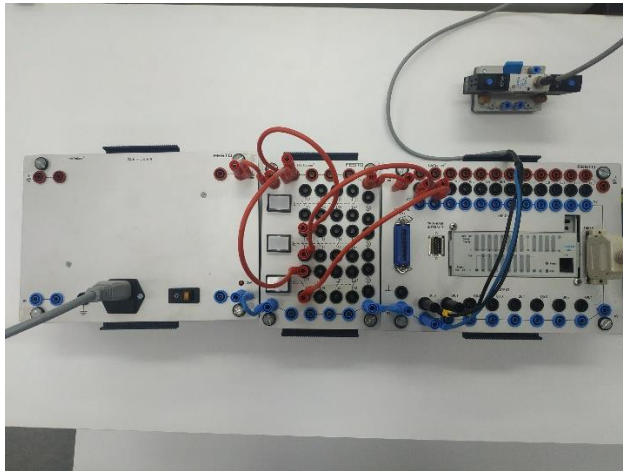


Рисунок 3.5 – Ручне управління соплами

Активуючи одну з кнопок, сигнал відправляється на розподілювач, який відкриває один з клапанів з стиснутим повітрям, яке йде на пневмосопло. Під час роботи в ручному режимі оператор має змогу самостійно активувати потрібне пневмосопло для перенаправлення деталей у відповідний контейнер залежно від їхнього кольору. Це забезпечує безперебійну роботу системи навіть за несправностей автоматичного управління, дозволяючи продовжити процес сортування з мінімальними затримками. Такий підхід підвищує надійність і функціональність системи, забезпечуючи резервний спосіб виконання операцій сортування.

3.3 Програмна реалізація проекту

3.3.1 Калібровка датчику та реалізація ідентифікації кольору

Першим етапом програмної реалізації автоматизованої системи сортування деталей за кольором є необхідність забезпечити правильне функціонування датчика кольору. Для цього було проведено його калібрування, яке є критично

важливим процесом для забезпечення точності зчитування кольорових характеристик об'єктів.

Калібрування датчика дозволяє адаптувати його до конкретних умов роботи, таких як освітлення, відстань до об'єкта та матеріал поверхні. На цьому етапі визначаються базові параметри кольорів, які будуть розпізнаватися системою, і встановлюються пороги для поділу кольорів. Процедура калібрування також включає тестування стабільності вимірів та налаштування чутливості датчика. [30]

Описаний далі ескіз слугує для калібрування датчика кольору TCS230. У процесі калібрування було визначено граничні значення частоти (мінімальне та максимальне) для кожного кольору. Це здійснюється шляхом фіксації показників датчика, коли він знаходиться в безпосередній близькості до об'єкта, а також на більшій відстані. Отримані значення дозволяють встановити частотний діапазон, який було використано в наступному етапі для визначення кольору об'єкта.

Програма починається з оголошення пінів (рис. 3.6), що використовуються для підключення датчика кольору TCS230. Спершу визначаються п'ять ключових контактів: S0, S1, S2, S3 і OUT, які відповідають за керування та отримання даних з датчика кольору. Вони підключені до GPIO4, GPIO5, GPIO6, GPIO7 і GPIO8.

```
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8

int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;
```

Рисунок 3.6 – Оголошення пінів

Додатково ініціалізуються змінні, які зберігають значення ширини імпульсів для червоного, зеленого та синього спектрів.

У секції налаштування (рис. 3.7) контакти S0-S3 конфігуруються як вихідні, оскільки вони використовуються для вибору режиму масштабування частоти та

визначення кольору, який необхідно зчитувати. Контакти S0 і S1 встановлюються в стан, що забезпечує масштабування частоти на 20%. Такий режим часто застосовується для інтеграції датчика кольору з Arduino, забезпечуючи оптимальну точність вимірювання. Вивід OUT датчика конфігурується як вхідний контакт Arduino для отримання імпульсних сигналів. На завершення, активується послідовний монітор для відображення зчитаних даних.

```
void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);

  pinMode(sensorOut, INPUT);

  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);

  Serial.begin(9600);
}
```

Рисунок 3.7 – Налаштування контактів

Послідовний монітор використовується для виведення значень частот, зчитаних датчиком кольору, що дозволяє в режимі реального часу відстежувати результати вимірювань і перевіряти коректність роботи системи. Також, він буде застосовуватись для зчитування значень кольору в Node-Red пізніше.

У основному блоці програми (рис. 3.8) виконується зчитування кольорів. Для цього датчик TCS230 використовує фільтри для виділення частот, що відповідають червоному, зеленому та синьому кольорам. Для активації фільтра червоного кольору контакти S2 і S3 встановлюються в стан **LOW**, після чого функція `pulseIn(sensorOut, LOW)` вимірює тривалість низького імпульсу на вихідному контакті датчика. Це значення частоти, яке відповідає інтенсивності

червоного кольору, записується у змінну `redFrequency` і відправляється на послідовний монітор для перегляду.

```
void loop() {
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);

  redFrequency = pulseIn(sensorOut, LOW);

  Serial.print("R = ");
  Serial.print(redFrequency);
  delay(1000);

  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);

  greenFrequency = pulseIn(sensorOut, LOW);

  Serial.print(" G = ");
  Serial.print(greenFrequency);
  delay(1000);

  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);

  blueFrequency = pulseIn(sensorOut, LOW);

  Serial.print(" B = ");
  Serial.println(blueFrequency);
  delay(1000);
}
```

Рисунок 3.8 – Основний код програми

Аналогічно виконується зчитування зеленого кольору, але для його активації контакти S2 і S3 встановлюються в стан **HIGH**, що дозволяє датчику виділити відповідну частоту. Отримане значення записується у змінну `greenFrequency` і також виводиться через послідовний монітор.

Для зчитування синього кольору контакти S2 встановлюються в стан **LOW**, а S3 – у стан **HIGH**. Частота, що відповідає синьому кольору, зчитується і записується у змінну `blueFrequency`, після чого відправляється до послідовного монітора.

Між кожною операцією зчитування передбачена затримка в 1 секунду (`delay(1000)`), що забезпечує стабільність системи, запобігаючи перевантаженню

датчика і Arduino, а також дає час для перегляду результатів на послідовному моніторі.

Серійний монітор виводитиме показники інтенсивності RGB. Датчик було розташовано на різних дистанціях (рис. 3.9) від цільового об'єкта, фіксуючи максимальні та мінімальні значення частот для червоного кольору (рис. 3.10).

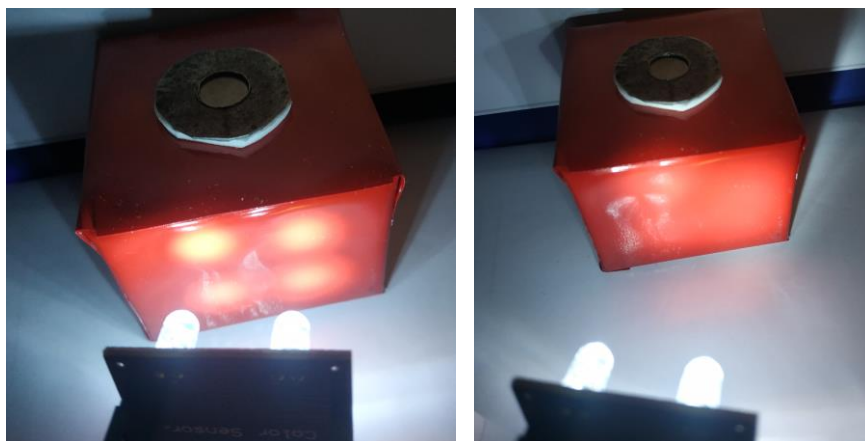


Рисунок 3.9 – Калібрування червоного кольору

```

Output  Serial Monitor  x
Message (Enter to send message to 'Arduin
--  --  --  --  --
R = 33 G = 32 B = 200
R = 116 G = 171 B = 141
R = 71 G = 141 B = 105
R = 205 G = 143 B = 125
R = 99 G = 308 B = 246
R = 192 G = 298 B = 240
R = 192 G = 298 B = 246
R = 199 G = 299 B = 246
R = 193 G = 292 B = 246
R = 200 G = 298 B = 246

```

Рисунок 3.10 – Вивід серійного монітору

Аналогічно, було повторно зроблено процедуру для синього та зеленого кольорів. У підсумку отримуємо шість різних частотних значень: два для червоного (мінімальне та максимальне), два для синього (мінімальне та максимальне) і два для зеленого (мінімальне та максимальне).

Наступний код був переписаний для відображення назви кольору в серійний

монітор, коли датчик ідентифікує колір деталі перед ним. У першому блоці (рис. 3.11) оголошуються змінні для зберігання значень частот, отриманих від датчика, а також значень, які відповідають інтенсивності кольору:

- **RED_freq**, **GREEN_freq**, **BLUE_freq** зберігають частоту сигналу для кожного кольору.
- **red_value**, **green_value**, **blue_value** містять значення, які відповідають інтенсивності кольору після обробки частоти.
- **red_avg**, **green_avg**, **blue_avg** використовуються для збереження середніх значень кольорів після обчислень на основі буферів.

```
int RED_freq = 0;
int GREEN_freq = 0;
int BLUE_freq = 0;

int red_value = 0;
int green_value = 0;
int blue_value = 0;

int red_avg = 0;
int green_avg = 0;
int blue_avg = 0;
```

Рисунок 3.11 – Оголошення змінних

У цьому блоці (рис. 3.12) створюються буфери для зберігання останніх вимірювань кожного кольору. Це необхідно для того, щоб зменшити вплив шумів і отримати більш стабільні значення кольорів.

- **bufferSize** визначає кількість значень, що зберігаються в буфері (5 значень).
- Масиви **redBuffer**, **greenBuffer** та **blueBuffer** зберігають виміряні значення для кожного кольору.
- **bufferIndex** використовується для відслідковування поточної позиції в буфері, і він циклічно оновлюється після кожного нового вимірювання.

```
// Буфери для зберігання останніх вимірювань кольорів
const int bufferSize = 5;
int redBuffer[bufferSize] = {0};
int greenBuffer[bufferSize] = {0};
int blueBuffer[bufferSize] = {0};
int bufferIndex = 0;
```

Рисунок 3.12 – Буфер вимірювань кольорів

В основний блок коду було додано функцію `map()`, яка перетворює виміряну частоту в значення інтенсивності кольору в діапазоні від 0 до 255.

Наступний блок (рис. 3.13) відповідає за збереження вимірених значень в буфери для кожного кольору. В кінці кожного циклу значення кольору додається в масив, а індекс буфера оновлюється. Після оновлення буфера обчислюються середні значення для кожного кольору за допомогою функції `calcAverage()`.

```
// Зберігаємо дані в буфери
redBuffer[bufferIndex] = red_value;
greenBuffer[bufferIndex] = green_value;
blueBuffer[bufferIndex] = blue_value;

bufferIndex = (bufferIndex + 1) % bufferSize; // Оновлюємо індекс буфера

// Обчислюємо середні значення для кожного кольору
red_avg = calcAverage(redBuffer);
green_avg = calcAverage(greenBuffer);
blue_avg = calcAverage(blueBuffer);
```

Рисунок 3.13 – Збереження даних і оновлення індексу

Останнім блоком програми виступає блок (рис. 3.14), який визначає котрий колір переважає, порівнюючи значення для червоного, зеленого та синього кольорів. Якщо різниця між максимальним значенням і іншими кольорами дуже мала (менше ніж 30), то програма виводить "Unknown" (невідомий колір). Інакше, вона виводить найпоширеніший колір серед трьох.

```
// Визначення кольору на основі найбільшого значення
int max_value = max(red_value, max(green_value, blue_value));

if ((max_value - red_value < 30) && (max_value - green_value < 30) && (max_value - blue_value < 30)) {
    Serial.println("Unknown");
} else if (red_value > green_value && red_value > blue_value) {
    Serial.println("Red");
} else if (green_value > red_value && green_value > blue_value) {
    Serial.println("Green");
} else if (blue_value > red_value && blue_value > green_value) {
    Serial.println("Blue");
}
```

Рисунок 3.14 – Визначення та перевірка кольору деталі

Результат ідентифікації кольору деталі та вивід інформації в буфер зображено на рисунку 3.15.

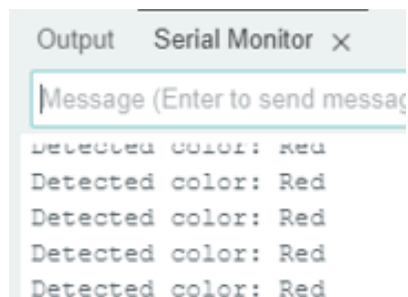


Рисунок 3.15 – Результат тестування датчика кольору

3.3.2 Налаштування зв'язку між Arduino і ПЛК

Наступним кроком у розвитку проекту буде реалізація передачі даних з плати Arduino до системи Node-RED через серіальний порт. Для отримання сигналу з Arduino в Node-RED було розроблено відповідний потік (flow). Після запуску Node-RED було здійснено установку необхідних бібліотек для роботи з послідовним портом.

Було зроблено програму з вузлами serial in, debug, switch, change та function. На рисунку 3.16 зображено налаштування кожного вузла програми.

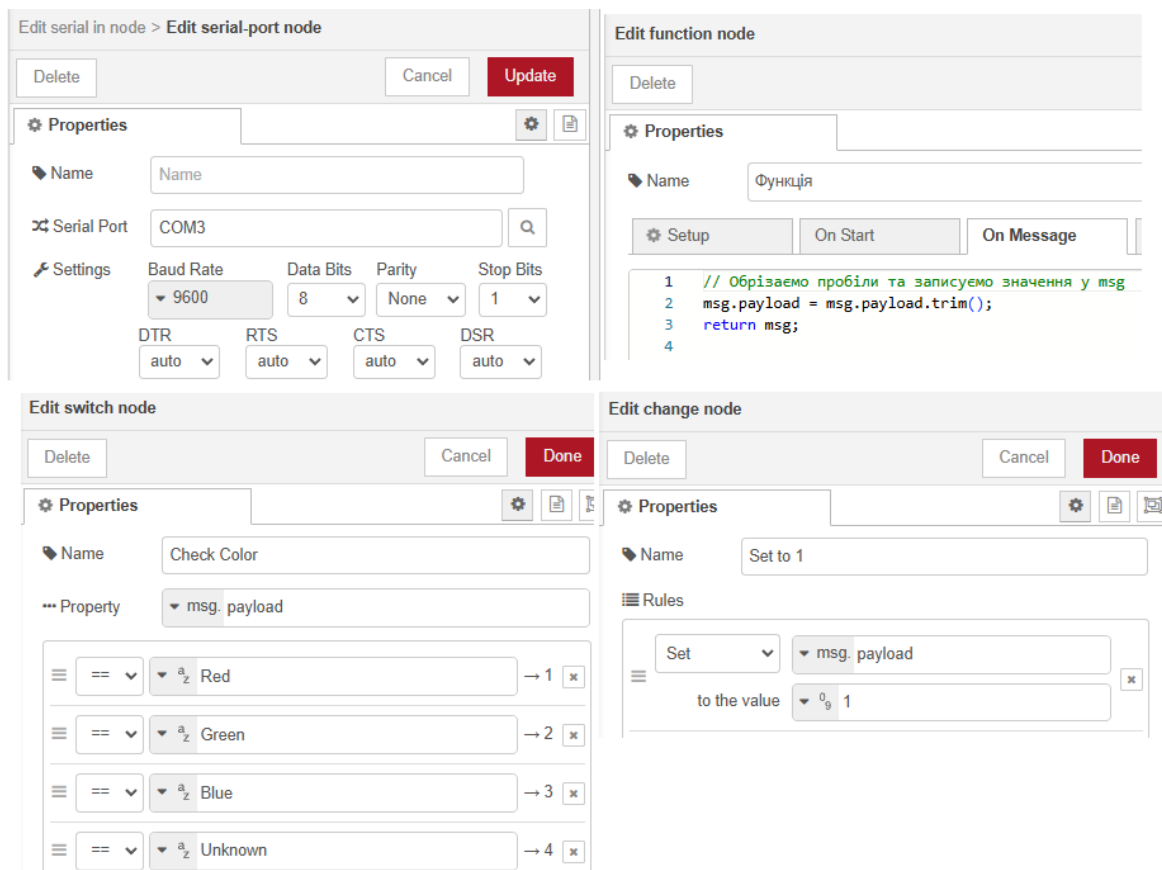


Рисунок 3.16 – Налаштування кожного вузла

Для вузла **serial in** було зроблено вибір порта Arduino та встановлення швидкості передачі даних на 9600. Вузол **function** використовується для отримання та перероблення строки з назвою кольору. Після цього отримане значення порівнюється з встановленими умовами в блоці **switch**, і в залежності від кольору, відправляється на необхідний вузол завдяки вузлу **change**.

Результатом є отримання кольору деталі в консоль Node-Red (рис. 3.17).

```
01.12.2024, 19:47:02 node: Перевірка червоного
msg.payload : string[3]
"Red"
01.12.2024, 19:47:05 node: Перевірка червоного
msg.payload : string[3]
"Red"
01.12.2024, 19:47:08 node: Перевірка червоного
msg.payload : string[3]
"Red"
```

Рисунок 3.17 – Результат

Для передачі кольору з Node-RED у тег проекту TIA Portal через OPC UA, спочатку було налаштовано проект лабораторного стенду в TIA Portal для роботи з OPC UA, створено необхідний тег для зберігання значення кольору. Після цього в Node-RED були використані вузли для підключення до OPC UA сервера та передачі значення в контролер на основі отриманих даних від датчика кольору.

На початку було увімкнено функцію запуску OPC UA сервера в налаштуваннях контролера. Проект був перезавантажений для збереження внесених змін.

OPC UA клієнти можуть здійснювати читання та запис тегів ПЛК, якщо для них дозволено використання в OPC UA комунікаціях. Для цього необхідно призначити теги ПЛК процесору S7-1200 як вузлам OPC UA через інтерфейс OPC UA сервера. Во вкладці «OPC UA communication» оберемо «Server interfaces», в якому можна створити новий інтерфейс (рис. 3.18). В новий інтерфейс серверу було перетягнуто тег для класифікації кольору деталі. Після цього було виконано компіляцію апаратного та програмного забезпечення станції ПЛК.

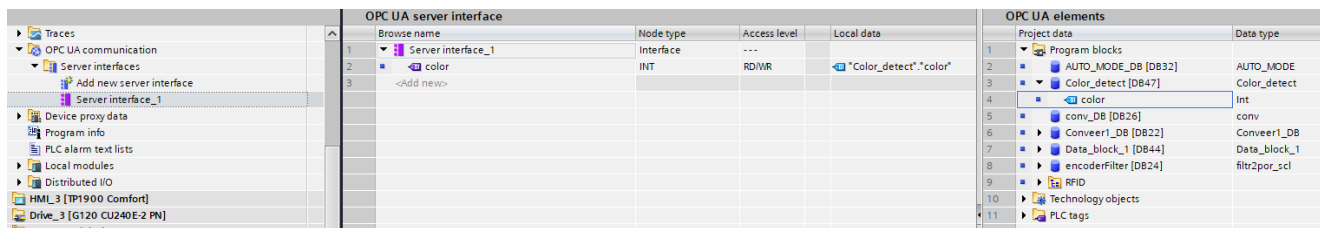


Рисунок 3.18 – Додавання тегу до нового інтерфейсу

Щоб прочитати дані кольору через OPC UA клієнт «UaExpert», було встановлено та запущено «UaExpert», після цього зроблено додавання та налаштування серверу (рис. 3.19). У наступному діалоговому вікні було введено URL-адресу та порт сервера OPC UA ЦП S7-1200 (opc.tcp://192.168.0.12:4840).

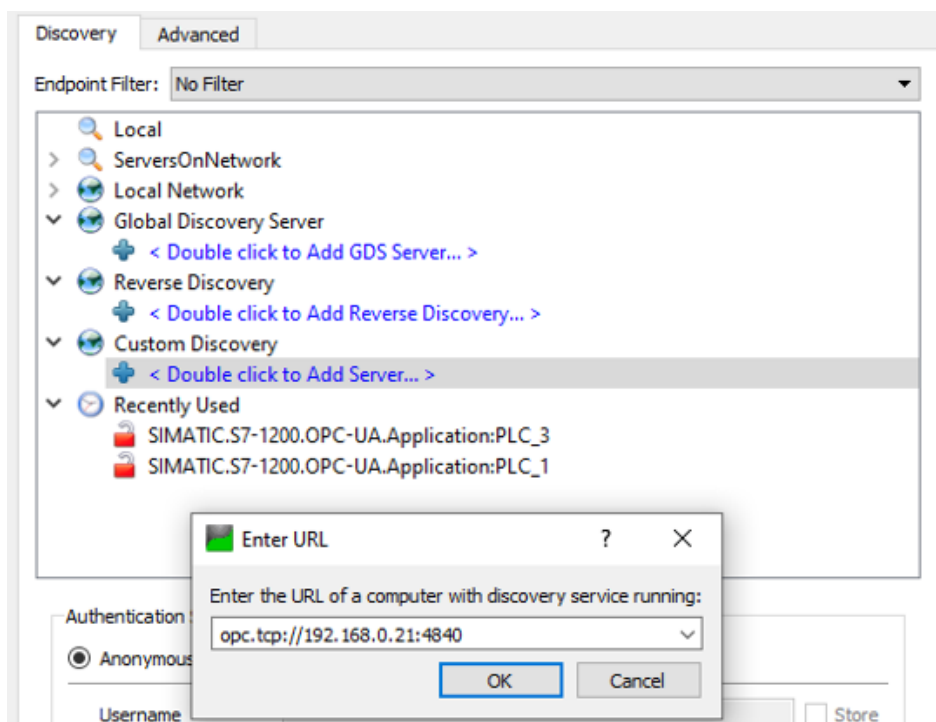


Рисунок 3.19 – Додавання серверу

Після обрання кінцевої точки серверу, до якої необхідно було встановити з'єднання, було виконано підключення до серверу OPC UA. Було створено дерево проекту, з папкою ServerInterfaces, в якій знаходиться необхідний тег кольору деталі. Завдяки UaExpert тепер було отримано мережевий адрес тегу для Node-Red (рис. 3.20). Даний тег буде використано для звертання Node-Red до контролера з можливістю записати необхідну інформацію про колір об'єкту в внутрішній тег PLC S7-1200.

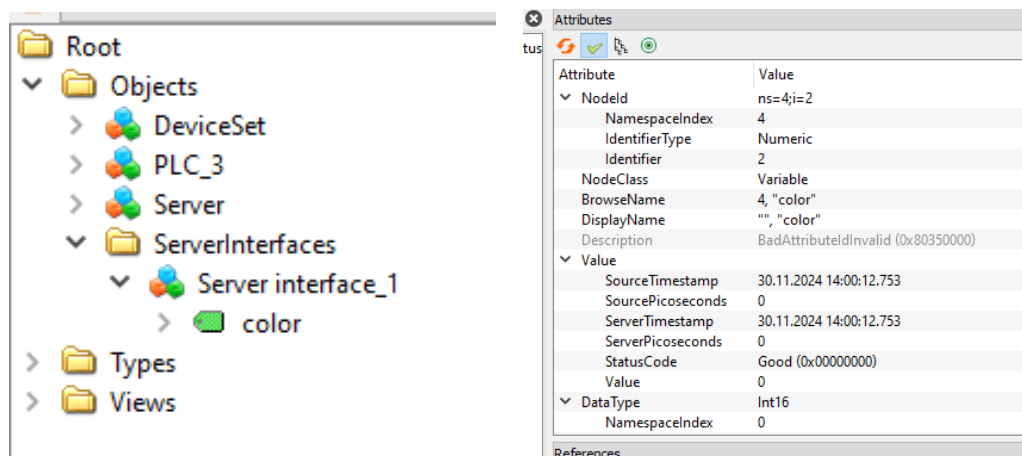


Рисунок 3.20 – Дерево проекту та властивості тегів

Для запису значення кольора деталі в ПЛК було використано вузли OPC Server в Node-Red. Налаштування блоків відправки повідомлення в ПЛК можна побачити на рисунку 3.21.

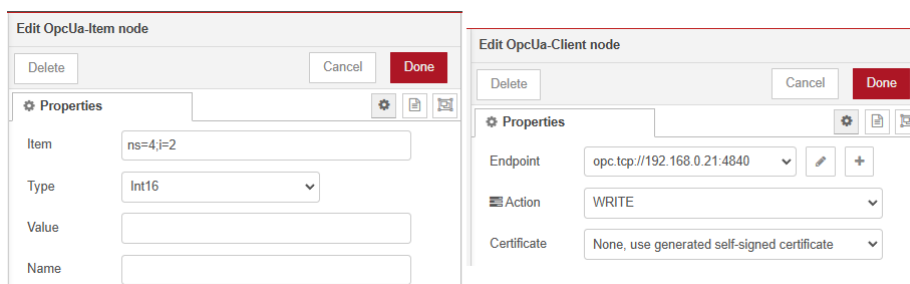


Рисунок 3.21 – Налаштування блоків

В вузлі OPC UA Item було записано адрес тегу та його тип. Налаштуванням блоку клієнта є вибір режиму – запис, та вибір кінцевої точки серверу. Весь ланцюг програми по передачі значення кольору деталі з Arduino в ПЛК зображено на рисунку 3.22.

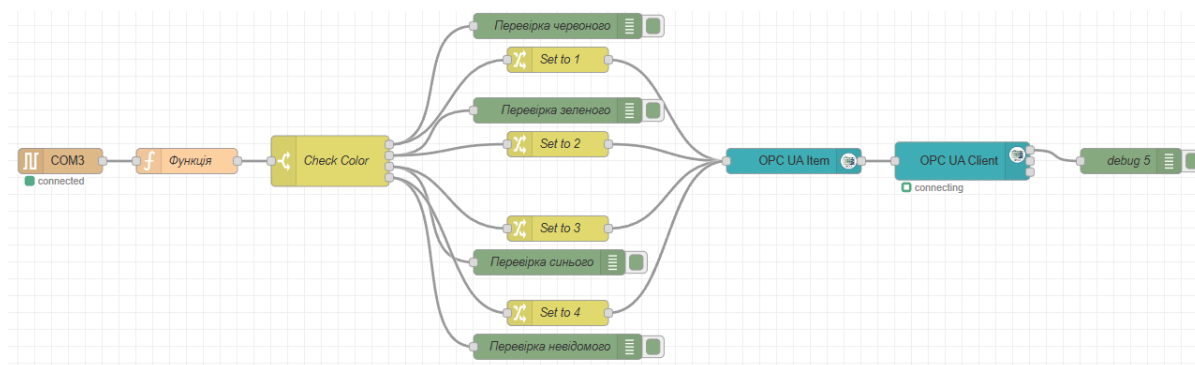


Рисунок 3.22 – Фрагмент програми

3.3.3 Модифікація програми стану системою сортування за кольором

Наступним кроком буде модифікація оригінальної програми лабораторного стану в TIA Portal. Було створено новий блок даних, який буде відповідати за збереження усіх змінних, необхідних для системи сортування (рис. 3.23).

Color_detect										
	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static									
2	color	int	0.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Отримана змінна кольору
3	color_promijna	int	2.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Змінна кольору для авто-моду
4	color_transfer	int	4.0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Змінна кольору для відправки в 2 ПЛК

Рисунок 3.23 – Новий блок даних

Для реалізації сортування в функції AUTO_MODE було додано код, який записує значення кольору, яке приходить з Node-Red, в проміжну змінну для подальшого використання її в програмі авто-режиму (рис. 3.24).

```
//Перевірка умов запуску автоматичного режиму
//Перевірка включення повзунка в автоматичний режим, кнопки пуск, кнопка стоп 0, перевірка помилок драйверів
IF "Switch_AUT/MAN" AND "Start_AUT" AND NOT "Stop_AUT" AND NOT "Conveyor".StatusBits.Error AND NOT "Axis_1".StatusBits.Error
AND NOT "Axis_2".StatusBits.Error AND NOT "Axis_3".StatusBits.Error
THEN
  "AUTO_mode_work" := 1;
  "Color_detect".color_promijna := "Color_detect".color; //копювання значення кольору в проміжну змінну
ELSE
  "AUTO_mode_work" := 0;
  "Start_AUT" := 0;
END_IF;
```

Рисунок 3.24 – Запис проміжної змінної

Після цього було реалізовано перевірку значення кольору деталі в програмі (рис. 3.25). Для цього було написано умови, які перевіряють колір деталі, яка в даний момент їде по конвеєру. Умовою є збігання значення тегу кольору з константою та значення положення конвеєру. При виконанні умови, наприклад, що зміна кольору деталі рівняється одиниці та положення конвеєру показує 650, то виконується активація проміжної змінної в TRUE для подальшої відправки в другий контролер і запуску одного з пневмосопел.

```
IF ("Color_detect".color_promijna = 1 AND "Data_block_1"."conv position" = 650) THEN
  "Color_detect".color_transfer := 1;
ELSIF ("Color_detect".color_promijna = 2 AND "Data_block_1"."conv position" = 750) THEN
  "Color_detect".color_transfer := 2;
ELSE
  "Color_detect".color_transfer := 0;
END_IF;
```

Рисунок 3.25 – Перевірка умови сортування деталі

Наступним кроком було створення блоку даних в другому ПЛК. Даний блок даних буде отримувати сигнал активації пневмосопла і записувати його до своїх тегів. Новий блок даних зображено на рисунку 3.26.

GET_COLOR										
	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	get_color	Int	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Отримане значення кольору деталі

Рисунок 3.26 – Новий блок даних другого ПЛК

Також, для нових блоків даних було вимкнено атрибут "Optimized Block Access" у налаштуваннях. У TIA Portal атрибут "Optimized Block Access" у налаштуваннях Data Block (DB) визначає спосіб доступу до змінних у цьому блоці. Це було зроблено для того, щоб другий контролер міг звертатися на пряму до адресів сигналів першого контролера, наприклад, DB1.DBW0, DB1.DBW2 тощо.

Так як обидва контролера знаходяться в одній підмережі (рис.3.27), було вирішено використати блок Get (S7-communication) для обміну даними між ними (рис. 3.28). У TIA Portal блоки Get та Put є частиною комунікаційної функціональності для обміну даними між різними ПЛК у мережі через протокол S7 Communication. Вони використовуються для передачі даних між ПЛК, коли потрібно забезпечити зв'язок між ними або з іншими пристроями, такими як НМІ (человек-машинний інтерфейс) чи SCADA системи. Ці блоки є частиною бібліотеки блоків комунікації S7. [31]

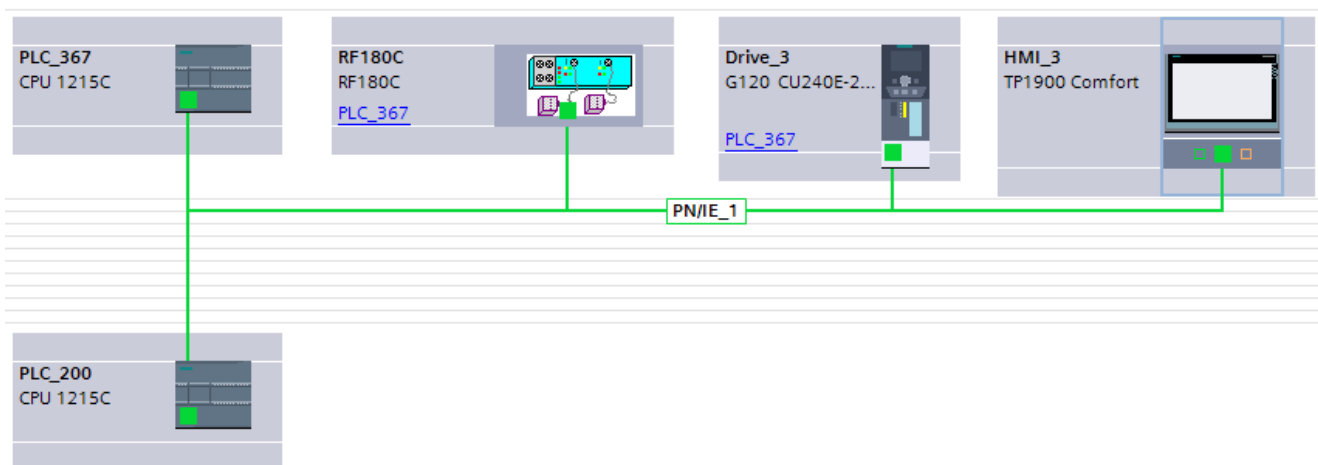


Рисунок 3.27 – Зображення мережевих зв'язків

Блок було перенесено в головний блок виконання коду програми. Даний блок активується завдяки змінній, яка формує періодичний сигнал з частотою 0.5 Гц (1 цикл кожні 2 секунди). Це означає, що сигнал "Clock_0.5Hz" буде чергувати значення TRUE (1) та FALSE (0) кожну секунду.

Даний блок звертається за адресою блоку даних першого контролера, який зберігає змінну кольору для обміну, і на виході записує значення отриманої змінної в тег блоку даних, який було створено вище по тексту.

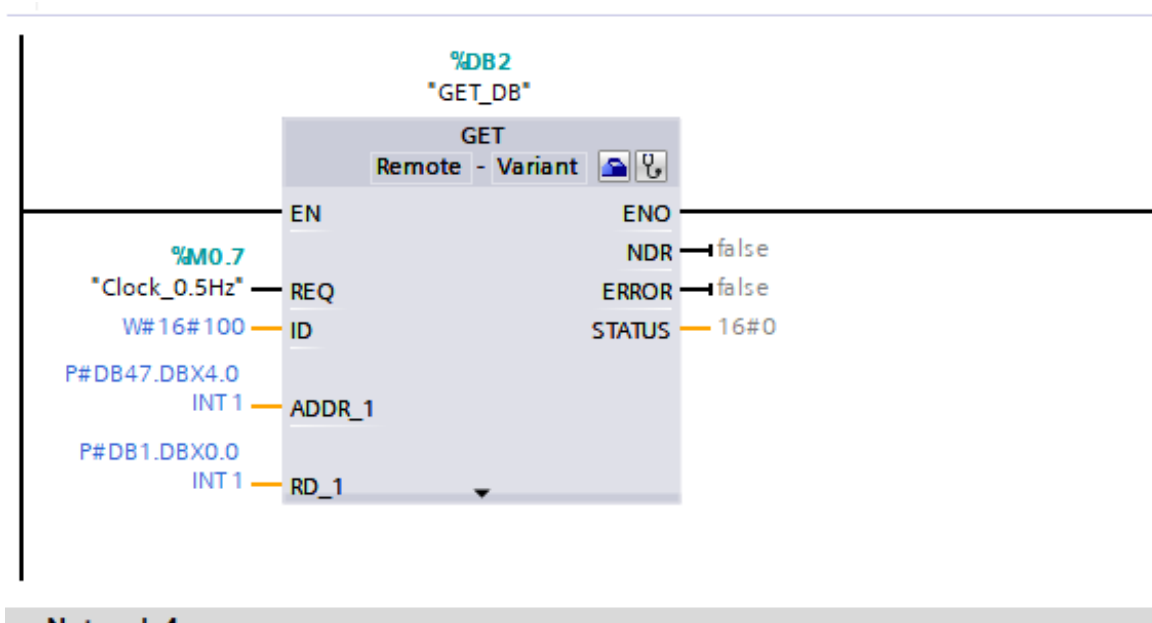


Рисунок 3.28 – Додавання блоку обміну між ПЛК

Останньою частиною коду (рис. 3.29), яка була модифікована, це функція активації пневмосопел в залежності від отриманого значення кольору від першого контролера.

При умові, що значення отриманої змінної, наприклад, дорівнює одиниці, контролер замкне нульовий дискретний вихід на один із електричних входів розподільника. Розподільник відкриє один з клапанів і кольорова деталь віддується повітрям під високим тиском із пневмосопла на необхідний конвеєр або в контейнер.

При умові, що значення, отримане другим контролером, дорівнює 3, то контролер не буде замикати свої дискретні виходи і кольорова деталь, наприклад зелена, проїде до кінця стрічки і наступний маніпулятор або контейнер прийме її.


```
IF "GET_COLOR".get_color = 1 THEN  
    "out0" := TRUE;  
ELSIF "GET_COLOR".get_color = 2 THEN  
    "out1" := TRUE;  
ELSE  
    "out0" := FALSE;  
    "out1" := FALSE;  
END_IF;
```

Рисунок 3.29 – Умова активації пневмосопел

Висновки до розділу:

Було представлено процес програмно-технічної реалізації системи керування процесом сортування, що включає проектування схем автоматизації, розробку алгоритмів роботи, а також реалізацію програмного забезпечення для окремих етапів функціонування системи.

Проектування схем автоматизації системи сортування передбачало створення структури, яка визначає взаємодію між основними компонентами системи, зокрема датчиками, пристроями для керування пневмосоплами та іншими елементами автоматизації. Це забезпечило необхідну точність та надійність роботи всіх підсистем, що беруть участь у процесі сортування.

Розроблений алгоритм роботи системи сортування на основі кольору передбачав ефективну реалізацію етапів виявлення кольору об'єкта, обробки цієї інформації та прийняття рішень щодо переміщення об'єкта за допомогою активації пневмосоплів.

Програмна реалізація проекту включала кілька етапів. Першим етапом була калібровка та написання коду для датчика, що забезпечило точність розпізнавання кольору та усунення помилок у роботі системи. Далі було налаштовано зв'язок між мікроконтролером Arduino та програмованим логічним контролером (ПЛК), що дозволило забезпечити двосторонню комунікацію між різними компонентами системи та інтеграцію їх роботи в єдине ціле.

Завершальним етапом стало вдосконалення програмного забезпечення стенду, що передбачало додавання системи сортування в існуючу систему.

ВИСНОВКИ

В ході виконання дипломної роботи було здійснено детальний аналіз та розробку автоматизованої системи сортування деталей за кольором з використанням пневмоавтоматики, що має на меті підвищення ефективності виробничих процесів. Система базується на детекції кольору деталей, після чого на основі отриманих даних пневмосопла автоматично викидають деталі в задані позиції на конвеєрному лінії.

Було проведено комплексний аналіз технологічного процесу сортування за кольоровими характеристиками та огляд існуючих рішень в області інтегрованих автоматизованих систем сортування (АСС). Початкова частина аналізу зосереджена на особливостях технологічного процесу сортування, де розглянуто ключові етапи та завдання, які виникають при обробці об'єктів, що потребують кольорового сортування. Оцінено вплив різних факторів на ефективність процесу, таких як точність розпізнавання кольорів, швидкість обробки і необхідність інтеграції з іншими елементами виробничої лінії.

У ході дослідження також була виконана класифікація існуючих систем сортування об'єктів за кольоровими характеристиками, що дозволило виявити основні типи технологій, застосовуваних в різних галузях. Особливу увагу приділено методам сортування за допомогою візуальних сенсорів, а також механізмам, що використовуються для переміщення об'єктів на конвеєрних лініях. Аналіз можливостей застосування технологій Industry 4.0 для інтелектуалізації процесів сортування деталей показав значний потенціал автоматизації та інтеграції систем управління в рамках єдиної цифрової екосистеми.

Огляд сучасних рішень в сфері автоматизованого сортування об'єктів за кольором виявив, що технології постійно вдосконалюються, зокрема в частині підвищення чутливості та точності датчиків кольору, а також вдосконалення програмного забезпечення для ефективного управління системами сортування.

Визначено, що впровадження системи сортування в роботу стану «маніпулятор-конвеєр» є кроком для підвищення її ефективності, зокрема, через

використання більш інтелектуальних алгоритмів та технологій, які дозволяють адаптувати систему до різноманітних умов роботи та вимог виробництва.

Алгоритм, який був розроблений, забезпечує точну ідентифікацію кольору деталі на конвеєрі та ефективне керування віддувкою пневмосоплом, що дозволяє розміщувати деталі відповідно до їхнього кольору в потрібному місці на конвеєрі.

Особливу увагу було приділено розробці та налаштуванню програмного забезпечення для управління системою. У процесі роботи з Arduino було проведено калібрування датчиків кольору, що дозволяє покращити точність ідентифікації кольорів. Важливим аспектом було налаштування взаємодії між платою Arduino та програмованим логічним контролером (ПЛК) для забезпечення безперебійного обміну даними та точного виконання алгоритмів сортування.

Система управління пневмосоплом була розроблена так, щоб враховувати поточну швидкість руху конвеєра та в залежності від кольору, виконувати автоматизовану віддувку деталей.

Також були проведені тести системи в реальних умовах лабораторного стенду. Результати тестувань показали високу ефективність системи сортування, зокрема здатність точно розпізнавати кольори і швидко сортувати деталі без помилок, а також надійність пневмосопла при виконанні завдань з віддувки деталей.

Подальшим напрямком розвитку цієї системи є вдосконалення точності розпізнавання кольорів шляхом інтеграції більш чутливого датчика, що дозволить працювати з більш складними умовами сортування (змінне освітлення, вібрація, тощо) та покращити продуктивність. Також можна працювати над вдосконаленням алгоритмів для автоматизації налаштування системи під різні виробничі умови, підвищення швидкості роботи маніпулятора і досягнення повної беззупинної роботи конвеєра, що дозволить значно збільшити швидкість сортування та знизити витрати на обслуговування системи.

Завершенням цієї роботи стало розроблення нових методичних матеріалів для проведення лабораторних занять із застосуванням додаткового обладнання на стенді аудиторії Siemens №367.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Конвеєри: види, класифікація та сфера застосування: URL: <https://ssk.ua/ua/blog/konvejery-vidy-klassifikaciya-i-sfera-primeneniya-482> (Дата звернення 07.09.24)
2. Контроль і сортування деталей URL: <https://budtehnika.pp.ua/208-kontrol-sortuvannya-detaley.html> (Дата звернення 8.09.24)
3. Оптичне та електронне сортування URL: <http://uk.techik-colorsorter.com/news/what-does-a-color-sorter-do/> (Дата звернення 10.09.24)
4. Вибір типу системи сортування в умовах певного асортименту продукції URL: <https://docs.broadcom.com/doc/18421> (Дата звернення 11.09.24)
5. Розумні сортувальні лінії (сортувальники) URL: <https://sbrobotics.ua/uslugi/sortuvalni-sistemi/> (Дата звернення 12.09.24)
6. Особливості застосування колірної корекції зображень URL: <https://ela.kpi.ua/server/api/core/bitstreams/6fad867e-e970-4fbc-95ba0d4b4d4b2163/content> (Дата звернення 13.09.24)
7. Розроблення технічного засобу автоматизації розподілення потоку деталей на конвеєрної лінії URL: <https://openarchive.nure.ua/entities/publication/57c84e7c-4110-4dd2-be52-037896a92edb> (Дата звернення 14.09.24)
8. Система автоматичного керування приводом робота-маніпулятора URL: <https://er.nau.edu.ua/bitstream/NAU/53262/1/%D0%A4%D0%90%D0%95%D0%> (Дата звернення 15.09.24)
9. Роботизовані маніпулятори особливого призначення URL: https://www.researchgate.net/publication/343340241_ROBOTIZOVANI_MANIPULATORI_OSOBLIVOGO_PRIZNACENNA (Дата звернення 16.09.24)
10. Інструментарій впровадження технологій індустрії 4.0 вітчизняними підприємствами URL: <https://economyandsociety.in.ua/index.php/journal/article/view/2312> (Дата звернення 17.09.24)

11. Технології сучасних кібер- фізичних систем URL: <https://files.znu.edu.ua/files/Bibliobooks/Inshi73/0053785.pdf> (Дата звернення 18.09.24)
12. Використання технологій Інтернету речей (ІоТ) URL: <https://cargofy.ua/uk/blog/vikoristannya-tehnologii-internetu-rechei-iot-dlya-optimizaciji-logistichnih-procesiv> (Дата звернення 19.09.24)
13. Основи роботи з Node-RED URL: <http://edu.asu.in.ua/mod/book/tool/print/index.php?id=123> (Дата звернення 20.09.24)
14. Технічний зір для системи керування сортувального роботу URL: <https://earchive.tpu.ru/bitstream/11683/47752/1/TPU541998.pdf> (Дата звернення: 22.09.2024)
15. Design of a new sorting colors system based on PLC, TIA portal, and factory I/O programs URL: https://www.researchgate.net/publication/378862876_Design_of_a_new_sorting_colors_system_based_on_PLC_TIA_portal_and_factory_IO_programs (Дата звернення: 23.09.2024)
16. Система управління промисловими роботами лабораторного стенду сортування технологічних об'єктів за кольором URL: <https://conf.ztu.edu.ua/wp-content/uploads/2018/05/125.pdf> (Дата звернення: 25.09.2024)
17. A pneumatic conveyor robot for color detection and sorting URL: <https://www.sciencedirect.com/science/article/pii/S2667241322000040> (Дата звернення: 26.09.2024)
18. Siemens 6EP1332-1SH71 Datasheet: URL: <https://docs.rs-online.com/b91a/0900766b8117fb77.pdf> (Дата звернення 27.09.24)
19. Siemens 6ES7215-1AG40-0XB0 Datasheet: URL: <https://media.automation24.com/datasheet/en/6ES72151AG400XB0.pdf> (Дата звернення 28.09.24)

20. Цвіркун Л. І., Грулер Г. Робототехніка та мехатроніка: навч. посіб. /Нац. гірн. ун-т. 3-тє вид., перероб.. і доповн./ Д.: НГУ, 2017. 224 с.
21. Modeling and Control of 2-DOF Robot Arm: URL: <https://www.ijeert.org/papers/v6-i11/3.pdf> (Дата звернення 28.09.24)
22. UNO R3 [Atmega 328P-AU+CH340G] URL: <https://www.makershop.de/download/uno-r3-manual.pdf> (Дата звернення 02.10.24)
23. TCS230 datasheet, converter equivalent URL: <https://s3-sa-east-1.amazonaws.com/robocore-lojavirtual/889/TCS230%20Datasheet.pdf> (Дата звернення 02.10.24)
24. Didactic InfoPortal URL: <https://ip.festo-didactic.com/Infoportal/EN/index.html> (Дата звернення 05.10.24)
25. Understanding Colour Sensors: Working Principle and Applications URL: <https://www.bestech.com.au/blogs/understanding-colour-sensors-working-principle-and-applications/> (Дата звернення 07.10.24)
26. Рівняння Бернуллі URL: <https://www.kspu.edu/FileDownload.ashx/%D0%94%D0%B8%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D1%96%D0%B0%D0%BB%D1%8C%D0%BD%D1%96%20%D1%80%D1%96%D0%B2%D0%BD%D1%8F%D0%BD%D0%BD%D1%8F%20211.pdf?id=17560201-ab80-452a-a422-0339b4e5895d> (Дата звернення 09.10.24)
27. Maxum OPC ServerUser Manual URL: https://cache.industry.siemens.com/dl/files/888/109763888/att_1007698/v1/OPCServerUserManual-V3.pdf (Дата звернення 11.10.24)
28. AutoCAD 2025 Instructor A Student Guide for In-Depth Coverage of AutoCAD's Commands and Features URL: <https://www.routledge.com/AutoCAD-2025-Instructor-A-Student-Guide-for-In-Depth-Coverage-of-AutoCADs-Commands-and-Features/Leach-Lockhart/p/book/9781630576493> (Дата звернення 12.10.24)

29. Кабель ДБ25 URL:
<https://www.digchip.com/datasheets/parts/datasheet/2136/DB25-PD.php> (Дата звернення 14.10.24)
30. TCS230/TCS3200 Color Sensor with Arduino URL:
<https://microcontrollerslab.com/tcs230-tcs3200-colour-sensor-arduino-tutorial/> (Дата звернення 15.10.24)
31. S7 communication between SIMATIC S7-1200 URL:
https://cache.industry.siemens.com/dl/files/115/82212115/att_1039292/v2/82212115_S7_communication_S7-1500_S7-1200_en.pdf (Дата звернення 21.10.24)
32. Маринич І. А., Тронь В. В. Методичні рекомендації до виконання кваліфікаційної роботи магістра для студентів спеціальності 151 “Автоматизація та комп’ютерно-інтегровані технології”. Кривий Ріг : Видавничий центр КНУ, 2022. 50с.
33. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура і правила оформлення. Київ, ДП «УкрННЦ», 2015. 26с. (Інформація та документація).
34. ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання Київ, ДП «УкрННЦ», 2016. 16 с. (Інформація та документація).
35. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила. Київ, ДП «УкрННЦ», 2013. 23 с. (Інформація та документація)
36. ДСТУ 3651.0-97 Метрологія. Одиниці фізичних величин. Основні одиниці фізичних величин Міжнародної системи одиниць. Основні положення, назви та позначення Київ, Держстандарт України, 1998. 27 с. (Інформація та документація).

Кодування програми калібрування датчика кольору

```
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8
int redMin = 25;
int redMax = 445;
int greenMin = 36;
int greenMax = 433;
int blueMin = 32;
int blueMax = 353;
int redPW = 0;
int greenPW = 0;
int bluePW = 0;
int redValue;
int greenValue;
int blueValue;
void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);
  Serial.begin(9600);
}
void loop() {
```



```
redPW = getRedPW();
redValue = map(redPW, redMin,redMax,255,0);
delay(200);
greenPW = getGreenPW();
greenValue = map(greenPW, greenMin,greenMax,255,0);
delay(200);
bluePW = getBluePW();
blueValue = map(bluePW, blueMin,blueMax,255,0);
delay(200);
Serial.print("Red = ");
Serial.print(redValue);
Serial.print(" - Green = ");
Serial.print(greenValue);
Serial.print(" - Blue = ");
Serial.println(blueValue);
if(redValue > greenValue && redValue > blueValue){
  Serial.println(" - RED detected!");
}
if(greenValue > redValue && greenValue > blueValue){
  Serial.println(" - GREEN detected!");
}
if(blueValue > redValue && blueValue > greenValue){
  Serial.println(" - BLUE detected!");
}
}
int getRedPW() {
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  int PW;
```

```
PW = pulseIn(sensorOut, LOW);  
return PW;  
}  
int getGreenPW() {  
    digitalWrite(S2,HIGH);  
    digitalWrite(S3,HIGH);  
    int PW;  
    PW = pulseIn(sensorOut, LOW);  
    return PW;  
}  
int getBluePW() {  
    digitalWrite(S2,LOW);  
    digitalWrite(S3,HIGH);  
    int PW;  
    PW = pulseIn(sensorOut, LOW);  
    return PW;  
}
```

Кодування програми ідентифікації датчика кольору

```
#define S0 4  
#define S1 5  
#define S2 6  
#define S3 7  
#define sensorOut 8  
int redMin = 25;  
int redMax = 445;  
int greenMin = 36;  
int greenMax = 433;  
int blueMin = 32;  
int blueMax = 353;
```

```
int redPW = 0;
int greenPW = 0;
int bluePW = 0;
int redValue;
int greenValue;
int blueValue;
void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);
  Serial.begin(9600);
}
void loop() {
  redPW = getRedPW();
  redValue = map(redPW, redMin,redMax,255,0);
  delay(200);
  greenPW = getGreenPW();
  greenValue = map(greenPW, greenMin,greenMax,255,0);
  delay(200);
  bluePW = getBluePW();
  blueValue = map(bluePW, blueMin,blueMax,255,0);
  delay(200);
  if(redValue > greenValue && redValue > blueValue){
    Serial.println("Red");
  }
}
```

```
if(greenValue > redValue && greenValue > blueValue){
  Serial.println("Green");
}
if(blueValue > redValue && blueValue > greenValue){
  Serial.println("Blue");
}
}
int getRedPW() {
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
  int PW;
  PW = pulseIn(sensorOut, LOW);
  return PW;
}
int getGreenPW() {
  digitalWrite(S2,HIGH);
  digitalWrite(S3,HIGH);
  int PW;
  PW = pulseIn(sensorOut, LOW);
  return PW;
}
int getBluePW() {
  digitalWrite(S2,LOW);
  digitalWrite(S3,HIGH);
  int PW;
  PW = pulseIn(sensorOut, LOW);
  return PW;
}
```

Кодування програми Node-RED

```

IF "Switch_AUT/MAN" AND "Start_AUT" AND NOT "Stop_AUT" AND NOT
"Conveyor".StatusBits.Error AND NOT "Axis_1".StatusBits.Error
  AND NOT "Axis_2".StatusBits.Error AND NOT "Axis_3".StatusBits.Error
THEN
  "AUTO_mode_work" := 1;
ELSE
  "AUTO_mode_work" := 0;
  "Start_AUT" := 0;
END_IF;
IF NOT "AUTO_mode_work" AND "Switch_AUT/MAN" THEN
  "Data_block_1"."stop D1" := 1;
  "Data_block_1"."stop D2" := 1;
  "Data_block_1"."stop D3" := 1;
  "Data_block_1"."conv stop" := 1;
END_IF;
IF "AUTO_mode_work" THEN
  IF NOT "STEP_0_END" THEN
    "STEP_0" := 1;
  ELSE
    "STEP_0" := 0;
  END_IF;
  IF "STEP_0" THEN
    "Data_block_1"."On/off D1" := 1;
    "Data_block_1"."On/off D2" := 1;
    "Data_block_1"."On/off D3" := 1;
    "Data_block_1"."stop D1" := 0;
    "Data_block_1"."stop D2" := 0;
    "Data_block_1"."stop D3" := 0;
    "Data_block_1"."home D1" := 1;
    "Data_block_1"."home D2" := 1;
  
```

```

"Data_block_1"."home D3" := 1;
"Color_detect".color_platform := "Color_detect".color;
#Start_Timer_0 := 1;
"IEC_Tim_OFF_0_DB".TON(IN := #Start_Timer_0,
                        PT := T#1s,
                        ET => #ET_TIME_ON_0,
                        Q => #Timer_0_END);
IF #Timer_0_END THEN
  "Data_block_1"."home D1" := 0;
  "Data_block_1"."home D2" := 0;
  "Data_block_1"."home D3" := 0;
  "STEP_0_END" := 1;
  "STEP_1" := 1;
END_IF;
END_IF;
IF "STEP_1" THEN
  "Data_block_1"."angle D3" := -45;
  "Data_block_1"."move D3" := 1;
  #Start_Timer_1 := 1;

  "IEC_Tim_OFF_1_DB".TON(IN := #Start_Timer_1,
                        PT := T#12s,
                        ET => #ET_TIME_ON_1,
                        Q => #Timer_1_END);
IF #Timer_1_END THEN
  "Data_block_1"."move D3" := 0;
  "magnit" := 1;
  #Start_Timer_2 := 1;
END_IF;

```

```
"IEC_Tim_OFF_2_DB".TON(IN := #Start_Timer_2,  
    PT := T#1s,  
    ET => #ET_TIME_ON_2,  
    Q => #Timer_2_END);  
IF #Timer_2_END THEN  
    "Data_block_1"."angle D3" := 40;  
    "Data_block_1"."move D3" := 1;  
    "Color_detect".color_promijna := "Color_detect".color_platform;  
    "Color_detect".color_platform := 0;  
    #Start_Timer_3 := 1;  
END_IF;  
"IEC_Tim_OFF_3_DB".TON(IN := #Start_Timer_3,  
    PT := T#10s,  
    ET => #ET_TIME_ON_3,  
    Q => #Timer_3_END);  
IF #Timer_3_END THEN  
    "Data_block_1"."move D3" := 0;  
    "STEP_1" := 0;  
    "STEP_2" := 1;  
END_IF;  
END_IF;  
IF "STEP_2" THEN  
    "Data_block_1"."angle D2" := -90;  
    "Data_block_1"."move D2" := 1;  
    #Start_Timer_4 := 1;  
    "IEC_Tim_OFF_4_DB".TON(IN := #Start_Timer_4,  
        PT := T#1s,  
        ET => #ET_TIME_ON_4,  
        Q => #Timer_4_END);
```

```
"IEC_Tim_OFF_5_DB".TON(IN := #Start_Timer_5,  
                        PT := T#1s,  
                        ET => #ET_TIME_ON_5,  
                        Q => #Timer_5_END);  
IF #Timer_4_END THEN  
    "Data_block_1"."move D2" := 0;  
    "STEP_2" := 0;  
    "STEP_3" := 1;  
END_IF;  
END_IF;  
IF "STEP_3" THEN  
    "Data_block_1"."angle D1" := 90;  
    "Data_block_1"."move D1" := 1;  
    #Start_Timer_5 := 1;  
    "IEC_Tim_OFF_5_DB".TON(IN := #Start_Timer_5,  
                            PT := T#1s,  
                            ET => #ET_TIME_ON_5,  
                            Q => #Timer_5_END);  
    IF #Timer_5_END THEN  
        "Data_block_1"."move D1" := 0;  
        "STEP_3" := 0;  
        "STEP_4" := 1;  
    END_IF;  
END_IF;  
IF "STEP_4" THEN  
    "Data_block_1"."angle D2" := 0;  
    "Data_block_1"."move D2" := 1;  
    #Start_Timer_6 := 1;
```



```
"IEC_Tim_OFF_6_DB".TON(IN := #Start_Timer_6,  
    PT := T#1s,  
    ET => #ET_TIME_ON_6,  
    Q => #Timer_6_END);  
IF #Timer_6_END THEN  
    "Data_block_1"."move D2" := 0;  
    "STEP_4" := 0;  
    "STEP_5" := 1;  
END_IF;  
END_IF;  
IF "STEP_5" THEN  
    "Data_block_1"."angle D3" := -15;  
    "Data_block_1"."move D3" := 1;  
    #Start_Timer_7 := 1;  
    "IEC_Tim_OFF_7_DB".TON(IN := #Start_Timer_7,  
        PT := T#15s,  
        ET => #ET_TIME_ON_7,  
        Q => #Timer_7_END);  
IF #Timer_7_END THEN  
    "Data_block_1"."move D3" := 0;  
    "magnit" := 0;  
    #Start_Timer_8 := 1;  
END_IF;  
"IEC_Tim_OFF_8_DB".TON(IN := #Start_Timer_8,  
    PT := T#1s,  
    ET => #ET_TIME_ON_8,  
    Q => #Timer_8_END);  
IF #Timer_8_END THEN  
    "Data_block_1"."angle D3" := 40;  
    "Data_block_1"."move D3" := 1;
```

```

    "Data_block_1"."conv ON/OFF" := 1;
    "Data_block_1"."conv stop" := 0;
    #Start_Timer_9 := 1;
END_IF;
"IEC_Tim_OFF_9_DB".TON(IN := #Start_Timer_9,
    PT := T#15s,
    ET => #ET_TIME_ON_9,
    Q => #Timer_9_END);
IF #Timer_9_END THEN
    "Data_block_1"."move D3" := 0;
    "STEP_5" := 0;
    "STEP_6" := 1;
END_IF;
END_IF;
IF "STEP_6" THEN
    "Data_block_1"."conv home" := 1;
    IF "RightSensor" THEN
        "Data_block_1"."conv position" := 820;
        "Data_block_1"."conv speed" := 50;
        "Data_block_1"."conv move" := 1;
        "Data_block_1"."angle D2" := 90;
        "Data_block_1"."move D2" := 1;
        "Start_Timer_10" := 1;
    END_IF;
    IF "Color_detect".color_promijna = 1 THEN
        "IEC_Tim_OFF_10_DB".TON(IN := "Start_Timer_10",
            PT := T#1s,
            ET => #ET_TIME_ON_10,
            Q => #Timer_10_END);
    END_IF;
END_IF;

```

```

IF #Timer_10_END AND "LeftSensor" THEN
    "Data_block_1"."move D2" := 0;
    "Data_block_1"."conv move" := 0;
    "Data_block_1"."conv ON/OFF" := 0;
    "Data_block_1"."conv home" := 0;
    "Data_block_1"."conv speed" := 0;
    "Start_Timer_10" := 0;
    "Data_block_1"."conv position" := 0;
    "STEP_6" := 0;
    "STEP_7" := 1;
END_IF;
ELSIF "Color_detect".color_promijna = 2 THEN
    "IEC_Tim_OFF_10_DB".TON(IN := "Start_Timer_10",
        PT := T#1s,
        ET => #ET_TIME_ON_10,
        Q => #Timer_10_END);
    IF #Timer_10_END AND "Data_block_1"."conv position" =
"Color_detect".position_pnevmo1 THEN
        "Color_detect".pnevmo1_status := 1;
        #Start_Timer_14 := 1;
        "IEC_Timer_0_DB".TON(IN := #Start_Timer_14,
            PT := T#1s,
            ET => #ET_TIME_ON_14,
            Q => #Timer_14_END);
        IF #Timer_14_END THEN
            "Start_Timer_10" := 0;
            #Start_Timer_14 := 0;
            "Color_detect".pnevmo1_status := 0;
            "Data_block_1"."move D2" := 0;
            "Data_block_1"."conv move" := 0;

```

```

"Data_block_1"."conv ON/OFF" := 0;
"Data_block_1"."conv home" := 0;
"Data_block_1"."conv speed" := 0;
"Data_block_1"."conv position" := 0;
"STEP_6" := 0;
"STEP_7" := 1;
END_IF;
END_IF;
ELSIF "Color_detect".color_promijna = 3 THEN
  "IEC_Tim_OFF_10_DB".TON(IN := "Start_Timer_10",
    PT := T#1s,
    ET => #ET_TIME_ON_10,
    Q => #Timer_10_END);
  IF #Timer_10_END AND "Data_block_1"."conv position" =
"Color_detect".position_pnevmo2 THEN
    "Color_detect".pnevmo2_status := 1;
    #Start_Timer_15 := 1;
    "IEC_Timer_0_DB".TON(IN := #Start_Timer_15,
      PT := T#1s,
      ET => #ET_TIME_ON_15,
      Q => #Timer_15_END);
    IF #Timer_15_END THEN
      "Start_Timer_10" := 0;
      #Start_Timer_15 := 0;
      "Color_detect".pnevmo2_status := 0;
      "Data_block_1"."move D2" := 0;
      "Data_block_1"."conv move" := 0;
      "Data_block_1"."conv ON/OFF" := 0;
      "Data_block_1"."conv home" := 0;
      "Data_block_1"."conv speed" := 0;

```

```

        "Data_block_1"."conv position" := 0;
        "STEP_6" := 0;
        "STEP_7" := 1;
    END_IF;
END_IF;
END_IF;
END_IF;
IF "STEP_7" THEN
    "Data_block_1"."angle D1" := 0;
    "Data_block_1"."move D1" := 1;
    #Start_Timer_11 := 1;
    "IEC_Tim_OFF_11_DB".TON(IN := #Start_Timer_11,
        PT := T#1s,
        ET => #ET_TIME_ON_11,
        Q => #Timer_11_END);
    IF #Timer_11_END THEN
        "Data_block_1"."move D1" := 0;
        "Data_block_1"."conv home" := 1;
        "STEP_7" := 0;
        "STEP_8" := 1;
    END_IF;
END_IF;
IF "STEP_8" THEN
    "Data_block_1"."angle D2" := 0;
    "Data_block_1"."move D2" := 1;
    #Start_Timer_12 := 1;
    "IEC_Tim_OFF_12_DB".TON(IN := #Start_Timer_12,
        PT := T#1s,
        ET => #ET_TIME_ON_12,
        Q => #Timer_12_END);

```

```

IF #Timer_12_END THEN
    "Data_block_1"."move D2" := 0;
    "Data_block_1"."angle D3" := 0;
    "Data_block_1"."move D3" := 1;
    #Start_Timer_13 := 1;
END_IF;
"IEC_Tim_OFF_13_DB".TON(IN := #Start_Timer_13,
    PT := T#15s,
    ET => #ET_TIME_ON_13,
    Q => #Timer_13_END);
IF #Timer_13_END THEN
    "Data_block_1"."move D3" := 0;
    "Data_block_1"."angle D1" := 0;
    "STEP_8" := 0;
    "STEP_0_END" := 0;
    "Data_block_1"."On/off D1" := 0;
    "Data_block_1"."On/off D2" := 0;
    "Data_block_1"."On/off D3" := 0;
    "Data_block_1"."move D1" := 0;
    "Data_block_1"."move D2" := 0;
    "Data_block_1"."move D3" := 0;
    "Data_block_1"."conv home" := 0;
    "Switch_AUT/MAN" := 0;
    RESET_TIMER("IEC_Tim_OFF_0_DB");
    RESET_TIMER("IEC_Tim_OFF_1_DB");
    RESET_TIMER("IEC_Tim_OFF_2_DB");
    RESET_TIMER("IEC_Tim_OFF_3_DB");
    RESET_TIMER("IEC_Tim_OFF_4_DB");
    RESET_TIMER("IEC_Tim_OFF_5_DB");

```

```
RESET_TIMER("IEC_Tim_OFF_6_DB");  
RESET_TIMER("IEC_Tim_OFF_7_DB");  
RESET_TIMER("IEC_Tim_OFF_8_DB");  
RESET_TIMER("IEC_Tim_OFF_9_DB");  
RESET_TIMER("IEC_Tim_OFF_10_DB");  
RESET_TIMER("IEC_Tim_OFF_11_DB");  
RESET_TIMER("IEC_Tim_OFF_12_DB");  
RESET_TIMER("IEC_Tim_OFF_13_DB");  
END_IF;  
END_IF;  
END_IF;
```

Проект методичних вказівок

Лабораторна робота №1

Тема: Дослідження датчика кольору TCS230.

Мета: Вивчення призначення, будови та принципу роботи датчика кольору TCS230

Теоретичні відомості

На рис. 1 наведено зображення розміщення датчика на стенді № 2. Модель датчика TCS230.

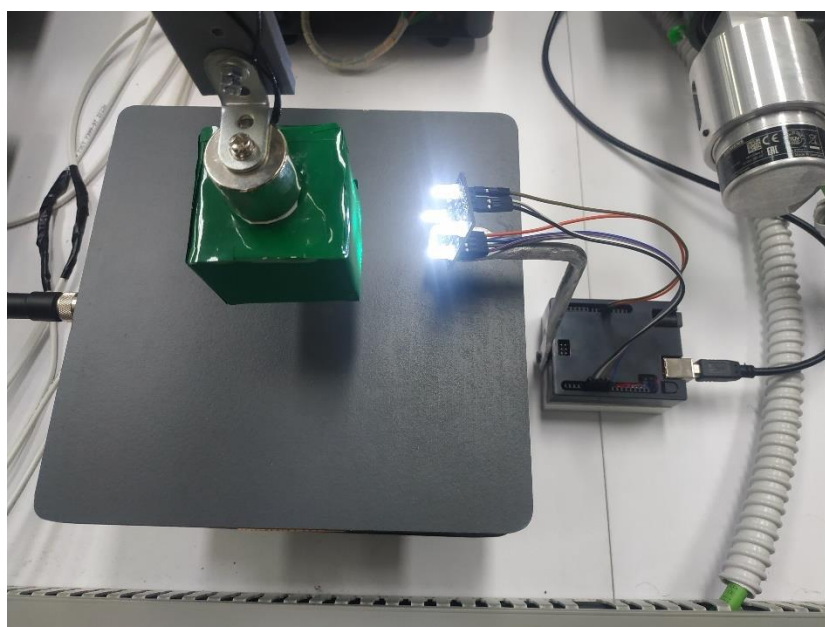


Рисунок 1 – Зображення датчика

Датчик кольору TCS230, показаний на рисунку 1, використовує сенсорний чіп TAOS TCS230 RGB для визначення кольору. Він також містить чотири білі світлодіоди, які освітлюють об'єкт перед ним. Датчик має масив фотодіодів із 4 різними фільтрами. Фотодіод - це просто напівпровідниковий пристрій, який перетворює світло в струм. Датчик має:

- 16 фотодіодів з червоним фільтром – чутливі до червоної довжини хвилі
- 16 фотодіодів із зеленим фільтром – чутливі до зеленої довжини хвилі
- 16 фотодіодів з синім фільтром – чутливі до синьої довжини хвилі

- 16 фотодіодів без фільтра

Вибірково вибираючи показники фотодіодного фільтра, ви можете визначити інтенсивність різних кольорів. Датчик має перетворювач струму в частоту, який перетворює показання фотодіодів у меандр із частотою, пропорційною інтенсивності світла вибраного кольору. Ця частота потім зчитується Arduino – це показано на рисунку 2.

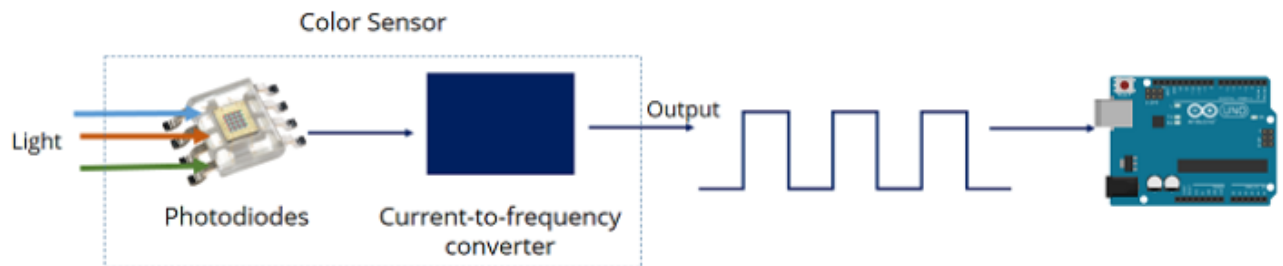


Рисунок 2 – Принцип роботи датчика

Для вибору кольору, який зчитує фотодіод, ви використовуєте контрольні контакти S2 і S3. Оскільки фотодіоди з'єднані паралельно, налаштування S2 і S3 LOW і HIGH у різних комбінаціях дозволяє вибрати різні фотодіоди. Таблиця 1 показує принцип вибору фільтра.

Таблиця 1 – Вибір фільтра

Photodiode type	S2	S3
Red	LOW	LOW
Blue	LOW	HIGH
No filter (clear)	HIGH	LOW
Green	HIGH	HIGH

Висновки S0 і S1 використовуються для масштабування вихідної частоти. Його можна масштабувати до таких попередньо встановлених значень: 100%, 20% або 2%. Масштабування вихідної частоти корисно для оптимізації показань датчиків для різних частотомірів або мікроконтролерів. Таблиця 2 показує принцип масштабування частоти.

Таблиця 2 – Масштабування частоти

Output frequency scaling	S0	S1
Power down	L	L
2%	L	H
20%	H	L
100%	H	H

Підключення датчика до плати Arduino зображено на схемі 3.

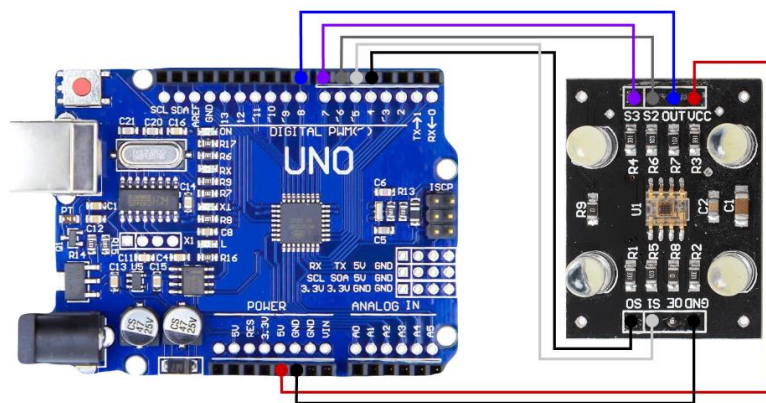


Рисунок 3 – Макетна схема системи аналізу модулю

Завдання

- Ознайомитися з теоретичною частиною лабораторної роботи.
- Вивчити принцип роботи датчика.
- Скласти розгорнуту специфікацію датчика.
- Вивчити та виконати алгоритм налаштування датчика.

Приклад виконання

Перший код програми стосується калібрування датчику. На рисунку 4 зображено першу частину коду. Код починається з визначення пінів, які використовуються для роботи датчика кольору TCS230:

- S0 та S1 відповідають за встановлення частоти вихідного сигналу датчика.
- S2 та S3 використовуються для вибору кольору (червоного, зеленого чи синього).

- sensorOut — це вихідний пін, який передає ширину імпульсу, відповідну інтенсивності кольору.

Крім того, визначаються калібрувальні значення для кожного кольору (redMin, redMax, greenMin тощо), отримані під час попереднього калібрування.

У функції setup() встановлюються початкові параметри:

- Піни S0, S1, S2, S3 налаштовуються як вихідні, а sensorOut — як вхідний.
- Частота вихідного сигналу датчика встановлюється на 20% (для зменшення навантаження на Arduino).
- Ініціалізується серійний монітор для виведення результатів у режимі реального часу.

```
#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8
// Calibration Values
// Get these from Calibration Sketch
int redMin = 25; // Red minimum value
int redMax = 445; // Red maximum value
int greenMin = 36; // Green minimum value
int greenMax = 433; // Green maximum value
int blueMin = 32; // Blue minimum value
int blueMax = 353; // Blue maximum value
// Variables for Color Pulse Width Measurements
int redPW = 0;
int greenPW = 0;
int bluePW = 0;
// Variables for final Color values
int redValue;
int greenValue;
int blueValue;
void setup() {
  // Set S0 - S3 as outputs
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  // Set Sensor output as input
  pinMode(sensorOut, INPUT);
  // Set Frequency scaling to 20%
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);
  // Setup Serial Monitor
  Serial.begin(9600);
}
```

Рисунок 4 – Лістинг програми

В головному циклі по черзі зчитуються значення кольорів (рисунок 5):

Викликається функція `getRedPW()`, яка налаштовує пін $S2 = \text{LOW}$ і $S3 = \text{LOW}$ для читання імпульсів червоного кольору.

Виміряна ширина імпульсу (`redPW`) масштабується в діапазон 0–255 за допомогою функції `map()`.

Аналогічно викликається функція `getGreenPW()`, де пін $S2 = \text{HIGH}$ і $S3 = \text{HIGH}$.

Читання виконується функцією `getBluePW()` з піном $S2 = \text{LOW}$ і $S3 = \text{HIGH}$.

Після зчитування інтенсивностей для кожного кольору результати виводяться у серійний монітор. Крім числових значень, визначається, який із кольорів має найбільшу інтенсивність, і повідомляється про це.

```
void loop() {
  // Read Red value
  redPW = getRedPW();
  // Map to value from 0-255
  redValue = map(redPW, redMin, redMax, 255, 0);
  // Delay to stabilize sensor
  delay(200);
  // Read Green value
  greenPW = getGreenPW();
  // Map to value from 0-255
  greenValue = map(greenPW, greenMin, greenMax, 255, 0);
  // Delay to stabilize sensor
  delay(200);
  // Read Blue value
  bluePW = getBluePW();
  // Map to value from 0-255
  blueValue = map(bluePW, blueMin, blueMax, 255, 0);
  // Delay to stabilize sensor
  delay(200);
  // Print output to Serial Monitor
  Serial.print("Red = ");
  Serial.print(redValue);
  Serial.print(" - Green = ");
  Serial.print(greenValue);
  Serial.print(" - Blue = ");
  Serial.println(blueValue);
  if(redValue > greenValue && redValue > blueValue){
    Serial.println(" - RED detected!");
  }
  if(greenValue > redValue && greenValue > blueValue){
    Serial.println(" - GREEN detected!");
  }
  if(blueValue > redValue && blueValue > greenValue){
    Serial.println(" - BLUE detected!");
  }
}
```

Рисунок 5 – Лістинг програми

На рисунку 6 зображено функції для зчитування кольорів.

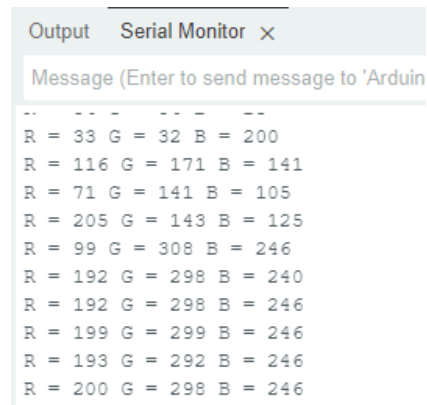
Функції `getRedPW()`, `getGreenPW()` та `getBluePW()` забезпечують налаштування датчика для вимірювання потрібного кольору:

- Вони встановлюють потрібні значення пінів S2 та S3, щоб датчик почав працювати у відповідному режимі.
- Після цього ширина імпульсу вимірюється функцією `pulseIn(sensorOut, LOW)` і повертається у функцію виклику.

```
// Function to read Red Pulse Widths
int getRedPW() {
    // Set sensor to read Red only
    digitalWrite(S2,LOW);
    digitalWrite(S3,LOW);
    // Define integer to represent Pulse Width
    int PW;
    // Read the output Pulse Width
    PW = pulseIn(sensorOut, LOW);
    // Return the value
    return PW;
}
// Function to read Green Pulse Widths
int getGreenPW() {
    // Set sensor to read Green only
    digitalWrite(S2,HIGH);
    digitalWrite(S3,HIGH);
    // Define integer to represent Pulse Width
    int PW;
    // Read the output Pulse Width
    PW = pulseIn(sensorOut, LOW);
    // Return the value
    return PW;
}
// Function to read Blue Pulse Widths
int getBluePW() {
    // Set sensor to read Blue only
    digitalWrite(S2,LOW);
    digitalWrite(S3,HIGH);
    // Define integer to represent Pulse Width
    int PW;
    // Read the output Pulse Width
    PW = pulseIn(sensorOut, LOW);
    // Return the value
    return PW;
}
```

Рисунок 6 – Лістинг програми

Для калібрування датчика спочатку треба піднести до датчика лист білого кольору. Дивлячись у серійний монітор (рис. 7), записати найменші значення кожного кольору. Після цього, підносячи лист чорного кольору, записати найбільші значення кольорів. Ці значення будуть використанні в наступному коді для точної ідентифікації кольору деталі.



```

Output  Serial Monitor  x
Message (Enter to send message to 'Arduin
..  ..  ..  ..  ..
R = 33 G = 32 B = 200
R = 116 G = 171 B = 141
R = 71 G = 141 B = 105
R = 205 G = 143 B = 125
R = 99 G = 308 B = 246
R = 192 G = 298 B = 240
R = 192 G = 298 B = 246
R = 199 G = 299 B = 246
R = 193 G = 292 B = 246
R = 200 G = 298 B = 246

```

Рисунок 7 – Серійний монітор

Наступний код складається з частини минулого коду. Шість значень калібрування, які було отримано з першого ескізу, вводяться у верхній частині ескізу, треба замінити «0» своїми фактичними значеннями (рис. 8).

```

#define S0 4
#define S1 5
#define S2 6
#define S3 7
#define sensorOut 8
int redMin = 25; // Red minimum value
int redMax = 445; // Red maximum value
int greenMin = 36; // Green minimum value
int greenMax = 433; // Green maximum value
int blueMin = 32; // Blue minimum value
int blueMax = 353; // Blue maximum value

// Variables for Color Pulse Width Measurements

int redPW = 0;
int greenPW = 0;
int bluePW = 0;

// Variables for final Color values

int redValue;
int greenValue;
int blueValue;

void setup() {

  // Set S0 - S3 as outputs
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);

  // Set Sensor output as input
  pinMode(sensorOut, INPUT);

  // Set Frequency scaling to 20%
  digitalWrite(S0,HIGH);
  digitalWrite(S1,LOW);

  // Setup Serial Monitor
  Serial.begin(9600);
}

```

Рисунок 8 – Лістинг коду

Також є три нові змінні, визначені для значень RGB, які треба вивести.

У циклі (рис.9) зчитуються кожне зі значень за допомогою тих самих функцій, що й у попередньому скетчі. Потім використовується функцію `map` Arduino, щоб змінити їх на значення RGB, використовуючи значення калібрування як еталон.

Нарешті, виводяться значення на послідовний монітор. За потреби можна внести незначні коригування значень калібрування.

Показання мають відповідати значенням RGB об'єкта, що сканується.

```
void loop() {
    // Read Red value
    redPW = getRedPW();
    // Map to value from 0-255
    redValue = map(redPW, redMin, redMax, 255, 0);
    // Delay to stabilize sensor
    delay(200);

    // Read Green value
    greenPW = getGreenPW();
    // Map to value from 0-255
    greenValue = map(greenPW, greenMin, greenMax, 255, 0);
    // Delay to stabilize sensor
    delay(200);

    // Read Blue value
    bluePW = getBluePW();
    // Map to value from 0-255
    blueValue = map(bluePW, blueMin, blueMax, 255, 0);
    // Delay to stabilize sensor
    delay(200);

    // Print output to Serial Monitor
    //Serial.print("Red = ");
    //Serial.print(redValue);
    //Serial.print(" - Green = ");
    //Serial.print(greenValue);
    //Serial.print(" - Blue = ");
    //Serial.println(blueValue);

    if(redValue > greenValue && redValue > blueValue){
        | | Serial.println("Red");
    }
    if(greenValue > redValue && greenValue > blueValue){
        | Serial.println("Green");
    }
    if(blueValue > redValue && blueValue > greenValue){
        | Serial.println("Blue");
    }
}
```

Рисунок 9 – Лістинг коду

Контрольні запитання

1. Принцип роботи досліджуваного датчика?
2. Принцип підключення датчика?
3. Які ще існують види датчиків ідентифікації кольору?
4. Переваги та недоліки даного типу датчика?
5. Призначення виходів датчика?
6. Як виконується налаштування датчика?

Лабораторна робота №2

Тема: Дослідження роботи системи керування автоматизованим сортуванням деталей за їх кольоровими характеристиками.

Мета: Вивчення особливостей обміну даними через Node-Red та OPC UA Server, коду програми та алгоритму роботи системи.

Теоретичні відомості

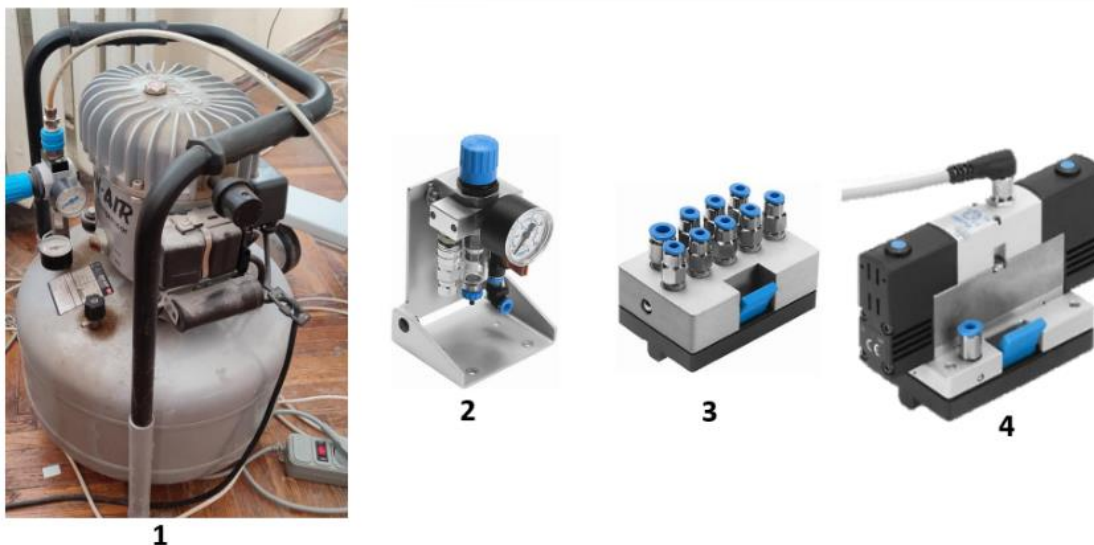
Система сортування деталей по кольору з допомогою пневматичних елементів складається з таких елементів:

- Датчик кольору з платою Arduino;
- Програма в Node-Red;
- Активованій сервер в OPC UA Server;
- Програма в TIA Portal.

Лістинг програми для Node-Red та TIA Portal надано разом з методичними вказівками до виконання даних лабораторних робіт. Для виконання роботи треба підключити плату Arduino до інженерної станції лабораторного стенду, завантажити через програмне забезпечення Arduino IDE код, використаний в минулій лабораторній роботі. Також, треба підключити пневматичну систему. Після цього запустити Node-Red і імпортувати код програми для отримання даних від датчику кольору. Після запуску OPC UA SERVER за адресою ПЛК стенду, відкрити проект лабораторного стенду і запустити виконання процесу через HMI.

Хід роботи

Для використання пневматичної системи треба використати дані елементи. Стенд складається з пневматичної (рис.10) та електричної системи (рис.11).



1 – масляний компресор; 2 – пусковий клапан; 3 - колектор 4 – х3/2 розподільник соленоїдний
Рисунок 10 – Використані елементи пневмоавтоматики стенду

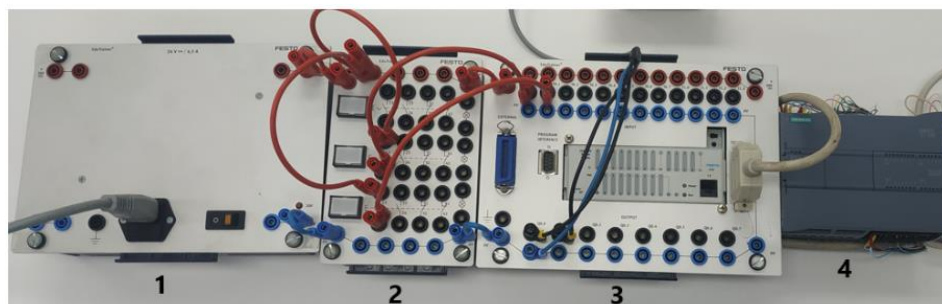


Рисунок 11 – Використані електричні елементи стенду

Після цього необхідно:

1. Завантажити проект датчику кольору в плату Arduino.
2. Запустити Node-Red та імпортувати код обміну даними.
3. Запустити OPC UA Server.
4. Запустити проект TIA Portal.

Після запуску програми в Arduino Ide та Node-Red, необхідно запустити OPC UA Server та налаштувати запуск серверу. Для цього треба натиснути Додати сервер та налаштувати айпі адресу та порт необхідного ПЛК стенду. Додавання серверу в OPC UA Server зображено на рисунку 12.

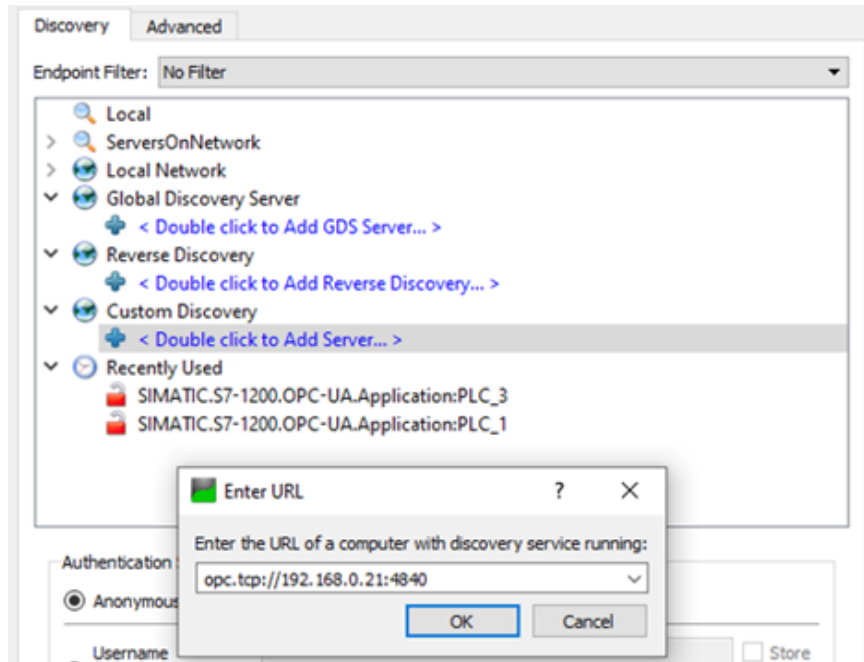


Рисунок 12 – Додавання серверу

Для запуску роботи програми потрібно перейти за допомогою кнопки Сортування кольору на вікно з запуском системи сортування (рис. 12). Після цього перевести режим роботи в автоматичний та натиснути «Пуск АВТ». Також необхідно встановити на платформу кольорову деталь навпроти датчика кольору, що супроводжується візуалізацією .

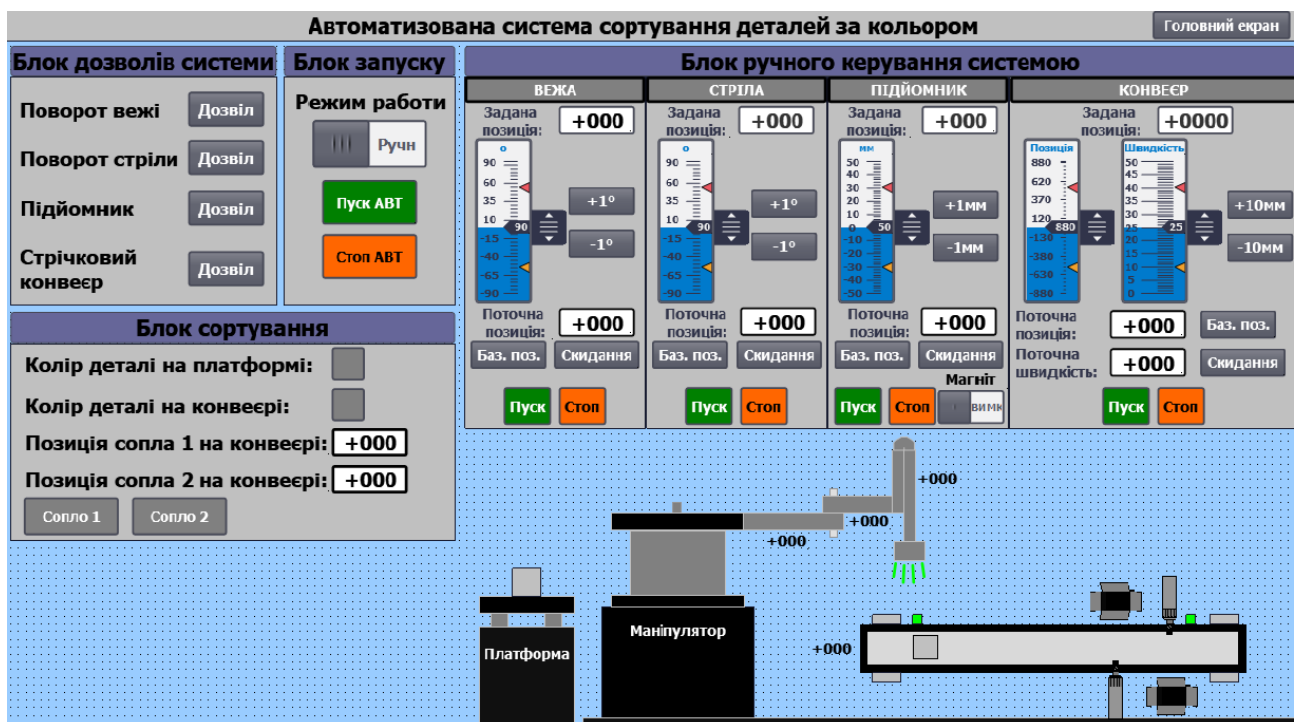


Рисунок 12 – НМІ сортування кольором

Дана панель дозволяє відстежувати параметр кольору деталі на платформі, на конвеєрі та активацію пневмосопел. Оператор також може задавати значення положення пневмосопел на конвеєрі. Також на панелі можна спостерігати зображення конвеєрної стрічки з видом зверху для анімації переміщення вантажу, з урахуванням його кольору, та його наступне сортування.

Завдання

1. Дослідити код програми.
2. Розглянути принцип роботи системи сортування за кольором.
3. Розглянути роботу програми та візуалізацію.
4. Дослідити реалізацію анімації на панелі оператора.

Контрольні запитання

1. Принцип роботи програми ідентифікації кольору?
2. Принцип роботи програми обміну даних в Node-Red?
3. Принцип роботи серверу в OPC UA Server?
4. Принцип роботи системи сортування за кольором.
5. Принцип роботи анімації переміщення вантажу?