Міністерство освіти і науки України
Запорізька державна інженерна академія

МІКРОПРОЦЕСОРНІ СИСТЕМИ АВТОМАТИЗАЦІЇ

Методичні вказівки
до лабораторного практикуму для студентів ЗДІА
спеціальності "Гідроенергетика"

рекомендовано до видання
на засіданні кафедри "Гідроенергетика"
протокол № 5 від " 2 " квітня 2006 р.

Запоріжжя
2006

Мікропроцесорні системи автоматизації. Методичні вказівки до лабораторного практикуму для студентів ЗДІА спеціальності "Гідроенергетика" (7.090503) / Укладачі:. В. М. Чван, В. В. Назаренко. - ЗДІА, 2006. - 25 с.

Укладачі:
В. М. Чван - ст. викладач,
В. В. Назаренко - асистент.

Відповідальний за випуск -
завідувач кафедрою "Гідроенергетика"
доцент В. В. Радченко.

ЗМІСТ

ВСТУП

Метою розробки методичних вказівок є практичне освоєння теоретичної частини курсу дисципліни "Мікропроцесорні системи автоматизації".

Невід'ємною частиною знань для сучасного фахівця-енергетика є область мікроелектроніки в частині створення багаторівневих мікропроцесорних систем керування (МПС) устаткуванням і технологічними процесами виробництва, застосування промислових контролерів, їхній оптимальний вибір і планування необхідної продуктивності обчислювальних засобів відповідно до рівня складності алгоритмів керування. Подальший розвиток інтегральної елементної бази припускає більш широке застосування в гідроенергетиці мікропроцесорних схем керування і здійснення багатофункціональної обробки інформації завдяки раціональному сполученню мікропроцесорних засобів із жорсткою структурою й гнучким програмним керуванням, а також широке використання наборів мікропроцесорних функціональних пристроїв, об'єднаних стандартними інтерфейсами.

Практичне освоєння процесів керування за допомогою мікропроцесорних контролерів модульної конструкції із сучасним програмним забезпеченням, що має переналагоджувану конфігурацію, що, а також із застосуванням детермінованої багатозадачної операційної системи, що працює в режимі реального часу, - є основним завданням лабораторного практикуму. У процесі роботи практично вивчаються апаратні і програмні засоби МПС: склад і характеристики промислових контролерів; програмне середовище для програмування відповідно до заданих алгоритмів, побудова оптимального варіанта МПС. Практичні знання, отримані при виконанні лабораторного практикуму, є необхідною частиною знань фахівця-енергетика, що відповідають сучасному рівню виробництва гідроенергетики.

1 МЕТА РОБОТИ

1.1 У процесі виконання лабораторного практикуму студент повинен освоїти:

- склад і характеристики промислових контролерів;
- правила побудови МПС;
- вибір апаратних засобів системи керування - оптимального варіанта набору модулів МПС;
- основи практичної побудови системи для заданого алгоритму керування;
- основи програмування із застосуванням проблемно-орієнтованої мови.

2 ТЕОРЕТИЧНА ЧАСТИНА

2.1 Мікропроцесорна система керування (МПС) у більшості випадків є обов'язковою частиною технологічного встаткування й забезпечує погоджене функціонування пристроїв, що входять у підсистеми. Технічні засоби її реалізують наступні функції:

1. збір і обробка в реальному часі різноманітних вимірюваних параметрів;
2. попередні перетворення різного роду інформації;
3. відображення інформації;
4. контроль і оцінка параметрів і порівняння зі стандартами;
5. діагностика, аналіз функціонування МПС із можливістю виклику діагностичних програм;
6. архівація великої кількості даних з можливістю перегляду й аналізу;
7. оформлення протоколів на паперовому і магнітному носії;
8. зв'язок з іншими обчислювальними системами;
9. видача сигналів керування для використання в системах сигналізації й керування.

2.2 Центральне місце в МПС займають мікропроцесорні пристрої керування - програмувальні логічні й регулюючі контролери, контролери змішаного типу, спеціалізовані контролери.

Цілий ряд позитивних якостей цих контролерів робить їх зручними для використання: модульна конструкція, робота із природним охолодженням, стійкість до вібраційних навантажень, гнучкі можливості розширення, наявність комунікаційних функцій, системи вбудованих функцій, широкий спектр варіантів побудови систем локального і розподіленого вводу-виводу. Гнучкість, розширюваність, можливість сполучення контролерів всіх модифікацій в одній системі керування дозволяють створити найсучасніші концепції децентралізованого керування. Застосовувана операційна система є багатозадачною і працює в режимі реального часу.

2.3 Вивчення побудови системи виробляється на базі контролерів фірми B&R (Bernecker+RainerIndustri-Elektronik). Контролери конструктивно розділені на окремі складові частини - модулі, які дозволяють робити їхній оптимальний вибір у кожному конкретному випадку без надлишків їх ресурсів в архітектурі.

Операції зв'язку виконує модуль процесора (CPU) через дві серії інтерфейсів: RS232 і CAN. Інтерфейси програмно-сумісні і можуть працювати як інтерфейси режиму online або інтерфейси даних. Інтерфейс RS232 призначений, насамперед, для програмування CPU. Він може бути також використаний як загальний інтерфейс (наприклад, для пристрою візуалізації, принтера, пристрою зчитування цифрового коду і т.п.). Інтерфейс CAN – стандартний інтерфейс польової шини для зв'язку з іншими системами керування, віддаленого розширення вводу-виводу з використанням модулів віддаленого контролера і контролера шини CAN.

2.4 Широке застосування мікропроцесорних засобів керування вимагає, щоб більшість фахівців, як конструкторів, так і технологів, опанувало прийомами програмування завдань керування спеціальним технологічним устаткуванням, а для цього необхідні доступній відносно прості засоби (мови) розробки програм керування. При створенні мов, орієнтованих на розробку завдань керування, розроблювач зіштовхується з різноманітними й найчастіше суперечливими й неадекватними вимогами користувачів. Одним з перших кроків при розробці мов керування є вибір ступеня проблемної орієнтації мови. Проблемно-орієнтована мова користувача повинен дозволяти описати алгоритм функціонування встаткування (мова опису), запрограмувати цей алгоритм на конкретних мікропроцесорних технічних засобах (базова мова) і реалізувати алгоритм мовою мікроЕОМ (машинна мова). Система керування технологічним устаткуванням повинна забезпечувати виконання двох основних функцій :керування циклом і параметрами.

2.5 Серед мов описів алгоритмів керування найбільш широке поширення одержали мови релейно-контактних символів. Відносно проста, але досить ефективна є мова, заснована на принципі тимчасових інтервалів: "час-команда", "час-параметр". Програмне забезпечення системи керування встаткуванням, написане мовою високого рівня й оформлене у вигляді окремих програмних модулів, з яких компонується загальна програма, має природну мову користувача, що дозволяє в діалоговому режимі через функцію "меню" змінювати настроювання програми керування в досить вузьких межах (наприклад, значення параметрів, тимчасові затримки, допусковый контроль і т.п). У загальному випадку спектр мов програмування для системи на базі контролерів B&R включає:

-ступеневі діаграми (LAD). Всі функціональні можливості контактів, логічних і функціональних схем об'єднані в LAD. Завдяки подібності із принциповою електричною схемою, ступенева діаграма – це найпростіша й візуально наочна форма програмування цифрових і аналогових функцій;

-список команд (IL). Це машинно-орієнтована мова, яку можливо використати, щоб створювати логічні з'єднання способом, аналогічним LAD;

-структурований текст (ST);

-послідовна функціональна схема (SFC);

-B&R Automation Basic (AB);

-ANSI C Мова високого рівня (стандартизований Сі). Високоефективна мова програмування, використовувана в системах автоматизації. Стандартна структура забезпечує швидке й ефективне програмування. Універсальний набір програм утворять бібліотеку програм, що підтримують роботу системи керування для різних конфігурацій контролера.

2.6 Досвід роботи з релейно-контактними схемами керування привів до створення відповідної мови - LAD, що складається з багатьох різнотипних символів. З них можна комбінувати й формувати багато різних логічних структур. Схеми LAD складаються з наступних компонентів: логічних, таймерів

і лічильників, символів входів, символів виходів, ліній сполучення. Позначення й назви символів підрозділяються згідно таблиці 2.6.

Таблиця 2.6 – Символи (контакти) LAD - позначення, назва

	Норм. Відкритий <c>		Норм. закритий <i>		Вхід позитивн, одновибратора <p>		Вхід .від´ємн одновібратора <n>
	Вхід одновібратора <b>		Вихід <shift><C>		Інверт. вихід <shift><I>		Фіксатор <shift><S>
	Дефіксатор <shift><R>		Вихід позитивн. одновібратора <shift><P>		Вихід від´ємн, одновибратора <shift><N>		Вихід одновібратора <shift><B>
	Перехід <j>		Повернення <e>		Функціональн. блок <f>		Аналогове значення <пробіл>
	Опис <d>		Мітка <l>				
	Лінія вліво <Alt><Left>		Лінія вправо <Alt><Right>		Лінія вгору <Alt><Up>		Лінія вниз <Alt><Down>
	Перевірить <Enter>						

Функціональне призначення символів (контактів):

- контакт “**нормально відкритий”** – призначений для цифрової змінної (типу BIT). Зв'язок між входом і виходом контакту згідно рис. 2.6.1:

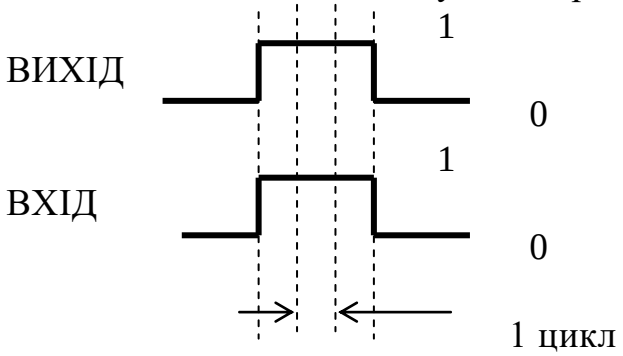

Рисунок .6.1 – Графік функції нормально відкритого контакту

- контакт “**нормально закритий”** – призначений для цифрової змінної (типу BIT). Зв'язок між входом і виходом контакту згідно рис. 2.6.2:

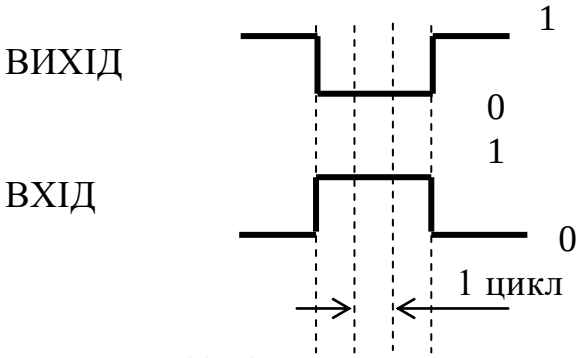

Рисунок 2.6.2 – Графік функції нормально закритого контакту

- контакт “**вхід позитивний одновібратора”** – призначений для цифрової змінної (типу BIT). Зв'язок між входом і виходом контакту згідно рис. 2.6.3:

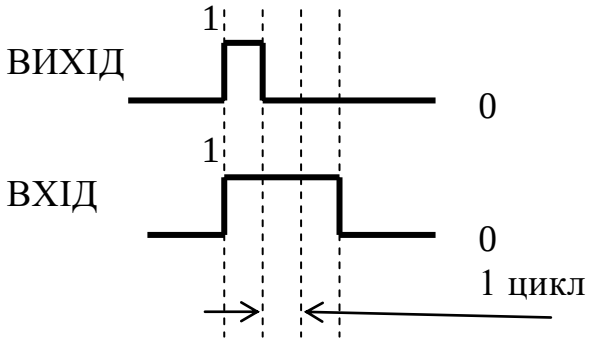

Рисунок 2.6.3 – Графік функції контакту "вхід позитивний одновібратора
Звичайно він використовується для входів функціонального блоку лічильника або реєстрації подій.

- контакт “**вхід негативний одновібратора”** – призначений для цифрової змінної (типу BIT). Зв'язок між входом і виходом контакту згідно рис. 2.6.4:

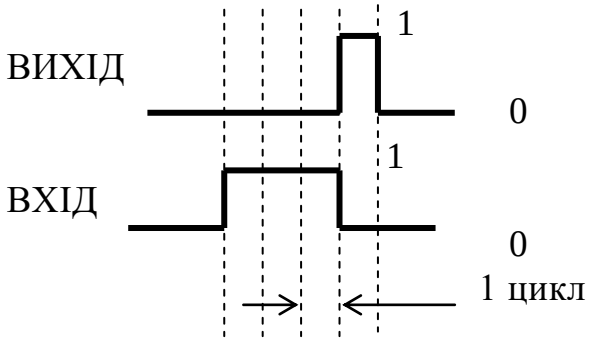

Рисунок 2.6.4 – Графік функції контакту "вхід негативний одновібратора"
Контакт використається для реєстрації подій;

- контакт “**вхід одновібратора”** – призначений для цифрової змінної (типу BIT). Зв'язок між входом і виходом контакту згідно рис. 2.6.5:

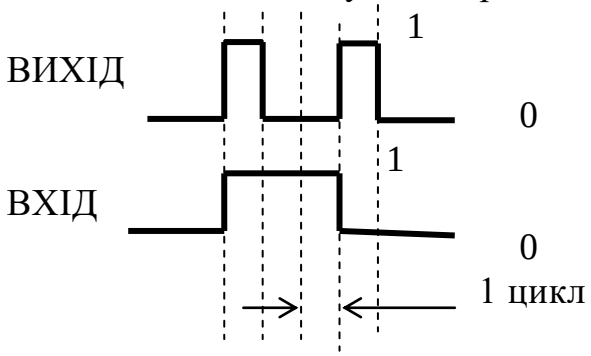

Рисунок 2.6.5 – Графік функції контакту "вхід одновібратора"
Контакт використовується для реєстрації подій;

- контакт “**перехід”** - контакт, що виконує передачу управління цепочці програми, яка розміщена після однойменної мітки (виконує функцію проміжної змінної);
- контакт “**повернення**” виконує функцію переривання послідуючого виконання програми;

- контакт “**мітка**” – контакт для переходу;
- контакт “**лінія вліво** (вправо, вгору, вниз)” зєднувальні лінії між контактами;
- контакт “**опис**” – коментарій програміста;
- контакт “**вихід”** – призначений для цифрової змінної (типу BIT). Зв'язок між входом і виходом контакту згідно рис. 2.6.6:

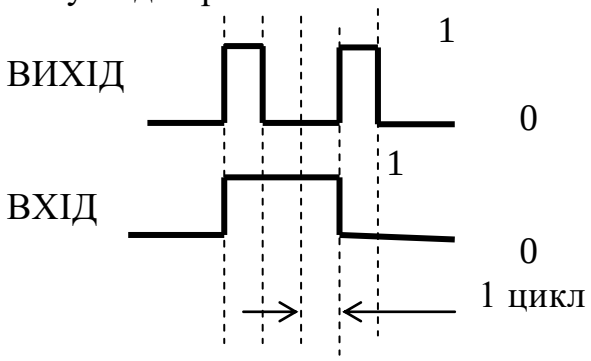

Рисунок 2.6.6 – Графік функції контакту "вихід"

- контакт “**інвертований вихід”** – призначений для цифрової змінної (типу BIT). Зв'язок між входом і виходом контакту згідно рис. 2.6.7:

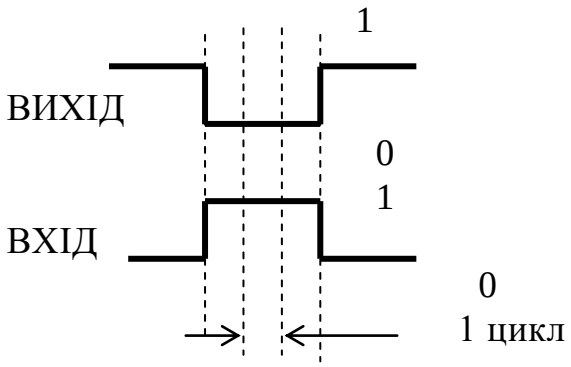

Рисунок 2.6.7 – Графік функції контакту "інвертований вихід"

- контакт “**аналогове значення”** – використовується для завдання аналогових значень на аналогових входах функціональних блоків;
- контакт “**фіксатор”** – цифровий вихідний контакт. Коли ліворуч від контакту з'являється логічна одиниця, вихідна змінна встановлюється й фіксується рівною логічній одиниці, навіть коли вхід знову стає рівним нулю. Зв'язок між входом і виходом контакту згідно рис. 2.6.8:

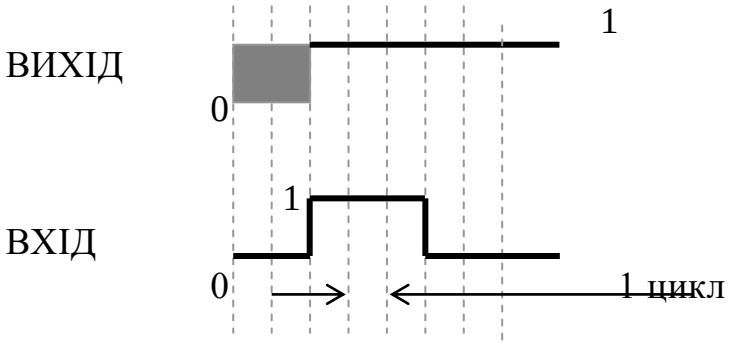

Рисунок 2.6.8 – Графік функції контакту "фіксатор"

Де: - не визначене значення ("0" або "1");

- контакт “**дефіксатор”** – цифровий вихідний контакт. Коли ліворуч від контакту з'являється логічна одиниця, вихідна змінна скидається і фіксується рівній логічному нулю, навіть коли вхід знову стає рівним нулю. Зв'язок між входом і виходом контакту згідно рис. 2.6.9:

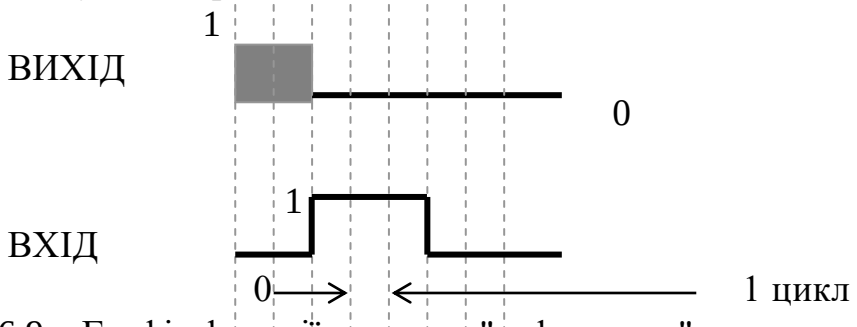

Рисунок 2.6.9 – Графік функції контакту "дефиксатор"

Де: - невизначене значення ("0" або "1");

- контакт “**вихід позитивний одновібратора”** – цифровий вихідний контакт. Зв'язок між входом і виходом контакту згідно рис 2.6.10:

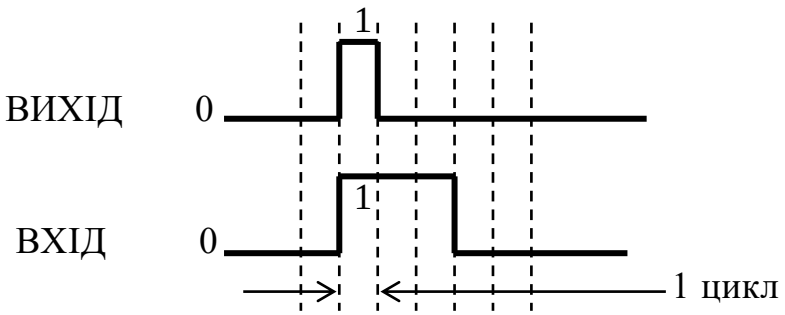

Рисунок 2.6.10 – Графік функції контакту "вихід позитивний одновібратора"

- контакт “**вихід негативний одновібратора”** – цифровий вихідний контакт. Зв'язок між входом і виходом контакту згідно рис. 2.6.11:

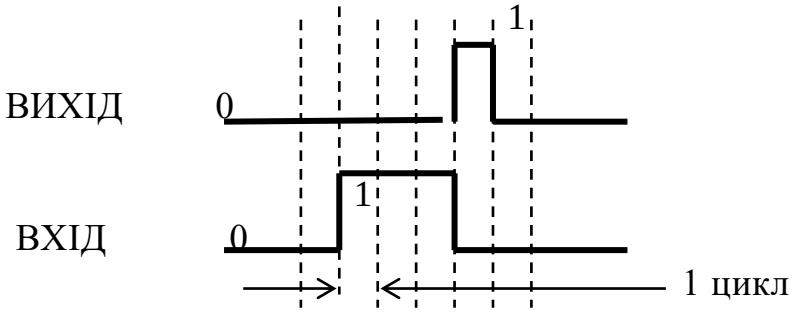

Рисунок 2.6.11 – Графік функції контакту "вихід негативний одновібратора"

- контакт “**вихід одновібратора”** – цифровий вихідний контакт. Зв'язок між входом і виходом контакту згідно рис. 2.6.12:

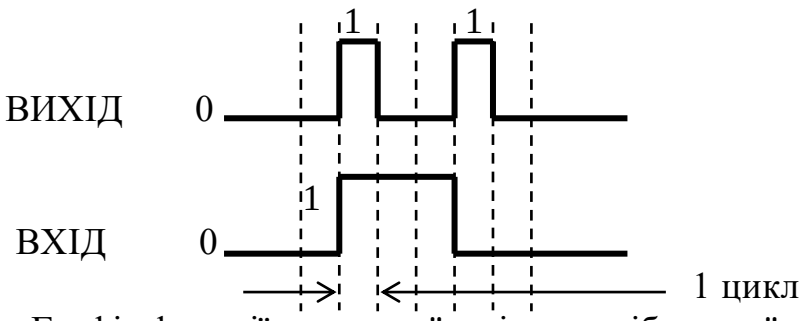

Рисунок 2.6.12 – Графік функції контакту "вихід одновібратора"

Багато стандартних функцій (функціональні блоки) включені в застосовувані програми і об'єднані в різні бібліотеки. У кожній бібліотеці розміщені функціональні блоки, використання яких дозволяє заощадити багато часу і зусиль при рішенні стандартних завдань.

Функціональні блоки (FBK) – представлений блоками бібліотек програми, у тому числі:
"Бібліотека **Standart** – представлена FBK визначення фронтів вхідних сигналів і установки/скидання стану:
F_TRI() – визначає задній фронт сигналу, у випадку, коли вхідний сигнал переходить із "1" в "0", F_TRI видає імпульс тривалістю 1 цикл завдання;
R_TRI() – визначає передній фронт сигналу, у випадку, коли вхідний сигнал переходить із "0" в "1", F_TRI видає імпульс тривалістю 1 цикл завдання;
RF_TRI() – визначає будь-який фронт сигналу, у випадку, коли вхідний сигнал переходить із "1" в "0" або з "0" в "1", RF_TRI видає імпульс тривалістю 1 цикл завдання;
RS() - RS-трігер, при вхідному сигналі "1" на вході S – на виході встановлюється "1", при вхідному сигналі "1" на вході R – на виході встановлюється "0". Вхід R є більше пріоритетним;
SR() - RS-трігер, при вхідному сигналі "1" на вході S – на виході встановлюється "1", при вхідному сигналі "1" на вході R – на виході встановлюється "0". Вхід S є більше пріоритетним;
TOF – блок вимикання із затримкою. На вході PT – задається час у мілісекундах, на вході IN – вхідний сигнал. У випадку переходу вхідного сигналу з "1" в "0", TOF відраховує заданий час і на виході Q установлює "0". На виході ET видається відлік часу.
Діаграма сигналів блоку вимикання згідно рис. 2.6.13.

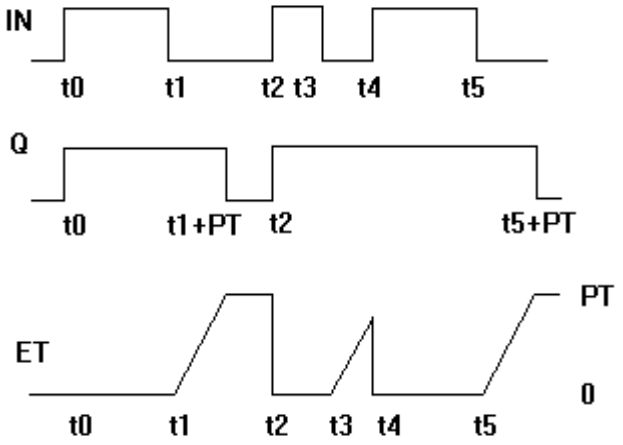

Рисунок 2.6.13 – Діаграма функції блоку TOF

TON – блок включення із затримкою. На вході PT задається час у мілісекундах, на вході IN – вхідний сигнал. У випадку переходу вхідного сигналу з "0" в "1", TON відраховує заданий час і на виході Q установлює "1". На виході ET видається відлік часу. Діаграма сигналів блоку включення згідно рис. 2.6.14.

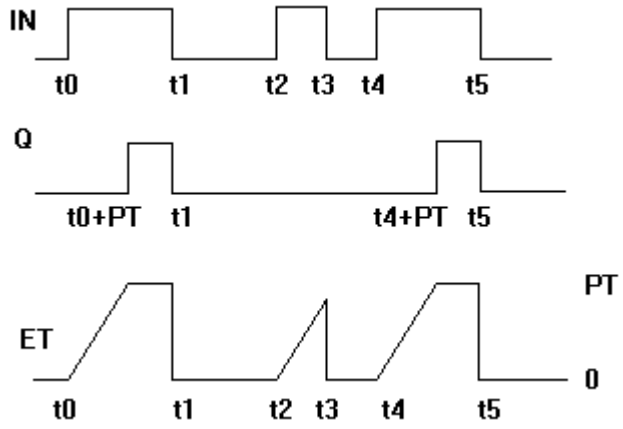

Рисунок 2.6.14 – Діаграма функції блоку TON

TP – блок генерації імпульсів заданої тривалості. На вході PT задається час у мілісекундах, на вході IN – вхідний сигнал. У випадку переходу вхідного сигналу з "0" в "1", TP на виході встановлює "1" і відраховує заданий час, після чого на виході Q установлюється "0". На виході ET видається відлік часу. Діаграма сигналів блоку згідно рис. 2.6.15.

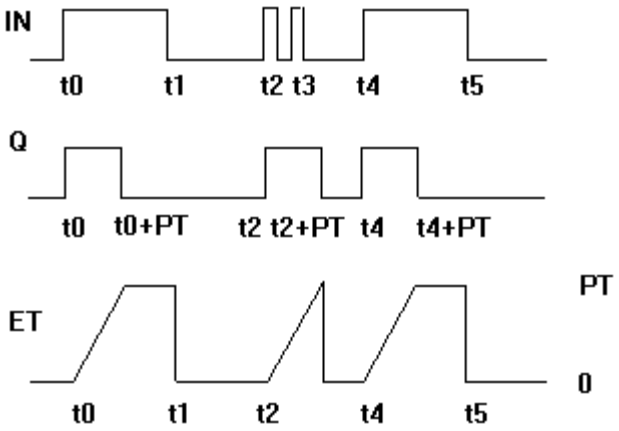

Рисунок 2.6.15 – Діаграма функції блоку TP

TOF_10ms, TON_10ms, TP_10ms - блоки, аналогічні блокам TOF, TON, TP, при цьому час задається в 10-мілісекундних інтервалах;
CTD() - лічильник, що зменшує. На вхід CD надходять імпульси, на вхід PV надходить стартове значення. Якщо на вхід LOAD подати "1" то лічильник ініціалізується, і на виході CV буде число, що надходить на вхід PV.
Якщо на вході CD сигнал переходить із "0" в "1", то значення на виході CV зменшиться на одиницю. Коли значення на виході CV досягне "0" на виході Q виставиться "1";
CTU() - лічильник, що збільшує. На вхід CU надходять імпульси, на вхід PV надходить значення для порівняння. Якщо на вхід RESET подати "1", то лічильнику становиться у вихідне (нульове) положення й на виході CV буде "0". Якщо на вході CU сигнал переходить із "0" в "1", то значення на виході CV збільшується на одиницю. Коли значення на виході CV досягне величини, що подано на вхід PV – на виході Q виставиться "1";
CTUD() – універсальний лічильник. Містить у собі всі входи й виходи, які присутні на 2 попередніх лічильниках за винятком виходу Q, замість нього присутні два виходи: QU – установлюється в "1", коли значення на виході CV досягне значення, що подається на вхід PV, і вихід QD - установлюється, коли значення на виході CV досягне "0".

Приклад згідно рис 2.6.16: при натисканні кнопки (k1) через 100 мілісекунд з'явиться сигнал (k), який може керувати яким-небудь устаткуванням.

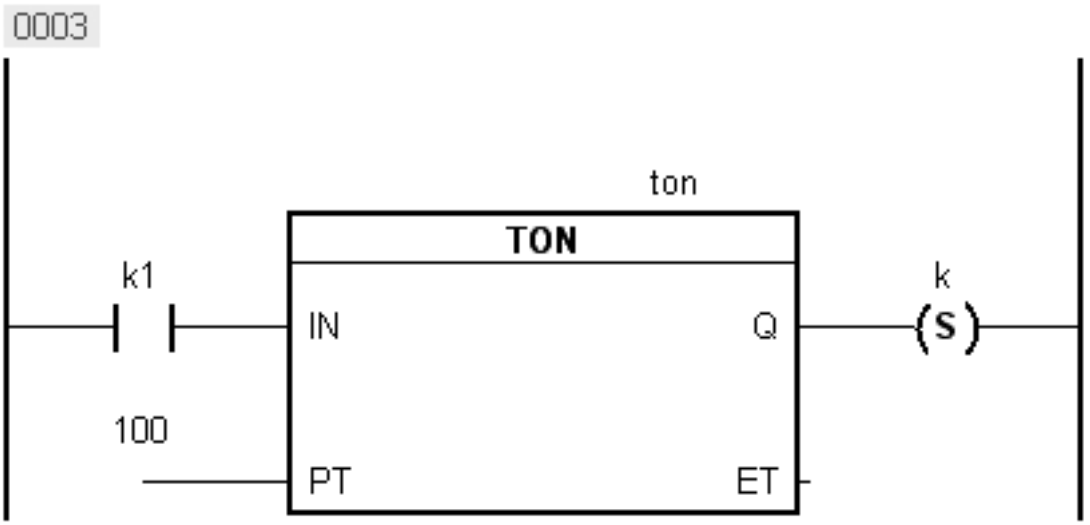

Рисунок 2.6.16 – Схема реалізації функції лінії затримки

Приклад згідно рис. 2.6.17: при натисканні кнопки **r** лічильник "обнуляється", при надходженні кожного сигналу на вхід (k2) лічильник на виході CV збільшиться на одиницю. Коли кількість імпульсів на виході CV досягне заданого значення входу PV (20) – вихід Q установиться в логічну "1". Так можна робити відлік якихось подій і після певного їхнього числа вмикати необхідний механізм.

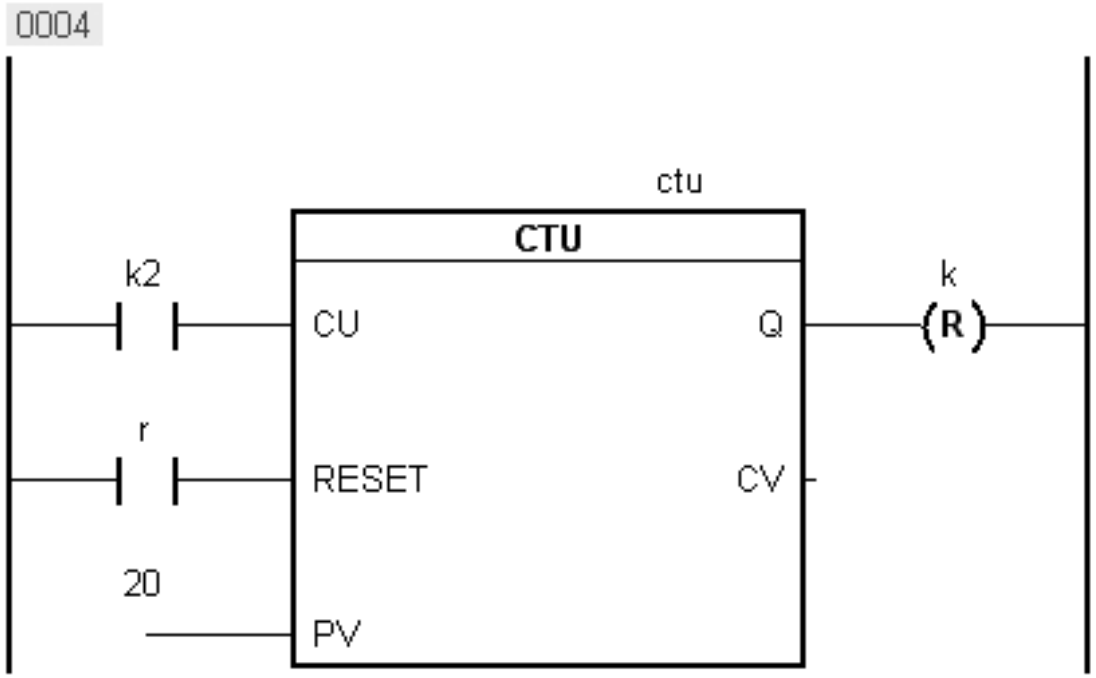

Рисунок 2.6.17 – Схема реалізації лічильника

Бібліотека **Operator** представлена блоками: порівняння, логічними і арифметичними.
Блоки порівняння мають 2 вхідних параметри будь-якого типу і один вихідний ("0" або "1"). Обидва вхідних параметри повинні бути одного типу.
У випадку виконання умови для даного блоку вихідний параметр установлюється в логічну "1". Є наступні блоки порівняння:

EQ() - '=' (визначає, якщо обоє вхідних значення еквівалентні);
GE() - '>=';
GT() - '>';
LE() - '<=';
LT() - '<';
NE() - '<>'.

Логічні блоки мають 2 вхідних параметри будь-якого типу і один вихідний того ж типу. Обоє вхідних параметри повинні бути одного типу. На виході видається результат даної логічної операції:

AND() – логічне "І";
NOT() – логічна інверсія;
OR() – логічне "АБО".

Арифметичні блоки мають 2 вхідних параметри будь-якого типу і один вихідний того ж типу. Обоє вхідні параметри повинні бути одного типу. На виході видається результат даної арифметичної операції:

ADD() - додавання
SUB() - віднімання
DI() - ділення
MUL() - множення

Приклад застосування блоку порівняння GT(): для контролю температури згідно рис. 2.6.18:

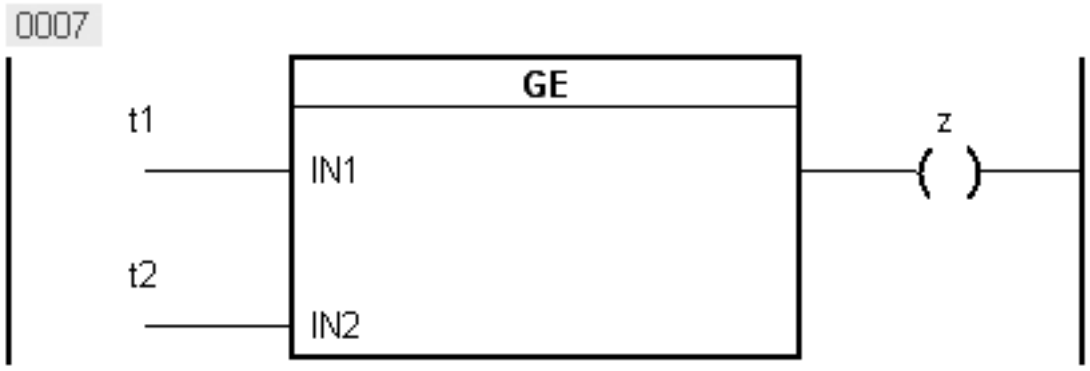

Рисунок 2.6.18 – Схема реалізації блоку порівняння

Змінна t1 містить поточну температуру, змінна t2 містить оптимальну температуру. У випадку, коли поточна температура стане вище або дорівнює оптимальної – на виході (для змінної z) установиться логічна "1".

Зазначені вхідні й вихідні елементи беруть участь у формуванні ланцюгів. Ланцюг - це з'єднання елементів, у тому числі хоча-б одного символу присвоєння релейних схем, що представляють собою послідовність команд, виконуваних програмувальними контролерами. Таким чином, ланцюг – це елемент програмування релейних мов і елемент булевих мов. Формат ланцюга може бути фіксованим або змінним. У першому випадку ланцюг повинен складатися з певного числа компонентів. Символ присвоювання дозволяє ідентифікувати рядок програми і призначати проміжну змінну або відповідний вихід. Логічні компоненти надаються входам, проміжним змінним або виходам. При використанні мови релейно-контактних схем програма вичерчує на екрані дисплея в певному масштабі схеми, що легко читаються. Програми виконуються по ланцюгах. Кожний ланцюг спрацьовується зліва праворуч. Це дає можливість програмістові зробити якийсь певний сигнал пріоритетним у порівнянні з іншими, наприклад для забезпечення безпеки. Формат ланцюга програмувальних контролерів містить різне число логічних компонентів. Для компенсації обмеженого числа компонентів у ланцюзі передбачається можливість злиття ланцюгів. Для того щоб запрограмувати контролер під конкретне завдання керування, однієї лише мови релейно-контактних символів або булевих рівнянь недостатньо. Необхідно виконати цілий ряд функцій програмування, таких як ідентифікація, ініціалізація, керування циклом, реалізація обчислень, відображення інформації і т.п. Ці функції виконуються за допомогою базової мови B&R Automation Studio (AS). При створенні проекту AS може автоматично розпізнавати апаратні засоби, а також призначати структуру апаратних засобів. У вікні проекту конфігурується продуктивність системи відповідно до вимог проекту, використовуючи конфігурацію програмного забезпечення. Програмування включає наступні етапи:

1. Оголошення змінних аналогових і цифрових входів і виходів. Імена змінних можуть мати довжину до 32 символів.
2. Створення типу об'єкта. Діалогове вікно "Insert Object" дозволяє вибирати наступні типи об'єктів:

- "Cyclic Object" – циклічний об'єкт, обробляється через певні інтервали з використанням певного часу циклу. Операційна система контролює час циклу, щоб упевнитися, що він не перевищений. Можна встановити пріоритети, привласнивши циклічним об'єктам різні ресурси (різні часи циклу). Циклічним об'єктам, які виконують важливі, критичні з погляду часу виконання завдання, призначають ресурс із більш коротким часом циклу, тому вони виконуються частіше, ніж об'єкти, яким призначені ресурси з більш довгим часом циклу. Циклічні об'єкти включають – циклічні об'єкти, таймерні об'єкти, об'єкти переривання, об'єкти особливого стану;
- "Data Object" - об'єкти даних – це модулі, які використаються для зберігання даних;

- "System Object" – модулі B&R – можуть передаватися на контролер як системні об'єкти. Системні об'єкти включають бібліотеки або об'єкти з додатковими системними функціями. Багато із цих об'єктів завантажуються автоматично програмою B&R Automation Studio і не повинні вставлятися вручну;
- Бібліотека - колекція стандартних функцій, що представлена в програмі B&R Automation Studio як модуль B&R. Кожна бібліотека містить функції, які допоможуть заощадити час і зусилля при рішенні стандартних проблем. При використанні, функція з бібліотеки переноситься в проект і встановлюється в ході завантаження;
    - "Advanced Object" - складні об'єкти (осі ЧПУ, профілі автоматизованого керування і т.д).

  Після того, як тип ресурсу був визначений у діалоговому вікні "Insert Object", відкривається діалогове вікно "New Object", щоб визначити ім'я об'єкта, тип об'єкта і ресурс. Можна визначити пріоритети для окремих циклічних об'єктів, привласнивши об'єктам різні ресурси. Різний час циклу може бути встановлене для кожного циклічного ресурсу і кожного таймерного ресурсу. Всі об'єкти, яким був привласнений певний ресурс, обробляються один раз протягом тимчасового циклу, певного для цього ресурсу. Об'єктам, які виконують важливі, критичні з погляду часу виконання завдання, призначається ресурс із більш коротким часом циклу, тому вони виконуються частіше, ніж об'єкти, яким привласнений ресурс із більш довгим часом циклу.

3 Створення програми – програмування заданої електричної схеми в редакторі ступеневих діаграм: розміщення цифрових входів, малювання сполучних ліній, розміщення цифрових виходів, перевірка і оптимізація, збереження, компіляція і передача на контролер.

Основним завданням програмного забезпечення є послідовне виконання технологічних операцій по заданому тимчасовому графіку (циклограмі) з використанням заданого набору дискретних команд і аналогових, що відповідають заданому виконанню контролера. Для виконання цього завдання програмне забезпечення реалізує наступні функції:

- програмування (введення технологічної програми, контроль введення даних);
- керування;
- індикацію ходу виконання техпроцесів

3 МАТЕРІАЛИ, ПРИЛАДИ, УСТАТКУВАННЯ

3.1 Лабораторний практикум проводиться із застосуванням наступного устаткування: персональний комп'ютер, контролер B&R 2003

4 ВКАЗІВКИ ПО ТЕХНІЦІ БЕЗПЕКИ

При виконанні лабораторного практикуму виконувати вимоги по охороні праці відповідно до інструкції № 328 "Вимоги по охороні праці при роботі з комп'ютерною технікою"

5 ПОРЯДОК ПРОВЕДЕННЯ ЛАБОРАТОРНОГО ПРАКТИКУМУ

5.1 Перший запуск програми: B&R Automation Studio (AS)

5.1.1 Включити комп'ютер і блок живлення контролера.

5.1.2 Зробити запуск програми AS. Після старту програми на екрані відображається заставка, потім відкривається вікно програми (див. рис. 5.3), що містить:

- головне меню з функціями згідно додатка А;
- панель інструментів для забезпечення швидкого доступу до різних команд і функцій;
- вікно повідомлень для показу інформації компілятора;
- рядок стану для короткої довідковоїі нформації команд меню або значків панелі інструментів, про процедури редагування, з'єднання між комп'ютером і контролером.

5.2 Установити з'єднання між контролером і программатором (комп'ютером) Для чого зробити настройку online-з'єднання командою "Options..." з меню "Tools". У вікні "Online-Configuration" установити значок "SERIAL", вікні "Properties" – значок "COM-2".

5.3 Автоматичне розпізнавання апаратних засобів контролера шляхом запуску команди "New Project..." з меню "File"

При цьому відкриється вікно "New Project Wizard". У цьому діалоговому вікні необхідно ввести ім'я проекту (Name), наприклад - "AS_QS", і шлях до каталогу проекту (Path) - "C:\PROJECTS". Відключить опцію "Upload hardware From target!".

Якщо каталогу не існує, то нажавши кнопку ви одержите запит, чи створити його.Відповідайте на цей запит, нажавши "Enter". Потім програма AS виконує автоматичне розпізнавання апаратних засобів. Хід виконання показується в діалоговому вікні. По закінченню відкрити наступне вікно командою "Next". У цьому діалоговому вікні показані всі розпізнані раніше дані, поряд з модулями контролера, знайденими в ході розпізнавання апаратних засобів (номер моделі модуля процесора (CPU) і блоку живлення (PS)). Ще раз перевірте дані, можна вертатися і виправляти дані – команда "Back". Розпізнавання апаратних засобів закінчується відкриттям проекту – команда "Finish". При цьому відкривається вікно проекту "Project Window" – основа кожного проекту. Воно складається із двох частин (рис. 5.3):

- у лівій частині вікна дається короткий огляд апаратних засобів, використаних у проекті – конфігурацію апаратного забезпечення. Якщо є апаратні

розходження, то модулі в поточному проекті, які відрізняються від модулів на контролері, відзначаються в такий спосіб:

 У цьому місці на контролері перебуває інший модуль! Ви можете замінити модуль, відзначивши його і викликавши контекстне меню (правою кнопкою маніпулятора).

 У цьому місці на контролері немає ніякого модуля!
- у правій частині вікна приводиться додаткова інформація і параметри настроювання для модуля, відзначеного в лівій частині вікна, кожний модуль має вкладку "Software", що дозволяє одержати доступ до конфігурації програмного забезпечення.

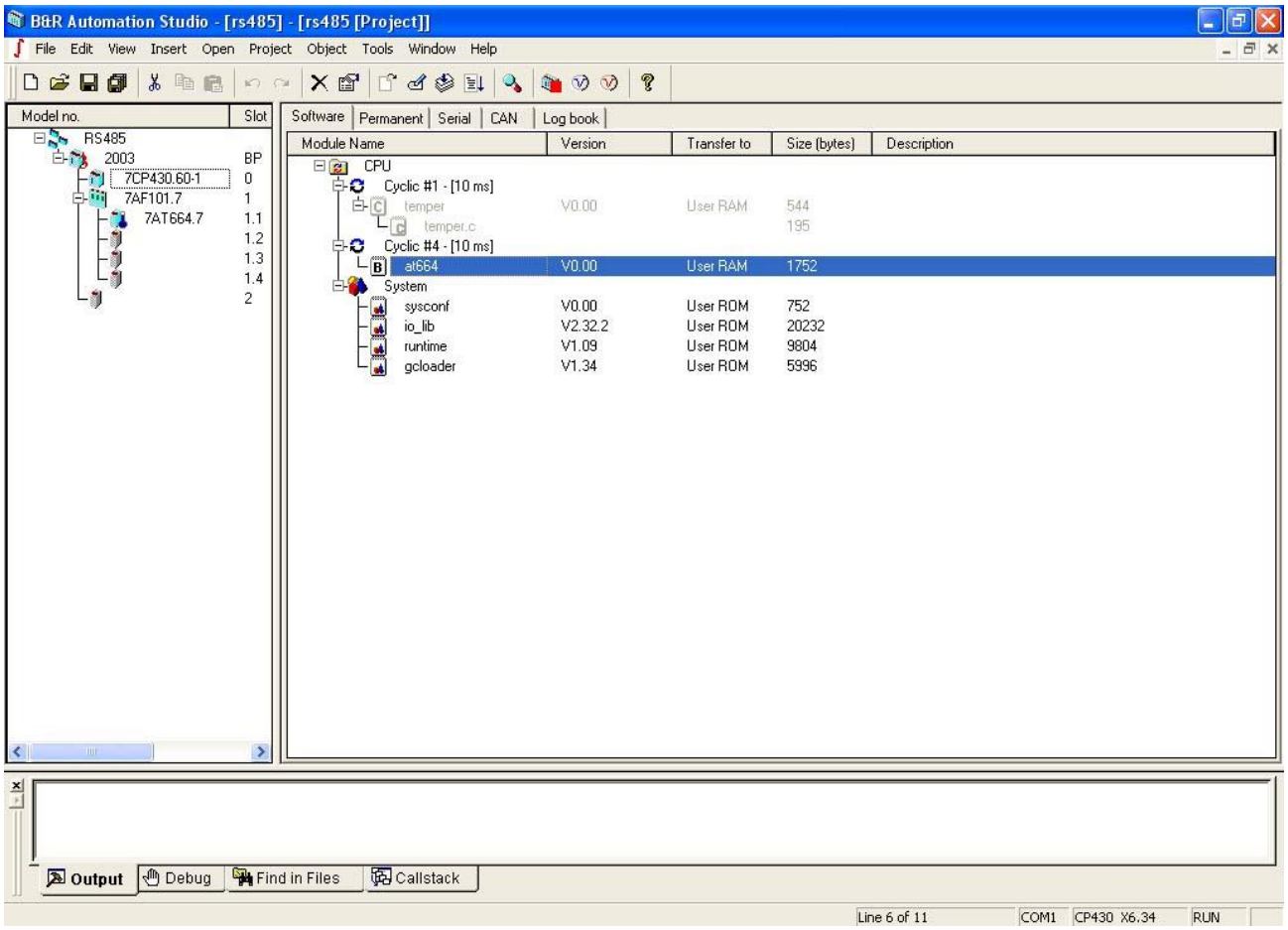

Рисунок 5.3 – Вікно проекту “Project Window”

Програма дозволяє робити вибір необхідної конфігурації системи і модулів відповідно до технічного завдання

5.4 Створення проекту з вибором необхідної конфігурації системи

Задамося вихідними даними для виконання роботи. Наприклад, необхідно створити проект для реалізації електричної схеми згідно рис. 5.4.

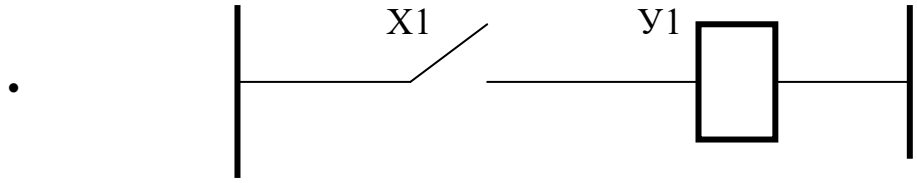

Рисунок 5.4 – Схема проекту електрична

де: X1 - вимикач,

У1 - об'єкт керування.

Напруга живлення схеми - =24 В, максимальний струм – не більше 2А.

5.5 Запустити команду "New Project" з меню "File". У діалоговому вікні, що відкрилося, ввести ім'я проекту і шлях до каталогу (див. п 5.3), дезактивувати опцію "Upload hardware from target". Відкрити наступне вікно командою "Next". Зробити вибір необхідного модуля процесора (CPU) зі списку можливих моделей модуля процесора, перейти до наступного вікна командою "Next". Перевірити правильність обраних даних, установлюваних у ході виконання програми, можна вертатися і виправляти дані.

Відкрити вікно проекту командою "Finish".

5.6 Зробити вибір додаткових модулів контролера (вхідних і вихідних): активізувавши значок "кошика" модуля процесора, виконати команду "Insert" – у відкритому вікні "Select Module" вибрати необхідні модулі, наприклад: модуль 7D1435.7 - вхідний, 7D0435.7 - вихідний.

5.7 Зробити оголошення змінних (X1, У1). Для змінної X1 – у лівій частині проекту активізувати символ модуля 7D1435.7 (цифровий вхідний модуль), при цьому в правому розділі вікна будуть показані вкладки "I/O" і "Description". Щоб привласнити ім'я змінної X1 першому цифровому входу на модулі 7D1435.7 потрібно встановити курсор у стовпець "PV Name" у першому рядку списку і натиснути клавішу "Пробіл", ввести ім'я змінної - X1. Елемент приймається після того, як натиснута клавіша "Enter". Ім'я змінної може мати довжину до 32 символів. Стовпець "Remark" містить короткий функціональний опис вводу – виводу на відзначеному модулі.

Цифровий вихід У1 – оголошується аналогічно для вихідного модуля 7D0435.7.

5.8 Створити циклічний об'єкт у такий спосіб:

- у лівій частині вікна проекту виберіть модуль процесора;
- виберіть вкладку "Software" у правому розділі вікна і запустіть команду "New Object" у меню "Insert". У показаному діалоговому вікні вибрати тип ресурсу "Cyclic" і підтвердіть опцію, нажавши кнопку "Next";
- визначити ім'я об'єкта, типу об'єкта і ресурсу в діалоговому вікні "New Object". У поле "Name" ввести ім'я для нового об'єкта, у поле "Type" для типу об'єкта - "Ladder Diagram", у поле "Resource" для ресурсу - "Cyclic#1 - [10ms]".

Можна визначити пріоритети для окремих циклічних об'єктів, привласнивши об'єктам різні ресурси. Різний час циклу може бути встановлене для кожного циклічного ресурсу і кожного таймерного ресурсу. Всі об'єкти, яким був

привласнений певний ресурс, обробляються один раз у ході тимчасового циклу, певного для цього ресурсу. Об'єктам, які виконують важливі, критичні з погляду часу виконання завдання, призначається ресурс з коротким часом циклу, тому вони виконуються частіше, ніж об'єкти, яким привласнений ресурс з більш довгим часом циклу.

Командою "Finish" циклічний об'єкт вводиться в конфігурацію програмного забезпечення, при цьому відкривається вікно редактора ступеневих діаграм (LAD).

5.9 Розробка програми в редакторі LAD для схеми, наведеної на рис. 5.4:

- розмістити цифровий вхід у редактор, активізувавши символ відповідного цифрового входу на інструментальній панелі або команду "Contact" у меню "Insert" і вибравши "Normally Open Contact" з показаного підменю.
- Привласнити контакту цифровий код X1, ввівши ім'я змінної X1, Елемент приймається після натискання клавіші "ENTER" або "Пробіл". З показаного списку виберіть "X1" і встановіть командою "Add". Список містить тільки змінні, які сумісні з типом даних нового цифрового входу.
- Розмістити сполучні лінії, використовуючи комбінацію клавіш "ALT + КУРСОР" або за допомогою кнопок інструментальної панелі.
- Розмістити цифровий вихід у редактор, активізувавши символ відповідного цифрового виходу на інструментальній панелі або команду "Coil" у меню "Insert" і вибравши "Coil" з показаного підменю
- Привласнити контакту цифровий код У1, ввівши ім'я змінної У1, елемент приймається після натискання клавіші "ENTER" або "Пробіл". З показаного списку виберіть "У1" і встановіть командою "Add". Список містить тільки змінні, які сумісні з типом даних нового цифрового виходу.
- Виконати перевірку і оптимізацію програми на наявність відкритих сполучних ліній або інших помилок активізувавши кнопку в інструментальній панелі або клавішу "ENTER".
- Зберегти програму, активізувавши відповідний значок на інструментальній панелі або нажати комбінацію клавіш "CTRL + S" або вибрати команду "Save" з меню "File". При збереженні програма знову перевіряється і оптимізується.
- Закрити редактор LAD командою "Close" у меню "File" або кнопкою в правому верхньому куті вікна редактора.

5.10 Виконати компіляцію і передачу розробленої програми в користувальницький RAM контролера командою "Transfer to Target" з меню "Project": за допомогою цієї команди весь проект компілюється, а потім передається в певну область пам'яті контролера. Хід і остаточний стан процедури компіляції показується у вікні повідомлень. Щоб уникнути помилок, у ході передачі AS перевіряє структуру апаратних засобів цільової системи і всіх попередньо встановлених програмних об'єктів. Виявлені проблеми будуть чітко зазначені, щоб не виникало додаткових помилок.

Після того, як проект був успішно переданий, видається наступне повідомлення згідно з рис. 5.10:

Рисунок 5.10 – Повідомлення про завершення передачі даних

Таким чином, проект відповідно до завдання здійснений у вигляді обраної апаратної частини і програмного забезпечення (робочої програми в користувальницькій пам'яті контролера). Цифровий вихід установлюється або скидається відповідно до логічного стану цифрового входу.

5.11 Зробити вибір апаратних засобів і розробити програми LAD відповідно до отриманого завдання.

6 СКЛАД ЗВІТУ

До складу звіту входять наступні дані:

1. склад промислового контролера B&R 2003;
2. характеристика модулів контролера B&R 2003;
3. обґрунтування вибору модулів контролера;
4. розроблені програми і структурну схему контролера відповідно до завдання;
5. висновки.

7 КОНТРОЛЬНІ ПИТАННЯ ДЛЯ САМОСТІЙНОЇ ПЕРЕВІРКИ І КОНТРОЛЮ ПІДГОТОВКИ СТУДЕНТІВ ДО РОБОТИ

1. Принципи побудови децентралізованої системи керування.
2. Етапи побудови МПС.
3. Вибір конфігурації апаратної частини контролера – основні принципи.
4. Основні етапи програмування на базі проблемно-орієнтованої мови, характеристика етапів.
5. Характеристика контактів панелі інструментів LAD.
6. Характеристика функціональних блоків LAD.

8 РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Б. В. Костров, В. Н. Ручкин. Мікропроцесорні системи. Навчальний посібник, ТЕХБУХ, М, 2005, 208 стор.

А. А. Сазонов. Мікропроцесорне керування технологічним устаткуванням мікроелектроніки. Навчальний посібник, М. Радіо і зв'язок, 1998, 264 стор.

Э. В. Евреинов. Цифрова і обчислювальна техніка. Підручник, М., Радіо і зв'язок, 1991, 464 стор.

9 ДОДАТОК А

Структура меню вікна проекту "Project Window":

1 File - файл:

- New Project (Ctrl+N) – створення нового проекту
- Open Project (Ctrl + D) – відкриття існуючого проекту
- Close – закриття поточного редактора
- Close Project – закриття поточного проекту
- Save (Ctrl + S) – збереження поточного файлу
- Save All – збереження всіх змінених файлів
- Page Setup – установки сторінки для роздруківки
- Print (Ctrl + P) - друк
- Export – експорт проекту
- Import - імпорт
- 1C:\projects\...\proi_001.GDM – список недавно відкритих файлів
- Exit – вихід із програми

2 Edit - редагування:

- Undo – скасування останньої операції
- Fiedo – повторення останньої операції
- Cut (Ctrl + X) – вирізка виділеного тексту
- Copy (Ctrl + C) – копіювання виділеного тексту
- Delete (DEL) – видалення виділеного тексту
- Go To (Ctrl + G) – перехід до рядка...
- Symbol (Space) – редагувати символ
- Variable – редагувати змінну
- Label – редагувати мітку
- Description – редагувати опис
- Propertien (Alt + Enter) – змінити установки, редагувати змінні

3 View - вид:

- Toolbars – активувати інструментальну панель
- Status Bar – активувати статусний список
- Qutput – активувати вікно повідомлень
- Type – активувати тип в LAD
- Scope – активувати область в LAD
- Remark – активувати коментар в LAD
- Monitor – активувати режим монітора
- Init Subroutine – перевірити програму ініціалізації
- Zoom – вибір масштабу відображення

4 Insert – вставити в ступеневі діаграми:

- Network – вставити нову мережу
- Contact – вставити новий контакт

- Coil – вставити новий одновібратор
- Jump – вставити новий перехід
- Function – вставити функціональний блок
- Column – вставити стовпець
- Row – вставити рядок

5 Open - відкрити:

- Data Types – відкрити адміністратор типів даних
- Library Manager – відкрити адміністратор бібліотек
- Declaration – відкрити оголошення змінних
- Watch – відкрити монітор PV (Watch)
- Trace – відкрити трассировщик

6 Project - проект:

- Build – компілювати проект (тільки змінені файли)
- BuildAll – компілювати всі файли
- Transfer To Target – завантажити проект у цільову систему
- Open Scheme – відкрити схему
- Save Scheme – зберегти схему
- Invert Scheme – інвертувати схему
- Reset Scheme – скинути схему
- Services – відкрити підменю "Services"
- Settings – загальні установки проекту

7 Project: Service - проект: сервіс:

- Stop Target – зупинити контролер
- Warm Restart – скидання з теплим перезавантаженням
- Cold Restart – скидання з холодним перезавантаженням
- Diagnostics –перехід у діагностичний режим
- Transfer Operating System – передача операційної системи
- Clear Memory – очистити пам'ять

8 Objekt - об'єкт:

- Rensme – перейменування об'єкта
- Debug – активація режиму налагодження (Сі завдання)
- Transferto – перенос в... (RAM, USER, ROM, FIXRAM...)
- Start – запустити завдання
- Stop – зупинити завдання
- Force – примусова установка значень змінних
- Force All Off – відключити примусову установку для всіх змінних у поточному моніторі
- Erase From Target – видалити об'єкт із контролера
- Upload From Target – переслати об'єкт із контролера
- Disable – дезактивувати об'єкт

9 Tools - інструменти:

- Options – установка програми

10 Window - вікно:

- Cascade – каскадне розташування вікон
- Tile – мозаїчне розташування вікон
- Arrange Icons – упорядкувати значки
- 1syein. GDM [Project] – список вікон

11 Help - довідка:

- Help – інтерактивна довідка
- How To Use Help – як користуватися довідкою по Automation Studio
- About Automation Studio – про Automation Studio