

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ
Кафедра “підйомно-транспортного та робототехнічного обладнання”

НАВЧАЛЬНИЙ ПОСІБНИК

**з дисциплін «Електронні, мікропроцесорні та обчислювальні
пристрої ГВС, ПТМ та ЛС»**

Рівень підготовки – бакалавр

Галузь знань: 13 – Механічна інженерія

Спеціальність: 131 – Прикладна механіка

Спеціалізація: – Мехатроніка та промислові роботи

– Інженерія логістичних систем

Спеціальність: 133 – Галузеве машинобудування

Спеціалізація: – Підйомно-транспортні, будівельні, дорожні
машини і обладнання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ
Кафедра “підйомно-транспортного та робототехнічного обладнання”

Михайлов Євген Павлович

НАВЧАЛЬНИЙ ПОСІБНИК

**з дисциплін «Електронні, мікропроцесорні та обчислювальні
пристрої ГВС, ПТМ та ЛС»**

Рівень підготовки – бакалавр

Галузь знань: 13 – Механічна інженерія

Спеціальність: 131 – Прикладна механіка

Спеціалізація: – Мехатроніка та промислові роботи

– Інженерія логістичних систем

Спеціальність: 133 – Галузеве машинобудування

Спеціалізація: – Підйомно-транспортні, будівельні, дорожні
машини і обладнання

Затверджено

на засіданні кафедри підйомно-
транспортного і робототехнічного
обладнання

Протокол № від .2018 р.

ОДЕСА: ОНПУ, 2018

Навчальний посібник з дисциплін «Електронні, мікропроцесорні та обчислювальні пристрої ГВС, ПТМ та ЛС» для студентів за фахом 131 – Прикладна механіка – спеціалізації – Мехатроніка та промислові роботи, Інженерія логістичних систем, 133 – Галузеве машинобудування – спеціалізація – Підйомно-транспортні, будівельні, дорожні машини і обладнання / Укладач: Михайлов Є. П. Одеса: ОНПУ. – 121 с.

Укладач: Михайлов Є. П. доц., кафедри підйомно-транспортного і робототехнічного обладнання.

Зміст

Передмова.....	5
Глава 1. Електронні пристрої як основа систем керування.....	6
1.1. Основні поняття і визначення.....	6
1.2. Основні компоненти електронних пристроїв.....	7
1.3. Напівпровідникові підсилювачі.....	12
Глава 2. Застосування аналогових та дискретних інтегральних мікросхем	15
2.1. Аналогові мікросхеми	15
2.2. Приклади використання аналогових мікросхем	15
2.3. Цифрові мікросхеми.....	17
2.4. Приклади використання цифрових мікросхем.....	18
Глава 3. Аналого-цифрові та цифро-аналогові перетворювачі.....	22
3.1. Принцип роботи та основні типи цифро-аналогових перетворювачів	22
3.2. Принцип роботи та основні типи аналого-цифрових перетворювачів	24
3.3. Приклади використання аналого-цифрових та цифро-аналогових перетворювачів в системах керування	26
Глава 4. Архітектура мікропроцесорних керуючих пристроїв. Уявлення даних у керуючих обчислювальних пристроях	29
4.1. Класифікація мікропроцесорних пристроїв	29
4.2. Архітектура керуючих обчислювальних пристроїв.....	29
4.3. Уявлення даних у керуючих обчислювальних пристроях.....	30
4.4. Структура обчислювальних пристроїв.....	32
Глава 5. Пристрої керування на основі мікропроцесорної техніки	36
5.1. Однокристальні мікроконтролери як основа для вбудованих пристроїв керування.....	36
5.2. Класифікація й основні типи однокристальних мікроконтролерів	36
5.3. Приклади пристроїв керування на основі однокристальних мікроконтролерів	40
Глава 6. Пристрої керування на основі мікроконтролерів	43
6.1. Основні типи одноплатних мікроконтролерів	43
6.2. Мови програмування одноплатних мікроконтролерів	45
6.3. Функції керування рухом та обробки даних	48
Глава 7. Програмовані логічні контролери на основі мікропроцесорної техніки ..	55
7.1. Класифікація, структура і склад програмованих логічних контролерів.	55
7.2. Модулі центральних процесорів.....	57
7.3. Сигнальні, функціональні та комунікаційні модулі	57
7.4. Мови програмування програмованих логічних контролерів.....	59

Глава 8. Проектування апаратних компонент систем керування на основі програмованих логічних контролерів.....	62
8.1. Складання проекту системи керування.....	62
8.2. Визначення структури і складу системи керування.....	64
8.3. Налагодження системи керування.....	67
Глава 9. Проектування програмних компонент систем керування на основі програмованих логічних контролерів.....	70
9.1. Структура програм програмованих логічних контролерів.....	70
9.2. Система команд програмованих логічних контролерів.....	71
9.3. Засоби налагодження програми та пошуку помилок.....	76
Глава 10. Структура і склад комплексних систем керування	82
10.1. Архітектура, структура та склад комплексних систем керування	82
10.2. Апаратні і програмні компоненти	85
10.3. Засоби автоматизації на основі пристроїв керування, сумісних з персональними комп'ютерами	88
Глава 11. Обчислювальні мережі у системах керування. Локальні та глобальні обчислювальні мережі.....	92
11.1. Види мереж. Поняття інтерфейсу.....	92
11.2. Локальні мережі для систем керування.....	92
11.3. Децентралізована периферія	96
11.4. Регіональні і глобальні мережі.....	99
Глава 12. Системи відображення технологічних процесів і керування.....	101
12.1. Системи відображення технологічних процесів і керування	101
12.2. Операторські панелі та операторські станції	101
12.3. Інструментальні програмні засоби для відображення технологічного процесу	102
Література.....	108
Додаток 1	109
Додаток 2.....	115

Передмова

Електронні, мікропроцесорні та обчислювальні пристрої є основою автоматизованих систем керування що широко використовуються в сучасному виробництві, а саме, у складі мехатронних компонент гнучких виробничих систем, підйомно-транспортних машин та в обладнанні логістичних систем.

Сучасна система автоматизованого керування у загальному вигляді складається з програмованого пристрою комп'ютерного керування, інформаційної системи, до якої належать різноманітні датчики, та виконавчих пристроїв, основою яких є електронні, мікропроцесорні та обчислювальні пристрої. Важливим прикладом використання таких пристроїв є мехатронні модулі та системи, що використовують для керування рухом.

Швидкий розвиток електронної та мікропроцесорної техніки дає можливість використовувати комп'ютерні системи для вирішення різних задач на різних рівнях керування. При цьому мікропроцесорні компоненти реалізують алгоритм керування та обробки даних на різних рівнях, а електронні компоненти вирішують задачі узгодження системи керування з об'єктами керування та обслуговуючим персоналом.

Таким чином не тільки сам пристрій керування, а також інші компоненти, а саме інформаційна система та виконавчі пристрої можуть мати вмонтовані обчислювальні пристрої для комп'ютерної обробки даних та керування, що об'єднуються у комплексну багаторівневу систему керування. Така система використовує комп'ютери (операторські станції) на верхньому рівні керування, пов'язаних з локальними системами керування. Оскільки окремі компоненти системи керування складних систем можуть знаходитись на досить великих відстанях, застосовуються принципи розподіленого керування на основі промислових мереж.

Сучасні інформаційні систем (наприклад, технічний зір) та приводні системи (наприклад, частотні перетворювачі) можуть мати власні досить складні системи обробки даних та керування на основі обчислювальних пристроїв.

Таким чином, у сучасному промисловому обладнанні можна виділити системи програмного керування, що здійснюються за допомогою обчислювальних пристроїв на основі мікропроцесорної техніки, та електронні компоненти, які використовують для реалізації інформаційних та виконавчих пристроїв, а також для узгодження технологічного обладнання з пристроями керування.

Дисципліна «Електронні, мікропроцесорні та обчислювальні пристрої ГВС, ПТМ та ЛС» призначена для вивчення студентами питань побудови, основ проектування та застосування електронних, мікропроцесорних та обчислювальних пристроїв в підйомно-транспортних машинах, гнучких виробничих та логістичних системах та поширює їх інженерні знання за рахунок розгляду різних типів електронних, мікропроцесорних та обчислювальних пристроїв та їх компонент, що базуються на сучасних засобах.

Наведені задачі та завдання до практичних занять.

Глава 1. Електронні пристрої як основа систем керування

1.1. Основні поняття і визначення

Складовими частинами або елементами електронних приладів є **електронні компоненти**.

Електронні компоненти поділяють на пасивні, активні, електромеханічні та оптоелектронні.

Пасивні електронні елементи не збільшують потужність електричного струму, що проходить через елемент.

До пасивних електронних компонент можна віднести резистори, конденсатори та індуктивності.

Активні елементи можуть збільшувати потужність електричного струму або здійснювати складне перетворювання електричних сигналів, що проходять через елемент.

До активних електронних компонент можна віднести такі напівпровідникові елементи як діоди та транзистори, що можуть здійснювати перетворення сигналів.



Рис. 1.1. Пасивні та активні електронні елементи

Електромеханічні електронні компоненти складаються з таких елементів як електричні з'єднувачі, перемикачі, реле і контактори (рис. 1.2).

Сучасні електронні пристрої широко використовують оптоелектронні елементи, такі як світлодіоди, фотоелементи, дисплеї, лазери.



Рис. 1.2. Електромеханічні електронні компоненти

Мікромініатюризація сучасних напівпровідникових приладів привела до появи такого класу електронних компонент як мікросхеми, які мають у своєму складі кількість елементів, що вимірюється такими значеннями як 10^9 , 10^{11} . Так, наприклад, сучасні мікросхеми так званої флеш-пам'яті мають ємкість до сотень гігабайтів, що потребує відповідну кількість запам'ятовуючих елементів (один гігабайт дорівнює $8 \cdot 2^{30}$).

Мікросхеми не в змозі повністю замінити всі електронні компоненти, оскільки мають обмежену потужність вихідних сигналів, а тому не можуть безпосередньо підключатися до багатой кількості виконавчих пристроїв, що потребують потужних сигналів.

Тому сучасні електронні компоненти можна також поділити на дискретні елементи та мікросхеми.

До дискретних елементів відносяться пасивні елементи та активні елементи, що виконуються у вигляді окремих конструктивних елементів (резистори, конденсатори, котушки індуктивності, різні напівпровідникові пристрої тощо).

Мікросхеми поділяють за ступенем інтеграції, яка визначає функціональні можливості.

До окремого класу можна віднести мікросхем можна віднести мікропроцесори, що реалізують обчислювальні пристрої, які здійснюють функції програмного керування та обробки даних.

Сучасні промислові системи керування використовують функції програмного керування та обробки даних на основі обчислювальних пристроїв для реалізації алгоритмів керування.

Далі будуть розглянуті електронні, мікропроцесорні та обчислювальні пристрої як основа сучасних систем керування. При цьому основним елементом керування є обчислювальний пристрій, що виконується на основі мікропроцесорної техніки, а різні електронні пристрої виконують функції узгодження обчислювальних пристроїв з датчиками, що надають інформацію про стан об'єктів керування, та з виконавчими пристроями, що здійснюють функціонування технологічного обладнання.

1.2. Основні компоненти електронних пристроїв

Розглянемо особливості пасивних електронних компонент, до яких можна віднести резистори, конденсатори та індуктивності.

Для пасивних електротехнічних компонент важливою характеристикою є залежність струму від напруги або вихідного сигналу від вхідного, які можна подати як залежність від часу або як залежність від частоти. Для пасивних елементів, а саме резисторів R , конденсаторів C та котушок індуктивності L ці залежності наведені на рис.1.3.

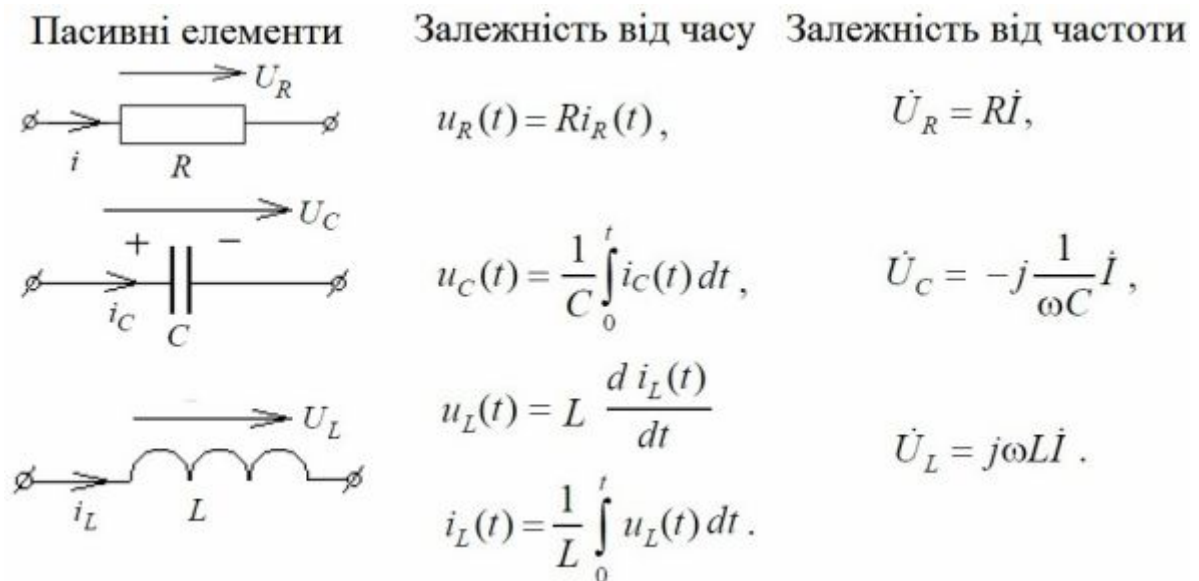


Рис.1.3. Залежності напруги та струму пасивних елементах від часу та частоти

Вказані залежності показують, як ці компоненти впливають на параметри струму та напруги.

Резистори знаходять використання для поділення напруги та обмеження струму. Схема подільника напруги та залежність вихідної напруги від параметрів подільника наведені на рис. 1.4.

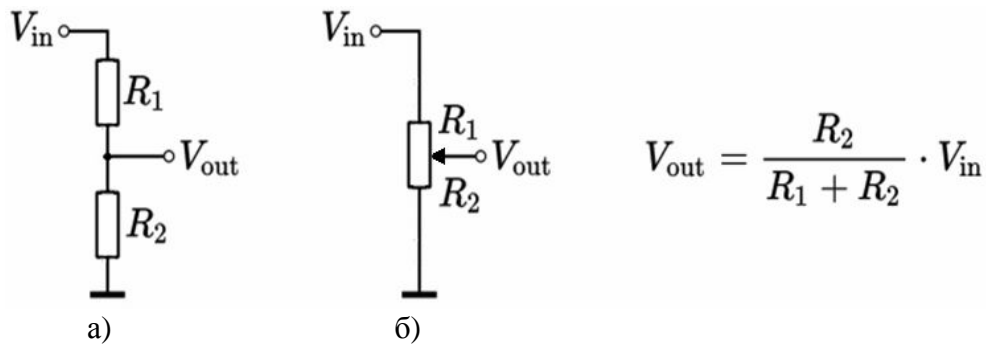


Рис. 1.4. Схема подільника напруги та залежність вихідної напруги від параметрів подільника

Така схема використовується для отримання різних напруг V_{out} з одної V_{in} (а). Вихідна напруга може змінюватися, якщо використовуються потенціометр (б - резистор з трьома виводами, один із яких рухомий, що використовується, як дільник напруги).

У резисторів проходження струму викликає нагрівання, яке пропорційне потужності $P = I U = U^2 / R = I^2 R$, що виділяється на резисторі.

Цю особливість треба враховувати, коли опір може змінюватися під впливом зовнішніх факторів, наприклад, коли контакти окислюються під впливом струму та підвищеної температури, що приводить до збільшення контактного опору. Це приводить до ще більшого нагріву контакту і збільшення опору, та може стати причиною займання.

Опір резистора залежить від температури, тому їх використовують як датчики температури.

Принцип дії термометрів опору базується на властивості провідників змінювати свій електричний опір при зміні температури.

Залежність опору провідника від його температури у найпростішому випадку виражається лінійною залежністю:

$$R_T = R_0 (1 + \alpha \cdot \Delta T)$$

де:

R_T — електричний опір при температурі T [Ом];

R_0 — електричний опір при початковій температурі T_0 [Ом];

α — температурний коефіцієнт електричного опору [K^{-1}];

ΔT — зміна температури, що становить $T - T_0$ [K].

Матеріали, які використовуються для виготовлення термометрів опору, повинні мати максимальний і постійний температурний коефіцієнт електричного опору, лінійну залежність опору від температури, мати відтворюваність властивостей і інертність до впливів навколишнього середовища.

Для виготовлення сенсорів термометрів опору використовують мідь, нікель, платину, вольфрам, що мають позитивний температурний коефіцієнт.

Часто на практиці для вимірювання деформації використовуються тензодатчики.

Принцип дії тензодатчика заснований на залежності активного опору провідника від його механічної деформації.

Тензодатчики наклеюються на деформується поверхню так, щоб прямолінійні ділянки провідника розтягувалися або стискалися відповідно з деформацією деталі.

Конструкція тензодатчика спірального типу та його умовне позначення наведені на рис. 1.5, де 1 - дротова спіраль, 2 - контакти підключення.

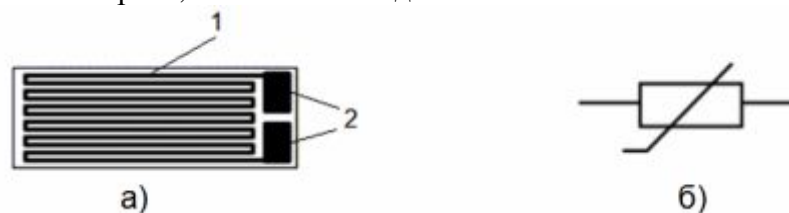


Рис. 1.5. Конструкція тензодатчика спірального типу (а) та його умовне позначення (б)

Такі датчики спільно з підсилювачами дозволяють вимірювати малі механічні деформації, що складають кілька мікронів.

Чутливість тензорезистора характеризується безрозмірним параметром - коефіцієнтом тензочувливості K_f , який визначається як

$$K_f = \Delta R / R_0 \cdot \varepsilon,$$

де ΔR - абсолютна зміна опору, викликане деформацією, Ом;
 R_0 - початковий опір недеформованого тензорезистора, Ом;
 ε - відносна деформація.

Відносна деформація визначається як:

$$\varepsilon = \Delta L / L_0,$$

де ΔL - абсолютна зміна довжини, м;
 L_0 - довжина недеформованого тензорезистора, м.

Зазвичай тензорезистори включають в одне або два плеча збалансованого моста Уїтстона (рис. 1.6), що живиться від джерела постійного струму (діагональ моста А-В).

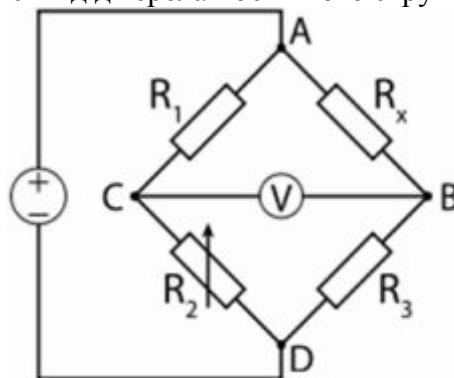


Рис. 1.6. Міст Уїтстона

При виконанні співвідношення $R_1 / R_2 = R_x / R_3$, де R_x – терморезистор, напруга діагоналі моста дорівнює нулю.

При деформації змінюється опір R_x , це викликає зниження потенціалу точки з'єднання резисторів R_x і R_3 (точки В) і зміна напруги діагоналі В-С моста - v визначається формулою:

$$v = V_b \cdot K_f \cdot \varepsilon / 4$$

де V_b - напруга живлення моста, В.

За допомогою змінного резистора R_2 проводиться балансування моста, так, щоб під час відсутності прикладеної сили напруга діагоналі зробити рівним нулю.

З діагоналі моста В-С знімається сигнал, який далі подається на вимірювальний прилад.

Ємність конденсатора залежить від діелектрика між пластинами, індуктивність котушки індуктивності залежить від металевого сердечника, що знаходиться в магнітному полі котушки.

Ця особливість використовуються в датчиках положення.

Індуктивні датчики положення засновані на зміні індуктивності обмотки з сердечником внаслідок переміщення сердечника або впливу металевих об'єктів на магнітне поле.

Ємнісні датчики положення відрізняються тим, що елемент, який впливає на ємність датчика, не повинен обов'язково бути металевим.

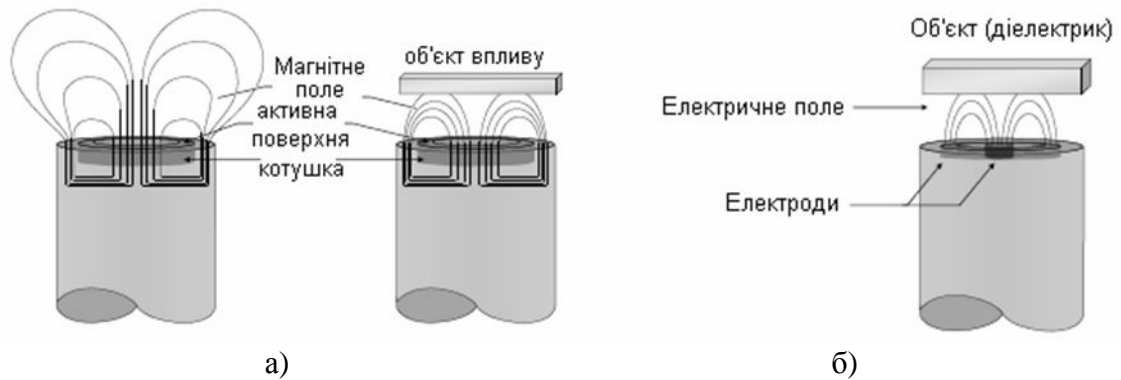


Рис. 1.7. Принцип дії індуктивних (а) та ємкісних (б) датчиків

Важливою властивістю конденсатора та індуктивності є те, що напруга на конденсаторі та струм, який проходить скрізь індуктивність, не можуть змінюватися миттєво. Наприклад, у пристроях живлення, де є конденсатори, напруга після відключення зберігається деякий час. В котушках реле або статорах двигунів, які є індуктивністю, струм не може змінюватися миттєво при відключенні від мережі живлення, що викликає появу високої напруги на розімкнутих контактах вимикача. Ці властивості пасивних елементів треба приймати до уваги, а в ряді випадків при їх використанні треба вжити спеціальні засоби безпеки.

Опір конденсаторів та індуктивностей залежить від частоти струму. Зі збільшенням частоти опір індуктивностей збільшується, а конденсаторів зменшується. Тому ці пасивні елементи можуть використовуватися для фільтрації частот, зокрема для захисту від електромагнітних перешкод. На рис 1.8 наведені приклади для фільтрів низьких частот (рис.1.8, а) та фільтрів високих частот (рис.1.8, б), та залежності їх перехідних $U(t)$ та частотних $K(\omega)$ характеристик.

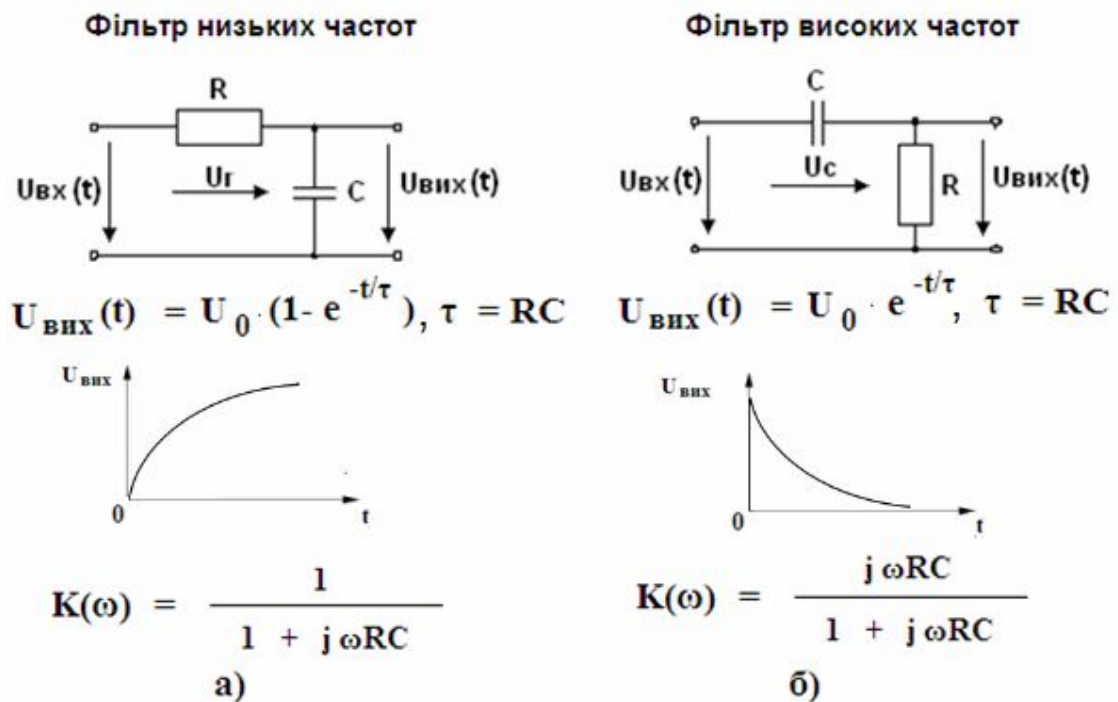


Рис.1.8. Приклади фільтрів: а - низьких частот, б - високих частот

Фільтр низьких частот виконує також функцію інтегрування, а фільтр високих частот – функцію диференціювання.

До активних елементів відносять вироби напівпровідникової техніки, що володіють функціями перетворення та підсилення сигналів, такі як діоди, тиристори, біполярні транзистори, польові транзистори тощо (рис. 1.9).

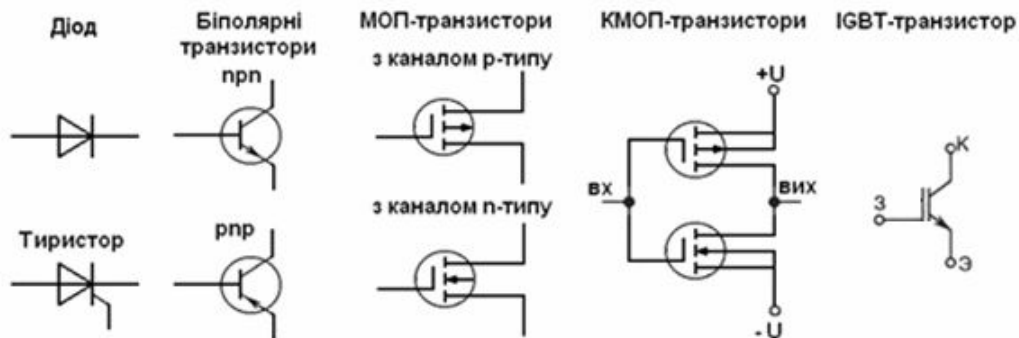


Рис. 1.9. Активні елементи

Особливістю КМОП-транзисторів є відсутність струму у стані спокою в ключовому режимі, оскільки один транзистор відкритий, а другий закритий. IGBT-транзистори поєднують позитивні характеристики біполярних та польових транзисторів, а саме, високий вхідний опір та великий вихідний струм.

До активних елементів можна також віднести оптоелектронні компоненти, такі як світлодіоди, фотодіоди та фототранзистори (рис. 1.10). Світлодіоди використовують як світлові індикатори, а разом з фотодіодами або фототранзисторами – як фотодатчики.

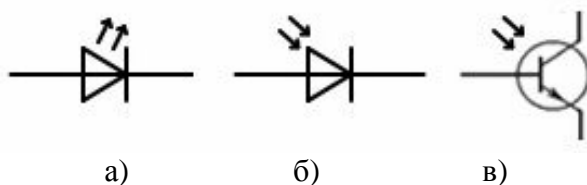


Рис. 1.10. Оптоелектронні компоненти: світлодіоди (а), фотодіоди (б) та фототранзистори (в)

Оптоелектронні пари широко використовують для гальванічної розв'язки вхідних та вихідних сигналів для захисту від небажаного впливу різних електромагнітних перешкод та як щільні датчики переміщення (рис. 1.11).

Резистор R на рис. 1.11 обмежує струм світлодіода.

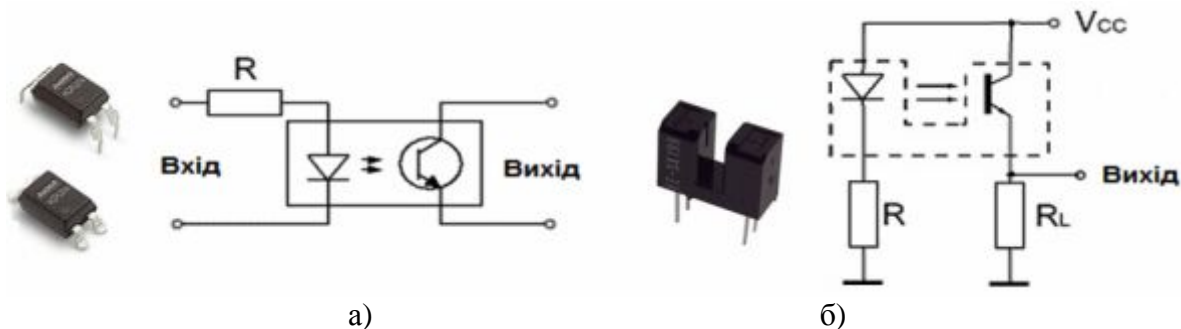


Рис. 1.11. Схеми пристроїв гальванічної розв'язки (а) та щільних датчиків переміщення (б)

Якщо світлодіод використовується як індикатор, то схему його підключення можна представити у вигляді рис. 1.12.

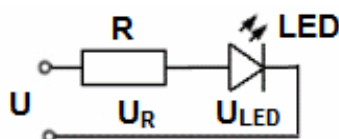


Рис. 1.12. Схема підключення світлодіода

Опір резистора R визначається падінням напруги на світлодіоді U_{LED} , допустимим струмом світлодіода I_{LED} та напругою живлення U

$$R = (U - U_{LED}) / I_{LED} .$$

Якщо використовувати напругу живлення $U = 5$ В, то для стандартного світлодіода ($U_{LED} = 2$ В, $I_{LED} = 20$ мА) отримуємо $R = 150$ Ом.

Діоди, які пропускають струм тільки в одному напрямку, використовуються для перетворення змінного струму в постійний (випрямлячі) у джерелах живлення. На рис. 1.13 наведені однофазний та трифазний випрямлячі.

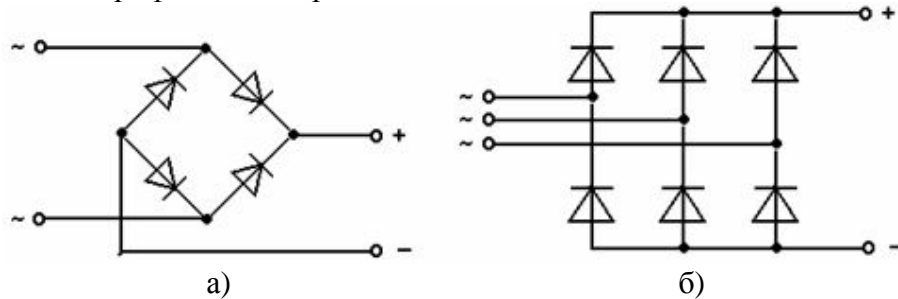


Рис. 1.13. Випрямлячі: а – однофазний, б - трифазний

Як відмічалось раніше, при відключенні контакторів та двигунів, які являють собою велику індуктивність, на розімкнутих контактах вимикача виникає велика напруга. На рис. 1.14 наведена схема замкнення струму при відключенні індуктивності за допомогою діоду.

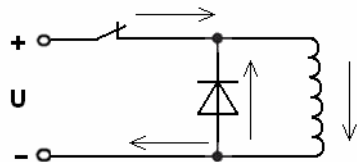


Рис. 1.14. Схема замкнення струму при відключенні індуктивності за допомогою діоду

1.3. Напівпровідникові підсилювачі

Біполярні та польові транзистори використовуються у підсилювачах напруги та струму. На рис 1.15 наведені схеми підсилювальних каскадів на біполярних транзисторах із спільним емітером (інвертуючий каскад, рис 1.15, а) та із спільним колектором (неінвертуючий каскад, рис 1.15, б).

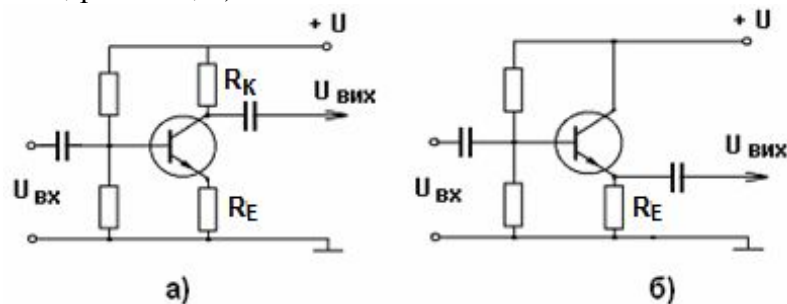


Рис 1.15. Схеми каскадів підсилювання: а - із спільним емітером, б - із спільним колектором

Струм колектору , бази та емітеру пов'язані такими залежностями:

$$I_E = I_B + I_K, \quad I_K = \beta I_B, \quad I_K = \alpha I_E, \quad \beta = \alpha / (1 + \alpha),$$

де β – коефіцієнт підсилювання по струму, що зазвичай знаходиться у діапазоні 10 - 300.

Схема із спільним емітером використовується як підсилювач напруги, при цьому коефіцієнт підсилювання дорівнює:

$$K_U = R_K / R_E.$$

Схема із спільним колектором використовується як підсилювач струму, при цьому коефіцієнт підсилювання дорівнює:

$$K_I = I_E / I_B = 1 + \beta.$$

Біполярні та польові транзистори відрізняються засобом підсилювання. Біполярні транзистори підсилюють струм, а польові транзистори напругу. Біполярні транзистори мають більшу потужність, а польові транзистори мають дуже малий вхідний струм.

Тиристри та транзистори використовуються у ключових схемах, які здійснюють включення і виключення силового кола за допомогою керованого вентильного приладу (тиристор або транзистор у відкритому або закритому стані). На рис. 1.16 наведені проста ключова транзисторна схема (а) для підключення двигуна постійного струму та так званий Н-міст (б), за допомогою якого можна змінювати напрямок струму двигуна (реверс).

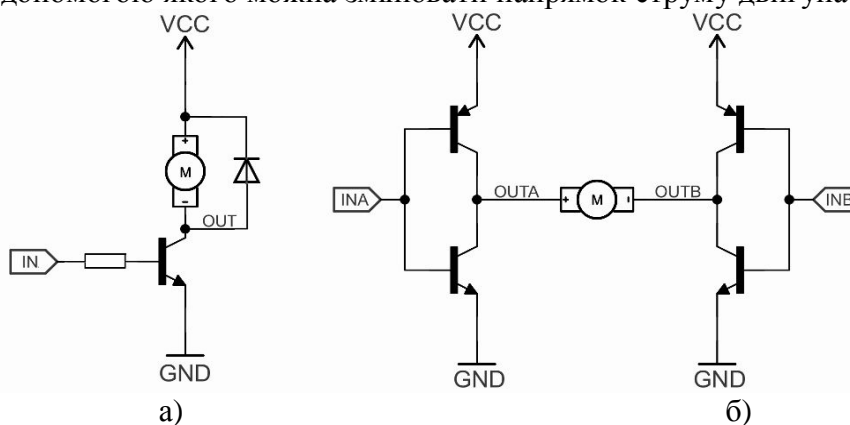


Рис. 1.16. Ключові схеми

У сучасних силових каскадах, які працюють у ключовому режимі, використовуються IGBT-транзистори, які складаються з двох транзисторних каскадів, на вході польовий, а на виході біполярний транзистор. Завдяки цьому вони мають змогу переключати струм великої потужності при дуже малому струмі на вході.

В сучасних цифрових схемах використовуються КМОП транзистори, які під час спокою не потребують струму, бо відкритим є тільки один з них, а другий завжди закритий.

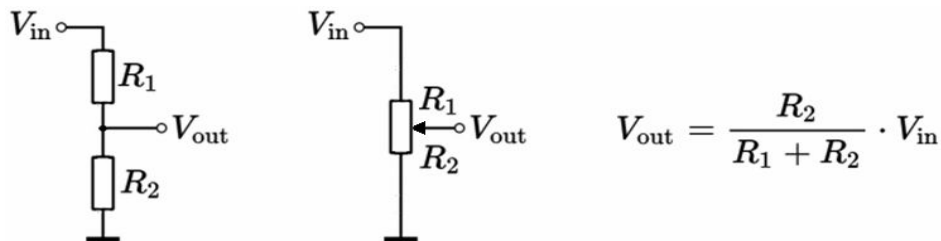
Контрольні запитання

1. Які особливості мають резистори?
2. Які функції виконує подільник напруги?
3. На якому елементі не можна миттєво змінити напругу?
4. У якого елемента не можна миттєво змінити струм?
5. Для чого використовують діоди?
6. Які особливості мають IGBT-транзистори?
7. Як визначити опір резистора для обмеження струму світлодіода?
8. Для чого використовуються напівпровідникові підсилювачі?
9. Чим відрізняються схеми із спільним емітером та спільним колектором?
10. Як працюють ключові схеми? Наведіть приклади ключових схем.

Завдання до практичних занять

Задача 1

Знайти параметри подільника напруги $R1$ та $R2$ для заданих значень коефіцієнта передачі V_{out} / V_{in} та загального опору $R1 + R2$.



V_{out} / V_{in}	$R1 + R2, \kappa\Omega$	$R1, \kappa\Omega$	$R2, \kappa\Omega$
0,1	1		
0,5	50		
0,25	270		
0,01	100		

Задача 2

Знайти V_{out} , для заданих значень V_{in} , $R1$ та $R2$.

V_{in}, B	$R1, \kappa\Omega$	$R2, \kappa\Omega$	V_{out}, B
24	6,8	2,2	
12	750	5,1	
5	4,7	0,27	
10	1,2	0,1	

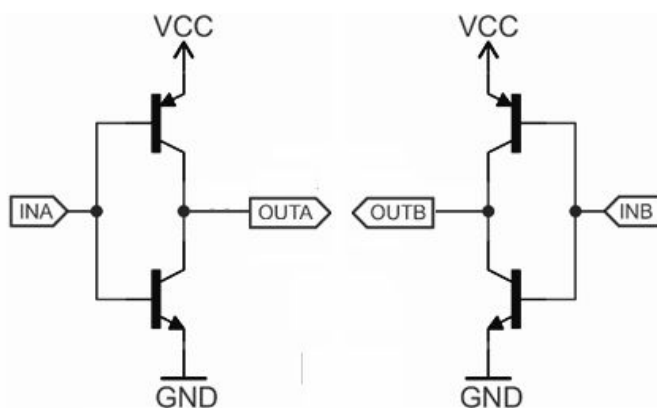
Задача 3

Знайти значення обмежувального опору стандартного світлодіода для напруги живлення U , якщо $U_{LED} = 2 B$, $I_{LED} = 20 mA$.

	U, B	R, Ω
	12	
	24	
	10	
	220	

Задача 4

Доповнити схему Н - моста захисними діодами.



Глава 2. Застосування аналогових та дискретних інтегральних мікросхем

2.1. Аналогові мікросхеми

Інтегральні мікросхеми вміщують багато активних та пасивних елементів на одному кристалі. Це дає можливість створити досить маленькі пристрої, які виконують складні функції.

У залежності від вигляду опрацьовуваних сигналів інтегральні мікросхеми діляться на аналогові і дискретні. Аналогові мікросхеми перетворюють аналогові сигнали, тобто сигнали, що можуть приймати різноманітні значення в межах заданого діапазону.

До аналогових мікросхем відносять операційні підсилювачі, стабілізатори току і напруги, випрямлячі і функціональні перетворювачі, які мають справу з аналоговими сигналами, а саме з сигналами, що можуть мати будь яке значення у зазначених межах.

Найбільш широке використання серед аналогових мікросхем мають схеми на базі операційних підсилювачів (ОП). Операційними називаються підсилювачі постійного струму, які мають коефіцієнт підсилення не менше кількох тисяч у діапазоні частот від нуля до десятків кілогерц.

Аналітичне визначення коефіцієнта передачі реального операційного підсилювача досить складне. Тому звичайно приймається ряд припущень, внаслідок яких реальний ОП замінюється ідеальним.

Для ідеального операційного підсилювача приймаються такі припущення: коефіцієнт підсилення $K_p \rightarrow \infty$, вхідний опір $R_{вх} \rightarrow \infty$, вихідний опір $R_{вих} \rightarrow 0$, смуга пропускання $f \rightarrow \infty$; при нульовому вхідному сигналі вихідна напруга дорівнює нулю.

Операційні підсилювачі мають, як правило, два входи (диференціальний вхід). Один з них інвертуючий вхід, збільшення вхідної напруги на якому призводить до зменшення вихідної напруги. Другий – неінвертуючий вхід, збільшення вхідної напруги на якому призводить до збільшення вихідної напруги. Інвертуючий вхід позначається кружком на вході ОП. Загальною особливістю більшості схем ОП є наявність кола негативного зворотного зв'язку з виходу на інвертуючий вхід. На рис. 2.1 наведені інвертуючий (рис. 2.1, а), неінвертуючий (рис. 2.1, б) та диференціальний (рис. 2.1, в) операційні підсилювачі з відповідними коефіцієнтами підсилення (K_i , K_n та K_d).

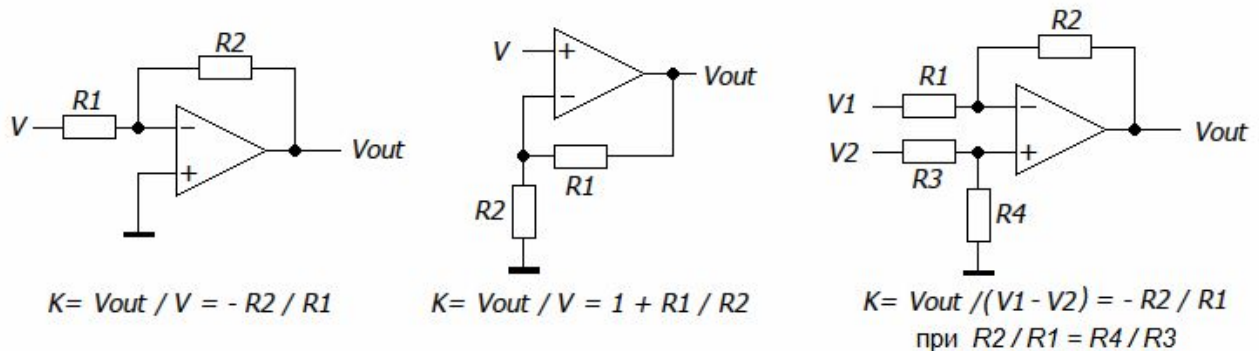


Рис. 2.1. Операційні підсилювачі:

а - інвертуючий, б – неінвертуючий, в - диференціальний

У пристроях керування операційні підсилювачі використовують у вхідних колах для підсилення слабких аналогових сигналів та у складі аналого-цифрових та цифро-аналогових перетворювачів.

2.2. Приклади використання аналогових мікросхем

На рис. 2.2 наведена спрощена схема підключення пьезодатчика з використанням неінвертуючого операційного підсилювача. Коефіцієнт підсилення обирається виходячи з

рівня сигналу на виході датчика та вимог до сигналу, що поступає на схему обробки інформації.

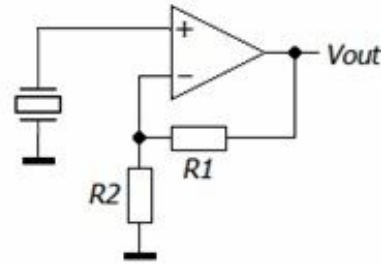


Рис. 2.2. Спрощена схема підключення пьезодатчика

На рис. 2.3 наведена спрощена схема підключення тензометричного датчика у вигляді мостової схеми до операційного підсилювача. Для $\delta \ll 1$ та $R_f \gg R$ маємо $V_0 \approx V_{REF} \delta R_f / 2R$.

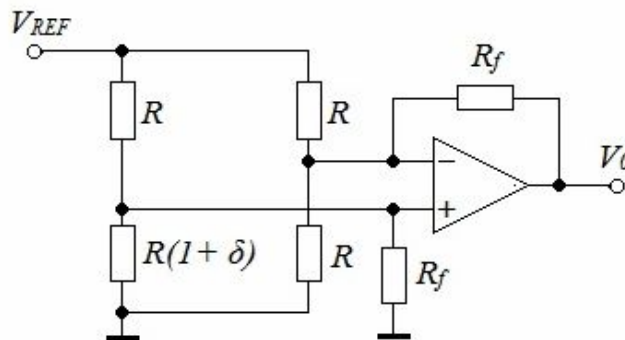


Рис. 2.3. Схема підключення тензометричного датчика у вигляді мостової схеми до операційного підсилювача

Операційні підсилювачі часто використовують як схеми порівняння двох напруг, наприклад, вхідної $V_{вх}$ та опорної $V_{оп}$, з якою здійснюється порівняння (рис. 2.4). Таку схему називають компаратором.

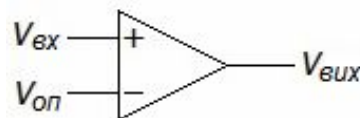


Рис. 2.4. Схема порівняння (компаратор)

Опорна напруга є незмінною в часі, а вхідна напруга — змінюється. Завдяки великому значенню коефіцієнта підсилювання операційного підсилювача без зворотного зв'язку навіть мале значення різниці $V_{вх} \square V_{оп}$ приводить до того, що вихідна напруга досягає мінімального (при $V_{вх} \square V_{оп} < 0$) або максимального (при $V_{вх} \square V_{оп} \geq 0$) значення.

Компаратори широко використовуються в системах контролю та автоматичного керування. Компаратори відносяться до елементів імпульсної техніки, оскільки їх вихідний сигнал є дискретним (цифровим) сигналом.

На рис. 2.5 наведений приклад використання компараторів у схемі визначення наявності об'єкту за допомогою інфрачервоного датчика. Опір фототранзистора залежить від відстані до об'єкту. Потенціометр R3 встановлює рівень сигналу спрацювання компаратора та тим самим, відстань до об'єкту, при якій компаратор спрацює.

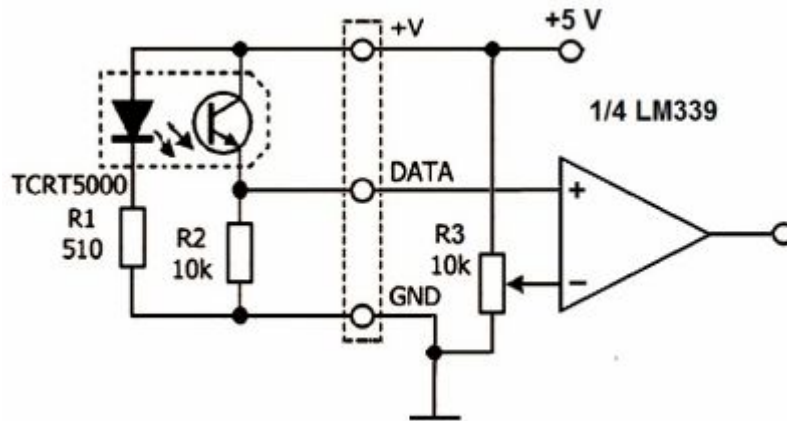


Рис. 2.5. Приклад використання компараторів у схемі визначення наявності об'єкту за допомогою інфрачервоного датчику

2.3. Цифрові мікросхеми

Цифрові інтегральні мікросхеми (цифрові елементи) є основою сучасної цифрової техніки. Електронна цифрова техніка почала розвиватися на основі широкого використання елементної бази імпульсної техніки, а основною причиною її широкого використання стала можливість ефективно реалізувати всі переваги імпульсної техніки завдяки успішному розвитку інтегральної технології. В свою чергу, перехід до цифрової техніки став потужним стимулом для швидкого розвитку мікроелектроніки. Це сприяло тому, що сьогодні цифрова мікроелектронна техніка є основою усіх сучасних обчислювальних пристроїв, які використовуються також в сучасних системах керування.

Особливе місце в сучасних системах та пристроях керування посідають мікропроцесори, їх широке впровадження дозволяє оптимізувати розв'язування цілого ряду дуже важливих задач регулювання, формування та обробки спеціальних сигналів, цифрової фільтрації, використання самонастроювальних та самонавчальних систем тощо.

За функціональним призначенням цифрові елементи умовно можна розподілити на логічні, запам'ятовувальні та допоміжні.

Логічні елементи - це елементарні комбінаційні пристрої з одним виходом, що реалізують, як правило, одну логічну функцію за законами алгебри Буля. На рис. 2.6 наведені основні логічні елементи, логічне І (а), на виході якого маємо 1, якщо на усіх входах 1, логічне АБО (б), на виході якого маємо 1, якщо хоч на одному вході 1, інверсія входу (в), коли вхідний сигнал змінюється на протилежний, інверсія виходу (г), коли вихідний сигнал змінюється на протилежний, RS-тригер (д), вихід якого встановлюється в 1, якщо на вхід S поступає 1, та скидається в 0, якщо на вхід R поступає 1.

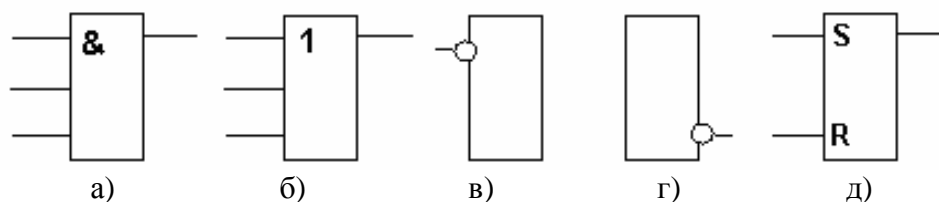


Рис.2.6. Основні логічні елементи

Логічні елементи та запам'ятовуючі елементи утворюють групу універсальних цифрових елементів, на основі яких можна будувати різні цифрові пристрої.

Запам'ятовуючі елементи, які складаються з логічних елементів, створюють елементарні комірки пам'яті, де зберігається один біт інформації ("0" або "1"). Запам'ятовуючі елементи здатні пам'ятати записану інформацію у двійковому вигляді. Для цього використовуються, наприклад, RS-тригери (див. рис. 2.5, д), у якого сигнал "1" на вході S включає вихід (вихідний сигнал "1"), а сигнал "1" на вході R виключає (вихідний сигнал "0"). Декілька тригерів дозволяють створити такі важливі елементи, як регістри.

Регістром називається функціональний пристрій, призначений для приймання, запам'ятовування, перетворення і передачі двійкової інформації. У загальному випадку регістр складається з тригерів і логічних елементів. Число тригерів, які запам'ятовують двійкову інформацію, визначає розрядність регістрів. Введення (запис) інформації у регістр і зчитування (виведення) інформації може бути послідовним або паралельним.

Регістр, призначений для послідовного (біт за бітом) виконання операції запису і зчитування згідно з сигналами синхронізації. Такі регістри ще називають регістром зсуву. Регістр з паралельним входом або виходом записує чи видає двійкову інформацію одночасно з усіх своїх розрядів за один такт сигналу синхронізації. Найбільш швидкодіючими є регістри з паралельними входом (записом) і виходом, бо запис або зчитування всієї інформації у нього відбуваються одночасно. Таки регістри використовуються як регістри пам'яті.

Регістри належать до найбільш поширених функціональних вузлів. Крім зберігання інформації, її зсуву та зчитування, які необхідні для виконання різних арифметичних та логічних операцій над двійковими числами (словами), за допомогою регістрів можна також перетворювати інформацію з одного виду в інший, наприклад, послідовного коду у паралельний або навпаки тощо.

2.4. Приклади використання цифрових мікросхем

Цифрові мікросхеми найчастіше використовують, коли треба здійснити просте логічне керування, або апаратними засобами здійснити різні логічні функції.

Найпростіші ключові схеми разом з підсилювачами потужності та логічними схемами використовують як драйвери для двигунів постійного струму та крокових двигунів.

Такі функції, наприклад, виконує мікросхема L293D (рис. 2.7).

Ця мікросхема являє собою 4-х канальний драйвер з напругою до 25 вольт і струмом до 600 мА. Драйвери включені за схемою Н-моста, що дозволяє пропускати струм через мотор в будь-якому напрямку. Мікросхема дозволяє підключити до 4 колекторних двигунів або 2 крокових моторів.

Умонтовані логічні схеми здійснюють прості функції керування та захисту.

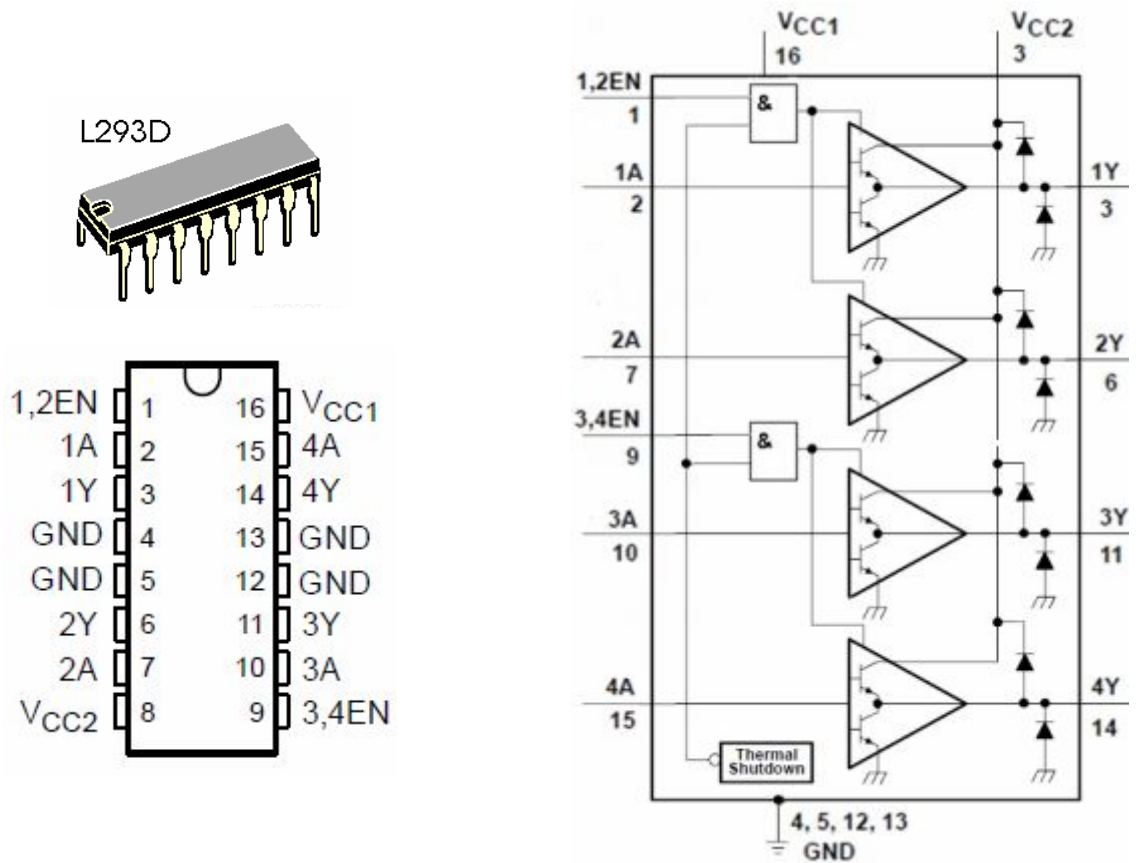


Рис. 2.7. Мікросхема L293D

Додаткові логічні мікросхеми можна використовувати для простих задач логічного керування, наприклад, для керування конвеєром у ручному режимі, якщо сталася поломка керуючого пристрою.

Для керування у ручному режимі використовується режим керування з двома кнопками, коли натиснута кнопка SB1 (SB3), двигун обертається в одну сторону, а коли SB2 (SB4), в іншу. Крім того треба передбачити захист від одночасного натискання на дві кнопки (якщо одноразово натиснути дві кнопки включення обертання двигуна у різних напрямках, схеми логічного І блокують включення двигунів). Схема, яка здійснює таке керування за допомогою логічних елементів І з інверсією входів, наведена на рис. 2.8.

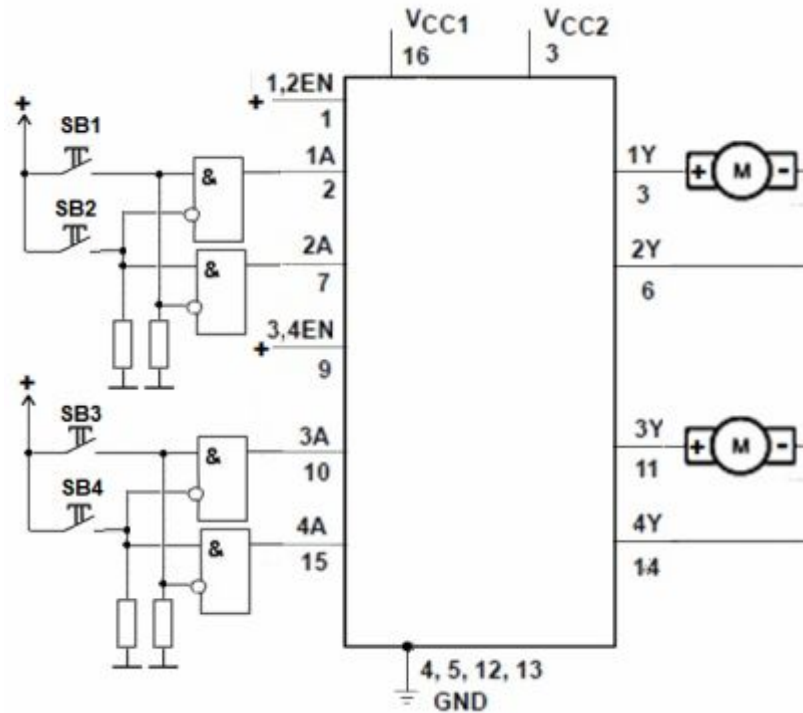


Рис. 2.8. Схема керування двома двигунами у ручному режимі

Контрольні запитання

1. Які підсилювачі називають операційними?
2. Які припущення приймають для ідеального операційного підсилювача?
3. Як визначити коефіцієнт передачі операційного підсилювача, що інвертує сигнал?
4. Як визначити коефіцієнт передачі операційного підсилювача, що не інвертує сигнал?
5. Як визначити коефіцієнт передачі диференціального операційного підсилювача?
6. Які цифрові елементи Ви знаєте?
7. У якому вигляді зберігається інформація у запам'ятовуючих пристроїв?
8. Яку залежність мають вхідні та вихідні сигнали у елементів логічного І та логічного АБО?
9. Який елемент називають регістром?
10. Як передбачити захист від одночасного натискання на дві кнопки?

Завдання до практичних занять

Задача 1

Знайти коефіцієнт підсилювання інвертуючого операційного підсилювача, якщо відомі $R1$ та $R2$.

K	$R1, \text{кОм}$	$R2, \text{кОм}$
	150	1
	10	5
	275	25
	100	0,1

Задача 2

Знайти коефіцієнт підсилювання неінвертуючого операційного підсилювача, якщо відомі $R1$ та $R2$.

<i>K</i>	<i>R1, кОм</i>	<i>R2, кОм</i>
	150	1
	10	5
	275	25
	100	0,1

Задача 3

Знайти коефіцієнт підсилювання диференціального операційного підсилювача, якщо відомі *R1* та *R2*.

<i>K</i>	<i>R1, кОм</i>	<i>R2, кОм</i>
	150	1
	10	5
	275	25
	100	0,1

Задача 4

Скласти схему керування електродвигуном постійного струму, яка включає двигун кнопкою "Пуск" та виключає кнопкою "Стоп", з використанням RS-тригера. Передбачити блокування від одночасного натискання кнопок "Пуск" та "Стоп" (пріоритет кнопки "Стоп").

Глава 3. Аналого-цифрові та цифро-аналогові перетворювачі

3.1. Принцип роботи та основні типи цифро-аналогових перетворювачів

Для забезпечення функцій введення та виводу аналогових даних використовуються аналого-цифрові (АЦП) та цифро-аналогові перетворювачі (ЦАП), які дозволяють використовувати цифрові пристрої для обробки аналогових сигналів и призначені для перетворення значень неперервного сигналу у цифрову форму, а цифрової інформації відповідно в аналогову форму.

До найважливіших параметрів та характеристик АЦП і ЦАП належать розрядність (кількість двійкових розрядів), діапазон ($U_{\min} \dots U_{\max}$) та рівні вхідних (вихідних) сигналів, точність перетворення, час перетворення та встановлення результату.

Точність перетворення ΔU визначається як відношення діапазону вимірювання ($U_{\max} - U_{\min}$) до максимальної кількості рівнів перетвореного сигналу ($q = 2^N$, де N – кількість двійкових розрядів), при цьому діапазон отриманих значень складає $0 \dots 2^N - 1$.

$$\Delta U = (U_{\max} - U_{\min}) / q.$$

Оскільки деякі схеми АЦП можна побудувати на базі ЦАП, доцільно спочатку розглянути найбільш поширені методи та схеми цифро-аналогового перетворення.

АЦП та ЦАП здійснюють перетворення цифрового коду та напруги у визначених діапазонах, тому у разі використання інших фізичних величин треба використовувати відповідні схеми перетворення таких величин у напругу для АЦП або навпаки для ЦАП.

Оскільки максимальне значення діапазону вимірювання відповідає значенню 2^N , то для отримання значення вимірювання для фізичної величини треба здійснити перетворення масштабу.

На рис. 3.1 наведений ЦАП на двійково-зважених резисторах. Це найпростіший щодо будови принципової схеми ЦАП, у якого матриця резисторів складена за принципом відтворення двійкового коду шляхом паралельного підключення двійково-зважених резисторів: номінал кожного резистора – аналог двійкового коду.

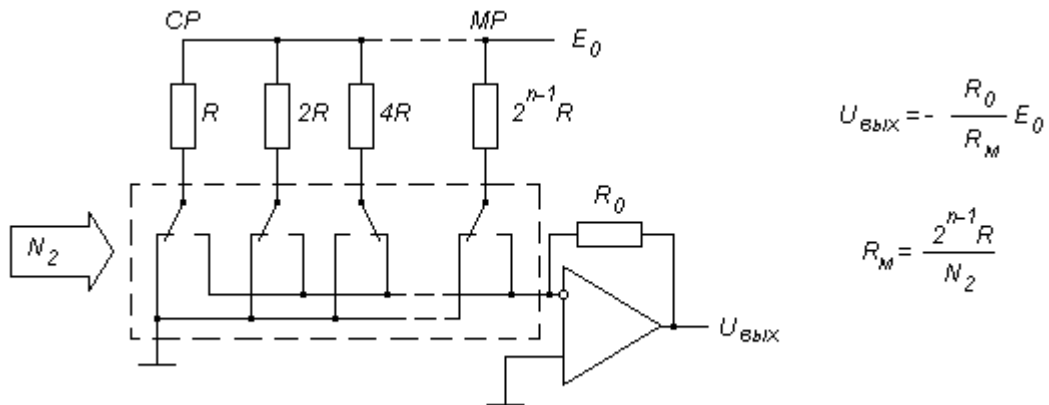


Рис. 3.1. ЦАП на двійково-зважених резисторах

На рис. 3.2 наведений ЦАП на основі матриці резисторів $R - 2R$. Відрізняється схемою матриці резисторів, яка також виконує функцію забезпечення вагового множника на двійкових входах ЦАП. Перевага матриці типу $R - 2R$ у простоті її виготовлення, бо для неї достатні тільки номінали резисторів R і $2R$.

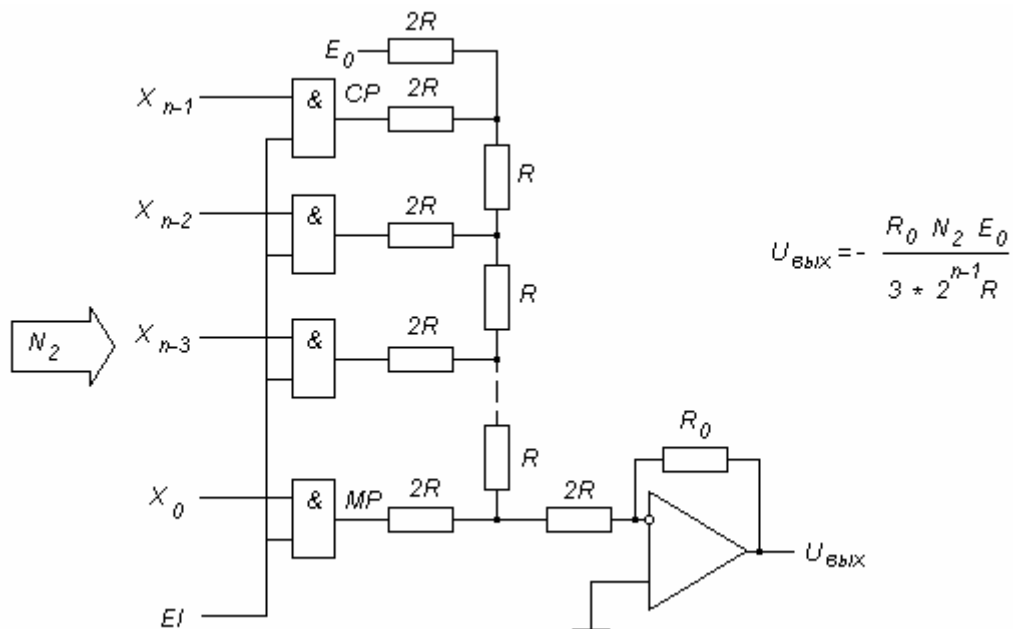


Рис. 3.2. ЦАП на основі матриці резисторів R – 2R

У випадках, коли треба здійснити перетворення цифрового сигналу у напругу для потужних систем, наприклад, для керування швидкістю обертання двигунів постійного струму, на виході ЦАП треба встановлювати підсилювач потужності. При цьому потужність поділяється між навантаженням та силовим елементом, який керує напругою, що приводить до значних втрат.

Для зменшення втрат при цифро-аналоговому перетворенні в сучасних силових перетворювачах використовують сигнали з широтно-імпульсною модуляцією. Вигляд сигналу з широтно-імпульсною модуляцією показаний на рис. 3.3.

Цей сигнал являє собою послідовність імпульсів з постійною частотою та змінною тривалістю. При зміні тривалості імпульсів змінюється середнє значення напруги. Для отримання середнього значення використовують схеми інтегрування. В системах керування такий сигнал подається на індуктивне навантаження (котушки індуктивності, електродвигуни), струм яких здійснює інтегрування напруги.

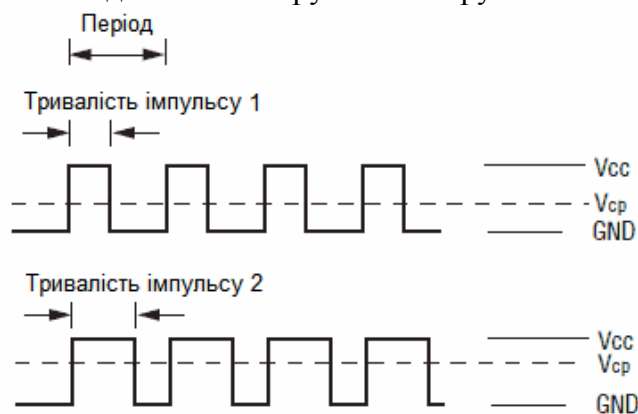


Рис. 3.3. Вигляд сигналу з широтно-імпульсною модуляцією

Формування сигналу з широтно-імпульсною модуляцією виконується за допомогою ключових схем, наприклад, на основі біполярних або IGBT-транзисторів. У таких схемах транзистор, що формує сигнал, знаходиться у двох станах – повністю відкритому, або повністю закритому. Це дає можливість значно зменшити витрати електроенергії

3.2. Принцип роботи та основні типи аналого-цифрових перетворювачів

Розглянемо принцип дії основних типів АЦП.

На рис. 3.4 наведений АЦП послідовного наближення (порозрядного зважування).

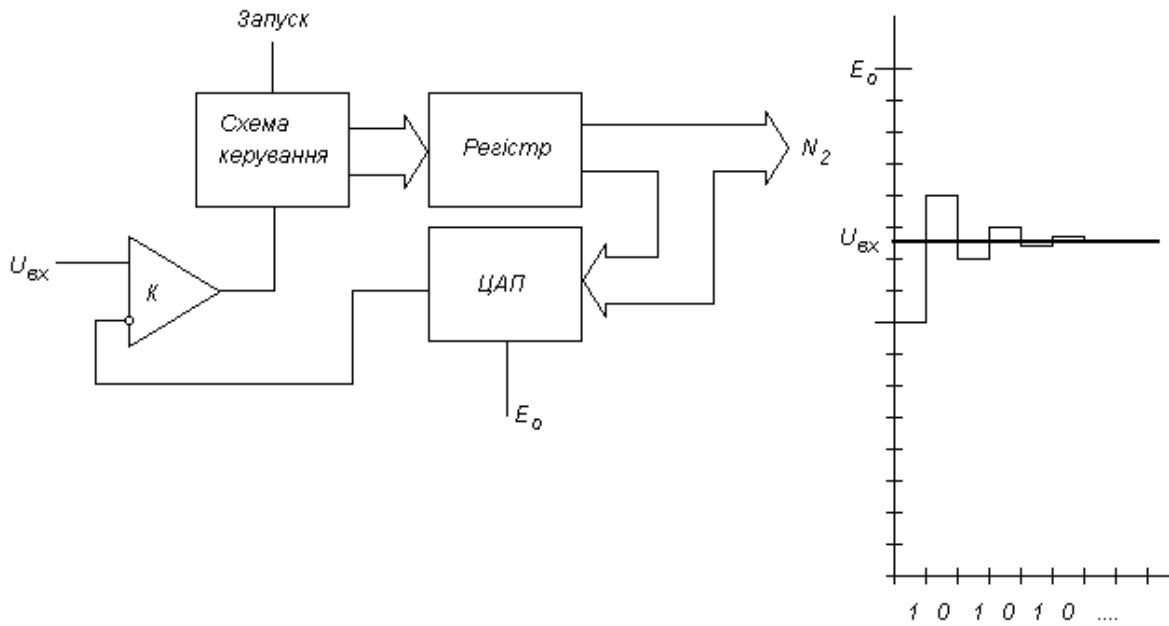


Рис. 3.4. АЦП послідовного наближення

Принцип дії таких АЦП полягає у тому, що схема керування спочатку записує 1 у старший розряд регістру, що задає код для ЦАП. При цьому на виході ЦАП отримаємо напругу, що дорівнює $E_0/2$. Компаратор K порівнює цю напругу з вхідною напругою $U_{вх}$, що перетворюється у двійковий код. Якщо $U_{вх} \geq E_0/2$, то 1 у старший розряд регістру замінюється на 0, а якщо $U_{вх} < E_0/2$, то 1 у старшому розряді регістру залишається.

Після цього схема керування записує 1 у наступний розряд і знову здійснюється порівняння напруги на виході ЦАП з $U_{вх}$ для визначення умови для залишення або скидання 1 у цьому розряді.

Аналогічна дія виконується для всіх розрядів, після чого у регістрі отримаємо двійковий код, що відповідає вхідній напрузі $U_{вх}$.

Сучасні АЦП послідовного наближення відрізняються невеликою вартістю та забезпечують точність перетворення до 16 розрядів з частотою дискретизації до 1 МГц (до 10^6 перетворень за секунду).

Це найбільш поширені АЦП, які використовують, наприклад, у складі однокристальних мікроконтролерів.

На рис. 3.5 наведений АЦП паралельного кодування. Це найбільш швидкодіючий АЦП.

АЦП складається з компараторів, кількість яких визначається кількістю рівнів напруги для сигналу, що перетворюється у n -розрядний двійковий код, і складає $2^n - 1$. Так для 8-розрядного АЦП потребується 255 ($2^8 - 1$) компараторів.

Сигнали порівняння для компараторів задає прецизійний подільник напруги, що має 2^n однакових резисторів.

АЦП паралельного кодування мають низьку точність (8 – 10 розрядів) та високу швидкість перетворення до 5 ГГц ($5 \cdot 10^9$ перетворень за секунду).

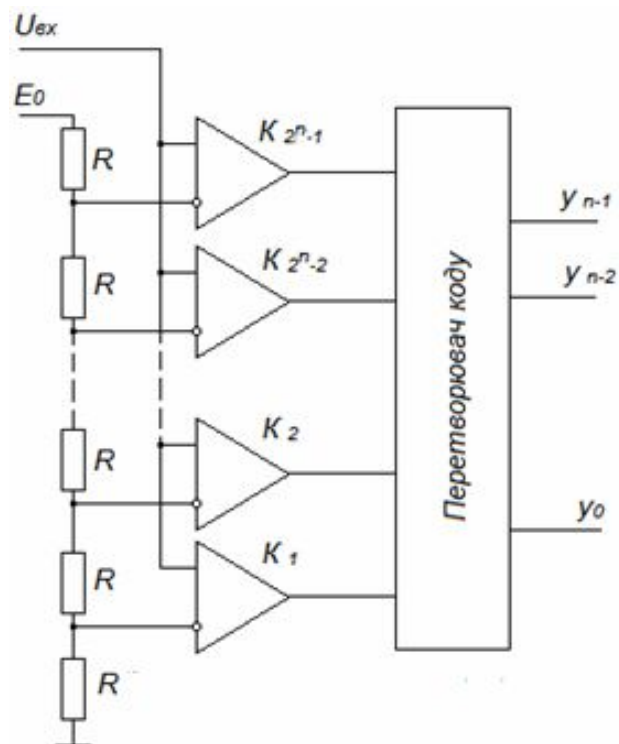


Рис. 3.5. АЦП паралельного кодування

На рис. 3.6,а наведена спрощена схема АЦП подвійного інтегрування.

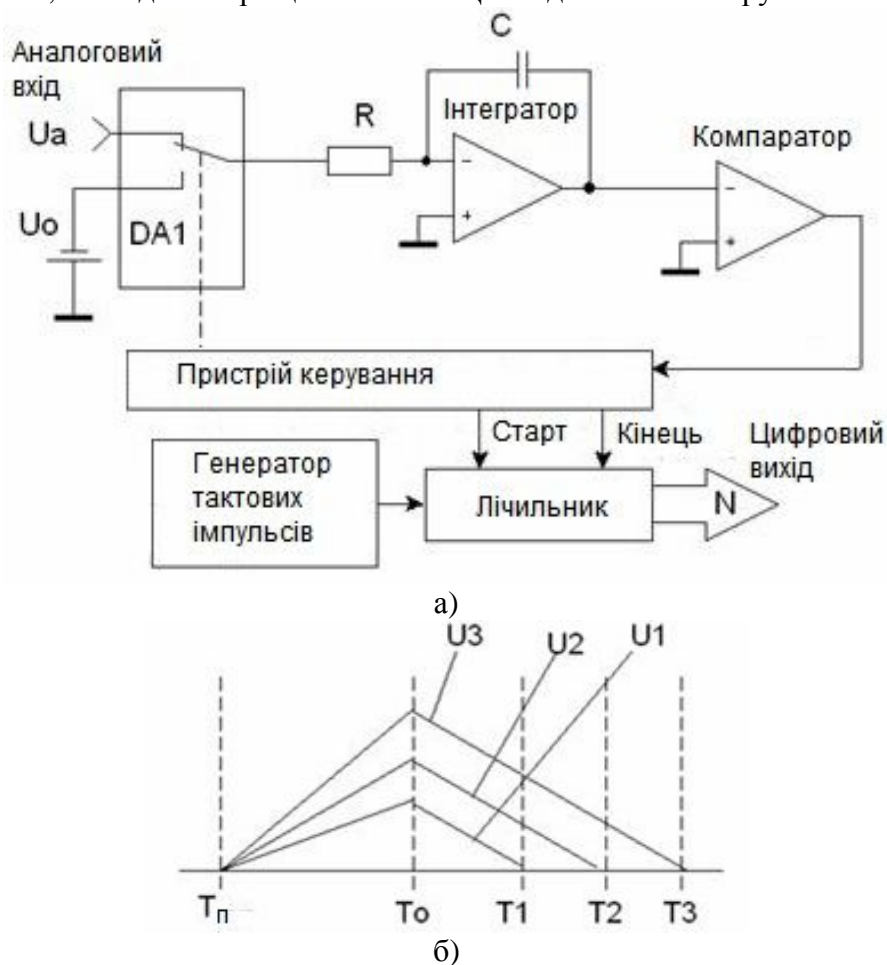


Рис. 3.6. АЦП подвійного інтегрування: спрощена схема (а) та залежності вхідної напруги та часу перетворення (б)

Принцип дії такого АЦП заснований на інтегруванні на конденсаторі C сигналу, що вимірюється, за визначений час (заряд з постійним струмом $I_{вх} = U_a/R$). Після цього конденсатор розряджається з постійним струмом $I_{роз} = U_0/R$, в результаті чого отримуємо час від моменту T_0 до моменту T_i , пропорційний сигналу, який вимірюється (рис. 3.5,б):

$T_0 - T_1$ для вхідної напруги U_1 ;

$T_0 - T_2$ для вхідної напруги U_2 ;

$T_0 - T_3$ для вхідної напруги U_3 .

В момент T_0 запускається лічильник, який підраховує кількість імпульсів і таким чином визначає тривалість часу розряду конденсатора до моменту T_i (схема перетворення часу у послідовність імпульсів), і після зупинки видає значення сигналу у цифровому вигляді.

Цей АЦП забезпечує високу точність та стійкість до помилок завдяки процесам усереднення та згладжування швидкозмінних шумів. Так час інтегрування тривалістю 20 мс дає можливість подавляти перешкоди з частотою 50 Гц (напруга живлення).

3.3. Приклади використання аналого-цифрових та цифро-аналогових перетворювачів в системах керування

АЦП та ЦАП використовуються як у вигляді окремих мікросхем, так і у складі систем керування.

На рис. 3.7 наведений АЦП у вигляді мікросхеми MCP3008, яка має вісім каналів. Розрядність кожного каналу - 10 біт, тобто мікросхема розрізняє 1024 градації вхідного сигналу. Діапазон робочих напруг - від 2,7 В до 5,5 В. Це дозволяє використовувати мікросхему з пристроями як на 5 В, так і на 3,3 В. Час перетворення складає 5 мкс для діапазону 5 В.



Рис. 3.7. АЦП у вигляді мікросхеми MCP3008

Такі мікросхеми використовують з мікроконтролерами, що не мають умонтованих АЦП.

Сучасні мікроконтролери можуть мати умонтовані АЦП та ЦАП. Так, наприклад, мікроконтролери Arduino, що будуть розглянуті далі, мають у своєму складі декілька каналів АЦП та можуть видавати сигнали з широтно-імпульсною модуляцією (ШИМ) для формування аналогових сигналів. Так Arduino Mega має 16 аналогових входів, що реалізуються за допомогою 10-розрядних АЦП, які мають час перетворення 100 мкс, та 15 виходів з широтно-імпульсною модуляцією (рис. 3.8).



Рис. 3.8. Arduino Mega

Сучасні промислові пристрої керування мають у своєму складі модулі вводу та виводу аналогових сигналів або аналогові входи умонтовані в процесорні модулі (компактні процесорні модулі, що мають умонтовані цифрові та аналогові входи та виходи).

На рис. 3.9 наведені модулі аналогових входів у складі промислового контролера S7-300.



Рис. 3.9. Модулі аналогових входів у складі промислового контролера S7-300 (а) та входи аналогових сигналів у складі компактного процесора (б)

Такі модулі найчастіше використовують АЦП подвійного інтегрування, що дозволяє отримати високу точність вимірювання (до 16 біт), та дозволяють вимірювати різні фізичні величини (напругу, струм, опір) з різними діапазонами. Окремі модулі дозволяють безпосередньо підключати різні датчики, наприклад, датчики температури, тензодатчики тощо.

Контрольні запитання

1. Які працюють ЦАП на двійково-зважених резисторах?
2. Які працюють ЦАП на основі матриці резисторів?
3. Як працюють перетворювачі на основі широтно-імпульсної модуляції?
4. Як працюють АЦП послідовного наближення?
5. Як працюють АЦП паралельного кодування?
6. Які працюють АЦП подвійного інтегрування?
7. У якому вигляді представляється цифрова інформація у АЦП та ЦАП?
8. Як визначити точність перетворення АЦП та ЦАП?
9. Як пов'язати цифрову інформацію у АЦП та ЦАП з значеннями фізичних величин?
10. Як здійснюється вимірювання аналогових величин у промислових контролерах?

Завдання до практичних занять

Задача 1

Знайти точність перетворення напруги у цифровий код для АЦП з вказаними *розрядністю та діапазоном вимірювання*.

ΔU визначається як відношення діапазону вимірювання (U_{\max} □ U_{\min}) до максимальної кількості рівнів перетвореного сигналу ($q = 2^N$, де N

ΔU	(U_{\max} □ U_{\min}), В	N
	10	10
	5	12
	24	16

Задача 2

Діапазон цифрових значень *n* модулів аналогових входів промислових контролерів S7-300 складає від - 27648 до + 27648. Яке перетворення масштабу K_m треба зробити, щоб отримати значення *для вказаних значень діапазону* (U_{\max} □ U_{\min}) *та точності вимірювання* $\Delta U_{\text{вим}}$.

K_m	(U_{\max} □ U_{\min}), В	$\Delta U_{\text{вим}}$, мВ
	10	1

	5	1
	24	1

Задача 3

Визначити час інтегрування для АЦП подвійного інтегрування, який може подавляти перешкоди з *частотою 50 та 60 Гц*.

Глава 4. Архітектура мікропроцесорних керуючих пристроїв. Уявлення даних у керуючих обчислювальних пристроях

4.1. Класифікація мікропроцесорних пристроїв

Мікропроцесорні пристрої являють собою обчислювальний пристрій з програмним керуванням, що складається з однієї або декількох мікросхем, та є основою сучасних обчислювальних комплексів та систем керування.

Мікропроцесорні пристрої поділяються на мікропроцесорні комплекти та однокристальні мікроконтролери.

Мікропроцесорні комплекти являють собою набір мікросхем, кожна з котрих є функціональним модулем обчислювального пристрою.

Однокристальні мікроконтролери мають на одному кристалі усі функціональні модулі обчислювального пристрою.

Мікропроцесорні комплекти використовують у складних обчислювальних комплексах, наприклад, у персональних комп'ютерах.

Однокристальні мікропроцесори найчастіше являються складовою частиною пристроїв різноманітної складності, що мають умонтовані системи керування, наприклад, сучасні датчики з програмною обробкою інформації, регульовані приводи, технологічне обладнання, побутова техніка тощо.

Мікропроцесорні пристрої можна поділити на універсальні, які мають досить великий набір функцій введення, виведення та обробки даних та можуть бути запрограмованими за допомогою засобів, що доступні для широкого кола користувачів (відкрита система), а також спеціалізовані, які мають структуру та програмне забезпечення, що призначено для виконання конкретних функцій, наприклад, керування побутовою технікою, технологічним обладнанням тощо. При цьому програмне забезпечення не доступно для користувачів (закрита система).

4.2. Архітектура керуючих обчислювальних пристроїв

Обчислювальні пристрої являють собою багаторівневі системи з визначеною структурою кожного рівня. Тому визначаючи структуру обчислювального пристрою використовують термін архітектура.

Архітектура обчислювального пристрою це набір типів даних, системи команд, логічної побудови і характеристик кожного окремо взятого рівня.

Серед обчислювальних пристроїв, найбільшого поширення отримали два типи архітектури: принстонська (фон Неймана) і гарвардська. Обидві архітектури виділяють такі основні вузли обчислювальних пристроїв: центральний процесор, пам'ять та пристрої введення-виведення даних. Різниця полягає в структурі пам'яті: в принстонській архітектурі програми і дані зберігаються в одному запам'ятовуючому пристрої і передаються в процесор одним каналом (рис. 4.1), тоді як гарвардська архітектура передбачає окремі запам'ятовуючі пристрої і канали передачі для команд і даних (рис. 4.2).



Рис. 4.1. Принстонська архітектура



Рис. 4.2. Гарвардська архітектура

Принстонська архітектура найчастіше використовується в універсальних обчислювальних пристроях, наприклад, у персональних комп'ютерах, оскільки вони мають дуже об'ємне програмне забезпечення. Це пов'язано зокрема з тим, що універсальні обчислювальні пристрої призначені для роботи безпосередньо з людьми, тому для відображення графічної інформації використовують кольорові дисплеї з високою роздільною здатністю, що потребує складних програм та великих обсягів пам'яті для обробки зображень у реальному часі. Тому програмне забезпечення зберігається на зовнішніх носіях інформації (наприклад, на жорстких дисках), а у разі потреби відповідні програмні компоненти завантажуються у пам'ять обчислювального пристрою.

Гарвардська архітектура найчастіше використовується в пристроях керування, оскільки мають обсяги програм значно менші, ніж в обчислювальних пристроях, крім того ця програма завантажуються на етапі проектування та налагодження пристроїв та використовується довгий час без змін. Тому програмне забезпечення в пристроях керування зберігається у програмній пам'яті, яка являє собою постійну пам'ять з можливістю перезавантаження.

4.3. Уявлення даних у керуючих обчислювальних пристроях

Внутрішня структура процесора має справу з логічними сигналами, тому Стан елементів процесора, запам'ятовуючих пристроїв та інших компонент обчислювальних пристроїв має 2 значення, що умовно позначається як 0 або 1. Тому уся внутрішня інформація визначається як двійкова система числення, що має набір цифр 0 та 1.

Оскільки двійкова система числення не зручна для використання, то для більш простого запису двійкових чисел використовується шістнадцяткова система числення.

Іноді датчики використовують двійкові числа для визначення десяткових чисел у так званому двійково-десятковому форматі (одна десяткова цифра подається як чотири двійкових цифри).

Порівняння між двійковими, десятковими, шістнадцятковими та двійково-десятковими числами наведено у табл. 4.1.

Таблиця 4.1

Числа			
Десяткові	Двійкові	Шістнадцяткові	Двійково-десяткові
0	0000 0000	0	0000 0000
1	0000 0001	1	0000 0001
2	0000 0010	2	0000 0010
3	0000 0011	3	0000 0011

4	0000 0100	4	0000 0100
5	0000 0101	5	0000 0101
6	0000 0110	6	0000 0110
7	0000 0111	7	0000 0111
8	0000 1000	8	0000 1000
9	0000 1001	9	0000 1001
10	0000 1010	A	0001 0000
11	0000 1011	B	0001 0001
12	0000 1100	C	0001 0010
13	0000 1101	D	0001 0011
14	0000 1110	E	0001 0100
15	0000 1111	F	0001 0101
16	0001 0000	10	0001 0110

Інформація, яку уміщує один двійковий розряд, називають бітом.

Вісім бітів створюють один байт - B.

Два байта утворюють слово - W.

Чотири байта утворюють подвійне слово - DW (D).

Більш значна інформація вимірюється в кіло- (2^{10} байтів), мега- (2^{20} байтів) або в гігабайтах (2^{30} байтів).

Для подання негативних чисел у двійковому коді використовується так званий доповняльний код, який отримується з прямого коду шляхом створення оберненого коду (заміною всіх нулів одиницями, а одиниць нулями) з подальшим створенням доповняльного коду з оберненого шляхом додавання одиниці до молодшого розряду. Ознакою негативного числа є одиниця у старшому розряді.

Наприклад, для двійкового числа, яке має 8 розрядів, позитивні числа мають значення від 0000 0000 до 0111 1111, де $0 = 0000\ 0000$, $1 = 0000\ 0001$, ... , $127 = 0111\ 1111$, а негативні числа мають значення від 1000 0000 до 1111 1111, де $-1 = 1111\ 1111$, $-2 = 1111\ 1110$, ... , $-128 = 1000\ 0000$

Таке подавання негативних чисел дає можливість представлення даних у вигляді цілих чисел зі знаком.

Така форма представлення чисел називається представленням з фіксованою комою.

В системах керування використовують 16- та 32-розрядні форми представлення цілих чисел з фіксованою комою.

У першому випадку діапазон чисел складає від -32768 до +32767 (Тип даних **Integer**, можливі позначення *int* або **INT**)

У другому випадку діапазон чисел складає від -2147383648 до +2147383647 (Тип даних **Double Integer**, можливі позначення *long* або **DINT**).

Якщо діапазон чисел з фіксованою комою недостатній, або потрібні дробові числа, наприклад, при використанні тригонометричних операцій, то числа представляють у формі з рухомою комою (можливе позначення *float* або **Real**).

Формат з рухомою комою у різних комп'ютерних системах може бути представлений різними способами, наприклад, у форматі *float*: 1,225E3, що означає $1,225 \cdot 10^3$.

Число A_n у формі з рухомою комою (ПК) може бути записане так:

$$A_n = m q^p,$$

де m - мантиса числа A ;

q - основа системи числення, для обчислювальних пристроїв $q = 2$;

p - порядок числа A (вказує на положення коми в числі)

Формат машинного зображення числа з рухомою комою повинен містити знакові частини і поля для мантис і порядку, наприклад, так:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | ← номер розряду

функціонуванням усіх складових компонент обчислювального пристрою.

У визначені програмою моменти часу центральний процесор опитує через пристрій введення інформації стан датчиків зовнішньої інформації і передає за допомогою пристрою виводу інформації команди керування діями виконавчих елементів об'єкта керування.

Центральний процесор має в собі реєстр команд та пристрій декодування, що визначають, яку дію виконує команда, арифметично-логічний пристрій, який виконує арифметичні та логічні команди, внутрішній запам'ятовуючий пристрій для підвищення швидкості виконання команд, лічильник адреси та інші. Важливим параметром центрального процесора є його розрядність, яка визначає найбільшу кількість двійкових розрядів - бітів, яку обробляє команда. Більшість сучасні процесорів має розрядність 32 або 64 біта. Процесори для пристроїв керування можуть мати розрядність 8 або 16 бітів.

Система команд мікропроцесорних пристроїв у загальному випадку включає такі групи команд:

команди пересилки, за допомогою яких провадиться обмін даними усередині обчислювального пристрою;

арифметичні команди над двійковими числами, що включають команди складання, віднімання, множення і ділення;

команди логічних операцій над двійковими числами, виконувани порозрядно (логічне И, логічне АБО, виключне АБО);

команди переходу, що включають умовні і безумовні команди передачі керування по зазначеній адресі і команди виклику підпрограм;

спеціальні команди.

Програма та всі необхідні дані зберігається у **запам'ятовуючому пристрої** або **пам'яті обчислювального пристрою** у двійковому вигляді. Пам'ять обчислювального пристрою звичайно складається з двох частин: оперативної і постійної пам'яті.

Оперативна пам'ять або **оперативний запам'ятовуючий пристрій** (ОЗП) - це пристрій, де розміщаються програми під час виконання, а також дані, які виконуються в програмі. Оперативна пам'ять характеризується більш високою швидкістю запису і читання і меншим обсягом, чим зовнішня пам'ять на магнітному носії інформації. При вимиканні електроживлення обчислювального пристрою вміст оперативного запам'ятовуючого пристрою не зберігається.

Оперативна пам'ять може бути **статичною** чи **динамічною**.

Статичні ЗП мають елементи запам'ятовування, що можуть мати два стабільних стани, наприклад, тригери, фізичний стан яких під час зчитування даних не руйнується. Статичні ЗП зберігають інформацію, поки є живлення. Сучасні статичні ЗП використовують КМОП-транзистори і тому майже не споживають енергію у режимі очікування.

В система керування найчастіше використовується статична пам'ять.

Динамічні ЗП побудовані на основі запам'ятовуючого елемента, що зберігає свій стан тільки певний проміжок часу. Таким елементом може бути конденсатор, в якому інформація зберігається у формі наявності або відсутності заряду. Оскільки конденсатор постійно розряджається, то записана в ньому інформація потребує періодичного відновлення.

В обчислювальних пристроях найчастіше використовується динамічна пам'ять.

Постійна пам'ять або **постійний запам'ятовуючий пристрій** (ПЗП) являє собою різновид пам'яті обчислювального пристрою, вміст котрої постійній, тобто зберігається при вимиканні. Запис нової інформації в ПЗП можлива лише за допомогою додаткових засобів або з відносно малою швидкістю, а читання може провадитися з великою швидкістю. Існують різні типи ПЗП, але в наш час найчастіше використовується постійна пам'ять з електричним записом даних (FLASH пам'ять), що дає можливість безпосередньо здійснювати запис даних. ПЗП на FLASH пам'яті відрізняються від ОЗП відносно малою швидкістю запису даних. У постійній пам'яті обчислювальних пристроїв зазвичай знаходяться програми і дані, які обслуговують роботу апаратури обчислювального

пристрою. У постійній пам'яті пристроїв керування зазвичай знаходяться уся програма і дані, які зберігаються довгий час.

ОПЗ та ПЗП разом являють собою внутрішню пам'ять обчислювального пристрою.

Крім того у складі обчислювального пристрою може бути зовнішня пам'ять, яка виконується на магнітних носіях інформації і має значно більший розмір, ніж внутрішня пам'ять. В останній час зовнішня пам'ять також може використовувати FLASH пам'ять.

Програма пристроїв керування звичайно знаходиться у внутрішній пам'яті, тому вони часто не потребують зовнішньої пам'яті.

В сучасних обчислювальних пристроях прийнята організація пам'яті, при якій адреса розподілена по байтам. Це означає, що кожний байт пам'яті має свою адресу.

Обсяг пам'яті обчислювальних пристроїв вимірюється в кіло- (2^{10} байтів), мега- (2^{20} байтів) або в гігабайтах (2^{30} байтів). Обсяг пам'яті керуючих пристроїв складає від десятків кілобайтів до одиниць мегабайтів, а у персональних комп'ютерів – гігабайти.

Пристрої вводу-виводу забезпечують обмін даними між компонентами обчислювального пристрою або між обчислювальним пристроєм та зовнішнім середовищем і відрізняються режимами обміну інформацією. У загальному випадку обмін даними між компонентами обчислювального пристрою у залежності від того, який з пристроїв є головним, може здійснюватися в таких режимах: програмному, переривання програми і прямого доступу до пам'яті.

Програмний режим самий простий і застосовується частіше усього. Обмін інформацією в цьому режимі здійснюється з ініціативи процесора в програмному режимі, за допомогою команд обміну від керуючої програми. Отримавши команду від керуючої програми, процесор встановлює на магістралі адресу відповідного пристрою вводу-виводу а потім читає або записує дані за допомогою магістралі даних. У такий спосіб виконується обмін інформацією між процесором і пристроями вводу-виводу.

Під час своєї звичайної роботи центральний процесор з заданою періодичністю виконує обмін інформацією незалежно від потреби такої дії.

Пристрій вводу-виводу інформації, при необхідності здійснює перетворення вхідних сигналів до цифрового вигляду для обробки в обчислювальному пристрої, або цифрові сигнали до вигляду, що відповідає структурі вихідних сигналів об'єкта та пристрою керування.

У **режимі роботи з перериванням** обмін даними відбувається з ініціативи зовнішнього пристрою. У випадку одержання сигналу переривання процесор припиняє виконання поточної програми і переходить на підпрограму обслуговування пристрою, який запитує переривання. Після завершення виконання підпрограми обслуговування процесор повертається до того кроку перерваної основної програми, на якому він зупинився, отримавши сигнал переривання.

Режим прямого доступу до пам'яті є самим швидким режимом передачі інформації, яку здійснюють між собою пам'ять і зовнішній пристрій. У цьому режимі ініціатором обміну є периферійний пристрій, причому обмін даними здійснюється без програмного керування з боку процесора. Після одержання вимоги прямого доступу процесор перериває виконання поточної програми і передає керування процесом передачі інформації зовнішньому пристрою, що виставив вимогу переривання. Цей пристрій, у свою чергу, самостійно здійснює обмін з пам'яттю по внутрішнім магістралям, задаючи адресацію і керуючи розмірами переданого масиву даних по своїй власній програмі. Після завершення обміну з пам'яттю зовнішній пристрій повертає керування магістралями процесору, що продовжує виконувати перервану програму. Звичайно режим прямого доступу використовується тоді, коли необхідно обмінюватися великими масивами інформації за мінімально короткий час, наприклад при передачі даних між накопичувачами на магнітних дисках і запам'ятовувачим пристроєм обчислювального пристрою.

Функції вводу-виводу в пристроях керування частіше всього здійснюється у програмному режимі, режим роботи з перериванням використовуються тільки тоді, коли

програмний режим не забезпечує потрібну швидкість передачі даних..

Всі функціональні пристрої, що входять до складу обчислювального пристрою взаємодіють між собою по магістралях адреси, даних і керування, що утворюють загальний канал, називаний системною магістраллю.

Для зчитування необхідної інформації на магістралі адресу встановлюють адресу (номер) опитуваного пристрою, у якому прийнята адреса зрівнюється з його власною адресою і у випадку, коли вони рівні, здійснюється передача інформації в канал магістралі даних.

Слід зазначити, що поділ каналу зв'язку між складовими елементами обчислювального пристрою на магістралі адреси, даних і переривання не є обов'язковим. У деяких типах обчислювальних пристроїв магістраль адреси об'єднана з магістраллю даних і поділ інформації про адреси і даних здійснюється за часом. У цьому випадку пристрій, що ініціює обмін, встановлює в загальній магістралі спочатку потрібну адресу і потім передає дані.

Пристрої керування часто мають різні магістралі для обміну даними усередині процесорного модулю та для обміну даними між модулями.

Контрольні запитання

1. Як поділяють мікропроцесорні пристрої за архітектурою?
2. З яких елементів складається спрощена структура обчислювального пристрою?
3. Чим відрізняються двійкова та десяткова системи числення?
4. Для чого використовується шістнадцяткова система числення?
5. Чим відрізняються позитивні та негативні числа у двійковій системі числення?
6. Які функції виконує центральний процесор?
7. Які основні групи команд включає система команд мікропроцесорних пристроїв?
8. Які функції виконує запам'ятовуючий пристрій?
9. Які функції виконує пристрій введення-виведення?
10. Які режими введення-виведення даних використовують обчислювальні пристрої?

Завдання до практичних занять

Задача 1

Представити у шістнадцятковій формі такі двійкові числа:

10101010, 010101010, 10001111, 10110110, 00111100, 00001110.

Задача 2

Представити у двійковій формі такі числа

250, - 250, 1024, - 1024, 27746, - 27746.

Задача 3

Визначити діапазон представлення даних у двійковій та десятковій формі, які отримує 12-розрядний АЦП для біполярних вхідних сигналів (представлення даних як цілі числа зі знаком).

Глава 5. Пристрої керування на основі мікропроцесорної техніки

5.1. Однокристальні мікроконтролери як основа для вбудованих пристроїв керування

Вбудована система (англ. embedded system) це спеціалізована комп'ютерна система або обчислювальний пристрій, призначений для виконання обмеженої кількості функцій, часто, з обмеженнями реального часу.

Відмінними властивостями вбудованих систем управління є їх компактність і можливість автономного використання в складі комплексних систем управління. Тому основою для автономних вмонтованих пристроїв керування найчастіше є однокристальні мікроконтролери.

Однокристальний мікроконтролер це закінчений обчислювальний пристрій, що виконується у вигляді однієї мікросхеми. Ці пристрої призначені в основному для вирішення завдань управління і часто мають вбудовані функції опитування різних датчиків (цифрові та аналогові входи) і видачі регульованих управляючих впливів (цифрові виходи або виходи з широтно-імпульсною модуляцією).

Як уже зазначалося, в пристроях керування найчастіше використовується гарвардська архітектура, оскільки мають обсяги програм значно менші, ніж в обчислювальних пристроях, та ці програми використовується довгий час без змін.

Однокристальні мікроконтролери можуть бути універсальними та спеціалізованими.

Спеціалізовані однокристальні мікроконтролери призначені для виконання функцій керування та обробки даних визначених пристроїв. Користувач, як правило, не має доступу до програмного забезпечення таких пристроїв, а налагодження найчастіше здійснюється шляхом встановлення значень для різних параметрів. Спеціалізовані однокристальні мікроконтролери найчастіше потребують спеціальних апаратних та програмних засобів налагодження. Оскільки проектування та виготовлення мікросхем такого рівня відрізняється дуже високою вартістю, їх варто використовувати при великих обсягах виготовлення мікросхем.

Універсальні однокристальні мікроконтролери призначені для використання широким колом користувачів. При цьому користувач, як правило, самостійно розробляє апаратне та програмне забезпечення, тому для універсальних мікроконтролерів найчастіше здійснюється за допомогою універсальних апаратних та програмних засобів проектування та налагодження. Як правило, такі засоби відрізняються низькою вартістю, або зовсім безкоштовні, а програмні засоби проектування доступні в Інтернеті.

5.2. Класифікація й основні типи однокристальних мікроконтролерів

Класифікацію однокристальних мікроконтролерів можна здійснити за їхніми основними характеристиками, такими як, наприклад, розрядність процесора, обсяг пам'яті, наявність додаткових засобів введення, виведення та обробки даних, наявність таких додаткових функцій як таймери, лічильники, тощо.

Оскільки однокристальні мікроконтролери здебільше використовують логічне керування, що не потребую складних алгоритмів обробки даних, то найчастіше в них використовують 8-розрядні процесори. У разі потреби можуть використовуватись 16- та 32-розрядні процесори.

Спрощена структурна схема таких мікроконтролерів наведена на рис. 5.1.

Найчастіше однокристальні мікроконтролери мають такі функціональні елементи:

- генератор тактових імпульсів (ГТІ), що синхронізує роботу мікроконтролера, та пристрій управління мікроконтролера (ПУМК);
- арифметико-логічний пристрій (АЛП) - призначений для виконання арифметичних і логічних операцій, та в сукупності з регістрами загального призначення виконує функції процесора;

- пам'ять програм - виконана у вигляді постійної пам'яті з можливістю перепрограмування, що призначена для запису та зберігання програми управління мікроконтролером;
- пам'ять даних застосовується як пам'ять для зберігання усіх можливих констант, табличних значень функцій і т.д.

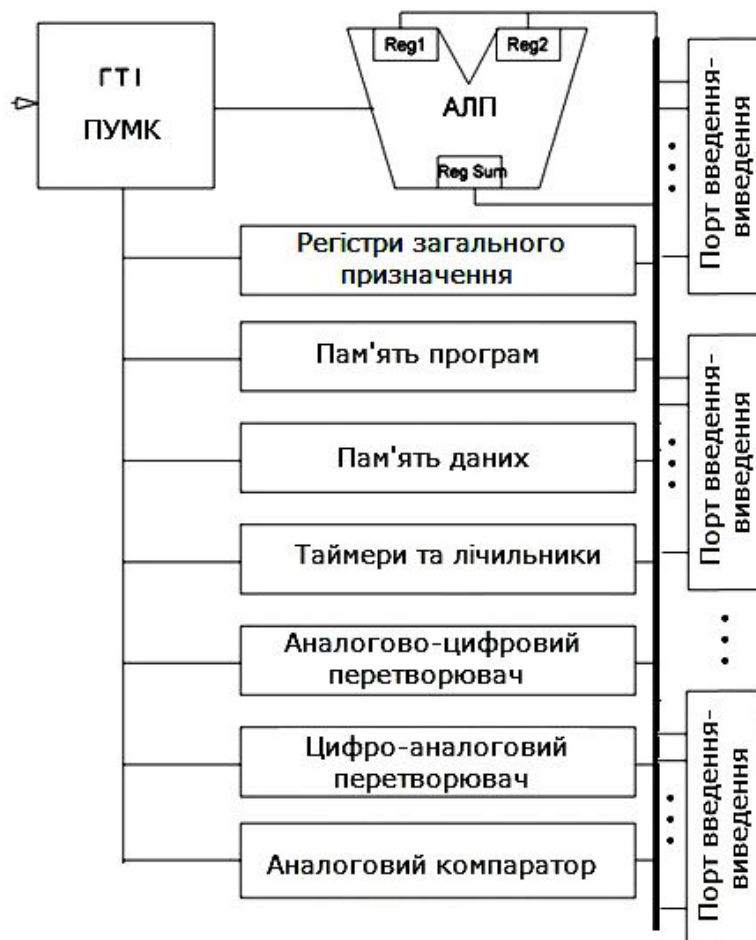


Рис. 5.1. Спрощена структурна схема мікроконтролерів

Мікроконтролер в своєму складі може мати і інші допоміжні елементи:

- таймери та лічильники застосовуються для здійснення затримок і установки інтервалів часу та підрахунку подій;
- аналогово-цифровий перетворювач (АЦП) необхідний для введення аналогового сигналу в мікроконтролер шляхом перетворювання його в цифровий вид;
- цифро-аналоговий перетворювач (ЦАП) виконує зворотну функцію, тобто сигнал з цифрового виду перетворює в аналоговий вигляд;
- аналоговий компаратор - призначений для порівняння двох аналогових сигналів на його входах.

Одним з перших однокристальних мікроконтролерів був мікроконтролер i8051 (MCS-51) фірми Intel (рис. 5.2).



Рис. 5.2. Мікроконтролер i8051

Мікропроцесор i8051 є однокристальним мікроконтролером гарвардської архітектури, що виконаний по n-МОН або КМОН-технології, та містить у собі:

- 8-бітний процесор з підтримкою логічних (булевих) операцій над окремими бітами,
- до 4096 байт вбудованої пам'яті програм (доступної тільки для читання),
- до 256 байт вбудованої пам'яті даних (доступної для читання і запису),
- підтримку адресного простору у 64 Кб для пам'яті програм і 64 Кб для пам'яті даних,
- два-три 16-бітних таймерів/лічильників,
- двосторонній універсальний асинхронний прийомо-передавач УАПП,
- 32 двосторонні лінії введення-виведення цифрових сигналів (у вигляді чотирьох 8-розрядних портів),
- генератор тактової частоти.

Вдалий набір периферійних пристроїв, можливість гнучкого вибору зовнішньої або внутрішньої програмної пам'яті і прийнятна ціна забезпечили цьому мікроконтролеру успіх на ринку.

На сьогоднішній день існує більше 200 модифікацій мікроконтролерів, сумісних з i8051, що випускаються двома десятками компаній.

В даний час широко використовуються мікроконтролери фірми Atmel, наприклад, мікроконтролери AVR (рис. 5.3).



Рис. 5.3. Мікроконтролери AVR

Мікроконтролери AVR мають такі основні характеристики:

- 8-розрядний процесор з широким набором команд;
- до 100 двонапрямлених ліній введення-виведення;
- аналого-цифрові перетворювачі з роздільною здатністю 12 біт і до 2 млн вибірок в секунду;
- широкий набір комунікаційних функцій, включаючи можливість підключення USB;
- 16-бітові таймери / лічильники з каналами порівняння;
- функції переривання тощо.

Мікроконтролери AVR мають гарвардську архітектуру, при якій для програми і даних використовуються різні пристрої пам'яті.

Ці мікроконтролери є універсальними пристроями.

Для розробки програмного забезпечення однокристальних мікроконтролерів використовуються персональні комп'ютери, на яких встановлено спеціальні мови програмування.

Після розробки програми вона перетворюється в машинні коди і за допомогою спеціальних пристроїв програмування (рис.5.4) завантажується в пам'ять контролера, після чого він встановлюється на робочій платі.

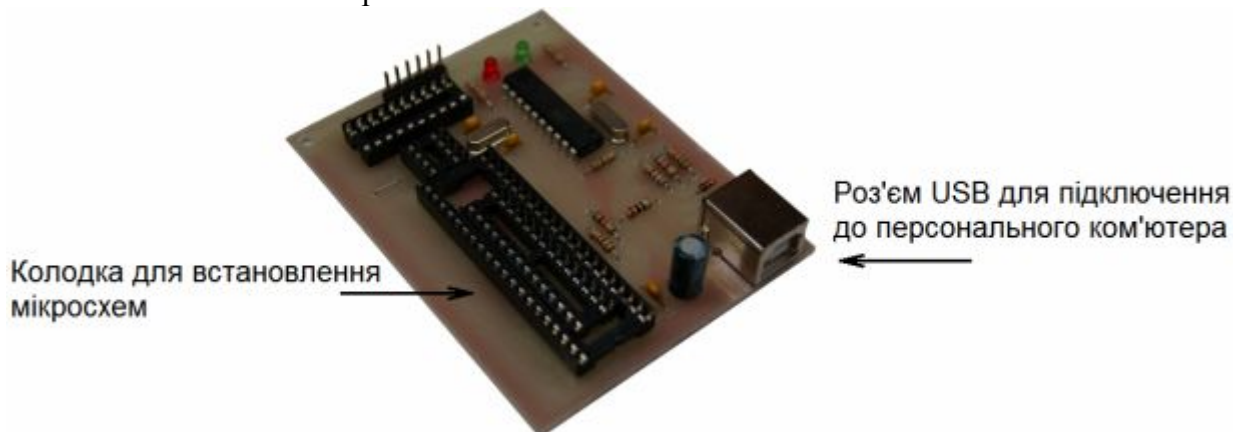


Рис. 5.4. Пристрій програмування для мікроконтролерів AVR

Програмування мікроконтролерів зазвичай здійснюється на Асемблері або Сі, хоча існують компілятори для інших мов, використовуються також інтерпретатори Бейсіка і Фортю.

Для налагодження програм використовуються програмні симулятори у вигляді спеціальних програм для персональних комп'ютерів, що імітують роботу мікроконтролера у вигляді програмної моделі. Для налагодження апаратних компонент використовують схемні емулятори, що являють собою електронні пристрої, які імітують мікроконтролер. Ці емулятори можна підключити замість однокристалого мікроконтролера до вбудованого пристрою та перевірити роботу програми.

У разі необхідності у вбудованих системах реалізації складних алгоритмів обробки даних, наприклад, обробки зображення у реальному часі, використовують так звані **Системи на кристалі** або **Системи на чипі** (від англ **System-on-a-chip**, або ще **SoC** чи **SOC**), що являють собою електронної схеми, які вміщують функціональні складові цілого пристрою (наприклад, компютера) на одній мікросхемі. Якщо не вдається розмістити всі необхідні схеми на одному напівпровідниковому кристалі, то використовується схема із декількох кристалів, розміщених в одному корпусі (System in Package — SiP).

Прикладом такої системи на одному чипі є процесор BCM2835, що використовується в одноплатних комп'ютерах Raspberry Pi (рис. 5.5)

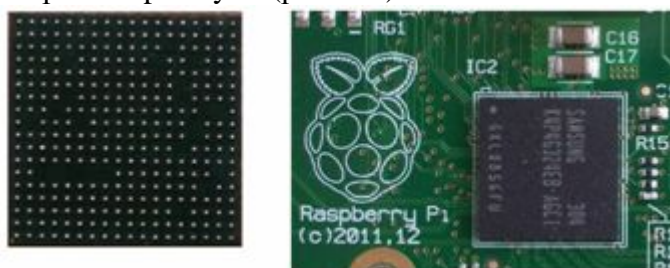


Рис. 5.5. Процесор BCM2835

Тактова частота процесора BCM2835 700 МГц. Він має вбудоване графічне ядро з підтримкою OpenGL (Open Graphics Library - відкрита графічна бібліотека для створення програмного забезпечення, що використовують двовірну і тривимірну комп'ютерну графіку), також є підтримка апаратного прискорення і FullHD-відео.

Процесор BCM2835 містить також ядро цифрового сигнального процесора - спеціалізований мікропроцесор, який призначений для цифрової обробки сигналів. Модулі оперативної пам'яті 256Мб / 512МБ розміщені на процесорі за технологією Package-on-Package (PoP) - установка один на одному.

Розміщення дискретної логіки і пам'яті штабелями, дозволяють зробити пристрій більш компактним.

5.3. Приклади пристроїв керування на основі однокристальних мікроконтролерів

Один з найбільш типових прикладів використання однокристальних мікроконтролерів є контролер інтерфейсу клавіатури, що використовується в персональних комп'ютерах.

Принцип роботи контролера клавіатури.

Клавіші клавіатури знаходяться на перехресті між горизонтальними та вертикальними лініями.

На горизонтальні лінії поступає сигнал логічної одиниці. У звичайному стані (якщо відсутнє натискання) лінії не замикаються, і результатом опитування на усіх горизонтальних лініях є сигнал логічної одиниці (на лініях є напруга).

На вертикальні лінії по черзі подається логічний нуль.

Якщо натиснути на клавішу, то вона замикає вертикальну і горизонтальну лінії, внаслідок чого на горизонтальній лінії встановлюється отриманий від вертикальної лінії сигнал логічного нуля. Відповідно до того, на яку саме вертикальну лінію поданий нуль і з якою горизонтальною лінією замість одиниці надійшов нульовий сигнал, визначається, яка клавіша була натиснута (рис. 5.6).

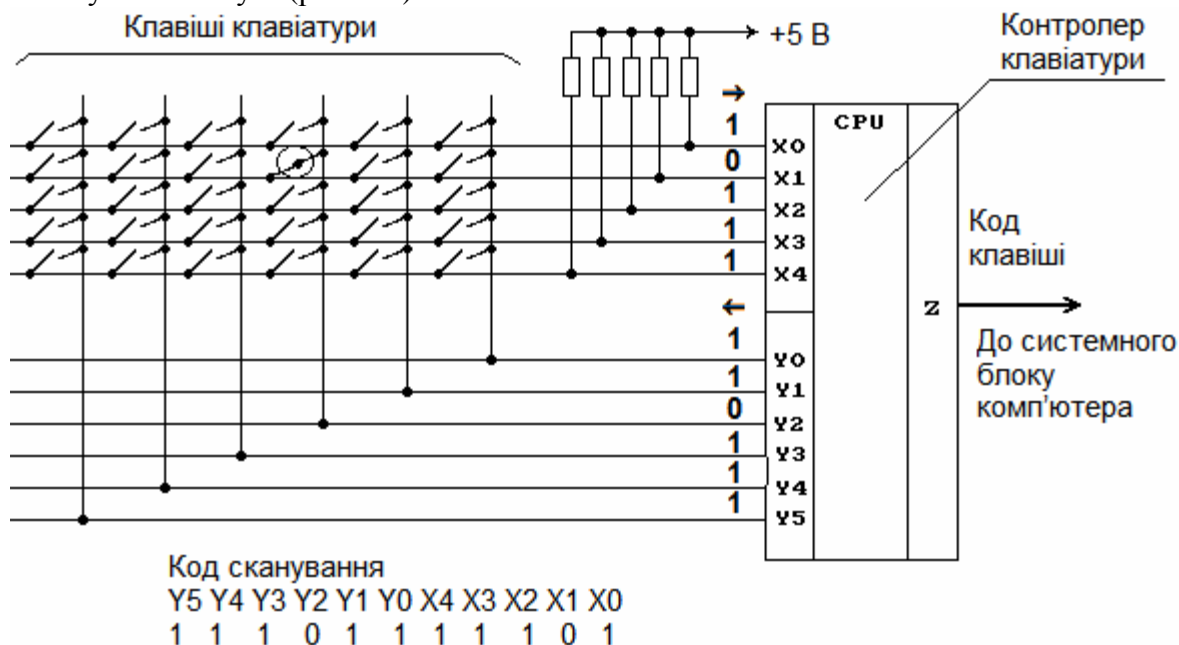


Рис. 5.6. Спрощений принцип дії клавіатури

Вказану функцію на старих комп'ютерах виконував центральний процесор, використовуючи програмний режим введення даних. Опитування клавіатури здійснювалось постійно, що навантажувало процесор.

У сучасних комп'ютерів цю функцію виконує контролер клавіатури, наприклад, контролер i8042. Він визначає код сканування отриманого сигналу, перетворює його на код відповідної клавіші (або комбінації клавіш), відправляє на центральний процесор запит на переривання, передаючи при цьому код клавіші, яка була приведена в дію. Таким чином центральний процесор приймає участь у опитуванні клавіатури тільки при натисканні клавіші.

Малі габарити та широкі можливості дозволяють використовувати однокристальні мікроконтролери в різних засобах автоматизації, наприклад, у датчиках для обробки інформації, у регульованих електроприводах

Прикладом використання вбудованих пристроїв керування різного рівня складності є регульовані електроприводи, наприклад, сервоприводи з відносно простими пристроями керування, та частотні перетворювачі з досить складними пристроями керування.

Сервоприводи використовуються у разі необхідності здійснити поворот на певний кут. Під сервоприводом в даному випадку розуміють механізм з електромотором, який можна повернути в заданий кут і утримувати в цьому положенні (рис. 5.7, а).

Для управління сервоприводами використовується широтно-імпульсна модуляція. При цьому кут повороту визначається тривалістю імпульсу (рис. 5.7, б).

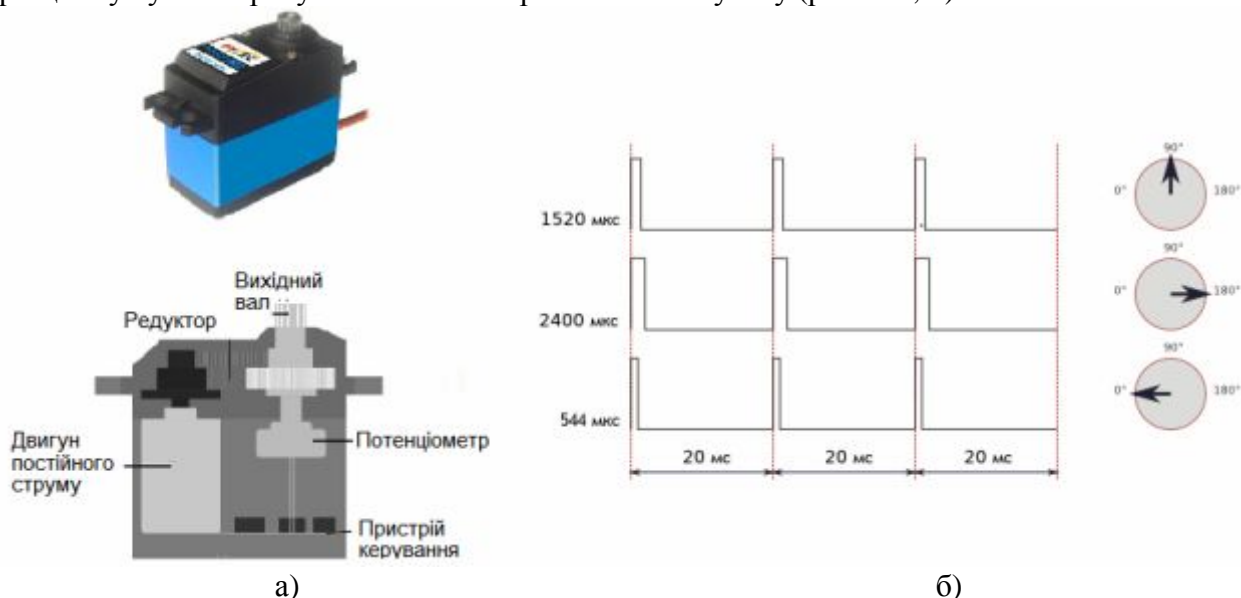


Рис. 5.7. Сервопривод (а) та залежність положення вала від тривалості імпульсів (б)

Пристрій керування порівнює тривалість імпульсу керування з значенням потенціометра, що визначає кут повороту вала, та утримує задане значення кута повороту вала.

В частотних перетворювачах (рис.5.8) пристрій керування задає дуже багато параметрів керування двигуном, таких як швидкість обертання, час розгону та гальмування, типи регулювання швидкості обертання та обертаючий момент. Крім того встановлюється багато параметрів безпеки, наприклад, максимальна швидкість обертання, максимальних струм, тощо. Загальна кількість параметрів різних частотних перетворювачів може досягати декількох сотень і тисяч. При цьому вбудовані пристрої керування є в операторській панелі, в самому пристрою керування та в силовому модулі.



Рис. 5.8. Частотний перетворювач фірми Сіменс SINAMICS G120

Вбудовані пристрої керування широко використовують у робототехнічних пристроях, у різній побутовій техніці, в автомобільній електроніці, в медичному обладнанні, у різних гаджетах (наприклад, мобільні телефони, цифрові фотоапарати та відеокамери) тощо.

Контрольні запитання

1. Чому однокристальні мікроконтролери є основою для вбудованих пристроїв керування?
2. Чим відрізняються однокристальні мікроконтролери?
3. Яка архітектура найчастіше використовується в пристроях керування та чому?
4. Які мови програмування найчастіше використовують для програмування однокристальних мікроконтролерів?
5. Які функціональні компоненти найчастіше мають однокристальні мікроконтролери?
6. Чим відрізняються однокристальні мікроконтролери i8051?
7. Чим відрізняються однокристальні мікроконтролери AVR?
8. Які засоби використовують для програмування однокристальних мікроконтролерів?
9. У чому полягає принцип дії клавіатури?
10. Як здійснюється керування сервоприводом?

Глава 6. Пристрої керування на основі мікроконтролерів

6.1. Основні типи одноплатних мікроконтролерів

Однокристалні мікроконтролери знайшли широке застосування для вирішення найрізноманітніших завдань управління, але, оскільки вони являють собою мікросхеми, то для проектування пристроїв на їх основі та подальшого використання потребується цілий ряд додаткових засобів, включаючи друковану плату, де встановлюється мікросхема, пристрій програмування тощо. Тому цілий ряд фірм налагодив випуск мікроконтролерів у вигляді однієї друкованої плати з можливістю підключення додаткових модулів і здатних вирішувати велике коло завдань, в тому числі обробки даних, керування різними приладами, управління рухом, тощо.

Діапазон таких контролерів простягається від простих систем логічного керування до мікрокомп'ютерів, здатних реалізовувати складні алгоритми, наприклад, обробку зображення у реальному часі.



Рис. 6.1. Одноплатні мікроконтролери

Однокристалні мікроконтролери i8051 використовують мікроконтролери фірми Phytex (рис. 6.2). Всі плати Phytex доступні в варіанті комплекту Rapid Development Kit, до складу якого входять додатковий модуль налагодження і плата-носії Development Board, що служить для програмування модуля і тестування програмного забезпечення.

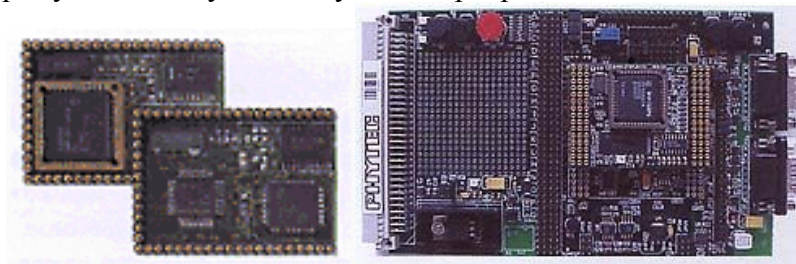


Рис.6.2. MicroMODUL-8051 та плата Development Board

Прикладом використання мікроконтролерів AVR може служити апаратно-програмний комплекс Arduino, який представляє собою набір засобів, пристосованих для побудови простих систем автоматики і робототехніки, орієнтований на непрофесійних користувачів. Програмна частина складається з безкоштовної програмної оболонки для написання програм, їх компіляції і програмування апаратури. Апаратна частина являє собою набір готових пристроїв, виконаних у вигляді друкованих плат. Повністю відкрита архітектура системи дозволяє вільно копіювати або доповнювати пристрої Arduino.

Arduino може використовуватися як для створення автономних об'єктів автоматики, так і підключатися до програмного забезпечення на комп'ютері через стандартні дротові та

бездротові інтерфейси. Також є велика кількість різних датчиків і виконавчих пристроїв, що дозволяють вирішувати широке коло завдань.

При цьому є як датчики руху, так і модулі регульованих приводів для електродвигунів постійного струму і крокових двигунів, а також сервоприводи, що дозволяють здійснити поворот на заданий кут.

Мікроконтролери Arduino відрізняються наявністю попередньо встановленого в них завантажувача (bootloader). За допомогою цього завантажувача користувач завантажує свою програму в мікроконтролер без використання окремих апаратних пристроїв програмування. Завантажувач з'єднується з комп'ютером через інтерфейс USB (якщо він є на платі) або за допомогою відповідного перехідного пристрою.

У лінійці пристроїв Arduino в основному застосовуються мікроконтролери Atmel AVR: ATmega328, ATmega168, ATmega2560, ATmega32U4, ATTiny85 з тактовою частотою 16 або 8 МГц.

Розглянемо деякі контролери Arduino (рис. 6.3).

Контролер Arduino Uno (рис. 6.3, а), побудований на ATmega328. Платформа має 14 цифрових входів / виходів (6 з яких можуть використовуватися як виходи ШІМ), 6 аналогових входів, кварцовий генератор 16 МГц, роз'єм USB, і кнопку скидання. Розмір плати $6,9 \times 5,3$ см.

Контролер Arduino Nano (рис. 6.3, б) побудований на ATmega328. Платформа має 14 цифрових входів / виходів (6 з яких можуть використовуватися як виходи ШІМ), 8 аналогових входів, кварцовий генератор 16 МГц, роз'єм Mini USB. Відмінною особливістю є малі розміри ($1,85 \times 4,2$ см), що дозволяють вбудовувати контролер в портативні пристрої.

Контролер Arduino Mega 2560 (рис. 6.3, в) виконаний на основі мікроконтролера ATmega2560.

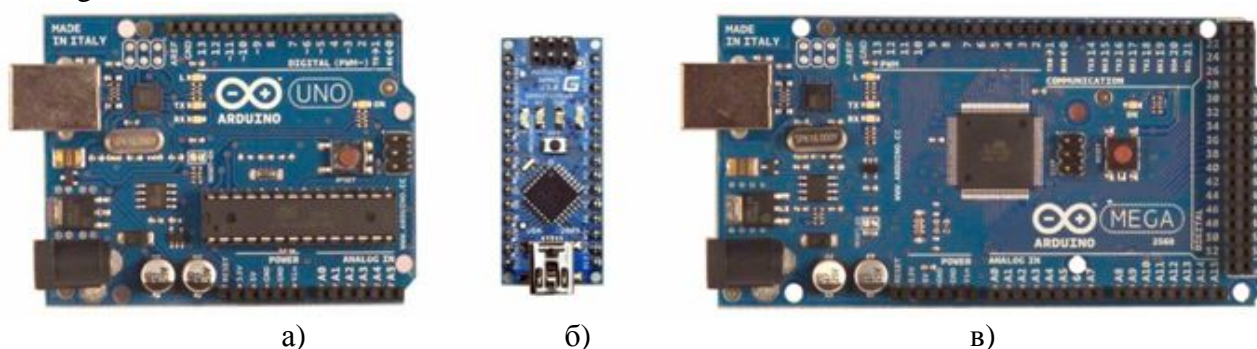


Рис. 6.3. Контролери Arduino Uno (а), Arduino Nano (б) та Arduino Mega 2560 (в)

У його склад входить: 54 цифрових входів / виходів (з яких 15 можуть використовуватися як виходи ШІМ), 16 аналогових входів, 4 апаратних приймально-передавачів для реалізації послідовних інтерфейсів UART, кварцовий генератор 16 МГц, роз'єм USB, роз'єм живлення, роз'єм ICSP для внутрішньосхемного програмування і кнопка скидання. Розмір плати $10,8 \times 5,3$ см.

Для реалізації складних алгоритмів обробки даних використовують одноплатний комп'ютер Raspberry Pi (рис. 6.4).



Рис. 6.4. Одноплатний комп'ютер Raspberry Pi

Raspberry Pi побудований на системі-на-чипі (SoC) BCM2835, яка включає процесор із тактовою частотою 700 МГц, графічний процесор VideoCore IV, і 512 чи 256Мб оперативної пам'яті. Замість твердого диску використовується SD карта.

Така апаратна конфігурація дозволяє відтворювати відео формату H.264 з роздільною здатністю 1080р.

Найбільш поширена мова програмування для Raspberry Pi — Python.

6.2. Мови програмування одноплатних мікроконтролерів

Для програмування мікроконтролерів найчастіше використовуються мови програмування Асемблер або С. У деяких випадках можливе використання інших мов, наприклад, Бейсіка і Форту.

Розглянемо деякі можливості програмування на Асемблері мікроконтролерів на основі MCS-51.

MCS-51 являє собою закінчений обчислювальний пристрій, що включає центральний процесор (арифметично-логічний пристрій), внутрішню пам'ять, набір регістрів спеціальних функцій, пристрій керування і порти вводу та виводу, що об'єднуються 8-розрядною шиною.

Арифметично-логічний пристрій виконує функції над двома типами даних: байтами і бітами. Це дозволяє спростити реалізацію таких функцій керування, як опитування двійковими датчиків і видачу керуючих сигналів релейного типу.

Дані можуть переміщатися усередині ОМК і з ними можуть виконуватися арифметичні і логічні команди, такі як складання, віднімання, множення, ділення, логічні операції І, АБО.

Для видачі двійкових керуючих сигналів на порти можуть використовуватися команди установки заданого біта в "1"

SETB bit

і скидання заданого біта в "0"

CLR bit,

де bit - пряма адреса біта, для якого виконується дана команда.

Для окремих ліній портів вводу-виводу прийняте таке позначення:

Pn.i

n - номер порту, n = 0...3;

i - номер лінії порту, i = 0...7.

У такий засіб видача сигналу "1" або "0" на задану лінію здійснюється за допомогою однієї команди, наприклад, команда

SETB P2.3

встановлює в "1" вихід P2.3.

Опитування двійкового датчика на замикання і розмикання може здійснитися за допомогою команд умовного переходу з перевіркою заданого біта на задане значення, як-то:

JB bit,L1 - перехід на мітку L1, якщо значення bit дорівнює "1";

JNB bit,L2 - перехід на мітку L2, якщо значення bit дорівнює "0".

За допомогою цих команд досить просто зробити програму керування, яка складається з послідовності дій наприклад:

- 1) включити двигун переміщення;
- 2) опитати кінцевий датчик, якщо датчик не спрацював, перейти до наступної дії, якщо спрацював, перейти до слідує команди;
- 3) виключити двигун переміщення.

Програма, яка виконує цю послідовність дій має вигляд:

```
SETB P1.1 ;включити двигун переміщення
JNB P2.1, L1 ;опитати кінцевий датчик, якщо датчик
;не спрацював, перейти на мітку L1,
;якщо спрацював, перейти до
;наступної ;команди
CLR P1.1 ;виключити двигун переміщення
```

L1: ...

Для мікроконтролерів широкого призначення існують програмні середовища, що здійснюють програмування, налагодження програми, проектування апаратних компонент тощо.

Так, наприклад, для програмування контролерів Arduino використовується середовище розробки Arduino, яка складається з безкоштовної програмної оболонки (IDE) для написання програм, їх компіляції та програмування апаратури.

Мова програмування Arduino є стандартним C++ з особливостями, які полегшують невідготуванню користувачам написання працюючої програми.

Програми для Arduino називають скетчі (від англ. Sketch), які складаються в текстовому редакторі. Після компіляції програма зберігається в файлах з розширенням ipn. Перед компіляцією ці файли обробляються препроцесором Arduino, у ході чого здійснюється виявлення помилок у разі наявності.

Також існує можливість створювати і підключати до проекту стандартні файли C++.

Обов'язкову в C++ функцію main () препроцесор Arduino створює сам, вставляючи туди необхідні дії.

Програміст повинен написати дві обов'язкові для Arduino функції setup () і loop (). Перша викликається одноразово при старті, друга виконується в нескінченному циклі.

В тексті своєї програми (скетчу) програміст може використовувати стандартні бібліотеки.

Менеджер проекту Arduino IDE має нестандартний механізм додавання бібліотек. Бібліотеки у вигляді вихідних текстів на стандартному C++ додаються в спеціальну папку в робочому каталозі IDE. При цьому назва бібліотеки буде додано до списку бібліотек в меню IDE. Програміст зазначає потрібні бібліотеки і вони вносяться до списку компіляції.

Arduino IDE не пропонує ніяких налаштувань компілятора і мінімізує інші настройки, що спрощує початок роботи для новачків і зменшує ризик виникнення помилок.

Найпростіша Arduino-програма складається з двох функцій:

- setup (): функція викликається одноразово при старті мікроконтролера.
- loop (): функція викликається після setup () в нескінченному циклі весь час роботи мікроконтролера.

Нижче наведено повний текст однієї з найпростіших програм (скетчу) миготіння світлодіодом, підключеного до виходу 13 Arduino, з періодом 1 секунди, де для управління цифровим виходом використовується функція digitalWrite (). Ця функція має два аргументи: номер виходу, на який виводиться дискретний сигнал, і значення сигналів HIGH (високий рівень) або LOW (низький рівень).

```
void setup () {
```

```
pinMode (13, OUTPUT); // Визначення вивода 13 як вихід
}
void loop () {
digitalWrite (13, HIGH); // Включення вивода 13, параметр HIGH
// - високого признак високого логічного рівня
delay (1000); // Цикл затримки на 1000 мс – 1 секунда
digitalWrite (13, LOW); // Виключення вивода 13, параметр LOW
delay (1000); // Цикл затримки на 1000 мс – 1 секунда }
```

Всі використовувані в прикладі функції є бібліотечними. У комплекті Arduino IDE є велика кількість прикладів програм.

Мова програмування Python для Raspberry Pi це інтерпретована об'єктно-орієнтована мова програмування високого рівня.

Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання існуючих компонентів.

Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій так і у вихідній формі на всіх основних платформах.

В мові програмування Python підтримується декілька парадигм програмування (систем ідей і понять, які визначають стиль написання комп'ютерних програм), а саме: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована

Інтерпретатор мови Python і багата стандартна бібліотека (як вихідні тексти, так і бінарні дистрибутиви для всіх основних операційних систем) можуть бути отримані з сайту Python www.python.org, і можуть вільно розповсюджуватися. Цей самий сайт має дистрибутиви та посилання на численні модулі, програми, утиліти та додаткову документацію.

Інтерпретатор мови Python може бути розширений функціями та типами даних, розробленими на C чи C++ (або на іншій мові, яку можна викликати із C). Python також зручна як мова розширення для прикладних програм, що потребують подальшого налагодження.

В режимі налагодження, інтерпретатор Python має інтерактивний режим роботи, при якому введені з клавіатури оператори відразу ж виконуються, а результат виводиться на екран.

Так виглядає спілкування з Python в інтерактивному режимі:

```
>>> 2 ** 100 # піднесення 2 до 100-го степеня
1267650600228229401496703205376L
>>> from math import * # імпорт математичних функцій
>>> sin (pi * 0.5) # обчислення синуса від половини пі
1.0
>>> help (sorted) # допомогу по функції sorted
Help on built-in function sorted in module __builtin__:
sorted (...)
sorted (iterable, cmp=None, key=None, reverse=False) -> new sorted
list
```


6.3. Функції керування рухом та обробки даних

Розглянемо, як здійснюється програмування на Асемблері MCS-51 на прикладі позиційного переміщення візка стележного крана-штабелера за допомогою щілинного датчика при переміщенні вздовж стележу (рис. 6.5).

Позиціонування здійснюється підрахуванням числа імпульсів при проходженні датчика вздовж осі з пластинами.

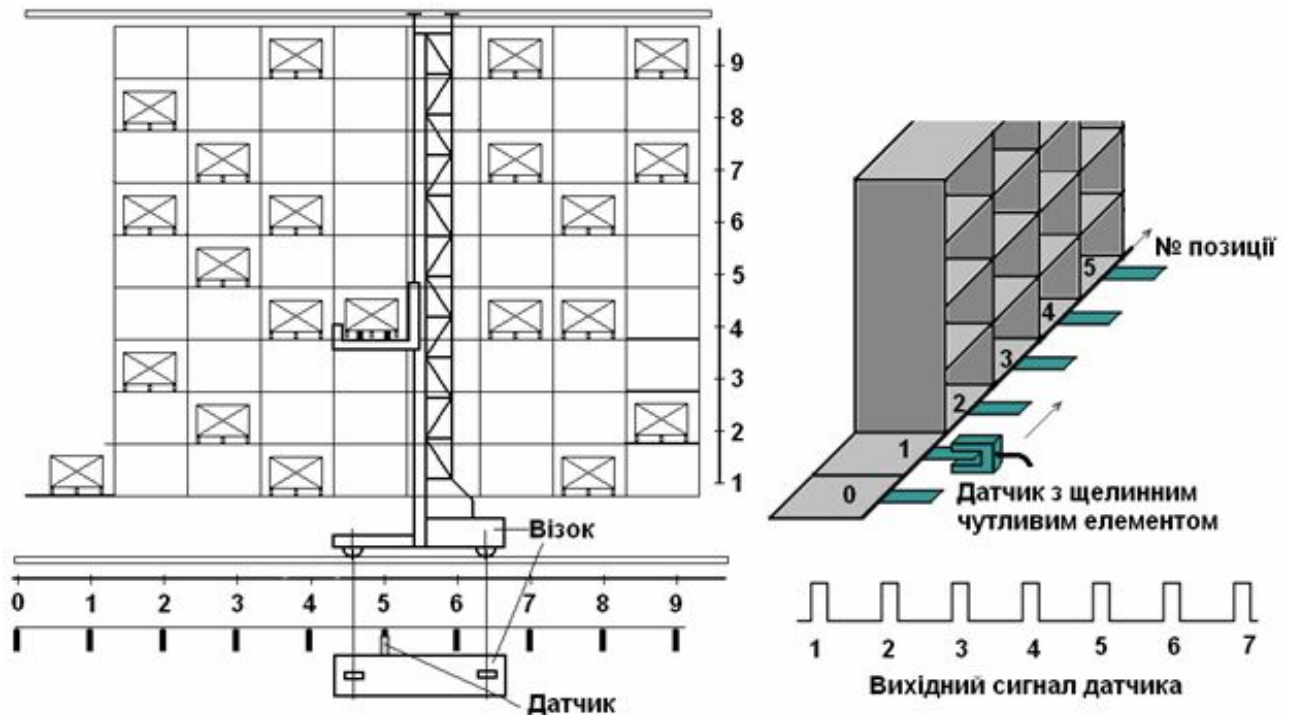


Рис.6.5. Позиційне керування за допомогою датчика з щілинним чутливим елементом

Для підрахування числа подій використовують програмний лічильник. При цьому у визначений регістр записується задане число циклів, і його вміст зменшується на одиницю в кожному наступному циклі доти, поки воно не стане рівним нулю. Цю операцію можна здійснити за допомогою команди

DJNZ Ri,L1 - зменшити на "1" значення регістра Ri, та перейти на мітку L1, якщо його значення не дорівнює "0";

При проходженні металеві пластини через щілину формується імпульс, тому треба використовувати програму опитування імпульсних сигналів, яка відрізняється тим, що необхідно виявити не тільки факт появи, але й закінчення імпульсу. У прикладі приведена програма опитування позитивного імпульсу ("0"->"1"->"0"), що фіксує момент переходу з "1" у "0" (задній фронт):

Програма позиційного керування для переходу на позицію 6 має вигляд:

```

MOV R4, # 6;записати номер позиції у регістр R4 (число 6)
L1:   SETB P1.1      ;включити двигун переміщення
L2:   JNB P2. 1,L2  ;чекання початку імпульсу на вході
      ;P2.1=1
L3:   JB P2. 1,L3   ;чекання кінця імпульсу на вході P1.3=0
      DJNZ R4,L1    ;зменшити на "1" значення регістра R4 та
      ;перейти на мітку L2, якщо його
      ;значення не дорівнює "0",
      CLR P1.1     ;виключити двигун переміщення
  
```

Розглянемо приклад простої програми маршрутослідкування для транспортного робота.

На рис. 6.6 наведений візок транспортного робота, який здійснює

маршрутослідкування за допомогою двох оптичних датчиків.

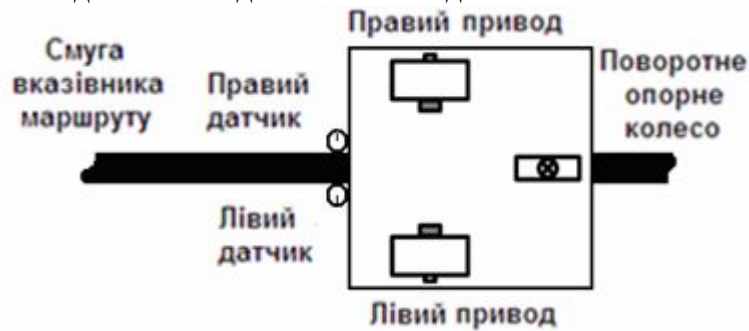


Рис. 6.6. Маршрутослідкування за допомогою двох оптичних датчиків

Візок має два привода – правий та лівий та два оптичних датчика – справа та зліва від смуги вказівника маршруту. У вихідному положенні оптичні датчики знаходяться справа та зліва від смуги вказівника маршруту (датчики не спрацьовують) і включені обидві двигуна (візок переміщується прямо). Якщо на смугу попадає лівий датчик, то він спрацьовує та здійснюється поворот вправо (зупиняється лівий двигун), а якщо на смугу попадає правий датчик, то здійснюється поворот вліво (зупиняється правий двигун).

Зупинка візка здійснюється за допомогою кнопки "СТОП".

Датчики та приводи підключені до ОМК таким чином:

P2. 1 – правий датчик

P2. 2 – лівий датчик

P1. 1 – правий двигун

P1. 2 – лівий двигун

У цьому разі програма маршрутослідкування має вигляд:

```
L1:      JNB  P2. 1, L2      ;якщо не спрацював правий датчик
          ;P2.1=0, перехід на L2
          SETB P1.2        ;якщо спрацював правий датчик
          ;P2.1=1, здійснити поворот вправо
          ;а саме включити лівий двигун
          CLR  P1.1        ;та виключити правий двигун
L2:      JNB  P2. 2, L3      ;якщо не спрацював лівий датчик
          ;P2.2=0, перехід на L3
          SETB P1.1        ;якщо спрацював лівий датчик
          ;P2.2=1, здійснити поворот вліво
          ;а саме включити правий двигун
          CLR  P1.2        ;та виключити лівий двигун
L3:      SETB P1.1          ; якщо не спрацював жодний датчик
          ;здійснити рух прямо, а саме
          ;включити правий двигун
          SETB P1.2        ;та включити лівий двигун
```

Крім Асемблера для програмування мікропроцесорних систем використовується мова програмування вищого рівня C.

Розглянемо приклади програмування для контролерів Arduino, що використовують мову C.

Контролери Arduino часто використовують для керування різними робототехнічними пристроями, тому мають вмонтовані функції керування рухом та обробки даних.

Так для керування швидкості обертання двигунів постійного струму використовується управління виходом ШІМ за допомогою функції analogWrite(). Ця функція має два аргументи: номер виходу, на який виводиться сигнал ШІМ, і число в діапазоні від 0 до 255, яке задає пропорційну тривалість імпульсу ШІМ.

Наприклад, функція

```
analogWrite(7,127);
```

встановлює швидкість обертання на рівні 0,5 від напруги живлення двигуна постійного струму, що підключений до виходу 7..

Контролери Arduino мають багато додаткових модулів і пристрів, призначених для здійснення переміщення, наприклад, для сервоприводів та крокових двигунів.

Для спрощення програмування для таких пристроїв створені бібліотеки з відповідними функціями.

Так для керування сервоприводом використовується бібліотека Servo, яка має наступні функції.

Функція підключення сервопривода до вказаного виходу контролера:

```
servo.attach(pin)
```

де *servo* – змінна типа Servo,

pin – номер виходу, який здійснює керування приводом.

Функція повернення приводу на вказаний кут:

```
servo.write(angle)
```

де *servo* – змінна типа Servo,

angle – кут повороту від 0 до 180, відповідає куту повороту в градусах.

Для керування кроковим двигуном використовується бібліотека Stepper.

Ця бібліотека має наступні функції.

Функція встановлення параметрів крокового двигуна:

```
Stepper myStepper = Stepper(steps, pin1, pin2, pin3, pin4);
```

де *myStepper* – ім'я крокового двигуна,

steps - кількість кроків на одне обертання,

pin1, pin2, pin3, pin4 - виходи контролера, що використовуються для керування двигуном (у залежності від типа двигуна використовують два або чотири вихода).

Функція встановлення швидкості обертання двигуна myStepper:

```
myStepper.setSpeed(speed);
```

де *speed* - швидкість обертання у кількості обертів за хвилину.

Функція встановлення кількості кроків, на яке буде обертатися двигун *myStepper* (позначка – означає, що двигун буде обертатися в іншу сторону):

```
myStepper.step(steps);
```

де *steps* - кількість кроків.

Далі наведений приклад програми, яка здійснює одне обертання в одну сторону та одне в іншу. Між циклами обертання затримка 0,5 с.

```
#include <Stepper.h> //підключення бібліотеки Stepper
```

```
const int stepsPerRevolution = 200; // кількість кроків на одне обертання
```

```
// підключення до виходів 8 - 11:
```

```
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);
```

```
void setup() {
```

```
    // встановлення швидкості 60 об/хв.:
```

```
    myStepper.setSpeed(60);
```

```
}
```

```
void loop() {
```

```
    myStepper.step(stepsPerRevolution); // обертання в одну сторону
```

```
    delay(500); // затримка 0,5 с
```

```

myStepper.step(-stepsPerRevolution); // обертання в іншу сторону
delay(500); // затримка 0,5 с
}

```

Контролери Arduino також мають функції для обробки даних. Розглянемо деякі з них. Контролери Arduino мають аналогові входи. Для зчитування у змінну значення сигналу з аналогового входу `pin` використовується функція `analogRead()`, яка видають значення від 0 до 1023.

Функція має такий синтаксис:

```
value = analogRead(pin);
```

де `value` – змінна, куди записується результат опитування .

У програмах керування часто використовується зміна масштабу.

Таку функцію треба використовувати, наприклад, для ручного керування сервопривода, на який подаються значення від 0 до 180, за допомогою потенціометра, підключеного до аналогового входу, що видає значення від 0 до 1023.

Для зміни масштабу можна використати функцію пропорційного перетворення значень від одного діапазону до другого `map ()` та функцію обмеження діапазону `constrain ()`.

Функція зміни масштабу `map ()` має такий синтаксис:

```
valNew = map(valOld, fromLow, fromHigh, toLow, toHigh);
```

Змінна `valOld` з нижньою `fromLow` і верхньою `fromHigh` межами діапазону перетворюється у змінну `valNew` з нижньою `toLow` і верхньою `toHigh` межами діапазону.

Функція обмеження значень змінної `constrain ()` має такий синтаксис:

```
valNew = constrain (valOld, min, max);
```

`min` та `max` відповідно мінімальне та максимальне значення змінної `valNew`.

Далі наведений фрагмент програми ручного керування сервоприводом `myservo` змінною `val`, в яку записується значення з потенціометра POT:

```

// керування сервоприводом
val = analogRead(POT); //зчитування сигналу з аналогового входу
val = map(val, 0, 1023, 0, 180); //перетворення масштабу
val = constrain(val, 0, 180); //обмеження діапазону
myservo.write(val); //запис сигналу керування до сервоприводу

```

Для вимірювання тривалості імпульсів, наприклад, у ультразвукових датчиків вимірювання відстані, використовується функція `pulseIn()`.

Ця функція зчитує імпульс (HIGH чи LOW) на визначеному вході (`pin`). Наприклад, якщо значення є високим (HIGH), `pulseIn()` чекає, поки стан `pin` зміниться на HIGH, починає підрахунок часу у мікросекундах, потім чекає, поки стан входу `pin` не зміниться на LOW, і зупиняє підрахунок.

```
pulseIn(pin, value)
```

```
pulseIn(pin, value, timeout)
```

де `pin` - номер піна для зчитування імпульсу (`int`)

`value` - тип імпульсу для зчитування HIGH чи LOW (`int`)

(опціонально)

`timeout` - час очікування у мікросекундах, стандартне значення 1 секунда (`unsigned long`)

Повертає тривалість імпульсу (у мікросекундах) або 0, якщо не було повного імпульсу за час очікування (`unsigned long`)

Розглянемо програму вимірювання відстані за допомогою ультразвукового датчика HC-SR04 (рис. 6.7).



Рис. 6.7. Датчик HC-SR04

Датчик має 4 вивода:

VCC: "+" живлення;

TRIG (T): вивід вхідного сигналу;

ECHO (R): вивід вихідного сигналу;

GND: "-" живлення (земля).

Тривалість вихідного сигналу залежить від відстані об'єкту до датчика.

Принцип роботи датчика можна умовно поділити на 4 етапи.

1. Подаємо імпульс тривалістю 10 мкс, на вивід Trig.

2. Датчик перетворює вхідний імпульс у 8 імпульсів частотою 40 КГц та відправляє вперед через передавач T.

3. Дійшовши до перешкоди, послані імпульси відбиваються і приймаються приймачем R. Отримуємо вихідний сигнал на виводі Echo.

4. Безпосередньо в контролері переводимо отриманий сигнал в відстань за формулою: ширина імпульсу (мкс) / 58 = дистанція (см).

Подключение датчика к Arduino:

VCC на +5 вольт Arduino.

Trig на пін 9.

Echo на пін 8.

GND на любий пін GND.

У наведеному нижче скетчі дистанція відсилається в порт комп'ютера, а при дистанції менше 30 сантиметрів включається світлодіод, що підключений до піну 13.

```

sketch_echo
#define Trig 9
#define Echo 8
#define ledPin 13
void setup()
{
pinMode(Trig, OUTPUT); // вихід
pinMode(Echo, INPUT); // вхід
pinMode(ledPin, OUTPUT);
Serial.begin(9600);
/* задаємо швидкість послідовного каналу */
}
unsigned int impulseTime=0;
unsigned int distance_sm=0;
void loop()
{
digitalWrite(Trig, HIGH);
/* Подаємо імпульс на вхід trig */
delayMicroseconds(10); // що дорівнює 10 мікросекунд

```

```

digitalWrite(Trig, LOW); // відключаємо
pulseTime=pulseIn(Echo, HIGH); // визначаємо тривалість імпульсу
distance_sm=pulseTime/58; // Перераховуємо в сантиметри
Serial.println(distance_sm); // Виводимо на послідовний канал
if (distance_sm<30) // Якщо відстань менше ніж 30 сантиметрів
{
digitalWrite(ledPin, HIGH); // включаємо світлодіод
}
else
{
digitalWrite(ledPin, LOW); // у протилежному випадку виключаємо
}
delay(100);
/* чекаємо 0,1 секунди, Наступний імпульс може бути відправлений тільки після зникнення
відбитого попереднього імпульсу.
Рекомендований період між імпульсами повинен бути не менше 50 мс.*
*/
}

```

Контрольні запитання

1. Чим відрізняються контролери Arduino Uno, Arduino Nano та Arduino Mega?
2. Які мови програмування найчастіше використовують для програмування мікроконтролерів?
3. Які команди використовують для видачі двійкових керуючих сигналів у MCS-51?
4. Які команди використовують для опитування двійкового датчика у MCS-51?
5. З яких функцій складається найпростіша Arduino-програма?
6. Яка функція використовується в Arduino для управління цифровим виходом?
7. Яка функція використовується в Arduino для управління виходом ШІМ?
8. Яка бібліотека використовується для керування сервоприводом?
9. Яка бібліотека використовується для керування кроковим двигуном?
10. Які функції мають Arduino для обробки даних?

Завдання до практичних занять

Задача 1

Скласти програму для керування швидкості двигуна постійного струму за допомогою потенціометра.

Задача 2

Скласти програму для керування послідовністю переміщень сервопривода з встановленням таких кутів повороту:

0°, 40°, 60°, 100°, 120°, 100°, 60°, 40°, 0°,

з затримкою між окремими переміщеннями 2 с.

Задача 3

Скласти програму вимірювання відстані за допомогою ультразвукового датчика HC-SR04 з таким підключенням: Trig - пін 2, Echo - пін 3.

Ультразвуковий датчик керує переміщенням візка.

Візок їде прямо з максимальною швидкістю, якщо відстань до перешкоди більше 40 см.

Візок їде прямо з швидкістю 0,5 від максимальної, якщо відстань до перешкоди більше 20 см.

Якщо відстань до перешкоди менше ніж 20 см, візок зупиняється. При зникненні перешкоди рух продовжується.

Двигун підключений до виходу 5 Arduino Nano.

Глава 7. Програмовані логічні контролери на основі мікропроцесорної техніки

7.1. Класифікація, структура і склад програмованих логічних контролерів.

Застосування систем програмного керування ГВС у даний час постійно збільшується в міру підвищення складності технологічних процесів і інших об'єктів керування. При цьому рівень складності вирішуваних завдань може змінюватися в дуже великому діапазоні - від найпростіших систем циклового керування до складних систем керування гнучкими виробничими модулями і цехами. Одним з основних вимог до систем програмного керування в цьому випадку є можливість адаптації під конкретні задачі і спроможність об'єднання локальних систем керування в єдину автоматизовану систему керування виробництвом. Для задоволення цих вимог широко використовуються комплексні системи керування виробництвом, що включають локальні системи керування на основі програмованих контролерів, промислові мережі, що об'єднують окремі локальні системи керування в єдину комплексну систему, промислові комп'ютери, на яких реалізується верхній рівень керування.

Такий підхід, заснований на модульному принципі, дозволяє забезпечити гнучкість системи, забезпечує можливість швидкої адаптації при зміні параметрів технологічних процесів. Наявність широкого набору програм для проектування, програмування і налагодження системи значно скорочує обсяг робіт при проектуванні і переустановці системи. Основою цього принципу є повна сумісність апаратних і програмних засобів, які використовуються для вирішення задач автоматизації різного рівня, та інтеграція усіх програмних засобів у єдине програмне середовище.

Основною задачею сучасних програмованих контролерів є керування технологічним процесом, тому вони мають у своєму складі умонтовані технологічні функції, а саме функції керування переміщенням, включаючи функції керування двигунами та позиціонування, а також функції регулювання для здійснення ПІД-регулювання, регулювання з використанням нечіткої логіки та регулювання на основі нейронних систем. Ці функції можуть бути реалізовані апаратними або програмними засобами за допомогою відповідних додаткових модулів чи функцій.

За рівнем задач, що вирішує ПЛК, поділяють:

- системи керування низького рівня, у яких кількість входів та виходів може дорівнювати декілька сотень,
- системи керування середнього рівня, у яких кількість входів та виходів може дорівнювати декілька тисяч,
- системи керування високого рівня, у яких кількість входів та виходів може дорівнювати десятки тисяч.

Системи керування низького рівня будуються як компактні контролери та мають вигляд окремого пристрою, який включає процесор та умонтовані входи та виходи. Вони здатні самостійно вирішувати прості задачі керування.

До таких систем керування можна віднести, наприклад, контролери фірми Сименс LOGO! (рис. 7.1), та SIMATIC S7-1200 (рис. 7.2).



Рис. 7.1. Контролер LOGO! з додатковими модулями та операторською панеллю

Системи автоматизації середнього та високого рівня будуються по модульному принципу (компактні контролери в разі потреби можуть розширюватися шляхом

підключення додаткових модулів) і складаються з центрального (процесорного) модуля, блока живлення, і різноманітних модулів вхідних та вихідних сигналів (сигнальні модулі). Зв'язок центрального і периферійних модулів здійснюється за допомогою системної шини (системний з'єднувач), що забезпечує передачу адрес, даних і керуючих сигналів.

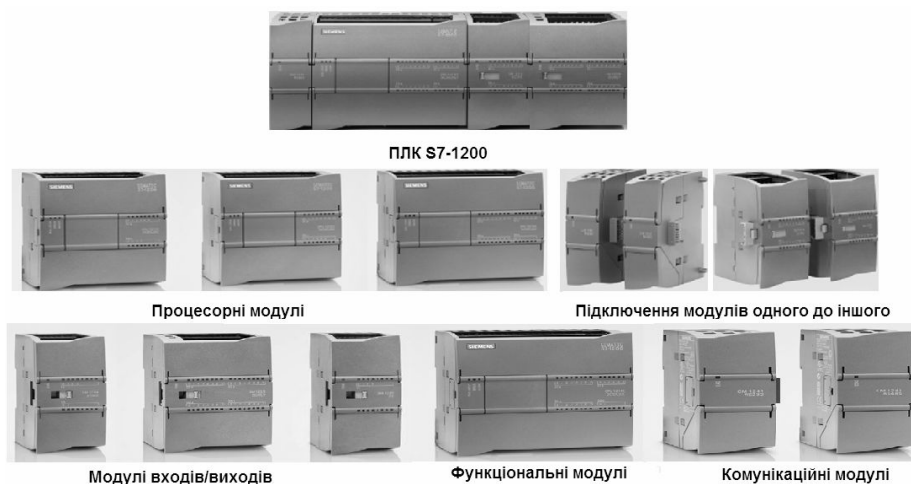


Рис. 7.2. ПЛК SIMATIC S7-1200

ПЛК складається з модулів, які встановлюються на носії модулів. У разі потреби може використовуватися побудова системи автоматизації, що мають декілька носіїв модулів (рис. 7.3). При цьому використовуються спеціальні інтерфейсні модулі підключення ІМ. Загальна кількість носіїв модулів і самих модулів визначається типом контролера.

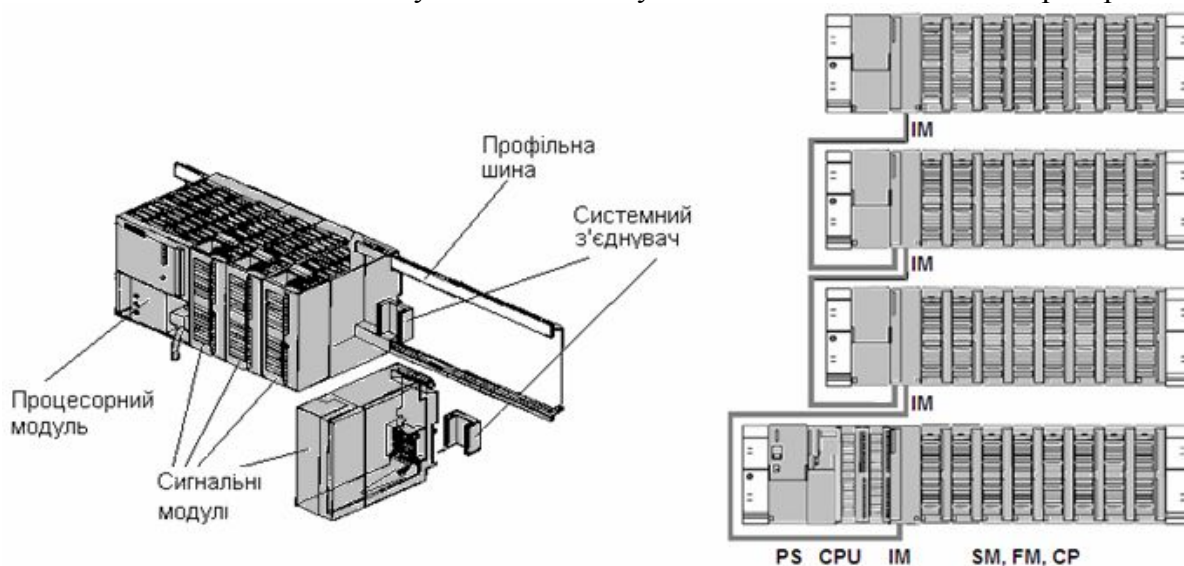


Рис.7.3. Програмований логічний контролер S7-300, побудований за модульним принципом

ПЛК можуть мати такі типи модулів (рис. 7.4):

- модулі (блоки) живлення (PS), які здійснюють функції живлення усієї системи перетворюючи напругу з мережі у потрібні постійні напруги;
- процесорні модулі (CPU);
- сигнальні модулі (SM), що забезпечують підключення цифрових і аналогових вхідних та вихідних сигналів;
- функціональні (інтелектуальні) модулі (FM), що виконують визначені функції незалежно від процесорного модуля, наприклад модулі позиціонування і регулювання;

- комунікаційні процесори (СР), що здійснюють зв'язок системи керування з іншими пристроями, у тому числі з операторськими станціями і промисловими мережами;
- інтерфейсні модулі (ІМ), що забезпечують зв'язок між окремими рядами у випадку багаторядної побудови системи автоматизації;
- модулі імітації та діагностики.

Кожен модуль має декілька варіантів, що дозволяє вибрати оптимальну за можливостями та вартістю конфігурацію системи керування.

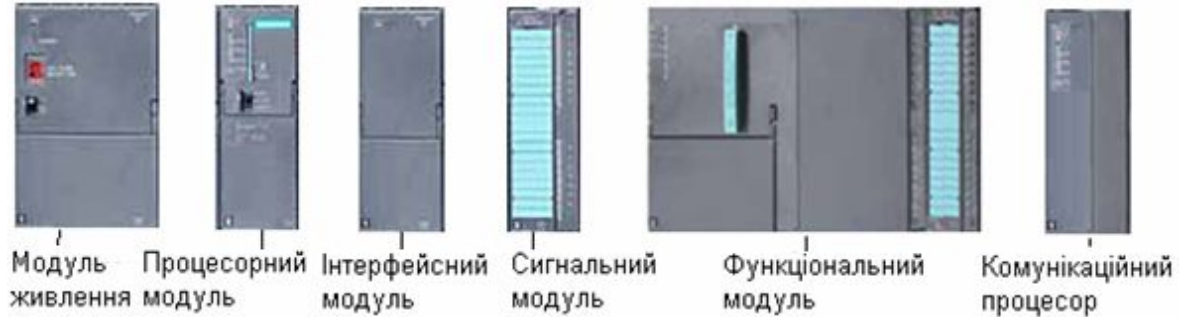


Рис. 7.4. Модулі програмованих логічних контролерів

7.2. Модулі центральних процесорів

Процесорні модулі представляють собою завершений обчислювальний пристрій з процесором, пам'яттю, та пристроями введення-виведення. Сімейство пристроїв керування як правило має набір процесорних модулів з різними можливостями програмного забезпечення, швидкодії, розміру пам'яті та можливостями підключення до мереж зв'язку. Це дозволяє вибрати оптимальне рішення для реалізації заданого алгоритму керування.

Структура процесорних модулів відрізняється від структури персональних комп'ютерів. Вони потребують значно менші обсяги пам'яті, тому програма зберігається не у зовнішніх носіях, а у постійній пам'яті (як правило це пам'ять з електричним перезаписом – FLASH пам'ять). Та частина програми, яка виконується та дані зберігаються у оперативній пам'яті. Ця пам'ять теж значно менша, ніж у персональних комп'ютерів. Пристрої введення-виведення призначені для підключення додаткових модулів за допомогою системної магістралі, та зовнішніх пристроїв за допомогою локальних мереж, наприклад, програмуючого пристрою, або іншого контролера.

Компактні процесорні модулі мають вмонтовані входи та виходи для підключення дискретних та аналогових сигналів, а також можуть виконувати додаткові технологічні функції, наприклад, обслуговування сигналів переривання, функції швидкодіючих лічильників, вимірювання частоти, функції позиціонування та регулювання тощо.

7.3. Сигнальні, функціональні та комунікаційні модулі

Сигнальні модулі поділяються на дискретні та аналогові модулі вхідних та вихідних сигналів. Дискретні модулі вхідних сигналів використовують сигнали постійного струму 24 В та змінного струму 120/220 В. Дискретні модулі вихідних сигналів дозволяють використовувати такі ж самі сигнали, але з різним струмом навантаження (0,5, 1, 2, 4 А). Крім того вони відрізняються кількістю входів (8, 16, 32, 64) та виходів (4, 8, 16, 32). Аналогові модулі мають різні вхідні та вихідні сигнали: напругу, струм, опір.

Спрощена структурна схема модуля аналогових входів та виходів наведена на рис. 7.5.

Оскільки модулі аналогових входів здатні працювати з різними сигналами, то спочатку проводиться нормування сигналів шляхом його перетворення у нормований сигнал. Потім сигнали за допомогою комутатора по черзі подаються на аналогово-цифровий перетворювач (АЦП), який перетворює їх у двійковий цифровий сигнал. Отримані з кожного входу значення надсилаються у внутрішню пам'ять модуля кожний за своєю адресою. Ці

значення опитує контролер із зазначенням адреси (команда **L PIW nnn**, де nnn - адреса відповідного входу) .

Модулі виходів аналогових приймають значення вихідних сигналів у внутрішню пам'ять модуля, а потім по черзі перетворюють їх в аналоговий сигнал за допомогою цифро-аналогового перетворювача (ЦАП). Значення вихідних сигналів завдають за допомогою команди **T PIW nnn**, де nnn – адреса відповідного виходу.

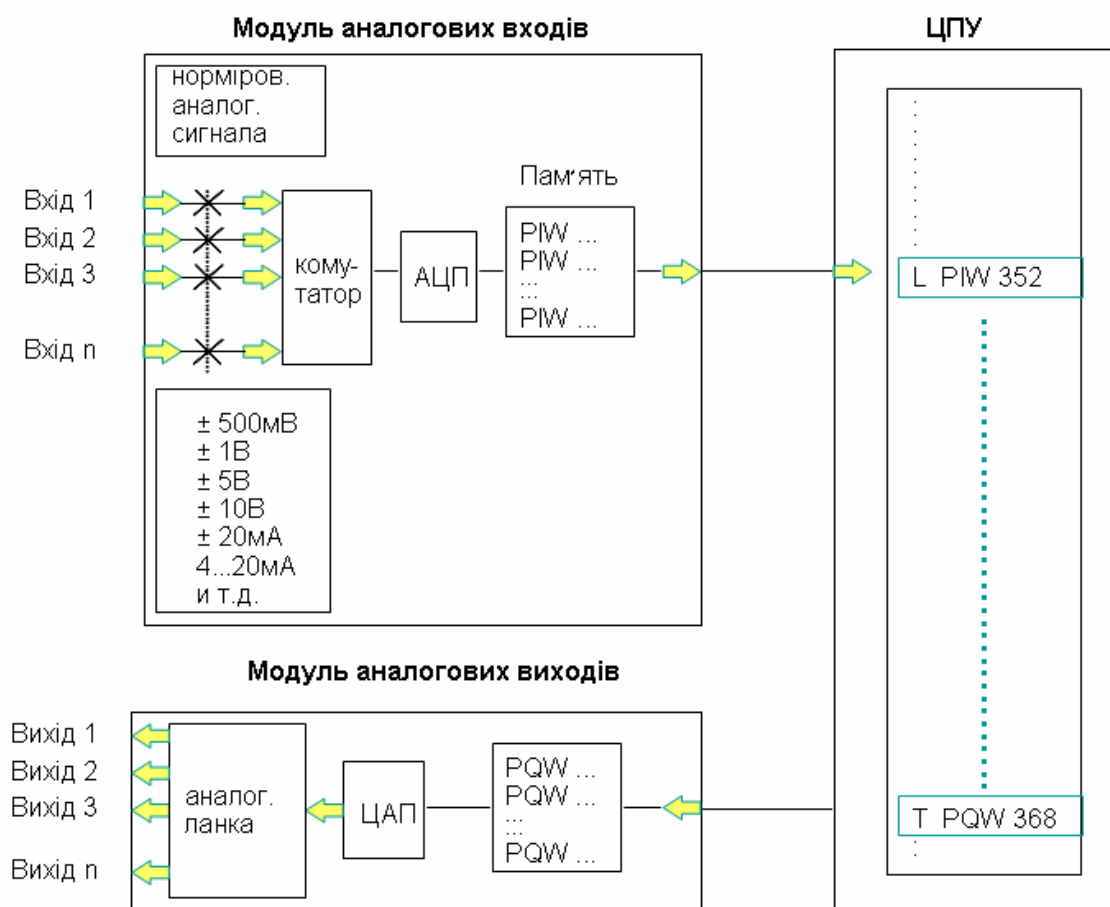


Рис. 7.5. Структура модулів аналогових входів та виходів.

Модулі дискретних та аналогових входів та виходів можуть забезпечити гальванічну розв'язку по входах та виходах.

Функціональні модулі відрізняються як функціями (лік, позиціонування і регулювання) так і кількістю каналів. Лічильні модулі призначені для підрахунку імпульсів фотоімпульсних датчиків і вимірюють позицію або швидкість переміщення і можуть використовуватись для простих систем позиціонування роботів. Модулі позиціонування призначені для створення систем позиціонування з різними виконавчими пристроями та датчиками зворотного зв'язку для позиціонування з однією або багатьма (до 4) осями. Такі модулі можуть виконувати функції позиційного та контурного керування роботом з використанням різних методів лінійної та колової інтерполяції.

Наприклад, модуль позиціонування FM353 контролера S7-300 здійснює керування одним кроковим двигуном.

Модуль позиціонування FM357-2 має чотири канали та здійснює позиційне або контурне керування кроковими чи серводвигунами (рис. 7.6). Цей модуль має також 4 вимірювальних входів для підключення датчиків положення.

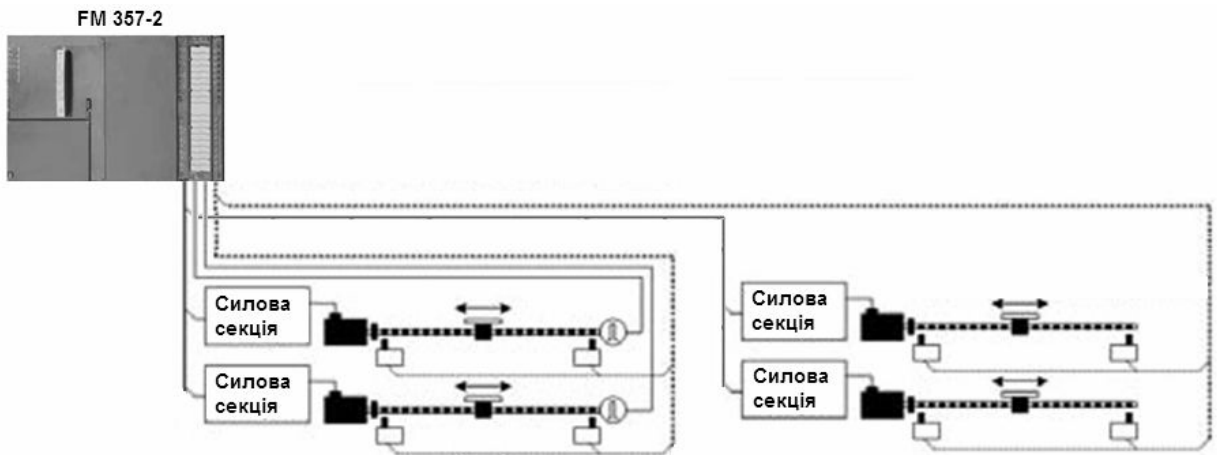


Рис. 7.6. Модуль позиціонування FM357-2

Модулі регулювання призначені для одно або багатоканального регулювання. Для цього використовуються ПД та ФАЗІ-регулятори.

Комунікаційні процесори здійснюють зв'язок системи керування з іншими пристроями по різних каналах зв'язку, наприклад, точка до точки, PROFIBUS та Industrial Ethernet.

7.4. Мови програмування програмованих логічних контролерів

На відміну від персональних комп'ютерів програма в ПЛК проектується один раз, а зміни в програмі здійснюються тільки у разі зміни технологічного процесу. Тому в складі ПЛК немає засобів проектування, а самі програмовані логічні контролери представляють собою так звану систему виконання.

Система проектування використовується тільки на етапах проектування апаратних та програмних компонент пристрою керування, налагодження програми та діагностики в процесі роботи.

Таким чином у системах керування поділяють системи виконання (ПЛК, операторська панель) та системи проектування (додаткові засоби проектування). У ролі систем керування найчастіше використовують персональні комп'ютери з відповідним програмним забезпеченням та засобами підключення до ПЛК (рис. 7.7).

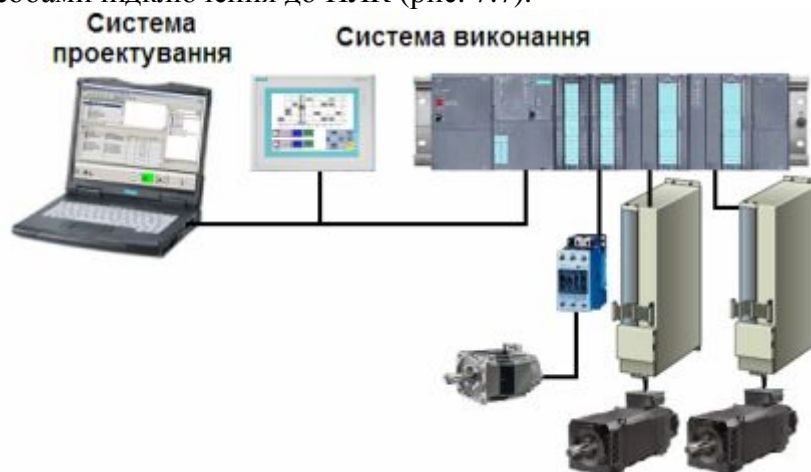


Рис. 7.7. Системи проектування та виконання

Мови програмування систем автоматизації розроблені на основі вимог стандарту ІЕС 61131-3.

Стандартизація мов програмування промислових контролерів має такі переваги:

Уніфікація промислового програмного забезпечення.

Графічна об'єктно-орієнтована представлення програми, що істотно скорочує час проектування.

Спільне використання даних. Доступ до даних здійснюється з всіх інструментальних засобів і інших програм. Це скорочує час розробки і знижує можливість виникнення помилок при запровадженні даних.

Уніфікована система інструментальних засобів. Для кожної фази виконання проекту можуть бути використані комфортні функції, що дозволяють вибирати конфігурацію апаратури і задавати параметри її настроювання, розробляти програмне забезпечення і документацію, виконувати наладку, запуск і обслуговування системи керування.

Відкритість. Системна платформа промислового програмного забезпечення відкрита для комп'ютерів, використання котрих істотно підвищує продуктивність системи.

Проблемно-орієнтовані інструментальні засоби. Інструментальні засоби можуть бути легко настроєні на використання в різноманітних галузях.

Частини програми можуть використовуватися багаторазово. Готові до використання частини програми можуть бути збережені в бібліотеках і використовуватися в наступних проектах

Структурне програмування. Проект може бути розбитий на складові частини, розробка яких може виконуватися паралельно.

Умонтовані діагностичні функції. Діагностичні функції дозволяють скоротити час простою системи і втрати, пов'язані з простоєм.

Промислове програмне забезпечення має модульну структуру. Інструментальні засоби можуть використовуватися незалежно друг від друга. Необхідність їх застосування визначається тільки характером задач автоматизації.

Програми підрозділяються на системні і прикладні.

Системні програми являють собою програми для реалізації внутрішніх робочих функцій пристрою керування. Однією з таких програм є операційна система контролера.

Прикладні програми являють собою програму для опрацювання сигналів і надання впливу на керований процес відповідно до задачі керування, яка складається за допомогою мов програмування..

Процесор контролера виконує команди послідовно друг за другом.

Існують такі основні форми представлення програми при програмуванні промислових контролерів:

- представлення у вигляді контактної плану (релейно-контактна схема),
- представлення у вигляді функціонального плану (функціональна схема),
- представлення у вигляді послідовності команд.

На рис. 7.8 наведені ці форми представлення програми.

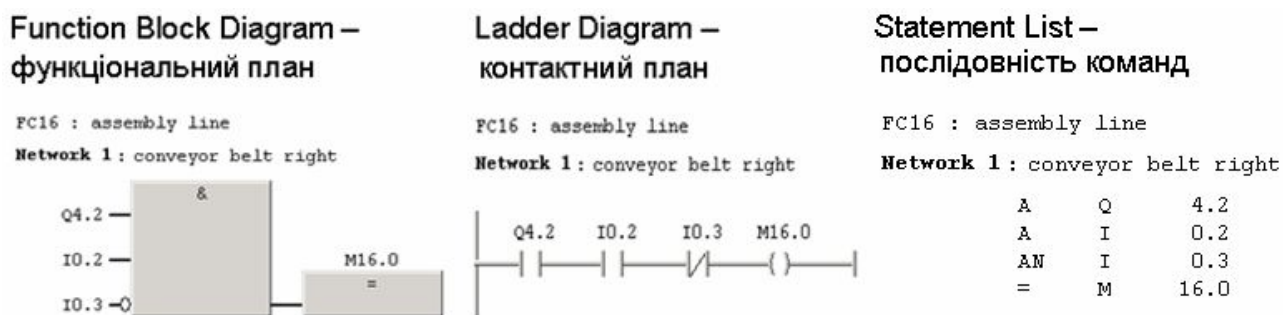


Рис. 7.8. Форми представлення програми при програмуванні ПЛК

Контактний план як би моделює роботу пристрою керування з використанням засобів релейної автоматики, представляючи його у вигляді принципової електричної схеми.

Функціональний план дозволяє наочно описати функції керування у вигляді схеми, що складається з окремих функціональних елементів.

Послідовність команд представляє програму у вигляді послідовності умовного позначення команд процесора.

Релейно-контактні і функціональні схеми дають зручне для оператора графічне уявлення алгоритму керування і не вимагають високого рівня підготовки у галузі обчислювальної техніки. Це дає можливість технологам приймати участь у розробці програми.

Представлення у вигляді послідовності команд найбільше близько до внутрішнього відображення керуючої програми в пристрої керування, тому воно найбільш часто використовується для системного програмування.

Зв'язок цих форм представлення з релейною автоматикою наведений у додатку 1.

У деяких випадках, найчастіше, коли мова йде о системах керування низького рівня, використовують тільки графічні форми представлення програми, а саме, контактний та функціональний плани.

Таке представлення програм використовується, наприклад, у ПЛК LOGO! фірми Сименс.

Програмування ПЛК LOGO! здійснюється за допомогою мови LOGO!Soft Comfort . Коротке представлення цієї мови наведено у додатку 2.

STEP7 надає користувачу всі три форми уявлення програм.

Контрольні запитання

1. Які характеристики визначають рівень складності контролера?
2. По якому принципу будуються програмовані логічні контролери?
3. З яких модулів складаються програмовані логічні контролери?
4. Які функції виконує модуль центрального процесора?
5. Які функції виконують сигнальні модулі?
6. Які дискретні сигнали використовують сигнальні модулі?
7. Як здійснюється введення аналогових сигналів?
9. Які функції виконують функціональні модулі?
9. Які функції виконують комунікаційні процесори?
10. У яких формах уявляється програма контролерів?

Глава 8. Проектування апаратних компонент систем керування на основі програмованих логічних контролерів

8.1. Складання проекту системи керування

Розглянемо програмні комплекси проектування програмованих логічних контролерів, наприклад, STEP7 фірми SIEMENS, що включає засоби проектування апаратних компонент системи керування, засоби програмування та засоби пошуку помилок під час налагодження та роботи системи.

Складання проекту системи керування робиться за допомогою усіх цих засобів.

У програмному комплексі STEP7 складання проекту здійснюється в керуючій програмі **SIMATIC-Manager**.

Нижче наведена структура проекту (рис. 8.1).

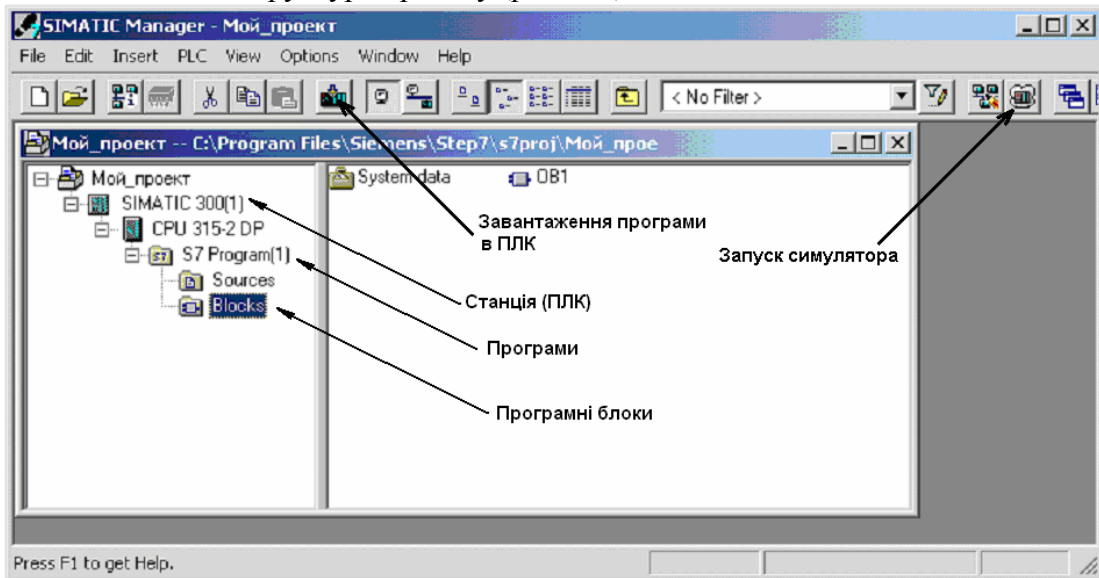


Рис. 8.1. Структура проекту в SIMATIC-Manager

Структура проекту складається з папки проекту (назва проекту визначається при її складанні). Проект складається з однієї або кількох станцій (якщо вони складають мережу). Для станції треба визначити її конфігурацію (склад), після чого створюється папка програм, де знаходяться програмні блоки та символічна таблиця.

Для створення конфігурації, програм та інших складових частин проекту використовуються різні засоби, які викликаються з керуючої програми "**SIMATIC-Manager**".

Проект може складатися з однієї або декількох станцій (ПЛК) (у кожній станції є папка з програмними блоками), а також додаткових компонент, у даному випадку це частотний перетворювач та операторська панель, за допомогою який можна перейти до відповідних програм (рис. 8.2)

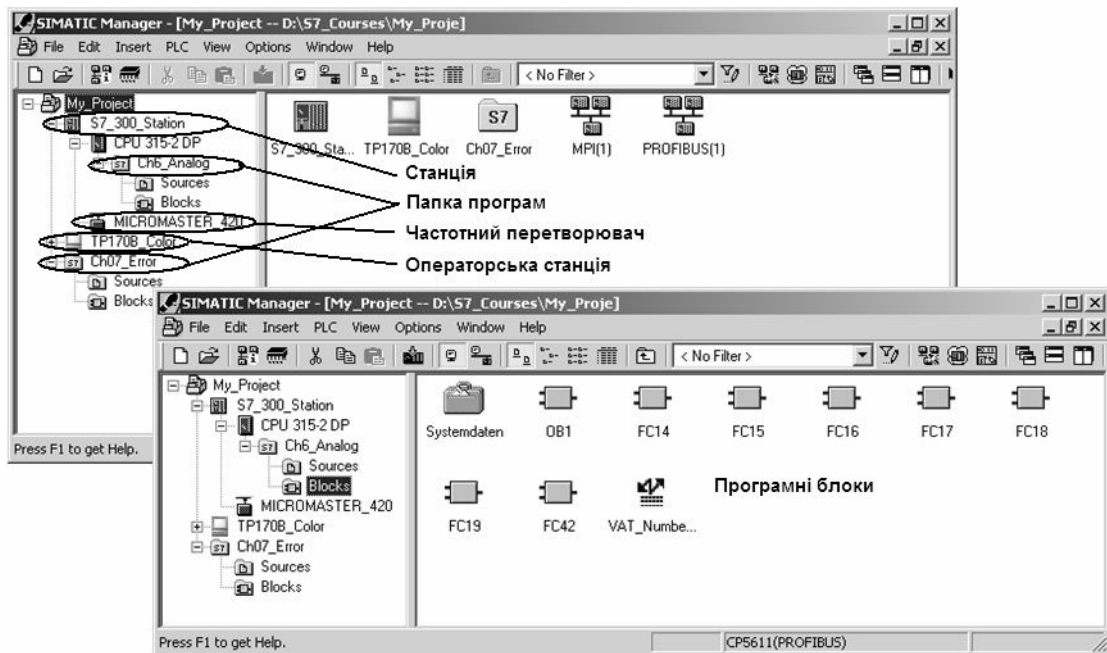


Рис.8.2. Структура проекту

Проектування проекту передбачає такі етапи.

Спочатку створюється сам проект з визначення його найменування та місця зберігання у пристрої програмування (рис. 8.3).

Після цього можливі такі варіанти проектування:

- комплексне проектування апаратних та програмних компонент;
- проектування тільки програмних компонент.

У першому випадку спочатку здійснюється проектування апаратних компонент з встановленням адрес усіх входів та виходів. Після закінчення проектування апаратних компонент створюється системний файл налагодження та встановлюється структура проекту. Далі здійснюється проектування програмних компонент з прив'язкою до реальних адрес пристрою керування.

У другому випадку в проект вставляються тільки папки для програм, де складаються заготовки для програм з відомими адресами входів і виходів, або замість адрес входів і виходів використовуються адреси у пам'яті (рис. 8.4).

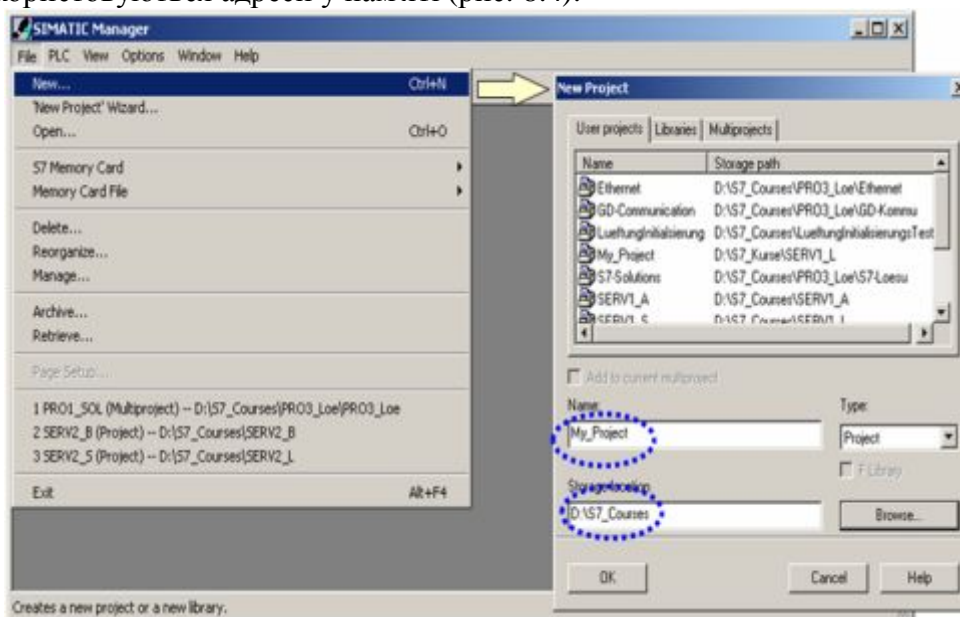


Рис. 8.3. Створення проекту

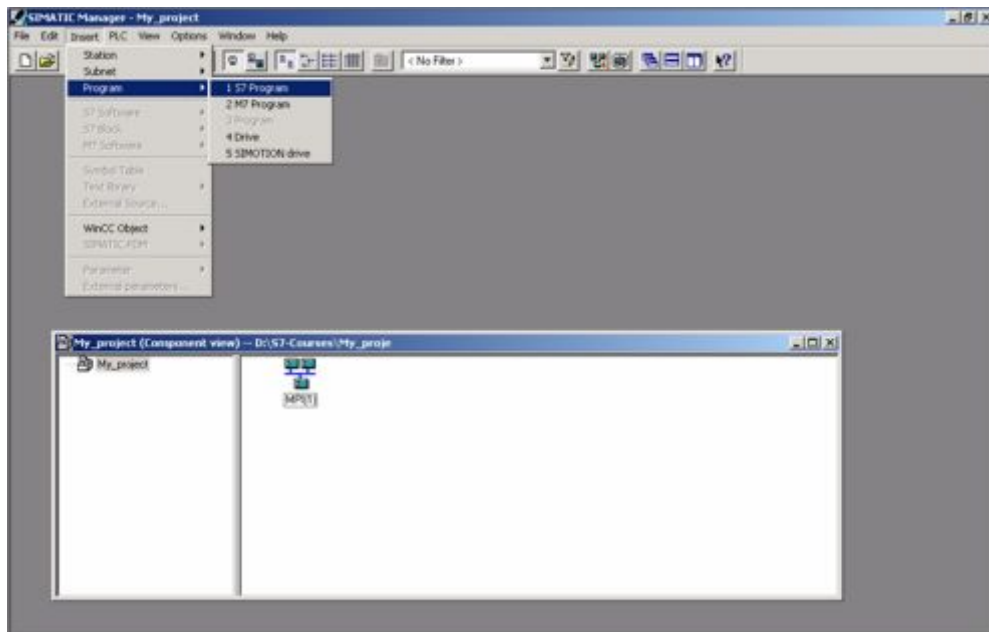


Рис. 8.4. Створення папки програм

У цій папці розміщуються усі програмні блоки, що створюються під час розробки програми.

Після того, як створені апаратні та програмні компоненти, здійснюється налагодження програм. Процес налагодження може бути поступовим і здійснюватися у разі готовності окремих складових частин програмних компонент.

8.2. Визначення структури і складу системи керування

Визначення структури і складу системи керування робиться за допомогою конфігуратора **HW Config**, який є складовою частиною мови програмування.

Він включає каталог з набором елементів системи та засоби визначення параметрів цих елементів. За їх допомогою здійснюються вибір конфігурації і завдання параметрів для налаштування апаратури і мереж. Таким чином усі параметри модулів, таких як центральний процесор, модулі введення та виводу як дискретні, так і аналогові та інші, визначаються програмним шляхом.

Для проектування апаратних компонент треба перейти до інструменту конфігурації (відкрити об'єкт "Станція"), після чого відкривається вікно конфігурації (рис. 8.5).

Проектування здійснюється шляхом вибору відповідних модулів з каталогу.

На початку треба вибрати носії модулів (RACK), а потім самі модулі.

У різних типів контролерів носії модулів можуть бути пасивними або активними. Так у контролера S7-300 носій модулів являє собою металеву рейку, на якій встановлюються модулі. У контролера S7-400 на носії модулів розташовані роз'єми, на які встановлюються модулі. При цьому існують носії модулів з різною кількістю роз'ємів (рис. 8.6).

Порядок встановлення модулів визначається типом контролера та типом модуля. Так на першому місці встановлюють модулі живлення, потім процесорні модулі.

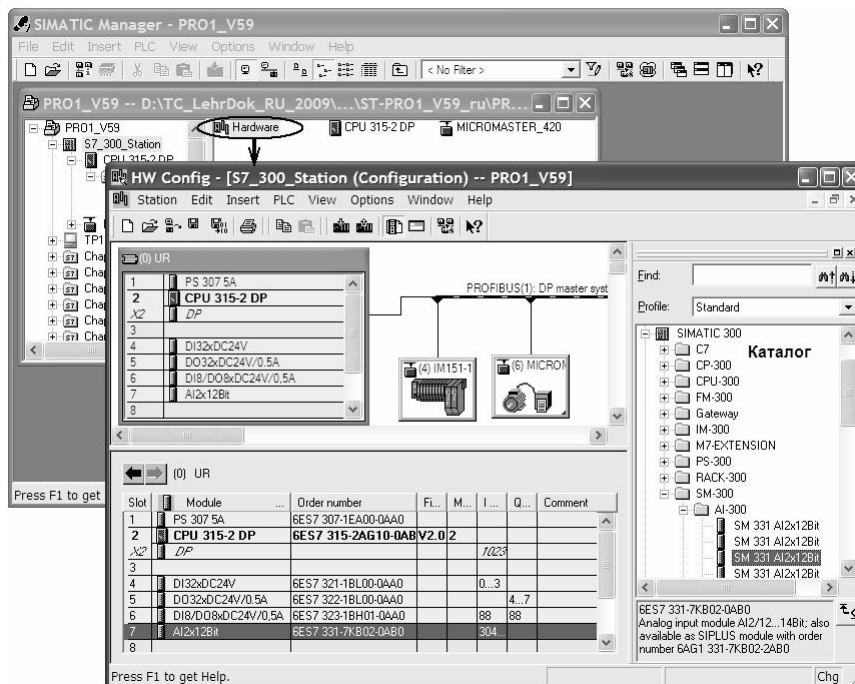


Рис. 8.5. Вікно конфігурації



Рис. 8.6. Носії модулів ПЛК S7-300 (а) та S7-400 (б)

Сигнальні, функціональні та комунікаційні модулі зазвичай встановлюють довільно. Інтерфейсні модулі у контролера S7-300 встановлюється на позиції 3 а у S7-400 на останньому.

Конфігуратор не дає можливості встановити модуль на неприпустиму позицію.

Адресація входів і виходів у модулях може задаватися самою системою, або у разі потреби користувачем з урахуванням певних правил.

На рис. 8.7 наведений приклад адресації окремих входів та виходів цифрових модулів контролера S7-300.

Адресація модулів S7-300

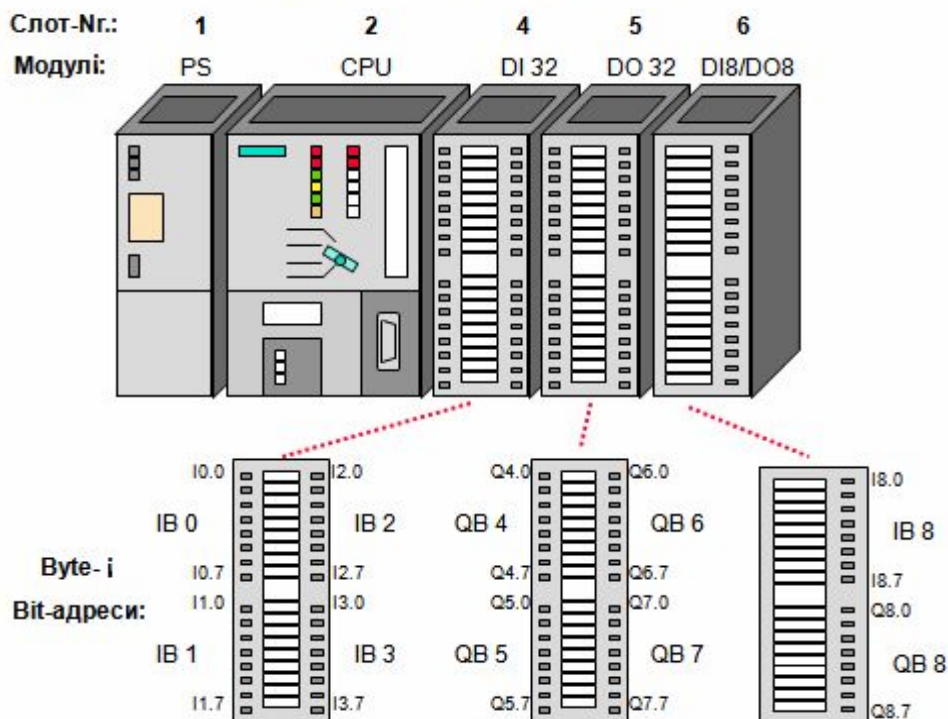


Рис. 8.7. Адресація цифрових входів та виходів контролера S7-300

На рис. 8.8 показана адресація цифрових модулів контролера S7-300 при використанні декількох носіїв модулів.

Стойка	Слот	Модуль	Адресація
Стойка 3	1	PS	
	2	IM (Получатель)	96.0 до 99.7
	3	IM (Получатель)	100.0 до 103.7
	4	IM (Получатель)	104.0 до 107.7
	5	IM (Получатель)	108.0 до 111.7
	6	IM (Получатель)	112.0 до 115.7
	7	IM (Получатель)	116.0 до 119.7
	8	IM (Получатель)	120.0 до 123.7
	9	IM (Получатель)	124.0 до 127.7
	10	IM (Получатель)	128.0 до 131.7
Стойка 2	1	PS	
	2	IM (Получатель)	64.0 до 67.7
	3	IM (Получатель)	68.0 до 70.7
	4	IM (Получатель)	72.0 до 75.7
	5	IM (Получатель)	76.0 до 79.7
	6	IM (Получатель)	80.0 до 83.7
	7	IM (Получатель)	84.0 до 87.7
	8	IM (Получатель)	88.0 до 91.7
	9	IM (Получатель)	92.0 до 95.7
	10	IM (Получатель)	96.0 до 99.7
Стойка 1	1	PS	
	2	IM (Получатель)	32.0 до 35.7
	3	IM (Получатель)	36.0 до 39.7
	4	IM (Получатель)	40.0 до 43.7
	5	IM (Получатель)	44.0 до 47.7
	6	IM (Получатель)	48.0 до 51.7
	7	IM (Получатель)	52.0 до 55.7
	8	IM (Получатель)	56.0 до 59.7
	9	IM (Получатель)	60.0 до 63.7
	10	IM (Получатель)	64.0 до 67.7
Стойка 0	1	PS	
	2	CPU	
	3	IM (Отправитель)	0.0 до 3.7
	4	IM (Отправитель)	4.0 до 7.7
	5	IM (Отправитель)	8.0 до 11.7
	6	IM (Отправитель)	12.0 до 15.7
	7	IM (Отправитель)	16.0 до 19.7
	8	IM (Отправитель)	20.0 до 23.7
	9	IM (Отправитель)	24.0 до 27.7
	10	IM (Отправитель)	28.0 до 31.7
	11	IM (Отправитель)	32.0 до 35.7

Рис. 8.8. Адресація цифрових модулів

На рис. 8.9 показана адресація модулів аналогових входів та виходів контролера S7-300.

Стойка 3	Блок питания	IM (Получатель)	640	656	672	688	704	720	736	752
			до 654	до 670	до 686	до 702	до 718	до 734	до 750	до 766
Стойка 2	Блок питания	IM (Получатель)	512	528	544	560	576	592	608	624
			до 526	до 542	до 558	до 574	до 590	до 606	до 622	до 638
Стойка 1	Блок питания	IM (Получатель)	384	400	416	432	448	464	480	496
			до 398	до 414	до 430	до 446	до 462	до 478	до 494	до 510
0	Блок питания	CPU (Отправитель)	256	272	288	304	320	336	352	368
			до 270	до 286	до 302	до 318	до 334	до 350	до 366	до 382
Слот	2	3	4	5	6	7	8	9	10	11

Рис. 8.8. Адресация модулей аналоговых входов та виходів

Різниця адрес цифрових та аналогових модулів пов'язана з тим, що цифровий модуль може мати до 32 входів або виходів, тому займає 4 байти в адресному просторі.

Аналоговий модуль може мати до 8 входів або виходів, кожний з котрих займає 2 байти, тому на один модуль відводиться 16 байтів.

За допомогою конфігуратора також встановлюються додаткові пристрої, такі як, децентралізована периферія, частотні перетворювачі тощо.

8.3. Налаштування системи керування

За допомогою конфігуратора **HW Config** здійснюється також налаштування системи керування.

Для кожного модуля можна відкрити вікно його налаштування, в якому можна здійснити встановлення відповідних параметрів.

Спочатку встановлюються головні параметри (рис.8.3), у даному випадку адреса локальної мережі MPI, за допомогою якої контролер підключається до пристрою програмування.

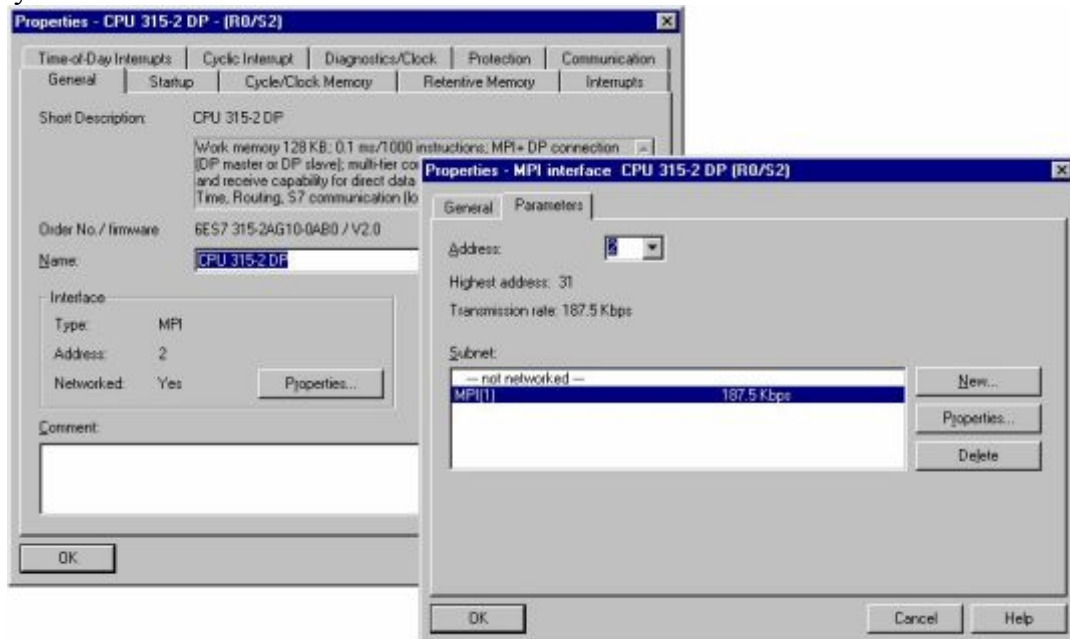


Рис. 8.3. Підключення до локальної мережі MPI

Далі показано, як у вікні процесорного модуля встановлюються потрібні параметри (рис. 8.4), наприклад, включення тактового байта меркерів (бітів пам'яті), який виконує функції тактових генераторів з різними частотами (від 10 до 0,5 Гц)

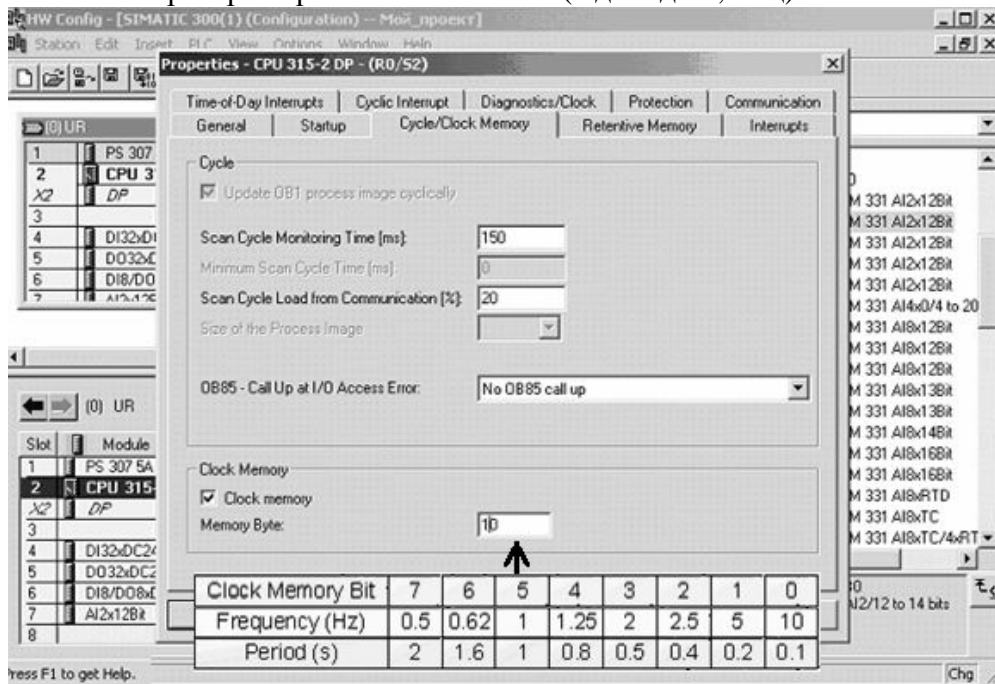


Рис. 8.4. Включення тактового байта меркерів

На рис. 8.5 показано встановлення параметрів для модуля аналогових входів (вибір типу вхідного сигналу та діапазону вимірювання).

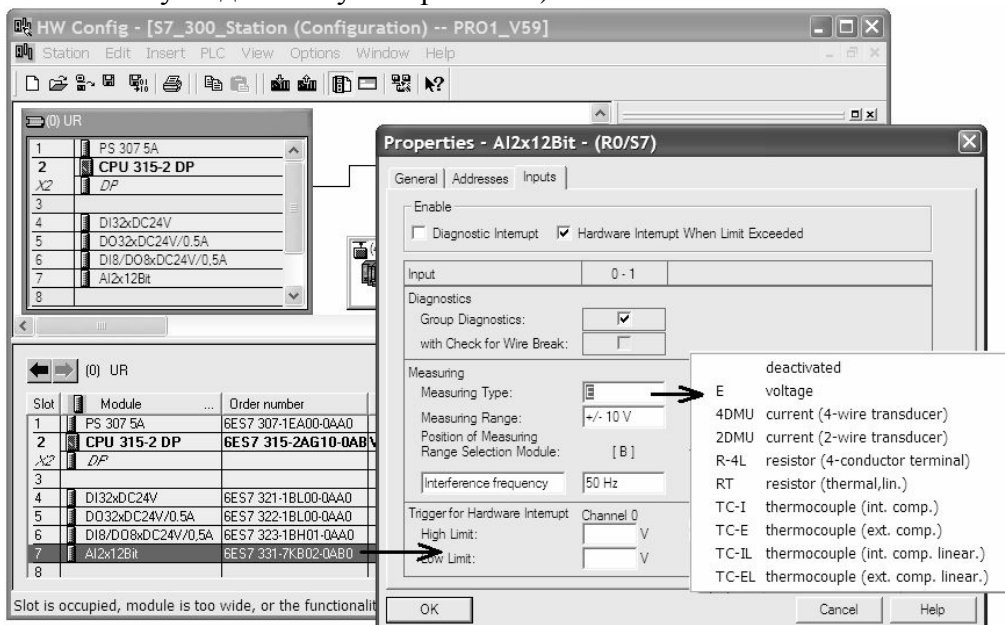


Рис. 8.5. Встановлення параметрів для модуля аналогових входів

Модулі аналогових входів можуть вимірювати напругу, струм, опір, температуру.

Для напруги, струму та опору можна встановити такі діапазони.

Напруга (біполярний сигнал):

$\pm 80\text{mV}$, $\pm 250\text{ mV}$, $\pm 500\text{ mV}$, $\pm 2,5\text{ V}$, $\pm 1\text{ V}$, $\pm 5\text{V}$, $\pm 10\text{V}$.

Напруга (однополярний сигнал): $0\dots 2\text{ V}$, $1\dots 5\text{ V}$.

Струм (біполярний сигнал): $\pm 3,2\text{ mA}$, $\pm 10\text{ mA}$, $\pm 20\text{ mA}$.

Струм (однополярний сигнал): $0\dots 20\text{ mA}$, $4\dots 20\text{ mA}$.

Опір: 0 ... 150 Ом, 0 ... 300 Ом, 0 ... 600 Ом

При вимірюванні температури модуль видає значення з точністю 0,1°C.

При налагодженні функціональних модулів кількість параметрів може бути досить великою.

При використанні додаткових модулів, які відсутні в конфігураторі, відповідні програмні модулі можна знайти в Інтернеті та завантажити їх в конфігуратор.

Контрольні запитання

1. Як здійснюється складання проекту системи керування?
2. З чого складається структура проекту системи керування?
3. Де знаходяться програмні блоки?
4. Як здійснюється проектування апаратних компонент (конфігурації системи керування)?
5. Та які позиції встановлюються модулі живлення та процесорні модулі
6. Як встановлюється адресація цифрових модулів?
7. Як встановлюється адресація аналогових модулів?
8. Як здійснюється налагодження окремих модулів?
9. Які параметри встановлюються на головному вікні налагодження процесорного модуля?
10. Які параметри встановлюються для модуля аналогових входів?

Глава 9. Проектування програмних компонент систем керування на основі програмованих логічних контролерів

9.1. Структура програм програмованих логічних контролерів

У мові програмування STEP7, згідно з стандартом, розрізняють блоки, що містять команди для обробки сигналів (організаційні **OB**, функції **FC** і функціональні блоки **FB**), а також блоки, у яких зберігаються дані (**DB**).

Організаційні блоки (**OB**) служать для керування прикладною програмою у формі переліку блоків програми. Є організаційні блоки для циклічної обробки програм, для обробки з керуванням по перериваннях і для обробки з керуванням за часом. Організаційний блок є обов'язковим, наприклад, **OB1** для циклічного виконання програми.

Функціональні блоки (**FB**) і функції (**FC**) реалізують часто повторювані або дуже складні функції. Функціональні блоки і функції можуть бути стандартними (**SFB**, **SFC**) або програмуються самим користувачем. Функціональні блоки і функції можуть мати параметри і параметрируються шляхом завдання різноманітних значень параметрів блока за допомогою операндів.

У блоках даних (**DB**) знаходяться дані, використовувані в програмі користувача.

Для кращої наочності блоки програми користувача (**OB**, **FB**, **FC**) можуть розділятися на окремі фрагменти - схеми.

У програмі використовуються такі змінні: входи **I**, наприклад, **I 10.1**, **IB 20**, **IW 40**, **ID 100**; виходи **Q**, наприклад, **Q 0.0**, **QB 12**, **QW 24**, **QD 30**; пам'ять **M** (меркери), наприклад, **M 200.0**, **MB 220**, **MW 300**, **MD 400**.

До змінних можна звертатися як до окремих бітів (**I 10.1**), так і байтів (**MB 220**), слів (**QW 24**), та подвійних слів (**ID 100**).

Структура програми програмованого логічного контролера SIMATIC-S7 наведена на рис. 9.1.

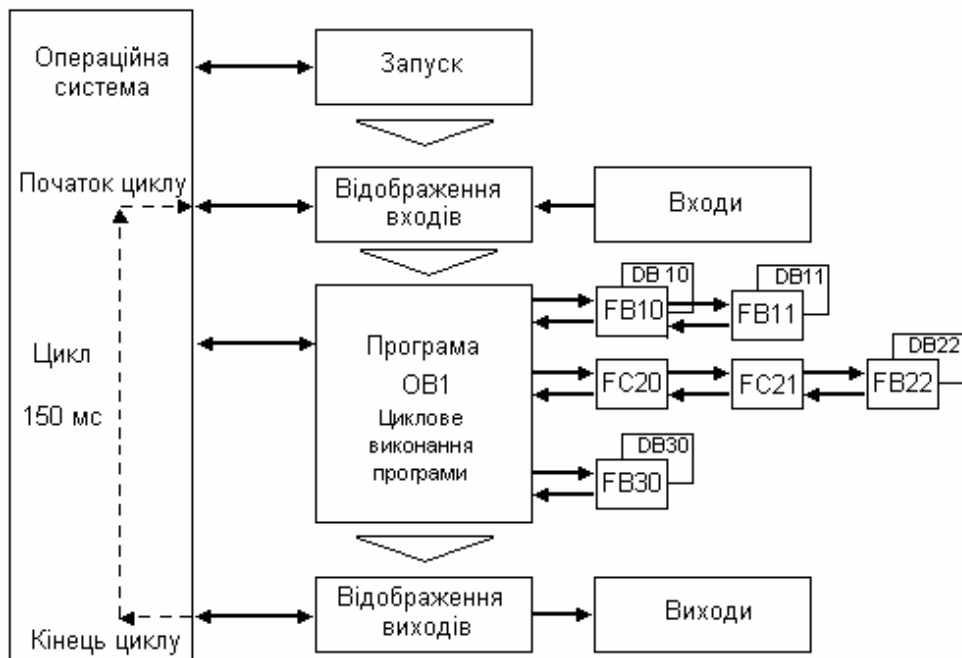


Рис. 9.1. Структура програми

Операційна система забезпечує роботу контролера та визначає порядок виконання програми. При включенні контролера запускається організаційний блок запуску (**OB 100**), який виконується тільки один раз. Потім починається циклічне виконання програми (**OB1**).

На початку циклу здійснюється опитування усіх входів, та отримані значення розміщуються у пам'яті в області відображення входів. При виконанні програми звертання до входів та виходів здійснюється через їхні області відображення у пам'яті. Наприкінці циклу значення з області відображення виходів записуються у виходи контролера. Час виконання ОВ1 обмежений максимальною тривалістю циклу (встановлено значення 150 мс, але його можна змінювати).

При упорядкуванні програми найчастіше використовується структурне програмування.

При структурному програмуванні програма ділиться на декілька блоків, що послідовно викликаються з організаційного блока (для циклічної обробки - з **ОВ1**).

Для створення програмних блоків ОВ, FB, FC використовують програмні редактори відповідно з формою представлення програми.

Редактори для представлення програм у вигляді контактного та функціонального плану дозволяють здійснювати вибір окремих команд за допомогою каталогу.

На рис 9.2 наведений програмний редактор для створення програми у вигляді контактного плану. Каталог з окремими командами знаходиться у лівій частині редактора.

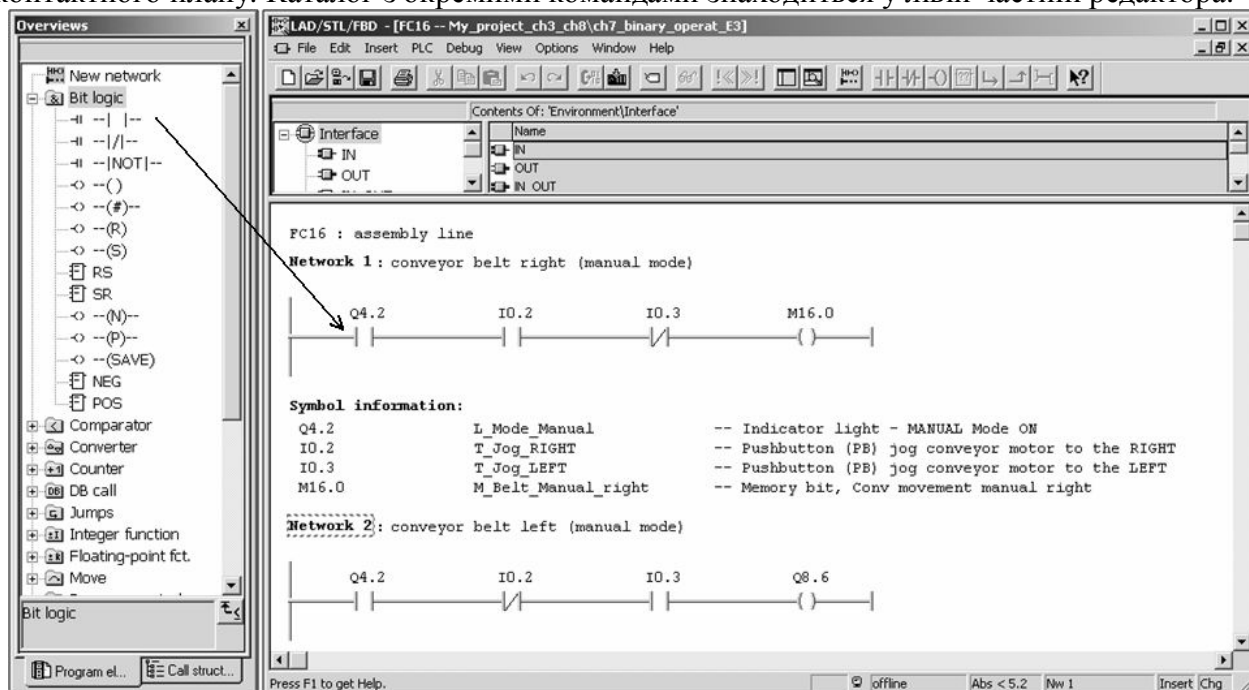


Рис 9.2. Програмний редактор для створення програми у вигляді контактного плану

9.2. Система команд програмованих логічних контролерів

Сучасні програмовані логічні контролери (ПЛК) мають логічні команди над двійковими числами, функції ліку (лічильники), часу (таймери), а також функції математичної та логічної обробки чисел у різних форматах.

Основні елементи логічних функцій для контактного та функціонального плану показані на рис. 9.3.

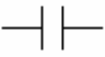
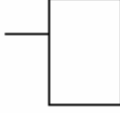
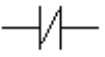
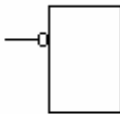
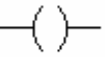
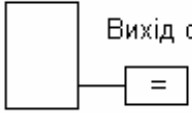
Елемент		Функція
Контактний план	Функціональний план	
Нормально відкритий контакт 	Вхід функції 	Результат опиту входу дорівнює "1", якщо стан сигналу операнда на вході дорівнює "1". Якщо стан сигналу операнда на вході дорівнює "0", то результат опиту також дорівнює "0".
Нормально закритий контакт 	Інверсний вхід функції 	Результат опиту входу дорівнює "0", якщо стан сигналу операнда на вході дорівнює "1". Якщо стан сигналу операнда на вході дорівнює "0", то результат опиту дорівнює "1".
Виконуючий пристрій 	Вихід функції 	Операнд, вихідні функції, має стан "1", якщо результат логічної операції цьєї функції дорівнює "1". Операнд має стан "0", якщо результат логічної операції цьєї функції дорівнює "0".

Рис. 9.3. Основні елементи контактного та функціонального плану

Нормально відкритий контакт відповідає входу функції, а нормально закритий контакт відповідає інверсному входу.

На рис. 9.4 показані основні логічні функції, такі як логічне І, логічне АБО та виключне АБО.

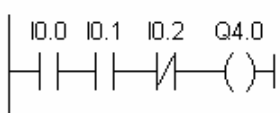
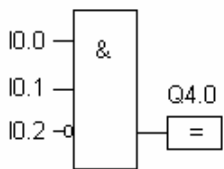
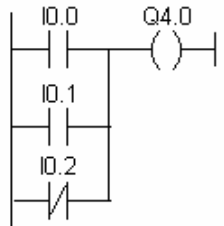
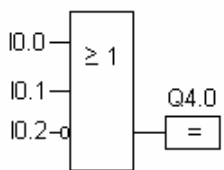
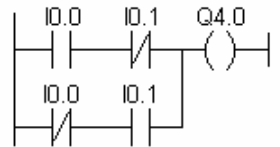
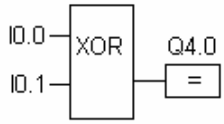
Функція	Контактний план	Функціональний план	Послідовність команд
Логічне І			A IO.0 A IO.1 AN IO.2 = Q4.0
Логічне АБО			O IO.0 O IO.1 ON IO.2 = Q4.0
Виключне АБО			X IO.0 X IO.1 = Q4.0

Рис. 9.4. Основні логічні функції

Результати виконання логічних функцій показані у таблиці 9.1.

Таблиця 9.1.

Вхідні сигнали		Вихідні сигнали Q4.0		
IO.0	IO.1	Логічне І	Логічне АБО	Виключне АБО
0	0	0	0	0

0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

До функцій над двійковими числами можна також віднести функції пам'яті, а само RS тригери (рис. 9.5), які запам'ятовують значення вхідних сигналів. Назва тригера визначається назвою виконуваних функцій, а само, S – set встановлює вихід у одиницю, а R – reset скидає вихід у нуль. При наявності сигналу на обох входах одиниць пріоритет має той вхід, який опитується останнім.

Функція	Контактний план	Функціональний план	Послідовність команд
Функція RS тригера з пріоритетом встановлення в "1"			A IO.0 R M0.0 A IO.1 S M0.0 A M0.0 = Q4.0
Функція SR тригера з пріоритетом скидання у "0"			A IO.0 S M0.0 A IO.1 R M0.0 A M0.0 = Q4.0
Скорочені функції S та R			A IO.0 S Q4.0 A IO.1 R Q4.0

Рис. 9.5. Функції пам'яті

Лічильники призначені для обліку вхідних імпульсних сигналів. Вони можуть лічити у прямому та зворотному напрямку. Приклад позначення такого лічильника наведено на рис. 9.6. На цьому прикладі: CU – лічильний вхід у прямому напрямку, CD – лічильний вхід у зворотному напрямку, S – вхід установки початкового значення, R – вхід скидання лічильника, PV – початкове значення, CV – значення лічильника у двійковому коді, CV_BCD – значення лічильника у двійково-десятьковому коді, Q – двійковий вихід лічильника.

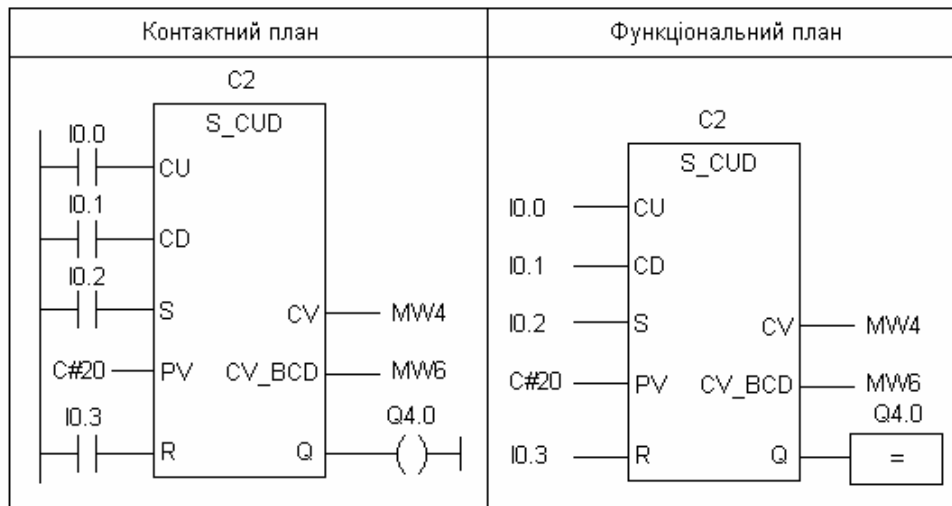


Рис. 9.6. Лічильник

Таймери виконують функції часу шляхом затримки сигналу або формування імпульсу певної тривалості. Приклад позначення таймера наведено на рис. 9.7. На цьому прикладі: S – вхід запуску таймера, R – вхід скидання таймера, TV – затримка, CV – значення лічильника таймера у двійковому коді, CV_BCD – значення лічильника таймера у двійково-десятковому коді, Q – двійковий вихід таймера.

Існують різні таймери, наприклад, таймери з затримкою включення, імпульсні таймери та таймери з затримкою виключення.

Функції порівняння порівнюють два числа IN1 та IN2, та в разі здійснення порівняння видають сигнал “1”, у протилежному випадку видають сигнал “0”. Виконуються такі порівняння: $IN1 == IN2$, $IN1 <> IN2$, $IN1 > IN2$, $IN1 < IN2$, $IN1 >= IN2$, $IN1 <= IN2$.

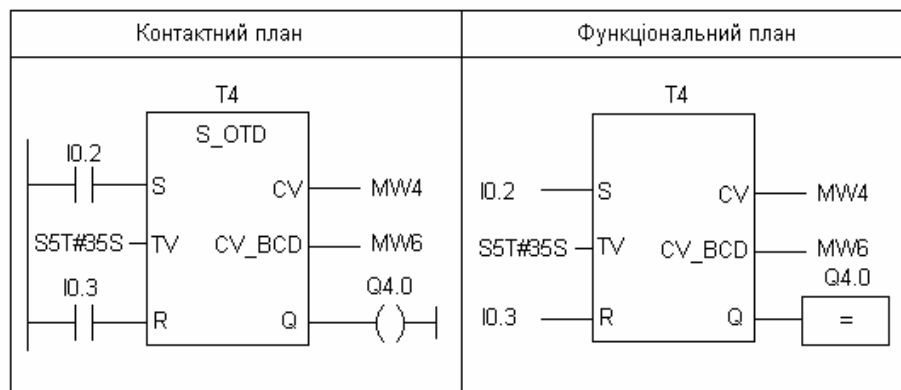


Рис. 9.7. Таймер

Приклад позначення функції порівняння сигналу з аналогового входу PIW 512, який може приймати значення від – 47648 до 47648 та константи 15000 наведено на рис. 9.8.

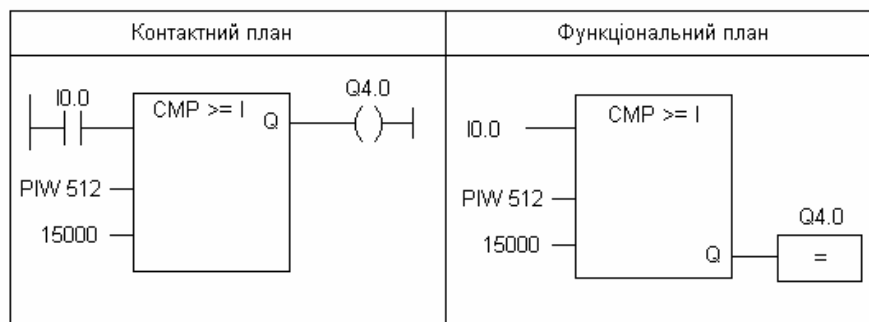


Рис. 9.8. Приклад позначення функції порівняння

Арифметичні функції виконують операції складання (ADD), віднімання (SUB), множення (MUL) та ділення (DIV) чисел у різних форматах. Приклад арифметичної функції складання для чисел типу Integer (негативні та позитивні цілі числа розміром 16 біт) наведений на рис. 9.9.

Розглянуті команди утворюють так званий базовий набір команд і використовуються для створення програм керування загального призначення. Для створення системних програм, наприклад програм, які описують детальну роботу окремих модулів системи, використовується розширений набір команд. Ці команди близькі до команд мови Асемблер.

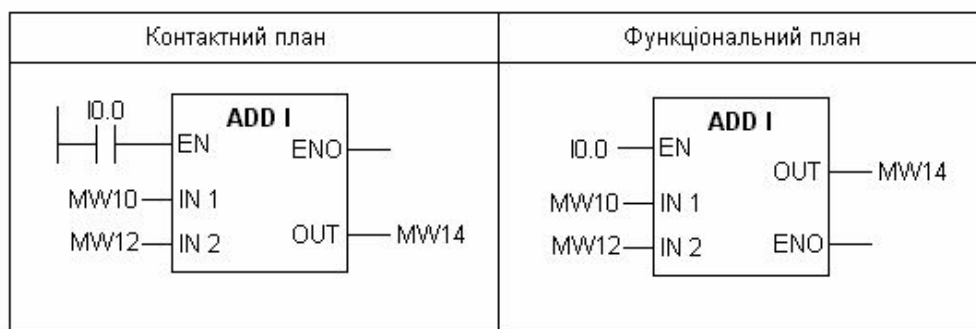


Рис. 9.9. Приклад арифметичної функції складання для чисел типу Integer

Розглянуті команди утворюють так званий базовий набір команд і використовуються для створення програм керування загального призначення. Для створення системних програм, наприклад програм, які описують детальну роботу окремих модулів системи, використовується розширений набір команд. Ці команди близькі до команд мови Асемблер.

У складі блоків є так звані функції та функціональні блоки. Особливістю таких блоків є можливість використовувати замість фактичних параметрів (входів, виходів, пам'яті) формальні параметри у вигляді позначень. При виклику такої функції або функціонального блока замість формальних параметрів треба підставити фактичні параметри (рис. 9.10).

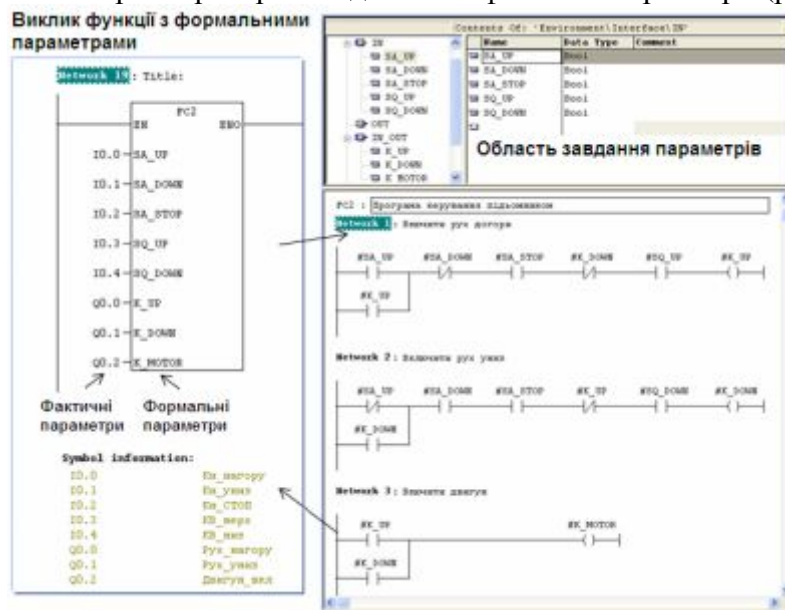


Рис. 9.10. Виклик функції з формальними параметрами

Це дає можливість при необхідності створити додаткові функції та використовувати їх у програмі.

У разі наявності декількох однакових пристроїв для їхнього керування можна створити відповідну функцію.

У складі мови програмування є велика бібліотека стандартних та системних функцій, наприклад, функцій для обміну даними у локальних мережах.

9.3. Засоби налагодження програми та пошуку помилок

Однією з важливіших задач при створенні систем керування є її налагодження та пошук помилок як на стадії проектування так і у період експлуатації. Програмний пакет STEP7 для ПЛК SIMATIC S7 мають великий набір засобів для вирішення цієї задачі. На етапі складення програми для її перевірки досить складно використовувати апаратні компоненти системи автоматизації, тому є можливість перевірки програми за допомогою програмної моделі ПЛК S7-PLCSIM.

Це так званий симулятор, в якому можна визначити усі змінні реального ПЛК, такі як входи, виходи, пам'ять, таймери, лічильники, та моделювати різні режими роботи ПЛК, наприклад, режими запуску, переривання, реакцію на помилки тощо (рис. 9.11).

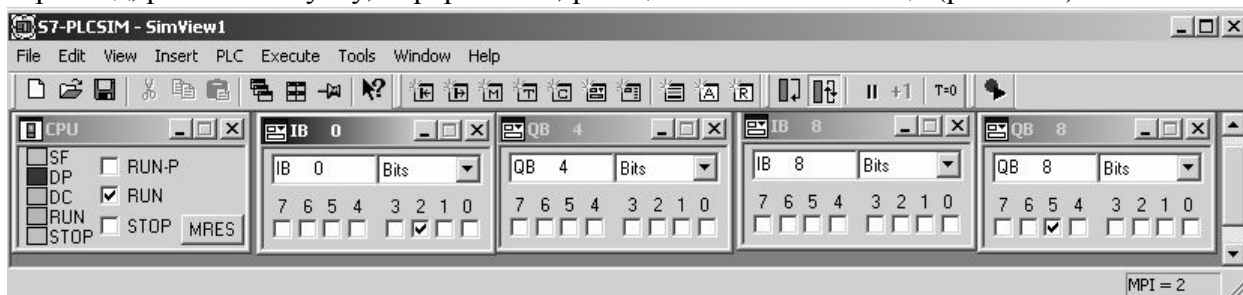


Рис. 9.11. Симулятор для ПЛК S7-PLCSIM

В симулятор можна завантажити програму та конфігурацію ПЛК, запустити її та простежити за реакцією ПЛК, а саме, як вона реагує на зміну входних змінних, що імітує сигнали датчиків, або як змінюється внутрішній стан ПЛК у часі, якщо працюють таймери і т.д.

Крім цього є можливість перевірити стан змінних у самої програми незалежно від того, виконується вона у симуляторі, чи у реальному ПЛК. Першою такою можливістю є перевірка стану програми (тестування програми). Для цього треба підключити пристрій програмування до ПЛК, або запустити стимулятор, завантажити програму, перевести ПЛК у режим RUN (робота), відкрити відповідний блок програми у редакторі, натиснути кнопку "окуляри" на панелі команд. Після цього для контактного та функціонального планів стан двійкових змінних буде визначений кольором – зелений для стану 1, та синій для стану 0. Числові змінні надаються у вигляді чисел з заданою формою відображення. Для послідовності команд усі змінні надаються у вигляді чисел (рис. 9.12).

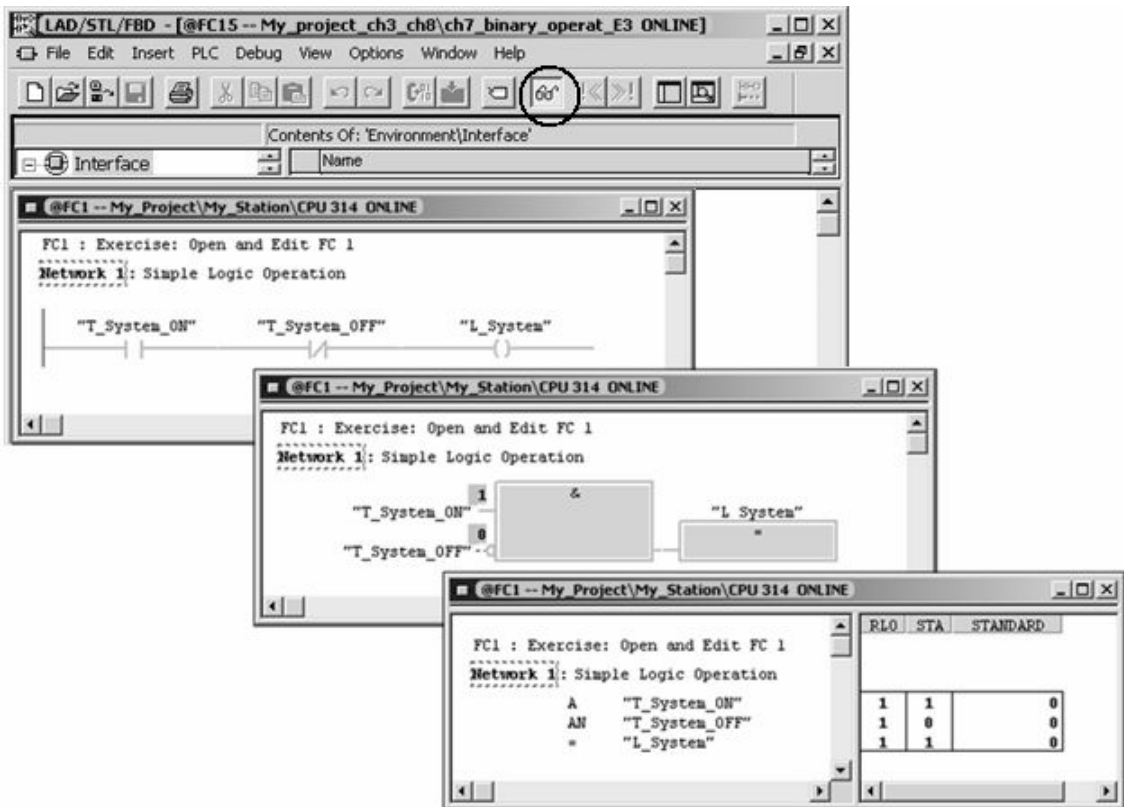


Рис. . 9.12. Тестування програми

Друга можливість перевірки стану змінних надається у таблиці перегляду та модифікації змінних (рис. 9.13). У таблицю треба занести усі змінні з програми, стан яких треба переглядати. Ця таблиця дає можливість змінювати існуючі дані з метою перевірки їх впливу на виконання програми.

Address	Symbol	Symbol comment	Display format	Status value	Modify value
Q 4.2	"L_MAN"	Manual Mode of Operation Light	BOOL	false	
Q 4.3	"L_AUTO"	Automatic Mode of Operation Light	BOOL	false	
I 0.2	"T_Jog_RT"	Jog Conveyor Right, Momentary Contact	BOOL	false	
I 0.3	"T_Jog_LT"	Jog Conveyor Left, Momentary Contact	BOOL	false	
Q 8.5	"K_RT"	Run Conveyor Right	BOOL	false	
Q 8.6	"K_LT"	Run Conveyor Left	BOOL	false	
IW 2	"TW_BCD"	BCD Push Buttons - Input Word	HEX	W#16#1021	

Рис. 9.13. Таблиця перегляду та модифікації змінних програми

На стадії налагодження програми для складних систем автоматизації такі таблиці доцільно скласти окремо для всіх вузлів системи, які можна налагоджувати окремо.

Операційна система ПЛК у кожному циклі перевіряє правильність виконання програми та наявність апаратних та програмних помилок. У складі організаційних блоків є блоки переривання за помилкою, які можна ввести в програму. Якщо операційна система знаходить таку помилку, вона викликає відповідний організаційний блок помилки. У разі відсутності такого блоку процесор переходить у стан "СТОП".

Організаційні блоки відповідають досить великому класу помилок, наприклад, у разі виникнення програмної помилки викликається організаційний блок OB121. Для більш детального визначення причин виникнення помилок у тимчасових змінних організаційних блоків міститься код помилки.

Наприклад, на рис 9.14 наведений організаційний блок OB121, у якому перевіряється код помилки.

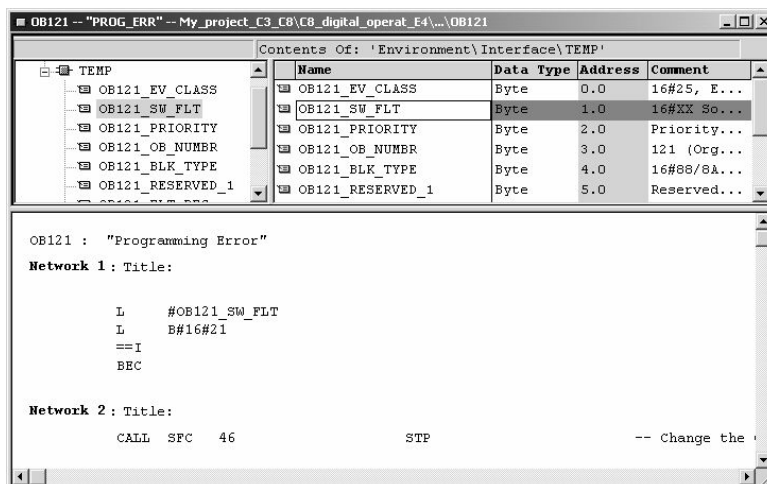


Рис. 9.14. Приклад OB121, у якому перевіряється код помилки

Якщо цей код дорівнює значенню B#16#21, то має місце помилка представлення числа у двійково-десятковому форматі та ця помилка ігнорується. Якщо має місце інша помилка, процесор зупиняється.

Після того, як помилка викликала зупинку процесора у разі відсутності відповідного організаційного блоку, є можливість визначити причину зупинки. Для цього треба відкрити вікно "**Module Information**". У цьому вікні є закладки, які несуть інформацію о причинах помилки, а також стану процесора на час зупини. Наприклад, на рис. 9.15 наведена закладка "**Diagnostic Buffer**" (діагностичний буфер).

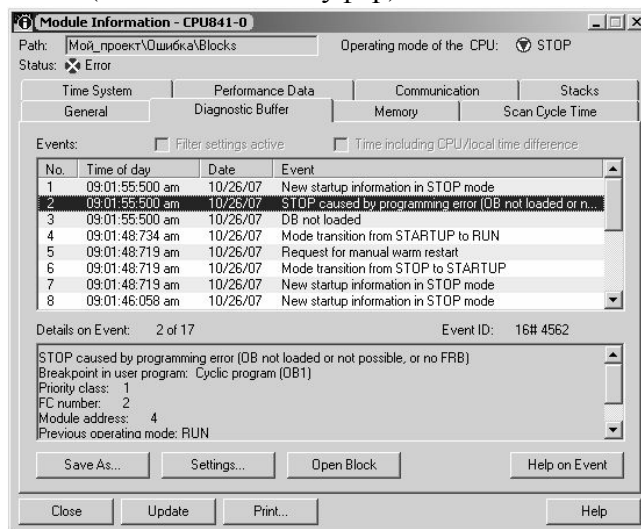


Рис. 9.15. Діагностичний буфер

У буфері зберігається інформація о причинах виникнення 10 останніх помилок, або інших подій, які визначає операційна система. Буфер представляється у вигляді таблиці, де визначаються час і дата виникнення події, а також опис цієї події. За допомогою кнопки **Open Block** можна відкрити програмний блок у місці, де виникла помилка.

Додаткову інформацію можна знайти в інших закладках, наприклад, інформацію про змінні у регістрах процесора, послідовність блоків, які викликані до появи помилки та інші.

Крім цього є можливість за допомогою вікна конфігурації отримати інформацію о помилках, які виникають у модулях, наприклад, відсутність зовнішнього живлення у

аналогових модулях, обрив дроту у модулів з діагностикою тощо (рис. 9.16). Але ця інформація може бути отримана тільки після зупинки процесора.

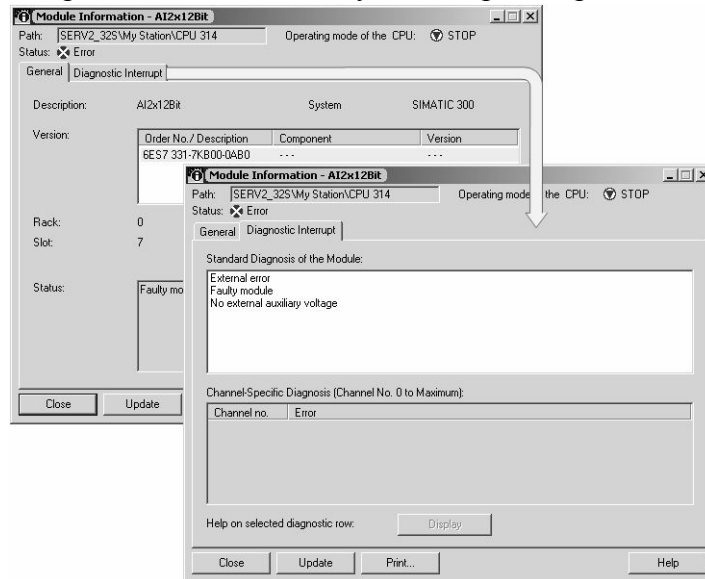


Рис. 9.16. Інформацію о помилках, які виникають у модулях

Якщо були викликані відповідні ОВ помилок, та процесор продовжує роботу, для визначення помилок треба в ОВ скласти програму, де буде запрограмована реакція на помилку, наприклад, видача повідомлення на операторську панель.

Для пошуку помилок у програмі призначена також інформація, яка міститься у так званих довідкових даних (рис. 9.17). До таких даних належать структура програми, таблиця перехресних посилань, перелік елементів входу, виходу, пам'яті, а також лічильників та таймерів, які використовуються у програмі.

Структура програми дозволяє знайти такі помилки, як наприклад, рекурсію, тобто ситуацію, коли блок викликає сам себе.

Таблиця перехресних посилань дає інформацію про місце використання змінних у програмі. Типовою помилкою є, наприклад, багаторазовий запис інформації на вихід (дійсним буде тільки останній запис).

З таблиці перехресних посилань можна безпосередньо перейти до програми.

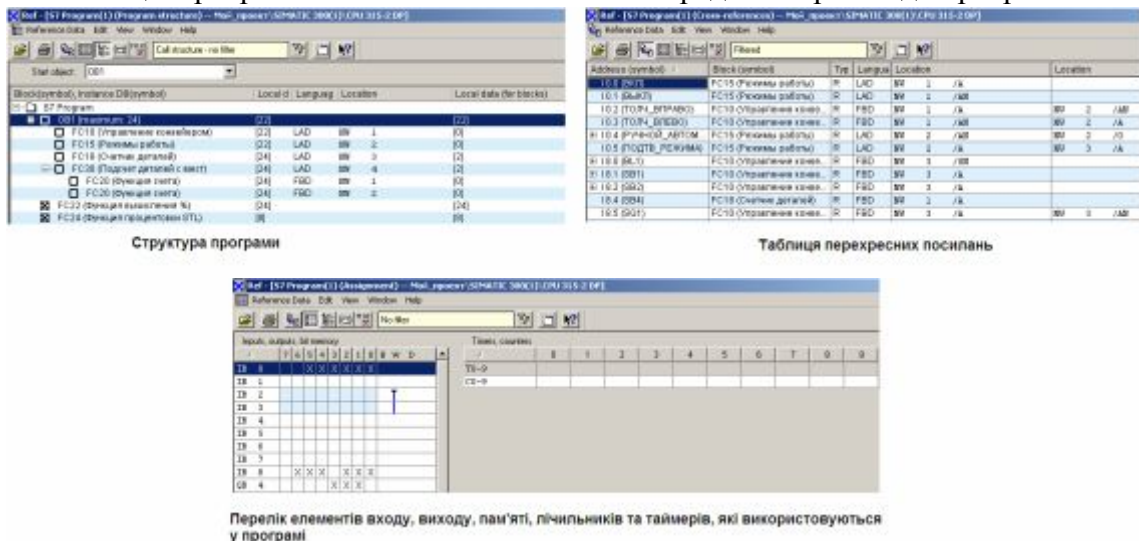


Рис. 9.17. Довідкові дані

Перелік елементів, які використовуються у програмі дає можливість знайти входи та виходи, про які забули при складанні програми, наприклад, входи для підключення блокувань, або виходи для підключення світових індикаторів.

Також можна знайти звернення до пам'яті у різних форматах, наприклад, коли за одною адресою є звертання як до біта, байта, слова чи подвійного слова.

Вказані засоби пошуку помилок дають можливість знайти помилки, на яку реагує операційна система, або помилки які були внесені під час проектування, а також помилки датчиків. Але вони не дозволяють знайти помилки у деяких виконавчих пристроях, наприклад, у тих випадках коли під час виконання програми відсутня можливість визначення сигналу, яких подається на такий пристрій. У такому разі можна використати функцію керування виходами у стані "СТОП". За допомогою таблиці на вихід подаються різні сигнали та визначається реакція виконавчих пристроїв.

Крім цього можуть бути додаткові засоби пошуку помилок, які інтегруються у базову програму.

Контрольні запитання

1. Які вимоги до мов програмування систем автоматизації потребує стандарт ІЕС61131-3?
2. З яких компонент складається програма контролерів?
3. Які змінні використовує програма контролерів?
4. У якій послідовності виконується програма контролерів?
5. Для чого використовуються організаційні блоки ОВ?
6. У яких формах уявляється програма контролерів?
7. Які логічні операції використовує програма контролерів?
8. Які функції пам'яті використовує програма контролерів?
9. Які функції часу та рахунку використовує програма контролерів?
10. Які операції на числами використовує програма контролерів?
11. Які засоби використовують для налагодження програми?
12. Яку інформацію дає буфер діагностики?
13. Яку інформацію дає таблиця перехресних вказівок?
14. Яку інформацію дає перелік елементів входу, виходу, пам'яті, лічильників та таймерів, які використовуються у програмі?
15. Яку інформацію дає структура програми?

Завдання до практичних занять

Задача 1

Скласти програму керування підйомником за допомогою мови LOGO!Soft Comfort (додаток 2).

Підйомник переміщається у двох напрямках: догори та униз. Для прямування нагору необхідно натиснути кнопку "нагору". Для прямування униз необхідно натиснути кнопку "униз". Прямування нагору або униз припиняється при натисненні кнопки "стій" або при досягненні кінцевих вимикачів "верх" або "низ". Передбачити захист від одночасного натискання кнопок догори та униз.

Задача 2

Скласти програму послідовного переміщення за допомогою мови LOGO!Soft Comfort (додаток 2).

Маніпулятор з цикловим керуванням здійснює такі пересування ВВЕРХ, ВНИЗ, ВПРАВО, ВЛІВО, ВПЕРЕД, НАЗАД.

Кожному пересуванню відповідає один вихід на виконавчий пристрій переміщення та один вихід для опитування кінцевого вимикача.

Треба скласти програму пересування, використовуючи функції пам'яті відповідно такій послідовності переміщень:

ВВЕРХ, ВПЕРЕД, ВПРАВО, ВНИЗ, ВВЕРХ, ВЛІВО, НАЗАД, ВНИЗ.

Задача 3

Скласти програму переміщення до визначеної позиції..

Маніпулятор з позиційним керуванням здійснює такі переміщення:
пересування УВЕРХ, УНИЗ, ВПРАВО, ВЛІВО

Кожному пересуванню відповідає один вихід на виконавчий пристрій переміщення та один вихід для опитування датчика пересування.

Позиціонування здійснюється під час руху УВЕРХ, УНИЗ та ВПРАВО, ВЛІВО.

Треба скласти програму пересування, використовуючи функції лічильника відповідно такій послідовності переміщень:

Переміщення ВПРАВО до позиції 4.

Переміщення УВЕРХ до позиції 2.

Глава 10. Структура і склад комплексних систем керування

10.1. Архітектура, структура та склад комплексних систем керування

Останнім часом широке використання знаходять комплексні системи керування. Відмінною особливістю таких систем є повна апаратна та програмна сумісність усього обладнання яке входить до складу загального пристрою керування, включаючи самі пристрої керування, датчики та виконавчі пристрої.

Крім цього у склад загальної системи керування повинні входити засоби відображення технологічного процесу та керування (засоби візуалізації).

Оскільки технологічне обладнання, як, наприклад, у автоматизованих транспортно-складських систем, може буди розподілено на досить великих територіях, треба мати засоби для об'єднання усього цього обладнання в єдину систему керування. Такими засобами є локальні промислові обчислювальні мережі.

Задачами системи проектування для комплексних систем керування є можливість проектування усіх компонент такої системи у межах єдиної програмної платформи.

Такою програмною платформою може бути система проектування програмованих логічних контролерів, базовий варіант якої має як правило обмежені функції, але при використанні додаткових пристроїв вони можуть нарощуватися шляхом інсталяції у базовий варіант додаткових програмних комплексів проектування.

Приклад комплексної системи керування наведений на рис. 10.1. На верхньому рівні системи керування встановлені операторські станції на основі персональних комп'ютерів та операторські панелі. На цьому рівні здійснюється стратегічне керування.

За допомогою локальної мережі Industrial Ethernet (IE) операторські станції здійснюють зв'язок з програмованими логічними контролерами (ПЛК), які знаходяться на середньому рівні системи керування та вирішують задачі тактичного керування, а саме, керування окремими пристроями.

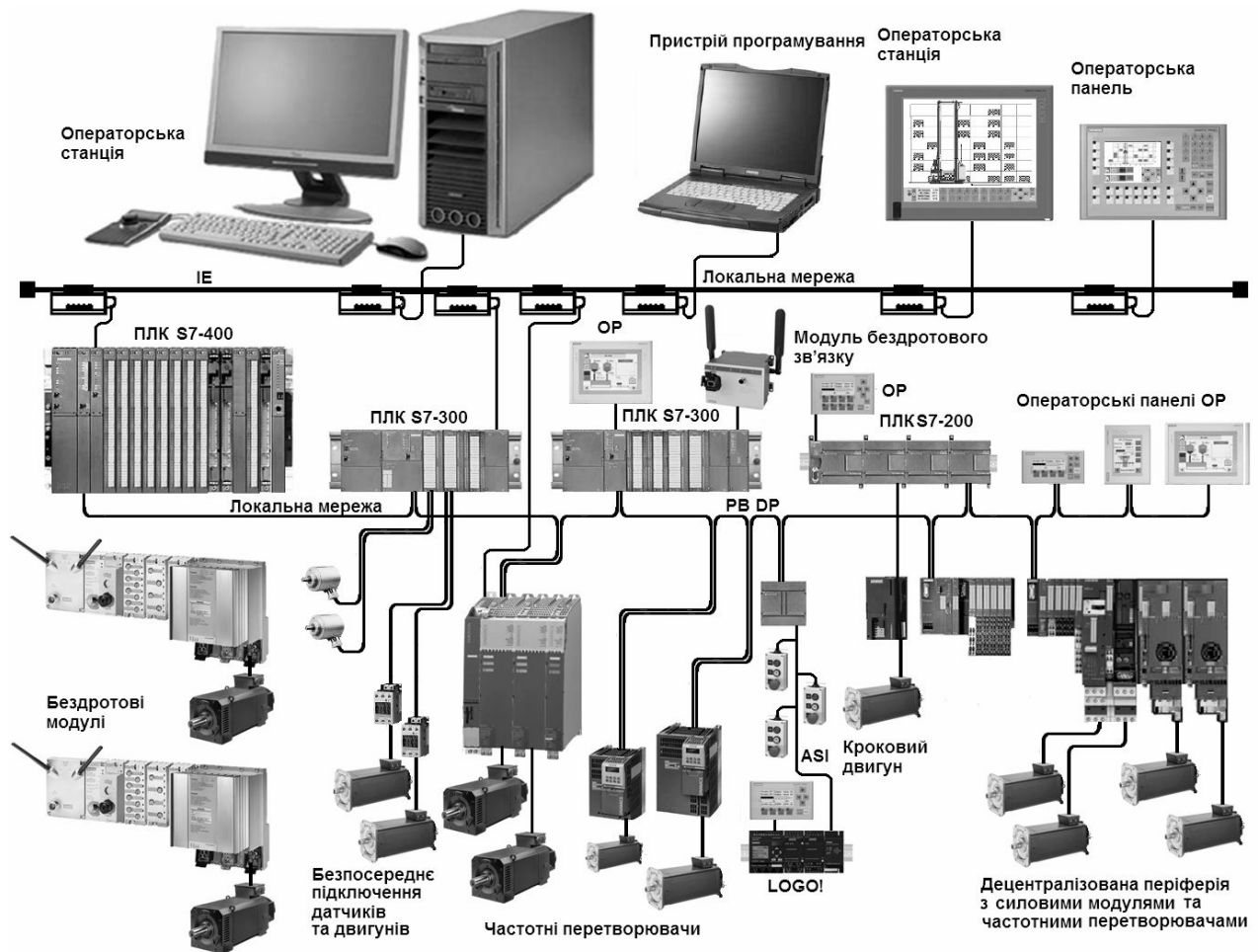


Рис. 10.1. Приклад комплексної системи керування

На цьому рівні реалізується розподілена система керування, яка представляє собою декілька об'єднаних локальних пристроїв керування, що здійснюють безпосереднє опитування датчиків та керування двигунами, або так звану децентралізовану систему керування з використанням децентралізованої периферії. На середньому рівні здійснюється тактичне керування.

На нижньому рівні знаходяться виконавчі пристрої та засоби їх безпосереднього керування, наприклад, електродвигуни змінного струму та частотні перетворювачі для регулювання швидкістю обертання.

Для об'єднання локальних пристроїв керування можна використовувати локальні мережі, які здатні працювати з децентралізованою периферією, наприклад, Profibus, Profinet та Profibus-DP.

Локальні мережі дозволяють здійснити зв'язок між окремими пристроями за допомогою дротовим, оптичних та бездротових ліній зв'язку. Використання бездротового зв'язку особливо важливо для транспортних роботів та при використанні супутникових технологій.

Розподілена система керування складається з декількох систем керування, які об'єднуються у єдину систему керування за допомогою локальних мереж, як це робиться, наприклад, у комплексній системі керування складом (рис. 10.2).

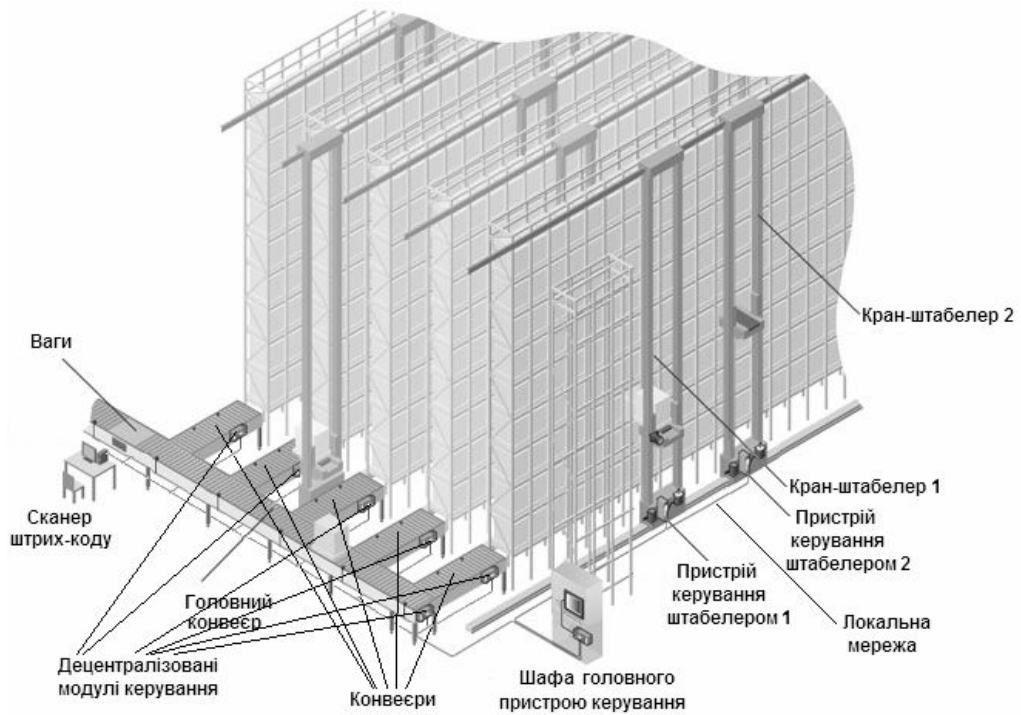


Рис. 10.2. Комплексна система керування складом

На рис. 10.3 наведений приклад апаратних компонент комплексної системи керування складом.

Ця система складається з пристроїв керування штабелерами, децентралізованих модулів керування конвеєрами, головного пристрою керування, до якого підключені децентралізовані модулі керування конвеєрами та інші пристрої керування за допомогою локальних мереж.

Децентралізовані модулі керування дають можливість використовувати керування за допомогою одного контролера на досить великій площі, оскільки підключаються до основного контролера за допомогою локальних мереж..

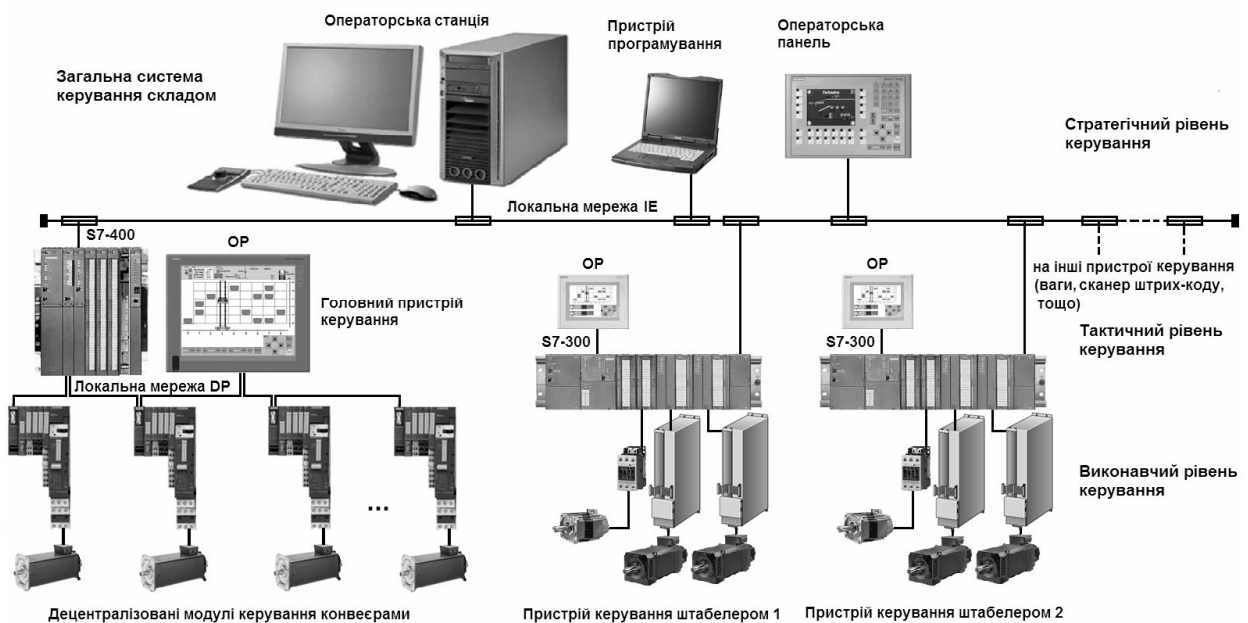


Рис. 10.3. Приклад апаратних компонент комплексної системи керування складом

10.2. Апаратні і програмні компоненти

Розглянемо компоненти комплексної системи керування на прикладі автоматизованої складської системи (рис. 10.4).

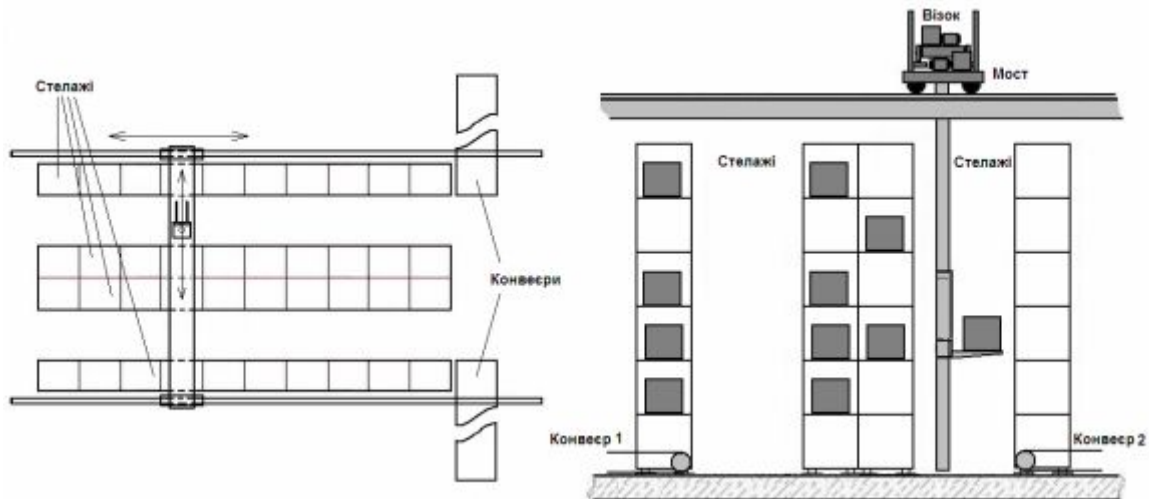


Рис. 10.4 . Автоматизована складська система

Ця система включає мостовий штабелер та два конвеєра. Оскільки усі приводи переміщень і датчики положення знаходяться далеко один від одного, то програмований логічний контролер (ПЛК), встановлений на візку, безпосереднє виконує переміщення по вертикалі з використанням функції позиціонування.

Для керування іншими переміщеннями використовуються модулі децентралізованої периферії (ДП1 – ДП5).

На рис. 10.5 наведена структура децентралізованої периферії для керування такою системою.

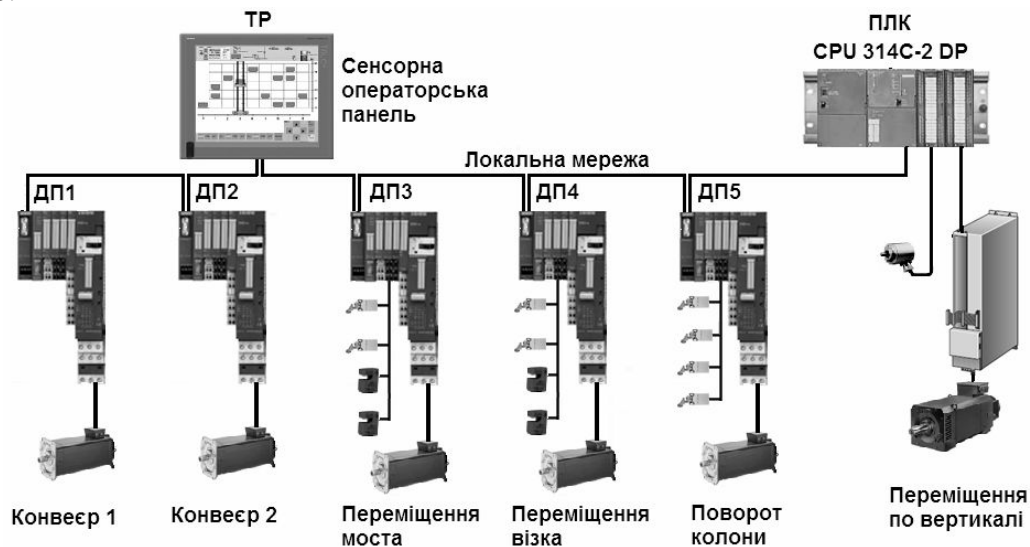


Рис. 10.5 . Приклад використання децентралізованої периферії

Електроприводи здійснюють переміщення моста та візка з позиціонуванням за допомогою щільного індуктивного датчика та поворот колони, а також рух конвеєрів.

Загальне керування виконує ПЛК. За допомогою сенсорної операторської панелі TP оператор задає позицію переміщення та спостерігає за роботою системи.

Програмовані контролери SIMATIC фірми Сименс дозволяють створювати такі розподілені системи автоматичного управління, в яких використовуються станції розподіленого вводу-виводу SIMATIC ET 200. Ці структури управління дозволяють

забезпечити високу гнучкість, ефективність, простоту і зручність обслуговування систем автоматизації.

У разі потреби змінювати швидкість переміщення до контролера можна здійснювати керування декількома частотними перетворювачами.

Налагодження таких пристроїв також здійснюється за допомогою засобу HW Config (рис. 10.8).

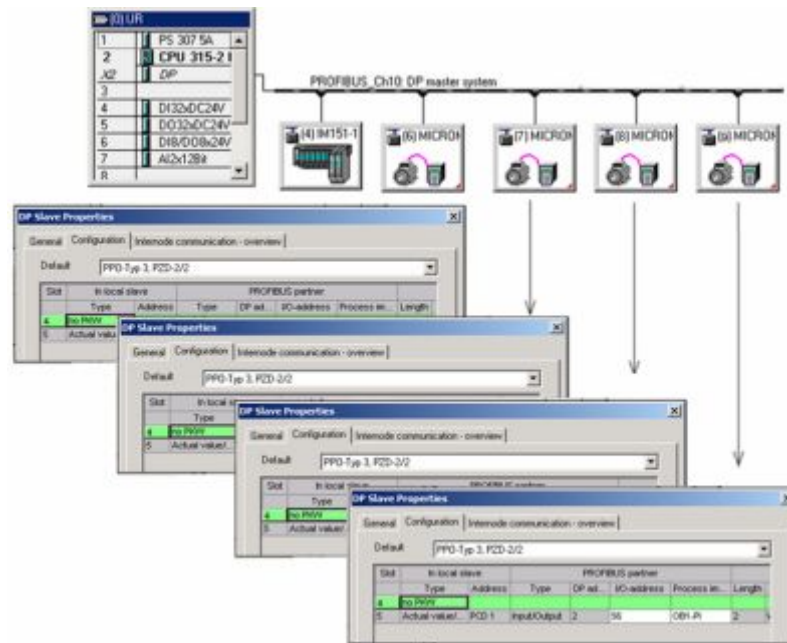


Рис. 10.8. Проектування та налагодження частотних перетворювачів

Для налагодження та керування виконавчими пристроями існують програмні інструменти комп'ютерного налагодження та керування, наприклад, програма STARTER для електроприводів фірми SIEMENS, яка інтегрується в пакет STEP7 та дає такі можливості для налагодження приводів.

За допомогою Project > New with Wizard створюється новий проект, а також встановлюється і перевіряється інтерфейс для зв'язку. Знаходяться наявні пристрої та вводяться в проект (рис. 10.8).



Рис. 10.8. Створення проекту

Здійснюється встановлення конфігурації приводу. Компоненти вибираються з каталогів.

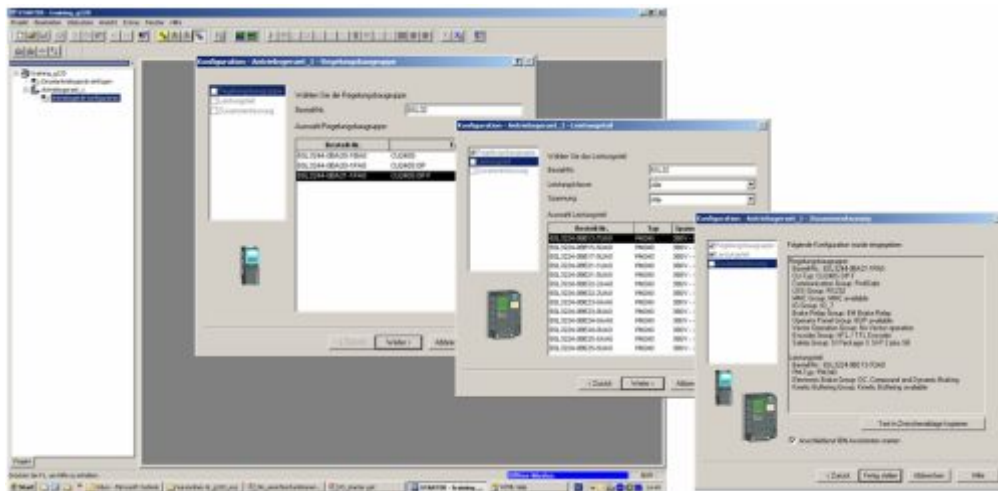


Рис. 10.9. Встановлення конфігурації приводу

Встановлення параметрів перетворювача може проводитися за допомогою масок параметрів або переліку параметрів (рис. 10.10).

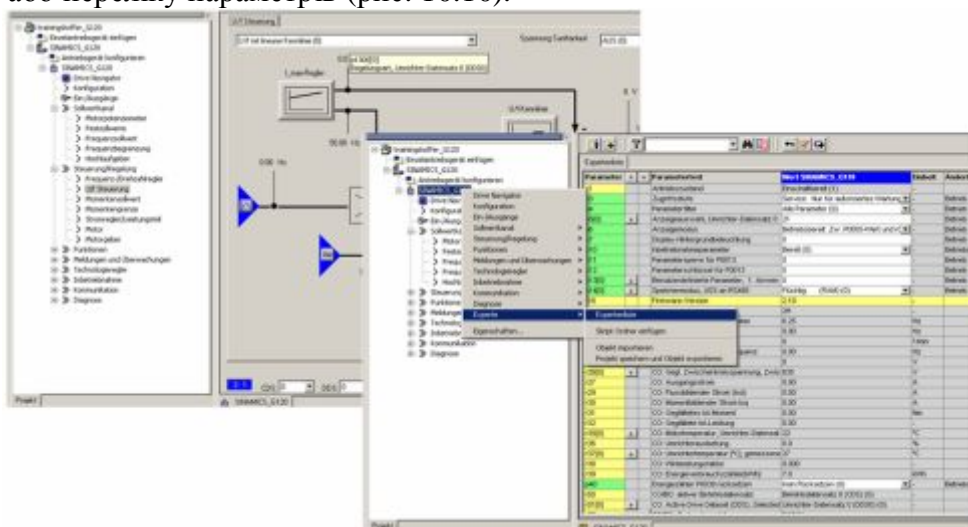


Рис. 10.10. Встановлення параметрів

Панель керування служить для управління і перегляду окремих приводів (рис. 10.11). За допомогою панелі керування можна управляти приводом. Завдяки цьому оператор, який здійснює введення приводу в дію може окремо протестувати кожен частину приводу, перш ніж почне працювати привід, що координується програмою, а в разі помилки протестувати, чи можна взагалі управляти окремим приводом з панелі управління або тут є проблеми.

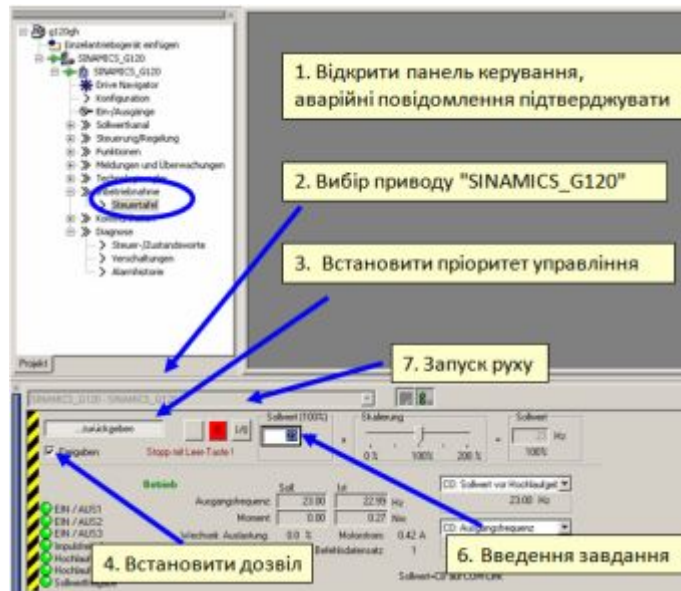


Рис. 10.11. Панель керування приводом

10.3. Засоби автоматизації на основі пристроїв керування, сумісних з персональними комп'ютерами

Поширене вживання персональних комп'ютерів розповсюджується також на комплексні системи керування, що використовуються у промисловості.

При цьому є два основних напрямків такого використання. Перше полягає в тому, що створюють персональні комп'ютери, що можуть працювати у складних виробничих умовах, це так звані промислові комп'ютери. Друге направлення - це використання контролерів, структура та програмне забезпечення яких сумісне з персональними комп'ютерами.

Промислові комп'ютери здатні виконувати функції незалежних робочих станцій, а також функції клієнтів та серверів. Вони відрізняються тим, що мають захист від шкідливих умов виробництва (рис. 10.12), та можуть бути встановлені у стандартні стойки та шафи керування (рис. 10.13). У разі потреби їх можна доповнювати додатковими компонентами.



Рис. 10.12. Промислові комп'ютери



Рис. 10.13. Промислові комп'ютери для встановлення у стандартні стойки керування (шафи 19")

Панельні промислові комп'ютери (рис. 12.14) також призначені для використання у стандартних шафах, панелях та щитах управління.

У разі потреби промислові комп'ютери доповнюються необхідним обладнанням, наприклад, клавіатурами з високою ступенем захисту та можливістю встановлення у шафу (рис. 12.15).



Рис. 12.14. Панельні промислові комп'ютери



Рис. 12.15. Клавіатури з високим ступенем захисту

Перші програмовані контролери з'явилися раніше персональні комп'ютерів, а оскільки оновлення таких систем значно повільніше, ніж оновлення офісного обладнання, до якого належать персональні комп'ютери, то вони ще досі мають структуру та програмне забезпечення, що значно відрізняється від персональних комп'ютерів. Але в останній час з'явилося досить велика кількість систем керування, сумісних з персональними комп'ютерами. Розглянемо деякі з них.

РС/104 — шина розширення (яка також дала назву формату плат), створена на базі 8-розрядного варіанта шини ISA, що використовувалась у персональних комп'ютерах. Відмінною особливістю механічного конструктиву РС/104 є розташування контактів не на ребрі плати, а перпендикулярно їй, що дозволяє встановлювати плати один на одну. Така конструкція дозволяє зібрати до 3-6 плат в один великий «сандвіч» і розмістити його в компактному герметичному корпусі, який матиме велику ударостійкість.

З появою шини PCI з'явився новий стандарт PC/104+ (PC/104Plus), в який додана шина PCI.

Плати PC/104 і PC/104+ сумісні між собою.

Роз'єм для плат PC/104 і PC/104+ досить часто встановлюють на бічну поверхню плат MicroPC та інших плат, призначених для промислових комп'ютерів, вбудованих систем, телекомунікації, цифрових приладів в якості мезонінної плати (плати, що дозволяє значно розширити можливості основного пристрою).

На рис. 12.16 наведений «Сендвіч» з одноплатного комп'ютера і мережевої плати у форматі PC/104 та одноплатний комп'ютер на базі шини PCI-104

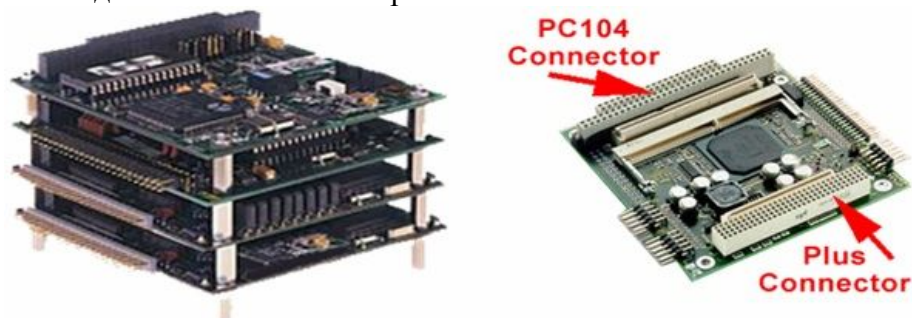


Рис. 12.16. «Сендвіч» з одноплатного комп'ютера і мережевої плати

Для побудови недорогих систем управління можна використовувати PC-сумісні контролери та модулі розподіленого вводу / виводу від компанії ICP DAS (рис. 12.17).

Основні характеристики:

- робочий температурний діапазон: від -25 до + 70 ° C;
- підтримка RS-232/485, Modbus TCP, Modbus RTU, DCON, Ethernet, CANOpen, DeviceNET;
- розвинена система програмування;
- вбудовані функції діагностики;
- великий вибір модулів введення / виведення



Рис. 16.17. Контролери ICP DAS

За допомогою модулів віддаленого вводу / виводу можна зібрати велику кількість сигналів з різноманітних об'єктів, передати їх в контролер, обробити і видати керуючі сигнали на виконавчі механізми і пристрої індикації.

Перевагами PC-сумісних контролерів ICP DAS є:

- невисока ціна апаратних засобів;
- використання відкритих протоколів, яке дозволяє інтегрувати в одну систему пристрої широкого спектра виробників;
- простота програмування і доступність широкого спектру програмного забезпечення, що мінімізує витрати часу і коштів на створення системи;

- простота інтеграції з системами управління більш високого рівня, що дозволяє спростити доступ до даних технологічних процесів з боку систем управління підприємством.

Контрольні запитання

1. З яких компонент складаються комплексні системи керування?
2. На які рівні можна поділити комплексні системи керування?
3. Для чого використовуються локальні мережі?
4. З яких апаратних компонент складається комплексна система керування складом?
5. Для чого використовується децентралізована периферія?
6. Для чого використовується програма STARTER?
7. Чим відрізняються промислові комп'ютери?
8. Для чого використовуються панельні промислові комп'ютери?
9. Які особливості має система керування PC/104?
10. Які особливості має система керування ICP DAS?

Глава 11. Обчислювальні мережі у системах керування. Локальні та глобальні обчислювальні мережі

11.1. Види мереж. Поняття інтерфейсу

Комп'ютерна мережа — система зв'язку між двома чи більше комп'ютерами. У ширшому розумінні комп'ютерна мережа (КМ) — це система зв'язку через кабельне чи повітряне середовище, яка включає самі комп'ютери різного функціонального призначення і мережеве обладнання.

Для передачі інформації можуть бути використані різні фізичні явища, як правило — різні види електричних сигналів чи електромагнітного випромінювання.

Середовищами передавання у комп'ютерних мережах можуть бути телефонні кабелі, та спеціальні мережеві кабелі: коаксіальні кабелі, виті пари, волоконно-оптичні кабелі, радіохвилі, світлові сигнали.

Існуючі на сьогодні мережі досить відрізняються за принципами взаємодії, характеристикам і рівнем запропонованих послуг.

За ступенем охоплення території і користувачів можна здійснити наступну класифікацію: глобальні мережі спільного використання, регіональні мережі і локальні обчислювальні мережі.

Глобальні мережі спільного користування виставляють наступні вимоги:

- наявність могутньої і гнучкої системи адресації, що дозволяє визначити велику кількість користувачів;
- висока ефективність передачі корисної інформації;
- використання складних протоколів маршрутизації.

Регіональні мережі призначені для об'єднання групи користувачів, наприклад жителів деякого міста і можуть носити глобальний або локальний характер.

Локальні обчислювальні мережі призначені для організації взаємодії обмеженої групи користувачів, що використовують спеціальні протоколи з спрощеним механізмом маршрутизації, адміністрування, передачі даних.

Для обміну даними в мережах використовують різні інтерфейси.

Інтерфейс являє собою систему зв'язку між різними вузлами і блоками складного обладнання, а також між технікою і користувачем. Він виражається в логічній (системи подання інформації) і фізичній (характеристики інформаційних сигналів) формі.

Таким чином інтерфейс це сукупність засобів і правил, що забезпечують взаємодію пристроїв обчислювальної машини або системи обробки інформації і (або) програм.

11.2. Локальні мережі для систем керування

Локальна мережа (Local Area Network, LAN)- це комп'ютерна мережа, географічно сконцентрована в одній будівлі або в комплексі будівель. Ознаками, що виділяють локальну мережу від, наприклад, великої мережі (WAN) – це значно вища швидкість передачі даних, невелика територія та відсутність потреби у виділених телекомунікаційних лініях.

Широка і постійно зростаюча номенклатура локальних обчислювальних мереж, мережні програмні продукти і технології покладають на потенційного користувача складну задачу вибору потрібної системи з великої кількості існуючих.

Сьогодні в світі нараховується десятки тисяч різних локальних обчислювальних мереж і для їх розгляду корисно мати систему класифікації, для якої треба визначити певні ознаки.

За призначенню локальні обчислювальні мережі можна розділити на: керуючі (організаційними, технологічними, адміністративними та іншими процесами), інформаційні (інформаційно-пошукові), розрахункові, інформаційно-розрахункові, обробки документальної інформації і так далі.

За організацією управління однорідні локальні обчислювальні мережі в залежності від наявності (або відсутності) центральної абонентської системи діляться на дві групи.

До першої групи відносяться мережі з централізованим управлінням.

До другої групи відносяться мережі з децентралізованим або розподіленим управлінням.

По формуванню передачі інформації локальні мережі поділяються на мережі з маршрутизацією інформації і селекцією інформації.

Взаємодія абонентських систем з маршрутизацією інформації забезпечується визначенням шляхів передачі блоків даних по адресах їх призначення. При цьому абонентські системи можуть взаємодіяти по різних шляхах (маршрутах) передачі блоків даних, а для скорочення часу передачі здійснюється пошук найкоротшого по часу маршруту.

В мережах з селекцією інформації взаємодія абонентських систем проводиться вибором (селекцією) адресованих їм блоків даних. При цьому всім абонентським системам доступні всі блоки даних, що передаються в мережі.

По топологічних ознаках локальні мережі поділяються на мережі з магістральною, кільцевою, зіркоподібною конфігурацією та інші (рис. 11.1).

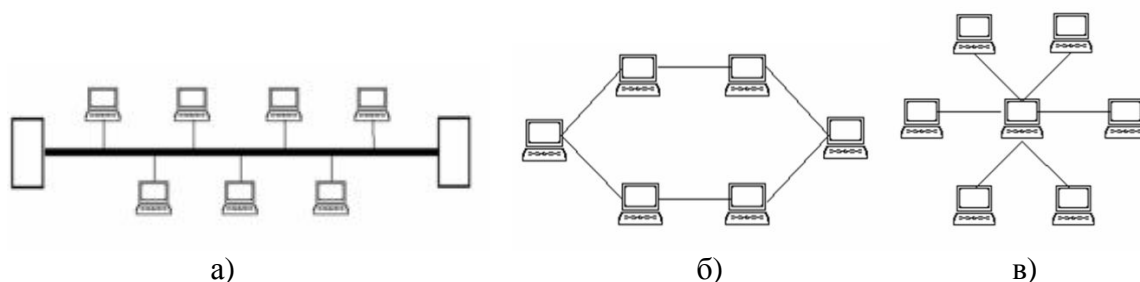


Рис. 11.1. Топології локальних мереж – магістральна(а), кільцева (б), зіркоподібна (в)

Магістральна (шинна) топологія реалізується у вигляді пасивного моноканалу (магістралі). Ця топологія найпоширеніша. Вона використовується у випадку, коли інформація передається рідко (в порівнянні з можливостями комп'ютерів), дані комплектуються в пакет, дістають адресу і після того, як магістраль стане доступною, відбувається передача повідомлення.

Кільцева топологія передбачає послідовне з'єднання абонентів з каналами передачі даних, внаслідок чого утворюється замкнуте кільце. Кожен абонент відіграє роль ретранслятора повідомлення з невеликою часовою затримкою.

Зіркоподібна топологія передбачає з'єднання каналів приєднаних до різних абонентів в одній точці, яка називається центральним вузлом.

Локальна мережа для систем керування або промислова мережа (англ. fieldbus) — це цифрова, двонаправлена, багатоточкова послідовна мережа, що зв'язує територіально розподілені датчики, виконавчі механізми, промислові контролери і використовується в промисловій автоматизації для побудови єдиного інформаційного і керуючого середовища, що об'єднує інтелектуальні технологічні пристрої і контролери цехового рівня.

Описується стандартом IEC 61158. Термін польова шина є дослівним перекладом англійського терміну fieldbus, термін промислова мережа є точнішим перекладом і саме він використовується в професійній технічній літературі.

Своїм сімейством промислових мереж SIMATIC NET фірма Siemens представляє відкриту гетерогенну комунікаційну систему, призначену для використання на всіх рівнях ієрархії систем автоматизованого управління в умовах промислового виробництва. Комунікаційні системи SIMATIC NET базуються на державних і міжнародних стандартах у відповідності з 7-рівневою моделлю ISO / OSI.

Основою таких комунікаційних систем служать локальні обчислювальні мережі (ЛОМ), які можуть бути реалізовані як: електричні, оптичні, бездротові, комбінація електричних / оптичних / бездротових, електричні іскробезпечні.

Фірма Сименс використовує різні промислові мережі, а саме Industrial Ethernet, PROFIBUS та PROFINET які є складовими частинами SIMATIC NET. Проектування мереж здійснюється за допомогою засобу NETPRO, який уходить до складу програмного пакету STEP7.

Приклад структури мережі з використанням компонентів PROFIBUS наведений на рис. 11.2.

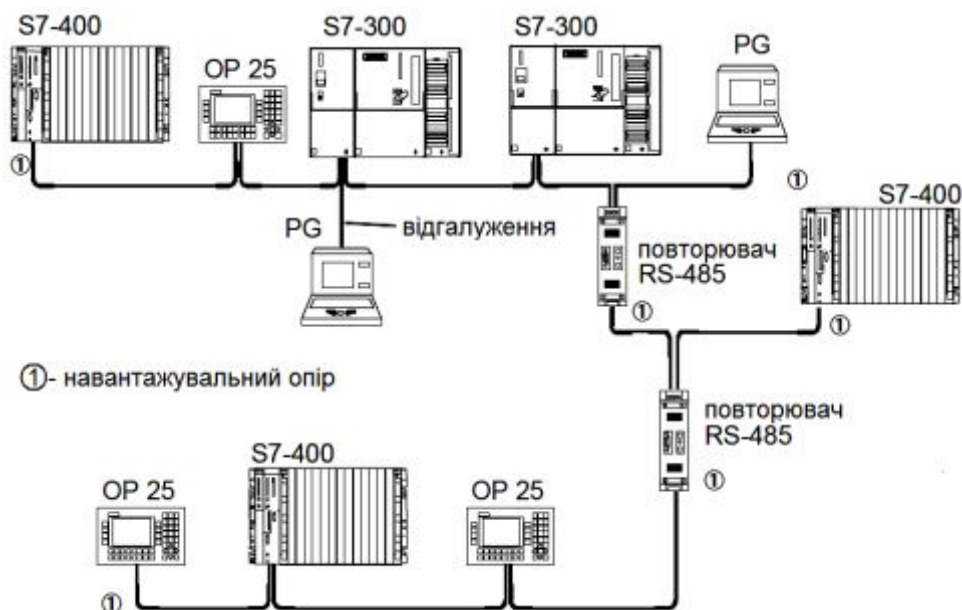


Рис. 11.2. Структура мережі з використанням компонентів PROFIBUS

Далі наведені основні характеристики мережі PROFIBUS.

Топологія мережі: шинна топологія, деревоподібна структура з використанням повторювачів.

Середя передачі: екранована вита пара.

Можливі довжини сегментів:

1000 м для швидкості передачі до 187.5 кбіт / с,

400 м для швидкості передачі до 500 кбіт / с,

200 м для швидкості передачі до 1.5 Мбіт / с,

100 м для швидкостей передачі 3.6 і 12 Мбіт / с,

Кількість послідовно підключених повторювачів: не більше 9.

Кількість вузлів: не більше 32 на одному шинному сегменті, не більше 127 в мережі при використанні повторювачів.

Швидкості передачі: 9.6 кбіт / с, 19.2 кбіт / с, 45.45 кбіт / с, 93.75 кбіт / с, 187.5 кбіт / с, 500 Кбіт / с, 1.5 Мбіт / с, 3 Мбіт / с, 6 Мбіт / с, 12 Мбіт / с.

Однією з найважливіших промислових комунікаційних мереж в області SIMATIC NET є Industrial Ethernet (IE). Industrial Ethernet визначається як комунікаційна мережа в техніці базової передачі даних відповідно до міжнародного стандарту IEEE 802.3 до процедури доступу CSMA / CD (Carrier Sense with collision detection).

Об'єднання в мережу при цьому здійснюється при швидкості передачі 10/100 MBit / s за допомогою оптоволоконної або екранованої 2-провідної лінії (Industrial Twisted Pair ITP). Industrial Ethernet являє собою мережу для керуючого і мережевого рівнів у відкритій незалежній від виробника системі комунікацій SIMATIC.

Industrial Ethernet пристосований для швидкої передачі великих масивів даних і надає можливість використання шлюзів (Gateway) і маршрутизаторів (Router) для об'єднання в мережу, що виходить за рамки стандартів.

Приклад структури мережі з використанням компонентів Industrial Ethernet наведений на рис. 11.3.

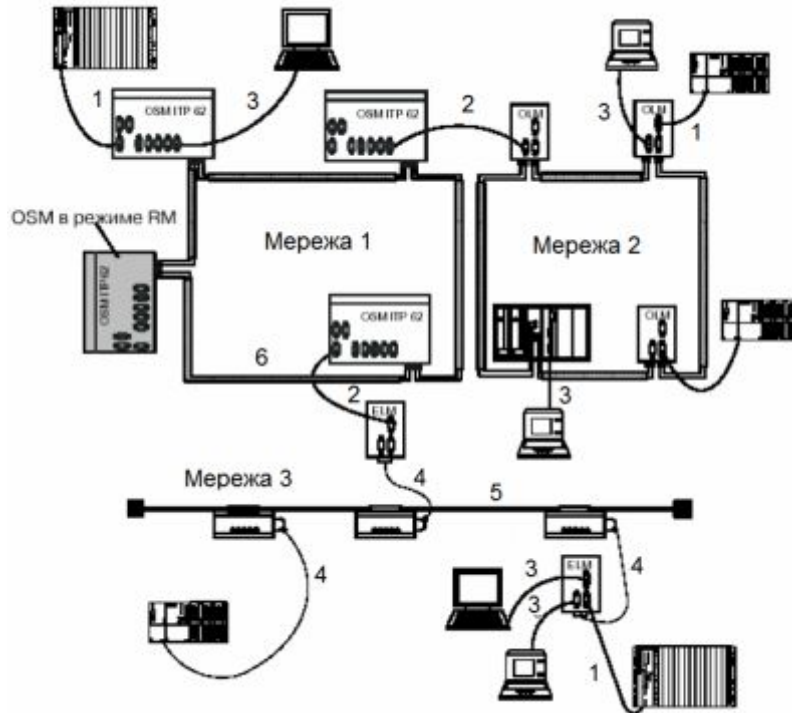


Рис. 11.3. Структура мережі з використанням компонентів Industrial Ethernet

Ця структура включає такі компоненти: 1 – стандартний кабель, 2 – з'єднувальний кабель (TP XP корд), 3 – кабель для підключення окремих пристроїв (TP корд 9/RJ-45), 4 – з'єднувальний кабель 727-1, 5 – триаксіальний кабель, 6 – опто-волоконний кабель.

PROFINET є інноваційним стандартом організацій користувачів PROFIBUS для реалізації загальних і наскрізних рішень автоматизації на основі Industrial Ethernet.

Приклад структури мережі з використанням компонентів PROFINET наведений на рис. 11.4.

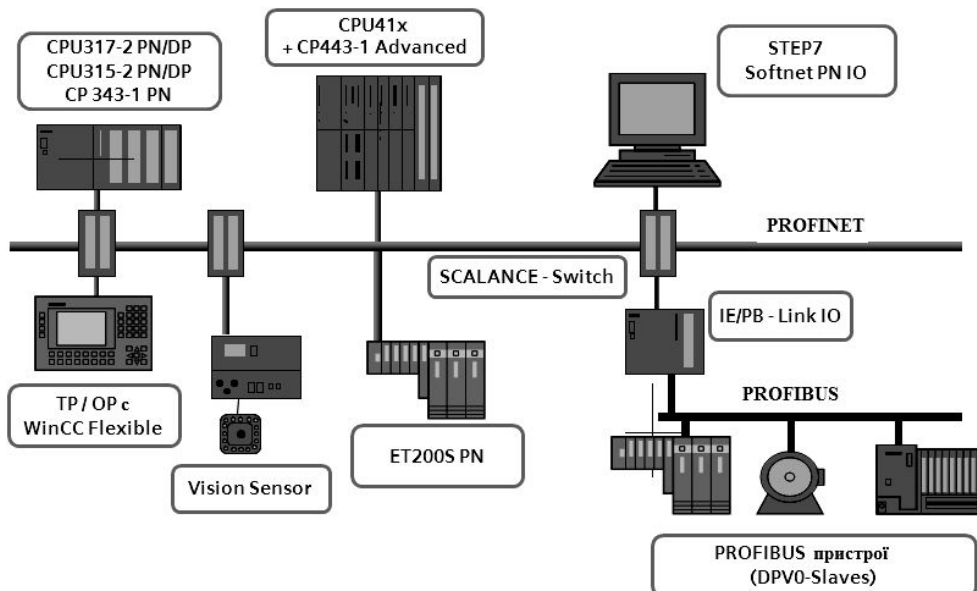


Рис. 11.4. Структура мережі з використанням компонентів PROFINET

За допомогою PROFINET прості децентралізовані польові прилади, а також критичні в часі застосування можна також зв'язати в Ethernet - комунікаціях, як розподілені системи автоматизації на основі компонент через підхід Component based Automation (CBA).

Основними характеристиками PROFINET є:

- відкритий стандарт для розподіленої автоматизації (PNO),

- наскрізна комунікація через польову шину і Ethernet,
- використання відкритих стандартів як TCP / IP, UDP і IT – стандарти,
- Real - Time - Ethernet - комунікації як базова функціональність,
- зв'язок з приводами (Motion Control) через IRT (Isochronous Real Time),
- Інтеграція існуючих інсталяцій PROFIBUS.

PROFINET повністю перекриває вимоги техніки автоматизації. PROFINET об'єднує досвід PROFIBUS і Industrial Ethernet.

Використання відкритого стандарту, простота використання та інтеграція існуючих компонент з самого початку визначено в понятті PROFINET. PROFINET вже інтегрований в міжнародний стандарт IEC 61158.

11.3. Децентралізована периферія

Компоненти SIMATIC дозволяють створювати системи децентралізованої периферії на основі розподіленого вводу-виводу з використанням локальних промислових мереж PROFINET і PROFIBUS. Для цих мереж випускається досить широка гамма різного периферійного обладнання (рис. 11.5).

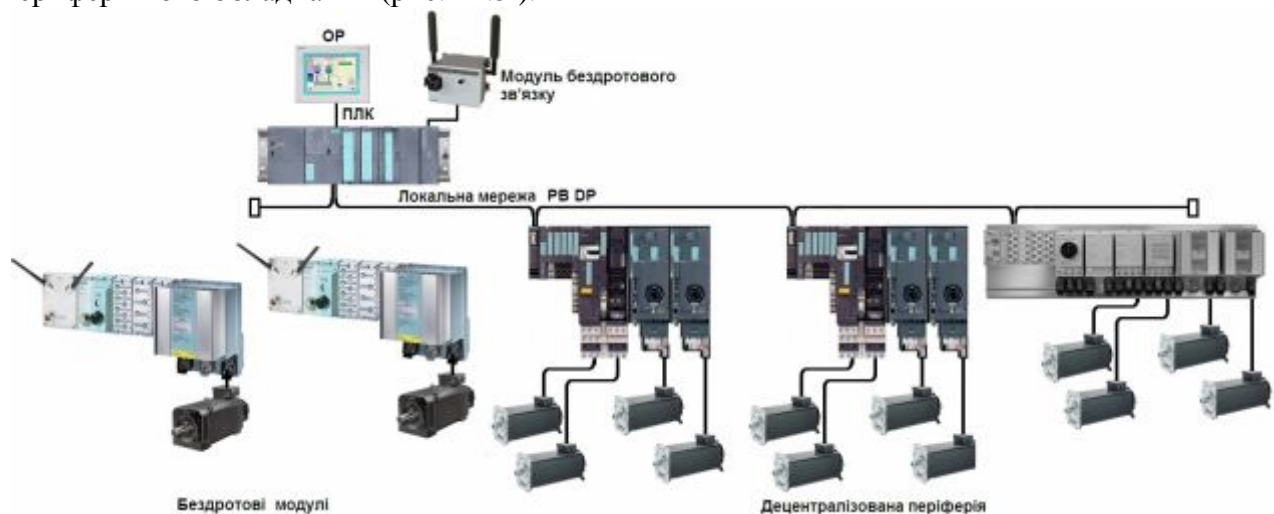


Рис. 11.5. Децентралізована периферія

В програмованих контролерах SIMATIC не існує відмінностей між системами локального і розподіленого вводу-виводу. Для конфігурування, налаштування параметрів, діагностики, обслуговування та підготовки технічної документації по системам локального і розподіленого вводу-виводу використовується один і той же програмне забезпечення.

Всі операції можуть виконуватися дистанційно з одного програматора, підключеного до будь-якої точки мережі PROFIBUS, PROFINET або Industrial Ethernet. Для звернення до входів та виходів систем локального і розподіленого вводу-виводу в програмі користувача використовуються одні й ті ж методи адресації.

Існують системи розподіленого вводу-виводу для установки в шафи управління і для установки поза шаф управління.

Прикладом системи розподіленого вводу-виводу для установки в шафи управління є станції ET200S, які використовуються у складі програмованих контролерів SIMATIC S7-300/S7-400/WinAC, що базуються на промислових мережах PROFIBUS DP або PROFINET IO.

Великий спектр модулів різного призначення дозволяє оптимально адаптувати станції ET200S до вимог різних задач керування складським обладнанням.

Інтерфейсні модулі забезпечують можливість безпосереднього підключення станції до електричних або оптичних каналів зв'язку PROFIBUS DP, а також електричним каналам зв'язку Industrial Ethernet, виконувати попередню обробку даних на рівні станції, забезпечувати підтримку профілю PROFISafe в розподілених системах автоматики безпеки.

При роботі в системах розподіленого вводу-виводу програмованих контролерів S7-400 забезпечується підтримка функцій "гарячої" заміни як електронних, так і силових модулів SIMATIC ET200S може включати в свій склад:

- інтерфейсний модуль IM 151 для підключення станції до мережі PROFIBUS DP або PROFINET IO і підтримки обміну даними з провідним мережевим пристроєм;
- електронні модулі вводу-виводу дискретних і аналогових сигналів;
- технологічні модулі для вирішення завдань позиціонування, зважування, швидкісного рахунку, обміну даними через послідовні канали зв'язку і т.д;
- фідери навантаження, призначені для комутації трифазних ланцюгів змінного струму з навантаженням до 7.5 кВт (наприклад, для підключення двигунів);
- перетворювачі частоти потужністю до 4 кВт.

У загальній складності одна станція ET200S дозволяє встановлювати до 63 модулів різного призначення і дозволяє обслуговувати до 128 дискретних або до 64 аналогових каналів вводу-виводу. Обмін даними з провідним пристроєм в мережі PROFIBUS DP виконується зі швидкістю до 12 Мбіт / с, в мережі PROFINET IO - зі швидкістю 10/100 Мбіт / с.

Усі інтерфейсні модулі можна розбити на пасивні та активні.

Пасивні інтерфейсні модулі призначені для підключення ET 200S до мереж PROFIBUS DP та PROFINET IO і підтримки обміну даними з ведучим DP пристроєм.

Активні або інтелектуальні інтерфейсні модулі IM 151-7 CPU підтримують всі функції пасивних інтерфейсних модулів і здатні виконувати попередню обробку даних на рівні станції. Ці модулі оснащені вбудованим центральним процесором, аналогічним за своїми характеристиками CPU 314.

Прикладом системи розподіленого вводу-виводу для установки поза шаф управління є станція ET200pro.

ET200pro призначена для побудови систем розподіленого вводу-виводу на основі PROFINET IO і PROFIBUS DP, має ступінь захисту IP65/IP66/IP67 і може монтуватися на кероване обладнання без використання шаф управління. Ці станції також мають активні або інтелектуальні інтерфейсні модулі.

У складі станції допускається використовувати електронні модулі введення і виведення дискретних і аналогових сигналів, силові модулі, модулі систем ідентифікації об'єктів MOBY, а також модулі систем автоматики безпеки і протиаварійного захисту PROFIsafe.

Силові модулі ET 200pro FC виконують функції перетворювачів частоти і здатні керувати роботою 3-фазних асинхронних електродвигунів потужністю до 1.1 кВт.

Станція має високу стійкість до механічних впливів і здатна зберігати працездатність при вібраційних навантаженнях з прискоренням до 5g, а також ударних навантаженнях з прискоренням до 25g.

Зовнішній вигляд системи розподіленого вводу-виводу ET200S та ET200pro наведений на рис. 11.6.

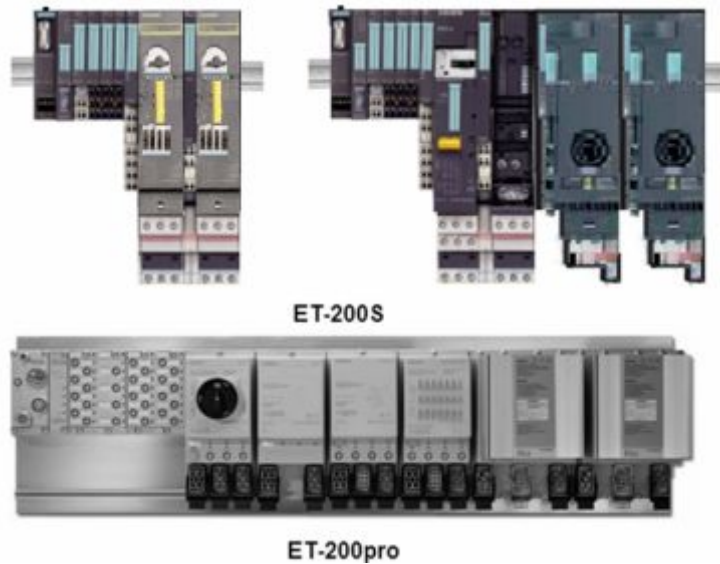


Рис. 11.6. Зовнішній вигляд системи розподіленого вводу-виводу ET200S та ET200pro

У разі використання у складі комплексних систем керування мобільних пристроїв використовуються розподілені системи керування з бездротовим зв'язком, наприклад, на основі інтерфейсного модуля IM 154-6 PN IWLAN системи розподіленого вводу-виводу ET200pro, який підтримує обмін даними між станцією ET 200pro і контролером PROFINET IO через IWLAN (рис. 11.7).



Рис. 11.7. Приклад розподіленої системи керування з використанням бездротового зв'язку

На рис. 11.8 наведений приклад використання бездротового зв'язку для керування візком автоматизованої транспортно-складської системи на основі супутникових технологій.

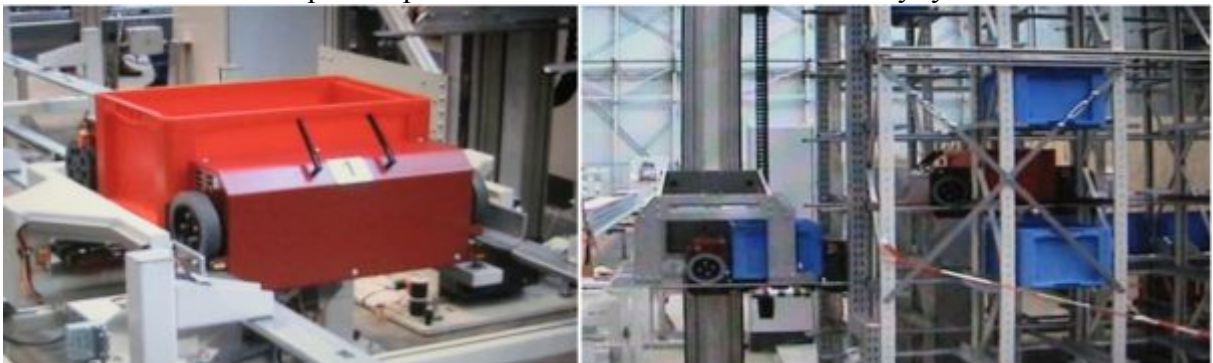


Рис. 11.8. Використання бездротового зв'язку для керування візком автоматизованої транспортно-складської системи на основі супутникових технологій

Проектування систем розподіленого вводу-виводу здійснюється за допомогою засобу HW Config, який входить у склад програмного пакету STEP7 (рис. 11.9).

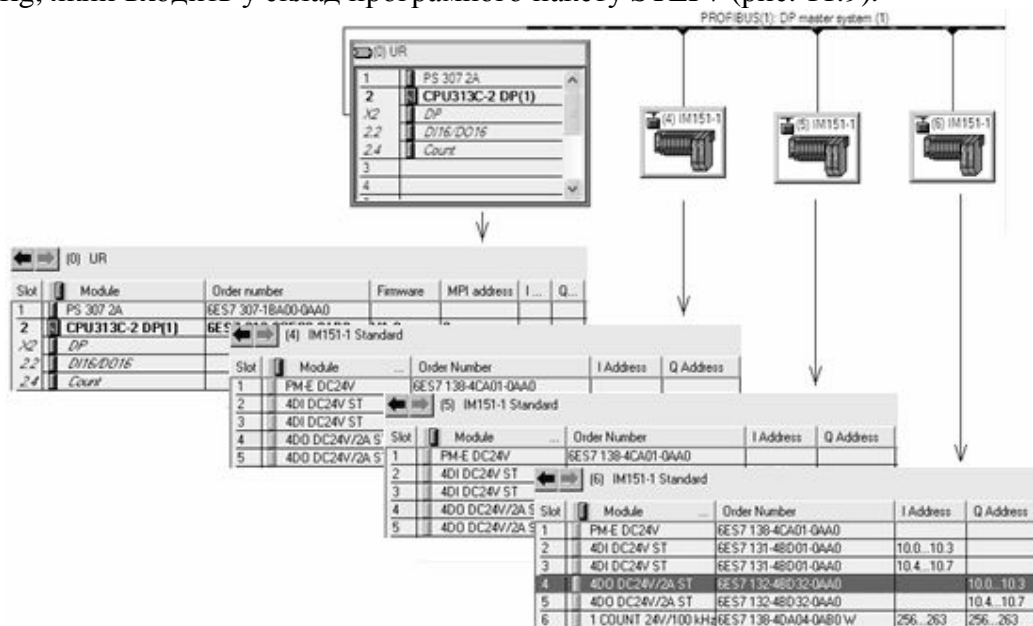


Рис. 11.9. Проектування систем розподіленого вводу-виводу здійснюється за допомогою засобу HW Config.

11.4. Регіональні і глобальні мережі

Регіональні мережі (Metropolitan Area Network, MAN) – великі комп'ютерні мережі, що зазвичай покривають місто.

Вони будуються на бездротовій інфраструктурі, або за допомогою оптоволоконних кабелів. MAN більш оптимізована до великих відстаней чим LAN.

Регіональна мережа може належати і підтримуватися однією організацією, але зазвичай використовується багатьма користувачами і організаціями.

Для цієї мережі використовувалися технології: ATM, FDDI and SMDS, але на сьогодні вони замінюються основаними на Ethernet технологіями (Metro Ethernet).

Глобальна мережа (Wide Area Network, WAN) охоплює значну географічну область, часто цілу країну або, навіть, континент. Найбільшою і найвідомішою глобальною мережею є Інтернет.

WAN використовується для з'єднання локальних мереж, які з'єднуються комунікаційними підмережами.

Завданням підмережі є передача повідомлень від однієї локальної сітки до іншої.

WAN часто будуються за допомогою виділених ліній, при кожному з кінців якої роутер (маршрутизатор) приєднується до локальної мережі з одного боку і до хабу в WAN з іншого. Проте виділені лінії дуже дорогі і мережі будуються за допомогою більш дешевих технологій комутацій каналів і комутацій пакетів.

Контрольні запитання

1. Для чого використовують промислові мережі?
2. Як поділяють мережі за ступенем охоплення території і користувачів?
3. Наведіть приклади типів локальних мереж, що поділяються по топологічних ознаках.
4. Які компоненти систем автоматизації об'єднують локальні мережі?
5. Наведіть приклади промислових локальних мереж.

6. Для чого використовують глобальні мережі?
7. Для чого використовується децентралізована периферія?
8. Як здійснюється налагодження децентралізованої периферії?
9. Для чого використовують бездротові системи зв'язку?
10. Наведіть приклади бездротових систем зв'язку.

Глава 12. Системи відображення технологічних процесів і керування

12.1. Системи відображення технологічних процесів і керування

Програмовані логічні контролери не мають вмонтованих систем відображення технологічного процесу та засобів введення даних. Цю задачу виконує так званий людино-машинний інтерфейс (НМІ) за допомогою операторських панелей та операторських станцій, які призначені для відображення текстової або графічної інформації.

Значне місце в таких системах займають комп'ютерні пристрої відображення інформації і керування, за допомогою яких оператор одержує інформацію про хід технологічного процесу і вводить необхідну додаткову інформацію або команди для керування системою. Наявність широкого спектра систем візуалізації й обслуговування, зокрема символічних та графічних операторських панелей ОП та промислових персональних комп'ютерів, забезпечує ефективну взаємодію систем керування та оператора.

Для розробки комп'ютерних пристроїв відображення інформації і керування використовується додаткове програмне забезпечення, за допомогою якого створюється ціла система зображень та повідомлень, що дають повне зображення керованого процесу або пристрою та інформацію о значеннях різних параметрів у числовому або графічному вигляді. Крім того створюються різні графічні засоби введення даних. У разі використання персональних комп'ютерів програмне забезпечення складається з системи проектування та системи виконання, за допомогою якої персональний комп'ютер перетворюється на операторську станцію.

12.2. Операторські панелі та операторські станції

Операторські панелі представляють собою спеціалізовані обчислювальні пристрої з засобами відображення та введення інформації (рис. 12.1). Для відображення інформації використовують символічні та графічні дисплеї. Введення інформації здійснюється за допомогою клавіатури або безпосереднє з екрану за допомогою маніпуляторів, наприклад, джойстиків або мишки. Замість клавіатури може використовувати сенсорні екрани. У цьому разі усі елементи керування зображуються на екрані. Операторські панелі можуть бути стаціонарними та ручними (наприклад, для керування роботом або штабелером). Для підключення панелей можна використовувати дротові ба бездротові засоби зв'язку.



Рис. 12.1. Операторські панелі

Операторські станції призначені для відображення інформації у стаціонарних умовах та будуються на основі персональних комп'ютерів (рис. 12.2).



Рис. 12.2. Промислові персональні комп'ютери

У промислових умовах використовують персональні комп'ютери з підвищеним рівнем захисту (промислові персональні комп'ютери). Як правило ці комп'ютери встановлюють у шафах керування, тому широко використовують панельні комп'ютери з сенсорним екраном. Клавіатура теж має підвищений захист. Найчастіше забезпечується рівень захисту IP65.

12.3. Інструментальні програмні засоби для відображення технологічного процесу

Для програмування засобів відображення та введення інформації використовується спеціальне програмне забезпечення.

Воно може бути окремим, або входити до складу програмних пакетів для програмування контролерів. Для програмування операторських станцій використовують досить складні системи проектування, які дозволяють одноразово проектувати декілька операторських станцій з різними рівнями доступу.

На рис. 12.3 показана серeda проектування для операторських панелей та операторських станцій WinCC flexible фірми Сименс, яка складається з структури проекту, області проектування, де відкриваються елементи структури, вікна властивостей елементів області проектування, вікна інструментів, за допомогою яких здійснюється проектування, та вікна виводу, в яке виводяться повідомлення про хід проектування.



Рис. 12.3. Серeda проектування для операторських панелей та операторських станцій WinCC flexible

Для відображення можна використовувати досить велику кількість екранів (зображень) для відображення усього технологічного процесу та окремих його компонентів та пристроїв. Для зображення інформації використовують різні графічні елементи, які знаходяться у

палітрі елементів. Такими елементами можуть бути, наприклад, графічні та числові поля для відображення даних, кнопки та інші елементи керування.

На рис. 12.4 наведені приклади встановлення на екрані полів вводу-виводу (зліва), та кнопок (справа) з вікна інструментів, та встановлення їх властивостей.

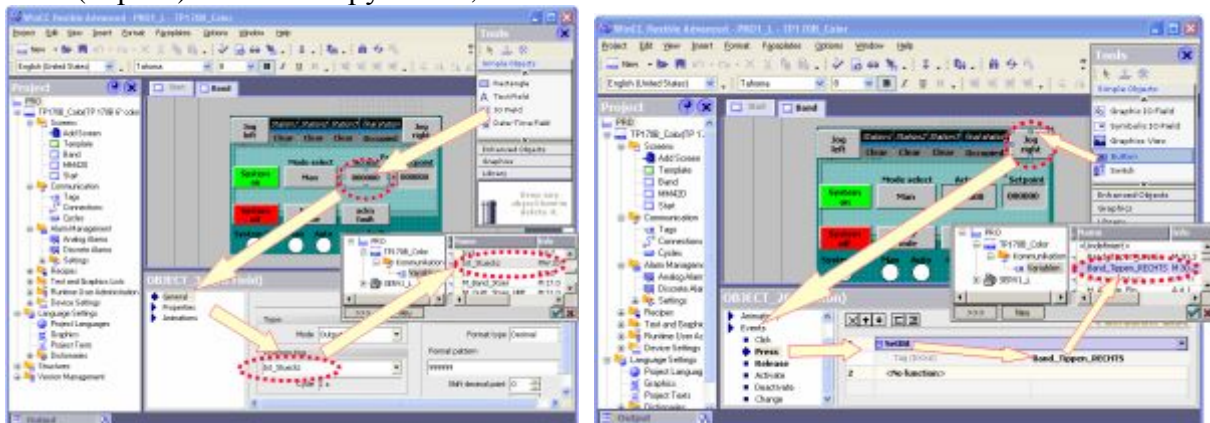


Рис. 12.4. Приклади встановлення на екрані полів вводу-виводу (зліва), та кнопок (справа)

На рис. 12.5 наведений приклад зображення складського обладнання з полями графічного представлення даних у вигляді стелажу з вантажем та штабелером та кнопками керування переміщенням у ручному режимі.

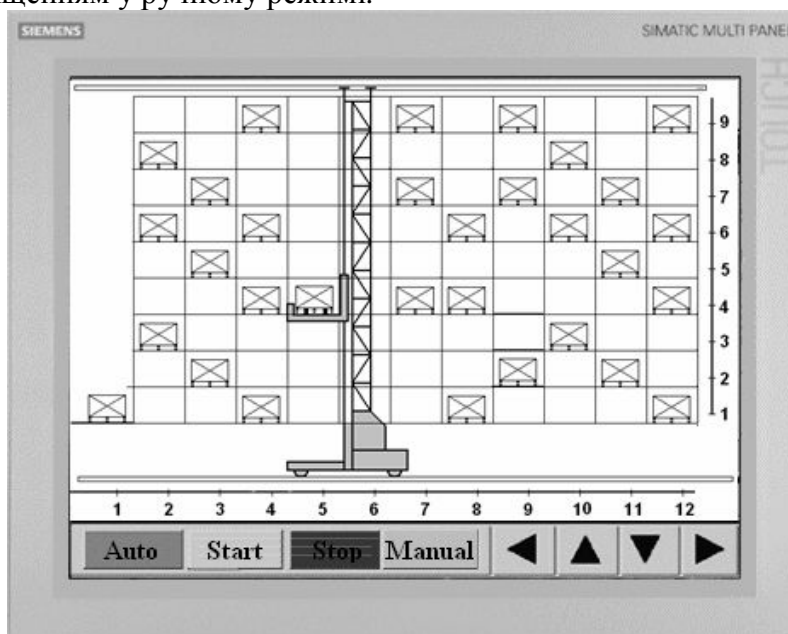


Рис. 12.5. Приклад зображення автоматизованої складської системи

Для відображення складних пристроїв існують різні бібліотеки з компонентами технологічного обладнання для різних задач. На рис. 12.6 наведені приклади таких елементів. Для цих елементів є можливість встановлення динамічних властивостей, наприклад, зміни кольору, розміру та положення в залежності від окремих параметрів, графічного зображення рівня заповнення у резервуарах, та інші.

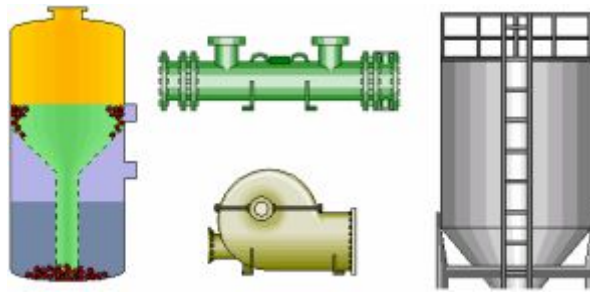


Рис. 12.6. Приклади компонент бібліотек зображень обладнання

При необхідності створення складної структури екранів проектування треба виконувати у такій послідовності:

- планування структури екранів проекту,
- створення екранів,
- створення системи навігації по екранах,
- конфігурування змісту екранів.

У разі виникнення якої-небудь події є можливість виводити на екран відповідні повідомлення. Розрізняють дискретні, аналогові повідомлення, повідомлення за номером та системні повідомлення. Дискретні повідомлення з'являються, коли відповідний біт пам'яті встановлюється в 1. Аналогові повідомлення пов'язані з виходом числових значень у пам'яті ПЛК за встановлені межі. Ці повідомлення видає операторська панель шляхом періодичного опитування відповідних змінних у ПЛК. Повідомлення за номером ініціює ПЛК. У разі виникнення події він надсилає відповідний номер, за яким операторська панель видає повідомлення. Системні повідомлення може надсилати панель та ПЛК (рис. 12.7).

	No.	Time	Date	Status	Text	GR
\$	140000	10:45:28am	9/12/2006	C	Connection established: S7300_DP, Station 2, Rack 0, Slot 2.	0
\$	110001	10:43:37am	9/12/2006	C	Change to operating mode online'	0
B..	1	10:22:42am	9/12/2006	CD	Maximum speed reached: 1262	0
A..	1	10:22:42am	9/12/2006	CDA	Speed too high	0
A..	1	10:22:24am	9/12/2006	CA	Speed too high	0
B..	1	10:22:12am	9/12/2006	C	Maximum speed reached: 1350	0

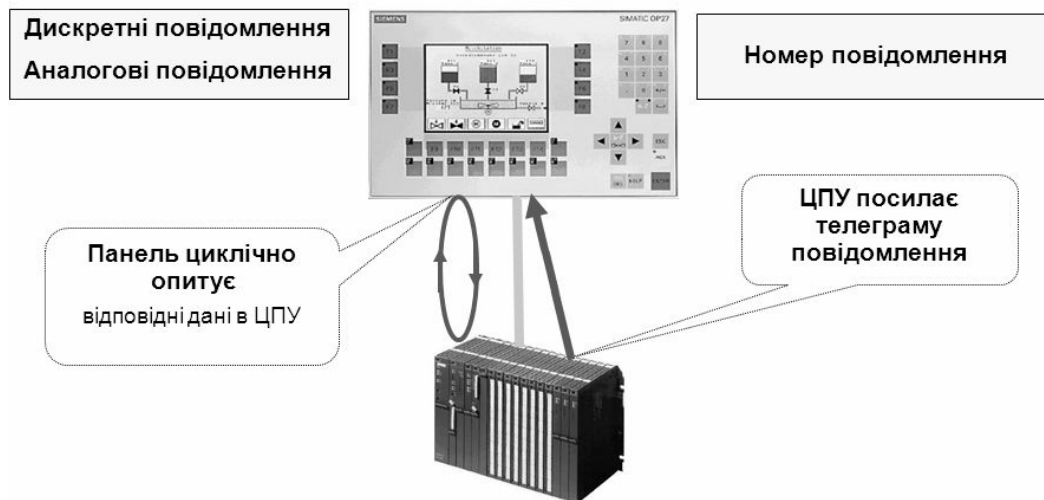


Рис. 12.7. Типи повідомлень у пристроях відображення

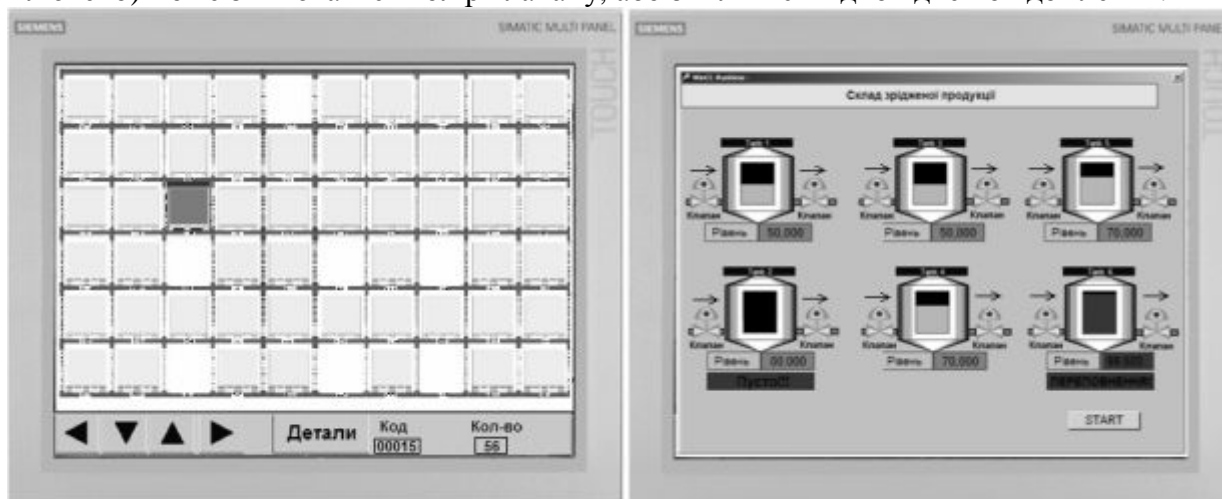
Пристрої відображення також дають можливість робити архіви значень процесу та повідомлень для збереження їх у самому пристрої або у комп'ютері верхнього рівня керування. Для документування значень процесу та повідомлень використовується система звітів. Ці архіви можна використовувати, наприклад, для зберігання інформації о наявності та кількості товарів у окремих стелажах та окремих комірках стелажів.

Більше можливостей у створенні систем візуалізації дає система людино-машинного інтерфеса WinCC (Windows Control Center). Ця система може бути розрахована на багато користувачів з використанням конфігурації "сервер-кліент". Така конфігурація дозволяє

підключити до одного сервера до 32 станцій управління. Проектування сервера може здійснюватися з клієнта.

Засоби візуалізації можуть давати різноманітну інформацію про стан і положення окремих машин як у графічному так і числовому вигляді, наприклад, для автоматизованої складської системи – положення крана-штабелера (рис. 12.5), про зміст комірок стелажу з вказівкою на вид продукту (наприклад, код продукту), та кількість товару даного типу у комірці (рис. 12.8, а). Шляхом переміщення вказівника комірки (наприклад, шляхом зміни кольору вказаної комірки) за допомогою кнопок переміщення можна отримати детальну інформацію о наявності продукції у комірці, типу продукції та її кількості.

Використовуючи засоби динамізації можна показати зміну стану складу. На рис. 12.8, б показано відображення рівня рідини у резервуарах на складі зрідженої продукції. У разі переповнення чи випорожнення резервуару на екрані з'являється повідомлення біля зображення відповідного резервуару (наприклад, Пусто!!!, або Переповнення!!!) При цьому значення рівню у резервуарі може змінювати колір, а сомо повідомлення може мигати. На цьому ж екрані можуть бути елементи керування клапанами завантаження та розвантаження резервуарів. У разі використання сенсорного екрану включення або виключення клапанів може здійснюватися шляхом натиску на клапан, а при зміні його стану (включено або виключено) може змінюватися колір клапану, або з'являтися відповідне повідомлення.



а)

б)

Рис. 12.8. Приклад візуалізації наявності товару на стелажу (а) та візуалізації рівня рідини у резервуарах на складі зрідженої продукції (б)

Для гнучкого керування промисловим обладнанням можна використовувати таку функцію, як створення рецептів, яка є складовою частиною таких систем спостереження та керування, як, наприклад, WinCC flexible фірми SIEMENS.

Рецепти використовуються в тих випадках, коли необхідно передавати взаємопов'язані дані у вигляді записів даних між операторської станцією і ПЛК. При цьому виходять з того, що одночасно використовуються не всі записи даних для рецепта, тому немає необхідності зберігати її усю в системі управління (ПЛК). Ця інформація, яка вимагає багато місця в пам'яті, може зберігатися в операторських станціях, а при необхідності окремі записи даних можуть передаватися або зчитуватися в ПЛК.

Ця функція має стандартні вікна для формування рецепту, де треба вказати компоненти рецепту та їхню кількість у визначеній розмірності.

Основне завдання, яке вирішують рецепти, це визначення кількості окремих компонентів, які складається продукт. Наприклад, при виробленні напоїв можуть виготовлятися різні варіанти кінцевого продукту (сік, нектар, напій), що відрізняються тільки кількістю окремих компонентів (вода, концентрат, цукор та ароматизатор), які можна задавати рецептом (рис. 12.9).

Litre Water	50	Recipe name	No.:
Litre Concentrate	50	Orange	1
Kilo Sugar	30	Data record name	No.:
Gram Aroma	50	Nectar	2

Save	Data to PLS
Load	Data from PLS

Рис 12.9. Рецепт для виготовлення напоїв

Для переходу з одного варіанту кінцевого продукту на інший у цьому разі треба лише змінити рецепт, після чого автоматично здійсниться переналагодження технологічного обладнання.

Цю функцію можна використати також для формування заказу на автоматизованому складі при виконанні функції комісіонування у логістиці.

Ця функція дає можливість сформувавши заказ (у вигляді так званого рецепту), де треба вказати компоненти заказу та їх кількість.

На рис. 12.10 показаний приклад формування заказу на автоматизованому складі за допомогою функції створення рецепту.

Имя рецепта:	№:
Заказ № 1	1
Имя блока данных:	№:
Заказчик № 1	1
Имя записи	Значение
Продукт № 12	12
Продукт № 25	24
Продукт № 22	35
Продукт № 05	125

Готов

Рис. 12.10. Приклад формування заказу за допомогою функції створення рецепту

При завантаженні рецепту у контролер у програму керування надходять змінні, які дозволяють змінити програму таким чином, щоб компоненти вказаного заказу в указаній кількості були переміщені у пункт формування цього заказу. Рецепти можна використовувати для переналагодження послідовності технологічних операцій. На рис. 12.11 наведений приклад встановлення послідовності та часу проведення операцій на лінії обробки.

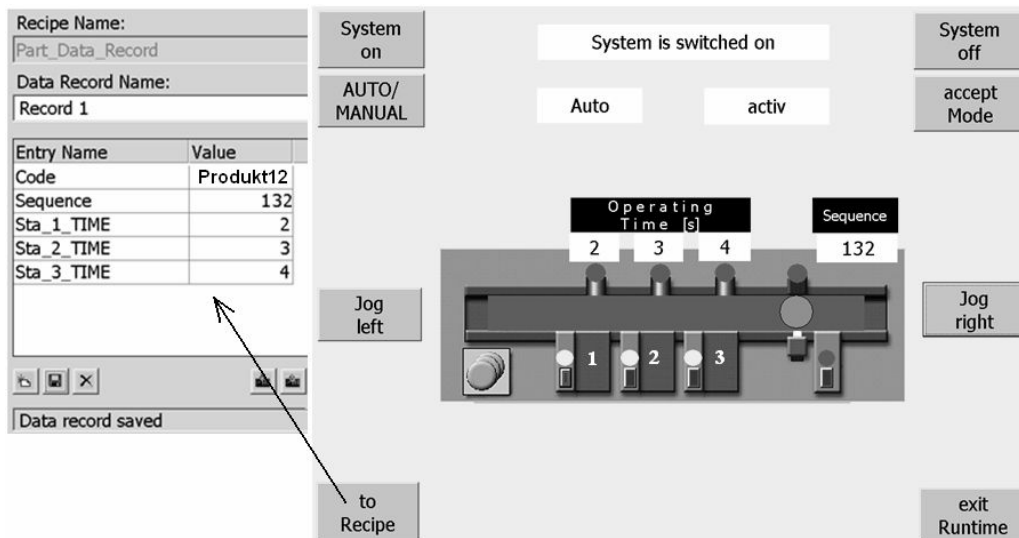


Рис. 12.11. Встановлення послідовності та часу проведення операцій на лінії обробки за допомогою рецепту

Конвеєр послідовно переміщує деталь за вказані робочі місця (1, 2, 3) та затримується на кожному місці згідно з встановленим часом. На прикладі задана послідовність переміщення 1, 3, 2 з затримкою на відповідних місцях 2, 4, та 3 с.

Цей підхід дає можливість швидкого переналадження процесу виготовлення виробів шляхом вибору відповідного набору параметрів технологічного процесу, які задаються заздалегідь підготовленими рецептами. У разі потреби можна швидко підготувати нові рецепти шляхом внесення нових значень для компонентів продукту або параметрів процесу.

Контрольні запитання

1. Які функції реалізує система візуалізації процесу?
2. Як здійснюється проектування систем візуалізації процесу?
3. У яких формах здійснюється представлення даних?
4. Яка послідовність виконання проекту створення складної структури екранів?
5. Які елементи використовують для керування об'єктами?
6. Для чого використовується бібліотеки символів?
7. Для чого призначені повідомлення у системах візуалізації процесу?
8. Які типи повідомлень використовують у системах візуалізації процесу?
9. Які засоби використовують для відображення інформації про стан об'єктів?
10. Для чого використовується функції створення рецепту?

Література

1. Поджаренко В.О., Кучерук В.Ю., Севастьянов В.М. Основы мікропроцесорної техніки. Навчальний посібник. - Вінниця: ВНТУ, 2006. - 226 с. (електронна версія)
2. Иванов Ю.И. Микропроцессорные устройства систем управления: Учебное пособие. / Ю.И.Иванов, ВЛ.Югай - Таганрог: Изд-во ТРТУ, 2005. - 133 с. (електронна версія)
3. Блум Джереми. Изучаем Arduino: инструменты и методы технического волшебства: Пер. с англ. - СПб.: БХВ-Петербург, 2015. - 336 с.: ил. (електронна версія)
4. Шмид Д. и др. Управляющие системы и автоматика /Шмид Д., Бауман А., Кауфман Х., Зиппель Б. Москва: Техносфера, 2007. – 584 с.: ил. (1 экз, кафедра)
5. SIMATIC Totally Integrated Automation. Информация о продуктах. Приборы, системы, консультации, обучение. ДП “Сименс Украина”, Департамент “Средства автоматизации и приводы” 2006 (CD SIMATIC документация, Каталоги по автоматизации, 01/2007). (електронна версія)
6. SIMATIC S7 Комплексная автоматизация -программирование для начинающих. Курс ST-PRO1. Версия 5.5, © SIEMENS AG, A&D 2004. (електронна версія)

Основи логічного комп'ютерного керування

Елементи релейних систем керування

Сучасні системи керування виконують в основному принципи логічного керування. До появи дешевих пристроїв керування на основі мікропроцесорної техніки широко використовувались пристрої логічного керування на основі релейних схем. Компонентами таких систем релейного керування є комутаційні елементи, а саме кнопки, перемикачі, реле, контактори та інші. Умовні позначання основних елементів релейних систем керування наведені на рис. Д1.1.

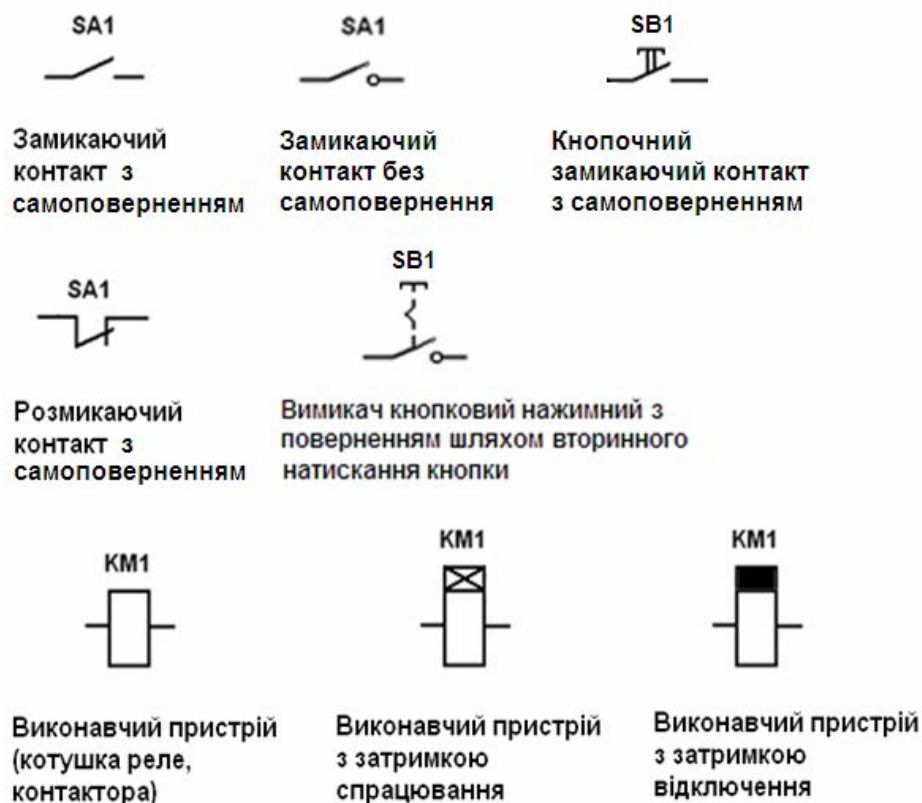


Рис. Д1.1 Умовні позначення основних елементів релейних систем керування

Принцип дії таких систем керування використовується у сучасних програмованих логічних контролерах (ПЛК). Тому кратко зупинимось на релейних системах керування.

На рис. Д1.2 наведений найпростіший електротехнічний пристрій керування, який складається з вмикача S1, реле K1 та контактора K10, що підключає двигун до мережі живлення.

Цій пристрій керування має відокремлене живлення ланцюга керування (24 В постійного струму), яке за допомогою вмикача S1 вмикає реле K1, а той у свою чергу вмикає контактор K10 та замикає ланцюг двигуна (трифазове живлення 400 В). Завдяки цьому здійснюється безпечно керування.

Вмикач S1, наведений на рис. Д1.2, утримується у нажатому стані за допомогою механічного пристрою, який виконує функцію пам'яті. Механічні пристрої можуть виконувати також функції часу (таймери) та рахунку (механічні лічильники).

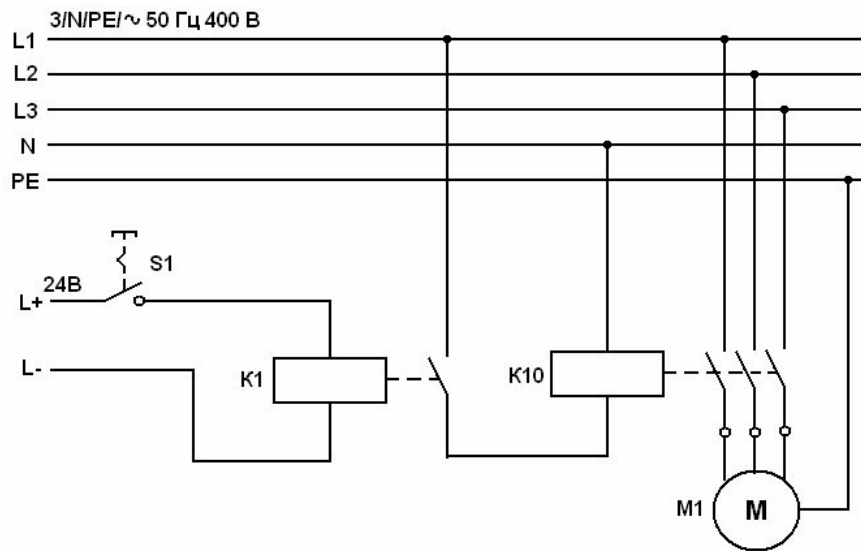


Рис. Д1.2 Найпростіший пристрій керування

За допомогою релейних схем можна також створити логічне керування з використанням логічних функцій (рис. Д1.3). Так послідовне підключення контактів виконує операцію логічного І, паралельне підключення контактів виконує операцію логічного АБО, а нормально замкнений контакт виконує операцію інверсії. Ці логічні операції є основою для інших логічних функцій, на основі яких створюються обчислювальні пристрої. На рис. 3 наведена також реалізація вказаних функцій за допомогою логічних мікросхем.

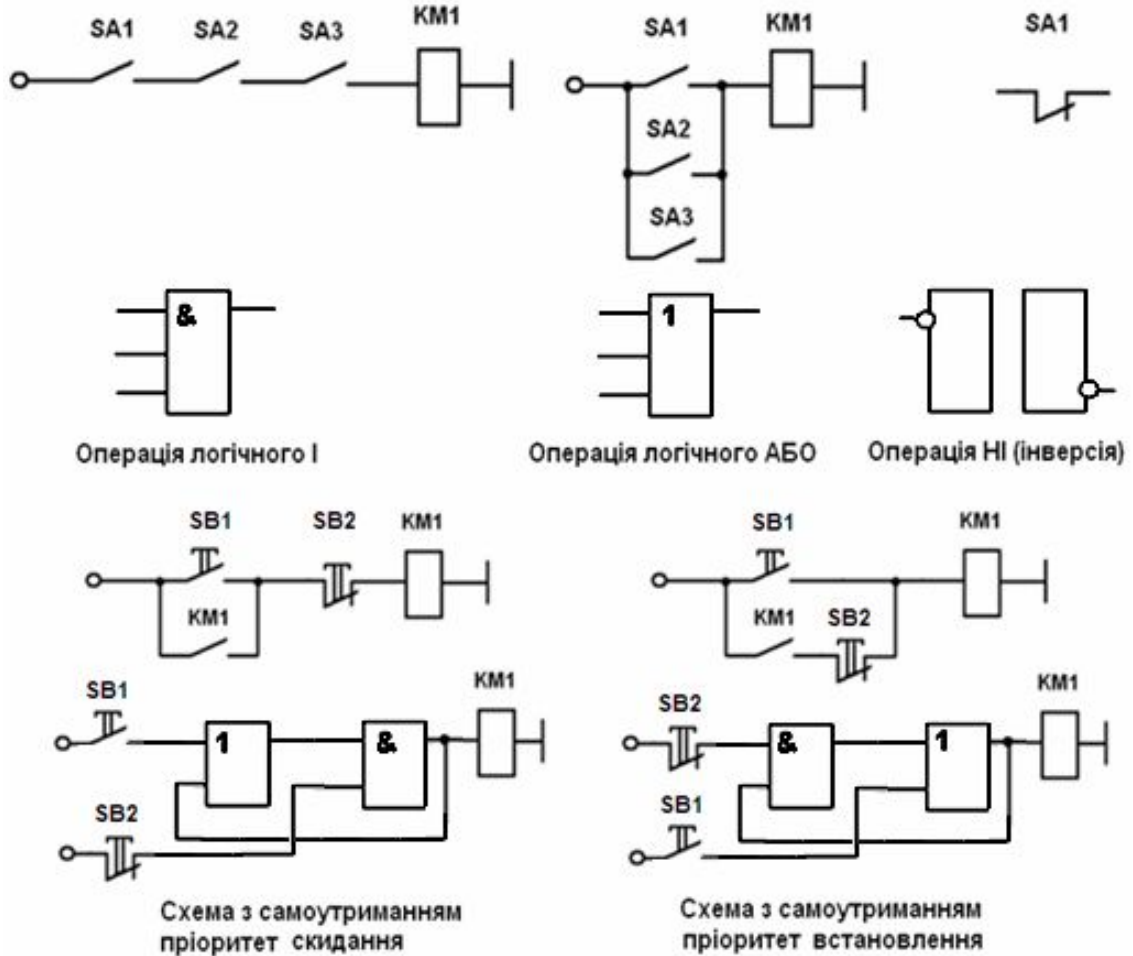


Рис. Д1.3. Основні операції логічного керування

Так, наприклад, функція пам'яті здійснюється за допомогою схеми з самоутриманням. Кнопка SB1 вмикає контактор КМ1. Допоміжний контакт контактора включається та замикає струм живлення контактора, коли кнопка SB1 вимикається. Для відключення контактора КМ1 використовується кнопка SB2. Таким чином кнопка SB1 вмикає контактор (функція встановлення S - set), а кнопка SB2 вмикає контактор (функція скидання R - reset).

Релейні пристрої керування свого часу знаходили досить широке застосування у промисловості. Недоліком таких схем є апаратна реалізація алгоритму керування, що робить неможливим його швидку зміну. Подолати цій недолік стало можливим завдяки сучасним пристроям програмного керування, у яких зміна алгоритму керування здійснюється шляхом зміни програми.

З іншого боку широке застосування релейних пристроїв керування наклало свій відбиток на сучасні мови програмування пристроїв керування. Так однією з форм представлення програм ПЛК є так звана релейно-контактна схема (контактний план, Ladder), яка буде розглянута далі.

Системи релейного та комп'ютерного керування

Для прикладу розглянемо, як здійснюється керування підйомником, зображеного на рис. Д1.4, за допомогою релейної схеми та програмного керування.

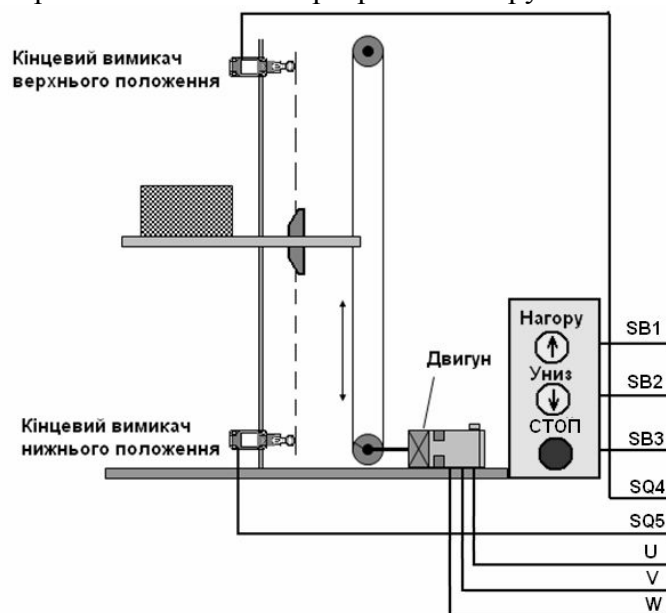


Рис. Д1.4. Схематичне зображення підйомника

Релейна схема керування складається з головного кола, яке підключає двигун до мережі живлення, та допоміжного кола, яке здійснює саме керування (рис. Д1.5). Керування здійснюється за допомогою трьох кнопок "Нагору" (SB1), "Униз" (SB2), "СТОП" (SB3), кінцевих вимикачів верхнього (SQ4) та нижнього (SQ5) положень. У підйомнику використовується трифазний двигун (фази U, V, W). Кнопки "Нагору" (SB1), "Униз" (SB2) двоколові, вони використовують замикаючий та розмикаючий контакти. Замикаючі контакти вмикають рух у відповідному напрямку, змінюючи напрямку руху за допомогою контакторів КМ2 та КМ3. Ці контактори змінюють підключення двох фаз V та W, що в свою чергу змінює напрям обертання двигуна. Контакт КМ1 підключає двигун до силової мережі трифазного струму. Контакти, що розмикаються, кнопок "Нагору" та "Униз" блокують включення двигуна при одночасному натисканні кнопок "Нагору" та "Униз".

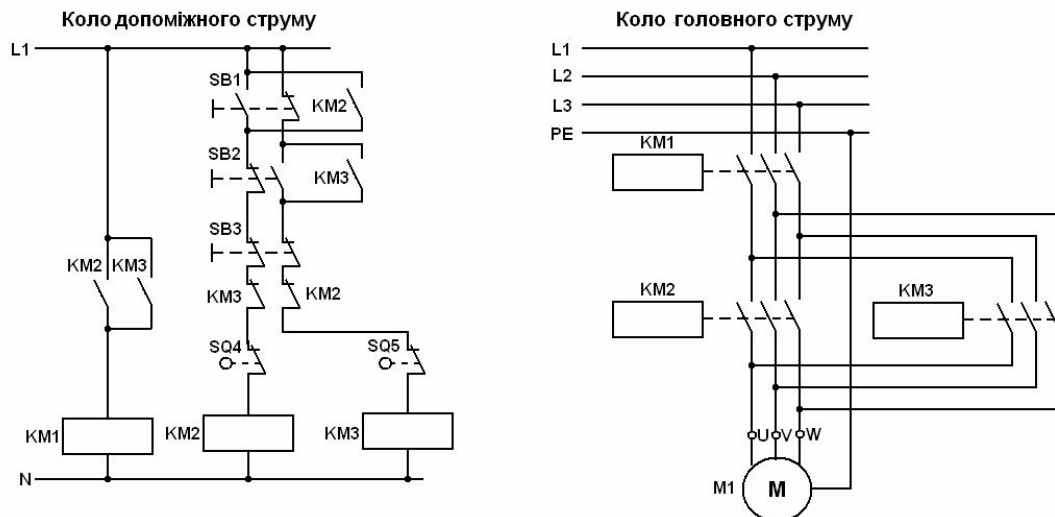


Рис. Д1.5. Схема керування підйомником

Допоміжні контакти, що розмикаються, контакторів KM2 та KM3 блокують включення одного напрямку руху, якщо включений другий. Тому при натисканні кнопок "Нагору" або "Униз" розмикаючий контакт спочатку відключить рух у іншому напрямку та блокування допоміжного розмикаючого контакту, а тільки потім спрацює контактор руху у відповідному напрямку.

Допоміжні замикаючі контакти контакторів KM2 та KM3, що включені паралельно замикаючим контактам кнопок "Нагору" та "Униз", замикають коло живлення контакторів і дозволяють продовжити рух, якщо відпустити ці кнопки. Кнопка "СТОП" теж двоколова, вона має два розмикаючих контакти. Це дозволяє підключити її до кіл живлення рухів в обох напрямках і відключити двигун незалежно від напрямку руху. Контактор KM1 підключає двигун до силової мережі, якщо включений один з напрямків руху.

Наведена схема включає різні функції логічного керування, реалізованих за допомогою релейних схем, а саме функції І, АБО та НІ, а також функції пам'яті. Релейні пристрої керування у певний час, а у деяких випадках і досі, знайшли дуже широке використання. Недоліком таких систем керування є неможливість швидкої зміни алгоритму керування, бо це потребує зміни електричної схеми, яка саме і реалізує цей алгоритм.

Заміна релейної логіки на напівпровідникові логічні схеми, а потім і мікросхеми не зняла цей недолік, бо залишилась апаратна реалізація алгоритму керування.

Суттєва зміна підходу до будовання систем керування здійснилась завдяки використанню цифрових обчислювальних пристроїв, які дають змогу створення алгоритму керування будь-якої складності за допомогою програм. Головною перевагою такого програмного, або комп'ютерного керування є можливість швидкої зміни програми, що в свою чергу змінює алгоритм керування та робить систему керування гнучкою.

Ця перевага знайшла своє втілення, наприклад, у так званих програмованих логічних контролерах (ПЛК), які мають модульну структуру та являють собою основу для створення гнучких промислових систем керування.

Розглянемо, як здійснюється керування підйомника, наведеного на рис. 3.1, за допомогою програмованого логічного контролера (рис. Д1.6).

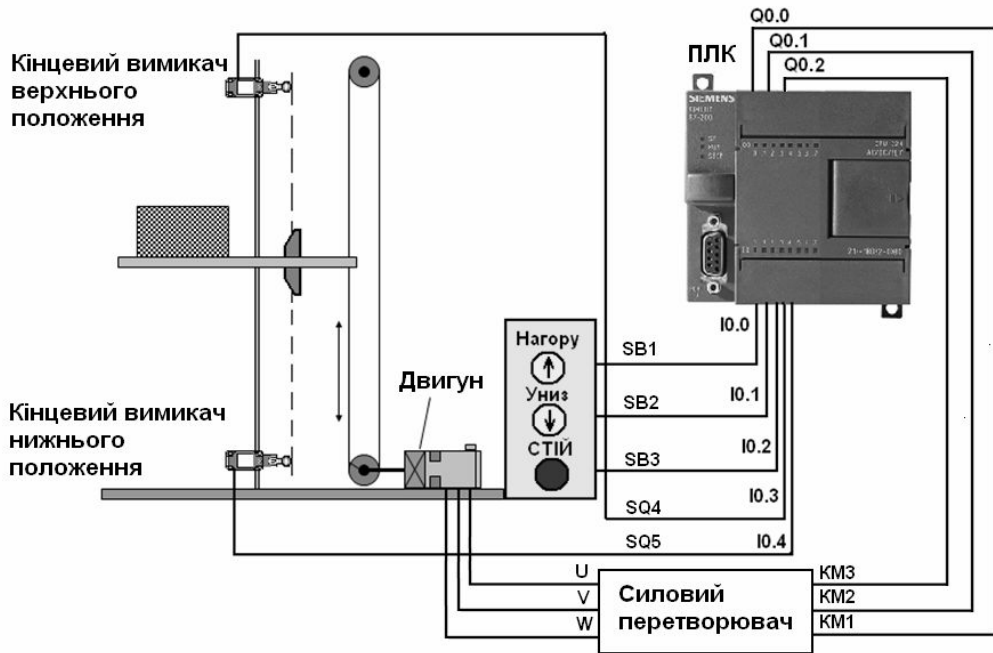


Рис. Д1.6. Керування підйомника за допомогою програмованого логічного контролера

Треба зазначити, що комп'ютерне керування у даному випадку замінює коло допоміжного струму, бо саме там закладений алгоритм керування. Коло головного струму, наведене на рис. Д1.5, залишається, бо воно здійснює підключення двигуна до мережі живлення та виконує функції силового перетворювача, який за допомогою вихідних сигналів ПЛК підключає двигун до мережі живлення та змінює його напрям обертання.

Одна з форм представлення програм у сучасних ПЛК, які більш детально будуть розглянуті далі, є форма у вигляді контактної плану (релейно-контактної схеми), що близька по своєму вигляду до електричної схеми, тому використаємо цю форму для програми керування підйомником.

На рис. Д1.7 наведено порівняння програми керування з колом допоміжного струму (див. рис. Д1.5), яке здійснює алгоритм керування.

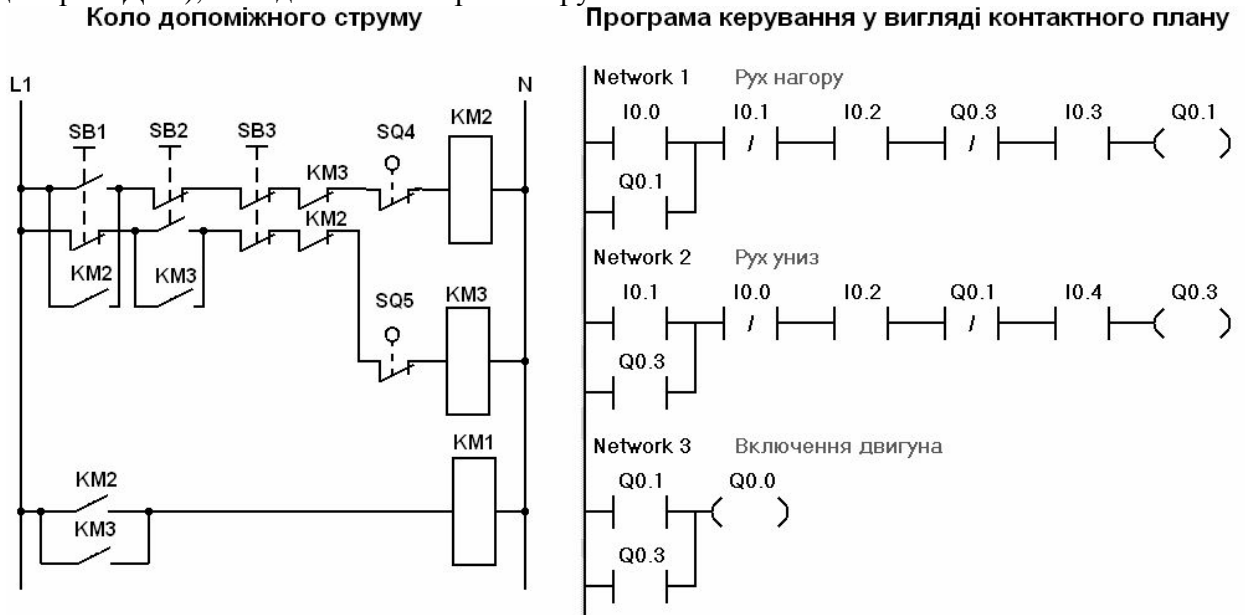


Рис. Д1.7. Порівняння програми керування у вигляді контактної плану з колом допоміжного струму

Якщо треба змінити алгоритм керування системи релейного керування, це потребує зміни електричної схеми кола допоміжного струму. При використанні комп'ютерного керування зміна алгоритму здійснюється лише шляхом зміни програми, а електрична схема не змінюється.

Мова програмування LOGO!Soft Comfort

Демоверсію мови програмування LOGO!Soft Comfort V8, яка дозволяє повністю здійснити програмування та перевірити програму на програмному емуляторі, можна знайти в Інтернеті на сторінці фірми Siemens AG:

<http://w3.siemens.com/mcms/programmable-logic-controller/en/logic-module-logo/demo-software/pages/default.aspx>

На рис. Д2.1 наведено вікно відкритої мови програмування.

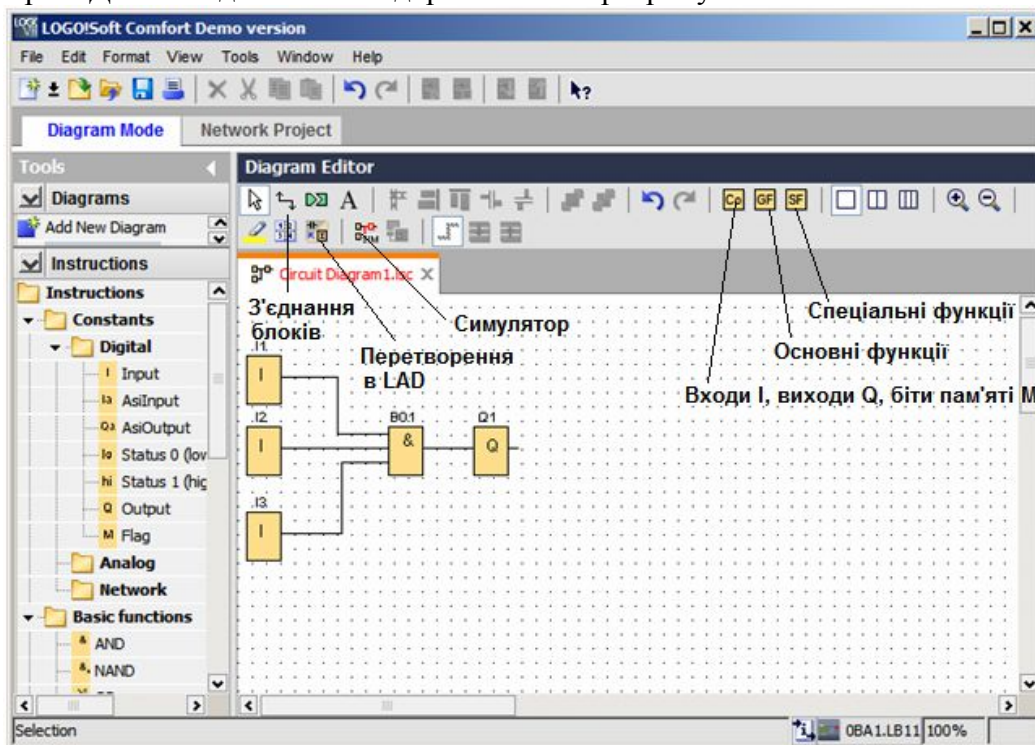


Рис. Д2.1. Вікно відкритої мови програмування

На рис. Д2.2 показані елементи, що використовуються для програмування, включаючи позначення змінних (входів, виходів, пам'яті), основні та спеціальні функції, а на рис. Д2.3 наведено, як здійснюється їх встановлення у програмі.



Рис. Д2.2. Елементи, що використовуються для програмування

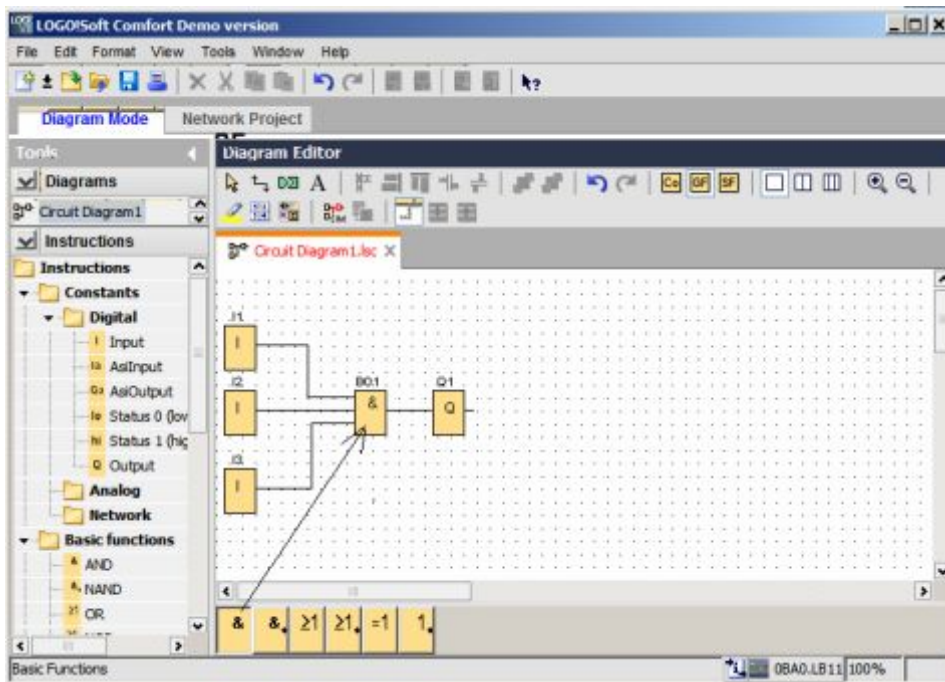


Рис. Д2.3. Встановлення елементів до програми

На рис. Д2.4 показано, як здійснюється робота з симулятором.

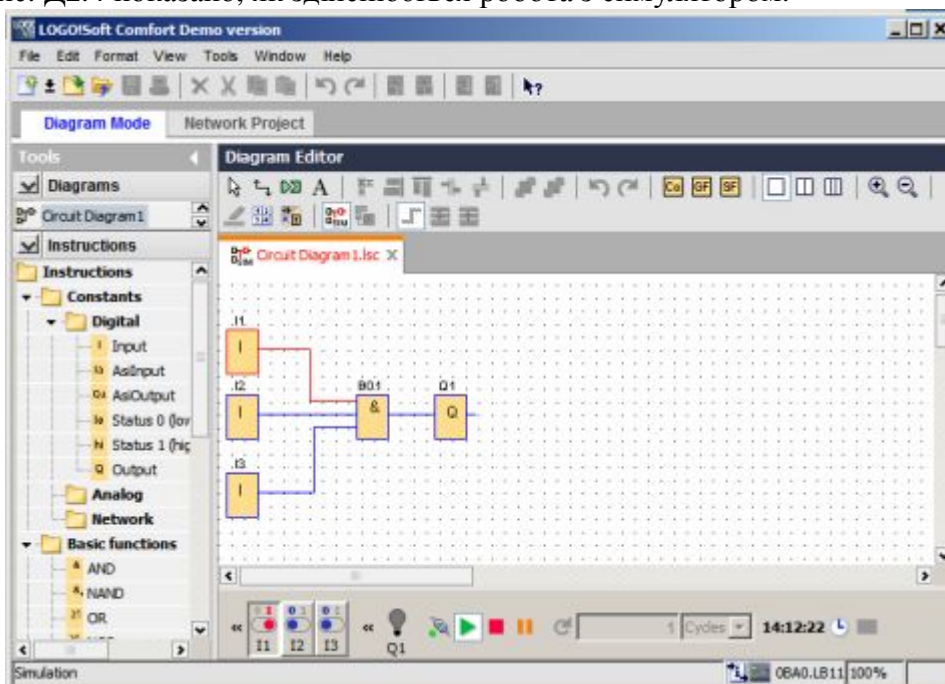


Рис. Д2.4. Робота з симулятором

Вище було наведено представлення програми у вигляді функціональної схеми (форма представлення Circuit Diagram).

На рис. Д2.5 наведено представлення програми у вигляді релейно-контактної схеми (форма представлення LAD).

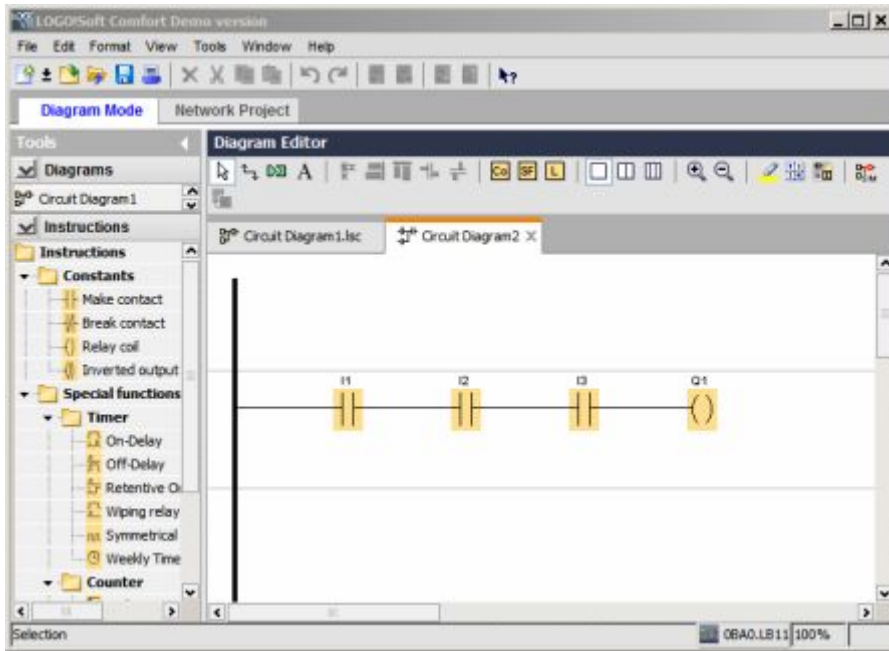


Рис. Д2.5. Представлення програми у вигляді релейно-контактної схеми (форма представлення LAD)

Далі наведені функції, що рекомендуються для використання на практичних заняттях.

Основні функції

При введенні програми в LOGO! у списку GF визначені основні функції, а само:


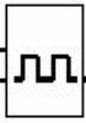

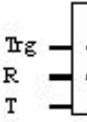

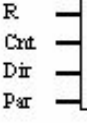
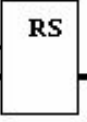
Зображення в LOGO!	Позначення основних функцій	Зображення в LOGO!	Позначення основних функцій
	AND - І		XOR (або, що виключає)
	OR - АБО		NAND (І НІ)
	NOT (НІ)		NOR (АБО НІ)

Таблиця для основних функцій

Входи			AND (І)	OR (АБО)	NAND (І НІ)	NOR (АБО НІ)
I1	I2	I3	Q	Q	Q	Q
0	0	0	0	0	1	1
0	0	1	0	1	1	0
0	1	0	0	1	1	0
0	1	1	0	1	1	0
1	0	0	0	1	1	0
1	0	1	0	1	1	0
1	1	0	0	1	1	0
1	1	1	1	1	0	0

Спеціальні функції

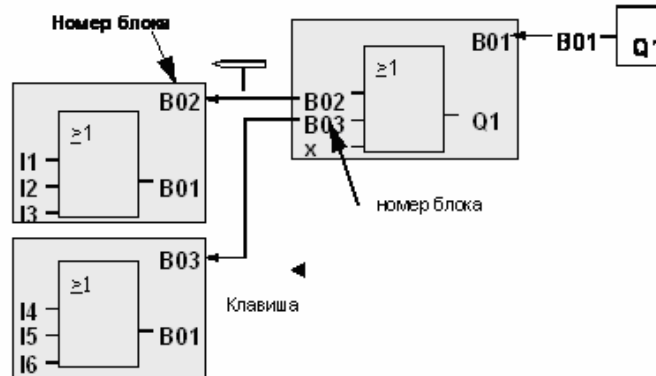
При введенні програми в LOGO! у списку SF визначені спеціальні функції, а само :

Зображення в LOGO!	Позначення спеціальних функцій	Зображення в LOGO!	Позначення спеціальних функцій
	Затримка включення		Датчик тактових імпульсів
	Затримка вимикання		Затримка включення, що зберігається
	Реле часу		Двонаправлений лічильник
	Реле із самопідхватом RS-тригер		

В усіх функціях **R** має пріоритет перед всіма іншими входами.

Завжди при введенні блок у програму, LOGO! привласнює йому номер, номер блока. Номер блока стоїть вгорі справа на дисплеї.

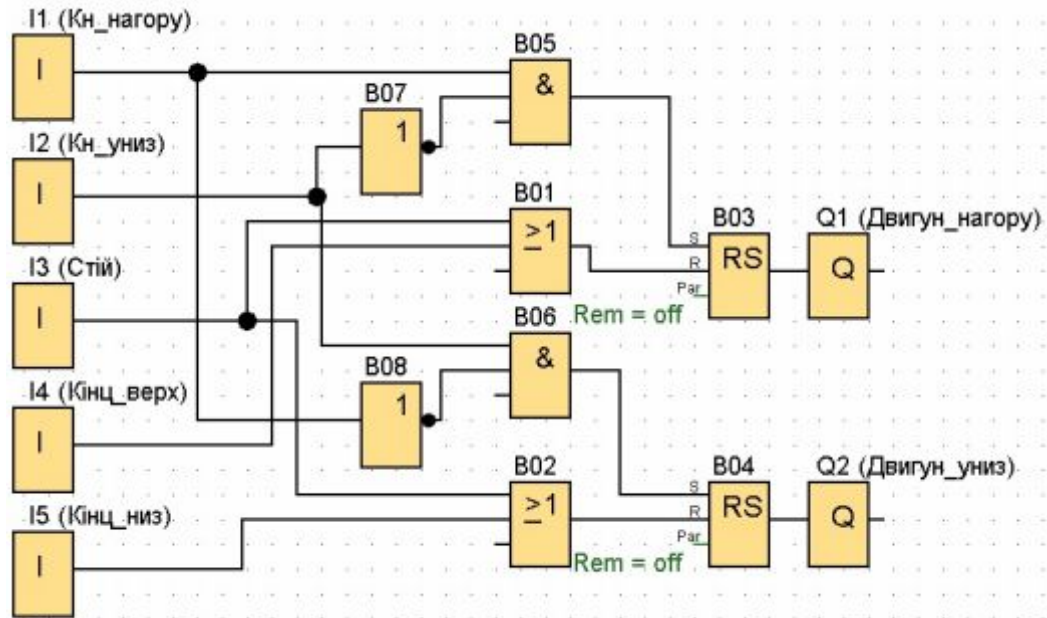
За допомогою номера блока LOGO! показує Вам зв'язок між блоками:



Приклади програм.

Програма керування підйомачем.

Програма, що здійснює керування підйомачем, має такий вигляд.



Підйомач переміщається у двох напрямках: догори та униз.

Для прямування нагору необхідно натиснути кнопку "Кн_нагору".

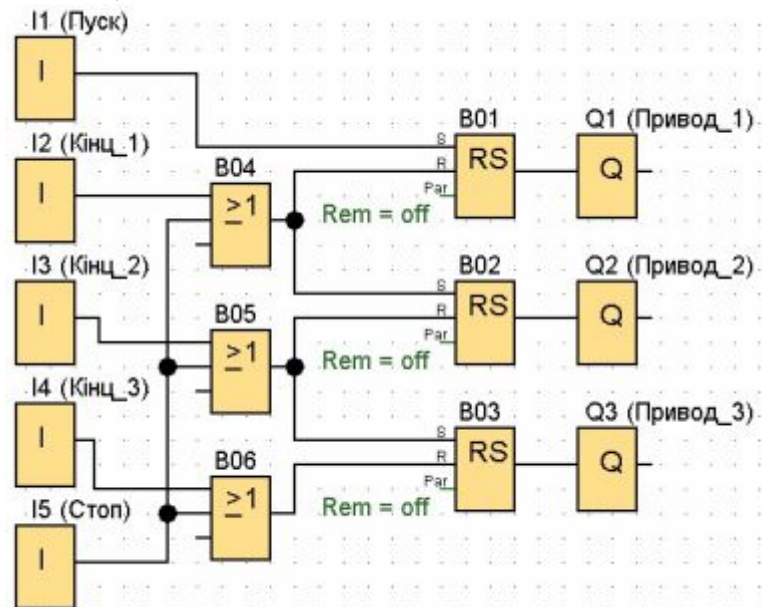
Для прямування униз необхідно натиснути кнопку "Кн_униз".

Прямування нагору або униз припиняється при натисненні кнопки "Стій" або при досягненні кінцевих вимикачів "Кінц_верх" або "Кінц_низ".

Блоки B05, B06, B07, B08 здійснюють захист від одночасного натискання кнопок переміщення догори та униз.

Програма послідовного переміщення

Програма, що здійснює послідовне переміщення трьох приводів, "Привод_1", "Привод_2" та "Привод_3", має такий вигляд.



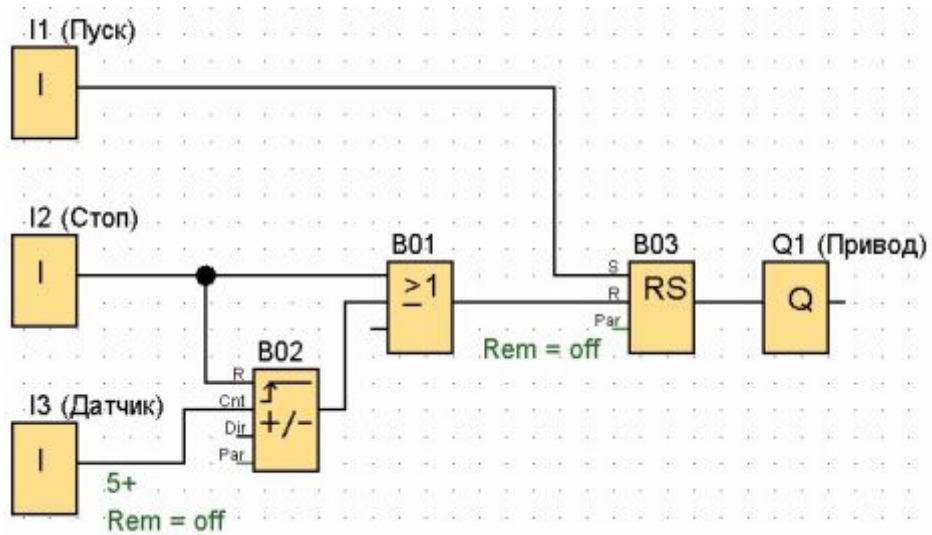
Переміщення приводу 1 запускається кнопкою "Пуск".

Кінцевий вимикач першого переміщення зупиняє привод 1 та запускає привод 2.

Аналогічно здійснюється переміщення приводу 2 та приводу 3.

Програма переміщення до вказаної позиції.

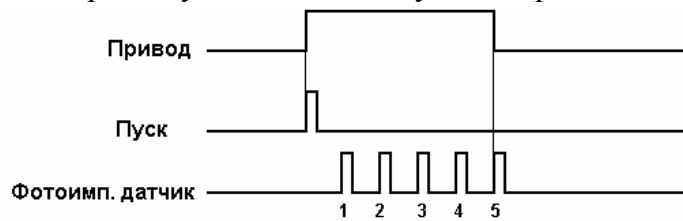
Програма, що здійснює переміщення до позиції 5, має такий вигляд.



Сигнал "Пуск" запускає привод.

Під час переміщення фотоімпульсний датчик видає імпульси.

На 5-ому імпульсі спрацьовує лічильник та зупиняє привод.



Сигнал "Стоп" зупиняє привод та перезавантажує лічильник.