

Awid Abd Kidham HANZAH

URZĄDZENIE TESTUJĄCE STEROWNIKA INTELSTER COMPACT

Streszczenie. W pracy zawarto opis prototypowej konstrukcji urządzenia testującego dla sterowników typu Intelster Compact. Urządzenie wyposażone zostało w układy pracy krokowej oraz pułapki adresowej niezbędne dla wykrycia błędów realizacji programu sterownika. Dzięki układom wyświetlającym kod i adres aktualnie wykonywanego rozkazu możliwe jest pełne monitorowanie pracy sterownika zarówno podczas prac uruchomieniowych jak i w rzeczywistych warunkach eksploatacji.

Dodatkową zaletą opisanego urządzenia jest jego niski koszt wynikający z odpowiedniego doboru elementów (także pod kątem dostępności na polskim rynku) oraz rezygnacji z autonomicznego programatora, w jaki wyposażone są podobne urządzenia produkowane przez firmy zachodnie (np. PC 4 K).

1. WSTĘP

Sterowniki przemysłowe, które obsługują procesy technologiczne za pomocą binarnych sygnałów sterujących (zarówno wejściowych, jak i wyjściowych) można rozważać jako klasyczne automaty skończone o charakterystycznym dla sterowanego procesu grafie przejść. W wielu przypadkach graf ten nie jest uzależniony od warunków pracy urządzenia w związku z czym nie zachodzi potrzeba jego modyfikacji podczas pracy sterownika, ani nawet przy adlustracji wstępnej. W takich sytuacjach ekonomicznie uzasadnione jest konstruowanie sterowników *sztywno programowanych*, a więc takich, w których graf przejść określony jest przez strukturę układu elektronicznego czy przekaźnikowego.

Nieco bardziej zaawansowaną klasę stanowią sterowniki *swobodnie programowane* (ang. *Programmed Controllers*), w których algorytm postępowania przechowywany jest w pamięci sterownika. Rozwiązanie to umożliwia wielkoseryjną produkcję standardowego modułu bez konieczności wprowadzania zmian konstrukcyjnych związanych ze specyfiką poszczególnych zastosowań.

System Intelster Compact [1,2] należy do drugiej klasy urządzeń sterujących. Jego cechą charakterystyczną jest podział wszystkich funkcji

sterownika pomiędzy dwa wyspecjalizowane moduły połączone elastyczną magistralą. Moduł A zawierający jednostkę centralną oraz pamięć programu odpowiedzialny jest za właściwą realizację logicznych funkcji sterownika. Moduł B zawierający układy wejściowe i wyjściowe oraz stopnie czasowe spełnia rolę sprzężenia z obiektem sterowanym. Ponieważ działanie urządzenia testującego dotyczyć będzie jedynie modułu A¹⁾, opis techniczny modułu B zostanie pominięty.

1.1. Opis modułu A

a) budowa

Elementy modułu rozmieszczone zostały na dwustronnej płytce drukowanej o wymiarach 360x240. Moduł wyposażony jest w dwa złącza pośrednie A i B oraz trzy złącza szufladowe C, W i O. Złącza te obsługują odpowiednio połączenia z:

- A - magistralą sterownika,
- B - jednostką BCD,
- C - urządzeniem testującym - programującym,
- W - szyną zasilacza.

Złącze O umożliwia wyprowadzenie sygnałów wyjściowych, trzy spośród których wyprowadzone zostały dodatkowo w postaci diod świecących umieszczonych na płytce modułu. Sygnalizują one:

- obecność napięć zasilających,
- realizację programu,
- obecność baterii (buforowanie zasilacza).

Oprócz jednostki centralnej moduł zawiera pamięć, układy kontrolne oraz pewne elementy części peryferyjnej sterownika. Całość połączona jest z ekranem izolującym układy zewnętrzne od wpływu jednostki centralnej.

b) działanie

Działanie jednostki centralnej sprowadza się do przetwarzania instrukcji zawartych w pamięci programu. Przetwarzanie każdej instrukcji trwa 2 μ s i odbywa się w trakcie cyklu podstawowego generatora (rys. 1).

Lista rozkazów (instrukcji) jednostki centralnej zawiera 18 pozycji, które można ująć w następujące grupy funkcjonalne:

¹⁾ Ze względu na konstrukcyjną zależność od charakterystycznych cech danego obiektu moduł B jest trudno testowalny za pomocą urządzeń uniwersalnych. Z drugiej strony prostota funkcji oraz brak mechanizmów sekwencyjnych umożliwiają łatwe wykrycie i lokalizację uszkodzenia w tym modelu bez konieczności stosowania specjalizowanych urządzeń testujących.

- rozkazy testujące,
- rozkazy wykonawcze,
- rozkazy organizacyjne,
- rozkazy pomocnicze.

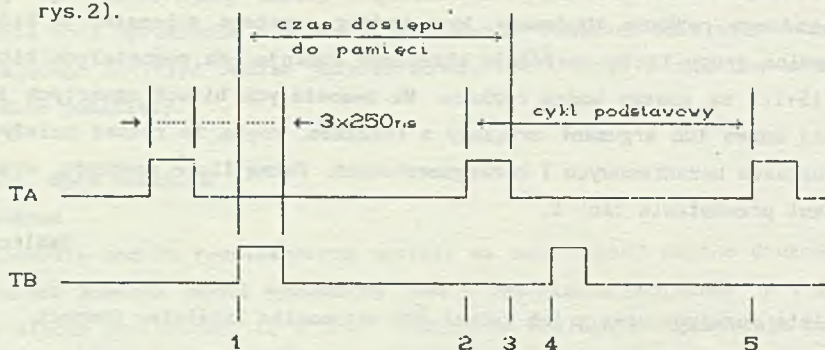
Kod każdego rozkazu zbudowany jest według prostego schematu, w którym poszczególne grupy bitów spełniają określone funkcje. Na pozostałych bitach słowa (15+12) są zawsze kodem rozkazu. Na pozostałych bitach zawarty jest zazwyczaj adres lub argument związany z rozkazem, chyba że rozkaz należy do grupy rozkazów bezadresowych i bezargumentowych. Pełną listę rozkazów, wraz z ich kodami przedstawia tab. 1.

Tablica 1

Lista rozkazów wraz z ich kodami dla sterownika Intelster Compact

Rozkaz	15 13 11 14 12 10 9 8 7 6 5 4 3 2 1 0	Typ rozkazu	Grupa rozkazów					
TH E TH TL E TL	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">1 1 0 1 X 0 0</td> <td rowspan="4" style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 10px; text-align: center;">← adres we/wy →</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">1 1 0 1 X 0 1</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">1 1 1 0 X 0 0</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">1 1 1 0 X 0 1</td> </tr> </table>	1 1 0 1 X 0 0	← adres we/wy →	1 1 0 1 X 0 1	1 1 1 0 X 0 0	1 1 1 0 X 0 1	t. wejść t. wyjść t. wejść t. wyjść	testujące
1 1 0 1 X 0 0	← adres we/wy →							
1 1 0 1 X 0 1								
1 1 1 0 X 0 0								
1 1 1 0 X 0 1								
S S J S N	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">1 0 1 1 X 0</td> <td rowspan="3" style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 10px; text-align: center;">← adres wyjścia →</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">1 0 1 0 X 0</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">1 0 0 1 X 0</td> </tr> </table>	1 0 1 1 X 0	← adres wyjścia →	1 0 1 0 X 0	1 0 0 1 X 0	włączanie	wyko- nawcze	
1 0 1 1 X 0	← adres wyjścia →							
1 0 1 0 X 0								
1 0 0 1 X 0								
R R J R N	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">0 0 1 1 X 0</td> <td rowspan="3" style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 10px; text-align: center;">← adres skoku →</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">0 0 1 0 X 0</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">0 0 0 1 X 0</td> </tr> </table>	0 0 1 1 X 0	← adres skoku →	0 0 1 0 X 0	0 0 0 1 X 0	włączanie		
0 0 1 1 X 0	← adres skoku →							
0 0 1 0 X 0								
0 0 0 1 X 0								
SP. SP J SP N	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">0 1 1 1</td> <td rowspan="3" style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 10px; text-align: center;">← adres skoku →</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">0 1 1 0</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">0 1 0 1</td> </tr> </table>	0 1 1 1	← adres skoku →	0 1 1 0	0 1 0 1	skoku	organi- zacyjne	
0 1 1 1	← adres skoku →							
0 1 1 0								
0 1 0 1								
LR DR	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">0 0 0 0 - - 1 1</td> <td rowspan="2" style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 10px; text-align: center;">← liczba →</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">0 0 0 0 - - 1 0 - - - - - - - - - -</td> </tr> </table>	0 0 0 0 - - 1 1	← liczba →	0 0 0 0 - - 1 0 - - - - - - - - - -	ładowanie dekrement			
0 0 0 0 - - 1 1	← liczba →							
0 0 0 0 - - 1 0 - - - - - - - - - -								
ZWB ZWE	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">0 0 0 0 0 0 0 0 1 0 - - - - - - - - - -</td> <td rowspan="2" style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 10px; text-align: center;">← liczba →</td> </tr> <tr> <td style="border-right: 1px solid black; padding-right: 5px;">0 0 0 0 0 0 0 0 1 1 - - - - - - - - - -</td> </tr> </table>	0 0 0 0 0 0 0 0 1 0 - - - - - - - - - -	← liczba →	0 0 0 0 0 0 0 0 1 1 - - - - - - - - - -	progr. czasu	pomoc- nicze		
0 0 0 0 0 0 0 0 1 0 - - - - - - - - - -	← liczba →							
0 0 0 0 0 0 0 0 1 1 - - - - - - - - - -								
NOP	1 1 1 1 - - - - - - - - - -	nic nie rób						

UWAGA: Symbol X użyty w tabeli oznacza bit, który, jeżeli jest ustawiony (=1), to adres efektywny wyznaczany jest względem zawartości rejestru indeksowego. Stan tego bitu wykrywany jest przez dekodery rozkazów i przekazywany jako niezależny sygnał X do układu wyświetlania (patrz rys.2).



- 1 - wystawienie adresu instrukcji n
- 2 - wykonanie instrukcji n+1
- 3 - pobranie instrukcji n
- 4 - wystawienie adresu instrukcji n+1
- 5 - wykonanie instrukcji n

Rys. 1. Wykres czasowy przedstawiający cykl Podstawowego Generatora

Fig. 1. Time diagram of the cycle in the Basce Generator

2. ZADANIE URZĄDZENIA TESTUJĄCEGO

Urządzenie testujące sterownika Intelster powinno umożliwiać jak najpełniejsze śledzenie realizacji wykonywanego programu. Informacje dostarczane przez to urządzenie powinny wystarczyć zarówno do wykrycia błędów w programie, jak i błędów wynikających z uszkodzenia układu sterownika. Spełnienie tych wymagań można uzyskać poprzez umożliwienie:

- przerwania i wznowienia wykonywania programu w dowolnej chwili podczas pracy ciągłej,
- pracy krokowej,
- zatrzymania programu na zadanym adresie (pułapka adresowa),
- podglądu kodu i adresu aktualnie wykonywanej instrukcji²⁾.

²⁾ W trakcie ciągłej pracy urządzenia podgląd ten jest oczywisty wyłączony. Wyświetlanie jest odblokowywane w momencie zatrzymania programu, a także podczas pracy krokowej.

W odróżnieniu od istniejących urządzeń testująco-programujących, które mogą być wykorzystane w serwisie sterownika Intelster (np. PC 4 K), proponowane rozwiązanie nie daje możliwości zmiany programu sterownika. Ograniczenie to nie stanowi wady przy założeniu, że urządzenie będzie wykorzystywane przy usuwaniu usterek sprzętowych w pracujących już sterownikach.

W celu wykorzystania urządzenia podczas prac uruchomieniowych konieczna jest jego współpraca z zewnętrznym programatorem. Dzięki rozdzieleniu funkcji testowania od programowania można do tego celu użyć dowolnego uniwersalnego programatora EPROM'ów. Takie rozwiązanie daje znaczne zmniejszenie kosztu urządzenia przy równoczesnym niewielkim zmniejszeniu jego walorów użytkowych.

3. KONSTRUKCJA URZĄDZENIA TESTUJĄCEGO

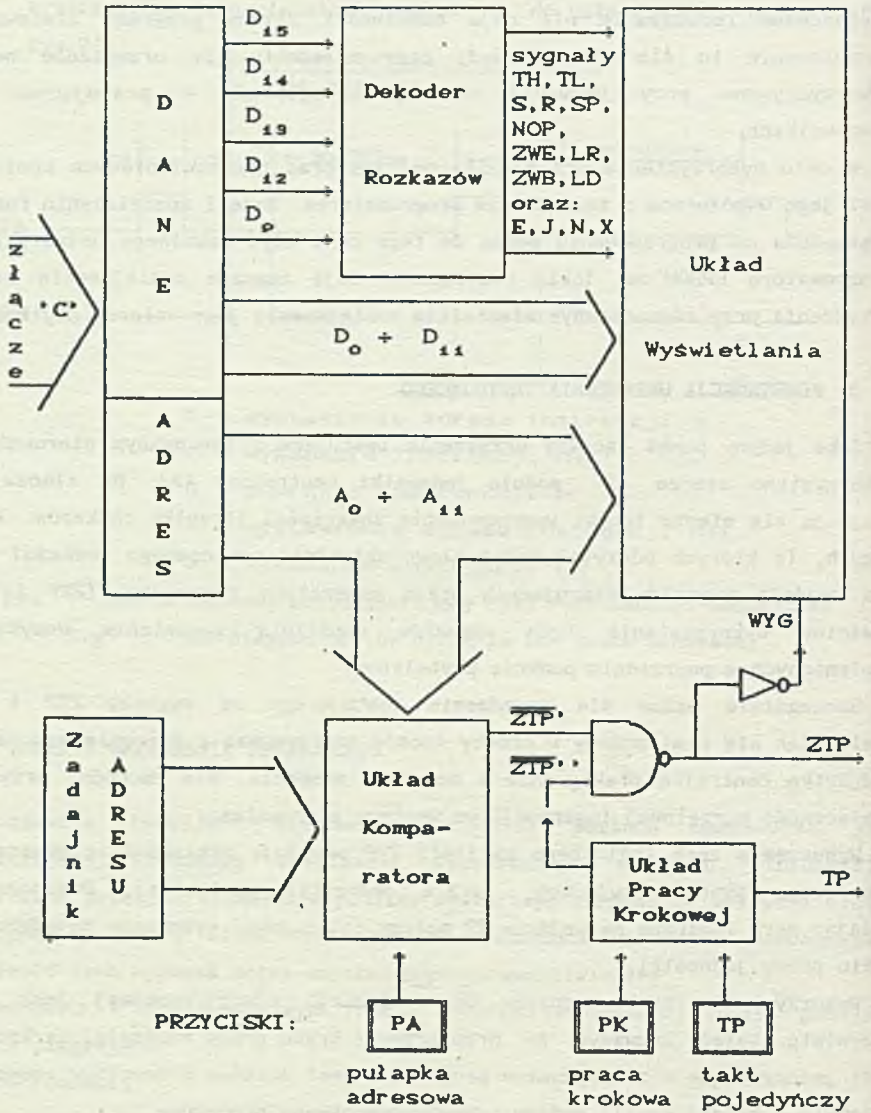
Jako jedyny punkt łączący urządzenie testujące z testowanym sterownikiem wykorzystano złącze C modułu jednostki centralnej (A). Na złączu tym znajdują się między innymi wyprowadzenia zawartości licznika rozkazów, linie danych, (z których odczytać można słowo aktualnie pobieranego rozkazu) oraz dwa wejścia sygnałów sterujących pracą generatora zegarowego (ZTP i TP). Właściwe wykorzystanie tych sygnałów umożliwi spełnienie wszystkich wymienionych w poprzednim punkcie postulatów.

Szczególnie ważne dla urządzenia testującego są sygnały ZTP i TP. Posługując się nimi możemy w prosty sposób zatrzymywać i ponownie uruchamiać jednostkę centralną praktycznie w dowolnym momencie. Nie zachodzi przy tym konieczność sprzętowej ingerencji we wnętrze sterownika.

Wymuszenie zera logicznego na linii ZTP powoduje zablokowanie generatora zegarowego synchronizującego pracę jednostki centralnej. Równocześnie podając zero logiczne na wejście TP możemy zainicjować wykonanie pojedynczego cyklu pracy jednostki.

Wykorzystanie tych sygnałów do realizacji pracy krokowej jest dość oczywiste. Należy zauważyć, że przełączenie trybu pracy z ciągłej na krokową jest jednoznaczne z zatrzymaniem programu i jest możliwe w dowolnym momencie. Podobnie w każdej chwili możemy wznowić wykonywaną procedurę.

W przypadku pułapki adresowej generacja sygnału ZTP musi nastąpić w momencie wykrycia zadanego adresu na szynie złącza C. Ponieważ operacja pobrania rozkazu odbywa się podczas fazy wykonania rozkazu poprzedniego (rys.1), przechwycony adres będzie adresem rozkazu następnego w stosunku do aktualnie wykonywanego, a więc program zostanie zatrzymany jeszcze przed wykonaniem wybranej instrukcji.



Rys. 2. Uproszczony schemat blokowy urządzenia testującego

Fig. 2. Simplified block diagram of the testing equipment

Osobnym problemem jest wyświetlenie informacji o pobranym rozkazie. Ponieważ śledzenie programu nie ma sensu podczas normalnej, ciągłej pracy jednostki, układ wyświetlania pozostaje wtedy zablokowany wewnętrznym sygnałem WYG sterownika. Odblokowanie następuje w momencie przełączenia systemu na pracę krokową, bądź też po zadziałaniu pułapki adresowej. Wyświetlana wówczas informacja obejmuje adres, kod i argument rozkazu pobranego.

Pewne trudności nastęrcza wyświetlenie adresu już pobranego rozkazu, ze względu bowiem na organizację cyklu rozkazowego, w momencie zatrzymania jednostki centralnej adres ten znika z magistrali i zostaje zastąpiony adresem następnego rozkazu. Ponieważ nie ma możliwości zatrzymania pracy jednostki centralnej w innym momencie, dla właściwego wyświetlenia adresu pobranego rozkazu należy posłużyć się dodatkowym rejestrem buforowym, do którego interesująca nas informacja będzie wpisywana z odpowiednim wyprzedzeniem.

4. SCHEMAT BLOKOWY URZĄDZENIA

Uproszczony schemat blokowy urządzenia testującego przedstawiony został na rys. 2. Aby w dowolnej chwili, po zatrzymaniu programu, dane związane z instrukcją pobieraną z pamięci w chwili przerwania były dostępne, stan magistrali danych oraz adresowej przepisywany jest do rejestru buforowego z każdym opadającym zboczem fazy TA zegara.

Wybrane bity słowa danych są następnie analizowane w dekoderze rozkazów, który ustala rodzaj wykonywanego rozkazu i przekazuje tę informację do układu wyświetlania. Konstrukcja dekodera oparta została na pamięci stałej (PROM 74S188) adresowanej bitami $D_{15,14,13,12,9}$ linii danych. Bity te wystarczają do jednoznacznego określenia jednej z ośmiu podstawowych grup rozkazów³⁾. Właściwa informacja wyprowadzana jest w postaci stanu wysokiego na jednym z ośmiu wyjść pamięci. Równolegle, w układach kombinacyjnych ustalane są informacje określenia analizowanej instrukcji⁴⁾.

3) TH, TL, S, R, SP, NOP, ZWB/ZWE, LR/DR.

4) Dodatkowe sygnały E, J, N, X dla pierwszych sześciu grup rozkazów oraz całkowite zdekodowanie rozkazów ZWE, ZWB, LR, DR.

Do układu wyświetlania, oprócz zdekodowanych sygnałów TH, TL, S, R, SP, NOP, ZWE, ZWB, LR, DR oraz sygnałów dodatkowych E, J, N, X doprowadzone są magistrala adresowa oraz młodsza część magistrali danych zawierająca argument rozkazu. Ważnym sygnałem wejściowym tego układu jest sygnał WYG, który blokuje wyświetlanie informacji podczas ciągłej pracy układu.

Osobny blok stanowi układ pułapki adresowej składający się z *zadajnika adresu* oraz *układu komparatora*. W momencie zrównania się adresów układ ten wysyła sygnał ZTP blokujący zegar jednostki centralnej. Jednocześnie generowany jest sygnał \overline{WYG} odblokowujący wyświetlanie.

Całość zrealizowana została przy wykorzystaniu minimalnej liczby prostych elementów logicznych, co przy trudnościach rynkowych może być istotną zaletą urządzenia. Niedostępność wyświetlaczy alfanumerycznych spowodowała konieczność zastąpienia ich zestawem diod luminescencyjnych oznaczających odpowiednie grupy rozkazów wypisane na płycie czołowej. Nie zmniejsza to jednak czytelności wyświetlanej informacji.

5. PODSUMOWANIE

Przedstawione urządzenie zostało zaprojektowane, wykonane i uruchomione w Instytucie Elektroniki Politechniki Śląskiej w Gliwicach. Wśród zalet urządzenia decydującego o jego przydatności należy wymienić prostotę obsługi, czytelność wyświetlanej informacji, a także łatwość naprawy i dostępność użytych elementów.

Urządzenie stanowi niejako uzupełnienie systemu Intelster Compact i jest nieodzowne szczególnie w fazie testowania nowych programów (podczas dostosowywania sterownika do nowych zastosowań). Niemniej przydatne może być wykorzystanie urządzenia w serwisie pracujących sterowników, gdzie wykorzystanie mechanizmu pracy krokowej może znacznie przyspieszyć lokalizację uszkodzenia w module jednostki centralnej.

LITERATURA

- [1] *Sterownik PC COMPACT. Przewodnik projektanta systemu*, praca zbiorowa.
- [2] *Zastosowanie programowalnego układu sterowania elektronicznego ciągami technologicznymi stacji przygotowania zakładu przeróbki węgla*, praca zbiorowa.

[3] J. Pieńkos, J. Turczyński, *Układy scalone TTL w systemach cyfrowych*, WKiŁ, Warszawa 1980.

Recenzent: Dr inż. Jerzy Rąbalski

Wpłynęło do Redakcji 1.09.1989 r.

УСТРОЙСТВО ТЕСТИРУЮЩЕЕ КОМАНДОАППАРАТ INTELSTER COMPACT

Р е з ю м е

В работе содержится описание прототипной конструкции тестирующего устройства для командоаппарата типа Intelster Compact. Устройство оснащено схемой шаговой работы, а также адресной ловушкой необходимой для обнаруживания ошибок при реализации программы командоаппарата. Благодаря системам визуально отображающим код и адрес своевременно выполненного приказа, является возможным полный контроль работы командсаппарата во время пусковых работ и в действительных условиях эксплуатации.

Дополнительным преимуществом описанного устройства является его низкая стоимость, вытекающая из соответственной подборки элементов (также с точки зрения доступности на польском рынке) и отказа от автономного программатора, каким оснащены подобные устройства выпускаемые западными фирмами (например PC 4K).

TESTING EQUIPMENT FOR THE INTELSTER COMPACT CONTROLLER

S u m m a r y

In the paper a prototype testing equipment for the Intelster. Compact controller is presented. The equipment is endowed with the stepwise work systems and an adress trap necessary to detect errors in the controller program execution. The display enables full monitoring of the controller activity in the real ezploition conditions by showing a code and address of an executed command.

The low cost resulting from the choice of elements (according to Polish market conditions) and omitting of an automatic programmer used in the west firm products (e.g. PC4K) is the additional advantage.