Ельперін І.В. Промислові контролери. Частина 2.: Конспект лекцій до вивчення дисц. для студ. спец. 6.092500 "Автоматизоване управління технологічними процесами" і "Комп'ютерно-інтегровані технологічні процеси і виробництва" напряму підготовки 0925 "Автоматизація і комп'ютерно-інтегровані технології". – К.: НУХТ, 2012. -106 с.

Рецензент **А.П.Ладанюк**, доктор техн.наук.

І.В.Ельперін, кандидат техн. наук.

Видання подається в авторській редакції

© І.В.Ельперін, 2012
© НУХТ, 2012

1. МІКРОПРОЦЕСОРНІ КОНТРОЛЕРИ MODICON TSX

1.1. Загальна структура технічних засобів

Мікропроцесорні контролери фірми Schnaider Electric останнім часом почали широко використовуватись вітчизняними інжиніринговими фірмами при розробці та впровадженні мікропроцесорних систем управління в різних галузях промисловості.

Фірма Schnaider Electric – це міжнародний концерн, який об'єднав провідних виробників засобів автоматизації США, Франції та Німеччини (Modicon, Telemecanique, Merlin Gerin, Square) і є одним із лідерів у галузі засобів промислової автоматизації. Компанія Schnaider Electric (SE) позиціонує себе на першому місці у світі щодо виробництва засобів автоматизації в галузі розподілу електроенергії і на третьому місці з виробництва ПЛК. Створення такого концерну не просто структурно об'єднало окремі фірми й заводи, а й дало можливість об'єднати інтелектуальні зусилля окремих фірм і започаткувати розробку та впровадження нового покоління управляючої мікропроцесорної техніки. Його технічне та програмне забезпечення відповідає міжнародному стандарту МЕК 1131. У зв'язку з цим на прикладі вивчення технічного та програмного забезпечення мікропроцесорних контролерів цієї фірми можна отримати знання для подальшого швидкого вивчення апаратних та програмних засобів ПЛК інших фірм, які підтримують стандарт МЕК 1131.

Концерном Schnaider Automation випускається велика гама ПЛК, які умовно можна поділити на дві гілки, які незважаючи на те, що вони об'єднані загальною ідеологією розвитку та модернізації і з кожним роком наближаються одна до одної, мають деякі особливості. До першої гілки можна віднести ПЛК Modicon TSX Nano, Modicon TSX Micro, Modicon TSX Premium, а до другої – Modicon TSX Quantum.

Гама контролерів SE починається з мініатюрного рісо-контролера Zelio Logic, який являє собою інтелектуальне реле. Основне призначення цих реле: керування електронікою будов (освітлення, ескалатори, вентиляція, контроль доступу, захисні бар'єри, температура у приміщеннях); керування простим обладнанням (насоси, компресори, маленькі преси); керування обладнанням теплиць, пекарень, змішувачів; попередня обробка сигналів для інших схем. Zelio Logic має: таймери; лічильники; аналогові компаратори; системний годинник для запуску або зупинки дій у визначений день і час; розташовані на лицьовій панелі невеличкий дисплей і кнопки для оперативного спостереження за виконанням програми й оперативного втручання в процес керування.

Modicon TSX Nano призначений для управління простими об'єктами, які мають невелику кількість (в основному, дискретних) входів-виходів, підходить для реалізації задач, які вимагають короткого часу робочого циклу (< 1 мс). Це моноблочний контролер,

у якому зосереджені всі функції. Має пам'ять EEPROM, годинник реального часу, порти комунікацій, швидкі лічильники (10 кГц), імпульсні і ШІМ виходи, аналогові входи-виходи, таймери.

Програмований контролер Twido, який фірма почала випускати у 2002 р., займає середню позицію між ПЛК TSX Nano і Micro і призначені для управління як совсім простими об'єктами, так і об'єктами, які вимагають використання сотен дискретних і десятків аналогових входів-виходів. Контролер має дві моделі Compact і Modular, які розрізняються як конструктивно, так і функціональними можливостями. В основному, для ПЛК Modular пропонується велика гама модулів дискретних і аналогових входів-виходів, комунікаційних модулів, дисплеїв оператора, карт розширення пам'яті і годинника реального часу і тін.

Modicon TSX Micro призначений для управління об'єктами, для роботи з якими достатньо кілька сотень дискретних входів-виходів та до 40 аналогових входів-виходів у різних галузях промисловості. Компактний і модульний TSX Micro має невеликий час робочого циклу і велику ємність пам'яті. Він має широкий діапазон модулів входів-виходів, включаючи віддалені модулі входів-виходів, забезпечує високу швидкість роботи лічильників, високу точність реалізації задач позиціювання, високу точність вимірювання аналогових сигналів, має широкі комунікаційні можливості.

ПЛК Modicon TSX Premium – це потужний контролер, який може працювати з тисячами дискретних та сотнями аналогових входів-виходів. Він має розподілену фізичну структуру; широкий діапазон модулів входів-виходів; багатозадачну операційну систему і потужну систему обробки переривань; вбудовані функції контурів регулювання; широкий діапазон модулів управління позиціюванням і рухом; швидкісні лічильники й модуль вимірювання ваги, вбудовану мережу Fipio, тепле резервування (Warm Standby).

ПЛК Modicon TSX Quantum оптимізований для розв'язання задач керування з централізованою структурою. Забезпечує зв'язок із верхнім рівнем для великих централізованих і розподілених структур, має високопродуктивні процесори для реалізації складних задач керування, захист пам'яті, високу експлуатаційну готовність, можливість нанесення захисного покриття для роботи у агресивних середовищах, забезпечує гаряче резервування (Hot Standby) – заміна модулів при працюючому контролері, має шину віддаленого вводу-виводу. Для програмування контролера використовується потужне програмне забезпечення Concept із бібліотекою функцій керування процесами з пов'язаними з ними супервізорними графічними шаблонами.

Така гама різних за потужністю контролерів класична і для інших фірм, які випускають ПЛК, що дає можливість вибрати оптимальний варіант для створення системи управління різними за складністю об'єктами і забезпечити найкращі показники функціональні можливості – вартість.

1.2. Мікропроцесорний контролер Modicon TSX Micro

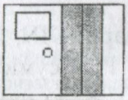
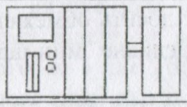
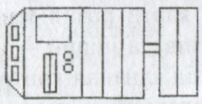
1.2.1 Фізична структура

Контролери TSX Micro являють собою проектно-компоновані вироби, для яких кількість і склад модулів вибирається залежно від розв'язуваної задачі керування і необхідних характеристик вхідних і вихідних сигналів. Конструктивно різні моделі ПЛК TSX Micro будуються з використанням одного з двох базових шасі, в які інтегровані мікропроцесорний модуль і слоти для встановлення модулів. Базові шасі відрізняються функціональними можливостями процесорного модуля, напругою живлення ПЛК, кількістю вільних слотів (2 або 3), а також можливістю під'єднання до нього міні-шасі розширення із двома додатковими слотами.

Випускається п'ять моделей ПЛК TSX Micro (табл.1.1): TSX 3705, TSX 3708, TSX 3710, TSX 3721, TSX 3722.

Таблиця 1.1

Типи моделей TSX Micro

Фізична структура	Тип моделі	Модуль, встановлений у базове шасі	Напруга живлення
	TSX 3705 028DR1	TSX DMZ 28DR	~ 100... 220 В
	TSX 3708 056DR1	TSX DMZ 28DR TSX DMZ 28DR	~ 100...220 В
	TSX 3710 128DT1 TSX 3710 128DR1 TSX 3710 128DTK1 TSX 3710 164DTK1 TSX 3710 028AR1 TSX 3710 028DR1	TSX DMZ 28DT TSX DMZ 28DR TSX DMZ 28DTK TSX DMZ 64DTK TSX DMZ 28AR TSX DMZ 28DR	=24 В =24 В =24 В =24 В ~ 100 ...220 В ~ 100 ...220 В
	TSX 3721 101 TSX 3721 001		=24 В ~ 100...220 В
	TSX 3722 101 TSX 3722 001	(інтегровані у про-цесорний модуль 8 аналогових входів і 1 аналоговий вихід, а також два 10 кГц лічильника)	=24 В ~ 100 – 220 В

До кожного базового шасі контролера входять: блок живлення на =24 В постійного струму (VDC) чи ~110...240 В змінного струму (VAC), процесорний модуль з енергонезалежною оперативною пам'яттю RAM і

системою резервного збереження інформації на базі FLASH ERROM, дисплейний блок, термінальний порт і кнопка перезапуску ПЛК.

На рис. 1.1 показано загальний вигляд ПЛК TSX Micro 3710. До складу базового шасі ПЛК TSX 37-10 також входить один з повноформатних дискретних модулів, тип якого і визначає назву контролера (табл.1.1). Отже, на базовому шасі залишається один вільний слот для встановлення модулів. Під'єднання до ПЛК міні-шасі розширення дає можливість використовувати ще два слота.

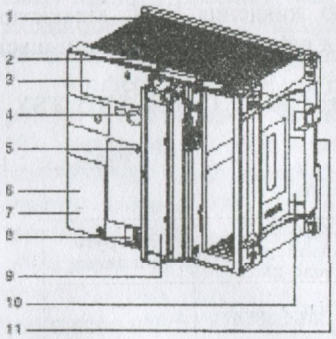


Рис. 1.1. Загальний вигляд ПЛК TSX 37-10

1 – шасі з двома слотами для модулів, вбудоване джерело живлення, модуль процесора з пам'яттю; *2* – отвори для монтажу; *3* – дисплейний блок; *4* – термінальний порт (TER); *5* – кнопка RESET (перезапуск); *6* – кришка для доступу до контактів джерела живлення; *7* – ярлик, заповнюваний при зміні батареї; *8* – кришка доступу до батарейки; *9* – модуль з 28 чи 64 входами-виходами, розміщений у першому слоті (базова конфігурація); *10* – кришка доступу до рознімання для під'єднання шасі розширення; *11* – вузол кріплення на DIN-рейці.

Термінальний порт RS485 із 8-штирьковим рознімом стандарту DIN, дає змогу під'єднати до ПЛК термінал програмування FTX чи PC- сумісну ПЕОМ, пульт програмування, принтер, а також під'єднувати ПЛК до шини UNI-TELWAY.

Базове шасі TSX 3705 відрізняється від шасі TSX 3710 тим, що на ньому відсутній рознім для під'єднання міні-шасі розширення. Крім того, базове шасі цього ПЛК живиться напругою ~ 100...240 В і до його складу входить тільки один тип повноформатного дискретного модуля TSX DMZ 28DR, який має 16 входів та 12 релейних виходів. Тобто цей ПЛК має досить обмежену кількість входів-виходів і до нього не може бути під'єднане міні-шасі розширення.

На відміну від моделі TSX 37-10, базове шасі TSX 37-21 і TSX 37-22 (рис.1.2) має два додаткових слоти для встановлення карт формату PCMCIA. Один з них призначений для встановлення зовнішньої карти розширення пам'яті RAM чи зовнішньої FLASH ERROM (до 128 Кслів), а інший – для комунікаційної карти з метою забезпечення можливості під'єднання контролера до різноманітних шин та мереж: асинхронний послідовний зв'язок, чи дуплекс напівдуплекс, UNI-TELWAY, MODBAS/JBAS і FIPWAY).

Крім того, на процесорному модулі встановлені два функціонально ідентичних термінальних порти RS485 із 8-штирковим рознімом стандарту DIN і позначених TER і AUX. Відмітною їхньою особливістю є те, що порт

AUX не може використовуватись для під'єднання пристроїв, що не мають власного джерела живлення шини. Тому, як правило, порт TER використовується для під'єднання пульта програмування FTX 117, кабелю-адаптера для під'єднання ПЕОМ і т.д., а порт AUX може використовуватися для під'єднання пристроїв людино-машинного інтерфейсу (операторських панелей), принтерів і т.ін.

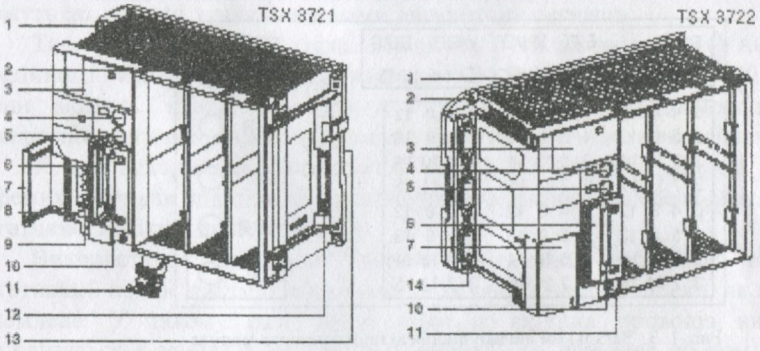


Рис. 1.2. Загальний вигляд ПЛК TSX 37-21 і TSX 37-22

1 – шасі з трьома слотами для модулів, вбудоване джерело живлення, процесор з його пам'яттю; *2* – отвір для монтажу ПЛК; *3* – дисплейний блок; *4* – термінальний порт (TER); *5* – порт для під'єднання людино-машинного інтерфейсу (AUX); *6* – слот для під'єднання карти розширення пам'яті. Якщо карта не встановлена, цей слот закривається кришкою, що не повинна видалятися. Її видалення викликає зупинку ПЛК; *7* – кришка для доступу до контактів джерела живлення; *8* – ярлик, заповнюваний у разі заповнення батарейки; *9* – контакти джерела живлення; *10* – слот для комунікаційної карти; *11* – кришка для доступу до резервної батарейки; *12* – рознімання для під'єднання шасі, зазвичай закритий кришкою, що видаляється; *13* – вузол кріплення на DIN-рейці; *14* – рознімання для під'єднання вбудованих аналогових і рахункових каналів.

Базове шасі TSX 3708 відрізняється від шасі TSX 3721/22 тим, що на ньому відсутній рознім для під'єднання міні-шасі розширення, тобто до нього, як і для TSX 3705 не може бути під'єднане міні-шасі розширення. Базове шасі цього ПЛК також живиться тільки напругою ~ 100/240, але до його складу входить не один, а два дискретних модулі TSX DMZ 28DR.

На кожному процесорному модулі ПЛК розміщено дисплейний блок, який відображає інформацію, необхідну в основному для діагностики працездатності ПЛК, його модулів і окремих каналів. Для цього він має (рис.1.3):

- 5 світлодіодних індикаторів (RUN, TER, I/O, ERR і BAT) для відображення стану ПЛК;
- 3 блоки по 32 індикатори за допомогою яких відображається:
- стан усіх входів-виходів у базовому шасі, шасі розширення, а також стан віддалених входів-виходів;

- діагностику модулів і окремих каналів на них;
- стан констант, внутрішніх і системних слів чи бітів у двійковому чи шістнадцятковому форматі;
- кнопки та індикатори (BASE, EXT, R I/O, WRD, DIAG, "64", "16") зміни режиму відображення інформації на дисплейному блоці.

Кількість вільних слотів у різних моделях контролера TSX Micro визначає можливу кількість входів-виходів із якою може працювати ПЛК.

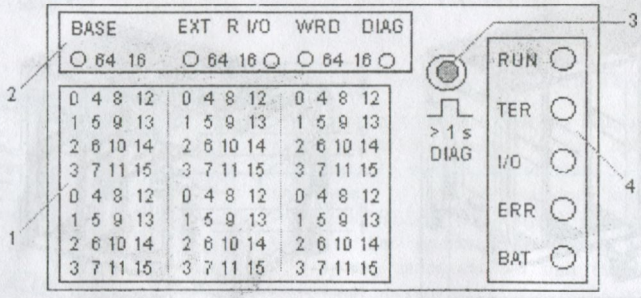


Рис. 1. 3. Загальний вигляд дисплею процесорного модуля

Міні-шасі розширення TSX RKZ 02 дає можливість додати до ПЛК два слота та приєднується до базового шасі за допомогою розніму і фіксується гвинтами. Якщо в шасі встановлені аналогові модулі чи дискретні модулі з релейними виходами до нього додатково повинно бути під'єднано джерело живлення =24 В постійного струму, тому що потужності джерела живлення базового шасі в цьому разі буде недостатньо.

Усі типи модулів, які використовуються для TSX Micro, можна умовно розбити на кілька груп: дискретні, аналогові, рахункові, комунікаційні і спеціалізовані (наприклад, модуль безпеки, модулі вимірювання ваги тощо). Для кращої адаптації до вимог користувача модулі дискретного входу-виходу доступні в двох форматах: стандартному, що займає слот цілком (2 настановні місця) і напівформатному, що займає тільки одне настановне місце, тобто в один слот можуть бути встановлені два напівформатних модулі. Усі інші типи модулів – напівформатні.

1.2.2 Модулі дискретних входів-виходів

Для ПЛК TSX Micro розроблена широка номенклатура дискретних модулів входів-виходів, що постійно поповнюється новими модифікаціями. При виборі модулів для реалізації конкретної задачі керування об'єктом користувач повинен аналізувати їхні функціональні можливості, технічні характеристики і конструктивні особливості. Це дає можливість підібрати конфігурацію контролера, яка найбільш повно відповідає вимогам до системи керування, забезпечуючи при цьому оптимальну її вартість.

Основними характеристиками дискретних модулів є кількість каналів входів-виходів і типи сигналів, які можна до них під'єднувати.

Аналізуючи вибір модуля за кількістю каналів, необхідно враховувати, що існує два типи дискретних модулів: модулі змішаних каналів входів-виходів, у яких на одному модулі присутні як вхідні, так і вихідні дискретні канали, й модулі для під'єднання винятково чи вхідних, чи вихідних каналів. Змішані модулі мають стандартний формат, а вхідні чи вихідні модулі можуть бути як стандартного, так і напівформатного формату.

При виборі модуля дискретного модуля треба враховувати, що вони можуть працювати з різними типами дискретних сигналів.

Так, вхідні дискретні сигнали можуть бути як змінного (VAC), так і постійного (VDC) струму різної напруги ±24 В VDC та ~100...120 В VAC. Вони можуть також забезпечувати різні схеми під'єднання вхідних дискретних сигналів, а саме за схемами позитивної або негативної логіки.

У разі використання позитивної логіки під'єднання загальні проводи окремих датчиків під'єднані до позитивного полюса джерела живлення, а негативної логіки – до негативного.

Використання негативної логіки під'єднання небажане, тому що негативний полюс джерела живлення, виходячи з вимог безпеки, як правило, заземлене. У такому разі, якщо один із вхідних проводів випадково від'єднається й увійде в контакт із "землею", це призведе до помилкового спрацьовування відповідного дискретного входу.

Вихідні дискретні канали можуть мати транзисторні чи релейні виходи, що у свою чергу можуть комутувати сигнали змінної чи постійної напруги зі струмом комутації від 0,1 до 3 А.

У технічних характеристиках враховується також ступінь захисту входів-виходів від перешкод і перевантажень. Так, усі канали дискретних модулів входів-виходів є ізольованими, а транзисторні дискретні виходи захищені від перевантажень і короткого замикання швидкодіючими електронними схемами. Релейні виходи додаткового захисту не мають.

Основною конструктивною особливістю модулів, крім їхнього формату, є спосіб під'єднання до них вхідних чи вихідних ланцюгів. Одні модулі обладнані спеціальними клемними колодками для під'єднання "під гвинт" (рис. 1.4), а інші – спеціальними рознімними з'єднувачами стандарту HE10 (рис. 1.5).

Дуже зручним є те, що клемні колодки є знімними. Тобто, якщо виникає потреба замінити модуль, спочатку знімається клемна колодка без порушення монтажу проводів, що до неї під'єднуються, потім замінюється модуль і на нього встановлюється колодка зі збереженим монтажем.

Під'єднання до модулів із клемною колодкою здійснюється безпосередньо із закріпленням проводів "під гвинт", якими обладнані всі клеми. До кожної клеми можуть бути під'єднані або неізольовані кінці проводів, перерізом від 0,28 до 1,5 мм^2, або проводи обладнані спеціальними наконечниками відкритого чи закритого типу.

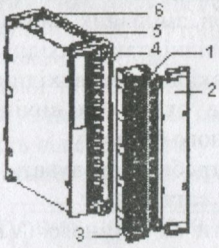


Рис. 1.4. Конструкція модуля з гвинтовою клемною колодкою

1 – знімна клемна колодка для безпосереднього під'єднання "під гвинт" сигналів входів-виходів; *2* – кришка клемної колодки, що також використовується для розміщення на них ярликів і міток; *3* – знімна кришка, що обмежує доступ до гвинтів і утримує їх на місці; *4* – блокувальний важіль, що використовується для витягування й фіксації модуля в слоті; *5* – електронна плата; *6* – металевий корпус

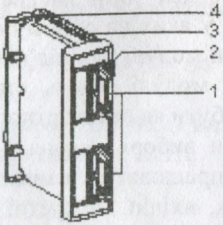
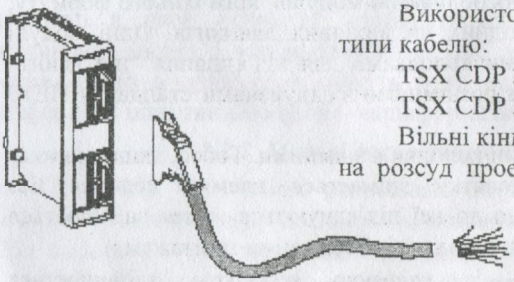


Рис.1.5. Конструкція модуля з рознімами стандарту HE10

1 – рознімами типу HE10, захищені кришками. Їх кількість залежить від типу модуля; *2* – блокувальний важіль, що використовується для витягування й фіксації модуля в слоті; *3* – електронна плата; *4* – металевий корпус.

При під'єднанні до модулів із НЕ 10 з'єднувачами використовуються спеціальні кабелі, обладнані відповідними рознімами. При цьому можливі два варіанти під'єднання.

У першому варіанті використовується кабель в який з одного кінця має НЕ 10 розніма, за допомогою якого кабель під'єднується до модуля, а з іншого кінця розташовані кодовані кольором вільні кінці проводу (Рис.1.6).



Використовуються два стандартних типи кабелю:

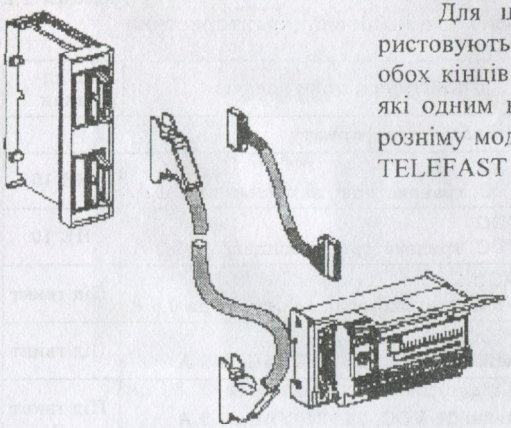
TSX CDP 301 – завдовжки 3 м;

TSX CDP 301 – завдовжки 5 м.

Вільні кінці кабелю під'єднуються на розсуд проектувальника системи.

Рис. 1.6. Під'єднання кабеля з вільними кінцями

У другому варіанті до модуля з НЕ 10 з'єднувачами під'єднуються пристрої для швидкого монтажу TELEFAST 2, до клемників якого вже безпосередньо під'єднуються датчики і виконавчі пристрої.



Для цього під'єднання використовуються два види кабелів, що з обох кінців облаштовані рознімами, які одним кінцем під'єднуються до розніму модуля, а іншим до розніму TELEFAST 2 (рис.1.7).

Рис. 1.7. Під'єднання за допомогою блоків TELEFAST

Блоки швидкого монтажу TELEFAST 2 використовуються не тільки як вилучені клемники, але і виконують ряд додаткових функцій: індикація стану сигналу, додатковий захист від перевантажень, перетворення одного виду сигналу на інший, підсилення потужності сигналів за рахунок установлення в них безконтактних або звичайних релейних елементів. Існує велика кількість різноманітних типів блоків швидкого монтажу, які мають позначення ABE-7..... Замість крапок проставляються літерні й цифрові позначення, що відповідають додатковим функціям, які виконує блок.

Основні характеристики для різних типів дискретних модулів наведені у табл. 1.2.

1.2.3 Модулі аналогових входів-виходів

Для роботи з аналоговими сигналами в ПЛК TSX Micro використовуються модулі аналогових входів-виходів, які є напівформатними модулями із гвинтовою клемною колодкою. Вони можуть бути розташовані у всіх слотах ПЛК за винятком першого, де завжди повинен бути встановлений повноформатний дискретний модуль.

Аналогові модулі TSX Micro відрізняються призначенням, кількістю каналів, характеристиками й діапазоном вимірювання. Для роботи з аналоговими вхідними сигналами від датчиків використовуються три типи модулів: TSX AEZ 801, TSX AEZ 802 та TSX AEZ 414.

Модулі TSX AEZ 801 та TSX AEZ 802 дають можливість під'єднувати вісім аналогових сигналів з одною загальною точкою під'єднання.

Модуль TSX AEZ 801 працює з аналоговими сигналами у вигляді напруги в діапазоні ± 10 В або 0...10 В, який можна вибирати для кожного каналу окремо в процесі конфігурування модуля.

Таблиця 1.2

Типи дискретних модулів і їхні основні характеристики

Позначення модуля	Кількість каналів	Характеристики каналів	Під'єднання
Модулі стандартного формату			
TSX DMZ 64DTK	32 входа 32 виходa	24 VDC 24 VDC, транзисторні, захищені, до 0,1 А	НЕ 10
TSX DMZ 28DTK	16 входів 12 виходів	24 VDC 24 VDC, транзисторні, захищені, до 0,5 А	НЕ 10
TSX DMZ 28DT	16 входів 12 виходів	24 VDC 24 VDC, транзисторні, захищені, до 0,5 А	Під гвинт
TSX DMZ 28DR	16 входів 12 виходів	24 VDC релейні 24 VDC, 24...240VAC,до 3 А	Під гвинт
TSX DMZ 28AR	16 входів 12 виходів	110 VAC релейні 24 VDC, 24...240VAC,до 3 А	Під гвинт
TSX DEZ 32D2	32 входи	24 VDC	Під гвинт
TSX DSZ 32T2	32 виходи	24 VDC, транзисторні, захищені, до 0,5 А	Під гвинт
TSX DSZ 32R5	32 виходи	Релейні 24 VDC, 24...240VAC, до 2 А	Під гвинт
Напівформатні модул.			
TSX DEZ 12D2K	12 входів	24 VDC	НЕ 10
TSX DEZ 12D2	12 входів	24 VDC	Під гвинт
TSX DEZ 08A4	8 входів	100/120 VAC	Під гвинт
TSX DEZ 08A5	8 входів	200/240 VAC	Під гвинт
TSX DSZ 08T2K	8 виходів	24 VDC, транзисторні, захищені, до 0,5 А	НЕ 10
TSX DSZ 08T2	8 виходів	24 VDC, транзисторні, захищені, до 0,5 А	Під гвинт
TSX DSZ 08R5	8 виходів	релейні 24 VDC, 24...240VAC, до 3 А	Під гвинт
TSX DSZ 04T22	4 виходa	24 VDC, транзисторні, захищені, до 2 А	Під гвинт

Модуль TSX AEZ 802 працює з аналоговими сигналами у вигляді струму в діапазоні 0...20 мА або 4...20 мА, який також можна вибирати для кожного каналу окремо в процесі конфігурування модуля.

Ці вхідні модулі виконують такі функції:

- сканування вхідних каналів за допомогою безконтактного мультіплексування і збір даних;
- аналого-цифрове перетворення вхідних вимірювань (12 бітне).

У додаток до цих функцій процесором ПЛК виконується наступна обробка:

- вхідний контроль перевищування рівня сигналу;
- фільтрація сигналів;
- перетворення вхідних вимірювань у формат користувача.

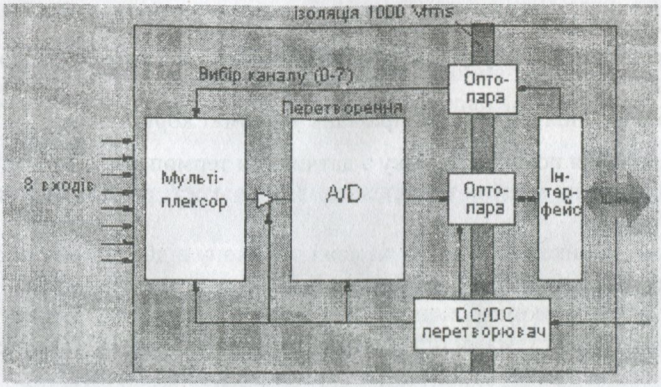


Рис. 1.8. Функціональна схема модулів TSX AEZ 801 та TSX AEZ 802

Час опитування всіх каналів модуля залежить від типу циклу опитування, яке вибирається при конфігуруванні модуля.

За **нормального циклу** час опитування не залежить від кількості каналів, які використовуються і становить 32 мс (4 мс на один канал).

За **швидкого циклу** опитування переглядаються тільки під'єднані канали, якщо навіть вони під'єднані не один за одним. Тоді час перегляду модуля обчислюється за формулою:

$$T = 4 \text{ мс} \cdot N,$$ де N – кількість підключених каналів.

Для будь-якого вибраного діапазону контролюється вихід за його межі. Система діагностики сигналізує про перевищення діапазону, якщо сигнал вийде за його межі на величину, що становить 5% від повного діапазону шкали. Наприклад, для діапазону 0…10 В, система діагностики виявить помилку, коли вимірювальний сигнал вийде за межі –0,5 В або 10,5 В.

Вхідний аналоговий модуль **TSX AEZ 414** є багатодіапазонним модулем з 4 диференційними гальванічно розподіленими каналами. Залежно від вибору, який зроблений у процесі конфігурування кожного каналу модуля, окремо може бути вибраний один з діапазонів:

- термопари із градуюванням B, E, J, K, L, N, R, S або U;
- термометри опору Pt100 або Ni1000 із можливістю під'єднання їх за 2-, 3- або 4- провідною схемою;
- 0…20 мА, 4…20 мА, ± 10 В, 0…10 В, 0…5 В (0-20 мА з зовнішніми шунтами) або 1…5 В (4…20 мА з зовнішніми шунтами).

Модуль виконує такі функції:

- вибір діапазону вхідного сигналу;
- перегляд вхідних каналів за допомогою мультиплексування;
- аналого-цифрове перетворення (16-бітне) вхідних сигналів;
- вхідний контроль перевищування рівня сигналу для вибраного діапазону;

- лінеаризація для термометрів опору Pt100 або Ni1000;
- лінеаризація та внутрішня або зовнішня компенсація холодного спаю для термопар;
- перетворення вхідних вимірювань у формат користувача;
- визначення помилок зв'язку з датчиками термопар.

Час опитування модуля не залежить від кількості під'єднаних каналів і складає 520 мс.

Контроль за виходом сигналу за межі діапазону відбувається аналогічно, як і для модулів TSX AEZ 801 та TSX AEZ 802.

Під'єднання термометрів опору до модуля TSX AEZ 414 виконується за загальноприйнятими схемами під'єднання за 2-, 3- і 4- провідними схемами.

У табл. 1.3 наведені діапазони вимірювання термометрів опору.

Таблиця 1.3

Діапазони термометрів опору

Діапазон	Нижня межа	Верхня межа
Pt100	− 200 °C (− 328 °F)	+ 850 °C (+ 1562 °F)
Ni1000	− 60 °C (− 76 °F)	+ 110 °C (+ 230 °F)

При роботі з термопарами, діапазони вимірювання яких наведені у табл. 1.4, компенсація холодного спаю виконується модулем. При цьому температура холодного спаю може вимірюватись внутрішнім датчиком, встановленим у модулі або дистанційно зовнішнім датчиком Pt100, який під'єднується до 0 каналу.

Таблиця 1.4

Діапазони термопар

Діапазон	Нижня межа	Верхня межа
B	0 °C (32 °F)	+ 1802 °C (+ 3276 °F)
E	− 270 °C(− 454 °F)	+ 812 °C (+ 1493 °F)
J	− 210 °C (− 346 °F)	+ 1065 °C (+ 1949 °F)
K	− 270 °C (− 454 °F)	+ 1372 °C (+ 2502 °F)
L	− 200 °C (− 328 °F)	+ 900 °C (+ 1652 °F)
N	− 270 °C (− 454 °F	+ 1300 °C (+ 2372 °F)
R	− 50 °C (− 58 °F)	+ 1769 °C (+ 3216 °F)
S	− 50 °C (− 58 °F)	+ 1769 °C (+ 3216 °F)
T	− 270 °C (− 454 °F)	+ 400 °C (+ 752 °F)
U	− 200 °C (− 328 °F)	+ 600 °C (+ 1112 °F)

Як аналогові вихідні модулі використовуються модулі TSX ASZ 401 та TSX ASZ 200.

Модуль TSX ASZ 401 має 4 аналогових виходи з загальною точкою, на яких вихідний сигнал може змінюватись у діапазоні ± 10 В з навантаженням до 2 ком без використання зовнішнього джерела живлення. Цей модуль виконує функції (рис. 1.9):

- обробка цифрових значень, які обчислює ПЛК у результаті виконання програми користувача;
- виявлення і обробка несправностей, що виникають у роботі модуля та встановлення значень виходів у стан, що відповідає аварійному режиму;
- цифро-аналогове перетворення.

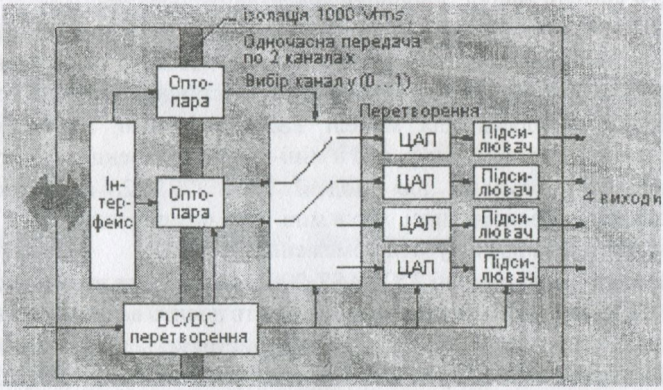


Рис. 1.9. Функціональна схема модуля аналогового виходу TSX ASZ 401

Прикладна програма користувача повинна забезпечити зміну значення аналогового сигналу в діапазоні від –10000 до +10000. У цьому разі на виході модуля сигнал буде змінюватись від –10 В до +10 В. Якщо значення, які розраховуються ПЛК, будуть менші–10000 або більші +10000, фізичний сигнал на виході модуля буде обмежений значеннями –10 В або +10 В.

Виходи модуля модифікуються парами в кінці кожного циклу задачі, з якою вони пов'язані. Час опитування модуля – 400 мс.

Модуль TSX ASZ 200 має 2 аналогових виходи з загальною точкою, працює без зовнішнього джерела живлення і має такі діапазони:

- ± 10 В з мінімальним навантаженням до 1 кОм;
- 0…20 мА з максимальним навантаження 600 Ом;
- 4…20 мА з максимальним навантаження 600 Ом.

Ці модулі виконують такі функції:

- вибір діапазону для кожного виходу: напруга або струм;
- обробка цифрових значень, які обчислює ПЛК у результаті виконання програми користувача;
- виявлення і обробка несправностей, що виникають у роботі модуля та встановлення значень виходів у стан, що відповідає аварійному режиму;
- цифро-аналогове перетворення (11 біт + знак).

Прикладна програма користувача повинна забезпечити зміну значення аналогового сигналу в діапазоні від –10000 до +10000 для діапазону ± 10 В та від 0 до +10000 для діапазонів 0…20 мА і 4…20 мА (для діапазону 4…20 мА

0 відповідає значенню 4 мА). При виході значень ПЛК за межі діапазону вони **обмежуються межовими значеннями**.

Крім модулів аналогових входів-виходів, у моделі TSX 3722 можна використовувати інтегровані у процесорний модуль 8 аналогових входів (8-бітових) і 1 аналоговий вихід, які використовуються у випадках, коли не вимагається високої точності обробки аналогових сигналів.

Модулі аналогових входів-виходів можуть бути встановлені у всі позиції TSX Micro, за винятком першого слота базового шасі. Максимальна кількість аналогових модулів, які можуть бути встановлені у ПЛК, обмежена:

- двома модулями для моделі TSX 3705/08/10, які можуть бути встановлені або в базовому шасі, або в міні-шасі розширення;
- чотирма модулями для моделі TSX 3721/22, які можуть бути встановлені або в базовому шасі, або в міні-шасі розширення.

При цьому треба враховувати обмеження:

- максимум два модулі TSX ASZ 200 можуть бути встановлені у базовому шасі ПЛК із-за того, що вони споживають досить велику потужність;
- якщо у модулі TSX AEZ 414 використовується внутрішня компенсація, цей модуль повинен бути встановлений у найменших позиціях ПЛК;
- якщо аналоговий модуль встановлений у міні-шасі розширення, під'єднаного до базового шасі, яке живиться змінним струмом, міні-шасі розширення повинно бути забезпечено живленням 24 В постійного струму.

1.2.4. Модулі лічильників

У контролерах TSX Micro використовуються різноманітні лічильні канали, функціональні можливості яких залежать як від типу каналу, так і від параметрів його конфігурування. Є три типи лічильних каналів:

- лічильні канали, які організовані на входах дискретного модуля, встановленного у першому слоті ПЛК;
- лічильні канали, інтегровані у процесорний модуль ПЛК;
- лічильні каналі на спеціалізованих модулях.

Ці канали відрізняються насамперед максимальною дозволеною частотою лічильних імпульсів. Так, для лічильників, організованих на входах дискретного модуля, ця частота складає 500 Гц, для інтегрованих лічильників – 10 кГц, а для спеціалізованих лічильних модулів – 40 кГц.

На різноманітних лічильних каналах можна реалізувати функції прямого, зворотного і реверсивного підрахунку імпульсів.

Функція прямого лічильника (рис. 1.10) передбачає підрахунок імпульсів від 0 до передуставленого значення, яке відоме як значення "уставки".Діапазон прямого лічильника від 0 до +16777215 (24 біти + знак). Вихідним станом для прямого лічильника є скидання його в 0. Це може бути реалізовано шляхом подачі спеціальних сигналів на додаткові входи лічильника або автоматично, при досягненні лічильником значення уставки

або програмно. Плинне значення лічильника неперервно порівнюється із двома порогами (0 і 1), які встановлюються користувачем.

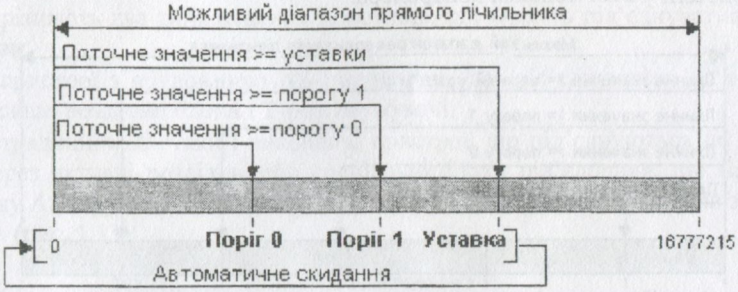


Рис. 1.10. Прямий лічильник

Функція зворотного лічильника (рис. 1.11) передбачає відлік імпульсів від попередньо встановленого значення (в діапазоні від 0 до +16777215) з сигналізацією того, що плинне значення досягло або стало менше 0. Діапазон зворотного лічильника від –16777215 до +16777215. Вихідним станом зворотного лічильника є встановлення його в передуставлене значення, перехід у яке організується так само, як і для прямого лічильника.

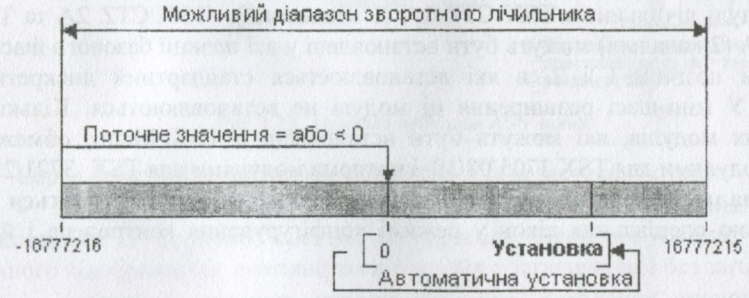


Рис. 1.11. Зворотний лічильник

Функція реверсивного лічильника (рис. 1.12) поєднує в собі функції прямого і зворотного лічильників. При цьому відлік ведеться від попередньо встановленого значення всередині діапазону реверсивного лічильника: від –16777215 до +16777215. Для цього типу лічильників, крім передуставленого значення, можуть бути встановлені значення верхньої і нижньої уставки, а також два порога (0 і 1).

Датчики, що генерують імпульси можуть, бути різного типу (безконтактні або з сухими контактами), але їх виходи повинні бути сумісні з входами модулів, до яких вони під'єднуються. Як правило для логічної 1 - це сигнал постійного струму 5…24 В.

Лічильні канали, що організовані на перших чотирьох входах дискретного модуля, встановленого у першому слоті ПЛК, можуть бути використані у всіх моделях контролера.

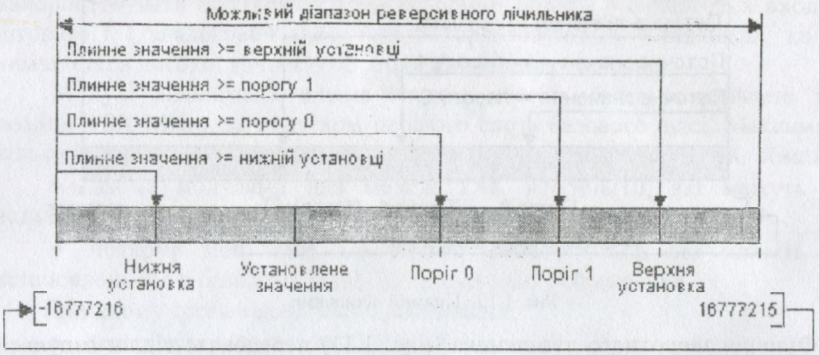


Рис. 1.12. Реверсивний лічильник

Лічильні канали, інтегровані у процесорний модуль ПЛК використовуються тільки у моделі TSX 3722. За аналогією з інтегрованими аналоговими сигналами, сигнали лічильників можуть бути під'єднані безпосередньо до рознімів на модулі процесора або через блок TELEFAST.

Модулі лічильників TSX CTZ 1A (1 канальний), TSX CTZ 2A та TSX CTZ 2AA (2 канальні) можуть бути встановлені у всі позиції базового шасі за винятком позицій 1 і 2, в які встановлюється стандартний дискретний модуль. У міні-шасі розширення ці модулі не встановлюються. Кількість лічильних модулів, які можуть бути встановлені в TSX Micro, обмежені двома модулями для TSX 3705/08/10 і чотирма модулями для TSX 3721/22.

Канали лічильників не програмуються, а конфігуруються за допомогою спеціальних вікон у режимі конфігурування контролера і його модулів.

1.2.5 Asi-шина

Кількість пристроїв, які може обслуговувати TSX Micro, може бути суттєво розширено за рахунок використання **AS-i шини.** Це 0-ий рівень польової шини, що призначена для передачі дискретної інформації між основним пристроєм, що працює в режимі ведучого (майстра) і датчиками і виконавчими пристроями, що виступають у ролі ведених.

Основними елементами AS-i шини є:

- ведучий (майстер) шини;
- ведені пристрої (датчики і виконавчі пристрої);
- спеціальне джерело живлення.

Фізично всі елементи, що утворять AS-i шину, з'єднані між собою спеціальним двопровідним кабелем.

У ролі ведучого шини виступає напівформатний модуль **TSX SAZ 10**, що керує всіма операціями обміну даних на AS-і шині. Цей модуль повинен бути встановлений тільки в 4 позиції базового шасі TSX Micro.

Розрізняють два типи ведених пристроїв, що можуть під'єднуватися до AS-і шини:

- пристрої з вбудованою AS-і функцією, що під'єднуються до шини через пасивні розділові блоки і Т-розгалужувачі;
- традиційні датчики і виконавчі пристрої, що під'єднуються до AS-і шини через активні розділові або спеціальні блоки під'єднання, що мають вбудовану AS-і функцію і забезпечують зв'язок традиційних пристроїв з AS-і шиною. (рис. 1.13).

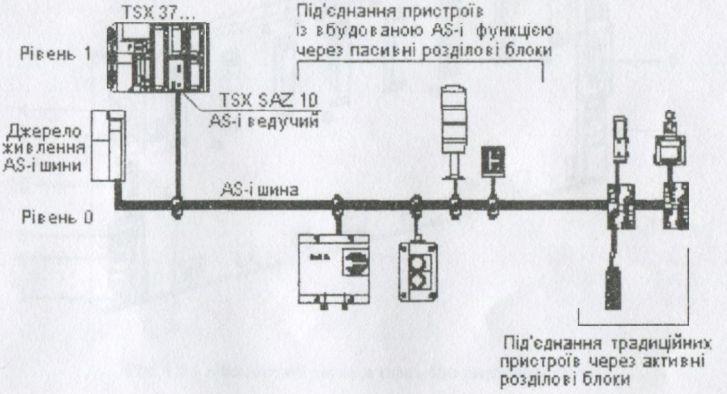


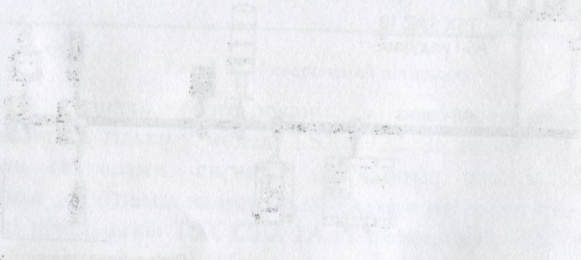
Рис. 1.13. Приклад побудови AS-і шини

Серед пристроїв із вбудованої AS-і функцією застосовуються:

- фотоелектричні датчики для виявлення об'єктів усіх видів із використанням 5 базисних систем: на просвіт, відбитого світла, поляризованого відображення, розсіяного світла з його загашенням і без загашення;
- індуктивні датчики наближення для виявлення різноманітних металевих об'єктів;
- кнопкові станції і клавіатури на 12 клавіш;
- пристрої світлової і звукової сигналізації;
- пристрої для керування приводами.

AS-і джерела живлення спеціально розроблені для живлення компонентів, зв'язаних із AS-і шиною. Вони під'єднуються безпосередньо до шини.

Кожна AS-і шина може містити максимум 31 ведений, кожен з яких може мати максимум чотири дискретних входи і/чи чотири виходи. Отже максимальна кількість дискретних входів-виходів, із яких ПЛК TSX Micro може обмінюватися по AS-і шині – 124 входів + 124 виходів, тобто 248



Стандартне шасі використовується у тому разі, коли контролер складається з одного шасі.

Шасі, яке може розширюватись, призначено для створення контролера, який може включати кілька (до 16) шасі. Такі шасі мають розніми, через які за допомогою спеціального кабелю внутрішні X Bus шини окремих шасі об'єднуються у загальну X Bus шину, що дає змогу обмінюватись сигналами і даними між модулями, встановленими у різні шасі. Загальний вигляд шасі, яке може розширюватись, наведений на рис. 1.15.

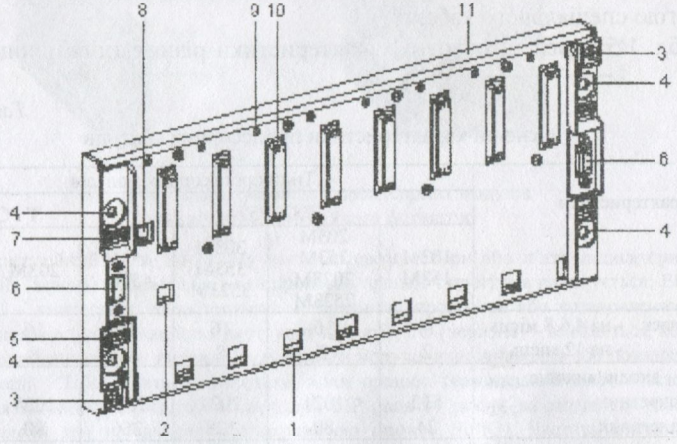


Рис.1.15. Загальний вигляд шасі, що розширюється:

1 – апертури для закріплення модулів; *2* – апертура, яка гарантує правильне розташування модуля живлення на шасі. Оскільки модуль живлення має виступ на лицьовому боці модуля, його неможливо встановити у іншій позиції; *3* – клема для заземлення шасі; *4* – отвори для закріплення шасі; *5* – місце для маркування адреси шасі; *6* – 9-штирковий рознім для під'єднання зовнішньої X Bus шини при під'єднанні до інших шасі; *7* – місце для маркування мережевої адреси контролера; *8* – мікроперемикачі для кодування адреси шасі; *9* – отвори з внутрішньою різьбою для закріплення модуля; *10* – 48-штирковий рознім для під'єднання модуля до шасі; *11* – металева пластина, що є основою для розташування електронної карти шини X Bus і захисту її від впливу електромагнітних перешкод; установлення модулів контролера

У стандартному шасі відсутні розніми для під'єднання X Bus шини й мікроперемикачі для кодування адреси шасі.

Якщо контролер складається з кількох шасі, то вони з'єднуються між собою за допомогою спеціального кабеля X Bus шини. Крім того, на кінцях X Bus шини повинні бути встановлені термінатори шини. Є два типи терміна торів – A і B, які мають позначення TSX TLY EX. Вони можуть бути встановлені у будь-якому порядку, але на одній шині повинні бути термінатори з різним літерним позначенням.

Для побудови більш складних структурно розподілених контролерів можна використовувати спеціальні модулі розширення X Bus шини – TSX REY 200.

1.3.2. Модулі процесорів

Фірми, які випускають мікропроцесорні контролери, постійно працюють над їх удосконаленням і розширенням функціональних можливостей, тому їх типи змінюються. Випускаються процесорні модулі, які умовно можна поділити на дві групи: TSX P57 xx3M та T PCX 57 xx3M.

Процесори TSX P57 xx3M встановлюються на шасі контролера, а T PCX 57 xx3M – на ISA шині PC. Процесорний модуль, розташований у корпусі PC, під'єднується до шасі з установленими модулями входів-виходів за допомогою спеціального кабелю.

У табл. 1.5 наведені основні характеристики різних типів процесорних модулів.

Таблиця 1.5

Основні характеристики процесорних модулів

Характеристики	Типи процесорних модулів					
	TSX P57				T PCX 57	
	103M 153M	203M 253M 2023M 2523M	303M 353M 3523M	453M 453M	203M	353M
Кількість шасі: - на 4,6,8 місць	4	16	16	16	16	16
- на 12 місць	2	8	8	8	8	8
Кількість входів/виходів у шасі: - дискретних	512	1024	1024	2048	1024	1024
- аналогових	24	80	128	256	80	128
Кількість спеціальних модулів:	8	24	32	64	24	32
Кількість мереж	1	1	3	4	1	3
Кількість конфігурованих контурів регулювання	-	10	15	20	10	15
Пам'ять: - вбудоване ОЗУ	32	48	64	128	48	96
- розширення ОЗУ	64	160	384	512	160	384

Модулі процесорів розрізняються функціональними можливостями, основними з яких є:

- кількість шасі, які можуть входити до складу ПЛК;
- кількість входів-виходів, які може обробити контролер;
- кількість спеціальних модулів;
- кількість і типи мереж, до яких може під'єднуватись ПЛК;
- кількість конфігурованих контурів регулювання;
- види і ємність пам'яті.

На рис. 1.16 показані загальні види процесорних модулів, які встановлюються на шасі.. За розміром вони бувають двох форматів: стандартного і подвійного.

Процесорні модулі 253M, 353M і 453M мають інтегровану польову шину FIPIO, до якої можна під'єднувати віддалені дискретні і аналогові сигнали. А модулі 2523M, 3523M крім шини FIPIO розніми для підключення мережі Ethernet.

При виборі процесора T PCX 57 xx3M, місце для встановлення процесорного модуля на шасі залишається вільним.

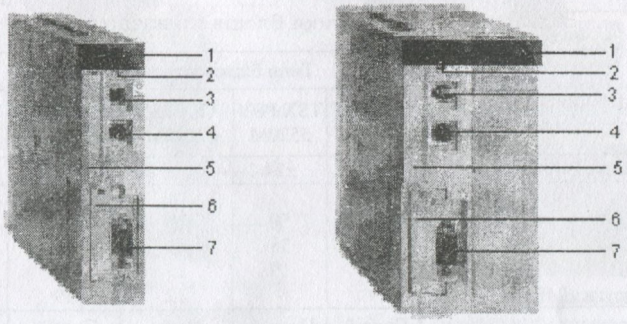


Рис.1.16 Загальний вигляд процесорних модулів стандартного і подвійного форматів:

1 – дисплейний блок, до складу якого входять чотири або п'ять індикаторних ламп: RUN (зелена) – ввімкнена, коли процесор працює і програма виконується; ERR (червона) – вмикається, коли виникає несправність процесора або встановлених у нього пристроїв (комунікаційних карт, карт пам'яті); I/O (червона) – вмикається, коли система самодіагностики виявить несправності модулів входів-виходів або помилки в конфігурації; TER (жовта) – миготить, коли працює термінальний порт. Частота миготіння визначається частотою передачі, FIP (жовта) указує на активність шини FIPIO (тільки для процесорів з інтегрованою шиною FIPIO). Частота миготіння визначається частотою передачі; *2* – кнопка під олівець RESET, при натисканні на яку відбувається холодний рестарт ПЛК; *3* – термінальний порт TER для під'єднання до нього периферійних пристроїв за протоколом UNI-TELWAY; *4* – термінальний порт AUX для під'єднання до нього периферійних пристроїв, які мають власне джерело живлення, за протоколом UNI-TELWAY; *5* – слот для встановлення карти розширення пам'яті формату PCMCIA типу 1. Якщо карта відсутня, на цьому місці обов'язково повинна бути встановлена спеціальна заглушка. Якщо її не буде, контролер зупиниться; *6* – слот для встановлення комунікаційної карти формату PCMCIA типу 3, яка дає можливість зв'язатись з процесором по мережах FIPWAY, FIPIO Agent, UNI-TELWAY або за послідовним протоколом зв'язку. Якщо комунікаційна карта відсутня, слот повинен бути закритий кришкою; *7* – 9-штирковий SUB D рознім для під'єднання до інтегрованої у процесор польової шини FIPIO. Цей рознім є тільки на процесорах, які мають таку шину

Модуль процесора з'єднується із шасі за допомогою кабелю X Bas шини. Вважається, що на процесорному модулі встановлений термінатор із позначенням A, тому на іншому кінці X Bas шини (останнє шасі) необхідно встановити термінатор лінії з позначенням B.

1.3.3. Блоки живлення

У кожному шасі повинен бути встановлений блок живлення. Пропонуються різні типи блоків живлення, які розрізняються живленням від змінного або постійного струму, потужністю, а також

розмірами. У табл.1.6 наведені типи й основні технічні характеристики блоків живлення.

Таблиця 1.6

Загальні характеристики блоків живлення

Загальні характеристики	Типи блоків живлення					
	TSX PSY 1610M	TSX PSY 3610M	TSX PSY 5520M	TSX PSY 2600M	TSX PSY 5500M	TSX PSY 8500M
Напруга живлення , В	±24	±24	±24	~100....240	~100...240	~100...240
Потужність, Вт:						
загальна	30	50	50	26	50	77
по напрузі ±24 В	15	35	35	25	35	75
по напрузі ±5 В	15	19	19	15	19	-
зовнішньої напруги ±24В	-	-	-	12	19	38
Формат	Стан-дартний	Подвій-ний	Подвій-ний	Стан-дартний	Подвій-ний	Подвій-ний

Як видно з табл. 1.6, блоки, що живляться змінною напругою, мають додатковий вихід для живлення ланцюгів датчиків напругою 24 В постійного струму.

Блок живлення для кожного шасі вибирається виходячи з типів і кількості модулів, які планується встановити у шасі. Для цього використовуються дані, наведені в інструкції за експлуатації, про потужність, яку споживає кожний модуль по напрузі =5 В і =24 В. Після цього розраховується загальна потужність, яку споживають всі модулі, встановлені у шасі, і підбирається блок живлення, який може задовольнити цим потребам.

На рис. 1.17 показано загальний вигляд блоків живлення стандартного й подвійного формату.

Сигнальне реле, що встановлене у кожному блоці живлення, виконує кілька функцій:

- якщо блок живлення розташований у шасі з установленим модулем процесора, то за нормальної роботи контролера контакт сигнального реле замкнений;
- якщо з якоїсь причини виконання програми припиняється і контролер переходить у режим STOP, контакт реле розмикається;

У блоках живлення, встановлених у інші шасі контролера, контакт сигнального реле замкнений у разі, якщо блок живлення працює нормально. В іншому разі цей контакт розмикається.

Отже, контакти сигнального реле можна використовувати у системах безпеки контролера і системи керування.

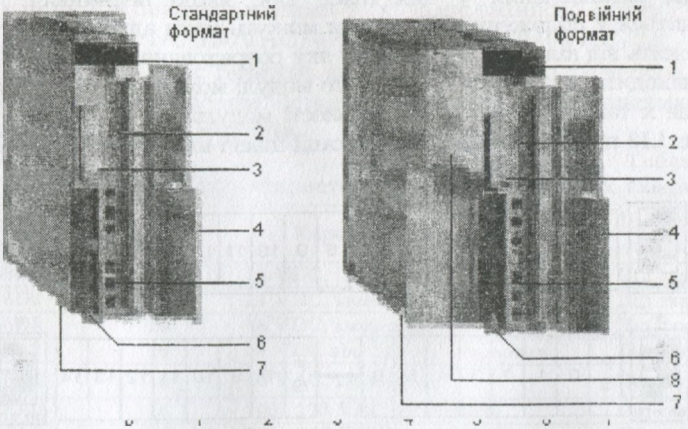


Рис. 1.17. Загальний вигляд блоків живлення:

1 – дисплейний блок, який включає: індикаторну лампу ОК (зелена), яка ввімкнена якщо блок працює нормально; індикаторну лампу BAT (червона), яка ввімкнена, якщо несправна або відсутня батарейка резервного живлення оперативної пам'яті процесора; індикаторну лампу 24 В (зелена), яка ввімкнена, якщо напруга датчиків ±24В у нормі. Цей індикатор встановлений у блоках живлення змінного струму; *2* – кнопка під олівець RESET, натискання якої викликає теплий перезапуск контролера; *3* – слот для встановлення батарейки резервного живлення оперативної пам'яті процесора; *4* – кришка для захисту контактів; *5* – виводи "під гвинт" для під'єднання: мережі живлення, контактів сигнального реле, живлення датчиків (для модулів живлення змінного струму); *6* – хомут для закріплення кабелю живлення; *7* – запобіжник; *8* – селектор живлення, який встановлений на блоках живлення TSX PSY 5500M і TSX PSY 8500M.

1.3.4. Принципи розміщення й адресації модулів у контролері

Кожне шасі, яке входить до складу контролера, має свою унікальну адресу. Якщо контролер складається з одного стандартного шасі, воно, по замовченю, має адресу 0.

Якщо контролер складається з кількох шасі, які розширюються, то адреса кожного шасі виставляється за допомогою мікроперемикачів, які розташовані на шасі (рис. 1.15, поз. 8). Для шасі, в якому розміщений процесорний модуль, встановлюється адреса 0 (рис. 1.18). Для інших шасі, за допомогою перших трьох мікроперемикачів, у двійковому коді виставляється адреса шасі. Порядок розташування шасі на шині X Bus не залежить від їхньої адресації.

У попередніх версіях PL7-Pro положення мікроперемикача №4 не використовується і він повинен перебувати у положенні ON.

В останніх версіях програмного забезпечення положення перемикача № 4 використовується для збільшення можливої кількості шасі, які використовуються при побудові контролера.

У цьому разі два шасі можуть мати однакове положення перших трьох перемикачів, а положення перемикача № 4 буде визначати адресацію

модулів, встановлених у цих шасі. Так, якщо перемикач № 4 буде знаходиться у положенні ON, модулі можуть мати адресації від 00 до x, де x залежить від кількості модулів, на яку розраховане шасі. Якщо перемикач №4 знаходиться у положенні OFF, то модулі можуть мати адресацію від 08 до x, де x також залежить від кількості модулів на яку розраховане шасі. На рис. 1.18 показані приклади адресації шасі і модулів в контролері.

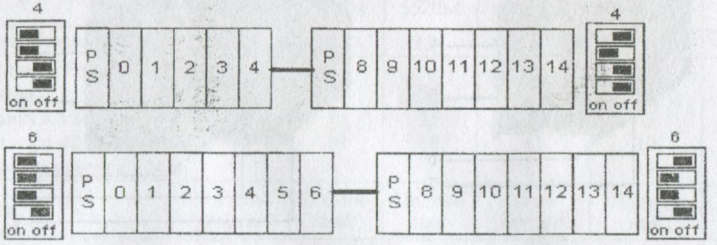


Рис.1.18. Приклади адресації шасі і модулі при використанні перемикача №

При цьому треба пам'ятати, що для такої комбінації не можна використовувати шасі, які розраховані на 12 місць.

У зв'язку з тим що модулі живлення і процесорні модулі можуть бути як стандартного так і подвійного формату, то може змінюватись як кількість місць, які відводяться для встановлення інших модулів, так і адреси, які вони можуть мати.

Наприклад, якщо блок живлення має подвійний формат, то процесорний модуль може бути встановлений тільки на місце під номером 01. Тоді інші модулі можуть займати місця, починаючи з номера 02. Якщо ж і процесорний модуль має подвійний формат, то модулі можуть займати місця, починаючи з номера 03. Для шасі, в яких використовується мікроперемикач № 4 у разі використання блока живлення подвійного формату інші модулі можуть займати місця, починаючи з номера 09.

1.3.5. Модулі дискретних входів-виходів

Для задовільнення різноманітних потреб користувача випускається широкий діапазон дискретних модулів входів-виходів, які розрізняються:

- кількістю каналів – 8, 16, 28, 32 або 64;
- типами входів:
  - модулі із входами постійного струму (DC) – 24, 48 VDC;
  - модулі із входами змінного струму (AC) – 24, 48, 110, 240 VAC;
- типами виходів:
  - модулі з релейними виходами;
  - модулі з безконтактними виходами постійного струму (DC)

24VDC/0,1A – 0,5A – 2A; 48VDC/ 0,25A – 1A;

- модулі з безконтактними виходами змінного струму (AC) 24VAC/ 1A; 130VAC/1A; 48VAC/2A; 240VAC/2A

- типами під'єднання: гвинтова клемна колодка або з'єднувачі HE10

У табл. 1.7 і 1.8 наведені основні технічні характеристики для різних типів дискретних модулів.

Таблиця 1.7

Основні технічні характеристики модулів дискретних входів

Позначення модуля	Кількість каналів	Характеристики каналів	Під'єднання
TSX DEY 08D2	8	24 VDC, ізольовані	Під гвинт
TSX DEY 16D2	8	24 VDC, ізольовані	Під гвинт
TSX DEY 16D3	16	48 VDC, ізольовані	Під гвинт
TSX DEY 16A2	16	24 VDC або 24VAC, ізольовані	Під гвинт
TSX DEY 16A3	16	48 VAC, ізольовані	Під гвинт
TSX DEY 16A4	16	100...120 VAC, ізольовані	Під гвинт
TSX DEY 16A5	16	200...240 VAC, ізольовані	Під гвинт
TSX DEY 16FK	16	24 VDC, ізольовані швидкі входи	HE 10
TSX DEY 32D2K	32	32 VDC, ізольовані	HE 10
TSX DEY 64D2K	64	64 VDC, ізольовані	HE 10
TSX DEY 32D3K	32	48 VDC, ізольовані	HE 10
TSX DMY 28FK	16 входів 12 виходів	24 VDC, ізольовані швидкі входи 24 VDC, ізольовані , 0,5 A	HE 10
TSX DMY 28RFK	16 входів 12 виходів	24 VDC, ізольовані швидкі входи 24 VDC, ізольовані , 0,5 A	HE 10

Таблиця 1.8

Основні технічні характеристики модулів дискретних виходів

Позначення модуля	Кількість каналів	Характеристики каналів	Під'єднання
TSX DSY 08T2	8	24 VDC/0,5A, захищені, транзисторні	Під гвинт
TSX DSY 08T22	8	24 VDC/2A, захищені, транзисторні	Під гвинт
TSX DSY 16T2	16	24 VDC/0,5A, захищені, транзисторні	Під гвинт
TSX DSY 08T31	8	48 VDC/1A, захищені, транзисторні	Під гвинт
TSX DSY 16T3	16	48 VDC/0,25A, захищені, транзисторні	Під гвинт
TSX DSY 08R5	8	24 VDC або 24...240 VAC, 3A, релейні, не захищені	Під гвинт
TSX DSY 16R5	16	24...48VDC або 24...240 VAC, 5A, релейні, захищені	Під гвинт
TSX DSY 08R5A	8	24 VDC або 24...240 VAC, 3A, релейні, не захищені	Під гвинт
TSX DSY 08R4D	8	24...120VDC, 5A, релейні, захищені	Під гвинт
TSX DSY 08S5	8	48...240 VAC, 2A, тиристорні, захищені	Під гвинт
TSX DSY 16S5	16	48...240 VAC, 1A, тиристорні, захищені	Під гвинт
TSX DSY 16S4	16	24...120 VAC, 1A, тиристорні, не захищені	Під гвинт
TSX DSY 32T2K	32	24 VDC/0,1A, захищені, транзисторні	HE 10
TSX DSY 64T2K	64	24 VDC/0,5A, захищені, транзисторні	HE 10

Серед модулів дискретних входів є модулі з так званими швидкими входами (TSX DEY 16FK, TSX DMY 28FK та TSX DMY 28RFK). Входи цих модулів, за аналогією з першими чотирма входами модуля розташованого на першому місці контролера TSX Micro), можна використовувати як звичайні дискретні входи, входи із защіпкою або входи для обробки подій.

Клемні колодки на модулях входів-виходів (рис.1.19) можна знімати. Це полегшує заміну цих модулів. Особливістю клемних колодок для TSX Premium є наявність спеціального кодувального пристрою *4*, який автоматично встановлюється у відповідне положення при першому встановленні клемної колодки на модуль. Кожний тип модуля має свій код, тому неможливо помилково встановити клемну колодку одного типу модулів на інші.

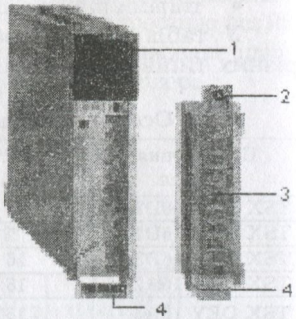


Рис. 1.19. Загальний вигляд модуля з клемною колодкою

1 – дисплейний блок; *2* – гвинт; *3* – знімна клемна колодка; *4* – кодувальний пристрій

При встановленні і закріпленні клемної колодки вона, спочатку, вставляється у кодувальний пристрій, а потім гвинтом *2* закріплюється на модулі. Клемна колодка *3* має кришку, яка закриває доступ до клем і має змінний ярлик, на якому із зовнішнього боку вказується тип модуля і можуть бути внесені позначення входів-виходів, а із внутрішнього боку показана схема під'єднання входів-виходів до модуля.

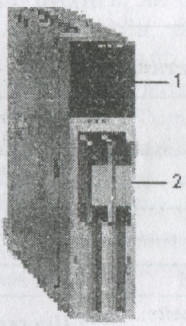


Рис. 1.20. Загальний вигляд модуля з конекторами

1 – дисплейний блок; *2* – конектори

Кількість конекторів, розташованих на лицьовій панелі модуля (рис. 1.20), залежить від кількості каналів, з якими працює цей модуль. Так, для модуля, розрахованого на 64 канали, кількість конекторів чотири, а для модуля, розрахованого на роботу з 16 каналами – один.

Аналогічно, як і для модулів з конекторами TSX Micro, зовнішні сигнали до таких типів модулів під'єднуються або за допомогою спеціальних блоків TELEFAST, або за допомогою спеціальних кабелів з розпушеними вільними кінцями.

Кожний модуль має дисплейний блок (рис. 1.21) на якому розташовані індикатори стану модуля: **RUN** (зелений), **ERR** і **I/O** (червоні), а також індикатори з позначенням номерів каналів входів-виходів. Кількість цих індикаторів відповідає кількості каналів модуля. Максимальна кількість таких індикаторів – 32. Якщо модуль розрахований на більшу кількість каналів (64), то у нижній частині

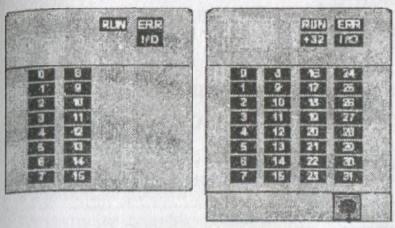


Рис. 1.21. Загальний вигляд дисплеїв для 16- канального і 64-канального

дисплею розташована кнопка переключення на іншу групу з 32 каналів. При цьому у верхній частині дисплея загоряється індикатор **+32.**

Індикатори каналів висвітлюються при спрацьовуванні відповідного вхідного або вихідного каналу.

У нормальному стані модуля повинен горіти тільки індикатор RUN. Висвітлення індикаторів ERR або I/O сигналізує про виявлення системою самодіагностики відмови модуля або окремих його каналів.

1.3.6. Модулі аналогових входів-виходів

У табл. 1.9 наведені основні технічні характеристики аналогових модулів.

Таблиця 1.9

Основні техніцчні характеристики модулів аналогових входів

Позначення модуля	Кількість каналів	Діапазон вимірювання	Характеристики каналів	Під'єднання
Модулі аналогових входів				
TSX AEY 1600	16	±10 В, 0...10 В, 0...5 В, 1...5 В, 0-20 мА, 4-20 мА	Входи з загальною точкою, розрядність АЦП 12 біт	2 SUB-D. конектора
TSX AEY 800	8	±10 В, 0...10 В, 0...5 В, 1...5 В, 0-20 мА, 4-20 мА	Входи з загальною точкою, розрядність АЦП 12 біт	1 SUB-D. конектор
TSX AEY 810	8	±10 В, 0...10 В, 0...5 В, 1...5 В, 0-20 мА, 4-20 мА	Ізоляція між каналами, розрядність АЦП 12 біт	1 SUB-D. конектор
TSX AEY 420	4	±10 В, 0...10 В, 0...5 В, 1...5 В, 0-20 мА, 4-20 мА	Ізоляція між каналами, роз- рядність АЦП 16 біт, швидкодіючий	1 SUB-D. конектор
TSX AEY 414	4	±10 В, 0...10 В, 0...5 В, 1...5 В, 0-20 мА, 4-20 мА, термопари, термометри опору	Високий рівень ізоляції між каналами, розряд- рядність АЦП 16 біт	Під гвинт
TSX AEY 1614	16	-80...+80 мВ; термопари B, E, J, K, L, N, R, S,T, U	Ізоляція між каналами, розрядність АЦП 16 біт	2 SUB-D. конектора
Модулі аналогових виходів				
TSX ASY 410	4	±10 В, 0-20 мА, 4-20 мА	Ізоляція між каналами, розрядність ЦАП 11 біт + знак	Під гвинт
TSX ASY 800	8	±10 В, 0-20 мА, 4-20 мА	Виходи з загальною точкою, розрядність ЦАП: 13 біт + знак для напруги, 13 біт для струму	1 SUB-D. конектор

Ці модулі розрізняються:

- кількістю каналів (4, 8, 16);
- характеристиками і діапазонами сигналів (напруга, струм, термопара, універсальні);

- наявністю гальванічного розподілення;
- типами під'єднання (25-штировий SUB D конектор або клемна колодка).

Модулі аналогових входів-виходів можна встановлювати у будь-який слот шасі. Їх можна знімати при підключеному живленні контролера.

Максимальна кількість аналогових каналів залежить від модуля процесора, який встановлений у контролері (табл. 1.5).

Аналогічно аналоговим модулям TSX Micro аналогові вхідні модулі TSX Premium виконують функції:

- сканування вхідних каналів за допомогою безконтактного мультиплексування і збір даних;
- аналого-цифрове перетворення вхідних вимірювань;
- фільтрація сигналів;
- перетворення вхідних вимірювань у формат користувача;
- моніторинг модуля: тестування ланок перетворення, вхідний контроль перевищування рівня сигналу, тест наявності клемної колодки.

Модулі аналогових виходів виконують функції:

- захист каналів модулів від перевантаження;
- адаптація до різноманітних виконавчих механізмів: вихідний сигнал у вигляді струму або напруги;
- цифроаналогове перетворення;
- перетворення даних прикладної програми у дані, які використовуються цифро-аналоговим перетворенням;
- моніторинг модуля: тест перетворення, тест виходу за межі, тест наявності клемної колодки.

Кількість TSX ASY 800 модулів, встановлених в одне шасі, обмежено двома модулями. Це викликано тим, що цей модуль споживає велику потужність по напрузі 24 В. Тому в разі потреби використання більшої кількості таких модулів необхідно забезпечити їх додаткове живлення зовнішнім джерелом живлення. Для цього на лицьовій панелі модуля розташовані спеціальні клеми.

Аналогові модулі мають дисплейний блок із розташованими на ньому трьома індикаторними лампами RUN, ERR і I/O, які відображають режим роботи модуля і можливі несправності (табл. 1.10).

Час опитування модулів (як і для TSX Micro) залежить від вибраного в процесі конфігурування режиму опитування. За нормального режиму час опитування становить 27 мс для модулів TSX AEY 800 і TSX AEY 810 та 51 мс для модуля TSX AEY 1600. За швидкого режиму час опитування становить близько 3 мс на канал.

Час опитування модуля TSX AEY 1614 залежить від опцій, які встановлені при конфігуруванні модуля і становить близько 80 мс на

канал. Швидкий модуль TSX AEY 420 має час опитування близько 2 мс, а універсальний вхідний модуль TSX AEY 414 – 550 мс.

Таблиця 1.10

Стан індикаторів

Стан модуля	Стан ламп		
	RUN	ERR	I/O
Нормальна робота	●	O	O
Модуль не сконфігурований	O	⊗	O
Зовнішня помилка (перевищення діапазону або помилка зовнішнього живлення 24 В)	●	O	●
Внутрішня помилка (відмова модуля): зв'язок з процесором можливий;	●	●	O
зв'язок з процесором не можливий;	O	●	O
Інша помилка (клемної колодки)	●	O	⊗

При роботі із вхідними аналоговими модулями можна виконувати процедуру юстирування окремих каналів. Наприклад, датчик тиску, який має умовну шкалу 1мВ/Па, показує 3200 Па, у той час, як фізичне значення тиску дорівнює 3100. Користувач може привести вимірювання до значення 3100. Після цієї процедури юстирування, канал вимірювання буде виконувати систематичне зміщення +10 до будь-якого нового значення. Це значення буде зберігатись у пам'яті ПЛК Для модулів аналогових входів можна також виконувати процедуру перекалібровки каналів, для того щоб усунути похибки вимірювання за рахунок дрейфу характеристик вимірювальних каналів. Перекалібровка виконується з термінала програмування за допомогою спеціального екрана калібровки.

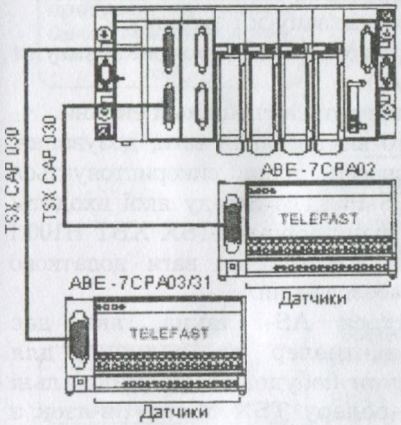


Рис. 1.22. Під'єднання аналогових модулів через TELEFAST-2

Сигнали до модулів з SUB-D конектором під'єднуються до контролера за допомогою блоків швидкого монтажу Telefast-2.

- Блок Telefast ABE-7CPA02 забезпечує під'єднання 8 каналів.
- Блок Telefast ABE-7CPA03/31 забезпечує під'єднання 8 каналів, а також живлення каналів 24 VDC.
- Блок Telefast ABE-7CPA12 забезпечує під'єднання 16 термопар. На ньому встановлений датчик температури для компенсації холодних спаїв.

1.3.7 Модулі лічильників

Для реалізації функцій підрахунку імпульсів у ПЛК TSX Premium використовуються три типи модулів лічильників.

Модулі CTY 2A/4A використовуються для підрахунку імпульсів із максимальною частотою 40 кГц у режимах прямого, зворотного і реверсивного лічильників. Вони відрізняються тільки кількістю каналів: 2 - для модуля TSX CTY 2A і 4 – для модуля TSX CTY 4A.

Модуль TSX CTY 2C використовується як реверсивний лічильник і для вимірювання імпульсів із частотою до 1 МГц.

Параметри лічильників встановлюються програмним шляхом у процесі конфігурування. До цих модулів можна під'єднувати датчики з безконтактними виходами (інкрементні декодери, фотоелектричні датчики, датчики положення та ін.) і датчики з механічними контактами (у цьому разі припустима частота імпульсів не повинна перевищувати 100 Гц).

1.3.8. Спеціальні модулі

Модулі TSX CAY 21/22, TSX CAY 41/42 та TSX CAY 33 використовуються для розв'язання задач керування серводвигунами з метою досягнення ефективного керування складним рухом машин і агрегатів за допомогою мікропроцесорного контролера.

Модулі TSX CAY 21/22 керують переміщенням по двох незалежних осях (канал 0 і 1). Модулі TSX CAY 41/42 можуть контролювати керування чотирма незалежними осями. Модуль TSX CAY 33 контролює керування трьома осями. Модулі дають можливість контролювати положення робочих органів машин, змінювати швидкість і прискорення їх переміщення, контролювати їх запуск і зупинку.

Для керування машинами, які застосовують крокові двигуни використовуються модулі TSX CFY 11/21.

Для вводу параметрів руху використовуються спеціальні екрани.

Для розв'язання задач автоматичного вимірювання ваги, дозування, регулювання подачі матеріалу на конвеєр тощо використовується інтегрована система вимірювання ваги IPS Plus, до складу якої входять: модуль вимірювання ваги TSX ISP Y100, індикатор ваги TSX XBT H100 і тензодатчики. Для роботи з операціями вимірювання ваги додатково поставляється спеціалізоване програмне забезпечення.

У TSX Premium використовується AS-i шина, яка дає можливість вико-ристовувати цей контролер як ведучий для управління цією шиною. Склад, принципи побудови і функціональні можливості цієї шини аналогічні конт-ролеру TSX Micro. Зв'язок з шиною відбувається через модуль TSX SAY 100.

З 1998 р. компанія Schneider Electric почала випускати нову гаму модулів розподіленого управління – Modicon TSX Momentum до складу якої входять модулі розподілених входів-виходів, процесорів, комунікаційних адаптерів і адаптерів розширення. TSX Momentum може під'єднуватись до ПЛК із використанням багатьох поширених відкритих промислових мереж: Modbus Plus, FIPIO, Ethernet, Interbus, Profibus, DeviceNet, ControlNet.

2. ПРОГРАМУВАННЯ КОНТРОЛЕРІВ MODICON TSX

2.1 Загальні положення

Для програмування ПЛК TSX Micro та TSX Premium використовується програмне забезпечення PL7, яке постійно розвивається і існує багато його версій. Тому дуже важливо звертати увагу на те щоб версія операційної системи, яка записана у пам'ять контролера співпадала з версією програмного забезпечення, за допомогою якого планується програмувати ПЛК.

Розробка прикладної програми включає в себе декілька етапів (рис. 2.1). На першому етапі розробки прикладного програмного забезпечення необхідно визначити структуру програми користувача, яка може бути однозадачною чи багатозадачною. Крім того вона може включати в себе підпрограми та задачі обробки подій. Кожна з цих задач програмується окремо. Програмування кожної задачі може відбуватись з використанням мов програмування, які підтримує ПЛК. У контролерах TSX Micro та TSX Premium можуть використовуватись 4 мови програмування: Крокових діаграм – Ladder Diagram, Список інструкцій – Instruction List, Структурований текст – Structured Text та Графсет – Grafset. Окремі розділи програми користувача можуть бути написані на різних мовах програмування, але при записі у контролер вони компілюються у єдиний машинний код.

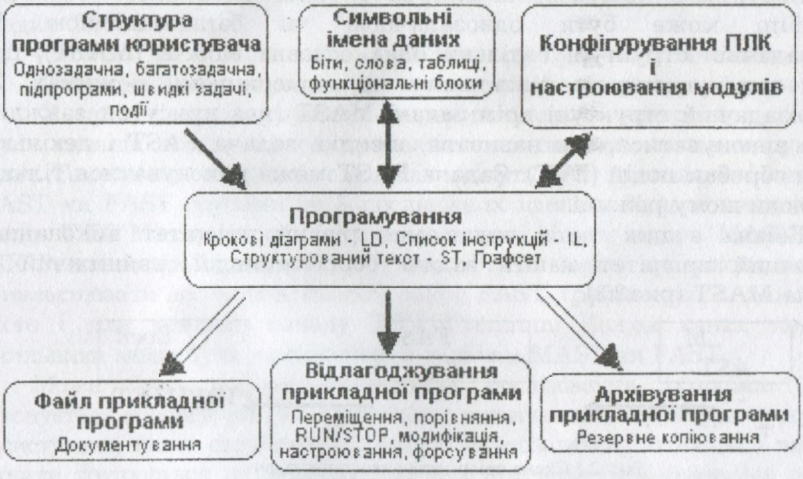


Рис. 2.1 Етапи розробки прикладного програмного забезпечення

Для розроблення прикладної програми необхідно виконати процедуру конфігурування контролера та настроювання окремих модулів.

У процесі роботи зі змінними можна використовувати не тільки адресну форму їх представлення а й присвоювати змінним символьні імена, які можна використовувати як при програмуванні, так і при відлагоджуванні програми.

Для виконання окремих етапів ство-рення прикладної програми користувача (ПрК) використовуються окремі редактори (рис.2.2) – Configuration (Конфігурування), Program (Програмування), Variables (Змінні), Documentation File (Документування), але зберігається вона (і це дуже зручно) у одному файлі з розширенням . stx.

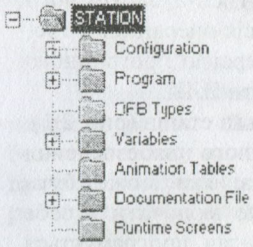


Рис.2.2 Загальне меню

В процесі розробки ПрК можливо використовувати спеціальні засоби для її відлагоджування (покрокове виконання ПрК, таблиці анімації (Animation Tables), вікна для відлагоджування (Runtime Screens), форсування змінних і т.ін.

Після закінчення розробки прикладної програми вона може бути захищена від несанкціонованого доступу, тобто ПрК не може бути переглянута, а тим більше виправлена.

У разі потреби окремі елементи програми користувача можуть бути задокументовані і роздруковані.

2.2 Структура програми користувача

Структура програми користувача ПЛК TSX Micro та TSX Premium може бути однозадачною чи багатозадачною. В однозадачній структурі є тільки одна основна задача (MAST), що може виконуватись у циклічному чи періодичному режимах. У багатозадачній структурі крім задачі MAST (яка присутня завжди) може виконуватися, так називана, швидка задача FAST і декілька задач обробки подій (EVT). Задача FAST може виконуватися тільки в періодичному режимі.

Кожна з цих типів задач має різний пріорітет виконання. Найвищий пріорітет мають задачі обробки подій, найнижчий – задача MAST (рис.2.3).

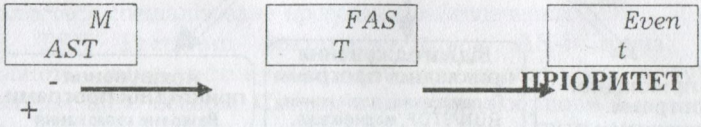


Рис. 2.3 Схема пріорітетів виконання задач

На початку виконання задачі, що володіє більш високим пріоритетом, виконання задачі з більш низьким пріоритетом припиняється, а після її закінчення продовжується виконання перерваної задачі (рис.2.4).

Структура програми визначається користувачем у залежності від властивостей об'єкта і задач системи керування.

Якщо система призначена для рішення приблизно однакових за значущістю і швидкодією задач керування можна обмежитися однозадачною структурою і використовувати тільки задачу MAST.

Якщо серед задач керування можна виділити таку, що вимагає більш оперативного реагування системи керування, її розв'язання можна доручити задачі FAST, вибравши багатозадачну структуру програми.

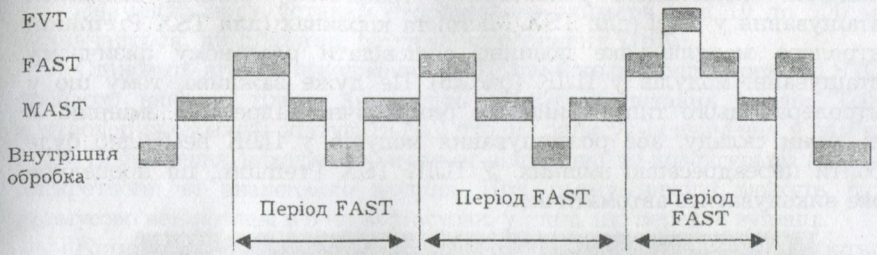


Рис.2.4 Структура виконання багатозадачної прикладної програми

Якщо ж деякі керуючі події вимагають практично миттєвого реагування на них, в системі керування необхідно використовувати задачі з обробки подій.

Кількість задач обробки подій EVTi залежить від моделі ПЛК. Як керуючі події, що можуть запустити задачі обробки подій EVTi, можуть використовуватися:

- дискретні сигнали на перших чотирьох входах дискретного модуля, встановленого в першому слоті TSX Micro або на швидких входах TSX Premium;
- момент досягнення заданих значень для будь-яких лічильників;
- повідомлення, отримані по промисловій контролерній мережі.

У процесі конфігурування окремих модулів контролера, окремі групи каналів входів-виходів призначаються окремим типам керуючих задач. Так канали дискретних модулів можуть бути привласнені задачам MAST чи FAST групами по 8 послідовних каналів. Канали аналогових входів TSX Micro можуть бути привласнені тільки задачі MAST, а у TSX Premium як задачам MAST, так і FAST. Аналогові виходи можна привласнювати як задачі MAST, так і FAST групами по 2 для TSX Micro і для кожного каналу TSX Premium. Кожен канал модуля лічильника може бути привласнений задачам MAST чи FAST.

Може бути рекомендована така послідовність розробки ПрК: виконується конфігурування контролера; визначається структура програми користувача; якщо структура програми багатозадачна, то вхідні і вихідні сигнали групуються по задачам; змінним присвоюються символьні імена; готується і відлагоджується програма для окремих підзадач програми.

2.3. Конфігурування контролера

У процесі конфігурування контролера розрізняють програмну і апаратну конфігурацію.

Для виконання програмної конфігурації задається кількість функціональних блоків (таймерів, лічильників, одновібраторів, драм-

контролерів та регістрів), а також кількість бітів, слів пам'яті і констант, які будуть використовуватись у прикладній програмі користувача.

Апаратна конфігурація виконується за кілька етапів. Спочатку, за допомогою спеціального редактора у графічному вигляді задається розташування у шасі (для TSX Micro) та корзинах (для TSX Premium) контролера модулів, яке повинно відповідати реальному фізичному розташуванні модулів у ПЛК (рис.2.5). Це дуже важливо, тому що у контролерах цього типу прийнята "географічна" адресація змінних. У разі зміни складу, або розташування модулів у ПЛК необхідно буде зробити переадресацію змінних. У ПЛК TSX Premium, ця процедура може виконуватись автоматично.

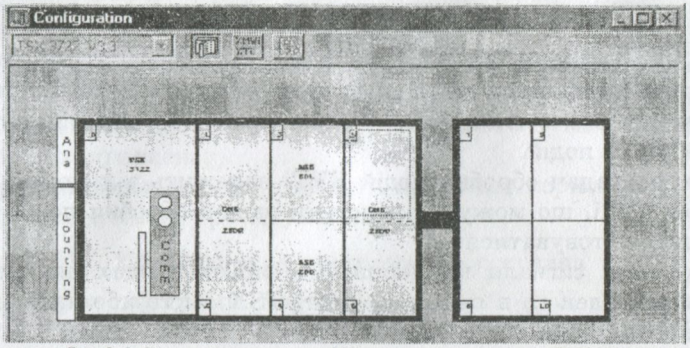


Рис.2.5 Вікно редактора конфігурування контролера TSX Micro

На другому етапі виконується конфігурування окремих модулів і, в залежності від типу модуля, задаються його параметри.

Для конфігурування процесорного модуля, по-перше, задається структура програми користувача і режим її виконання. Так для задачі MAST вибирається циклічний, або періодичний режим її виконання. У разі вибору періодичного режиму задається час його тривалості (від 3 до 255 мс). Якщо в ПрК буде використовуватись задача FAST, у зв'язку з тим що вона може виконуватись тільки у періодичному режимі, для неї окремо також задається тривалість періоду (від 2 до 255 мс).

Якщо за періодичного режиму виконання програми, час робочого циклу перевищить заданий період, ПЛК сигналізує про це установкою у 1 відпо-відних системних бітів. При цьому виконання програми продовжується..

Крім того, тривалість виконання робочого циклу ПрК (окремо для задачі MAST і FAST), як при циклічному, так і при періодичному режимах його виконання, контролюється **сторожовими таймерами**, значення часу яких також встановлюється при конфігурації процесорного модуля. Якщо тривалість робочого циклу задачі перевершує час встановлений у сторожовому таймері, контролер негайно зупиняється

(індикаторні лампи ERR і RUN починають миготіти). За допомогою системи тестування, аналізуючи значення системних бітів і слів, можна визначити причину виникнення цієї ситуації.

Значення часу уставки сторожового таймера (від 10 до 500 мс) визначається, як правило, після налагодження ПрК.

Становить інтерес поводження ПЛК при його зупинці. У режимі STOP процесор виконує тільки внутрішню обробку і **читання входів**. Тобто контролер продовжує опитувати свої входи і записує їх значення у пам'ять ПЛК. Відновлення виходів виконується відповідно до конфігурації кожного дискретного чи аналогового модуля. При цьому виходи можуть бути примусово встановлені в 0 чи зафіксувані у стані, що передує зупинці.

Крім того, при конфігуруванні процесорного модуля задаються ряд функцій, які обумовлюють поведінку ПЛК при його запуску і зупинці. Так функція RUN/STOP використовується для того, щоб запустити чи зупинити виконання ПрК за допомогою зовнішнього дискретного сигналу. Для ПЛК TSX Micro цей вхід чітко визначений і має адресу %I1.8. а для ПЛК TSX Premium адреса цього дискретного входу може бути вибрана довільно.

На процесорному модулі можна задати функцію, яка визначає поведінку ПЛК при його включенні, перезапуску (або під впливом зовнішних факторів, або кнопкою "Reset"). Якщо вибрана функція "Automatic start in Run" (Автоматичний старт в Run) при перезапуску контролера він автоматично переходить в режим виконання ПрК. Якщо ця функція не вибрана при перезапуску - ПЛК переходить у режим Stop.

Функція "%MWi Reset on cold start" задає режим роботи з внутрішніми словами пам'яті при перезапуску ПЛК. Якщо ця функція вибрана, то при перезапуску ПЛК внутрішні слова обнуляються, а якщо ні - внутрішні змінні зберігають значення, які вони мали перед відключенням або перезапуском ПЛК.

Також, якщо в процесі конфігуруваня процесорного модуля TSX Micro дискретному виходу %Q2.0 задана сигнальна функція, то коли ПЛК знаходиться в стані RUN цей вихід знаходиться в стані 1. З появою " помилки, що блокує" і яка приводить до переходу ПЛК у стан STOP – цей вихід переходить у стан 0. Таким чином, цей вихід може використовуватися у схемах безпеки, дублювання і т.п. Якщо ця функція не задана - цей вихід використовується як звичайний дискретний вихід.

Крім того, на процесорних модулях ПЛК виконується конфігурування параметрів і функцій, які викликані особливостями процесорного модуля, або додатково встановлених на них карт, а саме:

- для ПЛК TSX Micro 3722 на процесорному модулі відбувається конфігурування інтегрованих у процесорний модуль аналогових входів-виходів, а також лічильників;

- для ПЛК TSX Premium, для моделей, які мають вбудовану функцію підключення польової шини FIPIO, виконуються процедури її настройки;

- для ПЛК TSX Micro і TSX Premium виконуються настройки комунікаційних PCMCIA карт, встановлених у процесорний модуль;

2.4. Структура пам'яті користувача

Програма користувача зберігається у пам'яті ПЛК і включає:

- *дані* – дані системи і додатки;
- *програма* – виконуваний код задачі і коментарі до неї;
- *константи* – константи, початкові значення і дані конфігурації.

Фізично для збереження ПрК може використовуватися:

- внутрішня енергонезалежна оперативна пам'ять RAM, яка захищена батареєю напругою 3,6 В, що має ресурс роботи 2 роки;
- зовнішня енергонезалежна карта розширення пам'яті формату PCMCIA;
- зовнішня карта FLASH формату PCMCIA;
- внутрішня флеш-память (FLASH EPROM) обсягом 16 Кслів, що використовується як резервна пам'ять збереження ПрК (виключно у ПЛК TSX Micro) і тільки для того випадку коли ПрК зберігається у внутрішній пам'яті RAM і має обсяг не більш ніж 15 Кслів;

Структура пам'яті користувача і розміщення окремих розділів прикладної програми в різних типах пам'яті залежить від типу ПЛК, обсягу ПрК і зовнішніх пристроїв пам'яті, які використовуються (рис.2.6).

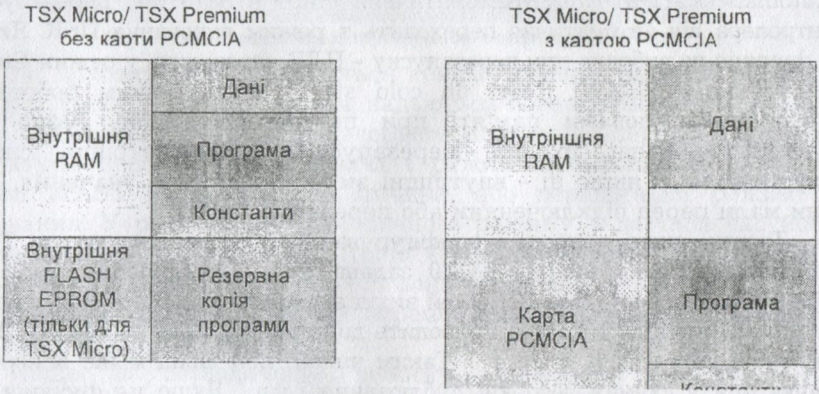


Рис..2.6. Структура пам'яті користувача

Якщо зовнішні карти пам'яті не використовуються уся ПрК (дані, програма і константи) зберігаються у внутрішній оперативній пам'яті RAM і обмежена її ємністю. Якщо використовується ПЛК TSX Micro і обсяг ПрК не перебільшує 15 Кслів, резервна копія ПрК зберігається у внутрішній FLASH EPROM.

Якщо оперативна пам'ять RAM розширена за допомогою зовнішньої карти пам'яті формату PCMCIA в ній зберігається ПрК і

константи, а внутрішня пам'ять RAM резервується тільки для збереження даних (рис.2.6).

Після налагодження програми користувача вона може бути перенесена на зовнішню карту FLASH EPROM. Коли така карта вставляється в ПЛК, починається автоматичне переміщення її змісту у внутрішню RAM і FLASH EPROM.

Незалежно від структури пам'яті ПЛК, а також від того де розташована прикладна програма - у внутрішній RAM або на карті PCMCIA, вона може бути захищена від доступу і копіювання в процесі конфігурування процесорного модуля.

Ємність зовнішньої карти пам'яті RAM і FLASH EPROM, яку можна використовувати, залежить від типу ПЛК і може мати розмір від 32 до 512 Кслів. Зовнішня карта пам'яті RAM також має енергонезалежну батарейку живлення.

2.5. Конфігурування модулів

2.5.1. Конфігурування дискретних модулів.

У процесі конфігурування окремих модулів виконуються процедури, які визначаються типом модуля. Але для всіх модулів входів-виходів, крім модулів аналогових виходів, групи входів-виходів присвоюються задачам MAST або FAST. Це робиться для того, щоб більш ефективно використовувати робочий час ПЛК та скоротити час опитування входів-виходів. Це можна пояснити тим, що при виконанні задачі FAST вона працює тільки з входами-виходами, які використовуються при її виконанні. Тому, якщо ці входи знаходяться в одній групі, то задача FAST опитає тільки цю групу, а коли ці входи розкидані по різним групам, ПЛК буде вимушений опитати всі входи групи, навіть і ті, які не використовуються задачою FAST.

2.5.1.1 Конфігурування модулів дискретних входів

У процесі конфігурування дискретних входів постійної напруги 24 VDC у ПЛК TSX Micro та для швидких модулів дискретних входів ПЛК TSX Premium крім постійного аналогового фільтра для фільтрації перешкод лінії передбачений цифровий фільтр, значення якого встановлюються в процесі конфігуруваня модуля програмним шляхом із запропонованого ряду значень:

Конфігурусмий час фільтрації, у мс															
0,1		1		2		3		4		5		6		7	
	0,6		1,5		2,5		3,5		4,5		5,5		6,5		7,5

Час фільтрації заданий по замовчувані – 4 мс.

Метою введення цього фільтра є виключення впливу перешкод які виникають під час спрацьовування контактів, що є джерелом дискретного сигналу.

Наприклад, джерелом дискретного сигналу є механічні контакти. При цьому існує велика вірогідність виникнення "деренчання" контактів,

що може призвести до помилок при виконанні ПрК. Щоб уникнути цього можна ввести фільтрацію цього сигналу, тобто встановити час у межах якого, ПЛК не буде реагувати на зміну дискретного сигналу. Для таких сигналів, час фільтрації повинен вибиратися великим (не менш 3 мс). І навпаки, у випадку використання дискретних входів як лічильних каналів, час фільтрації повинен бути мінімальним..

Для входів змінної напруги 110/120 чи 200/240 VAC є вбудований постійний фільтр, режим роботи якого в процесі конфігуруваня адаптується для частоти живлення 50 Гц чи 60 Гц.

Дискретні входи швидких модулів ПЛК TSX Premium та перші чотири входи модуля, встановленного у першому слоті ПЛК TSX Micro (%I1.0, %I1.1, %I1.2 та %I1.3) мають ту особливість, що в процесі конфігурування їм можна задати різні режими роботи:

- нормальні дискретні входи (за замовчуванням);
- замикальні входи;
- входи, позв'язані з обробкою подій;
- рахункові канали для прямого, зворотного і реверсивного лічильника (тільки для входів ПЛК TSX Micro).

Замикальні входи, використовуються для організації прийняття дуже коротких імпульсів, тривалість яких може бути менше, ніж час робочого циклу ПЛК. Тобто якщо дуже короткий імпульс надійшов у ПЛК у момент, коли процедура читання входів у робочому циклі вже закінчена. При використанні "нормальних" входів такий імпульс буде непоміченим ПЛК, а при використанні замикальних входів цей імпульс запам'ятається і буде оброблений у наступному робочому циклі ПЛК.

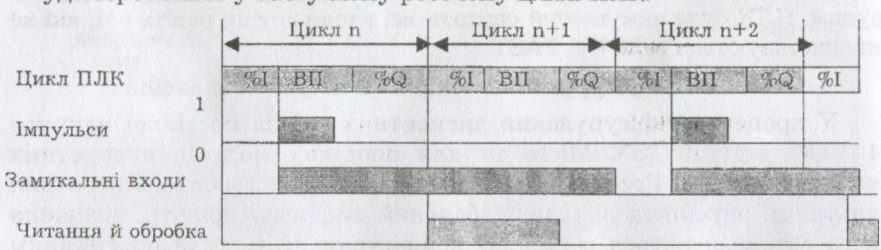


Рис.2.7 Діаграма обробки імпульса на замикальному дискретному вході

При використанні цих дискретних входів, як сигналів для **обробки подій** у процесі конфігуруваня ці входи прив'язуються до номерів подій, обробку яких вони викликають. Для ПЛК TSX Micro кількість подій до 16, а для ПЛК TSX Premium – 64.

Важливою особливістю зазначених вище 4-х дискретних входів ПЛК TSX Micro є те, що вони можуть бути сконфігуровані як **два рахункових канали**, кожний з яких може бути вибраний як прямий, зворотний чи реверсивний лічильник.

Максимальна частота такого лічильника для входів постійної напруги обмежена 500 Гц, а для змінної напруги 110/120 VAC - 20 Гц.

2.5.1.2 Конфігурування модулів дискретних виходів

У процесі конфігурування дискретних виходів насамперед, визначається в якому стані повинні залишитись дискретні виходи у разі аварійної зупинки ПЛК або при переводі його у стан STOP, щоб забезпечити найбільш безпечний режим роботи об'єкта керування. При цьому можна вибрати два режима:

- установка вихідного каналу у 0 ("Fallback to 0") - по замовчуванню;
- утримання виходів у стані, що передував зупинці ("Maintain state").

Крім того, для модулів дискретних виходів, канали яких оснащені схемами захисту від перенавантажень і коротких замкнень, в процесі конфігуруванні необхідно вибрати режим реактивації каналу, яка може виконуватись автоматично ("Automatic") чи програмним (Programmed") шляхом.

Наприклад, на якомусь виході виникло короткочасне перевантаження. У цьому випадку спрацює захист і цей канал вимкнеться. Якщо в процесі конфігуруваня була встановлена автоматична команда реактивації, то періодично, з інтервалом часу близько 10 секунд, будуть виконуватись спроби зняття захисту і включення каналу. Якщо в процесі конфігуруваня модуля була обрана програмна команда реактивації (задана за замовчуванням конфігурація), то реактивація дискретного виходу виконується фрагментом прикладної програми.

2.5.2. Конфігурування аналогових модулів.

2.5.2.1. Конфігурування модулів аналогових входів.

Насамперед, в процесі конфігурування аналогових модулів групи входів-виходів присвоюються задачам MAST або FAST.

У процесі конфігурування модулів аналогових входів, в залежності від типів контролера і аналогового модуля, встановлюються наступні параметри:

- тип циклу опитування (Normal або Fast);
- діапазон (Range);
- масштаб (Scale);
- фільтрація (Filter);
- виявлення клемної колодки (тільки для Premium).

За **нормальномого циклу**, час опитування модуля не залежить від кількості каналів, які підключенні. Наприклад для модуля TSX AEZ 801 нормальний цикл опитування складає 32 мс (4 мс на один канал)

За **швидкого циклу** опитування переглядаються тільки підключені канали, якщо навіть вони підключені не один за одним. Тоді час перегляду модуля обчислюється за формулою:

$$T = 4 \text{ мс} \times N,$$

де N – кількість підключених каналів.

Тобто, якщо у модулі використовуються канали 0, 1, та 7, то за нормального циклу перегляду час опитування модуля - 32 мс, а за швидкого–16 мс.

У зв'язку з тим, що всі модулі аналогових входів можуть піцювати з декількаами діапазонами вимірювання, при конфігуруванні користувач може вибрати бажаний **діапазон** з переліку можливих, з якими працює конкретний модуль. Так, наприклад, для модуля TSX AEZ 802 діапазон вимірювання може бути вибраний або 0-20 мА або 4-20 мА. А для модуля TSX AEY 1600 можуть вибрані діапазони: 0-10 В, ± 10 В, 0-5 В, 1-5 В, 0-20 мА або 4-20 мА.

Вимірювальна величина відображається у ПЛК у вигляді чисел **масштаб** яких вибирається у процесі конфігурування. При вимірюванні уніфікованих електричних сигналів для уніполярного діапазона (0-10 В, 0-5 В, 1-5 В, 0-20 мА, 4-20 мА стандартна форма запису (задана по замовченню) становить 0-10000 одиниць, а для біполярного (± 10 В) - від −10000 до +10000 одиниць. Але у процесі конфігурування є можливість довільно задати значення чисел, які будуть відповідати верхньому і нижньому діапазону. Ці числа можуть бути вибрані у діапазоні від −30000 до +30000.

Для роботи з модулями, що працюють з датчиками температури (TSX AEZ 414, TSX AEY 414, TSX AEY1614) користувач у процесі конфігурування, може вибрати відображення температури:

- у десятих градусах Цельсия;
- у десятих градусах Фаренгейта;
- у стандартних одиницях від 0 до 10000 одиниць, точно визначаючи мінімальну і максимальну температуру, що відповідає 0 і 10000.

Для захисту вимірювань від впливу різноманітних електричних або електромагнітних перешкод у ПЛК окрім апаратних засобів захисту від перешкод передбачена можливість використання цифрової фільтрації, настроювання якої виконується у процесі конфігурування кожного каналу. Фільтрація сигналу виконується за формулою:

$$Mes_n = (1-a) \times Val_n + a \times Mes_{n-1},$$

де: a = ефективность фільтра; Val_n = неопрацьоване вхідне значення; Mes_{n-1} = значення, яке передує плинному вхідному значенню; Mes_n = плинне вхідне значення.

Таблиця 2.1

Параметри фільтрації аналогових входів

Бажана ефективність	Значення для вибору	Відповідне значення a	Час відгуку	Частота фільтрації (Гц)
Без фільтру	0	0	0	Апаратний фільтр
Низький рівень фільтрації	1	0,750	111 мс	1,431
	2	0,875	240 мс	0,664
Середній рівень фільтрації	3	0,937	496 мс	0,321
	4	0,969	1,01 мс	0,158
Високий рівень фільтрації	5	0,984	2,03 мс	0,078
	6	0,992	4,08 мс	0,039

У режимі конфігурування, користувач вибирає значення фільтра з 7 можливих значень (від 0 до 6). Це значення потім може бути змінене навіть коли програма знаходиться у режимі RUN.

Опція "Terminal block detection" – виявлення клемної колодки використовується у ПЛК "Premium" для сигналізації відключення гвинтової клемної колодки від модуля аналогового входу

2.5.2.2. Конфігурування модулів аналогових виходів.

Можливості конфігурування модулів аналогових виходів для ПЛК TSX Micro і TSX Premium різні.

Так для модулів TSX ASZ 200 та TSX ASZ 401 ПЛК TSX Micro можливо вибрати тільки в якому стані повинні залишитись значення аналогових виходів у разі аварійної зупинки ПЛК або при переводі його у стан STOP. Можна вибрати два режима:

- установка вихідного каналу у 0 ("Fallback to 0") - по замовчуванню;
- утримання виходів у стані, що передував зупинці ("Maintain state").

При цьому необхідно пам'ятати, що вибір цього параметру одночасно встановлюється для всіх каналів модулів.

Для модулів TSX ASY 410 та TSX ASY 800 настроювання значення в якому вони повинні знаходитись у разі аварійної зупинки ПЛК задаються для кожного каналу окремо. Крім того, для кожного каналу може бути заданий тип і діапазон електричного вихідного сигналу (± 10 В, 0-20 мА, 4-20 мА).

Для модуля TSX ASY 800 додатково можна сконфігурувати джерело живлення модуля: внутрішнє (Internal) або зовнішнє (External. У разі використання зовнішнього джерела живлення, під час конфігурування можна вибрати опцію контролю за цим живленням (24 output supply monitoring).

2.5.3 Конфігурування модулів рахункових каналів

В процесі конфігурування модулів лічильників кожен канал конфігурується окремо.

Для конфігуруванні канала:

- вибирається тип лічильника (прямий, зворотній або реверсивний);
- вказується тип джерела імпульсних сигналів – механічний контакт, сигнал від електронної схеми, кодуючий пристрій;
- для кожного входу вказується фронт імпульсного сигналу (передній або задній) за яким буде проводитись підрахунок імпульсів;
- задається поведінка лічильника при досягненні ним граничних значень або значень уставок (скидання у 0 або у передвстановлене значення, утримання досягнутого значення);
- вказується у якій задачі (MAST або FAST) буде використовуватись даний лічильний канал;
- можна задати номер задачі обробки події що буде запускатись у момент досягнення лічильником уставки

Конфігурування каналів лічильників відбувається у спеціальних вікнах, за допомогою яких всі перелічені процедури виконуються досить просто шляхом вибору відповідних опцій.

2.6 Розробка прикладної програми користувача

2.6.1 Загальні положення

2.6.1.1. Вибір структури програми користувача

Розробка прикладного програмного забезпечення виконується в редакторі Program, структура якого наведена на рис. 2.8.

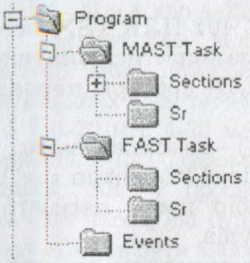


Рис.2.8 Структура програми користувача

Задачі MAST, FAST та Events можуть програмуватись незалежно одна від одної. В свою чергу задачі MAST та FAST можуть складатись з окремих секцій (по аналогії з секціями Ломіконта). Із секцією може бути пов'язаний внутрішній біт, зміна стану якого може включати або відключати секцію (аналогія ключа секції Ломіконта). Для кожної секції необхідно задати ім'я, а також можливо написати коментар. Для захисту програми користувача від несанкціонованого доступу для кожної секції можливо встановити функцію захисту від читання й запису. Задачі MAST та FAST можуть мати підпрограми, кількість яких залежить від типу ПЛК і може сягати 256. Кількість задач з обробки подій також залежить від типу ПЛК і може сягати до 64 для ПЛК TSX Premium.

Як уже зазначалося, програмне забезпечення PL7 Pro відповідає міжнародному стандарту IEC 1131.3 і дає можливість створювати прикладне програмне забезпечення з використанням чотирьох мов програмування: LD – крокових діаграм; IL – список інструкцій; ST – структурованого тексту та Grafset - послідовних функціональних схем.

Особливістю розробки прикладного програмного забезпечення є те, що одночасно в ПрК можуть використовуватись всі перелічені технологічні мови програмування. Тобто при створенні окремої секції, підпрограми чи програми обробки події пропонується вибрати мову програмування для цього фрагмента програми. Виключення складає мова Grafset яку можна використати тільки в основній задачі MAST і тільки в одній із секцій.

2.6.1.2 Типи змінних

У процессі програмування ПЛК використовуються змінні, які насамперед, можна поділити на дві основні групи: зовнішні та внутрішні.

Зовнішні змінні пов'язані з фізичними сигналами, які сприймаються вхідними модулями і позначаються %I або видаються вихідними модулями у вигляді керуючих електричних сигналів і позначаються %Q.

Внутрішні змінні - це об'єкти пам'яті ПЛК, які поділяються на:

- змінні для зберігання проміжної інформації при виконанні прикладної програми (%M);

- константи, інформація в яких на відміну від змінних типу %M не може змінюватися в процесі виконання програми користувача (% K);

- змінні в яких знаходиться інформація, яка характеризує стан виконання ПрК та апаратних засобів ПЛК, які називаються системними і позначаються символами (%S).

Для кожного з типів внутрішніх змінних у ПЛК зарезервована відповідна зона пам'яті.

Кожна зі змінних може представляти, або дискретну, або аналогову інформацію, тому розрізняють наступні формати змінних:

- X – біт;
- B – байт (формат для роботи з ланцюгом символів);
- W – слово одинарної довжини (16 біт);
- D – слово подвійної довжини (32 біта);
- F – слово із плаваючою комою.

2.6.1.3. Адресація каналів.

При адресації зовнішніх змінних застосовується так звана географічна адресація каналів входів-виходів, коли його адреса містить у собі номер позиції, в якій установлений модуль, і номер каналу на цьому модулі.

Оскільки у контролерах TSX Micro базовим при адресації є напівформатний модуль, то під позицією модуля розуміється настановне місце для його розміщення. Так як стандартний модуль займає весь слот цілком, те він займає два настановних місця, тобто дві позиції (рис.2.9).

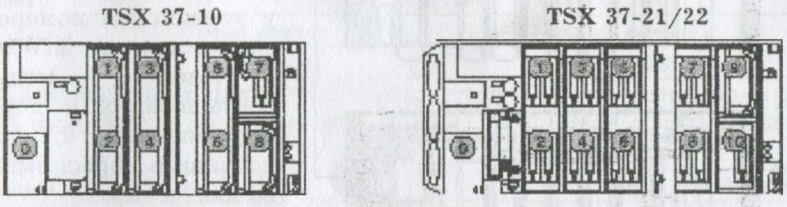
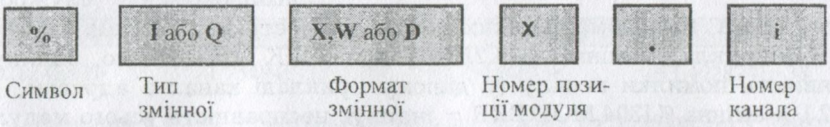


Рис. 2.9 Нумерація позицій модулів TSX Micro

Синтаксис адресації каналів модулів TSX Micro виглядає так:



Наприклад, 5-й вхідний канал дискретного модуля, розташованого в 4-ій позиції має адресу %I4.5, а 3 вихідний канал модуля, розташованого у 6-ій позиції - %QW6.3.

Звертаючись до стандартного модуля необхідно пам'ятати, що він займає дві позиції. Умовно можна представити, що він як би складається із двох половинок, кожна з який займає свою позицію. Тому адреси каналів, що належать одному стандартному модулю можуть мати різні номери позицій.

Наприклад, дискретний модуль, що має 28 входів-виходів (16 входів і 12 виходів) установлений у другому слоті ПЛК займає позиції 3 і 4. У цьому випадку 16 вхідних каналів в адресі мають позицію 3 (від %I3.0 до %I3.15), а 12 вихідних каналів – позицію 4 (від %Q4.0 до %Q4.11).

Інтегровані в процесорний модуль ПЛК TSX 37-22 канали мають такі позначення: 0 – термінальний комунікаційний порт; 1 – комунікаційний порт PCMCIA; від 2 до 9 – вбудовані аналогові входи; 10 – вбудований аналоговий вихід; 11,12 – вбудовані рахункові канали;

Контролери TSX Premium можуть складатися з декількох корзин, тому на відміну від ПЛК TSX Micro, синтаксис адресації зовнішніх змінних TSX Premium включає в себе і номер корзини:

%	I або Q	X,W або D	Y	X	.	i
Символ	Тип змінної	Формат змінної	Номер корзини (0-7)	Номер позиції модуля (0-10)		Номер канала

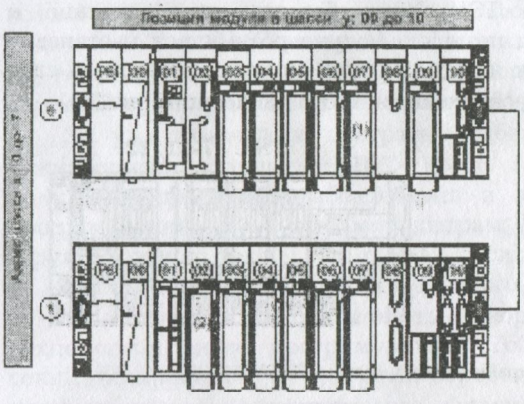


Рис. 2.10 Адресація модулів ПЛК TSX Premium

Наприклад, адреса каналу №4 аналогового вхідного модуля, розташованого на 5 місці у корзині під номером 2 записується як %IW205.4.

Для модулів, розташованих у корзині під номером 0, номер корзини в адресі змінних не вказується.

Крім адресації каналів, з окремими модулями і кана-лами асоціюються службові біти і слова, за допомогою яких можна діагностувати їх стан.

Наприклад, змінна %I4.2ERR, для ПЛК TSX Micro, індикує наявність помилки каналу (у даному прикладі канал з адресом % I4.2.), а змінна %I304.MOD.ERR – індикує несправність усього модуля (у даному прикладі вхідного модуля, який розташований на 4 позиції в корзині №3 ПЛК TSX Premium).

Внутрішні змінні мають такий синтаксис адресації:



Символ | Тип змінної | Формат змінної | Номер змінної

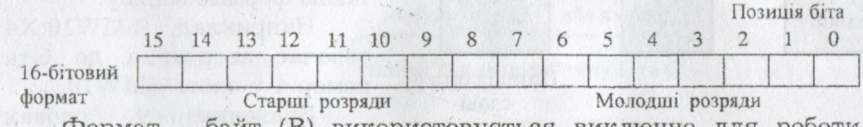
При адресації дискретних змінних %M і %S, формат змінної не вказується. Наприклад, %M10, %S50.

Номер змінної %M або %K лежить у межах визначеної під час конфігурації контролера кількості того чи іншого типу змінної.

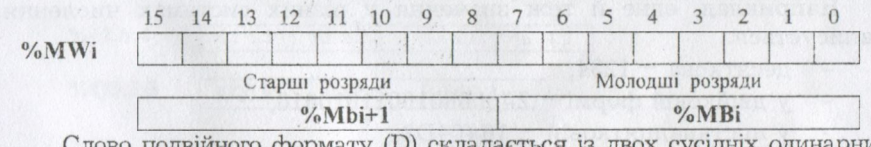
Зміст і призначення окремих системних бітів і слів визначаються системою діагностики та самотестування контролера. У додатках 1 і 2 наведена таблиця призначення основних системних бітів і слів.

2.6.1.4. Формати змінних

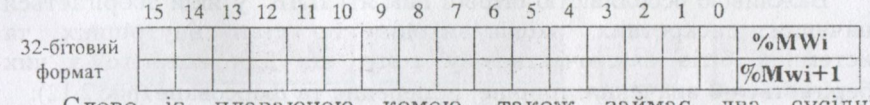
Формати аналогових змінних тісно пов'язані один з одним, так як усі вони використовують одну і ту ж зону пам'яті. Базовим форматом є 16 розрядне слово (W) у якому можна зберігати значення між − 32768 і +32767.



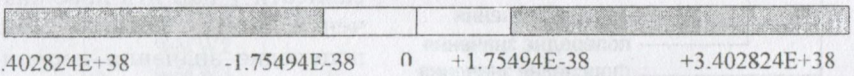
Формат - байт (B) використовується виключно для роботи з ланцюгами символів і складає половину базового слова.



Слово подвійного формату (D) складається із двох сусідніх одинарних слів і в ньому можна зберігати значення між − 2147483648 та +2147483647.



Слово із плаваючою комою, також займає два сусідніх одинарних слова і його значення може знаходитись у діапазоні − 3.402824E+38 та −1.75494E-38 або 1.75494E-38 та 3.402824E+38.



Значення у формі із плаваючою комою можуть бути записані у вигляді: 1285.28, 1,28528E3.

Із наведеної структури використання зони пам'яті (рис.2.11) видно, що подвійні слова і слова із плаваючою точкою, можуть перекривати одне

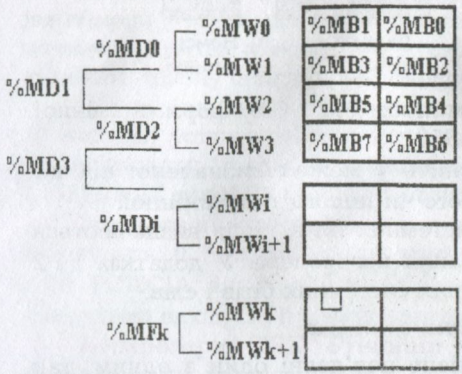
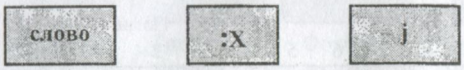


Рис.2.11 Структура подвійного слова

одне. Наприклад, до подвійного слова %MD0 входять одиниці слова %MW0 і %MW1, але до подвійного слова %MD1 також входить слово %MW1. Тобто при використанні слів подвійного формату і слів з плаваючою комою треба це враховувати й уважно слідкувати щоб не спотворити інформацію.

Це відноситься також і для інших типів змінних, а саме констант і зовнішніх змінних.

Якщо необхідно звернутися до окремого біта в середині якогось слова, можемо скористуватися такою формою запису:



Позиція біта у слові

Наприклад, %MW10:X4 означає звертання до біта номер 4 у слові %MW10.

Значення у словах можуть бути записані у десятковому, двійковому або шістнадцятковому численні, або у форматі із плаваючою комою.

Наприклад, одне й теж значення у різних системах числення записується:

- десяткова – 1234;
- у двійковій формі – 2#0000010011010010;
- у шістнадцятковій – 16#04D2;
- у форматі із плаваючою комою (або у реальному вигляді) – 12,34E3.

Важливою особливістю бітової пам'яті ПЛК у якій зберігається значення дискретних входів/виходів та стан внутрішніх та системних бітів заключається у тому, що для кожного з них зберігається 3 значення: плинне, попереднє та форсоване (рис.2.12):

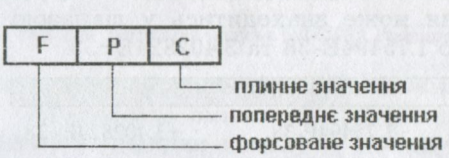


Рис.2.12 Структура бітової пам'яті

У процесі модернізації бітової пам'яті, перед тим як записати у пам'ять нове значення бітової змінної, його попереднє значення переписується у комірки пам'яті призначеного для його зберігання. Наявність цих двох значень дає змогу визначити момент переходу дискретної змінної з 0 на 1 (обробка переднього фронту) або з 1 на 0 (обробка заднього фронту).

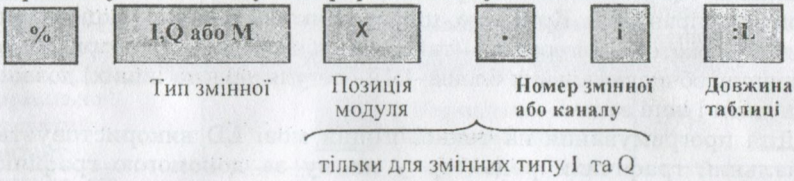
Форсований стан дискретної змінної, це її особистий стан при якому її значення встановлюється оператором з термінала програмування (по аналогії з режимом блокування змінної для ПЛК Ломіконт)

У деяких випадках можна користуватися не прямою, а індексною адресацією змінних. При цьому адреса змінної визначається як сума прямої адреси змінної + значення змінної, яка вказана у вигляді індексу. Наприклад, запис %MW 108 [%MW2] відповідає адресі яка визначається як :

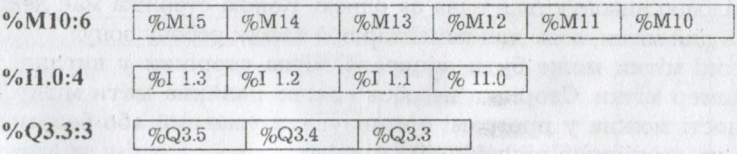
пряма адреса 108 + значення в слові %MW2.

Тобто, якщо у слові %MW2 записано значення 12, то вказана індексна адреса відповідає прямій адресі %MW120 (108+12=120).

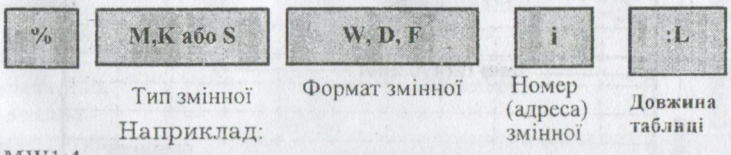
Крім перелічених типів змінних і їх адресації можливо працювати з таблицями бітів та слів. Для запису таблиці бітів використовується наступна форма запису:



Наприклад:



Для запису таблиці слів використовується наступна форма запису:



Наприклад:

%MW1:4

%MW1
%MW2
%MW3
%MW4

2.6.2 Технологічна мова програмування Ladder Diagram (LD)

2.6.2.1 Загальні положення

Мова **Ladder Diagram** (діаграма крокової логіки) у графічному вигляді представляє алгоритм вирішення задач логіко-командного управління. Це одна з мов програмування мікропроцесорних контролерів, які вперше були використані для заміни релейно-контактних схем управління. Тому у вітчизняній техніці ця мова мала саме таку назву "Мова релейно-контактних схем". Ця мова інтуїтивно зрозуміла любому електрику і розрахована на вирішення задач послідовного управління. Але неправильно розуміти, що цю мову можливо використовувати тільки при заміні відомого релейно-контактного варіанта схеми управління. Це графічна форма представлення логіко-командного управління, яке може використовуватись при описі будь-якого алгоритму управління. Крім того, широке використання функціональних блоків з бібліотеки алгоритмів (таймерів, лічильників, регістрів, блоків порівняння, обчислювальних блоків, ПІД- регуляторів та інших) дозволяє вирішувати і інші задачі.

Для програмування на технологічній мові LD використовується спеціальний графічний редактор у якому за допомогою графічних елементів розробляється програма користувача.

Програма складається з окремих сторінок із написаною програмою, які послідовно виконуються одна за одною. Кожна сторінка має декілька зон: зона для мітки, зона для коментарю, а також робочу зону.

В зоні мітки може бути записана мітка сторінки у вигляді %Li, де і – номер мітки. Сторінка не обов'язково повинна мати мітку. При її наявності можна у програмі організувати умовний або безумовний перехід на сторінку з відповідною міткою.

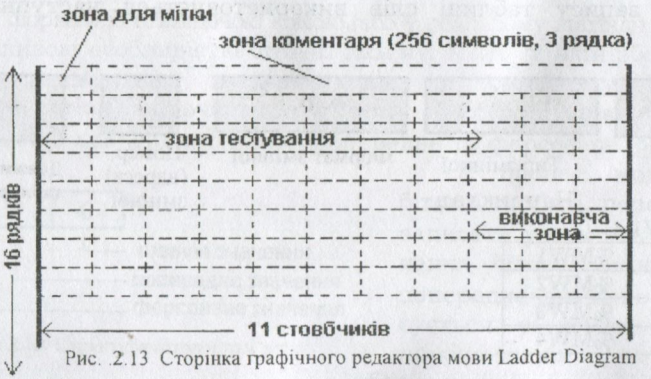


Рис. 2.13 Сторінка графічного редактора мови Ladder Diagram

Робоча зона, у свою чергу, має умовно дві зони: зону тестування і виконавчу зону. Зона тестування прилягає до лівої умовної шини і в цій зоні розташовуються елементи, стан і значення яких обумовлює

логіку виконання програми користувача. В виконавчій зоні, яка прилягає до правої умовної шини, розташовуються елементи, які обумовлюють виконання керуючих дій. Робоча зона сторінки має сітку з 11 стовпчиків і 16 рядків. Програмування полягає у виборі графічних елементів, які відповідають тим чи іншим логічним елементам або функціональним блокам, розташуванні їх у робочих зонах сторінок і показі зв'язків між ними у вигляді традиційних електричних ланцюгів. Вибір і нанесення графічних елементів у полі графічного редактора відбувається за допомогою активних кнопок, на яких нанесене графічне зображення вибраного елементу.

У таблиці 2.2 наведені елементи і їх графічне зображення, які використовуються в мові Ladder Diagram.

Таблиця 2.2

Зображення графічних елементів

Назва елементу	Зображення	Кнопка	Назва елементу	Зображення	Кнопка
Елементи, що розташовуються в тестовій зоні					
Нормально відкритий контакт	⊣ ⊢	F2	Нормально закритий контакт	⊣/⊢	F3
Контакт, що закривається по передньому фронту	⊣P⊢	F4	Контакт, що закривається по задньому фронту	⊣N⊢	F5
Виконуючі елементи					
Котушка на ввімкнення	-( )-	F9	Котушка на вимкнення	-(/)-	F10
Котушка на ввімкнення з защіпкою	-(S)-	F11	Котушка на вимкнення з защіпкою	-(R)-	F12
З'єднуючі елементи					
Горизонтальна лінія коротка	—	F6	Горизонтальна лінія довга	—	F7
Горизонтальна лінія ломана	⌐¬		Вертикальна лінія	|	F8
Елементи переходів					
Перейти на сторінку, яка має мітку %Li	⇉%Li	↑F2	Виклик підпрограми	SRi -(C)-	↑F6
Вихід з підпрограми	⟨RETURN⟩	↑F6	Зупинити виконання програми	⟨HALT⟩	↑F6

Як видно з наведених зображень при програмуванні на мові LD використовуються елементи, які звичайно використовуються при складанні релейно-контактних схем. Але ці зображення слід розуміти

ширше, як зображення логічних виразів. Наприклад, «нормально відкритий контакт» можна трактувати як вираз логічної умови – «якщо спрацював ...», або «якщо включився ...», або «якщо змінна .... перейшла в стан логічної 1» і т.ін. Так саме, елемент «котушка на ввімкнення» можна розглядати як виконавчу частину логічного виразу – «тоді ввімкнути ...» або «тоді змінній присвоїти значення логічної 1 ...» і т.ін.

В свою чергу, елемент "контакт, що закривається по передньому фронту" можна інтерпретувати як логічний вираз: "у момент ввімкнення ...." або "у момент зміни стану дискретної змінної з "0" на "1". Тоді елемент "контакт, що закривається по задньому фронту" можна інтерпретувати як логічний вираз: "у момент вимкнення ...." або "у момент зміни стану дискретної змінної з "1" на "0".

Виконавчі елементи можна інтерпретувати наступним чином:

-"котушка на ввімкнення" - "тоді включити....";

- " котушка на вимкнення" - "тоді відключити....";

- "котушка на ввімкнення з замиканням" – "тоді ввімкнути і запам'ятати стан.";

-"котушка на вимкнення з замиканням " – "тоді вимкнути й запам'ятати стан"

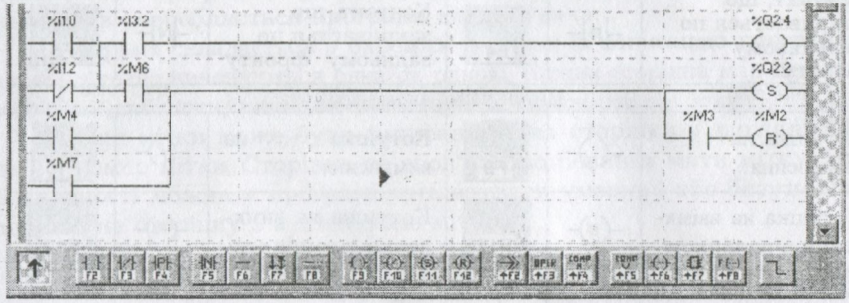
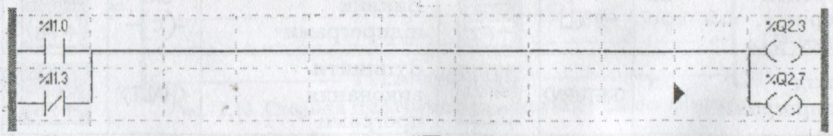


Рис. 2. 14 Фрагмент програми з елементами "катушек з защіпкою"

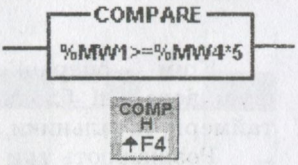
Ці прості логічні вирази можуть утворювати складні. Наприклад, для опису алгоритму роботи: якщо натиснута кнопка «Пуск» або вимкнувся датчик нижнього рівня необхідно ввімкнути насос і відключити клапан подачі повітря, програма на мові LD може виглядати наступним чином:



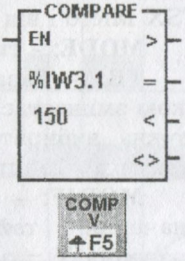
У цьому прикладі: кнопка "Пуск" підключена до дискретного входу з адресою %I1.2, датчик нижнього рівня до %I1.3, а насос і клапан подачі повітря відповідно до дискретних виходів %Q2.4 і %Q2.7

Крім простих елементів у програмі використовуються блоки, які виконують різноманітні функції. До них відносяться блоки порівняння, операційні блоки, таймери, лічильники, одновібратори, регістри, драм-контролери.

Горизонтальний блок порівняння викликається за допомогою кнопки, яка наведена на малюнку і має вигляд горизонтального прямокутника в середині якого записується умова порівняння двох операторів. Блок займає дві клітини. Умови порівняння можуть бути простими і складними та можуть включати до 4096 знаків. Порівняння перевіряється з використанням умов: <, >, =, <=, >=, <>. Блок порівняння розташовується у тестовій зоні і займає дві клітини. Функціонально, на мові релейно-контактних схем, блок порівняння можна порівняти з нормально відкритим контактом, який замикається коли умова, яка записана у блоці порівняння виконується і розмикається, коли умова не виконується.



Вертикальний блок порівняння відрізняється від горизонтального тим, що у ньому одночасно можна перевіряти декілька умов. Операторами для порівняння можуть бути слова різних типів одинарного або под-війного формату, індексовані слова, а також числові значення. У прикладі (рис.2.15) наведений вертикальний компаратор, де записані оператори, які одночасно перевіряються за чотирма умовами. Функціонально, вертикальний блок порівняння нібито складається з чотирьох горизонтальних компараторів, в яких перевіряються різні умови порівняння між одними й тими ж простими операторами. Так, якщо значення змінної %IW3.1 >150 то вмикаються виходи %Q2.1 та %Q2.4, якщо значення змінної %IW3.1 <150 то вмикаються виходи %Q2.3 та %Q2.4, , а якщо значення



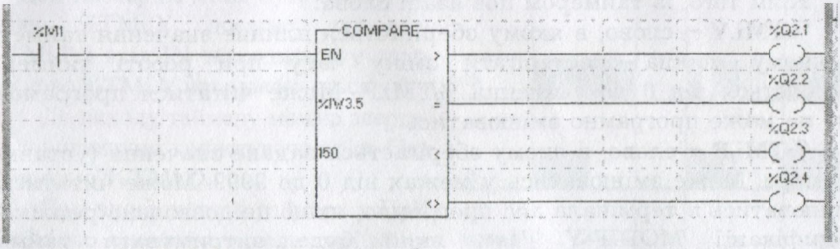


Рис. 2.15 Фрагмент програми з блоком вертикального компаратора

змінної %IW3.1 =150, то вмикається вихід %Q2.2.

Для виконання арифметичних і логічних операцій над змінними й числовими значеннями використовується **операційний блок**, який займає 4 клітини і в якому може бути записаний вираз з 4056 знаків.

Операційний блок відноситься до виконавчих елементів і розміщується біля лівої шини. Якщо операційному блоку передують тестові елементи, то вираз, який записаний у операційному блоці виконується за умови виконання відповідних умов, а якщо операційний блок не має елементів в тестовій зоні він виконується безумовно.

2.6.2.2. Таймери

Крім операційних блоків у ПрК використовуються функціональні блоки, які виконують найбільш поширені функції: таймери, лічильники, одновібратори, регістри

Розрізняють три типи таймерів (MODE) – TON, TOF та TP:

TON – призначений для організації затримки на вмикання;

TOF – призначений для організації затримки на вимикання;

TP – призначений для організації імпульсу заданої тривалості.

Графічно таймер відображається на схемі у вигляді прямокутника на якому можна побачити його вхід та вихід, а також настроєння:

%TMi – номер таймера, де **i** може змінюватись від 0 до 63 для TSX Micro і від 0 до 255 для TSX Premium.

MODE: - тип таймера: TON, TOF або TP;

TB: - база часу, показує дискретність із якою змінюється відлік часу в таймері. Базу часу можна вибирати з ряду: 1 хвилина (min), 1 s (секунда), 100 ms (мілісекунд), 10 ms

MODIF: – можливість модифікації заданого значення таймера. Якщо встановлюється значення **Y** – задане значення таймера (%TMi.P) може змінюватись у процесі виконання програми з терміналу або програмним шляхом. Якщо встановлюється значення **N** – задане значення може змінюватись тільки в режимі редагування програми.

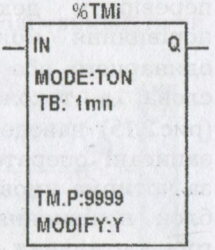


Рис.2.16
Блок таймера

Крім того, із таймером пов'язані слова:

%TMi.V – слово, в якому зберігається плинне значення таймера в якому можна спостерігати зміну часу при роботі таймера. Змінюється від 0 до значення %TMi.P. Може читатися програмою, але не може програмно змінюватись.

%TMi.P – слово, в якому зберігається задане значення (уставка) таймера. Може змінюватись у межах від 0 до 9999. Може читатись і змінюватись з термінала або програмно, якщо це дозволено режимом модифікації MODIF:Y. Час, який буде витримувати таймер розраховується як: **%TMi.P x TB.**

Розглянемо діаграми роботи таймерів різних типів.

Таймер типу TON стартує по передньому фронту сигналу на вході таймера **IN.** Поточне значення таймера %TMi.V змінює своє значення від 0 до %TMi.P з кроком, який дорівнює базі часу таймеру TB. Вихід таймера **Q** змінюється на 1, коли поточне значення таймера %TMi.V стане

дорівнювати уставці таймера %TMi.P і буде зберігати це значення до моменту поки на вході таймера IN зберігається стан 1. Як тільки на вході IN з'являється 0, таймер зупиняється, поточне значення %TMi.V скидається в значення 0 (навіть у тому випадку коли значення %TMi.V не досягло значення %TMi.P.

При роботі **таймера типу TOF** плинне значення %TMi.V встановлюється в 0 по передньому фронту на вході IN (навіть у тому випадку коли таймер находиться у режимі відліку часу). При виникненні заднього фронту на вході IN, таймер запускається. Поточне значення таймера %TMi.V змінює своє значення від 0 до %TMi.P з кроком, який дорівнює базі часу таймеру TB. Вихідне значення таймера Q переходить у 1 по передньому фронту на вході IN і повертається у 0 коли плинне значення %TMi.V досягне значення %TMi.P.

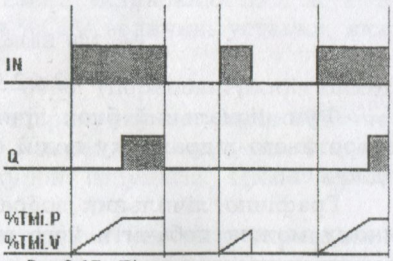


Рис 2.17 Діаграма спрацьовування таймера типу TON

Таймер типу TP запускається по передньому фронту на вході IN. При цьому плинне значення %TMi.V встановлюється в 0, якщо таймер вже не перебуває у режимі відліку часу. Поточне значення таймера %TMi.V змінює своє значення від 0 до %TMi.P з кроком, який дорівнює базі часу таймеру TP. Вихідне значення таймера переходить в 1 коли таймер запускається й повертається у 0, коли плинне значення досягне значення %TMi.P.

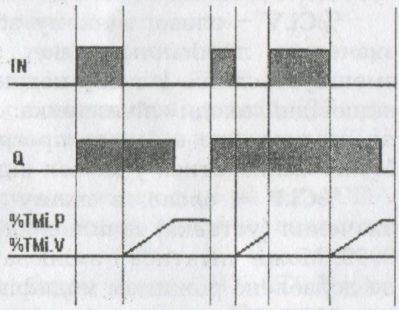


Рис.2..18 Діаграма спрацьовування таймера типу TOF

Цей таймер не може бути скинутий. Коли %TMi.V досягне значення %TMi.P і вхід IN знаходиться у стані 0, слово %TMi.V встановлю-ється у 0.

До виходу таймеру можна звертатись у програмі користувача як до біта з адресою **%TMi.Q.**

При використанні таймера у ПрК необхідно спочатку у редакторі змінних (Variabls) задати для вибраного таймера %TMi) його тип (TON, TOF або TP), задати базу часу (TB) і уставку часу (%TMi.P), а також задати режим модифікації.

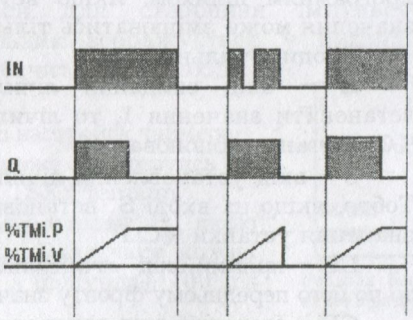


Рис.2.19 Діаграма спрацьовування таймера типу TP

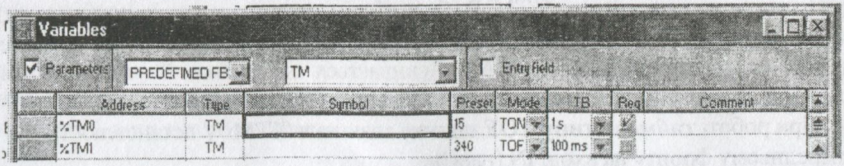


Рис. 2.20 Вікно конфігурації таймерів

2.6.2.3. Лічильники

Функціональний блок лічильника призначений для прямого та зворотнього підрахунку подій. Ці два режими можуть виконуватись одночасно.

Графічно лічильник зображується у вигляді прямокутника на якому можна побачити його входи та виходи, а також настроєння.

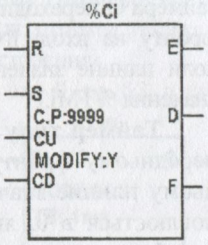


Рис.2.21
Блок лічильника

%Ci – номер лічильника, де **i** може змінюватись від 0 до 31 для Micro і від 0 до 255 для Premium.

%Ci.V – слово, в якому зберігається плинне значення лічильника, яке збільшується або зменшується на 1 коли подаються імпульси на відповідні входи лічильника. Може читатися програмою, але не може програмно змінюватись. Може змінюватись у межах від 0 до 9999.

%Ci.P – слово, в якому зберігається задане значення (уставка) лічильника. Може змінюватись у межах від 0 до 9999. Може читатись і змінюватись з термінала або програмно, якщо це дозволено режимом модифікації MODIF:Y.

MODIF: - можливість модифікації уставки лічильника. Якщо встановлюється значення **Y** – задане значення лічильника (%Ci.P) може змінюватись в процесі виконання програми з терміналу або програмним шляхом. Якщо встановлюється значення **N** – задане значення може змінюватись тільки в режимі редагування програми.

Входи лічильника:

R – вхід скидання лічильника. Тобто, якщо на вході R встановити значення 1, то лічильник скидається і значення в слові %Ci.V стане дорівнювати 0.

S – вхід установлення лічильника у передустановлене значення. Тобто, якщо на вході S встановити 1, то в слово %Ci.V запишеться значення уставки %Ci.P.

CU – прямий вхід лічильника. Якщо на вхід CU подається імпульс, то по його передньому фронту значення слова %Ci.V збільшиться на 1.

CD – зворотній вхід лічильника. Якщо на вхід CD подається імпульс, то по його передньому фронту значення слова %Ci.V зменшиться на 1.

Виходи лічильника:

E – вихід досягнення значення 0. Вихід установлюється в 1 коли при зворотньому підрахунку імпульсів поточне значення %Ci.V досягне значення 0 і перейде у значення 9999 коли лічильник продовжує зворотній підрахунок імпульсів. Після цього вихід E знову встановлюється в 0.

D – вихід досягнення уставки. Вихід встановлюється в 1 в момент досягнення поточного значення %Ci.V величині уставки, яка записана в слові %Ci.P.

F – вихід досягнення значення 9999 (переповнення). Вихід встановлю-ється в 1 коли при прямому підрахунку імпульсів поточне значення %Ci.V досягне значення 9999 і перейде у значення 0 коли лічильник продовжує прямий підрахунок імпульсів. Після цього вихід F знову встановлюється в 0.

Виходи E, D і F можуть використовуватись для управління процесом підрахунку імпульсів і створення каскадів із лічильників, коли кількість імпульсів, які необхідно підрахувати більше ніж 9999.

З виходами E, D і F асоційовані відповідні біти %Ci.E, %Ci.D і %Ci.F до яких можна звертатись у програмі користувача.

Лічильник може працювати у трьох режимах: прямому, зворотньому та реверсивному підрахунку імпульсів.

При прямому підрахунку - імпульси подаються на вхід CU і по передньому фронту кожного з цих імпульсів поточне значення %Ci.V збільшується на 1.

При зворотньому підрахунку - імпульси подаються на вхід CD і по передньому фронту кожного з цих імпульсів поточне значення %Ci.V зменшується на 1.

У реверсивному лічильнику задіяні обидва входи CU і CD. Входи проглядаються послідовно. Якщо імпульси на входи CU і CD потрапили одночасно, значення поточного значення %Ci.V – не зміниться.

2.6.2.4. Одновібратори (Monostable %MNi)

Одновібратори використовуються для створення імпульсу заданої тривалості. Графічно лічильник зображується у вигляді прямокутника на якому можна побачити його вхід , вихід, а також настроєння.

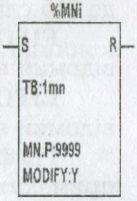


Рис.2.22 Блок одновібратора

Настроєнняя одновібратора подібні до настройок таймера.

%MNi – номер одновібратора, де **i** може змінюватись від 0 до 31 для TSX Micro і від 0 до 255 для TSX Premium.

TB: - база часу, показує дискретність із якою змінюється відлік часу в одновібраторі. Базу часу можна вибирати з ряду: 1 хвилина (min), 1 s (секунда), 100 ms (мілісекунд), 10 ms

MODIF: - можливість модифікації заданого значення одновібратора. Якщо встановлюється значення **Y** – задане значення одновібратора (%MNi.P) може змінюватись в процесі виконання

програми з терміналу або програмним шляхом. Якщо встановлюється значення **N** – задане значення може змінюватись тільки в режимі редагування програми.

%MNi.V – слово, в якому зберігається плинне значення одновібратора в якому можна спостерігати зміну часу при роботі одновібратора. Змінюється від 0 до значення %MNi.P. Може читатися програмою, але не може програмно змінюватись.

%MNi.P – слово, в якому зберігається задане значення (уставка) одновібратора. Може змінюватись у межах від 0 до 9999. Може читатись і змінюватись з терміналу або програмно, якщо це дозволено режимом модифікації MODIF:Y. Час тривалості імпульсу, який буде витримувати одновібратора розраховується як: **%MNi.P x TB.**

S - вхід запуску одновібратора. По передньому фронту імпульсу на даному вході поточне значення приймає значення уставки (%MNi.V = %MNi.P), яке потім зменшується до 0.

R - вихід одновібратора, який встановлюється в 1, коли працює одновібратор, і набуває значення 0, коли поточне значення %MNi.V стане дорівнювати 0.

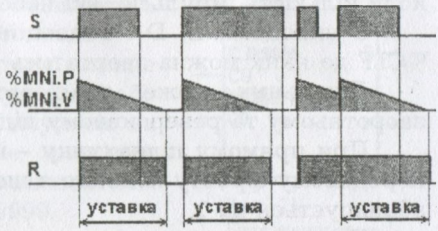


Рис.2.23 Діаграма роботи одновібратора

Коли на вхід S ("Запуск") подасться сигнал, то одновібратор починає працювати: поточне значення стає рівним уставці одновібратора (%MNi.V = %MNi.P) і вихід R встановлюється у значення 1.

Далі, протягом роботи одновібратора, його поточне значення зменшується до тих пір, поки не досягне значення 0. При цьому, вихід R також скидається у значення 0, тобто, після запуску одновібратора на його виході формується імпульс заданої тривалості.

2.6.2.5. Регістри (Register %Ri)

Регістр - це блок пам'яті, який використовується для зберігання до 255 слів (по 16 біт) за одним з двох можливих режимів:

FIFO - «першим прийшов - першим вийшов», відомий як "черга".

LIFO – «останнім прийшов - першим вийшов, відомий як "стек".

Графічно регістр зображується у вигляді прямокутника на якому можна побачити його входи та виходи, а також настроєння (рис.2.24)..

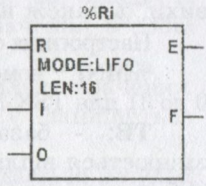


Рис.2.24 Блок регістра

%Ri – номер регістра, де **i** може змінюватись від 0 до 3 для TSX Micro і від 0 до 255 для TSX Premium.

MODE - режим роботи регістра (FIFO або LIFO).

LEN - кількість можливих слів у регістрі (1<=LEN<=255).

%Ri.I - вхідне слово регістра.

%Ri.O - вихідне слово регістра.

I – вхід, по команді, на якому значення вхідного слова %Ri.I записується у регістр.

O - вхід, по команді, на якому слова з регістра записується у вихідне слово %Ri.O.

R – вхід, по команді на якому регістр очищається й переходить у початковий стан.

E – вихід, який установлюється в 1 коли регістр пустий.

F - вихід, який установлюється в 1 коли регістр повний.

Нижче наведені принципи дії регістрів у режимах LIFO та FIFO.

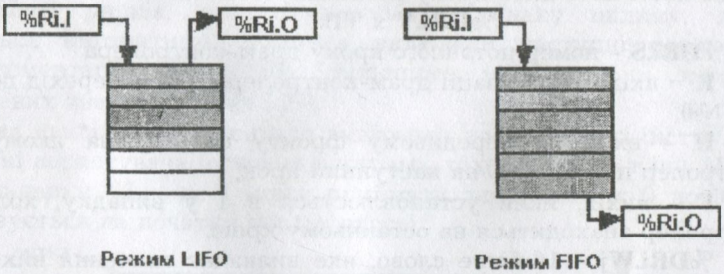


Рис.2.25 Принципи дії регістрів

У режимі LIFO по команді запису на вході **I** значення зі слова %Ri.I записується у регістр. По наступній команді запису нове значення зі слова %Ri.I записується у регістр, а попереднє слово зміщується на одну комірку вниз. І так далі. По команді читання у вихідне слово %Ri.O записується сло-во, яке записане у верхній частині регістра, тобто слово, яке було записане у регістр останнім. Звідси і назва режиму - «останнім прийшов - першим вий-шов». Тобто слова ніби заштовхуються у регістр, а виштовхує їх пружина.

У режимі FIFO при записі слів у регістр вони нібито проштовкуються через всі комірки регіструа і виходять з регістра з іншого боку,. Таким чином, першим із регістра буде зчитане слово, яке було записано в нього першим. Звідси назва - «першим прийшов - першим вийшов».

2.6.2.6 Драм-контролер (Drum controller %DRi)

Драм-контролер працює на принципі подібному до електромеханічного барабанного командоапарата, що складається з групи кулачків розташованих на одному барабані і положення яких можна змінювати. Кожний кулачок може вмикати один кінцевий вимикач. Коли барабан обертається кулачки переключають кінцеві вимикачі. Положення кулачків виставляється таким чином, щоб у задані моменти часу спрацьовували кінцеві вимикачі у заданій комбінації.

Драм-контролер виконує ту саму функцію, але комбінація спрацьовування дискретних виходів задається програмно.

%DRi – номер драм-контролера, де **i** може змінюватись від 0 до 7 для TSX Micro і від 0 до 255 для TSX Premium.

LEN – кількість кроків драм-контролера, яка може вибиратись від 1 до16.

TB: - база часу, показує дискретність із якою змінюється відлік часу в драм-контролері і вибирається з ряду: 1 хвилина (min), 1 s (секунда), 100 ms (мілісекунд), 10 ms

%DRi.V – відлік часу поточного кроку. Час трива-лості поточного кроку розраховується як:

%DRi.V x TB.

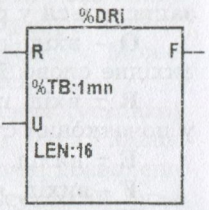


Рис.2.26 Блок драм-контролера

%DRi.S - номер поточного кроку драм-контролера.

R – вхід ініціалізації драм-контролера, тобто перехід до кроку №0.

U - вхід, по передньому фронту сигналу на якому, драм-контролер переходить на наступний крок;

F - вихід, який установлюється в 1 у випадку, коли драм-контролер знаходиться на останньому кроці.

%DRi.Wj - 16-бітне слово, яке визначає значення виходів на **j** кроці драм-контроллера %Dri.

У процесі програмування у редакторі змінних заповнюється матриця, згідно з якою встановлюються значення бітів у вихідному слові, із якими асоціюються зовнішні або внутрішні дискретні змінні.

У наведеному прикладі на нульовому кроці включені виходи: %Q2.1, %Q2.6, %M2, %M4, %M8, %Q2.3. На другому кроці - %Q2.4, %M1, %Q4.5, %Q4.7, %M22, %Q2.3, %M34. І так далі.

Кроки драм-контролера

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**Адреси**
0	1	0	1	0	1	0	1	1	1	0	1	0	0	0	1	0	%Q2.1
1	0	1	1	0	1	1	0	1	0	0	1	0	0	0	1	0	%Q2.4
2	0	1	0	0	1	1	1	1	0	0	1	0	0	0	1	0	%M1
3	1	0	0	0	1	1	1	1	1	0	1	0	1	1	1	0	%Q2.6
4	0	0	0	1	0	0	1	0	1	0	1	0	1	1	1	0	%Q4.1
5	1	0	0	1	0	1	1	0	0	0	0	0	1	1	0	0	%M2
6	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	%M4
7	0	1	0	0	1	0	0	0	1	1	1	0	1	0	1	0	%Q4.5
8	0	1	0	1	0	0	1	1	0	1	1	1	1	0	0	0	%Q4.7
9	0	0	0	0	1	0	1	1	0	1	0	1	0	1	0	1	%M3
A	1	0	0	1	1	0	0	1	1	1	0	1	0	1	0	1	%M8
B	0	0	0	1	0	0	0	0	1	0	1	1	1	0	0	1	%Q4.6
C	0	1	0	1	0	0	1	1	0	0	1	1	0	1	0	1	%M22
D	0	0	1	0	1	0	1	0	0	0	1	1	1	1	0	1	%M15
E	1	1	1	0	1	0	0	1	1	0	0	0	1	0	1	0	%Q2.3
F	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0	0	%M34

Значення бітів вихідного слова

2.6.3. Мова структурованого тексту (Structured Text)

2.6.3.1. Загальні положення

Мова структурованого тексту (ST) одна з можливих мов програмування на контролерах фірми SCHNEIDER. Програма на мові структурованого тексту, подібна до правил написання на відомих алгоритмічних мовах, таких як PASCAL, C, BASIC і т.п. і складається з програмних рядків, написаних із використанням відповідних правил побудови, інструкцій, стандартних процедур, зарезервованих слів і мнемонічних позначень, які визначають алгоритм обробки змінних різних типів.

Текст програми на мові структурованого тексту організована в послідовності рядків, яка починається з знаку оклику, який генерується автоматично, і може включати наступні елементи: мітку, коментарі і одну або більшість інструкцій і команд, розподілених знаком ";" (рис.2.27).

Мітка використовується для виділення послідовності інструкцій у програмі користувача (основна програма, підпрограма і т ін.). Мітка має позначення %Li:, де **i** може приймати значення від 0 до 999 і розташовується на початку послідовності.

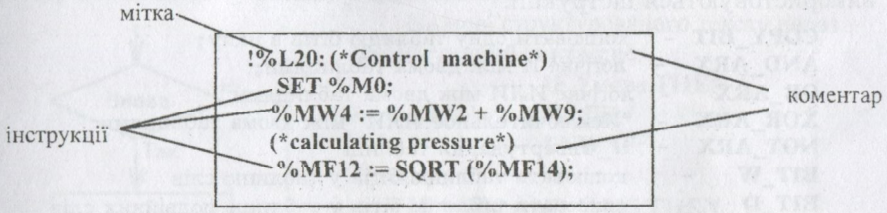


Рис.2.27 Фрагмент програми на мові Structured Text

Коментар може бути вставлений у любе місце програми й кількість їх не обмежена. Коментар може включати до 256 символів і записуватись у декілька рядків. Коментар розташовується між знаками (* та *). Коментарі зберігаються у пам'яті програми користувача.

Програма користувача складається з послідовності **інструкцій**, які розділяються одна від одної знаком ";".

У структурованому тексті використовуються наступні інструкції:

- інструкції над бітами;
- арифметичні й логічні операції над словами і подвійними словами;
- арифметичні інструкції над словами у форматі із плаваючою комою;
- інструкції порівняння слів у різних форматах;
- інструкції перетворення форматів чисел;
- інструкції з таблицями бітів і слів різних форматів;
- інструкції з послідовностями символів;
- алфавітно-цифрове перетворення;

• програмні інструкції;
• управляючі інструкції;
• інструкції зі стандартними функціональними блоками;
• спеціальні інструкції (зв'язку, ПІД-регулятори і т.ін.).

Для роботи над бітами використовуються наступні інструкції:

:= - операція присвоєння;
OR - логічне "ИЛИ";
AND - логічне "И";
XOR - логічне "исключающее ИЛИ";
NOT - інвертування;
RE - обробка переднього фронту;
FE - обробка заднього фронту;
RESET - скидання в "0";
SET - встановлення в "1".

В інструкціях порівняння значень слів у різних форматах використовуються наступні знаки порівняння: >, <, =, <=, =>, <>.

Під час роботи з таблицями бітів можна в таблицю бітів записати значення іншої таблиці , слова, подвійні слова і навпаки - біти таблиці прис-воїти слову або подвійному слову. Крім того, використовуються інструкції:

COPY_BIT - копіювати одну таблицю бітів в іншу;
AND_ARX - логічне И між двома таблицями;
OR_ARX - логічне ИЛИ між двома таблицями;
XOR_ARX - "Исключительное ИЛИ" між двома таблицями
NOT_ARX - інвертування таблиці
BIT_W - копіювати таблицю бітів у таблицю слів
BIT_D - копіювати таблицю бітів у таблицю подвійних слів
W_BIT - копіювати таблицю слів у таблицю бітів
D_BIT - копіювати таблицю подвійних слів у таблицю бітів

За цілочислових арифметичних операцій зі словами подвійної довжини використовуються арифметичні знаки: "+" - додавання, " – " - віднімання, " * " - множення, "/" – ділення. А також інструкції:

REM - залишок від цілочисельного ділення;
SQRT - цілочисельний корінь квадратний
ABS - абсолютне значення
INC - інкрементування
DEC - дікрементування

Для роботи зі словами в форматі з плаваючою комою використовуються ті ж самі інструкції (крім REM, INC, DEC), але в результаті отримаємо реальні числа.

Інструкції для роботи з таблицями слів і подвійних слів призначені для обміну словами між таблицями, арифметичні операціями між словами таблиць, операції порівняння таблиць і операції пошуку в таблицях рівних, максимальних і мінімальних значень.

До програмних інструкцій належать:

HALT - зупинка виконання програми
JUMP - перехід на мітку
Sri - виклик підпрограми за номером **i**
RETURN - вихід із підпрограми
MASKEVT - маскування подій у ПЛК
UNMASKEVT - демаскування подій у ПЛК

2.6.3.2. Керуючі структури

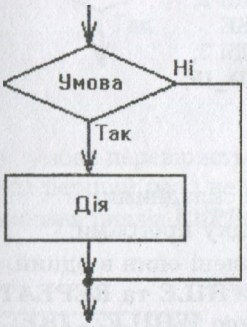
Використовують три керуючі структури:

- умовне виконання дії **IF**;
- умовний вираз повторюваних дій **WHILE** та **REPEAT**;
- повторювані дії **FOR**

Кожна керуюча структура включена між ключовими словами на початку і в кінці інструкції. Це дає можливість відрізнити одну команду від іншої, незалежно від типа. Керуюча структура може передувати або слідувати за іншою командою.

Умовне виконання дії IF

Умовний вираз IF ... THEN. В даному випадку **Дії** виконуються лише у випадку виконання **Умови**.



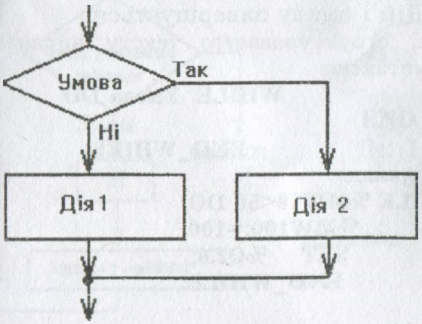
На мові структурованого тексту вираз має наступний синтаксис:

IF Умова **THEN**
Дії;
END_IF;

Приклад:

```
IF %M0 AND %M5 THEN
   RESET %M0;
   SET  %Q2.0;
    %MW:=0;
END_IF;
```

Умовний вираз IF ... THEN...ELSE. В даному виразі, якщо виконується **Умова**, то виконуються **Дії 2**, інакше виконуються **Дії 1**.



На мові структурованого тексту вираз має наступний синтаксис:

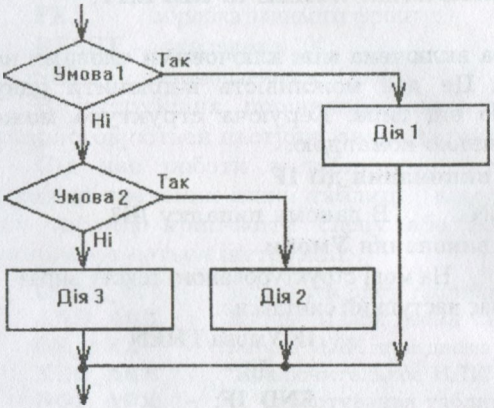
```
IF Умова THEN
   Дії 2;
ELSE
   Дії 1;
END_IF;
```

Умовний вираз IF ... THEN...ELSIF;

В даному виразі існує ланцюг **Умов** і **Дій**, які повинні виконуватись. Перегляд **Умов** виконується послідовно. Якщо виконується перша **Умова і**, то виконуються **Дії і**, подальший перегляд **Умов** закінчується і програма переходить до виконання інструкцій, які записані після слова **END_IF.** Інакше перевіряється наступна умова **Умова (і+1)** і т.д.

Якщо ж не виконалась жодна умова, то виконуються команди, які записані після інструкції **ELSE** (яка зовсім не обов'язкова і при її відсутності нічого не виконується).

Необхідно пам'ятати, що на кількість **ELSIF** у виразі обмежень не має, а команда **ELSE** може бути лише одна.



На мові структурованого тексту вираз має наступний синтаксис:

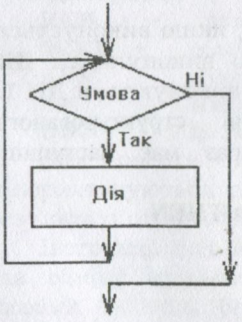
IF Умова1 **THEN**
Дії 1;
ELSIF Умова 2 **THEN**
Дії 2;
ELSE
Дії 3;
END_IF;

У всіх умовних виразах **IF** :

- умови можуть бути як простими так і складними;
- кожна дія записується в окремому рядку програми;
- декілька виразів **IF** можуть бути вкладені один в одний.

Умовний вираз дій, які повторюються WHILE та REPEAT

Цикл з передумовою WHILE ... DO;



Дії виконується поки виконується **Умова**. Як тільки **Умова** перестає виконуватись виконання **Дій і** циклу завершується.

На мові структурованого тексту вираз має наступний синтаксис:

WHILE Умова **DO**
Дії;
END_WHILE;

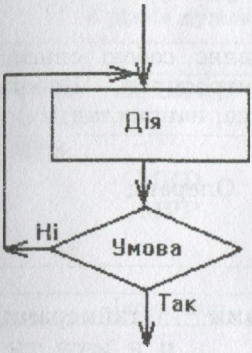
Приклад:

```
WHILE %MW10<50 DO
      %MW100:=100;
      SET   %Q2.0;
      END_WHILE;
```

- умови можуть бути як простими так і складними;
- кожна дія записується в окремому рядку програми;
- умова перевіряється до моменту виконання Дії. Якщо умова перевіряється перший раз і не виконується - то Дії не виконуються;
- декілька циклів **WHILE .. DO** можуть бути вкладені один у одний.

Цикл з після умовою REPEAT .. UNTIL;

Набір **Дій** виконується до моменту виконання **Умови**. Як тільки **Умова** виконається - цикл завершується. (Дії в середині циклу виконуються принаймні один раз).



На мові структурованого тексту вираз має наступний синтаксис:

REPEAT
Дії;
UNTIL Умова
END_REPEAT;

Приклад:
REPEAT
%M5:=FALSE;
SET %Q3.1;
%MW10:=125;
UNTIL
MW1<100
END_REPEAT;

- умови можуть бути як простими так і складними;
- кожна дія записується в окремому рядку програми;
- умова перевіряється після виконання Дії. Якщо умова перевіряється перший раз і не виконується - то Дії виконаються ще один раз;

декілька циклів **REPEAT .. UNTIL** можуть бути вкладені один у одний

Цикл із повторенням заданої кількості разів FOR...TO...DO;



Набір **Дій** виконується до тих пір, поки індекс циклу не пройде значення від **I** до **J** (індексом може бути тільки слово).

На мові структурованого тексту вираз має наступний синтаксис:

FOR Індекс:= **I TO J DO**
Дії;
END_FOR;

Де: **I** *(початкове значення)*
J *(кінцеве значення)*

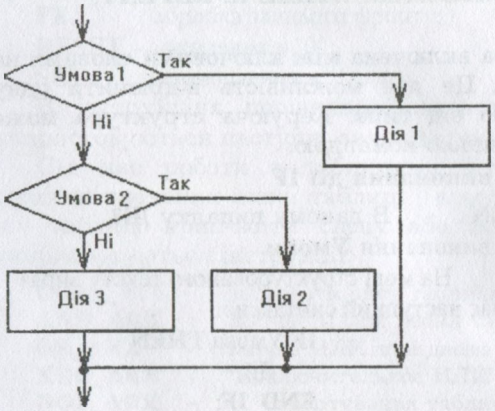
Приклад:
FOR %MW9:=**0** TO **90** DO
%M1:=TRUE;
SET %Q4.1;
END_FOR;

Умовний вираз IF ... THEN...ELSIF;

В даному виразі існує ланцюг **Умов** і **Дій**, які повинні виконуватись. Перегляд **Умов** виконується послідовно. Якщо виконується перша **Умова і**, то виконуються **Дії і**, подальший перегляд **Умов** закінчується і програма переходить до виконання інструкцій, які записані після слова **END_IF.** Інакше перевіряється наступна умова **Умова (і+1)** і т.д.

Якщо ж не виконалась жодна умова, то виконуються команди, які записані після інструкції **ELSE** (яка зовсім не обов'язкова і при її відсутності нічого не виконується).

Необхідно пам'ятати, що на кількість **ELSIF** у виразі обмежень не має, а команда **ELSE** може бути лише одна.



На мові структурованого тексту вираз має наступний синтаксис:

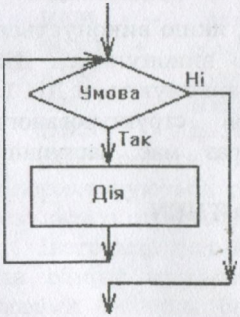
IF Умова1 **THEN**
  Дії 1;
ELSIF Умова 2 **THEN**
  Дії 2;
ELSE
  Дії 3;
END_IF;

У всіх умовних виразах **IF** :

- умови можуть бути як простими так і складними;
- кожна дія записується в окремому рядку програми;
- декілька виразів **IF** можуть бути вкладені один в одний.

Умовний вираз дій, які повторюються WHILE та REPEAT

Цикл з передумовою WHILE ... DO;



Дії виконується поки виконується **Умова**. Як тільки **Умова** перестає виконуватись виконання **Дій і** циклу завершується.

На мові структурованого тексту вираз має наступний синтаксис:

WHILE Умова **DO**
  Дії;
END_WHILE;

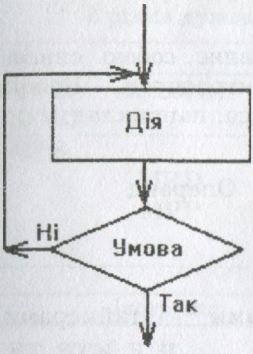
Приклад:

```
WHILE %MW10<50 DO
    %MW100:=100;
    SET   %Q2.0;
    END_WHILE;
```

- умови можуть бути як простими так і складними;
- кожна дія записується в окремому рядку програми;
- умова перевіряється до моменту виконання Дії. Якщо умова перевіряється перший раз і не виконується - то Дії не виконуються;
- декілька циклів **WHILE .. DO** можуть бути вкладені один у одний.

Цикл з після умовою REPEAT .. UNTIL;

Набір **Дій** виконується до моменту виконання **Умови**. Як тільки **Умова** виконається - цикл завершується. (Дії в середині циклу виконуються принаймні один раз).



На мові структурованого тексту вираз має наступний синтаксис:

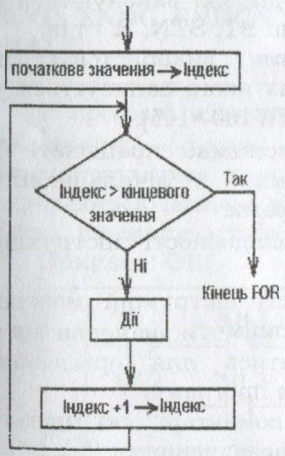
REPEAT
Дії;
UNTIL Умова
END_REPEAT;

Приклад:
```
REPEAT
   %M5:=FALSE;
   SET   %Q3.1;
   %MW10:=125;
UNTIL
MW1<100
END_REPEAT;
```

- умови можуть бути як простими так і складними;
- кожна дія записується в окремому рядку програми;
- умова перевіряється після виконання Дії. Якщо умова перевіряється перший раз і не виконується - то Дії виконаються ще один раз;

декілька циклів **REPEAT .. UNTIL** можуть бути вкладені один у одний

Цикл із повторенням заданої кількості разів FOR...TO...DO;



Набір **Дій** виконується до тих пір, поки індекс циклу не пройде значення від **I** до **J** (індексом може бути тільки слово).

На мові структурованого тексту вираз має наступний синтаксис:

FOR Індекс:= **I TO J DO**
Дії;
END_FOR;
Де: **I** *(початкове значення)*
J *(кінцеве значення)*

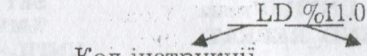
Приклад:
```
FOR %MW9:=0 TO 90 DO
    %M1:=TRUE;
     SET   %Q4.1;
END_FOR;
```

- коли індекс стає більшим ніж його кінцеве значення виконуються команди, які йдуть за ключовим словом **END_FOR**;
- індекс інкрементується автоматично і програміст не несе відповідальності за його поточне значення;
- кожна дія записується в окремому рядку програми;
- початкове й кінцеве значення індексу повинне бути значенням слова у цифровому виразі;
- декілька циклів **FOR..TO..DO** можуть бути вкладені один в одний.

2.6.4 Мова інструкцій (Instruction List)

2.6.4.1 Загальні положення

Програма на мові Instruction List (IL) уявляє собою список інструкцій, які послідовно виконуються контролером. Кожна інструкція складається з коду інструкції і операнда, наприклад:



Ці інструкції виконують дії з:

- входами-виходами контролера;
- стандартними функціональними блоками (таймерами, лічильниками, регістрами і т.ін.);
- арифметичними й логічними виразами;
- внутрішніми змінними.

Розрізняють два типи інструкцій:

- Тестові інструкції у яких аналізуються умови, які необхідно виконати для виконання відповідних дій: LD, AND, OR і т.ін.
- Виконавчі інструкції, які визначають дії, які виконуються в разі виконання умов, записаних у тестовій зоні: ST, STN, R і т.ін.

При роботі з числовими інструкціями використовується синтаксис мови структурованого тексту, вираз якого записується у квадратних дужках [.....]. Наприклад, LD [%MW100>145].

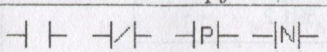
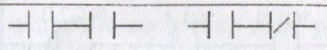
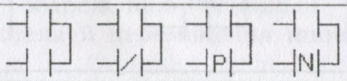
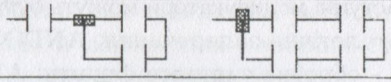
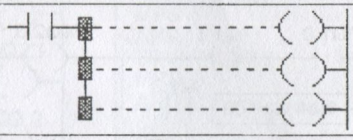
У процесі програмування на мові IL можливо працювати з функціональними блоками: таймерами, лічильниками, одновібраторами, регістрами й драм-контролерами.

У написанні програми на початку послідовності інструкцій автоматично генерується знак оклику !.

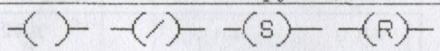
Всередені програми, окремі послідовності інструкцій можуть мати мітку, яка має синтаксис Li, де і може приймати значення від 0 до 999. Ці мітки можуть використовуватись для організації запрограмованого переходу до окремих частин програми.

В тілі програми можуть бути записані коментарі, які можуть включати до 222 символів і записуються між позначеннями (* .....*).

Тестові базові інструкції

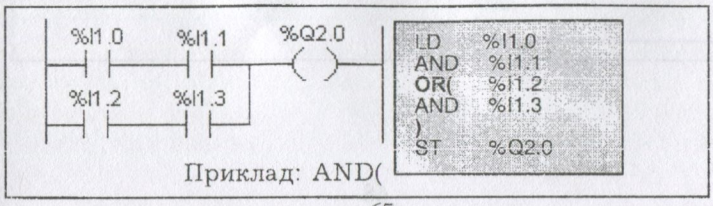
Інструкції	Еквівалентна функція
LD, LDN, LDR,LDF	
AND, ANDN, ANDR,ANDF	
OR, ORN, ORR, ORF	
AND(, OR(, 8 рівнів дужок	
N	Логічна функція
XOR, XORN, XORR, XORF	"Исключительное ИЛИ"
MPS **MRD** **MPP**	

Виконавчі базові інструкції

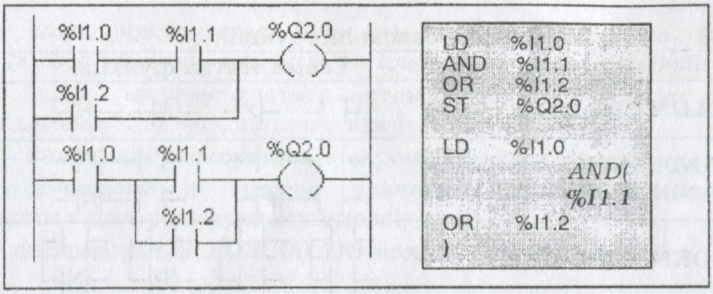
Інструкції	Еквівалентна функція
ST, STN, S, R	
JMP, JMPC, JMPCN	Програмований перехід (умовний або безумовний) до мітки, розташованої вище або нижче фрагменту.
SRn **RET, RETC, RETCN**	Програмований перехід до підпрограми. Повернення з підпрограми (безумовний або умовний)
END, ENDC, ENDCN **HALT**	Закінчення програми (безумовне або умовне). Зупинка виконання програми

У програмі одночасно з інструкціями AND (И) та OR (ИЛИ) для еквівалентної форми запису програми на мові LD можуть використовуватись круглі дужки. При цьому кількість дужок, що закриваються повинні відповідати кількості дужок, що відкриваються. Дужка, що закривається записується у програмі як окрема інструкція.

Приклад: OR(

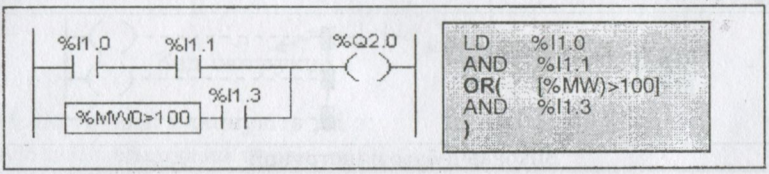


Приклад: AND(

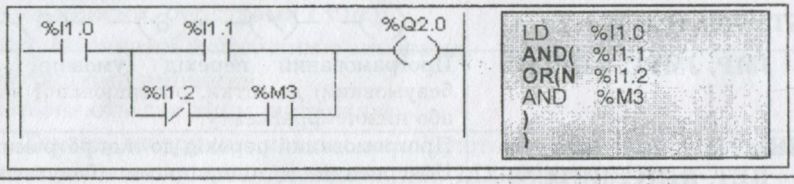


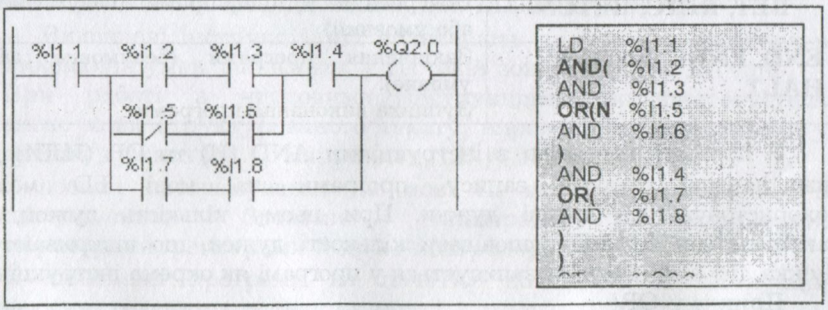
Наступні модифікатори можуть бути використані із круглими дужками:

- **N** – логічне заперечення: AND(N або OR(N ;
- **F** - обробка заднього фронту: AND(F або OR(F ;
- **R** - обробка переднього фронту: AND(R або OR(R ;
- **[** - порівняння



До восьми рівнів круглих дужок можуть вкладені одна в одну.



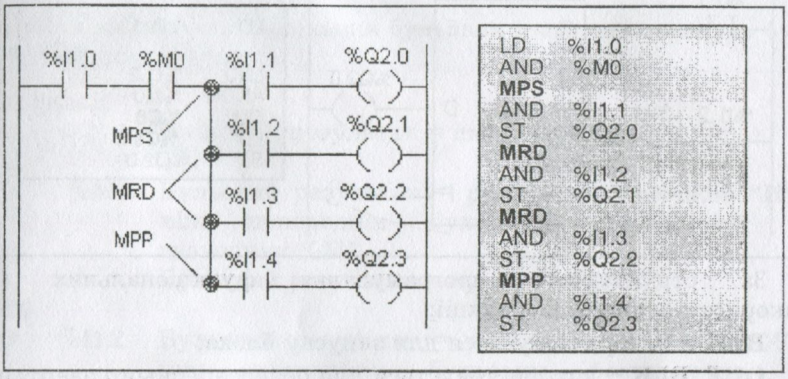


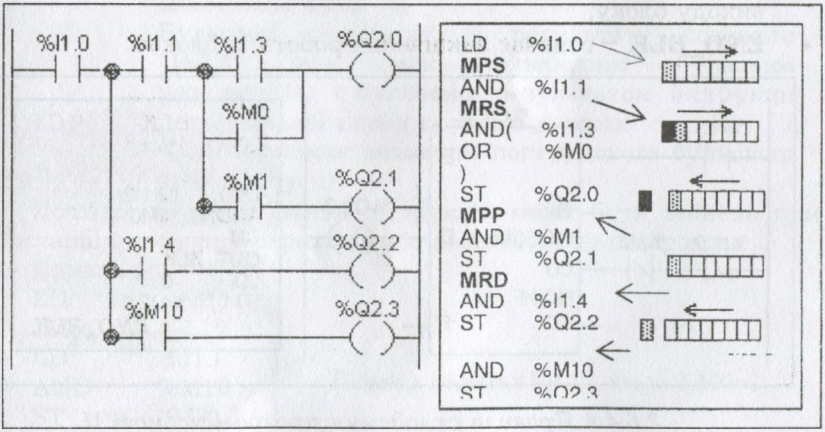
2.6.4.2 *Інструкції MPS, MRD, MPP*

Ці три типи інструкцій використовуються для програмування складних логічних виразів, які на мові LD мають досить розгалужену структуру.

Вони використовують буфер, відомий як стек, для роботи, із яким використовується три інструкції:

- **MRS** (Memory PuSh) – збереження останньої тестової інструкції у верхній частині стека й переміщення інших даних до його дна;
- **MRD** (Memory ReaD) – читання даних із верхньої частини стека;
- **MPP** (Memory PoP) – читання й поновлення вершини стека за рахунок переміщення інших даних до вершини стека



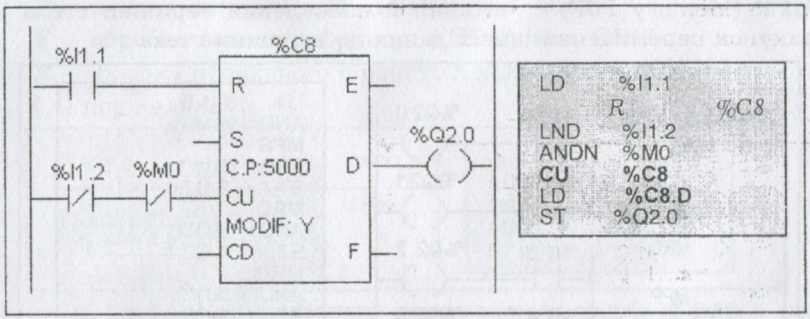


2.6.4.3 Принципи програмування функціональних блоків

Програмування функціональних блоків може виконуватись двома різними засобами:

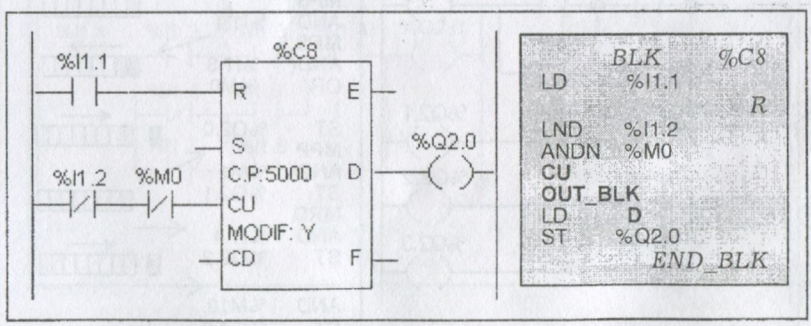
- за допомогою команд безпосереднього звертання до входів-виходів функціонального блока;
- за допомогою спеціальних інструкцій звертання до блоків BLK, OUT_BLK та END_BLK

За прямого звертання до функціонального блока у програмі використовуються інструкції, які відповідають позначенням входів-виходів відповідних блоків.



За структурного програмування функціональних блоків використо-вуються інструкції:

- **BLK** – використовується для запуску блока;
- **OUT_BLK** – використовується для безпосереднього звертання до виходу блоку;
- **END_BLK** – означає закінчення роботи із блоком.



2.6.4.4 Правила складання програм на мові IL

Інструкції у програмі Instruction List виконуються послідовно одна за одною. Перша інструкція у послідовності команд повинна починатись з інструкції LD або команди безумовного переходу JMP.

Усі інструкції за виключенням інструкцій LD і JMP використовують бульовий результат попередньої інструкції.

Приклад.

LD		Бульовий результат = стану біта %I1.1
%I1.1		
AND	**%M1**	Бульовий результат = результату логічного "И" між попереднім бульовим результатом і значенням %M1
OR		Бульовий результат = результату логічного
%M10		"ИЛИ" між попереднім бульовим результатом і значенням %M10
ST		%Q2.0 приймає значення попереднього бульового
%Q2.0		результату

Для зміни порядку виконання бульових інструкцій можуть бути використані круглі дужки.

Приклад.

LD		Бульовий результат = стану біта %I1.1
%I1.1		
AND	**%M1**	Бульовий результат = результату логічного "И" між попереднім бульовим результатом і значенням %M1
OR(		Бульовий результат = стану біта %M10
%M10		
AND	**%I1.2**	Бульовий результат = результату логічного "И" між попереднім бульовим результатом і значенням %I1.2
)		Бульовий результат = результату логічного "ИЛИ" між між попереднім бульовим результатом і бульовим результатом інструкції розташованої перед круглою дужкою
ST		%Q2.0 приймає значення попереднього бульового
%Q2.0		результату

Послідовність виконання інструкцій може бути змінена при використанні інструкцій переходів або звертання до підпрограм.

Приклад.

```
!   LD      %M0
    JMPC    %L10
!   LD      %I1.1
    AND     %M10
    ST      %Q2.0
!   %L10
    LD      %I1.3
    AND     %M20
    .....
```

Перехід до мітки %L10, якщо %M0=1

2.6.5 Мова "Графсет" (Grafcet)

2.6.5.1 Загальні положення

Найбільш відома із сучасних мов програмування для вирішення задач логіко-командного управління є мова "Графсет" і її модифікації, які широко застосовуються в різних типах програмованих логічних контролерів.

Теоретичною основою мови "Графсет" є мережі Петрі.

Назва цієї мови походить від французької "Grafcet", тобто від перших літер виразу " de Graphe de Commande des Etapes et Transitions", що перекладається як: "граф команд з етапами й переходами".

Основними перевагами цієї мови є можливість описувати паралельно-послідовні процеси в наглядній і компактній формі й автоматичне отримання по цьому опису (при наявності транслятора) управляючих програм.

Мова "Графсет" являє собою функціональні діаграми за допомогою яких у графічній формі описується виконання алгоритму управління об'єктом у вигляді набору **послідовних кроків**. Кожний крок програми включає фрагмент програми користувача, яка виконується на протязі часу в який даний крок є активним. Перехід з одного кроку на інший відбувається при виконанні відповідних **умов**, які також програмуються. Крім послідовного виконання кроків передбачається їх паралельне виконання, за умови об'єднання їх за допомогою логічних виразів "И" або "ИЛИ".

Розрізняють два типи кроків: початковий і простий. Початковий крок, або крок ініціалізації – активізується першим на початку виконання програми користувача або після холодного рестарту ПКЛ. Кількість простих кроків визначається типом контролера і складає для МПК TSX Micro – 96, а для TSX Premium – 128.

Кожна функціональна структура Графсет може мати тільки один початковий крок, але програма користувача може одночасно включати в себе декілька функціональних структур, кожна з яких має початковий крок. Це означає, що ці функціональні структури виконуються незалежно одна від одної паралельно.

Сама діаграма "Графсет" представляє собою набір графічних елементів із відповідними правилами модифікації, які визначають поведінку системи в динаміці.

На рис. 2.28 наведений приклад функціональної структури Графсет.

На початку роботи програми користувача починає виконуватись початковий крок 1.

Якщо умова, яка визначає перехід від 1 кроку 2 кроку буде виконуватися, то активізується 2 крок і дії записані в ньому будуть виконуватись. При цьому перший крок буде деактивізований.

Якщо умова переходу, яка записана після другого кроку виконується, то одночасно активізуються кроки 2 і 7, тау як в цьому місці програми використовується процедура розходження по **И** (подвійна горизонтальна лінія).

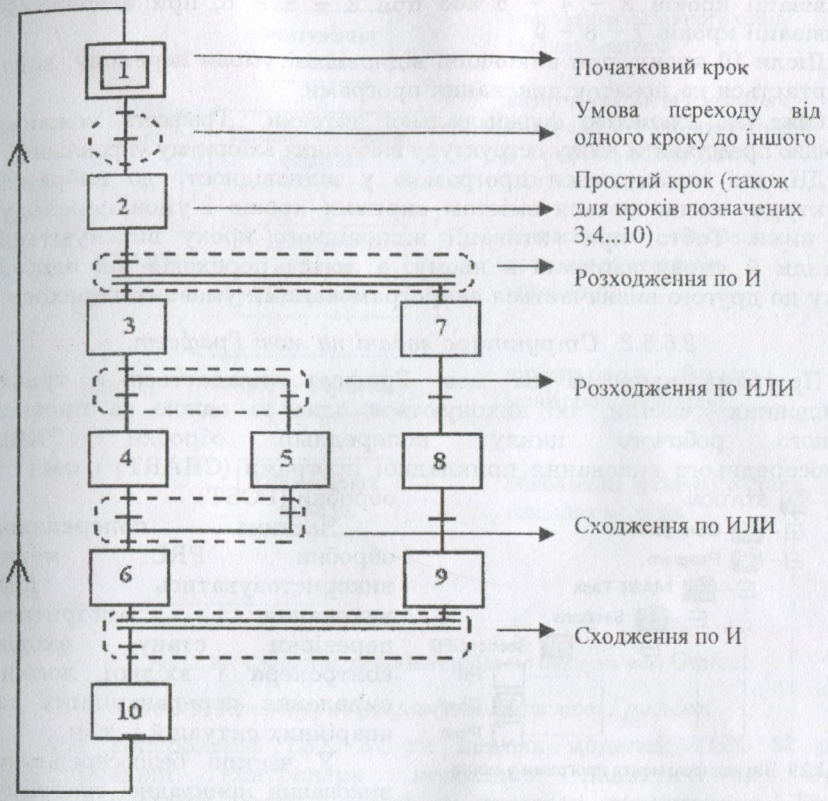


Рис. 2.28 Приклад функціональної структури

Далі виконання програми розгалужується.

Після 3 кроку, в залежності від того яка з умов переходу перед 4 або 5 кроком буде виконана першою, відповідно буде активізований 4 або 5 крок, так як у цьому місці програми використана процедура розходження по **ИЛИ** (одинарна горизонтальна лінія). Наприклад, спочатку спрацює умова перед кроком 5 і він стає активним. Тепер навіть якщо і стане виконуватись умова перед кроком 4, цей крок активним не стане. Далі програма буде чекати поки не виконається умова після 5 кроку й програма перейде до активізації 6 кроку. У цьому місці програми використана процедура сходження по ИЛИ. Тобто в цьому місці програми вона може йти або по шляху 3 – 4 – 6 або 3 – 5 – 6.

Перехід до 10 кроку може бути виконаний, тільки при одночасній активізації 6 і 9 кроку (так як у цьому місці програми використана процедура сходження по **И**) і виконання умови переходу, яке записане перед 10 кроком.

Тобто при переході до 10 кроку програма може йти по шляху активізації кроків 3 – 4 – 6 або при 3 – 5 – 6, при одночасній активізації кроків 7 – 8 – 9 .

Після 10 кроку, при виконанні відповідної умови переходу, вона повертається на початок виконання програми.

Отже, за допомогою функціональної діаграми Графсета можливо наглядно представити логіку і структуру виконання алгоритму управління.

Дії, які виконуються програмою у відповідності до вибраної структури, визначаються змістом окремих кроків і умов переходу між ними. Тобто, при активації відповідного кроку виконуються команди й умови записані в ньому, а логіка переходів від одного кроку до другого визначається запрограмованими умовами переходу.

2.6.5.2. Структура задачі на мові Графсет.

Програма написана на мові Графсет складається з трьох послідовних частин, які виконуються одна за одною на протязі кожного робочого циклу: попередньої обробки (PRL), безпосереднього виконання прикладної програми (CHART) і пост – обробки (POST).

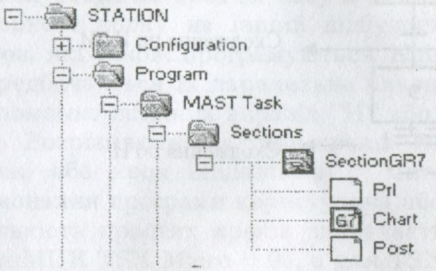


Рис.2.29 Вигляд фрагмента програми у меню

Частина попередньої обробки PRL може використовуватись для виконання алгоритмів перевірки стану входів контролера і вхідної логіки, виявлення передаварійних та аварійних ситуацій і т ін.

У частині безпосереднього виконання прикладної програми CART саме і записується функ-ціональна структура виконання алгоритму управління і програмуються окремі крокиії умови переходу між ними.

Пост-обробка POST може використовуватись для аналізу стану виходів контролера й перевірки коректності стану вихідних сигналів. Наприклад, може бути реалізований алгоритм визначення неприпустимих комбінацій вихідних сигналів, яке може призвести до виникнення аварійної ситуації. Іноді доцільно в робочій програмі працювати з внутрішніми змінними, а у частині POST робити переприсвоювання їх значень вихідним змінним.

Кожна з цих частин програмується окремо. При цьому для програмування можна використовувати різні мови програмування: Ladder Diagram, Structured Text, Instruction List. Частини попередньої і пост-обробки не обов'язкові.

Протягом виконання кожного робочого циклу виконується (рис. 2.30):

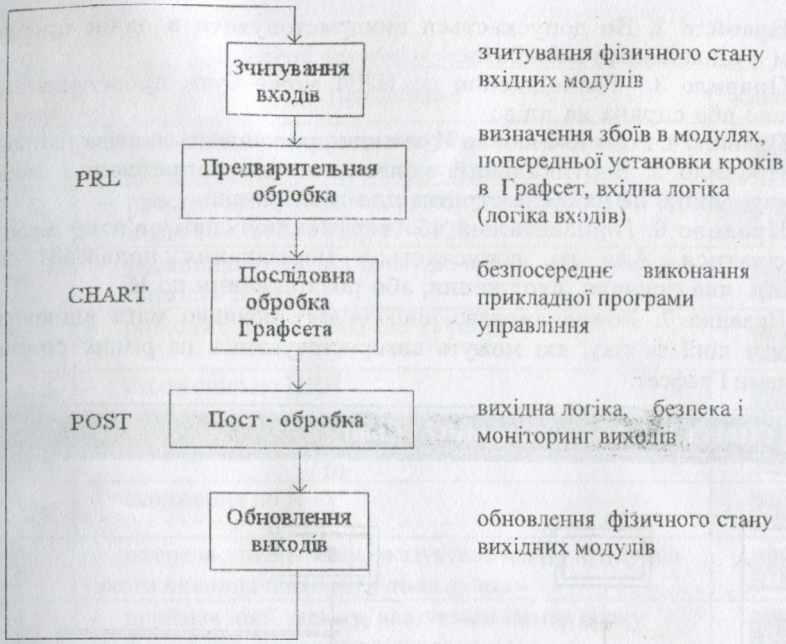


Рис. 2.30 Структура виконання робочого циклу на мові Grafcet

2.6.5.3. Процедура програмування на мові Графсет

Для контролерів TSX 37 та деяких моделей TSX 57 при конфігуруванні контролера необхідно вказати чи буде використовуватись мова Grafset у програмі користувача. Треба пам'ятати, що змінити це замовлення в процесі введення програми буде неможливо.

Процедура програмування на мові Grafset виконується в спеціальному графічному редакторі. Він створює 8 сторінок розбитих на 14 рядків і 11 колонок.

Рядки, розбиті на комірки призначенні для розміщення в них графічних елементів кроків, а інші стрічки призначені для розміщення в них зображень переходів.

Для створення функціональної діаграми Grafset використовуються графічні елементи, зображення яких нанесені також на кнопки розташовані у нижній частині робочого вікна редактора Grafset і за допомогою яких створюється функціональна структура.

У побудові функціональної діаграми Графсета необхідно виконувати наступні правила:

Правило 1. Кожний крок повинен мати номер від 0 до 128, указаний у центрі зображення.

Правило 2. Не допускається використовувати в одній програмі кроки з однаковими номерами.

Правило 3. Розходження по ИЛИ може бути проведено зліва направо або справа на вліво.

Правило 4. Розходження по И завжди проставляється зліва направо.

Правило 5. Вертикальний зв'язок завжди направлений зверху до низу, якщо не вказана стрілка для пояснення.

Правило 6. Горизонтальні або вертикальні лінії зв'язку можуть пересікатися. Але не допускається пересікання подвійної лінії зв'язку, яка показує сходження, або розходження по И.

Правило 7. Кожне джерело лінії зв'язку повинно мати відповідний приймач лінії зв'язку, які можуть використовуватись на різних сторінках програми Графсет.

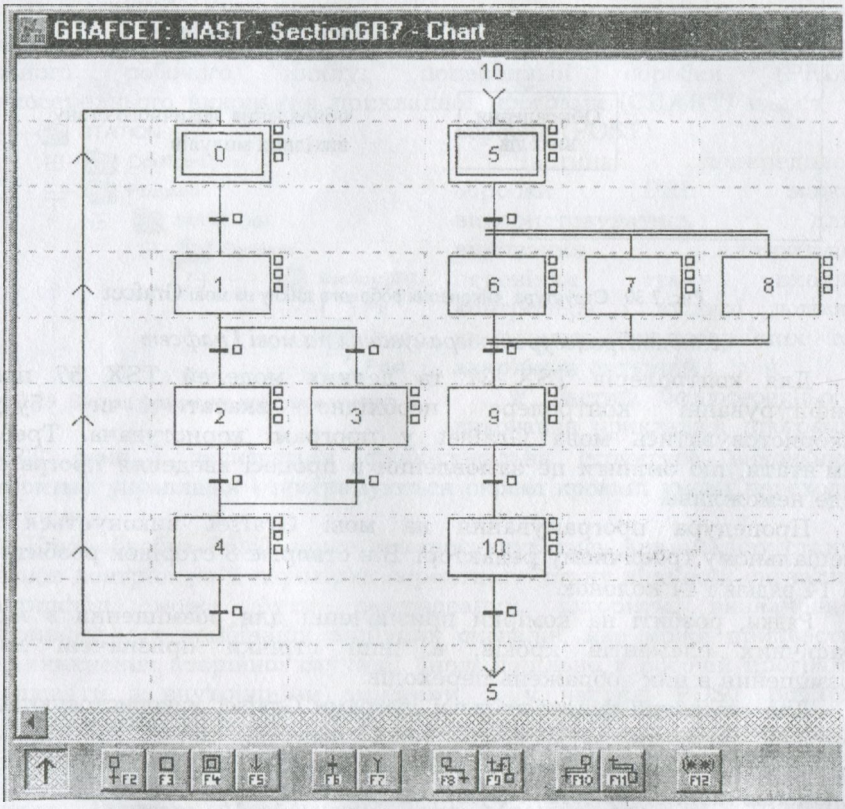


Рис.2.31 Загальний вигляд вікна графічного редактора мови Grafcet

Таблиця 2.3

Графічні елементи мови Grafset

Графічний елемент	Призначення	Кнопка
	- початковий крок, розташовується на початку гілки функціональної структури;	
	- простий крок, кількість кроків для TSX 3710 – 96, а для TSX 3722 і TSX 57 – 128;	
	- умовний перехід від кроку до кроку, максимальна кількість 1024	
	- розходження по ИЛИ (до 11 розгалужень)	
	- сходження по ИЛИ	
	- розходження по И (до 11 розгалужень)	
	- сходження по И	
	- джерело лінії зв'язку, вказується номер кроку від якого повинна підходити лінія зв'язку	
	- приймач лінії зв'язку, вказується номер кроку куди повинна підходити лінія зв'язку	

Поряд з функціональною структурою, на сторінках Графсета можуть використовуватись коментарі, які можуть складатись з 64 символів, розташованих у дві стрічки. Коментар зберігається у пам'яті МПК, тому використовувати їх треба обережно.

Для розташування коментарю на сторінці графічного редактора  використовується кнопка .

Після створення функціональної структури переходять до програмування окремих кроків і умов переходу між ними.

Особливістю програмування кроків є те, що для кожного з них можна запрограмувати три типи дій:

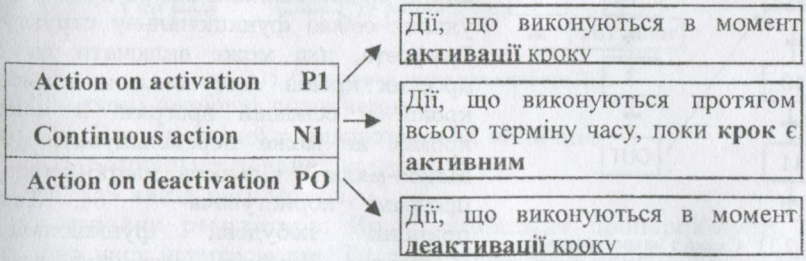
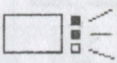


Рис. 2.32. Типи дій кроку

Необов'язково програмувати всі три типи дій. Їх використання залежить потреб алгоритму, який реалізується.

Для програмування дій кроків можна, за бажанням розробника ПрК, використовувати різні мови програмування: Ladder Diagram, Structured Text, Instruction List.

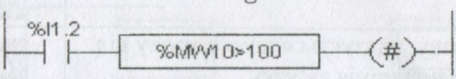
Для візуального спостереження за тим, які типи дій кроків запрограмовані, біля зображення кроку знаходяться 3 прямокутника.



Вони відповідають трьом типам дій : P1, N1, P0. Якщо якась із дій програмуються, то відповідний прямокутник замальовується.

Для програмування умов переходу також можна використовувати будь-яку мову програмування. Особливістю програмування умов є те, що вини вміщують тільки умовну частину. При програмуванні умови на мові Ladder Diagram це проявляється у тому, що замість графічного зображення виконавчих елементів у програмі використовується зображення –(#)--. Наприклад:

Ladder Diagram



Instruction List

```
LD %I1.2
AND [%MW10 < 10]
```

Structured Text

```
! %I1.2 AND %MW10 <10
```

З окремими кроками асоційовані відповідні об'єкти мови: %Xi та %Xi.T.

Біт %Xi = 1 коли відповідний крок є активним, а в слові %Xi.T виконується облік тривалості активності кроку в одиницях виміру 100 мс. Наприклад, якщо %X2.T дорівнює 150, то тривалість активності другого кроку складає 15 с.

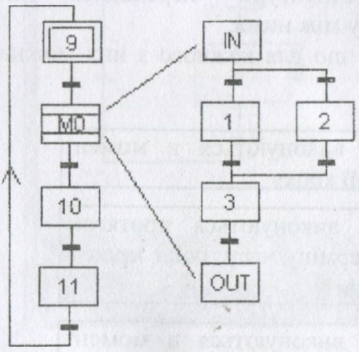


Рис.2.33 Схема з макро-кроками

У останніх версіях PL7 Pro для конт-ролерів TSX 57 (Premium) у функціональну структуру Графсета поряд з простими кро-ками введені макро-кроки. Кожний макро-крок також уявляє собою функціональну структуру Графсета, яка може включати до 250 простих кроків. Але загальна кількість кроків у основній програмі й макро-кроках не може перебільшувати 1024. Макси-мальна кількість макро-кроків у програмі користувача – 64. Такий принцип побудови функціональної структури Графсет дозво-ляє ввести декілька різних за укрупненням алгоритмів керування..

Тобто спочатку може бути складена укрупнена функціональна структура всередені якої можуть використовуватись макро-кроки в яких будуть запрогра-мовані фрагменти алгоритмів, які є складовими для загального. Кожна функціональна структура макро-кроку починається із кроку позначеного "IN" і закінчується кроком "OUT".

2.6.6. ПІД-регулювання

До складу бібліотеки алгоритмів програмного забезпечення PL7 Pro входять алгоритми ПІД-регулювання. Структурна схема ПІД-регулятора показана на рис. 2.34.

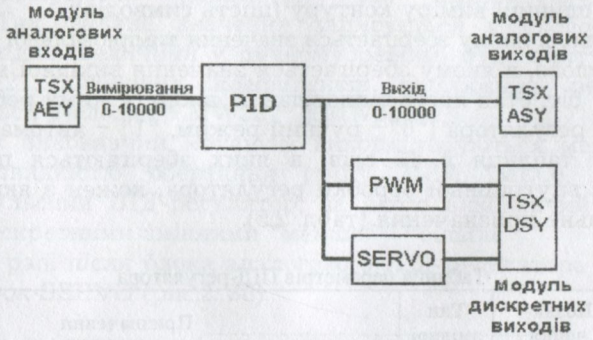


Рис. 2.34. Структурна схема ПІД-регулювання

До структурної схеми входять три функціональних блоки:

- блок ПІД-регулятора (PID);
- блок широтно-імпульсного перетворення (PWM);
- блок керування кроковими двигунами (SERVO).

Блок PID призначений для реалізації класичного аналогового ПІД-регулятора (рис.2.35).

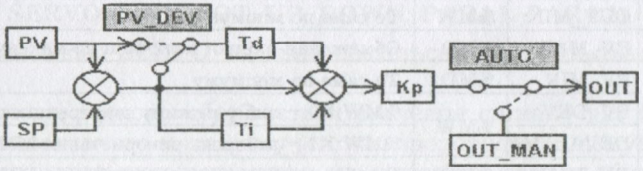


Рис. 2.35. Структурна схема аналогового ПІД-

На схемі показані позначення:

PV (Process value) – вимірювальна величина;

SP (Setpoint) – задане значення;

OUT – вихід регулятора;

настройки регулятора: Kp – коефіцієнт пропорційності, Ti – постійна часу інтегрування, Td – постійна часу диференціювання.

В регуляторі передбачено два перемикачі:

AUTO – для переходу з автоматичного режиму на ручний і навпаки;

PV_DEV – для вибору режиму диференціювання (диференціювання значення вхідної величини або розузгодження).

У процесі програмування ПІД-регулятора у функціональному блоці або у бібліотечному алгоритмі необхідно вказати слова й біти, що відповідають змінним, коефіцієнтам і перемикачам. Ці біти й слова необхідно розташувати у заданій послідовності. Прийнято такий синтаксис для програмування ПІД-регулятора:

PID ('TAG', 'UNIT', PV, OUT, AUTO, PARA),

де: TAG – назва PID контуру (вісім символів);

UNIT – одиниці виміру контуру (шість символів);

PV – слово, в якому зберігається значення вимірювальної величини;

OUT – слово, в якому зберігається значення вихідної величини;

AUTO – біт, стан якого відповідає за вибір режиму роботи ПІД-регулятора ("0"– ручний режим, "1" – автоматичний);

PARA – таблиця з 43 слів, в яких зберігаються параметри і результати внутрішньої обробки регулятора, кожен з яких має своє функціональне призначення (табл. 2.5).

Таблиця 2.5

Таблиця параметрів ПІД-регулятора

Номер слова у таблиці	Позначення	Тип змінної	Призначення
1	SP	%MW	Значення завдання (0-10000)
2	OUT_MAN	%MW	Значення вихідної величини у ручному режимі
3	KP	%MW	Коефіцієнт пропорційності (значення x 100)
4	TI	%MW	Постійна часу інтегрування (значення/10, с)
5	TD	%MW	Постійна часу дифенціювання (значення/10, с)
6	TS	%MW	Дискретність (значення/100, с)
7	OUT_MAX	%MW	Обмеження виходу по максимуму
8	OUT_MIN	%MW	Те саме по мінімуму
9–10	PV_MAX	%MD	Обмеження вхідної величини по максимуму
11–12	PV_MIN	%MD	Те саме по мінімуму
13	PV_DEV	%MW	%MW:X0 – вибір режиму диференціювання
	DEVAL_MM		%MW:X7 – заборона використання MMI
14-15	PV_MMI	%MD	Відображення вхідної величини на MMI (значення x100)
16-17	SP_MMI	%MD	Відображення заданої величини на MMI (значення x100)
18-43		%MW	Внутрішні змінні для обробки ПІД-регулятора

Назва регулятора, одиниці виміру і змінні, які зарезервовані для зв'язку із MMI (людино-машинний інтерфейс) використовуються тільки у тому разі, якщо до контролера під'єднана операторська панель CCX-17, яка автоматично генерує робочі вікна для контролю за роботою окремих

контурів регулювання. З цієї панелі можна змінювати задане значення контура регулювання, а також переходити на ручний режим роботи контура і з панелі, вручну керувати виконавчим механізмом. Якщо панель CCX-17 не використовується, ці значення можна не задавати:

PID (“, “, %IW3.0, %QW4.1, %M5, %MW20:43).

У цьому прикладі: датчик під’єднаний до аналогового входу контролера TSX Micro %IW3.0, а аналоговий виконавчий механізм – до аналогового виходу %QW4.1. Режим роботи контролера вибирається значенням біта %M5; у словах, що починаються з %MW20, записуються параметри, які використовуються у регуляторі. Так, задане значення записується у слово %MW20, значення вихідної величини у ручному режимі буде зчитуватись зі слова %MW21, значення коефіцієнтів настроєння регуляторів зберігатимуться у словах %MW22 – %MW24 і т.д.

Якщо як виконавчий механізм використовуються механізми з постійною швидкістю обертання (типу МЕК, МЕО), необхідно вибрати імпульсний ПІД-регулятор, за якого виконавчий механізм керується дискретними змінними “менше” - ”більше”.

У цьому разі після блока аналогового ПІД-регулятора необхідно під’єднати блок SERVO (рис.2. 36).

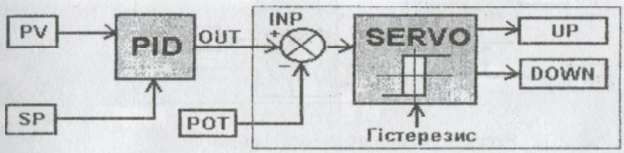


Рис. 2.36. Імпульсний ПІД-регулятор

Цей блок має такий синтаксис:

SERVO (INP, POT, UP, DOWN, PARA1, PARA2),

де: INP – вхід блока SERVO, на який подається вихід ПІД-аналогового регулятора;

POT – слово, в якому зберігається значення сигналу від датчика положення виконавчого механізму (сигнал зворотного зв‘язку);

UP – вихід імпульсного регулятора, що відповідає сигналу “більше”;

DOWN – вихід імпульсного регулятора, що відповідає сигналу “менше”;

PARA1 – таблиця із 43 слів (таблиця параметрів аналогового ПІД-регулятора;

PARA2 – таблиця параметрів блока SERVO із дев’яти слів (табл. 2.6).

Таблиця 2.6

Таблиця параметрів блока SERVO

Номер слова у таблиці	Позначення	Тип змінної	Призначення
1	T_MOTOR	%MW	Тривалість імпульсу (значення/100 с)
2	T_MINI	%MW	Мінімальна тривалість імпульсу (значення/100 с)
3	HYST	%MW	Значення гістерезису
4–9		%MW	Внутрішні змінні для обробки блока SERVO

Наприклад, якщо для наведеного прикладу необхідно керувати виконавчим механізмом постійної швидкості обертання, сигнал "більше" для якого знімається з дискретного виходу %Q2.0, а сигнал "менше" – з %Q2.1, програма буде мати вигляд:

PID (‘‘, ‘‘, %IW3.0, %QW4.1, %M5, %MW20:43)
SERVO (%QW4.1, %IW3.1, %Q2.0, %Q2.1, %MW20:43, %MW100:9).

Якщо у вигляді виконавчого механізму використовується крокований двигун, то після блока ПІД-регулятора використовується блок широтно-імпульсного модулятора PWM, структура якого

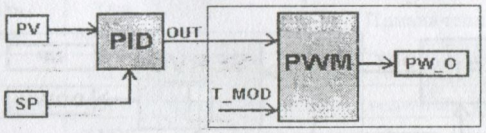


Рис. 2.37. Регулятор з широтно-імпульсною

показана на рис.2.37.

Синтаксис цього блоку досить простий:

PWM (INP, PW_O, PARA)

де: INP – вхід блоку PWM, на який подається вихід ПІД-аналогового регулятора;

PW_O – дискретний модульований вихід;

PARA – таблиця параметрів блока PWM із п'яти слів.

Першим параметром цієї таблиці є слово, в якому зберігається значення періоду модуляції T_MOD, який повинен бути меншим або дорівнювати періоду дискретизації TS ПІД-регулятора. T_MOD може змінюватись у межах від 0 до 32767 і вимірюється у 0,01 секунди (рис. 2.41).

Тривалість імпульсу залежить від значення вхідної величини INP і періоду модуляції T_MOD:

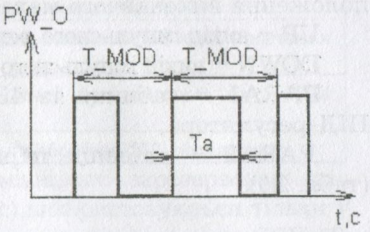


Рис. 2.38. Широтно-імпульсна модуляція

$Ta = \frac{INP \bullet T_MOD}{10000}$. Фрагменти програми користувача для реалізації алгоритмів ПІД-регуляторів на мові Structured Text мають синтаксис, наведений вище. На мові Ladder Diagram ці вирази записуються у функціональний блок, приклад якого наведений на рис. 2.39.

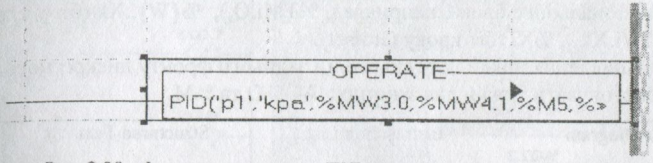


Рис. 2.39. Фрагмент програми ПІД-регулятора на мові Ladder

В останніх версіях процесорних модулів TSX Premium програмне налагоджування контурів керування виконується шляхом введення параметрів (технологія Plug and Play) під час конфігурування процесора ПЛК. Користувач заповнює попередньо сконфігуровані схеми контурів керування, вказуючи зв'язок їх з вхідними і вихідними величинами і вказуючи режими їх роботи.

Пропонується кілька найбільш поширених типових схем керування: 2- і 3-позиційного регулятора, локального аналогового і імпульсного ПІД-регулятора, каскадного та програмного керування (рис.2.40, рис.2.41, рис.2.42) і т.ін.

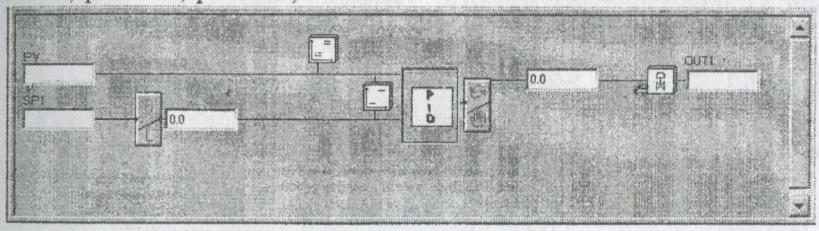


Рис. 2.40. Приклад конфігураційного вікна звичайного ПІД-регулятора

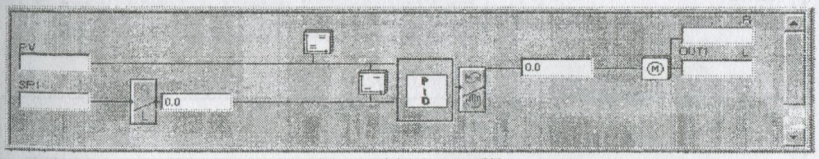


Рис. 2.41. Приклад конфігураційного вікна імпульсного ПІД-регулятора

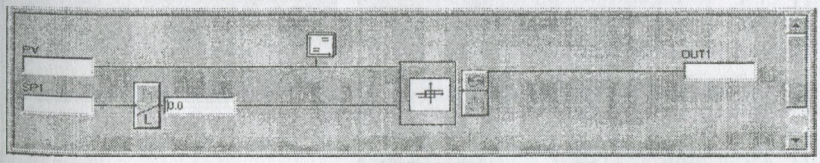


Рис. 2.42. Приклад конфігураційного вікна 2-позиційного регулятора

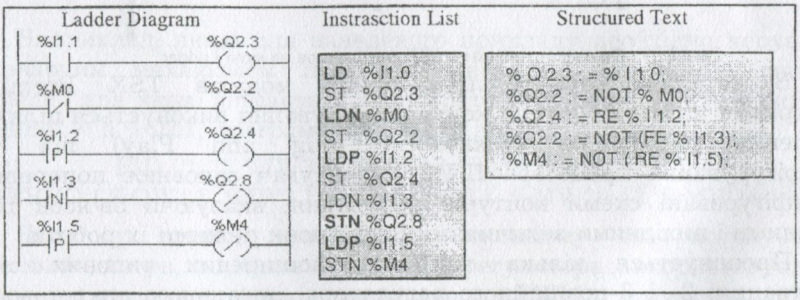
2. 7 Приклади виконання фрагментів прикладної програми

2.7.1 Програмування логічних інструкцій з дискретними змінними

2.7.1.1 Інструкції опитування

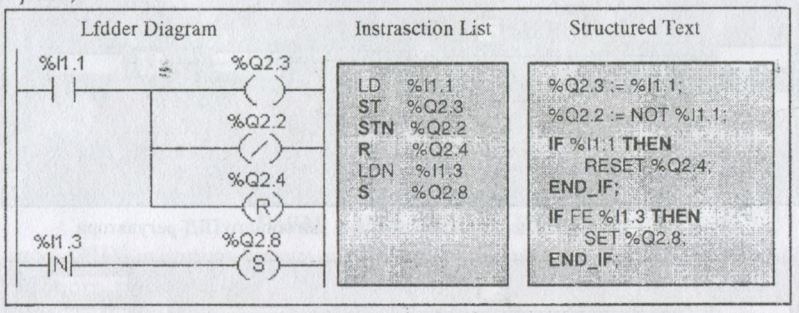
Передбачена можливість опитування дискретних змінних: %I, %Q, %M, %S, %BLK (біт функціонального блока, наприклад, %TMi.Q,), %{W}..Xk (біт у слові, наприклад, %MWi.Xk,), %Xi (біт кроку Grafset).

Інструкції виявлення появи переднього чи заднього фронту дискретного сигналу можливо використовувати тільки для змінних: %I, %Q та %M.

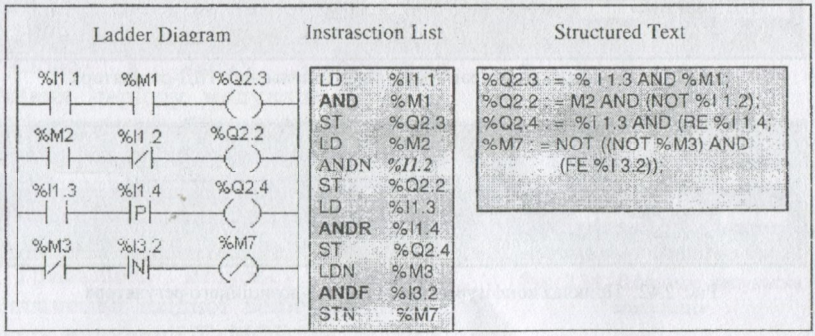


2.7.1.2 Інструкції присвоювання

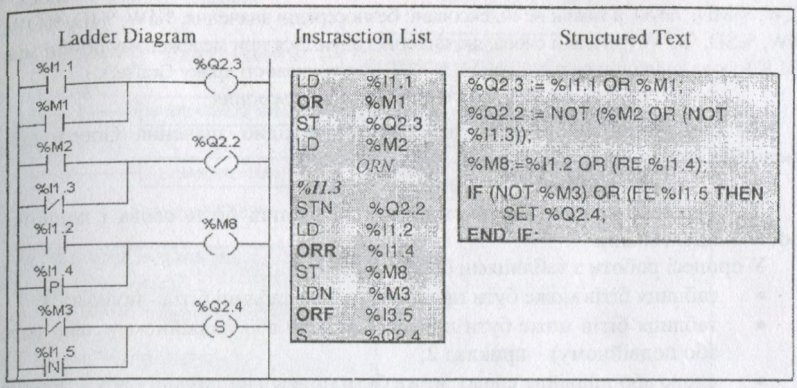
Інструкція присвоювання використовується для змінних: %I, %Q, %M, %S, %{W}..Xk, %Xi.



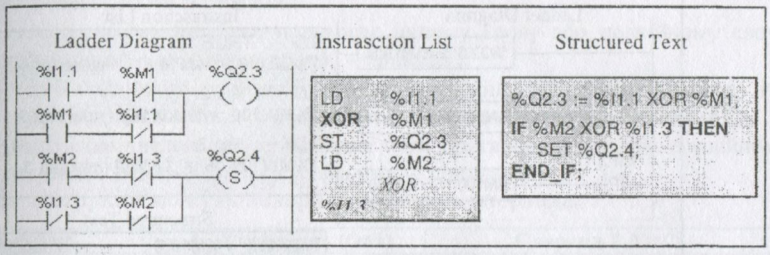
2.7.1.3 Логічне "И"



2.7.1.4 Логічне "ИЛИ"



2.7.1.5 Інструкція "Исключающее ИЛИ"

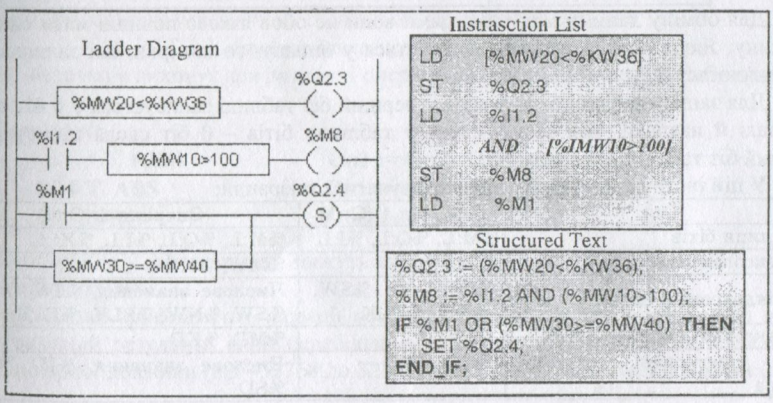


2.7.2 Інструкції з числами

Інструкції з числами використовуються для роботи з таблицями бітів, словами і подвійними словами.

2.7.2.1. Інструкції порівняння

Синтаксис: **Оп1** (>, >=, <, =<, = <>) **Оп2,**
де, Оп1 і Оп2 – операнд 1 і операнд2



Як операнди можуть використовуватись індексовані прості й подвійні слова: %MW, %KW, %MD, %KD, а також не індексовані: безпосередні значення, %IW, %ID, %QW, %QD, %SW, %SD, %NW (загальні слова, які використовуються при мережевому обміні між ПЛК), %BLK (слова функціональних блоків), %Xi.T (час тривалості кроку Grafset).

2.7.2.2 Інструкції присвоювання

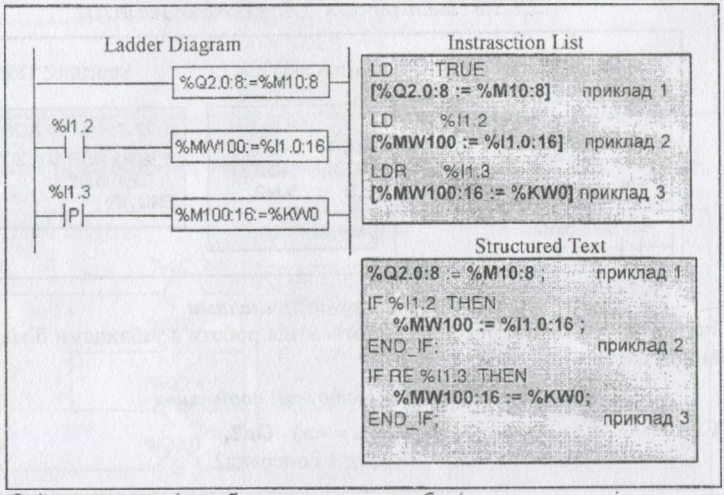
Ця інструкція використовується коли необхідно значення Операнда2 (Оп2) присвоїти Операнду1 (Оп1).

Синтаксис: **Оп2 : = Оп1**

Як операнди можуть використовуватись: таблиці бітів, слова і подвійні слова (прості й індексовані).

У процесі роботи з таблицями бітів:

- таблиця бітів може бути присвоєна іншій таблиці бітів - приклад 1;
- таблиця бітів може бути присвоєна слову або подвійному слову (простому або подвійному) – приклад 2;
- слово або подвійне слово може бути присвоєне таблиці бітів – приклад 3.

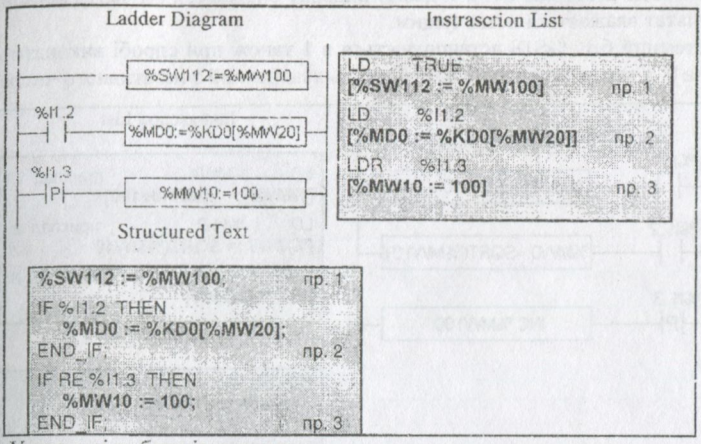


Для обміну даними між таблицями вони не обов'язково повинні мати однакову довжину. Якщо менша таблиця записується у більшу, то комірки, які залишаються, заповнюються 0.

Для запису таблиці бітів у слово, перший біт таблиці записується у 0 біт слова і так далі й навпаки, при записі слова у таблицю бітів – 0 біт слова записується у перший біт таблиці, і так далі.

У цій інструкції можуть використовуватись операнди:

Тип	Операнд 1 (Оп1)	Операнд 2 (Оп2)
Таблиця бітів	%M:L, %Q:L, %I:L	%M:L, %Q:L, %I:L, %Xi:L
Індексовані слова	%MW	%MW, %KW
Не індексовані слова	%QW, %SW, %NW, %BLK	Числове значення, %IW, %QW, %SW, %NW, %BLK, %Xi.T
Індексовані подвійні слова	%MD	%MD, %KD
Не індексовані подвійні слова	%QD, %SD	Числове значення, %ID, %QD, %SD



У процесі роботи із словами :

- значення слова може бути присвоєне іншому слову або подвійному слову (індексованому і не індексованому) – приклад 1;
- значення подвійного слова може бути присвоєне іншому подвійному слову або слову одинарної довжини (індексованому і не індексованому) – приклад 2;
- безпосереднє числове значення може бути присвоєне слову або подвійному слову (індексованому і не індексованому) – приклад 3.

У цій інструкції можуть використовуватись операнди:

Тип	Операнд 1 (Оп1)	Операнд 2 (Оп2)
Індексовані слова	%MW	%MW, %KW
Не індексовані слова	%QW, %SW, %NW, %BLK	Числове значення, %IW, %QW, %SW, %NW, %BLK, %Xi.T
Індексовані подвійні слова	%MD	%MD, %KD
Не індексовані подвійні слова	%QD, %SD	Числове значення, %ID, %QD, %SD

Записуючт значення подвійного слова в слово одинарної довжини, треба пам'ятати, що інформація, яка записана у подвійному слові може бути частково втрачена.

2.7.2.3 Арифметичні інструкції з цілими числами

Ця інструкція виконує арифметичні операції між двома операндами або з одним операндом.

Синтаксис:

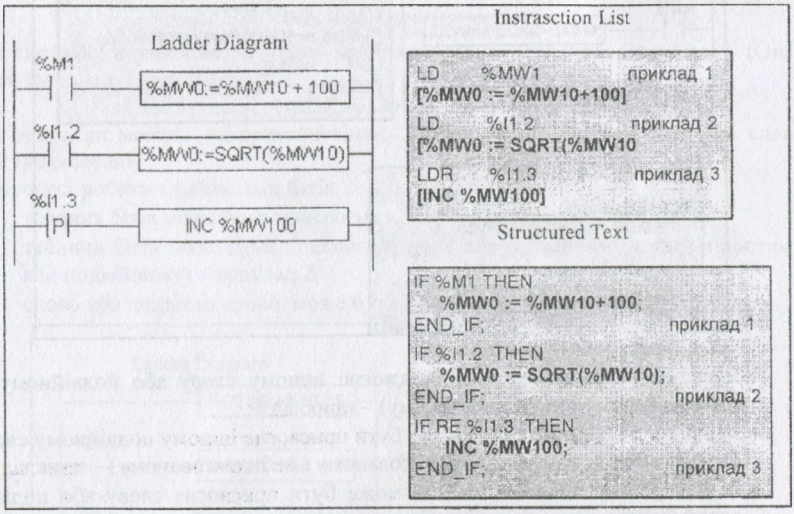
- **+, -, *, /, REM** **Оп1 := Оп2 Операція Оп3**
- **SQRT, ABS** **Оп1 := Операція (Оп2)**
- **INC, DEC** **Операція Оп1**

Як операнди у цій інструкції можуть використовуватись ті ж самі змінні, що і у попередній інструкції.

Виконуючи арифметичні операції необхідно пам'ятати, що в результаті може бути отриманий результат, який перебільшує значення, яке може бути записане у слово одинарної довжини (від –32768 до 32767) або у слово подвійної довжини (від –

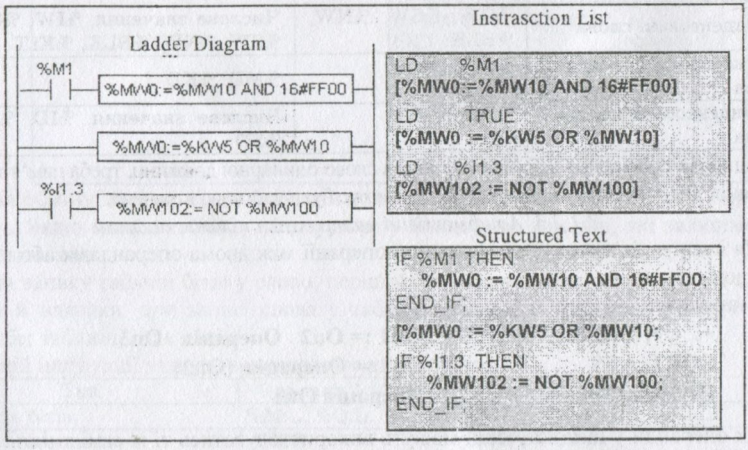
2 147 483 648 до 2 147 483 647). У цьому випадку системний біт %S18 встановлюється у 1 і результат вважається незначущим.

Системний біт %S18 встановлюється в 1 також при спробі виконати операцію ділення на 0, а також при спробі взяти корінь квадратний з негативного числа.



2.7.2.4 Логічні інструкції

Ці інструкції використовуються для виконання побітових логічних операцій (AND, OR, XOR, NOT) між двома операндами або над одним оператором.



2.7.2.5 Правила виконання інструкцій з числами

Кількість числових операндів та арифметичних або логічних операцій, які входять до числового виразу – необмежена.

Наприклад,

%MW25*3 – SQRT(%MW10) + %KW8*(%MW15 + %MW18) AND 16#FF

Інструкція виходу з підпрограми повинна обов'язково розташовуватись в кінці кожної підпрограми, але може бути використана також і в середині підпрограми для її дострокового закінчення.

На мові Instrasction List використовуються інструкції:

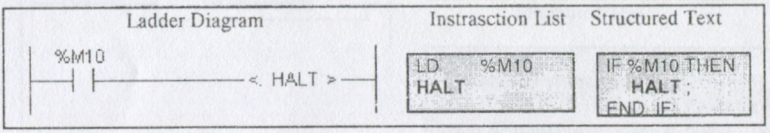
RETC – вихід із підпрограми, коли результат попередньої тестової інструкції 1.

RETCN - вихід із підпрограми, коли результат попередньої тестової інструкції 0.

RET – безумовний вихід із підпрограми.

2.7.3.2 Програмована зупинка виконання програми

Виконання програми користувача може бути зупинена інструкцією HALT. Це призводить до заморожування (зберіганні їх у стані, який передував зупинці) усіх змінних у цій програмі. Програмована зупинка може бути знята шляхом перезапуску й ініціалізації програми.



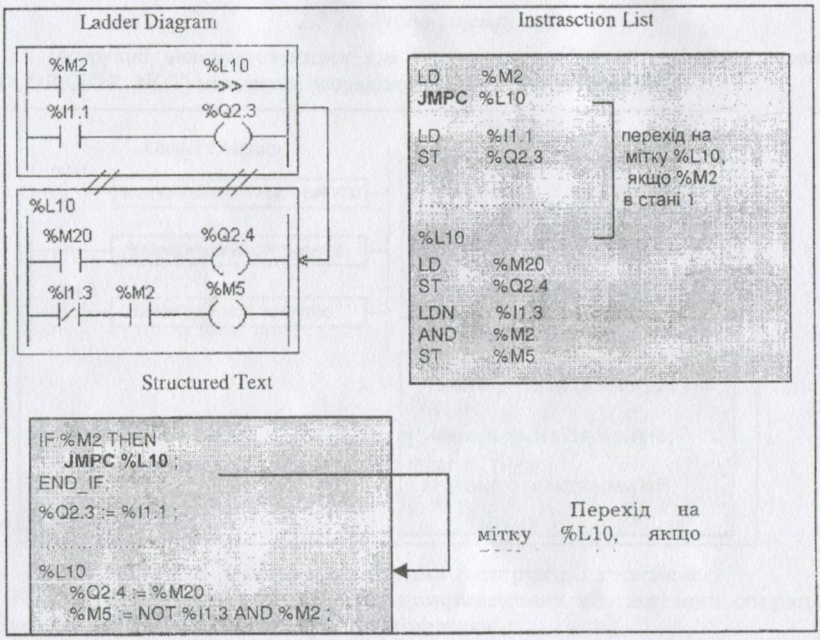
2.7.3.3 Програмований перехід

Використовується для програмованого переходу до мітки %Li.

JMP – безумовний перехід;

JMPC – перехід за умови, що результат попередньої тестової інструкції 1.

JMPCN - перехід за умови, що результат попередньої тестової інструкції 0.



Програмований перехід виконуються всередині або основної задачі (MAST), або підпрограми і т.ін.

Програмований перехід може відбуватись до фрагмента, який розташований **до** або **після** переходу у програмі користувача. Якщо перехід запрограмований до фрагмента, який знаходиться **попереду** інструкції переходу, треба бути дуже уважним, бо це може призвести до перебільшення дозволеної тривалості часу робочого циклу.

2.7.4. Програмування функціональних блоків

2.7.4.1. Програмування таймерів

Як було вказано у розділі 2.6.2.2, у редакторі змінних для кожного таймера повинні бути сконфігуровані параметри:

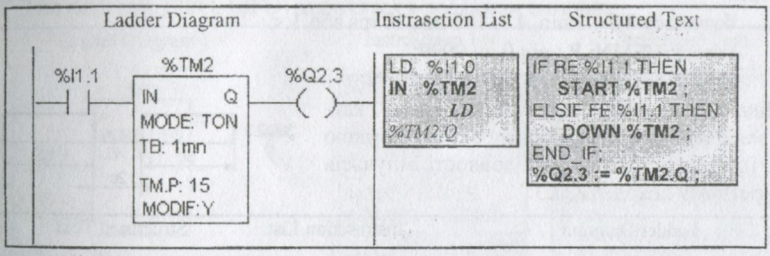
- режим MODE: TON, TOF або TP;
- база часу TB: 1min, 1s, 100ms або 10 ms;
- уставка %TMi.P: від 0 до 9999;
- режим модифікації MODIF: Y або N.

Після цього сконфігурований таймер вставляється у програму користувача.

Запуск таймера відбувається по передньому фронту сигналу на вході IN для таймерів TON та TP і по задньому фронту для таймера TOF.

Зупинка таймера відбувається по задньому фронту сигналу на вході IN для таймерів TON та TP і по передньому фронту для таймера TOF.

Це добре видно на мові Structured Text при записі умов спрацьовування інструкцій START (запуск) і DOWN (зупинка).



2.7.4.2 Програмування лічильників

Під час конфігурування лічильника у редакторі змінних виставляються параметри:

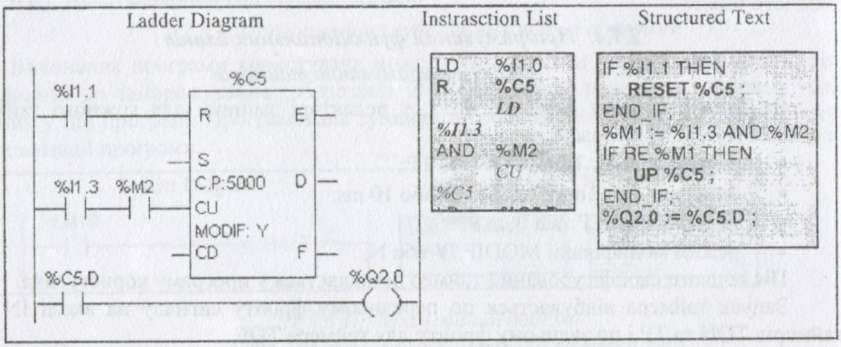
- уставка %Ci.P: від 0 до 9999;
- режим модифікації MODIF: Y або N.

Після цього сконфігурований лічильник вставляється у програму користувача.

На мові Structured Text при програмуванні реверсивного лічильника використовуються 4 інструкції:

- RESET %Ci (R): Скидання поточного значення %Ci.V у 0;
- PRESET %Ci (S): Запис значення уставки %Ci.P у поточне значення %Ci.V.
- UP %Ci (CU): Інкрементування (збільшення на 1) поточного значення %Ci.V.
- DOWN %Ci (CD): Дікрементування (зменшення на 1) поточного значення %Ci.V.

Лічильні входи CU і CD, яким відповідають інструкції UP і DOWN, спрацьовують по передньому фронту сигналів, тому при їх використанні необхідно підготувати опитування саме переднього фронту сигналів для цих інструкцій. У наведеному вище прикладі, для того щоб можна було виділити передній фронт на вході CU, вводиться проміжна змінна %M1, в якій записана умова спрацьовування цього входу і в інструкції UP %Ci опитується передній фронт цієї змінної.



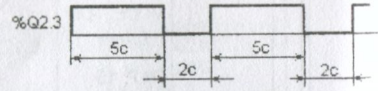
2.7.4.3. Програмування одновібраторів

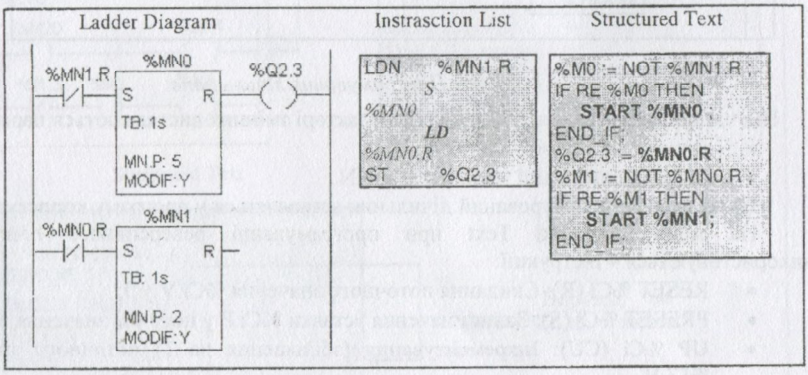
Одновібратори призначення для створення імпульсів підвищеної точності.

Під час конфігурування одновібратора в редакторі змінних повинні бути задані параметри:

- базу часу TB : 1min, 1 s, 100 ms, 10 ms або 1 s;
- уставку %MNi.P : від 0 до 9999;
- режим модифікації MODIF: Y або N.

Нижче наведений приклад, в якому каскадно включені два одновібратора з витримкою часу 5с і 2с для реалізації послідовності імпульсів на дискретному виході %Q2.3:





На мові Structured Text для запуску одновібратора використовується інструкція START %MNi. Ця інструкція виконується по передньому фронту сигналу на вході S одновібратора. Для цього в прикладі введені дві проміжні змінні %M0 і %M1.

На початку виконання наведеного фрагмента програми на вхід S одновібратора %MN0 подається 1 і цей одновібратор запускається. При цьому спрацьовує його вихід %MN0.R і включається %Q2.3. При спрацьовуванні виходу %MN0.R, на вхід одновібратора %MN1 подається 0. Після витримки імпульсу 5 с одновібратор %MN0 відключається, його вихід %MN0.R стає дорівнювати 0 і вихід %Q2.3 відключається. При відключенні виходу %MN0.R, на вході S одновібратора %MN1 з'являється 1 і він запускається. Одночасно його вихід %MN1.R розриває ланцюг на вході S одновібратора %MN0. Після витримки часу 2 с, одновібратор %MN0 відключається, а через його вихід запускається одновібратор %MN1 і так далі. При цьому вихід %Q2.3 включається на 5 с і відключається на 2 с.

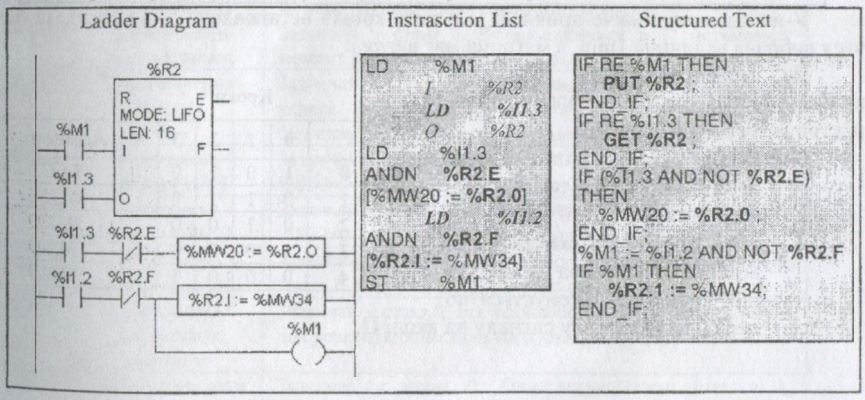
2.7.4.4. Програмування регістрів

Регістри використовуються як блоки пам'яті в яких можна зберігати до 255 16-бітових слів двома способами:

- Черга (першим прийшов – першим вийшов), відомий як регістр FIFO (First In – First Out)
- Стек (останнім прийшов – першим вийшов), відомий як регістр LIFO (Last In - First Out)

Під час конфігурування контролера необхідно вказати кількість регістрів, які будуть використовуватись у програмі користувача (від 1 до 4 для TSX 37 і від 1 до 255 для TSX 57), а також їх довжину, тобто кількість слів, яку можна зберігати у регістрі (від 1 до 255).

Тип регістру (FIFO, LIFO) вказується в редакторі змінних.



На мові Structured Text при програмуванні реверсивного лічильника використовуються 3 інструкції:

- RESET %Ri (R) - очищення регістра, тобто його ініціалізація;
- PUT %Ri (I) – запис значення слова, яке записане у вхідне слово %Ri.I в регістр:
- GET %Ri (O) – зчитування з регістра значення й запис його у вихідне слово %Ri.O.

Команди PUT %Ri та GET %Ri спрацьовують по передньому фронту на відповідних входах I та O.

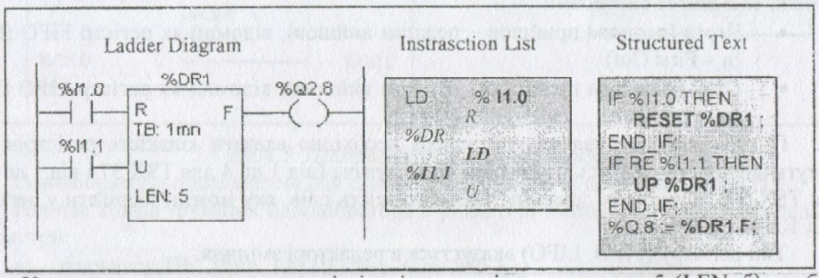
Якщо одночасно активізуються входи I та O перевага надається команді зберігання.

У наведеному прикладі значення слова %MW34 записується у регістр %R2 по команді на вході %I1.2 при умові, що регістр не заповнений (%R2.F =0). Зчитування слова з регістру відбувається по команді на вході %I1.3, при умові, що регістр не пустий (%R2.E =0). Це значення записується у слово %MW20.

2.7.4.5. Програмування драм-контролерів

Драм-контролер імітує роботу електромеханічного барабанного командоапарата.

Для програмування драм-контролера, у редакторі змінних необхідно вказати кількість кроків драм-контролера LEN (від 1 до 16), заповнити матрицю, згідно з якою встановлюються значення бітів в вихідному слові драм-контроллера з якими ассоціюються зовнішні або внутрішні диск-ретні змінні, а також вибрати базу часу TB (1min, 1 s, 100 ms, 10 ms або 1 s).



У наведеному нижче прикладі кількість кроків встановлена 5 (LEN=5), за базу часу вибрана величина 1min, а матриця має вигляд:

На мові Structured Text при програмуванні драм-контролера використовуються дві інструкції:

- RESET %DRi - ініціалізація драм-контролера, тобто переведення його у 0-ий крок;
- UP %DRi – команда переведення контролера на наступний крок, яка виконується по передньому фронту сигналу на вході U.

Кроки

Значення бітів	0	1	2	3	4	Адреси
0	1	0	1	0	1	%Q2.0
1	0	1	1	0	1	%Q2.1
2	0	1	0	0	1	%Q2.2
3	1	0	0	0	1	%Q2.3
4	0	0	0	1	0	%Q2.4

Основні системні біти ПЛК TSX Micro та TSX Premium

Біти	Функції	Опис
%S0*	Холодний перезапуск	Зазвичай у стані 0. У момент холодного перезапуску ПЛК установлюється в 1. Скидається в 0 системою. Може бути змінений з терміналу або програмно.
%S1*	Теплий перезапуск	Зазвичай у стані 0. У момент холодного перезапуску ПЛК установлюється в 1. Скидається в 0 системою. Може бути змінений з терміналу або програмно.
%S4* до %S7*	База часу	%S4 – база часу 10 мс; %S5 – база часу 100 мс; %S6 – база часу 1 с; %S7 – база часу 1 хв.
%S9**	Встановлення виходів у 0	Зазвичай у стані 0. Якщо %S9=1 – усі дискретні і аналогові виходи встановлюються в 0.Біт керується програмно або з терміналу.
%S9***	Встановлення виходу у аварійний режим	Зазвичай у стані 0. Якщо %S9=1 – усі дискретні виходи на всіх шинах встановлюються у 0 або 1 залежно від вибору, зробленого під час конфігурування
%S10*	Помилка входів/виходів	Зазвичай у стані 0. Встановлюється в 1 системою під час виявлення помилки входів/виходів.
%S11*	Перевищення сторожового таймера	Зазвичай у стані 0. Встановлюється в 1 системою у момент перевищення часу сторожового таймера, коли ПЛК переходить у режим STOP.
%S18*	Арифметичне переповнення	Зазвичай у стані 0. Встановлюється в 1 системою у момент переповнення операцій з словами.
%S19*	Перевищення періоду задачі	Зазвичай у стані 0. Встановлюється в 1 системою у момент перевищення періоду задачі.
%S20*	Переповнення індексу	Зазвичай у стані 0. Встановлюється в 1 системою коли адреса індексованого об'єкта стане меньшою або більшою числа заданого в процесі конфігурування
%S30*	Керування задачою MAST	Зазвичай у стані 1 – задача MAST активізована. Якщо %S30=0 - задача MAST деактивізується.
%S31*	Керування задачою FAST	Зазвичай у стані 1 – задача MAST активізована. Якщо %S30=0 - задача FAST деактивізується.
%S38*	Керування обробкою подій	Зазвичай у стані 1 – задача обробки подій активізована. Якщо %S30=0 - задача обробки подій деактивізується.
%S49**	Реактивація дискретних виходів	Зазвичай у стані 0. Біт всановлюється у 1 користувачем для реактивації несправного дискретного виходу.
%S50*	Поновлення дати і часу	Зазвичай у стані 0. При встановленні біта в 1, дозволяється змінювати значення слів %SW50-%SW53, де зберігаються значення дати і часу.
%S59*	Поновлення дати і часу у %SW59	Зазвичай у стані 0. Якщо %S59 =0, система не керує %SW59; якщо %S59 =1, система керує %SW59.
%S68*	Стан батарейки резервування RAM	%S68=0 – батарейка присутня і функціонує; %S68=1 – батарейка відсутня або не функціонує.
%S80*	Скидання лічильників повідомлень	Зазвичай у стані 0. Біт встановлюється у 1 користувачем для скидання лічильника повідомлень, які зберігаються у словах %SW80-%SW86.

Біти	Функції	Опис
%S96**	Коректність резервної копії прикладної програми	%S96=0, якщо резервна копія коректна; %S96=1, якщо резервна копія некоректна;
%S100*	Протокол термінального порта	Біт управляється системою в залежності від типу пристрою, який підключений до термінального порта: %S100=0 – протокол ведущого UNI-TELWAY; %S100=1 – протокол веденого UNI-TELWAY;
%S118***	Загальна помилка I/O на FIPIO шині	Зазвичай у стані 0. Біт встановлюється у 1 системою якщо виникає помилка на пристроях, які під'єднані до FIPIO шини.
%S119***	Загальна помилка I/O на X шині	Зазвичай у стані 0. Біт встановлюється у 1 системою якщо виникає помилка на пристроях, які під'єднані до X шини.

Примітка. * – системні біти загальні для TSX Micro та TSX Premium; ** – системні біти тільки для TSX Micro; *** – системні біти тільки для TSX Premium.

Додаток 2.

Основні системні слова ПЛК TSX Micro та TSX Premium

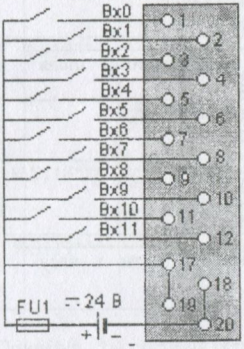
Слово	Функція	Опис
%SW0*	Час циклу задачі MAST	Використовується для зміни часу циклу (періо-дичний режим). Змінюється в межах 1-255 мс. При циклічному режимі %SW0=0.
%SW1*	Час циклу задачі FAST	Використовується для зміни часу періоду задачі FAST. Змінюється в межах 1-255 мс.
%SW8*	Керування читанням входів для задач	Заборона фази опитування входів кожної задачі: %SW8:X0 =1 заборона для задачі MAST; %SW8:X1 =1 заборона для задачі FAST.
%SW11*	Час сторожового таймера	Вміщує тривалість сторожового таймера в межах від 10 до 500 мс.
%SD18*	Лічильник абсо-лютного часу	Подвійне слово для розрахунку тривалості. Воно змінюється кожні 1/10 секунди (навіть коли ПЛК у режимі STOP).
%SW30*	Час виконання задачі MAST	Показує тривалість виконання останнього циклу задачі MAST
%SW31*	Максимальний час виконання задачі MAST	Показує максимальну тривалість виконання циклу задачі MAST
%SW32*	Мінімальний час виконання задачі MAST	Показує мінімальну тривалість виконання циклу задачі MAST
%SW33*	Час виконання задачі FAST	Показує тривалість виконання останнього циклу задачі FAST
%SW34*	Максимальний час виконання задачі FAST	Показує максимальну тривалість виконання циклу задачі FAST

Слово	Функція	Опис
%SW35*	Мінімальний час виконання задачі FAST	Показує мінімальну тривалість виконання циклу задачі FAST
%SW48*	Кількість подій	Показує кількість подій, опрацьованих після останнього холодного рестарта
%SW49* по %SW53*	Функції годинника реального часу	Вміщує у BCD-форматі поточні значення: %SW49: день неділі (1-Пн,….7-Нед); %SW50: Секунди (SS00); %SW51: Години/хвилини (ГГХХ); %SW52: Місяць/день (ММДД); %SW53: Сторіччя/рік (ССРР).
%SW54* по %SW58*	Функції годинника реального часу	Вміщує у BCD-форматі значення дати і часу моменту зупинки ПЛК: %SW54: Секунди/день неділі; %SW55: Години/хвилини; %SW56: Місяць/день; %SW57: Сторіччя/рік; %SW58: старший байт вміщує день неділі.
%SW58* (молодший байт)	Код останньої зупинки ПЛК	%SW58 =1, перевод із RUN у STOP з терміналу; %SW58 =2, помилка у програмі; %SW58 =4, із-за збоїв у живленні ПЛК; %SW58 =5, із-за апаратних збоїв; %SW58 =6, по інструкції HALT.
%SW59*	Настроювання значень дати/часу	Настроювання відбувається бітами: **Збільшення Зменьшення Параметр** біт 0 біт 8 день неділі біт 1 біт 9 секунди біт 2 біт 10 хвилини біт 3 біт 11 години біт 4 біт 12 день біт 5 біт 13 місяць біт 6 біт 14 рік біт 7 біт 15 сторіччя Можливість настроювання дозволяється бітом%S59.
%SW67** по %SW69**	Керування режимом WORD	Якщо %SW69=1, ці слова дозволяють використовувати дисплейний блок у режимі WORD: %SW67: команда і стан режиму WORD; %SW68: максимальний і плинний індекс; %SW69: номер першого об'єкту в зоні відображення
%SW124*	Тип несправності процесора	Вміщує тип останньої несправності процесора
%SW125*	Тип блокуючої помилки	Вміщує тип останньої блокуючої помилки
%SW126*	Адреса інструкції блокуючої помилки	Вміщує адресу інструкції, яка сгенерувала блокуючу помилку.
%SW128*** по %SW146***		Слова для відображення режиму на FIPIO шині

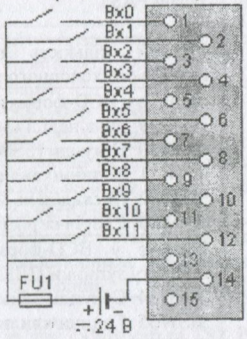
Примітка. * – системні біти загальні для TSX Micro та TSX Premium; ** – системні біти тільки для TSX Micro; *** – системні біти тільки для TSX Premium.

Схекми підключень модулів ПЛК TSX Micro

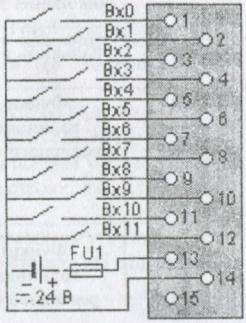
TSX DEZ 12D2K



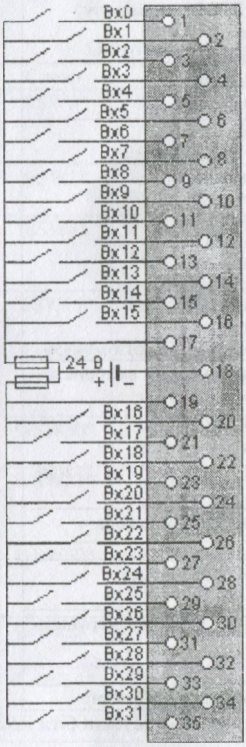
TSX DEZ 12D2
(позитивна логіка)



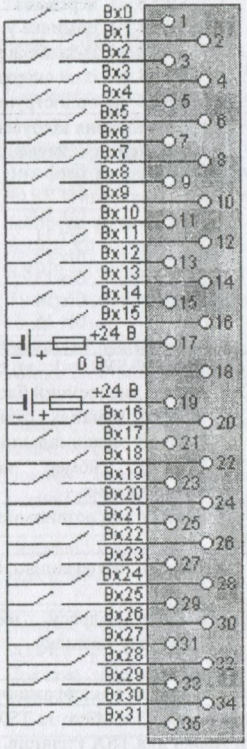
TSX DEZ 12D2
(негативна логіка)



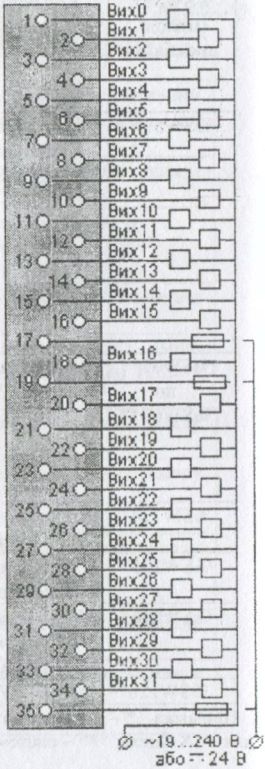
TSX DEZ 32D2



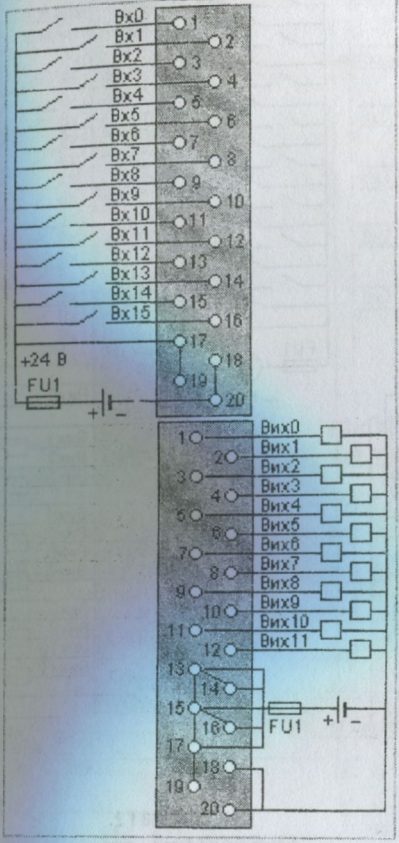
TSX DSZ 32T2



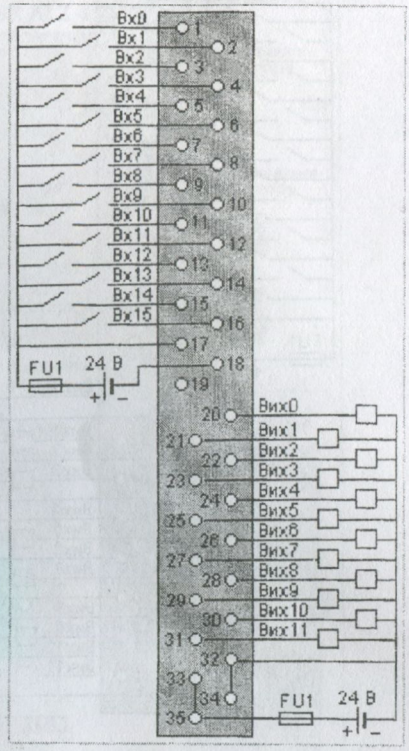
TSX DSZ 32R5



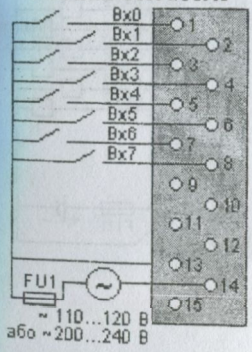
TSX DMZ 28DTK



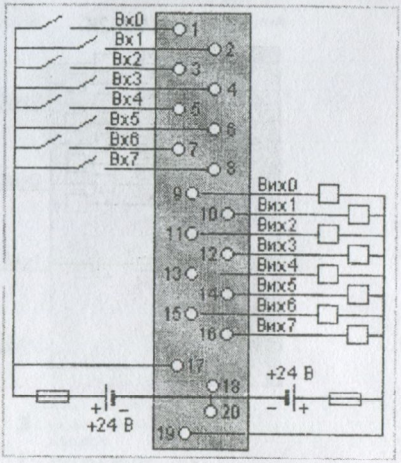
TSX DMZ 28DT



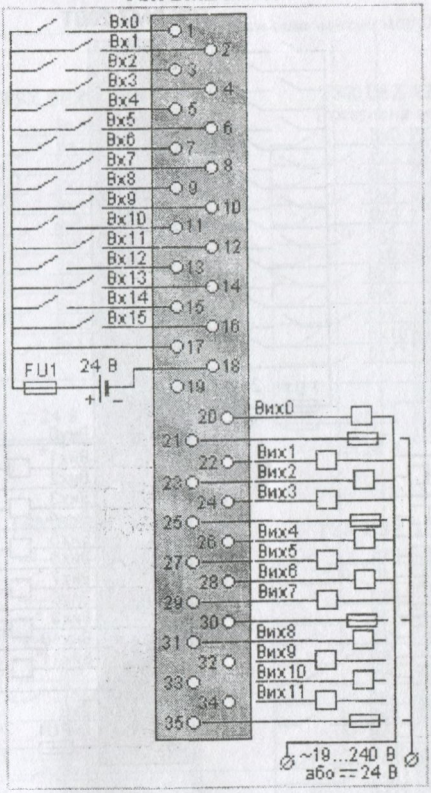
TSX DEZ 08A4/08A5



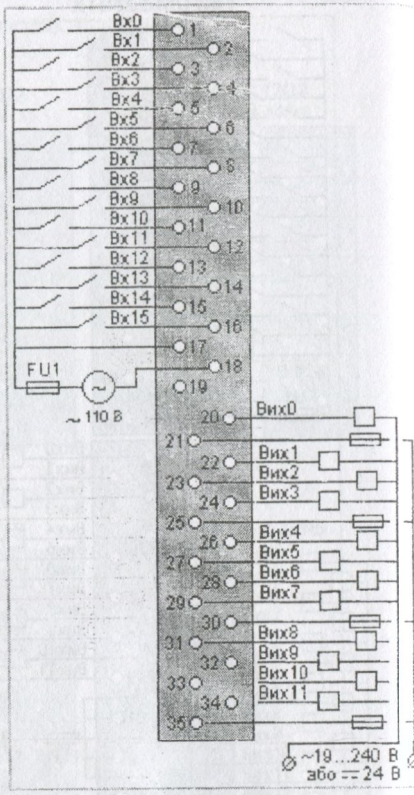
TSX DMZ 16DTK



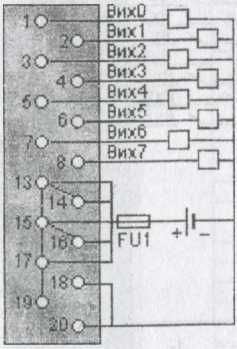
TSX DMZ 28DR



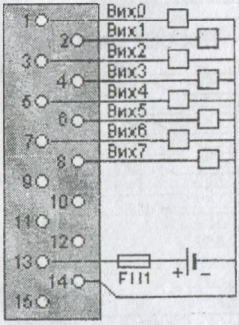
TSX DMZ 28AR



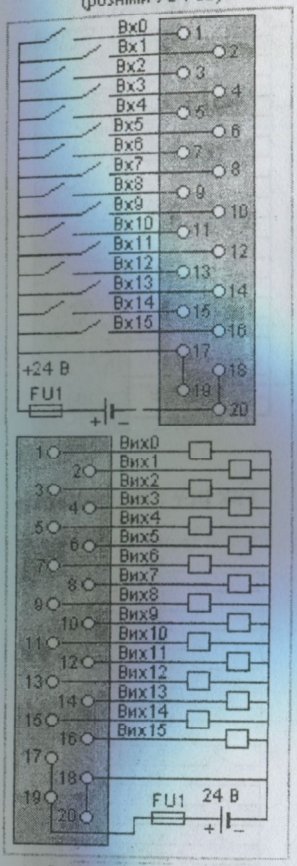
TSX DMZ 08T2K



TSX DSZ 08T2

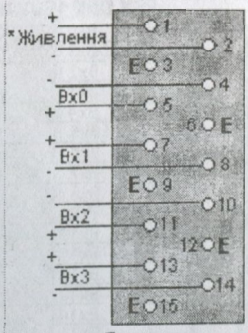


TSX DMZ 64 DTK
(розніми A0 і B0)



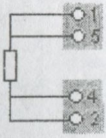
* підключення рознімів A1 і B1 аналогічне

TSX AEZ 414



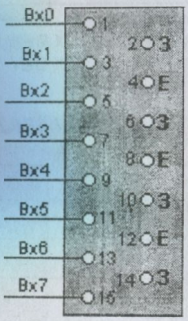
*Живлення датчиків термометрів опору

Приклад підключення термометра опору



E - підключення екрану

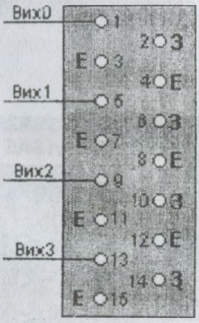
TSX AEZ 801/802



3 - загальна точка каналів

E - підключення екрану

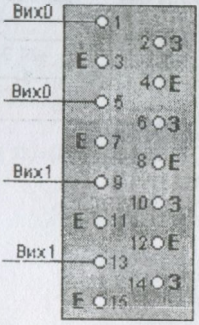
TSX ASZ 401



3 - загальна точка каналів

E - підключення екрану

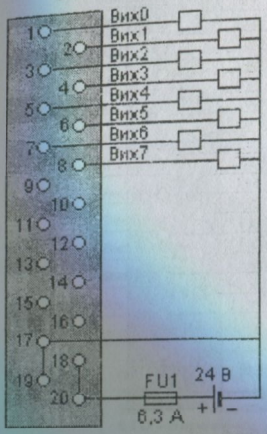
TSX ASZ 200



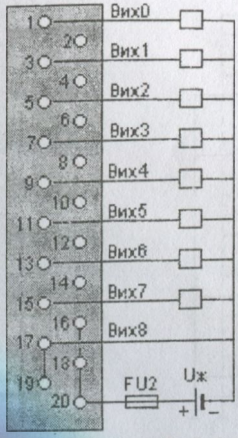
3 - загальна точка каналів

E - підключення екрану

TSX DSY 08T2

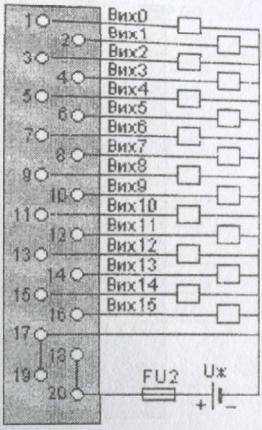


TSX DSY 08T22/08T31



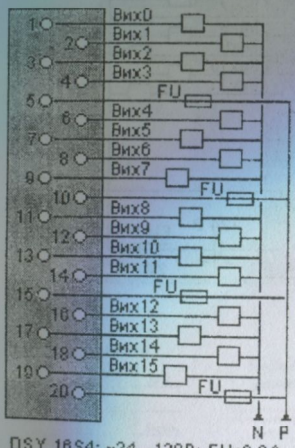
DSY 08T22: Uж=24В; FU2=16А
DSY 08T31: Uж=48В; FU2=10А

TSX DSY 16T2/16T3



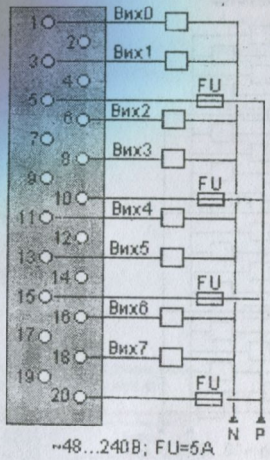
DSY 16T2: Uж=24 В, FU2=6,3А
DSY 16T3: Uж=48 В, FU2=10А

TSX DSY 16S4/16S5



DSY 16S4: ~24...120В; FU=6,3А
DSY 16S5: ~48...240В; FU=5А

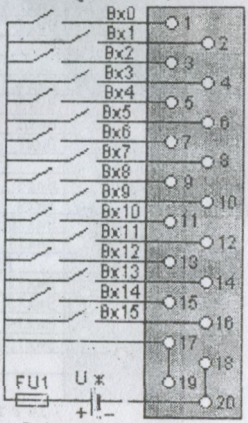
TSX DSY 08S5



~48...240В; FU=5А

TSX DEY 32D2K/64D2K/ 32D3K

(рознім A)

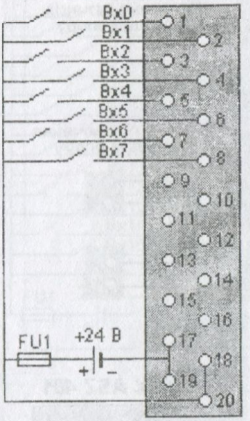


* підключення розніма B, C, D аналогічне
A - канали 00 - 15
B - канали 16 - 31
C - канали 32 - 47
D - канали 48 - 63

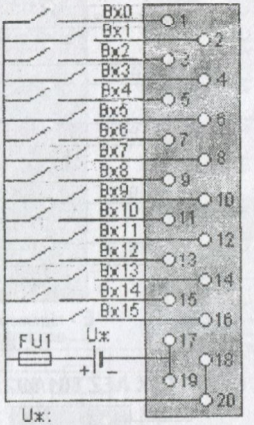
Uж = :
+24 В для DEY 32 D2K та DEY 64 D2K;
+48 В для DEY 32 D3K

Схекми підключень модулів ПЛК TSX Premium

TSX DEY 08D2

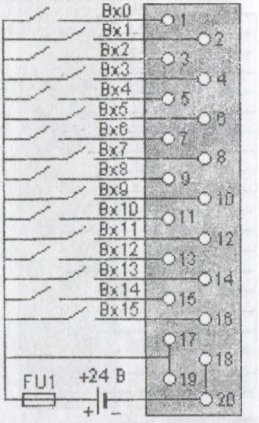


TSX DEY 16D2/D3

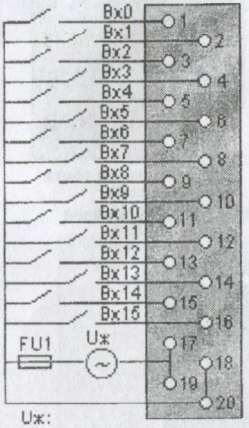


Uж:
+24 В для TSX DEY 16D2;
+48 В для TSX DEY 16D3

TSX DEY 16FK

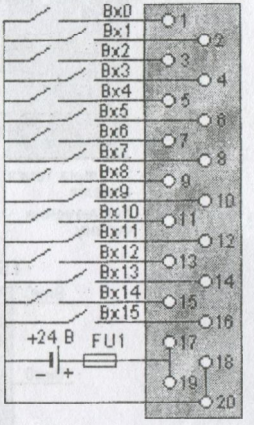


TSX DEY 16A2/16A3/ 16A4/16A5

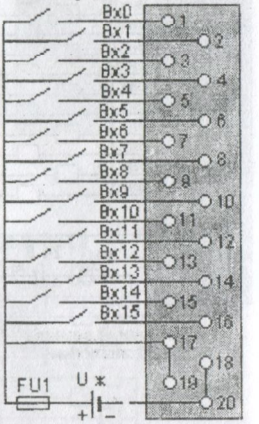


Uж:
~ 24 В для TSX DEY 16 A2
~48 В для TSX DEY 16 A3
~110 В для TSX DEY 16 A4
~220 В для TSX DEY 16 A5

TSX DEY 16A2
(негативна логіка)



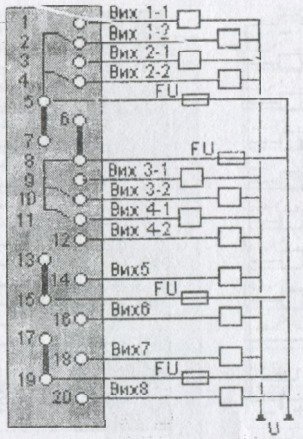
TSX DEY 32D2K/64D2K/ 32D3K
(рознім A)



* підключення рознімів B, C, D аналогічне
A - канали 00 - 15
B - канали 16 - 31
C - канали 32 - 47
D - канали 48 - 63

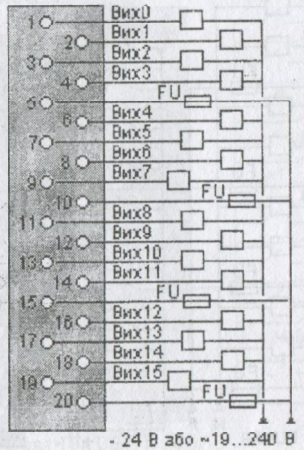
Uж = :
+24 В для DEY 32 D2K та DEY 64 D2K;
+48 В для DEY 32 D3K

TSX DSY 08R5A/08R4D



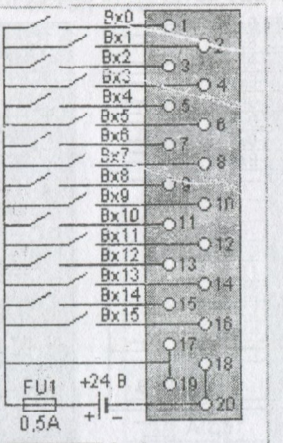
DSY 08RSA: U=~19...240 В або
-19...80В
DSY 08R4D: U= - 24...130В
FU= 6,3А

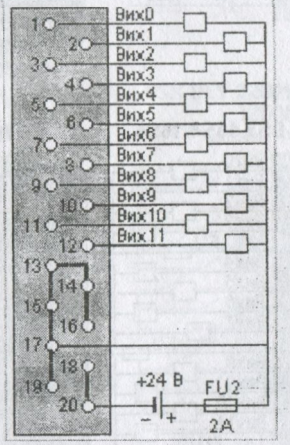
TSX DSY 08R5/16R5



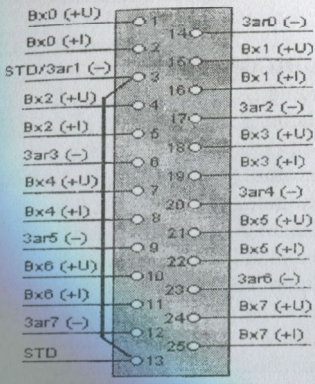
- 24 В або ~19...240 В

TSX DMY 28FK



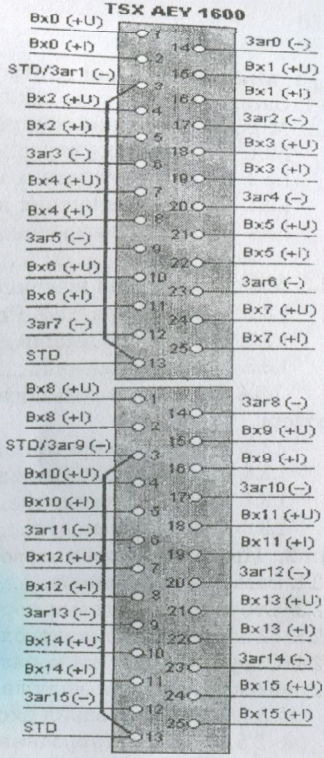


TSX AEY 800



Вх...(+U) - вхід по напрузі
Вх...(+I) - вхід по струму
Заг...(–) - загальний вхід каналу
STD - перемичка для виявлення видалення розніму

TSX AEY 1600

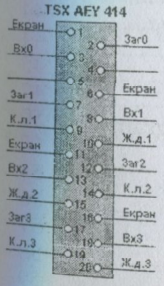


TSX ASY 410



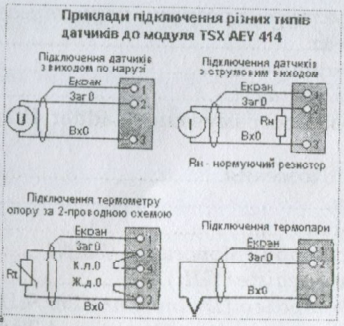
Вих... (U) - вихід по напрузі
Вих...(I) - вихід по струму
Заг...( ) - загальний вихід каналу

TSX AEY 414



Вх..... - вхід канала
Заг... - загальний вхід канала
Ж.д..... - живлення датчика
К.л..... - компенсація лінії

Приклади підключення різних типів датчиків до модуля TSX AEY 414



ЗМІСТ

Розділ 1. Мікропроцесорні контролери MODICON TSX 3
1.1. Загальна структура технічних засобів 3
1.2. Мікропроцесорний контролер Modicon TSX Micro 5
1.2.1. Фізична структура 5
1.2.2. Модулі дискретних входів-виходів 8
1.2.3. Модулі аналогових входів-виходів 11
1.2.4. Модулі лічильників 16
1.2.5. Asi-шина 18
1.3. Мікропроцесорні контролери Modicon TSX Premium 20
1.3.1. Архітектура TSX Premium 20
1.3.2. Модулі процесорів 22
1.3.3. Блоки живлення 23
1.3.4. Принципи розміщення й адресації модулів у контролері 25 26
1.3.5. Модулі дискретних входів-виходів 29
1.3.6. Модулі аналогових входів-виходів 31
1.3.7. Модулі лічильників 32
1.3.8. Спеціальні модулі 33
Розділ 2. Програмування контролерів MODICON TSX 33
2.1. Загальні положення 34
2.2. Структура програми користувача 35
2.3. Конфігурування контролера 38
2.4. Структура пам'яті користувача 39
2.5. Конфігурування модулів 39
2.5.1. Конфігурування дискретних модулів 39
2.5.1.1. Конфігурування модулів дискретних входів 41
2.5.1.2. Конфігурування модулів дискретних виходів 41
2.5.2. Конфігурування аналогових модулів 41
2.5.2.1. Конфігурування модулів аналогових входів 43
2.5.2.2. Конфігурування модулів аналогових виходів 43
2.5.3. Конфігурування модулів рахункових каналів 44
2.6. Розробка прикладної програми користувача 44
2.6.1. Загальні положення 44
2.6.1.1. Вибір структури програми користувача 44
2.6.1.2. Типи змінних 45
2.6.1.3. Адресація каналів 47
2.6.1.4. Формати змінних 50
2.6.2. Технологічна мова програмування Ladder Diagram (LD) 50 54
2.6.2.1. Загальні оложення 56
2.6.2.2. Таймери 57
2.6.2.3. Лічильники 58
2.6.2.4. Одновібратори (Monostable %MNi) 59
2.6.2.5. Регістри (Register %Ri) 61
2.6.2.6. Драм-контролер (Drum controller %DRi) 61
2.6.3. Мова структурованого тексту (Structured Text) 61

2.6.3.1. Загальні положення 63
2.6.3.2. Керуючі структури 66
2.6.4. Мова інструкцій (Instruction List) 66
2.6.4.1. Загальні положення 69
2.6.4.2. Інструкції MPS, MRD, MPP 70
2.6.4.3. Принципи програмування функціональних блоків 70
2.6.4.4. Правила складання програм на мові IL 72
2.6.5. Мова "Графсет" (Grafcet) 72
2.6.5.1. Загальні положення 74
2.6.5.2. Структура задачі на мові Графсет 75
2.6.5.3. Процедура програмування на мові Графсет 79
2.6.6. ПІД-регулювання 84
2.7. Приклади виконання фрагментів програм 84
2.7.1 Програмування логічних інструкцій 84
2.7.1.1 Інструкції опитування 84
2.7.1.2. Інструкції присвоювання 84
2.7.1.3 Логічне "И" 85
2.7.1.4. Логічне "ИЛИ" 85
2.7.1.5. Інструкція "Исключающее ИЛИ" 85
2.7.2. Інструкції з числами 85
2.7.2.1. Інструкції порівняння 86
2.7.2.2. Інструкції присвоювання 87
2.7.2.3. Арифметичні інструкції з цілими числами. 88
2.7.2.4. Логічні інструкції 88
2.7.2.5. Правила виконання інструкцій з числами 89
2.7.3. Програмні інструкції 89
2.7.3.1. Підпрограми 90
2.7.3.2. Програмована зупинка виконання програми. 90
2.7.3.3. Програмований перехід 91
2.7.4. Програмування функціональних блоків 91
2.7.4.1. Програмування таймерів 91
2.7.4.2. Програмування лічильників 92
2.7.4.3. Програмування одновібраторів 93
2.7.4.4. Програмування регістрів 94
2.7.4.5. Програмування драм-контролерів 95
Додаток 1. Основні системні біти ПЛК TSX Micro і TSX Premium 96
Додаток 2. Основні системні слова ПЛК TSX Micro і TSX Premium 98
Додаток 3. Схеми під'єднання до модулів TSX Micro 102
Додаток 4. Схеми під'єднання до модулів TSX Premium .....