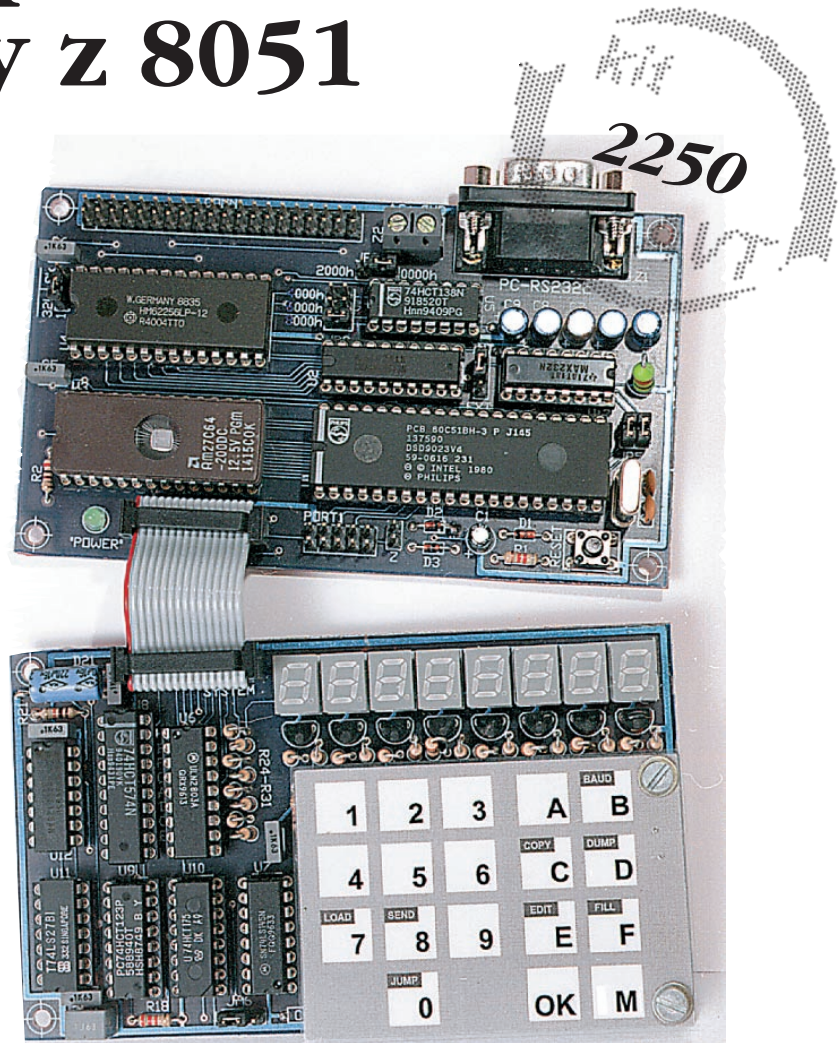


# Mikrokomputer edukacyjny z 8051

„Przygnieci” olbrzymią ilością listów od Czytelników, gorąco zainteresowanych tematem nauki programowania procesorów 8051 postanowiliśmy nieco wyprzedzić teorię przedstawianą w cyklu artykułów „Mikrokontrolery? To takie proste...” i opublikować praktyczny układ podstawowego systemu mikroprocesorowego, który będzie bazą sprzętową podczas przyszłych lekcji nauki pisania programów na '51-kę. Wyprzedzenie ma też na celu danie odpowiedniej ilości czasu Czytelnikom na zmontowanie i uruchomienie prezentowanego urządzenia, tak że w momencie rozpoczęcia praktycznych lekcji programowania, każdy krok prezentowany przez autora w artykule będzie można sprawdzić w praktyce na „pierwszym własnym systemie mikroprocesorowym, zmontowanym w domowym zaciszu...”.



Zaprojektowany przez autora i sprawdzony w redakcyjnym laboratorium „systemik” nazwany przez nas bez przesady „mikrokomputerkiem” będzie podstawą przy nauce programowania procesorów 8051. Dzięki niemu i zawartemu w nim programowi zwanemu dalej „monitorem” początkujący w dziedzinie procesorów będą mieli możliwość stawiania pierwszych kroków w pisaniu własnych mniej lub bardziej złożonych aplikacji. Monitor pełni bardzo podobną rolę jak BIOS w prawdziwym rasowym komputerze PC lub każdym innym, czyli pozwala na komunikowanie się procesora z zewnętrznymi układami dołączonymi do niego. W naszym przypadku będą to podstawowe „więzi” łączące 8051 z nami, czyli: klawiatura i wyświetlacz, oraz z komputerem (np. PC) : złącze transmisji szeregowej w standardzie RS232c.

Stworzony przez autora „monitor” i zawarte w artykułach przykłady programowania umożliwią także naukę osobom nie mającym w domu komputera. Każdy jednak musi sobie zdać sprawę, że chcąc po-

siąść głębszą wiedzę n.t. mikroprocesorów, wcześniej czy później takowy sprzęt chociażby w postaci pocziwego PC-XT (obecnie spotykany jedynie w muzeach, wersja PC-AT do nabycia na giełdach za kilkadziesiąt zł) trzeba będzie nabyć.

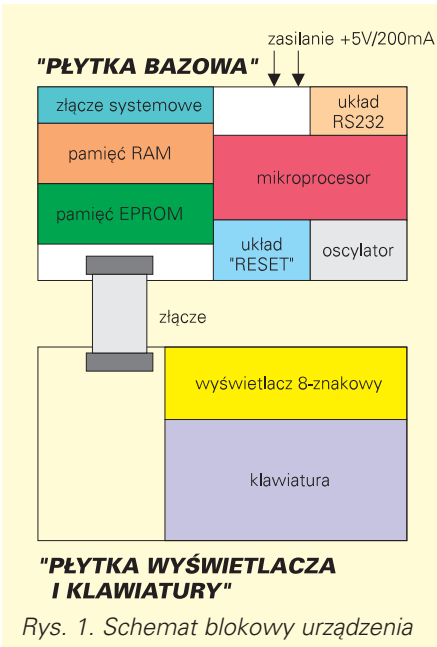
Przy projektowaniu autor kierował się maksymalnym uproszczeniem całej konstrukcji do niezbędnego minimum, pozwalającego jednak na swobodne „surfowanie” po '51-ce, z wykorzystaniem wszystkich jego możliwości, opisywanych w cyklu teoretycznym w EdW. Niemalże znaczenie miało tu też zminimalizowanie kosztów. Niestety nie udało się nie przekroczyć bariery kilkudziesięciu złotych, lecz ten jednorazowy poniesiony wydatek w przyszłości z pewnością przyniesie wymierne efekty, czego najlepszym przykładem jestem ja – wasz autor. „...Bo wszystko zaczęło się dawno temu od tego, kiedy zmontowałem właśnie taki systemik.....”, a teraz chcę Wam, drodzy Czytelnicy, go zaprezentować.

I jeszcze jedna uwaga. Być może dla niektórych z Was prezentowany układ

wyda się bardzo skomplikowanym i np. nie zrozumiecie wszystkiego o czym napiszę, nie przejmujcie się tym. Mogę zapewnić, że wszystkie opisane w układzie elementy „przetrawiać” będziemy jeszcze niejednokrotnie, toteż z pewnością w trakcie mającego się już niedługo rozpocząć kursu programowania, wyjaśnicie sobie wiele rzeczy, co do których możecie mieć wątpliwości. Ja hołduję zasadzie, że „najlepszą nauką jest praktyka...”, a tej jest najwięcej w elektronice, wierzcie mi. Dlatego prezentowany mikrokomputer będzie niezbędny każdemu z Was, kto zechce poznać tajniki programowania procesorów '51, a w przyszłości także wielu innych.

## Opis układu

Schemat blokowy całego systemu mikroprocesorowego przedstawia rysunek 1. Układ elektryczny naszego komputerka składa się z dwóch części. Pierwsza część to zestaw – płytka składająca się z procesora, zewnętrznymi pamięci programu i danych, dekodera adresowego oraz dodatko-



wych układów: resetu i układu współpracy z portem szeregowym komputera PC.

Druga płytką to moduł zawierający 8-pozycyjny wyświetlacz 7-segmentowy, 18-klawiszową klawiaturę wraz z układami dekodującymi i dopasowującymi do współpracy z procesorem 8051. Obie części opiszemy oddzielnie. Tak więc zaczynamy.

## Płytką bazowa

Schemat elektryczny płytki bazowej zawierającej mikroprocesor przedstawiono na **rysunku 2**. Tu proszę Czytelników o cierpliwość i spokój, ten na pozór dość skom-

plikowany schemat nie jest tak straszny, jak się za chwile okaże. Przypominam jeszcze raz że analizę możesz rozpocząć pod warunkiem że zapoznałeś się z IV cz. artykułu z serii „Mikrokontrolery...”, to takie proste” zamieszczonym w tym numerze EdW.

Sercem całego urządzenia jest znany Ci już mikrokontroler 80C51 – U1. W naszym układzie można także z powodzeniem zastosować opisywaną już wersję 80C52, lub obie starsze wersje wykonane w technologii HMOS (8051/8052).

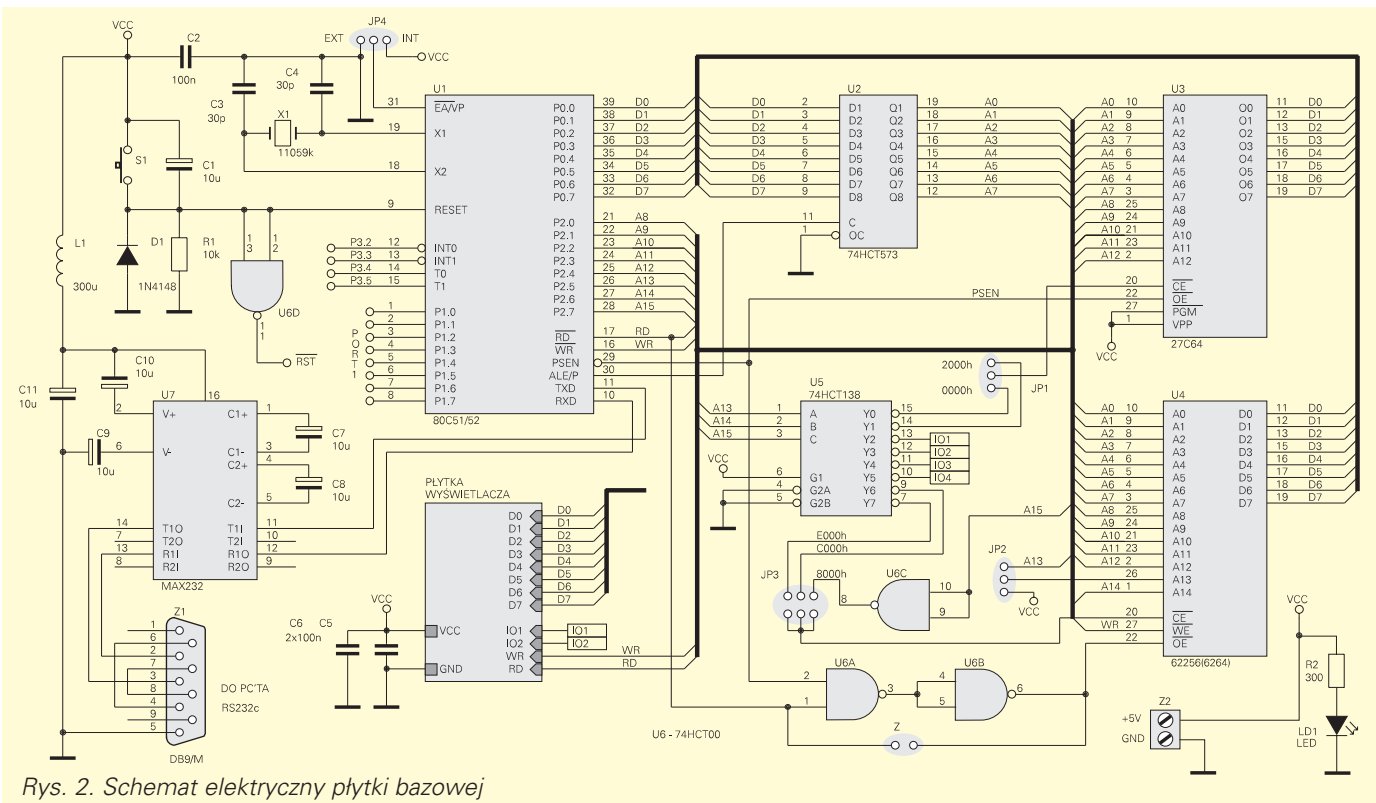
Popatrzmy, z lewej strony u góry widzimy znajomy układ zewnętrznego obwodu rezonatora kwarcowego X1 wraz z kondensatorami C3 i C4. Znaczenie tych elementów już znasz. Nieco niżej widać układ „resetu” złożony z elementów C1-S1 oraz dodatkowo R1 i D1. Klawisz S1 służy do ręcznego resetowania całego komputera, jego wciśnięcie powoduje zwarcie końcówki RESET procesora U1 do plusa zasilania, co w efekcie prowadzi do jego skasowania. Dodatkowa bramka NAND U6d wyprowadza ten sygnał lecz zanegowany na złącze systemowe CONN1, które nie jest pokazane na schemacie z rysunku 2 dla zwiększenia czytelności rysunku. Opis tego złącza oraz dodatkowych złączy znajdujących się na płytce drukowanej przedstawimy w części opisującej montaż układu. Końcówki 12...15 procesora oraz cały port 1 (końc. 1...8) pozostają niedołączane w celu dowolnego wykorzystania w naszych przyszłych aplikacjach.

Teraz spójrzmy na prawo od mikrokontrolera. Widać tu znany już Czytelnikom

układ zatrasku adresowego U2, w roli którego pracuje zachwalany przez autora kursu 74HCT574 (lub LS). Dalej mamy dołączoną zewnętrzną pamięć programu w postaci EPROM – U3 o pojemności 8kB (27C64). Dołączenie końcówek adresowych A0...A12 danych D0...D7 oraz sygnału odczytu /PSEN wydaje się być jasne. Końcówki tej pamięci o numerach 1 i 27 są wykorzystywane jedynie przy programowaniu pamięci EPROM (w odpowiednich programatorach) i przy normalnym użyciu układu powinny zawsze być zwarte do plusa zasilania (+5V).

Zanim wyjaśnię sposób dołączenia wejścia /CE pamięci U3, przyjrzyjmy się układowi U5 – który w naszym układzie pełni rolę dekodera adresowego. Mniej wtajemniczonych w technikę cyfrową uspokajam, że temat ten poruszymy przy okazji kursu programowania procesorów 8051. Jako U5 pracuje zwykły dekodery 1 z 8, którego wejścia wyboru A,B,C (piny 1,2,3) dołączone są do trzech najstarszych linii adresowych A13, A14, A15. W ten sposób w zależności od adresu jaki wystawia mikroprocesor na końcówki portów będzie uaktywnione jedno z wyjść dekodera (logiczne „0”) uaktywniając tym samym układ dołączony do jego wyjścia. W **tabeli 1** przedstawiono zakresy adresów i odpowiadające im, aktywne wyjścia dekodera.

Jak widać cała 64 kilobajtowa przestrzeń adresowa procesora U1 została podzielona na 8 części po 8kB, które są wybierane właśnie za pomocą 3 najstarszych linii adresowych i dekodera U5.



Jak się zaraz okaże w każdej z tych części pracuje odrębny układ komputerka, a dzięki kilku zworom można łatwo zmienić przydzielony mu adres.

Tabela 1

A15,A14,A13	adres	aktywne wyjście
0, 0, 0	0000h...1FFFh	Y0
0, 0, 1	2000h...3FFFh	Y1
0, 1, 0	4000h...5FFFh	Y2
0, 1, 1	6000h...7FFFh	Y3
1, 0, 0	8000h...9FFFh	Y4
1, 0, 1	A000h...BFFFh	Y5
1, 1, 0	C000h...DFFFh	Y6
1, 1, 1	E000h...FFFFh	Y7

Dwa pierwsze obszary po 8 kB przydzielono zewnętrznej pamięci programu (U3-EPROM). Zworka JP1 określa który z nich jest aktywny. Wprowadzono ją po to aby rozszerzyć możliwości naszego komputerka o tryb pracy procesora z wewnętrzną i zewnętrzną pamięcią programu. Zagadnieniem tym zajmiemy się dokładnie przy innej okazji, kiedy już poznamy możliwości systemu podczas nauki programowania.

Kolejne cztery obszary (po 8kB) nazwano umownie jako IO1...IO4. Pierwszy i drugi z nich (IO1 i IO2) wykorzystywane są do uaktywnienia płytki wyświetlacza i klawiatury. Pojawienie się logicznego „0” (przy stanie linii adresowych wg. tabeli 1) na wyjściu dekodera IO1 spowoduje uaktywnienie sekcji wyświetlacza, natomiast wyjście IO2 uaktywnia klawiaturę umożliwiając tym samym jej odczyt.

Dwa ostatnie wyjścia dekodera dołączono do jumpera JP3, dzięki któremu możliwe jest odpowiednie, w zależności od potrzeb, skonfigurowanie zewnętrznej pamięci danych której rolę pełni układ U4. W naszym układzie przewidziano możliwość zastosowania dwóch typów pamięci najbardziej rozpowszechnionych na rynku o pojemnościach 8kB (typ 6264) i 32kB (typ 62256). W zależności od tego którą z nich zastosujemy należy odpowiednio za pomocą jumpera JP2 dołączyć wyprowadzenie 26 tego układu do plusa zasilania (dla 6264) lub do linii i adresowej A13 (dla 62256).

Dodatkowo na jumper JP3 doprowadzono zanegowany sygnał z linii najstarszej adresowej A15. Linia ta decyduje przecież o tym czy obsługiwana jest część pamięci o adresach 0000h...7FFFh (pierwsze 32kB, A15=0), czy druga 8000h...FFFFh (drugie 32kB, A15=1). Jeżeli teraz zewrzymy za pomocą jumpera JP3 wejście wyboru SRAM U4 z tą linią to zauważmy że pamięć ta będzie obsługiwana (zapisywana lub odczytywana) począwszy od adresu 8000h (A15=1, -> zanegowane przez U6c daje w efekcie logiczne „0” to podane na wejście wy-

boru /CE układu U4 spowoduje jego uaktywnienie).

A do jakiego adresu pamięć będzie obsługiwana? Otóż w przypadku pamięci 32kB odczywiście do adresu maksymalnego, czyli FFFFh. Kiedy zaś włożymy w podstawkę jako U4 kostkę SRAM 6264 (8kB), adresem końcowym będzie 8000h + 8kB, czyli z prostego rachunku: 9FFFh (każde 2000h to przecież dziesięćnie 8192, czyli 8kB).

Zaraz, zaraz,...ale przecież np. dla adresu A000h linia A15 też będzie uaktywniała tę pamięć SRAM, co wtedy? Tak moi drodzy, wtedy adres zostanie tzw. „przewinięty”, czyli począwszy od adresu A000h będą od początku odczytywane kolejne komórki pamięci tak jak spod adresu 8000h. Zauważmy że to samo będzie dla adresu C000h i E000h. Można powiedzieć że w przestrzeni adresowej powyżej 8000h pamięć U4 o pojemności 8kB widziana jest w naszym układzie jako cztery jednakowe kopie – często nazywane „lustrami”. Nie jest to bynajmniej wada i jest charakterystyczne w przypadkach kiedy mamy do czynienia z prostymi dekoderni adresu, takimi jak nasz układ U5.

Pamiętajmy że sytuacja ta dotyczy tylko pamięci 6264, zaś dla kości 32kB (62256) każda z 32768 komórek jej pamięci jest reprezentowana przez oddzielny adres z zakresu 8000h...FFFFh.

Bramki U6a i U6b realizują iloczyn logiczny sygnałów /RD (odczytu z wewnętrznej pamięci danych) i sygnału /PSEN (odczytu z zewnętrznej pamięci programu) – patrz artykuł na str. 40. Dzięki temu pamięć U4 może pracować jako pamięć danych lub pamięć do wykonywania programów napisanych przez Ciebie drogi Czytelniku, a zapisywanych ręcznie za pomocą klawiatury i wyświetlacza, lub ładowanym z komputera PC.

Na płytce drukowanej przewidziano możliwość „ograniczenia” pracy pamięci U4 do trybu jako „tylko pamięci danych”. Do tego celu służy zworka Z, która dołącza sygnał /RD procesor U1 bezpośrednio do wejścia pamięci U4 – /OE. W tym jednak przypadku należy dodatkowo odizolować (np. przez odcięcie nóżki 6 U6) wyjście bramki U6b.

Tryb ten nie będzie jednak wykorzystywany w naszym kursie programowania, a jedynie ma uelastyczyć cały układ i przygotować go na wyzwania, jakie z pewnością postawisz przed nim, drogi Czytelniku, w przyszłości.

Pozostał do omówienia układ U7, który pełni rolę konwertera napięć o poziomach logicznych TTL (+5V, 0V) na poziomach logicznych TTL (+5V, 0V) na poziomie zgodnym ze standardem transmisji szeregowej RS232 (-12V, +12V), wykorzystywanej w komputerach PC, np. w typowym gnieździe dla myszki. Nie będę się wglębiał w teorię i sposób działania tej

## Wykaz elementów

### Płytki bazowa (AVT-2250/1)

#### Rezystory

R1: 8,2k  
R2: 220...330

#### Kondensatory

C1, C7...C11: 10uF/16V  
C2, C5, C6: 100nF  
C3, C4: 30...33pF

#### Półprzewodniki

U1: 80C51 (80C52)  
U2: 74HCT573 (LS573)  
U3: 27C64 EPROM  
U4: 6264 (62256) SRAM  
U5: 74HCT138 (LS138)  
U6: 74HCT00 (LS00)  
U7: MAX232 (ICL232)  
D1: 1N4148  
LD1: LED dowolna

#### Pozostałe

X1: Q 11,0592 MHz  
L1: 220...330 uH dławik  
S1: mikroszwic  
Z1: złącze DB9M do druku  
Z2: ARK2  
JP1, JP2, JP4: złącze typu „goldpin” 1x3  
JP3: jak wyżej, lecz 2x3  
Z: jak wyżej, lecz 1x2  
listwa „goldpin” 2x40 i 2x5 po 1 szt.  
jumpery – 4 szt.  
podstawki pod układy scalone  
płytki drukowane AVT-2250/1

### Płytki wyświetlacza i klawiatury (AVT-2250/2)

#### Rezystory

R2...R9: 390  
R10...R17: 6,8k  
R18: 33k  
R19...R23: 10k (8,2k...11k)  
R24...R31: 68

#### Kondensatory

C6: 1uF stały  
C7, C8, C10...C12: 100nF  
C9: 47uF/10V

#### Półprzewodniki

U8: 74HCT574 (LS574)  
U9: 74HCT123  
U10: 74HCT175 (LS175)  
U11: 74HCT27 (LS27)  
U12: 74HCT125 (LS125)  
U13: 74HCT02 (LS02)  
U14: ULN2803A  
U15: 74LS145  
T1...T8: BC328  
D2...D17: 1N4148 lub podobna  
DL1...DL8: LTS2801 (w.anoda)  
lub odpowiednik

#### Pozostałe

X1: Q 11,0592 MHz  
K1...K18: mikroszwic  
JP6: listwa typu „goldpin” 1x3  
listwa „goldpin” 2x8 – 1szt.  
jumper – 1szt.  
podstawki pod układy scalone  
płytki drukowane AVT-2250/2

**Uwaga:** W ofercie handlowej znajdują się oddzielnie zestawy: AVT-2250/1 – „płytki bazowa” oraz AVT-2250/2 – „płytki klawiatury i wyświetlacza”. Oba można nabyć na ogólnych zasadach sprzedaży kitów AVT.

części, wystarczy wiedzieć że jeden „scalak” (MAX232) oraz kilka „elektrolitów” (C7...C9) zamienia te poziomy, umożliwiając tym samym prawidłową

# Projekty AVT

transmisję pomiędzy naszym systemem a komputerem wyposażonym w gniazdo RS232. Dodatkowy dławik L1 zapobiega przedostawaniu się zakłóceń z układu U7 do szyny zasilającej nasz mikrokomputer. Kondensator C11 dodatkowo filtruje napięcie zasilające konwerter. Jego wyjście połączone jest do typowego gniazda 9-pinowego, co umożliwia bezproblemowe połączenie z PC tem za pomocą typowego kabla, którego opis przedstawimy w kolejnym odcinku.

## Płytką wyświetlacza

Do tej części układu dochodzą, jak widać z rysunku 2, sygnały szyny danych D0...D7, oraz dodatkowe: odczytu i zapisu: /RD i /WR, oraz sygnały uaktywniające: IO1 (wyświetlacz) i IO2 (klawiaturę). Spójrzmy na nieco skomplikowany sprzętowo schemat ideowy tej części przedstawiony na **rysunku 3**. Nie bój się drogi Czytelniku, „nie taki diabeł straszny, jak go... rysuje autor!”. Jak się za chwilę przekonasz działanie tej części nie jest tak skomplikowane, jak to wygląda na pierwszy rzut oka.

Wyświetlacz jak i klawiatura pracują w trybie multipleksowania, czyli w jednej chwili aktywny jest tylko jeden wyświetlacz oraz możliwe jest zbadanie tylko jednej sekcji klawiatury.

Zacznijmy od wyświetlaczy. Anody DL1...DL8 zasilane są poprzez tranzystory PNP T1...T8 za pośrednictwem wyjść dekodera 1 z 10 U15, w roli którego pracuje znany dobrze z domowego podwórka układ TTL 74LS145. Pojawienie się lo-

gicznego „0” na jednym z wyjść tego układu powoduje załączenie odpowiedniego tranzystora i w konsekwencji zasilanie jednej z ośmiu pozycji wyświetlacza. Rezystory w bazach wszystkich tranzystorów odpowiednio polaryzują je. Trzy najmłodsze wejścia dekodera U15 dołączone są do 4-bitowego zatrasku U10 (74LS175). Zadaniem tego ostatniego jest zapamiętanie jednej z ośmiu pozycji wyświetlacza, w tym celu wejścia D1, D2, D3 układu U10 dołączone są bezpośrednio do szyny danych mikroprocesora – do 3 najmłodszych jej bitów (D0...D2).

Zauważmy teraz że jeżeli procesor zadresuje obszar o adresie 4001h (patrz rys.2) uaktywniony zostanie sygnał IO1 z dekodera adresowego U5, przyjmując logiczne „0”. Dodatkowo przy zapisie przez procesor sygnał /WR przyjmie stan „0”, co w efekcie spowoduje pojawienie się tego na wejściu 3 bramki U11b. Wejście 4 tej bramki też będzie w stanie „0” bo IO1=0, natomiast za pośrednictwem linii A0, której stan (dla adresu 4001h) jest równy „1” a poprzez bramkę U11c jest negowany, wejście 5 bramki U11b także będzie w stanie logicznego „0”. W efekcie na wyjściu 6 tej bramki pojawi się logiczne „1”, co po podaniu na wejściu CLK U10 spowoduje zapisanie w zatrasku 3 najmłodszych bitów które określą zapalaną pozycję wyświetlacza. Trudne? Jeżeli tak, radzę przeczytać cierpliwie ten fragment jeszcze raz.

Sprawę pozycji wyświetlacza mamy z głowy, do szczęścia, a raczej do „zapale-

nia” znaku pozostaje nam odpowiednie wysterowanie segmentów A...G oraz kropki (nazywanej w przyszłości segmentem H lub DP). Realizowane to jest za pośrednictwem układu U8 (74HCT574) który steruje układem wykonawczym U14 – ULN2803. Wyjścia tego ostatniego dołączone są poprzez rezystory ograniczające prąd do segmentów wszystkich pozycji na raz.

Układ U8 jest 8-krotnym zatraskiem, aktywowanym (w odróżnieniu do 74HCT573) dodatnim zboczem sygnału podanego na jego wejście CLK. Zapis aktualnej cyfry – czyli kombinacji 0 i 1 w kolejności odpowiadającej ułożeniu segmentów na wyświetlaczu odbywa się bardzo podobnie jak w przypadku zapisu numeru pozycji. Różnica polega na tym że przy zapisie mikroprocesor powinien podać adres 4000h, czyli z wyzerowaną linią adresową A0. W konsekwencji tego na wejściach bramki U11a pojawią się logiczne zera, bo:

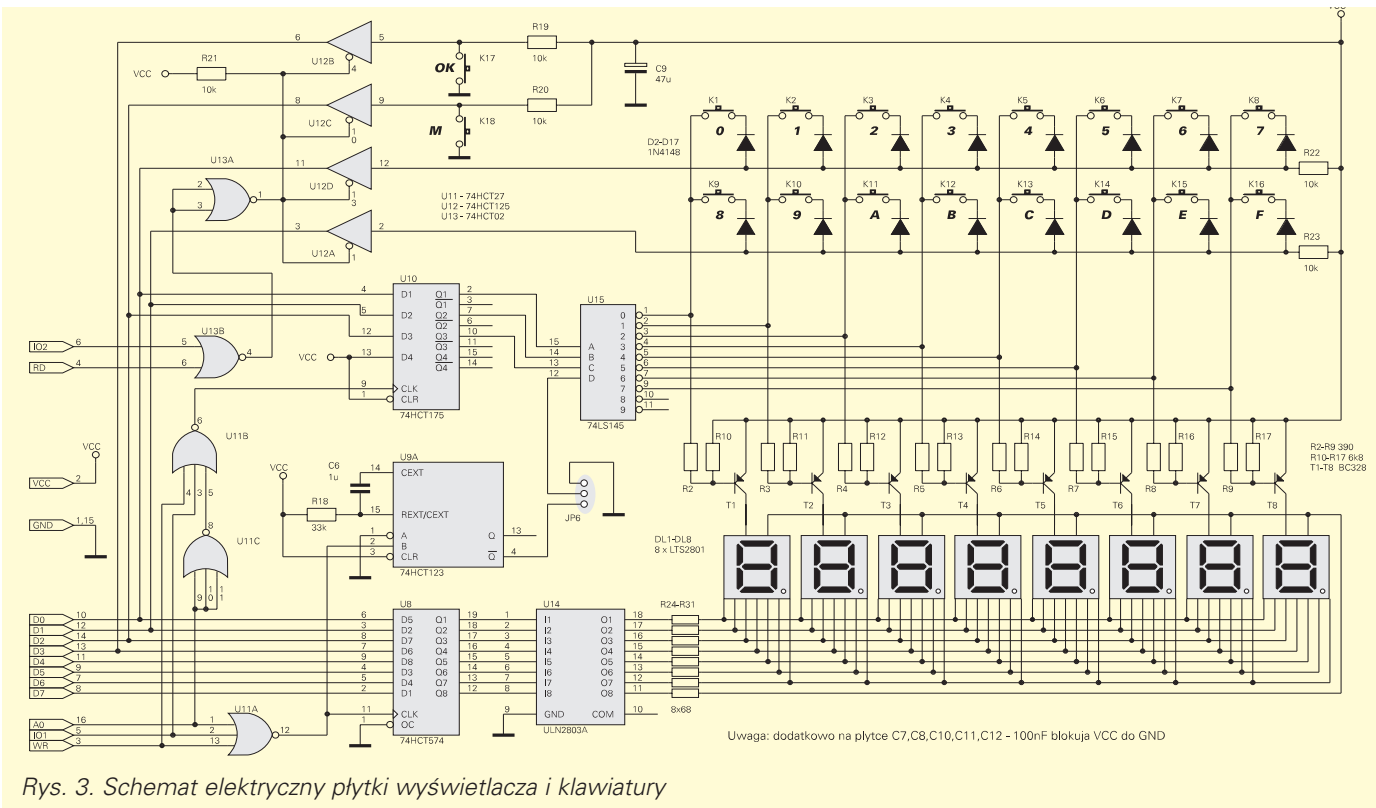
- sygnał A0=0 bo adres = 4000h
- sygnał /WR=0 bo procesor zapisuje
- sygnał IO1=0 bo dla adresu 4000h wyjście dekodera U5 (rys.2) IO1 jest =0.

W efekcie na wyjściu bramki U11a pojawi się logiczna „1”, której dodatnie zbocze zapisze w rejestrze U8 stan linii i danych D0...D7, określających w tym momencie cyfrę lub znak zapalany na wyświetlaczu. Kolejność bitów i odpowiadająca im segmenty są następujące:

bity: 7-6-5-4-3-2-1-0

segmenty: H-G-F-E-D-C-B-A

Dzięki temu upraszcza się kodowanie poszczególnych cyfr czy symboli pokazywa-



Rys. 3. Schemat elektryczny płytki wyświetlacza i klawiatury

nych na wyświetlaczu, np. dla zapalenia cyfry „4” trzeba podać na szynę danych bajt: 01100110 (binarnie), prawda że proste !

No dobrze ale po co jest ten generator monostabilny zbudowany z wykorzystaniem połówki układu U9? Otóż jest to proste zabezpieczenie wyświetlaczy przez „przeciążeniem” w wypadku zawieszenia się procesora lub ewentualnie nieprawidłowej obsługi przez wykonanie błędnego programu. Zauważmy wszakże, że wartość rezystorów w segmentach: R24...R31 jest bardzo mała, lecz konieczna ze względu na to że każda pozycja świeci się tylko przez 1/8 całego okresu. Jeżeli teraz np. w sytuacji awaryjnej procesora, wyświetlacz po prostu „stanie” – tzn. że będzie aktywna na stałe jedna i ta sama jego pozycja, to dość duży prąd segmentów może przegrzać struktury LED i w konsekwencji je uszkodzić, a tego byśmy nie chcieli.

Dlatego właśnie wprowadzono uniwibrator U9a. Jego wyjście /Q połączone jest z najstarszym wejściem „D” dekodera załączania pozycji U15. Jeżeli procesor pracuje normalnie, to sekwencyjnie zapisuje do rejestru segmentów U8 odpowiednie dane, co w efekcie powoduje pojawianie się impulsów wyzwających U9a na jego wejściu

B. Konsekwencją tego jest generowanie z podtrzymywaniem impulsu niskiego przez ten uniwibrator. To powoduje aktywację jednego z 8-miu pierwszych wyjść dekodera U15 i zapalenie kolejnych pozycji wyświetlacza.

Jeżeli oczywiście procesor przestanie to robić odpowiednio często (u nas 512 razy na sekundę) generowany impuls po bardzo krótkim czasie ustaloną wartość elementów R18 i C6, się skończy, na

wyjściu /Q U9a pojawi się logiczna „1”, czyli wszystkie pozycje zostaną natychmiast wygaszone. Dodatkowa zwora JP6 umożliwia odłączenie tej funkcji i permanentne uaktywnienie dekodera pozycji U15 poprzez zwarcie wejścia D do masy (czego nie zalecam robić, szczególnie podczas uruchamiania układu lub pierwszych kroków w programowaniu). Oczywiście w przypadku rezygnacji z pożytecznych usług uniwibratora U9, montaż tego układu w płytce jest zbędny.

Pozostaje sprawa odczytu klawiatury. Można powiedzieć ogólnie że klawisze są odczytywane parami w tych samych momentach kiedy uaktywniona jest jedna z ośmiu pozycji wyświetlacza. Popatrzmy przez chwilę na schemat elektryczny (rysunek 3).

Załóżmy że procesor zapisał znak do rejestru U8, po czym zapalił np. pozycję 2 wyświetlacza – aktywna końcówka 2 U15, to na linii dołączonej do tego wyjścia dekodera U15 panuje logiczne „0”. Teraz procesor podając na szynę adresową adres 6000h powoduje uaktywnienie wyjścia dekodera adresowego U5 (patrz rysunek 2) oznaczonego jako IO2. Na jego wyjściu pojawia się „0”. Wraz z nadejściem sygnału żądania odczytu przez procesor, sygnał /RD także przyjmie poziom „0”, co w konsekwencji spowoduje pojawienie się logicznej jedynki na wyjściu bramki U13b, a po zanegowaniu przez U13a, spowoduje uaktywnienie trójstanowych bramek zawartych w układzie U12, które „przeniosą” stany linii dołączonych do wejść tych bramek. Dzięki rezystorom R19, R20, R22, R23 przy nie naciśniętym żadnym klawiszu na wyjściach pojawią się logiczne „1”-ki które za pośrednictwem linii D0...D3 szyny danych odczytane zostaną przez procesor.

Teraz jeżeli np. naciśniemy klawisz „9”, zwieramy tym samym poprzez diodę wejście bramki U12a. Na jej wyjściu pojawi się także „0” co odczyta procesor i stwierdzi naciśnięcie klawisza. U1 dzięki znajomości, która aktualnie pozycja wyświetlacza jest aktywna, może łatwo obliczyć pozycję naciśniętego klawisza i stwierdzić że był to klawisz K10. Kwestią umowną i leżącą w rękach programisty jest nadanie mu akurat cyfry „9”. Tak dzieje się dla pozostałych klawiszy dołączonych sekcjami do wyjść dekodera U15. Trochę bardziej uprzywilejowany odczyt mają dwa specjalne klawisze K17 i K18. Zauważmy że przy każdym odczycie klawiatury przez procesor U1, niezależnie od aktualnie aktywnej kolumny wyświetlacza, procesor może stwierdzić fakt naciśnięcia jednego z tych klawiszy (lub obu naraz), odczytując stan linii D3 i D4 szyny danych.

Ta dodatkowa możliwość odmiennego odczytu klawiszy M i OK będzie potrzebna przy okazji nauki programowania.

Zastosowane diody D2...D17 mają za zadanie zapobiec błędnemu wyświetlaniu informacji na wyświetlaczu w przypadku naciśnięcia kilku klawiszy na raz, co jest efektem zwarcia kilku wyjść dekodera U7 ze sobą (przy braku tych diod oczywiście).

Jeżeli przebrnąłeś drogi Czytelniku przez ten wyczerpujący opis i rozumiesz jak działa moduł wyświetlacza i klawiatury, to dobrze! Jeżeli nie, to przeczytaj artykuł jeszcze raz następnego dnia, sięgając w razie potrzeby do katalogów układów serii TTL. Sądzę że nie będzie to jednak potrzebne.

**Sławomir Surowiński**

\* Opis montażu systemu zostanie opublikowany w następnym numerze EdW.

## Wykaz elementów

Płytki bazowa (AVT-2250/1)

### Rezystory

R1: 8,2k  
R2: 220...330

### Kondensatory

C1, C7...C11: 10uF/16V  
C2, C5, C6: 100nF  
C3, C4: 30...33pF

### Półprzewodniki

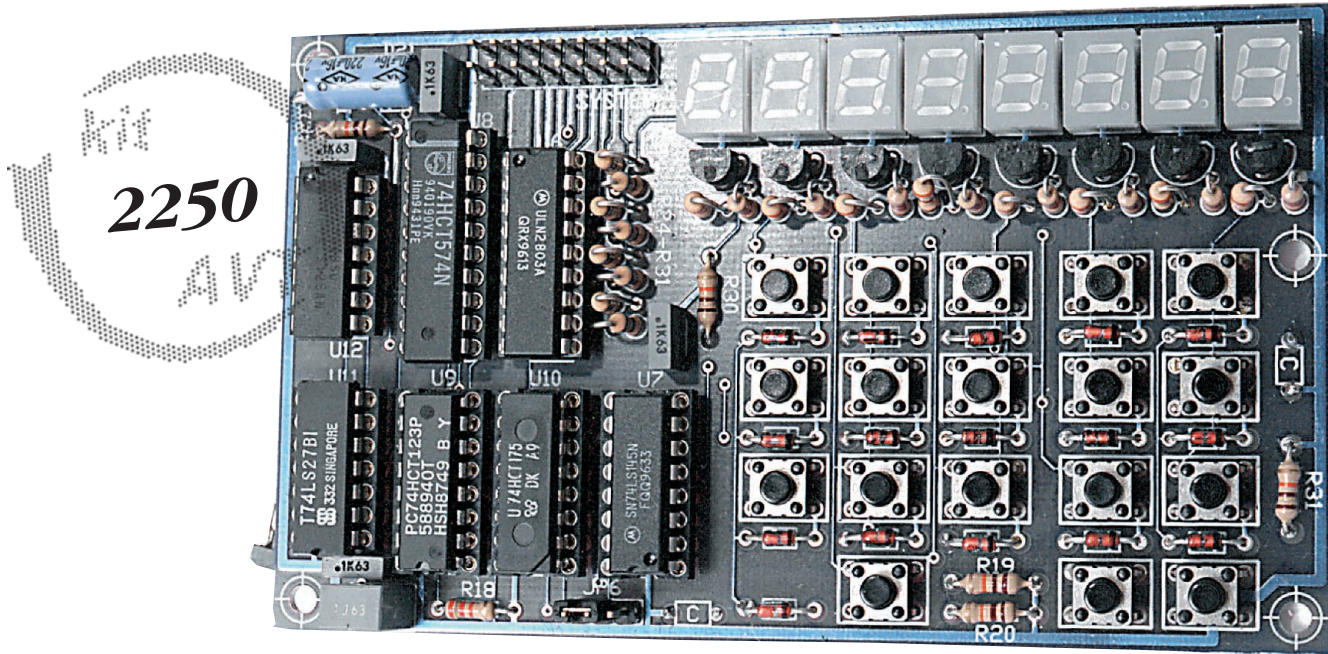
U1: 80C51 (80C52)  
U2: 74HCT573 (LS573)  
U3: 27C64 EPROM  
U4: 6264 (62256) SRAM  
U5: 74HCT138 (LS138)  
U6: 74HCT00 (LS00)  
D1: 1N4148  
LD1: LED dowolna

### Pozostałe

X1: Q 11,0592 MHz  
L1: 220...330 uH dławik  
S1: mikroswicz  
Z1: złącze DB9M do druku  
Z2: ARK2  
JP1, JP2, JP4: złącze typu „goldpin” 1x3  
JP3: jak wyżej, lecz 2x3  
Z: jak wyżej, lecz 1x2  
listwa „goldpin” 2x40 i 2x5 po 1 szt.  
jumpery – 4 szt.  
podstawki pod układy scalone  
płytki drukowana AVT-2250/1

# Mikrokomputer edukacyjny z 8051

część 2



## Montaż i uruchomienie

Cały układ elektryczny zmontowano na dwóch dwustronnych płytkach drukowanych z metalizacją otworów. Niestety ze względu na ilość elementów nie udało się wykonać wersji jednostronnej przy porównywalnym koszcie obu płytek.

Na rysunkach 4 i 5 przedstawiono rozmieszczenie elementów obu płytek mikrokomputerka: bazowej i wyświetlacza. Dodatkowo zmontowane urządzenie widoczne jest na zdjęciach w artykule. Po-

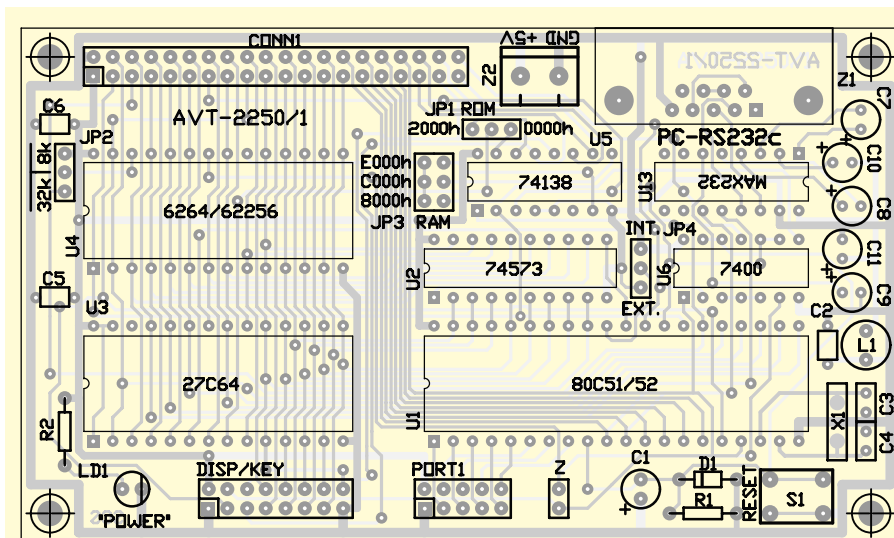
nieważ jednak jest to egzemplarz prototypowy układu, nie widać na nim dwóch układów scalonych (U6 i U13), zostały one dolutowane od spodu. Wszyscy nabywcy zestawu AVT-2250 otrzymają płytki drukowane takie same jak przedstawiono na rysunkach montażowych, dlatego ich tylko należy się trzymać montując cały układ, a nie zdjęć prototypu.

Ze względu na dość duże upakowanie elementów, należy bezwzględnie przestrzegać prawidłowych zasad montażu.

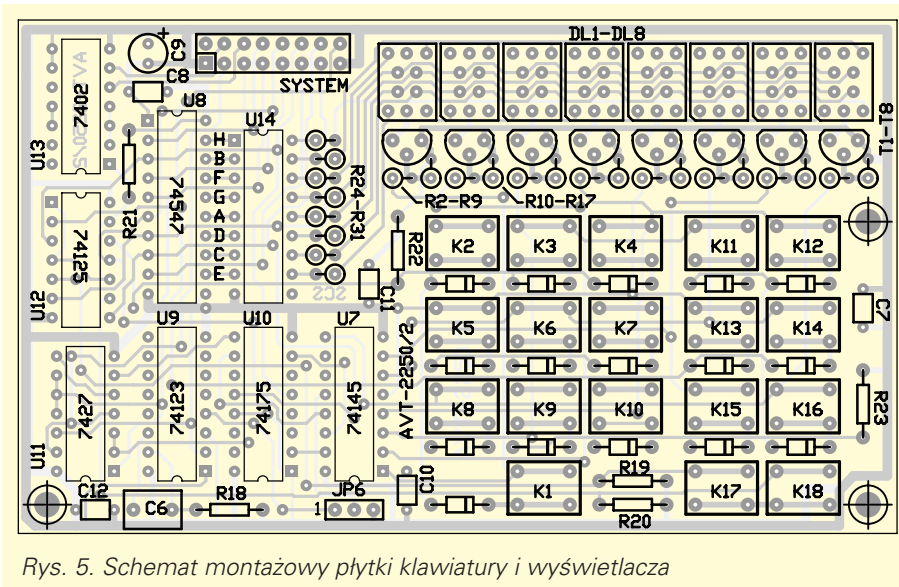
Do wlotowywania elementów najlepiej jest posłużyć się stacją lutowniczą o mocy max. 40W. W ostateczności wprawieni elektronicy amatorzy mogą dokonać montażu przy pomocy zwykłej transformatorówki z dobrze wykonanym grotem, przełączając moc na max. 45W. Pod żadnym pozorem nie należy używać popularnych niegdyś w składnicach harcerskich dużych lutownic z kolbą, które nadają się jedynie do ... lutowania dachów z blachy. Jak wynika z mojej praktyki w pracowni, wielu czytelników przesyła mi z prośbą o uruchomienie zmontowanego kity AVT, na widok których włos się na głowie jeży. Efektem prawdopodobnie używania lutownicy o zbyt dużej mocy są powypalane punkty lutownicze, przegrzane krople cyny na końcówkach elementów, przerwane i poodklejane ścieżki. Taki układ nie nadaj się do reanimacji a jedynie do kosza.

Dlatego jeżeli nie masz wprawy w posługiwaniu się lutownicą, poproś kogoś znajomego, aby fachowo bez uszkodzeń zmontował Ci obie płytki mikrokomputerka.

Montaż należy rozpocząć od płytki bazowej – kit AVT-2250/1. Kolejność montażu jest następująca: rezystory, dioda D1, podstawki pod układy scalone, dioda LED, kondensatory elektrolityczne a następnie stałe, mikroswicz S1, dławik, złącze portu szeregowego Z1 oraz zasilające Z2.



Rys. 4. Schemat montażowy płytki bazowej



Rys. 5. Schemat montażowy płytki klawiatury i wyświetlacza

Uwaga przy wlotowaniu rezonatora kwarcowego! Ze względu na metalizowane punkty na płytce drukowanej, element ten należy wlotować nieco (1mm) ponad powierzchnią płytki, tak aby metalowa obudowa rezonatora nie zwierała, poprzez punktu lutownicze, jego wyprowadzeń. Można także posłużyć się kawałkiem taśmy samoprzylepnej i „podłożyć” ją pod montowany rezonator.

Złącza typu goldpin należy przed wlotowaniem dopasować, odcinając potrzebne ich długości (ilości pinów) w zależności od typu złącza. I tak dla złącza systemowego (CONN1) będzie to listwa 2x40 pinów, złącza portu P1 (PORT1) listwa 2x5 pinów, jumpera JP3 – listwa 2x3 piny. Podobnie postępujemy przed wlotowaniem pozostałych jumperów: JP1, JP2, JP4, a następnie na wszystkie JPx zakładamy po 1 zworniku (jumper). W zestawach AVT-2250/1, /2 w wersji B założono wystarczającej długości listwy „goldpin”, które należy porozcinać najlepiej za pomocą ostrego nożyka introligatorskiego. Nie należy używać do tego celu np. zaokrąglonych szczypiec, bo może to spowodować odłamanie plastikowej listwy – wspornika, przez co w efekcie stracimy 1 rząd „goldpinów”.

Po zakończeniu montażu płytki bazowej, jeszcze raz sprawdzamy jakość lutowania, usuwamy ewentualne zwarcia pomiędzy punktami oraz dodatkowo sprawdzamy całą płytkę metodą „pod światło”, tak na wszelki wypadek, aby upewnić się że montaż jest w porządku. Bardziej przeczorni mogą przedzwonić dodatkowo płytkę używając do tego celu wskaźnika zwarców lub zwykłego multimetru. Należy przy tym wzorować się schematem ideowym z rysunkiem 2.

Przed włożeniem układów scalonych w podstawki należy bezwzględnie dokonać próby zasilania układu „na sucho”.

Do tego celu należy użyć zasilacza stabilizowanego o napięciu wyjściowym 5V TTL, i wydajności prądowej min. 200mA. Zasilacz taki powinien być zbudowany w oparciu o scalony stabilizator typu 7805, co zapewni prawidłowe warunki zasilania bez możliwości uszkodzenia drogiej kostki. Przykładowe rozwiązanie takiego układu idealnie nadające się do zastosowania w naszym urządzeniu opiszemy w kolejnym numerze EdW. Na razie musi nam wystarczyć chociażby zasilacz laboratoryjny lub popularna „wtyczka” ze stabilizacją na poziomie 5V TTL (TTL tzn. z tolerancją 0,25V).

Po dołączeniu zasilania do „pustego” ciągle układu, dioda LED powinna zaświecić. Należy skontrolować napięcia zasilające na podstawkach wszystkich układów scalonych, a także na złączach („goldpin”).

Jeżeli wszystkie podstawki są zasilane prawidłowo, odłączamy zasilanie, zwieramy na chwilę wyprowadzenia złącza zasilającego Z2 a następnie wkładamy układy scalone w podstawki zwracając szczególną uwagę na ich prawidłowy kierunek.

Jeszcze raz upewniwszy się o prawidłowym kierunku układów scalonych, odkładamy na bok płytkę bazową i zabieramy się do zmontowania płytki wyświetlacza z klawiaturą.

Płytkę tą wymiarami jest dopasowana do bazowej, wykonano ją także w wersji dwustronnej z metalizacją otworów. Postępujemy zgodnie z zasadami omówionymi wcześniej podczas opisu montażu płytki bazowej mikrokomputera.

Dla ułatwienia podaję praktycznie sprawdzoną kolejność montażu: rezystory poziome, diody D2...D17, podstawki pod układy scalone, kondensatory stałe, wyświetlacze, klawiaturę (K1...K18) oraz tranzystory. Następnie należy zamontować rezystory ułożone w pozycji pionowej, czyli R24...R31 oraz R2...R17 – wy-

rażne obrysy na płytce drukowanej ułatwią to zadanie. Dostarczane w zestawie mikroswitcze klawiatury należy wlotować w płytkę wciskając je „do końca”. Umżliwi to potem prawidłowe zamocowanie (wykonanej samodzielnie domowym sposobem !) folii klawiatury.

Po zmontowaniu płytki wyświetlacza, należy dokonać wstępnego sprawdzenia „na sucho”. Po sprawdzeniu poprawności montażu i usunięciu ewentualnej nadwyżki kalafonii (takie sytuacja nie powinna się zdarzać !) sprawdzamy wzrokowo jeszcze raz całość bardzo dokładnie. Teraz można włożyć układy scalone w podstawki na płytce wyświetlacza.

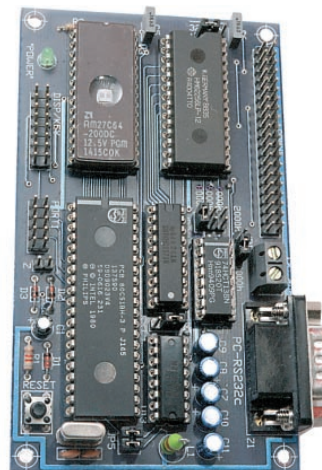
Odkładamy mikrokomputer na bok i przystępujemy do wykonania kabla łączącego obie płytki. Do tego celu służą 2 złącza typu Z-FC16, typowe „zaciskane” na 16-żyłowy kabel taśmowy, który znajduje się w kicie. Tym Czytelnikom, którzy nie wiedzą dokładnie jak prawidłowo wykonać tę czynność przypominam, że w numerze 7/96 EdW w artykule na stronie 10-11 pisaliśmy o tym, toteż serdecznie zapraszam do lektury. Wspomnę tylko że bez imadła lub kawałka płaskiej deski się nie obejdzie.

Przykład prawidłowo zmontowanego kabla przedstawia zdjęcie w artykule. Połączenie pomiędzy obiema płytkami nie powinno być dłuższe niż 5...10 cm.

Teraz można połączyć obie płytki. Układamy je na stole w pozycji jak pokazano na zdjęciu, płytkę wyświetlacza z klawiaturą znajduje się na dole a bazowa na górze a następnie spinamy całość wykonanym wcześniej kablem. Próbnikiem zwać warto jest skontrolować prawidłowe połączenie szyny zasilania +5V i masy GND pomiędzy dwiema płytkami.

Przed uruchomieniem należy uzbroić nasz układ w jumpery, i tak płytce bazowej zakładamy je na:

- JP1: pozycja „0000h”
- JP2: pozycja „8k” (w przypadku użycia jako U4 – 6264), lub pozycja „32k” (dla U4 – 62256)







- JP3: pozycja „8000h”
- JP4: pozycja „EXT.” (praca z zewnętrzną pamięcią programu EPROM U3)
- zwora Z powinna być rozwarta.

Na płycie wyświetlacza jumper JP6 zakładamy w pozycji „od lewej”, czyli łącząc piny 1-2.

Uff, chyba nie było tak ciężko, cierpliwości w praktyce te czynności sprawiają sporo uciechy szczególnie jeżeli za moment cały komputer zasilimy w celu jego przetestowania.

Uruchomienie układu prawidłowo zmontowanego ze sprawdzonych elementów jest bardzo proste. Do złącza zasilającego Z2 dołączamy wcześniej wspomniany zasilacz (+5V TTL/200mA) tu uwaga na polaryzację !!!, co w efekcie powinno spowodować pojawienie się na wyświetlaczach napisu

- HELLO

a następnie po 1 sekundzie napis zniknie a pozostanie zapalona jedynie kropka na pierwszym wyświetlaczu. Dodatkowo po włączeniu zasilania można układ zresetować wciskając na chwilę przycisk S1.

Jeżeli tak się dzieje wstępne uruchomienie można nazwać za zakończone. Działanie klawiatury oraz całego wyświetlacza można przetestować specjalną procedurą zawartą w dostarczanym w kicie EPROM ie, a uruchamianą poprzez naciśnięcie klawisza K10 (cyfra „9”). Komputer najpierw samoczynnie zacznie wypisywać na wyświetlaczu znaki „0123456789ABCDEF-.....”

a następnie po wyświetleniu napisu

PrESS

poprosi o naciskanie klawiszy, których kody wypisywane będą na wyświetlaczu. Wciśnięcie specjalnego klawisza K18 (“M”) zwanego „klawiszem powrotu do monitora” można przerwać działanie procedury testującej i powrócić do stanu oczekiwania, czyli wtedy gdy wyświetlacz jest wygaszony a pali się jedynie pierwsza kropka. Tu uwaga, w wyniku naciśnięcia klawisza K17 („OK”) wyświetlony zostanie znak „minus”.

Dalsze testy dotyczące działania portu szeregowego oraz testowanie pamięci

przeprowadzimy przy okazji kursu programowania procesora 8051.

Przy okazji cały układ w wersji z układami HCT nie powinien pobierać więcej niż 170mA (165 mA w układzie modelowym przy zapalanej na wyświetlaczu sekwencji cyfr „12345678”). Przy zastosowaniu układów serii TTL-LS pobór prądu może nieco wzrosnąć, co zależy też od ilości zapalonych segmentów.

## „Upiększanie” mikrokomputera

Na koniec kilka praktycznych wskazówek na temat wykonania niezbędnej w przyszłości klawiatury. Na wkładce wewnątrz numeru znajduje się szablon takiej klawiatury, który należy dokładnie wyciąć a następnie.. uwaga !... zwyčajnie „zafoliować” udając się do najbliższego punktu kserokopii. Tak wykonaną folię klawiatury przycinamy, tak aby pasowała ona wielkością do płytki wyświetlacza, a następnie wycinamy (lub wiercimy) dwa otwory o średnicy 3mm na jej prawym brzegu. Dzięki nim oraz otworom w tych samych miejscach na płycie drukowanej, za pomocą dwóch śrub M3-10mm przykręcamy folię dystansując ją na odpo-

wiedniej wysokości kilkoma dodatkowymi nakrętkami. Przymocowanie folii tylko w 2 miejscach w zupełności wystarczy, jest ona wystarczająco sztywna, to też z pewnością nie będzie ona podatna na uszkodzenia. Wypróbujmy teraz jej działanie, prawda że proste a jednocześnie doskonałe ...!

Pozostaje sprawa wyświetlaczy. Dobrze jest przykryć je kawałkiem barwionej folii lub filtrem w kolorze uzależnionym od zastosowanych wyświetlaczy. Zdecydowanie poprawi to kontrast całego pola odczytowego, co przy wielu godzinach spędzonych z naszym komputerkiem oszczędzi nam zmęczenia wzroku.

W układzie modelowym do zrobienia filtru wykorzystano odcięty pasek cienkiej pleksi (0,5 mm) który przyklejono do bocznych ścianek wyświetlaczy.

Można także pokusić się o bardziej wyrafinowane rozwiązanie w postaci zafoliowanego wraz z klawiaturą kawałka filtru, decyzja zależy od Ciebie drogi Czytelniku.

Tak wykonany mikrokomputer jest gotowy do pracy. Teraz pora na opis poszczególnych funkcji i sposób programowania... ale to już temat na kolejny artykuł.

**Sławomir Surowiński**

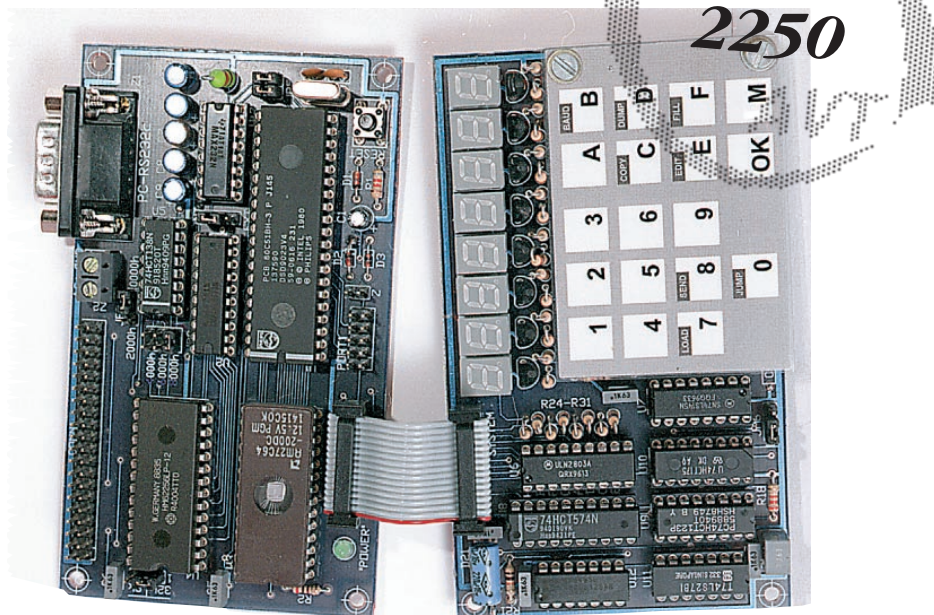
# Mikrokomputer edukacyjny z 8051 – oprogramowanie

## część 3

Kończymy opis komputerka edukacyjnego. W tej części opiszemy niezbędne do uruchomienia wyposażenie dodatkowe, zasilacz oraz kabel połączeniowy z komputerem wyposażonym w złącze transmisji szeregowej.

Wszystkim posiadaczom takiego sprzętu pozwoli to na szybkie ładowanie pisanych przez siebie w przyszłości programów do opisywanego systemu mikroprocesorowego.

Opiszemy też podstawowe funkcje dostępne z poziomu monitora, czyli programu nadzorującego pracę komputerka AVT-2250, a zawartego w pamięci EPROM dostarczanej wraz z zestawem do samodzielnego montażu.



Na początku artykułu małe przeprosiny ze strony redakcji. Sprawa dotyczy szablonu do wykonania klawiatury komputerka. Ponieważ z poprzednim numerze EdW zabrakło we wkładce miejsca, brakujący rysunek zamieszczamy w niniejszym numerze Elektroniki dla Wszystkich.

Pora teraz do wyposażenia naszego komputerka w niezbędne do pracy peryferia: zasilacz i kabel.

### Zasilacz +5V TTL

Ta część artykułu przyda się z pewnością tym z Was którzy, nie polegając na fabrycznych rozwiązaniach, chcą samodzielnie wykonać dobrej jakości zasilacz stabilizowany niezbędny do prawidłowej pracy naszego komputerka. Wystarczająco duża wydajność prądowa oraz niewielkie rozmiary pasujące idealnie do obudowy typu „wtyczka”, predystynują proponowane przez nas rozwiązanie specjalnie do współpracy z układem AVT-2250.

Schemat elektryczny układu zasilacza przedstawia **rysunek 1**. Zastosowano typowy układ ze stabilizatorem scalonym U1 typu 7805. Bezpiecznik B1 zabezpiecza pierwotne uzwojenie transformatora TR1 przed uszkodzeniami, spowodowanymi przecięciami lub przeciążaniem. Mostek M1 wraz z kondensatorami C1 i C2 prostują napięcie zmienne oraz wygładzają je. Stabilizacja na poziomie TTL czyli 5V z tolerancją +/- 0,25V odbywa się dzięki układowi U1, na wyjściu którego zastosowano dodatkowy kondensator elektrolityczny U3 oraz zapobiegający wzbudzeniom kondensator blokujący C4.

Całość mieści się na niewielkiej jednostronnej płytce drukowanej, dostosowanej rozmiarami do obudowy typu Z-21. Rozmieszczenie elementów przedstawia **rysunek 2**. W przypadku umieszczenia zasilacza w takiej budowie, należy przed rozpoczęciem montażu elementów odciąć rogi płytki zgodnie z obrysem, tak

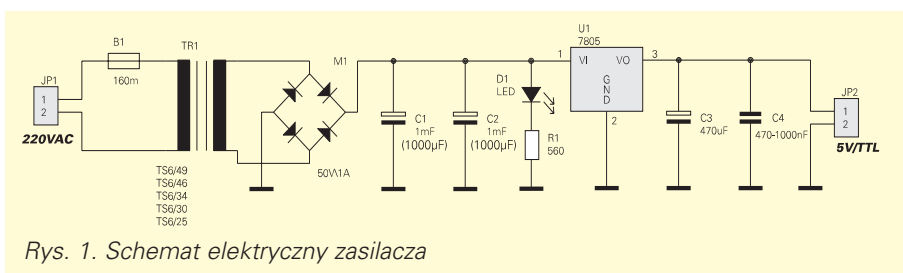
aby dała ona się wcisnąć w głąb dolnej części obudowy.

Montaż jest bardzo prosty. Należy wlotować wszystkie elementy zwracając uwagę na polaryzację. Diodę D1 sygnalizującą załączenie zasilacza należy wlotować na przedłużonych końcówkach, tak aby po zamknięciu pokrywy zasilacza, jej soczewka wystawała przez wywiercony w niej wcześniej otwór.

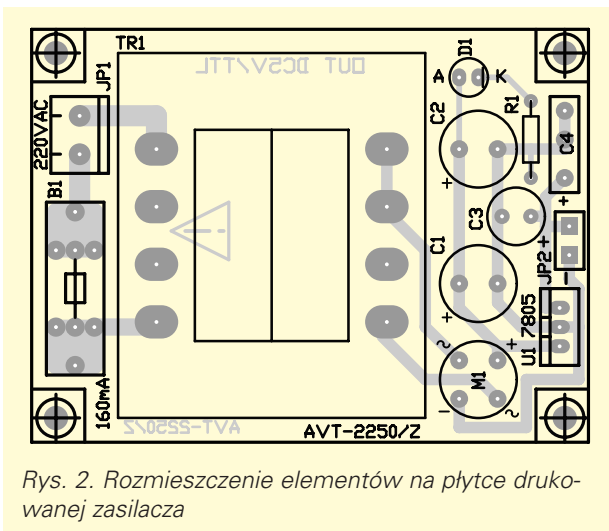
Kabel doprowadzający napięcie zasilające układ komputerka nie powinien być zbyt długi ( $l < 2m$ ). W miejscu wyprowadzenia poza obudowę przewód warto zabezpieczyć kawałkiem rurki termokurczliwej, lub specjalnego giętkiego przepustu dostępnego w handlu. Jako transformator najlepiej jest zastosować typ TS6/49, lecz można także użyć jednego z typów podanych na schemacie z rys.1, zwracając uwagę przy zakupie na zgodność wyprowadzeń z płytka drukowaną. Ze względów bezpieczeństwa nie zaleca się wykonywania przeróbek płytki od strony uzwojenia pierwotnego transformatora.

Po zmontowaniu zasilacza należy sprawdzić jego napięcie wyjściowe, które powinno wynosić +5V (z odchyłką 0,25V).

Uwaga: Wszelkie prace uruchomieniowe zasilacza należy przeprowadzić po kompletnym zmontowaniu całego układu, sprawdzeniu jakości i poprawności



Rys. 1. Schemat elektryczny zasilacza



montażu oraz zamknięciu zasilacza w obudowie. Osoby niedoświadczone powinny skorzystać z pomocy osób obeznanych z zasadami uruchamiania urządzeń zasilanych z sieci energetycznej 220V.

Na zakończenie uwaga dotycząca chłodzenia układu stabilizatora U1. Ponieważ nasz komputer pobiera stosunkowo mały prąd (<200mA) stosowanie radiatora nie jest konieczne. Ci z Was którzy zechcą przygotować proponowany zasilacz na większe obciążenia (>0,5A) mogą do układu U1 przykręcić kawałek cienkiej aluminiowej blachy formując ją przedtem tak aby bez problemu mieściła się pod pokrywą obudowy zasilacza.

## Kabel połączeniowy

Ta część artykułu przeznaczona jest nie tylko dla posiadaczy komputerów PC, ale i dla tych którzy ich nie mają. Otóż opisany poniżej kabel może przydać się także osobom które swoją edukację zaczną na kartce papieru. Za pomocą takiego kabla można będzie także połączyć dwa komputerki AVT-2250 w bardziej rozbudowanych zastosowaniach, których w tej chwili przecież nie można przewidzieć. Być może niektórzy z Was posiadają takie umiejętności programowania, że do spełnienia warunków postawionego przed sobą projektu potrzebny będzie nie jeden ale dwa procesory serii 8051. Dlatego kabel taki warto wykonać już dziś, chociażby dla świętego spokoju.

Oczywiście nie muszę przypominać że kabel jest niezbędny tym z Was którzy posiadają komputer klasy PC i zamierzają korzystać z niego przy nauce programowania przy pomocy naszego komputerka edukacyjnego. Ponieważ prezentowany komputer komunikuje się z komputerem PC poprzez złącze transmisji szeregowej (korzysta z niego np. myszka), niezbędny jest do połączenia stosowany kabel. Prawdę mówiąc odpowied-

ni do tego celu typ można nabyć w sklepie z artykułami komputerowymi. Należy poprosić o „kabel z 9-kontakowymi końcówkami żeńskimi”. Nie muszę w tym miejscu przypominać, że Ci z Was którzy używają w swoim PC-cie myszki, muszą sprawdzić czy mają drugie wolne złącze typu RS-232c. Na szczęście standardowo komputery mają po 2 porty transmisji szeregowej, toteż nie ma się o co martwić, jednak często oba gniazda różnią się od siebie wielkością. Stosuje się gniazda typu DB9M oraz DB25M, czyli odpowiednio 9 i 25 końcówkowe – oba męskie, czyli z widocznymi szpilkami (bolcami). Dodatkowe informacje na ten temat można znaleźć w artykule n.t. łącza szeregowego w numerze EdW 6,7/97.

W wypadku jeżeli wolne złącze COM ma gniazdo typu DB25M, należy dodatkowo zaopatrzyć się w odpowiednią „przelotkę”.

Jeżeli już kupimy kabel w sklepie, należy dodatkowo np. za pomocą próbnika zwarć sprawdzić, czy wszystkie końcówki na obu końcach są ze sobą połączone. Wyjątkiem są końcówki 2 i 3, które powinny się „krzyżować”. Jeżeli tak nie jest, należy rozebrać z jednej strony wtyczkę kabla i korzystając z lutownicy zamienić miejscami przewody od wspomnianych wyprawadzeń, tak aby końcówka 2 przechodziła na 3, i odwrotnie.

Tym którzy chcą tańszym kosztem i samodzielnie wykonać stosowany kabel połączeniowy, zamieszczam odpowiednie rysunki ze schematem połączeń – **rysunek 3a i b**.

Komputer AVT-2250 komunikuje się z komputerem PC w trybie półduplexowym bez sprzętowego potwierdzenia transmisji. Stąd odpowiednie zwory z obu stron kabla wykonane na wtyczkach pomiędzy sygnałami RTS i CTS oraz DSR i DTR. Na rysunku pokazano dwa warianty połączeń, dla posiadaczy wolnego złącza COM typu DB25 (długiego) oraz dla tych którzy mają wolne krótkie złącze – DB9.

Do transmisji wystarczy dowolny 3-żyłowy kabel, najlepiej miedziany typu „plecionka”. Długość kabla może wynosić nawet ponad 20 m (standard RS232c pozwala na stosowanie o wiele dłuższych kabli), jednak przy tak długim połączeniu, warto wiązkę 3 przewodów umieścić w ekranie. W handlu są dostępne odpowiednie przewody 3-żyłowe z ekranem wykonanym z plecionego drutu miedzianego. Ostatecznie można wykorzystać zwykły 3-żyłowy kabel sieciowy (z przewodem uziemiającym), lecz nie będzie to zbyt poręczne rozwiązanie.

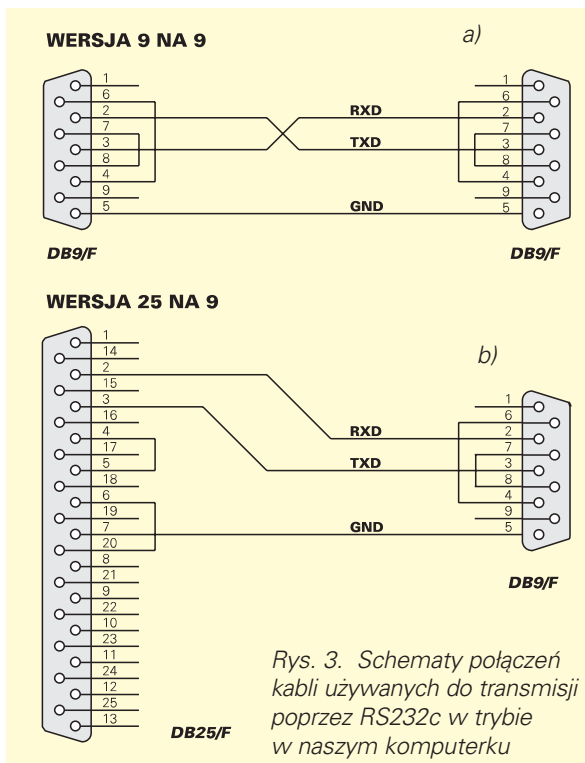
W każdym razie do transmisji wykorzystuje się 3 linie: TXD, RXD oraz GND. Sktóry oznaczają:

- RXD: linia odbioru danych (ang. „Receive data”)
- TXD: linia nadawania danych (ang. „Transmit data”);
- GND: to oczywiście wspólna masa (której nie zalecam łączyć z obudową wtyczki na kablu połączeniowym).

Kwestia nazewnictwa linii jako TXD i RXD jest oczywiście umowna i zależna od tego z której strony patrzymy na system: AVT-2250 <—> komputer PC, przecież dla PC ta linia odbioru danych będzie linią nadawania dla systemu AVT-2250. Ważne jest aby tak jak na rysunku 3, linie te były skrzyżowane.

W zależności od tego jakiego rodzaju wolne gniazdo COM mamy do dyspozycji w naszym PC-cie należy wybrać wariant z rys.3a lub 3b.

Jeżeli na Twoim stole leży już zmontowany i uruchomiony komputer edukacyjny, opisany zasilacz +5V oraz kabel



Rys. 3. Schematy połączeń kabli używanych do transmisji poprzez RS232c w trybie w naszym komputerku

połączeniowy, możesz zapoznać się z oprogramowaniem w które wyposażony jest Twój pierwszy zestaw na 51-kę.

## Monitor – systemowy „BIOS”

Szczerą prawdą jest stwierdzenie, że „komputer bez oprogramowania jest jak żołnierz bez karabinu”. Ta zasada ma swoje uzasadnienie w zestawie AVT-2250 który właśnie uruchomiłeś lub zamierzasz to zrobić.

Otóż aby ułatwić pierwsze kroki w programowaniu, komputer zawiera gotowy (stworzony przez autora) bardzo prosty system operacyjny, który w terminologii układów mikroprocesorowych nazywa się „monitorem”.

Dzięki niemu po włączeniu zasilania, układ umożliwia wykonanie podstawowych operacji wejścia-wyjścia, innymi słowami pozwala na komunikację pomiędzy człowiekiem a systemem, oraz ewentualnie w razie potrzeby pomiędzy systemem a komputerem PC. Podstawowymi układami wejścia-wyjścia dla Ciebie w tym przypadku są: klawiatura i wyświetlacz. W przypadku chęci połączenia się z komputerem PC takim elementem jest port transmisji szeregowy, który łączysz za pomocą wykonanego przed chwilą kabla.

Monitor zawiera kilka użytecznych funkcji, które są dostępne pod niektórymi klawiszami klawiatury. Komendy te to skróty lub wyrazy wzięte z terminologii angielskiej. Są to: LOAD, SEND, JUMP, BAUD, COPY, EDIT, FILL.

Wywołanie dowolnej z nich odbywa się przez naciśnięcie klawisza oznaczonego danym skrótem a następnie wprowadzenie pewnych wartości początkowych w zależności od wywołanej funkcji. Po włączeniu systemu, wyświetlacz pokazuje przez chwilę komunikat powitania: „-HELLO”, po czym zapala się kropka dziesiętna na pierwszym wyświetlaczu. Jest to sygnał zachęty i informacja że system jest gotowy do przyjęcia komendy od użytkownika. Można teraz wywołać jedną z funkcji, których znaczenie i sposób posługiwania się opiszę poniżej.

**LOAD** (klawisz 7) – dzięki tej komendzie możliwe jest wczytanie pliku wynikowego – z kodem maszynowym programu utworzonego przez Ciebie w komputerze PC. Po wciśnięciu klawisza na pierwszym wyświetlaczu pojawi się litera „L” – ang. „load”, czyli załadowanie. Oznacza to że system czeka na dane z komputera PC. Dane przesyłane są przez port szeregowy w formacie Intel-HEX. Format ten jest najbardziej rozpowszechnionym wśród systemów opartych na mikrokontrolerach 8-bitowych. Plik w takim formacie generowany jest

przez każdy program kompilatora na procesory rodziny MCS-51. Sposób generowania takiego pliku opiszę w dalszej części artykułu. Zainteresowanych szczegółowym opisem formatu pliku Intel-HEX odsyłam do lektury naszego bratniego pisma Elektroniki Praktycznej nr 10/97.

W przypadku braku transmisji z PC ta przez około minutę system wyświetli komunikat „Err” – ang. „error, błąd. Taki sam efekt wystąpi jeżeli transmisja będzie zawierała błędy, lub wysyłany z komputera PC plik nie będzie w formacie Intel-HEX.

W przypadku komunikatu o błędzie należy wcisnąć klawisz „M” – powrotu do monitora.

W fazie wysyłania danych przez komputer PC wyświetlacz pokazuje adres aktualnie ładowanego rekordu z danymi, co jednocześnie jest dla nas sygnałem o prawidłowej transmisji. Po zakończeniu systemu automatycznie przechodzi do pętli głównej monitora, czyli do oczekiwania na kolejną komendę użytkownika.

**SEND** (klawisz 8) – dzięki tej funkcji można przesłać zawartość całej lub wybranej części pamięci operacyjnej do komputera PC, celem jej przechowania. Pod pojęciem pamięci operacyjnej mam na myśli zewnętrzną 64kB przestrzeń adresową procesora 8051 odczytywaną tak jak zewnętrzna pamięć danych – czyli poprzez sygnał /RD procesora. W praktyce będzie to zawartość pamięci SRAM (U4) umieszczonej na płycie bazowej komputera. Po wywołaniu tej funkcji na pierwszym wyświetlaczu pojawi się literka „S” – ang. „send”, czyli wysyłanie, a po chwili na czterech ostatnich wyświetlaczach pojawi się napis: „- - - -”, co jest zachętą do wprowadzenia adresu początku obszaru który chcemy wysłać. 16-bitowy adres wprowadzamy (tak jak wszystkie dane) w postaci heksadecymalnej, używając klawiszy „0...9, A...F”, a więc np. „8205”. Po wprowadzeniu adresu początku musimy jeszcze w podobny sposób wprowadzić adres końcowy wysyłanego obszaru, np. „84FF”.

Następnie wciśnięcie klawisza „OK” rozpoczyna transmisję danych. Wyświetlacz podobnie jak w przypadku funkcji „LOAD” będzie pokazywał postęp w transmisji wyświetlając aktualnie wysyłany adres rekordu (paczki danych).

Przykład: wciśnięcie sekwencji: „8” (klawisz SEND), „8200” (adres początkowy), „85FF” (adres końcowy), „OK” spowoduje wysłanie danych z obszaru pamięci operacyjnej o adresach: 8200h...85FFh.

**JUMP** (klawisz 0) – jest to komenda skoku pod wskazany adres w zewnętrznej pamięci programu procesora. Ponieważ kostka SRAM – U4 spełnia także rolę ta-

kiej pamięci (poprzez iloczyn sygnałów /RD i /PSEN) dzięki tej funkcji można uruchomić załadowany wcześniej program użytkownika. Załadowany tzn. przesłany z komputera PC za pomocą komendy „LOAD” lub wpisany ręcznie, a w jaki sposób to za chwilę.

Wciśnięcie klawisza „JUMP” powoduje zapalenie na pierwszym wyświetlaczu litery „J” – ang. „jump”, skocz, po czym system oczekuje na wprowadzenie (podobnie jak w „LOAD” i „SEND”) 16-bitowego adresu pod który ma nastąpić skok. Adres ten z reguły jest początkiem programu użytkownika. Razem z „LOAD” funkcja ta będzie najczęściej używaną podczas ćwiczeń pisania programów na nasz systemik. Wpisanie adresu skoku a następnie potwierdzenie tego klawiszem „OK” rozpoczyna wykonywanie programu od tego właśnie adresu. Po włączeniu naszego komputera pamięć U4 zawiera z reguły przypadkowe dane, toteż wywołanie funkcji „JUMP” z podaniem adresu np. „8000” (zworka JP3) najprawdopodobniej spowoduje zawieszenie się systemu będące wynikiem przypadkowej interpretacji danych znajdujących się w kości SRAM. Efektem będzie prawdopodobnie wygaszony wyświetlacz. Wtedy może pomóc tylko wciśnięcie klawisza „RESET”.

**BAUD** (klawisz B) – funkcja pozwala na zmianę domyślnej prędkości transmisji poprzez port szeregowy. Standardowo po włączeniu zasilania system odbiera i wysyła dane do komputera PC z prędkością 4800 bodów (bitów na sekundę). Prędkość tę można jednak zwiększyć lub zmniejszyć. Standardowymi prędkościami przyjętymi w asynchronicznym przesyłaniu danych poprzez interfejs RS232c są: 1200, 2400, 4800, 9600, 19200 bodów. Możliwe są też większe prędkości: 38400, 57600, 115200. W naszym systemie można wybrać wszystkie wartości poza tą ostatnią.

Wciskając sekwencyjnie klawisz „BAUD” wyświetlacz DL1 pokazuje literkę „b” – ang. „baud”, natomiast na pozostałych ukazuje się następna prędkość transmisji w kolejności jak wypisano wcześniej. Na początku mniej wtajemniczonym radzę nie eksperymentować i pozostawić wartość 4800 bodów. Aby zakończyć i wyjść do pętli głównej monitora należy wcisnąć klawisz „OK” lub „M”.

**COPY** (klawisz C) – funkcja pozwala na kopiowanie wybranego obszaru pamięci operacyjnej (kość U4) pod inny podany adres. Wybranie tej funkcji powoduje pojawienie się litery „C” – ang. „copy”, kopiowanie na DL1, a następnie system czeka na podanie adresu początku oraz końca obszaru który ma być skopiowany, oraz na adres przeznaczenia, czyli adres

## Projekty AVT

pod który ma być skopiowany wskazany obszar. Rzecz jasna nie jest konieczne podawanie adresu końca przeznaczenia – jest on określony przez ilość bajtów wyznaczoną z różnicy końca i początku adresu źródłowego. Równanie poniżej obrazuje sposób wykonania operacji:  
*Adres\_startowy\_źródła ..... Adres\_końcowy\_źródła* —> *Adres\_startowy\_przeznaczenia*.

Funkcja ta nie spełnia tak ważnej roli jak np. opisane trzy pierwsze, lecz w naszych przykładach podczas dalszych części kursu będziemy z niej korzystać.

Przykład, wciśnięcie sekwencji: „C” (klawisz COPY), „8600” (adres początku źródła), „87EF” (adres końca źródła), „9500” (adres przeznaczenia), „OK” spowoduje skopiowanie obszaru pamięci operacyjnej o adresach: 8600h...87EFh do obszaru o adresach: 9500h...96EFh.

**EDIT** (klawisz E) – funkcja edycji (modyfikowania) poszczególnych komórek pamięci operacyjnej systemu (kostka U4). Bardzo ważna funkcja szczególnie dla Tych z Was, którzy nie posiadają jeszcze komputerów. Dzięki niej możliwe jest wpisanie własnego programu – krok po kroku a raczej „komórka po komórce”, po czym jego uruchomienie. Wciśnięcie klawisza EDIT powoduje zapalenie na DL1 litery „E” – ang. „edit”, edycja, po czym system czeka na podanie adresu początku obszaru który chcemy modyfikować. Po wprowadzeniu go można przystąpić do modyfikacji. Podczas tego cztery pierwsze wyświetlacze pokazują adres, a dwa ostatnie zawartość aktualnie modyfikowanej komórki. Adres można zmniejszać lub zwiększać używając klawiszy „<” (klawisz 4) lub „>” (klawisz 6), by zmo-

dyfikować daną komórkę trzeba nacisnąć klawisz „OK”. Dwa ostatnie wyświetlacze pokażą wtedy napis „-” co jest chęcią do wpisania nowej wartości.

Aby zakończyć edycję należy wcisnąć podobnie jak poprzednio klawisz „M”. Jeżeli masz przed sobą działający systemik, radzę spróbuj zmodyfikować kilka komórek pamięci. Pamiętaj tylko aby podać adres początkowy taki, pod którym znajduje się dowolna komórka z kostki U4 – spójrz na zwróć JP3. Oczywiście jeżeli np. początkowy adres edycji będzie leżał poza obszarem pamięci SRAM – U4, to nowo wpisana wartość pod taki adres będzie stracona.

**FILL** (klawisz F) – dzięki tej funkcji możliwe jest szybkie wypełnianie wskazanego obszaru pamięci operacyjnej daną wartością. Po wciśnięciu klawisza „FILL” DL1 pokaże literkę „F” – ang. „fill”, wypełnij, po czym poprosi o wprowadzenie adresu początku i końca obszaru który ma być wypełniony, po czym należy jeszcze wprowadzić wartość wypełnienia – stałą (8-bitową oczywiście). Wciśnięcie klawisza „OK” spowoduje wypełnienie zadanego obszaru.

Przykład, sekwencja: „F” (klawisz FILL), „8000” (adres początku), „FFFF” (adres końca), „00” (wartość wypełnienia), „OK”, spowoduje wyzerowanie wszystkich komórek pamięci U4 które znajdują się w obszarze o adresach 8000h...FFFFh.

Poza opisanymi funkcjami dostępna jest też procedura „TEST”, której znaczenie już znacie drodzy Czytelnicy z poprzedniej części niniejszego artykułu.

Podczas używania wszystkich funkcji należy pamiętać że ich działanie w każdej chwili można przerwać klawiszem „M”. Istotną informacją jest też

## Wykaz elementów

### Rezystory

R1: 510...680Ω

### Kondensatory:

C1, C2: 1000uF/16V

C3: 470uF/10V

C4: 470nF...1000nF

### Półprzewodniki:

M1: mostek 1A/50V

U1: 7805

D1: LED dowolna

### Pozostałe:

B1: bezpiecznik 160mA

TR1: TS6/49 (patrz tekst)

JP1: ARK-2

oprawka bezpiecznika do druku

plytka drukowana AVT-2250Z

### Uwagi

Kompletny kit zasilacza dostępny jest w ofercie handlowej jako kit AVT-2250/Z.

Dyskietka z oprogramowaniem na komputery PC dostępna jest jako AVT-2250/D.

Niezbędne wtyki do wykonania kabla połączeniowego można zamówić indywidualnie jako: „Wtyk DB9/F z obudową” lub DB25 z obudową w dziale obsługi czytelników.

ta, że nieumiejętne czy przypadkowe posługiwanie się funkcjami COPY, JUMP, FILL, EDIT może spowodować zawieszenie się naszego komputerka. Lecz nic to, warto na początku trochę porozrabiać i poćwiczyć, wszakże zawsze pod ręką jest klawisz „RESET”, a my nic ważnego jeszcze nie napisaliśmy jeżeli chodzi o programowanie samego procesora 8051.

Oto kilka ćwiczeń, które radzę wykonać zarówno tym którzy mają jak i nie mają komputera PC w swoim domu. Nie jest to wprawdzie już programowanie procesora 8051, ale ćwiczenie w nabyciu umiejętności posługiwania się funkcjami monitora systemu AVT-2250.

# Lekcja 1

## Ćwiczenie 1

Zadanie ma na celu poznanie funkcji edycji pamięci operacyjnej „EDIT”. Proszę ustawić jumper JP3 w pozycji 8000h (początek pamięci SRAM U4) na płytce bazowej komputerka. Następnie proszę o wpisanie pod kolejne adresy poczynając od adresu 8000h następującej sekwencji danych :

„ 75, 78, 06, 75, 79, 5B, 75, 7A, 4F, 75, 7B, 66, 75, 7C, 6D, 75, 7D, 7D, 75, 7E, 07, 75, 7F, 7F, 75, 71, 55, 80, FE ”

Ostatnia dana „FE” powinna znaleźć się pod adresem „801C”.

Kolejnym krokiem będzie np. zresetowanie systemu klawiszem „RESET”, po

czym sprawdzenie poprzez edycję, że wpisane dane są na swoim miejscu w kolejności jak podano wyżej.

Nie wyłączamy zasilania układu i bierzemy się do następnego ćwiczenia.

## Ćwiczenie 2

Zadanie ma na celu poznanie działania funkcji kopiowania „COPY” wybranych obszarów pamięci operacyjnej komputerka. Proszę o skopiowanie obszaru o adresach: 8000h...801Fh pod adres 9400h. Następnie proszę skorzystać z funkcji EDIT, sprawdź, że dane skopiowane pod adres docelowy 9400h są takie same jak podano w sekwencji w ćwiczeniu nr 1, aż do adresu 941Ch.

## Ćwiczenie 3

Zadanie pokaże jak szybko można wypełnić zadany obszar pamięci za pomocą funkcji

FILL. Proszę o wypełnienie obszaru o adresach 8000h...802Fh wartością wybraną przez siebie, a następnie sprawdzenie za pomocą funkcji EDIT czy dane spod tego obszaru, wpisane zresztą w ćwiczeniu nr 1, zostały zamazane wartością podaną przez Ciebie.

I wreszcie chyba najciekawsze ćwiczenie, wykorzystujące funkcję JUMP.

## Ćwiczenie 4

Proszę wykonać skok pod adres 9400h korzystając z funkcji „JUMP”.

## Podsumowanie

Prawidłowe wykonanie tego ćwiczenia oraz trzech poprzednich w efekcie powinno spowodować zapalenie na wyświetlaczach DL1...DL8 kolejnych cyfr od „1” do „8” z migającymi cyframi nieparzystymi, czyli: 1, 3, 5, 7. Jeżeli jesteś ciekawy co takiego się stało, to chcę Cię

poinformować, że **wpisałeś i wykonałeś pierwszy program na procesor 8051!!!**

Zadaniem jego było wyświetlenie kolejnych cyfr na wyświetlaczu z migotaniem nieparzystych. I choć to program niczym nie przydatny, to równie dobrze mógł być to program sterowania całego systemu alarmowego w Twoim mieszkaniu albo.... możliwości jest niezliczona ilość. I choć kilka kroków było zbędnych aby wykonać ten program (wystarczyłoby ćwiczenie nr 1 i 4) oraz w zasadzie nie wiedziałeś jaką rolę pełnią poszczególne liczby w sekwencji z ćwiczenia 1, to jednak powinieneś mieć już pojęcie o tym jak łatwo można zmusić jeden układ elektroniczny do wykonywania wielu funkcji i to w zależności do naszych potrzeb.

## Dla komputerowców

Na zakończenie krótkie ćwiczenie sprawdzające działanie portu transmisji szeregowej. Posiadacze komputerów PC mogą nabyć dyskietkę startową z programami niezbędnymi do tworzenia zbiorów z kodem maszynowym napisanego przez siebie programu. Dyskietka posiada symbol handlowy: AVT-2250 /D i można ją nabyć w dziale obsługi czytelników AVT.

Na dyskietce znajduje się assembler na procesory 8051 oraz dwa zbiory tekstowe z definicjami rejestrów specjalnych oraz adresów wykorzystane w komputerku AVT-2250: CONST.INC i BIOS.INC. Znaczenie ich zostanie wyjaśnione w następnym odcinku szkoły mikroprocesorowej. Przejdźmy zatem do sprawdzenia RS (eresy).

## Ćwiczenie 5

Jeżeli wykonałeś poprawnie poprzednie ćwiczenia, wyłącz zasilanie systemu AVT-2250. następnie podłącz komputer PC do systemiku za pomocą wykonanego kabla transmisji szeregowej (nie musisz wyłączać przy tym swego PC ta).

Jak wspomniałem wcześniej parametry transmisji z naszym systemem mikroprocesorowym są następujące: prędkość: 4800 bodów, bez kontroli parzystości, 8 bitów danych, 1 bit stopu.

Takie parametry wolnego portu COM w PC cie można ustawić z np. poziomu DOS a za pomocą polecenia:

```
> MODE COMx:4800,n,8,1
```

gdzie oczywiście w miejsce „x” należy wpisać numer portu COM dołączonego do naszego systemu.

Na dyskietce znajduje się plik wsadowy „USTAWCOM.BAT” który ustawia takie parametry. Jeżeli planujesz kontynuację nauki programowania przy pomocy systemu AVT-2250, powinieneś umieścić takie polecenie, ustawiające port COM w pliku AUTOEXEC.BAT. Zaoszczędzi Ci to każdorazowego ręcznego wywołania pliku

USTAWCOM.BAT po rozpoczęciu pracy z Twoim systemem mikroprocesorowym.

Dyskietka zawiera także drugi plik wsadowy „DO.BAT”, który służy do kompilowania zbiorów źródłowych napisanych w języku assemblera 8051 na zbiór wynikowy akceptowany przez system AVT-2250. Więcej szczegółów na temat assemblera możesz dowiedzieć się z artykułu o mikrokontrolerach z tego numeru EdW – radzę uważnie przeczytać.

Na dyskietce znajduje się także przykładowy program w postaci źródłowej w zbiorze „PRZYK01.S03”. Przyjmijmy rozszerzenie „S03” jako standardowe dla naszych programów assemblerowych.

Po podłączeniu systemu do komputera PC, włącz jego zasilanie. Ustaw parametry portu COM jak opisałem wcześniej. następnie wywołaj polecenie (w DOS):

```
> DO PRZYK01 COMx {Enter}
```

wpisując w miejsce „x” numer portu szeregowego (1,2...).

Program zamieni zbiór assemblerowy na postać wykonywalną – zbiór PRZYK01.HEX, następnie zrobi pauzę, podczas której powinieneś uruchomić w systemie funkcję „LOAD” .

Teraz naciskając dowolny klawisz w PC cie spowodujesz przesłanie zbioru

PRZYK01.HEX do portu COMx, czyli do systemu AVT-2250.

Program który wpisałeś rozpoczyna się od adresu 8000h w pamięci systemu, a efekt jego działania powinien być taki sam ja w przykładzie z ćwiczenia nr 1.

Aby się o tym przekonać uruchom funkcję „JUMP” i sprawdź co pokaże wyświetlacz.

Przerwij działanie programu (klawisz „M”) i porównaj, korzystając z funkcji „EDIT” zawartość kolejnych komórek pamięci począwszy od adresu 8000h (do 801Ch włącznie), z kodami instrukcji podanymi w ćwiczeniu nr 1.

Jeżeli wszystko przebiegło pomyślnie test portu szeregowego możesz uznać za zakończony. Jeżeli nie udało Ci się wykonać ćwiczenia prawidłowo, sprawdź, czy dołączyłeś system do właściwego portu COM twego PC ta, lub czy prawidłowo wykonałeś kabel połączeniowy.

Pozostałe lekcje zajmujące się już prawdziwym programowaniem kontrolerów 8051 znajdują się już w następnym numerze EdW w cyklu „Mikrokontrolery – to takie proste”.

**Sławomir Surowiński**

**\* Uwaga!** W cz.I artykułu (EdW 8/97) na str. 11, szpalta 3, wiersz 10 od dołu jest „U7”, powinno być „U15”.