

**Practical implications.** When analyzing the results of voting on the knowledge and skills that should have a particular specialist the methods of Data Mining allowed: to take into account not only the options of voting, in which a fairly large number of respondents vote for one or two specific competence, but whole blocks of competencies inherent to this profession; to determine interchangeable or related professions; to reveal the connection between the professions and the branches of science and production, where these professions are most in demand; to make recommendations for filling the educational training programs.

**Keywords:** *quality training, system analysis, employer's requirements analysis, Data Mining*

УДК 681.5:621.314.57

© С.М. Ткаченко

## **ДИНАМІЧНЕ НАЛАГОДЖЕННЯ ВИХОДІВ ТА ВХОДІВ ТИПУ «ТАК-НІ» У ТЕГОВІЙ ПАМ'ЯТІ ПРОМИСЛОВОГО КОНТРОЛЕРА**

© S. Tkachenko

## **DYNAMIC DEBUGGING OF THE OUTPUTS AND INPUTS OF THE "YES-NO" TYPE IN THE TAG MEMORY OF COMPTROLLERS**

Розроблено метод непрямой побітової адресації тегових областей пам'яті виходів та входів «Так-Ні» промислового контролера для динамічного налагодження і перепризначення каналів вводу і виводу сигналів технологічного устаткування без програмування.

Разработан метод непрямой побитовой адресации теговых областей памяти выходов и входив «Да-нет» промышленного контроллера для динамической отладки и переназначения каналов ввода и вывода сигналов технологического оборудования без программирования.

**Вступ.** Для багатьох підприємств України можливості використання промислових контролерів обмежені через високу вартість впровадження та експлуатації програмно-технічних засобів. За умови їх використання у порядку, передбаченому фірмою-розробником, кожний сигнал має бути заведений на окремий вхід чи вихід, а кожна інформаційна ознака, що передається або отримується від засобів відображення SCADA, повинна мати свій ліцензований тег, тобто, за перекладом з англійської, «ознаку». Вартість такої ліцензії, в залежності від потреби у тегах, знаходиться, приблизно, у межах від 300 до 4000 євро для різних виробників обладнання[1]. Ряд крупних виробників, таких, наприклад, як Siemens або Ge-Fanuc, для обміну дозволяють використовувати додаткові засоби, такі як масиви, строки та гранд-теги. Вони дозволяють в один ліцензований тег включити досить великі обсяги даних, порядку сотень або й тисяч байт, але тут універсального підходу не існує і робота зі складання та розкладання такої структури даних лягає на програміста.

У будь-якому випадку, неможливо на стадії проектування системи і написання програмного забезпечення передбачити всі оперативні зміни та особливості об'єкту автоматизації, тому доурахування всього переліку знову введених сигналів, а значить і кінцеве відлагодження програм контролерів проводиться безпосередньо під час пуско-налагоджувальних робіт на об'єкті впровадження, що потребує додаткових витрат. Більш того, будь-які зміни у переліку сигналів під час експлуатації, навіть тимчасові через вихід з ладу датчика, потребують дистанційної або безпосередньо на місці корекції програми і підприємство-Замовник знову отримає додаткові витрати. Проблема перш за все стосується сигналів «Так-Ні» [2].

З аналоговими та «цифровими» (як прийнято називати дискретні) сигналами проблема не настільки гостра. Таких сигналів набагато менше, вони, як правило, легко визначаються у програмному коді та досить важливі для керування технологічним процесом. Тому, коли такий датчик або його пристрій узгодження вийшов з ладу, без варіантів необхідна їх заміна або ремонт. Використати інший вільний вхід чи вихід комплексу технічних засобів навряд чи вдасться.

Таким чином, виникає актуальна задача розробки методів динамічного налагодження входів і виходів типу «Так-Ні» для забезпечення введення та виведення сигналів та узгодження їх зі структурами даних великого обсягу, що дозволить спростити задачу налагодження програмної частини системи керування під час пуско-налагодження та експлуатації системи. Спрощення має полягати у тому, що для підключення виходів і входів «Так-Ні» до комплексу технічних засобів системи керування не потрібно було б залучати програміста промислових контролерів.

**Мета досліджень.** Виходячи з аналізу існуючих підходів до складання алгоритмів керування технологічним обладнанням, які працюють лише з сигналами «Так-Ні», розробити метод динамічного перерозподілу інформаційних ознак сигналів між входами та виходами комплексу технічних засобів системи керування, яка використовує серійні промислові контролери.

**Основний матеріал досліджень.** Розглянемо загальні підходи до керування простими технологічними об'єктами, до яких можна віднести насоси, вентилятори, конвеєри, грохоти, сепаратори, нагрівачі з регулюванням по сигналу «Так-Ні» (наприклад, по термостату) і тому подібні установки. У будь-якому випадку, такий пристрій має одну інформаційну ознаку сигналу керування (включити-виключити) та множину інформаційних ознак стану об'єкта, з якими можуть бути, але не обов'язково, пов'язані вихідні сигнали «Так-Ні». Маємо деяку функцію

$$y = f(x_1; x_2; x_3; \dots x_n), \quad (1)$$

яка задає стан вихідного сигналу об'єкта з  $n$  входами. Наповнення функції (1) може бути різним [3]. Важливо, що для змінних  $x$  та  $y$  можуть, причому не-

обов'язково, бути виділені фізичні входи та вихід комплексу технічних засобів [3]. Для спрощення, далі не будемо розглядати керування по двом і більше виходам. В цих випадках або один вихід визначається станом іншого, або маємо справу зі складним об'єктом керування, який може бути представлено групою простих механізмів.

Найпростіший спосіб керування об'єктом, це реалізувати функцію (1) окремо під кожну одиницю обладнання. Такий підхід дозволяє використати бітову область меркерної пам'яті та тегової пам'яті вводу чи виводу будь-якого промислового контролера. Вказаний метод не придатний для великої кількості обладнання. По-перше, технологічне обладнання має не лише початковий та цільовий стани, але також і перехідні та аварійні. Тому у цифровому автоматі з'являється пам'ять, біти спрацювання таймерів, а сама реалізація може збільшитись до значних обсягів коду, що повинен бути відлагоджений. По-друге, для зв'язку з засобами SCADA потрібно використовувати або окремі ліцензійні теги під кожну бітову змінну (а це не лише змінні входів і виходів, але і ознаки стану обладнання в цілому), або писати несистематизований програмний код для збірки та розбірки складних бітових структур даних.

Сучасний загальноприйнятий підхід до програмування промислових контролерів пропонує використовувати упорядкування інформаційних ознак сигналів «Так-Ні» до масивів і реалізовувати функцію (1) для масивів обладнання у формі

$$Y = f(X_1; X_2; X_3; \dots X_n), \quad (2)$$

де  $Y$ , та  $X$  – упорядковані вектори вхідних та вихідних інформаційних ознак

механізмів, причому індекси елементів (інакше кажучи, номери позицій) – це деякі умовні ідентифікатори (або номери) керованих механізмів. Ідея полягає у тому, що програмний розподіл бітових ознак входів та виходів проводиться у визначеному місці програми промислового контролера по заздалегідь складеним таблицям. Потім реалізується алгоритм керування деяким абстрактним універсальним векторним цифровим автоматом, кількість входів «Так-Ні» якого достатня для керування технологічним механізмом з найбільшою кількістю сигналів[3]. Причому керування кожним механізмом цього автомату відбувається у порядку справжнього паралелізму (тобто не псевдопаралелізму, використаного багатозадачних операційних системах).

На практиці це виглядає наступним чином. Вхідні та вихідні біти керування механізмами з відповідних областей пам'яті переносяться з або до меркерної області у загальну змінну простого беззнакового типу. Перенос виконується з або до позицій змінних  $Y$  та тих  $X$ , що відповідають номерам входів керованих

механізмів. Якщо у конкретного механізму входів  $X$  менше максимального  $n$ , то

інформаційні ознаки  $X$ , яким не вистачає входів заповнюються одиницями чи

нулями, у залежності від того, яке значення не буде впливати на обчислення стану та вихідної змінної механізму.

Таким чином, розробник програмного забезпечення для промислових контролерів реалізує універсальний абстрактний автомат, до якого підключає всі керовані механізми у виділеному для цього програмному блоці, використовуючи для цієї мети цілий беззнаковий тип даних, такий як байт, слово або подвійне слово. У результаті, єдиний відлагоджений програмний блок за допомогою простих типів даних паралельно обробляє вектори входів чи виходів відповідно на 8, 16 або 32 механізми, розподіл яких показано на рисунку 1, де  $m$  – умовні

номери механізмів:

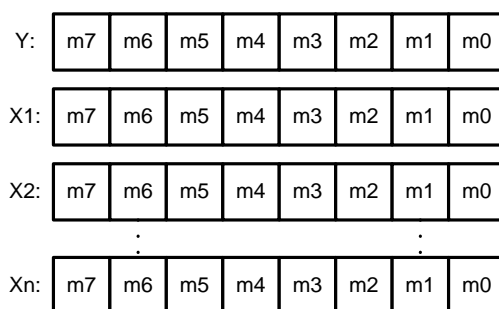


Рис. 1. Абстрактний вектор входів механізмів у вигляді позиційних бітів змінних типу «байт»

Перевага такого підходу у тому, що є можливість відлагодити керування механізмами у лабораторних умовах, а на об'єкті лише програмно підключити потрібні входи та виходи. До того ж, отримується явна економія ліцензійних тегів навіть без використання складних типів даних. Якщо розробник все ж вирішить їх використати, то скласти та розібрати масив або гранд-тег байтів, слів чи подвійних слів абстрактного автомату – задача значно простіша у порівнянні з окремими бітами і також може бути вирішена один раз у лабораторних умовах.

Сам по собі показаний підхід не вирішує задачу перерозподілу входів та виходів на етапі експлуатації без корекції програмного коду контролера. Для динамічного перерозподілу фізичних входів та виходів між механізмами абстрактного цифрового автомату повинна застосовуватись непряма адресація бітів, які несуть інформаційні ознаки сигналів. Таким чином, мають бути організовані асоціативні таблиці конфігурації механізмів, строки яких містять опис «номер механізму – номер входу чи вихід – номер фізичного входу чи виходу контролера». Такі конфігураційні таблиці досить просто організувати, використовуючи масиви простих типів або гранд-теги.

Складність наведеного шляху у тому, що механізми непрямой адресації у промислових контролерів недосконалі або відсутні. Той же синтаксис STEP 7 дозволяє звернутись до біту вхідної, вихідної чи меркерної області пам'яті або за заданим ім'ям біту, або вказавши ім'я чи адресу байта та через крапку – номер потрібного у байті біта. Використання такої непрямой адресації не має сен-

су, простіше прямо іменувати потрібні біти. Регістрова область пам'яті (або область гранд-тегів у контролерах Siemens) має повноцінну непряму адресацію, але не працює з бітами[4]. Тому необхідна методика повноцінної непрямої адресації бітів у меркерній пам'яті промислового контролера на базі доступних засобів релейно-контактної логіки промислових контролерів.

Засоби програмування релейно-контактної логіки Siemens дозволяють реалізувати непряму адресацію бітів меркерної шляхом використання масок та зсувів для кожного фізичного входу чи виходу.

Розглянемо цей метод. Весь адресний простір входів відображається до регістрової пам'яті засобами блокового копіювання. У регістровій пам'яті передбачено також простір виходів, який щоскана має аналогічно копіюватись до тегової пам'яті виходів. Позначимо їх як масиви  $A$  і  $B$ , бітового типу. Розмірності цих масивів співпадають з кількістю входів та виходів плюс 2 біти з нулем та одиницею для масиву  $A$ .

Вводимо бітовий масив-маску  $C = 0000 \dots 0001b$  бітового типу, причому його довжина співпадає з довжиною масиву  $A$ .

Тоді будь-який біт  $x_{jz}$  вхідного масиву  $X[m; n]$  для цифрового автомату (2)

можна встановити як

$$x_{jz} = \begin{cases} 1, & \text{якщо } (A \& (C * (j * z + 1))) > 0 \\ 0, & \text{якщо } (A \& (C * (j * z + 1))) = 0 \end{cases} \quad (3)$$

де індекси  $j$  та  $z$  – відповідно, умовний номер механізму у цифровому автоматі, та номер входу цього механізму, як показано на рисунку 1.

Для непрямої адресації виходів вводимо бітову маску  $d = FF FF FF FEh$  типу DWORD (вона може мати значення  $1111 1110b$  або  $FF FEh$  при використанні типів BYTE або WORD). Тоді біт  $b_l$  масиву  $B$  може бути встановлений як

$$b_l = \begin{cases} 1, & \text{якщо } (Y_z \& (D * z + 1)) > 0 \\ 0, & \text{якщо } (Y_z \& (D * z + 1)) = 0 \end{cases} \quad (4)$$

де індекс  $l$  – відповідно, номер фізичного виходу механізму  $Y_z$  у цифровому автоматі, показаному на рисунку 1.

Індекси  $j$  та  $z$  приймаються з конфігураційної таблиці механізмів. Причому, якщо вхід або вихід у конкретному механізмі не використовується, що та-

кож буде вказано у конфігурації  $b_l = 0$ , а  $x_{jz}$  встановлюється так, щоб не впливати на результати виконання автомату по іншим входам.

Очевидно, що фрагменти  $(C * (j * z + 1))$  формули (3) та  $(D * z + 1)$  формули (4) є арифметичним зсувом вліво для цілих беззнакових типів даних.

Формула (3) повинна повторюватись по  $m * n$  рази кожна для всіх бітів масиву  $A$ . Для типу DWORD  $m = 32$ .

Формула (4) і (8) повинна повторитись  $n$  разів для всіх бітів масиву  $B$ .

Сучасні промислові контролери мають засоби організації циклів під час виконання скану програми. Цей засіб потенційно може призвести до спрацювання сторожового таймеру контролера та до зупинки програми, або до більш тяжких наслідків на об'єкті, якщо сторожовий таймер відключити. Тому бажано складати алгоритми параметричних циклів, щоб гарантувати його закінчення. Реалізувавши відповідні цикли обчислень за формулами (3) і (4), отримаємо непряму адресацію входів і виходів сигналів «Так-Ні» у тегових областях пам'яті входів і виходів. Апробація розкритого алгоритму у реалізованій системі керування транспортно-технологічними маршрутами елеватора [5], яка об'єднувала приблизно 64 механізми з одним виходом по 4 входи на механізм на контролері Simatic 1214C показала, що на виконання алгоритму потрібно витратити близько 200 мс часу скану. Враховуючи, що сторожовий таймер промислового контролера за замовчуванням встановлено на 500 мс, витрати часу на бітову непряму адресацію хоч і значні, але прийнятні. У випадку збільшення входів і виходів рекомендується перейти на більш продуктивну елементну базу.

**Висновки.** Виходячи з аналізу існуючих підходів до складання алгоритмів керування технологічним обладнанням, які працюють з сигналами «Так-Ні», розроблено метод динамічного перерозподілу інформаційних ознак сигналів між входами та виходами комплексу технічних засобів системи керування на елементній базі серійних промислових контролерів. Метод використовує послідовні зсуви та накладення масок на бітові масиви у регістровій пам'яті промислового контролера для кожного виходу та входу «Так-Ні» керованого технологічного обладнання, реалізуючи, таким чином, їх непряму побітову адресацію. Такий підхід разом з використанням розглянутого у статті абстрактного цифрового автомату керування обладнанням та асоціативних таблиць з прописаною конфігурацією входів і виходів керованих механізмів дозволяє динамічно, у процесі експлуатації системи, пере-призначати виходи керування та входи «Так-Ні» системи керування без втручання у програмний код та без повної зупинки системи. Таке перепризначення може виконуватись засобами SCADA автоматизованого робочого місця системи керування технологічним процесом.

Подальший розвиток досліджень буде спрямовано на оптимізацію методу, яка б дозволила скоротити час скану програми контролера, що витрачається на непряму бітову адресацію.

#### **Перелік посилань**

1. Интерактивный каталог СА 01. Siemens, 2016.
2. РД 50-34.698-90. Автоматизированные системы. Требования к содержанию документов. Методические указания.
3. Ткаченко С.Н., Бабенко А.С. Моделирование работы транспортно-технологического оборудования элеватора как метод принятия решений для автоматизированного управления транспортно-технологическими маршрутами. // Дніпропетровськ: Редакційно-видавничий комплекс НГУ, Зб. наук. праць НГУ. – 2007. – №27. – С. 161-166.
4. Simatic S7. Программируемый контроллер S7-1200. Системное руководство. – Siemens, 11/2009. – 391 с.
5. Просяник А.В., Ткаченко С.М., Горбунов М.Ю., Просяник М.А. Функціональна структура для керування технологічними маршрутами транспортування зерна із застосуванням SCADA-систем.// Зб. наук. праць. – НГУ. – 2010. – №34 – Т. 2.

#### **ABSTRACT**

**Purpose.** Based on an analysis of existing approaches to the preparation process equipment control algorithms, which work with signals "Yes-No" to develop a method of dynamic reallocation of information signal tags between inputs and outputs of complex of technical facilities management system that use comptrollers.

**The methods** of research is to analyze of existing approaches to the management of technological equipment, to the use of tags for data exchange between comptrollers and SCADA-system, to analyze capabilities of comptrollers programming.

**Findings.** Shown that the sharing of the developed method masks and shifts for bit arrays, digital abstract machine of equipment control, associative configuration tables of inputs and outputs, allows to realize dynamic reallocation of signal tags between inputs and outputs of complex of technical facilities management system.

**The originality** is to developing a method of bit arrays and sequences indirect addressing for use with mass programming of comptrollers.

**Practical implications.** Possibility of dynamic reallocation of outputs and inputs "Yes-No" during of the management system commissioning and exploitation without interfering in the code comptroller.

**Keywords:** *dynamic reallocation of signal tags, bit indirect addressing, signals "Yes-No", comptroller*