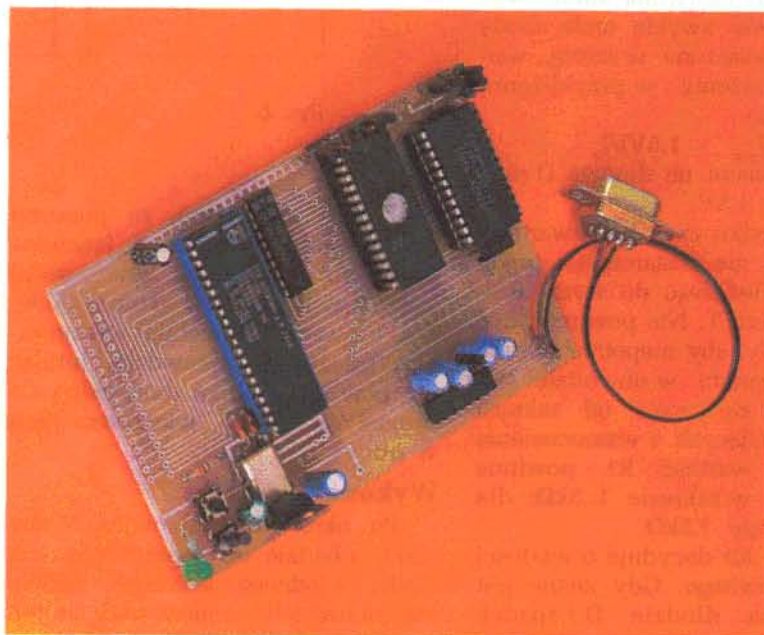


# Pakiet prototypowy mikrokomputera 8031/51

## kit AVT-107

*Przedstawiony w artykule układ powstał w wyniku wielu doświadczeń autora z systemami mikroprocesorowymi opartymi na mikrokomputerach jednoukładowych. Dość często występowała potrzeba sprawdzenia lub uruchomienia niezbyt skomplikowanego układu, a sprawę utrudniała konieczność budowania systemu od „zera”. Aby ułatwić i uprościć realizację tego typu koncepcji stworzono właśnie ten system.*



Pierwsze kroki oparte były na układach serii 8048/9, lecz wkrótce okazało się iż seria 8051/52 jest bardziej dopracowana i daje trochę większe możliwości, zarówno od strony hardware'u wbudowanego w mikroprocesor, jak i od strony oprogramowania.

Jak wiadomo, każdy system mikrokomputerowy (określenie może nieco na wyrost w stosunku do omawianego układu) składa się z jednostki centralnej - mikroprocesora, pamięci programu - na ogół stałej ROM, pamięci danych - RAM i portów wejścia wyjścia (I/O). Autor pokusił się o zaprojektowanie uniwersalnego systemu zawierającego te właśnie elementy, przy czym wszystkie istotne sygnały systemowe, tzn. adresy, dane i sygnały sterujące są wyprowadzone na listwę kontaktową, co znakomicie upraszcza projektowanie dowolnych systemów użytkowych.

Jako dodatkowe wyposażenie systemu przewidziano płytkę uniwersalną w dwóch wersjach - pierwsza z nich do typowych układów cyfrowych z wyjściami małej mocy, druga przystosowana do montażu

podzespołów większej mocy, przełączających nieco większe prądy. Płytki uniwersalne mają tak zaprojektowany druk aby jak najbardziej ułatwić mechaniczną stronę montażu - otwory mocujące i szyny sygnałów sterujących pokrywają się.

Jeszcze jedno słowo wyjaśnienia - dlaczego akurat 8051?

Na rynku światowym istnieje co najmniej kilka znaczących rodzin mikrokomputerów jednoukładowych. Chociażby R6500 Rockwell'a, MC6801X Motoroli (obydwa są odpowiednikami serii 6800), świetnej klasy rodzina Ziloga Z86XX czy też Z8X, Philipsa - 80XXX - niby oparta na Intelu 8051, ale wykazująca znaczną przewagę w sprawie sprzętowej mikrokomputera oraz w łatwości rozbudowy. W praktyce okazało się, iż 8051 jest nie do pobicia zarówno pod względem ceny jak i dostępności. Możliwości tego układu dla większości amatorów (nie całkiem amatorów, skoro podejmują się tak poważnego wyzwania) są całkowicie wystarczające, a niektóre wydumane funkcje innych układów pozostają zwykle nie wykorzystane. Tak więc motywy,

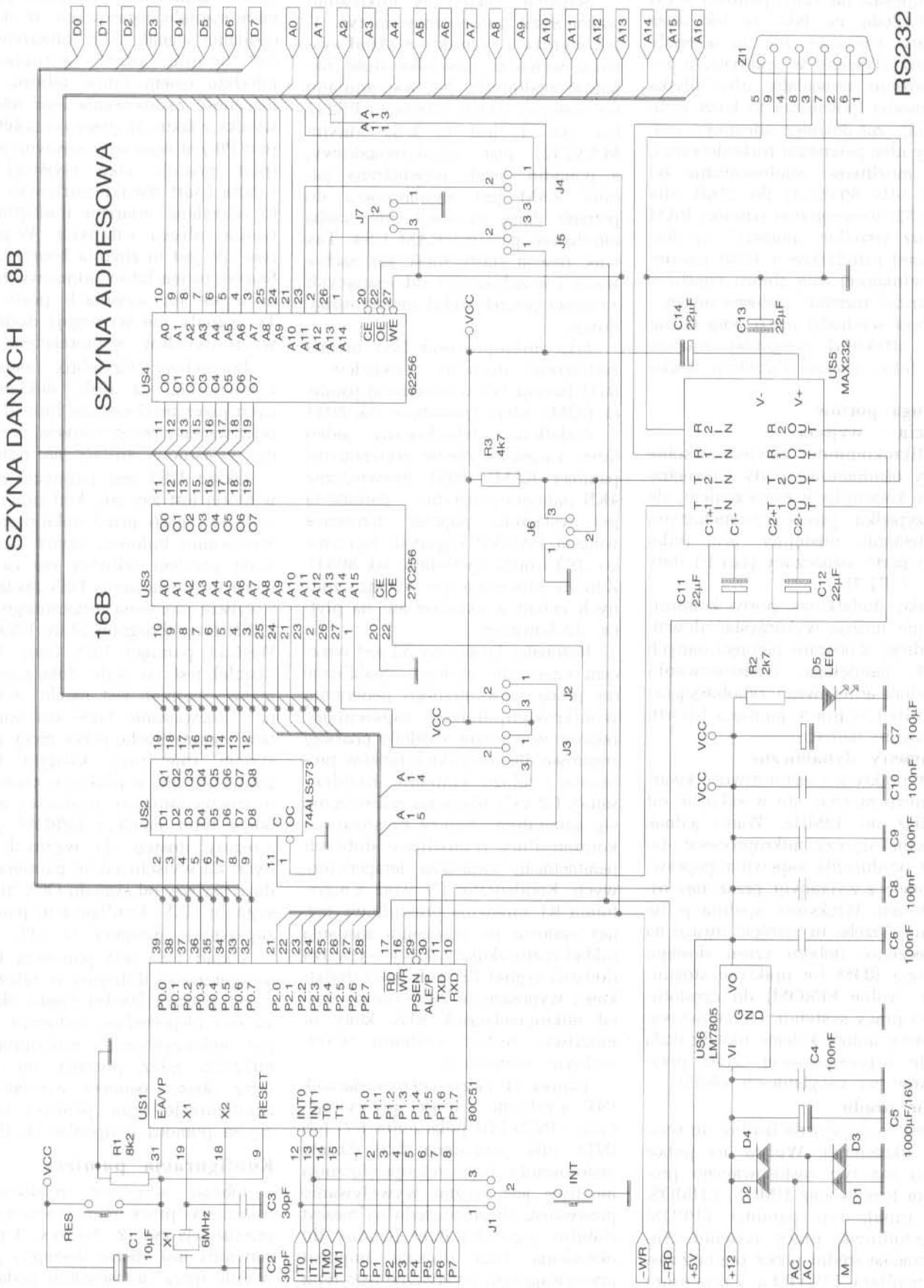
którymi kierował się autor są chyba jasne.

Jako dodatkową i całkowicie nieobowiązkową opcję rozszerzającą możliwości opisywanego mikrokomputera zastosowano interfejs szeregowy standardu RS232. Umożliwia on łatwą współpracę systemu z dowolnym komputerem wyposażonym w port szeregowy. W jednym z kolejnych numerów EP przedstawimy opis centralki alarmowej współpracującej z komputerem klasy PC.

### Możliwości systemu

#### Operacje na pamięci:

- możliwość zaadresowania do 64KB pamięci programu;
- możliwość zaadresowania do 64KB pamięci danych RAM (niezależna od ROM przestrzeń adresowa). Pewną wadą układu 8051 jest ograniczenie operacji na tej pamięci do jednego tylko rozkazu MOVX (możliwe są oczywiście różne argumenty), a dodatkową wadą (lub nie, w zależności od potrzeby chwili) jest konieczność adresowania tej pamięci poprzez wskaźnik danych DPTR lub jeden z rejestrów R. Taki sposób adresowania nazywa się po-



Rys. 1. Schemat elektryczny mikrokomputera

rednim. W modelu przewidziane zostało miejsce na 32KB pamięci RAM ze względu na fakt, iż tak duże obszary adresowe nie są z reguły wykorzystywane w praktyce, a powiększeniu musiałaby ulec płytka drukowana (powiększa to koszt urządzenia), zaś dekodery adresowe musiałby ulec pewnemu rozbudowaniu;

- możliwość zaadresowania od 128B (dla 8031/51) do 256B (dla 8032/52) wewnętrznej pamięci RAM (tu już wszelkie „numery“ są dozwolone) i dodatkowo 128B pamięci spełniającej rolę zbioru rejestrów (o bardzo różnych zastosowaniach - ich opis wychodzi nieco poza obszar tego artykułu) o ogólnej nazwie SFR (ang. Special Function Register).

#### Obsługa portów wejścia - wyjścia

Mikrokomputer zawiera w sobie cztery ośmiobitowe porty wewnętrzne do komunikacji z otoczeniem, ale w przypadku pracy z zewnętrznymi pamięciami dostępny jest tylko jeden port - oznaczony jako P1 (bity P1.0 - P1.7).

Jako dodatkowe porty komunikacyjne można wykorzystać dowolne adresy z obszaru pamięci danych RAM, pamiętając o zastosowaniu dekodera adresowego zapobiegającego kolizji w pracy pamięci i portu na szynie danych.

#### Parametry dynamiczne

W praktyce częstotliwość kwarcu może mieścić się w zakresie od 3.5MHz do 12MHz. Warto jednak upewnić się czy mikroprocesor danego producenta zapewnia poprawną pracę z wybranym przez nas oscylatorem. Większość spełnia je na pewno. Trzeba tu zwrócić uwagę na konieczność doboru czasu dostępu pamięci ROM (w praktyce stosunkowo wolne EPROM) do częstotliwości pracy systemu. Opracowywane przez autora kolejne układy będą oparte w przeważającej części przypadków na oscylatorach 6MHz.

#### Pobór prądu

Jest to naprawdę trudne do ścisłego określenia. Wpływ na pobór prądu ma typ zastosowanego procesora (czy wersja HMOS, CHMOS, czy inna), typ pamięci EPROM i częstotliwość pracy systemu. Autor szacuje średni pobór prądu z zastosowaniem EPROM'u i procesora w wersji CHMOS na ok. 40mA. Nie uwzględniono tu poboru mocy przez układy zewnętrzne.

#### Opis układu

Schemat elektryczny mikrokomputera został zamieszczony na **rys. 1**. Jak wynika z rysunku w skład systemu wchodzi niewielka ilość układów scalonych. Jak wspomniano we wstępie, układ interfejsu RS232 (na **rys. 1** jest to US5 Maxim MAX232) jest nieobowiązkowy, a ponadto, jeżeli wewnętrzna pamięć RAM jest wystarczająca do potrzeb danej aplikacji, nie trzeba montować pamięci RAM US4. Tak więc pewna elastyczność jest zachowana i w zależności od rosnących z czasem potrzeb układ można rozbudować.

Jako mikroprocesor US1 można zastosować dowolny z układów - 8031 (wersja bez wewnętrznej pamięci ROM), 8032 (podobnie jak 8031 - dodatkowo wbudowany jeden timer i większy obszar wewnętrznej pamięci RAM), 8051 (wewnętrzne 4KB pamięci programu - pamięć ta jest wyłączona poprzez sterowanie wejścia EA\VP sygnałem logicznego „0“) i 8052 (podobnie jak 8051). Zmiana procesora nie wymaga żadnych zmian w układzie ani na płycie drukowanej.

Rezonator kwarcowy X1 jest wzorcem czasu dla systemu. Stabilizuje on pracę wewnętrznego generatora w mikrokomputerze, zapewniając poprzez wewnętrzne dzielniki przebiegi zegarowe dla wszystkich bloków procesora i całego systemu. Kondensatory C2 i C3 ułatwiają wzbudzenie się generatora. Należy tu stosować kondensatory o możliwie dobrych parametrach, zwłaszcza temperaturowych. Kondensator C1 wraz z rezystorem R1 zapewnia prawidłowy restart systemu po włączeniu zasilania (układ różniczkujący, który wytwarza dodatni sygnał RESET). Jako dodatkowe wyposażenie zastosowany został mikroprzełącznik RES, który umożliwia restart systemu w dowolnym momencie.

Jumper J1 łączy mikroprzełącznik INT z jednym z wejść przerywających - INT0 (dla połączenia 1-2) lub INT1 (dla połączenia 2-3). Dzięki zastosowaniu tego mikroprzełącznika możliwe jest ręczne wywoływanie przerwania, ale ze względu na niezbyt stabilny jego charakter (trudny do określenia czas trwania impulsu przerywającego) przerwanie tego typu wymaga specjalnej obsługi. Proponowany program obsługi przedstawiono na **listingu 1**.

Układ US2 jest to rejestr typu „latch“ zapisywany wysokim poziomem sygnału zegarowego. W chwili opadania sygnału ALE zatraskiwany jest w nim adres, a dokładnie młodsze osiem bitów adresu. Konieczność zastosowania tego rejestru wynika z faktu, iż przez ośmiobitowy port P0 jest wysyłany zarówno adres (port pracuje jako wyjście) jak i dane (port dwukierunkowy). Jest to określane mianem multipleksowania adresu i danych. W praktyce nie jest to zbyt komplikacja. Starsze osiem bitów adresowych jest dostępne na wyjściach portu P2. Te sygnały nie wymagają dodatkowych układów wspierających.

Dociekliwy Czytelnik zauważy z pewnością, iż ilość deklarowanych przez producentów linii I/O nie pokrywa się z rzeczywistością, przyczyny wyjaśnione zostały we wstępie.

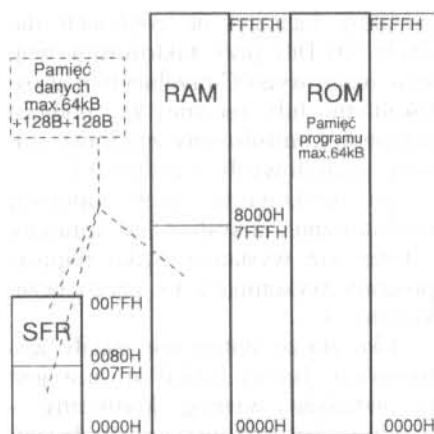
Układ US3 jest pamięcią stałą, w której jest zawarty kod programu wykonywanego przez mikroprocesor. Sterowanie buforem szyny danych w tej pamięci odbywa się za pomocą wyprowadzenia OE\ zasilanego z wyjścia procesora oznaczonego jako PSEN (ang. Program Store Enable). Wejście pamięci CE\ (ang. Chip Enable) jest na stałe dołączone do masy - pamięć jest ciągle „wybrana“. Rozwiązanie takie ma wadę - otóż wzrasta trochę pobór mocy przez system (nie przy każdym typie pamięci!), ale w praktyce zapewnia to trochę mniejsze trudności w doborze typu pamięci EPROM, gdyż z reguły dostęp do ważnych danych na wyjściu takiej pamięci jest dużo szybszy od sygnału OE\ niż od sygnału CE\. Konfigurację pamięci zapewniają jumpery J2 i J3.

Układ US4 jest pamięcią RAM o pojemności dobranej w zależności od potrzeb. Bardzo często okazuje się ona niepotrzebna, zwłaszcza jeżeli jest wykorzystywany mikroprocesor 8032/52, gdyż posiada on dość duży obszar pamięci wewnętrznej. Konfigurację typu pamięci ustala się za pomocą jumperów J4, J5, J7.

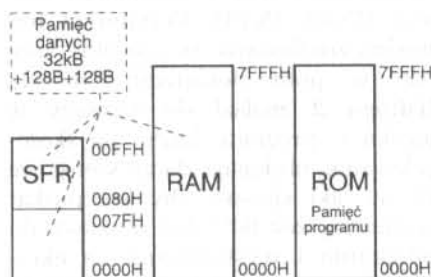
#### Konfiguracja pamięci

Obszar adresowy możliwy do osiągnięcia przez mikroprocesor 8051 przedstawia **rys. 2**. Na **rys. 3** przedstawiony jest obszar dostępny praktycznie (przy zastosowaniu podanych typów układów - bez dodatkowego dekodera adresowego).

Chcąc maksymalnie uprościć



Rys. 2. Obszar adresowy mikrokomputera 8051



Rys. 3. Obszar adresowy µP8051 dostępny praktycznie

system nie zastosowano żadnych dodatkowych dekodery adresowych, gdyż raczej mało prawdopodobne jest wykorzystanie wszystkich dostępnych możliwości systemu (choćby rozszerzenie pamięci danych), a dodatkowe elementy kosztują! Ewentualne rozbudowy są oczywiście możliwe, ale to już jest zależne od indywidualnych potrzeb konstruktorów.

Ponieważ w systemie przewidziano możliwość zastosowania praktycznie dowolnej pamięci RAM, np. 6116 (2048x8) lub 6164 (8192x2), ewentualnie 62256 (32768x8), wystąpiła konieczność konfigurowania sygnałów dla wybranego typu pamięci. Dokonuje się tego za pomocą jumperów J4 - selekcja A11 i WE\ dla n. 23, J5 - selekcja +Vcc i A13 dla n. 26 oraz J7 - selekcja WE\ i GND dla n. 20. W tabeli 1 przedstawione są położenia jumperów dla różnych pamięci.

Należy tu zwrócić uwagę na następującą sprawę: dla pamięci 62256 zastosowana została dość „rygorystyczna” metoda selekcji pamięci na czas zapisu. Procesor wybierając pamięć RAM sygnałem WR\ dokonuje jednoczesnego wyboru (selekcji - sygnałem CS\ ) układu, co dla zbyt powolnych układów może zakończyć się niepomyślnie. Należy sprawdzić czy pamięć, która ma być zastosowana w układzie nie ma specjalnych wymagań co do czasu wyprowadzenia sygnału CS\ w stosunku do WR\ lub odwrotnie. Dla pozostałych pamięci problem ten nie występuje - pamięć jest cały czas wybrana (CS\=0) i jest sterowana poziomami RD\ (dla odczytu) i WR\

Tab. 1. Ustawienie jumperów dla pamięci RAM

Oznaczenie	6116	6164	62256
J4	2-3	1-2	1-2
J5	1-2	2-3	2-3
J7	1-2	1-2	2-3

Tab. 2. Ustawienie jumperów dla pamięci EPROM

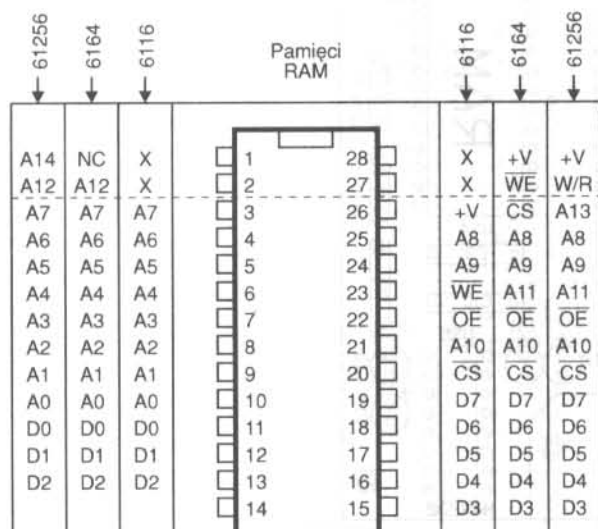
Oznaczenie	2764	27128	27256
J2	2-3	2-3	2-3
J3	1-2	2-3	2-3

(dla zapisu).

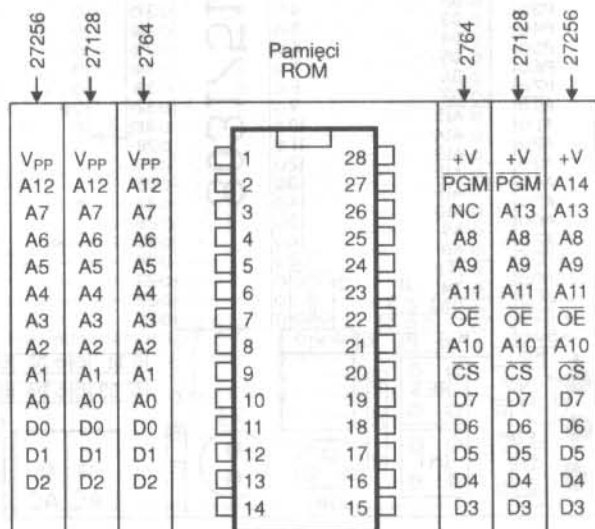
Jak wiadomo, mikroprocesor 8051 ma wydzielony osobny obszar adresowy dla pamięci programu i może zaadresować do 65535 komórek. W wielu zastosowaniach jest to zbyt dużo, a stosowanie dużych pamięci EPROM „na zapas” może się okazać niezbyt ekonomiczne. Z tego też powodu istnieje możliwość wyselekcjonowania rodzaju zastosowanej pamięci za pomocą jumperów J2 i J3. Przewidziano wybór jednej spośród następujących pamięci: 2764 (8192x8), 27128 (16384x8) i 27256 (32768x8). Jak widać, gama możliwych do zastosowania pamięci jest bardzo szeroka, a pominięcie układów 2716 i 2732 wiąże się z malejącą ich popularnością - dodatkowym atutem jest ciągły spadek cen pamięci o większych pojemnościach, które często osiągają ceny niższe niż pamięci o małych pojemnościach.

Położenia jumperów dla różnych typów pamięci podano w tabeli 2.

W celu ułatwienia zrozumienia



Rys. 4.



Rys. 5

operacji dokonywanych za pomocą jumperów na rys. 4 przedstawiono wyprowadzenia zastosowanych w systemie pamięci RAM, a na rys. 5 pamięci ROM.

**Montaż i uruchomienie**

Proponowany przez nas system jest stosunkowo łatwy do uruchomienia, nawet przez niezbyt wprawnego amatora. Praktycznie wystarczą podstawowe informacje z dziedziny elektroniki - nieco inaczej sprawa się ma z oprogramowaniem. W krajowej literaturze dotyczącej oprogramowania systemów mikrokomputerowych występuje spora ilość błędów, bardzo często uniemożliwiająca poprawne uruchomienie programu. Dlatego najlepiej jest korzystać z zachodnich publikacji, ewentualnie katalogów firmowych. Doskonałym „podręcznikiem” okazał się katalog Philipsa „Philips Semiconductors 80C51-Based 8-bit Microcontrollers”<sup>1)</sup>.

Do poprawnego „elektrycznie” uruchomienia systemu potrzebna będzie płytką drukowaną wykonana wg rysunków przedstawionych na wkładce. Rozmieszczenie elementów przedstawia rys. 6. Przy montażu elementów należy pamiętać o wspomnianych wcześniej uwagach dotyczących możliwości etapowego konstruowania systemu. Niezbędne jest zamontowanie podstawki pod mikroprocesor (40-nóżkowa), pod

„latch’a” US2 (20-nóżkowa) i pamięć EPROM US3 (28-nóżkowa). Warto także zamontować elementy zasilacza - układ US6 z radiatorem, diody D1-D4, kondensatory filtrujące C4, C5 i blokujące C6, C8-10. Dioda LED D5 sygnalizuje załączenie napięcia zasilającego. Kvarc, jeżeli jest montowany w metalowej obudowie, należy przylutować do ścieżki masy (niezbyt długo grzać!) poprowadzonej pod obrysem obudowy. Niezbędne są także C1, R1 - gwarantują poprawną inicjację procesora. Jeżeli decydujemy się na jeden, ściśle określony typ pamięci można pominąć montaż jumperów J2, J3 oraz J4, J5, J7. Konieczne jest wtedy wykonanie połączeń srebrzanką lub przewodem zgodnie z tabelami.

Wstępne uruchomienie systemu można wykonać za pomocą zwykłego miernika - detektora zwarc lub omomierza. Trzeba po prostu sprawdzić czy ścieżki nie są zwarte do masy lub plusa zasilania, ewentualnie czy nie występują zwarcia pomiędzy nimi. Trzeba także skontrolować wartość napięcia zasilającego układy scalone. UWAGA! Ten etap uruchomienia wykonujemy bez wmontowanych układów scalonych w podstawki!

Drugi etap to sprawdzenie poprawności działania przerytnika „latch” - do jego wyjść dołączamy np. LEDy lub wejście próbnika stanów logicznych TTL i zmieniając

poziomy logiczne na wejściach danych (D1-D8) przy taktowaniu wejścia zegarowego C przebiegiem prostokątnym lub ręcznie za pomocą jumpera, kontrolujemy zgodność stanów wyjściowych z zadanymi.

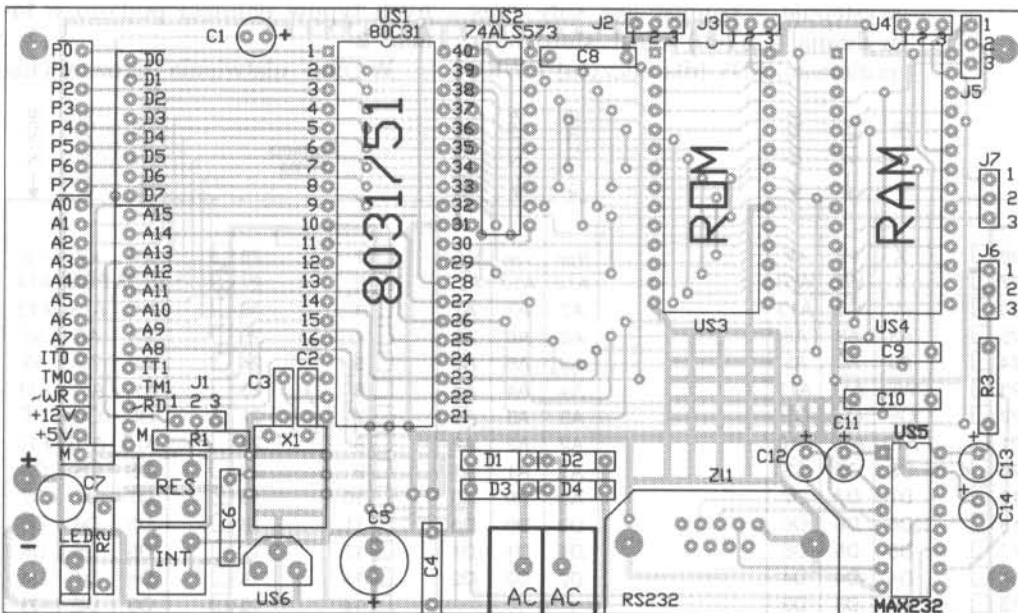
Po wykonaniu tych operacji można uznać iż system jest sprawny - teraz już wystarczy tylko napisać program wykonujący to, czego oczekujemy i ...

Jako złącze systemowe zostały zastosowane listwy GoldPin, nie jest to jednakże wymóg konieczny - możliwe jest dokonywanie połączeń zwykłą srebrzanką lub przewodem w izolacji.

**Uwagi końcowe**

Na listingu 1 przedstawiono przykładowy program obsługi złącza szeregowego mikrokomputera. Mikrokomputer wysyła komunikat do komputera, np. IBM PC, po zgłoszeniu przerwania z dowolnego wejścia (INT0, INT1). Komunikat ten można zdefiniować w dowolny sposób w polu deklaracji DB. Na listingu 2 znalazł się napisany w języku C program, który, po skompilowaniu, obsługuje złącze szeregowo PC w taki sposób, aby komunikat wysłany przez 8051 był możliwy do odebrania i wyświetlenia na ekranie.

Wspomniane zostało już wcześniej iż jako rozszerzenie systemu oferowane będą dwa rodzaje płytek druko-



Rys. 6. Rozmieszczenie elementów na płytce drukowanej

Listing 1.

```

;-----
; Wysłanie po każdym przerwaniu zewnętrznym przez MAX komunikatu
; tekstowego niezależnie od numeru wejścia (INT0, INT1) z
; którego przychodzi sygnał przerwania.
; 1993 - 06 - 06
;-----
ORG 0
AJMP POCZ
; skok do procedury inicjacyjnej
ORG 0003H
AJMP EXIN
; skok do procedury obsługi przerwania zewnętrznego
ORG 0013H
AJMP EXIN
; skok do procedury obsługi przerwania zewnętrznego
;-----
; Inicjacja
;-----
ORG 100H
POCZ:
MOV SCON,#52H
; ustawienie trybu pracy portu szeregowego
; SMO=0, SM1=1 -> tryb 1 = transmisja 8-bitowa
MOV TMOD,#20H
; ustawienie układu czasowo-licznikowego T1 jako timera w trybie 2
; tryb 2 = 8-bitowy licznik (licznikiem jest młodszy bajt TL1)
; z automatycznym wpisywaniem wartości początkowej
; zapisanej w TH1
MOV TH1,#0F3H
MOV TL1,#0F3H
; wpisanie wartości początkowej do TH1
; wartość ta ustawia szybkość transmisji na 1200 baudów (dla kwarcu 6MHz)
SETB TR1
; start timera T1
MOV IE,#85H
; zezwolenie na przerwanie zewnętrzne
SETB IT0
SETB IT1
; przerwanie zewnętrzne wyzwalane zboczem
STOP:
NOP
SJMP STOP
; oczekiwanie na przerwanie
;-----
; Stała "TEKST" zostanie wysłana do PC
;-----
TEKST: DB "Test portu szeregowego mikrokomputera AVT-107 \n\r"
;-----
; Procedura obsługi przerwania zewnętrznego
;-----
ORG 200H
EXIN:
; obsługa przerwania zewnętrznego
MOV DPTR,#TEKST ; do DPTR adres tekstu
MOV R2,#31H ; długość tekstu do wysłania
CLR A ; zeruj akumulator
MOVC A,@A+DPTR ; załadowanie do akumulatora znaku
; do wysłania przez RS
WAIT:
JNB TI,WAIT ; oczekiwanie na koniec transmisji
CLR TI ; zerowanie znacznika końca transmisji
MOV SBUF,A ; wysłanie znaku
INC DPTR ; zwiększenie DPTR na następny znak
CLR A ; zeruj akumulator
DJNZ R2,WAIT ; skok do WAIT gdy R2<>0
; R2=0 koniec transmisji
;-----
; Zerowanie zgłoszonych przerw,
; jeśli wystąpiły podczas transmisji
;-----
CLR IE0
CLR IE1
RETI ; powrót z przerwania zewnętrznego
END

```

Listing 2

```

/*
program czytający znaki z portu COM1 i wypisujący je na ekranie
do testowania mikrokomputera 8051 (AVT-107)
wciśnięcie dowolnego klawisza = wyjście z programu
*/
#include <dos.h>
#include <conio.h>
void main(void)
{
union REGS r,out_r;
r.h.ah = 0;
r.h.al = 0x83;
r.x.dx = 0;
int86(0x14,&r,&out_r); /* inicjalizacja COM1 na 1200 baudów,
bez bitu parzystości,
1 bit stopu,
długość danej = 8 bitów */

while (!kbhit())
{
while (!(out_r.h.ah & 0x01)) /* czy znak gotowy do odczytu? */
{
r.h.ah = 0x03;
r.x.dx = 0x00;
int86(0x14,&r,&out_r);
if (kbhit())
break;
}
r.h.ah = 0x02;
r.x.dx = 0x00;
int86(0x14,&r,&out_r); /* czytaj COM1 */
putch(out_r.h.al); /* pisz znak na ekran */
}
}

```

**WYKAZ ELEMENTÓW**

**Rezystory**

- R1: 8,2kΩ
- R2: 2,7kΩ
- R3: 4,7kΩ \*

**Kondensatory**

- C1: 10μF
- C2, C3: 30pF
- C4, C6, C8, C9, C10: 100nF
- C5: 1000μF/16V
- C7: 100μF
- C11, C12, C13, C14: 22μF \*

**Diody**

- D1, D2, D3, D4: 1N4001
- D5: LED

**Układy scalone**

- U1: 80C51
- U2: 74ALS573
- U3: 27C256
- U4: 62256 \*
- U5: MAX232 \*
- U6: LM7805

**Różne**

- INT, RES: mikroprzełącznik \*
- J1, J2, J3, J4, J5, J6, J7: jumper \*
- X1: kwarc 6MHz
- Z1: CONN DB9 \*

\* elementy, które nie są niezbędne

wanych, wykonanych wg koncepcji płytek uniwersalnych. Ich widoki znajdują się na wkładce.

Z pewnością znajdzie się wiele osób, których możliwości opisywanego tu systemu nie zadowolą, ale ma to być zaledwie wprowadzenie w świat techniki mikroprocesorowej. Autor ze swej strony deklaruje chęć rozwijania tej konstrukcji i w najbliższym czasie zostaną opublikowane opisy układów „przerabiających” ten skromny komputer na alarmową centralkę mikroprocesorową ze sporymi możliwościami programowania oraz zegar cyfrowy z budzi-

kami, sterowanymi wyjściami zegarowymi, itp. itd. Możliwości tych urządzeń ograniczona tylko fantazja programisty. Prowadzone są starania o rozwój aplikacji - jest to także zachęta dla Czytelników - jeżeli znajdzie się ktoś z dobrym pomysłem, ale nie mający odpowiednich warunków do jego realizacji jest duże prawdopodobieństwo realizacji koncepcji przez konstruktorów Redakcji EP. Zachęcamy do wspólnej zabawy! Jest naprawdę wciągająca....

**Piotr Zbysiński, AVT**

<sup>1)</sup> Wrzaz z kitem AVT-107 można zamówić odbitkę ksero rozdziału "Programmer's guide and instruction set" - 55 str, cena 66.000 zł.