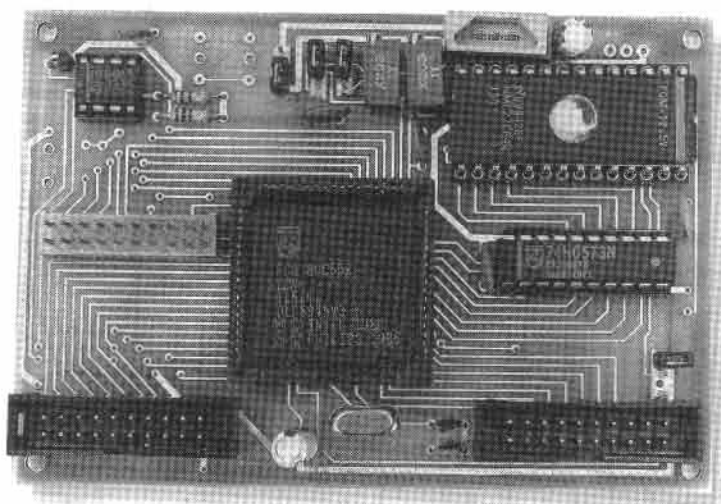


Płytki bazowa mikrokontrolera 80C552, część 1

kit AVT-280

Panująca wśród wielu entuzjastów elektroniki obiegowa opinia na temat sterowników procesorowych sprowadza się do przekonania, że jest to temat zbyt trudny i nieciekawym, by warto nim było zwracać sobie głowę i tracić czas. Przy okazji prezentacji nowej płytki prototypowej będziemy chcieli zmienić tę opinię przekonując, że wykorzystanie i programowanie procesorów jednoukładowych nie jest takie trudne, a osiąga się rewelacyjne efekty, przy nieskomplikowanym układzie elektronicznym.



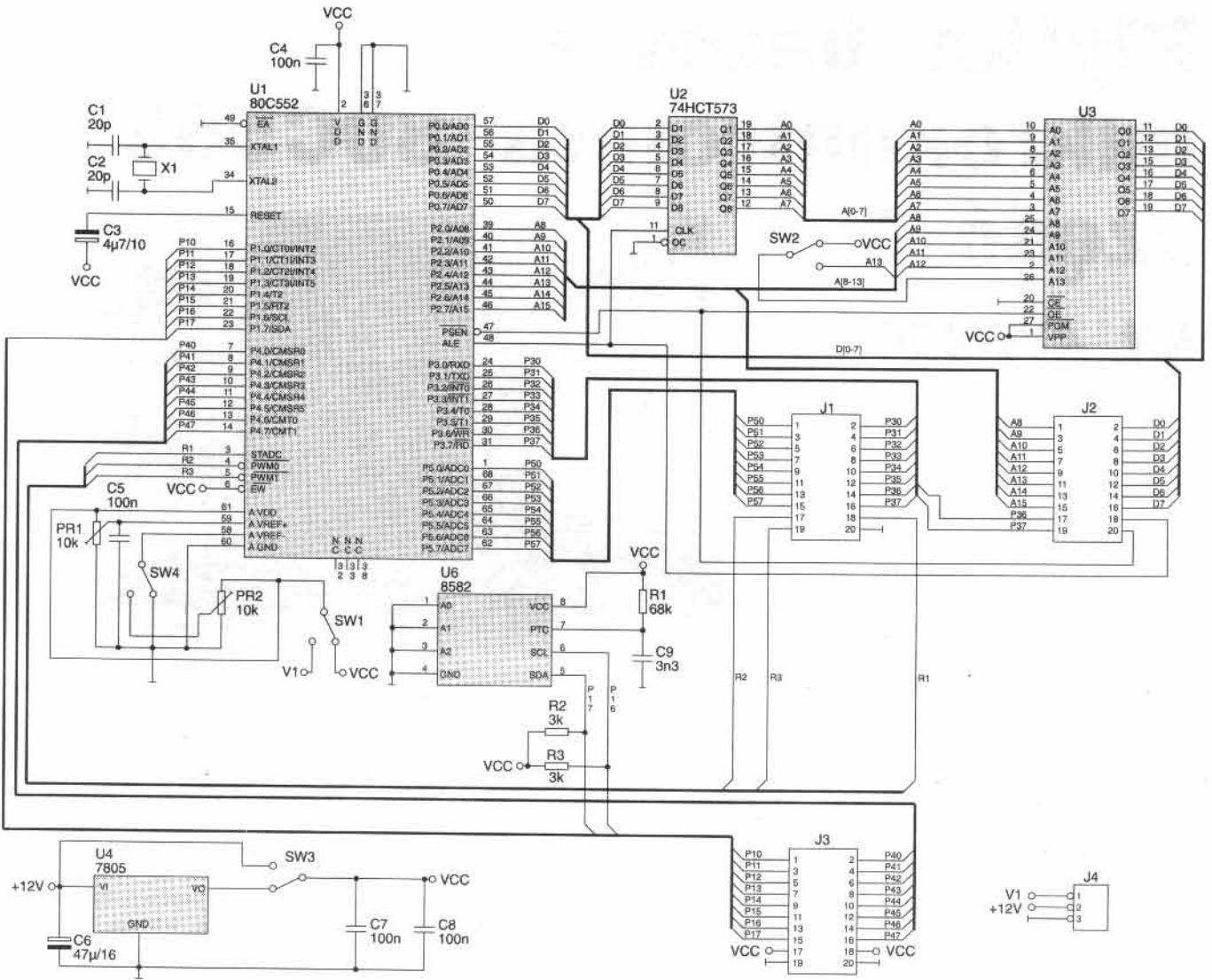
Sterowniki procesorowe szeroką falą wlewają się w nasze codzienne życie. „Braci” mniejszych procesorów z komputerów osobistych znaleźć dziś można nie tylko w specjalistycznym sprzęcie, ale także w wielu nowoczesnych samochodach, telewizorach, kuchenkach mikrofalowych czy w zestawach audio. Istnieje wiele ich typów, od uniwersalnych po ściśle specjalizowane. Dobrze jest mieć chociaż ogólne pojęcie o ich możliwościach. Niektóre, jak 8051, PIC, czy procesory serii ST62 już pojawiały się na łamach EP. Prezentowana płytki wykorzystuje procesor 80552, który jest kolejnym elementem linii rozwojowej 8051. Przy okazji opisu płytki zajmiemy się również nowymi możliwościami kostki w stosunku do jej znanego poprzednika. A ponieważ czytanie wyłącznie o rejestrach, bitach i portach byłoby nudne, zaprezentowanych zostanie kilka prostych, ale praktycznych programów wykorzystujących możliwości procesora. Połączenie płytki prototypowej z kilkoma dodatkowymi elementami zewnętrznymi przekształci ją w prosty woltomierz, miernik refleksu, częstościomierz lub zegar.

Poniższy opis przeznaczony jest głównie dla osób, które wcześniej zetknęły się z mikrosterownikiem 8051, ale skorzystają z niego

także kompletni nowicjusze uzupełniając brakujące informacje w licznych publikacjach na temat '51 (np. w zeszytach USKA).

Procesor 80552 wyposażony został w 5 portów, czyli układów wejścia/wyjścia, poprzez które komunikuje się on ze światem zewnętrznym. W porównaniu z procesorem 8051 nowością jest port 4 (P4), który może być wykorzystany jako port uniwersalny lub do obsługi alternatywnych funkcji związanych z TIMEREM 2. Dodatkowy port 5 (P5) jest tylko portem wejściowym, ale można do niego podłączać także sygnały analogowe. Kolejną różnicą w stosunku do poprzedników jest zestaw alternatywnych funkcji portu 1 (P1), związanych ze znacznie rozbudowanym systemem przerwań. Oprócz tego układ posiada kilka wyprowadzeń związanych z przetwornikiem analogowo-cyfrowym.

Budowa płytki odzwierciedla sprzętowe możliwości procesora, jej schemat ideowy pokazuje rys.1. W celu zminimalizowania wymiarów płytki ograniczono otoczenie procesora do niezbędnego minimum. 80552 wykonuje program zapisany w zewnętrznej pamięci EPROM. U3 to podstawka z EPROM-em, zatrask U2 jest niezbędny przy odczycie przez procesor kolejnych bajtów programu. Gniazda J1-J3 logicznie grupują

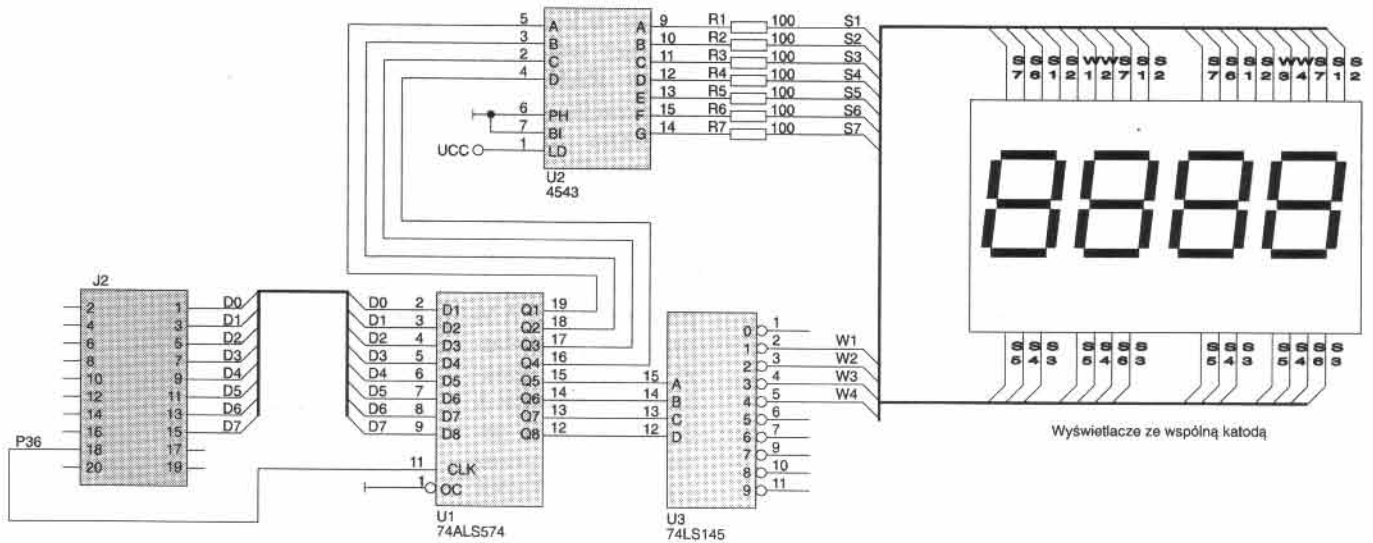


Rys. 1. Schemat elektryczny płytki bazowej.

poszczególne wyprowadzenia procesora. Do gniazda J2 doprowadzono wszystkie sygnały potrzebne do podłączenia zewnętrznej pamięci RAM lub niestandardowych urządzeń wejścia/wyjścia. Gniazdo J1 grupuje wyprowadzenia portów P3, P5 oraz wszystkich pinów związanych z funkcjami przetwarzania analogowego. J3 to porty P1 i P4, których funkcje łączy z działaniem TIMERA 2. Miejsca na gniazda umożliwiają wlotowanie podstawek i wtyków dla taśmy 20-żyłowej, mogącej połączyć wybrane porty z elementami peryferyjnymi montowanymi na dodatkowej płytce lub płytkach. Umożliwia to praktyczne wykorzystanie układu jako bloku sterującego w konkretnych urządzeniach. Do J4 podłączone jest zasilanie i analogowe napięcie odniesienia V1.

Elementami konfiguracyjnymi płytki są jumpery SW1-3. Ustawienie SW2 pozwala na stosowanie różnego typu EPROM-ów. Zwarcie 1-2 odpowiada 2732, 3-2 27128, brak zwory - 2764. Krótsze obudowy (np. w przypadku 2732) należy umieścić w podstawie tak, aby skrajne wyprowadzenia 12 i 13 znalazły w zaciskach podstawki 14 i 15. Jumperem SW1 wybiera się dodatnie analogowe napięcie odniesienia, w położeniu 1-2 jest to napięcie Vcc zasilające procesor, w położeniu 3-2 napięcie V1. SW4 wybiera ujemne napięcie odniesienia, 1-2 jest to masa układu, a w położeniu 3-2 napięcie z suwaka PR2. Jeżeli płytka zasilana będzie napięciem stabilizowanym +5V jumper SW3 powinien znaleźć się w pozycji 3-2, w przeciwnym położeniu do stabilizacji wy-

korzystywany będzie układ U4. Oprócz tego na płytce znajdują się jeszcze potencjometry PR1 i PR2 do precyzyjnego ustawienia napięcia odniesienia, elementy rezonatora X1, C1, C2, kondensator resetu C3 i kondensatory filtrujące. Jedynym układem nie związanym bezpośrednio z funkcjonowaniem procesora jest U5, w którego podstawie można osadzić pamięć EEPROM lub zegar czasu rzeczywistego. Wykorzystanie procesora jednoukładowego ma niewątpliwie wiele zalet, ale wymaga na początku kilku niezbędnych inwestycji. Po pierwsze, trzeba mieć dostęp do komputera osobistego, najlepiej klasy PC. Po drugie, trzeba posiadać program asemblerujący, który zamieni instrukcje kodu źródłowego pisane w języku symbolicznym na binarny kod



Rys. 2. Proponowany sposób podłączenia wyświetlacza do płytki bazowej.

procesora. Wszystkie przykłady podane w postaci mnemoników asemblera. Programy można także pisać w języku wyższego poziomu, np. C, a potem kompilować do postaci binarnej, ale jeśli chce się stworzyć naprawdę efektywne fragmenty programu o zwartym kodzie i dużej szybkości działania i tak trzeba sięgnąć po asembler. Trzecią koniecznością jest posiadanie programatora EPROM-ów, lub ich symulatora, aby móc uruchomić napisany program. W tym przypadku wybór wariantu zależy od możliwości i zasobności kieszeni, chociaż autor preferuje rozwiązanie pierwsze, eliminujące kłopotliwe przewody i bliższe ostatecznego kształtu układu, nad którym się pracuje. Wreszcie potrzebna jest płytka prototypowa procesora. Wszystkie wspomniane wyżej komponenty, z wyjątkiem komputera, znajdują się w ofercie handlowej AVT.

Dostępne na rynku asemblery różnią się niekiedy znacznie, jeśli chodzi o sposób posługiwania się nimi, możliwości i akceptowaną składnię programów źródłowych. Najlepiej gdyby asembler rozpoznawał nazwy symboliczne 80552 podane w dokumentacji źródłowej producenta procesorów. Niektóre asemblery nie rozpoznają nazw symbolicznych, a odwołanie się do konkretnych adresów, które w starym 8051 są nieużywane powoduje pojawienie się komunikatu błędu, chociaż program przeprowadza asemblację kwestiono-

wanej linii. W takim przypadku można zlekceważyć te komunikaty błędów. Jeśli jednak używamy asemblera dla 8051, który zdecydowanie odmawia asemblacji nieznanemu mu instrukcji 80552, to można obejść ten problem wpisując zamiast odrzuconej instrukcji jej kod przy pomocy dyrektywy DB. Np. instrukcję MOV P4,#0 (czyli prześlij pod adres 0C0h wartość 0), można zastąpić zapisem DB 075h,0C0h,0h. Najlepiej na początku przeprowadzić próbną asemblację kilku „podejrzanych” rozkazów i zorientować się jak reaguje na nie asembler na którym pracujemy.

Wartości liczbowe będą prezentowane w następujący sposób: litera -h- na końcu oznacza liczbę w zapisie hexadecymalnym, litera -b- oznacza zapis binarny z najstarszym bitem z lewej strony, liczba bez znaku to liczba dziesiętna.

Podobnie jak w '51, cechą charakterystyczną 80552 jest sposób traktowania pamięci. Procesor różni i adresuje jej cztery rodzaje: pamięć programu o max. rozmiarze 64kB (tylko do odczytu), zewnętrzną pamięć RAM, która może być dołączana w maksymalnej wielkości 64kB, wewnętrzną pamięć RAM o rozmiarze strony (256 bajtów) i obszar SFR czyli rejestrów o specjalnym przeznaczeniu, których adresy 080h do 0FFh pokrywają się z częścią wewnętrznego RAM-u. Mogą być jednak odczytywane i zapisywane przy pomocy trybu adresowania

bezpośredniego. Wartości liczbowe wpisywane do odpowiednich rejestrów decydują o sposobie działania procesora. Np. żeby ustawić poziom niski na wszystkich pinach portu 1 wystarczy wpisać do rejestru SFR o adresie 090h liczbę 0. Takie ujednoczenie sposobu sterowania pracą procesora ułatwia pisanie programów. 80552 wzbogacony został o szereg nowych możliwości: 10-bitowy przetwornik analogowo-cyfrowy, dodatkowy timer o numerze 2, zwiększoną liczbę przerwań sprzętowych, interfejs szeregowy szyny I2C. Sterowanie nimi odbywa się przy pomocy dodatkowych rejestrów SFR, które umieszczone zostały pod adresami nie używanymi przez 8051. Sposób posługiwania się tymi rozszerzeniami wraz z przykładami programów użytkowych w których zastosowano wspomniane nowe możliwości procesora przedstawione zostaną w dalszej części opisu.

Najpierw trzeba jednak stworzyć układ, który będzie informował programistę o efektach działania programu i procesora. Najprostszym sposobem jest podłączenie do płytki prototypowej układu 4-cyfrowego wyświetlacza LED, którego schemat przedstawiono na rys. 2. Wyświetlaniem informacji na wyświetlaczach o wspólnej katodzie steruje układ U2. Ponieważ wyświetlacze są multipleksowane, tzn. że w danej chwili świeci tylko jeden, ich kolejnym zapalaniem steruje multiplexer U3. Dane o wyświetla-

```

.....
; * LISTING 1
.....
; * Przykład programu obsługującego 4 segmentowy wyświetlacz *
.....
org 0
using 0
wys1 equ 051h
wys2 equ 052h
wys3 equ 053h
wys4 equ 054h
nrwys equ 055h
zeg2ms equ 056h

jmp start
org 0bh
jmp wyswietlacz ;wektor przerwania timera0
start:
mov wys1,#1 ;ustawienie parametrów początkowych
mov wys2,#2
mov wys3,#3
mov wys4,#4
mov nrwys,#4

mov tmod,#2h ;zaprogramowanie timera0 jako 1 bajtowego
;z automatycznym 3adowaniem
mov t10,#56 ;timer0 w po3czeniu z rejestrem zeg2ms
mov th0,#56 ;będzie odmierza3 czas 2ms
mov zeg2ms,#10
setb et0 ;zezwolenie na przerwanie timera0
setb tr0 ;w3czenie timera0
setb ea ;ogólne zezwolenie na przerwanie

program: jmp program ;od tego miejsca będą zaczyna3y się
;programy użytkowe

wyswietlacz: ;podprogram przerwania timera0
dinz zeg2ms,w2 ;sprawdzenie czy up3yn3 już czas 2ms
mov zeg2ms,#10
push acc ;zachowanie aktualnych wartości rejestrów
push ar0 ;użytych w podprogramie przerwania
dinz nrwys,w1
mov nrwys,#4
w1: mov a,#wys1
dec a
add a,nrwys ;obliczenie adresu rejestru zawieraj3cego
;cyfrę, która pojawi się na aktualnie
;zapalonym segmencie wyświetlacza

mov r0,a
mov a,@r0
anl a,#0fh
mov r0,a
mov a,nrwys ;formowanie bajtu steruj3cego wyświetlaniem!
swap a ;4 starsze bity to numer zapalnego
;wyświetlacza
;4 m3odsze bity to wyświetlana cyfra
movx @r0,a ;zapalenie kolejnego segmentu wyświetlacza
pop ar0 ;odtworzenie pierwotnych wartości rejestrów
pop acc ;użytych w podprogramie przerwania
w2: reti ;powrót z podprogramu przerwania

end

```

Listing 1. Listing w formacie IntelHex programu obsługującego wyświetlacz (układ wykonany wg. rys. 2).

nych cyfrach i numerze zapalnego wyświetlacza pobierane są z portu P0 i zatraskiwane w rejestrze U1 sygnałem zapisu do pamięci zewnętrznej, pobieranym z P3.6. Układ wyświetlacza można zmontować np. na oddzielnej płytce uniwersalnej i połączyć taśmą 20-żyłową z gniazdem J2 płytki prototypowej. Wydruk programu obsługującego wyświetlacz zamieszczono na **listingu 1**. Po winietce z nazwą znajduje się dyrektywa **ORG** informująca assembler, że program będzie się zaczynał od adresu 0 w pamięci programu. Dyrektywa **USING 0** informuje assembler A51, że używany będzie zestaw 0 rejestrów uniwersalnych R0-R7, w innych assemblerach dyrektywa ta powinna zostać zmieniona albo pominięta. Do obsługi wyświetlaczy program

przerwania **TIMERA 0**. Po stwierdzeniu, że upłynął już czas 2ms program oblicza numer kolejnego zapalnego wyświetlacza. Dla uproszczenia programu wyświetlacze zapalają się od 4 do 1, potem znów 4. Po obliczeniu numeru wyświetlacza wczytywana jest informacja do wyświetlenia z odpowiedniego rejestru wys1-4 i formowany bajt sterujący. Bajt ten jest zatraskiwany w U1 rozkazem zapisu do zewnętrznej pamięci RAM. Jeśli równoległe z wyświetlaczem była by fizycznie dołączona do procesora pamięć RAM to jej jeden adres musiał by być zastrzeżony dla wyświetlacza. W takim przypadku listing 1 musi być zmodyfikowany. Przed rozkazem **MOVX @R0,A** do rejestru R0 trzeba wpisać ten jeden zastrzeżony dla wyświetlacza adres z zakresu od 0 do 255.

używa 6 komórek pamięci wewnętrznej RAM. Ich lokalizacja i symboliczne nazwy nadane dla wygody programisty mogą być oczywiście zmienione. Oprócz komórek pamięci RAM program potrzebuje do swego działania **TIMERA 0** i przerwania generowanego przez ten timer. **TIMER 0** razem z rejestrem **zeg2ms** odmierza czas między zapaleniami kolejnych wyświetlaczy. Stosując rezonator 12MHz uzyskujemy czas 2ms, co przy tej liczbie wyświetlaczy zapewnia szybkie ich przełączanie.

Program na początku inicjuje rejestry wyświetlaczy oraz **TIMER 0** i włącza bity zezwolenia na przerwania. Cała obsługa wyświetlaczy znajduje się w krótkim podprogramie obsługi

WYKAZ ELEMENTÓW:

Rezystory

PR1, PR2: 10kΩ potencjometr wieloobrotowy

R1: 56kΩ

R2, R3: 3kΩ

Kondensatory

C1, C2: 22pF

C3: 4,7μF/10

C4, C5, C7, C8: 100nF

C6: 47μF/16

C9: 3,3nF

Półprzewodniki

U1: 80C552

U2: 74HGT573 lub podobny

U3: 2764 EPROM z programem demonstracyjnym

U4: 7805

Różne

X1: 12MHz kwarc

SW1, SW2, SW3, SW4: jumpery 3 pinowe ze zworami

J1, J2, J3, J1': komplet wtyków i gniazd do druku dla taśmy 20-żyłowej

wtyk i gniazdo do druku 3 piny

podstawka 68 pin

podstawka 28 pin precyzyjna

podstawka 20 pin

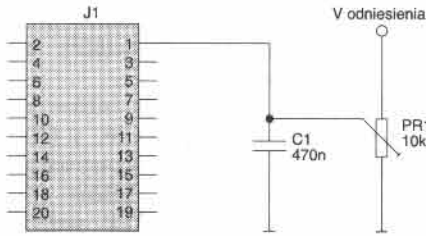
podstawka 8 pin

płytki drukowana

Opisany wyświetlacz w prosty sposób można rozszerzyć nawet do 9 segmentów. Do pracy w pełnym oświetleniu należy wtedy zastosować wyświetlacze o podwyższonej jasności.

Procesor 80552 wyposażony jest w przetwornik analogowo-cyfrowy o 10 bitowej rozdzielczości. Zakres konwersji określa dodatnie i ujemne napięcie odniesienia, ich wartości muszą mieścić się w zakresie +5V-0V. W przypadku gdy wiadomo, że napięcie wejściowe poddawane konwersji nie przekroczy np. +3V to obniżenie dodatniego napięcia odniesienia do tego poziomu pozwoli zwiększyć dokładność przetwornika. Początek kolejnej konwersji może być inicjowany programowo lub wyzwany dodatnim zboczem impulsu podawanego na końcówkę **STADC**.

Konwersja trwa 50 cykli co przy zegarze 12MHz daje czas 50ms. Dzięki multiplekserowi analogowemu na wejście przetwornika można podać wybrane wejście portu 5.



Rys. 3. Układ do testowania przetwornika A/C.

Korzystanie z przetwornika AC odbywa się przy pomocy dwóch rejestrów ADCH o adresie 0C6h i ADCON o adresie 0C5h znajdujących się oczywiście w obszarze SFR-ów. Pierwszy z wymienionych rejestrów przeznaczony tylko do odczytu zawiera 8 starszych bitów liczby otrzymanej w wyniku konwersji. Rejestr ADCON pełni funkcję sterującą i znaczenie jego poszczególnych bitów jest następujące:

b6 i b7 - pozostała część 10 bitowego wyniku konwersji przy czym b6 jest bitem najmłodszym, **b5** - jego ustawienie umożliwia start konwersji przy pomocy sygnału podanego na STADC, jego wyzerowanie blokuje tę możliwość,

b4 - flaga ustawiana przez procesor, oznacza zakończenie konwersji i dostępność wyniku,

b3 - pełni funkcję przełącznika, którego ustawienie inicjuje kolejną konwersję,

b2, b1, b0 - zapisana w tych bitach liczba binarna określa, które z wejść P5 zostanie przyłączone do przetwornika (np. ustawienie liczby 100b oznacza przyłączenie p5.4).

Korzystanie z przetwornika powinno wyglądać następująco. Najpierw należy upewnić się, że bity b4 i b3 rejestru ADCON są wyzerowane. Jeśli tak, to można wybrać wejście portu 5 i zdecydować czy konwersja będzie inicjowana sprzętowo czy programowo. W drugim przypadku ustawienie b3 rejestru ADCON rozpoczyna konwersję. O jej zakończeniu świadczy ustawienie b4 i wyzerowanie b3. Jeżeli zadawała nas 8 bitowy wynik konwersji, to możemy go odczytać bezpośrednio z rejestru ADCH w przeciwnym wypadku należy przy pomocy przesuwania bitów w lewo uformować 2 bajtową 10 bitową liczbę złożoną z bitów ADCH i pozosta-

```

;*****
;* LISTING 2
;*****
;* Przykład programu obsługującego przetwornik AC
;*****

;! fragment pominięty

hbajtd equ 057h
lbajtd equ 058h

;! fragment pominięty

program:                ;od tego miejsca zaczyna sie program uzytkowy
                        ;wykorzystujacy przetwornik AC
anl adcon,#11111000b    ;wybor do konwersji wejścia 0 multipleksera
anl adcon,#11011111b    ;start konwersji niezależny od stanu
                        ;wejścia STADC
konw1: anl adcon,#11101111b ;kasowanie flagi zakończenia
                        ;konwersji
orl adcon,#00001000b    ;start nowej konwersji AC
konw2: mov a,adcon      ;oczekiwanie na zakończenie konwersji
jnb acc.4,konw2
mov hbajtd,#0
mov lbajtd,adch
mov r1,#2
konw3: clr c            ;formowanie młodszego i starszego bajtu liczby
                        ;10 bitowej
mov a,lbajtd           ;otrzymanej po konwersji AC
rlc a
mov lbajtd,a
mov a,hbajtd
rlc a
mov hbajtd,a
djnz r1,konw3
mov a,lbajtd
mov a,adcon
anl a,#11000000b
rl a
rl a
orl a,lbajtd
mov lbajtd,a
mov r1,#0ffh
konw4: clr c           ;przekształcenie liczby 10 bitowej do postaci
                        ;dziesiętnej
inc r1
mov a,lbajtd
subb a,#100
mov lbajtd,a
mov a,hbajtd
subb a,#0
mov hbajtd,a
jnc konw4
mov a,lbajtd
add a,#100
mov lbajtd,a
mov a,r1
cjne a,#10,konw5
mov wys1,#1
mov wys2,#0
jmp konw6
konw5: mov wys1,#0
mov wys2,a
konw6: mov a,lbajtd   ;obliczanie liczby dziesiątek i
                        ;jedności

mov b,#10
div ab
mov wys3,a
mov wys4,b
jmp konw1

;! fragment pominięty

end

```

Listing 2. Program obsługi przetwornika A/C.

łych 2 bitów w ADCON. Na koniec należy wyzerować b4 i przetwornik jest znów gotów do pracy. Pracę z przetwornikiem można także realizować poprzez przerwanie, które jest wywoływane po zakończeniu konwersji. W tym celu należy ustawić bit zezwolenia na przerwanie przetwornika. Jest to bit EAD czyli b6 rejestru IEN0 (0A8h), wektor przerwania czyli adres od którego zaczyna się jego realizacja wynosi 053h.

Listing 2 zawiera krótki program wykorzystujący przetwornik AC. Na rys. 3 przedstawiono schemat układu, który trzeba do-

łączyć do płytki prototypowej. Zadanie programu polega na odczytywaniu wartości napięcia podawanego z suwaka PR1 na p5.0 i wyświetlaniu wyniku w postaci liczby z zakresu 0-1023. Jeżeli dodatnie napięcie odniesienia przetwornika ustawione zostanie na 1.023V to układ będzie wskazywał w miliwoltach wartość mierzonego napięcia.

Program z Listingu 2 należy dopisać do programu obsługi wyświetlacza z Listingu 1. Należy zadeklarować dwa nowe rejestry pomocnicze hbajtd i lbajtd. Rejestry wys1-4 można zainicjować wartością 0. Program przetwornika AC zaczyna się w tym miejscu Listingu 1 gdzie etykieta program: wskazuje na nieskończoną pętlę programu, którą trzeba usunąć.

Program zgodnie z podanym wcześniej opisem najpierw ustawia warunki pracy przetwornika. Po zakończeniu konwersji poprzez dwukrotne przesunięcie w akumulatorze otrzymanych po konwersji bitów utworzony zostaje dwubajtowy wynik zapisany w rejestrach lbajtd i hbajtd. Wielokrotne odejmowanie liczby 100, a potem podział przez 10, przekształca wynik do postaci dziesiętnej. Po zakończeniu przeliczeń konwersja rozpoczyna się od początku.

Kolejnym rozszerzeniem procesora jest dodatkowy TIMER 2. Z timerem związanych jest szereg nowych możliwości sprzętowo-programowych. Działanie TIMERA

2 różni go od pozostałych dwóch timerów. Tak jak one składa się z dwubajtowego licznika, nie można jednak wpisać do niego żadnej początkowej wartości, a jedynie wyzerować dodatnim zboczem impulsu podanego na końcówkę RT2 (p1.5). Z timerem współpracują rejestry zapamiętujące jego stan w danej chwili oraz rejestry porównujące zawartość TIMERA 2 z wartością w nich umieszczoną. Skutek tych porównań może modyfikować pracę procesora. Timer przypomina kręcące się koło, które może być zatrzymane i znów puszczane w ruch, a pojawiające się na nim liczby rosną od 0 do 0FFFFh i proces zaczyna się od nowa.

Rejestrem bezpośrednio sterującym pracą TIMERA 2 jest TM2CON (0EAh). Funkcje jego poszczególnych bitów są następujące:

b7 - jego ustawienie zezwala na przerwanie gdy TIMER 2 się przeładuje tzn. gdy zapisana w nim wartość zmieni się z 0FFFFh na 0,
b6 - jego ustawienie zezwala na przerwanie gdy młodszy bajt TIMERA 2 się przeładuje tj. gdy jego wartość zmieni się z 0FFh na 0,
b5 - ustawienie bitu oznacza zgodę na zewnętrzny reset przez podanie impulsu na końcówkę RT2,
b4 - flaga przerwania po przeładowaniu młodszego bajtu TIMER-a 2,
b3, b2 - dwubajtowa liczba binarna określająca stopień wstępnego podziału sygnału podawanego na wejście timera. Dla wartości 0 częstotliwość sygnału nie jest dzielona, dla wartości 3 sygnał wejściowy dzielony jest przez 8, pozostałe wartości oznaczają podziały pośrednie.
b1, b0 - dwubitowa liczba binarna określająca źródło sygnału podawanego na wejście timera.

0 - timer jest zatrzymany

1 - zegar procesora/12

2 - nie używany

3 - źródłem jest zewnętrzny sygnał podawany na T2 (p1.4)

Jak wspomniano wcześniej z TIMEREM 2 związane są 3 dwubajtowe rejestry porównujące. Ich nazwy oraz adresy starszego i młodszego bajtu są następujące: CM0- CMH0 (0C9h), CML0 (0A9h) CM1- CMH1 (0CAh), CML1 (0AAh) CM2- CMH2 (0CBh), CML2 (0ABh)

Zawartość rejestrów porównujących jest zerowana zewnętrznym sygnałem RT2.

Współdziałanie rejestrów porównujących z TIMEREM 2 polega na tym, że gdy zostanie stwierdzona zgodność zawartości któregoś z nich, może zostać wygenerowane żądanie przerwania. Nie jest to jednak jedyna możliwość reakcji na zgodność. Dzięki ustawieniu wybranych bitów w rejestrach RTE (0EFh) lub STE (0EEh), odpowiedni bit portu 4 może zostać automatycznie wyzerowany, ustawiony lub zmienić swój stan na przeciwny. 6 młodszych bitów rejestru STE współpracuje z CM0 i jeśli któryś z bitów STE jest ustawiony to odpowiadający mu bit P4 zostanie także ustawiony w momencie gdy zawartość CM0 będzie równa zawartości TIMERA 2. Np. wpisanie do rejestru STE wartości 3 spowoduje ustawienie

p4.0 i p4.1 (jeśli uprzednio były już ustawione to oczywiście nie się nie zmienią).

Z kolei 6 młodszych bitów rejestru RTE współpracuje z CM1 i może spowodować wyzerowanie odpowiadających im bitów P4. Dwa najstarsze bity RTE współpracują z CM2 wpływając na odpowiadające im bity P4 i zmieniają ich stan na przeciwny. Tu mała uwaga: operacja negacji nie jest przeprowadzana bezpośrednio na bitach 6 i 7 P4 lecz na dwóch wewnętrznych przerzutnikach, których stan przepisywany jest do 6 i 7 bitu portu 4. Stan przerzutników można odczytać badając bity 6 i 7 STE. Tak więc jeśli p4.7 będzie wyzerowany a b7 STE ustawiony to w wyniku negacji stan bitu portu nie zostanie zmieniony.

Ryszard Szymaniak, AVT