

Національний університет „Львівська політехніка”
Інститут енергетики і систем керування
Кафедра електромехатроніки та комп’ютеризованих електро-
механічних систем

Василь Ткачук



ОСНОВИ МЕХАТРОНІКИ

Конспект лекцій

3-тє видання, стереотипне

Рекомендовано Вченою Радою
Інституту енергетики і систем керування

Видавництво "Магнолія 2006"

Львів – 2025

Відтворення цієї книги або будь-якої її частини заборонено без письмової згоди видавництва. Будь-які спроби порушення авторських прав будуть переслідуватися у судовому порядку.

УДК 621.313.294

Рекомендовано науково-методичною радою
Інституту енергетики і систем керування НУ "Львівська політехніка"

Ткачук В.

Основи мехатроніки. Конспект лекцій для студентів спеціальності «Електроенергетика, електротехніка та електромеханіка» – 3-тє видання, стереотипне – Львів: Видавництво "Магнолія 2006" , 2025.- 266 с.

Конспект лекцій призначено для студентів кваліфікації бакалавр з електроенергетики, електротехніки та електромеханіки, а також як порада для викладачів-лекторів.

Також конспект призначений для задоволення потреб всіх тих, хто прагне опанувати проблеми, які пов'язані з синергетикою механіки, пневматики, гідравліки, електротехніки, електроніки, телекомунікації і комп'ютерної техніки – в загальному мехатронікою. Також буде корисною допомогою для інженерів і технічних працівників фірм, які займаються виробництвом мехатронних пристроїв, їх продажем та обслуговуванням.

Укладач:

Ткачук В.І. - доктор технічних наук, професор, професор кафедри електро-мехатроніки та комп'ютеризованих електромеханічних систем Національного університету „ Львівська політехніка”.

Віповідальний за випуск:

Біляковський І.Є. – кандидат техн. наук, доцент, доцент кафедри електро-мехатроніки та комп'ютеризованих електромеханічних систем Національного університету „ Львівська політехніка”.

Рецензенти:

Лозинський О.Ю. – доктор технічних наук, професор, заслужений діяч науки і техніки України, завідувач кафедри електромехатроніки та комп'ютеризованих електромеханічних систем Національного університету «Львівська політехніка»

Стахів П.Г. - доктор технічних наук, професор, завідувач кафедри теоретичної і загальної електротехніки Національного університету „ Львівська політехніка”.

ISBN 978-617-574-136-8

©Ткачук В.І., 2025

© Видавництво "Магнолія 2006", 2025

З М І С Т

Передмова	4
Лекція 1. Вступ до мехатроніки. Керування, регулювання та управління	5
Лекція 2. Основні принципи техніки керування.....	11
Лекція 3. Електричне керування.....	19
Лекція 4. Системи секвенційного керування.....	29
Лекція 5. Елементи електронних систем керування.....	33
Лекція 6. Електромеханічні перетворювачі.....	38
Лекція 7. Пневматичне керування.....	52
Лекція 8. Пневматичні вентилі	69
Лекція 9. Функціональні діаграми. Пропорційне керування	79
Лекція 10. Електропневматичні системи керування	84
Лекція 11 Гідравлічне керування	90
Лекція 12. Гідравлічні приводи	100
Лекція 13. Двійкове і цифрове керування	120
Лекція 14. Програмовані пристрої керування (мікроконтролери).....	133
Лекція 15. Базові функції. Логічні функції	140
Лекція 16. Сенсорика. Аналогові сенсори	146
Лекція 17. Основи і методи регулювання	162
Лекція 18. Регулятор і система регулювання	170
Лекція 19. Регулювання статичних об'єктів	177
Лекція 20. Техніка числового керування	183
Лекція 21. Числове керування	191
Лекція 22. Програмування систем НС.....	204
Лекція 23. Корекція розмірів інструментів	210
Лекція 24. Робототехніка.	217
Лекція 25. Програмування роботів	222
Лекція 26. Керування роботами.	226
Лекція 27. Комп'ютерно-орієнтоване виробництво	232
Лекція 28. Планування виробництвом за допомогою комп'ютера	238
Лекція 29. Управління якістю.	246
Лекція 30. Комп'ютерна техніка.	257

ПЕРЕДМОВА

Само слово мехатроніка складається з двох слів: механіка і електроніка. На практиці це означає, що головними керованими або регульованими величинами мехатронної системи можуть бути положення, швидкість і прискорення під час лінійного чи обертального переміщення; сила і обертальний момент; тиск і приплив рідини або газу; частота механічних, а також акустичних і оптичних коливань; температура; електричний струм, який викликає рух механічних елементів в магнітному полі. Означає це також, що головна різниця між механічною і мехатронною системою полягає на зміні керовальної або задавальної величини (вхідної або наставленої величини): в першому випадку є механічні або електромеханічні величини, в другому – такі величини, як напруга і електричний струм, а також інші електричні величини.

Мехатроніка - нова, що динамічно розвивається царина науки, техніки і промисловості, вимагає щораз більше фахово підготовлених до опанування визнаних сучасності спеціалістів.

Отже мехатроніка – це об'єднувальна комбінація механіки, електронного керування і системного мислення при проектуванні і процесі виробництва і даний конспект лекцій містить основні відомості з цієї нової науково-технічної дисципліни, яка надзвичайно динамічно розвивається, з особливою увагою до найбільш істотної її частини – керування й регулювання процесами виробництва, сенсорики, приводів виконавчих пристроїв і робототехніки, інформатики і керування процесом виробництва і його якості. Ці проблеми одночасно відносяться як до технічних пристроїв, так і до устаткувань промислових процесів неперервного характеру (перетворення енергії і інформації) і дискретного (виробництва продуктів).

В конспекті лекцій подано технологію і класифікацію, а також засади будови основних пристроїв автоматики, робототехніки і їх поєднань. Описано, передусім, аспект практичного застосування мехатроніки, різноманітне вирішення згаданих пристроїв: від механічних через пневматичні до електронних – найновіших, процесорної генерації. Доповненням основних відомостей з галузі будови загальних видів пристроїв є перегляди типових вирішень з прикладами і ефектами застосувань. Звернено також увагу на тенденцію розвитку цієї області.

Конспект лекцій задовольняє всім вимогам програмних основ в сфері навчання програмного блоку – «Основи мехатроніки» і забезпечує вивчення наступних програмних блоків. Згідно з програмними основами навчальний матеріал поділено на тематичні розділи, які торкаються суті мехатроніки, електротехнічних мереж, а також електронних, пневматичних і гідравлічних приладів. «Мультимедійні» ілюстрації добре відповідають як призвичаєним до «інтернетівського» дизайну молодих читачів, так і може бути добре використана як дидактична допомога викладачам під час проведення ними занять.

Потрібно звернути увагу на процес створення спеціальності «мехатроніка» в вищих навчальних закладах від другого і до четвертого рівнів акредитації.

Конспект лекцій призначено також для студентів післядипломної освіти, інженерів і технічних кадрів фірм, які займаються виробництвом сучасних мехатронних пристроїв і виробничих машин, а також для працівників фірм, які займаються їхнім продажем і сервісом – йдеться тут перед усім про відносно легку, позбавлену трудної теорії, практичну можливість самостійного навчання і розширення особистого кругозору.

Конспект лекцій підготовлено у Львівській політехніці в інституті енергетики і систем керування, де в одному з перших з українських вищих навчальних закладів готуються спеціалісти зі спеціалізації «Електромеханотроніка». Багаторічний досвід у викладанні електромеханотроніки дозволив підготувати доступний, заохочувальний до науки конспект лекцій з основ мехатроніки через багаточисельні рисунки, фотографії, креслення і таблиці.

Автор дякує рецензентам за увагу – буде вдячним також читачам за зміни і зауваження.

Тема: Вступ до мехатроніки.

1. Предмет вивчення в курсі "Мехатроніка".
2. Керування, регулювання та управління: *керування аналогове, бінарне та цифрове; комбінаційне та секвенційне керування; постійнозапрограмоване і програмувальне керування; регулювання; управління.*

Згідно визначення Міжнародної Федерації Теорії Машин і Механізмів *мехатроніка* – це комбінація механіки, електронного керування і системного мислення під час проектування виробів і виробничих процесів. Само слово мехатроніка складається з двох слів: механіка і електроніка. На практиці це означає, що надалі, як в типовій механічній системі, головними керованими або регульованими величинами (вихідними величинами) мехатронної системи можуть бути між іншими: положення, швидкість і прискорення під час лінійного чи обертального переміщення; сила і обертовий момент, які пов'язані з реалізацією переміщення або є їхнім наслідком; тиск і виникаючий з нього приплив рідини або газу; частота механічних, а також акустичних і оптичних коливань; температура і викликаючий нею тепловий обмін (видалення тепла); рух шарів рідини і газу під впливом зовнішніх чинників; електричний струм, який викликає рух механічних елементів в магнітному полі, або, перед усім, механічні або електромеханічні величини, які пов'язані з загальним розумінням руху, сили або моменту. Означає це також, що головна різниця між механічною і мехатронною системою полягає на зміні керувальної або задавальної величини (вхідної або наставленої величини): в першому випадку є механічні або електромеханічні величини, в другому – такі величини, як напруга і натяг електричного струму, а також інші електричні величини, які пов'язані двома змінними.

Перший раз терміном «мехатронічний» названо пристрій керування електричним мотором, який створений в 1971 р. в японській фірмі Yaskawa Electric Corporation і ця назва охоронялась торговим знаком аж до 1982 р. Вже від самого початку використання цього слова пов'язувалось з застосуванням електричного керування до механічних і електромеханічних систем.

В європейських вищих навчальних закладах підготовку інженерів-мехатроніків розпочато вже в другій половині 90-их років, а декілька років пізніше, в технічних фахових школах, підготовку техніків-мехатроніків. Вироби української промисловості витримують конкуренцію в рамках об'єднаної Європи тільки тоді, коли виконуються вимоги мехатроніки – корисного для кінцевого продукту поєднання механічних, електротехнічних, електронних і інформаційних властивостей. Вже звичайні мехатронні продукти, які щораз тісніше оточують нас не тільки у виробничій сфері, але також і в приватному житті: починаючи від медіальної, комп'ютерної і побутової сфери і закінчуючи сучасними медичної інженерії, оброблювальної, робототехнічної і транспортної сферами.

Мехатроніка - нова, що динамічно розвивається царина науки, техніки і промисловості, вимагає щораз більше фахово підготовлених до опанування визнань сучасності спеціалістів.

Отже мехатроніка – це об'єднувальна комбінація механіки, електронного керування і системного мислення при проектуванні і процесі виробництва і в даному навчальному предметі будуть розглядатись основні відомості з цієї нової науково-технічної дисципліни, яка надзвичайно динамічно розвивається, з особливою увагою до найбільш істотної її частини – керування й регулювання процесами виробництва, сенсорики, приводів виконавчих пристроїв і робототехніки, інформатики і керування процесом виробництва і його якості. Ці проблеми одночасно відносяться як до технічних пристроїв, так і до устаткування промислових процесів неперервного характеру (перетворення енергії і інформації) і дискретного (виробництва продуктів).

Подано в ньому технологію, класифікацію, а також засади будови основних пристроїв автоматики, робототехніки і їх поєднань. Вивчається, передусім, аспект практичного застосування мехатроніки, різноманітне вирішення згаданих пристроїв: від механічних через пневматичні до електронних – найновіших, процесорної генерації. Доповненням основних відомостей з галузі будови загальних видів пристроїв є перегляди типових вирішень з прикладами і ефектами застосувань. Звернено також увагу на тенденцію розвитку цієї області.

Керування, регулювання та управління. Для того щоб машини і установки могли самостійно, тобто автоматично працювати, вони обладнуються пристроями для керування, регулювання та

Питання для перевірки і засвоєння

1. Що розуміють під поняттям якість?
2. Нарисувати типовий перебіг величини якогось показника якості.
3. Перерахувати типи похибок недоліків.
4. Що таке ТГМ?
5. Яка тематика висвітлюється в книжці керування якістю?
6. Яким чином розділені норми PN ISO 9000-9004?

Тема: Комп'ютерна техніка:

- будова комп'ютера,
- об'єктно орієнтоване програмне забезпечення,
- передавання даних, цифрові сигнальні входи і виходи,
- інтерфейс DDE,
- складові частини програмного забезпечення.

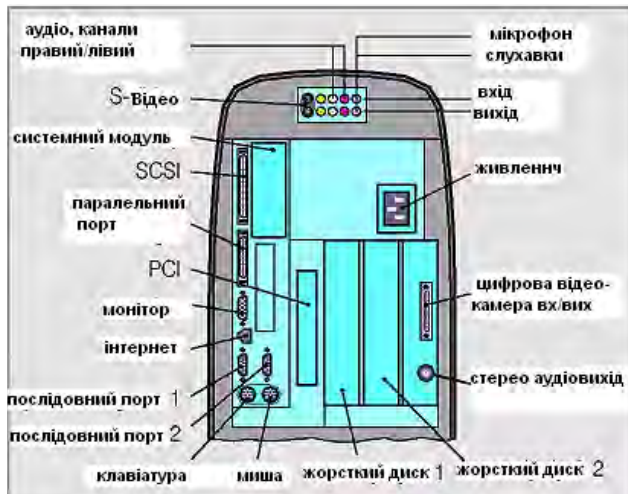


Рис.30.1. Розніми в компютері

вають також портами) (рис. 30.1).

Послідовний порт

Стандартно в комп'ютерах вбудовані принаймні два послідовні порти. Позначаються вони V24 або RS232C. Ці порти стандартизовані і мають 9 або 25 контактів (так званих пінів). Для програмних посилань їх позначають як COM1, COM2... і т.д. (COM скорочено з англ. *Communication*).

Послідовні порти потрібно попередньо конфігурувати - ввести наступні параметри:

- номер розніму = 0 = COM1, 1 = COM2
- швидкість модуляції = 300, 600, 1200, 2400, 4800, 9600, 1920
- кількість бітів даних = 7 або 8
- кількість бітів стоп = 1 або 2
- парність = N = немає, E = парний, 0 = непарний

Ці налаштування здійснюють програмно. На рис. 30.2 представлено приклад налаштування рознімів в програмі LabView.

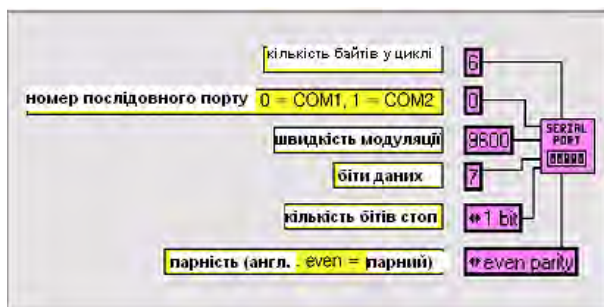


Рис.30.2. Встановлення послідовного порту в програмі LabView

Будова комп'ютера

До складу типового комп'ютерного комплекту входять:

- комп'ютерні плати в корпусі, в якому також розміщені приводи дискет, дисків CD-ROM, блок живлення тощо, а також закріплені зазвичай на задній стінці корпусу різними периферійних пристроїв (рис. 20.1):
- клавіатура,
- монітор,
- миша,
- динаміки.

Розніми

Для використання комп'ютера з метою вимірювання, керування або регулювання, необхідне застосування рознімів (які називаються

Паралельний порт

У кожному комп'ютері є також принаймні один паралельний порт. Називається він Центральним портом і має 36-пін. У програмному забезпеченні паралельним портам приписані позначення LPT1 і т.д. аж до LPT4. До портів цього типу приєднують, зокрема, принтери. Цей порт також використовують для дистанційного керування пристроями. У продажу для цієї мети є відповідні карти інтерфейсу.

Рознім *USB* (англ. *Universal Serial Bus* = універсальна послідовна магістраль (шина) дозволяє послідовне підключення багатьох, максимально 128 пристроїв, які також мають порт USB.

Вільні місця для під'єднань (розширення)

Як правило комп'ютер має вільні місця для карт комунікаційного інтерфейсу з зовнішніми пристроями.

Залежно від комп'ютера, є кілька варіантів:

- шина *ISA* = *Industrial Standard Architecture*,
- шина *PCI* = *Peripheral Component Interface*.

Шина PCI

Стандарт шини PCI був створений фірмою ІНТЕЛ. Це 64-бітова шина даних з максимальною частотою 33 МГц. Завдяки мультиплексуванню вона дозволяє поперемінне пересилання адрес і даних. Шину пов'язує з процесором пристрій керування, який уможливує обмін даними між магістраллю і периферійними пристроями під час обробки даних процесором.

Операційні системи

Стандартними операційними системами комп'ютерів PC є версії *Windows*. Наприклад, у промисловості застосовують операційну систему *Windows NT*. Система *Windows* є операційною системою, спрямованою на перероблення інформації, яка дозволяє одночасне виконання багатьох програм. Цю властивість називають багатозадачністю (*Multitasking*, англ. *Task* = завдання). На *рис. 20.3* представлено, як *Windows* записує всю інформацію в буферній пам'яті, класифікує і потім пересилає її до поточного застосування. Головна програма є власне кажучи тільки циклом, який підтримує стабільний хід цього процесу.

Комбінацією клавіш *Ctrl+Alt+Delete* можна відкрити вікно, в якому є список всіх активних на даний момент програм. За допомогою цього вікна можна також завершити будь-яку



Рис.30.3. Алгоритм роботи операційної системи Windows

програму. Можна це зробити також і тоді, коли якась програма працює неналежним чином. Переважно працюють на фоні інших завдань, наприклад, антивірусні програми – („Vshwin32”).

Інсталяція карт комунікаційного інтерфейсу для вимірювання, керування або регулювання

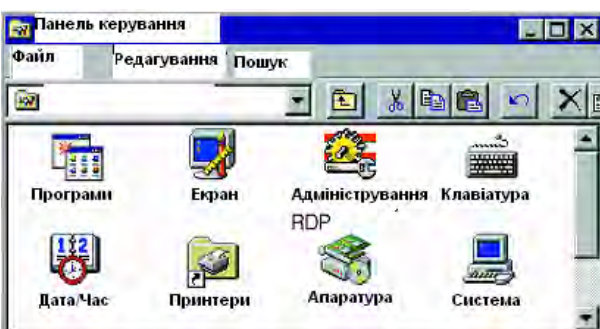


Рис.30.4. Піктограми у вікні «Панелі керування»

Якщо до вільного розніму розширення ми вкладемо карту, то після ввімкнення комп'ютера операційна система повинна виявити її існування. Якщо карта є типу „*Plug and Play*” (з англ. *вклади і вживай*), можливе її самостійне встановлення системою. У цей момент часу відбудеться призначення адрес і переривань. Виробник карти повинен поставляти і програмне забезпечення. Після встановлення система автоматично розпізнає карту інтерфейсу. Як правило, можливо є конфігурація карти, наприклад, встановлення меж вхідних напруг. З картою інтерфейсу виробник поставляє підпрограми, орієнтовані на різні мови програмування, наприклад, *VISUAL*

C, *DELPHI*, *VISUAL BASIC*, чи також *LabView*. За допомогою цих підпрограм можна вже розробити власні прикладні програми.

Інтерфейс споживача - об'єкти

Система *Windows* побудована з допомогою різних об'єктів, таких, як вікна, піктограми, діалогові вікна, меню тощо:

Вікно: Кожна програма, яка спілкується зі споживачем, повинна мати своє вікно, за допомогою якого отримує команди з клавіатури, мишки і на яке виводить перетворену інформацію. Всі інші

об'єкти допомагають функціям вікна. Вікно представляє собою вид об'єктних або програмних даних у комп'ютері. Після відкриття вікна вводиться в дію і активізується пов'язана з ним програма - після його закриття використання програми закінчується. Окантовка (рамка) зі смугами меню, заголовки, позначення тощо завжди безпосередньо керуються системою Windows; можна також збільшувати і зменшувати вікна. Від програміста залежать тільки вміст у вікні.

Піктограми: Піктограма є символом, який допомагає встановити зв'язок (як кнопка у вікні) з певною функцією. За допомогою піктограм споживач може викликати прикладні програми (рис. 30.4) Існує можливість зміни величини піктограм. Кожній програмі споживача приписується піктограма. Піктограми рисуються в редакторі малюнків і постійно пов'язані з певною програмою.

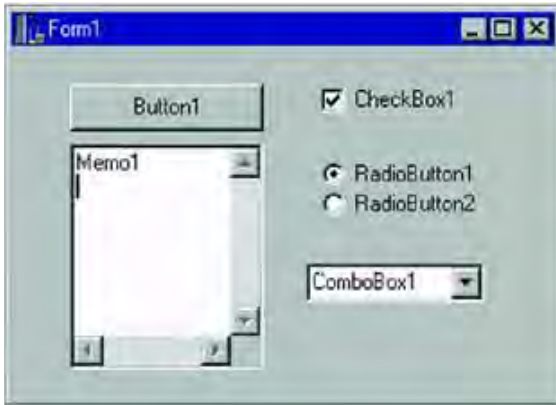


Рис.20.5. Приклад діалогових полів



Рис.30.6. Діалогове поле



Рис.30.7. Принцип об'єктно-орієнтованого поділу

функції встановлюються за допомогою класів (типів) об'єктів. Наприклад, це може бути регулятор класу PID.

З цього класу можна виділити правило регулювання типу PI, яке може працювати цілком самостійно. Ця обставина створює власне об'єкт - структурний блок. З одного класу можливе утворення дуже багатьох об'єктів. Цей принцип застосований у всій системі Windows. В оточенні програми можуть бути частково графічні процедури, які полягають у зверненні до бібліотеки і вміщенні у вікнах одиничних об'єктів. Пов'язують функціональні об'єкти у текстовому вигляді за допомогою відповідного

Діалогові вікна: Діалогові вікна надають споживачеві стандартні можливості відповіді на запитання системи. На рис. 30.5 показано кілька таких можливостей, серед яких можна виділити кнопки (Button), текстові поля (Memo), кнопки опцій (RadioButton), поля опцій (Combo-Box). Багатьом діалоговим вікнам відповідають часто використовувані, типові завдання, наприклад, читання, записування файлів або звернення до принтера. Ці стандартні діалогові вікна становлять частину модуля Windows SDK (Software Development Kit) і використовуються всіма програмами споживачів.

Діалогові і інформаційні поля (рис. 30.6) можуть мати дві різні видозміни, так звані модальні і немодальні. У випадку, коли відкрите модальне діалогове вікно, для продовження роботи з іншими задачами його треба закрити. Для немодальних вікон такої необхідності немає – можна працювати з іншими завданнями, ігноруючи це інформаційне вікно. Як правило, всі застереження є модального типу.

Об'єктно орієнтоване програмне забезпечення

Як в програмному забезпеченні для перетворення даних, так і в прикладному програмному забезпеченні, в широкому розумінні (промисловому, господарському), використовують об'єктну структуру. Здійснюється це шляхом поділу даного процесу на структурні блоки, які називають об'єктами і які містять всі дані та необхідні функції для роботи блоку. Одиничні об'єкти взаємодіють між собою шляхом обміну інформацією. Кожен об'єкт є автономний і орієнтований на певну сферу своїх завдань. На рис. 30.7 представлено приклад системи, яка керує процесом кондиціонування певного приміщення. Центр управління може відчитувати дані в приміщенні, яке треба кондиціонувати, але не може їх змінювати. Центр управління також хоч і може висилати значення завдань до регуляторів, але не може впливати безпосередньо на вибір регулювальної функції. Регулятор працює на підставі своєї внутрішньої логіки. Під час програмування дані і

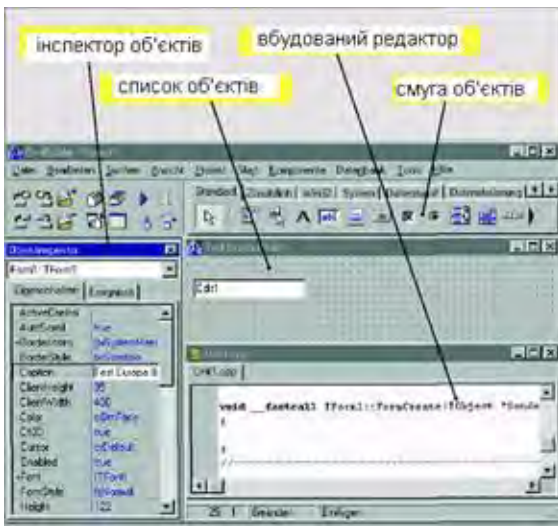


Рис.30.8. Оточення середовища C-BUILDER

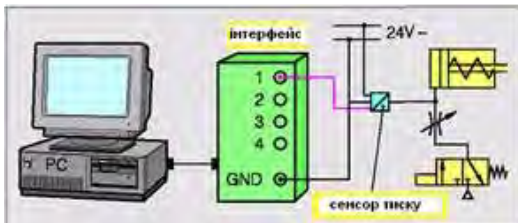


Рис.30.9. Схема з'єднань установки

Приклад функції опитування даних вимірювання

Таблиця 30.1

пакет програмування	функція
C-BUILDER (простий інтерфейс)	№ каналу DRUCK = ADCWert (1)
LabView (простий інтерфейс)	№ каналу 1 AD-12bit тиск K _{3.0} -10 0V1
LabView (інтерфейс типу Plug&Play)	№ пристрою № каналу максимальне значення 0 10,00 тиск AI ONE RT 0V1

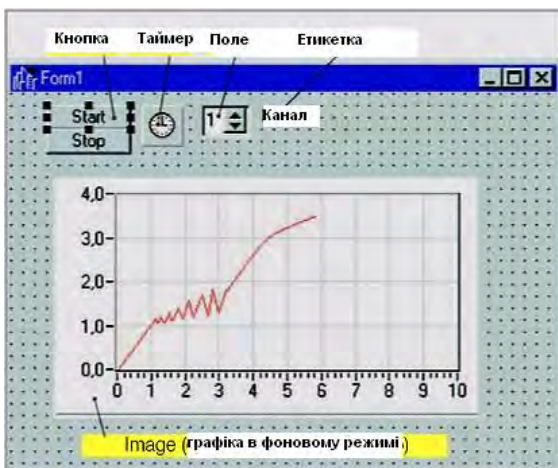


Рис.30.10. Графічна оболонка програми

редактора, наприклад, пакети DELPHI, C-BUILDER, VISUAL C, VISUAL BASIC. У так званих повних процедурах, об'єкти пов'язують графічно. Здійснюється це графічним формуванням об'єднань між окремими блоками. Це має місце в таких пакетах програмного забезпечення, як: LabView чи DasyLab.

Пакет програмного забезпечення C-BUILDER

Після створення вікна, відбувається розміщення окремих графічних об'єктів і пристосування їх властивостей до особливостей передбачених для них завдань. Можливим є звертання до існуючих об'єктів через відповідні смуги меню. До кожного об'єкту приписаний так званий інспектор об'єкту, який полегшує програмістові визначення його особливостей, а також приписування йому відповідних випадків. На рис. 30.8 показано скидання екрану середовища компілятора.

Приклади програми зчитування вимірювальних даних:

Нехай потрібно показувати на графіку зміну тиску в процесі руху поршня рушії однобiчної дії. Рис. 30.9 представляє схему з'єднань установки. Сенсор тиску подає аналоговий сигнал напруги 0...10 V при зміні тиску в межах 1...10 бар. Значення напруги перетворюється на 12-бітовий цифровий сигнал (0 - 4096). За допомогою готових функцій, напрацьованих виробником, можна, за допомогою цього 12-бітового числа, прямо зі значення напруги перейти до значення тиску. У табл.30.1 представлено приклад опитування цієї вимірюваної величини.

Програмування в середовищі C-BUILDER

Спочатку встановлюються особливості окремих об'єктів (рис.30.10), наприклад, для об'єкту „Таймер” встановлюється час циклу на 10 мс, для об'єкту „Image” (графіка діаграми) - вигляд фону і шкала. Кожен з об'єктів виробляє відповідний фрагмент первинного коду, який надалі програміст доповнює відповідними процедурами програми. Об'єкт ви-кликається за допомогою наданого коду.

Програмування в середовищі LabView

Після введення в дію середовища відкриваються два вікна.

Будова шару споживача

У шарі споживача, який називають панеллю, розташовані об'єкти (рис. 30.11). Об'єкт 1 є двійковою (логічною) змінною. Після його залучення, на діаграмі появиться (первинний код) символ TF. Разом з об'єктом 2, повинен встановитися номер каналу зчитування. Тому той об'єкт мусить бути числового типу. Початковим настроюванням був обраний тип цілих чисел. У первинному коді з'явиться голубий символ з етикеткою „U16”, що означає 16-бітове ціле додатне число. Об'єкт 3 є об'єктом виходу – тут застосовано часову діаграму з можливістю зміни її особливостей, наприклад, ко-

льору, масштабів. Максимальним значенням для осі У буде 5. Для осі Х значення масштабу мають бути узгодженим з часом циклу вимірювання. Для часового циклу 10 мс ділянка $dx = 0,01$. На *рис.30.12* представлено вікно, яке служить для встановлення цих властивостей (параметрів).

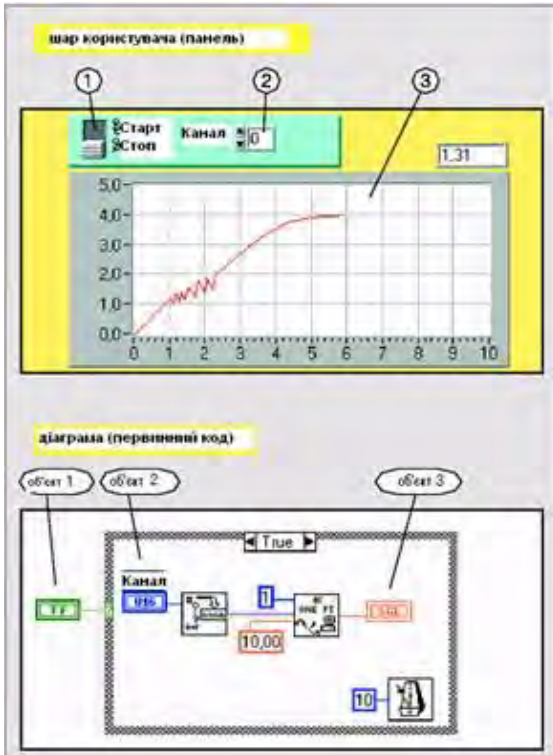


Рис.30.11.Графічне програмування в середовищі LabView

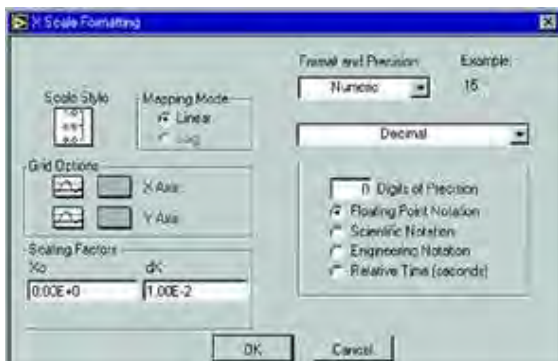


Рис.30.12. Встановлення об'єкту – часової діаграми (графіку)

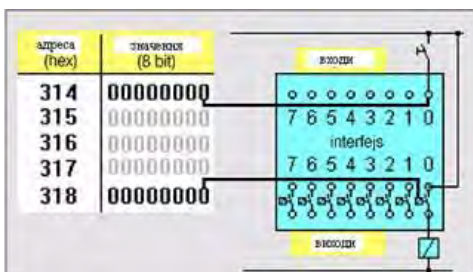


Рис.30.13. Адресування входів і виходів

перетворюють цілі значення 8 бітів за даною адресою – значення з діапазону від 0 до 255 і визначають їх як процесуальне відображення входу або виходу. Для дослідження стану окремих сигнальних входів потрібно застосовувати відповідну техніку, яку називають маскуванням. У випадку застосу-

вання *рис.30.12* представлено вікно, яке служить для встановлення цих властивостей (параметрів).

Формування первинного коду

Кожен об'єкт шару споживача викликає появу символу на стороні з первинним кодом. Об'єкт 1 може приймати два стани. До символу цього об'єкту додається структура „True”, в якій вписуються графічні символи всіх інших об'єктів. З бібліотеки функцій буде отримана функція опитування вимірюваного значення і обслуговування інтерфейсу. До цих функцій долучається номер пристрою, а потім і номер каналу. Тому, що обрана функція очікує ряду даних, числа повинні бути представлені рядом (послідовністю) даних. Далі записується максимальне значення - в даному випадку 10. Результат дій функції опитування значення є зараз зв'язаний з символом часової діаграми. Тому, що програма повинна працювати з визначеним часом циклу, потрібно отримати з бібліотеки функцій функцію часового циклу і встановити властиву сталу (10). Це означає, що кожні 10 мс розпочнеться черговий цикл програми.

Введення в дію програми

У головному меню програми LabView розташовані кнопки, які служать для обслуговування програмування. Після перемикавання об'єкту 1 програма вводиться в дію і виконує наступні завдання:

1. запитання, чи Кнопка 1 - „True”, якщо „False” - стрибок до 1,
2. якщо „True” - зчитати канал, перетворити числа в послідовність, зчитати номер пристрою, зчитати максимальне значення. Функція зчитує з периферійного пристрою аналогове значення, пересилає його до рисунка діаграми, де буде представлятися у вигляді лінії. Генератор імпульсів чекає 10 мс,
3. стрибок до 1.

Програма закінчиться в моменті клацання (натискання) на позначці „Стоп”.

Передавання даних

Цифрові сигнальні порти працюють, як правило, від стандартної постійної напруги 24 В. На *рис. 30.13* представлено принцип адресації сигналів. Вільний адресний простір в комп'ютері використовується для обслуговування сигнальних входів і виходів. У наведеному прикладі вхідною адресою є *314Hex*, вихідною – *318Hex*. Адреси вхідних портів розміщені в інтерфейсі. Якщо карта інтерфейсу є типу *Plug&Play*, то адреси визначає операційна система.

Кожна мова програмування має функції, які дозволяють читати за визначеною адресою та записувати під визначеною адресою. Ці функції завжди використовують та

вання цієї техніки процесуальне відображення пов'язується з прийнятим бінарним зразком (маскою) бажаного входу за допомогою функції кон'юнкції (I).

Приклад:

Треба визначити стан біта номер 1.

Номер біта = 76543210

Приклад відображення = 11, бінарний запис = 00001011

Маска для біта номер 1 (I) = 2, бінарний запис = 00000010

Актуальний стан біта = 1, значення 2.

В цілому, залежно від процесуального відображення, значення може бути 2 або 0.

Програмування в середовищі C-BUILDER.



Рис. 30.14. Читання і маскувння бітів – програм у середовищі C-BUILDER

На рис. 30.14 наведений приклад відображення процесу входу - з другого рядка програми (PAE) - інверсія тут необхідна з огляду на застосований спосіб формування вхідних сигналів. У третьому рядку значення висвічується в полі редагування процесового відображення входу (EditPAE). У рядках від 3 до 5 програми стан сигналів на входах E0, E1 і E2 визначається шляхом маскувння, а в рядках від 6 до 8 - виведений на моніторі (EditE0, EditE1, EditE2). Весь первинний текст програми знаходиться в об'єкті „Таймер”, налаштованому на цикл 10 мс. Це означає, що значення будуть поповнюватися кожні 10 мс.

Приклад: Секвенційне керування

На практиці у промисловості часто використовуються мови програмування високого рівня. У наведеному прикладі предмети висовуються послідовно за допомогою пневматичного рушія з випадного (гравітаційного) штабеля. Про стан пристрою для висування вказують сигналізатори у відповідному комунікативному полі пульта управління (рис. 30.15). Всі сигнальні входи і виходи пристрою висування з'єднані провідниками з цифровим інтерфейсом (рис. 30.16). У схемі системи керування розміщено також аварійний вимикач. Всі контактні вимикачі входів систем, окрім кнопкового вимикача „стоп” – є розімкненими контактами.

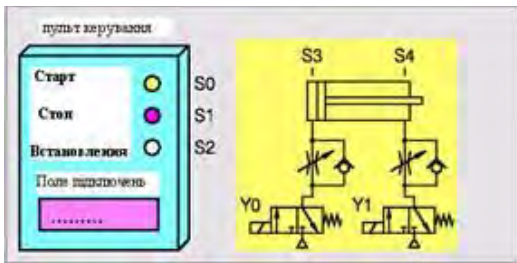


Рис.30.15. Пульт керування і схема пневматичної установки

Програмування в середовищі C-BUILDER здійснюється так само як робота програмованого контролера (PLC). Цикл програмованого керування реалізується тут об'єктом „Таймер”, настроєним на час, наприклад, 10 мс. Це означає, що Windows буде отримувати кожні 10 мс сигнал переривання, який викликає введення в дію функції Таймер 1. Змінні S, а також E0 ... E7 були задекларовані як глобальні змінні, бо як змінні локальні вони наново мали би бути ініціалізовані при кожному виклику функції, і через це могли би виникати неправильні результати роботи. Програма складається з трьох частин. У 1-й частині була задекларована структура процесу, в 2-й частині окремим кроком були призначені відповідні виходи; 3-тя частина встановлює процес формування текстової інформації і вивід її на монітор (рис. 30.17).

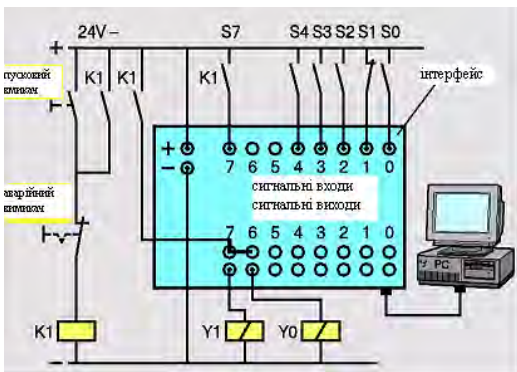


Рис.30.16. Схема електричних з'єднань пульта

Після написання програма компілюється до вигляду *.exe і в цій версії виводиться в дію безпосередньо на комп'ютері.

Об'єкт „Timer” встановлений час циклу 10 мс

декларування локальних змінних

зчитування відображення процесу виходів (PAA)

маскування виходів (E0...E7)

визначення вихідної позиції рушій (поршень втягнений) (GS)

визначення відображення процесу виходів (PAA)

встановлення змісту, призначеного для виведення

```

void __fastcall TForm1::Timer1Timer(TObject *Sender)
( int PAA,PAA;
  char Meldung[50];
  PAA=255-inportb(0x314); // зчитування відображення процесу
  E0=PAA&1; E1=PAA&2; E2=PAA&4; E3=PAA&8; E4=PAA&16; E7=PAA&128;
  // Частина 1: Перетворення
  GS=E3 && !E4; // секвенційна реалізація
  if( GS && !E0 && (S==0) ) S=1; // крок 1
  if( !E4 && !E3 && (S==1) ) S=2; // крок 2
  if( GS && (S==2) ) S=0;
  if( !E1 ) S=0; // умова вимкнення
  if( !E7 ) S=0; // умова вимкнення
  //Частина 2: Керування
  if(S==0) PAA=0;
  if(S==1) PAA=64; // висування поршня
  if(S==2) PAA=128; // втягнення поршня
  if( E2 && E1 && !S ) PAA=128; // урухоплення
  outportb(0x318,PAA); // відсилення відображення процесу
  //Частина 3: Виспання
  if ( E3 && E4 ) strcpy(Повідомлення, "Увага, аварія сенсора");
  if ( !E3 && !E4 && !S) strcpy(Повідомлення, "Поршень не висунутий");
  if ( S==1 ) strcpy(Повідомлення, "Поршень висувається");
  if ( S==2 ) strcpy(Повідомлення, "Поршень втягується.");
  if ( !E7 ) strcpy(Повідомлення, "Аварійна кнопка");
  Meldatext->Text=
  }
  //
  
```

Рис. 30.17. Організація секвенційного керування в середовищі C-BUILDER.

Інтерфейс DDE

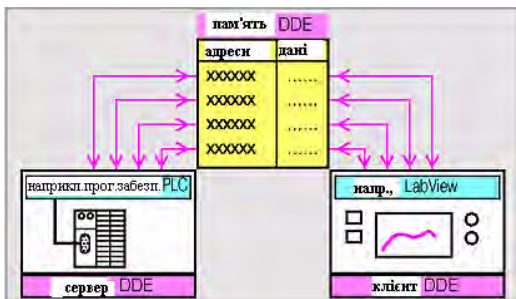


Рис.30.18. Основи інтерфейсу DDE

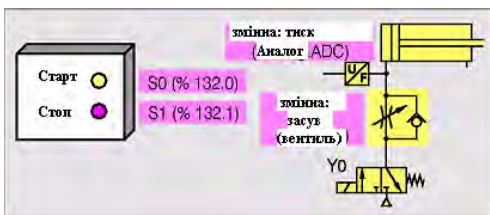


Рис.30.19. Приклад керування PLC

Інтерфейс DDE (англ. *Dynamic Data Exchange*) дозволяє здійснювати динамічне перетворення даних. У промисловості він часто використовується для моніторингу і візуалізації процесів. Інтерфейс DDE складається з сервера і клієнта DDE. Споживач - об'єкт, який представляє дані, - називається сервером, а той, який отримує дані - клієнтом. Сервер в безперервному режимі записує всі дані до пам'яті. Клієнт, також в безперервному режимі, забирає дані з пам'яті сервера DDE. Обмін даними триває, залеж-

но від комп'ютера, до 10 мс. На рис.30.18 представлено приклад, в якому програмне забезпечення контролера PLC, - окрім інших завдань - записує змінні і інформацію про стан процесу в пам'яті DDE, натомість програма LabView отримує дані з пам'яті DDE і на їх підставі здійснює візуалізацію процесу. Існує також можливість пересилання даних від клієнта через пам'ять DDE до сервера.

Під час програмування клієнта потрібно вказа-

ти три дані:

- DDE-Service: назва сервера, напр., PROSYS, EXCEL,
 - DDE-Topic: назва файлу сервера, напр., C:\xxxx\Visual.PRO.
 - DDE-Item: назва змінної, через яку буде відбуватися зв'язок, напр., %IB32).
- До змінної „Text” вписується значення змінної.

Приклад: Програмне забезпечення програмованого контролера (PLC) як сервера DDE у співпраці з програмою LabView

Змінні були представлені в вигляді позначень на рис. 30.19. Стан кнопок Старт і Стоп розпізнається прямо зі змінної опису стану процесу (байт вхідних даних %132).

На стороні панелі представлені об'єкти, доступні для оператора, а на стороні діаграми - функція „Open”, яка активує канал DDE (рис. 30.20). Функція „Start” вводить в дію зв'язок - дані доставляються через „Items”. Завдяки функції „Check”, дані, перетворені до вигляду послідовності знаків, чекають на подальшу обробку.

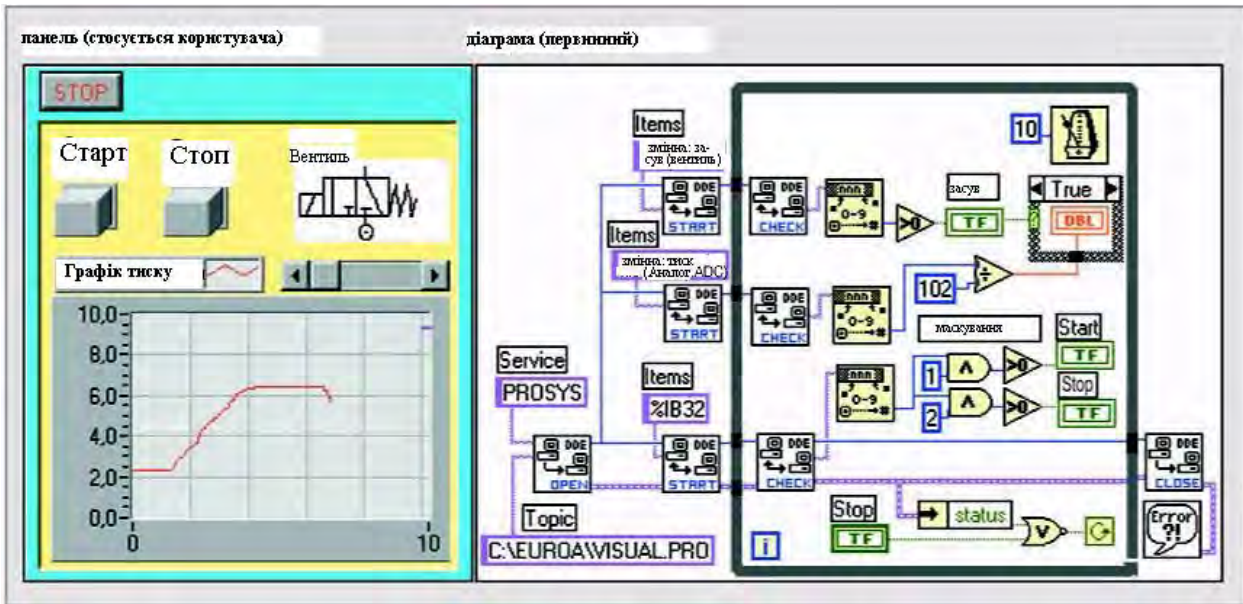


Рис. 30.20. Зв'язок DDE з LabView

Складові частини програмного забезпечення

Домінуючою тенденцією у виробництві програмного забезпечення є розвиток складових частин (модулів) програмного забезпечення. Завдяки цьому, конкретні рішення можна скласти оптимальним чином з різного виду складових частин і з різних бібліотек. Технологія ActiveX окреслює межі розвитку програмного забезпечення цього типу. COM і DCOM (англ. *Component Object Model* та *Distributed COM*) становлять середовище для об'єктного управління операційними системами для ActiveX. Об'єкт типу ActiveX є окремою складовою частиною програмного забезпечення, яка мусить мати місце в конкретному застосуванні. Таке застосування називається носієм ActiveX (англ. *ActiveX-Container*). Технологія ActiveX є основою нового стандарту - OPC (англ. *Object Linking for Process Control*).

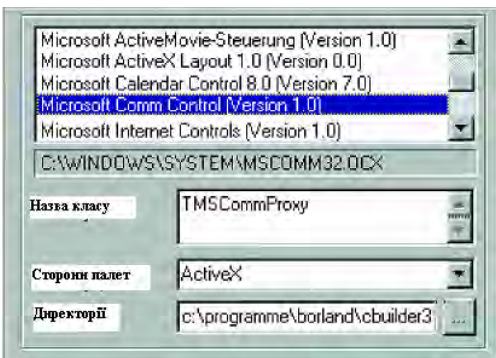


Рис.30.21.Імплементация ActiveX

Приклад: Зчитування даних із зчитувача ризику за допомогою порту RS232C

На екрані повинна з'явитися назва статті. Тому що C-BUILDER не має жодних засобів для обслуговування порту RS232C, треба використати модуль ActiveX-Control, завданням якого є зчитування з порту.

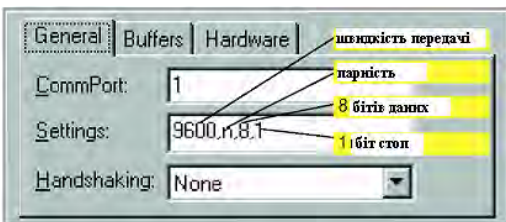


Рис.30.22.Встановлення особливостей

1. Модуль ActiveX-Control TMSCommProxy додається до програмного забезпечення через складову меню „Частини” (рис. 30.21). Цей модуль з'явиться як іконка в меню вибору для ActiveX-Control.

2. При опрацюванні застосування завдання, для нього копіюється модуль ActiveX-Control. Наступним кроком мусять бути встановлені особливості, наприклад, номер порту, налаштування отримуваних даних, підтвердження тощо. Це дозволяє

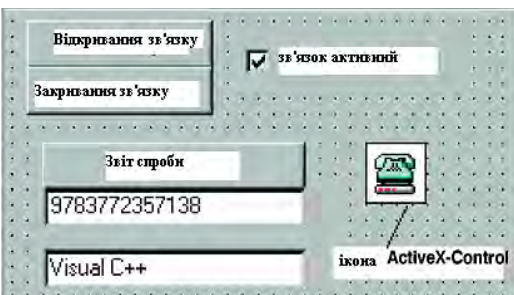


Рис.30.23. Вікно користувача

зробити меню „Особливості" (рис. 30.22).
3. На рис. 30.23 і 30.24 показана повна будова прикладу для застосування.

```

//первинний код для кнопки "Відкривання з'єднання"
{ MSCommProxy1->PortOpen=True; //Відкривання з'єднання з порту COM
  CheckBox1->Checked=True; } //Виділення CheckBox1

//первинний код для кнопки "Закривання з'єднання"
{ MSCommProxy1->PortOpen=False; // Закривання з'єднання з порту COM
  CheckBox1->Checked=False; } //Відміна виділення CheckBox1

//первинний код для кнопки "Зчитування спроби"
{ if (CheckBox1->Checked) //Коли порт COM відкритий
  { Daten->Text=MSCommProxy1->Input; //Пробне зчитування послідовного порту
    //Висування даних в поля даних, порівняння даних з відомими, виведення тексту
    Ware->Text=" ----- ";
    if (Daten->Text=="4001057905368") Ware->Text=" TURBO C-Buch ";
    if (Daten->Text=="9783772357138") Ware->Text=" Visual C++ ";
  }
}

```

Рис. 30.24. Первинний код для зчитування даних з послідовного порту

Питання для повторення і засвоєння

1. Опишіть найважливіші порти комп'ютера.
2. Поясніть процес *Multitasking*.
3. Назвіть елементи інтерфейсу споживача в операційній системі Windows.
4. Що можна розмістити у вільних місцях приєднань комп'ютера PC?
5. Що треба розуміти під процесом об'єктно орієнтованого програмування?
6. Назвіть можливі процедури програми.
7. Які переваги має програмування на основі графічних процедур?
8. Поясніть маскування одиничних бітів в байті.

Василь Іванович Ткачук

ОСНОВИ МЕХАТРОНІКИ

3-тє видання, стереотипне

Конспект лекцій

Формат 60x84/8. Папір друк. №2. Гарнітура Times New Roman

ПП «Магнолія 2006»

м. Львів-53, 79053, Україна, тел.: +380503701957

e-mail: magnol06@ukr.net

Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру видавців, виготівників і розповсюджувачів видавничої
продукції: серія ДК № 2534 від 21.06.2006 року,
видане Державним комітетом інформаційної політики,
телебачення та радіомовлення України

Надруковано у друкарні видавництва «Магнолія 2006»