

Seria: INFORMATYKA z. 23

Nr kol.1211

Zbigniew BIGEWSKI

## KOMUNIKACJA STEROWNIKÓW ZE ŚWIATEM ZEWNĘTRZNYM

Streszczenie. W publikacji przedstawiono sposoby komunikacji programowalnych sterowników przemysłowych ze światem zewnętrznym za pomocą specjalizowanych modułów koprocesorowych. W niektórych nieskomplikowanych procesach przemysłowych wykorzystanie do komunikacji ze światem zewnętrznym rozwiązań sieciowych nie jest uzasadnione ekonomicznie, a proste informacje o stanie procesu mogą być wysyłane ze sterownika za pomocą modułów komunikacyjnych. W niniejszej publikacji opisano możliwości techniczne tych modułów, sposoby ich programowania, jak również zasady ich komunikacji z jednostką centralną sterownika.

## THE SIMPLE COMMUNICATIONS METHODS WITH PROGRAMMABLE LOGIC CONTROLLERS

Summary. Sometimes a very simply communications are needed. In this cases a network communications is not good solutions. In this paper special hardware coprocessors and dialog modules are described. The communications methods between central processing unit and dialog modules are also presented.

## COMMUNICATION LES CONTROLEURS AVEC L'ENVIRONNEMENT EXTÉRIEUR

Résumé. Dans cette publication il sont présentés les méthodes de communication de contrôleurs industriels programmés avec l'environnement extérieur par les circuits des coprocesseurs spécialisées. Dans certains simples processus industrielles pour une communication, la résolution par les réseaux ne sont pas bien fondés au point de vue

d'économie, mais les informations simples peuvent être transmises par les modules de communication. Dans cet article il est décrits les possibilités techniques de ces modules, les méthodes de leurs programmer, et aussi les principes leurs communication avec l'unité centrale de CONTROLEUR.

## 1. Wprowadzenie

Sterowniki przemysłowe ze swej natury nie są wyposażane w urządzenia służące do wizualizacji działania sterownika. Bez stosowania specjalnych narzędzi nie ma możliwości uzyskania informacji o tym, w jakiej fazie działania znajduje się aktualnie realizowany proces przemysłowy. Informacja o stanie realizacji procesu jest potrzebna nie tylko w fazie uruchamiania i testowania programu, ale przede wszystkim w czasie działania gotowego już programu. Dzięki wizualizacji działania programu istnieje możliwość optymalizacji procesu technologicznego, wykrywania sytuacji nietypowych dla danego zastosowania, jak również możliwość natychmiastowego wychwytywania sytuacji awaryjnych.

Istnieje duża gama rozwiązań wizualizacji procesu przez zastosowanie chociażby rozwiązań firmowych, takich jak programy P-1200, P-3200 francuskiej firmy CEGELEC, czy też wizualizację procesu na kilku stanowiskach za pośrednictwem sieci komputerowej.

Często jednak dla małych czy też średnich systemów stosowanie powyższych rozwiązań nie jest uzasadnione ekonomicznie. Stąd też sterowniki klasy C-50, C-100, C-170 również firmy CEGELEC mogą być wyposażone w moduły pozwalające na komunikację sterownika ze światem zewnętrznym za pomocą komputera, terminala czy drukarki (rys.1).

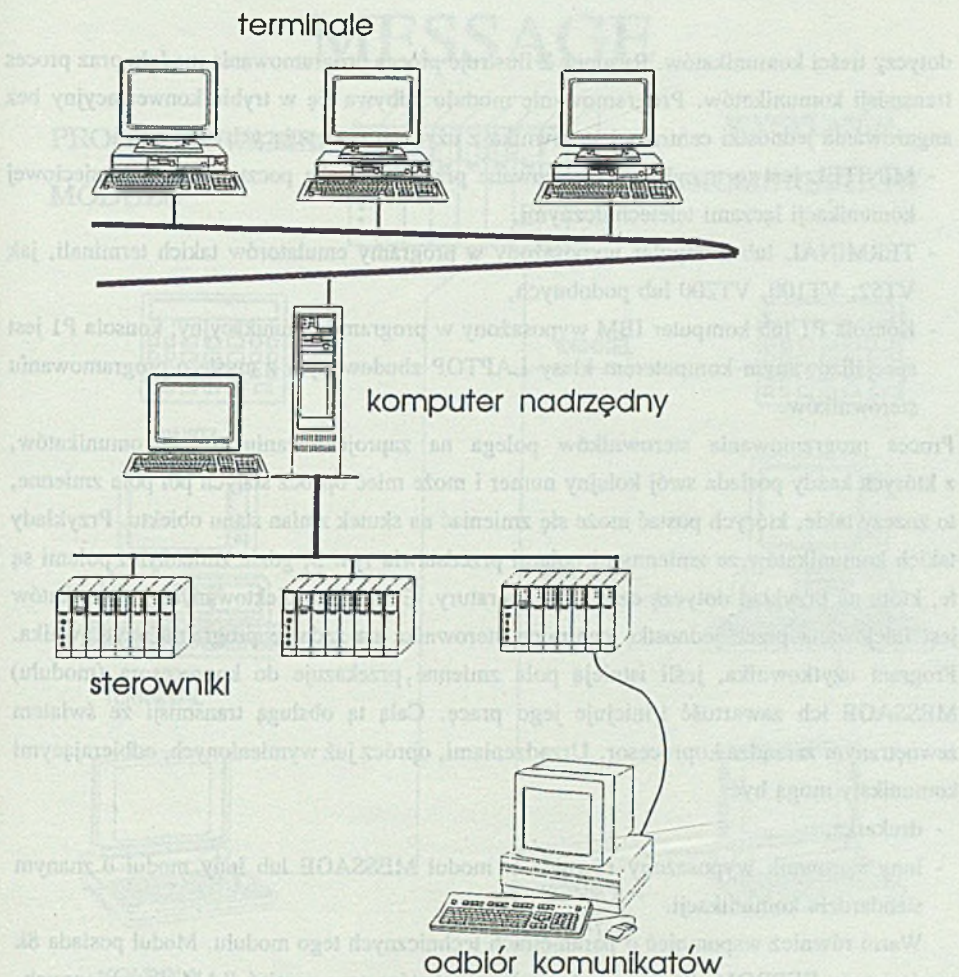
Pozwala to na zminimalizowanie kosztów systemu z jednoczesnym zapewnieniem realizacji pełnego informowania obsługi o stanie procesu, awariach czy nawet archiwizacji danych.

Dwa podstawowe takie moduły, których możliwości tu opisano, to tzw. moduł MESSAGE i moduł BASIC.

## 2. Moduł komunikacyjny typu MESSAGE

Nadaje się do informowania o procesach małych, w których liczba zmiennych decydujących o stanie procesu i niezbędnych dla operatora jest stosunkowo niewielka (rzędu kilkudziesięciu zmiennych).





Rys.1. Wizualizacja procesu

Fig.1. Control and monitoring station

Na rysunku przedstawiono typową strukturę przemysłowego systemu sterowania i nadzoru składającego się z sieci sterowników przemysłowych oraz komputera nadrzędnego, który umożliwia przyłączenie się do innych sieci lokalnych wyższego poziomu lub do sieci otwartych. W systemie tym moduł MESSAGE pełni rolę lokalnego urządzenia wizualizującego o ograniczonych możliwościach.

Moduł MESSAGE służy do komunikacji z użytkownikiem. W szczególności może wysyłać komunikaty opisujące aktualny stan sterownika, sygnalizować alarmy lub wizualizować pamięć sterownika. Moduł MESSAGE jest programowalny. Programowanie



dotyczy treści komunikatów. Rysunek 2 ilustruje proces programowania modułu oraz proces transmisji komunikatów. Programowanie modułu odbywa się w trybie konwersacyjny bez angażowania jednostki centralnej sterownika z użyciem następujących narzędzi:

- MINITEL jest to urządzenie opracowane przez francuską pocztę służące do sieciowej komunikacji łączami teletechnicznymi,
- TERMINAL lub komputer wyposażony w programy emulatorów takich terminali, jak VT52, VT100, VT200 lub podobnych,
- Konsola P1 lub komputer IBM wyposażony w program komunikacyjny; konsola P1 jest specjalizowanym komputerem klasy LAPTOP zbudowanym z myślą o programowaniu sterowników.

Proces programowania sterowników polega na zaprojektowaniu treści komunikatów, z których każdy posiada swój kolejny numer i może mieć oprócz stałych pól pola zmienne, to znaczy takie, których postać może się zmieniać na skutek zmian stanu obiektu. Przykłady takich komunikatów ze zmiennymi polami przedstawia rys. 3, gdzie zmiennymi polami są te, które na przykład dotyczą czasu i temperatury. Emisja zaprojektowanych komunikatów jest inicjowana przez jednostkę centralną sterownika na żądanie programu użytkownika. Program użytkownika, jeśli istnieją pola zmienne, przekazuje do koprocatora (modułu) MESSAGE ich zawartość i inicjuje jego pracę. Całą tą obsługą transmisji ze światem zewnętrznym zarządza koprocator. Urządzeniami, oprócz już wymienionych, odbierającymi komunikaty mogą być:

- drukarka,
- inny sterownik wyposażony również w moduł MESSAGE lub inny moduł o znanym standardzie komunikacji.

Warto również wspomnieć o parametrach technicznych tego modułu. Moduł posiada 8k pamięci typu EEPROM dla zapamiętania komunikatów oraz pamięć RAM dla zmiennych. Istnieje również pamięć typu stos, w której gromadzone są numery komunikatów do wysłania, podczas gdy moduł realizuje już transmisję. Szybkość transmisji jest programowalna i może wynosić od 75 do 19200 bodów.

Moduł wyposażony jest w 9-stykowe złącze programujące oraz w 25-stykowe złącze służące do wysyłania komunikatów.

W górnej części modułu znajdują się dwa zestawy mikroprzełączników, za pomocą których ustawia się parametry transmisji i tryby pracy urządzenia.

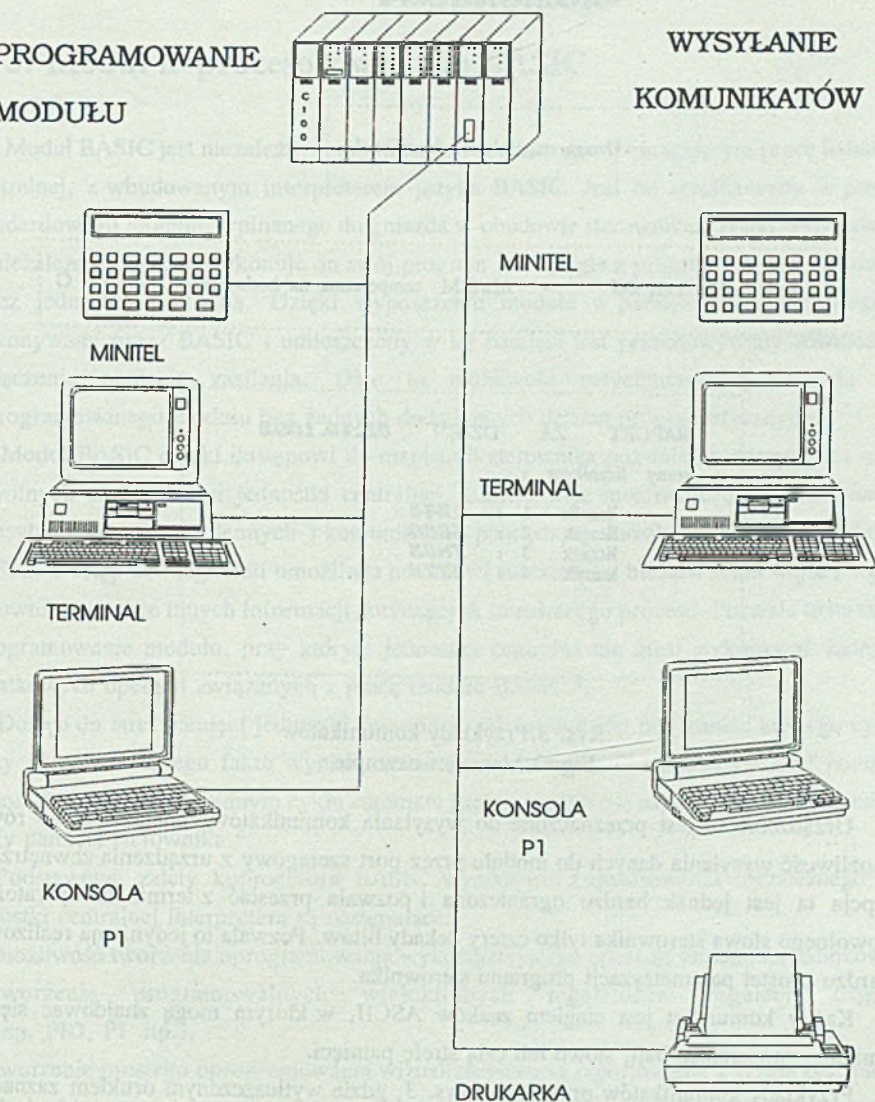
Moduł przez złącze 25-stykowe może realizować jeden z następujących protokołów transmisji:

- RS 232C lub pętla prądowa 20 mA
- RS 422; RS 485.

# MESSAGE

PROGRAMOWANIE  
MODUŁU

WYSYŁANIE  
KOMUNIKATÓW



Rys. 2. Moduł wysyłania komunikatów  
Fig. 2. Message transmission module



# KOMUNIKATY

Uwaga napelnianie zbiornika I

godz. **15** min **25** - ALARM temperatura na kotle wynosi **750°** C

RAPORT ZA DZIEŃ **02.10.1992**

stany licznikow :

|         |     |             |
|---------|-----|-------------|
| licznik | 1 : | <b>1244</b> |
| licznik | 2 : | <b>5666</b> |
| licznik | 3 : | <b>7865</b> |
| licznik | 4 : | <b>6778</b> |

Rys. 3. Przykłady komunikatów  
Fig. 3. Messages example

Urządzenie to jest przeznaczone do wysyłania komunikatów, chociaż istnieje również możliwość wysyłania danych do modułu przez port szeregowy z urządzenia zewnętrznego. Opcja ta jest jednak bardzo ograniczona i pozwala przesłać z terminala operatora do dowolnego słowa sterownika tylko cztery dekady bitów. Pozwala to jedynie na realizowanie bardzo prostej parametryzacji programu sterownika.

Każdy komunikat jest ciągiem znaków ASCII, w którym mogą znajdować się pola zmienne zawierające bajt, słowo lub całą strefę pamięci.

Przykłady komunikatów przedstawia rys. 3, gdzie wytłuszczonym drukiem zaznaczono pola zmiennych. Moduł pozwala na zaprogramowanie 100 komunikatów, z których każdy może zawierać 10 pól zmiennych i składać się z kilku linii tekstu.

Program jednostki centralnej sterownika obsługuje proces i w odpowiednich jego fazach daje polecenie koprocessorowi modułu MESSAGE o wysłaniu w świat komunikatu o odpowiednim numerze. Sama więc procedura wysłania komunikatu zajmuje w jednostce centralnej dwie linijki programu sterowania, a za całość związaną z samą transmisją

odpowiada koprocessor. Takie rozwiązanie nie wydłuża za bardzo cyklu pracy automatu i nie zajmuje dużo pamięci programu w samej jednostce centralnej sterownika.

### 3. Moduł koprocessorowy typu BASIC

Moduł BASIC jest niezależnym modulem koprocessora, wspomagającym pracę jednostki centralnej, z wbudowanym interpreterem języka BASIC. Jest on zrealizowany w postaci standardowego modułu, wpinanego do gniazda w obudowie sterownika. Dzięki wyposażeniu w niezależny procesor wykonuje on swój program równolegle z programem wykonywanym przez jednostkę centralną. Dzięki wyposażeniu modułu w pamięć EEPROM program wykonywany przez BASIC i umieszczony w tej pamięci jest przechowywany również po wyłączeniu napięcia zasilania. Daje to możliwość natychmiastowego użycia raz zaprogramowanego modułu bez żadnych dodatkowych działań programistycznych.

Moduł BASIC dzięki dostępowi do magistrali sterownika pozwala na odczyt jak i zapis dowolnych stref pamięci jednostki centralnej. Istnieje więc możliwość dwukierunkowego przesyłania danych, zmiennych i komunikatów pomiędzy jednostką centralną a modulem BASIC. Dostęp do magistrali umożliwia modułowi śledzenie na bieżąco stanu wejść i wyjść sterownika, a także innych informacji dotyczących sterowanego procesu. Pozwala to na takie oprogramowanie modułu, przy którym jednostka centralna nie musi wykonywać żadnych dodatkowych operacji związanych z pracą modułu BASIC.<sup>1</sup>

Dostęp do stref pamięci jednostki centralnej realizowany jest pod koniec każdego cyklu pracy automatu. Z tego faktu wynika ograniczenie na szybkość dostępu do stref pamięci jednostki centralnej. W jednym cyklu automatu istnieje możliwość zapisu bądź odczytu jednej strefy pamięci sterownika.<sup>2</sup>

Podstawowe zalety koprocessora BASIC wynikające z zastosowania niezależnego od jednostki centralnej interpretera są następujące:

- możliwość tworzenia oprogramowania wykorzystującego operacje zmiennoprzecinkowe,
- tworzenie programowalnych wielokrotnych regulatorów regulacji ciągłej (np. PID, PI itp.),
- tworzenie prostego oprogramowania wizualizacyjnego i raportowania z wykorzystaniem drukarki,

---

<sup>1</sup> z wyjątkiem inicjalizacji modułu.

<sup>2</sup> W standardowych zastosowaniach sterownika czas ten nie powinien przekraczać 8 ms dla sterownika C50 i 10 ms dla sterownika C100 dla programu zawierającego 1000 instrukcji.



- przyłączenie do sieci komputerowej i realizację obsługi łącza z poziomu języka BASIC,
- przyłączanie urządzeń o znanym choć niestandardowym protokole komunikacyjnym, gdyż koprocesor BASIC daje możliwość programowego sterowania takimi sygnałami, jak CTS, RTS itd.,
- organizację wymiany informacji ze sterownika (grupy sterowników) z komputerem nadrzędnym, gdy nie jest uzasadnione zastosowanie sieci przemysłowej,
- wizualizację algorytmów sterowania na pojedynczym sterowniku.

Na rysunku 5 przedstawiono możliwe do podłączenia urządzenia zewnętrzne, gdy koprocesor pracuje w trybie RUN (wykonywanie programu), natomiast na rysunku 6 - przedstawiono grupę urządzeń zewnętrznych, za pomocą których można prowadzić proces programowania. Są to urządzenia takie same jak w przypadku modułu typu MESSAGE, tak więc nie wymagają dodatkowego komentarza.

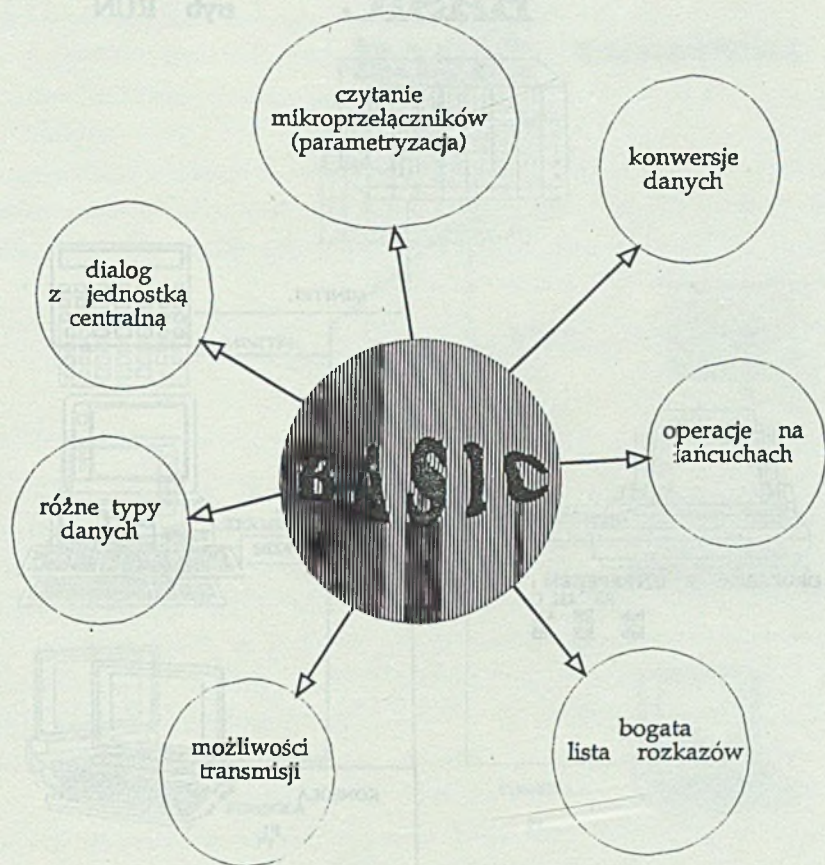
Warto również wspomnieć o parametrach technicznych modułu typu BASIC. Moduł zawiera specjalizowany procesor 8-bitowy, pamięć RAM, pamięć EEPROM, dwa interfejsy łącza szeregowego oraz układy współpracy z magistralą sterownika. Całość została zrealizowana w technologii CMOS, z izolacją galwaniczną do 2500V. Przeznaczenie pamięci jest następujące:

- pamięć RAM służy do przechowywania zmiennych programu modułu BASIC oraz umożliwia testowanie i uruchamianie programów w trybie interakcyjnym. Wielkość pamięci RAM modułu BASIC wynosi odpowiednio 22 Kb dla sterownika C100 oraz 12kB dla sterownika C50.

- pamięć EEPROM jest pamięcią zapisywalną przechowującą swoją zawartość także po wyłączeniu napięcia zasilania. Służy ona do przechowywania gotowego programu pracy modułu. Do pamięci EEPROM można przepisać uruchomiony i przetestowany program z pamięci RAM. Od tej pory istnieje możliwość wykonywania zapisanego w niej programu automatycznie po załączeniu napięcia zasilania sterownika. Istnieje także możliwość przepisania zawartości pamięci EEPROM do pamięci RAM. Pojemność pamięci EEPROM wynosi odpowiednio 16 Kb dla sterownika C100 i 6 Kb dla sterownika C50. Producent gwarantuje 10,000 operacji zapisu pamięci EEPROM.

Interfejs łącza szeregowego może pracować w jednym z czterech standardów transmisji zależnym od wersji modułu: w standardzie RS 232C -wersja X1, w standardzie RS 422 wersja X2, w standardzie RS 485 wersja X3 oraz w standardzie pętli prądowej - wersja X5.





Rys. 4. Funkcje i cechy modułu Basic

Fig. 4. Functions of Basic module

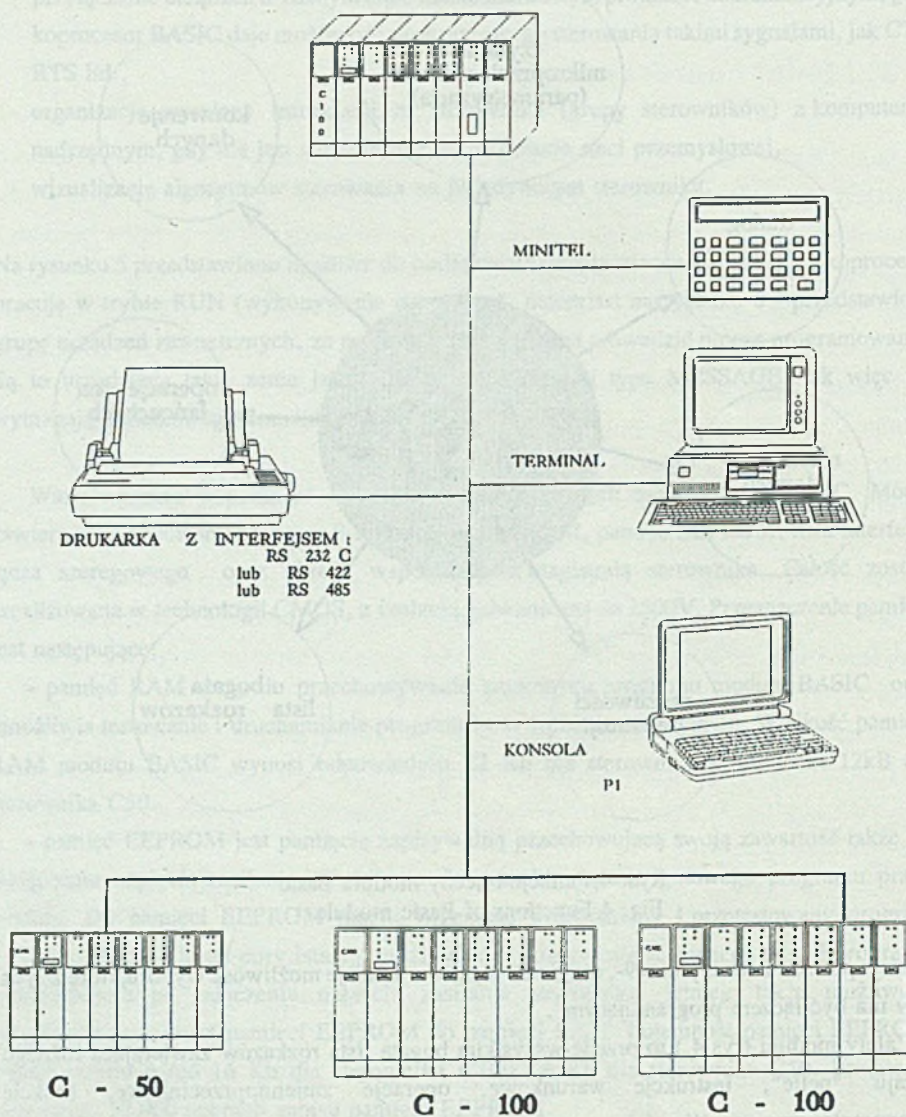
Interfejs posiada 2 łąca jedno 9-, drugie 25-stykowe. Istnieje możliwość wyboru, które z tych łączy ma być łącem programującym<sup>3</sup>.

Zalety modułu (rys 4.) to przede wszystkim bogata lista rozkazów zawierająca różnego rodzaju "pętle", instrukcje warunkowe, operacje zmiennoprzecinkowe, funkcje trygonometryczne, możliwości transmisji i odbioru danych, operacje na tekście i wiele innych.

<sup>3</sup> Wybór dokonywany jest przez odpowiednie ustawienie mikroprzełączników.

# BASIC

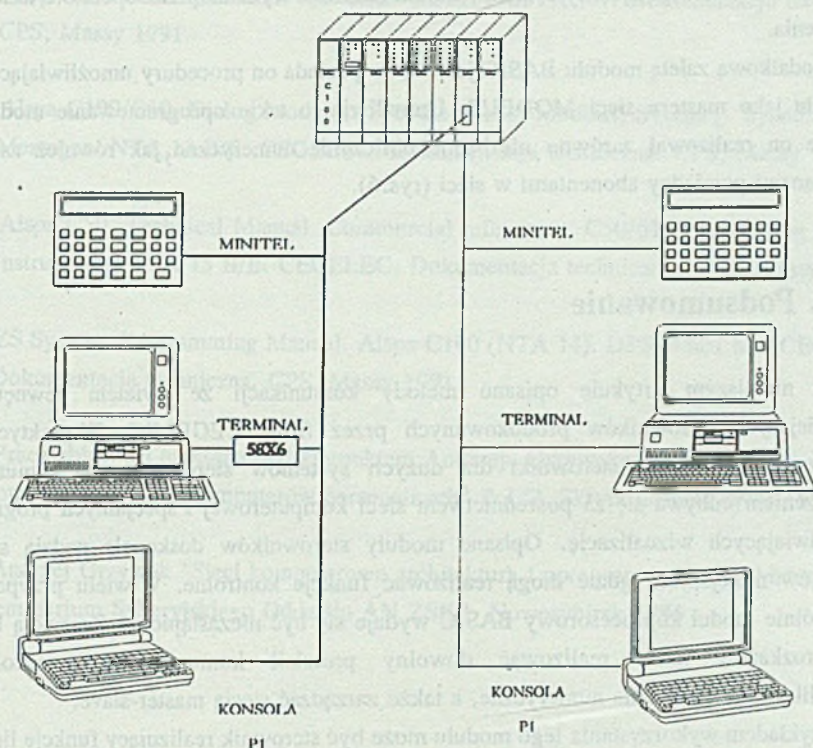
tryb RUN



Rys. 5. Tryb RUN  
Fig. 5. RUN mode



## BASIC PROGRAMOWANIE



Rys. 6. Programowanie modułu  
Fig. 6. Programming mode

Moduł BASIC może pracować w dwóch trybach: w trybie pracy ciągłej (RUN) i trybie pracy programowalnej (PROGRAMING) (rys 5. i rys. 6). Zmiana trybu pracy odbywa się za pomocą mikroprzełącznika.

W trybie RUN moduł wykonuje program zapisany w pamięci EEPROM, zostaje uruchomiony automatycznie w momencie startu modułu. W trybie tym pamięć RAM wykorzystywana jest jedynie do przechowywania wartości zmiennych wykorzystywanych przez program.

W trybie PROGRAMING po starcie modułu wysyłana jest na konsolę informacja identyfikująca moduł BASIC, a następnie czeka on na komendę wydaną przez operatora. Pamięć EEPROM nie jest wykorzystywana, a pamięć RAM w momencie startu jest

zerowana. Operator ma możliwość wpisania nowego programu, uruchomienia go, przepisania do pamięci EEPROM, jak również przepisania zawartości pamięci EEPROM do pamięci RAM. Po wykonaniu żądanej komendy moduł oczekuje wydania przez operatora następnego polecenia.

Dodatkową zaletą modułu BASIC jest to, że posiada on procedury umożliwiające pracę modułu jako mastera sieci MODBUS. Umożliwia to takie oprogramowanie modułu, że będzie on realizował zarówno niezbędne obliczenia numeryczne, jak również zarządzał wymianami pomiędzy abonentami w sieci (rys.6).

## 4. Podsumowanie

W niniejszym artykule opisano metody komunikacji ze światem zewnętrznym najmniejszych sterowników produkowanych przez firmę CEGELEC. W praktyce gdy stosowane są większe sterowniki dla dużych systemów sterowania, to komunikacja z otoczeniem odbywa się za pośrednictwem sieci komputerowej i specjalnych programów umożliwiających wizualizację. Opisane moduły sterowników doskonale nadają się dla systemów mniejszych, gdzie mogą realizować funkcje kontrolne. W wielu przypadkach szczególnie moduł koprocesorowy BASIC wydaje się być niezastąpiony. Ze swoją bogatą listą rozkazów może realizować dowolny protokół komunikacyjny, wykonywać skomplikowane obliczenia numeryczne, a także zarządzać siecią master-slave.

Przykładem wykorzystania tego modułu może być sterownik realizujący funkcje licznika ciepła pary przegrzanej dla kilku punktów pomiarowych. W sterowniku tym moduł BASIC może na podstawie pomiarów temperatury, ciśnienia i przepływu obliczać energię cieplną, ostrzegać o przekroczonych parametrach, a także sporządzać raporty, np. co godzinę czy za cały miesiąc pracy. Można sobie wyobrazić również sytuację, że sterownik będzie połączony z terminalem, za pomocą którego obsługa będzie nastawiała jakieś parametry sterowania, czy też obserwowała aktualne wartości mierzone.

Takich lub podobnych zastosowań sterowników wyposażonych w moduły koprocesorowe są w przemyśle tysiące, a wykorzystanie tych modułów zależy tylko od wyobraźni projektantów i użytkowników.



## LITERATURA

- [1] Alspa ZS. Communications networks. CGEE ALSTHOM. Dokumentacja techniczna. CPS, Massy 1991.
- [2] Alspa C100/C50. Dialog Modules. Sycodis N-bus. Modbus, Syretele. Sycodis N10. Messages. NTA 13 B/E. CEGELEC. Dokumentacja techniczna. CPS, Massy 1991.
- [3] Alspa C50. Technical Manual. Commercial references. C50/6120 Processing Unit Instructions. NTA 15 B/E. CEGELEC. Dokumentacja techniczna. CPS, Massy 1991.
- [4] ZS System. Programming Manual. Alspa C100 (NTA 14). DPS 50601 bA. CEGELEC. Dokumentacja techniczna. CPS, Massy 1991.
- [5] Praca zbiorowa napisana pod kierunkiem Andrzeja Grzywaka "Lokalne sieci komputerowe mikrokomputerów personalnych". Pol.Śl., Skrypt uczelniany nr 1682
- [6] Andrzej Grzywak "Sieci komputerowe architektura i oprogramowanie", Materiały Seminarium Syberyjskiego Oddziału AN ZSRR, Nowosybirsk 1986

Wpłynęło do Redakcji 19 lutego 1993 r.

## Abstract

Programmable logic controllers haven't any equipment for process visualizations. Monitoring big processes in the various industrial fields is impossible without special tools like programs P-1200 or P-3200. Sometimes simply communications are needed. In these cases network communications is not good solution, because it is too expensive for informations about several variables (fig.1). This paper describes two coprocessing modules for C-50/C-100 programmable controllers. These modules can be used to transmission the

simplest messages, archivisation of data or visualization state of simple processes. With the modules we can connect PLC to any system compatible with a serial link, for example: connect PLC with computer or printer is possible too (fig.2). The first module is MESSAGE module. They can be used to transmitting and/or receiving ASCII characters (fig.3). This module has three types of dialogue interfaces:

- RS 232C or current loop
- RS 422
- RS 485

It is possible one hundred messages to use. Especially it can send messages which describe current state of PLC, signal alarms or visualize memory of processing unit. It can be programmed by console P1 or terminal. Messages sent by module can be shown on console, terminal or printed on the printer with serial link.

The second one is BASIC module (fig.4). It is independent module of coprocessor helping work at central unit. They have their own memory which is programmed in Basic language and can be adapted to any nonstandard exchange protocol. The module incorporates the interpreter and text editor. In addition, they can perform complex calculations. It is possible to read the processing unit's data memory by exchanges with the automation program (fig.5 and fig.6).