PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
MERA-PIAP
Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81

OŚRODEK POMIARÓW RUCHU I CZASU

0744

Główny wykonawca dr inż. Edward Golonka

Wykonawcy mgr inż. Lech Nowakowski, inż. Z. Bojar, mgr inż. M.Muter st. tech. J. Zduniak

Konsultant

Nr zlecenia 1020B

Zleceniodawca

Przeprowadzenie i uruchomienie automatycznego systemu dla technicznej-kontroli zboczy i osuwisk przy pomocy Polskiej Aparatury Strunowej oraz opracowanie i wykonanie aparatury dla automatyzacji kaskady Soły i Dunajca Zadanie 11.3.9 - Etap 2. "Opracowanie i wykonanie modeli modułu miernika strunowego opartego na mikroprocesorze współpracującego z: a/ mikrokomputerem IBM-PC b/ programowanym kalkulatorem"

Pracę rozpoczęto dnia 1988.12.01 **zakończono dnia** 1989.03.31

Kierownik pracowni
dr inż.E.Golonka

Z-ca Dyrektora d/s Pomiarów
dr inż. J. Winiecki

Kierownik Ośrodka

Praca zawiera:

stron	10
rysunków	7
fotografii	—
tabel	4
tablic	—
załączników	13

Nr rejestr. 6227

Rozdzielnik - ilość egz:

Egz. 1	BOINTE
Egz. 2	IMGW-KONSULTEX
Egz. 3	ORC PIAP
Egz. 4	IMGW-KONSULTEX
Egz. 5	IMGW-KONSULTEX
Egz. 6	PIAP-ORC

Analiza deskryptorowa APARATURA POMIAROWA, APARATURA KONTROLNO-POMIAROWA BUDOWLI WODNYCH, APARATURA STRUNOWA, AUTOMATYZACJA I KOMPUTERYZACJA POMIARÓW BUDOWLI WODNYCH.

Analiza dokumentacyjna APARATURA KONTROLNO-POMIAROWA OPARTA NA METODZIE STRUNOWEJ /TENSOMETRIA STRUNOWA/ PRZEZNACZONA DO ZDALNYCH I DŁUGOTRWAŁYCH POMIARÓW STANU BUDOWLI WODNYCH.

Tytuły poprzednich sprawozdań

Nr rej 6095 Zadanie CPBR 11.10.13.2 Etap 1 - Rozeznanie patentowe

Nr rej. 6107 Zadanie CPBR 11.10.09.2 - Etap 1 - Rozeznanie tematu badania patentowe, projekt wstępny założenia techniczno - ekonomiczne.

Nr rej. 6125 - Zadanie CPBR 11.10.13.2 - Badania laboratoryjne modeli czujników strunowych SCCwp-05.

624.953

621.317.795

Zbiorniki wodne. Zapory pomiarowe i kontrolne, ich cele i specyfika

UKD

PIAP 41/88 10000

SPIS TREŚCI

Załączniki:

8/ Rys. 2 Postać podstawowego meldunku informacyjnego wg standartu SDLC .

9/ Rys. 3 Schemat blokowy sterownika BITBUS f-my STER-PROJEKT.

10/ Rys. 4 Schemat blokowy modelu stacjonarnego strunowego miernika.

11/ Rys. 5 Schemat ideowy modułu stacjonarnego strunowego miernika

12/ Rys. 6 Schemat blokowy modułu strunowego miernika polowego

13/ Rys. 7 Schemat ideowy modelu strunowego miernika polowego

1. SPRAWY FORMALNE.

1.1. Przedmiot pracy.

Przedmiotem pracy w tym etapie było opracowanie koncepcji i wykonanie modeli modułów mierników strunowych, stacjonarnego i polowego opartych na mikroprocesorze.

1.2. Zamawiający.

Praca została zamówiona przez Przedsiębiorstwo Usług Konsultingowych KONSULTEX w W-wie i jest umieszczona w JPBR jako zadanie wdrożeniowe 10.3.9.

1.3. Podstawa wykonania pracy.

Umowa 413/88 z dnia 29.11.88 /zlec. 1020B/ zawarta między Przedsiębiorstwem Usług konsultingowych KONSULTEX w W-wie a Przemysłowym Instytutem Automatyki i Pomiarów w W-wie.

1.4. Zakres pracy.

Praca ta w etapie drugim obejmowała:

- opracowanie koncepcji modułów mierników strunowych- stacjonarnego i polowego,
- opracowanie elektronicznej dokumentacji szkicowej dla modeli
- wykonanie modeli /część elektryczna/ modułów mierników
- wstępne sprawdzenie i potwierdzenie słuszności opracowanej koncepcji systemu.

2. OPRACOWANIE I WYKONANIE MODELI MIERNIKA STRUNOWEGO DO ZAUTOMATYZOWANYCH SYSTEMÓW POMIAROWYCH. OPARTEGO NA MIKROPROCESORZE WSPÓŁPRACUJACEGO: a/ MIKROKOMPUTEREM IBM-PC

b/ KALKULATOREM PROGRAMOWANYM PSION-ORGANIZER II

2.1. Wstęp.

Przedmiotem niniejszej pracy, Etap 2, zadanie ŻPBR 11.3.9 było opracowanie koncepcji i wykonanie 2-ch modeli mierników strunowych, stacjonarnego i polowego opartych na mikroprocesorze. Podstawą opracowania koncepcji modułów i systemu pomiarowego były, Założenia techniczne i Projekt wstępny patrz sprawozdanie PIAP Nr rej. 6107 - opracowane w etapie 1.

Zgodnie z Założeniami Technicznymi i Projektem Wstępnym moduł miernika strunowego oparty na mikroprocesorze stanowi podstawową część systemu pomiarowego, gdzie centralnym urządzeniem sterującym w przypadku modułu stacjonarnego /8-mio lub 16 wejściowego/ jest mikrokomputer IBM-PC a w przypadku modułu miernika polowego /jednowejściowego/ - kalkulator programowany typu Psion-Organiser II.

Dla przypomnienia moduły posiadają następujące symbole:

a/ moduł strunowego miernika stacjonarnego - SMP-08÷16

b/ moduł strunowego miernika polowego - SMP-01.

2.2. Omówienie podstawowej konstrukcji systemu pomiarowego.

Schemat blokowy systemu przedstawiono na Rys.1.

System składa się z szeregu identycznych modułów pomiarowych połączonych z komputerem nadrzędnym za pośrednictwem szeregowej linii transmisyjnej. Ilość modułów podłączonych do systemu może wynosić 64. Każdy z modułów jest urządzeniem 8-mio lub 16-to wejściowym /ilość czujników podłączonych do modułu/. Cały system będzie mógł sterowac pomiarem do 512 lub 1024 czujników strunowych. Ze względu na to, że system pomiarowy będzie rozłożony na dużej przestrzeni przyjęto stosowanie szeregowo-bitowego sposobu przesyłania informacji między kompu-

terem nadrzędnym a modułami pomiarowymi. Zakłada się, że poszczególne moduły pomiarowe połączone będą między sobą oraz komputerem nadrzędnym za pomocą symetrycznej linii przesyłowej zrealizowanej w standarcie BITBUS, dopasowanej falowo na obu końcach. Fizycznie - linia będzie miała postać dwużyłowej pary skręconych przewodów w ekranie /tzn. skrętki w ekranie/.

Przyjęto, że przesłanie informacji w magistrali będzie zgodne z protokółem warstwy sterowania łącza wg standartu SDLC /Synchronous Data Link Centrol/ opracowanym przez firmę IBM. Jest to protokół bitowo zorientowany, definiujący, synchroniczny, szeregowy sposób przesyłania informacji. Podstawową strukturę meldunku takiego protokółu przedstawiono na Rys.2. Meldunek wysłany z komputera nadrzędnego do modułu pomiarowego z poleceniem wykonania pomiaru miałby długość 7 bajtów.

Meldunek wysłany z komputera nadrzędnego zawierający dane pomiarowe jest dłuższy i wynosi 13 bajtów.

Maksymalna szybkość transmisji wynosi 3 kbity /s.

Zastosowanie nowoczesnego protokółu SDLC zapewnia realizację szybkiej i bezbłędnej transmisji synchronicznej z możliwością synchronizacji dużych bloków informacji zawierających setki i tysiące bitów.

Dla sterowania transmisją i w celu dopasowania do magistrali, komputer IBM-PC musi być wyposażony w specjalny pakiet kontrolera transmisji w standarcie BITBUS, który dołączony będzie do szyny systemowej komputera.

Pakiet ten Rys.3 jest rozwiązaniem już opracowanym i produkowanym przez firmę STER-PROJEKT w W-wie. Również każdy z modułów

pomiarowych musi być wyposażony w odpowiedni układ dopasowujący do magistrali BITBUS.

Jak wiadomo moduł miernika strunowego będzie urządzeniem uniwersalnym tzn. będzie mógł pracować zarówno jako jeden z bloków systemu pomiarowego oraz niezależnie od systemu, jako miernik przenośny, przeznaczony do pomiarów polowych.

Ze względów ekonomicznych moduły będą wykonywane w dwóch wersjach:

- stacjonarnej o symbolu - SMP-08 ÷ 16
- polowej o symbolu - SMP-01

Wersje te różnią się między sobą tylko liczbą wejść czujników oraz sposobem zasilania.

2.2.1 Stacjonarny moduł miernika strunowego-model.

Opis koncepcji rozwiązania przedstawiony jest w załączniku Nr 1, schemat blokowy na Rys. 4, a schemat ideowy na Rys.5. Podstawowym blokiem modułu miernika jest układ analogowy.

Blok ten zawiera układ wybierania i załączania jednego z kanałów wejściowych, układ pobudzenia struny do drgań, układ wzmocnienia sygnału pomiarowego na ciąg impulsów prostokątnych. Z przetworzonego w ciąg impulsów sygnału z czujnika, na podstawie programu rezydującego w pamięci ROM układ mikroprocesorowy zlicza czas trwania 32 okresów drgań struny. Wynik pomiaru zostaje przypisany do bufora.

Po uzyskaniu z komputera nadrzędnego komunikatu o uzyskaniu dostępu do magistrali, za pośrednictwem układu dopasowującego moduł do magistrali BUTBUS wysyłany jest do komputera nadrzędnego komunikat zawierający adres modułu, adres mierzonego czujnika oraz wyniki pomiaru okresu.

Mikroprocesor zapewnia w pełni programową, dwustronną komunikację między modułami pomiarowymi a komputerem nadrzędnym oraz stero-

waniem i wybieraniem jednego z kanałów układu analogowego i cyklem pomiarowym czujnika.

Dla wykonania pomiaru pojedyńczego czujnika komputer nadrzędny wysyła komunikat zawierający w nagłówku adres wybranego modułu, kod rozkazu, który ma wykazać a w polu informacyjnym numer kanału modułu pomiarowego.

Moduł pomiarowy po zdekadowaniu adresu i po sprawdzeniu, że rozkaz przeznaczony jest dla niego przystępuje do wykonania cyklu pomiarowego.

W konstrukcji zastosowano nowoczesny układ mikroprocesorowy 80C31. Układ zawiera wbudowane 4 kB pamięci ROM oraz 128 bajtów RAM. Wyposażony jest ponadto w szeregowy port we-wy, co zapewnia możliwość wykonywania w pełni programowej obsługi łącza szeregowego wg standartu BIT-BUS.

Ponadto ma wbudowany układ dwóch 16-bitowych liczników timerów dzięki czemu możliwa jest łatwa realizacja funkcji pomiaru czasu trwania okresu drgań.

Układ 80C31 realizowany jest w technologii CMOS, co zapewnia znikomy pobór mocy. Istnieje łatwa możliwość rozszeżenia funkcji układu poprzez stosowanie specjalistycznych układów scalonych LSI.

W wersji stacjonarnej moduł jest zasilany poprzez zasilacz z sieci napięcia zmiennego 220 V/. W związku z tym będzie koniecznym dobranie i dopasowanie zasilania impulsowego ze stabilizacją napięcia o minimalnych gabarytach.

Moduł stacjonarny musi być montowany w szczelnej obudowie, z której będą wyprowadzone gniazda umożliwiające połączenie modułu w system pomiarowy gniazda wejściowe do podłączenia czujników strunowych oraz gniazda niezbędne do podłączenia zasilania. Zastosowanie w konstrukcji nowoczesnych układów scalonych wykonanych w technice CMOS zapewni niewielki pobór mocy przez urządzenie.

2.2.2 Polowy moduł miernika strunowego - model.

Schemat blokowy tego urządzenia przedstawiono na Rys.6, a schemat ideowy na Rys.7. Wykorzystany jest w nim omówiony już w poprzednim punkcie moduł miernika stacjonarnego.

Miernik w wersji polowej posiada jedno wejście dla czujnika strunowego, zasilany jest z baterii 9V i współpracuje z zasilanym również bateryjnie kalkulatorem programowanym PSION - Organiser II. Zadaniem kalkulatora jest programowe inicjowanie pomiaru przez moduł miernika oraz zbieranie danych pomiarowych, wykonywanie niezbędnych obliczeń oraz gromadzenie danych w jego pamięci. Dla współpracy z kalkulatorem moduł pomiarowy wyposażony będzie w układ interfejsu do łącza szeregowego w standarcie RS-232C. Szczegółowy opis kalkulatora zamieszczono w instrukcji załącznik Nr 2. Zaletą tego rozwiązania jest możliwość transmisji wszystkich zapisanych w pamięci RAM zbiorów za pośrednictwem łącza RS-232C do innych komputerów wyposażonych w to łącze w tym także do komputera IBM-PC.

Tak więc, po wykonaniu pomiarów dowolnej ilości czujników wyniki tych pomiarów mogą być przechowywane w pamięci kalkulatora, mogą być wprowadzone do komputera IBM-PC w celu dokonania ich szerszej analizy i obróbki.

Moduł pomiarowy i kalkulator są zasilane każdy w własnej baterii.

2.3. Realizacja techniczna części mikroprocesorowej modułów pomiarowych.

Moduł pomiarowy stacjonarny.

Część mikroprocesorową stacjonarnego modułu pomiarowego Rys.5 zrealizowano w oparciu o mikrokontroler Intel 80C31. Jako pamięć programu może być zastosowany układ 2732 o pojemności 4kB lub układ 2764 o pojemności 8 kB. Jest to pojemność znaczna, umożliwiająca realizację nawet skomplikowanych algorytmów pomiaru

i obróbki danych pomiarowych.

Ponadto moduł zawiera interfejs sieci transmisji danych pomiarowych. Realizacja sprzętowa umożliwia wariantowe zastosowanie układów transmisyjnych zachodnich Am26LS31 i Am26LS32 lub krajowych UCY75110 i UCY75107. Obsługa protokołu transmisji jest realizowana programowo przez mikrokomputer. W układzie tym zastosowano optoizolację pomiędzy częścią mikroprocesorową i liniową oraz przewidziano zasilanie części liniowej z oddzielnego toru zasilania. Umożliwia to pracę układu w warunkach utrudniających wyrównanie potencjałów mas pomiędzy poszczególnymi modułami pomiarowymi.

Interfejs do części analogowej modułu pomiarowego zawiera bufory 7407 umożliwiające dopasowanie poziomów sygnałów. Pomiar realizowany jest mikroprogramowo z wykorzystaniem zasobów mikrokontrolera /wbudowanego timera/.

Unikalny numer modułu w sieci ustawiony jest przy pomocy sześciopozycyjnego przełącznika odczytywanego przez mikroprocesor podczas restartu.

Moduł pomiarowy przenośny.

Część mikroprocesorową przenośnego modułu pomiarowego Rys.7, zrealizowano w oparciu o mikrokontroler Intel 80C31. Jako pamięć programu może być zastosowany układ 2732 o pojemności 4 kB lub układ 2764 o pojemności 8 kB.

Ponadto moduł zawiera szeregowy interfejs transmisji danych RS-232C służący do przesyłania danych pomiarowych do kalkulatora. Obsługiwany on jest z wykorzystaniem wewnętrznego portu transmisji szeregowej, jaki zawiera mikroprocesor 8031.

2.4 Wnioski i dalsze prace.

Wykonane i uruchomione modele obydwu modułów pomiarowych,

stacjonarnego i polowego miały na celu sprowadzenie idei opracowanego systemu a także sprawdzenie działania podstawowych układów systemu, jak np. wybierania i pobudzania do drgań strun w czujnikach i zliczenia tej częstotliwości i ich kodowanie w pamięci modułu, sterowania modułem z komputera lub kalkulatora programowanego.

Wszystkie te elementy i układy zostały sprawdzone a opracowana koncepcja systemu pomiarowego potwierdzona w praktyce. W trzecim etapie tej pracy /zadanie wdrożeniowe IPBR 11.3.9/ wykonane modele zostaną poddane różnego rodzaju badaniom a w szczególności badaniom wpływu temperatury na działanie modułów i ich poprawioną pracę. Badania te muszą także odpowiedzieć na pytanie jakie ilości ciepła wydziela moduł stacjonarny, co bezpośrednio rzutuje na konstrukcję i wielkość jego obudowy hermetycznej, w której będzie on umieszczony wraz z zasilaczem lub też oddzielnie.

Dalsze badania muszą być wykonywane na okoliczność poprawnej transmisji szeregowej między modułami pomiarowymi a także nad częścią analogową układów modułu. W wyniku przeprowadzonych badań zostanie również opracowana i zaprojektowana płytka modułu a także zaprojektowany i wykonany od podstaw specjalny zasilacz modułu.

Przemysłowy Instytut

Automatyki i Pomiarów

MODUŁ DLA MIERNIKA STRUNOWEGO OPARTEGO NA MIKROPROCESORZE

KONCEPCJA

WARSZAWA, grudzień 1988

WYMAGANIA NA MODUŁ MIERNIKA STRUNOWEGO

UZUPEŁNIENIE OPRACOWANIA "KONCEPCJA MODUŁU MIERNIKA STRUNOWEGO OPARTEGO NA MIKROPROCESORZE"

1. Konstrukcja modułu miernika strunowego oparta będzie na mikrokomputrze jednoukładowym 8031, a w wersji polowej na 80C31.

2. Moduł zamknięty będzie w osobnej obudowie.

3. Moduł będzie zasilany w przypadku:
   - stacjonarnym z zasilaczą o napięciach +5V/+12V i oddzielonogo od niego galwanicznie napięcia ±5V. Napięcia zasilające powinny być rozdzielone galwaniczne od napięć prądu zmiennego i przewodu zerowego.
   - polowym z bateri lub akumulatora o napięciach +5V/±12V

4. Moduł będzie zawierał optoizolację układów współpracy z linią transmisji danych od reszty układów. (Tylko w wersji stacjonarnej).

5. Wymiary modułu mie mogą przekroczyć rozmiaru 240x240 mm.

6. Moduł będzie mógł współpracować z czujnikami dwóch typów:
   - pomiar inicjowany serią 4 impulsów o czestotliwości 850 Hz lub zbliżoną do drgań własnych; po 50 okresach następuje pomiar częstotliwości lub okresu z dokładnością nie mniejszą od 0,01%,
   - pomiar jest dokonywany bez wcześniejszego pobudzenia przez pomiar okresu lub częstotliwości z dokładnością nie mniejszą niż 0,01%.

7. Oprogramowanie modułu pomiarowego w wersji stacjonarnej powinno składać się z trzech podstawowych części:
   - transmisyjnej,
   - sterującej,
   - pomiarowej.

   Oprogramowanie transmisyjne realizuje w sposób programowy obsługę protokołu transmisyjnego BITBUS, przekazuje odebrane od komputera nadrzędnego polecenia do części sterującej i wysyła odpowiedzi do komputera nadrzędnego.

Transmisja jest realizowana w sposób programowy. Oprogramowanie pomiarowe dokonuje pomiaru okresu drgań struny wybranego czujnika.

Oprogramowanie sterujące realizuje zadany algorytm pomiarowy, testuje moduł, interpretuje polecenia odebrane od komputera nadrzędnego i przygotowuje dane do wysłania do komputera nadrzędnego.

Komputer nadrzędny może przysłać następujące rozkazy:

- testowania,
- wykonania pomiaru,
- ustalenia konfiguracji,
- pytania o wyniki pomiaru.

Rozkaz testowania powoduje przetestowanie zasobów modułu i wysłanie do komputera nadrzędnego informacji o jego stanie.

Rozkaz wykonania pomiaru powoduje wykonanie pomiarów czujników, które zostały wyspecyfikowane w rozkazie komputera nadrzędnego. Wyniki pomiarów przechowywane są w buforze do czasu wysłania.

Rozkaz ustalenia konfiguracji powoduje zapamiętanie przez moduł zestawienia i typów czujników dołączonych do niego.

Pytanie o wyniki pomiaru powoduje przesłanie wyników z bufora do komputera nadrzędnego.

8. Moduł pomiarowy w wersji polowej ma oprogramowanie składające się z dwóch części:
   - sterującej,
   - pomiarowej.

   Część pomiarowa jest identyczna jak w wersji stacjonarnej. Część sterująca może na polecenie operatora (przesłane do modułu z kalkulatora programowanego przez interfejs) przetestować moduł lub uruchomić pomiar czujnika i przekazać wynik pomiaru do kalkulatora.

9. Oprogramowanie pakietu komunikacyjnego w komputerze nadrzędnym składa się z dwóch podstawowych części:
   - transmisyjnej,

- sterującej.

Część transmisyjna oprogramowania obsługuje sprzętowe układy protokołu transmisyjnego BITBUS. Nawiązuje ona łączność z modułami pomiarowymi, wysyła do nich rozkazy od części sterującej, odbiera wyniki pomiarów i testów, i przekazuje je do części sterującej.

Część sterująca odpowiada za współpracę z procesorem głównym komputera nadrzędnego (8088) przez pamięć wspólną, przygotowuje rozkazy wysyłane do modułów pomiarowych, testuje pakiet.

Pakiet wykonuje wszystkie czynności na polecenie procesora 8088. Może otrzymać od niego rozkazy:

- testowania,
- zmiany konfiguracji czujników,
- wykonania pomiaru.

Rozkaz testowania powoduje przetestowanie zasobów pakietu ,a następnie wysłanie rozkazów testowania do wszystkich modułów pomiarowych w systemie. Po odebraniu od nich wyników testu informacja o stanie systemu jest przekazywana do procesora 8088 przez pamięć wspólną.

Rozkaz zmiany konfiguracji czujników powoduje wysłanie do poszczególnych modułów pomiarowych konfiguracji czujników, które mają być przez nie cyklicznie przepytywane.

Rozkaz wykonania pomiaru (przekazywany od procesora 8088 w postaci mapy bitowej z zapalonymi bitami odpowiadającymi czujnikom, które należy zmierzyć) powoduje wysłanie do odpowiednich modułów pomiarowych rozkazów wykonania pomiaru wskazanych czujników. Następnie do modułów pomiarowych wysyłane są pytania o wyniki pomiarów. Wyniki te przekazywane są przez pamięć wspólną do procesora 8088.

10. Dla potrzeb aplikacyjnych na komputerze IBM zostanie stworzona biblioteka procedur umożliwiających pomiar oraz testowanie systemu.

Procedura: **pomiar_czujników(mapa,wyniki)**

gdzie: **mapa** – adres obszaru, w którym w postaci mapy bitowej podany zostanie zestaw czujników wyznaczonych do pomiaru. Każdy bajt odpowiada jednemu modułowi pomiarowemu. Każdy bit odpowiada jednemu czujnikowi w odpowiednim module. Bit równy 1 oznacza wybranie czujnika.

wynik – adres obszaru o wystarczająco dużej wielkości, do którego zostaną wpisane wyniki pomiarów. Wynik jednego pomiaru opisywany jest przez 4 bajty: numer modułu (1 bajt), numer czujnika w module (1 bajt), wynik pomiaru (2 bajty).

Wywołanie procedury powoduje przesłanie do procesora komunikacyjnego polecenia wykonania pomiarów wyspecyfikowanych w parametrze **mapa**. Po otrzymaniu informacji o zakończeniu pomiarów następuje odczytanie wyników z pamięci wspólnej na pakiecie procesora komunikacyjnego. Zostają one wpisane do obszaru **wynik** jako ciąg 4 bajtowych rekordów zawierających numer modułu, numer czujnika oraz wynik pomiaru.

Funkcja: **test_systemu(wynik)**

Typ funkcji: boolean

gdzie: **wynik** – adres obszaru, do którego zostaną wpisane wyniki testu w przypadku wystąpienia uszkodzenia.

Funkcja zwraca wartość FALSE (0) jeżeli test systemu nie wykryje żadnych uszkodzeń, wtedy dane w obszarze **wynik** są nie istotne.

Funkcja zwraca wartość TRUE (1) jeżeli test systemu spowoduje wykrycie uszkodzeń. Jednocześnie do obszaru **wynik** zostaną wpisane wyniki testu w postaci ciągu dwubajtowych rekordów, przy czym pierwszy rekord odpowiada pakietowi komunikacyjnemu, a następne rekordy kolejnym modułom pomiarowym. Pierwszy bajt rekordu przynosi informację o uszkodzeniu całego modułu lub pakietu komunikacyjnego, drugi bajt rekordu w postaci

mapy bitowej koduje stany poszczególnych czujników.

Wywołanie funkcji powoduje przesłanie do procesora komunikacyjnego polecenia testu systemu. Po otrzymaniu informacji o zakończeniu testu, z pamięci wspólnej odczytywane są wyniki testu. W przypadku wystąpienia uszkodzenia do obszaru **wynik** zostają wpisane rezultaty testu i funkcja zwraca stosowną odpowiedź.

Procedura: **zmiana_konfiguracji_systemu(mapa)**

gdzie: **mapa** - adres obszaru, w którym w postaci mapy bitowej podany zostanie zestaw czujników wyznaczonych do pomiaru. Modułowi pomiarowemu od powiadają dwa bajty mapy. Każde dwa bity odpowiadają jednemu czujnikowi w odpowiednim module. Na bitach tych zakodowany jest typ czujnika. Wartość 00 oznacza brak czujnika. Wartość 01 oznacza czujnik, który trzeba pobudzać przed pomiarem. Wartość 10 oznacza czujnik nowego typu, którego nie trzeba pobudzać przed pomiarem.

Wywołanie procedury powoduje przesłanie do procesora komunikacyjnego polecenia zmiany konfiguracji czujników przeznaczonych do pomiarów cyklicznych zgodnie ze specyfikacją podaną parametrem **mapa.**

Biblioteka procedur umożliwiających pomiar i testowanie systemu może zostać przygotowana w dwóch wersjach:

- dla języka C (kompilator MicroSoft v. 5.0)
- dla języka Turbo Pascal v. 4.0

Przedstawiciel Zamawiającego: Przedstawiciel Wykonawcy:

S P I S T R E Ś C I

1. WPROWADZENIE

Podstawą wykonania opracowania były: *Wymagania i założenia techniczne do miernika strunowego dla automatycznych systemów pomiarowych opartego na mikroprocesorze* opracowane przez Zamawiającego. W stosunku do nich dokonano w porozumieniu z Zamawiającym, ze względu na przyjęte do stosowania układy analogowe, uściślenia ilości kanałów pomiarowych (p. 5) do ośmiu. Ponadto podczas analizy możliwych wariantów rozwiązania wzięto pod uwagę następujące kryteria wartościowania:

- Stopień spełnienia wymagań funkcjonalnych. Wskazane jest, aby rozwiązanie posiadało zapas, umożliwiający uwzględnienie pewnych zmian w wymaganiach, mogących się pojawić na dalszych etapach projektowania lub eksploatacji.
- Niezawodność urządzenia , zarówno w sensie bezbłędnej i niezakłóconej pracy, jak i niskiej awaryjności. Wymagania tu postawione powinny być spełniane ze znacznym współczynnikiem bezpieczeństwa, dla zabezpieczenia się przed wpływem czynników nieznanych na tym etapie pracy.
- Możliwie niski koszt produkcji. Tu w grę wchodzą takie elementy jak koszt i dostępność użytych podzespołów, koszty wykonania (tzw. technologiczność rozwiązania), łatwość uruchamiania i instalacji.
- Wysoka podatność naprawcza. Możliwość diagnostyki dwupoziomowej - z dokładnością do zespołu w warunkach

użytkowych, do podzespołu, a najlepiej elementu w typowych warunkach pracy serwisu.

2. KONCEPCJA ROZWIĄZANIA SPRZĘTOWEGO

Zadanie opracowania projektu koncepcyjnego sprzętu dla rozproszonego systemu pomiarowego, można zdekomponować na kilka zagadnień:

1. Rozwiązanie zagadnienia transmisji danych pomiarowych pomiędzy modułami pomiarowymi i komputerem nadrzędnym (standard elektryczny, medium, interfejsy).
2. Wybranie odpowiedniego rozwiązania dla konstrukcji mechanicznej i zasilania modułów pomiarowych.
3. Wybór rozwiązania dla układu mikroprocesorowego, realizującego pomiar i transmisję.

Powyższe zagadnienia należy przeanalizować w dwóch wariantach:

A. Dla modułu stacjonarnego.

B. Dla modułu polowego.

Na decyzje, dotyczące rozwiązania sprzętowego dla rozproszonego systemu pomiarowego, spełniającego przedstawione wyżej wymagania, najsilniej rzutują następujące uwarunkowania:

- Wymagania dotyczące transmisji danych między stacją centralną a modułami pomiarowymi:
  - duża ilość modułów pomiarowych,
  - niezbyt intensywny strumień danych transmitowanych,
  - niekrytyczny czas transmisji danych pomiarowych do

stacji centralnej,

- duże odległości pomiędzy poszczególnymi węzłami transmisji,
- brak pewności co do możliwości zapewnienia wspólnego zerowania dla poszczególnych węzłów z marginesem zapewniającym poprawną pracę typowych układów transmisyjnych.

- Wymagania mechaniczno-klimatyczne:
  - moduły będą pracować w warunkach polowych. Ze względu na wyjątkowo duża wilgotność środowiska należy stosować obudowy hermetyczne;
  - w związku z powyższym ważna jest charakterystyka cieplna modułów pomiarowych, a w szczególności moc pobierana oraz sposób odprowadzenia ciepła;
- Osługa operatorska i serwisowa. Ze względu na znaczną odległość pomiędzy modułami system powinien umożliwiać:
  - autodiagnostykę - możliwość wykrycia i wskazania uszkodzonego modułu,
  - elementy, umożliwiające miejscową ocenę sprawności modułu (manipulatory, sygnalizatory).

2.1. Analiza wariantów projektowych

2.1.1. Realizacja sieci transmisji danych pomiarowych

Podstawowe wymagania na łącze transmisji danych pomiarowych, tzn duża ilość węzłów i konieczność przenoszenia danych na duże odległości, znacznie zawężają zakres możliwych do zastosowania rozwiązań. Ze względu na minimalizację kosztów

prowadzenia sieci i ilości interfejsów jedynie sensownym jest przyjęcie dla sieci teletransmisji struktury typu magistrala (bus). Małe wymagania co do szybkości transmisji łagodzą wymagania na warunki propagacji sygnału w sieci. Wśród rozwiązań mogących spełnić założone wymagania można wymienić:

a) systemy pracujące w paśmie podstawowym (typowe rozwiązania sieci typu lokalnego LAN),

b) systemy pracujące w paśmie telefonicznym na łączu trwałym,

c) systemy typu prądowego (dalekopisowego),

d) łącza światłowodowe.

Rozwiązania wymienione w p. b), c) i d) wymagają dość kosztownych (porównując z przewidywanym kosztem pozostałej części urządzenia) urządzeń interfejsowych, takich jak np podzespoły modemowe. Wśród rozwiązań typu a) dostępne handlowo sieci oferują wprawdzie gotowy protokół i dużą szybkość transmisji, ale m.in. ze względu na nią mają zasięg mniejszy od wymaganego (niezbędne stosowanie tzw. repeaterów, tzn wzmacniaczy pośrednich), ponadto wymuszają stosowanie gotowych, zwykle nadmiarowych podzespołów. Z tego względu proponuje się przyjęcie stosowanego w PIE standardu sieci lokalnej, opartego na standardzie BITBUS f-my INTEL, z modyfikacjami dostosowującymi do specyfiki zadania.

W tym rozwiązaniu poziom elektryczny transmisji realizowany jest przy pomocy krajowych układów interfejsowych transmisji symetrycznej UCY75108 (odbiornik) i UCY75110 (nadajnik). Dane przesyłane są w kodzie NRZI. Dla szybkości transmisji

ok. 1200 bit/sek możliwa jest wyłącznie programowa, a więc najtańsza obsługa transmisji. W komputerze IBM PC proponuje się zastosować procesor komunikacyjny f-my STER-PROJEKT, realizujący ten standard transmisji, oparty o system mikroprocesorowy Z80, a umożliwiający modyfikacje parametrów transmisji. Sterownik ten został przedstawiony w dodatku B.

2.1.2. Realizacja zasilania stacjonarnych modułów pomiarowych

Zasilanie modułów pomiarowych może pochodzić z jednego z trzech źródeł:

1. Wspólne zasilanie zdalne poprzez kabel sieci transmisji danych. Przewidywana długość kabla wyklucza jednak ten sposób zasilania.
2. Zasilanie z przetworzeniem lokalnego napięcia stałego 24V.
3. Dołączenie do modułu zasilacza sieciowego;

Brak gwarantowanego wspólnego zerowania dla modułów pomiarowych wymusza stosowanie specjalnych zabiegów dla umożliwienia poprawnej pracy sieci teletransmisji. Nie można dopuścić do połączenia zer w różnych punktach sieci poprzez przewód transmisyjny, gdyż mogłoby to prowadzić do zniszczenia układów interfejsu. Ominąć tę trudność można na dwa sposoby:

1. Moduł pomiarowy powinien być separowany galwanicznie od otoczenia, a uziemiać się przez przewód masy kabla transmisyjnego i komputer stacji centralnej. Przy takim rozwiązaniu wymagane jest stosowanie obudowy z materiałów izolacyjnych (dotyczy to również konstrukcji

sond pomiarowych).

2. Moduł pomiarowy powinien być separowany galwanicznie od sieci transmisji danych. Można to osiągnąć stosując transformator impulsowy w torze transmisji danych; wprowadza on jednakże niepotrzebne tłumienie i zniekształcenia - ponadto wymaga stosowania kodu modulacyjnego bez składowej stałej (np. kod Manchester).

3. Interfejs transmisji danych powinien być separowany galwanicznie od reszty urządzenia (i zera zasilania). Można to osiągnąć stosując sprzężenie przy pomocy transoptorów. Wówczas do zasilania interfejsu transmisji należy wydzielić odrębny, separowany tor zasilania.

Spośród powyższych rozwiązań od strony elektrycznej najprostsze jest rozwiązanie 1) - narzucające jednak na konstrukcję mechaniczną wymagania, które mogą być trudne do spełnienia. Rozwiązanie 3) wydaje się w takiej sytuacji najkorzystniejsze.

2.1.3. Konstrukcja mechaniczna

Wymaganiem konstrukcyjnym, rzutującym na rozwiązanie elektryczne, jest w największym stopniu żądanie hermetyczności obudowy urządzenia. Powoduje ono konieczność szczegółowej oceny bilansu cieplnego urządzenia, gdyż skrajnie utrudnia warunki chłodzenia.

Przy przyjęciu następujących założeń na konstrukcję mechaniczną:

- urządzenie zmontowane jest na płytce o pow. ok. 300

cm^2, zamkniętej w hermetycznej płaskiej obudowie metalowej, poczernionej wewnątrz i z zewnątrz;

- obudowa umocowana jest pionowo w sposób umożliwiający swobodny opływ powietrza;

Mamy do czynienia z następującym rozpływem ciepła:

- od płytki do obudowy - promieniowanie i przewodzenie;
- od obudowy do otoczenia - promieniowanie i konwekcja,

Stosując przybliżone zależności otrzymujemy następujące oszacowanie:

$$\Delta T\ [^{\circ}C] = 10 * P\ [W]$$

Zatem przy stosowaniu układów serii standardowej o temp. pracy 0 ÷ 70 °C i założeniu maksymalnej temperatury otoczenia 40°C urządzenie może wydzielać moc do 3W. Warto zauważyć, że elementy stosowane do wykonania zasilacza mają z reguły wyższe dopuszczalne temperatury pracy. Warto więc rozważyć możliwość rozdzielenia obudów części zasilaczowej i pozostałych układów modułu.

2.1.4. System mikroprocesorowy

Wymagania przyjęte dla modułu pomiarowego jest w stanie spełnić właściwie każdy z dostępnych na rynku mikroprocesorów 8-bitowych. Wybór najwłaściwszego należy zatem oprzeć o sprecyzowane na wstępie kryteria. W poniższej tabeli oceniono najpowszechniej używane w kraju mikroprocesory wg następujących kryteriów:

1. Ilość układów w niezbędnej konfiguracji.
2. Koszt w/w układów.

3. Pobór prądu jw.

4. Dostępność na rynku krajowym.

5. Dostępność narzędzi uruchomieniowych.

6. Zapas możliwości funkcjonalnych.

7. Dostępność wersji CMOS.

8. Dostępność (także planowana) elementów produkcji krajowej.

(ocena w skali 0 - 10 pkt)

Mikroprocesor	1	2	3	4	5	6	7	8	Suma
8080A	-9	-10	-10	8	10	8	-	+	-3
8085	-6	-8	-8	8	10	10	-	+	6
Z80	-6	-6	-6	7	9	10	+	+	8
8035	-3	-1	-3	6	9	5	+	+	13
8031	-3	-4	-3	4	7	9	+	?	10

Powyższe oceny mają w dużym stopniu charakter arbitralny, niemniej pozwalają zilustrować zalety i wady poszczególnych rozwiązań. Wynika z nich, że najkorzystniejsze jest zastosowanie mikroprocesorów jednoukładowych, ze wskazaniem na 8031 ze względu na możliwości, zaś na 8035 ze względu na cenę i (w polskich warunkach) łatwiejszy dostęp.

Porównanie własności mikroprocesorów 8035 i 8031 przedstawiono w dodatku A.

2.2. Propozycja rozwiązania sprzętowego

2.2.1. Moduł pomiarowy stacjonarny

Część pomiarowa

Na podstawie konsultacji z Zamawiającym przewiduje się następująca konstrukcję części analogowej i jej interfejsu z częścia cyfrową. Pobudzenie czujników następuje w układzie tranzystorowym za pośrednictwem scalonego multipleksera analogowego. Sygnał częstotliwościowy z czujników po wstępnym wzmocnieniu w układzie tranzystorowym podawany jest na scalony selektor analogowy, a następnie na układ formujący, skąd już jako sygnał TTL jest odbierany przez część cyfrową. Interfejs między częścia analogową i cyfrową zawiera następujące sygnały:

1. Wybór n-ru pobudzanego kanału (3 linie),
2. Strob blokowania multipleksera,
3. Sygnał pobudzający drgania struny,
4. Strob blokowania selektora,
5. Wejście sygnału częstotliwościwego z czujnika.

Przyjęto następujące wymagania na dokonywanie pomiaru:

- jeśli struna drga, pobudzenie winno nastąpić w fazie z drganiami;
- między pobudzeniem a pomiarem powinien upłynąć czas, wystarczający na ustanie stanów nieustalonych;
- w czasie pobudzania tor wejściowy musi być zablokowany,
- pomiaru okresu należy dokonać z dokładnością nie

mniejszą, niż 100 ns.

Część mikroprocesorowa

Proponuje się rozwiązanie wariantowe:

A. Rozwiązanie najtańsze - koszt układów wchodzących w skład części mikroprocesorowej ok.13 DM.

Rozwiązanie to zawiera mikroprocesor 8035 w układzie z pamięcią zewnętrzną programu EPROM 2716, 2732 lub 2764, buforem adresowym 74LS373 oraz dzielnikiem częstotliwości 7493 .(Uwaga: jeśli okaże się, że do produkcji będzie użyty mikroprocesor ze źródeł zachodnich, wskazane jest zastosowanie układu 8039 - w tej samej cenie, a o nieco większych możliwościach).

Cechy rozwiązania:

- całkowicie programowa obsługa transmisji, umożliwiająca przesyłanie danych z prędkością 2400 bd(bit/sek),
- pomiar okresu z dokładnością 8μs, co wymaga zliczenia min. 80 cykli drgań dla osiągnięcia żądanej dokładności 0.1μs,
- konieczność rozdzielenia w czasie obsługi transmisji i dokonywania pomiaru, co wydłuża cykl pomiarowy w przypadku pomiaru grupowego kilku czujników dołączonych do tego samego modułu oraz istotne utrudnienie sterowania siecią.

B. Rozwiązanie pośrednie - koszt układów ok. 17 DM

W rozwiązaniu tym układ opisany w wariancie A) uzupełnia się o programowany licznik 8253 (lub 8254 -

który posiada wersję CMOS). Odpowiednik układu 8253 jest wytwarzany w ZSRR. W stosunku do poprzedniego uzyskuje się poprawę następujących parametrów:

- pomiar okresu z dokładnością 0.5μs , co wymaga zliczenia min. 5 cykli dla osiągnięcia zadanej dokładności;
- nie ma konieczności rozdziału w czasie transmisji i pomiaru;

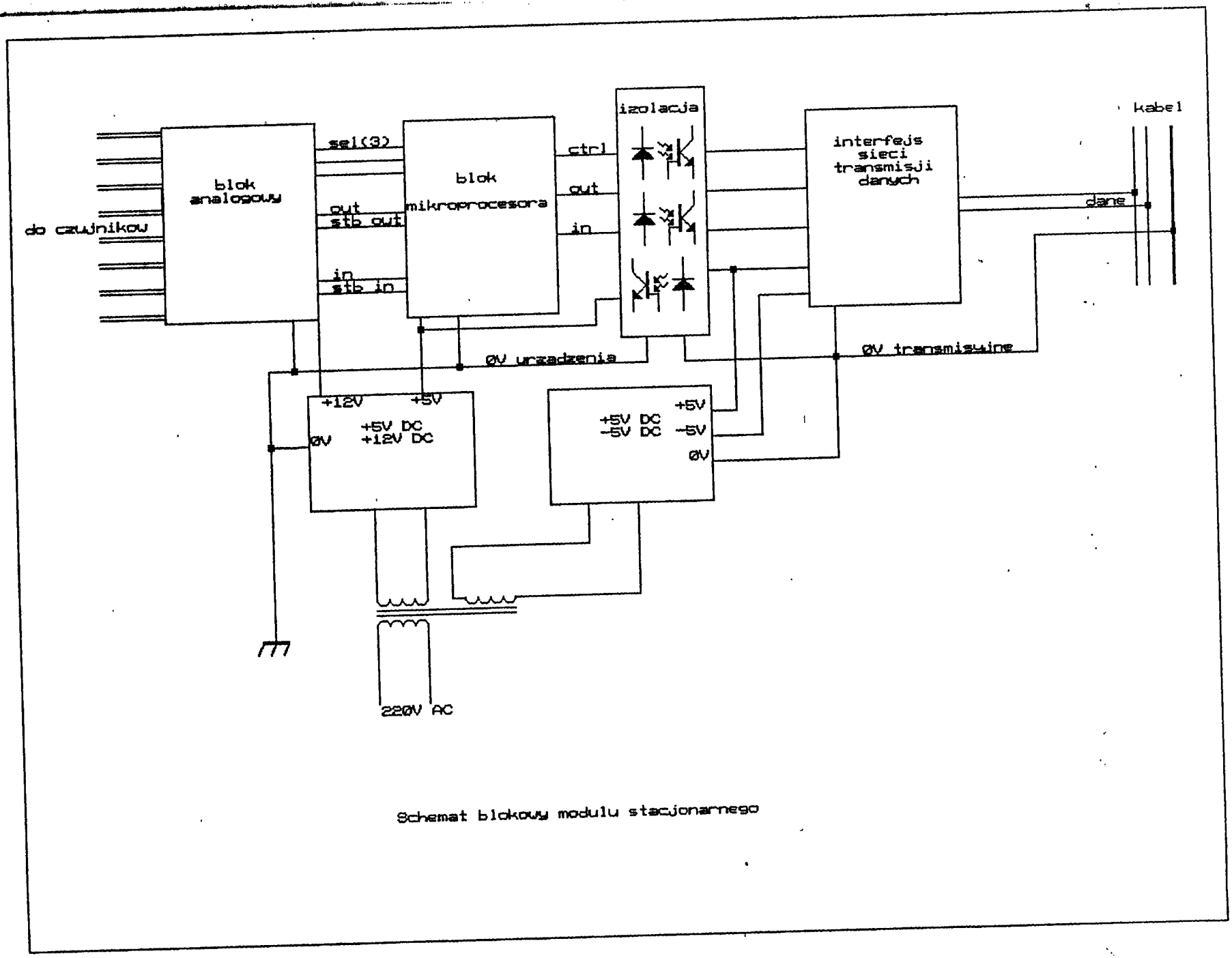
C. Rozwiązanie najdroższe - koszt układów ok.18DM

To rozwiązanie zawiera mikroprocesor 8031 z pamięcią zewnętrzną 2764 i rejestrem 74LS373. Rozwiązanie to ma następujące własności:

- prędkość transmisji przy czysto mikroprogramowej obsłudze ok 4800 bd;
- pomiar okresu z dokładnością 1μs, co wymaga zliczenia min. 10 cykli;
- nie ma konieczności rozdziału w czasie transmisji pomiaru.

Ponadto procesor ten ma znaczny zapas możliwości, co pozwalałoby w przyszłości na ewentualne rozszerzenie funkcji modułu pomiarowego, jak np. filtracja cyfrowa, obróbka statystyczna danych pomiarowych.

Zespół projektujący opowiada się za rozwiązaniem C jako dającym największe możliwości rozwoju dalszego systemu i najprostrzą realizacje sprzętową. Możliwe jest dalsze uproszczenie konstrukcji modułu w tym wariancie po zastosowaniu wersji mikroprocesora z pamięcią EPROM (8751).



Schemat blokowy modulu stacjonarnego

Interfejs transmisji danych

Proponuje się zastosowanie omawianego wcześniej interfejsu sieci typu BITBUS, pracującego z szybkością transmisji 2400 bd, w układzie z transoptorową separacją pomiędzy interfejsem i pozostałą częścią modułu pomiarowego i odrębnym torem zasilania układów interfejsu. Jako medium transmisyjne należy użyć dowolnego kabla skrętkowego, zawierającego dwa przewody transmisji danych (skrętka) o oporze nie większym niż 0,10 Ω/m i przewód (lub kilka połączonych) masy o oporze nie większym, niż 0.05 Ω/m (np. kabel YPMY 2x0.35mm^2 - skrętka w ekranie).

Zasilanie

Układ powinien być zasilany z sieci napięcia przemiennego 220V/50Hz. Zasilacz powinien dostarczać napięć stabilizowanych +12V/0.5A i +5V/0.5A do zasilania części zasadniczej oraz +5V/0.2A i -5V/0.2A w oddzielnym separowanym torze do zasilania interfejsu transmisji. Zmiany napięć stabilizowanych przy wahaniach napięcia sieci ±20% i zmianach obciążenia 0 - max nie powinny być większe niż ±3%

2.2.2. Moduł pomiarowy polowy

Część pomiarowa

Ponieważ moduł polowy ma obsługiwać jeden czujnik, zbędne są multiplekser i selektor oraz sygnały wyboru kanału. Poza tym założenia pozostają bez zmian.

Część mikroprocesorowa

Proponuje się zastoswanie tych samych wariantów, co w module

stacjonarnym. Transmisja szeregowa we współpracy z kalkulatorem programowanym będzie się odbywać za pośrednictwem złącza RS232c obsługiwanego programowo. W wariancie układowym C, tj. z użyciem mikroprocesora 8031 jest to szczególnie ułatwione, gdyż procesor ten posiada wbudowany port szeregowy. W module polowym należy zastosować wersje CMOS układów systemu mikroprocesorowego.

Transmisja danych

Do współpracy z kalkulatorem programowanym proponuje się zastosować standardowe złącze transmisji szeregowej RS232C obsługiwane programowo z zastosowaniem układów interfejsowych MC1488 i MC1489 lub SN57150 i SN75154 (lub ich odpowiedników), albo, jeśli to będzie możliwe, złącza równoległego typowego dla zastosowanego modelu kalkulatora. Decyzję tę należy podjąć na etapie projektu wstępnego na podstawie badań modelowych współpracy.

Zasilanie

Moduł przenośny powinien być zasilany bateryjnie. Typ i ilość baterii (lub akumulatorów) powinien zostać określony na podstawie badań modelowych modułu pomiarowego.

2.3. Procesor komunikacyjny dla IBM/PC

Proponuje się zastosowanie sterownika transmisji szeregowej BITBUS opracowanego w Zakładzie Rzemieślniczym STER-PROJEKT. Pakiet sterownika transmisji szeregowej BITBUS do komputera IBM PC/XT przeznaczony jest do realizacji sieci lokalnej sterowników mikroprocesorowych współpracyjących z mikrokomputerem IBM PC/XT. Pakiet pozwala na tworzenie

IBM PC/XT przeznaczony jest do realizacji sieci lokalnej sterowników mikroprocesorowych współpracyjących z mikrokomputerem IBM PC/XT. Pakiet pozwala na tworzenie konfiguracji:

- połączenie komputerów IBM PC/XT w sieć lokalna,
- dołączenie sterowników mikroprocesorowych wyposażonych w odpowiedni moduł współpracy do komputera IBM PC/XT,
- połączenie komputerów IBM PC/XT i sterowników mikroprocesorowych w sieć lokalna.

Komputer IBM PC/XT wyposażony w pakiet sterownika BITBUS może pełnić w sieci funkcję monitorująca, tzn. zbierania i obrazowania stanu sterowanego procesu, jak i funkcję sterująca, tzn. zmiany w sterownikch parametrów sterowanego procesu . Zastosowany moduł opisany został w dodatku B.

2.4. Wytyczne do działań na etapie projektu wstępnego

Podczas badań modelowych sprzętu szczególną uwagę należy zwrócić na:

- zbadanie warunków cieplnych pracy modułów stacjonarnych. Jeśli to będzie możliwe, należy dokonać pomiaru temperatur w warunkach zbliżonych do docelowych (obudowa,rozmiary, mocowanie),
- sprecyzowanie sposobu współpracy modułu przenośnego s minikalkulatorem,
- sprawdzenie jakości pracy sieci transmisji danych pomiarowych w warunkach możliwie zbliżonych do docelowych.

3. KONCEPCJA OPROGRAMOWANIA MODUŁU POMIAROWEGO

3.1. Wersja stacjonarna

Oprogramowanie modułu pomiarowego w wersji stacjonarnej powinno składać się z trzech podstawowych części:

- sterującej,
- transmisyjnej,
- pomiarowej.

Oprogramowanie transmisyjne realizuje w sposób programowy obsługę protokołu transmisyjnego BITBUS, przekazuje odebrane od komputera nadrzędnego polecenia do części sterującej i wysyła odpowiedzi do komputera nadrzędnego. Transmisja powinna być realizowana w sposób programowy ze względu na duży koszt układów transmisji w standardzie BITBUS dostosowanych do mikroprocesorów 8048 i 8051.

Oprogramowanie pomiarowe wykonuje pomiar częstotliwości drgań struny wybranego czujnika. Po pobudzeniu struny zliczany jest czas trwania np. 32 okresów drgań. Po obliczeniu częstotliwości wynik pomiaru jest zapamiętywany.

Oprogramowanie sterujące realizuje zadany algorytm pomiarowy, testuje moduł, interpretuje polecenia odebrane od komputera nadrzędnego i przygotowuje dane do wysłania do komputera nadrzędnego. Szczegółowy sposób działania tej części oprogramowania zależy od przyjętego rozwiązania:

Wariant I - moduł wykonuje wszystkie czynności na polecenie komputera nadrzędnego. Komputer nadrzędny może przysłać następujące rozkazy:

- testowania,
- wykonania pomiaru;,
- pytania o wyniki pomiaru.

Rozkaz testowania powoduje przetestowanie zasobów modułu i wysłanie do komputera nadrzędnego informacji o jego stanie.

Rozkaz wykonania pomiaru powoduje wykonanie pomiarów czujników, które zostały wyspecyfikowane w rozkazie komputera nadrzędnego. Wyniki pomiarów przechowywane są w buforze do czasu wysłania.

Pytanie o wyniki pomiaru powoduje przesłanie wyników z bufora do komputera nadrzędnego.

Wariant II - moduł wykonuje kolejno, w pętli pomiary częstotliwości wszystkich czujników dołączonych do niego. Odczytane wyniki zapamiętuje w buforze i modyfikuje je przy każdym kolejnym pomiarze. Odczytane wyniki są przy tym sprawdzane, a te które rażąco odbiegają od poprzednich pomiarów danego czujnika są odrzucane (szczegółowe kryteria odrzucania wyników pomiarów zostaną opracowane w trakcie badań systemu).

Komputer nadrzędny przysyła do modułu rozkazy:

- testowania,
- pytania o stan.

Rozkaz testowania powoduje przetestowanie zasobów modułu i wysłanie do komputera nadrzędnego informacji o jego stanie.

Pytanie o stan powoduje wysłanie do komputera

nadrzędnego wyników pomiarów tych czujników, które zmieniły swój stan od poprzedniego pytania o stan.

Wariant III - moduł może być programowo przełączany w dwa tryby pracy.

W trybie pierwszym moduł wykonuje wszystkie czynności na polecenie komputera nadrzędnego. Komputer nadrzędny może przysłać następujące rozkazy:

- testowania,
- wykonania pomiaru,
- pytania o wyniki pomiaru,
- przełączenia w drugi tryb pracy.

Rozkaz testowania powoduje przetestowanie zasobów modułu i wysłanie do komputera nadrzędnego informacji o jego stanie.

Rozkaz wykonania pomiaru powoduje wykonanie pomiarów czujników, które zostały wyspecyfikowane w rozkazie komputera nadrzędnego. Wyniki pomiarów przechowywane są w buforze do czasu wysłania.

Pytanie o wyniki pomiaru powoduje przesłanie wyników z bufora do komputera nadrzędnego.

Przełączenie w drugi tryb pracy powoduje, że moduł wykonuje kolejno, w pętli pomiary częstotliwości wszystkich czujników dołączonych do niego. Odczytane wyniki zapamiętuje w buforze i modyfikuje je przy każdym kolejnym pomiarze. Odczytane wyniki są przy tym sprawdzane, a te, które rażąco odbiegają od poprzednich pomiarów danego czujnika są odrzucane.

Komputer nadrzędny przysyła do modułu rozkazy:

- testowania,
- pytania o stan,
- przełączenia w pierwszy tryb pracy.

Rozkaz testowania powoduje przetestowanie zasobów modułu i wysłanie do komputera nadrzędnego informacji o jego stanie.

Pytanie o stan powoduje wysłanie do komputera nadrzędnego wyników pomiarów tych czujników, które zmieniły swój stan od poprzedniego pytania o stan.

Wariant IV - moduł wykonuje kolejno, w pętli pomiary częstotliwości wszystkich czujników, których sprawdzanie zlecił mu komputer nadrzędny. Odczytane wyniki zapamiętuje w buforze i modyfikuje je przy każdym kolejnym pomiarze. Odczytane wyniki są przy tym sprawdzane, a te, które rażąco odbiegają od poprzednich pomiarów danego czujnika są odrzucane.

Komputer nadrzędny przysyła do modułu rozkazy:

- testowania,
- ustalenia konfiguracji czujników,
- wykonania pomiaru,
- pytania o stan.

Rozkaz testowania powoduje przetestowanie zasobów modułu i wysłanie do komputera nadrzędnego informacji o jego stanie.

Rozkaz ustalenia konfiguracji czujników powoduje zmianę

zestawu czujników mierzonych cyklicznie przez moduł.

Rozkaz wykonania pomiaru powoduje wykonanie pomiaru czujnikow wyspecyfikowanych w rozkazie. Powinny to być czujniki, których moduł nie mierzy cyklicznie.

Pytanie o stan powoduje wysłanie do komputera nadrzędnego wyników pomiarów tych czujników, które zmieniły swój stan od poprzedniego pytania o stan lub zostały zmierzone po odebraniu rozkazu wykonania pomiaru.

Powyższy opis dotyczy modułu pomiarowego zbudowanego w oparciu o mikroprocesor 8051. Przy wykorzystaniu procesora 8048 nie jest możliwe jednoczesne wykonywanie pomiarów i obsługa transmisji (wynika to z konfiguracji mikroprocesora). W związku z tym moduł pomiarowy nie może wykonywać pomiarów czujników w sposób cykliczny Każdy pomiar wykonywać się musi na polecenie komputera nadrzędnego. Z tego powodu dla mikrokomputera 8031 możliwy jest jedynie wariant I

3.2. Wersja polowa

Moduł pomiarowy w wersji polowej ma proste oprogramowanie składajace się z dwóch części:

- sterujacej,
- pomiarowej.

Część pomiarowa jest identyczna jak w wersji stacjonarnej.

Część sterujaca może na polecenie operatora (przesłane do modułu z kalkulatora programowanego przez interfejs) uruchomić pomiar wybranego czujnika i przekazać wynik

pomiaru do kalkulatora lub przetestować moduł.

3.3. Oprogramowania pakietu komunikacyjnego komputera nadrzędnego

Oprogramowanie pakietu komunikacyjnego w komputerze nadrzędnymskłada się z dwóch podstawowych części:

- transmisyjnej,
- sterującej.

Część transmisyjna oprogramowania obsługuje sprzętowe układy protokołu transmisyjnego BITBUS. Nawiązuje ona łączność z modułami pomiarowymi, wysyła do nich rozkazy od części sterującej, odbiera wyniki pomiarów i testów, i przekazuje je do czesci sterującej.

Część sterująca odpowiada za współpracę z procesorem głównym komputera nadrzędnego (8088) przez pamięć wspólna, przygotowuje rozkazy wysyłane do modułów pomiarowych, testuje pakiet. Działanie jej jest różne dla poszczególnych rozwiązań oprogramowania:

Wariant I pakiet wykonuje wszystkie czynności na polecenie procesora 8088. Może otrzymać od niego rozkazy:

- testowania,
- wykonania pomiaru.

Rozkaz testowania powoduje przetestowanie zasobów pakietu ,a następnie wysłanie rozkazów testowania do wszystkich modułów pomiarowych w systemie. Po odebraniu od nich wyników testu informacja o stanie systemu jest przekazywana do procesora 8088 przez pamięć wspólna.

Rozkaz wykonania pomiaru (przekazywany od procesora 8088 w postaci mapy bitowej z zapalonymi bitami odpowiadającymi czujnikom, które należy zmierzyć) powoduje wysłanie do odpowiednich modułów pomiarowych rozkazów wykonania pomiaru wskazanych czujników. Następnie do modułów pomiarowych wysyłane są pytania o wyniki pomiarów. Wyniki te przekazywane są przez pamięć wspólną do procesora 8088.

Wariant II - pakiet cyklicznie przepytuje moduły pomiarowe o ich stan. Odebrane wyniki pomiarów umieszcza w tabeli w pamięci wspólnej pakietu. Są one cały czas dostępne dla oprogramowania procesora 8088. Procesor 8088 może przesłać do pakietu rozkaz testowania, który powoduje przetestowanie zasobów pakietu, a następnie wysłanie do wszystkich modułów pomiarowych rozkazu testowania. Wyniki testowania przekazywane są do procesora 8088.

Wariant III - pakiet może pracować w dwóch trybach. W pierwszym pakiet wykonuje wszystkie czynności na polecenie procesora 8088. Może otrzymać od niego rozkazy:

- testowania,
- wykonania pomiaru,
- przełączenia w drugi tryb pracy.

Rozkaz testowania powoduje przetestowanie zasobów pakietu,a następnie wysłanie rozkazów testowania do wszystkich modułów pomiarowych w systemie. Po odebraniu od nich wyników testu informacja o stanie systemu jest przekazywana do procesora 8088 przez pamięć wspólną.

Rozkaz wykonania pomiaru (przekazywany od procesora 8088 w postaci mapy bitowej z zapalonymi bitami odpowiadającymi czujnikom, które należy zmierzyć) powoduje wysłanie do odpowiednich modułów pomiarowych rozkazów wykonania pomiaru wskazanych czujników. Następnie do modułów pomiarowych wysyłane są pytania o wyniki pomiarów. Wyniki te przekazywane są przez pamięć wspólną do procesora 8088.

Przełączenie w drugi tryb pracy powoduje, że pakiet cyklicznie przepytuje moduły pomiarowe o ich stan. Odebrane wyniki pomiarów umieszcza w tabeli w pamięci wspólnej pakietu. Są one cały czas dostępne dla oprogramowania procesora 8088. Procesor 8088 może przesłać do pakietu rozkazy:

- testowania;
- przełączenia w pierwszy tryb pracy.

Rozkaz testowania powoduje przetestowanie zasobów pakietu, a następnie wysłanie do wszystkich modułów pomiarowych rozkazu testowania. Wyniki testowania przekazywane są do procesora 8088.

Rozkaz przełączenia w pierwszy tryb pracy powoduje przejście w tryb pracy na polecenie procesora 8088.

Wariant IV - pakiet cyklicznie przepytuje moduły pomiarowe o ich stan. Wybiera przy tym te moduły, które zostały zdefiniowane przez procesor 8088. Odebrane wyniki pomiarów umieszcza w tabeli w pamięci wspólnej pakietu. Są one cały czas dostępne dla oprogramowania procesora 8088. Procesor 8088 może przesłać do pakietu rozkazy:

- testowania;
- zmiany konfiguracji czujników;
- wykonania pomiaru wskazanych czujników.

Rozkaz testowania powoduje przetestowanie zasobów pakietu, a następnie wysłanie do wszystkich modułów pomiarowych rozkazu testowania. Wyniki testowania przekazywane są do procesora 8088.

Rozkaz zmiany konfiguracji czujników powoduje wysłanie do poszczególnych modułów pomiarowych konfiguracji czujników, które mają być przez nie cyklicznie przepytywane.

Rozkaz wykonania pomiaru wskazanych czujników powoduje wysłanie do odpowiednich modułów pomiarowych rozkazów pomiaru wskazanych czujników (powinny to być czujniki nie mierzone cyklicznie). Następnie pakiet wysyła do tych modułów pytanie o wyniki pomiarów i przekazuje odebrane wyniki do pakietu 8088 przez pamięć wspólną.

W wypadku, gdy moduły pomiarowe będą wykonane w oparciu o procesor 8048, pakiet komunikacyjny musi nieco zmienić sposób współpracy z nimi w wariantacg II - IV. Zamiast cyklicznego pytania o stan czujników muszą być do kolejnych modułów wysyłane rozkazy wykonania pomiarów wszystkich (w wariancie IV - wybranych do cyklicznego przepytywania) czujników, następnie pytania o wynik pomiaru tych czujników. Otrzymane wyniki będą umieszczane w tabeli w pamięci wspólnej.

3.4. Zestaw procedur bibliotecznych komputera nadrzędnego

Dla potrzeb aplikacyjnych na komputerze IBM zostanie stworzona biblioteka procedur umożliwiających pomiar oraz testowanie systemu.

Propozycje procedur i funckji.

Procedura: **pomiar_czujników(mapa,wyniki)**

gdzie: **mapa** - adres obszaru, w którym w postaci mapy bitowej podany zostanie zestaw czujników wyznaczonych do pomiaru. Każdy bajt odpowiada jednemu modułowi pomiarowemu. Każdy bit odpowiada jednemu czujnikowi w odpowiednim module. Bit równy 1 oznacza wybranie czujnika.

wynik - adres obszaru o wystarczająco dużej wielkości, do którego zostaną wpisane wyniki pomiarów. Wynik jednego pomiaru opisywany jest przez 4 bajty: numer modułu (1 bajt), numer czujnika w module (1 bajt), wynik pomiaru (2 bajty).

Realizacja procedury **pomiar_czujników** zależna jest trybu pracy systemu pomiarowego.

- wariant I - pomiary na żądanie

Wywołanie procedury powoduje przesłanie do procesora komunikacyjnego polecenia wykonania pomiarów wyspecyfikowanych w parametrze **mapa** czujników. Po

otrzymaniu informacji o zakończeniu pomiarów następuje odczytanie wyników z pamięci wspólnej na pakiecie procesora komunikacyjnego. Zostają one wpisane do obszaru **wynik** jako ciąg 4 bajtowych rekordów zawierających numer modułu, numer czujnika oraz wynik pomiaru.

- rozwiązanie II - pomiary cykliczne

Wywołanie procedury powoduje odczytanie z pamięci wspólnej na pakiecie procesora komunikacyjnego wyników pomiarów odpowiednich czujników wyspecyfikowanych parametrem **mapa**, a następnie wpisanie ich do obszaru **wynik** jako ciągu 4 bajtowych rekordów zawierających numer modułu, numer czujnika oraz wynik pomiaru.

- rozwiązanie III - przełączane tryby pracy

(pomiary na żądanie / pomiary cykliczne)

W zależności od aktualnego trybu pracy wywołanie procedury powoduje:

- w przypadku pomiarów czujników na żądanie przesłanie do procesora komunikacyjnego polecenia wykonania pomiarów wyspecyfikowanych w parametrze **mapa** czujników. Po otrzymaniu informacji o zakończeniu pomiarów następuje odczytanie wyników z pamięci wspólnej na pakiecie procesora komunikacyjnego. Zostają one wpisane do obszaru **wynik** jako ciąg 4 bajtowych rekordów zawierających numer modułu, numer czujnika oraz wynik pomiaru.
- w przypadku pomiarów cyklicznych: odczytanie z pamięci wspólnej na pakiecie procesora

komunikacyjnego wyników pomiarów odpowiednich czujników wyspecyfikowanych parametrem **mapa**, a następnie wpisanie ich do obszaru **wynik** jako ciągu 4 bajtowych rekordów zawierających numer modułu, numer czujnika oraz wynik pomiaru.

- rozwiązanie IV - hybrydowy tryb pracy (cykliczny pomiar zestawu czujników oraz pomiar na żądanie dla pozostałych czujników)

  Wywołanie procedury **pomiar_czujników** może jednocześnie dotyczyć czujników obu w/w grup.

  Na podstawie specyfikacji czujników w parametrze **mapa** następuje wybranie czujników, których pomiary odbywają się na żądanie. Jeżeli takie czujniki istnieją do procesora komunikacyjnego przesyłany jest rozkaz pomiaru tych czujników. Po otrzymaniu informacji o zakończeniu pomiaru następuje odczytanie z pamięci wspólnej wyników pomiarów, zarówno czujników pobudzonych na żądanie opratora jak i czujników pobudzanych cyklicznie, a wyszczególnionych w parametrze **mapa**. Wyniki zostają wpisane do obszaru **wynik** jako ciąg 4 bajtowych rekordów zawierających numer modułu, numer czujnika oraz wynik pomiaru.

Funkcja: **test_systemu(wynik)**

Typ funkcji: boolean

gdzie: **wynik** - adres obszaru, do którego zostaną wpisane wyniki testu w przypadku wystąpienia uszkodzenia.

Funkcja zwraca wartość FALSE (0) jeżeli test systemu nie wykryje żadnych uszkodzeń, wtedy dane w obszarze **wynik** są nie istotne.

Funkcja zwraca wartość TRUE (1) jeżeli test systemu spowoduje wykrycie uszkodzeń. Jednocześnie do obszaru **wynik** zostaną wpisane wyniki testu w postaci ciągu dwu-bajtowych rekordów, przy czym pierwszy rekord odpowiada pakietowi komunikacyjnemu, a następne rekordy kolejnym modułom pomiarowym. Pierwszy bajt rekordu przynosi informację o uszkodzeniu całego modułu lub pakietu komunikacyjnego, drugi bajt rekordu w postaci mapy bitowej koduje stany poszczególnych czujników.

Realizacja funkcji **test_systemu** jest nie zależna od przyjętego wariantu trybu pracy systemu.

Wywołanie funkcji powoduje przesłanie do procesora komunikacyjnego polecenia testu systemu. Po otrzymaniu informacji o zakończeniu testu, z pamięci wspólnej odczytywane są wyniki testu. W przypadku wystąpienia uszkodzenia do obszaru **wynik** zostają wpisane rezultaty testu i funcja zwraca stosowną odpowiedź.

Procedura: **zmiana_trybu_pracy(tryb)**

gdzie: **tryb** - określa nowy tryb pracy systemu. System może pracować w dwóch trybach: pomiary na żądanie operatora oraz pomiary cykliczne. Parametr **tryb** koduje nowy tryb pracy.

Istnienie procedury **zmiana_trybu_pracy** ma sens jedynie dla III rozwiązania sposobu pracy systemu pomiarowego.

Wywołanie procedury powoduje przesłanie do pakietu komunikacyjnego rozkazu przełączenia systemu w podany tryb pracy. Jeżeli system pracuje już w żądanym trybie, nie jest podejmowana żadna akcja.

Procedura: **zmiana_konfiguracji_systemu(mapa)**

gdzie: **mapa** - adres obszaru, w którym w postaci mapy bitowej podany zostanie zestaw czujników wyznaczonych do cyklicznego pomiaru. Każdy bajt odpowiada jednemu modułowi pomiarowemu. Każdy bit odpowiada jednemu czujnikowi w odpowiednim module. Bit równy 1 oznacza wybranie czujnika.

Istnienie procedury **zmiana_konfiguracji_systemu** ma sens jedynie dla IV rozwiązania trybu pracy systemu pomiarowego.

Wywołanie procedury powoduje przesłanie do procesora komunikacyjnego polecenia zmiany konfiguracji czujników przeznaczonych do pomiarów cyklicznych zgodnie ze specyfikacją podaną parametrem **mapa**.

Biblioteka procedur umożliwiających pomiar i testowanie systemu może zostać przygotowana w dwóch wersjach:

- dla języka C (kompilator MicroSoft v. 5.0)
- dla języka Turbo Pascal v. 4.0

3.5. Oszacowanie czasu odczytu wyników pomiaru

Założenia:

- predkość transmisji 1200 bitów/s,

- długość rozkazu wykonania pomiaru - 6 bajtów,
- długość potwierdzenia przyjęcia rozkazu - 4 bajty,
- czas pomiaru jednego czujnika - 64 ms.,
- długość pytania o wyniki - 4 bajty,
- długość rekordu zawierającego wyniki pomiarów - 20 bajtów.

1. Czas odczytu wyników pomiarów wszystkich (64 x 8) czujników w systemie:
   - rozwiązanie I - 30 - 40 s,
   - rozwiązanie II-IV - 10 - 20 ms.
2. Czas odczytu wyniku pomiaru jednego czujnika:
   - rozwiązanie I - 1 - 2 s,
   - rozwiązanie II-IV - 25 - 30 μs.

4. DIAGNOSTYKA I URUCHAMIANIE MODUŁÓW POMIAROWYCH

Nowoczesne rozwiązania konstrukcyjne pozwalają zawrzeć w systemie mikroprocesorowym rozbudowane funkcje diagnostyki i autodiagnostyki. Przyjęcie założenia o możliwie prostej konstrukcji sprzętowej i w jak największym stopniu przerzucenie realizacji zadań ze sprzętu na oprogramowanie, umożliwia w wielu przypadkach diagnostykę z dokładnością do pojedyńczych elementów elektronicznych.

Przewiduje się trzy sposoby realizacji funkcji diagnostycznych w systemie:

1. Wykrywanie błędów i uszkodzeń w trakcie realizacji podstawowych zadań systemu.
2. Wykonywanie na żądanie testowania sprawności systemu

pomiarowego.

3. Badanie i uruchamianie modułów pomiarowych przy pomocy dodatkowej aparatury i osprzętu.

Przypadek 1 jest naturalną konsekwencją programowej realizacji zadań systemu. Podczas programowego sterowania czujnikami należy wykrywać możliwie dużo nietypowych stanów systemu jak np. brak zbocza sygnału pomiarowego przez okres dłuższy niż przyjęty zakres pomiarowy. Powyższe dotyczy także obługi transmisji danych. Wszelkie stany niezgodne z zaproponowanym protokołem BITBUS powinny być wykrywane i sygnalizowane. Błędu powstałe w trakcie pomiaru powinny być sygnalizowane komputerowi nadrzędnemu, natomiast błędy i uszkodzenia układów transmisji danych mogą być wykryte przez komputer nadrzędny. W zależności od przyjętego trybu pracy uszkodzenia i błedy będą wykrywane w chwili dokonywania pomiarów na rozkaz komputera nadrzędnego lub w chwili powstania uszkodzenia w przypadku cyklicznego dokonywania pomiarów.

Przypadek 2 realizowany jest przez rozkaz **testuj** przesyłany z komputera nadrzędnego do modułu. W ramach tej funkcji możliwe jest wykonanie następujących testów:

- test sprawności arytmometru i wewnętrznych układów sterujących mikroprocesora,
- test poprawności pamięci programu EPROM,
- test wewnętrznej pamięci RAM,
- test układów liczników i układu przerwań,
- test poprawności pracy kanałów pomiarowych (dokonanie

pomiaru testowego na każdym z aktywnych kanałów).

Wskazane jest wyposażenie modułu pomiarowego w przycisk TEST, który uruchamiałby niezależnie powyższe sekwencje testowe, oraz w zestaw lampek (różnokolorowych diod LED) sygnalizujących przebieg i wyniki testu. Rozwiązanie takie umożliwa wykonanie tych testów przez obsługe na obiekcie w przypadku gdy zerwana zozstanie łączność z modułem. Wynik testu określi wtedy przyczynę uszkodzenia.

Przypadek 3 wystąpi podczas uruchamiania produkowanych modułów pomiarowych oraz podczas ich napraw i przeglądów serwisowych. Do wykonania tych badań niezbędne jest dodatkowe wyposażenie umożliwiające pełniejszą diagnostykę niż ma to miejsce w dwóch poprzednich przypadkach. Stanowisko, na którym będą prowadzone badania, powinno zapenić możliwość odpowiedniego sprzężenia wejść i wyjść modułu z aparaturą serwisową. Do pełnego zbadania i diagnozowania uszkodzeń konieczne może być wyprowadzenie części sygnałów z wnętrza modułu. W tym celu należy odpowiednio przygotować te punkty pomiarowe.

Przewiduje się trzy możliwe warianty wykonania stanowiska pomiarowego:

A. Prosty układ sprzęgający wejścia i wyjścia pomiarowe przez generatory o ustalonych czestotliwościach wzbudane prawidłowym sygnałem pobudzającym, z możliwością podłączenia oscyloskopu. Ponadto układ symulujący obciążenie linią transmisji danych z możliwością obserwacji stanu linii np. analizatorem protokołu lub sygnatur. Należy przewidzieć dodatkowe

wejście testowe powodujące uruchomienie testu układów nadawania i odbioru.

B. Układ sprzegający zawierający własny mikroprocesor, który będzie generował automatycznie odpowiednie sekwencje testowe i badał odpowiedzi układu testowanego. układ powinien być wyposażony w wyświetlacz podający wyniki testu.

C. Układ sprzęgający wykonany na bazie typowego mikrokomputera wyposażonego w moduł sprzężenia z układem badanym. Działanie jak w przypadku B z możliwością wygodnego sterowania przebiegiem testu i odczytu wyników.

DODATEK A

MIKROKMPUTERY JEDNOUKŁADOWE FIRMY INTEL

SPIS TREŚCI

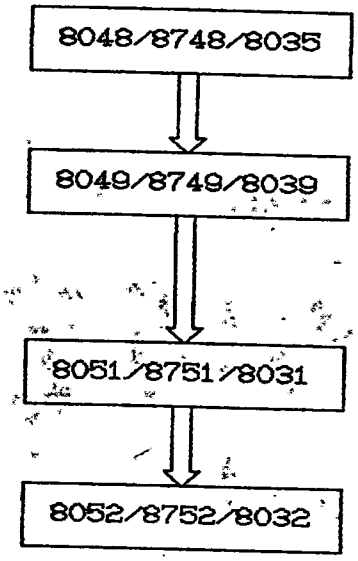
A.1. MIKROKOMPUTERY RODZINY MCS-48

Podstawowa rodzina mikroprocesorów jednoukładowych jest 8-bitowa rodzina układów firmy Intel MCS-48, przeznaczonych do wykorzystania w układach sterowania i zawierających w jednej 40-nóżkowej obudowie:

- jednostkę centralną CPU czyli mikroprocesor,
- wbudowany układ zegara,
- 64 lub 128 bajtów pamięci RAM, służącej do przechowywania danych bieżących,
- 0 ÷ 4 Kbajtów pamięci programu ROM ,
- układ timera/licznika, służący do odliczania określonych odcinków czasu lub zliczania zdarzeń zewnętrznych (impulsów),
- 27 (lub 16) linii wyjścia-wejścia zgrupowanych w 3 (2) porty, za pomocą których może się komunikować z otoczeniem.

Mikrokomputery tej rodziny mają pojedyncze napięcie zasilające: 5V

Linię rozwojową rodziny mikrokomputerów MCS-48 przedstawia rys.1. Wraz z rozwojem zwiększa się ilość pamięci RAM i ROM, ilość portów, liczników/timerów oraz cena układów. Układy umieszczone w jednym wierszu posiadają ten sam schemat funkcjonalny, w szczególności tę samą ilość pamięci programu, a różnią się jej typem i umieszczeniem wewnątrz układu lub w oddzielnym układzie dołączanym z zewnątrz.



rys.1.

A.1.1. Pamięć programu i danych

Układy 8048, 8049, 8051 i 8052 posiadają pamięć ROM umieszczoną wewnątrz układu i zapisywaną przez producenta.

Układy 8748, 8749,8751 i 8752 posiadają również pamięć wewnątrz układu. Jest to pamięć typu EPROM zapisywana za pomocą programatora przez użytkownika.

Układy 8035, 8039, 8031 i 8032 nie posiadają pamięci programu wewnątrz układu. Wymagają dołączenia z zewnątrz układów pamięci ROM lub EPROM.

Typ procesora	Wewnętrzna pamięć programu	Wewnętrzna pamięć RAM	Zewnętrzna pamięć programu	Zewnętr pamię RAM
8048	1 k ROM	64	3 k	256
8748	1 k EPROM	64	3 k	256
8035	-	64	4 k	256
8049	2 k ROM	128	2 k	256
8749	2 k EPROM	128	2 k	256
8039	-	128	4 k	256
8051	4 k ROM	128	60 k	64 k
8751	4 k EPROM	128	60 k	64 k
8031	-	128	64 k	64 k
8052	8 k ROM	256	56 k	64 k
8752	8 k EPROM	256	56 k	64 k
8032	-	256	64 k	64 k

A.1.2. Linie wejścia - wyjścia.

Układy rodziny MCS-48 mają 27 linii, które mogą być użyte dla funkcji wejścia - wyjścia. Linie te są zgrupowane w 3 porty: Port 1, Port 2 i Bus (magistrala), po 8 linii każdy, które służą jako porty wejściowe, wyjściowe lub dwukierunkowe i 3 wejścia testowe, które mogą być sprawdzane programowo przez instrukcje skoków warunkowych (powodować przejście przez procesor do wykonywania innej procedury).

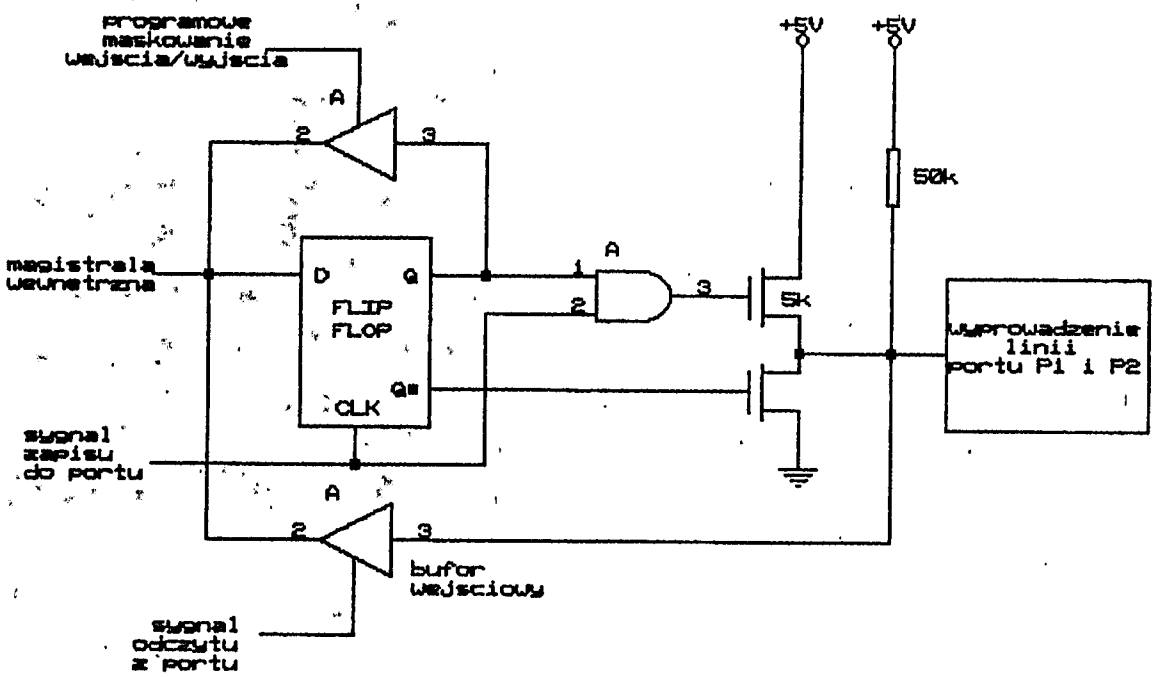
Porty 1 i 2 są jednakowe. Dane do tych portów są wpisywane przez procesor statycznie (zatrzaskiwane) i pozostają niezmienione aż do powtórnego zapisu. Jako porty wejściowe

nie są one zatrzaskiwane i dane muszą być utrzymane na liniach aż do momentu odczytania.

Jako wejścia linie portów spełniają wymagania standardu TTL, jako wyjście mogą sterować tylko jednym dołączonym wejściem TTL.

Porty 1 i 2 są określane jako quasi-dwukierunkowe, ze względu na budowę obwodu wyjściowego (patrz rys 2), która umożliwia odczytanie dannych wejściowych w czasie, gdy do portu wpisane są statycznie same jedynki - czyli wszystkie linie ustawione są w stan wysoki.

Bus jest 8-bitowym portem w pełni dwukierunkowym, zapisywanym strobami oddzielnymi dla wejścia i dla wyjścia. Jeżeli dwukierunkowość taka nie jest potrzebna Bus może służyć albo jako statycznie zapisywany port wyjściowy, albo nie zatrzaskiwany port wejściowy. Jako wejścia lub wyjścia mogą być wykorzystywane pojedyncze linie portu. Linie portu Bus poza czasem zapisu i odczytu pozostają w stanie wysokiej impedancji. W mikroprocesorach 8035, 8039 linie Bus stanowią multipleksowaną szynę danych/adresów do komunikacji z pamięcią zewnętrzną programu.



Rys. 2. Schemat pojedynczej linii portu.

Układ posiada trzy linie testowe: T0, T1 i INT.

T0 jest linią testową,której stan może warunkować skok w programie (instrukcja JT0 - jump if T0 - sprawdź poziom linii T0 i skocz, jeśli jest 1 oraz instrukcja JNT0 - skocz jeśli jest 0). T0 może być używany jako linia wyjściowa zegara.

T1 jest linią testową sprawdzaną podobnie instrukcjami JT1 i JNT1. Może być użyta jako linia wejściowa do zliczania za pomocą licznika wewnętrznego mikroprocesora impulsów pochodzących ze źródła zewnętrznego.

INT - linia przerwań - przy włączonych programowo przerwaniach testowana przez procesor w każdym cyklu maszynowym. Przy wyłączonych przerwaniach może być testowana instrukcją JNI podobnie jak dwie pozostałe linie.

A.1.3. System przerwań.

Przerwanie do mikroprocesora zgłaszane jest przez ustawienie stanu 0 na linii wejściowej INT. Przerwanie jest wyzwalane poziomem. Aktywnym stanem jest zero, co umożliwia dołączenie do jednej linii wielu urządzeń zgłaszających przerwania, które następnie mogą być rozróżniane przez procedurę obsługi przerwania. Linia przerwań jest testowana w każdym cyklu maszynowym i wykrycie stanu 0 powoduje skok do komórki pamięci programu o adresie 003, gdzie znajduje się pierwsza instrukcja procedury obsługi przerwania. Po zakończeniu procedury obsługi przerwania następuje powrót do poprzednio wykonywanego programu w miejscu, w którym nastąpiła przerwa w jego wykonywaniu.

System przerwań jest jednopoziomowy, tzn. podczas

wykonywania procedury obsługi przerwania wszystkie następne zgłoszenia przerwań są ignorowane. Ostatnia instrukcja procedury obsługi przerwania, którą zawsze jest RETR wznawia możliwość przyjmowania przerwań. Dotyczy to również przerwań wewnętrznych generowanych przez opisany dalej układ licznika/timera, które także nie mogą być obsłużone w czasie trwania procedury obsługi przerwania zewnętrznego.

Jeżeli występuje potrzeba realizacji dwu przerwań zewnętrznych, może zostać w tym celu wykorzystane wejście testujące T1. Wymaga to programowego włączenia licznika (instrukcja ENCTI) po wpisaniu do niego wartości FFH (binarnie: 11111111). Pojedynczy impuls na wejściu zliczającym T1 spowoduje przepełnienie licznika - skasowanie go do 0 i zgłoszenie przerwania wewnętrznego, którego procedura obsługi rozpoczyna się w komórce pamięci programu o adresie 007.

Zgłoszenie żądania, czyli poziom niski na linii INT musi być usunięte przed końcem obsługi przerwania. W przeciwnym przypadku zostanie ono potraktowane jako kolejne zgłoszenie przerwania i ponownie obsłużone.

A.1.4. Układ licznika/timera.

Rodzina procesorów MCS-48 zawiera wbudowany licznik, który umożliwia zliczanie zdarzeń zewnętrznych, a także odliczanie zadanych odcinków czasu bez ingerencji w wykonywanie programu przez procesor. W obu przypadkach wykorzystywany jest proces zliczania, inne jest tylko źródło zliczanych impulsów.

Licznik jest rejestrem 8-bitowym, zapisywanym i odczytywanym

przez program. Włączenie i wyłączenie pracy licznika oraz określenie jej trybu: licznik czy timer następuje również za pomocą instrukcji programu.

Zwiększenie o 1 maksymalnej zawartości licznika (11111111) powoduje wyzerowanie go (00000000), i wygenerowanie przerwania wewnętrznego. Fakt żądania przerwania jest zapamiętywany. Przerwanie jest obsługiwane, gdy nie ma zgłoszenia przerwania na linii INT. W przeciwnym przypadku przerwanie zewnętrzne zostaje obsłużone w pierwszej kolejności, a przerwanie wewnętrzne dopiero po nim.

W trybie pracy licznika do układu dołączone zostaje wejście T1. Kolejne przejścia 1->0 na linii T1 zwiększają o 1 (inkrementują) zawartość licznika. Maksymalna częstotliwość z jaką zmiany na wejściu T1 mogą być odczytywane wynosi 3 cykle instrukcji, czyli 7.5 μs przy częstotliwości generatora kwarcowego wynoszącej 6MHz. Poziom wysoki musi być stabilny przez co najmniej 100ns.

W trybie pracy timera do wejścia licznika dołączony jest wewnętrzny sygnał zegarowy (6MHz : 15) o częstotliwości 400 kHz. Sygnał ten jest następnie dzielony w układzie prescalera przez 32, co daje sygnał o f=12.5kHz, czyli T=80μs. 8-bitowy licznik jest inkrementowany co 80μs co pozwala odmierzać odcinki czasu od 80μs (przy wstępnym ustawieniu zawartości licznika na FFH) do 256x 80μs czli ok. 20ms (przy ustawieniu licznika na 0). Odcinki czasu dłuższe niż 20ms mogą być odmierzane poprzez zarezerwowanie na ten cel komórki pamięci danych i zliczanie w niej liczby odcinków 20-milisekundowych.

A.1.5. Zegar mikroprocesora.

Mikroprocesory rodziny MCS-48 posiadają wejścia X1 i X2, do których należy dołączyć rezonaror kwarcowy. Zakres częstotliwości rezonatora dopuszczalnych dla tych mikroprocesorów wynosi 1-6MHz., Przy częstotliwości rezonatora 6MHz czas wykonania 1 rozkazu (instrukcji) mikroprocesora wynosi 2.5μs, a dla instrukcji podwójnej długości 5μs.

A.2. MIKROKOMPUTERY RODZINY MCS-51

Rodzina mikrokomputerów MCS-51 stanowi kontynuację linii rozwojowej mikrokomputerów rodziny MCS-48. Są to mikrokomputery również 8-bitowe i posiadają w porównaniu z rodziną MCS-48:

- dwukrotnie więcej pamięci wewnętrznej RAM (128 lub 256 bajtów),
- możliwość zaadresowania 64k bajtów pamięci programu oraz 64k bajtów zewnętrznej paięci danych RAM,
- 2 (3) 16-bitowe liczniki/timery,
- 2 wejścia przerwań zewnętrznych,
- wbudowany układ transmisji szeregowej.

A.2.1. Linie wejścia/wyjścia.

Mikrokomputery MCS-51 posiadają 4 porty wejścia-wyjścia P0 - P3. Wszystkie porty są dwukierunkowe. Porty P0 i P2 są wykorzystywane do komunikacji z pamięcią zewnętrzną, co jest szczególnie istotne w przypadku mikroprocesora 8031, nie

posiadającego pamięci programu wewnątrz układu wymagającego dołaczenia jej z zewnątrz. Budowa obwodć wyjściowych linii portów jest analogiczna jak mikrokomputerze 8048.

A.2.2. System przerwań.

System przerwań jest bardziej złożony, niż w rodzini MCS-48. Przerwania mogą pochodzić z dwóch źródeł zewnętrznych (wejścia INT0 i INT1), od dwóch (trzech dla 8052 i 8032) liczników/timerów oraz z układu transmisji szeregowej. Każde z tych źródeł przerwań może być programowo wyłączone lub włączone. Programować można również hierarchię priorytetów tych przerwań przerwanie o wyższym priorytecie może przerwać obsługę przerwania o niższym priorytecie).

A.2.3. Układ licznika/timera.

Mikroprocesor 8051 ma 2 16-bitowe liczniki, które mogą zostać użyte jako 2 liczniki 8-bitowe i 1 16 bitowy, przy czym jeden z liczników 8-bitowych może zliczać jedynie impulsy wewnętrznego zegara mikroprocesora (funkcja timera - odliczanie odcinków czasu), natomiast drugi z nich może być użyty jako licznik impulsów zewnętrznych dołączonych do wejścia T0 i może zliczać maksymalnie 2^8=256 impulsów zewnętrznych. Licznik 16-bitowy pełni w tym modzie pracy liczników funkcję timera, mogąc odliczać odcinki czasu od 1μs (przy typowej częstotliwości rezonatora kwarcowego dla tego typu mikroprocesorów wynoszącej 12MHz) do 2^{16}x1μs=65.536ms.

Mikroprocesor 8032 posiada dodatkowo trzeci licznik

16-bitowy, mogący pracować zarówno jako timer jak i licznik impulsów.

A.2.4. Układ transmisji szeregowej.

Mikroprocesor 8031 posiada 1 wejście (RXD) i 1 wyjście (TXD) służące do transmisji danych szeregowych. Układ transmisji może pracować w jednym z czterech modów:

- w modzie 0 linia RXD stanowi linię danych, natomiast linia TXD jest linią zegara transmisji. Transmitowanych jest 8 bitów (najmniej znaczący jako pierwszy). Częstotliwość transmisji wynosi 1/12 częstotliwości kwarcu (1MHz).
- w modzie 1 dane są nadawane przez TXD, a odbierane przez RXD. Jednostka transmisji składa się z bitu startu (0), 8 bitów danych i bitu stopu (1). Do generacji zegara transmisji może być użyty jeden z układów timera.
- w modach 2 i 3 transmitowanych jest 11 bitów: bit startu (0), 8 bitów danych, programowany bit 9 oraz bit stopu (1). W modzie 2 częstotliwość transmisji wynosi 1/32 lub 1/64 częstotliwości kwarcu, a w modzie 3 jest generowana przez układ timera.

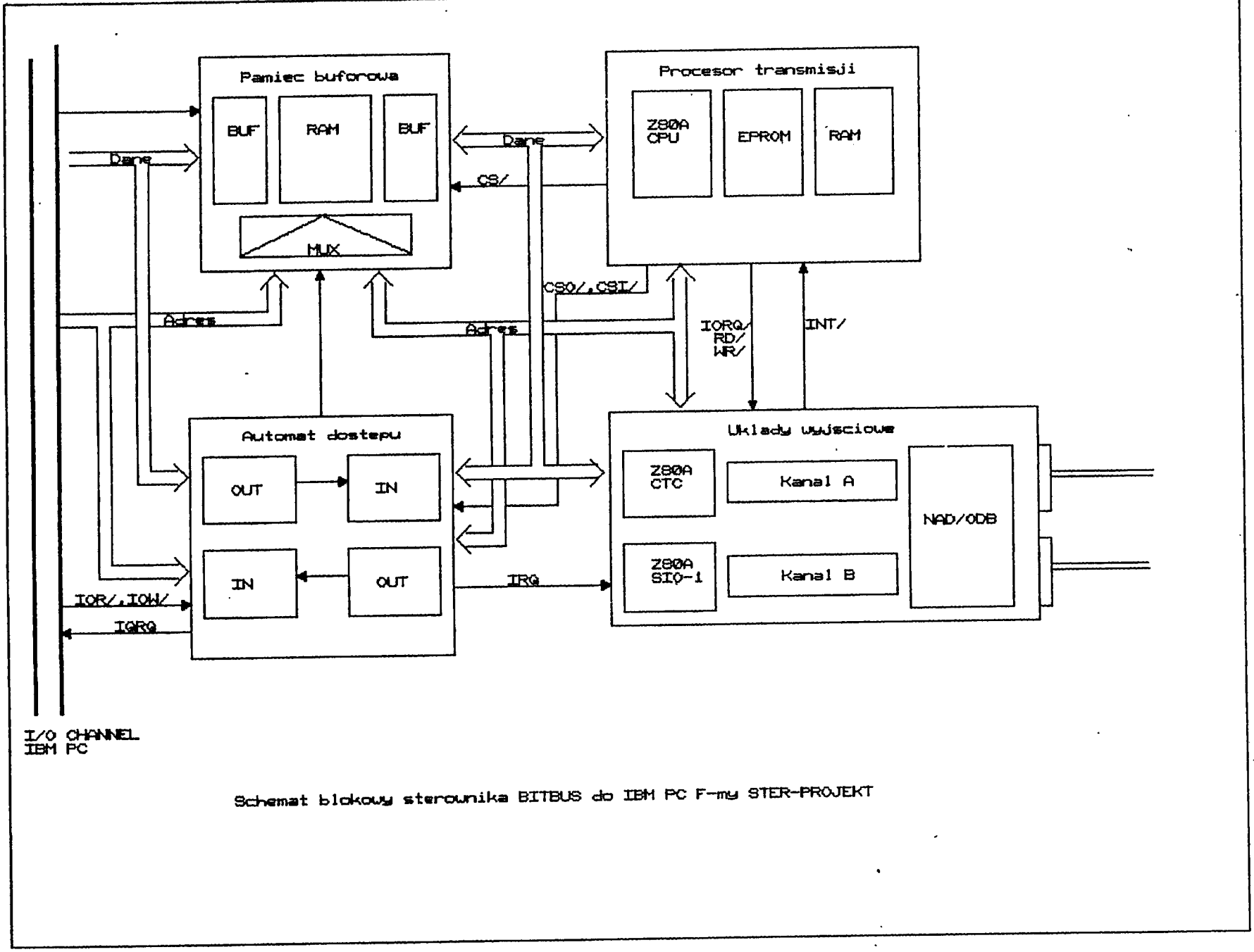
A.2.5. Zegar mikroprocesora.

Typowo stosowane są rezonatory kwarcowe o częstotliwości drgań 12 MHz.

DODATEK B

STEROWNIK TRANSMJI SZEREGOWEJ BITBUS

Moduł sterownika transmisji szeregowej BITBUS jest niezależnych systemem mikrokomputerowym, sterowanym przez procesor Z80A, którego zadaniem jest realizacja pełnego protokołu przesyłania danych do innego sterownika pracującego zgodnie z tym standardem. Dane do przesłania umieszczone zostają w pamięci buforowej RAM dostępnej również dla procesora (lub kanału DMA) komputera IBM PC/XT. Od strony komputera IBM PC moduł widziany jest jako 8 KB pamięci, do której wpisywane lub z której odbierane są dane oraz adres odbiorcy, oraz port wejściowy i wyjściowy pozwalający na przełączanie pamięci i przekazywanie sygnałów do procesora Z80A.



Schemat blokowy sterownika BITBUS do IBM PC F-my STER-PROJEKT

Rozmieszczenie zasobów w przestrzeni adresowej procesora 8088 jest następujące:

- buforowa pamięć RAM - A000:0000 do A000:3FFF
- port wyjściowy - 39F

  bit B0- linia informacyjna do procesora Z80A

  bit B1 - linia informacyjna do procesora Z80A

  bit B2 - 1 = zgłoszenie przerwania do procesora Z80A

  bit B3 - 1 = żądanie przydziału pamięci buforowej dla IBM PC

  bit B4 - 0 = blokowanie przerwania od procesora Z80A

  bit B5 - 0 = blokowanie przerwania od przydzielenia pamięci buforowej.

- port wejściowy- 39F

  bit B1 - 1 = przydzielona pamięć buforowa dl procesora IBM PC

  bit B2 - 0 = aktywne przerwanie przydzielenia pamięci buforowej

  bit B3 - 0 = aktywne przerwanie od procesora Z80A

  bit B4 - 1 = żądanie dostępu do pamięci dla procesora Z80A

  bit B5 - 1 = zgłoszenie przerwania odprocesora Z80A

  bit B6 - linia informacyjna od procesora Z80A

  bit B7 - linia informacyjna od procesora Z80A

Procesor Z80A ma do dyspozycji następujące zasoby:

- pamięć EPROM zawierająca program obsługi protokołu transmisji o pojemności do 8 KB,
- pamięć lokalna RAMo pojemności 8 KB przeznaczona do przechowywania danych wysyłanych i odbieranych oraz

zmiennych roboczych,

- pamięć buforowa RAM do przekazywania informacji pomiędzy procesorem Z80A i IBM PC,
- układ transmisji szeregowej Z80A-SIO realizujacy zamianę informacji równoleglej na szeregową w dwóch kanałach
- układ zależności czasowych Z80A-CTC odmierzający okresy czasowe, określający prędkość pracy kanałów szeregowych oraz przekazujący przerwania od procesora IBM PC/XT.
- układy wejścia - wyjścia sterujące przydziałem pamięci buforowej.

Zasoby procesora Z80A zlokalizowane są w przestrzeni adresowej procesora w sposób następujący:

Pamięć:

0000-1FFFH (2000-3FFFH) - pamięć EPROM

4000-5FFFH (6000-7FFFH) - lokalna pamięć RAM

8000-9FFFH (A000-BFFFH) - buforowa pamięć RAM

C000-FFFFH - obszar nie wykorzystywany

Wejście-wyjście

00-03H (04-0FH) - układ CTC

00H - licznik 0 - zgłaszanie przerwań co 1.024 ms

01H - licznik 1 - prędkość transmisji kanału A

02H - licznik 2 - prędkość transmisji kanału B

03H - licznik 3 - zgłaszanie przerwań od IBM PC i przydziału pamięci buforowej

10-13H (14-1FH) - układ SIO

10H - rejestr danych kanału A

11H - rejestr danych kanału B

12H - rejestr statusu kanału A

13H - rejestr statusu kanału B

20H (21-2FH) - bufor wyjściowy

B0 - linia komunikacyjna do IBM PC

B1 - linia komunikacyjna do IBM PC

B2 - 1 = zgłoszenie przerwania do IBM PC

B3 - 1 = żądanie dostępu do pamięci RAM

B4 - 0 = blokowanie przerwania od IBM PC

B5 - 0 = blokowanie przerwania od przydzielenia pamięci.

30H (31-3FH) - bufor wejściowy

B0 - linia komunikacyjna od IBM PC

B1 - linia komunikacyjna od IBM PC

B2 - 1 = zgłoszenie przerwania od IBM PC

B3 - 1 = żądanie dostępu do pamięci przez IBM PC

B4 - 0 = aktywne przerwanie od IBM PC

B5 - 0 = aktywne przerwanie od przydziału dostępu do pamięci

B6 - 1 = przydzielona pamięć buforowa dla Z80A

Parametry techniczne

Pakiet sterownika transmisji szeregowej BITBUS charakteryzuje się następującymi parametrami technicznymi:

- dwa kanały transmisji szeregowej
- szybkość przesyłania informacji od 2400 do 62 500 Kbitów/s
- długość linii przesyłowej do 2500 m
- możliwość dołączenia do 64 urządzeń do linii
- autonomiczna obsługa transmisji przez procesor Z80A
- komunikacja z IBM PC przez pamięć buforową

DODATEK C

OPIS PROTOKOŁU BITBUS

SPIS TREŚCI

1. WSTĘP

Protokół komunikacji BITBUS oparty jest na niskiej w kosztach szeregowej szynie sterującej, dostosowany do bardzo szybkich transmisji krótkich komunikatów sterujących pomiędzy zadaniami w systemie hierarchicznym.

Protokół BITBUS definiuja cztery warstwy: warstwa połączeń elektrycznych (ang. electrical interface), warstwa sterowania łącza (ang. data link protocol), warstwa sterowania przesyłem wiadomości (ang. message protocol) oraz warstwa aplikacyjna.

W systemie zostaną zrealizowane dwie najniższe warstwy protokołu określające wymagania na sieć oraz definiujące sposób komunikowania się węzłów w systemie hierarchicznym.

2. Warstwa połączeń elektrycznych

Warstwa połączeń elektrycznych protokołu BITBUS oparta jest na standardzie łącza RS. Możliwa jest praca w dwóch trybach: synchronicznym i samosynchronizującym.

Trybu synchronicznego używa się do transmisji z duża prędkością, przy czym maksymalna odległość między węzłami sieci wynosi 30 metrów, a liczba węzłów nie może przekroczyć 28. Szybkość transmisji wynosi od 500 kbitów/s do 2.4 Mbita/s. W trybie synchronicznym przesyła się dwie pary sygnałów różnicowych: dane oraz sygnał zegara.

W trybie przesyłania samosynchronizującego używa się tylko jednej pary sygnałów różnicowych dla danych kodowanych w

systemie NRZI z automatycznym wstawianiem zera po pięciu kolejnych jedynkach. Umożliwia to synchronizację zegara odbiorczego najrzadziej raz na pięć przesyłanych bitów. Szybkość transmisji wynosi 375 kbitów/s w segmentach linii o długości 300 m lub z szybkością 62.5 kbitów/s w segmentach do 1200 m. Maksymalna liczba węzłów wynosi 28. Poszczególne segmenty linii można sprzęgać ze sobą za pomocą retransmiterów. Maksymalna liczba węzłów wynosi wtedy 250, a długość sieci do kilku tysięcy metrów.

3. Protokół sterowania łącza

Protokół sterowania łącza BITBUS oparty jest na podzbiorze standardu SDLC (Synchronous Data Link Control), wypróbowanego i niezawodnego protokołu łączenia węzła nadrzędnego z wieloma węzłami podrzędnymi.

3.1 Charakterystyka protokołu transmisji BITBUS

3.1.1 Stan systemu

Podstawowym pojęciem wymagającym zdefiniowania w protokole BITBUS jest tzw. stan systemu.

Stacja nadrzędna ma pełną kontrolę nad swoim własnym stanem. Operacje wykonywane przez stacje podrzędne nie wymagają od nich wiedzy o stacji nadrzędnej. Z tego wynika, że stacja nadrzędna nie musi przekazywać stacjom podrzędnym wiadomości o swoim stanie.

Stan stacji podrzędnej musi być znany stacji nadrzędnej przy każdej operacji. Jednak z powodu fizycznej separacji obydwu stacji nie jest możliwa dokładna znajomość stanu stacji

podrzędnej przez cały czas (np. z powodu lokalnej inicjacji stacji podrzędnej, zaniku zasilania itp.). Wobec tego konieczne było zdefiniowanie mechanizmu przekazywania stanu stacji podrzędnych do nadrzędnej. Stacja nadrzędna zachowuje ostatni znany stan stacji każdej ze stacji podrzędnych, wykorzystuje go i sprawdza w każdej transmisji. Po wykryciu ewentualnej niezgodności ze stanem zakładanym podejmowane są odpowiednie działania. Na stan stacji podrzędnej składają się dwa elementy: tryb pracy i bieżące numery ramek.

Stacja podrzędna pracuje zawsze w jednym z dwóch trybów: w trybie odłączenia (NDM - Normal Disconnect Mode) lub w trybie odpowiedzi normalnej (NRM - Normal Response Mode).

Stacja podrzędna wchodzi w tryb NDM po lokalnej inicjacji lub po wykryciu niekorygowalnego błędu protokołu. W trybie tym oczekuje ona od stacji nadrzędnej rozkazu przejścia w tryb odpowiedzi normalnej. W trybie NDM stacja podrzędna nie może wymieniać informacji z nadrzędną.

W tryb NRM stacja podrzędna wchodzi po otrzymaniu specjalnego rozkazu od stacji nadrzędnej. Oprócz wejścia w nowy tryb stacja podrzędna synchronizuje się wtedy ze stacją nadrzędną, ustawiając bieżące numery ramek na 0. W trybie NRM stacja podrzędna może wymieniać informacje z nadrzędną tak długo, jak długo utrzymywana jest synchronizacja (tzn. dopóki nie ma błędu numeracji ramek).

Bieżące numery ramek wykorzystywane są w trybie NRM przez stację nadrzędną i każdą ze stacji podrzędnych, w celu zapobieżenia zagubieniu lub powtórzeniu ramek.

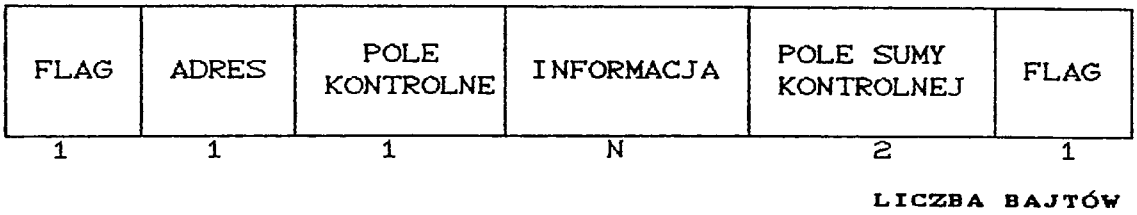
Zliczanie ramek dokonywane jest przez każdą stację na dwóch

parach 3-bitowych numerów ramek. Każda stacja podrzędna przechowuje numer odbiorczy (NR - Number Received) i numer nadawczy (NS - Number Send), zaś stacja nadrzędna przechowuje odpowiednie pary numerów dla każdej stacji podrzędnej, z którą się komunikuje. Jeśli numery ramek są prawidłowe, stacja podrzędna jest zsynchronizowana z nadrzędną. Bieżące numery ramek są elementem stanu stacji podrzędnej. NR określa numer następnej ramki przychodzącej, NS określa numer ramki oczekującej na wysłanie.

W czasie każdej transmisji w trybie NRM, numery te (NR i NS) są przesyłane i sprawdzane. Mogą przy tym zaistnieć trzy sytuacje: numeracja prawidłowa oraz błąd numeracji korygowalny i niekorygowalny. W przypadku błędu niekorygowalnego stacja nadrzędna musi ponownie zsynchronizować się ze stacją podrzędną przez przestawienie jej najpierw w tryb NDM, a potem ponownie w tryb NRM. Numery NR i NS są kolejnymi liczbami naturalnymi obliczanymi modulo 8.

3.1.2 Format ramki

Protokół transmisji jest odpowiedzialny za tworzenie ramek przesyłanych przez łącze BITBUS. Wszystkie ramki mają format przedstawiony na rys. 1. Każde z pól ramki składa się z jednego lub więcej bajtów (plus ewentualne "wtrącane" po każdych pięciu jedynkach zera). W bajcie, jako pierwszy transmitowany jest bit najmniej znaczący.



Rys. 1. Format ramki.

Pola flagów

Pola flagów są używane do "ograniczenia" ramki. Oba pola (jedno na początku, drugie na końcu ramki) tworzone są przez unikalną sekwencję bitów 01111110. Unikalność tej sekwencji jest zagwarantowana przez "wtrącanie" zera po każdych pięciu jedynkach, we wszystkich polach które nie są flagami. Pola flagów muszą wystąpić w każdej ramce.

Pole adresu

Pole adresu zawiera adres stacji podrzędnej uczestniczącej w transmisji. W ramce wysyłanej przez stację nadrzędną pole adresu identyfikuje jej odbiorcę, zaś w ramce wysyłanej przez stację podrzędną jej nadawcę. Jeśli zawartość pola adresu nie dotyczy żadnego urządzenia, ramka jest ignorowana. Pole adresowe ma 8 bitów długości i może zawierać wartości od 0 do 255. Wartości 0 i 251÷255 są zarezerwowane, pozostałe mogą być używane dowolnie. Pole adresu musi wystąpić w każdej ramce.

Pole kontrolne

Pole kontrolne używane jest do przekazywania polecenia i stanu między stacjami. Ma ono 8 bitów długości i pozwala realizować trzy typy operacji: synchronizację, nadzór i

przesyłanie informacji. Pole to musi być umieszczone we wszystkich ramkach.

a) Synchronizacja

Sekwencyjna transmisja danych między stacją nadrzędną a podrzędną jest synchronizowana przez stację nadrzędną. Proces synchronizacji jest realizowany za pomocą ramek nienumerowanych. Ramki nienumerowane nie wykorzystują bieżącej numeracji ramek w polu kontrolnym.

b) Nadzór

Do utrzymania synchronizacji stacji nadrzędnej z podrzędną potrzebna jest często wymiana wiadomości o stanie tych stacji, bez przekazywania sekwencji informacyjnej. Dokonuje się tego za pomocą ramek nadzorczych (tzn. ramek z nadzorczym polem kontrolnym). Ramki te są używane przez stację nadrzędną do odpytywania stacji podrzędnych i przez stacje podrzędne do potwierdzania odbioru prawidłowej ramki (tzn. z dobrym adresem i prawidłową sumą kontrolną CRC) od stacji nadrzędnej.

c) Przesyłanie informacji

Ramki informacyjne (z informacyjnym polem kontrolnym) są używane do przekazywania wiadomości. Ramki te można przesyłać tylko wtedy, gdy stacje uczestniczące w transmisji są zsynchronizowane ramkami kontrolnymi. Oprócz wiadomości, w ramkach informacyjnych przekazywane są te same informacje o stanie stacji, co w ramkach nadzorczych.

Pole informacyjne

Pole informacyjne używane jest do przekazywania treści

wiadomości. Pole to musi wystąpić w ramkach informacyjnych, a nie jest używane w ramkach nienumerowanych i nadzorczych.

Pole sumy kontrolnej

Pole sumy kontrolnej jest wykorzystywane do detekcji błędów na najniższym poziomie protokołu. Pole to zawiera 16-bitową sumę kontrolną CRC (ang. Cyclic Redundancy Check). Stacja transmitująca generuje i przesyła pole sumy kontrolnej, a stacja odbierająca sprawdza jego poprawność. Ramka z niepoprawną sumą kontrolną jest ignorowana. Suma kontrolna CRC jest generowana w standardzie CRC-CCITT wielomianem $x^{16}+x^{12}+x^{5}+1$. Pole to musi wystąpić w każdej ramce.

3.1.3 Budowa pola kontrolnego

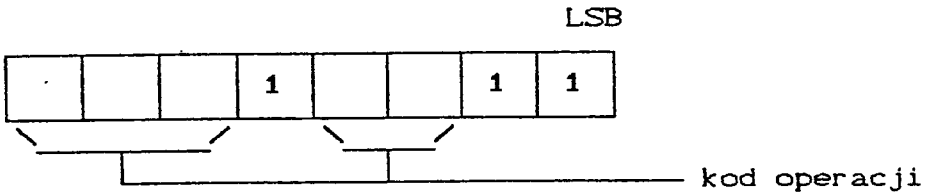
Operacje protokołu BITBUS wykorzystują trzy rodzaje pól kontrolnych: nienumerowane, nadzorcze i informacyjne. W porównaniu z SDLC, BITBUS wykorzystuje podzbiór zdefiniowanych pól kontrolnych z ustawionym na jedynkę bitem żądania/zakończenia. Wynika z tego, że stacja nadrzędna po przesłaniu ramki zawsze oczekuje na odpowiedzi od stacji podrzędnej (zapalony bit żądania) oraz, że po każdej wysłanej przez stację podrzędną ramce, sterowanie linią przekazywane jest do stacji nadrzędnej (zapalony bit zakończenia).

Ramki nienumerowane

Ramki nienumerowane są wykorzystywane przez BITBUS do synchronizacji stacji podrzędnych z nadrzędną.

Postać pola kontrolnego ramki nienumerowanej przedstawiono

na rysunku 2.



Rys. 2. Postać pola kontrolnego ramki nienumerowanej

Funkcje ramek nienumerowanych

Protokół BITBUS zawiera dwa rozkazy nienumerowane (SNRM i DISC) i dwie nienumerowane odpowiedzi (UA i FRMR). Pole kontrolne ramek nienumerowanych wygląda następująco:

SNRM	93H
DISC	53H
UA	73H
FRMR	97H

Odrzucenie ramki (FRMR - Frame Reject) jest wysyłane przez stację podrzędną do nadrzędnej po wykryciu ramki z nielegalnym polem kontrolnym (przy prawidłowych pozostałych elementach ramki). Ramka FRMR jest wysyłana jako odpowiedź na: każdą ramkę nienumerowaną w trybie NRM, każdą nadzorczą lub informacyjną ramkę o polu kontrolnym zawierającym niekorygowalny błąd numeracji. Po odebraniu ramki FRMR stacja nadrzędna powinna zainicjować sekwencję rozkazów ponownie synchronizującą stację podrzędną.

Potwierdzenie (UA - Unnumbered Acknowledge) jest wykorzystywane przez stację podrzędną dla potwierdzenia odbioru prawidłowej ramki nienumerowanej w trybie NDM.

Odłączenie (DISC - Disconnect) jest wysyłane przez stację

nadrzędną do podrzędnej w celu inicjacji synchronizacji. Rozkaz DISC powoduje przejście (lub pozostanie) stacji podrzędnej w tryb NDM. Stacja nadrzędna wykorzystuje ten rozkaz, gdy wykryje konieczność resynchronizacji (np. po inicjacji, wykryciu niekorygowalnego błędu numeracji itp.) lub jako odpowiedź na ramkę FRMR ze stacji podrzędnej. W przypadku, gdy stacja nadrzędna otrzyma od podrzędnej odpowiedź UA na wysłaną ramkę DISC, oznacza to, że stacja podrzędna jest w trybie NDM.

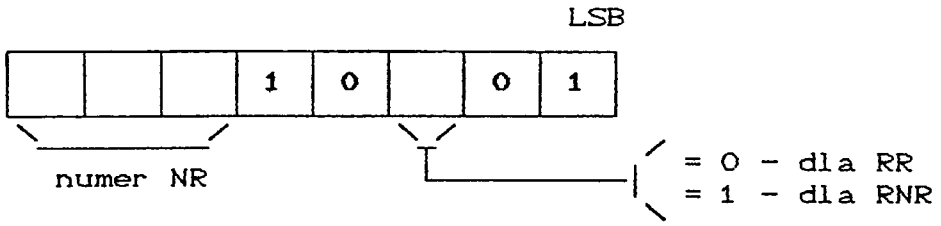
Ustawienie trybu odpowiedzi normalnej (SNRM - Set Normal Response Mode) wysyłane jest ze stacji nadrzędnej w celu synchronizacji stacji podrzędnej.

Gdy stacja podrzędna jest w trybie NDM, rozkaz SNRM powoduje jej przejście w tryb NRM, pozwalający na wymianę informacji między stacjami. Jeśli stacja podrzędna jest w trybie NRM rozkaz SNRM jest nielegalny.

Ramki nadzorcze

Ramki nadzorcze są wykorzystywane przez BITBUS do przekazywania stanu między stacjami w trybie NRM. Stacja nadrzędna używa ich do przepytywania stacji podrzędnych, a stacje podrzędne do potwierdzania prawidłowych ramek stacji nadrzędnej.

Postać pola kontrolnego ramki nadzorczej przedstawiono na rysunku 3.



Rys 3. Postać pola kontrolnego ramki nadzorczej

Funkcje ramek nadzorczych

BITBUS wykorzystuje dwie ramki nadzorcze: odbiornik gotowy (RR - Receiver Ready) i odbiornik nie gotowy (RNR - Receiver Not Ready).

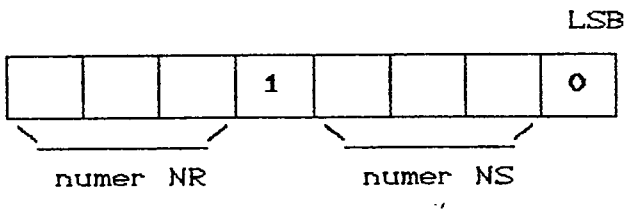
Stacja nadrzędna wykorzystuje ramkę nadzorczą RR do przepytywania stacji podrzędnej o ramki informacyjne. Stacja podrzędna używa ramki RR do potwierdzania prawidłowego (tzn. z prawidłowym adresem, dobrym CRC, przy wolnym buforze odbiorczym) odbioru ramki nadzorczej lub informacyjnej wtedy, gdy sama nie ma w buforze gotowej wiadomości dla stacji nadrzędnej (gdy ma tę wiadomość wysyła ramkę informacyjną).

Ramka nadzorcza NRM jest wykorzystywana w charakterze wiadomości o tym, że bufor jest aktualnie nie gotowy do odbioru prawidłowej ramki. Stacja nadrzędna używa ramek RNR do odczytania ze stacji podrzędnej jej stanu. Stacja podrzędna nie może odpowiedzieć na ramkę RNR ramką informacyjną. Stacja podrzędna wykorzystuję ramkę RNR jako zawiadomienie, że ostatnia ramka do niej skierowana była prawidłowa, ale nia została przyjęta ze względu na zapełnienie bufora. W tym przypadku stacja nadrzędna musi retransmitować ramkę.

Obie ramki RR i RNR zawierają numer NR, potwierdzający odebranie NR-1 ramek, tzn. NR oznacza numer następnej, oczekiwanej ramki. Odbiornik porównuje tę wartość z przechowywanym numerem NS. W przypadku, gdy wszystkie wiadomości były potwierdzone nadchodzący numer NR powinien być równy numerowi NS+1. Jeśli jest to prawdą, wówczas poprzednia ramka jest potwierdzona i numer NS jest zwiększany o 1, a wysłana wiadomość może być usunięta z bufora. Natomiast gdy po wysłanej ramce informacyjnej przychodzący numer NR jest równy numerowi NS, oznacza to, że ramka ta nie została przyjęta i powinna być wysłana powtórnie. Każda inna wartość numeru NR jest niekorygowalnym błędem numeracji zmuszającym stację nadrzędna do resynchronizacji z podrzędną.

Ramki informacyjne

Ramki informacyjne wykorzystywane są przez protokół BITBUS do przesyłania wiadomości. Są to jedyne ramki zawierające pole informacyjne.



Rys. 4. Postać pola kontrolnego ramki informacyjnej

Postać pola kontrolnego ramki informacyjnej przedstawiono na rysunku 4. Pole to zawiera numer NR dla potwierdzenia odbioru NR-1 ramek i numer NS dotyczący wysyłanej ramki informacyjnej.

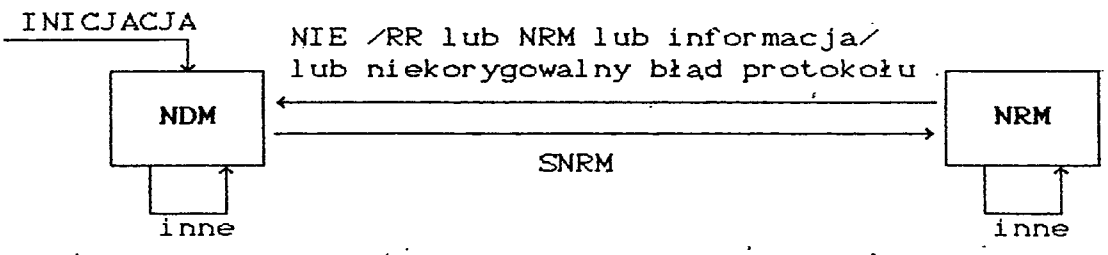
Funkcje ramki informacyjnej

Ramki informacyjne mogą być uważane za specjalny rodzaj ramek nadzorczych. Przenoszą one informacje o potwierdzeniu ramek przez numer NR w taki sposób, jak ramki nadzorcze. Numer ten jest zawsze porównywany z numerem odbiorczym NS, jak dla ramek nadzorczych. Dodatkowo, oprócz numeru NR, ramki informacyjne przenoszą wiadomość w polu informacyjnym i odpowiedni numer NS. Po otrzymaniu ramki informacyjnej odbiornik porównuje odebrany numer NS z własnym numerem NR. Jeżeli numery są równe, wiadomość może być przyjęta. Jeśli przychodzący numer NS jest równy numerowi odbiorczemu NR-1, stacja odbierająca może przyjąć ramkę, traktując ją jako powtórzenie poprzednio nadanej. Natomiast, gdy przychodzący numer NS nie jest równy ani NR, ani NR-1, stanowi to niekorygowalny błąd protokołu. Przypadek taki powoduje podjęcie przez stację nadrzędną akcji resynchronizacyjnej przed następnymi próbami transmisji.

3.1.4 Struktura przepływu informacji w protokole BITBUS

Stan stacji podrzędnej

Graf przejść między trybami w stacji podrzędnej przedstawiono na rysunku 5.



Rys. 5. Graf przejść pomiędzy trybami w stacji podrzędnej

Graf ten ma tylko dwa stany: NRM i NDM. Przejścia pomiędzy stanami , w zależności od przychodzących ramek przedstawione są w tabelce. (Wymienione w tabelce przejścia dotyczą oczywiście prawidłowych ramek.)

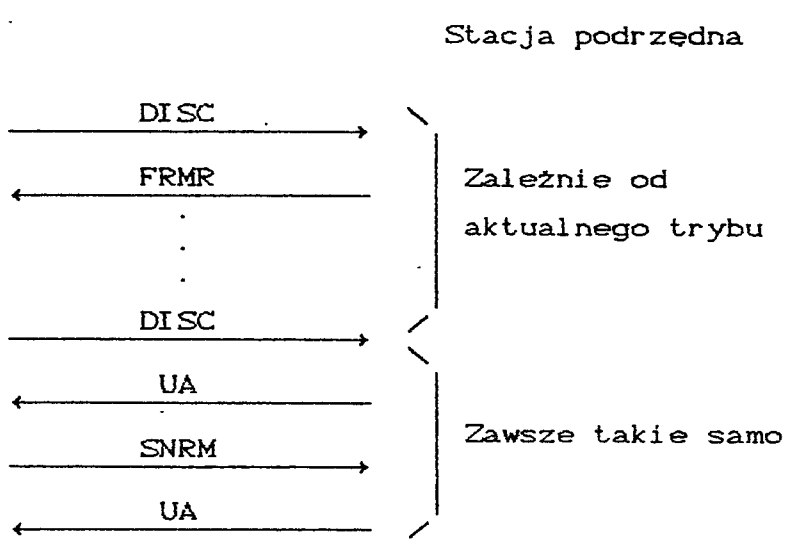
Stan	Ramka przychodząca	Stan następny	Odpowiedź
NDM	DISC	NDM	UA
NDM	SNRM	NRM	UA
NDM	inna	NDM	FRMR
NRM	I /informacyjna/	NRM	RR,RNR,I
NRM	RR	NRM	RR,RNR,I
NRM	RNR	NRM	RR,RNR
NRM	I,RR,RNR z błędem numeracji	NDM	FRMR
NRM	inna	NDM	FRMR

Sekwencja synchronizacyjna

Po inicjacji stacja nadrzędna musi zsynchronizować się ze wszystkimi stacjami podrzędnymi z którymi chce się komunikować. Synchronizacja odbywa się za pomocą ramki nienumerowanej. Synchronizacja konieczna jest również w przypadku wykrycia przez stację nadrzędną lub podrzędną niekorygowalnego błędu protokołu. Sekwencja synchronizacyjna jest wykorzystywana do ustawienia początkowego stanu stacji podrzędnej (tzn. trybu NRM z numerami NR i NS równymi 0). Rysunek 6 przedstawia typową sekwencję synchronizacyjną.

Stacja nadrzędna inicjuje sekwencję synchronizacyjną przez wysłanie ramki DISC. Odpowiedzia stacji podrzędnej jest ramka FRMR w trybie NRM lub ramka UA w trybie NDM. Stacja nadrzędna kontynuuje wysyłanie ramek DISC do czasu uzyskania odpowiedzi UA informującej o przejściu stacji podrzędnej w tryb NDM. Wtedy stacja nadrzędna kończy sekwencję

synchronizacyjną, wysyłając ramkę SNRM. Po otrzymaniu ramki SNRM w trybie NDM stacja podrzędna przechodzi w tryb NRM i odpowiada ramką UA. Po jej odebraniu stacja nadrzędna wie, że stacja podrzędna jest zsynchronizowana.



Rys. 6. Typowa sekwencja synchronizacyjna

Stacja podrzędna może też zainicjować sekwencję synchronizacyjną wysyłając ramkę FRMR w odpowiedzi na przychodzącą ramkę. Stacja nadrzędna odpowiada na ramkę FRMR, ramką DISC i dalsze operacje są identyczne jak wyżej opisane.

3.2 Błędy transmisji

Przekroczenie czasu oczekiwania

Błąd przekroczenia czasu oczekiwania na odpowiedź (time-out) zachodzi wtedy, gdy stacja nadrzędna nie otrzyma odpowiedzi na wysłaną ramkę w czasie 10 milisekund. Czas ten jest liczony od chwili zakończenia transmisji do chwili odebrania odpowiedzi i obliczenia CRC. Błąd ten może wystąpić wówczas, gdy stacja podrzędna o podanym adresie nie istnieje, gdy

stacja podrzędna zignorowała ramkę z powodu błędu CRC, lub gdy błąd CRC został wykryty przez stację nadrzędną w odpowiedzi. We wszystkich tych przypadkach stacja nadrzędna podejmuje ponowną próbę transmisji. Dwie nieudane próby transmisji traktowane są jako niekorygowalny błąd protokołu.

Błędne pole kontrolne

Błędne pole kontrolne wykryte przez jedną ze stacji traktowane jest jako niekorygowalny błąd protokołu. Wykrycie tego błędu przez stację podrzędną powoduje jej przejście w tryb NDM i wysłanie ramki FRMR (Frame Reject), natomiast wykrycie tego błędu przez stację nadrzędną powoduje przyjęcie przez nią założenia, że stacja podrzędna jest w trybie NDM, co wymaga resynchronizacji.

Błąd numeracji ramek

Błędy numeracji ramek są w niektórych przypadkach korygowalne. Jeżeli nie można takiego błędu skorygować traktowany jest on identycznie jak błąd przekroczenia czasu i błąd pola kontrolnego.

ZAŁĄCZNIK Nr.3

Tue Dec 12 1989 15:40

Listening programu obsługi modelu miernika stacjonarnego

2500 A.D. 8051 Macro Assembler - Version 4.00j

Input Filename : tama_bb.asm
Output Filename : tama_bb.obj

```
                        SYMBOLS
                        CODE
                ;*****************************************************
                ;*****************************************************
                ;
                ;           GLOWNA PETLA PROGRAMU OBSLUGI MODULU POMIA
                ;*****************************************************
                ;*****************************************************
0000            ;
0000 C2 96              CLR     P1.6            ; WYLACZENIE NADAJNIKA
0002 D2 95              SETB    P1.5            ; WLACZENIE ODBIORNIKA
0004 80 03              SJMP    START
                ;
                ; SKOK DO OBSLUGI PRZERWANIA ZEGAROWEGO
                ;
000B                    ORG     0BH
000B 02 03 5A           LJMP    Tim0_Int
                ;
000E 75 D0 00   START:  MOV     PSW,#0
0011 75 81 42           MOV     SP,#stack
0014 75 26 01           MOV     IDDEV,#1D       ; NUMER MODULU
                ;
0017 31 00      MAIN:   ACALL   RCCEIV          ; ODBIOR RAMKI OD MODULU N
0019 E5 20              MOV     A,FRAME
001B B4 10 3B           CJNE    A,#FRI,ERRFR    ; RAMKA INNA NIZ ZLECENIE
001E B9 03 38           CJNE    R1,#3,ERRFR     ; BLEDNA ILOSC BAJTOW RAMK
0021 E5 21              MOV     A,BUFOR
0023 B4 FF 0A           CJNE    A,#0FFH,POM     ; POMIAR
                ;TEST MODULU
0026 75 70 10           MOV     FRAME,#FR1
0029 79 01              MOV     R1,#1           ; DLUGOSC RAMKI
002B 75 21 00           MOV     BUFOR,#0        ; WYNIK TESTU
002E 80 [illegible]     SJMP    SNDFR
                ;
0030 54 78      POM:    ANL     A,#78H
0032 70 25              JNZ     ERRFR           ; BLEDNY NUMER CZUJNIKA
```

Tue Dec 12 1989 15:40

```
                                   ;
                                   ; POPRAWNA RAMKA
 0034     75 20 11                         MOV     FRAME,#FRR
 0037     79 00                            MOV     R1,#0
 0039     11 60                            ACALL   SEND                 ; ODPOWIEDZ - PRZYJECIE ZL
                                   ;
 003B     ?? 21 2A                         MOV     arg,BUFOR            ; NUMER CZUJNIKA
 003E     31 E2                            ACALL   pomiar               ; POMIAR CZUJNIKA
                                   ;
 0040     31 0D                            ACALL   RECEIV               ; ODBIOR PYTANIA O DANE
 0042     E5 20                            MOV     A,FRAME
 0044     B4 11 12                         CJNE    A,#FRR,ERRFR         ; RAMKA NIE RR
 0047     B9 0? 0F                         CJNE    R1,#2,ERRFR          ; BLEDNA ILOSC BAJTOW
                                   ;
                                   ; POPRAWNA RAMKA
 004A     75 20 10                         MOV     FRAME,#FRI
 004D     79 02                            MOV     R1,#2                ; DWA BAJTY ODPOWIEDZI
 004F     85 28 21                         MOV     BUFOR,result         ; ODPOWIEDZ
 0052     85 29 22                         MOV     BUFOR+1,result+1
                                   ;
 0055     11 60                    SNDFR:  ACALL   SEND                 ; WYSLANIE ODPOWIEDZI
 0057     80 DF                            SJMP    MAIN
                                   ;
                                   ;
                                   ; BLEDNA RAMKA
 0059                              ERRFR:
                                   ;       CJNE    A,#FRR,ERRFR1        ; RAMKA NIE RR
                                   ;       MOV     FRAME,#FRR
                                   ;       MOV     R1,#0
                                   ;       SJMP    SNDFR
                                   ;
                                   ;ERRFR1
 0059     75 20 97                         MOV     FRAME,#FFRMR
 005C     79 00                            MOV     R1,#0
 005E     80 F5                            SJMP    SNDFR
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;************************************************
                                   ;************************************************
                                   ;
                                   ;             PROCEDURA OBSLUGI PROTOKOLU HDLC  -  PROCE
                                   ;
                                   ;************************************************
                                   ;************************************************
                                   ;
                                   ;
                                   ;************************************************
                                   ;
                                   ;
                                   ; PARAMETRY TRANSMISJI
                                   ;
          0014                     KWANT   EQU     20                   ; CZAS IMPULSU ZEGARA W SE
          0018                     SPEED   EQU     24                   ; PREDKOSC TRANSMISJI W SE
          00D0                     TBIT    EQU     100000/(SPEED*KWANT)     ; DLUGOSC BITU W I
          001E                     CHTIME  EQU     600/KWANT            ; CZAS OCZEKIWANIA NA ZMIA
                                   ;
```

Tue Dec 12 1989 15:40

```
101                             ;
102                             ; KODY RAMEK HDLC
103                             ;
104             0011            FRR     EQU     11H
105             0010            FRI     EQU     10H
106             0097            FFRMR   EQU     97H
107             0073            FUA     EQU     73H
108                             ;
109                             ;
110                             ; UZYWANE ZMIENNE
111                             ;
112             0001            ID      EQU     1               ; NUMER PAKIETU
113                             ;
114                             ;*****************************************************************
115                             ;*****************************************************************
116                             ;
117                             ;
118     0000                            DATA
119                             ;
120     0020                            ORG     20H
121                             ;
122     0020                    FRAME:  .DS     1
123     0021                    BUFOR:  .DS     4
124     0025                    STATUS: .DS     1
125     0026                    IDDLV:  .DS     1
126                             ;
127     0060                            CODE
128                             ;*****************************************************************
129                             ;
130                             ; PROCEDURY TRANSMISJI UZYWAJA NASTEPUJACYCH ZMIENNYCH :
131                             ;   R0 - ADRES BUFORA
132                             ;   R1 - LICZNIK BAJTOW
133                             ;   R2 - LICZNIK BITOW W BAJCIE
134                             ;   R3 - ODCZYTANY (ZAPISYWANY) BIT
135                             ;   R4 - AKTUALNIE ODCZYTYWANY (ZAPISYWANY) BAJT
136                             ;   R5 - LICZNIK KOLEJNYCH JEDYNEK
137                             ;   R6   GENERATOE CRC - STARSZY BAJT
138                             ;   R7 - GENERATOR CRC - MLODSZY BAJT
139                             ;*****************************************************************
140                             ;
141                             ;
142                             ;*****************************************************************
143                             ;*****************************************************************
144                             ;
145                             ;                       NADAWANIE
146                             ;
147                             ;*****************************************************************
148                             ;*****************************************************************
149                             ;
150                             ;
151     0060    C2 8C           SEND:   CLR     TR0
152     0062    C2 8D                   CLR     TF0
153     0064    75 D0 00                MOV     PSW,#0          ; BANK REJESTROW 0
154     0067    75 89 01                MOV     TMOD,#1         ; TIMER 0 MOD 1 ,
155     006A    78 21                   MOV     R0,#BUFOR       ; ADRES BUFORA DANYCH
156     006C    7B 80                   MOV     R3,#80H         ; WYSYLANY BIT - 0
157     006E    7D 05                   MOV     R5,#5           ; LICZNIK JEDYNEK
```

Tue Dec 12 1989 15:10

```
158   0070   7A 08          MOV     R2,#8             ; LICZNIK BITOW
159   0072   7C 00          MOV     R4,#0             ; WYSYLANY BAJT
160   0074   7E FF          MOV     R6,#0FFH          ; INICJACJA GENERATORA CC
161   0076   7F FF          MOV     R7,#0FFH
162   0078   75 25 BF       MOV     STATUS,#0BFH      ; BAJT STANU
163   007B   D2 96          SETB    P1.6              ; WLACZENIE NADAJNIKA
164                        ;
165                        ; WLACZENIE NADAJNIKA
166   007D   C2 8C          SNDBIT: CLR     TR0
167   007F   75 8A 3B       MOV     TL0,#<(11-BIT)    ; POPRAWKA - 11 TIKOW ZEGA
168   0082   75 8C FF       MOV     TH0,#>(11-BIT)    ; CZAS OCZEKIWANIA NA NAST
169   0085   D2 8C          SETB    TR0
170   0087   C2 8D          CLR     TF0               ; FLAG ZEGARA
171   0089   EB             MOV     A,R3
172   008A   B2 90          XRL     P1,A              ; USTAWIENIE LINII NADAWZ
173                        ; PRZYGOTOWANIE NASTEPNEGO BITU
174   008C   20 2F 1C       JB      STATUS.7,SNDBI0   ; NADAWANIE FLAGU
175   008F   ED             MOV     A,R5
176   0090   60 0C          JZ      SNDBI2            ; BYLO NADANE 5 JEDYNEK -
177   0092   EC             MOV     A,R4              ; CY=BIT DO WYSLANIA
178   0093   13             RRC     A
179   0094   20 2E 02       JB      STATUS.6,CRCBJT   ; WYSYLANE BAJTY CRC
180   0097   31 C5          ACALL   CRCBIT            ; OBLICZENIE CRC
181   0099   EC             CRCBJT: MOV     A,R4
182   009A   13             RRC     A
183   009B   FC             MOV     R4,A
184   009C   40 06          JC      SNDBI7            ; JEDYNKA
185                        ; ZERO
186   009E   7D 05          SNDBI2: MOV     R5,#5             ; LICZNIK JEDYNEK=5
187   00A0   7B 80          MOV     R3,#80H           ; WARTOSC R3 DLA BITU 0
188   00A2   80 10          SJMP    SNDBI3
189                        ;
190                        ; JEDYNKA
191   00A4   7B 00          SNDBI7: MOV     R3,#0             ; WARTOSC R3 DLA BITU 1 -
192   00A6   DD 0B          DJNZ    R5,SNDBI3
193   00A8   0A             INC     R2                ; WTRACENIE ZERA - ZWIEKSZ
194   00A9   80 09          SJMP    SNDBI3
195                        ;
196                        ; FLAG
197   00AB   EC             SNDBI0: MOV     A,R4              ; CY=BIT
198   00AC   13             RRC     A
199   00AD   FC             MOV     R4,A
200   00AE   7B 00          MOV     R3,#0             ; WARTOSC R3 DLA 1
201   00B0   40 02          JC      SNDBI3
202   00B2   7B 80          MOV     R3,#80H           ; WARTOSC R3 DLA 0
203   00B4   DA 52          SNDBI3: DJNZ    R2,SBIT
204                        ; KONIEC BAJTU - PRZYGOTOWANIE NASTEPNEGO
205   00B6   7A 08          MOV     R2,#8             ; LICZNIK BITOW
206   00B8   20 26 16       JB      STATUS.0,FLP      ; NIE NADANY FLAG POCZATKO
207   00BB   20 29 1A       JB      STATUS.1,SNDBJ2   ; NIE NADANY ADRES
208   00BE   20 2A 1E       JB      STATUS.2,SNDBJ3   ; NIE NADANE POLE STERUJAC
209   00C1   20 2B 22       JB      STATUS.3,SNDBJ4   ; NIE NADANE DANE I 1-SZY
210   00C4   20 2C 32       JB      STATUS.4,SNDBJ6   ; NIE NADANY 2-GI BAJT CRC
211   00C7   20 2D 37       JB      STATUS.5,SNDBJ7   ; NIE NADANY FLAG KONCOWY
212   00CA   C2 96          CLR     P1.6              ; WYLACZNIE NADAJNIKA
213   00CC   C2 8C          CLR     TR0
214   00CE   C2 8D          CLR     TF0
```

Tue Dec 12 1989 15:40

```
272
273
274   0100   C2 8C          
275   0102   C2 8D          
276   0104   75 00 00       
277   0107   75 89 01       
278   010A   75 25 07       
279   010D   7D 07          
280   010F   AB B0          
281   0111   21 6E          
282
283
284   0113   C2 8C          
285   0115   75 8A 62       
286   0118   75 8C FF       
287   011B   D2 8C          
288   011D   C2 8D          
289   011F   C3             
290   0120   AB B0          
291   0122   ED             
292   0123   7D 07          
293   0125   64 02          
294   0127   60 03          
295   0129   64 03          
296   012B   70 1A          
297
298
299   012D   E5 25          
300   012F   64 00 19       
301   0132   C3             
302   0133   54 FE          
303   0135   60 44          
304   0137   7A 00          
305   0139   C3             
306   013A   31 99          
307   013C   BE F0 3C       
308   013F   BF B8 39       
309   0142   C2 8C          
310   0144   C2 8D          
311   0146   22             
312
313
314   0147   30 20 02       
315   014A   31 99          
316
317
318   014C   30 8D FD       
319
320   014F   C2 8C          
321   0151   75 8A C4       
322   0154   75 8C FF       
323   0157   D2 8C          
324   0159   C2 8D          
325   015B   E5 BC          
326   015D   08             
327   015E   20 E3 91       
328
```

```
;*****************************************************************
;
RECEIV: CLR     TR0
        CLR     TF0
        MOV     PSW,#0          ; BANK REJESTROW 0
        MOV     TMOD,#1         ; TIMER 0 MOD 1
        MOV     STATUS,#7       ; BAJT STANU
        MOV     R5,#7           ; LICZNIK JEDYNEK
        MOV     R3,P3
        AJMP    WAITB           ; OCZEKIWANIE NA ZMIANE
;
; JEST ZMIANA - BIT = 0
BIT0:   CLR     TR0
        MOV     TL0,#<(20+CHTIME-BIT)   ; POPRAWKA 20 TIKO
        MOV     TH0,#>(20+CHTIME-BIT)   ; CZAS OCZEKIWANIA
        SETB    TR0
        CLR     TF0             ; FLAG ZEGARA
        CLR     C               ; CY = BIT = 0
        MOV     R3,P3           ; ZAPAMIETANIE STANU LINII
        MOV     A,R5
        MOV     R5,#7           ; LICZNIK JEDYNEK
        XRL     A,#2
        JZ      NXTBIT          ; LICZNIK = 2 - WTRACONE 0
        XRL     A,#3            ; 2 XOR 1
        JNZ     RCVX2           ; LICZNIK <> 1 - "NORMALNE"
;
; FLAG
        MOV     A,STATUS
        CJNE    A,#0,RCV3       ; FLAG POCZATKOWY
        MOV     A,R1
        ANL     A,#0FEH
        JZ      RCV3            ; ZA KROTKA RAMKA
        MOV     R7,#0           ; USTAWINIE R2 NA WARTOC
        CLR     C               ; BIT 0
        ACALL   RCVBIT          ; REJESTRACJA BITU
        CJNE    R6,#0F0H,RCV3   ; NIEPRAWIDLOWE CRC
        CJNE    R7,#0B8H,RCV3   ; NIEPOPRAWNE CRC
        CLR     TR0
        CLR     TF0
        RET                     ; POPRAWNA RAMKA
;
RCVX2.  JB      STATUS.0,NXTBIT ; FLAG POCZATKOWY NIE BYL
        ACALL   RCVBIT          ; ODBIOR BITU
;
; OCZEKIWANIE NA NASTEPNY BIT
NXTBIT: JNB     TF0,NXTBIT
;
        CLR     TR0
        MOV     TL0,#<(-2*CHTIME)
        MOV     TH0,#>(-2*CHTIME)       ; CZAS OCZEKIWANIA
        SETB    TR0
        CLR     TF0             ; FLAG ZEGARA
        MOV     A,P3            ; ODCZYT LINII
        XRL     A,R3
        JB      A.3,RECEIV      ; NIEREJESTROWANA ZMIANA W
```

Tue Dec 12 1989 15:40

```
329
330   016C   C5 B0
331   0170   6B
332   0171   20 E3 AC
333   0174   30 00 F7
334
335
336   0177   C2 8C
337   0179   75 8A 7F
338   017C   75 8C FF
339   017F   D2 8C
340   0181   C2 B0
341   0183   D;
342   0184   DD CF
343
344
345   0186   21 0D
346
347
348   0188   78 21
349   018A   79 00
350   018C   7C 00
351   018E   7E FF
352   0190   7F FF
353   0192   75 25 06
354   0195   7A 08
355   0197   21 59
356
357
358
359
360
361   0199   EC
362   019A   13
363   019B   FC
364   019C   30 E9 07
365   019F   31 C5
366   01A1   DA 13
367
368
369   01A3   7A 08
370   01A5   20 29 0F
371   01A8   20 2A 14
372
373   01AB   B8 25 04
374
375   01AE   75 25 07
376   01B1   22
377
378   01B2   A6 04
379   01B4   08
380   01B5   09
381   01B6   22
382
383
384
385   01B7   EC
```

```
; LINIA NIEZMIENIONA - OCZEKIWANIE NA ZMIANE
WAITB.  MOV     A,P3
        XRL     A,R3
        JB      A.3,BIT0        ; ZMIANA W LINII - BIT = 0
        JNB     TF0,WAITB
;
; KONIEC CZASU - BIT = 1
        CLR     TR0
        MOV     TL0,#-(19+2*CHTIME-BIT) ; POPRAWKA 22 TCXI
        MOV     TH0,#-(19+2*CHTIME-BIT) ; CZAS OCZEKIWANIA
        SETB    TR0
        CLR     TF0             ; FLAG ZEGARA
        SETB    C               ; CY = ODEBRANY BIT
        DJNZ    R5,RCVX2        ; ZMNIEJSZENIE LICZNIKA JE
;
; 7 KOLEJNYCH JEDYNEK - BRAK SYNCHRONIZACJI
        AJMP    RECEIV
;
; INICJACJA STRUKTUR PO ODEBRANIU FLAGU POCZATKOWEGO
RCV3.   MOV     R0,#BUFOR
        MOV     R1,#0
        MOV     R4,#0
        MOV     R6,#0FFH        ; INICJACJA GENERATORA CRC
        MOV     R7,#0FFH
        MOV     STATUS,#6
        MOV     R2,#8
        AJMP    NXTBIT
;
;*****************************************************
;
; PODPROGRAM ODBIORU BITU
;
RCVBIT: MOV     A,R4            ; REJESTRACJA BITU
        RRC     A
        MOV     R4,A
        JB      STATUS.1,NOTADR ; CRC LICZONE Z PRZESUNIĘC
        ACALL   CRCBIT          ; OBLICZENIE CRC
NOTADR: DJNZ    R2,OMIT
;
; KONIEC BAJTU
        MOV     R2,#8
        JB      STATUS.1,DECBJ1 ; NIE ODEBRANY ADRES
        JB      STATUS.2,DECBJ2 ; NIE ODEBRANE POLE KONTRO
; ZAPISANIE BAJTU DO BUFORA ODBIORCZEGO
        CJNE    R0,#(BUFOR+4),OKB       ; NIE KONIEC BUFOR
; ZA DLUGA RAMKA
NADR:   MOV     STATUS,#7
        RET
;
OKB.    MOV     @R0,R4
        INC     R0
        INC     R1
OMIT.   RET
;
;
; SPRAWDZENIE ADRESU RAMKI
DECBJ1: MOV     A,R4
```

Tue Dec 12 1989 15:40

```
386   01BC   B5 26 F3
387
388   01BB   75 25 04
389   01BE   22
390
391   01BF   75 25 00
392   01C2   8C 20
393   01C4   22
394
395
396
397
398
399
400
401   01C5   40 04
402
403   01C7   EE
404   01C8   13
405   01C9   FE
406   01CA   EF
407   01CB   13
408   01CC   FF
409   01CD   40 0A
410   01CF   22
411
412   01D0   C3
413   01D1   EC
414   01D2   13
415   01D3   FE
416   01D4   EF
417   01D5   13
418   01D6   FF
419   01D7   40 F6
420
421
422   01D9   63 06 84
423   01DC   63 07 08
424   01DF   22
425
426
427
428
429
430          0082
431          0093
432          0094
433   0027
434   0027
435   0028
436   002A
437
438   002B
439   002D
440   002F
441
442   30 1
```

```
        CJNE    A,IDDEV,NADR    ; BLEDNY ADRES
; PRAWIDLOWY ADRES
        MOV     STATUS,#4
        RET

DECODE: MOV     STATUS,#0       ; STATE=0
        MOV     FRAME,R4
        RET

;
;
;*****************************************************
;
; PODPROGRAM OBLICZENIA CRC DLA BITU
;       BIT WCHODZACY - FLAG CY
;
CRCBIT: JC      BIT1            ; WCHODZACY BIT 1
; WCHODZACY BIT 0
        MOV     A,R6            ; "WSUNIECIE" 0 NA NAJM.OD
        RRC     A
        MOV     R6,A
        MOV     A,R7
        RRC     A
        MOV     R7,A
        JC      NTEQ            ; PRZENIESIENIE = 1 (RO.IE
ENDCRC. RET                     ; KONIEC PROCEDURY
;
BIT1.   CLR     C
        MOV     A,R6            ; WSUNIECIE 0
        RRC     A
        MOV     R6,A
        MOV     A,R7
        RRC     A
        MOV     R7,A
        JC      ENDCRC          ; PRZENIESIENIE = 1 (ROWNE
;
; PRZENIESIENIE NIEROWNE BITOWI WCHODZACEMU
NTEQ:   XRL     R6,#84H
        XRL     R7,#8
        RET
;
;
;
;procedura pomiarowa tamy II
;bit definition
pom_inp.                reg p3.2        ;wejscie pomiar
pom_pob.        reg p1.3        ;pobudzenie
pom_en2:        reg p1.4        ;blokada odbiornika
        DATA
stat.           .ds 1
result:         .ds 2
arg             .ds 1

pom_a:          .ds 2
pom_b.          .ds 2
pom_c:          .ds 2

d.l_ob:         .ds 2
```

Tue Dec 12 1989 15:10

```
                                del_bc          ds 2
                                del_de          .ds 2

                                abs_ab          .ds 2
                                abs_bc          .ds 2
                                abs_ca          .ds 2

                                rob:            ds 2
                                delay           .ds 2           ;opoznienie pobudzenia syn

                                psw_rest:       .ds 1
                                stack           .ds 1

                                typ_bit         reg arg.7
                                t0_mode         equ 01h
                                rm0_mask        equ 0fh
                                nbr_mask        equ 07h
                                reg_psw_field   equ 10h
                                reg_bank_2:     equ 10h

                                cycle_time      equ 20          ;w setkach nanosekund
                                milisek         equ 50000/cycle_time
                                sto_mikros      equ 1000/cycle_time
                                time1           equ 2*sto_mikros        ;200 us - szer.po
                                time2           equ 25*milisek          ;opoznienie 1 pomi
                                time3           equ 25*milisek          ;opoznienie kol. p
                                wait_arif       equ sto_mikros
                                inert           equ 2*sto_mikros/5      ;inercja po zboczu
                                status_ok_rec_pkd.      equ 0
                                status_1_rejected.      equ 1
                                status_2_rejected.      equ 80h
                                status_time_out         equ 81h

                                        CODE

                                ;stala typu CODE

0100  0A 00                     t1_max  db 10,0

                                ;program

0102                            pomiar:
                                        setb ea                 ;zezwol. na przerwania
                                        mov psw_rest,psw
                                        mov a,psw
                                        anl a,#not reg_psw_field
                                        orl a,#reg_bank_2
                                        mov psw,a

                                        ;inicjalizacja timera
                                        mov a,tmod
                                        anl a,#tmr_mask
```

Tue Dec 12 1989 15:40

```
500    01F3    44 01          orl a,#t0_mode
501    01F5    F5 89          mov tmod,a
502    01F7    D2 94          setb pom_en2
503
504                           ustaw numer kanalu
505    01F9    85 2A F0       mov b,arg
506    01FC    53 F0 07       anl b,#nbr_mask
507    01FF    E5 90          mov a,p1
508    0201    54 F8          anl a,#.not. nbr_mask
509    0203    45 F0          orl a,b
510    0205    F5 90          mov p1,a                ;wybor kanalu
511
512    0207    30 57 47       jb type_bit,pomiar_t1   ;zaleznie od typu czujnika
513
514                           ; pobudzenie asynchroniczne
515
516    020A    C2 93          clr pom_pob
517    020C    74 14          mov a,#time1/5
518    020E    12 03 DF       call waitA5
519    0211    D2 93          setb pom_pob
520    0213    74 78          mov a,#time2/100
521    0215    12 03 D4       call waitA100
522    0218    12 04 4F       call ini_meas           ;wstepny pomiar okresu
523
524                           ; 1 pomiar
525
526                           call ph_impulse
527                           ; mov a,#time3/100
528                           call waitA100
529    021B    12 04 11       call meas_nt            ;wynik w AB
530    021E    78 2B          mov R0,#pom_a
531    0220    12 03 BB       call st_ab
532
533                           ;DATA
534    0223
535    0223    78 27          mov R0,#result
536    0225    12 03 BB       call st_ab
537    0228    75 27 00       mov stat,#status_3_accepted
538    022B    02 03 4C       jmp pom_end
539
540                           ; 2 pomiar
541
542    022E    12 03 E6       call ph_impulse
543    0231    74 7D          mov a,#time3/100
544    0233    12 03 D4       call waitA100
545    0236    12 04 11       call meas_nt
546    0239    78 2D          mov R0,#pom_b
547    023B    12 03 BB       call st_ab
548
549                           ; 3 pomiar
550
551    023E    12 03 E6       call ph_impulse
552    0241    74 7D          mov a,#time3/100
553    0243    12 03 D4       call waitA100
554    0246    12 04 11       call meas_nt
555    0249    78 2F          mov R0,#pom_c
556    024B    12 03 BB       call st_ab
```

Tue Dec 12 1989 15:40

```
557
558   021E   02 07 63                  jmp analiza
559
560   0251                        pomiar_t1:
561
562                               ;       1 pomiar
563
564   0251   12 04 11                  call meas_nt            ;wynik w AB
565   0254   12 03 BB                  call st_ab
566   0257
567                               ;       2 pomiar
568
569   0257   12 04 11                  call meas_nt
570   025A   12 03 BB                  call st_ab
571
572                               ;       3 pomiar
573
574   0250   12 04 11                  call meas_nt
575   0260   12 03 BB                  call st_ab
576
577
578
579
580
581                               ;       analiza wynikow
582   0263                        analiza.
583                               ;       obliczenie roznic
584   0263   78 2B                     mov R0,#pom_a
585   0265   79 2D                     mov R1,#pom_b
586   0267   12 03 86                  call ode_2
587   026A   78 31                     mov R0,#del_ab
588   026C   12 03 BB                  call st_ab
589
590   026F   78 2D                     mov R0,#pom_b
591   0271   79 2F                     mov R1,#pom_c
592   0273   12 03 86                  call ode_2
593   0276   78 33                     mov R0,#del_bc
594   0278   12 03 BB                  call st_ab
595
596   027B   78 2F                     mov R0,#pom_c
597   027D   79 2B                     mov R1,#pom_a
598   027F   12 03 86                  call ode_2
599   0282   78 35                     mov R0,#del_ca
600   0284   12 03 BB                  call st_ab
601
602                               ;       moduly roznic
603   0287   78 31                     mov R0,#del_ab
604   0289   12 03 A8                  call abs_2
605   028C   78 37                     mov R0,#abs_ab
606   028E   12 03 BB                  call st_ab
607
608   0291   78 33                     mov R0,#del_bc
609   0293   12 03 A8                  call abs_2
610   0296   78 39                     mov R0,#abs_bc
611   0298   12 03 BB                  call st_ab
612   029B
613   029D   78 35                     mov R0,#del_ca
```

Tue Dec 12 1989 15:40

```
614   029D   12 03 A8          call abs_2
615   02A0   78 3B             mov R0,#abs_ca
616   02A2   12 03 BB          call st_ab
617
618                       ;     if max (abs i) <= limit srednia z 3 pomiarow
619   02A5
620                       ;     liczenie max abs - w R2 offset max
621   02A5   7A 37              mov R2,#abs_ab
622   02A7   A8 02              mov R0,R2
623   02A9   79 39              mov R1,#abs_bc
624   02AB   12 03 86           call ode_2
625   02AE   30 E7 02           jnb ACC.7,comp_2        ;ab>bc
626   02B1   7A 39              mov R2,#abs_bc
627   02B3                  comp_2.
628   02B3   A8 02              mov R0,R2
629   02B5   79 3B              mov R1,#abs_ca
630   02B7   12 03 86           call ode_2
631   02BA   30 E7 02           jnb ACC.7,comp_3        ;max>ca
632   02BD   7A 3B              mov R2,#abs_ca
633   02BF                  comp_3:
634   02BF   A8 02              mov R0,R2
635   02C1   90 01 E0           mov DPTR,#del_max
636   02C4   12 03 90           call odec_2
637   02C7   30 E7 54           jnb ACC.7,srednia_3
638
639                       ;     odrzucic najgorszy lub wszystkie
640                       ;     if min (abs j)<= limit then
641                       ;     i =min (abs j); srednia = pom(i) - del(i)
642                       ;     else odrzucenie calego pomiaru
643
644                       ;     szukaj min(abs j) - R2 - numer min pomiaru
645   02CA   7A 37              mov R2,#abs_ab
646   02CC   A8 02              mov R0,R2
647   02CE   79 39              mov R1,#abs_bc
648   02D0   12 03 86           call ode_2
649   02D3   20 E7 02           jb ACC.7,comp_4         ;ab<bc
650   02D6   7A 39              mov R2,#abs_bc
651   02D8                  comp_4.
652   02D8   A8 02              mov R0,R2
653   02DA   79 3B              mov R1,#abs_ca
654   02DC   12 03 86           call ode_2
655   02DF   20 E7 02           jb ACC.7,comp_5         ;min<ca
656   02E2   7A 3B              mov R2,#abs_ca
657   02E4                  comp_5.
658   02E4   A8 02              mov R0,R2
659   02E6   90 01 E0           mov DPTR,#del_max
660   02E9   12 03 90           call odec_2
661   02EC   20 E7 24           jb ACC.7,meas_rejected  ;min>limit
662
663                       ;     srednia z dwoch
664   02EF   C3                 clr C
665   02F0   EA                 mov A,R2
666   02F1   94 37              subb A,#abs_ab          ;offset minimum
667   02F3   FA                 mov R2,A
668   02F4   24 31              add A,#del_ab           ;delta(min)
669   02F6   F8                 mov R0,A
670   02F7   12 01 C0           call shr_2              ;delta/2
```

```
671   02FA   78 3D          mov R0,#rob
672   02FC   12 03 BB       call st_ab
673   02FF   74 2B          mov a,#pom_a
674   0301   2A             add A,R2                ;pom(min)
675   0302   F8             mov R0,a
676   0303   79 3D          mov R1,#rob
677   0305   12 03 86       call ode_2              ;pom-delta/2
678   0308   78 28          mov R0,#result
679   030A   12 03 BB       call st_ab              ;srednia w pole wyniku
680   030D   75 27 01       mov stat,#status_1_rejected
681   0310   02 03 4C       jmp pom_end
682
683                        ,wszystkie odrzucone
684   0313                 meas_rejected:
685   0313   75 27 30             mov stat,#status_3_rejected
686   0316   E5 00                mov A,0
687   0318   85 00 F0             mov B,0
688   031B   02 03 4C             jmp pom_end
689                        ; srednia z trzech
690   031E                 srednia_3:
691                        ;            zakladamy jednobajtowa sume roznic
692                        ;            srednia = a + ((c-a)-(a-b))/3
693
694   031E   78 35                mov R0,#del_ca
695   0320   79 31                mov R1,#del_ab
696   0322   12 03 86             call ode_2
697   0325   E5 F0                mov A,B                 ;mlodsza czesc roznicy
698   0327   20 E7 0A             jb ACC.7,del_3_minus
699   032A   75 F0 03             mov B,#3
700   032D   84                   div AB
701   032E   75 F0 00             mov B,#0
702   0331   02 03 3F             jmp del_3
703   0334                 del_3_minus:
704   0334   F4                   cpl A
705   0335   04                   inc A                   ;-A
706   0336   75 F0 03             mov B,#3
707   0339   84                   div AB
708   033A   F4                   cpl A
709   033B   04                   inc A
710   033C   75 F0 FF             mov B,#0FFH             ; rozszerzenie wyniku
711   033F                 del_3:
712   033F   25 2B                add A,pom_a
713   0341   F5 28                mov result,A
714   0343   E5 2C                mov A,pom_a+1
715   0345   35 F0                addc A,B
716   0347   F5 29                mov result+1,A
717   0349   75 27 00             mov stat,#status_3_accepted
718   034C                 pom_end:
719   034C   85 41 D0             mov psw,psw_rest
720   034F   78 28                mov R0,#result
721   0351   12 03 C0             call shr_2
722   0354   12 03 BB             call st_ab
723   0357   C2 AF                clr EA
724   0359   22                   ret
725
726   035A                 tim0_int:
727   035A   C2 8C                clr TR0
```

Tue Dec 12 1989 15:40

```
728   035C   C2 0D          clr TF0
729   035E   D2 93          setb pom_pob
730   0360   D2 94          setb pom_en2
731   0362   D0 E0          pop a
732   0364   D0 E0          pop a                ;wyjscie z inta
733   0366   D0 E0          pop a
734   0368   D0 E0          pop a                ;wyjscie z procedury
735   036A   74 00          mov A,#0
736   036C   75 F0 00       mov B,#0
737   036F   78 28          mov R0,#result
738   0371   12 03 BB       call st_ab
739   0374   85 41 D0       mov psw,psw_rest
740   0377   C2 AF          clr EA
741   0379   75 27 01       mov stat,#status_time_out
742   037C   32             reti
743
744                     ;procedury
745
746   037D              dod_2:
747   037D   E7                mov A,@R1
748   037E   26                add A,@R0
749   037F   09                inc R1
750   0380   08                inc R0
751   0381   F5 F0             mov B,A
752   0383   E7                mov A,@R1
753   0384   36                addc A,@R0
754   0385   22                ret
755
756   0386              ode_2:
757   0386   C3                clr C
758   0387   E6                mov A,@R0
759   0388   97                subb A,@R1
760   0389   F5 F0             mov B,A
761   038B   08                inc R0
762   038C   09                inc R1
763   038D   E6                mov A,@R0
764   038E   97                subb A,@R1
765   038F   22                ret
766   0390              odec_2:
767   0390   C3                clr C
768   0391   E5 00             mov A,0
769   0393   93                movc A,@A+DPTR
770   0394   96                subb A,@R0
771   0395   F5 F0             mov B,A
772   0397   08                inc R0
773   0398   E5 01             mov A,1
774   039A   93                movc A,@A+DPTR
775   039B   96                subb A,@R0
776   039C   22                ret
777
778   039D              cmpl_2:
779   039D   E6                mov A,@R0
780   039E   F4                cpl A
781   039F   04                inc A
782   03A0   F5 F0             mov B,A
783   03A2   08                inc R0
784   03A3   E6                mov A,@R0
```

Tue Dec 12 1989 15:40

```
785   03A4   F4                cpl A
786   03A5   34 00             addc A,#0
787   03A7   22                ret
788
789   03A8                  abs_2:
790   03A8   08                inc R0
791   03A9   E6                mov A,@R0
792   03AA   18                dec R0
793   03AB   20 E7 03          jb ACC.7,abs0
794   03AE   A6 F0             mov @R0,b
795   03B0   22                ret
796   03B1                  abs0.
797   03B1   71 9D             call cmpl_2
798   03B3   22                ret
799
800   03B4                  copy_2:
801   03B4   E6                mov A,@R0
802   03B5   F7                mov @R1,A
803   03B6   08                inc R0
804   03B7   09                inc R1
805   03B8   E6                mov A,@R0
806   03B9   F7                mov @R1,A
807   03BA   22                ret
808
809   03BB                  st_ab:
810   03BB   A6 F0             mov @R0,B
811   03BD   08                inc R0
812   03BE   F6                mov @R0,A
813   03BF   22                ret
814
815   03C0                  shr_2:
816   03C0   08                inc R0
817   03C1   E6                mov A,@R0
818   03C2   18                dec R0
819   03C3   20 E7 0B          jb ACC.7,shr_minus
820   03C6   C3                clr C
821   03C7                  shr_0:
822   03C7   13                rrc A
823   03C8   C0 E0             push A
824   03CA   E6                mov A,@R0
825   03CB   13                rrc a
826   03CC   F5 F0             mov B,a
827   03CE   D0 E0             pop A
828   03D0   22                ret
829   03D1                  shr_minus.
830   03D1   D3                setb C
831   03D2   61 C7             jmp shr_0
832
833
834   03D4                  waitA100.
835   03D4   F5 F0             mov B,A
836   03D6                  wait2.
837   03D6   74 12             mov A,#18
838   03D8   12 03 DF          call waitA5
839   03DB   D5 F0 F8          djnz B,wait2
840   03DE   22                ret
841
```

Tue Dec 12 1989 15:40

```
842   03DF                    waitA5:
843   03DF   00                       nop
844   03E0   00                       nop
845   03E1   00                       nop
846   03E2   D5 E0 FA                 djnz a,waitA5
847   03E5   22                       ret
848
849
850   03E6                    ph_impulse:
851   03E6   C2 94                    clr pom_en2             ;enable input
852
853   03E8   C2 8D                    clr TF0                 ;zezwol. na przerwanie tim
854   03EA   75 8C 00                 mov TH0,#0
855   03ED   75 8A 00                 mov TL0,#0
856   03F0   D2 A9                    setb ET0
857   03F2   D2 8C                    setb TR0
858
859   03F4   30 B2 FD                 jnb pom_inp,$
860   03F7   20 B2 FD                 jb pom_inp,$            ;wykrywanie 1-0
861   03FA   D2 94                    setb pom_en2
862
863   03FC   C2 8C                    clr TR0
864   03FE   C2 A9                    clr ET0
865
866   0400   75 F0 3F                 mov B,#delay
867   0403   74 40                    mov A,#delay+1
868   0405   12 04 A0                 call wait_in_timer      ;pobudzenie w środku ujem
869
870                                   ;pobudzenie w fazie
871
872   0408   C2 93                    clr pom_pob
873   040A   74 14                    mov a,#time1/5
874   040C   71 DF                    call waitA5
875   040E   D2 93                    setb pom_pob
876
877   0410   22                       ret
878
879   0411                    meas_nt:
880
881                                   ,przygotowanie timera
882
883   0411   C2 8D                    clr tf0
884   0413   75 8A 00                 mov tl0,#0
885   0416   75 8C 00                 mov th0,#0
886   0419   7A 28                    mov r2,#2*cycle_time    ;licznik petli pomiarow
887   041B   D2 A9                    setb et0
888
889                                   ;fazowanie
890   041D   C2 94                    clr pom_en2
891   041F   30 B2 FD                 jnb pom_inp,$
892   0422   74 14                    mov A,#t_inert
893   0124   71 DF                    call waitA5
894   0426   20 B2 FD                 jb pom_inp,$
895   0429   D2 8C                    setb tr0                ;start zliczania
896
897
898   042B                    pomloop:
```

Tue Dec 12 1989 15:40

```
899   042B   74 14          mov A,#t_inert
900   042D   71 DF          call waitA5
901   042F   30 B2 FD       jnb pom_inp,$
902   0432   74 14          mov A,#t_inert
903   0434   71 DF          call waitA5
904   0436   20 B2 FD       jb pom_inp,$
905   0439   DA F0          djnz r2,pomloop          ;+24 cykle =4us
906
907   043B   C2 8C          clr tr0                  ;stop timer +12 cykli =3us
908   043D   C2 A9          clr ET0
909   043F   C2 8D          clr TF0
910   0441   D2 94          setb pom_en2
911
912   0443   C3             clr C
913   0444   E5 8A          mov A,TL0
914   0446   94 02          subb a,#2                ;poprawka na djnz
915   0448   F5 10          mov B,A                  ;1-czesc
916   044A   E5 8C          mov A,TH0
917   044C   94 00          subb a,#0
918
919                         wynik pomiaru w AB
920   044E   22             ret
921
922                   ini_meas:                      ;pomiar wstepny
923   044F   C2 8D          clr tf0
924   0451   75 8A 00       mov tl0,#0
925   0454   75 8C 00       mov th0,#0
926   0457   D2 A9          setb et0
927   0459   D2 8C          setb TR0
928
929                         ;tazowanie
930   045B   C2 94          clr pom_en2
931   045D   30 B2 FD       jnb pom_inp,$
932   0460   74 14          mov A,#t_inert
933   0462   71 DF          call waitA5
934   0464   20 B2 FD       jb pom_inp,$
935   0467   C2 8C          clr TR0
936   0469   C2 8D          clr tf0
937   046B   75 8A 00       mov tl0,#0
938   046E   75 8C 00       mov th0,#0
939   0471   D2 8C          setb TR0                 ;start zliczania
940   0473   74 14          mov A,#t_inert
941   0475   71 DF          call waitA5
942   0477   30 B2 FD       jnb pom_inp,$
943   047A   74 14          mov A,#t_inert
944   047C   71 DF          call waitA5
945   047E   20 B2 FD       jb pom_inp,$
946   0481   C2 8C          clr TR0                  ;stop zliczania
947   0483   C2 A9          clr ET0
948   0485   D2 94          setb pom_en2
949   0487   85 8A 3F       mov delay,TL0
950   048A   85 8C 40       mov delay+1,TH0
951                         delay=okres/4-width_half
952   048D   78 3F          mov R0,#delay
953   048F   71 C0          call shr_2
954   0491   71 C0          call shr_2
955   0493   C3             clr C
```

```
956   0494   E6          mov A,@R0
957   0495   94 41       subb A,#<(width_half+15)
958   0497   F5 3F       mov delay,A
959   0499   08          inc R0
960   049A   E6          mov A,@R0
961   049B   94 00       subb A,#>(width_half+15)
962   049D   F5 40       mov delay+1,A
963   049F   22          ret
964
965                 wait_ln_timer:          ;czekaj AB cykli zegara +15
966   04A0   C0 E0       push A
967   04A2   F5 F0       mov B,A
968   04A4   F4          cpl A
969   04A5   04          inc A
970   04A6   F5 8A       mov TL0,a
971   04A8   D0 E0       pop A
972   04AA   F4          cpl A
973   04AB   34 00       addc A,#0
974   04AD   F5 8C       mov TH0,A
975   04AF   D2 8C       setb TR0
976   04B1   30 8D FD    jnb TF0,$
977   04B4   C2 8D       clr TF0
978   04B6   C2 8C       clr TR0
979   04B8   22          ret
980
981   04B9               end
982
982
```

Tue Dec 12 1989 15:10

Defined	Symbol Name	Value	References
98	BIT	= 00D0	167 168 285 286 337
284	BIT0	0120	332
412	BIT1	01D0	401
Pre	BSECT	0000	
123	BUFOR	0021	34 39 50 61 62
99	CHTIME	= 001E	285 286 321 322
Pre	CODE	000B	3 22 127 479
401	CRCBIT	01C5	180 365
181	CRCBJT	0099	179
Pre	DATA	0020	118 120 433
385	DECBJ1	01B7	370
391	DECBJ2	01BF	371
410	ENDCRC	01CF	419
69	ERRFR	0059	32 33 43 55 56
106	FFRMR	0097	76
218	FLP	00D1	206
122	FRAME	0020	31 37 46 54 59
105	FRI	= 0010	32 37 59
104	FRR	= 0011	46 55
107	FUA	= 0073	
112	ID	= 0001	28
125	IDDEV	0026	28 224 386
96	KWANT	0014	98 99
30	MAIN	0017	65
375	NADR	01AE	386
366	NOTADR	01A1	364
422	NTEQ	01D9	409
318	NXTBIT	0159	294 314 318 355
378	OKB	01B2	373
381	OMIT	01B6	366
42	POM	0030	35
348	RCV3	0188	300 303 307 308
361	RCVBIT	0199	306 315
314	RCVX2	0154	296 342
274	RECEIV	010D	30 53 327 345
Pre	RSECT	0000	
262	SBIT	0108	203 220 225 230 241
151	SEND	0060	48 64
197	SNDBI0	00AB	174
186	SNDBI2	009E	176
203	SNDBI3	00B4	188 192 194 201
191	SNDBI7	00A4	184
166	SNDBIT	007D	263
223	SNDBJ2	00D8	207
228	SNDBJ3	00DF	208
233	SNDBJ4	00E6	209
244	SNDBJ5	00F1	235
251	SNDBJ6	00F9	210
258	SNDBJ7	0101	211
238	SNDBJI	00EB	233
64	SNDFR	0055	40 78
97	SPEED	0018	98
26	START	000E	17
124	STATUS	0025	162 174 179 206 2[illegible]
			228 244 251 258 278
			375 388 391

Tue Dec 12 1989 15:40

Defined	Symbol Name		Value	References				
330	WAITB		016E	281	333			
796	abs0		03B1	793				
789	abs_2		03A8	604	609	614		
446	abs_ab		0037	605	621	645	666	
447	abs_bc		0039	610	623	626	647	650
448	abs_ca		003B	615	629	632	653	6
582	analiza		0263	558				
436	arg		002A	50	459	505		
778	cmpl_2		039D	797				
627	comp_2		02B3	625				
633	comp_3		02BF	631				
651	comp_4		02D8	649				
657	comp_5		02E4	655				
800	copy_2		03B4					
466	cycle_time	=	0014	467	468	886		
711	del_3		033F	702				
703	del_3_minus		0334	698				
442	del_ab		0031	587	603	668	695	
443	del_bc		0033	593	608			
444	del_ca		0035	599	613	694		
484	del_max		01E0	635	659			
451	delay		003F	866	867	949	950	952
746	dod_2		037D					
981	end		04B9					
922	ini_meas		044F	522				
879	meas_ni		0411	529	545	554	564	569
684	meas_rejected		0313	661				
467	milisek	=	01F4	470	471			
462	nbr_mask	=	0007	506	508			
756	ode_2		0386	586	592	598	624	6
766	odec_2		0390	636	660			
850	ph_impulse		03E6	542	551			
138	pom_a		002B	530	584	597	673	7
439	pom_b		002D	546	585	590		
440	pom_c		002F	555	591	596		
432	pom_en2	=	0094	502	730	851	861	8
718	pom_end		034C	538	681	688		
430	pom_inp	=	00B2	859	860	891	894	901
431	pom_pob	=	0093	516	519	729	872	8
489	pomiar		01E2	51				
560	pomiar_tl		0251	512				
898	pomloop		042B	905				
453	psw_rest		0041	491	719	739		
464	reg_bank_2	=	0010	494				
463	reg_psw_field	=	0018	493				
435	result		0028	61	62	535	678	713
450	rob		003D	671	676			
821	shr_0		03C7	831				
815	shr_2		03C0	670	721	953	954	
829	shr_minus		03D1	819				
690	srednia_3		031E	637				
809	st_ab		03BB	531	536	547	556	565
				611	616	672	679	722
454	stack		0042	27				
434	stat		0027	537	680	685	717	741
475	status_1_rejected	=	0001	680				
474	status_3_accepted	=	0000	537	717			

Tue Dec 12 1989 15:40

Defined	Symbol Name		Value	References				
476	status_3_rejected	=	0080	685				
477	status_time_out	=	0081	741				
468	sto_mikros	=	0032	469	472	473		
460	t0_mode	=	0001	500				
473	t_inert	=	0014	892	899	902	937	940
726	tim0_int		035A	23				
469	time1	=	0064	517	873			
470	time2		30D4	520				
471	time3	=	30D4	543	552			
461	tm0_mask	=	000F	499				
459	type_bit	=	0057	512				
836	wait2		03D6	339				
834	waitA100		03D4	521	544	553		
842	waitA5		03DF	518	838	846	874	903
965	wait_in_timer		04A0	868				
472	width_half	=	0032	957	961			

Lines Assembled : 987

Assembly Errors : 0

Tue Jan 1 1980 09:51

ZAŁĄCZNIK Nr 4

Listening programu obsługi modelu miernika polowego

2500 A.D. 8051 Macro Assembler - Version 4.00j

Input Filename : tama_rs.asm
Output Filename : tama_rs.obj

```
                                        SYMBOLS
                                        DATA
0000                                    ORG     20H
0020    30 30 30 30 30          AWP:    .BYTE   '00000'
0025                            AWLF:           .ds 3
        000D                    cr:             equ 0dh
        000A                    lf:             equ 0ah

0028                            STATUS:
0028                            stat:           .ds 1   ;0,1 POMIAR PRAWIDLOWY,80,81 BLEDNY
0029                            ABUF:
0029                            result:         .ds 2   ;BUFOR WYNIKOW
002B                            BUF:
002B                            arg:            .ds 1   ;TYP CZUJNIKA 0-STARY 1-NOWY

002C                            pom_a:          .ds 2
002E                            pom_b:          .ds 2
0030                            pom_c:          .ds 2

0032                            del_ab:         .ds 2
0034                            del_bc:         .ds 2
0036                            del_ca:         .ds 2

0038                            abs_ab:         .ds 2
003A                            abs_bc:         .ds 2
003C                            abs_ca:         .ds 2

003E                            rob:            .ds 2
0040                            delay:          .ds 2   ;opoznienie pobudzenia syncr

0042                            psw_rest:       .ds 1
0043                            stack:          .ds 1


0000                                    CODE
        00B5                    DSR:    REG     P3.5
        00B6                    DTR:    REG     P3.6
        00B7                    RTS:    REG     P3.7
        00B4                    CTS:    REG     P3.4
0000                                    ORG     0
0000    01 50                           AJMP    INIT

000B                                    ORG     0BH
```

Tue Jan 1 1980 09:51

```
 44    000B   02 02 70          LJMP    timO_int
 45
 46    0050                     ORG     50H
 47    0050   75 D0 00  INIT:   MOV     PSW,#0
 48    0053   75 81 43          MOV     SP,#stack
 49    0056   43 89 20          ORL     TMOD,#20H       ;SET T1MODE2 AUTO
 50    0059   43 98 52          ORL     SCON,#52H       ;SP MODE1,TI-ENABLE
 51    005C   43 88 40          ORL     TCON,#40H       ;FLAGS T1 ENABLES
 52    005F   75 8D E7          MOV     TH1,#0E7H       ;1200 BAUD
 53    0062   43 87 80          orl     pcon,#80H
 54    0065   D2 B1             SETB    TXD
 55    0067   D2 B0             SETB    RXD
 56    0069   D2 8E             SETB    TR1     ;START TIMER1 SP
 57    006B                  WAIT31:
 58    006B   C2 B6             CLR     DTR     ;SET DTR=0
 59    006D   20 B5 FD          JB      DSR,$   ;WAIT UNTIL DSR=0
 60    0070   C2 B7             CLR     RTS     ;SET RTS=0
 61           ;                JB      CTS,$   ;WAIT UNTIL CTS=0
 62    0072                  WAIT2:
 63    0072   11 E7          CONT1:   ACALL   INA     ;WAIT AND READ BOARD
 64    0074   B4 31 35                 CJNE    A,#'1',WAIT1    ;1 - WAIT1
 65    0077                  CONT2:
 66    0077   54 01                    ANL     A,#1
 67    0079   03                       RR      A
 68    007A   F5 2B                    MOV     arg,A
 69    007C   31 03                    ACALL   pomiar
 70
 71    007E   E5 28                    mov A,stat
 72    0080   23                       rl A
 73    0081   24 30                    add A,#'0'
 74    0083   12 00 F4                 call OUTA       ;send status
 75           ;                        jb stat.7,crlf
 76    0086   75 20 30                 MOV     AWP,#'0'
 77    0089   75 21 30                 MOV     AWP+1,#'0'
 78    008C   75 22 30                 MOV     AWP+2,#'0'
 79    008F   75 23 30                 MOV     AWP+3,#'0'
 80    0092   75 24 30                 MOV     AWP+4,#'0'
 81    0095   11 DC                    ACALL   HESC    ;2BYTEHEX(ABUF)-->5DIGITSASCII(AWP)
 82    0097   78 20             SKIP2:  MOV     R0,#AWP
 83    0099   E6                CONT:   MOV     A,@R0
 84    009A   11 F4                     ACALL   OUTA
 85    009C   08                       INC     R0
 86    009D   B8 25 F9                 CJNE    R0,#AWLF,CONT
 87    00A0                  crlf:
 88    00A0   74 0D                    mov A,#cr
 89    00A2   12 00 F4                 call OUTA
 90    00A5   74 0A                    mov A,#lf
 91    00A7   12 00 F4                 call OUTA
 92
 93    00AA   01 6B                    AJMP    WAIT31
 94    00AC   B4 30 C3          WAIT1:  CJNE    A,#'0',WAIT2
 95    00AF   01 77                    AJMP    CONT2
 96    00B1   7A 10             HESC2:  MOV     R2,#16
 97    00B3   E4                        CLR     A
 98    00B4   C3                        CLR     C
 99    00B5   C0 E0             HESC3:  PUSH    ACC
100    00B7   79 29                     MOV     R1,#ABUF
```

Tue Jan 1 1980 09:51

```
101   00B9  E7
102   00BA  25 E0
103   00BC  F7
104   00BD  09
105   00BE  E7
106   00BF  35 E0
107   00C1  F7
108   00C2  D0 E0
109   00C4  35 E0
110   00C6  D4
111   00C7  50 02
112   00C9  05 29
113   00CB  DA E8
114   00CD  FA
115   00CE  11 D5
116   00D0  EA
117   00D1  23
118   00D2  23
119   00D3  23
120   00D4  23
121   00D5  54 0F
122   00D7  24 30
123   00D9  F6
124   00DA  18
125   00DB  22
126   00DC  78 24
127   00DE  11 B1
128   00E0  11 B1
129   00E2  E5 29
130   00E4  11 D7
131   00E6  22
132   00E7  30 98 FD
133   00EA  C2 98
134   00EC  E5 99
135   00EE  A2 D0
136   00F0  B3
137   00F1  54 7F
138   00F3  22
139
140
141
142   00F4  A2 D0
143   00F6  B3
144   00F7  92 E7
145   00F9  30 99 FD
146   00FC  C2 99
147   00FE  F5 99
148   0100  22
149
150
151
152
153
154
155         00B2
156         0093
157         0094
```

```
          MOV     A,@R1
          ADD     A,ACC
          MOV     @R1,A
          INC     R1
          MOV     A,@R1
          ADDC    A,ACC     ;SHIFT LEFT ABUF,LEFT ABUF+1,CY
          MOV     @R1,A
          POP     ACC
          ADDC    A,ACC     ;ADD A , CY
          DA      A
          JNC     SKIP
          INC     ABUF      ;CY TO ABUF
SKIP:     DJNZ    R2,HESC3
          MOV     R2,A
          ACALL   ASCII
          MOV     A,R2
          RL      A
          RL      A
          RL      A
          RL      A
ASCII:    ANL     A,#0FH
OST1:     ADD     A,#'0'
          MOV     @R0,A.    ;WRITE ASCII TO AWP(I)
          DEC     R0
          RET
HESC:     MOV     R0,#AWP+4
          ACALL   HESC2
          ACALL   HESC2
          MOV     A,ABUF
          ACALL   OST1
          RET
INA:      JNB     R1,$
          CLR     RI
          MOV     A,SBUF
          MOV     C,P
          CPL     C
          ANL     A,#7FH
          RET

OUTA:               ;nieparzystosc,7 bitow 1 stop (600 baud)

          MOV     C,P
          CPL     C
          MOV     ACC./,C
          JNB     T1,$
          CLR     T1
          MOV     SBUF,A
          RET
;
;
;
;procedura pomiarowa tamy II
;bit definition
pom_inp:                  reg p3.2        ;wejscie pomiarowe
pom_pob:          reg p1.3          ;pobudzenie
pom_en2:          reg p1.4          ;blokada,odbiornika
```

Tue Jan 1 1980 09:51

```
158             005F            type_bit:       reg arg.7
159             0001            t0_mode:        equ 1
160             00F0            tm1_mask:       equ 0f0h
161             0007            nbr_mask:       equ 07h
162             0018            reg_psw_field:  equ 18h
163             0010            reg_bank_2:     equ 10h
164
165             0014            cycle_time:     equ 20          ;w setkach nanosekund
166             01F4            milisek:        equ 10000/cycle_time
167             0032            sto_mikros:     equ 1000/cycle_time
168             0064            time1:          equ 2*sto_mikros        ;200 us - szer. pobu
169             30D4            time2:          equ 25*milisek          ;opoznienie 1 pom ar
170             30D4            time3:          equ 25*milisek          ;opoznienie,kol. pom
171             0032            width_half:     equ sto_mikros
172             0014            t_inert:        equ 2*sto_mikros/5      ;inercja po zboczu
173             0000            status_3_accepted:      equ 0
174             0001            status_1_rejected:      equ 1
175             0080            status_3_rejected:      equ 80h
176             0081            status_time_out:        equ 81h
177
178                             ;stale typu CODE
179
180   0101      0A 00           del_max db 10,0
181
182
183                             ;program
184
185   0103                      pomiar:
186   0103      D2 AF                   setb ea                 ;zezwol. na przerwania
187   0105      85 D0 42                mov psw_rest,psw
188   0108      E5 D0                   mov a,psw
189   010A      54 E7                   anl a,#.not. reg_psw_field
190   010C      44 10                   orl a,#reg_bank_2
191   010E      F5 D0                   mov psw,a
192
193                             ;       inicjalizacja timera
194   0110      20 5F 54                jb type_bit,pomiar_t1
195   0113      E5 89                   mov a,tmod
196   0115      54 F0                   anl a,#tm1_mask
197   0117      44 01                   orl a,#t0_mode
198   0119      F5 89                   mov tmod,a
199   011B      D2 94                   setb pom_en2
200
201
202   011D      20 5F 47                jb type_bit,pomiar_t1   ;zaleznie od typu czujnik
203
204                             ;       pobudzenie asynchroniczne
205
206   0120      C2 93                   clr pom_pob
207   0122      74 14                   mov a,#time1/5
208   0124      12 02 F5                call waitA5
209   0127      D2 93                   setb pom_pob
210   0129      74 7D                   mov a,#time2/100
211   012B      12 02 EA                call waitA100
212   012E      12 03 65                call ini_meas           ;wstepny pomiar okresu
213
214                             ;       1 pomiar
```

Tue Jan 1 1980 09:51

```
215
216                                     ;       call ph_impulse
217                                     ;       mov a,#time3/100
218                                     ;       call waitA100
219     0131    12 03 27                        call meas_nt            ;wynik w AB
220     0134    78 2C                           mov R0,#pom_a
221     0136    12 02 D1                        call st_ab
222
223                                     ;       LATA
224     0139
225     0139    78 29                           mov R0,#result
226     013B    12 02 D1                        call st_ab
227     013E    75 28 00                        mov stat,#status_3_accepted
228     0141    02 02 62                        jmp pom_end
229
230                                     ;        2,pomiar
231
232     0144    12 02 FC                        call ph_impulse
233     0147    74 7D                           mov a,#time3/100
234     0149    12 02 EA                        call waitA100
235     014C    12 03 27                        call meas_nt
236     014F    78 2E                           mov R0,#pom_b
237     0151    12 02 D1                        call st_ab
238
239                                     ;       3 pomiar
240
241     0154    12 02 FC                        call ph_impulse
242     0157    74 7D                           mov a,#time3/100
243     0159    12 02 EA                        call waitA100
244     015C    12 03 27                        call meas_nt
245     015F    78 30                           mov R0,#pom_c
246     0161    12 02 D1                        call st_ab
247
248     0164    02 01 79                        jmp analiza
249
250     0167                            pomiar_t1:
251
252                                     ;       1 pomiar
253
254     0167    12 03 27                        call meas_nt            ;wynik w AB
255     016A    12 02 D1                        call st_ab
256     016D
257                                     ;        2 pomiar
258
259     016D    12 03 27                        call meas_nt
260     0170    12 02 D1                        call st_ab
261
262                                     ;       3 pomiar
263
264     0173    12 03 27                        call meas_nt
265     0176    12 02 D1                        call st_ab
266
267
268
269
270
271                                     ;       analiza wynikow
```

Tue Jan 1 1980 09:51

```
272   0179                          analiza:
273                                 ;       obliczenie roznic
274   0179   78 2C                          mov R0,#pom_a
275   017B   79 2E                          mov R1,#pom_b
276   017D   12 02 9C                       call ode_2
277   0180   78 32                          mov R0,#del_ab
278   0182   12 02 D1                       call st_ab
279
280   0185   78 2E                          mov R0,#pom_b
281   0187   79 30                          mov R1,#pom_c
282   0189   12 02 9C                       call ode_2
283   018C   78 34                          mov R0,#del_bc
284   018E   12 02 D1                       call st_ab
285
286   0191   78 30                          mov R0,#pom_c
287   0193   79 2C                          mov R1,#pom_a
288   0195   12 02 9C                       call ode_2
289   0198   78 36                          mov R0,#del_ca
290   019A   12 02 D1                       call st_ab
291
292                                 ;       moduly roznic
293   019D   78 32                          mov R0,#del_ab
294   019F   12 02 BE                       call abs_2
295   01A2   78 38                          mov R0,#abs_ab
296   01A4   12 02 D1                       call st_ab
297
298   01A7   78 34                          mov R0,#del_bc
299   01A9   12 02 BE                       call abs_2
300   01AC   78 3A                          mov R0,#abs_bc
301   01AE   12 02 D1                       call st_ab
302   01B1
303   01B1   78 36                          mov R0,#del_ca
304   01B3   12 02 BE                       call abs_2
305   01B6   78 3C                          mov R0,#abs_ca
306   01B8   12 02 D1                       call st_ab
307
308                                 ;       if max (abs 1) <= limit srednia z 3 pomiarow
309   01BB
310                                 ;       liczenie max abs - w R2 offset max
311   01BB   7A 38                          mov R2,#abs_ab
312   01BD   A8 02                          mov R0,R2
313   01BF   79 3A                          mov R1,#abs_bc
314   01C1   12 02 9C                       call ode_2
315   01C4   30 E7 02                       jnb ACC.7,comp_2        ;ab>bc
316   01C7   7A 3A                          mov R2,#abs_bc
317   01C9                          comp_2:
318   01C9   A8 02                          mov R0,R2
319   01CB   79 3C                          mov R1,#abs_ca
320   01CD   12 02 9C                       call ode_2
321   01D0   30 E7 02                       jnb ACC.7,comp_3        ;max>ca
322   01D3   7A 3C                          mov R2,#abs_ca
323   01D5                          comp_3:
324   01D5   A8 02                          mov R0,R2
325   01D7   90 01 01                       mov DPTR,#del_max
326   01DA   12 02 A6                       call odec_2
327   01DD   30 E7 54                       jnb ACC.7,srednia_3
328
```

Tue Jan 1 1980 09:51

```
329                                ;       odrzucic najgorszy lub wszystkie
330                                ;       if min (abs )<= limit then
331                                ;       i:=min (abs ); srednia = pom(i) - del(i)
332                                ;       else odrzucenie calego pomiaru
333
334                                ;       szukaj min(abs ) - R2 - numer min pomiaru
335  01E0  7A 38                           mov R2,#abs_ab
336  01E2  A8 02                           mov R0,R2
337  01E4  79 3A                           mov R1,#abs_bc
338  01E6  12 02 9C                        call ode_2
339  01E9  20 E7 02                        jb ACC.7,comp_4          ;ab<bc
340  01EC  7A 3A                           mov R2,#abs_bc
341  01EE                           comp_4:
342  01EE  A8 02                           mov R0,R2
343  01F0  79 3C                           mov R1,#abs_ca
344  01F2  12 02 9C                        call ode_2
345  01F5  20 E7 02                        jb ACC.7,comp_5          ;min<ca
346  01F8  7A 3C                           mov R2,#abs_ca
347  01FA                           comp_5:
348  01FA  A8 02                           mov R0,R2
349  01FC  90 01 01                        mov DPTR,#del_max
350  01FF  12 02 A6                        call odec_2
351  0202  20 E7 24                        jb ACC.7,meas_rejected   ;min>limit
352
353                                ;       srednia z dwoch
354  0205  C3                              clr C
355  0206  EA                              mov A,R2
356  0207  94 38                           subb A,#abs_ab           ;offset minimum
357  0209  FA                              mov R2,A
358  020A  24 32                           add A,#del_ab            ;delta(min)
359  020C  F8                              mov R0,A
360  020D  12 02 D6                        call shr_2               ;delta/2
361  0210  78 3E                           mov R0,#rob
362  0212  12 02 D1                        call st_ab
363  0215  74 2C                           mov a,#pom_a
364  0217  2A                              add A,R2                 ;pom(min)
365  0218  F8                              mov R0,a
366  0219  79 3E                           mov R1,#rob
367  021B  12 02 9C                        call ode_2               ;pom-delta/2
368  021E  78 29                           mov R0,#result
369  0220  12 02 D1                        call st_ab               ;srednia w pole wyniku
370  0223  75 28 01                        mov stat,#status_1_rejected
371  0226  02 02 62                        jmp pom_end
372
373                                ;wszystkie odrzucone
374  0229                           meas_rejected:
375  0229  75 28 80                        mov stat,#status_3_rejected
376  022C  E5 00                           mov A,0
377  022E  85 00 F0                        mov B,0
378  0231  02 02 62                        jmp pom_end
379                                ; srednia z trzech
380  0234                           srednia_3:
381                                ;       zakladamy jednobajtowa sume roznic
382                                ;       srednia = a + ((c-a)-(a-b))/3
383
384  0234  78 36                           mov R0,#del_ca
385  0236  79 32                           mov R1,#del_ab
```

Tue Jan 1 1980 09:51

```
386   0238   12 02 9C          call ode_2
387   023B   E5 F0             mov A,B                 ;mlodsza czesc roznicy
388   023D   20 E7 0A          jb ACC.7,del_3_minus
389   0240   75 F0 03          mov B,#3
390   0243   84                div AB
391   0244   75 F0 00          mov B,#0
392   0247   02 02 55          jmp del_3
393   024A                   del_3_minus:
394   024A   F4                cpl A
395   024B   04                inc A                   ;-A
396   024C   75 F0 03          mov B,#3
397   024F   84                div AB
398   0250   F4                cpl A
399   0251   04                inc A
400   0252   75 F0 FF          mov B,#0FFH             ; rozszerzenie wyniku
401   0255                   del_3:
402   0255   25 2C             add A,pom_a
403   0257   F5 29             mov result,A
404   0259   E5 2D             mov A,pom_a+1
405   025B   35 F0             addc A,B
406   025D   F5 2A             mov result+1,A
407   025F   75 28 00          mov stat,#status_3_accepted
408   0262                   pom_end:
409   0262   85 42 D0          mov psw,psw_rest
410   0265   78 29             mov R0,#result
411   0267   12 02 D6          call shr_2
412   026A   12 02 D1          call st_ab
413   026D   C2 AF             clr EA
414   026F   22                ret
415
416   0270                   tim0_int:
417   0270   C2 8C             clr TR0
418   0272   C2 8D             clr TF0
419   0274   D2 93             setb pom_pob
420   0276   D2 94             setb pom_en2
421   0278   D0 E0             pop a
422   027A   D0 E0             pop a                   ;wyjscie z inta
423   027C   D0 E0             pop a
424   027E   D0 E0             pop a                   ;wyjscie z procedury
425   0280   74 00             mov A,#0
426   0282   75 F0 00          mov B,#0
427   0285   78 29             mov R0,#result
428   0287   12 02 D1          call st_ab
429   028A   85 42 D0          mov psw,psw_rest
430   028D   C2 AF             clr EA
431   028F   75 28 81          mov stat,#status_time_out
432   0292   32                reti
433
434                            ;procedury
435
436   0293                   dod_2:
437   0293   E7                mov A,@R1
438   0294   26                add A,@R0
439   0295   09                inc R1
440   0296   08                inc R0
441   0297   F5 F0             mov B,A
442   0299   E7                mov A,@R1
```

Tue Jan 1 1980 09:51

```
443   029A   36                    addc A,@R0
444   029B   22                    ret
445
446   029C                 ode_2:
447   029C   C3                    clr C
448   029D   E6                    mov A,@R0
449   029E   97                    subb A,@R1
450   029F   F5 F0                 mov B,A
451   02A1   08                    inc R0
452   02A2   09                    inc R1
453   02A3   E6                    mov A,@R0
454   02A4   97                    subb A,@R1
455   02A5   22                    ret
456   02A6                 odec_2:
457   02A6   C3                    clr C
458   02A7   E5 00                 mov A,0
459   02A9   93                    movc A,@A+DPTR
460   02AA   96                    subb A,@R0
461   02AB   F5 F0                 mov B,A
462   02AD   08                    inc R0
463   02AE   E5 01                 mov A,1
464   02B0   93                    movc A,@A+DPTR
465   02B1   96                    subb A,@R0
466   02B2   22                    ret
467
468   02B3                 cmpl_2:
469   02B3   E6                    mov A,@R0
470   02B4   F4                    cpl A
471   02B5   04                    inc A
472   02B6   F5 F0                 mov B,A
473   02B8   08                    inc R0
474   02B9   E6                    mov A,@R0
475   02BA   F4                    cpl A
476   02BB   34 00                 addc A,#0
477   02BD   22                    ret
478
479   02BE                 abs_2:
480   02BE   08                    inc R0
481   02BF   E6                    mov A,@R0
482   02C0   18                    dec R0
483   02C1   20 E7 03              jb ACC.7,abs0
484   02C4   A6 F0                 mov @R0,b
485   02C6   22                    ret
486   02C7                 abs0:
487   02C7   51 B3                 call cmpl_2
488   02C9   22                    ret
489
490   02CA                 copy_2:
491   02CA   E6                    mov A,@R0
492   02CB   F7                    mov @R1,A
493   02CC   08                    inc R0
494   02CD   09                    inc R1
495   02CE   E6                    mov A,@R0
496   02CF   F7                    mov @R1,A
497   02D0   22                    ret
498
499   02D1                 st_ab:
```

Tue Jan 1 1980 09:51

```
500   02D1   A6 F0          mov @R0,B
501   02D3   08             inc R0
502   02D4   F6             mov @R0,A
503   02D5   22             ret
504
505   02D6                 shr_2:
506   02D6   08             inc R0
507   02D7   E6             mov A,@R0
508   02D8   18             dec R0
509   02D9   20 E7 0B       jb ACC.7,shr_minus
510   02DC   C3             clr C
511   02DD                 shr_0:
512   02DD   13             rrc A
513   02DE   C0 E0          push A
514   02E0   E6             mov A,@R0
515   02E1   13             rrc a
516   02E2   F5 F0          mov B,a
517   02E4   D0 E0          pop A
518   02E6   22             ret
519   02E7                 shr_minus:
520   02E7   D3             setb C
521   02E8   41 DD          jmp shr_0
522
523
524   02EA                 waitA100:
525   02EA   F5 F0          mov B,A
526   02EC                 wait2:
527   02EC   74 12          mov A,#18
528   02EE   12 02 F5       call waitA5
529   02F1   D5 F0 F8       djnz B,wait2
530   02F4   22             ret
531
532   02F5                 waitA5:
533   02F5   00             nop
534   02F6   00             nop
535   02F7   00             nop
536   02F8   D5 E0 FA       djnz a,waitA5
537   02FB   22             ret
538
539
540   02FC                 ph_impulse:
541   02FC   C2 94          clr pom_en2          ;enable input
542                         clr IF0              ;zezwol. na przerwanie time_
543   02FE   C2 8D          mov TH0,#0
544   0300   75 8C 00       mov TL0,#0
545   0303   75 8A 00       setb ET0
546   0306   D2 A9          setb TR0
547   0308   D2 8C
548                         jnb pom_inp,$
549   030A   30 B2 FD       jb pom_inp,$         ;wykrywanie 1-0
550   030D   20 B2 FD       setb pom_en2
551   0310   D2 94
552                         clr TR0
553   0312   C2 8C          clr ET0
554   0314   C2 A9
555                         mov B,#delay
556   0316   75 F0 40
```

Tue Jan 1 1980 09:51

```
557   0319   74 41          mov A,#delay+1
558   031B   12 03 B6       call wait_in_timer      ;pobudzenie w srodku ujemneg
559
560                          ;pobudzenie w fazie
561
562   031E   C2 93           clr pom_pob
563   0320   74 14           mov a,#time1/5
564   0322   51 F5           call waitA5
565   0324   D2 93           setb pom_pob
566
567   0326   22             ret
568
569   0327                 meas_nt:
570
571                          ;przygotowanie timera
572
573   0327   C2 8D           clr t+0
574   0329   75 8A 00        mov tl0,#0
575   032C   75 8C 00        mov th0,#0
576   032F   7A 28           mov r2,#2*cycle_time    ;licznik petli pomiarowej
577   0331   D2 A9           setb et0
578
579                          ;fazowanie
580   0333   C2 94           clr pom_en2
581   0335   30 B2 FD        jnb pom_inp,$
582   0338   74 14           mov A,#t_inert
583   033A   51 F5           call waitA5
584   033C   20 B2 FD        jb pom_inp,$
585   033F   D2 8C           setb tr0                ;start zliczania
586
587
588   0341
589   0341   74 14           mov A,#t_inert
590   0343   51 F5           call waitA5
591   0345   30 B2 FD        jnb pom_inp,$
592   0348   74 14           mov A,#t_inert
593   034A   51 F5           call waitA5
594   034C   20 B2 FD        jb pom_inp,$
595   034F   DA F0           djnz r2,pomloop         ;+24 cykle =4us
596
597   0351   C2 8C           clr tr0                 ;stop timer +12 cykli =2us
598   0353   C2 A9           clr ET0
599   0355   C2 8D           clr TF0
600   0357   D2 94           setb pom_en2
601
602   0359   C3             clr C
603   035A   E5 8A           mov A,TL0
604   035C   94 02           subb a,#2               ;poprawka na djnz
605   035E   F5 F0           mov B,A                 ;i-czesc
606   0360   E5 8C           mov A,TH0
607   0362   94 00           subb a,#0
608
609                          ;       wynik pomiaru w AB
610   0364   22             ret
611
612                        ini_meas:                 ;pomiar wstepny
613   0365   C2 8D           clr tf0
```

Tue Jan 1 1980 09:51

```
614   0367   75 8A 00          mov tl0,#0
615   036A   75 8C 00          mov th0,#0
616   036D   D2 A9             setb et0
617   036F   D2 8C             setb TR0
618
619                            ;fazowanie
620   0371   C2 94             clr pom_en2
621   0373   30 B2 FD          jnb pom_inp,$
622   0376   74 14             mov A,#t_inert
623   0378   51 F5             call waitA5
624   037A   20 B2 FD          jb pom_inp,$
625   037D   C2 8C             clr TR0
626   037F   C2 8D             clr tf0
627   0381   75 8A 00          mov tl0,#0
628   0384   75 8C 00          mov th0,#0
629   0387   D2 8C             setb TR0                ;start zliczania
630   0389   74 14             mov A,#t_inert
631   038B   51 F5             call waitA5
632   038D   30 B2 FD          jnb pom_inp,$
633   0390   74 14             mov A,#t_inert
634   0392   51 F5             call waitA5
635   0394   20 B2 FD          jb pom_inp,$
636   0397   C2 8C             clr TR0                 ;stop zliczania
637   0399   C2 A9             clr ET0
638   039B   D2 94             setb pom_en2
639   039D   85 8A 40          mov delay,TL0
640   03A0   85 8C 41          mov delay+1,TH0
641                            ;       delay=okres/4-width_half
642   03A3   78 40             mov R0,#delay
643   03A5   51 D6             call shr_2
644   03A7   51 D6             call shr_2
645   03A9   C3                clr C
646   03AA   E6                mov A,@R0
647   03AB   94 41             subb A,#<(width_half+15)
648   03AD   F5 40             mov delay,A
649   03AF   08                inc R0
650   03B0   E6                mov A,@R0
651   03B1   94 00             subb A,#>(width_half+15)
652   03B3   F5 41             mov delay+1,A
653   03B5   22                ret
654
655                        wait_in_timer:          ;czekaj AB cykli zegara +15
656   03B6   C0 E0             push A
657   03B8   F5 F0             mov B,A
658   03BA   F4                cpl A
659   03BB   04                inc A
660   03BC   F5 8A             mov TL0,a
661   03BE   D0 E0             pop A
662   03C0   F4                cpl A
663   03C1   34 00             addc A,#0
664   03C3   F5 8C             mov TH0,A
665   03C5   D2 8C             setb TR0
666   03C7   30 8D FD          jnb TF0,$
667   03CA   C2 8D             clr TF0
668   03CC   C2 8C             clr TR0
669   03CE   22                ret
670
```

670

Tue Jan 1 1980 09:51

671 03CF END
671 03CF END

Tue Jan 1 1980 09:51

Defined	Symbol Name		Value	References				
11	ABUF		0029	100	112	129		
121	ASCII.		00D5	115				
5	AWLF		0025	86				
4	AWP		0020	76	77	78	79	80
Pre	BSECT		0000					
13	BUF		002B					
Pre	CODE		0050	35	40	43	46	
83	CONT		0099	86				
63	CONT1		0072					
65	CONT2		0077	95				
39	CTS	=	00B4					
Pre	DATA		0020	2	3			
36	DSR	=	00B5	59				
37	DTR	=	00B6	58				
126	HESC		00DC	81				
96	HESC2		00B1	127	128			
99	HESC3		00B5	113				
132	INA		00E7	63				
47	INIT		0050	41				
122	OST1		00D7	130				
140	OUTA		00F4	74	84	89	91	
Pre	RSECT		0000					
38	RTS	=	00B7	60				
113	SKIP		00CB	111				
82	SKIP2		0097					
9	STATUS		0028					
94	WAIT1		00AC	64				
62	WAIT2		0072	94				
57	WAIT31		006B	93				
486	abs0		02C7	483				
479	abs_2		02BE	294	299	304		
24	abs_ab		0038	295	311	335	356	
25	abs_bc		003A	300	313	316	337	340
26	abs_ca		003C	305	319	322	343	346
272	analiza		0179	248				
14	arg		002B	68	158			
468	cmpi_2		02B3	487				
317	comp_2		01C9	315				
323	comp_3		01D5	321				
341	comp_4		01EE	339				
347	comp_5		01FA	345				
490	copy_2		02CA					
6	cr	=	000D	88				
87	crlf		00A0					
165	cycle_time	=	0014	166	167	576		
401	del_3		0255	392				
393	del_3_minus		024A	388				
20	del_ab		0032	277	293	358	385	
21	del_bc		0034	283	298			
22	del_ca		0036	289	303	384		
180	del_max		0101	325	349			
29	delay		0040	556	557	639	640	642
436	dod_2		0293					
612	ini_meas		0365	212				
7	lf	=	000A	90				
569	meas_nt		0327	219	235	244	254	259

Tue Jan 1 1980 09:51

Defined	Symbol Name		Value	References				
374	meas_rejected		0229	351				
166	milisek	=	01F4	169	170			
161	nbr_mask	=	0007					
446	ode_2		029C	276	282	288	314	320
456	odec_2		02A6	326	350			
540	ph_impulse		02FC	232	241			
16	pom_a		002C	220	274	287	363	402
17	pom_b		002E	236	275	280		
18	pom_c		0030	245	281	286		
157	pom_en2	=	0094	199	420	541	551	580
408	pom_end		0262	228	371	378		
155	pom_inp	=	0082	549	550	581	584	591
156	pom_pob	=	0093	206	209	419	562	565
185	pomiar		0103	69				
250	pomiar_t1		0167	194	202			
588	pomloop		0341	595				
31	psw_rest		0042	187	409	429		
163	reg_bank_2	=	0010	190				
162	reg_psw_field	=	0018	189				
12	result		0029	225	368	403	406	410
28	rob		003E	361	366			
511	shr_0		02DD	521				
505	shr_2		02D6	360	411	643	644	
519	shr_minus		02E7	509				
380	sredn1a_3		0234	327				
499	st_ab		02D1	221	226	237	246	255
				301	306	362	369	412
32	stack		0043	48				
10	stat		0028	71	227	370	375	407
174	status_1_rejected	=	0001	370				
173	status_3_accepted	=	0000	227	407			
175	status_3_rejected	=	0080	375				
176	status_time_out	=	0081	431				
167	sto_mikros	=	0032	168	171	172		
159	t0_mode	=	0001	197				
172	t_inert	=	0014	582	589	592	622	630
416	tim0_int		0270	44				
168	time1	=	0064	207	563			
169	time2	=	30D4	210				
170	time3	=	30D4	233	242			
160	tm1_mask	=	00F0	196				
158	type_bit	=	005F	194	202			
526	wait2		02EC	529				
524	waitA100		02EA	211	234	243		
532	waitA5		02F5	208	528	536	564	583
655	wait_in_timer		03B6	558				
171	width_half	=	0032	647	651			

Lines Assembled : 671 Assembly Errors : 0

ZAŁĄCZNIK Nr.6

Listening programu kalkulatora pomiarowego PSION-ORG-8 III modelu miernika polowego.

INSTALUJ.OPL Tuesday, April 18, 1989 Page 1

```
INSTAL:
LOCAL NUMBER%
IF EXIST("A:CZUJNIKI")
 OPEN "A:CZUJNIKI",A,NR%,BASE,J%,CC,WO,T,MIANO$
ELSE
 CREATE "A:CZUJNIKI",A,NR%,BASE,J%,CC,WO,T,MIANO$
ENDIF
CLS
PRINT "*  Instalacja  **  czujnikow   *";
PAUSE -100
START::
CLS
PRINT "Podaj nr czujnik(0-koniec):", :INPUT NUMBER%
IF NUMBER%>0
 FIRST
   ILE NOT EOF
  IF A.NR%=NUMBER%
   CLS
   IF UPPER$(CHR$(DISP(1,"Istnieje opis czujnika nr "+GEN$(NUMBER%,3)+CHR$(9)+"Usunac
"+CHR$(31)+"(Tak/Nie)")))="T"
    ERASE
    BREAK
   ELSE
    GOTO START::
  ENDIF
  ELSE
   NEXT
  ENDIF
 ENDWH
 A.NR%=NUMBER%
 CLS
 PRINT "Podaj parametry"
 PRINT "Baza  =", :INPUT A.BASE
 PRINT "Stala =", :INPUT A.CC
   NT "Wsp. J=", :INPUT A.J%
 PRINT "Miano -", :INPUT A.MIANO$
 A.T=PUMIAR:(0)
 A.WO=1.0/(A.T**2)
 APPEND
 GOTO START::
ELSE
 CLOSE
ENDIF
```

MIERZ.OPL Tuesday, April 18, 1989

```
MIERZ:
LOCAL NUMBER%
CLS
PRINT "Pomiar czujnika"
PRINT "nr";CHR$(31), :INPUT NUMBER%
IF EXIST("A:CZUJNIKI")
 OPEN "A:CZUJNIKI",A,NR%,BASE,J%,CC,WO,T,MIANO$
ELSE
 DO
  NUMBER%=DISP(1,"Brak opisu czujnikow"+CHR$(9)+"Nacisnij <EXE>"+CHR$(33))
 UNTIL NUMBER%=13
 STOP
ENDIF
FIRST
WHILE NOT EOF
 IF NUMBER%=A.NR%
  BREAK
 ELSE
  NEXT
 ENDIF
ENDWH
IF EOF
 DO
 UNTIL DISP(1,"Czujnik nr "+GEN$(NUMBER%,3)+" nie jest zainstalowany"+CHR$(33)+CHR$(9)+"Nac
exe>")=13
 STOP
ELSE
 DO
  A.T=POMIAR:(0)
  CLS
  PRINT GEN$(A.BASE+A.CC*((-1)**A.J%)*(A.WO-1.0/(A.T**2))*1.0E6,10),LEFT$(A.MIANU$,6)
 UNTIL VIEW(2,"Pomiar czujnika nr "+GEN$(A.NR%,3)+" <SPACE>-nastepny pomiar / <EXE>-koniec
       ")=13
ENDIF
```

WYSWIETL.OPL Tuesday, April 18, 1989

```
disp:
LOCAL TIME%
TIME%=-40
IF EXIST("A:CZUJNIKI")
 OPEN "A:CZUJNIKI",A,NR%,BASE,J%,CC,WO,T,MIANO$
 FIRST
 WHILE NOT EOF
  CLS
  PRINT "Czujnik",A.NR%
  PAUSE TIME%
  PRINT "Baza =",A.BASE
  PAUSE TIME%
  PRINT "J =",A.J%
  PAUSE TIME%
  PRINT "Cc =",A.CC
  PAUSE TIME%
  PRINT "WO =",GEN$(A.WO,10)
  PAUSE TIME%
  PRINT "Okres =",FIX$(A.T,4,7)
  PAUSE TIME%
  PRINT "Miano -",CHR$(34);A.MIANO$;CHR$(34)
  PAUSE TIME%
  NEXT
 ENDWH
 CLS
 DISP(1,"Koniec listy czujnikow                ")
ENDIF
```

POMIAR.OPL Tuesday, April 18, 1989

```
POMIAR:(TYP%)
LOCAL RESULT$(9)
LSET:(1200,1,7,1,0,1,-1,-1,-1,-1,-1,-1,-1,0,0)
TRIG$:(1,CHR$(48+TYP%))
RESULT$=LINPUT$:(8)
RETURN VAL(LEFT$(RESULT$,LEN(RESULT$)-2))*1.0E-4
```

ZAŁĄCZNIK Nr.5

Listening programu komputera IBM-PC dla modelu strunowego miernika

```
#define READY Ø
#define POMIAR 1
#define TEST 2
#define TRUE 1
#define FALSE Ø

unsigned int far *config=(unsigned int far *)ØxAØØØØØØØ;
unsigned int far *map=(unsigned int far *)ØxAØØØØØ8Ø;
unsigned char far *cmd=(unsigned char far *)ØxAØØØØ1ØØ;
unsigned char far *cntrl=(unsigned char far *)ØxAØØØØ1Ø1;
unsigned int far *result=(unsigned int far *)ØxAØØØØ4ØØ;
unsigned char far *tests=(unsigned char far *)ØxAØØØØ4ØØ;

extern mconnect(), mdiscon(), mz8Ø();

struct res {
        unsigned char nomod;
        unsigned char noczuj;
        unsigned int wynikp; };


/*zmiana konfiguracji systemu, w parametrze
  mapa 2 bity na czujnik: ØØ - nie ma, Ø1 - stary, 11 - nowy
typ*/

konfiguracja(mapa)
unsigned int *mapa;
{
 register int i;
 unsigned char far *ptr;
 mconnect();
 for(i=Ø, ptr=cntrl; i<4; ++i, ++ptr)
   *ptr=i+1;
 for(i=Ø; i<64; ++i)
   *(config+i)=*mapa++;
}

/*sprawdzenie sprawnosci systemu, funkcja zwraca wartosc:
  Ø - OK
  1 - uszkodzenie modulu, do obszaru wynik wpisane wyniku tes
tu w postaci
      ciagu 65 bajtow. Wartosc odpowiedniego bajtu rowna Ø oz
nacza modul
      sprawny, wartosc rozna od Ø oznacza uszkodzenie modulu.
 Bajt zerowy
      odpowiada procesorowi komunik., kolejne bajty modulom p
omiarowym.
  2 - nie ustawiono konfiguracji sytemu */
test_systemu(wynik)
unsigned char *wynik;
{
 register int i;
 int res=Ø;
 if(check()) return(2);
 *cmd=TEST;  i=3;
```

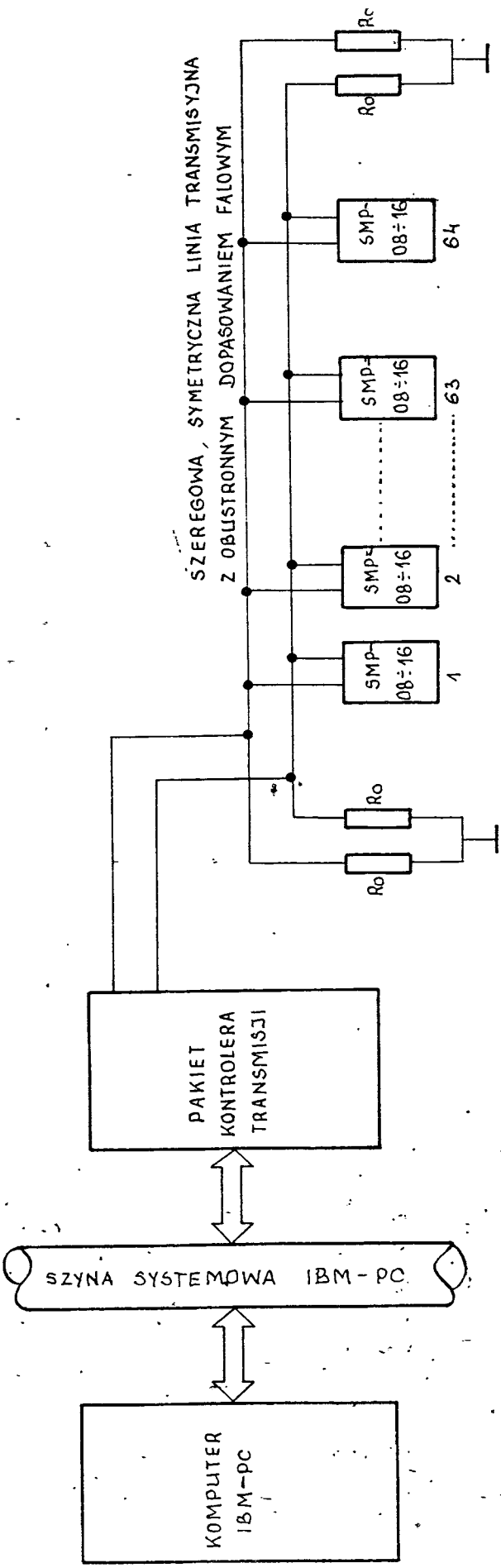
```
 do {
  mdiscon();
  while(mz8Ø()!=Ø);
  mconnect();
  if(*cmd!=READY)
    if(*cmd!=TEST) error(1);
    else continue;
  else break;
 } while(--i);
 if(i==Ø) error(2);
 for(i=Ø; i<65; ++i)
   if((*wynik++=*(tests+i))!=Ø) res=1;
 return(res);
}

/*sprawdza, czy byla ustawiona konfiguracja systemu (czy pami
ec przy IBM
  i czy ustawione bajty kontrolne),
  zwraca TRUE - konfiguracja nie ustawiona, FALSE - OK*/
check()
{
 register int i;
 unsigned char far *ptr;
 if(mz8Ø()==Ø) return(TRUE); /*pamiec odlaczona*/
 else {
   for(i=1, ptr=cntrl; i<5 && *ptr==i; ++i, ++ptr);
   if(i<5) return(TRUE);  /*nie ustawione bajty kontrolne*/
   else return(FALSE);   /*OK*/ }
}


/*pomiar, w parametrze mapa 2 bity na czujnik: ØØ - nie mierz
ony, Ø1 - pomiar.
 Funkcja zwraca: Ø - OK i wpisuje do obszaru wyniki rezultaty
 pomiarow
 w postaci 4-bajtowych rekordow: nr modulu (1 bajt), nr czujn
ika (1 bajt),
 wynik (2 bajty),
 1 - niezgodnosc mapy pomiarow z zadana konfiguracja czujniko
w,
 2 - nie ustawiona konfiguracja systemu */
pomiar_czujnikow(mapa,wyniki)
unsigned int *mapa;
struct res wyniki[];
{
 register int i,j;
 unsigned int *tmp;
 unsigned int far *pom;
 unsigned int far *pom1;
 unsigned int bit, val;
 if(check()) return(2);
 for(i=Ø, tmp=mapa; i<64; ++i, ++tmp)
   if((*(config+i) & (*tmp))!= *tmp) return(1);
 for(i=Ø, tmp=mapa; i<64; ++i)
   *(map+i)=*tmp++;
 *cmd=POMIAR; i=3;
 do {
  mdiscon();
```

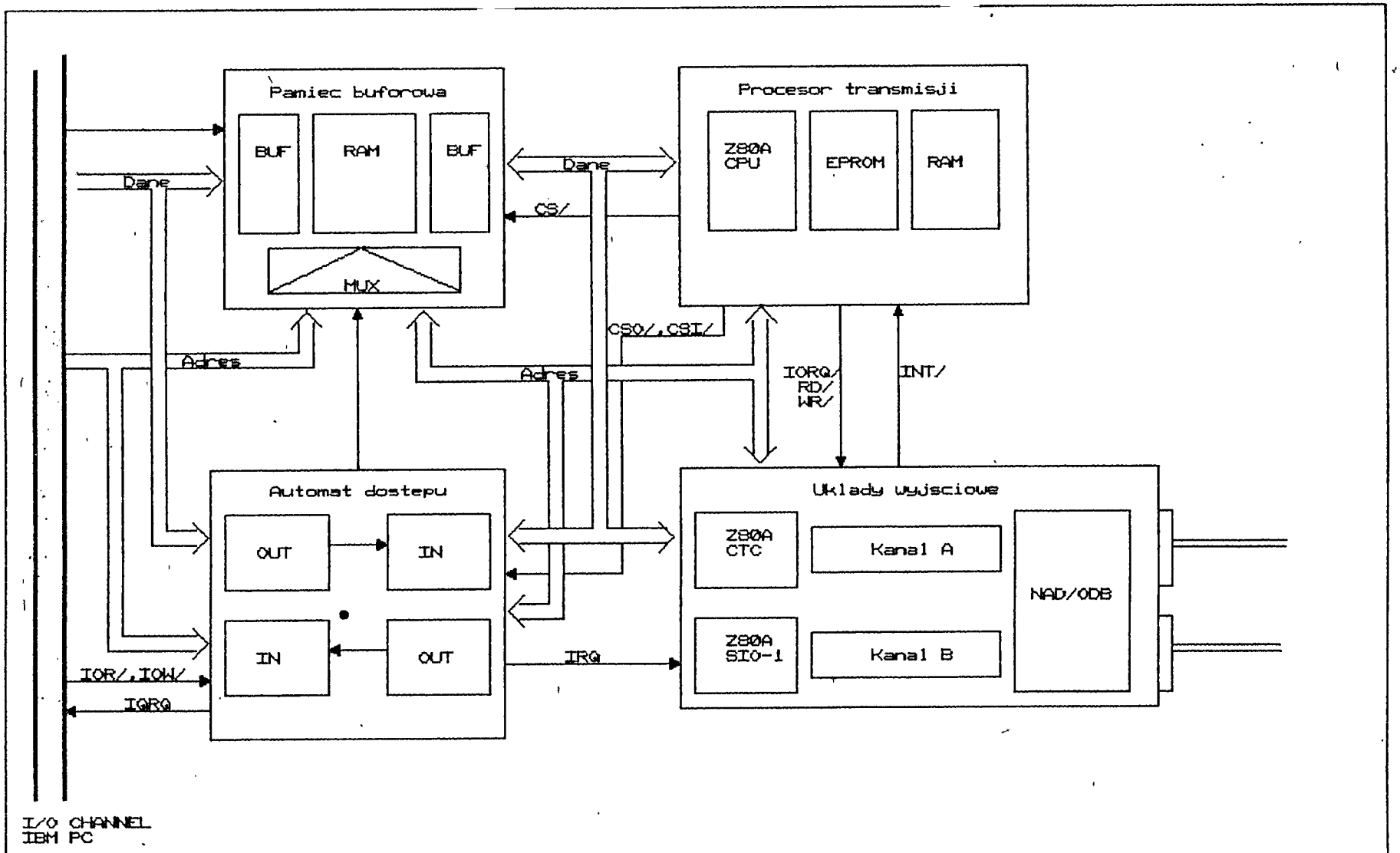
```
   while(mz8Ø()!=Ø);
   mconnect();
   if(*cmd!=READY)
     if(*cmd!=POMIAR) error(3);
     else continue;
   else break;
  } while(--i);
  if(i==Ø) error(4);
  pom1=(unsigned int far *)map;
  for(i=Ø, pom=result; i<64; ++i, ++pom1) {
    bit=1;
    for(j=Ø; j<8; ++j, ++pom) {
      if(((*pom1) & bit) !=Ø) {
        wyniki->nomod=i;
        wyniki->noczuj=j;
        wyniki->wynikp=*pom;
        ++wyniki; }
      bit<<=2; }
    }
  return(Ø);
}

error(int n)
{
  switch(n) {
    case 1: puts("\nzamazana pamiec wsp. w czasie TESTu\n"); b
reak;
    case 2: puts("\nnieudane trzy proby TESTu\n"); break;
    case 3: puts("\nzamazana pamiec wsp. w czasie POMIARu\n");
  break;
    case 4: puts("\nnieudane trzy proby POMIARu\n"); break;
  }
  exit();
}
```

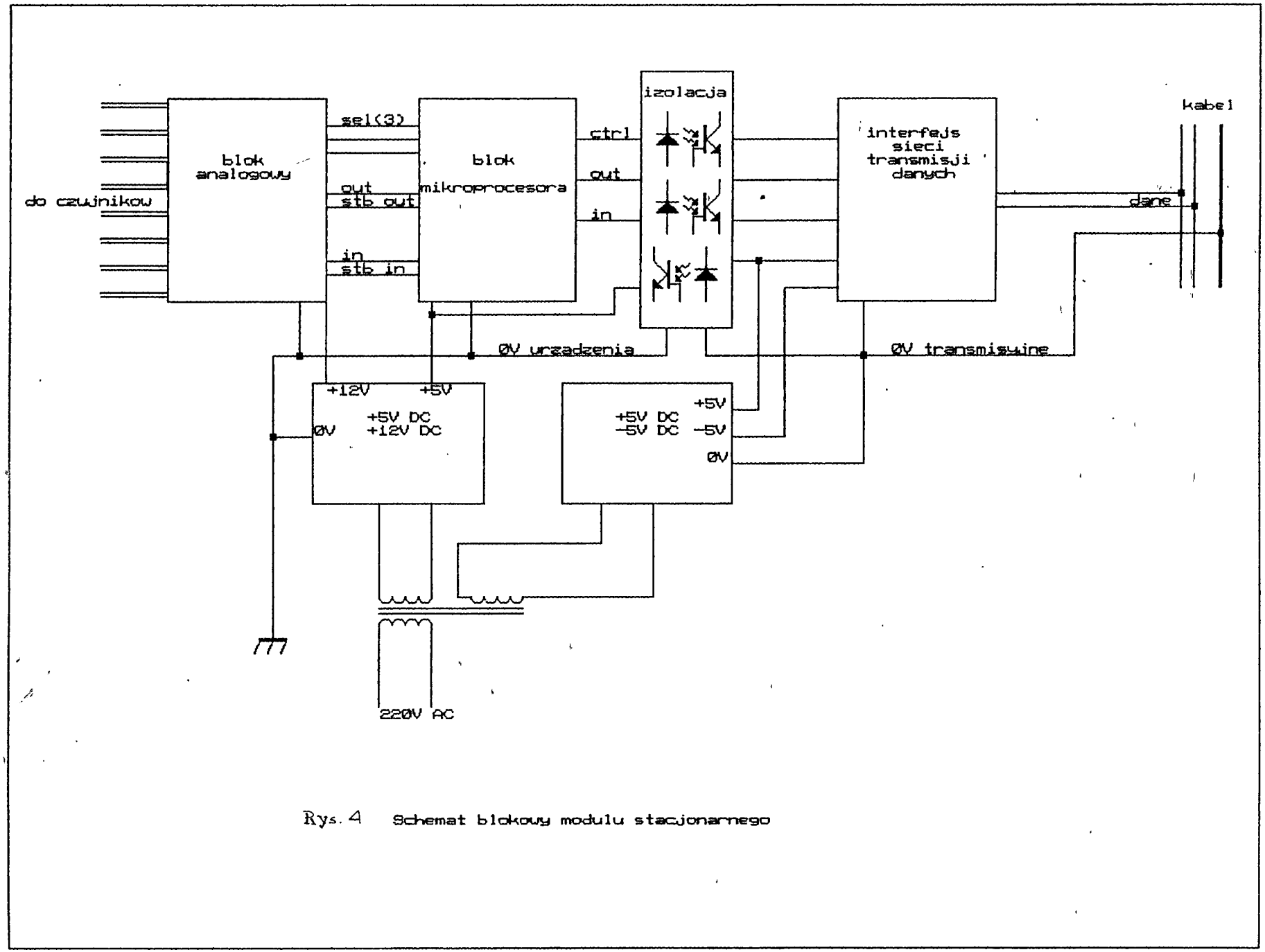


Rys. 1 Schemat blokowy systemu pomiarowego z zastosowaniem modułowych przyrządów pomiarowych SMP 08 - 16 współpracujących z komputerem IBM PC za pośrednictwem szeregowej magistrali komunikacyjnej.

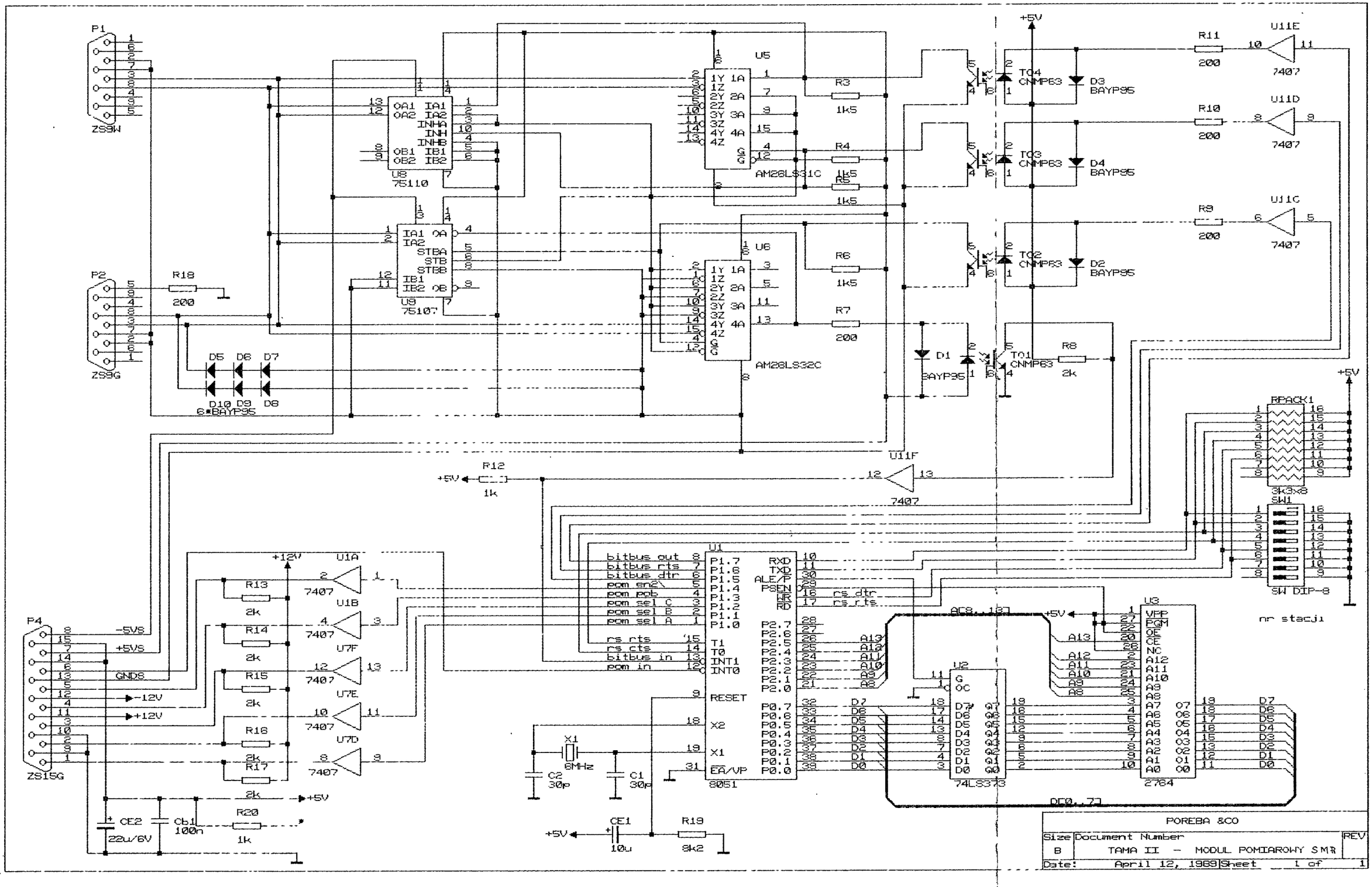
Rys.2 Postać podstawowego komunikatu informacyjnego wg warstwy sterowania łącza protokółu transmisji SDLC



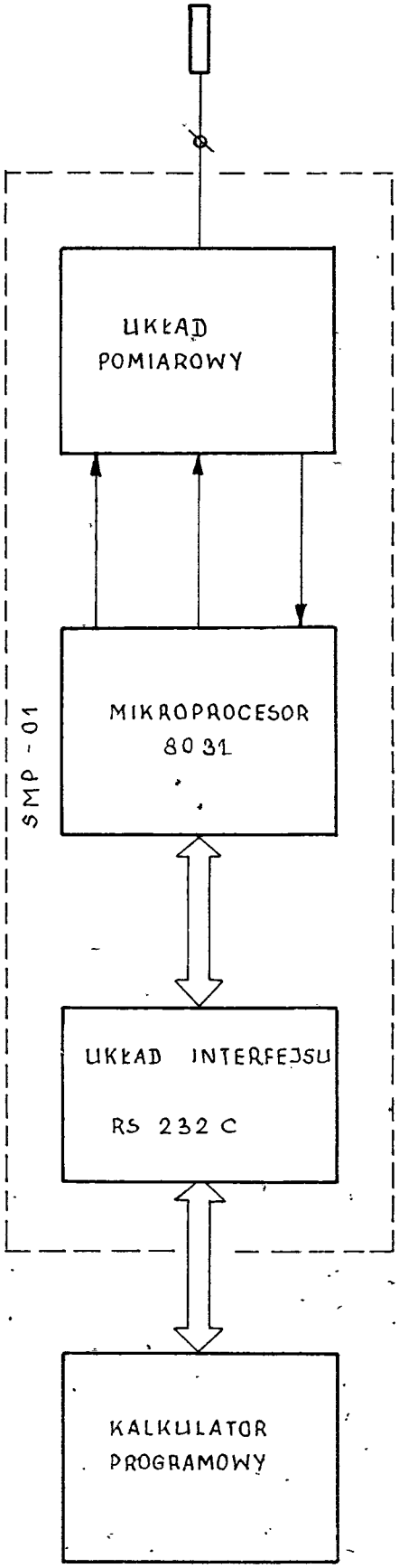
Rys. 3 Schemat blokowy sterownika BITBUS do IBM PC F-my STER-PROJEKT



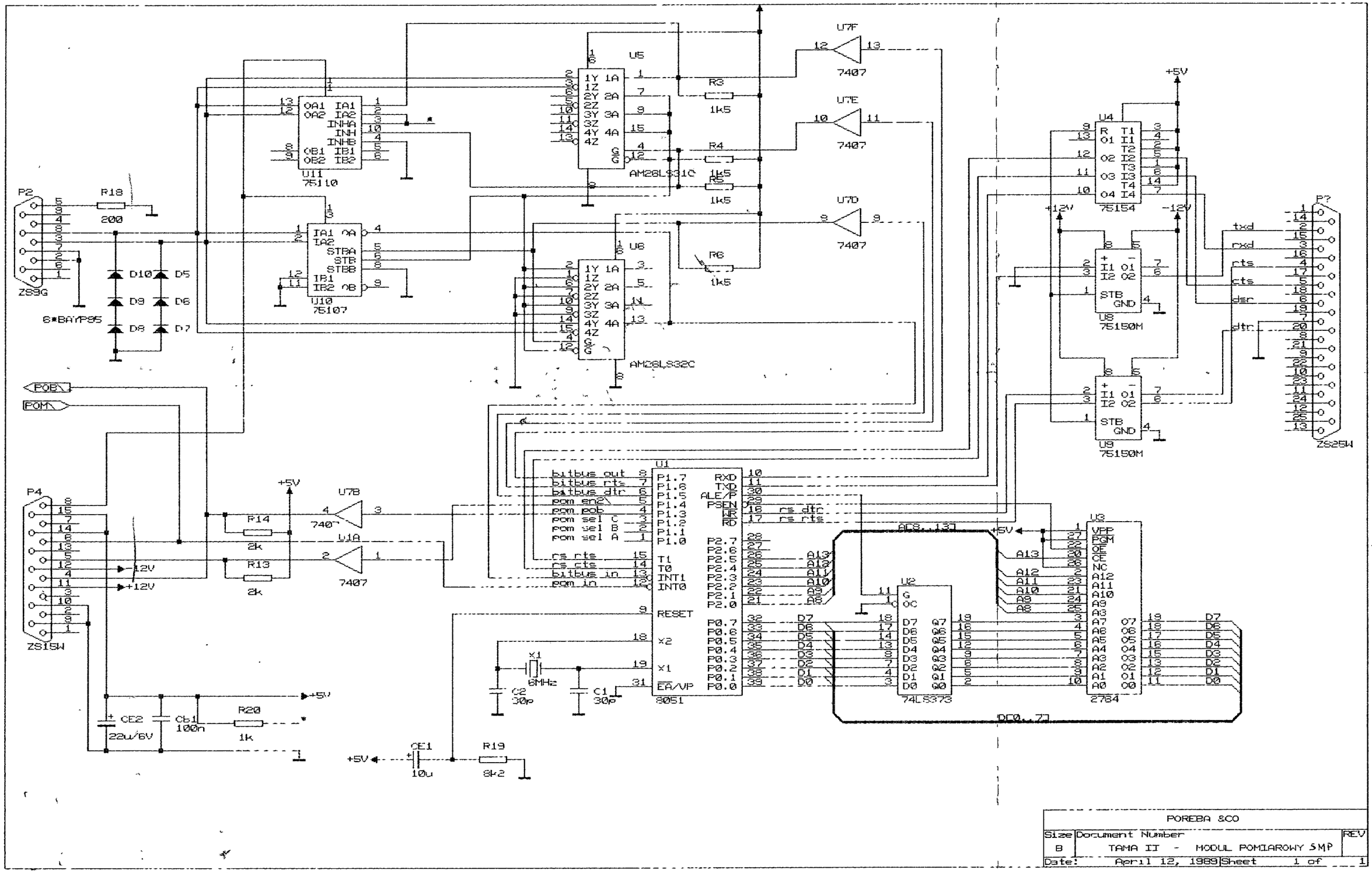
Rys. 4 Schemat blokowy modulu stacjonarnego


P1
ZS9W
P2
ZS9G
P4
ZS15G
U8
75110
U9
75107
U5
AM28LS31C
U6
AM28LS32C
U1
8051
U2
74LS373
U3
2764
U1A
U1B
U7F
U7E
U7D
U11E
U11D
U11C
U11F
7407
TO4 CNMP63
TO3 CNMP63
TO2 CNMP63
TO1 CNMP63
D1 BAYP35
D2 BAYP95
D3 BAYP95
D4 BAYP95
D5 D6 D7
D10 D9 D8
6*BAYP95
R3 1k5
R4 1k5
R5 1k5
R6 1k5
R7 200
R8 2k
R9 200
R10 200
R11 200
R12 1k
R13 2k
R14 2k
R15 2k
R16 2k
R17 2k
R18 200
R19 8k2
R20 1k
RPACK1
3k3x8
SW1
SW DIP-8
nr stacji
X1
8MHz
C1 30p
C2 30p
CE1 10u
CE2 22u/6V
Cb1 100n
+5V
+12V
-12V
-5VS
+5VS
GNDS
bitbus out
bitbus rts
bitbus dtr
pom en2\
pom pob
pom sel C
pom sel B
pom sel A
rs rts
rs cts
bitbus in
pom in
RESET
X2
X1
EA/VP
rs_dtr
rs_rts
AC8..15
DC0..7
POREBA &CO
Size B
Document Number
TAMA II - MODUL POMIAROWY SMR
REV
Date: April 12, 1989
Sheet 1 of 1

Rys. 5



Rys. 6 Schemat modułu pomiarowego czujników strunowych w wersji polowej współpracującego z kalkulatorem programowanym poprzez łącze RS 232C.



POREBA &CO		
Size B	Document Number: TAMA II - MODUL POMIAROWY SMP	REV
Date: April 12, 1989	Sheet 1 of 1	

Rys. 7