



MicroMade
Systemy Mikroprocesorowe

ul. Sikorskiego 33
64-920 PIŁA
tel/fax: (67) 13-24-14

Biuro Konstrukcyjne:
ul. Fiszer 14, 80-952 GDAŃSK
tel: (58) 47-15-35, fax: (58) 41-61-44

DSM-51

Dydaktyczny System Mikroprocesorowy

Dokumentacja Techniczna

© MicroMade 1992



Spis Treści

1	Charakterystyka systemu DSM-51	3
2	Przeznaczenie systemu DSM-51	3
3	Dane techniczne	4
4	Budowa systemu	5
4.1	Mikroprocesor	5
4.2	Układ Resetu i Watch-Doga	6
4.3	Pamięć	6
4.4	Dekoder adresów	6
4.5	Klawiatura i wyświetlacz	7
4.6	Wyjście akustyczne	7
4.7	Łączy szeregowo RS232	8
4.8	Gniazdo wejść/wyjść cyfrowych	8
4.9	Izolowane wejścia i wyjścia	8
4.10	System przerwań	8
4.11	Zewnętrzna magistrala systemowa	9
4.12	Klawiatura modułu DSM-51/K	9
4.13	Wyświetlacz LCD	9
4.14	Wejścia analogowe	10
4.15	Wyjście analogowe	10
4.16	Zasilacz	10
5	Uruchamianie oprogramowania w systemie DSM-51	10
5.1	Konfiguracja systemu	10
5.2	Uruchamianie programu w EPROMie	11
5.3	Wykorzystanie symulatora 'picco-512'	11
6	Standardowa konfiguracja systemu	11
6.1	Dekoder adresów	11
6.2	Sterownik przerwań	13
7	Przykładowe programy	13
7.1	Program speaker.asm	14
7.2	Program segmenty.asm	14
7.3	Program kb-lcd.asm	14
7.4	Program ad-da.asm	14
7.5	Programy rs232.asm i rs232.c	14



7.6 Program demo.asm	14
A Zbiory JEDEC standardowej konfiguracji	15
A.1 Zbiór addr-dec.jed	15
A.2 Zbiór int-ctrl.jed	17
B Listingi przykładowych programów	19
B.1 Program speaker.asm	19
B.2 Program segmenty.asm	20
B.3 Program kb-lcd.asm	21
B.4 Program ad-da.asm	23
B.5 Program rs232.asm	25
B.6 Program rs232.c	26
B.7 Program demo.asm	27
C Opis układu HD44780	
D Rysunki, Schematy	



1 Charakterystyka systemu DSM-51

Dydaktyczny System Mikroprocesorowy DSM-51 jest uniwersalnym sterownikiem zbudowanym w oparciu o popularny mikroprocesor jednoukładowy z rodziny 8051.

Składa się on z trzech modułów:

DSM-51 - podstawowa wersja systemu,

DSM-51/K - moduł dodatkowej klawiatury i wyświetlacza LCD,

DSM-51/A - moduł wejść/wyjść analogowych.

Podstawowa wersja systemu jest wyposażona w 24 dwukierunkowe linie wejść/wyjść cyfrowych, co daje szerokie możliwości sterowania układów zewnętrznych. Dodatkowe wejścia i wyjścia z izolacją galwaniczną przeznaczone są do sterowania układów, które nie mogą być bezpośrednio połączone z systemem.

Do komunikacji z użytkownikiem służy klawiatura (6 miniaturowych klawiszy), wyświetlacz (6 wskaźników siedmiosegmentowych), 6 diod świecących oraz miniaturowy głośniczek. Komunikacja z innymi systemami mikroprocesorowymi jest możliwa za pośrednictwem 2 kanałów szeregowych w standardzie RS232. Mogą one w szczególności być wykorzystane do sterowania systemu przez komputer.

DSM-51 jest ponadto wyposażony w układ Watch-Dog. Układ ten, gdy jest aktywny, wykonuje reset całego systemu jeśli praca procesora zostanie zakłócona. Pozwala to na pisanie i testowanie oprogramowania odpornego na zakłócenia.

W systemie zastosowano dwa programowalne układy logiczne typu GAL (Generic Array Logic) umieszczone w podstawkach. Jeden z nich jest przeznaczony do wytwarzania sygnałów sterujących poszczególnymi elementami systemu (sygnały CS, RD, WR). Drugi GAL jest wykorzystany do realizacji systemu przerwań. Takie rozwiązanie pozwala na różnorodne konfigurowanie otoczenia mikroprocesora znacznie rozszerzając możliwości wykorzystania systemu. Możliwa jest na przykład zmiana struktury pamięci systemu w czasie jego pracy, co pozwala na uruchamianie oprogramowania pod nadzorem programu monitorującego.

DSM-51 jest systemem otwartym. Może on być rozbudowywany poprzez dołączanie nowych urządzeń peryferyjnych do wyprowadzonej na zewnątrz magistrali systemowej.

Dla zastosowań wymagających lepszej komunikacji z użytkownikiem przewidziano wyposażenie systemu w moduł DSM-51/K. Moduł ten zawiera dodatkową klawiaturę (16 klawiszy) oraz wyświetlacz LCD (2 linie po 40 znaków) sterowany przez układ HD44780 firmy HITACHI.

Moduł DSM-51/A zawiera 8-bitowy przetwornik A/C obsługujący 8 wejść analogowych oraz 8-bitowy przetwornik C/A wytwarzający jeden analogowy sygnał wyjściowy. Moduł ten może znaleźć zastosowanie przy współpracy systemu z analogowymi urządzeniami sterującymi i kontrolno-pomiarowymi.

2 Przeznaczenie systemu DSM-51

Podstawowym przeznaczeniem Dydaktycznego Systemu Mikroprocesorowego DSM-51 jest wyposażenie laboratoriów technik mikroprocesorowych na wyższych uczelniach technicznych. Może on również znaleźć zastosowanie w pracowniach komputerowych szkół średnich ogólnokształcących i technikach o profilu elektronicznym.

DSM-51 umożliwia wykonywanie przez studentów zadań o różnorodnym poziomie trudności.

Do stosunkowo prostych zadań można zaliczyć pisanie krótkich programów obsługujących poszczególne elementy systemu przy wykorzystaniu typowej konfiguracji wpisanej przez producenta systemu do układów GAL. Przykłady gotowych programów tego typu są omówione w rozdz. 6.

Znacznie bardziej złożone jest pisanie większych programów wykonujących zadania wymagające współpracy z komputerem oraz wykorzystania wszystkich zasobów systemu. Różnorodność tego typu zadań zależy wyłącznie od inwencji osoby prowadzącej laboratorium.



Niektóre zaawansowane zadania programowe mogą wymagać samodzielnego zaprojektowania układów GAL. Przykładem takiego zadania może być opracowanie programu monitora pozwalającego śledzić pracę innego programu pracującego w systemie. Wymagana jest wtedy możliwość wykonywania kodu programu monitorowanego umieszczonego w pamięci RAM w jego normalnym obszarze adresowym (najczęściej od adresu 0000). Ponieważ prawidłowe rozpoczęcie pracy programu monitora wymaga obecności w tym obszarze pamięci EPROM, zatem konieczne jest przełączenie obszarów zajmowanych przez pamięć RAM i EPROM w czasie pracy programu. Problem ten można rozwiązać poprzez odpowiednie zaprojektowanie jednego z układów GAL.

System DSM-51 może być również wykorzystany do prowadzenia zajęć projektowych, w których głównym zadaniem studentów jest opracowanie, wykonanie i uruchomienie modeli różnych urządzeń sterowanych przez system.

Prawidłowe wykonanie różnych tego typu zadań przez studentów wymaga opanowania następujących umiejętności:

- projektowanie konfiguracji pamięci systemu mikroprocesorowego,
- projektowanie systemu przerwań,
- projektowanie programowalnych układów logicznych (GAL),
- oprogramowanie mikroprocesora 8051,
- programowanie układów GAL i pamięci EPROM,
- uruchamianie systemu mikroprocesorowego,
- oprogramowanie kanału RS232 od strony mikroprocesora i komputera,
- programowa obsługa wybranych elementów systemów mikroprocesorowych - wyświetlacz, klawiatura, przetworniki A/C, C/A, specjalizowane układy we/wy, Watch-Dog itp.,
- projektowanie układów wykonawczych współpracujących z systemem mikroprocesorowym.

3 Dane techniczne

- | | |
|---------------|-----------------------------------|
| • wymiary | - 240 x 260 x 40 mm |
| • zasilanie | - 9V |
| • pobór prądu | - 500 mA (zapalone wyświetlacze) |
| | - 150 mA (wygaszone wyświetlacze) |

DSM-51

- | | |
|-------------------------------|-----------------------|
| • wymiary | - 222 x 120 mm |
| • mikroprocesor | - Intel 8031 |
| • zegar | - 10 MHz |
| • pamięć EPROM | - 27C256 (32 kB) |
| • pamięć RAM | - 62256 (32 kB) |
| • stała czasowa Watch-Doga | - ok. 250 ms |
| • RS232 | - 2 kanały |
| • dwukierunkowe we/wy cyfrowe | - 24 linie |
| • izolowane wejścia cyfrowe | - 2 linie |
| • izolowane wyjścia cyfrowe | - 2 linie |
| • klawiatura | - 6 klawiszy |
| • wyświetlacz | - 6 cyfr |
| • diody świecące | - 6 diod |
| • wyjście akustyczne | - miniaturowy głośnik |



DSM-51/K

- wymiary płytki - 165 x 52 mm
- wymiary wyświetlacza - 182 x 34 mm
- klawiatura - 16 klawiszy
- wyświetlacz - LM018L (LCD 2 x 40 znaków)
- sterownik wyświetlacza - HITACHI - HD44780
- zasilanie - +5V z płyty DSM-51

DSM-51/A

- wymiary - 58 x 98 mm
- wejścia analogowe
 - liczba wejść - 8
 - zakres przetw. napięcia - 0...5V
 - przetwornik A/C - ADC0804
 - rozdzielczość - 8 bitów
 - czas przetwarzania - 100 μ s
- wyjścia analogowe
 - liczba wyjść - 1
 - zakres napięć wyjściowych - 0...5V
 - przetwornik C/A - DAC08
 - rozdzielczość - 8 bitów
 - czas ustalania przetwornika - 85 ns
- zasilanie - +8V z płyty DSM-51

4 Budowa systemu

Dydaktyczny System Mikroprocesorowy DSM-51 jest wykonany w postaci trzech płytek drukowanych, które wraz z wyświetlaczem LCD są zmontowane na aluminiowej płycie konstrukcyjnej.

Podstawowym modulem systemu jest płyta DSM-51, która stanowi samodzielny system mikroprocesorowy. Może ona być z powodzeniem wykorzystana bez pozostałych modułów systemu. Zawiera ona prostą klawiaturę i wyświetlacz siedmiosegmentowy, co jest wystarczające dla wielu praktycznych aplikacji.

Moduł DSM-51/K zawiera dodatkową rozbudowaną klawiaturę (16 klawiszy) i układy komunikacji systemu ze sterownikiem wyświetlacza LCD. Wyświetlacz jest zamontowany bezpośrednio do płyty konstrukcyjnej systemu.

Moduł DSM-51/A zawiera 8 wejść analogowych z 8 bitowym przetwornikiem A/C oraz jedno wyjście analogowe z 8 bitowym przetwornikiem C/A.

4.1 Mikroprocesor

W systemie DSM-51 zastosowano mikroprocesor Intel 8031 (U1). Jest to pozbawiona wewnętrznej pamięci ROM wersja 8-bitowego mikroprocesora jednokładowego 8051.

Mikroprocesor 8031 oprócz jednostki arytmetycznej zawiera również:

- 128 bajtów pamięci RAM,
- 2 timery 16 bitowe,
- układ transmisji szeregowej RS232,



- 4 dwukierunkowe porty we/wy (2 z nich są wykorzystane do obsługi zewnętrznej pamięci),
- 2 poziomowy system przerwań.

Wyprowadzona na jeden z portów szyna danych mikroprocesora jest multipleksowana z młodszym bajtem adresu (A0..A7). Pojawiający się chwilowo bajt adresu jest zapamiętywany w buforze U2 (74HC573) sterowanym sygnałem ALE z mikroprocesora.

4.2 Układ Resetu i Watch-Doga

Układ Resetu i Watch-Doga jest zbudowany na 5 inwerterach układu U8 (74HC14).

Po włączeniu napięcia zasilającego system rozpoczyna się ładowanie kondensatora C5 przez rezystor R3. Niskie napięcie na C5 wymusza stan wysoki na wyjściu negatora U8D (reset systemu). Osiągnięcie przez napięcie na C5 progu przełączania układu U8D (ok. 2.7V) kończy reset systemu.

Przycisk K1 umożliwia ręczne resetowanie systemu. Zwora Z2 jest przeznaczona do podłączenia zewnętrznego sygnału resetującego system.

Ustawienie zwory Z1 w pozycji WDON uaktywnia układ Watch-Dog. Występuje wtedy ładowanie kondensatora C4 przez rezystor R2. Gdy napięcie na C4 przekroczy próg przełączania układu U8C, nastąpi reset systemu. Aby utrzymać napięcie na C4 poniżej tego progu, procesor musi generować impulsy na PP4 (pin P1.4 procesora). Na PP4 powinien występować przebieg, w którym czasy trwania stanów '0' i '1' wynoszą co najmniej 200 μ s, a odstępy między kolejnymi ujemnymi zboczami są nie większe od 200 ms.

4.3 Pamięć

W mikroprocesorze 8031 przewidziano dwa rozdzielone obszary pamięci zewnętrznej - pamięć programu i pamięć danych. Każdy z tych obszarów ma rozmiar 64 kB (16 linii adresowych). Do odczytu pamięci programu służy sygnał PSEN generowany przez mikroprocesor. Pamięć danych może być zarówno odczytywana jak i zapisywana. Służą do tego odpowiednio sygnały RD i WR z mikroprocesora.

Wszystkie układy, które mają być obsługiwane przez mikroprocesor muszą być umieszczone w jednym z tych obszarów.

System DSM-51 zawiera:

- 32 kB pamięci programu - EPROM 27C256 (U3),
- 32 kB pamięci danych - RAM 62256 (U4).

Pamięć EPROM umieszczona jest w podstawce aby umożliwić jej programowanie poza systemem. Standardowo pamięć EPROM zajmuje pierwsze 32 kB obszaru pamięci programu, a pamięć RAM pierwsze 32 kB obszaru pamięci danych.

Sygnały sterujące pamięciami są wytwarzane w układzie U6 (GAL 16V8). Zastosowanie do tego celu programowalnego układu logicznego typu GAL pozwala modyfikować rozmieszczenie pamięci w przestrzeni adresowej mikroprocesora.

4.4 Dekoder adresów

Dekoder adresów systemu DSM-51 jest zbudowany na układach U6 i U7. Na linie wejściowe układu U6 typu GAL16V8 podane są:

- wybrane linie adresowe (A6, A7, A12..A15),



- sygnały sterujące z mikroprocesora (WR, RD, PSEN),
- stan pinu P1.7 procesora.

Na podstawie tych sygnałów układ U6 wytwarza następujące sygnały sterujące:

- CSE - wybór pamięci EPROM,
- RDR - odczyt pamięci RAM,
- CSR - wybór pamięci RAM,
- CSX - wybór magistrali zewnętrznej,
- RDX - odczyt z magistrali zewnętrznej,
- WRX - zapis do magistrali zewnętrznej,
- CSIO - wybór pozostałych układów we/wy,
- LCD - wybór wyświetlacza LCD.

Układ U7, gdy zostanie wybrany sygnałem CSIO, wytwarza kolejne sygnały sterujące:

- CSIC - wybór sterownika przerwań,
- CSDA - wybór przetwornika cyfrowo/analogowego,
- CSAD - wybór przetwornika analogowo/cyfrowego,
- CSMX - wybór multipleksa analogowego,
- CSKB - wybór klawiatury modułu DSM-51/K,
- CS55 - wybór układu 8255,
- CSDS - wybór układu przeglądania wyświetlaczy i klawiatury,
- CSDB - wybór bufora wyświetlaczy.

4.5 Klawiatura i wyświetlacz

Układ wyświetlacza wykorzystuje metodę sekwencyjnego przeglądania kolejnych wskaźników. System DSM-51 jest wyposażony w 6 siedmiosegmentowych wskaźników ze wspólną anodą.

Do załączania kolejnych wskaźników służy dekodery U18 sterowany poprzez bufor U17. Natomiast dane, wpisane do bufora U15 i wzmocnione w układzie U16, określają które segmenty wskaźnika zostaną zapalone. Doprowadzenie sygnału PP6 do dekodera U18 gwarantuje wygaszenie wyświetlacza przy resecie systemu. Sygnał ten może być również wykorzystany przy przeglądaniu do wygaszania wyświetlacza na czas zmiany danych.

Układy sterowania wyświetlaczem obsługują dodatkowo 6 diod świecących, które są włączone jako siódmy wskaźnik.

Jednocześnie z przeglądaniem wyświetlacza przeglądana jest klawiatura. Wciśnięcie klawisza powoduje pojawienie się stanu 0 na wejściu T1 mikroprocesora, gdy zostanie wybrany odpowiedni wskaźnik.

4.6 Wyjście akustyczne

System wyposażony jest w miniaturowy głośnik sterowany sygnałem PP5 (pin P1.5 procesora). Kondensator C11 zabezpiecza przed włączeniem na stałe prądu głośnika, co prowadziłoby do jego uszkodzenia.



4.7 Łącza szeregowo RS232

System jest wyposażony w dwa łącza RS232. Do dopasowania poziomów sygnałów do standardu RS232 służy układ U9 (MAX232).

Do gniazda G1 doprowadzone są sygnały TXD i RXD mikroprocesora. Linie te są obsługiwane przez wewnętrzny układ transmisji szeregowo mikroprocesora 8031.

Do gniazda G2 doprowadzony jest sygnał PP0 (pin P1.0 procesora) jako wyjście. Sygnał wejściowy z gniazda G2 jest doprowadzony do układu U5 (system przerwań). Takie rozwiązanie umożliwia zrealizowanie przerwaniowej obsługi transmisji szeregowo przez gniazdo G2.

4.8 Gniazdo wejść/wyjść cyfrowych

Do gniazda G6 doprowadzone są 24 dwukierunkowe linie cyfrowe z układu U10 (8255) oraz napięcie zasilające V8. Gniazdo to jest przeznaczone do podłączania zewnętrznych układów sterowanych przez system.

Układ U10, przy odpowiednim sterowaniu, może realizować równoległe przesyłanie danych przez porty A i B. Sygnały sterujące są wtedy obsługiwane przez port C. Przy tej konfiguracji, na pinach PC0 i PC3 pojawiają się sygnały przerwań do mikroprocesora, odpowiedzialne odpowiednio za port B i A. Piny te są włączone do systemu przerwań jako sygnały IPB i IPA.

Maksymalny prąd zasilania pobierany przez urządzenia podłączone do gniazda G5 i G6 łącznie nie może przekraczać 0.5A.

4.9 Izolowane wejścia i wyjścia

System posiada dwa wejścia i dwa wyjścia izolowane galwanicznie.

Wejścia są przystosowane do sterowania ich z układów OC pracujących przy napięciu 5V. Stanowią one obciążenie odpowiadające pojedynczemu wejściu TTL. Jedno z wejść steruje sygnał T0 procesora, a drugie podane jest do systemu przerwań jako sygnał IOI.

Wyjścia są wykonane poprzez wyprowadzenie kolektora i emitera fototranzystora. Fototranzystor w stanie włączenia może przewodzić prąd ok. 5 mA. Wyjścia są sterowane sygnałami PP2 i PP3 z mikroprocesora.

4.10 System przerwań

Sterownik przerwań jest zbudowany na układzie U5 (GAL16V8).

Jest on obsługiwany przez mikroprocesor tak jak inne układy we/wy. W tym celu są do niego doprowadzone sygnały sterujące RD, WR, CSIC oraz D0 i D1 z szyny danych. Sygnał PP1 z mikroprocesora jest podłączony aby umożliwić modyfikację systemu przerwań w czasie pracy systemu (np. na czas obsługi przerwania).

Dwa wyjścia z układu GAL podłączone są do wejść przerwań INT0 i INT1 mikroprocesora.

Fragmenty struktury GAL związane z czterema nie wykorzystanymi wyjściami pozostają do dyspozycji użytkownika. Można na nich zrealizować zapamiętywanie przerwań (poprzez 'zatrzasknięcie' bramki). W ten sposób istnieje możliwość zapamiętania przerwań z 4 linii wejściowych, i uaktywnienie przerwania mikroprocesora przy wystąpieniu któregośkolwiek z nich.

Przy obsłudze przerwania mikroprocesor może dowiedzieć się, które urządzenie rząda obsługi dokonując odczytu numeru przerwania (przez szynę danych), zgodnie z zaprojektowanym priorytetem przerwań. Skasowanie zapamiętanej w GALu informacji może być realizowane przez zapis do sterownika przerwań z podaniem na D0, D1 numeru obsługiwanej przerwy.



Tak zrealizowany sterownik przerwań będzie prawidłowo funkcjonował, jeśli przerwanie będzie krótkim impulsem, lub jeśli najpóźniej w czasie obsługi przerwania nastąpi zmiana stanu linii wymuszającej przerwanie.

Opisany tu sposób obsługi przerwań nie może być bezpośrednio zastosowany do przerwania z drugiego kanału RS232. Obsługę tego przerwania można wykonać przez jego blokowanie sygnałem PP1 (z jednoczesnym umożliwieniem odczytu stanu na linii IRS) na czas odbioru bajtu z kanału.

Do układu U5 dochodzą następujące sygnały przerwań:

IRS - wejście z gniazda G2 (drugi kanał RS232),

IAD - sygnał końca przetwarzania przetwornika A/C,

IOI - jedno z wejść izolowanych galwanicznie,

IX - wejście z gniazda G5 (magistrala systemowa),

IPB - sygnał przerwania z układu 8255 (port B),

IPA - sygnał przerwania z układu 8255 (port A).

4.11 Zewnętrzna magistrala systemowa

Gniazdo zewnętrznej magistrali systemowej (G5) umożliwia rozbudowę systemu.

Szyna adresowa i szyna danych systemu są doprowadzone do gniazda poprzez bufony U11, U12 i U13. Poza tym do gniazda G5 doprowadzone są sygnały sterujące zewnętrznej magistrali systemowej (CSX, RDX i WRX). Pin 27 gniazda G5 połączony jest do systemu przerwań (sygnał IX). Napięcie V8 występujące na G5 jest przeznaczone do zasilania urządzeń zewnętrznych.

Maksymalny prąd zasilania pobierany przez urządzenia podłączone do gniazda G5 i G6 łącznie nie może przekraczać 0.5A.

4.12 Klawiatura modułu DSM-51/K

Klawiatura modułu DSM-51/K składa się z 16 klawiszy. Są one połączone w układzie 2 x 8. Bufor odczytu klawiatury jest uaktywniany sygnałem CSKB generowanym przez dekodery adresów. Natomiast odczytywany zestaw klawiszy jest wybierany sygnałami adresowymi A0 i A1.

4.13 Wyświetlacz LCD

W systemie zastosowano wyświetlacz LM018L produkcji HITACHI. Jest to wyświetlacz ciekłokrystaliczny (2 linie po 40 znaków) obsługiwany przez sterownik HD44780.

Sterownik HD44780 jest przystosowany do sygnałów sterujących mikroprocesorów rodziny 6800. W systemie DSM-51 podłączono go w ten sposób, że linia adresowa A0 steruje wejściem RS (rozkaż/dane) sterownika, a linia A1 wejściem R/W (odczyt/zapis) sterownika. Sygnał Enable sterownika, który powinien pojawiać się przy stabilnych stanach wejść R/W i RS, jest wytwarzany w dekodzie adresów (sygnał LCD).

W celu spełnienia warunku stabilności wejść R/W i RS sygnał LCD należy uzależnić nie tylko od odpowiednich linii adresowych, ale również od sygnałów sterujących RD i WR.

Przy takim podłączeniu sterownik jest 'widziany' przez mikroprocesor jako cztery kolejne komórki pamięci. Do dwóch z tych komórek można tylko pisać, a dwie pozostałe można tylko odczytywać.

Rezystory R2 i R3 ustalają napięcie sterujące kontrastem wyświetlacza.



4.14 Wejścia analogowe

System ma 8 wejść analogowych (moduł DSM-51/A).

Sygnały analogowe z gniazda G1 (piny 1...8) są podawane na multiplexer U1 przez rezystory zabezpieczające R1...R8. Wpisanie numeru wejścia do bufora U2 (wybieranego sygnałem CSMX) powoduje podłączenie tego wejścia do 8 bitowego przetwornika analogowo cyfrowego U3 (ADC0804). Napięcie wejściowe jest przetwarzane w zakresie 0...5V.

Przetwornik jest wybierany sygnałem CSAD generowanym w dekodrze adresów. Wyjście przerwań przetwornika jest podłączone do systemu przerwań (sygnał IAD).

4.15 Wyjście analogowe

System ma jedno wyjście analogowe (moduł DSM-51/A).

Bajt, który ma być przetworzony na sygnał analogowy, jest wpisywany do bufora U4 (wybieranego sygnałem CSDA). Linie wejść cyfrowych przetwornika cyfrowo analogowego U5 (DAC08) są sterowane bezpośrednio z wyjść bufora U4. Napięcie z zakresu 0...5V, odpowiadające wpisanemu bajtowi, występuje na wyjściu wzmacniacza operacyjnego U6A. Jest ono podawane na pin 11 gniazda G1.

Układ U7 wytwarza ujemne napięcie o wartości około -5V niezbędne do pracy przetwornika U5.

4.16 Zasilacz

System DSM-51 wymaga podłączenia pojedynczego napięcia zasilającego o wartości 9V. Dioda D10 zabezpiecza system przed odwrotną polaryzacją napięcia zasilającego.

Stabilizator U14 wytwarza napięcie 5V niezbędne do zasilania większości układów systemu. Elementy pobierające większe moce (wyświetlacz, głośnik) są zasilone z napięcia V8 (ok. 8V) występującego przed stabilizatorem. Napięcie V8 jest doprowadzone również do gniazd G4, G5, G6.

Moduł wejść/wyjść analogowych DSM-51/A jest zasilany z płyty DSM-51 napięciem V8. Stabilizowane napięcie 5V zasilające przetworniki jest wytwarzane przez stabilizator U8 (78L05).

Oddzielenie zasilania przetworników od zasilania pozostałej części systemu pozwala uniknąć przenikania zakłóceń do części analogowej systemu po liniach zasilających.

5 Uruchamianie oprogramowania w systemie DSM-51

5.1 Konfiguracja systemu

Przed przystąpieniem do pisania programu należy dopasować konfigurację systemu do potrzeb realizowanego zadania. Polega to na zaprojektowaniu dekodera adresów i systemu przerwań, które są zrealizowane z wykorzystaniem programowalnych układów logicznych typu GAL. Prawidłowe zaprojektowanie tych elementów systemu może znacznie ułatwić późniejsze programowanie.

Układy GAL można zaprojektować samodzielnie lub posłużyć się w tym celu jednym z assemblerów. Przy samodzielnym projektowaniu GALa otrzymujemy plan połączeń jakie należy wykonać w strukturze układu. W wyniku zastosowania assemblera otrzymujemy zbiór w przyjętym dla tego typu układów formacie JEDEC.

Po zaprojektowaniu układów GAL należy je zaprogramować wykorzystując programator GALi. Układy GAL w przeciwieństwie do układów typu PAL mogą być programowane wielokrotnie.

Prawidłowość projektu konfiguracji można sprawdzić pisząc krótkie programy obsługi poszczególnych elementów systemu.



5.2 Uruchamianie programu w EPROMie

Uruchomienie przygotowanego programu polega na wpisaniu jego treści do pamięci EPROM, umieszczeniu jej w podstawce na płycie DSM-51 i załączeniu zasilania systemu.

Naciśnięcie w dowolnym momencie przycisku RESET spowoduje przerwanie pracy programu. Po jego zwolnieniu program wznowi swoją pracę od początku.

Przed zaprogramowaniem nowego programu do pamięci EPROM należy ją skasować (kilkuminutowe naświetlanie promieniami ultrafioletowymi). Pamięci EPROM mogą być programowane wielokrotnie, jednak po około 100 cyklach programowania mogą zacząć pojawiać się błędy.

5.3 Wykorzystanie symulatora 'picco-512'

Uruchamianie programów można znacznie przyspieszyć stosując symulator pamięci EPROM. Symulator to sterowane z komputera urządzenie włączane w podstawkę EPROMu w uruchamianym systemie.

Zastosowanie symulatora pozwala uniknąć czasu niezbędnego na kasowanie EPROMu i jego ponowne programowanie. Uruchomienie nowej wersji programu polega na przesłaniu jej do pamięci RAM znajdującej się w symulatorze, włączeniu symulacji i wykonaniu resetu w systemie. Operacja ta jest znacznie krótsza od kasowania i programowania EPROMu.

Drugą istotną zaletą zastosowania symulatora jest uniknięcie zużycia podstawki spowodowanego wielokrotnym wyjmowaniem i wkładaniem EPROMu.

Do uruchamiania i testowania oprogramowania w systemie DSM-51 można wykorzystać symulator 'picco-512' produkowany przez firmę MicroMade. Symulator ten może symulować EPROMy 2716...27512. Jest on sterowany przez łącze szeregowe RS232 z prędkością 57600 bodów.

Symulator 'picco-512' automatycznie generuje sygnał RESET przy ładowaniu do jego pamięci nowej wersji programu. Generowany reset polega na zwarciu przez rezystor $1k\Omega$ pinów gniazdka znajdującego się z boku symulatora. Podłączenie tego sygnału do zwory Z2 systemu ogranicza uruchomienie nowej wersji programu jedynie do jej przesłania do symulatora i włączenia symulacji.

6 Standardowa konfiguracja systemu

Standardowa konfiguracja systemu jest wpisana przez producenta systemu do układów GAL.

6.1 Dekoder adresów

Zależności opisujące standardowy dekodek adresów są przedstawione poniżej.

Sygnał wyboru EPROMu:

$$CSE = PSEN$$

Sygnał odczytu RAMu:

$$RDR = RD$$

Sygnał wyboru RAMu:

$$CSR = A15$$

Sygnał odczytu na magistrali zewnętrznej:

$$RDX = RD$$

Sygnał zapisu do magistrali zewnętrznej:

$$WRX = WR$$



MicroMade

Systemy Mikroprocesorowe

ul. Sikorskiego 33
64-920 PILA
tel/fax:(67)13-24-14

Biuro Konstrukcyjne:
ul. Fiszer 14, 80-952 GDAŃSK
tel: (58)47-15-35, fax: (58)41-61-44

Sygnal wyboru magistrali zewnętrznej:

$$CSX = \overline{A_{15}} * A_{14} * A_{13} * A_{12} * A_7 * \overline{A_6}$$

Sygnal wyboru pozostałych urządzeń we/wy:

$$CSIO = A_{15} * A_{14} * A_{13} * A_{12} * \overline{A_7} * \overline{A_6}$$

Sygnal wyboru wyświetlacza LCD:

$$LCD = A_{15} * A_{14} * A_{13} * A_{12} * A_7 * \overline{A_6} * (\overline{RD} + \overline{WR})$$

Uwagi!

1. Ze względu na obciążenie sygnału WR mikroprocesora do modułu DSM-51/A jest doprowadzony sygnał WRX zamiast WR. Należy o tym pamiętać projektując układ GAL.
2. Rozróżnienie adresów urządzeń we/wy jedynie przez część młodszych adresów ($A_3 \dots A_7$) umożliwia obsługę tych urządzeń rozkazami podającymi na szynę adresową tylko 8 bitowy adres (np: MOVX @R0,A). Starszą część adresu należy na stałe umieścić w porcie 2 mikroprocesora. Wykorzystanie tego sposobu adresowania jest zdecydowanie prostsze niż korzystanie za każdym razem z 16 bitowego rejestru DPTR.

Przy standardowej konfiguracji systemu pamięć EPROM zajmuje adresy 0000H...7FFFH obszaru pamięci programu, a pamięć RAM adresy 0000H...7FFFH obszaru pamięci danych.

Urządzenia we/wy zajmują końcowe adresy obszaru pamięci danych mikroprocesora. Przy ich adresowaniu starszy bajt adresu powinien mieć wartość 1111XXXXB (w programach przyjęto 0FFH). Wymagane wartości młodszych bajtów adresów, oraz przyjęte w programach wartości (HEX), poszczególnych urządzeń są zebrane w tabeli (sybol X oznacza dowolną wartość danego bitu).

adres	HEX	symbol	sygnał	urządzenie
00000XXXXB	00H	CSIC	CSIC	kontroler przerwań
00001XXXXB	08H	CSDA	CSDA	przetwornik C/A
00010XXXXB	10H	CSAD	CSAD	przetwornik A/C
00011XXXXB	18H	CSMX	CSMX	multiplexer analogowy
00100X01B	21H	CSKB0	CSKB	klawisze 0...7 (DSM-51/K)
00100X10B	22H	CSKB1	CSKB	klawisze 8...(DSM-51/K)
00101XXXXB	28H	CS55	CS55	układ 8255
00110XXXXB	37H	CSDS	CSDS	bufor wyboru wskaźnika
00111XXXXB	38H	CSDB	CSDB	bufor danych wskaźnika
10XXXX00B	80H	LCDWC	LCD	HD44780 - wpis rozkazów
10XXXX01B	81H	LCDWD	LCD	HD44780 - wpis danych
10XXXX10B	82H	LCDRC	LCD	HD44780 - odczyt stanu
10XXXX11B	83H	LCDRD	LCD	HD44780 - odczyt danych
11XXXXXXB	C0H	CSX	CSX	zewnętrzna magistrala

Zawartość GALa dekodera adresów (U6) jest opisana zbiorem addr-dec. jed w formacie JEDEC. Treść tego zbioru jest zawarta w dodatku A.



6.2 Sterownik przerwań

Przy standardowej konfiguracji systemu w sterowniku przerwań następuje zatrzymywanie (zgodnie z zasadą przedstawioną w rozdziale 4.10) czterech sygnałów przerwań: IAD, IOI, IPA, IPB. Wystąpienie impulsu na jednej z tych linii powoduje wygenerowanie przerwania INT1 do mikroprocesora.

Dodatkowym warunkiem dla zatrzaśnięcia przerwań IPA i IPB jest stan '0' na nodze PP1. Umożliwia to wykorzystanie sterownika przerwań bez konieczności wcześniejszego ustawiania układu 8255.

Natomiast przerwanie IRS podawane jest bezpośrednio na wejście INT0 mikroprocesora.

Należy pamiętać o prawidłowym sterowaniu trójstanowych wejść/wyjść D0 i D1 GALa. Mogą one być wyjściami tylko przy jednoczesnym wystąpieniu sygnałów CSIC i RD. Błąd wysterowania tych wyjść uniemożliwi prawidłową pracę systemu.

Zależności opisujące standardowy sterownik przerwań są przedstawione poniżej.

Pamięć przerwania IAD:

$$O_0 = (\overline{D_0} * \overline{D_1} * \overline{WR} * \overline{SCIC} * IAD) + (O_0 * IAD)$$

Pamięć przerwania IOI:

$$O_1 = (D_0 * \overline{D_1} * \overline{WR} * \overline{SCIC} * IOI) + (O_1 * IOI)$$

Pamięć przerwania IPA:

$$O_2 = (\overline{D_0} * D_1 * \overline{WR} * \overline{SCIC} * IPA) + (O_2 * IPA) + PP1$$

Pamięć przerwania IPB:

$$O_3 = (D_0 * D_1 * \overline{WR} * \overline{SCIC} * IPB) + (O_3 * IPB) + PP1$$

Sygnał przerwania INT0 procesora:

$$INT0 = IRS$$

Sygnał przerwania INT1 procesora:

$$INT1 = O_0 * O_1 * O_2 * O_3$$

Sygnały wyjściowe (te równania decydują o priorytecie przerwań):

$$D_0 = O_0 * \overline{O_1} + O_0 * O_2$$

$$D_1 = O_0 * O_1$$

Sygnały sterujące trójstanowymi buforami wyjściowymi:

$$D_0.TRST = \overline{RD} * \overline{SCIC}$$

$$D_1.TRST = \overline{RD} * \overline{SCIC}$$

Zawartość GALa kontrolera przerwań (U5) jest opisana zbiorem `int-ctrl.jed` w formacie JEDEC. Treść tego zbioru jest zawarta w dodatku A.

7 Przykładowe programy

W załączonych krótkich programach są zawarte proste przykłady obsługi poszczególnych elementów systemu.

Celem każdego z załączonych programów jest zademonstrowanie najprostszej metody obsługi wybranego elementu systemu. Programy są napisane w sposób przejrzysty, z szerokim komentarzem, a ich długość jest ograniczona do niezbędnego minimum. Treść przykładowych programów jest zamieszczona w dodatku B.



7.1 Program speaker.asm

Program demonstruje obsługę wyjścia akustycznego systemu i klawiatury płyty DSM-51.

Program początkowo generuje ton o częstotliwości 20kHz. Po każdym naciśnięciu klawisza K2 (pierwszy z prawej strony na płycie DSM-51) następuje dwukrotne obniżenie generowanej częstotliwości.

Program nie zawiera procedur tłumienia drgań styków klawisza.

7.2 Program segmenty.asm

Program demonstruje obsługę wskaźnika siedmiosegmentowego.

Realizuje on wybieranie sekwencyjne kolejnych wskaźników. Dla prostoty programu zapala on na wszystkich wskaźnikach ten sam segment. Przeglądany jest również wskaźnik siódmy (zespół diod świecących). Po stu cyklach przeglądu wszystkich wskaźników program zapala kolejny segment.

7.3 Program kb-lcd.asm

Program demonstruje obsługę klawiatury modułu DSM-51/K oraz wyświetlacza LCD.

Program, po programowej inicjalizacji sterownika wyświetlacza LCD, oczekuje na naciśnięcie klawisza. Gdy stwierdzi naciśnięcie klawisza wypisuje na wskaźniku kod klawisza i czeka na jego zwolnienie. Klawisz 'Enter' kasuje wyświetlacz i ustawia kursor na początku pierwszej linii wyświetlacza.

7.4 Program ad-da.asm

Program demonstruje obsługę modułu wejść/wyjść analogowych (DSM-51/A).

Program inicjalizuje pomiar napięcia na pierwszym wejściu analogowych (pin 1 gniazda G1 modułu DSM-51/A). Gdy przetwornik A/C zakończy przetwarzanie uzyskana wartość jest przepisywana do przetwornika C/A i rozpoczyna się kolejny pomiar napięcia na tym samym wejściu. Czynność ta jest powtarzana w pętli bez końca. Program wykorzystuje przerwanie od przetwornika A/C informujące o zakończeniu przetwarzania.

7.5 Programy rs232.asm i rs232.c

Program rs232.asm demonstruje obsługę pierwszego portu szeregowego systemu.

Program, po ustawieniu parametrów kanału rs232 (prędkość 4800, 8 bitów danych, 1 bit stopu) i odblokowaniu przerwania od kanału szeregowego, oczekuje na rozkaz nadania (przysyłanie przez RS znaku '?'). Po odebraniu każdego takiego rozkazu program nadaje hasło 'MicroMade'.

Program rs232.asm współpracuje z programem rs232.c na IBM PC, który w kółko wysyła znak '?' i wypisuje na ekranie odebrany z kanału COM1 tekst (zakończony znakiem nowej linii).

7.6 Program demo.asm

System DSM-51 jest dostarczany z EPROMem zawierającym program demo.asm. Program ten jest połączeniem kilku wyżej omówionych programów.

Program cały czas sekwencyjnie zapala kolejne segmenty wskaźników siedmiosegmentowych. Naciśnięcie któregoś z klawiszy na głównej płycie systemu powoduje zapalenie odpowiedniego LEDa i generowanie dźwięku o odpowiedniej częstotliwości. Naciśnięcie klawisza na klawiaturze modułu DSM-51/K powoduje wpisanie na wyświetlacz LCD odpowiedniego znaku. Klawisz 'Enter' kasuje wyświetlacz i ustawia kursor na początku pierwszej linii wyświetlacza.



A Zbiory JEDEC standardowej konfiguracji

A.1 Zbiór addr-dec.jed

TITLE :DSM-51 - Dekoder Adresow AUTHOR :PAWEL GALKA
PATTERN :Konfiguracja Standardowa COMPANY:MicroMade
REVISION:1.0 DATE :12/07/92

PAL16V8

_DSM_A_S*

QP20*

QF2194*

GO*F0*

L0000

111011111111111111111111111111111111
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000*

L0256

1111111111111111111111111111111011
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000*

L0512

1011111111111111111111111111111111
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000*

L0768

0111011101110111011101111111111111
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000*

L1024

1111111111111111111111111111111011
000000000000000000000000000000000000
000000000000000000000000000000000000
000000000000000000000000000000000000



Biuro Konstrukcyjne:
ul. Fiszer 14, 80-952 GDAŃSK
tel: (58)47-15-35, fax: (58)41-61-44

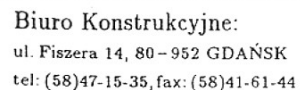
0000



A.2 Zbiór int-ctrl.jed

TITLE :DSM-51 - Sterownik Przerwan AUTHOR :PAWEŁ GALKA
PATTERN :Konfiguracja Standardowa COMPANY:MicroMade
REVISION:1.0 DATE :12/09/92

```
PAL16V8
_DSM_I_S*
QP20*
QF2194*
GO*F0*
L0000
11111111111111111111111111111111
11110111111111111111111111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000*
L0256
11111111111111111111111111111111
111011110111111111111111011101011
11111101011111111111111111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000*
L0512
11111111111111111111111111111111
111011111111011111111111011101011
11111111101011111111111111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000*
L0768
11111111111111111111111111111111
111111111111101111111111011111111
1110111111111111111111110010101011
01111111111111111111111111111111
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000*
L1024
11111111111111111111111111111111
1111111111111111111101011111111111
1110111111111111111111010110101011
01111111111111111111111111111111
00000000000000000000000000000000
00000000000000000000000000000000
```



0000



B Listingi przykładowych programów

B.1 Program speaker.asm

```
.TITLE 'DSM SPEAKER'
;=====

.CHIP 8051

T1 .REG P3.5 ;wejście odczytu klawiszy
SPEAKER .REG P1.5 ;wyjście sterowania głośnika
DS_ON .REG P1.6 ;linia wygaszania wyświetlacza

;Adresy Urządzeń I/O
IO_HIGH EQU OFFH ;1111XXXXB - starszy bajt adresu

CSDS EQU 030H ;00110XXXXB
CSDB EQU 038H ;00111XXXXB

STACK EQU 60H

START:
MOV SP,#STACK ;wskaznik stosu
MOV P2,#IO_HIGH ;starszy bajt adr. na port

;zerowanie bufora wyświetlacza
MOV R1,#CSDB
CLR A
MOVX @R1,A

;umożliwienie odczytu klawisza nr 0 (K2)
MOV R1,#CSDS
MOVX @R1,A
CLR DS_ON

MOV R2,#1 ;początkowy okres dźwięku
RING1:
CLR FO ;flaga wcisnięcia klawisza
RING:
CPL SPEAKER ;zmiana stanu wyjścia głośnika
ACALL OKRES ;opóźnienie
CPL SPEAKER ;zmiana stanu wyjścia głośnika
ACALL OKRES ;opóźnienie

JB T1,RING1 ;klawisz puszczony
JB FO,RING ;klawisz nadal wcisnięty
SETB FO ;nowe wcisnięcie klawisza
MOV A,R2
ADD A,R2 ;okres=okres*2
MOV R2,A
SJMP RING

OKRES:
MOV A,R2 ;opóźnienie w/g R2
```



```
      MOV      R3,A
OK2:   MOV      R4,#15
OK1:   DJNZ     R4,OK1
      DJNZ     R3,OK2
      RET
```

.END

B.2 Program segmenty.asm

```
.TITLE 'DSM SEGMENTY'
;=====

      .CHIP    8051

DS_ON  .REG     P1.6           ;linia wygaszania wyswietlacza

;Adresy Urzadzen I/O
IO_HIGH EQU     OFFH          ;1111XXXXB - starszy bajt adresu

CSDS   EQU     030H           ;00110XXXXB
CSDB   EQU     038H           ;00111XXXXB

STACK  EQU     60H

START:
      MOV      SP,#STACK      ;wskaznik stosu
      MOV      P2,#IO_HIGH

      MOV      R0,#CSDB       ;R0 - adres bufora wyswietlaczy
      MOV      R1,#CSDS       ;R1 - adres wyboru wskaznika

      MOV      R3,#1           ;jeden segment zapalony
      MOV      A,R3
      MOVX     @R0,A
      CLR      DS_ON          ;zapalenie wyswietlacza
      MOV      R4,#100        ;licznik cykli przegladania

PETLA1:
      MOV      R2,#7           ;licznik wskaznikow
PETLA2:
      MOV      A,R2
      DEC      A               ;kolejny wskaznik
      MOVX     @R1,A
      ACALL    TIME            ;opoznienie
      DJNZ     R2,PETLA2       ;petla po wskaznikach
      DJNZ     R4,PETLA1       ;petla po cyklach przegladania
      MOV      R4,#100        ;licznik cykli przegladania
      MOV      A,R3
      RL       A               ;kolejny segment wskaznikow
      MOV      R3,A
      MOVX     @R0,A
```



```
        SJMP     PETLA1

TIME:   MOV      R7,#255           ;opoznienie
TIME_PT: DJNZ     R7,TIME_PT
        RET

.END
```

B.3 Program kb-lcd.asm

```
.TITLE  'DSM KB-LCD'
;=====
.CHIP   8051

;Adresy Urzadzen I/O
IO_HIGH EQU    OFFH      ;1111XXXXB

LCD      EQU    080H      ;10XXXXXXB
LCDWC    EQU    080H      ;10XXXX00B
LCDWD    EQU    081H      ;10XXXX01B
LCDRC    EQU    082H      ;10XXXX10B
LCDRD    EQU    083H      ;10XXXX11B

CSKB     EQU    020H      ;00100XXXB
CSKB0    EQU    021H      ;00100X01B
CSKB1    EQU    022H      ;00100X10B

STACK    EQU    60H

;-----
;start programu
MOV      SP,#STACK        ;wskaznik stosu
MOV      P2,#IO_HIGH      ;starszy adresu dla urzadzen
                        ;wejsc/wyjsc na Port 2
MOV      R0,#LCDWD         ;mlodszy adres dla wyswietlacza LCD
MOV      DPTR,#CODES       ;adres kodow klawiszy dla wyswietlacza

        ACALL   LCD_INIT   ;inicjalizacja wyswietlacza LCD
;-----
;oczekiwanie na naciśnięcie klawisza
LCD_KEY:
        MOV     R1,#CSKB0   ;mlodszy adres dla klawiatury

        MOV     R2,#0       ;8 mlodszych klawiszy
        MOVX    A,@R1       ;odczytanie ich stanu
        CPL     A
        JNZ     TK0         ;klawisz naciśnięty

        MOV     R2,#8       ;8 starszych klawiszy
        INC     R1
        MOVX    A,@R1       ;odczytanie ich stanu
        CPL     A
```



```
JZ      LCD_KEY      ;zaden klawisz nie jest wcisniety

;klawisz nacisniety - identyfikacja klawisza i odpowiednia obsluga
TK0:
    MOV    R3,#7
TK1:
    RLC     A          ;zamiana kodu 1 z 8 na nr klawisza
    JC      TK2
    DJNZ    R3,TK1
TK2:
    MOV     A,R3
    ADD     A,R2      ;klawisze od 0 do 15
    CJNE    A,#15,TK3 ;czy to klawisz ENTER

    MOV     A,#01H    ;obsluga klawisza ENTER
    CALL    LCD_WRC   ;wyczyszczenie wyswietlacza LCD
    SJMP    TK4

TK3:
    MOVC    A,@A+DPTR ;pobranie kodu klawisza
    ACALL   LCD_WRD   ;i wyswietlenie na LCD

;oczekiwanie na puszczenie klawisza
TK4:
    MOVX    A,@R1     ;czy klawisz puszczoney
    CPL     A
    JNZ     TK4        ;nie

    ACALL   TIME      ;klawisz puszczoney - odczekanie czasu
    MOVX    A,@R1     ;powtorne sprawdzenie w celu eliminacji
    CPL     A         ;dragan zestykow
    JNZ     TK4        ;klawisz nadal nacisniety

    SJMP    LCD_KEY   ;czekaj na kolejny klawisz
;-----
CODES:
    DB      30H,31H,32H,33H ;0,1,2,3
    DB      34H,35H,36H,37H ;4,5,6,7
    DB      38H,39H,3CH,3EH ;8,9,lewo,prawo
    DB      5EH,76H,7FH     ;gora,dol,backspace
;-----
;inicjalizacja wyswietlacza LCD
LCD_INIT:
    ACALL   TIME2      ;reset wyswietlacza
    MOV     A,#38H
    CALL    LCD_WRC_NB
    ACALL   TIME2
    CALL    LCD_WRC_NB
    ACALL   TIME
    CALL    LCD_WRC_NB

    CALL    LCD_WRC     ;ustawienie funkcji: interface 8 bit,
                        ;2 linie wyswietlacza, fonty 5*7
```



```
MOV    A,#0FH          ;włączenie: całego wyświetlacza
CALL   LCD_WRC          ;i mrugającego kursora

MOV    A,#01H          ;wyczyszczenie wyświetlacza i ustawienie
CALL   LCD_WRC          ;kursora w pozycji 0

MOV    A,#06H          ;ustawienie przesuwania kursora w prawo
CALL   LCD_WRC
RET

;-----
LCD_WRC:                ;wpisanie kodu rozkazu do LCD
    ACALL WAIT_BUSY
LCD_WRC_NB:             ;wpisanie kodu rozkazu bez oczekiwania
    DEC    R0           ;na flage BUSY
    MOVX   @R0,A
    INC    R0
    RET

LCD_WRD:                ;wpisanie danej do LCD
    ACALL WAIT_BUSY
    MOVX   @R0,A
    RET

WAIT_BUSY:              ;oczekiwanie na flage BUSY = 0
    PUSH   A
    INC    R0
WAIT_BUSY1:
    MOVX   A,@R0
    JB     ACC.7,WAIT_BUSY1
    DEC    R0
    POP    A
    RET

;-----
TIME:                   ;petla czasowa ok. 600 us
    MOV    R7,#0
TPT:
    DJNZ   R7,TPT
    RET

TIME2:                  ;petla czasowa ok. 24 ms
    MOV    R5,#40
T2PT:
    ACALL TIME
    DJNZ   R5,T2PT
    RET

;-----
.END
```

B.4 Program ad-da.asm

```
.TITLE 'DSM AD-DA'
;=====
.CHIP 8051
```



```
;Adresy Urzadzen I/O
IO_HIGH EQU    OFFH    ;1111XXXXB

CSIC EQU    000H    ;00000XXXXB
CSDA EQU    008H    ;00001XXXXB
CSAD EQU    010H    ;00010XXXXB
CSMX EQU    018H    ;00011XXXXB

STACK EQU    60H

        AJMP    START

;-----
        ORG     13H
;przerwanie z wejscia INT1
        MOVX    A,@R1    ;odczyt numeru przerwania
        ANL     A,#03H    ;ze sterownika przerw
        JNZ     OTHER

        MOVX    A,@R0    ;obsługa przerwania z przetwornika A/D
        DEC     R0        ;odczyt wyniku z przetwornika A/D
        MOVX    @R0,A    ;zapis wyniku do przetwornika D/A
        INC     R0

        MOVX    @R0,A    ;inicjalizacja kolejnego pomiaru A/D
        CLR     A

OTHER:
        MOVX    @R1,A    ;skasowanie przerwania w sterowniku przerw
        RETI

;-----
START:
        MOV     SP,#STACK    ;ustawienie wskaźnika stosu

        MOV     R0,#CSMX    ;adres multiplexera analogowego
        CLR     A
        MOVX    @R0,A    ;pomiar przetwornikiem A/D z wejscia 0

        MOV     R1,#CSIC    ;adres sterownika przerw
        MOV     R0,#CSAD    ;adres przetwornika A/D

        CLR     A    ;wyczyszczenie sterownika przerw
        MOVX    @R1,A
        INC     A
        MOVX    @R1,A

        SETB    EX1    ;zezwozenie na przerwanie INT1
        SETB    EA

        MOVX    @R0,A    ;inicjalizacja pierwszego pomiaru A/D

STOP:
        SJMP    STOP
```



.END

B.5 Program rs232.asm

```
.TITLE 'DSM RS232'
;=====
.CHIP 8051

STACK EQU 60H

.AJMP START ;poczatek programu
;-----
ORG 23H ;przerwanie z RS232
JBC RI,RECEIVE ;rozpoznanie przerwania
SEND: ;bajt wyslany
CLR TI
ACALL TRANS ;nadanie nastepnego
RETI

RECEIVE: ;bajt odebrany
MOV A,SBUF
CJNE A,'#?',REC_END ;odpowiedz tylko na ?
MOV R2,#0 ;nr znaku do nadania = 0
ACALL TRANS ;nadanie pierwszego znaku
REC_END:
RETI
;-----
START:
MOV SP,#STACK ;wskaznik stosu
MOV DPTR,#TEXT ;wskaznik nadawanego tekstu

SETR: ;ustawienie portu szeregowego
;mod 1 - 1start,8bitow,1stop,4800 bodow
;f=10 MHz - T1 dzieli przez 11, SMOD=1
MOV SCON,#01010000B ;mod 1 portu szeregowego
ORL PCON,#80H ;SMOD=1
MOV TMOD,#00100000B ;mod 2 Timera 1 - 8bit auto-reload
MOV TL1,#256-11 ;wlasciwy podzielnik
MOV TH1,#256-11
SETB TR1 ;start Timer 1
SETB ES ;zezwozenie na przerwania od RS232
SETB EA ;zezwozenie na przerwania

STOP:
SJMP STOP ;zatrzymanie glownego programu
;-----
TRANS: ;nadanie kolejnego bajtu
MOV A,R2
INC R2 ;zwiekszenie numeru znaku
MOVC A,@A+DPTR ;pobranie znaku
JZ TRANS_END ;koniec tekstu
MOV SBUF,A ;nadanie znaku
TRANS_END:
```



RET

```
;-----  
TEXT:  
    DB      'MicroMade',13,10,0      ;nadawany tekst  
  
.END
```

B.6 Program rs232.c

```
/* DSM-51      Program RS232  
   Odbior danych z kanalu szeregowego  
   *****/  
  
#include <conio.h>  
#include <dos.h>  
  
void main(void)  
{  
    int comadr,lsr,c;  
  
    comadr=0x3F8;          /* adres COM1 */  
    lsr=comadr+5;  
  
    outp(comadr+3,0x80);    /* programowanie predkosci */  
    outp(comadr,0x18);      /* mlodszy bajt dzielnika - 4800 */  
    outp(comadr+1,0x00);    /* starszy bajt dzielnika */  
    outp(comadr+3,0x07);    /* 8 bitow danych,2 bity stopu,No parity */  
    outp(comadr+1,0x00);    /* blokada przerwania */  
    inp(comadr);            /* wyczyszczenie bufora wejscowego */  
  
    while(!kbhit()){  
        while((inp(lsr)&0x20)==0);    /* bufor wyjsciowy pusty */  
        outp(comadr,'?');            /* pytanie o linie tekstu */  
        do {  
            if((inp(lsr)&0x01)==1){    /* jest cos w buforze */  
                c=inp(comadr);  
                putch(c);  
                if(c=='\n')break;    /* koniec odbieranej linii */  
            }  
        }  
        while(!kbhit());            /* wyjscie na dowolny klawisz */  
    }  
    getch();  
}
```




B.7 Program demo.asm

```
.TITLE 'DSM DEMO'
;=====

.CHIP 8051

T1 .REG P3.5

SPEAKER .REG P1.5
DS_ON .REG P1.6

;Adresy Urzadzen I/O
IO_HIGH EQU OFFH ;1111XXXXB

LCD EQU 080H ;10XXXXXXB

CSKB EQU 020H ;00100XXXB
CSDS EQU 037H ;00110XXXB
CSDB EQU 038H ;00111XXXB

LCDWC EQU 080H ;10XXXX00B
LCDWD EQU 081H ;10XXXX01B
LCDRC EQU 082H ;10XXXX10B
LCDRD EQU 083H ;10XXXX11B

CSKB0 EQU 021H ;00100X01B
CSKB1 EQU 022H ;00100X10B

STACK EQU 60H

D7S_DAT EQU 30H
KEYB EQU D7S_DAT+6
KEY0 EQU KEYB+1
KEY1 EQU KEY0+1

BANK0 MACRO
    CLR RSO
    CLR RS1
MACEND

BANK3 MACRO
    SETB RS1
    SETB RSO
MACEND

; POCZATEK PROGRAMU

AJMP START

;-----
```



```
ORG      OBH
; przerwanie od timera 0
CPL      SPEAKER
RETI

;-----
START:
MOV      SP,#STACK          ;wskaznik stosu
MOV      P2,#IO_HIGH

BANK3
MOV      R0,#LCDWD          ;dla wyswietlacza LCD
MOV      R1,#CSKB0          ;dla klawiszy
BANK0

ACALL    LCD_INIT
ACALL    LCD_CLR

;ustawienie Timera 0 jako generatora dla glosnika
MOV      TMOD,#00000010B    ;mod 2 (auto-reload)Timera 0
SETB     ET0                ;zezwolenie na przerwania Timer 0
SETB     EA

;wpisanie wstepnych wartosci do pamieci wyswietlacza
MOV      R0,#CSDS
MOV      R1,#D7S_DAT
MOV      A,#1
MOV      R2,#6
WPIS:
MOV      @R1,A
INC      R1
DJNZ     R2,WPIS
MOV      KEYB,#0

;-----
;przegladanie wyswietlaczy
MOV      R4,#200
MOV      R2,#1
NEXT:
MOV      A,R3
RR       A
MOV      R3,A
DEC      R1
DJNZ     R2,NEXT_SEG
MOV      R2,#7              ;wyswietlacz 6
MOV      R3,#40H
MOV      R1,#D7S_DAT+6
NEXT_SEG:
SETB     DS_ON              ;zapalenie kolejnego wyswietlacza
MOV      A,R2
DEC      A
MOVX     @R0,A
MOV      A,@R1
INC      R0
MOVX     @R0,A
```



```
DEC    R0
CLR    DS_ON

KEY:
MOV    A,R3                ;sprawdzenie klawiatury
JB     T1,KEY_NO
ANL    A,KEYB
JNZ    KEY_END
MOV    A,R3
ORL    KEYB,A
MOV    B,#30
MOV    A,R2
DEC    A
MUL    AB
INC    A
MOV    TH0,A                ;ustawienie wlasciwego dzwieku
SETB   TRO
SJMP   KEY_END

KEY_NO:
XRL    A,#OFFH
ANL    KEYB,A
MOV    A,KEYB
JNZ    KEY_END
CLR    TRO

KEY_END:
ACALL  LCD_KEY                ;sprawdzenie dodatkowej klawiatury
ACALL  TIME

CJNE   R2,#1,NEXT
DJNZ   R4,NEXT

MOV    R1,#DYS_DAT            ;nowe wartosci w pamieci wyswietlacza
MOV    R2,#6
MOV    A,@R1
RL     A

MOVE:
MOV    @R1,A
INC    R1
DJNZ   R2,MOVE

MOV    R4,#200
MOV    R2,#1
SJMP   NEXT

;-----
LCD_KEY:
PUSH   PSW                    ;sprawdzenie dodatkowej klawiatury
BANK3
MOVX   A,@R1                ;klawisze 0..7
CPL    A
JNZ    TKO
MOV    KEY0,#0
INC    R1
```



```
MOVX    A,@R1           ;klawisze 8..15
DEC      R1
CPL      A
JNZ      TK3
MOV      KEY1,#0

K_END:
POP      PSW
RET

TK0:
MOV      R3,A
ANL      A,KEY0
JNZ      K_END           ;klawisz juz byl wcisniety
MOV      A,R3
ORL      KEY0,A

MOV      R3,#7

TK1:
RLC      A
JC       TK2
DJNZ     R3,TK1

TK2:
MOV      A,R3
SJMP     TK6

TK3:
MOV      R3,A
ANL      A,KEY1
JNZ      K_END           ;klawisz juz byl wcisniety
MOV      A,R3
ORL      KEY1,A

MOV      R3,#7

TK4:
RLC      A
JC       TK5
DJNZ     R3,TK4

TK5:
MOV      A,R3
ADD      A,#8

TK6:
CJNE     A,#15,TK7        ;czy klawisz ENTER
ACALL    LCD_CLR
SJMP     K_END

TK7:
MOV      DPTR,#CODES
MOVC     A,@A+DPTR
ACALL    LCD_WRD          ;wypisanie kodu klawisza na LCD
SJMP     K_END

CODES:
DB        30H,31H,32H,33H
DB        34H,35H,36H,37H
DB        38H,39H,3CH,3EH
```



```
DB      5EH,76H,7FH,0AEH

;-----
;zerowanie LCD
LCD_INIT:
    PUSH    PSW
    BANK3
    ACALL   TIME2
    MOV     A,#38H
    CALL    LCD_WRC_NB
    ACALL   TIME2
    CALL    LCD_WRC_NB
    ACALL   TIME
    CALL    LCD_WRC_NB
    CALL    LCD_WRC
    MOV     A,#08H
    CALL    LCD_WRC
    MOV     A,#06H
    CALL    LCD_WRC
    MOV     A,#01H
    CALL    LCD_WRC
    POP     PSW
    RET

;czyszczenie LCD
LCD_CLR:
    PUSH    PSW
    BANK3
    MOV     A,#01H
    CALL    LCD_WRC
    MOV     A,#0FH
    CALL    LCD_WRC

    POP     PSW
    RET

;-----
;wpis instrukcji do LCD
LCD_WRC:
    ACALL   WAIT_BUSY
LCD_WRC_NB:
    DEC     RO
    MOVX    @RO,A
    INC     RO
    RET

;wpis danej do LCD
LCD_WRD:
    ACALL   WAIT_BUSY
    MOVX    @RO,A
    RET

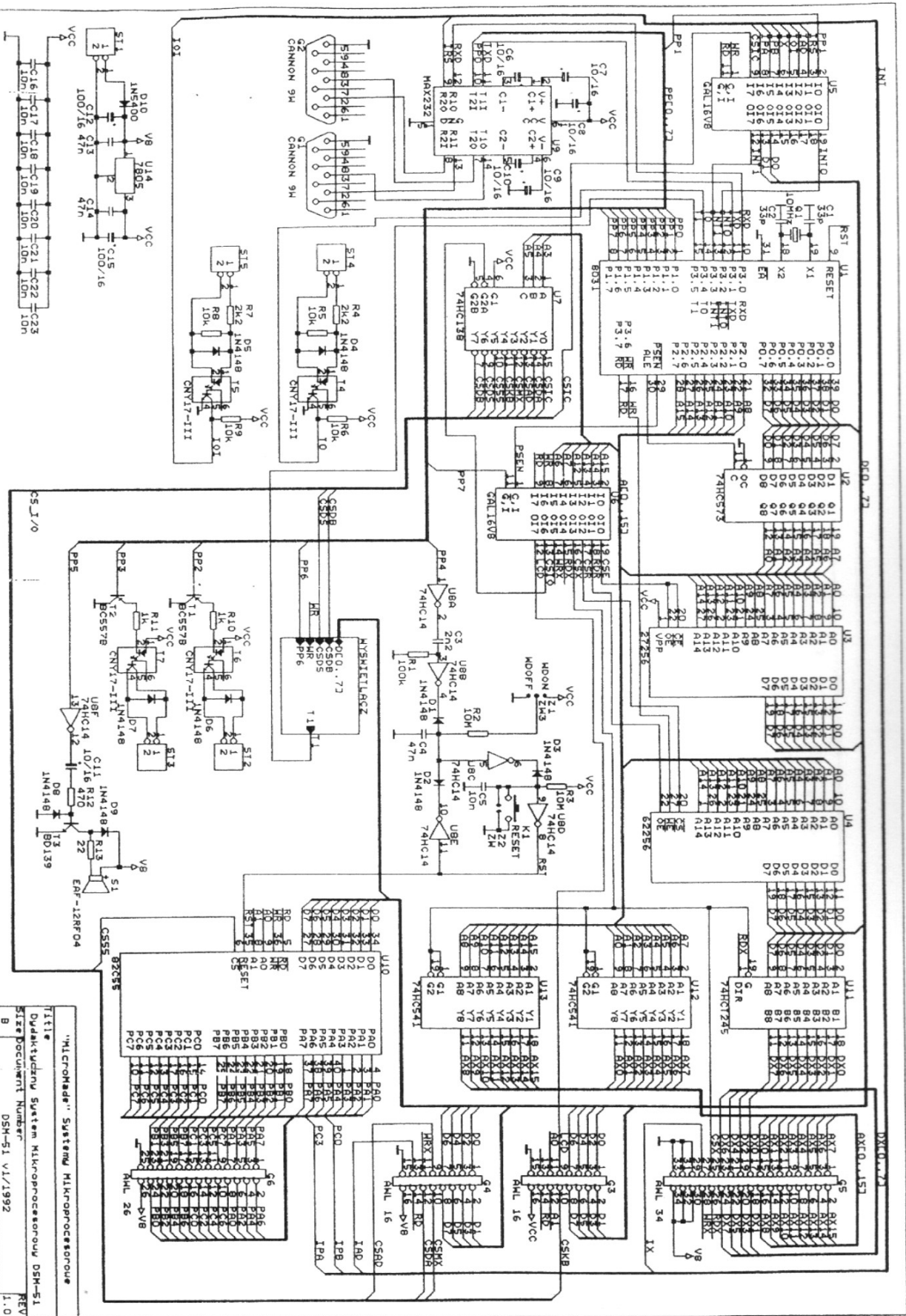
;czekanie na BUSY FLAG = 0
```



```
WAIT_BUSY:
    PUSH    A
    INC     R0
WAIT_BUSY1:
    MOVX    A, @R0
    JB      ACC.7, WAIT_BUSY1
    DEC     R0
    POP     A
    RET

;-----
TIME:
    MOV     R7, #0
TPT:
    DJNZ    R7, TPT
    RET
TIME2:
    MOV     R5, #40
T2PT:
    CALL    TIME
    DJNZ    R5, T2PT
    RET

.END
```



"MicroMaster" System Mikroprozessor

Titel

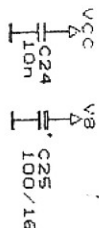
Dokumentation System Mikroprozessor DSM-51

Size Dokument Number

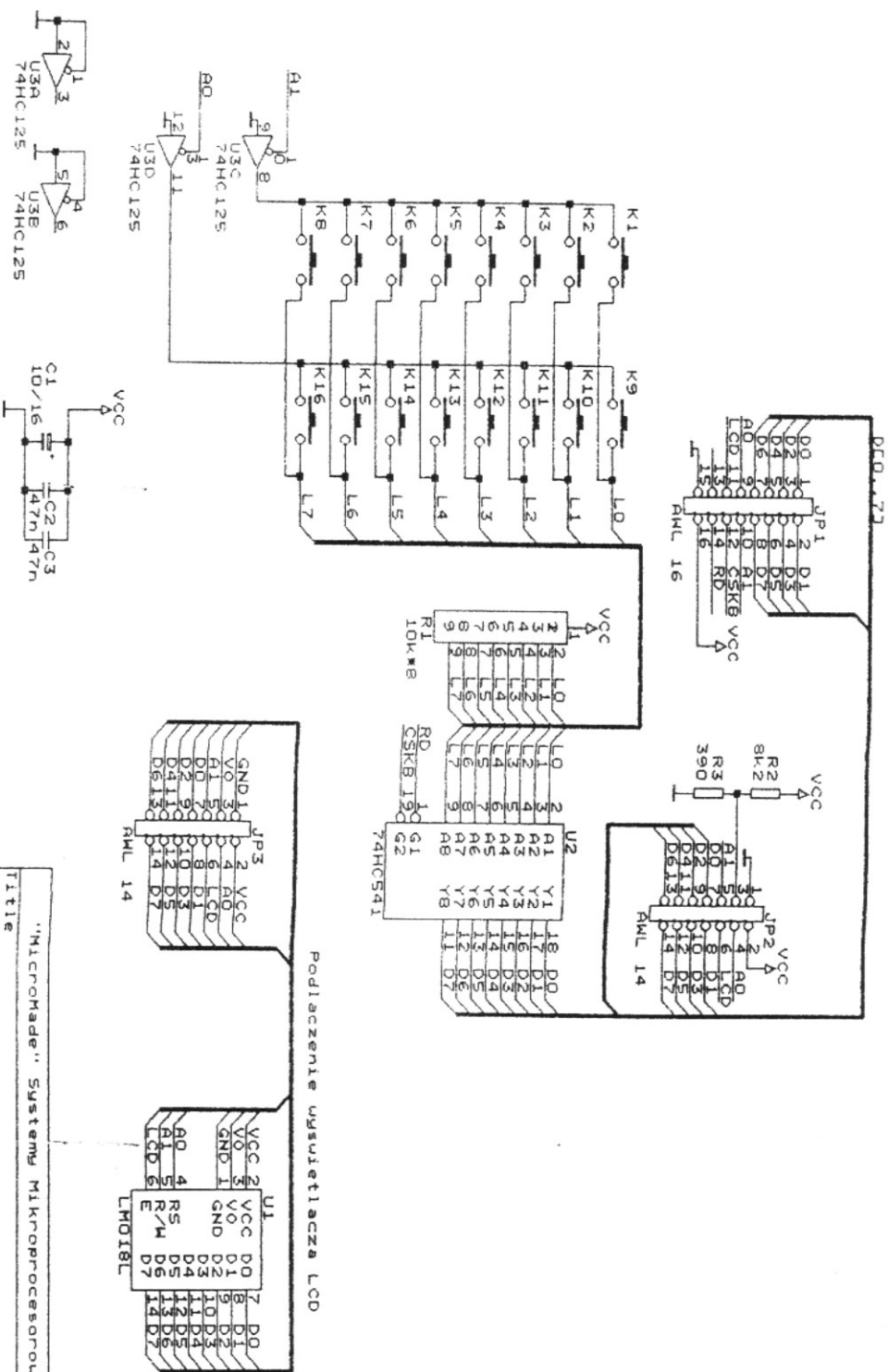
DSM-51 V1/1992

Date: December 15, 1992 Sheet 1 of 2

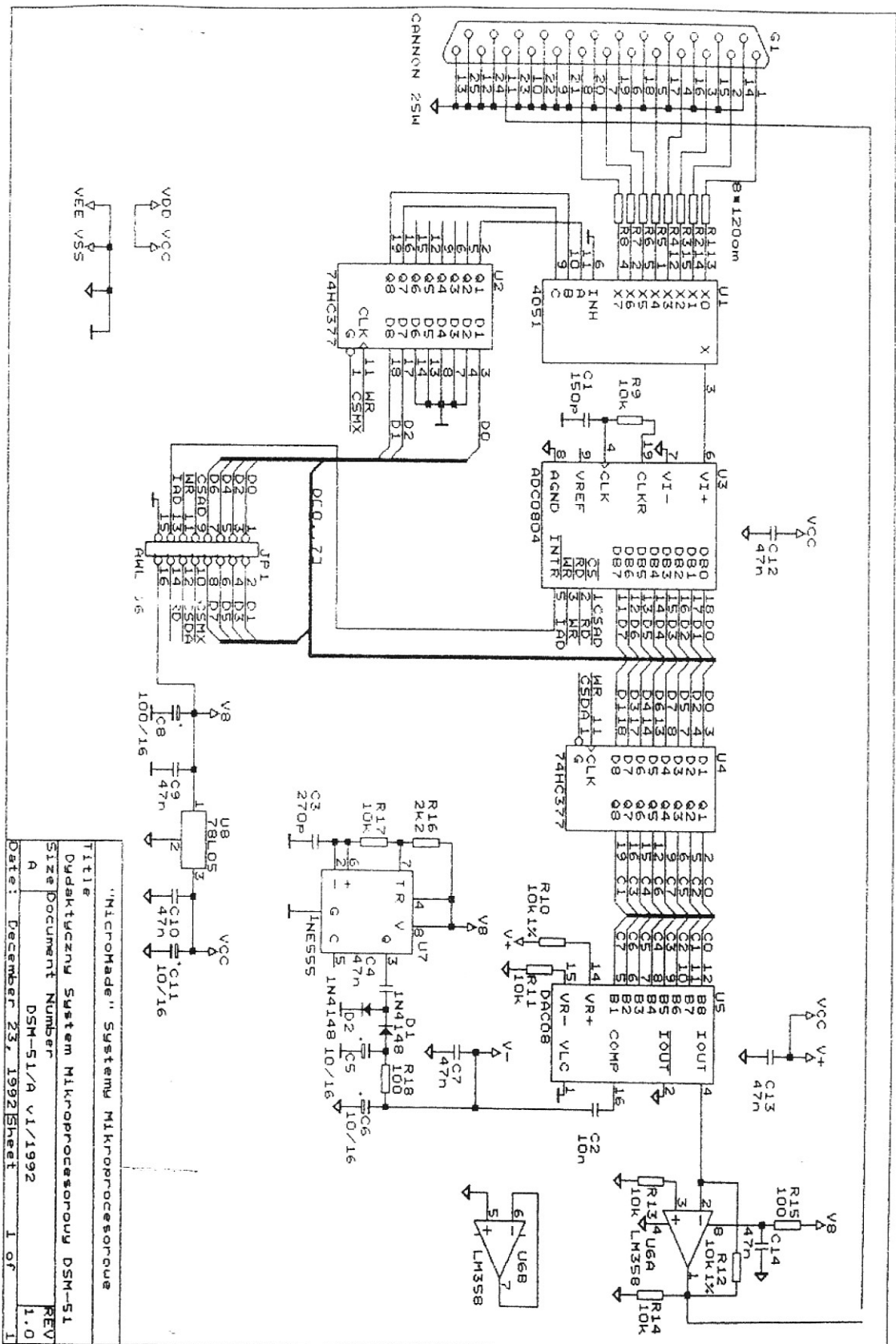
REV 1.0

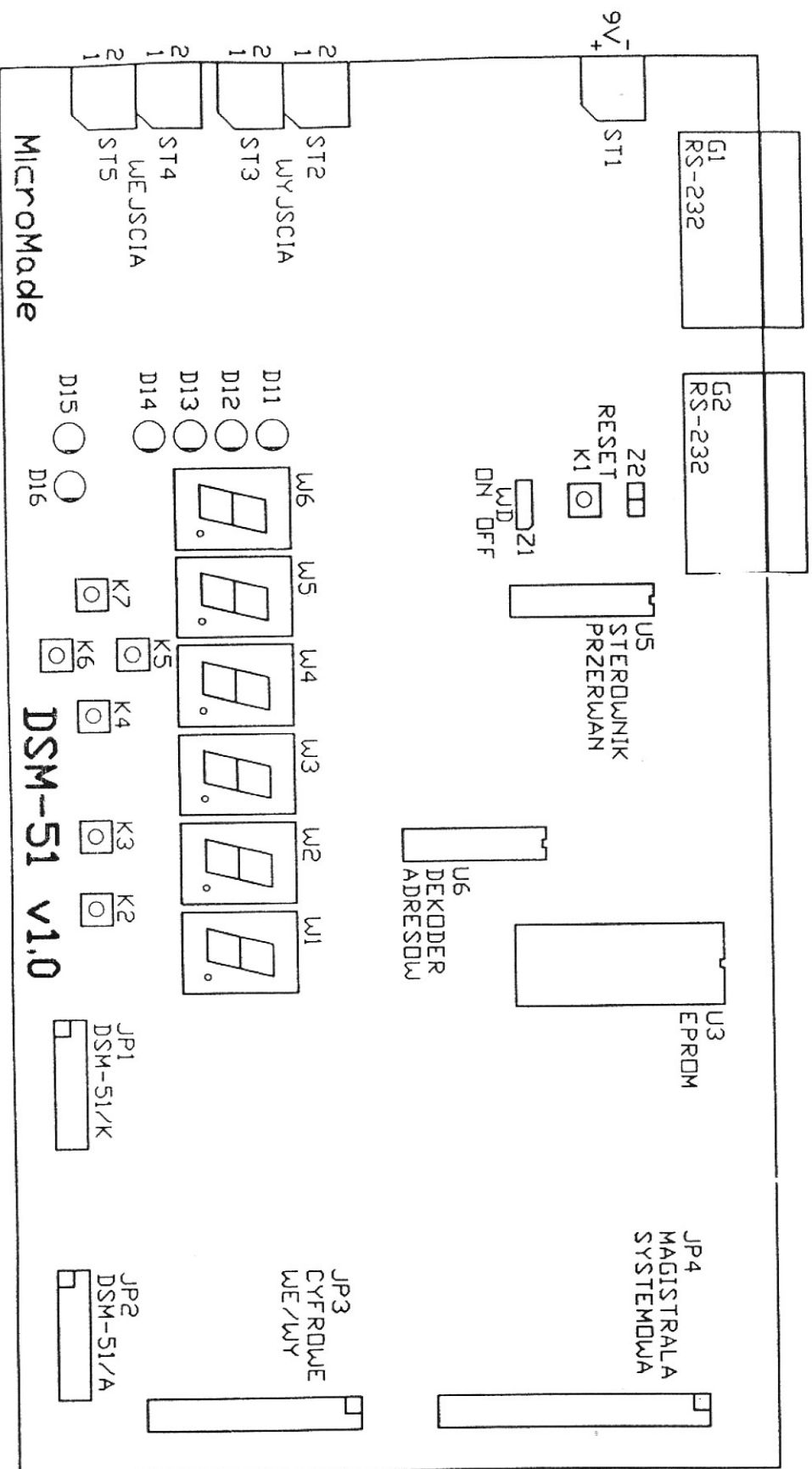


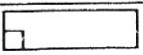
"MicroMade" Systemy Mikroprocesorowe	
Title	Dodatkowy System Mikroprocesorowy DSM-51
Size	Document Number
A	DSM-51 v1/1992
Date:	December 23, 1992 Sheet 2 of 2



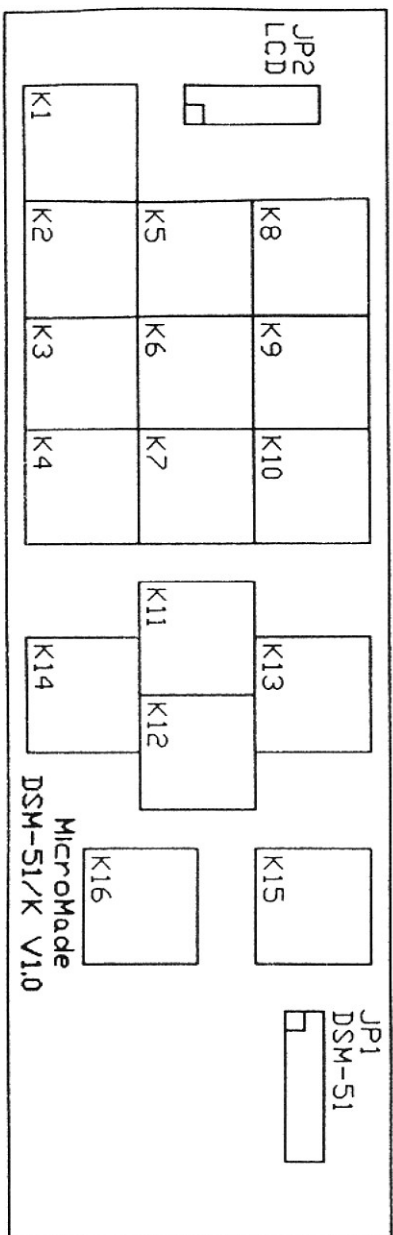
Title	
"MicroMaze" Systemy Mikroprocesorowe	
Dydaktyczny System Mikroprocesorowy DSM-S1	
Size Document Number	
A	
Date: December 15, 1992 Sheet 1 of 1	
REV	
1.0	







U¹
LCD LM018L



DSM-51/A V1.0

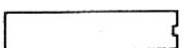


JP1
DSM-51

U5
DAC08



U3
ADC0804



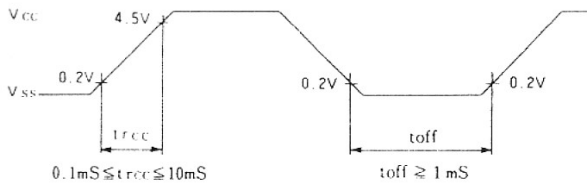
G1
ANALOGUE
WE/WY

1

MicroMode

The internal reset circuit will be operated properly when the following power supply conditions are satisfied.
If it is not operated properly, please perform initial setting along with the instruction.

Item	Symbol	Measuring Condition	Standard Value			Unit
			min.	typ.	max.	
Power Supply Rise Time	trcc	—	0.1	—	10	mS
Power Supply OFF Time	toff	—	1	—	—	mS



Note: toff defines period that power supply is off when power supply shut down momentarily or repeats on/off state.

RESET FUNCTION

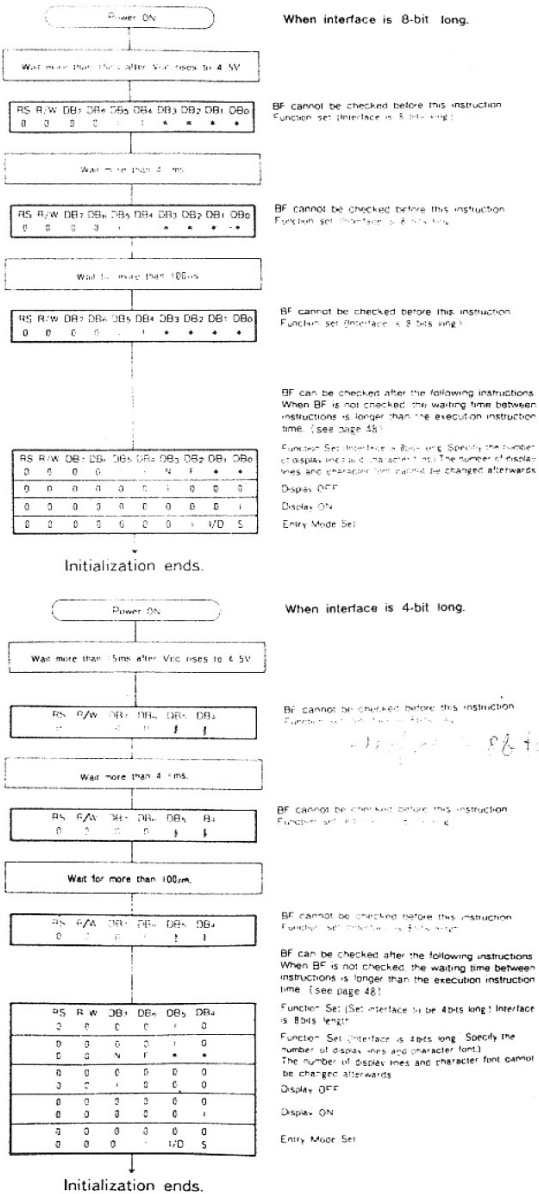
● **Initialization made by Internal Reset Circuit**
HD44780 automatically initializes (resets) when power is supplied (built-in internal reset circuit). The following instructions are executed in initialization. The busy flag (BF) is kept in busy state until initialization ends. (BF=1) The busy state is 10ms after Vcc reach to 4.5V.

- (1) Display clear
DL=1: 8bit long interface data
DL=0: 4bit F=0: 5×7dot character font
N=1: 2lines
N=0: 1line
- (3) Display ON/OFF control
D=0: Display OFF C=0: Cursor OFF B=0: Blink OFF
- (4) Entry mode set
I/D=1: +1(increment) S=0: No shift

Note: When conditions stated in "Power Supply Conditions Using Internal Reset Circuit" are not satisfied, the internal reset circuit will not operate properly and initialization will not be performed. Please make initialization using MPU along with "Initialization along with Instruction".

● Initialization along with Instruction

If power supply conditions are not satisfied, which for proper operation of internal reset circuit, it is required to make initialization along with instruction. Please make following procedures: —



INSTRUCTIONS (Except for DMC40401 series)

INSTRUCTIONS (Except for DMC4041 series)													Executed Time(max.)
Instruction	Code										Description		
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears all display and returns the cursor to the home position(Address 0).	1.64mS	
Cursor At Home	0	0	0	0	0	0	0	0	0	1 *	Returns the cursor to the home position(Address 0). Also returns the display being shifted to the original position. ODRAM contents remain unchanged.	1.64mS	
Entry Mode Set	0	0	0	0	0	0	0	0	1	I/D S	Sets the cursor move direction and specifies or not to shift the display. These operations are performed during data write and read.	40μS	
Display On/Off Control	0	0	0	0	0	0	0	1	D	C B	Sets ON/OFF of all display(D) cursor ON/OFF (C), and blink of cursor position character(B).	40μS	
Cursor/Display Shift	0	0	0	0	0	0	1	S/C	R/L	* *	Moves the cursor and shifts the display without changing DDRAM contents.	40μS	
Function Set	0	0	0	0	0	1	DL	N	F	* *	Sets interface data length(DL) number of display lines(L) and character font(F).	40μS	
CGRAM Address Set	0	0	0	0	1	A _{CG}					Sets the CGRAM address. CGRAM data is sent and received after this setting.	40μS	
DDRAM Address Set	0	0	0	1	A _{DD}					Sets the DDRAM address. DDRAM data is sent and received after this setting	40μS		
Busy Flag/ Address Read	0	1	BF	AC					Reads Busy flag(BF) indicating internal operation is being performed and reads address counter contents.			0μS	
CGRAM/DDRAM Data Write	1	0	WRITE DATA					Writes data into DDRAM or CGRAM.				40μS	
CGRAM/DDRAM Data Read	1	1	READ DATA					Reads data from DDRAM or CGRAM.				40μS	

Code										Description	Execution Time(max.)
I/D = 1: Increment I/D = 0: Decrement S = 1: With display shift S/C = 1: Display shift S/C = 0: Cursor movement R/L = 1: Shift to the right R/L = 0: Shift to the left DL = 1: 8-bit DL = 0: 4-bit N = 1: 2lines N = 0: 1line F = 1: 5×10dots F = 0: 5×7dots BF = 1: Internal operation is being performed BF = 0: Instruction acceptable										DDRAM: Display Data RAM CGRAM: Character Generator RAM ACG : CGRAM Address ADD : DDRAM Address Corresponds to cursor address. AC : Address Counter, used for both DDRAM and CGRAM IN : Invalid	fcp or fosc = 250kHz However, when frequency changes, execution time also changes Ex If fcp or fosc is 270kHz. $40\mu S \times \frac{250}{270} = 37\mu S$

FONT TABLE (5×11Dots)

Lower 4bit	Upper 4bit	CG RAM
0000	0000	0000
0001	0001	0001
0010	0010	0010
0011	0011	0011
0100	0100	0100
0101	0101	0101
0110	0110	0110
0111	0111	0111
1000	1000	1000
1001	1001	1001
1010	1010	1010
1011	1011	1011
1100	1100	1100
1101	1101	1101
1110	1110	1110
1111	1111	1111

(5×8Dots)

Lower 4bit	Upper 4bit
0000	0000
0001	0001
0010	0010
0011	0011
0100	0100
0101	0101
0110	0110
0111	0111
1000	1000
1001	1001
1010	1010
1011	1011
1100	1100
1101	1101
1110	1110
1111	1111

*CGRAM is Character Generator RAM which memorize characters that you can freely input by programme.
 *32 characters stated under upper 4-bit of 1110 and 1111 are 5×10 dots, and part of which is cut when you use in display which display fonts is 5×7 dots.
 Please note.

5×10 dots applied product:
 DMC16106A, DMC16101A-EB, DMC24138, DMC32132, DMC40131