

Andrzej Maczyński

STEROWNIKI PROGRAMOWALNE PLC

Budowa systemu i podstawy programowania

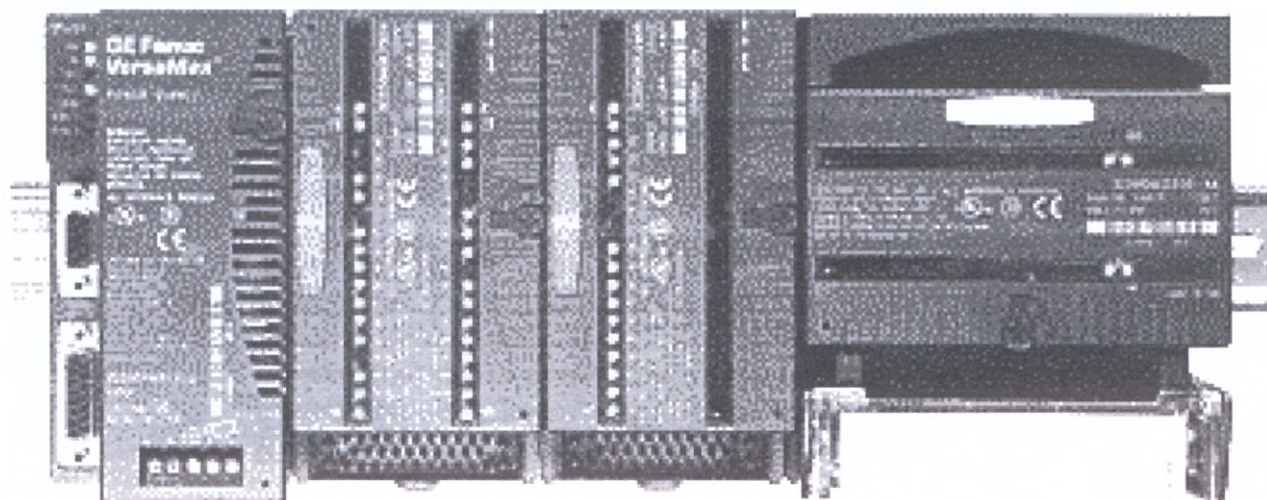


Biblioteka Politechniki Krakowskiej



100000340455

STEROWNIKI PROGRAMOWALNE PLC



Budowa systemu i podstawy programowania

Andrzej Maczyński

111-309553

Pracę recenzował dr hab. inż. Jacek Kłosiński

Autor jest adiunktem Katedry Podstaw Budowy Maszyn
Akademii Techniczno-Humanistycznej w Bielsku-Białej

Wydawca:
Astor Sp. z.o.o.
ul. Smoleńsk 29, 31-112 Kraków
tel. (012) 428-63-70
e-mail: info@astor.com.pl
www.astor.com.pl

ISBN 83-915620-6-9

3063-268/2022

Spis treści

Przedmowa	1
Od autora	3
1. Wiadomości wstępne	5
1.1. Pojęcia podstawowe	5
1.2. Sygnał.....	6
1.3. Podstawy algebry Boole'a.....	8
1.4. Realizacja techniczna i oznaczania funkcji logicznych na schematach.....	11
1.5. Układy logiczne.....	13
1.6. Systemy liczbowe.....	15
2. Sterowniki PLC – omówienie systemu	19
2.1. Rys historyczny	19
2.2. Zalety sterowników PLC.....	22
2.3. Opis systemu.....	25
2.3.1. <i>Hardware</i>	25
2.3.2. <i>Software</i>	28
2.3.3. <i>Elementy wejściowe</i>	28
2.3.4. <i>Elementy wyjściowe</i>	29
2.3.5. <i>Programator</i>	29
2.3.6. <i>Kasety rozszerzające</i>	30
2.4. Układy redundancyjne.....	30
2.5. Cykl pracy sterownika.....	31
2.6. Konfiguracja sterownika.....	34
3. Sterowniki serii 90-30 firmy GE Fanuc	37
3.1. Jednostki centralne.....	37
3.2. Moduły wejść/wyjść.....	40
3.3. Moduł szybkiego licznika HSC.....	42
3.4. Moduły pozycjonujące dla 1 lub 2 osi.....	43
3.5. Moduł pozycjonujący dla silnika krokowego.....	43
3.6. Moduł programowalnego koprocesora PCM – Programmable Coprocessor Module.....	44
3.7. Moduł komunikacyjny TCP/IP Ethernet.....	45
4. Podstawy programowania	47
4.1. Języki programowania sterowników wg IEC 61131.....	47
4.2. Typy danych i zmiennych.....	48
4.3. Elementy logiczne programu sterującego w sterownikach GE Fanuc serii 90-30.....	51
4.4. Podstawowe zasady przy pisaniu programu sterującego w języku LD na sterownikach GE Fanuc.....	58

4.5. Przykłady programowania.....	60
4.5.1. Zapis w języku LD funkcji logicznych.....	60
4.5.2. Wykorzystanie bloku funkcyjnego ONDTR.....	61
4.5.3. Wykorzystanie bloku funkcyjnego MUL.....	62
4.5.4. Podstawy użycia bloku funkcyjnego PID.....	63
5. Schemat funkcji sekwencyjnej SFC.....	69
5.1. Podstawy SFC.....	69
5.2. Przykład sterowania manipulatorem.....	71
6. Sieć przemysłowa GENIUS.....	77
6.1. Charakterystyka elementów sieci.....	78
6.1.1. Bloki GENIUS.....	78
6.1.2. Kontrolery sieci.....	80
6.1.3. System FIELD CONTROL.....	82
6.1.4. 90-70 Remote Control I/O Scanner.....	83
6.1.5. Interfejs komunikacyjny serii VersaMax.....	83
6.1.6. Hand Held Monitor HHM.....	83
6.2. Komunikacja w sieci GENIUS.....	84
6.3. Zasady łączenia urządzeń w sieci GENIUS.....	87
6.4. Redundancja i monitoring.....	87
6.4.1. Monitoring danych.....	87
6.4.2. Redundancja sieci i kontrolerów.....	88
6.4.3. Redundancja sterowników i kontrolerów.....	89
7. Monitoring – systemy SCADA.....	91
7.1. Wprowadzenie.....	91
7.2. Podstawowe cechy użytkowe systemów SCADA i dziedziny ich zastosowań.....	94
7.3. Omówienie głównych cech systemów SCADA.....	97
8. Podsumowanie.....	103
Dodatek D1. Prezentacja sterowników firmy GE Fanuc.....	105
Dodatek D2. Oprogramowanie narzędziowe GE Fanuc CIMPLICITY.....	127
Dodatek D3. Porty szeregowo.....	129

Przedmowa

Książka pt.: „Sterowniki programowalne PLC. Budowa systemu i podstawy programowania” autorstwa Andrzeja Maczyńskiego została napisana z myślą o studentach Akademii Techniczno - Humanistycznej w Bielsku-Białej jako pomoc dydaktyczna przy prowadzeniu zajęć związanych z automatyką procesów. Zawarto w niej następujące zagadnienia:

- podstawowe terminy i pojęcia analizy układów przełączających,
- budowa sterowników PLC wraz z opisem elementów wchodzących w skład sterownika,
- zasady tworzenia oprogramowania ze szczególnym uwypukleniem formalizmu SFC,
- opis elementów tworzących sieci przemysłowe i bazujące na nich systemy nadzorowania i monitorowania procesów przemysłowych.

Podręcznik został napisany jako kompendium wiedzy dotyczącej sterowników ze zwróceniem szczególnej uwagi na informacje przydatne w trakcie prowadzenia zajęć dydaktycznych (wykładów i ćwiczeń laboratoryjnych). Z tego powodu znalazło się w nim omówienie konkretnych sterowników i wykorzystujących je układów automatyki łącznie z dołączoną w postaci dodatku prezentacją sterowników jednej firmy (GE Fanuc Automation). Napisany jest w bardzo zwartej formie i może być wykorzystywany przez studentów różnych wydziałów dysponujących podstawowym przygotowaniem z zakresu informatyki i teorii sterowania jako uzupełnienie wykładów i pomoc w laboratorium. Może być także przydatny dla tych wszystkich, którzy zawodowo zajmują się układami automatyki wykorzystującymi sterowniki PLC.

Uważam książkę za cenną publikacją dotyczącą urządzeń, których opisy w ostatnich latach niemal całkowicie zdominowały wydawnictwa reklamowe i katalogowe konkurujących ze sobą producentów elementów i systemów sterowania.

Dr hab. inż. Jacek Kłosiński

Od autora

Podręcznik powstał na zapotrzebowanie zgłaszane przez studentów Wydziału Budowy Maszyn i Informatyki Akademii Techniczno-Humanistycznej (dawniej Filii Politechniki Łódzkiej) w Bielsku-Białej. Na rynku księgarskim daje się bowiem odczuć brak pozycji omawiających kompleksowo systemy automatyki oparte o sterowniki PLC. Szereg wyrywkowych informacji na temat sterowników można znaleźć w materiałach reklamowych i katalogach poszczególnych firm. W czasopiśmie poświęconych automatyce także ukazują się niekiedy artykuły poświęcone tym zagadnieniom. Niniejsza pozycja jest próbą zestawienia podstawowych informacji o sterownikach PLC i przedstawienia ich w możliwie przystępny sposób. Autor wyraża nadzieję, że praca będzie przydatna jako uzupełnienie wykładów oraz pomoc w zajęciach laboratoryjnych z przedmiotów: Sterowniki Programowalne, Sterowanie i Monitoring, Podstawy Automatyki i pokrewnych na wydziałach mechanicznych. Korzystać z niej mogą także studenci innych wydziałów: informatyki, elektrotechniki, ochrony środowiska, a też osoby związane zawodowo z automatyką, a chcące poznać układy oparte o sterowniki PLC. Może ona stanowić pomoc dla wszystkich, którzy uczestniczą w podstawowych kursach związanych z obsługą i programowaniem PLC.

Podręcznik podzielono na 8 rozdziałów:

1. **Wiadomości wstępne**, w którym przypomniano podstawowe pojęcia przydatne przy zapoznawaniu się z kolejnymi rozdziałami. Omówiono także podstawy algebry Boole'a i systemów liczbowych;
2. **Sterowniki PLC – omówienie systemu** przedstawiający historię, zalety i podstawy budowy systemów sterownikowych;
3. **Sterowniki serii 90-30 firmy GE Fanuc** – prezentujący współczesne sterowniki PLC średniej mocy na przykładzie jednej wybranej rodziny sterowników (taka metoda omawiania zagadnienia jest określana jako „case study”);
4. **Podstawy programowania** – w którym przedstawiono języki programowania zgodne z normą IEC 61131-3, typy danych i zmiennych, oraz omówiono elementy logiczne języka drabinkowego LD i podstawowe zasady programowania na przykładzie oprogramowania narzędziowego Logicmaster 90 dla sterowników GE Fanuc;
5. **Schemat funkcji sekwencyjnej SFC** – omawiający podstawy formalizmu SFC i zawierający prosty przykład sterowania manipulatorem;
6. **Sieć przemysłowa GENIUS** – prezentujący przykład sieci polowej wraz z jej możliwościami redundancyjnymi;
7. **Monitoring – systemy SCADA** – w którym zaprezentowano podstawowe zadania monitoringu przemysłowego oraz cechy systemów SCADA;
8. **Podsumowanie** – sygnalizujący prawdopodobne kierunki rozwoju sterowników PLC.

Ponadto załączono trzy dodatki:

1. **Prezentacja sterowników firmy GE Fanuc** – przedstawiający aktualną (II połowa 2001 roku) ofertę sterowników PLC firmy GE Fanuc Automation. Dokładniej omówiono w nim sterowniki małe (VersaMax Nano i Micro) i średnie (VersaMax). Prezentacja ta została zamieszczona w dodatku, ponieważ tego typu informacje, w przeciwieństwie do większości zawartych w części głównej skryptu, ulegają dość szybkiej dezaktualizacji;
2. **Oprogramowanie narzędziowe GE Fanuc CIMPLICITY** – zawierający krótką charakterystykę nowego zintegrowanego oprogramowania przemysłowego firmy GE Fanuc;
3. **Porty szeregowo** – zawierający krótką charakterystykę standardów RS232 i RS422/RS485.

Poszczególne rozdziały obejmują określone zagadnienia i mogą być analizowane oddzielnie. Jednak, ze względu na ich wzajemne powiązania, zaleca się Czytelnikowi krótkie zapoznanie się z rozdziałami wcześniejszymi. Na końcu każdego z nich zamieszczono spis literatury. Są tam pozycje źródłowe, z których zaczerpnięto informacje i część ilustracji, jak i pozycje zawierające dodatkowe dane na omawiany temat.

W publikacji świadomie używano niekiedy różnych określeń do opisu tego samego pojęcia np.: warstwa obiektowa – czujniki i elementy, krok – etap, itp. Celem takiego podejścia było zapoznanie Czytelnika z różnymi nazwami spotykanymi w literaturze. Brakuje bowiem jednoznaczności pojęć w prezentowanej dziedzinie i różni autorzy oraz producenci używają różnych słów na określenie tego samego pojęcia. Książka przedstawia informacje ogólne, stąd też często pojawiają się w niej stwierdzenia typu: zazwyczaj, najczęściej itp. Pewne rozwiązania techniczne różnych producentów mogą bowiem różnić się w szczegółach, natomiast ich idea jest zazwyczaj podobna.

W najbliższym czasie autor planuje opracować, wraz ze współpracownikami z Katedry Podstaw Budowy Maszyn, publikację w formie skryptu pt. „Sterowniki PLC. Laboratorium” stanowiącą uzupełnienie niniejszej pozycji. Zostanie w niej przedstawione programowanie i obsługa sterowników GE Fanuc przy użyciu oprogramowania narzędziowego Logicmaster oraz VersaPro. Zaprezentowane zostaną przykłady zadań z rozwiązaniami a także stanowiska laboratoryjne, w które wyposażone jest Laboratorium Sterowników PLC Katedry Podstaw Budowy Akademii Techniczno-Humanistycznej.

Autor zwraca się z prośbą do Szanownych Czytelników o zasygnalizowanie zauważonych usterek oraz przekazanie wszelkich innych uwag, które mogłyby pomóc w opracowaniu ewentualnych kolejnych wydań.

Na zakończenie autor wyraża serdeczne podziękowania dr. hab. inż. Jackowi Kłosińskiemu za cenne uwagi wnoszone podczas całego okresu opracowywania niniejszej publikacji. Słowa wdzięczności kieruje także do Pani mgr Ewy Bebek z firmy AF Seko w Bielsku-Białej za korektę części związanej z prezentacją sterowników GE Fanuc Automation. Dziękuję również firmie ASTOR za pomoc w wydaniu książki.

Październik 2001

Andrzej Maczyński
amaczynski@pb.bielsko.pl

1. Wiadomości wstępne

Na początku przedstawione zostały wiadomości wstępne, których przypomnienie pozwoli Czytelnikowi na lepsze zrozumienie dalszej części książki. Dotyczą one w szczególności podziału sygnałów występujących w sterowaniu procesami przemysłowymi, algebry Boole'a, definicji i podziału układów logicznych oraz systemów liczbowych. Czytelnik znający te zagadnienia może poniższy rozdział pominąć.

1.1. Pojęcia podstawowe

W kolejnych rozdziałach używane będą pojęcia, których definicje podano poniżej. Definicje te nie są być może ściśle, jednak całkowicie wystarczające dla potrzeb niniejszej publikacji.

Sterowanie	celowe oddziaływanie na obiekt sterowania w celu uzyskania działania tego obiektu zgodnego z pożądanym algorytmem działania.
Obiekt sterowania	urządzenie lub zestaw urządzeń, w którym przebiega sterowany proces, i w którym przez zewnętrzne oddziaływanie sterujące realizuje się pożądaný algorytm działania.
Proces sterowany	proces, w którym realizacja pożądanego algorytmu działania tego procesu wymaga odpowiedniego oddziaływania z zewnątrz.
Sterowanie automatyczne	sterowanie realizujące założony algorytm sterowania bez pośredniego udziału człowieka.
Informacja	wiadomość o charakterze i stanie zjawisk, procesów technologicznych i urządzeń.
Sygnal	umowna cecha określonej wielkości fizycznej zawierająca informacje.

W przemyśle, zwłaszcza w wytwórczym, często dzieli się procesy na **ciągłe**, **dyskretne** i **wsadowe**.

Proces ciągły	proces, w którym mamy do czynienia ze stałym, nieprzerwanym strumieniem produktu, którego ilość i jakość może być sterowana przez takie zmienne jak temperatura, ciśnienie, przepływ. Przykładami takich procesów może być produkcja papieru, ciągłe odlewanie stali czy proces oczyszczania ścieków.
Proces dyskretny	proces, w którym występują dyskretne, powtarzalne operacje produkcyjne, takie jak przy produkcji części do samochodu, narzędzi itp.
Proces wsadowy	proces, w którym po załadowaniu składników, komponentów, następuje proces technologiczny, w wyniku którego otrzymuje się produkt w postaci ciągłej – np. produkcja surówki w piecu, procesy w różnego rodzaju reaktorach chemicznych.

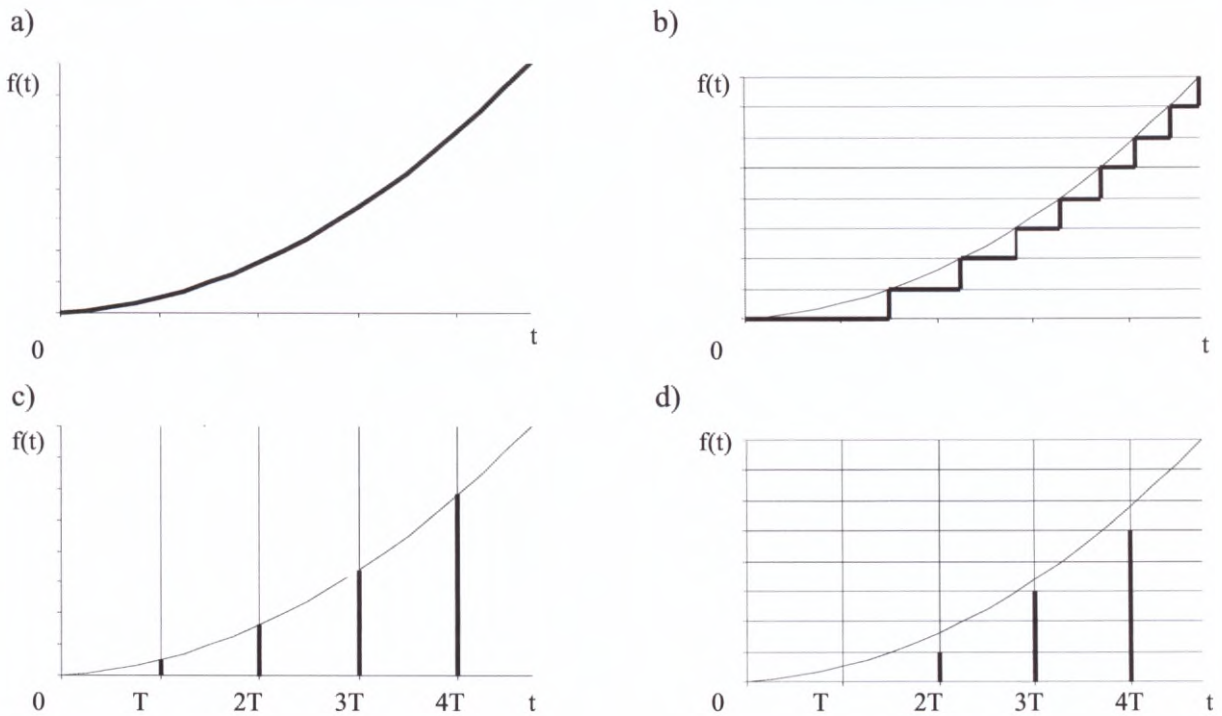
1.2. Sygnał

Bardzo ważnym pojęciem w automatyce jest wspomniany wyżej **sygnał**. W zależności od wartości, jakie może przyjmować, dzielimy go na **analogowy (ciągły)**, **kwantowany**, **próbkowany**, oraz **jednocześnie kwantowany i próbkowany** (rys. 1.1.).

Sygnał analogowy sygnał, którego wartości należą do ciągłego (nieprzeliczalnego) zbioru i mogą się zmieniać w dowolnej chwili.

Sygnał kwantowany sygnał, którego wartości należą do dyskretnego (przeliczalnego) zbioru i mogą się zmieniać w dowolnej chwili – sygnał dyskretny w poziomie.

Sygnał próbkowany sygnał będący ciągiem impulsów – sygnał dyskretny w czasie.



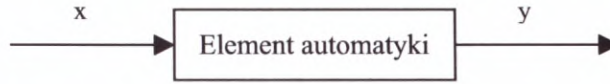
Rys. 1.1. Podstawowy podział sygnałów
 Sygnał a) analogowy b) kwantowany c) próbkowany d) kwantowany i próbkowany

W sterowaniu procesami przemysłowymi (np. montażowymi, obróbczymi itp.) bardzo istotną rolę pełni sygnał dwustanowy.

Sygnał dwustanowy sygnał skwantowany, który może przyjmować tylko dwie wartości umownie określane jako 1 i 0.

W przemysłowych systemach sterowania istnieją pewne standardy związane z sygnałami analogowymi i dwustanowymi. Zmiana wartości sygnału analogowego może być rejestrowana jako zmiana napięcia (sygnał napięciowy) lub prądu (sygnał prądowy). Dla sygnałów napięciowych napięcie zmienia się w zakresie ± 10 V (ewentualnie ± 5 V, 0 - 10 V lub 0 - 5 V). Dla sygnałów prądowych stosuje się jeden z dwóch zakresów zmiany natężenia prądu: zakres 0 - 20 mA oraz 4 - 20 mA. Wg PN-IEC 1133-2 pierwszy zakres nie jest zalecany do nowych instalacji. Sygnały dwustanowe są przedstawiane jako zmiana napięcia, przy czym napięcie nominalne wynosi najczęściej 12 V, 24 V, 125 V dla prądu stałego, oraz

120 V lub 240 V dla prądu przemiennego. Przykładowo, dla napięcia nominalnego 24 V, gdy napięcie sygnału znajduje się w zakresie 0 - 5 V wówczas jest on interpretowany jako 0, natomiast gdy napięcie sygnału znajduje się w zakresie 11,5 - 30 V wówczas jest on interpretowany jako 1. W zakresie 5 - 11,5 V sygnał jest niejednoznaczny.



Rys. 1.2. Sygnał wejściowy i wyjściowy

Innym podziałem sygnałów w automatyce jest podział na **sygnały wejściowe** i **sygnały wyjściowe**. Sygnały te przedstawiono na rys. 1.2., który jest intuicyjnie zrozumiały – x to sygnał wejściowy a y wyjściowy. Sygnał wyjściowy z jednego elementu automatyki jest najczęściej sygnałem wejściowym dla innego elementu.

W informatyce i dziedzinach pokrewnych pojedynczy sygnał dwustanowy nazywany jest **bitem**.

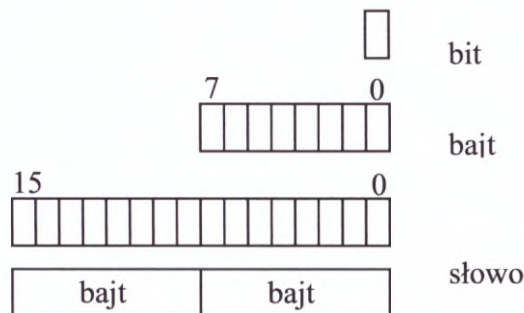
Bit sygnał, którego stan jest określony poprzez jedną z dwóch wartości 0 lub 1. Jest to najmniejsza jednostka informacji.

Bity zestawione w uporządkowany sposób (kod) tworzą sygnał binarny umożliwiającą przedstawienie większej liczby wartości. Zestawienie n bitów pozwala na zapisanie 2^n wartości (np. 2 bity umożliwiają zapis 4 kombinacji). Większe jednostki informacji to:

Bajt zestawienie 8 bitów. Bajt pozwala na zapis 256 wartości tj. wartości 0,...,255;

Słowo (Word) dwa kolejne bajty interpretowane jako 16 bitów;

Słowo podwójnej długości dwa kolejne słowa, czyli 32 bity.



Rys. 1.3. Podstawowe jednostki informacji i numeracja bitów

Na rys. 1.3. przedstawiono podstawowe jednostki informacji oraz sposób numeracji bitów. Numeracja zaczyna się od prawej strony od numeru 0 – **najmniej znaczący bit** (LSB – Least Significant Bit). Bit o numerze najwyższym to **najbardziej znaczący bit** (MSB – Most Significant Bit).

Często spotkać można przedrostek kilo (k) poprzedzający słowa bit, bajt itd. Przedrostek ten oznacza w tym przypadku 2^{10} . Zatem kilobit, kilobajt czy kilosłowo to

odpowiednio 1024 bity, bajty lub słowa. Używa się również przedrostków mega ($M=1.024 \times 1.024=1.048.576$) i giga ($G=1.073.741.824$).

1.3. Podstawy algebry Boole'a

Jak wspomniano w poprzednim podrozdziale, w sterowaniu różnymi procesami bardzo istotną rolę pełnią sygnały dwustanowe. Układy, dla których zarówno sygnały wejściowe jak i wyjściowe są sygnałami dwustanowymi nazywamy **układami logicznymi**. Do ich opisu stosuje się algebrę Boole'a, a w jej obrębie funkcje logiczne. Poniżej zdefiniowano funkcje logiczne jednej zmiennej, najważniejsze funkcje dwóch zmiennych, aksjomaty i reguły algebry Boole'a oraz sposoby graficznego przedstawiania funkcji logicznych.

Funkcje logiczne funkcje, dla których zarówno argumenty (zmienne wejściowe), jak i wartości (zmienne wyjściowe) przyjmują jedynie dwa stany, oznaczone umownie 0 i 1 (false, true).

Funkcje logiczne definiuje się najczęściej poprzez podanie tabeli wartości funkcji.

Funkcje logiczne 1 zmiennej $y=f(x)$ podano poniżej.

x \ y	0	1	Równanie	Nazwa funkcji
f_0	0	0	$y = 0$	stała zerowa
f_1	0	1	$y = x$	powtórzenie
f_2	1	1	$y = 1$	stała jedynka
f_3	1	0	$y = \bar{x}$	negacja, inwersja, NOT

Funkcji logicznych 2 zmiennych jest szesnaście. Wynika to z faktu, że istnieje jedynie 16 nie powtarzających się kombinacji zer i jedynek na czterech przeznaczonych na wartości funkcji miejscach w tablicy wartości funkcji. Poniżej podane zostaną najważniejsze funkcje logiczne 2 zmiennych.

1. Alternatywa (OR, suma logiczna, dysjunkcja)

x_1	x_2	y
0	0	0
0	1	1
1	1	1
1	0	1

$$y = x_1 + x_2$$

2. Koniunkcja (AND, iloczyn logiczny)

x_1	x_2	y
0	0	0
0	1	0
1	1	1
1	0	0

$$y = x_1 \cdot x_2$$

3. Negacja alternatywy (NOR, funkcja Peirce'a)

x_1	x_2	y
0	0	1
0	1	0
1	1	0
1	0	0

$$y = \overline{x_1 + x_2}$$

4. Negacja koniunkcji (NAND, funkcja Sheffera)

x_1	x_2	y
0	0	1
0	1	1
1	1	0
1	0	1

$$y = \overline{x_1 \cdot x_2}$$

W algebrze Boole'a występuje szereg **praw (aksjomatów)**, które przyjmuje się bez dowodu.

1. Prawa przemienności

$$x_1 + x_2 = x_2 + x_1$$

$$x_1 \cdot x_2 = x_2 \cdot x_1$$

2. Prawa łączności

$$(x_1 + x_2) + x_3 = x_1 + (x_2 + x_3)$$

$$(x_1 \cdot x_2) \cdot x_3 = x_1 \cdot (x_2 \cdot x_3)$$

3. Prawa rozdzielności

$$(x_1 + x_2) \cdot x_3 = x_1 \cdot x_3 + x_2 \cdot x_3$$
$$x_1 \cdot x_2 + x_3 = (x_1 + x_3) \cdot (x_2 + x_3)$$

4. Prawa (de Morgana)

$$\overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2}$$
$$\overline{x_1 \cdot x_2} = \overline{x_1} + \overline{x_2}$$

5. Prawa powtórzeń

$$x + x = x$$
$$x \cdot x = x$$

6. Prawa dopełnienia

$$x \cdot \overline{x} = 0$$
$$x + \overline{x} = 1$$

7. Prawa działania na elementach neutralnych

$$x \cdot 0 = 0$$
$$x + 0 = x$$
$$x \cdot 1 = x$$
$$x + 1 = 1$$

Oprócz praw, w algebrze Boole'a występują także **reguły**, które mogą być udowodnione przy użyciu przedstawionych powyżej związków. Przykłady reguł podano poniżej.

Reguły pochłaniania

1.
$$x_1 + \overline{x_1} \cdot x_2 = x_1 + x_2$$

Dowód:

$$x_1 + \overline{x_1} \cdot x_2 = (x_1 + \overline{x_1}) \cdot (x_1 + x_2) = 1 \cdot (x_1 + x_2) = x_1 + x_2$$

2.
$$x_1 + x_1 \cdot x_2 = x_1$$

3.
$$x_1 \cdot (x_1 + x_2) = x_1$$

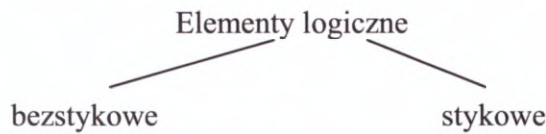
4.
$$x_1 \cdot (\overline{x_1} + x_2) = x_1 \cdot x_2$$

Reguły sklejania

5. $(x_1 + x_2) \cdot (x_1 + \overline{x_2}) = x_1$
 6. $x_1 \cdot x_2 + x_1 \cdot \overline{x_2} = x_1$

1.4. Realizacja techniczna i oznaczania funkcji logicznych na schematach

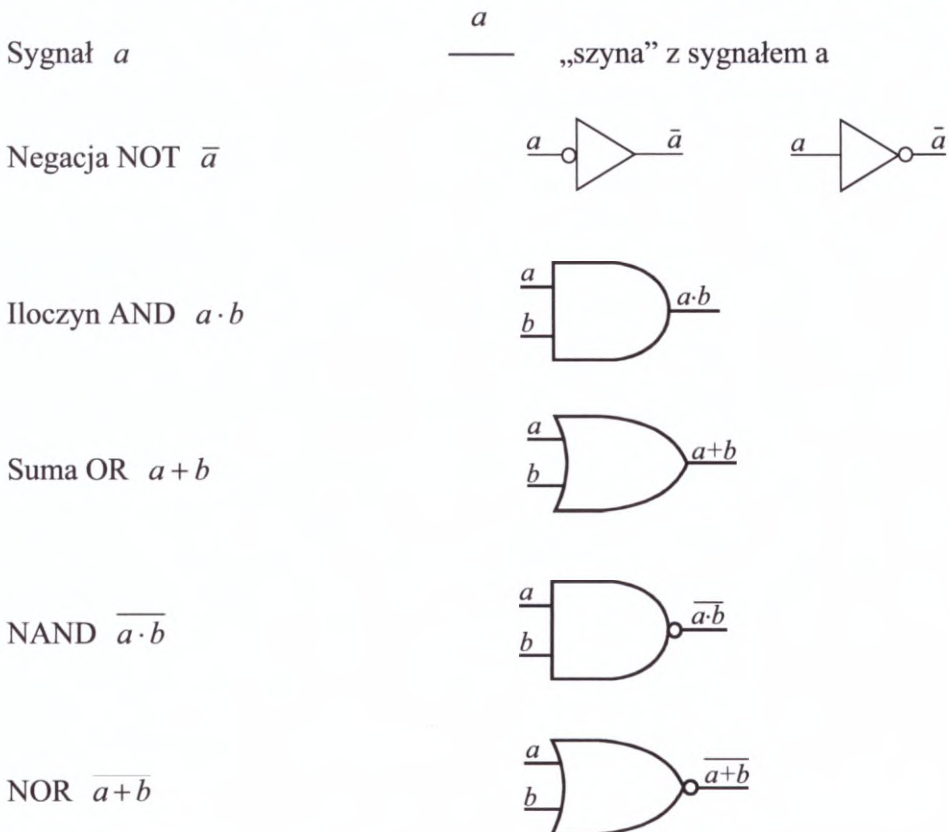
Podczas technicznej realizacji funkcji logicznych używa się tzw. elementów logicznych, które wykorzystują różne zjawiska fizyczne. Najczęściej elementy logiczne to elementy: elektroniczne, elektromagnetyczne, hydrauliczne, pneumatyczne, magnetyczne, światłowodowe. Ze względu na konsekwencje w oznaczaniu elementów logicznych na schematach, istotny jest ich podział na elementy **stykowe** i **bezstykowe** – patrz rys. 1.4.



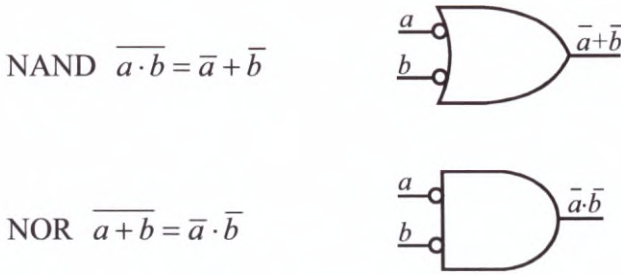
Rys. 1.4. Podział elementów logicznych

Wraz z wprowadzeniem do sterowania urządzeń programowalnych, tj. sterowników PLC, komputerów sterujących itp. coraz częściej funkcje logiczne realizuje się w sposób programowy.

Podstawowe symbole graficzne elementów logicznych (funktorów) w realizacji bezstykowej wg normy ANSI, używane często w literaturze, zamieszczono poniżej.

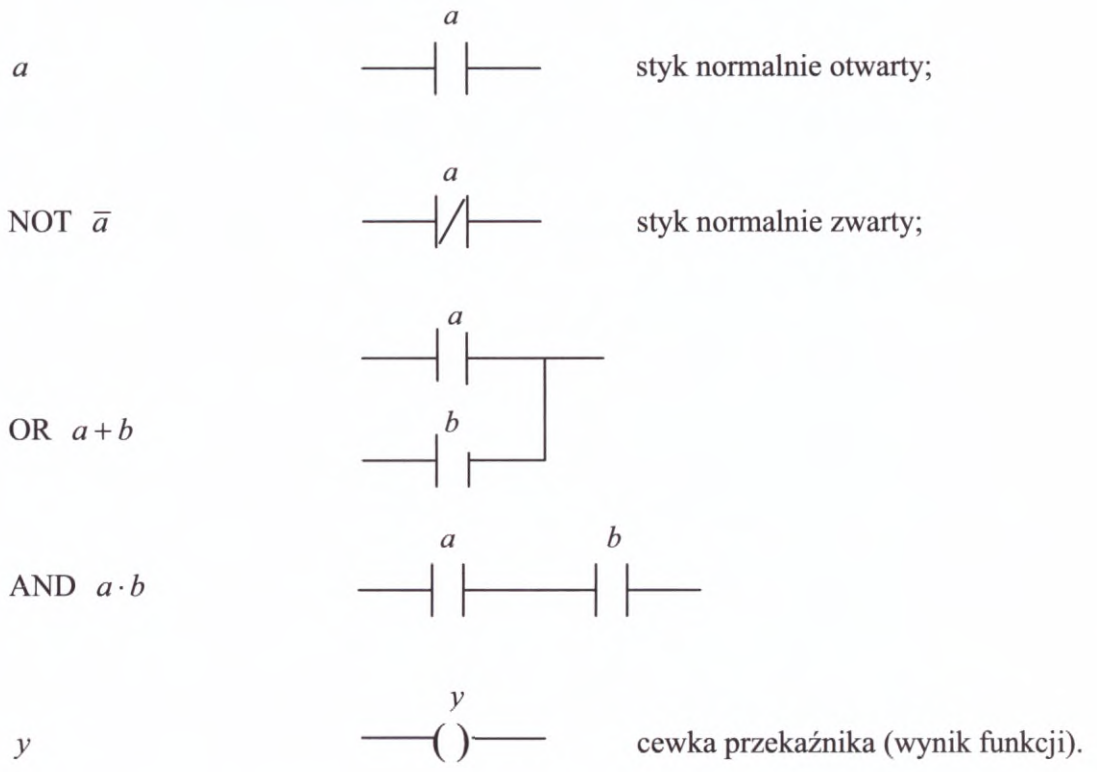


Korzystając z praw de Morgana funkcje NAND i NOR można zapisać w inny sposób i wówczas zmieni się także symbol funkcji:



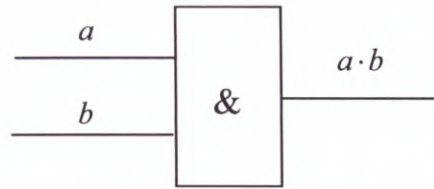
Przyglądając się prezentowanym symbolom można zauważyć, że kółko przed symbolem neguje wejście, natomiast za symbolem wyjście.

Symbole graficzne używane do przedstawiania elementów logicznych w realizacji stykowej to:



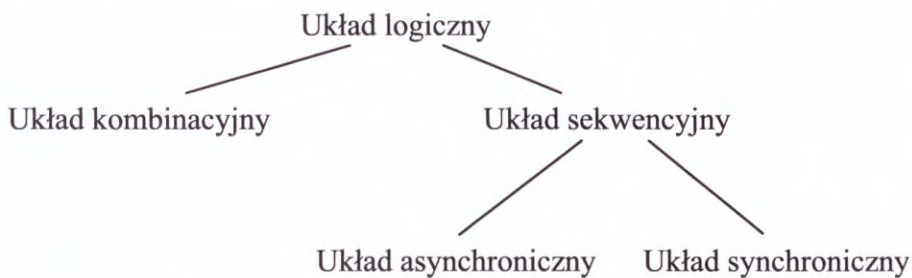
Używając tych oznaczeń można przedstawić zarówno funkcje NOR i NAND jak i inne, bardziej złożone.

Obecnie funkcje logiczne przedstawia się także przy pomocy symboli w postaci bloków funkcyjnych (PN-IEC 617-12+A1:1994 Symbole graficzne stosowane na schematach) np.

AND $a \cdot b$ 

1.5. Układy logiczne

Układ logiczny jest to układ, w którym zarówno sygnały wejściowe jak i wyjściowe przyjmują dwie wartości, umownie oznaczane jako 0 i 1. Dzieli się one na **kombinacyjne** i **sekwencyjne (z pamięcią)** – rys.1.5.



Rys. 1.5. Podział układów logicznych

Układ kombinacyjny układ logiczny, dla którego każda kombinacja sygnałów wejściowych (stan wejść) określa jednoznacznie kombinację sygnałów wyjściowych (stan wyjść).



Rys. 1.6. Schemat układu kombinacyjnego

Strukturę i działanie takiego układu (rys. 1.6.) opisuje zespół funkcji wyjścia:

$$y_i = f_i(x_1, x_2, \dots, x_n)$$

gdzie:

$$x_j, y_i \in \{0,1\} ; i = 1,2,\dots,m ; j = 1,2,\dots,n$$

Układ sekwencyjny układ logiczny, który posiada przynajmniej jeden stan wejść, któremu odpowiada kilka różnych stanów wyjść. To, który z tych stanów, dla zadanego stanu wejść, pojawi się na wyjściu układu zależy od poprzednich stanów wejść, tzn. od kolejności zmian sygnałów wejściowych.

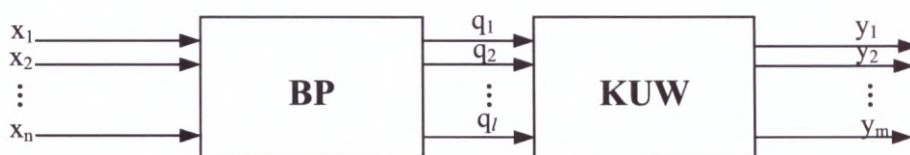
Wyróżnia się układy sekwencyjne:

asynchroniczne układy sekwencyjne, w których zmiany stanów wewnętrznych układu mogą występować w dowolnych chwilach czasu określonych przez zmiany stanu wejść tego układu;

synchroniczne układy sekwencyjne, w których zmiany stanów wewnętrznych mogą występować tylko w ściśle określonych, dyskretnych chwilach czasu $0, 1, 2, \dots$ wyznaczonych zmianą dodatkowego sygnału taktującego.

W układach sekwencyjnych można z reguły wydzielić dwa bloki: **blok pamięci (BP)** i **kombinacyjny układ wykonawczy (KUW)** – rys.1.7. W zależności od ich wzajemnych powiązań otrzymuje się:

1. Układ Moore'a (rys. 1.7.)



Rys. 1.7. Struktura układu sekwencyjnego Moore'a

Działanie układu Moore'a opisane jest przez funkcje wyjścia:

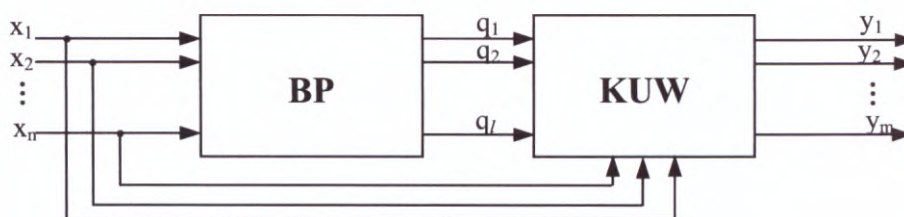
$$y_i = f_i(q_1, q_2, \dots, q_l); \quad i=1, 2, \dots, m$$

i dodatkowo przez funkcje przejścia:

$$q_i = f_i(x_1, x_2, \dots, x_n); \quad i=1, 2, \dots, l$$

przy czym funkcje wyjścia zależą wyłącznie od aktualnych wartości funkcji przejścia.

2. Układ Mealy'ego (rys. 1.8.)



Rys. 1.8. Struktura układu sekwencyjnego Mealy'ego

Działanie układu Mealy'ego opisane jest także przez funkcje wyjścia:

$$y_i = f_i(x_1, x_2, \dots, x_n, q_1, q_2, \dots, q_l); \quad i=1, 2, \dots, m$$

i przez funkcje przejścia:

$$q_i = f_i(x_1, x_2, \dots, x_n); \quad i=1, 2, \dots, l$$

lecz funkcje wyjścia zależą zarówno od aktualnych wartości funkcji przejścia jak i aktualnych wartości sygnałów wejściowych.

Powyższy podział układów logicznych (w szczególności sekwencyjnych) jest dość ściśle związany z klasycznymi metodami syntezy, prezentowanymi między innymi w podanej literaturze, dążącymi do uzyskania tzw. **postaci minimalnej** funkcji logicznej. Przy programowaniu sterowników PLC mogą być natomiast wykorzystywane zarówno metody klasyczne, jak i inne, wykorzystujące specyfikę układów programowalnych – np. formalizm SFC.

1.6. Systemy liczbowe

Podstawowe systemy liczbowe to:

- system binarny;
- dziesiętny (decymalny);
- szesnastkowy (heksadecymalny).

Są to **systemy pozycyjne**. Do podstawowych cech charakterystycznych systemu liczbowego zaliczamy:

- cyfry;
- podstawę systemu;
- wagę pozycji cyfry.

Własności systemów pozycyjnych są następujące:

- liczba cyfr jest równa podstawie systemu;
- wartość największej z cyfr jest mniejsza od podstawy o 1;
- każdej pozycji w zapisie liczby przyporządkowana jest waga wynikająca z numeru pozycji k oraz podstawy systemu p i wynosi p^{k-1} np. $k=3, p=10, p^{k-1}=100$;
- wartość liczby obliczana jest jako suma wartości cyfr pomnożonych przez wagę pozycji, które te cyfry przyjmują.

Przykład

Zapisać liczbę dwieście osiem w systemie dziesiętnym, binarnym i szesnastkowym.

System dziesiętny

cyfry	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
podstawa systemu	10
waga pozycji cyfry	potęga liczby 10: 1, 10, 100, 1000, ...

$$\begin{array}{r}
 8 \quad \times \quad 1 \quad = \quad 8 \\
 0 \quad \times \quad 10 \quad = \quad 0 \\
 2 \quad \times \quad 100 \quad = \quad 200 \\
 \hline
 208
 \end{array}$$

System binarny

cyfry	0, 1
podstawa systemu	2
waga pozycji cyfry	potęga liczby 2: 1, 2, 4, 8, 16, ...

0	x	1	=	0
0	x	2	=	0
0	x	4	=	0
0	x	8	=	0
1	x	16	=	16
0	x	32	=	0
1	x	64	=	64
1	x	128	=	128

				208

Zapis liczby 208 w systemie binarnym: 11010000

System szesnastkowy

cyfry	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
podstawa systemu	16
waga pozycji cyfry	potęga liczby 16: 1, 16, 256, 4096, ...

0	x	1	=	0
D	x	16	=	208

				208

Zapis liczby 208 w systemie szesnastkowym: D0

Często na końcu zapisu liczby w systemie szesnastkowym dodaje się literę H i wtedy zapis liczby 208 wygląda następująco: D0H.

Poza omówionymi systemami liczbowymi stosuje się niekiedy zapis liczb w tzw. kodzie BCD (w sterownikach GE Fanuc jest to BCD-4 tj. Four-Digit Binary Coded Decimal – czterocyfrowa liczba dziesiętna zakodowana w formacie BCD). Każda z cyfr dziesiętnych liczby przedstawiona jest za pomocą czterech cyfr binarnych.

Cyfra dziesiętna	Reprezentacja BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Zapis cyfry 208 w kodzie BCD-4 jest zatem następujący: 0000 0010 0000 1000

Szczegółowe informacje na temat układów liczbowych, kodów i sposobów przeliczania liczb z jednego systemu na drugi zainteresowany Czytelnik znajdzie między innymi w pozycji [2] ze spisu literatury.

Literatura

- [1] Bielecki A., Krzyżak W., Podstawy automatyki - ćwiczenia laboratoryjne, Wydawnictwo Politechniki Wrocławskiej, Wrocław, 1992.
- [2] Horowitz P., Hill W., Sztuka elektroniki. Część 2, WKŁ, Warszawa 1995.
- [3] Kamionka-Mikuła H., Małyśiak H., Pochopień B., Układy cyfrowe. Teoria i przykłady, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Wydanie II poszerzone, Gliwice, 2000.
- [4] Krakowiak S., Wprowadzenie do techniki automatyzacji, PWN, Warszawa, 1989.
- [5] Maczyński K., Kłosiński J., Pikoń Z., Suwaj S., Podstawy automatyki w przykładach i ćwiczeniach laboratoryjnych, Skrypty dla szkół wyższych, Łódź, 1989.
- [6] Maczyński K., Pikoń Z., Suwaj S., Podstawy automatyki, Materiały pomocnicze, Politechnika Łódzka Filia w Bielsku-Białej, Instytut Mechaniczno-Konstrukcyjny, Bielsko-Biała, 1976.
- [7] Małyśiak H., Kamionka-Mikuła H., Pochopień B., Wróbel E., Teoria automatów cyfrowych, Laboratorium, Wydawnictwo Politechniki Śląskiej, Gliwice, 2001.
- [8] PN -88 M-42000: Automatyka i pomiary przemysłowe. Terminologia.
- [9] PN-IEC 617-12+A1:1994 Symbole graficzne stosowane na schematach. Elementy logiczne binarne.
- [10] Stefański T., Teoria sterowania. Część II. Układy niedyskretne, nieliniowe, procesy stochastyczne oraz optymalizacja statystyczna i dynamiczna, Politechnika Świętokrzyska, Skrypt, Kielce, 1996.
- [11] Węgrzyn S., Podstawy automatyki, PWN, Warszawa, 1972.

2. Sterowniki PLC – omówienie systemu

Na świecie istnieje wielu producentów sterowników programowalnych PLC. Do największych firm tej branży, które narzucają światowe standardy związane z ich produkcją, budową i działaniem należą: Allen-Bradley, GE Fanuc Automation, Omron, SIEMENS, Modicon, Mitsubishi, Saia. Ponieważ budowa, sposób działania, funkcjonalność sterowników różnych firm jest podobna, w niniejszej publikacji ich prezentacje oparto o urządzenia i rozwiązania jednej firmy – GE Fanuc Automation.

2.1. Rys historyczny

Skrót PLC (Programmable Logic Controller) oznacza programowalny sterownik logiczny. Ponieważ nazwa ta nie odpowiada w pełni możliwościom współczesnych sterowników, często używa się obecnie określenia – sterownik swobodnie programowalny. Początki sterowników PLC sięgają końca lat 60-tych dwudziestego wieku. W 1968 roku w firmie General Motors powołano grupę inżynierów mającą opracować nowe rozwiązanie sterowania przemysłowymi układami sekwencyjnymi. Grupa ta pracowała pod kierownictwem Udo Struga. Układy sterowania logicznego, w tym także sterowania sekwencyjnego, realizowano w tamtych czasach (w przypadku prostych układów niekiedy również obecnie) za pomocą fizycznych elementów logicznych – najczęściej stykowych. Tego typu rozwiązania nazywa się czasami układami z „logiką zaszytą” lub „zadrutowaną”. Mają one szereg wad. Jedną z nich, szczególnie w układach złożonych gdzie występuje kilkadziesiąt lub kilkaset styków, jest znaczna awaryjność, inną mała elastyczność tzn. nawet niewielka zmiana w algorytmie sterowania powoduje konieczność tworzenia układu sterowania od podstaw. Ponadto charakteryzują się długim czasem zadziałania, dużym poborem energii, nagrzewaniem się podczas pracy, dużymi rozmiarami i pracochłonnością podczas tworzenia układu. Niekorzystny wpływ na jakość sterowania, zwłaszcza w układach sekwencyjnych, ma także zjawisko „hazardu” polegające na możliwości pojawienia się „fałszywych” sygnałów, a spowodowane nierównoczesnym przełączaniem się wszystkich styków związanych z danym sygnałem. Nowo opracowywane rozwiązanie tj. sterownik PLC miało być tych mankamentów pozbawione. U jego podstaw legły następujące założenia:

1. łatwość programowania;
2. możliwość szybkiej zmiany programu;
3. łatwość montażu i naprawy;
4. mniejsze wymiary niż zastępowane układy w „logice zaszytej”;
5. większa niezawodność w warunkach przemysłowych;
6. rozsądny koszt układu.

Ad. 1.

Istota nowego układu sterowania polegała na zastąpieniu sztywnej sieci połączeń elementów logicznych przez program sterujący wpisany do pamięci sterownika. Program ten wykonywany miał być cyklicznie przez specjalny procesor. Programowanie sterownika miało odbywać się przy pomocy języka programowania zorientowanego na problemy sterowania, ze szczególnym uwzględnieniem operacji bitowych. Język ten powinien przy tym być „przyjazny” dla użytkownika, łatwy do opanowania dla inżyniera dobrze znającego sterowany proces. Między innymi ta cecha miała odróżniać sterowniki PLC od komputerów sterujących programowanych z użyciem języków wyższego rzędu (np. assemblera) i wymagających

zatrudnienia wysokiej klasy informatyka. W przypadku sterowania w oparciu o klasyczny komputer przemysłowy cena oprogramowania dochodzi nieraz nawet do 90% kosztów wdrożenia całego systemu.

Ad. 2.

Jedną z istotnych cech sterowników miała być możliwość jego szybkiego przeprogramowania (najlepiej przy bezpośredniej współpracy ze sterowanym obiektem) stosownie do zmieniających się warunków i realizowanych zadań sterowania.

Ad. 3.

Sterowniki powinny umożliwiać łatwy montaż w bezpośrednim sąsiedztwie sterowanego obiektu. Aby zapewnić możliwie krótki czas usuwania ewentualnych awarii, naprawa odbywać powinna się drogą szybkiej wymiany uszkodzonego elementu (plug-in modules). Stąd też, realizowana także dzisiaj w przypadku sterowników średnich i dużych mocy, koncepcja modułowej budowy.

Ad. 4.

To założenie należy rozpatrywać uwzględniając uwarunkowania techniczne końca lat 60-tych XX wieku. Ponieważ były to początki maszyn cyfrowych, urządzenia programowane posiadały znaczne rozmiary. Obecnie, na skutek zastosowania układów scalonych, a związku z tym niewielkich wymiarów elementów elektronicznych, problem wielkości sterowników nie ma w większości zastosowań przemysłowych istotnego znaczenia.

Ad. 5.

Ponieważ sterowniki miały być instalowane w bezpośrednim sąsiedztwie sterowanego obiektu zatem mogły być narażone na niekorzystny wpływ szeregu czynników (warunki przemysłowe). W zależności od miejsca ich montażu mogły to być: wibracje, zapylenie, niskie lub wysokie temperatury, znaczne jej wahania, duża wilgotność itp. Równocześnie, aby wyeliminować jedną z podstawowych wad układów w „logice zaszytej”, sterowniki musiały w tych warunkach zapewnić większą niezawodność niż układy przekaźnikowe.

Ad. 6.

Aby osiągnąć powodzenie komercyjne należało wziąć pod uwagę także aspekt ekonomiczny. Układ sterujący opierający się o zastosowanie sterowników powinny mieć zbliżony koszt do układów przekaźnikowych realizujących analogiczne zadania sterowania.

Pierwsze sterowniki realizowały zadania sterowania logicznego i stąd ich nazwa PLC - Programmable Logic Controller. Na początku lat siedemdziesiątych stały się integralną częścią systemów automatycznego sterowania. Ich celem, zgodnie z przedstawionymi założeniami, było zastąpienie układów przekaźnikowych, głównie w sterowaniu sekwencyjnym typu bębnowego lub krzywkowego. Początkowo znalazły zastosowanie w przemyśle motoryzacyjnym, skąd stopniowo były adoptowane do innych gałęzi przemysłu i dziedzin sterowania (zarządzanie budynkami, ochrona obiektów itp.). Rozwój sterowników PLC następował równoległe z rozwojem maszyn liczących, w tym komputerów, i związany był głównie z postępowaniem w elektronice oraz informatyce (sieci komputerowe). Nieustannie rosły ich możliwości; wprowadzano coraz to nowe funkcje programowe takie jak: operacje matematyczne, czasomierze, liczniki, funkcje operujące na słowach bitowych. Oprócz przetwarzania sygnałów dwustanowych wyposażono je w możliwość przetwarzania sygnałów analogowych (przetworniki AC/DC i DC/AC). Pozwoliło to na realizację przez sterowniki PLC np. funkcji regulatora PID. Mogą służyć kontroli i rejestracji danych, niektóre zostały wyposażone w zegar czasu rzeczywistego. Bardzo rozbudowano są ich możliwości

komunikacyjne – mogą pracować niemal we wszystkich standardowych, komputerowych sieciach polowych typu LAN oraz w sieci Internet. Dzięki temu często wykorzystywane są w rozproszonych systemach sterowania. Obecnie produkowane sterowniki PLC posiadają, w zależności od typu sterownika, od kilkudziesięciu do nawet czterystu predefiniowanych funkcji. Wszystko to sprawia, że nazwa sterownik PLC nie odpowiada możliwościom współczesnych konstrukcji. Zachowała się ona jednak do dzisiaj i określa bardzo szeroki wachlarz urządzeń – od przeznaczonych do sterowania małymi obiektami (pojedyncze maszyny) do urządzeń (lub ich sieci) sterujących całymi zakładami – posiadających jednak wspólną „filozofię” ich programowania i działania.

Istotne znaczenie w rozwoju sterowników PLC miało wyposażenie ich w 1976 w kasety oddalone (kasety rozszerzające, remote I/O racks). Umożliwiły one sterownikom, przy wykorzystaniu prostych połączeń komunikacyjnych, obsługę dużej liczby wejść/wyjść przy odległościach sięgających kilkuset metrów od jednostki centralnej. W 1977 firma Allen-Bradley Corporation zastosowała po raz pierwszy mikroprocesor 8080 i dodatkowo koprocesor logiczny (dla realizacji operacji bitowych). W latach osiemdziesiątych zaczęto stosować tzw. moduły dodatkowe (inteligentne), które dzięki zastosowaniu w nich osobnych procesorów mogą realizować skomplikowane operacje i obliczenia. Niezwykle ważne było wprowadzenie na rynek, przez niektóre czołowe firmy, w 1983 roku sterowników kompaktowych. Były to małe sterowniki o ograniczonej liczbie punktów wejść/wyjść, posiadające jednak znaczne możliwości funkcjonalne przy znacznie niższych cenach –



Rys. 2.1. Przykład sterownika kompaktowego

rys. 2.1. Między innymi ich pojawienie się przyczyniło się do wielkiego wzrostu popularności sterowników PLC we współczesnych układach sterowania – w połowie lat osiemdziesiątych roczny wzrost sprzedaży sterowników w Niemczech przekraczał 20%. Lata dziewięćdziesiąte to gwałtowny rozwój tzw. modułów inteligentnych (dodatkowych, specjalizowanych). Wyposażone są one we własne układy mikroprocesorowe, które pozwalają na realizację nieraz bardzo złożonych algorytmów przetwarzania sygnałów, zapewniają zaawansowaną komunikację itp. Są one niejednokrotnie dedykowanymi układami sterowania, które w prosty sposób można połączyć ze sterownikiem.

Obecnie producenci oferują najczęściej całe rodziny sterowników. Charakterystyczne jest to, że sterowniki należące do danej rodziny:

- są programowane przy użyciu tego samego języka (języków) z użyciem tego samego programu narzędziowego i ewentualnie programatora;
- program może być przenoszony z jednego sterownika rodziny na drugi tzn. program przeznaczony na sterownik „słabszy” będzie działał także na sterowniku „mocniejszym” – na odwrót nie zawsze, gdyż niektóre funkcje zawarte w programie mogą wymagać odpowiedniej („mocniejszej”) jednostki centralnej;
- posiadają takie same zmienne programowe (sterowniki „mocniejsze” mogą mieć poszerzony ich zakres);
- moduły są wymienne tzn. moduł przeznaczony na sterownik „słabszy” będzie działał także na sterowniku „mocniejszym” – na odwrót nie zawsze, gdyż niektóre moduły dodatkowe wymagają odpowiedniej jednostki centralnej.

Często producenci zapewniają także możliwość przenoszenia programów pomiędzy swoimi sterownikami należącymi do różnych rodzin, oraz programowanie ich przy użyciu tych samych narzędzi.

Niekiedy można spotkać się z podziałem sterowników na:

- małe (do 128 wejść/wyjść dwustanowych) – najczęściej są to sterowniki kompaktowe – przeznaczone do obsługi pojedynczej maszyny lub niewielkich węzłów technologicznych;
- średnie (128-512 wejść/wyjść dwustanowych) – przeznaczone do obsługi pojedynczego węzła technologicznego, prostej linii produkcyjnej itp., zazwyczaj o budowie modułowej;
- duże (powyżej 512 wejść/wyjść dwustanowych) – przeznaczone do obsługi całej linii produkcyjnej, fragmentu zakładu zawierającego kilka węzłów technologicznych, często wykorzystywane jako sterowniki nadrzędne w sieci zbudowanych ze sterowników średnich i małych, o budowie modułowej (rys. 2.2.).



Rys. 2.2. Przykład dużego sterownika o budowie modułowej

W połowie lat osiemdziesiątych zaczęły powstawać systemy monitoringu i wizualizacji oparte o komputery klasy PC tzw. systemy SCADA (Supervisory Control and Data Acquisition). Połączenie tych dwóch rozwiązań: tj. sterowników PLC i systemów SCADA stworzyło projektantom doskonale narzędzie do konstruowania zdecentralizowanych (rozproszonych) systemów sterowania.

2.2. Zalety sterowników PLC

Poniżej krótko omówiono niektóre z zalet sterowników PLC. Nie zostały one zestawione wg kryterium ważności ponieważ cecha, która w pewnych warunkach ma decydujące znaczenie, w innych może być nieistotna i na odwrót.

Szeroki zakres zastosowań. Początkowo sterowniki PLC były stosowane w układach odznaczających się znacznym stopniem złożoności, przetwarzających dużą liczbę sygnałów logicznych. W tych układach najbardziej widoczne były wady układów przekąźnikowych o „logice zaszytej”, a równocześnie tylko tam możliwe było spełnienie kryterium ekonomicznego o porównywalnym koszcie obu rozwiązań. Na skutek malejących na

przeobrażeniu lat cen, sterowniki PLC znajdowały powoli zastosowanie również w coraz to prostszych układach. Pojawienie się w latach osiemdziesiątych sterowników kompaktowych sprawiło, że obecnie ich zastosowanie jest opłacalne nawet do sterowania pojedynczym urządzeniem (tam gdzie tradycyjny układ sterowania zawierałby kilka czy kilkanaście elementów logicznych). Ceny współczesnych sterowników kompaktowych zaczynają się bowiem już w okolicach 500 zł.

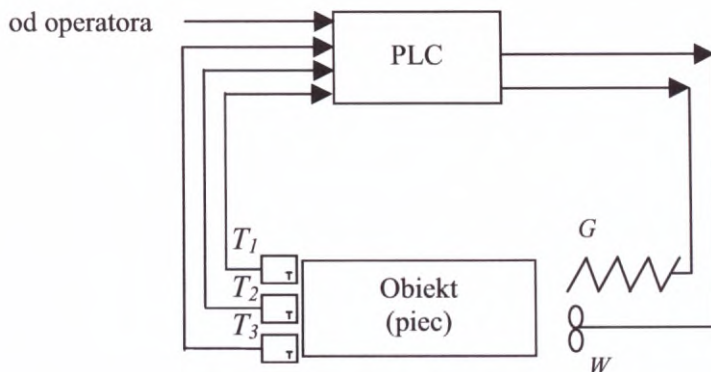
Szybkość przetwarzania. Dzięki zastosowaniu specjalizowanych mikroprocesorów (często uzupełnionych koprocesorem operacji bitowych oraz niekiedy koprocesorem matematycznym) sterowniki reagują znacznie szybciej na sygnały z obiektu niż układy tradycyjne. Czas reakcji liczy się zazwyczaj w mikrosekundach.

Koszt oprogramowania. Ze względu na prosty sposób programowania sterowników realizacja nawet skomplikowanych algorytmów sterowania nie jest związana z dużymi kosztami. Koszt całego systemu zależy głównie od liczby i rodzaju czujników oraz elementów wykonawczych, w drugim rzędzie od zastosowanych rozwiązań sprzętowych układu sterującego, a tylko w niewielkim stopniu od złożoności samego programu sterującego.

Modułowość. Cecha ta dotyczy obecnie części sterowników małych, oraz generalnie wszystkich średnich i dużych. Zapewnia ona tzw. „skalowalność” systemu, czyli możliwość jego dostosowania pod względem sprzętowym do wymagań stawianych przez zadanie sterowania. Dzięki ofercie wielu modułów jednostek centralnych, modułów wejść/wyjść różniących się od siebie zarówno liczbą punktów jak i parametrami elektrycznymi, modułów dodatkowych, które można zestawiać ze sobą w różne konfiguracje, istnieje możliwość dopasowania sterownika do konkretnego obiektu. Konfiguruje się sterownik z odpowiednią jednostką centralną (zapewniającą wymagane własności użytkowe), z pamięcią o wystarczającej pojemności dla programu sterującego i danych, z modułami wejść/wyjść zapewniającymi najwygodniejsze podłączenie wszystkich sygnałów – dopasowane do nich pod względem elektrycznym. Modułowość zapewnia także wygodną rozbudowę systemu w przyszłości. Łączy się też bardzo ściśle z kolejną zaletą PLC.

Wygoda obsługi. W przypadku awarii istnieje możliwość szybkiej wymiany uszkodzonego modułu na nowy. Operacji tej dokonuje się najczęściej bez użycia jakichkolwiek narzędzi dzięki zatraskowemu połączeniu modułów między sobą (np. sterowniki VersaMax firmy GE Fanuc Automation), lub z kasetą montażową. Bogata i przejrzysta dokumentacja tworzona automatycznie przez system podczas programowania także wpływa na wygodę obsługi. Sterowniki wyposażane są w rozbudowane układy diagnostyczne kontrolujące poprawność pracy poszczególnych podzespołów urządzenia, umożliwiającym śledzenie w trybie „on-line” procesu wykonywania programu, obserwację stanu zmiennych itp.

Przejrzystość struktury sterowania. Bez względu na realizowany algorytm sterowania wszystkie połączenia elektryczne pomiędzy sterownikiem a warstwą obiektową (czujnikami i urządzeniami wykonawczymi) pozostają takie same. Przebieg sterowanego procesu zależy jedynie od programu zawartego w pamięci sterownika. W strukturze układu sterowania wyróżnić można sterowany obiekt z oprzyrządowaniem pomiarowym i elementami wykonawczymi oraz sterownik PLC. Wszystkie czujniki zamocowane na obiekcie i przyciski z pulpitu operatora są wejściami sterownika, natomiast wszystkie elementy wykonawcze są jego wyjściami. Na rys. 2.3. przedstawiono strukturę sterowania na przykładzie pieca wyposażonego w trzy czujniki temperatury $T1-T3$, grzałkę G i wentylator W .



Rys. 2.3. Przykład struktury sterowania

Elastyczność. Z uwagi na fakt, że algorytm sterowania zapisany jest w pamięci sterownika w postaci programu, istnieje możliwość szybkiej jego modyfikacji lub podmiany. W wielu rozwiązaniach zmiany takie możliwe są nawet podczas normalnej pracy sterownika (w trybie RUN). Dzięki temu łatwo można dostosować algorytm do zmieniających się wymagań odnośnie sterowanego procesu.

Możliwość przetwarzania znacznej liczby sygnałów. Zastosowanie jednego dużego sterownika pozwala na podłączenie kilkuset (kilku tysięcy) sygnałów dwustanowych, natomiast sieć, oparta już o średnie sterowniki, umożliwia przetwarzanie kilkudziesięciu tysięcy sygnałów dwustanowych. Tak znaczne liczby sygnałów występują w niektórych dużych instalacjach np.: do sterowania liniami produkcyjnymi Opla w fabryce General Motors w Eisenach wykorzystywane są sieci sterowników obsługujące 30 do 50 tysięcy sygnałów, w układzie zarządzającym wieżowcem Petronas Tower w Kuala Lumpur także wykorzystuje się około 30 tysięcy sygnałów (czujniki ruchu, dymu, gazu, urządzenia alarmowe itd.).

Dokładność sterowania. Wspomniana poprzednio szybkość przetwarzania danych oraz możliwość analizy znacznej liczby sygnałów stwarzają możliwość uzyskania dużej dokładności sterowania. Dotyczy to w szczególności takich wielkości jak: temperatura, natężenie przepływu, czas, ciśnienie itp. W układach, gdzie dokładność sterowania jest cechą szczególnie pożądaną, sterowniki PLC wydają się być niezastąpione.

Niezawodność. Jeśli liczbę występujących w całym układzie sterowania (patrz rys. 2.3.) błędów i awarii w funkcjonowaniu systemu określimy jako 100% to, jak wykazują doświadczenia praktyczne, ich liczba przypadająca na sterownik nie przekracza 3%. Najczęściej usterki występują w warstwie obiektowej. Ponieważ najszybciej zużywającymi się i ulegającymi najczęściej awariom są elementy ruchome, w budowie sterowników PLC ograniczono je do minimum. Praktycznie jedynymi częściami ruchomymi są styki w dwustanowych przekaźnikowych modułach wyjściowych. Należy jednak pamiętać, że tego typu moduły można zastąpić np. modułami tranzystorowymi lub tyrystorowymi. Eliminacja części ruchomych ma oczywiste konsekwencje związane z budową sterowników. Ponieważ w sterownikach PLC nie stosuje się wentylatorów, chłodzenie procesorów odbywa się tylko na drodze chłodzenia grawitacyjnego (niewymuszonego). Ogranicza to możliwość stosowania „najmocniejszych” mikroprocesorów.

Duże możliwości programowe. Współczesne sterowniki programowane są przy użyciu jednego z czterech języków dopuszczonych przez normę IEC 61131 „Programmable Controllers”. Dodatkowo istnieje możliwość zastosowania formalizmu SFC (Sequential Function Chart – schemat funkcji sekwencyjnej SFC), który został opracowany w celu programowania zadań sterowania sekwencyjnego. Niektóre ze sterowników umożliwiają,

w zależności od preferencji programisty, użycia dowolnego z tych języków, bądź nawet wykonania poszczególnych części oprogramowania w różnych językach. Jest to szczególnie przydatne przy programowaniu złożonych układów, gdy programistów jest kilku, a każdy odpowiedzialny jest za wybrany fragment aplikacji. W przypadku zastosowania dodatkowych modułów programowanych, do ich programowania używa się między innymi języków Microsoft C, C++ i Megabasic. Obecnie niektóre ze sterowników, oprócz języków wymienionych w normie IEC 61131, umożliwiają bezpośrednie programowanie w językach wyższego rzędu np. Microsoft C, C++, Visual Basic, SCL (Structured Control Language – Siemens), Turbo Pascal lub State Logic. Urządzenia firmy SIEMENS z rodziny SIMATIC S7 dostarczają możliwość użycia CFC (Continuous Function Chart) – formalizmu podobnego do SFC, a przeznaczonego do technologicznie zorientowanego opisu algorytmów ciągłych.

Duże możliwości komunikacyjne. Standardowo sterowniki wyposażone są w łącze szeregowe RS-422/485 i/lub RS-232 oraz protokoły komunikacyjne SNP Master/Slave, Modbus RTU Slave, Custom ASCII, S-BUS. Dodatkowo można zastosować szereg modułów komunikacyjnych np. przeznaczonych do sieci Ethernet TCP/IP, do sieci polowych (procesowych) – np.: GENIUS firmy GE Fanuc Automation, PROFIBUS, LonWorks. Sterowniki przystosowane są zarówno do łączności bezpośredniej, jak i modemowej oraz radiomodemowej. Możliwość podłączenia sterowników PLC do różnych standardowych przemysłowych sieci polowych typu Fieldbus (zwanymi niekiedy sieciami miejscowymi) przy wykorzystaniu standardowych protokołów komunikacyjnych sprawia, że są one idealne do zastosowania w rozproszonych systemach sterowania, w których mogą być używane urządzenia różnych producentów. Równocześnie sieci typu Fieldbus można łączyć z sieciami biurowymi i lokalnymi LAN (Local Area Network) tj. sieciami nadrzędnymi tworząc układy o różnej, dostosowanej do potrzeb sterowanego obiektu, konfiguracji.

2.3. Opis systemu

Struktura prostego układu wykorzystującego sterownik PLC została przedstawiona na rys. 2.3. Oprócz zaznaczonych na rysunku elementów w skład systemu wchodzić musi dodatkowo programator. Można zatem wyróżnić następujące elementy systemu:

1. hardware;
2. software;
3. elementy wejściowe;
4. elementy wyjściowe;
5. programator.

2.3.1. Hardware

Gdy mówimy o hardware mamy na myśli realizację sprzętową sterownika. W niniejszym paragrafie zostanie przybliżona budowa sterownika modułowego, jako konstrukcji bardziej „dojrzałej”, o większej elastyczności, a co za tym idzie większych możliwościach niż sterownik kompaktowy. Znaczną część zawartych tutaj informacji można jednak odnieść także do tych drugich, pamiętając tylko, że zamiast modułów wejść/wyjść, w konstrukcjach kompaktowych występują punkty wejść/wyjść. Główne części składowe sterownika PLC to:

1. jednostka centralna (CPU – Central Processing Unit);
2. zasilacz;

3. podstawowe moduły wejść/wyjść (I/O);
4. moduły inteligentne (dodatkowe, specjalizowane).

Ad. 1.

Jednostka centralna to przede wszystkim **procesor** i **pamięć**. To ona w głównej mierze decyduje o możliwościach sterownika, a zatem o jego przeznaczeniu.

Stosowane procesory to zarówno procesory 8-, 16- jak i 32-bitowe. Ponieważ mikroprocesory te, zwane także procesorami słownymi (bajtowymi), nie pozwalają na wystarczająco szybkie wykonywanie operacji bitowych – między innymi ze względu na konieczność maskowania rejestrów - wykorzystuje się zazwyczaj równolegle koprocesory bitowe (ISCP). Wówczas, dzięki specjalnej jednostce koordynującej, oba procesory mają dostęp do pamięci sterownika, przy czym procesor słowny wykonuje operacje: matematyczne, transformacji kodów, zliczania, przesyłania słów pomiędzy pamięcią a układami zewnętrznymi, natomiast procesor bitowy operacje logiczne oraz załącza i wyłącza wyjścia dwustanowe. W przypadku „mocnych” sterowników występują dodatkowo koprocesory matematyczne dla operacji zmiennoprzecinkowych. Jednostki centralne sterowników GE Fanuc Automation bazują głównie na mikroprocesorach firmy Intel: 16-bitowych – 80188 (8 i 10 MHz), 80C186 (12 MHz), 80C188XL (20 MHz), i 32-bitowych – 80386EX (25 MHz), 80386 (16 MHz), 80486DX (32, 64 i 94 MHz). Firma SAIA-Burgess Electronics AG stosuje 32-bitowe mikroprocesory firmy MOTOROLA 68340 i 68000, sterowniki Modicon TSX Quantum wyposażane są w mikroprocesory Intel i486DX i i586.

Pamięć sterownika to pamięć operacyjna ROM (Read-Only Memory), reprogramowalna pamięć EPROM (Flash EPROM) oraz podtrzymywana bateryjnie pamięć RAM (Random Access Memory) przeznaczona do przechowywania oprogramowania systemowego, programu użytkownika, elementów konfiguracji. Pojemności pamięci mieszczą się w przedziale kilku kB do 1MB. Niektóre rozwiązania pozwalają zwiększyć dostępną pamięć poprzez zastosowanie dodatkowych modułów pamięci.

Ad. 2.

Zasilacze sterowników zapewniają wewnętrzne zasilanie jednostki centralnej, modułów wejść/wyjść oraz modułów dodatkowych. W zależności od potrzeb mogą być zasilane ze źródeł napięcia przemiennego 120/240 V, stałego 125 V lub 24/48 V. Często są one równocześnie źródłem zasilania 24 VDC, które może być wykorzystane w obwodach zewnętrznych, zarówno wejściowych jak i wyjściowych. Zazwyczaj posiadają także baterię litową o bardzo dużej trwałości, zapewniającej podtrzymywanie danych, programu i konfiguracji zawartych w pamięci RAM w przypadku utraty zasilania zewnętrznego.

Ad. 3.

Moduły wejść/wyjść można podzielić na pięć grup:

- a. **moduły wejść dyskretnych** – przekształcają napięcie wejściowe (stałe lub przemiennie) na sygnał logiczny wykorzystywany przez sterownik. Obwody wejściowe bywają odizolowane od obwodów logicznych za pomocą optoizolatora. Zazwyczaj dostępne bywają w wersji 8-, 16- lub 32-punktowej. Specjalnym rodzajem modułów wejść dyskretnych jest moduł symulatora wejść dwustanowych, który jest szczególnie przydatny podczas testowania i uruchamiania nowych aplikacji;
- b. **moduły wyjść dyskretnych** – przekształcają sygnał wyjściowy, otrzymany ze sterownika, na napięcie (stałe lub zmienne) o określonej wartości zasilające, bezpośrednio lub przez przekaźniki, urządzenia sterowane przez sterownik. Każdy obwód wyjściowy bywa

odizolowany półprzewodnikowo. Zazwyczaj wykonane są w wersji 8-, 16- lub 32-punktowej, choć spotyka się też rozwiązania 5-, 6-, 12-punktowe;

- c. **moduły wejść analogowych** – zapewniają konwersje analogowego sygnału na odpowiadającą mu liczbę. Moduły te mogą być prądowe lub napięciowe (tzn. zmiana sygnału analogowego może być reprezentowana przez zmianę natężenia prądu lub jego napięcia). Dostępne są zwykle wersje 4- lub 8-kanalowe, choć bywają także wersje o większej liczbie kanałów np. w serii sterowników 90-30 firmy GE Fanuc Automation 16-kanalowe;
- d. **moduły wyjść analogowych** – zapewniają konwersję liczby, generowanej przez sterownik, na odpowiadający jej sygnał analogowy. Moduły te mogą być prądowe lub napięciowe. Dostępne są zwykle wersje 2-, 4- lub 8-kanalowe;
- e. **kombinowane moduły wejść i wyjść** – są to moduły dyskretne posiadające zarówno punkty wejść jak i wyjść (np. 8/8, 16/8, 16/16, 20/12), lub moduły analogowe posiadające kanały wejść/wyjść (np. 4/2).

Niektórzy z producentów oferują także tzw. moduły dyskretne programowalne. Moduł taki posiada określoną liczbę punktów, a użytkownik decyduje, które z nich będą pełniły rolę wejść, a które wyjść.

Podsumowując, można stwierdzić, że zadaniem modułów podstawowych jest dostosowanie poziomu sygnałów pomiędzy sterownikiem a obiektem, oraz zabezpieczenie sterownika przed przeciążeniami i zakłóceniami generowanymi przez obiekt. Wyposażone są najczęściej w diody typu LED wskazujące stan pracy modułu, poprawność jego konfiguracji, stan każdego z wejść/wyjść (moduły dwustanowe), informujące o awarii np. przepaleniu bezpieczników.

Ad. 4.

Moduły inteligentne wyposażone są zazwyczaj w układy mikroprocesorowe. Pozwalają na wykonywanie szeregu funkcji takich jak: realizacja złożonych procedur sterowania, przetwarzanie sygnałów, gromadzenie danych, komunikacja. Niekiedy do tej grupy zalicza się także moduły wejściowe przystosowane do bezpośredniej współpracy z termoparami, termometrami oporowymi, mostkami tensometrycznymi (moduły wagowe), które eliminują konieczność stosowania wszelkich urządzeń pośrednich (np. przetworników). Poniżej wymienione zostaną najbardziej popularne moduły tej grupy. Należy jednak pamiętać, iż producenci ciągle rozszerzają ich asortyment. Aby przybliżyć ich istotę w rozdziale 3 omówiono nieco dokładniej wybrane moduły serii 90-30 firmy GE Fanuc Automation. Oczywiście inni producenci oferują sprzęt o podobnej budowie i własnościach użytkowych.

Do najbardziej popularnych modułów inteligentnych należy zaliczyć moduły:

- wejść do pomiaru temperatury współpracujące z termoparą;
- wejść do pomiaru temperatury współpracujące z termometrem oporowym np. PT100 lub NI120;
- wagowe, współpracujące z mostkami tensometrycznymi;
- licznika impulsów o wysokiej częstotliwości;
- pozycjonujące;
- programowalne;
- komunikacyjne;

- MODBUS;
- dla łącz szeregowych RS232/485;
- Ethernet;
- GENIUS;
- DeviceNet;
- Profibus;
- głośnomówiące;
- Fuzzy Logic.

2.3.2. Software

Software to program logiczny użytkownika znajdujący się w pamięci jednostki centralnej CPU. Jak wynika z wcześniejszych rozważań istnieją bardzo duże możliwości związane z tworzeniem różnych algorytmów sterowania na sterownikach PLC. Wynika to zarówno z możliwości przetwarzania przez sterowniki sygnałów binarnych i analogowych, użycia podczas programowania szeregu funkcji, procedur, dużych możliwości komunikacyjnych, jak i z dostępności wielu języków programowania. Część z nich została opracowana specjalnie w celu ułatwienia programowania typowych procesów sterowania – formalizmy SFC, CFC. Podstawę programowania PLC stanowią języki wymienione w normie IEC 61131, przy czym najczęściej używany, najbardziej rozpowszechniony jest język drabinkowy (Ladder Diagram, LD). Wynika to z jego podobieństwa do schematów przekaźnikowych. W rozdziale 4 krótko zostały scharakteryzowane języki programowania zgodne z IEC 61131, a następnie omówiono podstawy języka LD na przykładzie oprogramowania Logicmaster 90 firmy GE Fanuc Automation. Tam też podane są typy danych i zmiennych występujące w sterownikach oraz dostępne funkcje (dla sterowników GE Fanuc serii 90-30). W rozdziale 5 znaleźć można natomiast informacje dotyczące formalizmu SFC (w oparciu o jego realizację dla sterowników GE Fanuc Automation).

2.3.3. Elementy wejściowe

Elementami wejściowymi mogą być wszelkiego typu czujniki, zarówno dwustanowe jak i analogowe, zainstalowane na obiekcie, a także dane od operatora z paneli sterujących lub z innych programów i urządzeń współpracujących ze sterownikiem czy to w ramach sieci czy systemu monitoringu. Dokładne omówienie budowy i zasad działania czujników wykracza poza ramy niniejszego skryptu. Aby uzmysłowić Czytelnikowi różnorodność elementów wejściowych poniżej wymieniono niektóre z najczęściej stosowanych:

- przyciski na panelu operatora;
- czujniki temperatury;
- czujniki poziomu;
- mierniki przepływu;
- wyłączniki krańcowe;
- czujniki wilgotności;
- czujniki ciśnienia;
- wyłączniki i przyciski zainstalowane na obiekcie;

- czujniki do kontroli położenia, odkształceń, obecności obiektów, ich grubości itp.:
 - światłowodowe;
 - fotoelektryczne (w tym także światłowodowe);
 - zbliżeniowe;
 - ultradźwiękowe;
 - laserowe;
- czujniki pH;
- czujniki siły;
- czujniki koloru;
- enkodery.

2.3.4. Elementy wyjściowe

Także w przypadku elementów wyjściowych mogą to być, w zależności od sterowanego obiektu, bardzo różne urządzenia. Dokładne ich omówienie wykracza poza ramy niniejszego skryptu. Poniżej wymieniono tylko niektóre z nich:

- sygnalizatory:
 - lampki;
 - dzwonki;
 - syreny;
- silniki;
- siłowniki;
- zawory;
- pompy;
- wentylatory;
- nagrzewnice;
- falowniki.

2.3.5. Programator

Zasadniczym zadaniem programatora jest umożliwienie pisania programu sterującego i przesyłanie go do pamięci sterownika. Służy także do uruchamiania, zatrzymania sterownika, jego konfiguracji oraz monitorowania wykonania programu. Jako programatora używa się:

1. programatora ręcznego – rys. 2.4.;
2. komputera klasy PC z odpowiednim oprogramowaniem narzędziowym;
3. stacji roboczych np. Workmaster II.

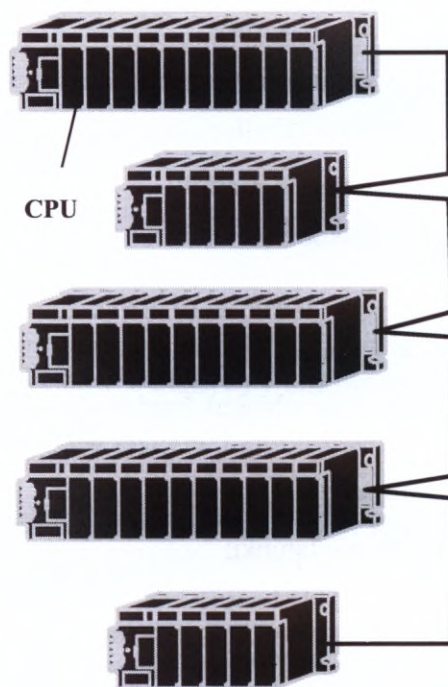


Rys. 2.4. Programator ręczny

Ze względu na wygodę najczęściej stosuje się drugie rozwiązanie. Prezentowane w rozdziale 4 funkcje programowe sterownika oparte są na oprogramowaniu narzędziowym LogixMaster 90 firmy GE Fanuc Automation przeznaczonym na komputery PC.

2.3.6. Kasety rozszerzające

Popularnym, a przy tym bardzo przydatnym rozwiązaniem jest możliwość stosowania tzw. **kaset rozszerzających** – rys. 2.5. Poprzez ich zastosowanie można, używając specjalnego okablowania, skonfigurować sterownik zabudowany na kilku kasetach. Jednostka centralna CPU znajduje się zawsze w kasecie podstawowej, w dodatkowych kasetach można umieszczać moduły wejść/wyjść lub moduły inteligentne. Pozwala to na podłączenie do sterownika znacznie większej liczby sygnałów niż wynika to z możliwości samej kasety podstawowej. Dodatkowo, ponieważ odległość między kasetami może wynosić kilkanaście metrów (w przypadku tzw. **kaset oddalonych** w sterownikach 90-30 firmy GE Fanuc Automation nawet 210 m) powstaje sterownik rozproszony. Dzięki temu moduły wejść/wyjść sterownika można umieścić bliżej sterowanego obiektu skracając znacznie długość przewodów sygnałowych. Zmniejsza to zarówno koszt okablowania, jak i zwiększa odporność układu na zakłócenia. Przy zastosowaniu specjalnego modułu nadawczego i odbiorczego kasety rozszerzające w sterownikach VersaMax firmy GE Fanuc Automation można odsunąć nawet na odległość 750 m.



Rys. 2.5. Układ kaset rozszerzających

Analogiczne rozwiązania stosuje się w przypadku sterowników kompaktowych. Kasety rozszerzające (tzw. ekspandery) zawierają dodatkowe dyskretne punkty wejść/wyjść lub kanały analogowe. Dzięki nim można w pewnym zakresie zwiększyć liczbę obsługiwanych sygnałów ponad to, co zapewnia sterownik podstawowy. Odległości, na które można odsunąć te kasety są jednak zdecydowanie mniejsze niż w sterownikach modułowych (od kilkunastu centymetrów do kilku metrów).

2.4. Układy redundancyjne

W wielu procesach, w których stosuje się sterowniki PLC, niezwykle istotny jest warunek bezpieczeństwa – niezawodności układu sterowania i jego odporności na awarie. Do tego typu instalacji zaliczyć można układy stosowane do sterowania różnego typu elektrowniami czy też niektórymi procesami chemicznymi np. przeprowadzanymi w zakładach petrochemicznych. W celu zapewnienia wymaganego poziomu bezpieczeństwa stosuje się wtedy tzw. **systemy redundancyjne** (rezerwację systemu). Rezerwacja może być oparta o program logiczny znajdujący się w sterowniku, bądź też jej podstawę stanowi kombinacja zabezpieczeń sprzętowych i programowych. Najczęściej stosuje się to drugie rozwiązanie jako stwarzające znacznie większe możliwości zabezpieczenia systemu. Są to więc specjalne układy ponadwymiarowe, wykorzystujące głównie pracę sterowników w sieci. Znane są systemy oparte na zdwojeniu urządzeń np. sterowników czy łączny komunikacyjnych, jak i, do szczególnie odpowiedzialnych zadań, systemy oparte na

potrojeniu urządzeń – jest to tzw. **redundancja TMR** (Triple Modular Redundancy). W takim układzie potrójnego rezerwowania mogą pracować niektóre ze sterowników GE Fanuc serii 90-70 (dodatek D1) przeznaczone do pracy jako układy ESD. Jest oczywiste, że systemy z rezerwacją są znacznie droższe i to zarówno ze względu na konieczność dublowania (potrójania) urządzeń, jak i opracowania znacznie bardziej skomplikowanego programu sterującego. Są jednak procesy, gdzie zaniechanie redundancji mogłoby prowadzić do tragicznych konsekwencji (przemysł chemiczny, elektrownie jądrowe). Poniżej omówiono trzy najważniejsze rodzaje systemów redundancyjnych.

ESD – Emergency Shutdown System

Jest to system awaryjnego zamykania procesu. Stosowany szeroko w przemyśle petrochemicznym, olejowym i gazownictwie (Emergency Shutdown and Fire & Gas Detection). Jego zadaniem jest bezpieczne zamknięcie cyklu produkcyjnego w sytuacji awaryjnej. System posiada wbudowane mechanizmy eliminujące zamykanie procesu w błahych przypadkach jak np. awaria czujnika. Mechanizmem tym może być przykładowo korzystanie z trzech czujników wejściowych tego samego sygnału (np. temperatury), których stany są przekazywane przez odrębne sieciowe magistrale, by następnie ustalić sterowanie jako wynik głosowania „dwa z trzech”. Przykładem takiego systemu może być oferowany przez firmę GE Fanuc modułowy system rezerwacji GMR (Genius Modular Redundancy). Najbardziej rozbudowana wersja takiego systemu posiada 3 jednostki centralne, a realizacja sterowania procesu oparta jest na analizie danych z tych 3 jednostek. Typ pracy układu konfiguruje operator. Pełny ESD korzysta z redundowanych wejść jak i magistral komunikacyjnych, oraz wyjść pracujących w trybie Fail Safe lub Fault Tolerant. Posiada certyfikat bezpieczeństwa TÜV w klasie 6 co odpowiada normie SIL 3.

Hot Standby PLC Redundancy (gorąca rezerwa)

W tym systemie występują dwie jednostki centralne. Pierwsza z nich zajmuje się kontrolą procesu, druga znajduje się w stanie oczekiwania. W momencie wystąpienia awarii w aktywnej jednostce CPU następuje przejście sterowania procesem przez jednostkę rezerwową. Podczas konfigurowania systemu określa się dokładnie warunki w jakich ma nastąpić przejście kontroli nad procesem przez jednostkę rezerwową. System Hot Standby PLC Redundancy może zostać wyposażony dodatkowo w mechanizm, który zapewnia synchronizację pracy jednostki znajdującej się w stanie oczekiwania (rezerwowej) i jednostki aktywnej. Synchronizacja ta obejmuje zarówno synchronizację cykli pracy obu jednostek CPU jak i danych. Dzięki temu możliwe jest bezuderzeniowe przejście kontroli nad procesem przez jednostkę rezerwową. System Hot Standby PLC Redundancy daje bowiem możliwość kopiowania danych procesowych w każdym cyklu pracy systemu z jednostki aktywnej do rezerwowej. Umożliwia także programowanie oraz usuwanie awarii w trybie „on-line”.

Duplex PLC Redundancy

W systemie Duplex PLC Redundancy także występują dwie jednostki centralne. Stany wejść przekazywane są jednocześnie do obu jednostek centralnych, przy czym obie realizują program sterujący. W sytuacji, gdy obydwie jednostki CPU generują jednakowe stany wyjść, taki właśnie stan przyjmują fizyczne wyjścia systemu. W sytuacji gdy sterowniki wysyłają odmienne stany wyjść, o ich ostatecznym stanie decyduje stan skonfigurowany w danym wyjściu jako standardowy – bezpieczny (Duplex Default State). System ten dotyczy układów z sygnałami dwustanowymi.

2.5. Cykl pracy sterownika

Praca sterownika odbywa się w tzw. **cyklu pracy**. Oprócz wykonania części logicznej programu sterującego, cykl pracy sterownika zawiera fazy: odczytu danych wejściowych, wysłania danych wyjściowych, komunikacji, diagnostyki. Zwykle sterowniki pracują w trybie standardowego cyklu pracy. Oznacza on, że sterownik wykonuje część logiczną programu, odczytuje i zapisuje wejścia/wyjścia (skanuje wejścia/wyjścia), a po wykonaniu jednego cyklu natychmiast przystępuje do wykonania cyklu następnego. Możliwa jest także praca ze stałym czasem trwania cyklu. W przypadku zatrzymania pracy sterownika może on skanować wejścia/wyjścia (w sterownikach GE Fanuc tryb STOP/IOSCAN) lub nie (w sterownikach GE Fanuc tryb STOP/NO IO).

W standardowym trybie pracy każdy cykl jest wykonywany tak szybko jak to możliwe. W konsekwencji, czas trwania poszczególnych cykli może się zmieniać. Istnieje jednak możliwość ustawienia sterownika na pracę ze stałym czasem trwania cyklu. Jednym z powodów dla zastosowania tego trybu może być potrzeba zapewnienia uaktualnienia stanu wejść sterownika w stałych odstępach czasu. Innym, wprowadzenie pewnego odstępu czasowego pomiędzy fazą obsługi wyjść sterownika a fazą obsługi wejść w następnym cyklu, co umożliwi ustalenie się stanu wejść po otrzymaniu danych wyjściowych z programu. Z powodu zmian czasu trwania poszczególnych faz cyklu, zaleca się ustawienia stałego czasu cyklu o co najmniej 10 ms dłuższego niż maksymalny zaobserwowany czas cyklu w trybie standardowym dla danego zadania sterowania. Zapobiega to występowaniu błędów w pracy sterownika spowodowanych przekroczeniem ustalonego czasu trwania cyklu.

Na rys. 2.6. przedstawiono, na przykładzie urządzeń GE Fanuc, cykl pracy sterowników. Poniżej omówiono poszczególne jego fazy.

Inicjacja cyklu – początek cyklu

W tej fazie sterownik wykonuje wszystkie operacje niezbędne do rozpoczęcia cyklu. W przypadku gdy sterownik ustawiony jest w trybie ze stałym czasem trwania cyklu, rozpoczęcie nowego cyklu opóźniane jest do momentu aż upłynie ustalony limit. Jeśli określony czas trwania cyklu zostanie przekroczony sterownik informuje o błędzie, a następny cykl rozpoczyna się bez opóźnienia.

Obsługa wejść

Po inicjacji cyklu sterownik odczytuje stan wejść. Kolejno odczytywane są wejścia wszystkich modułów, a ich wartości zostają zapisane w odpowiednich komórkach pamięci odpowiadających wejściom dwustanowym, analogowym, ewentualnie sygnałom przesyłanym przez sieć.

Wykonanie części logicznej programu sterującego

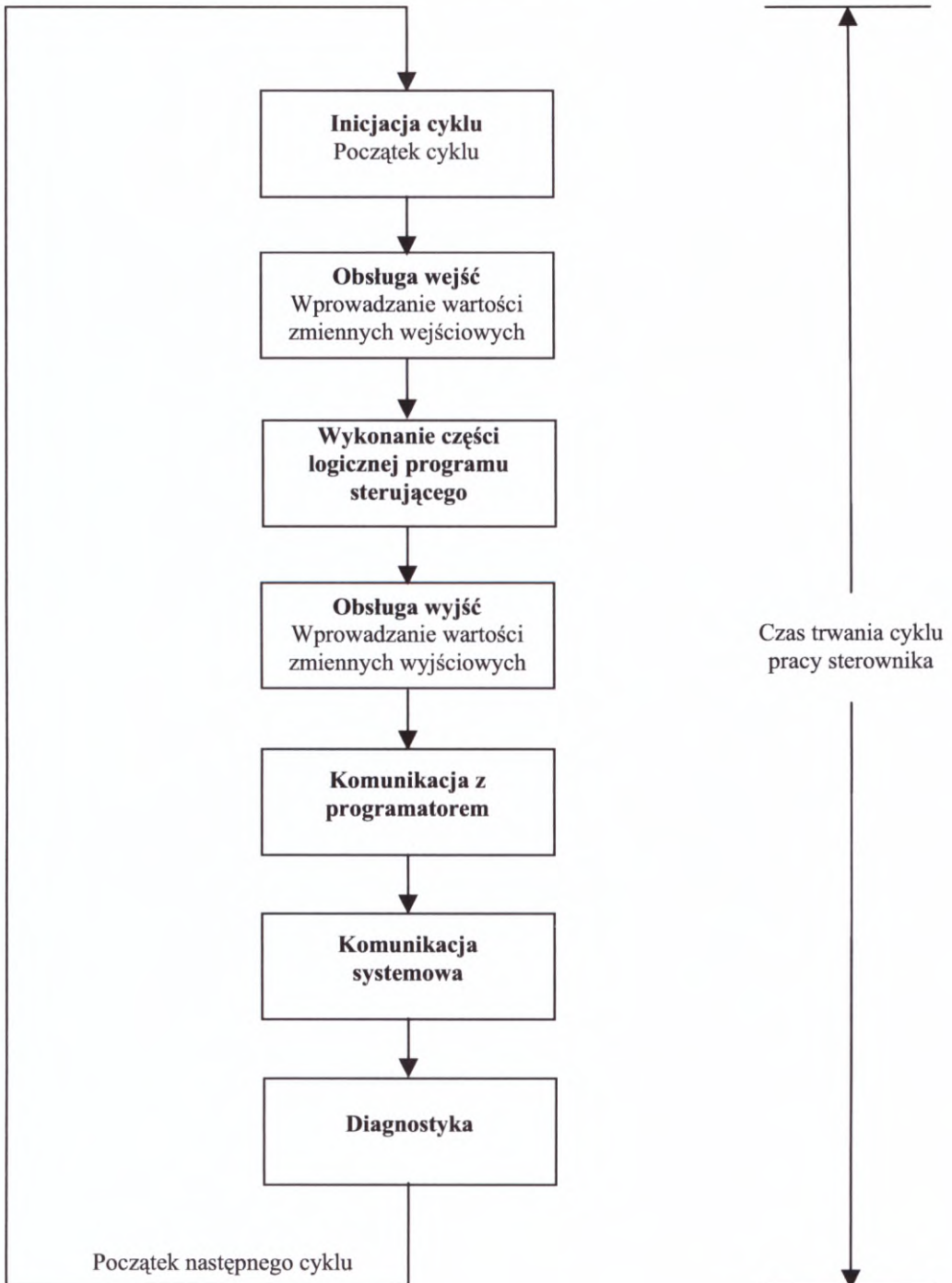
Faza ta następuje bezpośrednio po odczycie wejść. Mikroprocesor, wspólnie z opcjonalnymi koprocesorami: bitowym i matematycznym, wykonuje instrukcje zawarte w programie dla wczytanych w poprzedniej fazie danych. W wyniku tego działania, wylicza aktualne wartości sygnałów wyjściowych, które powinny zostać przekazane elementom wykonawczym.

Obsługa wyjść

W tej fazie wyliczone wartości sygnałów wyjściowych zostają przypisane fizycznym wyjściom sterownika.

Komunikacja z programatorem

Faza ta jest pomijana w przypadku gdy do sterownika nie jest podłączony programator. Natomiast, jeśli programator jest podłączony, to w jednym cyklu sterownika realizowana jest tylko jedna operacja, tzn. reaguje on na jedno naciśnięcie klawisza lub odpowiada na jedno żądanie komunikacji. Dodatkowo faza ta ograniczona jest do 6 ms, tak aby nie nastąpiło znaczne wydłużenie czasu realizacji całego cyklu. Gdy komunikacja związana z jednym żądaniem wymaga dłuższego czasu jest ona realizowana w kolejnych cyklach, lecz w żadnym jej czas nie może przekroczyć 6 ms.



Rys. 2.6. Cykl pracy sterownika

Komunikacja systemowa

Podczas fazy komunikacji systemowej realizowane są żądania komunikacji z modułami inteligentnymi. Także w tym przypadku, gdy w układzie nie są zainstalowane tego typu moduły, faza ta jest pomijana. Żądania komunikacji obsługiwane są w kolejności zgłoszeń. Żaden moduł nie jest uprzywilejowany w stosunku do innych. Dodatkowo ta faza działania programu jest ograniczona do 50 ms, tak aby nie nastąpiło znaczne wydłużenie czasu realizacji całego cyklu. Gdy komunikacja wymaga dłuższego czasu jest ona realizowana w kolejnych cyklach, lecz w żadnym jej czas nie może przekroczyć 50 ms. Czas ten może być w sposób programowy ustawiany tak aby nie wydłużał znacząco wielości skanu programu. W jednym cyklu komunikacja jest nawiązywana tylko z jednym modułem.

Diagnostyka

Na koniec każdego cyklu pracy sterownika następuje obliczenie sumy kontrolnej programu sterującego. W przypadku gdy suma kontrolna nie jest zgodna z wartością porównawczą, mikroprocesor generuje sygnał o błędzie. Gdy wystąpi błąd krytyczny sterownik jest ustawiany w tryb STOP.

W przeciętnych warunkach czas trwania pojedynczego cyklu waha się w granicach od kilku milisekund do dziesiątych części sekundy. Zależy on przede wszystkim od złożoności samego programu, zastosowanych modułów oraz oczywiście od jednostki centralnej.

2.6. Konfiguracja sterownika

Jak wynika z przedstawionych wcześniej informacji można, korzystając z dostępnych modułów, tworzyć w szerokim zakresie różne konfiguracje sprzętowe sterownika PLC. Powoduje to konieczność określenia z jakiego rodzaju elementów (jednostek CPU, modułów wejść/wyjść, modułów inteligentnych) składa się konkretny system. Jednocześnie należy przypisać im odpowiednie adresy logiczne oraz ustalić ewentualne zasady ich współpracy (wymiana danych w sieci, układy redundancyjne). Proces ten nazywamy konfigurowaniem i może on zostać zdefiniowany następująco:

Konfigurowanie proces przypisywania adresów logicznych i innych parametrów sprzętowym modułom systemu. Może on być wykonany przed lub po zaprogramowaniu sterownika.

W konfiguracji można wyodrębnić dwie części:

- konfigurację jednostki centralnej, w trakcie której przyporządkowuje się pamięć, ustawia aktualną datę i czas, określa się oznaczenie identyfikacyjne sterownika;
- konfigurację modułów wejść/wyjść (ewentualnie także modułów inteligentnych) – w jej trakcie dokonuje się opisu modułów wejść/wyjść podłączonych do sterownika, przypisuje się adresy logiczne i dokonuje wyboru opcji pracy poszczególnych modułów.

Literatura

- [1] Ciuk A., SIMATIC integruje automatykę, Technika Zagraniczna, Maszyny Technologie Materiały, Wydawnictwo SIGMA, zeszyt 2/97, str. 16 - 17
- [2] Clements-Jewery K., Jeffcoat W., The PLC Workbook. Programmable Logic Controllers Made Easy, Prentice Hall, Hemel Hempstead, 1996
- [3] Dobrzyniecki T., Sterowniki mikroprocesorowe w przemyśle, Biuletyn Automatyki ASTOR, 1/97 (11), str. 7 - 8
- [4] Faracik G., Merwart P., Rezerwacja w sterownikach programowalnych GE Fanuc, Biuletyn Automatyki ASTOR, 2/99 (20), Raport specjalny, str. VI - VII
- [5] Grandek K., Rojek R., Mikroprocesorowe sterowniki programowalne, Skrypt uczelniany nr 142, Wyższa Szkoła Inżynierska w Opolu, Opole, 1991
- [6] Katalog sterowników GE Fanuc serii 90-30 (LI-ASK-PLC-GE3), Astor, Kraków
- [7] Kwaśniewski J., Programowalne sterowniki przemysłowe w systemach sterowania, Kraków, 1999
- [8] Legierski T., Kasprzyk J., Wyrwał J., Hajda J., Programowanie sterowników PLC, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 1998
- [9] Mandado E., Marcos J., Perez S. A., Programmable Logic Devices and Logic Controllers, Prentice Hall, 1996
- [10] OMRON, ABC programowania, Materiały OMRON Electronics Sp z o.o., Warszawa, 1998
- [11] PN-89 M –2007/03 Automatyka, pomiary przemysłowe. Oznaczenia na schematach. Symbole graficzne na schematach obwodowych.
- [12] PN-IEC 61131-3:1998 Sterowniki Programowalne. Języki programowania.
- [13] Sterownik programowalne Serii 90-20 i 90-30. Podręcznik programisty, Astor, Kraków, 1993
- [14] Sterowniki programowalne, wkładka do pisma Pomiary Automatyka Kontrola, Zeszyt nr 10/1998
- [15] Tutaj R., Sterownik PLC w środowisku wielosieciowym, Pomiary Automatyka Robotyka, nr 4 (37)/2000, str. 7 - 11
- [16] Warnock I.G., Programmable Controllers, Operation and Application, Prentice Hall, Hemel Hempstead, 1998

3. Sterowniki serii 90-30 firmy GE Fanuc

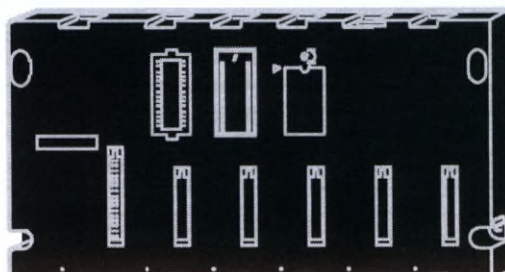
Jak wspomniano na wstępie, poniższy rozdział został poświęcony zaprezentowaniu serii modułowych sterowników 90-30 firmy GE Fanuc Automation. Powstała ona w 1989 i jest systematycznie rozwijana i unowocześniana. Sterowniki tej serii należą do najchętniej stosowanych w instalacjach automatyki na świecie, są także często używane w Polsce. Niniejsze omówienie nie jest wyczerpujące. Jego celem nie jest bowiem dokładna prezentacja wszystkich elementów wchodzących w skład serii, lecz przybliżenie Czytelnikowi zagadnień związanych z budową i możliwościami typowych sterowników. Szczególną uwagę poświęcono wybranym modułom inteligentnym (podrozdziały 3.3. - 3.7.). W osobnym – 6 rozdziale – przedstawiono przemysłową sieć GENIUS. Na podstawie omówienia wybranego modułu można wnioskować o możliwościach podobnych modułów zastosowanych w sterownikach innych producentów (jest to prezentacja zagadnienia tzw. metodą "case study").

3.1. Jednostki centralne

W drugiej połowie 2001 roku firma GE Fanuc oferowała 10 rodzajów jednostek centralnych serii 90-30. Podstawowe ich parametry zostały przedstawione w Tab. 3.1., 3.2. oraz dodatku D1 w podrozdziale D1.4.

Moduły sterowników serii 90-30 montowane są na płycie montażowej tzw. kasecie podstawowej (CPU baseplate) – rys. 3.1.

Są to płyty z przewidzianym miejscem na zamontowanie zasilacza oraz z gniazdami do instalowania modułów. Kasety podstawowe dla jednostek centralnych CPU311, CPU313 oraz CPU323 posiadają wbudowaną standardowo w płycie jednostkę centralną. Pierwsze gniazdo z lewej strony płyty służy do zamontowania modułu zasilacza. Na płycie nie ma żadnych przełączników wymagających wstępnego ustawienia. Kaseca może być 5- (CPU311, CPU313) lub 10-gniazdowa (CPU 323) tzn. przystosowana do zamontowania pięciu lub dziesięciu modułów. Podstawowe dane sterowników wyposażonych w te jednostki CPU zestawiono w tab. 3.1.



Rys. 3.1. Kaseca podstawowa – CPU baseplate

Tab. 3.1. Charakterystyka kaset z jednostkami centralnymi CPU311 oraz CPU313/323:

Parametry	CPU311	CPU313/323
Procesor	80188	80188
Zegar taktujący	8 MHz	10 MHz
Liczba punktów wejść *	160	160/320
Liczba punktów wyjść *	160	160/320
Pamięć o organizacji rejestrowej	1 kB	2 kB
Maksymalny obszar pamięci zarezerwowany dla programu sterującego napisanego przez użytkownika	6 kB	12 kB
Szybkość wykonywania programu sterującego (w którym występują jedynie styki)	18 ms/1 kB części logicznej	0,6 ms/1 kB części logicznej

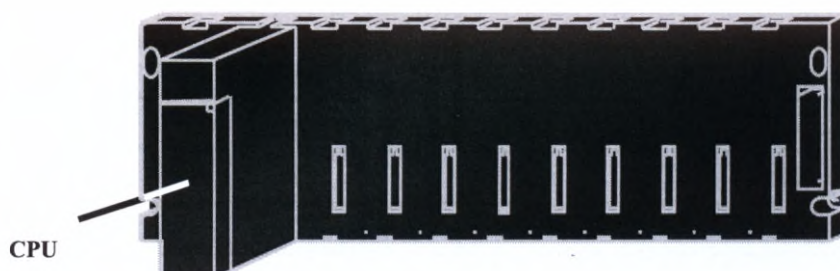
Funckje wewnętrzne	CPU311	CPU313/323
Przełączniki wyjściowe	320	320
Przełączniki wewnętrzne	1024	1024
Liczniki/przełączniki czasowe	170	170
Rejestry przemieszczające	istnieją	istnieją
Rejestry z danymi	512	1024
Wejścia analogowe (12-bitowe)	64	64
Wyjścia analogowe (12-bitowe)	32	32
Możliwość wymuszonej zmiany wartości zmiennych z blokadą	brak	brak
Żywotność baterii litowej podtrzymującej pamięć (z obciążeniem) **	2 lata	2 lata
Żywotność baterii litowej podtrzymującej pamięć (bez obciążenia) ***	8 ... 10 lat	8 ... 10 lat
Wymiary kasety 10-gniazdowej [mm]	443 x 142 x 130	
Wymiary kasety 5-gniazdowej [mm]	245 x 142 x 130	

* Całkowita liczba punktów wejść i wyjść zależy od zastosowanego typu modułów wejść/wyjść. W przypadku modułów 16-punktowych maksymalna liczba punktów $10 \times 16 = 160$ dla kasety 10-gniazdowej oraz $5 \times 16 = 80$ dla kasety 5-gniazdowej. W przypadku modułów 32-punktowych maksymalna liczba punktów to $10 \times 32 = 320$ dla kasety 10-gniazdowej oraz $5 \times 32 = 160$ dla kasety 5-gniazdowej.

** W zależności od temperatury pracy sterownika.

*** W temperaturze 25°C.

Mocniejsze jednostki centralne nie są zabudowane w kasie podstawowej lecz występują w postaci osobnego modułu, który należy podłączyć do pierwszego gniazda kasety (rys. 3.2.). Jednostki te pozwalają na obsługę 4 (CPU331, CPU341) lub 7 (od CPU 350 wzwyż) kaset rozszerzających. Podstawowe ich parametry przedstawia poniższa tabela (tab. 3.2.).



Rys. 3.2. Kaseta podstawowa z jednostką CPU w formie modułu

Tab. 3.2. Charakterystyka sterowników z jednostkami centralnymi
CPU331, CPU341, CPU350, CPU360:

Parametry	CPU331	CPU341	CPU350/352	CPU360/363/364
Procesor	80188	80C188XL	80386EX	80386EX
Szybkość	8 MHz	20 MHz	25 MHz	25 MHz
Liczba punktów wejściowych	512	512	2048	2048
Liczba punktów wyjściowych	512	512	2048	2048
Maksymalny obszar pamięci zarezerwowany dla programu sterującego napisanego przez użytkownika	16 kB	80 kB	32/240 kB zależy od ilości zadeklarowanych AI i AQ	max 240 kB zależy od ilości zadeklarowanych AI i AQ
Szybkość wykonywania programu sterującego (w którym występują jedynie styki)	0,4 ms/1 kB części logicznej	0,3 ms/1 kB części logicznej	0,22 ms/1 kB części logicznej	0,22 ms/1 kB części logicznej
Funkcje wewnętrzne				
Przełączniki wyjściowe	512	512	2048	2048
Przełączniki wewnętrzne	1024	1024	4096	4096
Liczniki/przełączniki czasowe	> 500	> 500	> 500	> 2000
Rejestry przemieszczające	istnieją	istnieją	istnieją	istnieją
Rejestry z danymi	2048	9999	9999	konfigurowalne
Wejścia analogowe (12-bitowe)	128	1024	2048/max 32000 konfigurowalne	max 32000 konfigurowalne
Wyjścia analogowe (12-bitowe)	64	256	512/max 3200 konfigurowalne	max 32000 konfigurowalne
Możliwość wymuszonej zmiany wartości zmiennych z blokadą	istnieje	istnieje	istnieje	istnieje
Arytmetyka zmiennoprzecinkowa	nie ma	nie ma	programowa/sprzętowa	programowa
Dodatkowe porty komunikacyjne	-	-	-/RS485, RS232	-/RS485, RS232/Ethernet TCP/IP
Żywotność baterii litowej podtrzymującej pamięć (z obciążeniem) *	6 miesięcy			
Żywotność baterii litowej podtrzymującej pamięć (bez obciążenia) **	8 ... 10 lat			

* W zależności od temperatury pracy sterownika.

** W temperaturze 25°C.

W celu dokładnego zapoznania się z dostępnymi jednostkami centralnymi serii 90-30 zaleca się Czytelnikowi odwiedzenie strony internetowej firmy Astor www.astor.com.pl, lub sięgnięcie do aktualnego „Katalogu sterowników GE Fanuc serii 90-30”.

3.2. Moduły wejść/wyjść

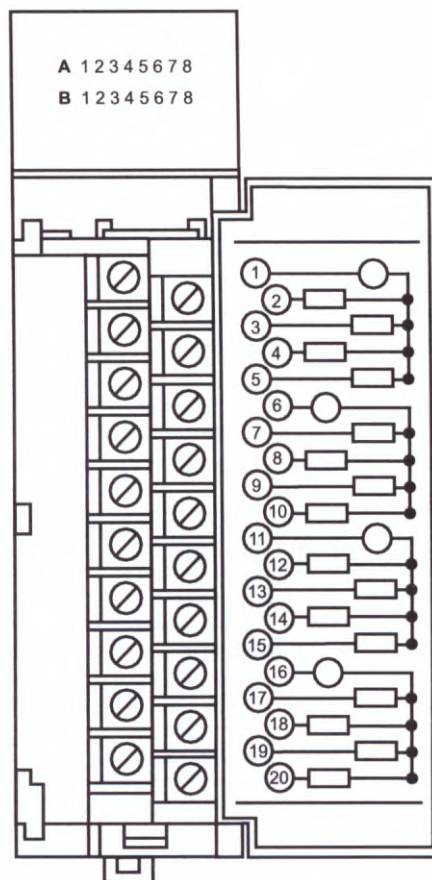
Moduły wejść/wyjść są wspólne dla całej serii 90-30 i mogą być stosowane z dowolną jednostką centralną. W ofercie występują między innymi:

1. Moduły wyjść dwustanowych (rys. 3.3.):

- moduł wejść dyskretnych, izolowanych, 8-punktowy, dla napięcia wejściowego 120 V prądu przemiennego;
- moduł wejść dyskretnych, izolowanych, 8-punktowy, dla napięcia wejściowego 240 V prądu przemiennego;
- moduł wejść dyskretnych, 16-punktowy, dla napięcia wejściowego 120 V prądu przemiennego;
- moduł wejść dyskretnych, 16-punktowy, dla napięcia wejściowego 24 V prądu stałego lub przemiennego, działający w logice dodatniej lub ujemnej;
- moduł wejść dyskretnych, 8-punktowy, dla napięcia wejściowego 125 V prądu stałego, działający w logice dodatniej lub ujemnej;
- moduły wejść dyskretnych, 8- i 16-punktowe, dla napięcia wejściowego 24 V prądu stałego, działające w logice dodatniej lub ujemnej;
- moduł wejść dyskretnych, 32-punktowy, dla napięcia wejściowego 5/12 V prądu stałego, działający w logice dodatniej lub ujemnej;
- moduł wejść dyskretnych, 32-punktowy, dla napięcia wejściowego 24 V prądu stałego, działający w logice dodatniej lub ujemnej;
- 8- lub 16-punktowy symulator wejść.

2. Moduły wyjść dwustanowych:

- moduł wyjść dyskretnych, 12-punktowy, dla napięcia wyjściowego 120 V prądu przemiennego, o natężeniu prądu do 0,5 A;
- moduł wyjść dyskretnych, 8-punktowy, dla napięć wyjściowych 120 i 240 V prądu przemiennego, o natężeniu prądu do 2 A;
- moduł wyjść dyskretnych, 16-punktowy, dla napięcia wyjściowego 120 V prądu przemiennego, o natężeniu prądu do 0,5 A;
- moduł wyjść dyskretnych, 5-punktowy, o wyjściach izolowanych, dla napięć wyjściowych 120 i 240 V prądu przemiennego, o natężeniu prądu do 2 A;
- moduły wyjść dyskretnych, 8-punktowe, dla napięć wyjściowych 12/24 V prądu stałego, o natężeniu prądu do 2 A, wykonane w logice dodatniej lub ujemnej;



Rys. 3.3. Moduł wyjść dwustanowych

- moduł wyjść dyskretnych, 6-punktowy, o wyjściach izolowanych, dla napięcia wyjściowego 125 V prądu stałego, o natężeniu prądu do 1 A;
- moduły wyjść dyskretnych, 8- i 16-punktowe, dla napięć wyjściowych 12/24 V prądu stałego, o natężeniu prądu do 0,5 A, wykonane w logice dodatniej lub ujemnej;
- moduł wyjść dyskretnych, 16-punktowy, dla napięć wyjściowych 12/24 V prądu stałego, o natężeniu prądu do 1 A, z zabezpieczeniem przed zwarcieniem;
- moduł wyjść dyskretnych, 32-punktowy, dla napięć wyjściowych 5/24 V prądu stałego (TTL), o natężeniu prądu od 5 mA do 0,5 A;
- moduł wyjść dyskretnych, 32-punktowy, dla napięć wyjściowych 12/24 V prądu stałego, o natężeniu prądu do 0,5 A;
- moduł wyjść dyskretnych, przekaźnikowych zwiernych, izolowanych, 8-punktowy, o natężeniu prądu do 4 A;
- moduł wyjść dyskretnych, przekaźnikowych rozwieranych, izolowanych, 8-punktowy, o natężeniu prądu do 8 A;
- moduł wyjść dyskretnych, przekaźnikowych zwiernych, 16-punktowy, o natężeniu prądu do 2 A;
- moduł wyjść dyskretnych, przekaźnikowych zwiernych, izolowanych, 8-punktowy, o natężeniu prądu do 8 A.

3. Moduły wejść analogowych (rys. 3.4.):

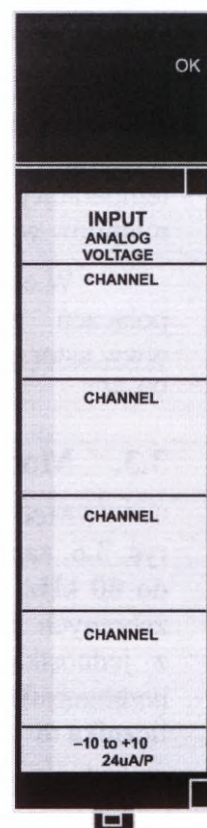
- 4-kanałowy, prądowy moduł wejść analogowych;
- 4-kanałowy, napięciowy moduł wejść analogowych;
- 16-kanałowy, napięciowy moduł wejść analogowych;
- 16-kanałowy, prądowy moduł wejść analogowych;
- 4- i 8-kanałowe moduły wejściowe do pomiaru temperatury współpracujące z termoparą;
- 6-kanałowe moduły wejściowe do pomiaru temperatury współpracujące z termometrem oporowym PT100 lub NI120;
- 4- i 8-kanałowe moduły wagowe.

4. Moduły wyjść analogowych:

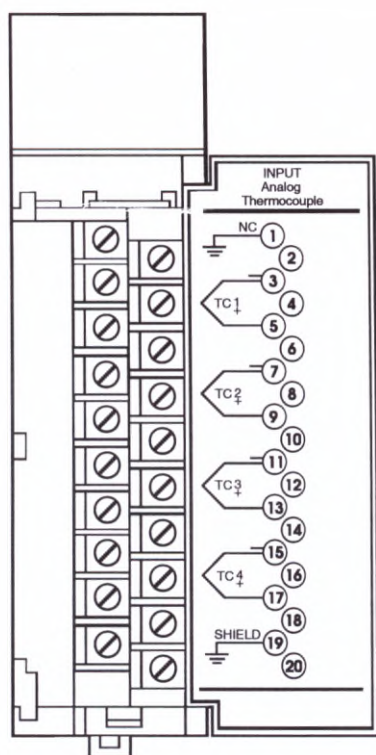
- 2-kanałowy, napięciowy moduł wyjść analogowych;
- 2-kanałowy, prądowy moduł wyjść analogowych;
- 8-kanałowy, prądowo-napięciowy moduł wyjść analogowych.

5. Moduły kombinowane:

- kombinowany moduł: 8 wejść dla napięcia 24 V prądu stałego oraz 8 wyjść przekaźnikowych zwiernych, o natężeniu prądu do 2 A;



Rys. 3.4. Moduł wejść analogowych



Rys. 3.5. Moduł do współpracy z termoparą

- kombinowany moduł: 8 wejść dla napięcia 120 V prądu przemiennego oraz 8 wyjść przekaźnikowych zwiernych, o natężeniu prądu do 2 A;
- prądowo-napięciowy moduł kombinowany wejść i wyjść analogowych: 4 kanały wejściowe, 2 kanały wyjściowe.

Jak widać z dokonanego powyżej przeglądu istnieje duża różnorodność modułów ze względu na liczbę obsługiwanych sygnałów oraz ich charakterystyki prądowe. Moduły analogowe posiadają rozdzielczość 12 bitową. Firma GE Fanuc zalicza specjalne moduły do bezpośredniej współpracy z termoparami (rys. 3.5.), termometrami oporowymi i mostkami tensometrycznymi do modułów wejść analogowych. Poniżej omówiono moduły dla termopar. Umożliwiają one bezpośrednie podłączenie termopary do sterownika, eliminując wszelkie urządzenia pośrednie (przetworniki, itp.) i zapewniając kompensację zimnych złączy. Dostępne są moduły współpracujące z termoparami typu J, K, T, N, E, R, S oraz B. Każdy kanał jest niezależnie konfigurowany dla odpowiedniego typu termopary. Sygnał analogowy z termopary (proporcjonalny do wartości mierzonej temperatury) jest przetwarzany przez moduł na zlinearyzowany sygnał cyfrowy o rozdzielczości

18 bitów. Tak więc moduł ten jest traktowany przez sterownik jako moduł wejść analogowych. Moduły są dostępne w wersji 4-kanałowej i 8-kanałowej. Pozwalają one na wykrycie warunków odpowiadających przerwie w obwodzie określonego kanału; wartość temperatury przyjmuje wtedy maksymalną wartość z dopuszczalnego zakresu. Rozdzielczość modułów wynosi 0.5 °C lub 0.1 °C w przypadku modułu o wysokiej rozdzielczości.

W celu dokładnego zapoznania się z parametrami modułów wejść/wyjść i schematami połączeń zaleca się Czytelnikowi odwiedzenie strony internetowej firmy Astor www.astor.com.pl, lub sięgnięcia do aktualnego „Katalogu sterowników GE Fanuc serii 90-30”.

3.3. Moduł szybkiego licznika HSC

Moduł licznika impulsów wysokiej częstotliwości HSC (High Speed Counter) – rys. 3.6. zapewnia bezpośrednie przetwarzanie sygnału w postaci impulsów o częstotliwości do 80 kHz. Oznacza to zliczanie sygnałów wejściowych, przetwarzanie informacji (liczby zebranych impulsów) i ustawienie stanu wyjść bez konieczności komunikowania się z jednostką centralną. Najczęściej generatorami impulsów zliczanych przez HSC są przetworniki impulsowo-obrotowe, tzw. enkodery. Typowe zastosowania przemysłowe licznika to:

- pomiary natężenia przepływu w turbinach;
- pomiary prędkości;
- pomiar położenia
- przenoszenie materiałów;
- sterowanie procesem produkcyjnym.

Moduł pracuje w jednym z trzech konfigurowalnych trybów (A, B, C). Przy konfiguracji A moduł pracuje jak cztery niezależne 16-bitowe liczniki jednokierunkowe, przy konfiguracji B jako dwa niezależne 32-bitowe liczniki dwukierunkowe a przy konfiguracji C jak jeden 32-bitowy licznik różnicowy (moduł oblicza różnicę dwóch szybko zmieniających się sygnałów). Dla trybów B i C sygnał może być przekazywany w jednym z trzech systemów:

1. Impuls-kierunek;
2. Impuls w górę/impuls w dół;
3. tzw. tryb A quad B, co oznacza dwa sygnały prostokątne, przesunięte w fazie o 90 stopni.

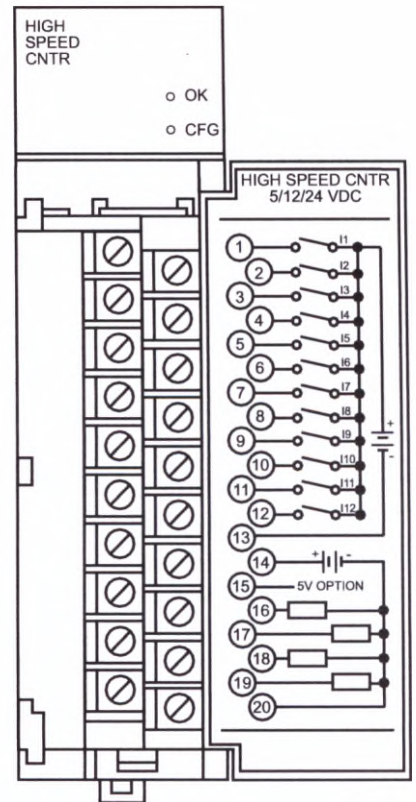
Moduł posiada cztery konfigurowalne wyjścia cyfrowe.

3.4. Moduły pozycjonujące dla 1, 2 lub 4 osi

Moduły te można wykorzystać jako kompletny układ sterowania napędem przy ruchu w jednej, dwóch lub czterech osiach. Mogą pracować w trybie standardowym (klasyczny układ pozycjonowania) lub w trybie nadażnym (elektroniczne sprzężenie ruchów). Tryb pracy jest wybierany z poziomu oprogramowania Logicmaster 90 lub VersaPro i CIMPLICITY Machine Edition dla Windows 95/98/NT/2000. Moduły mogą przechowywać do 10 programów ruchu, przy czym pierwszy program może być wpisany za pomocą standardowego oprogramowania Logicmaster 90, VersaPro i CIMPLICITY Machine Edition. Do wprowadzenia większej liczby programów należy zastosować oprogramowanie specjalne. Cechy modułu to między innymi:

- polecenia bezwzględne i przyrostowe;
- prostoliniowe i paraboliczne kształty przebiegów przyspieszeń;
- możliwość skoku warunkowego podczas ruchu;
- cykl powrotu do położenia bazowego;
- "program 0" wprowadzany wraz z konfiguracją sterownika;
- automatyczne przesyłanie danych pomiędzy tablicami danych sterownika a modułem pozycjonującym;
- trwałe przechowywanie 10 programów ruchu i 40 procedur;
- jednostki wybierane przez użytkownika;
- bezpośrednie sterowanie ruchem przez moduł, z pominięciem jednostki centralnej sterownika.

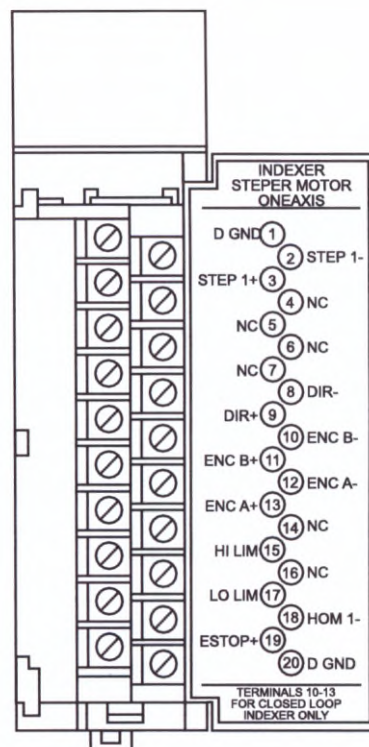
Moduły mogą prowadzić sterowanie napędem poprzez sygnał analogowy $\pm 10\text{VDC}$ lub mogą współpracować ze specjalizowanymi napędami GE Fanuc sterowanymi cyfrowo. (takie rozwiązanie zapewnia wysoką dokładność pozycjonowania).



Rys. 3.6. Moduł szybkiego licznika HSC

3.5. Moduł pozycjonujący dla silnika krokowego

Moduł ten (rys. 3.7.) funkcjonuje jako programowalny generator impulsów w układzie sterowania silnika krokowego. Dostępny jest w wersji jednoosiowej (ze sprzężeniem zwrotnym od enkodera lub bez sprzężenia zwrotnego) oraz trójosiowej (bez sprzężenia zwrotnego). Parametry ruchu dla silnika (położenie docelowe zadawane w wartościach bezwzględnych lub przyrostowo, prędkość, przyspieszenie, itp.) zadawane są z poziomu programu sterującego (przechowywane w rejestrach sterownika). Moduł jest wyposażony w diody LED wskazujące na bieżący tryb pracy silnika (zatrzymanie wirnika, ruch ze stałą prędkością obrotową, przyspieszanie, hamowanie). Moduł wyposażony jest w standardowe 20-wtykowe złącze służące do połączenia modułu z enkoderem, sterownikiem i innymi urządzeniami towarzyszącymi.



Rys. 3.7. Moduł dla silnika krokowego

3.6. Moduł programowalnego koprocesora PCM - Programmable Coprocessor Module

Moduł ten umożliwia realizowanie dodatkowych zadań niezależnie od jednostki centralnej sterownika. Może współpracować z jednostkami centralnymi od CPU 331 wzwyż. Zawiera procesor 80188, pamięć EPROM, podtrzymywaną bateryjnie pamięć RAM oraz zegar kontrolny. Może posiadać 192 kB lub 640 kB pamięci RAM. Ponadto wyposażony jest w trzy diody kontrolne wskazujące status modułu, przycisk zerujący oraz w złącze 25-wtykowe, obsługujące dwa szeregowe porty komunikacyjne: RS232 oraz RS232/485. Moduł PCM jest programowany w języku MegaBasic lub Microsoft C z komputera typu IBM PC.

Może być wykorzystany do:

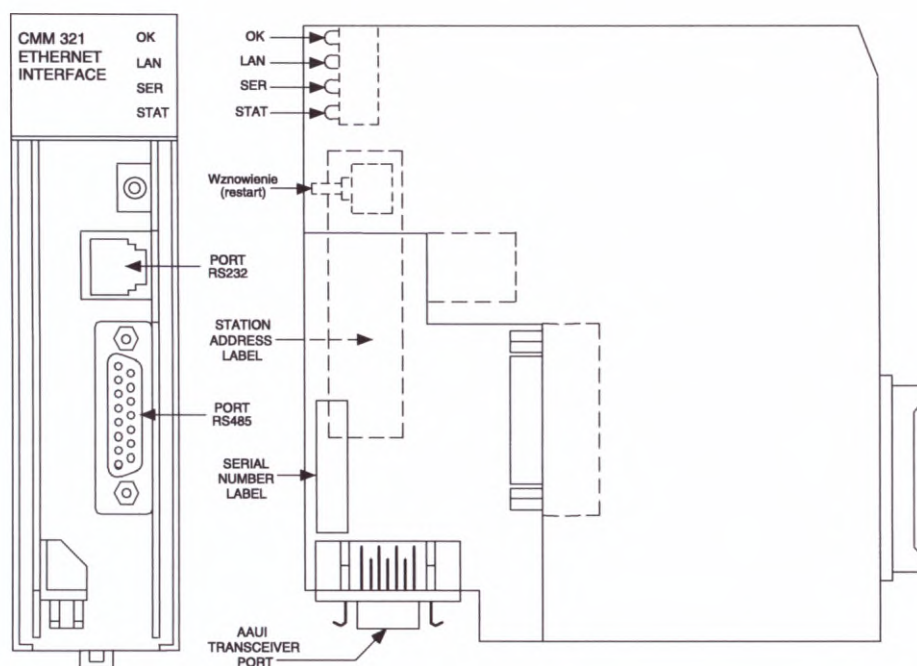
- gromadzenia dużej ilości danych;
- komunikacji poprzez port szeregowy z inteligentnymi urządzeniami – wagi, interfejsy kodu paskowego, modemy, urządzenia pomiarowe – istnieje możliwość pisania programów do obsługi niestandardowych protokołów;
- drukowania raportów;
- realizacji skomplikowanych obliczeń niekrytycznych czasowo.

Moduł PCM może być skonfigurowany do działania jako:

- jeden lub dwa niezależne porty CCM;
- jeden port CCM i aplikacja MegaBasic korzystająca także z jednego portu;
- jedna aplikacja MegaBasic korzystająca z jednego lub obu portów.

Protokół CCM (Communications Control Module) może pracować w topologiach: master, slave, peer to peer.

3.7. Moduł komunikacyjny TCP/IP Ethernet



Rys. 3.8. Moduł komunikacyjny TCP/IP Ethernet

Moduł ten (rys. 3.8.) stanowi interfejs pomiędzy sterownikiem serii 90-30 a siecią Ethernet TCP/IP LAN i umożliwia komunikację z komputerem oraz innymi urządzeniami w sieci. Może pracować jako klient (sam inicjuje komunikację z innymi sterownikami wyposażonymi w moduły Ethernet) lub serwer (jego rola ogranicza się do odpowiedzi na żądania przesłania danych z innych urządzeń, np. komputera nadrzędnego lub innego sterownika skonfigurowanego jako klient). Moduł konfigurowany jest za pomocą oprogramowania Logicmaster 90, VersaPro lub CIMPLICITY Machine Edition dla Windows 95/98/NT/2000. Pracuje w protokole SRTP (Service Request Transfer Protocol), który jest wersją protokołu SNP dla Ethernet TCP/IP (Transfer Control Protocol / Internet Protocol). Moduł posiada jeden port ze złączem AAUI i aby połączyć go do sieci należy zastosować transceiver. Szybkość transmisji danych w sieci wynosi do 10 MB/s. Moduł ten można wykorzystywać do:

- programowania ON-Line sterownika w sieci;
- wymiany danych przez programy aplikacyjne.

Literatura

- [1] Katalog sterowników GE Fanuc serii 90-30 (LI-ASK-PLC-GE3), Astor, Kraków
- [2] Legierski T., Kasprzyk J., Wyrwał J., Hajda J., Programowanie sterowników PLC, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 1998
- [3] Programmable Coprocessor Module and Support Software. User's Manual, GFK-0255J, GE Fanuc Automation, August 1996
- [4] Stec Ł., Moduł wagowy w sterowniku GE Fanuc, Biuletyn Automatyki ASTOR, 4/99 (22), str. 6
- [5] TCP/IP Ethernet Communications for the Series 90-30 PLC. User's Manual, GFK-1084B, GE Fanuc Automation, August, 1997

Część danych i rysunków prezentowanych w niniejszym rozdziale zaczerpnięto z dysków informacyjno-reklamowych CD ASTOR 97 i ASTOR CD 2001.

4. Podstawy programowania

W podrozdziale 4.1. została przedstawiona krótko norma IEC 61131 (dawniej IEC 1131) w części dotyczącej programowania (część trzecia normy). Następnie omówiono typy danych i zmiennych oraz podstawy języka LD dla sterowników GE Fanuc, w szczególności serii 90-30 i oprogramowania narzędziowego Logicmaster 90.

4.1. Języki programowania sterowników wg IEC 61131

W początkowej fazie rozwoju sterowników producenci sprzętu i oprogramowania narzędziowego proponowali znacznie różniące się między sobą sposoby ich programowania. Stanowiło to znaczne utrudnienie dla projektantów i użytkowników, zmuszając ich do ciągłego dostosowywania się do nowych wymagań. Wraz ze wzrostem popularności sterowników PLC, coraz ważniejszym zagadnieniem stawała się ich standaryzacja, szczególnie w zakresie programowania. W końcu w 1993 roku Międzynarodowa Komisja Elektroniki wydała normę IEC 1131 „Programmable Controllers” (obecnie IEC 61131). Składa się ona z pięciu części (kolejne trzy są w opracowaniu):

1. Informacje ogólne (General Information);
2. Sprzęt i wymagania testowe (Equipment and Test Requirements);
3. Języki programowania (Programming Languages);
4. Wytyczne użytkownika (User Guidelines);
5. Wymiana informacji (Messaging Service).

Dwie pierwsze części zostały przyjęte w Polsce bez zmian jako PN-IEC 1131-1:1996 i PN-IEC 1131-2:1996, natomiast trzecia jako PN-EN 61131-3:1998.

Z punktu widzenia niniejszego rozdziału najistotniejsza jest trzecia część normy IEC 61131-3, dotycząca języków programowania. Dzięki wprowadzeniu przez nią ogólnych zasad, tj. ujednocnieniu koncepcji programowania sterowników, programowanie różnych systemów PLC stało się znacznie łatwiejsze. Użytkownik, umiejąc programować jeden typ sterownika, nie powinien mieć większych trudności z nauką programowania innego systemu. Norma wyróżnia cztery języki:

dwa graficzne:

- język LD (Ladder Diagram) – schemat drabinkowy;
- język FBD (Function Block Diagram) – język schematów bloków funkcji;

i dwa tekstowe:

- język listy rozkazów IL (Instruction List) – lista instrukcji;
- język tekstu strukturalnego ST (Structured Text).

Poza nimi dopuszcza także formalizm SFC (Sequential Function Chart) – sekwencyjną tablicę funkcji (graf sekwencji). Pozwala on na opisywanie zadań sterowania sekwencyjnego za pomocą grafów zawierających kroki i tranzycje między nimi. Do opisu etapów i tranzycji używa się jednego ze zdefiniowanych w normie języków. Formalizm SFC oparty o język LD dla sterowników GE Fanuc został przedstawiony w rozdziale 5.

Język LD – schemat drabinkowy

Często określany językiem drabinkowym. Jest on podobny do schematu stykowych obwodów przekaźnikowych. Dopuszcza użycie szeregu funkcji: arytmetycznych, logicznych, relacji oraz bloków funkcyjnych: liczników, czasomierzy, regulatora PID. Ze względu na swoje podobieństwo do obwodów przekaźnikowych, a przez to łatwość przyswojenia dla użytkowników znających metody budowy układów sterujących w „logice zaszytej”, jest on najbardziej rozpowszechnionym i najchętniej używanym językiem programowania sterowników PLC.

Język FBD – język schematów bloków funkcji

Jest on odpowiednikiem schematów przepływu sygnału dla obwodów logicznych przedstawionych w formie połączonych bramek logicznych oraz bloków funkcyjnych.

Język listy rozkazów IL

Nazywany czasami listą instrukcji. Jest odpowiednikiem języka typu assembler, którego zbiór instrukcji obejmuje operacje logiczne, arytmetyczne, relacje oraz funkcje przerzutników, czasomierzy, liczników itp.

Język tekstu strukturalnego ST

Jest on nazywany także tekstem strukturalnym. Jest odpowiednikiem języka algorytmicznego wysokiego poziomu, zawierającego struktury programowe np.:

If...then...else...end_if,

For...to...do...end_for,

While...do...end_while,

Case...of...end_case,

Repeat...end_repeat.

4.2. Typy danych i zmiennych

Norma IEC 61131-3 definiuje także elementarne typy danych jakie mogą być używane w systemach sterownikowych. **Typ** określa zbiór wartości jakie mogą przyjmować dane, a zarazem zbiór operacji, jakie mogą być na nich wykonywane. W tab. 4.1. zestawiono wszystkie elementarne typy danych dopuszczone w normie. W tab. 4.2. opisano typy danych stosowane w sterownikach GE Fanuc serii 90-30. Nazwy typu używane przez producentów nie zawsze pokrywają się z nazwami podanymi w normie np. BOOL i BIT.

Tab. 4.1. Elementarne typy danych wg IEC 61131-3 (PN-EN 61131-3:1998)

L.p.	Słowo kluczowe	Typ danej	Liczba używanych bitów
1.	BOOL	logiczna (Boolean)	1
2.	SINT	całkowita 8-bitowa (Short integer)	8
3.	INT	całkowita 16-bitowa (Integer)	16
4.	DINT	całkowita 32-bitowa (Double integer)	32
5.	LINT	całkowita 64-bitowa (Long integer)	64
6.	USINT	całkowita 8-bitowa bez znaku (Unsigned short integer)	8
7.	UINT	całkowita 16-bitowa bez znaku (Unsigned integer)	16
8.	UDINT	całkowita 32-bitowa bez znaku (Unsigned double)	32

		integer)	
9.	ULINT	całkowita 64-bitowa bez znaku (Unsigned long integer)	64
10.	REAL	liczba rzeczywista (Real number)	32
11.	LREAL	liczba rzeczywista 64-bitowa (Long real)	64
12.	TIME	czas trwania (Duration)	*
13.	DATE	data (Date)	*
14.	TIME_OF_DAY lub TOT	godzina dnia (Time of date)	*
15.	DATE_AND_DAY lub DT	data i czas (Date and time of day)	*
16.	STRING	ciąg znaków o zmiennej długości (Variable-length character string)	*
17.	BYTE	bajt	8
18.	WORD	słowo	16
19.	DWORD	ciąg 32 bitów (słowo podwójne)	32
20.	LWORD	ciąg 64 bitów (słowo podwójne)	64

* Długość tego typu elementów danych jest zależna od implementacji.

Tab. 4.2. Podstawowe typy danych dla sterowników GE Fanuc serii 90-30

L.p.	Słowo kluczowe	Nazwa - typ danej	Opis
1.	BIT	Bit	Dana zajmująca najmniejszą komórkę pamięci, mogąca przyjmować wartość 0 lub 1.
2.	BYTE	Byte – bajt	Bajt to 8 bitów.
3.	WORD	Word – słowo	Wykorzystuje 16 kolejnych bitów pamięci sterownika lecz, w przeciwieństwie do ciągu bitów reprezentującego liczbę, bity mogą być niezależne od siebie. Każdy bit posiada swój własny stan logiczny 0 lub 1. Zakres: od 0 do FFFF (w układzie heksadecymalnym).
4.	DWORD	Double word – słowo podwójne	Słowo podwójne ma takie same cechy jak słowo, z wyjątkiem tego, iż używa 32 kolejnych bitów zamiast 16
5.	BCD-4	Four-Digit Binary Coded Decimal – czterocyfrowa liczba dziesiętna zakodowana w formacie BCD	Każda z czterech cyfr tej liczby jest zakodowana w czterech bitach w systemie dwójkowym. Zakres: od 0 do 9.999
6.	REAL*	Floating point – liczba rzeczywista	Liczby rzeczywiste zajmują 32 bity pamięci. Zakres: od $\pm 1,401298E-45$ do $\pm 3,402823E+38$
7.	INT	Signed integer – liczba całkowita ze znakiem	Liczby całkowite ze znakiem zajmują 16 bitów pamięci (16 bit jest bitem znaku). Zakres: od -32768 do 32767
8.	DINT	Double precision signed integer – liczba całkowita podwójnej precyzji ze znakiem	Liczby całkowite podwójnej precyzji ze znakiem zajmują 32 bitów pamięci (32 bit jest bitem znaku). Zakres: od -2.147.483.648 do +2.147.483.647

* typ REAL dostępny jest dla jednostek od CPU 350 wzwyż.

Sterowniki GE Fanuc z jednostkami CPU 331 i wyższymi posiadają również zegar czasu rzeczywistego. Posiada on dwie funkcje:

- Rok (dwie cyfry) – Miesiąc – Dzień miesiąca;
- Godzina – Minuta – Sekunda – Dzień tygodnia.

Podczas pisania programu sterującego elementom logicznym przypisuje się **zmienne**. W sterownikach PLC występują dwa główne typy zmiennych:

- **dyskretne** (jednobitowe – przyjmujące wartości 0 lub 1);
- **rejestrów** (służące do przechowywania danych innych niż pojedyncze bity).

Wśród zmiennych dyskretnych można wyodrębnić zmienne użytkownika (ich wartości zmieniają się wskutek wykonywania części logicznej programu sterującego w sposób zaprogramowany przez użytkownika lub zmian zaistniałych na sterowanym obiekcie) oraz zmienne systemowe, których wartości zależą od chwilowego stanu sterownika i są ustawiane przez system. W tab. 4.3. przedstawiono typy zmiennych rejestrów, a w tab. 4.6. dyskretnych, wraz symbolami, występujące w sterownikach GE Fanuc serii 90-30.

Tab. 4.3. Zmienne rejestrowe w sterownikach 90-30

Typ	Opis
%AI	Zmienna reprezentująca rejestr wejścia analogowego. Symbolowi powinien towarzyszyć adres rejestru (np. %AI0004).
%AQ	Zmienna reprezentująca rejestr wyjścia analogowego. Symbolowi powinien towarzyszyć adres rejestru (np. %AQ0003).
%R	Zmienna oznaczająca rejestr, w którym można przechowywać dane programu (np. wyniki obliczeń). Symbolowi powinien towarzyszyć adres rejestru (np. %R00014).

Tab. 4.4. Zmienne dyskretne w sterownikach 90-30

Typ	Opis
%I	Zmienna reprezentująca fizyczne wejście dyskretne. Po symbolu następuje adres zmiennej w tablicy stanu wejść (np. %I00231). Wartości zmiennych typu %I są umieszczane w tablicy stanu wejść. Przechowywany jest w niej aktualny stan wszystkich wejść sterownika uzyskany z modułów wejściowych podczas ostatniego odczytu. Adres zmiennej jest przyporządkowany modułom wejść dyskretnych podczas konfiguracji sterownika. Do momentu przyporządkowania adresu zmiennej moduł wejść nie przesyła żadnych danych do sterownika.
%Q	Zmienna reprezentująca fizyczne wyjście dyskretne. Po symbolu następuje adres zmiennej w tablicy stanu wyjść (np. %Q00056). Wartości zmiennych typu %Q są umieszczane w tablicy stanu wyjść. Przechowywany jest w niej aktualny stan wszystkich wyjść sterownika ustalonych przez program sterujący po wykonaniu poprzedniego cyklu. Wartości te są również przesyłane do modułów wyjściowych. Zmienna typu %Q może posiadać pamięć stanu (tzn. zachowywać poprzednią wartość po wyłączeniu zasilania sterownika i ponownym jego włączeniu) jeśli przypisana jest przekaźnikowi z pamięcią lub stanowi parametr wyjściowy bloku funkcyjnego. W przypadku przypisania przekaźnikowi bez pamięci jej wartość po wyłączeniu sterownika zostaje wyzerowana.
%M	Symbol reprezentujący wewnętrzną zmienną dyskretną programu sterującego. Zmienna typu %M może posiadać pamięć stanu jeśli jest przypisana przekaźnikowi z pamięcią. W przypadku przypisania przekaźnikowi bez pamięci, zmienna nie posiada pamięci stanu.

%T	Symbol reprezentujący chwilową zmienną dyskretną, bez możliwości wykorzystania pamięci stanu. Po wyłączeniu sterownika i jego ponownym włączeniu wartość tej zmiennej jest zawsze ustawiona na 0.
%S	Symbol %S oznacza zmienne systemowe. Umożliwiają one dostęp do danych systemowych, takich jak informacje o błędach działania sterownika i pracy modułów wejść/wyjść. Wśród zmiennych systemowych można wyodrębnić cztery grupy oznaczone odpowiednio %S, %SA, %SB i %SC. W tab.4.5 przedstawiono część zmiennych %S, które są szczególnie przydatne podczas programowania. W celu zapoznania się z innymi zmiennymi systemowymi odsyła się Czytelnika do literatury [9], [10].
%G	Symbol ten reprezentuje zmienne globalne umożliwiające dostęp do danych wspólnych dla kilku sterowników. Zmienne typu %G posiadają zawsze pamięć stanu i nie mogą być przypisywane przekaźnikom bez pamięci.

Tab. 4.5. Wybrane zmienne systemowe %S sterowników 90-30

Zmienna	Nazwa pomocnicza	Opis
%S0001	FST_SCN	Bieżący cykl jest pierwszym cyklem pracy sterownika – w pierwszym cyklu przyjmuje wartość 1, w następnych 0.
%S0002	LST_SCN	Bieżący cykl nie jest ostatnim cyklem pracy sterownika – zmienna przełączana jest z 1 na 0 kiedy aktualny cykl jest cyklem ostatnim.
%S0003	T_10MS	Podstawa czasu generatora sygnału prostokątnego: 0,01 s.
%S0004	T_100MS	Podstawa czasu generatora sygnału prostokątnego: 0,1 s.
%S0005	T_SEC	Podstawa czasu generatora sygnału prostokątnego: 1 s.
%S0006	T_MIN	Podstawa czasu generatora sygnału prostokątnego: 1 min.
%S0007	ALW_ON	Styk zwarty na stałe – stała 1
%S0008	ALW_OFF	Styk otwarty na stałe – stała 0

Uwaga: Wartości zmiennych systemowych %S można jedynie odczytywać, nie można ich zmieniać.

4.3. Elementy logiczne programu sterującego w sterownikach GE Fanuc serii 90-30

W rozdziale opisano krótko wszystkie instrukcje logiczne dostępne podczas budowy programu sterującego dla sterowników GE Fanuc serii 90-30. Instrukcje te są zestawione w oprogramowaniu narzędziowym Logicmaster 90 w 9 grup:

1. przekaźniki i styki;
2. przekaźniki czasowe i liczniki;
3. funkcje matematyczne;
4. relacje matematyczne;
5. operacje bitowe;
6. operacje na danych;
7. operacje tablicowe;
8. funkcje konwersji;
9. funkcje sterujące.

W tab. 4.6. – 4.14. przedstawiono instrukcje należące do kolejnych grup wraz z ich krótką charakterystyką. Bardziej wyczerpujące informacje można znaleźć w publikacji [10].

Ad. 1. Przekazniki i styki

Tab. 4.6. Przekazniki i styki w sterownikach 90-30

Symbol	Nazwa	Opis
— —	Styk otwarty	Przewodzi sygnał gdy wartość logiczna przypisanej zmiennej jest równa 1.
— /—	Styk zamknięty	Przewodzi sygnał gdy wartość logiczna przypisanej zmiennej jest równa 0.
—(—)	Przełącznik o stykach otwartych	Przełącznik o stykach otwartych, zwieranych w momencie dotarcia sygnału do przełącznika. Wartość logiczna przypisanej mu zmiennej jest wtedy ustawiana na 1.
—(SM)—	Przełącznik ustawienia z pamięcią SET	Przełącznik o stykach zwieranych w momencie dotarcia sygnału (wartość logiczna przypisanej mu zmiennej ustawiana jest wtedy na 1). Styki pozostają zwarte, a wartość zmiennej pozostaje równa 1 do czasu zadziałania sprzężonego przełącznika —(RM)—. Stan przełącznika pozostaje niezmienny nawet w przypadku odcięcia zasilania sterownika i ponownego jego włączenia.
—(RM)—	Przełącznik ustawienia z pamięcią RESET	Przełącznik o stykach rozwieranych w momencie dotarcia sygnału (wartość logiczna przypisanej mu zmiennej ustawiana jest wtedy na 0). Styki pozostają rozwarte, a wartość zmiennej pozostaje równa 0 do czasu zadziałania sprzężonego przełącznika —(SM)—. Stan przełącznika pozostaje niezmienny nawet w przypadku odcięcia zasilania sterownika i ponownego jego włączenia.
	Połączenie pionowe	
—	Połączenie poziome	
—<+>	Przełącznik kontynuacji	Przełącznik kontynuacji – jeśli dopływa do niego sygnał zwierający następujący po nim styk kontynuacji. Jeśli sygnał nie dopływa styk kontynuacji również pozostaje otwarty.
<+>—	Styk kontynuacji	Styk kontynuacji zwierany i rozwierany przez przełącznik kontynuacji.
—(/M)—	Przełącznik o stykach zamkniętych z pamięcią	Przełącznik o stykach zamkniętych rozwieranych w momencie dotarcia sygnału do przełącznika. Wartość logiczna przypisanej mu zmiennej ustawiana jest wtedy na 0. Stan przełącznika pozostaje niezmienny nawet w przypadku odcięcia zasilania sterownika i ponownego jego włączenia.
—(/)—	Przełącznik o stykach zamkniętych	Przełącznik o stykach zamkniętych rozwieranych w momencie dotarcia sygnału do przełącznika. Wartość logiczna przypisanej mu zmiennej ustawiana jest wtedy na 0.
—(M)—	Przełącznik o stykach otwartych z pamięcią	Przełącznik o stykach otwartych zwieranych w momencie dotarcia sygnału do przełącznika. Wartość logiczna przypisanej mu zmiennej ustawiana jest wtedy na 1. Stan przełącznika pozostaje niezmienny nawet w przypadku odcięcia zasilania sterownika i ponownego jego włączenia.
—(S)—	Przełącznik ustawienia SET	Przełącznik o stykach zwieranych w momencie dotarcia sygnału (wartość logiczna przypisanej mu zmiennej ustawiana jest wtedy na 1). Styki pozostają zwarte, a wartość zmiennej pozostaje równa 1 do czasu zadziałania sprzężonego przełącznika —(R)—.
—(R)—	Przełącznik ustawienia RESET	Przełącznik o stykach rozwieranych w momencie dotarcia sygnału (wartość logiczna przypisanej mu zmiennej ustawiana jest wtedy na 0). Styki pozostają rozwarte, a wartość zmiennej pozostaje równa 0 do czasu zadziałania sprzężonego przełącznika —(S)—.

—(↑)—	Przełącznik uaktywniany zboczem narastającym	Przełącznik uaktywniany zboczem sygnału (typ pozytywny). Jeśli do przełącznika nie docierał sygnał, to w momencie dotarcia sygnału wartość logiczna przypisanej zmiennej dyskretnej jest ustawiana na 1 do czasu wykonania tej instrukcji w następnym cyklu. Jeśli w kilku następnych cyklach pracy sterownika szczebel zawierający omawiany przełącznik jest pomijany to wartość przypisanej mu zmiennej pozostanie równa 1 przez czas trwania tych cykli.
—(↓)—	Przełącznik uaktywniany zboczem opadającym	Przełącznik uaktywniany zboczem sygnału (typ negatywny). Jeśli do przełącznika docierał sygnał, to w momencie wystąpienia przerwy w dopływie sygnału wartość logiczna przypisanej zmiennej dyskretnej jest ustawiana na 1 do czasu wykonania tej instrukcji w następnym cyklu. Jeśli w kilku następnych cyklach pracy sterownika szczebel zawierający omawiany przełącznik jest pomijany to wartość przypisanej mu zmiennej pozostanie równa 1 przez czas trwania tych cykli.

Ad. 2. Przełączniki czasowe i liczniki

Tab. 4.7. Przełączniki czasowe i liczniki sterowników 90-30

Symbol	Nazwa	Opis
ONDTR	Przełącznik czasowy z pamięcią	Przełącznik czasowy z pamięcią zlicza czas gdy dopływa do niego sygnał i zatrzymuje naliczoną wartość, gdy sygnał przestaje dopływać. Gdy sygnał zacznie dopływać ponownie zliczanie czasu jest kontynuowane. Przesłanie sygnału wyjściowego następuje gdy zliczona wartość osiągnie wartość zadaną. Czas może być zliczany w dziesiątych, setnych lub tysięcznych częściach sekundy. Zakres mierzonej wartości wynosi 0 - +32.767 jednostek czasu. Bieżąca wartość pamięci przełącznika może być wyzerowana poprzez przesłanie sygnału na wejście zerujące (R). Blok funkcyjny ONDTR wykorzystuje 3 kolejne słowa pamięci sterownika.
OFTD	Przełącznik czasowy wyłączający	Przełącznik zlicza czas, w którym nie dopływa do niego sygnał. Gdy naliczona wartość osiągnie wartość zadaną przełącznik wyłącza wyjście (ustawia go w stan OFF). Gdy na wejściu przełącznika pojawi się sygnał zostaje wyzerowany licznik czasu, a wyjście ustawione na ON. Czas może być zliczany w dziesiątych, setnych lub tysięcznych częściach sekundy. Zakres mierzonej wartości wynosi 0 - +32.767 jednostek czasu. Blok funkcyjny OFTD wykorzystuje 3 kolejne słowa pamięci sterownika.
TMR	Przełącznik czasowy bez pamięci	Przełącznik czasowy bez pamięci zlicza czas gdy dopływa do niego sygnał i zostaje wyzerowany, gdy sygnał przestaje dopływać. Przesłanie sygnału wyjściowego następuje gdy zliczona wartość osiągnie wartość zadaną. Czas może być zliczany w dziesiątych, setnych lub tysięcznych częściach sekundy. Zakres mierzonej wartości wynosi 0 - +32.767 jednostek czasu. Blok funkcyjny TMR wykorzystuje 3 kolejne słowa pamięci sterownika.
UPCTR	Licznik zliczający w górę	Licznik zliczający w górę – jego wartość jest zwiększana o 1 za każdym razem gdy wartość dopływającego sygnału zmienia się z 0 na 1. Licznik przesyła sygnał wyjściowy gdy zliczona wartość osiągnie wartość zadaną. Zakres licznika wynosi 0 - +32.767 impulsów. Bieżąca wartość pamięci licznika może być wyzerowana poprzez przesłanie sygnału na wejście zerujące (R). Blok funkcyjny UPCTR wykorzystuje 3 kolejne słowa pamięci sterownika.

DNCTR	Licznik zliczający w dół	Licznik zliczający w dół służy do odliczania impulsów sygnału od zadanej wartości do 0. Maksymalna wartość zadana wynosi +32.767 impulsów. Minimalna wartość bieżąca wynosi -32.768 impulsów. Bieżąca wartość licznika zostaje zmniejszona o 1 za każdym razem gdy wartość dopływającego sygnału zmienia się z 0 na 1. Licznik przesyła sygnał wyjściowy gdy zliczona wartość osiągnie zero. Bieżąca wartość pamięci licznika może być wyzerowana (tj. ustawiona na wartość zadana) poprzez przesłanie sygnału na wejście zerujące (R). Blok funkcyjny DNCTR wykorzystuje 3 kolejne słowa pamięci sterownika.
-------	--------------------------	---

Uwaga: Podstawa czasu 0,001 s jest dostępna dla wersji 4.5 i wyższych oprogramowania sprzętowego jednostki CPU i programu Logicmaster 90-30.

Ad. 3. Funkcje matematyczne

Tab. 4.8. Funkcje matematyczne w sterownikach 90-30

Symbol	Nazwa	Opis
ADD	Dodawanie	Dodawanie dwóch liczb. Blok funkcyjny ADD przesyła sygnał wyjściowy gdy wynik działania nie przekracza dopuszczalnego zakresu wartości.
SUB	Odejmowanie	Odejmowanie dwóch liczb. Blok funkcyjny SUB przesyła sygnał wyjściowy gdy wynik działania nie przekracza dopuszczalnego zakresu wartości.
MUL	Mnożenie	Mnożenie dwóch liczb. Blok funkcyjny MUL przesyła sygnał wyjściowy gdy wynik działania nie przekracza dopuszczalnego zakresu wartości.
DIV	Dzielenie	Dzielenie dwóch liczb. Wynikiem jest część całkowita ilorazu. Blok funkcyjny DIV przesyła sygnał wyjściowy gdy wynik działania nie przekracza dopuszczalnego zakresu wartości i nie ma próby dzielenia przez zero.
MOD	Dzielenie z resztą	Dzielenie dwóch liczb z resztą. Wynikiem jest reszta z dzielenia. Znak wyniku jest zawsze taki sam jak znak pierwszego parametru wejściowego (I1). Blok funkcyjny MOD przesyła sygnał wyjściowy gdy nie ma próby dzielenia przez zero.
SQRT	Pierwiastek kwadratowy	Blok funkcyjny SQRT wylicza pierwiastek kwadratowy z liczby, która jest jego parametrem wejściowym. Gdy operacja jest wykonywana na danych typu INT to wynikiem jest część całkowita pierwiastka. Blok funkcyjny SQRT przesyła sygnał wyjściowy gdy wynik działania nie przekracza dopuszczalnego zakresu wartości. Jeśli następuje próba wykonania niewykonalnej operacji (np. pierwiastek z liczby ujemnej) sygnał wyjściowy nie jest przesyłany.
SIN*	Sinus	
COS*	Cosinus	
TAN*	Tangens	
ASIN*	Arcus sinus	
ACOS*	Arcus cosinus	
ATAN*	Arcus tangens	
DEG*	Konwersja na stopnie	Kiedy do bloku DEG dochodzi sygnał, zmienia on wartość wejściową podaną w radianach na wyjściową wyrażoną w stopniach.
RAD*	Konwersja na radiany	Kiedy do bloku RAD dochodzi sygnał, zmienia on wartość wejściową podaną w stopniach na wyjściową wyrażoną w radianach.
LOG10*	Logarytm dziesiętny	

LN*	Logarytm naturalny	
EXP*	Potęga e	
EXPT*	Potęga x	

* Funkcje te wykonywane są tylko na danych typu REAL i dlatego wymagają co najmniej CPU 350

Ad. 4. Relacje matematyczne

Tab. 4.9. Relacje matematyczne w sterownikach 90-30

Symbol	Nazwa	Opis
EQ	Równy	Blok funkcyjny EQ przesyła sygnał wyjściowy gdy porównywane wielkości są równe.
NE	Nierówny	Blok funkcyjny NE przesyła sygnał wyjściowy gdy porównywane wielkości nie są równe.
GT	Większy	Blok funkcyjny GT przesyła sygnał wyjściowy gdy pierwszy parametr bloku jest liczbą większą od drugiego parametru.
GE	Większy-równy	Blok funkcyjny GE przesyła sygnał wyjściowy gdy pierwszy parametr bloku jest liczbą większą lub równą drugiemu parametrowi.
LT	Mniejszy	Blok funkcyjny LT przesyła sygnał wyjściowy gdy pierwszy parametr bloku jest liczbą mniejszą od drugiego parametru.
LE	Mniejszy-równy	Blok funkcyjny LE przesyła sygnał wyjściowy gdy pierwszy parametr bloku jest liczbą mniejszą lub równą drugiemu parametrowi.
RANGE*	Zakres	Blok funkcyjny RANGE przesyła sygnał wyjściowy gdy parametr bloku mieści się w podanym zakresie.

* Funkcja jest dostępna dla wersji 4.5 i wyższych oprogramowania sprzętowego jednostki CPU i programu Logicmaster 90-30.

Ad. 5. Operacje bitowe

Bloki funkcyjne tej grupy wykonują operacje logiczne na ciągach bitów. Maksymalna długość ciągu wynosi 256 słów. Wymagany typ danych to WORD. Dane są pogrupowane w 16-bitowe słowa, ale traktowane jak nieprzerwany ciąg bitów z pierwszym bitem pierwszego słowa stanowiącym bit najbardziej znaczący (MSB) i ostatnim bitem ostatniego słowa stanowiącym bit najmniej znaczący (LSB).

Tab. 4.10. Operacje bitowe w sterownikach 90-30

Symbol	Nazwa	Opis
AND	Koniunkcja	Logiczne „i” dwóch słów bitowych (dwóch ciągów zerjedynkowych).
OR	Alternatywa	Logiczne „lub” dwóch słów bitowych (dwóch ciągów zerjedynkowych).
XOR	Alternatywa wyłączająca	Logiczne „albo” dwóch słów bitowych (dwóch ciągów zerjedynkowych).
NOT	Negacja	Logiczna negacja słowa bitowego.
SHL	Przesunięcie w lewo	Przesunięcie słowa bitowego o jedno miejsce w lewo (na zwolnioną pozycję wpisywane jest 0).
SHR	Przesunięcie w prawo	Przesunięcie słowa bitowego o jedno miejsce w prawo (na zwolnioną pozycję wpisywane jest 0).
ROL	Rotacja w lewo	Przesunięcie słowa bitowego o jedno miejsce w lewo przy czym bit, który przed przesunięciem był bitem pierwszym po przesunięciu staje się ostatnim bitem nowego słowa.

ROR	Rotacja w prawo	Przesunięcie słowa bitowego o jedno miejsce w prawo przy czym bit, który przed przesunięciem był bitem ostatnim po przesunięciu staje się pierwszym bitem nowego słowa.
BITTST	Test bitu	Sprawdzenie wartości logicznej wyszczególnionego bitu w ciągu słów bitowych.
BITSET	Ustawienie bitu	Przypisanie wartości logicznej równej 1 wyszczególnionemu bitowi w ciągu słów bitowych.
BITCLR	Zerowanie bitu	Przypisanie wartości logicznej równej 0 wyszczególnionemu bitowi w ciągu słów bitowych.
BITPOS	Lokalizowanie	Zlokalizowanie pierwszego bitu o wartości logicznej równej 1 w danym słowie bitowym (lub w ciągu słów). Pozycja zlokalizowanego bitu jest udostępniana przez parametr POS.
MSKCMP*	Porównanie i maskowanie	Wykonuje porównanie zawartości dwóch różnych ciągów bitów z możliwością maskowania wybranych bitów.

* Funkcja jest dostępna dla wersji 4.5 i wyższych oprogramowania sprzętowego jednostki CPU i programu Logicmaster 90-30.

Ad. 6. Operacje na danych

Tab. 4.11. Operacje na danych w sterownikach 90-30

Symbol	Nazwa	Opis
MOVE	Przemieszczenie	Przesłanie jednego bitu, większej liczby bitów lub słów bitowych w obrębie pamięci sterownika.
BLKMOV	Przemieszczenie grupy wartości	Przemieszczenie grupy siedmiu stałych do pamięci sterownika.
BLKCLR	Zerowanie	Przypisanie wartości logicznej 0 jednemu lub większej liczbie słów w pamięci sterownika.
SHFREG	Rejestr przemieszczający	Przemieszczenie jednego lub kilku: bitów albo słów, jako bloku w pamięci sterownika.
BITSEQ	Wędrująca jedyńka	Cykliczne przemieszczanie w górę lub dół bitu o wartości 1 w pewnym obszarze pamięci, w którym pozostałe bity mają wartość 0.
COMREQ	Żądanie komunikacji	Zainicjowanie komunikacji z jednym z modułów inteligentnych sterownika.

Ad. 7. Operacje tablicowe

Tab. 4.12. Operacje tablicowe w sterownikach 90-30

Symbol	Opis
SRH_EQ	Przeszukiwanie tablicy danych w celu znalezienia wartości równych wartości zadanej.
SHR_NE	Przeszukiwanie tablicy danych w celu znalezienia wartości różnych od wartości zadanej.
SHR_GT	Przeszukiwanie tablicy danych w celu znalezienia wartości większych od wartości zadanej.
SHR_GE	Przeszukiwanie tablicy danych w celu znalezienia wartości większych lub równych wartości zadanej.
SHR_LT	Przeszukiwanie tablicy danych w celu znalezienia wartości mniejszych od wartości zadanej.
SHR_LE	Przeszukiwanie tablicy danych w celu znalezienia wartości mniejszych lub równych wartości zadanej.
ARRMOV	Skopiowanie określonej liczby danych z tablicy źródłowej do tablicy docelowej.

Ad. 8. Funkcje konwersji

Tab. 4.13. Funkcje konwersji w sterownikach 90-30

Symbol	Opis
→BCD-4	Konwersja wartości na 4-cyfrowy kod zapisu BCD. Jeśli liczba, której konwersja ma zostać dokonana wykracza poza dopuszczalny zakres (jest większa od 9.999) konwersja nie następuje i sygnał wyjściowy nie zostaje przesłany.
→WORD*	Konwersja liczby REAL na dane typu WORD. Blok funkcyjny →WORD nie przesyła sygnału wyjściowego jedynie wtedy gdy liczba, której konwersja ma zostać dokonana, wykracza poza dopuszczalny zakres (od 0 do +65.535). Konwersja zostaje wtedy dokonana na wartość minimalną lub maksymalną.
→INT	Konwersja wartości na dane typu INT. Blok funkcyjny →INT nie przesyła sygnału wyjściowego jedynie wtedy gdy liczba, której konwersja ma zostać dokonana, wykracza poza dopuszczalny zakres (od -32.768 do +32.767). Konwersja zostaje wtedy dokonana na wartość minimalną lub maksymalną.
→TDINT	Konwersja wartości na dane typu DINT (liczba całkowita podwójnej precyzji). Blok funkcyjny →TDINT nie przesyła sygnału wyjściowego jedynie wtedy gdy liczba, której konwersja ma zostać dokonana wykracza poza dopuszczalny zakres (od -2.147.483.648 do +2.147.483.647). Konwersja zostaje wtedy dokonana na wartość minimalną lub maksymalną.
→REAL*	Konwersja wartości na daną typu REAL. Blok funkcyjny →REAL zawsze przesyła sygnał wyjściowy.
→→TRINT*	Obcięcie, zmiana liczby typu REAL na liczbę 16-bitowa typu INT. Zakres od -32.768 do +32.767
→→TRDINT*	Obcięcie, zmiana liczby typu REAL na liczbę 32-bitowa typu DINT. Zakres od -2.147.483.648 do +2.147.483.647

* Funkcje te wymagają co najmniej CPU 350.

Ad. 9. Funkcje sterujące

Tab. 4.14. Funkcje sterujące w sterownikach 90-30

Symbol	Opis
CALL	Przywołanie określonej procedury podczas wykonywania programu sterującego. Funkcja CALL zawsze przesyła sygnał wyjściowy.
DOIO	Natychmiastowe uaktualnienie stanu wyszczególnionych wejść i wyjść sterownika. Kopia stanu wejść i wyjść może zostać zapisana w pamięci.
PIDISA	Regulator proporcjonalno – całkująco - różniczkujący typu ISA (standardowy).
PIDIND	Regulator proporcjonalno - całkująco - różniczkujący typu IND (o niezależnych wyrazach).
END	Instrukcja oznaczająca tymczasowy koniec programu. Program sterujący jest wykonywany od pierwszego do ostatniego szczebla drabiny logicznej lub do napotkania instrukcji END, w zależności od tego, który z tych elementów pojawi się pierwszy. Instrukcja ta jest przydatna przy wykrywaniu błędów podczas uruchamiania programu.
COMMENT	Komentarz. Tekst może być wpisany po użyciu klawisza ZOOM.
SVCREQ	Uruchomienie jednej ze specjalnych funkcji sterownika (odczyt/zmiana aktualnego czasu i daty, wymazywanie tabeli błędów działania sterownika, odczyt czasu pracy od momentu włączenia sterownika itp.).
MCRN	Instrukcja powodująca wykonanie wszystkich szczebli programu sterującego zawartych między instrukcją MCRN a odpowiadającą jej instrukcją ENDMCRN, bez dopływu sygnału z szyny zasilającej. Instrukcja MCRN może być zapętłona z innymi instrukcjami MCRN. Jest ona zawsze ostatnią instrukcją szczebla.

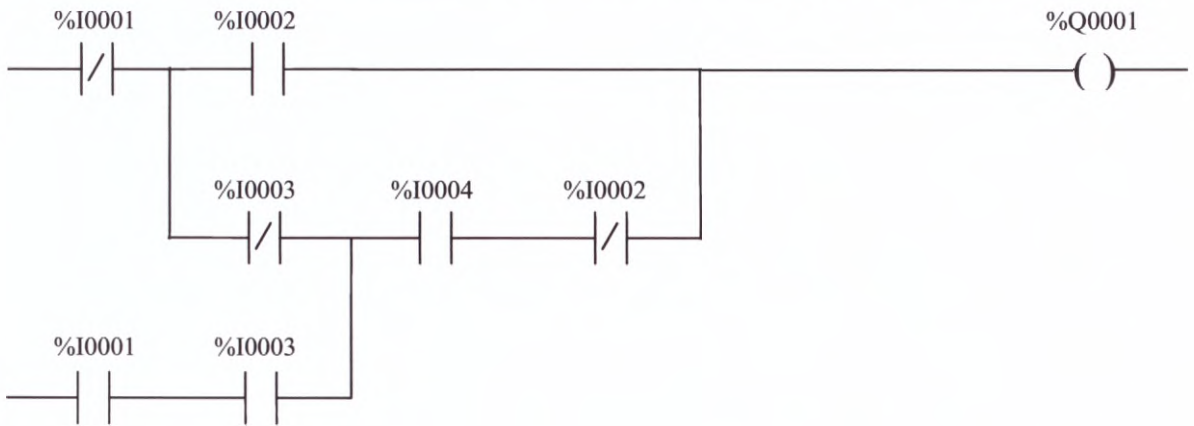
ENDMCRN	Koniec fragmentu programu, którego początek jest zawsze zaznaczony przez instrukcję MCRN. Instrukcja jest zawsze ostatnią instrukcją szczebla.
JUMPN	Instrukcja skoku do miejsca drabiny logicznej programu oznaczonego etykietą LABELN. Może być zapętlona z innymi instrukcjami JUMPN i MCRN.
LABELN	Etykieta skoku dla funkcji JUMPN.
MCR	Instrukcja powodująca wykonanie wszystkich szczebli programu sterującego zawartych między instrukcją MCR a odpowiadającą jej instrukcją ENDMCR, bez dopływu sygnału z szyny zasilającej. Instrukcja MCR nie może być zapętlona z innymi instrukcjami MCR. Jest ona zawsze ostatnią instrukcją szczebla.
ENDMCR	Koniec fragmentu programu, którego początek jest zawsze zaznaczony przez instrukcję MCR. Instrukcja jest zawsze ostatnią instrukcją szczebla.
JUMP	Instrukcja skoku do miejsca drabiny logicznej programu oznaczonego etykietą LABEL. Nie może być zapętlona z innymi instrukcjami JUMP i MCR.
LABEL	Etykieta skoku dla funkcji JUMP.

4.4. Podstawowe zasady przy pisaniu programu sterującego w języku LD na sterownikach GE Fanuc

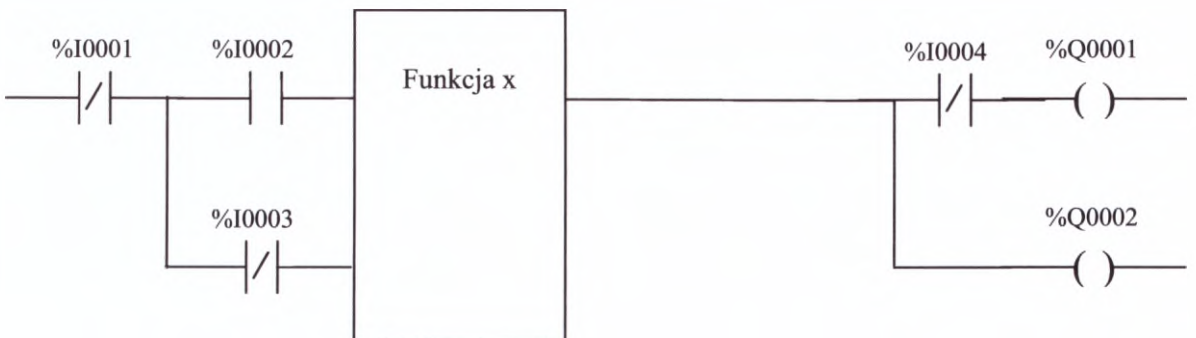
Program sterujący w języku drabinkowym LD składa się z tzw. **szczebli**, które zbudowane są z elementów logicznych. Mogą to być zarówno odpowiednio połączone styki i przekaźniki, jak i bloki funkcyjne. Podczas pisania programu (przy użyciu oprogramowania narzędziowego LM90) należy przestrzegać następujących zasad:

1. Program wykonywany jest wg kolejności szczebli występujących w schemacie drabinkowym, od szczebla pierwszego do ostatniego.
2. Sygnał (prąd) przepływa od lewej strony szczebla (od szyny zasilającej) do prawej.
3. Jeśli w szczeblu występują połączenia równoległe, to najpierw sprawdzana jest linia położona najniżej.
4. W każdym szczeblu może występować maksymalnie osiem linii równoległych.
5. W każdej linii może znajdować się do dziesięciu elementów połączonych szeregowo.
6. Przełączniki występują zawsze po prawej stronie na końcu linii.
7. W jednym szczeblu nie mogą wystąpić przełączniki uruchamiane przez różną kombinację styków (przez różne funkcje logiczne). Przełączniki takie muszą znajdować się w oddzielnych szczeblach drabiny.
8. Jeśli szczebel zawiera przełącznik załączany z boczem sygnału sterującego to powinien to być jedyny przełącznik w tym szczeblu.
9. Szczebel musi zawierać przynajmniej jeden styk przed przełącznikiem, instrukcją, blokiem funkcyjnym, funkcją lub linią pionową – aby spełnić ten wymóg, można użyć styk otwarty ze zmienną systemową %S007 (ALW_ON).
10. Konstrukcja szczebla musi być poprawna z logicznego punktu widzenia.
11. Nie może wystąpić rozgałęzienie mające początek lub koniec wewnątrz innego rozgałęzienia – rys. 4.1.
12. Nie wolno budować szczebla tak, aby sygnał (prąd) przesyłany był w lewą stronę.

13. Każda funkcja i blok funkcyjny posiadają wejście uaktywniające (enable), które jest zawsze zaznaczone na schemacie jako wejście główne. Funkcja lub blok wykonywane są tylko wtedy, gdy do tego wejścia dopływa sygnał (prąd).
14. W szczeblu zawierającym funkcje lub blok funkcyjny nie mogą wystąpić rozgałęzienia rozpoczynające się od linii pionowej z wyjątkiem prowadzących bezpośrednio do przekaźników – rys. 4.2.
15. W szczeblu zawierającym funkcje lub blok funkcyjny po prawej stronie bloku nie mogą wystąpić żadne styki – rys. 4.2.
16. Funkcje posiadają zwykle wyjście umożliwiające przekazanie sygnału (prądu) do elementów znajdujących się po prawej stronie funkcji. Wyjście to znajduje się na schemacie po prawej stronie u góry i jest oznaczone jako OK. W zależności od rodzaju funkcji, sygnał na wyjściu OK. może pojawić się w każdym cyklu sterowania, w którym funkcja jest aktywna, bądź tylko wtedy, gdy funkcja została wykonana poprawnie. Dla większości funkcji wyjście to jest opcjonalne i nie musi być używane.
17. Używanie rejestrów, z których korzystają bloki (liczniki, czasomierze, funkcje PID) w innych blokach może prowadzić do ich niepoprawnego działania.



Rys. 4.1. Przykład niepoprawnego rozgałęzienia



Rys. 4.2. Przykład niepoprawnego przyłączenia bloku funkcyjnego i przekaźnika Q0001

18. Można używać procedury (podprogramy). Procedury mogą być zagnieżdżone (do 8 poziomów). Procedura może wywoływać sama siebie, nie jest to jednak zalecane rozwiązanie.

Większość z przedstawionych powyżej zasad obowiązuje także podczas programowania przy użyciu oprogramowania VersaPro v. 2.01 i CIMPLICITY Machine Edition (z wyjątkiem punktów 4,5,9,11).

Przed przystąpieniem do edycji programu sterującego zaleca się przeprowadzenie tzw. **deklaracji zmiennych**. Jest to nadanie zmiennym występującym w programie nazw symbolicznych, wygodniejszych dla programisty. Operację tą przeprowadza się w tablicy deklaracji zmiennych, która składa się z trzech kolumn:

- Reference;
- Nickname;
- Reference Description.

Zakładając, że obiekt z rys. 2.3. jest wyposażony w dwustanowe czujniki temperatury $T1-T3$, grzałkę G sterowaną za pomocą sygnału analogowego oraz wentylator W sterowany za pomocą sygnału dwustanowego, przykładowa deklaracja mogłaby wyglądać następująco:

VARIABLE DECLARATION TABLE

<u>REFERENCE</u>	<u>NICKNAME</u>	<u>REFERENCE DESCRIPTION</u>
%I0001	T1	Czujnik 20 stopni
%I0002	T2	Czujnik 60 stopni
%I0003	T3	Czujnik 80 stopni
%Q0008	W	Wentylator
%AQ0001	G	Grzałka

4.5. Przykłady programowania

Poniżej przedstawiony zostanie zapis kilku funkcji logicznych w języku drabinkowym. Omówiony zostanie także sposób użycia bloku przekaźnika czasowego z pamięcią, funkcji mnożenia dwóch liczb i bloku PIDISA.

4.5.1. Zapis w języku LD funkcji logicznych

Przykład

Napisać w języku drabinkowym program sterujący składający się z trzech następujących funkcji logicznych:

$$z_1 = a + b$$

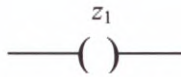
$$z_2 = \bar{a} \cdot c$$

$$z_3 = (a \cdot \bar{d} + c \cdot d) \cdot b \cdot \bar{e}$$

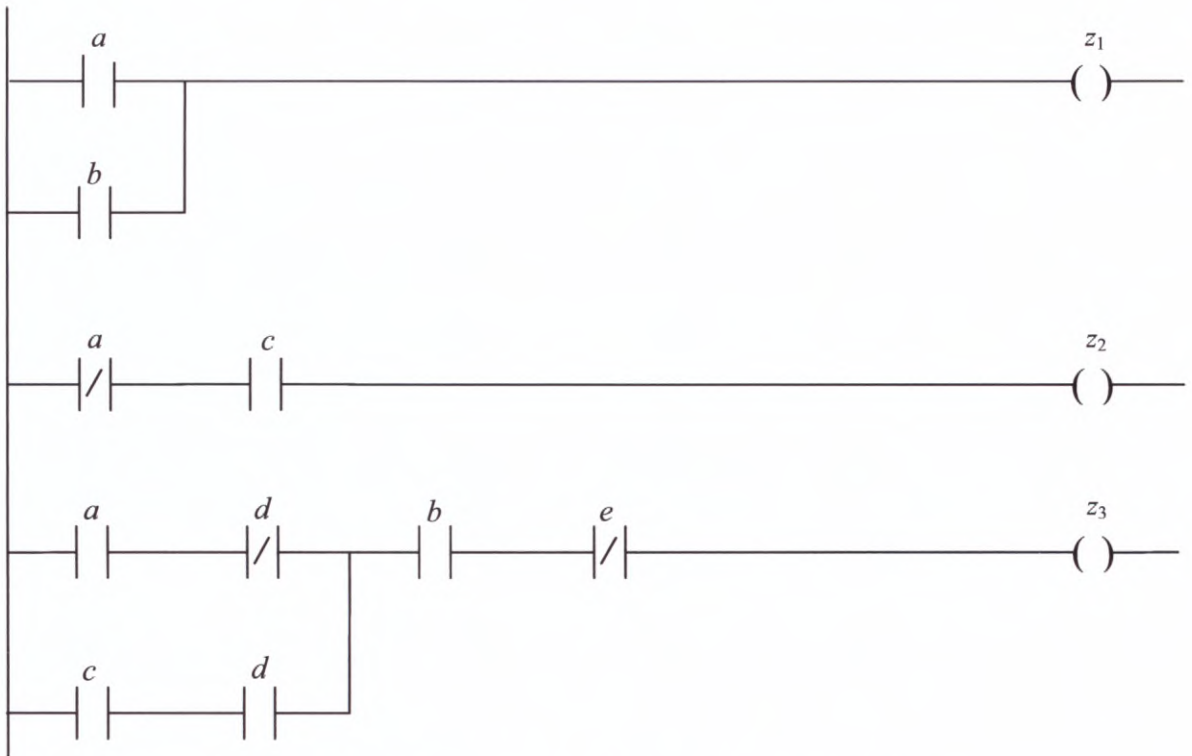
W języku drabinkowym (patrz tab. 4.6. str. 52) sygnałowi:

a odpowiada symbol 

\bar{a} odpowiada symbol 

natomiast wynikowi funkcji z_1 

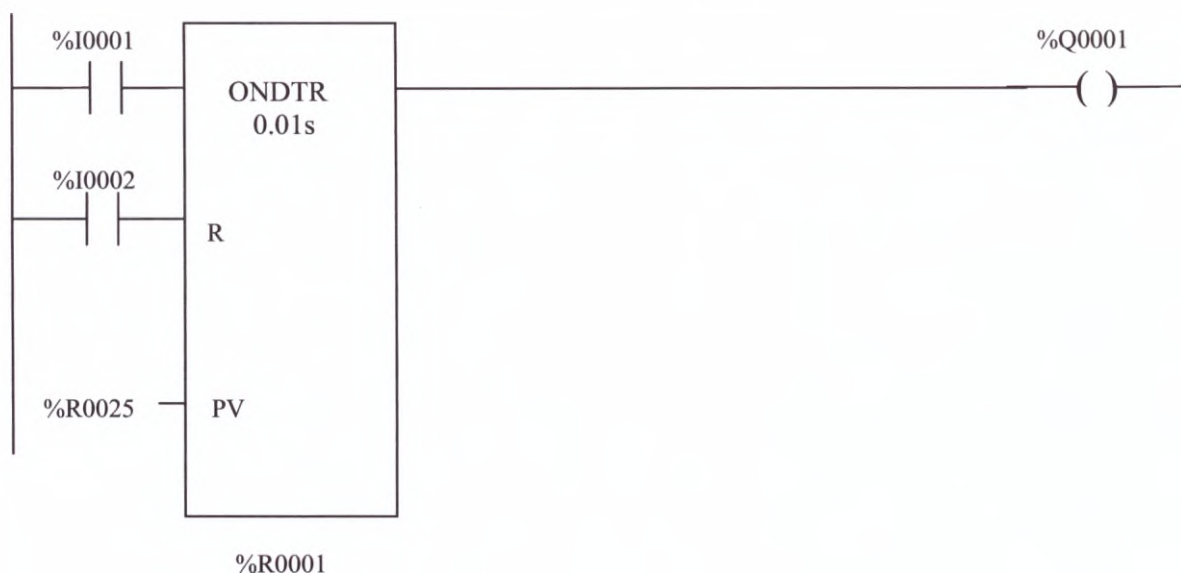
Drabinka reprezentująca powyższy program będzie składała się z trzech oddzielnych szczebli przedstawionych poniżej – rys. 4.3.



Rys. 4.3. Przykład programu realizującego typowe funkcje logiczne

4.5.2. Wykorzystanie bloku funkcyjnego ONDTR

Przełącznik czasowy z pamięcią przedstawiony na rys. 4.4 zlicza czas, gdy dopływa do niego sygnał tj. gdy zwarty jest styk %I0001. Przełącznik wstrzymuje zliczanie czasu w momencie gdy sygnał przestaje dopływać. Gdy zacznie dopływać ponownie zliczanie czasu jest kontynuowane. Bieżąca wartość zapisywana jest w rejestrze %R0001. Przesłanie sygnału wyjściowego – włączenie przełącznika %Q0001 – następuje, gdy zliczona wartość osiągnie wartość zadaną, podaną na wejście PV. W prezentowanym przykładzie jest to aktualna wartość rejestru %R0025, ale może to być także wartość wejścia/wyjścia analogowego lub wartość stała. Czas może być zliczany w dziesiątych, setnych lub tysięcznych częściach sekundy i okres ten jest podany poniżej nazwy bloku. Zakres mierzonej wartości wynosi $0 \div +32.767$ jednostek czasu. Bieżąca wartość pamięci przełącznika może być wyzerowana poprzez przesłanie sygnału na wejście zerujące (R) – załączenie styku %I0002. Blok

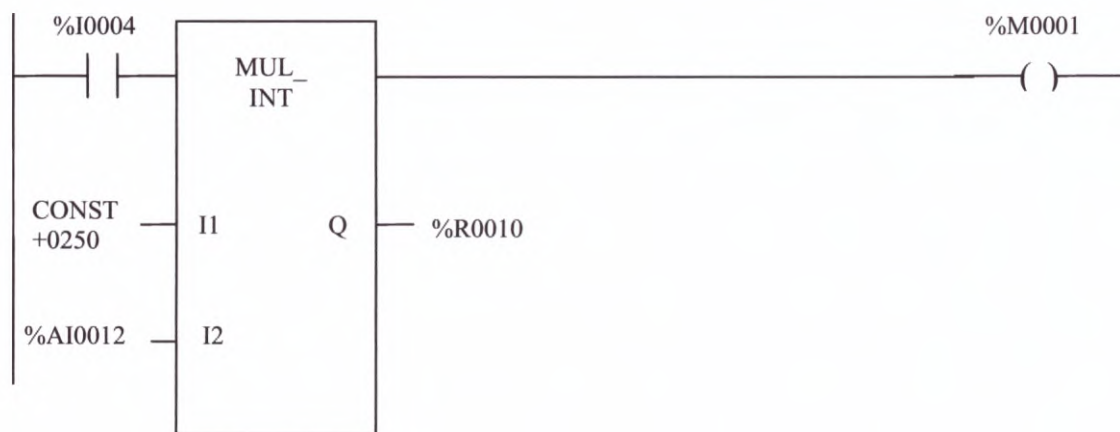


Rys. 4.4. Zastosowanie bloku funkcyjnego ONDTR

funkcyjny ONDTR wykorzystuje 3 kolejne słowa pamięci sterownika, czyli na rys. 4.4. od %R0001 do %R0003.

4.5.3. Wykorzystanie bloku funkcyjnego MUL

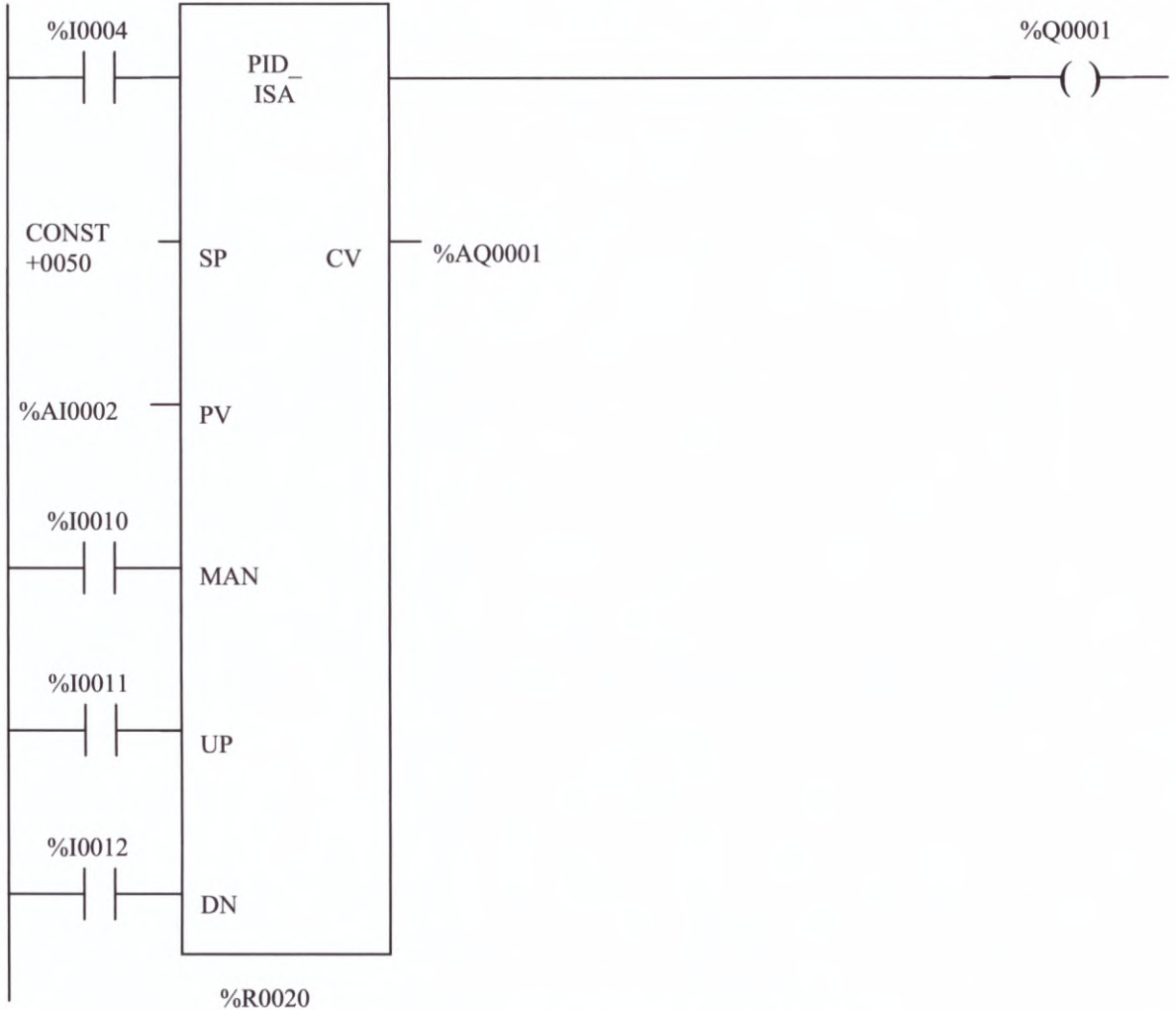
Blok funkcyjny MUL (rys. 4.5.) wykonuje mnożenie dwóch liczb podanych na wejścia I1 i I2 po warunkiem, że dopływa do niego sygnał – załączony jest styk %I0004. Na wejścia podane mogą być stałe, wartości wejść/wyjść analogowych lub rejestrów. W prezentowanym przykładzie stała wartość 250 jest mnożona przez aktualną wartość wejścia analogowego %AI0012. Wynik przesyłany jest na wyjście Q – w przykładzie zostanie zapamiętany w rejestrze %R0010 (może to być także np. wyjście analogowe AQ). Funkcja może być wykonana na danych typu INT, DINT lub REAL (dla jednostki centralnej od CPU 350 wzwyż) i ich oznaczenie jest dodawane do nazwy bloku. Blok funkcyjny MUL przesyła sygnał wyjściowy (załączy przekaźnik %M0001) gdy wynik działania nie przekroczy dopuszczalnego zakresu wartości. Użycie wyjścia (przekaźnika %M0001) nie jest wymagane.



Rys. 4.5. Zastosowanie bloku funkcyjnego MUL

4.5.4. Podstawy użycia bloku funkcyjnego PID

Jednym z najczęściej wykorzystywanych algorytmów w układach regulacji jest algorytm PID. Ponieważ sterowniki PLC są urządzeniami cyfrowymi, regulator PID realizowany jest w nich w wersji dyskretnej, nie ciągłej. Blok PID występuje w sterownikach GE Fanuc w dwóch odmianach: standardowej – PIDISA i o niezależnych wyrazach – PIDIND. Uruchomienie bloku (załączenie styku %I0001 na rys. 4.6.) powoduje rozwiązanie



Rys. 4.6. Zastosowanie bloku funkcyjnego PID

równania zamkniętego układu regulacji. W cyklu pracy sterownika wykonywana jest tylko jedna iteracja algorytmu. Na podstawie dwóch wartości wejściowych: **wartości zadanej** (SP – set point) i **wartości bieżącej** – wielkości regulowanej (PV – process variable), oraz zadanych parametrów regulatora, blok wylicza **wartość sygnału sterującego** (CV – control variable). Blok funkcyjny PID wykorzystuje 40 rejestrów pamięci sterownika, w których przechowuje zbiór parametrów regulatora. Rejestry te nie mogą być używane przez inny blok PID.

Na schemacie LD blok posiada siedem wielkości wejściowych:

- wejście sygnału logicznego uruchamiające blok (enable) (na rys. 4.6. styk %I0004);
- wartość zadaną SP (na rys. 4.6. stała CONST 0050);
- wielkość regulowaną PV (na rys. 4.6. wejście analogowe %AI0002);
- przełącznik automatycznego/ręcznego trybu pracy MAN (na rys. 4.6. styk %I0010);
- zwiększenie wielkości wyjściowej w trybie pracy ręcznej UP (na rys. 4.6. styk %I0011);
- zmniejszenie wielkości wyjściowej w trybie pracy ręcznej DN (na rys. 4.6. styk %I0012);
- adres (register) pierwszego rejestru lokalizującego blok 40 rejestrów używanych przez blok (na rys. 4.6. rejestr %R0020),

oraz dwie wyjściowe:

- wyjście sygnału przesyłanego po pomyślnym wykonaniu zadania (ok) (na rys. 4.6. przekaźnik %Q0001);
- sygnał sterujący CV (na rys. 4.6. wyjście analogowe %AQ0001).

Gdy do wejścia „enable” dopływa sygnał, a jednocześnie nie dopływa do wejścia MAN realizowany jest algorytm PID na podstawie wartości parametrów SP i PV, a wynik przesyłany jest na wyjście CV. Jeśli algorytm zostanie wykonany bez przeszkód na wyjście „ok” wysyłany jest sygnał. Gdyby z jakichkolwiek powodów algorytm PID nie został zrealizowany, blok nie przesyła sygnału wyjściowego.

Gdy sygnał dopływa jednocześnie do wejścia „enable” i wejścia MAN blok pracuje w trybie ręcznym. Parametr wyjściowy CV utrzymuje swoją bieżącą wartość, która może być regulowana poprzez podawanie sygnału na wejścia UP lub DN. Blok funkcyjny PID zapewnia bezuderzeniowe przejście pomiędzy trybami sterowania ręcznego i automatycznego. W tab. 4.15. podano najważniejsze parametry bloku PID, do których ma dostęp użytkownik, wraz z ich adresami.

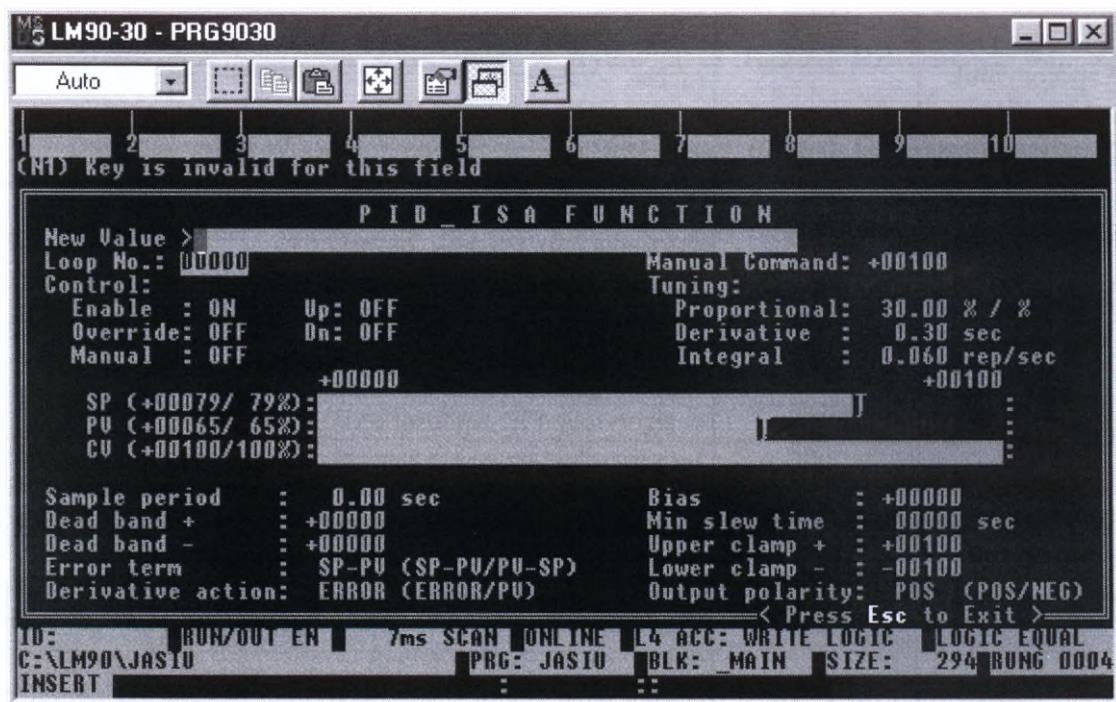
Tab. 4.15. Podstawowe parametry konfiguracyjne bloku PID

Parametr	Adres w sterowniku*	Opis
Nr układu regulacji	%rejestr	Liczba całkowita dodatnia pozwalająca na identyfikację układu regulacji poprzez jego numer. Zastosowanie nr układu regulacji jest opcjonalne.
Algorytm	%rejestr + 1	Liczba całkowita bez znaku ustawiana przez sterownik w celu rozróżnienia realizowanego algorytmu. Gdy wartość równa się 1 to realizowany jest algorytm standardowy ISA, gdy 2 to algorytm o niezależnych wyrazach IND.
Okres próbkowania (impulsowania)	%rejestr + 2	Odstęp czasowy (mierzony w setnych częściach sekundy) pomiędzy dwoma kolejnymi wykonaniami bloku PID. Algorytm jest realizowany w zadanych odstępach czasu, a blok zlicza czas, który upłynął od ostatniego wykonania algorytmu z dokładnością 0,0001 s. Jeśli wartość tego parametru wynosi 0 algorytm PID wykonywany jest w każdym cyklu sterownika, w którym do bloku dopływa sygnał uruchamiający „enable”. Zakres: 0 ÷ 10,9 min.

Górna (dolna) granica strefy martwej	%rejestr + 3 (%rejestr + 4)	Wartość całkowita ze znakiem określająca górną (dolną) granicę strefy nieczułości, to jest maksymalna wartość bezwzględna, o jaką może odchylić się wartość bieżąca od wartości zadanej w górę (w dół). Jeśli uchyb wielkości regulowanej (różnica między PV i SP) mieści się w zakresie wyznaczonym przez te parametry algorytm PID jest realizowany przy przyjęciu założenia, że uchyb jest równy 0. Gdy strefa nieczułości nie jest wymagana wartość parametrów powinna wynosić 0.
Współczynnik wzmocnienia proporcjonalnego	%rejestr + 5	Liczba całkowita ze znakiem określająca współczynnik wzmocnienia regulatora PID. Jednostka 0,01%%; zakres: 0 ÷ 327,67%%
Czas różniczkowania	%rejestr + 6	Wartość całkowita ze znakiem, która stanowi czas różniczkowania. Jednostka 0,01 s; zakres: 0 ÷ 327,67 s.
Częstotliwość całkowania	%rejestr +7	Wartość całkowita ze znakiem, która stanowi częstotliwość całkowania Jednostka - liczba całkowania/s; zakres: 0 ÷ 327,67 1/s.
Przesunięcie punktu pracy	%rejestr +8	Liczba całkowita ze znakiem dodawana do obliczonej w algorytmie PID wartości wyjścia regulatora.
Górna (dolna) granica sygnału ustawiającego	%rejestr +9 (%rejestr +10)	Liczba całkowita ze znakiem określająca górny (dolny) zakres dla wielkości wyjściowej wypracowanej przez regulator. Gdy wypracowany sygnał wyjściowy jest większy (mniejszy) od górnej (dolnej) granicy jako wartość CV przyjmowana jest wartość górnej (dolnej) granicy sygnału ustawiającego. Wartość górnej granicy musi być niemniejsza od granicy dolnej.
Minimalny czas zmiany sygnału ustawiającego o 100%	%rejestr +11	Liczba całkowita dodatnia określająca minimalny czas zmiany sygnału wyjściowego regulatora w zakresie od 0% do 100%. Jeśli obliczona w algorytmie wartość wyjściowa zmienia się szybciej jest ona korygowana tak, aby spełnić warunek wynikający z tego parametru. Jeśli ograniczenie prędkości wyjścia nie jest istotne parametr powinien być ustawiony na 0. Jednostki – s/pełny zakres; zakres: 0 ÷ 32.767 s.
Parametr konfiguracyjny	%rejestr +12	5 bitów tego słowa służy do modyfikowania kolejnych parametrów regulatora PID. Pozostałe bity powinny być ustawione na 0. Bit 0 - określa sposób obliczania uchybu. Gdy jego wartość logiczna wynosi 1 uchyb jest obliczany ze wzoru SP-PV, gdy 1 to uchyb jest równy PV-SP Bit 1 - polaryzacja sygnału ustawiającego CV. Gdy bit ten równa się 0 to CV przyjmuje bezpośrednią wartość wyniku algorytmu PID, gdy wynosi 1 to CV przyjmuje polaryzację przeciwną. Bit 2 - sposób różniczkowania – gdy wartość bitu wynosi 0 różniczkowany jest uchyb, w przeciwnym razie sygnał PV. Bit 3 - związany z przekroczeniem strefy nieczułości. Bit 4 - związany z mechanizmem zapobiegającym przekroczeniu wartości granicznych.

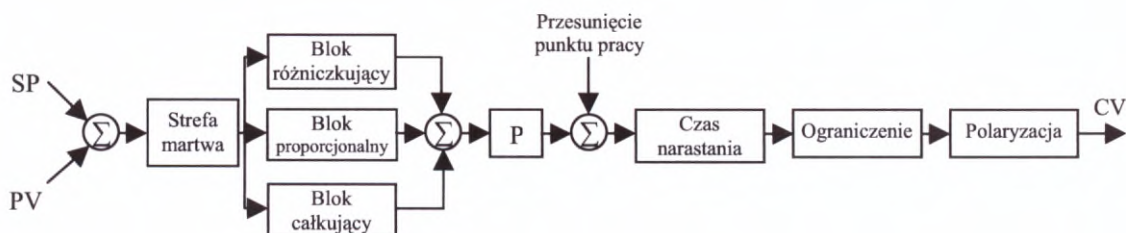
* w kolumnie „Adres w sterowniku” oznaczenie „%rejestr” oznacza adres pierwszego rejestru bloku danych (%R0020 na rys. 4.6.).

Ustawienia parametrów konfiguracyjnych przedstawionych w tab. 4.15. można dokonać bądź w programie przy użyciu funkcji przesuwania danych (MOVE, BLKMOV – tab. 4.11.), bądź z poziomu oprogramowania narzędziowego Logicmaster 90 korzystając z ekranu konfiguracyjnego bloku PID – rys. 4.7. Możliwe jest także wykorzystanie w tym celu systemu monitoringu – patrz rozdział 7.

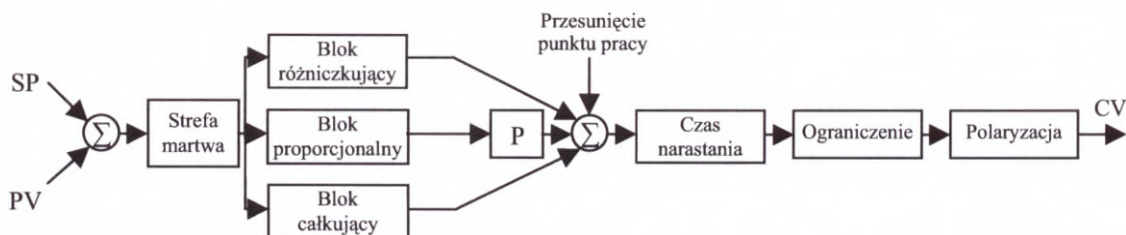


Rys. 4.7. Ekran konfiguracyjny bloku funkcyjnego PID

Na rys. 4.8. przedstawiono schemat blokowy algorytmu PIDISA, a na rys. 4.9. algorytmu PIDIND. Blok P realizuje mnożenie przez współczynnik wzmocnienia proporcjonalnego.



Rys. 4.8. Schemat blokowy algorytmu PIDISA



Rys. 4.9. Schemat blokowy algorytmu PIDIND

Przesunięcie punktu pracy może być wykorzystywane nie tylko do przeniesienia wartości wyjścia na odpowiedni poziom, ale także w układach regulacji z korekcją od zakłócenia, zwanych sterowaniem ze sprzężeniem w przód (Feedforward Control).

Literatura

- [1] Kwaśniewski J., Programowalne sterowniki przemysłowe w systemach sterowania, Kraków, 1999
- [2] Legierski T., Kasprzyk J., Wyrwał J., Hajda J., Programowanie sterowników PLC, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 1998
- [3] Logicmaster™ 90. Series 90™-30/20/Micro Programming Software. User's Manual, GFK-0466K, GE Fanuc Automation, May, 1997
- [4] Mandado E., Marcos J., Perez S. A., Programmable Logic Devices and Logic Controllers, Prentice Hall, 1996
- [5] PN-EN 61131-3:1998 Sterowniki programowalne. Języki programowania
- [6] Series 90-30/20/Micro. Programmable Controllers. Reference Manual. GFK-0467J, GE Fanuc Automation, May, 1997
- [7] Series 90™-30 System manual, GFK-1411, GE Fanuc Automation, November, 1997
- [8] Stec Ł., Regulatory PID (proporcjonalno-całkująco-różniczkujące) w sterownikach PLC, Biuletyn Automatyki ASTOR, 2/96 (8), str. 3,10
- [9] Sterownik programowalne Serii 90-20 i 90-30. Kurs programowania, Astor, Kraków, 1993
- [10] Sterownik programowalne Serii 90-20 i 90-30. Podręcznik programisty, Astor, Kraków, 1993

5. Schemat funkcji sekwencyjnej SFC

Na początku lat siedemdziesiątych XX w. wystąpiła potrzeba modelowania bardzo złożonych systemów sterowania sekwencyjnego dla celów projektowania programów sterownikowych. W 1977 stawianym wymaganiom sprostał opis nazywany metodą **Grafcet**. Metoda ta umożliwia zapis algorytmu procesu, ale wymaga modelowania (wyznaczenia) **tranzycji** tzn. logicznych zależności przyczynowo-skutkowych, określających realizację poszczególnych operacji procesu. W normie IEC 61131-3 przyjęto formalizm SFC (Sequential Function Chart – schemat funkcji sekwencyjnej). Obie te metody tj. metoda Grafcet jak i SFC są do siebie bardzo podobne, obie bowiem stanowią modyfikację sieci Petriego typu P/T (Pozycja/Tranzycja).

5.1. Podstawy SFC

Metoda SFC jest metodą przeznaczoną specjalnie do opisu sterowania przemysłowych procesów sekwencyjnych. Jest to metoda graficzna, w której działanie układu sekwencyjnego jest przedstawione jako sekwencja (graf) **kroków (etapów)** i **tranzycji (przejęć)**. Z każdym etapem skojarzony jest zbiór odpowiednich działań, a każdemu przejściu towarzyszy warunek przejścia.

Krok (etap – step) można zdefiniować jako zestaw działań sterownika skojarzonych z etapem procesu. Krok może być **aktywny** lub **nieaktywny**. Aktywność kroku określa wskaźnik kroku, który przybiera wartość 1 gdy krok jest aktywny i 0 gdy krok jest nieaktywny. Inna zmienna informuje o czasie trwania danego kroku – określa ona czas aktywności kroku. Jeśli krok jest nieaktywny jej wartość pozostaje bez zmian. Stan początkowy procesu określa **krok początkowy** (initial step). Jego symbolem jest prostokąt rysowany linią podwójną (rys. 5.1. a.). W sieci SFC może wystąpić tylko jeden krok początkowy. Pozostałe kroki w sieci nazywane są **krokami regularnymi** (regular step), a ich symbolem jest prostokąt (rys. 5.1. b.).



Rys. 5.1. Symbol kroku a) początkowego
b) regularnego

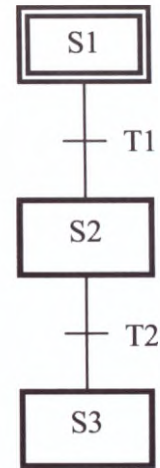
Przejście (tranzycja) reprezentuje warunki logiczne realizacji poszczególnych kroków programu. Spełnienie warunków przejścia (występującego bezpośrednio po aktualnie aktywnym kroku) powoduje, że krok (kroki) poprzedzający staje się nieaktywny, a aktywny staje się krok (kroki) bezpośrednio występujący po danym przejściu. Symbolem przejścia jest poziomy odcinek prostopadły do pionowych kierunków łączenia kroków – rys. 5.2.



Rys. 5.2. Symbol przejścia

Aby skorzystać z metody SFC należy zatem narysować algorytm (graf) działania układu poprzez odpowiednie łączenie kroków i etapów – rys. 5.3. Następnie, przy użyciu jednego z języków programowania, trzeba określić działanie sterownika związane z każdym z kroków, oraz logiczne warunki przejść związane z tranzycjami.

Sposób działania sieci SFC przedstawionej na rys. 5.3. jest następujący. Po zainicjowaniu sterownika lub bloku funkcyjnego zawierającego sieć aktywny jest krok początkowy S1. Sterownik wykonuje działania związane z krokiem S1. Po spełnieniu warunków logicznych określających przejście T1 następuje wyłączenie kroku S1 (staje się on nieaktywny) oraz załączenie kroku S2 (staje się on aktywny). Sterownik wykonuje działania związane z krokiem S2. Następnie po spełnieniu warunków logicznych określających przejście T2 następuje wyłączenie kroku S2 oraz załączenie kroku S3. Sterownik wykonuje działania związane z krokiem S3.

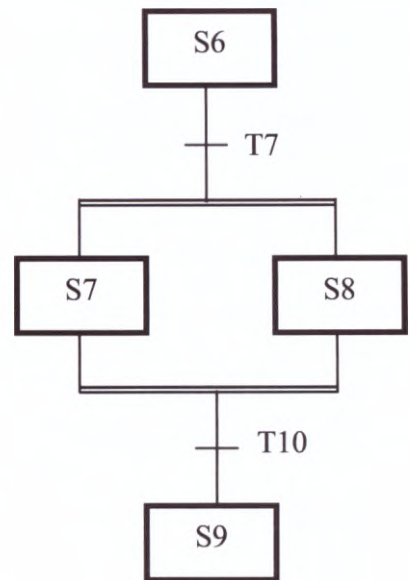


Rys. 5.3. Przykład sieci SFC

Jak wspomniano wcześniej, warunki początkowe grafu SFC są określone przez etap początkowy. Jest on w stanie aktywnym z chwilą inicjacji programu użytkownika lub bloku funkcyjnego zawierającego sieć SFC. Zmiana stanu procesu polega na przechodzeniu pomiędzy kolejnymi, bezpośrednio połączonymi krokami i zależy od warunków przejścia między nimi. Przejście jest dostępne (warunki logiczne z nim związane są analizowane przez sterownik) wtedy i tylko wtedy gdy wszystkie bezpośrednio poprzedzające je etapy są aktywne. Kasowanie przejścia występuje wtedy, gdy jest ono aktywne i jest spełniony warunek przejścia z nim związany. Kasowanie przejścia powoduje dezaktywację (wyłączenie) wszystkich bezpośrednio poprzedzających go kroków i aktywację wszystkich kroków występujących bezpośrednio po nim.

Podczas budowania grafu SFC należy przestrzegać następujących zasad:

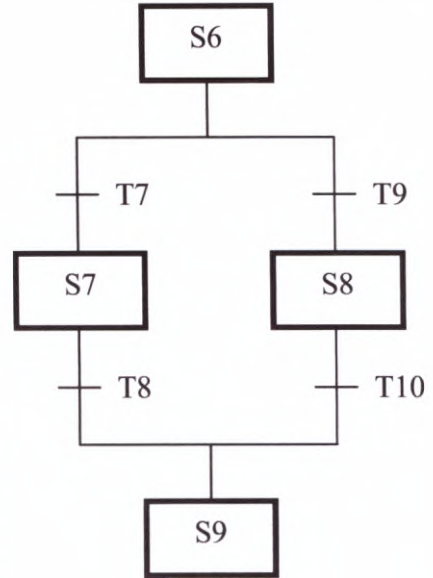
1. Niedozwolone jest bezpośrednie łączenie dwóch kroków. Kroki zawsze musi rozdzielać przejście.
2. Niedozwolone jest bezpośrednie łączenie dwóch przejść. Przejścia zawsze musi rozdzielać krok.
3. Możliwe jest tworzenie **procedur współbieżnych** (jednoczesnych, równoczesnych) – rys. 5.4. W przypadku procedur współbieżnych spełnienie pewnego warunku przejścia (na rysunku T7) prowadzi do jednoczesnej aktywacji kilku różnych kroków (w przykładzie S7 i S8). Sekwencje takich kroków są rysowane równoległe. Po równoczesnej aktywacji sekwencji współbieżnych dalsze wykonywanie każdej z nich przebiega niezależnie od pozostałych. W każdej takiej sekwencji może wystąpić dowolna liczba kroków. Rozpoczęcie i zakończenie procedur współbieżnych przedstawia się za pomocą podwójnej linii poziomej. Tranzycja rozpoczynająca procedury współbieżne (T7) umieszczona jest nad podwójną linią poziomą, natomiast tranzycja kończąca procedury współbieżne (T10) umieszczona jest pod podwójną linią poziomą.



Rys. 5.4. Procedury współbieżne

4. Istnieje możliwość realizacji **wyboru sekwencji** (rys. 5.5.). Rozpoczęcie i zakończenie wyboru sekwencji przedstawia się za pomocą pojedynczej linii

poziomej. Tranzycje rozpoczynające wybór sekwencji (T7, T9) umieszczone są pod pojedynczą linią poziomą, natomiast tranzycje kończąca wybór sekwencji (T8, T10) umieszczone są nad pojedynczą linią poziomą. Każda gałąź może posiadać dowolną liczbę kroków. Szczególny przypadek występuje gdy jedna z gałęzi nie zawiera żadnych kroków. Jest to tzw. **ominięcie sekwencji**. W przypadku przedstawionym na rys. 5.5. spełnienie warunku przejścia T7 spowoduje uaktywnienie kroku S7. Gałąź zawierająca krok S8 zostanie pominięta. Natomiast spełnienie warunku przejścia T9 spowoduje uaktywnienie kroku S8, gałąź zawierająca krok S7 zostanie pominięta.



Rys. 5.5. Wybór procedury

5. W przypadku procedur współbieżnych nie wolno wprowadzać wyboru sekwencji w taki sposób, aby linia kończąca wybór sekwencji znajdowała się poza (poniżej) wspólną linią kończącą procedury współbieżne.
6. W celu cyklicznej realizacji procedury sekwencyjnej istnieje możliwość zdefiniowania **skoku** (jump). W niektórych oprogramowaniach narzędziowych do budowy grafu SFC skok realizuje się jako gałąź powracającą do jednego z wcześniejszych kroków.

5.2. Przykład sterowania manipulatorem

W niniejszym podrozdziale przedstawiony zostanie prosty przykład sterowania sekwencyjnego ruchami siłowników manipulatora. Najpierw zbudowany zostanie algorytm sterowania w postaci grafu SFC. Następnie, przy użyciu języka drabinkowego, zostaną zdefiniowane działania sterownika związane z poszczególnymi krokami i warunki logiczne dla przejść. Rozwiązanie wykonane zostało w oparciu o oprogramowanie Logicmaster 90 firmy GE Fanuc Automation i odpowiedniej nakładki programowej zawierającej SFC.

Schemat siłowników manipulatora przedstawiono na rys. 5.6. Układ składa się z dwóch siłowników: pionowego A i poziomego B. Z siłownikiem A związane są wyłączniki krańcowe AD i AG, natomiast z siłownikiem B wyłączniki krańcowe BL i BP.

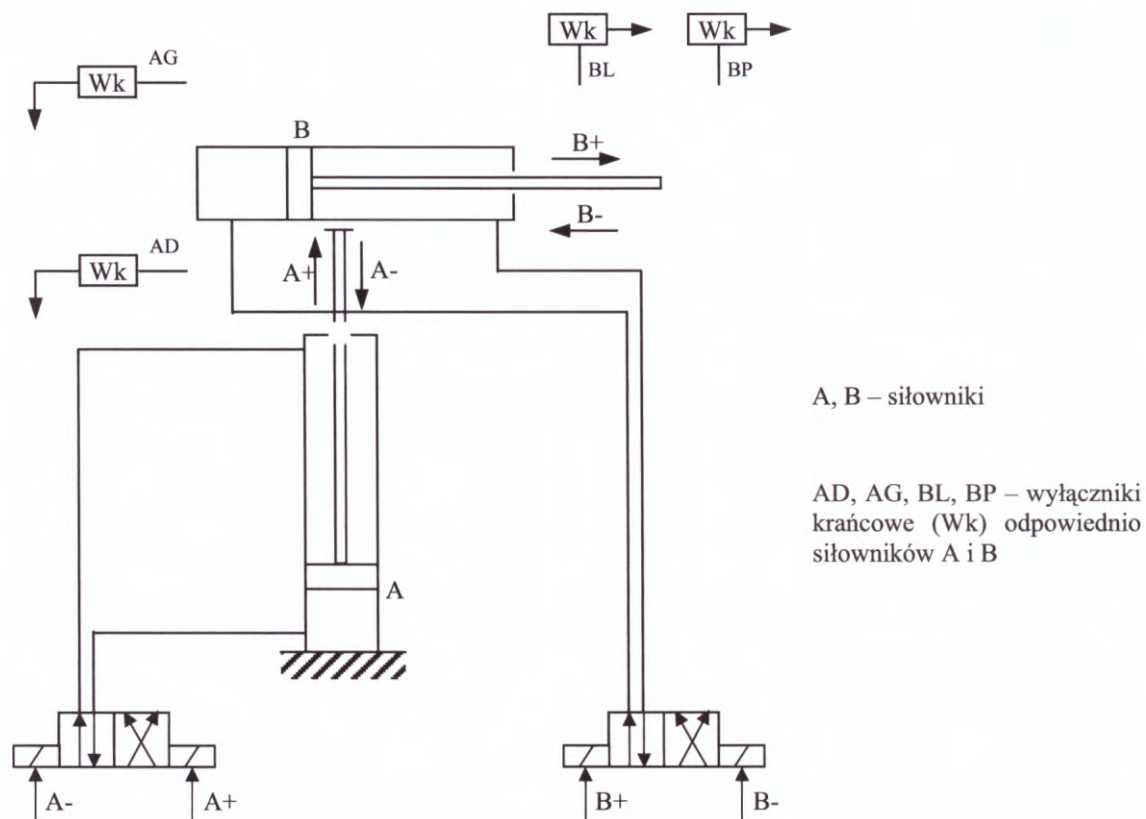
W układzie występują następujące sygnały sterujące:

- A+ powodujący ruch siłownika A do góry;
- A- powodujący ruch siłownika A do dołu;
- B+ powodujący ruch siłownika B w prawo;
- B- powodujący ruch siłownika B w lewo.

Należy zaprogramować układ na realizujący następującą sekwencję:

- pozycja początkowa siłowników: siłownik A w dolnym skrajnym położeniu (AD=1, AG=0), siłownik B w lewym skrajnym położeniu (BL=1, BP=0);
- po naciśnięciu przez operatora przycisku Start (S) włączany jest sygnał sterujący A+;

- osiągnięcie przez siłownik A górnego skrajnego położenia ($AD=0$, $AG=1$) powoduje włączenie sygnału sterującego $B+$;
- po osiągnięciu przez siłownik B skrajnego prawego położenia ($BL=0$, $BP=1$) następuje równoczesne wycofanie siłownika B (sygnał sterujący $B-$) i siłownika A (sygnał sterujący $A-$);
- po powrocie przez siłowniki w położenia początkowe układ jest gotowy do powtórzenia sekwencji.



Rys. 5.6. Schemat układu manipulatora

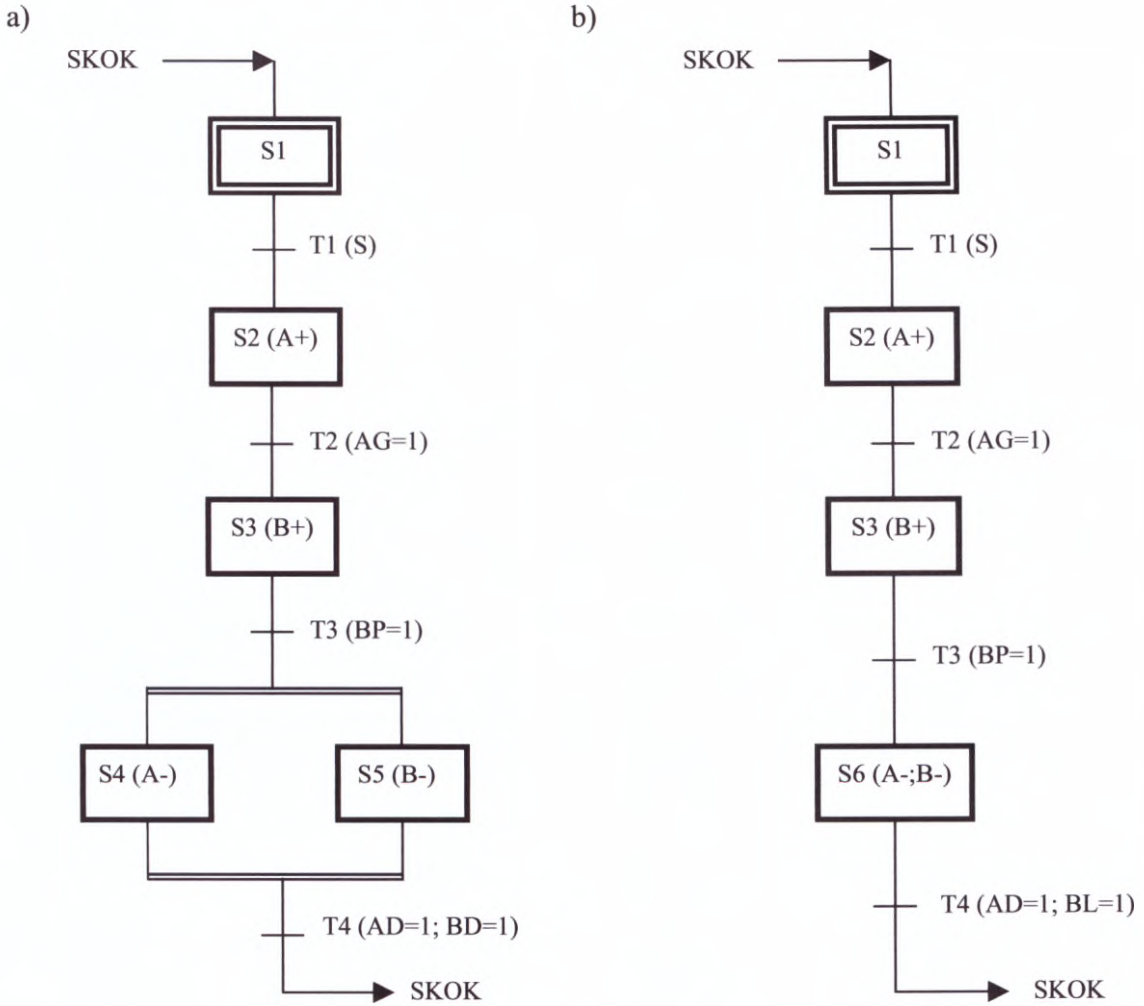
Zadanie zostało rozwiązane dwoma sposobami:

- z wykorzystaniem procedur współbieżnych;
- bez procedur współbieżnych.

Sieci SFC dla obu rozwiązań zostały przedstawione na rys. 5.7. Obok nazw kroków podano w nawiasach działanie sterownika związane z danym krokiem. Analogicznie w nawiasie obok nazwy przejścia podano logiczny warunek przejścia. Obie sieci różnią się jedynie zastąpieniem procedur współbieżnych (kroki S4 i S5 - rys. 5.7.a) jednym krokiem S6 w sieci z rys. 5.7.b.

Poniżej zdefiniowano kroki i warunki przejść w języku drabinkowym. Zgodnie z zasadą 9 podaną w rozdziale 4.4. dotyczącą użycia języka LD w oprogramowaniu Logimaster 90 nie jest możliwe bezpośrednie połączenie przekaźnika do szyny zasilającej.

Aby spełnić ten wymóg formalny podczas definiowania kroków użyto styku normalnie otwartego z przypisaną zmienną systemową %S0007 (ALW_ON). Krok początkowy S1 jest krokiem pustym. Sterownik nie wysyła w nim żadnych sygnałów sterujących, a jedynie czeka na spełnienie warunków logicznych związanych przejściem T1.



Rys. 5.7. Grafy SFC dla zadania sterowania manipulatorem a) z procedurami współbieżnymi b) bez procedur współbieżnych

Przejście T1



Możliwa jest też druga definicja tranzycji T1 sprawdzająca dodatkowo początkowe położenie siłowników.



Krok S2



Tranzycja T2



Krok S3



Tranzycja T3



Krok S4



Krok S5**Tranzycja T4****Krok S6**

Po spełnieniu warunku logicznego związanego z przejściem T4 następuje skok (nazwany „SKOK”) do kroku S1. Sekwencja może zostać wykonana powtórnie.

Należy wspomnieć, że program SFC firmy GE Fanuc dostarcza trzy zmienne, które mogą być pomocne w programowaniu. Są to:

- Sy.t - zmienna typu integer określająca czas aktywności kroku y;
- Sy.x - zmienna bitowa przyjmująca wartość 1 gdy krok y jest aktywny i 0 gdy krok y jest nieaktywny;
- Sy.f - zmienna bitowa kontrolująca czas aktywności kroku y (może to być czas minimalny lub maksymalny). Jeśli czas aktywności kroku y przekroczy określoną wartość to zmienna Sy.f zmienia wartość z 0 na 1.

Literatura

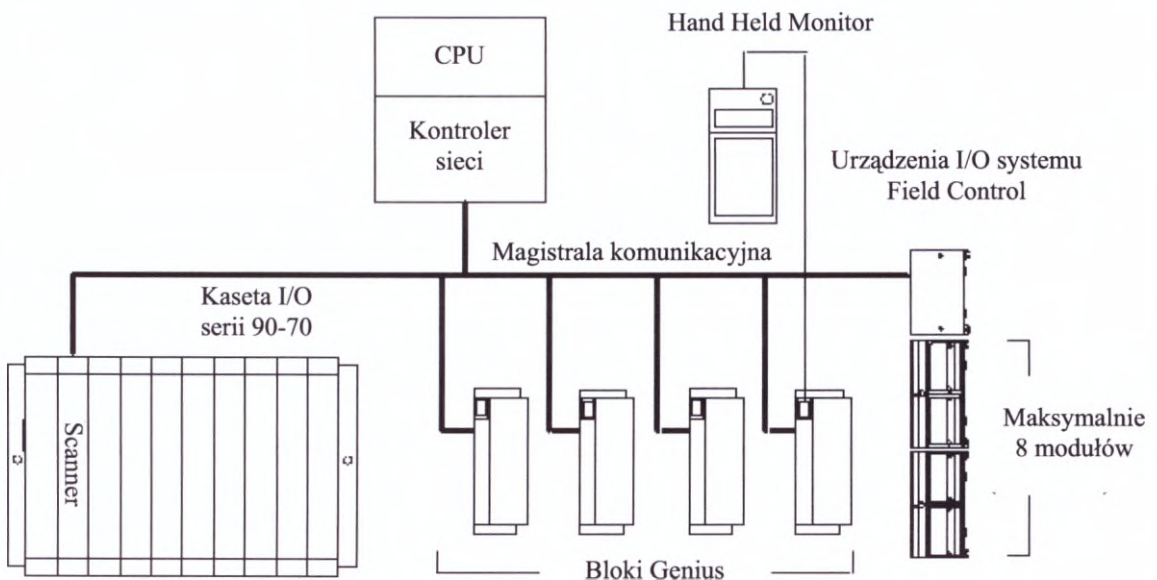
- [1] Legierski T., Kasprzyk J., Wyrwał J., Hajda J., Programowanie sterowników PLC, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 1998
- [2] Mandado E., Marcos J., Perez S. A., Programmable Logic Devices and Logic Controllers, Prentice Hall, 1996
- [3] Mikulczyński T., Samsonowicz Z., Automatyzacja dyskretnych procesów produkcyjnych. Metody modelowania procesów dyskretnych i programowania PLC, Wydawnictwa Naukowo-Techniczne, Warszawa, 1997
- [4] PN-EN 61131-3:1998 Sterowniki programowalne. Języki programowania
- [5] Series 90™. Sequential Function Chart. Programming Language. User's Manual, GFK-0854A, GE Fanuc Automation, October, 1994

6. Sieć przemysłowa GENIUS

Jako przykład prostej sieci przemysłowej zostanie przedstawiona sieć GENIUS firmy GE Fanuc Automation (rys. 6.1.). Jest to ekonomiczny i elastyczny system mający zastosowanie w układach sterowania, którym stawiane są wysokie wymagania w dziedzinie niezawodności i szybkości działania. System GENIUS pozwala na konfigurowanie wielu różnych systemów z rezerwacją CPU, magistrali i układów wejść/wyjść (certyfikat TÜV w klasie 4 i 5).

Podstawowe dane charakteryzujące sieć GENIUS:

- powstała w 1986 r.;
- maksymalna realna szybkość transmisji danych 153 kbaud;
- możliwość podłączenie do 32 urządzeń w jednej gałęzi sieci;
- długość magistrali w ramach jednej gałęzi do 2200 m (przy ograniczeniu szybkości transmisji do 38 kBaud);
- 600 m przy prędkości maksymalnej;
- możliwość rozszerzenia zasięgu do 15 km – przy zastosowaniu światłowodów;
- maksymalna liczba gałęzi wynosi:
 - 32 dla sterownika 90-70;
 - 8 dla sterownika 90-30.



Rys. 6.1. Przykładowa sieć GENIUS

Magistrala GENIUS pozwala na pracę w sieci:

- **Bloków oddalonych wejść/wyjść – bloków GENIUS (GENIUS I/O Blocks)** – dostępne są moduły:
 - wejść/wyjść dyskretnych;
 - moduły wejść/wyjść analogowych;
 - moduły specjalne:
 - ♦ licznik impulsów wysokiej częstotliwości;
 - ♦ moduły dla termopar i termometrów rezystancyjnych RTD;
 - ♦ moduł do monitorowania sieci prądu trójfazowego Power TRAC.
- **Sterowników serii 90-30 i 90-70** – karty komunikacyjne dla tych sterowników zapewniają asynchroniczną obsługę komunikacji pomiędzy sterownikiem PLC a urządzeniami na magistrali (komunikacja z jednostką centralną CPU w cyklu pracy sterownika oraz komunikacja z urządzeniami na magistrali w cyklu magistrali) oraz diagnostykę systemu (zapis komunikatów diagnostycznych w tablicach błędów sterownika).
- **Komputerów PC** wyposażonych w kartę PCIM (PC Interface Module) - jest to interfejs magistrali umożliwiający komunikację pomiędzy komputerem PC a elementami systemu GENIUS. Oprogramowanie interfejsu umożliwia konfigurację interfejsu oraz sterowanie przepływem danych do i z urządzeń na magistrali.
- **Modułów wejść/wyjść w systemie FieldControl** – pracę modułu w sieci GENIUS zapewnia interfejs magistrali (Bus Interface Unit), który obsługuje wejścia i wyjścia podłączonych modułów oraz prowadzi komunikację z urządzeniami na magistrali.
- **90-70 Remote Control I/O Scanner** – umożliwia podłączenie do sieci GENIUS kasety serii 90-70 z modułami I/O.
- **Urządzeń serii VersaMax** – pracują one jako system rozproszonych wejść/wyjść – zamiast jednostki centralnej sterownika VersaMax należy zamontować interfejs komunikacyjny sieci GENIUS.
- **Hand Held Monitora** – stosowany przy instalacji, konfigurowaniu, testowaniu i serwisowaniu systemu, samej magistrali GENIUS jak również bloków GENIUS i Field Control (monitorowanie stanu wejść/wyjść, wymuszanie stanu wejść/wyjść).

6.1. Charakterystyka elementów sieci

W podrozdziale tym omówione zostaną kolejno elementy mogące współpracować z siecią GENIUS.

6.1.1. Bloki GENIUS

Bloki GENIUS (rys. 6.2.) są inteligentnymi, samodzielnymi, konfigurowalnymi modułami wejść/wyjść. Każdy blok posiada pełne możliwości komunikacyjne i umożliwia podłączenie szerokiej gamy urządzeń wejściowych i/lub wyjściowych. Dostępne są bloki dyskretne, analogowe i specjalnego przeznaczenia – wszystkie typy można łączyć ze sobą w ramach tej samej sieci. Każdy blok GENIUS składa się z podstawy i części elektronicznej. Wszystkie połączenia związane z siecią, urządzeniami zewnętrznymi, a także z ręcznym programatorem (HHM) są doprowadzone do podstawy. Konfiguracja bloku zapisana jest w pamięci EEPROM znajdującej się także w podstawie. W części elektronicznej znajduje się

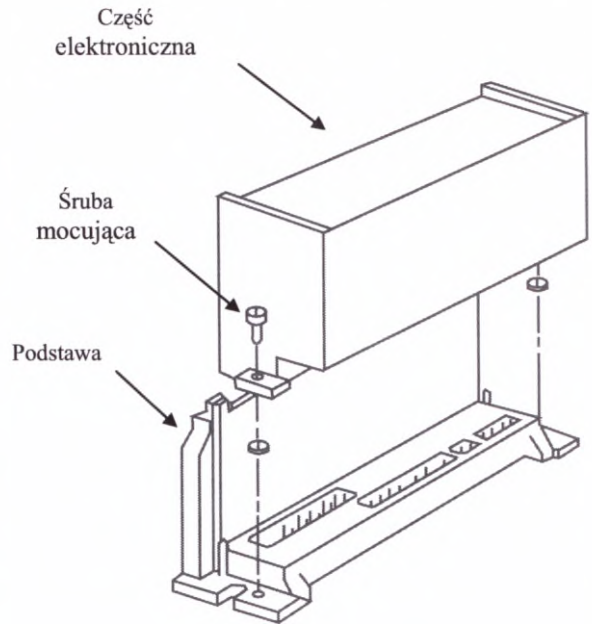
układ mikroprocesorowy bloku oraz punkty I/O wyposażone w diody diagnostyczne. W odróżnieniu od konwencjonalnych modułów wejść/wyjść montowanych w kasetach podstawowych sterownika, bloki GENIUS mogą być instalowane w dowolnym miejscu w odległości do około 2200 m od sterownika lub komputera. Możliwe jest także połączenie urządzeń z wykorzystaniem modemów i okablowania światłowodowego zwiększającego zasięg sieci do ok. 15 km. Mogą być montowane w dowolnym, wolnym od zanieczyszczeń miejscu zapewniającym odpowiedni obieg powietrza.

Do najważniejszych bloków GENIUS należą:

- Bloki dyskretne posiadające:
 - 32 wejść/wyjść;
 - możliwość konfigurowania pojedynczego punktu I/O jako wejścia lub wyjścia;
 - max prąd 0,5 A/pkt., max prąd 16 A/blok;
 - wejściowy czas filtracji 10 do 100 ms;
 - możliwość pracy w redundancji;
 - szerokie możliwości diagnostyczne;
 - możliwość pracy w trybie Output Default State / Hold Last State.

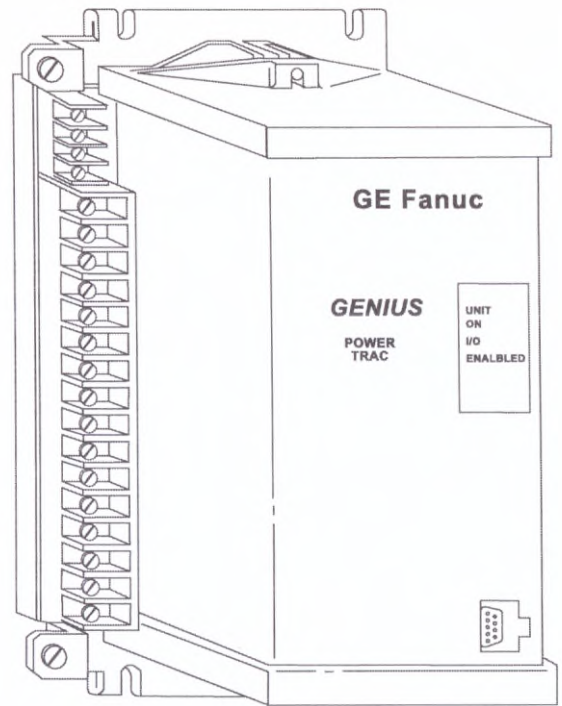
Tryb Output Default State oznacza, że w przypadku utraty komunikacji z kontrolerem sieci wyjście jest ustawiane w stan zdefiniowany wcześniej przez użytkownika. W trybie Hold Last State stan wyjścia zachowuje w takim przypadku ostatnią wartość.

- Bloki analogowe posiadające:
 - 6 punktów wejściowych, 6 wyjściowych lub 4 wejścia i 2 wyjścia;
 - zakres wejść/wyjść: 0 - 10 V, -10 - 10 V, 0 - 5 V, -5 - 5 V, 4 - 20 mA;
 - wejściowy czas filtracji do 1024 ms;
 - możliwość pracy w redundancji;
 - szerokie możliwości diagnostyczne;
 - skalowanie wejść/wyjść;
 - rozdzielczość 12 bitów + znak.



Rys. 6.2. Budowa bloku GENIUS

- Bloki specjalne
 - High-speed Counter – szybki licznik o częstotliwości zliczania impulsów do 200 kHz, ma możliwości zbliżone do tradycyjnego HSC;
 - PowerTRAC (rys. 6.3.) – używany w wielu aplikacjach przemysłowych i systemach monitoringu mocy: monitoruje wejścia prądowe i napięciowe, oblicza napięcie skuteczne, prąd, moc czynną, moc bierną, współczynnik mocy, zlicza moc [kWh] – może współpracować z siecią jedno- i trójfazową;
 - Bloki temperaturowe – do pomiarów przy użyciu termopar i czujników rezystancyjnych posiadają: możliwość linearyzacji charakterystyk, alarmowania, dokładność 0,1 °C, wejściowy czas filtracji 400 – 1600 ms.



Rys. 6.3. Blok PowerTRAC

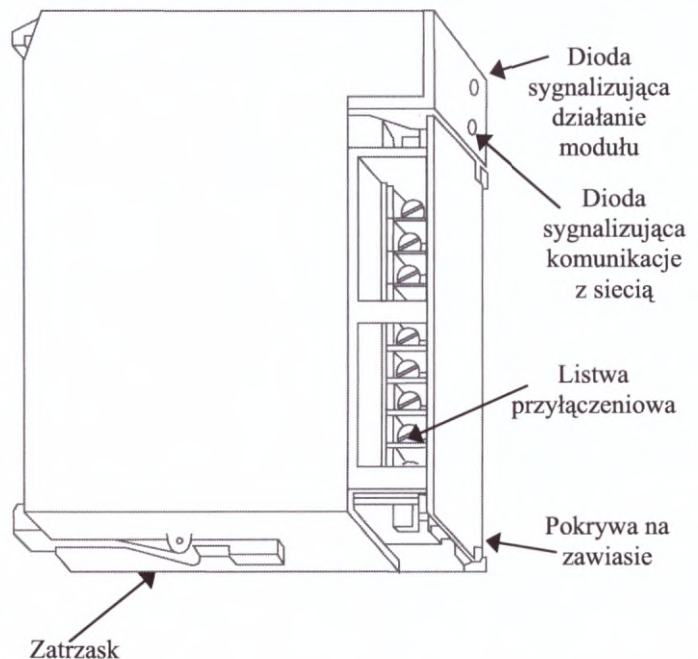
6.1.2. Kontrolery sieci

Kontrolery sieci (moduły komunikacyjne) mają za zadanie zapewnić komunikację CPU lub komputera z siecią GENIUS. W przypadku PLC kontroler automatycznie przesyła dane wejść/wyjść pomiędzy CPU i siecią. Nie jest przy tym wymagany żaden specjalny program do obsługi I/O. Dane diagnostyczne odbierane przez kontroler z sieci są automatycznie odczytywane przez CPU i zapisywane w tablicy błędów sterownika. Informacje te następnie można odczytać przy użyciu oprogramowania narzędziowego do sterowników. W przypadku komputera do odczytywania stanów wejść/wyjść i danych diagnostycznych wymagane jest odpowiednie oprogramowanie i karta komunikacyjna. Komputer nie posiada bowiem wbudowanej logiki do przetwarzania tego typu funkcji automatycznie, tak jak ma to sterownik PLC.

Kontrolery dla systemu 90-30

- GENIUS PLUS (GCM+) – rys. 6.4.

Umożliwia komunikację pomiędzy sterownikami GE-Fanuc serii 90-30 oraz innych serii za pośrednictwem magistrali



Rys. 6.4. Moduł GENIUS PLUS

komunikacyjnej GENIUS przy wykorzystaniu danych globalnych. Oznacza to że może on wymieniać dane tylko z innymi sterownikami i komputerami PC wyposażonymi w odpowiednie karty komunikacyjne. Moduł ten nie może pracować jako kontroler oddalonych wejść/wyjść takich jak moduły GENIUS, Field Control czy też VersaMax, a jedynie pozwala na nasłuch danych wysyłanych przez te urządzenia. W systemie 90-30 można zainstalować jeden lub kilka modułów GENIUS PLUS, każdy z własną magistralą GENIUS. Każdy moduł może wysyłać do sieci do 128 bajtów informacji oraz odbierać wszystkie informacje z pozostałych modułów pracujących w sieci. Może być zainstalowany w dowolnego rodzaju CPU w kasecie podstawowej, rozszerzającej lub oddalonej.

- GENIUS BUS CONTROLLER (GBC)

Posiada znacznie rozszerzony zakres funkcji w porównaniu z modułem GCM+. Realizuje asynchroniczną obsługę komunikacji w postaci danych globalnych i datagramów pomiędzy jednostką centralną sterownika a urządzeniami podłączonymi do magistrali komunikacyjnej GENIUS - maksymalnie 31 urządzeń na jednej magistrali. W systemie 90-30 można zainstalować do ośmiu modułów GENIUS BUS CONTROLLER, każdy z własną magistralą GENIUS, przy czym moduł taki nie może pracować w tej samej kasecie, w której jest już zainstalowany moduł komunikacyjny GENIUS PLUS. Moduł GBC umożliwia pracę w systemie redundantnym (ze zdublowaną magistralą), zapewnia też szerokie możliwości diagnostyczne.

Kontrolery dla systemu 90-70

Moduł komunikacyjny sieci GENIUS dla serii sterowników 90-70 ma podobne możliwości do kontrolera sieci GBC dla serii 90-30.

Kontrolery dla komputera PC

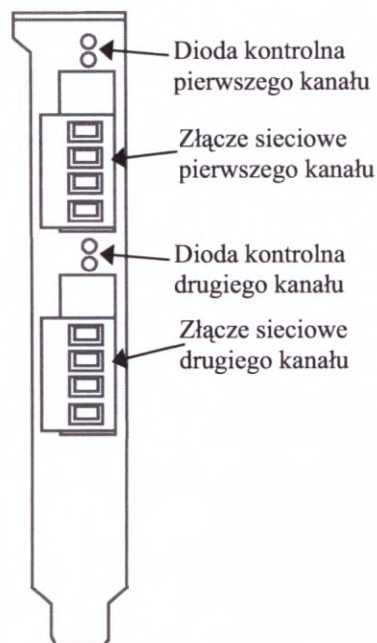
- KARTA PCIM - INTERFEJS MAGISTRALI GENIUS

Jest dostępna w dwóch wersjach:

- jednokanałowej;
- dwukanałowej (rys. 6.5.).

Umożliwia komunikację pomiędzy komputerem PC a rozproszonymi systemami wejść/wyjść GENIUS i FIELD CONTROL oraz sterownikami serii 90-30 i 90-70. Komputer podłączony przez kartę PCIM do systemu GENIUS staje się jednym z elementów systemu i może korzystać z magistrali komunikacyjnej. Karta PCIM współpracuje ze sterownikiem 90-30 poprzez moduły komunikacyjne GCM+ i GBC. W skład zestawu wchodzi również oprogramowanie karty PCIM, umożliwiające:

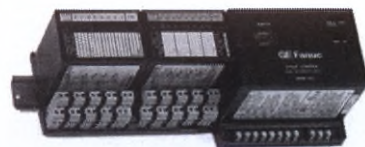
- konfigurowanie karty;
- diagnostykę systemu.



Rys. 6.5. Karta PCIM

6.1.3. System FIELD CONTROL

System Field Control (rys. 6.6. i 6.7.) to modułowy układ wejść/wyjść z możliwością pracy w sieci GENIUS (lub innych magistral - np. Profibus DP). Moduły wpinane są do magistrali komunikacyjnej za pośrednictwem interfejsu komunikacyjnego (Bus Interface Unit). W systemie Field Control istnieje możliwość zastosowania lokalnego procesora (Field Processor) umożliwiającego lokalną realizację procedur sterujących dzięki czemu można zdecentralizować system sterowania.



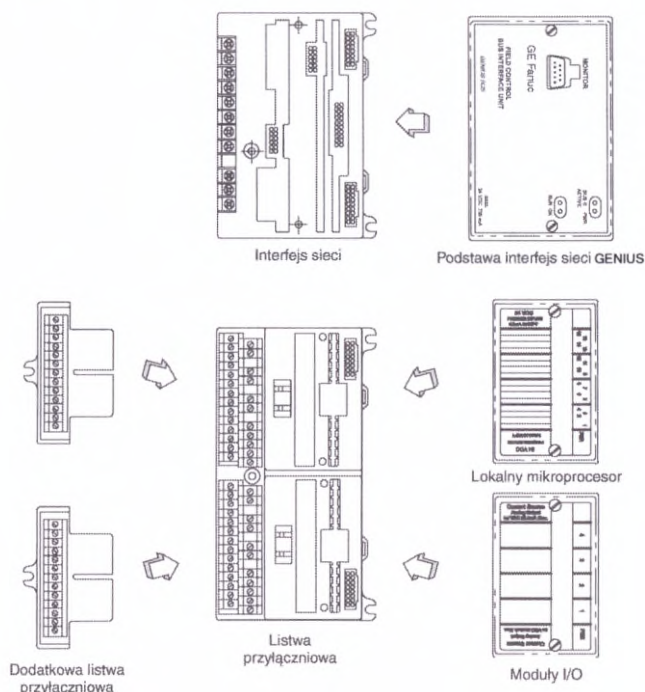
Rys. 6.6. System Field Control zamontowany na szynie montażowej DIN

Zadania interfejsu magistrali to:

- obsługa wejść i wyjść podłączonych modułów;
- prowadzenie komunikacji z urządzeniami na magistrali;
- umożliwienie konfigurowania i przechowywania parametrów konfiguracyjnych modułów - maksymalnie 8 modułów w grupach po 2 na podstawie;
- zapewnienie diagnostyki modułów.

Dostępne moduły:

- wejścia/wyjścia dyskretne;
- wejścia/wyjścia analogowe;
- moduły dla termometrów oporowych (RTD) i termopar;
- moduły dla zadajników potencjometrycznych;
- moduły dla mostków tensometrycznych.



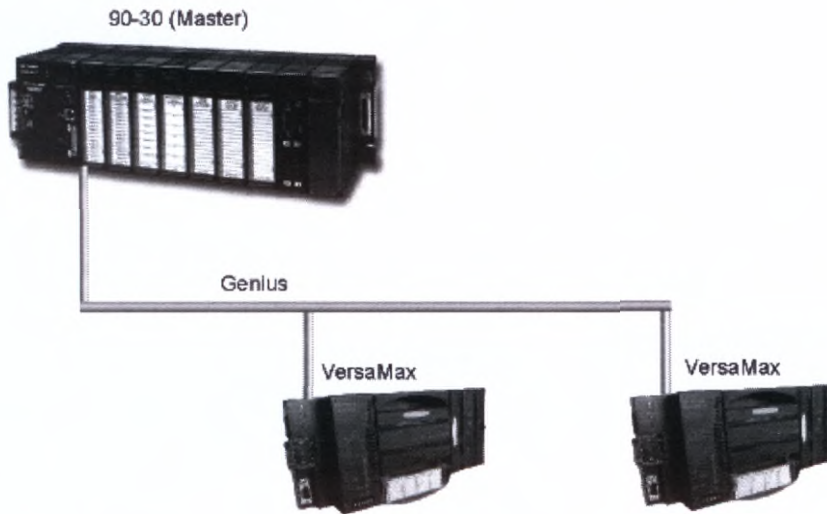
Rys. 6.7. Elementy składowe systemu Field Control

6.1.4. 90-70 Remote Control I/O Scanner

Umożliwia podłączenie do sieci GENIUS kasyety montażowej sterownika serii 90-70 z modułami I/O. Podstawowe jego cechy to:

- zawiera Series 90-70 Remote Control I/O Scanner i moduły I/O serii 90-70;
- traktowany jako jedno urządzenie w sieci;
- obsługuje do 1024 wejść/wyjść dyskretnych i do 64 wejść/wyjść analogowych;
- możliwość mieszania kanałów analogowych z dyskretnymi.

6.1.5. Interfejs komunikacyjny serii VersaMax



Rys. 6.8. Zastosowanie urządzeń serii VersaMax w sieci GENIUS

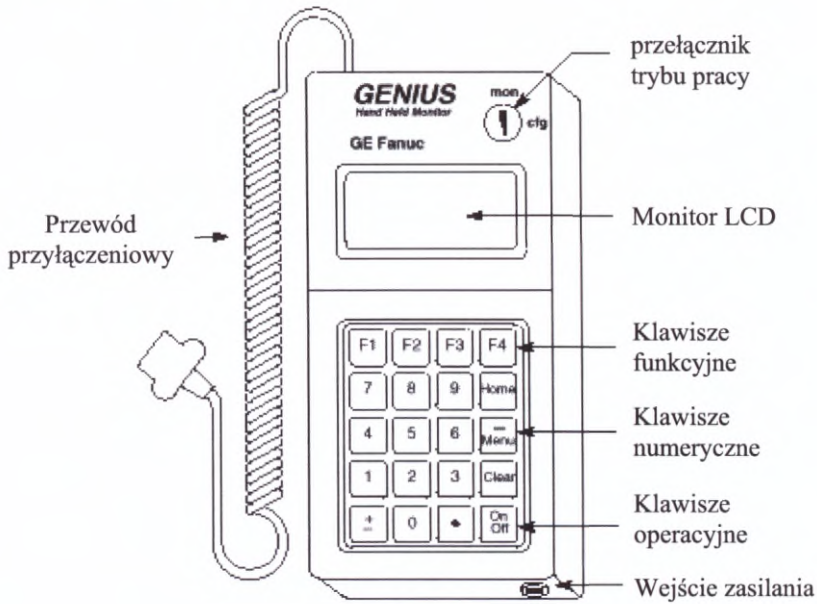
Zastosowanie interfejsu komunikacyjnego sieci GENIUS zamiast jednostki centralnej CPU sterownika serii VersaMax umożliwia użycie urządzeń VersaMax jako systemu rozproszonych wejść/wyjść. Przykład takiego rozwiązania przedstawiono na rys. 6.8. Więcej informacji o możliwościach urządzeń serii VersaMax, w tym dotyczących ich pracy w innych systemach rozproszonych, znajduje się w Dodatku D1.3.

6.1.6. Hand Held Monitor HHM

HHM (rys. 6.9.) przenośne urządzenia, które służy do:

- konfiguracji bloków Geniuss i modułów Field Control (określanie nr bloku, prędkości transmisji, konfiguracji wejść/wyjść);
- monitorowania sieci:
 - ◆ informacje ogólne;
 - ◆ wartości wejść/wyjść poszczególnych urządzeń w sieci;
 - ◆ czas cyklu magistrali;
 - ◆ błędy;
 - ◆ informacje dodatkowe z HSC i PowerTRAC'a;

- wymuszania stanów wyjściowych;
- usuwania błędów;
- testowania i diagnostyki.



Rys. 6.9. Hand Held Monitor

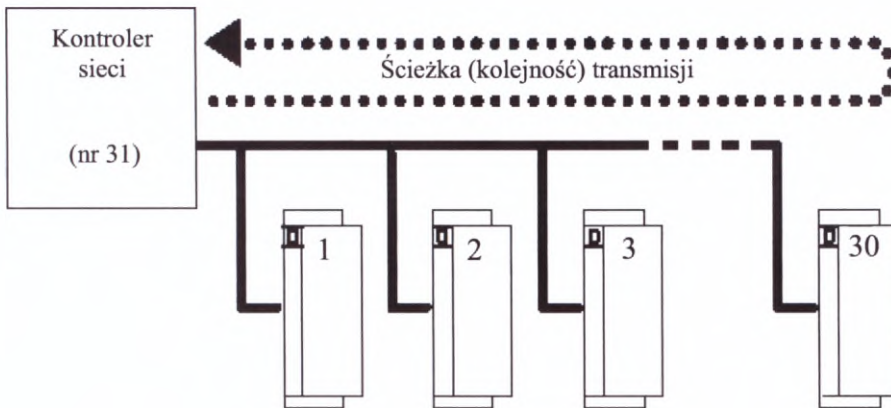
6.2. Komunikacja w sieci GENIUS

Podstawowe parametry charakteryzujące komunikację w sieci GENIUS zestawiono poniżej:

- Szybkość [kB/s]
 - 153,6 ST - standardowo w nowych blokach;
 - 153,6 EXT – zalecana – lepiej tłumi zakłócenia;
 - 76,8;
 - 38,4;
- Odległość
 - 2200 m przy 38,4 kB/s;
 - 1370 m przy 76,8 kB/s;
 - 1070 m przy 153,6 kB/s EXT;
 - 600 m przy 153,6 kB/s ST;
 - 15240 m przy użyciu światłowodów i modemów optycznych.
- Maksymalna liczba urządzeń (włączając kontrolery sieci i HHM)
 - 32 przy 153,6 i 76,8 kB/s;
 - 16 przy 38,4 kB/s.

- Protokół sieci GENIUS

GENIUS jest siecią typu token-passing dzięki czemu informacje są przesyłane w stałym cyklu - Bus Scan Time - (rys. 6.10.) zależnym od liczby i rodzaju urządzeń na magistrali, szybkości transmisji i ilości danych oraz rodzaju systemu (zwykły lub redundancyjny). Najczęściej czas trwania jednego cyklu wynosi 20 - 200 ms w zależności od wielkości całego układu.

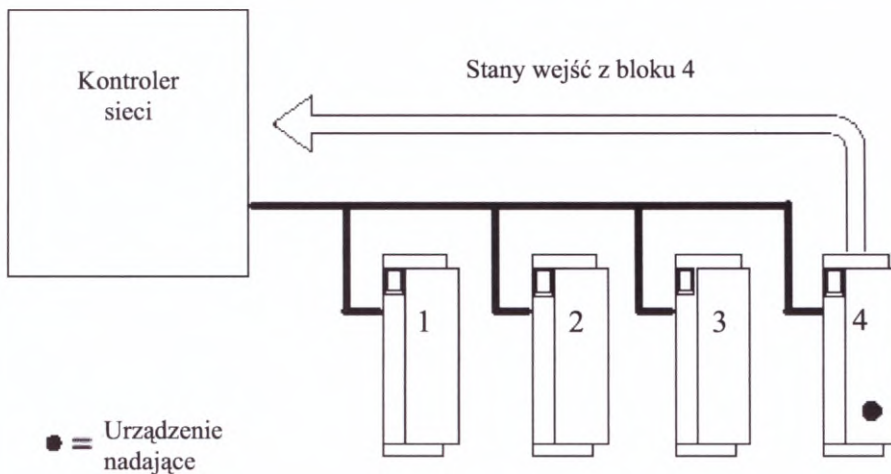


Rys. 6.10. Cykl magistrali GENIUS

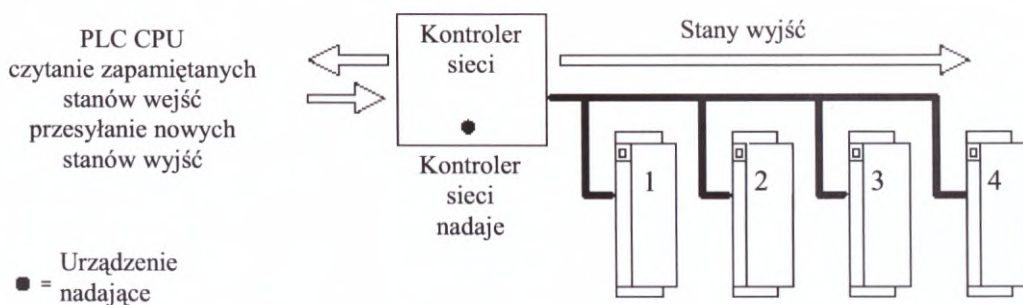
- Rodzaje przesyłanych danych

- stany I/O

- ♦ stany wejść są wysyłane w każdym skanie do wszystkich PLC w sieci (rys. 6.11.);
- ♦ stany wyjść są wysyłane w każdym skanie (cyklu) selektywnie przez PLC (do konkretnego urządzenia) korzystając z właściwości Outputs Enable/Disable;



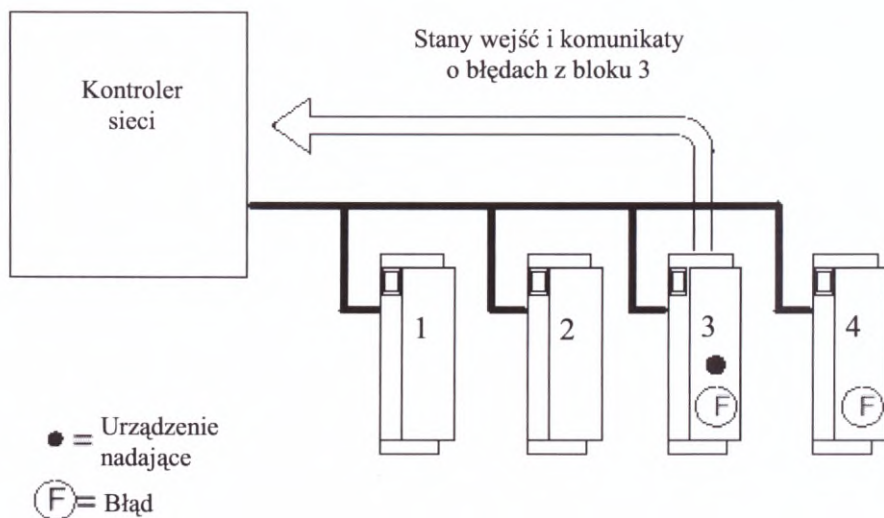
Rys. 6.11. Przesyłanie sygnałów wejściowych w sieci GENIUS



Rys. 6.12. Przesyłanie sygnałów wyjściowych w sieci GENIUS

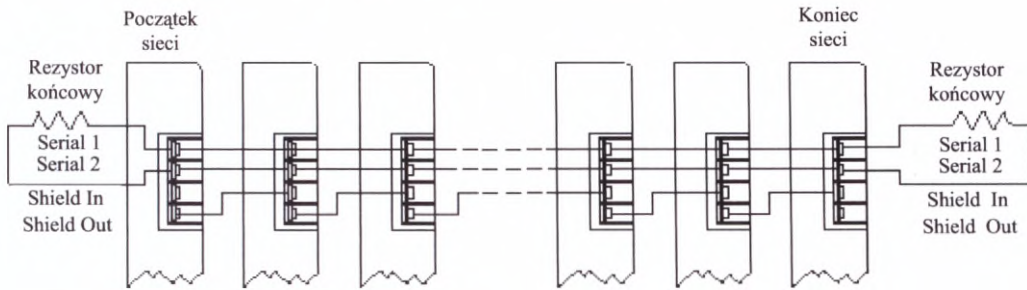
- dane globalne – to dane wysyłane automatycznie w każdym skanie sieci przez każde urządzenie pracujące w sieci (90-30 GBC może wysyłać i odbierać do 128 bajtów danych globalnych w każdym skanie), nie wymagają programowania – są określane na etapie konfiguracji GBC.
- datagramy
 - ◆ przesyłane do zaadresowanego urządzenia na magistrali;
 - ◆ możliwość potwierdzania odbioru i retransmisji;
 - ◆ wymagają osobnej instrukcji w programie sterującym;
 - ◆ w każdym cyklu magistrali może zostać wysłany jeden datagram;
 - ◆ datagramy powyżej 128 bajtów przesyłane są w wielu cyklach;
 - ◆ są wykorzystywane do obsługi wyjść systemów GENIUS i Field Control;
 - ◆ umożliwiają zmianę konfiguracji układów wejść/wyjść.

Rodzaj datagramów stanowią dane diagnostyczne przesyłane przez blok w przypadku wystąpienia błędu – w jednym skanie może być przesłana tylko 1 dana diagnostyczna (rys. 6.13.).



Rys. 6.13. Przesyłanie informacji o błędzie w sieci GENIUS

6.3. Zasady łączenia urządzeń w sieci GENIUS



Rys. 6.14. Zasady łączenia urządzeń w sieci GENIUS

Podstawowe wskazówki dotyczące łączenia urządzeń w sieci GENIUS (rys. 6.14.) wymieniono poniżej:

- dowolna kolejność – najkorzystniej zgodnie z nadanymi identyfikatorami;
- połączenia należy dokonywać odpowiednim kablem – tzw. skrętką lub specjalnym kablem prefabrykowanym;
- punkt Serial 1 łączyć z punktami Serial 1 bloku poprzedniego i następnego (jeśli taki występuje) – transmisja danych;
- punkt Serial 2 łączyć z punktami Serial 2 bloku poprzedniego i następnego (jeśli taki występuje) – transmisja danych;
- punkt Shield In każdego urządzenia łączyć z punktem Shield Out poprzedniego;
- punkt Shield In pierwszego i Shield Out ostatniego urządzenia pozostawić wolne;
- sieć zakończyć na obu końcach odpowiednim rezystorem lub prefabrykowaną końcówką zawierającą rezystor;
- nie wolno tworzyć rozgałęzień typu „T” lub „gwiazda”.

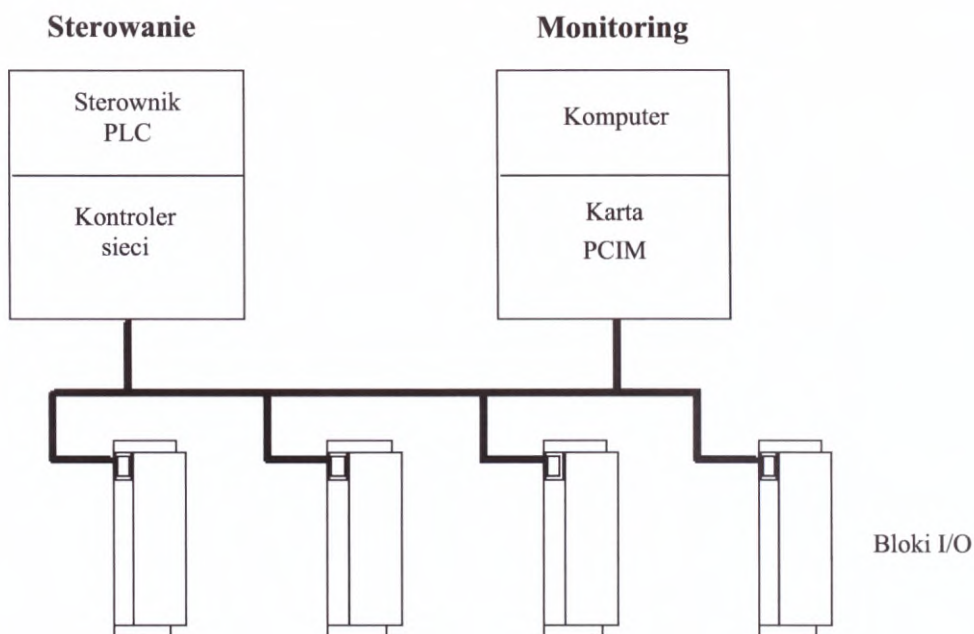
6.4. Redundancja i monitoring

Bardzo istotną zaletą sieci GENIUS są jej możliwości związane z monitoringiem oraz z redundancją poszczególnych elementów sieci. W przypadku sieci opartej o sterowniki serii 90-30 możliwa jest redundancja: magistrali (kable i kontrolerów) oraz CPU i kontrolerów typu Hot Standby lub Duplex. Dla sterowników 90-70 istnieje możliwość zastosowania redundancji z trzema kontrolerami sieci (Triple Modular Redundancy) w systemach GMR.

6.4.1. Monitoring danych

Do monitorowania sieci (rys. 6.15.) można użyć dodatkowego sterownika (komputera), który może otrzymywać informacje z sieci dotyczące:

- danych;
- błędów;
- alarmów.



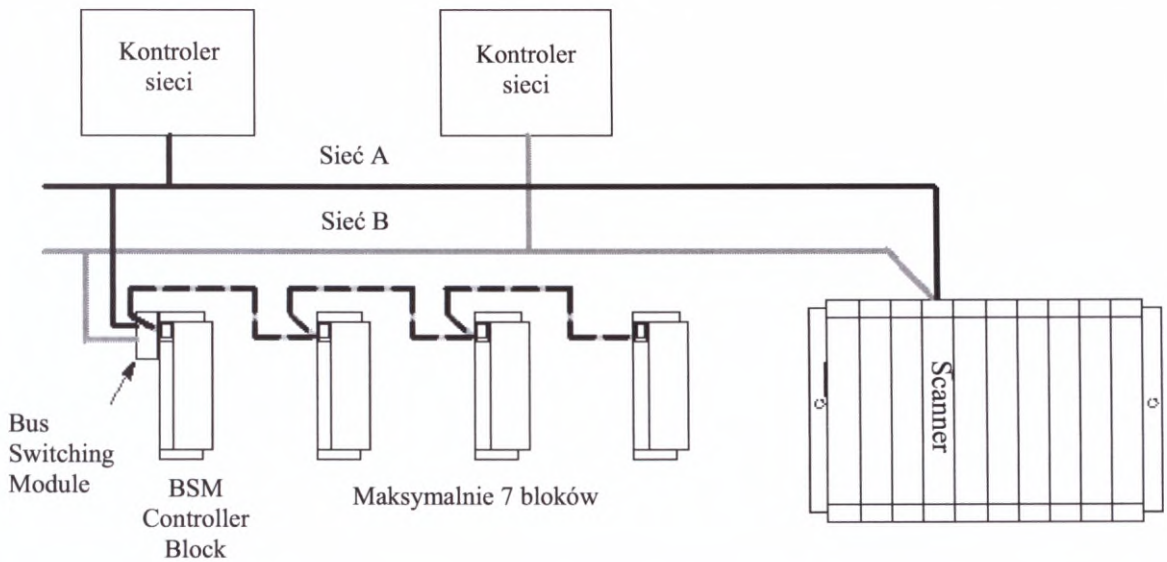
Rys. 6.15. Monitoring danych w sieci GENIUS

Kontrolery w urządzeniach monitorujących muszą mieć ustawiony parametr kontroli nad wyjściami na DISABLE.

6.4.2. Redundancja sieci i kontrolerów

W celu zabezpieczenia sieci przed zerwaniem łączności na skutek uszkodzenia kabla lub kontrolera (np. zanikiem napięcia zasilającego) stosuje się rozwiązanie przedstawione na rys. 6.16. W tym przypadku:

- używa się dwóch kabli sieciowych;
- każda magistrala ma swój własny kontroler (w przypadku jednego sterownika 90-30 mogą być dwa kontrolery w tej samej kasecie);
- występuje dodatkowy moduł Bus Switching Module (BSM) przyłączony do bloku GENIUS (nie każdego typu). Blok ten jest wówczas skonfigurowany jako BSM Controller Block;
- BSM Controller Block monitoruje magistrale i po stwierdzeniu braku łączności z siecią Aysterowuje BSM do przełączenia na sieć B.



Rys. 6.16. Redundancja magistrali i kontrolerów w sieci GENIUS

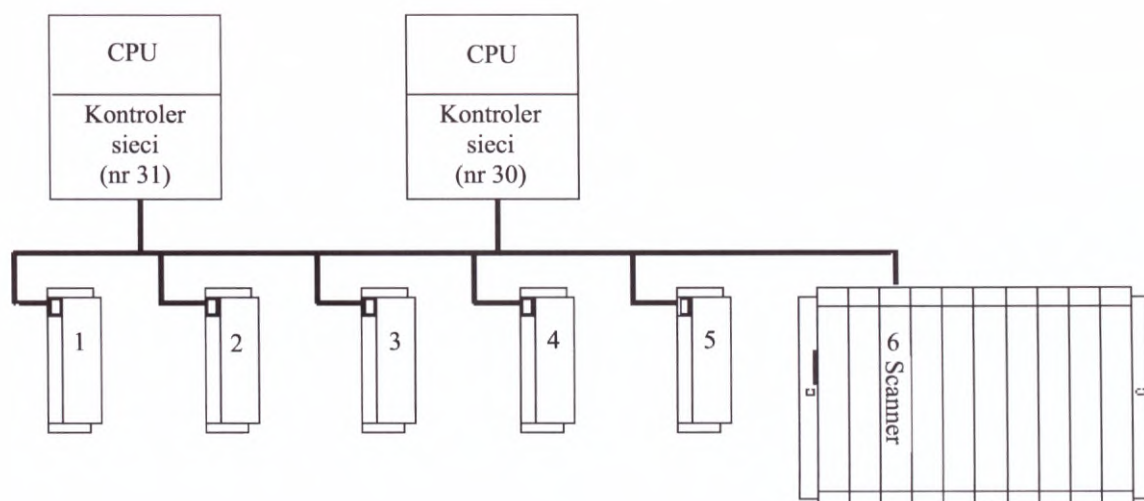
6.4.3. Redundancja sterowników i kontrolerów

Rozwiązanie zabezpieczające przed awarią kontrolera sieci lub CPU przedstawiono na rys. 6.17. Możliwa jest redundancja typu Hot Standby (zarówno dla bloków dyskretnych jak i analogowych) oraz Duplex (tylko dla bloków dyskretnych lub Remote I/O Scanners z samymi modułami dyskretnymi).

Hot Standby Redundancy

W przypadku redundancji typu Hot Standby:

- GBC 31 kontroluje wyjścia (musi mieć nr 31);
- jeśli przez 3 scany nie ma łączności z GBC 31 to wyjścia automatycznie przechodzą pod kontrolę GBC 30 (musi mieć nr 30);
- jeśli brak łączności z GBC 31 i GBC 30 to wyjścia przechodzą w Output Default lub Hold Last State (konfigurowalne);
- GBC 31 ma zawsze priorytet.



Rys. 6.17. Redundancja sterowników (procesora i kontrolera) w sieci GENIUS

Duplex Redundancy

W przypadku redundancji Duplex:

- wyjścia są kontrolowane przez oba GBC i gdy ich sygnały są zgodne to wyjścia są takysterowane, jeśli nie to przechodzą w Duplex Default State. Przykład pokazano poniżej – tab. 6.1.

Tab. 6.1. Przykład Duplex Redundancy

Wysterowany stan z urządzenia nr 31	Wysterowany stan z urządzenia nr 30	Duplex Default State	Aktualny stan wyjścia
On	On	Nie istotny	On
Off	On	Off	Off
Off	Off	Nie istotny	Off
On	Off	On	On

Literatura

- [1] GENIUS. I/O System and Communications. User's Manual, GEK-90486F-1, GE Fanuc Automation, November, 1994
- [2] GENIUS™ I/O. Discrete and Analog Blocks. User's Manual, GFK-90486D-2, GE Fanuc Automation, September, 1993
- [3] GENIUS™. Hand-held Monitor. User's Guide, GFK-0121E, GE Fanuc Automation, June, 1994
- [4] Series 90™-30. Enhanced GENIUS™ Communications Module. User's Manual, GFK-0695, GE Fanuc Automation, July, 1992
- [5] Series 90™-30. GENIUS® Bus Controller. User's Manual, GFK-1034B, GE Fanuc Automation, April, 1996

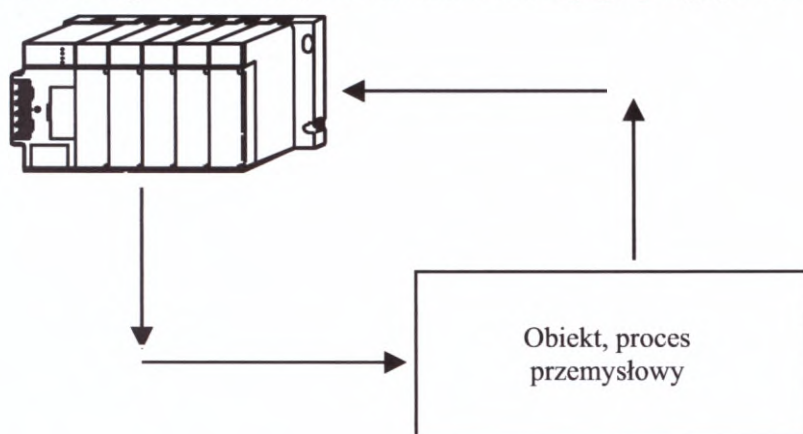
Podczas opracowywania niniejszego rozdziału korzystano także z dysków informacyjno-reklamowych CD ASTOR 97 i ASTOR CD 2001.

7. Monitoring – systemy SCADA

Z rozwojem sterowników PLC i komputerów klasy PC ściśle związany jest rozwój systemów nadzorowania i monitorowania procesów przemysłowych. Na początku lat dziewięćdziesiątych powstały pierwsze tego typu systemy oparte o komputery PC określane nazwą SCADA – Supervisory Control and Data Acquisition – Nadzór i Gromadzenie Danych.

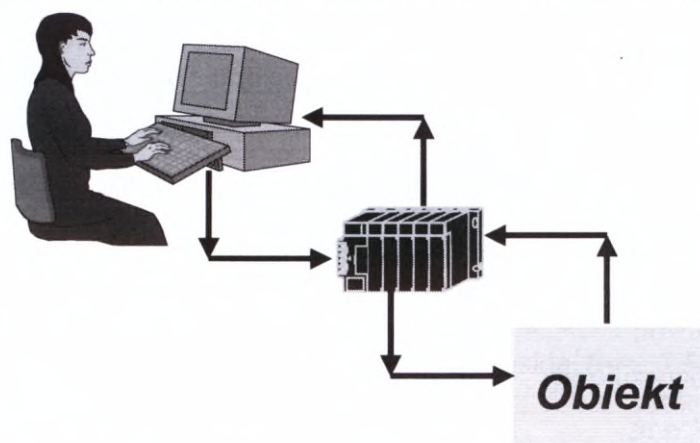
7.1. Wprowadzenie

Na rys. 7.1. przedstawiony jest sterownik PLC sterujący dowolnym obiektem lub procesem przemysłowym. Układ taki może działać poprawnie, lecz obsługa, bez podłączenia dodatkowego urządzenia np.: ręcznego programatora lub komputera z oprogramowaniem narzędziowym do obsługi sterownika, nie ma możliwości jego podglądu ani wpływania na



Rys. 7.1. Samodzielny PLC sterujący obiektem lub procesem przemysłowym

proces poprzez zmianę nastaw, wartości zadanych itp. Należy mocno podkreślić stwierdzenie – **po zaprogramowaniu, połączeniu z obiektem i uruchomieniu sterownik PLC może samodzielnie sterować pracą układu.** Jednak, w celu umożliwienia obsłudze prowadzenia kontroli procesu i ewentualnie jego korygowanie, stosuje się systemy monitoringu i wizualizacji. Obecnie często są to systemy SCADA. Zastosowanie ich wymaga połączenia sterownika PLC z komputerem PC, na którym zainstalowany został odpowiedni system – rys. 7.2., zapewniający możliwość współpracy ze sterownikami programowalnymi, regulatorami



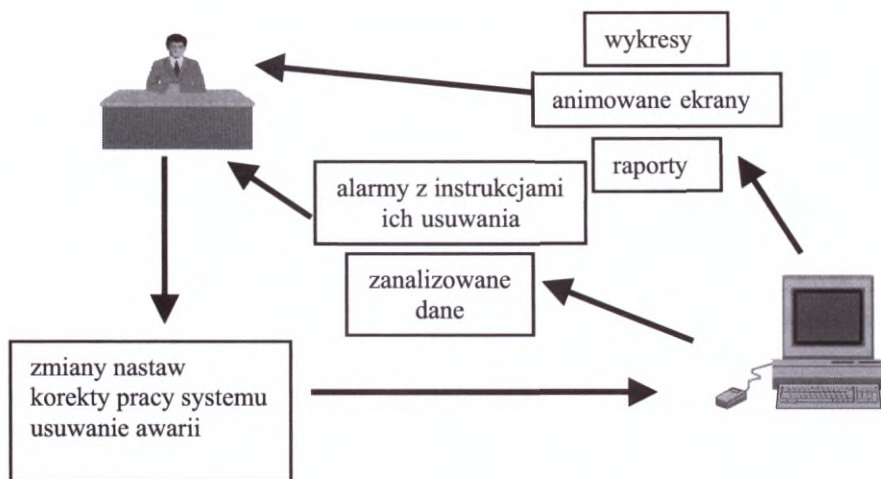
Rys. 7.2. Przykład wizualizacji i monitoringu za pomocą systemu SCADA

mikroprocesorowymi i koncentratorami danych różnych producentów.

Do najważniejszych zadań systemu monitoringu i wizualizacji należy zaliczyć:

- skupienie informacji w określonym miejscu (dyspozytorni);
- prezentacja informacji w sposób zgodny z potrzebami dyspozytora;
- możliwość oddziaływania na proces tzw. nadrzędne, zdalne sterowanie.

Prezentacja danych może odbywać się za pomocą różnych narzędzi – patrz rys. 7.3. Najczęściej są to animowane ekrany (ekrany synoptyczne), wykresy bieżące i historyczne oraz raporty.



Rys. 7.3. Podstawowe narzędzia wizualizacyjne systemu SCADA

Każdy, najprostszy nawet system monitoringu składa się z następujących elementów (warstw sprzętowych):

1. obwody obiektowe (warstwa na styku z obiektem technologicznym – tj. między innymi: czujniki, urządzenia pomiarowe, urządzenia wykonawcze);
2. sterowniki (stacje procesowe);
3. urządzenia i łącza transmisyjne (pomiędzy stacją procesową i stacją operatorską);
4. stacja operatorska.

Dwie pierwsze warstwy zostały omówione we wcześniejszych rozdziałach. Poniżej krótko scharakteryzowano urządzenia i łącza transmisyjne oraz stacje procesowe.

Ad. 3.

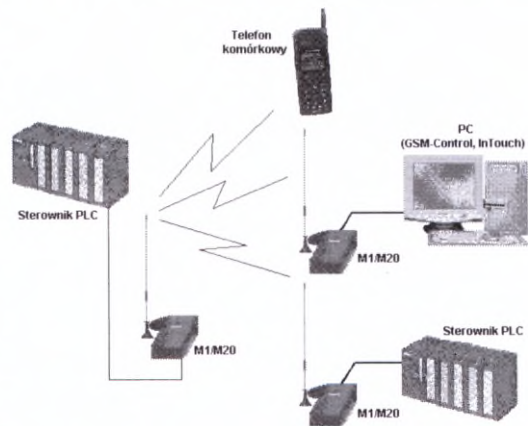
Rodzaj zastosowanych urządzeń i łączy transmisyjnych zależy od wielu czynników, w tym przede wszystkim od: odległości pomiędzy sterownikiem i stacją operatorską, terenu (miasto, tereny niezabudowane itp.), jego ukształtowania, istniejącej infrastruktury telekomunikacyjnej, wymaganej niezawodności, ilości i częstotliwości przesyłanych informacji.

Najczęściej wykorzystuje się:

- łącza kablowe;
- połączenia z wykorzystaniem telefonii stacjonarnej;

- ◆ łącza ogólnodostępne;
- ◆ łącza dzierżawione;
- połączenia z wykorzystaniem telefonii komórkowej;
- łącza światłowodowe;
- połączenia z wykorzystaniem fal radiowych – radiomodemy;
- łącza satelitarne.

Jak wynika z powyższego przeglądu mogą to być różnorodne rozwiązania: od bardzo prostych – w przypadku przesyłania sygnałów na niewielkie odległości wystarczy połączenie kablem (skrętka), do niezwykle złożonych np.: łącza satelitarne czy też połączenia wykorzystujące układy radiomodemowe w różnych konfiguracjach. Dokładne omówienie tego tematu wykracza poza ramy niniejszej publikacji. Nie mniej jednak warto wspomnieć, że połączenie z wykorzystaniem GSM (rys. 7.4.) pozwala w szczególności na przesyłanie krótkich wiadomości SMS pod konkretny numer telefonu. Rozwiązanie to może być zatem wykorzystywane w tych przypadkach, gdy nie jest możliwa lub konieczna stała obecność operatora na stanowisku wizualizacyjnym. Operator z kolei może za pomocą swojego telefonu komórkowego i SMS-u przysyłać dane do procesu np.: ustawić wartość zadaną, nastawy regulatora lub włączyć/wyłączyć urządzenie. Generalnie połączenia za pośrednictwem GSM stosuje się do sporadycznego, krótkotrwałego przesyłania niewielkiej ilości danych. Dla częstych połączeń bezprzewodowych stosuje się radiomodemy ze względu na znacznie niższe nakłady finansowe w dłuższej perspektywie czasu.



Rys. 7.4. Przykład systemu wykorzystującego komunikację za pomocą telefonii GSM

Ad. 4.

Stacja operatorska może występować pod postacią:

- tablic synoptycznych;
- wyświetlaczy;
- paneli operatorskich;
- komputerów.

Jak wspomniano wcześniej obecnie coraz częściej do monitoringu używa się komputerów klasy PC (samodzielnych lub połączonych w układy sieciowe) z zainstalowanym specjalistycznym oprogramowaniem SCADA. Początkowo pracowały one pod kontrolą różnych systemów operacyjnych tj. DOS, UNIX, OS/2. Obecnie standardem są wersje 32-bitowe dla Windows 98 i Windows NT. Stacje operatorskie SCADA powinny przewyższać wygodą obsługi różnorodne wyświetlacze i panele operatorskie (rys. 7.5.). W przypadku systemów SCADA panele takie nie muszą być stosowane, ewentualnie mogą stanowić rezerwę dla stacji operatorskich. Systemy SCADA uzupełniają, a w niektórych przypadkach zastępują konwencjonalne metody wizualizacji (głównie tablice synoptyczne).



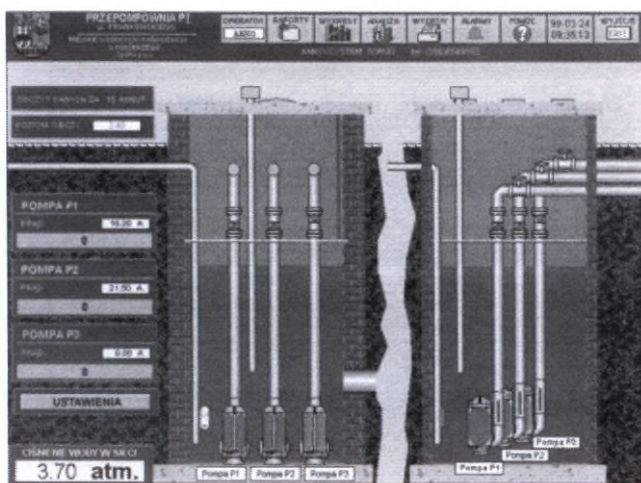
Rys. 7.5. Panel operatorski firmy GE Fanuc model 420T

7.2. Podstawowe cechy użytkowe systemów SCADA i dziedziny ich zastosowań

Typowe cechy użytkowe systemów SCADA to między innymi:

- sterowanie procesami i akwizycja danych w czasie rzeczywistym;
- współpraca z rozproszonym systemem sterowania;
- obszerna rozproszona baza danych;
- pełna obsługa wizualizacyjna stanu procesu (intuicyjny interfejs graficzny);
- natychmiastowa sygnalizacja pojawiających się stanów alarmowych;
- archiwizacja informacji o stanach alarmowych, działaniach operatora i innych zdarzeniach w systemie;
- tworzenie wykresów bieżących i historycznych (trendów);
- zaawansowane raportowanie;
- statystyczna kontrola procesów (SPC);
- możliwość współpracy ze sterownikami i innymi urządzeniami automatyki produkowanymi przez różnych producentów (najczęściej około 200 - 250 rozmaitych driverów I/O);
- współpraca z arkuszami kalkulacyjnymi (np. Excel) i bazami danych (np. Access, dBase, Oracle, Sybase, SQL);
- zabezpieczenie dostępu do systemu;
- zmienne receptury procesów;
- stała integralność bazy danych czasu rzeczywistego;
- możliwość stosowania układów redundancyjnych;
- dynamiczne konfigurowanie - możliwość tworzenia i rozbudowy aplikacji w już działającym systemie bez konieczności zatrzymywania procesu czy utraty istotnych danych;
- łatwa rozbudowa systemu o nowe elementy.

Gałęzie przemysłu, w których stosowane są systemy SCADA:



Rys. 7.6. Widok przepompowni ścieków P1 w Miejskich Wodociągach i Kanalizacji Spółka z o.o. w Kołobrzegu

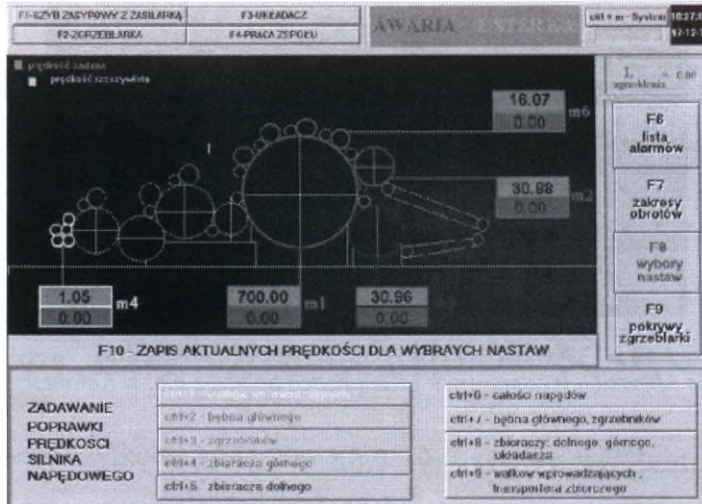
- uzdatnianie i dystrybucja wody – rys. 7.6.;
- oczyszczanie ścieków;
- wytwarzanie i zaopatrzenie w ciepło;
- zaopatrzenie w energię;
- kierowanie ruchem;
- kontrola budynków – rys. 7.7.;
- przemysł materiałowy;
- górnictwo;



Rys. 7.7. Wizualizacja w krakowskim Hotelu "Pod Różą" – poniżej zamieszczono krótki opis układu

Ekran synoptyczny pozwala na przeglądanie temperatur w całym hotelu, na poszczególnych piętrach i w każdym pokoju osobno. Funkcjonalność systemu daje każdemu klientowi hotelu możliwość zadania temperatury, jaką pragnie mieć w swoim pokoju. Stanowisko komputerowe znajduje się w recepcji i stąd panie recepcjonistki mogą zadawać temperaturę w każdym pomieszczeniu według kilku reguł.

- przemysł wytwórczy;
- przemysł elektrotechniczny;
- przemysł chemiczny;
- przemysł włókienniczy – rys. 7.8.;



Rys. 7.8. Ekran główny II – Zgrzeblarka - poniżej zamieszczono krótki opis układu

Projekt systemu kontrolno - pomiarowego dla pracy zespołu zgrzeblarkowego CU691.13/2500 wraz z układaczem 5W870 wykonany przez Zakładu Systemów Sterowania "OPTIMUS - SEKO" dla Bielskiej Fabryki Maszyn Włókienniczych "BEFAMA". System ma za zadanie monitorowanie pracy elementów zespołu zgrzeblarkowego, w tym zadawanie parametrów technologicznych niezbędnych dla poprawnej pracy, alarmowanie o zaistniałych sytuacjach awaryjnych, dostarczanie danych historycznych informujących o pracy napędów zespołu.

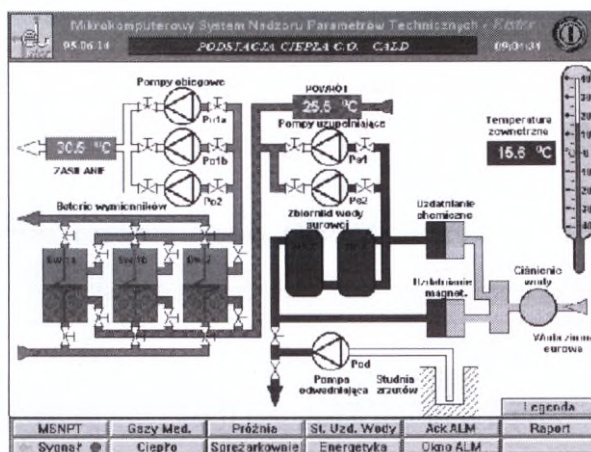
- przemysł samochodowy;
- przemysł papierniczy;
- przemysł spożywczy – rys. 7.9.;



Rys. 7.9. Główny ekran systemu wizualizacji w browarze "Heweliusz"

- kształcenie i szkolnictwo;
- systemy bezpieczeństwa (głównie obiektów rozległych);

- lecznictwo – rys. 7.10.



Rys. 7.10. Wizualizacja sieci gazów medycznych w Instytucie Pediatrii CM UJ

7.3. Omówienie głównych cech systemów SCADA

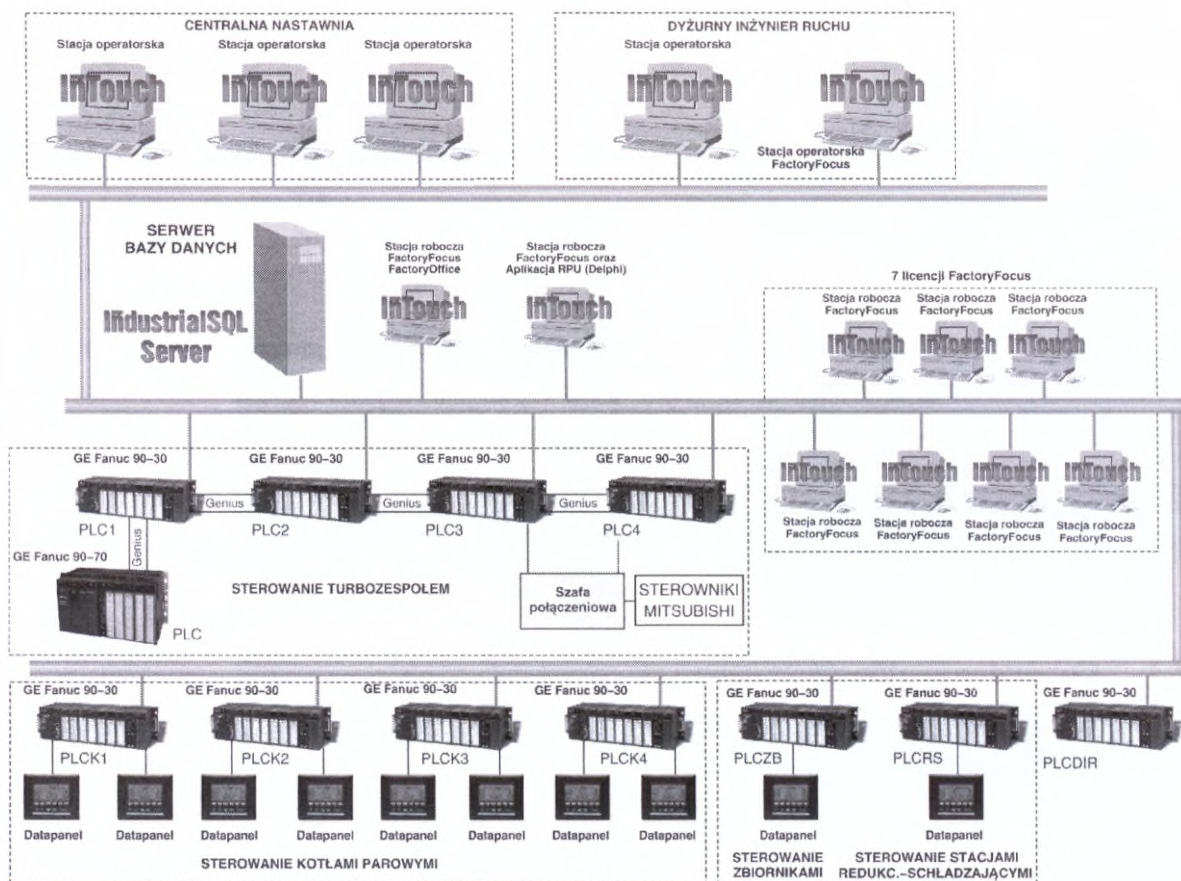
W niniejszym rozdziale, w oparciu o pozycję [1], omówiono niektóre z charakterystycznych cech systemów SCADA. Trzeba zaznaczyć, że poszczególne systemy różnią się zakresem i sposobem realizacji określonych funkcji. Podlegają przy tym stałemu rozwojowi, co ma wpływ na zmianę ich własności użytkowych. Dlatego zaprezentowane poniżej cechy są cechami orientacyjnymi, mogącymi w różnym stopniu charakteryzować konkretny system czy konkretną aplikację. Nie są one również uszeregowane według ważności. W niektórych rozwiązaniach pewne cechy są bowiem niezwykle istotne, podczas gdy w innych mogą być drugorzędne.

Sterowanie procesami i akwizycja danych w czasie rzeczywistym

Dane procesowe powinny być przesyłane pomiędzy sterownikiem (stacją procesową) a serwerem SCADA w możliwie krótkim czasie. W przeciętnych warunkach, w przypadku istotnych informacji, czas ten nie powinien przekraczać dziesiątych części sekundy. Równocześnie system powinien zapewnić możliwość zróżnicowania sposobu przetwarzania (odczytywania, uaktualniania) danych. Przetwarzanie to może być cykliczne lub sporadyczne (asynchroniczne). Cyklicznie są przetwarzane zazwyczaj analogowe i binarne sygnały pomiarowe, natomiast sporadycznie zmienne wprowadzane przez obsługę np. pomiary laboratoryjne, stany alarmowe itp. Przetwarzanie cykliczne wykonuje się często z różnymi deklarowanymi okresami (inne bowiem są wymagania na okres przetwarzania sygnałów z przetworników pomiarowych przepływu, a inne dla sygnałów pomiarowych temperatury).

Współpraca z rozproszonym systemem sterowania

Jednym z podstawowych warunków stawianym systemom SCADA jest możliwość stworzenia zdecentralizowanej, przestrzennie rozproszonej struktury systemu nadzorowania i wizualizacji procesu. Przykład takiego systemu przedstawiono na rys. 7.11.



Rys. 7.11. Przykład zdecentralizowanego układu sterowania z wykorzystaniem systemu SCADA działający w Elektrociepłowni Zielona Góra S.A.

Obszerna rozproszona baza danych

Dane procesowe mogą być gromadzone, archiwizowane na dowolnym serwerze SCADA. Ilość danych, które mogą być zapamiętane w systemie zależy zatem od liczby serwerów SCADA, pojemności używanych pamięci masowych i sposobu kompresji danych. W systemie sieciowym wszystkie dane z dowolnej stacji systemu powinny być dostępne dla wszystkich innych stacji.

Pełna obsługa wizualizacyjna stanu procesu (intuicyjny interfejs graficzny)

Jednym z najważniejszych zadań systemów SCADA jest wizualizacja procesu przy użyciu ekranów synoptycznych (animowanych). Dlatego też powinny zapewniać:

- wysoki poziom grafiki – najczęściej jest to grafika wektorowa (obiektowa) lub rastrowa (punktowa);
- duże możliwości animacyjne: zmiana koloru, migotanie, przesunięcia, skalowanie, obrót, napełnianie profili zamkniętych, pojawianie się i znikanie obiektów, wyświetlanie wartości liczbowych;
- możliwość importu bitmap;
- szeroką paletę dostępnych kolorów – nawet do 16,7 mln kolorów dla bitmap i grafiki wektorowej;

- wygodne narzędzia do rysowania obrazów i ich animacji – intuicyjną pracę zapewnia między innymi wykorzystanie narzędzi znanych z innych programów pracujących pod systemem Windows np.: „cut and paste” oraz „drag and drop”;
- możliwie obszerną bibliotekę gotowych obiektów takich jak: przełączniki, zbiorniki, zawory itp. – często jest ich kilkaset;
- możliwość tworzenia i zapamiętywania własnych obiektów – możliwość uzupełniania biblioteki;
- możliwość tworzenia różnych typów obrazów, przy czym jedne z nich mogą być otwierane z poziomu innych;
- łatwość „nawigacji” w systemie;
- możliwość wykorzystania technik multimedialnych – obrazu z kamery przemysłowej, przygotowanej wcześniej sekwencji filmowej, informacji przekazywanej głosem.

Natychmiastowa sygnalizacja pojawiających się stanów alarmowych

Ważnym zadaniem systemów SCADA jest sygnalizacja wykrytych alarmów. Generowane mogą być przy tym różne ich rodzaje: systemowe o wykrytych uszkodzeniach sprzętowych, błędach programowych, błędach transmisji oraz alarmy i ostrzeżenia technologiczne informujące o przekroczeniach dopuszczalnych granic, dopuszczalnych szybkości zmian zmiennych procesowych lub nieprawidłowych stanach zmiennych binarnych. Sygnalizowane są standardowo na obrazach synoptycznych oraz, w układzie chronologicznym, na obrazie alarmów. System SCADA powinien zapewniać filtrację alarmów. Dla każdego użytkownika powinna istnieć możliwość takiego ograniczenia strumienia sygnalizowanych alarmów, aby uzyskiwał on tylko te informacje alarmowe, które go bezpośrednio dotyczą. Chodzi o zabezpieczenie obsługi przed przeciążeniem spowodowanym nadmiernym nagromadzeniem informacji.

Archiwizacja informacji o stanach alarmowych, działaniach operatora i innych zdarzeniach w systemie

Systemy SCADA posiadają możliwość archiwizacji nie tylko danych procesowych, ale również zgłaszanych alarmów, innych ważnych zdarzeń w systemie a także, co jest niezwykle istotne, działań operatora. Pozwala to w szczególności na analizę przyczyn zaistnienia sytuacji krytycznych w układzie.

Tworzenie wykresów bieżących i historycznych (trendów)

Dane procesowe mogą być prezentowane w formie wykresów tworzonych na bieżąco, pokazujących aktualny przebieg danych lub wykresów historycznych, pokazujących przebieg danych w przeszłości. System SCADA powinien przy tym umożliwić:

- przedstawienie na wykresie, w dowolnej stacji systemu SCADA, dowolnej zmiennej z instalacji;
- skalowanie na bieżąco osi czasu w celu obserwacji przebiegów w żądanym okresie np.: godziny, zmiany – tj. 8 godzin, doby itd.;
- skalowanie osi wartości zmiennych oraz mechanizm skalowania automatycznego;
- przewijanie wykresu;
- wyświetlanie wartości zmiennych we wskazanych chwilach;
- przedstawianie wartości w jednostkach fizycznych i procentach zakresu pomiarowego;

- wyświetlanie wartości statystycznych (średnich, maksymalnych, minimalnych) we wskazanym zakresie;
- prezentacje wykresów typu $X=f(Y)$.

Zaawansowane raportowanie

Podstawowe wymaganie dotyczy łatwości definiowania raportów. Każdy proces ma swoją specyfikę i wymaga opracowania dla niego odpowiednich raportów. Łatwość ich opracowywania i uruchamiania ma zasadnicze znaczenie. Powinna istnieć możliwość inicjacji raportów w następujących trybach: na żądanie, cyklicznie, po wystąpieniu określonego zdarzenia.

Otwartość systemu

Na tę cechę systemów SCADA składają się między innymi:

- współpraca ze sterownikami i innymi urządzeniami automatyki produkowanymi przez różnych producentów (najczęściej około 200 - 250 rozmaitych driverów I/O);
- możliwość indywidualnego pisania driverów dla nietypowych, rzadziej stosowanych urządzeń automatyki;
- współpraca z arkuszami kalkulacyjnymi (np. Excell) i bazami danych (np. Access, dBase, Oracle, Sybase, SQL);
- możliwość wymiany danych z innymi systemami (między innymi przez standardowe mechanizmy – DDE, OLE2);
- wykorzystywanie standardowego systemu operacyjnego i oprogramowania sieciowego;
- zastosowanie standardowych sieci lokalnych (Ethernet);
- współpraca ze standardowymi sieciami polowymi (Profibus, CAN) i innymi rozpowszechnionymi rozwiązaniami (Modbus itp.).

Stała integralność bazy danych czasu rzeczywistego

Jest ona uzyskiwana dzięki wykorzystaniu odpowiednich przerwanych innych zadań wykonywanych w środowisku Windows. Windows jest bowiem wielozadaniowym systemem operacyjnym z wyłączeniem co jest warunkiem niezawodnej akwizycji i rejestracji danych pomiarowych. Dodatkowo błąd w oprogramowaniu jednego zadania powoduje przerwanie przez system operacyjny tylko tego zadania, podczas gdy inne pracują normalnie.

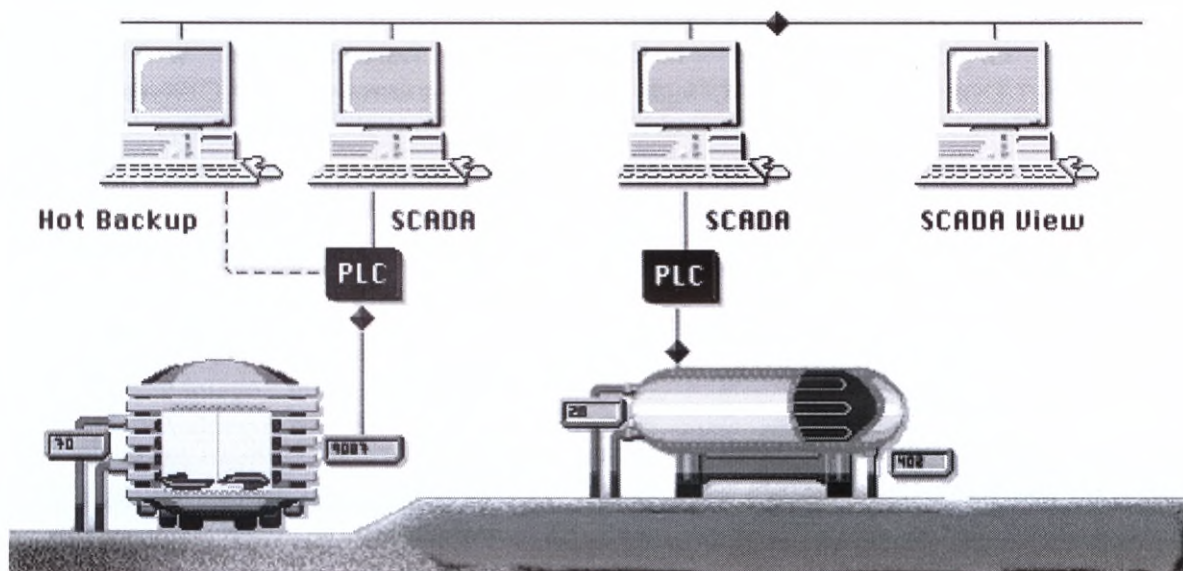
Zabezpieczenie dostępu

Systemy SCADA mają rozbudowane możliwości zabezpieczenia określonych funkcji i obiektów przed dostępem niepowołanych osób. Powinny zapewniać możliwość przydzielenia poszczególnym użytkownikom różnych uprawnień do: administracji systemem, sterowania procesem, modyfikacji parametrów, dostępu do danych. Każdy z użytkowników powinien na początku pracy zalogować się do systemu. Oczywiście jest, że inne uprawnienia muszą mieć: projektant systemu, inżynier, operator, dyrektor.

Redundancja – projektowalny stopień niezawodności

Dla podwyższenia niezawodności systemów automatyki wielu producentów wprowadziło sprzętowe i programowe rozwiązania redundancyjne. Polegają one na podwojeniu, a nawet potrojeniu elementów systemu, oraz na zapewnieniu komunikacji pomiędzy tymi elementami. Również producenci systemów SCADA przewidują takie

rozwiązania, a więc redundancje stacji operatorskich, nadzorczych, serwerów oraz redundancji połączeń sieciowych pomiędzy stacjami systemu oraz stacjami operatorskimi i sterownikami programowalnymi. Najczęściej są to układy Hot Standby Redundancy („gorąca rezerwa”), czasem nazywane Hot Backup (np. przez firmę PC Soft International) – rys. 7.12.



Rys. 7.12. Prosty przykład redundancji stacji operatorskiej systemu SCADA [7]

Jak wspomniano, omówione cechy systemów SCADA w różnym stopniu występują w konkretnych rozwiązaniach. Zależy to zarówno od projektu, od stopnia jego zaawansowania jak i od cech użytkowych systemu, w którym został on wykonany. Na rynku polskim funkcjonuje co najmniej kilkunastu poważnych producentów lub dystrybutorów tych systemów. Do najważniejszych dostępnych w Polsce systemów SCADA należy zaliczyć: *InTouch* firmy Wonderware, *FIX Dynamics*TM firmy Intellution®, *Wizcon* – PC Soft International, *VIP*TM – Festo, *SIMATIC WinCC* – SIEMENS, *Modicon FactoryLink* – Schneider i *USDATA* oraz *GENESIS for Windows* – Iconics Inc. Z polskich produktów można wymienić: *ASIX* firmy ASKOM Sp. z o.o., *OSA-2* – ARVIS, *PRO-2000* – MikroB S.A.

Literatura

- [1] Kościelny J.M., Systemy nadzorowania i wizualizacji procesów przemysłowych – wymagania, kryteria oceny, *Pomiary Automatyka Kontrola*, 3 – 1998, str. 69 - 72
- [2] Kwieciński J., System FoxSCADA firmy Foxboro, *Pomiary Automatyka Kontrola*, 3 – 1998, str. 89
- [3] Legierski T., Kasprzyk J., Wyrwał J., Hajda J., Programowanie sterowników PLC, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 1998
- [4] Leguttek M., GSM-Control i SCADAAlarm – system alarmowania za pomocą urządzeń telekomunikacyjnych, *Biuletyn Automatyki ASTOR 3/2000 (25)*, str. 9 - 11

W niniejszym rozdziale szereg informacji i zdjęć zaczerpnięto z dysków informacyjno-reklamowych:

- [5] CD ASTOR 97
- [6] ASTOR CD 1999/2000
- [7] Wizcon Puts It All Together - SCADA Applications Development Tool, PC SOFT International

Pojęcia podstawowe związane z systemami monitoringu.

Monitorowanie	jest zdolnością do wyświetlania operatorowi w czasie rzeczywistym danych związanych z automatyzowanym obiektem. Dostępne są różnorodne numeryczne, tekstowe i graficzne formaty wyświetlania danych mające zapewnić uzyskanie lepszej czytelności i lepszego dostępu do danych.
Sterowanie nadrzędne	jest to zdolność do monitorowania danych w czasie rzeczywistym zespolona z możliwością zmiany przez operatora wartości zadanej, jak również innych kluczowych parametrów procesu z poziomu komputera sterującego.
Alarmowanie	jest zdolnością rozpoznawania zdarzeń wyjątkowych i natychmiastowego raportowania tych zdarzeń.
Sterowanie	w przypadku systemów SCADA pod pojęciem tym rozumiemy zdolność do automatycznego stosowania algorytmów dla dostrajania zmiennych procesowych, a więc i do utrzymywania tych zmiennych w zadanych ograniczeniach. Sterowanie jest posunięte o krok do przodu w stosunku do sterowania nadrzędnego poprzez usunięcie konieczności bieżącego współdziałania z człowiekiem. Użytkownik może wykorzystywać komputer do sterowania całym procesem lub tylko jego częścią.

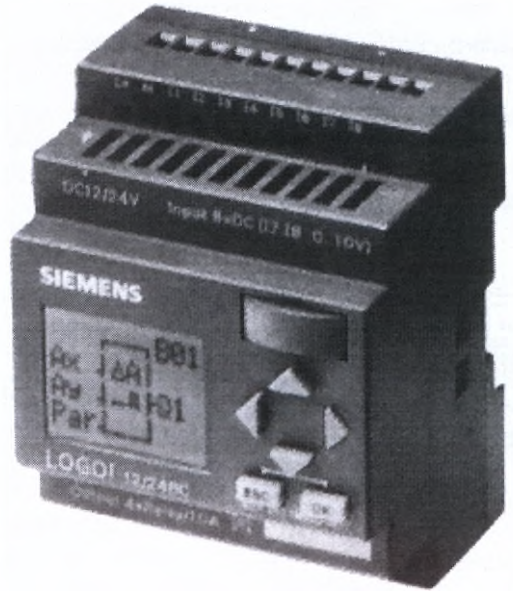
8. Podsumowanie

Na podstawie dokonanego przeglądu sterowników PLC, metod ich programowania oraz możliwości użytkowych stwierdzić można, że należą do najwyszczególniejszych urządzeń stosowanych we współczesnej automatyce. Spotkać można je dosłownie wszędzie. Sterują pojedynczymi przejściami dla pieszych, szeregiem powiązanych ze sobą skrzyżowań, drzwiami w supermarketach, oczyszczalniami ścieków, złożonymi procesami przemysłowymi, znajdują zastosowanie nawet w elektrowniach jądrowych. Różnią się między sobą parametrami, ale łączą ich wspólna filozofia działania i łatwość programowania.

Na zakończenie wymieniono kilka najbardziej prawdopodobnych kierunków rozwoju sterowników PLC.

Dalsza miniaturyzacja

Z uwagi na znaczne zainteresowanie klientów, producenci zaczynają wprowadzać na rynek bardzo niewielkie sterowniki przystosowane do bezpośredniego montażu na listwie DIN. Przykładem takiego rozwiązania jest sterownik LOGO firmy SIEMENS lub ALPHA firmy MITSUBISHI ELECTRIC. LOGO to jedno z pierwszych tego typu urządzeń. Znajduje zastosowanie zarówno w przemyśle jak i w gospodarstwie domowym. Jego zaletą jest wbudowany wyświetlacz LCD, prosta klawiatura oraz możliwość programowania bez dokładnej znajomości języka programowania. Użytkownik ma bezpośredni wgląd przy programowaniu w blokowy schemat elektryczny układu. Wymienna pamięć EPROM podnosi bezpieczeństwo pracy sterownika i zapewnia możliwość skopiowania programu. Po przyłączeniu interfejsu RS232 możliwe jest programowanie LOGO przy użyciu komputera PC i zasymulowanie działania projektowanego sterownika. Niektóre jego wersje są wyposażone w zegar czasu rzeczywistego i interfejs umożliwiający współpracę ze sterownikiem nadrzędnym.



Rys. 8.1. Mikrosterownik LOGO firmy SIEMENS

Wzrost mocy obliczeniowej

Oprócz rozwoju sterowników mikro i kompaktowych, następuje ciągły rozwój sterowników PLC w kierunku stosowania coraz szybszych procesorów i większych pamięci. Także sterowniki małe wyposażane są w jednostki CPU, które dawniej dostępne były tylko dla sterowników średnich i dużych.

Nowe moduły inteligentne

Oferta modułów inteligentnych (specjalnych) staje się coraz bogatsza. Firmy wprowadzają nowe, bardziej zaawansowane rozwiązania. W ostatnim czasie coraz więcej z ich oferuje np. moduły Fuzzy Logic czy też moduły głośnomówiące.

Integracja PLC z panelami operatorskimi

Do sterowania pojedynczymi urządzeniami stosuje się niekiedy sterowniki zespolone z panelem operatorskim. Dzięki takiemu rozwiązaniu można uzyskać oszczędności poprzez: redukcję okablowania i czasu instalacji, szybki montaż oraz prostotę programowania. Mogą one stanowić także uzupełnienie systemu monitoringu opartego o systemy SCADA. Przykładem takich konstrukcji jest seria sterowników OCS firmy GE Fanuc Automation.

Współpraca z siecią Internet

Coraz częściej wykorzystuje się sieć Internet do programowania sterowników i monitoringu. Dzięki użyciu specjalnych modułów tzw. Web Serwerów sterownik może być włączony w sieć ogólnosiwiatową.

Dalsza standaryzacja

Nieunikniona wydaje się także postępująca standaryzacja urządzeń i oprogramowania. Użytkownik nie powinien być skazany na jednego dostawcę. Często już teraz do sterowników jednego producenta można stosować moduły pochodzące z innych firm. Przykładem standaryzacji w dziedzinie komunikacji jest sieć LonWorks stosowana głównie do sterowania inteligentnymi budynkami. Na świecie istnieją setki producentów sprzętu (od czujników, poprzez sterowniki, a na elementach wykonawczych kończąc) zgodnego z tym standardem.

Integracja programów narzędziowych do obsługi sterowników PLC z programami do wizualizacji

Zaletą takiego rozwiązania jest przede wszystkim jedna wspólna baza danych, co w znakomity sposób ułatwia i przyspiesza zarówno proces tworzenia jak i późniejszej obsługi konkretnej aplikacji. Do niedawna systemy takie były ograniczone do urządzeń i oprogramowania jednego producenta – np. program VIP firmy Festo. Obecnie zaczynają pojawiać się rozwiązania pozwalające na współpracę z taką jedną bazą danych elementów automatyki różnych producentów – najnowszy program CIPLICITY firmy GE Fanuc.

Literatura

- [1] Skowronek A., Internet w sterownikach programowalnych Modicon TSX, Napędy i sterowanie, Nr 4(12) Rok II, 2000, str. 24 - 25
- [2] Januszek M., CIMPLICITY ME – zintegrowany system programów narzędziowych, Biuletyn Automatyki, Grudzień 2001, ASTOR 4/2001 (30), str. I – II

Dodatek D1. Prezentacja sterowników firmy GE Fanuc

Poniżej przedstawiono krótką charakterystykę sterowników PLC oferowanych przez firmę GE Fanuc Automation. Prezentowane informacje oparte zostały o dane z II połowy 2001 roku. Ze względu na bardzo szybki rozwój tych urządzeń, zaleca się Czytelnikowi weryfikację zamieszczonych danych. Informacje o najnowszych produktach i aktualnie produkowanym sprzęcie firmy GE Fanuc można znaleźć na stronie internetowej firmy Astor (autoryzowany sprzedawca sterowników GE Fanuc Automation od 1992 roku) www.astor.com.pl. lub w aktualnym „Katalogu sterowników GE Fanuc”.

Jak wynika z poprzednich rozdziałów sterowniki PLC można podzielić na dwie podstawowe grupy:

1. kompaktowe;
2. modułowe.

Do pierwszej grupy należą dwie serie sterowników proponowanych przez firmę GE Fanuc Automation, a mianowicie seria **VersaMax Nano** i **VersaMax Micro**. Sterowniki modułowe reprezentowane są przez serie: **VersaMax, 90-30** i **90-70**. Nieco dokładniej przedstawiono sterowniki **VersaMax**, jako że jest to stosunkowo nowe rozwiązanie. Szereg informacji o sterownikach serii 90-30 zostało zawartych w rozdziale 3 niniejszego opracowania.

D1.1. Sterowniki programowalne serii VersaMax Nano

Sterowniki serii VersaMax Nano należą do najmniejszych dostępnych sterowników na rynku (rys. D1.1. i D1.2.). Dostępne są dwa ich typy: IC200NDR001 z 6 wejściami dyskretnymi 24 VDC i 4 dwustanowymi przekaźnikowymi wyjściami oraz podobny IC200NDD001 lecz wyposażony w 4 dwustanowe wyjścia tranzystorowe. Oba wyposażone są w port RS232 do programowania i komunikacji z innymi urządzeniami. W tym małym sterowniku ukryte są jednak możliwości większych jednostek centralnych.

Pierwsza z nich to możliwość ustawienia w VersaMax Nano portu komunikacyjnego do pracy w protokole SNP/SNP-X, Modbus RTU lub Custom ASCII Read/Write. Dzięki temu można w prosty sposób łączyć Nano z innymi sterownikami w sieć, przy użyciu kabla lub radiomodemu; można też wykorzystać go do współpracy z innymi urządzeniami, np. skanerami kodów kreskowych, czytnikami kart magnetycznych, itp. Kolejna funkcja typowa dla większych sterowników, a zaimplementowana w sterowniku Nano, to możliwość przełączenia wejść sterownika do pracy w tryb jednego dwukierunkowego lub trzech jednokierunkowych liczników sygnałów szybkozmiennych (do 10 kHz). Funkcja ta może posłużyć do sterowania prostymi układami napędowymi w systemach transportowych czy maszynach pakujących; pozwala też - przy użyciu przystawki zamieniającej sygnał analogowy na częstotliwościowy - na wprowadzenie sygnału analogowego. Istnieje możliwość dołączenia do sterownika Nano



Rys. D1.1. Sterownik VersaMax Nano



Rys. D1.2. Porównanie Sterownika VersaMax Nano z dłonią

zewnętrznego wyłącznika RUN/STOP, który może także służyć do kasowania komunikatów w tablicy błędów sterownika. Jednostka centralna sterownika Nano posiada 4 kB pamięci programowej RAM, do której program sterujący jest zawsze przepisywany z nieulotnej pamięci FLASH w trakcie uruchamiania sterownika.

Programowanie sterownika VersaMax Nano odbywa się przy użyciu programu VersaPro z możliwością wykorzystania języka drabinkowego i listy instrukcji (możliwe jest równoczesne stosowanie obu języków wykorzystując procedury programowe). Do dyspozycji są bloki funkcyjne, takie z jakich korzystają użytkownicy sterowników VersaMax czy 90-30. Prócz prostych operacji na bitach lub blokach danych można również zastosować funkcje matematyczne do operacji na liczbach zmiennoprzecinkowych, posłużyć się procedurami lub zastosować gotowe bloki regulatora PID. Sterowniki VersaMax Nano, dzięki konwerterowi VersaMax SE, można włączyć do sieci Ethernet.

Podstawowe dane techniczne sterowników VersaMax Nano

IC200NDD101

- 6 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 4 wyjścia dyskretne tranzystorowe dla napięcia 12/24 VDC,
- port RS232,
- zasilanie prądem stałym 20-30 VDC,
- 3 × szybki licznik do 10 kHz,
- wymiary: 75 x 80 x 47 mm,

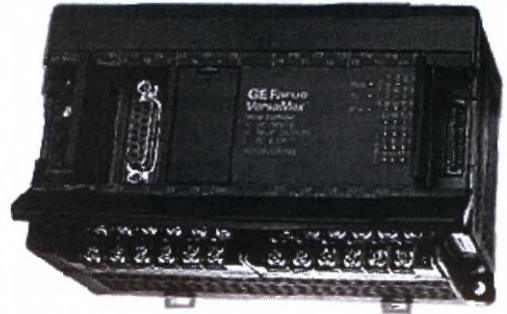
IC200NDR001

- 6 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 4 wyjścia przekaźnikowe dla napięcia 24 VDC i 220 VAC
- port RS232,
- zasilanie prądem stałym 20-30 VDC,
- 3 × szybki licznik do 10 kHz,
- wymiary: 75 x 80 x 47 mm,

D1.2. Sterowniki programowalne serii VersaMax Micro

Sterowniki VersaMax Micro (rys. D1.3.) zastąpiły rodzinę 90-Micro, jednocześnie zwiększając funkcjonalność sterowników tej grupy. Jedną ze zmian jest zamiana pierwszego portu na standard RS232 ze złączem RJ-45 oraz dodanie przełącznika, który może spełniać trzy funkcje: załączania/wyłączania sterownika, zabezpieczenia przed przegraniem programu i konfiguracji oraz kasowania komunikatów w tablicach błędów sterownika.

W sterownikach VersaMax Micro obsługujących 14 sygnałów dodany został port do podłączenia modułów rozszerzających (maksymalnie czterech), co podnosi elastyczność konfiguracji układu sterowania. Moduły te mogą być oddalone od jednostki głównej na odległość do 2,5 metra. Pierwszy port sterowników Micro obsługujących 14 sygnałów oraz port drugi w sterownikach 23 i 28 punktowych obsługuje 3 porty komunikacyjne (SNP/SNP-X Slave, Serial I/O i modbus RTU Slave).



Rys. D1.3. Sterownik VersaMax Micro

Dodatkowo, w przypadku sterowników VersaMax Micro obsługujących 23 lub 28 sygnałów dwustanowych, port drugi może pracować również jako Master w protokole SNP/SNP-X, dając w ten sposób możliwość zbudowania rozproszonego systemu sterowania. Także możliwości samej jednostki centralnej zostały rozszerzone o obsługę operacji zmiennoprzecinkowych i wykorzystanie procedur w programie.

Do programowania sterowników VersaMax Micro służy oprogramowanie narzędziowe VersaPro lub CIMPLICITY ME Logic Development PLC (to samo co w przypadku sterowników VersaMax Nano), w pełni przystosowane do pracy pod kontrolą systemu operacyjnego Windows 95/98/NT. Sterowniki VersaMax Micro, dzięki konwerterowi VersaMax SE, można włączyć do sieci Ethernet.

Podstawowe dane techniczne sterowników VersaMax Micro (14-punktowe)

IC200UDR001

- 8 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 6 wyjść przekaźnikowych,
- port RS232,
- zasilanie prądem przemiennym 85-264 VAC,
- 4 x szybki licznik do 10 kHz,
- możliwość rozbudowy wejść/wyjść,
- wymiary: 95 x 90 x 76 mm,

IC200UDR002

- 8 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 6 wyjść przekaźnikowych,
- port RS232/RS485,
- zasilanie prądem stałym 20-30 VDC,
- 4 x szybki licznik do 10 kHz,
- możliwość rozbudowy wejść/wyjść,
- wymiary: 95 x 90 x 76 mm,

IC200UAA003

- 8 wejść dyskretnych dla napięcia wejściowego 85-132 VAC,
- 6 wyjść dyskretnych dla napięcia 100-240 VAC,
- port RS232/RS485,
- zasilanie prądem przemiennym 85-264 VAC,
- możliwość rozbudowy wejść/wyjść,
- wymiary: 95 x 90 x 76 mm,

IC200UDD104

- 8 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 6 wyjść dyskretnych dla napięcia 24 VDC,
- port RS232/RS485,
- zasilanie prądem stałym 20-30 VDC,
- 4 x szybki licznik do 10 kHz,
- możliwość rozbudowy wejść/wyjść,
- wymiary: 95 x 90 x 76 mm,

Podstawowe dane techniczne sterowników VersaMax Micro (23/28-punktowe)

IC200UAL006

- 13 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 9 wyjść przekaźnikowych,
- 1 wyjście dyskretne dla napięcia 24 VDC,
- 2 wejścia analogowe,
- 1 wyjście analogowe,
- dwa porty RS232 i RS 485,
- zasilanie prądem przemiennym 85-264 VAC,
- 4 x szybki licznik do 10 kHz,
- możliwość rozbudowy wejść/wyjść,
- wymiary: 150 x 90 x 76 mm,

IC200UDR005

- 16 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 11 wyjść przekaźnikowych,
- 1 wyjście dyskretne dla napięcia 24 VDC,
- dwa porty RS232 i RS485,
- zasilanie prądem przemiennym 85-264 VAC,
- 4 x szybki licznik do 10 kHz,
- możliwość rozbudowy wejść/wyjść,
- wymiary: 150 x 90 x 76 mm,

IC200UDR010

- 16 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 11 wyjść przekaźnikowych,
- 1 wyjście dyskretne dla napięcia 24 VDC,
- dwa porty RS232 i RS485,
- zasilanie prądem stałym 20-30 VDC,
- 4 x szybki licznik do 10 kHz,
- możliwość rozbudowy wejść/wyjść,
- wymiary: 150 x 90 x 76 mm,

IC200UDD110

- 16 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 12 wyjść dyskretnych dla napięcia 24 VDC,
- dwa porty RS232 i RS485,
- zasilanie prądem stałym 20-30 VDC,
- 4 x szybki licznik do 10 kHz,
- możliwość rozbudowy wejść/wyjść,
- wymiary: 150 x 90 x 76 mm,

IC200UAA007

- 16 wejść dyskretnych dla napięcia wejściowego 85-132 VAC,
- 12 wyjść dyskretnych dla napięcia 100-240 VAC,
- dwa porty RS232 i RS485,
- zasilanie prądem przemiennym 85-264 VAC,
- szybki licznik do 10 kHz,
- możliwość rozbudowy wejść/wyjść,
- wymiary: 150 x 90 x 76 mm,

Podstawowe dane techniczne ekspanderów (modułów rozszerzających)

IC200UEX011

- 8 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 6 wyjść przekaźnikowych,
- zasilanie prądem przemiennym 85-264 VAC,
- wymiary: 95 x 90 x 76 mm,

IC200UEX012

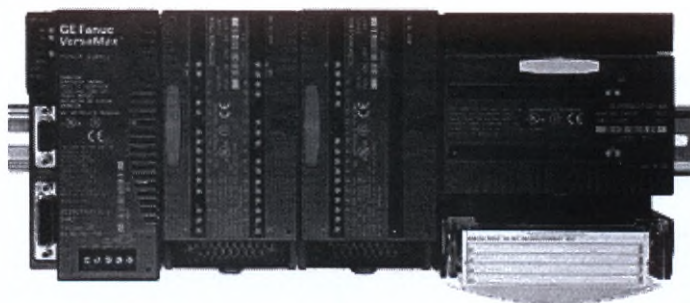
- 8 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 6 wyjść przekaźnikowych,
- zasilanie prądem stałym 20-30 VDC,
- wymiary: 95 x 90 x 76 mm,

IC200UEX014

- 8 wejść dyskretnych dla napięcia wejściowego 24 VDC,
- 6 wyjść dyskretnych dla napięcia 24 VDC,
- zasilanie prądem stałym 20-30 VDC,
- wymiary: 95 x 90 x 76 mm.

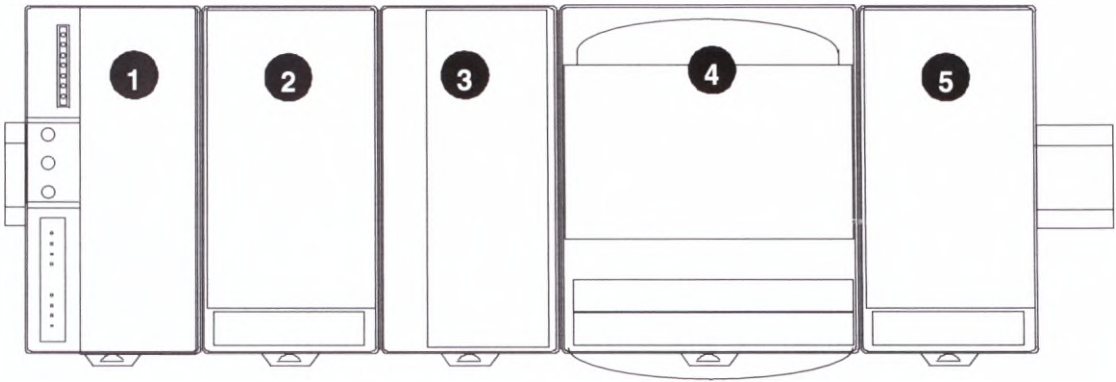
D1.3. Sterowniki programowalne serii VersaMax

VersaMax (rys. D1.4.) to całkowicie nowy, uniwersalny sterownik opracowany przez firmę GE Fanuc Automation. VersaMax jest systemem modułowym, który może pracować jako niezależny sterownik lub może stanowić (jako typowy sterownik lub tylko jako odrębny układ wejść/wyjść) część większego, rozproszonego systemu sterowania, zawierającego również inne sterowniki PLC i układy wejść/wyjść innych serii, panele operatorskie, komputery PC, itp.



Rys. D1.4. Sterownik VersaMax

Sterownik VersaMax jest budowy modułowej, gdzie każdy element jest montowany bezpośrednio na listwie montażowej DIN. Prócz jednostki centralnej lub interfejsu komunikacyjnego, może posiadać 8 modułów do obsługi sygnałów obiektowych. Dzięki zastosowaniu modułów nadawczych i odbiorczych istnieje również możliwość podłączenia dodatkowych 7 kaset rozszerzających (w każdej może być 8 modułów). Moduły VersaMax łączy się zatrzaskowo, bezpośrednio ze sobą - nie wymagają żadnej kasety ani kabli połączeniowych. Sterowniki VersaMax posiadają wbudowany system autokonfiguracji, jak również mogą być konfigurowane przy pomocy oprogramowania VersaPro i CIMPLICITY ME. Moduły zasilaczy, jednostek centralnych i moduły komunikacyjne można instalować tylko w pozycji pionowej, natomiast moduły wejść/wyjść można łączyć z pozostałymi modułami również obrócone o 90°. Moduły wejść/wyjść mogą być wyposażone w oddalone listwy zaciskowe kilku typów, połączone z modułami za pomocą kabli. Przykładową konfigurację sterownika przedstawia rys. D1.5. W dalszej części opracowania omówiono poszczególne elementy układu.



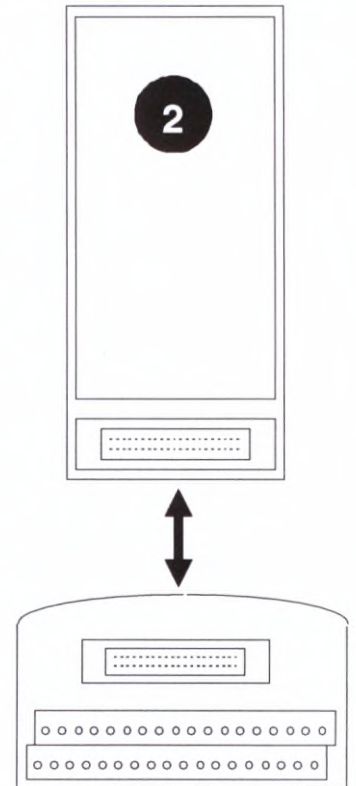
Rys. D1.5. Przykład konfiguracji sterownika VersaMax

1. Jednostka centralna lub interfejs komunikacyjny wraz z zasilaczem

Pierwszym elementem systemu VersaMax jest jednostka centrala jeśli VersaMax będzie pracował jako sterownik. Jeżeli przeznaczeniem VersaMax jest praca jako układ rozproszonych wejść/wyjść w miejsce jednostki centralnej instalowany jest interfejs komunikacyjny (NIU - ang. Network Interface Unit) do sieci GENIUS, Profibus DP lub DeviceNet. Istnieje także możliwość zastosowania modułu dla sieci Ethernet. Bezpośrednio na module jednostki centralnej (interfejsu komunikacyjnego) jest montowany zasilacz sterownika dostarczający napięcie +3,3 i +5 VDC dla modułów wejść/wyjść. Z lewej strony jednostki centralnej dołączany jest moduł IC200ETM001 jeżeli w sterowniku będą stosowane kasyety rozszerzające.

2. Moduł do obsługi sygnałów obiektowych z podstawką do połączenia z oddalonym terminalem połączeniowym

Każdy z modułów wejść/wyjść jest montowany w oddzielnej podstawce, która stanowi część magistrali systemowej sterownika. Rozwiązanie takie daje możliwość wymiany modułu bez konieczności odpinania kabli obiektowych, wyłączenia zasilania i zatrzymania pracy sterownika. Podstawki pod moduły mogą zawierać terminal do podłączenia sygnałów lub gniazdo do podłączenia zdalnego terminala (element 2 rys. D1.6.). W tym przypadku moduł jest montowany pionowo, a poprzez złącze znajdujące się w dolnej części podstawki sygnały są doprowadzane do modułu z terminala oddalonego od sterownika. Sygnały mogą też być połączone bezpośrednio z szyny zaciskowej w szafie, jednak w tym wypadku niezbędne jest wykonanie kabla indywidualnie.



Rys. D1.6. Podstawka pod moduł

3. Dodatkowy zasilacz

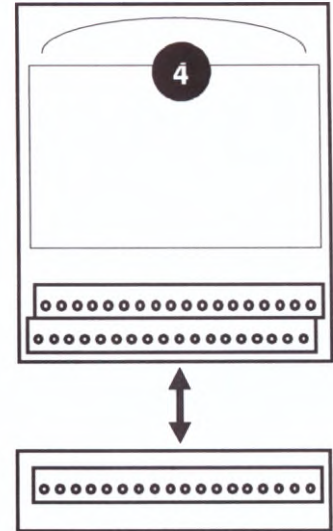
W wypadku stosowania w sterowniku modułów ze zwiększonym poborem prądu z magistrali wewnętrznej sterownika, może okazać się konieczne zastosowanie dodatkowego zasilacza. W takim wypadku pomiędzy moduły sterownika wpina się specjalną podstawkę (IC200PWB001) wraz ze standardowym zasilaczem sterownika. Po zamontowaniu wszystkie moduły znajdujące się na prawo od zasilacza dodatkowego będą przez niego zasilane.

4. Moduł zamontowany w podstawce z terminalami przyłączeniowymi (rys. D1.7.)

Każdy z modułów wejść/wyjść może być zamontowany w podstawce posiadającej terminal do podłączenia kabli sygnałowych. Każda z podstawek posiada 36 złączy, których liczbę można powiększyć poprzez dołożenie kolejnych listew zaciskowych.

5. Moduł komunikacyjny

VersaMax w wypadku pracy jako sterownik umożliwia podłączenie modułu komunikacyjnego do sieci Profibus-DP lub DeviceNet. W tym celu należy do sterownika dołączyć podstawkę dla modułów komunikacyjnych (IC200CHS006), do której wpinany jest moduł komunikacyjny.



Rys. D1.7. Podstawka z terminalami przyłączeniowymi

W skład systemu VersaMax wchodzi następujące moduły:

Jednostki centralne

IC200CPU001 rys. D1.8.

Jednostka centralna, program sterujący 12 kB, szybkość wykonywania programu 1,80 ms/kB, port RS232 i RS485



Rys. D1.8. CPU IC200CPU001

IC200CPU002

Jednostka centralna, program sterujący 20 kB, szybkość wykonywania programu 1,80 ms/kB, port RS232 i RS485

IC200CPU005

Jednostka centralna, program sterujący 64 kB, szybkość wykonywania programu 0,80 ms/kB, port RS232 i RS485

IC200CPUE05

Jednostka centralna, program sterujący 64 kB, szybkość wykonywania programu 0,80 ms/kB, port RS232 i RS485, 10 Mbit port sieci Ethernet

Moduły rozszerzające

IC200ETM001 Moduł komunikacyjny nadawczy do kaset rozszerzających montowany do IC200CPUxxx

IC200ERM001 Moduł komunikacyjny do kaset rozszerzających - max 750 m; z izolacją

IC200ERM002 Moduł komunikacyjny do kaset rozszerzających - max 15 m; bez izolacji

Uwaga:

- Moduł ETM001 należy stosować dla kilku kaset rozszerzających lub dla jednej kasety rozszerzającej, gdy odległość jest większa niż 1 m.
- Moduł ERM002 należy stosować dla kaset rozszerzających na dystansie krótszym niż 15 m.
- Moduł ERM001 należy stosować na dystansie od 15 do 750 m tylko gdy do jednostki centralnej jest dołączony ETM001. Moduł posiada port z optoizolacją.
- VersaMax pozwala na zastosowanie max 7 kaset rozszerzających.
- Stosując kasety rozszerzające na dystansie większym niż 1 m wymagany jest terminator ACC201 montowany w ostatniej kasecie. Terminator jest dostarczany z modułem ETM001.

Zasilacze

IC200PWR001

Zasilacz 24 VDC – rys.D1.9

IC200PWR002

Zasilacz 24VDC ze zwiększoną obciążalnością 3,3 VDC

IC200PWR101

Zasilacz 120/240 VAC

IC200PWR102

Zasilacz 120/240 VAC ze zwiększoną obciążalnością 3,3 VDC

IC200PWB001

Podstawa pod dodatkowy zasilacz



Rys. D1.9. Zasilacz IC200PWR001

Kasety montażowe

IC200CHS001 Kasetka I/O typu Barrier

IC200CHS002 Kasetka I/O typu Box – rys. D1.10

IC200CHS003 Kasetka I/O typu Conector – rys. D1.11

IC200CHS005 Kasetka I/O typu Spring

IC200CHS006 Kasetka do modułów komunikacyjnych

IC200CHS011 Interfejs połączeniowy dla kasetki typu Barrier

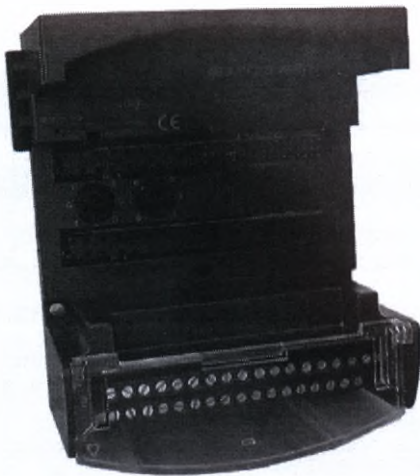
IC200CHS012 Interfejs połączeniowy dla kasetki typu Box

IC200CHS014 Kasetka I/O typu Box do podłączania termopar zapewniająca kompensację zimnych końców

IC200CHS015 Interfejs połączeniowy dla kasetki typu Spring

IC200CHS022 Kasetka I/O typu Box do montażu pionowego modułów

IC200CHS025 Kasetka I/O typu Spring do montażu pionowego modułów



Rys. D1.10. Kaseta montażowa IC200CHS002



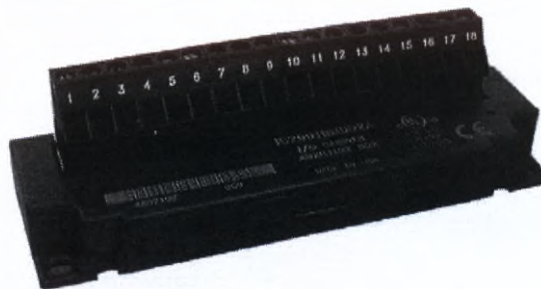
Rys. D1.11. Kaseta montażowa IC200CHS003

Dodatkowe terminale dołączane do podstawek pod moduły wejść/wyjść

IC200TBM001 Zewnętrzny terminal połączeniowy typu Barrier

IC200TBM002 Zewnętrzny terminal połączeniowy typu Box – rys. D1.12.

IC200TBM005 Zewnętrzny terminal połączeniowy typu Spring



Rys. D1.12. Dodatkowy terminal IC200TBM002

Moduły wejść dyskretnych

IC200MDL141 8 wejść 240 VAC, logika dodatnia

IC200MDL241 16 wejść 240 VAC, logika dodatnia

IC200MDL640 16 wejść 24 VDC, logika dodatnia/ujemna

IC200MDL650 32 wejścia 24 VDC, logika dodatnia/ujemna

Moduły wyjść dyskretnych

IC200MDL730	8 wyjść 24 VDC, 2 A
IC200MDL740	16 wyjść 24 VDC, 0.5 A, logika dodatnia
IC200MDL741	16 wyjść 24 VDC, 0.5 A, logika dodatnia, zabezpieczenie przed zwarcieniem i przeciążeniem
IC200MDL742	32 wyjścia 24 VDC, 0.5 A, logika dodatnia, zabezpieczenie przed zwarcieniem i przeciążeniem
IC200MDL750	32 wyjścia 24 VDC, 0.5 A, logika dodatnia
IC200MDL930	moduł izolowanych wyjść przekaźnikowych, 8-punktowy, 2 A
IC200MDL940	moduł izolowanych wyjść przekaźnikowych, 16 punktowy, 2 A

Moduły wejść/wyjść dyskretnych mieszanych

IC200MDD840	Moduł kombinowany 20 wejść dyskretnych 24 VDC i 12 wyjść przekaźnikowych 2 A
IC200MDD841	Moduł kombinowany 20 wejść dyskretnych 24 VDC i 12 wyjść dyskretnych 24 VDC, 0,5 A, HSC
IC200MDD842	Moduł kombinowany 16 wejść dyskretnych 24 VDC i 16 wyjść dyskretnych 24 VDC, 0,5 A z ESCP
IC200MDD843	Moduł kombinowany 10 wejść dyskretnych 24 VDC i 6 wyjść przekaźnikowych 2 A
IC200MDD844	Moduł kombinowany 16 wejść dyskretnych 24 VDC i 16 wyjść dyskretnych 24 VDC, 0,5 A – rys. D1.13
IC200MDD845	Moduł kombinowany 16 wejść dyskretnych 24 VDC i 8 wyjść przekaźnikowych 2 A
IC200MDD847	Moduł kombinowany 8 wejść dyskretnych 220 VAC i 8 wyjść dyskretnych 2 A

Moduły wejść analogowych

Rys. D1.13. Moduł kombinowany dwustanowy IC200MDD843

IC200ALG230	Moduł wejść analogowych, 4-kanalowy, napięciowo-prądowy
IC200ALG240	Moduł wejść analogowych, izolowanych, 8-kanalowy, napięciowo-prądowy
IC200ALG260	Moduł wejść analogowych, 8-kanalowy, napięciowo-prądowy
IC200ALG620	Moduł wejść do pomiaru temperatury, 4-kanalowy
IC200ALG630	Moduł wejść termoparaowych, 7-kanalowy

Moduły wyjść analogowych

IC200ALG320	Moduł wyjść analogowych, 4-kanalowy, prądowy
--------------------	--

IC200ALG321 Moduł wyjść analogowych, 4-kanałowy, napięciowy, 0-10 VDC

IC200ALG322 Moduł wyjść analogowych, 4-kanałowy, napięciowy, +/- 10 VDC

IC200ALG331 Moduł wyjść analogowych, 4-kanałowy, napięciowy/prądowy, rozdzielczość 16 bitów, izolowany

Moduły wejść/wyjść analogowych mieszanych

IC200ALG430 Moduł kombinowany 4 wejść i 2 wyjść analogowych, prądowy – rys. D1.14.

IC200ALG431 Moduł kombinowany 4 wejść 0-10 VDC i 2 wyjść analogowych



Rys. D1.14. Moduł analogowy mieszany IC200ALG430

Interfejsy komunikacyjne - NIU

IC200DBI001 Interfejs komunikacyjny do sieci DeviceNet

IC200GBI001 Interfejs komunikacyjny do sieci GENIUS – rys. D1.15.

IC200PBI001 Interfejs komunikacyjny do sieci Profibus DP – rys. D1.16.

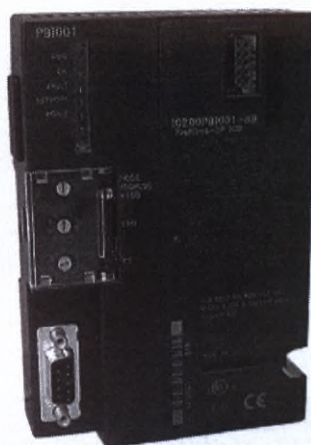
IC200EBI001 Interfejs komunikacyjny do sieci Ethernet

Uwaga:

Moduły te montowane są w miejsce jednostki centralnej, gdy system VersaMax jest wykorzystywany jako układ wejść/wyjść rozproszonych.



Rys. D1.15. Interfejs GENIUS IC200GBI001



Rys. D1.16. Interfejs Profibus DP IC200PBI001

Moduły komunikacyjne

IC200BEM002

Moduł komunikacyjny Slave do sieci Profibus DP – rys. D1.16.

IC200BEM103

Moduł komunikacyjny Master/Slave do sieci DeviceNet

Uwaga:

Moduły komunikacyjne do sterowników VersaMax stosuje się, gdy łączymy je w sieć. Z modułem wymagane jest zastosowanie podstawki IC200CHS006.



Rys. D1.17. Moduł komunikacyjny IC200BEM002

Aksesoria do sterowników VersaMax

IC200ACC001 Zapasowa bateria do jednostki centralnej sterownika VersaMax

IC200ACC201 Zapasowy terminator do kaset rozszerzających sterownika VersaMax

IC200ACC202 Wtyczki do kabla do kaset rozszerzających sterownika VersaMax

IC200ACC301 Atrapa modułu wejść/wyjść (wypełniacz)

IC200ACC302 16-punktowy symulator wejść

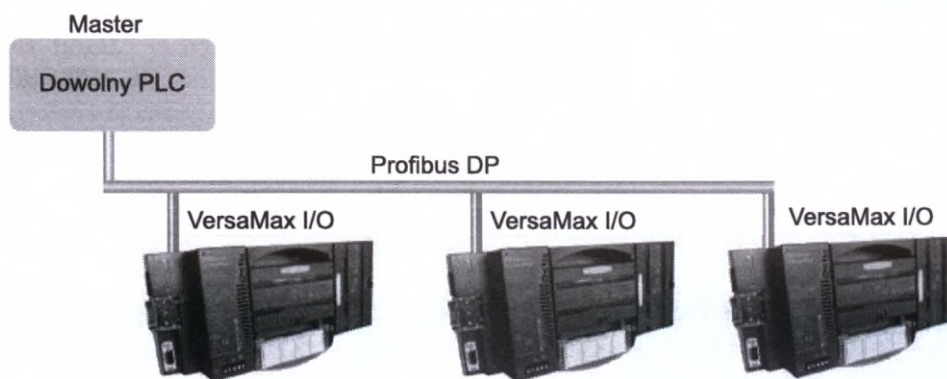
IC200ACC304 Wtyczki do kabli połączeniowych do oddalonych terminali

D1.3.1. Układ rozproszonych wejść/wyjść VersaMax

Stosując urządzenia I/O VersaMax z odpowiednimi interfejsami komunikacyjnymi można otrzymać nowoczesne i ekonomiczne rozwiązanie układu wejść/wyjść rozproszonych dla sieci Profibus DP, GENIUS lub DeviceNet.

Sieć Profibus DP

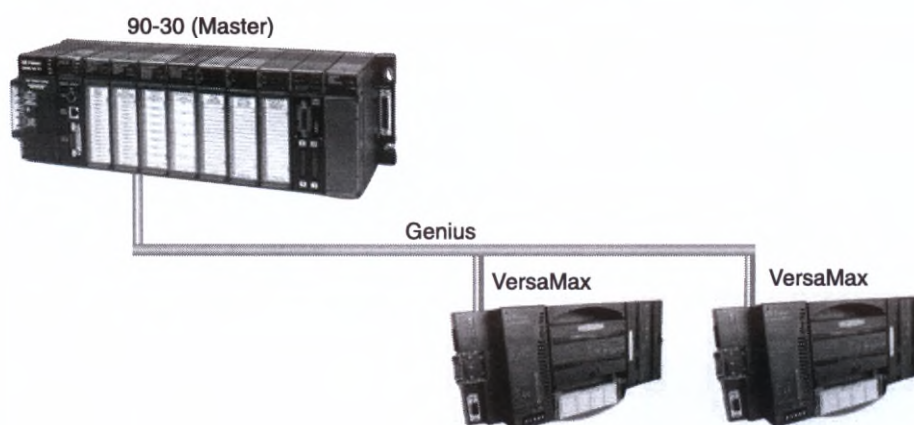
VersaMax w sieci Profibus DP może spełniać rolę układu wejść/wyjść rozproszonych dla sterowników dowolnych producentów (rys. D1.18.). W takim przypadku do konfiguracji VersaMax I/O nie jest wymagane żadne oprogramowanie konfiguracyjne.



Rys. D1.18. Układy I/O serii VersaMax w sieci Profibus DP

Sieć GENIUS

Sterownik VersaMax może pracować jako układ wejść/wyjść rozproszonych w sieci GENIUS – rys. D1.19.

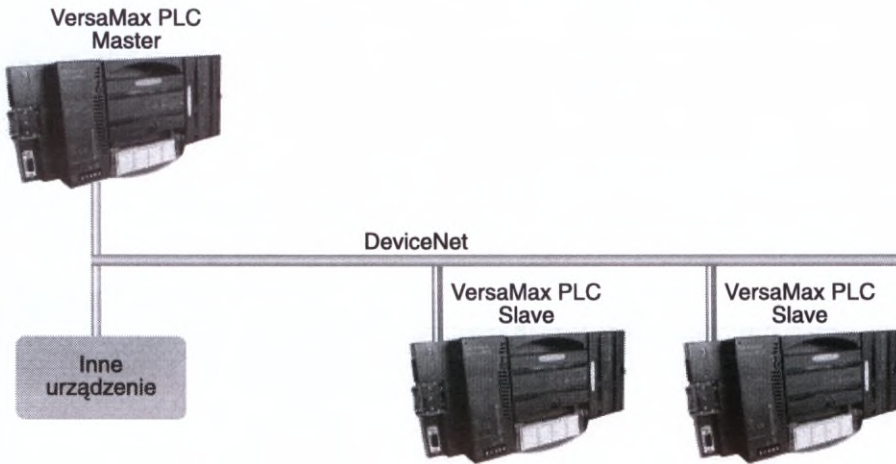


Rys. D1.19. Zastosowanie urządzeń serii VersaMax w sieci GENIUS

Sieć DeviceNet

W sieci DeviceNet VersaMax może pracować jako sterownik pełniąc rolę sterownika nadrzędnego (Master) lub podrzędnego (Slave). Może także pełnić rolę układu wejść/wyjść rozproszonych pracujących dla sterowników innych producentów.

System VersaMax jest programowany za pomocą oprogramowania VersaPro i CIMPLICITY ME dla Windows 95/98/NT i 2000. Posiada szerokie możliwości diagnostyczne, jest łatwy w instalacji i obsłudze.



Rys. D1.20. Urządzenia VersaMax w sieci DeviceNet

D1.4. Sterowniki programowalne GE Fanuc serii 90-30

Sterowniki programowalne serii 90-30 (rys. D1.21.) są łatwymi w instalacji, obsłudze oraz programowaniu sterownikami średniej wielkości o bardzo szerokim zakresie zastosowań. Są obecnie jednymi z najbardziej popularnych sterowników przemysłowych w Polsce. Sterowniki serii 90-30 mogą pracować jako autonomiczne systemy sterowania, bądź też stanowić część większych układów. Znajdują zastosowanie zarówno przy automatyzacji pojedynczych maszyn (np. wtryskarek), jak i kompletnych procesów produkcyjnych (zrobotyzowane linie montażowe, procesy ciągłe np. w przemyśle chemicznym i innych). Oferują również całą gamę dodatkowych funkcji w oparciu o specjalizowane moduły i oprogramowanie. Programowane mogą być przy użyciu oprogramowania Logicmaster 90 dla DOS (pracującego jako sesja DOS w środowisku Windows), oprogramowania VersaPro, uruchamianego na komputerze typu PC z systemem operacyjnym Windows 95/98/NT oraz najnowszego systemu CIMPLICITY. Procesory CPU311 do CPU341 mogą być skonfigurowane i programowane programatorem ręcznym PRG300.



Rys. D1.21. Sterownik serii 90-30

Podstawowe zalety sterowników serii 90-30 to:

Łatwość instalacji systemu dzięki modułowej budowie opartej na 5- lub 10- gniazdowych kasetach. Montaż i demontaż modułów nie wymaga żadnych narzędzi.

Możliwość modyfikowania i rozszerzania poprzez dołączanie dodatkowych kaset (maksymalnie siedem), które mogą być oddalone od kasyety zawierającej jednostkę centralną nawet o 210 m. Tym samym unika się budowania specjalizowanych sieci komunikacyjnych.

Bogaty wybór modułów typowych i specjalizowanych umożliwiające stworzenie jednorodnego systemu sterowania, co w konsekwencji likwiduje problem różnorodności urządzeń sterujących w zakładzie. Kaseca podstawowa systemu zawiera jednostkę centralną. Dostępny jest szereg jednostek centralnych o bardzo zróżnicowanych właściwościach (każda kaseca musi zawierać moduł zasilacza). Oprócz standardowych modułów wejść i wyjść (dyskretnych i analogowych), w sterowniku można zainstalować moduły współpracujące bezpośrednio z urządzeniami pomiarowymi (termopary, termometry oporowe, tensometry), jedno- i dwuosiove moduły pozycjonujące (stosowane w serwonapędach), moduły sterowania silnikami krokowymi, moduły licznika impulsów wysokiej częstotliwości, moduły procesora sygnałów szybkozmiennych, moduł programowalnego koprocatora oraz szereg modułów komunikacyjnych.

Zgodność z innymi sterownikami GE Fanuc w zakresie konfiguracji i oprogramowania zarówno z większymi sterownikami serii 90-70, jak i z małymi sterownikami serii VersaMax Micro, VersaMax Nano oraz VersaMax.

Łatwość programowania i konfigurowania „on-line” za pomocą oprogramowania VersaPro poprzez RS232, sieć Ethernet (TCP/IP), łącze telefoniczne lub radiomodemy. Program pracuje w środowisku Windows 95/98/NT i pozwala na programowanie w języku drabinkowym (LD) lub w postaci listy instrukcji (IL). Zestaw funkcji wykorzystywanych w programie sterującym obejmuje styki, przekaźniki czasowe, liczniki, funkcje matematyczne, relacje, operacje na pojedynczych bitach i blokach danych, operacje tablicowe, funkcje konwersji i różne funkcje sterujące, w tym algorytm regulatora PID z możliwością programowego dostrajania parametrów regulatora w trakcie trwania procesu. Programowanie sterownika ma charakter strukturalny – można wykorzystywać procedury.

Różnorodne sposoby komunikacji z innymi elementami systemu realizowane:

- przy wykorzystaniu wbudowanych portów RS232/RS485, modułów komunikacyjnych lub portów RS232/RS485 jednostek centralnych oraz protokołów komunikacyjnych: SNP/SNPX, CCM2, Modbus RTU, Custom ASCII i innych;
- przy wykorzystaniu bardzo szybkiej magistrali komunikacyjnej GENIUS (będącej parą skręconych, ekranowych przewodów) i modułów komunikacyjnych;
- przy wykorzystaniu standardowych magistral Modbus, Profibus DP, Fip, Interbus S, LonWorks i wielu innych;
- przy wykorzystaniu najnowocześniejszej i najbardziej popularnej standardowej magistrali komunikacyjnej Ethernet i modułu komunikacyjnego TCP/IP Ethernet.

Sterowniki serii 90-30 umożliwiają także:

- zabezpieczenie dostępu do programu sterującego oraz pamięci sterownika;

- diagnostykę i konfigurowanie modułów sterownika;
- programowanie w językach: SFC (Sequential Function Chart), Microsoft C, Megabasic, State Logic.

Podstawowe dane techniczne jednostek centralnych.

Jednostki 16-bitowe (Low End)

Model 311 (jednostka wbudowana w kasetę 5-gniazdową)

- procesor 80188, 8 MHz,
- możliwość obsługi 160 wejść/wyjść,
- 512 rejestrów z danymi,
- program sterujący do 6 kB,
- szybkość wykonywania programu sterującego (zawierającego tylko styki): 18 ms/kB.

Modele 313 i 323 (jednostka wbudowane w kasety odpowiednio 5 i 10 gniazdowe)

- procesor 80188, 10 MHz,
- możliwość obsługi 160/320 wejść/wyjść,
- 1024 rejestrów z danymi,
- program sterujący do 12 kB,
- szybkość wykonywania programu sterującego (zawierającego tylko styki): 0,6 ms/kB.

Model 331 (jednostka centralna jako osobny moduł)

- procesor 80188, 8 MHz,
- możliwość obsługi 1024 wejść/wyjść,
- 2048 rejestrów z danymi,
- program sterujący do 16 kB,
- szybkość wykonywania programu sterującego (zawierającego tylko styki): 0,4 ms/kB,
- możliwość rozszerzenia systemu do 5 kaset (49 gniazd).

Model 341 (jednostka centralna jako osobny moduł)

- procesor 80188XL, 20 MHz,
- możliwość obsługi 1024 wejść/wyjść,
- 9999 rejestrów z danymi,
- program sterujący do 80 kB,
- szybkość wykonywania programu sterującego (zawierającego tylko styki): 0,3 ms/kB,
- możliwość rozszerzenia systemu do 5 kaset (49 gniazd),
- ze względów finansowych zastępowany jest przez CPU350 lub CPU 360.

Jednostki 32-bitowe (High End)

Model 350 (jednostka centralna jako osobny moduł)

- procesor 80386EX, 25 MHz,
- możliwość obsługi 4096 wejść/wyjść,

- 9999 rejestrów z danymi,
- program sterujący do 32 kB,
- szybkość wykonywania programu sterującego (zawierającego tylko styki): 0,22 ms/kB,
- możliwość rozszerzenia systemu do 8 kaset (79 gniazd),
- blokada dostępu do pamięci Flash,
- możliwość wykonywania operacji zmiennoprzecinkowych.

Model 352 (jednostka centralna jako osobny moduł)

- konfigurowalny obszar rejestrów z danymi – max. 32640,
- konfigurowalny obszar we/wy analogowych – max. 32640,
- pamięć 80 kB,
- wbudowany koprocessor arytmetyczny (sprzętowa realizacja operacji zmiennoprzecinkowych),
- 2 porty komunikacyjne RS232 i RS485,
- pozostałe parametry jak w CPU350.

Model 360 (jednostka centralna jako osobny moduł)

- konfigurowalny obszar rejestrów z danymi – max. 32640,
- konfigurowalny obszar we/wy analogowych – max. 32640,
- pamięć 246 kB,
- pozostałe parametry jak w CPU350.

Model 363 (jednostka centralna jako osobny moduł)

- wbudowane porty komunikacyjne RS232 i RS485 (obsługuje protokoły SNP/SNP-X, MODBUS RTU Slave, Custom ASCII),
- pozostałe parametry jak w CPU360.

Model 364 (jednostka centralna jako osobny moduł)

- wbudowany port komunikacyjny do sieci Ethernet TCP/IP (interfejsy AAUI lub UTP),
- pozostałe parametry jak w CPU360.

Maksymalna liczba modułów wejść/wyjść w systemie 90-30

dla wejść i wyjść dyskretnych		
w systemie z jednostką centralną CPU311/313		10
w systemie z jednostką centralną CPU331/341		49
w systemie z jednostką centralną CPU35x/36x		79
dla 4-kanałowych modułów wejść analogowych	8 (CPU311/313) 64 (CPU35x/36x)	40 (CPU331/341)
dla 16-kanałowych modułów wejść analogowych	4 (CPU311/313) 30 (CPU341)	16 (CPU331) 48 (CPU35x/36x)
dla 2-kanałowych modułów wyjść analogowych, napięciowych	6 (CPU311/313) 30 (CPU341)	16 (CPU331) 48 (CPU35x/36x)
dla 2-kanałowych modułów wyjść analogowych, prądowych	3 (CPU311/313) 15 (CPU341)	15 (CPU331) 24 (CPU35x/36x)
UWAGA: W przypadku, gdy napięcie + 24 VDC jest dostarczane przez użytkownika, liczba modułów wynosi:	3 (CPU311/313) 49 (CPU341)	32 (CPU331) 79 (CPU35x/36x)
dla 8-kanałowych modułów wyjść analogowych	4 (CPU311/313) 32 (CPU341)	8 (CPU331) 79 (CPU35x/36x)
dla modułów kombinowanych 4 wejścia i 2 wyjścia analogowe	10 (CPU311/313) 79 (CPU35x/36x)	21 (CPU331/341)

Maksymalna liczba modułów specjalizowanych w systemie 90-30

licznik impulsów wysokiej częstotliwości		
w systemie z jednostką centralną CPU311/313		4
w systemie z centralną CPU331/341/35x/36x		8
moduły pozycjonujące dla jednej lub dwóch osi		
w systemie z jednostką centralną CPU311/313		4
w systemie z centralną CPU331/341/35x/36x		8
moduł procesora wejść/wyjść		
w systemie z jednostką centralną CPU311/313		4
w systemie z jednostką centralną CPU331/341/35x/36x		8
moduły komunikacyjne		
GENIUS PLUS (CMM302)		4
GENIUS BUS CONTROLLER (BEM331) *		8
Ethernet (CMM321)		max 2 w jednej kasecie
moduł komunikacyjny dla złącz szeregowych (CMM311)		
w systemie z jednostką centralną CPU311/313		nie stosowany
w systemie z jednostką centralną CPU331/341/35x/36x		4
moduły programowalnego koprocatora		
w systemie z jednostką centralną CPU311/313		nie stosowany
w systemie z jednostką centralną CPU331/341/35x/36x		4

* Moduły komunikacyjne GENIUS PLUS nie mogą pracować w tej samej kasecie z modułami GENIUS BUS CONTROLLER

W niniejszym skrypcie, ze względu na dużą różnorodność, nie zostały zaprezentowane szczegółowo moduły i akcesoria dostępne dla serii sterowników 90-30. Zainteresowani Czytelnicy znajdą dodatkowe informacje na stronie internetowej firmy Astor www.astor.com.pl. lub w aktualnym „Katalogu sterowników GE Fanuc”.

D1.5. Sterowniki programowalne GE Fanuc serii 90-70

Sterowniki programowalne serii 90-70 (rys. D1.22.) są przeznaczone dla dużych, złożonych instalacji o znacznej liczbie wejść i wyjść, wymagających dużej szybkości przetwarzania danych, dokładnych obliczeń, wysokiego poziomu bezpieczeństwa (systemy redundantne certyfikowane atestem TÜV w klasie 4, 5 i 6). Są zaprojektowane do pracy w środowisku sieciowym, umożliwiając włączenie do systemu również innych wyrobów GE Fanuc - np. sterowników serii 90-30, elementów systemu GENIUS, Field Control, itp. Jednocześnie, dzięki modułowej konstrukcji i otwartej architekturze systemu, sterowniki serii 90-70 są łatwe w instalacji, obsłudze i programowaniu, oferując wiele dodatkowych funkcji w oparciu o specjalizowane moduły i różnorodne oprogramowanie.



Rys. D1.22. Sterownik serii 90-70

Do najistotniejszych cech serii 90-70 zaliczyć należy:

Otwartą architekturę systemu – konstrukcja sterownika serii 90-70 wykorzystuje magistralę VME, stanowiącą międzynarodowy, otwarty standard. Dzięki VME, oprócz modułów wytwarzanych przez GE Fanuc w systemie 90-70, można zastosować moduły i karty ponad 300 producentów. GE Fanuc współpracuje ściśle z producentami wysokiej jakości systemów wizyjnych, systemów do sterowania ruchem, układów napędowych, modułów komunikacyjnych i innych, całkowicie zgodnych ze sterownikiem serii 90-70. Wykorzystując je można zbudować system o strukturze dokładnie odpowiadającej potrzebom użytkownika.

Dwuprocessorową jednostkę centralną – każda jednostka centralna sterownika serii 90-70 zawiera dwa procesory: mikroprocesor firmy Intel realizujący wszystkie zadania związane ze sterowaniem i obsługą systemu oraz koprocessor logiczny zwiększający szybkość wykonywania instrukcji logicznych. Jednostka centralna prowadzi również samodzielnie rejestrację błędów w systemie bez konieczności dodatkowego programowania.

Bogaty wybór modułów, w tym modułów specjalizowanych – oferta modułów wejść/wyjść sterownika serii 90-70 jest bardzo szeroka. Obejmuje moduły dyskretne 16- i 32-punktowe o różnych napięciach zasilania oraz moduły analogowe o wysokiej rozdzielczości z możliwością skalowania wielkości mierzonej i definiowania alarmów. Oprócz standardowych modułów wejść/wyjść w sterowniku można zainstalować moduły współpracujące bezpośrednio z urządzeniami pomiarowymi (termopary, termometry oporowe, tensometry), wielosiowe moduły pozycjonujące (Power Mate), moduł licznika impulsów wysokiej częstotliwości, moduł programowalnego koprocessora przyspieszający obliczenia prowadzone w czasie rzeczywistym oraz szereg modułów komunikacyjnych (RS232/485, GENIUS, Profibus, FIP, Ethernet TCP/IP).

Programowanie – sterowniki programowalne serii 90-70 mogą być programowane na kilka sposobów, opisanych poniżej.

- W logice drabinkowej za pomocą oprogramowania Logicmaster 90 dla DOS, CIMPLICITY ME lub VersaPro Professional Edition uruchamianego na

komputerze PC. Zestaw funkcji wykorzystywanych w programie sterującym obejmuje styki, przekaźniki czasowe, liczniki, funkcje matematyczne, relacje, operacje na pojedynczych bitach i blokach danych, operacje tablicowe, funkcje konwersji i różne funkcje sterujące, w tym algorytm regulatora PID z możliwością programowego dostrajania parametrów regulatora w trakcie trwania procesu. Programowanie sterownika ma charakter strukturalny - można wykorzystywać procedury. Zmienne mogą być adresowane dynamicznie i pośrednio. Dostęp do programu sterującego oraz pamięci sterownika może być zabezpieczony wielopoziomowo za pomocą haseł dostępu. Oprogramowanie umożliwia również konfigurowanie modułów sterownika oraz szeroką diagnostykę (podgląd pamięci sterownika, tablic błędów, itp.).

- W języku SFC (*Sequential Function Chart*), wykorzystującym grafy przejść do zilustrowania procesu i logikę drabinkową do sformułowania warunków przejść pomiędzy różnymi stanami. Stanowi on nakładkę programową na program Logicmaster 90.
- W języku C, umożliwiającym tworzenie bloków programowych, które są wykonywane kilkakrotnie szybciej niż bloki napisane w logice drabinkowej. Język programowania C używany w sterownikach serii 90-70 wykorzystuje powszechnie stosowany kompilator Microsoft C/C++.
- Przy wykorzystaniu logiki stanów (*State Logic*), opisującej proces jako zbiór zadań, składających się z określonej liczby stanów. Program sterujący napisany w logice stanów jest łatwo diagnozować i modyfikować, może on być też napisany w języku ojczystym operatora (np. polskim).

Komunikację – pomiędzy sterownikiem serii 90-70 i innymi elementami systemu (sterownikami GE Fanuc, systemem wejść/wyjść, komputerami, panelami operatorskimi oraz innymi urządzeniami), która może być realizowana na wiele sposobów przy wykorzystaniu:

- portów szeregowych oraz protokołów komunikacyjnych: SNP/SNPX, CCM2, RTU Modbus,
- magistrali komunikacyjnej GENIUS (będącej parą skręconych, ekranowanych przewodów) i modułów komunikacyjnych,
- magistrali FIP,
- magistrali komunikacyjnej TCP/IP Ethernet,
- praktycznie wszystkich standardów komunikacyjnych z użyciem modułów komunikacyjnych innych producentów (VME).

W skład serii 90-70 wchodzi sterowniki o różnych możliwościach i przeznaczeniu. Oto krótka charakterystyka poszczególnych jednostek centralnych sterowników serii 90-70 pod kątem ich potencjalnych zastosowań:

- Modele CPX732, 772, 782, 935, 928 - standardowe, uniwersalne jednostki centralne o dużej mocy obliczeniowej. Programowane w logice drabinkowej i/lub Microsoft C.
- Modele CGR772, 935 - jednostki centralne przeznaczone do stosowania w systemach redundantnych, gdzie ciągłość pracy systemu ma znaczenie krytyczne. Jednostki te, wyposażone w dodatkowe moduły pozwalają na budowę układu rezerwacji z synchronizacją danych pomiędzy CPU w każdym skanie programu sterującego.

- Modele CPU788, 789, CPM790 - jednostki centralne przeznaczone do stosowania w systemach zabezpieczeń, głównie w instalacjach typu ESD oraz Fire & Gas Detection. W zależności od konfiguracji jednostek centralnych oraz systemu wejść i wyjść układy te posiadają certyfikaty TÜV w klasie 3 – 6 co odpowiada normom SIL 1 do 3.

Szybkość wykonywania programu sterującego zawierającego tylko instrukcje logiczne dla wszystkich jednostek centralnych jest jednakowa i wynosi od 0.22 do 0.4 ms/kB.

Podstawowe dane techniczne jednostek centralnych

CPU	Procesor	Pamięć dla programu	Liczba wejść/wyjść dyskretnych	Liczba wejść/wyjść analogowych
CPX732	80486DX4 96MHz	128 kB	2k	8k
CPX772	80486DX4 96MHz	512 kB	2k	8k
CPX782	80486DX4 96MHz	1024 kB	12k	8k
CPX935	80486DX4 96MHz	1024 fast kB	12k	8k
CPX928	80486DX4 96MHz	6 MB	12k	8k
CGR772	80486DX4 96MHz	512 kB	2k hot-standby	8k
CGR935	80486DX4 96MHz	1024 kB	12k hot-standby	8k
CPU788	80386 16 MHz	206 kB	100 Triplex	8k
CPU789	80386 16 MHz	206 kB	2k Triplex	8k
CPU790	80486DX4 64MHz	512 kB	2k Tipler	8k

W niniejszym skrypcie, ze względu na dużą różnorodność, nie zostały zaprezentowane szczegółowo moduły i akcesoria dostępne dla serii sterowników 90-70. Zainteresowani Czytelnicy dodatkowe informacje znajdą na stronie internetowej firmy Astor www.astor.com.pl.

Literatura

- [1] Katalog sterowników GE Fanuc serii VersaMax Nano i Micro, MiniOCS, OCS, VersaMax, 90-30, Astor, Listopad 2000, Kraków

Część danych i rysunków prezentowanych w Dodatku D1 zaczerpnięto z dysku informacyjno-reklamowego ASTOR CD 2001.

Dodatek D2. Oprogramowanie narzędziowe GE Fanuc CIMPLICITY

Na początku 2002 roku dostępne będzie w Polsce nowy zestaw oprogramowania przemysłowego CIMPLICITY. W jego skład wchodzi:

- zintegrowany system programów narzędziowych **CIMPLICITY Machine Edition (ME)**,
- pakiet do wizualizacji procesów przemysłowych **CIMPLICITY Plant Edition (PE)**.

Poniżej przedstawiono najważniejsze cechy oprogramowania CIMPLICITY ME.

Wspólna baza danych

Oprogramowania odznaczają się bardzo wysokim poziomem integracji poszczególnych programów wchodzących w skład systemu. Dzięki temu, zmienne deklarowane np. podczas pisania programu sterującego dla sterownika są widoczne również dla innych modułów oprogramowania.

Jedno środowisko programistyczne – kilka programów

CIMPLICITY ME składa się z pięciu komponentów (programów). Mogą one tworzyć jeden produkt, lub być używane oddzielnie. Są to:

- CIMPLICITY Logic Developer

Jest to narzędzie do tworzenia programów sterujących pozwalające na:

- programowanie i konfigurowanie sterowników;
- pisanie programów sterujących uruchamianych na komputerach klasy PC (Soft PLC),
- programowanie obrabiarek sterowanych numerycznie,
- programowanie sterowników w języku State Logic (dla wybranych jednostek centralnych sterowników 90-30 i 90-70).

Głównym komponentem tego produktu jest **Logic Developer PLC** służący do konfigurowania PLC i pisania programów sterujących. Funkcjonalnością i sposobem obsługi przypomina oprogramowanie VersaPro. W niedługim czasie ma on zastąpić VersaPro i stać się podstawowym narzędziem do programowania sterowników GE Fanuc.

- CIMPLICITY View

Pakiet do tworzenia wizualizacji na poziomie maszyn. Jego głównym zadaniem jest wizualizacja niewielkich układów. Pomimo to posiada wiele cech typowych systemów SCADA: bibliotekę gotowych elementów, możliwość obsługi alarmów, wykresy bieżące i historyczne, zabezpieczenia dostępu oraz możliwość współpracy z Internetem.

- CIMPLICITY Motion

Oprogramowanie przeznaczone do konfigurowania i programowania napędów elektrycznych GE Fanuc.

- **CIMPLICITY fxManager**
Służy do zarządzania programami, plikami oraz ich wersjami. Zabezpiecza między innymi przed przypadkowymi zmianami w aplikacjach wizualizacyjnych i sterujących.
- **CIMPLICITY QuicView**
Przeznaczony jest do konfigurowania i programowania graficznych paneli operatorskich QuickPanel (dostępny będzie w 2003 roku).

Zgodność ze standardami wymiany danych

Oprogramowanie CIPPLICITY ME obsługuje podstawowe standardy przemysłowe takie jak: XML, COM/DCOM, OPC i ActiveX.

Szybki dostęp do właściwości obiektu

W oprogramowaniu występuje specjalne okno umożliwiające łatwy i szybki dostęp do parametrów wybranego obiektu (może to być zmienna, okno wizualizacji lub cały podsystem).

Rozbudowany system pomocy

Skrypty

Dostępny jest język skryptowy, który umożliwia zapisanie algorytmów trudnych do zdefiniowania w inny sposób.

Przeznaczeniem drugiego składnika pakietu CIMPLICITY tj. CIMPLICITY PE jest wizualizacja procesów przemysłowych lub też całych zakładów produkcyjnych. Służy do tworzenia dużych systemów wizualizacji (do kilkudziesięciu stacji operatorskich) wykorzystujących architekturę klient/serwer z możliwością redundancji. Oprogramowanie to automatycznie wyszukuje podłączone do systemu sterowniki GE Fanuc, lecz umożliwia także komunikację ze sterownikami innych producentów.

Literatura

- [1] Januszek M., Kompleksowa funkcjonalność CIMPLICITY ME, Biuletyn Automatyki RAPORT, Grudzień 2001, ASTOR 4/2001 (30), str. IV – V
- [2] Merwart P., CIMPLICITY ME – zintegrowany system programów narzędziowych, Biuletyn Automatyki, RAPORT, Grudzień 2001, ASTOR 4/2001 (30), str. I – II

Dodatek D3. Porty szeregowo

W systemach sterowników PLC korzysta się najczęściej z komunikacji szeregowo. Dla jej potrzeb wykorzystuje się dwa rodzaje łączy:

- standard RS232;
- standard RS422/RS485.

D2.1. Port RS232

Standard RS232 normalizuje interfejs pomiędzy urządzeniem końcowym dla danych (DTE - Data Terminal Equipment) a urządzeniem komunikacyjnym dla danych (DCE – Data Communication Equipment). W 1962 roku zdefiniowano standard RS-232C, natomiast w 1986 jego nowsza wersję RS-232D. W standardzie RS232 transmisja odbywa się szeregowo bit po bicie, przy czym definiuje się dwa rodzaje transmisji:

- asynchroniczna transmisja znakowa;
- transmisja synchroniczna.

Najczęściej stosowana jest transmisja asynchroniczna. Polega na przesyłaniu pojedynczych znaków, które posiadają ściśle określony format. Początek znaku stanowi bit startu, jałowy z punktu widzenia przesyłanej informacji. Następnie transmitowane jest pole danych (w kolejności od bitu najmniej znaczącego LSB). Za polem danych może wystąpić bit kontrolny. Transmitowany znak kończy jeden lub dwa bity stopu. Typowe wartości szybkości transmisji asynchronicznej znakowej wynoszą: 300, 600, 1200, 2400, 4800, 9600, 19200 bodów [bit/s] przy długości kabla do 15 m. Na odległości rzędu 3 m prędkość transmisji może wynosić 38.400 bodów, a w przypadku jeszcze mniejszych odległości nawet 115.200 bodów. Długość transmitowanych danych to najczęściej 8 bitów. Podczas transmisji może być wprowadzona kontrola transmisji (kontrola parzystości – even parity, kontrola nieparzystości – odd parity).

D3.2. Port RS422/RS485

Współpraca nadajników i odbiorników RS422/RS485 odbywa się zazwyczaj za pośrednictwem kabla o czterech parach skręconych przewodów, przy czym w standardzie RS485 liczba przewodów może ulec zmniejszeniu do jednej pary. Dodatkowo występują przewody: zasilający i GND, które nie biorą jednak bezpośrednio udziału w komunikacji. Zasilanie może być lokalne, nie musi być prowadzone skrętką. Standard RS485 dopuszcza jednoczesne istnienie nadajników i odbiorników na jednej linii. Całkowita długość kabla nie powinna przekraczać:

- 600 m w standardzie RS422;
- 1200 m w standardzie RS485.

Współpraca urządzeń wyposażonych porty RS422/RS485 może być skonfigurowana na tryb Multidrop¹, dzięki czemu możliwa staje się wymiana danych między sterownikami za pośrednictwem łączy szeregowo. Dla przykładu, sterowniki GE Fanuc serii 90-30

¹ Tryb Multidrop polega na współpracy wielu urządzeń za pośrednictwem łączy szeregowo. Jedno z urządzeń jest wyróżnione i nosi nazwę urządzenia Master, pozostałe mają charakter urządzeń podporządkowanych i nazywane są urządzeniami Slave. O zainicjowaniu wymiany informacji decyduje urządzenie Master, urządzenia Slave jedynie odpowiadają na jego polecenie. Systemem odmiennym od Multidrop jest połączenie typu Point-To-Point.

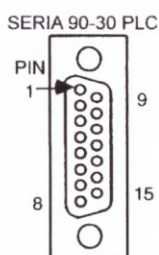
umożliwiają, bez stosowania dodatkowych repeaterów, zastosowanie ośmiu urządzeń w sieci Multidrop. Przy stosowaniu repeaterów można zbudować sieć sterowników 90-30 składającą się maksymalnie z 32 urządzeń.

Jeśli sterowniki GE Fanuc mają współpracować z urządzeniami posiadającymi port RS232 to niezbędne jest zastosowanie konwertera standardów RS232/RS485. W przypadku gdy komputer jest wyposażony w port RS422/RS485 to sterownik można podłączyć bezpośrednio do komputera.

Standardy RS422/RS485 są lepsze od standardu RS232 pod względem odporności na zakłócenia oraz maksymalnej długości toru transmisji. Prędkości transmisji w obu standardach są identyczne.

D3.3. Port szeregowy sterowników (GE Fanuc)

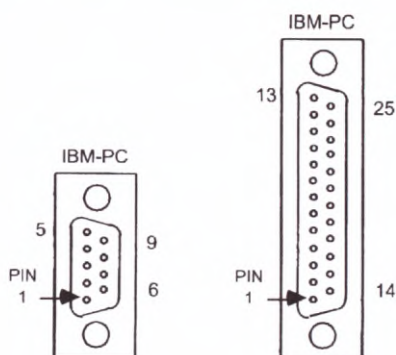
Większość sterowników, w szczególności firmy GE Fanuc, jest kompatybilnych ze standardem RS422/RS485. Do połączenia z urządzeniami wyposażonymi w port RS232 wymagają zatem konwertera standardów RS232/RS485. Rysunek D3.1. przedstawia widok gniazda portu szeregowego sterowników serii 90-30.



Rys. D3.1. Gniazdo portu szeregowego sterowników 90-30

D3.4. Port szeregowy w komputerze IBM-PC

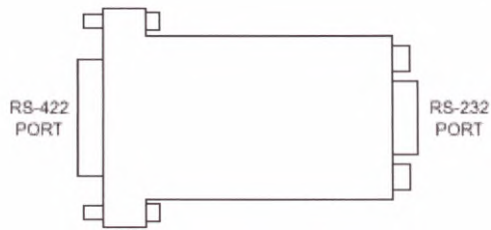
Komputery są najczęściej wyposażone w porty zgodne ze standardem RS232. Posiadają zazwyczaj 9- lub 25-pinowe gniazdo męskie typu CANON (rys. D3.2.). Dla potrzeb komunikacji ze sterownikiem wystarcza gniazdo 9-pinowe.



Rys. D3.2. Gniazda portu szeregowego komputerów IBM-PC

D3.5. Konwerter standardów RS232/RS485

Przedstawiony na rys. D3.3. konwerter umożliwia podłączenie komputera IBM-PC ze sterownikami GE Fanuc po łączy szeregowym. Jest to autonomiczne urządzenie, zasilane z gniazda portu szeregowego sterownika, do którego jest podłączone. Jeśli odległość sterownika od konwertera wynosi powyżej 3 m należy zastosować lokalne zasilanie konwertera. Nadaje się on do współpracy z modemami. Urządzenie komunikujące się z wykorzystaniem konwertera może pracować zarówno w trybie Multidrop jak i Point-To-Point.



Rys. D3.3. Konwerter RS232/RS485

Literatura

- [1] Horowitz P., Hill W., Sztuka elektroniki. Część 2, WKŁ, Warszawa 1995

30,00

Biblioteka Politechniki Krakowskiej



III-309553

Biblioteka Politechniki Krakowskiej



10000340455



ISBN 83-915620-6-9



LI-ASK-SPLC-GE1