

**АВТОМАТИЗАЦІЯ ЗБОРУ ДАНИХ З ЕЛЕКТРОМЕХАНІЧНИХ
КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ДЛЯ ОФІСНИХ ДОДАТКІВ MICROSOFT**

Сучасні системи збору даних для диспетчеризації технологічних процесів або займають достатньо багато часу для розробки (при використанні безкоштовного програмного забезпечення), або ж достатньо дорогі (при використанні готових систем SCADA). Тому актуальним є використання офісних додатків Microsoft для реалізації даної задачі. В роботі запропоновано принцип побудови систем збору даних для їх впровадження в офісних додатках Microsoft, що дозволяє здійснювати аналіз даних в реальному часі, їх статистичну обробку без спеціалізованого програмного забезпечення.

Ключові слова: система збору даних, офісний додаток, обмін даними, реальний час.

S.M. LEVITSKIY, M.P. ROZVODIUK
Vinnytsia National Technical University

**THE AUTOMATION OF DATA ACQUISITION FROM
ELECTROMECHANICAL COMPUTERIZE SYSTEMS FOR MICROSOFT OFFICE APPLICATIONS**

Abstract – Modern data acquisition system for dispatching processes take a lot of time for develop (using free legacy software) or quite expensive (when using ready-made SCADA). So important thing is using the Microsoft Office applications for the implementation of this task. The paper presents the principle of building data acquisition systems for implementation in Microsoft Office applications, allowing to analyze data in real time, their statistical processing without special software.

Keywords: data acquisition system, desktop application, data exchange, real time clock.

Вступ

Сучасні системи збору даних представлені як системи диспетчеризації, розробка яких має досить громіздкий та тривалий процес при використанні безкоштовного програмного забезпечення (legacy software development), або є достатньо капіталомістким продуктом при використанні готових систем SCADA [1-3]. Більшість з комунальних систем збору даних призначено для нагляду за станом технологічного обладнання індивідуальних теплових пунктів та котельні, засобів електроенергообліку і дозволяють: забезпечити служби підприємства актуальними й достовірними даними про роботу обладнання; забезпечити оперативно-технологічний і управлінський персонал підприємства інформацією про хід технологічного процесу і поточний стан обладнання; контролювати вихід інструментальних і технологічних параметрів за граничні аварійні значення; створити передумови для організації єдиної системи WEB-доступу до технологічних параметрів та стану роботи обладнання. Практично на кожному персональному комп'ютері чи ноутбуку використовується офісні додатки Microsoft. Тому доцільно було б використати їх можливості для реалізації даних задач, не витрачаючи додаткових зусиль та коштів на розробку нових чи закупку готових систем SCADA.

Мета дослідження. Метою роботи є розробка принципу побудови систем збору даних в реальному часі для стандартних офісних додатків персональних комп'ютерів.

Матеріал і результати дослідження

Принцип побудови системи збору даних будемо реалізовувати на прикладі мережевого керування перетворювачем частоти (ПЧ) Altivar 71 фірми Schneider Electric [4].

Для управління ПЧ Altivar 71 в мережі Ethernet використовується комунікаційна карта VW3A3310. Конфігурація комунікаційної карти здійснюється через меню ПЧ, як наведено на рис. 1 [5].

Для моніторингу та управління ПЧ необхідно здійснювати читання / запис наступних реєстрів (табл. 1).

Вказана таблиця реєстрів пристрою є зразком, а при узагальненому підході до побудови систем збору даних пам'ять пристроїв представляється як масив реєстрів, де зберігаються дані як оперативні (наприклад, струм, потужності, частота в електричних лічильниках або температура та витрати в теплових лічильниках), так і накопичувальні (спожита електрична або теплова енергія, спожиті об'єми води, газу, теплоносія тощо). Крім того виробники в реєстрах зазначають ідентифікаційні дані пристроїв (серійний номер, дату випуску, номер партії тощо), які за необхідності також включають до систем збору даних.

В запропонованому підході надання даних до офісного додатку має здійснюватися проміжною програмою-сервером, яка надаватиме дані головній програмі (хост-програмі). Основна задача серверу полягає в отриманні даних з пристрою(ів) за допомогою вбудованого драйвера Ethernet TCP/IP протоколу та

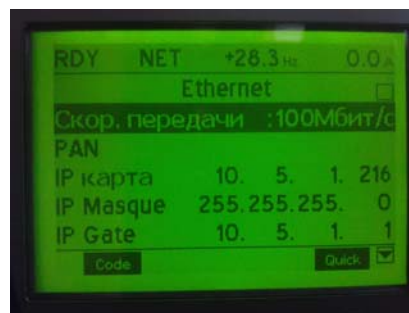


Рис. 1. Налаштування комунікаційної карти ПЧ

Інформаційно-вимірювальні та обчислювальні системи і комплекси в технологічних процесах
 діагностуванні зв'язку з пристроєм(ями), такі програмні сервери мають назву OPC-серверів (OLE for Process Control). Задача хост-програми – відображення, архівування даних, формування команд пристроям відповідно до дій оператора.

Таблиця 1

Список регістрів ПЧ для моніторингу та управління

Регістр (код)	Адреса (dec)	Адреса (hex)	Функція
Команда (CMDD)	8601	0x2199	запис / читання
Завдання (LFRD)	8602	0x219A	запис / читання
Статус (ETAD)	8603	0x219B	читання
Швидкість (RFRD)	8604	0x219C	читання (об/хв)
Напруга (UOP)	3201	0x0C81	читання (В)
Струм (LCR)	3204	0x0C84	читання (0,1 А)

В якості сервера в прикладі використано OPC-сервер MasterOPC Universal Modbus Server [5]. Послідовність налаштування сервера наступна: створення комунікаційного вузла з вказаною IP-адресою та портом; створення пристрою з вибраним протоколом обміну даними; створення групи тегів для пристрою з назвою-ідентифікатором; створення списку тегів відповідно до адрес регістрів. Вказана послідовність наведена на рис. 2-4.

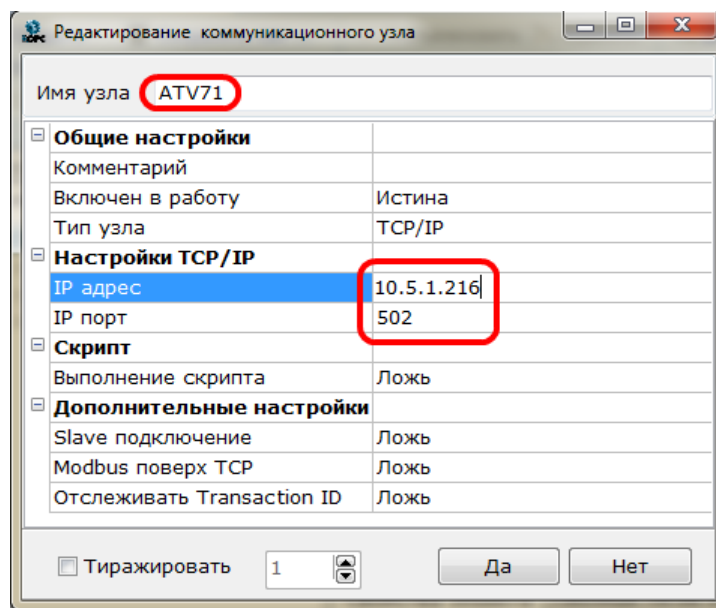


Рис. 2. Налаштування комунікаційного вузла OPC-сервера

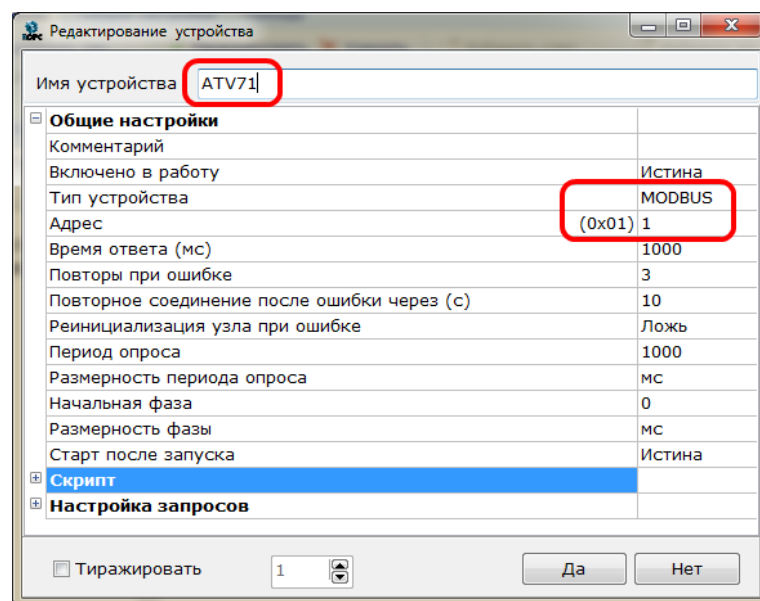


Рис. 3. Налаштування пристрою – джерела даних OPC сервера

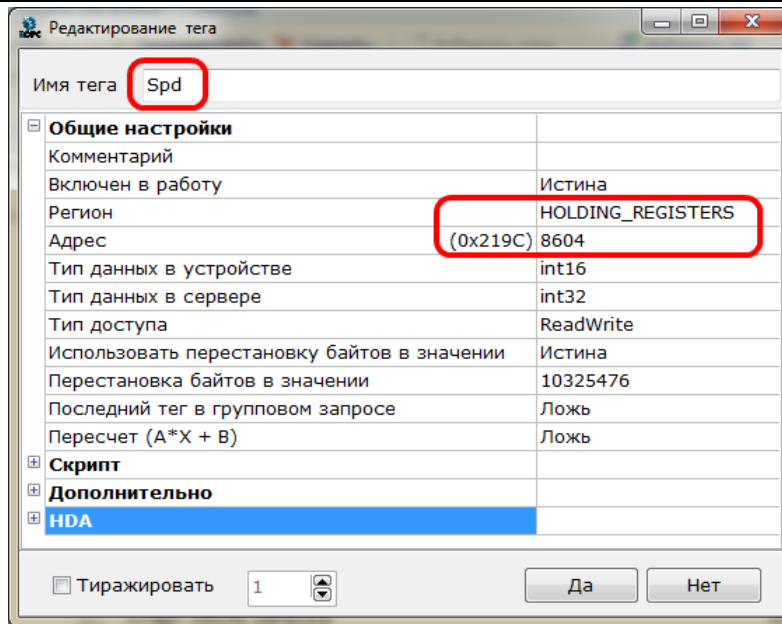


Рис. 4. Налаштування тегів OPC сервера

Після проведених налаштувань відповідно до табл. 1 конфігурація сервера зберігається і його можна запустити для тестування зв'язку з ПЧ. Запущений сервер в роботі при відсутності помилок зв'язку відповідає вигляду вікна на рис. 5.

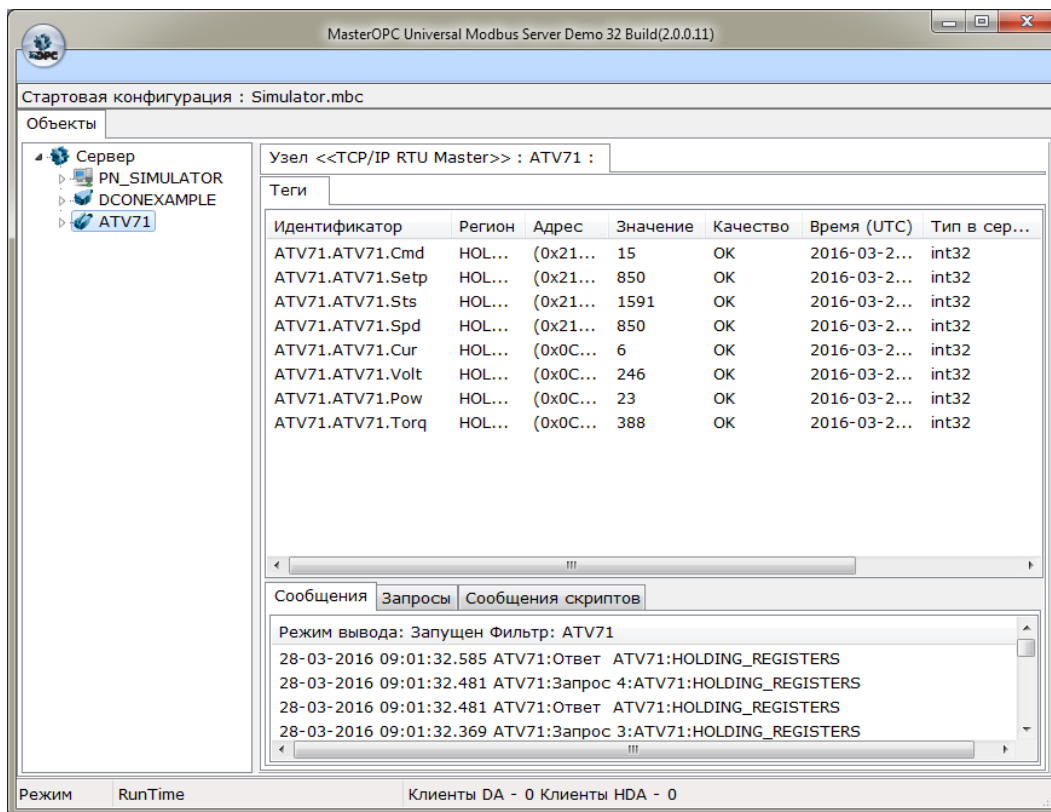


Рис. 5. OPC сервер в режимі роботи

В якості хост-програми використаємо додаток MS Excel. Отримання даних з OPC в реальному часі в додатку відбувається шляхом періодичного виклику макросу зв'язку з OPC-сервером. Структура макросу відповідає запропонованому алгоритму на рис. 6.

Перед розробкою коду програми створюється файл в офісному додатку, в якому і будуть зберігатись зібрані з допомогою OPC-сервера дані. Для запропонованого прикладу з ПЧ зразок конфігурації файлу наведено на рис. 7. В додатку використано елементи Button та ScrollBar з форм керування.

Відповідно до розробленого алгоритму, в вбудованому редакторі Visual Basic for Application розробляється код основної програми. Перед розробкою слід підключити до проекту бібліотеку обміну даними з OPC під назвою OPCDAAuto.dll (OPC Data Access dynamic library) за зразком на рис. 8.

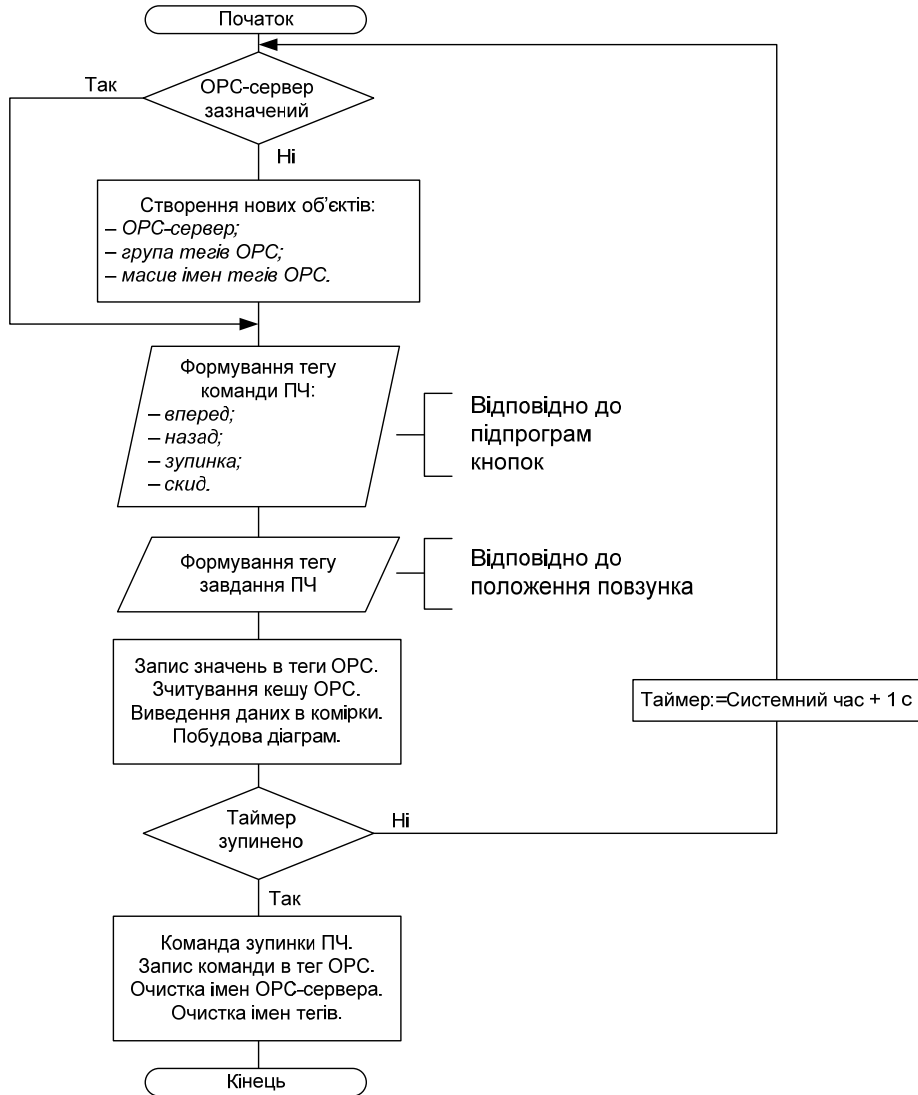


Рис. 6. Алгоритм макросу хост-програми обміну даними з OPC-сервером

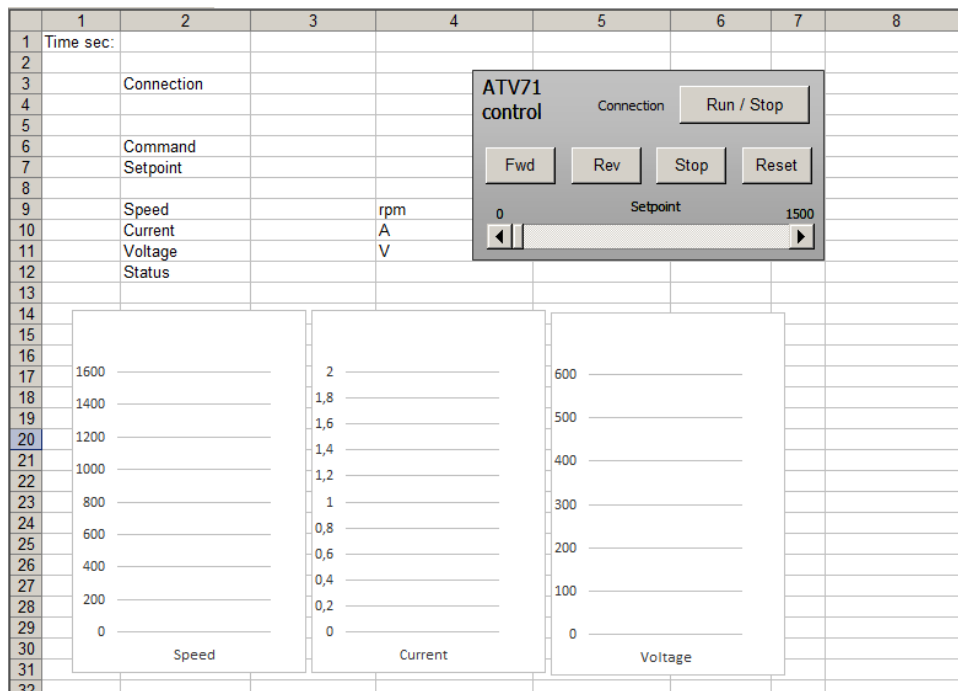


Рис. 7. Файл додатку системи збору даних на прикладі ПЧ

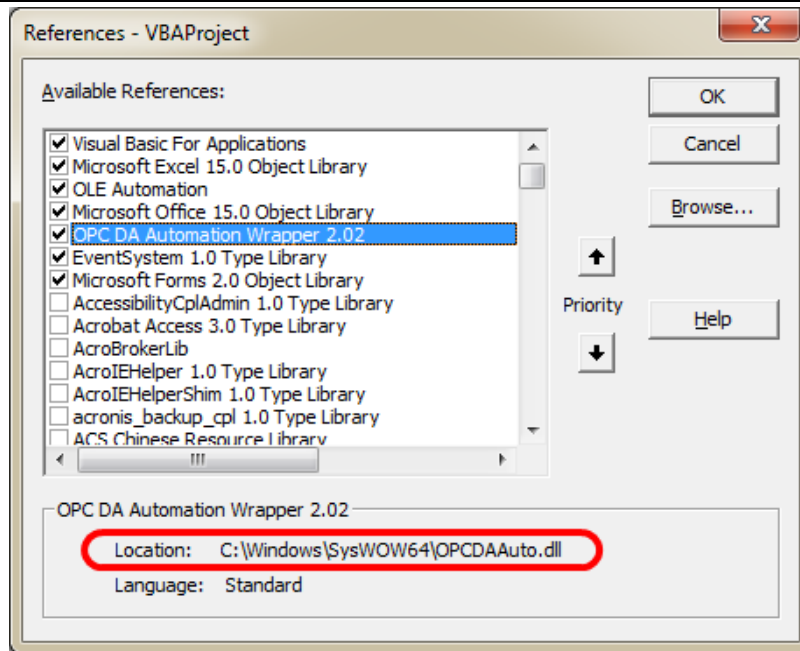


Рис. 8. Включення динамічних бібліотек в макрос додатку

Якщо вказані бібліотеки вже додано до списку, то наступним етапом являється розробка коду програми (названа Connect(), яка працює в фоновому режимі при відкритому додатку). Код програми наведено нижче.

```

Public Server As OPCServer           ' OPC-сервер (OLE for Process Control)
Public Group As OPCGroup
Public StopConnect As Boolean        ' Старт / Стоп з'єднання
Public tagname(7) As String, OPCTags(7) As OPCItem ' імена тегів OPC
Public Value, Quality, TimeStamp As Variant ' параметри тегів OPC
Public i, j, TimeCount As Integer
Sub Connect()                        ' основна програма отримання даних
If Server Is Nothing Then
Set Server = New OPCServer           ' установка нового OPC-сервера
Server.Connect "InSAT.ModbusOPCServer.DA" ' ім'я нового OPC-сервера
Set Group = Server.OPCGroups.Add("ATV71") ' ім'я комунікаційного вузла OPC-сервера
For i = 1 To 6
tagname(i) = "ATV71.ATV71."         ' імена тегів "ім.вузла.ім.пристрою.ім.тега"

tagname(1) = tagname(1) & "Cmd"
tagname(2) = tagname(2) & "Setp"
tagname(3) = tagname(3) & "Spd"
tagname(4) = tagname(4) & "Cur"
tagname(5) = tagname(5) & "Volt"
tagname(6) = tagname(6) & "Sts"

Group.OPCItems.AddItem tagname(i), i ' додавання імен в список тегів OPC
Set OPCTags(i) = Group.OPCItems.Item(i) ' створення масиву тегів OPC
Next i
End If
If StopConnect = True Then          ' дія при роз'єднанні з OPC-сервером
OPCTags(1).Write Лист1.Cells(6, 3).Value ' команда зупинки ПЧ
Server.Disconnect                   ' відключення клієнта від сервера
Set Server = Nothing                ' очистка пам'яті з'єднання
Set Group = Nothing
Set OPCTags(7) = Nothing
Лист1.Cells(3, 3).Value = "OFF"     ' відображення стану з'єднання
Exit Sub                             ' вихід з основної програми
End If
For i = 1 To 6
OPCTags(i).Read OPCache, Value      ' зчитування кешу даних OPC
Next i

```

```

' ===== Write =====
OPCtags(1).Write Лист1.Cells(6, 3).Value      ' Запис команди ПЧ в OPC
OPCtags(2).Write Лист1.Cells(7, 3).Value      ' Запис завдання ПЧ в OPC
' ===== Read =====
Лист1.Cells(9, 3).Value = OPCtags(3).Value    ' Зчитування поточної швидкості (об/хв)
Лист1.Cells(10, 3).Value = OPCtags(4).Value / 10 ' Зчитування поточного струму (0,1 А)
Лист1.Cells(11, 3).Value = OPCtags(5).Value   ' Зчитування вихідної напруги (В)
Лист1.Cells(12, 3).Value = OPCtags(6).Value   ' Слово статусу ПЧ
DoEvents
Application.OnTime Now() + TimeSerial(0, 0, 1), "Connect" ' повт. виклик програми через 1 с
TimeCount = TimeCount + 1
  If TimeCount > 60 Then
    TimeCount = 0
  End If
Лист1.Cells(1, 2).Value = TimeCount           ' відображення роботи таймеру
Лист1.Cells(3, 3).Value = "ON"               ' відображення стану з'єднання
End Sub

```

Додаткові сценарії, які виконуються в додатку, призначені для управління зв'язком або формування команди оператора і призначаються для відповідних елементів керування – кнопок. Сценарій кнопки, яка призначена для запуску/зупинки обміну даними наведено нижче.

```

' ===== RUN/STOP Button =====
Sub Control_Connect()
j = j + 1
  If j > 2 Then
    j = 1
  End If
  Select Case j
Case 1
  Лист1.Cells(6, 3).Value = 7 ' Stop command
  StopConnect = True
Case 2
  StopConnect = False
  Application.Run ("Connect")
End Select
End Sub

```

Сценарій кнопки для пуску ПЧ в напрямку «Вперед» вимагає формування команди за значенням 15, що має бути записана у відповідний тег OPC. Однак для уникнення самовільного пуску ПЧ в тому випадку, коли оператор натиснув кнопку «Вперед», а обмін даними не був запущений, системою формується попереджувальне повідомлення і автоматичний скид команди.

```

Sub Fwd()
If StopConnect = False Then
  Лист1.Cells(6, 3).Value = 15
Else
  MsgBox "Start connection before", vbExclamation & vbOKOnly, "No
Connection"
End If
End Sub

```

За наведеними зразками розробляються сценарії для кнопок пуску ПЧ в напрямку «Назад» (команда = 2063), зупинка (команда = 7) та скид (команда = 6). Зразок прив'язки відповідного сценарію до кнопки наведено на рис. 9.

Зміна завдання формується через положення повзунка елементу керування ScrollBar. В його параметрах налаштування слід прив'язати положення до комірки Excel, що виділена під введення завдання швидкості ПЧ.

Після проведених налаштувань файл системи збору даних зберігається і після натиснення кнопки з прив'язаним макросом Control_Connect() отримуємо дані з ПЧ в додатку за зразком на рис. 10. Діаграми можна додати та налаштувати за вимогами оператора. Зразок додатку в роботі наведено на рис. 11, а відповідний стан ПЧ на рис. 12.

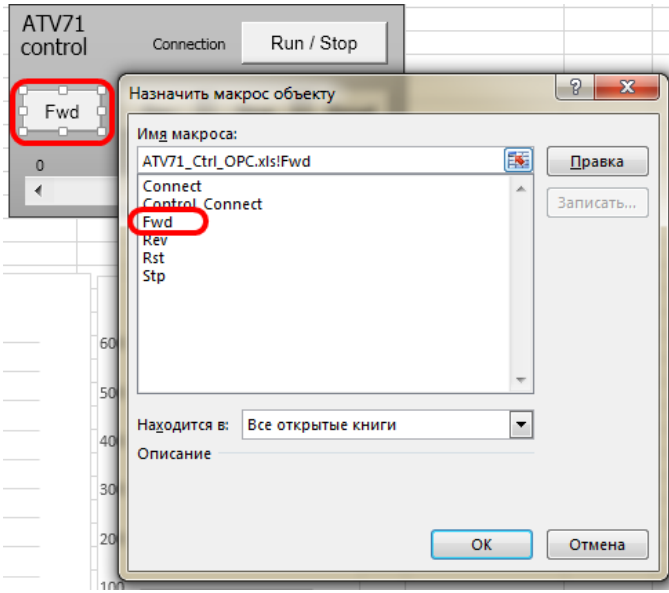


Рис. 9. Прив'язка елементів керування додатку до розроблених сценаріїв

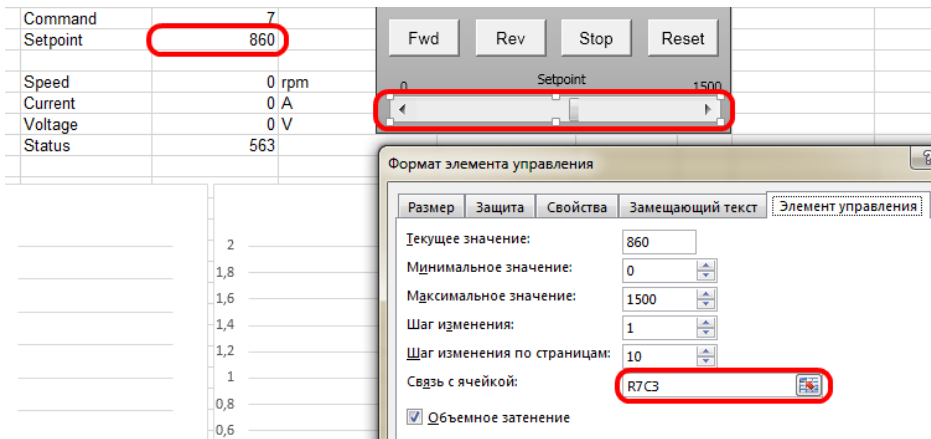


Рис. 10. Прив'язка елементу ScrollBar до комірки завдання швидкості ПЧ

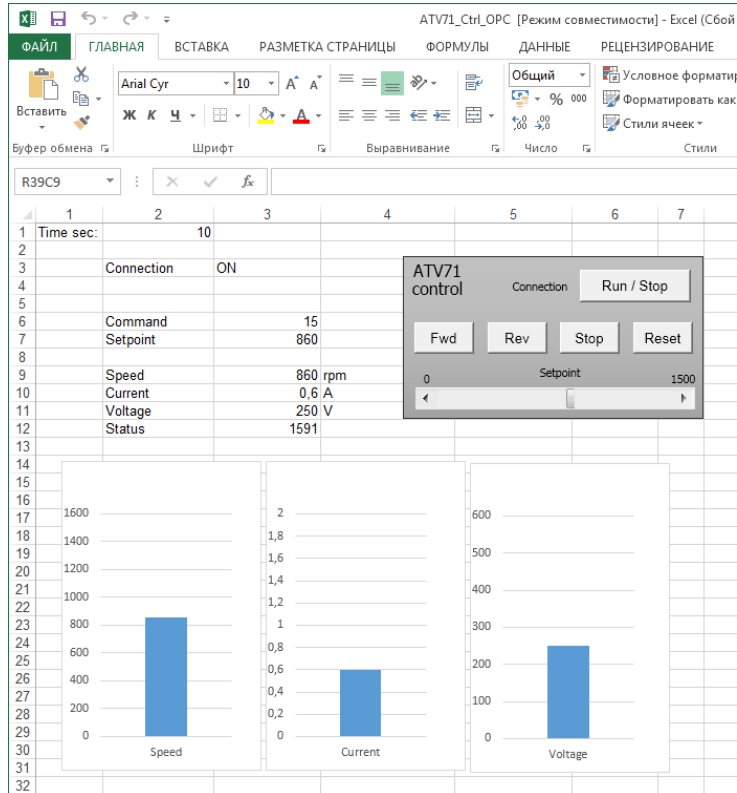


Рис. 11. Система збору даних в роботі

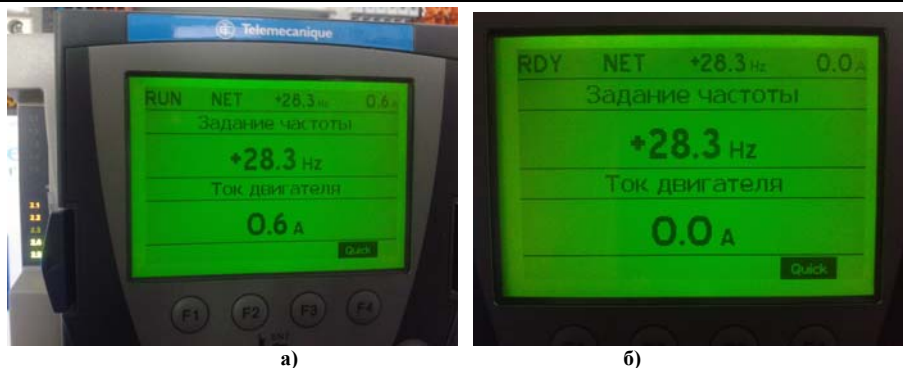


Рис. 12. Стан ПЧ при керуванні від додатку (а – запущено вперед, б – зупинка)

Висновки

Розроблено принципи побудови систем збору даних для їх впровадження в офісних додатках Microsoft, що дозволяють реалізувати аналіз даних в реальному часі, їх статистичну обробку без спеціалізованого програмного забезпечення.

Література

1. Системи диспетчеризації і система SCADA. – [Електронний ресурс]. – Режим доступу: <http://um.co.ua/11/11-7/11-70267.html>
2. ОБЛІК ТЕПЛА, ГАЗА, ЕЛЕКТРИКИ, ВОДИ -В ОДНІЙ ПРОГРАМІ. – [Електронний ресурс]. – Режим доступу: <http://www.navicount.com/ua/>
3. Книжник Є. Н. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВІДДАЛЕНОГО МОНИТОРИНГУ ТЕПЛОЕНЕРГЕТИЧНИХ ОБ'ЄКТІВ [Електронний ресурс] / Є.Н. Книжник, А.Л. Перекрест, А.В. Маслівець // Інженерні та освітні технології в електротехнічних і комп'ютерних системах. № 3/2013 (3). – Режим доступу: http://eetecs.kdu.edu.ua/2013_03/EETECSS2013_0302.pdf
4. Altivar 71. Преобразователи частоты для асинхронных электродвигателей. Руководство по программированию. – Schneider Electric, Telemecanique. – 12/2005. – 262 с.
5. Altivar 71. Modbus TCP/IP Ethernet card VW3A3310. User's manual. – Schneider Electric, Telemecanique. – atv71_Ethernet_EN_V1. – 05/2005. – 75 p.

References

1. Systemy dyspetcheryzatsii i systema SCADA. – [Elektronnyj resurs]. – Rezhym dostupu: <http://um.co.ua/11/11-7/11-70267.html>
2. Oblik tepla, haza, elektryky, vody –V Odnij prohrami. – [Elektronnyj resurs]. – Rezhym dostupu: <http://www.navicount.com/ua/>
3. Knizhnik Ye.N. Informatsiine zabezpechennia dlia viddalenooho monitorynhu teploenerhetychnykh ob'ektiv [Elektronnyj resurs] / Ye.N. Knizhnik, A.L. Perekrast, A.V. Maslivets // Inzhenerni ta osvichni tekhnologii v elektrotekhnichnykh i kompiuternykh systemakh. – № 3/2013 (3). – Rezhym dostupu: http://eetecs.kdu.edu.ua/2013_03/EETECSS2013_0302.pdf
4. Altivar 71. Preobrazovateli chastoty dlia asynkhronnykh elektrodvyhatelej. Rukovodstvo po prohrammyrovanyiu. – Schneider Electric, Telemecanique. – 12/2005. – 262 s.
5. Altivar 71. Modbus TCP/IP Ethernet card VW3A3310. User's manual. – Schneider Electric, Telemecanique. – atv71_Ethernet_EN_V1. – 05/2005. – 75 p.

Рецензія/Peer review : 14.5.2016 р.

Надрукована/Printed : 30.6.2016 р.
Рецензент: д.т.н., проф., Кухарчук В.В.