

# 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 to nabycia na giełdach za kilkadziesiąt zł) trzeba będzie nabyć.

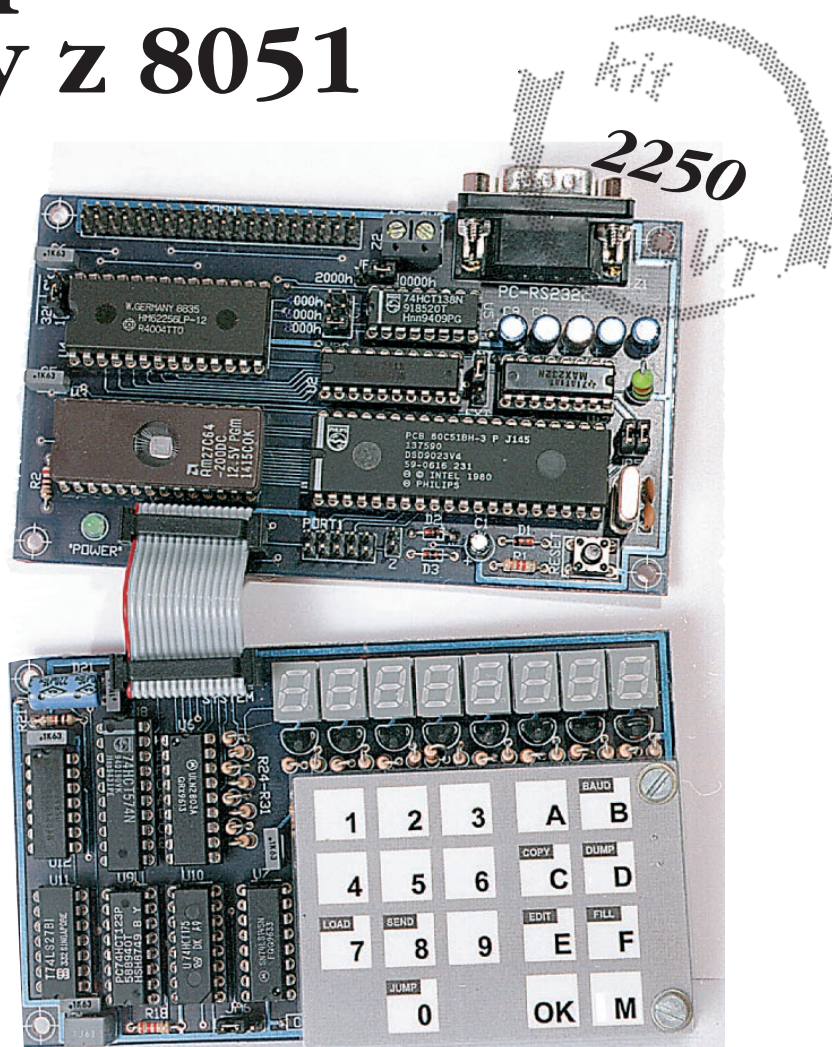
Przy projektowaniu autor kierował się maksymalnym uproszczeniem całej konstrukcji do niezbędnego minimum, pozwalającego jednak na swobodne „surfowanie” po '51-ce, z wykorzystaniem wszystkich jego możliwości, opisywanych w cyklu teoretycznym w EdW. Niemalże znaczenie miało tu też zminimalizowanie kosztów. Niestety nie udało się nie przekroczyć bariery kilkudziesięciu złotych, lecz ten jednorazowy poniesiony wydatek w przyszłości z pewnością przyniesie wymierne efekty, czego najlepiej 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.

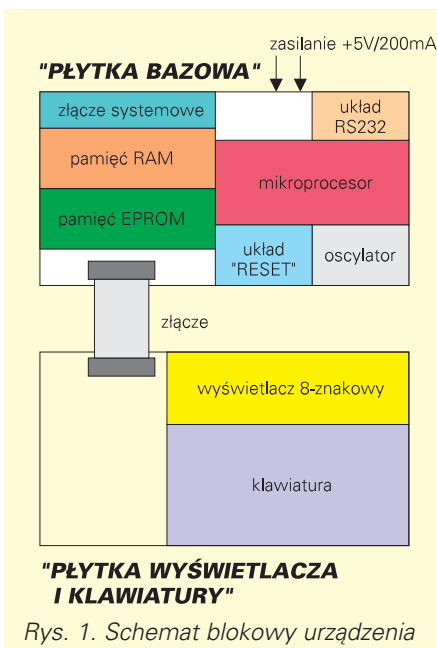
I jeszcze jedna uwaga. Być może dla niektórych z Was prezentowany układ

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, wyjaśnię sobie wiele rzeczy, co do których możecie mieć wątpliwości. Ja hołduję zasadzie, że „najlepszą nauką jest praktyka...”, a tej jest najwięcej w elektronice, wiercie 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 – płyta 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 bazową

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 chwilę 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).

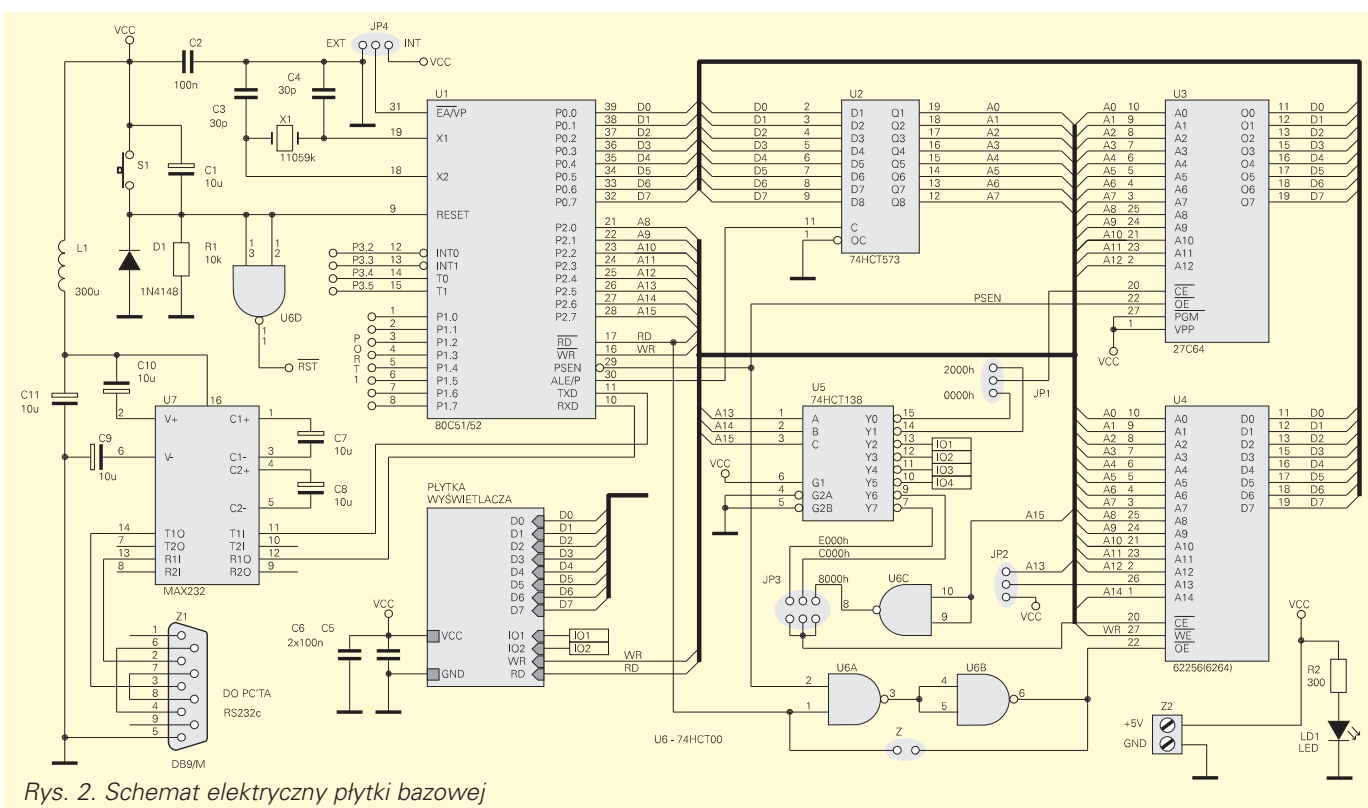
Popatrzmy, z lewej strony u góry widzimy znajomy układ zewnętrznego obwodu rezonatora kwarcowego X1 wraz z kondensatorami C3 i C4. Znaczenie tych elementów już znasz. Nieco niżej widać układ „resetu” złożony z elementów C1-S1 oraz dodatkowo R1 i D1. Klawisz S1 służy do ręcznego resetowania całego komputerka, 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 złącz znajdujących się na płytce drukowanej przedstawimy w części opisującej montaż układu. Końcówki 12...15 procesora oraz cały port 1 (końc. 1...8) pozostają niedołączone 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 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 oczywiście 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 zewnętrznej 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 odgięcie 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 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 diodowy L1 zapobiega przedostawianiu 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.

### Płytki 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 (klawiatura). 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 zasilenie 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 z adresuje 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 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 pojawiają się logiczne zera, bo:

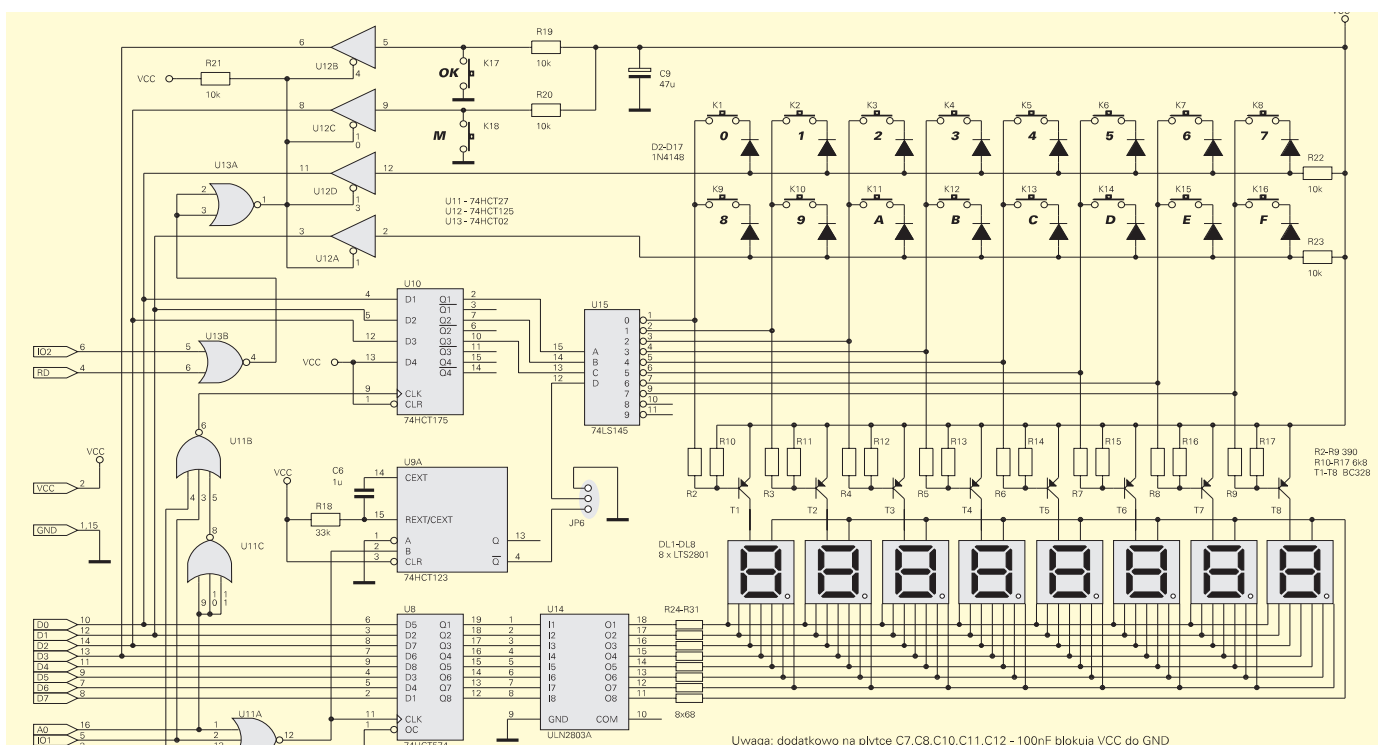
- sygnał A0=0 bo adres = 4000h
- sygnał /WR=0 bo procesor zapisuje
- sygnał IO1=0 bo dla adresu 4000h wyjście dekodera U5 (rys.2) IO1 jest =0.

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 ustalonym wartością 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