

Національний університет кораблебудування
імені адмірала Макарова

Навчально-науковий інститут автоматики та електротехніки

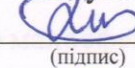
(повна назва інституту/факультету)

Кафедра автоматики

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри



I. С. Білюк

(підпис)

« 15 » 12 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня магістра

зі спеціальності 141 Електроенергетика, електротехніка та електромеханіка
(код та назва спеціальності)

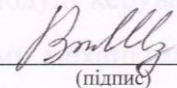
освітня програма Електромеханічні системи автоматизації та електропривод
(назва освітньої програми)

на тему: Проектування системи керування робота-маніпулятора

Виконав: студент групи 6371м

Шамраєв Валерій Віталійович

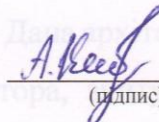
(прізвище, ім'я, по батькові)



(підпис)

Керівник асистент. каф. автоматики, к.т.н.,
Козлов А.Ю.

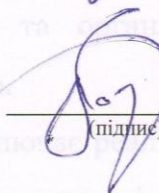
(посада, науковий ступінь, вчене звання, прізвище та ініціали)



(підпис)

Рецензент доцент кафедри СЕЕС,
к.т.н., доцент Новогрецький С.М.

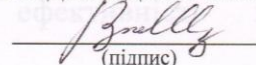
(посада, науковий ступінь, вчене звання, прізвище та ініціали)



(підпис)

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

Студент



(підпис)

м. Миколаїв – 2025 року

АНОТАЦІЯ

Магістерська кваліфікаційна робота присвячена розробленню, проєктуванню та впровадженню програмованого логічного контролера (ПЛК) на основі мікроконтролера PIC16F1937 для підвищення ефективності та функціональних можливостей системи керування робота-маніпулятора МП-9С. Необхідність створення компактного, доступного та легко програмованого контролера зумовлена потребою удосконалення існуючих навчальних і промислових робототехнічних комплексів, у яких використовуються застарілі або малоефективні блоки керування.

У роботі проведено аналіз сучасних типів роботів-маніпуляторів, принципів побудови їх систем керування та вимог до апаратного забезпечення. На основі порівняльного аналізу мікроконтролерних платформ обґрунтовано вибір PIC16F1937 як базового елемента для створення ПЛК завдяки його функціональній гнучкості, підтримці апаратних модулів, низькому енергоспоживанню та можливості обробки великої кількості сигналів введення-виведення.

У процесі дослідження розроблено структурно-логічну схему магістерської роботи, реалізовано повну електричну схему ПЛК, яка включає модулі керування входами та виходами, драйвер ULN2803 для роботи з виконавчими механізмами, інтерфейс RS-485 для зовнішнього обміну даними, блок індикації на основі LCD-дисплея та модуль локального керування на основі набору кнопок. Дана архітектура дозволяє забезпечити надійне комутування приводів маніпулятора, зчитування датчиків положення, реалізацію алгоритмів позиціонування та організацію дистанційної взаємодії з комп'ютером або промисловою мережею.

У роботі розроблено програмне забезпечення ПЛК, яке включає реалізацію алгоритмів керування маніпулятором, обробку даних сенсорів, комунікаційні протоколи та механізми діагностики. Проведено моделювання та аналіз працездатності системи, що підтвердило ефективність застосованого мікроконтролерного рішення.

Результатом роботи є створення універсального ПЛК, який дозволяє модернізувати систему керування маніпулятором МП-9С, підвищити його

функціональність, точність, надійність і адаптивність. Запропонована система може бути використана у навчальних лабораторіях, дослідницьких стендах, а також у промислових комплексах малої та середньої складності.

Ключові слова: програмований логічний контролер, мікроконтролер, робот-маніпулятор, система керування, автоматизація, позиціонування.

ABSTRACT

The master's qualification thesis is devoted to the development, design, and implementation of a programmable logic controller (PLC) based on the PIC16F1937 microcontroller in order to improve the efficiency and functional capabilities of the control system of the MP-9S robotic manipulator. The need to create a compact, affordable, and easily programmable controller is driven by the necessity to modernize existing educational and industrial robotic systems that utilize outdated or inefficient control units.

The thesis presents an analysis of modern types of robotic manipulators, principles of their control system design, and hardware requirements. Based on a comparative analysis of microcontroller platforms, the PIC16F1937 was selected as the core element for PLC development due to its functional flexibility, support for hardware modules, low power consumption, and ability to process a large number of input and output signals.

During the research, a structural and logical scheme of the master's thesis was developed, and a complete electrical circuit of the PLC was designed. The circuit includes input and output control modules, a ULN2803 driver for actuator control, an RS-485 interface for external data exchange, an indication unit based on an LCD display, and a local control module based on a set of buttons. This architecture ensures reliable switching of manipulator drives, reading of position sensors, implementation of positioning algorithms, and organization of remote interaction with a computer or industrial network.

The PLC software was developed, including the implementation of manipulator control algorithms, sensor data processing, communication protocols, and diagnostic mechanisms. Simulation and performance analysis of the system were conducted, confirming the effectiveness of the applied microcontroller-based solution.

The result of the work is the creation of a universal PLC that enables modernization of the MP-9S manipulator control system, increasing its functionality, accuracy, reliability, and adaptability. The proposed system can be used in educational laboratories, research stands, as well as in small- and medium-complexity industrial systems.

Keywords: programmable logic controller, microcontroller, robotic manipulator, control system, automation, positioning

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ТА ОПИС ТЕХНІЧНОГО ОБЛАДНАННЯ.....	8
1.1.Основні типи роботів-маніпуляторів та їх застосування.....	8
1.2.Основні технічні характеристики маніпулятора МП-9С.....	14
1.3.Системи керування ЕЦПУ-6030 та програмне забезпечення для маніпулятора.....	20
Висновок за розділом.....	26
РОЗДІЛ 2 МЕТОДОЛОГІЯ МАГІСТЕРСЬКОГО ДОСЛІДЖЕННЯ	28
2.1 Побудова структурно-логічної схеми магістерського дослідження.....	28
2.2 Структурно-логічна схема магістерського дослідження	34
2.3 Вибір методу вирішення поставленої задачі дослідження	35
РОЗДІЛ 3 ВИБІР МІКРОКОНТРОЛЕРА ТА ОБҐРУНТУВАННЯ ЙОГО АРХІТЕКТУРИ ДЛЯ ПЛК	39
3.1 Вибір мікроконтролера.....	39
3.2 Технічні характеристики мікроконтролера PIC16F1937	44
3.3 Реалізація інтерфейсу зв'язку RS485.....	60
3.4 Структура та опис спроектованої схеми.....	71
Висновки за розділом	74
РОЗДІЛ 4 ВПРОВАДЖЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ	76
4.1 Розробка та опис функціональної схеми маніпулятора	76
4.2 Керуючі алгоритми програми.....	78
4.3 Код програми.....	94
Висновки за розділом	113
ВИСНОВОК.....	114
Список використаних джерел	115
ДОДАТОК А Слайди мультимедійної презентації	117

ВСТУП

Сучасна промисловість вимагає впровадження новітніх технологій для забезпечення високої ефективності та безпеки виробничих процесів. Одним із ключових елементів для досягнення цих цілей є автоматизація систем управління, яка дозволяє значно підвищити продуктивність, знизити вплив людського фактору та покращити якість виконання технологічних операцій. Промислові роботи-маніпулятори займають важливе місце в автоматизованих лініях, оскільки вони забезпечують точність позиціонування та швидкість переміщення об'єктів у просторі.

Метою даної магістерської роботи є розробка та впровадження автоматизованої системи керування роботом-маніпулятором МП-9С. Для досягнення цієї мети використано сучасну елементну базу, зокрема мікроконтролер PIC16F1937 та промисловий інтерфейс RS-485, що дозволяє реалізувати необхідні алгоритми управління, забезпечити завадостійкість каналів зв'язку та гнучкість роботи маніпулятора.

Завдання роботи: вивчити види та технічні характеристики роботів-маніпуляторів, зокрема моделі МП-9С, обґрунтувати вибір мікроконтролера PIC16F1937 та методологію дослідження, реалізувати апаратну частину інтерфейсу RS-485, розробити функціональну схему автоматизованої системи та керуючі алгоритми, написати програму для мікроконтролера та розглянути питання охорони праці при експлуатації системи.

Впровадження автоматизованої системи керування повинно дозволити підвищити надійність та точність роботи маніпулятора, забезпечити стабільний обмін даними в умовах промислових завод та покращити контроль за виконанням технологічних операцій, що в кінцевому результаті сприятиме підвищенню ефективності виробничих процесів.

РОЗДІЛ 1

АНАЛІЗ ТА ОПИС ТЕХНІЧНОГО ОБЛАДНАННЯ

1.1 Основні типи роботів-маніпуляторів та їх застосування

Основним елементом промислового робота є маніпулятор. Він фактично працює як механічна «рука» людини, яка виконує переміщення й орієнтацію об'єктів у просторі. У загальному випадку його структура включає:

- задавальний орган (систему керування),
- виконавчі та передавальні механізми,
- робочий орган із захоплювачем або інструментом

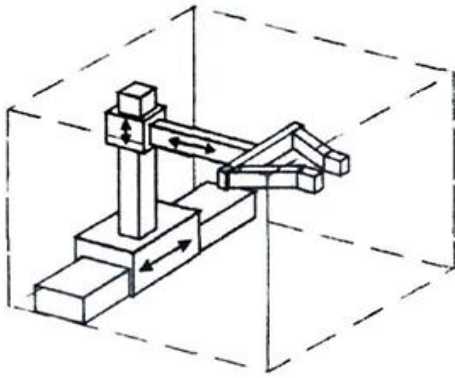
Залежно від способу керування, маніпулятори бувають біотехнічні (людина-оператор керує ними) та автоматичні. До автоматичних відносять автооператори, промислові роботи та системи з інтерактивним керуванням.

Промисловий робот визначається як перепрограмований автоматичний маніпулятор. Його головне завдання – це переміщення в просторі об'єктів маніпулювання: заготовок, деталей, інструментів чи контрольно-вимірювальних пристроїв.

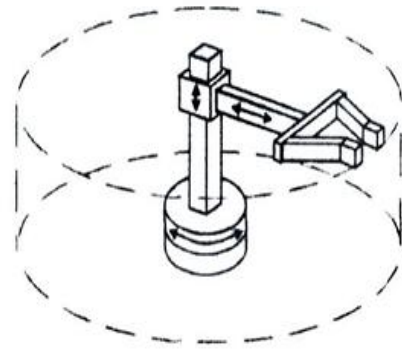
У цілому, залежно від конструктивної схеми механічної системи, «рука» робота може працювати в робочому об'ємі різної форми. Саме цей об'єм і визначає можливості робота. Всередині нього рука рухається в різних системах координат (декартова, циліндрична, сферична та ін.), як показано на схемах.

Варто додати, що в сучасному виробництві роботи рідко працюють окремо. Вони часто об'єднуються в роботизовані технологічні комплекси. Такий комплекс може складатися з верстатів із числового програмного керування, одного чи кількох промислових роботів, транспортних і накопичувальних пристроїв, об'єднаних загальною системою керування. Наприклад, найпростіший тип – це роботизована технологічна ланка, де робот виконує допоміжні операції біля одного верстата. Більш складні утворюють роботизовані дільниці або лінії, де кілька роботів взаємодіють у єдиному технологічному процесі

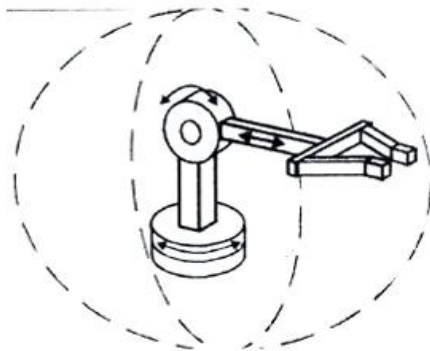
Залежно від конструктивної схеми маніпуляторної системи рука промислового робота і маніпулятора може знаходитись у робочому об'ємі, що має ту чи іншу форму, а її рухи здійснюються в різноманітних системах координат (рис1.1).



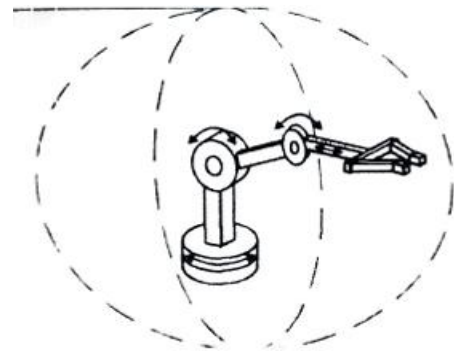
а) Декартова координатна конструкція



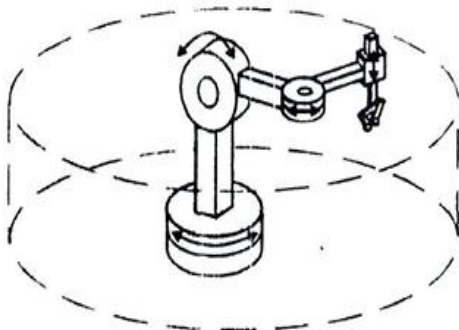
б) Циліндрична координатна конструкція



в) сферична координатна конструкція



г) конструкція вертикально шарнірного
рукава



г) конструкція горизонтального –
шарнірного рукава

Рис.1.1 Конструктивні схеми маніпуляторної системи та її основні рухи

Класифікацію промислових робіт можна проводити за різними ознаками: за характером виконуваних операцій, за ступенем спеціалізації, галуззю застосування, числом ступенів рухомості та іншими критеріями.

За ступенем спеціалізації промислові роботи поділяють на три основні групи: спеціальні, цільові (їх ще називають спеціалізованими) та багатоцільові. Спеціальні роботи зазвичай виконують лише одну визначену операцію або обслуговують конкретну модель обладнання. Цільові роботи призначені для певного виду операцій, наприклад, тільки для зварювання, фарбування чи складання. А от багатоцільові промислових робіт більш універсальні – вони можуть виконувати і основні, і допоміжні операції, у тому числі й ті, що потребують різних прийомів.

Якщо брати класифікацію за характером виконуваних операцій, то виділяють три групи:

Виробничі (технологічні) промислових робіт. Вони виконують основні операції технологічного процесу. Це, наприклад, зварювання, фарбування, складання чи навіть механічна обробка. По суті, ці роботи беруть безпосередню участь у виробництві.

Піднімально-транспортні (допоміжні) промислових робіт. Вони виконують дії типу «взяти – перенести – покласти». Такі роботи часто застосовуються для завантаження й розвантаження верстатів, встановлення і зняття деталей чи інструментів, а також у транспортно-складальних операціях.

Універсальні промислових роботи. Вони поєднують функції двох попередніх груп і можуть працювати як на основних, так і на допоміжних операціях. Тобто вони є найбільш гнучкими.

Функціональні можливості промислового робота в більшості випадків визначаються типом системи програмного керування. Більшість сучасних робіт є жорстко програмованими, тобто їхня програма дій задана заздалегідь і не змінюється в процесі роботи. Але існують також адаптивні роботи – вони можуть коригувати свою програму на основі інформації про навколишнє середовище, використовуючи сенсорні системи. Ще більш сучасні – це гнучко програмовані

(або інтегральні) промислові роботи, які самі формують програму своїх дій, виходячи з поставленої мети та отриманої інформації.

У практиці використовуються три основні типи системи програмного керування:

- позиційні (робот рухається від точки до точки),
- контурні (рух відбувається по неперервній траєкторії),
- комбіновані (поєднують обидва підходи).

Також за способом формалізації задавальної інформаційної системи програмного керування поділяють на:

- циклові програмного керування,
- аналогові програмного керування,
- числові програмного керування,
- аналого-числові, або гібридні.

Найпростішими вважаються роботи з циклові програмного керування, бо їхня програма складається з послідовності дій, які відтворюються за допомогою упорів і кінцевих перемикачів. В аналогових системах використовується безперервно змінна інформація, наприклад у вигляді напруги чи струму. У числові програмного керування дані зберігаються у вигляді цифрових кодів на носії, а гібридні системи поєднують обидва підходи.

Класифікація промислових роботів може здійснюватися за різними ознаками: за характером виконуваних операцій, ступенем спеціалізації, галуззю застосування, числом ступенів рухомості та іншими параметрами.

Також основними конструктивно-технологічними показниками промислових роботів є:

Робоча зона – простір, у якому може знаходитися робочий орган маніпулятора. Вона визначається лінійними та кутовими переміщеннями його рухомих ланок. Якщо працює кілька промислових роботів одночасно, виділяють ще й «зону спільного обслуговування».

Траєкторія переміщень – встановлюється за принципом мінімізації шляху та кількості ступенів рухомості. Вибір залежить від форми і розташування робочих зон обладнання, числа позицій і планування комплексу.

Швидкості переміщень – задаються відповідно до необхідної продуктивності, траєкторії рухів та часу можливих простоїв обладнання.

Вантажопідйомність – максимальна маса об'єкта (разом із захоплювачем), яку рука робота може переміщати при заданих умовах. Для деяких промислові роботи важливим є також зусилля затиску і крутний момент.

Число ступенів рухомості – кількість незалежних координатних рухів відносно опорної системи (основи) робота.

Похибка позиціювання – відхилення фактичної позиції від заданої програмою при повторних переміщеннях. Вона визначається у лінійних або кутових одиницях.

Мобільність – здатність робота здійснювати рухи. За цим показником виділяють стаціонарні та пересувні роботи.

Тип привода – вибирається з урахуванням енергетичних та технологічних вимог (електромеханічний, пневматичний, гідравлічний, пневматичний тощо).

Керуюча система – забезпечує виконання програм руху. Важливими показниками є спосіб позиціювання, кількість керованих координат, ємність пам'яті та швидкодія.

Захватні пристрої та інструменти – проектуються відповідно до параметрів об'єкта маніпулювання та послідовності виконання операцій.

Прикладом одного з сучасних роботів може слугувати лінійка Epson E2L Robots (моделі E2L65 та E2L85) (рис1.2). Це роботи типу SCARA, які мають чотири осі свободи руху та відзначаються високою точністю й швидкістю роботи.

Основні технічні характеристики:

- робочий радіус – від 650 до 850 мм залежно від моделі;
- максимальна швидкість переміщення – до 6100 мм/с;
- точність позиціювання – до $\pm 0,02$ мм;
- вантажопідйомність – 2 кг номінально та до 5 кг максимально;

- момент інерції – 0,010...0,120 кг*м²;
- час спрацювання – близько 0,38–0,44 с;
- потужність двигунів по осях – від 150 до 400 Вт;
- зусилля притискання по осі Z – до 200 Н.

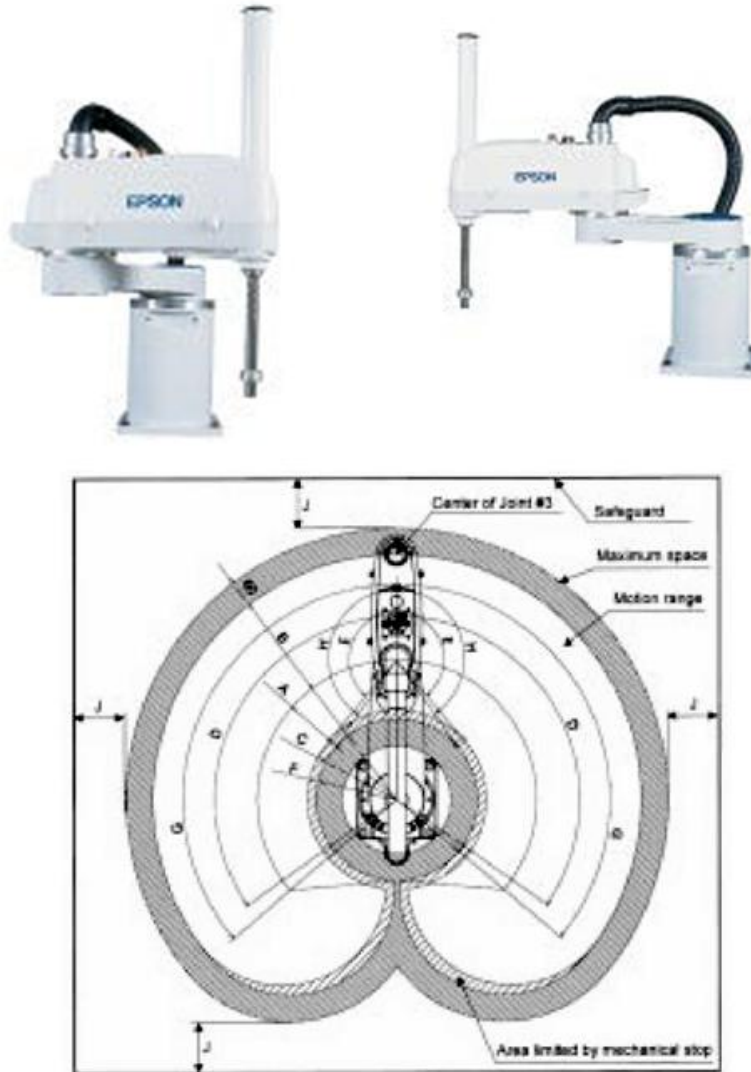


Рисунок 1.2 – Конструкція та робоча зона роботів Epson E2L

Дані моделі використовують електричну систему живлення і мають невелику масу (близько 31 кг), що робить їх зручними для інтеграції у виробничі комплекси. Умови експлуатації досить широкі: від +5 до +40 °C при вологості до 80%.

Завдяки поєднанню високої швидкодії, точності та компактності, роботи Epson E2L широко застосовуються в автоматизованих виробничих системах для

виконання складальних операцій, пакування, транспортування дрібних деталей та інших технологічних завдань.

1.2. Основні технічні характеристики маніпулятора МП-9С

Як об'єкт автоматизації в даній роботі розглядається промисловий робот МП-9С. Цей маніпулятор належить до класу пневматичних пристроїв циклової дії та призначений для автоматизації допоміжних операцій, таких як встановлення та зняття деталей на технологічному обладнанні.

Загальне компонування роботизованого комплексу представлено на Рис. 1.3.

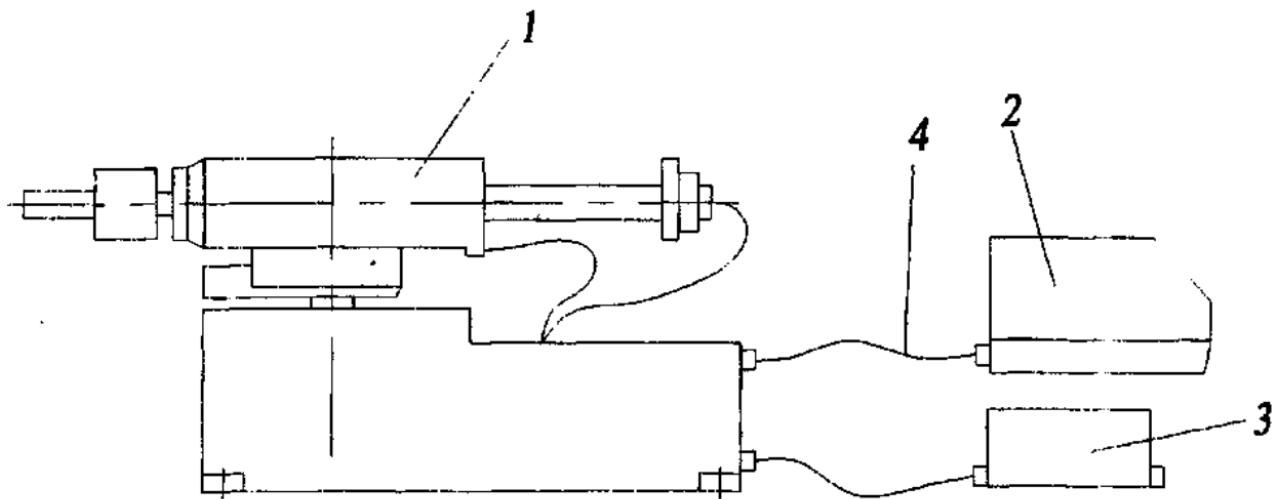


Рисунок 1.3. Схема промислового робота МП-9С: 1 – маніпулятор Т74.030.007; 2 – електронний цикловий програмний пристрій ЕЦПУ-6030 Т72.556.013; 3 – вузол підготовки повітря Т75.886.004; 4 – з'єднувальні кабелі.

Робот є підлоговим і функціонує в циліндричній системі координат. Кінематична структура забезпечує два поступальних і один обертальний рух. Особливістю даного циклового маніпулятора є дискретний характер позиціонування: ланки можуть перебувати лише в скінченному числі стійких станів (по дві точки позиціонування на кожен ступінь рухомості).

Кінематична схема механізму з трьома ступенями рухомості наведена на рис. 1.4

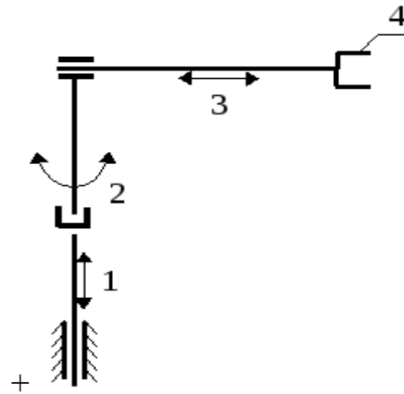


Рисунок 1.4. Кінематична схема маніпулятора:

1, 2, 3 – ступені рухомості; 4 – захват.

Конструктивно маніпулятор складається з основи, руки та захоплювального пристрою. В основі розміщені механізми підйому та повороту, а також пневмоапаратура. Всі виконавчі механізми (приводи) є пневматичними і керуються за допомогою пневморозподільників клапанного типу з електрокеруванням.

Принципова схема приводу одного ступеня рухомості показана на рис. 1.5. Як виконавчі двигуни використовуються пневмоциліндри одно- або двосторонньої дії з прямолінійним рухом поршня.

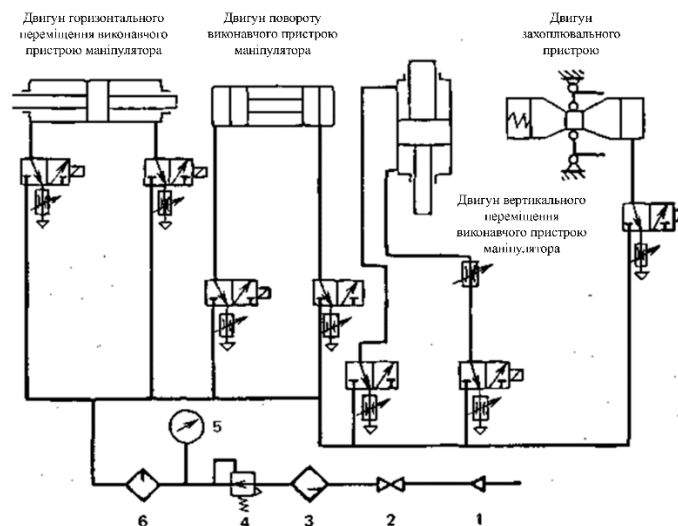


Рисунок 1.5. Пневматична схема приводу

Регулювання швидкості вихідної ланки здійснюється методом дроселювання вихлопу (змінюючи витрати стисненого повітря дроселем). Для організації зворотного зв'язку в системі керування використовуються електромагнітні контакти. Сигнал про завершення руху формується при наблизенні постійного магніту, встановленого на рухомій частині приводу, до геркона (керованого електромагнітного контакту). Для безударної зупинки в кінцевих положеннях застосовуються гідравлічні демпфери.

Розглянемо детально конструкцію основних вузлів маніпулятора, оскільки їх характеристики впливають на проектування системи керування.

Механізм підйому (рис. 1.6) виконаний за схемою з нерухомим штоком (2) і рухомим корпусом циліндра (4). Особливістю конструкції є відмінність робочих площ поршня, що впливає на динаміку підйому та опускання. Всередині штока розташований вал механізму повороту. Позиціювання по вертикалі налаштовується механічними упорами, на яких закріплені керовані електромагнітними контактами.

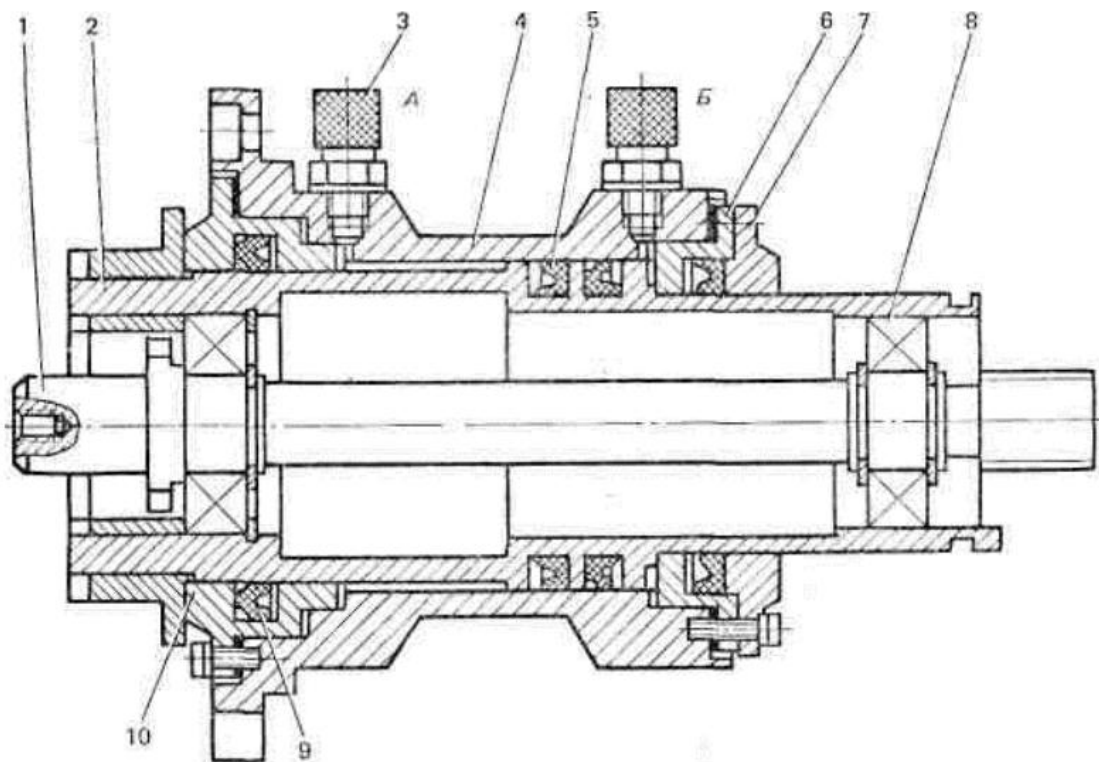


Рисунок 1.6. Конструкція механізму підйому

Механізм повороту (рис. 1.7.) реалізований на базі рейкової передачі. Шток-рейка (10) перетворює поступальний рух поршня в обертальний рух вала. Кут повороту обмежується упорами в муфті (рис. 1.8.), що з'єднує вал з виконавчим пристроєм.

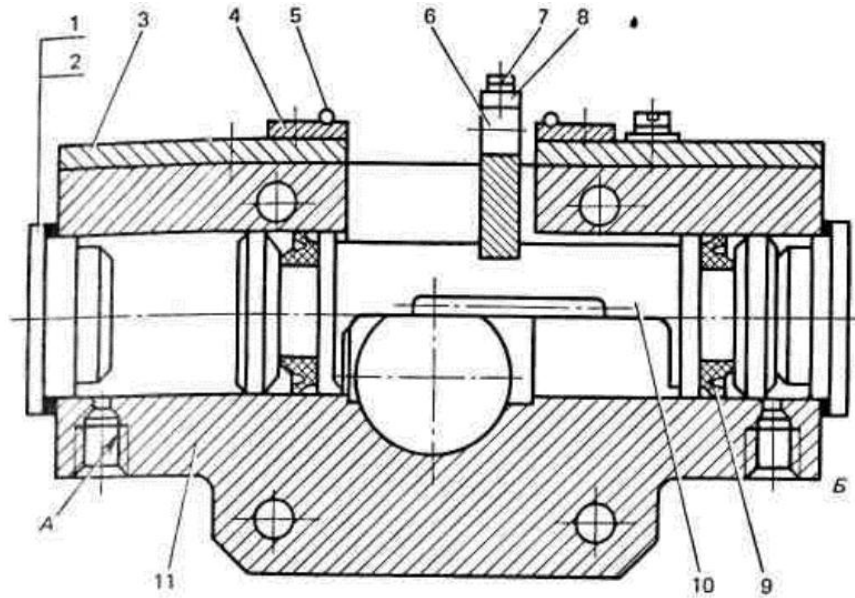


Рисунок 1.7. Конструкція механізму поворота

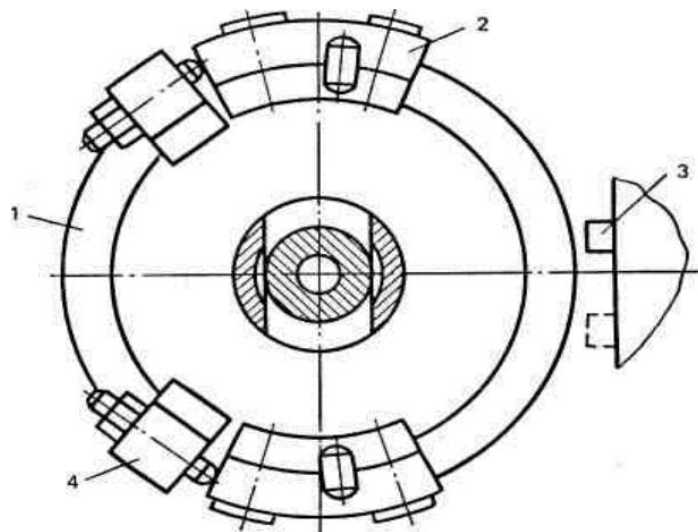


Рисунок 1.8. Розташування упорів механізму поворота в муфті

Виконавчий пристрій робота (рис. 1.9) забезпечує горизонтальне висунення схвата. Шток з поршнем (10) переміщується в гільзі, а напрямна (17) запобігає його провороту навколо осі. Налаштування вильоту здійснюється регулювальними упорами (7, 14) і мікрогвинтом (6).

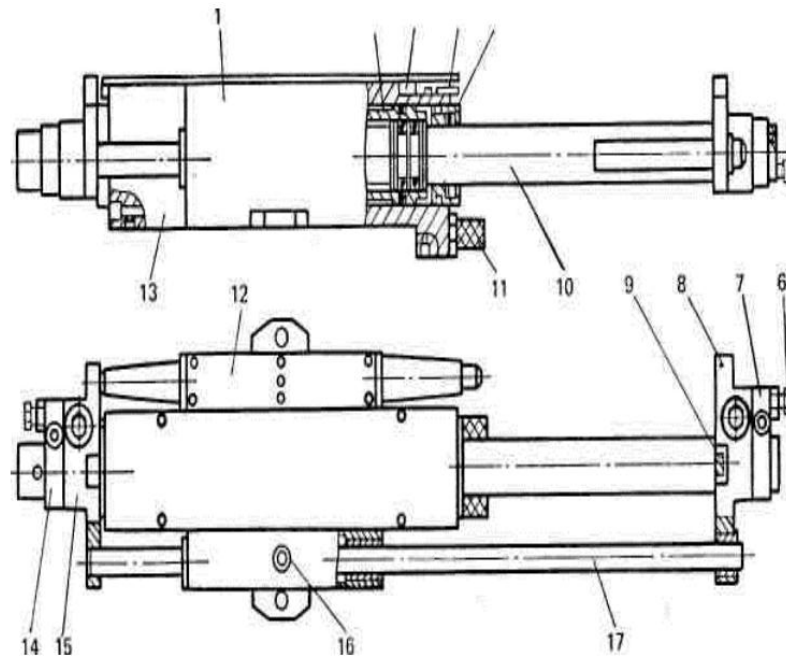


Рисунок 1.9. Конструкція виконавчого пристрою робота

Важливим елементом динамічної системи є гідравлічні амортизатори (демпфери), що забезпечують плавне гальмування ланок (рис. 1.10 та 1.11). Принцип їх дії заснований на дроселюванні рідини через зазор змінного перерізу при натисканні упора на шток демпфера.

Привід захоплювального пристрою реалізований у вигляді циліндра односторонньої дії зі зворотною пружиною. Стискання важелів відбувається при подачі повітря (рух поршня вліво), а розтискання — під дією пружин при знятті тиску.

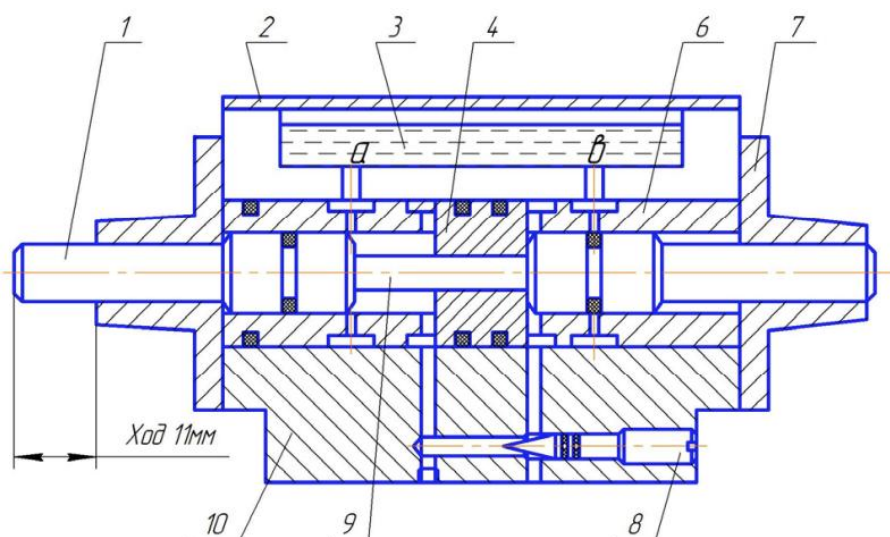


Рисунок 1.10. Гідравлічний амортизатор кута поворота руки: 1 – шток; 2 – кришка; 3 – мастило (І-20А); 8 – голка дросельна; 10 – корпус.

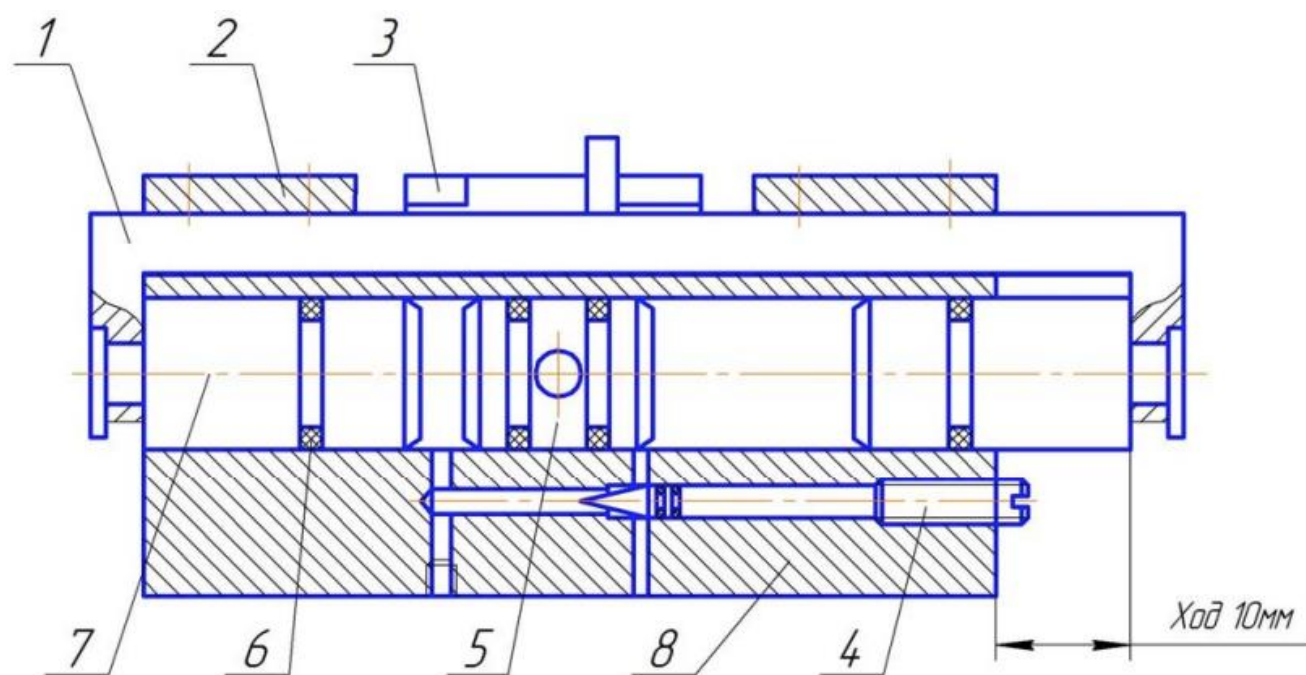


Рисунок 1.11. Гідравлічний амортизатор висунення руки

1.3. Системи керування ЕЦПУ-6030 та програмне забезпечення для маніпулятора

Керування маніпулятором МП-9С здійснюється за допомогою електронного циклового програмного пристрою ЕЦПП-6030 (ЕЦПУ-6030) (рис.1.12). Дана система належить до класу синхронних програмних автоматів із жорстким циклом керування та призначена для координації роботи двопозиційних ступенів рухомості маніпулятора та пов'язаного технологічного обладнання.



Рисунок 1.12. Прилад ЕЦПУ-6030

Технічні характеристики та принцип дії:

Взаємодія системи керування (СК) з виконавчими механізмами робота здійснюється електричними сигналами постійного струму напругою 24 В. Керуючі команди надходять на котушки електропневматичних розподільників, які комутують потоки стисненого повітря в порожнини пневмоприводів.

Зворотний зв'язок реалізовано за шляховим принципом: сигнал про виконання руху (за винятком роботи захватів) формується герконовими датчиками (КЕМ) при наближенні постійних магнітів. Для керування захватом використовується часовий принцип (програмована витримка часу). Перехід до

наступного кадру програми відбувається лише після отримання сигналу відповіді про виконання поточної команди.

Основні технічні параметри ЕЦПП-6030:

- Тип системи: Циклова;
- Кількість керованих ланок: до 6 (з них 4 — за шляховим принципом, 2 — за змішаним);
- Ємність програми: до 30 кадрів;
- Кількість технологічних команд: 6;
- Діапазон регулювання витримки часу: 0...0,7 с.
- Функціональна структура пристрою ЕЦПП-6030 представлена на рис. 1.13.

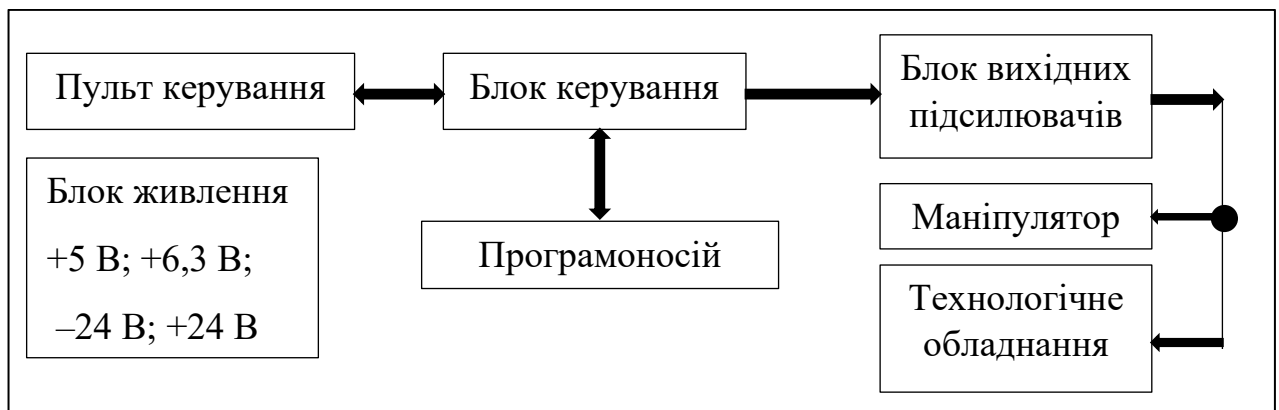


Рисунок 1.13. Структурна схема пристрою ЕЦПП-6030

Пристрій підтримує чотири режими роботи: «Ручний» (налагодження), «Команда» (покрокове відпрацювання), «Цикл» (одноразовий прогін) та «Автомат» (циклічна робота).

Принцип дії Блоку керування базується на цикличному зчитуванні та виконанні команд. Процес починається з ініціалізації схеми пуску-зупинки, яка переводить пристрій у робочий режим. При цьому вузол керування 1, що відповідає за логіку послідовності операцій, активує лічильник кадрів. Поточний стан лічильника визначає адресу команди, яка за допомогою дешифратора вибірки зчитується з програмоносія.

Зчитана інформація через схему узгодження надходить до вузла керування 2, де відбувається її розподіл за типом дії. Залежно від змісту команди, активується один із виконавчих модулів:

При надходженні команди руху сигнал передається на схему відпрацювання команд ланками маніпулятора, яка комутує відповідні силові виходи для керування пневмоприводами.

При необхідності взаємодії із зовнішнім обладнанням задіюється схема відпрацювання технологічних команд.

Для операцій, що потребують фіксованої часової затримки (наприклад, стискання захвату), вмикається схема відпрацювання команд витримки часу.

Ключовим етапом циклу є перевірка виконання команди. Сигнали від датчиків положення маніпулятора через входні кола Вузла узгодження надходять до схеми відпрацювання команд опитування. Ця схема порівнює реальний стан датчиків із заданим у програмі.

Як тільки умова завершення поточної операції виконується (наприклад, спрацював кінцевий вимикач або сплив час таймера), схема формування сигналу переходу генерує імпульс дозволу на виконання наступного кроку. Цей сигнал надходить назад до вузлу керування 1, де проходить через схему завадозахисту. Остання фільтрує хибні спрацювання та «брязкіт» контактів датчиків, після чого формує лічильний імпульс для лічильника кадрів. Значення лічильника збільшується на одиницю, ініціюючи вибірку та виконання наступної команди програми.

У випадку наявності у програмі команд розгалуження («Пропуск» або «Перехід»), керування передається відповідній схемі у вузлу керування 1, яка примусово змінює стан лічильника кадрів, минаючи стандартну послідовність.

Вузол керування 1 відповідає за адресну вибірку команд. Його ключовим елементом є лічильник кадрів, схема якого наведена на рис. 1.14. лічильник кадрів зберігає номер поточного кадру та адресує дешифратор вибірки кадрів (ДВК). Для захисту від помилкових спрацьовувань у схему введено завадозахист.

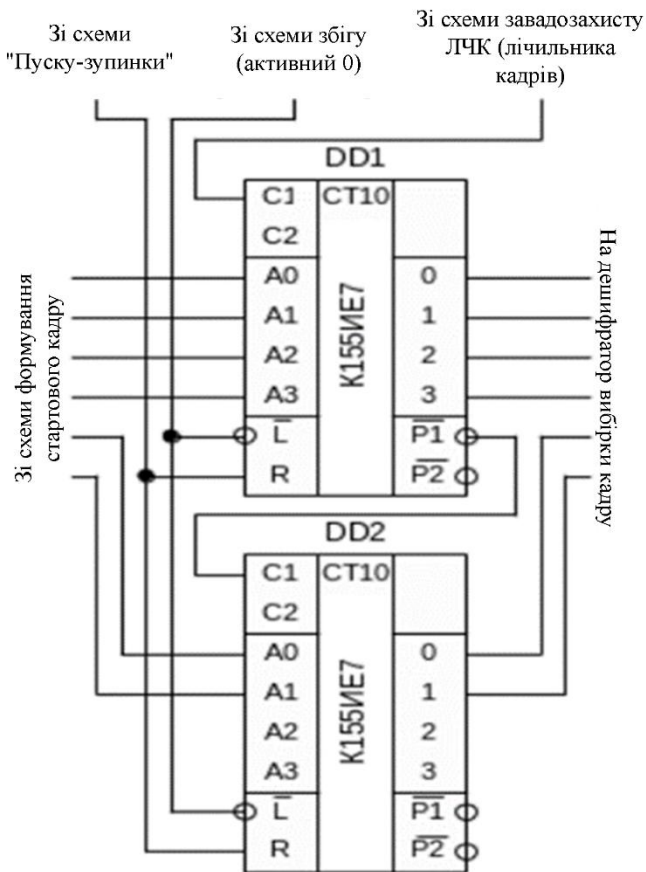


Рисунок 1.14. Принципова схема лічильника кадрів

Схема пуску-зупинки реалізована на D-тригері. При натисканні кнопки «Пуск» тригер переходить в одиничний стан, дозволяючи роботу автомата.

Загальний принцип функціонування блоку керування полягає у послідовній вибірці команд із програмоносія. Після встановлення тригера пуску-зупинки у вузол керування 1 формується сигнал вибірки ячейки програмоносія, інформація з якої через Вузол узгодження надходить до вузлу керування 2. Після виконання команди маніпулятором (сигнал від геркона), на виході схеми формування сигналу переходу до наступного кроку з'являється імпульс. Цей імпульс активує схему завадозахисту, яка, після придушення брязкоту контактів, формує сигнал додавання "1" у Лічильник кадрів, забезпечуючи перехід до виконання чергового кроку програми.

Пульт керування пристроєм ЕЦПП–6030 призначений для оперативного керування та відображення його стану. На лицьовій панелі пульта розташовані такі органи керування (Рис 1.15.) та має команди які казані в (Таб. 1.1.):



Рисунок 1.15 Лицьова панель пульта ЕЦПІ–6030

- Перемикач РЕЖИМ РОБОТИ: встановлює один із чотирьох режимів («Автомат», «Цикл», «Команда», «Ручний»).
- Кнопки ручного керування ланками маніпулятора.
- Кнопки керування програмою: ПУСК, СТОП, СКИНУТИ ЛЧК (Лічильник Кадрів), +1ЛЧК (одночасний перехід до наступного кадру).

Кнопка МЕРЕЖА.

На пульті також розміщені елементи індикації: табло індикації стану маніпулятора, табло «РОБОТА», табло індикації номера кроку та табло «МЕРЕЖА».

Таблиця 1.1.- система команд пристрою ЭЦПУ-6030.

Група команд	Номер команди	Назва команди	Код кадру	
			Верхнє поле	Нижнє поле
1	2	3	4	5
I	1	Висунення руки 	1	x
	2	Втягування руки 	2	x
	3	Поворот руки праворуч 	3	x
	4	Поворот руки ліворуч 	4	x
	5	Підйом руки 	x	1
	6	Опускання руки 	x	2
	7	Захватний пристрій зачинено 	x	5
1	2	3	4	5
I	8	Захватний пристрій відчинено 	x	6
II	9	Технологічна команда 1	9	1
	10	Теж саме, 2	9	2
	11	Теж саме, 3	9	3
	12	Теж саме, 4	9	4
	13	Теж саме, 5	9	5
	14	Теж саме, 6	9	6
III	15	Опитування 1	7	x
	16	Опитування 2	8	x
	17	Опитування 3	x	7
	18	Опитування 4	x	8
	19	Витримка часу	x	9
IV	20	Пропуск	9	7
	21	Перехід	9	8
	22	Зупинка	9	9
	23	Кінець програми	0	0

Програмоносій пристрою побудований на базі багатопозиційних десяткових плоских перемикачів. Перемикачі згруповані у 2 набірні поля по 30 штук у кожному, що забезпечує можливість оперативного ручного набору програми.

Разом із тим, такий апаратний спосіб задання алгоритму, хоч і вирізняється наочністю, є суттєвим стримуючим фактором в умовах сучасного динамічного виробництва. Фізичний процес набору програми на комутаційному полі характеризується високою трудомісткістю, оскільки вимагає від оператора механічного перемикавання десятків тумблерів для кожного кадру програми, що значно збільшує час простою обладнання під час переналагодження на нову технологічну операцію.

Крім того, відсутність у системі ЕЦПП-6030 енергонезалежної пам'яті для зберігання бібліотеки технологічних програм унеможливорює швидке перемикавання між різними типами деталей. Кожна зміна завдання вимагає повторного повного набору коду "з нуля", що неминуче призводить до підвищення ймовірності помилок оператора (так званого "людського фактора"). Відлагодження введеної програми ускладнене відсутністю сучасних засобів візуалізації стану входів/виходів та покрокової діагностики логіки.

Також слід зазначити, що електромеханічна природа пристрою введення (велика кількість контактних пар перемикачів) з часом призводить до зниження надійності системи через окислення контактів та вібраційні навантаження, характерні для промислових цехів. Відсутність мережевих інтерфейсів робить даний пристрій повністю ізольованим від інформаційної системи підприємства (АСК ТП), що не дозволяє реалізувати дистанційний моніторинг та збір статистики роботи робота.

Висновок за розділом

Детальний аналіз схемотехнічних та функціональних рішень пристрою ЕЦПП-6030 дозволяє зробити висновок, що, незважаючи на його історичну значущість та конструктивну надійність, дана система керування є морально та технічно застарілою.

Основні недоліки системи, які обмежують її ефективне використання на сучасному етапі автоматизації:

Жорстка циклічна логіка: Управління здійснюється за принципом програмного автомата з фіксованим циклом. Це унеможлиблює реалізацію складних адаптивних алгоритмів, гнучке керування траєкторією (окрім кінцевих точок) та швидку переналадку під нові технологічні завдання.

Обмежена програмна ємність: Ємність програми до 30 кадрів є недостатньою для сучасних виробничих процесів, що вимагають розгалужених та довгих алгоритмів.

Низький рівень інтеграції: Використання великої кількості дискретних логічних елементів (тригерів, лічильників) ускладнює діагностику, знижує загальну надійність та потребує значних експлуатаційних витрат у порівнянні з мікропроцесорними системами.

Таким чином, керуючий пристрій ЕЦПП-6030 на даний момент являє собою обмежений орган керування, який гостро потребує модернізації шляхом переходу на сучасну мікропроцесорну базу (наприклад, промисловий контролер або спеціалізований мікроконтролер). Необхідність такої модернізації є ключовою передумовою для подальшого проєктування ефективної системи керування роботом-маніпулятором у межах даної дипломної роботи.

РОЗДІЛ 2

МЕТОДОЛОГІЯ МАГІСТЕРСЬКОГО ДОСЛІДЖЕННЯ

2.1 Побудова структурно-логічної схеми магістерського дослідження

Метод (від грецького *methodos* – шлях до чого-небудь) – в найбільш загальному випадку означає засіб досягнення мети, спосіб дослідження явища, який визначає планомірний підхід до їх наукового пізнання та встановлення істини. Науковий метод – це спосіб пізнання явищ дійсності в їх взаємозв'язку та розвитку, спосіб досягнення поставленої мети і завдань дослідження і відповідає на запитання: «Як пізнавати?». Методика дослідження – це система правил використання методів, прийомів та способів для проведення будь-якого дослідження. Свідоме застосування науково обґрунтованих методів слід розглядати як найсуттєвішу умову отримання нових знань. Дослідник, який добре знає методи дослідження і можливості їх застосування, витрачає менше зусиль і працює успішніше, ніж той, хто у своєму дослідженні спирається лише на інтуїцію або діє за принципом «спроб і помилок». Звісно, що точні і правильні методи – не єдині компоненти, що забезпечують успішність наукового дослідження. Методи не можуть, наприклад, замінити творчу думку дослідника, його здібність аналізувати, робити висновки і передбачення. Але застосування правильних методів спрямовує хід думок дослідника, відкриває перед ним найкоротший шлях для досягнення мети і забезпечує таким чином можливість раціонально витрачати енергію і час науковця. Кожний метод наукового пізнання слід розглядати як систему регулятивних принципів практичної і теоретичної діяльності людини. Методів пізнання об'єктивної дійсності відомо дуже багато. Правильний вибір методів дослідження потребує знання їх класифікації.

Розрізняють методи теоретичних та емпіричних досліджень. Такий розподіл методів завжди умовний, оскільки з розвитком пізнання один науковий метод може переходити з однієї категорії в іншу.

Методи наукових досліджень теоретичного рівня. На теоретичному рівні дослідження одним з найпоширеніших наукових методів є абстрагування 26 (таблиця 2.1). В цілому дієслово «абстрагувати» означає відокремлювати в думках одні властивості, зв'язки та відношення явища або об'єкта від інших. Відповідно зміст даного методу полягає у відверненні від неістотних ознак предмета дослідження й одночасному виділенні тих, які мають найбільший інтерес. Використання методу абстрагування на початковому етапі дослідження дає змогу превентивно вирішити наукову проблему, акцентуючи увагу на її основних аспектах.

Таблиця 2.1 – Наукові методи теоретичних досліджень

Методи	Характеристики
Абстрагування	Уявне відвернення від неістотних, другорядних ознак явища або об'єкта, зв'язків і відношень між ними та виокремлення кількох сторін, які цікавлять дослідника
Аналіз	Метод пізнання, який дає змогу гіпотетично поділити явище або об'єкт на окремі складові частини та досліджувати їх відокремлено одна від одної
Синтез	Метод, який є результатом об'єднання окремих складових частин явища або об'єкта в єдине ціле, досліджуючи загальні ознаки і властивості
Моделювання	Опосередкований метод дослідження явища або об'єкта, який ґрунтується на застосуванні моделі як засобу дослідження, коли вивчити явище або об'єкт не можливо
Аксіоматичний метод	Метод побудови наукової теорії, за яким деякі твердження приймаються без доведень, а всі інші знання виводяться з них відповідно до певних логічних правил
Ідеалізація	Гіпотетичне конструювання явища або об'єкта, яких немає в дійсності або які за суб'єктивних причин практично нездійсненні

Методи	Характеристики
Формалізація	Метод вивчення явища або об'єкта через відображення їх структури в знаковій формі за допомогою штучних мов, наприклад, математики, фізики, хімії тощо
Дедукція	Умовивід, у якому висновок про той чи той елемент множини робиться на підставі знання про загальні властивості усієї множини
Індукція	Умовивід від часткового до загального, від окремих фактів до узагальнень, коли на основі знань про частини предметів певного класу робиться висновок про клас загалом

Вибір конкретних методів дослідження диктується характером фактичного матеріалу, умовами і метою конкретного дослідження.

В науковій сфері методи аналізу і синтезу зіставляються між собою.

Якщо вивчити явище або об'єкт складно та недоцільно, використовується метод моделювання. Незважаючи на те, що даний метод пов'язаний з ризиковістю, він дає змогу ґрунтовно дослідити закономірності виникнення й існування того чи іншого явища або об'єкта, реально оцінити їх взаємозалежність. При порівнянні альтернативних моделей кращою вважають ту з них, якій відповідає мінімальне значення середнього ризику. Величина похибки відображає залежність між предметом дослідження та впливом низки екзогенних чинників.

Емпіричні методи наукових досліджень

На емпіричному рівні використовується широкий спектр методів наукового дослідження, які можна об'єднати в наступну групу (таблиця 2.2).

Таблиця 2.2 – Емпіричні методи наукових досліджень

Методи	Характеристики
Вимірювання	Визначення кількісних характеристик явища або об'єкта за допомогою спеціальних технічних пристроїв та одиниць виміру

Методи	Характеристики
Порівняння	Процес зіставлення явища або об'єкта з метою з'ясування подібності та відмінності між ними, а також знаходження загального, що може бути властивим двом чи кільком явищам або об'єктам дослідження
Спостереження	Системне, цілеспрямоване, спеціально організоване візуальне сприймання явища або об'єкта, опосередковане раціональним знанням
Експеримент	Метод вивчення явища або об'єкта, коли дослідник активно та цілеспрямовано впливає на нього для створення штучних чи використання природних умов, необхідних для виявлення його ознак і властивостей

Найскладнішою є методика експериментальних досліджень, як лабораторних, так і польових.

Натомість на теоретичному рівні використовується пізнавальна діяльність, в якій структура реального експерименту відтворюється в уяві, тобто розумовий експеримент.

Модель (лат. *modus* — міра, франц. — *modele* — зразок) — штучно створений об'єкт у вигляді схеми, креслення, логіко-математичних знакових формул, фізичної конструкції й т.п., що аналогічний (подібний) досліджуваному об'єкту, процесу відображає й відтворює в більше простому, зменшеному виді структуру, властивості, взаємозв'язки й відносини між елементами досліджуваного об'єкта, процесу, безпосереднє вивчення якого зв'язано якими-небудь труднощами, більшими витратами коштів або просто недоступно, і тим самим полегшує процес одержання інформації про предмет, що цікавить нас, явищі, процеси

Пізніше були осмислені модельні властивості креслень, малюнків, карт — реальних об'єктів штучного походження, які дозволили втілити абстракцію більш високого рівня. Тобто наступний крок заключався в визнанні того, що моделями

можуть служити не тільки реальні об'єкти, а і абстрактні, ідеальні побудови, типовим прикладом яких є математичні моделі.

Коротко кажучи – Модель це системне відображення оригіналу.

Термін "модель" використовується в різних областях діяльності людини і як правило означає спрощену картину реальності або прообраз майбутнього.

У дослідженнях модель заміняє оригінал так, що її вивчення дає нові знання про об'єкт дослідження. Термін "модель" в науці являє собою штучно створену систему, яка відображає або здатна відтворити основні сторони реальної системи, яка називається оригіналом. Моделювання – це дослідження оригіналу не безпосередньо, а непрямым шляхом з допомогою штучно побудованої системи, яка називається моделлю. Моделювання включає процес побудови, вивчення та примінення моделей.

Виникає запитання : для чого необхідне моделювання , у яких випадках необхідно розробляти модель, а коли проводити пряме дослідження оригіналу? При цьому слід відмітити, що оригінал може бути недоступним для безпосереднього дослідження з причини відсутності часу і засобів, або відсутності самого оригіналу (розробка проекту, тощо). Крім того, модель завжди простіша оригіналу і дослідження на ній обходиться дешевше і потребує менше часу. Модель дозволяє оперативно міняти умови досліду (що не завжди можливо із оригіналом), дозволяє багаторазово повторити експеримент до пізнання суті.

При цьому не передбачається повна ідентичність оригіналу оскільки в такому разі втрачаються переваги моделі і всяка суть моделювання, та і вона перестає бути моделлю, а це вже буде копія-двійник, копія-дублер. Модель повинна замінити оригінал лише в принципі, по суті.

В моделі оригінал можна уявити в чистому вигляді без сторонніх впливів і непотрібних деталей.

Серед відомих методів дослідження складних електромеханічних систем можна виділити фізичне, імітаційне, аналогове та математичне моделювання.

Метод фізичного моделювання передбачає дослідження на моделі, яка відтворює основні геометричні, фізичні, динамічні та функціональні

характеристики та властивості оригіналу (але не обов'язково всі властивості) та має великі можливості з визначення різноманітних якостей електромеханічних систем ті їх елементів. Він базується на трьох теоремах аналізу подібності.

Практично необмежені можливості з виявлення ефективного функціонування складної електромеханічної системи має метод математичного моделювання. Він ґрунтується на здатності математично описувати рівняннями різні за природою явища (процеси) та виявляти різні функціональні зв'язки між елементами системи.

Аналогове моделювання базується на однаковому для моделі та натури математичному описі та використовується для імітації на основі аналогової фізичної системи по її елементам. При цьому кожному з фізичних елементів натури в моделі відповідає певний еквівалент. У якості моделей використовуються електричні кола, складені з пасивних та активних елементів.

Імітаційне моделювання використовується частіше за всього при дослідженні складних електромеханічних систем та представляють собою процес формування моделі реальної системи та проведення над нею численних експериментів з метою виявлення шляхів її створення, вдосконалення та (або) ефективного використання.

При математичному моделюванні застосовуються аналітичні та статистичні моделі. Аналітичні моделі більш грубі, враховують менше число факторів, завжди потребують деяких припущень та спрощень. Проте вони більше пристосовані для пошуку оптимальних рішень.

Статистичні моделі у порівнянні з аналітичними більш точні, але громіздкі та погано доступні для огляду.

При математичному моделюванні взаємозв'язки і закономірності оригіналу відображаються в математичній формі.

Моделювання починається із всестороннього вивчення об'єкту дослідження і постановки задач (проблеми). Проблема і задачі повинні бути чітко сформульовані. Далі слід знайти в реальній формі чи розробити модель оригіналу, яка б відображала чи могла відобразити основні сторони оригіналу.

По розробленій моделі проводять власне експеримент. Змінюють в певних межах вхідні параметри і досліджують вихідні. В результаті таких експериментів перевіряють модель на адекватність відображення оригіналу, отримують нові дані про оригінал, визначають числові параметри позитивних і негативних змін.

Комп'ютерна модель (або чисельна) – це комп'ютерна програма, яка реалізує абстрактну модель деякої системи. Комп'ютерні моделі використовуються для отримання нових знань про об'єкт, що моделюється або для приблизної оцінки поведінки математичних систем, які досить складні для аналітичного дослідження. Комп'ютерне моделювання є одним з ефективних методів вивчення складних систем. Комп'ютерні моделі простіші та їх зручніше досліджувати в силу можливості проводити обчислювальні експерименти в тих випадках, коли реальні експерименти немає можливості проводити через фінансові труднощі або фізичні перешкоди. Для підтримки математичного моделювання розроблені системи комп'ютерної математики, наприклад, *Maple*, *Mathematica*, *Mathcad*, *MATLAB*, *VisSim* та інші. В даній роботі використовується програма *Simulink* пакету *MATLAB*.

2.2 Структурно-логічна схема магістерського дослідження

Методологія - це система базисних принципів, методів, методик, способів і засобів їх реалізації в організації та побудові науково-практичної діяльності. Вона включає в себе методи теоретичного та емпіричного пізнання і являє собою алгоритм пошуку і досягнення мети.

Основне призначення методики дослідження полягає у тому, щоб на основі відповідних принципів (вимог, умов, обмежень тощо) забезпечити успішне вирішення визначених мети і завдань наукового дослідження.

Структурно-логічна схема є однією з форм планування роботи і візуалізації процесу, яка дозволяє з одного боку описувати процедуру (алгоритм) дослідження і досягнення мети тобто сформулювати робочу гіпотезу дослідження, а з другого – установити логічну послідовність виконання структурних розділів і підрозділів магістерської роботи.

Спочатку вибирається тема дослідження та обґрунтовується її актуальність та складається індивідуальний календарний план роботи.

Ведеться пошук літератури за темою, її аналіз, складання бібліографії.

Розробляється перший варіант теоретичної частини:

- постановка мети і конкретних завдань дослідження;
- визначення об'єкту й предмета дослідження;
- вибір методів (методики) проведення дослідження.

Виконується опис процесу дослідження згідно з обраним методом. Моделювання процесів у електроприводі гвинтового конвеєра проводиться шляхом математичного моделювання з використанням пакету Simulink програми MATLAB.

Наводиться інтерпретація отриманих результатів дослідження.

На підставі виконаних розрахунків та отриманих характеристик проводиться узагальнення отриманих теоретичних результатів.

На заключному етапі роботи проводиться розробка заходів охорони праці, впровадження результатів магістерського дослідження у навчальний процес.

2.3 Вибір методу вирішення поставленої задачі дослідження

Дана магістерська робота не передбачає проведення експериментальних досліджень, має теоретичний характер, тому основним методом для відображення отриманих результатів є моделювання.

Моделювання як метод наукового пізнання виникло в зв'язку з необхідністю вирішення завдань, які з тих чи інших причин не можуть бути вирішені безпосередньо. Вони виникають у випадках, коли об'єкт або недосяжний для дослідника, або він ще не існує і потрібно обрати оптимальний варіант його створення, або дослідження реального об'єкта вимагає багато часу та економічно не вигідне тощо. При моделюванні використовується принцип аналогії, ґрунтуючись на якому в процесі наукового дослідження висуваються гіпотези, тобто передбачення, що будуються на невеликій кількості дослідних даних,

спостережень, інтуїтивних припущень, перевірка правильності яких здійснюється шляхом експерименту.

Модель будується для того, щоб відобразити характеристики об'єкта (елементи, взаємозв'язки, структурні та функціональні властивості), суттєві з точки зору мети дослідження. Отже, моделювання пов'язане зі спрощенням, огрубленням прототипу, абстрагуванням від ряду його властивостей, ознак, сторін.

Процес дослідження реальних систем, що охоплює побудову моделі, дослідження її властивостей і перенесення одержаних відомостей на реальну систему, називають моделюванням.

Моделі поділяються на такі види:

- інтуїтивні (виражені на папері);
- фізичні (макети, муляжі, що дозволяють наочно ілюструвати, саме ті процеси, які мають місце у природі. За допомогою таких моделей можна вивчати вплив окремих параметрів на певний фізичний процес);
- математичні (дозволяють кількісно досліджувати явища, які важко піддаються вивченню за допомогою фізичних моделей);
- знакові (вираженні мовою, алгоритмами, графічно, математично);
- функціональні (описують залежності між входом і виходом системи);
- структурні (описують оргструктури систем).

Послідовність робіт при моделюванні виробничої системи:

- дослідження модельованої системи і постановка задачі - вивчення літературних джерел, проведення досліджень, тощо;
- формалізація задачі - виявлення особливостей системи, виділення головного, основного. На цьому етапі визначають, які параметри системи, яку ми моделюємо необхідно відобразити в моделі;
- розробка математичної моделі задачі і її запис в структурній формі.

Розробити математичну модель задачі це означає побудувати спеціальну таблицю чи схему, у якій усі економічні, технологічні та інші виробничі умови і вимоги виражені у вигляді нерівностей і рівнянь і об'єднані цільовою функцією. Математична модель може бути подана у структурній, загальній символній чи

числовій формі. Послідовність розробки моделі може бути різною, але зручніше спочатку скласти систему змінних величин, записати систему обмежень, а потім цільову функцію. В моделі виділяють змінні величини, константи і коефіцієнти.

Змінні в моделі відображають невідомі величини:

- аналіз кількісних залежностей параметрів задачі;
- збір інформації і її обробка;
- побудова числової моделі;
- вибір математичного методу вирішення задачі;
- вирішення задачі на ЕОМ;
- аналіз результатів вирішення і коригування моделі;
- вирішення задачі на ЕОМ по скоригованій моделі.

Головними рівнями дослідження та моделювання систем є мікро- та макрорівень. Мікрорівневе моделювання системи пов'язане з детальним описом кожного компонента системи, дослідженням її структури, функцій, взаємозв'язків, тощо. Практична реалізація найважливішого етапу мікромодельювання – виявлення елементів системи та взаємозв'язків між ними – пов'язана з необхідністю подолання суперечності між бажанням повного дослідження кожної з підсистем та елементів системи, реальною можливістю дослідити при цьому структуру системи загалом і принципи її функціонування.

Макрорівневе моделювання полягає в ігноруванні детальної структури системи та вивченні лише загальної поведінки системи як єдиного цілого. У процесі дослідження реальних систем і побудови їх моделей використовуються різні методи моделювання, що залежать від характеристик об'єкта, рівня знань про нього, мети дослідження та вимог до моделі.

Аксіоматичне моделювання полягає у відповідній інтерпретації та переведенні змістовного опису системи на мову чітких математичних термінів і відношень, у процесі чого усуваються неясності, суперечності, неповнота або надлишковість, які властиві вербальному описові системи.

Емпірико-статистичне моделювання використовує широко відомий кібернетичний принцип “чорної скрині”, що не дозволяє отримати модуль

структури системи, причинно-наслідкових зв'язків і механізмів її функціонування. В результаті моделювання отримують моделі типу “вхід - вихід”, які базуються на теоретичних гіпотезах про форми взаємозв'язку між входами і виходами системи.

Оптимізаційне моделювання передбачає включення у модель як взаємозв'язків між змінними та параметрами, так і критерії якості функціонування системи. Імітаційні моделі складних систем надзвичайно поширені внаслідок своєї універсальності, можливості проведення чисельних експериментів, передбачення різноманітних змін.

За мірою повноти опису моделювання поділяють на повне, неповне та наближене. Повне моделювання передбачає побудову моделі, адекватної об'єкту дослідження у просторі та часі. Для неповного моделювання ця адекватність не зберігається. При наближеному моделюванні беруться до уваги лише найважливіші аспекти системи.

Дослідження математичної моделі дає змогу отримати характеристики реального об'єкта чи системи. Вигляд математичної моделі залежить як від природи системи, так і від завдань дослідження.

Математична модель системи містить, як правило, опис множини можливих станів системи та закон переходу з одного стану в інший. Математичне моделювання охоплює імітаційне, інформаційне, структурне, ситуаційне тощо.

Розрахунок задач магістерського дослідження проводиться за допомогою електронної обчислювальної техніки за допомогою програми Mathcad. На базі складених диференціальних рівнянь проводиться математичне моделювання системи керування електропривода. За допомогою програми Simulink досліджуються перехідні характеристики системи при зміні задаючих параметрів.

Слід зазначити, що метод моделювання – вивчення явищ за допомогою моделей – сьогодні має особливе значення для вирішення складних фізичних завдань, чому значною мірою сприяє зростанню продуктивності комп'ютерної техніки. Саме тому в цьому магістерському дослідженні як основа використаний метод моделювання.

РОЗДІЛ 3

ВИБІР МІКРОКОНТРОЛЕРА ТА ОБҐРУНТУВАННЯ ЙОГО АРХІТЕКТУРИ ДЛЯ ПЛК

3.1 Вибір мікроконтролера

Сучасна автоматизація промислових об'єктів базується на використанні вбудованих систем керування (Embedded Systems), основним обчислювальним ядром яких є мікроконтролер. Мікроконтролер — це спеціалізована мікропроцесорна система, реалізована на одному кристалі, що поєднує високопродуктивний процесор з необхідним набором периферійних пристроїв для взаємодії з фізичними об'єктами.

На відміну від мікропроцесорів загального призначення, що використовуються в персональних комп'ютерах і орієнтовані на високу обчислювальну потужність та взаємодію з користувачем, архітектура мікроконтролер оптимізована для задач керування в реальному часі. Головними критеріями тут виступають надійність, енергоефективність, реактивність (швидкість реакції на зовнішні події) та наявність розвинених засобів апаратного інтерфейсу.

Архітектурні особливості сучасних мікроконтролерів

Більшість сучасних 8-ми та 32-розрядних мікроконтролерів (зокрема сімейства AVR, PIC, STM8) побудовані на базі Гарвардської архітектури. Її фундаментальна відмінність від класичної архітектури Фон-Неймана полягає у фізичному розділенні просторів пам'яті команд та пам'яті даних, а також використанні окремих шин для доступу до них (рис. 3.1).

Таке розділення дозволяє реалізувати конвеєрну обробку даних: у той час, поки центральний процесор (CPU) виконує поточну інструкцію, блок вибірки вже зчитує наступну команду з пам'яті програм. Це суттєво підвищує швидкодію системи, дозволяючи виконувати більшість інструкцій за один машинний цикл.

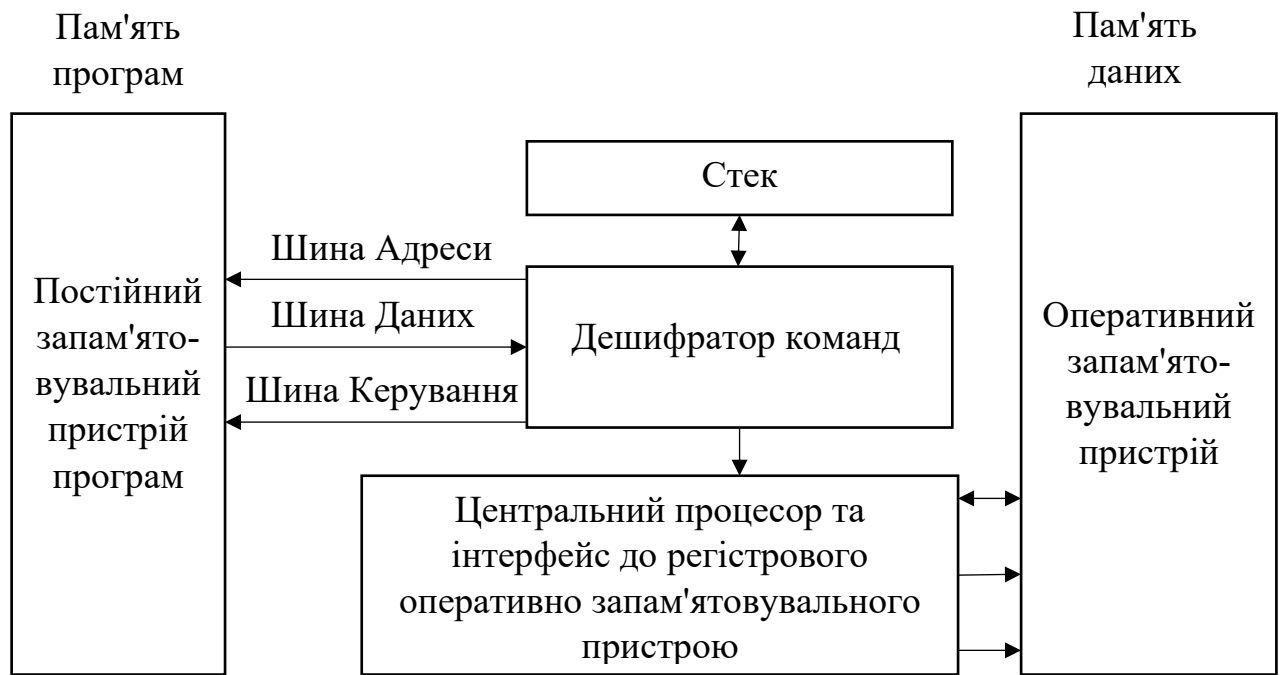


Рисунок 3.1. Структурна схема Гарвардської архітектури обчислювального ядра

До складу процесорного ядра входять:

- Арифметико-логічний пристрій: Виконує математичні та логічні операції над даними.
- Регістри загального призначення: Використовуються як надоперативна пам'ять для тимчасового зберігання операндів.
- Лічильник команд (Program Counter): Вказує адресу поточної інструкції.
- Регістр статусу (Status Register): Містить прапорці результатів операцій (переповнення, нульовий результат, перенесення тощо), які впливають на логіку розгалуження програми.

Організація пам'яті даних та регістрова архітектура мікроконтролера

Ефективність функціонування мікропроцесорної системи керування значною мірою залежить від архітектури підсистеми пам'яті. У сучасних 8-розрядних RISC-мікроконтролерах (наприклад, сімейства PIC) пам'ять даних (SRAM) реалізована як єдиний адресний простір регістрів, які функціонально поділяються на дві групи:

- Регістри спеціальних функцій (SFR — Special Function Registers): Керують роботою ядра процесора та периферійних модулів.
- Регістри загального призначення (GPR — General Purpose Registers): Використовуються для зберігання змінних та проміжних результатів обчислень.

Банкова структура оперативної пам'яті

Для оптимізації адресації обмеженого адресного простору 8-розрядних систем використовується сегментація пам'яті на банки. Типова структура організації пам'яті даних з поділом на два банки представлена на рис.3.2

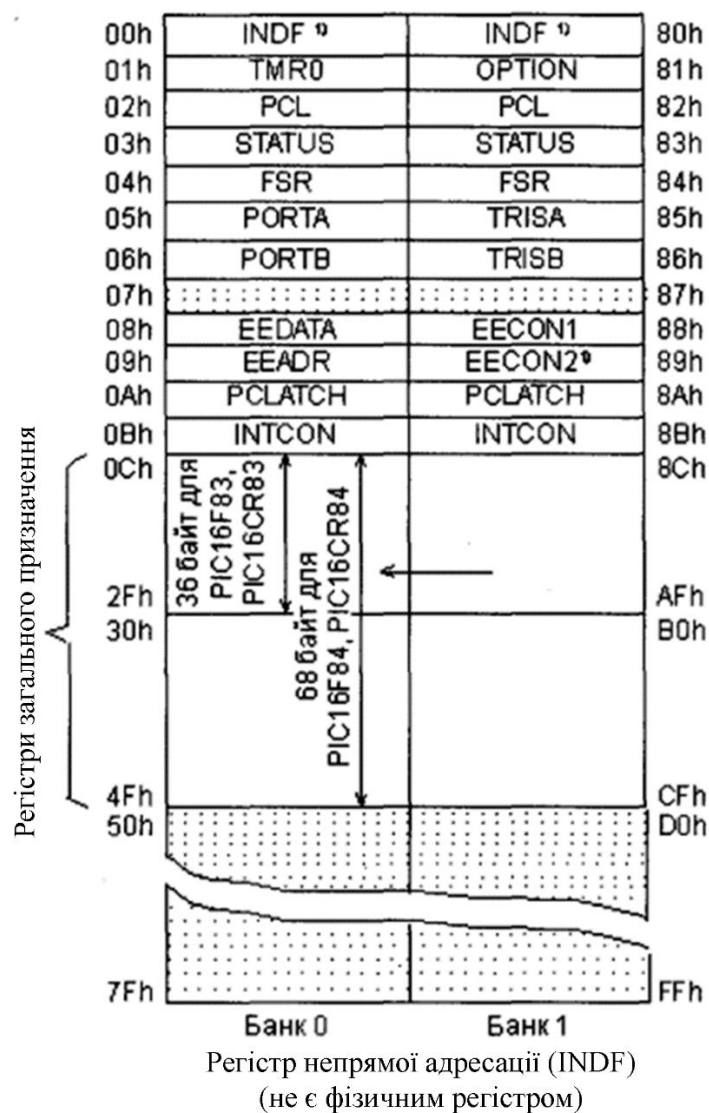


Рисунок 3.2. Структура організації пам'яті даних з розбиттям на банки

Молодші адреси кожного банку відведені під регістри спеціальних функцій. Деякі критично важливі регістри (наприклад, STATUS, INTCON) продубльовані в усіх банках для забезпечення швидкого доступу незалежно від поточної конфігурації. Інші регістри доступні лише в конкретному банку (наприклад, регістр напрямку порту TRISB зазвичай знаходиться у Банку 1, тоді як регістр даних PORTB — у Банку 0). Перемикання між банками здійснюється програмно шляхом встановлення відповідних бітів у регістрі статусу.

Старші адреси відведені під регістри загального призначення (GPR). Їх кількість варіюється залежно від моделі мікроконтролера і може становити від десятків байт до кількох кілобайт.

Організація портів введення-виведення. Взаємодія мікроконтролера із зовнішнім середовищем здійснюється через двонаправлені порти введення-виведення (I/O Ports). Кожен порт (Port A, Port B тощо) має відповідний регістр даних (наприклад, PORTA) та регістр керування напрямком передачі даних (наприклад, TRISA).

Алгоритм налаштування порту:

- Вибір напрямку: Запис логічної "1" у відповідний біт регістра TRIS переводить пін у режим високоімпедансного входу (Input), запис "0" — у режим виходу (Output).
- Читання/Запис: Зчитування стану входу виконується зверненням до регістра PORT, запис у вихідну заціпку також здійснюється через цей регістр.

Для узгодження електричних рівнів із зовнішніми датчиками (наприклад, кнопками або герконами) входи можуть програмно підключатися до внутрішніх підтягуючих резисторів (Weak Pull-up), що спрощує схемотехніку пристрою, усуваючи необхідність у зовнішніх дискретних компонентах.

Система тактування та синхронізації. Робота всіх вузлів мікроконтролера суворо синхронізована тактовими імпульсами. Продуктивність контролера прямо пропорційна тактовій частоті F_{osc} . Джерелом тактових імпульсів виступає осцилятор, який може бути реалізований у кількох варіантах (рис. 3.3).

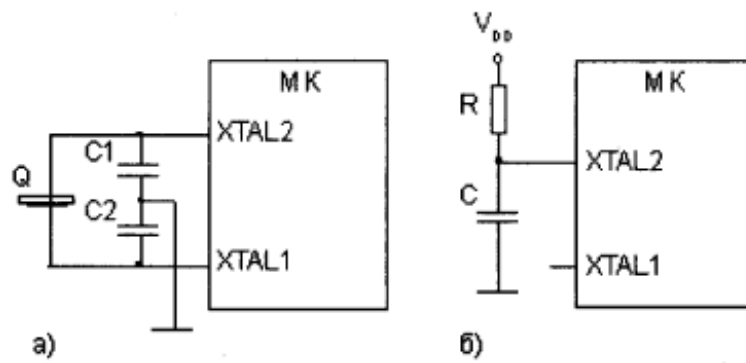


Рисунок 3.3. Схеми підключення тактових генераторів: а) зовнішній кварцовий резонатор; б) RC-ланцюг

Для задач точного керування часовими інтервалами (наприклад, формування сигналів широко імпульсного модулятора або реалізації протоколів зв'язку UART) перевага надається зовнішнім кварцовим резонаторам, стабільність частоти яких становить $10^{-5} \dots 10^{-6}$. У бюджетних рішеннях, де висока точність не є критичною, застосовуються внутрішні RC-генератори.

Периферійні пристрої та інтерфейси. Функціональність мікроконтролера визначається набором інтегрованих периферійних модулів, які дозволяють розвантажити центральний процесор.

Порти введення-виведення (GPIO). Це фізичні виводи мікроконтролера, які програмно налаштовуються як входи (для зчитування станів датчиків) або виходи (для керування силовими ключами, світлодіодами, реле). Кожен пін порту зазвичай має буферну схему, захисні діоди та можливість підключення внутрішніх "підтягуючих" резисторів (Pull-up).

Таймери та лічильники. Апаратні модулі, що працюють паралельно з основним процесором. Вони використовуються для:

- Відліку точних часових інтервалів;
- Підрахунку зовнішніх подій (імпульсів від енкодерів);
- Формування сигналів широтно-імпульсної модуляції для плавного керування швидкістю електродвигунів.

Система переривань (Interrupts). Найважливіший механізм для систем реального часу. Переривання дозволяють мікроконтролеру миттєво реагувати на

асинхронні події (наприклад, спрацювання кінцевого вимикача робота). При виникненні події виконання основної програми призупиняється, контролер зберігає контекст і переходить до виконання підпрограми обробки переривання (ISR), після чого повертається до основної задачі.

Аналого-цифровий перетворювач. Оскільки більшість фізичних величин (температура, тиск, позиція потенціометра) є аналоговими, для їх обробки використовується аналого-цифровий перетворювач. Він перетворює вхідну напругу у цифровий код. Розрядність аналого-цифрового перетворювача (зазвичай 10 або 12 біт) визначає точність вимірювання. Наприклад, для 10-бітного аналого-цифрового перетворювача діапазон вхідної напруги розбивається на $2^{10} = 1024$ дискретних рівні.

Комунікаційні інтерфейси. Для обміну даними з іншими контролерами, персональний компютер або інтелектуальними датчиками використовуються стандартні промислові інтерфейси:

- USART (RS-232/RS-485): Асинхронний послідовний інтерфейс, стандарт де-факто для промислових мереж.
- SPI/I2C: Синхронні інтерфейси для підключення внутрішньосистемної периферії (пам'яті, драйверів дисплеїв).

Вибір конкретної архітектури та моделі мікроконтролера для модернізації робота МП-9С повинен базуватися на аналізі необхідної кількості ліній вводу-виводу, вимог до швидкодії та надійності системи.

3.2 Технічні характеристики мікроконтролера PIC16F1937

Для реалізації системи керування обрано мікроконтролер PIC16F1937 (рис 3.4.), що базується на покращеному 8-розрядному ядрі середнього рівня (Enhanced Mid-range CPU). Дана архітектура оптимізована для створення високонадійних вбудованих систем.

Набір команд до 49 інструкцій. Це включає оптимізації для компіляторів мови C, що дозволяє зменшити обсяг генерованого коду на 30-40% порівняно зі

старими ядрами. Всі інструкції виконуються за один цикл (4 такти генератора), крім переходів, які займають два цикли, яка забезпечується наступними апаратними рішеннями:

- Апаратний стек (Hardware Stack): Використовується стек глибиною 16 рівнів. Контролер оснащений механізмом захисту: при переповненні стека (*Stack Overflow*) або виході за його межі (*Underflow*) автоматично генерується сигнал скидання (Reset). Це запобігає непередбачуваній поведінці маніпулятора у випадку програмних збоїв.

- Автоматичне збереження контексту: При виникненні переривань основні регістри автоматично зберігаються в тіньових регістрах (*Shadow Registers*), що зменшує час реакції системи на події від датчиків.

Для корпусів QFN критично важливо спроектувати "Thermal Land" на платі. Це мідний квадрат під мікросхемою, з'єднаний з полігоном землі (Ground Plane) через масив перехідних отворів (thermal vias). Це забезпечує механічну міцність і відведення тепла. Недотримання цієї вимоги може призвести до перегріву або механічного відриву чіпа при вібрації.

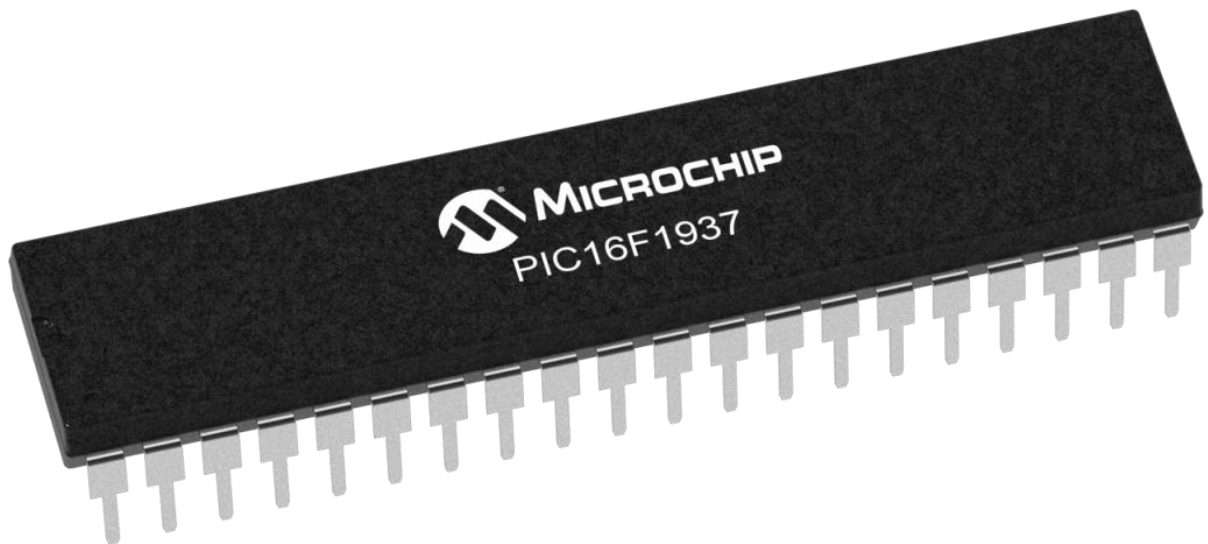


Рисунок 3.4 Мікроконтролер PIC16F1937

Розподіл портів введення-виведення. Вибір контролера у 40-вивідному корпусі (PDIP-40 рис.3.5) надав розширені можливості для проєктування схеми без використання додаткових регістрів розширення. На основі аналізу функціонального призначення виводів, було додан опис ключових описів функцій пінів для автоматизації робота МП-9С:

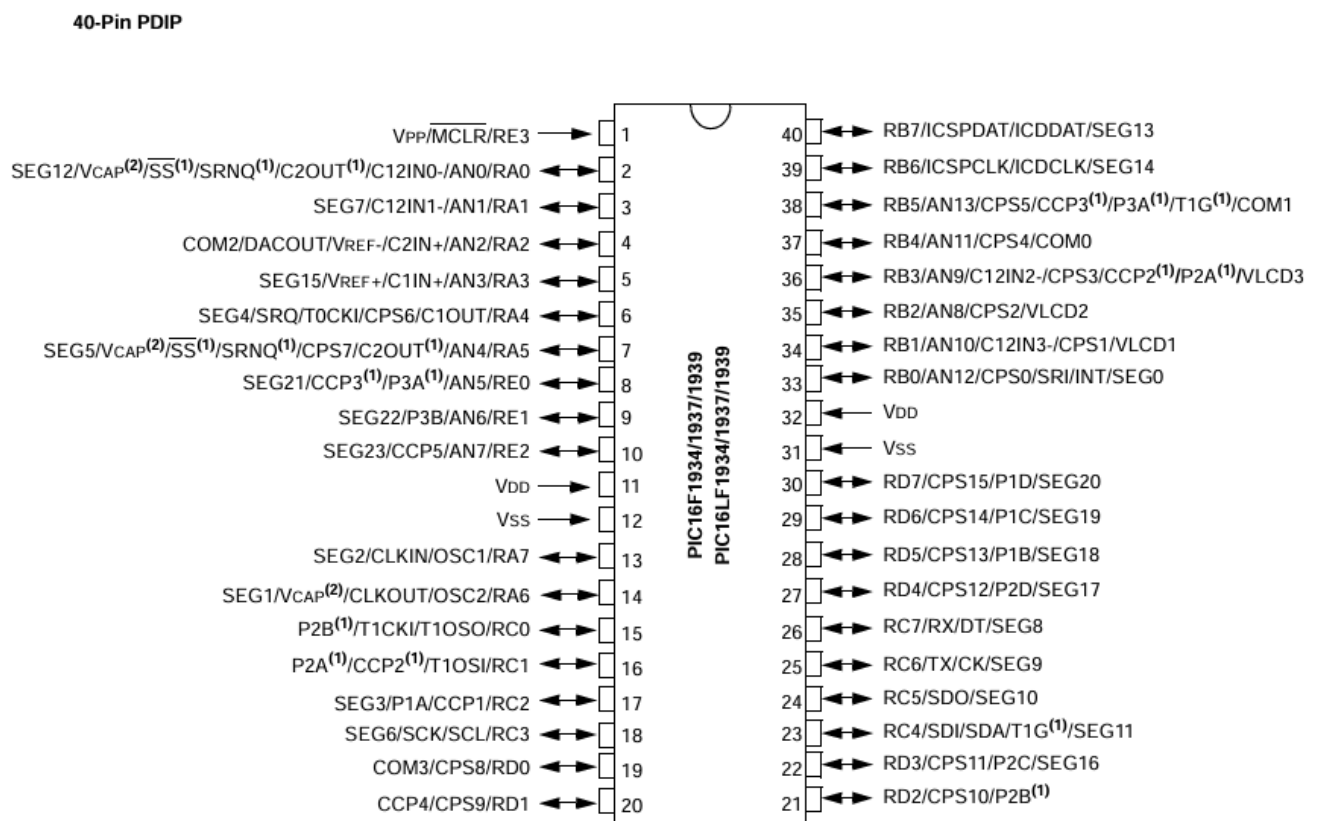


Рисунок 3.5 Призначення виводів мікроконтролера PIC16F1937

Підсистема живлення реалізована за допомогою VDD и VSS (таб.3.1.):

- VDD (Позитивне живлення): представляю собою вхід для подачі основної напруги на мікроконтролер. Для серії PIC16F1937 цей вхід є досить гнучким у якого робочий діапазон: від 1.8В до 5.5В, що дозволяє жити пристрій від різних джерел: 5В (USB), 3.3В (стандартна логіка) або навіть від батарейок (3В).
- VSS (Опорна земля): Це загальний вивід ("мінус"), відносно якого вимірюються всі інші рівні напруги в схемі. Всі виводи VSS на корпусі мікросхеми повинні бути з'єднані із загальною шиною заземлення плати.

Таблиця 3.1. Опис виводів живлення

Назва	Функція	Тип введення	Тип виводу	Опис
VDD	VDD	Живлення	—	Позитивне джерело живлення
VSS	VSS	Живлення	—	Опорна земля (GND).

Підсистема скидання мікроконтролера реалізована за допомогою комбінованого виводу MCLR (таб. 3.2.).

MCLR (Master Clear): Представляє собою вхід зовнішнього апаратного скидання. Це активний низький рівень (Active Low), що означає, що для перезавантаження мікроконтролера на цей вхід необхідно подати логічний "0" (GND). Вхід оснащений схемою Тригера Шмітта (ТШ) для захисту від випадкових спрацювань через завади та має вбудований підтягувальний резистор (Internal Pull-up), що спрощує схему підключення.

Таблиця 3.2. Опис виводу скидання (RESET)

Назва	Функція	Тип введення	Тип виводу	Опис
RE3/ MCLR/ VPP	RE3	ТТЛ	—	Вхід загального призначення (тільки вхід).
	MCLR	ТШ	—	Майстер-скидання (Master Clear) із внутрішньою підтяжкою.
	VPP	ВН	—	Напруга програмування (Висока напруга).

VPP (Programming Voltage): Цей же вивід використовується для входу в режим програмування. Для цього на нього подається висока напруга (близько 8-9В), що переводить контролер у режим запису пам'яті (Flash/EEPROM).

RE3: У разі, якщо функція зовнішнього скидання відключена (через біти конфігурації), цей пін може функціонувати як звичайний цифровий вхід RE3.

Підсистема тактування та стабілізації напруги ядра реалізована за допомогою виводів OSC1/CLKIN, OSC2/CLKOUT та VCAP (таб. 3.3).

Таблиця 3.3. Опис виводів тактування та стабілізації

Назва	Функція	Тип входу	Тип виходу	Опис
RA6/ OSC2/ CLKOUT/ VCAP/ SEG1	RA6	ТТЛ	КМОН	Порт введення/виведення загального призначення.
	OSC2	—	XTAL	Вихід для підключення Кварцу/Резонатора (режими LP, XT, HS).
	CLKOUT	—	КМОН	Вихід тактової частоти FOSC/4.
	VCAP	Живлення	—	Конденсатор фільтра для внутрішнього стабілізатора напруги.
	SEG1	—	АН	Аналоговий вихід для РК-дисплея.
RA7/ OSC1/ CLKIN/ SEG2	RA7	ТТЛ	КМОН	Порт введення/виведення загального призначення.
	OSC1	XTAL	—	Вхід для Кварцу/Резонатора (режими LP, XT, HS).
	CLKIN	КМОН	—	Вхід зовнішнього тактового сигналу (режим EC).
	SEG2	—	АН	Аналоговий вихід для РК-дисплея.

- VCAP (Voltage Regulator Capacitor): Представляє собою вхід для підключення зовнішнього фільтруючого конденсатора для внутрішнього LDO-стабілізатора напруги. Оскільки ядро PIC16F1937 працює на нижчій напрузі, ніж периферія, цей вивід є критичним для стабільності роботи центрального процесора.

Відсутність конденсатора призведе до нестабільної роботи або постійного перезавантаження (Reset).

- OSC1 (Oscillator Input) / CLKIN: Представляє собою вхід підсилювача інвертора генератора. У режимах з використанням кварцового резонатора сюди підключається одна з ніжок кварцу. У режимі зовнішнього тактування (EC) сюди подається готовий тактовий сигнал рівня комплементарно структурного метало-оксидного-напівпровідника.

- OSC2 (Oscillator Output) / CLKOUT: Представляє собою вихід підсилювача інвертора генератора. Використовується для замикання контуру зворотного зв'язку кварцового резонатора. Якщо налаштовано використання внутрішнього генератора, цей пін може виконувати функцію CLKOUT, виводячи частоту $F_{OSC}/4$, що корисно для синхронізації зовнішніх пристроїв або діагностики швидкодії системи.

- SEG1 / SEG2 (LCD Segment Outputs): Представляють собою спеціалізовані аналогові виходи (АН) вбудованого драйвера рідкокристалічного дисплея. Мікроконтролер генерує на цих виводах напругу складної форми (multiplexed waveforms) для безпосереднього керування сегментами "голого" РК-скла, що дозволяє реалізувати екран без використання зовнішніх драйверів.

Підсистема внутрішньосхемного програмування та налагодження реалізована за допомогою виводів ICSPCLK та ICSPDAT (таб. 3.4.), які спільно з піном VPP (ВН) утворюють інтерфейс ICSP™ (In-Circuit Serial Programming).

- ICSPCLK (In-Circuit Serial Programming Clock): Представляє собою вхід тактування для процесу програмування. Це лінія синхронізації, по якій програматор (наприклад, PICkit) передає тактові імпульси для запису даних у Flash-пам'ять контролера. Вхід використовує тригер Шмітта (ТШ) для забезпечення надійності сигналу.

Таблиця 3.4. Опис виводів програмування (ICSP)

Назва	Функція	Тип входу	Тип виходу	Опис
RB6/ ICSPCL/ ICDCLK/ SEG14	RB6	ТТЛ	КМОН	Порт введення/виведення загального призначення.
	ICSPCLK	ТШ	—	Вхід тактування послідовного програмування.
	ICDCLK	ТШ	—	Вхід тактування внутрішньосхемного налагодження.
	SEG14	—	АН	Аналоговий вихід для РК-дисплея.
RB7/ ICSPDAT /ICDDAT/ SEG13	RB7	ТТЛ	КМОН	Порт введення/виведення загального призначення.
	ICSPDAT	ТШ	КМОН	Двонаправлена лінія даних ICSP™.
	ICDDAT	ТШ	КМОН	Двонаправлена лінія даних налагодження.
	SEG13	—	АН	Аналоговий вихід для РК-дисплея.

- ICSPDAT (In-Circuit Serial Programming Data): Представляє собою двонаправлену лінію даних. Через цей пін завантажується код програми в пам'ять мікроконтролера, а також зчитується його вміст при верифікації. Під час налагодження (Debug) через цю лінію передаються дані про стан регістрів та точок зупинки.

Підсистема цифрового зв'язку реалізована за допомогою виводів, що належать до порту C (таб. 3.5).

- SCL (Serial Clock) та SDA (Serial Data): Представляють собою лінії синхронізації та даних для інтерфейсу I2C™. Особливістю цих виводів є вихід типу "Відкритий Стік" (BC). Це означає, що мікроконтролер може лише притягувати лінію до землі (GND), але не може видавати високий рівень. Для роботи цього інтерфейсу обов'язкове використання зовнішніх підтягувальних

резисторів (Pull-up) до шини живлення. Входи мають спеціалізовані рівні спрацювання (I2C) для відповідності стандарту протоколу.

- SCK, SDI, SDO: Представляють собою повний набір ліній для інтерфейсу SPI (Serial Peripheral Interface). SCK — лінія тактування, SDI (Serial Data In) — вхід даних від підлеглого пристрою, а SDO (Serial Data Out) — вихід даних. Цей інтерфейс працює на високих швидкостях і використовує стандартні рівні КМОН.

- TX (Transmit) та RX (Receive): Представляють собою лінії передачі та прийому модуля EUSART (універсальний синхронний/асинхронний прийомопередавач). TX — це цифровий вихід, який передає послідовні дані до іншого пристрою (наприклад, ПК або GSM-модуля), а RX — це цифровий вхід з тригером Шмітта (ТШ), що приймає дані. Цей інтерфейс є основою для реалізації протоколів RS-232, RS-485 та LIN.

Таблиця 3.5. Опис комунікаційних виводів (MSSP та EUSART)

Назва	Функція	Тип входу	Тип виходу	Опис
RC3/ SCK/ SCL/ SEG6	RC3	ТШ	КМОН	Порт введення/виведення загального призначення.
	SCK	ТШ	КМОН	Тактовий сигнал інтерфейсу SPI.
	SCL	I2C	BC	Тактовий сигнал інтерфейсу I2C™.
	SEG6	—	АН	Аналоговий вихід для РК-дисплея.
RC4/ SDI/ SDA/ T1G/ SEG11	RC4	ТШ	КМОН	Порт введення/виведення загального призначення.
	SDI	ТШ	—	Вхід даних інтерфейсу SPI.
	SDA	I2C	BC	Вхід/вихід даних інтерфейсу I2C™.
	T1G	ТШ	—	Вхід керування (Gate) Таймера 1.
	SEG11	—	АН	Аналоговий вихід для РК-дисплея.

продовження таблиці 3.5.

Назва	Функція	Тип входу	Тип виходу	Опис
RC5/ SDO/ SEG10	RC5	ТШ	КМОН	Порт введення/виведення загального призначення.
	SDO	—	КМОН	Вихід даних інтерфейсу SPI.
	SEG10	—	АН	Аналоговий вихід для РК-дисплея.
RC6/ TX/ CK/ CCP3/ P3A/ SEG9	RC6	ТШ	КМОН	Порт введення/виведення загального призначення.
	TX	—	КМОН	Вихід передавача асинхронного USART.
	CK	ТШ	КМОН	Тактовий сигнал синхронного USART.
	SEG9	—	АН	Аналоговий вихід для РК-дисплея.
RC7/ RX/ DT/ P3B/ SEG8	RC7	ТШ	КМОН	Порт введення/виведення загального призначення.
	RX	ТШ	—	Вхід приймача асинхронного USART.
	DT	ТШ	КМОН	Дані синхронного USART.
	SEG8	—	АН	Аналоговий вихід для РК-дисплея.

Підсистема обробки аналогових сигналів реалізована за допомогою групи виводів, позначених як ANx та VREF (таб. 3.6).

- ANx (Analog Inputs): Це входи багатоканального 10-бітного аналого-цифрового перетворювача (АЦП). Коли вивід налаштовано як аналоговий вхід (через регістр ANSEL), цифровий вхідний буфер відключається. Це дозволяє вимірювати напругу зовнішніх датчиків без спотворення сигналу та знижує споживання струму, виключаючи паразитні перемикання цифрової логіки.

Таблиця 3.6. Аналогові виходи та опорна напруга

Назва	Функція	Тип входу	Тип виходу	Опис
RA2/ AN2/ C2IN+/ VREF-/ DACOUT/ COM2	RA2	ТТЛ	КМОН	Порт введення/виведення загального призначення.
	AN2	АН	—	Вхід каналу 2 АЦП.
	VREF-	АН	—	Вхід негативної опорної напруги для АЦП.
	DACOUT	—	АН	Вихід джерела опорної напруги (ЦАП).
	C2IN+	АН	—	Позитивний вхід компаратора C2.
RA3/ AN3/ C1IN+/ VREF+/ COM3/ SEG15	RA3	ТТЛ	КМОН	Порт введення/виведення загального призначення.
	AN3	АН	—	Вхід каналу 3 АЦП.
	VREF+	АН	—	Вхід позитивної опорної напруги для АЦП.
	C1IN+	АН	—	Позитивний вхід компаратора C1.
	CPS0	АН	—	Вхід 0 ємнісного сенсора.

- VREF+ (Voltage Reference Positive): Вхід зовнішньої позитивної опорної напруги. Зазвичай АЦП використовує напругу живлення (VDD) як еталон (верхню межу вимірювання). Однак, якщо VDD нестабільна (наприклад, просідає при роботі двигунів) або потрібна висока точність вимірювання (наприклад, діапазон 0...2.048В), на цей пін подається стабільна зовнішня напруга. Це значно підвищує точність перетворення.

- VREF- (Voltage Reference Negative): Вхід зовнішньої негативної опорної напруги. Дозволяє змістити нижню межу вимірювання АЦП вище рівня землі (VSS). Наприклад, для вимірювання сигналу в діапазоні від 1В до 3В з максимальною роздільною здатністю.

- DACOUT (Digital-to-Analog Output): Вихід вбудованого 5-бітного цифро-аналогового перетворювача. Він може генерувати одну з 32 сходинок напруги, яка використовується як опорна для внутрішніх компараторів або виводиться назовні для керування іншими схемами.

Опис додаткових підсистем (Таймер 1, Розширений ШІМ, Сенсори)

Ця група виводів розкриває розширені можливості PIC16F1937, зокрема керування часом, двигунами та сенсорними інтерфейсами (таб. 3.7).

Генератор Таймера 1 (T1OSO / T1OSI): Виводи RC0 та RC1 виконують унікальну функцію — вони дозволяють підключити окремий низькочастотний кварцовий резонатор (зазвичай 32.768 кГц) безпосередньо до драйвера Timer1.

- Призначення: Це дозволяє реалізувати годинник реального часу (RTC), який продовжує працювати та рахувати час, навіть коли основний процесор знаходиться в режимі глибокого сну (Sleep) для економії батареї.

Розширений модуль ССР/PWM (PxA, PxB): На відміну від звичайних мікроконтролерів, де ШІМ (PWM) має лише один вихід, PIC16F1937 оснащений Enhanced ССР. Виводи з маркуванням P1A/P1B, P2A/P2B, P3A/P3B працюють у парі.

- P1A, P2A, P3A: Основні виходи ШІМ.
- P1B, P2B, P3B: Додаткові (комплементарні) виходи.

Застосування: Це дозволяє керувати Н-мостом (H-Bridge) для реверсивного керування двигунами постійного струму або напівмостовими перетворювачами напруги апаратно, без складного програмного коду.

Ємнісний сенсорний модуль (CPS): Вивід RD0/CPS8 є частиною модуля mTouch.

Призначення: Вхід ємності сенсорного модуля підключається до мідної площадки на платі, яка працює як сенсорна кнопка. Внутрішній генератор вимірює зміну ємності при наближенні пальця користувача, дозволяючи створювати герметичні клавіатури без механічних контактів.

Таблиця 3.7. Аналогові виходи та опорна напруга

Назва	Функція	Тип входу	Тип виходу	Опис
RC0/ T1OSO/ T1CKI/ P2B	RC0	ТШ	КМОН	Порт введення/виведення загального призначення.
	T1OSO	XTAL	XTAL	Вихід генератора Таймера 1 (підключення кварцу 32.768 кГц).
	T1CKI	ТШ	—	Вхід зовнішнього тактування для Таймера 1.
	P2B	—	КМОН	Вихід ШІМ (канал В модуля CCP2).
RC1/ T1OS/ CCP2/ P2A	RC1	ТШ	КМОН	Порт введення/виведення загального призначення.
	T1OSI	XTAL	XTAL	Вхід генератора Таймера 1.
	CCP2	ТШ	КМОН	Захоплення/Порівняння/ШІМ (Модуль 2).
	P2A	—	КМОН	Вихід ШІМ (канал А модуля CCP2).
RC2/ CCP1/ P1A/ SEG3	RC2	ТШ	КМОН	Порт введення/виведення загального призначення.
	CCP1	ТШ	КМОН	Захоплення/Порівняння/ШІМ (Модуль 1).
	P1A	—	КМОН	Вихід ШІМ (канал А модуля CCP1).
	SEG3	—	АН	Аналоговий вихід для РК-дисплея.
RC5/ SDO/ SEG10	RC5	ТШ	КМОН	Порт введення/виведення загального призначення.
	SDO	—	КМОН	Вихід даних інтерфейсу SPI.
	SEG10	—	АН	Аналоговий вихід для РК-дисплея.

продовження таблиці 3.7.

Назва	Функція	Тип входу	Тип виходу	Опис
RC7/ RX/ DT/ P3B/ SEG8	RC7	ТШ	КМОН	Порт введення/виведення загального призначення.
	RX	ТШ	—	Вхід приймача асинхронного USART.
	DT	ТШ	КМОН	Дані синхронного USART.
	P3B	—	КМОН	Вихід ШІМ (канал В модуля CCP3).
	SEG8	—	АН	Аналоговий вихід для РК-дисплея.
RD0/ CPS8/ COM3	RD0	ТШ	КМОН	Порт введення/виведення загального призначення.
	CPS8	АН	—	Вхід 8 ємнісного сенсора (Capacitive Sensing).
	COM3	—	АН	Аналоговий вихід загальної шини (Common) РК-дисплея.
RE0/ AN5/ P3A/ CCP3/ SEG21	RE0	ТШ	КМОН	Порт введення/виведення загального призначення.
	AN5	АН	—	Вхід каналу 5 АЦП.
	P3A	—	КМОН	Вихід ШІМ (канал А модуля CCP3).
	CCP3	ТШ	КМОН	Захоплення/Порівняння/ШІМ (Модуль 3).
	SEG21	—	АН	Аналоговий вихід для РК-дисплея.

Драйвер РК-дисплея (COMx): Особливу увагу слід звернути на вивід RD0/COM3.

- COMx vs SEGx: Якщо виводи SEG керують окремими сегментами, то виводи COM (Common) є спільними електродами (підкладкою) дисплея. Наявність COM3 свідчить про те, що даний мікроконтролер підтримує мультиплексування дисплея до 4-х рівнів (1/4 Duty), що дозволяє керувати великою кількістю сегментів (до 96) з мінімальною кількістю виводів.

Організація пам'яті програм та векторів переривань

Мікроконтролер PIC16F1937, побудований на базі покращеного ядра (Enhanced Mid-Range Core), використовує 15-розрядний лічильник команд (Program Counter), що теоретично дозволяє адресувати до 32К слів пам'яті.

Згідно зі специфікацією, для обраної моделі PIC16F1937 реалізовано наступну конфігурацію пам'яті:

- Обсяг пам'яті програм: 8192 слів (8К слів x 14 біт).
- Адресний простір: від 0000h до 1FFFh.
- Межі доступу: Звернення до комірок пам'яті за межами адреси 1FFFh

призведе до циклічного переходу в початок реалізованого простору (wrap-around), що необхідно враховувати при розробці програмного забезпечення.

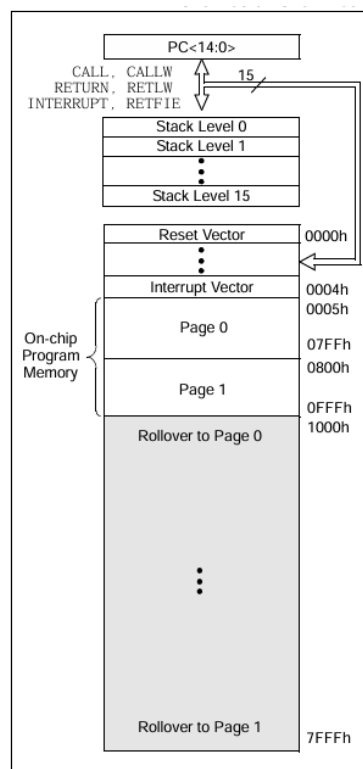


Рисунок 3.6. Карта пам'яті програм мікроконтролера PIC16F1937

Особливе значення для надійності системи керування мають апаратні вектори(рис.3.6):

Вектор скидання (Reset Vector): Розташований за адресою 0000h. Саме звідси починається виконання програми після подачі живлення або перезавантаження.

Вектор переривань (Interrupt Vector): Розташований за адресою 0004h. Сюди передається керування при виникненні будь-якого дозволеного переривання (від таймера, зміни стану порту тощо).

Архітектура та регістрова модель портів введення-виведення

Загальна кількість портів мікроконтролера залежить від корпусу та задіяної периферії. У роботі з портами використовується стандартизована логіка керування. Кожен порт (PORTA, PORTB, PORTC тощо) обслуговується трьома основними регістрами (рис 3.7.), що дозволяє уникнути класичної проблеми "читання-модифікація-запис" (Read-Modify-Write issue):

- TRISx (Data Direction Register): Регістр напрямку передачі даних.
- Встановлення біта в 1 налаштовує вивід як Вхід (Input, високоімпедансний стан).
- Скидання біта в 0 налаштовує вивід як Вихід (Output).
- PORTx (Input Register): Дозволяє зчитати фізичний логічний рівень на ніжці мікроконтролера.
- LATx (Output Latch): Регістр-защівка вихідних даних. Запис даних у цей регістр оновлює стан вихідних драйверів пінів. Використання LATx замість PORTx при записі підвищує стабільність роботи при керуванні ємнісними або індуктивними навантаженнями (наприклад, довгими кабелями до датчиків).

Оскільки маніпулятор МП-9С функціонує в умовах промислового виробництва, де можливі електромагнітні завади та перепади напруги живлення, при виборі мікроконтролера PIC16F1937 враховувалася наявність вбудованих апаратних засобів захисту.

Згідно з документацією, мікроконтролер має наступні механізми забезпечення відмовостійкості:

- Сторожовий таймер (WDT — Watchdog Timer): Незалежний апаратний таймер з власним RC-генератором (31 кГц). Його функція полягає в автоматичному перезавантаженні (Reset) мікроконтролера у випадку програмного збою («зависання»), викликаного сильною електромагнітною завадою. У даному проєкті WDT налаштовується на період спрацювання, що перевищує максимальний час виконання одного циклу програми.
- Скидання при зниженні напруги (BOR — Brown-out Reset): Схема моніторингу напруги живлення. Якщо напруга VDD падає нижче критичного рівня (налаштовується програмно, наприклад, 2.7В або 1.9В), контролер автоматично утримується в стані скидання до відновлення нормального живлення. Це запобігає непередбачуваній роботі процесора та помилковому включенню клапанів при нестабільному живленні.
- Таймер увімкнення живлення (PWRT — Power-up Timer): Забезпечує затримку запуску програми на 64 мс після подачі живлення. Це дозволяє напрузі стабілізуватися перед початком виконання коду ініціалізації портів.

3.3 Реалізація інтерфейсу зв'язку RS485

Для забезпечення надійного обміну діагностичною інформацією та командами керування між маніпулятором МП-9С та зовнішніми керуючими пристроями (персональним комп'ютером, системою ЧПК верстата) у проєкті обрано стандарт RS-485 (офіційна назва EIA/TIA-485).

Згідно з міжнародною класифікацією, RS-485 (рис.3.8) — це рекомендований стандарт, що описує електричні характеристики передавачів (драйверів) та

приймачів для використання у збалансованих (симетричних) багатоточкових системах передачі даних. Важливо зазначити, що даний стандарт визначає лише фізичний рівень (перший рівень моделі OSI) передачі сигналів, тобто параметри електричних імпульсів, не регламентуючи при цьому протокол обміну та формат даних.

Обґрунтування вибору для системи керування роботом

Вибір саме цього інтерфейсу обумовлений тим, що на сьогодні RS-485 є «робочою конячкою» (industry's interface workhorse) промислової автоматизації. Він став фізичною основою для більшості промислових мереж, таких як Modbus, Profibus DP, BitBus та інших, завдяки найкращому співвідношенню ціни, продуктивності та надійності при сучасному рівні розвитку технологій.

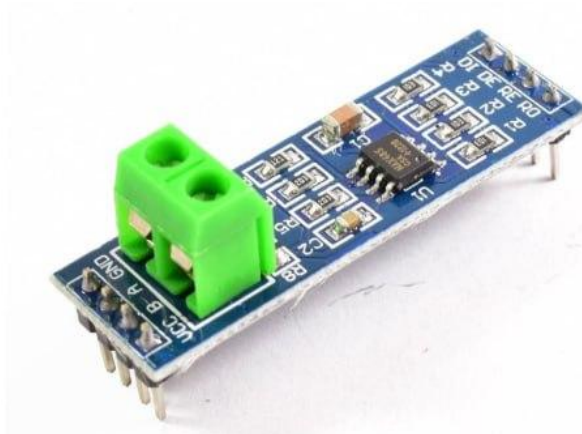


Рисунок 3.8 Інтерфейс RS 485

Ключові переваги RS-485, що стали вирішальними для даного проєкту:

- Висока завадостійкість: На відміну від інтерфейсу RS-232, де сигнал передається відносно спільної землі, RS-485 використовує диференціальний (балансний) спосіб передачі. Інформація зчитується як різниця потенціалів між двома проводами виті пари. Це дозволяє ефективно працювати в умовах сильних електромагнітних завад, характерних для виробничих цехів, оскільки зовнішня завада наводиться на обидва проводи однаково і компенсується при відніманні сигналів у приймачі.

- Багатоточковість (Multipoint): Стандарт дозволяє об'єднувати в єдину мережу до 32 стандартних пристроїв (трансиверів) на одній лінії, що дає можливість керувати групою роботів або периферійних модулів з одного контролера. При використанні сучасних мікросхем із вхідним опором 1/8 від стандартного, кількість пристроїв може бути збільшена до 256.

- Велика дальність зв'язку: Максимальна довжина лінії зв'язку може досягати 1200 метрів (4000 футів) при швидкості передачі до 100 кбіт/с, що значно перевищує можливості стандартного RS-232 (до 15 м).

- Економічність: Для організації двостороннього напівдуплексного обміну даними достатньо всього однієї витой пари проводів.

Основні технічні характеристики обраного інтерфейсу наведено нижче:

- Режим роботи: Напівдуплексний (дані передаються по черзі в обидва боки по одній парі).

- Максимальна швидкість передачі: до 10 Мбіт/с (на коротких відстанях до 12 м).

- Діапазон синфазної напруги: від -7 В до +12 В, що дозволяє витримувати значні перепади потенціалів «землі» між віддаленими вузлами.

- Чутливість приймача: ± 200 мВ. Це означає, що для розпізнавання логічного рівня достатньо, щоб різниця напруг між проводами перевищувала це значення.

Фундаментальною відмінністю RS-485 від однополярних інтерфейсів (таких як RS-232 або TTL, де напруга вимірюється відносно «землі») є використання балансної (диференціальної) схеми передачі. Сигнал у лінії формується не як абсолютне значення напруги на провіднику, а як різниця потенціалів між двома сигнальними лініями: прямою (A, non-inverting) та інверсною (B, inverting).

Логічні рівні та електричні параметри згідно зі стандартом, логічний стан лінії визначається різницею напруг $U_{AB} = U_A - U_B$:

- Логічна «1» (Mark, OFF): Відповідає стану, коли потенціал на лінії A вищий за потенціал на лінії B більше ніж на 200 мВ ($U_A - U_B > +200\text{мВ}$).

- Логічний «0» (Space, ON): Відповідає стану, коли потенціал на лінії B вищий за потенціал на лінії A ($U_A - U_B < -200\text{mV}$).
- Зона невизначеності: Діапазон напруг від -200 mV до $+200\text{ mV}$ є забороненим; стан приймача в цьому діапазоні не гарантується, що забезпечує гістерезис для захисту від шумів.

На практиці драйвери передавачів [Kugelstadt T. The RS-485 Design Guide] формують на виході значно вищі рівні напруги (мінімум $\pm 1.5\text{ V}$ на навантаженні 54 Ом), що забезпечує значний запас завадостійкості (Noise Margin) перед порогом чутливості приймача у $\pm 200\text{ mV}$. (рис.3.9).



Рисунок 3.9. Мінімальні рівні сигналу шини, зазначені для RS-485

Забезпечення завадостійкості. Висока надійність інтерфейсу в умовах промислових завод базується на двох факторах:

- Придушення синфазної завади (Common Mode Rejection): Зовнішнє електромагнітне поле (від електродвигунів, зварювального обладнання) наводить паразитну ЕРС на обох провідниках лінії зв'язку майже однаково. Оскільки приймач реагує лише на різницю потенціалів, ця синфазна складова автоматично віднімається ($(U_A + U_{\text{noise}}) - (U_B + U_{\text{noise}}) = U_A - U_B$) і не впливає на корисний сигнал. (рис.3.10).

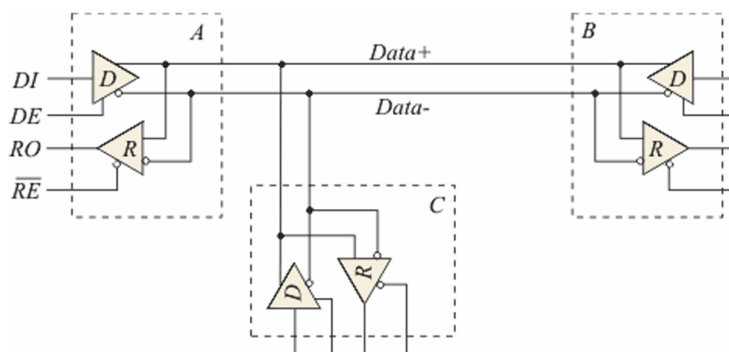


Рисунок 3.10. З'єднання трьох пристроїв з інтерфейсом RS-485

- Використання витої пари (Twisted Pair): Для фізичного середовища передачі обов'язковим є використання кабелю з витими провідниками. Скручування проводів гарантує, що вони знаходяться на однаковій середній відстані від джерела завад, що вирівнює наведені струми. Крім того, згідно з «правилом гвинта», магнітні поля, створювані струмами в провідниках А і В, спрямовані зустрічно і взаємно компенсуються, що значно зменшує випромінювання завад самою лінією зв'язку назовні.

При проектуванні фізичного рівня каналу зв'язку RS-485 критичне значення має вибір правильної топології з'єднання пристроїв. Згідно з рекомендаціями стандарту та посібника з проектування [Kugelstadt T. The RS-485 Design Guide], єдиною допустимою топологією для високошвидкісного обміну даними є «шина» (Daisy Chain).

Вимоги до топології у цій конфігурації всі пристрої мережі (маніпулятор МП-9С, пульт керування, система ЧПК) підключаються до магістрального кабелю послідовно, один за одним. Формування довгих відгалужень (Stubs) від магістралі вкрай небажане, оскільки вони діють як неузгоджені довгі лінії, що викликають вторинні відбиття сигналу, які спотворюють корисний імпульс.

Використання топології «зірка» (коли всі пристрої підключаються окремими кабелями в одну точку) або «кільце» для RS-485 заборонено, оскільки це призводить до неконтрольованих відбиттів сигналу та неможливості коректного узгодження хвильового опору лінії. Якщо конструктивно необхідно зробити відгалуження до пристрою, його довжина має бути мінімальною (електрично короткою) — такою, щоб час проходження сигналу по ньому був меншим за 1/10 часу наростання фронту імпульсу (Rise Time).

Узгодження лінії (Termination) оскільки кабель зв'язку (вита пара) розглядається як довга лінія з розподіленими параметрами, для запобігання ефекту відбиття сигналу від кінців кабелю (Signal Reflection) необхідно застосовувати узгоджувальні резистори — термінатори.

Згідно з фізичними принципами поширення хвиль, відбиття виникає, коли хвильовий опір кабелю (Z_0) не дорівнює опору навантаження. Для усунення цього

ефекту на обох фізичних кінцях магістралі встановлюються резистори R_T , номінал яких дорівнює характеристичному хвильовому опору кабелю.

- Для стандартного кабелю виті пари (наприклад, AWG 24) характеристичний імпеданс зазвичай становить 120 Ом.
- Отже, термінатори опором 120 Ом встановлюються паралельно сигнальним лініям А і В тільки на першому та останньому вузлах мережі (див. Рис. 2.X).

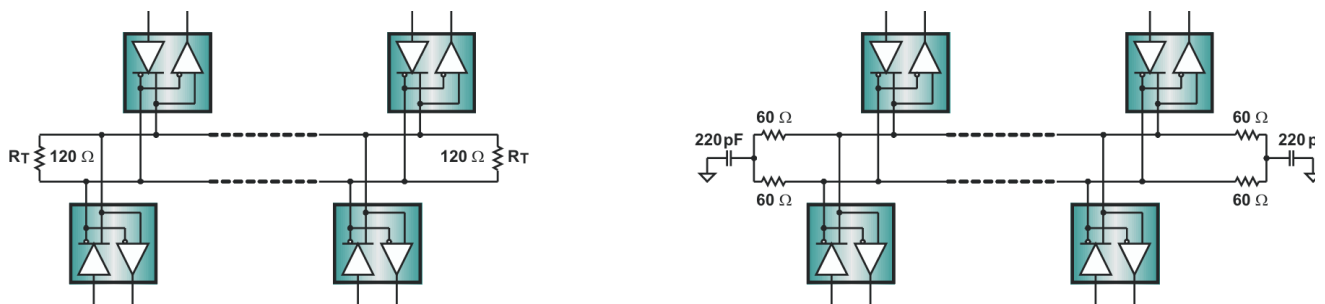


Рисунок 3.11. Правильне узгодження лінії RS-485

У системах робота може бути реалізований за допомогою джампера (перемички) на платі контролера, що дозволяє підключати резистор, якщо робот є кінцевим пристроєм у мережі, або відключати його, якщо він є проміжним вузлом.

При проектуванні кабельної мережі RS-485 необхідно враховувати фундаментальну залежність між максимально допустимою швидкістю передачі даних (Data Rate) та фізичною довжиною кабелю. Ця залежність є обернено пропорційною: чим довша лінія, тим нижчою має бути швидкість для забезпечення цілісності сигналу.

Фізичні обмеження накладаються двома факторами:

- На низьких частотах (довгі лінії): Основним обмежуючим фактором є активний опір кабелю (DC resistance). Падіння напруги на провідниках призводить до того, що сигнал на кінці лінії стає меншим за поріг чутливості приймача. Для стандартного кабелю виті пари (24 AWG) максимальна рекомендована довжина складає близько 1200 метрів при швидкості до 100 кбіт/с.

- На високих частотах (короткі лінії): Обмеження вносить ємність кабелю та час наростання фронту сигналу, що викликає явище джитера (Jitter) — фазового тремтіння імпульсів.

Графічна залежність Для вибору робочої точки системи використовується стандартний графік залежності довжини від швидкості (Cable Length vs Data Rate), наведений у рис3.12.

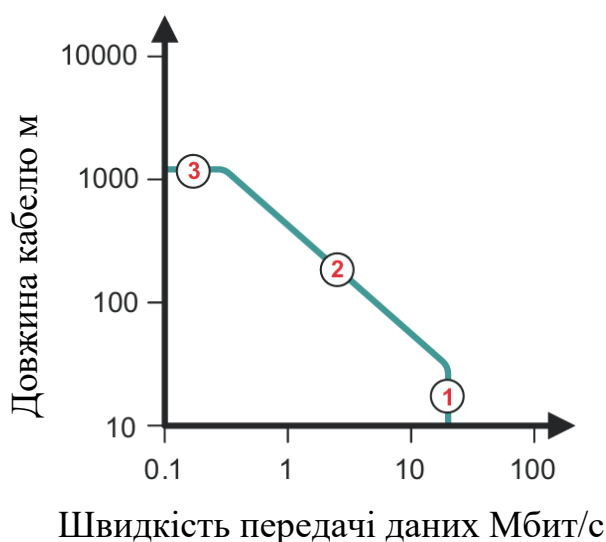


Рисунок 3.12. Правильне узгодження лінії RS-485

Аналіз даної характеристики дозволяє виділити три робочі зони:

- Зона високих швидкостей: При довжині лінії до 12 метрів можлива передача даних на швидкостях до 10 Мбіт/с (і навіть до 50 Мбіт/с з сучасними трансиверами).
- Перехідна зона: При збільшенні довжини від 12 м до 1200 м максимальна швидкість лінійно зменшується.
- Зона довгих ліній: При довжині понад 1200 м робота не гарантується.

Розрахунок для системи керування роботом В умовах роботизованого комплексу відстань від робота МП-9С до шафи керування або стійки ЧПК зазвичай не перевищує 30–50 метрів. Згідно з графіком, для такої відстані інтерфейс RS-485 дозволяє використовувати швидкості порядку 2–5 Мбіт/с. Однак, для забезпечення максимальної завадостійкості та сумісності зі стандартними СОМ-портами промислових ПК, у проєкті обрано фіксовану швидкість обміну 115 200 біт/с (або

9600 біт/с). Ця точка на графіку знаходиться глибоко в "безпечній зоні" (conservative operating area), що гарантує стабільний зв'язок навіть при використанні недорогого неекранованого кабелю та наявності зовнішніх електромагнітних завад.

Вплив джитера (нестабільності фронтів сигналів) при такій довжині лінії та швидкості не перевищує допустимих 5–10%, що не впливає на вірогідність розпізнавання бітів приймачем.

Незважаючи на те, що RS-485 використовує диференціальний спосіб передачі сигналу і теоретично не потребує спільної «землі» для розпізнавання логічних рівнів, на практиці ігнорування питань заземлення є найпоширенішою причиною виходу обладнання з ладу.

Проблема різниці потенціалів земель (GPD) У промислових умовах маніпулятор та керуючий ПК можуть житися від різних фаз електромережі та знаходитися на значній відстані один від одного. Це призводить до виникнення різниці потенціалів земель (GPD — Ground Potential Difference). Згідно зі стандартом RS-485, допустимий діапазон синфазної напруги (Common Mode Voltage) на входах приймача становить від -7 В до +12 В. Якщо різниця потенціалів між «землею» робота і «землею» комп'ютера перевищить ці межі, через мікросхему драйвера потече вирівнюючий струм руйнівної сили, що призведе до перегорання порту або нестабільності зв'язку.

Ефект «земляної петлі» (Ground Loop) Пряме з'єднання заземлюючих шин віддалених пристроїв через сигнальний кабель створює замкнений контур — «земляну петлю». В умовах роботи потужного електрообладнання (двигунів маніпулятора) у цьому контурі індукуються значні паразитні струми, які накладаються на корисний сигнал у вигляді шуму.

Технічне рішення: гальванічна ізоляція для повного усунення зазначених проблем у розробленій системі керування застосовано метод гальванічної розв'язки. Згідно з рекомендаціями [Kugelstadt T. The RS-485 Design Guide], ізоляція повинна розривати прямий електричний контакт між шиною даних та логічною схемою контролера.

Реалізація розв'язки включає два компоненти:

- Ізоляція сигнальних ліній: Використання цифрових ізоляторів (наприклад, на базі оптопар або ємнісних бар'єрів), які передають логічні сигнали RX/TX/DE за допомогою світла або електромагнітного поля, без електричного контакту.
- Ізоляція живлення: Використання ізованого DC-DC перетворювача, який формує незалежну напругу живлення для мікросхеми трансивера RS-485.

Така схема (рис. 3.13) дозволяє "плаваючій" землі трансивера слідувати за змінами потенціалу в лінії, не впливаючи на чутливу електроніку мікроконтролера, та гарантує надійну роботу навіть при стрибках напруги в системі заземлення цеху до кількох кіловольт.

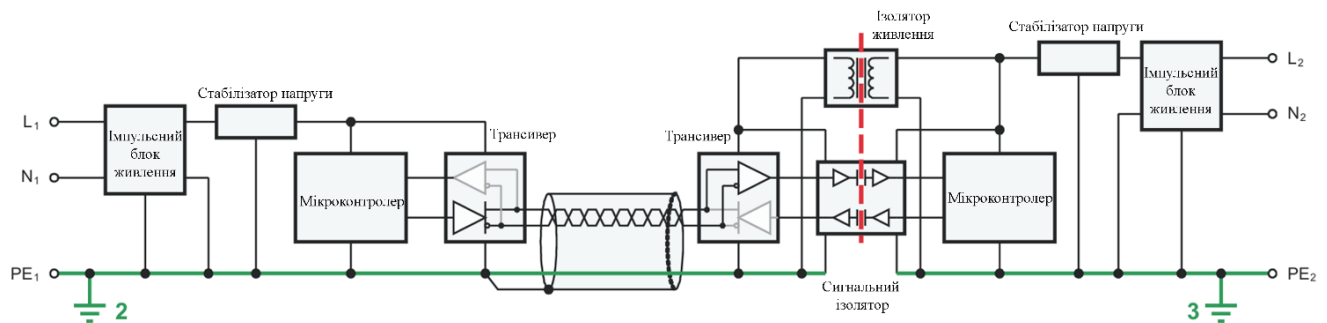


Рисунок 3.13. Схема гальванічної ізоляції двох віддалених вузлів мережі зі спільною точкою заземлення

Порівняльний аналіз інтерфейсів RS-232, RS-422 та RS-485

Вибір стандарту фізичного рівня для промислової мережі базується на аналізі трьох ключових критеріїв (таб. 3.8.): завадостійкості, топологічних можливостей та дальності зв'язку. Нижче наведено детальний аналіз розглянутих альтернатив.

Інтерфейс RS-232 (EIA/TIA-232) Це найстаріший стандарт, розроблений для підключення термінального обладнання (DTE) до комунікаційного (DCE), наприклад, комп'ютера до модему.

- Фізичні обмеження: Використовує несиметричну (unbalanced) схему передачі, де сигнал вимірюється відносно спільної «землі» (Signal Ground). Логічні рівні складають $\pm 5 \pm 15$ В.

Недоліки для проєкту:

- Вразливість до завад: Оскільки «земля» є спільною для сигналів і живлення, будь-які перешкоди на шині заземлення (наприклад, від роботи зварювального апарату в цеху) безпосередньо додаються до корисного сигналу, що призводить до помилок передачі.
- Обмежена дальність: Велика амплітуда сигналу (до 25 В у розмаху) обмежує швидкість наростання фронтів (Slew Rate), тому стандартна гарантована дальність складає лише 15 метрів, що недостатньо для цехових умов.
- Архітектура «Точка-Точка»: Стандарт дозволяє з'єднати лише два пристрої. Для керування групою виконавчих механізмів довелося б тягнути окремий кабель до кожного вузла, що ускладнює конструкцію робота.

Таблиця 3.8. Порівняльна характеристика інтерфейсів

Параметр	RS-232	RS-422	RS-485 (Обрано)
Спосіб передачі сигналу	Однофазний (несиметричний)	Диференціальний (симетричний)	Диференціальний (симетричний)
Топологія мережі	Точка-точка (Point-to-Point)	Багатоточечна (Multidrop)	Багатоточечна (Multipoint)
Кількість пристроїв (Драйверів/Приймачів)	1 / 1	1 / 10	32 / 32 (до 256)
Режим роботи	Повний дуплекс	Повний дуплекс	Напівдуплекс (2 дроти) або Повний (4 дроти)
Максимальна довжина лінії	15 м	1200 м	1200 м
Максимальна швидкість передачі	20 кбіт/с (до 115 кбіт/с)	10 Мбіт/с	10 Мбіт/с (до 35 Мбіт/с)
Чутливість приймача	±3 В	±200 мВ	±200 мВ
Вхідний опір приймача	3...7 кОм	≥ 4 кОм	≥ 12 кОм

Інтерфейс RS-422 (EIA/TIA-422)

Є еволюційним розвитком, що впровадив диференціальну передачу сигналу.

- Особливості: Сигнал передається по двох проводах (А і В), як і в RS-485. Це забезпечує високу дальність (до 1200 м) та швидкість.
- Критичний недолік: Стандарт визначає архітектуру шини як Multi-drop (багатоточкова розсилка), а не Multipoint. Це означає, що в мережі може бути лише один активний передавач (Master) і до 10 приймачів (Slaves).
- Чому не підходить: У системі керування роботом необхідний двосторонній обмін: контролер робота повинен не лише отримувати команди, а й надсилати звіти про стан датчиків. RS-422 вимагав би організації чотирипровідної лінії (дві пари: одна на прийом, одна на передачу), що збільшує вартість кабельної продукції.

Інтерфейс RS-485 (Обраний стандарт)

Є стандартом де-факто для побудови розподілених систем керування (Fieldbus).

- Перевага архітектури Multipoint: Головною відмінністю від RS-422 є можливість роботи декількох передавачів (до 32 стандартних, до 256 спеціальних) на одній парі проводів. Драйвери RS-485 мають спеціальний режим «третього стану» (High-Z), що дозволяє їм відключатися від лінії, коли вони не передають дані, звільняючи канал для інших пристроїв.
- Електрична витривалість: Стандарт вимагає працездатності при синфазних напругах від -7 В до +12 В. Це означає, що навіть якщо потенціали «землі» біля робота і біля стійки керування відрізняються на кілька вольт (що часто буває на виробництві через потужні струми витоку), зв'язок не перерветься і порти не згорять.
- Чутливість: Приймачі RS-485 розпізнають сигнал рівнем всього 200 мВ, тоді як передавачі видають у лінію мінімум 1.5 В. Такий величезний запас завадостійкості (Noise Margin) дозволяє використовувати недорогий кабель витої пари навіть у середовищі з сильними електромагнітними полями.

Підсумовуючи тільки стандарт RS-485 забезпечує поєднання двонаправленого обміну, мінімальної кількості проводів (2 жили), роботи в режимі справжньої мережі та високої стійкості до промислових перешкод, що робить його безальтернативним вибором для даного дипломного проєкту.

3.4 Структура та опис спроектованої схеми.

Спроектована принципова електрична схема (рис. 3.14) системи керування маніпулятором на базі мікроконтролера PIC16F1937 є комплексним апаратним рішенням, яке поєднує модулі введення, виведення, комунікації та індикації. Кожен із функціональних вузлів виконує окрему роль у забезпеченні стабільної роботи системи керування роботизованим маніпулятором, а їх інтеграція формує завершену автоматизовану систему. На рисунку подано узагальнену структурну схему, на основі якої розроблено електричне з'єднання компонентів.

Центральним елементом конструкції виступає мікроконтролер PIC16F1937 (IC1), виконаний у корпусі з 40 виходами, що забезпечує достатню кількість портів для підключення виконавчих механізмів, органів керування та допоміжних модулів. У схемі задіяні майже всі порти мікроконтролера, що дозволяє реалізувати необхідну логіку керування восьмиканальною системою приводів маніпулятора, а також забезпечити підтримку інтерфейсу зв'язку і локального меню.

До входів мікроконтролера підключено групу тактових кнопок S1–S6. Схемотехнічно вони реалізовані з використанням підтягувальних резисторів R9–R14, які підключені до шини живлення (+5V). Такий підхід забезпечує стабільний логічний рівень «1» на вході мікроконтролера у відпущеному стані, а при натисканні кнопки вхід замикається на землю (GND), формуючи активний рівень логічного «0». Кнопки виконують функції локального керування маніпулятором або використані для налаштування режимів роботи, навігації по меню та запуску тестів.

Окремою групою у схемі реалізовано канал керування виконавчими реле, які підключаються до роз'ємів XP1–XP8 (Rel1–Rel8). Сигнали керування від портів мікроконтролера проходять через струмообмежувальні резистори R1–R8, які

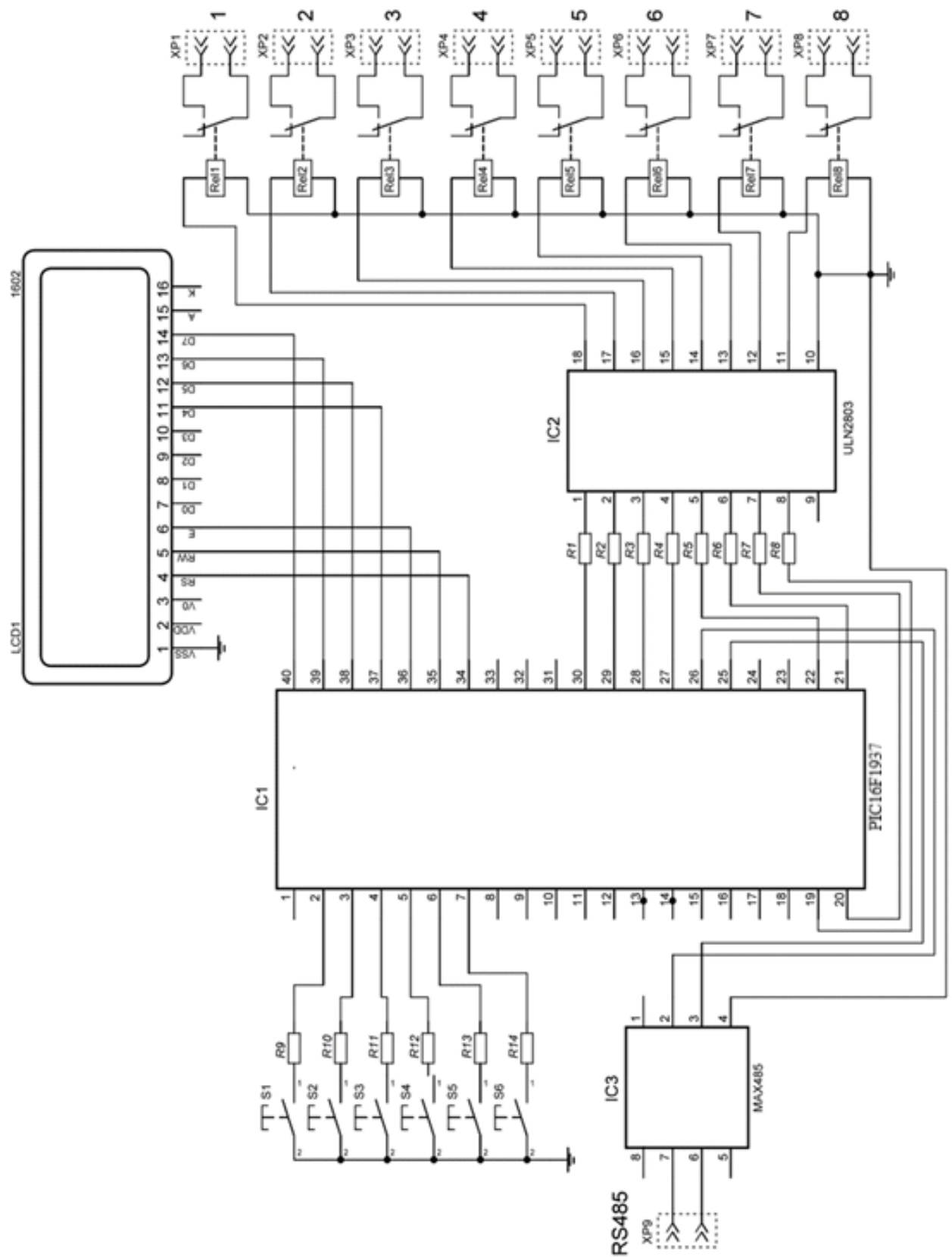


Рисунок. 3.14 Принципова електрична схема системи керування маніпулятором

формують необхідний струм бази для вхідних каскадів силового драйвера. Це дозволяє мікроконтролеру чітко комутувати навантаження на кожному з восьми каналів. Наявність восьми окремих ліній керування дозволяє незалежно вмикати та вимикати кожен ступінь свободи маніпулятора або активувати допоміжні механізми (захват, гальма).

Для безпосередньої комутації силових навантажень використано драйвер транзисторного типу ULN2803 (IC2). Він являє собою восьмиканальну збірку складених транзисторів Дарлінгтона, яка дозволяє мікроконтролеру керувати реле та електромагнітами зі значним струмом споживання. Виходи драйвера підключені до роз'ємів XP1–XP8, до яких приєднуються обмотки реле. Використання ULN2803 забезпечує розв'язку за струмом між ніжною логічною частиною та індуктивним навантаженням, значно знижуючи ризик теплового пошкодження портів PIC16F1937.

Взаємодія мікроконтролера з ULN2803 реалізована за прямою логікою: при подачі логічної «1» на вихід порту контролера, сигнал через резистор (наприклад, R1) відкриває відповідний канал драйвера, замикаючи вихід на землю. Це активує підключене до роз'єму Rel1 реле. Наявність вбудованих захисних діодів у складі мікросхеми ULN2803 (підключених до виводу 10) ефективно гасить сплески зворотної ЕРС, що неминуче виникають при комутації котушок реле, захищаючи вихідні транзистори від пробую.

Для забезпечення зовнішнього зв'язку система містить модуль інтерфейсу RS485 на базі мікросхеми MAX485 (IC3). Цей модуль дає змогу організувати стійкий до завад обмін даними між контролером маніпулятора та персональним комп'ютером або пультом оператора. Завдяки RS485 система може працювати на значних відстанях в умовах промислового цеху. MAX485 узгоджений з портами PIC16F1937 через лінії RX/TX, а керування напрямком передачі здійснюється через об'єднані виводи дозволу.

Для індикації стану системи передбачено підключення LCD-дисплея (LCD1) типу 1602. Він взаємодіє з мікроконтролером через паралельний інтерфейс, використовуючи лінії керування (RS, RW, E) та шину даних. Наявність дисплею

дозволяє виводити на екран інформацію про поточні координати, номер виконуваної програми, діагностичні повідомлення або коди помилок. Користувач отримує можливість візуально контролювати процес роботи без необхідності підключення зовнішнього монітора.

Схема також передбачає коректну організацію кіл живлення та заземлення. Усі цифрові модулі (IC1, IC3) та драйвер (IC2) мають спільну точку підключення до землі, що мінімізує різницю потенціалів та паразитні контури струму. Розводка плати виконана таким чином, щоб силові струми комутації реле не впливали на стабільність роботи тактового генератора мікроконтролера та чутливих ліній зв'язку.

Таким чином, розроблена схема є логічно завершеною структурою, що поєднує всі необхідні модулі для повноцінного керування маніпулятором. Мікроконтролер PIC16F1937 виступає інтелектуальним ядром, ULN2803 забезпечує силову комутацію реле через порти XP1–XP8, MAX485 реалізує промисловий канал зв'язку, а периферія у вигляді кнопок та дисплея створює зручний інтерфейс користувача. Така архітектура повністю відповідає технічному завданню та забезпечує надійність роботи комплексу.

Висновки за розділом

У третьому розділі було виконано проектування апаратної частини системи керування робота-маніпулятора МП-9С на основі мікроконтролера PIC16F1937. Розроблена принципова електрична схема охоплює всі ключові вузли: центральний мікроконтролер, драйвер потужності ULN2803 для комутації навантажень, модуль візуальної індикації на LCD, блок кнопок ручного керування та промисловий інтерфейс зв'язку RS-485 на базі трансивера MAX485. Це дозволило створити повноцінну та гнучку керуючу платформу, повністю сумісну з вимогами до автоматизованих систем.

Обґрунтовано вибір мікроконтролера PIC16F1937 як обчислювального ядра системи завдяки наявності достатньої кількості портів введення-виведення,

розвиненої периферії та можливості реалізовувати складні логічні алгоритми. Схема електричних з'єднань сформована таким чином, щоб забезпечити максимальну зручність взаємодії цифрової частини з виконавчими механізмами та органами керування.

Інтеграція восьмиканального драйвера ULN2803 дозволила забезпечити надійне узгодження низьковольтних логічних сигналів із потужними виконавчими елементами (реле Rel1–Rel8), гарантуючи захист портів мікроконтролера від перевантажень та зворотних струмів. Реалізація блоку тактових кнопок забезпечує можливість локального налаштування та оперативного контролю, а інтерфейс RS-485 створює канал для стійкого до завад віддаленого обміну даними, що є критично важливим для інтеграції маніпулятора у промислову мережу.

Виконаний аналіз функціональних зв'язків між розробленими вузлами підтвердив цілісність, надійність та логічну завершеність структури схеми. Отримана апаратна реалізація повністю відповідає технічному завданню та слугує надійною базою для подальшої розробки програмного забезпечення.

Таким чином, у межах розділу сформовано завершену апаратну архітектуру контролера, що забезпечує ефективну модернізацію системи керування маніпулятора МП-9С та створює необхідні передумови для впровадження сучасних алгоритмів автоматичного керування.

РОЗДІЛ 4

ВПРОВАДЖЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

4.1 Розробка та опис функціональної схеми маніпулятора

Функціональна схема (рис. 4.1) відображає ієрархічні взаємозв'язки між основними апаратними модулями розробленого програмованого логічного контролера на базі мікроконтролера PIC16F1937 та виконавчими механізмами маніпулятора МП-9С. Структура системи побудована за модульним принципом, щоб забезпечити автономне, стабільне й безпечне керування приводами маніпулятора з можливістю організації зворотного зв'язку від датчиків положення, кінцевих вимикачів та аналогових сенсорів. Центральним елементом усієї системи є ПЛК, функції якого реалізуються за допомогою мікроконтролера PIC16F1937, котрий виконує обробку даних у реальному часі, формування керуючих сигналів і взаємодію з оператором.

У верхній частині функціональної схеми розташований інтерфейс користувача, який включає блок локальних кнопок керування S1–S6, рідкокристалічний індикатор LCD1602 та інтерфейс для підключення персонального комп'ютера. Кнопки підключені до цифрових портів Port A (RA0–RA5) мікроконтролера. Вони дозволяють реалізувати ручний режим керування ланками, вибір автоматичних програм, калібрування системи та аварійну зупинку. LCD-індикатор, підключений через паралельний інтерфейс, слугує для відображення поточного стану системи, діагностичних повідомлень, координат маніпулятора та кодів помилок. Підключення ПК здійснюється через промисловий інтерфейс RS-485 на базі трансивера MAX485, що забезпечує надійну передачу даних на великі відстані та захист портів контролера від електромагнітних завад.

Центральним ядром схеми є ПЛК на основі мікроконтролера PIC16F1937, який виконує функції арбітра шини, реалізації алгоритмів траєкторного керування та генерації вихідних імпульсів. На його базі реалізовані наступні логічні модулі: блок обробки команд від інтерфейсу користувача, модуль таймерів для

синхронізації рухів, та модуль керування виконавчими пристроями. Згідно з функціональною схемою, для забезпечення зворотного зв'язку (Closed-Loop Control) у структурі контролера зарезервовано використання вільних апаратних ресурсів:

- Port E (виводи 8, 9, 10 / RE0–RE2) призначені для підключення аналогових датчиків положення (потенціометрів), оскільки ці виводи мають функцію входів АЦП (AN5–AN7).
- Вхід RB0/INT (вивід 33) призначений для підключення цифрових датчиків (енкодерів або кінцевих вимикачів), що потребують миттєвої реакції через механізм зовнішнього переривання.

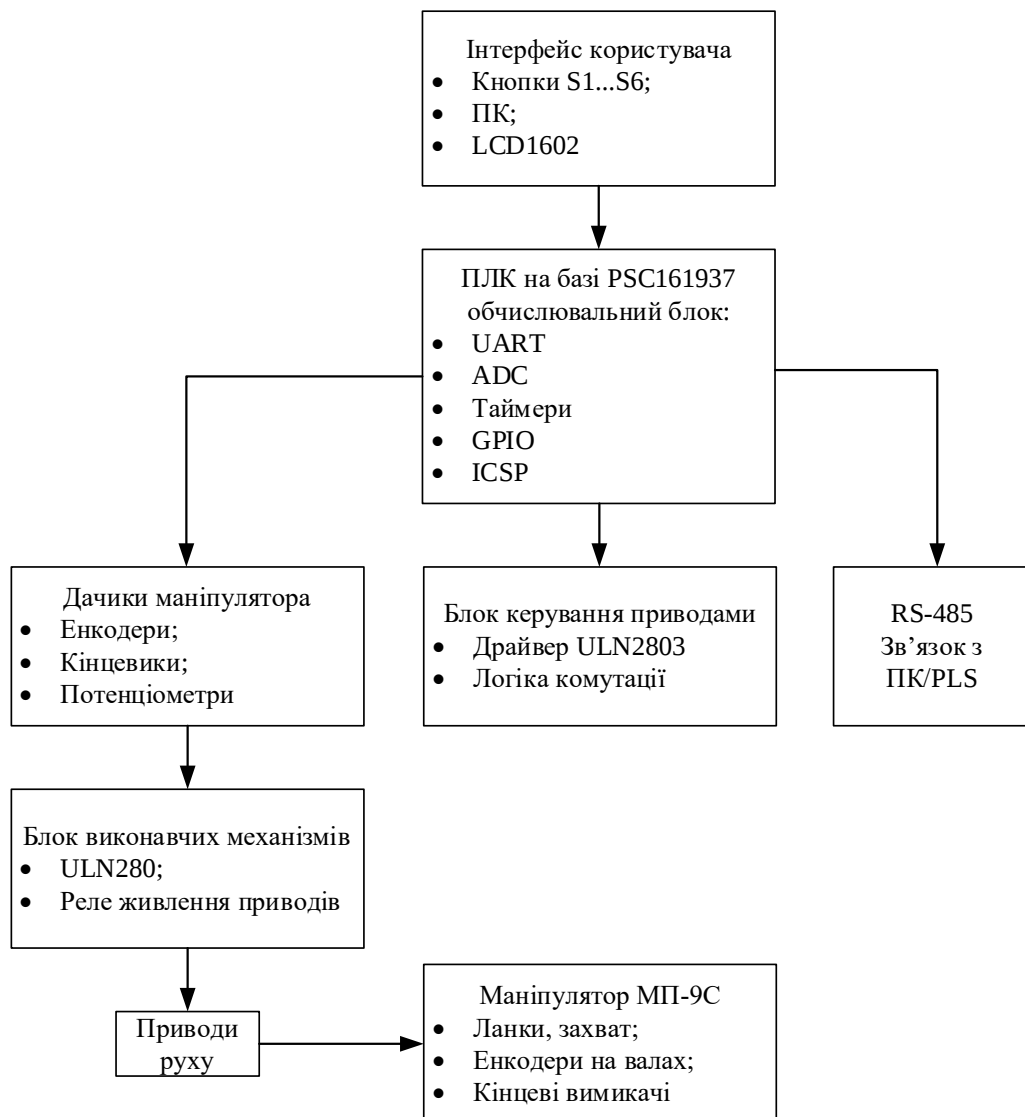


Рисунок 4.1. Функціональна схема

Важливою частиною системи є блок керування приводами, який функціонально виконує роль підсилювача потужності та комутатора полярності. Він включає мікросхему драйвера ULN2803 та блок силових реле. Виходи мікроконтролера підключаються до входів ULN2803, яка забезпечує узгодження низьковольтної логіки з котушками реле та захист від зворотних струмів. Силові реле, в свою чергу, реалізують функцію мостової комутації (H-bridge), забезпечуючи подачу напруги живлення на двигуни та зміну її полярності для реверсування руху. Таке рішення дозволяє керувати потужними навантаженнями, забезпечуючи повну гальванічну ізоляцію між двигунами та мікроконтролером.

Безпосереднім виконавчим елементом системи виступає маніпулятор МП-9С, який складається з набору електроприводів постійного струму, редукторів та рухомих ланок. Драйвери через релейну групу подають силові сигнали на приводи маніпулятора, забезпечуючи рух по заданих осях.

Комунікаційний модуль MAX485, який з'єднує мікроконтролер з ПК або іншими системами автоматизації, забезпечує можливість зовнішнього моніторингу та інтеграції в SCADA-системи. Через нього передаються параметри руху маніпулятора, статуси датчиків, аварійні повідомлення та оновлення конфігураційних параметрів ПЛК.

Таким чином, розроблена функціональна схема формує повноцінну систему керування, де мікроконтролер PIC16F1937 виступає ядром, блок ULN2803 з реле забезпечує силову комутацію, а наявність зарезервованих портів (Port E, RB0) гарантує можливість підключення сенсорів, зазначених у технічному завданні.

4.2 Керуючі алгоритми програми

Розроблений алгоритм керування роботом-маніпулятором (рис.4.2.) є ключовим компонентом програмного забезпечення системи автоматизації, що визначає логіку взаємодії між обчислювальним ядром (мікроконтролером PIC16F1937), виконавчими комутаційними механізмами та сенсорною підсистемою. Основне призначення алгоритму полягає у забезпеченні стабільного,

точного та безпечного функціонування маніпулятора як в ручному, так і в автоматичному режимах.

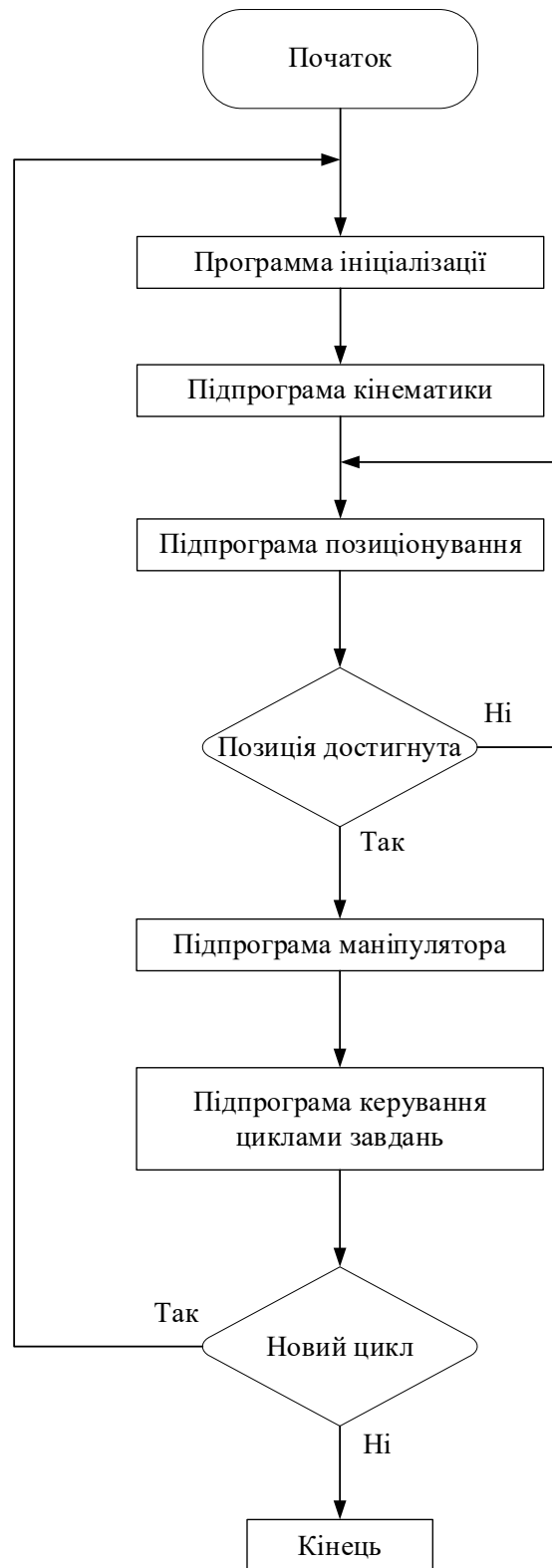


Рисунок 4.2. Алгоритм основної програми

Загальна структура програмного забезпечення побудована за модульним принципом і складається з головного циклу (Main Loop) та набору функціональних підпрограм, які викликаються за необхідності. Такий підхід дозволяє оптимізувати використання ресурсів мікроконтролера та забезпечити передбачуваний час реакції на зовнішні події.

Підпрограма ініціалізації (рис. 4.3) є критично важливим етапом, що забезпечує безпечний старт системи. Її виконання починається одразу після апаратного скидання мікроконтролера.

Алгоритм ініціалізації включає наступні кроки:

- Скидання регістрів МК: На першому етапі виконується примусове очищення регістрів спеціального призначення (SFR) та змінних оперативної пам'яті для уникнення невизначених станів. Перевіряються біти конфігурації (Configuration Bits).
- Налаштування портів GPIO: Відбувається запис у регістри напрямку TRIS. Порти, підключені до кнопок та кінцевиків, конфігуруються як входи, а порти керування драйвером ULN2803 — як виходи.

Ініціалізація таймерів та переривань: Налаштовуються таймери (Timer0/Timer1) для відліку системного часу. Активується система переривань та перевіряється коректність їх генерації.

- Ініціалізація генератора імпульсів (ШІМ/Таймінг): Згідно з алгоритмом, виконується налаштування модуля генерації сигналів (CCP/PWM). У контексті релейної схеми цей етап перевіряє стабільність тактової частоти, необхідної для точного відліку часових інтервалів комутації реле. Якщо сигнал нестабільний, система виконує повторне налаштування.
- Ініціалізація RS-485/UART: Налаштовується швидкість обміну даними (Baud Rate). Система очікує тестовий пакет відповіді від блоку приводів або ПК.

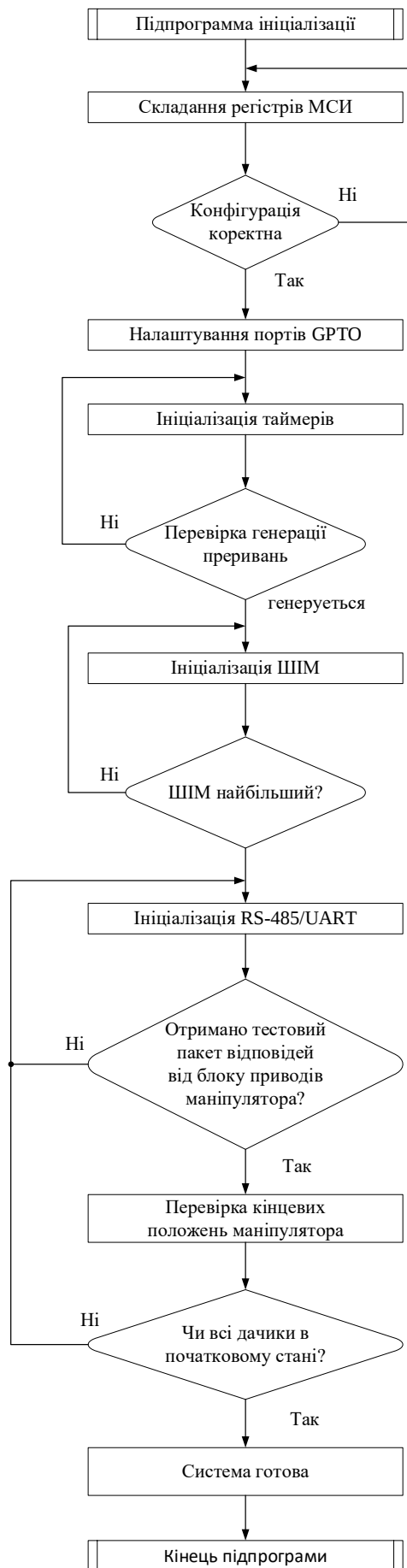


Рисунок 4.3. Алгоритм підпрограми ініціалізації

- Перевірка кінцевих положень: Останнім етапом є опитування датчиків. Алгоритм перевіряє, чи знаходяться всі механізми в допустимій зоні і чи датчики перебувають у початковому стані.

Тільки за умови проходження всіх перевірок ("ТАК" на схемі) система встановлює статус "Система готова".

Підпрограма кінематики (рис. 4.4) відповідає за математичний розрахунок руху. Вона перетворює абстрактні координати цілі у фізичні команди для виконавчих механізмів.

Логіка роботи підпрограми:

- Зчитування та отримання координат: Система отримує дані про поточне положення маніпулятора (через датчики зворотного зв'язку на Port E/RB0) та нові цільові координати від головної програми.
- Обчислення помилки: Розраховується вектор різниці між поточним та бажаним станом. Якщо помилка знаходиться в межах допуску, рух не потрібен.
- Розрахунок зворотної кінематики: Якщо помилка значна, алгоритм виконує обчислення необхідних кутів поворотів суглобів (ланок) маніпулятора для досягнення цільової точки. Враховуються геометричні довжини ланок.
- Аналіз обмежень та колізій: Перед початком руху перевіряється, чи не виходить розрахована точка за межі дозволеної робочої зони. Також розраховується траєкторія з урахуванням обмежень швидкості та прискорення (для запобігання перевантаженню реле та механіки).
- Формування та надсилання команд: На основі розрахунків формується бітова маска для драйвера ULN2803, яка активує відповідні реле. Команда надсилається на виконавчі механізми, після чого цикл повторюється до досягнення цілі.

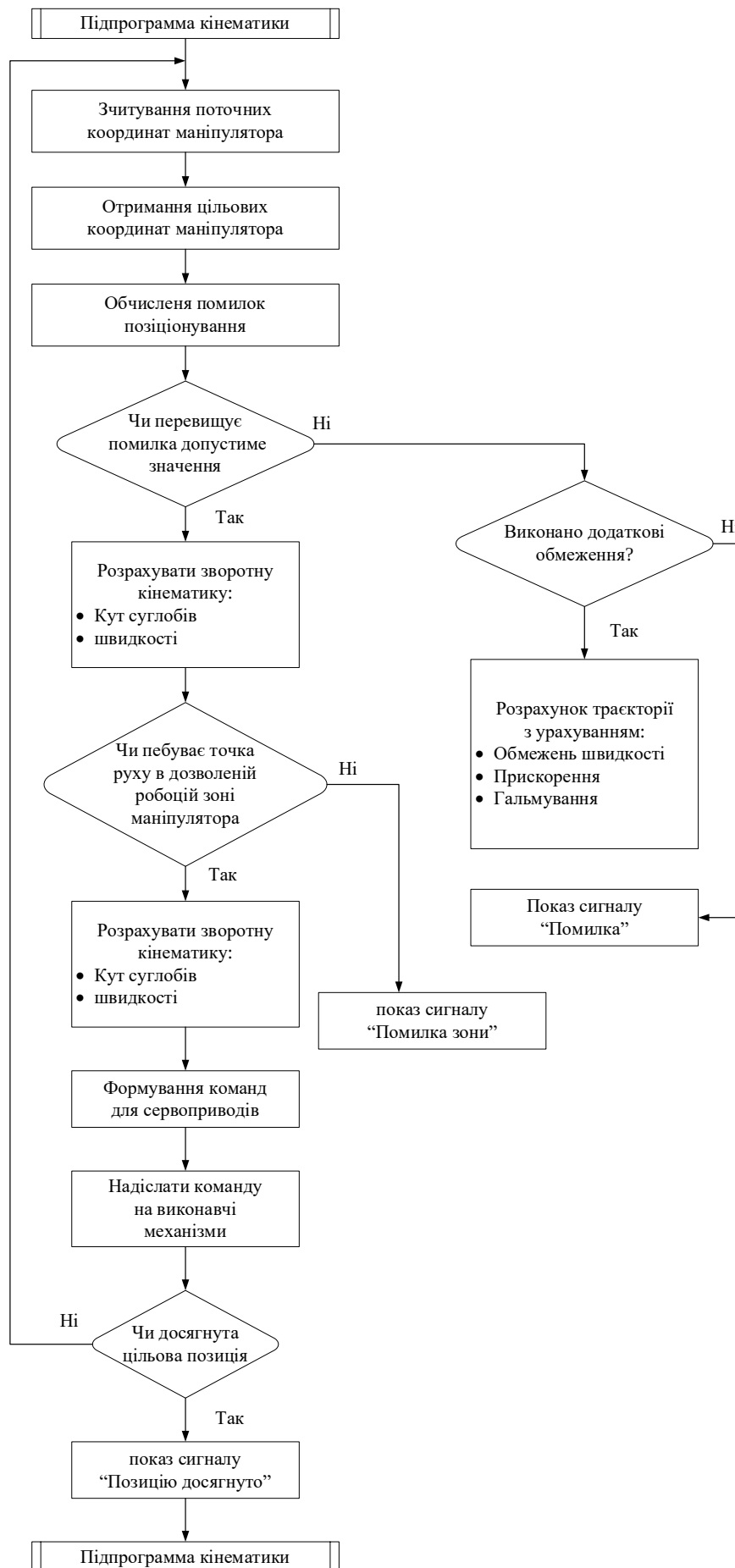


Рисунок 4.4. Алгоритм підпрограми кінематики

Підпрограма позиціонування (рис. 4.5) є виконавчим ядром системи руху, що працює в тісній інтеграції з модулем кінематики. Її основна задача — перетворити абстрактні математичні координати цільової точки у фізичні сигнали комутації реле, забезпечуючи переміщення робочого органу маніпулятора з точки А в точку Б з заданою точністю.

Алгоритм реалізує ітераційний підхід замкненого циклу регулювання (Closed-Loop Control) і складається з наступних етапів:

Отримання цільової позиції (X, Y, Z, φ): На вході підпрограма отримує масив даних від модуля кінематики або головної програми, що містить координати бажаного положення виконавчого органу та необхідний кут орієнтації захвату.

Зчитування поточних координат: Мікроконтролер ініціює опитування сенсорної підсистеми. Виконується зчитування аналогових сигналів з потенціометрів (через АЦП на портах Port E) або підрахунок імпульсів від енкодерів (через переривання на RB0). Отримані «сирі» дані фільтруються програмним методом для усунення шумів та перераховуються у метричні або кутові величини.

Розрахунок вектора помилки: Визначається різниця (Δ) між цільовим значенням та фактичним положенням для кожної ступені рухливості. Цей вектор визначає не лише відстань до цілі, а й знак, що вказує на необхідний напрямок обертання приводу.

- Перевірка нелінійних умов та блокувань: Перед початком руху система виконує предиктивний аналіз безпеки:
- Перевіряється, чи не виходить цільова точка за межі програмних лімітів робочої зони.
- Аналізується стан кінцевих вимикачів.
- Якщо виявлено механічну перешкоду або заборонену конфігурацію, алгоритм переходить по гілці «Ні», формує код помилки позиціонування та повертає його в головну програму для аварійної зупинки.

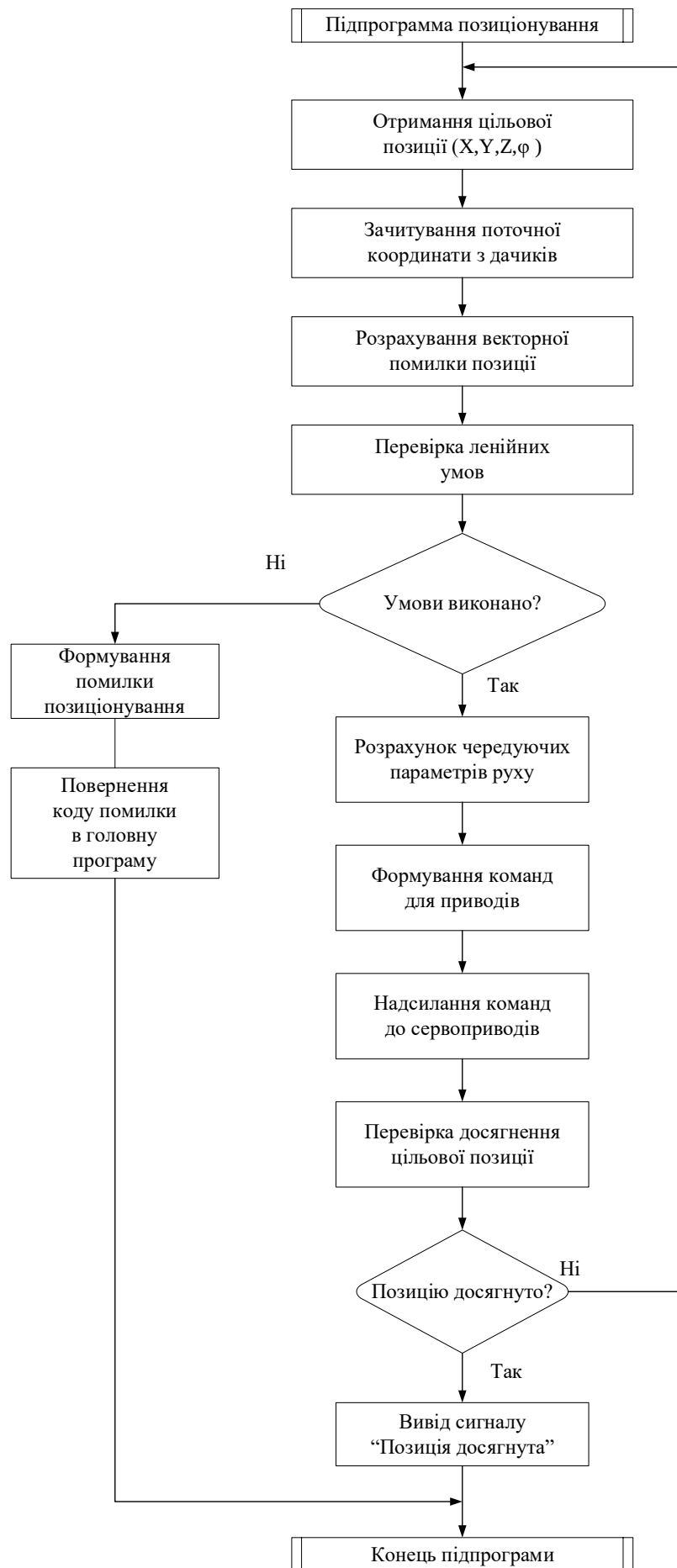


Рисунок 4.5. Алгоритм підпрограми позиціонування

Розрахунок керуючих параметрів руху: Якщо умови безпеки виконано («ТАК»), алгоритм визначає стратегію комутації. Враховуючи релейний характер керування (через ULN2803), замість плавного регулювання швидкості застосовується логіка трипозиційного релейного регулятора з гістерезисом:

- Визначається полярність підключення двигуна (Прямий хід або Реверс) на основі знаку помилки.
- Встановлюється зона нечутливості (Deadband), щоб уникнути частих перемикань реле («дребезгу») при незначних відхиленнях від цілі.

Формування та надсилання команд: Мікроконтролер генерує керуючі сигнали на порти, підключені до драйвера ULN2803. Відбувається фізичне перемикання відповідних реле, що подає живлення на двигуни. Цикл моніторингу досягнення цілі: Після активації приводів система переходить у режим безперервного відстеження:

- У кожній ітерації циклу перевіряється умова: $|\Delta| \leq \text{Допуск}$.
- Також контролюється час виконання руху (Watchdog timer): якщо ціль не досягнуто за розрахунковий час (наприклад, через заклинювання), рух примусово зупиняється.
- Як тільки позицію досягнуто (помилка стала меншою за допуск), мікроконтролер знімає сигнали з реле (гальмування вибігом або динамічне гальмування) і генерує статусний сигнал «Позиція досягнута».

Завершення роботи підпрограми відбувається передачею управління диспетчеру завдань для переходу до наступної операції технологічного циклу.

Підпрограма «Маніпулятор» (рис. 4.6) є спеціалізованим функціональним модулем програмного забезпечення, що відповідає за безпосередню роботу кінцевого ефектора (робочого інструменту) робота. На відміну від підпрограм кінематики, які керують просторовим положенням ланок, цей алгоритм

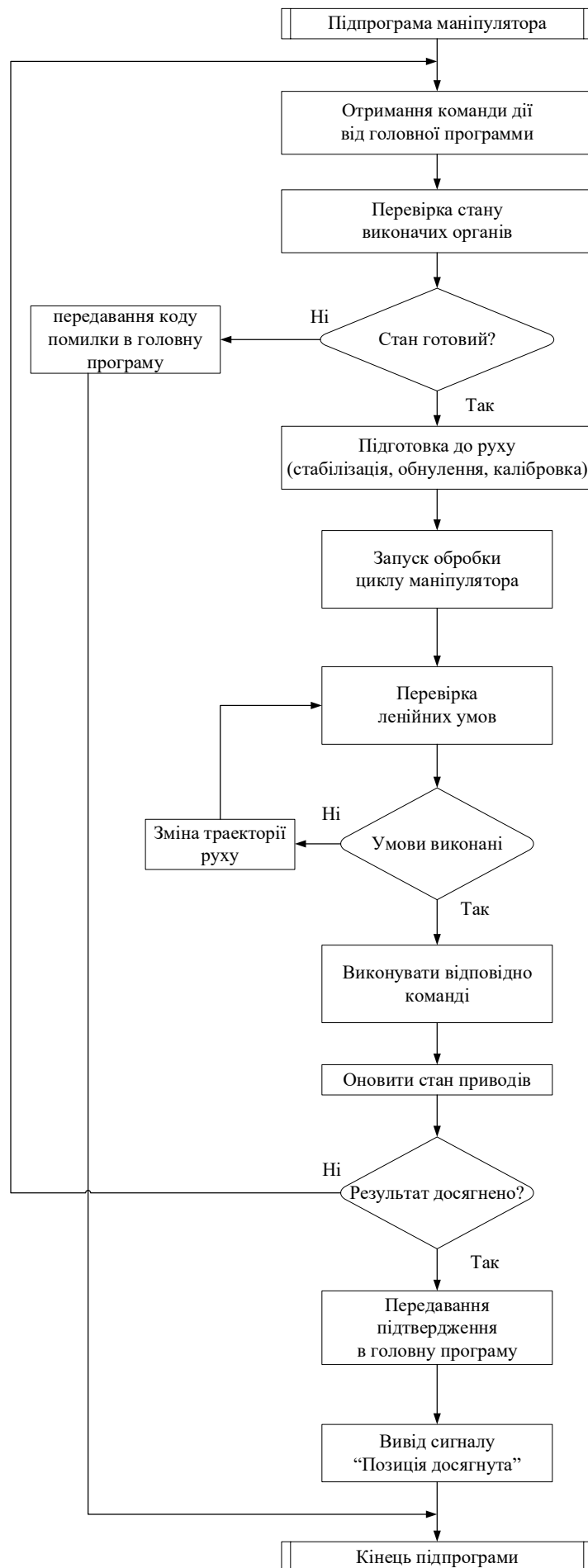


Рисунок 4.6. Алгоритм підпрограми маніпулятора

фокусується на логіці виконання технологічних операцій: захоплення об'єкта, його надійне утримання, транспортування та вивільнення у заданій точці.

Алгоритм реалізує послідовну логіку обробки подій та складається з наступних ключових етапів: Отримання та декодування команди дії: На вхід підпрограми надходить код операції від головного диспетчера завдань. Система розпізнає тип необхідної дії:

- Захват (GRIP): Активація механізму змикання губок.
- Вивільнення (RELEASE): Розмикання захвату.
- Утримання (HOLD): Підтримка стану із заблокованим приводом (для економії енергії та запобігання перегріву реле).

Діагностика стану виконавчих органів: Перед початком будь-якої дії виконується перевірка готовності системи («Стан готовий?»). Алгоритм опитує статус приводів та датчиків. Критично важливою є перевірка того, що маніпулятор знаходиться у нерухомому стані (швидкість дорівнює нулю), оскільки спроба захоплення об'єкта під час руху може призвести до його пошкодження або падіння. Якщо виявлено помилку (наприклад, заклинювання або некоректний статус), генерується код помилки, який передається у головну програму для обробки аварії.

Підготовка до руху (Стабілізація): Оскільки механічна конструкція має певну інерційність та пружність, після зупинки в цільовій точці можуть виникати залишкові коливання. Алгоритм вводить програмну затримку (стабілізацію) перед активацією захвату, щоб гарантувати точне позиціонування інструменту відносно об'єкта. Також на цьому етапі може виконуватися обнулення лічильників таймерів, які відповідають за час роботи приводу захвату.

Запуск циклу обробки та перевірка нелінійних умов: Це інтелектуальне ядро підпрограми. Система аналізує специфічні умови, що можуть вплинути на успішність операції. До нелінійних умов належать:

- Перевірка наявності об'єкта в зоні досяжності (якщо встановлено відповідний оптичний або контактний датчик на вхід RB0).

- Контроль безпечної конфігурації (наприклад, заборона відкриття захвату, якщо маніпулятор знаходиться у зоні транспортування, щоб не впустити вантаж).
- Якщо умови не виконані, алгоритм передбачає гілку корекції («Змінити траєкторію руху»), що дозволяє виконати мікро-переміщення для кращого позиціонування захвату відносно деталі.

Виконання дії (Активація приводів): Мікроконтролер формує керуючий сигнал на відповідний канал драйвера ULN2803. Це призводить до перемикавання реле, яке подає живлення на двигун або електромагніт захвату.

- Враховуючи відсутність датчиків струму, керування зусиллям затискання реалізується часовим методом: двигун вмикається на строго визначений інтервал часу (T_{action} , достатній для повного закриття або відкриття механізму, після чого автоматично вимикається. Це запобігає перегріву обмоток двигуна при роботі "в упор".

Оновлення статусу та перевірка результату: Після завершення активної фази дії алгоритм оновлює внутрішні змінні стану (прапори GRIP_CLOSED або GRIP_OPEN). Виконується фінальна перевірка («Результат досягнуто?»). Якщо операція пройшла успішно, у головну програму відправляється сигнал підтвердження (ACK), що дозволяє переходити до наступного кроку технологічного циклу (наприклад, підйому деталі). У разі невдачі система може ініціювати повторну спробу або зупинити роботу, вимагаючи втручання оператора. Фокусується на логіці виконання технологічних операцій: захоплення об'єкта, його надійне утримання, транспортування та вивільнення у заданій точці.

Алгоритм реалізує послідовну логіку обробки подій та складається з наступних ключових етапів:

Отримання та декодування команди дії: На вхід підпрограми надходить код операції від головного диспетчера завдань. Система розпізнає тип необхідної дії:

- Захват (GRIP): Активація механізму змикання губок.
- Вивільнення (RELEASE): Розмикання захвату.

- Утримання (HOLD): Підтримка стану із заблокованим приводом (для економії енергії та запобігання перегріву реле).

Діагностика стану виконавчих органів: Перед початком будь-якої дії виконується перевірка готовності системи («Стан готовий?»). Алгоритм опитує статус приводів та датчиків. Критично важливою є перевірка того, що маніпулятор знаходиться у нерухомому стані (швидкість дорівнює нулю), оскільки спроба захоплення об'єкта під час руху може призвести до його пошкодження або падіння. Якщо виявлено помилку (наприклад, заклинювання або некоректний статус), генерується код помилки, який передається у головну програму для обробки аварії.

Підготовка до руху (Стабілізація): Оскільки механічна конструкція має певну інерційність та пружність, після зупинки в цільовій точці можуть виникати залишкові коливання. Алгоритм вводить програмну затримку (стабілізацію) перед активацією захвату, щоб гарантувати точне позиціонування інструменту відносно об'єкта. Також на цьому етапі може виконуватися обнулення лічильників таймерів, які відповідають за час роботи приводу захвату.

Запуск циклу обробки та перевірка нелінійних умов: Це інтелектуальне ядро підпрограми. Система аналізує специфічні умови, що можуть вплинути на успішність операції. До нелінійних умов належать:

- Перевірка наявності об'єкта в зоні досяжності (якщо встановлено відповідний оптичний або контактний датчик на вхід RB0).
- Контроль безпечної конфігурації (наприклад, заборона відкриття захвату, якщо маніпулятор знаходиться у зоні транспортування, щоб не впустити вантаж).
- Якщо умови не виконані, алгоритм передбачає гілку корекції («Змінити траєкторію руху»), що дозволяє виконати мікро-переміщення для кращого позиціонування захвату відносно деталі.

Виконання дії (Активація приводів): Мікроконтролер формує керуючий сигнал на відповідний канал драйвера ULN2803. Це призводить до перемикання реле, яке подає живлення на двигун або електромагніт захвату.

Враховуючи відсутність датчиків струму, керування зусиллям затискання реалізується часовим методом: двигун вмикається на строго визначений інтервал часу T_{action} , достатній для повного закриття або відкриття механізму, після чого автоматично вимикається. Це запобігає перегріву обмоток двигуна при роботі "в упор".

Оновлення статусу та перевірка результату: Після завершення активної фази дії алгоритм оновлює внутрішні змінні стану (прапори GRIP_CLOSED або GRIP_OPEN). Виконується фінальна перевірка («Результат досягнуто?»). Якщо операція пройшла успішно, у головну програму відправляється сигнал підтвердження (ACK), що дозволяє переходити до наступного кроку технологічного циклу (наприклад, підйому деталі). У разі невдачі система може ініціювати повторну спробу або зупинити роботу, вимагаючи втручання оператора.

Підпрограма керування циклами завдань

Завершальний рівень логіки керування (рис. 4.7) відповідає за автоматизацію послідовності технологічних операцій. Цей модуль виконує функцію диспетчера, який інтерпретує список завдань і координує виклик інших виконавчих підпрограм (кінематики, позиціонування, маніпулятора) у правильному порядку.

Алгоритм працює за принципом скінченного автомату (State Machine) і складається з наступних етапів:

Ініціалізація та перевірка списку: Робота починається з блоку «Завантаження списку доступних завдань». Система зчитує масив даних з енергонезалежної пам'яті (EEPROM) мікроконтролера або отримує його через інтерфейс RS-485. Одразу виконується логічна перевірка: «Список порожній?».

- Якщо ТАК: Це означає, що програма не задана. Система виводить на дисплей повідомлення «Показання команди 'Введіть завдання'», спонукаючи оператора до дій, і завершує роботу підпрограми, переходячи в режим очікування введення.

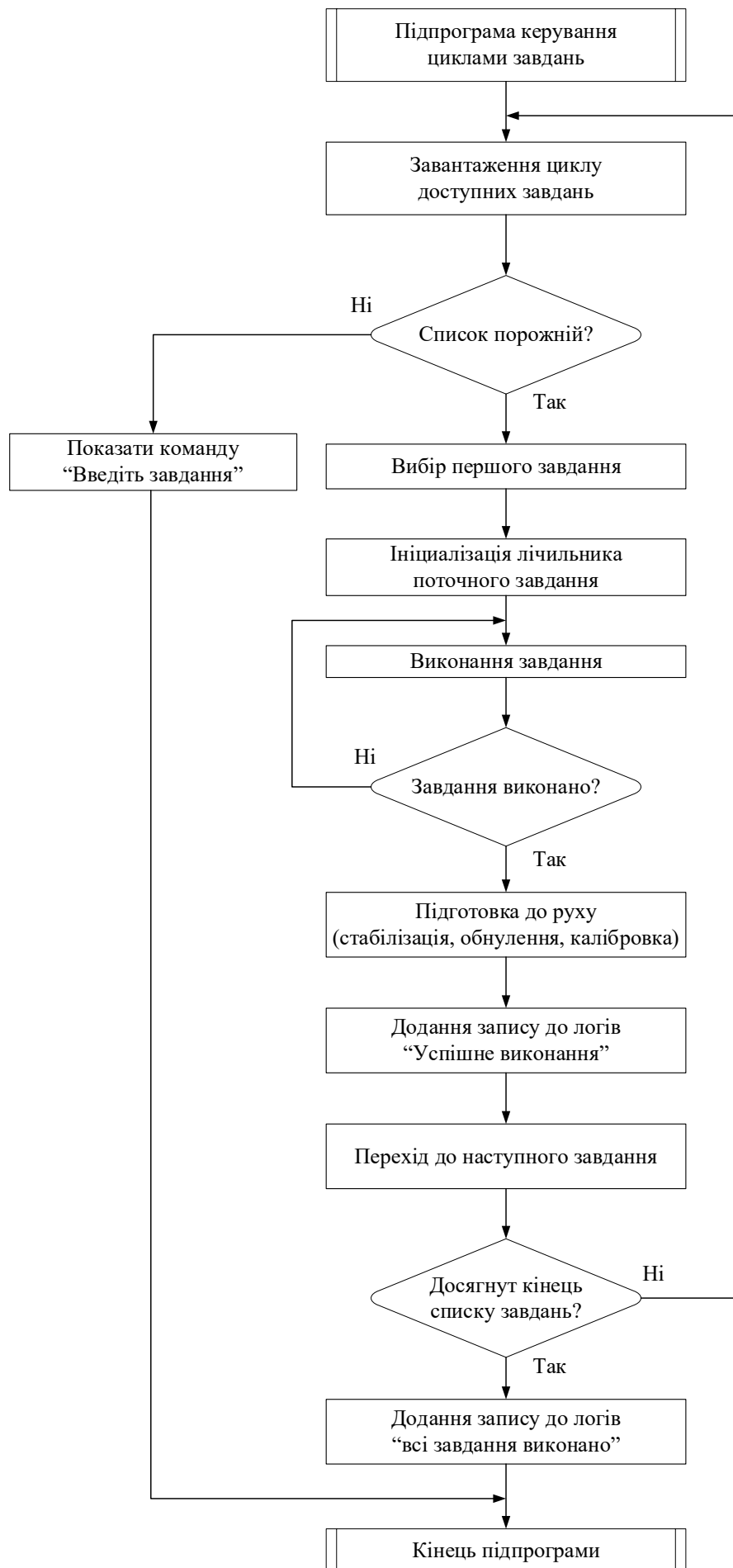


Рисунок 4.7. Алгоритм підпрограми керування циклами завдань

- Якщо НІ: Система переходить до автоматичного виконання.

Підготовка до циклу: Виконується блок «Вибір першого завдання» та «Ініціалізація лічильника поточного завдання». Програмний вказівник (Pointer) встановлюється на нульовий індекс масиву завдань ($i = 0$), готуючи систему до послідовної обробки.

Виконання операції: Центральним елементом алгоритму є блок «Виконання завдання». Диспетчер зчитує тип поточної команди (наприклад, переміщення в точку X,Y або замикання захвату) і передає керування відповідній функціональній підпрограмі.

Моніторинг статусу (Цикл очікування): Після запуску виконавчих механізмів диспетчер переходить у цикл опитування статусу: «Завдання виконано?».

- Якщо НІ: Система залишається в цьому блоці (або повертається до виконання), постійно перевіряючи прапори стану від підпрограм нижнього рівня (наприклад, очікує сигнал від кінцевика або завершення таймера захвату).
- Якщо ТАК: Це означає, що фізична дія завершена успішно, і можна переходити до фіксації результату.

Логування та звітність: Згідно з блок-схемою, після успішного завершення етапу виконується «Додання запису до логів 'успішне виконання'». Ця інформація може зберігатися в журналі подій або виводитися на LCD-дисплей для інформування оператора про прогрес (наприклад, "Крок 3/10: ОК").

Циклічний перехід та завершення: Виконується блок «Перехід до наступного завдання» (інкремент лічильника $i = i + 1$). Далі слідує умова: «Досягнуто кінець списку завдань?».

- Якщо НІ: Алгоритм повертається по петлі зворотного зв'язку до блоку виконання, завантажуючи параметри наступного кроку.
- Якщо ТАК: Це свідчить про повне завершення технологічного циклу.

Система виконує фінальну дію — «Додання запису до логів 'всі завдання

виконано'», після чого підпрограма завершує свою роботу, переводячи маніпулятор у вихідне положення або режим очікування.

4.3 Код програми

У цьому підрозділі представлено програмну реалізацію системи керування роботом-маніпулятором МП-9С. Враховуючи вимоги до надійності промислової автоматизації, наочності логіки керування та необхідності простої інтеграції в існуючі технологічні процеси, для програмування мікроконтролера PIC16F1937 було обрано мову Ladder Diagram (LD) — мову релейно-контактних схем, що є стандартом IEC 61131-3.

Такий підхід дозволив реалізувати архітектуру Soft-PLC, де мікроконтролер функціонує як повноцінний промисловий логічний контролер. Програма виконується циклічно (Scan Cycle), зчитуючи стан фізичних входів (кнопок, кінцевиків, сигналів датчиків), виконуючи логічні операції та оновлюючи стани виходів (керування ULN2803, індикація).

Структура та лістинг програми у якого код побудовано у вигляді послідовності логічних кіл (Rungs). Кожне коло описує умову виконання певної дії. Структура програми є модульною і включає блоки ініціалізації, діагностики, основного циклу руху, обробки аварій та комунікації.

Нижче наведено повний лістинг логіки керування, реалізованої в контролері.

```
| // ПЛК — основний цикл керування маніпулятором
|----[ NO ] Power_On_Reset -----( OUT ) SystemInit_Start
|----[ NO ] SystemInit_Start -----( OUT ) Init_Run
| // Ініціалізація системи
|----[ NO ] Init_Run -----( OUT ) Init_Clear_Registers
|----[ NO ] Init_Run -----( OUT ) Init_Config_GPIO
|----[ NO ] Init_Run -----( OUT ) Init_Timers
|----[ NO ] Init_Run -----( OUT ) Init_PWM
```

```

|----[ NO ] Init_Run -----( OUT ) Init_UART_RS485
|----[ NO ] Init_Run -----( OUT ) Init_Sensors_Check
|----[ NO ] Init_Run -----( OUT ) Init_Drives_Check
| // Самодіагностика після ініціалізації
|----[ NO ] Init_Sensors_Check -----( OUT ) SelfDiagnostics_Start
|----[ NO ] Init_Drives_Check -----( OUT ) SelfDiagnostics_Start
|----[ NO ] SelfDiagnostics_Start -----( OUT ) Check_Encoders
|----[ NO ] SelfDiagnostics_Start -----( OUT ) Check_Endstops
|----[ NO ] SelfDiagnostics_Start -----( OUT ) Check_Power
|----[ NO ] SelfDiagnostics_Start -----( OUT ) Check_RS485_Link
| // Перевірка готовності системи
|----[ NO ] Check_Encoders -----( OUT ) ENC_OK
|----[ NO ] Check_Endstops -----( OUT ) ENDSTOPS_OK
|----[ NO ] Check_Power -----( OUT ) POWER_OK
|----[ NO ] Check_RS485_Link -----( OUT ) RS485_OK
|----[ NO ] ENC_OK & ENDSTOPS_OK & POWER_OK & RS485_OK ----( OUT )
INIT_OK
|----[ NO ] INIT_OK ----- ( OUT ) SYSTEM_READY
| // Запуск головного циклу після ініціалізації
|----[ NO ] SYSTEM_READY -----( OUT ) Main_Loop_Active
| // Завантаження траєкторії
|----[ NO ] Main_Loop_Active -----( OUT ) Trajectory_Load_Request
|----[ NO ] Trajectory_Load_Request -----( OUT ) LoadTrajectory_Start
|----[ NO ] LoadTrajectory_Start -----( OUT ) Parse_Trajectory
|----[ NO ] LoadTrajectory_Start -----( OUT ) Traj_Buffer_Write
|----[ NO ] Parse_Trajectory -----( OUT ) Traj_Valid
|----[ NO ] Traj_Valid -----( OUT ) Trajectory_Loaded
| // Цикл обробки траєкторій
|----[ NO ] Trajectory_Loaded -----( OUT ) KinematicsCycle_Active
|----[ NO ] KinematicsCycle_Active -----( OUT ) Read_Current_Position

```

```

|----[ NO ] Read_Current_Position -----( OUT ) Read_Encoders
|----[ NO ] Read_Current_Position -----( OUT ) Read_ADC_Sensors
|----[ NO ] Read_Encoders -----( OUT ) POS_CURRENT_UPDATED
|----[ NO ] Read_ADC_Sensors -----( OUT ) SENSORS_UPDATED
|----[ NO ] POS_CURRENT_UPDATED & SENSORS_UPDATED ----( OUT )
Get_Target_Point
|----[ NO ] Get_Target_Point -----( OUT ) KinematicsSolver_Start
| // Кінематика та перевірки
|----[ NO ] KinematicsSolver_Start -----( OUT ) IK_Calc
|----[ NO ] IK_Calc -----( OUT ) IK_Solution_Found
|----[ NO ] IK_Solution_Found -----( OUT ) Check_Dynamic_Limits
|----[ NO ] Check_Dynamic_Limits -----( OUT ) LIMITS_OK
|----[ NO ] LIMITS_OK -----( OUT ) Trajectory_Generated
|----[ NO ] Trajectory_Generated -----( OUT ) CollisionCheck_Start
|----[ NO ] CollisionCheck_Start -----( OUT ) Sensor_Proximity_Read
|----[ NO ] Sensor_Proximity_Read -----( OUT ) NO_OBSTACLE
|----[ NO ] NO_OBSTACLE -----( OUT ) AllowMotion
|----[ NO ] AllowMotion -----( OUT ) ExecuteMotion_Start
| // Виконання руху
|----[ NO ] ExecuteMotion_Start -----( OUT ) MotionControl_Start
|----[ NO ] MotionControl_Start -----( OUT ) PWM_Update
|----[ NO ] MotionControl_Start -----( OUT ) PID_Update
|----[ NO ] PWM_Update -----( OUT ) Drive_Command_Out
|----[ NO ] Drive_Command_Out -----( OUT ) ULN2803_Drive
|----[ NO ] ULN2803_Drive -----( OUT ) Motor_Actuate
|----[ NO ] Motor_Actuate -----( OUT ) Motion_In_Progress
| // Перевірка досягнення позиції
|----[ NO ] Motion_In_Progress -----( OUT ) Check_Position_Error
|----[ NO ] Check_Position_Error -----( OUT ) Error_Within_Tolerance
|----[ NO ] Error_Within_Tolerance -----( OUT ) PositionReached_Flag

```

```

|----[ NO ] PositionReached_Flag -----( OUT ) ExecuteManipulatorActions
| // Дії маніпулятора
|----[ NO ] ExecuteManipulatorActions -----( OUT ) Gripper_Close
|----[ NO ] ExecuteManipulatorActions -----( OUT ) Gripper_Open
|----[ NO ] ExecuteManipulatorActions -----( OUT ) Hold_Object_Check
|----[ NO ] Hold_Object_Check -----( OUT ) GRIP_OK
|----[ NO ] GRIP_OK -----( OUT ) Task_Step_Complete
|----[ NO ] Task_Step_Complete ----- ( OUT ) CycleManager_Invoke
| // Керування циклами завдань
|----[ NO ] CycleManager_Invoke -----( OUT ) Get_Next_Task
|----[ NO ] Get_Next_Task -----( OUT ) Next_Task_Available
|----[ NO ] Next_Task_Available -----( OUT ) Select_Next_Task
|----[ NO ] Select_Next_Task -----( OUT ) Update_Task_Counter
|----[ NO ] Update_Task_Counter -----( OUT ) Task_Parameters_Load
|----[ NO ] Task_Parameters_Load -----( OUT ) CycleManager_Return
|----[ NO ] CycleManager_Return -----( OUT ) KinematicsCycle_Active
| // Обробка помилок та аварій
|----[ NO ] ANY_ERROR -----( OUT ) EmergencyStop_Activate
|----[ NO ] EmergencyStop_Activate -----( OUT ) Stop_All_Motors
|----[ NO ] Stop_All_Motors -----( OUT ) Disable_Drives
|----[ NO ] Disable_Drives -----( OUT ) Log_Emergency_Event
|----[ NO ] Log_Emergency_Event ----- ( OUT ) Notify_PC
|----[ NO ] Notify_PC -----( OUT ) RS485_Alarm_Send
| // Логування та моніторинг
|----[ NO ] Main_Loop_Active -----( OUT ) Event_Logger_Update
|----[ NO ] Event_Logger_Update -----( OUT ) Log_Position
|----[ NO ] Event_Logger_Update -----( OUT ) Log_Sensor_States
|----[ NO ] Event_Logger_Update -----( OUT ) Log_Drive_States
|----[ NO ] Event_Logger_Update -----( OUT ) Log_Communications
| // Обробка комунікацій

```

|----[NO] RS485_RX -----(OUT) RS485_Message_Received
|----[NO] RS485_Message_Received -----(OUT) Parse_Command
|----[NO] Parse_Command -----(OUT) Command_Valid
|----[NO] Command_Valid -----(OUT) Apply_Command
|----[NO] Apply_Command -----(OUT) Update_Trajectory_Buffer
|----[NO] Apply_Command -----(OUT) Override_Task_List

| // *Захист та безпека*

|----[NO] E_STOP_PRESSED -----(OUT) EmergencyStop_Activate
|----[NO] Overcurrent_Detected -----(OUT) Reduce_PWM
|----[NO] Reduce_PWM -----(OUT) Notify_Maintenance
|----[NO] Overtemperature -----(OUT) Pause_Operation
|----[NO] Pause_Operation -----(OUT) Cooldown_Timer_Start
|----[NO] Cooldown_Timer_Expired -----(OUT) Resume_Operation

| // *Повторні спроби ініціалізації при помилці*

|----[NO] INIT_FAIL & Retry_Count < 3 ----(OUT) Retry_Init
|----[NO] Retry_Init -----(OUT) Init_Run
|----[NO] INIT_FAIL & Retry_Count >=3 ----(OUT) SYSTEM_LOCK
|----[NO] SYSTEM_LOCK -----(OUT) Notify_Admin

| // *Синхронізація та часові події*

|----[NO] Timer_1ms_Tick -----(OUT) System_Tick
|----[NO] System_Tick -----(OUT) Heartbeat_LED_Toggle
|----[NO] System_Tick -----(OUT) Watchdog_Reset
|----[NO] Watchdog_Reset -----(OUT) Watchdog_Service

| // *Аварійне журналювання*

|----[NO] EmergencyStop_Activate -----(OUT) Save_Error_State
|----[NO] Save_Error_State -----(OUT) Flash_Write_Error
|----[NO] Flash_Write_Error -----(OUT) Halt_System

| // *Тестові режими та сервіс*

|----[NO] Service_Mode_Active -----(OUT) Manual_Control_Enable
|----[NO] Manual_Control_Enable -----(OUT) Bypass_Safety_Limits

|----[NO] Manual_Control_Enable -----(OUT) Direct_Motor_Drive
 | *// Оновлення параметрів ПІД*

|----[NO] PID_Tuning_Mode -----(OUT) PID_Update_Request
 |----[NO] PID_Update_Request ----- (OUT) PID_Write_to_EEPROM
 |----[NO] PID_Write_to_EEPROM -----(OUT) PID_Apply_New
 | *// Керування індикацією*

|----[NO] SYSTEM_READY -----(OUT) LCD_Display_Update
 |----[NO] LCD_Display_Update ----- (OUT) Display_Position
 |----[NO] LCD_Display_Update ----- (OUT) Display_Status
 |----[NO] LCD_Display_Update ----- (OUT) Display_Errors
 | *// Точки відмовостійкості — фіксація стану приводів*

|----[NO] Motion_Control_Error -----(OUT) Motor_Freeze
 |----[NO] Motor_Freeze -----(OUT) Engage_Brakes
 |----[NO] Engage_Brakes -----(OUT) Log_Motor_State
 | *// Плавне завершення циклу*

|----[NO] Main_Loop_Active -----(OUT) Final_System_Check
 |----[NO] Final_System_Check ----- (OUT) Update_Watchdog_Counter
 |----[NO] Final_System_Check ----- (OUT) Save_Logs_to_Flash
 |----[NO] Final_System_Check ----- (OUT) Powerdown_Peripherals
 |----[NO] Powerdown_Peripherals -----(OUT) Enter_LowPower
 | *// Резервне відновлення задач*

|----[NO] Task_List_Empty & Autosave_Present --(OUT) Restore_Task_List
 |----[NO] Restore_Task_List -----(OUT) Task_List_Loaded
 |----[NO] Task_List_Loaded -----(OUT) CycleManager_Invoke
 | *// Захист при втраті зв'язку*

|----[NO] RS485_Link_Lost -----(OUT) Hold_Position
 |----[NO] Hold_Position -----(OUT) Reduce_Motion_Speed
 |----[NO] Bring_To_Safe_State ----- (OUT) Disable_New_Motions
 | *// Періодична калибрування*

|----[NO] Calibration_Timer_Elapsed -----(OUT) Run_Calibration

```

|----[ NO ] Run_Calibration -----( OUT ) Calibrate_Encoders
|----[ NO ] Calibrate_Encoders ----- ( OUT ) Cal_Home_Axes
|----[ NO ] Cal_Home_Axes -----( OUT ) Calibration_Complete
| // Зменшення зносу — чергування режимів
|----[ NO ] Operating_Hours > Threshold ----( OUT ) Maintenance_Alert
|----[ NO ] Maintenance_Alert -----( OUT ) Schedule_Service
| // Захист даних при аварії
|----[ NO ] Any_Fatal_Error -----( OUT ) Secure_Data_Flash
|----[ NO ] Secure_Data_Flash -----( OUT ) Disable_Communications
|----[ NO ] Disable_Communications -----( OUT ) System_Halt
| // Завершення та вихід у сплячий режим
|----[ NO ] Idle_Mode_Requested -----( OUT ) Prepare_Sleep
|----[ NO ] Prepare_Sleep -----( OUT ) Save_State
|----[ NO ] Save_State -----( OUT ) Turn_Off_Peripherals
|----[ NO ] Turn_Off_Peripherals -----( OUT ) Enter_Sleep_Mode
| // Логування завершення сеансу
|----[ NO ] Enter_Sleep_Mode -----( OUT ) Log_Session_End
|----[ NO ] Log_Session_End -----( OUT ) Power_Off_Sequence
|----[ NO ] Power_Off_Sequence ----- ( OUT ) System_Off
| // Оновлення ПЗ (оновлення прошивки)
|----[ NO ] Firmware_Update_Request -----( OUT ) Verify_Update_Package
|----[ NO ] Verify_Update_Package -----( OUT ) Backup_Current_FW
|----[ NO ] Backup_Current_FW -----( OUT ) Apply_Firmware_Update
|----[ NO ] Apply_Firmware_Update -----( OUT ) Reboot_System
|----[ NO ] Reboot_System -----( OUT ) SystemInit_Start
| // ПЛК — основний цикл керування маніпулятором
|----[ NO ] Power_On_Reset -----( OUT ) SystemInit_Start
|----[ NO ] SystemInit_Start -----( OUT ) Init_Run
| // Ініціалізація системи
|----[ NO ] Init_Run -----( OUT ) Init_Clear_Registers

```

```

|----[ NO ] Init_Run -----( OUT ) Init_Config_GPIO
|----[ NO ] Init_Run -----( OUT ) Init_Timers
|----[ NO ] Init_Run -----( OUT ) Init_PWM
|----[ NO ] Init_Run -----( OUT ) Init_UART_RS485
|----[ NO ] Init_Run -----( OUT ) Init_Sensors_Check
|----[ NO ] Init_Run -----( OUT ) Init_Drives_Check
| // Самодіагностика після ініціалізації
|----[ NO ] Init_Sensors_Check -----( OUT ) SelfDiagnostics_Start
|----[ NO ] Init_Drives_Check -----( OUT ) SelfDiagnostics_Start
|----[ NO ] SelfDiagnostics_Start -----( OUT ) Check_Encoders
|----[ NO ] SelfDiagnostics_Start -----( OUT ) Check_Endstops
|----[ NO ] SelfDiagnostics_Start -----( OUT ) Check_Power
|----[ NO ] SelfDiagnostics_Start -----( OUT ) Check_RS485_Link
| // Перевірка готовності системи
|----[ NO ] Check_Encoders -----( OUT ) ENC_OK
|----[ NO ] Check_Endstops -----( OUT ) ENDSTOPS_OK
|----[ NO ] Check_Power -----( OUT ) POWER_OK
|----[ NO ] Check_RS485_Link -----( OUT ) RS485_OK
|----[ NO ] ENC_OK & ENDSTOPS_OK & POWER_OK & RS485_OK ----( OUT )
INIT_OK
|----[ NO ] INIT_OK ----- ( OUT ) SYSTEM_READY
| // Запуск головного циклу після ініціалізації
|----[ NO ] SYSTEM_READY -----( OUT ) Main_Loop_Active
| // Завантаження траєкторії
|----[ NO ] Main_Loop_Active -----( OUT ) Trajectory_Load_Request
|----[ NO ] Trajectory_Load_Request -----( OUT ) LoadTrajectory_Start
|----[ NO ] LoadTrajectory_Start -----( OUT ) Parse_Trajectory
|----[ NO ] LoadTrajectory_Start -----( OUT ) Traj_Buffer_Write
|----[ NO ] Parse_Trajectory -----( OUT ) Traj_Valid
|----[ NO ] Traj_Valid -----( OUT ) Trajectory_Loaded

```

| // Цикл обробки траєкторій

|----[NO] Trajectory_Loaded -----(OUT) KinematicsCycle_Active

|----[NO] KinematicsCycle_Active -----(OUT) Read_Current_Position

|----[NO] Read_Current_Position -----(OUT) Read_Encoders

|----[NO] Read_Current_Position -----(OUT) Read_ADC_Sensors

|----[NO] Read_Encoders -----(OUT) POS_CURRENT_UPDATED

|----[NO] Read_ADC_Sensors -----(OUT) SENSORS_UPDATED

|----[NO] POS_CURRENT_UPDATED & SENSORS_UPDATED ----(OUT)

Get_Target_Point

|----[NO] Get_Target_Point -----(OUT) KinematicsSolver_Start

| // Кінематика та перевірки

|----[NO] KinematicsSolver_Start -----(OUT) IK_Calc

|----[NO] IK_Calc -----(OUT) IK_Solution_Found

|----[NO] IK_Solution_Found -----(OUT) Check_Dynamic_Limits

|----[NO] Check_Dynamic_Limits -----(OUT) LIMITS_OK

|----[NO] LIMITS_OK -----(OUT) Trajectory_Generated

|----[NO] Trajectory_Generated -----(OUT) CollisionCheck_Start

|----[NO] CollisionCheck_Start -----(OUT) Sensor_Proximity_Read

|----[NO] Sensor_Proximity_Read -----(OUT) NO_OBSTACLE

|----[NO] NO_OBSTACLE -----(OUT) AllowMotion

|----[NO] AllowMotion -----(OUT) ExecuteMotion_Start

| // Виконання руху

|----[NO] ExecuteMotion_Start -----(OUT) MotionControl_Start

|----[NO] MotionControl_Start -----(OUT) PWM_Update

|----[NO] MotionControl_Start -----(OUT) PID_Update

|----[NO] PWM_Update -----(OUT) Drive_Command_Out

|----[NO] Drive_Command_Out -----(OUT) ULN2803_Drive

|----[NO] ULN2803_Drive -----(OUT) Motor_Actuate

|----[NO] Motor_Actuate -----(OUT) Motion_In_Progress

| // Перевірка досягнення позиції

|----[NO] Motion_In_Progress ----- (OUT) Check_Position_Error
|----[NO] Check_Position_Error -----(OUT) Error_Within_Tolerance
|----[NO] Error_Within_Tolerance -----(OUT) PositionReached_Flag
|----[NO] PositionReached_Flag -----(OUT) ExecuteManipulatorActions

| // *Дії маніпулятора*

|----[NO] ExecuteManipulatorActions -----(OUT) Gripper_Close
|----[NO] ExecuteManipulatorActions -----(OUT) Gripper_Open
|----[NO] ExecuteManipulatorActions -----(OUT) Hold_Object_Check
|----[NO] Hold_Object_Check -----(OUT) GRIP_OK
|----[NO] GRIP_OK -----(OUT) Task_Step_Complete
|----[NO] Task_Step_Complete ----- (OUT) CycleManager_Invoke

| // *Керування циклами завдань*

|----[NO] CycleManager_Invoke -----(OUT) Get_Next_Task
|----[NO] Get_Next_Task -----(OUT) Next_Task_Available
|----[NO] Next_Task_Available -----(OUT) Select_Next_Task
|----[NO] Select_Next_Task -----(OUT) Update_Task_Counter
|----[NO] Update_Task_Counter -----(OUT) Task_Parameters_Load
|----[NO] Task_Parameters_Load -----(OUT) CycleManager_Return
|----[NO] CycleManager_Return -----(OUT) KinematicsCycle_Active

| // *Обробка помилок та аварій*

|----[NO] ANY_ERROR -----(OUT) EmergencyStop_Activate
|----[NO] EmergencyStop_Activate -----(OUT) Stop_All_Motors
|----[NO] Stop_All_Motors -----(OUT) Disable_Drives
|----[NO] Disable_Drives -----(OUT) Log_Emergency_Event
|----[NO] Log_Emergency_Event ----- (OUT) Notify_PC
|----[NO] Notify_PC -----(OUT) RS485_Alarm_Send

| // *Логування та моніторинг*

|----[NO] Main_Loop_Active -----(OUT) Event_Logger_Update
|----[NO] Event_Logger_Update -----(OUT) Log_Position
|----[NO] Event_Logger_Update -----(OUT) Log_Sensor_States

|----[NO] Event_Logger_Update -----(OUT) Log_Drive_States
|----[NO] Event_Logger_Update -----(OUT) Log_Communications
| *// Обробка комунікацій*
|----[NO] RS485_RX -----(OUT) RS485_Message_Received
|----[NO] RS485_Message_Received -----(OUT) Parse_Command
|----[NO] Parse_Command -----(OUT) Command_Valid
|----[NO] Command_Valid -----(OUT) Apply_Command
|----[NO] Apply_Command -----(OUT) Update_Trajectory_Buffer
|----[NO] Apply_Command -----(OUT) Override_Task_List
| *// Захист та безпека*
|----[NO] E_STOP_PRESSED -----(OUT) EmergencyStop_Activate
|----[NO] Overcurrent_Detected -----(OUT) Reduce_PWM
|----[NO] Reduce_PWM -----(OUT) Notify_Maintenance
|----[NO] Overtemperature -----(OUT) Pause_Operation
|----[NO] Pause_Operation -----(OUT) Cooldown_Timer_Start
|----[NO] Cooldown_Timer_Expired -----(OUT) Resume_Operation
| *// Повторні спроби ініціалізації при помилці*
|----[NO] INIT_FAIL & Retry_Count < 3 ----(OUT) Retry_Init
|----[NO] Retry_Init -----(OUT) Init_Run
|----[NO] INIT_FAIL & Retry_Count >=3 ----(OUT) SYSTEM_LOCK
|----[NO] SYSTEM_LOCK -----(OUT) Notify_Admin
| *// Синхронізація та часові події*
|----[NO] Timer_1ms_Tick -----(OUT) System_Tick
|----[NO] System_Tick -----(OUT) Heartbeat_LED_Toggle
|----[NO] System_Tick -----(OUT) Watchdog_Reset
|----[NO] Watchdog_Reset -----(OUT) Watchdog_Service
| *// Аварійне журналювання*
|----[NO] EmergencyStop_Activate -----(OUT) Save_Error_State
|----[NO] Save_Error_State -----(OUT) Flash_Write_Error
|----[NO] Flash_Write_Error -----(OUT) Halt_System

| // *Тестові режими та сервіс*

|----[NO] Service_Mode_Active -----(OUT) Manual_Control_Enable

|----[NO] Manual_Control_Enable -----(OUT) Bypass_Safety_Limits

|----[NO] Manual_Control_Enable -----(OUT) Direct_Motor_Drive

| // *Оновлення параметрів ПІД*

|----[NO] PID_Tuning_Mode -----(OUT) PID_Update_Request

|----[NO] PID_Update_Request ----- (OUT) PID_Write_to_EEPROM

|----[NO] PID_Write_to_EEPROM -----(OUT) PID_Apply_New

| // *Керування індикацією*

|----[NO] SYSTEM_READY -----(OUT) LCD_Display_Update

|----[NO] LCD_Display_Update ----- (OUT) Display_Position

|----[NO] LCD_Display_Update ----- (OUT) Display_Status

|----[NO] LCD_Display_Update ----- (OUT) Display_Errors

| // *Точки відмовостійкості — фіксація стану приводів*

|----[NO] Motion_Control_Error -----(OUT) Motor_Freeze

|----[NO] Motor_Freeze -----(OUT) Engage_Brakes

|----[NO] Engage_Brakes -----(OUT) Log_Motor_State

| // *Плавне завершення циклу*

|----[NO] Main_Loop_Active -----(OUT) Final_System_Check

|----[NO] Final_System_Check ----- (OUT) Update_Watchdog_Counter

|----[NO] Final_System_Check ----- (OUT) Save_Logs_to_Flash

|----[NO] Final_System_Check ----- (OUT) Powerdown_Peripherals

|----[NO] Powerdown_Peripherals -----(OUT) Enter_LowPower

| // *Резервне відновлення задач*

|----[NO] Task_List_Empty & Autosave_Present --(OUT) Restore_Task_List

|----[NO] Restore_Task_List -----(OUT) Task_List_Loaded

|----[NO] Task_List_Loaded -----(OUT) CycleManager_Invoke

| // *Захист при втраті зв'язку*

|----[NO] RS485_Link_Lost -----(OUT) Hold_Position

|----[NO] Hold_Position -----(OUT) Reduce_Motion_Speed

|----[NO] Bring_To_Safe_State ----- (OUT) Disable_New_Motions
| *// Періодична калібрування*
|----[NO] Calibration_Timer_Elapsed -----(OUT) Run_Calibration
|----[NO] Run_Calibration -----(OUT) Calibrate_Encoders
|----[NO] Calibrate_Encoders ----- (OUT) Cal_Home_Axes
|----[NO] Cal_Home_Axes -----(OUT) Calibration_Complete
| *// Зменшення зносу — чергування режимів*
|----[NO] Operating_Hours > Threshold ----(OUT) Maintenance_Alert
|----[NO] Maintenance_Alert -----(OUT) Schedule_Service
| *// Захист даних при аварії*
|----[NO] Any_Fatal_Error -----(OUT) Secure_Data_Flash
|----[NO] Secure_Data_Flash -----(OUT) Disable_Communications
|----[NO] Disable_Communications -----(OUT) System_Halt
| *// Завершення та вихід у сплячий режим*
|----[NO] Idle_Mode_Requested -----(OUT) Prepare_Sleep
|----[NO] Prepare_Sleep -----(OUT) Save_State
|----[NO] Save_State -----(OUT) Turn_Off_Peripherals
|----[NO] Turn_Off_Peripherals -----(OUT) Enter_Sleep_Mode
| *// Логування завершення сеансу*
|----[NO] Enter_Sleep_Mode -----(OUT) Log_Session_End
|----[NO] Log_Session_End -----(OUT) Power_Off_Sequence
|----[NO] Power_Off_Sequence ----- (OUT) System_Off
| *// Оновлення ПЗ (оновлення прошивки)*
|----[NO] Firmware_Update_Request -----(OUT) Verify_Update_Package
|----[NO] Verify_Update_Package -----(OUT) Backup_Current_FW
|----[NO] Backup_Current_FW -----(OUT) Apply_Firmware_Update
|----[NO] Apply_Firmware_Update -----(OUT) Reboot_System
|----[NO] Reboot_System -----(OUT) SystemInit_Start
| *// Кінець програми*

Аналіз функціональних блоків коду. Логіка програми базується на послідовному виконанні умовних переходів. Розглянемо ключові блоки:

Ініціалізація та Самодіагностика: Використовується системний біт `Power_On_Reset`, який активує ланцюжок ініціалізації апаратних ресурсів (`Init_Config_GPIO`, `Init_Timers`). Після налаштування запускається блок `SelfDiagnostics_Start`, який опитує датчики. Головний дозвіл на роботу (`SYSTEM_READY`) формується лише за умови логічного множення ("І") всіх сигналів справності: `ENC_OK` (енкодери), `ENDSTOPS_OK` (кінцевики) та `RS485_OK` (зв'язок).

Основний цикл та Кінематика: За наявності дозволу `Main_Loop_Active` контролер переходить до завантаження траєкторії. Блок `Read_Current_Position` зчитує дані з АЦП та енкодерів. Логіка `KinematicsSolver_Start` виконує порівняння координат і, якщо шлях вільний (`NO_OBSTACLE`), активує дозвіл на рух `AllowMotion`.

Керування приводами (Motion Control): Цей блок (`MotionControl_Start`) перетворює логічні команди у фізичні сигнали для мікросхеми `ULN2803`. Використовуються змінні `PWM_Update` та `PID_Update` для реалізації часових алгоритмів керування реле (Time-proportional control) та корекції позиції. Кінцевий вихід `ULN2803_Drive` безпосередньо комутує обмотки реле.

Безпека та Аварії: Реалізовано пріоритетну обробку аварій. Сигнал `ANY_ERROR` або натискання кнопки `E_STOP_PRESSED` миттєво активує ланцюг `EmergencyStop_Activate`, який блокує всі виходи (`Disable_Drives`) та записує подію в лог.

Інтерфейс користувача та верифікація роботи. Для перевірки коректності функціонування розробленого програмного забезпечення (Soft-PLC) та забезпечення зручної взаємодії оператора з системою керування було розроблено діагностичний графічний інтерфейс (HMI). Інтерфейс отримує телеметричні дані від мікроконтролера `PIC16F1937` через канал зв'язку RS-485 і візуалізує стани внутрішніх регістрів, входів/виходів та змінних процесу в реальному часі.

Пульт керування МП-9С

Блок зв'язку (RS-485)

Порт: COM3 [9600, 8, N, 1]

З'єднання з ПЛК: [OK]

Пінги (Ping): 4ms

СИСТЕМНИЙ СТАТУС

Живлення: [OK]

Аварія (E-Stop): [OFF]

ПЕРЕВІРКА ПЕРИФЕРІЇ

Модуль	Статус	Значення / Код
Драйвер ULN2803	ГОТОВИЙ	Всі реле OFF
Датчик кута J1	НОРМА	ADC: 512 (2.5V)
Датчик кута J2	НОРМА	ADC: 128 (0.6V)
Датчик кута J3	НОРМА	ADC: 715 (3.5V)
Датчик Захвату	НОРМА	ADC: 1023 (5.0V)
Ліміти руху	АКТИВНІ	Hard Limits Check

СТАН ПОЧАТКОВОЇ КАЛІБРОВКИ

Нульові кути

J1

J2

J3

Захват

Встановлено базовий стан

"ІНІЦІАЛІЗАЦІЮ ЗАВЕРШЕНО"

ПРОДОВЖИТИ

ПЕРЕЗАПУСК

Рисунок 4.8. Діалогове вікно «Результат ініціалізації»

Діалогове вікно «Результат діагностики системи» Після подачі живлення та виконання процедури апаратного скидання система переходить у режим ініціалізації. Результати виконання підпрограми SystemInit відображаються у першому діагностичному вікні (рис. 4.8).

Інтерфейс надає вичерпну інформацію про готовність апаратних засобів:

Блок зв'язку (RS-485): Відображає параметри налаштування порту (COM3, швидкість 9600 біт/с, 8 біт даних, без парності). Статус «[OK]» та низький час відгуку (Ping: 4ms) підтверджують стабільність фізичного з'єднання між ПК та ПЛК.

Системний статус: Індикатори підтверджують наявність стабільного живлення та відсутність сигналу аварійної зупинки (E-Stop: [OFF]), що є обов'язковою умовою для розблокування приводів. Перевірка периферії: Таблиця деталізує стан кожного модуля:

- Драйвер ULN2803: Має статус «ГОТОВИЙ», при цьому зафіксовано безпечний стан виходів («Всі реле OFF»), що гарантує нерухомість маніпулятора при старті.
- Аналогові датчики (J1–J3, Захват): Відображаються «сирі» значення АЦП та відповідні напруги (наприклад, для J3: ADC 715 / 3.5V), що свідчить про справність потенціометрів та входних кіл мікроконтролера.
- Ліміти руху: Статус «АКТИВНИ» підтверджує, що програмні обмежувачі робочої зони увімкнено.
- Калібрування: Блок «Стан початкової калібровки» інформує про успішне знаходження нульових положень для всіх осей та встановлення захвату в базовий стан.
- Зелений індикатор «Ініціалізацію завершено» дозволяє оператору перейти до автоматичного режиму роботи.

Діалогове вікно «Моніторинг автоматичного циклу» Під час виконання технологічної програми оператор використовує вікно моніторингу (рис. 4.9), яке візуалізує динаміку процесу керування.

Моніторинг виконання автоматичного циклу: Маніпулятор МП-9С

ПОТОЧНЕ ЗАВДАННЯ

Завдання #12
[Переміщення в точку С2]

ЗАГАЛЬНИЙ СТАТУС

УСПІШНО

Режим: АВТОМАТ

ЕТАПИ АЛГОРИТМУ

Ініціалізація

Кінематика
(dO1=12°, dO2=-5°)

Позиціонування

Маніпуляція

Контроль циклів

ТЕЛЕМЕТРІЯ ПРИВОДІВ

Вісь J1: 52° (err 0.4)

Вісь J2: 17° (err 0.3)

Вісь J3: 90° (err 0.6)

Захват: ВІДКРИТО

Час руху: 1.37 с

ПРОТОКОЛ ПОДІЙ

> [10:23:40] RS-485: пакет команди отримано

> [10:23:41] Розрахунок кінематики виконано

> [10:23:42] Досягнуто позиції С2

> [10:23:43] Операція маніпуляції виконана

"ЦИКЛ ЗАВЕРШЕНО"

НАСТУПНЕ ЗАВДАННЯ

ЗАВЕРШИТИ РОБОТУ

Рисунок 4.9. Діалогове вікно «Моніторинг робочого циклу»

Функціональні блоки вікна відображають ключові параметри алгоритму керування: Поточне завдання: Ідентифікує номер операції (Завдання #12) та її опис («Переміщення в точку C2»), що дозволяє відстежувати хід виконання технологічної карти.

Етапи алгоритму: Група індикаторів демонструє послідовне виконання підпрограм Soft-PLC: успішне проходження ініціалізації, розрахунок кінематики (з відображенням розрахованих відхилень $\Delta\theta_1 = 12^\circ$, $\Delta\theta_2 = -5^\circ$), завершення позиціонування та виконання маніпуляції. Телеметрія приводів: Відображаються фактичні кутові координати ланок, отримані через зворотний зв'язок:

- Вісь J1: 52° (похибка 0.4);
- Вісь J2: 17° (похибка 0.3);
- Вісь J3: 90° (похибка 0.6).

Низькі значення похибки підтверджують ефективність обраного алгоритму позиціонування з зонами нечутливості. Також фіксується стан робочого органу («Захват: ВІДКРИТО») та загальний час руху (1.37 с). Протокол подій: Консоль логування фіксує часові мітки критичних подій (отримання пакету, розрахунок кінематики, досягнення позиції), що забезпечує можливість детального аналізу роботи системи та розбору нештатних ситуацій.

Циклограма роботи маніпулятора Для детальної верифікації часових характеристик розробленого програмного забезпечення (Soft-PLC) було побудовано циклограму роботи маніпулятора (рис. 4.10), яка відображає динаміку зміни логічних станів на виходах мікроконтролера та реакцію сенсорної системи протягом одного повного технологічного циклу. Графік охоплює інтервал у 10 умовних одиниць часу і дозволяє проаналізувати узгодженість керуючих сигналів:

Фаза маніпуляції ($t=2\dots3$): На початку циклу спостерігається активація команди «захват» (червона лінія) синхронно зі зміною стану датчика «Стан захвату» (темно-синя лінія). Це підтверджує коректну роботу підпрограми маніпулятора, яка забезпечує замикання захвату перед початком транспортування.

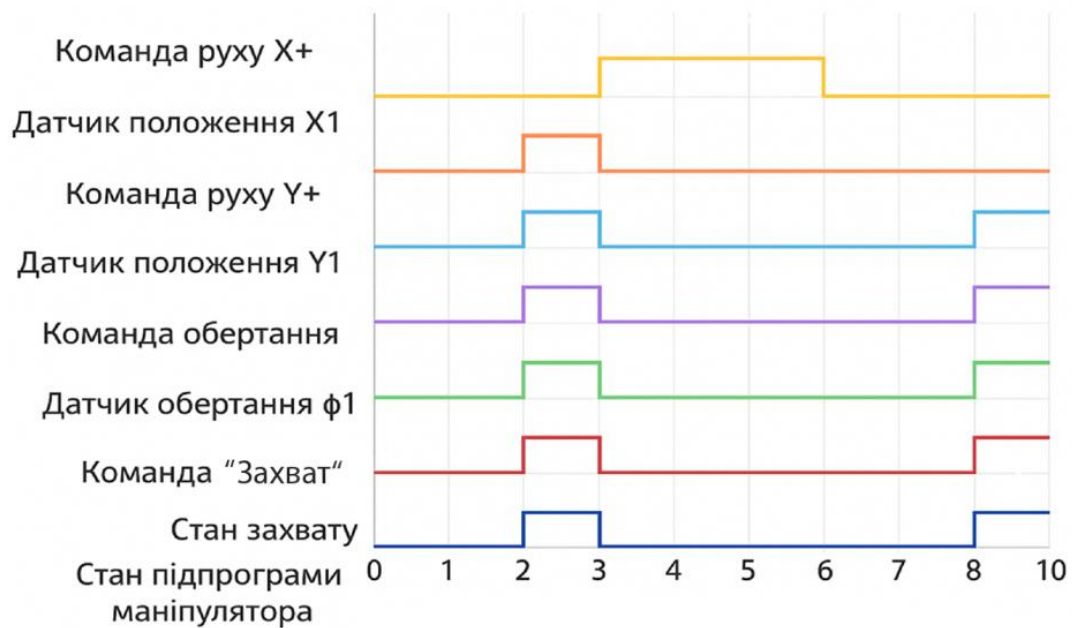


Рисунок 4.10. Циклограма роботи маніпулятора

Фаза транспортування ($t=3\dots8$): Найбільш показовим є інтервал $t=3\dots6$, де фіксується одночасна наявність логічної одиниці на лініях «Команда руху X+» (жовта) та «Команда руху Y+» (блакитна). Це свідчить про те, що розроблений контролер на базі PIC16F1937 успішно виконує багатопотокове керування, забезпечуючи інтерпольований рух одразу за двома осями, що скорочує загальний час циклу.

Синхронізація датчиків: Зміна станів датчиків положення (помаранчева та світло-блакитна лінії) чітко корелює з фронтами керуючих команд. Відсутність «дребезгу» на графіках підтверджує ефективність програмних фільтрів, реалізованих у модулі обробки входів.

Завершення циклу ($t=8$): У момент часу $t=8$ відбувається вимкнення приводу осі Y та повторна активація захвату (ймовірно, для вивільнення деталі), що відповідає логіці завершення переміщення у цільову точку.

Аналіз циклограми підтверджує, що програмна реалізація забезпечує чітку часову детермінованість сигналів, відсутність конфліктів між реверсивними командами та коректну послідовність виконання технологічних операцій.

Висновки за розділом

У даній частині дослідження виконано комплексну розробку алгоритмічного та програмного забезпечення для системи керування промисловим маніпулятором МП-9С на базі мікроконтролера PIC16F1937. Реалізація охопила повний цикл створення вбудованого програмного забезпечення: від формування блок-схем алгоритмів до написання коду та верифікації роботи системи.

Основним досягненням стала розробка деталізованої алгоритмічної бази, що включає процедури ініціалізації обладнання, розрахунку зворотної кінематики, дискретного позиціонування та керування технологічними циклами. Алгоритми враховують специфіку релейного керування, забезпечуючи захисні часові інтервали (Dead Time) та програмну фільтрацію сигналів від датчиків. Для програмної реалізації було обрано підхід «Soft-PLC», що дозволило трансформувати універсальний мікроконтролер у повноцінний промисловий контролер. Застосування логіки стандарту IEC 61131-3 (Ladder Diagram) гарантувало високу надійність системи, детермінованість виконання команд та чітку структуру циклу сканування, що є критично важливим для завдань автоматизації реального часу.

Значну увагу приділено питанням експлуатаційної надійності, що реалізовано через інтеграцію багаторівневої системи безпеки. Логіка дозволу руху базується на послідовній перевірці статусів усіх критичних вузлів, включаючи енкодери, кінцеві вимикачі та силовий драйвер ULN2803. Такий підхід забезпечує безаварійну роботу маніпулятора та захист електромеханічної частини від перевантажень.

Працездатність системи підтверджено експериментально за допомогою розробленого діагностичного інтерфейсу. Отримані дані телеметрії через діалогові вікна моніторингу засвідчили коректність роботи АЦП, стабільність каналу зв'язку RS-485 та точність відпрацювання траєкторій. Побудована циклограма роботи додатково підтвердила часову узгодженість керуючих сигналів та відсутність конфліктів при багатоосьовому переміщенні.

ВИСНОВОК

У магістерській кваліфікаційній роботі вирішено актуальне науково-практичне завдання, що полягає у розробленні програмованого логічного контролера на основі мікроконтролера PIC16F1937 для модернізації системи керування роботом-маніпулятором МП-9С. Метою роботи було підвищення функціональних можливостей, точності, надійності та адаптивності системи керування, що є важливим для сучасних навчальних і промислових робототехнічних комплексів.

У процесі виконання роботи проведено аналіз сучасних роботів-маніпуляторів, принципів побудови їх систем керування та вимог до апаратного й програмного забезпечення. На основі порівняльного аналізу мікроконтролерних платформ обґрунтовано доцільність використання мікроконтролера PIC16F1937, який забезпечує необхідну кількість входів і виходів, підтримку апаратних модулів, низьке енергоспоживання та гнучкість програмної реалізації алгоритмів керування.

Розроблено структурно-логічну та повну електричну схеми програмованого логічного контролера, що включають модулі введення-виведення, драйвер виконавчих механізмів, інтерфейс RS-485. Запропонована архітектура забезпечує надійну комутацію приводів маніпулятора, зчитування сигналів датчиків, реалізацію алгоритмів позиціонування та можливість дистанційної взаємодії з комп'ютером або промисловими мережами.

У роботі розроблено програмне забезпечення ПЛК, яке реалізує алгоритми керування роботом-маніпулятором, обробку даних сенсорів, обмін інформацією та функції діагностики

У результаті виконання магістерської роботи створено універсальний програмований логічний контролер, який дозволяє модернізувати систему керування маніпулятором МП-9С, підвищити його функціональність, точність і надійність.

Список використаних джерел

1. Бойко В.І., Гуржій А.М. Схемотехніка електронних систем: Підручник. – Київ: Вища школа, 2010. – 480 с.
2. Васильєв О. Г., Ольшевський С.І. Методичні вказівки до виконання лабораторних робіт з дисципліни “Системи керування промисловими робототехнічними комплексами”. – Миколаїв: НУК, 2018. – 60 с.
3. Васильєв О. Г., Ольшевський С.І., Черно О. О. Методичні вказівки до виконання курсової роботи з дисципліни “Системи керування промисловими робототехнічними комплексами”. – Миколаїв: НУК, 2020. – 138 с.
4. Голінько В.І., Фрундін В.Ю., Чеберячко Ю.І., Іконніков М.Ю. Охорона праці в галузі: конспект лекцій. – Дніпро: НГУ, 2013. – 84 с.
5. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлення. – Київ: ДП «УкрНДНЦ», 2016. – 26 с.
6. Иванов В. Б. И20 Программирование микроконтроллеров для начинающих. Визуальное проектирование, язык С, ассемблер. – К.: "МК-Пресс", СПб.: "КОРОНА-ВЕК", 2016. – 176 с.
7. Кісельов В.Б. Мікропроцесорні системи керування електроприводами: конспект лекцій. – Київ: КПІ ім. Ігоря Сікорського, 2017. – 115 с.
8. Крак Ю.В. Основи теорії керування та робототехніки: навчальний посібник. – Київ: ВПЦ «Київський університет», 2021. – 152 с.
9. Кузьмінський Р.Д., Яцкім О.В. Мікропроцесорні системи: Навчальний посібник. – Львів: Видавництво Львівської політехніки, 2019. – 236 с.
10. Матвієнко М.П. Комп’ютерна електроніка та схемотехніка: Навчальний посібник. – Київ: Видавництво «Ліра-К», 2018. – 412 с.
11. Павленко І.І., Мажара В.А. Роботизовані технологічні комплекси. – Київ: КПІ ім. І. Сікорського, 2018. – 386 с.
12. Попович М.Г., Ковальчук О.В. Теорія автоматичного керування: Підручник. – Київ: Либідь, 2007. – 656 с.
13. Проць Я.І. Захоплювальні пристрої промислових роботів: Навчальний

посібник. – Тернопіль: Тернопільський державний технічний університет ім. І. Пулюя, 2008. – 232 с.

14. Пуховський Є.С., Малафєєв Ю.М. Проектування гнучких виробничих систем машинобудування. – Київ: Вища школа, 2011. – 268 с.

15. Сафін Ф.Ф., Бікбулатова В.З., Хайретдінов Э.Ф. Автоматизація процесів в ЗМД: лабораторний практикум з дисципліни «Автоматизація, робототехніка та ДПС ковальсько-штампувального виробництва» [Електронний ресурс]. – 2021.

16. Тимкович В.В. Датчики та виконавчі елементи мехатронних систем: Навчальний посібник. – Харків: ХНАДУ, 2016. – 184 с.

17. Ткачов В.В., Проць Я.І. Основи робототехніки: Навчальний посібник. – Тернопіль: ТНТУ імені Івана Пулюя, 2019. – 208 с.

18. Шпак З.Я. Програмування мовою С: Навчальний посібник. – Львів: Видавництво Львівської політехніки, 2011. – 436 с.

19. HD44780U (LCD-II) Dot Matrix Liquid Crystal Display Controller/Driver: User Manual. – Hitachi, Ltd., 1999. – 60 p.

20. IEC 61131-3:2013. Programmable controllers – Part 3: Programming languages. – International Electrotechnical Commission, 2013.

21. Kernighan B.W., Ritchie D.M. The C Programming Language. 2nd Edition. – Prentice Hall, 1988. – 272 p.

22. Kugelstadt T. The RS-485 Design Guide: Application Report. – Dallas: Texas Instruments Incorporated, 2022. – 29 p.

23. MAX485/MAX490/MAX491 Low-Power, Slew-Rate-Limited RS-485/RS 422 Transceivers: Data Sheet. – Maxim Integrated Products, 2014. – 20 p.

24. PIC16F193X/LF193X Data Sheet: 28/40/44-Pin Flash-Based, 8-Bit CMOS Microcontrollers with LCD Driver and nanoWatt XLP Technology. – Chandler: Microchip Technology Inc., 2009. – 418 p.

25. ULN2803A Darlington Transistor Arrays: Data Sheet [Electronic resource]. – Texas Instruments Incorporated, 2016. – 19 p.

ДОДАТОК А

Слайди мультимедійної презентації

Кваліфікаційна робота для здобуття
ступеню магістр на тему:

«Проектування системи керування робота-
маніпулятора»

Виконав: студент групи 6371м
Шамраєв Валерій Віталійович
Керівник: Козлов А.Ю.

Мета роботи: розроблення та реалізація автоматизованої системи керування промисловим роботом-маніпулятором МП-9С на основі спеціалізованого програмованого логічного контролера, побудованого на базі мікроконтролера PIC16F1937, з підтримкою обміну даними по інтерфейсу RS-485.

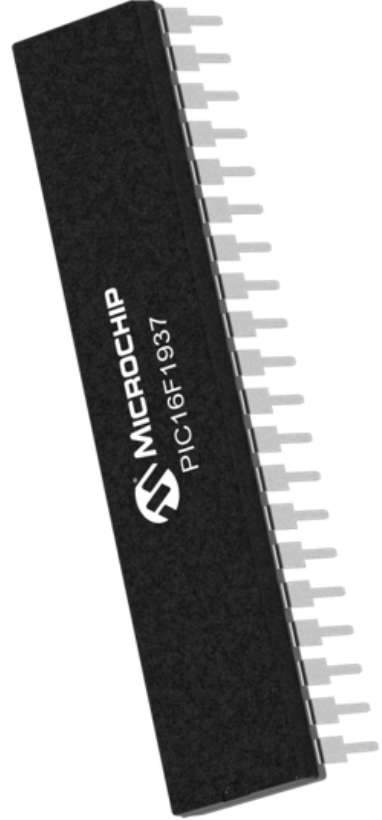
Об’єкт дослідження: система автоматичного керування позиціонуванням та переміщенням ланок промислового робота-маніпулятора МП-9С.

Предмет дослідження: апаратна реалізація вузлів контролера на базі мікроконтролера PIC16F1937 та розроблення алгоритмічного забезпечення за принципом Soft-PLC (із застосуванням логіки Ladder Diagram) для керування електроприводами, обробки сигналів датчиків та організації мережевої взаємодії.

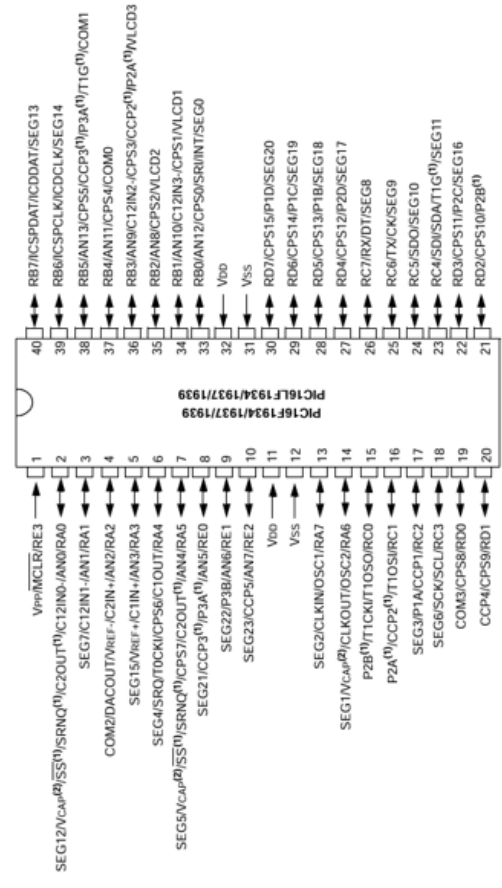
Маніпулятор МП-9С



Мікроконтролер PIC16F1937

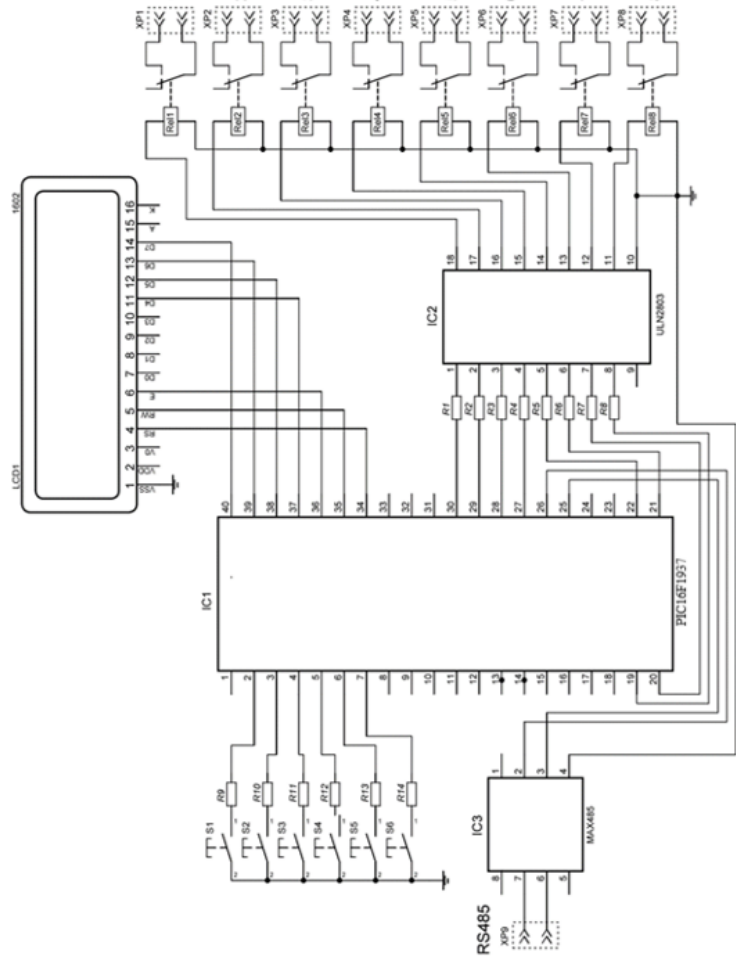


40-Pin PDIP



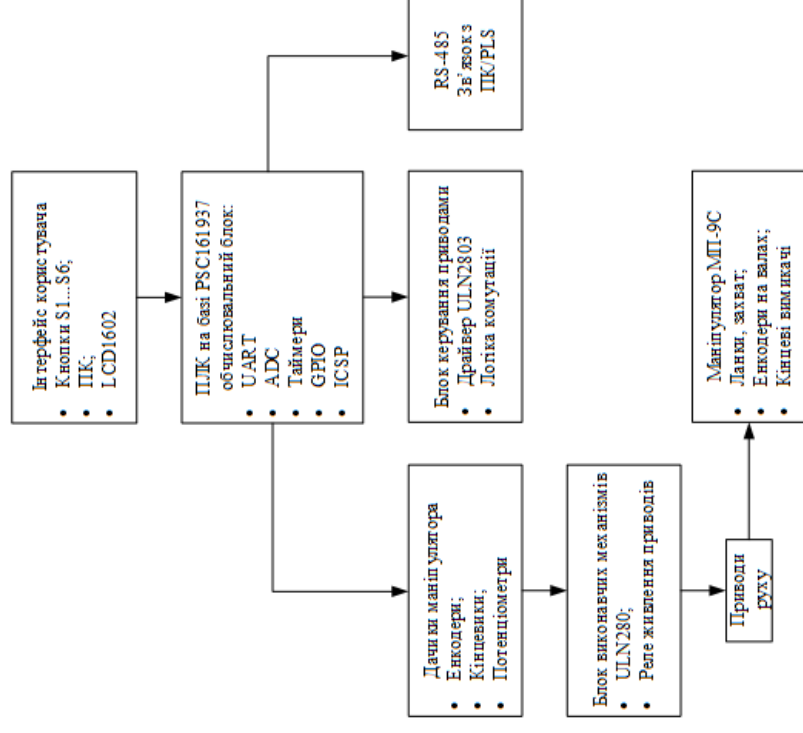
Слайд 4 – Мікроконтролер PIC16F1937

Принципова електрична схема ПЛК, розробленого на базі мікроконтролера PIC16F1937



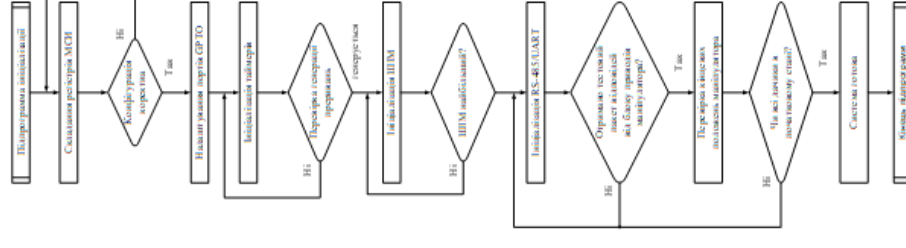
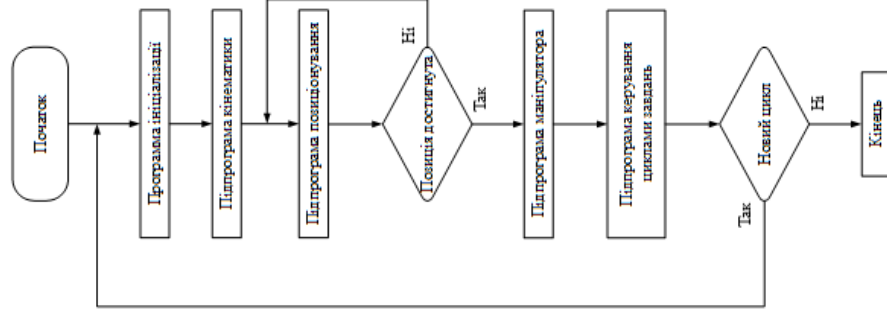
Слайд 5 – Принципова електрична схема

Функціональна схема логічного контролера, створеного на базі мікроконтролера PIC16F1937

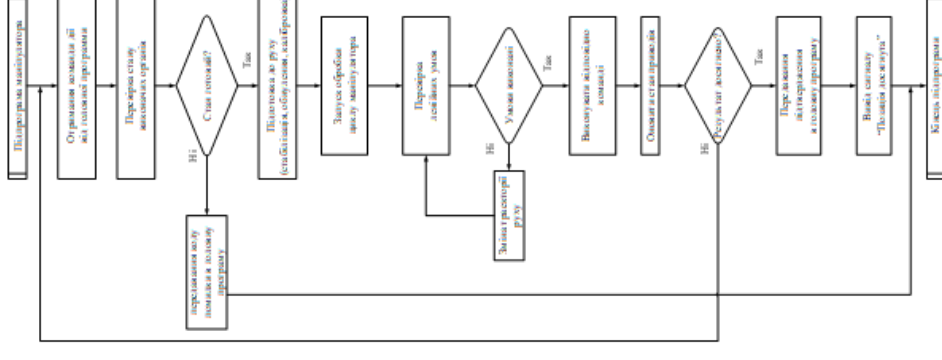
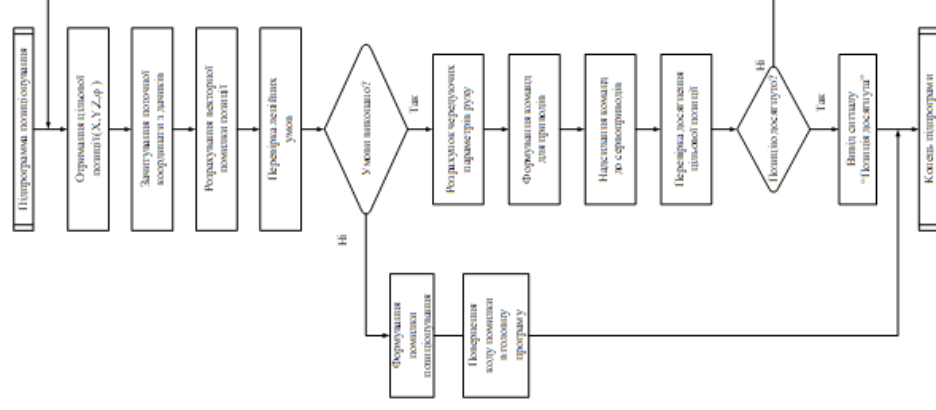
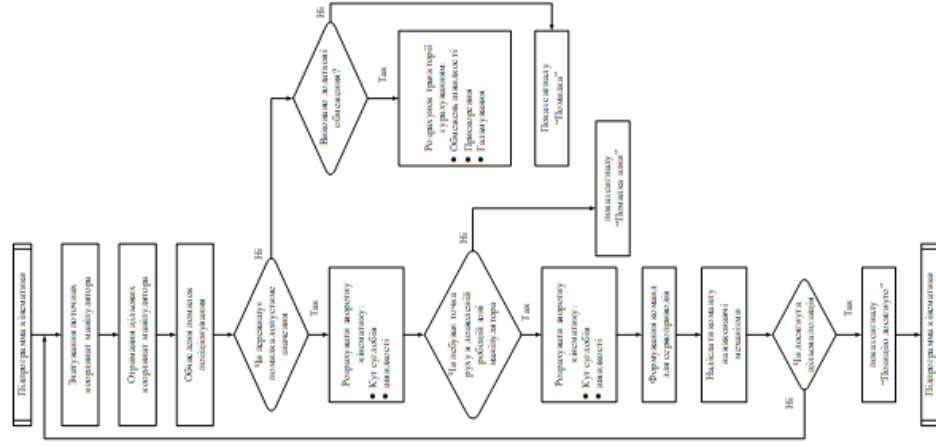


Слайд 6 – Функціональна схема

Алгоритми керуючої програми



Алгоритми підпрограми



Слайд 8 – Алгоритми підпрограми

Результати функціонування програми моніторингу та діагностики по RS485

Пульт керування МП-9С

Блок зв'язку (RS-485)
Порт: COM3 [9600, 8, N, 1]
З'єднання з ПЛК: [OK]
Пінги (Ping): 4ms

СИСТЕМНИЙ СТАТУС
Живлення: [OK]
Аварія (E-Stop): [OFF]

ПЕРЕВІРКА ПЕРИФЕРІЇ

Модуль	Статус	Значення / Код
Драйвер ULN2803	ГОТОВИЙ	Всі реле OFF
Датчик кута J1	НОРМА	ADC: 512 (2.5V)
Датчик кута J2	НОРМА	ADC: 128 (0.6V)
Датчик кута J3	НОРМА	ADC: 715 (3.5V)
Датчик Захвату	НОРМА	ADC: 1023 (5.0V)
Ліміти руху	АКТИВНІ	Hard Limits Check

СТАН ПОЧАТКОВОЇ КАЛІБРОВКИ
Нульові кути
J1 J2 J3
Захват
Встановлено базовий стан

"ІНІЦІАЛІЗАЦІЮ ЗАВЕРШЕНО"

ПРОДОВЖИТИ

ПЕРЕЗАПУСК

Моніторинг виконання автоматичного циклу: Манипулятор МП-9С

ПОТОЧНЕ ЗАВДАННЯ
Завдання #12
[Переміщення в точку C2]

ЗАГАЛЬНИЙ СТАТУС
УСПІШНО
Режим: АВТОМАТ

ЕТАПИ АЛГОРИТМУ

Ініціалізація	
Кінематика ($\phi O1=12^\circ$, $\phi O2=5^\circ$)	
Позиціонування	
Манипуляція	
Контроль циклів	

ТЕЛЕМЕТРІЯ ПРИВОДІВ
Вісь J1: 52° (err 0.4)
Вісь J2: 17° (err 0.3)
Вісь J3: 90° (err 0.6)
Захват: ВІДКРИТО
Час руху: 1.37 с

ПРОТОКОЛ ПОДІЙ
> [10:23:40] RS-485: пакет команди отримано
> [10:23:41] Розрахунок кінематики виконано
> [10:23:42] Досягнуто позиції C2
> [10:23:43] Операція манипуляції виконана

"ЦИКЛ ЗАВЕРШЕНО"

НАСТУПНЕ ЗАВДАННЯ

ЗАВЕРШИТИ РОБОТУ

Висновки:

- розроблено логічний контролер на базі PIC16F1937 для керування роботом-маніпулятором МП-9С.
- реалізовано обмін даними через інтерфейс RS-485 та алгоритми керування у форматі Ladder-логіки.
- забезпечено апаратний захист та безпечну експлуатацію маніпулятора в автоматичному режимі.
- проведено тестування системи, що підтвердило стабільність позиціонування та виконання робочих циклів.